institute of CODING

Northumbria University NEWCASTLE

White Frog

PLAN FOR JOBS

PORTFOLIO DATA SCIENCE BOOTCAMP

Priscila Boada

NORTHUMBRIA UNIVERSITY   NEWSCASTLE

Contents

Part 1 – Systems Data

Task 1 Document your pseudo-code

Looking at a systems approach to developing a self-driving vehicle solution, provide evidence of your pseudo-code that describes some common actions a self-driving vehicle might perform.

Pseudocode 1. Approaching Pelican crossings.

```
IF amber light is flashing
AND pedestrian wants to cross
THEN stop
ELSE proceed with caution
```

Pseudocode 2. Approaching Zebra crossings.

```
IF pedestrian wants to cross
    IF pedestrian is reasonable far
    THEN slow down
    ELSE stop
ELSE proceed with caution
END IF
```

Pseudocode 3. Turning right at crossroads

```
Use your mirrors
GIVE a right-turn signal well before you turn right
```

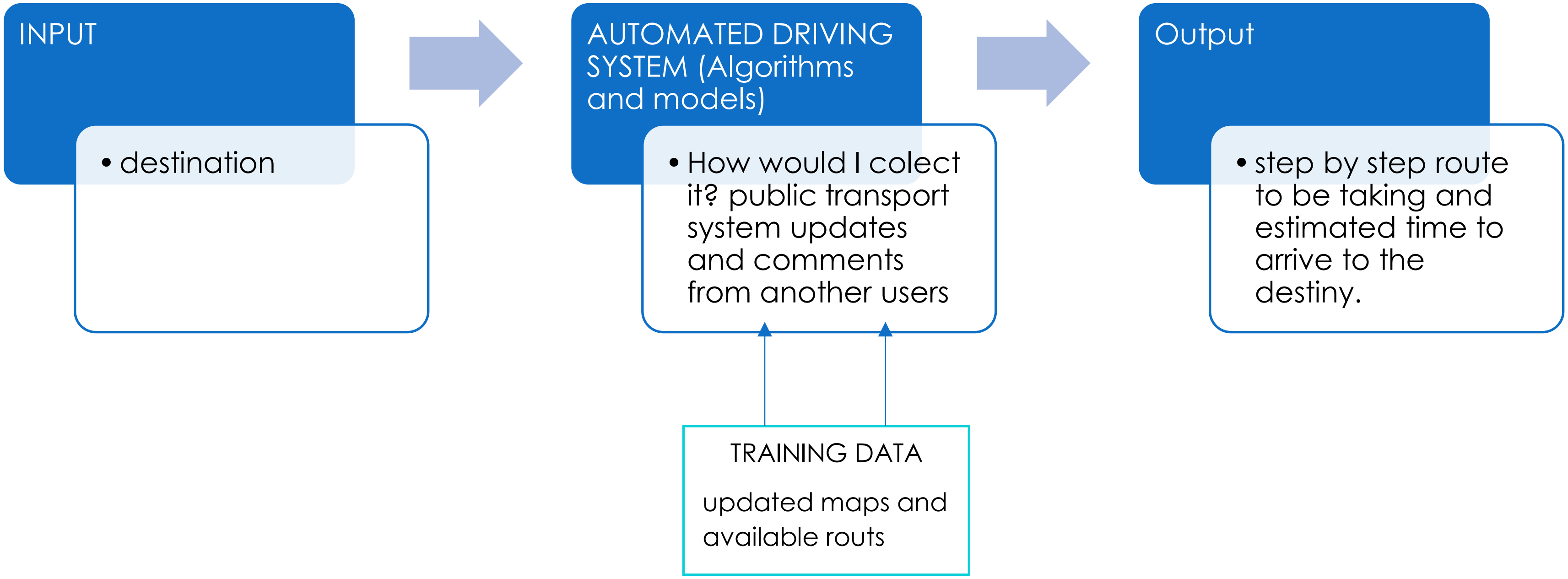
IF the oncoming vehicle is also turning right

THEN keep the other vehicle on your right and turn right behind it

ELSE turn right

Task 2 Training data

Think about the practicalities of building a self-driving vehicle system. What inputs might you want to gather; what training data would you use and how would you collect it; what outputs would you have to consider the possibility of (in addition to those shown in the figure above)?



INPUT

- destination

AUTOMATED DRIVING SYSTEM (Algorithms and models)

- How would I colect it? public transport system updates and comments from another users

Output

- step by step route to be taking and estimated time to arrive to the destiny.

TRAINING DATA

updated maps and available routs

Task 3 Human activity replication

What tacit skills do we utilise, when driving, that are inherently difficult to program?

- **Ability to decide under unexpected conditions**: For example, if someone is breaking the law and not stopping with a red traffic light, the self-driving vehicle may not know how to manoeuvre under those circumstances. Another example is the decisions that a self-driving car may have to take under extreme weather conditions since some routes may be affected with no previous notice and in that case, the vehicle would not know if the most reasonable is to stop or change the route.

- **Ability to read facial expressions:** Humans interactions include body language and facial expressions that allow taking decisions but self-driving vehicles may not be able to read this input and therefore take an incorrect turn. For example, when a pedestrian wants to cross the street usually makes eye contact with the driver to let them know that they are crossing, however, a machine cannot detect this external interaction.

How can we best replicate the human ability to devise solutions to totally unfamiliar or unforeseen problems?

By creating several fictitious scenarios that may train the program to make a reasonable decision. Furthermore, it could be added to the database series of examples of what humans did under similar circumstances. Moreover, it could be added a machine learning algorithm that allows the interaction with humans, ask them what they should do under a specific condition so that next time they would learn the approach to be taken.

How do we choose the 'least bad' scenario when faced with an inevitable loss of life for one or more parties?

The first action to be taken would be to minimize the number of deaths. Furthermore, depending on the impact, I believe that it should be considered to prioritize the life of the person that would have a bigger probability of surviving. However, because it is a moral dilemma, it could be considered to randomize the victims but always give priority to save as many people as possible.

Task 4 Document insights

Document the insights that you got in response to the questions raised in Exercise 2 by the PaperBoss Ltd management team:

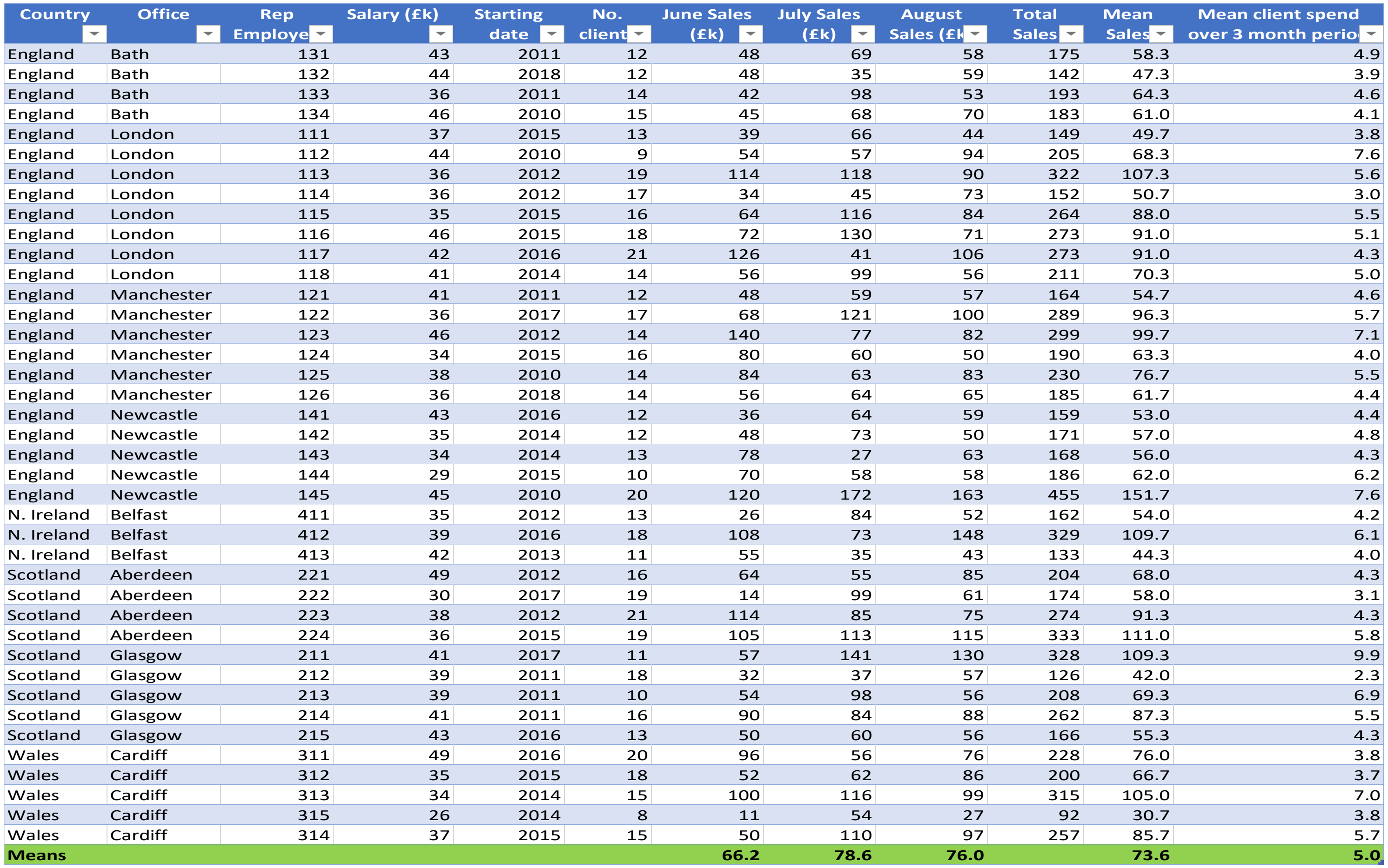
What is the mean client spend over the three months?

Solution:

1. Compute the average Sales over the three months per employee ID.

2. Divide the result by the number of clients that each Employee has.
3. From the result above, it is computed the mean and that value is the mean client.

Calculations:



Country	Office	Rep Employe	Salary (£k)	Starting date	No. client	June Sales (£k)	July Sales (£k)	August Sales (£k	Total Sales	Mean Sales	Mean client spend over 3 month perio
England	Bath	131	43	2011	12	48	69	58	175	58.3	4.9
England	Bath	132	44	2018	12	48	35	59	142	47.3	3.9
England	Bath	133	36	2011	14	42	98	53	193	64.3	4.6
England	Bath	134	46	2010	15	45	68	70	183	61.0	4.1
England	London	111	37	2015	13	39	66	44	149	49.7	3.8
England	London	112	44	2010	9	54	57	94	205	68.3	7.6
England	London	113	36	2012	19	114	118	90	322	107.3	5.6
England	London	114	36	2012	17	34	45	73	152	50.7	3.0
England	London	115	35	2015	16	64	116	84	264	88.0	5.5
England	London	116	46	2015	18	72	130	71	273	91.0	5.1
England	London	117	42	2016	21	126	41	106	273	91.0	4.3
England	London	118	41	2014	14	56	99	56	211	70.3	5.0
England	Manchester	121	41	2011	12	48	59	57	164	54.7	4.6
England	Manchester	122	36	2017	17	68	121	100	289	96.3	5.7
England	Manchester	123	46	2012	14	140	77	82	299	99.7	7.1
England	Manchester	124	34	2015	16	80	60	50	190	63.3	4.0
England	Manchester	125	38	2010	14	84	63	83	230	76.7	5.5
England	Manchester	126	36	2018	14	56	64	65	185	61.7	4.4
England	Newcastle	141	43	2016	12	36	64	59	159	53.0	4.4
England	Newcastle	142	35	2014	12	48	73	50	171	57.0	4.8
England	Newcastle	143	34	2014	13	78	27	63	168	56.0	4.3
England	Newcastle	144	29	2015	10	70	58	58	186	62.0	6.2
England	Newcastle	145	45	2010	20	120	172	163	455	151.7	7.6
N. Ireland	Belfast	411	35	2012	13	26	84	52	162	54.0	4.2
N. Ireland	Belfast	412	39	2016	18	108	73	148	329	109.7	6.1
N. Ireland	Belfast	413	42	2013	11	55	35	43	133	44.3	4.0
Scotland	Aberdeen	221	49	2012	16	64	55	85	204	68.0	4.3
Scotland	Aberdeen	222	30	2017	19	14	99	61	174	58.0	3.1
Scotland	Aberdeen	223	38	2012	21	114	85	75	274	91.3	4.3
Scotland	Aberdeen	224	36	2015	19	105	113	115	333	111.0	5.8
Scotland	Glasgow	211	41	2017	11	57	141	130	328	109.3	9.9
Scotland	Glasgow	212	39	2011	18	32	37	57	126	42.0	2.3
Scotland	Glasgow	213	39	2011	10	54	98	56	208	69.3	6.9
Scotland	Glasgow	214	41	2011	16	90	84	88	262	87.3	5.5
Scotland	Glasgow	215	43	2016	13	50	60	56	166	55.3	4.3
Wales	Cardiff	311	49	2016	20	96	56	76	228	76.0	3.8
Wales	Cardiff	312	35	2015	18	52	62	86	200	66.7	3.7
Wales	Cardiff	313	34	2014	15	100	116	99	315	105.0	7.0
Wales	Cardiff	315	26	2014	8	11	54	27	92	30.7	3.8
Wales	Cardiff	314	37	2015	15	50	110	97	257	85.7	5.7
Means						**66.2**	**78.6**	**76.0**		**73.6**	**5.0**

Result:

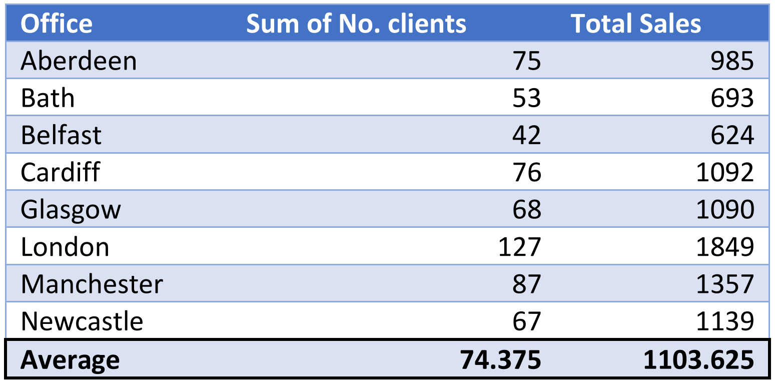
Mean client-spend over the three months: 5.0 (£K)

Are there any underperforming offices, compared to the others?

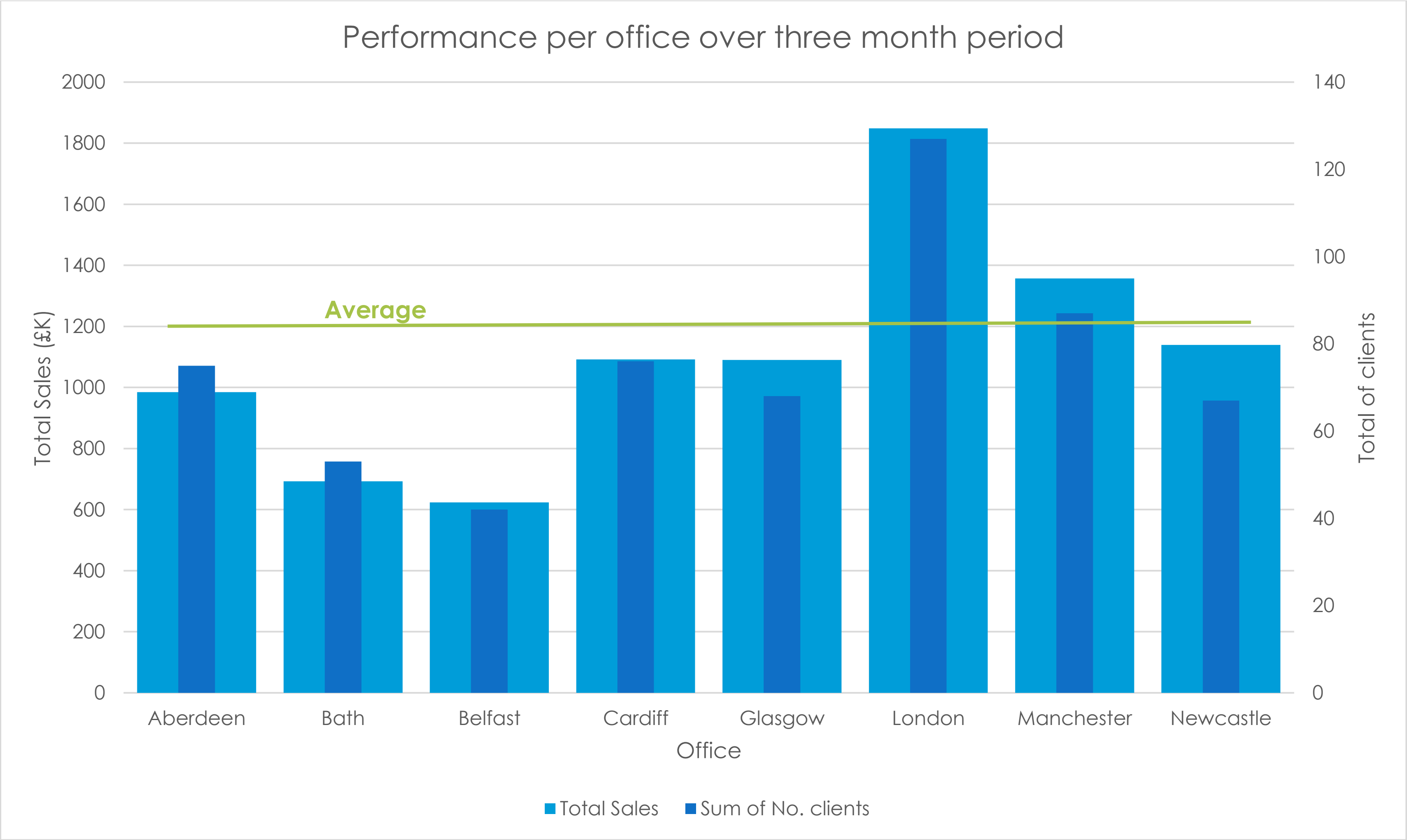
Solution:

1. Compute the total sales per period (June Sales+ July Sales+ August Sales)
2. Compute the total sales per period per office
3. Represent graphically the results to have a clearer view of the sales per office.
4. Compute the mean of total sales per period per office to have a reference and conclude.

Calculations:



Office	Sum of No. clients	Total Sales
Aberdeen	75	985
Bath	53	693
Belfast	42	624
Cardiff	76	1092
Glasgow	68	1090
London	127	1849
Manchester	87	1357
Newcastle	67	1139
Average	**74.375**	**1103.625**



Result

Comparing the total sales and clients over the three months, it is noticed Belfast and Bath offices are way under the average. This leads to the conclusion that both offices are underperforming compared to the others.

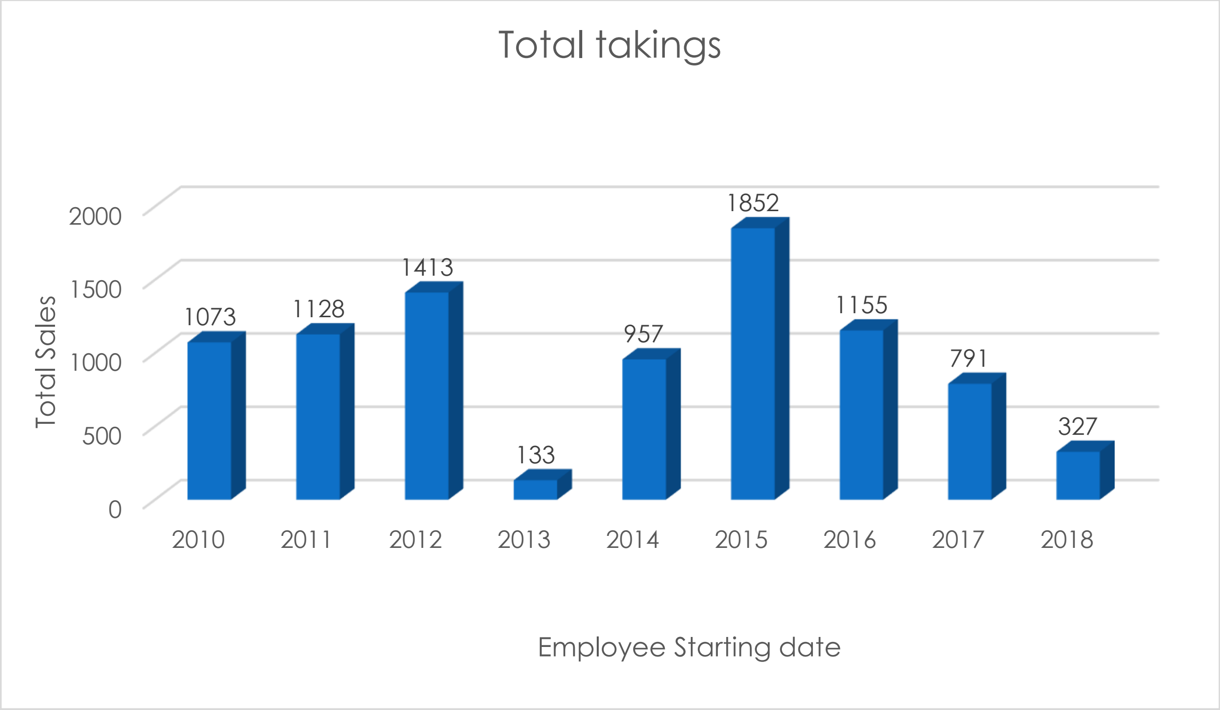
Does sales rep experience have any impact on sales (measured in total takings)?

Solution:

1. Compute the total sales over the three months per representant Employee
2. Compute the total sales per employee per entry year
3. Represent the information graphically and comment on the trends.

Calculations:

Rep Employee ID	Sum of Total Sales
2010	1073
2011	1128
2012	1413
2013	133
2014	957
2015	1852
2016	1155
2017	791
2018	327
Grand Total	**8829**



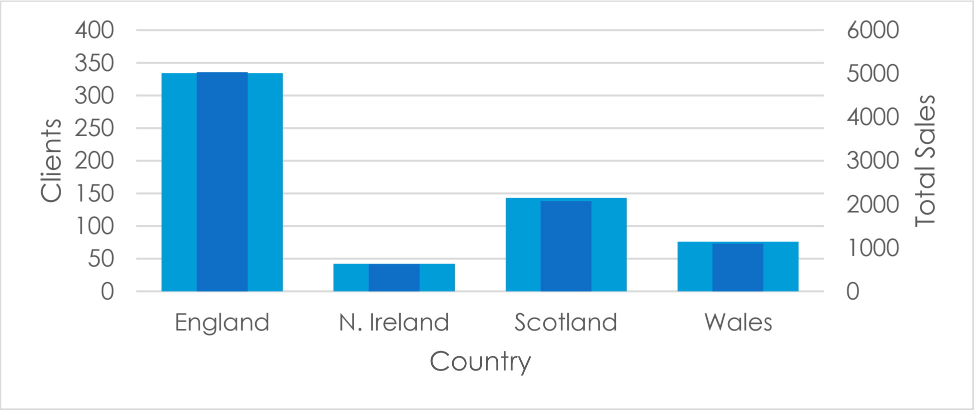
Result

From the data above it is noticed that there is no association between the sales rep experience and the impact on sales. Since both variables are independent, it is concluded that **the sales rep experience does not have any impact on sales.**

Additional insights

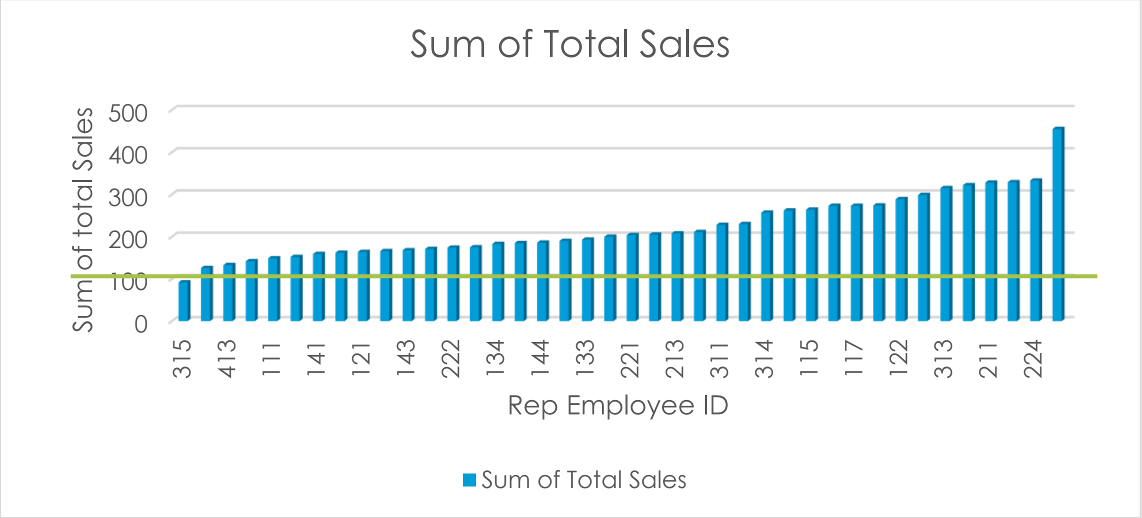
- England is the country with the largest number of clients and amount of sales, then follows Scotland, Wales and N. Ireland.

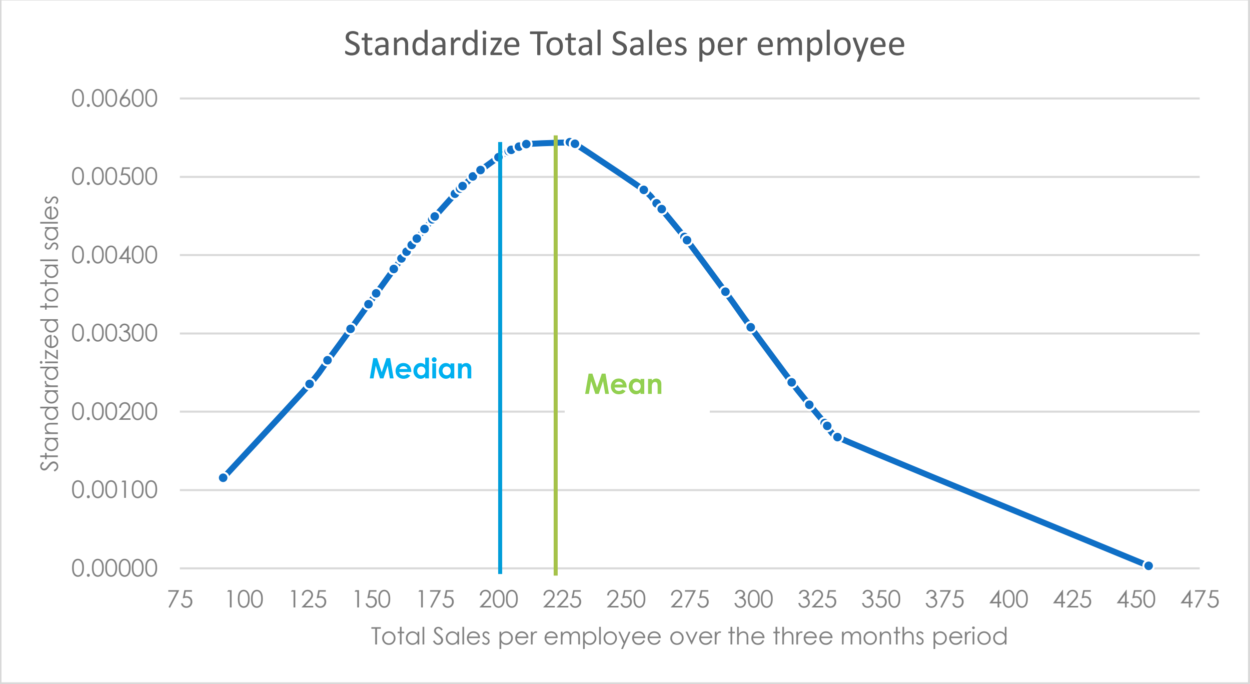
Country	Sum of Total Sales	Sum of No. clients
England	5038	334
N. Ireland	624	42
Scotland	2075	143
Wales	1092	76
Grand Total	**8829**	**595**



- The minimum salary is 26 £K, the average is 38.9£K and the maximum salary is 49£K

- The impact on sales is closely correlated to the representant Employee





Total Sales	8829
Mean Sales	220.725
Standard deviation Sales	72.993
Median Sales	202

- From the standardized total takings per Rep employee, it is seen that the mean is to the right of the median, this indicates that the curve is right-skewed, which means that there are more sales above the average than the ones below the average. Moreover, the shape of the bell indicates that the standard deviation is large, this implies that there is a difference between the performance of one employee and the other. To add more, there is one outlier employee (ID 145) who stands out from ` others and there is also another outlier employee (ID 315) who underperforms compared to the others. Taking all into consideration, it is concluded that the **impact on sales depends on the Rep employee**.

Part 2 – Working with data

Task 1 List required data types

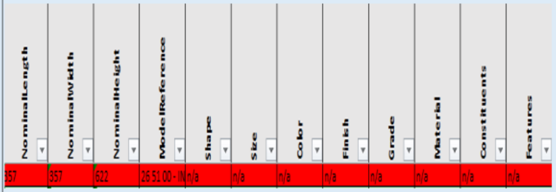
Consider the task of replacing a single lightbulb in a dormitory. What data would the maintenance person need to carry out that task? List all the things you think the maintenance person would need.

- Location of the dormitory
- Type of bulb: cap fitting, technology (Halogen, Energy Saving (CFL) or LED), brightness level (lumens and Watts), Warm or cold light.
- Lifespan of bulb.
- Equipment to change the bulb: ladders, electrical material.
- Time that takes this process and other logistic information
- Supplier of the bulb and others equipments
- Cost of material
- Staff in charge of this task

Task 2 List available data types

Using your list from task 1 explore the COBie data and list all the data you can get access to – some of this should hopefully be available in the COBie file. Make a note of which tab it sits under – there may be information that is helpful located in different tabs of the spreadsheet.

It's also useful to highlight what is missing in the COBie file that might be useful. For existing buildings, you could add this in to help with future maintenance tasks. For new buildings you can make sure you request this data in future data exchanges. NB: There's something very obvious missing from the spreadsheet that would be useful to have for this task.

Data	**Location in COBie file**			**Things useful to add**
	Tab	**Column**	**Information**	
Location of the dormitory	Facility	name	East Dormitory	
		ProjectDescription	One of Four Dormitories on the E.Healey Site	
Type of bulb: cap fitting, technology (Halogen, Energy Saving (CFL) or LED), brightness level (lumens and Watts), Warm or cold light.	Type	ModelNumber NominalLength NominalWidth NominalHeight ModelReference Shape Size Color Finish Grade Material Constituents Features		Link the type of material (column name) with the facility. As it is, it is not possible to know what belongs where.
Lifespan of bulb.	Type	ExpectedLife, DurationUnit	n/a, year	In tab Job add when the bulbs will need to be changed according to their lifespan
Equipment to change the bulb: ladders, electrical material.	Resources	Description		Add components required to proceed to change the bulb
Time that takes this process and other logistic information	Job			Add a column that provides an estimated time to perform this task
Supplier of the bulb and others equipments	Type	Supplier	gdunstan@ses-ltd.co.uk	
Cost of material	Type	ReplacementCost		
Staff in charge of this task	Job			Add a column that includes who will be the responsible of this activity

NominalLength	NominalWidth	NominalHeight	ModelReference	Shape	Size	Color	Finish	Grade	Material	Constituents	Features
357	357	622	26 51 00 - IN	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

Task 3 Identify information gaps

It's not always relevant to put all the data you need in a COBie spreadsheet. Are there any additional bits of information that might be useful to help the maintenance person carry out their task? You might need to think outside the box for this one – for example what might someone need to change a bulb in a room that has very tall ceilings, or what might someone need to know if the bulb was located in a suspended ceiling.

Data	Link information to Location in COBie file		Things useful to add
	Tab	**Column**	
Lifespan of bulb.	Type	ExpectedLife, DurationUnit	Link to tab Job and add when the bulbs will need to be changed according to their lifespan
Equipment to change the bulb: ladders, electrical material.	Resources	Description	Link to tab Resources and add components required to proceed to change the bulb
Time that takes this process and other logistic information	Job		Link to Job and provide an estimated time to perform this task
Supplier of the bulb and others equipments	Type	Supplier	State how long takes to get bulb from the supplier
Staff in charge of this task	Job		Link to tab job and add the responsible of this activity

Task 4 Document your pseudo-code

For this task you are to develop some pseudocode to create reports from your COBie data on the following enquiries about light fixtures. A light fixture is any component that requires a light bulb.

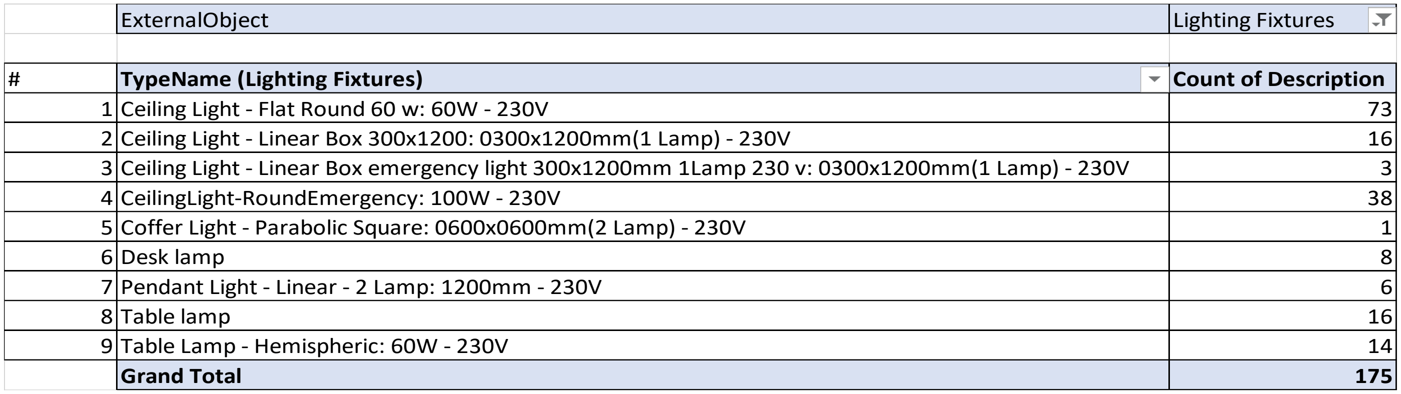
Remember you need to plan the logic of how you might find and extract this information from the spreadsheet. Once you have the logic you can define your logic using pseudocode notation:

a) How many light fixtures are there in the facility?

Pseudocode

```
lights←0
Get first ExternalObject
WHILE (ExternalObject = 'Lighting Fixtures') AND (There are more elements)
        lights← lights+1
        Get next ExternalObject
ENDWHILE
print 'The total of Lighting Fixtures is:' ligths
```

Result



	ExternalObject	Lighting Fixtures
#	**TypeName (Lighting Fixtures)**	**Count of Description**
1	Ceiling Light - Flat Round 60 w: 60W - 230V	73
2	Ceiling Light - Linear Box 300x1200: 0300x1200mm(1 Lamp) - 230V	16
3	Ceiling Light - Linear Box emergency light 300x1200mm 1Lamp 230 v: 0300x1200mm(1 Lamp) - 230V	3
4	CeilingLight-RoundEmergency: 100W - 230V	38
5	Coffer Light - Parabolic Square: 0600x0600mm(2 Lamp) - 230V	1
6	Desk lamp	8
7	Pendant Light - Linear - 2 Lamp: 1200mm - 230V	6
8	Table lamp	16
9	Table Lamp - Hemispheric: 60W - 230V	14
	Grand Total	**175**

There are 175 light fixtures

b) How many different types of light fixtures are there in the facility?

Pseudocode

```
Get first ExternalObject
WHILE (ExternalObject = ‘Lighting Fixtures’) AND (There are more elements)
        Print the Description of the ExternalObject
        Get next ExternalObject
ENDWHILE
Count unique value from the list of Description
print ‘There are’ result count ‘of different types of light fixtures’
```

Result
There are 9 types of light fixtures (see table above as reference)

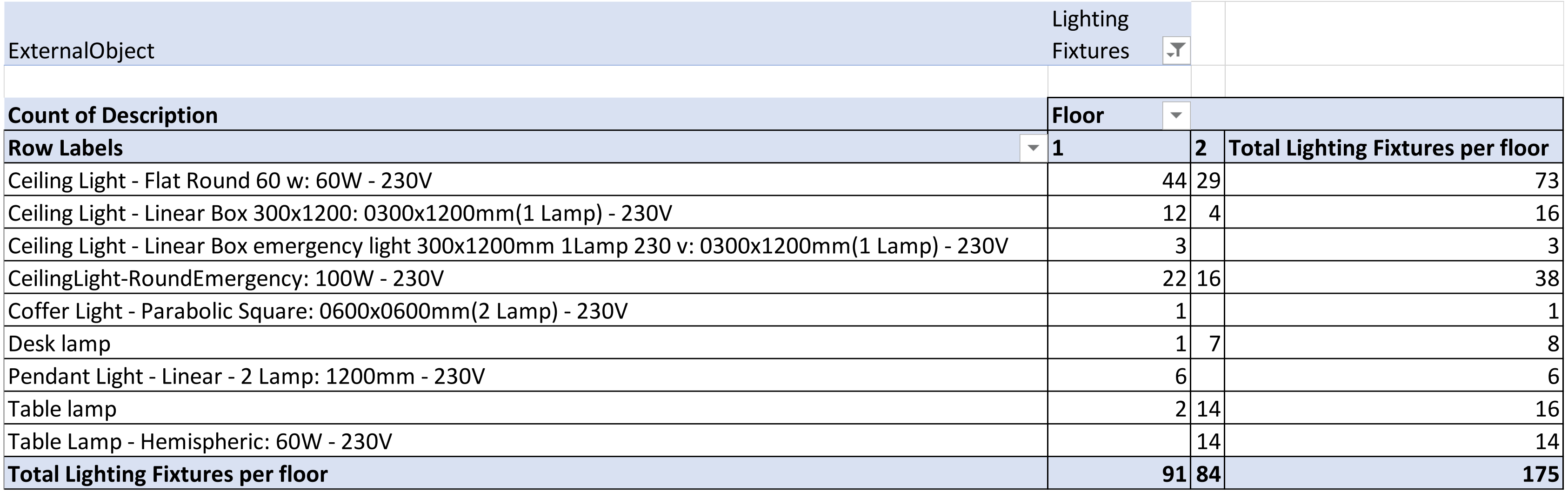
c) How many different light fixtures are there on each floor (e.g. how many on level 1, how many on level 2 and how many on level 3)?

Pseudocode

```
lightsFloor1←0
lightsFloor2←0
lightsFloor3←0

Get first ExternalObject
WHILE (ExternalObject = ‘Lighting Fixtures’) AND (There are more elements)
        IF (Space=’1*’)
                lightsFloor1← lightsFloor1+1;
        ELSE IF (Space=’2*’)
                LightsFloor2← lightsFloor2+1;
        ELSE
                LightsFloor3← lightsFloor3+1;
        Get next ExternalObject
```

```
ENDWHILE
print 'The total of Lighting Fixtures on level 1 is:' lightsFloor1
print 'The total of Lighting Fixtures on level 2 is:' lightsFloor2
print 'The total of Lighting Fixtures on level 3 is:' lightsFloor3
```

Result



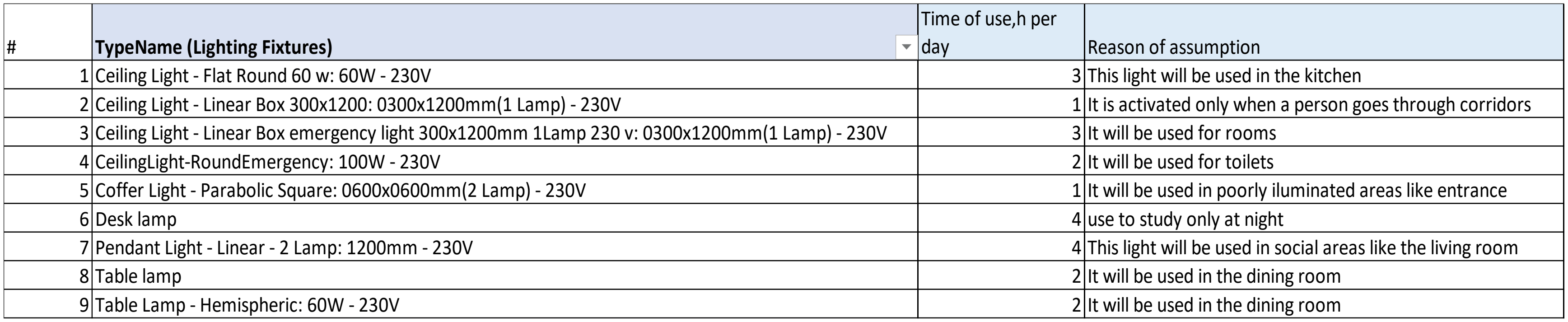
ExternalObject	Lighting Fixtures		
Count of Description	**Floor**		
Row Labels	**1**	**2**	**Total Lighting Fixtures per floor**
Ceiling Light - Flat Round 60 w: 60W - 230V	44	29	73
Ceiling Light - Linear Box 300x1200: 0300x1200mm(1 Lamp) - 230V	12	4	16
Ceiling Light - Linear Box emergency light 300x1200mm 1Lamp 230 v: 0300x1200mm(1 Lamp) - 230V	3		3
CeilingLight-RoundEmergency: 100W - 230V	22	16	38
Coffer Light - Parabolic Square: 0600x0600mm(2 Lamp) - 230V	1		1
Desk lamp	1	7	8
Pendant Light - Linear - 2 Lamp: 1200mm - 230V	6		6
Table lamp	2	14	16
Table Lamp - Hemispheric: 60W - 230V		14	14
Total Lighting Fixtures per floor	**91**	**84**	**175**

There are 91 different light fixtures for Level 1, 84 for Level 2 and 0 for Level 3

d) Let's say each light fixture uses the 'Philips Energy Saving CFL Stick Lamp 18W BC (B22d) 1100lm'bulb. This bulb has an average life 6000 hours. If each bulb was installed at the same time and used in the same way e.g. 8 hours per day, how many months do you think it would be before the bulb needed replacing?

$$\frac{6000\ \text{h}}{} \;\Big|\; \frac{1\ \text{day}}{8\ \text{h}} \;\Big|\; \frac{1\ \text{month}}{30\ \text{Days}} \;\Big| = 25\ \text{months}$$

e) Look at the different types of light fixtures you have in your results from task 4b. Make some assumptions about the usage of each bulb and write your assumptions down e.g. a desk lamp may be used for only 2 hours a day, but a ceiling light in a dormitory room might be used for 8 hours. List each light fixture type and the assumed usage per day for each type.



#	TypeName (Lighting Fixtures)	Time of use,h per day	Reason of assumption
1	Ceiling Light - Flat Round 60 w: 60W - 230V	3	This light will be used in the kitchen
2	Ceiling Light - Linear Box 300x1200: 0300x1200mm(1 Lamp) - 230V	1	It is activated only when a person goes through corridors
3	Ceiling Light - Linear Box emergency light 300x1200mm 1Lamp 230 v: 0300x1200mm(1 Lamp) - 230V	3	It will be used for rooms
4	CeilingLight-RoundEmergency: 100W - 230V	2	It will be used for toilets
5	Coffer Light - Parabolic Square: 0600x0600mm(2 Lamp) - 230V	1	It will be used in poorly iluminated areas like entrance
6	Desk lamp	4	use to study only at night
7	Pendant Light - Linear - 2 Lamp: 1200mm - 230V	4	This light will be used in social areas like the living room
8	Table lamp	2	It will be used in the dining room
9	Table Lamp - Hemispheric: 60W - 230V	2	It will be used in the dining room

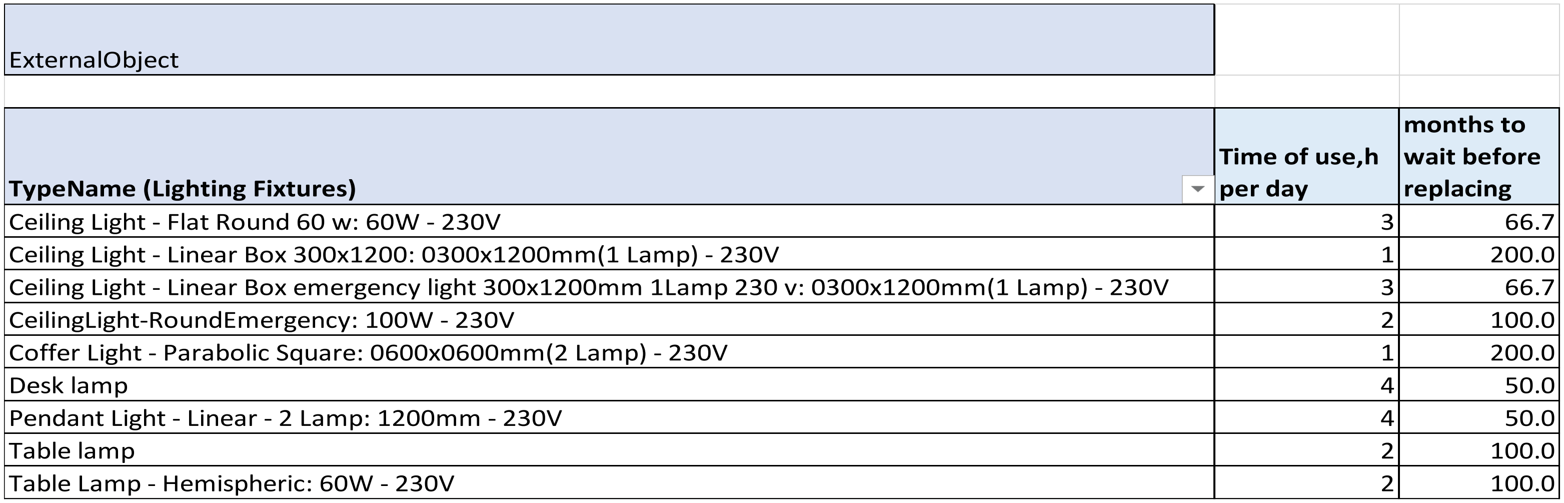
f) Based on your assumptions from the last task, write some pseudocode that predicts how many months after installation, a light bulb will need replacing, for each light fixture type.

Pseudocode

Starting from unique values known (taken from tab components column description)

```
TimeOfUsePerTypeName=[(CeilingLightFlatRound60w:60W-230V,3), (CeilingLight-LinearBox300x1200:0300x1200mm(1Lamp)-230V,1), (CeilingLight-LinearBoxemergencylight300x1200mm1Lamp230v:0300x1200mm(1Lamp)-230V,3), (CeilingLight-RoundEmergency:100W-230V,2), (CofferLight-ParabolicSquare:0600x0600mm(2Lamp)-230V,1), (Desklamp,4), (PendantLight-Linear-2Lamp:1200mm-230V,4), (Tablelamp,2) (TableLamp-Hemispheric:60W-230V,2)] ←0

Get first TimeOfUsePerTypeName
WHILE (TimeOfUsePerTypeName exists)
    Lifespan=6000/ (TimeOfUsePerTypeName[1]*30]
    print TimeOfUsePerTypeName[0], TimeOfUsePerTypeName[1], Lifespan
    Get next TimeOfUsePerTypeName
ENDWHILE
```

Result



ExternalObject		
TypeName (Lighting Fixtures)	**Time of use,h per day**	**months to wait before replacing**
Ceiling Light - Flat Round 60 w: 60W - 230V	3	66.7
Ceiling Light - Linear Box 300x1200: 0300x1200mm(1 Lamp) - 230V	1	200.0
Ceiling Light - Linear Box emergency light 300x1200mm 1Lamp 230 v: 0300x1200mm(1 Lamp) - 230V	3	66.7
CeilingLight-RoundEmergency: 100W - 230V	2	100.0
Coffer Light - Parabolic Square: 0600x0600mm(2 Lamp) - 230V	1	200.0
Desk lamp	4	50.0
Pendant Light - Linear - 2 Lamp: 1200mm - 230V	4	50.0
Table lamp	2	100.0
Table Lamp - Hemispheric: 60W - 230V	2	100.0

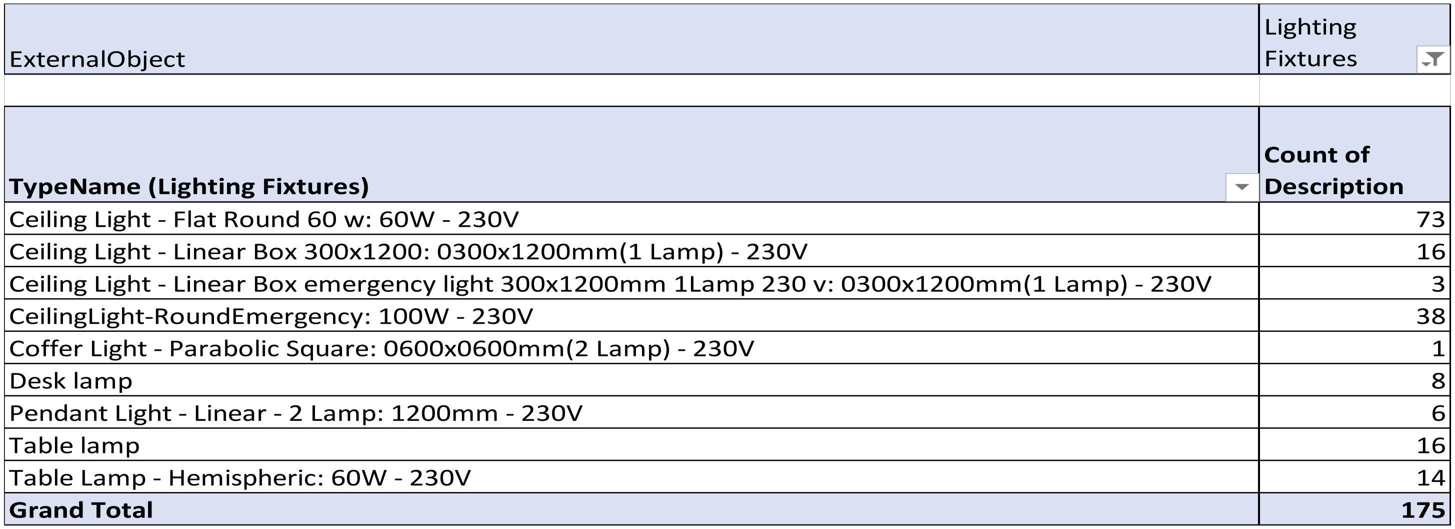
g) Extend your pseudocode from the last task to predict how many light bulbs will need replacing for each fixture type.

Pseudocode

Starting from the information in the tab component

```
Get first ExternalObject
WHILE (ExternalObject = 'Lighting Fixtures') AND (There are more elements)
    Get Description
    Get next ExternalObject
ENDWHILE
Get UniqueValueDescription from Description
UniqueValueDescription←0
For [x] in UniqueValueDescription
    numberOfLigthBulbToReplace [x]=count of [x] in UniqueValueDescription
    print (UniqueValueDescription, numberOfLigthBulbToReplace)
ENDFOR
```

Result

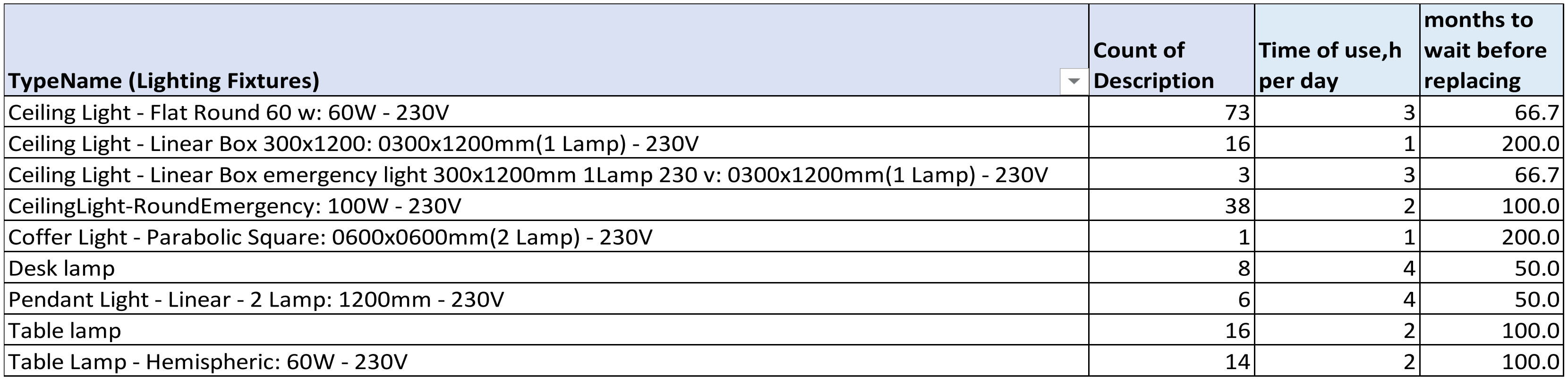


ExternalObject	Lighting Fixtures

TypeName (Lighting Fixtures)	Count of Description
Ceiling Light - Flat Round 60 w: 60W - 230V	73
Ceiling Light - Linear Box 300x1200: 0300x1200mm(1 Lamp) - 230V	16
Ceiling Light - Linear Box emergency light 300x1200mm 1Lamp 230 v: 0300x1200mm(1 Lamp) - 230V	3
CeilingLight-RoundEmergency: 100W - 230V	38
Coffer Light - Parabolic Square: 0600x0600mm(2 Lamp) - 230V	1
Desk lamp	8
Pendant Light - Linear - 2 Lamp: 1200mm - 230V	6
Table lamp	16
Table Lamp - Hemispheric: 60W - 230V	14
Grand Total	**175**

h) Extend your pseudocode from your last task to state what month and year each light bulb type will need replacing based on an installation date of September 2021.

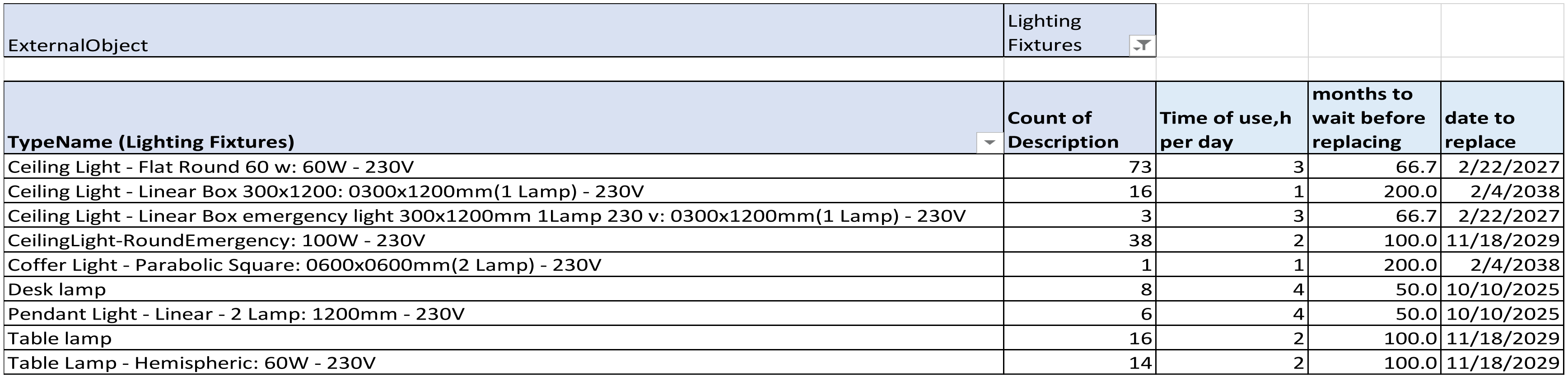
Starting from the values obtained in part f) and g)



TypeName (Lighting Fixtures)	Count of Description	Time of use,h per day	months to wait before replacing
Ceiling Light - Flat Round 60 w: 60W - 230V	73	3	66.7
Ceiling Light - Linear Box 300x1200: 0300x1200mm(1 Lamp) - 230V	16	1	200.0
Ceiling Light - Linear Box emergency light 300x1200mm 1Lamp 230 v: 0300x1200mm(1 Lamp) - 230V	3	3	66.7
CeilingLight-RoundEmergency: 100W - 230V	38	2	100.0
Coffer Light - Parabolic Square: 0600x0600mm(2 Lamp) - 230V	1	1	200.0
Desk lamp	8	4	50.0
Pendant Light - Linear - 2 Lamp: 1200mm - 230V	6	4	50.0
Table lamp	16	2	100.0
Table Lamp - Hemispheric: 60W - 230V	14	2	100.0

```
Get first TypeName
Get first lifespanmonth
WHILE (lifespanmonth exists)
    Take first month
        yearlifespan= month/12
        residualmonthlifespan= (yearlifespan- integer(yearlifespan))*12
        residualdaylifespan= (residualmonthlifespan - integer(residualmonthlifespan))*30
        day=1+ residualmonthlifespan
        month=9+ residualmonthlifespan
        IF month >12
            month=month-12
            year=1+ integer(yearlifespan)+2021
        ELSE
            year= integer(yearlifespan)+2021
        ENDIF
    print (TypeName,year,month,day)
    Get next TypeName
    Get next lifespanmonth
ENDWHILE
```

Results



ExternalObject	Lighting Fixtures

TypeName (Lighting Fixtures)	Count of Description	Time of use,h per day	months to wait before replacing	date to replace
Ceiling Light - Flat Round 60 w: 60W - 230V	73	3	66.7	2/22/2027
Ceiling Light - Linear Box 300x1200: 0300x1200mm(1 Lamp) - 230V	16	1	200.0	2/4/2038
Ceiling Light - Linear Box emergency light 300x1200mm 1Lamp 230 v: 0300x1200mm(1 Lamp) - 230V	3	3	66.7	2/22/2027
CeilingLight-RoundEmergency: 100W - 230V	38	2	100.0	11/18/2029
Coffer Light - Parabolic Square: 0600x0600mm(2 Lamp) - 230V	1	1	200.0	2/4/2038
Desk lamp	8	4	50.0	10/10/2025
Pendant Light - Linear - 2 Lamp: 1200mm - 230V	6	4	50.0	10/10/2025
Table lamp	16	2	100.0	11/18/2029
Table Lamp - Hemispheric: 60W - 230V	14	2	100.0	11/18/2029

Part 3 – Data Analysis

Task 1

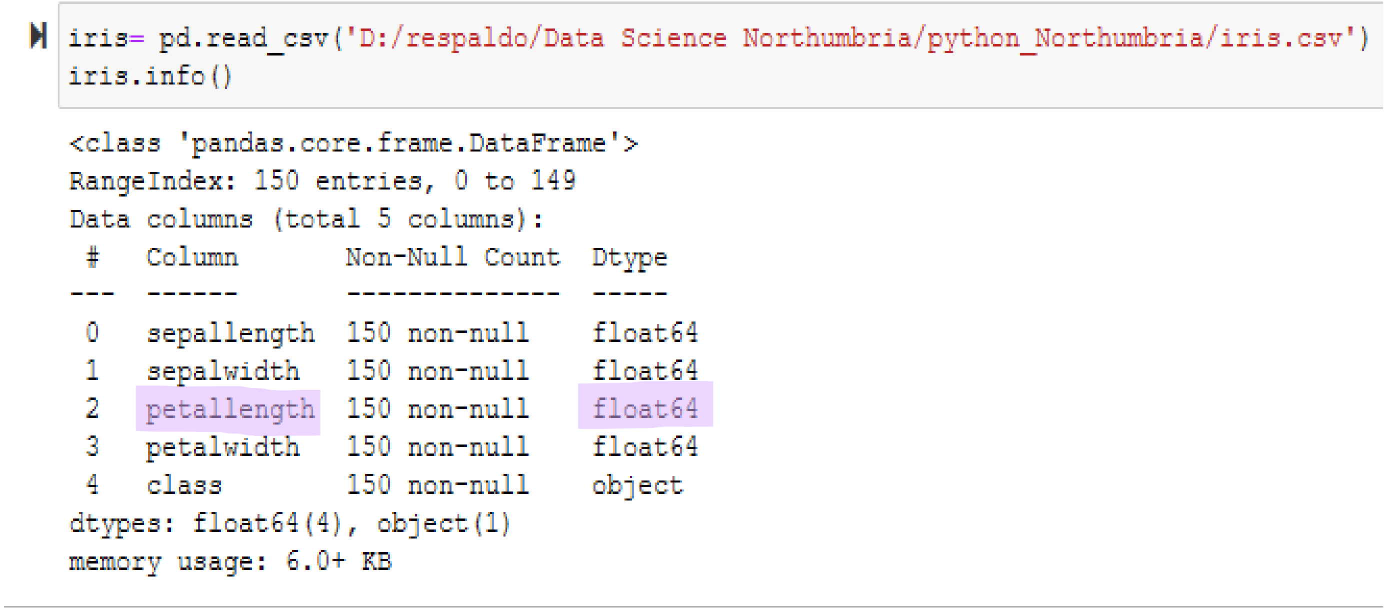
Using the iris dataset from week 4, load the iris data and convert the data to a dataframe.

Task 1.1 Type of data

Find out the structure of this dataframe. What data type has been used for Petal.Length?

It has been used a float64 as data type for petallength

Evidence



```
iris= pd.read_csv('D:/respaldo/Data Science Northumbria/python_Northumbria/iris.csv')
iris.info()
```

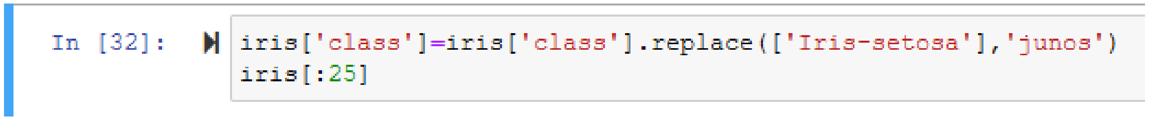
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   sepallength  150 non-null    float64
 1   sepalwidth   150 non-null    float64
 2   petallength  150 non-null    float64
 3   petalwidth   150 non-null    float64
 4   class        150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Task 1.2 Error correction method

You have received notice that all of the plants marked as "setosa" have been incorrectly labelled and should instead be recorded as "junos". Update the data to reflect this. Describe how you went about updating the plant label to correct the error in the data surrounding *Setosa* to *Junos*

To replace the value in the data Frame, we select the column class, which holds the description 'Iris-setosa' and then we replace this description by Junos

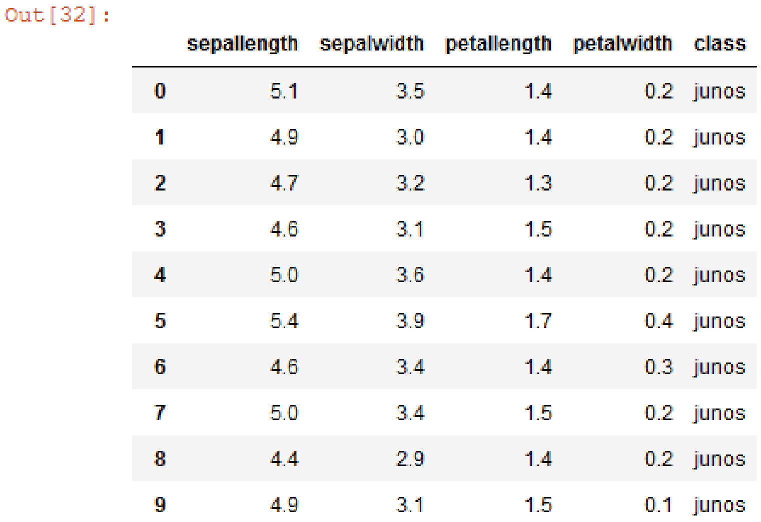
Evidence:



```
In [32]:  iris['class']=iris['class'].replace(['Iris-setosa'],'junos')
          iris[:25]
```

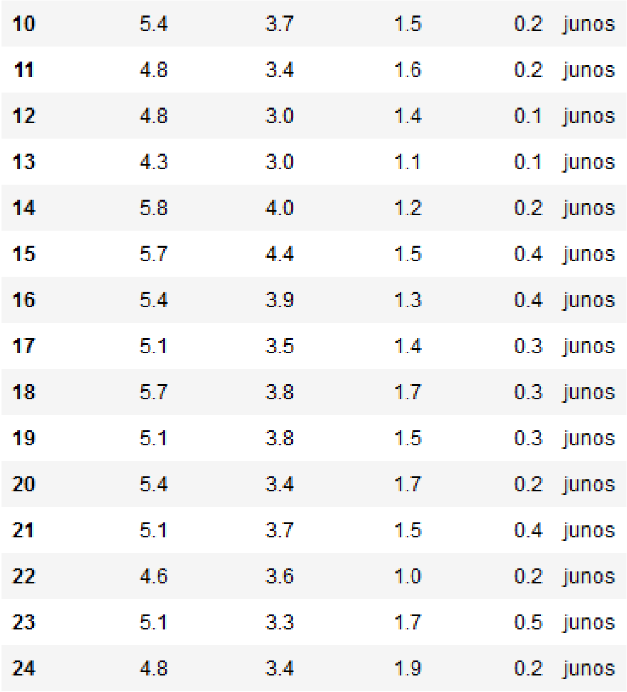
Task 1.3 Document your code

Print the first 25 observations of the "junos" class.



Out[32]:

	sepallength	sepalwidth	petallength	petalwidth	class
0	5.1	3.5	1.4	0.2	junos
1	4.9	3.0	1.4	0.2	junos
2	4.7	3.2	1.3	0.2	junos
3	4.6	3.1	1.5	0.2	junos
4	5.0	3.6	1.4	0.2	junos
5	5.4	3.9	1.7	0.4	junos
6	4.6	3.4	1.4	0.3	junos
7	5.0	3.4	1.5	0.2	junos
8	4.4	2.9	1.4	0.2	junos
9	4.9	3.1	1.5	0.1	junos



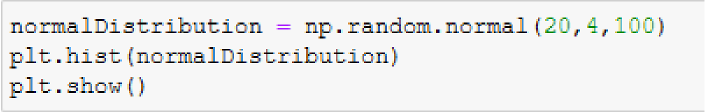
10	5.4	3.7	1.5	0.2	junos
11	4.8	3.4	1.6	0.2	junos
12	4.8	3.0	1.4	0.1	junos
13	4.3	3.0	1.1	0.1	junos
14	5.8	4.0	1.2	0.2	junos
15	5.7	4.4	1.5	0.4	junos
16	5.4	3.9	1.3	0.4	junos
17	5.1	3.5	1.4	0.3	junos
18	5.7	3.8	1.7	0.3	junos
19	5.1	3.8	1.5	0.3	junos
20	5.4	3.4	1.7	0.2	junos
21	5.1	3.7	1.5	0.4	junos
22	4.6	3.6	1.0	0.2	junos
23	5.1	3.3	1.7	0.5	junos
24	4.8	3.4	1.9	0.2	junos

Task 2 Evidence your histograms

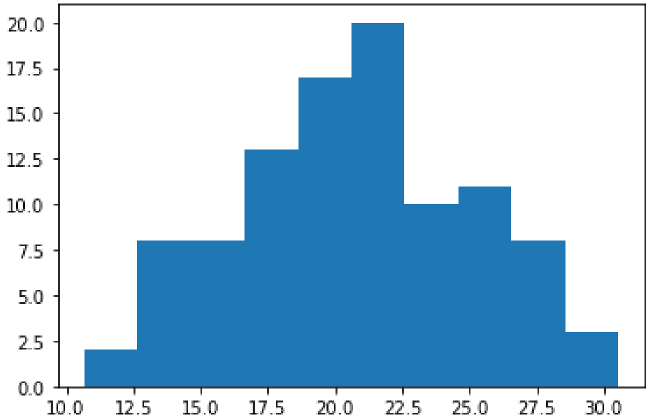
Create a dataset of 100 observations with 3 features. The first should follow a normal distribution, with a mean of 20 and a standard deviation of 4. The second feature should follows uniform distribution, with values between 15 and 25, and the third has a poisson distribution with a lambda value of 5.

a) Produce a series of histograms showing the distribution of each variable.

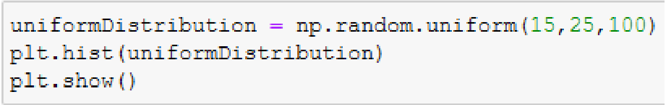
Normal distribution



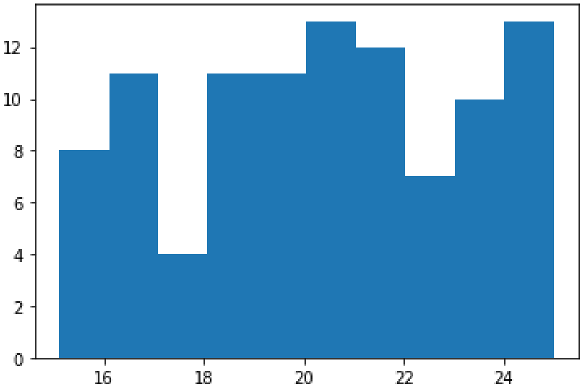
```
normalDistribution = np.random.normal(20,4,100)
plt.hist(normalDistribution)
plt.show()
```



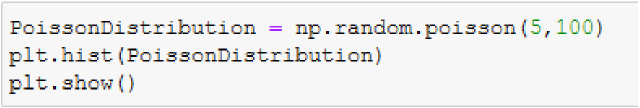
Uniform Distribution



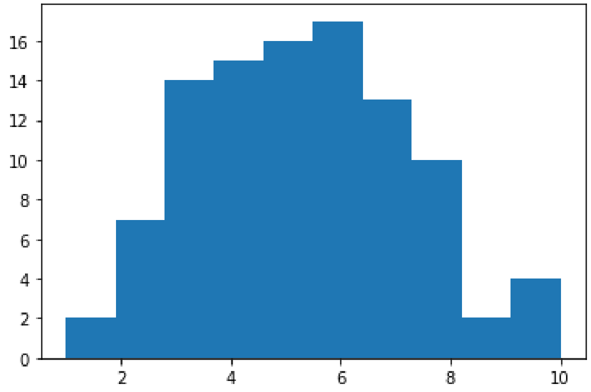
```
uniformDistribution = np.random.uniform(15,25,100)
plt.hist(uniformDistribution)
plt.show()
```



Poisson Distribution



```
PoissonDistribution = np.random.poisson(5,100)
plt.hist(PoissonDistribution)
plt.show()
```



b) Save this dataset as a .csv file

See CSV file Appendix 1. Distributions

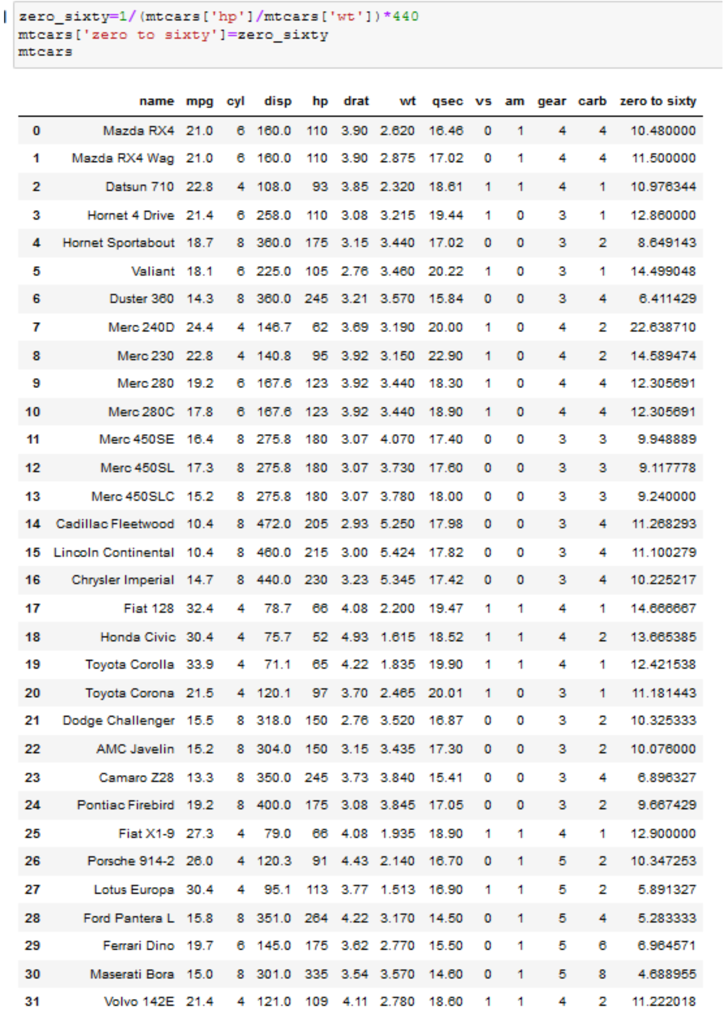
Task 3

Using the mtcars data from week 4, load mtcars. An approximation of the 0-60 time of a car can be calculated through the formula:

$$\left({}^{1}/_{(Horsepower\ /\ Weight)}\right)\cdot 440$$

Task 3.1 Evidence your new vector

Using this, create a new vector of 0-60 times for the cars, and then attach this to the dataset. Evidence your code for creating a new vector of 0-60 times

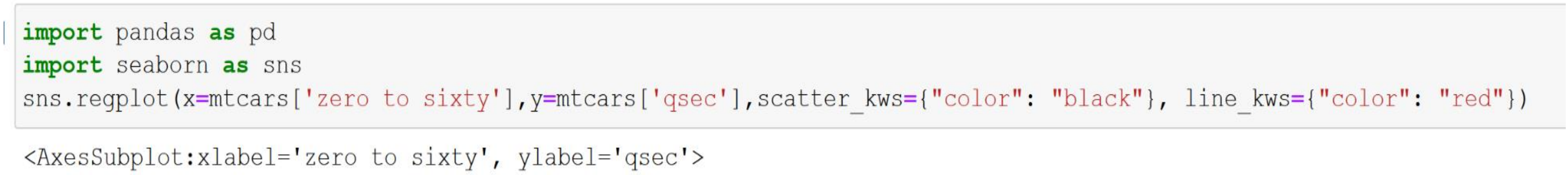


```
zero_sixty=1/(mtcars['hp']/mtcars['wt'])*440
mtcars['zero to sixty']=zero_sixty
mtcars
```

	name	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	zero to sixty
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4	10.480000
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4	11.500000
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1	10.976344
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1	12.860000
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2	8.649143
5	Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1	14.499048
6	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4	6.411429
7	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2	22.638710
8	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2	14.589474
9	Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4	12.305691
10	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4	12.305691
11	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3	9.948889
12	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3	9.117778
13	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3	9.240000
14	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4	11.268293
15	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4	11.100279
16	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4	10.225217
17	Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1	14.666667
18	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2	13.665385
19	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1	12.421538
20	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1	11.181443
21	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2	10.325333
22	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2	10.076000
23	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4	6.896327
24	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2	9.667429
25	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1	12.900000
26	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2	10.347253
27	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2	5.891327
28	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4	5.283333
29	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6	6.964571
30	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8	4.688955
31	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2	11.222018

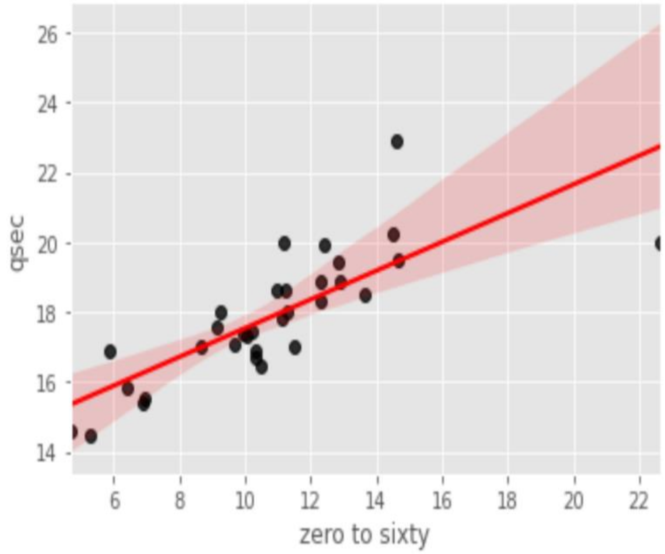
Task 3.2 Evidence your scatter plot

Create a scatter plot showing the relationship between the 0-60 time and the quarter-mile time. Illustrate this relationship with a linear regression line, coloured red. What does this tell us about the relationship between these features? Paste evidence of the scatter plot showing the relationship between the 0-60 time and the quarter-mile time.



```
import pandas as pd
import seaborn as sns
sns.regplot(x=mtcars['zero to sixty'],y=mtcars['qsec'],scatter_kws={"color": "black"}, line_kws={"color": "red"})
```

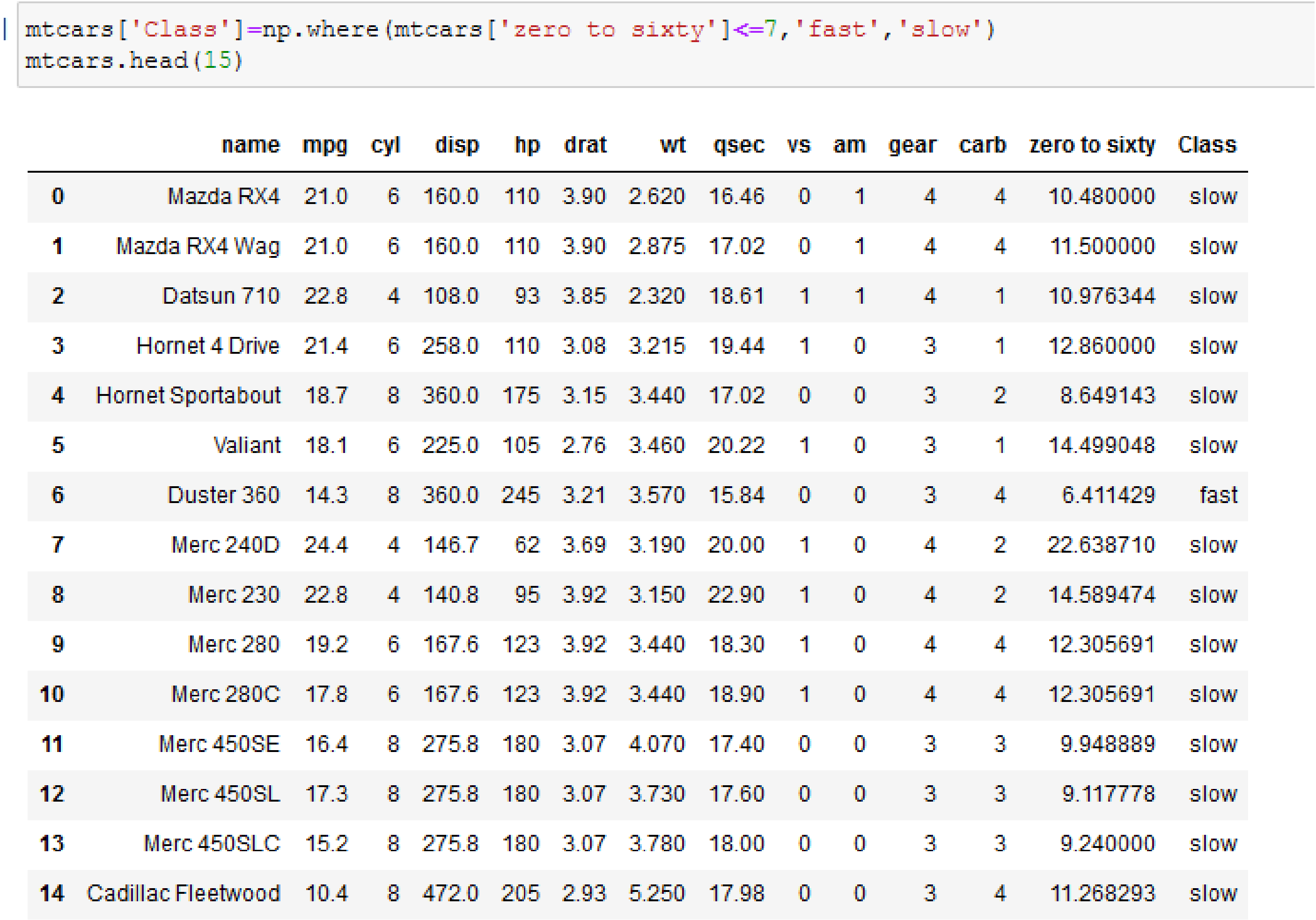
```
<AxesSubplot:xlabel='zero to sixty', ylabel='qsec'>
```



As the graphic shows the the quarter-mile time (qsec) and 0-60 time are positively associated, therefore a higher 0-60 time implies a higher quarter-mile time too.

Task 3.3 document your new variable

Create a new variable showing whether a car is classed as “fast” or “slow”. A fast car has a 0-60 of less than 7 seconds, otherwise it is classed as slow. Provide evidence of the code you created.



```
mtcars['Class']=np.where(mtcars['zero to sixty']<=7,'fast','slow')
mtcars.head(15)
```

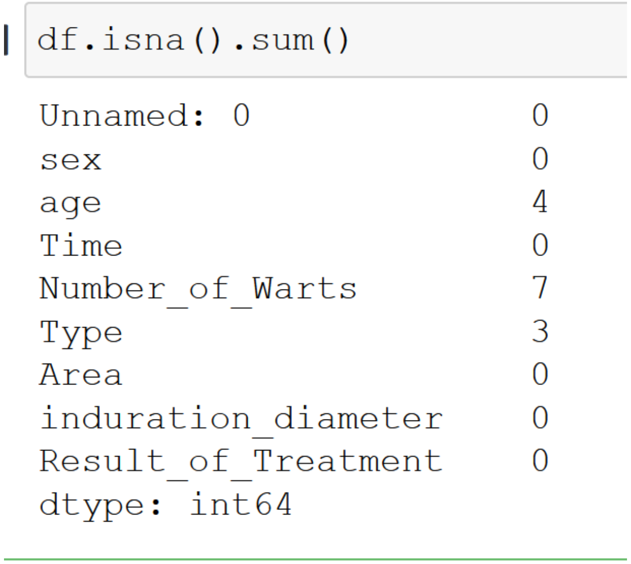
	name	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	zero to sixty	Class
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4	10.480000	slow
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4	11.500000	slow
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1	10.976344	slow
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1	12.860000	slow
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2	8.649143	slow
5	Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1	14.499048	slow
6	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4	6.411429	fast
7	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2	22.638710	slow
8	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2	14.589474	slow
9	Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4	12.305691	slow
10	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4	12.305691	slow
11	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3	9.948889	slow
12	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3	9.117778	slow
13	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3	9.240000	slow
14	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4	11.268293	slow

Task 4 Features with missing data

Document your answers in relation to which features within the immunotherapy.csv dataset

a. What features within the dataset have missing data?

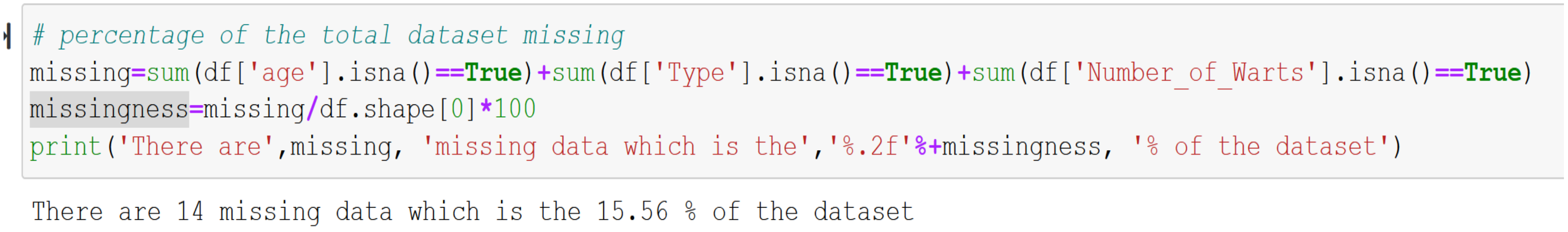
The columns below have missing data



```
df.isna().sum()
```

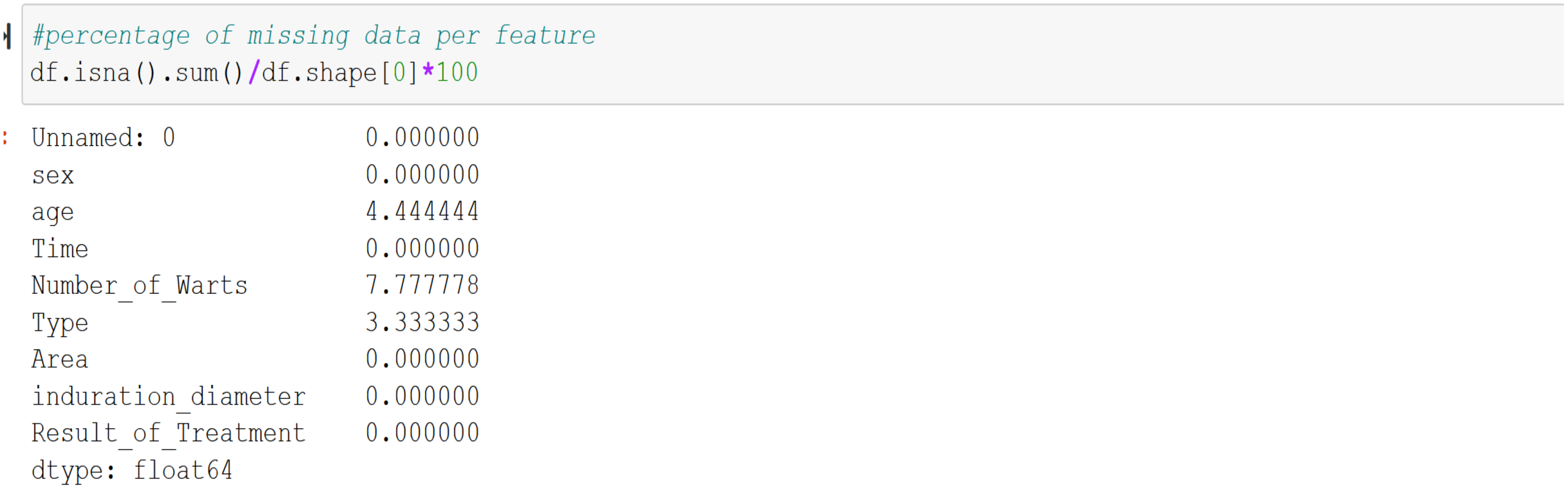
```
Unnamed: 0             0
sex                    0
age                    4
Time                   0
Number_of_Warts        7
Type                   3
Area                   0
induration_diameter    0
Result_of_Treatment    0
dtype: int64
```

b. What percentage of the total dataset is missing. What percentage of data is missing for the features identified in Part A.



```
# percentage of the total dataset missing
missing=sum(df['age'].isna()==True)+sum(df['Type'].isna()==True)+sum(df['Number_of_Warts'].isna()==True)
missingness=missing/df.shape[0]*100
print('There are',missing, 'missing data which is the','%.2f'%+missingness, '% of the dataset')
```

```
There are 14 missing data which is the 15.56 % of the dataset
```



```
#percentage of missing data per feature
df.isna().sum()/df.shape[0]*100
```

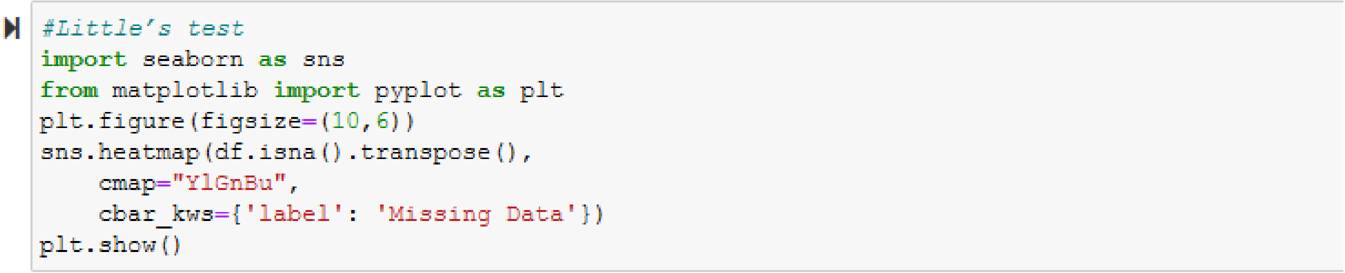
```
Unnamed: 0             0.000000
sex                    0.000000
age                    4.444444
Time                   0.000000
Number_of_Warts        7.777778
Type                   3.333333
Area                   0.000000
induration_diameter    0.000000
Result_of_Treatment    0.000000
dtype: float64
```

Task 5 Method for replacing missing values

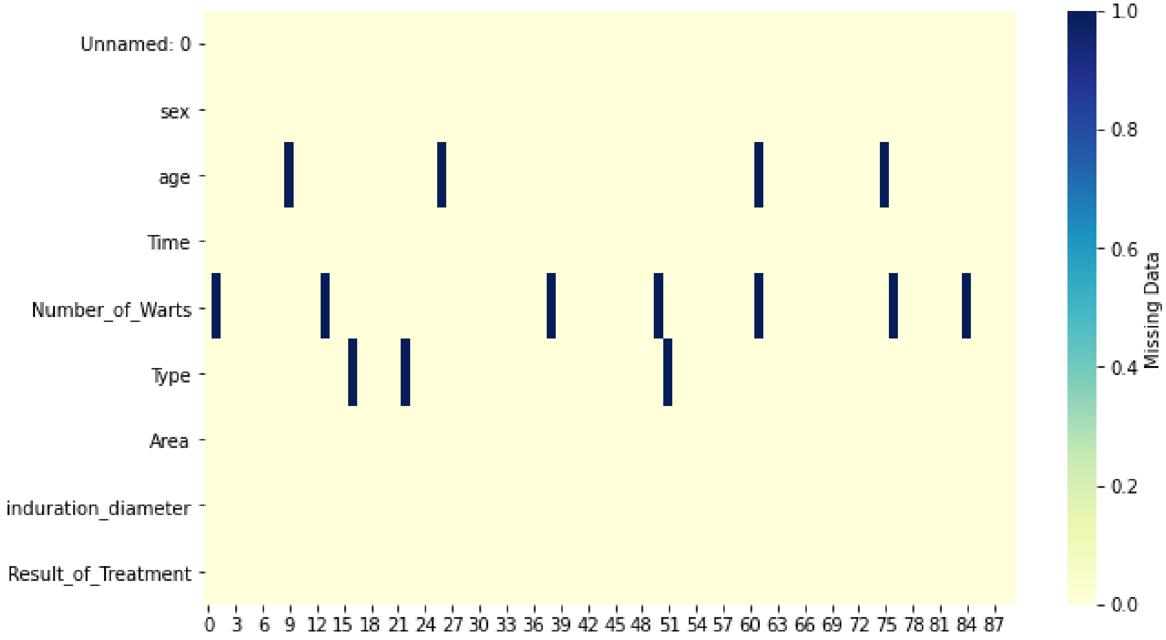
Describe the chosen method for replacing missing values in each of the features and evidence your use of this method to replace the values so that each feature has complete data.

Method

1. Apply the little test to find out if the data is missing completely at ramdom



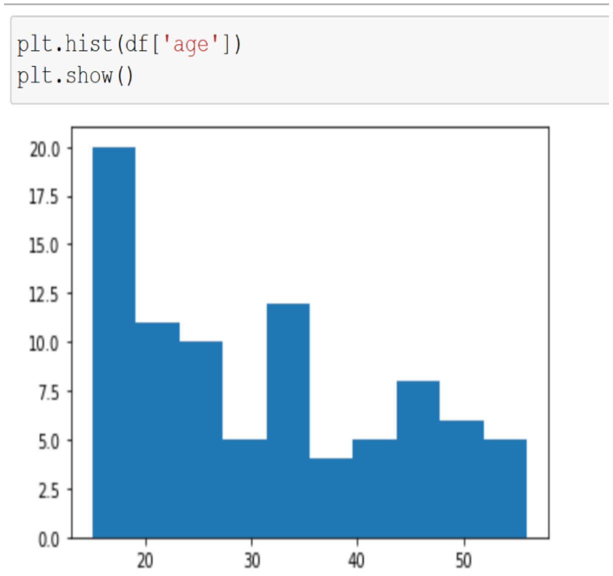
```
#Little’s test
import seaborn as sns
from matplotlib import pyplot as plt
plt.figure(figsize=(10,6))
sns.heatmap(df.isna().transpose(),
    cmap="YlGnBu",
    cbar_kws={'label': 'Missing Data'})
plt.show()
```



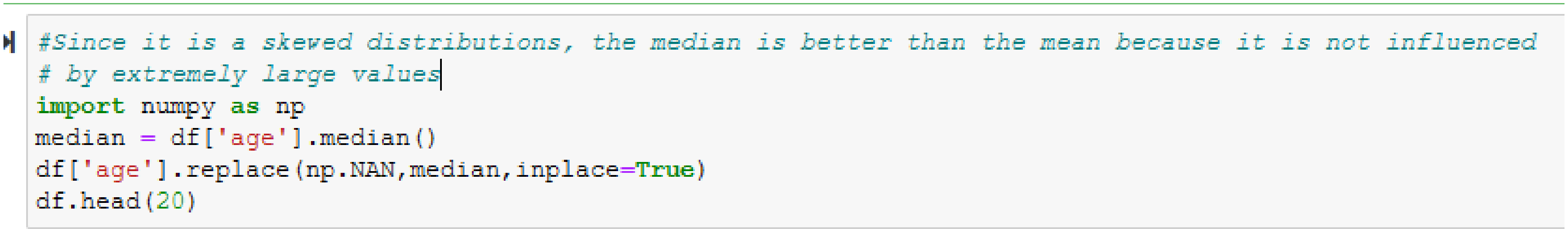
the heatmap shows that the values have a p.value>0.05 therefore the data can be treated as MCAR

2. Treating data per column: Analize the distribution, find out what method of imputation can be use and replace the values

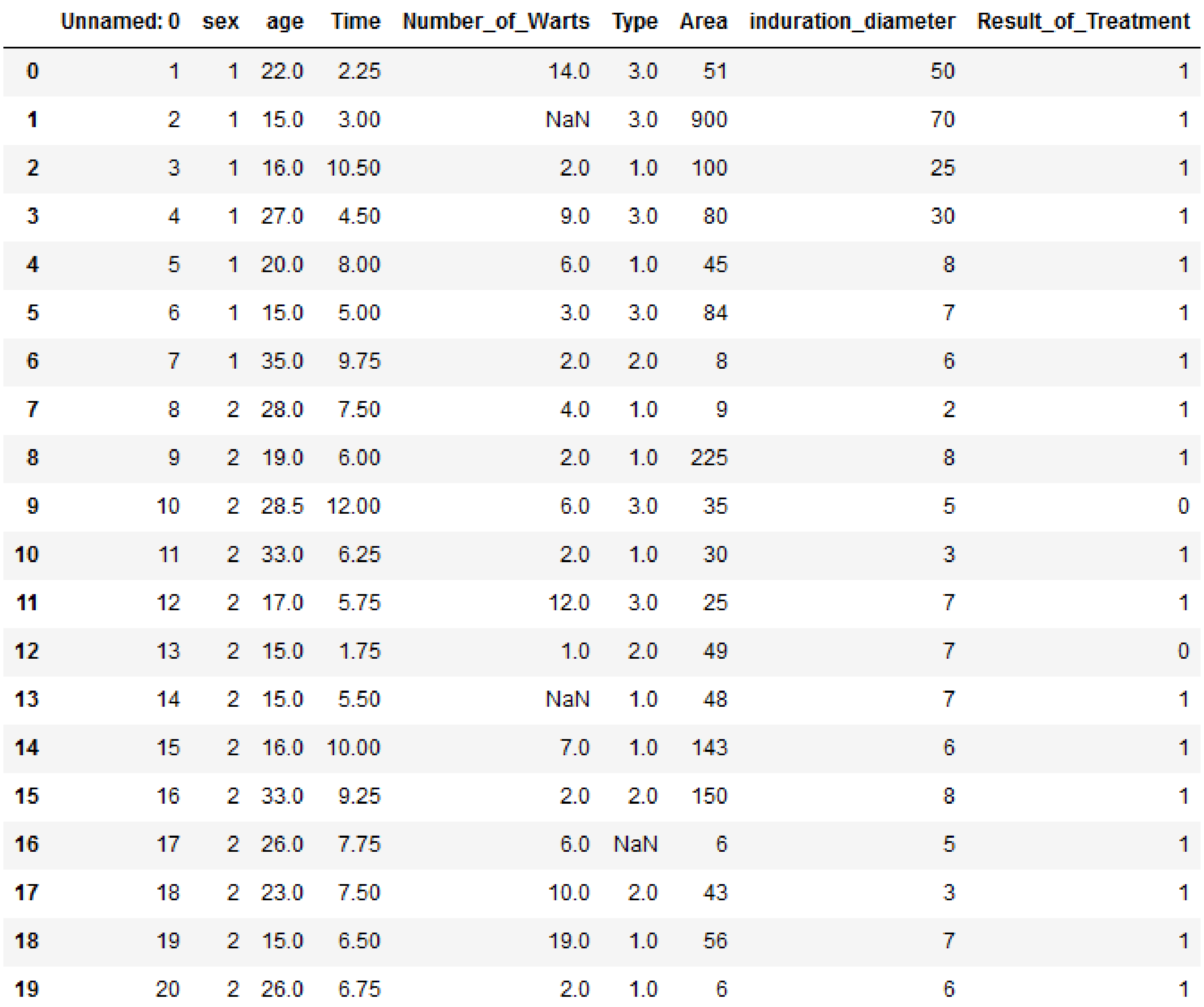
Age



```
plt.hist(df['age'])
plt.show()
```

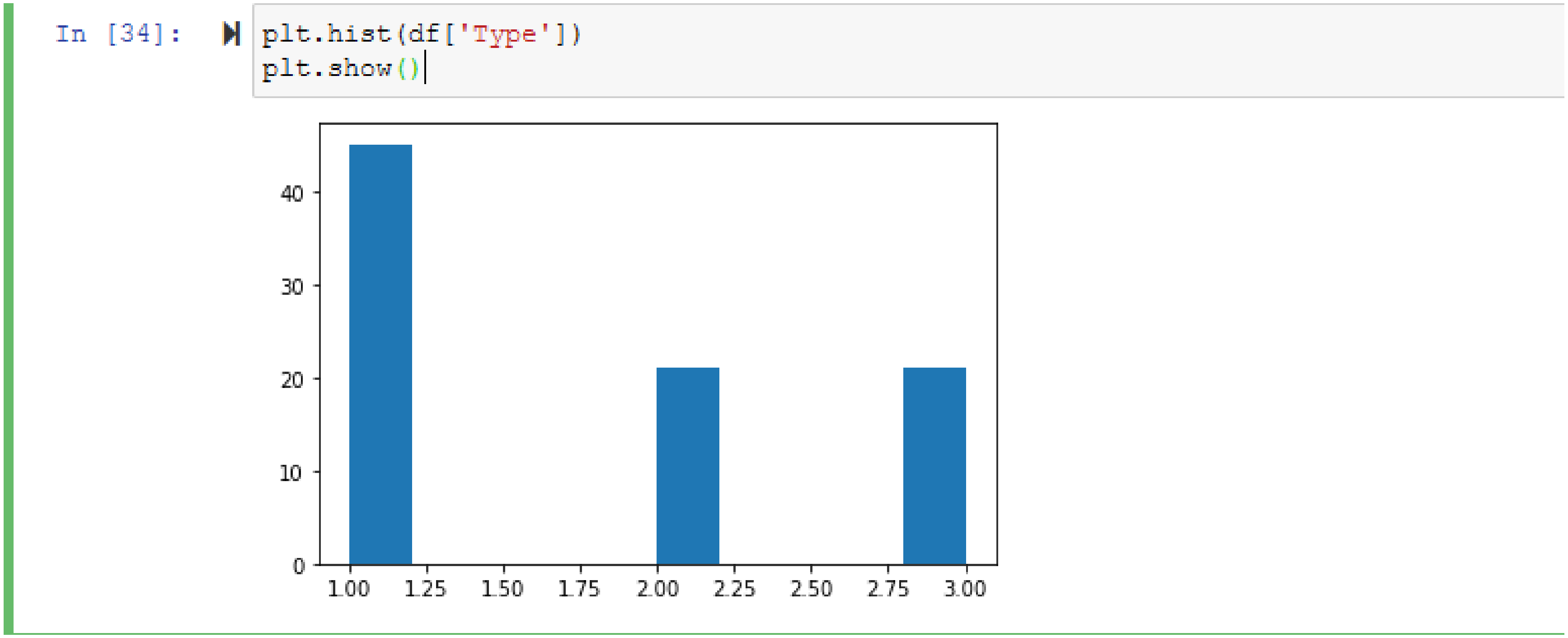


```
#Since it is a skewed distributions, the median is better than the mean because it is not influenced
# by extremely large values
import numpy as np
median = df['age'].median()
df['age'].replace(np.NAN,median,inplace=True)
df.head(20)
```

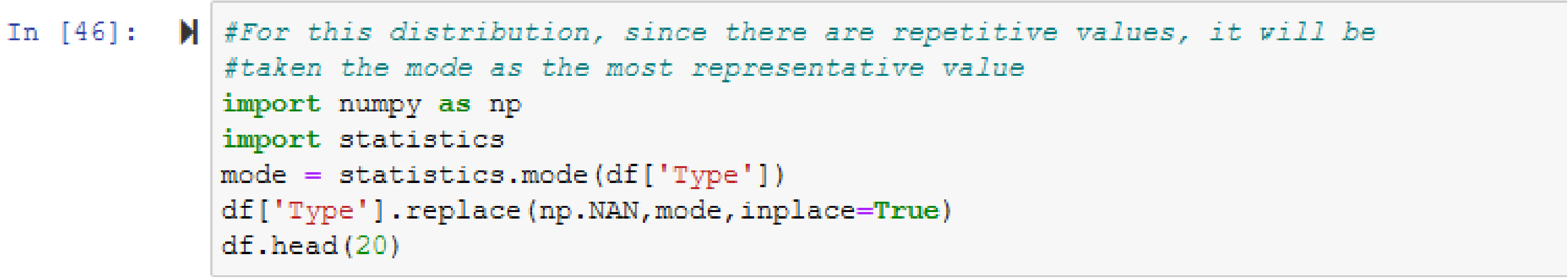


	Unnamed: 0	sex	age	Time	Number_of_Warts	Type	Area	induration_diameter	Result_of_Treatment
0	1	1	22.0	2.25	14.0	3.0	51	50	1
1	2	1	15.0	3.00	NaN	3.0	900	70	1
2	3	1	16.0	10.50	2.0	1.0	100	25	1
3	4	1	27.0	4.50	9.0	3.0	80	30	1
4	5	1	20.0	8.00	6.0	1.0	45	8	1
5	6	1	15.0	5.00	3.0	3.0	84	7	1
6	7	1	35.0	9.75	2.0	2.0	8	6	1
7	8	2	28.0	7.50	4.0	1.0	9	2	1
8	9	2	19.0	6.00	2.0	1.0	225	8	1
9	10	2	28.5	12.00	6.0	3.0	35	5	0
10	11	2	33.0	6.25	2.0	1.0	30	3	1
11	12	2	17.0	5.75	12.0	3.0	25	7	1
12	13	2	15.0	1.75	1.0	2.0	49	7	0
13	14	2	15.0	5.50	NaN	1.0	48	7	1
14	15	2	16.0	10.00	7.0	1.0	143	6	1
15	16	2	33.0	9.25	2.0	2.0	150	8	1
16	17	2	26.0	7.75	6.0	NaN	6	5	1
17	18	2	23.0	7.50	10.0	2.0	43	3	1
18	19	2	15.0	6.50	19.0	1.0	56	7	1
19	20	2	26.0	6.75	2.0	1.0	6	6	1

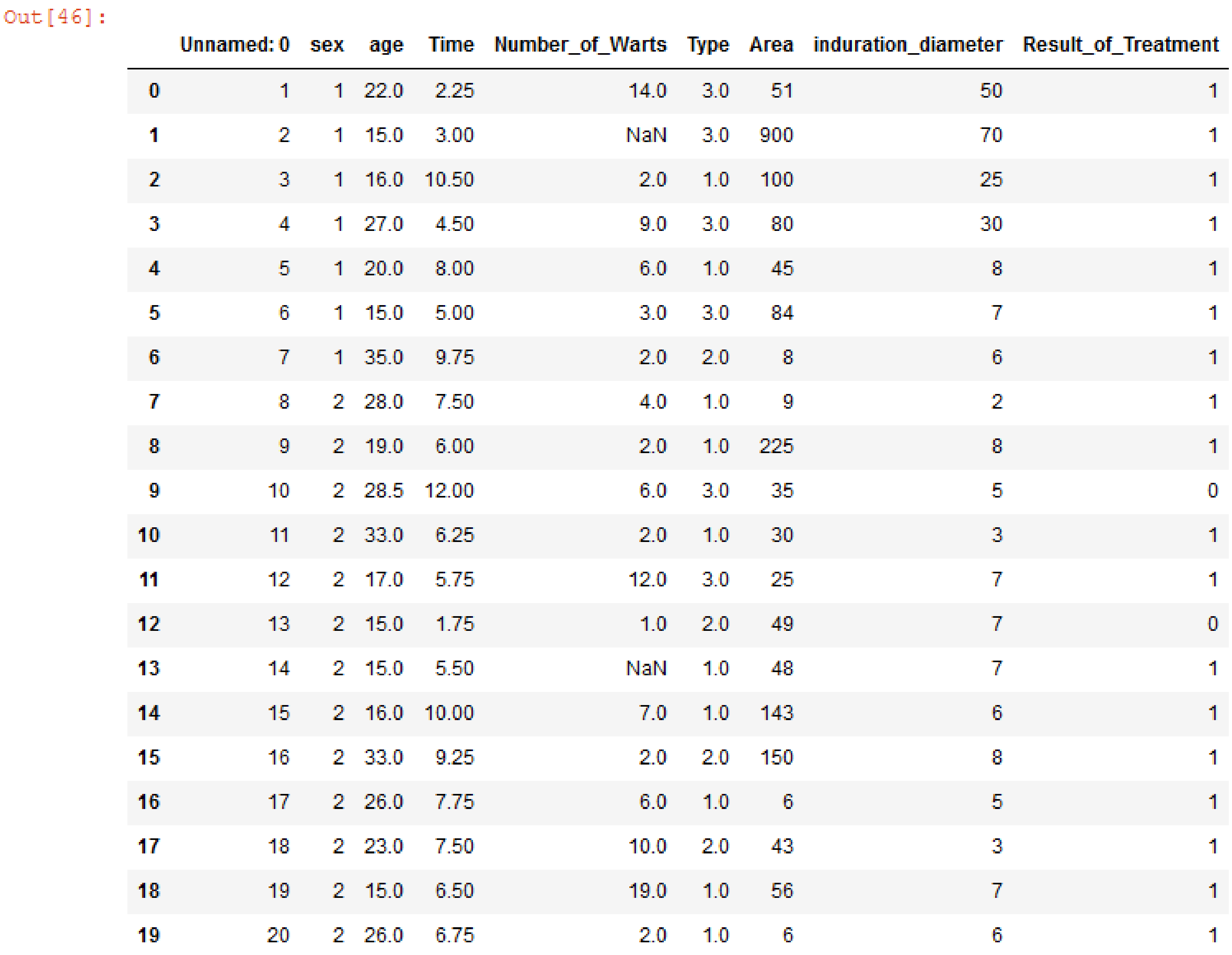
Type



```
In [34]:  plt.hist(df['Type'])
          plt.show()
```



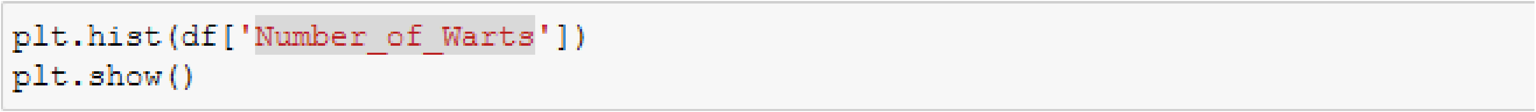
```
In [46]:  #For this distribution, since there are repetitive values, it will be
          #taken the mode as the most representative value
          import numpy as np
          import statistics
          mode = statistics.mode(df['Type'])
          df['Type'].replace(np.NAN,mode,inplace=True)
          df.head(20)
```



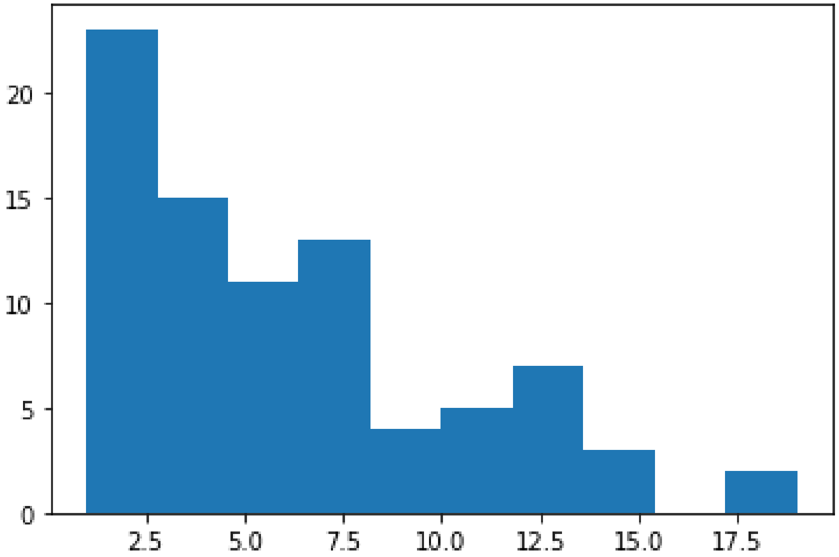
Out[46]:

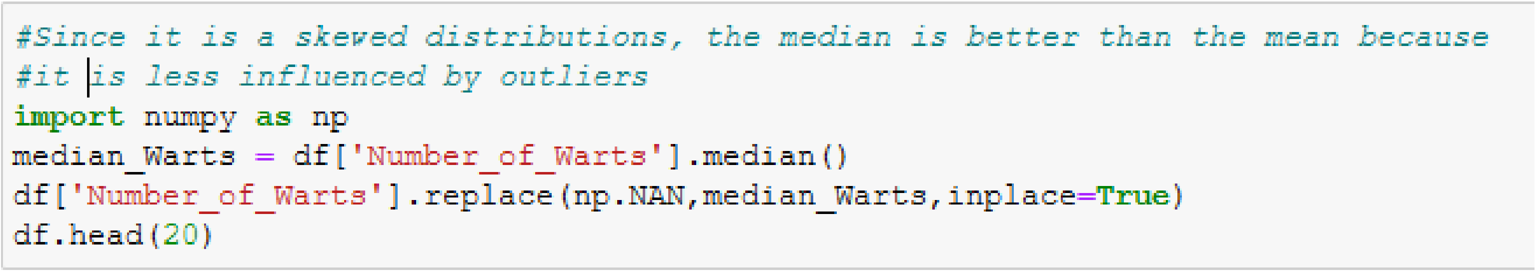
	Unnamed: 0	sex	age	Time	Number_of_Warts	Type	Area	induration_diameter	Result_of_Treatment
0	1	1	22.0	2.25	14.0	3.0	51	50	1
1	2	1	15.0	3.00	NaN	3.0	900	70	1
2	3	1	16.0	10.50	2.0	1.0	100	25	1
3	4	1	27.0	4.50	9.0	3.0	80	30	1
4	5	1	20.0	8.00	6.0	1.0	45	8	1
5	6	1	15.0	5.00	3.0	3.0	84	7	1
6	7	1	35.0	9.75	2.0	2.0	8	6	1
7	8	2	28.0	7.50	4.0	1.0	9	2	1
8	9	2	19.0	6.00	2.0	1.0	225	8	1
9	10	2	28.5	12.00	6.0	3.0	35	5	0
10	11	2	33.0	6.25	2.0	1.0	30	3	1
11	12	2	17.0	5.75	12.0	3.0	25	7	1
12	13	2	15.0	1.75	1.0	2.0	49	7	0
13	14	2	15.0	5.50	NaN	1.0	48	7	1
14	15	2	16.0	10.00	7.0	1.0	143	6	1
15	16	2	33.0	9.25	2.0	2.0	150	8	1
16	17	2	26.0	7.75	6.0	1.0	6	5	1
17	18	2	23.0	7.50	10.0	2.0	43	3	1
18	19	2	15.0	6.50	19.0	1.0	56	7	1
19	20	2	26.0	6.75	2.0	1.0	6	6	1

Number of Warts

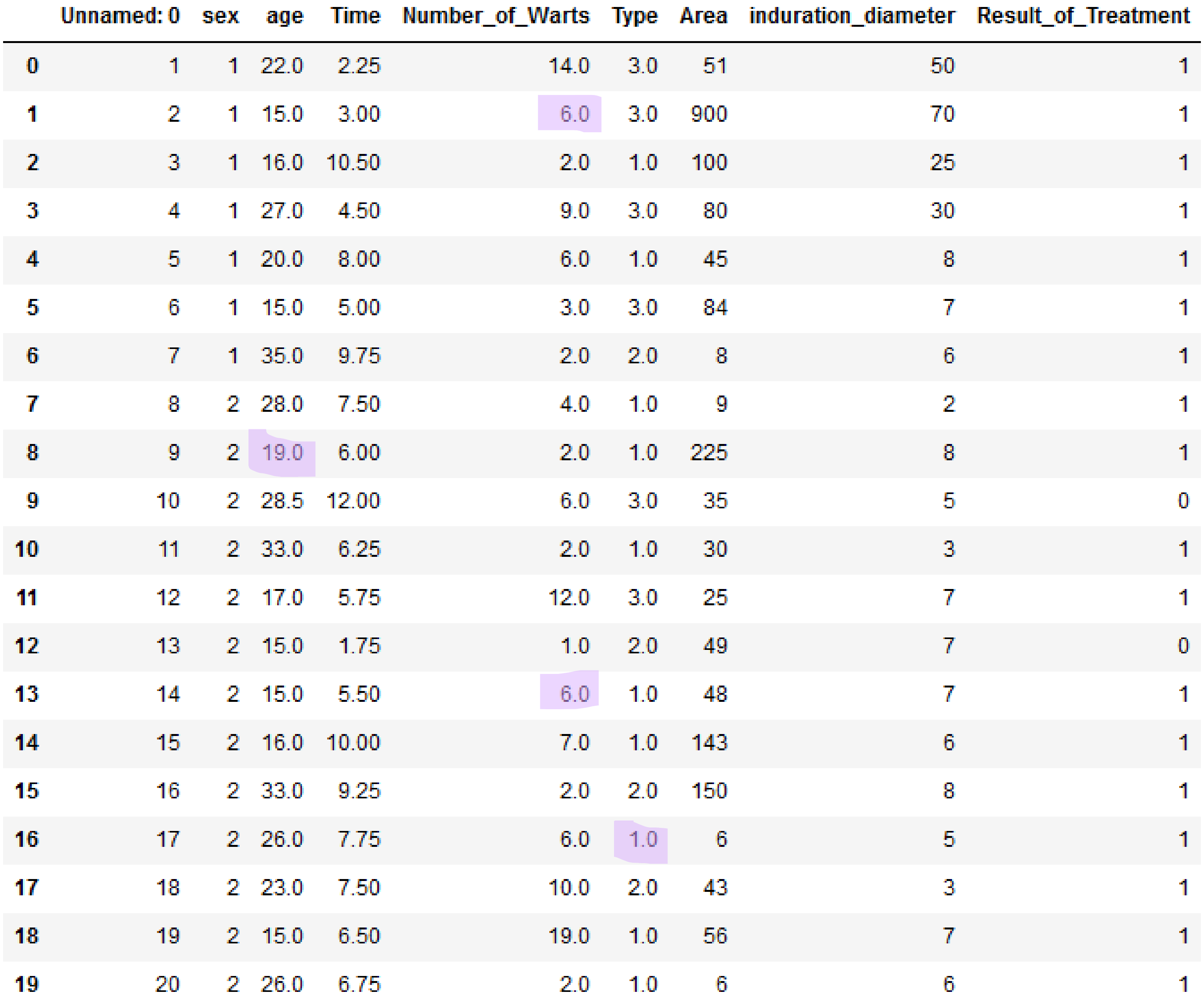


```
plt.hist(df['Number_of_Warts'])
plt.show()
```





```
#Since it is a skewed distributions, the median is better than the mean because
#it is less influenced by outliers
import numpy as np
median_Warts = df['Number_of_Warts'].median()
df['Number_of_Warts'].replace(np.NAN,median_Warts,inplace=True)
df.head(20)
```



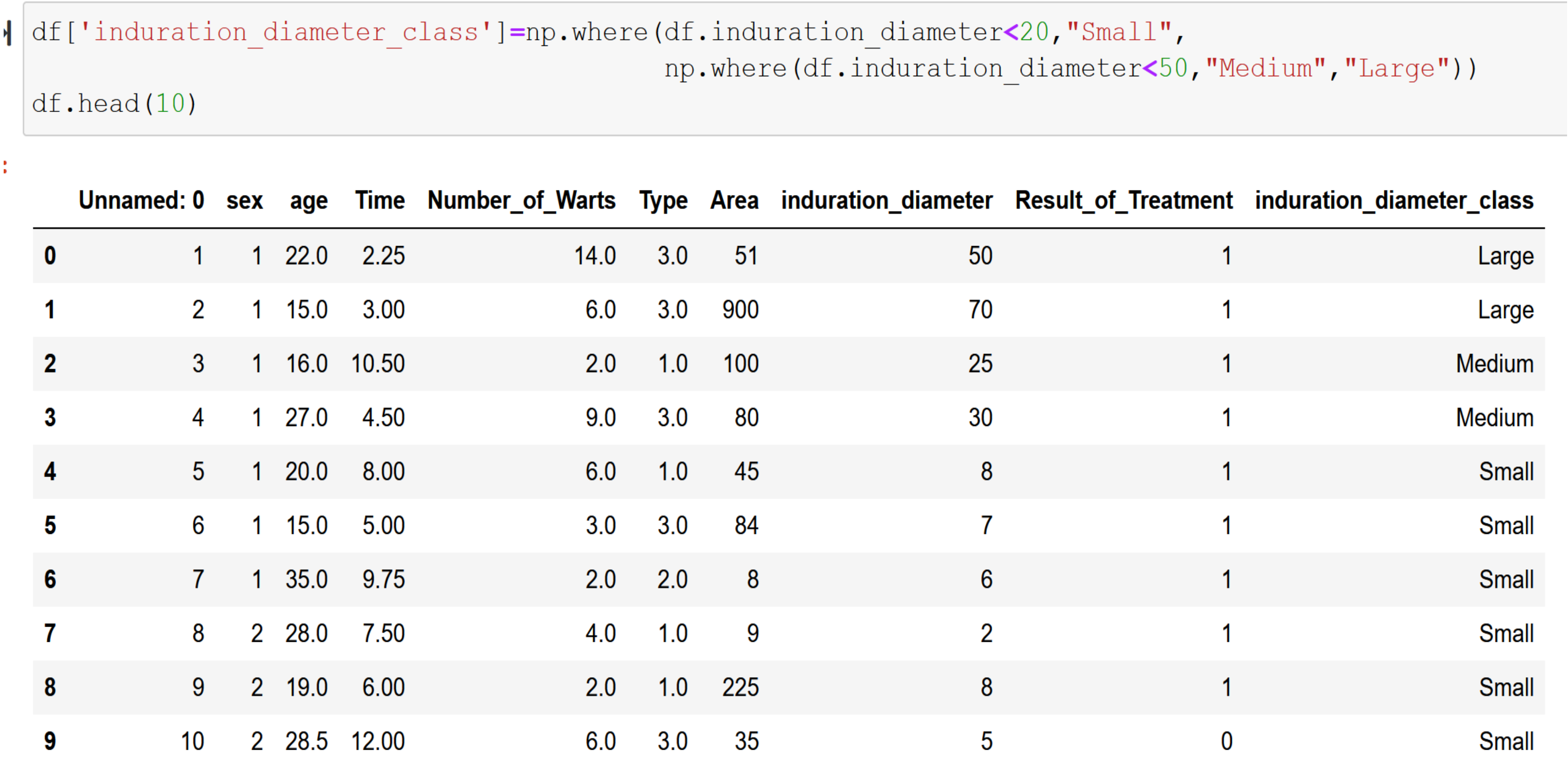
	Unnamed: 0	sex	age	Time	Number_of_Warts	Type	Area	induration_diameter	Result_of_Treatment
0	1	1	22.0	2.25	14.0	3.0	51	50	1
1	2	1	15.0	3.00	6.0	3.0	900	70	1
2	3	1	16.0	10.50	2.0	1.0	100	25	1
3	4	1	27.0	4.50	9.0	3.0	80	30	1
4	5	1	20.0	8.00	6.0	1.0	45	8	1
5	6	1	15.0	5.00	3.0	3.0	84	7	1
6	7	1	35.0	9.75	2.0	2.0	8	6	1
7	8	2	28.0	7.50	4.0	1.0	9	2	1
8	9	2	19.0	6.00	2.0	1.0	225	8	1
9	10	2	28.5	12.00	6.0	3.0	35	5	0
10	11	2	33.0	6.25	2.0	1.0	30	3	1
11	12	2	17.0	5.75	12.0	3.0	25	7	1
12	13	2	15.0	1.75	1.0	2.0	49	7	0
13	14	2	15.0	5.50	6.0	1.0	48	7	1
14	15	2	16.0	10.00	7.0	1.0	143	6	1
15	16	2	33.0	9.25	2.0	2.0	150	8	1
16	17	2	26.0	7.75	6.0	1.0	6	5	1
17	18	2	23.0	7.50	10.0	2.0	43	3	1
18	19	2	15.0	6.50	19.0	1.0	56	7	1
19	20	2	26.0	6.75	2.0	1.0	6	6	1

Task 6 Document your new feature

To make the identification of potentially troublesome issues for patients, you have been requested to create a new feature recording the induration diameter in a more straightforward way

Document your code for computing a new feature, with the induration diameter coded as Small, Medium and Large.

- Small, when the diameter is less than 20.
- Medium, when the diameter ranges from 20 to 50.
- Large, when the diameter is greater than 50.



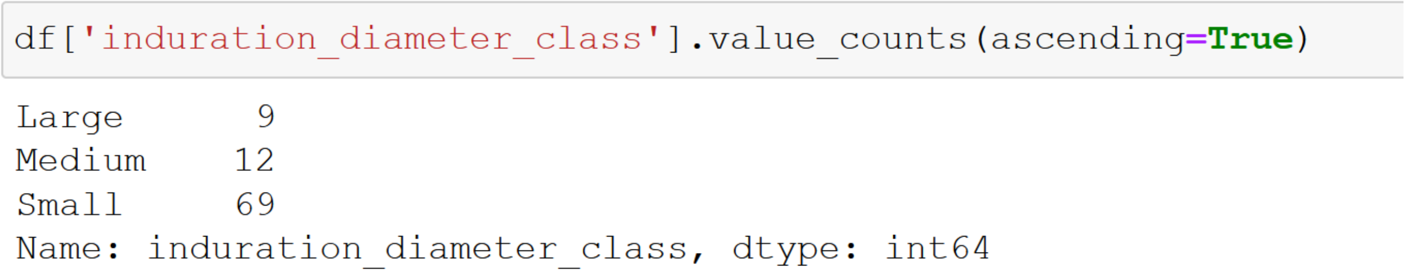
```
df['induration_diameter_class']=np.where(df.induration_diameter<20,"Small",
                                         np.where(df.induration_diameter<50,"Medium","Large"))
df.head(10)
```

	Unnamed: 0	sex	age	Time	Number_of_Warts	Type	Area	induration_diameter	Result_of_Treatment	induration_diameter_class
0	1	1	22.0	2.25	14.0	3.0	51	50	1	Large
1	2	1	15.0	3.00	6.0	3.0	900	70	1	Large
2	3	1	16.0	10.50	2.0	1.0	100	25	1	Medium
3	4	1	27.0	4.50	9.0	3.0	80	30	1	Medium
4	5	1	20.0	8.00	6.0	1.0	45	8	1	Small
5	6	1	15.0	5.00	3.0	3.0	84	7	1	Small
6	7	1	35.0	9.75	2.0	2.0	8	6	1	Small
7	8	2	28.0	7.50	4.0	1.0	9	2	1	Small
8	9	2	19.0	6.00	2.0	1.0	225	8	1	Small
9	10	2	28.5	12.00	6.0	3.0	35	5	0	Small

Task 7 Document the size of induration

Based upon this new representation of the data, which size of induration appears most frequently?

The most frequent value is "small" with 69 appearances.



```
df['induration_diameter_class'].value_counts(ascending=True)
```

```
Large       9
Medium     12
Small      69
Name: induration_diameter_class, dtype: int64
```

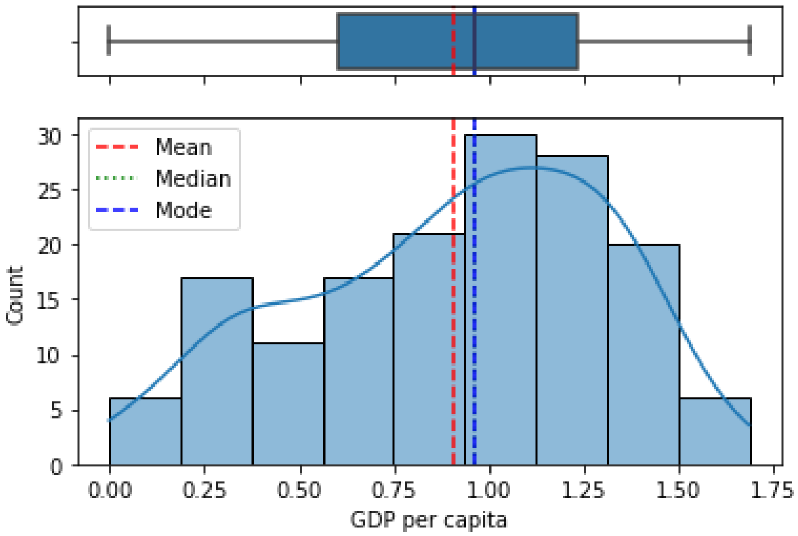
Part 4 – Exploratory Data Analysis

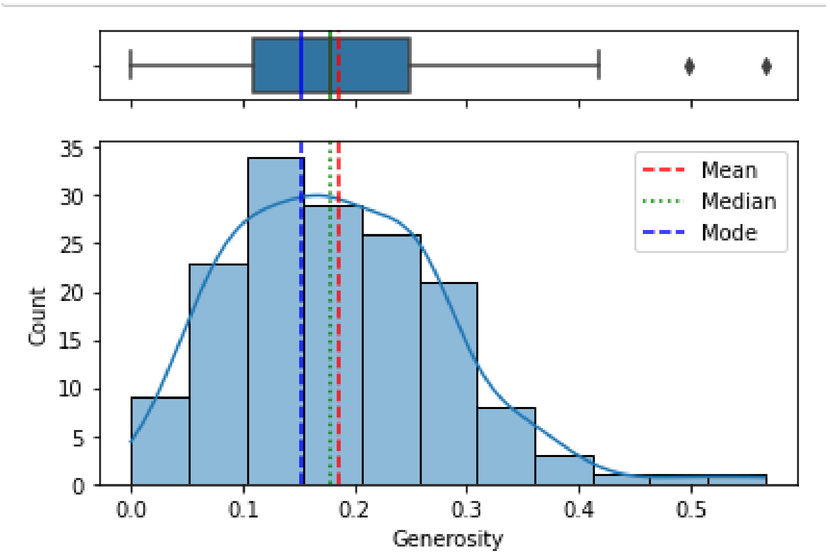
For the tasks below, happiness.csv is provided, and load it as Pandas dataframe. Remove the feature containing the country names.

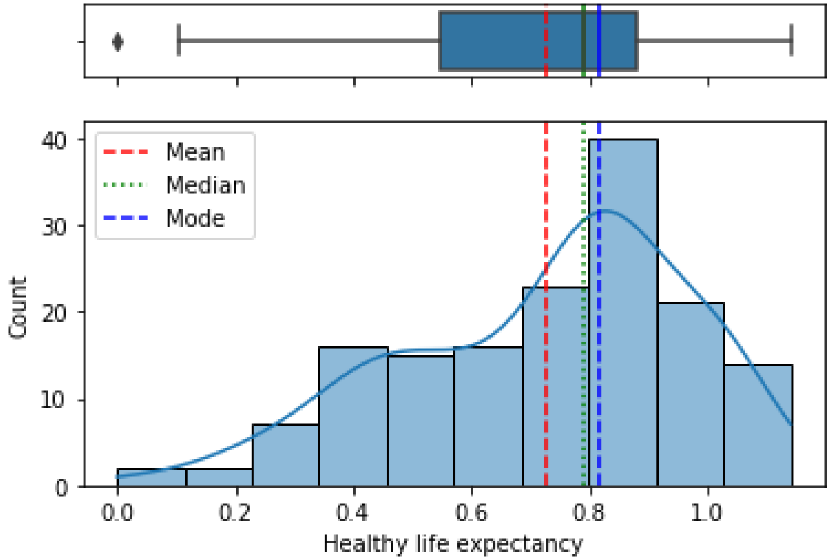
Task 1 Evidence table of findings

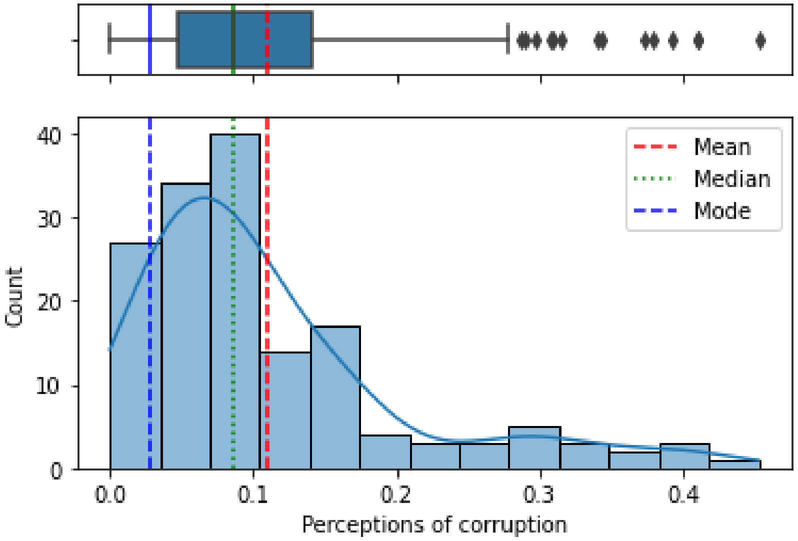
Perform some univariate EDA on the features within the dataset. Complete the table below

Feature	Skewness	Kurtosis	Appropriate measure of central tendency MoCT	MoCT Value	Outliers?
GDP per capita	Negative (-0.385)	Negative (-0.770)	Median	0.960	No
Generosity	Positive (0.746)	Positive (1.173)	Median	0.178	Yes
Healthy life expectancy	Negative (-0.614)	Negative (-0.303)	Median	0.789	Yes
Perceptions of corruption	Positive (1.650)	Neutral (2.412*) *Value close to 3	Median	0.086	Yes


Mean
Median
Mode
Count
0
5
10
15
20
25
30
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
GDP per capita

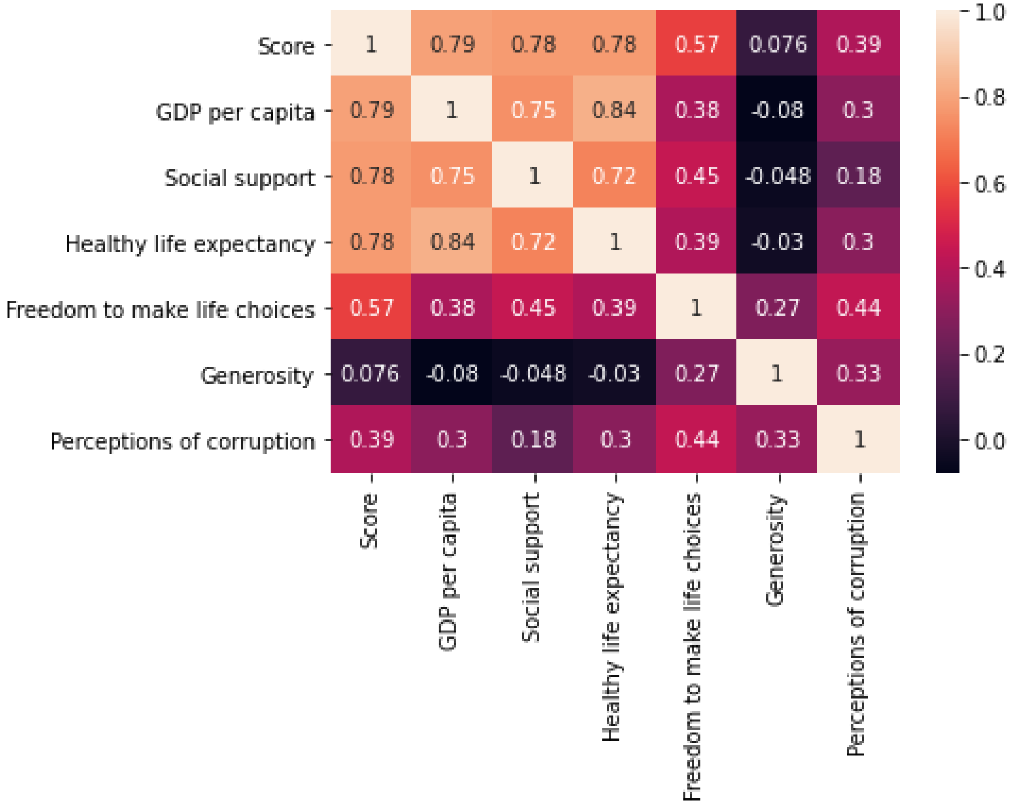

Mean
Median
Mode
Count
0
5
10
15
20
25
30
35
0.0
0.1
0.2
0.3
0.4
0.5
Generosity


Mean
Median
Mode
Count
0
10
20
30
40
0.0
0.2
0.4
0.6
0.8
1.0
Healthy life expectancy


Mean
Median
Mode
Count
0
10
20
30
40
0.0
0.1
0.2
0.3
0.4
Perceptions of corruption

Task 2 Provide correlation chart

Produce a correlation chart for the dataset. When developing a model to predict the overall satisfaction score:



2.1. Which features would act as the strongest predictor?

The strongest predictors are the ones that have the strongest correlation, meaning that the Pearson correlation coefficient is greater than |±0.6|. The features that have correlations with a score greater than 0.6 are:

- GDP per capita (0. 79)
- Social support (0.78)
- Healthy life expectancy (0.78)

2.2. Which features may you wish to remove from the dataset, and why?

GDP per capita or Healthy life expectancy because these features are strongly correlated (correlation>0.8). Therefore, if both features are used, there will be issues with multicollinearity.

Task 3 Document how techniques can help to satisfy assumptions

3. Document how the techniques covered thus far can help to satisfy:

3.1. The assumption of feature independence

The independence of a feature can be identified using the Pearson Correlation feature. As it is displayed in the chart, values higher than 0.8 may indicate a strong correlation and therefore this could signify that the features are dependent on each other.

The features that have a high degree of interdependency for this dataset are GDP per capita and healthy life expectancy. Thus, by using the Pearson Correlation technique we can identify that the other features can be assumed as independent.

3.2. The assumption of observational independence

There is observational independence when 'any data point in a set of data is statistically independent from the rest.' This means that its value is not influenced by the value of any other observation in the set, thus the observations are not correlated.

A way to identify observational independence is by plotting the distribution of each feature. If is seen variation between the data, then it can be assumed observational independence. On the contrary, if the distribution is skewed, this could signify that the outliers are somehow correlated.

3.3. The assumption of the approximation of normality

The technique that helps to identify how close to an approximation of normality is a quantile-normal plot. By plotting a line that represents a normal distribution and the feature analyzed, it is identified any deviation that shows non-normality.

3.4. The assumption of accurate data

To determine if the data is accurate, the technique to use is to plot the distribution of each feature. If outliers or unusual data are found, and there is no cause for them to exist then it could be thought that the data might have some errors, therefore is not accurate.

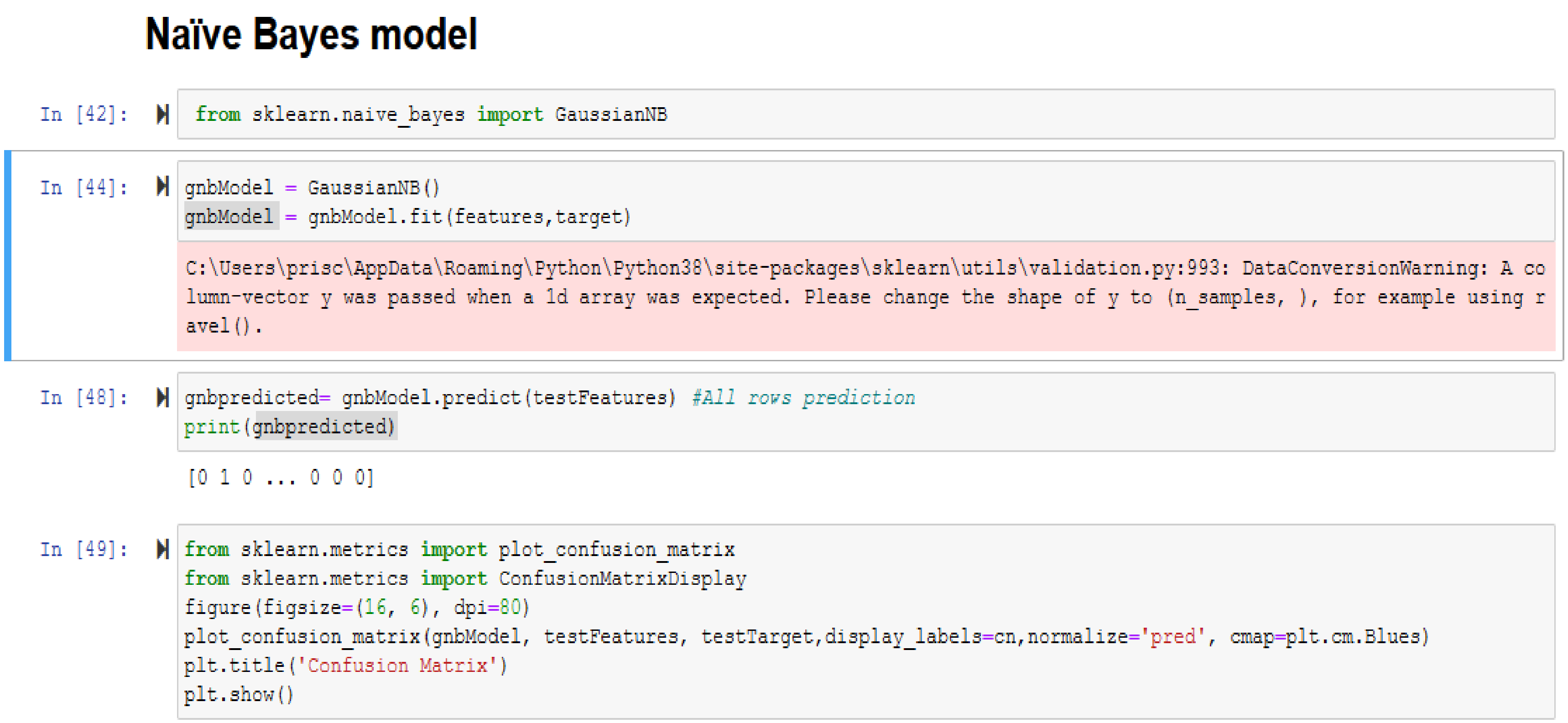
Part 5 - Classification and Clustering

The evidence required for these tasks relates to **Week 9** in which you were using a variety of clustering and classification techniques. Note only the work developed in the **Tasks** section of **Chapter 5 – Tasks Classification Models** is required to be evidenced here.

Task 1 Predictive accuracy via a Naïve Bayes model

Try to build and test Naïve Bayes model yourself. Evidence the predictive accuracy you obtained through the creation of the Naïve Bayes model you created for **Task 1**

The accuracy is 0.753



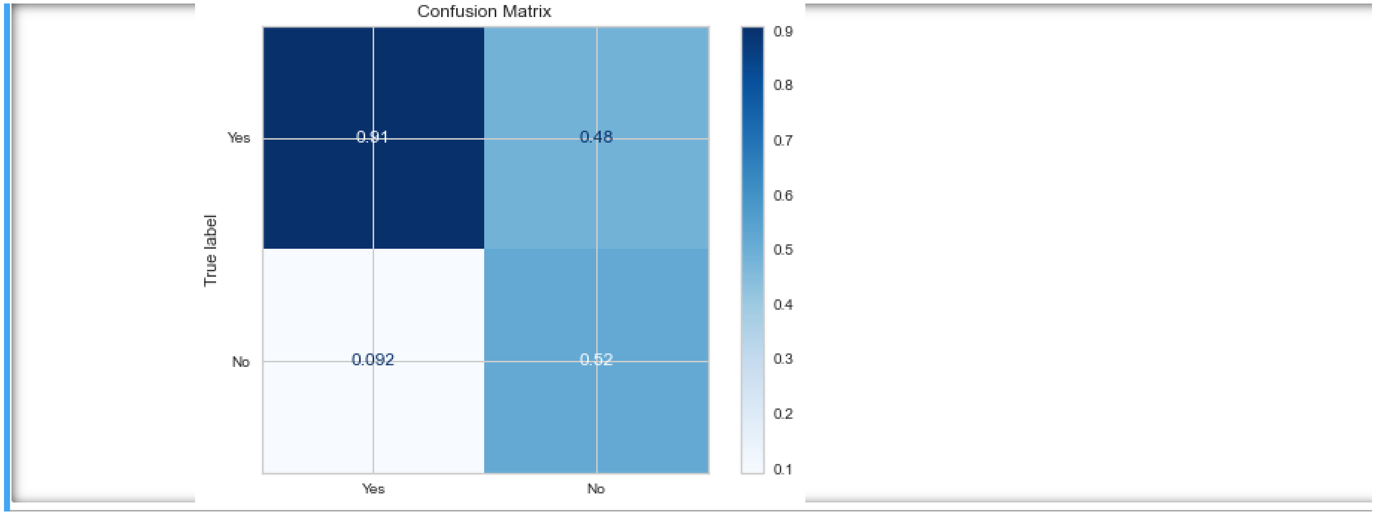
Naïve Bayes model

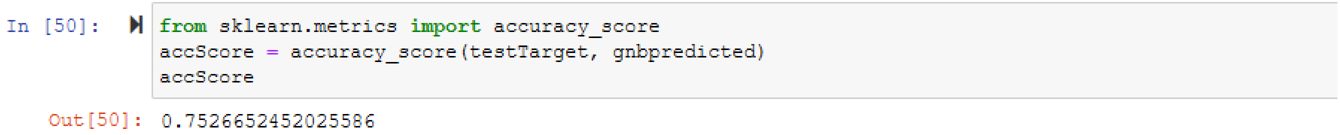
```
In [42]:  from sklearn.naive_bayes import GaussianNB
```

```
In [44]:  gnbModel = GaussianNB()
           gnbModel = gnbModel.fit(features,target)

           C:\Users\prisc\AppData\Roaming\Python\Python38\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A co
           lumn-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using r
           avel().
```

```
In [48]:  gnbpredicted= gnbModel.predict(testFeatures) #All rows prediction
           print(gnbpredicted)

           [0 1 0 ... 0 0 0]
```

```
In [49]:  from sklearn.metrics import plot_confusion_matrix
           from sklearn.metrics import ConfusionMatrixDisplay
           figure(figsize=(16, 6), dpi=80)
           plot_confusion_matrix(gnbModel, testFeatures, testTarget,display_labels=cn,normalize='pred', cmap=plt.cm.Blues)
           plt.title('Confusion Matrix')
           plt.show()
```





```
In [50]:  from sklearn.metrics import accuracy_score
          accScore = accuracy_score(testTarget, gnbpredicted)
          accScore

Out[50]:  0.7526652452025586
```

Task 2 Evidence the random forest table which identifies best performance

The process of tuning the parameters of a model to attempt to maximise the performance is called parameter optimisation. Complete the table below, tuning the parameters for the random forest with values of your choosing and recording the results (the first row represents the example completed in the exercise). Which combination performs the best?

Attempt Number	Max_features	n_estimators	Accuracy%
1	5	500	0.8017057569296375
2	5	1000	0.8024164889836531
3	5	200	0.7974413646055437
4	15	500	0.7967306325515281
5	3	500	0.783226723525231

The best performance corresponds to attempt 2, where Max_features= 5, n_estimators=1000 and the accuracy is 0.802.

Task 3 Evidence the kernel table which identifies best performance

One of the key parameters that can be tuned when creating an SVM model is the choice of kernel that is used. There are three forms of kernel that can be used: rbf, linear, poly and sigmoid. For example, to use a linear kernel, the following command would be used:

SvmModel = svm.SVC(kernel='rbf',gamma='scale')

Complete the table below, along with the example from the exercise. Which kernel performs the best?

Attempt Number	**Kernel**	**Accuracy (%)**
1	rbf kernel	0.7093105899076049
2	Linear kernel	0.7938877043354655
3	Polynomial kernel	0.7398720682302772
4	Sigmoid kernel	0.7057569296375267

The kernel that performs the best is the linear kernel having an accuracy of 0.794.

Task 4 Validate best method

Document why you think that method produced the best results in the task above in terms of predictive capacity as per **Task 4.**

The method that produced the best results is Random Forest because its accuracy is slightly higher (0.802) than the SVM accuracy (0.794). Moreover, the performance of the Random Forest algorithm was faster than the SVC model. Therefore, for this data set, the best method is Random Forest.

Task 5 Provide table

Provide the completed table which documents your construction of a multivariate linear model, using "target" as the response variable and three of the continuous features available within the dataset as per **Task 2.**

Model attempt	IF1 NAME	Is significant?	IF2 NAME	Is significant?	IF3 NAME	Is significant?	R-squared	Model accuracy (as correlation coefficient)
1	Age	Yes					0.171	-0.215146
2	Age	Yes	CRIM	Yes			0.241	-0.155104
	Age	Yes	ZN	Yes				
3	Age	Yes	CRIM	Yes	DIS	Yes	0.250	-0.093491
	Age	Yes	CRIM	Yes	INDUS	Yes		
	Age	Yes	RM	Yes	DIS	Yes		

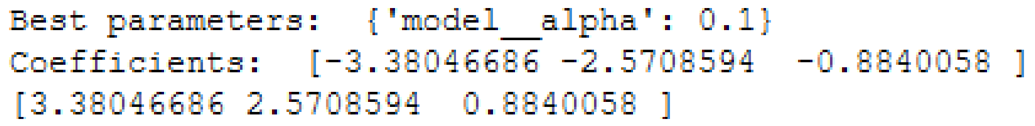
Task 6 Record the parameters that were used for the Random Forest Regressor

Build a Random Forest Regressor using this data. Record the parameters that were used for the RF regressor.

Task 6.1 What does it tell you?

Document the importance of features using Randomized Search, and their standardised beta coefficients as per **Task 3 a)**

Analysing the coefficients obtained for this model, it is clear that the feature that has a higher impact on the prediction of MEDV is Age, then is followed by CRIM and finally DIS. This can be told through the weights of the standardised beta coefficients.



```
Best parameters:  {'model__alpha': 0.1}
Coefficients:  [-3.38046686 -2.5708594  -0.8840058 ]
[3.38046686 2.5708594  0.8840058 ]
```

Task 6.2 How does it compare?

Document how the predictive performance of the model compares with the linear models

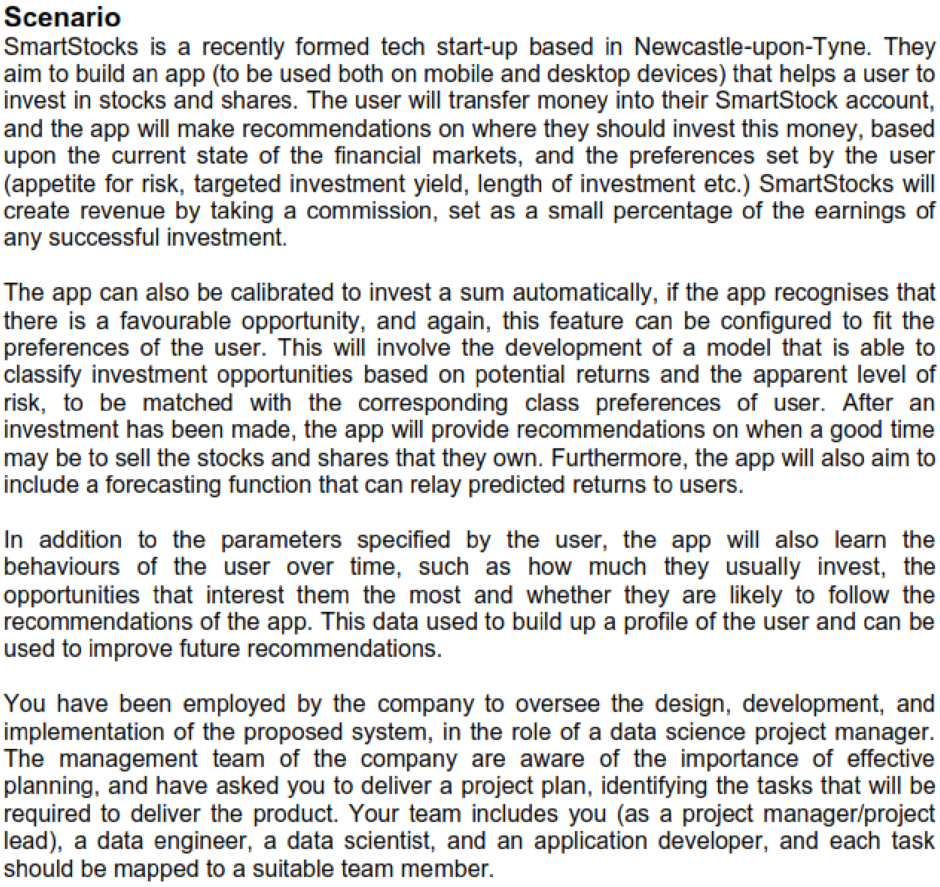
the Random Forest model used with the features 'AGE','CRIM','DIS' has a $R^2 = 0.676$. In contrast, the multilinear model for the same features has a $R^2 = 0.250$. Comparing both values, it is concluded that the Random Forest model has better predictive performance since the R^2 is higher than the computed for the multilinear model.

Part 6 – Project Plan

In **Week 10 – Chapter 4 – Tasks project Planning**, you were given the scenario of a recently formed tech start-up called SmartStocks, that aimed to build an app for stock and share trading. You were asked to document a high-level project plan, outlining the tasks that would be involved to develop and implement an application. Please provide evidence in your portfolio for the following activities from the **Week 10 – Chapter 4 – Tasks**:

Task 1 Identify data and method of collection

Document the data that would be required for the system to work as intended and how this data could be collected or obtained



Scenario

SmartStocks is a recently formed tech start-up based in Newcastle-upon-Tyne. They aim to build an app (to be used both on mobile and desktop devices) that helps a user to invest in stocks and shares. The user will transfer money into their SmartStock account, and the app will make recommendations on where they should invest this money, based upon the current state of the financial markets, and the preferences set by the user (appetite for risk, targeted investment yield, length of investment etc.) SmartStocks will create revenue by taking a commission, set as a small percentage of the earnings of any successful investment.

The app can also be calibrated to invest a sum automatically, if the app recognises that there is a favourable opportunity, and again, this feature can be configured to fit the preferences of the user. This will involve the development of a model that is able to classify investment opportunities based on potential returns and the apparent level of risk, to be matched with the corresponding class preferences of user. After an investment has been made, the app will provide recommendations on when a good time may be to sell the stocks and shares that they own. Furthermore, the app will also aim to include a forecasting function that can relay predicted returns to users.

In addition to the parameters specified by the user, the app will also learn the behaviours of the user over time, such as how much they usually invest, the opportunities that interest them the most and whether they are likely to follow the recommendations of the app. This data used to build up a profile of the user and can be used to improve future recommendations.

You have been employed by the company to oversee the design, development, and implementation of the proposed system, in the role of a data science project manager. The management team of the company are aware of the importance of effective planning, and have asked you to deliver a project plan, identifying the tasks that will be required to deliver the product. Your team includes you (as a project manager/project lead), a data engineer, a data scientist, and an application developer, and each task should be mapped to a suitable team member.

Data required	How can be collected or obtained?
Potential customers	Taken from information available on social media, define if a person could be a potential customer, according to their preferences.
User's personal information (Name, sex, age, bank account number, country or region)	Obtained from the user when the account is created and the user interacts in the app.
Amount of money to transfer to the SmartStock account	Obtained after the user makes the transfer to the SmartStock. This operation will activate a trigger to look for investment recommendations.
Stok exchange streaming for a chosen country or region	Obtained from historical information extracted from official stocks
Factors affecting the stock market • Supply and demand • Company related factors • Investor sentiment • Interest rates • Politics • Current events • Natural calamities • Exchange rates	• Stock markets • Newspapers articles • National and international statistics about products and services offered by the companies where the inversion would be made. • Newsletter of the business • Specific information published about the companies where the inversion would be made.

Task 2 Evidence high-level project plan

Produce the high-level project plan, based on the stages presented in either CRISP-DM or TDSL. For each stage, identify the key tasks, provide a brief description of each (indicating why it is required) and assign a suitable team member as task owner

In addition, you should provide commentary on the following issues, which should contribute towards understanding the eventual feasibility of the product:

Phase 1: Business understanding SmartStocks business needs to build an app that helps a user to invest in stocks and shares. **Responsible (Task owner): Project manager**	
1.1. Determine Business Objectives	
Key tasks	**Description (Why is required)**
Background	• Determine if the app I want to build is needed to define potential customers. • Know if there is another business doing the same to determine how will we differentiate from others.
Business Objectives	• Determine what is the minimum number of clients and transactions that I need to define if the business is profitable. • Define if the business will have a social cause to attract a specific group of customers.
Business success criteria	• Define the number of users and profit wanted to achieve during a time to evaluate if the app developed has achieved its objectives.
1.2. Assess Situation	
Key tasks	**Description (Why is required)**
Inventory of resources	Required to define the resources I have and the ones I need to get, this includes: • Personnel • Data • Computing resources • Software
Requirements, assumptions and constraints	Data needed to define: • schedules • security and legal issues. • Size of the data. If it has not a good size, what are the implications? • Where the app can be used (scope)
Risks and contingencies	• To know the potential risks or events that may affect the project or cause it to fail. • The contingencies are useful to have an action plan to apply in case a risk or event takes place.
Terminology	Required to assure that there is a reasonable knowledge of the field where the app will work.
Costs and benefits	To compare the costs of the project with the potential benefits to the business if it is successful.
1.3. Determine data mining goals	
Key tasks	**Description (Why is required)**
Business and data mining success criteria	To define the accuracy of the algorithms used and their power of prediction.
1.4. Produce project plan	
Key tasks	**Description (Why is required)**
Project Plan	To define the stages of the project, their duration, duration, inputs, outputs and dependencies.
Initial assessment of tools and techniques	To define if the methodology and tools used are producing the expected result.

Phase 2: Data Understanding What data do we have/need? Is it clean? Take the data mentioned in task 1 **Responsible (Task owner): Data Engineer**	
Key tasks	**Description (Why is required)**
Describe data	Required to define the format of the data, its quantity and if it is enough to satisfy the requirements of the project.
Explore data	To define the key attributes, relationships and simple statistical analyses that help us to get first insights and set hypothesis.
Verify data quality	Needed to know if the data: • covers the scope of the app • has errors and its frequency • is complete or has missing values
Data quality report	To know the status of the data and if there is a quality issue how could be addressed.
Phase 3: Data Preparation How do we organize the data for modelling? **Responsible (Task owner): Data Scientist**	
Key tasks	**Description (Why is required)**
Select data	In function of the status of the data define which data will be relevant to achieve the goals of the app.
Clean data	To assure that the data taken has a good quality.
Construct required data	• To apply transformations or derived attributes to the data to create a positive impact on the analysis of results. • To generate records for special cases scenarios.
Integrate data	• To merge data that have different information about the same objects. • To aggregate new values that summarized information from multiple records or tables.
Phase 4: Modelling What modelling techniques should we apply **Responsible (Task owner): Data Scientist**	
Key tasks	**Description (Why is required)**
Select modelling technique	To document the actual modelling technique and the assumptions made about the data
Generate test design	To test the model's quality and validity (define which data will be used for testing, training or evaluating)
Build model	To have one or more models that give us the suggestions for the best inversions
Assess model (This task should also be revised by the project manager)	To define the accuracy of the models and determine which is the best model to apply and under what conditions.
Phase 5: Evaluation **Responsible (Task owner): Project manager**	

Key tasks	**Description (Why is required)**
Evaluate results	To determine the degree to which the model meets the business objectives and evaluate the weakness of the models from the business perspective.
Review process	To summarise the process and highlight activities missed and those which have to be repeated
Determine next steps	To list the possible actions to take considering the actual budget
Phase 6: Evaluation	
Key tasks	**Description (Why is required)**
Plan deployment **(Task owner): Application developer**	To define the strategy and the action to be taken to launch the app.
Plan monitoring and maintenance **(Task owner): Application developer**	To define the monitoring and maintenance strategy to keep up the app up to day
Produce final report **Responsible: All participate in this task indicating their participation**	To summarize the results of the app
Review project **Responsible (Task owner): Project manager**	To assess what went right or wrong and possible ways to improve. This step will be in constant revision, to ensure that the app is working up to standard.

Task 3 Key ethical issues

The key ethical issues that might be created by the app as per **Task 3**

How this technology could be misused?

The algorithms used may be corrupted to favour the investment in some companies or to prevent it from other companies.

Is the training data fair and representative?

Since the app will be fed by the newspapers' information, it is more likely that big companies are more likely to be taken into account, therefore the algorithm may be biased since it may exclude small entrepreneurs. To address this issue it could be created a segment that includes and promotes investment in small businesses.

Potential sources of bias in the data

Much of the information to predict positive investments will come from news taken from recognised founts but this data may be biased so it will be important to define the credibility of the news and contrast it against other founts.

User consent for the use of the data

- The data to acquire potential customers will come from social media, therefore, it is needed to be cautious with this stage, since private information can be taken without permission so it is important to ensure that there is explicit permission to use that information.

- To get more information about the business and improve the power of predictability, data like business newsletters are taken into consideration, so it is needed to have also explicit permission to use that information.
- Since the app will make suggestions to the user about future investments it will be required to analyse the user's behaviour and the access to this private information should be communicated to the users to then be consented to them.

It is worthy to mention that the activities below are taken into consideration during the next phases:

- **Phase 6 evaluation**: shut down the tool if the results produce negative outcomes
- **Phase 6 evaluation:** monitor the output to ensure the model remains fair
- **Phase 1.2 Requirements, assumptions and constraints:** Plan to protect and secure user data

Task 4 Legal implications

The legal implications of the app and if these could prevent any part of the system from being implemented as planned

To define potential customers, personal data will be taken, therefore, as part of the documentation of the projects, it must be proven that the data taken is specific, explicit and has legitimate purposes. Moreover, during the data cleaning, it must be checked that the data is relevant and relevant and limited to achieve the business objectives.

To avoid the identification of the potential customers, their names and any other personal information that may allow identifying the potential customer will be deleted from the dataset during the first steps of data cleaning.

If the above steps are not taken, then the project may not be implemented as planned since the business depends on attracting users that want to use the app to invest, thus it must be guaranteed that their personal information is kept under the applicable legislation.