Lezione del 28 Febbraio 2023

Lezione 1

Questo non sarà un vero e proprio corso di statistica, ma si concentrerà sul come l'informatica si presta a ricavare informazioni dai dati.
Skill richieste per questo corso:

- Saper scrivere codice
- Possedere conoscenze matematiche di base (è consigliato aver dato matematica del continuo)
- Possedere senso critico per essere in grado di poter interpretare i risultati ottenuti

Lezioni:

- Martedì, 10:30-12:30 Aula V3
- Giovedì, 13:30-16:30 Aula Magna

Materiale didattico

- "Introduzione alla statistica", libro trattato solo per alcuni capitoli, dei quali ci vengono forniti i pdf
- "Probabilità e statistica per l'ingegneria e le scienze", preferibilmente da acquistare per questo corso, vi è anche una versione in inglese gratuita
- Dispense su github
    - File con estensione .ipynb

Sulla pagina Ariel del corso troviamo anche un calendario delle lezioni con i vari capitoli dei libri o i pdf trattati a lezione.

Esame:

- Prova scritta composta da parte scritta e parte su Python
- Prova orale

Si lavorerà in Python, linguaggio molto pratico. Consigli:

- Installarlo con environment compartimentalizzati
- Utilizzare "Anaconda"
    - Package manager
    - Fornisce automaticamente ambiente in compartimenti e tramite il file di config installa automaticamente i package richiesti
- Scaricare dalla versione 3.7 in su

Per attivare un certo environment bisogna usare il comando "conda" seguito da "activate *nome_environment*". Noi useremo Python come

linguaggio imperativo ma soprattutto come linguaggio interpretato. Inoltre dovrò poter mostrare graficamente i dati tramite grafici ecc... Potremmo usare anche Jupyter, una sorta di editor di testo misto ad un IDE che interpreta vari pezzi di codice scritti all'interno del testo. Con il comando "jupyter notebook" si apre la directory sul browser. Tramite questo strumento poi potremmo aprire le note e le dispense con pezzi di codice eseguibili sul posto.

Python
Dati e tipi di dati
Il tipo di un dato ne identifica le operazioni possibili (quello che può fare). Siamo stati abituati in precedenza da altri linguaggi a dichiarare tutte le variabili per permettere un type-checking statico in fase di compilazione.
In python il type-checking è dinamico in fase di esecuzione, quindi non abbiamo bisogno di dichiarazione di tipi.
In Java ad esempio abbiamo tipi primitivi e classi che implementano altri tipi composti, in python invece abbiamo solo i tipi "classi", cioè tutto è una classe.

Tipi di dati semplici e strutturati
Semplici:
- Informazione atomica
- Int
- Float
- Bool (True o False)

Strutturati:
- Aggregano altri tipi di dati

Gli interi hanno rappresentazione arbitraria con memoria dinamica, mentre i float hanno interpretazione classica con Nan e infinito.

Operandi
Abbiamo gli operandi classici ed alcuni con piccole differenze:
- "/" divisione reale
- "//" divisione intera
- "**" elevamento a potenza

In Jupyter per eseguire il codice all'interno di una cella dobbiamo premere Shift+Invio.

In python non abbiamo il tipo carattere, abbiamo solo il tipo stringa non interpretabile come semplice o strutturato.

Le stringhe sono delimitate da una o due virgolette. Nel caso usassimo tre apici potremmo creare stringhe che vanno a capo direttamente nel codice senza usare "\n".
Quando creiamo una funzione di solito vengono usate queste stringhe per descriverla.

Dati strutturati
Il type checking su questi tipi non ragiona sui tipi stessi ma sui metodi che implementano e le cose che possono fare (duck typing).
Classicamente ci vengono in mente i seguenti tipi:

- Array
- Lista
- Insieme
- Mappa

Array
Strutture dati con accesso posizionale, classicamente dovrebbero essere omogenei ma in python vengono implementati come liste o tuple, che non hanno il vincolo di omogeneità mantenendo l'accesso posizionale.
La differenza tra le due è che le tuple sono immutabili (a livello di puntatori) mentre le liste sono mutabili.

- Lista: riferimenti ad una serie di oggetti python;
- Tupla: riferimenti ad una serie di oggetti python, questi riferimenti non possono essere cambiati ma nel caso gli oggetti puntati possono essere mutati a livello di stato.

Le liste sono delimitate da parentesi quadre e le tuple da parentesi tonde.

a = [1, 'due', 3.14] b = ['qui', 'quo', 'qua'] t = (a, b) t	Output: ([1, 'due', 3.14], ['qui', 'quo', 'qua'])

Accessi come vettori e metodi di queste strutture con il punto. Con il comando "t[1].append('nonna papera')" il comando verrà eseguito perchè non modifico i puntatori ma cambio gli elementi all'interno dell'oggetto puntato.

Liste
Non le vedremo in questo corso

Insiemi
Aggregano elementi eterogenei, non sono ordinati in sequenza e non vi sono duplicati. Possiamo anche chiederci se vi è una certa istanza di un oggetto all'interno di essi.

Si inizializzano con le parentesi graffe.

a = {1, 'tre', 0.9} a	{0.9, 1, 'tre'}

Se possono essere ordinati in qualche modo vengono ordinati, secondo un ordine diverso rispetto all'input.
Controllo l'appartenenza tramite il comando:

0 in {1, 'tre', 0.9}	False

Posso usare questo operatore anche per le liste e le tuple, proprio come posso invocare "len" per la dimensione.

Mappe o dizionari
Si dichiarano come insiemi, ma in questo caso abbiamo un insieme di coppie di oggetti (chiave-valore).
{1:'uno', 2:'due'}
Neanche le mappe sono vincolate nell'uniformità.
Attenzione, le chiavi devono essere "hashable", quindi immutabili in modo da evitare, cambiando la variabile come chiave cambi il suo valore di hash.

Accesso
Accesso classico al dizionario, della forma x[y] dove y è la chiave nel dizionario x. Per tuple e liste y è un intero (o un numero negativo se si parte dal fondo), inoltre possiamo usare più numeri con i due punti per fare sublicing.

t = list(range(20)) t	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]

"list(. . .)" è un costruttore della classe degli oggetti lista mentre "range(20)" è un generatore di numeri da 0 a 19.

t[1:3] t[-1:-3]	[1, 2] [] #non funziona come ci aspetteremmo, dato che comunque il primo indice viene dopo il secondo

“[]“ Struttura dati vuota, si crea con forma espressiva o con costruttore vuoto.
Una tupla con un solo elemento è l’elemento stesso. Per creare lo stesso una tupla con un solo elemento la inizializziamo nella forma “(1,)“.
Per i dizionari, posso controllare se c’è un elemento associato ad una chiave con l’operatore d’accesso, se non vi è presente quella chiave viene lanciata un’eccezione.

Strutture di controllo
- If
- For
- While

Il for funziona in modo diverso dagli altri linguaggi, infatti itera su una struttura dati in modo naturale.

Esempio:
```
for personaggio in b:
    print(personaggio)
```

Non vi è presenza di graffe ma abbiamo i due punti e l’indentazione svolge un ruolo fondamentale semanticamente.
Su un dizionario solitamente itero sulle chiavi o sulle coppie chiave-valore.

Lezione del 2 Marzo 2023

Lezione 2

Riprendendo le stringhe
Cosa succede se provo ad applicare l'operatore d'accesso ad una stringa?

'cannella'[5]	'l' #non è un carattere ma una stringa

Per controllare posso fare:

type('cannella'[5])	str #classe che implementa il tipo string

'cannella'[2:5]	'nne'
'cannella'[:-1]	'cannell' #stampa tutto tranne l'ultimo carattere

Quali strutture dati possiamo utilizzare per implementare quelli che solitamente noi implementiamo come record, cioè insieme di informazioni che compongono le caratteristiche di un elemento?
Strutture dati già viste:

- Liste
- Tuple
- Mappe
- Insiemi

Cosa ci rende appetibile l'utilizzo delle tuple per la creazione di record?
Il fatto che sia ordinata, che sia di dimensione fissa:

(1500, 'verde', 'maserati')

Con gli insieme invece perdiamo la caratteristica dell'ordinamento.
Avendo per esempio:

record = [1500, 'verde', 'maserati']
record[1]
#per il colore

Così non possiamo sapere a quale campo dei record corrisponde ogni numero all'interno dell'operatore di accesso.
Invece con:

CILINDRATA = 1 record[CILINDRATA] #le costanti vengono dichiarate tutte in maiuscolo solitamente	1500

L'ultima struttura dati non ancora considerata sono i dizionari/mappe. Anch'esse non saranno ordinate ma possiamo utilizzare l'operatore d'accesso sui nomi dei campi.

record = {'cilindrata': 1500, 'colore': 'verde'} record['cilindrata']	1500

E' molto onerosa come struttura dati poiché per ogni record dobbiamo riscrivere i vari campi del dizionario.

Sappiamo che in python ogni elemento è un oggetto di una classe, quindi potremo invocare i propri metodi (non funzioni, dato che dovranno essere invocate su delle istanze degli oggetti).
L'accesso ai metodi avverrà con il ".".

l = ['qui', 'quo', qua', 'nonna papera'] l.sort() #"effetto collaterale", modifica lo stato dell'oggetto ma non mostra risultati l	['nonna papera', 'qua', 'qui', quo']

Potrei anche usare una funzione esterna che restituisce la lista ordinata senza modificare lo stato dell'oggetto.

sorted(l)	['nonna papera', 'qua', 'qui', quo']

Nel caso volessi specificare come dovrebbero essere ordinati, ad esempio al contrario, faccio:

l.sort(reverse = True) #in questo caso "reverse = True" svolge il ruolo di parametro opzionale	['quo', 'qui', 'qua', 'nonna papera']

Nel caso avessimo una funzione con, ad esempio, 50 argomenti, se fossero tutti argomenti condizionali avrei un problema nel doverli inserire tutti ad ogni chiamata e nel ricordarmi la loro posizione.
Questo si risolve utilizzando gli argomenti opzionali, che vanno dichiarati con il nome come abbiamo visto nell'esempio.
Essi possono convivere dichiarandoli nel modo esatto e definendo dei valori di default per le variabili opzionali.

In python le funzioni sono oggetti di prima classe del linguaggio, quindi esse possono passate come argomenti delle funzioni e metodi.
Quindi possiamo utilizzare delle funzioni da noi costruite per scegliere il criterio di ordinamento.
Le funzioni in python vengono definite con "def _nomefunzione_(parametri):" e nelle righe successive avremo indentazione, per far restituire i valori alla funzione utilizzo "return".

def lunghezza(stringa): return len(stringa) l.sort(reverse = True, key = lunghezza) #similare all'utilizzo dei tipi funzionali in goLang	['quo', 'qui', 'qua', 'nonna papera']

Avrei potuto utilizzare anche "len" direttamente per key, ma questo è un esempio a scopo didattico.
Anche in python abbiamo le funzioni anonime o lambda function, le quali non avranno un nome perchè verranno usate una sola volta.
Esse vengono costruite nel seguente modo:

lambda s: len(s)
#lambda prende il posto di "def", s è il parametro d'ingresso e ciò che vi è dopo i due punti è il valore restituito. Esse devono essere definite su una sola riga!
#avrei potuto anche invocarla come "l.sort(key = lambda s: len(s))"

E se avessi una funzione anonima con più di un input?

lambda x, y: x+y

La scorsa volta abbiamo visto le strutture di controllo, ad esempio il for viene costruito in maniera diversa dagli altri linguaggi, è come se fosse un for range in goLang ed itera su una struttura dati compatibile.
Esempio:

for c in 'cannella': print(c) #ricordiamo che la funzione print va a capo ogni volta	c a n n e l l a

E se volessimo stampare solo i caratteri in posizione dispari?

1) Potremmo utilizzare il sublicing con "inizio:fine:passo", quindi potremmo fare uno slicing che non considera gli elementi successivi;
2) Potremmo usare "list(enumerate('cannella'))" ottenendo una lista di tuple con gli elementi e la loro rispettiva posizione, successivamente utilizzando il for, dato che noi sappiamo che

all'interno della lista di tuple, ognuna possiede due elementi possiamo fare un for con due parametri.

for i, c in list(enumerate('cannella')): print(c) #possiamo anche togliere "list()" dato che enumerate ci fornisce un generatore che il for riesce a scandire	c a n n e l l a

Notiamo quindi che "enumerate" e "range" restituiscono degli oggetti che possono essere scanditi in un ciclo for (generatori o oggetti iterabili).

for i in range(10): print(i)	0 1 2 3 4 5 6 7 8 9

Altre strutture di controllo:

- If
- While

Struttura del While:

while(condizione):
    corpo

Struttura dell'If:

if(condizione):
    pass
elif(seconda condizione):
    pass
else:
    pass

L'istruzione "pass" è un'istruzione che non fa niente, ha funzione di segnaposto per del codice da scrivere successivamente. Viene riconosciuto dal compilatore/interprete in modo che non dia errori.

if 4 < 4: print('vero')	
if 4: print('vero')	'vero'

Posso interpretare tutte le espressioni come valori booleani, se sono equivalenti a qualcosa di nullo saranno False, True altrimenti.

if '': print('vero')	

Iniziamo adesso una sezione che non riguarda solo la programmazione, ma anche l'analisi dei dati.
Partiamo con il concetto di frequenza assoluta, cioè il numero di volte che si presenta un dato. Avendo un dataset molto ampio, non avrebbe molto senso mostrare tutti i dati, ma dovremmo eseguire un analisi dati complessiva per fornire informazioni per trarne informazioni generali.
Ad esempio, se avessimo un dataset e ci interessasse la cilindrata, potremmo descrivere questi dati per il nostro dataset in modo succinto.
Per farlo contiamo quante volte si ripete un determinato dato.
Ovviamente poi posso costruire una tabella delle frequenza, che per ogni valore distinto conserva la frequenze assolute del dataset.
Mettiamo di avere:

names = ['Acquaman', 'Ant-Man', 'Batman', 'Black Widow', 'Captain America', 'Daredevil', 'Elektra', 'Flash', 'Green Arrow', 'Human Torch', 'Hancock', 'Iron Man', 'Mystique', 'Professor X', 'Rogue', 'Superman', 'Spider-Man', 'Thor', 'Northstar'] years = [1941, 1962, None, None, 1941, 1964, None, 1940, 1941, 1961, None, 1963, None, 1963, 1981, None, None, 1962, 1979]	

Due liste che rappresentano dei supereroi e l'anno della loro prima apparizione nei loro fumetti, all'interno della lista con gli anni utilizziamo il "None" come il null di Java, infine useremo una libreria come struttura dati per le frequenze assolute.

freq = {}	{1941: 3, 1962: 2, None: 7, 1964: 1, 1940:

for y in years:
  if y in freq:
  freq[y] += 1
  else:
  freq[y] = 1
freq
#Quando incontriamo un y non ancora presente nel dizionario, avremo una "key error", quindi la struttura di controllo va aggiunta necessariamente
	1, 1961: 1, 1963: 2, 1981: 1, 1979: 1}

Posso inserire queste istruzione direttamente in una funzione per le frequenze assolute.

def get_sorted_freq(sequence):
  freq = {}
  for y in sequence:
  if y in freq:
  freq[y] += 1
  else:
  freq[y] = 1
  return sorted(list(freq.items()), key = lambda p : p[1], reverse = True)
 	
print(get_sorted_freq(years))
	[(None, 7), (1941, 3), (1962, 2), (1963, 2), (1964, 1), (1940, 1), (1961, 1), (1981, 1), (1979, 1)]

Potremmo rimuovere l'if all'interno della funzione avendo un valore di base per il dizionario. Questo possiamo ottenerlo attraverso package/moduli di python, in questo caso lo abbiamo in un modulo di base. Si inizializzano con "from ... import ...".

from collections import defaultdict
 	
def get_sorted_freq(sequence):
  freq = defaultdict(int)
  for y in sequence:
  freq[y] += 1
  return sorted(list(freq.items()), key = lambda p: p[1], reverse = True)
 	
print(get_sorted_freq(years))
	[(None, 7), (1941, 3), (1962, 2), (1963, 2), (1964, 1), (1940, 1), (1961, 1), (1981, 1), (1979, 1)]

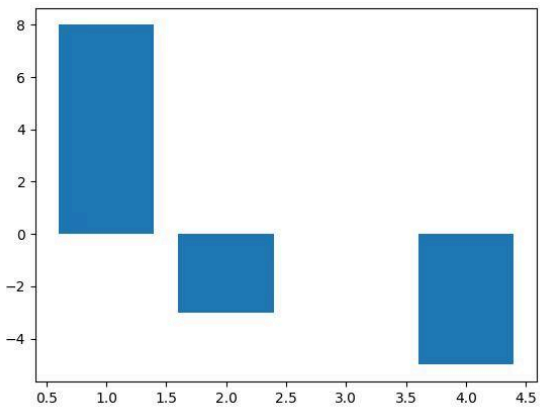
Adesso che abbiamo visto come importare le librerie, proviamo a fare un grafico. Vi sono alcuni moduli di usi generali utilizzati comunemente, ad esempio abbiamo numpy, pandas, matplotlib. Vengono importate come "import _nome_" e possiamo inserire un alias alla loro destra con "as _alias_".

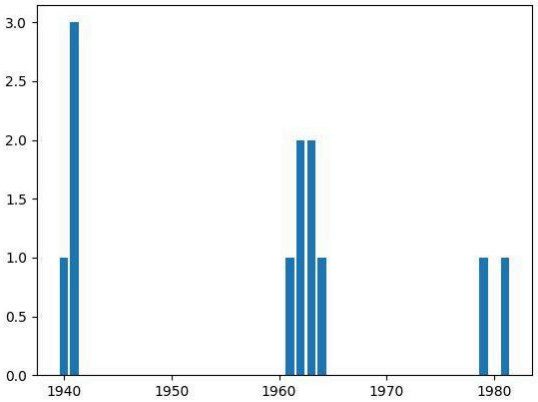
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

Numpy importa gli array su python

a = np.array([1, 6, 9]) print(a)	[1 6 9]
print(a.dtype)	int32

Pandas invece mi permette di implementare delle strutture dati per lavorare con i dataset. Infine matplotlib mi permette di realizzare grafici. Possiamo creare un grafico a barre grazie a questa libreria inserendo informazioni sulle ascisse e le ordinate.

plt.bar([1, 2, 4], [8, -3, -5]) plt.show()	
import numpy as np import pandas as pd import matplotlib.pyplot as plt from collections import defaultdict names = ['Acquaman', 'Ant-Man', 'Batman', 'Black Widow', 'Captain America', 'Daredevil', 'Elektra', 'Flash', 'Green Arrow', 'Human Torch', 'Hancock', 'Iron Man', 'Mystique', 'Professor X', 'Rogue',	[[1941 3] [1962 2] [1963 2] [1964 1] [1940 1] [1961 1] [1981 1] [1979 1]] [1941 1962 1963 1964 1940 1961 1981 1979] [3 2 2 1 1 1 1]

'Superman', 'Spider-Man', 'Thor', 'Northstar']
years = [1941, 1962, None, None, 1941, 1964, None, 1940, 1941, 1961, None, 1963, None, 1963, 1981, None, None, 1962, 1979]
 	
def get_sorted_freq(sequence):
  freq = defaultdict(int)
  for y in sequence:
  freq[y] += 1
  return np.array(sorted(list(freq.items()), key = lambda p: p[1], reverse = True))
 	
freq = get_sorted_freq(years)[1:]
print(freq)
#togliamo il primo valore null
x, y = freq.T
print(x, y)
#traspongo l'array bidimensionale e passo le dimensioni a due valori distinti
plt.bar(x, y)
plt.show()
	

Posso sostituire queste due righe con "plt.bar(*freq.T)", se avessimo avuto dei parametri opzionali avrei usato due asterischi.

Apertura di file

```
import csv
#modulo per aprire i file csv
with open('data/heroes.csv', 'r') as heroes_file:
        heroes_reader = csv.reader(heores_file, delimiter = ',', quotechar = '"')
        heroes = list(heroes_reader)[1:]
```

Il file “heroes.csv” contiene 735 righe, ognuna con le informazioni relative a un supereroe, separate da virgola nella seguente forma:

- name;identity;birth_place;publisher;height;weight;gender;first_appearance;eye_color;hair_color;strength
- A-Bomb;Richard Milhouse Jones;Scarsdale, Arizona;Marvel Comics;203;441;M;2008;Yellow;No Hair;100;Agent Bob;Bob;;Marvel Comics;178;81;M;2007;Brown;Brown;10

Il comando “with” è la parola chiave che implementa un pattern per accedere ad una risorsa per poi rilasciarla, quando termina il corpo rilascia automaticamente la risorsa (come se fosse un monitor).
Il risultato sarà una lista di liste.
Faccio lo slicing perchè probabilmente nella prima lista avremo un intestazione.

Proviamo ad estrarre da questa lista di liste una lista con gli anni di pubblicazione, convertendo i valori da stringa in intero.
Questo ci sarà possibile in modo efficiente con la "list comprehension".
In generale ho "espr(x) for x in lista if condizione".

[h[7] for h in heroes]

Successivamente dobbiamo poterla convertire in una lista di interi.

[h[7] for h in heroes if h[7] != '']

Operatore ternario

a = b < 9 ? 8 : 7
a = 8 if b < 9 else a = 7

Quindi, utilizzando l’operatore ternario, possiamo trasformare la lista in:

years = [int(h[7]) if h[7] != '' else None for h in heroes]
names = [h[0] for h in heroes]

Introduciamo pandas
L’istruzione base pd.Series(years) ci restituisce un array elementare indicizzato (in questo caso lo chiameremo “serie”).
Posso cambiare quindi anche gli indici con pd.Series(years, index = names).
Da questo posso anche fare accessi tramite i nostri indici personalizzati.

firs_appearance = pd.Series(years, index = names) first_appearance['Wonder Woman'] #dato che abbiamo valori “None” in years verranno tutti convertiti in float	1941.0

Possiamo anche fare slicing e sublicing tramite .iloc con indici interi e .loc con i nostri indici personalizzati, facendo attenzione a conoscere quali indici vengano prima o dopo degli altri nel secondo caso.
Nel caso del sublicing con i nostri indici personalizzati, l'ultimo indice verrà compreso.
Oltre al sublicing posso utilizzare .iloc[[x, y, z]] per ottenere gli elementi nelle posizioni x, y, z, oppure inserendo una lista della dimensione dell'array con valori booleani per scegliere quali mostrare.

Lezione del 7 Marzo 2023

Lezione 3

List comprehension

first_appearance[[1970 <= y < 1975 for y in first_appearance]]

Viene valutata l'espressione a destra una volta valutata deve restituire qualcosa su cui è possibile iterare, in questo caso una serie di numpy (array indicizzato).
Il nome y conterrà di volta in volta gli elementi su cui itero.
Per ogni elemento y che poi vado a valutare, controllo l'espressione all'inizio della lista comprehension, il risultato di ogni analisi dell'espressione viene poi messo in una lista.

Negli array di numpy si possono fare, oltre alle operazioni classiche, operazioni elemento per elemento, ad esempio avendo due array a e b, facendo la somma a+b ottengo un array con le somme di ogni coppia di elementi.

Come abbiamo le list comprehension, abbiamo anche le set comprehension e le dict comprehension.
Per la set comprehension la sintassi è sostanzialmente la stessa, cambiano solo le parentesi interne diventano graffe.

first_appearance[{1970 <= y < 1975 for y in first_appearance}]

Non abbiamo ambiguità perchè sappiamo come vengono distinti dizionari ed insiemi.
Infatti per le dict comprehension la sintassi è un minimo più complicata con all'inizio una coppia di chiave/valori.

first_appearance è una serie di numpy con il primo anno di apparizione e come indici i nomi dei supereroi corrispondenti, le prime parentesi quadre rappresentano l'operatore d'accesso.

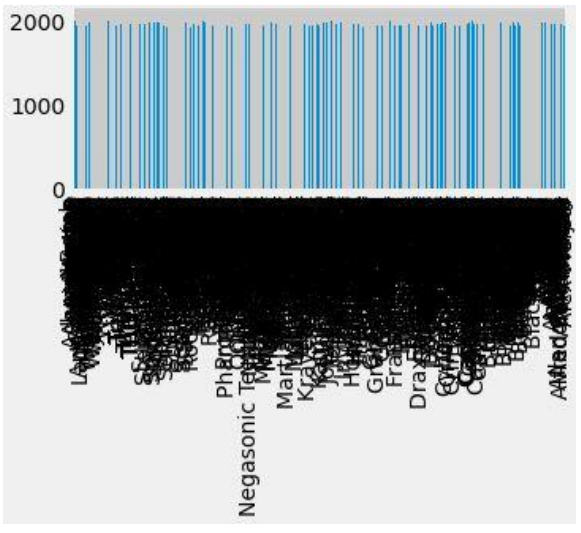
All'interno dell'operatore di accesso creo una lista di booleani attraverso la list comprehension.
Usando quindi la lista di booleani creata come operatore di accesso, ottengo soltanto gli elementi identificati come True.
E' possibile ottenere un risultato simile con una sintassi più pulita:

first_appearance[first_appearance > 2010]

In questo caso, nell'espressione ho elementi di dimensione diversa, attraverso il broadcasting in python viene aumentata la dimensione dello scalare come se fosse un array con lo stesso elemento ripetuto, ottenendo quindi all'interno dell'operatore di accesso come risultato un array con valori booleani al suo interno, per poi usarli per ottenere solo i supereroi apparsi dopo il 2010.

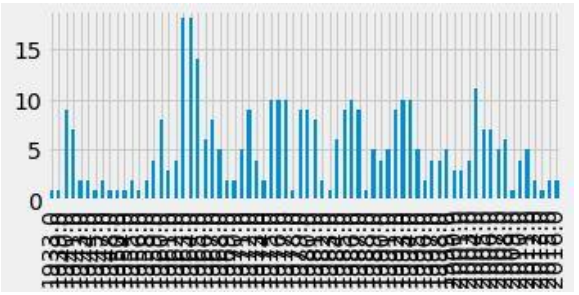
Rivediamo alcune cose dall'ultima lezione.
La scorsa volta abbiamo creato una funzione per calcolare le frequenze assolute.
Osservazione, possiamo usare il metodo .head() per mostrare i primi 5 elementi della serie per avere un'idea dei valori e degli indici al proprio interno.
Usando first_appearance.value_count() ottengo una serie con i valori come gli indici e la rispettiva frequenza assoluta come valori.
La serie inoltre sarà ordinata in ordine decrescente.

Abbiamo inoltre visto come rappresentare graficamente questi risultati:

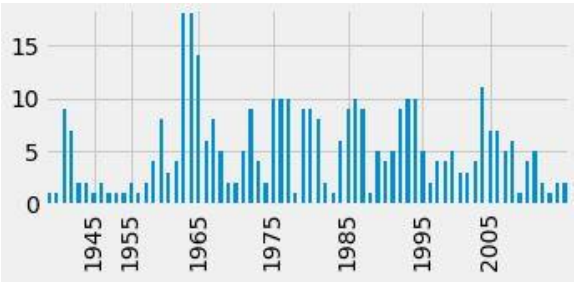
first_appearance.plot.bar() plt.show()	

Osservazione, .plot non è l'invocazione di un metodo perchè non abbiamo le parentesi tonde, quindi sarà l'invocazione di un oggetto creato a partire da quello che si trova prima.
Il risultato sarà un grafico con sotto i nomi (gli indici) della serie e come altezza delle barre l'anno. Questo grafico oltre ad essere graficamente scorretto non ci fornisce nessuna informazione aggiuntiva utile.

Potrei mettere insieme le ultime due cose viste, cioè frequenza assoluta degli anni e rappresentazione grafica.

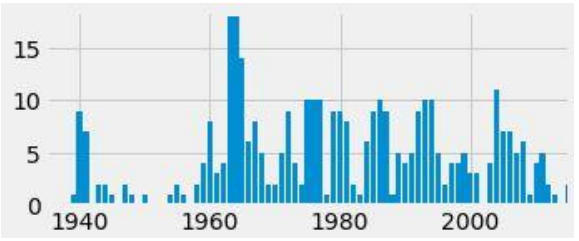
first_appearance.value_counts().plot.bar() plt.show()	

In questo caso l'informazione ottenuta è più significativa, anche se graficamente rimane pesante. Almeno le etichette in basso non sono più i nomi dei supereroi ma gli anni, quindi potrei omettere alcune date.

years = np.arange(1945, 2010, 10)
#array di numpy con arange corrispondente a range, ma crea un array direttamente
index_pos = [fist_app_freq.index.get_loc(y) for y in years]
#first_app_freq è una lista con all'interno i value counts (tabella delle frequenze) ordinata non secondo la frequenza ma secondo l'indice (l'anno),
con .index ottengo una lista con gli indici, con .get_loc(y) ottengo la posizione di quell'indice
first_app_freq.plot.bar()
plt.xticks(index_pos, year)
#modifico gli indici nell'asse x, come primo elemento scelgo quali indici da mostrare e come secondo elemento cosa mostrare come indice
plt.ylim((0, 18.5))
	

#modifico i limiti di altezza del grafico a livello visivo
plt.show()	

Il grafico ottenuto è ancora non perfetto, infatti le etichette sull'asse x non sono distanziate in modo uniforme. Probabilmente questo è dovuto al fatto che non abbiamo alcuni dati negli indici non presenti.
Per sistemare il grafico dovremmo abbandonare pandas utilizzando solo plt.

plt.bar(first_app_freq.index, first_app_freq.values) plt.xlim((1935, 2015)) plt.ylim(0, 18.5) plt.show()	

Operazioni le serie
Consideriamo le seguenti domande:

1) Quanti supereroi sono apparsi a partire dal 1960?
2) Quanti tra il 1940 e il 1965?
3) Quanti prima del 1970?

sum(first_app_freq[1960:])	329
sum(first_app_freq[1940:1966])	106
sum(first_app_freq[:1970])	130

In pandas la somma tra due serie non avviene come quella di numpy (elemento per elemento), ma in questo caso avviene per elementi con lo stesso indice.

height = pd.Series([float(h[4]) if h[4] else None for h in heroes], index = Names) weight = pd.Series([float(h[5]) if h[5] else None for h in heroes], index = Names)	#abbiamo creato quindi due serie con altezze e pesi con i supereroi
(height/100)[:10]	#visualizzo i primi 10 elementi con l'altezza divisa per 100, nessun problema, l'operazione viene fatta elemento per elemento

height.apply(lambda h: (h/100)**2)[:10]	#nel caso volessi usare un operatore non presente in python posso creare una funzione anonima con .apply() #prendo l'altezza in centimetri, la porto in metri e la elevo al quadrato
bmi = weight / height.apply(lambda h: (h/100)**2) bmi.sort_values(ascending = False)[:10]	#in questo caso posso calcolare il bmi dei supereroi facendo operazioni sullo stesso indice, ordinandoli poi in modo decrescente
standard_weight = weight[(weight < 100) & (weight > 40)] standard_height = height[(height < 200) & (height >120)]/100	#in python l'operatore booleano si inizializza con "and", con "&" ho un operatore logico per i vettori
(standard_weight / (standard_weight/100)**2)[:15]	#non sappiamo in questo caso se le due serie create abbiano la stessa dimensione ed inoltre gli stessi indici (un supereroe potrebbe avere altezza standard ma non il peso) #non abbiamo errori, soltanto Nan nel caso un supereroe compaia in solo una serie

Dataframe
E una collezione di singoli individui (di serie) che hanno lo stesso tipo di indice. La struttura dati corrispondente è pd.Dataframe.
Per crearli possiamo direttamente leggere file csv.

heroes = pd.read_csv('data/heroes.csv', sep =';', index_col = 0)

Name	Identity	Birth place	Publisher	Height	Weight	Gender	First appearance	Eye color	Hair color	Strength	Intelligence
A-Bomb	Richard Milhouse Jones	Scarsdale, Arizona	Marvel Comics	203.21	441.95	M	2008.0	Yellow	No Hair	100.0	moderate
Abraxas	Abraxas	Within Eternity	Marvel Comics	NaN	NaN	M	NaN	Blue	Black	100.0	high
Abomination	Emil Blonsky	Zagreb, Yugoslavia	Marvel Comics	203.04	441.98	M	NaN	Green	No Hair	80.0	good
Adam Monroe	NaN	NaN	NBC - Heroes	NaN	NaN	M	NaN	Blue	Blond	10.0	good
Agent 13	Sharon Carter	NaN	Marvel Comics	173.41	61.03	F	NaN	Blue	Blond	NaN	NaN
Air-Walker	Gabriel Lan	Xandar, a planet in the Tranta system, Androme...	Marvel Comics	188.59	108.23	M	NaN	Blue	White	85.0	average
Agent Bob	Bob	NaN	Marvel Comics	178.25	81.45	M	2007.0	Brown	Brown	10.0	low
Abe Sapien	Abraham Sapien	NaN	Dark Horse Comics	191.24	65.35	M	1993.0	Blue	No Hair	30.0	high
Abin Sur	NaN	Ungara	DC Comics	185.52	90.90	M	1959.0	Blue	No Hair	90.0	average
Angela	NaN	NaN	Image Comics	NaN	NaN	F	NaN	NaN	NaN	100.0	high
Animal Man	Bernhard Baker	NaN	DC Comics	183.80	83.39	M	1965.0	Blue	Blond	50.0	average
Agent Zero	Christoph Nord	Unrevealed location in former East Germany	Marvel Comics	191.29	104.17	M	NaN	NaN	NaN	30.0	good
Colin Wagner	NaN	NaN	HarperCollins	NaN	NaN	M	NaN	Grey	Brown	NaN	NaN
Angel Dust	Christina	NaN	Marvel Comics	165.78	57.21	F	NaN	Yellow	Black	55.0	moderate
Angel Salvadore	Angel Salvadore Bohusk	NaN	Marvel Comics	163.57	54.67	F	2001.0	Brown	Black	10.0	moderate
Zoom	Hunter Zolomon	NaN	DC Comics	185.90	81.93	M	NaN	Red	Brown	10.0	average
Lady Deathstrike	Yuriko Oyama	Osaka, Japan	Marvel Comics	175.85	58.89	F	1985.0	Brown	Black	30.0	good
Yoda	Yoda	NaN	George Lucas	66.29	17.01	M	1980.0	Brown	White	55.0	high
Zatanna	Zatanna Zatara	NaN	DC Comics	170.29	57.77	F	NaN	Blue	Black	10.0	high
Yellowjacket II	Rita DeMara	NaN	Marvel Comics	165.58	52.36	F	NaN	Blue	Strawberry Blond	10.0	average

. . .

Come accediamo ad un dataframe?
Con l'operatore di accesso classico accedo alle colonne, tramite parentesi quadrate con all'interno una stringa, ottenendo come risultato una serie.
Il tipo di dato all'interno delle serie sarà oggetto, per permettere che vi siano dei dati mancanti.
Possiamo accedere alle colonne attraverso un'altra sintassi.

heroes['Gender'] heroes.Gender #non funziona se il nome della colonna possiede degli spazi	Name A-Bomb M Abraxas M Abomination M Adam Monroe M Agent 13 F .. Alan Scott M Amazo M Ant-Man M Ajax M Alex Mercer M Name: Gender, Length: 735, dtype: object

Se utilizzo l'operatore di accesso con slicing accedo alle righe.

heroes['Agent 13':'Air-Walker']

Name	Identity	Birth place	Publisher	Height	Weight	Gender	First appearance	Eye color	Hair color	Strength	Intelligence
Agent 13	Sharon Carter	NaN	Marvel Comics	173.41	61.03	F	NaN	Blue	Blond	NaN	NaN
Air-Walker	Gabriel Lan	Xandar, a planet in the Tranta system, Androme...	Marvel Comics	188.59	108.23	M	NaN	Blue	White	85.0	average

In questo caso però vengono restituite solo 2 righe.
Per fare sublicing possiamo usare .loc e .iloc, potendo filtrare sia righe che colonne insieme.

heroes.loc['Professor X', 'Height':'Weight']	Height 183.74 Weight 86.89 Name: Professor X, dtype: object

Se voglio accedere ad un elemento singolo uso .at e .iat.
Quando ordino un dataframe devo scegliere per quale colonna ordinare:

heroes.sort_values(by = 'Weight')	#ottengo un nuovo dataframe ordinato

Posso eseguire delle query direttamente su questi dataframe specificando le colonne.

heroes_with_year = heroes[heroes['First appearance'] > 1900]
heroes_with_year.head()

Name	Identity	Birth place	Publisher	Height	Weight	Gender	First appearance	Eye color	Hair color	Strength	Intelligence
A-Bomb	Richard Milhouse Jones	Scarsdale, Arizona	Marvel Comics	203.21	441.95	M	2008.0	Yellow	No Hair	100.0	moderate
Agent Bob	Bob	NaN	Marvel Comics	178.25	81.45	M	2007.0	Brown	Brown	10.0	low
Abe Sapien	Abraham Sapien	NaN	Dark Horse Comics	191.24	65.35	M	1993.0	Blue	No Hair	30.0	high
Abin Sur	NaN	Ungara	DC Comics	185.52	90.90	M	1959.0	Blue	No Hair	90.0	average
Animal Man	Bernhard Baker	NaN	DC Comics	183.80	83.39	M	1965.0	Blue	Blond	50.0	average

heroes_with_year = heroes[(heroes['First appearance'] > 1900) & (heroes['Eye color'] == 'Blue')]	#Ottengo sempre un dataframe, ma sono filtrati per anno maggiore di 1900 e con gli occhi azzurri.

Le parentesi vengono specificate per non lasciare che venga valutata l'espressione tramite l'ordine del linguaggio.

Lezione del 9 Marzo 2023

Lezione 4

Ripetiamo che lo scopo di questo insegnamento è fornirci degli strumenti formali per gestire l'incertezza ed infine trarre delle conclusioni dai nostri dati.

Incertezza = esperienza che, se ripetuta più volte, non ha sempre lo stesso risultato.

Rischio = valutazione delle conseguenze nel caso una nostra previsione sia falsa o non come aspettata.

Ci sono 3 argomenti principali teorici che andremo ad affrontare:

1) Statistica descrittiva
2) Calcolo delle probabilità
3) Statistica inferenziale

Statistica descrittiva

Si basa sull'esplorazione di dati conosciuti. Immaginiamo di avere un dataset, considerando un suo attributo ho bisogno di strumenti per condensare grandi quantità di informazioni in dati descrittivi generali. Non solamente dati ma anche semplicemente dei grafici.
Quello che compio inizialmente è l'esplorazione dei dati, dalla quale ricavo informazioni che possono poi andare ad affiancare con altri nostri risultati ottenuti.

Calcolo delle probabilità

E' un modello matematico che ci permette di descrivere i casi di incertezza attraverso l'analisi dei casi possibili e della loro frequenza. Le risposte alle nostre domande nel calcolo delle probabilità saranno sotto forma di frequenza (quante volte un evento si verifica rispetto a tutti i casi possibili).

Statistica inferenziale

Una volta modellata la realtà secondo il calcolo delle probabilità, posso prendere dei dati già ottenuti tramite la statistica descrittiva per modellare o approssimare dati futuri o ancora sconosciuti.

Osservazioni

Un osservazione x_i è l'analisi di un fenomeno che può variare. L'insieme delle osservazioni è detto "campione" ed è rappresentato nel seguente modo:
$\{x_1, x_2, \dots, x_n\}$ dove n prende il nome di dimensione/taglia del campione.

Popolazione
Le osservazioni fatte per generare un campione sono eseguite su un insieme più grande chiamato proprio "popolazione".
L'analisi dei dati non viene solitamente fatta su tutta la popolazione ma su un sottoinsieme che viene detto "campione rappresentativo".

Tipi di frequenza:
- Frequenza assoluta
  - Quante volta un valore osservabile si presenta nel nostro campione
- Frequenza relativa
  - Frequenza assoluta diviso la dimensione del campione

Sono quasi equamente espressive ma la frequenza relativa ci permette di calcolare le percentuali di presentazione di tale fenomeno in modo più facile, inoltre dipende meno dalla dimensione del campione.

Media campionaria:
Ci fornisce informazioni su quale dato si aggirino globalmente i dati nel campione.
Viene detta anche valutazione della centralità dei dati.
Viene indicata nel seguente modo:
$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
Media aritmetica o nel contesto della statistica proprio media campionaria (dato che viene eseguita su un campione).
Nella classe "series" ci sono vari metodi per il calcolo statistico, anche quello per il calcolo della media campionaria.
Cosa succederebbe alla media se decidessi di aggiungere una costante ad ogni elemento del campione?
$\forall i = 1, \dots, n \qquad y_i = x_i + b \text{ con } b \in \Re$

$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{i=1}^{n} x_i + b = \frac{1}{n}\sum_{i=1}^{n} x_i + \frac{1}{n}\sum_{i=1}^{n} b$
La prima parte è uguale a $\bar{x}$ mentre la seconda è uguale a b, quindi il nostro risultato sarà:
$= \bar{x} + b$

Questa viene detta traslazione della media, utile se abbiamo dei valori elevati ma con minima variazione tra loro.
Analogamente, se avessi moltiplicato i nostri elementi del campione per una costante avrei:
$y_i = ax_i$ con $a \in \Re$

$\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{i=1}^{n} ax_i = \frac{1}{n}\sum_{i=1}^{n} x_i \cdot \frac{1}{n}\sum_{i=1}^{n} a$

La prima parte è uguale a $\overline{x}$ mentre la seconda è uguale a a, quindi il nostro risultato sarà:
$= a\overline{x}$
Questa operazione invece viene detta scalatura della media, utile per trovare la media in una determinata unità di misura per poi convertirla facilmente.

Dopo queste considerazioni possiamo dunque concludere che avendo $y_i = ax_i + b$ la nostra media sarà $\overline{y} = a\overline{x} + b$.
Trasformazione lineare, la media non cambia quindi l'operatore è lineare.

Per esempio, immaginiamo di avere un campione che misura quante settimane sono passate da quanto una persona ha iniziato la scuola guida a quando ha preso effettivamente la patente.
$\{2,\ 110,\ 5,\ 7,\ 6,\ 7,\ 3\}$ l'insieme dei nostri x_i

$\overline{x} = 20$
Quando ho un valore fuori scala, detto nel linguaggio della statistica un "outlier", ho un valore insensato che modifica e deforma la media campionaria. Quindi la media campionaria si dice non robusta rispetto agli outlier.

Un altro indice per la centralità di un campione è la mediana campionaria. Si ottiene prendendo i valori, disponendoli in ordine ed infine si prende il valore in mezzo al campione.
Essa, per definizione, sarà maggiore o uguale di almeno la metà dei dati e minore o uguale di almeno la metà dei dati.
Nel caso in cui il campione sia di dimensione pari, la mediana campionaria viene definita come la media fra i due dati in mezzo.
La mediana campionaria è robusta rispetto agli outlier.

Abbiamo anche un terzo indice di centralità, la moda campionaria, essa è l'osservazione con la maggiore frequenza (assoluta o relativa).
Non sempre abbiamo un valore modale unico.

Limiti di questi indici di centralità

- La mediana è calcolabile solo se le osservazioni sono ordinabili globalmente
- La media campionaria richiede valori numerici per la somma e la divisione

Tipi di attributi in statistica
- Scalari, numerici o quantitativi
    - Ereditano le caratteristiche dei numeri
    - Si possono fare media, moda e mediana
- Categorici o qualitativi
    - Ordinali e non
    - Ordinati nel caso tramite la scala Likert categorica (ad esempio: Molto d'accordo, d'accordo, indeciso, in disaccordo e fortemente in disaccordo)
    - Possono essere anche rappresentati con i numeri senza ereditare le loro caratteristiche
    - Si possono dare moda o mediana per quelli ordinali, mentre per quelli non ordinali si può fare solo la moda

Comparazione tra moda e mediana campionaria
La mediana campionaria equivale, a volte, ad un valore nel campione.
La moda campionaria invece è sempre un valore del campione.

Immaginiamo adesso di avere un campione non descritto in modo estensivo ma tramite tabella delle frequenze.
Ad esempio, numero di abiti venduti in base alla loro taglia.

Valore	Frequenza assoluta
3	2
4	1
5	3

La somma delle frequenze assolute ci fornisce la taglia del campione $= 6$.
Possiamo anche ottenere il campione iniziale moltiplicando i dati per le loro frequenze.
Campione iniziale:
$\{3,\ 3,\ 4,\ 5,\ 5,\ 5\}$
Possiamo calcolare anche la media tramite Dati·Frequenze al numeratore e la taglia del campione ottenuta precedentemente al denominatore.

Mettiamo di avere dei valori osservabili

$v_1, v_2, \dots, v_n$

E un campione

$x_1, x_2, \dots, x_n$

All'interno del campione ogni valore osservabile avrà una propria frequenza assoluta

$v_1 \quad f_1$

$v_2 \quad f_2$

$\dots \quad \dots$

$v_k \quad f_k$

A questo punto il nostro campione sarà formato ad esempio da x_1, x_2 e x_3 che magari non saranno altro che il valore v_1 ripetuto per la sua frequenza assoluta f_1 (in questo caso 3).

Quindi la nostra media diventerà:

$\frac{x_1+x_2+\dots+x_n}{n} = \frac{1}{n}\sum_{j=1}^{k} f_j v_j = \sum_{j=1}^{k} \frac{f_j}{n} v_j$

$\frac{f_j}{n}$ non è altro che la frequenza relativa quindi

$= \sum_{j=1}^{k} f_j' v_j$

Supponiamo adesso di avere due campioni

$a = \{1,\ 2,\ 5,\ 6,\ 6\} \qquad \bar{a} = 4$

$b = \{-40,\ 0,\ 5,\ 20,\ 35\}\ \bar{b} = 4$

Cosa potrebbe esprimere la differenza tra questi due differenti campioni?

Considero quindi la dispersione dei dati, fissato un valore centrale di riferimento controllo la distanza dei dati da esso.

Per valutare la dispersione ho vari indici, ad esempio la somma degli scarti tra i valori e la media.

$\sum_{i=1}^{n} (x_i - \bar{x})$

Però in questo modo ottengo sempre zero.

$\sum_{i=1}^{n} (x_i - \bar{x}) = \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \bar{x} = \sum_{i=1}^{n} x_i - n\bar{x}$

Dato che

$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ allora $\sum_{i=1}^{n} x_i = n\bar{x}$

Quindi avrò

$n\bar{x} - n\bar{x} = 0$

Potrei considerare il valore assoluto dello scarto dei valori per poi considerare la sua media.

$\frac{1}{n}\sum_{i=1}^{n} |x_i - \bar{x}|$

Però il valore assoluto risulta essere molto fastidioso nelle semplificazioni.

Posso utilizzare la media degli scarti quadratici.

$\frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x})^2$

Questa viene detta varianza campionaria e viene indicata con s^2.
Proviamo a trasformare anche la varianza scalandola.

$y_i = ax_i$

$s_y^2 = \frac{1}{n}\sum_{i=1}^{n} (ax_i - a\bar{x})^2 = a^2\frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x})^2 = a^2 s_x^2$

E se provassi a traslare la varianza?

$y_i = x_i + b$

$s_y^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i + b - (\bar{x} + b))^2 = s_x^2$

Quindi linearmente ho:

$y_i = ax_i + b$

$s_y^2 = a^2 s_x^2$

Se dobbiamo calcolare la varianza su dei valori che rappresentano delle unità di misura potremmo avere dei problemi trovando come risultato unità del tipo l^2, Kg^2, ...
Per questo viene utilizzata la derivazione campionaria standard.

$s = \sqrt{s^2}$

Con $y_i = ax_i$ avremo $s_y = |as_x|$.

Questo non è l'unico indice per misurare la dispersione.
In questo secondo caso verrà usata la mediana con la sua definizione.
Immaginiamo di poter scegliere dove far ricadere la mediana.

Percentile campionario

$p = \{1, 2, \dots, 100\}$

$\{x_1, \dots, x_n\}$ campione

Il percentile di livello p è quella osservazione x_i maggiore o uguale di almeno i $\frac{p}{100}n$ delle osservazioni e minore o uguale di almeno i $(1 - \frac{p}{100})n$ delle osservazioni.
Mediana = percentile di livello 50

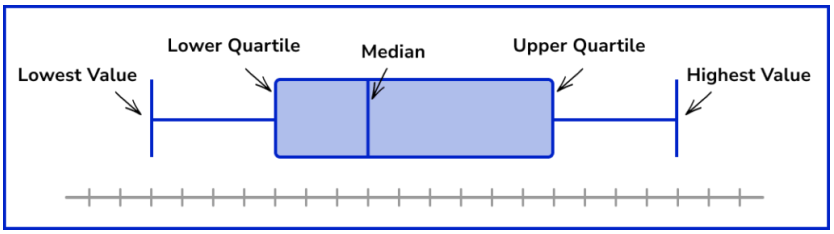
Esempio
Diciamo di avere dodici osservazioni nel nostro campione e di doverne calcolare il novantesimo percentile.
Quindi devo trovare la x_i che sia ≥ di 10.8 osservazioni e ≤ di 1.2 osservazioni.
x_{11} è maggiore o uguale delle 11 osservazioni precedenti (essa compresa).
x_{11} è minore o uguale delle 2 osservazioni successive (essa compresa).

Nel caso $\frac{p}{100}n$ venga un numero intero ho due osservazioni appetibili, quindi scelgo la media fra di loro come per la mediana.

Se cambiassi la scala dell'indice potrei ottenere un decile $\{1, 2, \dots, 10\}$ oppure un modo più lasco dividendolo in 4 gruppi avrei un quartile.
Quando la scala ha un intervallo reale (di solito la scala reale è tra 0 e 1) abbiamo dei quantili.

Usando i quartili ottengo una rappresentazione grafica di una correlazione tra centralità e mediana.

Box Plot
Detto anche diagramma a scatola e baffi, rappresenta graficamente la distribuzione del campione.



La scatola è disegnata a partire dal primo quartile al terzo quartile, disegnando una linea in corrispondenza della mediana.
Successivamente vengono disegnate due linee che terminano in corrispondenza del valore minimo e del massimo.
Utilizzo la mediana per la centralità e ottengo due range di valori:

- Range (tra minimo e massimo)
- Range interquartile (tra primo e terzo quartile)

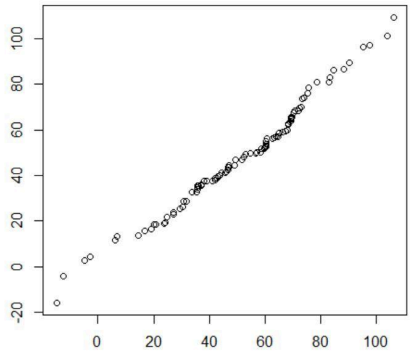
Possiamo trovare degli outlier nei Box Plot se un dato dista una costante (di solito 1.5) per la dimensione del range interquartile da uno dei due quartili.
Una volta identificati vengono marcati con dei punti o asterischi e non vengono considerati.

Definiamo un altro indice di dispersione, chiamato coefficiente di variazione.

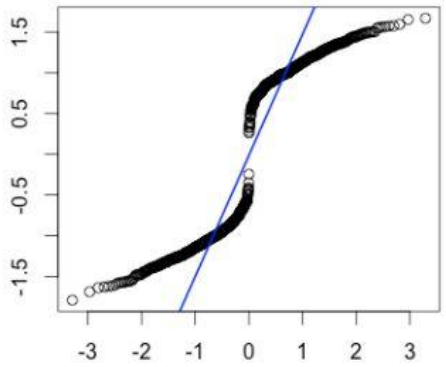
$s\ ^{*}= \frac{s}{|\bar{x}|}$

Anche questo indice misura la variazione di due campioni diversi.

Ci si presenteranno sicuramente dei casi in cui vorrò andare a controllare se 2 set di informazioni provengono dagli stessi dati.
Per questo utilizziamo il diagramma quantile quantile, diagramma QQ o QQ-Plot.
Inseriamo sugli assi cartesiani i quantili dei due gruppi di informazioni da analizzare.



Se le popolazioni hanno ottenuto un risultato molto simile avremo dei dati che si aggirano intorno alla bisettrice di quel piano, nel caso contrario la maggioranza dei dati sarà molto lontano da essa.
Nel caso siano solo pochi punti ad essere distanti, a quel punto avremo identificato gli outlier.



Lezione del 14 Marzo 2023

Lezione 5

Informazione di servizio:
Per il ricevimento del giovedì mattina alle 10:30 non c'è bisogno della prenotazione, basta andare in ufficio e, nel caso, aspettare il proprio turno.

Durante l'ultima lezione abbiamo definito la varianza campionaria, in realtà non viene definita propriamente in quel modo, fuori dalla sommatoria i dati vengono in realtà divisi per $n-1$ non per n.
La spiegazione del motivo di $n-1$ avverrà verso la fine del corso.
Possiamo solamente accennare che il fatto che venga usato $n-1$ ci fornisce una proprietà che useremo in statistica inferenziale.

In questa lezione metteremo insieme gli argomenti visti nelle ultime due lezioni.
Iniziamo considerando il seguente dataset:

import csv import numpy as np import pandas as pd import matplotlib.pyplot as plt heroes = pd.read_csv('heroes.csv', sep =';', index_col = 0) heroes_with_year = heroes[heroes['First appearance'] < 2020] print(heroes_with_year.head())	#Broadcasting tra colonna del dataframe per ottenere tramite broadcasting una lista di booleani per accedere ad il dataframe solo alle righe con valore True

Name	Identity	Birth place	Publisher	Height	Weight	Gender	First appearance	Eye color	Hair color	Strength	Intelligence
A-Bomb	Richard Milhouse Jones	Scarsdale, Arizona	Marvel Comics	203.21	441.95	M	2008.0	Yellow	No Hair	100.0	moderate
Agent Bob	Bob	NaN	Marvel Comics	178.25	81.45	M	2007.0	Brown	Brown	10.0	low
Abe Sapien	Abraham Sapien	NaN	Dark Horse Comics	191.24	65.35	M	1993.0	Blue	No Hair	30.0	high
Abin Sur	NaN	Ungara	DC Comics	185.52	90.90	M	1959.0	Blue	No Hair	90.0	average
Animal Man	Bernhard Baker	NaN	DC Comics	183.80	83.39	M	1965.0	Blue	Blond	50.0	average

Nell'ultima lezione abbiamo visto i dati quantitativi e qualitativi.
Una delle principali distinzioni che si possono fare è sul come vengono misurati:

- Si parla di dati quantitativi se l'esito della misurazione è una quantità numerica
- Si parla invece di dati qualitativi (o categorici, o nominali) quando la misurazione è fatta scegliendo un'etichetta a partire da un insieme disponibile

Classificazione dei dati qualitativi
I dati qualitativi vengono spesso ulteriormente classificati come binari/booleani, nominali oppure ordinali. Si parla di dati binari o booleani quando l'osservazione può avere solo due esiti tra loro non confrontabili (volendo si può parlare di dati booleani per enfatizzare che si sta valutando la presenza o l'assenza di una proprietà, e di dati binari quando esistono due possibili etichette): in tal senso, il carattere *Gender*, che può assumere solo i valori *M* e *F*, è quindi un carattere qualitativo binario.
Anche nei dati nominali (detti anche sconnessi), di cui i dati binari rappresentano un caso particolare, i valori osservabili non sono tra loro confrontabili, sebbene non vi sia limite sul numero di diverse etichette. Saranno dunque dati qualitativi nominali, oltre al già considerato *Gender*, anche *Name*, *Identity*, *Birth place*, *Publisher*, *Gender*, *Eye color* e *Hair color*.
Detto in altri termini, in questo tipo di dati (e quindi anche nel caso binario/booleano) è solo possibile stabilire una relazione di equivalenza tra i valori osservabili: pertanto, due osservazioni potranno avere valori uguali oppure diversi, e nulla più si potrà dire sul loro rapporto.
Nei dati ordinali, invece, è possibile stabilire una relazione d'ordine tra i valori osservabili, e quindi quando due valori saranno diversi sarà anche possibile dire quale tra i due sia il più piccolo e quale il più grande. Nel nostro dataset, solo Intelligence è un dato qualitativo ordinale.

Classificazione dei dati quantitativi
Per quanto riguarda i dati quantitativi, viene spesso fatto riferimento alla differenza tra dati discreti e continui in funzione del tipo di insieme di valori che questi possono assumere.
Va in realtà notato che i dati che elaboriamo sono memorizzati su un computer e quindi i valori reali vengono approssimati tramite valori all'interno di un insieme finito (dunque discreto).
Vale più la pena ragionare in termini di caratteri per cui ha senso dare significato a un singolo valore (come nel caso dell'anno di prima apparizione, in cui ha senso considerare gli eroi apparsi nel 1970) e di caratteri in cui di norma ha senso considerare un intervallo di valori (come nel caso dei rimanenti caratteri: ha di solito poco senso considerare, per esempio, un eroe alto esattamente 178 centimetri o con un indice di forza pari a 42).

Frequenze assolute e relative nella loro visualizzazione
Abbiamo visto che possiamo calcolare la media campionaria come media tra le osservazioni o media pesata delle frequenze assolute.
Fino ad adesso abbiamo calcolato le frequenze assolute nelle serie con il metodo .value_counts() ottenendo una tabella delle frequenze ordinata per la frequenza.

print(heroes_with_year['Publisher'].value_counts())

```
Marvel Comics        205
DC Comics            121
Dark Horse Comics     12
George Lucas          11
ABC Studios            4
Image Comics           3
Star Trek              1
Universal Studios      1
Hanna-Barbera          1
Rebellion              1
Name: Publisher, dtype: int64
```

Esiste in realtà un altro metodo per calcolarle.

publisher_freq = pd.crosstab(index = heroes_with_year['Publisher'], columns = ['Abs. frequence'], colnames = [''])
print(publisher_freq)

	Abs. frequence
Publisher	
ABC Studios	4
DC Comics	121
Dark Horse Comics	12
George Lucas	11
Hanna-Barbera	1
Image Comics	3
Marvel Comics	205
Rebellion	1
Star Trek	1
Universal Studios	1

In questo caso notiamo che la colonna degli indici ha un nome, viene restituito un dataframe il quale viene ordinato secondo gli indici.
Se volessimo invece visualizzare le frequenze relative?

publisher_rel_freq = publisher_freq/publisher_freq.sum()
print(publisher_rel_freq)

Publisher	Rel. frequence
ABC Studios	0.011111
DC Comics	0.336111
Dark Horse Comics	0.033333
George Lucas	0.030556
Hanna-Barbera	0.002778
Image Comics	0.008333
Marvel Comics	0.569444
Rebellion	0.002778
Star Trek	0.002778
Universal Studios	0.002778

Con il metodo .sum() applicato ad un dataframe ottengo una serie con la somma degli elementi per ogni colonna, in questo caso ho una sola colonna.
In realtà sarebbe stato possibile creare questo dataframe con le frequenze relative aggiungendo un parametro opzionale durante la dichiarazione.

publisher_rel_freq = pd.crosstab(index = heroes_with_year['Publisher'], columns = ['Abs. frequence'], colnames = [''], normalize = True)
print(publisher_rel_freq)

Publisher	Rel. frequence
ABC Studios	0.011111
DC Comics	0.336111
Dark Horse Comics	0.033333
George Lucas	0.030556
Hanna-Barbera	0.002778
Image Comics	0.008333
Marvel Comics	0.569444
Rebellion	0.002778
Star Trek	0.002778
Universal Studios	0.002778

Per rimuovere dalla visualizzazione i numeri dopo la virgola ho molteplici possibilità.

*print(publisher_rel_freq.apply(lambda p: 100 * np.round(p, 3)))*

Publisher	Rel. frequence
ABC Studios	1.1
DC Comics	33.6
Dark Horse Comics	3.3
George Lucas	3.1
Hanna-Barbera	0.3
Image Comics	0.8
Marvel Comics	56.9
Rebellion	0.3
Star Trek	0.3
Universal Studios	0.3

Ricordiamo che si possono applicare le funzioni ad un metodo dato che in python anch'esse sono oggetti. La funzione in questo caso viene applicata a tutti i valori del dataframe.
Passo una funzione anonima, essa evoca prima il metodo .round() che prende dei valori con la virgola e li arrotonda con determinati numeri dopo la virgola (in questo caso 3) e subito dopo viene moltiplicato per 100 per ottenere il valore percentuale.
Abbiamo altri metodi per arrotondare, solo che numpy per questi metodi (anche ad esempio il seno/coseno oppure il logaritmo) sono vettorizzati, quindi possiamo farne il broadcasting sui vettori.
Questa funzione restituisce un nuovo dataframe per non operare 'in place' e modificare il nostro dataframe.
Di solito abbiamo, gli stessi metodi esistono sia in place, sia non in place in modo che restituiscano una nuova struttura dati.

I dati ottenuti, essendo percentuali dovrebbero essere seguiti da un '%', come potremmo sistemare questa cosa?

*print(publisher_rel_freq.apply(lambda p: np.round(100*p, 2)).astype(str).apply(lambda s: s + '%'))*

Publisher	Rel. frequence
ABC Studios	1.11%
DC Comics	33.61%
Dark Horse Comics	3.33%
George Lucas	3.06%
Hanna-Barbera	0.28%
Image Comics	0.83%
Marvel Comics	56.94%
Rebellion	0.28%
Star Trek	0.28%
Universal Studios	0.28%

Potrei convertire un tipo in stringa con il costruttore str() ma in questo caso ho una serie di elementi, quindi utilizzo .astype(str) per convertire elemento per elemento.
Aggiungendo delle parentesi all'inizio e alla fine posso spezzare la riga ad ogni invocazione del metodo indentando.

Avendo:

```
gender_freq = pd.crosstab(index =
heroes_with_year['Gender'],
columns = ['Abs. frequence'],
colnames = [''])
print(gender_freq)
```

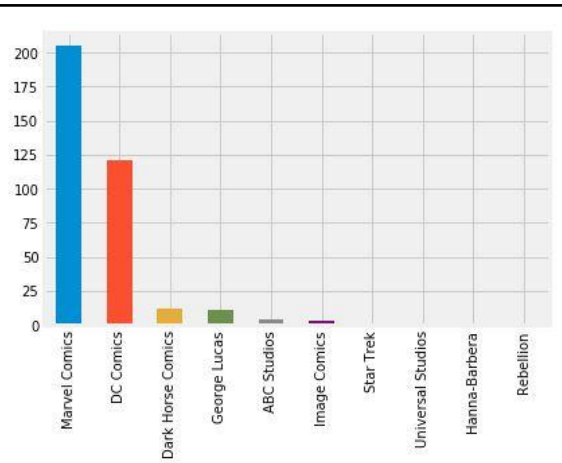
Gender	Abs. frequence
F	87
M	272

Se volessi ordinare il dataframe con prima 'M' e dopo 'F':

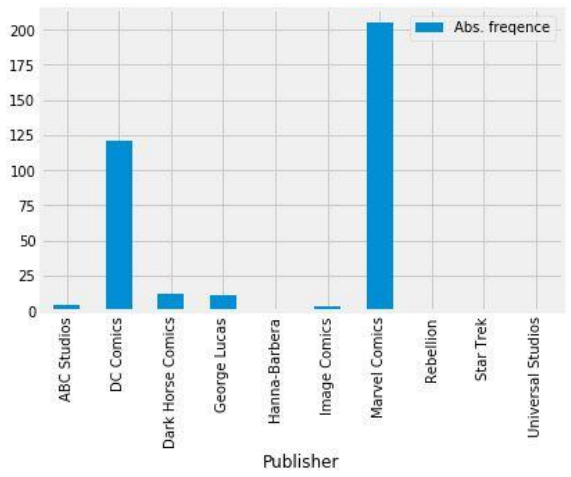
```
gender_freq.loc[['M', 'F']]
#accesso agli indici tramite una
lista con il nome delle righe
```

Gender	Abs. frequence
M	272
F	87

Produzione di grafici
Abbiamo già visto come produrre grafici in precedenza con l'utilizzo di "plot.bar()".

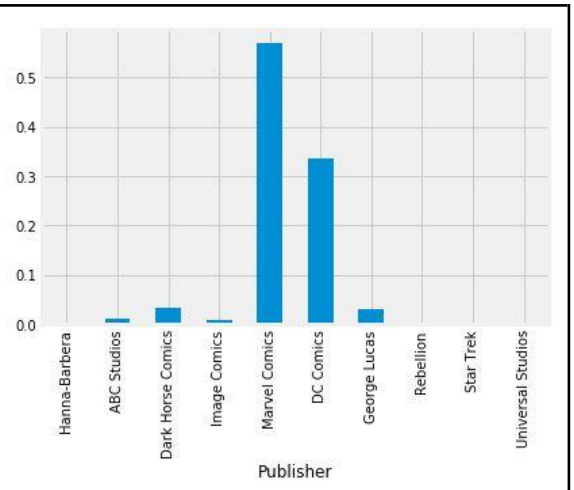
heroes_with_year['Publisher'].value_counts().plot.bar() plt.show()	

In questo caso non otteniamo informazioni quantitative, anche se sono comunque dei dati numerici che compongono il grafico a barre. Il fatto che non sia un dato quantitativo mi permette di usare il grafico a barre a cuor leggero.
Cosa succederebbe se usassi una crosstab per la creazione di un grafico a barre?

publisher_freq.plot.bar() plt.show() #in questo caso abbiamo anche una legenda con i nomi delle colonne, posso anche disattivare questa legenda inserendo come argomento in plot.bar(legend = False)	

Notiamo che in questo caso l'asse delle ordinate ha un nome ed i dati sono ordinati in base agli indici.
Se volessi ordinare il grafico in modo diverso?

```
publisher_order =
['Hanna-Barbera', 'ABC Studios',
'Dark Horse Comics',
                'Image Comics', 'Marvel
Comics', 'DC Comics',
                'George Lucas',
'Rebellion',
                'Star Trek', 'Universal
Studios']

publisher_rel_freq.loc[publisher_or
der,:].plot.bar(legend=False)
plt.show()
```



Creo una lista che poi passo come operatore di accesso, come nel caso di M e F.

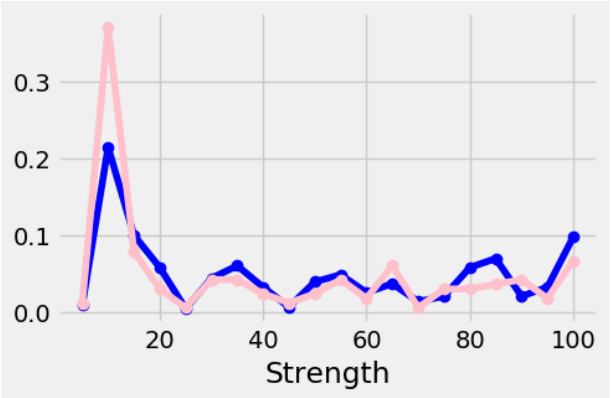
Le frequenze relative ci aiutano in molti casi, immaginiamo di voler differenziare il nostro campione in due popolazioni.
Noi diciamo di voler fare questa analisi per stratificazione, prendendo il campione e dividendolo in gruppi per un certo attributo.
Una volta averli differenziati posso analizzare il diagramma a barre con le frequenze relative (con quelle assolute le popolazioni devono avere più o meno lo stesso numero di componenti).

```
male_strength_freq = pd.crosstab(index=heroes.loc[heroes['Gender']=='M',
                                                  'Strength'],
                                 columns='Abs. freq.')
female_strength_freq =
pd.crosstab(index=heroes.loc[heroes['Gender']=='F',
                                                  'Strength'],
                                 columns='Abs. freq.')

num_male = sum(male_strength_freq['Abs. freq.'])
num_female = sum(female_strength_freq['Abs. freq.'])

print('Ci sono {} supereroi e {} supereroine'.format(num_male,
num_female))
```

In questo caso creiamo delle crosstab con frequenze assolute dei livelli di forza dei supereroi e delle supereroine.
Se provassimo a stampare i grafici delle loro frequenze relative?

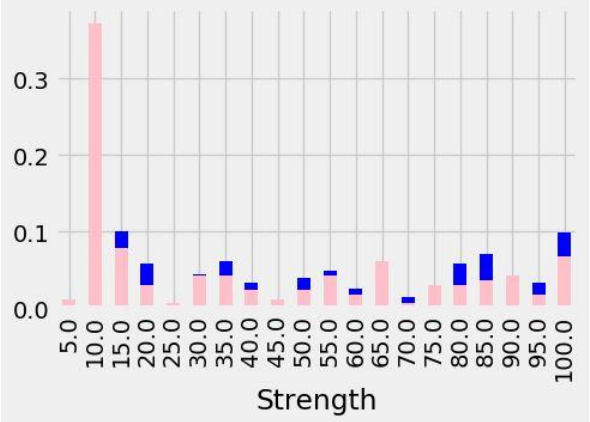
```
male_strength_freq = pd.crosstab(index=heroes.loc[heroes['Gender']=='M',
                                                  'Strength'],
```

```
                                columns='Rel. freq.', normalize=True)
female_strength_freq =
pd.crosstab(index=heroes.loc[heroes['Gender']=='F',
                                        'Strength'],
                                columns='Rel. freq.',normalize= True)

male_strength_freq.plot(marker='o', color='blue', legend=False)
female_strength_freq.plot(marker='o', color='pink', legend=False)
plt.show()
```



La funzione però non ha calcolato il grafico nella sua interezza, ha soltanto calcolato i punti per poi unirli.
Proviamo con i grafici a barre:

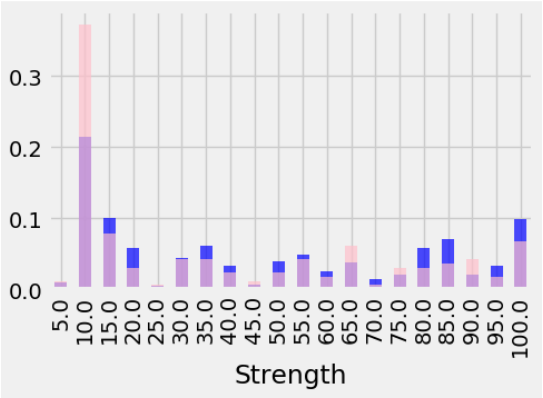
```
male_strength_freq.plot.bar(color='blue', legend=False)
female_strength_freq.plot.bar(color='pink', legend=False)
plt.show()
```



Potenzialmente però in questo caso, dato che il grafico a barre rosa è stato stampato successivamente a quello blu, potrebbero essere stati coperti dei dati.

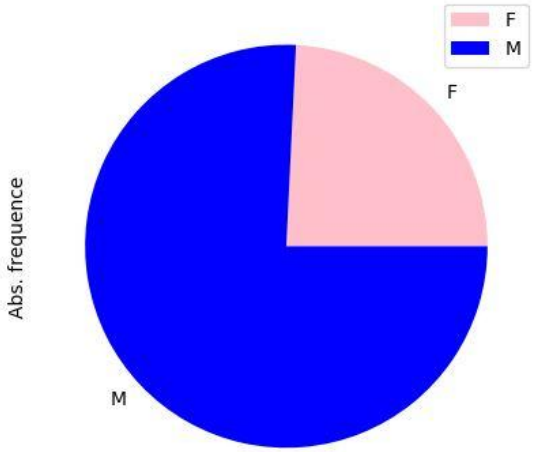
```
male_strength_freq.plot.bar(color='blue', alpha=.7)
```

female_strength_freq.plot.bar(color='pink', alpha=.7)
plt.show()



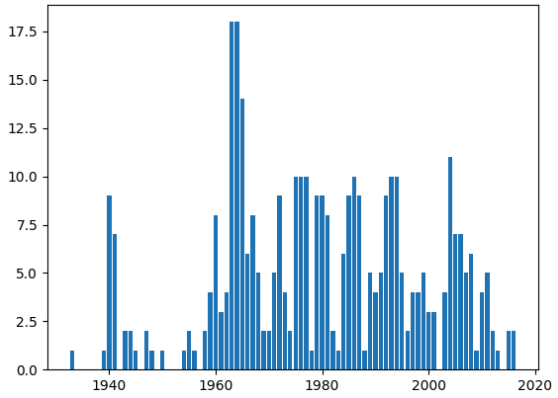
In questo caso riesco a vedere entrambi i campioni. Successivamente vedremo come produrre dei grafici con le barre affiancate.
Potrei migliorare due cose ancora di questo, la spaziatura e rimuovere i numeri in floating point sull'asse x.

Esiste un altro modo per visualizzare dei dati categorici qualitativi non ordinabili, cioè utilizzando l'aerogramma (grafico a torta).

gender_freq.plot.pie(y = 'Abs. frequence', colors = ['pink', 'blue']) plt.show()	

Potremmo migliorare il grafico rimuovendo la legenda in quanto ridondante.

```
first_app_freq = heroes_with_year['First appearance'].value_counts()
plt.bar(first_app_freq.index, first_app_freq.values)
plt.show()
```



Potremmo migliorare il grafico facendo in modo che ogni barra corrisponda ad un gruppo di anni.

```
plt.vlines(first_app_freq.index, 0, first_app_freq.values)
plt.show()
```

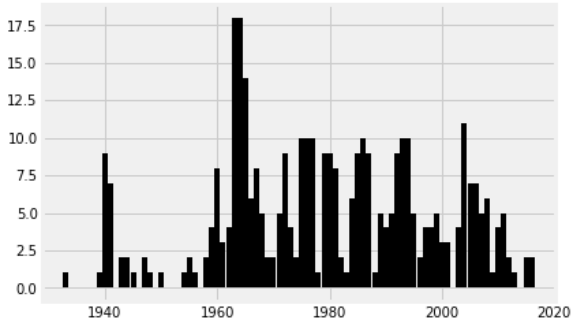
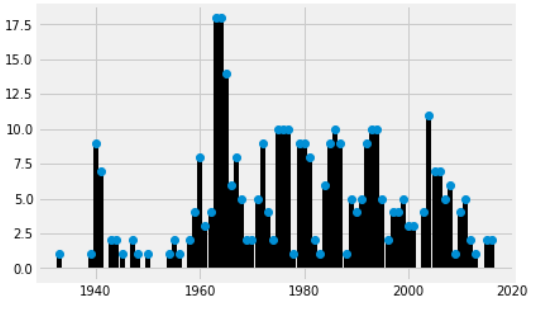


Grafico a bastoncini, 3 argomenti nella dichiarazione, 0 rappresenta l'ordinata dalla quale parte il segmento. Anche qui dovrei dichiarare una lista con il punto di partenza di ogni segmento ma inserendo solamente un numero viene fatto il broadcasting per tutti.

Potremmo aggiungere come elemento grafico un puntino sopra questi segmenti in modo da produrre un grafico a fiammiferi.

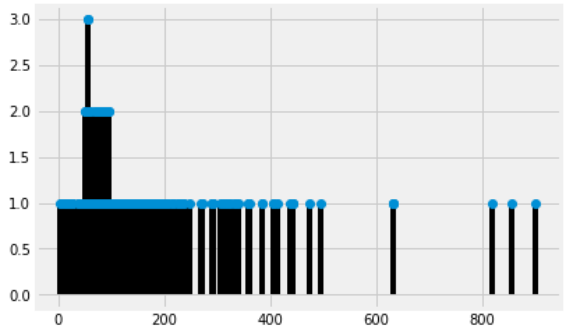
```
plt.vlines(first_app_freq.index, 0, first_app_freq.values)
plt.plot(first_app_freq.index, first_app_freq.values, 'o')
plt.show()
```



Come terzo argomento nella seconda chiamata per il grafico ho 'o' che rappresenta il puntino.

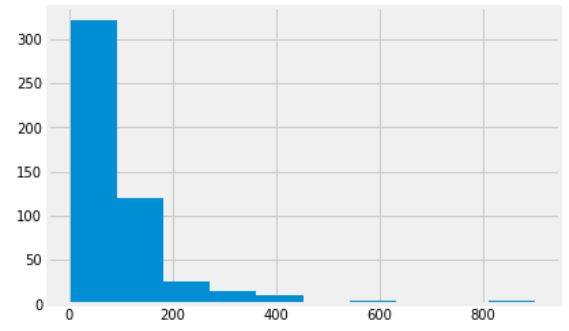
Possiamo usare questo tipo di grafico anche in caso di dati non molto espressivi.

```
weight_freq =
heroes['Weight'].value_counts()

plt.vlines(weight_freq.index, 0,
weight_freq.values)
plt.plot(weight_freq.index,
weight_freq.values, 'o')
plt.show()
```



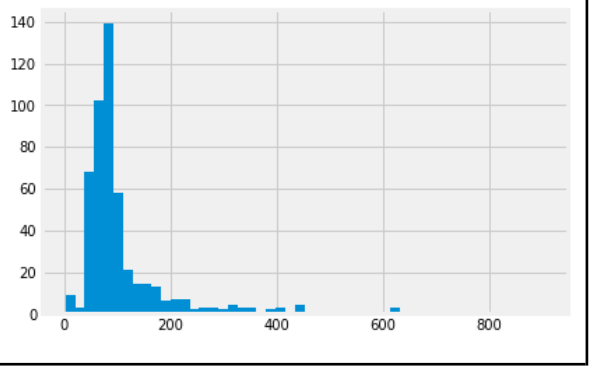
Quando abbiamo questo fenomeno è meglio fare aggregazione di valori, quando i nostri valori singolarmente non ci interessano ma ci interessa il loro range.
Possiamo usare l'istogramma per la rappresentazione di questi range.

```
heroes['Weight'].hist()
plt.show()
#nel caso di default otteniamo
poche barre per rappresentare i
range
```



Per aumentare il numero di barre modifico il parametro "bins"

```
heroes['Weight'].hist(bins = 50)
plt.show()
```

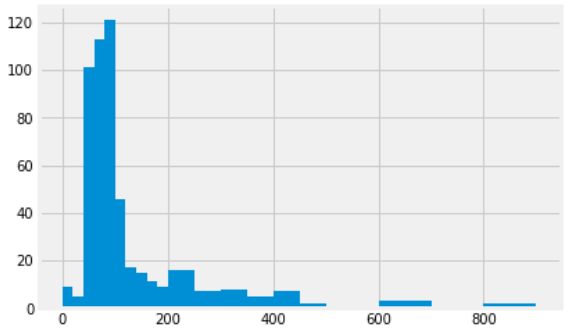


Notiamo che la crescita non è simmetrica (la crescita non ha la stessa progressione della decrescita) ed è unimodale (unico valore modale).

Inoltre la prima barra vicina allo 0 è più alta della successiva, potrebbero essere dei valori inseriti erroneamente o usando lo 0 per aggirare il NaN.

L'attributo bins è molto versatile, per definire i range nelle varie parti del grafico ad esempio.

```
heroes['Weight'].hist(bins =
np.hstack(np.arange(0, 200, 20),
np.arange(200, 500, 50),
np.arange(500, 1000, 100)))
plt.show()
```



Con gli .arange creo degli array con i range al loro interno mentre con .hstack li unisco.
Adesso che però le barre non hanno tutte la stessa base, le informazioni della frequenza su un determinato range ci vengono fornite dall'area.

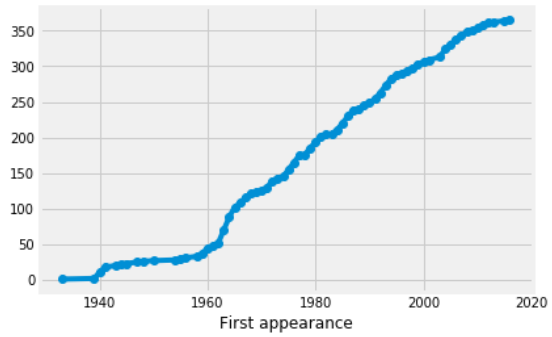
Frequenze cumulate
Mettiamo di voler vedere quanti supereroi sono apparsi prima del 1970.
Potremmo usare uno strumento per le frequenze cumulate.

```
first_app_freq_cumulate =
pd.crossbar(index =
heroes_with_year['First
appearance'], columns =
[Cumulate freq.], colnames =
['']).cumsum()
first_app_freq_cumulate.iloc[:10]
```

First appearance	Cumulate freq.
1933.0	1
1939.0	2
1940.0	11
1941.0	18
1943.0	20
1944.0	22
1945.0	23
1947.0	25
1948.0	26
1950.0	27

La somma cumulativa prende il valore corrispettivo all'indice e gli somma il valore precedente della somma cumulativa, quindi sarà anche crescente oltre che per gli indici anche per i valori.

Possiamo anche mostrarne il grafico.

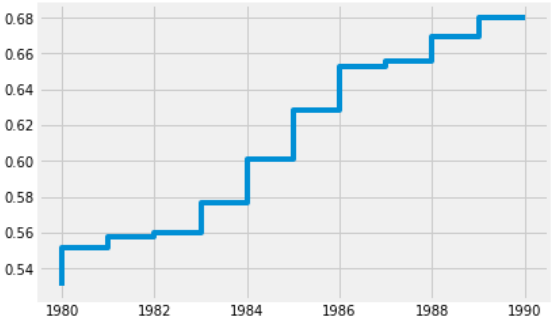
first_app_freq_cumulate.plot(marker = 'o', legend = False) plt.show()	

Funzione cumulativa empirica
Dato un insieme di osservazioni $\{x_1, x_2, \dots, x_n\}$ è definita come

$\widehat{F}: \Re \to [0, 1]$ tale per cui per ogni $x \in \Re$ assume un valore pari alla frequenza relativa delle osservazioni che risultano essere minori o uguali a x.

$$\widehat{F}(x) = \frac{\#\{x_i \leq x\}}{n} = \frac{1}{n}\sum_{i=1}^{n} I_{(-\infty, x]}(x_i)$$

Dove I è la funzione indicatrice.
Per un generico argomento x, la funzione cumulativa empirica assumerà pertanto come valore la frequenza relativa cumulata del più grande tra i valori osservati $x_i < x$.

import statsmodels.api as sm ecdf = sm.distributions.ECDF(heroes_with_year['First appearance']) x = np.arange(1980, 1991) y = ecdf(x) plt.step(x, y) plt.show()	

Lezione del 16 Marzo 2023

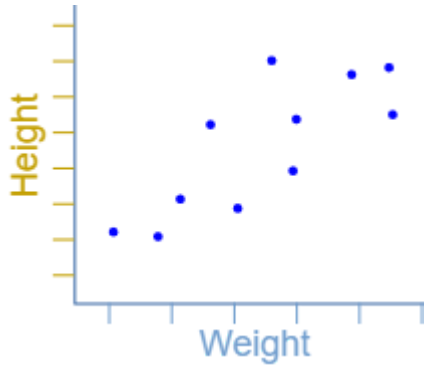
Lezione 6

Dalla lezione precedente è stata riscontrata una incongruenza tra le frequenze relative della forza dei supereroi e delle supereroine. Questo era dovuto agli indici diversi nei due grafici.
Possiamo sistemare questo errore con il metodo .reindex() nel quale inserisco tutti i possibili valori degli indici di forza.

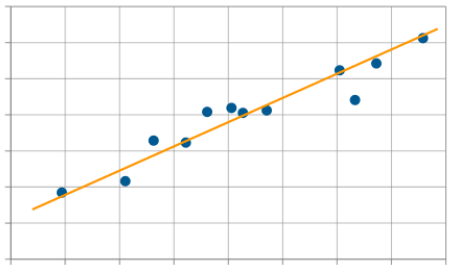
Questa cosa può essere fatta anche tramite il metodo .unique() su una determinata colonna.

Fino ad adesso, per tutti gli indici visti fin ora (media, mediana, ...) venivano applicati sulle singole serie.
Considerando un individuo e due attributi possiamo chiederci se 2 attributi di questo individuo siano collegati tra di loro.
Ad esempio, avendo uno dei due argomenti nel record posso poi andare ad inferire qualcosa sull'argomento mancante?

Prendiamo ad esempio un grafico cartesiano con come valori i 2 attributi da analizzare.
Attraverso i record riesco ad inserire, elemento per elemento, i punti nel grafico.
Esempio con peso e altezza all'interno di uno scatter plot.



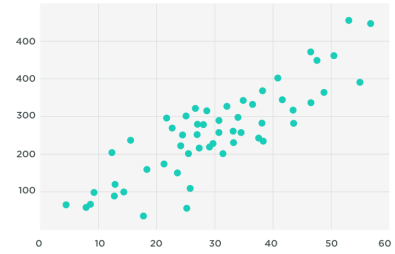
Da questa tabella non notiamo una dipendenza rigorosa tra i due attributi, ma notiamo che comunque con il crescere dell'altezza cresce anche il peso (e viceversa).
Eliminando dei punti isolati vedo quindi che comunque questi dati hanno un andamento all'incirca crescente.



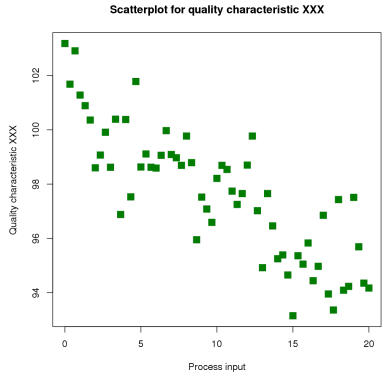
Una volta trovato un andamento avrei trovato anche un modello per descrivere questo tipo di relazione, quindi potrei anche fare a meno di una delle due colonne nel dataset.
Posso anche usare il grafico per controllare se esistono valori di peso o altezza fuori scala oppure per vedere dato uno dei due valori come sarà all'incirca l'altro.

Relazione diretta

Al crescere di un parametro cresce anche l'altro e viceversa con la decrescita.
Le relazioni dirette non sono sempre lineari, possono anche essere logaritmiche ad esempio.



Le relazioni possono anche essere indirette, con il principio contrario delle precedenti.



Non è detto che esistano solo questi due casi di grafici. Nel caso due dati non siano minimamente connessi a livello logico tra loro otterrei un grafico con punti uniformemente sparsi.
Fissando poi un valore in questo tipo di grafico possiamo vedere una serie di valori dell'altro parametro associati.

E' possibile che analizzando un dataframe io possa ottenere una relazione che non esiste nella realtà?
Abbiamo vari casi:

- Non abbiamo abbastanza dati
- Abbiamo sbagliato qualcosa
- La relazione è supportata dai dati ma non era ancora stata scoperta

Riusciamo anche ad avere un indice della correlazione fra due attributi?
Mettiamo di avere una relazione diretta con:

- Attributo x con osservazioni $x_1, x_2, \dots, x_n$ e media $\bar{x}$

- Attributo y con osservazioni $y_1, y_2, \dots, y_n$ e media $\bar{y}$

Abbiamo che:
x_i è piccolo se e solo se y_i è piccolo oppure x_i è grande se e solo se y_i è grande.
Dove:
- Piccolo = minore della media
- Grande = maggiore della media

Ottengo
$(x_i \leq \bar{x} \wedge y_i \leq \bar{y}) \vee (x_i \geq \bar{x} \wedge y_i \geq \bar{y})$
$(x_i - \bar{x} \leq 0 \wedge y_i - \bar{y} \leq 0) \vee (x_i - \bar{x} \geq 0 \wedge y_i - \bar{y} \geq 0)$
$(x_i - \bar{x})(y_i - \bar{y}) \geq 0$
O sono tutti e due maggiori di zero o tutti e due minori di zero.
$\sum_{i=i}^{n}(x_i - \bar{x})(y_i - \bar{y})$ sarà allora positivo nel caso di relazione diretta.

Vediamo ora il caso di relazione inversa
x_i piccolo se e solo se y_i grande oppure x_i grande se e solo se y_i piccolo.
Ottengo
$(x_i \leq \bar{x} \wedge y_i \geq \bar{y}) \vee (x_i \geq \bar{x} \wedge y_i \leq \bar{y})$
$(x_i - \bar{x})(y_i - \bar{y}) \leq 0$
Nel caso di relazione inversa il prodotto sarà negativo, quindi anche
$\sum_{i=i}^{n}(x_i - \bar{x})(y_i - \bar{y})$ sarà negativa nel caso di relazione inversa.

Covarianza campionaria
Si indica con $cov(x, y)$
$\frac{1}{n-1}\sum_{i=i}^{n}(x_i - \bar{x})(y_i - \bar{y})$
Se è > 0 è diretta mentre se è < 0 è inversa.
Nel caso non ci sia una relazione tra i due attributi la covarianza tenderà a zero.

Abbiamo anche un altro indice derivato da quest'ultimo detto "indice di correlazione lineare".
$\rho_{x,y} = \frac{cov(x,y)}{s_x s_y}$
Questo secondo indice, dal nome, ci fa intuire che sia un indice per controllare non solo se la relazione sia diretta o inversa, ma anche se sia lineare.

Quindi avendo una correlazione lineare tra gli attributi mi ritroverei nel caso:

$\exists a,\ b \in \Re$

$\forall i = 1,\ 2,\ \ldots,\ n$

$y_i = a + bx_i$

$\overline{y} = a + b\overline{x}$

$s_y = |b| s_x$

$$cov(x,\ y) = \frac{1}{n-1}\sum_{i=i}^{n}(x_i - \overline{x})b(x_i - \overline{x}) = b \cdot \frac{1}{n-1}\sum_{i=i}^{n}(x_i - \overline{x})(x_i - \overline{x}) = b\frac{1}{n-1}\sum_{i=i}^{n}(x_i - \overline{x})^2$$

$$= bs_x^{\ 2}$$

$$\rho_{x,\,y} = \frac{bs_x^{\ 2}}{s_x |b| s_x} = \frac{b}{|b|}$$

Quindi sarà uguale a 1 se $b > 0$ mentre sarà uguale a -1 se $b < 0$.
Ottengo 1 con una relazione lineare diretta mentre ottengo -1 con relazione lineare indiretta.

$-1 \leq \rho \leq 1$

Più sono vicino agli estremi del dominio e più la mia relazione sarà tendente alla linearità.
Viene anche chiamato "indice di correlazione lineare di Pearson".
Questo indice funziona anche per relazioni lineari in generale.

Esempio
Due campioni

$x_1,\ \ldots\ ,\ x_n$

$y_1,\ \ldots\ ,\ y_n$

Mettiamo di voler applicare due trasformazioni diverse a questi campioni:

$x_i' = a + bx_i$

$y_i' = c + dy_i$

$$\rho' = \frac{1}{n-1} \cdot \frac{\sum_{i=i}^{n}(x_i' - \overline{x'})(y_i' - \overline{y'})}{s_x' s_y'}$$

$x_i' - \overline{x'} = b(x_i - \overline{x})$

$y_i' - \overline{y'} = d(y_i - \overline{y})$

$s_x' = |b| s_x$

$s_y' = |d| s_y$

$$= \frac{1}{n-1} \cdot |b| \cdot |d| \cdot \frac{\sum_{i=i}^{n}(x_i - \overline{x})(y_i - \overline{y})}{|b||d| s_x s_y} = \frac{1}{n-1} \cdot \frac{bd}{|b||d|} \cdot \frac{\sum_{i=i}^{n}(x_i - \overline{x})(y_i - \overline{y})}{s_x s_y} = \frac{bd}{|b||d|} \cdot \rho$$

Il risultato sarà ρ se b e d sono concordi, mentre sarà $-$ ρ se b e d sono discordi.
Mentre avere dei valori di ρ vicini a 1 e -1 ci consente di implicare che vi sia una relazione lineare, avere dei valori di ρ o di $cov(x, y)$ vicini allo zero non ci garantisce che esista o meno una relazione tra i due attributi.

Mettiamo adesso di essere interessati ad un solo attributo.
Mettiamo anche che questo attributo sia interpretabile come valore di ricchezza (esempio: soldi, spazio occupato di una stanza, ecc...).
Immaginiamo che questo attributo rappresenti il patrimonio di una famiglia e che noi volessimo chiederci com'è distribuita la ricchezza totale (possiamo chiederlo per qualsiasi cosa di distribuibile).

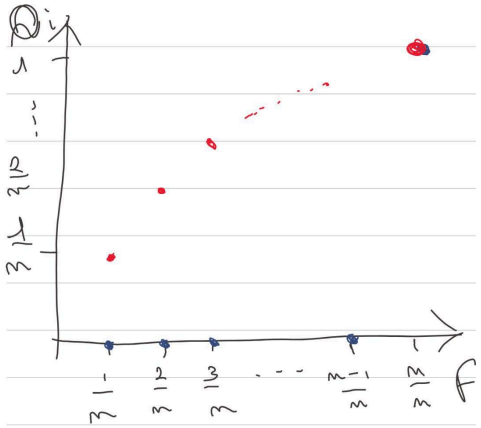
Abbiamo
$a_1, a_2, \dots, a_n$ ricchezze
$\sum_{i=i}^{n} a_i$ totale della ricchezza
Abbiamo due casi limite:

- Caso sperequato
  - $a_1, a_2, \dots, a_{n-1} = 0$
  - $a_n = tot$
- Caso equo
  - $a_1, a_2, \dots, a_n = \overline{a}$

Questi sono solo due casi, sarebbe utile avere un indice per capire a quale di questi due casi tendano i campioni delle ricchezze.
Utilizzeremo due indici:

- $Q_i = \frac{1}{tot} \sum_{j=1}^{i} a_j$ indica la ricchezza di una frazione di individui sulla ricchezza totale
- $F_i = \frac{i}{n}$ frazione di individui sul totale

$F_i \leq Q_i$
Se provassi a disegnare un grafico della situazione:



Caso medio
Caso sperequato

Congiungendo i punti nel primo caso ottengo la bisettrice di questo piano cartesiano, nel secondo caso avrei un grafico maggiormente vicino all'asse x.
Il grafico del primo caso prende il nome di "curva di Lorentz".
Come faccio a calcolare l'area di concentrazione tra la curva di Lorentz ed il nostro campione?
Notiamo che se l'area è uguale a 0 sono nel caso equo.

Indice di Gini (per la concentrazione)

$$G = \frac{\sum_{i=1}^{n-1} Q_i - F_i}{\sum_{i=1}^{n-1} F_i}$$

Il denominatore fa in modo che l'indice sia normalizzato.
Notiamo inoltre che il denominatore si può semplificare

$$\sum_{i=1}^{n-1} F_i = \sum_{i=1}^{n-1} \frac{i}{n} = \frac{1}{n} \sum_{i=1}^{n-1} i = \frac{1}{n} \frac{(n-1)n}{2} = \frac{n-1}{2}$$

Quindi l'indice diventa:

$$G = \frac{\sum_{i=1}^{n-1} Q_i - F_i}{(n-1)/2}$$

Trasformazione dei dati
Fino ad oggi abbiamo visto solo la traslazione e citato la normalizzazione.
Partiamo con un insieme di dati:
$x_1, x_2, \dots, x_n$ Osservazioni

Valori	Valori trasformati	Frequenze

v_1	v_1'	f_1
v_2	v_2'	f_2
...	...	...
v_m	v_m'	f_m

($m \leq n$)
Posso concludere che le frequenze di questi valori non cambieranno se e solo se la funzione applicata ad essi è iniettiva, quindi analizzeremo solo questo tipo di funzioni.

Traslazione
$v \to v + k \quad k \in \Re$
Media, mediana e moda vengono traslate.
Il range interquartile e dell'intera distribuzione rimane invariato.
$k = -(min\, x_i)$
Il primo valore si trova sullo zero.
$k = \bar{x}$
I valori vengono centrati rispetto alla media.

Scalatura
$v \to \frac{v}{h} \qquad h \in \Re$
Media, mediana e moda scalano.
I range e le varianze cambiano.
$h = min\, x_i$
Il numero più piccolo sarà 1
$h < min\, x_i$
Tutti i dati trasformati sono maggiori di 1

Trasformazione lineare
Prendendo $v_i \in [a, b]$ e $v_i' \in [c, d]$
Segnando su due assi diversi i due domini otterrei una retta passante per i punti (a, c) e (b, d).
Adesso non faccio altro che applicare la formula della retta:
$\frac{v'-c}{d-c} = \frac{v-a}{b-a}$
$v' = c + \frac{d-c}{b-a}(v - a)$

Casi particolari:
- $c = 0$ e $d = 1$

- $v' = \frac{v-a}{b-a}$

- $c = -1$ e $d = 1$
  - $v' = -1 + \frac{2}{b-a}(v - a) = \frac{2(v-a)}{b-a} - 1$

Standardizzazione
$v' = \frac{v-\bar{x}}{s_x}$
La media campionaria standardizzata sarà uguale a zero.
La varianza?
$(s_x')^2 = s^2_{x_i-\bar{x}} \cdot \frac{1}{s^2_x} = 1$

Trasformazione logaritmica
$v \to log\ v$
Serve per portare dati di unità di misura diverse sullo stesso piano, ad esempio 10, 100000, e 10000000000 possono stare sullo stesso piano se in log_{10}.

Lezione del 21 Marzo 2023

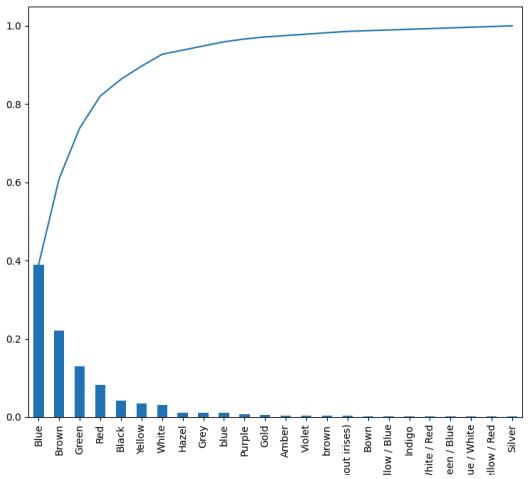
Lezione 7

Diagrammi di Pareto
Con l'utilizzo delle frequenze assolute o relative ho accesso a informazioni aggiuntive.
Oggi introduciamo un nuovo tipo di grafico, il diagramma di Pareto, che unisce la rappresentazione di frequenze e frequenze cumulate.

N.B. Frequenza cumulata = frequenza dell'elemento + delle osservazioni precedenti.

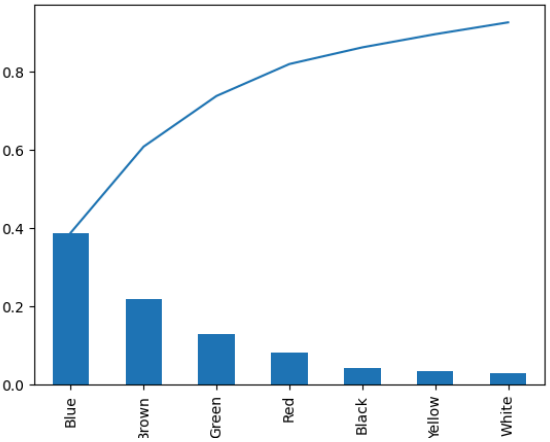
Nel caso in cui i parametri non siano ordinabili posso creare questi diagrammi ordinandoli tramite le frequenze.
Lo scopo del diagramma di Pareto è quello di mostrarci quale insieme dei valori possibili cumula ad una certa frequenza.
Quindi grazie a questo grafico ottengo direttamente il sottoinsieme minimo di sottopopolazioni che raggiungono una certa frequenza cumulativa desiderata.
Le frequenze relative vengono rappresentate da delle barre, mentre quelle cumulate da una spezzata.
Siamo sicuri che la linea spezzata non venga nascosta dalla prima barra dato che per definizione parte proprio da lì.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

eye_color = heroes['Eye color']
eye_color_freq =
eye_color.value_counts(normalize=
True)

eye_color_freq.cumsum().plot()
eye_color_freq.plot.bar()
plt.show()
```



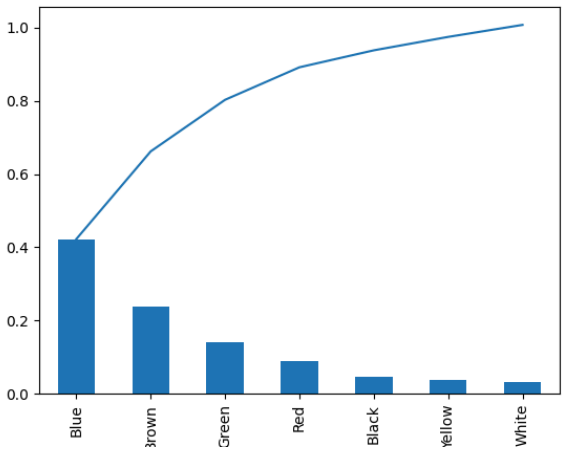
Posso mettere un filtro sui valori bassi per la quale non vedrei cambiamenti significativi tra le barre ma soltanto la spezzata crescere.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

eye_color = heroes['Eye color']
eye_color_freq =
eye_color.value_counts(normalize=
True)

eye_color_freq[eye_color_freq>.02].
cumsum().plot()
eye_color_freq[eye_color_freq>.02].
plot.bar()
plt.show()
```



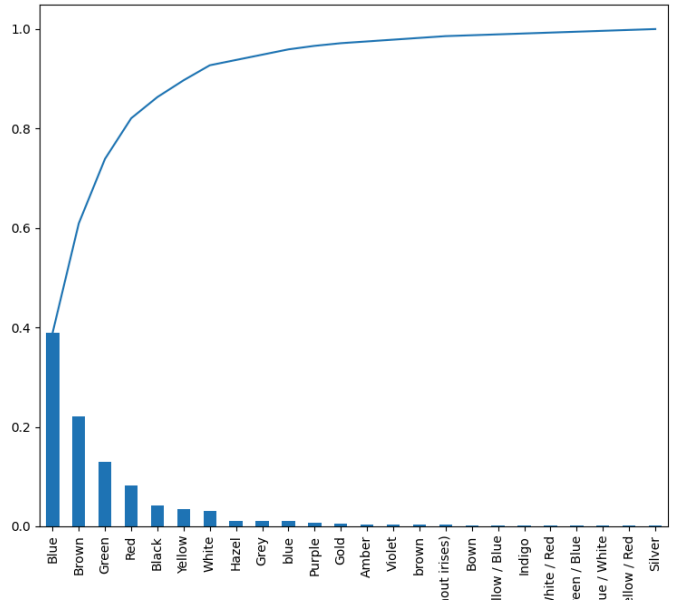
La parte interessante del grafico risiede soltanto nella prima parte dove posso vedere il minimo sottoinsieme che raggiunge una certa percentuale.
Potrei, oltre al rimuovere le frequenze sotto il 2%, normalizzare il resto dei dati dividendolo per la somma delle frequenze dei dati non scartati.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

eye_color = heroes['Eye color']
eye_color_freq =
eye_color.value_counts(normalize=
True)

norm_eye_color_freq =
eye_color_freq[eye_color_freq>.02]/
0.92
norm_eye_color_freq.cumsum().plo
t()
norm_eye_color_freq.plot.bar()
plt.show()
```

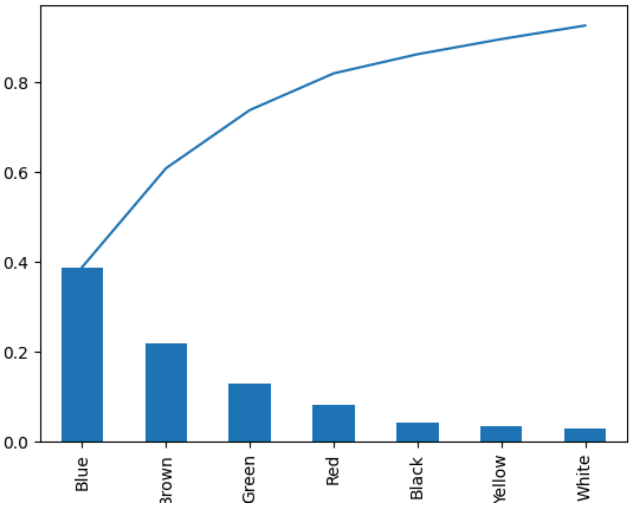


Possiamo quindi implementare anche noi una funzione che crei un grafico di Pareto, inserendo la soglia minima e la normalizzazione una volta rimosse le frequenze inferiori.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv', sep =';', index_col = 0)

def my_pareto(data, threshold = 0.02, renormalize = False):
    freq = data.value_counts(normalize = True)
    freq = freq[freq > threshold]
    if renormalize:
        freq = freq/sum(freq)
    freq.cumsum().plot()
    freq.plot.bar()
    plt.show()

my_pareto(heroes['Eye color'], threshold=0)
```

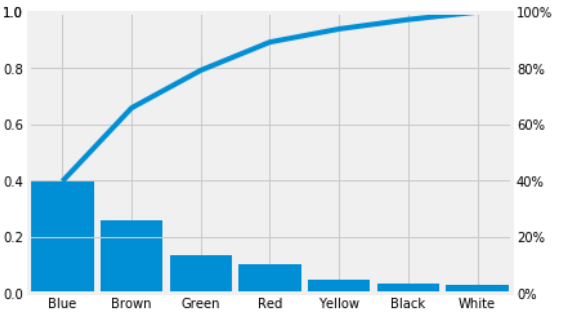


my_pareto(heroes['Eye color'], threshold=0.015)



Oltre a creare manualmente questa funzione possiamo usare un package chiamato “paretochart“.

```
from paretochart import pareto

eye_color = heroes['Eye color']
eye_color_freq =
eye_color.value_counts(normalize=
True)

common_colors =
eye_color_freq[eye_color_freq >
.02].index
common_color_data =
```



```
eye_color[eye_color.isin(common_colors)]

pareto(common_color_data.value_counts(normalize=True),
       labels=common_colors)

plt.show()
```

Frequenze congiunte e marginali
Durante la lezione precedenti abbiamo considerato correlazioni tra attributi a coppie, guardando individuo per individuo i valori per questa coppia per controllare se siano o meno relazionate.
Finora abbiamo usato crosstab con una sola colonna.
Possiamo creare anche crosstab con più colonne.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv', sep =';', index_col = 0)

int_gender_freq = pd.crosstab(index = heroes['Intelligence'], columns = heroes['Gender'])
print(int_gender_freq)
```

Gender	F	M
Intelligence		
average	38	101
good	78	165
high	27	112
low	0	13
moderate	21	37

In questo caso però notiamo che i valori degli indici non sono ordinati secondo la scala di Likert ma in ordine alfabetico.
Dato che in questa tabella abbiamo due insieme di frequenze otteniamo delle frequenze congiunte.
Un crosstab in python non è altro che un dataset, per cambiare gli indici potremmo usare il metodo reindex().

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

int_gender_freq =
pd.crosstab(index =
heroes['Intelligence'], columns =
heroes['Gender'])

int_gender_freq =
int_gender_freq.reindex(['low',
'moderate', 'average', 'good', 'high'])
print(int_gender_freq)
```

Gender	F	M
Intelligence		
low	0	13
moderate	21	37
average	38	101
good	78	165
high	27	112

Per poi invece cambiare l'ordine delle colonne, siccome crosstab produce dataset, uso .loc().

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

int_gender_freq =
pd.crosstab(index =
heroes['Intelligence'], columns =
heroes['Gender'])

int_gender_freq =
int_gender_freq.reindex(['low',
'moderate', 'average', 'good', 'high'])

print(int_gender_freq.loc[:,['M', 'F']])
```

Gender	M	F
Intelligence		
low	13	0
moderate	37	21
average	101	38
good	165	78
high	112	27

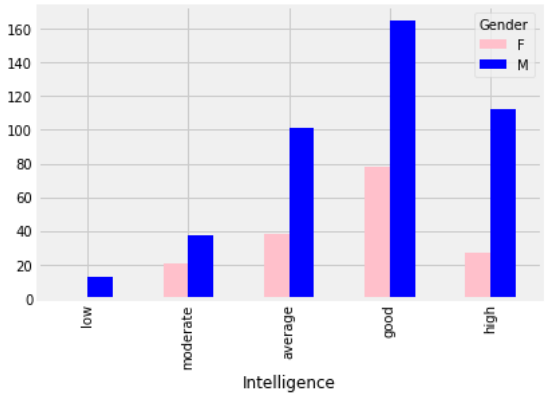
Dopo averlo ordinato posso accedervi come sublicing (in questo caso però non ottengo un grafico delle frequenze dato che sto eliminando dei dati).

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv', sep =';', index_col = 0)

int_gender_freq = pd.crosstab(index = heroes['Intelligence'], columns = heroes['Gender'])

int_gender_freq = int_gender_freq.reindex(['low', 'moderate', 'average', 'good', 'high'])

print(int_gender_freq.loc['moderate':'good', :])
```

Gender	F	M
Intelligence		
moderate	21	37
average	38	101
good	78	165

Cosa succede adesso se provo a invocare il metodo plot.bar()?

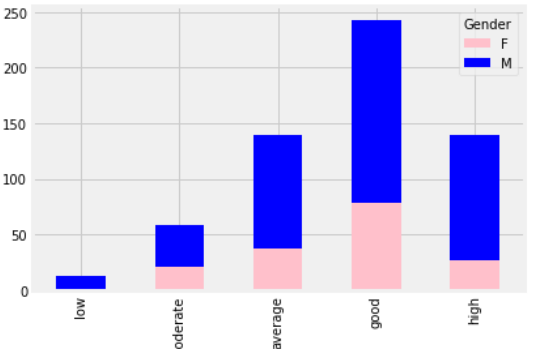
```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv', sep =';', index_col = 0)

int_gender_freq = pd.crosstab(index = heroes['Intelligence'], columns = heroes['Gender'])

int_gender_freq = int_gender_freq.reindex(['low', 'moderate', 'average', 'good', 'high'])

int_gender_freq.plot.bar(color = ['pink', 'blue'])
plt.show()
```



Questo non è un buon grafico dato che sto graficando le frequenze assolute per i due generi, che però non sono equamente rappresentati nel campione, dovremmo normalizzare.
Potremmo usare nel metodo .plot.bar() l'attributo stacked = True per mettere una barra sopra l'altra.

In questo caso non è molto utile, sarebbe utile nel caso la loro somma fosse il 100% per mostrare la distribuzione.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

int_gender_freq =
pd.crosstab(index =
heroes['Intelligence'], columns =
heroes['Gender'])

int_gender_freq =
int_gender_freq.reindex(['low',
'moderate', 'average', 'good', 'high'])

int_gender_freq.plot.bar(color=['pin
k', 'blue'], stacked=True)
plt.show()
```



Mettiamo adesso di voler controllare il peso invece dell'intelligenza.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

print(pd.crosstab(index=heroes['W
eight'],
columns=[heroes['Gender']]).iloc[:10,:
])
```

Gender	F	M
Weight		
2.34	0	1
4.82	0	1
9.79	0	1
14.67	0	1
16.20	0	1
17.01	0	1
18.11	0	1
18.45	0	1
19.00	0	1
25.73	0	1

Vediamo che per ogni valore di peso abbiamo una corrispondenza.

Questo non è per niente informativo, come faccio a risolvere questo problema?
Considero adesso valori di range di peso.
Lo facciamo con un minimo pre-processing di weight.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv', sep =';', index_col = 0)

print(pd.crosstab(index=pd.cut(heroes['Weight'], bins=[30, 50, 80, 100, 200, 500, 1000]), columns=[heroes['Gender']]))
```

Gender	F	M
Weight		
(30, 50]	4	4
(50, 80]	116	88
(80, 100]	5	111
(100, 200]	11	84
(200, 500]	5	38
(500, 1000]	1	5

Ottengo range aperti da un lato per fare in modo che il valore a sinistra venga considerato una sola volta.
Se aggiungo il parametro margins = True alle frequenze congiunte vengono aggiunte le somme di righe e colonne.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv', sep =';', index_col = 0)

int_gender_freq = pd.crosstab(index = heroes['Intelligence'], columns = heroes['Gender'], margins = True)
print(int_gender_freq)
```

Gender	F	M	All
Intelligence			
average	38	101	139
good	78	165	243
high	27	112	139
low	0	13	13
moderate	21	37	58
All	164	428	592

In questo modo nell'ultima colonna ottengo le frequenze assolute delle intelligenze e nell'ultima riga ottengo le frequenze assolute delle colonne (il numero totale di elementi per ogni colonna).
Questo tipo di frequenze vengono dette "frequenze marginali".
Se normalizziamo queste tabelle con le frequenze marginali ho due metodi, inserendo il parametro normalise passando come parametro 'all' per normalizzare per il totale.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

int_gender_freq =
pd.crosstab(index =
heroes['Intelligence'], columns =
heroes['Gender'], margins = True,
normalize='all')
print(int_gender_freq)
```

Gender	F	M	All
Intelligence			
average	0.064189	0.170608	0.234797
good	0.131757	0.278716	0.410473
high	0.045608	0.189189	0.234797
low	0.000000	0.021959	0.021959
moderate	0.035473	0.062500	0.097973
All	0.277027	0.722973	1.000000

Possiamo normalizzare anche secondo gli indici, infatti usando 'index' si otterrà una tabella in cui i valori su ogni riga sommano a 1.

```
pd.crosstab(index=heroes['Intelligence'], columns=heroes['Gender'], margins=True, normalize='index')
```

Gender	F	M
Intelligence		
average	0.273381	0.726619
good	0.320988	0.679012
high	0.194245	0.805755
low	0.000000	1.000000
moderate	0.362069	0.637931
All	0.277027	0.722973

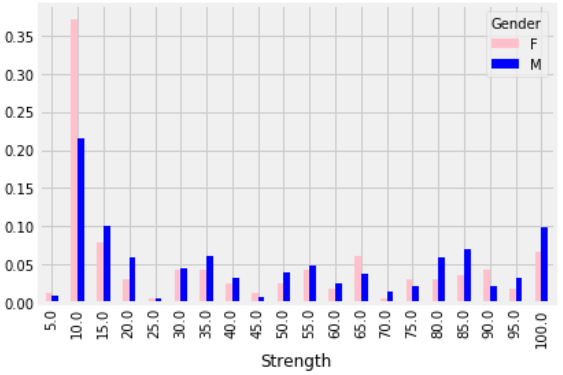
E ovviamente anche secondo le colonne, infatti indicando invece ‘columns’ viene generata una tabella in cui tutte le colonne sommano al valore unitario.

```
pd.crosstab(index=heroes['Intelligence'], columns=heroes['Gender'], margins=True, normalize='columns')
```

Gender	F	M	All
Intelligence			
average	0.231707	0.235981	0.234797
good	0.475610	0.385514	0.410473
high	0.164634	0.261682	0.234797
low	0.000000	0.030374	0.021959
moderate	0.128049	0.086449	0.097973

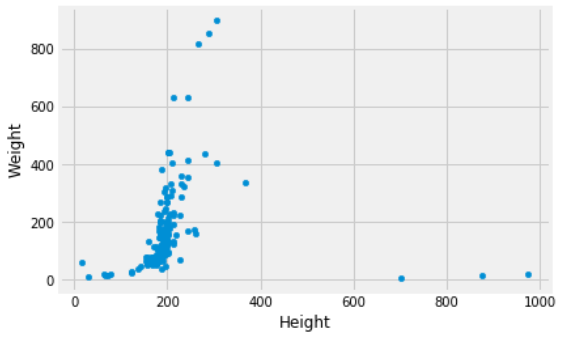
Normalizzando per colonne ottengo le frequenze relative dei generi, quindi posso poi usarlo per creare dei grafici normalizzati.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv', sep =';', index_col = 0)

pd.crosstab(index=heroes['Strength'], columns=[heroes['Gender']], normalize='columns').plot.bar(color=['pink', 'blue'], stacked=False)
plt.show()
```



Scatter plot o diagramma di dispersione
Per produrre questo tipo di grafico in python ho bisogno di informazioni che vengono da due parametri diversi.

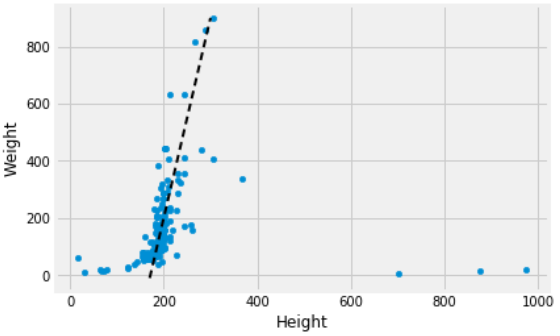
```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv', sep =';', index_col = 0)

heroes[heroes['Gender']=='M'].plot.scatter('Height', 'Weight')
plt.show()
```



Dato che visivamente notiamo che vi è una relazione diretta

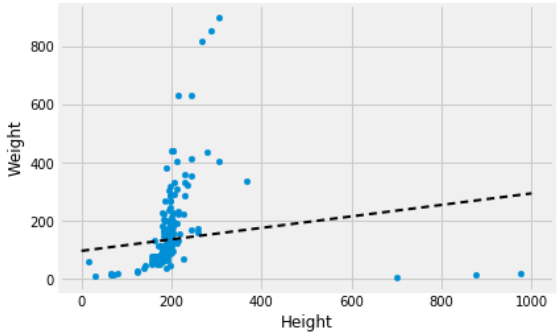
rimuovendo gli outlier potrei tracciare una retta che non si discosta da questi punti.
Con matplotlib è un po' complicato, possiamo andare per tentativi.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

heroes[heroes['Gender']=='M'].plot.s
catter('Height', 'Weight')

trend = lambda x: -1200 + x * 7
x_range = [170, 300]
line, = plt.plot(x_range,
list(map(trend, x_range)),
color='black')
line.set_dashes([3, 2])
line.set_linewidth(2)
plt.show()
```



Oppure possiamo risalire a questa retta nel modo più preciso, ma molto più complicato.

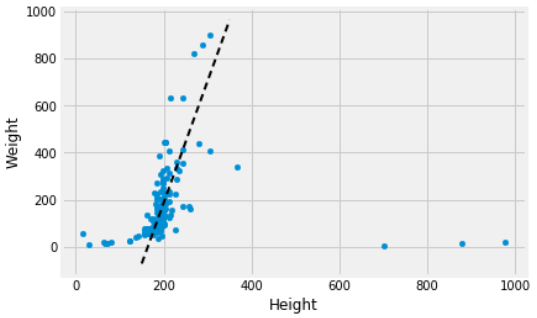
```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)
from sklearn import linear_model

regr =
linear_model.LinearRegression()

heroes_with_data =
heroes[heroes['Gender']=='M'].copy(
).dropna()

X = heroes_with_data.loc[:,
['Height']]
Y = heroes_with_data['Weight']

regr.fit(X, Y)
```



```
heroes[heroes['Gender']=='M'].plot.scatter('Height', 'Weight')

line, = plt.plot([0, 1000], regr.predict([[0], [1000]]), color='black')
line.set_dashes([3, 2])
line.set_linewidth(2)

plt.show()
```

Notiamo comunque che con questo metodo gli outlier influiscono sulla retta finale. Come risolviamo?

```
heroes_with_data = heroes_with_data[heroes_with_data['Height']<300]

X = heroes_with_data.loc[:, ['Height']]
Y = heroes_with_data['Weight']

regr.fit(X, Y)

heroes[heroes['Gender']=='M'].plot.scatter('Height', 'Weight')

line, = plt.plot([150, 350], regr.predict([[150], [350]]), color='black')
line.set_dashes([3, 2])
line.set_linewidth(2)

plt.show()
```

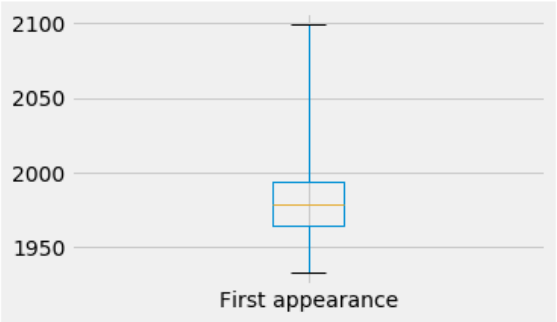


Nelle lezioni precedenti abbiamo parlato di quantili e box plot. Partiamo dall'indicare come ottenere gli indici principali in python:

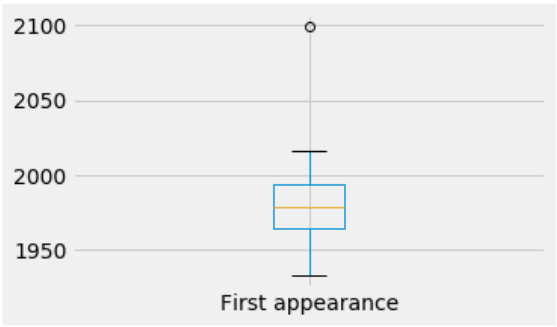
```
year = heroes['First appearance']
#Varianza campionaria
year.var()
#Deviazione standard
year.std()
#Tutti gli indici descrittivi generali
year.describe()
#Quantile definito
year.quantile(.15)
```

Anche per le serie di python posso invocare dei metodi per i quantili e per i box plot.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

year = heroes['First appearance']
year.plot.box(whis='range')
plt.show()
```

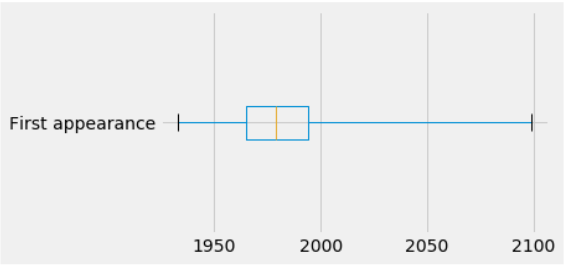



```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

year = heroes['First appearance']
year.plot.box()
plt.show()
```



Senza la specifica dell'argomento 'whis' si ottiene una versione diversa del grafico in cui vengono pre calcolati gli outlier e indicati poi con un pallino.
Per una versione differente orizzontale bisogna fare:

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

year = heroes['First appearance']
year.plot.box(vert=False,
whis='range')
plt.show()
```



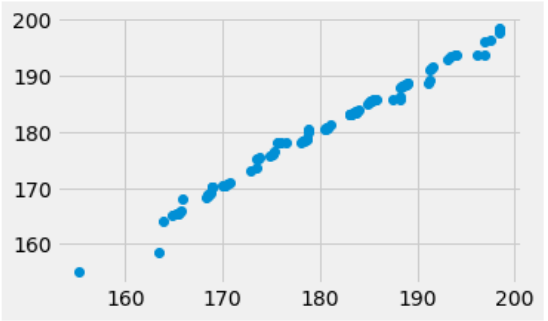
Anche i diagrammi QQ possono essere portati su python a mano o tramite librerie specifiche.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

marvel =
heroes.loc[(heroes['Publisher']=='M
arvel Comics') &
(heroes['Height'].between(150, 200))]

dc =
heroes.loc[(heroes['Publisher']=='DC
Comics') &
(heroes['Height'].between(150, 200))]

marvel_sample =
marvel['Height'].sample(120)
dc_sample =
dc['Height'].sample(120)

print((marvel_sample.quantile(.2),
dc_sample.quantile(.2)))
```

```
(170.186, 170.456)
```

In questo caso ottengo soltanto un punto nel diagramma. Se volessi ottenere il diagramma completo ripeto il passaggio.

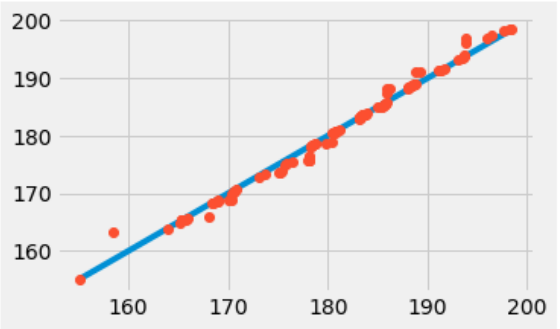
```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

marvel =
heroes.loc[(heroes['Publisher']=='M
arvel Comics') &
(heroes['Height'].between(150, 200))]

dc =
heroes.loc[(heroes['Publisher']=='DC
```



```
Comics') &
(heroes['Height'].between(150, 200))]

marvel_sample =
marvel['Height'].sample(120)
dc_sample =
dc['Height'].sample(120)

levels = np.linspace(0, 1, 100)
plt.plot(marvel_sample.quantile(lev
els), dc_sample.quantile(levels), 'o')
plt.show()
```

Mostriamo adesso una correlazione fra i due dati tracciando la bisettrice del quadrante.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

marvel =
heroes.loc[(heroes['Publisher']=='M
arvel Comics') &
(heroes['Height'].between(150, 200))]

dc =
heroes.loc[(heroes['Publisher']=='DC
Comics') &
(heroes['Height'].between(150, 200))]

marvel_sample =
marvel['Height'].sample(120)
dc_sample =
dc['Height'].sample(120)

print((marvel_sample.quantile(.2),
dc_sample.quantile(.2)))

levels = np.linspace(0, 1, 100)
plt.plot(marvel_sample.quantile(lev
els), dc_sample.quantile(levels), 'o')
plt.show()

plt.plot([min(dc_sample),
```

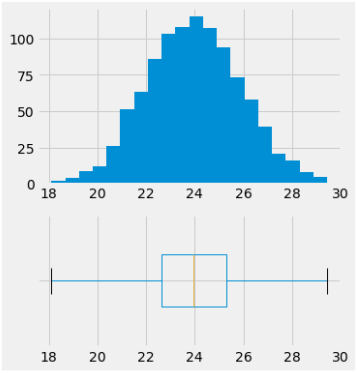


max(dc_sample)], [min(dc_sample), max(dc_sample)]) plt.plot(dc_sample.quantile(levels), marvel_sample.quantile(levels), 'o') plt.show()	

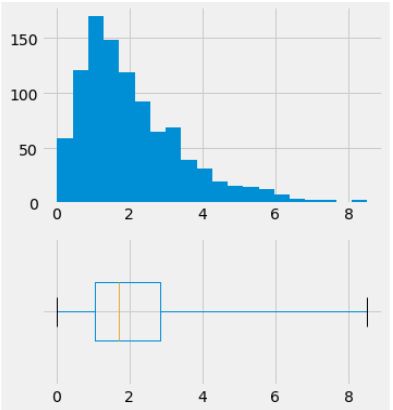
Proprietà particolare
Se in un istogramma trovo che le barre delle frequenze aggregate hanno un ordinamento unimodale (che può essere espressa in vari modi) non lineare e simmetrico, questo andamento "a campana" possiede due nome tecnici:

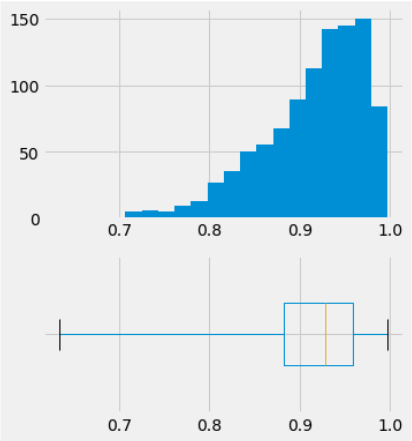
- Curva Gaussiana
- Curva normale

La cosa interessante di questo caso è che i dati rappresentati da questo istogramma, hanno un andamento per il quale moda e mediana si sovrappongono.
Lo vediamo accostando l'istogramma al box plot.



Quindi quando abbiamo questa simmetria a livello di grafico di solito moda, media e mediana tendono a sovrapporsi.
Se il grafico non è ordinato diciamo che possiede uno "skew" al lato dove abbiamo più dati.





In questo grafico inoltre:

- Il 68% dei dati si trova ad una deviazione standard dalla media
- Il 95% dei dati si trova ad due deviazione standard dalla media
- Il 99% dei dati si trova ad tre deviazione standard dalla media

```
def check_empirical_rule(n):
    within =
len(sample[np.abs(sample -
sample.mean()) < n*sample.std()])
    return  within / len(sample)

pd.DataFrame([check_empirical_r
ule(n) for n in range(1, 4)],
columns=['%'],
            index=range(1, 4))
```

	%
1	0.672269
2	0.947479
3	0.993697

#accedo controllando la distanza di n*deviazione standard dalla media

Eterogeneità
La maggior parte degli indici ai quali abbiamo fatto riferimento si poteva calcolare solo in base a dei dati quantitativi numerici.
Avendo invece un campione qualitativo ricado solitamente in alcuni casi particolari:

- Stesso valore qualitativo con massima omogeneità
- Vari valori qualitativi con massima eterogeneità

Il mio obiettivo è quello di trovare un indice che mi permetta di vedere vicino a quale caso particolare io mi trovi.

Indice di eterogeneità di Gini
$x_1, \dots, x_n$ ← Campione

v_1	f_1

v_2	f_2
...	...
v_m	f_m

Con $m \leq n$

$I = 1 - \sum_{i=1}^{m} f_i^2$

Questo indice quindi varierà tra 0 e 1 (0 compreso).
Come viene catturato questo indice di centralità?
Di sicuro $\exists j \;\; f_j > 0$

Notiamo anche che le frequenze relative non sono mai negative.

Quindi $\sum_{i=1}^{m} f_i^2$ è strettamente positivo.

Infine, $1 - \sum_{i=1}^{m} f_i^2$ è strettamente minore di 1.

Al contrario:

$\forall i \;\; f_i^2 \leq f_i$

$\sum_{i=1}^{m} f_i^2 \leq \sum_{i=1}^{m} f_i = 1$

$1 - \sum_{i=1}^{m} f_i^2 \geq 0$

Caso di minima eterogeneità
Ho un solo valore con frequenza relativa 1

$I = 1 - 1^2 = 0$

Caso di massima eterogeneità

$\forall i \;\; f_i = \frac{1}{m}$

$I = 1 - \sum_{i=1}^{m} \left(\frac{1}{m}\right)^2 = 1 - m\frac{1}{m^2} = \frac{m-1}{m} < 1$

Questo indice, senza conoscere i valori, non ci fornisce informazioni assolute nel caso massimo.

Abbiamo anche un indice normalizzato:

$I' = I \cdot \frac{k-1}{k}$

Abbiamo altri indici di eterogeneità, ad esempio l'entropia (prossima lezione) il quale può essere utilizzato per creare alberi di decisione.

Lezione del 23 Marzo 2023

Lezione 8

Durante la scorsa lezione abbiamo parlato del nostro primo indice per l'eterogeneità, cioè l'indice di Gini.
Esiste però anche un altro indice di eterogeneità, cioè l'entropia.
Anche questo indice si basa sulle frequenze relative:

$$H = \sum_{i=1}^{k} f_i log_2 \frac{1}{f_i}$$

Mostreremo, come per l'indice di Gini, come si comporta nei casi limite.

Per $H = 0$
Abbiamo una somma di positivi uguale a zero. Quindi avremo $k - 1$ valori di f_i che varranno 0 ed uno che varrà 1 in modo che $log_2 1 = 0$ e che quindi la sommatoria vada a zero. Quindi caso di eterogeneità minima.

Eterogeneità massima

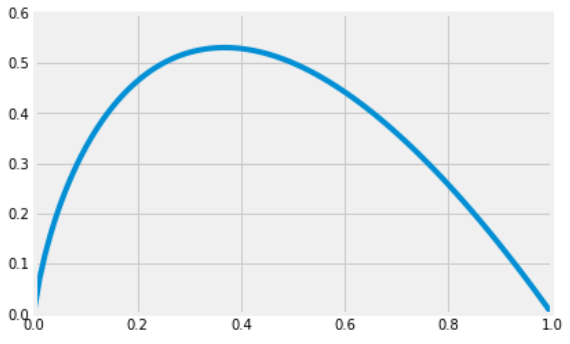
$$\forall i \quad f_i = \frac{1}{k}$$

$$\sum_{i=1}^{k} \frac{1}{k} log_2 k = log_2 k$$

L'indice di eterogeneità ci permette di iniziare a dare uno sguardo al machine learning.

Machine learning
Grafico dell'entropia in python

```
x = np.arange(0.001, 1.01, 0.01)
y = list(map(lambda f: -f * np.log2(f), x))
plt.plot(x, y)
plt.ylim(0, 0.6)
plt.xlim(0, 1)
plt.show()
```

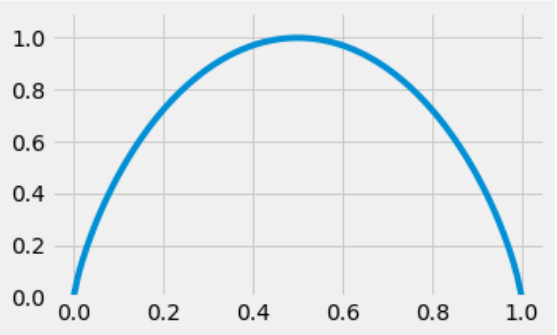


Abbiamo precedentemente notato che il valore massimo assunto dall'entropia è $log_2 k$, quindi sarà possibile normalizzare questo indice:

$H' = \frac{H}{log_2 k}$

Grafico dell'entropia normalizzata in python:

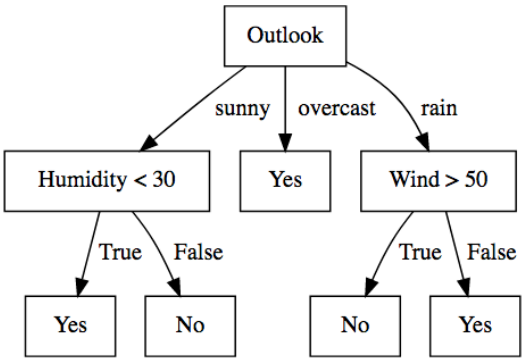
```
def entropy_2_val(f):
    return 0 if f in (0, 1) else - f *
np.log2(f) - (1-f) * np.log2(1-f)

x = np.arange(0, 1.01, .01)
y = list(map(entropy_2_val, x))
plt.plot(x, y)
plt.ylim((0, 1.1))
plt.show()
```



I dati in forma non normalizzata non ci dicono molto sull'eterogeneità del campione se non conosciamo il numero di valori possibili.
Quando invece sono normalizzati posso sapere invece quanto sono omogenei, e quindi quanto più si avvicinano a 1.

Alberi di decisione
Mettiamo di avere una serie di oggetti e dovendoli inserire in delle categorie (ad esempio, avere un sistema che riconosce la persona alla quale aprire la porta a partire dal volto).
Sarebbe comodo avere un sistema esplicito di ragionamento per cui il risultato si ottiene per una determinata interpretazione.

Consideriamo per esempio l'albero riportato qui sotto, che fa riferimento a un dataset molto semplice che in funzione delle condizioni meteorologiche permette di capire se si può uscire a giocare. La radice richiede di iniziare valutando che tempo fa (attributo Outlook): se è nuvoloso (overcast) si arriva a una foglia che dice che si può uscire (Yes); se invece dovesse essere soleggiato viene richiesto di valutare se l'umidità abbia o meno un valore inferiore a 30; nel primo caso si potrebbe uscire, altrimenti no. Il processo di classificazione funziona in modo analogo nel caso di tempo piovoso (rain).



Le foglie riguardano le classificazioni finali delle interrogazioni mentre i nodi intermedi sono interrogazioni vere e proprie sugli attributi passati.
Notiamo che il pensiero che c'è dietro questi alberi è che esista un ragionamento alla base di ciò, quello che faccio è descriverlo in modo formale.
Nel caso non ci fosse, sarebbe comodo porterlo creare a partire dagli esempi per inferirlo in modo automatico.
Immaginiamo di voler costruire (magari ottenendolo dai dati) un albero che mi possa dire se un eroe/eroina sia buono/a o cattivo/a.
Un metodo per farlo è far partire la radice con la domanda "più interessante", di solito su un solo attributo.
Come faccio a capire qual'è il tipo di domanda migliore da fare alla radice?
L'idea è quella di trovare delle domande che mi permettano di dividere equamente i campioni da analizzare.
In realtà non considero solo quello, ma considero anche come si comporta l'omogeneità nei campioni diversi trovati, in modo da avere sempre sottogruppi omogenei.
In linea di principio quindi considero la media delle eterogeneità, nel caso sia alta mi trovo in una divisione non molto utile, in caso contrario ho una buona divisione omogenea.
Quindi la domanda migliore diventerà quella con la media di eterogeneità meno elevata.
Quando in una risposta ho un omogeneità (circa) massima ottengo un caso limite al quale posso accostare una foglia per la risposta.

Ritorniamo all'esempio precedente, non ho una colonna del dataset per indicare se i supereroi siano buoni o cattivi.
Creiamo quindi un mini dataset sulla falsariga di quello usato fino ad oggi con una colonna in più che indica la suddetta correlazione.

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

heroes = pd.read_csv('heroes.csv',
sep =';', index_col = 0)

good_guys = heroes.loc[['Wonder
Woman',
                        'Aquaman',
                        'Cyborg',
                        'Flash II']]
bad_guys = heroes.loc[['Black
Manta',
                       'Penguin',
                       'Joker',
                       'Deathstroke',
                       'Bizarro']]
all_guys = pd.concat([good_guys,
bad_guys])

features = ['Height', 'Weight',
'Gender', 'First appearance',
            'Hair color', 'Eye color',
'Strength', 'Intelligence']
X = all_guys[features]

print(X)
```

Name	Height	Weight	Gender	First appearance	Hair color	Eye color	Strength	Intelligence
Wonder Woman	183.13	74.74	F	1941.0	Black	Blue	100.0	high
Aquaman	185.71	146.96	M	1941.0	Blond	Blue	85.0	high
Cyborg	198.12	173.81	M	1980.0	Black	Brown	55.0	good
Flash II	183.41	88.32	M	1956.0	Blond	Blue	50.0	high
Black Manta	188.12	92.78	M	1967.0	No Hair	Black	30.0	good
Penguin	157.89	79.13	M	1941.0	Black	Blue	10.0	good
Joker	196.07	86.91	M	1940.0	Green	Green	10.0	high
Deathstroke	193.87	101.98	M	1980.0	White	Blue	30.0	good
Bizarro	191.00	155.57	M	1958.0	Black	Black	95.0	moderate

Adesso creiamo un colonna per capire quali supereroi sono buoni o cattivi:

```
Y =
pd.concat([pd.DataFrame(['good
guy'] * len(good_guys),

index=good_guys.index),
           pd.DataFrame(['bad guy']
* len(bad_guys),

index=bad_guys.index)])

print(Y)
```

```
Name
Wonder Woman   good guy
Aquaman        good guy
Cyborg         good guy
Flash II       good guy
Black Manta     bad guy
Penguin         bad guy
Joker           bad guy
Deathstroke     bad guy
Bizarro         bad guy
```

Adesso proviamo a cercare la prima condizione da inserire alla radice, ad esempio ‘Strength’ <= 40:

print(Y[X['Strength'] <= 40])	Name Black Manta bad guy Penguin bad guy Joker bad guy Deathstroke bad guy

Il risultato è molto interessante in quanto le etichette sono tutte uguali, e quindi l'eventuale nodo successivo nell'albero sarebbe una foglia che etichetta i casi come "bad_guy". Le cose cambiano, sebbene poco, se consideriamo le osservazioni che non soddisfano la condizione:

print(Y[X['Strength'] > 40])	Name Wonder Woman good guy Aquaman good guy Cyborg good guy Flash II good guy Bizarro bad guy

In questo caso vediamo che abbiamo una sola etichetta "bad_guy". Possiamo calcolare quindi anche l'indice di Gini per le seguenti suddivisioni utilizzando:

```
def gini_2_val(f):
    return 1 - f**2 - (1-f)**2

def gini(series):
    return 1 - (sum(series.value_counts(normalize=True)
                .map(lambda f: f**2)))
```

freq = Y[X['Strength'] <= 40][0].value_counts(normalize=True) freq_bad = freq['bad guy'] gini_left = gini_2_val(freq_bad)	Output: 0.0 0.31999999999999984

print(gini_left) freq = Y[X['Strength'] > 40][0].value_counts(normalize=True) freq_bad = freq['bad guy'] gini_right = gini_2_val(freq_bad) print(gini_right)	

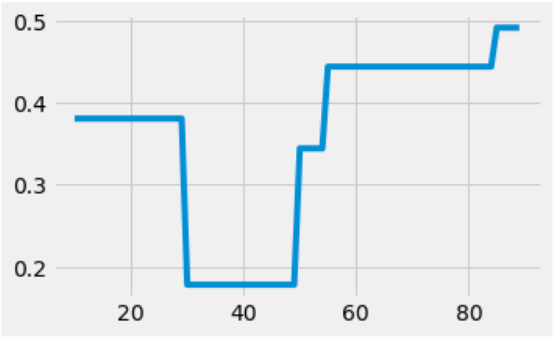
Possiamo anche fare una media pesata per numerosità di questi due indici:

weight_left = len(Y[X['Strength'] <= 40]) / len(Y) weight_right = len(Y[X['Strength'] > 40]) / len(Y) print(gini_left * weight_left + gini_right * weight_right)	Output: 0.1777777777777777

Avendo visto adesso come svolgerlo in un singolo caso potrei generalizzare creando una funzione apposita per controllare come cambia l'eterogeneità al variare di una serie di valori:

```
def split_value(attribute, value, index):
    freq = (Y[X[attribute] <= value])[0].value_counts(normalize=True)
    freq_bad = freq['bad guy']
    index_left = index(freq_bad)
    weight_left = len(Y[X[attribute] <= value]) / len(Y)
    freq = (Y[X[attribute] > value])[0].value_counts(normalize=True)
    freq_bad = freq['bad guy']
    index_right = index(freq_bad)
    weight_right = len(Y[X[attribute] > value]) / len(Y)
    return index_left * weight_left + index_right * weight_right
```

```
x_vals = range(10, 90)

plt.plot(x_vals,
        list(map(lambda v:
split_value('Strength', v, gini_2_val),
                x_vals)))
plt.show()
```



Notiamo quindi come il valore di eterogeneità è minimizzato per 'Strenght' che va dai 30 ai 50.
Per continuare la costruzione dell'albero di decisione dovremmo continuare a trovare divisioni con minima eterogeneità media.
Per farlo in automatico però possiamo utilizzare una libreria esterna "sklearn" per la trasformazione degli attributi non numerici e per la successiva creazione dell'albero di decisione.

```
from sklearn.preprocessing
import LabelEncoder

gender_encoder = LabelEncoder()
gender_encoder.fit(all_guys['Gende
r'])

eye_col_encoder = LabelEncoder()
eye_col_encoder.fit(all_guys['Eye
color'])

hair_col_encoder = LabelEncoder()
hair_col_encoder.fit(all_guys['Hair
color'])

intelligence_encoder =
LabelEncoder()
_ =
intelligence_encoder.fit(all_guys['In
telligence'])

all_guys['Gender'] =
gender_encoder.transform(all_guy
s['Gender'])
all_guys['Eye color'] =
eye_col_encoder.transform(all_guy
s['Eye color'])
all_guys['Hair color'] =
hair_col_encoder.transform(all_guy
```

Name	Height	Weight	Gender	First appearance	Hair color	Eye color	Strength	Intelligence
Wonder Woman	183.13	74.74	0	1941.0	0	1	100.0	1
Aquaman	185.71	146.96	1	1941.0	1	1	85.0	1
Cyborg	198.12	173.81	1	1980.0	0	2	55.0	0
Flash II	183.41	88.32	1	1956.0	1	1	50.0	1
Black Manta	188.12	92.78	1	1967.0	3	0	30.0	0
Penguin	157.89	79.13	1	1941.0	0	1	10.0	0
Joker	196.07	86.91	1	1940.0	2	3	10.0	1
Deathstroke	193.87	101.98	1	1980.0	4	1	30.0	0
Bizarro	191.00	155.57	1	1958.0	0	0	95.0	2

```
s['Hair color'])
all_guys['Intelligence'] =
intelligence_encoder.transform(all
_guys['Intelligence'])

X = all_guys[features]
print(X)
```

Questo nuovo data frame contenente solo valori numerici sarà poi utile per utilizzare un oggetto della classe "DecisionTreeClassifier" per costruire l'albero di decisione, passando al metodo fit i data frame che descrivono rispettivamente i supereroi e le loro etichette.

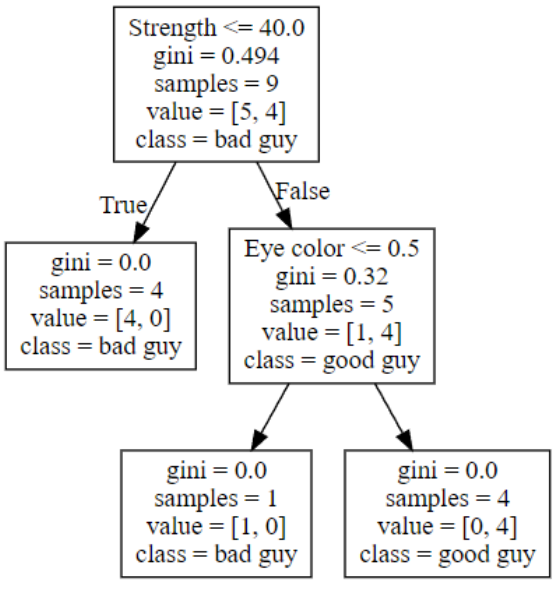
```
from sklearn import tree

clf = tree.DecisionTreeClassifier()
clf = clf.fit(X, Y)

predictions =
clf.predict([X.loc[name] for name
in X.index])
print(predictions)
```

Output:

```
array(['good guy', 'good guy', 'good
guy', 'good guy', 'bad guy',
       'bad guy', 'bad guy', 'bad guy',
'bad guy'], dtype=object)
```

Ed infine possiamo rappresentare l'albero graficamente

```
import graphviz

graphviz.Source(tree.export_grap
hviz(clf, out_file=None,

class_names=['bad guy', 'good
guy'],

feature_names=features))
```



Il fatto di inferire questo modello dai dati non ci assicura che poi possa funzionare anche per altri valori diversi da quelli usati per inferire il risultato.
Possiamo però controllare l'accuratezza provando il modello su altri esempi esterni dei quali sappiamo il risultato.

L'albero di decisione trovato si chiama "classificatore". In generale, quando ottengo un meccanismo automatico di riconoscimento di appartenenza ad una classe ho un classificatore.
Posso calcolare la sua accuratezza come già citato, stando attenti a non considerare il fatto che possano esistere classificazioni diversificate in proporzioni (ad esempio malati terminali all'1% sani al 99%).
Inoltre dobbiamo pesare quale classificazione finale sia meno grave nel caso in cui sia errata.
Metodo per classificare l'accuratezza di un classificatore su dati e classificazioni sbilanciate.
In generale tutto parte da una "matrice di confusione" (confusion matrix).

Matrice di confusione
Matrice $M_{n \times n}$ con come n numero di classificazioni possibili.
Se inseriamo come righe e colonne (intercambiabili di posizione ma non di significato) i valori esatti e le predizioni del classificatore ottengo:

CONFUSUSION MATRIX	ACTUAL	
PREDICTED	True Positive (TP)	False Positive (FP)
	False Negative (FN)	True Negative (TN)

Nella matrice di confusione inserisco poi nelle entrate le frequenze assolute dei valori vero positivi, vero negativi, falso positivi e falso negativi.
Sommando le classi (valori reali) ottengo le frequenze dei valori positivi e negativi.
Abbiamo quindi il nostro primo indice:

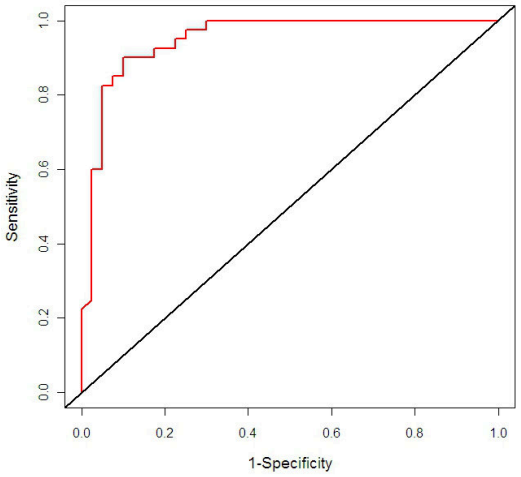
- Sensibilità $\frac{TP}{TP+FN}$

Percentuale dei casi positivi sui quali il classificatore non ha sbagliato.
Secondo indice:

- Specificità $\frac{TN}{TN+FP}$

Percentuale dei casi negativi sui quali il classificatore non ha sbagliato.

Immaginiamo adesso di avere un piano cartesiano con come ascisse 1-specificità e come ordinata la sensibilità.



Nel caso di classificatore completamente positivo la sensibilità è massima dato che tutti i positivi vengono riconosciuti e la specificità è minima dato che i negativi vengono tutti considerati positivi.
Classificatore costante positivo CP
Punto sul grafico: (1, 1)

Classificatore costante negativo CN
Punto sul grafico: (0, 0)

Il caso ideale sarebbe un caso in cui il classificatore non sbaglia mai, quindi con sensibilità e specificità uguali a 1.
Punto sul grafico: (0, 1)

Infine il classificatore che sbaglia sempre ha coordinate (1, 0).

Immaginiamo di poter tracciare ora una diagonale tra CP e CN.
Invece immaginiamo adesso di avere un classificatore randomico con $1/2$ di possibilità di sbagliare.
Avrei un punto (½, ½).

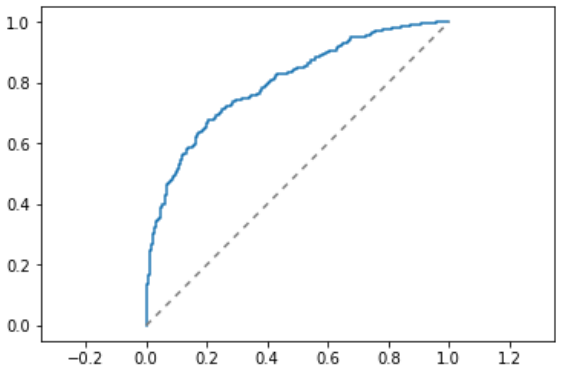
Se il classificatore fosse randomico e truccato per dare ⅘ delle volte un positivo avrei un punto (⅘ , ⅘).

La traiettoria diagonale ci indica, in base alla vicinanza rispetto a CP o CN, quanto il classificatore sia favorevole ai positivi o ai negativi.

Classificatori a soglia
Classificatori di un generico oggetto calcolando una quantità e verificando poi che quest'ultima sia superiore a una soglia prefissata.

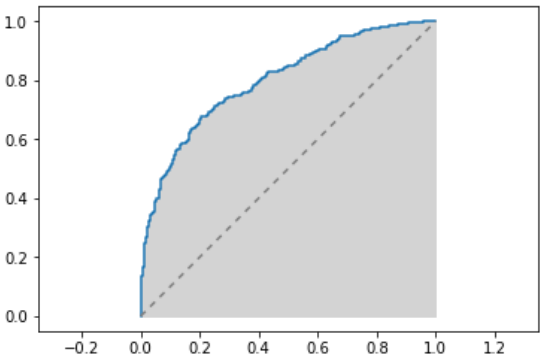
La curva ottenuta dalla serie sui classificatori a soglia si chiama curva ROC.
L'area sottostante invece si chiama AROC.
Curva ROC in python:

```
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression

X, y = make_classification(n_samples=10000, n_features=10, n_classes=2, n_informative=5)
Xtrain = X[:9000]
Xtest = X[9000:]
ytrain = y[:9000]
ytest = y[9000:]

clf = LogisticRegression()
clf.fit(Xtrain, ytrain)

from sklearn import metrics

preds = clf.predict_proba(Xtest)[:,1]
fpr, tpr, _ = metrics.roc_curve(ytest, preds)

plt.plot(fpr, tpr)
plt.plot([0, 1], [0, 1], dashes=[3, 3], color='gray')
plt.xlim([-0.01, 1])
plt.ylim([0, 1.01])
plt.axis('equal')
plt.show()
```



AROC in python:

```
plt.fill_between(fpr, [0]*len(tpr), tpr,
color='lightgray')
plt.plot(fpr, tpr)
plt.plot([0, 1], [0, 1], dashes=[3, 3],
color='gray')
plt.xlim([0, 1])
plt.ylim([0, 1.1])
plt.axis('equal')
plt.show()
```



Ultimo argomento della statistica descrittiva
Mettiamo di avere un campione di persone alla quale somministrare o far finta di somministrare un farmaco per controllarne gli effetti.

Anova
Analysis of Variance
Su un totale di n osservazioni ho G gruppi di ampiezza m_1 , m_2 , … , m_G dove m_1 sono il numero di osservazioni del primo gruppo.

$$\sum_{i=i}^{G} m_i = n$$

Notazione:

$x_i^{\,g} = i$-esimo elemento nel g-esimo gruppo

$$\bar{x}^{g} = \frac{1}{mg}\sum_{i=1}^{mg} x_i^{\,g}$$

$$\bar{x} = \frac{1}{n}\sum_{j=1}^{G}\sum_{i=1}^{mj} x_i^{\,j}$$

SS_T = Sum of squares
Somma di tutti gli scarti quadratici nella media campionaria

$$SS_T = \sum_{j=1}^{G}\sum_{i=1}^{mj} (x_i^{\,j} - \bar{x})^2$$

Dalla quale:

$$s^2 = \frac{SS_T}{n-1}$$

SS_W = Varianza entro i gruppi

$$SS_W = \sum_{j=1}^{G}\sum_{i=1}^{mj} (x_i^{\,j} - \bar{x}^{j})^2$$

Dalla quale:

$s^2 = \frac{SS_W}{n-G}$

SS_B = Varianza tra i gruppi

$SS_B = \sum_{g=1}^{G} mg(\overline{x}^g - \overline{x})^2$

Dalla quale:

$s^2 = \frac{SS_B}{G-1}$

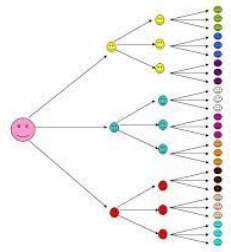
$SS_T = SS_W + SS_B$

Calcolo combinatorio
Studio in quanti modi diversi è possibile formare raggruppamenti di oggetti, dipendenti da molte caratteristiche:

- Ordine, se conta o meno
- Se si possono ripetere o meno
- ...

In funzione di questi fattori possiamo avere diverse permutazioni.

Principio fondamentale del calcolo combinatorio
Se noi distinguiamo le scelte possibili e elenchiamo successivamente il numero di azioni possibili, moltiplicando il numero di scelte per questo numero di azioni ottengo il numero di permutazioni possibili.



Primo caso: Permutazioni
Avendo
a_1, a_2, ... , a_n Scelte

Una permutazione di n oggetti è una possibile sequenza di questi oggetti.
C'è un modo di calcolare il numero di permutazioni possibili con n elementi?
Immaginiamo di avere un certo numero di posti per le nostre n scelte.
In prima posizione ho n scelte, nel secondo $n - 1$, nel terzo $n - 2$ e così via.

P_n = numero di permutazioni = $\prod_i^n i = n!$

E se gli oggetti non fossero tutti diversi?
In questi casi, avendo un multi insieme di oggetti distinguibili a gruppi abbiamo permutazioni distinguibili.
Mettiamo di avere 5 oggetti divisi in due gruppi, uno da 2 elementi e uno da 3.
$\frac{5!}{2!3!}$
Coefficiente multinomiale

Coefficiente binomiale: $\binom{N}{R} := \frac{N!}{(N-R)!R!}$

Coefficiente multinomiale: $\binom{N}{R_1;R_2;\ldots;R_k} := \frac{N!}{R_1!R_2!\cdots R_k!}$

Disposizioni
Abbiamo una serie di oggetti da mettere in sequenza, in questo caso però non devo selezionare tutti gli oggetti ma un suo sottoinsieme.
Abbiamo 2 tipi di disposizioni, con ripetizioni e senza.
Oggetti $a_1, a_2, \ldots, a_n$
$k \leq n$ con k dimensione

Disposizioni senza ripetizione
$D_{n,k} = \frac{n!}{(n-k)!}$

Disposizioni con ripetizioni
$d_{n,k} = n^k$

Combinazioni
Sono le disposizioni in cui non conta l'ordine.
$C_{n,k} = (n \quad k) = \frac{n!}{(n-k)!k!}$
Coefficiente binomiale

Lezione del 28 Marzo 2023

Lezione 9

Nella scorsa lezione abbiamo introdotto il calcolo combinatorio.
Esso servirà per calcolare la probabilità di alcuni eventi.

La probabilità è il nostro strumento per lavorare con l'incertezza nella casualità degli scenari possibili.
Di solito la probabilità si descrive tramite l'interpretazione del fenomeno.
Ad esempio, con un approccio soggettivista dipendente dal soggetto osservante.
Un altro approccio è quello frequentista, il quale deduce la frequenza di eventi futuri da osservazioni del fenomeno passate.
Si può anche definire la frequenza assoluta di un esito con il limite del numero di osservazioni di un evento che tende all'infinito.

Capiamo come possiamo usare la matematica per modellare la probabilità.
La base è un evento casuale con un insieme degli esiti possibili.
Definiamo:
Ω = Insieme degli esiti, spazio campionario, insieme universo

Esempio:
Lancio del dado
$\Omega = \{1, 2, 3, 4, 5, 6\}$

Nella definizione non cambia se il dado sia bilanciato o meno.
L'insieme degli esiti può anche essere infinito (o continuo).

Esempio:
Peso di un supereroe scelto a caso
$\Omega = \Re^{+}$

Definizione:
$e \in \Omega$
Viene chiamato esito (o evento elementare)

Formalmente un evento è un sottoinsieme (solitamente insieme singoletto) dell'insieme degli esiti.
$E \subseteq \Omega$

E' un sottoinsieme perchè potrei non essere interessato ad un solo elemento.

Esempio:
Dado in cui cerco l'evento in cui il numero sia pari.

Definizione
Evento = qualcosa che capita o non capita

$E = \{2, 4, 6\}$

Abbiamo definito tutto tramite insiemi, quindi abbiamo già la teoria da applicarvi.

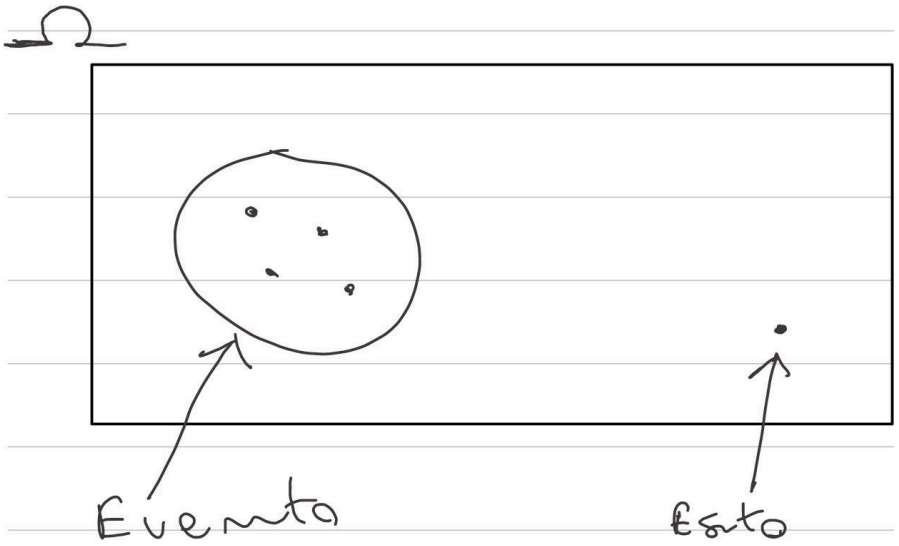


Diagramma di Eulero Venn

Esempi di eventi/insiemi
$\forall E, F$
$e \in E \cup F \Leftrightarrow e \in E \vee e \in F$
Per ogni coppia di eventi, un esito sta nella loro unione se e solo se è un esito del primo o del secondo evento.
Quindi uno tra i due eventi si verifica.
$e \in E \cap F \Leftrightarrow e \in E \wedge e \in F$
Un esito appartiene all'intersezione di due eventi se e solo se si verificano entrambi gli eventi.

Esempio:
Estrazione da una scatola di fumetti
$M = \{fumetti\ Marvel\}$
$D = \{fumetti\ DC\}$

$M \cap D$ = estraggo un fumetto sia Marvel, sia DC (=∅)

Evento insieme vuoto:
Evento che non si verifica mai

$e \in E/F \Leftrightarrow e \in E \wedge e \notin F$
Si verifica il primo evento ma non il secondo.
Questa operazione è asimmetrica.

$e \in \Omega / E = \overline{E}$
Evento che si ottiene quando non si verifica l'evento E.

Ω è anch'esso un evento, il quale si verifica sempre (detto evento certo).
Quindi:

- $\emptyset$ evento impossibile
- Ω evento certo

$E \subseteq F \Leftrightarrow E \rightarrow F$
Ogni esito di E è un esito di F, quindi un evento F si verifica sempre quando si verifica un evento sottoinsieme E.

$E = F$
Eventi uguali, bisogna mostrare che sono entrambi sottoinsiemi dell'altro.

Unione e intersezione sono binarie ma possono essere estese.

$$E_1 \cup E_2 \cup \ \ldots \ \cup E_n = \bigcup_{i=1}^{n} E_i$$

$$E_1 \cap E_2 \cap \ \ldots \ \cap E_n = \bigcap_{i=1}^{n} E_i$$

Queste proprietà godono di:

- Commutabilità
  - $A \cup B = B \cup A$
- Associatività
  - $A \cup (B \cup C) = (A \cup B) \cup C$
- Distributività
  - $A \cup (B \cap C) = (A \cup B) \cap (A \cup B)$
  - $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- Leggi di de Morgan
  - $\overline{A \cup B} = \overline{A} \cap \overline{B}$
  - $\overline{A \cap B} = \overline{A} \cup \overline{B}$

Dimostrazione prima legge di de Morgan
$\forall e \in \overline{A \cup B} \Leftrightarrow e \in \overline{A} \cap \overline{B}$

$e \in \overline{A \cup B} \Leftrightarrow e \notin A \cup B \Leftrightarrow e \notin A \wedge e \notin B \Leftrightarrow e \in \overline{A} \wedge e \in \overline{B} \Leftrightarrow e \in \overline{A} \cap \overline{B}$

Adesso che abbiamo modellato insiemisticamente la probabilità, dovremmo dare un valore numerico all'incertezza di probabilità di un determinato evento.
Definiamo:

$P: A \to \Re^+$

Dove:

P = Funzione di probabilità

A = Famiglia di eventi o algebra degli eventi

$A = \{E_1, E_2, \ldots\}$

$\forall i \; E_i \in \Omega$

Insieme di eventi
Insieme di insieme di esiti.
Esso deve essere chiuso rispetto al complemento, all'unione ed all'intersezione.
Questo dominio deve soddisfare un insieme di proprietà:

1) $\Omega \in A$
2) $\forall E \;\; E \in A \Rightarrow \overline{E} \in A$
3) $\forall E, F \;\; E \in A \wedge F \in A \Rightarrow E \cup F \in A$

Osservazione

$A \cup B = \overline{\overline{A \cap B}} = \overline{(\overline{A} \cup \overline{B})}$

Grazie a de Morgan la chiusura rispetto all'intersezione è implicita.

Come sarà fatta quindi un algebra degli eventi?
Se Ω è finito:

$|\Omega| = n$

$|A| = 2^n$

Insieme delle parti (insieme dei sottoinsiemi o delle partizioni)

$\Omega = \{e_1, e_2, \ldots, e_n\}$

$\forall E \;\; E \in \Omega \; (E \in 2^{\Omega})$

$E = \{e_{i1}, e_{i2}, \ldots, e_{ik}\}$

$= \{e_{i1}\} \cup \{e_{i2}\} \cup \ldots \cup \{e_{ik}\}$

Se questi insiemi singoletto appartengono ad A allora anche $E \in A$.

Si chiama algebra degli eventi perchè è definita tramite le sue proprietà algebriche.

$P: A \to \Re^+$

Definita in modo indiretto e assiomatico.

Assiomi di Kolmogorov:

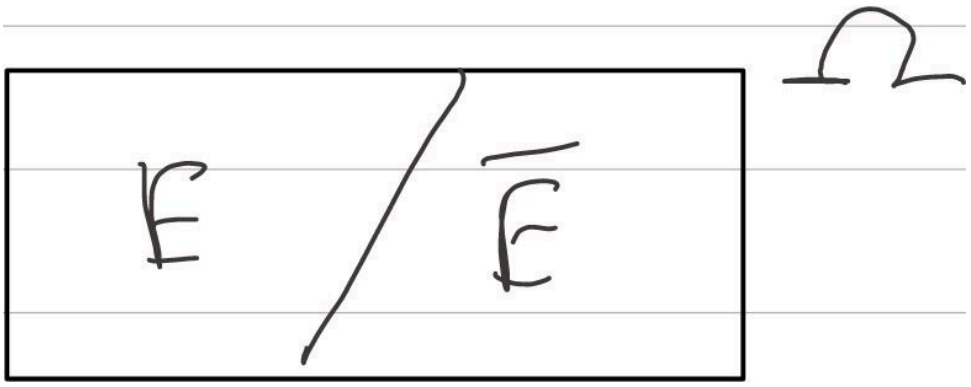
1) $\forall E \in A \;\; P(E) \geq 0$
2) $P(\Omega) = 1$
3) $\forall E, F \in A \, con \, E \cap F = \emptyset$ (due eventi senza intersezione, cioè che non si verificano insieme) $\Rightarrow P(E \cup F) = P(E) + P(F)$

Probabilità frequentista:
Limite della frequenza relativa per il numero di ripetizioni all'infinito.

Dimostrazioni di teoremi elementari di probabilità a partire dagli assiomi:

- $P(\overline{E}) = 1 - P(E)$

Per l'assioma 2, $1 = P(\Omega)$.



$E \cup \overline{E} = \Omega \qquad E \cap \overline{E} = \emptyset$
$P(\Omega) = P(E \cup \overline{E}) = P(E) + P(\overline{E})$
Quindi:
$P(\overline{E}) = P(\Omega) - P(E)$
$P(\overline{E}) = 1 - P(E)$

- Posso estendere il terzo assioma richiedendo che gli elementi siano a 2 a 2 disgiunti

- $E, F \in A \qquad E \subseteq F \qquad P(E) \leq P(F)$

$F = E \cup E' \; (\rightarrow F/E)$
$P(F) = P(E \cup E')$
$E \cap E' = \emptyset$
Per il terzo assioma:
$P(F) = P(E) + P(E')$
Per il primo assioma:
$P(E') \geq 0$
Allora
$P(E) \leq P(E) + P(E')$

$P(E) \leq P(F)$

- Per definizione, $E \subseteq \Omega$ $P(E) \leq P(\Omega)$

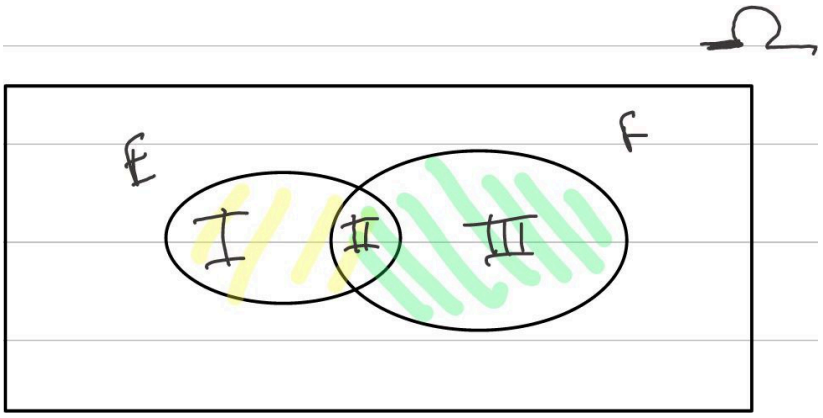
Quindi:
$0 \leq P(E) \leq 1$

Lezione del 30 Marzo 2023

Lezione 10

Durante la scorsa lezione abbiamo iniziato a parlare di probabilità.
$P(A)$
Dove P è la funzione di probabilità e A è l'algebra degli eventi.
Successivamente abbiamo visto gli assiomi di Kolmogorov e cosa comportano.

$P(E \cup F)$
Gli assiomi di Kolmogorov ci aiutano soltanto nel caso i due eventi non abbiano intersezione dato che:
$P(E \cup F) = P(E) + P(F)$ con $E \cap F = \emptyset$
Nel caso generale però:
$P(E \cup F) = P(E) + P(F) - P(E \cap F)$



$I = E \backslash F = E \cap \overline{F}$
$II = E \cap F$
$III = F \backslash E = F \cap \overline{E}$
Sono eventi a 2 a 2 disgiunti
Quindi $E \cup F = I \cup II \cup III$

$P(E \cup F) = P(I) + P(II) + P(III)$
Ricordiamo che $I \cup II = E$ e $II \cup III = F$

$P(E) = P(I) + P(II)$
$P(F) = P(II) + P(III)$
$P(E \cup F) = P(I) + P(II)[= P(E)] + P(III)$
$P(E \cup F) = P(E) + P(III) + P(II)[= P(F)] - P(II)$
$P(E \cup F) = P(E) + P(F) - P(II)[= P(E \cap F)]$

Esempio sulle sigarette:
E: maschi americani che fumano sigarette $= 0.28$
F: maschi americani che fumano sigari $= 0.07$
$E \cap F$: maschi americani che fumano entrambe $= 0.05$

Queste sono le frequenze assolute, che nella maggior parte dei casi, nello stesso fenomeno, riflettono la loro probabilità.

Qual'è la probabilità che scegliendo un americano a caso esso non fumi?
$P(non\ fumi) = 1 - P(fumi) = 1 - P(E \cup F) = P(E) + P(F) + P(E \cap F)$
$= 1 - (0.28 + 0.05 - 0.07) = 1 - (0.3) = 0.7$

Modelli per le probabilità:
Molto spesso abbiamo situazioni della realtà che possono essere ricondotte a modelli della probabilità "parametrici".

Esempio:
Lancio della moneta, di un dado, giro di una roulette, l'estrazione della tombola, ...
- ★ Esiti equiprobabili

Si parla quindi di spazi degli esiti equiprobabili.
$\Omega = \{l_1, l_2, \dots, l_N\}$
Dove N è il numero degli esiti o "Parametro".

Equiprobabilità:
$\exists p \in \Re^+ \qquad P(\{e_i\}) = p$
$\{e_i\}$ evento singoletto che contiene uno ed uno solo degli esiti possibili

Sappiamo che:
$1 = P(\Omega)$
Possiamo vedere omega come unione degli insiemi singoletto di tutti gli indici
$= P(\bigcup_{i=1}^{N} \{e_i\}) = \sum_{i=1}^{N} P(\{e_i\}) = \sum_{i=i}^{N} p = Np$

Quindi $p = \frac{1}{N}$

Se ho degli spazi campionari con esiti equiprobabili la dimensione degli esiti deve essere finita.

$E \in A$
$E = \{e_{i1}, e_{i2}, \dots, e_{ik}\}$
$P(E) = P(\bigcup_{j=1}^{k} \{e_{ij}\}) = \sum_{j=i}^{N} P(\{e_{ij}\}) = kp = \frac{k}{N}$

Esempio
Un'urna contiene 6 biglie bianche e 5 nere. Ne estraggo 2 senza reimmissione. Qual'è la probabilità di tirare 2 biglie di colore diverso?
6B 5N
Compio 2 estrazioni
$P(due\ colori\ diversi)$
L'esito è l'estrazione di 2 biglie, senza reimmissione (ripetizioni), quindi le disposizioni senza ripetizioni di 11 oggetti in 2 posti $= 11 \cdot 10 = 110 = N$
Casi favorevoli:

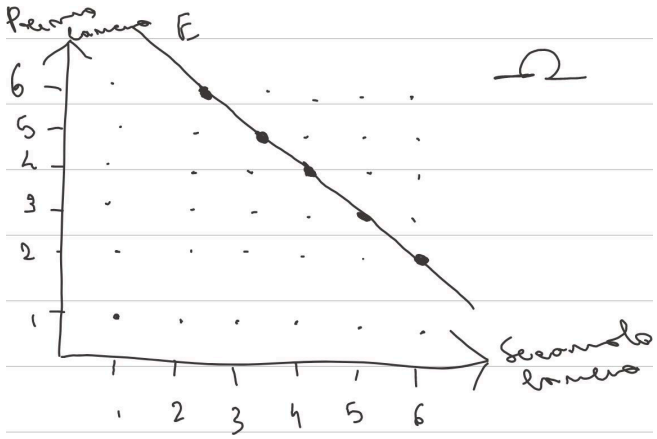
- Prima bianca, seconda nera
  - 30 casi
- Prima nera, seconda bianca
  - 30 casi

$P = \frac{60}{110} = \frac{6}{11}$

Altro esempio:
Una commissione di 5 persone deve essere scelta casualmente tra 6 uomini e 9 donne.
Qual'è la possibilità che vengono scelti 3 uomini e 2 donne?
$6U + 9D = 15TOT$
$P(3U + 2D) = \frac{commissioni\ favorevoli}{numero\ totale\ di\ commissioni}$
Combinazioni, sottoinsiemi, coefficiente binomiale $(11 \quad 5)$
Casi favorevoli:
$(6\ 3)(9\ 2)$
Probabilità:
$= \frac{(6\ 3)(9\ 2)}{(11\ 5)} = \frac{720}{3003} \approx 0.2397\dots$

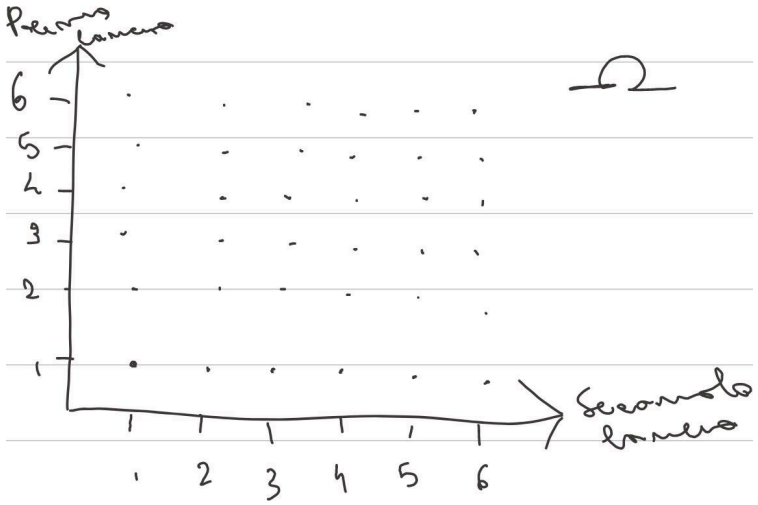
Altro esempio:
Probabilità che in una stanza di N persone ci siano 2 persone nate lo stesso giorno.
$\frac{(365 \cdot 364 \cdot 363 \cdot \dots \cdot 365 - N + 1)}{365^N} = \frac{365!}{N! \cdot 365^N}$

Altro esempio:
Lanciando due dadi bilanciati qual'è la probabilità che la somma degli esiti dei due dadi sia 8?
L'esito è una coppia di numeri tra 1 e 6.

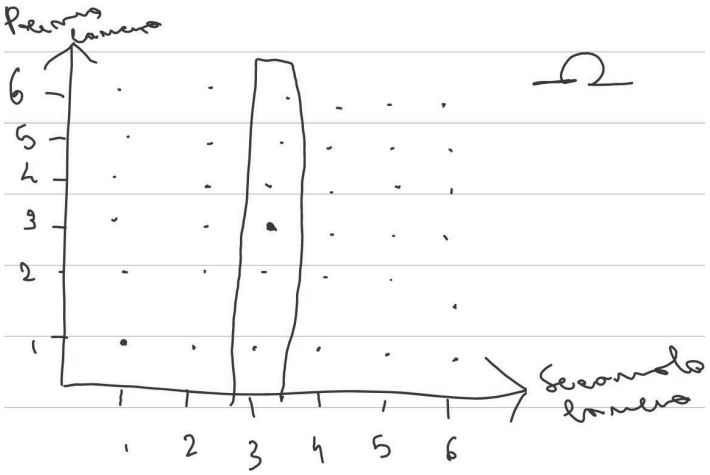


$P = \frac{5}{36}$

Riprendendo il caso precedente:



Potremmo avere dei casi con incertezza parziale la quale ci permette di eliminare alcuni casi.
Avendo per esempio come informazione la faccia d'un dado, la probabilità per raggiungere un certo numero diventa quindi $\frac{1}{6}$.
Quindi le informazioni parziali rimuovono dei casi possibili e quindi elementi da Ω.
Per esempio, se il primo dado avesse avuto come risultato 3:



Questa si chiama probabilità condizionata.

Dati $E,\ F \in A$ la probabilità di E dato F, indicata con $P(E|F) = \frac{P(E \cap F)}{P(F)}$
F si dice evento condizionante mentre E si dice evento condizionato.

Quindi dopo l'esito di F, F diventa Ω' e E diventa $E \cap F$.

E = somma uguale a 8
F = primo lancio del dado uguale a 3
$P(E|F) = \frac{1/36}{1/6} = 1/6$

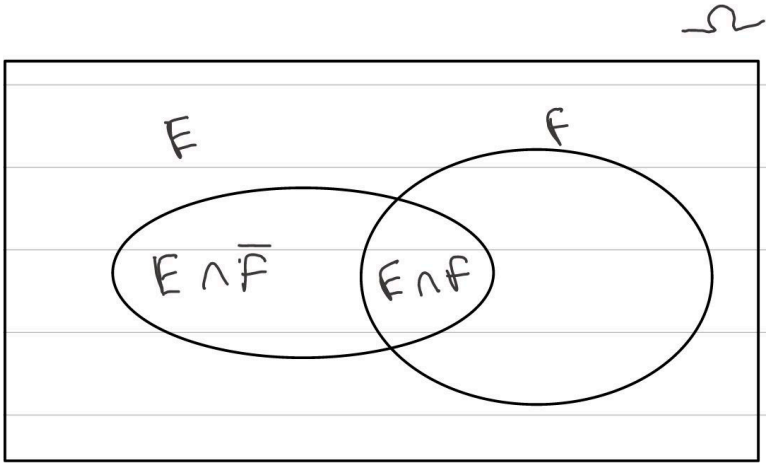
Esempio:
Una confezione contiene 5 pennarelli guasti, 10 difettosi e 25 funzionanti.
Prendendone 1, esso scrive, qual'è la probabilità che esso continui a scrivere?
5G 10D 25F
40TOT
$P(F|\overline{G}) = \frac{P(F \cap \overline{G})}{P(\overline{G})} = \frac{25/40}{1-5/40} = \frac{25}{35}$
$\overline{G} = \{pennarelli\ funzionanti\ o\ difettosi\ (non\ guasti)\}$
$F = \{funzionanti\}$
$= 5/7$

Potevamo farlo anche trasformando lo spazio:
10D 25F
35TOT

Altro esempio:
Un'azienda organizza una cena per una festa per padri con figli maschi.
Qual'è la probabilità che uno degli impiegati abbia 2 figli maschi?
$\Omega = \{MF,\ FM,\ MM,\ FF\}$

Sappiamo che il signor Jones partecipa con suo figlio maschio.
$A = \{MF, FM, MM\}$
$B = \{MM\} = B \cap A$
$P(B|A) = \frac{P(B \cap A)}{P(A)} = \frac{1/4}{3/4} = 1/3$

Teorema delle probabilità totali



Mettiamo di non poter calcolare la probabilità di E, ci sono casi in cui la probabilità che si verifichi o meno F ci aiuti a calcolarlo.
$(E \cap \overline{F}) \cup (E \cap F) = E \cap (F \cup \overline{F}) = E \cap \Omega = E$

E se le intersechiamo?
$(E \cap \overline{F}) \cap (E \cap F) = \emptyset$

Quindi per Kolmogorov
$P(E) = P(E \cap F) + P(E \cap \overline{F})$

Guardiamo adesso la formula della probabilità condizionata:
$P(B|A) = \frac{P(B \cap A)}{P(A)}$
$P(B \cap A) = P(B|A)P(A) = P(A|B)P(B)$

Applichiamo il risultato alla formula precedente.
$P(E) = P(E \cap F) + P(E \cap \overline{F}) = P(E|\overline{F})P(\overline{F}) + P(E|F)P(F)$

Esempio:
Il 30% dei clienti è incline agli incidenti (0.4 di probabilità di incidente per cliente) mentre il resto è meno incline (0.2 di probabilità di incidente).

$I = \{cliente\ incline\ agli\ incidenti\}$
$C = \{cliente\ che\ farà\ un\ incidente\ il\ prossimo\ anno\}$

$P(C)$?
Sappiamo che

$P(I) = 0.3 \qquad P(\bar{I}) = 0.7$

$P(C|I) = 0.4 \qquad P(C|\bar{I}) = 0.2$

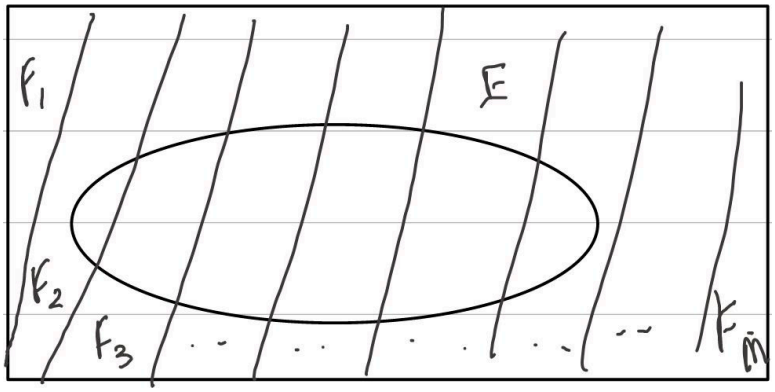
$P(C) = P(C|I)P(I) + P(C|\bar{I})P(\bar{I}) = 0.4 \cdot 0.3 + 0.2 \cdot 0.7 = 0.12 + 0.14 = 0.26$

Altro esempio:
Viene eseguito un sondaggio il quale contiene una domanda delicata, essa però viene sottoposta soltanto ad una parte dei partecipanti al test. Qual'è la probabilità che venga risposto di si alla domanda delicata?

$P(SI) = P(SI|Domanda\ delicata)P(Domanda\ delicata) + P(SI|Domanda\ di\ controllo)$
$P(Domanda\ di\ controllo)$

$P(SI|Domanda\ delicata) = \frac{P(SI) - P(SI|Domanda\ di\ controllo)P(Domanda\ di\ controllo)}{P(Domanda\ delicata)}$

Mettiamo di non avere un singolo evento F ma una partizione di Ω.



$\bigcup_{i=1}^{n} F_i = \Omega$

$\forall i, j\ F_i \cap F_j = \emptyset$

$P(E) = \sum_{i=1}^{n} P(E|F_i)P(F_i)$

Dimostrazione:

$E = \bigcup_{i=1}^{n} (E \cap F_i)$

$\forall i, j\ (E \cap F_i) \cap (E \cap F_i) = \emptyset$

$= E \cap \bigcup_{i=1}^{n} F_i = E \cap \Omega = E$

E quindi:

$$P(E) = \sum_{i=1}^{n} P(E \cap F_i)$$

$$= \sum_{i=1}^{n} P(E|F_i)P(F_i)$$

Lezione del 4 Aprile 2023

Lezione 11

Nell'ultima lezione abbiamo visto il teorema delle probabilità totali. Avendo quindi una partizione di Ω spazio campionario, cioè $\{F_1, \dots, F_n\}$ partizione di n eventi:

$$P(E) = \sum_{i=1}^{n} P(E|F_i)P(F_i)$$

Esempio:
Mettiamo di avere 3 macchinari con diverse probabilità di produrre pezzi difettosi.

	Pezzo difettoso	Produzione
A	2%	60%
B	3%	30%
C	4%	10%

Qual'è la probabilità che un pezzo scelto a caso tra quelli prodotti sia difettoso?
A = pezzo prodotto dal macchinario A
B = pezzo prodotto dal macchinario B
C = pezzo prodotto dal macchinario C
D = pezzo difettoso

$P(D|A) = 0.02$ $\quad$ $P(A) = 0.6$
$P(D|B) = 0.03$ $\quad$ $P(B) = 0.3$
$P(D|C) = 0.04$ $\quad$ $P(C) = 0.1$

$$P(D) = P(D|A)P(A) + P(D|B)P(B) + P(D|C)P(C)$$
$$= 0.012 + 0.009 + 0.004 = 0.025$$

E' come se fosse una media pesata delle probabilità condizionate.

Immaginiamo di avere eseguito un test medico.
Mettiamo che questo test sia errato nell'1% dei casi, quindi una persona malata verrà rilevata correttamente il 99% dei casi.

M = malato
P = esito positivo
$P(P|M)$ = probabilità che il test sia positivo su un paziente malato
$P(M|P)$ = probabilità che prendendo una persona con risultato positivo essa sia malata
$P(P|M) = 0.99$

Ci servono altre informazioni per risalire all'altra probabilità:

- In che modo si sbaglia il test

$P(P|\overline{M})$ = probabilità che una persona non malata risulti positiva al test $= 0.01$

- La probabilità che una persona sia malata

$P(M)$ = diffusione della malattia $= 0.005$

$$P(M|P) = \frac{P(M \cap P)}{P(P)}$$

$$P(M \cap P) = P(P)P(M|P) = P(M)P(P|M) = P(P \cap M)$$
(Intersezione simmetrica)

$$P(P) = P(P|M)P(M) + P(P|\overline{M})P(\overline{M})$$

$$P(M|P) = \frac{P(P|M)P(M)}{P(P|M)P(M)+P(P|\overline{M})P(\overline{M})} = \frac{0.99 \cdot 0.005}{0.99 \cdot 0.005 + 0.01 \cdot 0.995} \approx 0.3322$$

Quindi in meno di un terzo dei casi, quando ho un risultato positivo la persona è effettivamente malata.

Esempio numerico:
Popolazione di 200 individui:

- 1 malato = 0.99positivi
- 199 sani = 1.99 negativi (0.01 falsi positivi)

Teorema di Bayes
Stesse premesse del teorema delle probabilità totali:

- Partizione di $\Omega = \{F_1, \dots, F_n\}$ eventi

$$P(F_j|E) = \frac{P(E \cap F_j)}{P(E)} = \frac{P(E|F_j)P(F_j)}{\sum_{i=1}^{n} P(E|F_i)P(F_i)}$$

Esercizio:

Abbiamo un investigatore privato ed un sospettato.
Egli è convinto al 60% della colpevolezza del sospettato (probabilità soggettivistica).
Si scopre che il criminale deve essere mancino, il sospettato è mancino.
La convinzione dell'investigatore aumenta?

C = il sospettato è colpevole
M = il colpevole è mancino

$P(C) = 0.6 \qquad P(M) = 0.2$
$P(M|C)$ = probabilità che sia mancino il colpevole = 1

$P(C|M) = \frac{P(C \cap M)}{P(M)} = 0.68$

Altro esempio:
E' caduto un aereo e devo cercarlo tra 3 zone diverse.
All'inizio non ho motivo di privilegiare alcuna zona rispetto alle altre.
$\forall i = 1, 2, 3$ R_i = evento che si verifica se l'aereo è atterrato nella zona i
$P(R_i) = 1/3$
α_i = probabilità di non trovare l'aereo se lo si cerca nella zona i
E = evento che si verifica quando cercando nella zona 1 non trovo l'aereo
$P(E|R_2) = 1$
$P(E|R_3) = 1$
$P(E|R_1) = \alpha_1$
$P(E) = P(E|R_1)P(R_1) + P(E|R_2)P(R_2) + P(E|R_3)P(R_3) =$
$\frac{\alpha_1}{3} + 1/3 + 1/3 = \frac{\alpha_1+2}{3}$

$P(R_1|E) = \frac{P(E|R_1)P(R_1)}{P(E)} = \frac{\alpha_1/3}{\alpha_1+2/3} = \frac{\alpha_1}{\alpha_1+2}$
$P(R_2|E) = \frac{P(E|R_2)P(R_2)}{P(E)} = \frac{1/3}{\alpha_1+2/3} = \frac{1}{\alpha_1+2}$
$P(R_3|E) = \frac{P(E|R_3)P(R_3)}{P(E)} = \frac{1/3}{\alpha_1+2/3} = \frac{1}{\alpha_1+2}$

$P(R_1|E) + P(R_2|E) + P(R_3|E) = 1$

Immaginiamo adesso di ottenere come risultato:
- $P(E|F) = P(E)$

Il fatto che il verificarsi di un evento non ci fornisca nessuna informazione su un altro evento ci dice che essi sono indipendenti.

$(P(E|F) =) \frac{P(E \cap F)}{P(F)} = P(E)$

$P(E \cap F) = P(E)P(F)$

Ovviamente questa cosa è simmetrica:

$(P(F|E) =) \frac{P(F \cap E)}{P(E)} = P(F)$

$P(E \cap F) = P(E)P(F)$

Quindi la nostra relazione è prevalentemente simmetrica.

Alcuni esempi:

Ho un mazzo da 52 carte ben mescolato e sono interessato a:

- Evento che si verifica quando pesco un asso (A)
- Evento che si verifica quando pesco una carta di cuori (C)

La pescata rappresenta uno spazio equiprobabile.

$P(A) = 4/52 = 1/13$

$P(C) = 13/52 = 1/4$

$A \cap C$ = pescata di un asso di cuori

$P(A \cap C) = 1/52$

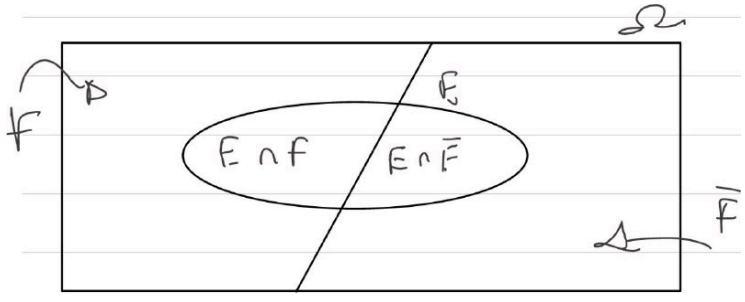
$P(A)P(C) = 1/4 \cdot 1/13 = 1/52$

I due eventi sono quindi indipendenti.

Una volta che so che due eventi sono indipendenti allora so anche che E ed $\overline{F}$ sono indipendenti.

$P(E \cap F) = P(E)P(F)$

$P(E \cap \overline{F}) = P(E)P(\overline{F})$



$E = (E \cap F) \cup (E \cap \overline{F})$

$P(E) = P(E \cap F) + P(E \cap \overline{F})$

$P(E \cap \overline{F}) = P(E) - P(E)P(F)$

$P(E \cap \overline{F}) = P(E)(1 - P(F))$

$P(E \cap \overline{F}) = P(E)P(\overline{F})$

Analogamente la situazione si ripete anche per tutti gli eventi derivanti da questi due.

Come possiamo estenderlo a più di due eventi?
Potrei provare a verificare a 2 a 2 se gli eventi siano indipendenti ma sarebbe un ragionamento fallace.
Mettiamo come esempio il lancio dei due dadi (con equiprobabilità).
E = somma dei dadi uguale a 7
F = primo dado uguale a 3
G = secondo dado uguale a 4
$P(E) = 1/6$
$P(F) = 1/6$
$P(G) = 1/6$

$P(E \cap F) = 1/36 \quad P(E \cap G) = 1/36 \quad P(F \cap G) = 1/36$
$P(E)P(F) = 1/36 \quad P(E)P(G) = 1/36 \quad P(F)P(G) = 1/36$

Questi 3 eventi sono a 2 a 2 indipendenti, però:
$P(E|F \cap G) = 1$

In realtà quello nel caso dei 3 bisogna aggiungere una sola condizione, che anche i tre eventi insieme siano indipendenti, quindi:
$P(E \cap F \cap G) = P(E)P(F)P(G)$

Anche qui risultano indipendenti eventi derivati.
Esempio:
E, F, G indipendenti $\Rightarrow E, F \cup G$ indipendenti
$P(E \cap (F \cup G)) = P(E)P(F \cup G) = P((E \cap F) \cup (E \cap G))$
$= P(E \cap F) + P(E \cap G) - P((E \cap F) \cap (E \cap G))$ $(E \cap F \cap G)$
$= P(E)P(F) + P(E)P(G) - P(E)P(F)P(G)$
$= P(E)(P(F) + P(G) - P(F)P(G))$
$= P(E)(P(F) + P(G) - P(F \cap G))$
$= P(E)P(F \cup G)$

Serie
Funziona quando funzionano tutte

Parallelo
Funziona quando almeno uno funziona

$$P(funziona\ in\ serie) = P(\bigcup_{i=1}^{n}\{i - esimo\ che\ funziona\})$$

$$= \prod_{i=1}^{n} P(i - esimo\ che\ funziona)$$

$$P(funziona\ in\ parallelo) = P(funziona\ almeno\ uno) = 1 - P(non\ funzionano\ tutti)$$

$$= 1 - P(\bigcap_{i=1}^{n}\{i - esimo\ che\ non\ funziona\}) = \prod_{i=1}^{n} P(i - esimo\ che\ non\ funziona)$$

$$= 1 - \prod_{i=i}^{n}(1 - P(i\ che\ funziona))$$

Lezione del 18 Aprile 2023

Lezione 12

Classificatori naive di Bayes
Una semplice applicazione del teorema di Bayes ci permette di costruire un particolare classificatore naive di Bayes.
Consideriamo il caso più semplice possibile nella quale si osserva una serie di individui sulla presenza o l'assenza di una determinata proprietà.
Ottengo per ognuno di essi un osservazione $x_i \in \{0, 1\}$.
Ipotizziamo inoltre di aver associato ad ogni individuo un etichetta $y_i \in \{0, 1\}$ che denota l'appartenenza o meno ad una data classe.
Prendendo ad esempio il dataset dei supereroi supponiamo che la proprietà osservata faccia riferimento all'avere oppure no gli occhi neri ($x_i = 1$ ha gli occhi neri, $x_i = 0$ li ha di un altro colore) e la classe considerata indichi se l'editore corrispondente sia Marvel Comics ($y_i = 1$ è marvel, 0 altrimenti).
Indichiamo con:

- N = supereroe a caso ha gli occhi neri
- M = supereroe a caso è Marvel

Grazie al teorema di Bayes sappiamo che vale:
$P(M|N) = \frac{P(N|M)P(M)}{P(N)}$
Ora, se a partire dal dataset fosse possibile ottenere una stima delle probabilità che compaiono al secondo membro di questa equazione, sarebbe quindi immediato ottenere una stima di $P(M|N)$ da utilizzare per dire qualcosa sul fatto che un generico supereroe con gli occhi neri sia oppure no un supereroe Marvel.

In particolare, se il valore ottenuto fosse sufficientemente alto potremmo spingerci oltre e dire che d'ora in avanti potremo classificare tutti i supereroi con gli occhi neri di cui non conosciamo l'editore come supereroi Marvel.
Allo stesso modo, se il valore ottenuto fosse particolarmente basso potremmo comportarci in modo analogo, concludendo però che gli occhi neri indicano un editore diverso dalla Marvel.

Perchè usare questo classificatore?
Perchè potrebbe capitare di avere quel determinato tipo di attributo per alcuni dati e con questo classificatore potremmo fare un implicazione per i valori mancanti.

- $P(N|M)$ può essere approssimata calcolando la frequenza relativa con cui un supereroe Marvel del dataset ha gli occhi neri
- $P(M)$ può essere approssimata calcolando la frequenza relativa con cui un supereroe del dataset è edito dalla Marvel
- $P(N)$ può essere approssimata calcolando la frequenza relativa con cui un supereroe del dataset ha gli occhi neri

In Python:

```
import csv
import pandas as pd
import numpy as np

heroes = pd.read_csv('heroes.csv', sep=';', index_col=0)

marvel_heroes = heroes[heroes['Publisher']=='Marvel Comics']
p_marvel = len(marvel_heroes) / len(heroes)

black_eyes_heroes = heroes[heroes['Eye color']=='Black']
p_blackeyes = len(black_eyes_heroes) / len(heroes)

blackeyes_and_marvel_heroes = heroes[(heroes['Publisher']=='Marvel Comics') & \
                                     (heroes['Eye color']=='Black')]

p_blackeyes_given_marvel = len(blackeyes_and_marvel_heroes) / len(marvel_heroes)
```

```
Out:
0.45833333333333326
0.5416666666666665
```

```
#primo classificatore
print(p_blackeyes_given_marvel *
p_marvel / p_blackeyes)

#classificatore inverso
nonmarvel_heroes =
heroes[heroes['Publisher']!='Marvel
Comics']
blackeyes_and_nonmarvel_heroes
=
heroes[(heroes['Publisher']!='Marve
l Comics') & \
                        (heroes['Eye
color']=='Black')]

p_notmarvel =
len(nonmarvel_heroes) /
len(heroes)
p_blackeyes_given_notmarvel =
len(blackeyes_and_nonmarvel_her
oes) / len(nonmarvel_heroes)

p_blackeyes_given_notmarvel *
p_notmarvel / p_blackeyes
```

Calcolando le probabilità in quel modo (con un solo attributo, cioè gli occhi neri) non otteniamo informazioni molto dettagliate e rigide, possiamo estendere il numero di attributi ed il numero di casi possibili (non solo occhi neri ma tutti i colori possibili).
Consideriamo tutti i possibili valori per il colore degli occhi $\{o_1, \dots, o_n\}$ con l'evento corrispondente:
$O = o_i$ un supereroe a caso ha gli occhi del colore o_i
Lo stesso per il colore dei capelli:
$C = c_j$ un supereroe a caso ha gli occhi del colore c_j

$$P(M|O = o_i \cap C = c_j) = \frac{P(O=o_i \cap C=c_j|M)P(M)}{P(O=o_j \cap C=c_j)}$$

Anche qui possiamo approssimare tutto come visto in precedenza.

Variabile aleatoria
Ogni volta che osservo la probabilità varia secondo leggi aleatorie.
Immaginiamo di avere:

- Ω insieme degli esiti
- A algebra degli eventi
- P funzione di probabilità

Cosa succede se considero $X: \Omega \to \Re$ funzione che codifica ogni esito con un numero reale?
Questa funzione rappresenta una variabile aleatoria.
Se noi chiamiamo X = somma degli esiti del lancio di due dadi (con abuso di notazione) e $X = 3$
(Sarebbe corretto scrivere $\{X = 3\}$ cioè evento che si verifica lanciando due dadi e calcolando X sugli esiti ottengo 3).
$P(\{X = 3\}) = P(\{(1, 2), (2, 1)\}) = 2/36$
$P(\{X = 2\}) = P(\{1, 1\})$
...
$P(\{X = 12\})$
Man mano che aumenta il numero, aumenta la cardinalità dell'insieme.

N.B.
Non sempre una funzione che va da Ω a $\Re$ è una variabile aleatoria.

$\forall \alpha \in R \qquad X(\alpha) = \{w \in \Omega, X(w) \leq \alpha\} \in A$
$\{w \in \Omega, X(w) \leq \alpha\}$ sottoinsieme di Ω, cioè un evento
Se questo succede, allora X è una variabile aleatoria.

Quindi una variabile aleatoria è una funzione che mappa esiti casuali di Ω nel campo reale.

Partendo da una terna (Ω, A, P) posso creare tante variabili aleatorie.
Esempio:
$Y = esito\ del\ primo\ lancio$
$P(Y = i) = 1/6 \cdot I_{A=\{1, \dots, 6\}}(i) \qquad \forall i \in \Re$
I valori che possono essere assunti da una variabile aleatoria si chiamano specificazioni, nel nostro caso le specificazioni vanno da 1 a 6.
I_A funzione indicatrice
$I_A(x) = 1$ se $x \in A$, 0 altrimenti
A è fissato (di solito con l'insieme delle specificazioni).
Nel nostro caso se passo una specificazione come i otterrò come risultato $1/6$, altrimenti 0.

Esempio dei pennarelli:
F = funzionanti $\qquad$ D = difettosi
$P(F) = 0.7 \qquad P(D) = 0.3$
Ne compro due:
$\Omega = \{(d, d), (d, f), (f, d), (f, f)\}$
X = numero di pennarelli funzionanti

Supporto di $X = 0, 1, 2$
Per essere precisi, facciamo l'ipotesi che i due eventi che danno come risultato 1 siano indipendenti.
$P(X = 0) = 0.3 \times 0.3 = 0.09$
$P(X = 1) = P(\{(d, f), (f, d)\} = P(\{f, d\} \cup \{d, f\})$ (intersezione vuota)
$(0.3 \times 0.7) + (0.7 \times 0.3) = 0.42$
$P(X = 2) = 0.7 \times 0.7 = 0.49$

Strumenti matematici
Cos'è che fa la differenza nelle variabili aleatorie?
E' il dominio (supporto o insieme delle specificazioni) che può essere discreto o continuo.

Funzione di ripartizione (F_X)
$F_X: \Re \to [0, 1]$
$F_X(x) = P(X \leq x)$ quando indico una specificazione (x) uso le lettere minuscole.
Prendiamo:
$\{x \leq b\} = \{x \leq a\} \cup \{a < x \leq b\}$
$P(\{x \leq b\}) = P(\{x \leq a\}) + P(\{a < x \leq b\})$
$F_X(b) = F_X(a) + P(\{a < x \leq b\})$
$P(\{a < x \leq b\}) = F_X(b) - F_X(a)$

Lezione del 20 Aprile 2023

Lezione 13

Durante la scorsa lezione abbiamo introdotto le variabili aleatorie, più precisamente quelle su dominio discreto.
Inoltre abbiamo introdotto anche la funzione di ripartizione:
$F_X: \Re \to [0, 1]$
$F_X(x) = P(X \leq x)$
Nel caso di variabili aleatorie discrete abbiamo inoltre la funzione di massa di probabilità, associata anch'essa ad una variabile aleatoria X.
$f_X: \Re \to [0, 1]$
$f_X(x) = P(X = x)$

Quali sono le proprietà che questa funzione deve soddisfare?

- $\forall x \in \Re \qquad f_X(x) \geq 0$

- Se indichiamo con D il supporto della variabile aleatoria (insieme delle sue specificazioni)

$\sum_{x \in D} f_X(x_i) = 1$

$P(X = x_1) + P(X = x_2)$

Dove x_1 e x_2 sono eventi disgiunti

$= P(X = x_1 \vee X = x_2)$

Se su questi due esiti, quello tra parentesi è l'evento certo

$= 1$

Che relazione c'è tra la funzione di ripartizione e quella di massa di probabilità?

Esempio:

Variabile aleatoria X

$D_x = \{1, 2, 3\}$

$P(X = 1) = 1/2$

$P(X = 2) = 1/3$

x	$f_X(x)$
1	1/2
2	1/3
3	p

So che:

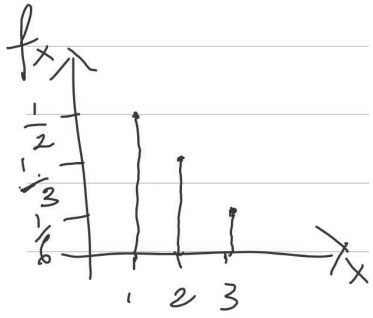
$f_X(1) + f_X(2) + f_X(3) = 1$

$1/2 + 1/3 + p = 1$

$p = 1/6$

Tracciamo il grafico

Attenzione, il dominio è $\mathfrak{R}$, in teoria potrei calcolare la probabilità di valori non nel dominio aleatorio (evento impossibile).



Proviamo a disegnare adesso il grafico della funzione di ripartizione:
Partiamo dalla funzione stessa:

$F_X(-4) = P(X \leq -4) = 0$

$F_X(1) = P(X \leq 1) = P(X = 1) = 1/2$

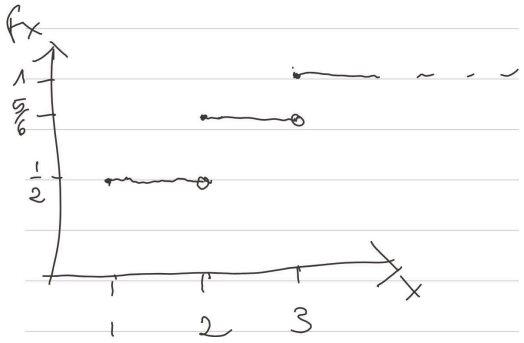
$F_X(1.3) = P(X \leq 1) = P(X = 1) = 1/2$

$F_X(2) = P(X \leq 2) = P(X = 1) + P(X = 2) = 5/6$

$F_X(2.1) = P(X \leq 2) = P(X = 1) + P(X = 2) = 5/6$

$F_X(3) = P(X \leq 3) = P(X = 1) + P(X = 2) + P(X = 3) = 1$

$F_X(4) = P(X \leq 3) = P(X = 1) + P(X = 2) + P(X = 3) = 1$



Il grafico di una funzione di ripartizione su una variabile aleatoria è quindi una funzione costante a tratti.
Data una funzione di ripartizione, quali sono le proprietà che deve rispettare?

- $\forall x \in \Re \qquad F_X(x) \geq 0$
- $\lim_{x \to +\infty} F_X(x) = 1$
- Deve essere una funzione continua da destra

Inoltre, considerando i salti della funzione posso risalire alla differenza della funzione di ripartizione tra i due valori.

Esempio:
$F_X(2) - F_X(1) = P(1 < X \leq 2) = P(X = 2) = 1/3 = f_X(2)$

Concetto di valore atteso di una variabile aleatoria X
(Partiremo con il concetto di valore atteso di variabili aleatorie discrete per poi successivamente passare a quelle continue)
Ovviamente considerando il loro supporto D.

$D = \{x_1, x_2, \ldots\}$

$E(X) = \sum_{x_i \in D} x_i P(X = x_i)$

Fondamentalmente è una media pesata delle specificazioni della variabile.
A differenza della media ho bisogno oltre ai valori, anche la loro probabilità (o la loro funzione di massa di probabilità).

Esempio:
Contesto del gioco d'azzardo, molti esiti possibili con le loro probabilità corrispondenti (con vincite positive e negative).

Vincite possibili $x_1, \dots, x_m$
Numero giocate $n_1, \dots, n_m$
$n_i = quante\ volte\ gioco\ e\ vinco\ x_i$
Vincita media $= \frac{n_1 x_1 + \dots + n_m x_m}{m} = \frac{n_1}{m} x_1 + \dots + \frac{n_m}{m} x_m$
Dove $\frac{n_1}{m}$ e $\frac{n_m}{m}$ sono frequenze relative, le quali approssimano il concetto di probabilità
$= P(X = x_1) x_1 + \dots + P(X = x_m) x_m = E(X)$

Esempi:
Lancio di una moneta bilanciata:
Variabile aleatoria X con $D = \{0, 1\}$
$P(X = 0) = 1/2 \qquad P(X = 1) = 1/2$
$E(X) = 0 \cdot 1/2 + 1 \cdot 1/2 = 1/2$
E se la moneta non fosse bilanciata?
$P(X = 0) = 2/3 \qquad P(X = 1) = 1/3$
$E(X) = 0 \cdot 2/3 + 1 \cdot 1/3 = 1/3$

Notiamo che il valore atteso non dipende dalle specificazioni.
Notiamo anche che in questo caso il valore atteso sia direttamente la probabilità di ottenere 1.

Trasformazione di variabili aleatorie
$g: \Re \to \Re$
$g(X) = Y$
Se conosco $E(X)$ posso risalire anche al valore atteso di Y?
Consideriamo la specificazione $D = \{0, 1, 2\}$ e $Y = g(X) = x^2$

y	x	$P(X = x) = P(Y = y)$

0	0	0.2
1	1	0.5
4	2	0.3

$E(Y) = E(X^2) = 0 \cdot 0.2 + 1 \cdot 0.5 + 4 \cdot 0.3 = 1.7$
$= 0^2 \cdot 0.2 + 1^2 \cdot 0.5 + 2^2 \cdot 0.3 = 1.7$

$E(g(X)) = \sum_{x_i \in D} g(x_i) P(X = x_i)$

E se $g(X) = aX + b$

$E(aX + b) = \sum_{x_i \in D} (ax_i + b) P(X = x_i) = \sum_{x_i \in D} ax_i P(X = x_i) + \sum_{x_i \in D} bP(X = x_i)$

$= a \sum_{x_i \in D} x_i P(X = x_i) [= E(X)] + b \sum_{x_i \in D} P(X = x_i) [= 1]$
$= aE(X) + b$
Caso particolare
$a = 0$
$E(b) = b$

Cos'è b?
L'unico valore aleatorio osservabile

$b = 0$
$E(aX) = aE(X)$

Come abbiamo visto la varianza per la distribuzione e per la media campionaria possiamo vedere quanto i valori si disperdono dal valore atteso.

Esempio:
Con W, Y, Z come variabili aleatorie ma:

- $P(W = 0) = 1$
  - $E(W) = 0$
- $P(Y = -1) = 1/2$ $\quad P(Y = 1) = 1/2$
  - $E(Y) = -1 \times 1/2 + 1 \times 1/2 = 0$
- $P(Z = -100) = 1/2$ $\quad P(Z = 100) = 1/2$
  - $E(Z) = 0$

Il loro valore atteso non cambia ma la loro dispersione rispetto ad esso si.

X variabile aleatoria
$\mu = E(X)$
$E(|X - \mu|)$
Di solito non si usa il valore assoluto ma il quadrato, poiché più comodo da manipolare.
$E((X - \mu)^2) = Var(X)$ (Varianza)
$= E(X^2 - 2X\mu + \mu^2)$
Per linearità posso fare:
$= E(X^2) + E(-2\mu X) + E(\mu^2)$
Inoltre posso portare fuori eventuali costanti moltiplicative:
$= E(X^2) - 2\mu E(X) + \mu^2$
$= E(X^2) - 2\mu^2 + \mu^2$
$= E(X^2) - (E(X))^2$
La prima parte viene chiamata momento secondo della variabile aleatoria ($E(X^2)$)

Esempio:
Sia X la variabile aleatoria corrispondente al valore sulla faccia di un dato bilanciato.
$D = \{1, 2, 3, 4, 5, 6\}$
$f_X(x) = 1/6 \cdot I_D(x)$
$$E(X) = \sum_{i=1}^{6} i \cdot 1/6 = 1/6 \cdot \sum_{i=1}^{6} i = 7/2$$

Calcoliamo adesso la varianza
$$Var(X) = E((X - 7/2)^2) = \sum_{i=1}^{6} (i - 7/2)^2 \cdot 1/6 \ldots$$
Usiamo la formula alternativa
Calcolo il momento secondo:
$$E(X^2) = \sum_{i=1}^{6} i^2 \cdot 1/6 = 1/6 \cdot \sum_{i=1}^{6} i^2$$
Ricordiamo che:
$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$
Quindi:
$$= 1/6 \cdot \frac{6 \cdot 7 \cdot 13}{6} = 91/6$$

$Var(X) = 91/6 - 49/4 = 35/12 = \frac{36-1}{12} = \frac{6^2-1}{12}$
Noteremo che succederà sempre per spazi equiprobabili.

Come cambia la varianza con le trasformazioni lineari?
Considero:
$X \qquad Y = aX + b$
$Var(Y) = E((Y - E(Y))^2)$
$E(Y) = aE(X) + b$
Quindi:
$= E((aX + b - aE(X) - b)^2)$
$= E((a(X - E(X))^2)$
$= a^2 E((X - E(X))^2)$ $[= Var(X)]$
$= a^2 Var(X)$

Caso particolare:
$a = 0$
$Var(b) = 0$
Variabile aleatoria costante non si discosta mai dal valore atteso b.

Deviazione standard
La deviazione standard campionaria era stata introdotta precedentemente per normalizzare i risultati ottenuti ed anche per normalizzare le unità di misura.
Anche per le variabili aleatoria abbiamo la stessa cosa.

$E(X) = \mu_X$
$Var(X) = \sigma_X^2$

Deviazione standard
$= \sqrt{Var(X)} = \sigma_X$

Possiamo considerare le variabili aleatorie a coppie proprio come con i numeri.
X, Y variabili aleatorie

Funzione di ripartizione estesa per le variabili aleatorie
$F_{X,Y}(x, y) = P(\{X \leq x\} \wedge \{Y \leq y\})$
Funzione di ripartizione congiunta (di X e Y).

$$\lim_{y \to +\infty} F_{X,Y}(x, y) = \lim_{y \to +\infty} P(\{X \leq x\} \wedge \{Y \leq y\}) = P(X \leq x) = F_X(x)$$
"Ripartizione marginale"

Funzione di massa di probabilità congiunta (di X e Y)
$f_{X,Y}(x, y) = P(X = x \wedge Y = y)$

$$= \sum_{x_i \in D_X} F_{X,Y}(x_i, y_j) = \sum_{x_i \in D_X} P(X = x_i \wedge Y = y_j)$$
Fissato un y_j sommo gli x_i nella specificazione di X.

$$= P(\bigcup_{x_i \in D_X} \{X = x_i \wedge Y = y_j\}) = P(Y = y_j) = f_Y(y_j)$$
"Funzione di massa di probabilità marginale"

In statistica descrittiva avevamo introdotto le funzioni congiunte per il concetto di relazione.
Nel caso delle variabili aleatorie possiamo re-introdurre tale concetto, utilizzandolo per il concetto di indipendenza tra variabili aleatorie.

Immaginiamo di avere 2 specificazioni per 2 variabili aleatorie, conoscendone una cosa posso dire sull'altra?
X, Y indipendenti se e solo se $\forall A, B \subseteq \Re$
$P(X \in A, Y \in B) = P(X \in A)P(Y \in B)$

Proprietà:
Se X e Y sono indipendenti, allora $f_{X,Y}(x, y) = f_X(x)f_Y(y)$ (vale il viceversa)
$\forall x, y$
Dimostrazione $\Rightarrow$
X, Y indipendenti $\Rightarrow \forall x, y \quad \{x\} \subseteq \Re$ e $\{y\} \subseteq \Re \Rightarrow P(X \in \{x\}, Y \in \{y\}) = P(X \in \{x\})P(Y \in \{y\}) = P(X = x, Y = y) = P(X = x)P(Y = y)$
$[f_{X,Y}(x, y) = f_X(x)f_Y(y)]$
Dimostriamo il viceversa
$\forall x, y \ f_{X,Y}(x, y) = f_X(x)f_Y(y) \rightarrow$ fisso $A, B \subseteq \Re$
$A = \{x_i\} \qquad B = \{y_j\}$

$$= P(X \in A, y \in B) = \sum_{x \in A} \sum_{y \in B} P(X = x, Y = y)$$

$$= \sum_{x \in A} \sum_{y \in B} f_{X,Y}(x, y)$$

$$= \sum_{x \in A} \sum_{y \in B} f_X(x)f_Y(y)$$
Sommatoria separabile

$= \sum_{x \in A} f_X(x) \sum_{y \in B} f_Y(y)$
$= P(X \in A)P(X \in B)$
$\forall A, B \subseteq R \quad X, Y$ sono indipendenti

Questo si generalizza fino a serie di n variabili aleatorie (vettori di variabili aleatorie).
L'indipendenza si ottiene allo stesso modo:
$X_1, \dots, X_n$
$\forall A_1, \dots, A_n \subseteq \Re$
$P(\bigcap_{i=1}^{n} \{x_i \in A_i\}) = \prod_{i=1}^{n} P(x_i \in A_i)$

Lezione del 27 Aprile 2023

Lezione 14

Durante la lezione precedente abbiamo visto la differenza tra variabili aleatorie univariate e multivariate e come cambiano quindi anche i loro indici matematici.

Mettiamo adesso di avere una funzione con variabili aleatorie:
$g: (X, Y) := Z$
$E(Z) = E(g(X, Y))$

$= \sum_{x, y} g(x, y) f_{X,Y}(x, y)$
Nella sommatoria, x e y variano nei rispettivi supporti (o domini) di X e Y

$(\sum_{x \in D_X} \sum_{y \in D_Y})$.
$g(X, Y) = X + Y$

$E(X + Y) = \sum_{x, y} (x + y) f_{X,Y}(x, y) = \sum_{x, y} x f_{X,Y}(x, y) + \sum_{x, y} y f_{X,Y}(x, y)$

$= \sum_{x \in D_X} x \sum_{y \in D_Y} f_{X,Y}(x, y) + \sum_{y \in D_Y} y \sum_{x \in D_X} f_{X,Y}(x, y)$

$\sum_{y \in D_Y} f_{X,Y}(x, y)$ e $\sum_{x \in D_X} f_{X,Y}(x, y)$ sono funzioni di massa di probabilità marginali, e quindi possiamo riscriverle come $f_X(x)$ e $f_Y(y)$.

$$\sum_{x \in D_X} x f_X(x) + \sum_{y \in D_Y} y f_Y(y) = E(X) + E(Y)$$

E se volessi calcolare il valore atteso della somma di 3 variabili aleatorie?

$E(X + Y + Z) = E((X + Y) + Z)$

$(X + Y) = W$

$= E(W) + E(Z) = E(X + Y) + E(Z)$

$= E(X) + E(Y) + E(Z)$

Quindi:

$$E(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} E(X_i)$$

Questa proprietà può tornare utile quando vogliamo considerare una variabile aleatoria come una somma di più variabili aleatorie per semplificare il calcolo del valore atteso.

Esempio

X = somma degli esiti del lancio di due dadi bilanciati

$$E(X) = \sum_{i=2}^{12} iP(X = i) = \ \ldots$$

Ma se vediamo:

$X = X_1 + X_2$

Come variabili aleatorie che rappresentano l'esito del lancio del primo e del secondo dado, sapendo che il valore atteso del lancio di un singolo dado è 7/2, possiamo giungere alla conclusione che il valore atteso di X sia 7.

Esercizi:

Devo mandare n lettere composte da un foglio e una busta.
Mi cadono le buste.
Rimetto a ciascuna busta i fogli.
Qual'è la probabilità che il foglio sia finito nella sua lettera?
Definiamo:

$\forall i = 1, \ldots, n$

$X_i = 1$ se l'i-esima lettera coincide con la busta, 0 altrimenti

$P(X_i = 1) =?$

$= \frac{1}{n}$

X = numero di lettere nella loro busta

$X = \sum_{i=1}^{n} X_i = n$ se tutte le lettere sono nella loro busta, 0 se nessuna lettera è nella sua busta

$E(X_i) = ?$

$= 0 \cdot P(X_i = 0) + 1 \cdot P(X_i = 0) = \frac{1}{n}$

$E(X) = E(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} E(X_i) = 1$

Altro esercizio:
In ogni confezione di un determinato prodotto c'è un buono sconto (ce ne sono di 20 tipi inseriti equi probabilmente). Se apro 10 confezioni quanti buoni diversi trovo?

20 tipi di buoni diversi
10 confezioni comprate
X =n° di buoni diversi trovati nelle 10 confezioni

Approccio precedente (si chiama approccio composizionale)

$X = \sum_{i=1}^{20} X_i$

$X_i = 1$ se l'i-esimo buono è in almeno una confezione, 0 altrimenti

$E(X_i) = P(X_i = 1) = P(in\ almeno\ una\ delle\ 10\ confezioni\ c'è\ il\ buono\ i-esimo)$

$= 1 - P(non\ c'è\ nemmeno\ un\ buono\ i-esimo\ nelle\ 10\ confezioni)$

$= 1 - P(\bigcap_{j=1}^{10} buono\ i\ non\ presente\ nella\ confezione\ j)$

$= se\ gli\ eventi\ sono\ indipendenti\ faccio\ il\ prodotto\ delle\ probabilità$

$= 1 - \prod_{j=1}^{10} P(buono\ i\ non\ è\ nella\ confezione\ j)$

$P(buono\ i\ nella\ confezione\ j) = \frac{1}{20}$

$P(buono\ i\ non\ è\ nella\ confezione\ j) = \frac{19}{20}$

$= 1 - \prod_{j=1}^{10} \frac{19}{20} = 1 - (\frac{19}{20})^{10}$

Quindi:

$E(X) = \sum_{i=1}^{20} E(X_i) = \sum_{i=1}^{20} 1 - (\frac{19}{20})^{10} = 20(1 - (\frac{19}{20})^{10}) \approx 8.025$

E se volessimo sostituire una variabile aleatoria con un valore fisso (quindi avente tutte le sue specificazioni uguali)? Quale valore dovremmo utilizzare?

Variabile aleatoria X
$\mu = E(X)$

Considerando:
$E((X - c)^2)$, dove $(X - c)$ è la distanza tra la variabile aleatoria ed un eventuale valore fisso da sostituire.
Sommiamo e sottraiamo μ all'interno del quadrato:
$E(((X - \mu) + (\mu - c))^2) = E((X - \mu)^2 + 2(X - \mu)(\mu - c) + (\mu - c)^2)$
$= E((X - \mu)^2) + 2(\mu - c)E(X - \mu) + (\mu - c)^2$
Dove:
$E(X - \mu) = E(X) - E(\mu) = \mu - \mu = 0$ e $(\mu - c)^2 \geq 0$
Quindi:
$E((X - c)^2) \geq E((X - \mu)^2)$ $[= \sigma^2]$

Quindi come faccio a collassare la nostra variabile aleatoria in una costante?
Scegliamo una costante con errore minimo, quindi vicina al valore atteso.

N.B.
$Var(X + X) = Var(2X) = 4Var(X)$ non $2Var(X)$

Covarianza tra due variabili aleatorie
$Cov(X, Y) = E((X - \mu_Y)(Y - \mu_Y))$
Dove:
$\mu_X = E(X)$ $\mu_Y = E(Y)$
Proprietà:

1) Essa è simmetrica (per la simmetria del prodotto)
2) $Cov(X, Y) = E(XY - \mu_Y X - \mu_X Y + \mu_X \mu_Y) = E(XY) - \mu_Y E(X) - \mu_X E(Y) + \mu_X \mu_y$
   $= E(XY) - \mu_X \mu_Y - \mu_X \mu_Y + \mu_X \mu_Y = E(XY) - \mu_X \mu_Y = E(XY) - E(X)E(Y)$

Notiamo quindi cosa accade alla covarianza se applico trasformazioni lineari agli argomenti:

3) $Cov(aX, Y) = E((aX - \mu_{aX})(Y - \mu_Y)) = E((aX - aE(X))(Y - \mu_Y))$
   $= aE((X - \mu_X)(Y - \mu_Y)) = aCov(X, Y)$

4) $Cov(X + b, Y) = E((X + b)Y - E(X + b)E(Y))$
$= E(XY + bY - E(X)E(Y) + bE(Y))$
$= E(XY - E(X)E(Y)) + bE(Y + E(Y)) = Cov(X, Y)$

5) $Cov(X + Y, Z) = E((X + Y)Z - E(X + Y)E(Z))$
$= E(XZ + YZ) - (E(X) + E(Y))E(Z)$
$= E(XZ) + E(YZ) - E(X)E(Z) - E(Y)E(Z)$
$= E(XZ) - E(X)E(Z) + E(YZ) - E(Y)E(Z) = Cov(X, Z) + Cov(Y, Z)$

6) $Cov(X, X) = E((X - \mu_X)(X - \mu_X)) = E((X - \mu)^2) = Var(X)$
Funziona anche con la formula alternativa:
$E(XX) - E(X)E(X) = E(X^2) - E(X)^2$

7) $Var(X + Y) = E((X + Y)^2) - (E(X + Y))^2$
$= E(X^2 + 2XY + Y^2) - (E(X) + E(Y))^2$
$= E(X^2) + 2E(XY) + E(Y^2) - E(X)^2 - 2E(X)E(Y) - E(Y)^2$
$= Var(X) + Var(Y) + 2Cov(X, Y)$
Se X e Y fossero uguali?
$= Var(X + X)\ [= 4Var(X)] = Var(X) + Var(X) + 2Cov(X, X) = 4Var(X)$

Se ci fossero più di due variabili aleatorie avrei la somma delle varianze e la somma di tutte le possibili covarianze a 2 a 2.

Possiamo dire qualcosa sulla covarianza di due variabili aleatorie indipendenti?
X, Y Variabili aleatorie indipendenti

$E(XY) = \sum_{x,y} xyf_{X,Y}(x, y)$
Dato che sono indipendenti, possiamo fattorizzare la funzione in due marginali $f_X(x)f_Y(y)$

$= \sum_{x,y} xyf_X(x)f_Y(y) = \sum_x xf_X(x) \sum_y yf_Y(y) = E(X)E(Y)$

Quindi:
$Cov(X, Y)$ con X, Y indipendenti
$= E(XY) - E(X)E(Y) = E(X)E(Y) - E(X)E(Y) = 0$

Nel caso di covarianza campionaria uguale a zero i due attributi venivano considerati indipendenti.

Sempre con X e Y indipendenti otteniamo che:
$Var(X + Y) = Var(X) + Var(Y) + 2Cov(X, Y)\ [= 0]$

In generale, se ho n variabili aleatorie indipendenti, la varianza della somma è uguale alla somma delle varianze.

Esercizio:
Considerando la somma di 10 lanci di dadi bilanciati, qual'è il valore della varianza?
$X_1, X_2, \dots, X_{10}$
X_i = esito del lancio del dado numero i

$$X = \sum_{i=1}^{10} X_i$$

$$Var(X) = Var(\sum_{i=1}^{10} X_i) = \sum_{i=1}^{10} Var(X_i) = \sum_{i=1}^{10} \frac{35}{12} = \frac{175}{6}$$

E se avessi avuto 10 lanci di monete?
Cambia solo la varianza singola

X_i = lancio della moneta i
$X_i = 1$ testa, 0 altrimenti

$$E(X_i) = \frac{1}{2}$$

$$Var(X_i) = E(X_i^2) \; [= X_i \; dato \; che \; 1^2 = 1 \; e \; 0^2 = 0] - E(X_i)^2 = 1/2 - 1/4 = 1/4$$

$$Var(X) = \sum_{i=1}^{10} Var(X_i) = \frac{5}{2}$$

La varianza, oltre a mostrare l'indipendenza tra due variabili aleatorie, quantifica la dipendenza delle loro distribuzioni.

Funzione indicatrice di un evento
Avendo un evento A
$I_A = 1$ se l'evento A si verifica, 0 altrimenti
Questa è una variabile aleatoria

Avendo:
$A, B \in A \; [Algebra \; degli \; eventi] \rightarrow X = I_A, Y = I_B$
$XY = 1$ se $A \wedge B$ si verifica, 0 altrimenti

$E(X) = P(X = 1)$
$E(Y) = P(Y = 1)$
$E(XY) = P(X = 1 \wedge Y = 1)$
$Cov(X, Y) = E(XY) - E(X)E(Y) = P(X = 1 \wedge Y = 1) - P(X = 1)P(Y = 1)$
Mettiamo caso che la covarianza sia positiva:

$P(X = 1 \wedge Y = 1) > P(X = 1)P(Y = 1)$
$\frac{P(X=1 \wedge Y=1)}{P(Y=1)} > P(X = 1)$
La frazione rappresenta una probabilità condizionata:
$P(X = 1|Y = 1) > P(X = 1)$
Se so che $Y = 1$ è più probabile che X sia uguale a 1.

Una covarianza positiva rende significativo sapere che con una variabile aleatoria grande abbiamo probabilità maggiore che anche la seconda sia grande.
Relazione diretta

Mettiamo adesso invece che la covarianza sia negativa.
Il procedimento rimane invariato cambia solo il segno:
$P(X = 1|Y = 1) < P(X = 1)$
Relazione inversa

Come per la varianza campionaria abbiamo avuto problemi per la normalizzazione del range di valori, anche nelle variabili aleatorie possiamo riscontrare lo stesso problema.

Coefficiente di correlazione tra due variabili aleatorie.
$Cov(2X, 2Y) = 4Cov(X, Y)$
Risultato controintuitivo dato che, scalando due variabili aleatorie, la relazione fra loro non dovrebbe cambiare.
Usiamo quindi:
$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}$
Infatti:
$\rho_{2X,2Y} = \frac{Cov(2X,2Y)}{\sigma_{2X}\sigma_{2Y}}$
$\sigma_{2X} = \sqrt{Var(2X)} = 2\sqrt{Var(2X)} = 2\sigma_X$
$\rho_{2X,2Y} = \frac{4Cov(X,Y)}{2\sigma_X 2\sigma_Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} = \rho_{X,Y}$
Inoltre questo coefficiente è adimensionale.

Lezione del 2 Maggio 2023

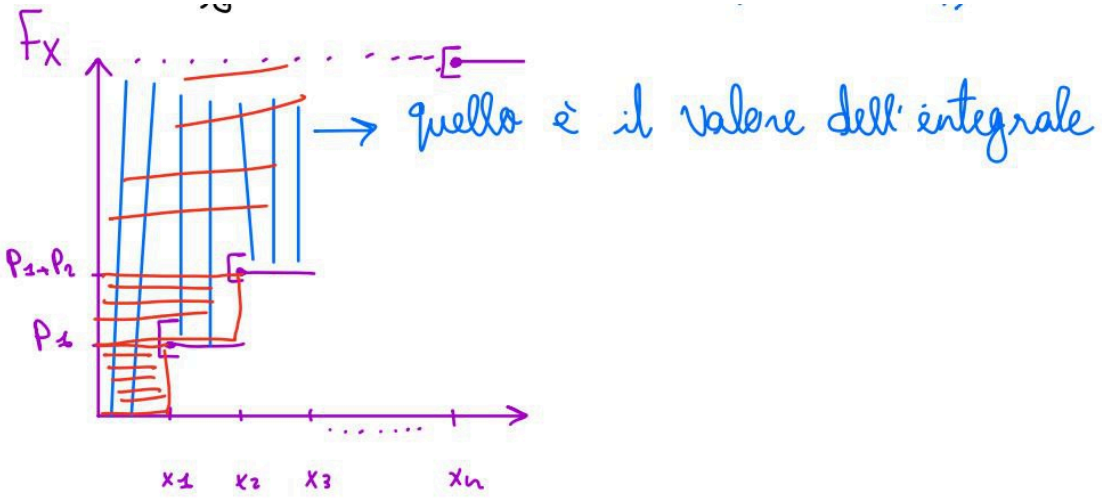
Lezione 15

Nella scorsa lezione abbiamo visto le proprietà degli indici di varianza e di covarianza delle variabili aleatorie discrete.

Mettiamo adesso di avere una variabile aleatoria X caratterizzata dalla sua funzione di massa di probabilità $f_X(x_i) = P_i$.
Vediamo adesso un modo alternativo per calcolare $E(X)$

Sappiamo che $E(X) = \sum_{x_i \in D_X} x_i P_i$
e sappiamo anche che la funzione di ripartizione F_X ha un grafico a tratti.
Ogni volta che F_X fa un salto quanto vale?
Esso sarà uguale al valore dell'ascissa in cui F_X ha fatto il salto.

$E(X) = \int_0^{+\infty} 1 - F_X(x)dx$ (solo quando $X \geq 0$)



Per sapere il valore di questo integrale dobbiamo calcolare l'area dei rettangoli in rosso:

$\Rightarrow \int_0^{+\infty} 1 - F_X(x)dx = x_1 p_1 + x_2(p_1 + p_2) + \ldots \; [= x_1 f_X(x_1) + \ldots]$

Teorema della disuguaglianza di Markov
Per le variabili aleatorie maggiori o uguali di zero $[X \geq 0]$
$\forall a \in \Re \qquad P(X \geq a) \leq \frac{E(X)}{a}$
Cosa succede quando a aumenta?
$\frac{E(X)}{a}$ diminuisce a sua volta quindi diminuisce (intuitivamente) anche la probabilità.

Dimostrazione:

$E(X) = \sum_{x \in D_X} x f_X(x)$ dividiamo in due la sommatoria

$E(X) = \sum_{x<a} xf_X(x) + \sum_{x\geq a} xf_X(x)$
Dato che la prima sommatoria è di elementi strettamente maggiori o uguali di zero posso rimuoverla sostituendo l'uguale con ≥

$E(X) \geq \sum_{x\geq a} xf_X(x)$
Adesso so che tutte le x che compaiono nella sommatoria sono $\geq a$.

Posso riscrivere quindi la sommatoria come $\sum_{x\geq a} af_X(x) = a\sum_{x\geq a} f_X(x)$ dove

$\sum_{x\geq a} f_X(x)$ possiamo scriverla come $P(X \geq a)$ (dato che nella sommatoria sommo le probabilità nelle quali X è maggiore o uguale di a), ottenendo come risultato:
$= aP(X \geq a)$
$\Rightarrow E(X) \geq aP(X \geq a) \Rightarrow P(X \geq a) \leq \frac{E(X)}{a}$
Tesi dimostrata.

Inoltre:
$P(X < a) = 1 - P(X \geq a)$
Quindi $P(X < a) \geq 1 - \frac{E(x)}{a}$

Teorema della disuguaglianza di Tchebycheff (o Chebyshev)
Ho una variabile aleatoria X con valore atteso $E(X) = \mu \in \Re$ e varianza $Var(X) = \sigma^2$.
$\forall r > 0$
$\Rightarrow P(|X - \mu| \geq r) \leq \frac{\sigma^2}{r^2}$
(Il valore assoluto all'interno della probabilità è un modo di calcolare la distanza fra la variabile aleatoria ed il suo valore atteso).

Dimostrazione:
Se $|X - \mu| \geq r \Leftrightarrow (X - \mu)^2 \geq r^2$ e visto che $r > 0$ allora $\Rightarrow$ vale il viceversa.
Cosa posso dire sulla probabilità di due eventi che co implicano?
Che la loro probabilità è uguale.
$\Rightarrow P(|X - \mu| \geq r) = P((X - \mu)^2 \geq r^2)$
Chiamo adesso $Y := (X - \mu)^2$
Ottenendo:
$\Rightarrow P(|X - \mu| \geq r) = P(Y \geq r^2)$
Per il teorema della disuguaglianza di Markov appena visto:

$[P(X \geq a) \leq \frac{E(X)}{a}]$
Allora:
$\Rightarrow P(|X - \mu| \geq r) \leq \frac{E(Y)}{r^2}$
Ma $E(Y) = E((X - \mu)^2) = Var(X) = \sigma^2$ quindi
$\Rightarrow \leq \frac{\sigma^2}{r^2}$

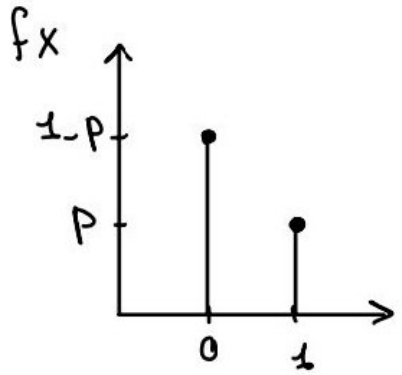
Cosa succede al crescere di r?
L'evento "$|X - \mu| \geq r$" diventa sempre meno probabile.

E se mettessimo $r := k\sigma$?
$P(|X - \mu| \leq k\sigma) \leq \frac{\sigma^2}{k^2\sigma^2} = \frac{1}{k^2}$ con $k > 0$

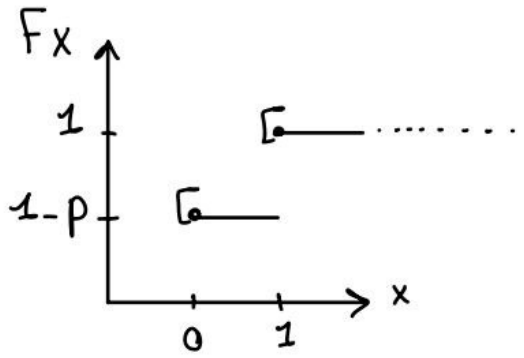
Esempio:
$X = numero\ di\ pezzi\ prodotti$
$E(X) = 50$
$P(X \geq 75) =?$
Qual'è la probabilità che vengano prodotti più di 75 prodotti? Visto che ho una $X > 0$ posso applicare Markov
$\Rightarrow P(X \geq 75) \leq \frac{E(X)}{75} = 50/75 = 2/3$
E se sapessimo che $Var(X) = 25$ potremmo applicare anche Tchebycheff.
Ragioniamo su $P(40 < X < 60)$, scriviamolo adesso in forma di disuguaglianza:
$E(X) = \mu = 50$
$P(40 - 50 < X - \mu < 60 - 50) = P(-10 < X - \mu < 10)$
$= 1 - P(|X - \mu| \geq 10) \geq [non \leq perch\grave{e}\ ho\ "1 - "]1 - \frac{\sigma^2}{100} = 1 - 25/100$
$1 - 1/4 = 3/4$
$\Rightarrow 1 - P(|X - \mu| \geq 10) \geq 3/4$

Modelli di variabili aleatorie
Primo modello (o famiglia) di variabili aleatorie discrete.
Esperimento Bernoulliano
Se dobbiamo costruire una variabile aleatoria a partire da questo esperimento dobbiamo mappare gli esiti:
$\Omega = \{p, n\}$
$A = 2^{\Omega}$
$P(p) = p$
$P(n) = 1 - p$
Fissiamo un valore per $p \in [0, 1] \rightarrow$ parametri

Variabile aleatoria di Bernoulli X con supporto $D_X = \{0, 1\}$
Funzione di massa di probabilità:
$f_X(X) = p$ se $x = 1$ oppure è uguale a $1 - p$ se $x = 0$



Funzione di ripartizione
$F_X(x) = 0$ se $x < 0$, è uguale a $1 - p$ se $0 \leq x < 1$ oppure è uguale ad 1 se $x \geq 1$



Applichiamo la definizione di del valore atteso per calcolare $E(X)$

$$E(X) = \sum_{x \in D_X} x f_X(x) = o f_X(0)\,[= 0] + 1 f_X(1) = p$$

$\Rightarrow E(X)$ con variabile aleatoria X di Bernoulli → Valore atteso uguale al suo parametro

Possiamo usare anche $\int_0^{+\infty} 1 - F_X(x) = 1 - (1 - p) = p$

Calcoliamo adesso $Var(X)$
Con la definizione

$Var(X) = E((X - p)^2) = \sum_{x \in D_X} (x - p)^2 f_X(x) = (0 - p)^2(1 - p) + (1 - p)^2(p)$

$= (1 - p)(p^2 + (1 - p)p) = (1 - p)(p^2 + p - p^2) = p(1 - p)$

Altro metodo per calcolare $Var(X)$

$E(X^2) - E(X)^2 = E(X) - E(X)^2 = p - p^2 = p(1 - p)$

Possiamo riscrivere $f_X(x)$ in un altro modo?

Per adesso abbiamo $f_X(x) = p$ se $x = 1$, $1 - p$ se $x = 0$ e 0 altrimenti (per far in modo che il dominio sia $\Re$).

Possiamo quindi riscriverla come:

$f_X(x) = p^x(1 - p)^{1-x} I_{\{0, 1\}}(x)$

Possiamo riscrivere quindi anche $F_X(x) =$?

$F_X(x) = (1 - p) I_{[0, 1)}(x) + I_{[1, +\infty)}(x)$

Immaginiamo di ripetere l'esperimento di Bernoulli n volte in condizione di indipendenza.

In questo modo i parametri della nostra seconda famiglia di variabili aleatorie diventano 2.

Questa viene chiamata famiglia Binomiale.

$X = numero\ di\ successi\ (dell\ esperimento\ bernoulliano)$

$X \sim B(n, p)$ (X è distribuito come una variabile aleatoria di tipo binomiale di parametri n e p)

Definiamo $D_X = \{1, 2, \dots, n\}$

Come sono n e p?

$n \in \aleph \qquad p \in [0, 1]$

Definiamo $f_X(x)$

Prendo $i \in D_X$ e calcolo $P(X = i)$

Immaginiamo che i successi sono nei primi esperimenti:

$s, s, s, \dots, s\ [i\ volte],\ f, f, \dots, f\ [n - i\ volte]$

Calcolo quindi la probabilità di questo evento.

- Probabilità di avere successo $= p$
- Probabilità di avere un fallimento $= 1 - p$

$\Rightarrow p \cdot p \cdot \dots \cdot p \cdot (1 - p) \cdot (1 - p) \cdot \dots \cdot (1 - p) = p^i(1 - p)^{n-i}$

$\Rightarrow P(X = i) = (n\ i)[binomio\ di\ Newton] p^i(1 - p)^{n-i}$ se $i \in \{1, 2, \dots, n\}$ altrimenti aggiungo $\cdot\ I_{\{1, 2, \dots, n\}}(x)$

Controlliamo che sia una funzione di massa di probabilità corretta:

- Positiva

Sempre per definizione

- Sommi a 1

$$\sum_{i=0}^{n} f_X(x_i) = \sum_{i=0}^{n} (n\ i)p^i(1-p)^{n-i} = (p+1-p)^n = 1$$

Esempio:

$P(pennarello\ difettoso) = 0,01$

Ho 10 pennarelli per confezione.

Se ho più di un pennarello difettoso posso chiedere il rimborso.

Qual'è la percentuale di confezioni di pennarelli restituite?

Quindi, qual'è la probabilità con la quale una confezione di pennarelli venga restituita?

$= P(su\ 10\ pezzi\ almeno\ 2\ difettosi) = P(X \geq 2) = 1 - P(X < 2)$

Introduco $X = numero\ di\ pennarelli\ difettosi\ in\ una\ confezione \sim B(10,\ 0,01)$

$$\sum_{x \geq 2} f_X(x) = \sum_{i=2}^{n} (n\ i)p^i(1-p)^{n-i} = P(X < 2)$$

$$\Rightarrow 1 - P(X < 2) = 1 - \sum_{i=0}^{1} (n\ i)p^i(1-p)^{n-i} = 1 - 0.99^{10} - 10 \cdot 0.01^1 \cdot 0.99^9$$

≈ 0.0043 $\quad p' = 0.43\%$ percentuale di restituzione di una confezione.

Altro esempio:

Se compro 3 confezioni, qual'è la probabilità che io ne restituisca esattamente 1?

$Y = numero\ di\ confezioni\ restituite\ su\ 3\ acquistate \sim B(3,\ p')$

$P(Y = 1) = f_X(1) = (3\ 1)p'^1(1-p)^2 \approx 0.013$

E di restituirne almeno una?

$P(Y > 0) = 1 - P(Y = 0) = 1 - (1-p)^3 \approx 0.0128$

Lezione del 4 Maggio 2023

Lezione 16

Durante la scorsa lezione abbiamo introdotto le famiglie di distribuzione, in particolare quella Bernoulliana e quella Binomiale, nella quale ripeto n volte un esperimento bernoulliano contando i successi.

$X \sim B(n,\ p)$

$$E(x) = \sum_{x=0}^{n} x(n\ x)p^{x}(1 - p)^{n-x}$$

Dove:

$$(n\ x)p^{x}(1 - p)^{n-x} = f_{X}(x)$$

Inoltre, per $i = 1, \dots, n$
Abbiamo una variabile aleatoria bernoulliana
$X_i = 1$ se ho un successo all'i-esima prova, 0 altrimenti

$$X = \sum_{i=i}^{n} X_i$$

$$E(X) = E(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} E(X_i)$$

$X_i \sim B(p)$ (modello bernoulliano)

$$= \sum_{i=1}^{n} p = np$$

E per la varianza?

$$Var(X) = Var(\sum_{i=1}^{n} X_i)$$

Dato che le variabili aleatorie bernoulliane sono tra loro indipendenti, ho:

$$= \sum_{i=1}^{n} Var(X_i) = \sum_{i=1}^{n} p(1 - p) = np(1 - p)$$

Immaginiamo adesso di avere due variabili aleatorie binomiali:

$X_1 \sim P(n_1, p)$

$X_2 \sim P(n_2, p)$

$$X_1 = \sum_{i=1}^{n_1} X_{1,i}$$

$$X_2 = \sum_{i=1}^{n_2} X_{2,i}$$

$$X_1 + X_2 = \sum_{i=1}^{n_1} X_{1,i} + \sum_{i=1}^{n_2} X_{2,i}$$

Possiamo farlo dato che sono distribuite entrambe come una bernoulliana di parametro p ($\sim B(p)$).
Definiamo ora:

$Y_i = X_{1,i}$ per $i = 1, \dots, n_1$

$Y_{i+n_1} = X_{2,i}$ per $i = 1, \dots, n_2$

$$= \sum_{i=1}^{n_1+n_2} Y_i$$

Quindi

$X_1 + X_2 \sim P(n1 + n2, p)$

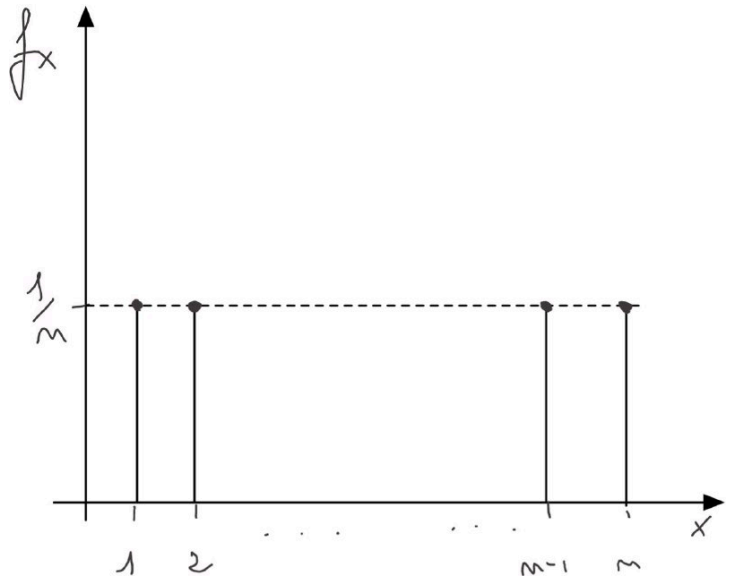
Famiglia di distribuzione uniforme discreta

$X \sim U(n)$

$D_X = \{1, \dots, n\}$

$f_X(x) = \frac{1}{n} I_{D_X}(x)$

$n \in \aleph$



$F_X(x) = P(X \leq x) = \sum_{i \leq x} P(X = i)$

$x \in \aleph$ e $x \leq n$

$$= \sum_{i=1}^{x} P(X = i) = \sum_{i=1}^{x} f_X(i) = \sum_{i=1}^{x} \frac{1}{n} = \frac{x}{n}$$

Però $F_X(x)$ ha come dominio tutto $\Re$, quindi le assunzioni $x \in \aleph$ e $x \leq n$ non vanno considerate.

Consideriamo quindi la formula senza le assunzioni:

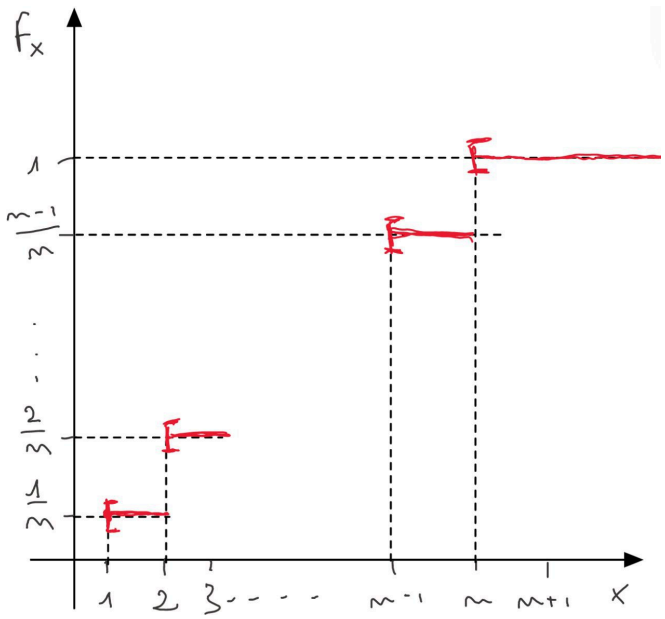
$$\sum_{i=1}^{x\,[intero\ inferiore]} P(X = i) = \frac{x\,[intero\ inferiore]}{n}$$

Se x è negativo ottengo 0

Se x supera n concettualmente dovrei ottenere sempre 1, ma con questa formula ottengo numeri maggiori.

Per sistemare utilizzo funzioni indicatrici:

$$= \frac{|x|[intero\ inferiore]}{n} I_{[1,n)}(x) + I_{[n,+\infty)}(x)$$



Quindi:

$F_X(x) = \frac{\lfloor x \rfloor [intero\ inferiore]}{n}$ se $1 \leq x < n$, 0 se $x < 1$ e 0 altrimenti

Calcoliamo il valore atteso:

$$E(X) = \sum_{x=1}^{n} x f_X(x) = \sum_{x=1}^{n} x \frac{1}{n} = \frac{1}{n} \sum_{x=1}^{n} x = \frac{1}{n} \cdot \frac{n(n+1)}{2} = \frac{n+1}{2}$$

Varianza:

$$Var(X) = E(X^2) - E(X)^2$$

$$E(X^2) = \sum_{x=1}^{n} x^2 \frac{1}{n} = \frac{1}{n} \sum_{x=1}^{n} x^2$$

Ricordiamo che:

$$\sum_{i=1}^{n} i^2 = \frac{i(i+1)(2i+1)}{6}$$

Quindi:

$$= \frac{1}{n} \cdot \frac{n(n+1)(2n+1)}{6} = \frac{(n+1)(2n+1)}{6}$$

$$Var(X) = \frac{(n+1)(2n+1)}{6} - \frac{(n+1)^2}{4} = \frac{n+1}{2}\left(\frac{2n+1}{3} - \frac{n+1}{2}\right) = \frac{n+1}{2}\left(\frac{4n+2-3n-3}{6}\right)$$

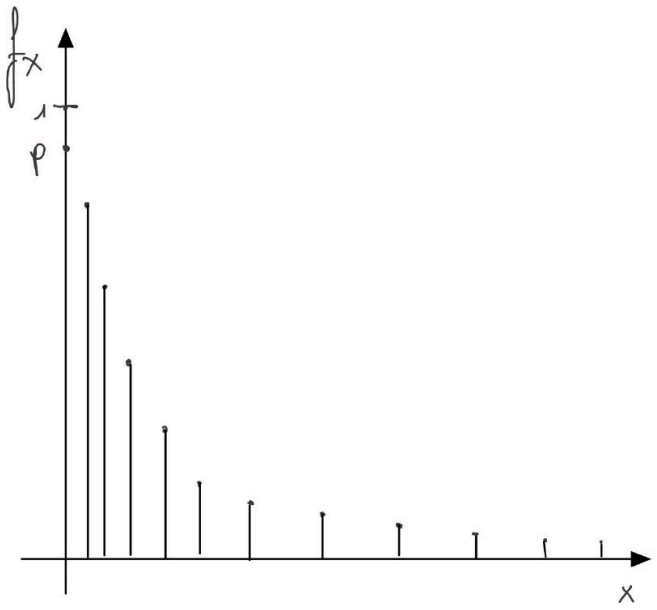
$$= \frac{n+1}{2}\left(\frac{n-1}{6}\right) = \frac{n^2-1}{12}$$

Fino ad adesso abbiamo analizzato variabili aleatorie discrete con supporto finito.
Primo esempio di famiglia di distribuzione con supporto infinito:
Famiglia geometrica
Similitudine informatica:

- Modello binomiale
    - Ciclo for
- Modello geometrico
    - Ciclo while

Concettualmente, ripeto un esperimento bernoulliano in condizione di indipendenza fino a quando non si verifica una condizione un determinato numero di volte.
Quindi, conto quante volte ripeto l'esperimento fallendo prima di ottenere un successo.
$X \sim G(p)$
$p \in (0, 1]$
Il parametro non può essere uguale a zero perchè avrei infinito come risultato, mentre nel caso fosse 1 avremo come risultato sempre 0.
$X = i$
Se ho fatto la ripetizione dell'esperimento bernoulliano ottenendo i insuccessi seguiti da 1 successo ($i \in \aleph \cup \{0\}$ [$nel\ caso\ ottenga\ un\ successo\ al\ primo\ tentativo$]).

$f_X(x) = P(X = x)$ [$x\ fallimenti\ e\ x + 1\ successi$]
$= P(R_1[esito\ prima\ ripetizione\ dell'esperimento] = F \wedge R_2 = F \wedge \ ...$
$\wedge R_x = F \wedge R_{x+1} = S)$
Dato che sono eventi indipendenti:
$= P(R_1 = F\ [= 1 - p]) \cdot \ ... \ \cdot P(R_x = F) \cdot P(R_{x+1} = S\ [= p])$
$= (1 - p)^x p \cdot I_{\aleph \cup \{0\}}(x)$
Sappiamo per definizione che sarà sempre positiva, ma adesso controlliamo che sommi ad 1.
$\sum_{x=0}^{+\infty} f_X(x) = \sum_{x=0}^{+\infty} (1 - p)^x p = p \sum_{x=0}^{+\infty} (1 - p)^x$
Ricordiamo la serie geometrica:
$\sum_{i=0}^{+\infty} \alpha^i = \frac{1}{1-\alpha}$
Converge quando $-1 < \alpha < 1$
Quindi:
$= p \cdot \frac{1}{1-(1-p)} = 1$



$$P(X > n) = \sum_{x>n} p(1-p)^x = \sum_{x=n+1}^{+\infty} p(1-p)^x = p(1-p)^{n+1} \sum_{x=n+1}^{+\infty} (1-p)^{x-n-1}$$

Definiamo:

$y := x - n - 1$

$x = n + 1 \qquad y = 0$

$x = +\infty \qquad y = +\infty$

Quindi:

$$= p(1-p)^{n+1} \sum_{y=0}^{+\infty} (1-p)^y = p\frac{1}{p}(1-p)^{n+1} = (1-p)^{n+1}$$

$$F_X(x) = P(X \leq x) = 1 - P(X > x) = 1 - (1-p)^{x+1}$$

Funziona con i numeri interi, altrimenti devo troncarlo all'intero inferiore.

$$P(X \geq n) = P(X > n-1) = (1-p)^n$$

$$P(X \geq i + j | X \geq i)$$

Ho avuto i fallimenti, qual'è la probabilità che ce ne siano altri j?

$$= \frac{P(X \geq i+j, X \geq i)}{P(X \geq i)} = \frac{P(X \geq i+j)}{P(X \geq i)} = \frac{(1-p)^{i+j}}{(1-p)^i} = (1-p)^j = P(X \geq j)$$

Questo risultato ci indica l'indipendenza degli esperimenti precedenti da quelli successivi.

Questa proprietà si chiama "assenza di memoria".

Calcoliamo il valore atteso

$$E(X) = \sum_{x=0}^{+\infty} x f_X(x) \; [= (1-p)^x p]$$

Inciso:

$$\sum_{i=0}^{+\infty} i\alpha^{i} = \alpha \sum_{x=0}^{+\infty} i\alpha^{i-1}$$

$$\frac{d}{dx}\alpha^{i} = i\alpha^{i}$$

$$= a\sum_{x=0}^{+\infty} \frac{d}{dx}\alpha^{i} = a\frac{d}{dx}\sum_{x=0}^{+\infty} \alpha^{i} = \alpha\frac{d}{dx}(1-\alpha)^{2} = a(-1)(1-\alpha)^{-2}(-1)$$

$$= \frac{\alpha}{(1-\alpha)^{2}}$$

Quindi:

$$= p\sum_{x=1}^{+\infty} x(1-p)^{x} = p\frac{1-p}{(1-(1-p))^{2}} = \frac{1-p}{p}$$

Calcoliamo la varianza:

$$Var(X) = E(X^{2}) - E(X)^{2}$$

$$E(X^{2}) = \sum_{x=0}^{+\infty} x^{2}(1-p)^{x}p = p\sum_{x=0}^{+\infty} x^{2}(1-p)^{x}$$

Inciso:

$$\frac{d}{dp}(-i(1-p)^{i}) = -i(-1)(i)(1-p)^{i-1} = i^{2}(1-p)^{i-1}$$

Quindi:

$$= p(1-p)\sum_{x=0}^{+\infty} x^{2}(1-p)^{x-1} = p(1-p)\sum_{x=0}^{+\infty} \frac{d}{dp}(-x(1-p)^{x})$$

$$= -p(1-p)\frac{d}{dp}\sum_{x=0}^{+\infty} x(1-p)^{x} = -p(1-p)\frac{d}{dp}\cdot\frac{1-p}{p^{2}}$$

$$= p(1-p)\frac{p^{2}+(1-p)2p}{p^{4}} = (1-p)\frac{p+2(1-p)}{p^{2}} = (1-p)\frac{2-p}{p^{2}}$$

$$Var(X) = E(X^{2}) - E(X)^{2} = \frac{(1-p)(2-p)}{p^{2}} - \frac{(1-p)^{2}}{p^{2}} = \frac{(1-p)}{p^{2}}(2-p-1+p) = \frac{1-p}{p^{2}}$$

Esercizio:

$P(ammalato\ in\ un\ mese) = p = 0.1$

$X = numero\ di\ influenze$

Modello binomiale

12 mesi

Esperimento bernoulliano: in un mese mi sono ammalato o meno.

Le ripetizioni sono indipendenti.

$P(andare\ dal\ medico) = P(X \geq 3)$

$X \sim P(12, p)$

$P(X = 3) = (12\ 3)p^{3}(1-p)^{9} \approx 0.085$

$$P(X \geq 3) = 1 - P(X < 3) = 1 - \sum_{x=0}^{2} p^x (1-p)^{12-x} \approx 0.11 = p'$$

Qual'è la probabilità di andare dal medico per la prima volta tra 5 anni?
$Y = numero\ di\ anni\ senza\ medico\ \sim G(p')$
Cambia l'esperimento bernoulliano ma rimane comunque la condizione d'indipendenza.

$$P(medico\ fra\ 5\ anni) = P(Y = 4) = (1-p')^4 p' \approx 0.069$$

$P(non\ vado\ dal\ medico\ l'anno\ prossimo\ essendoci\ andato\ l'anno\ scorso)$
$= P(Y \geq 3 | Y \geq 2)$
Per assenza di memoria:
$P(Y \geq 1) = 1 - P(Y = 0) = 1 - p'$

Lezione del 9 Maggio 2023

Lezione 17

Nella lezione precedente abbiamo introdotto una prima distribuzione discreta con supporto infinito.
Adesso vediamone un altro:
Prendiamo come punto di partenza la funzione di massa di probabilità di una funzione binomiale:

$$f_X(x) = (x\ n) p^x (1-p)^{n-x}$$

Cosa succede se n cresce all'infinito?

- n cresce
- $np = \lambda$ costante
- Quindi p decresce

$$p = \frac{\lambda}{n}$$

$$\frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}$$

$$= \frac{n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot (n-x+1)[x\ fattori](n-x)!}{x![x\ fattori](n-x)!} \cdot \frac{\lambda^x}{n^x} \cdot \left(1 - \frac{\lambda}{n}\right)^{n-x}$$

$$= \frac{n}{n} \cdot \frac{n-1}{n} \cdot \ldots \cdot \frac{n-x+1}{n} \cdot \frac{\lambda^x}{x!} \cdot \frac{\left(1-\frac{\lambda}{n}\right)^n}{\left(1-\frac{\lambda}{n}\right)^x}$$

Per n che tende a infinito:

- Le seguenti frazioni

$$\frac{n}{n} \cdot \frac{n-1}{n} \cdot \ldots \cdot \frac{n-x+1}{n}$$

Tenderanno tutte a 1

- $(1 - \frac{\lambda}{n})^x$

Tenderà ad 1

- $(1 - \frac{\lambda}{n})^n$

Tenderà a $e^{-\lambda}$
Dato che:

$$e = \lim_{n \to +\infty} (1 + \frac{1}{n})^n$$

e $\lim_{n \to +\infty} (1 + \frac{\alpha}{n})^n = e^{\alpha}$

Quindi in totale ottengo:

$$= e^{-\lambda} \frac{\lambda^x}{x!}$$

$$f_X(x) = e^{-\lambda} \frac{\lambda^x}{x!} \cdot I_{\aleph \cup \{0\}}(x)$$

Estensione della binomiale con ripetizioni che tendono ad infinito.
Come controllo che sia una funzione di massa di probabilità?

1) Deve sommare ad 1
2) Sempre positiva

Sappiamo che non può essere infinita, controlliamo il primo criterio:

$$\sum_{x=0}^{+\infty} e^{-\lambda} \frac{\lambda^x}{x!} = e^{-\lambda} \sum_{x=0}^{+\infty} \frac{\lambda^x}{x!}$$

Sviluppo di McLaurin

$$\sum_{x=0}^{+\infty} \frac{\lambda^x}{x!} = e^{\lambda}$$

Quindi il totale è $= 1$

Modello in variabile di Poisson
$x \sim P(\lambda)$
Calcoliamone il valore atteso:

$$E(X) = \sum_{x=0}^{+\infty} x e^{-\lambda} \frac{\lambda^x}{x!}$$

Dato che per $x = 0$ il suo apporto alla somma è nullo, posso considerare la sommatoria a partire da 1

$$= e^{-\lambda} \sum_{x=1}^{+\infty} \frac{\lambda^x}{(x-1)!} = e^{-\lambda} \lambda \sum_{x=1}^{+\infty} \frac{\lambda^{x-1}}{(x-1)!}$$

$$Y = X - 1$$

$$= e^{-\lambda} \lambda \sum_{y=0}^{+\infty} \frac{\lambda^y}{y!} = e^{-\lambda} \lambda e^{\lambda} = \lambda$$

Calcoliamo adesso la varianza attraverso il momento secondo:

$$E(X^2) = \sum_{x=0}^{+\infty} x^2 e^{-\lambda} \frac{\lambda^x}{x!} = e^{-\lambda} \sum_{x=1}^{+\infty} x \frac{\lambda^x}{(x-1)!}$$

$$= e^{-\lambda} \lambda \sum_{x=1}^{+\infty} x \frac{\lambda^{x-1}}{(x-1)!}$$

$$Y = X - 1$$

$$= e^{-\lambda} \lambda \sum_{y=0}^{+\infty} (y+1) \frac{\lambda^y}{y!}$$

$$= \lambda \sum_{y=0}^{+\infty} (y+1) \cdot e^{-\lambda} \frac{\lambda^y}{y!} \; [= f_Y(y)]$$

La sommatoria equivale quindi a $E(Y+1) = E(Y) + 1$ con $Y \sim P(\lambda)$

$$= \lambda(E(Y) + 1) = \lambda(\lambda + 1) = \lambda^2 + \lambda$$

Quindi:

$$Var(X) = E(X^2) - E(X)^2 = \lambda^2 + \lambda - \lambda^2 = \lambda = E(X)$$

Esempio:
Una compagnia di assicurazioni controlla ogni giorno se tutti i clienti abbiano fatto un incidente o meno.
In media riceve 5 richieste di rimborso al giorno.

$X = numero\ di\ richieste\ al\ giorno$

$E(X) = 5$

$X \sim P(\lambda)$

$\lambda = E(X) = 5$

Quale frazione delle giornate vedrà arrivare meno di 3 richieste?

$$P(X < 3) = P(X = 0 \cup X = 1 \cup X = 2) = \sum_{x=0}^{2} f_X(x)$$

$$= e^{-\lambda}\left(\frac{\lambda^0}{0!} \cdot \frac{\lambda^1}{1!} \cdot \frac{\lambda^2}{2!}\right) = e^{-5}\left(1 + 5 + \frac{25}{2}\right) \approx 0.1247$$

Qual'è la probabilità che in 5 giorni, in 3 di essi arrivino esattamente 4 richieste?

$$P(4\ richieste\ in\ un\ giorno) = P(X = 4) = e^{-5} \frac{5^4}{4!} \approx 0.1755 = p$$

$P(3\ su\ 5\ giorni\ ho\ 4\ richieste)$

Distribuzione binomiale

$Y = B(5, p)$

$$P(X = 3) = (5\ 3) p^3 (1-p)^2 \approx 0.0367$$

Quando ha senso approssimare una binomiale ad una distribuzione di Poisson?
Vediamo di quanto differiscono di approssimazione i due modelli.
Esempio:

Macchinario con $P(difettoso) = 0.1$
$X = numero\ di\ pezzi\ difettosi\ su\ 10$
$X \sim B(10,\ 0.1)$ $\quad$ $X' \sim P(np = 1)$
$P(Al\ più\ un\ difettoso)$
$P(X \leq 1) = (10\ \ 0)0.1^{0} \cdot 0.9^{10} + (10\ \ 1)0.1^{1} \cdot 0.9^{9} \approx 0.7361$
$P(X' \leq 1) = e^{-1}(\frac{1^{0}}{0!} + \frac{1^{1}}{1!}) \approx 0.7358$

Come posso vedere di quanto differiscono i risultati ottenuti?
Sovrappongo i grafici o faccio un rapporto per ottenere un risultato matematico più rigoroso.

Ultimo modello delle famiglia binomiali discrete, poco usato
Modello di estrazione senza reimmissione.
Se ci fosse la reimmissione avremmo un classico modello binomiale.
Nel nostro caso invece abbiamo il modello Ipergeometrico.
Tre parametri:

1) $N = numero\ di\ funzionanti$
2) $M = numero\ di\ difettosi$
3) $n = numero\ di\ estrazioni$

La variabile aleatoria considera il numero di oggetti funzionanti su $N + M$ elementi dopo n estrazioni.
$X \sim H(N,\ M,\ n)$
$f_{X}(x) = P(X = x)$
$= \frac{numero\ di\ modi\ in\ cui\ posso\ estrarre\ n\ oggetti\ con\ x\ funzionanti}{n\ estrazioni\ possibili}$
$= \frac{(N\ \ x)(M\ \ n-x)}{(N+M\ \ n)} \cdot I_{?}(x)$

Esempio:
$N = 1$ $\quad$ $M = 5$ $\quad$ $n = 2$
$(1\ \ 2) \leftarrow$ impossibile per interpretazione combinatoria impossibile $= 0$
$\frac{1!}{2!(-1)!}$

Useremo lo stesso come supporto
$I_{\{0,\ 1,\ \dots\ ,\ n\}}(x)$
Stando però attenti durante i calcoli.

$\forall i = 1,\ 2,\ \dots\ ,\ n$
Per le singole estrazioni
$X_{i} = 1$ se l'i-esimo oggetto è funzionante, 0 altrimenti
$X_{i} \sim B(p)$

$p = P(i\ esimo\ oggetto\ funzionante) = \frac{N}{N+M}$

Approccio composizionale:

$E(X) = E(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} E(X_i) = np = \frac{nN}{N+M}$

$Var(X_i) = p(1-p) = \frac{N}{N+M} \cdot \frac{M}{N+M}$

$Var(X) = Var(\sum_{i=1}^{n} X_i)$

Questa formula funziona solo se le variabili sono indipendenti, in questo caso non lo sono quindi calcolo la covarianza

Per $i \neq j$

$Cov(X_i, X_j) = E(X_i X_j) - E(X_i)E(X_j)$

$X_i X_j = 1$ se $X_i = 1,\ X_j = 1$, 0 altrimenti

$= P(X_i = 1,\ X_j = 1) - (\frac{N}{N+M})^2$

Non sono indipendenti, non posso fare il prodotto

$= P(X_i = 1 | X_j = 1)P(X_j = 1) - (\frac{N}{N+M})^2$

$= \frac{N-1}{N-1+M} \cdot \frac{N}{N+M} - (\frac{N}{N+M})^2$

$= \frac{N}{N+M}(\frac{N-1}{N-1+M} - \frac{N}{N+M})$

$= \frac{N}{N+M}(\frac{(N-1)(N+M)-N(N+M-1)}{(N+M-1)(N+M)})$

$= \frac{N}{N+M}(\frac{N^2+NM-N-M-N^2-NM+N}{(N+M-1)(N+M)}) = \frac{-NM}{(N+M)^2(N+M-1)} < 0$

Quindi:

$Var(X) = Var(\sum_{i=1}^{n} X_i)$

$= \sum_{i=1}^{n} Var(X_i)\ [n\ termini] + \sum_{i \neq j} Cov(X_i, X_j)\ [n(n-1)\ termini]$

$= nVar(X_i) + n(n-1)Cov(X_i, X_j)$

$= n\frac{N}{N+M} \cdot \frac{M}{N+M} - n(n-1)\frac{NM}{(N+M-1)(N+M)^2}$

$= \frac{nNM}{(N+M)^2}(1 - \frac{n-1}{N+M-1}) = np(1-p)(1 - \frac{n-1}{N+M-1})$

Notiamo che, per $N + M$ che tende all'infinito la formula diventa

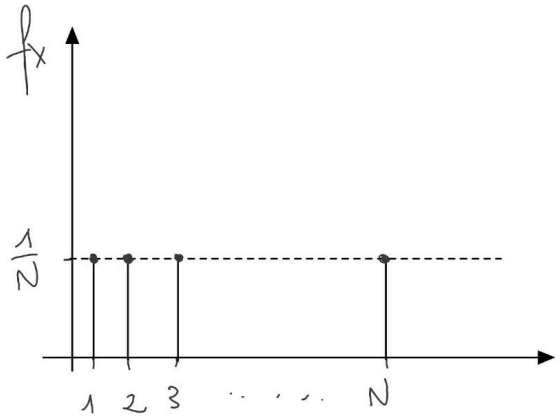
$= np(1-p)$

Quindi si riduce ad una distribuzione binomiale.

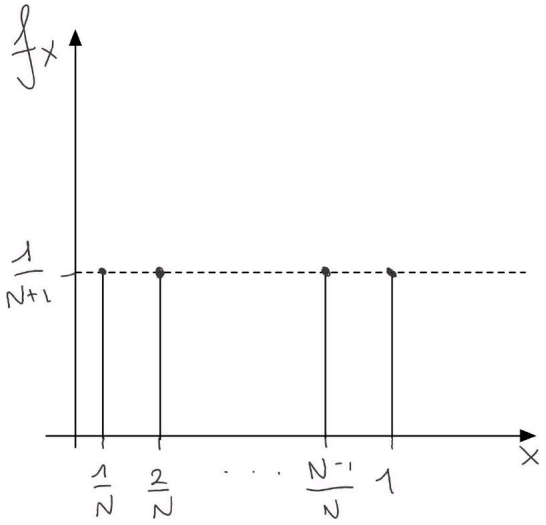
Lezione dell'11 Maggio 2023

Lezione 18

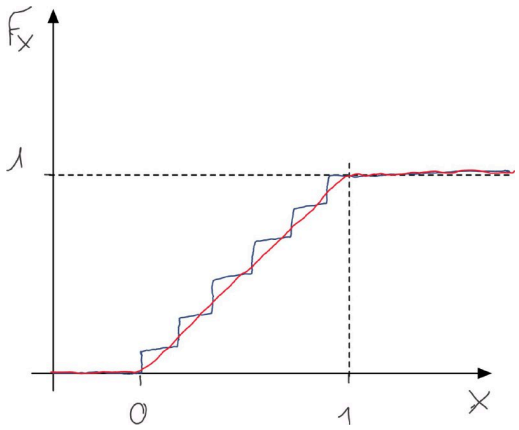
Oggi inizieremo a parlare di variabili aleatorie continue.
come vengono introdotte?
Se ad esempio consideriamo una distribuzione uniforme discreta su un insieme di numeri che vanno da zero a N.



Immaginiamo adesso di aumentare N progressivamente, anche con limite all'infinito, avremo una funzione di massa di probabilità con valori nulli.
Se invece considerassimo N valori equispaziati nell'intervallo tra 0 e 1.



Se in questo caso, facessimo trandere N all'infinito e considerassimo la funzione di ripartizione:



Avremo degli scalini di altezza $\frac{1}{N+1}$, con N che tende a $+\infty$ essi si assottiglieranno sempre di più (passando dal grafico in blu a quello in rosso).

$P(X = \frac{1}{2}) = 0$

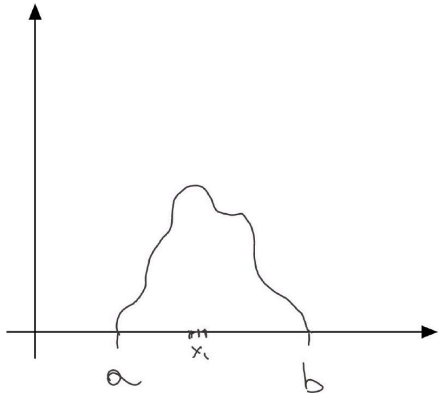
Richieste di una funzione di massa di probabilità:

1) Non deve essere negativa
2) $\lim_{x \to +\infty} F_X(x) = 1$
3) Deve essere continua da destra

Però nel nostro caso, non avendo più dei gradini definiti, non ho più una funzione di massa di probabilità.
Quindi il supporto di queste variabili aleatorie non sarà più un insieme di valori ma un intervallo continuo.
La probabilità in un punto adesso non si ottiene più alla stessa maniera.
La probabilità dei singoli elementi sarà nulla, avrà quindi più senso calcolare la probabilità per range di valori.
Nel nostro caso andremo a misurare la densità.

Funzione di densità di probabilità:

1) Funzione continua
2) In ogni punto, rispetto al suo intorno, ho la sua probabilità.



= Probabilità che un valore si avvicini a x_1

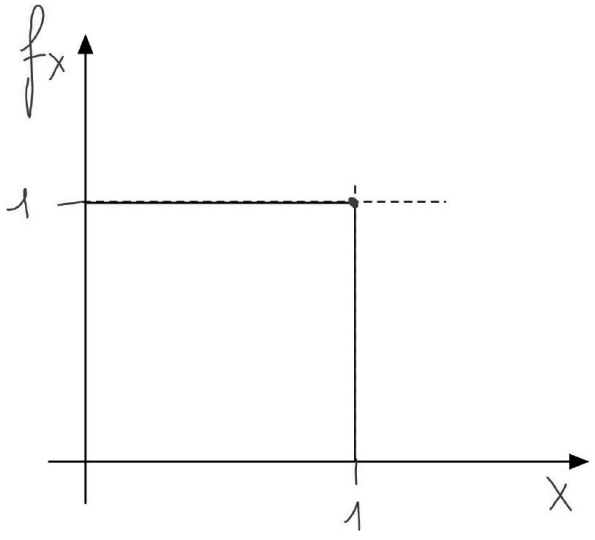
Per l'esempio della distribuzione precedente:
$f_X(x) = kI_{[0,\,1]}(x)$
$P(0 \leq X \leq 1) = 1$

Per un intorno generico di a:
$\int_{a-\frac{\varepsilon}{2}}^{a+\frac{\varepsilon}{2}} f_X(x)dx \approx \varepsilon f_X(a)$
$= P(a - \frac{\varepsilon}{2} \leq X \leq a + \frac{\varepsilon}{2})$

$P(a \leq X \leq b) = \int_a^b f_X(x)dx$

Quindi, in generale:
$\int_{-\infty}^{+\infty} f_X(x)dx = 1$
$= \int_0^1 kdx = k\int_0^1 1dx = 1$

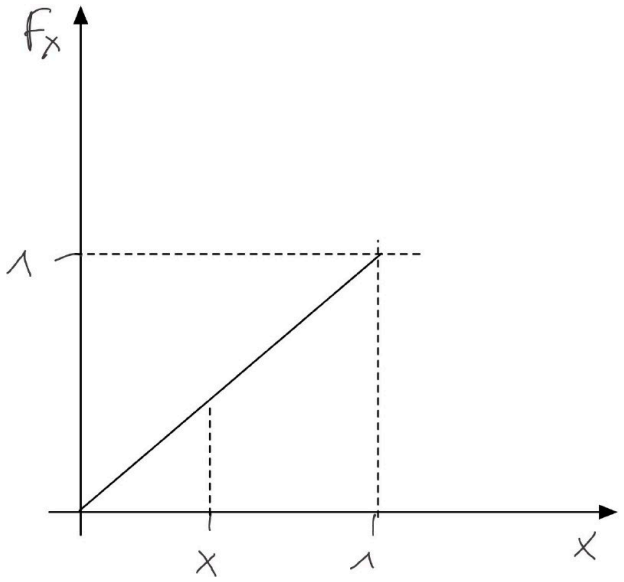
Allora ho $k = 1$ e quindi per questo particolare tipo di distribuzione:
$f_X(x) = I_{[0,\,1]}(x)$



$P(a \leq X \leq b) = \int_a^b 1dx = b - a$

Modello delle distribuzioni uniformi continue
$X \sim U([0,\,1])$

$F_X(x) = P(X \leq x)$



$= \int_0^x f_X(x)dx = \int_0^x 1dx = x$

$P(a < x \leq b) = F_X(b) - F_X(a) = b - a$

Funziona comprendendo o meno gli estremi, dato che hanno densità nulla.

$F_X(x) = \int_{-\infty}^{x} f_X(v)dv$

$\lim_{x \to -\infty} F_X(x) = 0$

Per il teorema fondamentale del calcolo integrale, se derivo la funzione di ripartizione ottengo la funzione di densità.

$E(X) = \int_{-\infty}^{+\infty} x f_X(x)dx$

$E(g(X)) = \int_{-\infty}^{+\infty} g(x) f_X(x)dx$

Per $x \geq 0 \Rightarrow E(X) = \int_0^{+\infty} 1 - F_X(x)dx$

Esempio:

$f_X(x) = c(4x - 2x^2)$ se $0 \leq x \leq 2$, 0 altrimenti

$P(X > 1) =?$

Prima ricaviamo c

$$\int_0^2 c(4x - 2x^2)dx = 1$$

$$c\int_0^2 (4x - 2x^2)dx = 1$$

$$c(2x^2[_0^2 - \frac{2}{3}x^3[_0^2) = 1$$

$$1 = c(8 - \frac{16}{3})$$

$$1 = c\frac{8}{3} \qquad\qquad c = \frac{3}{8}$$

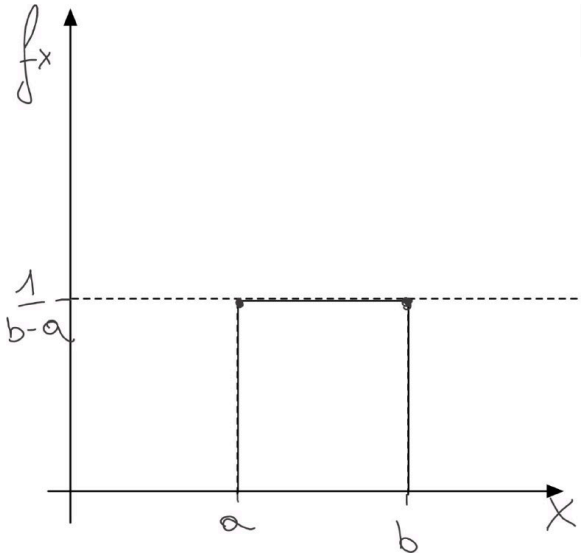
$$f_X(x) = \frac{3}{8}(4x - 2x^2)I_{[0,2]}(x)$$

$$P(X > 1) = \int_1^2 \frac{3}{8}(4x - 2x^2)dx \approx 1/2$$

Modello uniforme continuo
Modello con densità uniforme in un certo intervallo

$$X \sim U([a, b])$$

$$f_X(x) = \frac{1}{b-a}I_{[a,b]}(x)$$



$$\int_{-\infty}^{+\infty} f_X(x)dx = \int_a^b \frac{1}{b-a}dx = \frac{1}{b-a}\int_b^a 1dx = \frac{1}{b-a}(b - a) = 1$$

$$P(X \le x) = \int_a^x f_X(v)dv = \int_a^x \frac{1}{b-a}dv = \frac{1}{b-a}\int_a^x 1dv$$

$$= \frac{x-a}{b-a}$$

$$F_X(x) = \frac{x-a}{b-a}I_{[a,b)}(x) + I_{[b,+\infty)}(x)$$

Calcoliamone il valore atteso

$$E(X) = \int_a^b x f_X(x)dx = \int_a^b x \frac{1}{b-a} dx = \frac{1}{b-a} \int_a^b x dx$$
$$= \frac{1}{b-a} \cdot \frac{x^2}{2}\Big[_a^b = \frac{1}{b-a}\left(\frac{b^2}{2} - \frac{a^2}{2}\right) = \frac{1}{b-a}\left(\frac{(b-a)(b+a)}{2}\right) = \frac{b+a}{2}$$

Calcoliamo adesso la varianza partendo dal momento secondo:

$$E(X^2) = \int_a^b x^2 f_X(x)dx = \frac{1}{b-a} \int_a^b x^2 dx = \frac{1}{b-a}\left(\frac{x^3}{3}\Big[_a^b\right)$$
$$= \frac{1}{b-a}\left(\frac{b^3-a^3}{3}\right) = \frac{(b-a)(b^2+ab+a^2)}{(b-a)3} = \frac{a^2+ab+b^2}{3}$$

$$Var(X) = E(X^2) - E(X)^2 = \frac{b^2+ab+a^2}{3} - \frac{(b+a)^2}{4}$$
$$= \frac{b^2+ab+a^2}{3} - \frac{b^2+2ab+a^2}{4} = \frac{4b^2+4ab+4a^2-3b^2-6ab-3a^2}{12}$$
$$= \frac{a^2+b^2-2ab}{12} = \frac{(b-a)^2}{12}$$

Esempio:
Un autobus passa dalle 7:00 alle 7:30 due volte alle 7:15 e alle 7:30.
$P(aspetto\ meno\ di\ 5\ minuti)$
$= P(7{:}10 \leq X \leq 7{:}15 \vee 7{:}25 \leq X \leq 7{:}30)$
$X = numero\ di\ minuti\ dopo\ le\ 7\ in\ cui\ arrivo\ alla\ fermata$
$P(\{10 \leq X \leq 15\} \cup \{25 \leq X \leq 30\})$
Sono eventi disgiunti, quindi:
$= P(10 \leq X \leq 15) + P(25 \leq X \leq 30)$
$= F_X(15) - F_X(10) + F_X(30) - F_X(25)$
Dato che:
$f_X(x) = \frac{1}{30} I_{[0,\,30]}(x)$
e $F_X(x) = \frac{x}{30} I_{[0,\,30)}(x) + I_{[30,\,+\infty)}(x)$

$= 15/30 - 10/30 + 30/30 - 25/30 = 1/3$

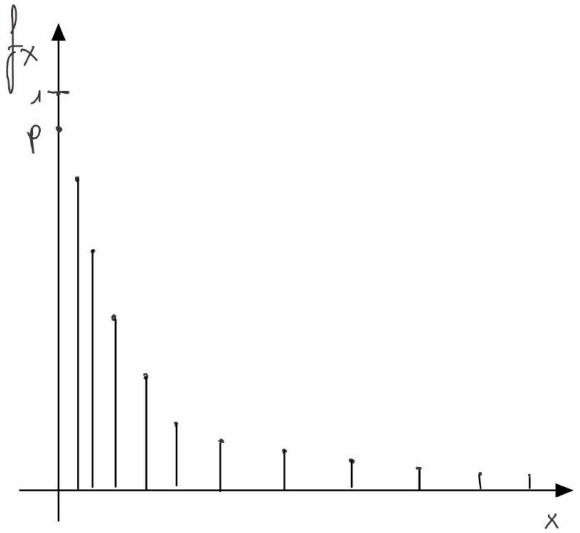
$P(attendere\ almeno\ dodici\ minuti)$
$P(\{0 \leq X \leq 3\} \cup \{15 < X \leq 18\}) = 6/30 = 1/5$

Quantili
Possiamo declinare il concetto di quantile anche per le variabili aleatorie?
q-esimo
$q \in [0,\,1]$ di una X variabile aleatoria è quella $x \in X$ tale che $P(X \leq x) = q$
$F_X(x_q) = q$

$x_q = F_X^{-1}(q)$
Quando però possiamo invertire una funzione di ripartizione?
Nel caso continuo è sempre possibile, per lo zero e l'uno si prendono le specificazioni minori e maggiori.

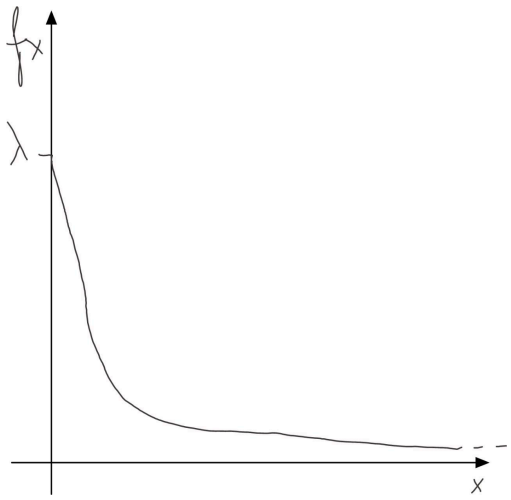
Il fatto di introdurre i quantili all'interno delle variabili aleatorie ci permette di calcolarne la mediana.
Nel caso di variabili aleatorie continue, la mediana è uguale al valore atteso.

Altro modello di distribuzione
Partiamo dalla distribuzione geometrica:



Prendiamo le nostre X_i, possiamo pensare di moltiplicarle per p/λ con $p \in (0, 1)$ e $\lambda > 0$.
Dato che il supporto è infinito, in questo caso sto soltanto rendendo la distribuzione "meno rarefatta".
Inoltre, per p che tende a zero otterrei una curva.
$f_X(x) = \lambda e^{-\lambda x} I_{[0, +\infty)}(x)$

Distribuzione esponenziale
$X \sim E(\lambda)$



Funzione non negativa per definizione, controlliamo che l'integrale sommi ad 1:

$$\int_0^{+\infty} \lambda e^{-\lambda x} dx = \int_0^{+\infty} e^{-\lambda x} \lambda dx$$

Inciso:

$$(e^{-\lambda x})' = e^{-\lambda x} \cdot (-\lambda)$$

Quindi:

$$= -\int_0^{+\infty} -e^{-\lambda x} \lambda dx = -e^{-\lambda x}\Big[_0^{+\infty} = 0 - (-e^0) = 1$$

$$F_X(x) = \int_{-\infty}^{x} \lambda e^{-\lambda u} du$$

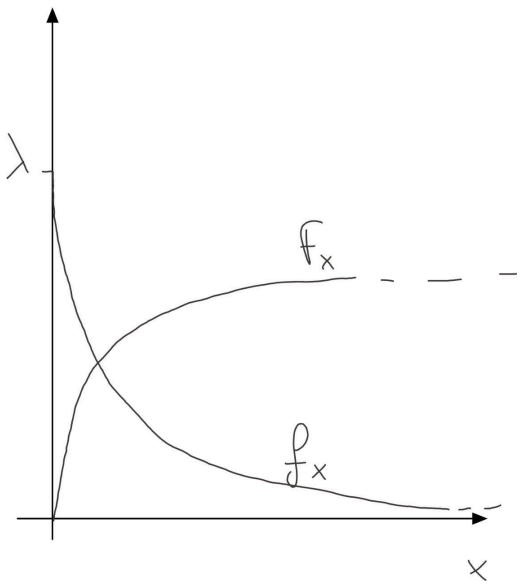
Con:

$$v = \lambda u$$

$$dv = \lambda du$$

$$= \int_0^{\lambda x} e^{-v} dv = -e^{-v}\Big[_0^{\lambda x} = -e^{-\lambda x} + 1$$

$$= 1 - e^{-\lambda x}$$



Calcoliamone adesso il valore atteso:

$$E(X) = \int_0^{+\infty} x[f]\lambda e^{-\lambda x}[g']dx$$

$$g'(x) = \lambda e^{-\lambda x}$$

$$g(x) = -e^{-\lambda x}$$

$$= -xe^{-\lambda x}[_0^{+\infty}[= 0] - (-\int_0^{+\infty} e^{-\lambda x}dx) = 1/\lambda \int_0^{+\infty} \lambda e^{-\lambda x}dx[F_X = 1] = 1/\lambda$$

Adesso calcoliamo anche la varianza partendo dal momento secondo:

$$E(X^2) = \int_0^{+\infty} x^2[f]\lambda e^{-\lambda x}dx$$

$$= (-x^2 e^{-\lambda x})[_0^{+\infty}[= 0] + 2\int_0^{+\infty} xe^{-\lambda x}dx$$

$$= 2/\lambda \int_0^{+\infty} x\lambda e^{-\lambda x}[f_X]dx = 2/\lambda E(X) = \frac{2}{\lambda^2}$$

Quindi:

$$Var(X) = \frac{2}{\lambda^2} - \frac{1}{\lambda^2} = \frac{1}{\lambda^2}$$

Questa distribuzione gode di proprietà interessanti:

- Proprio come quella discreta geometrica, possiede assenza di memoria

$$P(X > s + t|X > t) = P(X > s)$$

$$\frac{P(X>s+t, X>t)}{P(X>t)} = P(X > s)$$

$P(X > s + t) = P(X > s)P(X > t)$

$P(X > t) = 1 - F_X(t) = e^{-\lambda t}$

$P(X > s + t) = 1 - F_X(s + t) = e^{-\lambda(s+t)} = e^{-\lambda t} \cdot e^{-\lambda s}$

- $X \sim E(\lambda)$, introduciamo $c > 0$ e $Y = cX$

$F_Y(x) = P(Y \leq x) = P(cX \leq x) = P(X \leq \frac{x}{c}) = F_X(\frac{x}{c}) = 1 - e^{-\lambda\frac{x}{c}}$

$\lambda' = \frac{\lambda}{c}$

$F_X(\frac{\lambda}{c}) = 1 - e^{-\lambda' x}$

Quindi ho dimostrato che $Y \sim E(\lambda')$

- Immaginiamo di voler sapere il valore più piccolo tra alcune variabili aleatorie

$X_1, \dots, X_n \qquad Y = min(X_i)$

$X_i \sim E(\lambda_i)$

$P(Y > x) = P(min(X_i) > x)$

$min(X_i) > x \leftrightarrow \forall i \; X_i > x$

$= P(\bigcap_{i=1}^{n} \{X_i > x\}) = \prod_{i=1}^{n} P(X_i > x)$

$F_Y(x) = 1 - \prod_{i=1}^{n} P(X_i > x) = 1 - \prod_{i=1}^{n} 1 - F_{X_i}(x)$

Questa regola vale per tutte le variabili aleatorie esponenziali, nel caso delle geometriche:

$= 1 - \prod_{i=1}^{n} 1 - e^{-\lambda_i x} = 1 - e^{-(\sum_{i=1}^{n} \lambda_i)x}$

$\lambda' = \sum_{i=1}^{n} \lambda_i$

$= 1 - e^{-\lambda' x}$

$Y \sim E(\lambda') \sim E(\sum_{i=1}^{n} \lambda_i)$

Esempio:

$X = anni\ di\ funzionamento\ di\ un\ macchinario$

$X \sim E(\frac{1}{8})$

$P(funziona\ per\ altri\ 10\ anni?) = P(X > t + 10 | X > t)$

Per assenza di memoria:

$P(X > 10) = e^{-10/8} \approx 0.2865$

Lezione del 16 Maggio 2023

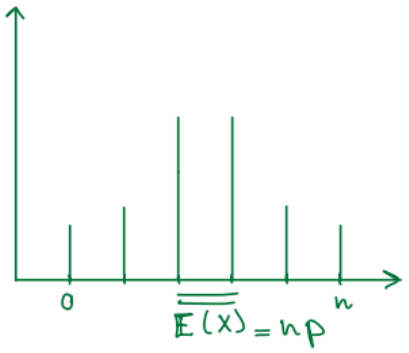
Lezione 19

Nella scorsa lezione abbiamo iniziato ad analizzare variabili aleatorie con supporto continuo.

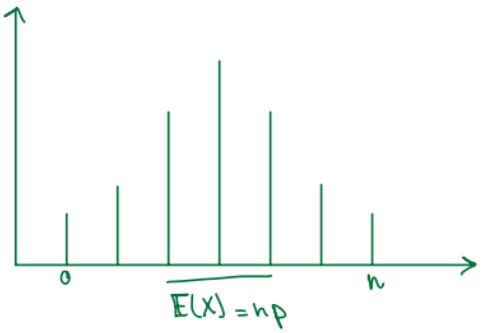
Guardiamo adesso com'è fatto il grafico di $f_X(x)$ con $X \sim B(n, p)$ con $D_X = \{0, 1, \dots, n\}$.

Sappiamo che $f_X(x) = (n\ p)p^x(1-p)^{n-x} \cdot I_{D_X}(x) = P(X = x)$

Possiamo vedere che quando la funzione cresce, notiamo che $f_X(x+1) > f_X(x)$ fino ad un certo punto (più precisamente $E(X)$), dopodiché la disuguaglianza non è più vera perchè la funzione inizia a decrescere.



Possiamo anche avere il seguente grafico, dipendentemente dalla relazione tra n e p.



Distribuzione Gaussiana (distribuzione normale)
$X \sim N(\mu, \sigma)$
X è una variabile aleatoria che segue una distribuzione gaussiana con parametri μ e σ.

Guardiamo adesso la sua funzione di densità di probabilità $f_X(x)$

$$f_X(x) = \frac{1}{\sqrt{2\pi}\cdot\sigma} \cdot e^{\frac{-1/2\cdot(x-\mu)^2}{\sigma^2}} \quad \text{con } \mu \in \Re, \sigma \in \Re^+$$

Non abbiamo aggiunto la funzione indicatrice I_{D_X} perchè $D_X = \Re$.

Per sapere com'è fatto il suo grafico facciamo uno studio su $f_X(x)$:

$$f'_X(x) = \frac{1}{\sqrt{2\pi}\cdot\sigma} \cdot e^{\frac{-1/2\cdot(x-\mu)^2}{\sigma^2}} \cdot \left(-\frac{1}{2}\cdot\frac{2}{\sigma^2}(x-\mu)\right) > 0$$

$\frac{-(x-\mu)}{\sigma^2} > 0 \Rightarrow x - \mu > 0 \Rightarrow x < \mu$

(Quindi funzione crescente per $x < \mu$)

$$\Rightarrow f'_X(x) = \frac{1}{\sqrt{2\pi\sigma^3}}(\mu - x)e^{-\frac{1}{2}\cdot\frac{(x-\mu)^2}{\sigma^2}}$$

$$f''_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\left(-e^{-\frac{1}{2}\cdot\frac{(x-\mu)^2}{\sigma^2}} + (x-\mu)e^{-\frac{1}{2}\cdot\frac{(x-\mu)^2}{\sigma^2}} \cdot \left(-\frac{1}{2}\cdot\frac{2(x-\mu)}{\sigma^2}\right)\right.$$

$$\Rightarrow f''_X(x) = -\frac{1}{\sqrt{2\pi\sigma^2}}\cdot e^{-\frac{1}{2}\cdot\frac{(x-\mu)^2}{\sigma^2}} [sempre > 0]\left(1 - \frac{(x-\mu)^2}{\sigma^2}\right) > 0$$

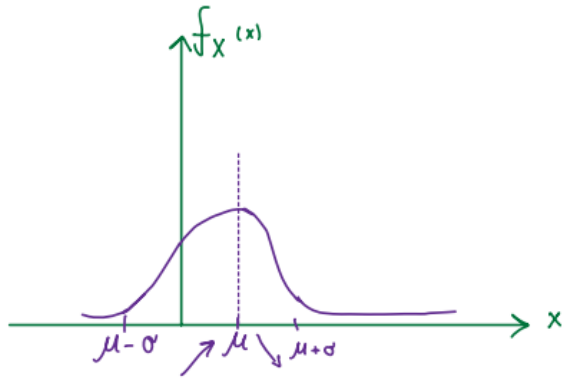
$$\Rightarrow \left(\frac{x-\mu}{\sigma}\right)^2 > 1$$

Sappiamo che $x^2 > 1$ equivale a dire $|x| > 1$, quindi:

$\Rightarrow \left|\frac{x-\mu}{\sigma}\right| > 1 \Rightarrow |x-\mu| > \sigma$ (σ posso portarlo fuori perchè è > 0).

$\Rightarrow x > \mu + \sigma \wedge x < \mu - \sigma$

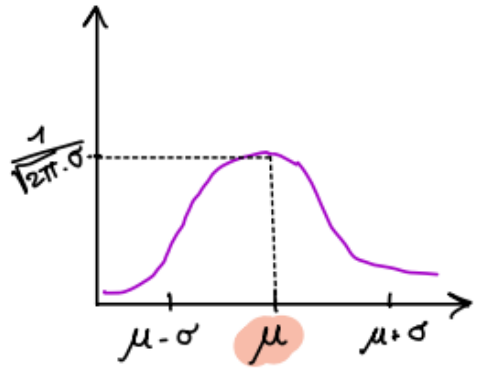
Quindi sarà concava tra $\mu + \sigma$ e $\mu - \sigma$ mentre sarà convessa negli altri punti:



Possiamo usare la distribuzione gaussiana per approssimare una distribuzione binomiale con un n troppo grande.

Cerchiamo adesso la funzione di ripartizione:
Potremmo integrare la funzione di massa di probabilità per trovare la funzione di ripartizione, il problema è che non esiste una forma analitica da noi conosciuta per integrare questa funzione, in teoria dovremmo quindi usare delle "acrobazie matematiche" pericolose.
Invece faremo degli atti di fede dando per vere alcune assunzioni.

Dal grafico di $f_X(x)$ cerchiamo di capire il valore di μ e σ:



- Primo atto di fede:

μ sta al centro del grafico ⇒ $E(X) = \mu$

- Secondo atto di fede:

σ invece indica quanto i valori si distanziano da μ ⇒ σ è la deviazione standard ⇒ σ^2

Vediamo alcune proprietà della distribuzione gaussiana:
Se $X \sim (\mu, \sigma)$ e $Y = aX + B$ con $a, b \in \Re$ e $b \neq 0$
$\Rightarrow Y \sim N(a\mu + b, |a|\sigma)$

Ma se abbiamo $Z = \frac{X-\mu}{\sigma}$ (sottraiamo il valore atteso della variabile aleatoria X e dividiamo per la deviazione standard avremo un operazione di standardizzazione, già vista).
Calcoliamo $E(Z)$ e $Var(Z)$:
$E(Z) = E(\frac{X-\mu}{\sigma}) = \frac{1}{\sigma}E(X - \mu) = \frac{1}{\sigma}(E(X) - E(\mu)[= \mu\ dato\ che\ è\ una\ costante])$
$= \frac{1}{\sigma}(E(X) - \mu) = \frac{1}{\sigma}(\mu - \mu) = 0$
$Var(Z) = Var(\frac{X-\mu}{\sigma}) = \frac{1}{\sigma^2}Var(X - \mu) = \frac{Var(X)}{\sigma^2} = 1$
Da questo concludiamo che anche $Z \sim N(0, 1)$ con $\mu = 0, \sigma = 1$ e si dice che Z è una variabile aleatoria normale (o gaussiana) standard.

Adesso calcoliamo $F_X(x) = P(X \leq x)$ standardizzando X:
$P(\frac{X-\mu}{\sigma} \leq \frac{x-\mu}{\sigma}) = P(Z \leq \frac{x-\mu}{\sigma}) = F_Z(\frac{x-\mu}{\sigma})$

Riproducibilità di variabili aleatorie:
$X_i \sim N(\mu_i, \sigma_i) \qquad 1 \leq i \leq n \quad$ con X_i indipendenti fra loro

$$Y := \sum_{i=1}^{n} X_i \Rightarrow Y \sim N(\sum_{i=1}^{n} \mu_i, \sqrt{\sum_{i=1}^{n} \sigma_i^2})$$

$$E(Y) = \sum_{i=1}^{n} E(X_i) = \sum_{i=1}^{n} \mu_i$$

$$Var(Y) = \sum_{i=1}^{n} Var(X_i) = \sum_{i=1}^{n} \sigma_i^2$$

Applicazione del concetto di mediana (la chiamiamo m) su una variabile aleatoria:

$P(X \leq m) = 1/2 = F_X(m) \Rightarrow m = F_X^{-1}(1/2)$

$P(X > m) = 1/2$

Esiste una specificazione che si lascia alla sua destra metà di $f_X(x)$ e esattamente metà a sinistra di $f_X(x)$.

Solo se abbiamo X continua possiamo calcolare $F_X^{-1}(x)$, esempio:

$X \sim E(\lambda)$ con $F_X(x) = 1 - e^{-\lambda x}$

$1 - e^{-\lambda x} = 1/2$

$e^{-\lambda x} = 1/2$

Applichiamo il logaritmo naturale ad entrambe le parti:

$- \lambda x = ln(1/2) \Rightarrow x = -\frac{1}{\lambda} ln(1/2)$

$\Rightarrow F_X^{-1}(m) = -\frac{1}{\lambda} ln(1/2)$

Possiamo estendere il concetto di quantile su una variabile aleatoria?
Differenza tra mediana e quantile:
La mediana è un tipo specifico di quantile, è il valore centrale di un insieme di dati ordinati, mentre il quantile è un valore che divide un insieme di dati in due parti di dimensioni dipendenti dal quantile.

x_q = quantile di una Variabile Aleatoria X con $q \in (0, 1)$

$P(X \leq x_q) = q = F_X(x_q)$

$\Rightarrow x_q = F_X^{-1}(q)$

$1 - e^{-\lambda x} = q$

$- e^{-\lambda x} = q - 1$

$1 - q = e^{-\lambda x}$
Applico il logaritmo naturale ad entrambe le parti:
$ln(1 - q) = -\lambda x$
$x = -\frac{1}{\lambda} ln(1 - q)$
$\Rightarrow F_X^{-1}(q) = -\frac{1}{\lambda} ln(1 - q)$

Teorema di De Maine-Laplace
Se ho $X \sim B(n, p) \Rightarrow \sim$ ° $[approssimativamente\ distribuita] N(np, \sqrt{np(1 - p)})$
Immaginiamo di avere $X_1, \dots, X_n \sim B(p)$ indipendenti (seguono una distribuzione di Bernoulli).
$Y = \sum_{i=1}^{n} X_i = Y \sim B(n, p) \Rightarrow Y \sim °N(np, \sqrt{np(1 - p)})$

Un altro risultato importante (teorema centrale del limite):
Se ho X con $E(X) = \mu$ e $Var(X) = \sigma^2$, immaginiamo di avere
$X_1, \dots, X_n \sim X$ $[stessa\ generica\ distribuzione]$ indipendenti
Se $Y := \sum_{i=1}^{n} X_i \Rightarrow Y \sim °N(n\mu, \sqrt{n\sigma})$

Lezione del 18 Maggio 2023

Lezione 20

Nella scorsa lezione abbiamo visto la famiglia di distribuzione gaussiana (o normale).

Iniziamo a vedere degli esercizi pratici:
Problema 1
$X_i = risarcimento\ del\ cliente\ i$
$E(X_i) = 320$
$\sigma_X = 540$
$X = risarcimento\ cumulato\ su\ tutti\ i\ clienti$
$= \sum_{i=1}^{n} X_i$
$n = numero\ di\ clienti = 25000$
$P(X > 8,3 \cdot 10^6) =?$

Applichiamo il teorema centrale del limite.

Se sommo un certo numero di variabili aleatorie indipendenti con la stessa distribuzione, posso approssimarle secondo una distribuzione gaussiana.
Condizioni:

1) Stessa generica distribuzione
2) Indipendenti
3) Media e varianza definite

$E(X_i) = \mu_X$

$\sum_{i=1}^{n} X_i \sim {}^{\circ}N(n\mu_X, \sqrt{n}\sigma_X) \sim Y$ [*introduciamo una nuova variabile aleatoria*]

$P(X > 8,3 \cdot 10^6) \approx P(Y > 8,3 \cdot 10^6) = 1 - P(Y \leq 8,6 \cdot 10^6)$

$= 1 - F_Y(8,3 \cdot 10^6)$

Avendo una variabile normale di parametri μ, σ

$A \sim N(\mu, \sigma)$

$F_A(x) = P(A \leq x) = P(\frac{A-\mu}{\sigma} \leq \frac{x-\mu}{\sigma})$

$Z \sim N(0, 1)$

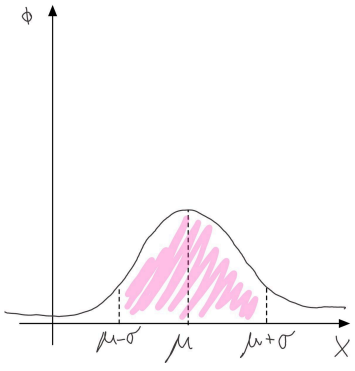
$= P(Z \leq \frac{x-\mu}{\sigma}) = \Phi_Z(\frac{x-\mu}{\sigma})$

Φ (Phi maiuscola) funzione di ripartizione per distribuzioni normali standard.
ϕ (phi minuscola) funzione di densità di probabilità per distribuzioni normali standard.

Quindi dalla formula di prima ho:

$= 1 - \Phi(\frac{8,3 \cdot 10^6 - 8 \cdot 10^6}{8,54 \cdot 10^4}) \sim 1 - \Phi(3,51) \sim 0.00022$

Come avevamo visto per gli attributi distribuiti su una curva gaussiana, anche per le variabili aleatorie abbiamo:



$P(\mu + \sigma) - P(\mu - \sigma) = 0.75$
$P(\mu + 2\sigma) - P(\mu - 2\sigma) = 0.85$
$P(\mu + 3\sigma) - P(\mu - 3\sigma) = 0.99$

Problema 2
Il 30% degli iscritti segue le lezioni.
$P(scegliendo\ uno\ studente\ a\ caso\ \text{è}\ frequentante) = 0.3 = p$
$\forall i = 1, \dots, 450$
$X_i = esperimento\ bernoulliano = 1$ se l'i-esimo studente è frequentante, 0 altrimenti
$X_i = B(p)$

$X = numero\ degli\ studenti\ a\ lezione = \sum_{i=1}^{450} X_i$

Teorema centrale del limite:
$\sim {}^{\circ}N(450 \cdot p\ [= 135], \sqrt{450p(1-p)}[= 9,72])$
$P(X > 150)$
Problema di granularità
Se descrivo X come normale, $X > 150$ e $X > 150,001$ sono diversi.
Allora considero un intorno di 150 con precisione $0,5$
$P(X > 150) = P(X \geq 150,5) = 1 - P(X \leq 150,5)$
$= 1 - P(Z < \frac{150,5-135}{9,72}) = 1 - \Phi(\frac{150,5-135}{9,72}) \approx 0,06$

Problema 3
$X_t = precipitazioni\ a\ Los\ Angeles\ nell'anno\ t$
$\sim N(12,08,\ 3,1)$
$P(X_{2024} + X_{2025} > 25)\ =?$
$X_{2024} + X_{2025} = S \sim N(24,16,\ 3,1\sqrt{2})$
$= P(S > 25) = 1 - F_X(25) = 1 - \Phi(\frac{25-24,16}{\sqrt{2}\cdot 3,1}) \approx 0,4240$

$P(X_{2024} > X_{2025} + 3)\ =?$
$P(X_{2024} - X_{2025} > 3)$
$D = X_{2024} - X_{2025} \sim N(0, \sqrt{2} \cdot 3,1)$

Dato che:
$Var(X - Y) = Var(X) + Var(-Y) = Var(X) + Var(Y)$

$= P(D > 3) = 1 - F_D(3) = 1 - \Phi(\frac{3}{\sqrt{2}\cdot 3,1}) \approx 0,2463$

Ultimo grande argomento del corso:
Statistica inferenziale
Qual'è la domanda che si pone questo tipo di statistica?
Immaginiamo di voler tirare fuori dalle informazioni da un esperimento non deterministico tramite un campione di osservazione.
Anche qui ritornano i concetti di:

- Popolazione
  - Variabile aleatoria $X \sim F$ (una certa distribuzione)
  - Vogliamo sapere qual'è la sua distribuzione
- Campione
  - $X_1, \dots, X_n \sim F$ i.i.d. (indipendentemente, identicamente distribuite) successione di variabili aleatorie
  - n si dice dimensione o taglia del campione
  - Una volta osservato un campione ho le sue specificazioni $x_1, \dots, x_n$

F (la distribuzione della popolazione) è completamente sconosciuta.
In questo caso la statistica inferenziale è "non parametrica".
Quando F è parzialmente conosciuta allora si parla di statistica inferenziale parametrica (so che appartiene ad una certa famiglia, ma non so a quale distribuzione appartiene, quindi non conosco i suoi parametri).

Concentriamoci sul secondo caso
$F(\theta)$
θ??
Faremo approssimazioni o stime di theta.
Lo posso fare in due modi:

1) Stima puntuale fornendo un numero simile
2) Stima per intervalli fornendo un range di approssimazione

In questo corso ci concentreremo sulla statistica inferenziale parametrica puntuale.

Introduciamo adesso il concetto di Statistica/Stimatore.
D_X Supporto
$t: D_X^{\,n} \to \Re$
(n elementi specifici di X, ovvero il campione, il codominio è R)

$t(x_1, \dots, x_n) = \hat{\theta}$ $[stima\ per\ theta]$
$t(x_1, \dots, x_n) \approx \theta$ $[circa\ theta]$

Non voglio però davvero stimare theta, ma $\tau(\theta)$ (tau di theta).
Quindi $t(x_1, \dots, x_n) = \hat{\tau} \approx \tau(\theta)$

Esempio:
$X \sim E(\lambda)$
$X = tempo\ prima\ che\ arrivi\ la\ metro$
Non conosco il parametro $\theta = \lambda$, però quello che vorrei sapere è altro, magari $E(X)$
$E(\lambda) = \frac{1}{\lambda} =: \tau(\theta)$

Oltre ad applicare queste approssimazioni dobbiamo vedere quanto siano corrette.
$\theta \in \aleph$
$X \qquad f_X(x) = I_{[\theta-1/2,\, \theta+1/2]}(x)$
Estraggo un campione x_1
$t(x_1) = int(x_1)$ intero più vicino
$t(X_1, \dots, X_n)$ = Una statistica su una serie di variabili aleatorie

Come capisco se la statistica che sto usando è ben focalizzata?
Popolazione X
Campione $Y_1, \dots, Y_n$
Parametro θ
Valore da stimare $\tau(\theta)$
Statistica t
Una statistica t si dice "non deviata" rispetto a $\tau(\theta)$ se e solo se
$E(t(Y_1, \dots, Y_n)) = \tau(\theta)$

Uso il valore atteso per cogliere la centralità del campione aleatorio (statistiche non deviate, non distorte, corrette (unbiased)).

Esempio pratico:
$\tau(\theta) = E(X)$
$t(X_1, \dots, X_n) = \frac{1}{n}\sum_{i=1}^{n} X_i = \overline{X}$ stimatore del valore atteso uguale alla media campionaria
$E(t(X_1, \dots, X_n)) = E(X)$
$E(\overline{X}) = E(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{1}{n}\sum_{i=1}^{n} E(X_i)$
$= \frac{1}{n} \cdot nE(X) [= possiamo\ farlo\ perch\text{è}\ ogni\ X_i\ \text{è}\ distribuita\ come\ X]$

$= E(X)$
Si trova come estimatore del valore atteso

Si indica come dispersione o bias:
$b_{\tau(\theta)}(t(X_1, \dots, X_n)) = E(t(X_1, \dots, X_n) - \tau(\theta))$

Vediamo quanto sono ravvicinate le stime della media campionaria e della varianza:

$Var(\overline{X}) = Var(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{1}{n^2}Var(\sum_{i=1}^{n} X_i)$

$= \frac{1}{n^2}\sum_{i=1}^{n} Var(X_i)$ $[dato\ che\ sono\ indipendenti]$

$= \frac{1}{n^2}nVar(X)$ $[dato\ che\ sono\ identicamente\ distribuite\ come\ X]$

$= \frac{1}{n}Var(X)$

Concetto intermedio:
Scarto quadratico medio
Definito come:

$MSE_{\tau(\theta)}(t(X_1, \dots, X_n)) = E((t(X_1, \dots, X_n) - \tau(\theta))^2)$

Fino ad ora abbiamo immaginato n come fisso.

Consistenza in media quadratica:

$\lim_{n \to +\infty} MSE_{\tau(\theta)}(t(X_1, \dots, X_n))$

Se questo è zero, lo stimatore è consistente in media quadratica.
Se lo stimatore non è deviato:

$E(t(X_1, \dots, X_n)) = \tau(\theta)$

$MSE_{\tau(\theta)} = E((t(X_1, \dots, X_n) - E(t(X_1, \dots, X_n)))^2)$

Chiamiamo per comodità $t(X_1, \dots, X_n) := T_n$

$MSE_{\tau(\theta)} = E((T_n - E(T_n))^2) = Var(T_n) = Var(t(X_1, \dots, X_n))$

Quindi $Var(\overline{X}) = \frac{1}{n}Var(X) = MSE_{E(X)}(\overline{X})$
(Quindi la varianza dell'estimatore è uguale allo scarto quadratico medio dell'estimatore).

Esercizio:
$X \sim N(\mu, \sigma)$
$\theta = \sigma$

$\tau(\theta) = \sigma^2$
Proponiamo uno stimatore per un campione di due elementi X_1, X_2

$t(X_1, X_2) = (X_1)^2 - X_1X_2$
Calcoliamone il valore atteso:
$E(T_n) = E(X_1^2) - E(X_1X_2)$ [fattorizzazione]

$E(X_1^2) - E(X_1)E(X_2)$
Dato che X_1, X_2 e X seguono la stessa distribuzione

$E(X^2) - E(X)^2 = Var(X)$
Quindi T_n è uno stimatore non variato.

$MSE_{\tau(\theta)}(t(X_1, \dots, X_n)) = E((T_n - \tau(\theta))^2)$

$= E(((T_n - E(T_n)) + (E(T) - \tau(\theta)))^2)$

$= E((T_n - E(T_n))^2 + 2E(T_n - E(T_n))E(E(T_n) - \tau(\theta)) + E(E(T_n) - \tau(\theta))^2$
Dove $2E(T_n - E(T_n))E(E(T_n) = 0$

$MSE_{\tau(\theta)} = Var(T_n) + b_{\tau(\theta)}(T_n)^2$

Esempio:
$Var(T_n) = \frac{n\sigma^2}{q}$

$b_{\tau(\theta)} = \frac{n}{3}\mu - \mu = \mu(n/3 - 1)$

$MSE_\mu(T_n) = \frac{n\sigma^2}{q} + \sigma^2(n/3 - 1)^2$

$\lim_{n \to +\infty} MSE_\mu(T_n) = +\infty$

Lezione del 23 Maggio 2023

Lezione 21

Durante la lezione precedente abbiamo iniziato a parlare di statistica inferenziale.

Statistica inferenziale
Perchè inferenziale?
Perchè ci permette di analizzare un campione e generalizzare i risultati ottenuti sulla popolazione.
Perchè puntuale?

Perchè la usiamo per stimare (fornire un numero) su una quantità per me ignota, $\tau(\theta)$.
Perchè parametrica?
Perchè sia per la popolazione che per il campione cerco la famiglia di distribuzione a meno di un parametro.
Quindi abbiamo una statistica inferenziale parametrica puntuale.
Questa statistica è una funzione che prende un campione (una serie di specifiche di variabili aleatorie) e lo associa ad un numero in $\Re$.

Ho una popolazione $X \sim F(\theta)$ (potremmo avere anche più di un parametro per la nostra variabile aleatoria, ma al massimo uno ignoto).
θ parametro ignoto
Abbiamo una quantità ignota $\tau(\theta)$
E un campione $X_1, \dots, X_n \qquad \forall i \; X_i \sim F(\theta)$

Esempio:
$X \sim B(p) \qquad \theta = p$
$\tau(\theta) = \tau(p) = E(X) = p$
Oppure $\tau(\theta) = \tau(p) = Var(X) = p(1-p)$
Oppure $\tau(\theta) = P(scegliendo\ 10\ individui\ sono\ tutti\ 1) = p^{10}$

Il nostro scopo è trovare una stima, la chiamiamo $\hat{\tau}$ che deve essere $\approx \tau(\theta)$, come facciamo?
Introducendo quella che abbiamo chiamato "statistica", funzione $t(X_1, \dots, X_n) = T$ e le specificazioni di questa T_n sono $\hat{\tau}$.

Esercizi:
$X \sim B(p) \qquad \theta = p \qquad \tau(\theta) = \tau(p) = p$
Ho un campione X_1, X_2, X_3
$T = \frac{X_1 + 2X_2 + X_3}{5}$
T è corretta per stimare p?
Per sapere se una certa statistica sia corretta o meno bisogna calcolare il suo valore atteso.
$E(T) = E(\frac{X_1 + 2X_2 + X_3}{5}) = \frac{1}{5}E(X_1) + 2E(X_2) + E(X_3)$
Dato che $X_1, X_2, X_3 \sim B(p)$ come X e sono indipendenti
$\Rightarrow \frac{1}{5}(E(X) + 2E(X) + E(X)) = \frac{4}{5}p$
$\Rightarrow T$ non è corretta perchè doveva essere uguale a p

Calcolo quindi il bias:

$b_p(T) = E(T) - p = \frac{4}{5}p - p = -\frac{p}{5}$
$b_p(T)$ = bias di T rispetto a p

$MSE_p(T) = E((T-p)^2)$ oppure $Var(T) + (b_p(T))^2$ (usiamo questa)
$Var(T) = Var(\frac{X_1+2X_2+X_3}{5}) = \frac{1}{25}(Var(X_1) + 4Var(X_2) + Var(X_3))$
Sempre per il fatto che $X_1, X_2, X_3 \sim F \sim X$ indipendenti e per il fatto che $Var(X) = p(1-p)$
$\Rightarrow \frac{6}{25}Var(X) = \frac{6}{25}p(1-p)$

$\Rightarrow MSE_p(T) = \frac{6}{25}p(1-p) + \frac{p^2}{25} = \frac{1}{25}p(6-6p+p) = \frac{p(6-5p)}{25}$

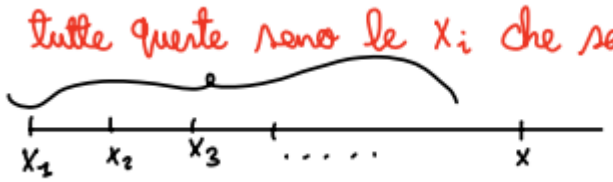
Come interpretiamo il MSE?
E' un indicatore di quanto bene un modello di previsione si adatta ai dati osservati (l'errore che si commette stimando quella quantità ignota usando la statistica T intorno a cosa viaggia).

N.B:
Indipendentemente dalla distribuzione, la media campionaria è sempre uno stimatore non distorto per il valore atteso $E(X)$.
(Uno stimatore non è per forza l'unico).

Esempio:
$X \sim U([0, \theta])$ $\tau(\theta)$
Cosa succede se usiamo $E(\overline{X})$?
Cioè se usassimo la media campionaria $\overline{X}$ come stimatore.
$E(\overline{X}) = E(X) = \frac{0+\theta}{2} = \frac{\theta}{2}$ (se moltiplicassi per 2 avrei il risultato sperato).

$2E(\overline{X}) = \theta \Rightarrow E(2\overline{X}) = \theta \Rightarrow$ usiamo $T'' = 2\overline{X}$ come stimatore e a questo punto abbiamo uno stimatore non distorto.

Proponiamo adesso $T' = Max_{1\le i\le n} X_i$ (vedendo θ come estremo destro del mio campione $\Rightarrow$ usiamo il valore massimo del campione per stimare θ).
Calcoliamo $F_{T'}(x) = P(T' \le x) = P(Max\, X_i \le x) = P(\forall i\ X_i \le x)$

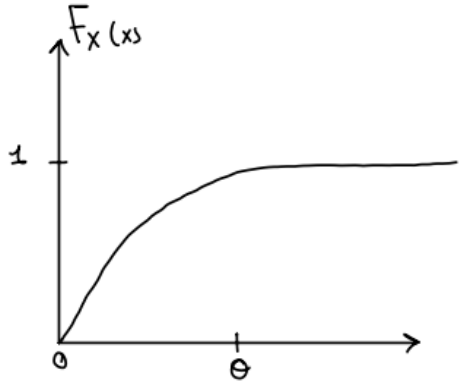


$$P(\forall i\ X_i \leq x) = P(\bigcap_{i=1}^{n} X_i \leq x)[per\ le\ X_i\ indipendenti] = \prod_{i=1}^{n} P(X_i \leq x)$$

$$= \prod_{i=1}^{n} F_{X_i}(x) = \prod_{i=1}^{n} F_X(x)$$

$$\Rightarrow P(\forall i\ X_i \leq x) = (F_X(x))^n$$

Come possiamo vedere $F_X(x)$?



Quanto vale $F_X(x)$?

$$F_X(x) = \begin{cases} 0 & -\infty < x < a \\ \frac{x-a}{b-a} & a \leq x < b \\ 1 & b \leq x < +\infty \end{cases}$$

$$\Rightarrow F_X(x) = \frac{x-0}{\theta-0} = x/\theta = (F_X(x))^n = (\frac{x}{\theta})^n$$

Calcoliamo invece la funzione di densità di T', visto che abbiamo la funzione di ripartizione, possiamo derivare per ottenere quella di densità:

$$\Rightarrow f_{T'}(x) = F_{T'}'(X) = \frac{d}{dx}(\frac{x}{\theta})^n = n \cdot (\frac{x}{\theta})^{n-1} \cdot \frac{1}{\theta} = \frac{n \cdot x^{n-1}}{\theta^n}$$

Calcoliamo adesso il valore atteso di T':

$$E(T') = \int_{-\infty}^{+\infty} x f_{T'}(x)dx = \int_0^{\theta} x \frac{n \cdot x^{n-1}}{\theta^n} dx = \frac{n}{\theta^n} \int_0^{\theta} x^n dx$$

$$= \frac{n}{\theta^n} \cdot \frac{x^{n+1}}{n+1}\Big|_0^{\theta} = \frac{n}{\theta^n}(\frac{\theta^{n+1}}{n+1}) = \frac{n}{n+1}\theta$$

Quindi se $T' = Max\ X_i$ con $E(T') = \frac{n}{n+1}\theta$, se avessimo invece scelto $T' = \frac{n+1}{n} Max\ X_i \Rightarrow E(T') = \theta \Rightarrow T'$ sarebbe stato un corretto stimatore per θ

.

N.B:
Questo $\frac{n+1}{n}$ è un fattore di correzione per togliere la distorsione di uno stimatore, attraverso un'operazione che si chiama "plugin".

⇒ Sia T' che T'' sono due stimatori non distorti per θ, sono entrambi consistenti in media quadratica?
Dobbiamo calcolare MSE e poi calcolarne il limite per $n \to +\infty$.

$T'' = 2\overline{X}$

$MSE_{\theta}(T'') = Var(T'') + (bias_{\theta}(T))^2$

$Var(T'') = Var(2\overline{X}) = 4Var(\overline{X}) = \frac{4}{n}Var(X)$

Dato che:

$Var(X) = \frac{(b-a)^2}{12} = \frac{(\theta-0)^2}{12} = \frac{\theta^2}{12}$

$\Rightarrow \frac{4}{n} \cdot \frac{\theta^2}{12} = \frac{\theta^2}{3n}$

$bias_{\theta}(T'') = E(T'') - \theta = \theta - \theta = 0$

$MSE_{\theta}(T'') = \frac{\theta^2}{3n}$

Adesso calcoliamone il limite per controllare se è consistente.

$\lim_{n \to +\infty} MSE_{\theta}(T'') = \lim_{n \to +\infty} \frac{\theta^2}{3n} = 0 \Rightarrow T''$ è consistente

Calcoliamo adesso il MSE ed il limite ad infinito per T':

$T' = \frac{n+1}{n} Max\ X_i$

$MSE_{\theta}(T') = Var(T') + (bias_{\theta}(T'))^2$ [$sappiamo\ già\ che$ è $zero$]

$MSE_{\theta}(T') = Var(T') = E(T'^2) - E(T')^2 [= \theta^2]$

$= E((\frac{n+1}{n} Max\ X_i)^2) - E(T')^2$

$= (\frac{n+1}{n}) E((Max\ X_i)^2) - E(T')^2$

$E(T'^2\ [T'\ vecchio,\ senza\ \frac{n+1}{n}]) = \int_0^{\theta} x^2 f_{T'}(x)dx = \frac{n}{\theta^n} \int_0^{\theta} x^{n+1} dx = \frac{n}{\theta^n} \cdot \frac{x^{n+2}}{n+2} [_0^{\theta}$

$= \frac{n}{\theta^n} (\frac{\theta^{n+2}}{n+2}) = \frac{n}{n+2}\theta^2$

$\Rightarrow MSE_{\theta}(T') = (\frac{n+1}{n})^2 \cdot \frac{n}{n+2}\theta^2 - \theta^2 = \theta^2(\frac{(n+1)^2}{n(n+2)} - 1)$

$= \theta^2(\frac{n^2+2n+1-n^2-2n}{n(n+2)}) = \frac{\theta^2}{n(n+2)}$

$\lim_{n \to +\infty} MSE_\theta(T') = \lim_{n \to +\infty} \frac{\theta^2}{n(n+2)} = 0$

⇒ Entrambi gli stimatori sono consistenti in media quadratica, ma possiamo scegliere T' dato che va a zero più velocemente.

Quale stimatore è meglio?
$MSE_\theta(T') \leq MSE_\theta(T'')$

$\frac{\theta^2}{n(n+2)} \leq \frac{\theta^2}{3n} \Rightarrow \frac{1}{n+2} \leq \frac{1}{3} \Rightarrow 3 \leq n + 2 \Rightarrow n \geq 1$

Quindi per $n \geq 1$ T' è migliore di T''.

Lezione del 25 Maggio 2023

Lezione 22

Uno dei due criteri utilizzati per stabilire l'efficienza di una statistica è la consistenza in media quadratica (errore medio del campione che tende all'infinito).

$X: popolazione \sim F(\theta)$
$\tau(\theta) = t(X_1, \dots, X_n)$
$T_1, \dots, T_n$
Variabile aleatoria corrispondente ai valori restituiti dalle rispettive statistiche associate.
$|T_n - \tau(\theta)|$
Altro modo per misurare la differenza tra il valore effettivo da calcolare ed il risultato dei nostri estimatori, quindi l'errore.
Tutto quello nel valore assoluto è anch'essa una variabile aleatorie, essendo una funzione su una variabile aleatoria (T_n).

Se volessi trasformarlo invece in un evento?
$|T_n - \tau(\theta)| < \varepsilon$
Questo è un evento.
Posso chiedermi con quale probabilità accade questo evento:
$P(|T_n - \tau(\theta) < \varepsilon)$
Cosa succede al limite per n che tende a $+\infty$ di questa probabilità?

$$\lim_{n \to +\infty} P(|T_n - \tau(\theta) < \varepsilon) = 1 \Rightarrow$$ Debolmente consistente

Infatti:

$$P(|T_n - \tau(\theta) < \varepsilon)$$

$$= P((T_n - \tau(\theta))^2 < \varepsilon^2)$$

$$= 1 - P((T_n - \tau(\theta))^2 \geq \varepsilon^2)$$

Per la disuguaglianza di Markov

$$P(X \geq a) \leq \frac{E(X)}{a}$$

Quindi:

$$\geq 1 - \frac{E((T_n - \tau(\theta))^2) \; [=MSE]}{\varepsilon^2}$$

$$= 1 - \frac{MSE_{\tau(\theta)}(T_n)}{\varepsilon^2}$$

Adesso:
Stimatore consistente in media quadratica:

- $MSE_{\tau(\theta)}(T_n)$
- $P((T_n - \tau(\theta))^2 < \varepsilon^2)$ tende a 1
- Debolmente consistente

Abbiamo visto la media campionaria come stimatore non distorto del valore atteso.
Oltre alle sue altre proprietà possiede la "Legge dei grandi numeri (Forte)".
(Le altre proprietà sono correttezza nello stimatore e consistenza in media quadratica).
Se provo a confrontare il limite per n all'infinito della media campionaria con il valore atteso della popolazione:

$$\lim_{n \to +\infty} \overline{X_n} = E(X)$$

(Evento)

$$P(\lim_{n \to +\infty} \overline{X_n} = E(X)) = 1$$

Quindi quando ho la media campionaria su un campione infinito non rimane una variabile aleatoria ma diventerà una costante pari al valore atteso.

Versione debole:

$$\lim_{n \to +\infty} P(|\overline{X_n} - E(X)| > \varepsilon) = 0$$

$\forall \varepsilon > 0$

Altro stimatore general purpose
Concentriamoci sulla stima della varianza di una popolazione.
Richiedo media e varianza finite.
Varianza campionaria:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

$$E(s^2) = \sigma^2 = Var(X)$$

Dimostrazione per passi:

1) Sommatoria

$$\sum_i (X_i - \overline{X}) = \sum_i (X_i^2 - 2X_i\overline{X} + \overline{X}^2) = \sum_i X_i^2 - 2\overline{X}\sum_i X_i + n\overline{X}^2$$

Dato che:

$$\overline{X} = \frac{\sum_i X_i}{n} \qquad \sum_i X_i = n\overline{X}$$

Allora:

$$= \sum_i X_i^2 - 2n\overline{X}^2 + n\overline{X}^2 = \sum_i X_i^2 - n\overline{X}^2$$

2) Momento secondo

$$E(X^2) = Var(X) + E(X)^2$$

3) Valore atteso della media

$$E(\overline{X}) = Var(\overline{X}) = \frac{\sigma^2}{n}$$

Allora:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2 \Rightarrow (n-1)s^2 = \sum_{i=1}^{n}(X_i - \overline{X})^2$$

Per il primo punto:

$$= \sum_i X_i^2 - n\overline{X}^2$$

Applichiamo il valore atteso:

$$(n-1)E(s^2) = \sum_i E(X_i^2) - nE(\overline{X}^2)$$

Punto due:

$$= \sum_i (Var(X_i) + E(X_i)^2) - n(Var(\overline{X}) + E(\overline{X})^2)$$

$$= \sum_i Var(X_i) + \sum_i E(X_i)^2 - nVar(\overline{X}) - nE(\overline{X})^2$$

X_i facente parte del campione, indipendente
Punto tre:
$= nVar(X) + nE(X)^2 - n\frac{\sigma^2}{n} - nE(X[media\ stimatore\ non\ distorto])^2$
$= n\sigma^2 + n\mu^2 - \sigma^2 - n\mu^2$
$= (n-1)\sigma^2$

Se voglio stimare qualcosa che non sia la media o la varianza?
Stime fatte con metodo plug-in.
$X \sim U([0, \theta])$
$E(\overline{X}) = E(X) = \frac{\theta}{2}$
$2E(\overline{X}) = \theta$
$E(2\overline{X}) = \theta \qquad T = 2\overline{X}$

Uso il metodo plugin se con uno stimatore classico ottengo una funzione di teta:
$E(T) = f(\theta)$
Lo trasformo in:
$E(f(T)) = \tau(\theta)$
Però non funziona sempre:
$X \sim E(\lambda)$
$\theta = \lambda$
$E(\overline{X}) = \frac{1}{\lambda}$
$\lambda = \frac{1}{E(\overline{X})}$
Non posso riutilizzare lo stesso metodo, se prendo $\frac{1}{\overline{X}}$ però otteniamo una nuova stima.
$T = \frac{1}{\overline{X}} = \frac{n}{\sum_i X_i} \qquad \tau = \frac{1}{\overline{X}} \approx \lambda$

Come controllo che non sia distorto?
- Calcolo il bias per vedere se sia zero
- Calcolo il valore atteso per vedere se sia uguale a $\tau(\theta)$

$E(T) = E(\frac{1}{\overline{X}}) = ?$
Non posso semplificare ulteriormente!

Altro esempio:
E se avessi voluto stimare il parametro di una popolazione geometrica?
$X \sim G(p)$

$E(\overline{X}) = E(X) = (1 - p)p$
$pE(\overline{X}) = 1 - p$
$p(1 + E(\overline{X})) = 1$
$p = \frac{1}{1-E(\overline{X})}$
$T = \frac{1}{1-\overline{X}}$

Non sappiamo comunque calcolarne il valore atteso per controllare se sia distorto.
Potrei farlo anche con la varianza, dipende da quale diventa più facile da usare.
$s = \sqrt{s^2}$
Deviazione standard campionaria:
$= \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2}$
$E(s) = \sigma$?
Per controllare facciamo il valore assoluto.
Non possiamo (sappiamo) calcolare comunque il valore atteso di una radice quadrata non conoscendo la distribuzione.

Riprendendo la legge dei grandi numeri.
Quante osservazioni sono abbastanza?
Ad esempio, misurando qualcosa ho sempre un errore.
Distanza Terra-X321B $= d$
Misurazioni:
$X_1, \dots, X_n$
Popolazione:
$X \sim N(d, 2)$
Qual'è la larghezza del campione che avvicina d a $\overline{X}$?
Devo fissare 2 cose:

1) Soglia r $\quad |\overline{X} - d| \leq r$
2) Probabilità $P(|\overline{X} - d| \leq r)$ fissata a 0.95

$P(|\overline{X} - d| \leq r) \geq 0.95$
Il numero di osservazioni è ancora non fissato.
Come trovo il minimo?
$P(-r \leq \overline{X} - d[E(X)] \leq r) \geq 0.95$
$E(\overline{X}) = d$
$Var(\overline{X}) = 4/n$

Standardizzazione

$\frac{X-E(X)}{Var(X)} = Y \sim N(0,\ 1)$

Quindi dividendo quello dentro le parentesi della probabilità per σ ($= \sigma/\sqrt{n}$):

$P(-\frac{r}{2/\sqrt{n}} \leq \frac{\bar{X}-d}{2/\sqrt{n}} \leq \frac{r}{2/\sqrt{n}})$

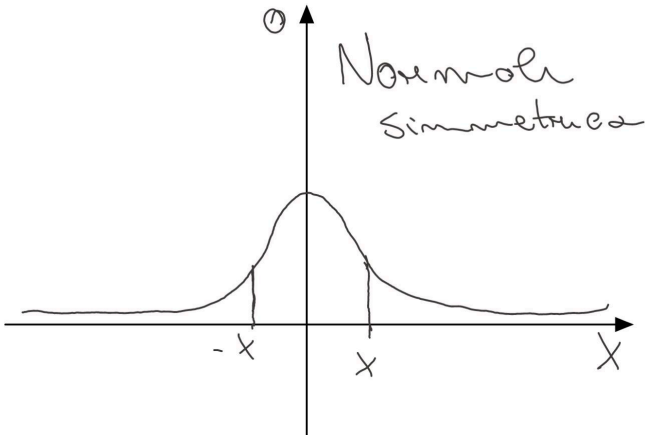
$\frac{\bar{X}-d}{2\sqrt{n}} = Z \sim N(0,\ 1)$

$= P(-\frac{r}{2}\sqrt{n} \leq Z \leq \frac{r}{2}\sqrt{n})$

$= \Phi(\frac{r}{2}\sqrt{n}) - \Phi(-\frac{r}{2}\sqrt{n})$

Considerando che:

$\Phi(-x) = P(Z \leq -x) = 1 - P(Z \geq -x)$



$= 1 - P(Z \leq x) = 1 - \Phi(x)$

Posso quindi complementare la funzione di ripartizione della distribuzione normale simmetrica (è simmetrica anche la standard).

Quindi:

$= \Phi(\frac{r}{2}\sqrt{n}) - 1 + \Phi(\frac{r}{2}\sqrt{n}) = 2\Phi(\frac{r}{2}\sqrt{n}) - 1 \geq 0.95\ [imposto\ prima]$

$\Phi(\frac{r}{2}\sqrt{n}) \geq 0.975$

$\Phi(\alpha) \geq 0.975$

Dato che Φ è crescente, allora sarà anche invertibile (dato che è anche continua).

$\alpha \geq \Phi^{-1}(0.975)$

Quindi:

$\frac{r}{2}\sqrt{n} \geq \Phi^{-1}(0.975)$

$n \geq (2/r \cdot \Phi^{-1}(0.975))^2$

Se r fosse 0.5 e se $\Phi^{-1}(0.975) \approx 1.96$

$n \geq 61.4$....

$n = |(2/r \cdot \Phi^{-1}(0.975))^2|$ $[parte\ intera\ superiore]$

Se ho un campione e ne calcolo la media

$X_1, \dots, X_n \rightarrow \overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$

$X \qquad E(x) = \mu \qquad\qquad Var(X) = \sigma^2$

$\sum_i X_i \sim °N(\mu, \frac{\sigma}{\sqrt{n}})$

$\tau(\theta) = \mu$

$P(|\overline{X} - \mu| \leq r) \geq 1 - \delta$

$P(-r \leq \overline{X} - \mu \leq r) \geq 1 - \delta$

$= P(\frac{-r}{\sigma/\sqrt{n}} \leq \frac{\overline{X}-\mu}{\sigma/\sqrt{n}} \leq \frac{r}{\sigma/\sqrt{n}})$

Non sappiamo se X sia normale, però possiamo approssimare ad una distribuzione normalizzata standard:

$\approx P(-\frac{r}{\sigma/\sqrt{n}} \leq Z \leq \frac{r}{\sigma/\sqrt{n}}) \qquad Z \sim N(0, 1)$

$= \Phi(\frac{r}{\sigma/\sqrt{n}}) - \Phi(-\frac{r}{\sigma/\sqrt{n}})$

$= 2\Phi(\frac{r}{\sigma/\sqrt{n}}) - 1 \geq 1 - \delta$

$\Phi(\frac{r}{\sigma/\sqrt{n}}) \geq 1 - \delta/2$

Inverto adesso la funzione di ripartizione:

$\frac{r}{\sigma/\sqrt{n}} \geq \Phi^{-1}(1 - \delta/2)$

$n \geq (\Phi^{-1}(1 - \frac{\delta}{2})\frac{\sigma}{r})^2$

$n \geq [(\Phi^{-1}(1 - \frac{\delta}{2})\frac{\sigma}{r})^2]$ $parte\ intera\ superiore$

Abbiamo δ e r inversamente proporzionali a n.

Se $\delta = 0$ allora Φ^{-1} tende a $+\infty$.

σ invece è direttamente proporzionale a n.

Risolvendo l'espressione per δ ho:

$\delta \geq 2(1 - \Phi(\frac{\sigma}{r}\sqrt{n}))$

Per r:

$r \geq \Phi^{-1}(1 - \delta/2)\frac{\sigma}{n}$

Solitamente poi sigma dobbiamo stimarlo dai dati $\widehat{\sigma}$.

Avendo precedentemente il teorema centrale del limite però, se otteniamo un n troppo piccolo dobbiamo alzarlo almeno ad una trentina.

Quello che volevamo calcolare però era:
$P(|\overline{X} - \mu| \leq r) = 1 - P(|\overline{X} - \mu| \geq r)$

Per la disuguaglianza di Tchebycheff:
$P(|X - \mu| \geq \varepsilon) \leq \frac{Var(X)}{\varepsilon^2}$
$P(|\overline{X} - \mu| \geq r) \leq \frac{Var(\overline{X})}{\varepsilon^2} \leq \frac{Var(X)}{n\varepsilon^2}$

Quindi:
$\leq 1 - \frac{\sigma^2}{nr^2} \geq 1 - \delta$
$\frac{\sigma^2}{nr^2} \leq \delta$
Quindi col secondo metodo abbiamo:

- $\delta \geq \frac{\sigma^2}{nr^2}$
- $n \geq \frac{\sigma^2}{\delta r^2}$
- $r \geq \frac{\sigma}{\sqrt{n}\delta}$

Quale dei due metodi è il migliore?
TCL (teorema centrale del limite):

- $n \geq \Phi^{-1}(1 - \frac{\delta}{2})^2 \cdot \frac{\sigma^2}{r^2}$

TCH (teorema di Tchebycheff):

- $n \geq \frac{\sigma^2}{\delta r^2}$

$TCL \leq TCH$
$\Phi^{-1}(1 - \frac{\delta}{2})^2 \cdot \frac{\sigma^2}{r^2} \leq \frac{\sigma^2}{\delta r^2}$
$\Phi^{-1}(1 - \frac{\delta}{2})^2 \leq 1/\delta$
$\Phi^{-1}(1 - \frac{\delta}{2}) \leq \frac{1}{\sqrt{\delta}}$
$1 - \frac{\delta}{2} \leq \Phi(\frac{1}{\sqrt{\delta}})$

Controllare graficamente (con python).
Questo è sempre vero, quindi TCL ci fornisce un numero più basso.
Questo perchè la disuguaglianza di Tchebycheff richiede meno ipotesi (quindi più generale) e poco informativo.

Lezione del 30 Maggio 2023

Lezione 23

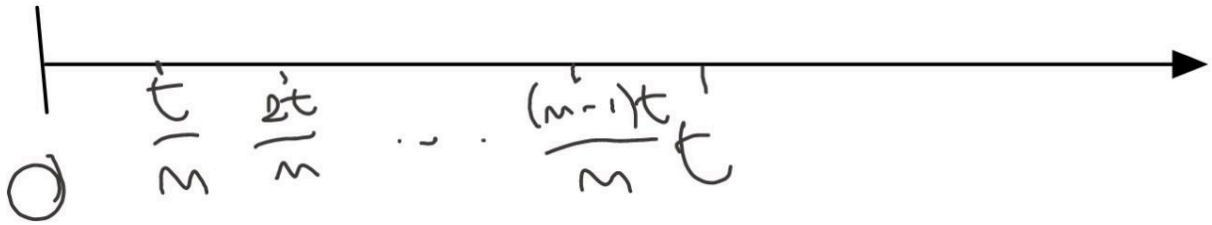
Oggi vedremo gli ultimi 2 argomenti del corso.
Il primo argomento è legato al calcolo delle probabilità.

Processo di Poisson
Processo: qualcosa che ha uno svolgimento nel tempo
Visualizzare quindi processi tenendo conto dell'indice temporale (processi stocastici), in momenti casuali.
Tipicamente quello a cui sono interessato è contare in un intervallo di tempo quante volte accade l'evento interessato.
t: tempo
$t = 0$ istante iniziale
$N(t)$ = numero di eventi che accadono tra zero e t $(0,\ t]$
La N è maiuscola perchè è una variabile aleatoria (non nota a priori e quando assume un valore, esso è un numero).
Si dice invece che, se non fissiamo t, $N(t)$ è una famiglia di variabili aleatorie dei processi stocastici.
Una particolare famiglia loro sottoinsiemi è il processo di Poisson, il quale possiede 5 proprietà:

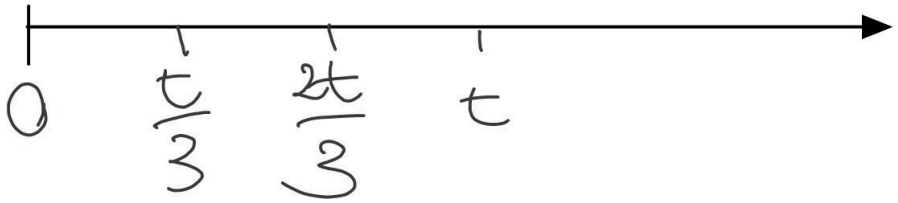
1) $N(0) = 0$
2) Prendendo due intervalli di tempo, le due variabili aleatorie su intervalli disgiunti sono indipendenti
3) Le variabili aleatorie dipendono solo dalla lunghezza dell'intervallo, non dai loro estremi
4) $\lim_{h \to 0} \frac{P(N(h)=1)}{h} = \lambda$
   $P(N(h) = 1) \approx \lambda h$
   La probabilità che avvenga un evento in un lasso di tempo è proporzionale all'ampiezza del lasso di tempo
5) $\lim_{h \to 0} \frac{P(N(h)\geq 2)}{h} = 0$
   $P(N(h) \geq 2) \approx 0$

Posso dimostrare che:
$N(t) \sim P(\lambda h)$

Fissiamo n
Dividiamo quindi t in n intervalli:



$P(N(t) = k)$
Esempio concreto:
$k = 2$
$n = 3$



Sono interessato a $N(t) = 2$
Ho k eventi che si trovano in n intervalli.
A = ogni avvenimento accade in un diverso intervallo
B = tutte le altre possibilità

$P(N(t) = k) = P(A \cup B)$ $[disgiunti]$
$= P(A) + P(B)$
$P(B)$ = almeno due accadimenti sono nello stesso intervallo, con n che tende ad infinito questa probabilità tenderà a zero.
$P(A)$ = in un intervallo in cui avviene l'accadimento è uguale a $\lambda\frac{t}{3}$ (quindi in due intervalli), dove non vi è quel accadimento è uguale a $1 - \lambda\frac{t}{3}$.

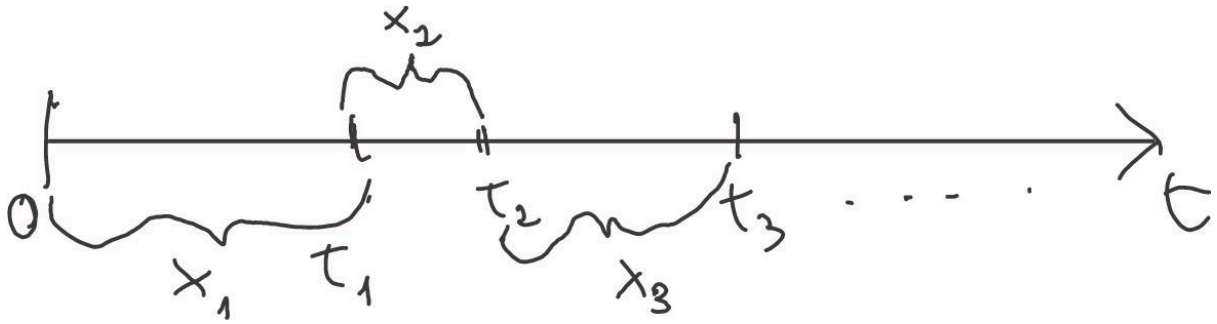
$P(A) + P(B) = (\lambda\frac{t}{3})^k(1 - \lambda\frac{t}{3})^{n-k}$
Questa formula sarebbe corretta se avessi fissato gli intervalli in cui avvengono gli accadimenti, quindi aggiungo un coefficiente binomiale:
$= (\lambda\frac{t}{3})^k(1 - \lambda\frac{t}{3})^{n-k}(n\ k)$

Funzione di massa di probabilità della distribuzione binomiale.
Sembrerebbe che:
$N(t) \sim B(n, \lambda\frac{t}{n})$

Questo è con n fissato, se n cresce ad infinito, una binomiale con n tendente all'infinito è una distribuzione di Poisson.

$N(t) \sim B(n, \lambda\frac{t}{n}) \to [n \to +\infty]\, N(t) \sim P(\lambda\frac{t}{n} \cdot n) = P(\lambda t)$.

Se avessi considerato un altro intervallo, ad esempio $(s, s+t]$ avrei ottenuto lo stesso risultato.



X_1 intertempo fra $t = 0$ e t_1 (specificazione di una variabile aleatoria)
$P(x_1 > t)$ = tempo di primo accadimento = $P(N(t) = 0)$

$P(N(t) = k) = e^{-\lambda t}\frac{(\lambda t)^k}{k!} = e^{-\lambda t}\frac{(\lambda t)^0}{0!} = e^{-\lambda t}$

$F_{X_1}(t) = P(X_1 \leq t) = 1 - P(X_1 \geq t) = 1 - e^{-\lambda t}$
$X_1 \sim E(\lambda)$

$P(X_2 > s + t | X_1 = s)$
$P(nessun\ accadimento\ tra\ s\ e\ s+t | X_1 = s)$
Intervalli di tempo disgiunti $(0, s]$ e $(s, s+t]$, distribuzioni indipendenti.
$= P(nessun\ accadimento\ tra\ s\ e\ s+t) = P(N(t) = 0) = e^{-\lambda t}$
$X_2 \sim E(\lambda)$

Non cambierebbe nulla per X_3, X_4, ...

Abbiamo quindi ricavato una relazione fra il modello discreto di Poisson e quello continuo esponenziale.

Altro argomento
Metodo generale per trovare degli stimatori
Stimatori di massima verosimiglianza
X: popolazione
$\sim F(\theta)$
$X_1, \dots, X_n$ campione di variabili aleatorie indipendenti e identicamente distribuite.

$f_{X_1,\dots,X_n}(x_1,\dots,x_n) = P(X_1 = x_1 \wedge X_2 = x_2 \wedge \dots \wedge X_n = x_n)$
Fattorizzo:
$\prod_{i=i}^{n} P(X_i = x_i) = \prod_{i=1}^{n} f_X(x_i)$

$f_{X_1,\dots,X_n}(x_1,\dots,x_n)$ Verosimiglianza

Se abbiamo quindi due valori di theta tra i quali scegliere, calcolando $\prod_{i=1}^{n} f_X(x_i)$ per questi due valori trovo quello più verosimile, ovvero quello più vicino ad uno.

Formalizziamo:
$L(x_1, x_2, \dots, x_n; \theta)$

Una volta che ho osservato il campione, dipenderà solo da θ.
$\hat{\theta} = arg\,Max\, L(x_1,\dots,x_n;\theta)$
Massimizza la verosimiglianza fissati $x_1,\dots,x_n$ al variare di θ.

Esempio:
$X \sim B(p)$
$f_X(x) = p^x(1-p)^{1-x}$
$L(x_1,\dots,x_n;p) = \prod_{i=1}^{n} f_X(x_i) = \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i}$
$= p^{\sum_i x_1}(1-p)^{\sum_i 1-x_i}$

Molte volte è più facile massimizzare il logaritmo dato che è monotono crescente:

$log(L(x_1,\dots,x_n;p)) = \sum_i x_1 log\, p + (\sum_i 1 - x_i) log\,(1-p)$

Gli x_i li conosco? Come trovo il massimo?

- Faccio la derivata e la pongo uguale a zero

$\frac{d}{dp} log(L(x_1,\dots,x_n;p))$
$\Rightarrow \frac{\sum_i x_i}{p} + \frac{\sum_i (1-x_i)}{1-p} \cdot (-1) = 0$

$$\frac{\sum_i x_i}{\widehat{p}} = \frac{\sum_i (1-x_i)}{1-\widehat{p}}$$

$$\Rightarrow (1-\widehat{p})\sum_i x_i = \widehat{p}\sum_i (1-x_i) \Rightarrow \sum_i x_i - \widehat{p}\sum_i x_i = n\widehat{p} - \widehat{p}\sum_i x_i$$

$$\Rightarrow \widehat{p} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Media campionaria, stimatore del nostro p, valore atteso.

Altro esempio:
Caso della distribuzione esponenziale

$$f_X(x) = \lambda e^{-\lambda x}$$

$$L(x_1, \dots, x_n; \lambda) = \prod_{i=1}^{n} \lambda e^{-\lambda x} = \lambda^n e^{-\lambda \sum_{i=1}^{n} x_i}$$

Applico il logaritmo:

$$log(L(x_1, \dots, x_n; \lambda))$$

$$= n log(\lambda) - \lambda \sum_{i=1}^{n} x_i$$

Derivo tutto e pongo uguale a zero:

$$\frac{d}{d\lambda} log(L(x_1, \dots, x_n; \lambda))$$

$$= \frac{n}{\lambda} - \sum_{i=1}^{n} x_i = 0$$

$$\Rightarrow \frac{n}{\widehat{\lambda}} = \sum_{i=1}^{n} x_i$$

$$\Rightarrow \widehat{\lambda} = \frac{n}{\sum_{i=1}^{n} x_i} = \frac{1}{\overline{X}}$$

Altro esempio:

$$X \sim N(\mu, \sigma)$$

$$L(x_1, \dots, x_n; \mu, \sigma) = \prod_{i=i}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x_i-\mu)^2}$$

$$= \left(\frac{1}{\sqrt{2\pi}}\right)^n \cdot \left(\frac{1}{\sigma}\right)^n \cdot e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2}$$

Applico il logaritmo:

$$log(L(x_1, \dots, x_n; \mu, \sigma))$$

$= nlog(\frac{1}{\sqrt{2\pi}}) - nlog(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$

Come faccio a fare la derivata con due variabili?
Saremo nel caso in cui uno dei due sia noto.
(Provare da soli)

Adesso procediamo nel caso in cui essi siano entrambe sconosciute.

Inciso:
Derivate parziali
Derivate che considerano una delle due fisse alla volta, per poi mettere a sistema le due soluzioni:

$\frac{d}{d\mu}log(L(\mu, \sigma))$

$=-\frac{1}{2\sigma^2}\sum_{i=1}^{n}2(x_i - \mu) = -\frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \mu)$

$\frac{d}{d\sigma}log(L(\mu, \sigma))$

$=-\frac{n}{\sigma} + (\sum_{i=1}^{n}(x_i - \mu)^2)(-\frac{1}{2}(-2\sigma^{-3})) = \frac{\sum_{i=1}^{n}(x_i-\mu)^2}{\sigma^3} - \frac{n}{\sigma}$

Mettiamole a sistema e uguali a 0:

1) $-\frac{1}{\widehat{\sigma^2}}\sum_{i=1}^{n}(x_1 - \widehat{\mu}) = 0$
2) $-\frac{n}{\sigma} + \frac{1}{\widehat{\sigma^3}}\sum_{i=1}^{n}(x_i - \mu)^2 = 0$

1) $\sum_{i=1}^{n}x_i - n\widehat{\mu} = 0$
2) $\frac{1}{\widehat{\sigma^2}}\sum_{i=1}^{n}(x_i - \widehat{\mu})^2 = n$

1) $\mu = \frac{1}{n}\sum_{i=1}^{n}x_i$ media campionaria $= \overline{X}$
2) $\widehat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{X})^2$ circa la varianza campionaria

Lezione dell'1 Giugno 2023

Lezione 24
Esercitazione con ripasso
Esame:

1) Probabilità e variabili aleatorie
2) Statistica inferenziale quindi ci saranno anche delle domande teoriche
3) Statistica descrittiva, esplorazione di un dataset
4) Esplorazione di un dataset, applicare quindi un po' di statistica descrittiva e inferenziale sapendo l'ipotesi

Avremo:
Jupyter, terminale e browser con upload per scaricare il tema d'esame, il dataset e per caricare la soluzione in un file unico (il tema d'esame sarà un notebook).
Avremo python, shepy, numpy.
Potremo portare al massimo il libro e un quaderno (più efficace con le formule per memorizzarle e enunciati dei teoremi).

Esercizio 1:
Sia X una variabile aleatoria di Poisson, e sia λ il numero medio di eventi che accadono in un intervallo di ampiezza prefissata.
Punto 1:

- Quali valori può assumere X?

$N \cup \{0\}$

- Si esprima, in funzione di λ, la probabilità $P(X = k)$ che accadano esattamente k eventi nell'intervallo considerato

$X \sim P(\lambda) \qquad P(X = k) = f_X(k) = e^{-\lambda}\frac{\lambda^k}{k!}$

- Si esprimano, in funzione di λ, valore atteso e varianza di X

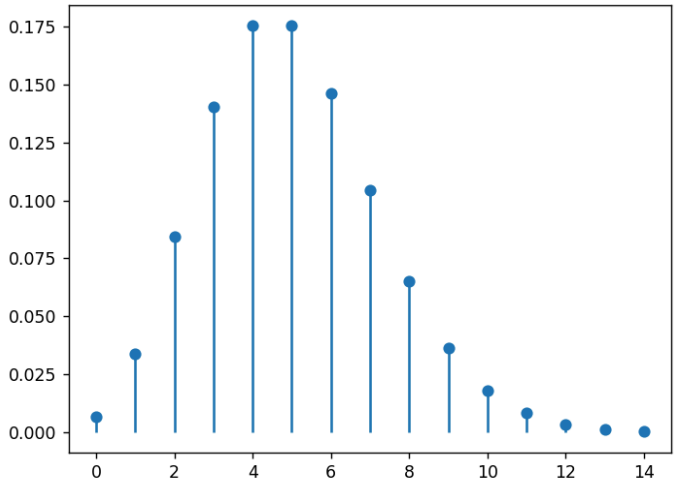
$E(X) = \lambda \qquad Var(X) = \lambda$
(In funzione di = scrivere la formula senza calcolarla).

Punto 2:
Fissiamo solo in questo punto $\lambda = 5$

- Si tracci il grafico della funzione di massa di probabilità di X

```
import math
import matplotlib.pyplot as plt

def p_poisson(x, l):
    return l**x * math.exp(-1 * l) / math.factorial(x)

l = 5
x = range(15)
p_x = list(map(lambda _: p_poisson(_, l), x))
```

*plt.vlines(x, [0]*len(x), p_x)*
plt.plot(x, p_x, 'o')
plt.show()



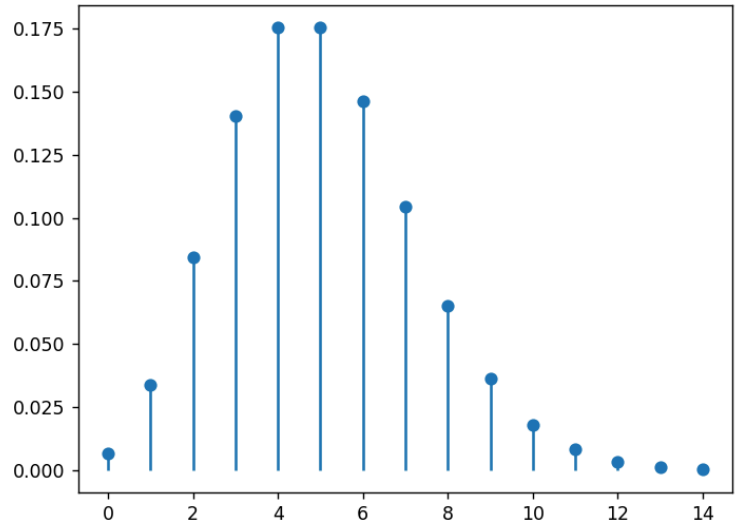
Potevamo ottenere lo stesso risultato utilizzando il package "scipy.stats", il quale permette di creare un oggetto pari ad una distribuzione.

import scipy.stats as st
import matplotlib.pyplot as plt

l = 5
x = range(15)

X = st.poisson(l)

#X.pmf(x)
#metodo per calcolare la funzione di massa di probabilità, possiamo passare uno o un array di valori
plt.vlines(x, 0, X.pmf(x))
plt.plot(x, X.pmf(x), 'o')
plt.show()



- Si calcoli la probabilità che X sia maggiore di 6

$P(X > 6) = 1 - P(X \leq 6) = 1 - F_X(6)$

```
import scipy.stats as st
l = 5
X = st.poisson(l)

a = 1 - sum([X.pmf(i) for i in range(7)])
print(a)
#altro metodo per calcolare direttamente la funzione di ripartizione
b = 1 - X.cdf(6)
print(b)
```

Output:
0.23781653702706163
0.2378165370270613

- Si determini il più piccolo valore di X, tale che la sua funzione di ripartizione sia > 0.8

```
c = 0
while X.cdf(c) < 0.8:
    c += 1

print(c)
```

Output:
7

Per esser sicuri di non aver fatto errori controlliamo che $F_X(6)$ sia minore di 0.8 mentre $F_X(7)$ non lo sia.

print((X.cdf(6), X.cdf(7)))

Output:
(0.7621834629729387, 0.8666283259299925)

Inoltre possiamo controllare che 7 sia il quantile 0.8 di X, attraverso il metodo specifico:

print(X.ppf(0.8))

Output:
7.0

Punto 3:

- Dati $a, b \in \Re$, sia $Y := aX - b$, calcolare in funzione di λ, a e b il valore atteso e la varianza di Y

$E(Y) = E(aX - b) = aE(X) - b = a\lambda - b$

$Var(Y) = Var(aX - b) = a^2 Var(X) = a^2\lambda$

Esercizio 2:
Un'azienda è proprietaria di alcune sorgenti di alta montagna. L'azienda ha da poco acquistato un dispositivo per il filtraggio dell'acqua, che filtra 5 litri di acqua per volta e, per ogni operazione, registra su un file alcune informazioni tra cui il nome della sorgente dalla quale è stata prelevata l'acqua, la durezza dell'acqua analizzata, il numero di particelle riscontrate di alcuni elementi (per esempio magnesio, sodio, oro, argento, ferro, piombo, iodio). Dalle prime prove di utilizzo si è notato con sorpresa che, nel file prodotto dal dispositivo, la colonna corrispondente all'oro non contiene sempre il valore zero, quindi nell'acqua si possono trovare tracce di oro.
Il file "ComposizioneAcqua.csv" (contenuto nella directory "data") contiene i dati che siamo interessati ad analizzare. Ecco la descrizione degli attributi:

- NomeSorgente: nome della sorgente dalla quale è stata prelevata l'acqua
- Oro: numero di particelle d'oro riscontrate in 5 litri d'acqua
- DurezzaAcqua: durezza dell'acqua (indice legato alla presenza di calcio)

Punto 1:

- Quanti casi sono presenti nel dataset?

```
import pandas as pd

acqua = pd.read_csv('ComposizioneAcqua.csv')
print(acqua.head())

#il numero di casi del dataset si ottiene semplicemente con il metodo len
print(len(acqua))
```

	NomeSorgente	Oro	DurezzaAcqua
0	Sorgente1	0	21.201381
1	Sorgente1	0	25.294662
2	Sorgente1	4	12.435279
3	Sorgente1	3	16.146828
4	Sorgente1	0	21.091517

1650 casi.

Punto 2:
- Quanti litri d'acqua sono stati analizzati complessivamente?

Dato che ogni caso corrisponde a 5 litri d'acqua analizzati, basta moltiplicare i casi per 5.

```
#litri d'acqua analizzati
print(len(acqua) * 5)
```

Output:
8250

Punto 3:
- Quante sono le sorgenti dalle quali sono stati prelevati i campioni d'acqua?

E' necessario estrarre la serie dell'attributo da analizzare con valori unici, definiamo una funzione che ne calcoli la lunghezza automaticamente:

```
#funzione per calcolare il numero di valori unici in una serie
```

```
def num_values(series):
    return len(series.unique())

print(num_values(acqua['NomeSorgente']))
```

Output:
5

Punto 4:

- Le diverse sorgenti sono rappresentate in modo uniforme nel dataset?

Il modo in cui le diverse sorgenti sono rappresentate nel dataset viene calcolato in termini della rispettiva eterogeneità. A sua volta, l'eterogeneità si può calcolare usando diversi indici. Richiamiamo qui di seguito l'implementazione del calcolo dell'indice di Gini nella sua versione originale (la funzione "gini") e in quella normalizzata (la funzione "normalized_gini", che utilizza la funzione "num_values" definita al punto precedente).

```
#funzioni di gini
def gini(series):
    return 1 - sum(series.value_counts(normalize=True)
                   .map(lambda f: f**2))

def normalized_gini(series):
    s = num_values(series)
    return s * gini(series) / (s-1)

#calcolo dell'indice di eterogeneità di gini per il dataset
print(normalized_gini(acqua['NomeSorgente']))
```

Output:
0.980257116620753

Il valore è prossimo all'unità, denotando dunque un'elevata eterogeneità che corrisponde a un alto livello di uniformità per le sorgenti.

Punto 5:

- Si calcoli la tabella delle frequenze delle particelle d'oro su 5 litri d'acqua

Non viene specificato se abbiamo bisogno di frequenze relative o assolute (cambia solo "normalized = True").
Basta invocare il metodo "value_counts" (con normalize = True per ottenere quelle relative e sort = False per ordinare in modo crescente).

```
#frequenze relative con value_counts()
gold_rel_freq = acqua['Oro'].value_counts(normalize=True, sort=False)
print(gold_rel_freq)

#possiamo usare anche pd.crosstab()
pd.crosstab(index=acqua['Oro'],
        columns=['Frequenza'],
        normalize=True,
        colnames=[''])
```

```
0     0.276364
1     0.296364
2     0.198788
3     0.086667
4     0.048485
5     0.038182
6     0.029091
7     0.012121
8     0.007273
9     0.002424
10    0.002424
11    0.001818
Name: Oro, dtype: float64
```

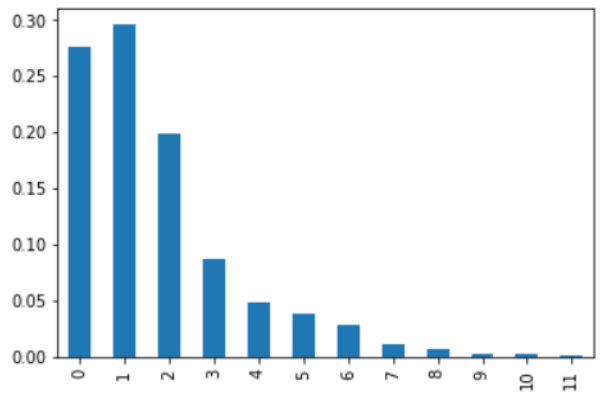
Oro	Freqenza
0	0.276364
1	0.296364
2	0.198788
3	0.086667
4	0.048485
5	0.038182
6	0.029091
7	0.012121
8	0.007273
9	0.002424
10	0.002424
11	0.001818

Punto 6:

- Si tracci un grafico opportuno per la visualizzazione di tali frequenze

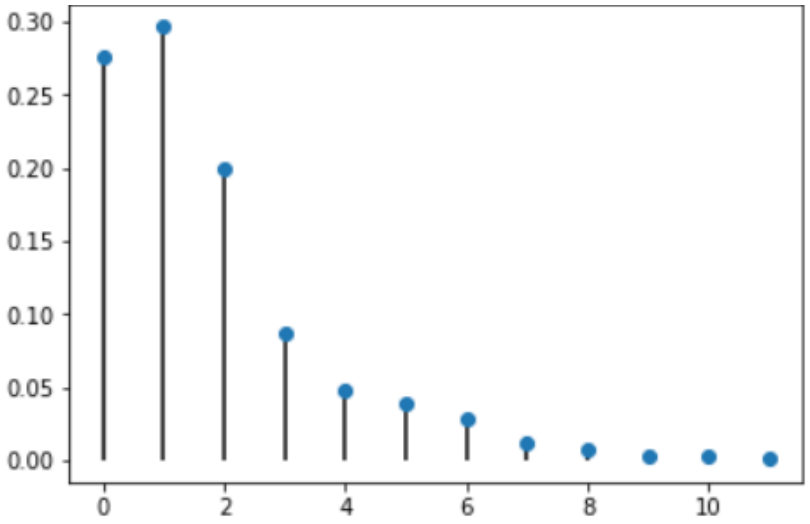
Un modo veloce di farlo è generare un grafico a barre:

```
#grafico a barre
gold_rel_freq.plot.bar()
plt.show()
```



Essendo però l'attributo di tipo numerico, si corre il rischio che ogni barra venga percepita come associata più a un intervallo di valori piuttosto che a un unico numero. Per evitare questo fraintendimento è più opportuno generare un grafico a bastoncini, come già fatto nel primo esercizio.

```
#grafico a bastoncini
plt.vlines(gold_rel_freq.index, 0, gold_rel_freq.values)
plt.plot(gold_rel_freq.index, gold_rel_freq.values, 'o')
plt.show()
```



Punto 7:

- La distribuzione delle frequenze osservate è compatibile con un modello di Poisson?

Osservando il grafico precedente notiamo che la distribuzione sarebbe compatibile con la Poisson.
Per essere più specifici però potremmo calcolare la media e la varianza per vedere se coincidono:

```
#media
print(acqua['Oro'].mean())

#varianza
print(acqua['Oro'].var())
```

Output:
1.7224242424242424
3.4025896319164715

I due valori sono sensibilmente diversi, quindi non è riconducibile ad un modello di Poisson nel quale media e varianza coincidono.

Punto 8:

- Si stimi il valore atteso di particelle d'oro in 5 litri d'acqua

Lo abbiamo già stimato nel punto precedente calcolando la media:
1.7224242424242424

Punto 9:

- Sia X la variabile casuale che conta il numero di particelle di oro riscontrate in 5 litri di acqua. Scrivere lo stimatore utilizzato al punto precedente, specificare la numerosità del campione a cui è applicato e dire se è uno stimatore non distorto.

Come già detto nel punto precedente, lo stimatore usato precedentemente è la media campionaria.
Utilizzando X variabile aleatoria corrispondente alla popolazione,

X_1, … , X_n campione e $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$.

Per controllare se la numerosità del campione corrisponde al numero di analisi, controlliamo che non ci siano valori nulli dell'attributo:

```
#controllo che non ci siano valori nulli
#primo metodo, conto i valori nulli
print(pd.isnull(acqua['Oro']).sum() != 0)
#secondo metodo, restituisce True se c'è un valore null, false altrimenti
print(pd.isnull(acqua['Oro']).any())
```

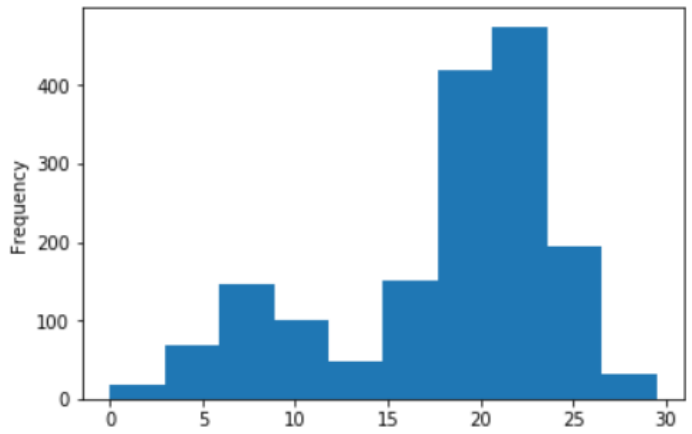
Entrambi False.
Inoltre sappiamo che la media campionaria è uno stimatore non distorto del valore atteso.

Esercizio 3:
Punto 1:

- Si tracci un grafico opportuno per rappresentare la durezza dell'acqua

A differenza dell'attributo "Oro", qui abbiamo dei valori reali, quindi è più opportuno utilizzare un istogramma.

```
#rappresentazione grafica dell'attributo "DurezzaAcqua"
acqua['DurezzaAcqua'].plot.hist()
plt.show()
```

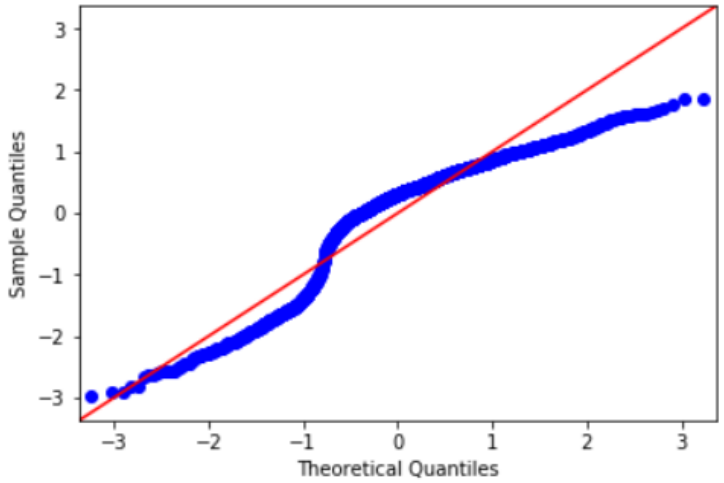


Punto 2:

- La distribuzione delle frequenze osservate è compatibile con un modello normale?

Dal grafico precedente si vede come l'istogramma abbia una forma bimodale, cioè con due massimi locali. Il modello normale è invece caratterizzato da una forma a campana. Pertanto i dati osservati non sono compatibili con una distribuzione normale. Possiamo approfondire l'analisi visualizzando il diagramma Q-Q per l'attributo e sovrapponendolo a una linea che indica la curva attesa in caso di normalità dei dati. Per fare ciò possiamo utilizzare la funzione "qqplot" del package "statmodels.api":

```
#visualizzazione del grafico qq grazie al metodo del package statsmodels.api
import statsmodels.api as sm

sm.qqplot(acqua['DurezzaAcqua'], fit=True, line='45')
plt.show()
```



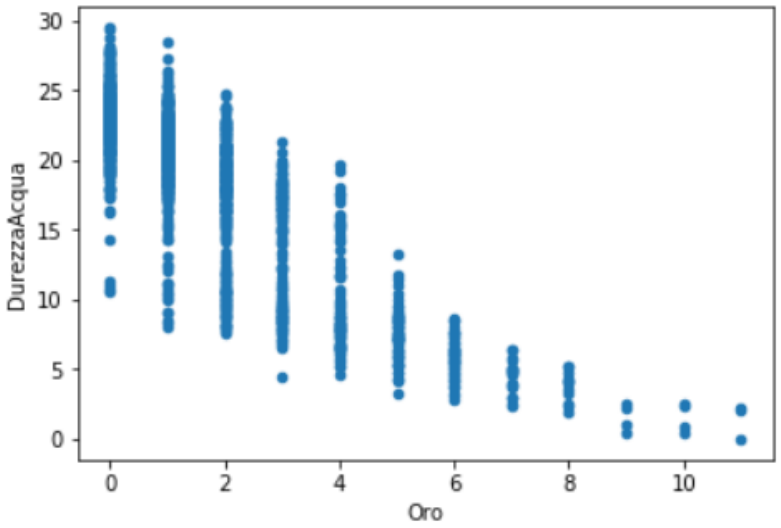
- fit = True è per normalizzare i dati in modo che la curva si sovrapponga alla bisettrice del primo e del terzo quadrante
- line = '45' è per visualizzare la suddetta bisettrice

Punto 3:

- L'ufficio analisi chimiche ipotizza che ci sia una relazione tra la quantità di oro riscontrata e la durezza dell'acqua. Si produca un grafico e si calcoli un indice numerico per convincere il titolare della ditta che l'ipotesi è accettabile.

Per valutare un'eventuale relazione tra i due attributi tracciamo il corrispondente grafico di dispersione:

```
#grafico di dispersione correlato
acqua.plot.scatter('Oro', 'DurezzaAcqua')
plt.show()
```



In effetti il grafico evidenzia una relazione lineare di tipo inverso.

Calcoliamo adesso il coefficiente di correlazione lineare:

#calcolo del coefficiente di correlazione lineare
acqua['Oro'].corr(acqua['DurezzaAcqua'])

Output:
-0.8427686305078547

Punto 4:
- Che tipo di relazione avete riscontrato?

Il fatto che il coefficiente di correlazione sia sufficientemente vicino a -1 ci indica che esiste una relazione inversa lineare fra i due attributi.

Punto 5:
- Osservando i grafici prodotti nei primi due punti di questo esercizio, ai chimici viene il sospetto che non tutte le sorgenti siano caratterizzate dalla stessa durezza media dell'acqua, e in particolare che nel dataset ci siano due gruppi distinti dal punto di vista della durezza dell'acqua. Si valuti se questa ipotesi è condivisibile.

L'ipotesi viene confermata dal primo grafico che presenta una bimodalità, quindi una sovrapposizione di due campane.

Punto 6:
- Si calcoli la durezza media dell'acqua per ogni sorgente d'acqua.

Il metodo "groupby" è possibile suddividere un dataframe in gruppi diversi in funzione del valore assunto da un determinato attributo. Invocando inoltre il metodo ".mean()" sul risultato è possibile quindi applicare la media degli attributi restanti.

#calcolo del valore medio per ogni sorgente
print(acqua.groupby('NomeSorgente').mean())

NomeSorgente	Oro	DurezzaAcqua
Sorgente1	0.970000	21.009675
Sorgente2	1.048000	20.836725
Sorgente3	1.120000	20.696062
Sorgente4	1.068000	20.917371
Sorgente5	4.228571	7.754144

Dal risultato si vede come le prime quattro sorgenti abbiano valori molto vicini alla media dell'attributo “DurezzaAcqua”, mentre l'ultima media risulta sensibilmente minore.

Volendo si sarebbe potuto anche utilizzare un filtro, selezionando le righe del dataset corrispondenti via via a una sorgente specifica per poi calcolare manualmente le singole medie.
Esempio:

acqua[acqua['NomeSorgente']=='Sorgente1'].mean()

Output:
Oro 0.970000
DurezzaAcqua 21.009675
dtype: float64

Punto 7:

- Dal punto precedente dovrebbe essere emerso che la sorgente 5 presenta una durezza dell'acqua nettamente inferiore a quella delle altre sorgenti, le quali invece hanno una durezza media abbastanza simile. Selezionate e memorizzate nella variabile “sorgente_5” gli attributi Oro e DurezzaAcqua soltanto della sorgente 5; selezionate e memorizzate nella variabile “altre_sorgenti” i medesimi attributi per tutte le altre sorgenti.

Utilizziamo un filtro simile al precedente per la memorizzazione di queste due variabili:

#memorizzazione dei gruppi di sorgenti
sorgente_5 = acqua[acqua['NomeSorgente']=='Sorgente5']
altre_sorgenti = acqua[acqua['NomeSorgente']!='Sorgente5']

Punto 8:

- Si stimi il valore atteso del numero di particelle d’oro in 5 litri d’acqua nel caso della sorgente 5 e per tutte le altre sorgenti

Utilizziamo come stimatore la media campionaria per tutti e due i dataset:

#stima dei valori attesi di sorgente_5 e altre_sorgenti
print(sorgente_5['Oro'].mean())
print(altre_sorgenti['Oro'].mean())

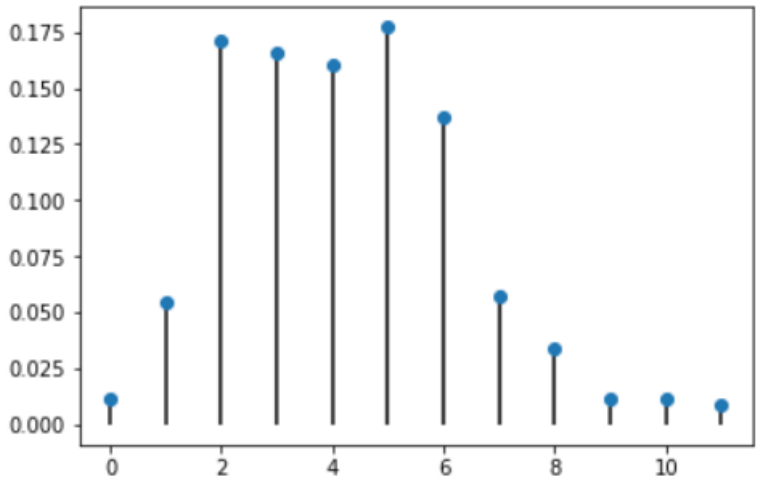
Output:
4.228571428571429
1.0476923076923077

Punto 9:

- La distribuzione delle frequenze osservate nella sorgente 5 è compatibile con un modello di Poisson?

Visualizziamo innanzitutto il grafico delle frequenze:

```
#grafico delle frequenze della sorgente 5
s5_rel_freq = sorgente_5['Oro'].value_counts(normalize=True, sort=False)
plt.vlines(s5_rel_freq.index, [0]*len(s5_rel_freq), s5_rel_freq.values)
plt.plot(s5_rel_freq.index, s5_rel_freq.values, 'o')
plt.show()
```



L'andamento unimodale ci conferma che la sorgente 5 possa essere distribuita come una Poisson.
Per confermarlo stimiamo media e varianza, nel caso siano uguali avrò un ulteriore conferma.

```
#stima della media e della varianza della sorgente 5
print(sorgente_5['Oro'].mean(), sorgente_5['Oro'].var())
```

Output:
(4.228571428571429, 4.38886614817847)

La loro somiglianza quindi ci riconferma una distribuzione di Poisson.

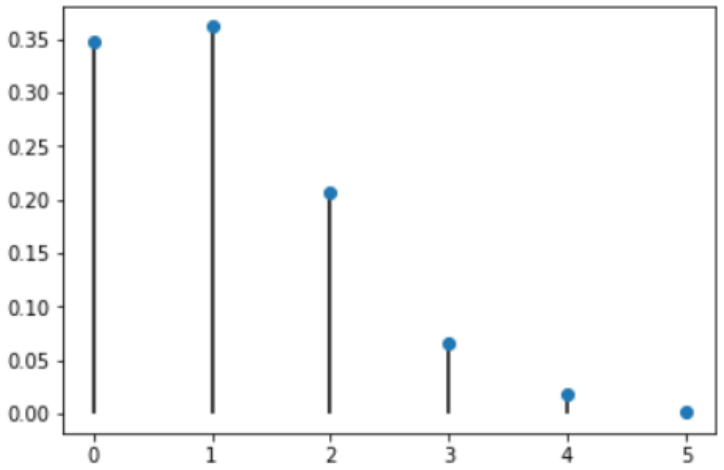
Punto 10:

- La distribuzione delle frequenze osservate nelle altre sorgenti è compatibile con quella di Poisson?

Tracciamo come prima il grafico:

```
#grafico delle frequenze delle altre sorgenti
```

```
altre_rel_freq = altre_sorgenti['Oro'].value_counts(normalize=True,
sort=False)
plt.vlines(altre_rel_freq.index, [0]*len(altre_rel_freq), altre_rel_freq.values)
plt.plot(altre_rel_freq.index, altre_rel_freq.values, 'o')
plt.show()
```



Anche qui abbiamo unimodalità, stimiamo media e varianza:

```
#stima della media e della varianza della sorgente 5
print(altre_sorgenti['Oro'].mean(), altre_sorgenti['Oro'].var())
```

Output:
(1.0476923076923077, 0.9923349321963675)

Anche qui abbiamo una distribuzione di Poisson.

Punto 11:

- Durante l'attività di filtraggio dalla sorgente 5, una particella di oro ha inceppato il dispositivo di filtraggio dell'acqua. Si calcoli la probabilità di non trovare altre particelle di oro nei prossimi 10 litri di acqua, dopo aver sbloccato il dispositivo.

Siccome il numero X di particelle d'oro per cinque litri di acqua provenienti dalla sorgente 5 è descrivibile tramite un modello di Poisson, la quantità di litri d'acqua da analizzare tra le due scoperte successive di una particella è descrivibile tramite una legge esponenziale di parametro uguale al valore atteso di X diviso 5.
Noi non conosciamo il valore atteso di X, ma lo abbiamo già stimato e a partire da tale stima possiamo ottenere il parametro per la legge esponenziale interessata.

#parametro
l_5 = sorgente_5['Oro'].mean()/5

Ora, indicata con T la variabile aleatoria che corrisponde alla quantità di litri d'acqua da analizzare prima di trovare la prossima particella d'oro, la probabilità di non trovare particelle nei prossimi 10 litri sarà:
$P(T > 10) = 1 - P(T \leq 10) = 1 - F_T(10)$
Per calcolare questa probabilità possiamo utilizzare il package "scipy.stats" che mette a disposizione una classe "expon" per le distribuzioni esponenziali.
Per motivi che a noi resteranno sconosciuti dovremmo passare il valore inverso come argomento "scale" per il parametro.

#creazione di una distribuzione esponenziale
T_5 = st.expon(scale=1/l_5)

Il valore cercato sarà quindi uguale a:

#valore da calcolare
print(1 - T_5.cdf(10))

Output:
0.00021237799893225606

Punto 12:

- Dal punto in cui il dispositivo applicato alla sorgente 5 ha individuato l'ultima particella di oro sono stati filtrati altri 5 litri d'acqua senza trovare altro oro. Qual è la probabilità di non trovare oro ancora per i prossimi 10 litri d'acqua?

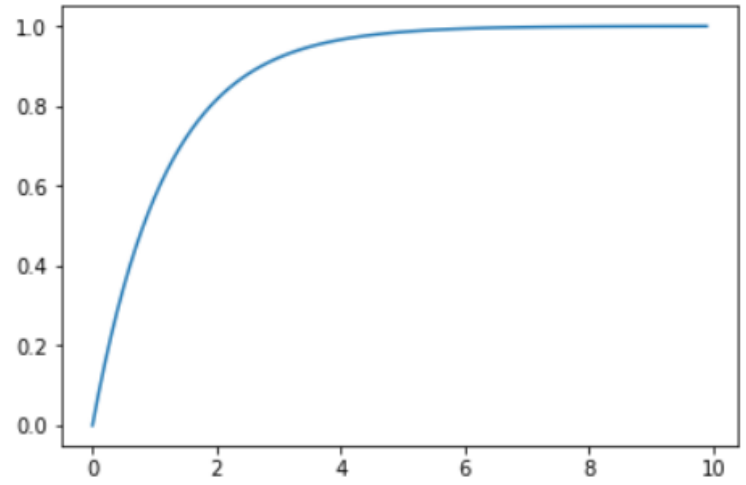
La proprietà di assenza di memoria della distribuzione esponenziale ci permette di dire che la probabilità che non si trovino particelle nei dieci litri successivi all'ultima scoperta è uguale alla probabilità che i primi 10 litri analizzati da quando è partito il macchinario non abbiano trovato nessuna particella.
Quindi il risultato è uguale a quello precedente

Punto 13:

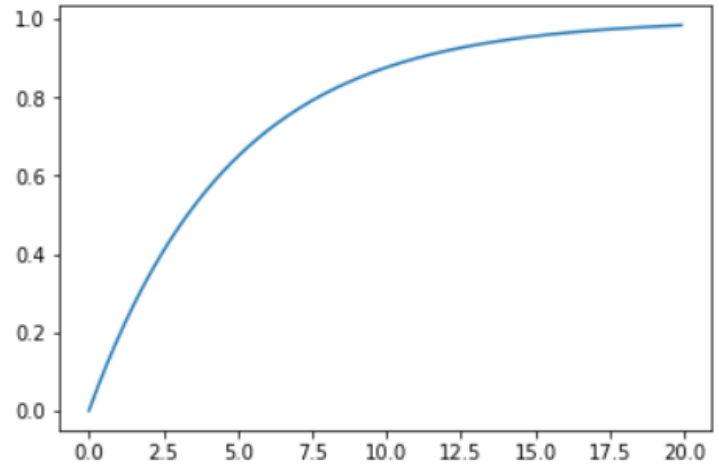
- Si tracci il grafico della funzione di ripartizione della variabile casuale Y = "quantità di acqua da analizzare (espressa in litri" prima di incontrare la prossima particella di oro" nel caso della sorgente 5 e nel caso delle altre sorgenti

Per quanto visto ai punti precedenti, la variabile aleatoria Y segue una legge esponenziale descritta dall'oggetto memorizzato in T_5. Il grafico della corrispondente funzione si ottiene nel modo seguente:

```
#grafico della funzione di ripartizione di T_5
import numpy as np

t = np.arange(0, 10, .1)
plt.plot(t, T_5.cdf(t))
plt.show()
```




```
#grafico della funzione di ripartizione della variabile aleatoria esponenziale per le altre sorgenti
l_altre = altre_sorgenti['Oro'].mean()/5
T_altre = st.expon(scale=1/l_altre)
t = np.arange(0, 20, .1)
plt.plot(t, T_altre.cdf(t))
plt.show()
```



Esercizio 4:
Punto 1:

- Secondo il titolare dell'azienda il valore atteso di particelle di oro riscontrate è abbastanza elevato da poter pensare di estrarre l'oro dall'acqua per venderlo. Prima di iniziare questa nuova attività è bene soppesare l'errore compiuto nella stima di tale parametro. Calcolare la probabilità che, per la sorgente 5, l'errore compiuto nella stima del valore atteso di particelle di oro riscontrate in 5 litri d'acqua sia al più di 0.1 particelle, in eccesso o in difetto.

Indicando con $\overline{X}$ la media campionaria, con λ il valore atteso che si vuole stimare e con σ_X la deviazione standard della popolazione, fissato un generico $\varepsilon > 0$ e applicando il teorema centrale del limite si ha:

$$P(|\overline{X} - \lambda| < \varepsilon) = P(-\varepsilon < \overline{X} - \lambda < \varepsilon) = P(-\frac{\varepsilon}{\sigma_X/\sqrt{n}} < \frac{\overline{X}-\lambda}{\sigma_X/\sqrt{n}} < \frac{\varepsilon}{\sigma_X/\sqrt{n}})$$

$$\approx P(-\frac{\varepsilon}{\sigma_X/\sqrt{n}} < Z < \frac{\varepsilon}{\sigma_X/\sqrt{n}}) = \Phi(\frac{\varepsilon}{\sigma_X/\sqrt{n}}) - \phi(-\frac{\varepsilon}{\sigma_X/\sqrt{n}})$$

$$= \Phi(\frac{\varepsilon}{\sigma_X/\sqrt{n}}) - (1 - \Phi(\frac{\varepsilon}{\sigma_X/\sqrt{n}})) = 2\Phi(\frac{\varepsilon}{\sigma_X/\sqrt{n}}) - 1$$

Nel nostro caso $\varepsilon = 0.1$, n è pari al numero di casi del dataframe memorizzato in "sorgente_5" e σ_X può essere approssimato calcolando la deviazione standard campionaria:

```
#calcolo del teorema centrale del limite
sigma_x = sorgente_5['Oro'].std()
n = len(sorgente_5)
eps = 0.1

Z = st.norm()
print(2 * Z.cdf(eps * n ** 0.5 / sigma_x) - 1)
```

Output:
0.6281498695096628

Punto 2:

- Quanti litri d'acqua si dovrebbero ancora analizzare affinché la probabilità che, per la sorgente 5, l'errore compiuto nella stima del valore atteso di particelle di oro riscontrate in 5 litri d'acqua sia al più di 0.1 particelle, in eccesso o in difetto sia almeno uguale a 0.9?

A partire da quanto visto nel punto precedente dovrà essere:

$$2\Phi(\frac{\varepsilon}{\sigma_X/\sqrt{n}}) - 1 > 0.9$$

$\Rightarrow \Phi(\frac{\varepsilon}{\sigma_X/\sqrt{n}}) > 0.95$

Applicando l'inverso di Φ membro a membro e risolvendo rispetto a n si ottiene:

$n > [(\frac{\sigma_X}{\varepsilon}\Phi^{-1}(0.95))^2]$ parte intera superiore

Il calcolo di Φ^{-1} corrisponde al calcolo dei quantili, quindi possiamo ottenere il valore di "n_0" che indica il numero minimo di osservazioni da effettuare nel modo seguente:

```
#calcolo del numero minimo per eccedere di 0.9
n_0 = math.ceil((sigma_x / eps * Z.ppf(0.95))**2)
print(n_0)
```

Output:
1188

Concludendo, sarà necessario effettuare ancora un numero di osservazioni pari a:

```
print(n_0 - len(sorgente_5))
```

Output:
838

Che corrispondono a tali litri d'acqua:

```
print(5 * (n_0 - len(sorgente_5)))
```

Output:
4190

Punto 3:

- Per semplicità ipotizziamo che una particella estratta dalla sorgente 5 possa essere venduta a 1.5 euro e che il dispositivo per l'estrazione dell'oro abbia un costo fisso di 1000 euro. Stimare il guadagno atteso nel caso in cui si voglia estrarre l'oro da 1000 litri di acqua e la varianza del guadagno.

L'oro estratto in 1000 litri d'acqua della sorgente 5 corrisponde a 200 analisi fatte ognuna su cinque litri, e quindi il numero di particelle corrispondenti è pari alla somma di 200 variabili poissoniane $X_1, \dots, X_{200}$ aventi tutte lo stesso parametro λ (la cui stima abbiamo

precedentemente eseguito tramite media). Tale somma N avrà ancora distribuzione poissoniana con parametro uguale a 200λ.
Il guadagno sarà quindi pari a $G = 1.5N - 1000$ e dunque:
$E(G) = 1.5E(N) - 1000 = 1.5 \cdot 200\lambda - 1000 = 100(3\lambda - 10)$
Possiamo stimare il guadagno atteso come:

```
#stima del guadagno atteso
guad_att = 100 * (3 * sorgente_5['Oro'].mean() - 10)
print(guad_att)
```

Output:
268.57142857142867

La varianza di G sarà invece uguale a:
$Var(G) = 1.5^2 Var(N) = 1.5^2 200\lambda$
Possiamo stimare questo valore con (con tanto di deviazione standard):

```
#stima della varianza e della deviazione standard
var_g = 1.5**2 * 200 * sorgente_5['Oro'].mean()
print(var_g**0.5)
```

Output:
43.62175079999819