**Data Set: Telecom Users**

**Overview of the Dataset:**

The "Telecom Users" dataset contains information related to various fields that describe telecom users.

1. Gender: Indicates the gender of the telecom user (e.g., male or female).
2. Senior Citizen: Indicates whether the user is a senior citizen (typically defined as 65 years or older).
3. Partner: Indicates whether the user has a partner or not.
4. Dependent: Indicates whether the user has any dependents or not.
5. Tenure: Represents the duration (in months) for which the user has been a customer of the telecom service provider.
6. Phone Service: Indicates whether the user has a phone service or not.
7. Multiple Lines: Indicates whether the user has multiple phone lines or not.
8. Internet Services: Indicates the type of internet service subscribed by the user.
9. Online Security: Indicates whether the user has subscribed to online security services.
10. Online Backup: Indicates whether the user has subscribed to online backup services.
11. Device Protection: Indicates whether the user has subscribed to device protection services.
12. Tech Support: Indicates whether the user has subscribed to tech support services.
13. Streaming TV: Indicates whether the user has streaming TV services.
14. Streaming Movies: Indicates whether the user has streaming movie services.
15. Contract: Represents the type of contract the user has with the telecom service provider (e.g., month-to-month, one-year, or two-year contract).
16. Paperless Billing: Indicates whether the user has opted for paperless billing.
17. Payment Method: Indicates the user's preferred payment method.
18. Monthly Charges: Represents the amount charged to the user on a monthly basis.
19. Total Charges: Represents the total charges incurred by the user over the entire tenure.
20. Churn: Indicates whether the user has churned (i.e., discontinued the telecom services) or not.

This dataset provides valuable information about telecom users, allowing for analysis and insights into factors that may influence churn rates.

Total Rows: 6050

Total Columns: 23

Reduced to 20 columns

**Overview of the Tool Used: Python**

Python is a versatile and widely used programming language that is extensively utilized for data analysis, including the analysis of the "Telecom Users" dataset.:

1. Versatile and Easy to Use
2. Abundance of Libraries
3. Data Manipulation and Exploration
4. Data Visualization
5. Machine Learning and Statistical Analysis
6. Integration and Collaboration
7. Scalability and Performance

**Summary**

Top of Form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| data. shape | | :(6050, 2) | |  | |
| df=data.iloc[:, 2:22] | |  | |  | |
| df.shape | | :(6050, 20) | |  | |
| df.describe() | |  | |  | |
| df['gender'].value\_counts() | |  | |  | |
| df.nunique() | |  | |  | |
| df.isnull().sum() | |  | |  | |
| df.corr(): | |  | |  | |
| index | SeniorCitizen | | Tenure | | MonthlyCharges |
| SeniorCitizen | 1.0 | | 0.0059803317085461915 | | 0.2218037411683066 |
| tenure | 0.0059803317085461915 | | 1.0 | | 0.25474544206056093 |
| MonthlyCharges | 0.2218037411683066 | | 0.25474544206056093 | | 1.0 |

**Question 1.**

1. Total Male Subscribers (which are not Senior Citizens) with the following:

a. Phone Service

b. Internet Service

c. Device Protection

d. Streaming TV

e. Paperless billing

**Code:**

import pandas as pd

total\_male\_non\_senior\_subscribers = len(df[(df["gender"] == "Male") & (df["SeniorCitizen"] == 0)])

print("Total male subscribers who are not senior citizens:", total\_male\_non\_senior\_subscribers)

import matplotlib.pyplot as plt

filtered\_df = df.loc[(df["gender"] == "Male") & (df["SeniorCitizen"] == 0), ["gender", "SeniorCitizen", "PhoneService", "InternetService", "PaperlessBilling", "DeviceProtection","StreamingTV"]]

**# Count the number of rows for each combination of PhoneService, InternetService, and BillingServices**

countdf\_Male = filtered\_df.groupby(["PhoneService", "InternetService", "PaperlessBilling"]).size().reset\_index(name="count")

**# Create a bar plot**

plt.figure(figsize=(10, 6))

plt.bar(range(len(countdf\_Male)), countdf\_Male["count"])

plt.xlabel("Combination")

plt.ylabel("Count")

plt.title("Count of Male Subscribers (Non-Senior) by Service Combination")

plt.xticks(range(len(countdf\_Male)), countdf\_Male["PhoneService"] + "-" + countdf\_Male["InternetService"] + "-" + countdf\_Male["PaperlessBilling"], rotation=45)

**# Add labels to the bars**

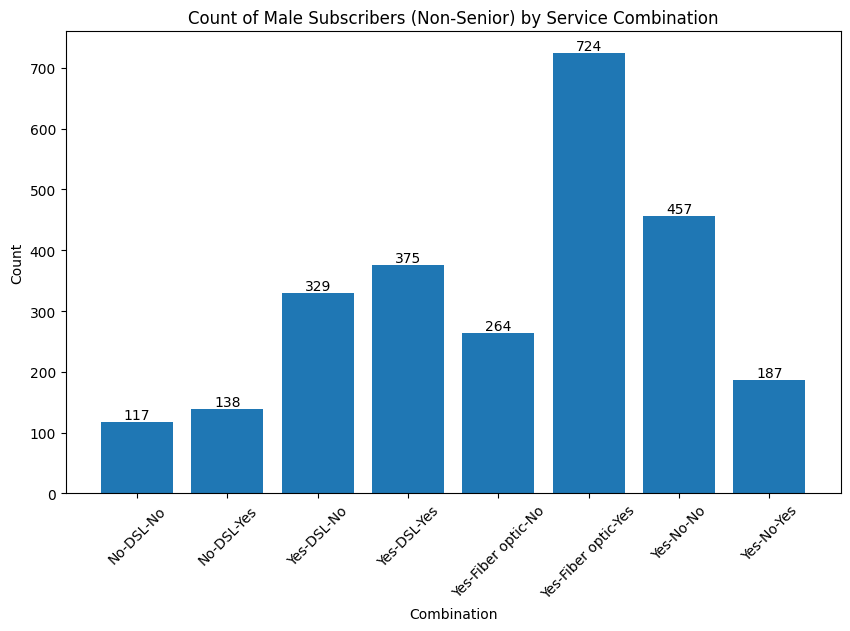
for i, count in enumerate(countdf\_Male["count"]):

    plt.text(i, count, str(count), ha="center", va="bottom")

plt.show()

**Output:**

**Total male subscribers who are not senior citizens: 2591**



Total Female Subscribers (which are not Senior Citizens)

a. Phone Service

b. Internet Service

c. Device Protection

d. Streaming TV

e. Paperless billing

Code:

total\_female\_non\_senior\_subscribers= len(df[(df['gender']=='Female') & (df['SeniorCitizen']==0)])

print("Total male subscribers who are not senior citizens:", total\_female\_non\_senior\_subscribers)

import matplotlib.pyplot as plt

filtered\_df = df.loc[(df["gender"] == "Female") & (df["SeniorCitizen"] == 0), ["gender", "SeniorCitizen", "PhoneService", "InternetService", "PaperlessBilling", "DeviceProtection","StreamingTV"]]

**# Count the number of rows for each combination of PhoneService, InternetService, and BillingServices**

countdf\_Female = filtered\_df.groupby(["PhoneService", "InternetService", "PaperlessBilling"]).size().reset\_index(name="count")

**# Create a bar plot**

plt.figure(figsize=(10, 6))

plt.bar(range(len(countdf\_Female)), countdf\_Female["count"])

plt.xlabel("Combination")

plt.ylabel("Count")

plt.title("Count of Female Subscribers (Non-Senior) by Service Combination")

plt.xticks(range(len(countdf\_Female)), countdf\_Female["PhoneService"] + "-" + countdf\_Female["InternetService"] + "-" + countdf\_Female["PaperlessBilling"], rotation=45)

# Add labels to the bars

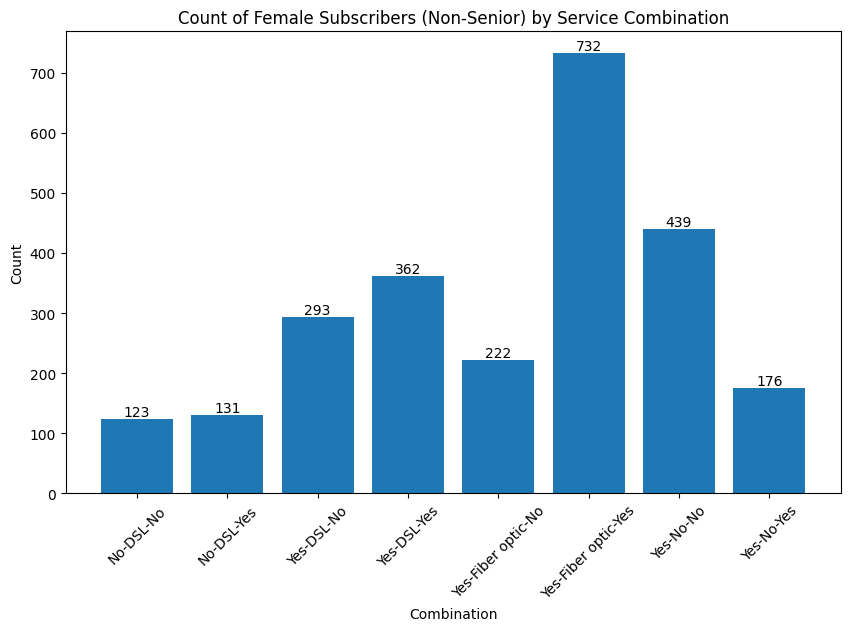
for i, count in enumerate(countdf\_Female["count"]):

    plt.text(i, count, str(count), ha="center", va="bottom")

plt.show()

**Output:**

Total male subscribers who are not senior citizens: 2478



**Question-3**

**Correlation of Senior Citizen, Tenure and Monthly Charges**

import seaborn as sns

**# Create a correlation matrix**

correlation\_matrix = df[["SeniorCitizen", "tenure", "MonthlyCharges"]].corr()

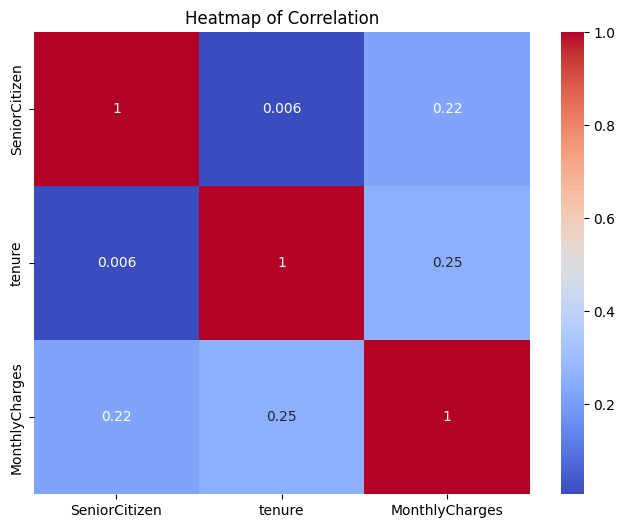
**# Create a heatmap using Seaborn**

plt.figure(figsize=(8, 6))

sns.heatmap(correlation\_matrix, annot=True, cmap="coolwarm")

plt.title("Heatmap of Correlation")

plt.show()



The relationship between the Monthly charge and Tenure is high. The next highest relationship is between the Senior Citizen and Monthly Charge

**Question - 5**

**Total Variable Count in Partners**

Code:

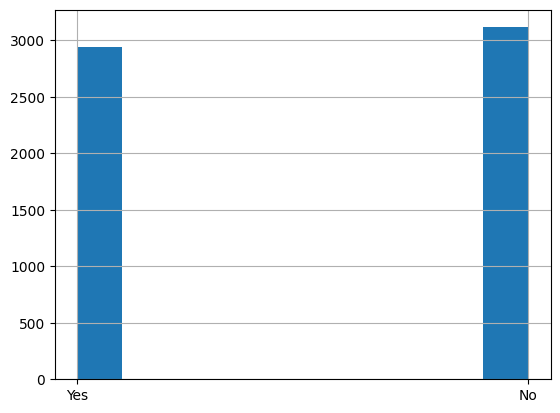
ba1\_partner= df["Partner"].value\_counts()

df.Partner.hist()

**Output:**

No 3114

Yes 2936



**Question - 5**

1. Filter and count the dataset to include only the relevant rows that match the given conditions:
   * Gender = Male
   * SeniorCitizen = 0 (Not a senior citizen)
   * Tenure = 1
   * Churn = Yes

**Code:**

Churn\_Male\_Tenure\_1= df.loc[(df["gender"]=="Male") & (df["SeniorCitizen"]==0) & (df["tenure"]==1) & (df["Churn"]=="Yes")]

Churn\_Male\_Tenure\_1\_grp = Churn\_Male\_Tenure\_1.groupby("InternetService").size().reset\_index(name="count")

Churn\_Male\_Tenure\_1\_grp

**Output:**

|  |  |  |
| --- | --- | --- |
| index | InternetService | count |
| 0 | DSL | 50 |
| 1 | Fiber optic | 63 |
| 2 | No | 27 |

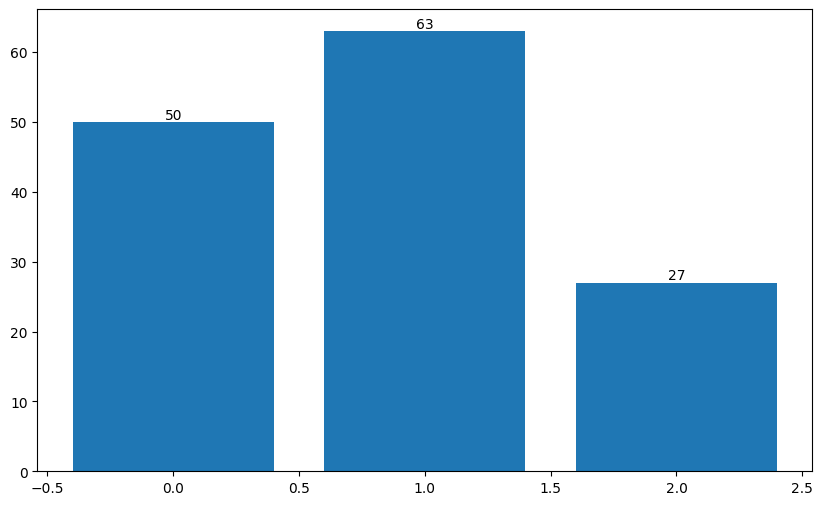
**Chart:**

plt.figure(figsize=(10,6))

plt.bar(range(len(Churn\_Male\_Tenure\_1\_grp)), Churn\_Male\_Tenure\_1\_grp["count"])

for i, count in enumerate(Churn\_Male\_Tenure\_1\_grp["count"]):

    plt.text(i, count, str(count), ha="center", va="bottom")



**Question - 6**

1. Filter and count the dataset to include only the relevant rows that match the given conditions:
   * Churn = "Yes"
   * MonthlyCharges >=75
   * Group by gender

**Code:**

# Filter and Count the dataset to include only the relevant rows that match the given conditions:

# Churn = "Yes" & MonthlyCharges >=75

df1= df[(df["Churn"]=="Yes") & (df["MonthlyCharges"] >= 75)]

count\_df1\_gender\_yes\_G75 = df1.groupby("gender").size().reset\_index(name="count")

count\_df1\_gender\_yes\_G75

# Create a bar plot

plt.figure(figsize=(6, 6))

plt.bar(range(len(count\_df1\_gender\_yes\_G75)), count\_df1\_gender\_yes\_G75["count"])

plt.xlabel("Churn=Yes & Monthly Charges >=75")

plt.ylabel("Count")

plt.title("Count of Subscribers (Both Male and Female) withdrawn by Monthly Charge is greater than 75")

plt.xticks(range(len(count\_df1\_gender\_yes\_G75)), count\_df1\_gender\_yes\_G75["gender"], rotation=45)

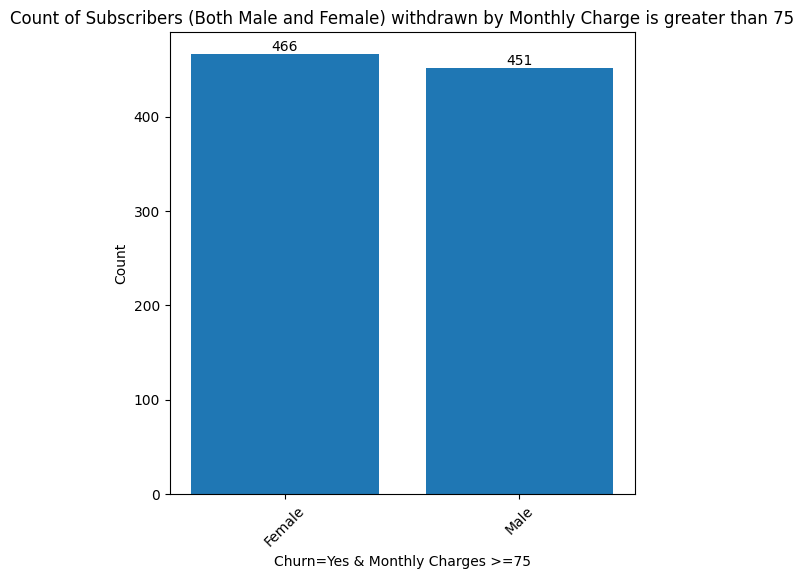
# Add labels to the bars

for i, count in enumerate(count\_df1\_gender\_yes\_G75["count"]):

    plt.text(i, count, str(count), ha="center", va="bottom")

plt.show()

Output:



**Question - 7**

1. Filter and count the dataset to include only the relevant rows that match the given conditions:
   * Churn = "Yes"
   * MonthlyCharges >=75
   * SeniorCitizen
   * Group by gender

**Code:**

df1= df[(df["Churn"]=="Yes") & (df["MonthlyCharges"] >= 75) & (df["SeniorCitizen"]==1)]

count\_df1\_Seniorcitizen\_yes\_G75 = df1.groupby("gender").size().reset\_index(name="count")

count\_df1\_Seniorcitizen\_yes\_G75

# Create a bar plot

plt.figure(figsize=(6,6))

plt.bar(range(len(count\_df1\_Seniorcitizen\_yes\_G75)), count\_df1\_Seniorcitizen\_yes\_G75["count"])

plt.xlabel("Churn = Yes & Monthly Charges >75 & Senior Citizen")

plt.ylabel("Count")

plt.title("Count of Senior Citizen (Both  Male and Female) withdrawn by Monthly Charge is greater than 75")

plt.xticks(range(len(count\_df1\_Seniorcitizen\_yes\_G75)), count\_df1\_Seniorcitizen\_yes\_G75["gender"], rotation=45)

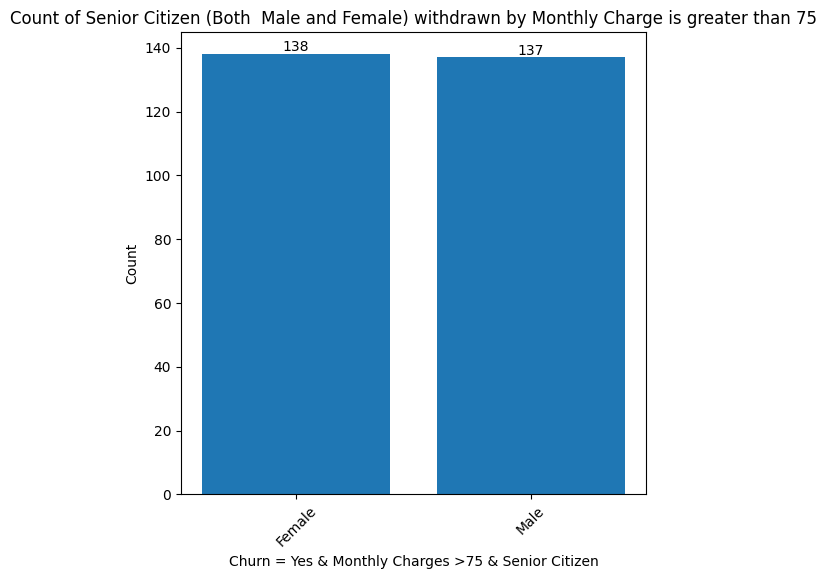
# Add labels to the bar

for i, count in enumerate (count\_df1\_Seniorcitizen\_yes\_G75["count"]):

  plt.text(i,count,str(count),ha="center", va="bottom")

plt.show()

**Output:**

****

**Question - 8**

1. Filter and count the dataset to include only the relevant rows that match the given conditions:
   * Churn = "Yes"
   * MonthlyCharges >=75
   * Non-SeniorCitizen
   * Group by gender

**Code:**

df1= df[(df["Churn"]=="Yes") & (df["MonthlyCharges"] >= 75) & (df["SeniorCitizen"]==0)]

count\_df1\_notseniorcitizen\_yes\_G75 = df1.groupby("gender").size().reset\_index(name="count")

count\_df1\_notseniorcitizen\_yes\_G75

# Creating bar Plot

plt.figure(figsize=(6,6))

plt.bar(range(len(count\_df1\_notseniorcitizen\_yes\_G75)), count\_df1\_notseniorcitizen\_yes\_G75["count"])

plt.xlabel("Churn = Yes & Monthly Charges >=75 & Non-Senior Citizen")

plt.ylabel("Count")

plt.title("Count of Non-Senior Citizen (Both  Male and Female) withdrawn by Monthly Charge is greater than 75")

plt.xticks(range(len(count\_df1\_notseniorcitizen\_yes\_G75)),count\_df1\_notseniorcitizen\_yes\_G75["gender"], rotation=45)

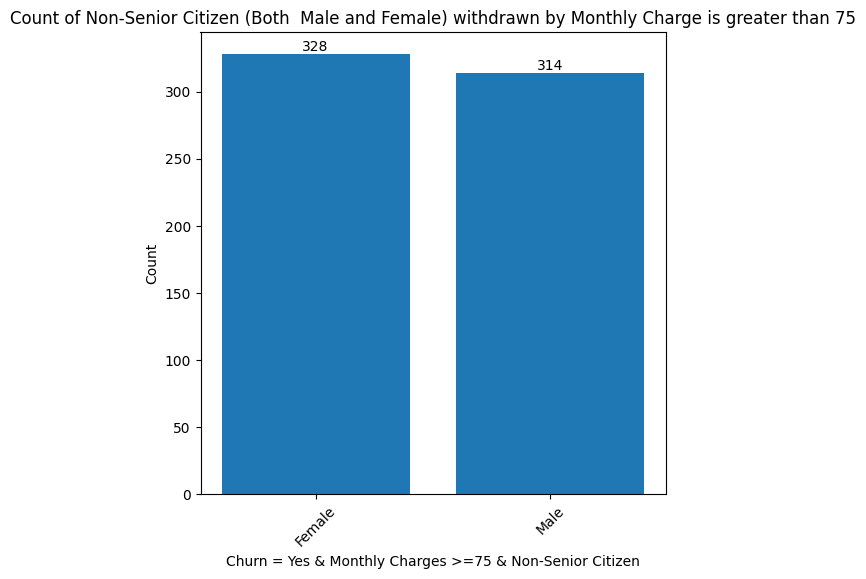
# Add labels to the bar

for i, count in enumerate(count\_df1\_notseniorcitizen\_yes\_G75["count"]):

  plt.text(i,count,str(count), ha="center", va = "bottom")

plt. show()

Output:

****

**Question - 9**

1. Filter and count the dataset to include only the relevant rows that match the given conditions:
   * Churn = "Yes"
   * MonthlyCharges <=75
   * Group by gender

**Code:**

df1= df[(df[“Churn”]==”Yes”) & (df[“MonthlyCharges”] <= 75)]

count\_df1\_gender\_yes\_L75 = df1.groupby(“gender”).size().reset\_index(name=”count”)

count\_df1\_gender\_yes\_L75

# Create a bar plot

plt.figure(figsize=(6, 6))

plt.bar(range(len(count\_df1\_gender\_yes\_L75)), count\_df1\_gender\_yes\_L75[“count”])

plt.xlabel(“Churn=Yes & Monthly Charges <=75”)

plt.ylabel(“Count”)

plt.title(“Count of Subscribers (Both Male and Female) withdrawn by Monthly Charge is less than 75”)

plt.xticks(range(len(count\_df1\_gender\_yes\_L75)), count\_df1\_gender\_yes\_L75[“gender”], rotation=45)

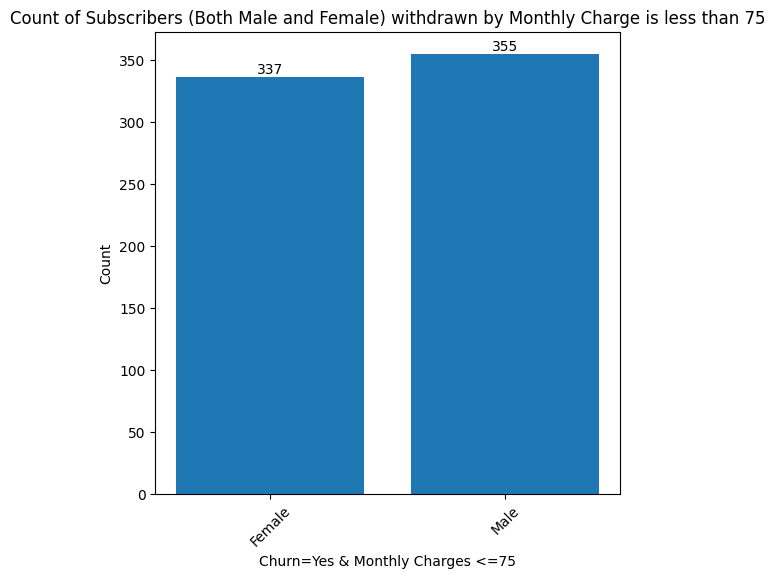
# Add labels to the bars

for I, count in enumerate(count\_df1\_gender\_yes\_L75[“count”]):

    plt.text(I, count, str(count), ha=”center”, va=”bottom”)

plt.show()

**Output:**

****

**Question - 10**

1. Filter and count the dataset to include only the relevant rows that match the given conditions:
   * Churn = "Yes"
   * MonthlyCharges <=75
   * Senior Citizen
   * Group by gender

**Code:**

df1= df[(df["Churn"]=="Yes") & (df["MonthlyCharges"] <= 75) & (df["SeniorCitizen"]==1)]

count\_df1\_Seniorcitizen\_yes\_L75 = df1.groupby("gender").size().reset\_index(name="count")

count\_df1\_Seniorcitizen\_yes\_L75

# Create a bar plot

plt.figure(figsize=(6,6))

plt.bar(range(len(count\_df1\_Seniorcitizen\_yes\_L75)), count\_df1\_Seniorcitizen\_yes\_L75["count"])

plt.xlabel("Churn = Yes & Monthly Charges <=75 & Senior Citizen")

plt.ylabel("Count")

plt.title("Count of Senior Citizen (Both  Male and Female) withdrawn by Monthly Charge is less than 75")

plt.xticks(range(len(count\_df1\_Seniorcitizen\_yes\_L75)), count\_df1\_Seniorcitizen\_yes\_L75["gender"], rotation=45)

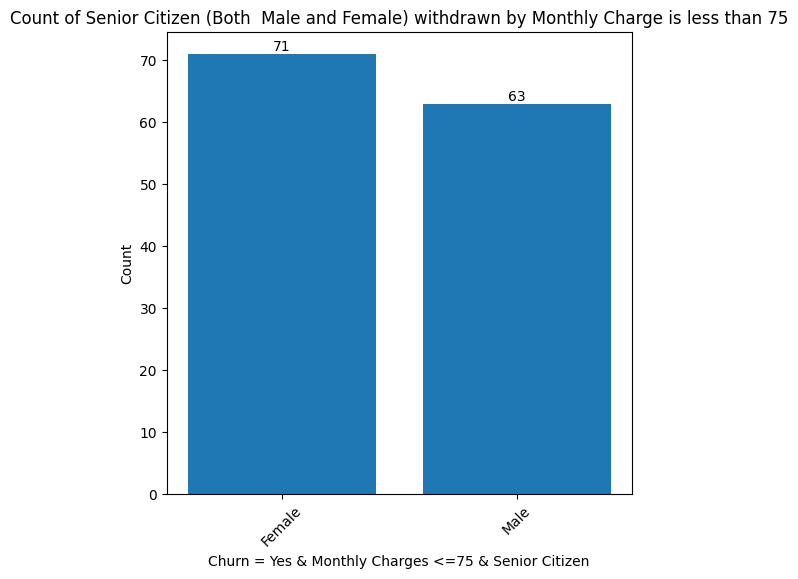
# Add labels to the bar

for i, count in enumerate (count\_df1\_Seniorcitizen\_yes\_L75["count"]):

  plt.text(i,count,str(count),ha="center", va="bottom")

plt.show()

**Output:**

****

**Question - 11**

1. Filter and count the dataset to include only the relevant rows that match the given conditions:
   * Churn = "Yes"
   * MonthlyCharges <=75
   * Non-Senior Citizen
   * Group by gender

**Code:**

df1= df[(df["Churn"]=="Yes") & (df["MonthlyCharges"] <= 75) & (df["SeniorCitizen"]==0)]

count\_df1\_notseniorcitizen\_yes\_L75 = df1.groupby("gender").size().reset\_index(name="count")

count\_df1\_notseniorcitizen\_yes\_L75

# Creating bar Plot

plt.figure(figsize=(6,6))

plt.bar(range(len(count\_df1\_notseniorcitizen\_yes\_L75)), count\_df1\_notseniorcitizen\_yes\_L75["count"])

plt.xlabel("Churn = Yes & Monthly Charges <=75 & Non-Senior Citizen")

plt.ylabel("Count")

plt.title("Count of Non-Senior Citizen (Both Male and Female) withdrawn by Monthly Charge is less than 75")

plt.xticks(range(len(count\_df1\_notseniorcitizen\_yes\_L75)),count\_df1\_notseniorcitizen\_yes\_L75["gender"], rotation=45)

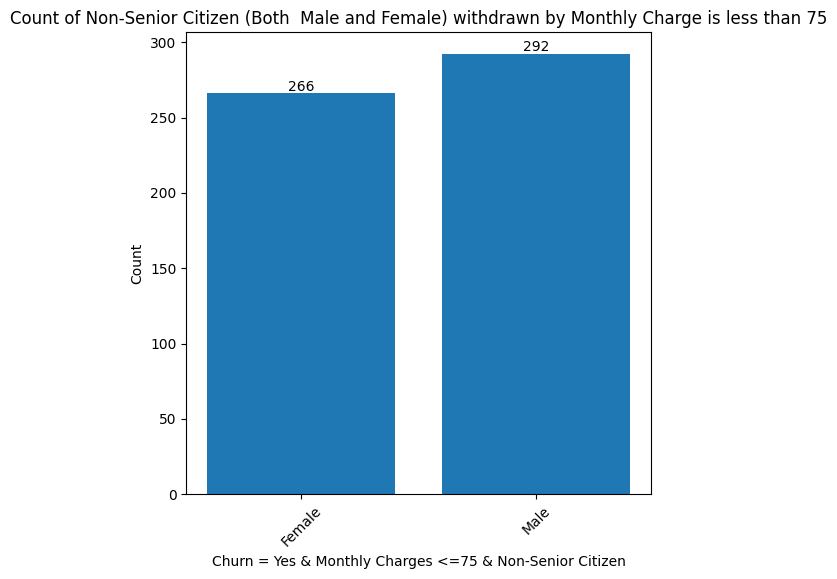
# Add labels to the bar

for i, count in enumerate(count\_df1\_notseniorcitizen\_yes\_L75["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt. show()

**Count:**

****

**Question - 12**

1. Filter and count the dataset to include only the relevant rows that match the given conditions:
   * Churn = "No"
   * MonthlyCharges >=75
   * Group by gender

**Code:**

df1= df[(df["Churn"]=="No") & (df["MonthlyCharges"] >= 75)]

count\_df1\_gender\_no\_G75 = df1.groupby("gender").size().reset\_index(name="count")

count\_df1\_gender\_no\_G75

# Create a bar plot

plt.figure(figsize=(6, 6))

plt.bar(range(len(count\_df1\_gender\_no\_G75)), count\_df1\_gender\_no\_G75["count"])

plt.xlabel("Churn=Yes & Monthly Charges >=75")

plt.ylabel("Count")

plt.title("Count of Subscribers (Both Male and Female) not withdrawn by Monthly Charge is greater than 75")

plt.xticks(range(len(count\_df1\_gender\_no\_G75)), count\_df1\_gender\_no\_G75["gender"], rotation=45)

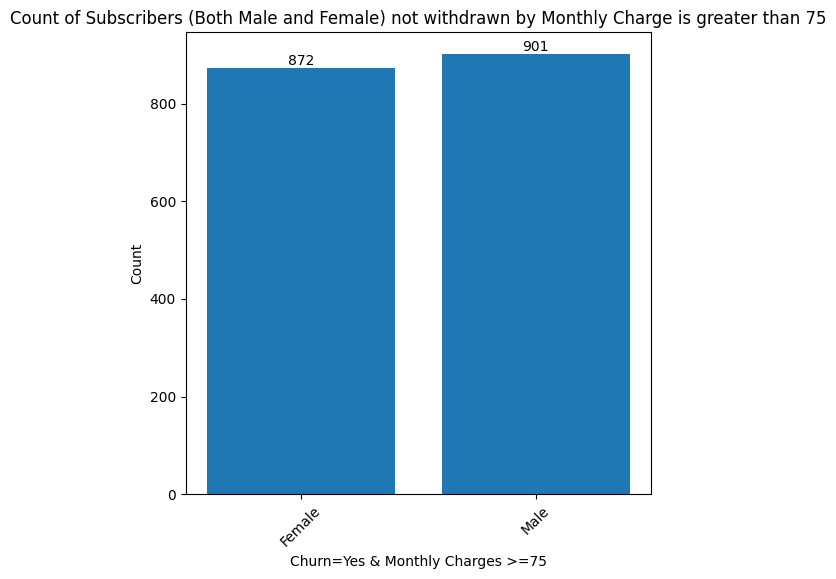
# Add labels to the bars

for i, count in enumerate(count\_df1\_gender\_no\_G75["count"]):

    plt.text(i, count, str(count), ha="center", va="bottom")

plt.show()

**Output:**

****

**Question - 13**

1. Filter and count the dataset to include only the relevant rows that match the given conditions:
   * Churn = "No"
   * MonthlyCharges >=75
   * Senior Citizen
   * Group by gender

**Code:**

df1= df[(df["Churn"]=="No") & (df["MonthlyCharges"] >= 75) & (df["SeniorCitizen"]==1)]

count\_df1\_Seniorcitizen\_no\_G75 = df1.groupby("gender").size().reset\_index(name="count")

count\_df1\_Seniorcitizen\_no\_G75

# Create a bar plot

plt.figure(figsize=(6,6))

plt.bar(range(len(count\_df1\_Seniorcitizen\_no\_G75)), count\_df1\_Seniorcitizen\_no\_G75["count"])

plt.xlabel("Churn = No & Monthly Charges >75 & Senior Citizen")

plt.ylabel("Count")

plt.title("Count of Senior Citizen (Both  Male and Female) not withdrawn by Monthly Charge is greater than 75")

plt.xticks(range(len(count\_df1\_Seniorcitizen\_no\_G75)), count\_df1\_Seniorcitizen\_no\_G75["gender"], rotation=45)

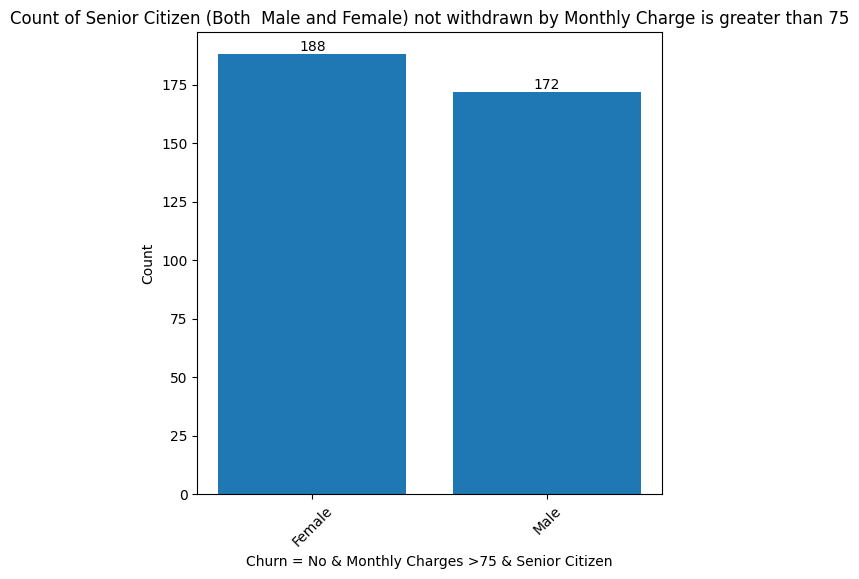
# Add labels to the bar

for i, count in enumerate (count\_df1\_Seniorcitizen\_no\_G75["count"]):

  plt.text(i,count,str(count),ha="center", va="bottom")

plt.show()

**Output:**

****

**Question - 14**

1. Filter and count the dataset to include only the relevant rows that match the given conditions:
   * Churn = "No"
   * MonthlyCharges >=75
   * Non-Senior Citizen
   * Group by gender

**Code:**

df1= df[(df["Churn"]=="No") & (df["MonthlyCharges"] >= 75) & (df["SeniorCitizen"]==0)]

count\_df1\_notseniorcitizen\_no\_G75 = df1.groupby("gender").size().reset\_index(name="count")

count\_df1\_notseniorcitizen\_no\_G75

# Creating bar Plot

plt.figure(figsize=(6,6))

plt.bar(range(len(count\_df1\_notseniorcitizen\_no\_G75)), count\_df1\_notseniorcitizen\_no\_G75["count"])

plt.xlabel("Churn = No & Monthly Charges >=75 & Non-Senior Citizen")

plt.ylabel("Count")

plt.title("Count of Non-Senior Citizen (Both  Male and Female) not withdrawn by Monthly Charge is greater than 75")

plt.xticks(range(len(count\_df1\_notseniorcitizen\_no\_G75)),count\_df1\_notseniorcitizen\_no\_G75["gender"], rotation=45)

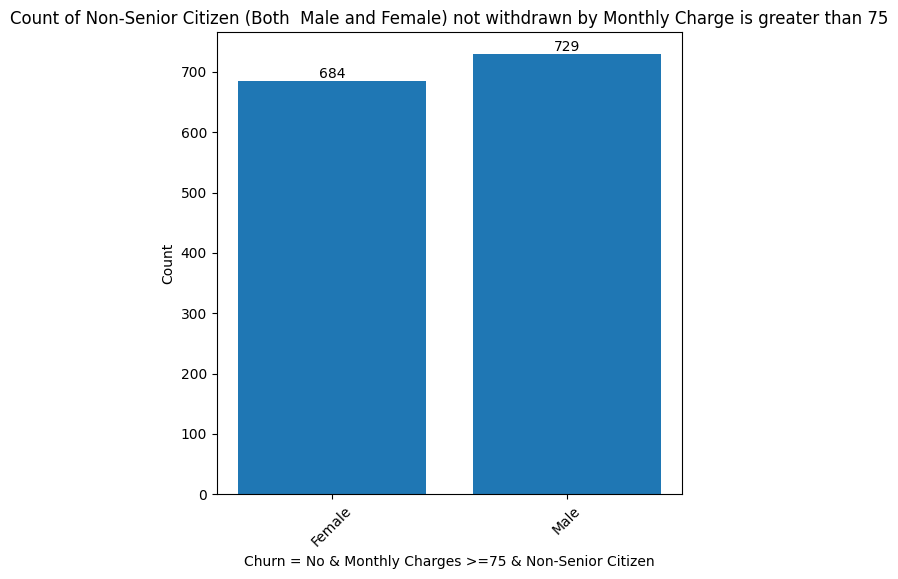
# Add labels to the bar

for i, count in enumerate(count\_df1\_notseniorcitizen\_no\_G75["count"]):

  plt.text(i,count,str(count), ha="center", va = "bottom")

plt. show()

**Output:**

****

**Question - 15**

1. Filter and count the dataset to include only the relevant rows that match the given conditions:
   * Churn = "No"
   * MonthlyCharges <=75
   * Group by gender

**Code:**

df1= df[(df["Churn"]=="No") & (df["MonthlyCharges"] <= 75)]

count\_df1\_gender\_no\_L75 = df1.groupby("gender").size().reset\_index(name="count")

count\_df1\_gender\_no\_L75

# Create a bar plot

plt.figure(figsize=(6, 6))

plt.bar(range(len(count\_df1\_gender\_no\_L75)), count\_df1\_gender\_no\_L75["count"])

plt.xlabel("Churn=No & Monthly Charges <=75")

plt.ylabel("Count")

plt.title("Count of Subscribers (Both Male and Female) not withdrawn by Monthly Charge is less than 75")

plt.xticks(range(len(count\_df1\_gender\_no\_L75)), count\_df1\_gender\_no\_L75["gender"], rotation=45)

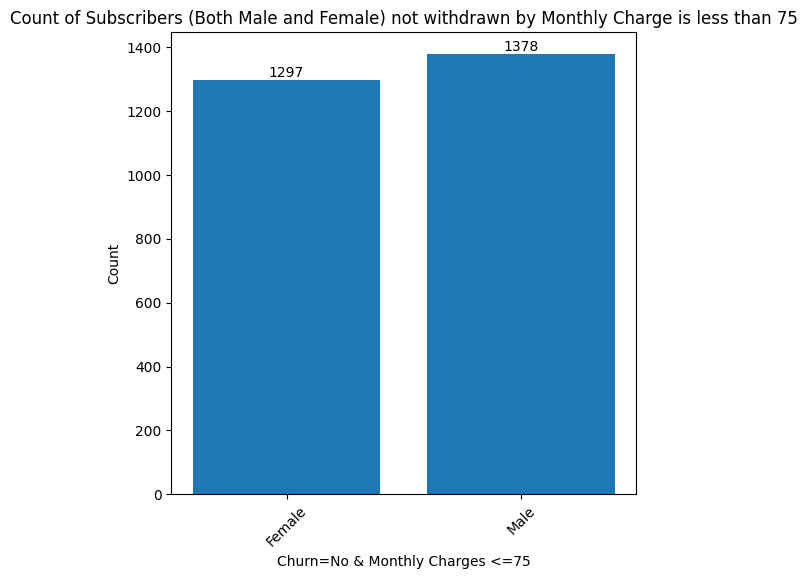
# Add labels to the bars

for i, count in enumerate(count\_df1\_gender\_no\_L75["count"]):

    plt.text(i, count, str(count), ha="center", va="bottom")

plt.show()

**Output:**

****

**Question - 16**

1. Filter and count the dataset to include only the relevant rows that match the given conditions:
   * Churn = "No"
   * MonthlyCharges <=75
   * Senior Citizen
   * Group by gender

**Code:**

df1= df[(df["Churn"]=="No") & (df["MonthlyCharges"] <= 75) & (df["SeniorCitizen"]==1)]

count\_df1\_Seniorcitizen\_no\_L75 = df1.groupby("gender").size().reset\_index(name="count")

count\_df1\_Seniorcitizen\_no\_L75

# Create a bar plot

plt.figure(figsize=(6,6))

plt.bar(range(len(count\_df1\_Seniorcitizen\_no\_L75)), count\_df1\_Seniorcitizen\_no\_L75["count"])

plt.xlabel("Churn = No & Monthly Charges <=75 & Senior Citizen")

plt.ylabel("Count")

plt.title("Count of Senior Citizen (Both  Male and Female) not withdrawn by Monthly Charge is less than 75")

plt.xticks(range(len(count\_df1\_Seniorcitizen\_no\_L75)), count\_df1\_Seniorcitizen\_no\_L75["gender"], rotation=45)

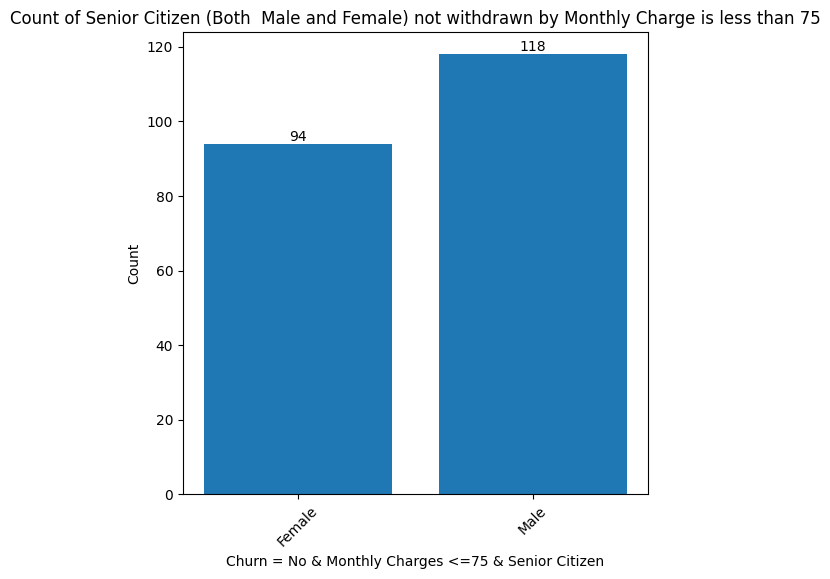
# Add labels to the bar

for i, count in enumerate (count\_df1\_Seniorcitizen\_no\_L75["count"]):

  plt.text(i,count,str(count),ha="center", va="bottom")

plt.show()

**Output:**

****

**Question - 17**

1. Filter and count the dataset to include only the relevant rows that match the given conditions:
   * Churn = "No"
   * MonthlyCharges <=75
   * Non-Senior Citizen
   * Group by gender

**Code:**

df1= df[(df["Churn"]=="No") & (df["MonthlyCharges"] <= 75) & (df["SeniorCitizen"]==0)]

count\_df1\_notseniorcitizen\_no\_L75 = df1.groupby("gender").size().reset\_index(name="count")

count\_df1\_notseniorcitizen\_no\_L75

 # Creating bar Plot

plt.figure(figsize=(6,6))

plt.bar(range(len(count\_df1\_notseniorcitizen\_no\_L75)), count\_df1\_notseniorcitizen\_no\_L75["count"])

plt.xlabel("Churn = No & Monthly Charges <=75 & Non-Senior Citizen")

plt.ylabel("Count")

plt.title("Count of Non-Senior Citizen (Both  Male and Female) not withdrawn by Monthly Charge is less than 75")

plt.xticks(range(len(count\_df1\_notseniorcitizen\_no\_L75)),count\_df1\_notseniorcitizen\_no\_L75["gender"], rotation=45)

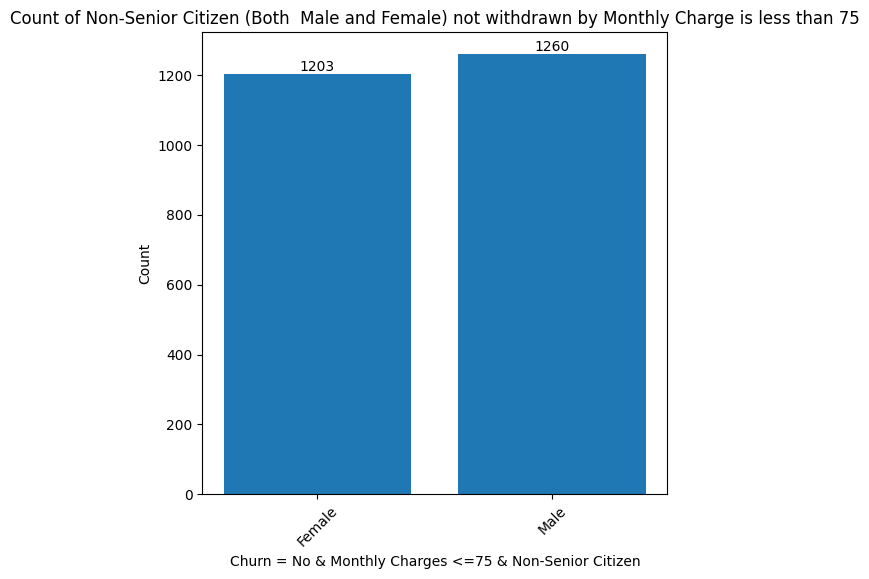
# Add labels to the bar

for i, count in enumerate(count\_df1\_notseniorcitizen\_no\_L75["count"]):

  plt.text(i,count,str(count), ha="center", va = "bottom")

plt. show()

**Output:**

****

**Question - 18**

1. Compare and draw the graph for the given conditions:
   * 'No of gender withdrawn who has Monthlycharge is greater than 75
   * 'No of gender not withdrawn who has Monthlycharge is greater than 75'

**Code:**

import matplotlib.pyplot as plt

# Define the data

datasets = [

    {

        'label': 'No of gender withdrawn who has Monthlycharge is greater than 75',

        'gender': ['Male', 'Female'],

        'count': [451, 466],

        'color': 'blue'

    },

    {

        'label': 'No of gender not withdrawn who has Monthlycharge is greater than 75',

        'gender': ['Male', 'Female'],

        'count': [901, 872],

        'color': 'orange'

    }

]

# Set up the chart

categories = datasets[0]['gender']

bar\_width = 0.15

index = range(len(categories))

# Plot the bars for each dataset

for i, dataset in enumerate(datasets):

    plt.bar([j + (i \* bar\_width) for j in index], dataset['count'], bar\_width, color=dataset['color'], label=dataset['label'])

# Add labels and title

plt.xlabel('Gender')

plt.ylabel('Count')

plt.title('Comparison of Gender Counts')

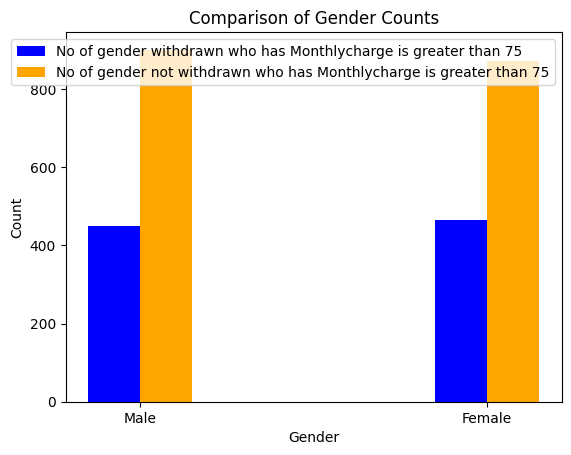
plt.xticks([j + (bar\_width \* (len(datasets) - 1) / 2) for j in index], categories)

plt.legend()

# Display the chart

plt.show()

**Output:**

****

**Question - 19**

1. Compare and draw the graph for the given conditions:
   * 'No of gender withdrawn who has Monthlycharge is greater than 75
   * 'No of gender not withdrawn who has Monthlycharge is greater than 75'

**Code:**

import matplotlib.pyplot as plt

# Define the data

datasets = [

    {

        'label': 'No of SeniorCitizen withdrawn who has Monthlycharge is greater than 75',

        'gender': ['Male', 'Female'],

        'count': [137, 138],

        'color': 'blue'

    },

    {

        'label': 'No of SeniorCitizen not withdrawn who has Monthlycharge is greater than 75',

        'gender': ['Male', 'Female'],

        'count': [172, 188],

        'color': 'orange'

    }

]

# Set up the chart

categories = datasets[0]['gender']

bar\_width = 0.15

index = range(len(categories))

# Plot the bars for each dataset

for i, dataset in enumerate(datasets):

    plt.bar([j + (i \* bar\_width) for j in index], dataset['count'], bar\_width, color=dataset['color'], label=dataset['label'])

# Add labels and title

plt.xlabel('Gender')

plt.ylabel('Count')

plt.title('Comparison of Gender Counts')

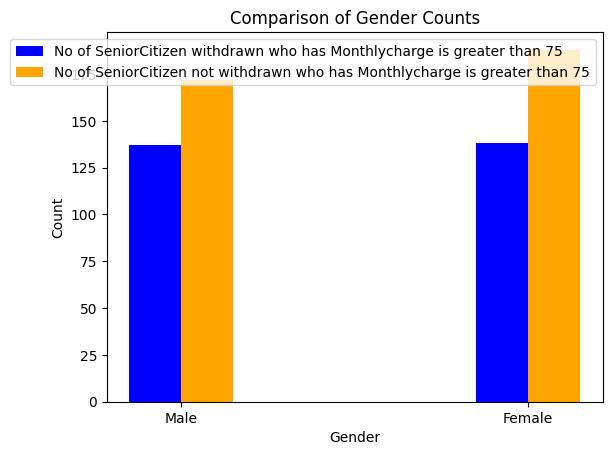
plt.xticks([j + (bar\_width \* (len(datasets) - 1) / 2) for j in index], categories)

plt.legend()

# Display the chart

plt.show()

**Output:**

****

**Question - 20**

1. Compare and draw the graph for the given conditions:
   * No of Non-SeniorCitizen withdrawn who has Monthlycharge is greater than 75',
   * No of Non-SeniorCitizen not withdrawn who has Monthlycharge is greater than 75

**Code:**

import matplotlib.pyplot as plt

# Define the data

datasets = [

    {

        'label': 'No of Non-SeniorCitizen withdrawn who has Monthlycharge is greater than 75',

        'gender': ['Male', 'Female'],

        'count': [314, 328],

        'color': 'blue'

    },

    {

        'label': 'No of Non-SeniorCitizen not withdrawn who has Monthlycharge is greater than 75',

        'gender': ['Male', 'Female'],

        'count': [729, 684],

        'color': 'orange'

    }

]

# Set up the chart

categories = datasets[0]['gender']

bar\_width = 0.15

index = range(len(categories))

# Plot the bars for each dataset

for i, dataset in enumerate(datasets):

    plt.bar([j + (i \* bar\_width) for j in index], dataset['count'], bar\_width, color=dataset['color'], label=dataset['label'])

# Add labels and title

plt.xlabel('Gender')

plt.ylabel('Count')

plt.title('Comparison of Gender Counts')

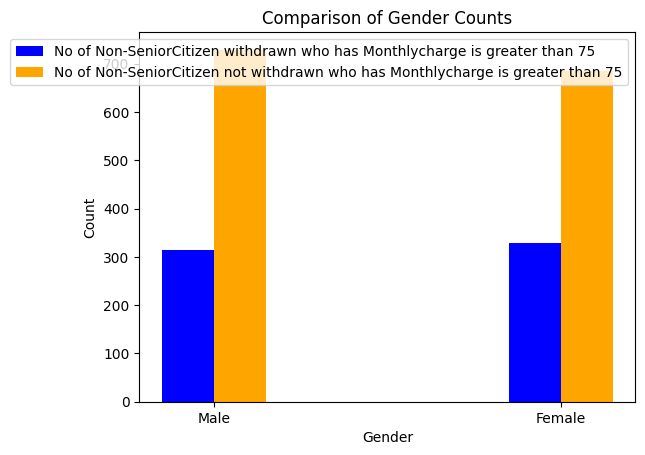
plt.xticks([j + (bar\_width \* (len(datasets) - 1) / 2) for j in index], categories)

plt.legend()

# Display the chart

plt.show()

**Output:**

****

**Additional Summary**

**AQ1.** Filter and count based on the following condition

* Multiplelines = ‘No internet Service’
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport = 'No'
* Churn rate = ‘No’
* Group by ‘Gender’

**Code:**

df1= df[(df['MultipleLines'] == 'No internetService')|(df['OnlineSecurity']== 'No') | (df['OnlineBackup']=='No') | (df['DeviceProtection']=='No')| (df['TechSupport']=='No')| (df['Churn'] =='No')]

df1\_NointernetService\_No = df1.groupby("gender").size().reset\_index(name= 'count')

df1\_NointernetService\_No

# plot Bar chart

plt.figure(figsize=(6,6))

plt.bar(range(len(df1\_NointernetService\_No)), df1\_NointernetService\_No['count'])

plt.xlabel("Multpliline= No (or) internetService or Onlinesecurity=No or OnlineBackup=No or DeviceProtection=No, or TechSupport=No (or) Churn = No")

plt.ylabel("Count")

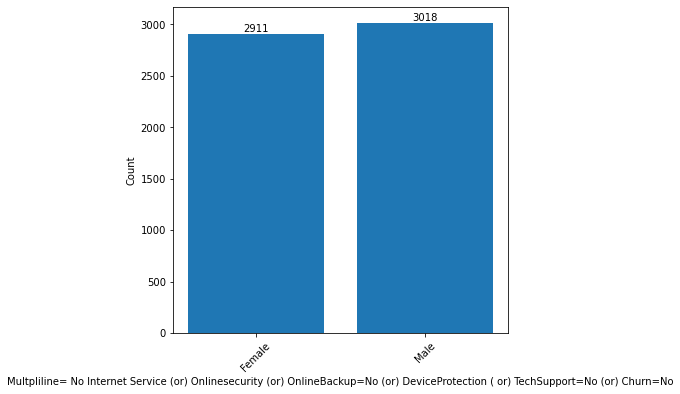
plt.xticks(range(len(df1\_NointernetService\_No)), df1\_NointernetService\_No['gender'], rotation=45)

for i, count in enumerate(df1\_NointernetService\_No["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt.show()

**Bar Plot:**

****

**AQ2.** Filter and count based on the following condition

* Multiplelines = No internet Service
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport = 'No'
* Churn rate = Yes
* Group by Gender

**Code:**

df1= df[(df['MultipleLines'] == 'No internetService')|(df['OnlineSecurity']=='No')| (df['OnlineBackup']=='No') | (df['DeviceProtection']=='No')| (df['TechSupport']=='No')| (df['Churn'] =='Yes')]

df1\_NointernetService\_Yes = df1.groupby("gender").size().reset\_index(name= 'count')

df1\_NointernetService\_Yes

# plot Bar chart

plt.figure(figsize=(6,6))

plt.bar(range(len(df1\_NointernetService\_Yes)), df1\_NointernetService\_Yes['count'])

plt.xlabel("Multpliline= No internetService or Onlinesecurity=No or OnlineBackup=No or DeviceProtection=No, or TechSupport=No or Churn=Yes")

plt.ylabel("Count")

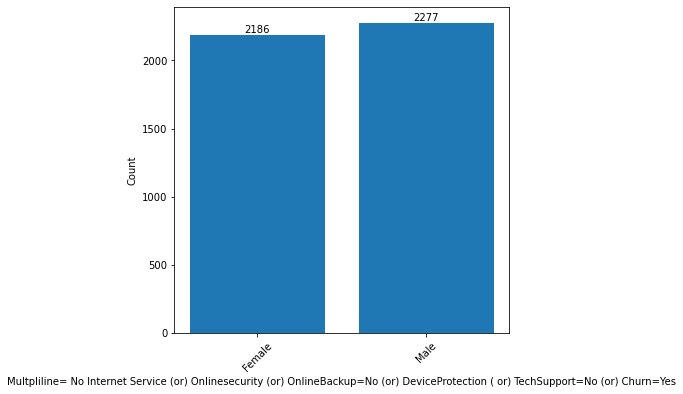
plt.xticks(range(len(df1\_NointernetService\_Yes)), df1\_NointernetService\_Yes['gender'], rotation=45)

for i, count in enumerate(df1\_NointernetService\_Yes["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt.show()

**Bar Plot:**

****

**AQ3.** Filter and count based on the following condition

* Multiplelines = No
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport = 'No'
* Churn rate = No
* Group by Gender

**Code:**

df1= df[(df['MultipleLines'] == 'No')|(df['OnlineSecurity']=='No')|(df['OnlineBackup']=='No') | (df['DeviceProtection']=='No')| (df['TechSupport']=='No')| (df['Churn'] =='No')]

df1\_No\_No = df1.groupby("gender").size().reset\_index(name= 'count')

df1\_No\_No

# plot Bar chart

plt.figure(figsize=(6,6))

plt.bar(range(len(df1\_No\_No)), df1\_No\_No['count'])

plt.xlabel("Multpliline= No (or) Onlinesecurity=No or OnlineBackup=No or DeviceProtection=No, or TechSupport=No (or) Churn=No")

plt.ylabel("Count")

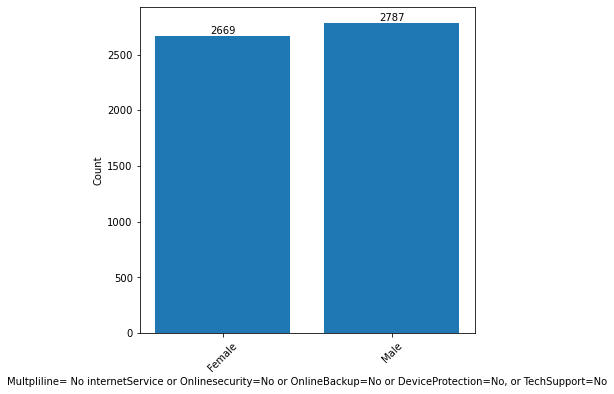
plt.xticks(range(len(df1\_No\_No)), df1\_No\_No['gender'], rotation=45)

for i, count in enumerate(df1\_No\_No["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt.show()

**Bar Plot:**

****

**AQ4.** Filter and count based on the following condition

* Multiplelines = No
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport = 'No'
* Churn rate = Yes
* Group by Gender

**Code:**

df1= df[(df['MultipleLines'] == 'No')|(df['OnlineSecurity']=='No')|(df['OnlineBackup']=='No') | (df['DeviceProtection']=='No')| (df['TechSupport']=='No')| (df['Churn'] =='Yes')]

df1\_No\_Yes = df1.groupby("gender").size().reset\_index(name= 'count')

df1\_No\_Yes

# plot Bar chart

plt.figure(figsize=(6,6))

plt.bar(range(len(df1\_No\_Yes)), df1\_No\_Yes['count'])

plt.xlabel("Multpliline= No (or) Onlinesecurity=No or OnlineBackup=No or DeviceProtection=No, or TechSupport=No, Churn=Yes")

plt.ylabel("Count")

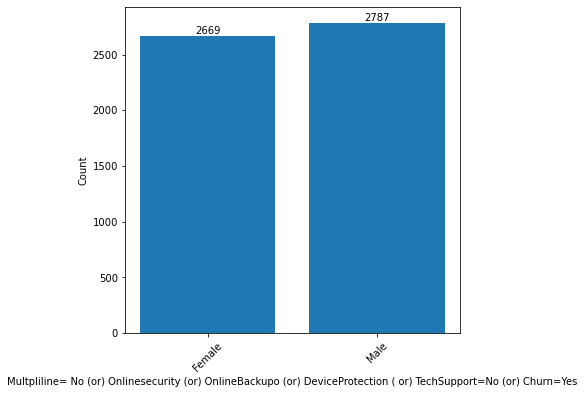
plt.xticks(range(len(df1\_No\_Yes)), df1\_No\_Yes['gender'], rotation=45)

for i, count in enumerate(df1\_No\_Yes["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt.show()

**Bar Plot:**

****

**AQ5.** Filter and count based on the following condition

* Multiplelines = No
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport = 'No internet Service'
* Churn rate = No
* Group by Gender

**Code:**

df1= df[(df['MultipleLines'] == 'No')|(df['OnlineSecurity']=='No internet Service') | (df['OnlineBackup']=='No internet Service') | (df['DeviceProtection']=='No internet Service')| (df['TechSupport']=='No internet Service')| (df['Churn'] =='No')]

df1\_No\_Nointernetsevice\_No = df1.groupby("gender").size().reset\_index(name= 'count')

df1\_No\_Nointernetsevice\_No

# plot Bar chart

plt.figure(figsize=(6,6))

plt.bar(range(len(df1\_No\_Nointernetsevice\_No)), df1\_No\_Nointernetsevice\_No['count'])

plt.xlabel("Multpliline= No (or) Onlinesecurity (or) OnlineBackup=No (or) DeviceProtection ( or) TechSupport=No Internet Service (or) Churn=No")

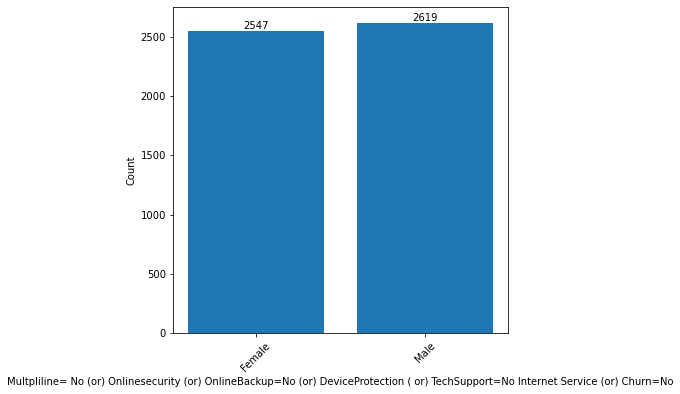
plt.ylabel("Count")

plt.xticks(range(len(df1\_No\_Nointernetsevice\_No)), df1\_No\_Nointernetsevice\_No['gender'], rotation=45)

for i, count in enumerate(df1\_No\_Nointernetsevice\_No["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt.show() **Bar Plot:**

****

**AQ6.** Filter and count based on the following condition

* Multiplelines = No
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport = 'No internet Service'
* Churn rate = Yes
* Group by Gender

**Code:**

df1= df[(df['MultipleLines'] == 'No')|(df['OnlineSecurity']=='No internet Service')|(df['OnlineBackup']=='No internet Service') | (df['DeviceProtection']=='No internet Service')| (df['TechSupport']=='No internet Service')| (df['Churn'] =='Yes')]

df1\_No\_Nointernetsevice\_Yes = df1.groupby("gender").size().reset\_index(name= 'count')

df1\_No\_Nointernetsevice\_Yes

# plot Bar chart

plt.figure(figsize=(6,6))

plt.bar(range(len(df1\_No\_Nointernetsevice\_Yes)), df1\_No\_Nointernetsevice\_Yes['count'])

plt.xlabel("Multpliline= No (or) Onlinesecurity (or) OnlineBackup=No (or) DeviceProtection ( or) TechSupport=No Internet Service (or) Churn=Yes")

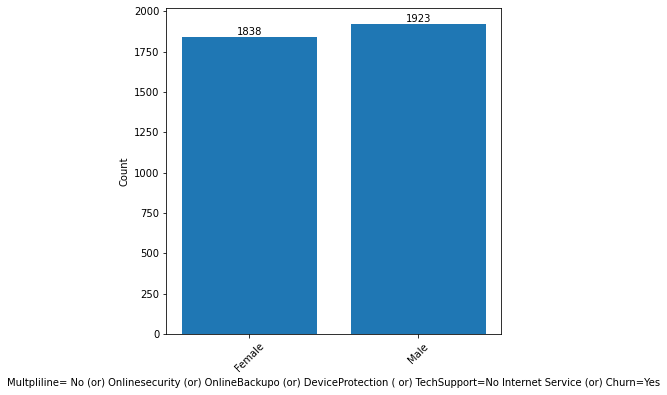
plt.ylabel("Count")

plt.xticks(range(len(df1\_No\_Nointernetsevice\_Yes)), df1\_No\_Nointernetsevice\_Yes['gender'], rotation=45)

for i, count in enumerate(df1\_No\_Nointernetsevice\_Yes["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt.show() **Bar Plot:**

****

**AQ7.** Filter and count based on the following condition

* Multiplelines = Yes
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport = 'No internet Service'
* Churn rate = No
* Group by Gender

**Code:**

df1= df[(df['MultipleLines'] == 'Yes')|(df['OnlineSecurity']=='No internet Service')|(df['OnlineBackup']=='No internet Service') | (df['DeviceProtection']=='No internet Service')| (df['TechSupport']=='No internet Service')| (df['Churn'] =='No')]

df1\_Yes\_Nointernetsevice\_No = df1.groupby("gender").size().reset\_index(name= 'count')

df1\_Yes\_Nointernetsevice\_No

# plot Bar chart

plt.figure(figsize=(6,6))

plt.bar(range(len(df1\_Yes\_Nointernetsevice\_No)), df1\_Yes\_Nointernetsevice\_No['count'])

plt.xlabel("Multpliline= Yes (or) Onlinesecurity (or) OnlineBackup=No (or) DeviceProtection ( or) TechSupport=No Internet Service (or) Churn=No")

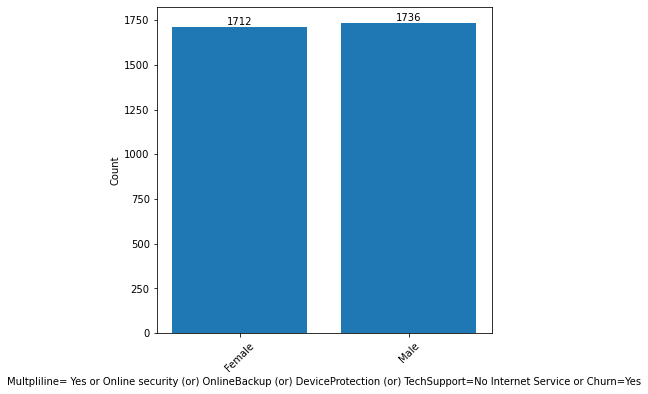
plt.ylabel("Count")

plt.xticks(range(len(df1\_Yes\_Nointernetsevice\_No)), df1\_Yes\_Nointernetsevice\_No['gender'], rotation=45)

for i, count in enumerate(df1\_Yes\_Nointernetsevice\_No["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt.show() **Bar Plot:**

****

**AQ8.** Filter and count based on the following condition

* Multiplelines = Yes
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport = 'No internet Service'
* Churn rate = Yes
* Group by Gender

**Code:**

df1= df[(df['MultipleLines'] == 'Yes')|(df['OnlineSecurity']=='No internet Service')|(df['OnlineBackup']=='No internet Service') | (df['DeviceProtection']=='No internet Service')| (df['TechSupport']=='No internet Service')| (df['Churn'] =='Yes')]

df1\_Yes\_Nointernetsevice\_Yes = df1.groupby("gender").size().reset\_index(name= 'count')

df1\_Yes\_Nointernetsevice\_Yes

# plot Bar chart

plt.figure(figsize=(6,6))

plt.bar(range(len(df1\_Yes\_Nointernetsevice\_Yes)), df1\_Yes\_Nointernetsevice\_Yes['count'])

plt.xlabel("Multpliline= Yes or Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport=No Internet Service or Churn=Yes")

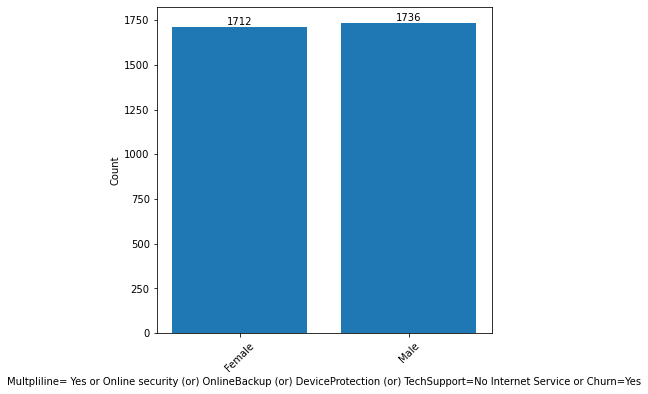
plt.ylabel("Count")

plt.xticks(range(len(df1\_Yes\_Nointernetsevice\_Yes)), df1\_Yes\_Nointernetsevice\_Yes['gender'], rotation=45)

for i, count in enumerate(df1\_Yes\_Nointernetsevice\_Yes["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt.show() **Bar Plot:**

****

**AQ9.** Filter and count based on the following condition

* Multiplelines = Yes
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport = 'No'
* Churn rate = No
* Group by Gender

**Code:**

df1= df[(df['MultipleLines'] == 'Yes')|(df['OnlineSecurity']=='No') |(df['OnlineBackup'] == 'No') | (df['DeviceProtection']=='No')| (df['TechSupport']=='No')| (df['Churn'] =='No')]

df1\_Yes\_No\_No = df1.groupby("gender").size().reset\_index(name= 'count')

df1\_Yes\_No\_No

# plot Bar chart

plt.figure(figsize=(6,6))

plt.bar(range(len(df1\_Yes\_No\_No)), df1\_Yes\_No\_No['count'])

plt.xlabel("Multpliline= (or) Yes Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport=No (or) Churn=No")

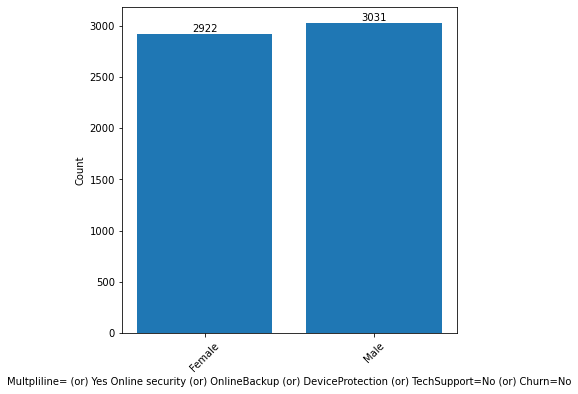
plt.ylabel("Count")

plt.xticks(range(len(df1\_Yes\_No\_No)), df1\_Yes\_No\_No['gender'], rotation=45)

for i, count in enumerate(df1\_Yes\_No\_No["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt.show() **Bar Plot:**

****

**AQ10.** Filter and count based on the following condition

* Multiplelines = Yes
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport = 'No'
* Churn rate = Yes
* Group by Gender

**Code:**

df1= df[(df['MultipleLines'] == 'Yes')|(df['OnlineSecurity']== 'No')|(df['OnlineBackup']== 'No') | (df['DeviceProtection']=='No')| (df['TechSupport']=='No')| (df['Churn'] =='Yes')]

df1\_Yes\_No\_Yes = df1.groupby("gender").size().reset\_index(name= 'count')

df1\_Yes\_No\_Yes

# plot Bar chart

plt.figure(figsize=(6,6))

plt.bar(range(len(df1\_Yes\_No\_Yes)), df1\_Yes\_No\_Yes['count'])

plt.xlabel("Multpliline= Yes (or) Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport=No (or) Churn=Yes")

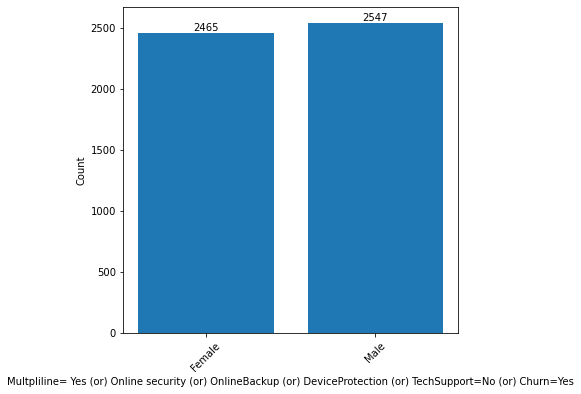
plt.ylabel("Count")

plt.xticks(range(len(df1\_Yes\_No\_Yes)), df1\_Yes\_No\_Yes['gender'], rotation=45)

for i, count in enumerate(df1\_Yes\_No\_Yes["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt.show() **Bar Plot:**

****

**AQ11.** Filter and count based on the following condition

* Multiplelines = Yes
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport = 'No'
* Churn rate = Yes
* Group by Gender and Senior Citizen

**Code:**

df1= df[(df['MultipleLines'] == 'Yes')|(df['OnlineSecurity']=='No')|(df['OnlineBackup']== 'No') | (df['DeviceProtection']=='No')| (df['TechSupport']=='No')| (df['Churn'] =='Yes')]

df1\_Yes\_No\_Yes = df1.groupby(["gender","SeniorCitizen"]).size().reset\_index(name= 'count')

df1\_Yes\_No\_Yes

# Prepare data for stacked bar chart

labels = df1\_Yes\_No\_Yes.apply(lambda x: f'{x["gender"]}\n({x["SeniorCitizen"]})', axis=1)

count = df1\_Yes\_No\_Yes["count"]

# Plot stacked bar chart

plt.figure(figsize=(8, 6))

plt.bar(labels, count, label="Count")

# Add labels and format the chart

plt.xlabel("Gender (Senior Citizen)")

plt.ylabel("Count")

plt.title("Stacked Bar Chart")

plt.legend()

# Display values on top of bars

for i, val in enumerate(count):

plt.text(i, val + 10, str(val), ha="center", va="top")

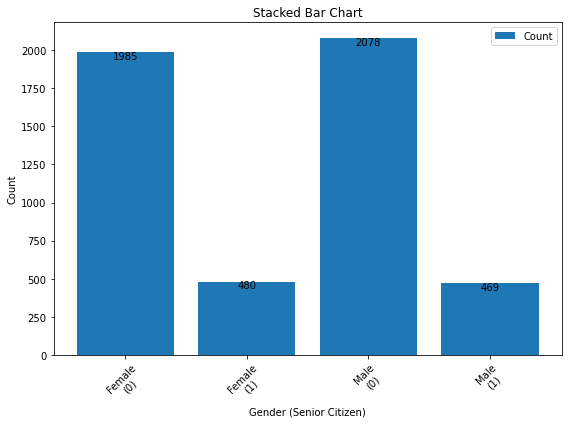
# Rotate x-axis labels if needed

plt.xticks(rotation=45)

# Show the chart

plt.tight\_layout()

plt.show() **Bar Plot:**

****

**AQ12.** Filter and count based on the following condition

* Multiplelines = Yes
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport t = 'No'
* Churn rate = No
* Group by Gender, Senior Citizen

**Code:**

df1= df[(df['MultipleLines'] == 'Yes')|(df['OnlineSecurity']=='No')|(df['OnlineBackup']== 'No') | (df['DeviceProtection']=='No')| (df['TechSupport']=='No')| (df['Churn'] =='No')]

df1\_Yes\_No\_No = df1.groupby(["gender", "SeniorCitizen"]).size().reset\_index(name= 'count')

df1\_Yes\_No\_No

# plot Bar chart

plt.figure(figsize=(6,6))

plt.bar(range(len(df1\_Yes\_No\_No)), df1\_Yes\_No\_No['count'])

plt.xlabel("Multpliline= No • Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport =No (or) Churn=No")

plt.ylabel("Count")

plt.xticks(range(len(df1\_Yes\_No\_No)), df1\_Yes\_No\_No['gender'], rotation=45)

for i, count in enumerate(df1\_Yes\_No\_No["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt.show() plt.text(i, val + 10, str(val), ha="center", va="top")

# Rotate x-axis labels if needed

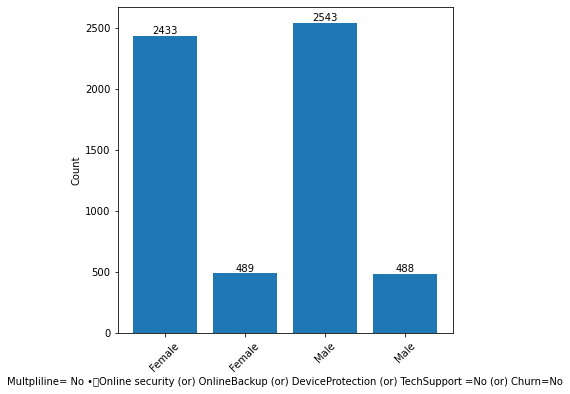
plt.xticks(rotation=45)

# Show the chart

plt.tight\_layout()

plt.show()

**Bar Plot:**

****

**AQ13.** Filter and count based on the following condition

* Multiplelines = Yes
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport = 'No internet Service'
* Churn rate = Yes
* Group by Gender, Senior Citizen

**Code:**

df1= df[(df['MultipleLines'] == 'Yes')|(df['OnlineSecurity']=='No internet Service')|(df['OnlineBackup'] == 'No internet Service') | (df['DeviceProtection']=='No internet Service')| (df['TechSupport']=='No internet Service')| (df['Churn'] =='Yes')]

df1\_Yes\_Nointernetsevice\_Yes = df1.groupby(["gender", "SeniorCitizen"]).size().reset\_index(name= 'count')

df1\_Yes\_Nointernetsevice\_Yes

# plot Bar chart

plt.figure(figsize=(6,6))

plt.bar(range(len(df1\_Yes\_Nointernetsevice\_Yes)), df1\_Yes\_Nointernetsevice\_Yes['count'])

plt.xlabel("Multpliline= Yes (or) Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport=No Internet Service (or Churn=Yes)")

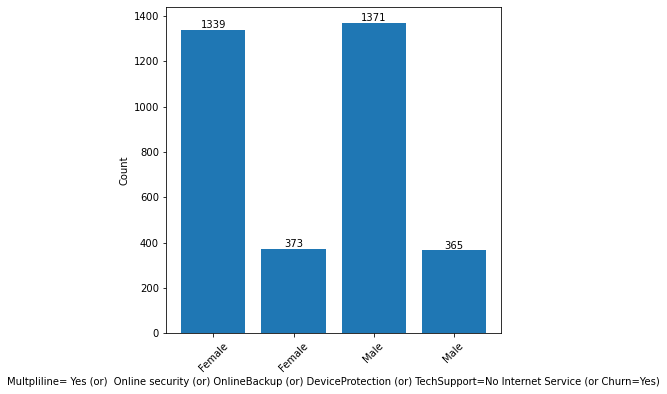
plt.ylabel("Count")

plt.xticks(range(len(df1\_Yes\_Nointernetsevice\_Yes)), df1\_Yes\_Nointernetsevice\_Yes['gender'], rotation=45)

for i, count in enumerate(df1\_Yes\_Nointernetsevice\_Yes["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt.show() **Bar Plot:**

****

**AQ14.** Filter and count based on the following condition

* Multiplelines = Yes
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport = 'No internet Service'
* Churn rate = No
* Group by Gender, Senior Citizen

**Code:**

df1= df[(df['MultipleLines'] == 'Yes')|(df['OnlineSecurity']=='No internet Service')|(df['OnlineBackup']=='No internet Service') | (df['DeviceProtection']=='No internet Service')| (df['TechSupport']=='No internet Service')| (df['Churn'] =='No')]

df1\_Yes\_Nointernetsevice\_No = df1.groupby(["gender", "SeniorCitizen"]).size().reset\_index(name= 'count')

df1\_Yes\_Nointernetsevice\_No

# plot Bar chart

plt.figure(figsize=(6,6))

plt.bar(range(len(df1\_Yes\_Nointernetsevice\_No)), df1\_Yes\_Nointernetsevice\_No['count'])

plt.xlabel("Multpliline= Yes (or) Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport=No Internet Service (or) Churn=No")

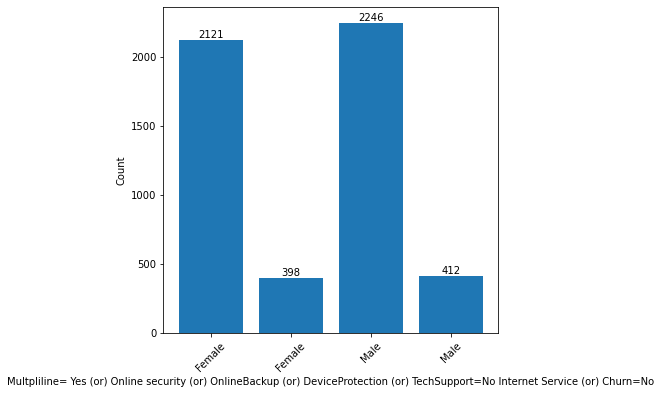
plt.ylabel("Count")

plt.xticks(range(len(df1\_Yes\_Nointernetsevice\_No)), df1\_Yes\_Nointernetsevice\_No['gender'], rotation=45)

for i, count in enumerate(df1\_Yes\_Nointernetsevice\_No["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt.show() **Bar Plot:**

****

**AQ15.** Filter and count based on the following condition

* Multiplelines = No
* Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport = 'No internet Service'
* Churn rate = Yes
* Group by Gender, Senior Citizen

**Code:**

df1= df[(df['MultipleLines'] == 'No')|(df['OnlineSecurity']=='No internet Service')| (df['OnlineBackup']=='No internet Service') | (df['DeviceProtection']=='No internet Service')| (df['TechSupport']=='No internet Service')| (df['Churn'] =='Yes')]

df1\_No\_Nointernetsevice\_Yes = df1.groupby(["gender", "SeniorCitizen"]).size().reset\_index(name= 'count')

df1\_No\_Nointernetsevice\_Yes

# plot Bar chart

plt.figure(figsize=(6,6))

plt.bar(range(len(df1\_No\_Nointernetsevice\_Yes)), df1\_No\_Nointernetsevice\_Yes['count'])

plt.xlabel("Multpliline= No (or) Online security (or) OnlineBackup (or) DeviceProtection (or) TechSupport=No" (or) Churn=Yes)

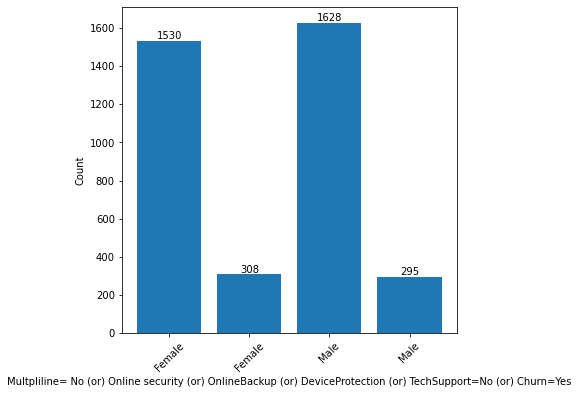
plt.ylabel("Count")

plt.xticks(range(len(df1\_No\_Nointernetsevice\_Yes)), df1\_No\_Nointernetsevice\_Yes['gender'], rotation=45)

for i, count in enumerate(df1\_No\_Nointernetsevice\_Yes["count"]):

plt.text(i,count,str(count), ha="center", va = "bottom")

plt.show() **Bar Plot:**

****

**Insight**

* Number of Female customers who have discontinued their telecom services is high with respect to monthly charges is greater than 75 and both gender than Male
* Number of Male customers who have continued their telecom services is high with respect to monthly charges greater than 75 than females.
* Number of Male Non-senior citizen customers who have continued their telecom services is high with respect to monthly charges greater than 75 than Female.
* Number of Male Non-senior citizen customer who have continued their telecom services is high with respect to monthly charges less than 75 than Female.
* Males are continuing their telecom services though no internet service, Online backup, Online security, Device protection, Tech support.