

# Cheatsheet Statistica

Giacomo Comitani

June 2025

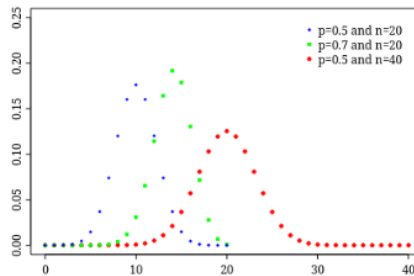
## 1 Modelli Statistici

- **Bernoulli:**  $X \sim B(p)$  : Esperimento avente solamente due possibili esiti: successo o fallimento. **Discreto**

- Massa:  $p^x(1-p)^{1-x} \mathbb{I}_{\{0,1\}}(x)$
- Ripartizione:  $(1-p) \mathbb{I}_{[0,1)}(x) + \mathbb{I}_{[1,+\infty)}(x)$
- Valore atteso:  $p$
- Varianza:  $p(1-p)$

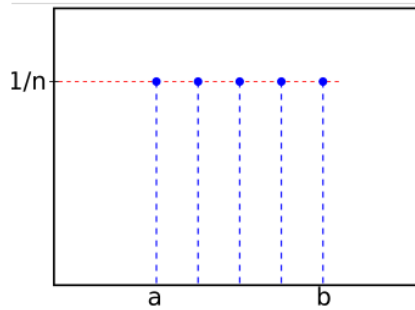
- **Binomiale:**  $X \sim B(n, p)$ : Tanti eventi bernoulliani in serie. **Discreto**

- Massa:  $\binom{n}{x} p^x (1-p)^{n-x} \mathbb{I}_{\{0, \dots, n\}}(x)$
- Ripartizione:  $\left( \sum_{i=0}^{\lfloor x \rfloor} \binom{n}{i} p^i (1-p)^{n-i} \right) \mathbb{I}_{[0, n]}(x) + \mathbb{I}_{(n, +\infty)}(x)$
- Valore atteso:  $np$
- Varianza:  $np(1-p)$
- Proprietà: Riproducibilità



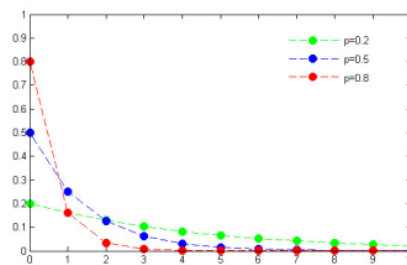
- **Uniforme discreto:**  $X \sim U(n)$ : Gli esiti della variabile aleatoria sono equiprobabili. **Discreto**

- Massa:  $\frac{1}{n} \mathbb{I}_{\{1, \dots, n\}}(x)$
- Ripartizione:  $\frac{\lfloor x \rfloor}{n} \mathbb{I}_{[1, n]}(x) + \mathbb{I}_{(n, +\infty)}(x)$
- Valore atteso:  $\frac{n+1}{2}$
- Varianza:  $\frac{n^2-1}{12}$



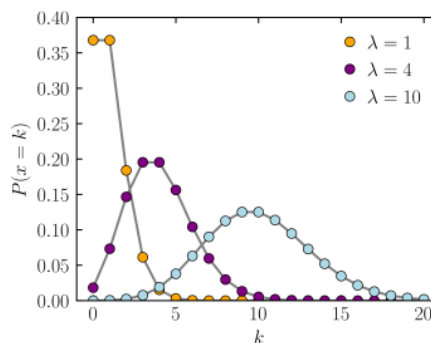
- **Geometrico:**  $X \sim G(p)$ : La variabile assume il numero di insuccessi consecutivi prima che si verifichi un successo in una serie di esperimenti Bernoulliani indipendenti e identicamente distribuiti. **Discreto**

- Massa:  $p(1-p)^x \mathbb{I}_{\{0,1,2,\dots\}}(x)$
- Ripartizione:  $(1 - (1-p)^{\lfloor x \rfloor + 1}) \mathbb{I}_{[0,+\infty)}(x)$
- Valore atteso:  $\frac{1-p}{p}$
- Varianza:  $\frac{1-p}{p^2}$
- Proprietà: Assenza di memoria



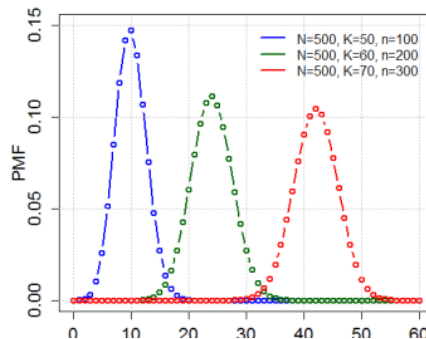
- **Poisson:**  $X \sim P(\lambda)$ : La variabile assume il numero di eventi che si verificano in un dato intervallo di tempo, sapendo che mediamente se ne verificano un numero  $(0, +\infty)$ . Tutti gli eventi sono indipendenti. **Discreto**

- Massa:  $\frac{e^{-\lambda} \lambda^x}{x!} \mathbb{I}_{\{0,1,2,\dots\}}(x)$
- Ripartizione: *NON vista nel corso*
- Valore atteso:  $\lambda$
- Varianza:  $\lambda$
- Proprietà: Approssimazione binomiale, riproducibilità



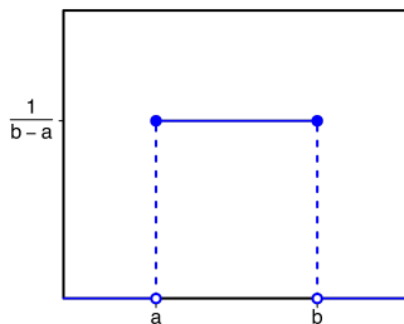
- **Ipergeometrico:**  $X \sim H(n, M, N)$ : La variabile assume il numero di oggetti corretti estratti da un'urna di oggetti binari durante un'estrazione senza reimmissione dopo  $n$  estrazioni. **Discreto**

- Massa:  $\frac{\binom{N}{x} \binom{M}{n-x}}{\binom{N+M}{n}} \mathbb{I}_{\{0, \dots, n\}}(x)$
- Ripartizione: *NON vista nel corso*
- Valore atteso:  $\frac{nN}{N+M}$
- Varianza:  $\frac{n(N+M-n)NM}{(N+M)^2(N+M-1)}$



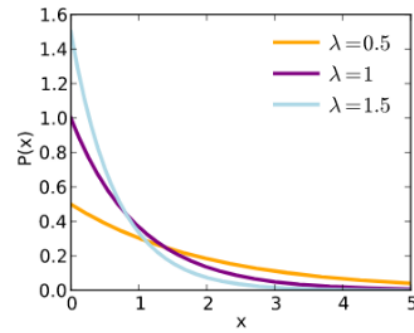
- **Uniforme continuo:**  $X \sim U(a, b)$ : Gli esiti della variabile aleatoria sono tutti equiprobabili. **Continuo**

- Densità:  $\frac{1}{b-a} \mathbb{I}_{[a,b]}(x)$
- Ripartizione:  $\frac{x-a}{b-a} \mathbb{I}_{[a,b]}(x) + \mathbb{I}_{(b,+\infty)}(x)$
- Valore atteso:  $\frac{a+b}{2}$
- Varianza:  $\frac{(b-a)^2}{12}$



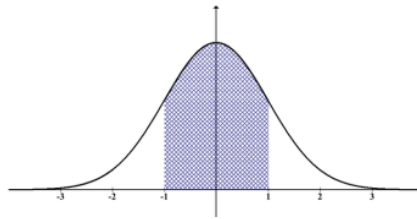
- **Esponenziale:**  $X \sim E(\lambda)$ : La variabile assume il tempo di attesa tra due eventi, che mediamente accadono ogni  $(0, +\infty)$  unità di tempo. **Continuo**

- Densità:  $\lambda e^{-\lambda x} \mathbb{I}_{[0,+\infty)}(x)$
- Ripartizione:  $(1 - e^{-\lambda x}) \mathbb{I}_{[0,+\infty)}(x)$
- Valore atteso:  $\frac{1}{\lambda}$
- Varianza:  $\frac{1}{\lambda^2}$
- Proprietà: Assenza di memoria, scalatura, proprietà su massimo e minimo

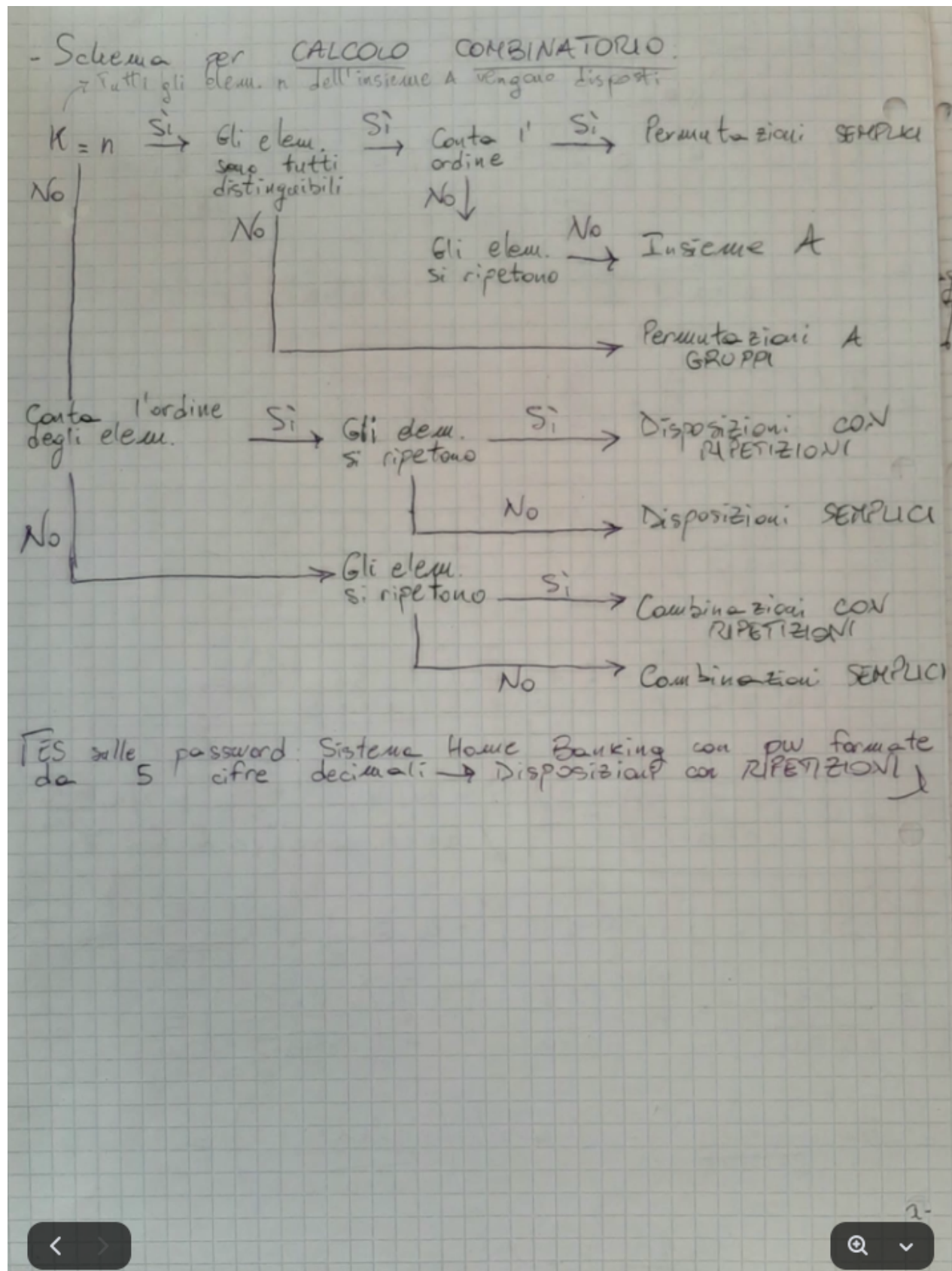


• **Gaussiana (Normale):**  $X \sim G(\mu, \sigma)$ . **Continuo**

- Densità:  $\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
- Ripartizione:  $\int_{-\infty}^x \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt$
- Valore atteso:  $\mu$
- Varianza:  $\sigma^2$
- Proprietà: Standardizzazione, riproducibilità



## 2 calcolo combinatorio



### 2.0.1 Formule di Calcolo Combinatorio

#### Fattoriale

$$n! = n \cdot (n - 1) \cdot (n - 2) \cdots 2 \cdot 1$$

#### Permutazioni

##### Permutazioni semplici (senza ripetizione)

$$P(n) = n!$$

##### Permutazioni con ripetizione

 Se alcuni elementi si ripetono:

$$P(n; n_1, n_2, \dots, n_k) = \frac{n!}{n_1! \cdot n_2! \cdots n_k!}$$

#### Disposizioni

##### Disposizioni semplici (senza ripetizione)

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

##### Disposizioni con ripetizione

$$D'_{n,k} = n^k$$

#### Combinazioni

##### Combinazioni semplici (senza ripetizione)

$$C_{n,k} = \binom{n}{k} = \frac{n!}{k!(n - k)!}$$

##### Combinazioni con ripetizione

$$C'_{n,k} = \binom{n + k - 1}{k} = \frac{(n + k - 1)!}{k!(n - 1)!}$$

### 2.0.2 Relazioni utili

$$\begin{aligned} \binom{n}{0} &= \binom{n}{n} = 1 & \binom{n}{1} &= \binom{n}{n-1} = n \\ \binom{n}{k} &= \binom{n-1}{k-1} + \binom{n-1}{k} & & \text{(relazione di Pascal)} \end{aligned}$$

Espansione del binomio di Newton:

$$(a + b)^n = \sum_{k=0}^n \binom{n}{k} a^{n-k} b^k$$

## 3 Analisi dei dati con python

### Librerie Importanti

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import scipy.stats as st
5 import statsmodels.api
6 import sklearn
7 import itertools
```

### Importazione e caricamento dei dati

```
1 import matplotlib.pyplot as plt
2 import pandas as pd
3 import numpy as np
4 import scipy.stats as st
5
6 ztl = pd.read_csv('ztl.csv', delimiter=';', decimal='.')
7 ztl
```

### Gestione Valori Mancanti

```
1 valMancanti = {col: ztl[col].isna().sum() for col in ztl.columns}
2 dataF = pd.DataFrame(index=valMancanti.keys(), data=valMancanti.values(), columns=['
3     Valori Mancanti'])
4 dataF
```

### Visualizzazione dei dati

La scelta del grafico dipende dalla **natura del dato** (qualitativo o quantitativo) e dalla **relazione tra le modalità**.

#### Dati qualitativi

- **Grafico a torta (pie chart):** adatto a variabili **nominali non ordinabili** (es. tipo\_motore). Evidenzia bene le proporzioni tra categorie distinte.

```
1 freq = auto['tipo_motore'].value_counts()
2 freq.plot.pie(autopct='%1.1f%%')
3 plt.show()
4
```

- **Grafico a barre (bar plot):**

- preferibile per variabili **ordinali** (es. livello\_rumore: silenzioso, medio, rumoroso), in cui l'ordine ha significato.

```
1 auto['livello_rumore'].value_counts().sort_index().plot.bar()
2 plt.show()
3
```

- **Caso binario (es. maschio/femmina):** entrambi i grafici (torta o barre) sono adatti. La torta mostra le proporzioni, le barre facilitano il confronto visivo diretto.

## Dati quantitativi

- **Istogramma:** usato per variabili **continue** (es. tempo, altezza), suddivide il dominio in intervalli (bin) e mostra la distribuzione.

```
1 data['tempo'].hist(bins=20)
2 plt.show()
3
```

- **Grafico a barre o vlins:** usato per variabili **discrete** (es. passaggi), con valori interi distinti. Le vlins sono più leggibili in presenza di molte modalità.

```
1 freq = ztl['passaggi'].value_counts()
2 plt.plot(freq.index, freq, 'o')
3 plt.vlines(freq.index, 0, freq)
4 plt.xlim(0, 30)
5 plt.show()
6
```

## Gestione Outlier

Per individuare e rimuovere gli outlier posso utilizzare il **boxplot**

```
1 # Visualizzarli graficamente
2 ztl['passaggi'].plot.box()
3 plt.show()
4
5 # Eliminarli dal dataset
6 q3 = ztl['passaggi'].quantile(0.75)
7 ztl_filtrato = ztl[ztl['passaggi'] <= q3].reset_index(drop=True)
```

## Tabelle di Frequenza

### Congiunta relativa

```
1 pd.crosstab(ztl['abbonamento'], ztl['altamente-inquinante'], normalize=True)
```

### Congiunta assoluta

```
1 pd.crosstab(data['attributo 1'], data['attributo 2'])
```

### Relativa semplice

```
1 ztl['abbonamento'].value_counts(normalize=True)
```

### Relativa cumulata

Serve anche a determinare se le due variabili sono **Identicamente distribuite**

```
1 ztl['passaggi'].value_counts(normalize=True).sort_index().cumsum()
```



## Cumulata con binning

```
1 auto['numero_occupanti'].value_counts(normalize=True, bins=10).sort_index().cumsum()
```

## Correlazione

```
1 ztl['abbonamento'].corr(ztl['passaggi'])
2 ztl.plot.scatter('abbonamento', 'passaggi')
3 plt.show()
```

- ris  $\rightarrow$  +1 probabile correlazione linearmente diretta
- ris  $\rightarrow$  0 correlazione improbabile
- ris  $\rightarrow$  -1 probabile correlazione linearmente indiretta

## Filtraggio Dati

```
1 ztl_giornalieri = ztl[ztl['abbonamento'] == 0]
```

## Indici Statistici

### Gini per concentrazione

```
1 def gini_concentrazione(series):
2     freqs = series.value_counts(normalize=True).sort_values().values
3     n = len(freqs)
4     if n < 2: return 0.0
5     Q = np.cumsum(freqs)[: -1]
6     F = np.arange(1, n) / n
7     return (F - Q).sum() / F.sum()
8
9 gini_concentrazione(auto['tipo_motore'].dropna())
```

### Gini per eterogeneità

```
1 def gini2(series):
2     return 1 - sum(series.value_counts(normalize=True).map(lambda f: f ** 2))
3
4 gini2(auto['tipo_motore'])
```

## Conversione booleana per correlazioni

```
1 accessi['allarme2'] = accessi['allarme'].apply(lambda x: x == 'ON')
2 accessi['carico_sistema'].corr(accessi['allarme2'])
```

## Indici Statistici

### Indici di centralità

Moda di un carattere

```
1 data['attributo'].mode()
```

Media campionaria di un carattere

```
1 data['attributo'].mean()
```

Mediana di un carattere

```
1 data['attributo'].median()
```

### Indici di dispersione

Varianza campionaria

```
1 data['attributo'].var()
```

deviazione standard campionaria

```
1 data['attributo'].std()
```

## Analisi Dataset

### Elemento massimo

```
1 dato = data[data['attributo'] == max(data['attributo'])]
```

### Elemento minimo

```
1 dato = data[data['attributo'] == min(data['attributo'])]
```

### Valori assumibili da un carattere

```
1 list(data['attributo'].unique())
```

### Tipo e forza di correlazione

```
1 data['attributo'].std()
```

### QQ Plot

```
1 import statsmodels.api as sm
2 mu = data['campo'].mean()
3 std = data['campo'].std()
4 sm.qqplot(data['campo'], dist = st.norm, loc=mu, scale=std, line='45', fit=True)
```

## PiePlot

```
1 freq = data['c1'].value_counts()
2 df = pd.DataFrame({'freq':freq})
3 print(df)
4
5 freq.plot.pie()
6 plt.show()
```

## Individuazione outlier (visivamente)

In questo caso conviene fare il box plot che posso fare con:

```
1 ztl['passaggi'].plot.box()
2 plt.show()
```

## Rimozione degli outlier

```
1 data = data[data['attributo'] <= data['attributo'].quantile(0.75(guardo boxplot))].
   reset_index(drop = True)
```

## Correlazione tra due attributi

- Faccio `.corr` per usare `Pearson` e poi confermo ipotesi con uno scatter:

```
1 ztl.plot.scatter('abbonamento', 'passaggi')
2 plt.show()
```

## ScatterPlot (2 parametri)

```
1 plt.scatter(heroes['height'], heroes['Weight'])
2 plt.show()
```

## ScatterPlot (3 parametri)

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

## Tabella delle frequenze relative cumulate

```
1 freq_rel_cumulate = auto['numero_occupanti'].value_counts(normalize = True).
   sort_index().cumsum()
2
3 # Qui il prof preferisce utilizzare i bins, occhio pero che cosi restituisce un
4 # intervallo e non un valore, quindi per i calcoli meglio usare senza bins
5
6 freq_rel_cumulate_print_prof = auto['numero_occupanti'].value_counts(normalize =
   True, bins = 10).sort_index().cumsum()
7
8 freq_rel_cumulate_print_prof
```

### Tabella delle frequenze relative NON cumulate

```
1 freq_rel_carpooling = auto_ridotto['carpooling'].value_counts(normalize = True)
2 freq_rel_carpooling
```

### Convertire valore da stringa a booleano

```
1 accessi['allarme2'] = accessi['allarme'].apply(lambda x : x == 'ON')
2
3 accessi['carico_sistema'].corr(accessi['allarme2'])
```

### Calcolo dimensione del campione togliendo tutti gli elementi mancanti

```
1 campione_senza_null = ztl_giornalieri['passaggi'] - ztl_giornalieri['passaggi'].
  isnull().sum()
2 n = len(campione_senza_null)
```

## 4 calcolo probabilità

Ricordiamo innanzitutto che, se ad esempio ci viene chiesto di calcolare  $P(X > 30)$ :

$$P(X > 30) = 1 - P(X \leq 30)$$

Il problema si pone quando la richiesta è di calcolare  $P(X \geq 30)$ :

- Nel caso delle **variabili aleatorie discrete**, il calcolo diventa:

$$P(X \geq 30) = 1 - P(X \leq 29)$$

poiché  $P(X \geq 30) = P(X = 30) + P(X = 31) + \dots$

- Nel caso delle **variabili aleatorie continue**, non si pone questo problema, perché:

$$P(X = 30) = 0$$

quindi si può scrivere semplicemente:

$$P(X \geq 30) = P(X > 30) = 1 - P(X \leq 30)$$

Proprietà del valore atteso

Posto ad esempio  $Z = 2X - Y$ :

- **Valore atteso**  $= E[Z] = E[2X - Y] = E[2X] - E[Y] = 2 \cdot p - p = 2p - p = p$
- **Varianza**:  $Var(Z) = Var(2X - Y) = Var(2X) + Var(-Y) = 4 \cdot Var(X) + Var(Y) = 4 \cdot [p(1 - p)] + [p(1 - p)] = 4 \cdot (p - p^2) + [p - p^2] = 4p - 4p^2 + p - p^2 = -5p^2 + 5p = 5p(1 - p)$

### Proprietà della varianza

La varianza della funzione indicatrice è la probabilità dell'evento moltiplicata per la probabilità dell'evento complementare:

$$VAR(I) = P(A) * P(\bar{A})$$

La varianza NON opera in modo lineare:  $VAR(aX + b) = a^2 VAR(X)$

La varianza della somma di due variabili aleatorie  $X$  e  $Y$  vale:

- $VAR(X + Y) = VAR(X) + VAR(Y) + 2COV(X, Y)$
- $VAR(X - Y) = VAR(X) + VAR(Y) - 2COV(X, Y)$

Se le due variabili sono **indipendenti identicamente distribuite**, allora la covarianza vale zero, quindi le formule di prima diventano:

- $VAR(X + Y) = VAR(X) + VAR(Y)$
- $VAR(X - Y) = VAR(X) + VAR(Y)$

Ricordo che la varianza della media campionaria vale:  $VAR(\bar{X}) = \frac{VAR(X)}{n}$

Ricordo infine che  $COV(X, Y) = E[XY] - E[X]E[Y]$

## 5 Esercizio 1

Visualizzare Graficamente una variabile aleatoria/ Visualizzarne le specificazioni

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 import scipy.stats as st
5
6 # Modifica i parametri
7 p = 0.7
8 n = 30
9 num_sample = 100000
10
11 # Modifica la distribuzione
12 binom = st.binom(n, p)
13 X = binom.rvs(num_sample)
14
15 # Modifica La variabile aleatoria
16 Z = 2 * X
17
18 specificazioni, freq_assolute = np.unique(Z, return_counts = True)
19 print("Specificazioni: ", specificazioni)
20 print("Numero di occorrenze per specificazione: ", freq_assolute)
21
22 pmf = freq_assolute / num_sample
23 plt.stem(specificazioni, pmf)
24
25 plt.show()
```

Tracciare una funzione generica

```
1 import pandas as pd
2 import scipy.stats as st
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 h = 4/10
7
8 def f(x):
9     return (1-(1-h)**(x+1))
10
11 x = np.linspace(1, 30, 100)
12 y = [f(n) for n in x]
13
14
15 plt.plot(x, y)
16
17 plt.grid(True)
18 plt.show()
```

### Tracciare la funzione di ripartizione

```
1  import pandas as pd
2  import scipy.stats as st
3  import numpy as np
4  import matplotlib.pyplot as plt
5
6  h = 4/10
7
8  def f(x):
9      return (1-(1-h)**(np.floor(x)+1))
10
11  x = np.linspace(0, 15)
12  y = [f(n) for n in x]
13
14
15  plt.step(x, y)
16
17  plt.grid(True)
18  plt.show()
```

### Dimostrare che $f$ è una funzione di densità

Sappiamo che la funzione di densità deve rispettare le seguenti proprietà:

- $f_X(x) \geq 0$
- $\int_{-\infty}^{\infty} f_X(x) dx = 1$

### La descrizione data di $f$ permette di dire che la distribuzione di $X$ è approssimabile in modo accettabile usando il teorema centrale del limite? Giustificare il ragionamento

Sappiamo che la variabile aleatoria  $X$  ha media finita, (abbiamo calcolato sopra il valore atteso), e quindi ha varianza finita. Inoltre so che il supporto è finito, quindi posso applicare il teorema centrale edel limite per approssimare  $X$  in modo accettabile.

### Verificare che $f$ sia una funzione di massa

La funzione di massa è una funzione  $f$  che deve rispettare le seguenti proprietà:

- Non può essere negativa
- La somma dei valori della funzione di massa per tutti gli  $x$  deve fare 1

### Indicate i valori di $a$ e $b$ per i quali $Z$ risulta essere una variabile aleatoria, motivando la vostra risposta

Una variabile aleatoria è una variabile che ha un valore NON costante.

Nel nostro caso,  $Z$  è una variabile aleatoria per ogni  $a, b \in \mathbb{R}$ . Tuttavia, nel caso in cui ' $a = 0$ ' e ' $b = 0$ ', la variabile ' $Z$ ' assume sempre valore ' $Z = 0$ ' con probabilità = 1, quindi diventa una variabile aleatoria degenera

### Formule della varianza

- $VAR(aX) = a^2 * VAR(X)$
- $VAR(X + b) = VAR(X)$

- $VAR(\bar{X}) = \frac{VAR(X)}{n}$
- $VAR(X) = E[X^2] - E[X]^2$

## 6 Esercizio 2

- Stimatore non deviato: valore atteso è uguale al parametro che voglio stimare
- Stimatore deviato: valore atteso NON è uguale al parametro che voglio stimare
- Determinare stimatore non distorto:
  - Se riesco uso plug-in
  - Se nell'applicazione del plug-in ho operatori non lineari, uso metodo di massima verosimiglianza

Se nell'applicazione del plug-in ho operatori non lineari, uso metodo di massima verosimiglianza

### MSE

Scarto quadratico medio (MSE) =  $VAR(Stimatore) + Bias^2(Stimatore)$

Consistenza in media quadratica:

- $\lim_{n \rightarrow \infty} MSE = 0$ : gode della proprietà
- $\lim_{n \rightarrow \infty} MSE \neq 0$ : NON gode della proprietà

### Applicare il teorema centrale del limite

$$P(|T - p| \leq \epsilon) = 2\Phi\left(\frac{\epsilon\sqrt{n}}{\sigma}\right) - 1$$

Se mi chiede la taglia minima del campione: risolvo per  $n$

### Utilizzando il teorema centrale del limite, determinate la distribuzione approssimata dello stimatore $T$ che avete ottenuto al punto 5

Il teorema centrale del limite afferma che, per  $n$  grande:

$$\bar{X} \sim N\left(\mu, \frac{\sigma^2}{n}\right)$$

Sappiamo però che  $T = 2 * \bar{X}$ , quindi, per le proprietà delle trasformazioni lineari delle normali, possiamo dire che:

$$T \sim N\left(2\mu, \frac{4\sigma^2}{n}\right)$$

### Formule Utili

$$E[X^2] = \sum_i (x_i)^2 * p(x_i)$$

## 7 Calcolo combinatorio python

utilità

```

1  from math import factorial as fact
2  from scipy.special import binom
3  # fattoriale
4  fact(5) # 120
5  # coefficiente binomiale
6  binom(5, 2) # 10

```



## combinazioni/disposizioni/permutazioni

```
1 # N = numerosità insieme da cui pescare
2 # k = numero oggetti da estrarre
3 from scipy.special import comb, perm
4 # combinazioni, con o senza ripetizioni (ordine non conta)
5 comb(N, k, repetition=False)
6 # disposizioni (o permutazioni in caso k = N) SENZA RIPETIZIONI (ordine conta)
7 perm(N, k)
```

## Appendice statistica

### Correzione Bessel

Usa la correzione di Bessel quando:

- Hai un **campione** di dati
- Vuoi stimare la **deviazione standard della popolazione** ( $\sigma$ )

### Formule

<b>Popolazione</b> ( $\sigma$ nota)	$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$
<b>Campione</b> (stima di $\sigma$ )	$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$

### Perché $n - 1$ ?

- Corregge il **bias** nella stima
- $n - 1$  = gradi di libertà (perdiamo 1 grado usando  $\bar{x}$ )
- Senza correzione si **sottostima**  $\sigma$

### Implementazione

```
1 import numpy as np
2 s = np.std(dati, ddof=1) # ddof=1 -> divide per (n-1)
```

## Proprietà della Media Campionaria

Data un campione  $X_1, X_2, \dots, X_n$ , dove le variabili sono **i.i.d.** la media campionaria è:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

- **Stima imparziale:**

$$\mathbb{E}[\bar{X}] = \mu$$

La media campionaria è una stima imparziale della media della popolazione.

- **Varianza:**

$$\text{Var}(\bar{X}) = \frac{\sigma^2}{n}$$

La varianza diminuisce all'aumentare della dimensione del campione.

- **Distribuzione asintotica (Teorema del Limite Centrale):**

$$\bar{X} \xrightarrow{d} \mathcal{N}\left(\mu, \frac{\sigma^2}{n}\right) \quad \text{per } n \rightarrow \infty$$

- **Efficienza:** Tra le stime imparziali della media,  $\bar{X}$  ha varianza minima sotto certe condizioni.

## Funzione di Ripartizione (CDF)

Data una variabile casuale  $X$ , la funzione di ripartizione è definita in generale come:

$$F_X(x) = \mathbb{P}(X \leq x)$$

### Caso Discreto

Se  $X$  è discreta e assume valori  $x_1, x_2, \dots$ , con funzione di massa di probabilità  $p_X(x_i) = \mathbb{P}(X = x_i)$ , allora:

$$F_X(x) = \sum_{x_i \leq x} p_X(x_i)$$

- $F_X(x)$  è una funzione a gradini
- È destra-continua
- Cresce solo nei punti  $x_i$  dove  $X$  ha massa di probabilità

### Caso Continuo

Se  $X$  è continua con densità di probabilità  $f_X(x)$ , allora:

$$F_X(x) = \int_{-\infty}^x f_X(t) dt$$

- $F_X(x)$  è continua
- Se derivabile, vale:  $f_X(x) = \frac{d}{dx} F_X(x)$
- La probabilità in un punto è nulla:  $\mathbb{P}(X = x) = 0$

### Proprietà Comuni

- $\lim_{x \rightarrow -\infty} F_X(x) = 0$
- $\lim_{x \rightarrow +\infty} F_X(x) = 1$
- $F_X(x)$  è monotona non decrescente
- È destra-continua

$$E[X^2]$$

### **Caso Discreto**

Se  $X$  è discreta con funzione di massa  $p_X(x)$ :

$$\mathbb{E}[X^2] = \sum_{x \in \text{Im}(X)} x^2 \cdot p_X(x)$$

### **Caso Continuo**

Se  $X$  è continua con densità di probabilità  $f_X(x)$ :

$$\mathbb{E}[X^2] = \int_{-\infty}^{+\infty} x^2 \cdot f_X(x) dx$$