Performance Assesment D207: Exploratory Data Analysis

Information

Matthew Blasa
Student ID: 001781641
MS Data Analytics (05/01/2021)
Program Mentor: Kirk Kelly
(503)805-0297
mblasa@wgu.edu

A1. Question to Analyze

Which customers are at high risk of churning? What features of the customer responses predict churn?

A2. Analysis Benefits

Finding which customers are at the greatest risk of churn is important, since it allows stakeholders to find key patterns for churn and diagnose them. This will allow them to reduce churn, improve customer experiences, and win back customers.

Stakeholders in the company will benefit by knowing, with some measure of confidence, which customers are at highest risk of churn because this will provide weight for decisions in marketing improved services to customers with these characteristics and past user experiences.

A3. Data Identification

Data we will be using is the dependent variable Churn, which is a binary and cateogrical.

The target variables are Tenure" (the number of months the customer has stayed with the provider), "MonthlyCharge" (the average monthly charge to the customer) & "Bandwidth_GB_Year" (the average yearly amount of data used, in GB, per customer).

I will also be using discrete numerical data from the survey from customers. This customer survey had rankings of individual customers' experiences, on a scale of 1 -8, with 1 being most important, and 8 being least important. The categories of customer service factors were: "timely response", "timely fixes", "timely replacements", "reliability", "options", "respectful response", "courteous exchange" & "evidence of active listening".

B1. Code

I will use Chi-Square Test to run a hypothesis test. Specifically, I will use Chi-Square Goodness of

Fit Test to generate a p-value, since the data I will be examining will be categorical and ordinal.

Library Imports

In [1]:

```
import numpy as np
import pandas as pd
from pandas import DataFrame

#visualization
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

#statistics
import pylab
import statsmodels.api as sm
import statistics
from scipy import stats

#Chi-Square
from scipy.stats import chisquare
from scipy.stats import chi2_contingency
```

In [2]:

```
# Load data set into Pandas dataframe
df = pd.read_csv('churn_clean.csv')
```

In [3]:

```
# Rename the survey columns to the response critera.
df.rename(columns = {'Item1':'TimelyResponse',
                    'Item2':'Fixes',
                     'Item3':'Replacements',
                     'Item4':'Reliability',
                     'Item5':'Options',
                     'Item6':'Respectfulness',
                     'Item7':'Courteous',
                     'Item8':'Listening'},
          inplace=True)
```

In [4]:

```
df.describe()
```

Out[4]:

	CaseOrder	Zip	Lat	Lng	Population	Children	Ag
count	10000.00000	10000.000000	10000.000000	10000.000000	10000.000000	10000.0000	10000.0000(
mean	5000.50000	49153.319600	38.757567	-90.782536	9756.562400	2.0877	53.0784(
std	2886.89568	27532.196108	5.437389	15.156142	14432.698671	2.1472	20.6988!
min	1.00000	601.000000	17.966120	-171.688150	0.000000	0.0000	18.0000(
25%	2500.75000	26292.500000	35.341828	-97.082812	738.000000	0.0000	35.0000(
50%	5000.50000	48869.500000	39.395800	-87.918800	2910.500000	1.0000	53.0000(
75%	7500.25000	71866.500000	42.106908	-80.088745	13168.000000	3.0000	71.0000(

	CaseOrder	Zip	Lat	Lng	Population	Children	Ag
max	10000.00000	99929.000000	70.640660	-65.667850	111850.000000	10.0000	89.0000

In [5]:

```
contingency = pd.crosstab(df['Churn'], df['TimelyResponse'])
contingency
```

Out[5]:

TimelyResponse	1	2	3	4	5	6	7
Churn							
No	158	1002	2562	2473	994	146	15
Yes	66	391	886	885	365	53	4

In [6]:

```
contingency_pct = pd.crosstab(df['Churn'], df['TimelyResponse'], normalize='in
contingency_pct
```

Out[6]:

TimelyResponse	1	2	3	4	5	6	7
Churn							
No	0.021497	0.136327	0.348571	0.336463	0.135238	0.019864	0.002041
Yes	0.024906	0.147547	0.334340	0.333962	0.137736	0.020000	0.001509

In [7]:

```
plt.figure(figsize=(12,8))
sns.heatmap(contingency, annot=True, cmap="YlGnBu")
```

Out[7]: <AxesSubplot:xlabel='TimelyResponse', ylabel='Churn'>

B2. Output

In [8]:

```
from scipy.stats import chi2
significance = 0.05
p = 1 - significance
dof = chi2_contingency(contingency)[2]
critical_value = chi2.ppf(p, dof)
critical_value
```

Out[8]:

```
12.591587243743977
```

In [9]:

```
# Chi-square test of independence
chi, pval, dof, exp = chi2_contingency(contingency)
print('p-value is: ', pval)
significance = 0.05
p = 1 - significance
critical_value = chi2.ppf(p, dof)
```

```
p-value is:  0.6318335816054494
```

In [10]:

```
print('chi=%.6f, critical value=%.6f\n' % (chi, critical_value))
```

```
chi=4.332078, critical value=12.591587
```

Chi square is smaller than the critical value, meaning the results are not statistically siginficant.

In [11]:

```
if chi > critical_value:
    print("""At %.2f level of significance, we reject the null hypotheses and
They are not independent.""" % (significance))
else:
    print("""At %.2f level of significance, we accept the null hypotheses.
They are independent.""" % (significance))
```

```
At 0.05 level of significance, we accept the null hypotheses.
They are independent.
```

B3. Justification

We are using Chi-Square since we are looking at categorical variables, and since it is a non-parametric test (Bruce p. 214). The first categorical variable is churn is binary, so it is approproiate for a non-parametic test. The other categorical variable that we are using, 'timely response' is an ordinal, meaning that it is ranking based.

C. Univariate Statistics

There are two continuous variables in the data set:

1. MonthlyCharge

2. Bandwidth_GB_Year

There are also two categorical variables, which are ordinal:

1. Item1 (Timely response) - relabeled "TimelyResponse"
2. Item7 (Courteous exchange) - relabeled "Courteous"

In [12]:
```
df.describe()
```

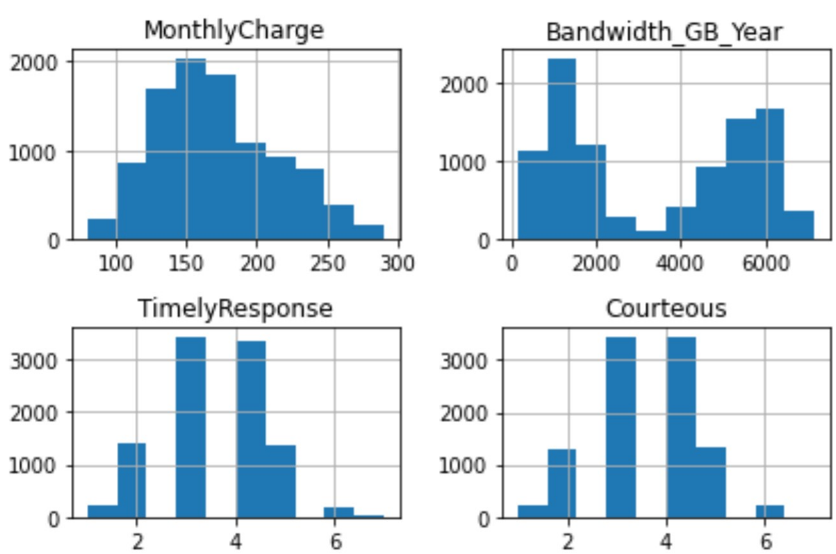
Out[12]:

	CaseOrder	Zip	Lat	Lng	Population	Children	Ag
count	10000.00000	10000.000000	10000.000000	10000.000000	10000.000000	10000.0000	10000.0000
mean	5000.50000	49153.319600	38.757567	-90.782536	9756.562400	2.0877	53.0784
std	2886.89568	27532.196108	5.437389	15.156142	14432.698671	2.1472	20.6988
min	1.00000	601.000000	17.966120	-171.688150	0.000000	0.0000	18.0000
25%	2500.75000	26292.500000	35.341828	-97.082812	738.000000	0.0000	35.0000
50%	5000.50000	48869.500000	39.395800	-87.918800	2910.500000	1.0000	53.0000
75%	7500.25000	71866.500000	42.106908	-80.088745	13168.000000	3.0000	71.0000
max	10000.00000	99929.000000	70.640660	-65.667850	111850.000000	10.0000	89.0000

8 rows × 23 columns

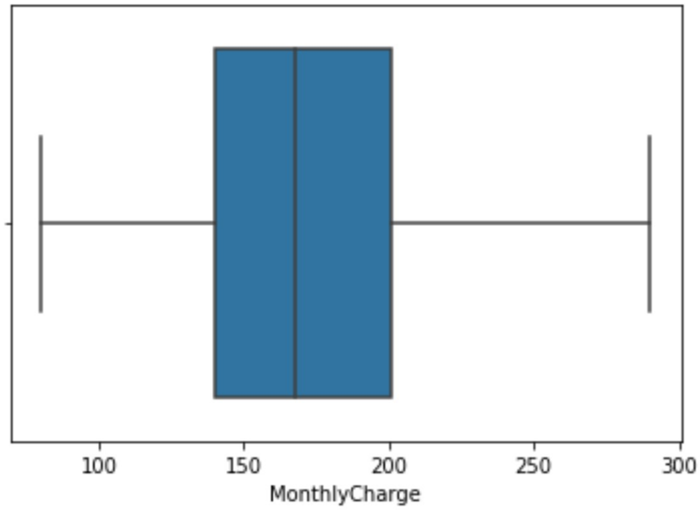
C1. Visualization of Results

In [13]:
```
# Create histograms of contiuous & categorical variables
df[['MonthlyCharge', 'Bandwidth_GB_Year', 'TimelyResponse', 'Courteous']].hist
plt.savefig('churn_plot.jpg')
plt.tight_layout()
```



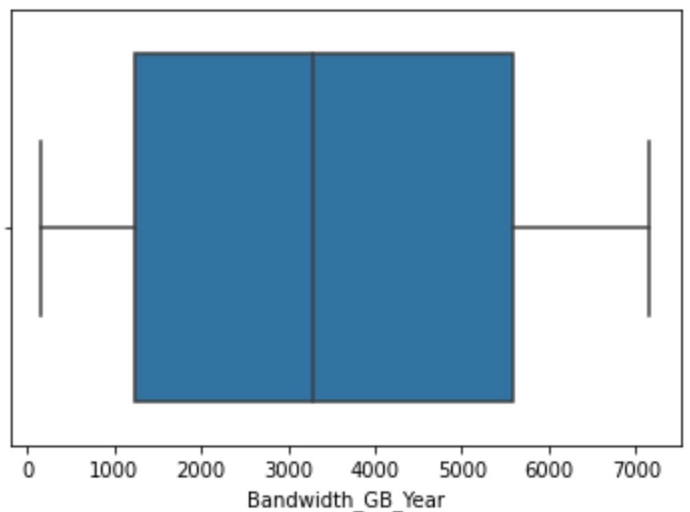
In [14]:
```
# Create Seaborn boxplots for continuous & categorical variables
sns.boxplot('MonthlyCharge', data = df)
plt.show()
```

```
C:\Users\blasa\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWa
rning: Pass the following variable as a keyword arg: x. From version 0.12, the
only valid positional argument will be `data`, and passing other arguments wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [15]:
```
sns.boxplot('Bandwidth_GB_Year', data = df)
plt.show()
```

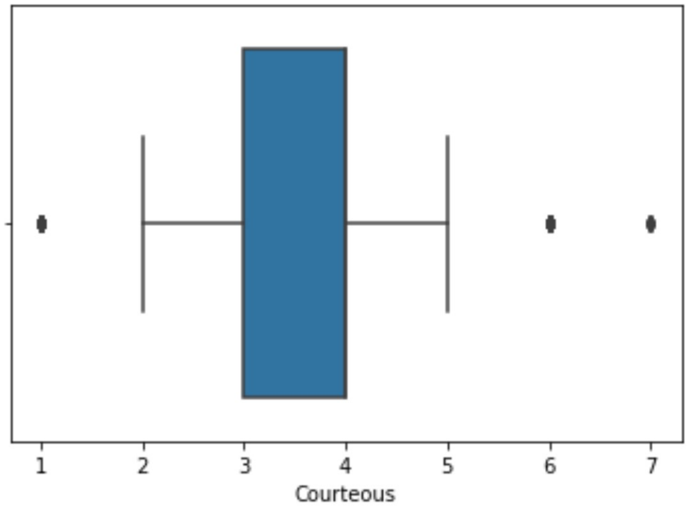
```
C:\Users\blasa\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWa
rning: Pass the following variable as a keyword arg: x. From version 0.12, the
only valid positional argument will be `data`, and passing other arguments wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [16]:
```
sns.boxplot('Courteous', data = df)
plt.show()
```

```
C:\Users\blasa\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWa
rning: Pass the following variable as a keyword arg: x. From version 0.12, the
```

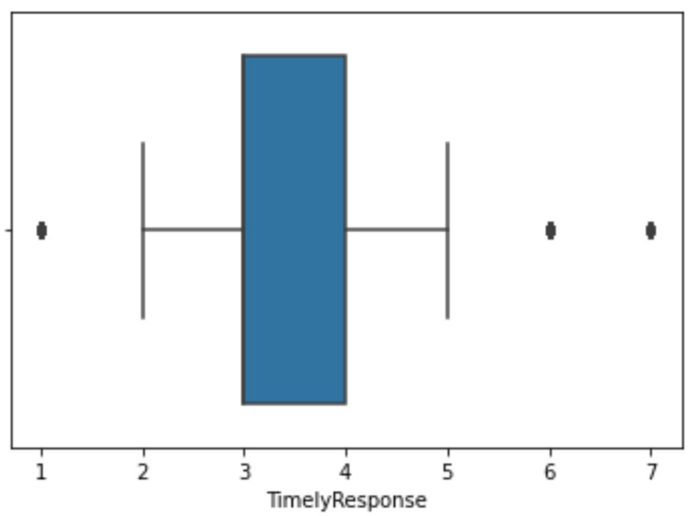
```
only valid positional argument will be `data`, and passing other arguments wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [17]:

```
sns.boxplot('TimelyResponse', data = df)
plt.show()
```

```
C:\Users\blasa\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWa
rning: Pass the following variable as a keyword arg: x. From version 0.12, the
only valid positional argument will be `data`, and passing other arguments wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



D1. Visualization of Findings

Two continuous variables:

1. MonthlyCharge
2. Bandwidth_GB_Year

Two categorical variables:

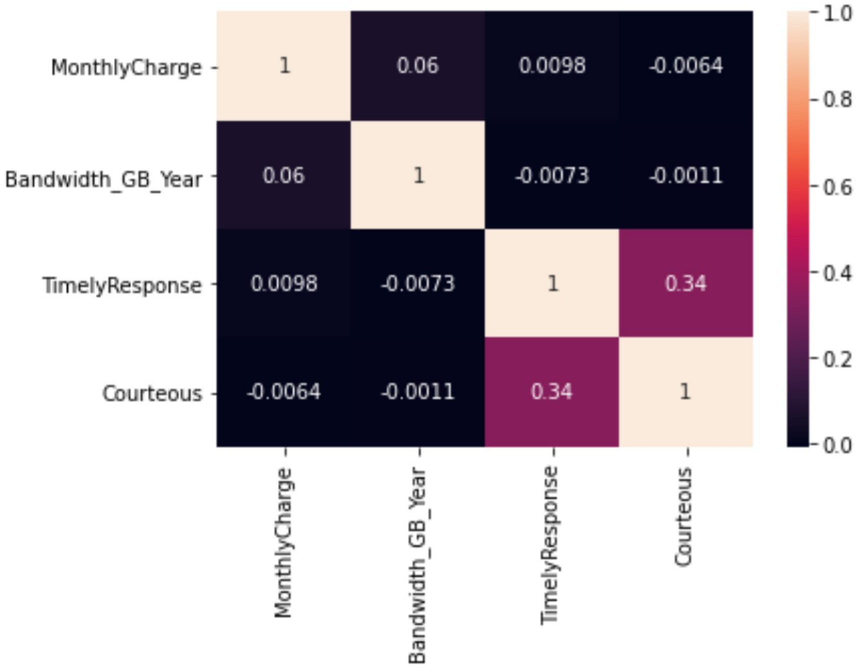
1. Churn - (Binary)
2. Item7 Courteous - (ordinal)

In [18]:

```
# Create dataframe for heatmap bivariate analysis of correlation
churn_bivariate = df[['MonthlyCharge', 'Bandwidth_GB_Year', 'TimelyResponse',
```

In [19]:

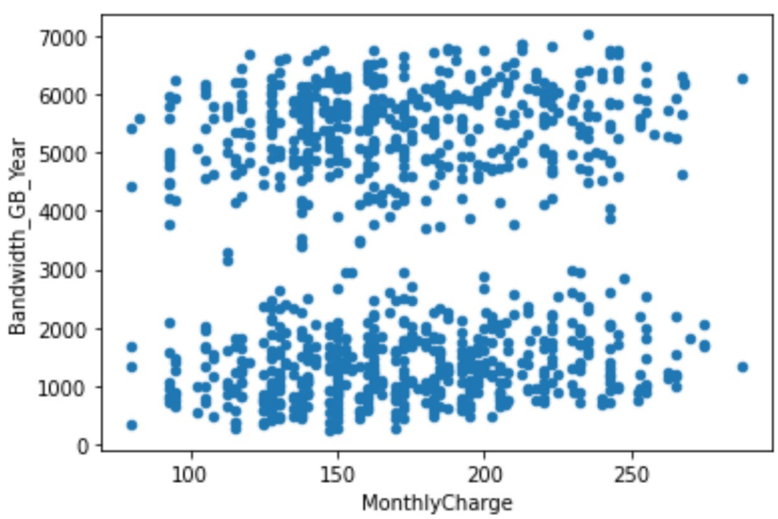
```
#correlation
sns.heatmap(churn_bivariate.corr(), annot=True)
plt.show()
```



In [20]:

```
# Scatter plot of continuous variables MonthlyCharge & Bandwidth_GB_Year
churn_bivariate[churn_bivariate['MonthlyCharge'] < 400].sample(1000).plot.scat
```

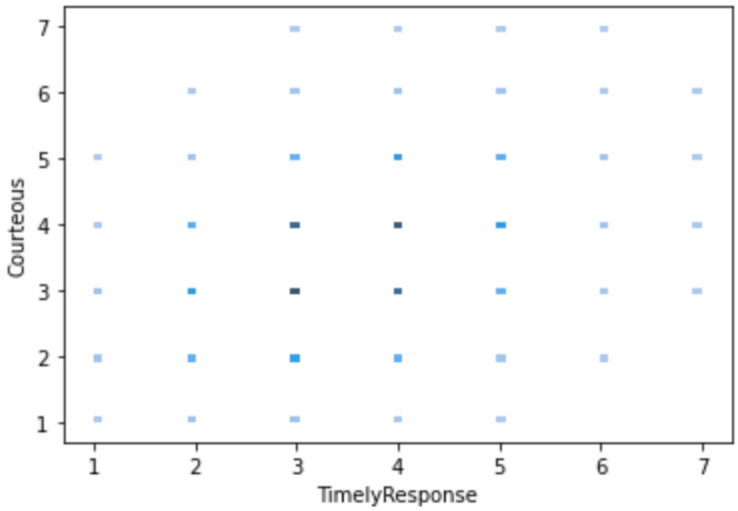
Out[20]: <AxesSubplot:xlabel='MonthlyCharge', ylabel='Bandwidth_GB_Year'>



In [21]:

```
# Scatter plot of categorical variables TimelyResponse & Courteous
churn_bi = churn_bivariate[churn_bivariate['TimelyResponse']<9]
sns.histplot(data=churn_bi, x="TimelyResponse", y="Courteous")
```

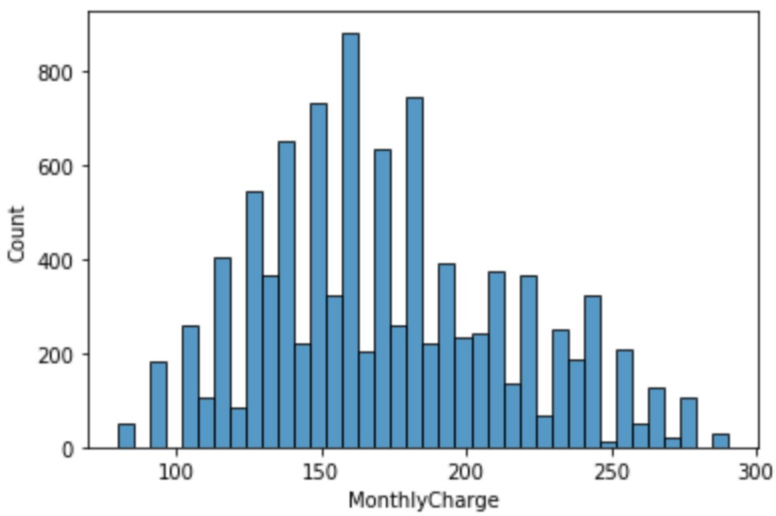
Out[21]: <AxesSubplot:xlabel='TimelyResponse', ylabel='Courteous'>



In [22]:

```
monthly_ch = churn_bivariate[churn_bivariate['MonthlyCharge'] < 300]
sns.histplot(data=monthly_ch, x='MonthlyCharge')
```

Out[22]: <AxesSubplot:xlabel='MonthlyCharge', ylabel='Count'>



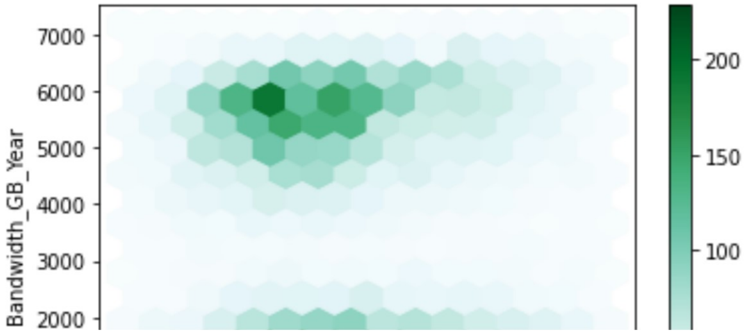
In [23]:

```
bivariate_churn = churn_bivariate[churn_bivariate['MonthlyCharge']<400]
```

In [24]:

```
bivariate_churn.plot.hexbin(x='MonthlyCharge', y='Bandwidth_GB_Year', gridsize
```

Out[24]: <AxesSubplot:xlabel='MonthlyCharge', ylabel='Bandwidth_GB_Year'>



E1. Results of Hypothesis Test

With a p-value as large as our output from our chi-square significance testing, p-value = 0.6318335816054494, we cannot reject the null hypothesis at a standard significance level of alpha = 0.05. It is unclear given the cleaned data available whether there is a statistically significant relationship between the survey responses, and if they caused the customer to churn. Since we must accept the null hypothesis, there is no effect or relationship between the variables.

E2 Limitations of Analysis

With a p-value = 0.6318335816054494, there are several assumptions that need to be investigated.

When we run a chi-square test, there is an assumption that observations are independent of each other (Okada). There is a possiblity that some of the customer groups may be related, since we do not know if they are living together or are from the same building. If a telecommunication problem affects a entire building, then there is the possiblity that they will generate similar responses - they are not truly independent. The data does not have specific addresses, only unverified GPS coordinates, so we do not know if the individuals are within the same building or household.

E3 Recommended Course of Action

The tests show very little correlation between the variables in timely action and courteous exchange. Another chi-square should be run on other categorical variables in the data that rejects the null hypothesis of churn. In order improve independence of observations to properly run a chi-square test, address and type of housing need to be identified. This will allow us to account issue that may occur in communal buildings, such as apartments, condos, or apartment complexes, so we can differentiate different addresses when we run the test.

A churn rate maybe higher in certain types of housing, due to the shared telecommunications equipment failing with multiple customers in a single complex. This may result certain areas having similar customer service responses. This may explain the high churn rates, and help the company identify areas that might result in lower survey ratings.

A better solution may to run statistical siginificance tests on continuous variables, or between cateogrical variables and continious ones, such as outages and timely action.

F. Video

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=fd21b7b0-8bbe-45ae-83bd-ad5c0123167c

G. Sources for Third-Party Code

Farrell, Peter, et al. The Statistics and Calculus with Python Workshop: A Comprehensive Introduction to Mathematics in Python for Artificial Intelligence Applications. Packt Publishing, 2020.

Okada, Shinichi. “Gentle Introduction to Chi-Square Test for Independence.” Medium, 20 May 2021, towardsdatascience.com/gentle-introduction-to-chi-square-test-for-independence-7182a7414a95.

H. Sources

Bruce, Peter, et al. Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python. 2nd ed., e-book, O’Reilly Media, 2020.

Okada, Shinichi. “Gentle Introduction to Chi-Square Test for Independence.” Medium, 20 May 2021, towardsdatascience.com/gentle-introduction-to-chi-square-test-for-independence-7182a7414a95.

Walker, Michael. *Python Data Cleaning Cookbook: Modern Techniques and Python Tools to Detect and Remove Dirty Data and Extract Key Insights*. Packt Publishing, 2020.

In [ ]:

```

```