Assignment - 8 PySpark SQL

September 25, 2024

[1]:
```
sc
```

[1]:
```
<SparkContext master=local[*] appName=PySparkShell>
```

[2]:
```
spark
```

[2]:
```
<pyspark.sql.session.SparkSession at 0x7fe215cfd0f0>
```

[3]:
```
sc.stop()
from pyspark import SparkConf,SparkContext
config = SparkConf().setMaster("local[4]").setAppName("PysparkAssignment8")
sc = SparkContext(conf=config)
```

[4]:
```
from pyspark.sql import SparkSession
#getorcreate
spark = SparkSession.builder.appName("PysparkAssignment8").getOrCreate()
```

[5]:
```
spark
```

[5]:
```
<pyspark.sql.session.SparkSession at 0x7fe215cfdfd0>
```

[6]:
```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

[7]:
```
telco_data = spark.read.csv("file:///home/hadoop/Downloads/Telco_Customer_Churn.
 ↪csv",
                            inferSchema=True, header=True)
```

[8]:
```
telco_data.show()
```

```
+----------+------+-------------+-------+----------+------+------------+--------
--------+--------------+------------------+------------------+--------------
----+------------------+------------------+------------------+--------------+
---------------+------------------+--------------+------------+-----+
|customerID|gender|SeniorCitizen|Partner|Dependents|tenure|PhoneService|
MultipleLines|InternetService|     OnlineSecurity|       OnlineBackup|
DeviceProtection|        TechSupport|        StreamingTV|    StreamingMovies|
```

```
Contract|PaperlessBilling|
PaymentMethod|MonthlyCharges|TotalCharges|Churn|
+----------+------+------------+-------+----------+------+-----------+--------
--------+--------------+------------------+------------------+--------------
----+------------------+------------------+------------------+-------------+
---------------+------------------+-------------+-----------+-----+
|7590-VHVEG|Female|            0|    Yes|        No|     1|          No|No phone
service|            DSL|                No|               Yes|
No|                No|               No|                No|Month-to-month|
Yes|    Electronic check|         29.85|       29.85|   No|
|5575-GNVDE|  Male|            0|     No|        No|    34|         Yes|
No|            DSL|               Yes|                No|               Yes|
No|                No|               No|      One year|              No|
Mailed check|         56.95|      1889.5|   No|
|3668-QPYBK|  Male|            0|     No|        No|     2|         Yes|
No|            DSL|               Yes|               Yes|                No|
No|                No|               No|Month-to-month|             Yes|
Mailed check|         53.85|      108.15|  Yes|
|7795-CFOCW|  Male|            0|     No|        No|    45|          No|No phone
service|            DSL|               Yes|                No|
Yes|               Yes|                No|                No|      One year|
No|Bank transfer (au…|          42.3|     1840.75|   No|
|9237-HQITU|Female|            0|     No|        No|     2|         Yes|
No|    Fiber optic|                No|                No|                No|
No|                No|               No|Month-to-month|             Yes|
Electronic check|          70.7|      151.65|  Yes|
|9305-CDSKC|Female|            0|     No|        No|     8|         Yes|
Yes|    Fiber optic|                No|                No|               Yes|
No|               Yes|               Yes|Month-to-month|             Yes|
Electronic check|         99.65|       820.5|  Yes|
|1452-KIOVK|  Male|            0|     No|       Yes|    22|         Yes|
Yes|    Fiber optic|                No|               Yes|                No|
No|                Yes|               No|Month-to-month|
Yes|Credit card (auto…|          89.1|      1949.4|   No|
|6713-OKOMC|Female|            0|     No|        No|    10|          No|No phone
service|            DSL|               Yes|                No|
No|                No|               No|               No|Month-to-month|
No|        Mailed check|         29.75|       301.9|   No|
|7892-POOKP|Female|            0|    Yes|        No|    28|         Yes|
Yes|    Fiber optic|                No|                No|               Yes|
Yes|               Yes|               Yes|Month-to-month|             Yes|
Electronic check|         104.8|     3046.05|  Yes|
|6388-TABGU|  Male|            0|     No|       Yes|    62|         Yes|
No|            DSL|               Yes|               Yes|                No|
No|                No|               No|      One year|              No|Bank
transfer (au…|         56.15|     3487.95|   No|
|9763-GRSKD|  Male|            0|    Yes|       Yes|    13|         Yes|
No|            DSL|               Yes|                No|                No|
```

```
No|                 No|                 No|Month-to-month|              Yes|
Mailed check|         49.95|      587.45|   No|
|7469-LKBCI|  Male|            0|     No|        No|    16|         Yes|
No|             No|No internet service|No internet service|No internet
service|No internet service|No internet service|No internet service|      Two
year|              No|Credit card (auto…|         18.95|       326.8|   No|
|8091-TTVAX|  Male|            0|    Yes|        No|    58|         Yes|
Yes|    Fiber optic|                 No|                 No|               Yes|
No|                Yes|                Yes|      One year|
No|Credit card (auto…|        100.35|      5681.1|   No|
|0280-XJGEX|  Male|            0|     No|        No|    49|         Yes|
Yes|    Fiber optic|                 No|                Yes|               Yes|
No|                Yes|                Yes|Month-to-month|             Yes|Bank
transfer (au…|         103.7|      5036.3|  Yes|
|5129-JLPIS|  Male|            0|     No|        No|    25|         Yes|
No|    Fiber optic|                Yes|                 No|              Yes|
Yes|                Yes|                Yes|Month-to-month|             Yes|
Electronic check|         105.5|     2686.05|   No|
|3655-SNQYZ|Female|            0|    Yes|       Yes|    69|         Yes|
Yes|    Fiber optic|                Yes|                Yes|               Yes|
Yes|                Yes|                Yes|      Two year|
No|Credit card (auto…|        113.25|     7895.15|   No|
|8191-XWSZG|Female|            0|     No|        No|    52|         Yes|
No|             No|No internet service|No internet service|No internet
service|No internet service|No internet service|No internet service|      One
year|              No|        Mailed check|         20.65|     1022.95|   No|
|9959-WOFKT|  Male|            0|     No|       Yes|    71|         Yes|
Yes|    Fiber optic|                Yes|                 No|               Yes|
No|                Yes|                Yes|      Two year|              No|Bank
transfer (au…|         106.7|     7382.25|   No|
|4190-MFLUW|Female|            0|    Yes|       Yes|    10|         Yes|
No|            DSL|                 No|                 No|              Yes|
Yes|                 No|                No|Month-to-month|
No|Credit card (auto…|          55.2|      528.35|  Yes|
|4183-MYFRB|Female|            0|     No|        No|    21|         Yes|
No|    Fiber optic|                 No|                Yes|              Yes|
No|                 No|                Yes|Month-to-month|             Yes|
Electronic check|         90.05|      1862.9|   No|
+----------+------+------------+-------+----------+------+------------+--------
--------+---------------+------------------+------------------+---------------
----+------------------+------------------+------------------+--------------+
---------------+--------------------+--------------+------------+-----+
only showing top 20 rows
```

[9]:
```
telco_data.createOrReplaceTempView('TelcoData')
```

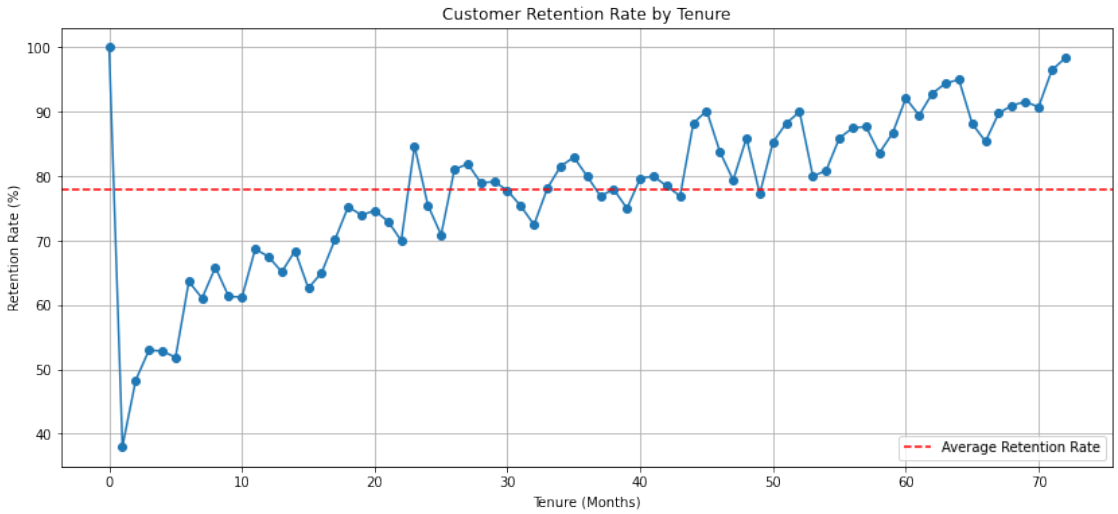
[10]:
```
spark.sql("select * from TelcoData").show(5)
```

```
+----------+------+-------------+-------+----------+------+-----------+--------
--------+--------------+-------------+-----------+---------------+----------
-+-----------+---------------+-------------+---------------+-----------------
--+--------------+------------+-----+
|customerID|gender|SeniorCitizen|Partner|Dependents|tenure|PhoneService|   Multi
pleLines|InternetService|OnlineSecurity|OnlineBackup|DeviceProtection|TechSuppor
t|StreamingTV|StreamingMovies|      Contract|PaperlessBilling|
PaymentMethod|MonthlyCharges|TotalCharges|Churn|
+----------+------+-------------+-------+----------+------+-----------+--------
--------+--------------+-------------+-----------+---------------+----------
-+-----------+---------------+-------------+---------------+-----------------
--+--------------+------------+-----+
|7590-VHVEG|Female|            0|    Yes|        No|     1|          No|No phone
service|            DSL|             No|         Yes|              No|
No|         No|             No|Month-to-month|             Yes|    Electronic
check|         29.85|       29.85|   No|
|5575-GNVDE|  Male|            0|     No|        No|    34|         Yes|
No|            DSL|           Yes|          No|              Yes|         No|
No|             No|      One year|              No|        Mailed check|
56.95|      1889.5|   No|
|3668-QPYBK|  Male|            0|     No|        No|     2|         Yes|
No|            DSL|           Yes|         Yes|              No|         No|
No|             No|Month-to-month|             Yes|        Mailed check|
53.85|      108.15|  Yes|
|7795-CFOCW|  Male|            0|     No|        No|    45|          No|No phone
service|            DSL|           Yes|          No|             Yes|
Yes|         No|             No|      One year|              No|Bank transfer
(au…|          42.3|     1840.75|   No|
|9237-HQITU|Female|            0|     No|        No|     2|         Yes|
No|    Fiber optic|            No|          No|              No|         No|
No|             No|Month-to-month|             Yes|    Electronic check|
70.7|      151.65|  Yes|
+----------+------+-------------+-------+----------+------+-----------+--------
--------+--------------+-------------+-----------+---------------+----------
-+-----------+---------------+-------------+---------------+-----------------
--+--------------+------------+-----+
only showing top 5 rows
```

0.0.1 A. Analyze how customer retention varies based on how long the customer has stayed with the company (tenure).

[11]:
```
res = spark.sql("""SELECT
        tenure,
        COUNT(customerID) AS total_customers,
        SUM(CASE WHEN Churn = "No" THEN 1 ELSE 0 END) AS retained_customers,
        (SUM(CASE WHEN Churn = "No" THEN 1 ELSE 0 END) / COUNT(customerID)) *␣
 ↪100
        AS retention_rate
    FROM
        TelcoData
    GROUP BY
        tenure
    ORDER BY
        tenure
""").toPandas()
```

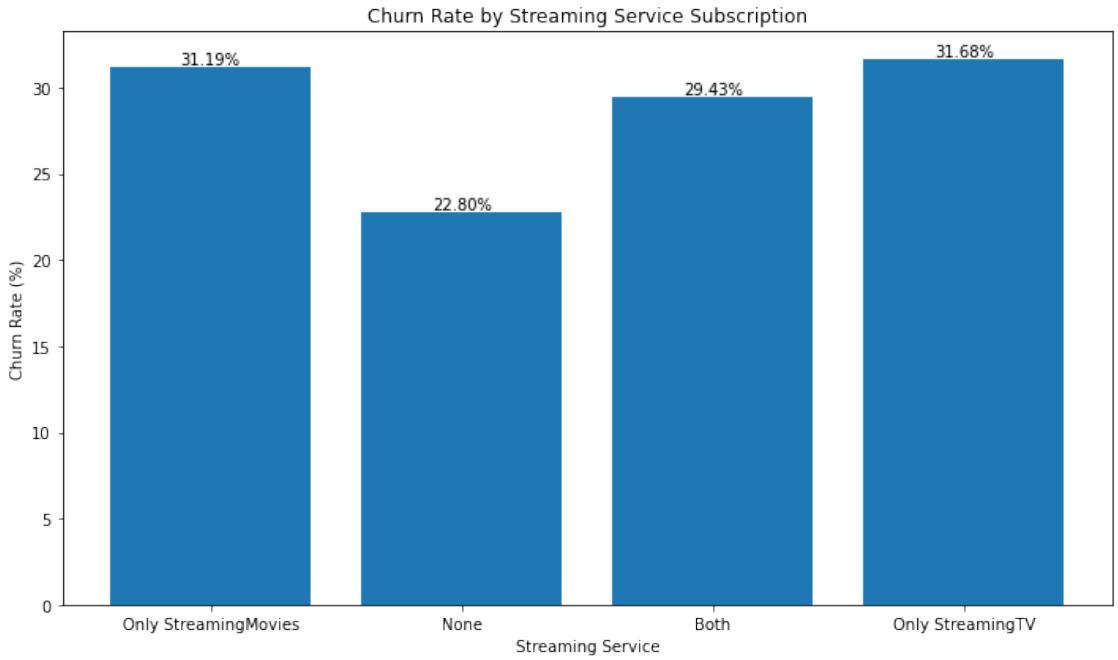
[12]:
```
plt.figure(figsize=(14, 6))
plt.plot(res['tenure'], res['retention_rate'], marker='o', linestyle='-')
plt.title('Customer Retention Rate by Tenure')
plt.xlabel('Tenure (Months)')
plt.ylabel('Retention Rate (%)')

plt.grid()
plt.axhline(y=res['retention_rate'].mean(), color='r', linestyle='--',␣
 ↪label='Average Retention Rate')
plt.legend()
plt.show()
```



0.0.2 B. Investigate the churn rate of customers who subscribe to streaming services like StreamingTV and StreamingMovies.

[13]:
```
res = """
        SELECT
        CASE
            WHEN StreamingTV = 'Yes' AND StreamingMovies = 'Yes' THEN 'Both'
            WHEN StreamingTV = 'Yes' THEN 'Only StreamingTV'
            WHEN StreamingMovies = 'Yes' THEN 'Only StreamingMovies'
            ELSE 'None'
        END AS streaming_service,
        COUNT(customerID) AS total_customers,
        SUM(CASE WHEN Churn = "Yes" THEN 1 ELSE 0 END) AS churned_customers,
        (SUM(CASE WHEN Churn = "Yes" THEN 1 ELSE 0 END) / COUNT(customerID)) *␣
 ↪100 AS churn_rate
    FROM
        TelcoData
    GROUP BY
        streaming_service
"""

result=spark.sql(res)
result.show()
result = result.toPandas()

plt.figure(figsize=(10, 6))
plt.bar(result['streaming_service'], result['churn_rate'])

plt.title('Churn Rate by Streaming Service Subscription')
plt.xlabel('Streaming Service')
plt.ylabel('Churn Rate (%)')

for i, v in enumerate(result['churn_rate']):
    plt.text(i, v, f'{v:.2f}%', ha='center', va='bottom')

plt.tight_layout()
plt.show()
```

```
+-------------------+--------------+-----------------+------------------+
|   streaming_service|total_customers|churned_customers|        churn_rate|
+-------------------+--------------+-----------------+------------------+
|Only StreamingMovies|           792|              247|31.186868686868685|
|               None|          3544|              808|22.799097065462753|
|               Both|          1940|              571| 29.43298969072165|
|    Only StreamingTV|           767|              243|31.681877444589308|
+-------------------+--------------+-----------------+------------------+
```

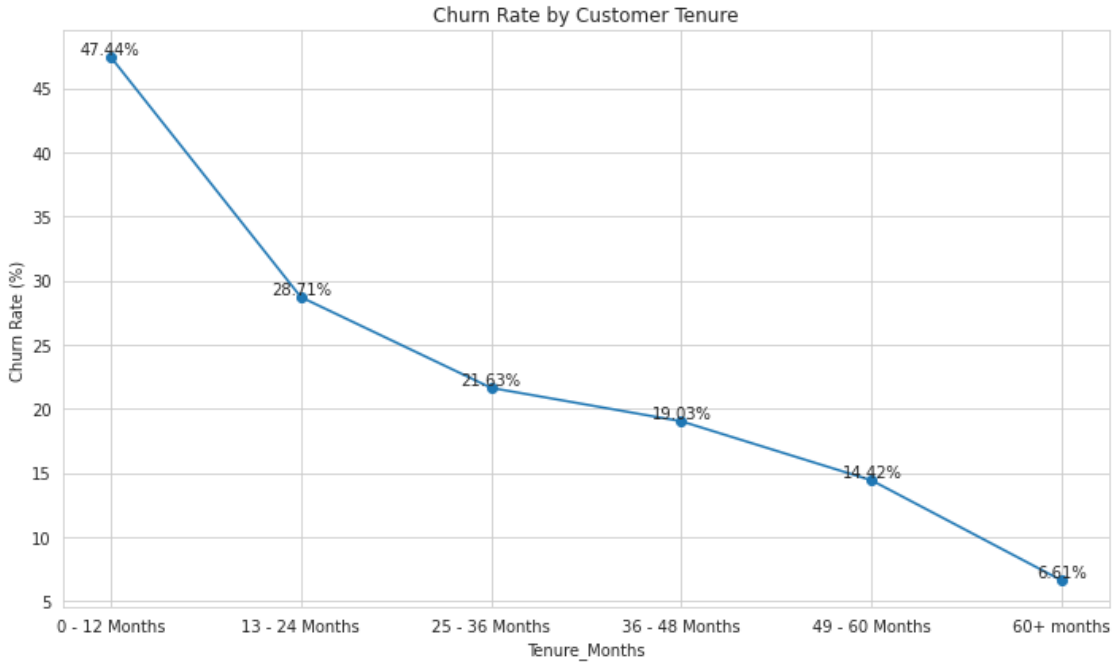


0.0.3 C. Write Spark SQL to group customers by their tenure (e.g., 0-12 months, 13-24 months, etc.) and analyze churn rates in these tenure groups.

[47]:
```
res = spark.sql("""
    SELECT
    CASE
        WHEN tenure BETWEEN 0 AND 12 THEN '0 - 12 Months'
        WHEN tenure BETWEEN 13 AND 24 THEN '13 - 24 Months'
        WHEN tenure BETWEEN 25 AND 36 THEN '25 - 36 Months'
        WHEN tenure BETWEEN 36 AND 48 THEN '36 - 48 Months'
        WHEN tenure BETWEEN 49 AND 60 THEN '49 - 60 Months'
        WHEN tenure > 60 THEN '60+ months'
    END AS Tenure_Months,
    COUNT(customerID) AS total_customers,
        SUM(CASE WHEN Churn = "Yes" THEN 1 ELSE 0 END) AS churned_customers,
        (SUM(CASE WHEN Churn = "Yes" THEN 1 ELSE 0 END) / COUNT(customerID)) *
        100 AS churn_rate
    FROM
        TelcoData
    GROUP BY
        Tenure_Months
    ORDER BY
        Tenure_Months
""")
```

```
res.show()


result = res.toPandas()

plt.figure(figsize=(10, 6))
plt.plot(result['Tenure_Months'], result['churn_rate'], marker='o',␣
 ↪linestyle='-')


plt.title('Churn Rate by Customer Tenure')
plt.xlabel('Tenure_Months')
plt.ylabel('Churn Rate (%)')

for i, v in enumerate(result['churn_rate']):
    plt.text(i, v, f'{v:.2f}%', ha='center', va='bottom')

plt.tight_layout()
plt.show()
```

```
+--------------+---------------+-----------------+------------------+
| Tenure_Months|total_customers|churned_customers|        churn_rate|
+--------------+---------------+-----------------+------------------+
| 0 - 12 Months|           2186|             1037|47.438243366880144|
|13 - 24 Months|           1024|              294|        28.7109375|
|25 - 36 Months|            832|              180|21.634615384615387|
|36 - 48 Months|            762|              145|19.028871391076116|
|49 - 60 Months|            832|              120|14.423076923076922|
|    60+ months|           1407|               93| 6.609808102345416|
+--------------+---------------+-----------------+------------------+
```

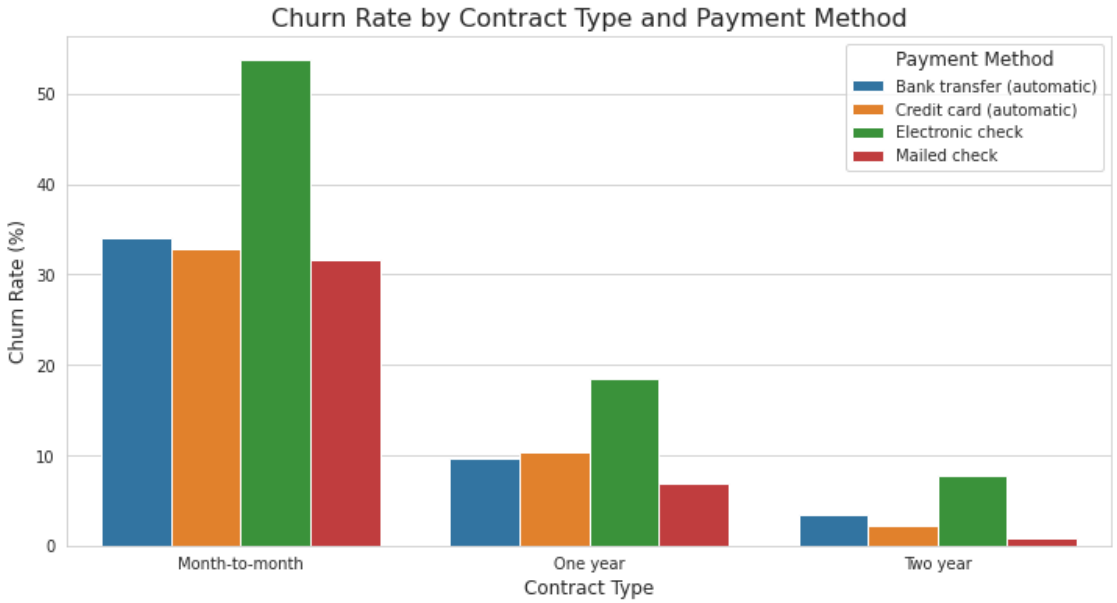


0.0.4 D. Analyze the impact of contract types and payment methods on churn rates.

[40]:
```
res = """
    SELECT
        Contract,
        PaymentMethod,
        COUNT(customerID) AS total_customers,
            SUM(CASE WHEN Churn = "Yes" THEN 1 ELSE 0 END) AS churned_customers,
            (SUM(CASE WHEN Churn = "Yes" THEN 1 ELSE 0 END) /␣
 ↪COUNT(customerID)) * 100 AS churn_rate
        FROM
            TelcoData
        GROUP BY
            Contract, PaymentMethod
        ORDER BY
            Contract, PaymentMethod
"""
result = spark.sql(res)
result.show()

df= result.toPandas()
plt.figure(figsize=(12, 6))
sns.set_style("whitegrid")
```

```
ax = sns.barplot(x='Contract', y='churn_rate', hue='PaymentMethod', data=df)


plt.title('Churn Rate by Contract Type and Payment Method', fontsize=16)
plt.xlabel('Contract Type', fontsize=12)
plt.ylabel('Churn Rate (%)', fontsize=12)
plt.legend(title='Payment Method', title_fontsize='12', fontsize='10')

plt.xticks(rotation=0)

plt.show()
```

```
+--------------+--------------------+--------------+----------------+---------
---------+
|      Contract|       PaymentMethod|total_customers|churned_customers|
churn_rate|
+--------------+--------------------+--------------+----------------+---------
---------+
|Month-to-month|Bank transfer (au…|            589|
201|34.125636672325975|
|Month-to-month|Credit card (auto…|            543|             178|
32.78084714548803|
|Month-to-month|    Electronic check|           1850|
994|53.729729729729726|
|Month-to-month|        Mailed check|            893|             282|
31.57894736842105|
|      One year|Bank transfer (au…|            391|              38|
9.718670076726342|
|      One year|Credit card (auto…|            398|
41|10.301507537688442|
|      One year|    Electronic check|            347|
64|18.443804034582133|
|      One year|        Mailed check|            337|              23|
6.824925816023739|
|      Two year|Bank transfer (au…|            564|              19|
3.368794326241135|
|      Two year|Credit card (auto…|            581|
13|2.2375215146299485|
|      Two year|    Electronic check|            168|              13|
7.738095238095238|
|      Two year|        Mailed check|            382|
3|0.7853403141361256|
+--------------+--------------------+--------------+----------------+---------
---------+
```

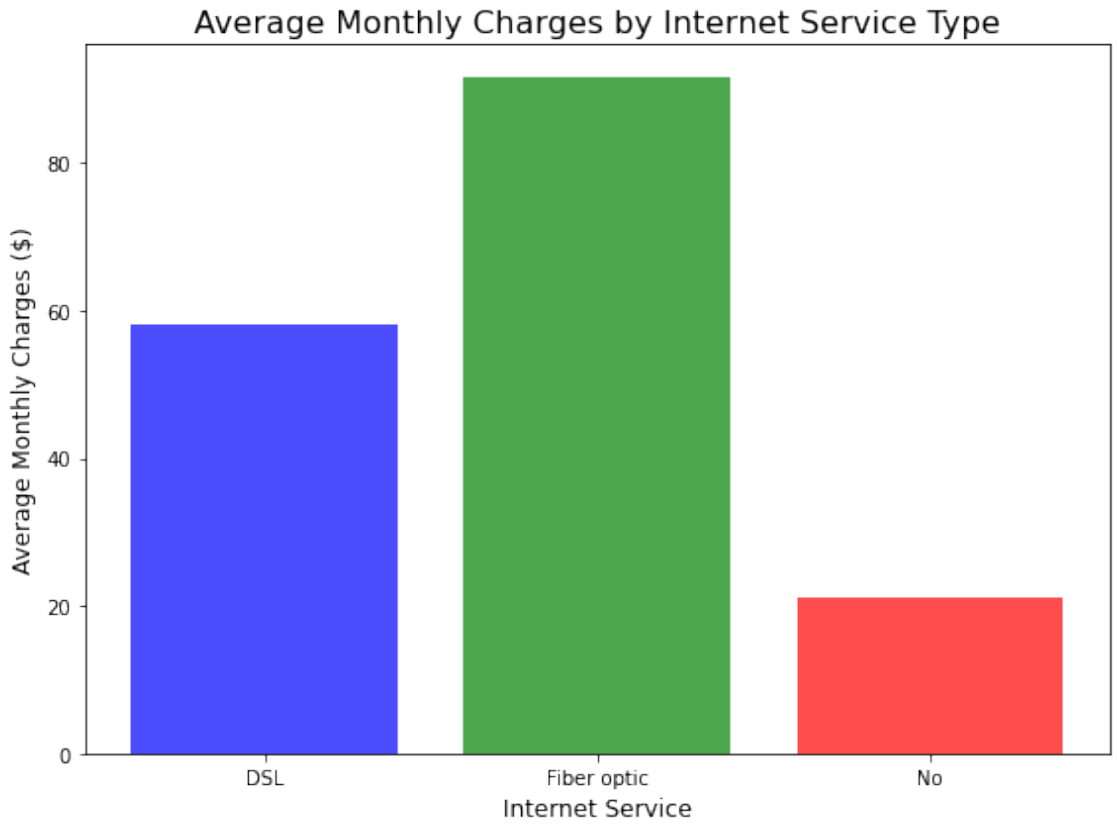


0.0.5 E. Explore the distribution of monthly charges for customers based on their type of internet service.

[16]:
```
res = spark.sql("""
    SELECT
        InternetService,
        COUNT(customerID) AS total_customers,
        ROUND(AVG(MonthlyCharges),2) AS avg_charges,
        MIN(MonthlyCharges) AS min_charges,
        MAX(MonthlyCharges) AS max_charges,
        ROUND(STDDEV(MonthlyCharges),2) AS stddev_charges
    FROM
        TelcoData
    GROUP BY
        InternetService
    ORDER BY
        InternetService

""")

res.show()

df = res.toPandas()
plt.figure(figsize=(8, 6))
```

```
bars = plt.bar(df['InternetService'], df['avg_charges'], alpha=0.7,␣
 ↪color=['blue', 'green', 'red'])

plt.title('Average Monthly Charges by Internet Service Type', fontsize=16)
plt.xlabel('Internet Service', fontsize=12)
plt.ylabel('Average Monthly Charges ($)', fontsize=12)


plt.tight_layout()
plt.show()
```

```
+---------------+---------------+-----------+-----------+-----------+-----------
---+
|InternetService|total_customers|avg_charges|min_charges|max_charges|stddev_char
ges|
+---------------+---------------+-----------+-----------+-----------+-----------
---+
|            DSL|           2421|       58.1|      23.45|       94.8|
16.26|
|    Fiber optic|           3096|       91.5|      67.75|     118.75|
12.66|
|             No|           1526|      21.08|      18.25|       26.9|
2.16|
+---------------+---------------+-----------+-----------+-----------+-----------
---+
```



0.0.6 F. Identify the top 10 customers who have contributed the most revenue to the company, based on total charges.

[17]:
```
spark.sql("""
SELECT
    customerID,
    gender,
    tenure,
    MonthlyCharges,
    TotalCharges,
    Contract
FROM
    TelcoData
ORDER BY
    CAST(TotalCharges AS DOUBLE) DESC
LIMIT 10
""").show()
```

```
+----------+------+------+--------------+------------+--------+
|customerID|gender|tenure|MonthlyCharges|TotalCharges|Contract|
```

```
+----------+------+------+-------------+------------+--------+
|2889-FPWRM|  Male|    72|        117.8|      8684.8|One year|
|7569-NMZYQ|Female|    72|       118.75|     8672.45|Two year|
|9739-JLPQJ|Female|    72|        117.5|      8670.1|Two year|
|9788-HNGUT|  Male|    72|       116.95|      8594.4|Two year|
|8879-XUAHX|  Male|    71|       116.25|     8564.75|Two year|
|9924-JPRMC|  Male|    72|        118.2|     8547.15|Two year|
|0675-NCDYU|Female|    72|        116.4|     8543.25|Two year|
|6650-BWFRT|Female|    72|       117.15|      8529.5|Two year|
|0164-APGRB|Female|    72|        114.9|      8496.7|Two year|
|1488-PBLJN|Female|    72|       116.85|      8477.7|Two year|
+----------+------+------+-------------+------------+--------+
```

0.0.7 G. Calculate the churn rate segmented by gender and whether the customer is a senior citizen

[18]:
```
spark.sql("""select max(CAST(TotalCharges AS DOUBLE)) as total from␣
 ↪TelcoData""").show()
```

```
+------+
| total|
+------+
|8684.8|
+------+
```

0.0.8 H. Write query to calculate Correlation between dependents and churn. Explore whether having dependents affects customer churn rates

[43]:
```
corrquery = """
SELECT
    CORR(
        CASE WHEN Dependents = 'Yes' THEN 1 ELSE 0 END,
        CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END
    ) AS correlation_dependents_churn
FROM
    TelcoData
"""

churn_rate = """
SELECT
    Dependents,
    COUNT(*) AS total_customers,
    SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS churned_customers,
```

```
    (SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) / COUNT(*)) * 100 AS␣
 ↪churn_rate
FROM
    TelcoData
GROUP BY
    Dependents
"""

correlation_result = spark.sql(corrquery)
churn_rates_result = spark.sql(churn_rate)

correlation_result.show()
churn_rates_result.show()
```

```
+---------------------------+
|correlation_dependents_churn|
+---------------------------+
|        -0.16422140157972537|
+---------------------------+

+----------+--------------+-----------------+------------------+
|Dependents|total_customers|churned_customers|        churn_rate|
+----------+--------------+-----------------+------------------+
|        No|          4933|             1543|31.279140482465028|
|       Yes|          2110|              326|15.450236966824646|
+----------+--------------+-----------------+------------------+
```

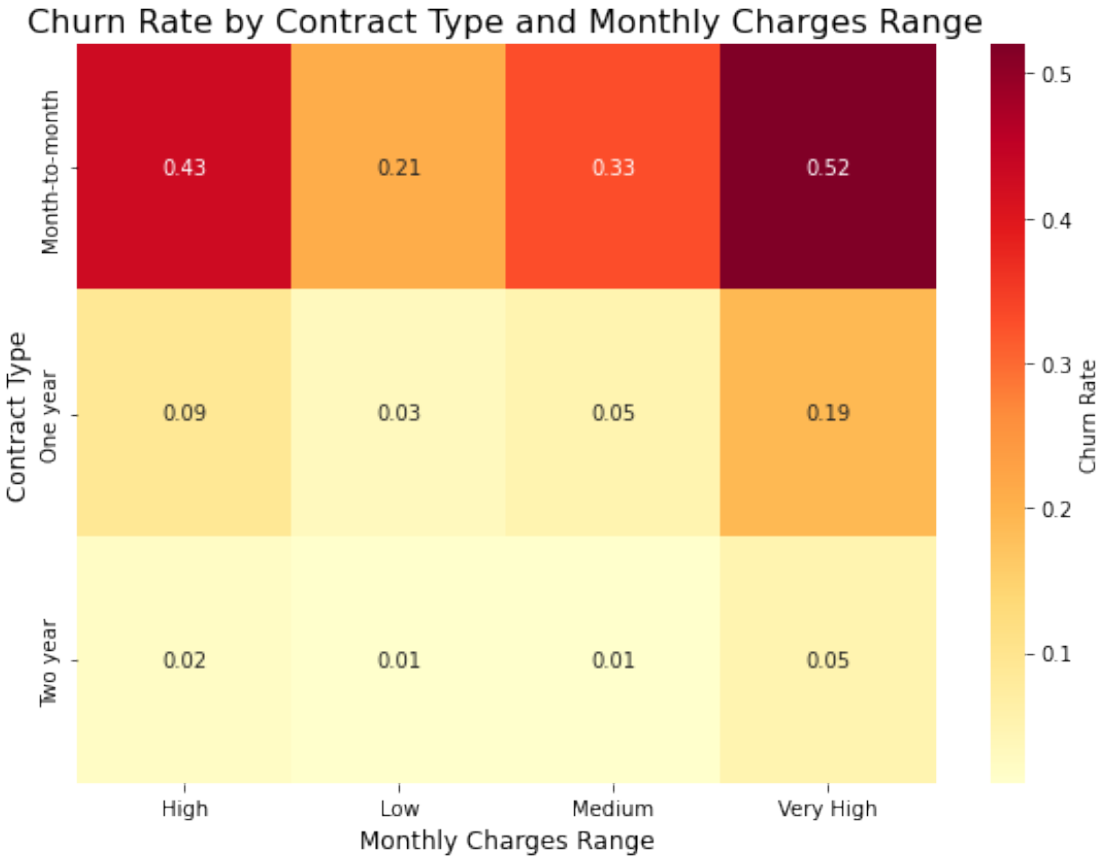
0.0.9 I. Predict potential churn rates by analyzing the relationship between monthly charges, contract types, and the churn rate.

[20]:
```
res = """
SELECT
    Contract,
    CASE
        WHEN MonthlyCharges < 20 THEN 'Low'
        WHEN MonthlyCharges >= 20 AND MonthlyCharges < 50 THEN 'Medium'
        WHEN MonthlyCharges >= 50 AND MonthlyCharges < 80 THEN 'High'
        ELSE 'Very High'
    END AS MonthlyCharges_Range,
    COUNT(customerID) AS total_customers,
    SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS churned_customers,
    ROUND(SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) / COUNT(customerID),␣
 ↪2) AS churn_rate
FROM
    TelcoData
```

```
GROUP BY
    Contract,
    MonthlyCharges_Range
ORDER BY
    churn_rate DESC
"""

res= spark.sql(res)
res.show()
df = res.toPandas()
pivot_df = df.pivot(index='Contract', columns='MonthlyCharges_Range',␣
↪values='churn_rate')


plt.figure(figsize=(8, 6))
sns.heatmap(pivot_df, annot=True, cmap='YlOrRd', fmt='.2f', cbar_kws={'label':␣
↪'Churn Rate'})

plt.title('Churn Rate by Contract Type and Monthly Charges Range', fontsize=16)
plt.xlabel('Monthly Charges Range', fontsize=12)
plt.ylabel('Contract Type', fontsize=12)

plt.tight_layout()
plt.show()
```

```
+--------------+--------------------+---------------+-----------------+---------
-+
|
Contract|MonthlyCharges_Range|total_customers|churned_customers|churn_rate|
+--------------+--------------------+---------------+-----------------+---------
-+
|Month-to-month|           Very High|           1474|              767|
0.52|
|Month-to-month|                High|           1291|              553|
0.43|
|Month-to-month|              Medium|            877|              286|
0.33|
|Month-to-month|                 Low|            233|               49|
0.21|
|      One year|           Very High|            567|              110|
0.19|
|      One year|                High|            434|               37|
0.09|
|      Two year|           Very High|            636|               33|
0.05|
|      One year|              Medium|            317|               15|
0.05|
```

```
|      One year|                Low|           155|                4|
0.03|
|      Two year|               High|           347|                8|
0.02|
|      Two year|             Medium|           487|                5|
0.01|
|      Two year|                Low|           225|                2|
0.01|
+--------------+-------------------+--------------+-----------------+---------
-+
```



0.0.10 J. Determine the churn rate for customers who have multiple services (Phone, Internet, and Streaming), which can help understand whether bundling services leads to higher or lower churn. Calculate churn rate for customers with multiple services.

[21]:
```
query = """
SELECT
    CASE
        WHEN PhoneService = 'Yes' AND InternetService != 'No' AND
        (StreamingTV = 'Yes' OR StreamingMovies = 'Yes') THEN 'Multiple␣
 ↪Services'
        ELSE 'Single or No Service'
    END AS service_bundle,
    COUNT(*) AS total_customers,
    SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS churned_customers,
    (SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS␣
 ↪churn_rate
FROM
    TelcoData
GROUP BY
    service_bundle
"""

result = spark.sql(query)
result.show()
```

```
+--------------------+---------------+-----------------+-----------------+
|      service_bundle|total_customers|churned_customers|       churn_rate|
+--------------------+---------------+-----------------+-----------------+
|Single or No Service|           3927|              901|22.94372294372294|
|   Multiple Services|           3116|              968|31.06546854942234|
+--------------------+---------------+-----------------+-----------------+
```

0.0.11 K. Churn Impact by device protection and online backup services. Write query to investigate whether having device protection or online backup services affects churn rates.

[22]:
```
query = """
SELECT
    CASE
        WHEN DeviceProtection = 'Yes' AND OnlineBackup = 'Yes' THEN 'Both'
        WHEN DeviceProtection = 'Yes' THEN 'Only Device Protection'
        WHEN OnlineBackup = 'Yes' THEN 'Only Online Backup'
        ELSE 'Neither'
    END AS protection,
```

```
    COUNT(*) AS total_customers,
    SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS churned_customers,
    (SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS␣
 ↪churn_rate
FROM
    TelcoData
GROUP BY
    protection
ORDER BY
    churn_rate DESC
"""

result = spark.sql(query)
result.show()
```

```
+--------------------+---------------+-----------------+-----------------+
|          protection|total_customers|churned_customers|       churn_rate|
+--------------------+---------------+-----------------+-----------------+
|             Neither|           3510|             1024|29.17378917378917|
|Only Device Prote…|           1104|              322|29.16666666666667|
|  Only Online Backup|           1111|              300|27.00270027002700|
|                Both|           1318|              223|16.91957511380880|
+--------------------+---------------+-----------------+-----------------+
```

0.0.12 L. Explore churn rates among customers who do not have phone service and investigate if it influences customer retention.

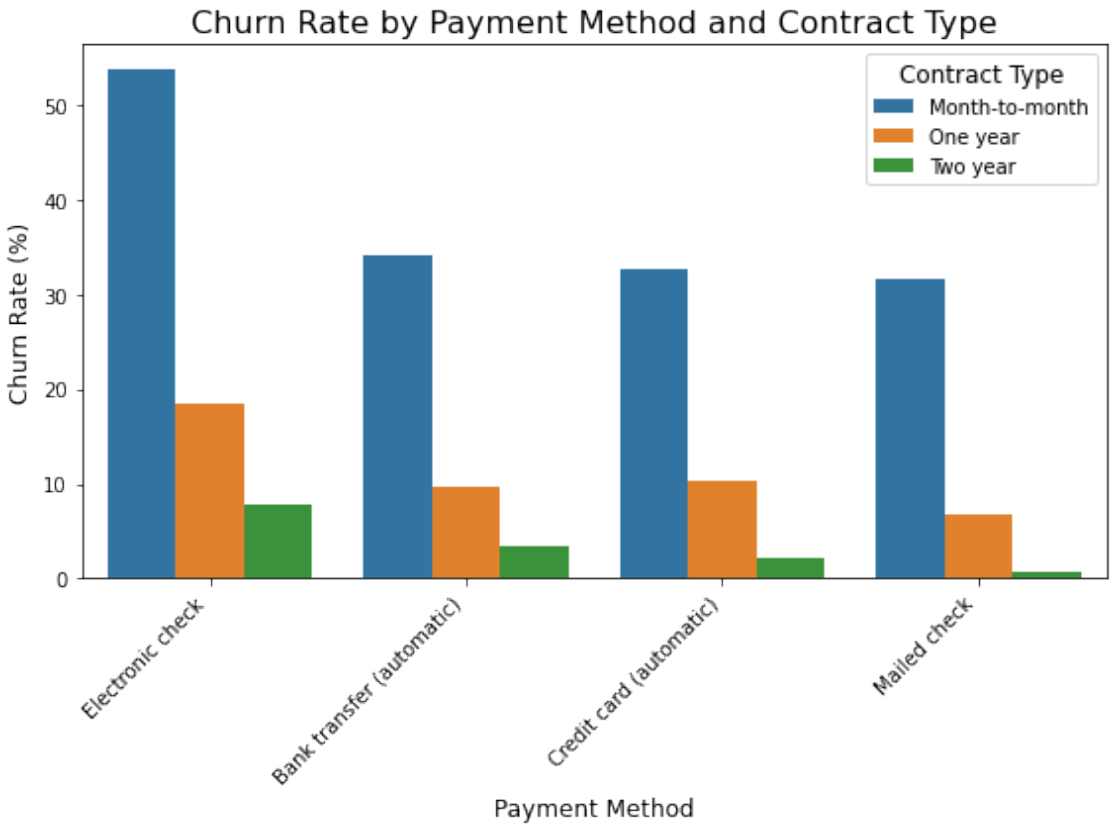
[23]:
```
query = """
SELECT
    PhoneService,
    COUNT(*) AS total_customers,
    SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS churned_customers,
    (SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS␣
 ↪churn_rate
FROM
    TelcoData
GROUP BY
    PhoneService
ORDER BY
    PhoneService DESC
"""

result = spark.sql(query)
result.show()
```

```
+------------+--------------+----------------+-----------------+
|PhoneService|total_customers|churned_customers|       churn_rate|
+------------+--------------+----------------+-----------------+
|         Yes|          6361|            1699|26.70963684955196|
|          No|           682|             170|24.92668621700880|
+------------+--------------+----------------+-----------------+
```

0.0.13 M. Understand the relationship between payment methods and contract types on customer churn. This query will help you discover which combinations are most prone to churn.

[24]:
```
import matplotlib.pyplot as plt
query = """
SELECT
    PaymentMethod,
    Contract,
    COUNT(*) AS total_customers,
    SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS churned_customers,
    (SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS␣
 ↪churn_rate
FROM
    TelcoData
GROUP BY
    PaymentMethod, Contract
ORDER BY
    churn_rate DESC
"""

result = spark.sql(query)
result.show()

result = result.toPandas()
plt.figure(figsize=(8, 6))
sns.barplot(x='PaymentMethod', y='churn_rate', hue='Contract', data=result)


plt.title('Churn Rate by Payment Method and Contract Type', fontsize=16)
plt.xlabel('Payment Method', fontsize=12)
plt.ylabel('Churn Rate (%)', fontsize=12)
plt.legend(title='Contract Type', title_fontsize='12', fontsize='10')

plt.xticks(rotation=45, ha='right')

plt.tight_layout()
plt.show()
```

```
+--------------------+-------------+---------------+----------------+---------
--------+
|       PaymentMethod|      Contract|total_customers|churned_customers|
churn_rate|
+--------------------+-------------+---------------+----------------+---------
--------+
|    Electronic check|Month-to-month|           1850|
994|53.72972972972973|
|Bank transfer (au…|Month-to-month|            589|
201|34.12563667232598|
|Credit card (auto…|Month-to-month|            543|
178|32.78084714548803|
|        Mailed check|Month-to-month|            893|
282|31.57894736842105|
|    Electronic check|      One year|            347|
64|18.44380403458213|
|Credit card (auto…|      One year|            398|
41|10.30150753768844|
|Bank transfer (au…|      One year|            391|               38|
9.71867007672634|
|    Electronic check|      Two year|            168|               13|
7.73809523809524|
|        Mailed check|      One year|            337|               23|
6.82492581602374|
|Bank transfer (au…|      Two year|            564|               19|
3.36879432624113|
|Credit card (auto…|      Two year|            581|               13|
2.23752151462995|
|        Mailed check|      Two year|            382|                3|
0.78534031413613|
+--------------------+-------------+---------------+----------------+---------
--------+
```

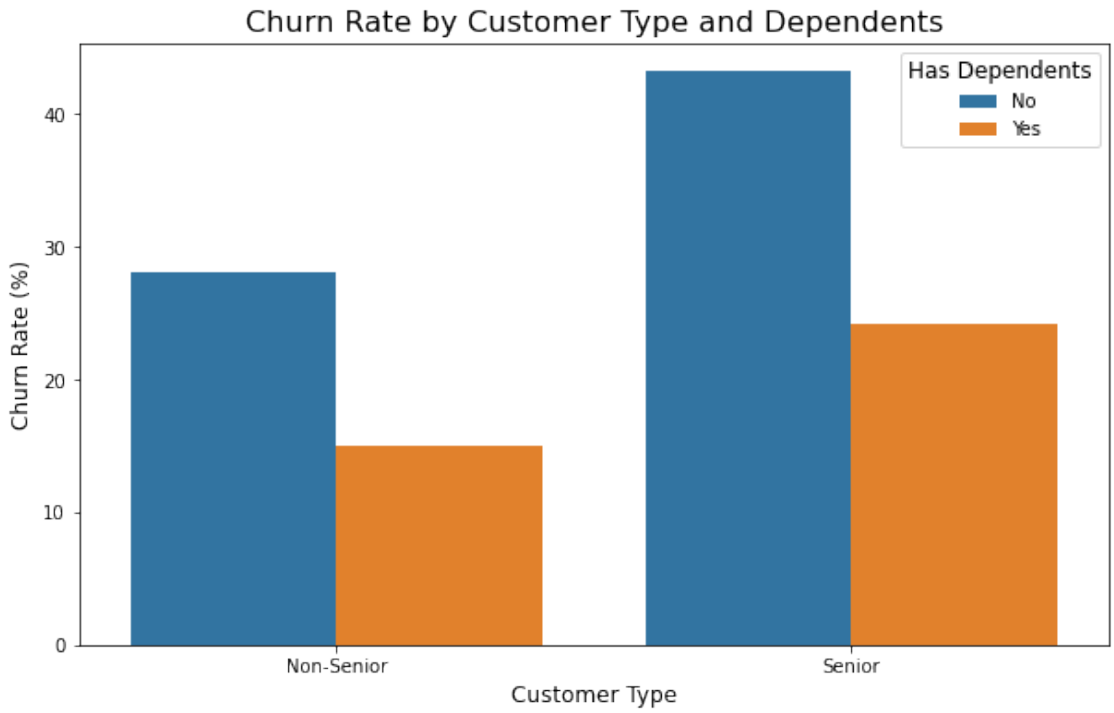


0.0.14 N. Analyze how customer churn is affected by senior citizen status and whether the customer has dependents.

[25]:
```
query = """
SELECT
    CASE WHEN SeniorCitizen = 1 THEN 'Senior' ELSE 'Non-Senior' END AS␣
 ↪customer_type,
    Dependents,
    COUNT(*) AS total_customers,
    SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS churned_customers,
    (SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS␣
 ↪churn_rate
FROM
    TelcoData
GROUP BY
    SeniorCitizen, Dependents
ORDER BY
    customer_type, Dependents
"""
```

```
result = spark.sql(query)
result.show()

result = result.toPandas()
plt.figure(figsize=(10, 6))
sns.barplot(x='customer_type', y='churn_rate', hue='Dependents', data=result)

plt.title('Churn Rate by Customer Type and Dependents', fontsize=16)
plt.xlabel('Customer Type', fontsize=12)
plt.ylabel('Churn Rate (%)', fontsize=12)
plt.legend(title='Has Dependents', title_fontsize='12', fontsize='10')

plt.show()
```

```
+-------------+----------+---------------+-----------------+------------------+
|customer_type|Dependents|total_customers|churned_customers|       churn_rate|
+-------------+----------+---------------+-----------------+------------------+
|   Non-Senior|        No|           3882|             1089|28.05255023183926|
|   Non-Senior|       Yes|           2019|              304|15.05695889053987|
|       Senior|        No|           1051|              454|43.19695528068506|
|       Senior|       Yes|             91|               22|24.17582417582418|
+-------------+----------+---------------+-----------------+------------------+
```

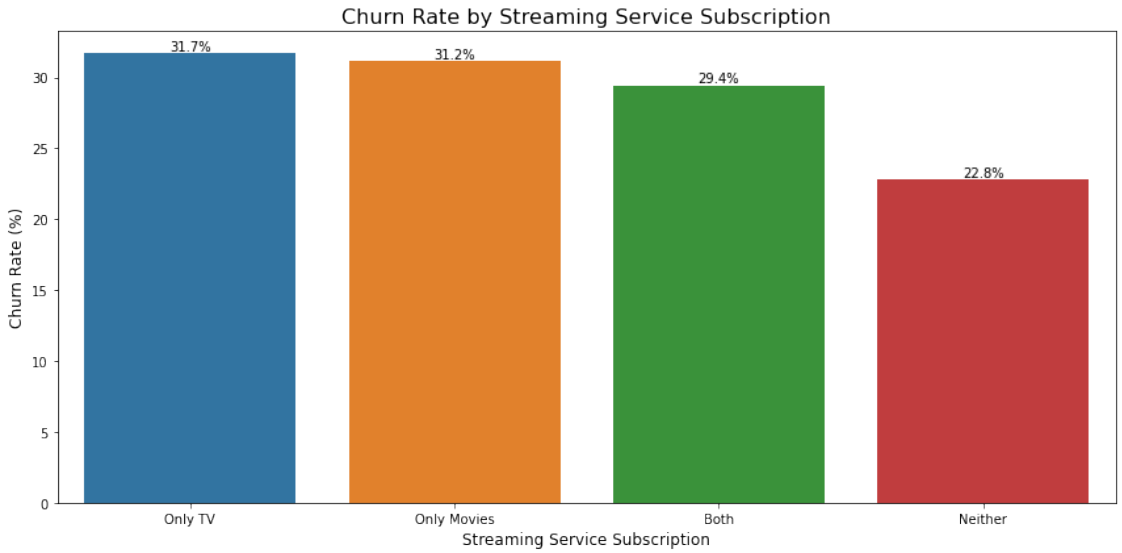


0.0.15 O. Explore whether subscribing to streaming services like Streaming TV and Streaming Movies influences the churn rate.

[26]:
```
query = """
SELECT
    CASE
        WHEN StreamingTV = 'Yes' AND StreamingMovies = 'Yes' THEN 'Both'
        WHEN StreamingTV = 'Yes' THEN 'Only TV'
        WHEN StreamingMovies = 'Yes' THEN 'Only Movies'
        ELSE 'Neither'
    END AS streaming,
    COUNT(*) AS total_customers,
    SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS churned_customers,
    (SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS␣
 ↪churn_rate
FROM
    TelcoData
GROUP BY
    streaming
ORDER BY
    churn_rate DESC
"""

result = spark.sql(query)
result.show()

df = result.toPandas()

plt.figure(figsize=(12, 6))
sns.barplot(x='streaming', y='churn_rate', data=df, order=df['streaming'])


plt.title('Churn Rate by Streaming Service Subscription', fontsize=16)
plt.xlabel('Streaming Service Subscription', fontsize=12)
plt.ylabel('Churn Rate (%)', fontsize=12)

# To Add value labels on top of each bar
for i, v in enumerate(df['churn_rate']):
    plt.text(i, v, f'{v:.1f}%', ha='center', va='bottom')


plt.tight_layout()
plt.show()
```

```
+----------+--------------+----------------+-----------------+
```

```
|  streaming|total_customers|churned_customers|       churn_rate|
+-----------+---------------+-----------------+-----------------+
|    Only TV|            767|              243|31.68187744458931|
|Only Movies|            792|              247|31.18686868686869|
|       Both|           1940|              571|29.43298969072165|
|    Neither|           3544|              808|22.79909706546275|
+-----------+---------------+-----------------+-----------------+
```



0.0.16 P. Understand how tenure and MonthlyCharges differ between churned and non-churned customers. This can provide insights into the behavior of long-term customers.

[48]:
```
query = """
SELECT
    Churn,
    AVG(tenure) AS avg_tenure,
    AVG(MonthlyCharges) AS avg_monthly_charges
FROM
    TelcoData
GROUP BY
    Churn
"""

result = spark.sql(query)
result.show()
```

```
+-----+-----------------+------------------+
```

```
|Churn|        avg_tenure|avg_monthly_charges|
+-----+------------------+-------------------+
|   No| 37.56996521066873|   61.2651236953999|
|  Yes|17.979133226324237|   74.4413322632423|
+-----+------------------+-------------------+
```

0.0.17 Q. Compare monthly charges and churn rates between newer customers and long-time customers.

[28]:
```
query = """
SELECT
    CASE
        WHEN tenure <= 12 THEN 'New Customer'
        ELSE 'Long-time Customer'
    END AS category,
    AVG(MonthlyCharges) AS avg_monthly_charges,
    (SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS␣
 ↪churn_rate
FROM
    TelcoData
GROUP BY
    category
"""

result = spark.sql(query)
result.show()
# df = result.toPandas()
```

```
+------------------+-------------------+------------------+
|          category|avg_monthly_charges|       churn_rate|
+------------------+-------------------+------------------+
|Long-time Customer|  68.66107679637616|17.12991558575252|
|      New Customer|  56.09778133577306|47.43824336688015|
+------------------+-------------------+------------------+
```

[49]:
```
spark.sql('''select monthlycharges, tenure from TelcoData where tenure = 0''').
 ↪show()
```

```
+--------------+------+
|monthlycharges|tenure|
+--------------+------+
|         52.55|     0|
|         20.25|     0|
|         80.85|     0|
|         25.75|     0|
```

```
|          56.05|     0|
|          19.85|     0|
|          25.35|     0|
|           20.0|     0|
|           19.7|     0|
|          73.35|     0|
|           61.9|     0|
+--------------+------+
```

0.0.18 R. What is the correlation between senior citizen status and churn rate?

[29]:
```
query = """
SELECT
    CASE WHEN SeniorCitizen = 1 THEN 'Senior' ELSE 'Non-Senior' END AS␣
 ↪customer_type,
    COUNT(*) AS total_customers,
    SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS churned_customers,
    (SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS␣
 ↪churn_rate
FROM
    TelcoData
GROUP BY
    SeniorCitizen
"""

result = spark.sql(query)
result.show()
```

```
+-------------+---------------+-----------------+------------------+
|customer_type|total_customers|churned_customers|       churn_rate|
+-------------+---------------+-----------------+------------------+
|       Senior|           1142|              476|41.68126094570928|
|   Non-Senior|           5901|             1393|23.60616844602610|
+-------------+---------------+-----------------+------------------+
```

0.0.19 S. Partition customers based on whether they are senior citizens and divide them into 5 groups based on tenure. [Use NTILE.]

[53]:
```
partition_query = """
    SELECT
        CustomerID,
        CASE WHEN SeniorCitizen = 1 THEN 'Yes' ELSE 'No' END AS SeniorCitizen,
        tenure,
```

```
        NTILE(5) OVER (PARTITION BY CASE WHEN SeniorCitizen = 1 THEN 'Yes' ELSE␣
 ↪'No' END ORDER BY tenure) AS tenure_ntile
        FROM TelcoData
    ORDER BY SeniorCitizen , tenure_ntile
"""

result = spark.sql(partition_query)
result.show()
```

```
+----------+-------------+------+------------+
|CustomerID|SeniorCitizen|tenure|tenure_ntile|
+----------+-------------+------+------------+
|3930-ZGWVE|           No|     1|           1|
|7590-VHVEG|           No|     1|           1|
|3679-XASPY|           No|     1|           1|
|3213-VVOLG|           No|     0|           1|
|2775-SEFEE|           No|     0|           1|
|7310-EGVHZ|           No|     1|           1|
|6380-ARCEH|           No|     1|           1|
|5709-LVOEQ|           No|     0|           1|
|7644-OMVMY|           No|     0|           1|
|2923-ARZLG|           No|     0|           1|
|4075-WKNIU|           No|     0|           1|
|1066-JKSGK|           No|     1|           1|
|8665-UTDHZ|           No|     1|           1|
|2273-QCKXA|           No|     1|           1|
|5919-TMRGD|           No|     1|           1|
|4472-LVYGI|           No|     0|           1|
|3115-CZMZD|           No|     0|           1|
|4367-NUYAO|           No|     0|           1|
|1371-DWPAZ|           No|     0|           1|
|2520-SGTTA|           No|     0|           1|
+----------+-------------+------+------------+
only showing top 20 rows
```

0.0.20 T. Use PERCENT_RANK to identify the top 5% of customers by MonthlyCharges.

[58]:
```
top_5 = """
WITH ranked_customers AS (
    SELECT
        *,
        PERCENT_RANK() OVER (ORDER BY MonthlyCharges desc) AS percentile
    FROM
        TelcoData
```

```
)
SELECT
    customerID,
    MonthlyCharges,
    percentile
FROM
    ranked_customers
WHERE
    percentile <= 0.05
ORDER BY
    MonthlyCharges DESC
"""

result = spark.sql(top_5)
result.show(truncate=False)
```

```
+----------+--------------+----------------------+
|customerID|MonthlyCharges|percentile            |
+----------+--------------+----------------------+
|7569-NMZYQ|118.75        |0.0                   |
|8984-HPEMB|118.65        |1.4200511218403862E-4|
|5989-AXPUC|118.6         |2.8401022436807724E-4|
|5734-EJKXG|118.6         |2.8401022436807724E-4|
|8199-ZLLSA|118.35        |5.680204487361545E-4 |
|9924-JPRMC|118.2         |7.100255609201931E-4 |
|2889-FPWRM|117.8         |8.520306731042318E-4 |
|3810-DVDQQ|117.6         |9.940357852882703E-4 |
|9739-JLPQJ|117.5         |0.001136040897472309 |
|2302-ANTDP|117.45        |0.0012780460096563477|
|6904-JLBGY|117.35        |0.0014200511218403862|
|4282-MSACW|117.2         |0.0015620562340244249|
|6650-BWFRT|117.15        |0.0017040613462084636|
|9788-HNGUT|116.95        |0.001846066458392502 |
|1488-PBLJN|116.85        |0.0019880715705765406|
|0017-IUDMW|116.8         |0.0021300766827605793|
|8628-MFKAX|116.75        |0.002272081794944618 |
|3680-CTHUH|116.6         |0.0024140869071286567|
|3258-ZKPAI|116.6         |0.0024140869071286567|
|3795-CAWEX|116.55        |0.002698097131496734 |
+----------+--------------+----------------------+
only showing top 20 rows
```

0.0.21 U. Find customers who fall within the top 5% of the distribution based on monthly charges. Compare total charges with the next customer in the same internet service type, based on monthly charges.

[32]:
```
query = """
WITH ranked_customers AS (
    SELECT
        *,
        PERCENT_RANK() OVER (ORDER BY MonthlyCharges DESC) AS percentile,
        LEAD(TotalCharges) OVER (PARTITION BY InternetService ORDER BY␣
 ↪MonthlyCharges DESC)
        AS next_total_charges
    FROM
        TelcoData
)
SELECT
    customerID,
    InternetService,
    MonthlyCharges,
    TotalCharges,
    next_total_charges,
    (TotalCharges - next_total_charges) AS total_charges_difference
FROM
    ranked_customers
WHERE
    percentile <= 0.05
ORDER BY
    InternetService, MonthlyCharges DESC
"""

result = spark.sql(query)
result.show()
```

```
+----------+--------------+--------------+------------+------------------+-----
-------------------+
|customerID|InternetService|MonthlyCharges|TotalCharges|next_total_charges|total
_charges_difference|
+----------+--------------+--------------+------------+------------------+-----
-------------------+
|7569-NMZYQ|    Fiber optic|        118.75|     8672.45|            8477.6|
194.85000000000036|
|8984-HPEMB|    Fiber optic|        118.65|      8477.6|           7990.05|
487.5500000000002|
|5989-AXPUC|    Fiber optic|         118.6|     7990.05|            7365.7|
624.3500000000004|
|5734-EJKXG|    Fiber optic|         118.6|      7365.7|           7804.15|
-438.4499999999998|
```

```
|8199-ZLLSA|    Fiber optic|        118.35|     7804.15|           8547.15|
-743.0|
|9924-JPRMC|    Fiber optic|         118.2|     8547.15|            8684.8|
-137.64999999999964|
|2889-FPWRM|    Fiber optic|         117.8|      8684.8|            8308.9|
375.89999999999964|
|3810-DVDQQ|    Fiber optic|         117.6|      8308.9|            8670.1|
-361.2000000000007|
|9739-JLPQJ|    Fiber optic|         117.5|      8670.1|            5438.9|
3231.2000000000007|
|2302-ANTDP|    Fiber optic|        117.45|      5438.9|           8436.25|
-2997.3500000000004|
|6904-JLBGY|    Fiber optic|        117.35|     8436.25|           8035.95|
400.3000000000002|
|4282-MSACW|    Fiber optic|         117.2|     8035.95|            8529.5|
-493.5500000000002|
|6650-BWFRT|    Fiber optic|        117.15|      8529.5|            8594.4|
-64.89999999999964|
|9788-HNGUT|    Fiber optic|        116.95|      8594.4|            8477.7|
116.69999999999891|
|1488-PBLJN|    Fiber optic|        116.85|      8477.7|           8456.75|
20.950000000000728|
|0017-IUDMW|    Fiber optic|         116.8|     8456.75|           8277.05|
179.70000000000073|
|8628-MFKAX|    Fiber optic|        116.75|     8277.05|            7049.5|
1227.5499999999993|
|3680-CTHUH|    Fiber optic|         116.6|      7049.5|           8337.45|
-1287.9500000000007|
|3258-ZKPAI|    Fiber optic|         116.6|     8337.45|            8152.3|
185.15000000000055|
|3795-CAWEX|    Fiber optic|        116.55|      8152.3|           6382.55|
1769.75|
+----------+---------------+--------------+------------+------------------+-----
-------------------+
only showing top 20 rows
```

0.0.22 V. Find the top 5 customers with the highest MonthlyCharges within each Contract type.

[60]:
```
query = """
WITH ranked_customers AS (
    SELECT
        *,
        ROW_NUMBER() OVER (PARTITION BY Contract ORDER BY MonthlyCharges DESC)␣
 ↪AS rank
```

```
    FROM
        TelcoData
)
SELECT
    customerID,
    Contract,
    MonthlyCharges,
    rank
FROM
    ranked_customers
WHERE
    rank <= 5
ORDER BY
    Contract, rank
"""

result = spark.sql(query)
result.show()
```

```
+----------+--------------+--------------+----+
|customerID|      Contract|MonthlyCharges|rank|
+----------+--------------+--------------+----+
|2302-ANTDP|Month-to-month|        117.45|   1|
|8016-NCFVO|Month-to-month|         116.5|   2|
|9659-QEQSY|Month-to-month|        115.65|   3|
|4361-BKAXE|Month-to-month|         114.5|   4|
|6710-HSJRD|Month-to-month|         114.1|   5|
|5734-EJKXG|      One year|         118.6|   1|
|8199-ZLLSA|      One year|        118.35|   2|
|2889-FPWRM|      One year|         117.8|   3|
|4282-MSACW|      One year|         117.2|   4|
|3680-CTHUH|      One year|         116.6|   5|
|7569-NMZYQ|      Two year|        118.75|   1|
|8984-HPEMB|      Two year|        118.65|   2|
|5989-AXPUC|      Two year|         118.6|   3|
|9924-JPRMC|      Two year|         118.2|   4|
|3810-DVDQQ|      Two year|         117.6|   5|
+----------+--------------+--------------+----+
```

0.0.23 W. Calculate the churn rate in each Contract type and rank the contracts by churn rate.

[62]:
```
query = """
WITH contract_churn AS (
    SELECT
```

```
        Contract,
        COUNT(*) AS total_customers,
        SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS churned_customers,
        (SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS␣
 ↪churn_rate
    FROM
        TelcoData
    GROUP BY
        Contract
)
SELECT
    Contract,
    total_customers,
    churned_customers,
    churn_rate,
    RANK() OVER (ORDER BY churn_rate DESC) AS churn_rank
FROM
    contract_churn
ORDER BY
    churn_rank
"""

result = spark.sql(query)
result.show()
```

```
+--------------+---------------+-----------------+-----------------+----------+
|      Contract|total_customers|churned_customers|       churn_rate|churn_rank|
+--------------+---------------+-----------------+-----------------+----------+
|Month-to-month|           3875|             1655|42.70967741935484|         1|
|      One year|           1473|              166|11.26951799049559|         2|
|      Two year|           1695|               48| 2.83185840707965|         3|
+--------------+---------------+-----------------+-----------------+----------+
```

0.0.24 X. Perform an in-depth analysis of customers using window functions to understand customer rankings, distribution, and trends in charges and tenure.

[63]:
```
result = spark.sql('''
SELECT
    customerID,
    tenure,
    RANK() OVER (ORDER BY MonthlyCharges DESC) AS ChargeRank,
    RANK() OVER (ORDER BY tenure DESC) AS TenureRank,
    NTILE(4) OVER (ORDER BY MonthlyCharges) AS Quartile,
    SUM(MonthlyCharges) OVER (ORDER BY customerID ROWS UNBOUNDED PRECEDING) AS␣
 ↪RunningTotal,
```

```
    CASE
        WHEN tenure <= 12 THEN 'New Customers'
        WHEN tenure BETWEEN 13 AND 24 THEN '1-2 Years'
        ELSE 'Long Term Customers'
    END AS TenureGroup
from TelcoData
order by RunningTotal


''')
result.show()
```

```
+----------+------+----------+----------+--------+------------------+-----------
--------+
|customerID|tenure|ChargeRank|TenureRank|Quartile|      RunningTotal|
TenureGroup|
+----------+------+----------+----------+--------+------------------+-----------
--------+
|0002-ORFBO|     9|      3860|      5190|       2|              65.6|      New
Customers|
|0003-MKNFE|     9|      4140|      5190|       2|             125.5|      New
Customers|
|0004-TLHLJ|     4|      3295|      5806|       3|             199.4|      New
Customers|
|0011-IGKFF|    13|      1087|      4749|       4|             297.4|
1-2 Years|
|0013-EXCHZ|     3|      2348|      5982|       3|381.29999999999995|      New
Customers|
|0013-MHZWF|     9|      3686|      5190|       2|450.69999999999993|      New
Customers|
|0013-SMEOE|    71|       236|       363|       4|             560.4|Long Term
Customers|
|0014-BMAQU|    63|      2251|      1190|       3|            645.05|Long Term
Customers|
|0015-UOCOJ|     7|      4887|      5432|       2|            693.25|      New
Customers|
|0016-QLJIS|    65|      1675|      1034|       4|             783.7|Long Term
Customers|
|0017-DINOC|    54|      4993|      1820|       2| 828.9000000000001|Long Term
Customers|
|0017-IUDMW|    72|        16|         1|       4|             945.7|Long Term
Customers|
|0018-NYROU|     5|      3744|      5673|       2|1014.6500000000001|      New
Customers|
|0019-EFAEP|    72|       761|         1|       4|           1115.95|Long Term
Customers|
|0019-GFNTW|    56|      5007|      1676|       2|            1161.0|Long Term
```

```
Customers|
|0020-INWCK|    71|      1204|       363|       4|           1256.75|Long Term
Customers|
|0020-JDNXP|    34|      4052|      3140|       2|            1318.0|Long Term
Customers|
|0021-IKXGC|     1|      3383|      6420|       3|            1390.1|      New
Customers|
|0022-TCJCI|    45|      4005|      2446|       2|            1452.8|Long Term
Customers|
|0023-HGHWL|     1|      5605|      6420|       1|1477.8999999999999|      New
Customers|
+----------+------+----------+----------+--------+------------------+-----------
--------+
only showing top 20 rows
```

[ ]: