Part 1

A1.
Research question: What is the combined effect of a customer's geographic location, service outages, and monthly charges on their overall tenure with telecommunications companies?

A2.
The goal of my analysis is to determine how customer's factors influence their tenure. Utilizing multiple linear regression should permit this investigation, and the ability to provide recommendations to the telecommunication companies.

Part 2

B1.
Multiple linear regression can only be applied if all technical conditions are met. These assumptions are critical for the accuracy and reliability of the model's parameter estimates and predictions. The four assumptions of a multiple linear regression model include:

1. Linearity: The relationship between the independent and dependent variables follows a linear model. The changes in the dependent variables are proportional to changes in the independent variables.
2. Independence: The observations in the analysis are independent of each other, meaning the values of each dependent variable do not influence observations.
3. Normality: The differences between the observed values and the predicted values, residuals, are normally distributed. This condition ensures that the p-value and confidence intervals estimates are an accurate reflection of the population values.
4. Equal variability: The spread of the residuals should be approximately the same for all values of the independent variables.

B2.
Python is the selected programming language because the data set did not require detailed visualizations over periods of time, and it did not require intense statistical technique to clean data. Python is a much more general, robust program that does not require specific libraries and syntax, which makes it easier and more time effective to achieve our goal of conducting multiple linear regression. Furthermore, pandas, numpy, missingno, seaborn, matplotlib, scikitlearn, fancy impute, and statsmodels were all used to conduct various phases of the analysis. The various phases of the analysis for this report included data cleaning, data transformation, feature selection, model construction, and statistical calculations. In comparison to R, I would have had to use specific syntax for each library installed, and this could have unnecessarily increased the complexity of my investigation.

B3.
Multiple linear regression technique allows you to model the relationship between the continuous dependent variable and multiple independent variables or predictors simultaneously, which allows us to better understand real world problems that involve how multiple factors jointly affect a decision. This is the appropriate technique for analyzing the research question because the target variable, the customer's tenure, is a continuous variable. Also, it allows me to investigate how multiple independent variables such as customer's geographic location, service outages, and monthly charges influence the continuous dependent variable, the customer's tenure.

Jasmine Cooper
D208
Linear Regression Modeling
October 1, 2023
Western Governor's University

Multiple linear regression provides quantitative insights into the strength and direction of the relationship between the independent variables and the dependent variable. This helps to estimate the coefficients of the independent variables to indicate the magnitude of their impact on the dependent variable, customer tenure. Also, this technique assumes a linear relationship between the independent variables and the dependent variable. This aligns with the idea that changes in predictor or independent variables should have a linear effect on the dependent variable, customer tenure. Additionally, multiple linear regression offers the ability to assess the statistical significance of the independent variables to determine which factors are most influential in predicting customer tenure.

In conclusion, multiple linear regression is the appropriate technique for analyzing the research question because it accommodates the continuous dependent variable and allows for the investigation of multiple independent variables at once, provides insights into the relationships, and offers statistical tests to assess the significance of these relationships.

In text citations:
("Technical conditions for linear regression", n.d.)

Part 3

C1.
The data cleaning process entails investigating missing and duplicate values to decide how to impute missing values. The goal of this process is to gain insight into missing variables' values for effective data imputation. Clean data is critical for preparing the dataset to conduct multiple linear regression and provide meaningful recommendations to answer the research question.

Data Cleaning Plan:

Step 1: Exploratory Data Analysis

1. Loaded CSV file into Jupyter notebook using pandas library and converted the CSV file into a Data Frame.
2. Used the methods, 'describe()', 'info()', and 'dtypes()' to observe the summary statistics, data types, and nullity, the data types of the data frame.
3. Identified if there was duplicated information using the 'duplicated()' method.
4. Visualized missing data across the entire data frame using missingno and matplotlib libraries, using the 'bar()' and 'show()' methods.
5. Created a correlation matrix using the 'corr()' method, and utilized the correlation matrix to create a heatmap using the seaborn library to understand the correlation between variables missing values.
6. Determined missingness type(s) if applicable.

Step 2: Clean Data

1. Upon completion of investigation, the variables below had missing values:
    a. Children- 7505 (numerical)
    b. Income- 7510 (numerical)
    c. Techie- 7523 (non-numerical)

d. Age- 7525 (numerical)
e. Phone- 8974 (non-numerical)
f. Bandwidth GB Year- 8979 (numerical)
g. Tech Support- 9009 (non-numerical)
h. Tenure- 9069 (numerical)

2. By analyzing the summary statistics, I noticed that the population had a minimum value of zero. Since it is highly unlikely population is zero, I replaced all population records with values equal to zero with NaN values utilizing the numpy library and the 'nan' and 'isnan' methods. This allows Python to recognize missing values properly prior to imputation. The missing Population values were imputed using the K-Nearest Neighbor technique.
3. The numerical values appear to be missing at random because the missing values could probably be predicted by other observed values in the data set. For example, missing number of children values could be relative to age and/or marital status (e.g., if a customer is an unmarried young adult, they are likely to be within the age range 18-30, and not list the number of children that they have since it might not be applicable). To treat numerical missing values, I decided to use the K-Nearest Neighbor imputation technique because missing values can likely be predicted based on similar observed variables and values.
4. The non-numerical values also appear to be missing at random because the missing values could also probably be predicted by other observed values in the data set. For example, missing Techie values could also be relative to age (e.g., if a customer is between ages 50-90, it is unlikely that they would be considered a Techie). Likewise, missing values for Phone and Tech Support could likely be related to age due to the fast advancement of technology overtime. To impute non-numerical variables' values, I decided to use the Simple Imputation mode technique, and used frequency to fill in missing data. Furthermore, to validate this imputation technique, I analyzed the clean data to ensure that it fit the well-considered estimate, and the clean data portrayed that the group that are not techie's lies between ages 50-90.

Code for C1:
See code/script attached and below:

Please see Churn Data.ipynb attached for this code.

```
import pandas as pd
import numpy as np
import missingno as msno
churn_raw_data = pd.read_csv ('/Users/jasminemoniquecooper/Downloads/churn_raw_data.csv')
pd.set_option('display.max_columns', None)
churn_raw_data.head(10)

churn_raw_data.dtypes

churn_raw_data.describe()

churn_duplicates = churn_raw_data.duplicated()
churn_raw_data[churn_duplicates]

churn_raw_data.isna().sum()
```

```
#overall summary of missing data in the data frame

#visualize missingness
column_order = churn_raw_data.isnull().sum().sort_values().index
msno.bar(churn_raw_data[column_order])
plt.show()

import seaborn as sns
import matplotlib.pyplot as plt

# Generate a correlation matrix
correlation_matrix = churn_raw_data.corr()

# Set the figure size to make the heatmap larger
plt.figure(figsize=(12, 10))

# Create a heat map using seaborn with customizations
heatmap = sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5,
annot_kws={"size": 8})

# Increase the font size of the color bar (optional)
cbar = heatmap.collections[0].colorbar
cbar.ax.tick_params(labelsize=12)

# Increase the font size of the annotations
for text in heatmap.texts:
    text.set_size(8)

# Display the heat map
plt.show()

churn_raw_data.Population[churn_raw_data.Population == 0].count()

#determine if missing income values are related to employment

# Select the rows where income is missing
missing_income = churn_raw_data[churn_raw_data['Income'].isnull()]

# Group the missing income data by employment status
grouped = missing_income.groupby(['Employment'])

# Count the number of missing income values for each employment status
missing_count = grouped.size()

# Calculate the percentage of missing income values for each employment status
percentage_missing = missing_count / churn_raw_data.groupby(['Employment']).size() * 100
```

```
# Display the results
print(percentage_missing)
#determine if missing income values are related to employment and age

# Select the rows where income is missing
missing_income = churn_raw_data[churn_raw_data['Income'].isnull()]

# Group the missing income data by employment status
grouped = missing_income.groupby(['Employment', pd.cut(missing_income['Age'], bins=[0, 18, 30, 50,
np.inf])])

# Count the number of missing income values for each employment status
missing_count = grouped.size()

# Calculate the percentage of missing income values for each employment status
percentage_missing = missing_count / churn_raw_data.groupby(['Employment',
pd.cut(churn_raw_data['Age'], bins=[0, 18, 30, 50, np.inf])]).size() * 100

# Display the results
print(percentage_missing)

#part time employment between ages 0-18 is missing data the most

import pandas as pd

# Select the rows where the 'Techie' column is null
missing_techie = churn_raw_data[churn_raw_data['Techie'].isnull()]

# Define age group bins
age_bins = [18, 30, 50, float('inf')]

# Group the missing 'Techie' data by age
grouped = missing_techie.groupby(pd.cut(missing_techie['Age'], bins=age_bins))

# Count the number of missing 'Techie' values for each age group
missing_count = grouped.size()

# Calculate the percentage of missing 'Techie' values for each age group
total_count_by_age_group = churn_raw_data.groupby(pd.cut(churn_raw_data['Age'],
bins=age_bins)).size()
percentage_missing = (missing_count / total_count_by_age_group) * 100

# Display the results
print(percentage_missing)

import pandas as pd
```

```
# Select the rows where 'number of children' is missing
missing_children = churn_raw_data[churn_raw_data['Children'].isnull()]

# Define age group bins
age_bins = [18, 30, 50, float('inf')]  # Adjust the bins as needed

# Group the missing 'number of children' data by marital status and age
grouped = missing_children.groupby(['Marital', pd.cut(missing_children['Age'], bins=age_bins)])

# Count the number of missing 'number of children' values for each marital status and age group
missing_count = grouped.size()

# Calculate the percentage of missing 'number of children' values for each marital status and age group
total_count_by_group = churn_raw_data.groupby(['Marital', pd.cut(churn_raw_data['Age'],
bins=age_bins)]).size()
percentage_missing = (missing_count / total_count_by_group) * 100

# Display the results
print(percentage_missing)

import numpy as np
churn_raw_data.loc[churn_raw_data.Population == 0, 'Population'] = np.nan
churn_raw_data.Population[np.isnan(churn_raw_data.Population)]

pip install fancyimpute
from fancyimpute import KNN

columns_to_impute = ['Children', 'Income', 'Age', 'Bandwidth_GB_Year', 'Population', 'Tenure']

# Create an instance of the KNN imputer
knn_imputer = KNN()

# Get the column indices of the columns to impute
columns_to_impute_indices = [churn_raw_data.columns.get_loc(col) for col in columns_to_impute]

# Perform imputation on the selected columns
imputed_values = knn_imputer.fit_transform(churn_raw_data.iloc[:, columns_to_impute_indices])

# Assign the imputed values back to the original dataset
churn_raw_data.iloc[:, columns_to_impute_indices] = imputed_values

# Verify if the imputations are in the original dataset
print(churn_raw_data.head())

from sklearn.impute import SimpleImputer

column_to_impute_two = ['Techie', 'Phone', 'TechSupport']
```

```
# Create the SimpleImputer object with strategy='most_frequent'
imputer = SimpleImputer(strategy='most_frequent')

# Fit and transform the categorical variables using the imputer
churn_raw_data[column_to_impute_two] =
imputer.fit_transform(churn_raw_data[column_to_impute_two])

import pandas as pd
# Define age group bins
age_bins = [18, 30, 50, float('inf')]

# Group the data by age
grouped = churn_raw_data.groupby(pd.cut(churn_raw_data['Age'], bins=age_bins))

# Count the number of 'Techie' values for each age group
tech_count = grouped['Techie'].value_counts().unstack(fill_value=0)

# Calculate the percentage of 'Techie' values for each age group
tech_percentage = (tech_count / tech_count.sum(axis=1).values[:, None]) * 100

# Display the results
print(tech_percentage)

#data clean
import matplotlib.pyplot as plt
column_order = churn_raw_data.isnull().sum().sort_values().index
msno.bar(churn_raw_data[column_order])
plt.show()

cleaned_data_two = churn_raw_data
new_name_two = 'churn_cleaned_data'
new_cleaned_data_two = cleaned_data_two.copy()
new_cleaned_data_two.name = new_name_two
new_cleaned_data_two.to_csv('new_cleaned_data_two.csv', index=False)
churn_raw_data.to_csv(r'/Users/jasminemoniquecooper/Downloads/new_cleaned_data_two.csv')
```

C2.

For the dependent variable, tenure, I have 10,000 observations to examine the summary statistics:

	Tenure
count	10000.000000
mean	34.624193
std	25.903916
min	1.000259
25%	8.197844
50%	36.852175
75%	60.457955
max	71.999280

- The mean indicates that on average customers stay with telecommunications companies for approximately 34 days
- The standard deviation of 25 days shows that individual customer tenures can vary from the mean by plus or minus 25 days
- The minimum tenure observed is 1 day, while the maximum tenure is approximately 71 days which is indicative of the range of customer durations
- The 25th percentile is around 8 days, meaning that 25% of customers stay for 8 days or less
- The 75th percentile is around 60 days, meaning that 75% of customers stay for 60 days or less
- The median, which is around 36 days, meaning that 50% of customers stay for more than 36 days, and 50% of customers stay for less than 36 days

The proximity of the mean and median suggests that the distribution may approximate a bell curve or normal distribution.
For the independent variables, Outage_sec_perweek, MonthlyCharge, and Area I have 10,000 observations to examine the summary statistics:

Independent variable 1: Outage_sec_perweek – average number of seconds per week of system outages in the customer's neighborhood

- The mean indicates that on average customers experience an 11 second average outage in their neighborhood per week
- The standard deviation of 7 seconds shows that average outages per week in the customer's neighborhood can vary from the mean by plus or minus 7 seconds
- The minimum seconds of outages per week is negative, which is not a valid measure of time, so I will say that the minimum amount of time a customer on average experiences an outage in their neighborhood is 0 seconds. In comparison, the maximum amount of time a customer may on average experience an outage in their neighborhood is 47 seconds.
- The 25th percentile is around 8 seconds, meaning that 25% of customers on average experience an outage of 8 seconds per week or less
- The 75th percentile is around 12 seconds, meaning that 75% of customers on average experience an outage of 12 seconds per week or less
- The median, which is around 10 seconds, meaning that 50% of customers on average experience an outage for more than 10 seconds per week, and 50% of customers experience an outage, on average, for less than 10 seconds per week

The proximity of the mean and median suggests that the distribution may approximate a bell curve or normal distribution.

Independent variable 2: MonthlyCharge – The amount charged to the customer monthly

- The mean indicates that on average customers are charged $174 monthly
- The standard deviation of $43 shows that customer's monthly charges can vary from the mean by plus or minus $43

- The minimum amount a customer is charged monthly is $77, while the maximum monthly charge to a customer is $315, which is indicative of the range of customer monthly charges
- The 25th percentile is $141, meaning that 25% of customers receive a monthly charge of $141 or less
- The 75th percentile is $203, meaning that 75% of customers receive a monthly charge of $203 or less
- The median, which is $169, meaning that 50% of customers receive a monthly charge more than $169, and 50% of customers receive a monthly charge less than $169

The proximity of the mean and median suggests that the distribution may approximate a bell curve or normal distribution.

Independent variable 3: Area – area type (rural, urban, or suburban)

Since the area is a non-numerical, categorical independent variable then I can evaluate summary statistics by examining the frequency of occurrence for each area type.

Frequency of Occurrence for Each Area Type:
- Rural – count of observations in the rural area
- Urban – count of observations in the urban area
- Suburban – count of observations in the suburban area

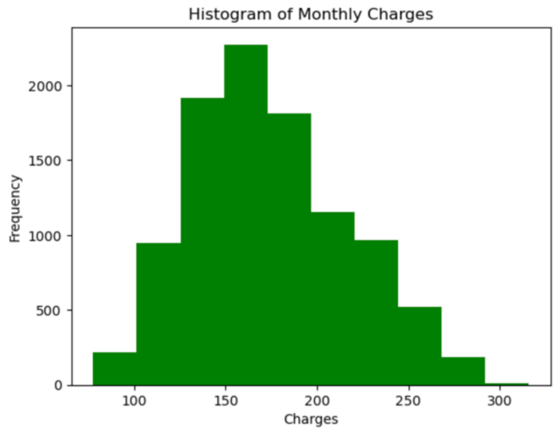
Based on the output below, it appears that the 10,000 observations are evenly distributed between all 3 area types: rural, urban, and suburban. The balance in the distribution of area types indicates that the dataset provides a representative sample, which is critical for drawing meaningful conclusions in our research.

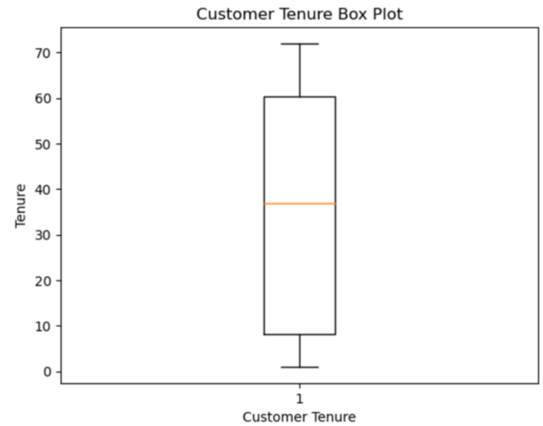
	MonthlyCharge	Outage_sec_perweek
count	10000.000000	10000.000000
mean	174.076305	11.452955
std	43.335473	7.025921
min	77.505230	-1.348571
25%	141.071078	8.054362
50%	169.915400	10.202896
75%	203.777441	12.487644
max	315.878600	47.049280

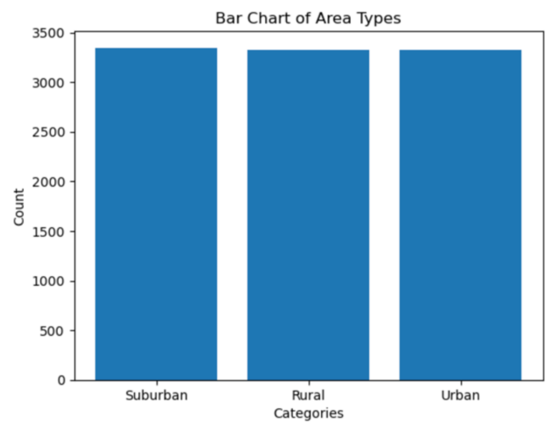
```
Mode (most frequent 'Area'):  Suburban

Frequency counts of 'Area':
Suburban    3346
Rural       3327
Urban       3327
Name: Area, dtype: int64

Percentage distribution of 'Area':
Suburban    33.46
Rural       33.27
Urban       33.27
Name: Area, dtype: float64
```

C3.

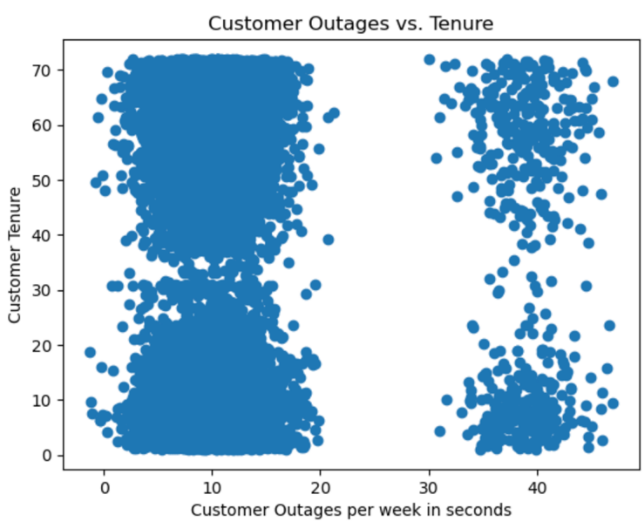
Univariate visualizations:

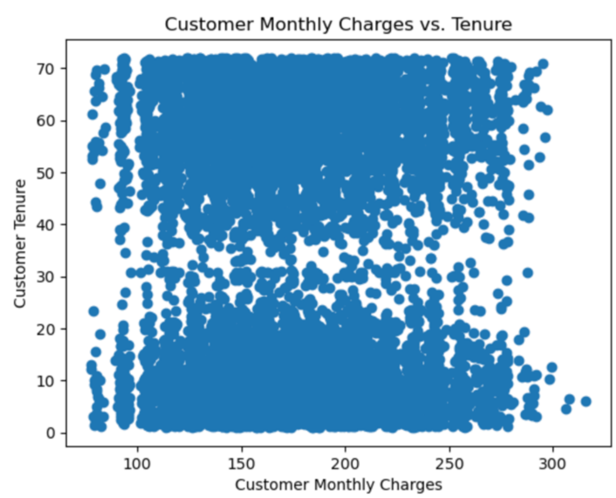




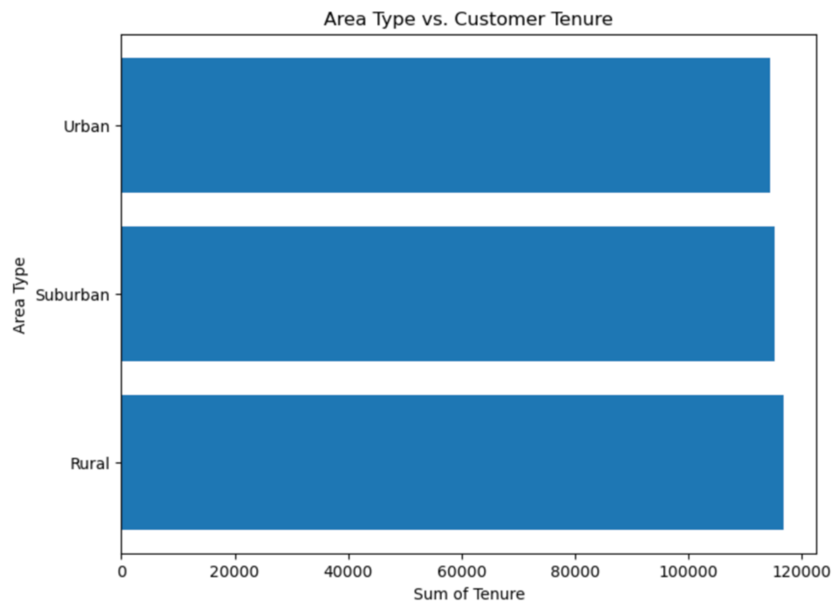


Bivariate visualizations:





Jasmine Cooper
D208
Linear Regression Modeling
October 1, 2023
Western Governor's University



C4.
Converting columns from data type object to data type category was necessary because there are only a few different values to be selected from each variable and it helps to save memory and ensures that Python recognizes these variables as categorical variables in other libraries and visualizations. Specifically, converting the Area variable allows me to properly calculate the frequency of each area type and portray accurate univariate and bivariate visualizations. Furthermore, to effectively conduct feature selection for the reduced model, I employed one hot encoding, which created three separate binary variables for the Area variable.

List of columns converted from data type object to data type category:

- Employment, Marital, Gender, Churn, Techie, Contract, Post_modem, Tablet, InternetService, Phone, Multiple, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, PaperlessBilling, PaymentMethod, and Area

Code for C4 (changed variable names as needed):

```
churn_clean_data["Area"] = churn_clean_data["Area"].astype('category')
churn_clean_data["Area"].dtypes

#convert Area from category to numerical
#one hot encoding for three categories

data = {'Area': ['Urban', 'Suburban', 'Rural', 'Urban', 'Suburban']}

# Performs one-hot encoding
df_encoded = pd.get_dummies(churn_clean_data, columns=['Area'])
```

print(df_encoded)

C5.
Please see ‘new_cleaned_data_two.csv’ file attached to submission

In text citations:
(“Completeness”, n.d.)
(“Importing flat files using pandas”, n.d.)
(“Is data missing at random?”, n.d.)
(“Mean, median, & mode imputations”, n.d.)
(“Imputing using facncyimpute”, n.d.)
(“Measures of center”, n.d.)
(“Measures of spread”, n.d.)
(Gudikandula, 2018)

Part 4

D1.
Initial linear regression model:
from statsmodels.formula.api import ols
mdl_initial = ols("Tenure ~ Outage_sec_perweek + MonthlyCharge + Area + 0", data = churn_clean_data).fit()
print(mdl_initial.summary())

Initial model summary:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Tenure   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                 -0.000
Method:                 Least Squares   F-statistic:                    0.5786
Date:                Tue, 26 Sep 2023   Prob (F-statistic):              0.678
Time:                        14:08:37   Log-Likelihood:                -46732.
No. Observations:               10000   AIC:                         9.347e+04
Df Residuals:                    9995   BIC:                         9.351e+04
Df Model:                           4
Covariance Type:            nonrobust
======================================================================================
                         coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------
Area[Rural]           35.6743      1.171     30.478      0.000      33.380      37.969
Area[Suburban]        35.0060      1.169     29.948      0.000      32.715      37.297
Area[Urban]           34.9748      1.169     29.914      0.000      32.683      37.267
Outage_sec_perweek     0.0185      0.037      0.498      0.618      -0.054       0.091
MonthlyCharge         -0.0046      0.006     -0.768      0.443      -0.016       0.007
==============================================================================
Omnibus:                    42080.856   Durbin-Watson:                   0.412
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1252.417
Skew:                           0.054   Prob(JB):                    1.10e-272
Kurtosis:                       1.270   Cond. No.                     1.33e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.33e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

D2.

To evaluate the performance and accuracy as well as reduce the initial model, I will use the Root Mean Squared Error (RMSE). The RMSE is a robust way to assess the quality of the model's predictions by quantifying the average magnitude of prediction errors. This choice is justified for several reasons:

1. Minimizes prediction errors: To effectively answer the research question, precise and reliable predictions are required. RMSE is designed to measure and minimize prediction errors.
2. Quantifying Model Fit: RMSE quantifies the goodness of fit between the predictive model and the actual data. A lower RMSE signifies that the model's predictions do not vary much from the actual data, ensuring accuracy.
3. Measuring Uncertainty: RMSE measures the spread of residuals and is directly related to the measure of uncertainty. Lower RMSE values indicates a better fitting model, and a higher degree of confidence in the model's predictions. This is essential for providing effective course(s) of action related to the research question.

Utilizing RMSE as the model evaluation metric, will strive to achieve a model that aligns with the research question and demonstrates an accuracy in making predictions. In summary, RMSE will enhance model interpretability, reduce overfitting, and provide precise predictions, which are all necessary pieces of information for addressing the research question effectively.

D3.
Reduced linear regression model:

```
mdl_reduced = ols("Tenure ~ Area_Rural + Area_Suburban + Area_Urban + 0", data =
feature_churn_data).fit()
print(mdl_reduced.summary())
```

Reduced model summary:

```
[12]: from statsmodels.formula.api import ols
      print(mdl_reduced.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Tenure   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                 -0.000
Method:                 Least Squares   F-statistic:                    0.7819
Date:                Fri, 29 Sep 2023   Prob (F-statistic):              0.458
Time:                        20:14:41   Log-Likelihood:                -46732.
No. Observations:               10000   AIC:                         9.347e+04
Df Residuals:                    9997   BIC:                         9.349e+04
Df Model:                           2
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
Intercept        25.9684      0.194    133.662      0.000      25.588      26.349
Area_Rural        9.1141      0.372     24.482      0.000       8.384       9.844
Area_Suburban     8.4451      0.372     22.732      0.000       7.717       9.173
Area_Urban        8.4093      0.372     22.589      0.000       7.680       9.139
==============================================================================
Omnibus:                    42066.461   Durbin-Watson:                   0.412
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1252.902
Skew:                           0.054   Prob(JB):                    8.63e-273
Kurtosis:                       1.269   Cond. No.                     1.82e+15
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.02e-27. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

E1.

The data analysis process involved the following:

1. Initial Model Analysis – I initiated the data analysis process by constructing an initial multiple linear regression model. Then, to assess the performance, I computed the RMSE as the model evaluation metric to serve as a base line for comparison.
2. Data Transformation – I recognized that the categorical independent variable, Area, needed to be included into the model, and I employed one-hot encoding. This transformation resulted in the creation of three separate binary variables: Area_Urban, Area_Suburban, and Area_Rural.
3. Feature selection – To refine the model, I choose to conduct recursive feature selection to identify the most influential variables. Through this process, it was determined that the three encoded variables (Area_Urban, Area_Suburban, and Area_Rural) were the most significant contributors to the model's performance.
4. Reduced Model Analysis – Using the selected features, I built a reduced linear regression model. Then, I recalculated the RMSE for the reduced model to gauge its predictive accuracy.
5. Comparison Analysis – Upon comparing the RMSE values of the initial and reduced models, it was determined that both models yielded an RMSE of approximately 25. This outcome indicates that the reduced model, which only includes the most essential features, can achieve a comparable predictive performance with a simpler model.

In text citations:
("More than two explanatory variables", n.d.)

E2.

Output and all calculations of data analysis performed:

1. Initial Model Analysis
    a. Calculations and output summary provided in D1
    b. Initial Model RMSE calculation:

```
[10]: residuals = mdl_initial.resid
      print(residuals)
```

```
0      -27.514666
1      -32.915893
2      -18.672315
3      -17.643798
4      -32.802992
          ...    
9995    33.091200
9996    26.182767
9997   -21.538278
9998    37.066703
9999    29.152857
Length: 10000, dtype: float64
```

```
[11]: RSS = np.sum( np.square(residuals) )
      print (RSS)
```

```
6707904.60854712
```

```
[12]: mean_squared_residuals = np.sum( np.square(residuals) ) / len(residuals)
      print(mean_squared_residuals)
```

```
670.790460854712
```

```
[13]: MSE = np.mean( np.square(residuals) )
      print(MSE)
```

```
670.790460854712
```

```
[14]: RMSE = np.sqrt(np.mean( np.square(residuals)))
      print(RMSE)
```

```
25.899622793676205
```

2. Data Transformation calculations and outputs:

a. One hot encoding:

```
[18]: #convert Area from category to numerical
      #one hot encoding for three categories

      data = {'Area': ['Urban', 'Suburban', 'Rural', 'Urban', 'Suburban']}

      # Performs one-hot encoding
      df_encoded = pd.get_dummies(churn_clean_data, columns=['Area'])
      print(df_encoded)
```

```
      Unnamed: 0.1  Unnamed: 0  CaseOrder Customer_id  \
0                0           1          1     K409198   
1                1           2          2     S120509   
2                2           3          3     K191035   
3                3           4          4      D90850   
4                4           5          5     K662701   
...            ...         ...        ...         ...   
9995          9995        9996       9996     M324793   
9996          9996        9997       9997     D861732   
9997          9997        9998       9998     I243405   
9998          9998        9999       9999     I641617   
9999          9999       10000      10000      T38070   

                               Interaction          City State  \
0     aa90260b-4141-4a24-8e36-b04ce1f4f77b   Point Baker    AK   
1     fb76459f-c047-4a9d-8af9-e0f7d4ac2524  West Branch    MI   
2     344d114c-3736-4be5-98f7-c72c281e2d35       Yamhill    OR   
3     abfa2b40-2d43-4994-b15a-989b8c79e311       Del Mar    CA   
4     68a861fd-0d20-4e51-a587-8a90407ee574    Needville    TX   
...                                    ...           ...   ...   
9995  45deb5a2-ae04-4518-bf0b-c82db8dbe4a4   Mount Holly    VT   
9996  6e96b921-0c09-4993-bbda-a1ac6411061a   Clarksville    TN   
9997  e8307ddf-9a01-4fff-bc59-4742e03fd24f      Mobeetie    TX   
9998  3775ccfc-0052-4107-81ae-9657f81ecdf3    Carrollton    GA   
```

```
     DeviceProtection TechSupport StreamingTV StreamingMovies  \
0                  No          No          No             Yes   
1                  No          No         Yes             Yes   
2                  No          No          No             Yes   
3                  No          No         Yes              No   
4                  No         Yes         Yes              No   
...               ...         ...         ...             ...   
9995              Yes          No          No              No   
9996              Yes          No         Yes              No   
9997               No          No          No              No   
9998               No         Yes         Yes             Yes   
9999              Yes          No          No             Yes   

     PaperlessBilling            PaymentMethod     Tenure  MonthlyCharge  \
0                 Yes  Credit Card (automatic)   6.795513     171.449762   
1                 Yes Bank Transfer(automatic)   1.156681     242.948015   
2                 Yes  Credit Card (automatic)  15.754144     159.440398   
3                 Yes             Mailed Check  17.087227     120.249493   
4                  No             Mailed Check   1.670972     150.761216   
...               ...                      ...        ...            ...   
9995               No         Electronic Check  68.197130     159.828800   
9996               No         Electronic Check  61.040370     208.856400   
9997              Yes Bank Transfer(automatic)  13.446717     168.220900   
9998              Yes  Credit Card (automatic)  71.095600     252.628600   
9999              Yes         Electronic Check  63.350860     218.371000   

      Bandwidth_GB_Year  item1  item2  item3  item4  item5  item6  item7  \
0            904.536110      5      5      5      3      4      4      3   
1            800.982766      3      4      3      3      4      3      4   
2           2054.706961      4      4      2      4      4      3      3   
3           2164.579412      4      4      4      2      5      4      3   
4            271.493436      4      4      4      3      4      4      4   
...                 ...    ...    ...    ...    ...    ...    ...    ...   
9995        6511.253000      3      2      3      3      4      3      2   
9996        5695.952000      4      5      5      4      4      5      2   
```

```
            Marital  Gender Churn  Outage_sec_perweek  Email  Contacts  \
0           Widowed    Male    No            6.972566     10         0   
1           Married  Female   Yes           12.014541     12         0   
2           Widowed  Female    No           10.245616      9         0   
3           Married    Male    No           15.206193     15         2   
4         Separated    Male   Yes            8.960316     16         2   
...             ...     ...   ...                 ...    ...       ...   
9995        Married    Male    No            9.265392     12         2   
9996       Divorced    Male    No            8.115849     15         2   
9997  Never Married  Female    No            4.837696     10         0   
9998      Separated    Male    No           12.076460     14         1   
9999  Never Married    Male    No           12.641760     17         1   

      Yearly_equip_failure Techie        Contract Port_modem Tablet  \
0                        1     No        One year        Yes    Yes   
1                        1    Yes  Month-to-month         No    Yes   
2                        1    Yes        Two Year        Yes     No   
3                        0    Yes        Two Year         No     No   
4                        1     No  Month-to-month        Yes     No   
...                    ...    ...             ...        ...    ...   
9995                     0     No  Month-to-month        Yes    Yes   
9996                     0     No        Two Year         No     No   
9997                     0     No  Month-to-month         No     No   
9998                     0     No        Two Year         No    Yes   
9999                     0     No  Month-to-month        Yes     No   

     InternetService Phone Multiple OnlineSecurity OnlineBackup  \
0        Fiber Optic   Yes       No            Yes          Yes   
1        Fiber Optic   Yes      Yes            Yes           No   
2                DSL   Yes      Yes             No           No   
3                DSL   Yes       No            Yes           No   
4        Fiber Optic    No       No             No           No   
...              ...   ...      ...            ...          ...   
9995             DSL   Yes      Yes             No          Yes   
9996     Fiber Optic   Yes      Yes            Yes          Yes   
```

```
                      County    Zip       Lat        Lng  Population  \
0     Prince of Wales-Hyder  99927  56.25100 -133.37571        38.0   
1                    Ogemaw  48661  44.32893  -84.24080     10446.0   
2                   Yamhill  97148  45.35589 -123.24657      3735.0   
3                 San Diego  92014  32.96687 -117.24798     13863.0   
4                 Fort Bend  77461  29.38012  -95.80673     11352.0   
...                     ...    ...       ...        ...         ...   
9995                Rutland   5758  43.43391  -72.78734       640.0   
9996             Montgomery  37042  36.56907  -87.41694     77168.0   
9997                Wheeler  79061  35.52039 -100.44180       406.0   
9998                Carroll  30117  33.58016  -85.13241     35575.0   
9999              Habersham  30523  34.70783  -83.53648     12230.0   

                 Timezone                               Job  Children  \
0           America/Sitka  Environmental health practitioner  0.005273   
1         America/Detroit          Programmer, multimedia  1.000000   
2     America/Los_Angeles         Chief Financial Officer  4.000000   
3     America/Los_Angeles                       Solicitor  1.000000   
4         America/Chicago             Medical illustrator  0.000000   
...                   ...                             ...       ...   
9995     America/New_York    Sport and exercise psychologist  3.000000   
9996      America/Chicago        Consulting civil engineer  4.000000   
9997      America/Chicago     IT technical support officer  0.873586   
9998     America/New_York                  Water engineer  1.000000   
9999     America/New_York              Personal assistant  1.000000   

            Age                     Education Employment        Income  \
0     68.000000               Master's Degree  Part Time  28561.990000   
1     27.000000    Regular High School Diploma    Retired  21704.770000   
2     50.000000    Regular High School Diploma    Student  21157.834718   
3     48.000000              Doctorate Degree    Retired  18925.230000   
4     83.000000               Master's Degree    Student  40074.190000   
...         ...                           ...        ...           ...   
9995  54.872456  Some College, Less than 1 Year    Retired  55723.740000   
9996  48.000000    Regular High School Diploma  Part Time  33267.476603   
```

```
      item8  Area_Rural  Area_Suburban  Area_Urban
0         4           0              0           1
1         4           0              0           1
2         3           0              0           1
3         3           0              1           0
4         5           0              1           0
...     ...         ...            ...         ...
9995      3           1              0           0
9996      5           1              0           0
9997      5           1              0           0
9998      4           0              0           1
9999      1           0              0           1

[10000 rows x 55 columns]
```

Jasmine Cooper
D208
Linear Regression Modeling
October 1, 2023
Western Governor's University

3. Feature selection calculations and outputs:

```
[20]: feature_churn_data = df_encoded[['Tenure', 'Outage_sec_perweek', 'MonthlyCharge',
                                      'Area_Rural', 'Area_Suburban', 'Area_Urban']]

      print(feature_churn_data)
```

```
         Tenure  Outage_sec_perweek  MonthlyCharge  Area_Rural  Area_Suburban  \
0      6.795513            6.972566     171.449762           0              0
1      1.156681           12.014541     242.948015           0              0
2     15.754144           10.245616     159.440398           0              0
3     17.087227           15.206193     120.249493           0              1
4      1.670972            8.960316     150.761216           0              1
...         ...                 ...            ...         ...            ...
9995  68.197130            9.265392     159.828800           1              0
9996  61.040370            8.115849     208.856400           1              0
9997  13.446717            4.837696     168.220900           1              0
9998  71.095600           12.076460     252.628600           0              0
9999  63.350860           12.641760     218.371000           0              0

      Area_Urban
0              1
1              1
2              1
3              0
4              0
...          ...
9995           0
9996           0
9997           0
9998           1
9999           1

[10000 rows x 6 columns]
```

```
[21]: #import libraries
      from sklearn.feature_selection import RFE
      from sklearn.linear_model import LinearRegression

      # Extract your target variable (y) and feature matrix (X) from your data
      X = feature_churn_data.drop('Tenure', axis=1)
      y = feature_churn_data['Tenure']

      #estimator (machine learning model) for RFE
      estimator = LinearRegression()

      #create the RFE object with the estimator and number of features to select
      rfe = RFE(estimator, n_features_to_select=3)
      rfe.fit(X, y)  # X is the feature matrix, y is the target variable

      #fit RFE to the data
      rfe.fit(X, y)

      #get the selected features and it returns a boolean mask to indicate which features are selected
      selected_features = rfe.support_
      print("Selected Features:", selected_features)

      #rank the selected features and lower rankings mean the features are more important
      feature_ranking = rfe.ranking_
      print("Feature Ranking:", feature_ranking)

      # use the selected features to create a reduced feature matrix
      #train your linear regression model with only these features
      X_reduced = X.loc[:, selected_features]
      print("Reduced Feature Matrix (X_reduced):", X_reduced)
```

```
Selected Features: [False False  True  True  True]
```

```
Selected Features: [False False  True  True  True]
Feature Ranking: [2 3 1 1 1]
Reduced Feature Matrix (X_reduced):       Area_Rural  Area_Suburban  Area_Urban
0              0              0           1
1              0              0           1
2              0              0           1
3              0              1           0
4              0              1           0
...          ...            ...         ...
9995           1              0           0
9996           1              0           0
9997           1              0           0
9998           0              0           1
9999           0              0           1

[10000 rows x 3 columns]
```

4. Reduced Model Analysis:

a. Calculations and output summary provided in D3

```
[10000 rows x 3 columns]
```

```
[22]: mdl_reduced = ols("Tenure ~ Area_Rural + Area_Suburban + Area_Urban + 0", data = feature_churn_data).fit()
```

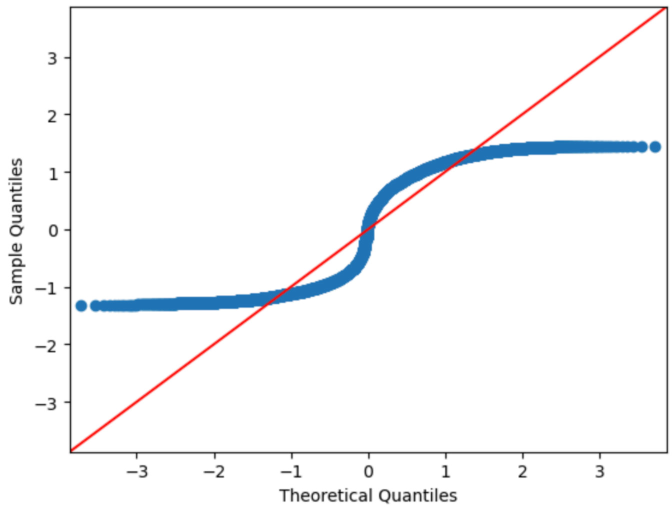
```
[26]: from sklearn.model_selection import train_test_split
      from sklearn.metrics import mean_squared_error

      # Step 3: Split Data into Training and Testing Sets
      X = feature_churn_data[['Area_Rural', 'Area_Suburban', 'Area_Urban']]
      y = feature_churn_data['Tenure']

      # Split data into 80% training and 20% testing
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

      # Step 4: Train the Model
      model = LinearRegression()
      model.fit(X_train, y_train)

      # Step 5: Make Predictions
      y_pred = model.predict(X_test)

      # Step 6: Evaluate with RMSE
      rmse = np.sqrt(mean_squared_error(y_test, y_pred))
      print("Root Mean Squared Error (RMSE):", rmse)
```

```
Root Mean Squared Error (RMSE): 25.923698189470592
```

5. Residual plot for the reduced model (qqplot):

[25]:
```python
from statsmodels.api import qqplot
qqplot(data = mdl_reduced.resid, fit=True, line="45")
```

[25]:



6. Model's residual standard error for the reduced model:

[30]:
```python
#residual standard error of reduced model
mse = mdl_reduced.mse_resid
rse = np.sqrt(mse)
print('rse:', rse)
```

```
rse: 25.904481450646614
```

E3. Please see Churn Clean Data.ipynb for code

In text citations:
(“Goodness-of-Fit”, n.d.)
(“The curse of dimensionality”, n.d.)
(“Selecting features for model performance”, n.d.)
(Lekhana_Ganji, 2023)
(“Working with model objects”, n.d.)

Part 5

F1.
Data Summary:

Regression equation for the reduced model:
Y = 9.1141*Area_Rural + 8.4451*Area_Suburban + 8.4093*Area_Urban + 25.9684

Interpretation of the coefficients of the reduced model:
$Y = the\ dependent\ variable\ the\ model\ is\ predicting$

$\beta_0 = The\ constant\ intercept\ term$
$\beta_1 = the\ coefficient\ for\ the\ Area\ Rural\ independent\ variable$
$\beta_2 = the\ coefficient\ for\ the\ Area\ Suburban\ indepedent\ variable$
$\beta_3 = the\ coefficient\ for\ the\ Area\ Urban\ indepedent\ variable$
$The\ indepedent\ variable\ coefficients\ represent\ the\ change\ in\ the\ dependent\ variable\ for\ one\ unit\ change\ in\ the\ corresponding\ independent\ variable, while\ holding\ all\ other\ variables\ constant.$

Statistical and practical significance of the reduced model:
The p-values for the independent variables were below a well-known significance level of alpha = 0.05, and therefore, were found to be statistically significant. This indicates that these independent variables have a statistically significant impact on the dependent variable.

While the coefficients of the independent variables are statistically significant, it is also important to consider their practical significance. The reduced model could be meaningful in the real world if telecommunication companies want to assess where the highest customer churn occurs based on geographical location.

Data Analysis limitations:
While the analysis has produced meaningful insights, the analysis process limits the telecommunication companies from determining other predictor factors for customer churn. The initial model, which serves as the foundation for feature selection and reduction was constructed with a limited set of independent variables to reduce overall model complexity. This constraint limited the range of independent variables available for consideration during feature selection.

After the recursive feature selection process, the selected features were focused on the customer's geographic location as a significant predictor of customer churn can be a limitation. Although meaningful, it limits the companies' ability to explore and identify other potential predictor factors that might influence churn behavior. In conclusion, customer churn can be a complex decision that can be influenced by a plethora of factors, and more independent variables could have helped to relieve some limitations.

F2.
Based on my results, the recommended course of action would be to investigate how customer's geographic location affects customer's churn. Using customer's geographical location, the telecommunication companies should zone in on one geographical location, and then rerun the initial model with additional independent variables, and the predicted dependent variable being the geographical location where customer churn is highly likely. By using this course of action, the telecommunications companies can develop detailed insights to proactively prevent customer churn.

In text citations:
(LaMorte, 2016)
("Inference on transformed variables", n.d.)

Part 6
H.
Citations for code:

DataCamp. (n.d.). More than two explanatory variables [Video file]. Retrieved from https://campus.datacamp.com/courses/intermediate-regression-with-statsmodels-in-python/multiple-linear-regression-3?ex=7

DataCamp. (n.d.). Completeness [Video file]. Retrieved from https://campus.datacamp.com/courses/cleaning-data-in-python/advanced-data-problems-3?ex=8

DataCamp. (n.d.). Importing flat files using pandas [Video file]. Retrieved from https://campus.datacamp.com/courses/introduction-to-importing-data-in-python/introduction-and-flat-files-1?ex=15

DataCamp. (n.d.). Mean, median, & mode imputations [Video file]. Retrieved from https://campus.datacamp.com/courses/dealing-with-missing-data-in-python/imputation-techniques?ex=1

DataCamp. (n.d.). Imputing using fancyimpute [Video file]. Retrieved from https://campus.datacamp.com/courses/dealing-with-missing-data-in-python/advanced-imputation-techniques?ex=1

DataCamp. (n.d.). The curse of dimensionality [Video file]. Retrieved from https://campus.datacamp.com/courses/dimensionality-reduction-in-python/feature-selection-i-selecting-for-feature-information?ex=1

DataCamp. (n.d.). Selecting features for model performance [Video file]. Retrieved from https://campus.datacamp.com/courses/dimensionality-reduction-in-python/feature-selection-ii-selecting-for-model-accuracy?ex=1

Lekhana_Ganji. (2023, April 18). Machine Learning - One Hot Encoding of Datasets in Python. GeeksforGeeks. https://www.geeksforgeeks.org/ml-one-hot-encoding-of-datasets-in-python/

DataCamp. (n.d.). Working with model objects [Video file]. Retrieved from https://campus.datacamp.com/courses/introduction-to-regression-with-statsmodels-in-python/predictions-and-model-objects-2?ex=5

DataCamp. (n.d.). Visualizing model fit [Video file]. Retrieved from https://campus.datacamp.com/courses/introduction-to-regression-with-statsmodels-in-python/assessing-model-fit-e78fd9fe-6303-4048-8748-33b19c4222fe?ex=4

DataCamp. (n.d.). Quantifying model fit [Video file]. Retrieved from https://campus.datacamp.com/courses/introduction-to-regression-with-statsmodels-in-python/assessing-model-fit-e78fd9fe-6303-4048-8748-33b19c4222fe?ex=1

I.
Citations for content:

DataCamp. (n.d.). Technical conditions for linear regression [Video file]. Retrieved from https://campus.datacamp.com/courses/inference-for-linear-regression-in-r/technical-conditions-in-linear-regression?ex=1

DataCamp. (n.d.). Goodness-of-fit [Video file]. Retrieved from https://campus.datacamp.com/courses/introduction-to-linear-modeling-in-python/making-model-predictions?ex=8

DataCamp. (n.d.). Is the data missing at random? [Video file]. Retrieved from https://campus.datacamp.com/courses/dealing-with-missing-data-in-python/does-missingness-have-a-pattern?ex=1

DataCamp. (n.d.). Measures of center [Video file]. Retrieved from https://campus.datacamp.com/courses/introduction-to-statistics-in-python/summary-statistics-1?ex=4

DataCamp. (n.d.). Measures of spread [Video file]. Retrieved from https://campus.datacamp.com/courses/introduction-to-statistics-in-python/summary-statistics-1?ex=7

DataCamp. (n.d.). Inference on transformed variables [Video file]. Retrieved from https://campus.datacamp.com/courses/inference-for-linear-regression-in-r/building-on-inference-in-simple-linear-regression?ex=1

LaMorte, W. (2016, May 31). The Multiple Linear Regression Equation. Multivariate Methods. https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704-ep713_multivariablemethods/bs704-ep713_multivariablemethods2.html

Gudikandula, P. (2018, November 9). Exploratory Data Analysis (beginner), Univariate, Bivariate and Multivariate – Habberman dataset. purnasaigudikandula.medium.com. Retrieved from https://purnasaigudikandula.medium.com/exploratory-data-analysis-beginner-univariate-bivariate-and-multivariate-habberman-dataset-2365264b751