Text

Description automatically generated

**CSE303: Lab Assignment – 1 (Tasks on Lab 04)**

**Individual Assignment**

|  |  |
| --- | --- |
| Student ID | 2019-1-60-063 |
| Student Name | Sazzad Hossen |
| Course Name | Statistics for Data Science |
| Course Code | CSE303 |
| Section | 02 |

**Submitted to:**

**Md Al-Imran**

Lecturer

Department of Computer Science & Engineering

**Date of submission**: 28 March 2022

**LAB TASKS**

import pandas as pd

df = pd.read\_csv('C:\\Users\\Hp\\OneDrive\\Desktop\\10TH SEMESTER\\CSE 303\\LAB 303\\LAB 4\\dataset\_lab04.csv')

print("DATASET INFORMATION : ")

df.info()

print("1. How many rows and columns this dataframe has? Print this information. ")

def lab04\_task1\_2019\_1\_60\_063():

 print ('Number of rows: ', df.shape[0])

 print ('Number of columns: ', df.shape[1])

lab04\_task1\_2019\_1\_60\_063()

print("2. Describe (numerical summary) the time and amount column. Print this information.")

def lab04\_task2\_2019\_1\_60\_063():

 print(df[['Time', 'Amount']].describe())

lab04\_task2\_2019\_1\_60\_063()

print("3. There are 31 columns in the dataset. Compute some statistical measures like mean, median, standard deviation, variance using Pandas Function for at least two columns. Print this information.")

#For time and Amount column

def lab04\_task3\_2019\_1\_60\_063():

 df.columns = ['Time','V1','V2','V3','V4','V5','V6','V7','V8','V9','V10','V11','V12','V13','V14','V15','V16','V17','V18','V19','V20','V21','V22','V23','V24','V25','V26','V27','V28','Amount','Class']

 Time = df['Time']

 print("\nMeasures of Time\n")

 print("Time Mean: " ,Time.mean())

 print("Time Standard Deviation: ",Time.std())

 print("Time Variance: ",Time.var())

 print("Time Lower Quartile: " ,Time.quantile(0.25))

 print("Time Median: " ,Time.median())

 print("Time Upper Quartile: " ,Time.quantile(0.75))

 print("Time Skewness: " ,Time.skew())

 print("Time Kurtosis: " ,Time.kurt())

 print("Time Min: ",Time.min())

 print("Time Max: ",Time.max())

 Amount = df['Amount']

 print("\nMeasures of Amount\n")

 print("Amount Mean: " ,Amount.mean())

 print("Amount Standard Deviation: ",Amount.std())

 print("Amount Variance: ",Amount.var())

 print("Amount Lower Quartile: " ,Amount.quantile(0.25))

 print("Amount Median: " ,Amount.median())

 print("Amount Upper Quartile: " ,Amount.quantile(0.75))

 print("Amount Skewness: " ,Amount.skew())

 print("Amount Kurtosis: " ,Amount.kurt())

 print("Amount Min: ",Amount.min())

 print("Amount Max: ",Amount.max())

lab04\_task3\_2019\_1\_60\_063()

print("4. Show the Box Plot of Time and Amount column. Also print the value of Q1, Median, Q3, IQR. Are there any outliers? Explain your answer and print it.")

def lab04\_task4\_2019\_1\_60\_063():

 import assignModule as am

 import matplotlib.pyplot as plt

 df.boxplot(column = ['Time','Amount'])

 plt.title("Desired box plot", color='green')

 plt.show()

 Time = df['Time']

 Amount = df['Amount']

 print("Time Lower Quartile Q1: " ,Time.quantile(0.25))

 print("Time Median Q2: ",Time.median())

 print("Time Upper Quartile Q3: " ,Time.quantile(0.75))

 IQR1= am.IQR(Time)

 print("Time Interquartile renge(IQR) : ",IQR1)

 print("Amount Lower Quartile Q1: " ,Amount.quantile(0.25))

 print("Amount Median Q2: " ,Amount.median())

 print("Amount Upper Quartile Q3: " ,Amount.quantile(0.75))

 IQR2= am.IQR(Amount)

 print("Amount Interquartile renge(IQR) : ",IQR2)

 am.Whiskers(Time,IQR1)

 am.Whiskers(Amount,IQR2)

lab04\_task4\_2019\_1\_60\_063()

print("5. Show the Histogram of Time and Amount column. Print the value of the Skewness and Kurtosis using appropriate Pandas functions. Comment on the type of the data distribution and print it.")

def lab04\_task5\_2019\_1\_60\_063():

    import assignModule as am

    df.hist(column = ["Time", "Amount"], bins = 5, color='darkblue')

    Time = df['Time']

    print("Skewness of time: " , Time.skew()) #negative or left-skewed

    print("Kurtosis of time: " , Time.kurt()) ##Platykurtosis

    print("\n")

    am.skewness(Time)

    am.kurtosis(Time)

    print("\n")

    Amount = df['Amount']

    print("Skewness of amount: " , Amount.skew())  # positive or right skewed

    print("Kurtosis of amount: " , Amount.kurt()) # Leptokurtosis

    am.skewness(Amount)

    am.kurtosis(Amount)

lab04\_task5\_2019\_1\_60\_063()

print("6. Find the percentage of records with class value = 0 (Non-Fraudulent) and class value = 1 (Fraudulent).Print this information. ")

def lab04\_task6\_2019\_1\_60\_063():

    class0 = len(df[df['Class']==0])\*100

    class1 = len(df[df['Class']==1])\*100

    print("Percentage of class value 0 :",class0/len(df['Class']))

    print("Percentage of class value 1 :",class1/len(df['Class']))

lab04\_task6\_2019\_1\_60\_063()

print("7. Show the result you have got in 6 using a Histogram.")

def lab04\_task7\_2019\_1\_60\_063():

    import matplotlib.pyplot as plt

    df\_histo = df[['Class']]

    df\_histo.hist(column = ['Class'],color='g', bins = 5)

    plt.xlabel('class', color='magenta')

    plt.ylabel('data', color='darkblue')

    plt.title("Desired histogram", color='orange')

    plt.show()

lab04\_task7\_2019\_1\_60\_063()

print("8. Show the result you have got in 6 using a Bar chart. Create the bar chart on the percentage value, not on the total number of occurrences. ")

import matplotlib.pyplot as plt

def lab04\_task8\_2019\_1\_60\_063():

    Non\_Fraudulent = (df.loc[df['Class']==0])\*100

    Fraudulent = (df.loc[df['Class']==1])\*100

    a=Non\_Fraudulent.size/df.size

    b=Fraudulent.size/df.size

    x=[0,1]

    y=[a,b]

    x1\_tick = ['a','b']

    plt.bar(x,y, tick\_label= x1\_tick, color='brown')

    plt.xlabel('class', color='black')

    plt.ylabel('percentage', color='darkblue')

    plt.title("Desired bar chart", color='orange')

    plt.show()

lab04\_task8\_2019\_1\_60\_063()

print("9. Show the Histrogram (data distribution) of a few other columns (your choice) showing both positive and negative skew and also leptokurtic and platykurtic data distribution. So, you should display at least four Histograms.")

def lab04\_task9\_2019\_1\_60\_063():

    import assignModule as am

    df.hist(column = ["V23","V24", "V28", "V12"], bins = 5)

   #plt.title("Histogram for negative skewed, platykurtic kurtosis, positive skewed,leptokurtic kurtosis", color='darkblue')

    print("Skewness of V23 :",df['V23'].skew()) # Negative or left-skewed

    print("Kurtosis of V24 :",df['V24'].kurt()) # platykurtic kurtosis

    print("V23 Skewness using assign module: ", am.skewness(df['V23']))

    print("V24 Kurtosis using assign module: ", am.kurtosis(df['V24']))

    print("\n")

    print("Skewness of V28 :",df['V28'].skew()) #positive or right skewed

    print("Kurtosis of V12 :",df['V12'].kurt()) #Leptokurtic kurtosis

    print("V28 Skewness using assign module: ", am.skewness(df['V28']))

    print("V12 Kurtosis using assign module: ", am.kurtosis(df['V12']))

lab04\_task9\_2019\_1\_60\_063()

print("10. Find the highest positive correlation among all attributes. While finding the correlation, use appropriate code, not manually. Print this information accordingly.")

def lab04\_task10\_2019\_1\_60\_063():

    correlation= df.corr()

    #print(correlation)

    positive\_corr= correlation[correlation>0]

    st=positive\_corr.unstack()

    sort= st.sort\_values(kind="quicksort", ascending=False)

    print(sort)

    print('Highest correlation among all attributes is: ')

    result1=sort[32:33]

    print(result1)

lab04\_task10\_2019\_1\_60\_063()

print("11. Support your findings in Question 10 using a Scatter Plot.")

def lab04\_task11\_2019\_1\_60\_063():

    correlation= df.corr()

    #print(correlation)

    positive\_corr= correlation[correlation>0]

    print('As maximum correlation value from Amount and V7 so the scatter plot between them:')

    positive\_corr.plot.scatter(x='Amount',y='V7', c='magenta')

lab04\_task11\_2019\_1\_60\_063()

print("12. Find the highest negative correlation among all attributes. While finding the correlation, use appropriate code, not manually. Print this information accordingly.")

def lab04\_task12\_2019\_1\_60\_063():

    correlation= df.corr()

    #print(correlation)

    negative\_corr= correlation[correlation<0]

    st=negative\_corr.unstack()

    sort= st.sort\_values(kind="quicksort", ascending=False, na\_position='last')

    print(sort)

    print('Highest negative correlation among all attributes is: ')

    print(sort[1:2])

lab04\_task12\_2019\_1\_60\_063()

print("13. Support your findings in Question 12 using a Scatter Plot.")

def lab04\_task13\_2019\_1\_60\_063():

    correlation= df.corr()

    #print(correlation)

    negative\_corr= correlation[correlation<0]

    print('As minimum correlation value from V6 and V22 so the scatter plot between them:')

    negative\_corr.plot.scatter(x ='V6', y= 'V22', c='darkblue')

lab04\_task13\_2019\_1\_60\_063()

print("14. Create a Box Plot of the Amount Column. ")

def lab04\_task14\_2019\_1\_60\_063():

     df.boxplot(column = ['Amount'])

lab04\_task14\_2019\_1\_60\_063()

print("15. Now create two other box plots side by side. The first one will show the Amount column value for which the class value = 0 (Non-Fraudulent) and the second one will show the Amount column value for which the class value = 1 (Fraudulent). Do you find any particular pattern by just considering Amount column. Explain your answer and print it accordingly.")

def lab04\_task15\_2019\_1\_60\_063():

    import matplotlib.pyplot as plt

    import assignModule as am

    Non\_Fraudulent =df[['Amount', 'Class']].query('Class == 0')

    amountClass0 = Non\_Fraudulent['Amount']

    Fraudulent =df[['Amount', 'Class']].query('Class == 1')

    amountClass1 = Fraudulent['Amount']

    columns = [amountClass0, amountClass1]

    fig,ax = plt.subplots()

    ax.boxplot(columns)

    plt.title("Desired box plot", color='green')

    plt.show()

    print("amountClass0 Lower Quartile Q1: " ,amountClass0.quantile(0.25))

    print("amountClass0 Median Q2: ",amountClass0.median())

    print("amountClass0 Upper Quartile Q3: " ,amountClass0.quantile(0.75))

    IQR3= am.IQR(amountClass0)

    print("amountClass0 Interquartile renge(IQR) : ",IQR3)

    print("amountClass1 Lower Quartile Q1: " ,amountClass1.quantile(0.25))

    print("amountClass1 Median Q2: " ,amountClass1.median())

    print("amountClass1 Upper Quartile Q3: " ,amountClass1.quantile(0.75))

    IQR4= am.IQR(amountClass1)

    print("amountClass1 Interquartile renge(IQR) : ",IQR4)

    print("amountClass0 informations : \n")

    am.Whiskers(amountClass0,IQR3)

    print("\n")

    print("amountClass1 informations : \n")

    am.Whiskers(amountClass1,IQR4)

    print("\nSo from these two patterns we can see the 1st one which is for class 0 has lots of data upto the end. This is valid because in our dataset we see all rows are from class 0.")

    print("where the second plot 2 which is forfradulent class 1 that is almost 0. So the box plot is very small. Not even near by the 1st section. So non-fradulent and fradulent collumn for amount class has huge difference")

    print("So we can say there is almost no row for amount where class is 0 wheras almost every row for amount is of class 0.")

lab04\_task15\_2019\_1\_60\_063()

**assignModule.py**

def IQR(a):

    IQR= a.quantile(0.75)-a.quantile(0.25)

    return IQR

def Whiskers(a,IQR):

     upper\_whisker= ((a.quantile(0.75))+(IQR\*1.5))

     print("So, upper whisker : ",upper\_whisker)

     lower\_whisker= ((a.quantile(0.25))-(IQR\*1.5))

     print("So, lower whisker : ",lower\_whisker)

     print(" Min: ",a.min())

     print(" Max: ",a.max())

     if(lower\_whisker>a.min() and (upper\_whisker<a.max())):

         print("There is outliers in column as minimum and maximum value is outside of whiskers")

     else:

         print("There is no outliers in column as minimum and maximum value is inside of whiskers")

def skewness(a):

    if(a.skew()==0):

        return ("Symmetric or bell-shaped curve")

    elif(a.skew()<0):

        return("As skewness of time is less than 0,\nwe can see data distribution curve is negatively or left skewed.")

        return("Which means longer tail on the left side")

        return("There extreme values are too much and mean < median\n")

    else:

        return("As skewness of amount is greater than 0,\n we can see data distribution curve is positively or right skewed.\n")

        return("Which means longer tail on the right side")

        return("There extreme values are too much and median < mean\n")

def kurtosis(a):

    if(a.kurt()>3):

       return("As kurtosis of time is greater than 3, it is called Leptokurtosis.\n")

       return("Here the data dispersion is also very less.. almost none.\nNo extreme values")

    elif(a.kurt()<3):

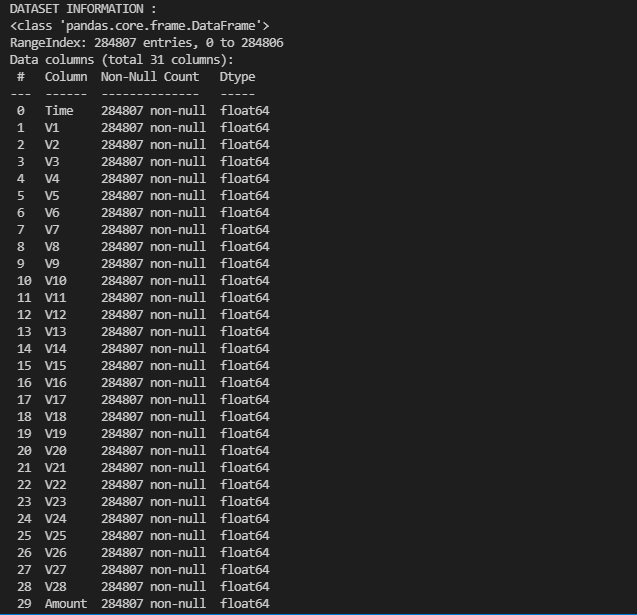
        return("As kurtosis of time is less than 3, it is called platykurtosis.\n")

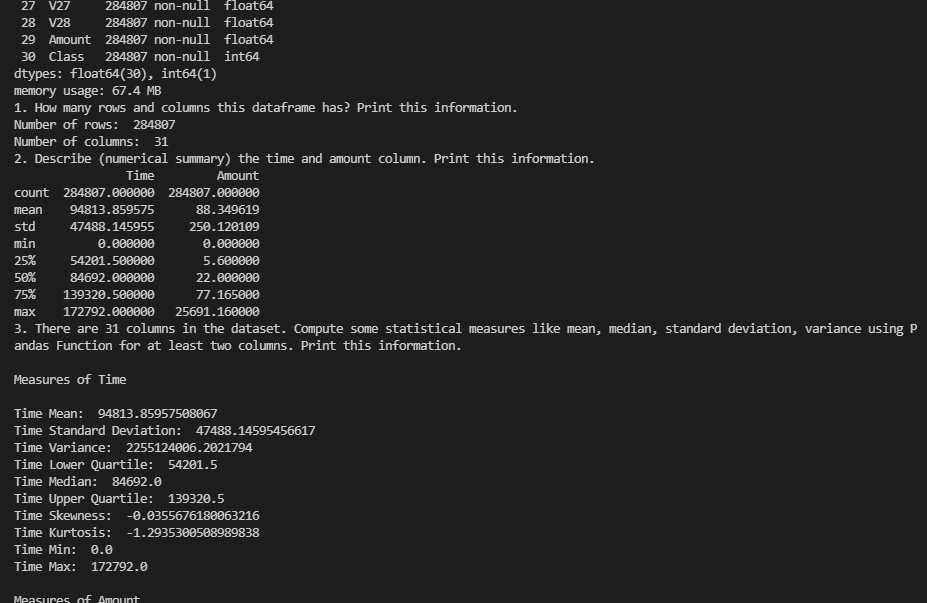
        return("Here the data dispersion is also very much")

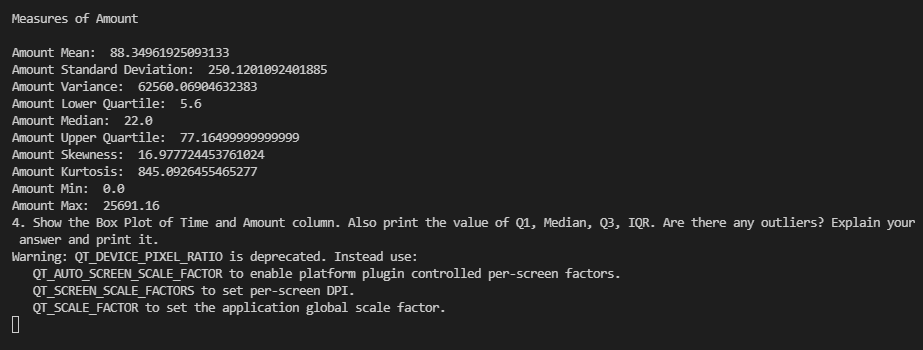
    else:

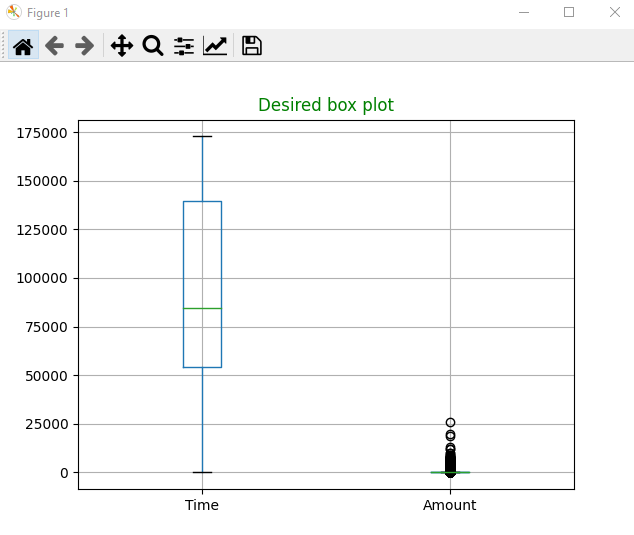
        return("As the kurtosis is 3 we can call it Mesikurtic kurtosis")

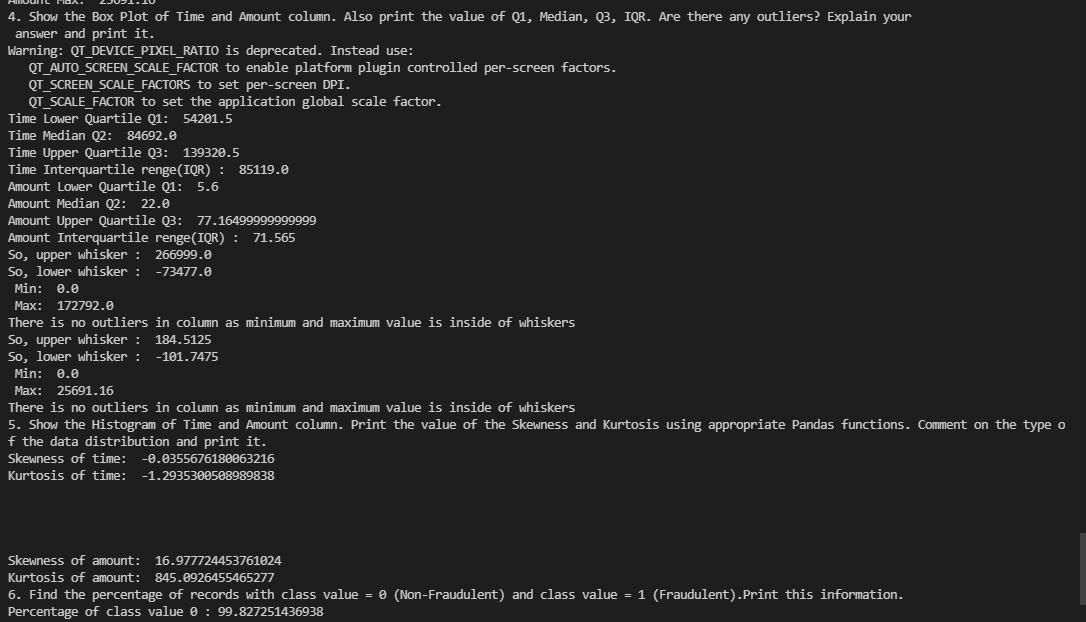
**OUTPUT**

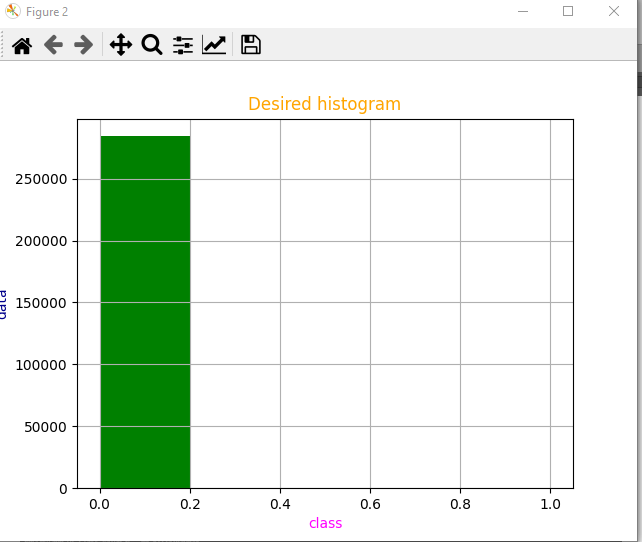
****

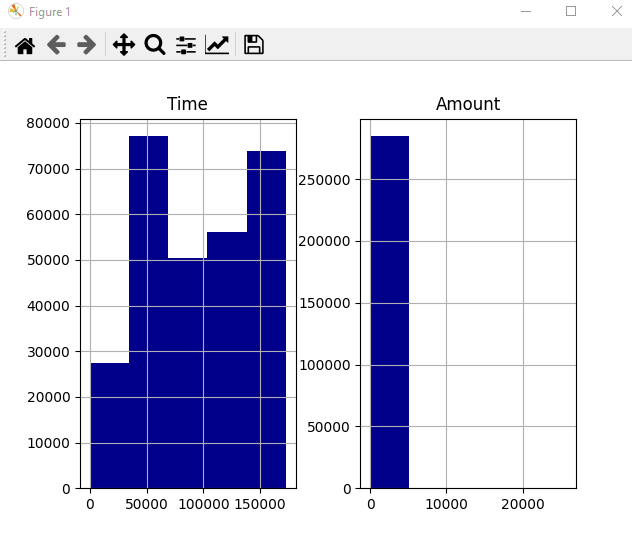
****

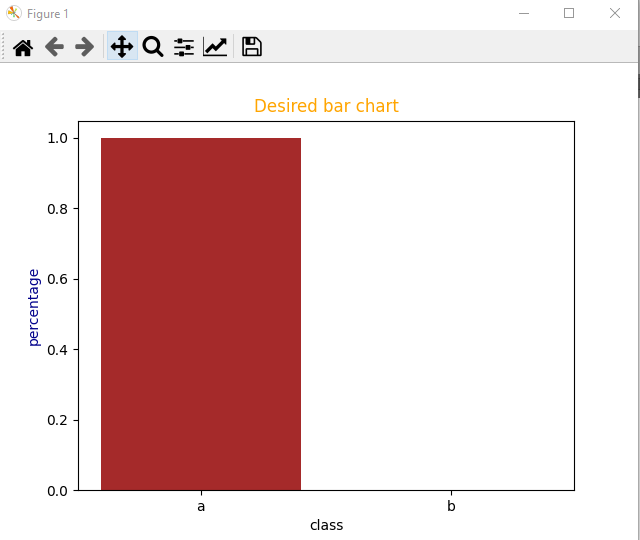
****

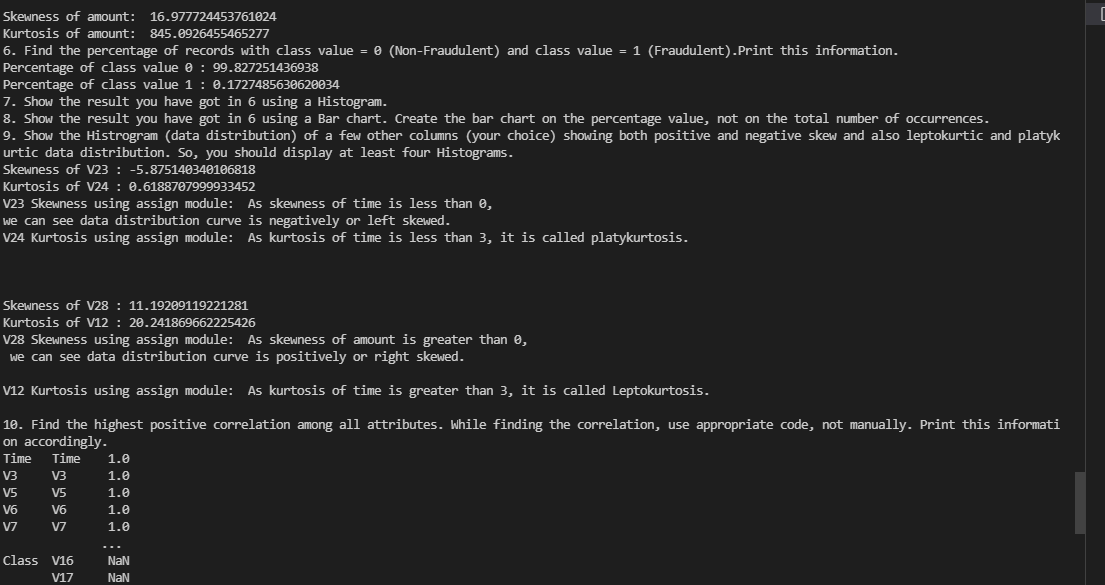
****

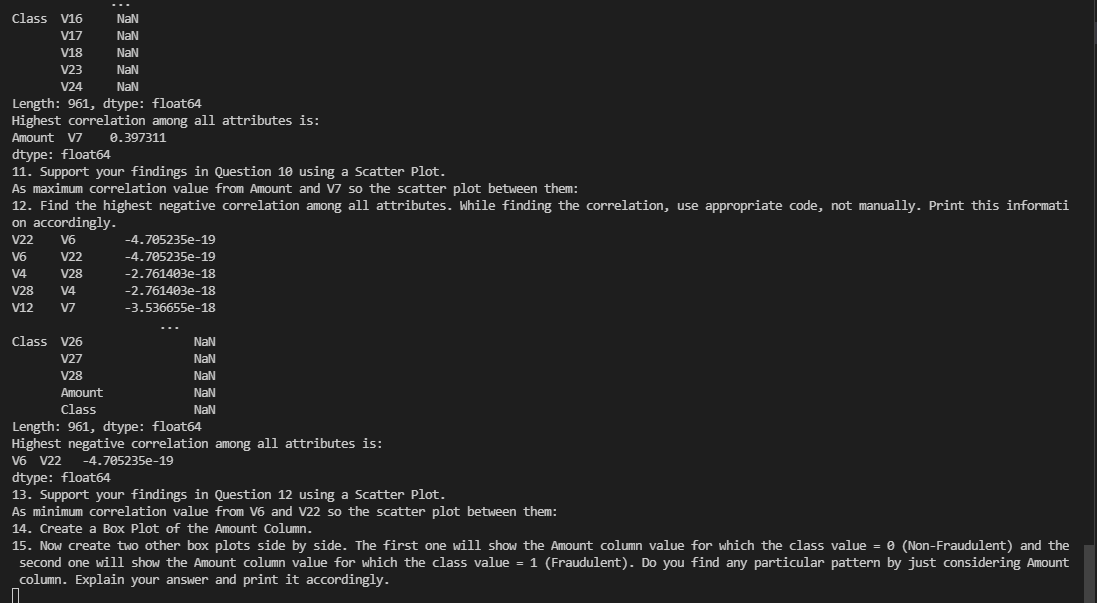
****

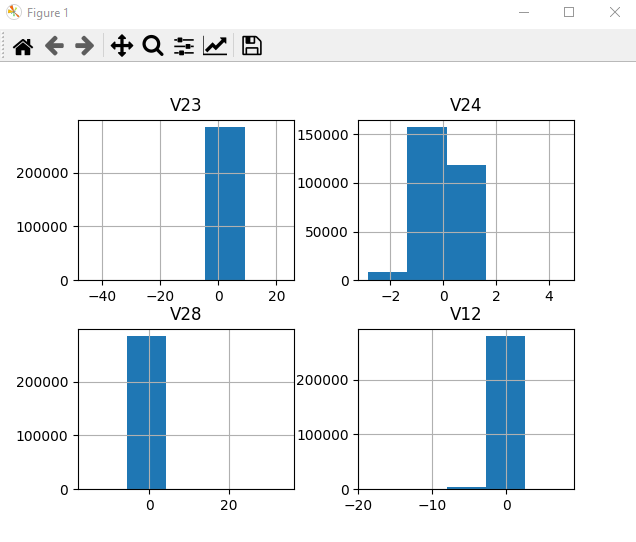
****

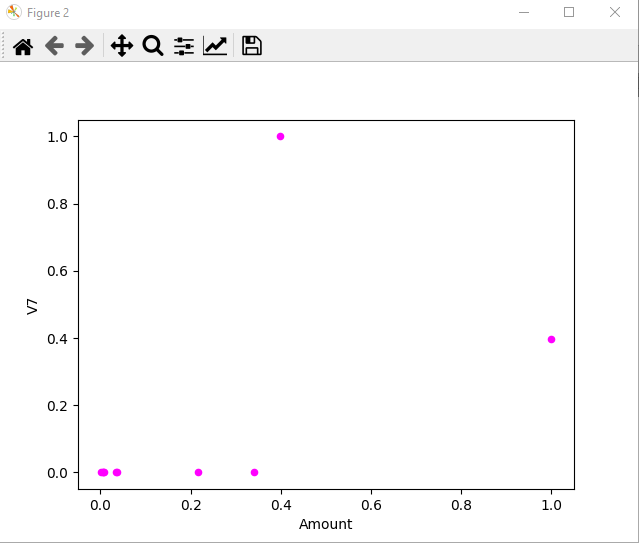
****

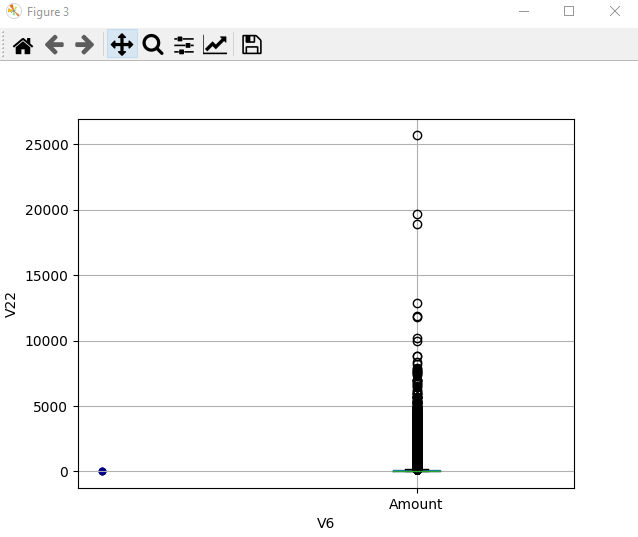
****

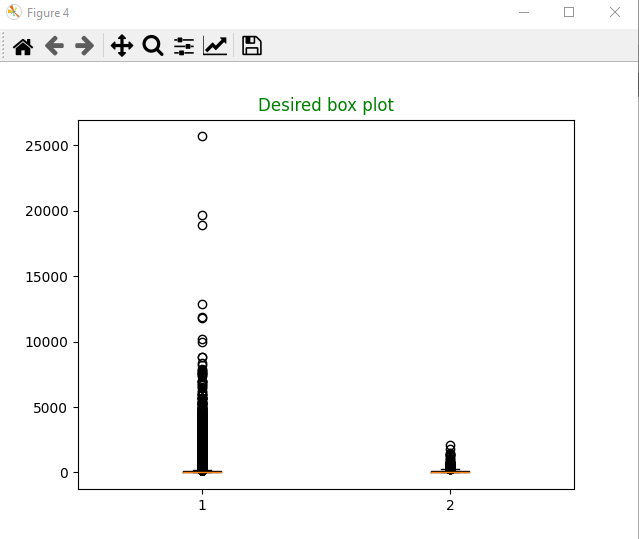
****

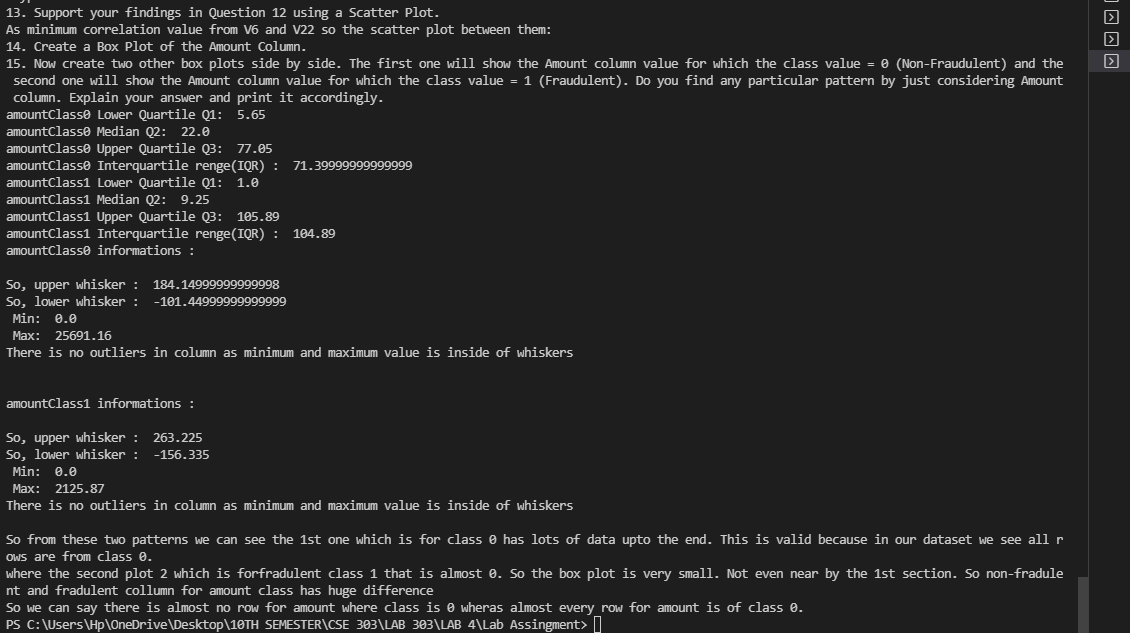
****

****

****

****

****

****