EDA

October 15, 2021

1 1. UNIVARIATE ANALYSIS

1.1 1.1 Measuring the central tendency

1.1.1 1. mean

Arithmetic mean

[1]:
```
# Importing libraries
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:
```
# Load dataset
index = [1,2,3,4,5]
age = [31, 22, 43, 24, 35]
```

[3]:
```
# Graph details
plt.xlabel("Index")
plt.ylabel("Age")
plt.title("Employees Age")
# Plot graph
plt.plot(index, age, color="#880066", marker="D", linewidth=3, markersize=10)
plt.tight_layout()
plt.show()
```

[4]:
```
# Arithmetic mean of age variable
import numpy as np
age = np.array(age)
print("Arithmetic mean : " ,age.mean())
```

```
Arithmetic mean :  31.0
```

Weighted arithmetic mean

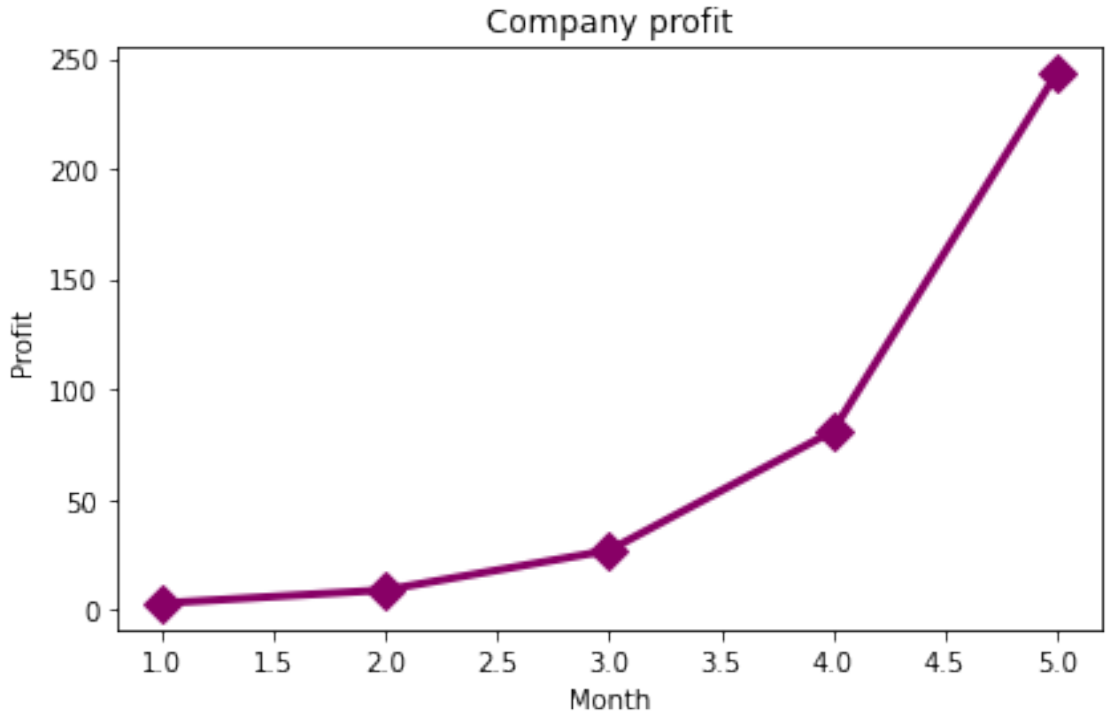
[5]:
```
# Weighted arithmetic mean of age variable
age = [31, 22, 43, 24, 35]
weight = [0.2, 0.9, 0.4, 0.3, 0.6]

numerator = 0
for i in [0,1,2,3,4]:
    numerator += age[i]*weight[i]
denominator = sum(weight)

print("Weighted arithmetic mean : " ,numerator/denominator)
```

```
Weighted arithmetic mean :  29.750000000000004
```

Geometric mean

[6]:
```
# Load dataset
month = [1,2,3,4,5]
profit = [3,9,27,81,243]
```

[7]:
```
# Graph details
plt.xlabel("Month")
plt.ylabel("Profit")
plt.title("Company profit")
# Plot graph
plt.plot(month, profit, color="#880066", marker="D", linewidth=3, markersize=10)
plt.tight_layout()
plt.show()
```



[8]:
```
# Arithmetic mean of age variable
import numpy as np
profit = np.array(profit)
print("Arithmetic mean : " ,profit.mean())
```

```
Arithmetic mean :  72.6
```

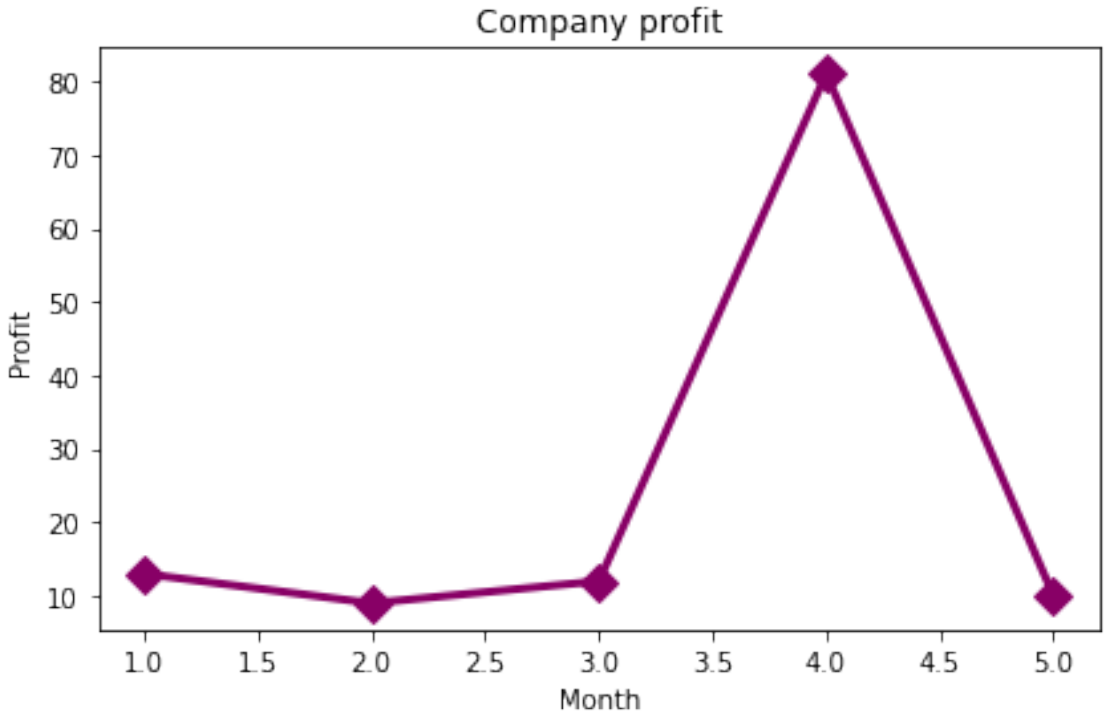
[9]:
```
# Geometric mean
from scipy import stats
print("Geometric mean : " ,stats.gmean(profit))
```

```
Geometric mean :  27.0
```

Harmonic mean

[10]:
```
# Load dataset
month = [1,2,3,4,5]
profit = [13, 9, 12, 81, 10]
```

[11]:
```
# Graph details
plt.xlabel("Month")
plt.ylabel("Profit")
plt.title("Company profit")
# Plot graph
plt.plot(month, profit, color="#880066", marker="D", linewidth=3, markersize=10)
plt.tight_layout()
plt.show()
```



[12]:
```
# Arithmetic mean of age variable
import numpy as np
profit = np.array(profit)
print("Arithmetic mean : " ,profit.mean())
```

```
Arithmetic mean :  25.0
```

[13]:
```
# Geometric mean
from statistics import geometric_mean
print("Geometric mean : " ,geometric_mean(profit))
```

```
Geometric mean :  16.261867768764628
```

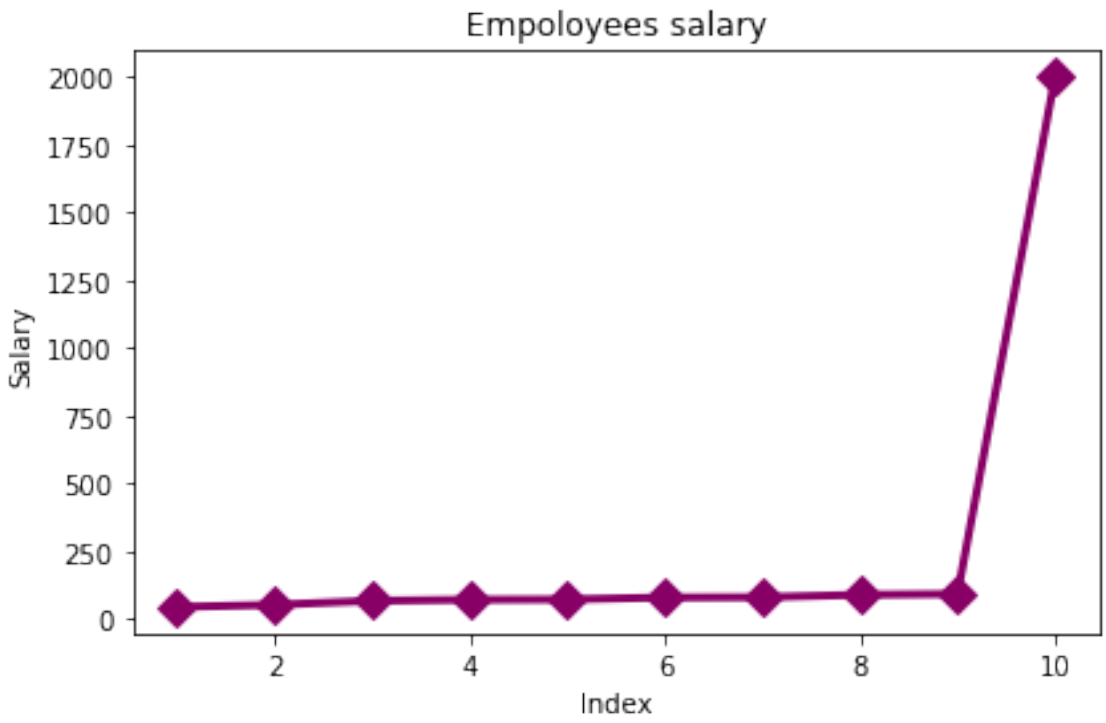
[14]:
```
# Harmonic mean
import statistics
print("Harmonic Mean : ", statistics.harmonic_mean(profit))
```

```
Harmonic Mean :  13.030565524068804
```

1.1.2 2. Median

[15]:
```
# Load dataset
index = [1,2,3,4,5,6,7,8,9,10]
salary = [45, 53, 68, 72, 73, 80, 81, 90, 93, 2000]
```

[16]:
```
# Graph details
plt.xlabel("Index")
plt.ylabel("Salary")
plt.title("Empoloyees salary")
# Plot graph
plt.plot(index, salary, color="#880066", marker="D", linewidth=3, markersize=10)
plt.tight_layout()
plt.show()
```



[17]:
```
# Arithmetic mean of age variable
import numpy as np
salary = np.array(salary)
print("Arithmetic mean : " ,salary.mean())
# Geometric mean
from statistics import geometric_mean
print("Geometric mean : " ,geometric_mean(salary))
# Harmonic mean
import statistics
print("Harmonic Mean : ", statistics.harmonic_mean(salary))
```

```
Arithmetic mean :  265.5
Geometric mean :  99.2197917603175
Harmonic Mean :  76.57376717397129
```

[18]:
```
# Median
print("Median : " ,np.median(salary))
```

```
Median :  76.5
```

1.1.3 3. Median

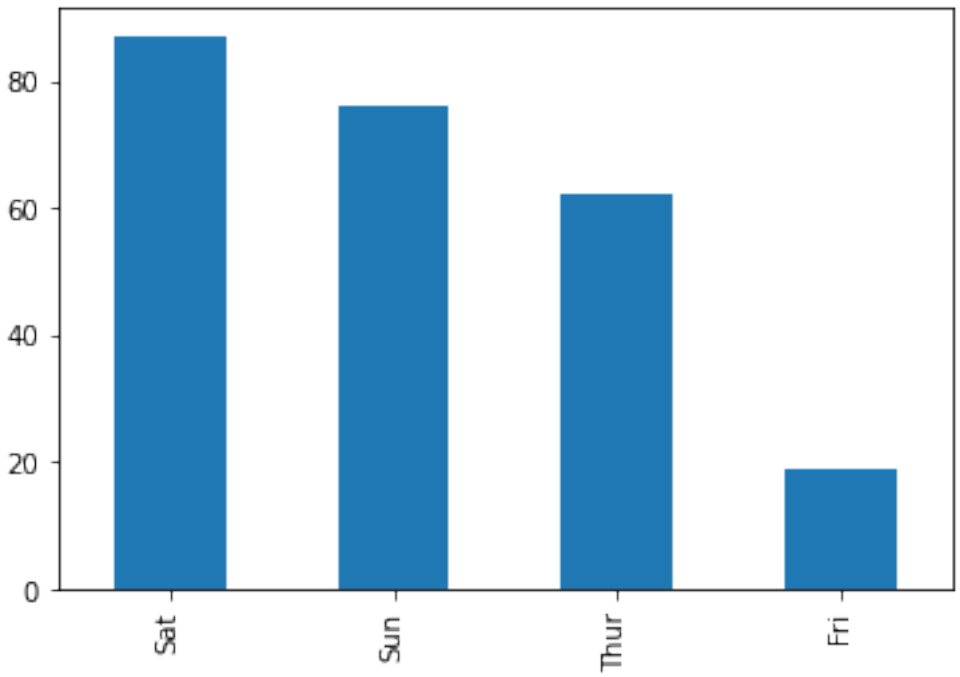
[19]:
```
# Load random set of values
import pandas as pd
data = pd.read_csv("E:\\MY LECTURES\\DATA SCIENCE\\3.Programs\\dataset\\Meal␣
 ↪and Tips.csv")
data.head()
```

[19]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

[20]:
```
data["day"].value_counts().plot(kind="bar")
```

[20]: <AxesSubplot:>



1.2 1.2 Measuring the dispersion

1.2.1 1. Range

[21]:
```
# Load dataset
salary = [45, 53, 68, 72, 73, 80, 81, 90, 93, 2000]
```

[22]:
```
range = [min(salary), max(salary)]
print("The range in salary data is :", range)
```

```
The range in salary data is : [45, 2000]
```

1.2.2 2. Quantile

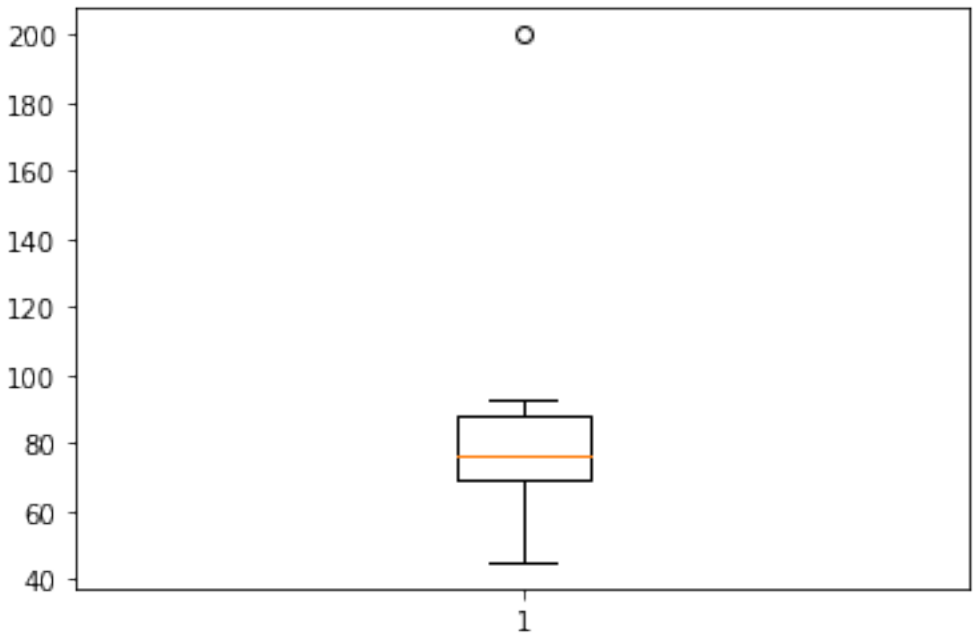
[23]:
```
# Load dataset
salary = [45, 53, 68, 72, 73, 80, 81, 90, 93, 200]
```

[24]:
```
# Quantile
print("Dataset        : ", salary)
print("Q2 quantile    : ", np.quantile(salary, .50))
print("Q1 quantile    : ", np.quantile(salary, .25))
print("Q3 quantile    : ", np.quantile(salary, .75))
print("100th quantile : ", np.quantile(salary, .1))
```

```
Dataset        :  [45, 53, 68, 72, 73, 80, 81, 90, 93, 200]
Q2 quantile    :  76.5
Q1 quantile    :  69.0
Q3 quantile    :  87.75
100th quantile :  52.2
```

Box-whisker plot

[25]:
```
plt.boxplot(salary)
plt.show()
```



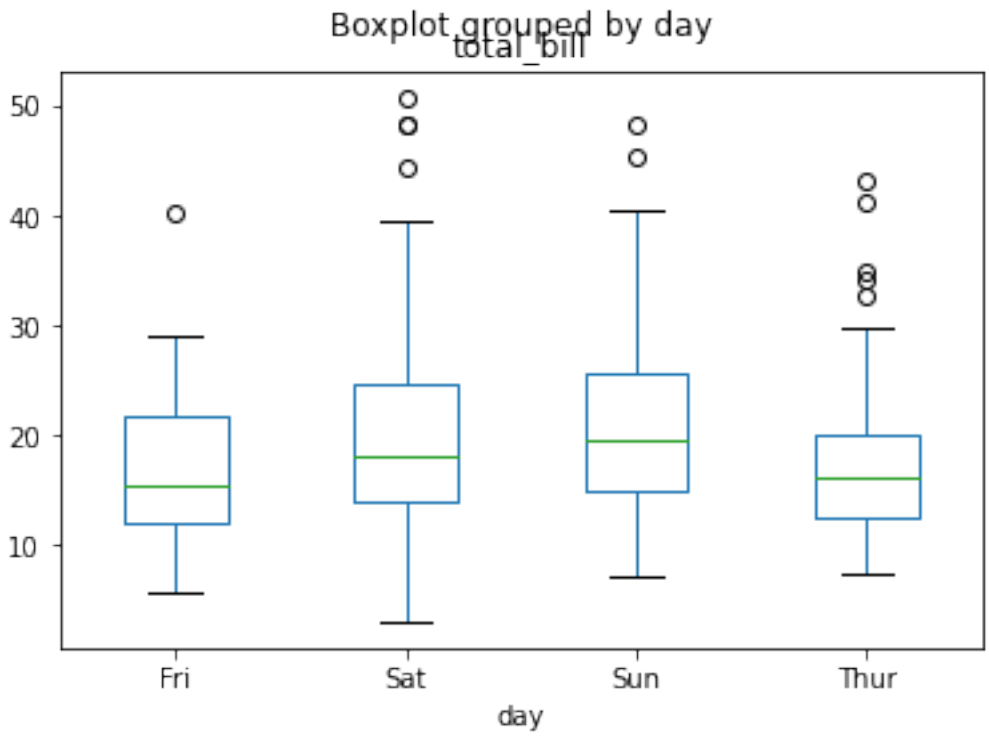
[26]:
```
# Load random set of values
import pandas as pd
data = pd.read_csv("E:\\MY LECTURES\\DATA SCIENCE\\3.Programs\\dataset\\Meal␣
 ↪and Tips.csv")
data.head()
```

[26]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

[27]:
```
data.boxplot(by ='day', column =['total_bill'], grid = False)
```

[27]:
```
<AxesSubplot:title={'center':'total_bill'}, xlabel='day'>
```



1.2.3 3. Variance

[28]:
```
# Load dataset
dog_height = [600, 470, 170, 430, 300]
```

[29]:
```
# Variance
print("Variance is  : ", np.var(dog_height))
```

```
Variance is  :  21704.0
```

1.2.4 4. Standard deviation

[30]:
```
# Load dataset
Maths = [85, 95, 75, 80, 90 ]
Science = [88, 79, 91, 85, 82]
```

[31]:
```
# Sum
print("Sum of math subject :",sum(Maths))
print("Sum of science subject :",sum(Science))
```

```
Sum of math subject : 425
Sum of science subject : 425
```

[32]:
```
import pandas as pd
Math = pd.DataFrame(Maths)
Math.describe()
```

[32]:
```
                0
count   5.000000
mean   85.000000
std     7.905694
min    75.000000
25%    80.000000
50%    85.000000
75%    90.000000
max    95.000000
```

[33]:
```
Sci = pd.DataFrame(Science)
Sci.describe()
```

[33]:
```
                0
count   5.000000
mean   85.000000
std     4.743416
min    79.000000
25%    82.000000
50%    85.000000
75%    88.000000
max    91.000000
```

2 2. BIVARIATE ANALYSIS

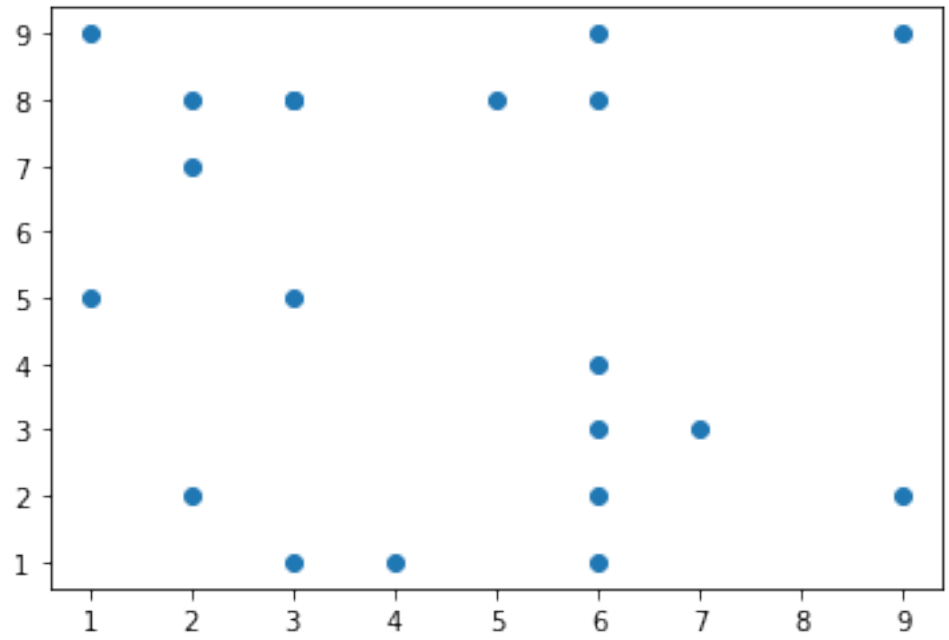
2.1 2.1 Scatter plot

[34]:
```
#Import libraries
import pandas as pd
from matplotlib import pyplot as plt
```

[35]:
```
# Load random set of values
x = np.random.randint(1,10,20)
y = np.random.randint(1,10,20)
print("x = ",x)
print("y = ",y)
```

```
x =  [6 6 4 5 6 3 6 7 3 2 1 2 3 1 6 9 6 3 9 2]
y =  [3 4 1 8 2 8 8 3 1 7 9 2 8 5 1 2 9 5 9 8]
```

[36]:
```
# Draw scatter plot
plt.scatter(x,y)
plt.show()
```



2.2 2.2 Covariance plot

[37]:
```
# Load random set of values
data = pd.read_csv("E:\\MY LECTURES\\DATA SCIENCE\\3.Programs\\dataset\\Meal␣
 ↪and Tips.csv")
data.head()
```

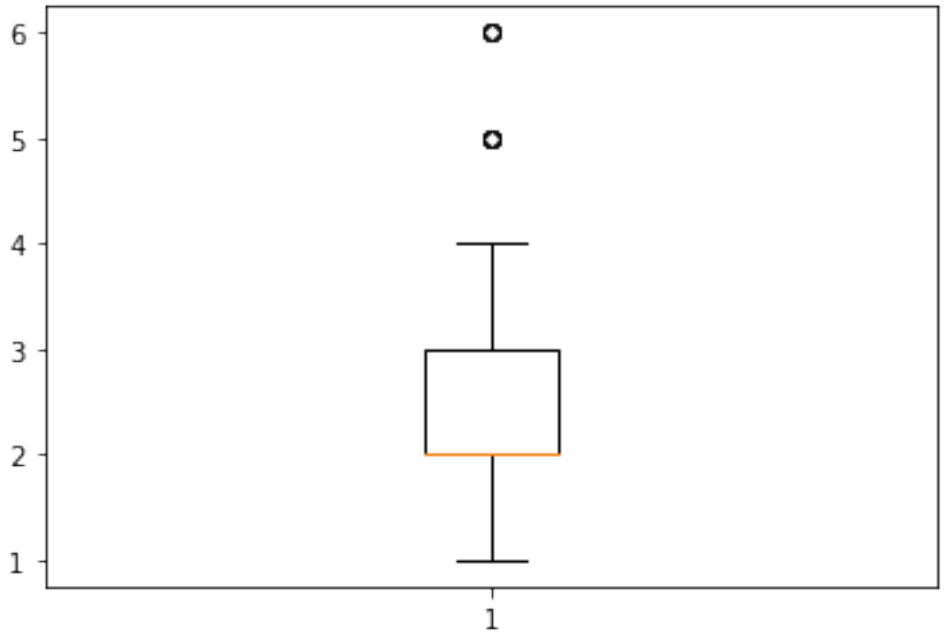
[37]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

[38]:
```
plt.boxplot(data["size"])
```

[38]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e90f167f10>,
  <matplotlib.lines.Line2D at 0x1e90f1742b0>],
 'caps': [<matplotlib.lines.Line2D at 0x1e90f174610>,
  <matplotlib.lines.Line2D at 0x1e90f174970>],
```

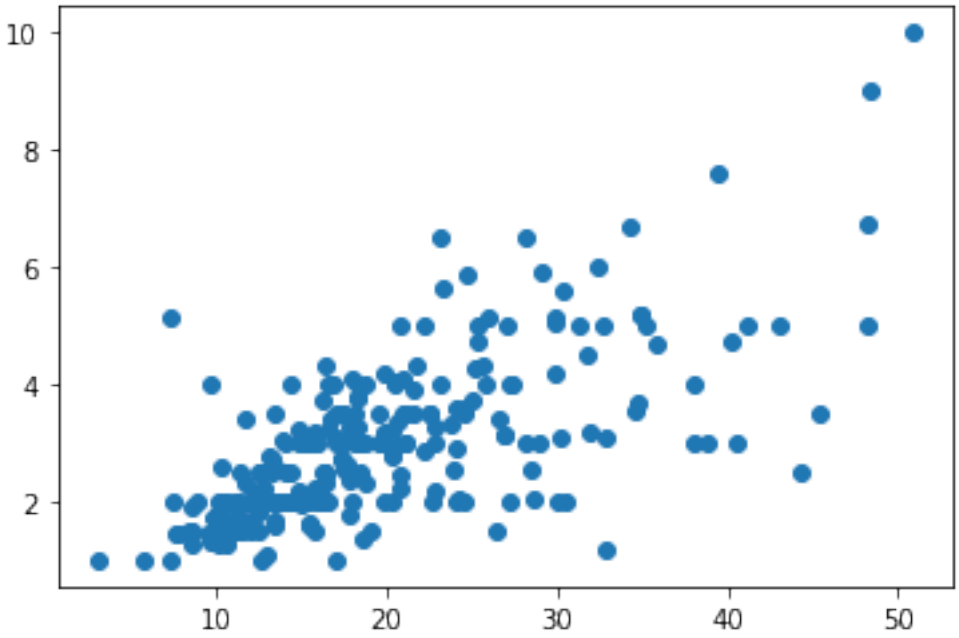
```
'boxes': [<matplotlib.lines.Line2D at 0x1e90f167bb0>],
'medians': [<matplotlib.lines.Line2D at 0x1e90f174cd0>],
'fliers': [<matplotlib.lines.Line2D at 0x1e90f17f070>],
'means': []}
```



[39]:
```
# Covariance calculation
x = np.array(data["total_bill"])
y = np.array(data["tip"])
covariance = np.cov(x, y)[0][1]
print(covariance)
```

```
8.323501629224854
```

[40]:
```
# Draw scatter plot
plt.scatter(x,y)
plt.show()
```

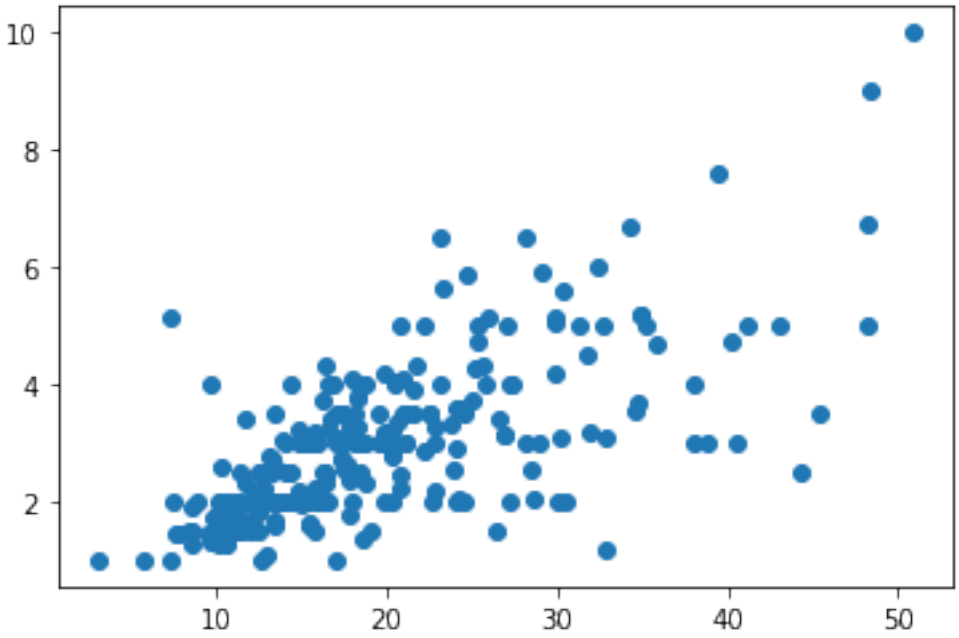


2.3 2.3 Correlation

[41]:
```
# Correlation calculation
x = np.array(data["total_bill"])
y = np.array(data["tip"])
covariance = np.corrcoef(x, y)[0][1]
print(covariance)
```

```
0.6757341092113641
```

[42]:
```
# Draw scatter plot
plt.scatter(x,y)
plt.show()
```



2.4 2.4 Cross tab

[43]:
```
import seaborn as sns
data1 = pd.read_csv("E:\\MY LECTURES\\DATA SCIENCE\\3.
 ↪Programs\\dataset\\bank-data.csv")
data1.head()
```

[43]:

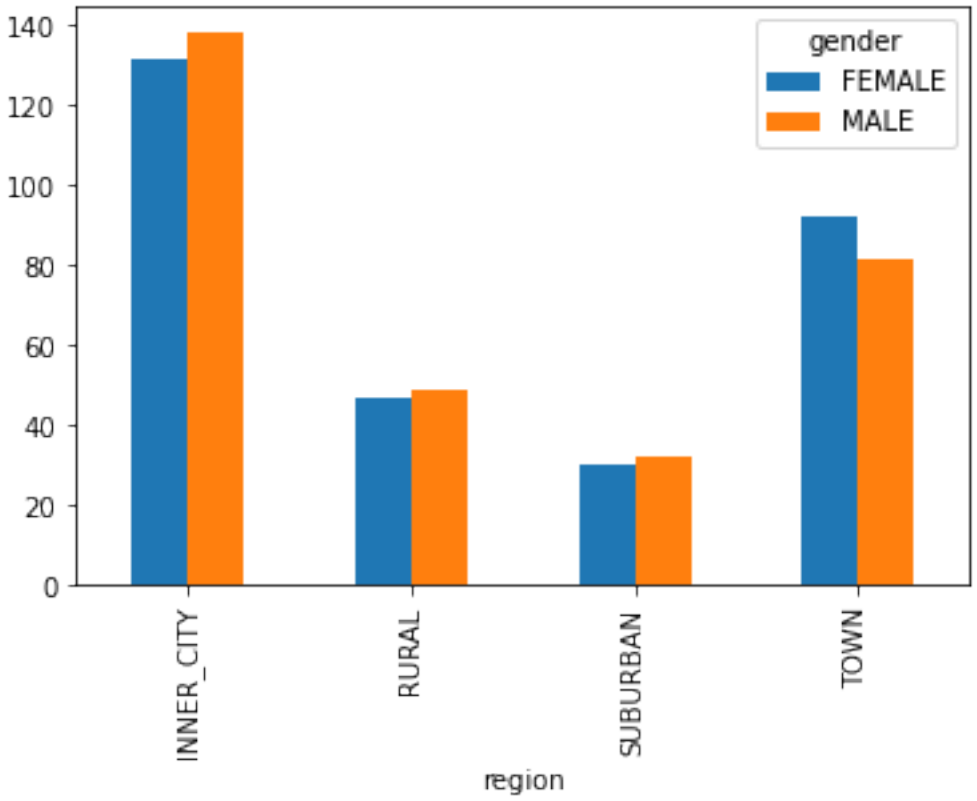
	id	age	gender	region	income	married	children	car	save_act	current_act	mortgage	pep
0	ID12101	48	FEMALE	INNER_CITY	17546.0	NO	1	NO	NO	NO	NO	YES
1	ID12102	40	MALE	TOWN	30085.1	YES	3	YES	NO	YES	YES	NO
2	ID12103	51	FEMALE	INNER_CITY	16575.4	YES	0	YES	YES	YES	NO	NO
3	ID12104	23	FEMALE	TOWN	20375.4	YES	3	NO	NO	YES	NO	NO
4	ID12105	57	FEMALE	RURAL	50576.3	YES	0	NO	YES	NO	NO	NO

[44]:
```
pd.crosstab(data1["region"],data1["gender"])
```

[44]:
```
gender       FEMALE  MALE
region
INNER_CITY      131   138
RURAL            47    49
SUBURBAN         30    32
TOWN             92    81
```

[45]:
```
pd.crosstab(data1["region"],data1["gender"]).plot(kind="bar")
```

[45]:
```
<AxesSubplot:xlabel='region'>
```



2.5 2.5 Chi-square test

[46]:
```
input = pd.crosstab(data1["region"],data1["gender"])
input
```

[46]:
```
gender       FEMALE  MALE
region
INNER_CITY      131   138
RURAL            47    49
```

```
SUBURBAN        30    32
TOWN            92    81
```

[47]:
```
from scipy.stats import chi2_contingency
stat, p, dof, expected = chi2_contingency(input)

# interpret p-value
s = 0.05
print("p value is ", round(p,3) )
print("p is greater than s")
print("gender has no relationship with area they come from")
```

```
p value is  0.804
p is greater than s
gender has no relationship with area they come from
```