# import the required libraries  
  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns

df = pd.read\_csv("/content/customer\_churn (4).csv")

df

customerID gender SeniorCitizen Partner Dependents tenure \  
0 7590-VHVEG Female 0 Yes No 1   
1 5575-GNVDE Male 0 No No 34   
2 3668-QPYBK Male 0 No No 2   
3 7795-CFOCW Male 0 No No 45   
4 9237-HQITU Female 0 No No 2   
... ... ... ... ... ... ...   
7038 6840-RESVB Male 0 Yes Yes 24   
7039 2234-XADUH Female 0 Yes Yes 72   
7040 4801-JZAZL Female 0 Yes Yes 11   
7041 8361-LTMKD Male 1 Yes No 4   
7042 3186-AJIEK Male 0 No No 66   
  
 PhoneService MultipleLines InternetService OnlineSecurity ... \  
0 No No phone service DSL No ...   
1 Yes No DSL Yes ...   
2 Yes No DSL Yes ...   
3 No No phone service DSL Yes ...   
4 Yes No Fiber optic No ...   
... ... ... ... ... ...   
7038 Yes Yes DSL Yes ...   
7039 Yes Yes Fiber optic No ...   
7040 No No phone service DSL Yes ...   
7041 Yes Yes Fiber optic No ...   
7042 Yes No Fiber optic Yes ...   
  
 DeviceProtection TechSupport StreamingTV StreamingMovies Contract \  
0 No No No No Month-to-month   
1 Yes No No No One year   
2 No No No No Month-to-month   
3 Yes Yes No No One year   
4 No No No No Month-to-month   
... ... ... ... ... ...   
7038 Yes Yes Yes Yes One year   
7039 Yes No Yes Yes One year   
7040 No No No No Month-to-month   
7041 No No No No Month-to-month   
7042 Yes Yes Yes Yes Two year   
  
 PaperlessBilling PaymentMethod MonthlyCharges TotalCharges \  
0 Yes Electronic check 29.85 29.85   
1 No Mailed check 56.95 1889.5   
2 Yes Mailed check 53.85 108.15   
3 No Bank transfer (automatic) 42.30 1840.75   
4 Yes Electronic check 70.70 151.65   
... ... ... ... ...   
7038 Yes Mailed check 84.80 1990.5   
7039 Yes Credit card (automatic) 103.20 7362.9   
7040 Yes Electronic check 29.60 346.45   
7041 Yes Mailed check 74.40 306.6   
7042 Yes Bank transfer (automatic) 105.65 6844.5   
  
 Churn   
0 No   
1 No   
2 Yes   
3 No   
4 Yes   
... ...   
7038 No   
7039 No   
7040 No   
7041 Yes   
7042 No   
  
[7043 rows x 21 columns]

# EDA: Exploratory data Analysis  
  
df.head() #returns the first 5 records

customerID gender SeniorCitizen Partner Dependents tenure PhoneService \  
0 7590-VHVEG Female 0 Yes No 1 No   
1 5575-GNVDE Male 0 No No 34 Yes   
2 3668-QPYBK Male 0 No No 2 Yes   
3 7795-CFOCW Male 0 No No 45 No   
4 9237-HQITU Female 0 No No 2 Yes   
  
 MultipleLines InternetService OnlineSecurity ... DeviceProtection \  
0 No phone service DSL No ... No   
1 No DSL Yes ... Yes   
2 No DSL Yes ... No   
3 No phone service DSL Yes ... Yes   
4 No Fiber optic No ... No   
  
 TechSupport StreamingTV StreamingMovies Contract PaperlessBilling \  
0 No No No Month-to-month Yes   
1 No No No One year No   
2 No No No Month-to-month Yes   
3 Yes No No One year No   
4 No No No Month-to-month Yes   
  
 PaymentMethod MonthlyCharges TotalCharges Churn   
0 Electronic check 29.85 29.85 No   
1 Mailed check 56.95 1889.5 No   
2 Mailed check 53.85 108.15 Yes   
3 Bank transfer (automatic) 42.30 1840.75 No   
4 Electronic check 70.70 151.65 Yes   
  
[5 rows x 21 columns]

df.tail() #returns last 5 records

customerID gender SeniorCitizen Partner Dependents tenure \  
7038 6840-RESVB Male 0 Yes Yes 24   
7039 2234-XADUH Female 0 Yes Yes 72   
7040 4801-JZAZL Female 0 Yes Yes 11   
7041 8361-LTMKD Male 1 Yes No 4   
7042 3186-AJIEK Male 0 No No 66   
  
 PhoneService MultipleLines InternetService OnlineSecurity ... \  
7038 Yes Yes DSL Yes ...   
7039 Yes Yes Fiber optic No ...   
7040 No No phone service DSL Yes ...   
7041 Yes Yes Fiber optic No ...   
7042 Yes No Fiber optic Yes ...   
  
 DeviceProtection TechSupport StreamingTV StreamingMovies Contract \  
7038 Yes Yes Yes Yes One year   
7039 Yes No Yes Yes One year   
7040 No No No No Month-to-month   
7041 No No No No Month-to-month   
7042 Yes Yes Yes Yes Two year   
  
 PaperlessBilling PaymentMethod MonthlyCharges TotalCharges \  
7038 Yes Mailed check 84.80 1990.5   
7039 Yes Credit card (automatic) 103.20 7362.9   
7040 Yes Electronic check 29.60 346.45   
7041 Yes Mailed check 74.40 306.6   
7042 Yes Bank transfer (automatic) 105.65 6844.5   
  
 Churn   
7038 No   
7039 No   
7040 No   
7041 Yes   
7042 No   
  
[5 rows x 21 columns]

df.shape #returns the total rows and columns present in the data

(7043, 21)

df.info() #returns the entire information of the data

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 7043 entries, 0 to 7042  
Data columns (total 21 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 customerID 7043 non-null object   
 1 gender 7043 non-null object   
 2 SeniorCitizen 7043 non-null int64   
 3 Partner 7043 non-null object   
 4 Dependents 7043 non-null object   
 5 tenure 7043 non-null int64   
 6 PhoneService 7043 non-null object   
 7 MultipleLines 7043 non-null object   
 8 InternetService 7043 non-null object   
 9 OnlineSecurity 7043 non-null object   
 10 OnlineBackup 7043 non-null object   
 11 DeviceProtection 7043 non-null object   
 12 TechSupport 7043 non-null object   
 13 StreamingTV 7043 non-null object   
 14 StreamingMovies 7043 non-null object   
 15 Contract 7043 non-null object   
 16 PaperlessBilling 7043 non-null object   
 17 PaymentMethod 7043 non-null object   
 18 MonthlyCharges 7043 non-null float64  
 19 TotalCharges 7043 non-null object   
 20 Churn 7043 non-null object   
dtypes: float64(1), int64(2), object(18)  
memory usage: 1.1+ MB

df.describe()

SeniorCitizen tenure MonthlyCharges  
count 7043.000000 7043.000000 7043.000000  
mean 0.162147 32.371149 64.761692  
std 0.368612 24.559481 30.090047  
min 0.000000 0.000000 18.250000  
25% 0.000000 9.000000 35.500000  
50% 0.000000 29.000000 70.350000  
75% 0.000000 55.000000 89.850000  
max 1.000000 72.000000 118.750000

df.describe(include="all")

customerID gender SeniorCitizen Partner Dependents tenure \  
count 7043 7043 7043.000000 7043 7043 7043.000000   
unique 7043 2 NaN 2 2 NaN   
top 7590-VHVEG Male NaN No No NaN   
freq 1 3555 NaN 3641 4933 NaN   
mean NaN NaN 0.162147 NaN NaN 32.371149   
std NaN NaN 0.368612 NaN NaN 24.559481   
min NaN NaN 0.000000 NaN NaN 0.000000   
25% NaN NaN 0.000000 NaN NaN 9.000000   
50% NaN NaN 0.000000 NaN NaN 29.000000   
75% NaN NaN 0.000000 NaN NaN 55.000000   
max NaN NaN 1.000000 NaN NaN 72.000000   
  
 PhoneService MultipleLines InternetService OnlineSecurity ... \  
count 7043 7043 7043 7043 ...   
unique 2 3 3 3 ...   
top Yes No Fiber optic No ...   
freq 6361 3390 3096 3498 ...   
mean NaN NaN NaN NaN ...   
std NaN NaN NaN NaN ...   
min NaN NaN NaN NaN ...   
25% NaN NaN NaN NaN ...   
50% NaN NaN NaN NaN ...   
75% NaN NaN NaN NaN ...   
max NaN NaN NaN NaN ...   
  
 DeviceProtection TechSupport StreamingTV StreamingMovies \  
count 7043 7043 7043 7043   
unique 3 3 3 3   
top No No No No   
freq 3095 3473 2810 2785   
mean NaN NaN NaN NaN   
std NaN NaN NaN NaN   
min NaN NaN NaN NaN   
25% NaN NaN NaN NaN   
50% NaN NaN NaN NaN   
75% NaN NaN NaN NaN   
max NaN NaN NaN NaN   
  
 Contract PaperlessBilling PaymentMethod MonthlyCharges \  
count 7043 7043 7043 7043.000000   
unique 3 2 4 NaN   
top Month-to-month Yes Electronic check NaN   
freq 3875 4171 2365 NaN   
mean NaN NaN NaN 64.761692   
std NaN NaN NaN 30.090047   
min NaN NaN NaN 18.250000   
25% NaN NaN NaN 35.500000   
50% NaN NaN NaN 70.350000   
75% NaN NaN NaN 89.850000   
max NaN NaN NaN 118.750000   
  
 TotalCharges Churn   
count 7043 7043   
unique 6531 2   
top No   
freq 11 5174   
mean NaN NaN   
std NaN NaN   
min NaN NaN   
25% NaN NaN   
50% NaN NaN   
75% NaN NaN   
max NaN NaN   
  
[11 rows x 21 columns]

df.isnull()

customerID gender SeniorCitizen Partner Dependents tenure \  
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   
... ... ... ... ... ... ...   
7038 False False False False False False   
7039 False False False False False False   
7040 False False False False False False   
7041 False False False False False False   
7042 False False False False False False   
  
 PhoneService MultipleLines InternetService OnlineSecurity ... \  
0 False False False False ...   
1 False False False False ...   
2 False False False False ...   
3 False False False False ...   
4 False False False False ...   
... ... ... ... ... ...   
7038 False False False False ...   
7039 False False False False ...   
7040 False False False False ...   
7041 False False False False ...   
7042 False False False False ...   
  
 DeviceProtection TechSupport StreamingTV StreamingMovies Contract \  
0 False False False False False   
1 False False False False False   
2 False False False False False   
3 False False False False False   
4 False False False False False   
... ... ... ... ... ...   
7038 False False False False False   
7039 False False False False False   
7040 False False False False False   
7041 False False False False False   
7042 False False False False False   
  
 PaperlessBilling PaymentMethod MonthlyCharges TotalCharges Churn   
0 False False False False False   
1 False False False False False   
2 False False False False False   
3 False False False False False   
4 False False False False False   
... ... ... ... ... ...   
7038 False False False False False   
7039 False False False False False   
7040 False False False False False   
7041 False False False False False   
7042 False False False False False   
  
[7043 rows x 21 columns]

df.isna().sum() #returns the null values in the data

customerID 0  
gender 0  
SeniorCitizen 0  
Partner 0  
Dependents 0  
tenure 0  
PhoneService 0  
MultipleLines 0  
InternetService 0  
OnlineSecurity 0  
OnlineBackup 0  
DeviceProtection 0  
TechSupport 0  
StreamingTV 0  
StreamingMovies 0  
Contract 0  
PaperlessBilling 0  
PaymentMethod 0  
MonthlyCharges 0  
TotalCharges 0  
Churn 0  
dtype: int64

df.count()

customerID 7043  
gender 7043  
SeniorCitizen 7043  
Partner 7043  
Dependents 7043  
tenure 7043  
PhoneService 7043  
MultipleLines 7043  
InternetService 7043  
OnlineSecurity 7043  
OnlineBackup 7043  
DeviceProtection 7043  
TechSupport 7043  
StreamingTV 7043  
StreamingMovies 7043  
Contract 7043  
PaperlessBilling 7043  
PaymentMethod 7043  
MonthlyCharges 7043  
TotalCharges 7043  
Churn 7043  
dtype: int64

df.TechSupport.value\_counts() #returns the counts of the unique values present in the columns

No 3473  
Yes 2044  
No internet service 1526  
Name: TechSupport, dtype: int64

**A) Data Manipulation:**

**a. Extract the 5th column & store it in ‘customer\_5’**

customer\_5 = df.iloc[:,4]

print(customer\_5)

0 No  
1 No  
2 No  
3 No  
4 No  
 ...   
7038 Yes  
7039 Yes  
7040 Yes  
7041 No  
7042 No  
Name: Dependents, Length: 7043, dtype: object

customer\_5 = df.loc[:,["Dependents"]]

print(customer\_5)

Dependents  
0 No  
1 No  
2 No  
3 No  
4 No  
... ...  
7038 Yes  
7039 Yes  
7040 Yes  
7041 No  
7042 No  
  
[7043 rows x 1 columns]

customer\_5 = df.Dependents

customer\_5

0 No  
1 No  
2 No  
3 No  
4 No  
 ...   
7038 Yes  
7039 Yes  
7040 Yes  
7041 No  
7042 No  
Name: Dependents, Length: 7043, dtype: object

**b. Extract the 15th column & store it in ‘customer\_15’**

customer\_15 = df.iloc[:,14]

print(customer\_15)

0 No  
1 No  
2 No  
3 No  
4 No  
 ...   
7038 Yes  
7039 Yes  
7040 No  
7041 No  
7042 Yes  
Name: StreamingMovies, Length: 7043, dtype: object

customer\_15 = df.loc[:,["StreamingMovies"]]

print(customer\_15)

StreamingMovies  
0 No  
1 No  
2 No  
3 No  
4 No  
... ...  
7038 Yes  
7039 Yes  
7040 No  
7041 No  
7042 Yes  
  
[7043 rows x 1 columns]

customer\_15 = df.StreamingMovies

customer\_15

0 No  
1 No  
2 No  
3 No  
4 No  
 ...   
7038 Yes  
7039 Yes  
7040 No  
7041 No  
7042 Yes  
Name: StreamingMovies, Length: 7043, dtype: object

**c. Extract all the male senior citizens whose Payment Method is Electronic check & store the result in ‘senior\_male\_electronic’**

df.gender.value\_counts()

Male 3555  
Female 3488  
Name: gender, dtype: int64

df.SeniorCitizen.value\_counts()

0 5901  
1 1142  
Name: SeniorCitizen, dtype: int64

df.PaymentMethod.unique()

array(['Electronic check', 'Mailed check', 'Bank transfer (automatic)',  
 'Credit card (automatic)'], dtype=object)

df.PaymentMethod.value\_counts()

Electronic check 2365  
Mailed check 1612  
Bank transfer (automatic) 1544  
Credit card (automatic) 1522  
Name: PaymentMethod, dtype: int64

senior\_male\_electronic = df[(df["gender"]=="Male") & (df["SeniorCitizen"]==1) & (df["PaymentMethod"]=="Electronic check")]

senior\_male\_electronic

customerID gender SeniorCitizen Partner Dependents tenure \  
20 8779-QRDMV Male 1 No No 1   
55 1658-BYGOY Male 1 No No 18   
57 5067-XJQFU Male 1 Yes Yes 66   
78 0191-ZHSKZ Male 1 No No 30   
91 2424-WVHPL Male 1 No No 1   
... ... ... ... ... ... ...   
6837 6229-LSCKB Male 1 No No 6   
6894 1400-MMYXY Male 1 Yes No 3   
6914 7142-HVGBG Male 1 Yes No 43   
6967 8739-WWKDU Male 1 No No 25   
7032 6894-LFHLY Male 1 No No 1   
  
 PhoneService MultipleLines InternetService OnlineSecurity ... \  
20 No No phone service DSL No ...   
55 Yes Yes Fiber optic No ...   
57 Yes Yes Fiber optic No ...   
78 Yes No DSL Yes ...   
91 Yes No Fiber optic No ...   
... ... ... ... ... ...   
6837 Yes No Fiber optic No ...   
6894 Yes Yes Fiber optic No ...   
6914 Yes Yes Fiber optic No ...   
6967 Yes Yes Fiber optic No ...   
7032 Yes Yes Fiber optic No ...   
  
 DeviceProtection TechSupport StreamingTV StreamingMovies Contract \  
20 Yes No No Yes Month-to-month   
55 No No Yes Yes Month-to-month   
57 Yes Yes Yes Yes One year   
78 No No Yes Yes Month-to-month   
91 No Yes No No Month-to-month   
... ... ... ... ... ...   
6837 No No Yes No Month-to-month   
6894 Yes No Yes Yes Month-to-month   
6914 Yes No Yes Yes Month-to-month   
6967 No No Yes No Month-to-month   
7032 No No No No Month-to-month   
  
 PaperlessBilling PaymentMethod MonthlyCharges TotalCharges Churn   
20 Yes Electronic check 39.65 39.65 Yes   
55 Yes Electronic check 95.45 1752.55 Yes   
57 Yes Electronic check 108.45 7076.35 No   
78 Yes Electronic check 74.75 2111.3 No   
91 No Electronic check 74.70 74.7 No   
... ... ... ... ... ...   
6837 Yes Electronic check 79.70 497.6 No   
6894 Yes Electronic check 105.90 334.65 Yes   
6914 Yes Electronic check 103.00 4414.3 Yes   
6967 Yes Electronic check 89.50 2196.15 Yes   
7032 Yes Electronic check 75.75 75.75 Yes   
  
[298 rows x 21 columns]

**d. Extract all those customers whose tenure is greater than 70 months or their Monthly charges is more than 100$ & store the result in ‘customer\_total\_tenure’**

customer\_total\_tenure = df[(df["tenure"]>70) | (df["MonthlyCharges"]>100)]

customer\_total\_tenure

customerID gender SeniorCitizen Partner Dependents tenure \  
8 7892-POOKP Female 0 Yes No 28   
12 8091-TTVAX Male 0 Yes No 58   
13 0280-XJGEX Male 0 No No 49   
14 5129-JLPIS Male 0 No No 25   
15 3655-SNQYZ Female 0 Yes Yes 69   
... ... ... ... ... ... ...   
7023 1035-IPQPU Female 1 Yes No 63   
7034 0639-TSIQW Female 0 No No 67   
7037 2569-WGERO Female 0 No No 72   
7039 2234-XADUH Female 0 Yes Yes 72   
7042 3186-AJIEK Male 0 No No 66   
  
 PhoneService MultipleLines InternetService OnlineSecurity ... \  
8 Yes Yes Fiber optic No ...   
12 Yes Yes Fiber optic No ...   
13 Yes Yes Fiber optic No ...   
14 Yes No Fiber optic Yes ...   
15 Yes Yes Fiber optic Yes ...   
... ... ... ... ... ...   
7023 Yes Yes Fiber optic No ...   
7034 Yes Yes Fiber optic Yes ...   
7037 Yes No No No internet service ...   
7039 Yes Yes Fiber optic No ...   
7042 Yes No Fiber optic Yes ...   
  
 DeviceProtection TechSupport StreamingTV \  
8 Yes Yes Yes   
12 Yes No Yes   
13 Yes No Yes   
14 Yes Yes Yes   
15 Yes Yes Yes   
... ... ... ...   
7023 Yes No Yes   
7034 Yes No Yes   
7037 No internet service No internet service No internet service   
7039 Yes No Yes   
7042 Yes Yes Yes   
  
 StreamingMovies Contract PaperlessBilling \  
8 Yes Month-to-month Yes   
12 Yes One year No   
13 Yes Month-to-month Yes   
14 Yes Month-to-month Yes   
15 Yes Two year No   
... ... ... ...   
7023 Yes Month-to-month Yes   
7034 No Month-to-month Yes   
7037 No internet service Two year Yes   
7039 Yes One year Yes   
7042 Yes Two year Yes   
  
 PaymentMethod MonthlyCharges TotalCharges Churn   
8 Electronic check 104.80 3046.05 Yes   
12 Credit card (automatic) 100.35 5681.1 No   
13 Bank transfer (automatic) 103.70 5036.3 Yes   
14 Electronic check 105.50 2686.05 No   
15 Credit card (automatic) 113.25 7895.15 No   
... ... ... ... ...   
7023 Electronic check 103.50 6479.4 No   
7034 Credit card (automatic) 102.95 6886.25 Yes   
7037 Bank transfer (automatic) 21.15 1419.4 No   
7039 Credit card (automatic) 103.20 7362.9 No   
7042 Bank transfer (automatic) 105.65 6844.5 No   
  
[1259 rows x 21 columns]

**e. Extract all the customers whose Contract is of two years, payment method is Mailed check & the value of Churn is ‘Yes’ & store the result in ‘two\_mail\_yes’**

two\_mail\_yes = df[(df["Contract"]=="Two year") & (df["PaymentMethod"]=="Mailed check") & (df["Churn"]=="Yes")]

two\_mail\_yes

customerID gender SeniorCitizen Partner Dependents tenure \  
268 6323-AYBRX Male 0 No No 59   
5947 7951-QKZPL Female 0 Yes Yes 33   
6680 9412-ARGBX Female 0 No Yes 48   
  
 PhoneService MultipleLines InternetService OnlineSecurity ... \  
268 Yes No No No internet service ...   
5947 Yes Yes No No internet service ...   
6680 Yes No Fiber optic No ...   
  
 DeviceProtection TechSupport StreamingTV \  
268 No internet service No internet service No internet service   
5947 No internet service No internet service No internet service   
6680 Yes Yes Yes   
  
 StreamingMovies Contract PaperlessBilling PaymentMethod \  
268 No internet service Two year No Mailed check   
5947 No internet service Two year Yes Mailed check   
6680 No Two year Yes Mailed check   
  
 MonthlyCharges TotalCharges Churn   
268 19.35 1099.6 Yes   
5947 24.50 740.3 Yes   
6680 95.50 4627.85 Yes   
  
[3 rows x 21 columns]

**f. Extract 333 random records from the customer\_churndataframe& store the result in ‘customer\_333’**

customer\_333 = df.sample(n=333)

customer\_333

customerID gender SeniorCitizen Partner Dependents tenure \  
1674 2808-CHTDM Female 0 Yes Yes 23   
4565 1866-DIOQZ Female 0 Yes No 71   
3980 8868-WOZGU Male 0 No No 28   
5528 9127-QRZMH Male 0 Yes No 44   
4301 7729-XBTWX Male 0 Yes Yes 66   
... ... ... ... ... ... ...   
2106 4547-FZJWE Male 0 Yes Yes 55   
5439 5959-BELXA Male 1 No No 32   
4131 2876-VBBBL Female 0 No No 1   
5778 3090-QFUVD Female 0 No No 1   
3906 6372-RFVNS Female 0 Yes No 1   
  
 PhoneService MultipleLines InternetService OnlineSecurity ... \  
1674 Yes Yes No No internet service ...   
4565 No No phone service DSL Yes ...   
3980 Yes Yes Fiber optic No ...   
5528 Yes No DSL Yes ...   
4301 Yes No No No internet service ...   
... ... ... ... ... ...   
2106 Yes No DSL Yes ...   
5439 Yes Yes Fiber optic No ...   
4131 Yes No No No internet service ...   
5778 Yes No No No internet service ...   
3906 Yes No Fiber optic No ...   
  
 DeviceProtection TechSupport StreamingTV \  
1674 No internet service No internet service No internet service   
4565 Yes Yes Yes   
3980 Yes No Yes   
5528 No Yes No   
4301 No internet service No internet service No internet service   
... ... ... ...   
2106 No No No   
5439 No No Yes   
4131 No internet service No internet service No internet service   
5778 No internet service No internet service No internet service   
3906 No No Yes   
  
 StreamingMovies Contract PaperlessBilling \  
1674 No internet service Two year No   
4565 Yes One year No   
3980 Yes Month-to-month Yes   
5528 No One year Yes   
4301 No internet service Two year Yes   
... ... ... ...   
2106 Yes One year No   
5439 Yes Month-to-month Yes   
4131 No internet service Month-to-month Yes   
5778 No internet service Month-to-month No   
3906 No Month-to-month No   
  
 PaymentMethod MonthlyCharges TotalCharges Churn   
1674 Mailed check 25.10 611.45 No   
4565 Bank transfer (automatic) 66.80 4689.15 No   
3980 Electronic check 105.70 2979.5 Yes   
5528 Bank transfer (automatic) 59.85 2603.95 No   
4301 Mailed check 19.75 1344.5 No   
... ... ... ... ...   
2106 Credit card (automatic) 59.45 3157 No   
5439 Credit card (automatic) 96.15 3019.25 Yes   
4131 Mailed check 20.25 20.25 Yes   
5778 Credit card (automatic) 20.30 20.3 No   
3906 Electronic check 78.80 78.8 Yes   
  
[333 rows x 21 columns]

**g. Get the count of different levels from the ‘Churn’ column**

df["Churn"].unique()

array(['No', 'Yes'], dtype=object)

df["Churn"].value\_counts()

No 5174  
Yes 1869  
Name: Churn, dtype: int64

**B) Data Visualization:**

a. Build a bar-plot for the ’InternetService’ column:

i. Set x-axis label to ‘Categories of Internet Service’.

ii. Set y-axis label to ‘Count of Categories’.

iii. Set the title of plot to be ‘Distribution of Internet Service’.

iv. Set the color of the bars to be ‘orange’.

df["InternetService"].value\_counts()

Fiber optic 3096  
DSL 2421  
No 1526  
Name: InternetService, dtype: int64

x = df["InternetService"].value\_counts().keys().tolist()

x

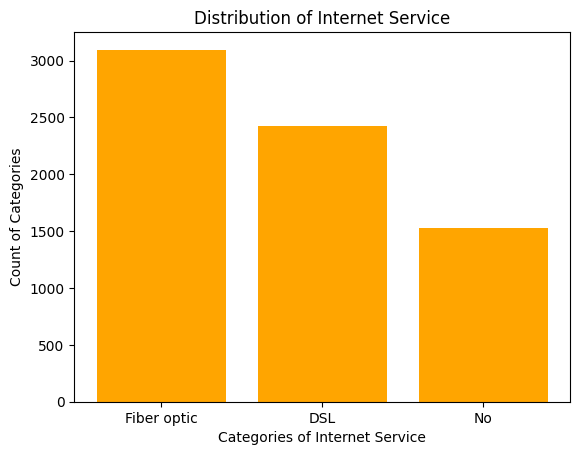
['Fiber optic', 'DSL', 'No']

y = df["InternetService"].value\_counts().tolist()

y

[3096, 2421, 1526]

plt.bar(x,y,color="orange")  
plt.xlabel("Categories of Internet Service")  
plt.ylabel("Count of Categories")  
plt.title("Distribution of Internet Service")  
plt.show()



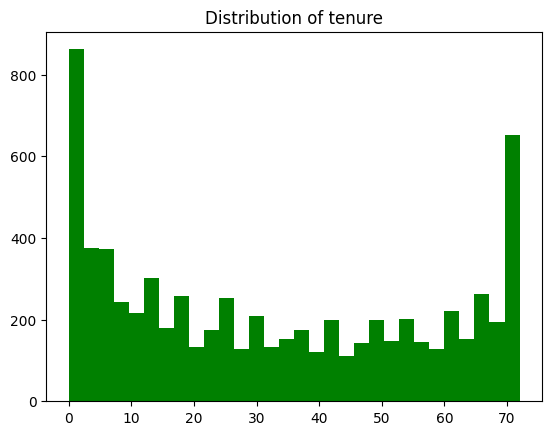
b. Build a histogram for the ‘tenure’ column:

i. Set the number of bins to be 30

ii. Set the color of the bins to be ‘green’

iii. Assign the title ‘Distribution of tenure’

plt.hist(df["tenure"],bins=30,color="green")  
plt.title("Distribution of tenure")  
plt.show()



c. Build a scatter-plot between ‘MonthlyCharges’ & ‘tenure’. Map ‘MonthlyCharges’ to the y-axis & ‘tenure’ to the ‘x-axis’:

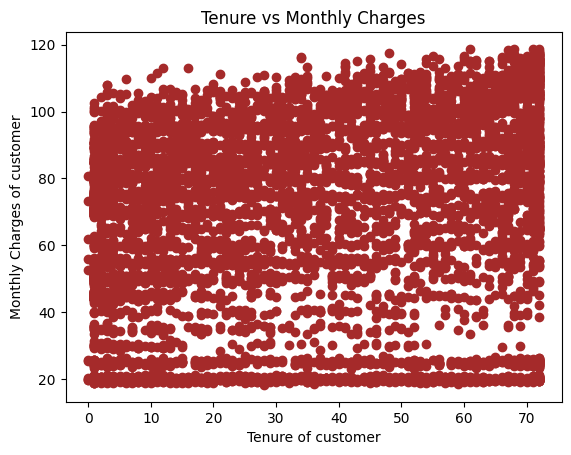
i. Assign the points a color of ‘brown’

ii. Set the x-axis label to ‘Tenure of customer’

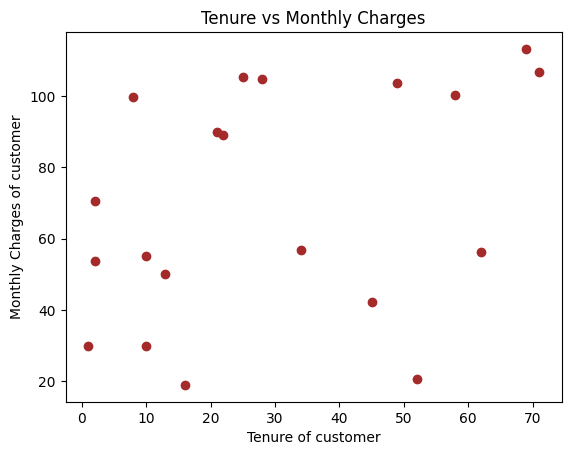
iii. Set the y-axis label to ‘Monthly Charges of customer’

iv. Set the title to ‘Tenure vs Monthly Charges’

plt.scatter(x=df["tenure"],y=df["MonthlyCharges"],color="brown")  
plt.xlabel("Tenure of customer")  
plt.ylabel("Monthly Charges of customer")  
plt.title("Tenure vs Monthly Charges")  
plt.show()

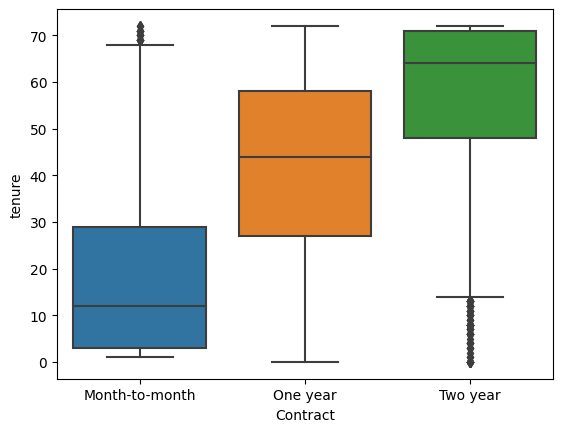


plt.scatter(x=df["tenure"].head(20),y=df["MonthlyCharges"].head(20),color="brown")  
plt.xlabel("Tenure of customer")  
plt.ylabel("Monthly Charges of customer")  
plt.title("Tenure vs Monthly Charges")  
plt.show()



d. Build a box-plot between ‘tenure’ & ‘Contract’. Map ‘tenure’ on the y-axis & ‘Contract’ on the x-axis.

sns.boxplot(x = df["Contract"], y = df["tenure"])  
plt.show()



**C) Linear Regression:**

a. Build a simple linear model where dependent variable is ‘MonthlyCharges’ and independent variable is ‘tenure’

i. Divide the dataset into train and test sets in 70:30 ratio.

ii. Build the model on train set and predict the values on test set

iii. After predicting the values, find the root mean square error

iv. Find out the error in prediction & store the result in ‘error’

v. Find the root mean square error

from sklearn import linear\_model  
from sklearn.model\_selection import train\_test\_split

x = df.loc[:,["tenure"]].values

x

array([[ 1],  
 [34],  
 [ 2],  
 ...,  
 [11],  
 [ 4],  
 [66]])

y = df.loc[:,["MonthlyCharges"]].values

y

array([[ 29.85],  
 [ 56.95],  
 [ 53.85],  
 ...,  
 [ 29.6 ],  
 [ 74.4 ],  
 [105.65]])

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,train\_size=0.70,random\_state=0)

from sklearn.linear\_model import LinearRegression

lr = LinearRegression()

lr.fit(x\_train,y\_train)

LinearRegression()

y\_pred = lr.predict(x\_test)

y\_pred

array([[ 58.2 ],  
 [116.6 ],  
 [ 71.95],  
 ...,  
 [109.95],  
 [ 24.55],  
 [ 81.6 ]])

from sklearn.metrics import mean\_squared\_error

mse = mean\_squared\_error(y\_test,y\_pred)

mse

6.144826981170695e-29

rmse = np.sqrt(mse)

rmse

7.838894680483146e-15

D) Logistic Regression:

a. Build a simple logistic regression model where dependent variable is ‘Churn’ & independent variable is ‘MonthlyCharges’

i. Divide the dataset in 65:35 ratio

ii. Build the model on train set and predict the values on test set

iii. Build the confusion matrix and get the accuracy score

x = df.loc[:,["MonthlyCharges"]].values

x

array([[ 29.85],  
 [ 56.95],  
 [ 53.85],  
 ...,  
 [ 29.6 ],  
 [ 74.4 ],  
 [105.65]])

y = df.loc[:,["Churn"]].values

y

array([['No'],  
 ['No'],  
 ['Yes'],  
 ...,  
 ['No'],  
 ['Yes'],  
 ['No']], dtype=object)

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,train\_size=0.65,random\_state=0)

from sklearn.linear\_model import LogisticRegression

lo\_r = LogisticRegression()

lo\_r.fit(x\_train,y\_train)

/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().  
 y = column\_or\_1d(y, warn=True)

LogisticRegression()

y\_pred = lo\_r.predict(x\_test)

y\_pred

array(['No', 'No', 'No', ..., 'No', 'No', 'No'], dtype=object)

from sklearn.metrics import confusion\_matrix, accuracy\_score

confusion\_matrix(y\_test,y\_pred)

array([[1815, 0],  
 [ 651, 0]])

accuracy\_score(y\_test,y\_pred)

0.7360097323600974

b. Build a multiple logistic regression model where dependent variable is ‘Churn’ & independent variables are ‘tenure’ & ‘MonthlyCharges’

i. Divide the dataset in 80:20 ratio

ii. Build the model on train set and predict the values on test set

iii. Build the confusion matrix and get the accuracy score

x = df.loc[:,["tenure","MonthlyCharges"]]  
y = df.loc[:,["Churn"]]

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,train\_size=0.80,random\_state=0)

from sklearn.linear\_model import LogisticRegression

lo\_r1 = LogisticRegression()

lo\_r1.fit(x\_train,y\_train)

/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().  
 y = column\_or\_1d(y, warn=True)

LogisticRegression()

y\_pred = lo\_r1.predict(x\_test)

y\_pred

array(['No', 'No', 'No', ..., 'No', 'No', 'No'], dtype=object)

y

Churn  
0 No  
1 No  
2 Yes  
3 No  
4 Yes  
... ...  
7038 No  
7039 No  
7040 No  
7041 Yes  
7042 No  
  
[7043 rows x 1 columns]

from sklearn.metrics import confusion\_matrix,accuracy\_score

confusion\_matrix(y\_test,y\_pred)

array([[934, 107],  
 [212, 156]])

accuracy\_score(y\_test,y\_pred)

0.7735982966643009

from sklearn.tree import DecisionTreeClassifier  
dec\_tree = DecisionTreeClassifier()

F) Random Forest:

a. Build a Random Forest model where dependent variable is ‘Churn’ & independent variables are ‘tenure’ and ‘MonthlyCharges’

i. Divide the dataset in 70:30 ratio

ii. Build the model on train set and predict the values on test set

iii. Build the confusion matrix and calculate the accuracy

x = df.loc[:,["MonthlyCharges","tenure"]]  
y = df.loc[:,["Churn"]]

x\_train, x\_test, y\_train, y\_test=train\_test\_split(x, y, test\_size=0.3, random\_state= 0)

from sklearn.ensemble import RandomForestClassifier  
RFC=RandomForestClassifier(n\_estimators=100)  
RFC.fit(x\_train, y\_train)

DecisionTreeClassifier()

y\_predict=RFC.predict(x\_test)  
y\_predict

array(['No', 'No', 'No', ..., 'No', 'No', 'No'], dtype=object)

from sklearn.metrics import confusion\_matrix, accuracy\_score  
confusion\_matrix(y\_predict, y\_test), accuracy\_score(y\_predict, y\_test)

(array([[1345, 332],  
 [ 215, 221]]),  
 0.7411263606247042)