D207_Performance_Assessment

June 1, 2021

1 Performance Assessment | D207 Exploratory Data Analysis Ryan L. Buchanan Student ID: 001826691 Masters Data Analytics (12/01/2020) Program Mentor: Dan Estes (385) 432-9281 (MST) rbuch49@wgu.edu

1.0.1 A1. Question for Analysis:

Which customers are at high risk of churn? And, which customer features/variables are most significant to churn?

1.0.2 A2. Benefit from Analysis:

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.

1.0.3 A3. Data Identification:

Most relevant to our decision making process is the dependent variable of "Churn" which is binary categorical with only two values, "Yes" or "No". In cleaning the data, we discovered relevance of the continuous numerical data columns "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). Finally, the discrete numerical data from the survey responses from customers regarding various customer service features is relevant in the decision-making process. In the surveys, customers provided ordinal numerical data by rating 8 customer service factors ("timely response", "timely fixes", "timely replacements", "reliability", "options", "respectful response", "courteous exchange" & "evidence of active listening") on a scale of 1 to 8 (1 = most important, 8 = least important).

1.0.4 B1. Code:

Chi-square testing will be used.

1.0.5 Standard imports

[1]:
```
# Standard data science imports
import numpy as np
import pandas as pd
from pandas import DataFrame
```

```
# Visualization libraries
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

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

# Import chisquare from SciPy.stats
from scipy.stats import chisquare
from scipy.stats import chi2_contingency
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19:
FutureWarning: pandas.util.testing is deprecated. Use the functions in the
public API at pandas.testing instead.
  import pandas.util.testing as tm
```

[3]:
```
# Load data set into Pandas dataframe
df = pd.read_csv('churn_clean.csv')
```

[4]:
```
# Rename last 8 survey columns for better description of variables
df.rename(columns = {'Item1':'TimelyResponse',
                     'Item2':'Fixes',
                     'Item3':'Replacements',
                     'Item4':'Reliability',
                     'Item5':'Options',
                     'Item6':'Respectfulness',
                     'Item7':'Courteous',
                     'Item8':'Listening'},
          inplace=True)
```

[5]:
```
contingency = pd.crosstab(df['Churn'], df['TimelyResponse'])
contingency
```

[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

[6]:
```
contingency_pct = pd.crosstab(df['Churn'], df['TimelyResponse'],␣
↪normalize='index')
contingency_pct
```

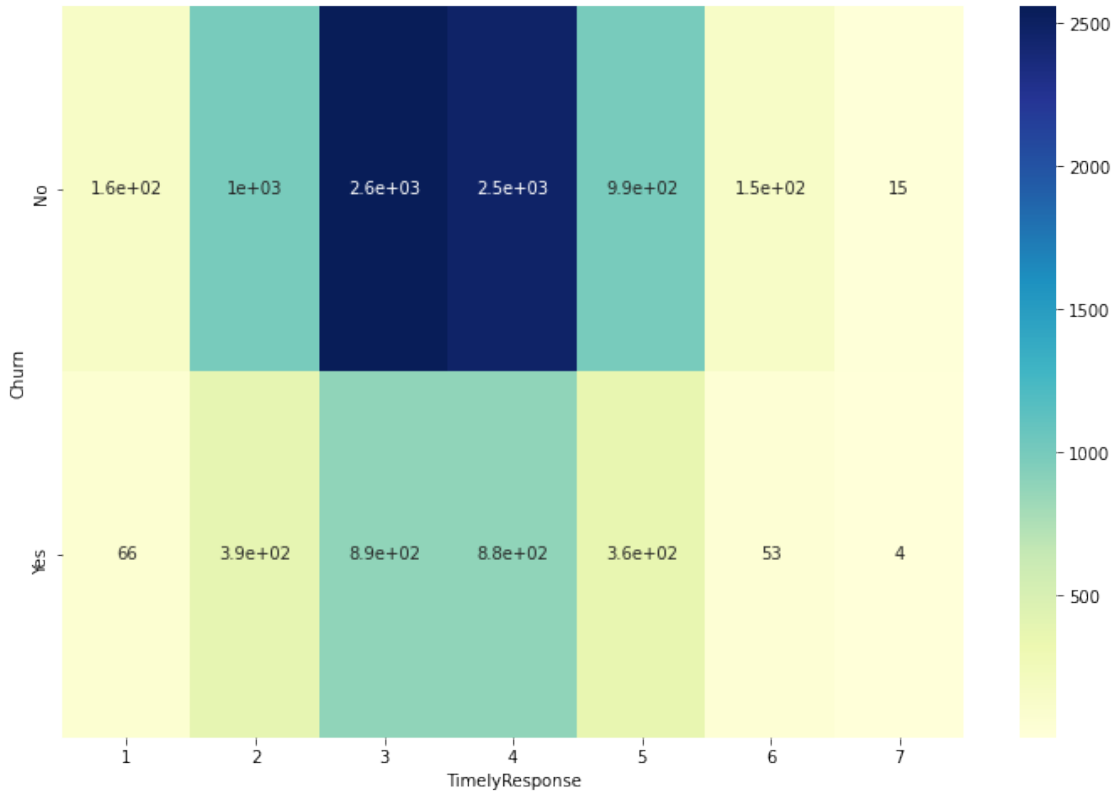
[6]:

TimelyResponse	1	2	3	...	5	6	7
Churn				...			
No	0.021497	0.136327	0.348571	...	0.135238	0.019864	0.002041

```
Yes             0.024906  0.147547  0.334340  ...  0.137736  0.020000  0.001509

[2 rows x 7 columns]
```

[7]:
```
plt.figure(figsize=(12,8))
sns.heatmap(contingency, annot=True, cmap="YlGnBu")
```

[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f63a0aeba90>
```



1.0.6 B2. Output:

[8]:
```
# Chi-square test of independence
c, p, dof, expected = chi2_contingency(contingency)
print('p-value = ' + str(p))
```

```
p-value = 0.6318335816054494
```

1.0.7 B3. Justification:

In this analysis, we are looking at churn from a telecom company ("Did customers stay with or leave the company?"). "Churn" is a binomial, categorical dependent variable. Therefore, we will use chi-square testing as it is a non-parametric test for this "yes/no" target variable. Our other categorical variable, "TimelyResponse", is at the ordinal level.

1.0.8 C. Univariate Statistics:

Two continuous variables:
1. MonthlyCharge 2. Bandwidth_GB_Year Two categorical (ordinal) variables: 1. Item1 (Timely response) - relabeled "TimelyResponse" 2. Item7 (Courteous exchange) - relabeled "Courteous"

[9]:
```
df.describe()
```

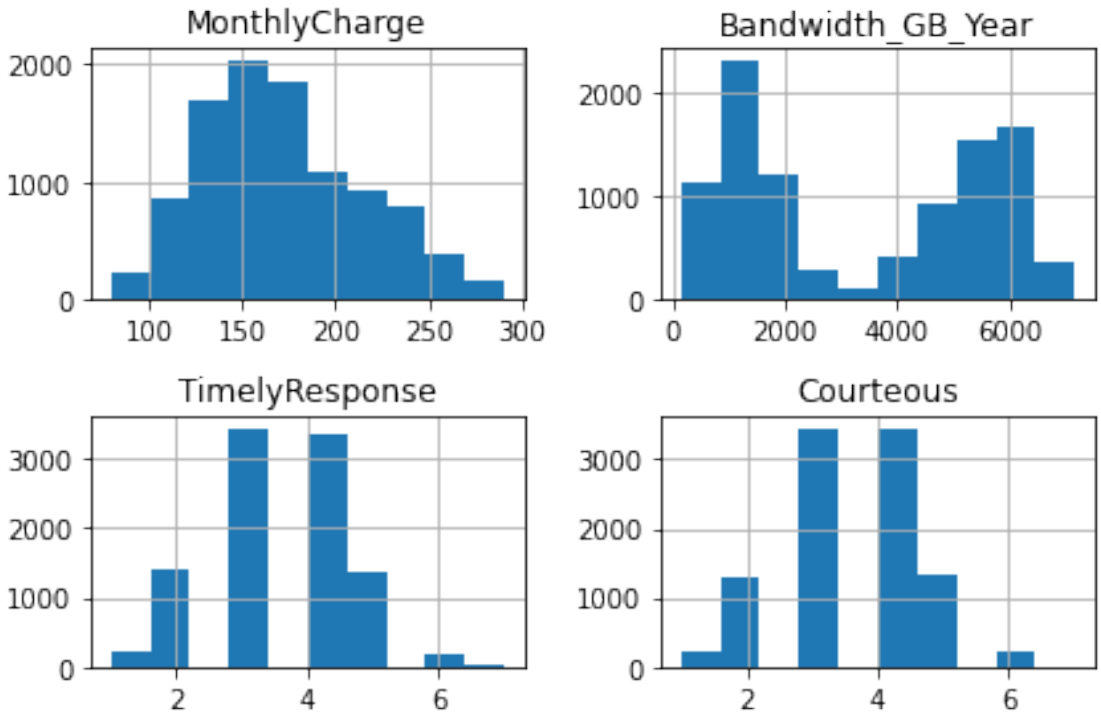
[9]:

	CaseOrder	Zip	...	Courteous	Listening
count	10000.00000	10000.000000	...	10000.000000	10000.000000
mean	5000.50000	49153.319600	...	3.509500	3.495600
std	2886.89568	27532.196108	...	1.028502	1.028633
min	1.00000	601.000000	...	1.000000	1.000000
25%	2500.75000	26292.500000	...	3.000000	3.000000
50%	5000.50000	48869.500000	...	4.000000	3.000000
75%	7500.25000	71866.500000	...	4.000000	4.000000
max	10000.00000	99929.000000	...	7.000000	8.000000

[8 rows x 23 columns]

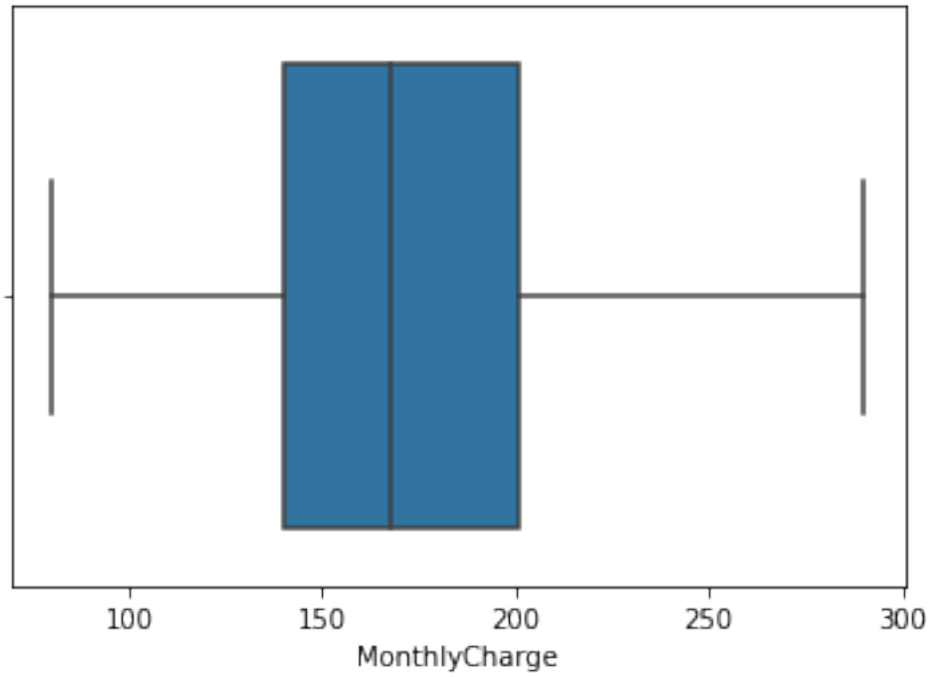
1.0.9 C1. Visual of Findings:

[10]:
```
# Create histograms of contiuous & categorical variables
df[['MonthlyCharge', 'Bandwidth_GB_Year', 'TimelyResponse', 'Courteous']].hist()
plt.savefig('churn_pyplot.jpg')
plt.tight_layout()
```



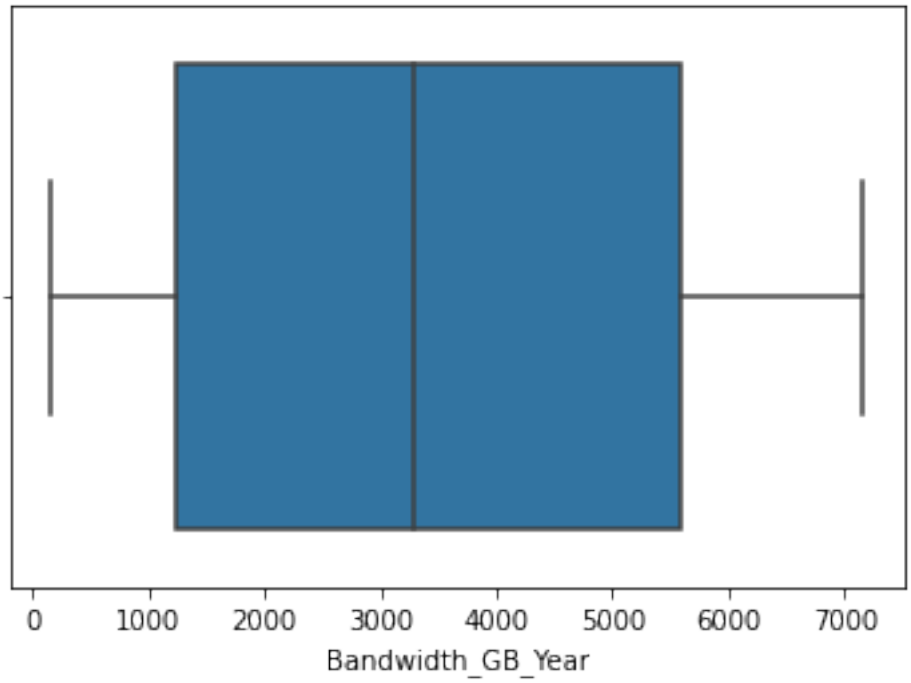
[11]:
```
# Create Seaborn boxplots for continuous & categorical variables
sns.boxplot('MonthlyCharge', data = df)
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
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 without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```



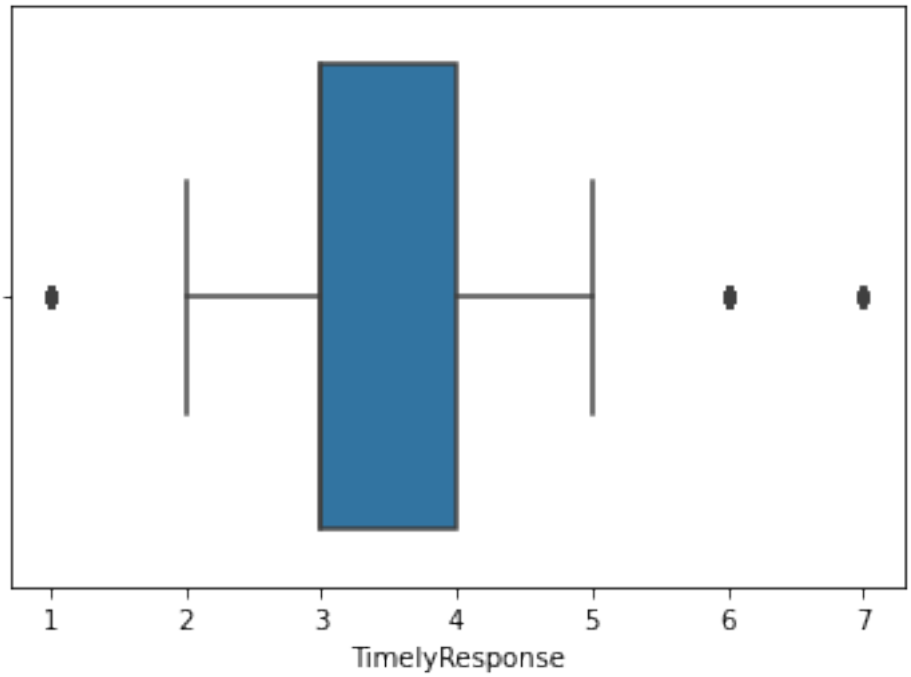
[12]:
```
sns.boxplot('Bandwidth_GB_Year', data = df)
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
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 without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```



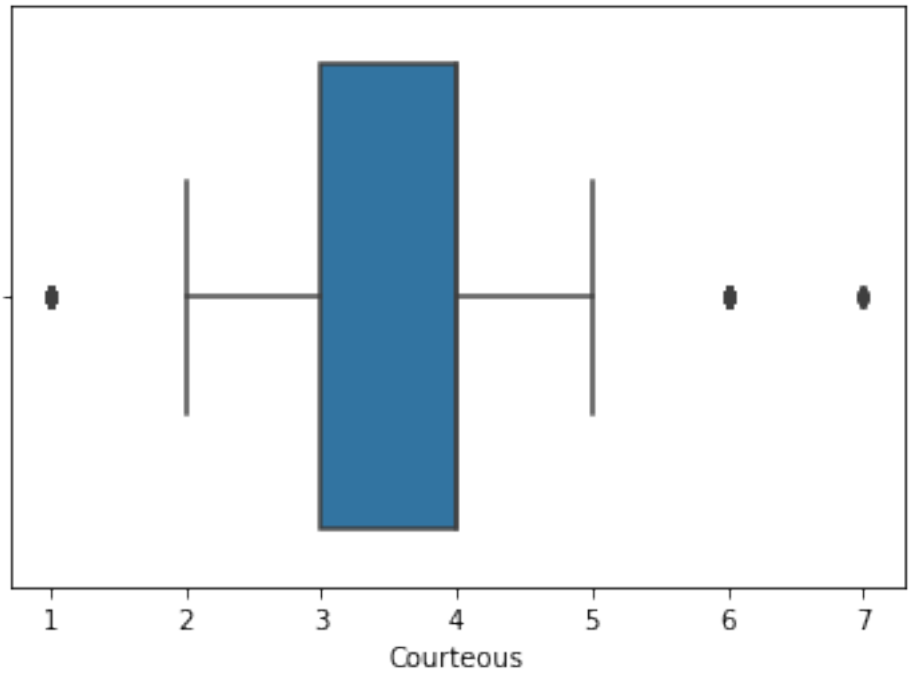
[13]:
```
sns.boxplot('TimelyResponse', data = df)
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
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 without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```



[14]:
```
sns.boxplot('Courteous', data = df)
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
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 without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```



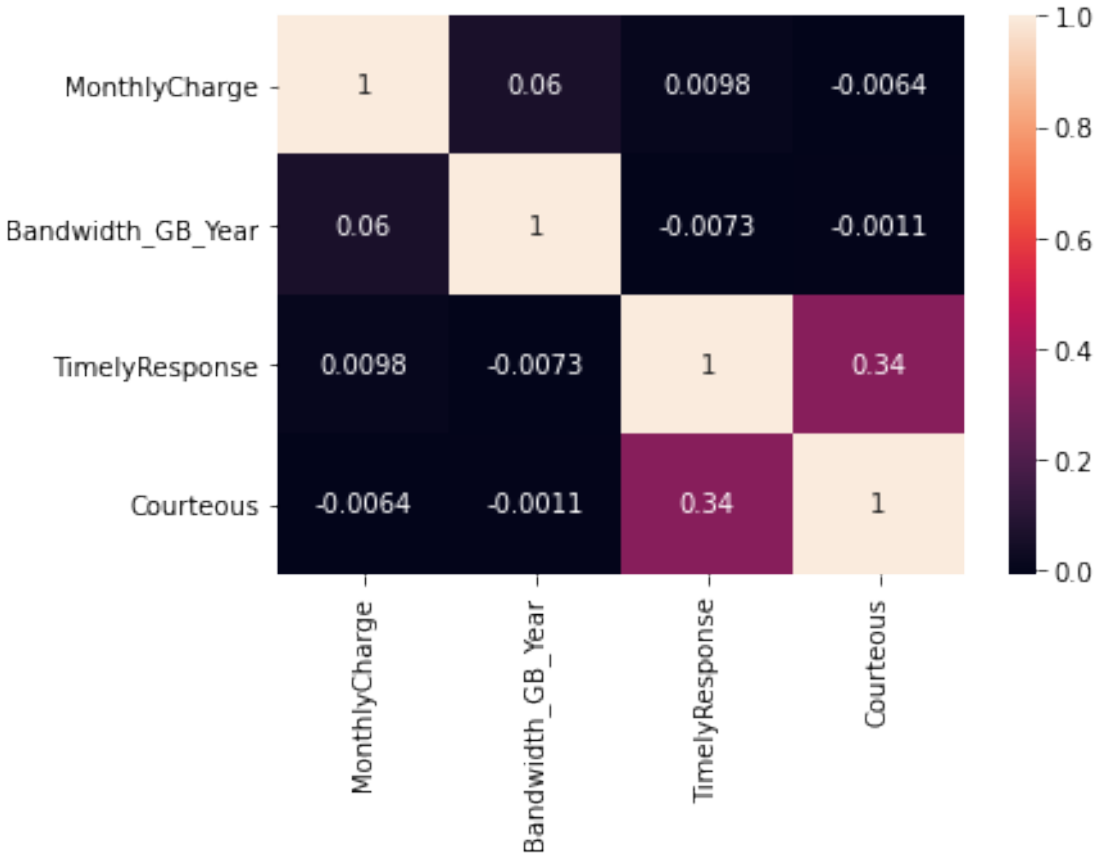
1.0.10 D. Bivariate Statistics

Two continuous variables:
1. MonthlyCharge 2. Bandwidth_GB_Year Two categorical (binomial & ordinal, respectively) variables: 1. Churn 2. Item7 (Courteous exchange) - relabeled "Courteous"

1.0.11 D1. Visual of Findings:

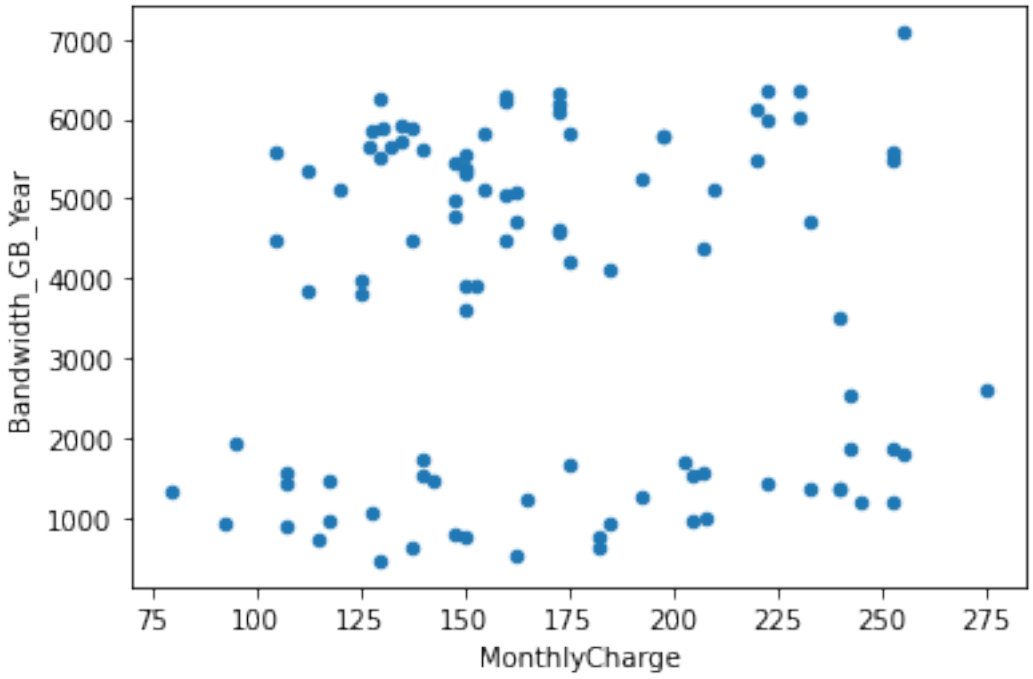
[15]:
```
# Create dataframe for heatmap bivariate analysis of correlation
churn_bivariate = df[['MonthlyCharge', 'Bandwidth_GB_Year', 'TimelyResponse',␣
 ↪'Courteous']]
```

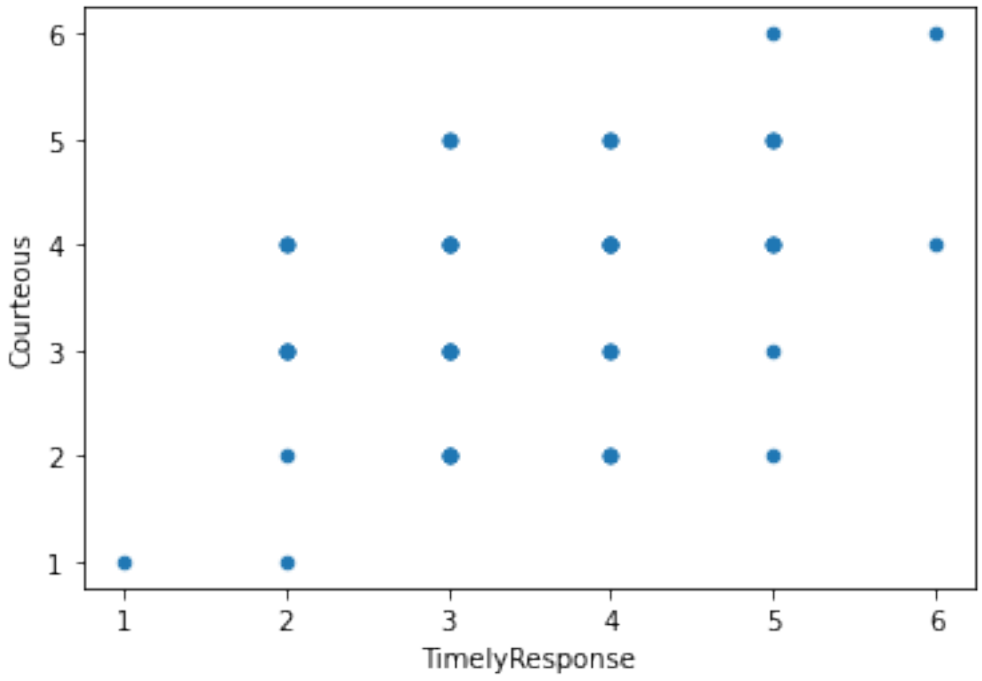
[16]:
```
sns.heatmap(churn_bivariate.corr(), annot=True)
plt.show()
```



[17]:
```
# Create a scatter plot of continuous variables MonthlyCharge &␣
↪Bandwidth_GB_Year
churn_bivariate[churn_bivariate['MonthlyCharge'] < 300].sample(100).plot.
↪scatter(x='MonthlyCharge',

↪ y='Bandwidth_GB_Year')

# Create a scatter plot of categorical variables TimelyResponse & Courteous
churn_bivariate[churn_bivariate['TimelyResponse'] < 7].sample(100).plot.
↪scatter(x='TimelyResponse',

↪ y='Courteous')
```

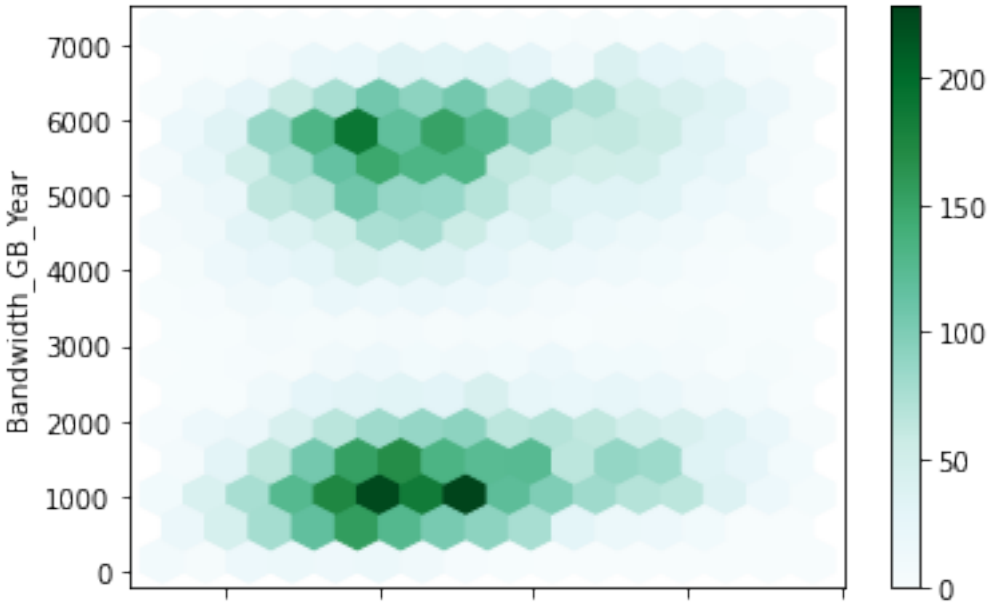
[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f639782ff90>
```





[18]:
```
churn_bivariate[churn_bivariate['MonthlyCharge'] < 300].plot.
↪hexbin(x='MonthlyCharge', y='Bandwidth_GB_Year', gridsize=15)
```

[18]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f6397a25a50>`



1.0.12 E1. Results of Analysis

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 (essentially, "How well did we the telecom company take care of you as a customer?") & whether or not this caused customers to leave the company.

1.0.13 E2. Limitations of Analysis:

Clearly, with a p-value that is so high, p-value = 0.6318335816054494, we need to investigate further & perhaps gather more & better data. It is troubling that this dataset has been so limited in our ability to gather meaningful & actionable information.

1.0.14 E3. Recommended Course of Action:

While tests show very little correlation & perhaps no linear relations between the variables involved in timely action with regard to customer satisfaction (TimelyResponses, Fixes, Replacements & Respectfulness), we believe that these elements should be given greater emphasis and hopefully help reduce the churn rate from the large number of 27% & "increase the retention period of customers" by targeting more resources in the direction prompt customer service (Ahmad, 2019, p. 1). Again, this seems an intuitive result but now decision-makers in the company of reasonable verification of what might have been a "hunch".

1.0.15 F. Video

Link here

1.0.16 G. Sources for Third-Party Code

Kaggle. (2018, May 01). Bivariate plotting with pandas. Kaggle. https://www.kaggle.com/residentmario/bivariate-plotting-with-pandas#

Sree. (2020, October 26). Predict Customer Churn in Python. Towards Data Science. https://towardsdatascience.com/predict-customer-churn-in-python-e8cd6d3aaa7

Wikipedia. (2021, May 31). Bivariate Analysis. https://en.wikipedia.org/wiki/Bivariate_analysis#:~:text=Biv

1.0.17 H. Sources

Ahmad, A. K., Jafar, A & Aljoumaa, K. (2019, March 20). Customer churn prediction in telecom using machine learning in big data platform. Journal of Big Data. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0191-6

Altexsoft. (2019, March 27). Customer Churn Prediction Using Machine Learning: Main Approaches and Models. Altexsoft. https://www.altexsoft.com/blog/business/customer-churn-prediction-for-subscription-businesses-using-machine-learning-main-approaches-and-models/

Bruce, P., Bruce A. & Gedeck P. (2020). Practical Statistics for Data Scientists. O'Reilly.

Freedman, D. Pisani, R. & Purves, R. (2018). Statistics. W. W. Norton & Company, Inc.

Frohbose, F. (2020, November 24). Machine Learning Case Study: Telco Customer Churn Prediction. Towards Data Science. https://towardsdatascience.com/machine-learning-case-study-telco-customer-churn-prediction-bc4be03c9e1d

Griffiths, D. (2009). A Brain-Friendly Guide: Head First Statistics. O'Reilly.

NIH. (2020). National Library of Medicine. https://www.nlm.nih.gov/nichsr/stats_tutorial/section2/mod11

P-Values. (2020). StatsDirect Limited. https://www.statsdirect.com/help/basics/p_values.htm

[ ]:
```
!wget -nc https://raw.githubusercontent.com/brpy/colab-pdf/master/colab_pdf.py
from colab_pdf import colab_pdf
colab_pdf('D207_Performance_Assessment.ipynb')
```

```
--2021-06-01 09:46:27--  https://raw.githubusercontent.com/brpy/colab-
pdf/master/colab_pdf.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)...
185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1864 (1.8K) [text/plain]
Saving to: colab_pdf.py

colab_pdf.py        100%[===================>]   1.82K  --.-KB/s    in 0s

2021-06-01 09:46:27 (35.9 MB/s) - colab_pdf.py saved [1864/1864]

Mounted at /content/drive/
```

```
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Extracting templates from packages: 100%
```