Project Purpose

This project will use the data that was previously cleaned in the Jupyter Notebook "Data Cleaning". Some basic data exploration and hypothesis testing will be performed to answer the question "Are customers with a higher average monthly bill more likely to terminate their services?"

Hypothesis Test ¶

- Variable Information

The two variables necessary to perform the analysis are the "Churn Label" variable and "Monthly Charges". The Churn variable is a categorical string data type with two possible values, yes and know, and describes whether or not the customer has terminated their services with the telecom company. The MonthlyCharge variable is numerical in type and the value is the average monthly bill for each customer over the duration of their contract.

- Perform T-test

In [41]:
```
# Import necessary libraries
import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
from scipy.stats import ttest_ind
```

In [42]:
```
# Read in data and view head
df = pd.read_csv('churn_clean_data.csv')
pd.options.display.max_columns = None
df.head()
```

Out[42]:

	CustomerID	Count	Country	State	City	Zip Code	Lat Long	Latitude	Longitude	Gender	Senior Citizen	Partner	Dependents	M
0	3668-QPYBK	1	United States	California	Los Angeles	90003	33.964131, -118.272783	33.964131	-118.272783	Male	No	No	No	
1	9237-HQITU	1	United States	California	Los Angeles	90005	34.059281, -118.30742	34.059281	-118.307420	Female	No	No	Yes	
2	9305-CDSKC	1	United States	California	Los Angeles	90006	34.048013, -118.293953	34.048013	-118.293953	Female	No	No	Yes	
3	7892-POOKP	1	United States	California	Los Angeles	90010	34.062125, -118.315709	34.062125	-118.315709	Female	No	Yes	Yes	
4	0280-XJGEX	1	United States	California	Los Angeles	90015	34.039224, -118.266293	34.039224	-118.266293	Male	No	No	Yes	

In [43]:
```
# Separate MonthlyCharge into separate variables based on whether or not there was customer churn
y = df.loc[df['Churn Label'] == 'Yes','Monthly Charges']
n = df.loc[df['Churn Label'] == 'No','Monthly Charges']
ttest = (ttest_ind(y,n))
```

In [44]:
```
# Print output of T-test
print("The results for the T-test are: ", ttest)
```

```
The results for the T-test are:  Ttest_indResult(statistic=16.53673801593631, pvalue=2.706645606888261e-60)
```

- T-Test Justification and Summary

The choice of what type of statistical test to choose came down to the variables that I chose to analyze. The variable "Churn Label" is a categorical variable with two possible values: yes or no. The variable "Monthly Charges" contains numeric values based on the average amount the customer is billed each quarter. Chi-square tests are appropriate when you are looking at multiple categorical variables (ex. churn label and gender) (Flom, 2019). ANOVA is used when the categorical variable has three or more possible values (Chauhan, 2020). A T-test is appropriate when looking at two variables in which one is categorical with exactly 2 values and the other numerical (Chauhan, 2020). Since the Churn Label has values of yes and no and the monthly charge variable is numerical, the T-test was used.

For this test, the hypothesis is as follows:

- H_0: Customers with a higher average monthly bill are not more likely to terminate their services

The alternate hypothesis therefore is:

- H_1: Customers with a higher average monthly bill are more likely to terminate their services

The p-value is quite large at 2.7. A typical significance level is 0.05, and the p-value is much higher than this level of significance. Therefore, we do not have enough information to reject the null hypothesis.

Univariate Statistics

In [45]:
```
# Create a subset dataframe of the categorical and continuous variables for
#    univariate and bivariate statistics and display head
df1 = df[['Churn Label', 'Contract', 'Monthly Charges', 'Tenure Months']] # (ref 1)
df1.head()
```

Out[45]:

	Churn Label	Contract	Monthly Charges	Tenure Months
0	Yes	Month-to-month	53.85	2.0
1	Yes	Month-to-month	70.70	2.0
2	Yes	Month-to-month	99.65	8.0
3	Yes	Month-to-month	104.80	28.0
4	Yes	Month-to-month	103.70	49.0

In [46]:
```
# View statistical description of continuous variables
df1.describe()
```

Out[46]:

	Monthly Charges	Tenure Months
count	7043.000000	7043.000000
mean	64.761692	32.421707
std	30.090047	24.526082
min	18.250000	1.000000
25%	35.500000	9.000000
50%	70.350000	29.000000
75%	89.850000	55.000000
max	118.750000	72.000000

The above calculations give us the following information about the chosen continuous variables:

Monthly Charges:

- The data ranges from $18.25 to \$118.75 in monthly charges per month
- The mean amount paid is $64.76 per month
- The median amount paid per month is $70.35
- The standard deviation for the MonthlyCharge variable is $30.09
- The interquartile range for this variable is between $35.50 and \$89.85
- There are 7,043 total values for this variable

Tenure Months:

- The data ranges from 1.0 months to 72 months in tenure
- The mean length for outages per week is 32.4 months
- The median length for outages per week is 29.0 months
- The standard deviation for the tenure is 24.5
- The interquartile range for this variable is between 35.5 months and 55.0 months
- There are 7,043 total values for this variable

In [47]:
```
# Calculate the total number of values present for each categorical variable
for col in ['Churn Label', 'Contract']:
    print(col, ': ', df1[col].count())
```

```
Churn Label :  7043
Contract :  7043
```

In [51]:
```
# View counts of categorical variables
for col in ['Churn Label', 'Contract']:
  print(df1[col].value_counts())
```

```
No     5174
Yes    1869
Name: Churn Label, dtype: int64
Month-to-month    3875
Two year          1695
One year          1473
Name: Contract, dtype: int64
```

In [52]:
```
# View percentages for each categorical value
for col in ['Churn Label', 'Contract']:
  print(df1[col].value_counts(normalize=True)) # (ref 2)
```

```
No     0.73463
Yes    0.26537
Name: Churn Label, dtype: float64
Month-to-month    0.550192
Two year          0.240664
One year          0.209144
Name: Contract, dtype: float64
```

The distribution of the categorical variables is as follows:

Churn Label

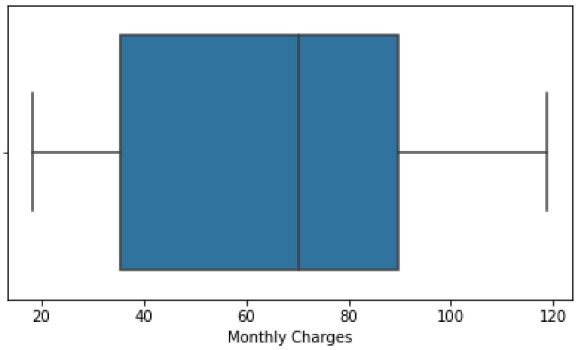
- There are 7,043 total values present in this column
- Of the 7,043 values, 5,174 customers did not terminate their contract while 1,869 customers terminated
- 73.5% of customers did not terminate their contract, while 26.5% of customers did

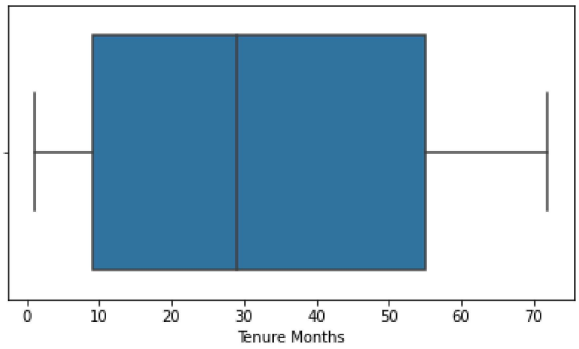
Contract

- There are 7,043 total values present in this column
- Of the 7,043 values: 3,875 customers are on a month-to-month contract, 1,695 are on a two-year contract, and 1,473 are on a one-year contract
- 55.0% of customers are on a month-to-month contract, 24.1% are on a two-year contract, and 20.9 are on a one-year contract

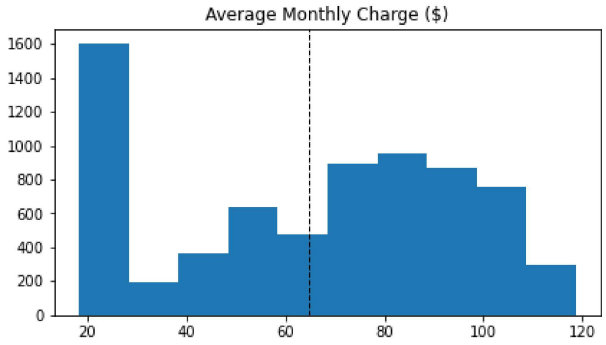
Univariate Visual Findings

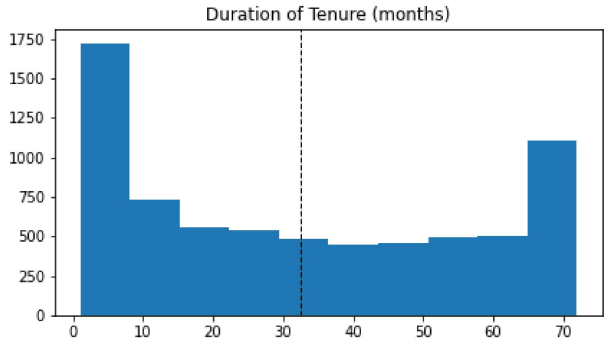
In [53]:
```
# View boxplots of continuous variables to view median and interquartile ranges
figure, axes = plt.subplots(nrows=1, ncols=2, figsize=(15,8))
plt.subplot(2, 2, 1)
sns.boxplot(x=df1['Monthly Charges'])
plt.subplot(2, 2, 2)
sns.boxplot(x=df1['Tenure Months'])

plt.show();
```



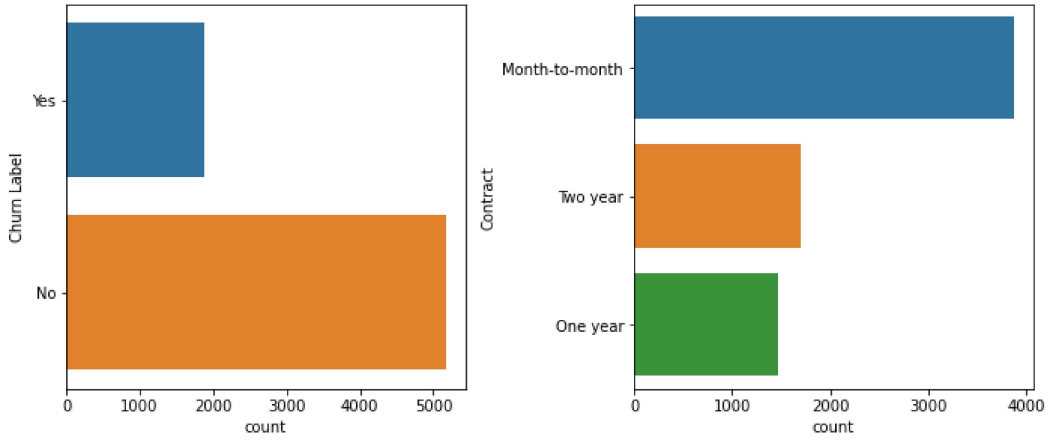


In [54]:
```
# Plot histograms of continuous variables including dashed line displaying mean (ref G3)
figure, axes = plt.subplots(nrows=1, ncols=2, figsize=(15,8))
plt.subplot(2, 2, 1)
plt.hist(df1['Monthly Charges'])
plt.axvline(df['Monthly Charges'].mean(), color = 'black', linestyle = 'dashed', linewidth = 1)
plt.title('Average Monthly Charge ($)')
plt.subplot(2, 2, 2)
plt.hist(df1['Tenure Months'])
plt.axvline(df['Tenure Months'].mean(), color = 'black', linestyle = 'dashed', linewidth = 1)
plt.title('Duration of Tenure (months)')
plt.show();
```

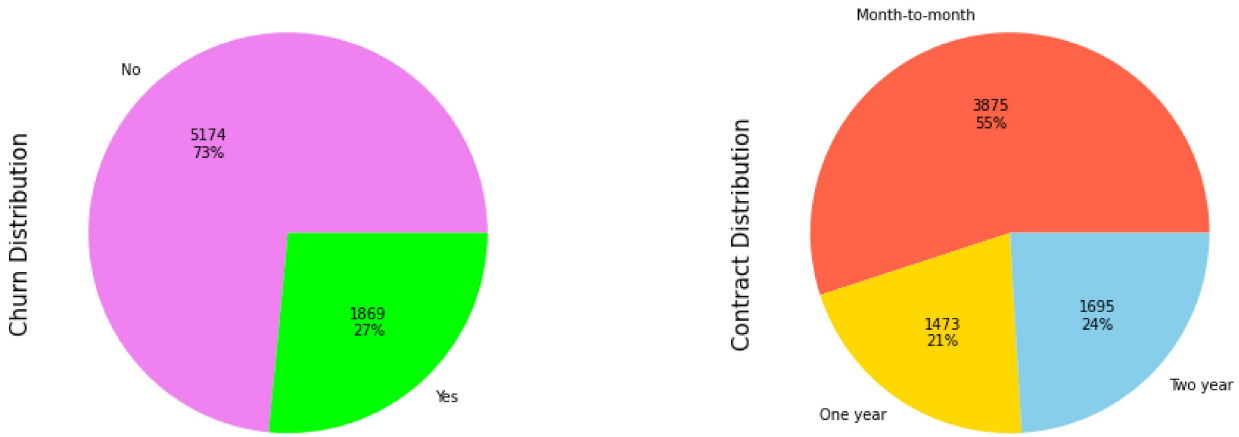




In [55]:
```
# Display bar plots for categorical variables
figure, axes = plt.subplots(nrows=1, ncols=2, figsize=(15,8))
plt.subplot(2, 3, 1)
sns.countplot(data = df, y = 'Churn Label')
plt.subplot(2, 3, 2)
sns.countplot(data = df, y = 'Contract')
figure.tight_layout()
plt.show();
```



In [56]:
```
# Display pie chart of categorical variable distributions (ref G4)
def label_function(val):
    return f'{val / 100 * len(df):.0f}\n{val:.0f}%'

fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(15, 5))

df1.groupby('Churn Label').size().plot(kind ='pie', autopct=label_function, textprops = {'fontsize': 10},
                                       colors = ['violet', 'lime'], ax = ax1)
df1.groupby('Contract').size().plot(kind = 'pie', autopct=label_function, textprops = {'fontsize': 10},
                                    colors = ['tomato', 'gold', 'skyblue'], ax = ax2)
ax1.set_ylabel('Churn Distribution', size = 15)
ax2.set_ylabel('Contract Distribution', size = 15)
plt.tight_layout()
plt.show()
```



Bivariate Statistics

- Two continuous variables

In [57]:
```
# Calculate correlation matrix of continuous variables
data = df1.corr()
print(data)
```

```
                  Monthly Charges  Tenure Months
Monthly Charges          1.000000       0.246638
Tenure Months            0.246638       1.000000
```

- The correlation matrix shows a weak relationship between the average monthly bill and months of tenure

- One continuous one categorical

In [58]:
```
# Calculate distribution of continuous variables when Churn is yes
churn_y = df1[(df1['Churn Label'] == 'Yes')]
churn_y.describe()
```

Out[58]:

	Monthly Charges	Tenure Months
count	1869.000000	1869.000000
mean	74.441332	17.979133
std	24.666053	19.531123
min	18.850000	1.000000
25%	56.150000	2.000000
50%	79.650000	10.000000
75%	94.200000	29.000000
max	118.350000	72.000000

The descriptive statistics for the customers who terminate services is as follows:

Monthly Charges:

- The data ranges from $18.85 to \$118.35 in monthly charges per month
- The mean amount paid is $74.44 per month
- The median amount paid per month is $79.65
- The standard deviation for the MonthlyCharge variable is $24.67
- The interquartile range for this variable is between $56.15 and \$94.20
- There are 1,869 total values for this variable

Tenure Months:

- The data ranges from 1.0 months to 72.0 months tenure
- The mean length for outages per week is 17.98 months
- The median length for outages per week is 10.0 months
- The standard deviation for the Tenure Months variable is 19.5
- The interquartile range for this variable is between 2.0 months and 29.0 months
- There are 1,869 total values for this variable

In [59]:
```
# Calculate distribution of continuous variables when Churn is no
churn_n = df1[(df1['Churn Label'] == 'No')]
churn_n.describe()
```

Out[59]:

	Monthly Charges	Tenure Months
count	5174.000000	5174.000000
mean	61.265124	37.638787
std	31.092648	24.052557
min	18.250000	1.000000
25%	25.100000	15.000000
50%	64.425000	38.000000
75%	88.400000	61.000000
max	118.750000	72.000000

The descriptive statistics for the customers who do not terminate services is as follows:

Monthly Charges:

- The data ranges from $18.25 to \$118.75 in monthly charges per month
- The mean amount paid is $61.27 per month
- The median amount paid per month is $64.43
- The standard deviation for the MonthlyCharge variable is $31.09
- The interquartile range for this variable is between $25.10 and \$88.40
- There are 5,174 total values for this variable

Tenure Months:

- The data ranges from 1.0 months to 72.0 months in tenure
- The mean length for outages per week is 37.6 months
- The median length for outages per week is 38.0 months
- The standard deviation for the Tenure Months variable is 24.1
- The interquartile range for this variable is between 15.0 months and 61.0 months
- There are 5,174 total values for this variable

In [60]:
```
# Calculate distribution of continuous variables when Contract is month-to-month
contract_m = df1[(df1.Contract == 'Month-to-month')]
contract_m.describe()
```

Out[60]:

	Monthly Charges	Tenure Months
count	3875.000000	3875.000000
mean	66.398490	18.036645
std	26.926599	17.689054
min	18.750000	1.000000
25%	45.850000	3.000000
50%	73.250000	12.000000
75%	88.875000	29.000000
max	117.450000	72.000000

The descriptive statistics for the customers who are on a month-to-month contract is as follows:

Monthly Charges:

- The data ranges from $18.75 to \$117.45 in monthly charges per month
- The mean amount paid is $66.40 per month
- The median amount paid per month is $73.25
- The standard deviation for the MonthlyCharge variable is $26.92
- The interquartile range for this variable is between $45.85 and \$88.88
- There are 3,875 total values for this variable

Tenure Months:

- The data ranges from 1.0 months to 72.0 months in tenure
- The mean length for outages per week is 18.0 months
- The median length for outages per week is 12.0 months
- The standard deviation for the Tenure Months variable is 17.7
- The interquartile range for this variable is between 3.0 months and 29.0 months
- There are 3,875 total values for this variable

In [61]:
```
# Calculate distribution of continuous variables when Contract is one year
contract_1 = df1[(df1.Contract == 'One year')]
contract_1.describe()
```

Out[61]:

	Monthly Charges	Tenure Months
count	1473.000000	1473.000000
mean	65.048608	42.066783
std	31.840539	19.005973
min	18.250000	1.000000
25%	26.900000	27.000000
50%	68.750000	44.000000
75%	94.800000	58.000000
max	118.600000	72.000000

The descriptive statistics for the customers who are on a month-to-month contract is as follows:

Monthly Charges:

- The data ranges from $18.25 to \$118.60 in monthly charges per month
- The mean amount paid is $65.05 per month
- The median amount paid per month is $68.75
- The standard deviation for the MonthlyCharge variable is $31.84
- The interquartile range for this variable is between $26.90 and \$94.80
- There are 1,473 total values for this variable

Tenure Months:

- The data ranges from 1.0 months to 72.0 months in tenure
- The mean length for outages per week is 19.0 months
- The median length for outages per week is 44.0 months
- The standard deviation for the Tenure Months variable is 19.0
- The interquartile range for this variable is between 27.0 months and 58.0 months
- There are 1,473 total values for this variable

In [62]:
```
# Calculate distribution of continuous variables when Contract is two years
contract_2 = df1[(df1.Contract == 'Two year')]
contract_2.describe()
```

Out[62]:

	Monthly Charges	Tenure Months
count	1695.000000	1695.000000
mean	60.770413	56.926083
std	34.678865	17.777710
min	18.400000	1.000000
25%	24.025000	48.000000
50%	64.350000	64.000000
75%	90.450000	71.000000
max	118.750000	72.000000

The descriptive statistics for the customers who are on a month-to-month contract is as follows:

Monthly Charges:

- The data ranges from $18.40 to \$118.75 in monthly charges per month
- The mean amount paid is $60.77 per month
- The median amount paid per month is $64.35
- The standard deviation for the MonthlyCharge variable is $34.68
- The interquartile range for this variable is between $24.03 and \$90.45
- There are 1,695 total values for this variable

Tenure Months:

- The data ranges from 1.0 months to 72.0 months in tenure
- The mean length for outages per week is 56.9 months
- The median length for outages per week is 64.0 months
- The standard deviation for the Tenure Months variable is 17.8
- The interquartile range for this variable is between 48.0 months and 71.0 months
- There are 1,695 total values for this variable

- Two categorical

In [63]:
```
# Create table table for Churn Label by Contract (ref 5)
churn_table = pd.crosstab(index = df1['Contract'], columns = df1['Churn Label'])
churn_table
```

Out[63]:

Churn Label	No	Yes
Contract		
Month-to-month	2220	1655
One year	1307	166
Two year	1647	48

- For customers on a month-to-month contract, 1,655 chose to end their contract while 2,220 did not
- For customers on a one year contract, 166 chose to end their contract while 1,307 did not
- For customers on a two year contract, 48 chose to end their contract while 1,647 did not

In [64]:
```
# Create table for Contract by Churn Label (ref 5)
crosstab_table = pd.crosstab(index = df1['Churn Label'], columns = df1['Contract'])
crosstab_table
```

Out[64]:

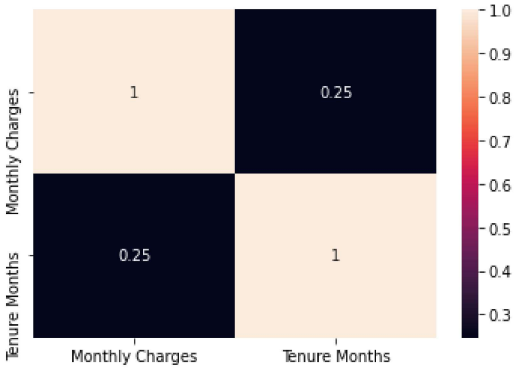
Contract	Month-to-month	One year	Two year
Churn Label			
No	2220	1307	1647
Yes	1655	166	48

- Of customers who ended their contract, 1,655 were on a month-to-month contract, 166 were on a one year contract, and 48 were on a two year contract
- Of customers who did not end their contract, 2,220 were on a month-to-month contract, 1,307 were on a one year contract, and 1,647 were on a two year contract

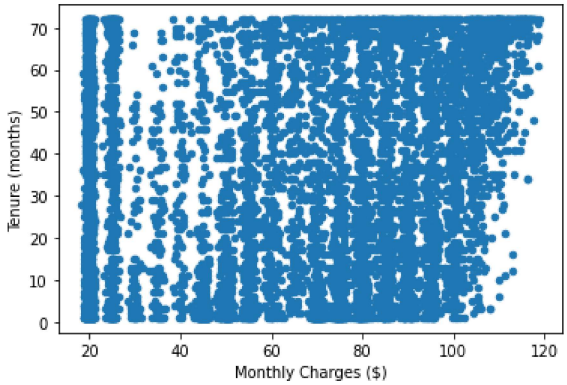
Bivariate Visual Findings

- Two continuous variables

In [65]:
```
# Display a heatmap showing correlation between continuous variables
sns.heatmap(df1.corr(), annot = True)
plt.show()
```

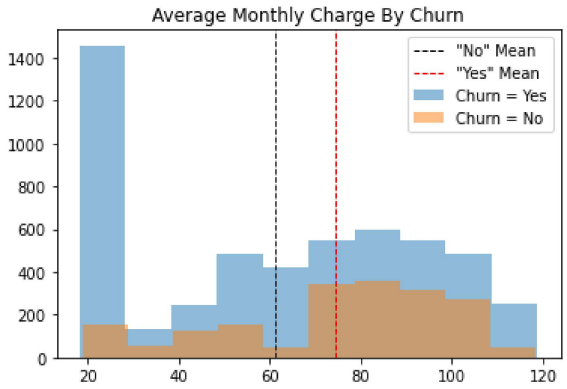


In [66]:
```
# Display scatterplot distribution between continuous variables
ax = df1.plot.scatter(x = 'Monthly Charges', y = 'Tenure Months')
ax.set_xlabel('Monthly Charges ($)')
ax.set_ylabel('Tenure (months)')
plt.show()
```

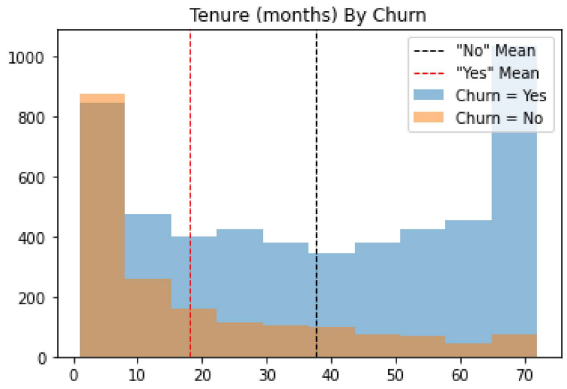


- One continuous one categorical

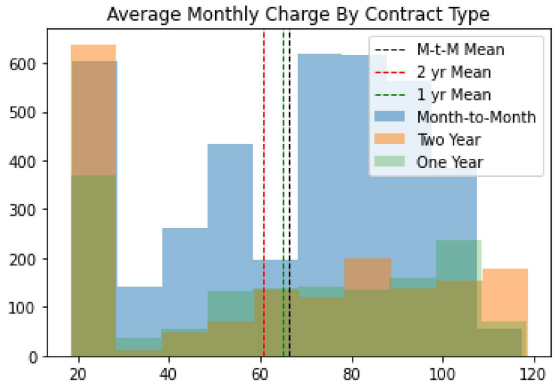
In [67]:
```
# Plot histogram of the Monthly Charges by Churn Label (ref 6)
plt.hist(churn_n['Monthly Charges'], label = 'Churn = Yes', alpha=0.5)
plt.axvline(churn_n['Monthly Charges'].mean(), color = 'black', linestyle = 'dashed', linewidth = 1, label =
'"No" Mean')
plt.hist(churn_y['Monthly Charges'], label = 'Churn = No', alpha=0.5)
plt.axvline(churn_y['Monthly Charges'].mean(), color = 'red', linestyle = 'dashed', linewidth = 1, label ='"Ye
s" Mean')
plt.legend()
plt.title('Average Monthly Charge By Churn');
```



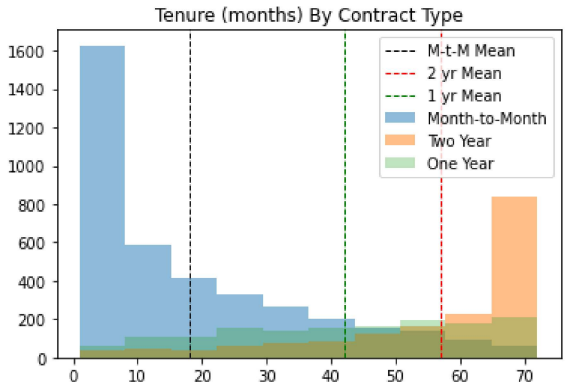
In [68]:
```
# Plot histogram of the Tenure Months by Churn Label (ref 6)
plt.hist(churn_n['Tenure Months'], label = 'Churn = Yes', alpha=0.5)
plt.axvline(churn_n['Tenure Months'].mean(), color = 'black', linestyle = 'dashed', linewidth = 1, label ='"N
o" Mean')
plt.hist(churn_y['Tenure Months'], label = 'Churn = No', alpha=0.5)
plt.axvline(churn_y['Tenure Months'].mean(), color = 'red', linestyle = 'dashed', linewidth = 1, label ='"Yes"
Mean')
plt.legend()
plt.title('Tenure (months) By Churn');
```



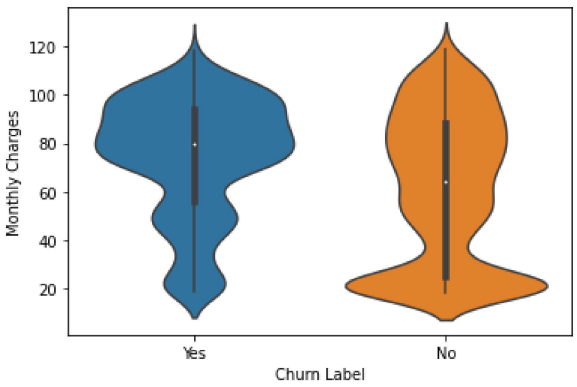
In [69]:
```
# Plot histogram of the Monthly sCharge by Contract (ref 6)
plt.hist(contract_m['Monthly Charges'], label = 'Month-to-Month', alpha=0.5)
plt.axvline(contract_m['Monthly Charges'].mean(), color = 'black', linestyle = 'dashed', linewidth = 1, label
='M-t-M Mean')
plt.hist(contract_2['Monthly Charges'], label = 'Two Year', alpha=0.5)
plt.axvline(contract_2['Monthly Charges'].mean(), color = 'red', linestyle = 'dashed', linewidth = 1, label =
'2 yr Mean')
plt.hist(contract_1['Monthly Charges'], label = 'One Year', alpha=0.3)
plt.axvline(contract_1['Monthly Charges'].mean(), color = 'green', linestyle = 'dashed', linewidth = 1, label
='1 yr Mean')
plt.legend()
plt.title('Average Monthly Charge By Contract Type');
```



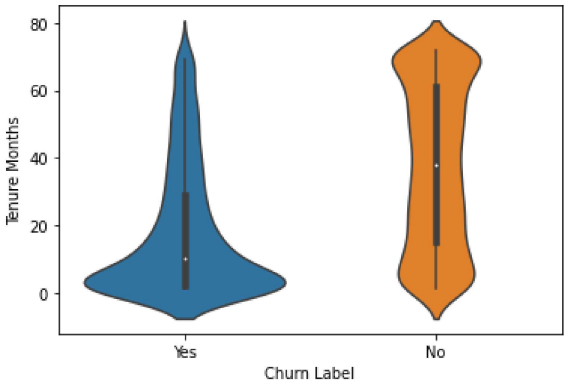
In [70]:
```
# Plot histogram of the Tenure Months by Contract (ref 6)
plt.hist(contract_m['Tenure Months'], label = 'Month-to-Month', alpha=0.5)
plt.axvline(contract_m['Tenure Months'].mean(), color = 'black', linestyle = 'dashed', linewidth = 1, label =
'M-t-M Mean')
plt.hist(contract_2['Tenure Months'], label = 'Two Year', alpha=0.5)
plt.axvline(contract_2['Tenure Months'].mean(), color = 'red', linestyle = 'dashed', linewidth = 1, label ='2
 yr Mean')
plt.hist(contract_1['Tenure Months'], label = 'One Year', alpha=0.3)
plt.axvline(contract_1['Tenure Months'].mean(), color = 'green', linestyle = 'dashed', linewidth = 1, label =
'1 yr Mean')
plt.legend()
plt.title('Tenure (months) By Contract Type');
```



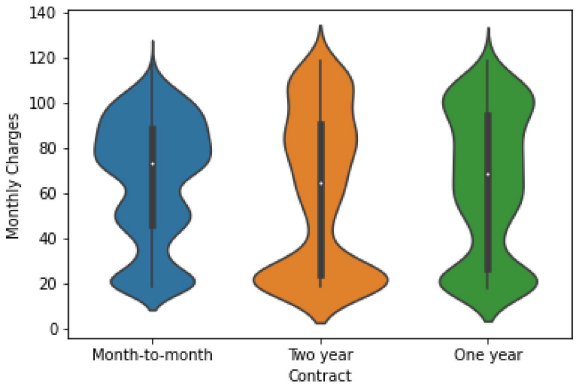
In [71]:
```
# Plot Monthly Charges by Churn Label (ref 7)
ax = sns.violinplot(x= 'Churn Label', y = 'Monthly Charges', data=df1)
```



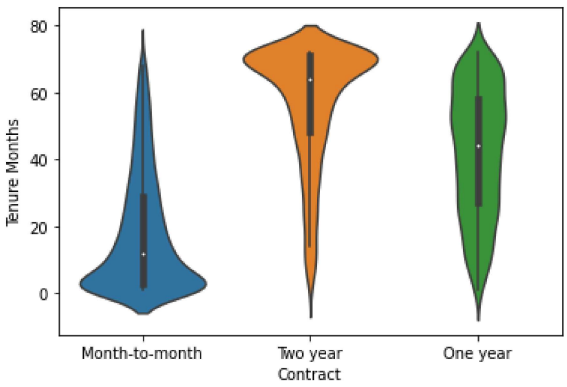
In [72]:
```
# Plot Tenure Months by Churn Label (ref G7)
ax = sns.violinplot(x= 'Churn Label', y = 'Tenure Months', data = df1)
```



In [73]:
```
# Plot MonthlyCharge by Contract (ref 7)
ax = sns.violinplot(x= 'Contract', y = 'Monthly Charges', data=df1)
```

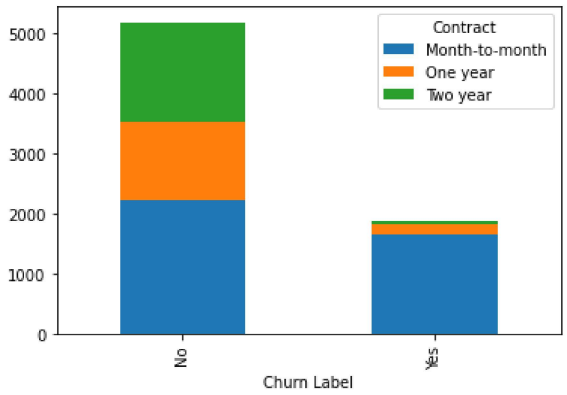


In [74]:
```
# Plot Tenure Months by Contract (ref 7)
ax = sns.violinplot(x= 'Contract', y = 'Tenure Months', data=df1)
```

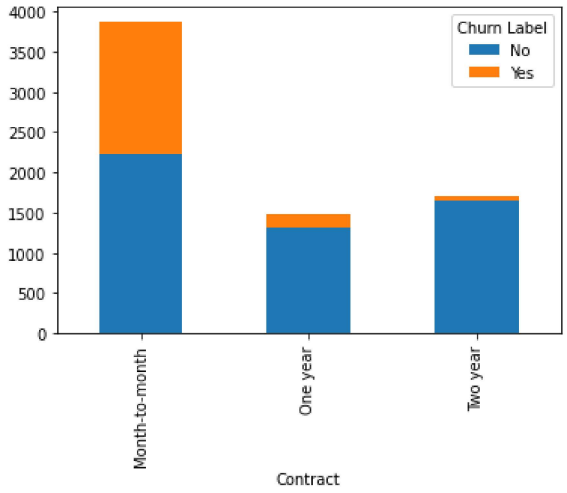


- Two categorical

In [75]:
```
# Visualize Churn Label by Contract
tbl = pd.pivot_table(df1.groupby(['Churn Label','Contract']).size().reset_index(),
values=0,
index='Churn Label',
columns=['Contract'],
aggfunc=np.sum)
tbl.plot(kind='bar',stacked=True);
```



In [76]:
```
# Visualize Contract by Churn Label by Contract
tbl = pd.pivot_table(df1.groupby(['Churn Label','Contract']).size().reset_index(),
values=0,
index='Contract',
columns=['Churn Label'],
aggfunc=np.sum)
tbl.plot(kind='bar',stacked=True);
```



Results of Analysis

The t-test provided a p-value of 2.7. Such a large p-value is not enough evidence to reject the null hypothesis, meaning that there is no correlation between the average amount that a customer is billed and their likelihood to terminate services. Further evidence of this is shown in the bivariate statistics section above where we see that the mean and median average amount that customers pay per month is simalar for customers who choose to terminate services compared to those who do not.

Limitations of Analysis

This analysis is limited to the two variables that I chose, the Churn Labels and Monthly Charges. Other factors could play in to a customer's decision to terminate their services with the telecommunications company. Also, the analysis might have benefited from random sampling. I chose not to sample, but sampling the data could have minimalized possible bias and provided more data quality (Bruce, 2020).

Recommended Course of Action

Based on the results of the analysis, the stakeholders should consider pricing models that would foster a sense of customer loyalty. In my experience, many providers offer great rates for new customers to join their service. Often, once the rates expire, they increase tremendously. This can cause customers to jump to a new service as soon as their contract ends to obtain better rates and benefits. By offering competitive pricing to long term customers, customers might opt to stick with the provider even if it costs a bit more to avoid the hassle of changing providers. I would recommend that some sort of long term A/B testing be done to see if a new model retains more customers and provides higher overall revenue.

Helpful Coding References

1. https://pandas.pydata.org/pandas-docs/stable/getting_started/intro_tutorials/03_subset_data.html (https://pandas.pydata.org/pandas-docs/stable/getting_started/intro_tutorials/03_subset_data.html)
2. https://stackoverflow.com/questions/14281871/given-a-pandas-series-that-represents-frequencies-of-a-value-how-can-i-turn-tho (https://stackoverflow.com/questions/14281871/given-a-pandas-series-that-represents-frequencies-of-a-value-how-can-i-turn-tho)
3. https://stackoverflow.com/questions/51590602/histogram-show-values-of-vertical-lines-in-legends (https://stackoverflow.com/questions/51590602/histogram-show-values-of-vertical-lines-in-legends)
4. https://stackoverflow.com/questions/63687789/how-do-i-create-a-pie-chart-using-categorical-data-in-matplotlib (https://stackoverflow.com/questions/63687789/how-do-i-create-a-pie-chart-using-categorical-data-in-matplotlib)
5. https://adataanalyst.com/data-analysis-resources/visualise-categorical-variables-in-python/ (https://adataanalyst.com/data-analysis-resources/visualise-categorical-variables-in-python/)
6. https://www.kite.com/python/answers/how-to-plot-two-histograms-on-one-plot-using-matplotlib-in-python (https://www.kite.com/python/answers/how-to-plot-two-histograms-on-one-plot-using-matplotlib-in-python)
7. https://seaborn.pydata.org/generated/seaborn.violinplot.html (https://seaborn.pydata.org/generated/seaborn.violinplot.html)

Sources

Bruce, P. A. (2020). Practical Statistics for Data Scientists, 50 Essential Concepts Using R and Python. Sebastopol, CA: O'Reilly Media, Incorporated. ISBN: 978-1492072942

Chauhan, N. (2020, May 04). P Value, T test, Chi Square test, ANOVA, When to use Which Strategy? Retrieved December 14, 2020, from https://medium.com/datadriveninvestor/p-value-t-test-chi-square-test-anova-when-to-use-which-strategy-32907734aa0e (https://medium.com/datadriveninvestor/p-value-t-test-chi-square-test-anova-when-to-use-which-strategy-32907734aa0e)

Flom, P. (2019, March 02). The Difference Between a T-Test & a Chi Square. Retrieved December 14, 2020, from https://sciencing.com/difference-between-ttest-chi-square-8225095.html (https://sciencing.com/difference-between-ttest-chi-square-8225095.html)