

# Bibbia SAD: Formule e snippet Python

A.R.B risingpirates.com

June 30, 2024

## Contents

<b>1</b>	<b>Elementi Fondamentali</b>	<b>2</b>
<b>2</b>	<b>Gestione Dati da File</b>	<b>3</b>
<b>3</b>	<b>Generazione Grafici</b>	<b>5</b>
<b>4</b>	<b>Statistica Descrittiva</b>	<b>7</b>
<b>5</b>	<b>Teoremi principali</b>	<b>10</b>
<b>6</b>	<b>Modelli</b>	<b>12</b>
<b>7</b>	<b>Formulario Teoria della Probabilità</b>	<b>22</b>
<b>8</b>	<b>Formulario Variabili Aleatorie</b>	<b>23</b>
<b>9</b>	<b>Formulario Statistica Descrittiva</b>	<b>25</b>
<b>10</b>	<b>Formulario Statistica Inferenziale</b>	<b>27</b>
<b>11</b>	<b>Formulario Calcolo Combinatorio</b>	<b>28</b>
<b>12</b>	<b>Formulario Classificatori</b>	<b>29</b>
<b>13</b>	<b>Cenni Matematici</b>	<b>30</b>

---

# 1 Elementi Fondamentali

## 1.1 Import

```
1 import csv
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import scipy.stats as sp ( a volte uso st. e a volte sp. )
5 import statsmodels.api as sm
6 import numpy as np
```

## 1.2 Lista

In Python, una lista è una struttura dati eterogenea e ad accesso posizionale e dinamica: pertanto in essa viene memorizzata tipicamente una sequenza di elementi, che possono essere di tipo diverso e a cui è possibile accedere direttamente specificando la corrispondente posizione.

```
1 Dichiarazione: iron_man = ['e11','e12','e13']
```

Accesso di una lista:

```
1 iron_man[1]
2 iron_man[-2]
```

Slicing delle liste:

```
1 iron_man[4:6]
2 iron_man[-4:-2]
3 names[-5:] #per stampare gli ultimi 5 elementi
```

Verificare se un elemento è nella lista:

```
1 'Thing' in listname
```

Eliminare un elemento nella lista + shift elementi

```
1 del nomelista[0]
```

Inserire un elemento nella lista + shift elementi

```
1 nomelista.insert(4, 'Aquaman')
```

Numero di elementi nella lista:

```
1 len(nomelista)
```

Ordinare una lista:

```
1 names.sort()
```

Ordinare una lista in modo non crescente:

```
1 nomelista.sort(reverse=True)
```

Definire una funzione con argomento x e che restituisce una espressione:

```
1 lambda x: <espressione> (es: successore = lambda n: n+1; successore(9))
```

Ordinare una lista in base alla lunghezza con il parametro key:

```
1 nomelista.sort(key=lambda n: len(n))
```

Trasforma ogni elemento in un valore su cui basare l'ordinamento.

---

## 2 Gestione Dati da File

### 2.1 File CSV

per aprire invece il file CSV in un dataframe:

```
1 df = pd.read_csv('file.csv', sep=',')
2 df = pd.read_csv('file.csv', sep=',', decimal=',', thousands=',')
```

### 2.2 DataFrame

Aprire CSV in dataframe

```
1 df = pd.read_csv('file.csv', sep=',')
```

Settare index in modo che sia ordinato e strutturato per quell'index

```
1 df.set_index('column')
```

Visualizzare le colonne del mio dataframe

```
1 df.columns
```

Informazioni sull'index:

```
1 df.index
```

Accedere alla singola colonna

```
1 df['Colonna']
```

Numero di elementi non ripetuti

```
1 len(df['Colonna'].unique())
```

Lunghezza DataFrame

```
1 len(df)
2 len(df.index)
```

GroupBy

```
1 media = df.groupby('col1')['col_valore'].mean().reset_index()
2 somma = df.groupby('col1')['col_valore'].sum().reset_index()
```

Filtrare i dati di una colonna del dataframe

```
1 #possibile specificare fuori n colonne da 'affiancare'
2 filtered = df[df['column'] > "val"][['col1', 'col2']]
3 #per filtrare via valori numerici
4 notnumbers=df[df['column'].isna()]
5 #per filtrare via valori nulli
6 notnumbers=df[df['column'].notna()]
7 #per filtrare i tipi
8 select_dtypes(include='tipo')
```

vedere quali attributi contengono valori non numerici:

```
1 df.isna().any()
```

Visualizzare solo alcune colonne di un dataframe

```
1 df[['col1', 'col2', 'col3']]
```

---

## 2.3 CrossTabs

Generare un cross tab con frequenze assolute di una colonna di un dataframe:

```
1 ct = pd.crosstab(index=df['col'], columns=['FA'])
```

Generare un cross tab con frequenze relative di una colonna di un dataframe:

```
1 ct = pd.crosstab(index=df['col'], columns=['FR'], normalize=True)
```

Aggiungere una funzione lambda ai dati crosstab:

```
1 ct.apply(lambda p: 100 * np.round(p, 3))
```

Aggiungere un simbolo dopo le frequenze assolute o relative di una crosstab:

```
1 (ct.apply(lambda p: np.round(100*p, 2))  
2 .astype(str)  
3 .apply(lambda s: s + '%'))
```

Generare un crosstab con le frequenze cumulate:

```
1 ct = pd.crosstab(index=df['col'], columns=['FR'], normalize=True)  
2 ct.cumsum()
```

Descrivere il crosstab:

```
1 ct.describe()
```

Generare una tabella di Contingenza

```
1 abs_cont = pd.crosstab(index=df['index'], columns=df['col2'])  
2 rel_cont = pd.crosstab(index=df['index'], columns=df['col2'], normalize=True)
```

Prendere max e min cumulate di una crosstab con frequenze relative

```
1 max_val_y = ctf.cumsum().max().max()  
2 min_val_y = ctf.cumsum().min().min()
```

## 2.4 Series

Filtrare una series:

```
1 serie[serie > soglia]
```

Creare una Series da una lista:

```
1 serie = pd.Series(lista)
```

Aggiungere un valore a tutti gli elementi:

```
1 serie_plus_valore = serie + valore
```

Sostituire i valori nulli con un valore specifico:

```
1 serie_filled = serie.fillna(valore)
```

Rimuovere i valori nulli:

```
1 serie_dropped = serie.dropna()
```

Applicare una funzione a tutti gli elementi:

```
1 serie_squared = serie.apply(lambda x: x**2)
```

Applicare una funzione numpy a tutti gli elementi:

```
1 serie_log = serie.apply(np.log)
```

---

## 3 Generazione Grafici

### 3.1 Generic

Impostare gli step dell'asse y:

```
1 y=np.arange(min_val_y, max_val_y + step, step)
2 plt.yticks(y)
```

Impostare gli step dell'asse x:

```
1 x=np.arange(min_val_x, max_val_x + step, step)
2 plt.xticks(x)
```

Subplots

```
1 # figsize = dimensioni della figura: (larghezza, altezza)
2 fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(6, 6))
3 serie1.plot.box(vert=False, whis=[0, 100], ax=ax1, widths=0.4)
4 ax1.set_title('Primo Subplot')
5
6 serie2.plot.box(vert=False, whis=[0, 100], ax=ax2, widths=0.4)
7 ax2.set_title('Secondo Subplot')
8 # Nascondi l'etichetta dell'asse y per il primo subplot
9 ax1.yaxis.label.set_visible(False)
10 plt.show()
```

### 3.2 Istogrammi

Creare il grafico a barre con gli anni ordinati:

```
1 series_counts = series.value_counts().sort_index()
2 series_counts.plot.bar()
3 plt.show()
```

Creare il grafico a barre di due Serie:

```
1 plt.bar(series1, series2, color='c')
```

Aggiungere delle Linee nel Grafico per asse Y

```
1 plt.axhline(valore_y, color='c', linestyle='dashed', linewidth=1)
```

Aggiungere delle Linee nel Grafico per asse X

```
1 plt.axvline(valore_x, color='c', linestyle='dashed', linewidth=1)
```

Inserire rotazione labels asse X ed Y:

```
1 plt.xticks(rotation=45)
2 plt.yticks(rotation=45)
```

### 3.3 ECDF

per stampare la Empirical Cumulative Distribution Function

```
1 dist = sm.distributions.ECDF(series.dropna())
2 plt.plot(dist.x, dist.y)
3 plt.show()
```

---

### 3.3.1 Generare istogramma da CrossTab

Generare un diagramma generico

```
1 ct.plot() # genera un diagramma generico
```

Generare un diagramma a barre

```
1 ct.plot.bar() # genera un diagramma a barre
```

Generare istogrammi multibarra:

```
1 ct = pd.crosstab(index = df['Index'], columns=[col1, col2])
2 # colonne una di fianco all'altra
3 ct.plot.bar(legend=False, color=['c1', 'c2'])
4 # colonne una sopra l'altra
5 ct.plot.bar(legend=False, color=['c1', 'c2'], stacked=True)
```

### 3.4 Scatter Plot

Se ho due serie di cui proiettare lo scatter plot posso semplicemente fare:

```
1 plt.scatter(serie1, serie2)
2 plt.show()
```

### 3.5 QQ - Plot

Permette di Generare un QQ Plot tra due serie

```
1 sm.qqplot(serie, fit=True, line='45', dist=distribuzione)
2 #nel caso dist non venisse specificato, il qqplot usa una dist normale
3 distribuzione=st.norm # normale
4 distribuzione=st.uniform # uniforme continuo
5 distribuzione=st.exponn # esponenziale
```

### 3.6 BOX-Plot

BoxPlot data una series

```
1 # whis=percentile range of whiskers
2 series.plot.box(whis=[lower,upper], vert=False, patch_artist=True)
```

tramite Matplotlib

```
1 plt.boxplot(data, vert=False, patch_artist=True, whis=[0, 100])
```

### 3.7 Pie-Chart

PieChart data una series

```
1 # Genera un grafico a torta con Pandas
2 series.plot.pie(autopct='%1.1f%%', startangle=90, shadow=True)
```

tramite Matplotlib

```
1 # Genera un grafico a torta con Matplotlib
2 labels = ['A', 'B', 'C', 'D']
3 sizes = [15, 30, 45, 10]
4 explode = (0.1, 0, 0, 0) # "esplode" il primo pezzo (A)
5 plt.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
6         shadow=True, startangle=90)
7 plt.axis('equal') # Assicura che il grafico sia disegnato come un cerchio
8 plt.show()
```

---

## 4 Statistica Descrittiva

### 4.1 Frequenze

#### 4.1.1 Frequenze Assolute

Utilizzando `value_counts`:

```
1 fa = df['colonna'].value_counts()
```

Utilizzando `groupby` e `size`:

```
1 fa = df.groupby('colonna').size()
```

Utilizzando `crosstab`:

```
1 fa = pd.crosstab(index=df['colonna'], columns='count')
```

#### 4.1.2 Frequenze Relative

Utilizzando `value_counts`:

```
1 fr = df['colonna'].value_counts(normalize=True)
```

Utilizzando `groupby` e `size`:

```
1 fr = df.groupby('colonna').size() / len(df)
```

Utilizzando `crosstab`:

```
1 fr = pd.crosstab(index=df['colonna'], columns='count', normalize='index')
```

#### 4.1.3 Frequenze Cumulate

Utilizzando `value_counts`:

```
1 fc = df['colonna'].value_counts().sort_index().cumsum()
```

Utilizzando `groupby` e `size`:

```
1 fc = df.groupby('colonna').size().sort_index().cumsum()
```

Utilizzando `crosstab`:

```
1 fc = pd.crosstab(index=df['colonna'], columns='count').cumsum()
```

---

## 4.2 Indici di Dispersione

**Deviazione Standard:** Utilizzando `std`:

```
1 deviazione_standard = set_di_dati.std()
```

**Varianza:** Utilizzando `var`:

```
1 varianza = set_di_dati.var()
```

**Intervallo (Range):** Calcolando la differenza tra il valore massimo e il valore minimo:

```
1 intervallo = max(set_di_dati) - min(set_di_dati)
```

**Coefficiente di Variazione:** Calcolando il rapporto tra la deviazione standard e la media:

```
1 coefficiente_variazione = set_di_dati.std() / set_di_dati.mean()
```

## 4.3 Indici di Posizione Statistici

**Media:** Utilizzando `mean`:

```
1 media = set_di_dati.mean()
```

**Mediana:** Utilizzando `median`:

```
1 mediana = set_di_dati.median()
```

**Moda:** Utilizzando `mode`:

```
1 moda = set_di_dati.mode()
```

**Percentile:** Utilizzando `percentile` dalla libreria NumPy:

```
1 primo_percentile = np.percentile(set_di_dati, 25)
2 secondo_percentile = np.percentile(set_di_dati, 50) # Mediana
3 terzo_percentile = np.percentile(set_di_dati, 75)
```

**Quantile:** Utilizzando il metodo `.quantile`:

```
1 primo_quantile = Serie.quantile(.25)
2 secondo_quantile = Serie.quantile(.50) # Mediana
3 terzo_quantile = Serie.quantile(.75)
```

**Valore Minimo e Massimo:**

```
1 valore_minimo = min(set_di_dati)
2 valore_massimo = max(set_di_dati)
```

---

## 4.4 Indici di Correlazione

**Correlazione di Pearson:**

```
1 correlazione_pearson = campione1.corr(campione2)
```

**Covarianza Campionaria:**

```
1 covarianza_campionaria = campione1.cov(campione2)
```

## 4.5 Indici di Eterogeneità

**Indice di Gini**

```
1 def gini(series):
2     return 1 - sum(series.value_counts(normalize=True)
3                     .map(lambda f: f**2))
4
5 def normalized_gini(series):
6     s = num_values(series)
7     return s * gini(series) / (s-1)
```

**Entropia:** Utilizzando `entropy` dalla libreria `scipy.stats`:

```
1 entropia = st.entropy(series)
2 leng=len(series.value_counts(normalize=True))
3 norm = entropia*1/math.log(leng,math.e)
```

utilizzando una funzione custom:

```
1 def my_entropy(series):
2     return 0-sum(series.value_counts(normalize=True).map(lambda f: f*math.log(f,math.e)
3     ))
4
5 def normalized_entropy(series):
6     leng=len(series.value_counts(normalize=True))
7     return my_entropy(series)*1/math.log(leng,math.e)
```

---

## 5 Teoremi principali

### 5.1 Teorema delle probabilità totali

**Enunciato:**

Siano  $F = F_1 \cup \dots \cup F_n$   $n$  partizioni disgiunte dell'insieme  $\omega$  e sia  $E \in \omega$  :

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

**Python:**

In questo caso specifico abbiamo  $X > t$  come evento condizionato e  $T = t$  come eventi condizionanti

$$\mathbb{P}(X > t) = \sum_{t=1}^{10} (X > t|T = t) * \mathbb{P}(T = t)$$

```
1 # Definisce una distribuzione uniforme su [0, 10)
2 X = st.uniform(0, 10)
3 # Definisce una distribuzione discreta uniforme tra 1 e 9
4 T = st.randint(1, 10)
5 # Calcolo di P(X > t | T = t) usando il teorema delle probabilita totali
6 t_values = range(1, 10) # Valori possibili per T
7 sum = 0 # Inizializza la somma a zero
8 # Calcola la probabilita totale
9 for ti in t_values:
10     p_ti = T.pmf(ti) # P(T = t_i)
11     p_x_greater_t_given_ti = 1 - X.cdf(ti) # P(X > t_i | T = t_i)
12     sum += p_x_greater_t_given_ti * p_ti # Aggiorna la somma
13 print(sum) # Stampa il risultato
```

---

## 5.2 Taglia minima di un campione

Data una variabile aleatoria  $X$  vogliamo stimare la taglia minima  $n$  di un campione tale che abbia probabilità molto alta di avere il valore di  $\bar{X}$  molto vicino al valore atteso  $\mu$  quindi

$$\mathbb{P}(|\bar{X} - \mu| \leq \epsilon) \geq \alpha$$

Se sviluppiamo questa Probabilità:

$$\begin{aligned}\mathbb{P}(|\bar{X} - \mu| \leq \epsilon) &= \mathbb{P}(-\epsilon \leq \bar{X} - \mu \leq \epsilon) = \\ \mathbb{P}\left(\frac{-\epsilon}{\frac{\sigma}{\sqrt{n}}} \leq \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}} \leq \frac{\epsilon}{\frac{\sigma}{\sqrt{n}}}\right) &= \mathbb{P}\left(\frac{-\epsilon\sqrt{n}}{\sigma} \leq Z \leq \frac{\epsilon\sqrt{n}}{\sigma}\right) = \\ \phi\left(\frac{\epsilon\sqrt{n}}{\sigma}\right) - \phi\left(\frac{-\epsilon\sqrt{n}}{\sigma}\right) &= \phi\left(\frac{\epsilon\sqrt{n}}{\sigma}\right) - 1 + \phi\left(\frac{\epsilon\sqrt{n}}{\sigma}\right) = \\ 2 \cdot \phi\left(\frac{\epsilon\sqrt{n}}{\sigma}\right) - 1 &\geq \alpha \\ \phi\left(\frac{\epsilon\sqrt{n}}{\sigma}\right) &\geq \left(\frac{\alpha + 1}{2}\right)\end{aligned}$$

per comodità ora  $\beta = \left(\frac{\alpha+1}{2}\right)$  e possiamo risolverlo per  $n$  oppure per  $\epsilon$

$$n \geq \frac{\sigma^2}{\epsilon^2} (\Phi^{-1}(\beta))^2$$

$$\epsilon \geq \frac{\sigma}{\sqrt{n}} (\Phi^{-1}(\beta))$$

**Python:** Questo script permette di calcolare la prob

$$\mathbb{P}(|\bar{X} - \mu| \leq \epsilon) = 2 \cdot \phi\left(\frac{\epsilon\sqrt{n}}{\sigma}\right) - 1$$

```
1 #RICORDA: Phi equivale alla CDF di una normale
2 sigma = campione.std()
3 n = len(campione.dropna())
4 epsilon = 0.25
5 Z = st.norm()
6 result = 2 * Z.cdf(epsilon * sqrt(n) / sigma) - 1
```

---

## 6 Modelli

### 6.1 Bernoulli $X \sim B(p)$

La **distribuzione di Bernoulli** calcola le probabilità di un evento a due esiti con probabilità

- **Utilizzo:** Usata per modellare esperimenti con due possibili esiti, ad esempio successo/fallimento, testa/croce, ecc.
- **Supporto:**  $D_X = \{0, 1\}$
- **Parametri:**  $0 \leq p \leq 1$  probabilità del successo
- **Valore Atteso:**  $E(X) = p$
- **Varianza:**  $\text{Var}(X) = p(1 - p)$

#### 6.1.1 Setup

```
1 dist = st.bernoulli(0.3)
```

#### 6.1.2 PMF

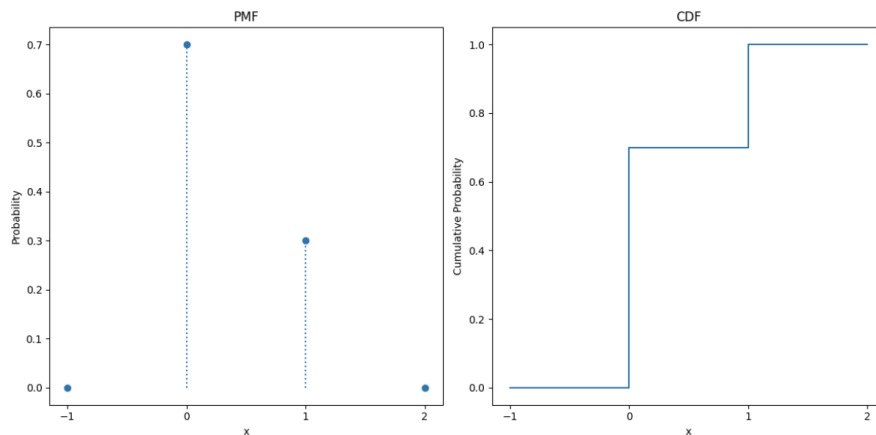
$$P(X = x) = \begin{cases} p & \text{se } x = 1 \\ 1 - p & \text{se } x = 0 \end{cases}$$

```
1 x = np.arange(-1, 3, 1)
2 plt.vlines(x, 0, dist.pmf(x), linestyle='dotted')
3 plt.plot(x, dist.pmf(x), 'o')
4 plt.show()
```

#### 6.1.3 CDF

$$F(x) = \begin{cases} 0 & \text{se } x < 0 \\ 1 - p & \text{se } 0 \leq x < 1 \\ 1 & \text{se } x \geq 1 \end{cases}$$

```
1 plt.step(x, dist.cdf(x), where='post')
2 plt.show()
```



---

## 6.2 Binomiale $X \sim B(n, p)$

La distribuzione binomiale rappresenta una serie di  $n$  variabili aleatorie bernoulliane con stessa probabilità  $p$  (o un evento bernoulliano ripetuto  $n$  volte) di cui vogliamo sapere il numero di successi.

- **Utilizzo:** modellare il numero di successi in una serie di esperimenti bernoulliani e di cui si vuole conoscere il numero di successi.
- **Parametri:**  $n$  numero di tentativi e  $p$  probabilità bernoulliana
- **Supporto:**  $D_X = \{0, 1, 2, \dots, \infty\}$
- **Valore Atteso:**  $E(X) = np$
- **Varianza:**  $Var(X) = np(1 - p)$

### 6.2.1 Setup

```
1 dist = st.binom(10, 0.4)
```

### 6.2.2 PMF

$$p_X(x) = P(X = x) = \binom{n}{x} p^x (1 - p)^{n-x}$$

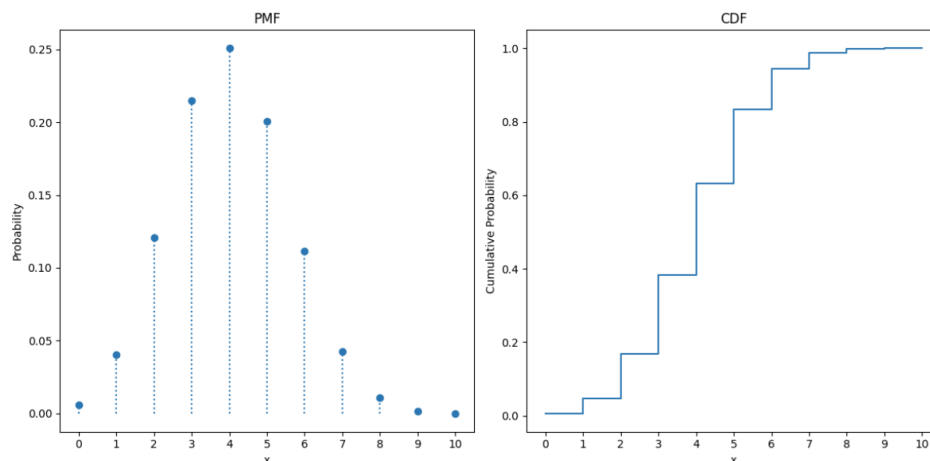
dove  $k$  è il numero di successi.

```
1 x = np.arange(0, 11, 1)
2 plt.vlines(x, 0, dist.pmf(x), linestyle='dotted')
3 plt.plot(x, dist.pmf(x), 'o')
4 plt.show()
```

### 6.2.3 CDF

$$F_X(x) = \sum_{i=0}^k \binom{n}{i} p^i (1 - p)^{n-i}$$

```
1 plt.step(x, dist.cdf(x), where='post')
2 plt.show()
```



## 6.3 Poisson $X \sim P(\lambda)$

La distribuzione di Poisson esprime le probabilità che si verifichino un numero  $n$  di elementi in un determinato lasso di tempo  $t$  (discreto!) sapendo che mediamente se ne verificano  $\lambda$ .

- **Utilizzo:** Usata per modellare il numero di eventi rari che si verificano in un dato intervallo di tempo o spazio, quando il numero di prove è grande e la probabilità di successo per singola prova è molto piccola.
- **Parametri:**  $\lambda > 0$  tasso di arrivo medio degli eventi.
- **Supporto:**  $D_X = \{0\} \cup \mathbb{N}^+$
- **Valore Atteso:**  $E(X) = \lambda$
- **Varianza:**  $Var(X) = \lambda$

### 6.3.1 Setup

```
1 dist = st.poisson(lambda)
```

### 6.3.2 PMF

$$p_X(x) = \frac{e^{-\lambda} \lambda^x}{x!}$$

```
1 x = np.arange(min_val, max_val, step)
2 plt.vlines(x, 0, dist.pmf(x), linestyle='dotted')
3 plt.plot(x, dist.pmf(x), 'o')
4 plt.show()
```

### 6.3.3 CDF

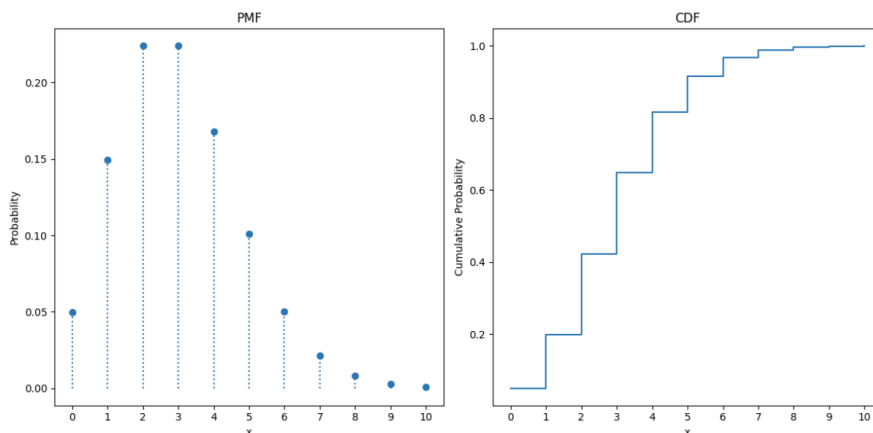
```
1 plt.step(x, dist.cdf(x), where='post')
2 plt.show()
```

### 6.3.4 Relazione col modello Binomiale

quando  $n$  cresce molto e  $p$  decresce poisson approssima una binomiale

### 6.3.4 Riproducibilità

$$X_1 + X_2 \sim P(\lambda_1 + \lambda_2)$$



---

## 6.4 Uniforme Discreta $X \sim U(n)$

La distribuzione uniforme discreta rappresenta un evento con  $n$  possibili esiti i quali hanno tutti la stessa probabilità di successo  $p = \frac{1}{n}$ .

- **Utilizzo:** Usata per modellare un evento con un numero finito di possibili esiti, ognuno con la stessa probabilità di occorrenza.
- **Parametri:**  $n$  numero di possibili esiti dell'evento.
- **Supporto:**  $D_X = [1, n]$
- **Valore Atteso:**  $E(X) = \frac{n+1}{2}$
- **Varianza:**  $Var(X) = \frac{n^2-1}{12}$

### 6.4.1 Setup

```
1 dist=st.randint(1,n)
```

### 6.4.2 PMF

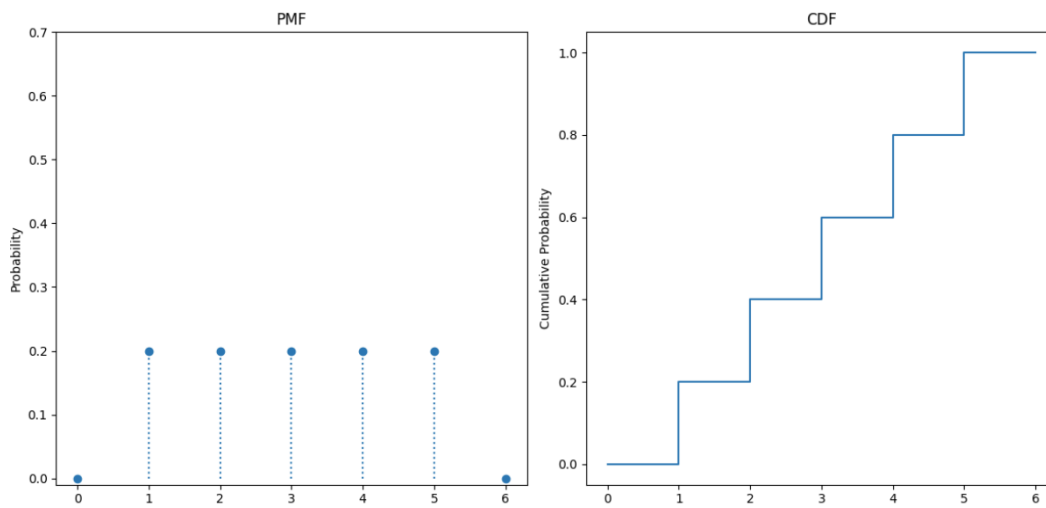
$$p_X(x) = \frac{1}{n}$$

```
1 x=np.arange(min_val,max_val,step)
2 plt.vlines(x,0,dist.pmf(x),linestyles='dotted')
3 plt.plot(x,dist.pmf(x),'o')
4 plt.show()
```

### 6.4.3 CDF

$$F_X(x) = \frac{x}{n}$$

```
1 plt.step(x, dist.cdf(x))
2 plt.show()
```



---

## 6.5 Geometrica $X \sim G(p)$

La distribuzione geometrica serve per calcolare il numero di insuccessi prima di ottenere un successo (senza contarla) ripetendo più volte un esperimento bernoulliano con probabilità  $p$ .

- **Utilizzo:** Usata per modellare il numero di insuccessi prima di ottenere il primo successo in una serie di tentativi indipendenti identicamente distribuiti con probabilità di successo  $p$ .
- **Parametri:**  $0 < p < 1$  è la probabilità di successo in ogni tentativo.
- **Supporto:**  $D_X = \{1, 2, \dots\}$
- **Valore Atteso:**  $E(X) = \frac{1-p}{p}$
- **Varianza:**  $Var(X) = \frac{1-p}{p^2}$

### 6.5.1 Setup

```
1 dist = st.geom(p)
```

### 6.5.2 PMF

$$p_X(x) = p(1-p)^x$$

```
1 x = np.arange(min_val, max_val, step)
2 plt.vlines(x, 0, dist.pmf(x), linestyle='dotted')
3 plt.plot(x, dist.pmf(x), 'o')
4 plt.show()
```

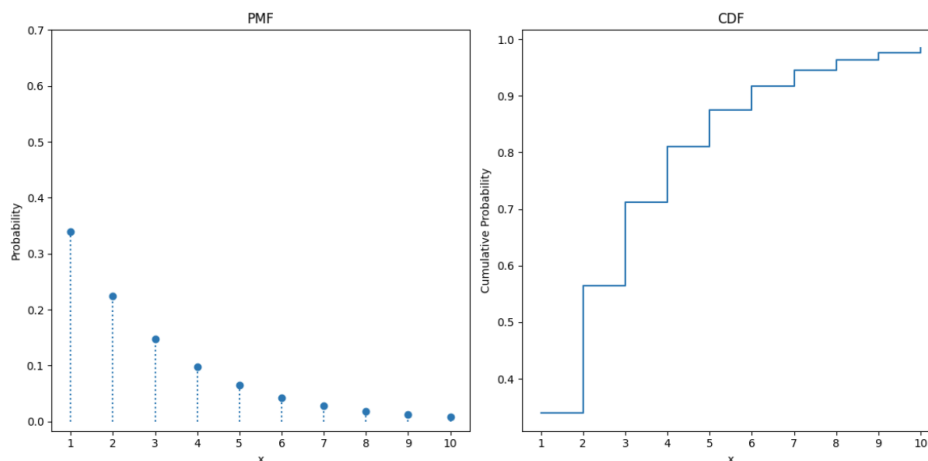
### 6.5.3 CDF

$$F_X(x) = 1 - (1-p)^{x+1}$$

```
1 plt.step(x, dist.cdf(x), where='post')
2 plt.show()
```

### 6.5.4 Assenza di Memoria

$$\mathbb{P}(X > s+t | X > s) = \mathbb{P}(X > t)$$



---

## 6.6 Ipergeometrica $X \sim H(N, n, M)$

La distribuzione ipergeometrica calcola la probabilità di estrarre  $n$  oggetti "giusti" in  $n$  estrazioni da una popolazione composta da  $N$  oggetti "giusti" e  $M$  oggetti "sbagliati".

- **Utilizzo:** Usata per modellare il numero di oggetti "giusti" estratti da una popolazione mista di oggetti "giusti" e "sbagliati" in  $n$  estrazioni senza reinserimento.
- **Parametri:**  $N$  dimensione della popolazione "giusta",  $n$  è il numero di estrazioni, e  $M$  dimensione della popolazione "sbagliata".
- **Supporto:**  $D_X = \{0, 1, \dots, \min(n, N)\}$
- **Valore Atteso:**  $E(X) = np = n \frac{N}{N+M}$
- **Varianza:**  $Var(X) = \frac{NM}{(N+M)^2}$

### 6.6.1 Setup

```
1 dist = st.hypergeom(M, n, N)
```

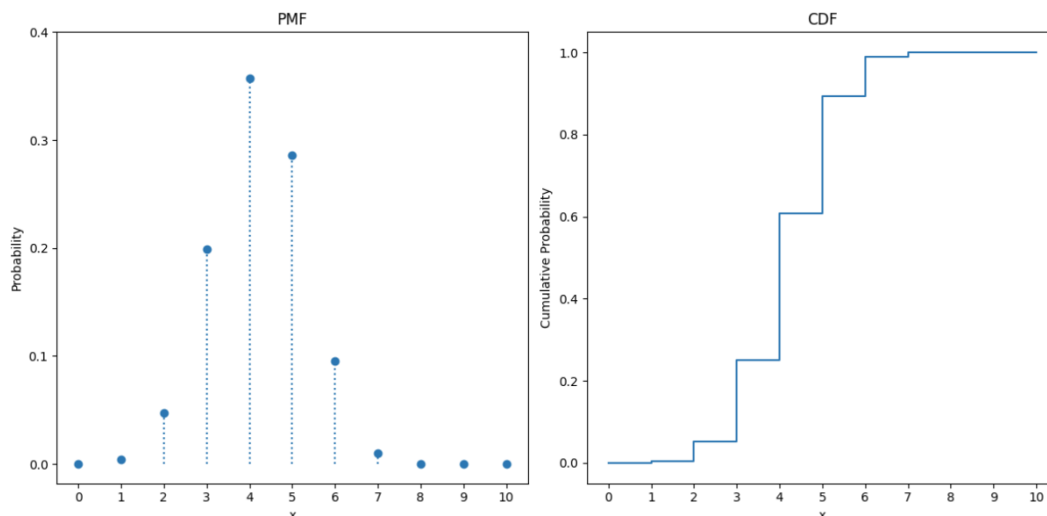
### 6.6.2 PMF

$$p_X(x) = \frac{\binom{N}{x} \binom{M}{n-x}}{\binom{N+M}{n}}$$

```
1 x = np.arange(min_val, max_val, step)
2 plt.vlines(x, 0, dist.pmf(x), linestyle='dotted')
3 plt.plot(x, dist.pmf(x), 'o')
4 plt.show()
```

### 6.6.3 CDF

```
1 plt.step(x, dist.cdf(x))
2 plt.show()
```



---

## 6.7 Uniforme Continua $X \sim U(a, b)$

La distribuzione uniforme continua rappresenta un evento che è equiprobabile per tutti i punti appartenenti all'intervallo  $[a, b]$ .

- **Utilizzo:** Usata per modellare eventi in cui ogni punto all'interno di un intervallo ha la stessa probabilità di occorrenza.
- **Parametri:**  $a$  limite inferiore dell'intervallo e  $b$  limite superiore.
- **Supporto:**  $D_X = [a, b]$
- **Valore Atteso:**  $E(X) = \frac{a+b}{2}$
- **Varianza:**  $Var(X) = \frac{(b-a)^2}{12}$

### 6.7.1 Setup

```
1 dist = st.uniform(a,b)
```

### 6.7.2 PDF

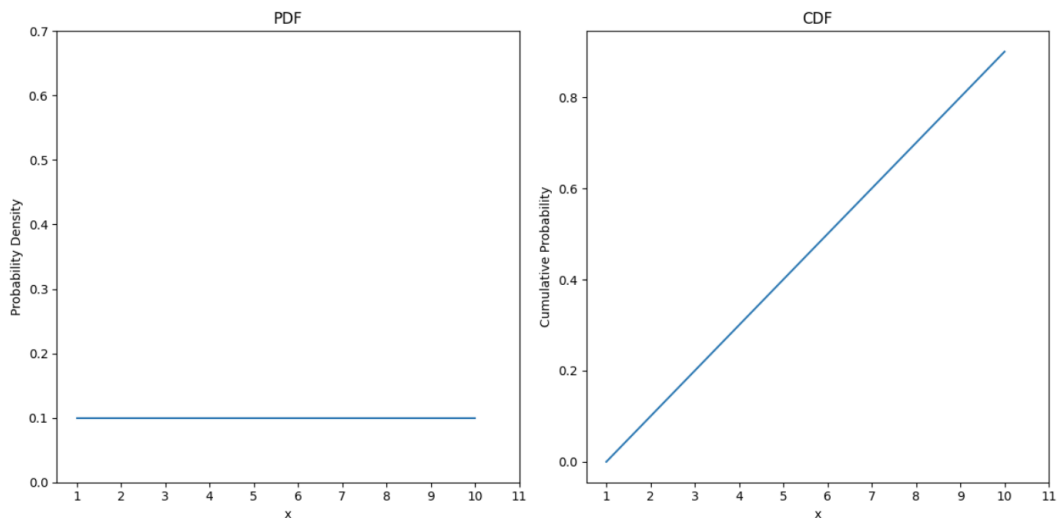
$$f_X(x) = \frac{1}{b-a}$$

```
1 x = np.arange(min_val,max_val,step)
2 plt.step(x, dist.pdf(x))
3 plt.show()
4 # Devo farla cosi "ristretta" per farla uscire bene
```

### 6.7.3 CDF

$$F_X(x) = \frac{x-a}{b-a}$$

```
1 plt.plot(x, dist.cdf(x))
2 plt.show()
```



---

## 6.8 Esponenziale $X \sim E(\lambda)$

La distribuzione esponenziale descrive il tempo di attesa del verificarsi di un certo evento che ha  $\lambda$  probabilità di avverarsi.

- **Utilizzo:** Usata per modellare il tempo di attesa tra eventi che seguono una distribuzione di Poisson con tasso  $\lambda$ .
- **Parametri:**  $\lambda$  tasso di decrescita della distribuzione (numero eventi medi in un range temporale)
- **Supporto:**  $D_X = [0, +\infty)$
- **Valore Atteso:**  $E(X) = \frac{1}{\lambda}$
- **Varianza:**  $Var(X) = \frac{1}{\lambda^2}$

### 6.8.1 Setup

```
1 # scale = E(X) e loc = shift, di base non necessario
2 dist = st.expon(scale=(1/lambda), loc=0)
```

### 6.8.2 PDF

$$f_X(x) = \lambda e^{-\lambda x}$$

```
1 x = np.arange(min_val, max_val, step)
2 plt.plot(x, dist.pdf(x))
3 plt.show()
```

### 6.8.3 CDF

$$F_X(x) = 1 - e^{-\lambda x}$$

dove  $x \geq 0$ .

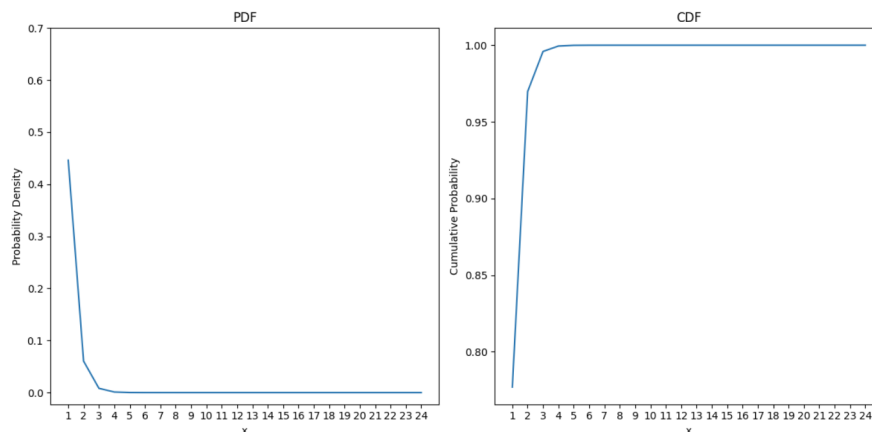
```
1 plt.plot(x, dist.cdf(x))
2 plt.show()
```

### 6.8.4 Scalatura

$$Y = \alpha \cdot X \implies Y \sim E(\lambda') \text{ con } \alpha \geq 1$$

### 6.8.4 Assenza di Memoria

$$\mathbb{P}(X > s + t | X > s) = \mathbb{P}(X > t)$$



## 6.9 Normale $X \sim N(\mu, \sigma^2)$

La distribuzione normale approssima variabili aleatorie che tendono a concentrarsi attorno ad un valore medio  $\mu$  con varianza  $\sigma^2$ .

- **Utilizzo:** modellare fenomeni che si distribuiscono in modo approssimativamente normale.
- **Parametri:**  $\mu$  è valore atteso della distribuzione e  $\sigma^2$  la varianza.
- **Supporto:**  $D_X = \mathbb{R}$
- **Valore Atteso:**  $E(X) = \mu$
- **Varianza:**  $Var(X) = \sigma^2$

### 6.9.1 Setup

```
1 dist = st.norm(val_atteso, dev_std)
```

### 6.9.2 PDF

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

```
1 x = np.arange(min_val, max_val, step)
2 plt.plot(x, dist.pdf(x))
3 plt.show()
```

### 6.9.2 CDF

$$F_X(x) = \int_{-\infty}^x f_X(v) dv = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(v-\mu)^2}{2\sigma^2}\right) dv$$

```
1 plt.plot(x, dist.cdf(x))
2 plt.show()
```

### 6.9.3 Trasformazioni

Data  $Y = aX + b$  allora

$$E(Y) = a\mu + b$$

$$Var(Y) = a^2\sigma^2$$

### 6.9.4 Indipendenza

Siano  $X_1$  e  $X_2$  due Normali indipendenti:

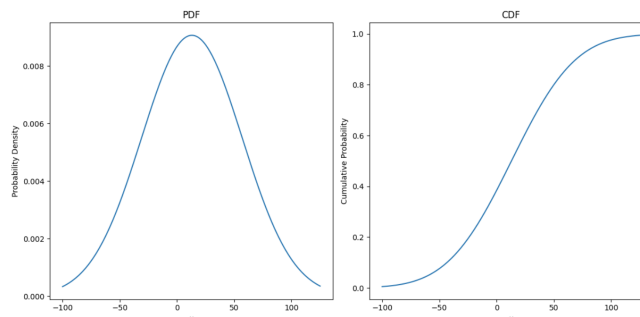
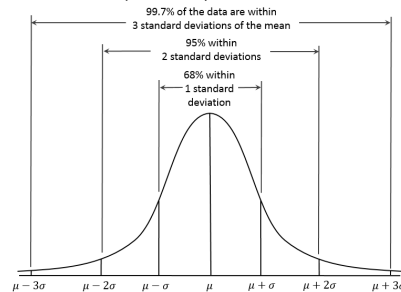
$$X_1 + X_2 \sim (\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$$

### 6.9.3 Regola empirica dei quartili

$$P(\mu - 1\sigma \leq X \leq \mu + 1\sigma) \approx 0.6827$$

$$P(\mu - 2\sigma \leq X \leq \mu + 2\sigma) \approx 0.9545$$

$$P(\mu - 3\sigma \leq X \leq \mu + 3\sigma) \approx 0.9973$$



---

## 6.10 Normale Standard $X \sim N(0, 1)$

Trasformazione di una variabile  $X$  con media  $\mu$  e varianza  $\sigma^2$  in una nuova variabile casuale  $Z$  con media 0 e varianza 1. Utile poichè permette di confrontare variabili normali su una stessa scala. La standardizzazione avviene attraverso la seguente formula:

$$Z = \frac{X - \mu}{\sigma} \sim N(0, 1)$$

## 6.11 Setup

```
1 dist=st.norm()
```

## 6.12 PDF

$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

```
1 dist.pdf(value)
```

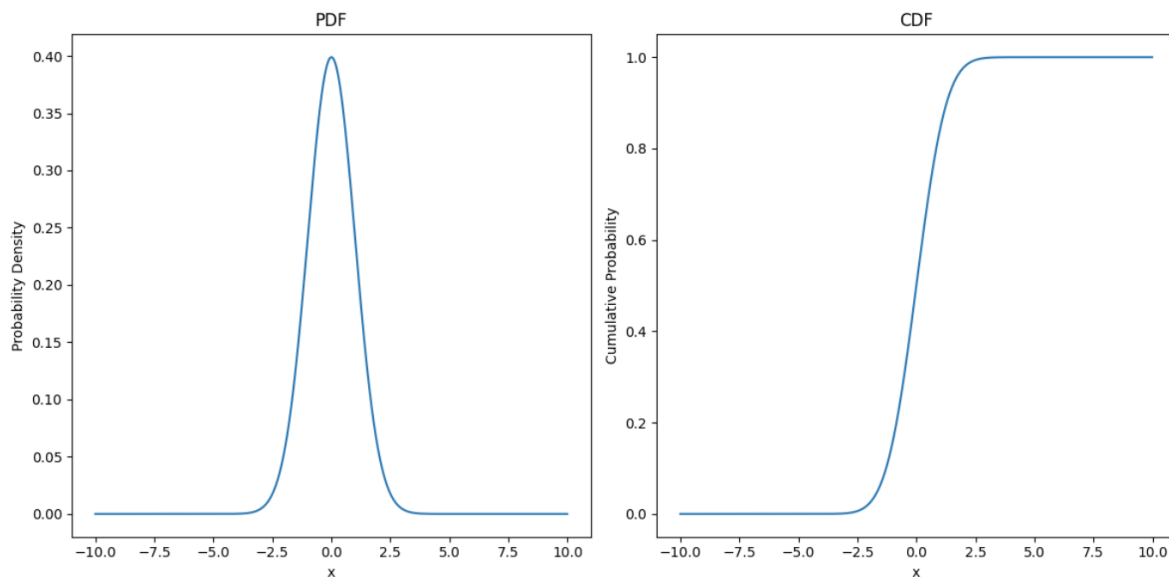
## 6.13 CDF

$$F_X(x) \equiv \phi(x) = \mathbb{P}(X \leq x) = \mathbb{P}(Y \leq \frac{x - \mu}{\sigma})$$

inoltre

$$\phi(-x) = 1 - \phi(x) \implies \phi(x) + \phi(-x) = 1$$

```
1 #Calcolo di PHI
2 dist.cdf(value)
3 #Calcolo dell'inversa di phi
4 dist.ppf(value)
```



---

## 7 Formulario Teoria della Probabilità

### Algebra degli Eventi:

Dati  $E_1, \dots, E_n$  eventi,  $\mathcal{A}$  è algebra degli eventi se:

1.  $\forall E \in \mathcal{A}$ , 2.  $\forall E \in \mathcal{A}, \bar{E} \in \mathcal{A}$ , 3.  $\Omega \in \mathcal{A}$ , 4.  $E, F \in \mathcal{A} \Rightarrow E \cup F \in \mathcal{A}$ .

### Assiomi di Kolmogorov:

definiamo  $\mathbb{P} : \mathcal{A} \Rightarrow \mathbb{R}$  funzione di probabilità se

1.  $\forall E \in \Omega \Rightarrow 0 \leq \mathbb{P}(E) \leq 1$ .
2.  $P(\Omega) = 1$ .
3.  $\forall i, j : i \neq j \quad E_i \cap E_j = \emptyset \quad P(\bigcup_{i=1}^n E_i) = \sum_{i=1}^n P(E_i)$

### Probabilità Complementare:

$$P(\bar{A}) = 1 - P(A).$$

### Probabilità Evento Impossibile:

$$P(\emptyset) = 0.$$

### Probabilità dell'Unione di Due Eventi:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

### Formula Generale delle Probabilità:

$$P(A) = \frac{\text{casi Favorevoli}}{\text{casi Possibili}}$$

### Spazio Equiprobabile:

Dato  $\Omega = 1, 2, \dots, n$  allora  $\forall \omega \in \Omega \quad P(\omega) = \frac{1}{n}$

### Teorema delle Probabilità Totali:

$$P(E) = P(E|F) \cdot P(F) + P(E|\bar{F}) \cdot P(\bar{F})$$

$$\text{Date } B_1, B_2, \dots, B_n \text{ partizioni di } \Omega \text{ ed } A \in \Omega \implies P(A) = \sum_{i=1}^n P(A|B_i) \cdot P(B_i)$$

### Teorema delle Probabilità Condizionate:

Dato  $A$  evento condizionato e  $B$  evento condizionante  $P(A|B) = \frac{P(A \cap B)}{P(B)}$  se  $P(B) > 0$ .

### Teorema di Bayes:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} = \frac{P(B|A) \cdot P(A)}{P(A|B) \cdot P(B) + P(A|\bar{B}) \cdot P(\bar{B})}$$

### Teorema di Bayes per le partizioni:

$$P(A_j|B) = \frac{P(B|A_j) \cdot P(A_j)}{\sum_i P(B|A_i) \cdot P(A_i)}$$

### Indipendenza tra eventi

$P(E \cap F) = P(E) \cdot P(F) \implies E, F$  ed  $E, \bar{F}$  indipendenti  
 $E, F, G$  indipendenti  $\implies E, F \cup G$  indipendenti

---

## 8 Formulario Variabili Aleatorie

**Definizione:** Dato  $(\Omega, \mathcal{A}, \mathbb{P})$  spazio di probabilità:  $\{X = \alpha\} = \{\omega \in \Omega : X(\omega) = \alpha\}$

**PMF:**  $p_X : \mathbb{R} \rightarrow [0, 1]$   $p_X(x) = P(X = x)$

**CDF:**  $F_X : \mathbb{R} \rightarrow [0, 1]$   $F_X(x) = P(X \leq x)$

### 8.1 Variabili Aleatorie Discrete

#### Valore Atteso

$$E(X) = \mu = \sum_x x \cdot p_X(x) \rightarrow Y = aX + b \rightarrow E(Y) = aE(X) + b$$

$$\text{Inoltre: } \forall c \in \mathbb{R} \quad E[(X - c)^2] \geq E[(X - \mu)^2]$$

#### Varianza

$$Var(X) = \sigma^2 = E[(X - \mu)^2] = E(X^2) - E(X)^2 \rightarrow Y = aX + b \rightarrow Var(Y) = a^2 Var(X)$$

#### Deviazione Standard

$$\sigma_X = \sqrt{Var(X)} \rightarrow Y = aX + b \rightarrow \sigma_Y = |a|\sigma_X$$

### 8.2 Variabili Aleatorie Multivariate

#### PMF Congiunta

$$p_{X,Y} : \mathbb{R}^2 \rightarrow [0, 1]$$

$$p_{X,Y}(x, y) = P(X = x \cap Y = y)$$

#### CDF Congiunta

$$F_{X,Y} : \mathbb{R}^2 \rightarrow [0, 1]$$

$$F_{X,Y}(x, y) = P(X \leq x \cap Y \leq y)$$

#### Valore Atteso Congiunto

$$E(X + Y) = E(X) + E(Y)$$

$$E(XY) = E(X) \cdot E(Y)$$

#### Varianza Congiunta

$$Var(X, X) = Var(X) + Var(X) + 2Cov(X, X)$$

$$Var(X + Y) = Var(X) + Var(Y) + 2Cov(X, Y)$$

$$Var(X - Y) = Var(X) + Var(Y) - 2Cov(X, Y)$$

#### Covarianza

$$Cov(X, Y) = E[(X - \mu_x) - (Y - \mu_y)]$$

$$Cov(X, Y) = E(XY) - E(X)E(Y)$$

$$Cov(X, Y) = 0 \text{ Se } X \text{ ed } Y \text{ sono } \mathbf{indipendenti}$$

$$Cov(X, X) = Var(X)$$

$$Cov(X, Y) = E(XY) - E(X) - E(Y)$$

$$Cov(aX, Y) = a \cdot Cov(X, Y)$$

$$Cov(X + Y, Z) = Cov(X, Z) + Cov(Y, Z)$$

#### Indipendenza

$X, Y$  indipendenti  $\iff \forall A, B \in \mathbb{R}: X \in A, Y \in B$  sono indipendenti

$X, Y$  indipendenti s.s.e:

$$\bullet F_{X,Y}(x, y) = F_X(x) \cdot F_Y(y)$$

$$\bullet p_{X,Y}(x, y) = p_X(x) \cdot p_Y(y)$$

$$\bullet P(X, Y) = P(X) \cdot P(Y)$$

$$\bullet P(X \cap Y) = P(X) \cdot P(Y)$$

#### Coefficiente di Correlazione

$$\rho = \frac{Cov(X, Y)}{\sigma_X \cdot \sigma_Y} = \frac{Cov(X, Y)}{\sqrt{Var(X) \cdot Var(Y)}}$$

---

## 8.3 Variabili Aleatorie Continue

### Densità di Probabilità

La densità di probabilità, indicata come  $f_X(x)$ , è una funzione che descrive la distribuzione di probabilità di una variabile aleatoria continua  $X$ . Essa soddisfa le seguenti proprietà:

- $f_X(x) \geq 0$  per ogni  $x \in \mathbb{R}$ .
- $\int_{-\infty}^{\infty} f_X(x) dx = 1$ .
- La probabilità che  $X$  cada in un intervallo  $[a, b]$  è data dall'integrale della densità di probabilità su tale intervallo:  $P(a \leq X \leq b) = \int_a^b f_X(x) dx$ .

### Funzione di Ripartizione

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

### Valore Atteso

$$E(X) = \int_{-\infty}^{\infty} x \cdot f_X(x) dx$$
$$E(g(x)) = \int_{-\infty}^{\infty} g(x) \cdot f_X(x) dx$$

### Disuguaglianza di Markov

$$\forall a > 0 \quad P(X \geq a) \leq \frac{E(X)}{a}$$
$$\forall a > 0 \quad P(X < a) \geq 1 - \frac{E(X)}{a}$$

### Varianza

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

### Disuguaglianza di Chebyshev

$$\forall r > 0 \quad P(|X - \mu| \geq r) \leq \frac{\sigma^2}{r^2}$$
$$\forall r > 0 \quad P(|X - \mu| < r) \geq 1 - \frac{\sigma^2}{r^2}$$
$$\forall r > 0 \text{ e } k = \frac{r}{\sigma} \quad P(|X - \mu| \geq K\sigma) \leq \frac{1}{k^2}$$

---

## 9 Formulario Statistica Descrittiva

### 9.1 Indici di Centralità:

**Media Campionaria:**

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

**Moda e Mediana:**

**Mediana:** valore centrale dei dati

**Moda:** valore più frequente in un insieme di dati

### 9.2 Indici di Dispersione:

**Varianza Campionaria:**

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2.$$

**Deviazione Campionaria Standard:**

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2}.$$

**Intervallo Interquartile:**

$$IQR = Q_3 - Q_1$$

**Coefficiente di Variazione:**

$$CV = \frac{\sigma}{\bar{X}}.$$

**ANOVA**

$$\bar{x}^g = \frac{1}{n_g} \sum_{i=1}^{n_g} x_i^g \quad \bar{x} = \frac{1}{n} \sum_{g=1}^G \sum_{i=1}^{n_g} x_i^g = \frac{1}{n} \sum_{g=1}^G n_g \bar{x}^g.$$

**(total)**  $\text{var}_T = \frac{SS_T}{n-1}$ , con  $SS_T = \sum_{g=1}^G \sum_{i=1}^{n_g} (x_i^g - \bar{x})^2$ : la varianza totale del campione;

**(within)**  $\text{var}_W = \frac{SS_W}{n-G}$ , con  $SS_W = \sum_{g=1}^G \sum_{i=1}^{n_g} (x_i^g - \bar{x}^g)^2$ : la varianza di ogni elemento del gruppo;

**(between)**  $\text{var}_B = \frac{SS_B}{G-1}$ , con  $SS_B = \sum_{g=1}^G n_g (\bar{x}^g - \bar{x})^2$ : la varianza tra ogni gruppo e l'insieme completo.

$$SS_T = SS_W + SS_B$$

### 9.3 Indici di Correlazione:

**Covarianza:**

$$\text{Cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$$

$$\text{Cov}(X, Y) = \frac{1}{n-1} \left( \sum_i x_i y_i - n \bar{x} \bar{y} \right)$$

> 0 diretta, < 0 indiretta

**Coefficiente di Correlazione di Pearson:**

$$\rho = \frac{1}{n-1} \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{s_x s_y} = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}$$

### 9.4 Indici di Eterogeneità:

**Indice di Gini:**

$$I = 1 - \sum_{i=1}^n (f_i)^2$$

$$I' = \frac{I}{n-1} \text{ con } (0 = \min, 1 = \max)$$

(f frequenze relative.)

**Indice di Entropia :**

$$H = \sum_{i=1}^n f_i \ln\left(\frac{1}{f_i}\right)$$

$$H' = \frac{H}{\ln(n)}$$

## 9.5 Indici di Concentrazione:

### Curva di Lorentz

$F_i = \frac{i}{n} \rightarrow$  *posizione* percentuale dell'osservazione  $i$  nell'insieme;

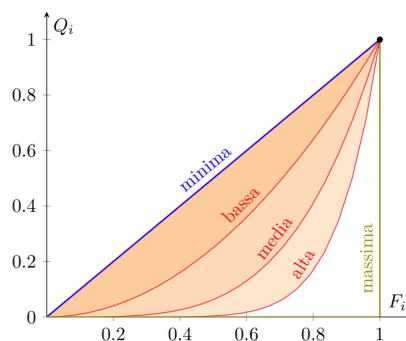
$Q_i = \frac{1}{\text{tot}} \sum_{k=1}^i a_k \rightarrow$  frazione di ricchezza totale posseduta dai primi  $i$  individui.

### Indice di Gini (concentrazione):

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

### Indice di Gini Normalizzato(concentrazione):

$$I' = \frac{2}{n-1} \sum_{i=1}^{n-1} F_i - Q_i$$



## 9.6 Tabella riassuntiva Trasformazioni

Indice	$g(x) = x \pm k$	$g(x) = hx$
Media	$\bar{x} \pm k$	$h\bar{x}$
Mediana	$m_x \pm k$	$hm_x$
Moda	$M_x \pm k$	$hM_x$
Quantile	$q_x \pm k$	$hq_x$
Varianza	$s_x^2$	$h^2 s_x^2$
Dev. std.	$s_x$	$ h s_x$
Range	$r_x$	$hr_x$
IQR	$\text{IQR}_x$	$h\text{IQR}_x$

---

## 10 Formulario Statistica Inferenziale

### Proprietà degli Stimatori

#### Bias (Distorsione) di uno Stimatore

$b_{\tau(\theta)}(T) = E(T) - \tau(\theta)$  Uno stimatore  $T$  è **non distorto** sse  $b_{\tau(\theta)}(T) = 0$ .

#### Errore Quadratico Medio (MSE)

$$MSE_{\tau(\theta)}(T) = E((T - \tau(\theta))^2) = Var(T) + (b_{\tau(\theta)}(T))^2$$

#### Consistenza in Media Quadratica

$$\forall \theta \lim_{n \rightarrow \infty} MSE_{\tau(\theta)}(T_n) = 0$$

#### Consistenza Debole rispetto a $\tau(\Omega)$

$$\forall \epsilon > 0 \lim_{n \rightarrow \infty} P(|T_n - \tau(\theta)| < \epsilon) = 1$$

#### Stimatore: Media Campionaria

La media è stimatore non distorto e consistente in media quadratica per il valore atteso:

$$T_n = t(X_1, X_2, \dots, X_n) = \bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

#### Stimatore: Varianza

La varianza campionaria  $S^2$  è uno stimatore non distorto per la varianza  $\sigma^2$ :

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

### Legge dei Grandi Numeri

#### Legge Forte dei Grandi Numeri

$$P\left(\lim_{n \rightarrow \infty} \bar{X}_n = \mu\right) = 1$$

#### Legge debole dei Grandi Numeri

$$P\left(\lim_{n \rightarrow \infty} |\bar{X}_n - \mu| > \epsilon\right) = 0 \text{ con } \epsilon > 0$$

### 10.1 Disuguaglianza di Chebyshev

$$\forall \epsilon > 0 \ P(|\bar{X} - \mu| \geq \epsilon) \leq \frac{\sigma^2}{n \cdot \epsilon^2}$$

$$\forall \epsilon > 0 \ P(|\bar{X} - \mu| \leq \epsilon) \geq 1 - \frac{\sigma^2}{n \cdot \epsilon^2}$$

## 11 Formulario Calcolo Combinatorio

### Permutazioni Semplici

$n$ : numero totale di elementi

$$p_n = n!$$

### Disposizioni Semplici

$n$ : numero totale di elementi

$k$ : numero di elementi scelti

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

### Combinazioni Semplici

$n$ : numero totale di elementi

$k$ : numero di elementi scelti

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

### Permutazioni con Ripetizione

$n$ : numero totale di elementi

$n_1, n_2, \dots, n_k$ : ripetizioni di ciascun elemento

$$p_{n_1, n_2, \dots, n_k} = \frac{n!}{n_1! n_2! \dots n_k!}$$

### Disposizioni con Ripetizione

$n$ : numero totale di elementi

$k$ : numero di elementi scelti

$$D_{n,k} = n^k$$

### Combinazioni con Ripetizione

$n$ : numero totale di elementi

$k$ : numero di elementi scelti

$$C_{n,k} = \binom{n+k-1}{k}$$

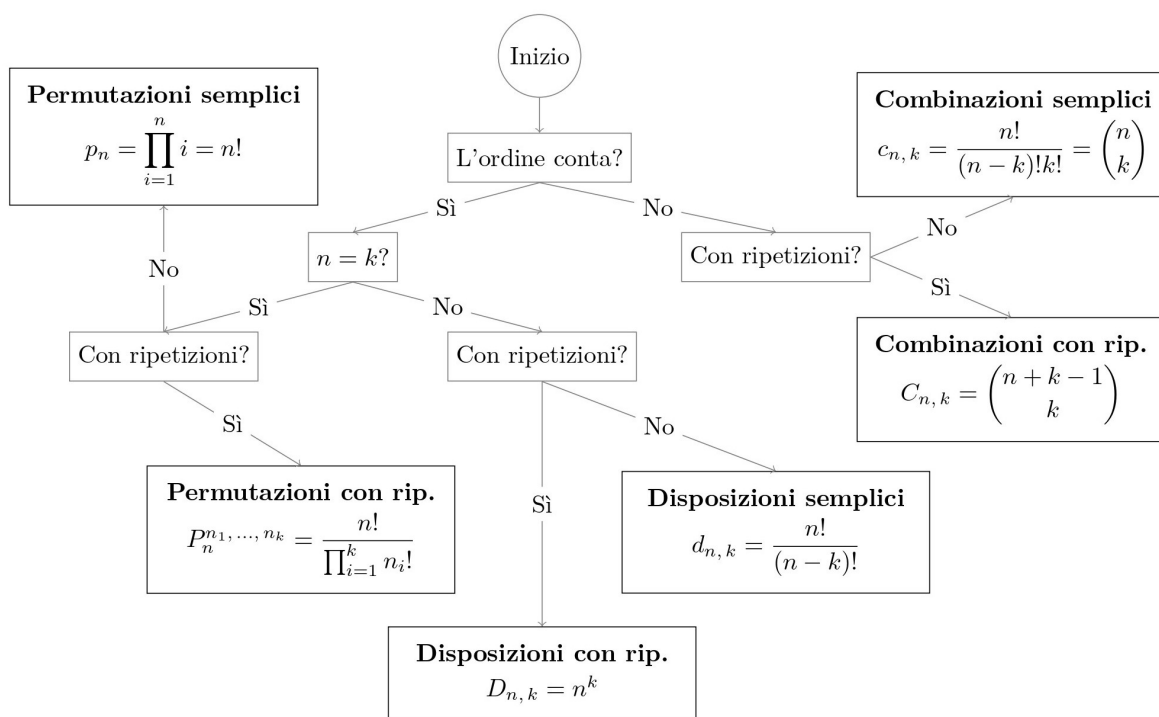


Figure 1: Riassunto delle formule combinatorie

## 12 Formulario Classificatori

		Effettivo	
		Positivi	Negativi
Predizione	Positivo	Vero Positivo (VP)	Falso Positivo (FP)
	Negativo	Falso Negativo (FN)	Vero Negativo (VN)
Totals		TP	TN

- FN = caso positivo viene classificato come negativo
- FP = caso negativo viene classificato come positivo
- $Sens = \frac{VP}{TP}$ ,  $Spec = \frac{VN}{TN}$ ,
- $RFP = 1 - Spec$ ,  $RVP = Sens$ ,  $RVN = Spec$

### Classificatori Costanti positivi

associano indiscriminatamente ogni oggetto alla classe positiva

$Sens = 1$ ,  $Spec = 0$

### Classificatori Costanti negativi

associano indiscriminatamente ogni oggetto alla classe negativa

$Sens = 0$ ,  $Spec = 1$

### Classificatori Ideali

Classificatore che non commette errore

$Sens = 1$ ,  $Spec = 1$

### Classificatori Casuali

assegna un generico oggetto a una classe scelta uniformemente a caso

$Sens = \frac{1}{2}$ ,  $Spec = \frac{1}{2}$

### Classificatori a soglia

classificazione di un generico oggetto calcolando una quantità e verificando che quest'ultima sia superiore a una soglia  $\theta$  prefissata.

Consideriamo intervallo  $D = \{\theta_{min} = \theta_0, \dots, \theta_n = \theta_{max}\}$  allora  $\forall \theta \in D$  calcoliamo sensibilità e specificità e plottiamo su un piano cartesiano ( $Sens$ ,  $RFP$ ), unendo tutti i punti otterremo una curva **ROC**. L'area tra asse X e la curva viene chiamata **AUC** e più si avvicina ad 1 migliore è il mio classificatore.

### Classificatori Naive Bayes

$$P(Y = y_k | X_1 = x_1, \dots, X_n = x_n) \approx \frac{P(Y = y_k) \prod_{i=1}^n P(X_i = x_i | Y = y_k)}{P(X_1 = x_1, \dots, X_n = x_n)} \quad (1)$$

Per trovare la classe  $k^*$  che massimizza la probabilità di indovinare la classificazione:

$$k^* = P(E = e_k | X_1 = x_1 \wedge \dots \wedge X_n = x_n) = \arg \max_k \prod_i P(X_i = x_i | E = e_k) \cdot P(E = e_k)$$

---

## 13 Cenni Matematici

### 13.1 Integrali

#### Tabella integrali fondamentali

Formula	Forma Generalizzata
$\int f'(x) dx = f(x) + c$	$\int f'(g(x))g'(x) dx = f(g(x)) + c$
$\int a dx = ax + c$	$\int a f(x) dx = a \int f(x) dx + c$
$\int x^n dx = \frac{x^{n+1}}{n+1} + c, \quad n \neq -1$	$\int [f(x)]^n f'(x) dx = \frac{[f(x)]^{n+1}}{n+1} + c, \quad n \neq -1$
$\int \frac{1}{x} dx = \log  x  + c$	$\int \frac{f'(x)}{f(x)} dx = \log  f(x)  + c$
$\int \sin x dx = -\cos x + c$	$\int \sin(f(x))f'(x) dx = -\cos(f(x)) + c$
$\int \cos x dx = \sin x + c$	$\int \cos(f(x))f'(x) dx = \sin(f(x)) + c$
$\int \frac{1}{\cos^2 x} dx = \tan x + c$	$\int \frac{f'(x)}{\cos^2(f(x))} dx = \tan(f(x)) + c$
$\int \frac{1}{\sin^2 x} dx = -\cot x + c$	$\int \frac{f'(x)}{\sin^2(f(x))} dx = -\cot(f(x)) + c$
$\int \sinh x dx = \cosh x + c$	$\int \sinh(f(x))f'(x) dx = \cosh(f(x)) + c$
$\int \cosh x dx = \sinh x + c$	$\int \cosh(f(x))f'(x) dx = \sinh(f(x)) + c$
$\int e^x dx = e^x + c$	$\int e^{f(x)} f'(x) dx = e^{f(x)} + c$
$\int e^{kx} dx = \frac{e^{kx}}{k} + c$	$\int e^{kf(x)} f'(x) dx = \frac{e^{kf(x)}}{k} + c$
$\int a^x dx = \frac{a^x}{\log_e a} + c$	$\int a^{f(x)} f'(x) dx = \frac{a^{f(x)}}{\log_e a} + c$

#### Integrazione per Parti

$$\int f(x)g'(x) dx = f(x)g(x) - \int f'(x)g(x) dx + c$$

## 13.2 Derivate

### Tabella derivate

Funzione	Derivata
$y = f(x)^n$	$y' = n f(x)^{n-1} \cdot f'(x)$
$y = \frac{1}{f(x)}$	$y' = -\frac{1}{f(x)^2} \cdot f'(x)$
$y = \sqrt{f(x)}$	$y' = \frac{1}{2\sqrt{f(x)}} \cdot f'(x)$
$y = \sin(f(x))$	$y' = \cos(f(x)) \cdot f'(x)$
$y = \cos(f(x))$	$y' = -\sin(f(x)) \cdot f'(x)$
$y = \tan(f(x))$	$y' = \frac{1}{\cos^2(f(x))} \cdot f'(x) = (1 + \tan^2(f(x))) \cdot f'(x)$
$y = \cot(f(x))$	$y' = -\frac{1}{\sin^2(f(x))} \cdot f'(x)$
$y = e^{f(x)}$	$y' = e^{f(x)} \cdot f'(x)$
$y = \ln(f(x))$	$y' = \frac{1}{f(x)} \cdot f'(x)$
$y = \log_a(f(x))$	$y' = \frac{1}{f(x) \ln a} \cdot f'(x)$
$y = \arcsin(f(x))$	$y' = \frac{1}{\sqrt{1-(f(x))^2}} \cdot f'(x)$
$y = \arccos(f(x))$	$y' = -\frac{1}{\sqrt{1-(f(x))^2}} \cdot f'(x)$
$y = \arctan(f(x))$	$y' = \frac{1}{1+(f(x))^2} \cdot f'(x)$
$y = \operatorname{arccot}(f(x))$	$y' = -\frac{1}{1+(f(x))^2} \cdot f'(x)$

### 13.2.1 Regole di derivazione

Formula	Descrizione
$\frac{d}{dx} [k \cdot f(x)] = k \cdot f'(x)$	$k \cdot f(x)$
$\frac{d}{dx} [f(x) \pm g(x) \pm h(x)] = f'(x) \pm g'(x) \pm h'(x)$	$f(x) + g(x) + \dots$
$\frac{d}{dx} [f(x) \cdot g(x)] = f'(x) \cdot g(x) + f(x) \cdot g'(x)$	$f(x) \cdot g(x)$
$\frac{d}{dx} [f(x) \cdot g(x) \cdot h(x)] = f'(x) \cdot g(x) \cdot h(x) + f(x) \cdot g'(x) \cdot h(x) + f(x) \cdot g(x) \cdot h'(x)$	$f(x) \cdot g(x) \cdot h(x)$
$\frac{d}{dx} \left[ \frac{f(x)}{g(x)} \right] = \frac{f'(x) \cdot g(x) - f(x) \cdot g'(x)}{[g(x)]^2}$	$\frac{f(x)}{g(x)}$
$\frac{d}{dx} [f(g(x))] = f'(g(x)) \cdot g'(x)$	$f(g(x))$
$\frac{d}{dx} [f(x)^{g(x)}] = f(x)^{g(x)} \left[ g'(x) \cdot \ln[f(x)] + g(x) \cdot \frac{f'(x)}{f(x)} \right]$	$f(x)^{g(x)}$