Power Outage

Name(s): Luke, Andrew

Website Link: (your website link)

Power Outage Analysis

In this Jupyter Notebook, we will analyze a dataset containing major power outage data in the continental U.S. The dataset covers the period from January 2000 to July 2016.

Import Statements

Pandas

Pandas is a powerful data manipulation library. We will use it to load, clean, and analyze the power outage dataset.

NumPy

NumPy is a library for numerical operations in Python. It provides support for large, multi-dimensional arrays and matrices. We may use it for numerical computations related to the power outage data.

OS

The os module provides a way to interact with the operating system. We might use it for handling file paths or checking the existence of files.

Folium

Folium is a Python library that makes it easy to visualize spatial data and create interactive maps. We will use it to create an interactive map displaying the geographical distribution of power outages.

Geopy

Geopy provides tools for geocoding (finding the latitude and longitude of an address) and reverse geocoding (finding the address of a set of latitude and longitude coordinates). We may use it in conjunction with Folium for location-based analysis.

HeatMap (Folium Plugin)

The HeatMap plugin from Folium allows us to create a heatmap layer on our interactive map. It can be used to visualize the intensity or concentration of power outages in different geographical areas.

In [ ]:
```
import pandas as pd
import numpy as np
import os
from scipy.stats import ks_2samp

# Interactive Map
import folium
from geopy.geocoders import Nominatim
from folium.plugins import HeatMap

import plotly.express as px
pd.options.plotting.backend = 'plotly'
```

Data Cleaning and Preparation

In this section, we perform several steps to clean and prepare the power outage data for analysis.

In [ ]:
```
# Read the Excel file into a pandas DataFrame
outage = pd.read_excel("outage.xlsx", sheet_name="Masterdata")

# Drop informational rows
outage_cleaned = outage.drop(range(4)).dropna(axis=1, how='all')

# Set column names based on the first row
outage_cleaned.columns = outage_cleaned.iloc[0]

# Drop rows related to units and variables
outage_cleaned = outage_cleaned.drop([4, 5])
outage_cleaned = outage_cleaned.drop(columns="variables")

# Combine 'OUTAGE.START.DATE' and 'OUTAGE.START.TIME' into a new datetime column
outage_cleaned['OUTAGE.START'] = pd.to_datetime(outage_cleaned['OUTAGE.START.DATE']

# Combine 'OUTAGE.RESTORATION.DATE' and 'OUTAGE.RESTORATION.TIME' into a new dateti
outage_cleaned['OUTAGE.RESTORATION'] = pd.to_datetime(outage_cleaned['OUTAGE.RESTOR

# Drop the original date and time columns
outage_cleaned = outage_cleaned.drop(['OUTAGE.START.DATE', 'OUTAGE.START.TIME', 'OU

# Replace "NA" with NaN for missing values
outage_cleaned.replace("NA", np.nan, inplace=True)

# Display the cleaned DataFrame
outage_cleaned.head()
```

Out[ ]:

	4	OBS	YEAR	MONTH	U.S._STATE	POSTAL.CODE	NERC.REGION	CLIMATE.REGION	AN
6		1	2011	7	Minnesota	MN	MRO	East North Central	
7		2	2014	5	Minnesota	MN	MRO	East North Central	
8		3	2010	10	Minnesota	MN	MRO	East North Central	
9		4	2012	6	Minnesota	MN	MRO	East North Central	
10		5	2015	7	Minnesota	MN	MRO	East North Central	

5 rows × 54 columns

In [ ]:

Data Visualization

The provided code utilizes the matplotlib and seaborn libraries to create histograms for four key columns in the power outage dataset: 'OUTAGE.DURATION', 'DEMAND.LOSS.MW', 'CUSTOMERS.AFFECTED', and 'RES.PRICE'. Below are some objectives we are attempting to find by illustrating these graphs.

Distribution of OUTAGE.DURATION

Objective:

- Understand the distribution of outage durations to identify common patterns and outliers.

Justification:

- Outage duration is a crucial metric, providing insights into the temporal aspect of power outages.
- A histogram with a kernel density estimate (KDE) allows us to observe the central tendency and spread of outage durations.

Distribution of DEMAND.LOSS.MW

Objective:

- Analyze the distribution of demand loss in terms of megawatts during power outages.

Justification:

- 'DEMAND.LOSS.MW' represents the amount of demand lost during an outage, offering insights into the severity of power disruptions.
- A histogram with KDE enables visualization of the range and frequency of demand losses.

Distribution of CUSTOMERS.AFFECTED

Objective:

- Explore the impact of power outages on the number of customers affected.

Justification:

- 'CUSTOMERS.AFFECTED' provides information on the scale of the outage in terms of affected users.
- A histogram with KDE helps understand the distribution of the number of affected customers.

Distribution of RES.PRICE

Objective:

- Investigate the distribution of residential electricity prices during power outage events.

Justification:

- 'RES.PRICE' represents the residential electricity price, which may correlate with the severity or causes of power outages.
- A histogram with KDE allows us to visualize the distribution of residential electricity prices.

Summary:

These visualizations serve as essential exploratory tools to gain a deeper understanding of key features in the power outage dataset. By analyzing the distributions of relevant columns, we can identify trends, outliers, and potential relationships between variables, providing a foundation for further analysis and hypothesis formulation.

In [ ]:
```
import plotly.graph_objects as go

# Plot distribution of OUTAGE.DURATION
fig1 = go.Figure(data=[go.Histogram(x=outage_cleaned['OUTAGE.DURATION'], nbinsx=30)
```

```
fig1.update_layout(title_text='Distribution of OUTAGE.DURATION', xaxis_title='Durat
fig1.show()
fig1.write_html(r'Assets\Distribution_of_OUTAGE.html', include_plotlyjs='cdn')

# Plot distribution of DEMAND.LOSS.MW
fig2 = go.Figure(data=[go.Histogram(x=outage_cleaned['DEMAND.LOSS.MW'], nbinsx=30)]
fig2.update_layout(title_text='Distribution of DEMAND.LOSS.MW', xaxis_title='Demand
fig2.show()

# Plot distribution of CUSTOMERS.AFFECTED
fig3 = go.Figure(data=[go.Histogram(x=outage_cleaned['CUSTOMERS.AFFECTED'], nbinsx=
fig3.update_layout(title_text='Distribution of CUSTOMERS.AFFECTED', xaxis_title='Nu
fig3.show()

# Plot distribution of RES.PRICE
fig4 = go.Figure(data=[go.Histogram(x=outage_cleaned['RES.PRICE'], nbinsx=30)])
fig4.update_layout(title_text='Distribution of RES.PRICE', xaxis_title='Residential
fig4.show()
```

Exploratory Data Analysis (EDA) Visualizations

Uncovering Patterns and Comparisons

In this section, we delve into visualizations aimed at uncovering patterns and making insightful comparisons within the power outage dataset. Utilizing scatter plots and box plots, we explore relationships between outage duration and affected customers, as well as variations in customer impact across different NERC regions. These visualizations serve as a foundational step in understanding the nuances and potential factors influencing power outage events.

Scatter Plot of OUTAGE.DURATION vs. CUSTOMERS.AFFECTED

Description: This scatter plot is created to visually assess the potential correlation or pattern between the duration of power outages ('OUTAGE.DURATION') and the number of customers affected ('CUSTOMERS.AFFECTED'). Each point on the plot represents a specific power outage event, with the x-axis indicating the duration of the outage in minutes and the y-axis representing the number of affected customers. By examining the distribution of points, one can infer whether longer durations tend to result in a higher number of affected customers or if there are other patterns worth exploring.

Box Plot of NERC.REGION vs. CUSTOMERS.AFFECTED

Description: This box plot is designed to compare the distribution of the number of customers affected ('CUSTOMERS.AFFECTED') across different NERC (North American Electric Reliability Corporation) regions. Each box represents the interquartile range (IQR) of the data

for a specific region, with the median indicated by the horizontal line inside the box. Outliers are also displayed as individual points. By examining these box plots, one can gain insights into the variations in the impact of power outages on customers across different NERC regions.

In [ ]:
```
# Create a scatter plot
fig1 = px.scatter(outage_cleaned, x='OUTAGE.DURATION', y='CUSTOMERS.AFFECTED', titl
fig1.update_xaxes(title_text='OUTAGE.DURATION (minutes)')
fig1.update_yaxes(title_text='CUSTOMERS.AFFECTED')
fig1.show()
fig1.write_html(r'Assets\scatter.html', include_plotlyjs='cdn')

# Create a box plot
fig2 = px.box(outage_cleaned, x='NERC.REGION', y='CUSTOMERS.AFFECTED', title='Box P
fig2.update_xaxes(title_text='NERC.REGION')
fig2.update_yaxes(title_text='CUSTOMERS.AFFECTED')
fig2.show()
fig2.write_html(r'Assets\box_plot.html', include_plotlyjs='cdn')
```

Customer Impact Analysis

Aggregating Customer Impact Data (Groupby)

Description: In this section, the code groupes the data by 'State' and calculates the average number of customers affected ('CUSTOMERS.AFFECTED') for each state. This provides insights into the regional variations in the impact of power outages. The result is displayed in a DataFrame, showcasing the average customer impact for each state.

Aggregating Customer Impact Data (Pivot Table)

Description: In this section, the code groupes the data by 'State' and calculates the average number of customers affected ('CUSTOMERS.AFFECTED') for each state. This provides insights into the regional variations in the impact of power outages. The result is displayed in a DataFrame, showcasing the average customer impact for each state.

In [ ]:
```
# Group by 'State' and calculate the average 'CUSTOMERS.AFFECTED' for each year
average_customers_by_state= outage_cleaned.groupby("U.S._STATE")["CUSTOMERS.AFFECTE
average_customers_by_state = average_customers_by_state.to_frame().reset_index()

# Display the result
print("Average Customers Affected by State:")
average_customers_by_state
```

```
Average Customers Affected by State:
```

Out[ ]:

	U.S._STATE	CUSTOMERS.AFFECTED
0	Alabama	94328.800000
1	Alaska	14273.000000
2	Arizona	64402.666667
3	Arkansas	47673.846154
4	California	201365.716535
5	Colorado	41060.636364
6	Connecticut	60339.230769
7	Delaware	3475.000000
8	District of Columbia	194709.222222
9	Florida	289369.090909
10	Georgia	120680.187500
11	Hawaii	147237.200000
12	Idaho	5833.333333
13	Illinois	207027.159091
14	Indiana	69551.441176
15	Iowa	94000.000000
16	Kansas	108000.000000
17	Kentucky	130531.000000
18	Louisiana	151003.100000
19	Maine	54839.352941
20	Maryland	120534.866667
21	Massachusetts	77983.400000
22	Michigan	152878.244444
23	Minnesota	124006.571429
24	Mississippi	5000.000000
25	Missouri	50611.076923
26	Montana	NaN
27	Nebraska	87070.666667
28	Nevada	22220.000000
29	New Hampshire	13869.818182

	U.S._STATE	CUSTOMERS.AFFECTED
30	New Jersey	160216.806452
31	New Mexico	166666.666667
32	New York	190675.866667
33	North Carolina	99624.833333
34	North Dakota	34500.000000
35	Ohio	136782.611111
36	Oklahoma	160683.210526
37	Oregon	43958.583333
38	Pennsylvania	168536.823529
39	South Carolina	251913.125000
40	South Dakota	NaN
41	Tennessee	59317.352941
42	Texas	223232.095745
43	Utah	10227.727273
44	Vermont	0.000000
45	Virginia	149429.058824
46	Washington	101944.022727
47	West Virginia	179794.333333
48	Wisconsin	45876.000000
49	Wyoming	11833.333333

In [ ]:
```
# Pivot the data to examine average 'CUSTOMERS.AFFECTED' by 'CLIMATE.REGION' and 'C
pivot_table = outage_cleaned.pivot_table(values='CUSTOMERS.AFFECTED', index='CLIMAT

# Display the pivot table
print("\nAverage Customers Affected by Climate Region and Cause Category:")
pivot_table
```

```
Average Customers Affected by Climate Region and Cause Category:
```

Out[ ]:

CAUSE.CATEGORY	equipment failure	fuel supply emergency	intentional attack	islanding	public appeal	
CLIMATE.REGION						
Central	87750.000000	0.0	110.714286	9666.666667	NaN	14
East North Central	NaN	NaN	660.111111	0.000000	7600.000000	13
Northeast	28575.750000	0.5	1055.580247	0.000000	18600.000000	16
Northwest	46651.500000	NaN	92.592593	0.000000	4000.000000	16
South	62721.666667	NaN	1042.833333	14500.000000	4917.636364	22
Southeast	145420.200000	NaN	0.000000	NaN	0.000000	20
Southwest	55666.666667	0.0	327.423077	35230.000000	NaN	8
West	198608.142857	0.0	14060.000000	5039.192308	NaN	36
West North Central	NaN	NaN	0.000000	NaN	34500.000000	1

Enhancing Exploration with Folium: Geospatial Analysis

In this section, we aim to enhance the exploration of power outage effects by incorporating interactive geospatial analysis using Folium. The provided code includes a function to retrieve latitude and longitude information for each U.S. state. By applying this function to the 'U.S._STATE' column, the dataset is enriched with geographical coordinates. Rows with missing latitude and longitude information are then removed to ensure accuracy in the geospatial analysis.

Now that the latitude and longitude information is available, we can proceed to create interactive Folium maps to visualize and analyze the effects of power outages by state.

In [ ]:
```
# Initialize the geolocator
geolocator = Nominatim(user_agent="my_geocoder")

# Function to get latitude and longitude
def get_lat_lon(location):
    try:
        location = geolocator.geocode(location)
        return (location.latitude, location.longitude)
    except:
        return None
```

```
# Apply the function to the 'U.S._STATE' column
average_customers_by_state['Latitude, Longitude'] = average_customers_by_state['U.S

# Split the 'Latitude, Longitude' column into separate 'Latitude' and 'Longitude' c
average_customers_by_state[['Latitude', 'Longitude']] = pd.DataFrame(average_custom

# Drop the 'Latitude, Longitude' column
average_customers_by_state = average_customers_by_state.drop('Latitude, Longitude',

# Display the resulting DataFrame
average_customers_by_state
```

Out[ ]:

	U.S._STATE	CUSTOMERS.AFFECTED	Latitude	Longitude
0	Alabama	94328.800000	33.258882	-86.829534
1	Alaska	14273.000000	64.445961	-149.680909
2	Arizona	64402.666667	34.395342	-111.763275
3	Arkansas	47673.846154	35.204888	-92.447911
4	California	201365.716535	36.701463	-118.755997
5	Colorado	41060.636364	38.725178	-105.607716
6	Connecticut	60339.230769	41.650020	-72.734216
7	Delaware	3475.000000	38.692045	-75.401331
8	District of Columbia	194709.222222	38.893847	-76.988043
9	Florida	289369.090909	27.756767	-81.463983
10	Georgia	120680.187500	32.329381	-83.113737
11	Hawaii	147237.200000	19.593801	-155.428370
12	Idaho	5833.333333	43.644764	-114.015407
13	Illinois	207027.159091	40.079661	-89.433729
14	Indiana	69551.441176	40.327013	-86.174693
15	Iowa	94000.000000	41.921673	-93.312270
16	Kansas	108000.000000	38.273120	-98.582187
17	Kentucky	130531.000000	37.572603	-85.155141
18	Louisiana	151003.100000	30.870388	-92.007126
19	Maine	54839.352941	45.709097	-68.859020
20	Maryland	120534.866667	39.516240	-76.938207
21	Massachusetts	77983.400000	42.378877	-72.032366
22	Michigan	152878.244444	43.621195	-84.682435
23	Minnesota	124006.571429	45.989659	-94.611329
24	Mississippi	5000.000000	32.971528	-89.734850
25	Missouri	50611.076923	38.760481	-92.561787
26	Montana	NaN	47.375267	-109.638757
27	Nebraska	87070.666667	41.737023	-99.587382
28	Nevada	22220.000000	39.515882	-116.853722
29	New Hampshire	13869.818182	43.484913	-71.655399

	U.S._STATE	CUSTOMERS.AFFECTED	Latitude	Longitude
30	New Jersey	160216.806452	40.075738	-74.404162
31	New Mexico	166666.666667	34.580207	-105.996048
32	New York	190675.866667	40.712728	-74.006015
33	North Carolina	99624.833333	35.672964	-79.039292
34	North Dakota	34500.000000	47.620146	-100.540737
35	Ohio	136782.611111	40.225357	-82.688140
36	Oklahoma	160683.210526	34.955082	-97.268406
37	Oregon	43958.583333	43.979280	-120.737257
38	Pennsylvania	168536.823529	40.969989	-77.727883
39	South Carolina	251913.125000	33.687439	-80.436374
40	South Dakota	NaN	44.647176	-100.348761
41	Tennessee	59317.352941	35.773008	-86.282008
42	Texas	223232.095745	31.263890	-98.545612
43	Utah	10227.727273	39.422519	-111.714358
44	Vermont	0.000000	44.599072	-72.500261
45	Virginia	149429.058824	37.123224	-78.492772
46	Washington	101944.022727	38.895037	-77.036543
47	West Virginia	179794.333333	38.475841	-80.840841
48	Wisconsin	45876.000000	44.430898	-89.688464
49	Wyoming	11833.333333	43.170026	-107.568534

Visualizing Customer Impact with Folium Heatmap

This code snippet utilizes Folium to create an interactive heatmap visualizing the impact of power outages based on the number of affected customers. The map is centered on Minnesota, serving as the default location due to the order of the Excel file. The heatmap layer is constructed using latitude, longitude, and the 'CUSTOMERS.AFFECTED' column. Each point on the map contributes to the heatmap intensity, providing a spatial representation of the customer impact.

In [ ]:
```
# Remove Nan entries
average_customers_by_state = average_customers_by_state[~average_customers_by_state
```

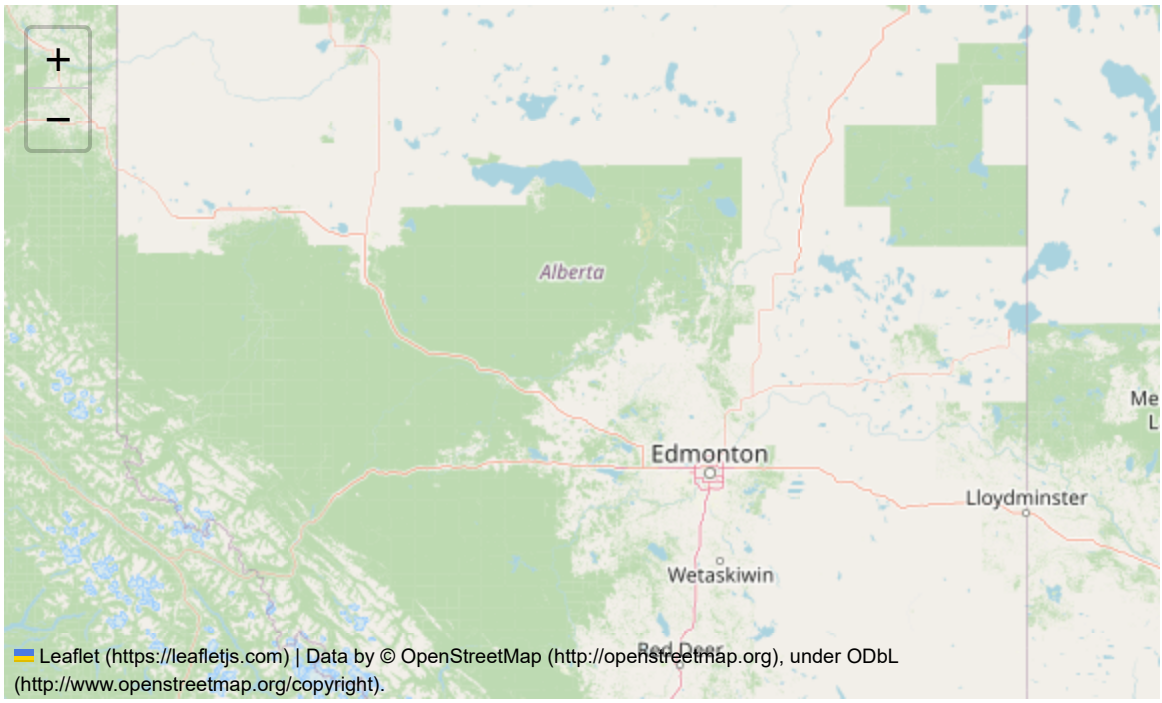
```
# Create a Map instance
m = folium.Map(location=[46.7296, -94.6859], zoom_start=6)  # Location coordinates

# Prepare data for the heatmap
data = average_customers_by_state[['Latitude', 'Longitude', 'CUSTOMERS.AFFECTED']].

# Add the heatmap to the map
HeatMap(data).add_to(m)

# Display the map
m
```

Out[ ]:



Average Outage Duration per State

In this section, we're interested in understanding how the duration of power outages varies across different states. To do this, we first calculate the average outage duration for each state using the groupby function in pandas. This gives us a new DataFrame, avg_outage_duration, where each row corresponds to a state and the 'OUTAGE.DURATION' column contains the average outage duration for that state.

Next, we create a bar plot using Plotly Express (px.bar). The x-axis of the plot represents the different states, and the y-axis represents the average outage duration. Each bar in the plot corresponds to a state. The height of the bar represents the average outage duration for that state.

By visualizing the data in this way, we can easily compare the average outage duration across different states.

In [ ]:
```
# Calculate the average outage duration per state
avg_outage_duration = outage_cleaned.groupby('U.S._STATE')['OUTAGE.DURATION'].mean(

# Create a bar plot
fig = px.bar(avg_outage_duration, x='U.S._STATE', y='OUTAGE.DURATION',
             labels={'U.S._STATE':'State', 'OUTAGE.DURATION':'Average Outage Durati
             title='Average Outage Duration per State')

# Show the plot
fig.show()
fig.write_html(r'Assets\avg_outage_duration.html', include_plotlyjs='cdn')
```

Assessment of Missingness

In [ ]:
```
# NMAR Analysis

# We believe the column CAUSE.CATEGORY.DETAIL is likely to be NMAR as the column
# revolves around a detailed description of the event categories, and too complex
# of a description may cause nothing to be marked down instead. Possible data
# that could help make it MAR would be the uniqueness or complexity of the cause
# since more complex causes may not be easily inputed into the data.


# Missingness Dependency

df = pd.read_csv('outage.csv')

def ks_query(missing, dependent):
    mar = df.copy()
    mar['missing'] = mar[missing].isna()
    res = ks_2samp(mar.query('missing')[dependent], mar.query('not missing')[depend
    return res

dur_vs_cust = ks_query('OUTAGE.DURATION', 'CUSTOMERS.AFFECTED')
dur_vs_sales = ks_query('OUTAGE.DURATION', 'TOTAL.SALES')

duration_missing = df.copy()
duration_missing['missing'] = duration_missing['OUTAGE.DURATION'].isna()

px.histogram(duration_missing, x='CUSTOMERS.AFFECTED', color='missing', histnorm='p
             title="customers affected by missingness of outage duration", barmode=

# px.histogram(duration_missing, x='TOTAL.SALES', color='missing', histnorm='probab
#              title="total sales by missingness of outage duration", barmode='over
```

Hypothesis Testing

In [ ]:
```
# Null Hypothesis
# The duration of outages in the years 2005 comes from the same population as 2015.

# Alternative Hypothesis
# The duration of outages in the year 2015 are shorter than the duration of outages
```

```
# Test Statistic
# Difference in group means
```

Permutation Test for Mean Outage Duration (2005 vs. 2015)

In this section, we perform a permutation test to assess whether there is a significant difference in the mean outage duration between the years 2005 and 2015.

Hypotheses

- **Null Hypothesis (H0):** There is no difference in mean outage duration between the years 2005 and 2015.
- **Alternative Hypothesis (H1):** There is a significant difference in mean outage duration between the years 2005 and 2015.

Test Statistic

We choose the difference in mean outage duration between the two years as our test statistic. The observed difference is calculated using the actual data, and we compare this against a distribution of differences obtained by shuffling the outage duration data.

Significance Level

We set the significance level at 0.05, which is a common choice in hypothesis testing.

Permutation Test Procedure

1. **Data Preparation:**
    - We filter the dataset to include only rows from the years 2005 and 2015.
    - Missing values are dropped from the dataset.
2. **Observation:**
    - The observed difference in mean outage duration between 2005 and 2015 is computed.
3. **Permutation Test Loop (500 Repetitions):**
    - In each iteration, the outage duration data is shuffled, and the difference in mean outage duration is calculated.
    - These differences form a distribution under the null hypothesis.

4. **P-value Calculation:**
    - The proportion of permuted mean differences that are greater than or equal to the observed difference is calculated.

Results

The resulting p-value is the probability of observing a difference in mean outage duration as extreme as the observed difference under the assumption that there is no true difference between the years 2005 and 2015.

Conclusion

- **P-value:** 0.0
- **Conclusion:** If the p-value is less than the chosen significance level (0.05), we reject the null hypothesis in favor of the alternative hypothesis, suggesting a significant difference in mean outage duration between the years 2005 and 2015.

Justification

- The choice of a permutation test is appropriate when the assumptions of parametric tests are not met, or the distribution of the data is unknown.
- The test statistic (difference in means) aligns with the question of interest, comparing the average outage duration between the two years.
- A significance level of 0.05 is commonly used and provides a balance between Type I and Type II errors.

In [ ]:
```
n_repetitions = 500

shuffled = df[(df['OUTAGE.START'].str.startswith('2005')) | (df['OUTAGE.START'].str
shuffled['old_year'] = shuffled['OUTAGE.START'].str.startswith('2005')

observed_difference = shuffled.groupby('old_year')['OUTAGE.DURATION'].mean().diff()

differences = []
for _ in range(n_repetitions):

    with_shuffled = shuffled.assign(Shuffled_Duration=np.random.permutation(shuffle

    group_means = (
        with_shuffled
        .groupby('old_year')
        .mean()
        .loc[:, 'Shuffled_Duration']
    )
    difference = group_means.diff().iloc[-1]
```

```
    differences.append(difference)

(np.array(differences) >= observed_difference).mean()
```

Out[ ]:
```
0.0
```

In [ ]:
```

```