Sampling the Dataset

1. Why does `skew(customer_df)` return six values?
2. Why did I take the transpose to of stats = customer_df.describe().T
3. What can we discern from the statistical description of the dataset?
4. How do our samples compare to the overall dataset?
5. When we repeatedly sample the dataset, what is the effect on the averaged mean?
6. Is a 10% sample good enough for plotting?

Correlation and Redundancy

1. What does it mean if I drop a feature and the others can predict it?
2. Why do I specify that the feature should be dropped on axis 1?
3. Why should we do this process multiple times?
4. How can we see redundancy in a pair plot?
5. How can we see redundancy in a correlation plot?

Transforming Data

1. How can we see skew in the relationship between the mean and the median?

2. Looking at the distribution plot of the data, what happens to the plots when we scale the data?

3. Does scaling the data effect the skew? Check the numbers.

4. Does taking the log of the data effect the skew? Check the numbers.

5. Looking at the distribution plot of the data, what happens to the plots when we deskew the data?

6. OPTIONAL Why would we not want to scale the data before applying the box-cox transform?

7. How does the log transform compare to the Box-Cox transform?

8. Why would we want to use a log over Box-Cox? Why would we want to use Box-Cox over the log?

9. What does pickling do?

Identifying and Removing Outliers

1. What does `param=1.5` in the argument definition for this function do?
```
def display_outliers(dataframe, col, param=1.5):
    Q1 = np.percentile(dataframe[col], 25)
    ...
```

2. What does each of these lines do?

    1. `less_than_Q1 = dataframe[col] < Q1 - tukey_window`

    2. `greater_than_Q3 = dataframe[col] > Q3 + tukey_window`

    3. `tukey_mask = (less_than_Q1 | greater_than_Q3)`

    4. `return dataframe[tukey_mask]`

3. What should we do with regard to outliers?