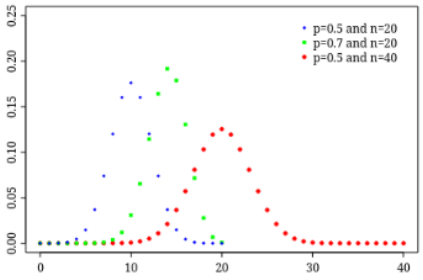
Cheatsheet Statistica

Giacomo Comitani

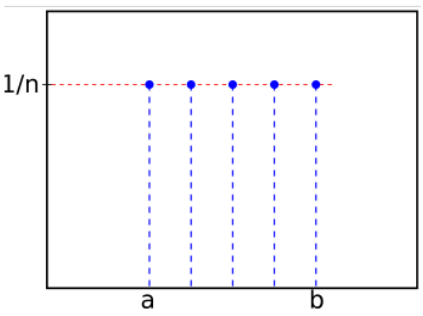
June 2025

1 Modelli Statistici

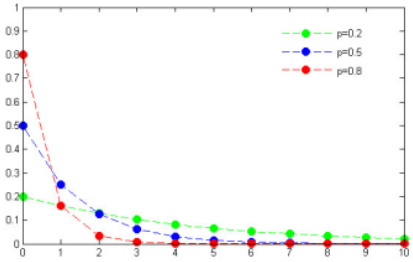
- **Bernoulli**: $X \sim B(p)$: Esperimento avente solamente due possibili esiti: successo o fallimento. **Discreto**
  - Massa: $p^x(1-p)^{1-x}\,\mathbb{I}_{\{0,1\}}(x)$
  - Ripartizione: $(1-p)\,\mathbb{I}_{[0,1)}(x) + \mathbb{I}_{[1,+\infty)}(x)$
  - Valore atteso: p
  - Varianza: $p(1-p)$
- **Binomiale**: $X \sim B(n,p)$: 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



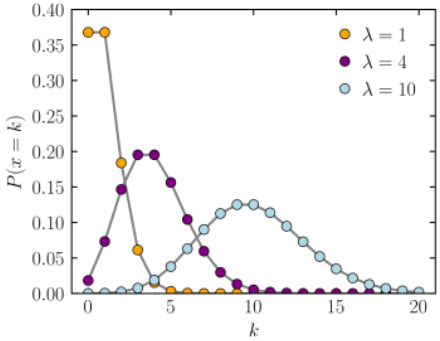
- **Uniforme discreto**: $X \sim U(n)$: Gli esiti della variabile aleatoria sono equiprobabili. **Discreto**
  - Massa: $\frac{1}{n}\,\mathbb{I}_{\{1,\dots,n\}}(x)$
  - Ripartizione: $\frac{\lfloor x \rfloor}{n}\,\mathbb{I}_{[1,n]}(x) + \mathbb{I}_{(n,+\infty)}(x)$
  - Valore atteso: $\frac{n+1}{2}$
  - 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: $\left(1-(1-p)^{\lfloor x \rfloor + 1}\right) \mathbb{I}_{[0,+\infty)}(x)$
  - Valore atteso: $\frac{1-p}{p}$
  - Varianza: $\frac{1-p}{p^2}$
  - Proprietà: Assenza di memoria

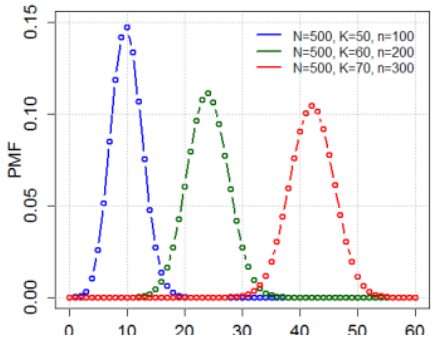


- **Poisson**: $X \sim P(\lambda)$: La variabile assume il numero di eventi che si verificano in un dato intervallo di tempo, sapendo che mediamente se ne verificano un numero $\lambda \in (0, +\infty)$. Tutti gli eventi sono indipendenti. **Discreto**
  - Massa: $\frac{e^{-\lambda}\lambda^x}{x!} \, \mathbb{I}_{\{0,1,2,\dots\}}(x)$
  - Ripartizione: *NON vista nel corso*
  - Valore atteso: λ
  - Varianza: λ
  - Proprietà: Approssimazione binomiale, riproducibilità

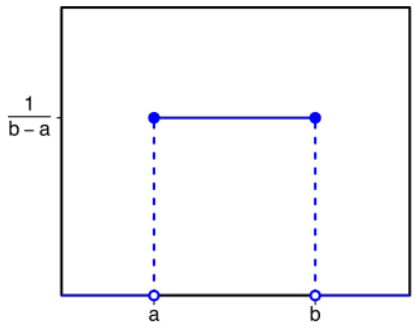


- **Ipergeometrico**: $X \sim H(n, M, N)$: La variabile assume il numero di oggetti corretti estratti da un'urna di oggetti binari durante un'estrazione senza reimmissione dopo n estrazioni. **Discreto**

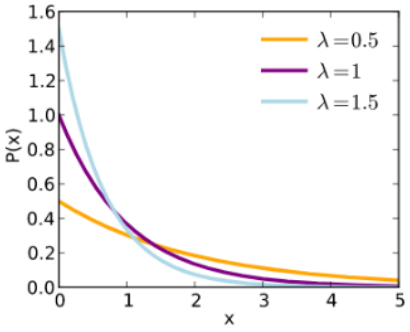
- Massa: $\frac{\binom{N}{x}\binom{M}{n-x}}{\binom{N+M}{n}}\,\mathbb{I}_{\{0,\dots,n\}}(x)$
- Ripartizione: *NON vista nel corso*
- Valore atteso: $\frac{nN}{N+M}$
- Varianza: $\frac{n(N+M-n)NM}{(N+M)^2(N+M-1)}$



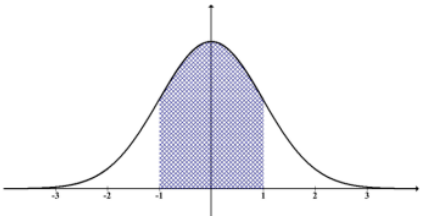
- **Uniforme continuo**: $X \sim U(a,b)$: Gli esiti della variabile aleatoria sono tutti equiprobabili. **Continuo**
  - Densità: $\frac{1}{b-a}\,\mathbb{I}_{[a,b]}(x)$
  - Ripartizione: $\frac{x-a}{b-a}\,\mathbb{I}_{[a,b]}(x) + \mathbb{I}_{(b,+\infty)}(x)$
  - Valore atteso: $\frac{a+b}{2}$
  - Varianza: $\frac{(b-a)^2}{12}$



- **Esponenziale**: $X \sim E(\lambda)$: La variabile assume il tempo di attesa tra due eventi, che mediamente accadono ogni (0, +∞) unità di tempo. **Continuo**
  - Densità: $\lambda e^{-\lambda x}\,\mathbb{I}_{[0,+\infty)}(x)$
  - Ripartizione: $\left(1-e^{-\lambda x}\right)\mathbb{I}_{[0,+\infty)}(x)$
  - Valore atteso: $\frac{1}{\lambda}$
  - Varianza: $\frac{1}{\lambda^2}$
  - Proprietà: Assenza di memoria, scalatura, proprietà su massimo e minimo



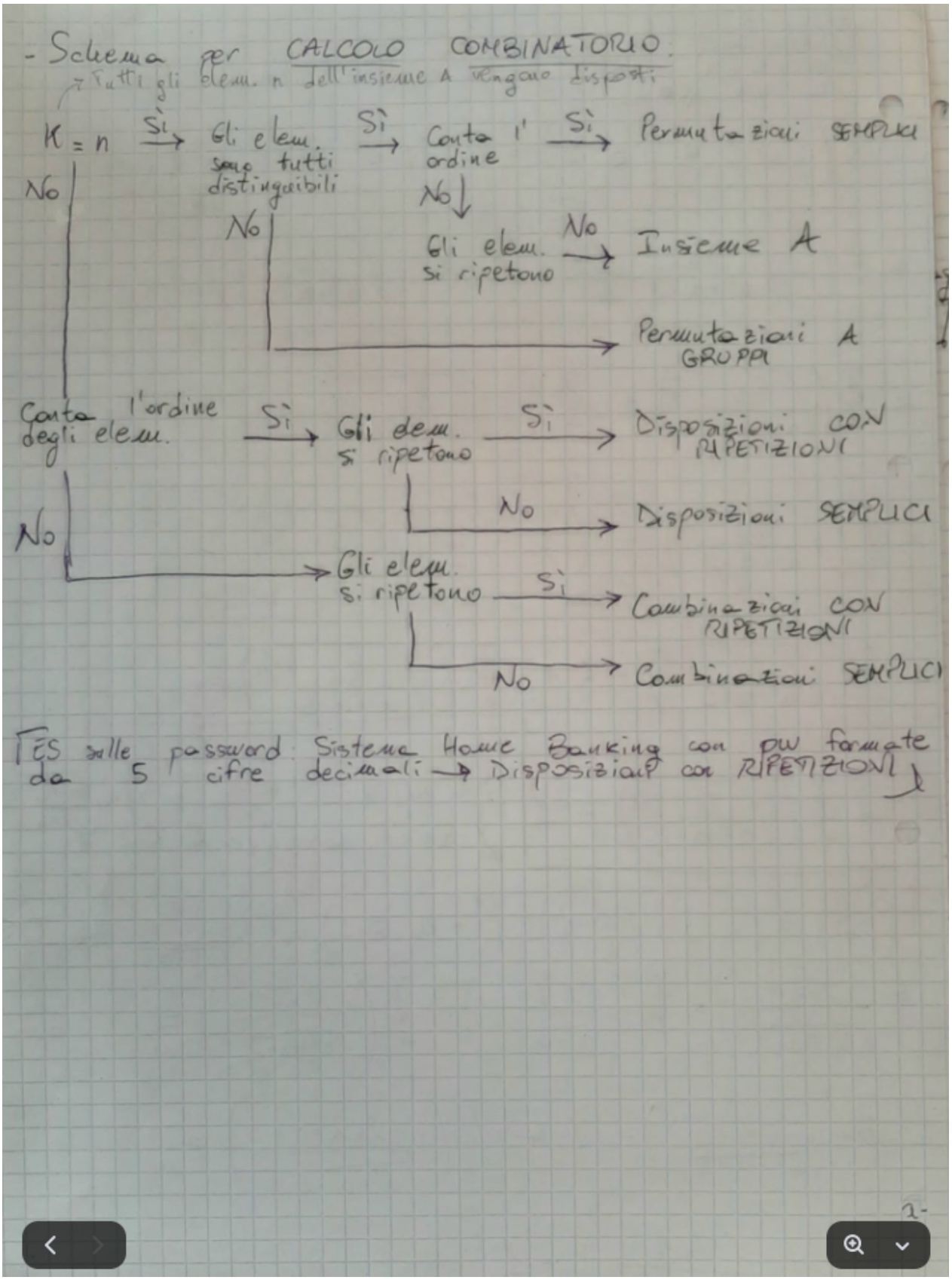
- **Gaussiana (Normale)**: $X \sim G(\mu, \sigma)$. **Continuo**
  - Densità: $\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
  - Ripartizione: $\int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-\mu)^2}{2\sigma^2}}\, dt$
  - Valore atteso: μ
  - Varianza: σ^2
  - Proprietà: Standardizzazione, riproducibilità



Distribuzione	**Stabile sotto $Y = aX + b$?**	**Note**
Bernoulliana	No	Il supporto cambia da $\{0, 1\}$ a $\{b, a + b\}$; non è più una variabile Bernoulli.
Binomiale	No	Non resta binomiale: somma di variabili $aX + b$ non ha struttura binomiale.
Uniforme discreta	Sì	Resta uniforme discreta su un nuovo intervallo: traslazione e scalatura del supporto.
Geometrica	No	La forma esponenziale viene distrutta dalla trasformazione.
Poisson	No	Solo somme di variabili Poisson (con stesso λ) restano Poisson, ma non con $aX + b$.
Ipergeometrica	No	Modifica la struttura combinatoria; non resta ipergeometrica.
Uniforme continua	Sì	Resta uniforme continua su un intervallo trasformato: $U(a, b) \to U(aa + b, ab + b)$.
Esponenziale	Parzialmente	Moltiplicazione positiva ($a > 0$) cambia solo il parametro $\lambda \to \lambda/a$, ma aggiunta (b) rompe la forma.
Normale (Gaussiana)	Sì	Perfettamente stabile: $X \sim \mathcal{N}(\mu, \sigma^2) \Rightarrow aX + b \sim \mathcal{N}(a\mu + b, a^2\sigma^2)$.

ATTENZIONE ALLA RIPRODUCIBILITÀ:

- BINOMIALE: Le variabili devono essere **Indipendenti**, il parametro n può essere diverso per ogni variabile, ma il parametro p deve essere uguale!
- POISSON: Le variabili devono essere **Indipendenti**, $X_1 + X_2 \sim P(\lambda_1 + \lambda_2)$. Se le variabili sono anche **i.i.d.** allora $X_1 + X_2 \sim P(2\lambda)$
- NORMALE: Date le variabili aleatorie $X_1, \ldots, X_n$ gaussiane e **indipendenti**, allora: $Y \sim N(\sum_{i=1}^{n} \mu_i, \sqrt{\sum_{i=1}^{n} \sigma_i^2})$


- Schema per CALCOLO COMBINATORIO.
(tutti gli elem. n dell'insieme A vengono disposti)
K = n
Sì
Gli elem. sono tutti distinguibili
Sì
Conta l'ordine
Sì
Permutazioni SEMPLICI
No
No
No
Gli elem. si ripetono
No
Insieme A
Permutazioni A GRUPPI
Conta l'ordine degli elem.
Sì
Gli elem. si ripetono
Sì
Disposizioni CON RIPETIZIONI
No
Disposizioni SEMPLICI
No
Gli elem. si ripetono
Sì
Combinazioni CON RIPETIZIONI
No
Combinazioni SEMPLICI
ES sulle password Sistema Home Banking con pw formate da 5 cifre decimali → Disposizioni CON RIPETIZIONI

1.0.1 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:

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

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

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

1.0.2 Relazioni utili

$$\binom{n}{0} = \binom{n}{n} = 1 \qquad \binom{n}{1} = \binom{n}{n-1} = n$$

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \quad \text{(relazione di Pascal)}$$

Espansione del binomio di Newton:

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

2 Analisi dei dati con python

Librerie Importanti

```
    import pandas as pd
    import matplotlib.pyplot as plt
    import numpy as np
    import scipy.stats as st
    import statsmodels.api
    import sklearn
    import itertools
```

Importazione e caricamento dei dati

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import scipy.stats as st

ztl = pd.read_csv('ztl.csv', delimiter=';', decimal='.')
ztl
```

Gestione Valori Mancanti

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

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

```

- **Grafico a barre (bar plot)**:
  - preferibile per variabili **ordinali** (es. `livello_rumore`: silenzioso, medio, rumoroso), in cui l'ordine ha significato.

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

```

  - **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.

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

```

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

```
freq = ztl['passaggi'].value_counts()
plt.plot(freq.index, freq, 'o')
plt.vlines(freq.index, 0, freq)
# plt.xlim(0, 30)
plt.show()

```

Gestione Outlier

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

```
    # Visualizzarli graficamente
    ztl['passaggi'].plot.box()
    plt.show()

    # Eliminarli dal dataset
    q3 = ztl['passaggi'].quantile(0.75)
    ztl_filtrato = ztl[ztl['passaggi'] <= q3].reset_index(drop=True)
```

Tabelle di Frequenza

Congiunta relativa

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

Congiunta assoluta

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

Relativa semplice

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

Relativa cumulata

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

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

Cumulata con binning

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

Correlazione

```
plt.scatter(accessi['allarme2'], accessi['carico_sistema'])
plt.show()
```

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

Filtraggio Dati

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

Indici Statistici

Gini per concentrazione

```
    def gini_concentrazione(series):
        freqs = series.value_counts(normalize=True).sort_values().values
        n = len(freqs)
        if n < 2: return 0.0
        Q = np.cumsum(freqs)[:-1]
        F = np.arange(1, n) / n
        return (F - Q).sum() / F.sum()

    gini_concentrazione(auto['tipo_motore'].dropna())
```

Gini per eterogeneità

```
def gini2(series):
    return 1 - sum(series.value_counts(normalize=True).map(lambda f: f ** 2))

gini2(auto['tipo_motore'])
```

Conversione booleana per correlazioni

```
data['convertito'] = data['da_convertire'].apply(lambda x: x == 'ON')
```

Indici Statistici

Indici di centralità

Moda di un carattere

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

Media campionaria di un carattere

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

Mediana di un carattere

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

Indici di dispersione

Varianza campionaria

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

deviazione standard campionaria

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

Analisi Dataset

Elemento massimo

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

Elemento minimo

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

Valori assumibili da un carattere

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

Tipo e forza di correlazione

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

QQ Plot attributo-normale

```
    import statsmodels.api as sm
    mu = data['attributo_1'].mean()
    sigma = data['attributo_1'].std()
    sm.qqplot(data['attributo_1'],dist = st.norm, line = '45', loc = mu, scale = sigma)
    plt.show()
```

QQ Plot tra due attributi

```
    import statsmodels.api as sm

    sm.qqplot_2samples(data['attributo_1'], data['attributo_2'], line = '45')
    plt.show()
```

QQ Plot attributo-distribuzione qualsiasi discreta

```
    import statsmodels .api as sm

    mu = allarme_si_finale.mean()

    sm.qqplot(allarme_si_finale, dist=st.poisson(mu), line='45')
    plt.show()
```

PiePlot

```
    freq = data['c1'].value_counts()
    df = pd.DataFrame({'freq':freq})
    print(df)

    freq.plot.pie()
    plt.show()
```

Grafico a barre

```
    freq_punteggio = data['attributo'].value_counts()
    plt.bar(freq_punteggio.index, freq_punteggio.values)
    plt.show()
```

Individuazione outlier (visivamente)

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

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

Rimozione degli outlier

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

Correlazione tra due attributi

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

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

A seconda del risultato ris ottenuto:

- ris → +1 probabile correlazione linearmente diretta
- ris → 0 correlazione improbabile
- ris → -1 probabile correlazione linearmente indiretta

ScatterPlot (2 parametri)

```
    data.plot.scatter('attrinbuto_1', 'attributo_2')
```

ScatterPlot (3 parametri)

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

Tabella delle frequenze relative cumulate

```
    freq_rel_cumulate = auto['numero_occupanti'].value_counts(normalize = True).
    sort_index().cumsum()

    # Qui il prof preferisce utilizzare i bins, occhio pero che cosi restituisce un
    # intervallo e non un valore, quindi per i calcoli meglio usare senza bins

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

    freq_rel_cumulate_print_prof
```

Tabella delle frequenze relative NON cumulate

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

Convertire valore da stringa a booleano

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

    accessi['carico_sistema'].corr(accessi['allarme2'])
```

Calcolo dimensione del campione togliendo tutti gli elementi mancanti

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

Linespace e Arange

- **linespace**: np.linespace(start, stop, num), dove num è il numero di punti da generare nell'intervallo specificato
- **arange**: np.arange(start, stop, step), dove step è l'incremento tra i valori consecutivi (se omesso è uguale a 1)

Aggiunta nuovo attributo nel dataset

```
    # il nuovo attributo dev'essere chiamato punteggio ed è dato dal doppio delle basse
     emissioni - carpooling:
    lista_nuovo_attributo = (2 * auto['basse_emissioni'] - auto['carpooling'].to_numpy
    ())

    auto['punteggio'] = lista_nuovo_attributo

    # stampa del nuovo attributo aggiunto al dataset:
    auto['punteggio']
```

individuazione e rimozione valori nulli

```
    # Prima di tutto verifico che non ci siano valori nulli:
    print(ztl_giornalieri['categoria'].isnull().sum())

    # Eventualmente per toglierli
    senza_nulli = ztl_giornalieri.dropna(subset=['categoria'])

    print(senza_nulli['categoria'].isnull().sum())
```

3 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 $P(X \geq 30) = P(X = 30) + P(X = 31) + \ldots$.

- 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:

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

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: $VAR(\overline{X}) = \frac{VAR(X)}{n}$

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

4 Esercizio 1

Visualizzare Graficamente una variabile aleatoria/ Visualizzarne le specificazioni

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import scipy.stats as st

# Modifica i parametri
p = 0.7
n = 30
num_sample = 100000

# Modifica la distribuzione
binom = st.binom(n, p)
X = binom.rvs(num_sample)

# Modifica La variabile aleatoria
Z = 2 * X

specificazioni, freq_assolute = np.unique(Z, return_counts = True)
print("Specificazioni: ", specificazioni)
print("Numero di occorrenze per specificazione: ", freq_assolute)

pmf = freq_assolute / num_sample
plt.stem(specificazioni, pmf)

plt.show()
```

Tracciare una funzione generica

```
import pandas as pd
import scipy.stats as st
import numpy as np
import matplotlib.pyplot as plt

h = 4/10

def f(x):
    return (1-(1-h)**(x+1))

x = np.linspace(1, 30, 100)
y = [f(n) for n in x]


plt.plot(x, y)

plt.grid(True)
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 negativaù
- 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 quantità il cui valore non può essere predeterminato con certezza ma è soggetto a variazioni casuali, ovvero che ha un valore NON costante.

Nel nostro caso, Z è una variabile aleatoria per ogni $a, b \in R$. 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 degenere

Formule della varianza

- $VAR(aX) = a^2 * VAR(X)$
- $VAR(X + b) = VAR(X)$
- $VAR(\overline{X}) = \frac{VAR(X)}{n}$
- $VAR(X) = E[X^2] - E[X]^2$

5 Esercizio 2

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

Se nell'appliczione 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:

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

Il metodo di Massima Verosimiglianza

Oltre al metodo plug-in, potrei utilizzare il metodo di massima verosimiglianza nel caso in cui avessi operatori non lineari.

Per capire come utilizzarlo correttamente, partiamo da questi assunti:

- $f_X(x) \rightarrow$ Funzione di Massa/Densità di Probabilità **Marginale**.
- $f_X(x_i) = L(p) = \prod_{i=1}^{n} f_X(x_i, p) \rightarrow$ Funzione di Massa/Densità di Probabilità **Congiunta**.

Esempio con Funzione di Massa di Probabilità Marginale della distribuzione Bernoulliana, a partire dalla quale voglio stimare un parametro fissato p:

La Funzione Marginale è: $f_X(x) = p^x(1-p)^{1-x}$, con $X \in \{0, 1\}$

La Funzione Congiunta è invece: $L(p) = \prod_{i=1}^{n} f_X(x_i, p) = p^{\sum_{i=1}^{n} x_i}(1-p)^{n-\sum_{i=1}^{n} x_i}$

A questo punto bisogna utilizzare le proprietà del **logaritmo naturale** per non avere le sommatorie come esponenti:

$\Rightarrow \ln[L(p)] = ln[p^{\sum_{i=1}^{n} x_i}(1-p)^{n-\sum_{i=1}^{n} x_i}] \Rightarrow$

$\Rightarrow \ln[p^{\sum_{i=1}^{n} x_i}(1-p)^{n-\sum_{i=1}^{n} x_i}] = \sum_{i=1}^{n} x_i \cdot \ln(p) + (n - \sum_{i=1}^{n} x_i)\ln(1-p) =$

Ora devo:

- Derivare rispetto al parametro che intendiamo stimare (in questo caso è p);
- Uguagliare il tutto a 0 ed isolare il parametro da stimare.

$\Rightarrow \sum_{i=1}^{n} x_i \cdot \frac{1}{p} - (n - \sum_{i=1}^{n} x_i)\frac{1}{1-p} = 0 \Rightarrow$

$\Rightarrow \sum_{i=1}^{n} \frac{x_i}{p} = \frac{n-\sum_{i=1}^{n} x_i}{\Rightarrow}$

$\Rightarrow \frac{\sum_{i=1}^{n} x_i(1-p)}{p(1-p)} = \frac{np-(\sum_{i=1}^{n} x_i)p}{p(1-p)}$

$\Rightarrow p = \frac{1}{n}\sum_{i=1}^{n} x_i \rightarrow$ che è la **media campionaria** di X, per definizione stimatore **non distorto** per p.

Applicare il teorema centrale del limite

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

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:

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

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

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

I dati a disposizione permettono di garantire, con probabilità maggiore o uguale a 0.99, che la stima fatta nell'Esercizio 5.2 comporti un errore (in valore assoluto) di massimo $\epsilon = 0.1$

A questo punto devo:

- Calcolare la taglia minima del camoione isolando la n
- Calcolare la taglia del campione effettivo (occhio ad eliminare eventuali valori nulli)
- Confrontare i valori: se il **Secondo valore** è >= del primo, allora la stima è **affidabile**

Formule Utili

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

6 Esercizio 3

Grafico empirico della funzione di ripartizione

```
    # Estrai e ordina i valori della colonna 'richieste'
    richieste = accessi['richieste'].sort_values().values
    n_total = len(richieste)

    # Calcola la CDF empirica (y = frazione cumulativa)
    y = np.arange(1, n_total + 1) / n_total  # Frazione cumulativa (es. 0.2, 0.4, ..., 1.0)

    # Crea il grafico ECDF
    plt.step(richieste, y, label = 'ECDF')
    plt.xlabel('Numero di Richieste')
    plt.ylabel('Frazione Cumulativa (CDF)')
    plt.title('Distribuzione Cumulativa delle Richieste')
    plt.grid(True, linestyle='--', alpha=0.7)

    # Esempio: traccia una linea orizzontale per un f specifico (es. f=95%)
    f = 95
    plt.axhline(y = f/100, color='red', linestyle='--', label=f'{f}% dei casi')
    plt.legend()

    plt.show()
```

Fissato un generico numero f compreso fra 0 e 100, vogliamo determinare il numero n che rende vera l'affermazione nell'f percento dei casi sono arrivate al massimo n richieste"

```
    f = 95
    richieste = accessi['richieste'].sort_index()
    richieste.quantile(f/100)
```

7 Calcolo combinatorio python

utilità

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

combinazioni/disposizioni/permutazioni

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

Formule

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

Perché $n-1$?

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

Implementazione

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

Proprietà della Media Campionaria

Data un campione $X_1, X_2, \ldots, X_n$, 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):**

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

- **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 $x_1, x_2, \ldots$, 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

- $\lim_{x \to -\infty} F_X(x) = 0$
- $\lim_{x \to +\infty} F_X(x) = 1$
- $F_X(x)$ 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 $f_X(x)$:

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