D207 – Exploratory Data Analysis

Morrell J. Parrish

Western Governors University

Table of Contents

A. RESEARCH QUESTION 3

A2. Benefits of Analysis 3

A3. Identifying Data 3

B. DATA ANALYSIS DESCRIPTION 3

B2. Output and Results of Analysis 8

B3. Justification of Analytical Technique 12

C. UNIVARIATE STATISTICS 12

C2. Visual of Findings 12

D. BIVARIATE STATISTICS 14

D2. Visual of Findings 14

E. SUMMARY 16

F. PANOPTO VIDEO RECORDING 16

REFERENCES 17

D207 – Exploratory Data Analysis

A. Research Question

During this course of research, we will explore and identify which customers are at a greater risk for churn; does the internet service type, area and customer service skills affect whether customers churn or not?

A2. Benefits of Analysis

The benefits of this analysis will provide businesses with a better understanding of the services they provide for specific areas, as well as their customer service skills and customer satisfaction ratings; for example, if the analysis reveals a dependency or correlation between areas of service, customer service skills, satisfaction ratings, and customers who churn, businesses can delve into improving their customer service skills, which will raise their customer satisfaction ratings and will over all help retain customers.

A3. Identifying Data

The data we will be using will consists of the following items: caseorder, internet service type, churn, and the survey response categories 1 through 8 (timely responses, timely fixes, timely replacement, reliability, options, respectful responses, courteous exchange, and evidence of active listening).

B. Data Analysis Description

```
# Import the following Libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from scipy.stats import chi2_contingency
from scipy.stats import chi2
```

```
# Read Churn data

df = pd.read_csv("churn_clean.csv",
                 usecols=["CaseOrder", "InternetService", "Churn", "Area", "Multiple",
                "Item1", "Item2", "Item3", "Item4", "Item5",
                "Item6", "Item7", "Item8"], index_col="CaseOrder")

# Selecting a random sample from our Churn data

data = df.sample(n=100)
print(data)

# Print description of sample data set

print(data.describe())

# Sample dataset column info

data.info()

# Creating a Contingency table for Churn and InternetService variables

contingency_table = pd.crosstab(data['Churn'], data['InternetService'], margins=True,
margins_name="Total")

print(contingency_table)

# Perform Chi-Square Test on Churn and InternetService variables

stat, p, dof, expected = chi2_contingency(contingency_table)
print("stat = ", stat)
print("p-value = ", p)
print("Degrees of Freedom =", dof)
print("Expected = ", expected)

# Calculate alpha and critical values and interpret test-statistic

prob = 0.95
critical = chi2.ppf(prob, dof)
print('significance=%.3f, p=%.3f' % (1 - prob, p))
print("critical value = ", critical)

# Test result summary

print('probability=%.3f, critical=%.3f, stat=%.3f' % (prob, critical, stat))
if abs(stat) >= critical:
```

```
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')

# Interpret p-value

alpha = 1.0 - prob
print('significance=%.3f, p=%.3f' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')

# Creating a Contingency table for Area and Churn variables

contingency_table_2 = pd.crosstab(data['Churn'], data['Area'], margins=False)

print(contingency_table_2)

# Perform Chi-Square Test on Churn and Area variables

stat, p, dof, expected = chi2_contingency(contingency_table_2)
print("stat = ", stat)
print("p-value = ", p)
print("Degrees of Freedom =", dof)
print("Expected = ", expected)

# Calculate alpha and critical values and interpret test-statistic

prob = 0.95
critical = chi2.ppf(prob, dof)
print('significance=%.3f, p=%.3f' % (1 - prob, p))
print("critical value = ", critical)

# Test result summary

print('probability=%.3f, critical=%.3f, stat=%.3f' % (prob, critical, stat))
if abs(stat) >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')

# Interpret p-value

alpha = 1.0 - prob
print('significance=%.3f, p=%.3f' % (alpha, p))
```

```
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')

# Creating a Contingency table for Churn and Multiple variables

contingency_table_3 = pd.crosstab(data['Churn'], data['Multiple'], margins=False)

print(contingency_table_3)

# Creating dataframe for our Univariate Statistics

df2 = pd.read_csv("churn_clean.csv", usecols=["MonthlyCharge", "Bandwidth_GB_Year"])

# Create histograms for MonthlyCharge and Bandwidth_GB_Year

print(df2)

histogram = df2.hist()

plt.show()

# Creating histograms for Item1 and Item2

df3 = pd.read_csv("churn_clean.csv", usecols=["Item1", "Item2"])

print(df3)

# Rename Item1 and Item2

dictionary = {'Item1': 'Timely_Responses', 'Item2': 'Timely_Fixes'}

df3.rename(columns=dictionary, inplace=True)

hist = df3.hist()

plt.show()

# Creating Boxplots for Variables

df4 = pd.read_csv("churn_clean.csv", usecols=["MonthlyCharge", "Bandwidth_GB_Year",
                                              "Item1", "Item2"])

sns.boxplot(y=df4["MonthlyCharge"])
```

```
plt.show()

sns.boxplot(y=df4["Bandwidth_GB_Year"])

plt.show()

sns.boxplot(y=df4["Item1"])

plt.ylabel("Timely_Responses")

plt.show()

sns.boxplot(y=df4["Item2"])

plt.ylabel("Timely_Fixes")

plt.show()

# Randomly selecting data from our Churn dataset

df5 = pd.read_csv("churn_clean.csv", usecols=["MonthlyCharge", "Bandwidth_GB_Year",
                  "Area", "Item1", "Item2", "Item3", "Item4", "Item5", "Item6", "Item7", "Item8"])

data3 = df5.sample(n=100)

print(data3)

# Create a scatter plot of continuous variables MonthlyCharge & Bandwidth_GB_Year

x = data3['MonthlyCharge']
y = data3['Bandwidth_GB_Year']

plt.scatter(x, y, c='red')
plt.scatter(x, y, c='blue')

plt.xlabel("MonthlyCharge")
plt.ylabel("Bandwidth_GBYear")

plt.plot(np.unique(x), np.poly1d(np.polyfit(x, y, 1))(np.unique(x)), color='red')

plt.show()

# Correlation Matrix

sns.heatmap(df5.corr(), linewidths=.3, annot=True)
plt.show()
```

B2. Output and Results of Analysis

```
main.py
~/PycharmProjects/pythonProject2/main.py
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from scipy.stats import chi2_contingency
from scipy.stats import chi2

# Read Churn data

df = pd.read_csv("churn_clean.csv",
                 usecols=["CaseOrder", "InternetService", "Churn", "Area", "Item1", "Item2", "Item3", "Item4", "Item5",
                          "Item6", "Item7", "Item8"], index_col="CaseOrder")

# Selecting a random sample from our Churn data

data = df.sample(n=100)
print(data)
```

Figure 1: Input

```
               Area Churn InternetService  Item1  ...  Item5  Item6  Item7  Item8
CaseOrder                                         ...
4292          Rural   Yes             DSL      4  ...      2      5      4      3
5621          Urban    No     Fiber Optic      2  ...      3      2      2      2
999        Suburban    No     Fiber Optic      4  ...      3      4      2      3
2800          Rural   Yes     Fiber Optic      4  ...      5      2      2      4
4477          Rural    No     Fiber Optic      6  ...      3      6      6      5
...             ...   ...             ...    ...  ...    ...    ...    ...    ...
7836       Suburban    No     Fiber Optic      6  ...      4      4      5      4
4866       Suburban   Yes             DSL      3  ...      3      2      3      3
4588       Suburban    No            None      6  ...      3      5      4      4
9061          Rural    No     Fiber Optic      4  ...      4      2      3      4
242        Suburban    No     Fiber Optic      4  ...      4      3      3      3

[100 rows x 11 columns]
```

Figure 1: Output

```
# Print description of sample data set

print(data.describe())
```

Figure 2: Input

```
           Item1       Item2       Item3  ...       Item6       Item7       Item8
count  100.000000  100.000000  100.000000  ...  100.000000  100.000000  100.000000
mean     3.630000    3.500000    3.580000  ...    3.440000    3.420000    3.480000
std      0.981187    0.969223    1.036505  ...    1.085441    1.084137    1.077596
min      1.000000    1.000000    1.000000  ...    1.000000    1.000000    1.000000
25%      3.000000    3.000000    3.000000  ...    3.000000    3.000000    3.000000
50%      4.000000    3.000000    4.000000  ...    4.000000    3.000000    3.000000
75%      4.000000    4.000000    4.000000  ...    4.000000    4.000000    4.000000
max      6.000000    6.000000    6.000000  ...    6.000000    6.000000    6.000000

[8 rows x 8 columns]
```

Figure 2: Output

```
26  # Sample dataset column info
27
28  data.info()
```

Figure 3: Input

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100 entries, 4292 to 242
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Area             100 non-null    object
 1   Churn            100 non-null    object
 2   InternetService  100 non-null    object
 3   Item1            100 non-null    int64
 4   Item2            100 non-null    int64
 5   Item3            100 non-null    int64
 6   Item4            100 non-null    int64
 7   Item5            100 non-null    int64
 8   Item6            100 non-null    int64
 9   Item7            100 non-null    int64
 10  Item8            100 non-null    int64
dtypes: int64(8), object(3)
memory usage: 9.4+ KB
```

Figure 3: Output

```
30  # Creating a Contingency table for Churn and InternetService variables
31
32  contingency_table = pd.crosstab(data['Churn'], data['InternetService'], margins=False)
33
34  print(contingency_table)
```

Figure 4: Input

InternetService	DSL	Fiber Optic	None
Churn			
No	28	35	7
Yes	14	10	6

Figure 4: Output

main.py

```
35
36  # Perform Chi-Square Test on Churn and InternetService variables
37
38  stat, p, dof, expected = chi2_contingency(contingency_table)
39  print("stat = ", stat)
40  print("p-value = ", p)
41  print("Degrees of Freedom =", dof)
42  print("Expected = ", expected)
43
44  # Calculate alpha and critical values and interpret test-statistic
45
46  prob = 0.95
47  critical = chi2.ppf(prob, dof)
48  print('significance=%.3f, p=%.3f' % (1 - prob, p))
49  print("critical value = ", critical)
```

Figure 5: Input

```
stat =  8.282295482295483
p-value =  0.21814254024204363
Degrees of Freedom = 6
Expected =  [[ 30.    29.25  15.75  75.  ]
 [ 10.     9.75   5.25  25.  ]
 [ 40.    39.    21.   100.  ]]
significance=0.050, p=0.218
critical value =  12.591587243743977
```

Figure 5: Output

```python
# Test result summary

print('probability=%.3f, critical=%.3f, stat=%.3f' % (prob, critical, stat))
if abs(stat) >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')

# Interpret p-value

alpha = 1.0 - prob
print('significance=%.3f, p=%.3f' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```

Figure 6: Input

```
probability=0.950, critical=12.592, stat=8.282
Independent (fail to reject H0)
significance=0.050, p=0.218
Independent (fail to reject H0)
```

Figure 6: Output

```python
# Creating a Contingency table for Area and Churn variables

contingency_table_2 = pd.crosstab(data['Churn'], data['Area'], margins=False)

print(contingency_table_2)
```

Figure 7: Input

Area	Rural	Suburban	Urban
Churn			
No	21	25	27
Yes	8	14	5

Figure 7: Output

```python
# Perform Chi-Square Test on Churn and Area variables

stat, p, dof, expected = chi2_contingency(contingency_table_2)
print("stat = ", stat)
print("p-value = ", p)
print("Degrees of Freedom =", dof)
print("Expected = ", expected)

# Calculate alpha and critical values and interpret test-statistic

prob = 0.95
critical = chi2.ppf(prob, dof)
print('significance=%.3f, p=%.3f' % (1-prob, p))
print("critical value = ", critical)
```

Figure 8: Input

```
stat =  3.6721845629891616
p-value =  0.15943925407494108
Degrees of Freedom = 2
Expected =  [[21.17 28.47 23.36]
 [ 7.83 10.53  8.64]]
significance=0.050, p=0.159
critical value =  5.991464547107979
```

Figure 8: Output

```
# Test result summary

print('probability=%.3f, critical=%.3f, stat=%.3f' % (prob, critical, stat))
if abs(stat) >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')

# Interpret p-value

alpha = 1.0 - prob
print('significance=%.3f, p=%.3f' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```

Figure 9: Input

```
probability=0.950, critical=5.991, stat=3.672
Independent (fail to reject H0)
significance=0.050, p=0.159
Independent (fail to reject H0)
```

Figure 9: Output

```
# Creating a Contingency table for Churn and Multiple variables

contingency_table_3 = pd.crosstab(data['Churn'], data['Multiple'], margins=False)

print(contingency_table_3)

df2 = pd.read_csv("churn_clean.csv", usecols=["MonthlyCharge", "Bandwidth_GB_Year", "Item1", "Item2"]
                  )
```

Figure 10: Input

```
Multiple  No  Yes
Churn
No        34   30
Yes       18   18
```

Figure 10: Output

```
# Randomly selecting data from our Churn dataset

data2 = df2.sample(n=100)

print(data2)
```

Figure 11: Input

```
      MonthlyCharge  Bandwidth_GB_Year  Item1  Item2
4317     165.018200        6274.153212      6      5
5497     242.628100        5500.343313      4      4
1138     129.991500        5771.251702      5      4
1157     240.114900        1571.238893      3      3
5309     230.134400        1256.972232      5      4
...             ...                ...    ...    ...
3937     182.453800        6273.353878      4      3
3711     139.981577         738.946457      3      3
6224     104.992300         498.439920      4      2
5548     210.127000        1215.186143      4      4
6700     124.964300         965.466162      4      4

[100 rows x 4 columns]
```

Figure 11: Output

B3. Justification of Analytical Technique

A chi-square test is a statistical test which is used to compare observed and expected results. The goal of this test is to identify whether a disparity between actual and predicted data is due to chance or to a link between the variables under consideration; the categorical variable we are analyzing is churn rate, we want to determine if a relationship exists between our variable and if a relationship is proven to exist what affect does it have on churn rates (Chi Square, nd).

C. Univariate Statistics

To perform our univariate statistics, we will be using "***monthlycharge***" and "***bandwidth_gb_year***" for our continuous variables and *"**item 1**"* (timley_responses) and "***item 2***" (timely_fixes) as our categorical variables from our churn dataset.

C2. Visual of Findings

```
115  # Create histograms for MonthlyCharge and Bandwidth_GB_Year
116
117  histogram = df2.hist()
118
119  plt.show()
120
121  # Creating histograms for Item1 and Item2
122
123  df3 = pd.read_csv("churn_clean.csv", usecols=["Item1", "Item2"])
124
125  # Rename Item1 and Item2
126
127  dict = {'Item1': 'Timely_Responses',
128          'Item2': 'Timely_Fixes'}
129
130  df3.rename(columns=dict, inplace=True)
131  hist = df3.hist()
132  plt.show()
```

Figure 12: Input - Creating Histograms for Monthly Charge, Bandwidth, Item 1 and 2

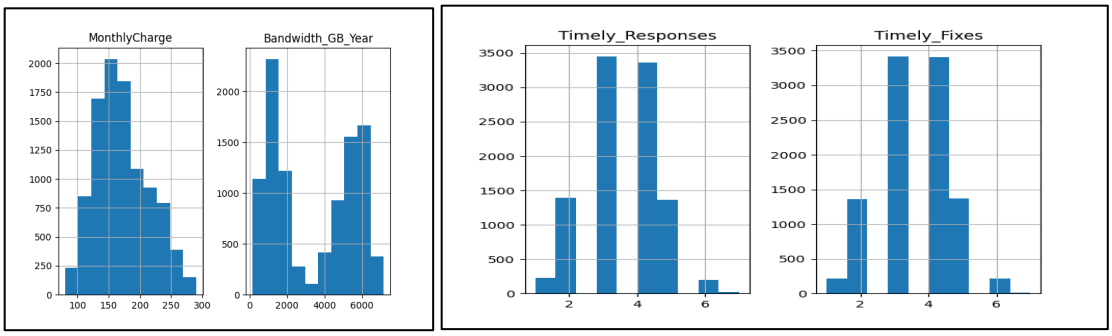

Figure 12: Output

```
# Creating Boxplots for Variables

df4 = pd.read_csv("churn_clean.csv", usecols=["MonthlyCharge", "Bandwidth_GB_Year", "Item1", "Item2"])

sns.boxplot(y=df4["MonthlyCharge"])

plt.show()

sns.boxplot(y=df4["Bandwidth_GB_Year"])

plt.show()

sns.boxplot(y=df4["Item1"])
plt.ylabel("Timely_Responses")

plt.show()

sns.boxplot(y=df4["Item2"])
plt.ylabel("Timely_Fixes")

plt.show()
```

Figure 13: Boxplots for our Variables

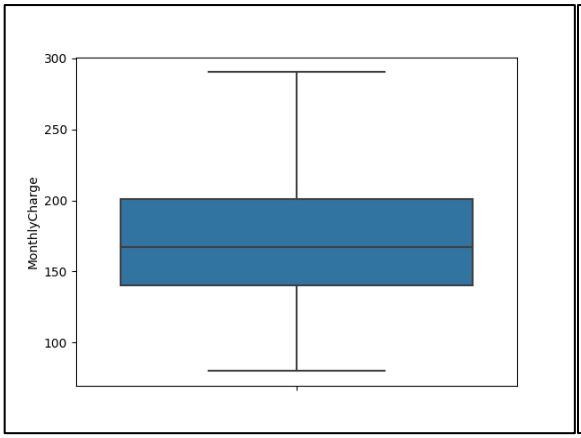

Figure 13: Output - Monthly Charges.

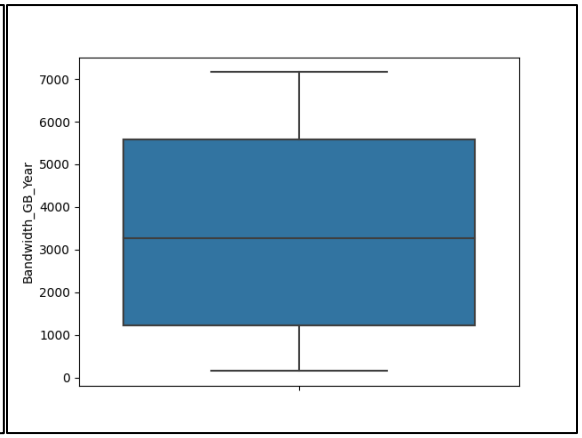

Figure 13: Output - Bandwith_GB_Year

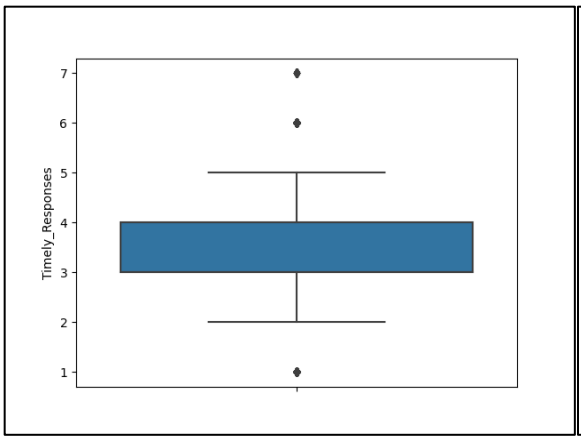

Figure 13: Output - Timely Responses

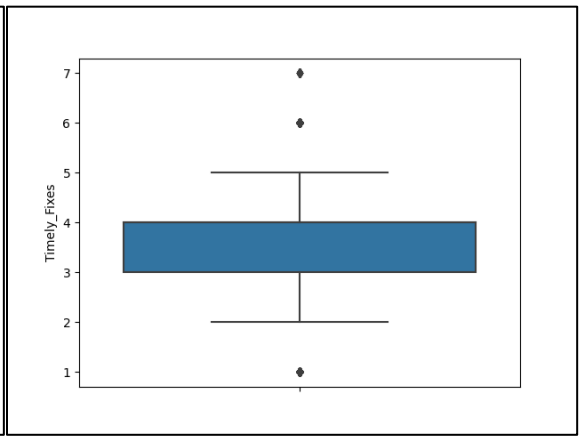

Figure 13: Output - Timely Fixes

D. Bivariate Statistics

To perform our bivariate statistics, we will be using "***monthlycharge***" and "***bandwidth_gb_year***" for our continuous variables and *"**item 1**"* (timley_responses) and "***item 2***" (timely_fixes) as our categorical variables from our churn dataset. I will be using heatmaps and scatterplots to perform the bivariate statistics; the result of the scatterplot shows that there isn't a strong relationship between or variables; however, our correlation matrix shows that there is some correlation between Item 1 (***timely_responses***) an Item 2 (***timely_fixes***).

D2. Visual of Findings

```
120  # Create a scatter plot of continuous variables MonthlyCharge & Bandwidth_GB_Year
121
122  x = data2['MonthlyCharge']
123  y = data2['Bandwidth_GB_Year']
124
125  plt.scatter(x, y, c='red')
126  plt.scatter(x, y, c='blue')
127
128  plt.xlabel("MonthlyCharge")
129  plt.ylabel("Bandwidth_GBYear")
130
131  print(np.corrcoef(x, y))
132
133  plt.plot(np.unique(x), np.poly1d(np.polyfit(x, y, 1))(np.unique(x)), color='red')
134
135  plt.show()
```

Figure 14: Input - Scatterplot Monthly Charge and Bandwidth

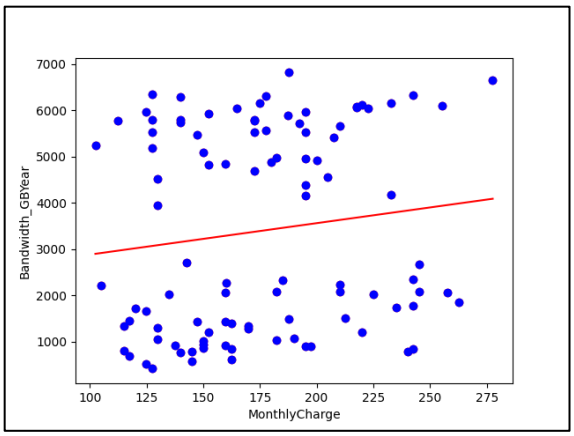

Figure 14: Output

```
186  # Correlation Matrix
187  sns.heatmap(df5.corr(), linewidths=.3, annot=True)
188  plt.show()
```

Figure 15: Input - Heatmap (Correlation Matrix)

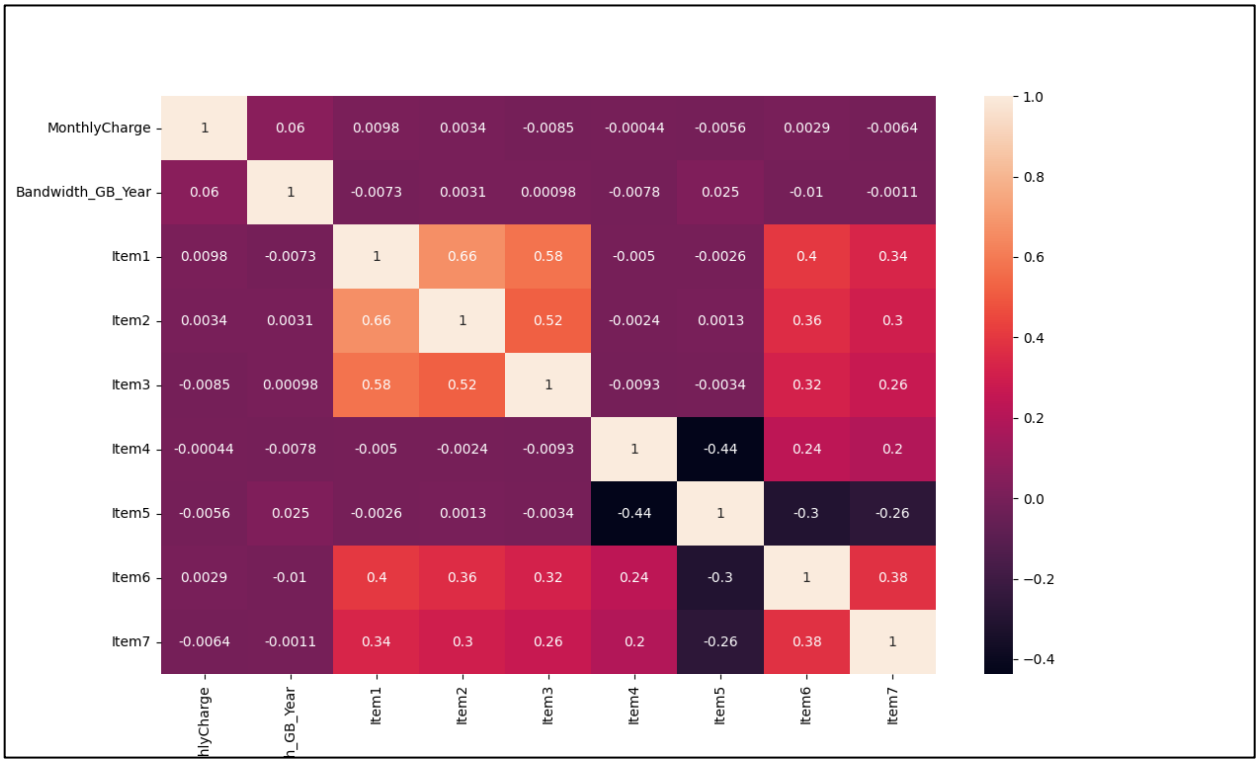

Figure 15: Output

E. Summary

We performed Chi-Square test on five different variables - ***churn, item 1, item 2, internet service type***, and ***area***; from our analysis we can see that there is no direct correlation between any of our chosen variables and customers who churn. All tests proved that the variables are independent of each other and have no impact on weather a customer churn or not. The dataset may have some limitations because of how unbalanced it is, only a small portioned of customers have churned.

I would recommend that another study be done with more balanced data. I would also focus more on ***tenure, contract, monthly charge,*** and ***area***. By doing this, this will allow the company to see if these variables have an impact on customers who churn. Furthermore, it will also allow the company to revamp its contracts length and monthly fees.

F. Panopto video recording

Video Link

References

Chi Square. Chi Square | Practical Applications of Statistics in the Social Sciences | University of Southampton. (n.d.). Retrieved May 3, 2022, from https://www.southampton.ac.uk/passs/full_time_education/bivariate_analysis/chi_square.page

Choueiry, G. (n.d.). *P-value: A simple explanation for non-statisticians*. Quantifying Health. Retrieved May 3, 2022, from https://quantifyinghealth.com/p-value-explanation/

DataCamp. (n.d.). Retrieved May 3, 2022, from https://app.datacamp.com/learn/courses/data-manipulation-with-pandas

OEM2 Task 1: EDA: Exploratory Data Analysis (n.d.). Retrieved May 3, 2022, from https://tasks.wgu.edu/student/000194226/course/23260006/task/2786/overview