D207 Performance Assessment

Is there a significant difference between the monthly charge of customers who churn and those who do not?

Tyson Biegler

Student ID: ***012170282***

D207 - Exploratory Data Analysis

Date: 08/07/2024

Western Governors University

School of Technology, College of IT

Master of Science, Data Analytics

The initial setup

Im going to load in the libraries and the csv. After that I will rename the survey responses to better match the data dictionary so I can use them later.

In [1]:
```
# Initial setup
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
pd.options.display.max_columns = None #So that I can see all the columns
import statistics
from scipy import stats
```

```
import pylab as pl
import seaborn as sns
from scipy.stats import chi2_contingency
```

In [2]:
```
# importing csv
df = pd.read_csv('C:/Users/tyson/Documents/GitHub/WGU_MSDA_Portfolio/Exploratory Data Analysis - D207/Raw/churn_clear
df.head()
```

Out[2]:

	CaseOrder	Customer_id	Interaction	UID	City	State	County	Zip	Lat	
0	1	K409198	aa90260b-4141-4a24-8e36-b04ce1f4f77b	e885b299883d4f9fb18e39c75155d990	Point Baker	AK	Prince of Wales-Hyder	99927	56.25100	-133
1	2	S120509	fb76459f-c047-4a9d-8af9-e0f7d4ac2524	f2de8bef964785f41a2959829830fb8a	West Branch	MI	Ogemaw	48661	44.32893	-84
2	3	K191035	344d114c-3736-4be5-98f7-c72c281e2d35	f1784cfa9f6d92ae816197eb175d3c71	Yamhill	OR	Yamhill	97148	45.35589	-123
3	4	D90850	abfa2b40-2d43-4994-b15a-989b8c79e311	dc8a365077241bb5cd5ccd305136b05e	Del Mar	CA	San Diego	92014	32.96687	-117
4	5	K662701	68a861fd-0d20-4e51-a587-8a90407ee574	aabb64a116e83fdc4befc1fbab1663f9	Needville	TX	Fort Bend	77461	29.38012	-95

In [3]:
```
#renaming the survey responses to be more intuitive
#I will use Item6 and Item8 later.

df.rename(columns={'Item1':'Timely_response',
                   'Item2':'Timely_fixes',
                   'Item3':'Timely_replacements',
```

```
                    'Item4':'Reliability',
                    'Item5':'Options',
                    'Item6':'Respectful',
                    'Item7':'Courteous',
                    'Item8':'Active_listening'
                    },
          inplace=True)

df.columns[-8:] #making sure they were all changed correctly
```

Out[3]:
```
Index(['Timely_response', 'Timely_fixes', 'Timely_replacements', 'Reliability',
       'Options', 'Respectful', 'Courteous', 'Active_listening'],
      dtype='object')
```

A Describing the situation

A1. Research question

I want to discover **"Is there a significant difference between the monthly charge of customers who churn and those who do not?"**

A2. Benefits of analysis

The data dictionary mentions that some telecom companies can experience a 25% churn rate, and therefore, investigating the factors that contribute to customer churn is of utmost importance. If stakeholders knew precisely which factors contributed to churn, based on statistical analysis, they would have data-driven insights that would enable them to make meaningful changes for their customers. Some of these insights could lead to adjusting pricing, offering incentives based on tenure, increase training in customer interactions, focus more or less on different demographics and so on.

A3. Identification of the data

Because my question aims to determine if a specific variable impacts churn, the dependent variable will be 'Churn.' 'Churn' is a categorical variable that consists of 'Yes' and 'No' values. I will convert Churn to a binary before proceeding with statistical tests. Lastly, I will compare the MonthlyCharge of customers who have churned and those who have not.

I will be using the following variables to answer the question from part A1

Numeric variable

- MonthlyCharge

Qualitative variable

- Churn (Dependant Variable)

B. Describe the analysis

B1. Statstical test

In [4]:
```
#checking the unique values in Churn to make sure they are Yes and No
df['Churn'].unique()
print(df['Churn'].unique())
#converting Churn to binary
df['Churn'] = df['Churn'].map({'Yes':1, 'No':0})
print(df['Churn'].unique())
```

```
['No' 'Yes']
[0 1]
```

In [5]:
```
#distinguishing between churned or not within MonthlyCharge
monthly_charge_churned = df[df['Churn'] == 1]['MonthlyCharge']
print(monthly_charge_churned.describe())
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
monthly_charge_not_churned = df[df['Churn'] == 0]['MonthlyCharge']
print(monthly_charge_not_churned.describe())
```

```
count    2650.000000
mean      199.295175
std        41.268191
min        92.455140
25%       167.484705
50%       200.118500
75%       232.641455
max       290.160419
Name: MonthlyCharge, dtype: float64
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
count    7350.000000
mean      163.008973
std        39.322148
min        79.978860
25%       137.439154
50%       159.964200
75%       184.978458
max       290.160400
Name: MonthlyCharge, dtype: float64
```

In [6]:

```python
N1 = len(monthly_charge_churned)
N2 = len(monthly_charge_not_churned)

mean_churned = statistics.mean(monthly_charge_churned)
mean_not_churned = statistics.mean(monthly_charge_not_churned)

var_churned = statistics.variance(monthly_charge_churned)
var_not_churned = statistics.variance(monthly_charge_not_churned)

# Calculating the standard deviation of the variance
s = np.sqrt((var_churned + var_not_churned) / 2)

print('The mean of churned is: ', mean_churned)
print('The mean of not churned is: ', mean_not_churned)

# Calculating the t statistic
t = (mean_churned - mean_not_churned) / (s * np.sqrt(2 / (N1 + N2)))

# Degrees of freedom
degrees_freedom  = N1 + N2 - 2

# p-value after comparison with the t-distribution
```

```
p = 2 * (1 - stats.t.cdf(abs(t), df=degrees_freedom ))


print(str(N1) + " customers have churned")
print(str(N2) + " customers have not churned")
print("t = " + str(t))
print("p = " + str(p))

if p < 0.05:
    print("Reject the null hypothesis. There is a statistically significant difference")
else:
    print("Fail to reject the null hypothesis. There is no statistically significant difference")
```

```
The mean of churned is:  199.29517509886793
The mean of not churned is:  163.00897252612245
2650 customers have churned
7350 customers have not churned
t = 63.65711597680452
p = 0.0
Reject the null hypothesis. There is a statistically significant difference
```

B2. Output of analysis

There are 2650 churned customers, whose mean monthly charge is $199.30, whereas the 7350 customers who have not churned have a mean monthly charge of $163.01. The t-test indicates a statistically significant difference in monthly charges between the two groups. Therefore, I will reject the null hypothesis that the means of the two groups are the same.

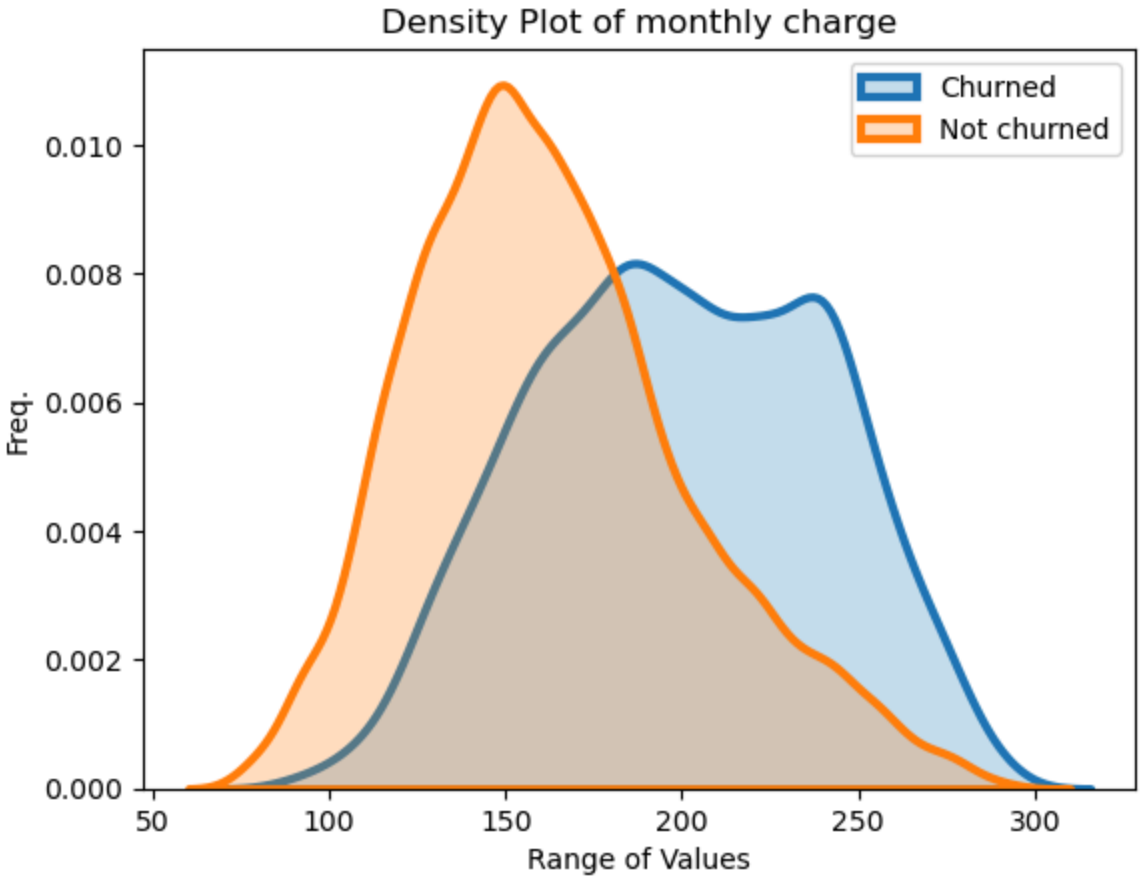
The t-value of 63.65 is very large, indicating a drastic difference between the two means of the two groups. The p-value of 0 indicates that the findings here are not likely to have occurred by random chance. likewise, a p-value less than .05 would suggest these findings are statistically significant. Therefore, we can reject the null hypothesis that there is no difference between the means of the groups.

In [7]:
```
#Desity plot of the monthly charge of customer who churned and customers who have not
#(Sewell, n.d.)

a=monthly_charge_churned
b=monthly_charge_not_churned

#seaborn histogram
```

```
sns.kdeplot(a, fill=True, linewidth=3, label='a')
sns.kdeplot(b, fill=True, linewidth=3, label='b')

#adding labels
plt.title('Density Plot of monthly charge')
plt.xlabel('Range of Values')
plt.ylabel('Freq.')

#adding legend
plt.legend(['Churned', 'Not churned'])
```

```
C:\Users\tyson\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecate
d and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\tyson\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecate
d and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[7]: <matplotlib.legend.Legend at 0x242ab86fad0>



B3. Justification of techniques

A t-test is most appropriate for this data because the monthly charge is a continuous numeric variable with only 2 groups being 'churned' and 'not churned.' (Martin, n.d)

C. Univariate statistics

In the data cleaning process I found a possible relationship between the two continuous variables Tenure and Bandwidth_GB_Year. Both have a bimodal distribution. Tenure has a mean tenure of 34.53, and Bandwidth_GB_Year has a mean GB usage of 3392.34 GB per year.

The two categorical variables that I am interested in are Respectful and Active_listening. Both of which are renamed survey response variables. Both variables appear visually similar and indicate that customers value 'Respectful' and 'Active_listening' at comparable rates. Both bar charts show a right-skewed chart, with most responses falling in the higher importance range.

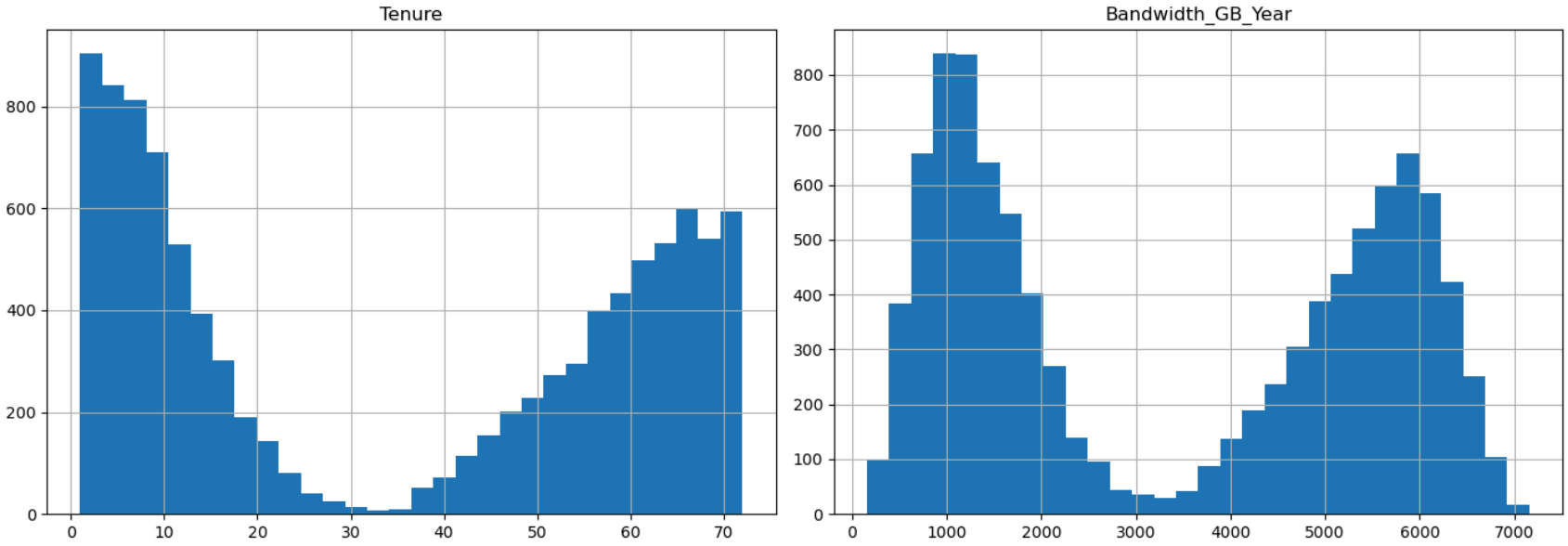
In [8]:
```
df[['Tenure','Bandwidth_GB_Year']].describe()
```

Out[8]:

	Tenure	Bandwidth_GB_Year
count	10000.000000	10000.000000
mean	34.526188	3392.341550
std	26.443063	2185.294852
min	1.000259	155.506715
25%	7.917694	1236.470827
50%	35.430507	3279.536903
75%	61.479795	5586.141370
max	71.999280	7158.981530

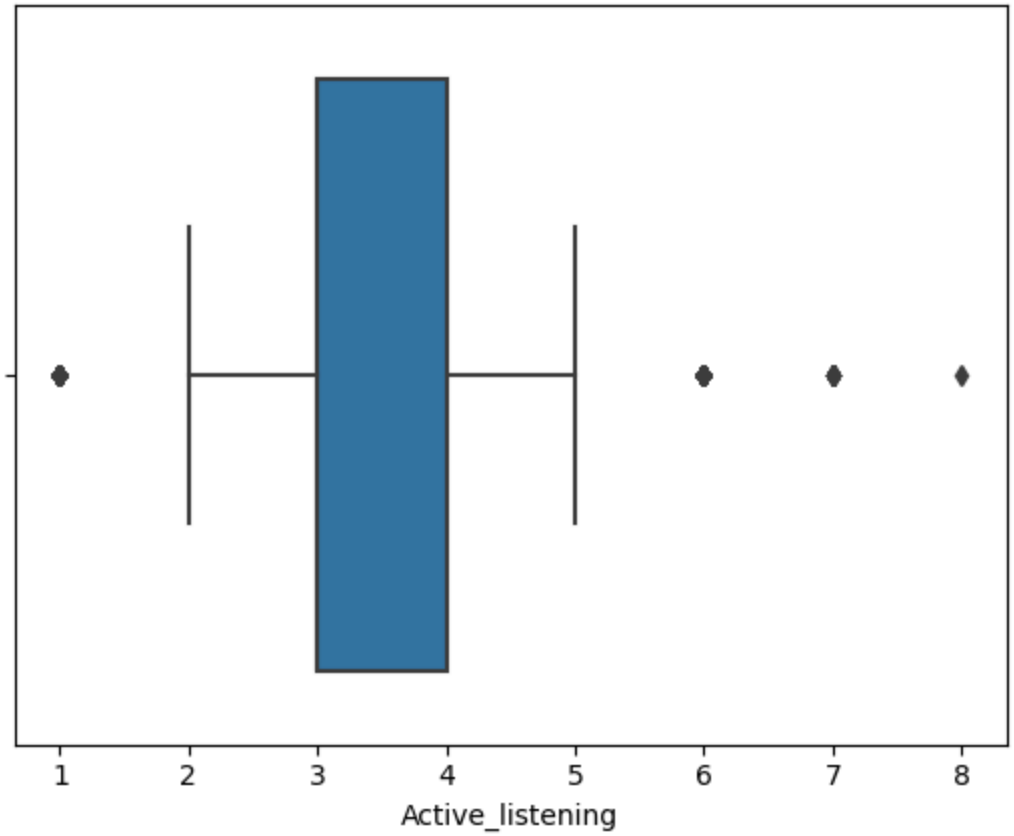
C1. Visual representation of findings

In [9]:
```
# Creating histograms for continuous variables 'Tenure' and 'Bandwidth_GB_Year'
df[['Tenure', 'Bandwidth_GB_Year']].hist(bins=30, figsize=(14, 5))

# Adjusting layout for better spacing
plt.tight_layout()

# Displaying the histograms
plt.show()
```



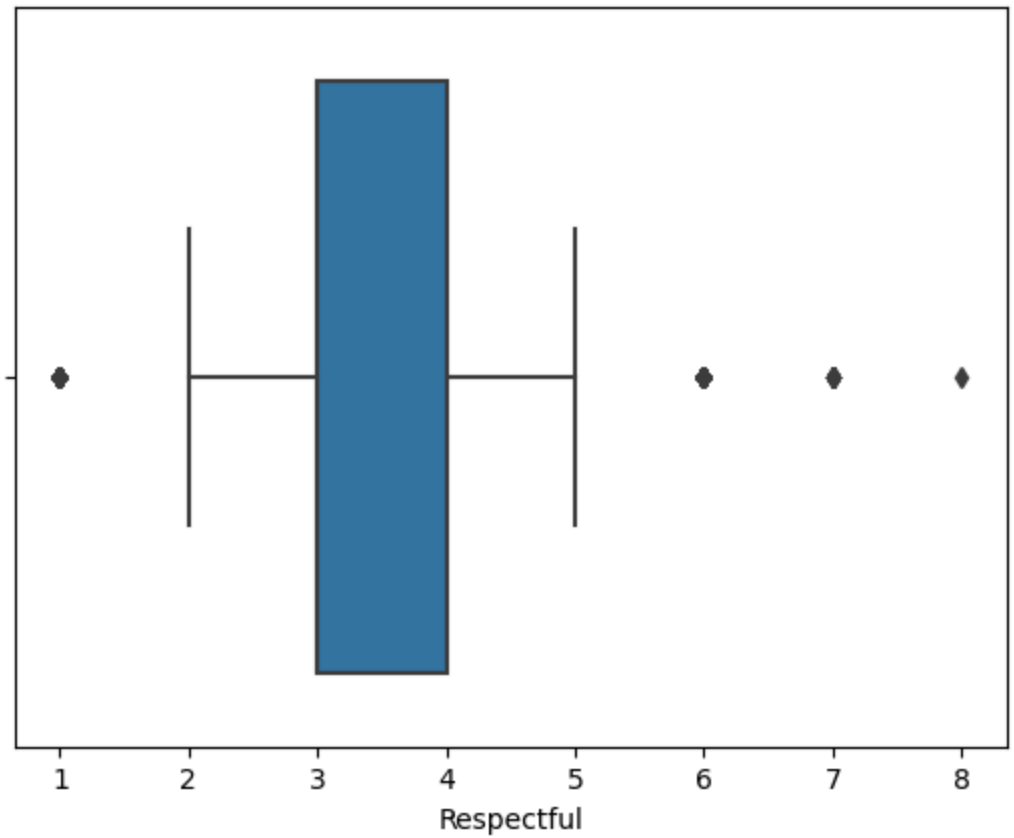
In [10]:
```
#(Waskom, n.d.)
#Creating a boxplot distribution for 'Active_listening'
sns.boxplot(x=df['Active_listening'])

count = df.groupby(['Active_listening']).size().reset_index(name='Count').rename(columns={'Active_listening': 'Rating
Active_listening_min = df['Active_listening'].min()
Active_listening_max = df['Active_listening'].max()
Active_listening_mode = df['Active_listening'].mode()[0]

# Print statistics
print("Active_listening Statistics:")
print(f"Min: {Active_listening_min}")
print(f"Max: {Active_listening_max}")
print(f"Mode: {Active_listening_mode}")
print("---------------")
print("'Active_listening' ratings count:")
print(count)
```

```
Active_listening Statistics:
Min: 1
Max: 8
Mode: 3
---------------
'Active_listening' ratings count:
   Rating  Count
0       1    206
1       2   1378
2       3   3461
3       4   3400
4       5   1335
5       6    205
6       7     14
7       8      1
```



In [11]:
```
#Creating a boxplot distribution for 'Respectful'
sns.boxplot(x=df['Respectful'])

count = df.groupby(['Respectful']).size().reset_index(name='Count').rename(columns={'Respectful': 'Rating'})
respectful_min = df['Respectful'].min()
respectful_max = df['Respectful'].max()
respectful_mode = df['Respectful'].mode()[0]

# Print statistics
print("Respectful Statistics:")
print(f"Min: {respectful_min}")
print(f"Max: {respectful_max}")
print(f"Mode: {respectful_mode}")
print("---------------")
print("'Respectful' ratings count:")
print(count)
```

```
Respectful Statistics:
Min: 1
Max: 8
Mode: 3
---------------
'Respectful' ratings count:
   Rating  Count
0       1    190
1       2   1427
2       3   3445
3       4   3333
4       5   1382
5       6    210
6       7     12
7       8      1
```



D. Bivariate statistics

In [12]:
```
#calculating the pearson correlation coefficient to see if there is a linear relationship between the two.
#(datacamp, n.d.)

pearson = stats.pearsonr(df.Tenure, df.Bandwidth_GB_Year)
print(pearson)
```

```
PearsonRResult(statistic=0.9914951921640127, pvalue=0.0)
```

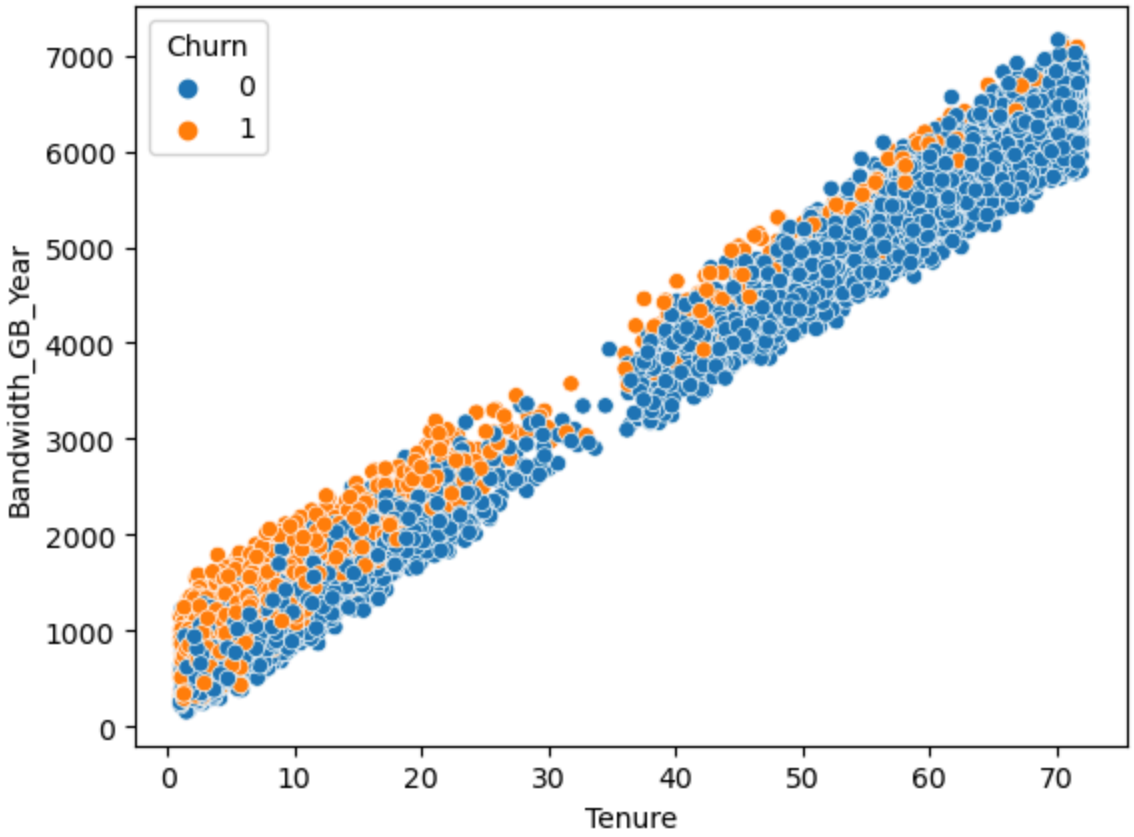
With a p-value of 0.0, we can reject the null hypothesis that there is no linear relationship between Tenure and Bandwidth_GB_Year. The Pearson correlation statistic shows a 0.991 value, indicating a strong positive linear relationship.

The stacked barchart of 'Active_listening' and 'Respectful' shows that each rating holds a similar amount of importance with the customers. We can also see that most customers rated these two factors in the mid to high range of importance with most of the rankings falling between three and four.

D1. Visual representation of findings

In [13]:
```
#Scatter plot of Tenure and Bandwidth_GB_Year displaying the strong linear relationship
sns.scatterplot(data=df, x="Tenure", y="Bandwidth_GB_Year", hue='Churn')
```

Out[13]:
```
<Axes: xlabel='Tenure', ylabel='Bandwidth_GB_Year'>
```



This is a visualization of linear relationship between Tenure and Bandwidth_GB_Year. As you can see there is a strong positve relationship and once again with a p-value of 0 I can say that the linear relationship between these variables is statistically

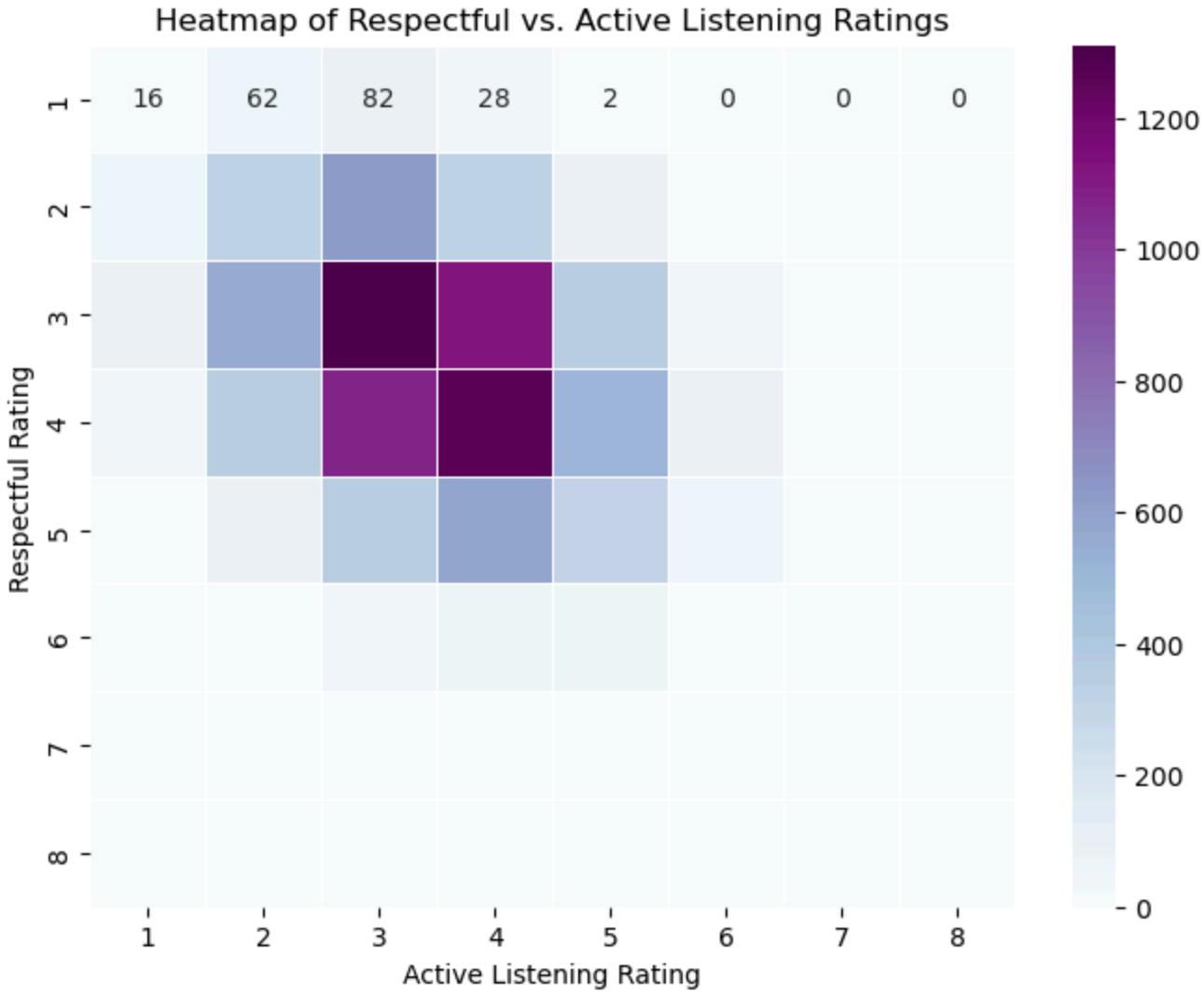
significant.

In [16]:
```
#Creating Visuals and statistics for the categorical variabels 'Respectful' and 'Active_listening'

# Creating a contingency table
contingency_table = pd.crosstab(df['Respectful'], df['Active_listening'])

# Printing the contingency table
print("Contingency Table:")
print(contingency_table)

# Calculating the Chi-square test for independence
chi2, p, dof, expected = chi2_contingency(contingency_table)

# Printing the results
print("Chi-square Test for Independence:")
print(f"Chi2: {chi2}")
print(f"Degrees of Freedom: {dof}")
print(f"P-value: {p} There is a statistically significant association between the two ratings")


# Creating a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(contingency_table, annot=True, fmt='d',linewidth=.5, cmap="BuPu")
plt.title('Heatmap of Respectful vs. Active Listening Ratings')
plt.xlabel('Active Listening Rating')
plt.ylabel('Respectful Rating')
plt.show()
```

Contingency Table:

Active_listening	1	2	3	4	5	6	7	8
Respectful								
1	16	62	82	28	2	0	0	0
2	63	321	619	335	83	6	0	0
3	88	549	1310	1117	345	36	0	0
4	33	360	1072	1262	518	82	6	0
5	6	79	339	579	311	63	5	0
6	0	7	37	75	71	17	2	1
7	0	0	2	4	4	1	1	0
8	0	0	0	0	1	0	0	0

Chi-square Test for Independence:
Chi2: 1123.271706712873
Degrees of Freedom: 49
P-value: 4.121221398308686e-203 There is a statistically significant association between the two ratings



E. Summary of implications

E1. Results of the hypotheses test

The t-test on monthly charges indicated a statistically significant difference between customers who have churned and customers who have not. Therefore, I will reject the null hypothesis that there is no difference between the groups' monthly charges. The

bivariate statistics showed a strong relationship between Tenure and Bandwidth_GB_Year and resulted in a rejection of the null hypothesis that the is not a linear relationship between the two. I also found a statisticaly significant association with the customer's ranking of 'Active listening' and 'Respectful' and therefore rejected the null hypothesis that there is not an association between these two factors.

E2. Limitations of this analysis

The survey responses measure the customer's ranking of the importance of certain factors rather than their experience with customer service. Because of this, the relationships between survey responses and churn are not indicative of a problem that can be addressed. For example, if a customer ranks 'Active_listening' as a 1 (most important) and then the customer churns, this does not indicate that the customer did not experience active listening and therefore churned. The survey responses might be more insightful if they represented the customer's experience with these factors while communicating with the representatives from the company. If customers churned based on recent interactions, then we could make suggestions in terms of implementing training or coaching strategies to improve these customer interactions.

The t-test indicated a statistically significant difference between the average monthly charge for customers who churned and customers who have not. This suggests that monthly charge is associated with churn status. However, without additional statistical tests we cannot determine if monthly charge directly causes churn or if other variables influence the observed difference.

E3. Recommendations based on results

Based on the previous explanations, I would recommend that the survey responses be revised to collect feedback on customer experiences. As stated earlier, making this change will improve customer service and performance.

Because we know that churned customers have a mean monthly charge that is $36.29 more than the customers that do not churn, I recommend that the company makes efforts to find the cause of the price difference and address it by offering incentives for long-term customers, promotional deals for bundling services or even usage-based pricing incentives.

Based on the Pearson correlation test, we found a strong positive relationship between a customer's tenure and bandwidth usage

per year. Because customers are increasingly using more GB per year they will notice an increase in time spent at a task. for example, if someone is using an average of 1GB of data vs someone who is trying to comsume 100GB of data in the same time frame, there will be some performance issues. Therefore, I recommend that the company offer speed tier incentives based on tenure since we know that the GB useage increases as tenure increases.

F Panopto video

The panapto video link is provided in the uploaded documents.

G. Web sources for third party code

- Sewell, W. (n.d.). *D207 webinar EP 3* [Video]. Western Governors University, Panopto. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=58f509ac-47df-4544-9f0a-aeb5016ab8f7
- DataCamp. (n.d.). *Performing experiments in Python: The basics of statistical hypothesis testing*. https://campus.datacamp.com/courses/performing-experiments-in-python/the-basics-of-statistical-hypothesis-testing?ex=9
- Waskom, M. L. (n.d.). seaborn.boxplot. In Seaborn: Statistical data visualization. Retrieved from https://seaborn.pydata.org/generated/seaborn.boxplot.html#seaborn.boxplot

H. Other sources

- Martin, G. (2019, June 10). *Statistics made easy ! ! ! Learn about the t-test, the chi square test, the p value and more* [Video]. YouTube. https://www.youtube.com/watch?v=I10q6fjPxJ0

In [15]:
```
#This is just to quickly show that the code ran successfull without any errors.
print("Code ran without errors!")
```

```
Code ran without errors!
```