Esame di Statistica e analisi dei dati (01/02/2022)

Esercizio 1 ¶

Sia X una variabile casuale che segue una legge binomiale di parametri $n \in \mathbb{N}$ e $p \in [0, 1]$. **1.** Quali valori può assumere X? **2.** Per n fissato, quali valori può assumere $\mathrm{E}(X)$ (il valore atteso di X)? E quali valori può assumere la sua varianza $\mathrm{Var}(X)$? **3.** Per n fissato, tracciate su due sistemi di riferimento cartesiano i grafic di $\mathrm{E}(X)$ e di $\mathrm{Var}(X)$ al variare di p, evidenziando tutte le informazioni che ritenete rilevanti.

Esercizio 2

Sia $\overline{X}_m$ la media campionaria di un campione casuale $X_1, \ldots, X_m$ estratto da una popolazione descritta dalla variabile aleatoria X dell'esercizio precedente (suggerimento: fate attenzione a distinguere bene la taglia del campione, indicata con m, dal parametro n precedentemente introdotto). **1.** Esprimete $\mathrm{E}(\overline{X}_m)$ in funzione di n e p. **2.** Esprimete $\mathrm{Var}(\overline{X}_m)$ in funzione di m, n e p. **3.** Controllate che $\mathrm{Var}(\overline{X}_m) \leq \frac{n}{4m}$. **4.** Sia m un valore abbastanza piccolo da **non** poter applicare l'approssimazione normale. Controllate che, per ogni $\epsilon > 0$, vale la diseguaglianza: $\mathrm{P}(|\overline{X}_m - np| \leq \epsilon) \geq 1 - \frac{n}{4m\epsilon^2}$.

Esercizio 3

Anche in questo esercizio $\overline{X}_m$ è la media campionaria di un campione casuale $X_1, \ldots, X_m$ estratto da una popolazione descritta dalla variabile aleatoria X del primo esercizio. Assumeremo inoltre che n sia fissato e strettamente maggiore di 1. **1.** Dimostrate che $\overline{X}_m$ è uno stimatore **distorto** di p e calcolate il suo bias. **2.** Proponete uno stimatore T_m che sia non distorto per p, motivando la vostra proposta. **3.** Esprimete $1 - p$ in funzione di $\mathrm{E}(X)$. **4.** Determinate uno stimatore S_m del parametro $q = 1 - p$. **5.** Lo stimatore trovato al punto precedente è non distorto? Giustificate la risposta.

Risolto tutto nel foglio.

Esercizio 4

Collegatevi al sito *upload.di.unimi.it*, selezionate l'esame di *Statistica e analisi dei dati* per l'appello odierno e scaricate il file `mtcars.txt`. Questo file contiene, tra le altre, le seguenti informazioni riguardo al design e alle prestazioni di diversi modelli di automobili (fonte: Motor Trend US magazine, 1974):

- *modello*: identificatore univoco;
- *consumo*: consumo di carburante (espresso in km/l);
- *cilindrata*: cilindrata (espressa in cavalli vapore)
- *testfreni*: numero di test dei freni falliti su dieci prove.

In questo file il carattere di tabulazione (`\t`) separa le colonne e i numeri reali sono stati registrati usando il carattere `,` come separatore dei decimali.

1. Qual è la percentuale dei casi nel dataset che contengono almeno un valore mancante? **2.** Tracciate il boxplot del carattere *consumo* e determinate qual è o quali sono i modelli di auto che possono essere

considerati degli *outlier* rispetto a questo attributo. Basandovi poi **esclusivamente** sull'analisi qualitativa del grafico prodotto, fornite delle approssimazioni del primo, secondo e terzo quartile dell'attributo *consumo*. **3.** Verificate che la percentuale di *outlier* nell'attributo *consumo* equivale alla percentuale di casi del dataset che hanno almeno un valore mancante. Questo fatto è da ascriversi a pura casualità oppure esiste un nesso tra i due concetti? **4.** Tracciate un grafico, diverso dal boxplot, che secondo voi ben rappresenta la distribuzione dell'attributo *consumo*. Giustificate la vostra scelta. **5.** I dati evidenziano una relazione tra il consumo e la cilindrata di un'auto? In caso affermativo, di che tipo è tale relazione? Motivate la vostra risposta anche sulla base di un'opportuna visualizzazione grafica e sul calcolo di un indice numerico. **6.** I dati a disposizione permettono di valutare l'ipotesi che i valori dell'attributo *consumo* siano compatibili con un modello normale? Motivate la vostra risposta.

In [4]:

```
#importiamo il dataset
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import csv

cars = pd.read_csv('mtcars.txt', delimiter = '\t',decimal = ',')
cars
```

Out[4]:

	modello	consumo	cilindri	cilindrata	peso	test400metri	motore	trasmissione	marce
0	Mazda RX4	0.55	6	110	1.19	16.46	0	1	4
1	Mazda RX4 Wag	0.55	6	110	1.31	17.02	0	1	4
2	Datsun 710	0.52	4	93	1.05	18.61	1	1	4
3	Hornet 4 Drive	0.55	6	110	1.46	19.44	1	0	3
4	Hornet Sportabout	0.60	8	175	1.56	17.02	0	0	3
5	Valiant	0.62	6	105	1.57	20.22	1	0	3
6	Duster 360	0.70	8	245	1.62	15.84	0	0	3
7	Merc 240D	0.48	4	62	1.45	20.00	1	0	4
8	Merc 230	0.52	4	95	1.43	NaN	1	0	4
9	Merc 280	0.59	6	123	1.56	18.30	1	0	4
10	Merc 280C	0.62	6	123	1.56	18.90	1	0	4
11	Merc 450SE	0.65	8	180	1.85	17.40	0	0	3
12	Merc 450SL	0.63	8	180	1.70	17.60	0	0	3
13	Merc 450SLC	0.68	8	180	1.72	18.00	0	0	3
14	Cadillac Fleetwood	0.78	8	205	2.39	17.98	0	0	3
15	Lincoln Continental	0.78	8	215	2.47	17.82	0	0	3
16	Chrysler Imperial	0.69	8	230	2.43	17.42	0	0	3
17	Fiat 128	0.31	4	66	1.00	19.47	1	1	4
18	Honda Civic	0.35	4	52	0.73	18.52	1	1	4
19	Toyota Corolla	0.28	4	65	0.83	19.90	1	1	4
20	Toyota Corona	0.54	4	97	1.12	20.01	1	0	3
21	Dodge Challenger	0.67	8	150	1.60	16.87	0	0	3

	modello	consumo	cilindri	cilindrata	peso	test400metri	motore	trasmissione	marce
22	AMC Javelin	0.68	8	150	1.56	17.30	0	0	3
23	Camaro Z28	0.72	8	245	1.75	15.41	0	0	3
24	Pontiac Firebird	0.59	8	175	1.75	17.05	0	0	3
25	Fiat X1-9	0.42	4	66	0.88	18.90	1	1	4
26	Porsche 914-2	0.45	4	91	0.97	16.70	0	1	5
27	Lotus Europa	0.35	4	113	0.69	16.90	1	1	5
28	Ford Pantera L	0.66	8	264	1.44	14.50	0	1	5
29	Ferrari Dino	0.58	6	175	1.26	15.50	0	1	5
30	Maserati Bora	0.68	8	335	1.62	14.60	0	1	5
31	Volvo 142E	0.55	4	109	1.26	18.60	1	1	4

In [5]:

```
#1. Qual è la percentuale dei casi nel dataset che contengono almeno un valore manca
len(cars.dropna()) / len(cars)
# come vediamo, 31/32 = percentuale che cerchiamo
```

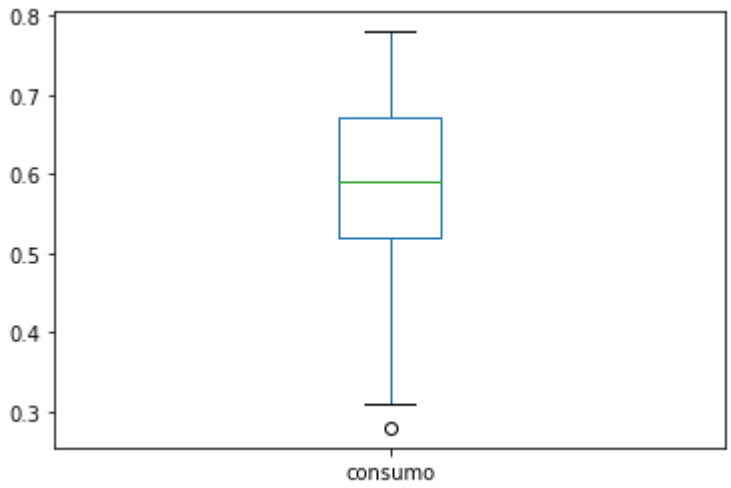
Out[5]:

```
31
```

In [8]:

```
#2. Tracciate il boxplot del carattere consumo e determinate qual è o quali sono i m
# di auto  che possono essere considerati degli outlier rispetto a questo attributo.
# Basandovi poi esclusivamente sull'analisi qualitativa del grafico prodotto,
#fornite delle approssimazioni del primo, secondo e terzo quartile dell'attributo co
cars['consumo'].plot.box()
plt.show()

#ommetendo l'argomento whis otteniamo il grafico di boxplot che evidenzia eventuali
```



Out[8]:

```
0.28
```

In [9]:

```
#dal grafico vediamo che c'e un unico outlier, il minimo
minimoconsumo = cars['consumo'].min()
cars[cars['consumo'] == minimoconsumo].modello
```

Out[9]:

```
19    Toyota Corolla
Name: modello, dtype: object
```

In [ ]:

```
# Basandovi poi esclusivamente sull'analisi qualitativa del grafico prodotto,
#fornite delle approssimazioni del primo, secondo e terzo quartile dell'attributo co

# guardando il box plot, il primo e il terzo quartile corrispondo alle due basi del
# quindi guardano il grafico possiamo dire che il primo quartile = 0,51 e il terzo q
# mentre il secondo quartile e' la mediana  che nel grafico si indica con la linea v
# rettangolo che possiamo approssimare a = 0,59
```

In [ ]:

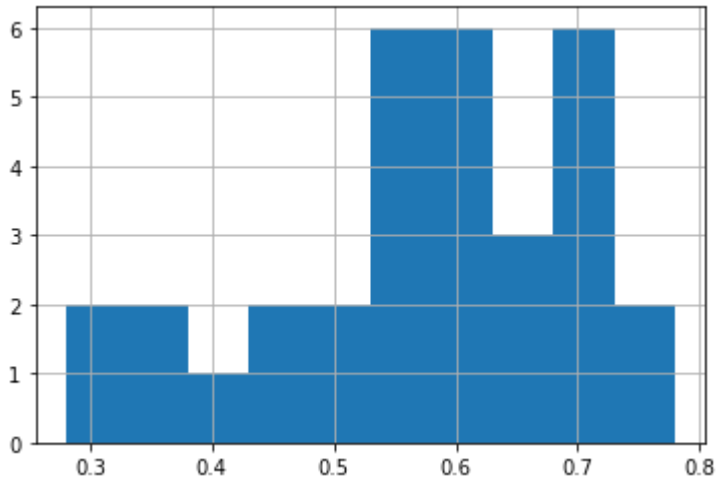
```
#3. Verificate che la percentuale di outlier nell'attributo consumo equivale alla pe
# casi del dataset che hanno almeno un valore mancante.
# Questo fatto è da ascriversi a pura casualità oppure esiste un nesso tra i due con
```

In [10]:

```
#4. Tracciate un grafico, diverso dal boxplot, che secondo voi ben rappresenta
# la distribuzione dell'attributo consumo. Giustificate la vostra scelta

cars['consumo'].hist()
plt.show()

# siccome i valori del attributo consumo sono numeri decimali, consideriamo dei inte
```

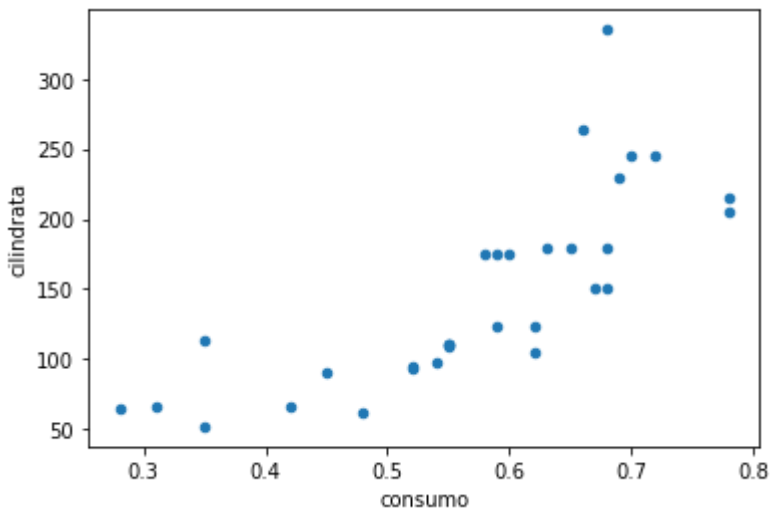


In [ ]:

```
#5. I dati evidenziano una relazione tra il consumo e la cilindrata di un'auto?
#   In caso affermativo, di che tipo è tale relazione? Motivate la vostra risposta
#   anche sulla base di un'opportuna visualizzazione grafica e sul
#   calcolo di un indice numerico.
```

In [14]:

```
cars.plot.scatter('consumo','cilindrata')
plt.show()
```



Il diagramma di dispersion (scatter plot) ci permette di valutare se c'e' una relazione che lega i due caratteri considerati in modo visivo. Guardando il grafico vediamo che all'aumentare del consumo, si aumenta pure la

cilindrata della macchina. Quindi c'e una sorta di relazione lineare diretta. Possiamo calcolarci pure il coefficiente di corellazione lineare per una maggiore sicurezza su questa relazione.

In [15]:

```
cars['consumo'].corr(cars['cilindrata'])
```

Out[15]:

```
0.7725092672461736
```

Siccome il coefficiente di correlazione lineare e' abbastanza vicino ad 1, conferma che tra i due attributi consumo e cilindrata sussista una relazione di lineare di tipo diretto.

In [16]:

```
#6. I dati a disposizione permettono di valutare l'ipotesi che i valori
#   dell'attributo consumo siano compatibili con un modello normale?
#   Motivate la vostra risposta.

cars['consumo'].describe()
```

Out[16]:

```
count    32.000000
mean      0.573125
std       0.128752
min       0.280000
25%       0.520000
50%       0.590000
75%       0.672500
max       0.780000
Name: consumo, dtype: float64
```

La legge normale e' caratterizzato da una forma campana (la sua funzione di densita' e unimodale, cioe' con un massimo locale). Guardando il grafico delle frequenze osservate per l'attributo consumo (istogramma al punto 4), possiamo dire che i dati osservati per quel attributo non sono compatibili con una distribuzione normale.

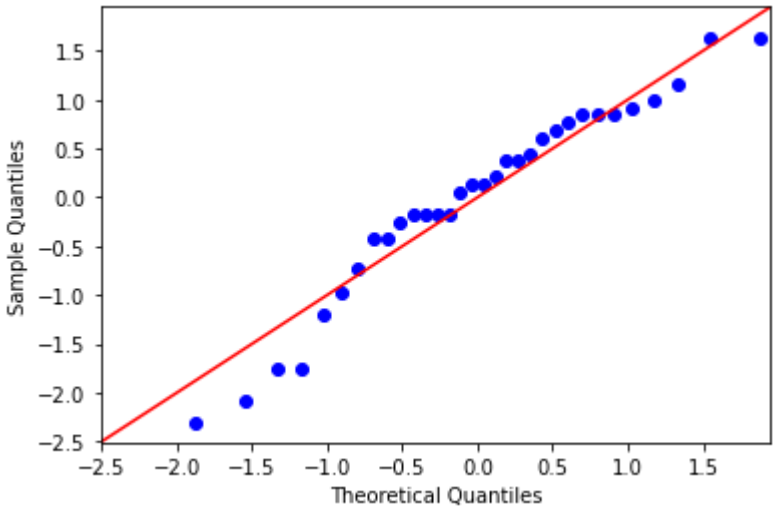
Usando il diagramma Q-Q per l'attributo 'consumo' possiamo approfondire questa analisi, la linea indica la curva attesa in caso di normalita dei dati. Dal grafico otteniamo ulteriore conferma del fatto che la legge normale non sia compatibile con i dati rilevati per l'attributo consumo.

In [17]:

```
import statsmodels.api as sm

sm.qqplot(cars['consumo'], fit = True, line='45')
plt.show()
```

```
/usr/lib64/python3.9/site-packages/statsmodels/graphics/gofplots.py:99
3: UserWarning: marker is redundantly defined by the 'marker' keyword
argument and the fmt string "bo" (-> marker='o'). The keyword argument
will take precedence.
  ax.plot(x, y, fmt, **plot_style)
```



Esercizio 5

Consideriamo ora l'attributo *testfreni*, il cui valore è stato calcolato nel modo seguente: ogni veicolo è stato sottoposto dieci volte a un test per verificare il corretto funzionamento dei freni, e l'attributo contiene il numero di test falliti. Indichiamo con p la probabilità che un veicolo non passi il test sopra indicato, e supponiamo che i dati a disposizione siano ben rappresentativi della popolazione di auto in circolazione (nel periodo di fine anni '70). **1.** Tracciate un grafico opportuno per descrivere l'attributo *testfreni*. Giustificate la scelta fatta. **2.** Considerate i valori osservati per l'attributo *testfreni* come un campione estratto da una popolazione descritta da una variabile aleatoria X. Giustificando la vostra risposta, suggerite un modello per la distribuzione di X. **3.** Stimate il valore atteso di X, indicando la taglia del relativo campione. Quale stimatore avete utilizzato? Motivando la vostra risposta, dite se tale stimatore è non distorto per la quantità che volete stimare. **4.** Stimate la probabilità q che un'auto in circolazione negli anni '70 **passasse** il sopra indicato test di funzionamento dei freni. Quale stimatore avete utilizzato? Motivando la vostra risposta, dite se tale stimatore è non distorto per la quantità che volete stimare. **5.** Fissato $\alpha = 0.85$, determinate l'errore massimo commesso con probabilità maggiore o uguale a α, per eccesso o per difetto, nella stima del valore atteso di X. In altre parole trovate un valore ϵ tale che $\mathrm{P}(|T_m - \mathrm{E}(X)| \leq \epsilon) \geq \alpha$, dove T_m indica lo stimatore che avete utilizzato al punto 3.

1. Tracciate un grafico opportuno per descrivere l'attributo testfreni. Giustificate la scelta fatta.

In [22]:

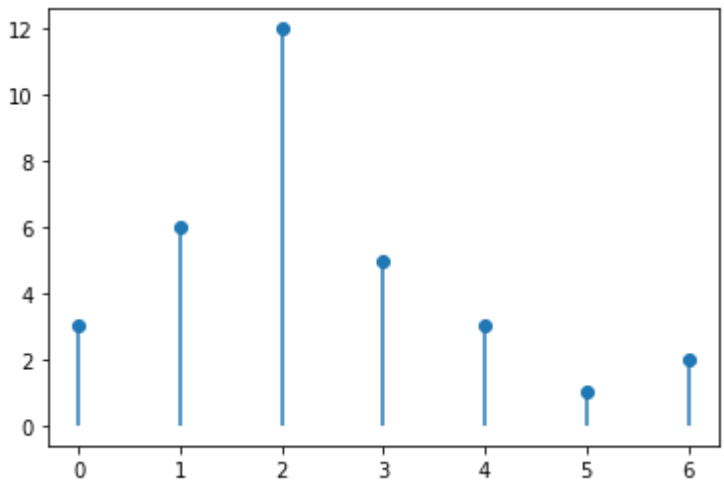
```
# siccome l'attributo e' di tipo numerico (tra 0 e 6), possiamo
# generare un grafico a bastoncini delle frequenze assolute.
cars['testfreni']
testfreni_freq_assol = pd.crosstab(index = cars['testfreni'],
                                   columns=['Frequenza assoluta'],
                                   colnames=[''])
testfreni_freq_assol
```

Out[22]:

	Frequenza assoluta
testfreni	
0	3
1	6
2	12
3	5
4	3
5	1
6	2

In [23]:

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



2. Considerate i valori osservati per l'attributo testfreni come un campione estratto da una popolazione descritta da una variabile aleatoria X. Giustificando la vostra risposta, suggerite un modello per la distribuzione di X.

Un modello sensato per la distribuzione di X sarebbe il modello geometrico. In cui ripetiamo in modo indipendete l'esperimento Bernoulliano di parametro p finche' otteniamo il primo successo. L'esperimento Bernoulliano nel nostro caso e'

successo = un veicolo passa il test (con P(successo) = 1 - p) fallimento = non passa il test (con P(fallimento) = p)

3. Stimate il valore atteso di X, indicando la taglia del relativo campione. Quale stimatore avete utilizzato? Motivando la vostra risposta, dite se tale stimatore è non distorto per la quantità che volete stimare.

In [26]:

```
#Sapendo che la media campionaria e' SEMPRE uno stimatore non distorto per il valore
# della popolazione, calcolando la media campionaria ( .mean() ) per quel attributo
# una stima per il valore atteso di X. Calcolando la media campionaria ottengo una s
# oscilla sempre intorno quel valore centrale (centralita') .
cars['testfreni'].mean()
```

Out[26]:

```
2.3125
```

In [27]:

```
# la taglia del relativo campione e' il numero delle righe del attributo testfreni (
# cioe n = 32
len(cars['testfreni'].dropna())
```

Out[27]:

```
32
```

4. Stimate la probabilità q che un'auto in circolazione negli anni '70 passasse il sopra indicato test di funzionamento dei freni. Quale stimatore avete utilizzato? Motivando la vostra risposta, dite se tale stimatore è non distorto per la quantità che volete stimare.

Usiamo di nuovo la media campionaria per calcolare tale probabilita'. Come detto precedentemente tale stimatore e' sempre uno stimatore non distorto per la quantita che voglio stimare (il valore atteso). Valore atteso della distribuzione geometrica e' 1-p/p

In [28]:

```
cars['testfreni'].mean()
```

Out[28]:

```
2.3125
```

5. Fissato $\alpha=0.85$, determinate l'errore massimo commesso con probabilità maggiore o uguale a α, per eccesso o per difetto, nella stima del valore atteso di X. In altre parole trovate un valore ϵ tale che $P(|Tm-E(X)|\leq\epsilon)\geq\alpha$, dove Tm indica lo stimatore che avete utilizzato al punto 3.

In [39]:

```
import scipy.stats as st
sigma_x = cars['testfreni'].std()
n = len(cars['testfreni'].dropna())
Z = st.norm()
alpha = 0.85
eps = (sigma_x / n**0.5) * Z.ppf(0.925)
eps
```

Out[39]:

```
0.39017066302882414
```

Come visto a lezione, il calcolo di PHI inverso corrisponde al calcolo dei quantili (Z.ppf(0.925))..... Per cui ϵ = 0.39

In [ ]:

```

```