PYTHON END TERM ASSIGNMENT

TEAM MEMBER:1 ABISHANTH C

ROLL NUMBER: D21001

TEAM MEMBER2: NAMMI GAYAN PRATHYUSH

ROLL NUMBER: D21023

PROBLEM STATEMENT

Exploratory data analysis is a way to better understand your data which helps in further Data preprocessing And data visualization is key, making the exploratory data analysis process streamline and easily analyzing data using wonderful plots and charts. Creating visuals using data is a time consuming process and a routine procedure in Exploratory data analysis.Each and every time creating visuals using the same set of algorithms is a non value added action

OBJECTIVE:

To create a function, which harvest graphs by segregating numeric and categorical variables and produce the related graphs to each variable.

Why to address this problem:

In this data driven world, there is enormous data to be utilized for provinding better improvements in respectie fields. So time here is a value added asset for creating new data related works.In the process of Exploratory data analysis, creating visuals is a important step so that the user can create insights using the visuals.For every time we are doing EDA the process of creating visual by a set of algorithm consumes lots of time which can used for some other process by optimising it here.

APPROACH:

A function is defined with three arguements namely data,columns and directory.Data refers the dataset used , cols refers to columns taken into consideration and directory refers the folder where the visuals have to be saved.For each numeric variable in the data set, a histogram and boxplot is created as single image file using the subplot function and for each categorical variable a barplot is created and saved as a image file.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
```

```
def graphs(data,cols=0,directory=os.getcwd()+"\\"):        # Defining
the function by initialising default columns to zero and default
directory to working directory
    c=0
```

```
    catdat=[]                                                    #
Initializing count and lists to store categorical data and numerical
data
    numdat=[]
    data_types=[]
    if cols==0:
        col=list(data.columns)                                  # If the
columns are not given, then all the columns of the dataset
        data_types=data.dtypes                                  # Storing
the datatypes of the columns in a list
        for i in data_types:
            if i=="object":
                catdat.append(col[c])                            # Storing
all the categorical variables in a list
            else:
                numdat.append(col[c])                            # Storing
all the numerical variables in another list
            c+=1
    else:
        for i in cols:
            data_types.append(data[i].dtypes)                   # If the
columns are then only those columns will be considered
        for i in data_types:
            if i=='O':
                catdat.append(cols[c])                           # Storing
all the categorical variables in a list
            else:
                numdat.append(cols[c])                           # Storing
all the numerical variables in another list
            c=c+1
    for i in numdat:
        fig=plt.figure(figsize=(18,18))                                  #
Setting the figure size
        plt.subplot(2, 1, 1)                                             #
Using subplot function to plot more than 1 graph in a png file
        plt.hist(data[i])                                               #
Plotting a histogram for the respective numerical variable
        plt.xlabel(i,fontsize=26)
        plt.ylabel("Frequency",fontsize=26)                              #
Putting xlabel, ylabel and title for the graph
        plt.title("HISTOGRAM ON %s" %(i.upper()),fontsize=24)
        plt.tick_params(axis='x', labelsize=24)
        plt.tick_params(axis='y', labelsize=24)
        plt.subplot(2,1,2)                                              #
Adding another chart into the same subplot
        plt.boxplot(data[i],vert=False)                                 #
Plotting a boxplot for the respective categorical variable
        plt.title("BOXPLOT ON %s" %(i.upper()),fontsize=24)             #
Putting title for the graph
        plt.tick_params(axis='x', labelsize=24)
```

```
        plt.tick_params(axis='y', labelsize=24)
        plt.savefig(directory+i.upper()+'.png')                    #
Saving the file in the respective directory in '.png' format
        plt.show()
    for j in catdat:

data[j].value_counts().plot(kind="bar",figsize=(16,16),color="coral",f
ontsize=12)    # Plotting a bar chart for the respective categorical
variable
        plt.xlabel(j,fontsize=26)
        plt.ylabel("Frequency",fontsize=26)
# Putting xlabel, ylabel and title for the chart
        plt.title("BARPLOT ON %s" %(j.upper()),fontsize=24)
        plt.tick_params(axis='x', labelsize=24)
        plt.tick_params(axis='y', labelsize=24)
        plt.savefig(directory+j.upper()+'.png')
# Saving the file in the respective directory in '.png' format
        plt.show()

cars=pd.read_csv('cars.csv')

graphs(cars)
```

REPORT:

A function graphs was created which reads the data given by the user ,segregates the columns as numeric and categorical, creates a historgram and box plot for each numeric variable and a bar plot for each categorical variable.If the user wants only certain columns to be plotted then also the fuction works as per user input.Using this function on any dataset that the user has and wants to create visuals for it,the user can create the visuals with one line of code rather coding a entire algorithm which saves lots of time which can be used later to draw useful insights from the dataset.

Youtube links:

Abishanth: https://youtu.be/i26gtz2YUL4

Prathyush: https://youtu.be/jZ0BuHqb2gA