Felix_Week2_Assign_Pima

August 17, 2022

[ ]:

```
#Q 1. Import the necessary libraries and briefly explain the use of each library

# import the important packages
import pandas as pd # library used for data manipulation and analysis
import numpy as np # library used for working with arrays
import matplotlib.pyplot as plt # library for plots and visualisations
import seaborn as sns # library for visualisations

%matplotlib inline

import scipy.stats as stats # this library contains a large number of␣
 ↪probability distributions as well as

from scipy.stats import norm # this library is used for normal distribution
```

[186]:

```
#Q3. Show the last 10 records of the dataset. How many columns are there?
# There are 9 Columns
diabetes = pd.read_csv("diabetes.csv")
diabetes.tail(10)
diabetes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

[162]:
```
#Q4. Show the first 10 records of the dataset
diabetes.head(10)
```

[162]:
```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin        BMI  \
0            6      148             72             35       79  33.600000
1            1       85             66             29       79  26.600000
2            8      183             64             20       79  23.300000
3            1       89             66             23       94  28.100000
4            0      137             40             35      168  43.100000
5            5      116             74             20       79  25.600000
6            3       78             50             32       88  31.000000
7           10      115             69             20       79  35.300000
8            2      197             70             45      543  30.500000
9            8      125             96             20       79  31.992578

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
5                     0.201   30        0
6                     0.248   26        1
7                     0.134   29        0
8                     0.158   53        1
9                     0.232   54        1
```

[178]:
```
# Q5. What do you understand by the dimension of the dataset? Find the␣
 ↪dimension of the `pima` dataframe.
# What I understand by its dimension of the dataset is it is a DataFrame since␣
 ↪being two-dimensional It contains 768 rows
# and 9 colums
diabetes.ndim
```

[178]:
```
2
```

[182]:
```
diabetes.shape
```

[182]:
```
(768, 9)
```

[181]:
```
# Q6. What do you understand by the size of the dataset? Find the size of the␣
 ↪`pima` dataframe.
# since the size of the dataset is 6912, it is composed of 768 rows and 9␣
 ↪columns.
diabetes.size
```

[181]:
```
6912
```

[190]:
```
#Q7. What are the data types of all the variables in the data set?
diabetes.dtypes
```

[190]:
```
Pregnancies                   int64
Glucose                       int64
BloodPressure                 int64
SkinThickness                 int64
Insulin                       int64
BMI                         float64
DiabetesPedigreeFunction    float64
Age                           int64
Outcome                       int64
dtype: object
```

[ ]:
```

```

[197]:
```
# Q8 What do you mean by missing values? Are there any missing values in the␣
 ↪`pima` dataframe?
# missing values of cells that are voided of values. There are no missing␣
 ↪values in the dataset.
diabetes.isnull()
```

[197]:
```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin    BMI  \
0          False    False          False          False    False  False
1          False    False          False          False    False  False
2          False    False          False          False    False  False
3          False    False          False          False    False  False
4          False    False          False          False    False  False
..           …        …              …              …        …      …
763        False    False          False          False    False  False
764        False    False          False          False    False  False
765        False    False          False          False    False  False
766        False    False          False          False    False  False
767        False    False          False          False    False  False

     DiabetesPedigreeFunction    Age  Outcome
0                       False  False    False
1                       False  False    False
2                       False  False    False
3                       False  False    False
4                       False  False    False
..                        …      …        …
763                     False  False    False
764                     False  False    False
765                     False  False    False
766                     False  False    False
767                     False  False    False
```

```
[768 rows x 9 columns]
```

[191]:

```
# Q9. What does summary statistics of data represents? Find the summary␣
 ↪statistics for all variables except 'Outcome'
#in the `pima` data? Take one column/variable from the output table and explain␣
 ↪all the statistical measures.

# Summary statistics give an overview or summary of descritive statistics on␣
 ↪the variables of the sataset.
diabetes.describe()
```

[191]:

```
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  121.675781      72.250000      26.447917  118.270833
std       3.369578   30.436252      12.117203       9.733872   93.243829
min       0.000000   44.000000      24.000000       7.000000   14.000000
25%       1.000000   99.750000      64.000000      20.000000   79.000000
50%       3.000000  117.000000      72.000000      23.000000   79.000000
75%       6.000000  140.250000      80.000000      32.000000  127.250000
max      17.000000  199.000000     122.000000      99.000000  846.000000

              BMI  DiabetesPedigreeFunction         Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    32.450805                  0.471876   33.240885    0.348958
std      6.875374                  0.331329   11.760232    0.476951
min     18.200000                  0.078000   21.000000    0.000000
25%     27.500000                  0.243750   24.000000    0.000000
50%     32.000000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
max     67.100000                  2.420000   81.000000    1.000000
```

[196]:

```
# Countis the number of BloodPressure entries
# mean is the average of BloodPressure values
# std is the standard deviation (deviation from the mean) of BloodPressure␣
 ↪values
#min is the mininum value of BloodPressure values
# 25% is the 25 percentile mark
#50% is the 50 percentile mark(median) of BloodPressure values
# 75% is the 75 percentilemark of BloodPressurevalues
# max is the maximum value of BloodPressure values
diabetes["BloodPressure"].describe()
```

[196]:

```
count    768.000000
mean      72.250000
std       12.117203
min       24.000000
```

```
25%       64.000000
50%       72.000000
75%       80.000000
max      122.000000
Name: BloodPressure, dtype: float64
```

[198]:
```
diabetes.BloodPressure
```

[198]:
```
0      72
1      66
2      64
3      66
4      40
       ..
763    76
764    70
765    72
766    60
767    70
Name: BloodPressure, Length: 768, dtype: int64
```

[214]:
```
# Estimate the mean and standard deviation of BloodPressure values
mu = diabetes["BloodPressure"].mean()
print("The estimated mean is", round(mu,2))
```

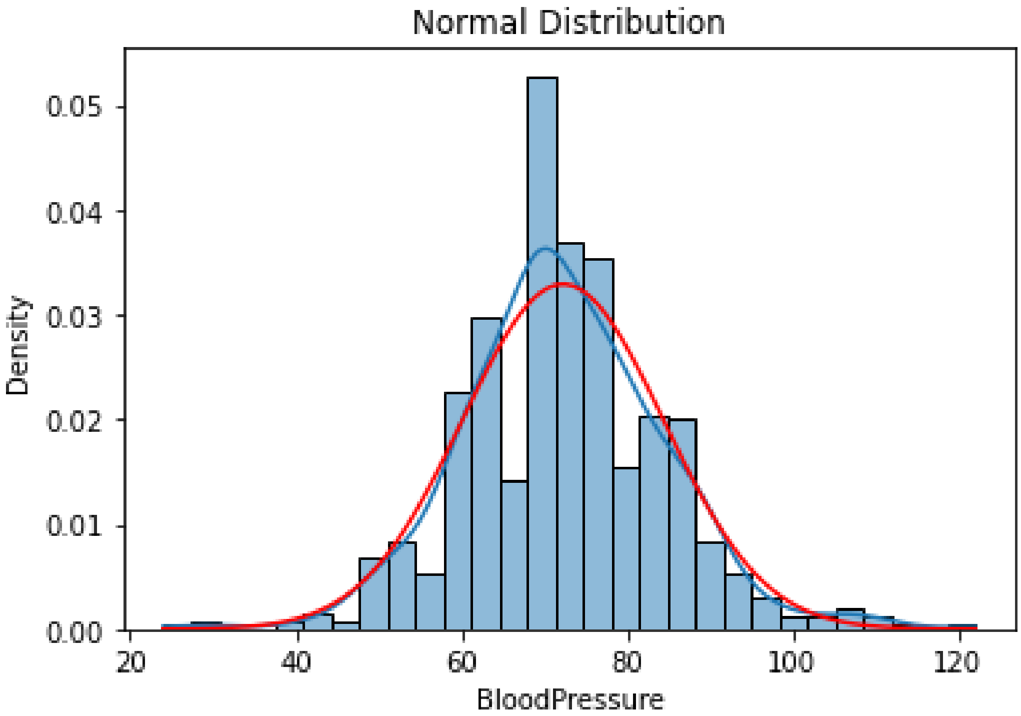
```
The estimated mean is 72.25
```

[215]:
```
sigma = diabetes["BloodPressure"].std()
print("The estimated standard deviation is", round(sigma, 2))
```

```
The estimated standard deviation is 12.12
```

[210]:
```
# Q 10. Plot the distribution plot for the variable 'BloodPressure'. Write␣
 ↪detailed observations from the plot.

# Below, as we can see, theblue curve display the shape of sata distribution␣
 ↪while the red curve the PDF
# (Probabilitydensity function). It is obvious, the dataset is approsimately␣
 ↪normal. Therefore, our assumption is that
# the data distribution from the dataset to be normal and normality assumption␣
 ↪is what our calculations will be based upon.

density = pd.DataFrame()
density["x"] = np.linspace(diabetes["BloodPressure"].min() - 0.01,␣
 ↪diabetes["BloodPressure"].max() + 0.01, 100)
density["pdf"]= norm.pdf(density["x"], mu, sigma)

fig, ax = plt.subplots()
```

```
# plot the distribution of data using histogram
sns.histplot(diabetes["BloodPressure"], ax=ax, kde=True, stat="density")

# plot the pdf of the normal distribution
ax.plot(density["x"], density["pdf"], color="red")
plt.title("Normal Distribution")
plt.show()
```



[218]:

```
# Q 11. What is the 'BMI' for the person having the highest 'Glucose'?

# The BMI for the person having the highest Glucose is 67.10

diabetes["BMI"].describe()
```

[218]:

```
count    768.000000
mean      32.450805
std        6.875374
min       18.200000
25%       27.500000
50%       32.000000
75%       36.600000
max       67.100000
Name: BMI, dtype: float64
```

[226]:
```
# Q 12. Q 12.1 What is the mean of the variable 'BMI'?
#12.3 What is the mean of the variable 'BMI'?
diabetes["BMI"].mean()

#12.2 What is the median of the variable 'BMI'?
#median = 32
diabetes["BMI"].median()

#12.3 What is the mode of the variable 'BMI'?
diabetes["BMI"].mode()


#12.4 Are the three measures of central tendency equal?
# Yes, all three measures of central tendency are equal.
#mean = median = mode = 32.0
```

[226]:
```
0    32.0
Name: BMI, dtype: float64
```

[237]:
```
# use describe method to find the mean
diabetes["Glucose"].describe()
```

[237]:
```
count    768.000000
mean     121.675781
std       30.436252
min       44.000000
25%       99.750000
50%      117.000000
75%      140.250000
max      199.000000
Name: Glucose, dtype: float64
```

[245]:
```
# Sort the Glucose values in ascending orders

diabetes.sort_values('Glucose')
```

[245]:
```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
62             5       44             62             20       79  25.0
680            2       56             56             28       45  24.2
146            9       57             80             37       79  32.8
537            0       57             60             20       79  21.7
352            3       61             82             28       79  34.4
..           …        …             …             …        …   …
579            2      197             70             99       79  34.7
408            8      197             74             20       79  25.9
8              2      197             70             45      543  30.5
561            0      198             66             32      274  41.3
```

```
661            1      199             76             43       79  42.9

     DiabetesPedigreeFunction  Age  Outcome  Glucose_ranked
62                      0.587   36        0           768.0
680                     0.332   22        0           767.0
146                     0.096   41        0           766.0
537                     0.735   67        0           766.0
352                     0.243   46        0           764.0
..                        …   …      …             …
579                     0.575   62        1             6.0
408                     1.191   39        1             6.0
8                       0.158   53        1             6.0
561                     0.502   28        1             2.0
661                     1.394   22        1             1.0

[768 rows x 10 columns]
```

[246]:

```
# Q 13. How many women's 'Glucose' level is above the mean level of 'Glucose'?


# locate the mean = 121.675781 and rank it.

# The mean is located below "Glucose_ranked" of 343. So, there are 343 values␣
 ↪above the mean

diabetes.iloc[340:348]
```

[246]:

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
340            1      130             70             13      105  25.9
341            1       95             74             21       73  25.9
342            1      120             68             35       79  32.0
343            5      122             86             20       79  34.7
344            8       95             72             20       79  36.8
345            8      126             88             36      108  38.5
346            1      139             46             19       83  28.7
347            3      116             69             20       79  23.5

     DiabetesPedigreeFunction  Age  Outcome  Glucose_ranked
340                     0.472   22        0           258.0
341                     0.673   36        0           626.0
342                     0.389   22        0           365.0
343                     0.290   33        0           343.0
344                     0.485   57        0           626.0
345                     0.349   49        0           297.0
346                     0.654   22        0           205.0
347                     0.187   23        0           400.0
```

[257]:

```
#Q 14. How many entries (women) have their 'BloodPressure' equal to the median␣
 ↪of 'BloodPressure'
# and their 'BMI' less than the median of 'BMI'?

# There are 113 entries (women) that their 'BloodPressure' equal to the median␣
 ↪of 'BloodPressure'
# and their 'BMI' less than the median of 'BMI'


# BloodPressure median = 72.0

# 1st, sort BloodPressure
# Second, locate all values of median =72, then sum the corresponding women's␣
 ↪entries that are equal to the BloodPlressure
# of median 72.0.

# So, there are 234 entries (women)
diabetes["BloodPressure"].median()
```

[257]:

```
72.0
```

[255]:

```
diabetes["BMI"].median()
```

[255]:

```
32.0
```

[285]:

```
ranked = diabetes.sort_values(['BloodPressure','BMI'])
```

[286]:

```
ranked
```

[286]:

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
597            1       89             24             19       25  27.8
18             1      103             30             38       83  43.3
125            1       88             30             42       99  55.0
599            1      109             38             18      120  23.1
4              0      137             40             35      168  43.1
..           …        …              …              …        …    …
549            4      189            110             31       79  28.5
43             9      171            110             24      240  45.4
177            0      129            110             46      130  67.1
691           13      158            114             20       79  42.3
106            1       96            122             20       79  22.4

     DiabetesPedigreeFunction  Age  Outcome  Glucose_ranked
597                     0.559   21        0           675.0
18                      0.183   33        0           537.0
125                     0.496   26        1           684.0
599                     0.407   26        0           471.0
```

```
4                       2.288   33        1           218.0
..                         …  …      …             …
549                     0.680   37        0            22.0
43                      0.721   54        1            69.0
177                     0.319   26        1           272.0
691                     0.257   44        1           112.0
106                     0.207   27        0           613.0

[768 rows x 10 columns]
```

[287]:

```
ranked.iloc[19:349]
```

[287]:

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
258            1      193             50             16      375  25.9
243            6      119             50             22      176  27.1
687            1      107             50             19       79  28.3
98             6       93             50             30       64  28.7
313            3      113             50             10       85  29.5
..           …      …            …             …      …   …
515            3      163             70             18      105  31.6
262            4       95             70             32       79  32.1
570            3       78             70             20       79  32.5
191            9      123             70             44       94  33.1
241            4       91             70             32       88  33.1

     DiabetesPedigreeFunction  Age  Outcome  Glucose_ranked
258                     0.655   24        0            16.0
243                     1.318   33        1           376.0
687                     0.181   29        0           495.0
98                      0.356   23        0           640.0
313                     0.626   25        0           426.0
..                         …  …      …             …
515                     0.268   28        1            92.0
262                     0.612   24        0           626.0
570                     0.270   39        0           739.0
191                     0.374   40        0           331.0
241                     0.446   22        0           658.0

[330 rows x 10 columns]
```

[289]:

```
# Q 15. Below is the pairplot of variables 'Glucose', 'SkinThickness' and␣
 ↪'DiabetesPedigreeFunction'.
#Write you observations from the plot.

# My observation is that the variables take many different forms of␣
 ↪distribution. It shows the relationship
# with each other.Some show histogram while others are skewed.
```

```
diabetes.head(20)
```

```
[289]:     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin        BMI  \
       0             6      148             72             35       79  33.600000
       1             1       85             66             29       79  26.600000
       2             8      183             64             20       79  23.300000
       3             1       89             66             23       94  28.100000
       4             0      137             40             35      168  43.100000
       5             5      116             74             20       79  25.600000
       6             3       78             50             32       88  31.000000
       7            10      115             69             20       79  35.300000
       8             2      197             70             45      543  30.500000
       9             8      125             96             20       79  31.992578
       10            4      110             92             20       79  37.600000
       11           10      168             74             20       79  38.000000
       12           10      139             80             20       79  27.100000
       13            1      189             60             23      846  30.100000
       14            5      166             72             19      175  25.800000
       15            7      100             69             20       79  30.000000
       16            0      118             84             47      230  45.800000
       17            7      107             74             20       79  29.600000
       18            1      103             30             38       83  43.300000
       19            1      115             70             30       96  34.600000

           DiabetesPedigreeFunction  Age  Outcome  Glucose_ranked
       0                      0.627   50        1           148.0
       1                      0.351   31        0           701.0
       2                      0.672   32        1            35.0
       3                      0.167   21        0           675.0
       4                      2.288   33        1           218.0
       5                      0.201   30        0           400.0
       6                      0.248   26        1           739.0
       7                      0.134   29        0           410.0
       8                      0.158   53        1             6.0
       9                      0.232   54        1           311.0
       10                     0.191   30        0           459.0
       11                     0.537   34        1            76.0
       12                     1.441   57        0           205.0
       13                     0.398   59        1            22.0
       14                     0.587   51        1            82.0
       15                     0.484   32        1           576.0
       16                     0.551   31        1           382.0
       17                     0.254   31        1           495.0
       18                     0.183   33        0           537.0
       19                     0.529   32        1           410.0
```

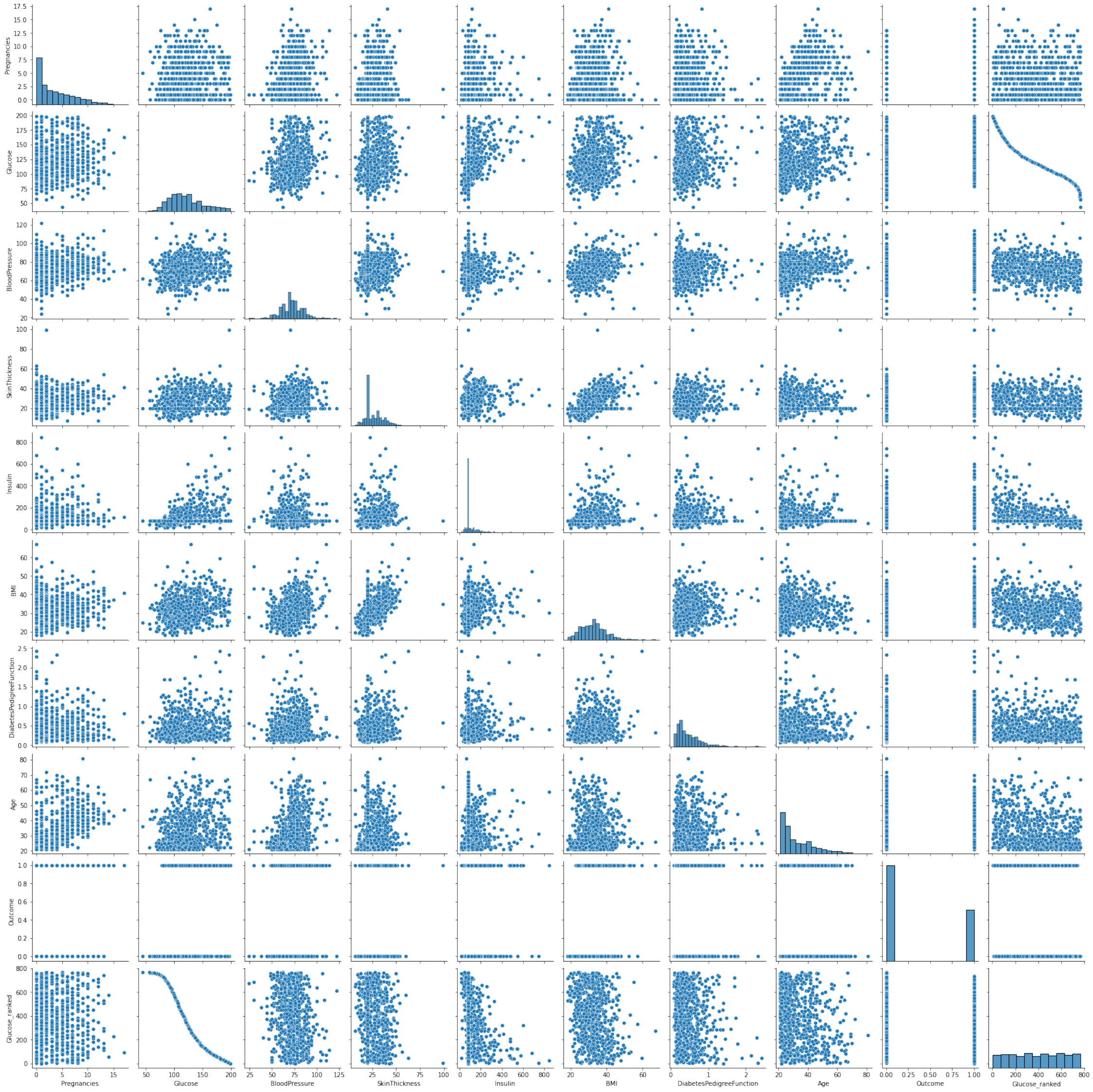
[290]:
```
#set the figure size
plt.figure(figsize = (11,11))
```

[290]:
```
<Figure size 792x792 with 0 Axes>
```

```
<Figure size 792x792 with 0 Axes>
```

[291]:
```
#plot a pairt plot
sns.pairplot(diabetes)
```

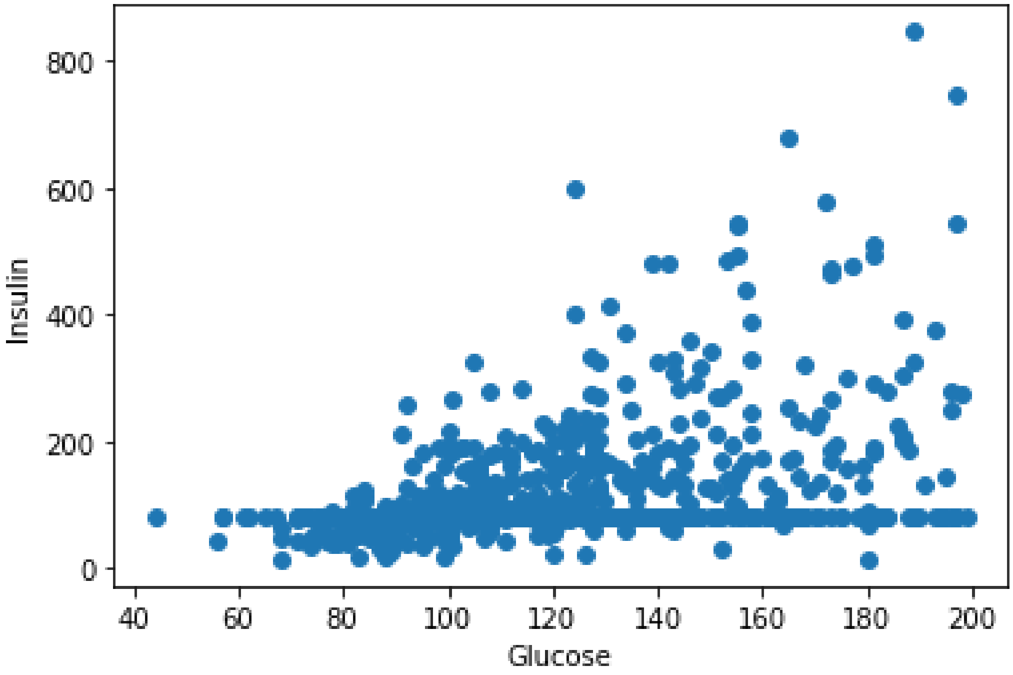
[291]:
```
<seaborn.axisgrid.PairGrid at 0x291c136b4f0>
```



[293]:
```
plt.show()
```

[294]:

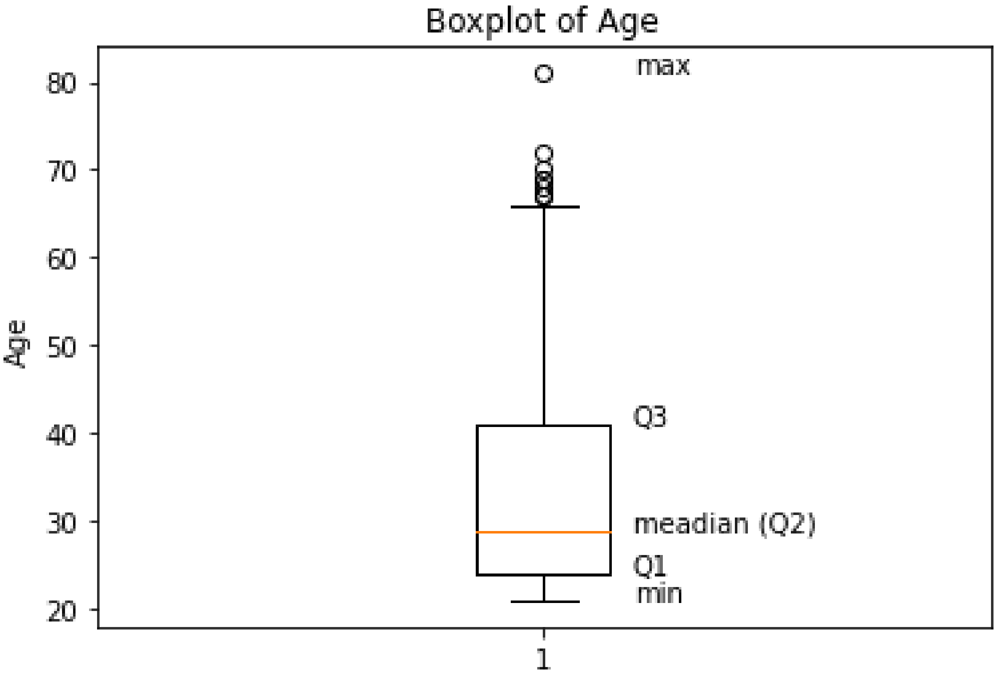
```
# Q 16. Plot the scatterplot between 'Glucose' and 'Insulin'. Write your␣
 ↪observations from the plot.

# The idea from the scatter plot,the more glucose,the more insulin. There is a␣
 ↪rough positive correlation between the two

#data
X = diabetes['Glucose']
Y = diabetes['Insulin']

#Plot the scatter plot
plt.scatter(X,Y)

# add the axes labels to the plot
plt.xlabel('Glucose')
plt.ylabel('Insulin')

# display the plot
plt.show()
```



[296]:

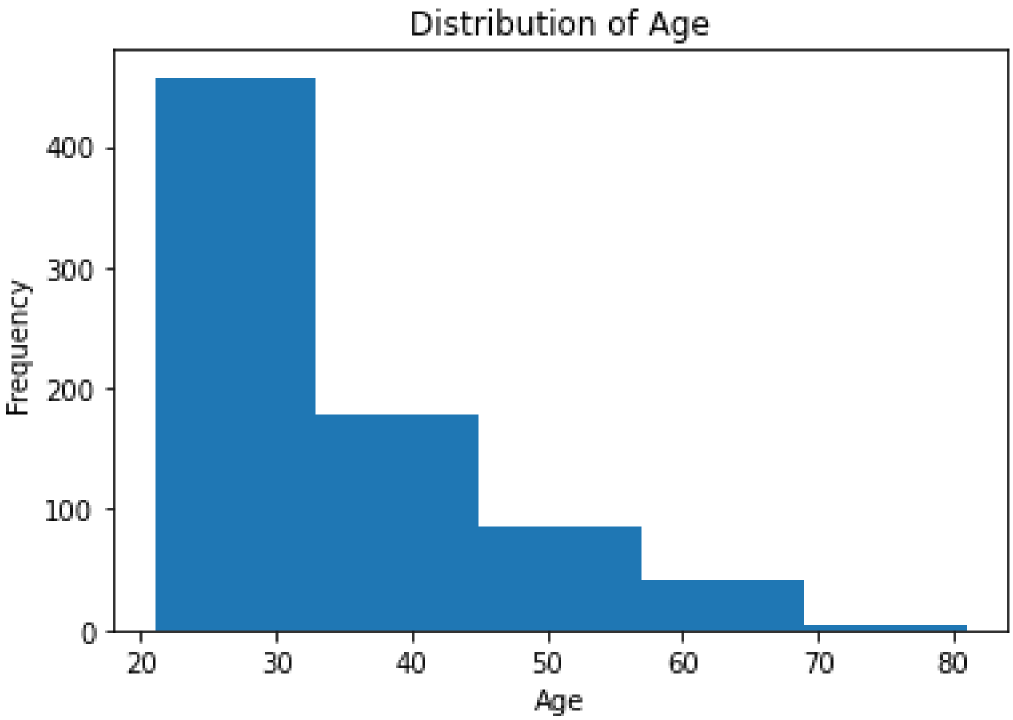
```
# Q 17. Plot the boxplot for the 'Age' variable. Are there outliers?


Yes, they are outliers: these points outside the whisker
```

```
# plot a distribution of Age
plt.boxplot(diabetes['Age'])

# add labels for five numbersummary
plt.text(x = 1.1, y = diabetes['Age'].min(), s='min')
plt.text(x = 1.1, y = diabetes.Age.quantile(0.25), s ='Q1')
plt.text(x = 1.1, y = diabetes['Age'].median(), s ='meadian (Q2)')
plt.text(x = 1.1, y = diabetes.Age.quantile(0.75), s ='Q3')
plt.text(x = 1.1, y = diabetes['Age'].max(), s ='max')

# ass the graphtitle andaxes labels
plt.title('Boxplot of Age')
plt.ylabel('Age')

# display the plot
plt.show()
```



[298]:

```
# Q 18. Plot histograms for variable Age to understand the number of women in␣
↪different Age groups given that they have diabetes or not.
# Explain both histograms and compare them.

# The Age variable is positively skewed.
# plot the histogram
# specify the number of bins, using 'bins' parameter
plt.hist(diabetes['Age'], bins = 5)

# add the graph title and axes labels
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')

# display theplot
plt.show()
```



[299]:
```
# Q 19. What is Inter Quartile Range of all the variables? Why is it used?␣
 ↪Which plot visualizes the same?
# Interquatile IQR = Q3 - Q1 = 41.0000 - 24.000 = 17.000.

# Interquatile Range is used, in a modified boxplot, to represent outliers as␣
 ↪special points. The value of IQR can be used
# to idetify outliers as follows: above Q3 by anamount greater than 1.5 X IQR␣
 ↪or below Q1 by an amount greater than 1.5 X IQR

diabetes["Age"].describe()
```

```
[299]: count    768.000000
       mean      33.240885
       std       11.760232
       min       21.000000
       25%       24.000000
       50%       29.000000
       75%       41.000000
       max       81.000000
       Name: Age, dtype: float64
```

```
[300]: # Q 20. Find and visualize the the correlation matrix. Write your observations␣
        ↪from the plot.

       # The corrrelation matrix shows correlation coefficients between sets of␣
        ↪variables in the dataset. For instance some
       # varables could be postively correlated while others could be negatively␣
        ↪correlated. Pregancies and insulin are negative
       # correlated.
       corrM = diabetes.corr()
```

```
[301]: corrM
```

```
[301]:                           Pregnancies   Glucose  BloodPressure  SkinThickness  \
       Pregnancies                  1.000000  0.128022       0.208987       0.009393
       Glucose                      0.128022  1.000000       0.219765       0.158060
       BloodPressure                0.208987  0.219765       1.000000       0.130403
       SkinThickness                0.009393  0.158060       0.130403       1.000000
       Insulin                     -0.018780  0.396137       0.010492       0.245410
       BMI                          0.021546  0.231464       0.281222       0.532552
       DiabetesPedigreeFunction    -0.033523  0.137158       0.000471       0.157196
       Age                          0.544341  0.266673       0.326791       0.020582
       Outcome                      0.221898  0.492884       0.162879       0.171857
       Glucose_ranked              -0.137137 -0.973619      -0.235622      -0.151535

                                  Insulin       BMI  DiabetesPedigreeFunction  \
       Pregnancies              -0.018780  0.021546                 -0.033523
       Glucose                   0.396137  0.231464                  0.137158
       BloodPressure             0.010492  0.281222                  0.000471
       SkinThickness             0.245410  0.532552                  0.157196
       Insulin                   1.000000  0.189919                  0.158243
       BMI                       0.189919  1.000000                  0.153508
       DiabetesPedigreeFunction  0.158243  0.153508                  1.000000
       Age                       0.037676  0.025748                  0.033561
       Outcome                   0.178696  0.312254                  0.173844
       Glucose_ranked           -0.386221 -0.232370                 -0.119526

                                      Age   Outcome  Glucose_ranked
```

```
Pregnancies                 0.544341  0.221898       -0.137137
Glucose                     0.266673  0.492884       -0.973619
BloodPressure               0.326791  0.162879       -0.235622
SkinThickness               0.020582  0.171857       -0.151535
Insulin                     0.037676  0.178696       -0.386221
BMI                         0.025748  0.312254       -0.232370
DiabetesPedigreeFunction    0.033561  0.173844       -0.119526
Age                         1.000000  0.238356       -0.274993
Outcome                     0.238356  1.000000       -0.481950
Glucose_ranked             -0.274993 -0.481950        1.000000
```

[ ]: