Power Outages

This project uses major power outage data in the continental U.S. from January 2000 to July 2016. Here, a major power outage is defined as a power outage that impacted at least 50,000 customers or caused an unplanned firm load loss of atleast 300MW. Interesting questions to consider include:

- Where and when do major power outages tend to occur?
- What are the characteristics of major power outages with higher severity? Variables to consider include location, time, climate, land-use characteristics, electricity consumption patterns, economic characteristics, etc. What risk factors may an energy company want to look into when predicting the location and severity of its next major power outage?
- What characteristics are associated with each category of cause?
- How have characteristics of major power outages changed over time? Is there a clear trend?

Getting the Data

The data is downloadable here.

A data dictionary is available at this article under *Table 1. Variable descriptions*.

Cleaning and EDA

- Note that the data is given as an Excel file rather than a CSV. Open the data in Excel or another spreadsheet application and determine which rows and columns of the Excel spreadsheet should be ignored when loading the data in pandas.
- Clean the data.
    - The power outage start date and time is given by `OUTAGE.START.DATE` and `OUTAGE.START.TIME` . It would be preferable if these two columns were combined into one datetime column. Combine `OUTAGE.START.DATE` and `OUTAGE.START.TIME` into a new datetime column called `OUTAGE.START` . Similarly, combine `OUTAGE.RESTORATION.DATE` and `OUTAGE.RESTORATION.TIME` into a new datetime column called `OUTAGE.RESTORATION` .
- Understand the data in ways relevant to your question using univariate and bivariate analysis of the data as well as aggregations.

Hint 1: pandas can load multiple filetypes: `pd.read_csv` *,* `pd.read_excel` *,* `pd.read_html` *,* `pd.read_json` *, etc.*

Hint 2: `pd.to_datetime` *and* `pd.to_timedelta` *will be useful here.*

Tip: To visualize geospatial data, consider Folium or another geospatial plotting library.

Assessment of Missingness

- Assess the missingness of a column that is not missing by design.

Hypothesis Test

Find a hypothesis test to perform. You can use the questions at the top of the notebook for inspiration.

Summary of Findings

Introduction

In this project I performed an analysis of data relation to major power outages in the U.S. The data on the major power outages included outages in the years from January 2000 to July 2016 and included information like start date and time, restoration date and time, state location, urban and rural population information, and wattage information among other data. To determine what factors would be helpful for predicting future outages, I believe that first one must figure out when the majority of outages occur, the most frequent outage locations and what makes those locations different from others before looking into more detailed data.

In this project I will be answering the following questions:

- Which states have the most major power outages?
- Are there certain months that can be predicted to have more major power outages?
- Do states with a larger percentage of urban population have more major power outages?

Cleaning and EDA

Cleaning

- I read in the Excel file and immediately dropped the first columns, `Major power outage events in the continental U.S.` , as it was simply the title of the dataset and no relevant to the data itself. To clean the data, I dropped the first four rows of the dataset which had all missing values. The missing values were due to the excel file have a few empy rows at the top for readbility.

- I then set the columns to the variables in the now-first row of my dataset and dropped that row and the row containing units just below that in order to get just the data for each column. I then set the `OBS` column as the index.
- To fully clean the dataset, I checked the relevant rows for any type compatability issues and created two news rows: `OUTAGE.START`, which combined the information from the `OUTAGE.START.DATE` and `OUTAGE.START.TIME` columns and `OUTAGE.RESTORATION`, which combined the information from the `OUTAGE.RESTORATION.DATE` and `OUTAGE.RESTORATION.TIME` columns.
- I also made sure the `YEAR` column was of type `int` and the `POPPCT_URBAN` column was of type `float` as I will need to use these columns later.

EDA

- I plotted the number of outages per year by state using a line graph. This allowed me to determine my cutoff for hwo many power outages is a lot compared to other states and to see if there were any patterns in the data, like certain years having more outages than others.
  - There were quite a few instances where some states had more than 10 major outages in a year, so I chose that as my cutof for 'a lot' of outages. I then took that cutoff and found the states that had more than 10 outages in a given year. The top outage states with greater than 10 major outages per year were Washington, California, New York, Texas, Florida, Maryland, Michigan, Delaware, Pennsylvania, New Jersey, Ohio, and Utah.
  - California showed up the most, 8 times in a 9 year period, with Texas showing up 5 times and Washington showing up 3 times. Washington had the most major power outages in a single year, with 29 in 2011. In fact, in 2011, 9 out of 50 states had more than 10 major power outages!
- As far as interesting aggregates, I took a look at which months had the most outages and found that June, July and August are the months with the most power outages! This can be explained pretty easily, because during the hottest months people use their A/C a lot more, putting more strain on power grids and causing more widespread power outages. The increased use of power also can cause wildfires, which also cause power outages.
- For my univariate analysis, I plotted a kde plot of urban population percentage and the results were surprising! The plot was bimodal as there were two peaks, one around about 72% and one around 89%. The average is just above 80% and the graph as a whole seems to skew to the left. The drop off at 100% makes sense as there is no such

thing as population percentage being over 100%.

Assessment of Missingness

The `MONTH` column had missing data, and a quick look at the years where `MONTH` information was missing showed nearly all of the missing data was from 2000. With a quick permutation test I was able to confirm that the data was Missing At Random (MAR) with a p-value of 6.181061884547034e-08 and depends upon the `YEAR` column. With that same permutation test I was also able to prove that the `MONTH` column does not depend on the `POSTAL.CODE` column with a p-value of 0.7064996657865272.

The `OUTAGE.START.DATE` column also had missing values and this I attributed to be Not Missing At Random (NMAR), because even though the dates were only missing when the month was also missing, it actually shows that the data to formulate the date itself was unavailable, and therefore the values for the date itself was missing and it was not due to a different column.

Hypothesis Test

For my hypothesis test, I wanted to see if my suspicions are correct that states that have a higher urban population have more major power outages. To do so I formulated by hypotheses as follows:

- Null Hypothesis: Urban population percentage does not affect amount of power outages.
- Alternative Hypothesis: States with higher urban population have more power outages.

To test my hypothesis, I got the average urban population percentage for all states and used that to distinguish my lower urban population states from my higher urban population states (below the averge is lower, above the averge is higher). I used the difference in group means as my test statistic since the distributions are quantitative in nature and my hypothesis requires a check of not just the differences but also if they are higher or lower.

Using 3000 repetitions of a permutation test, I found the difference in group means and compared each difference to the observed. I got a p-value of 0.0 and compared it with my significance level of 0.01 (chosen so my answer would be as conservative as possible), therefore rejecting my null hypothesis. It is possible that the test I did is not the best test to determine the acceptance of rejection of my hypothesis and in the future I would have liked to have had specific cities within states to further pinpoint if the entire state is more at risk for outages or just certain high-density cities.

Code

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
import seaborn as sns
import datetime as dt
%matplotlib inline
%config InlineBackend.figure_format = 'retina'  # Higher resolution figures
```

Cleaning and EDA

In [ ]:
```
# TODO
outage = pd.read_excel('data/outage.xlsx')
outage = outage.drop(columns=['Major power outage events in the continental U

outage = outage.drop(index=[0,1,2,3], axis=0) # drop np.nan rows befor column
outage.reset_index(inplace=True) # reset index bc of drop above
outage = outage.drop(columns=['index']) # drop the index col made from reset
outage.columns = outage.iloc[0] # set cols to first row
outage = outage.drop(index=[0,1]) # drop variables row (was just made column
outage = outage.set_index('OBS') # set observed outages as index bc its the s

# columns
outage['YEAR'] = outage['YEAR'].astype(int) # year has no np.nan, make all in
# outage['ANOMALY.LEVEL'] = outage['ANOMALY.LEVEL'].astype(float) # level sho
outage['POPPCT_URBAN'] = outage['POPPCT_URBAN'].astype(float)

def to_date(date, time):
    if type(date) != dt.datetime or type(time) != dt.time:
        return np.nan
    else:
        combine = date.combine(date, time)
        return combine.strftime('%Y-%m-%d %H:%M:%S')

# start date and time
listy = []
for x,y in zip(outage['OUTAGE.START.DATE'], outage['OUTAGE.START.TIME']):
    listy.append(to_date(x,y))
outage['OUTAGE.START'] = listy

# restoration date and time
listy2 = []
for x1,y1 in zip(outage['OUTAGE.RESTORATION.DATE'], outage['OUTAGE.RESTORATIO
    listy2.append(to_date(x1,y1))
outage['OUTAGE.RESTORATION'] = listy2

outage.head() # cleaned dataset
```

Out[ ]:

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

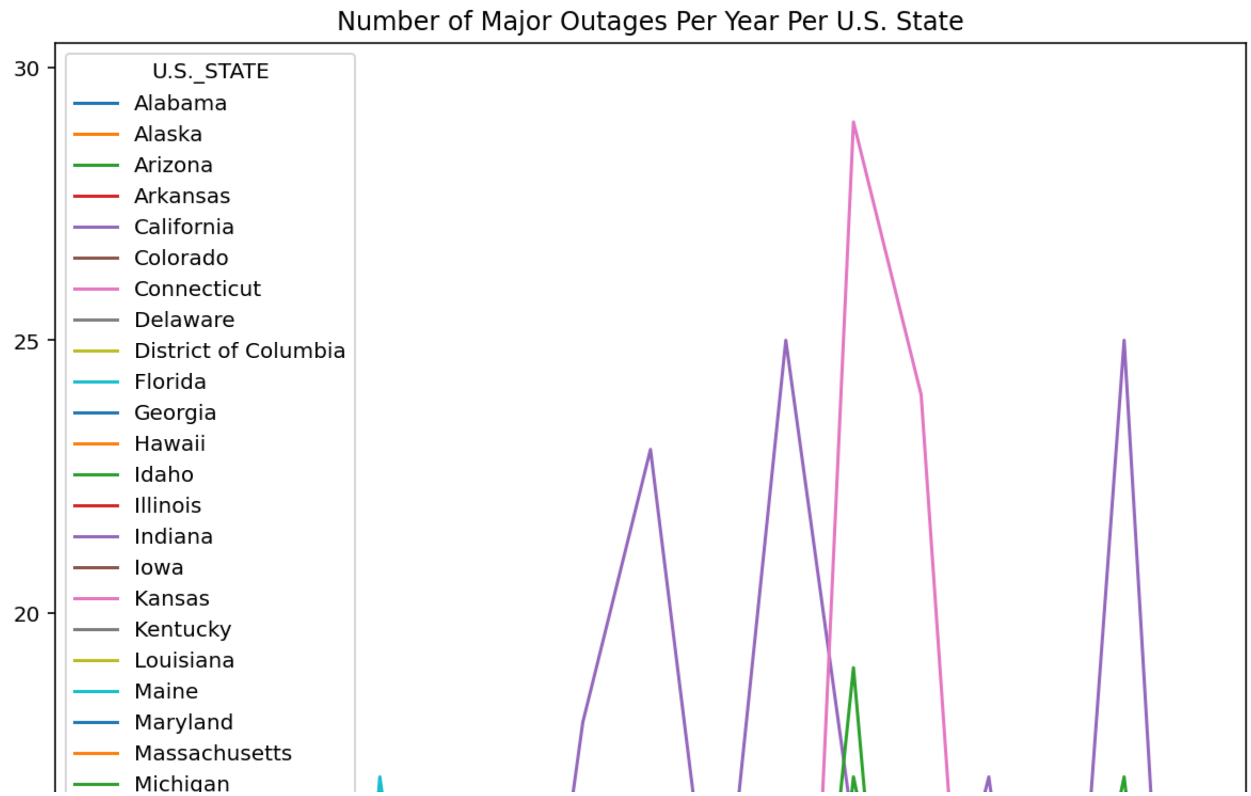
5 rows × 57 columns

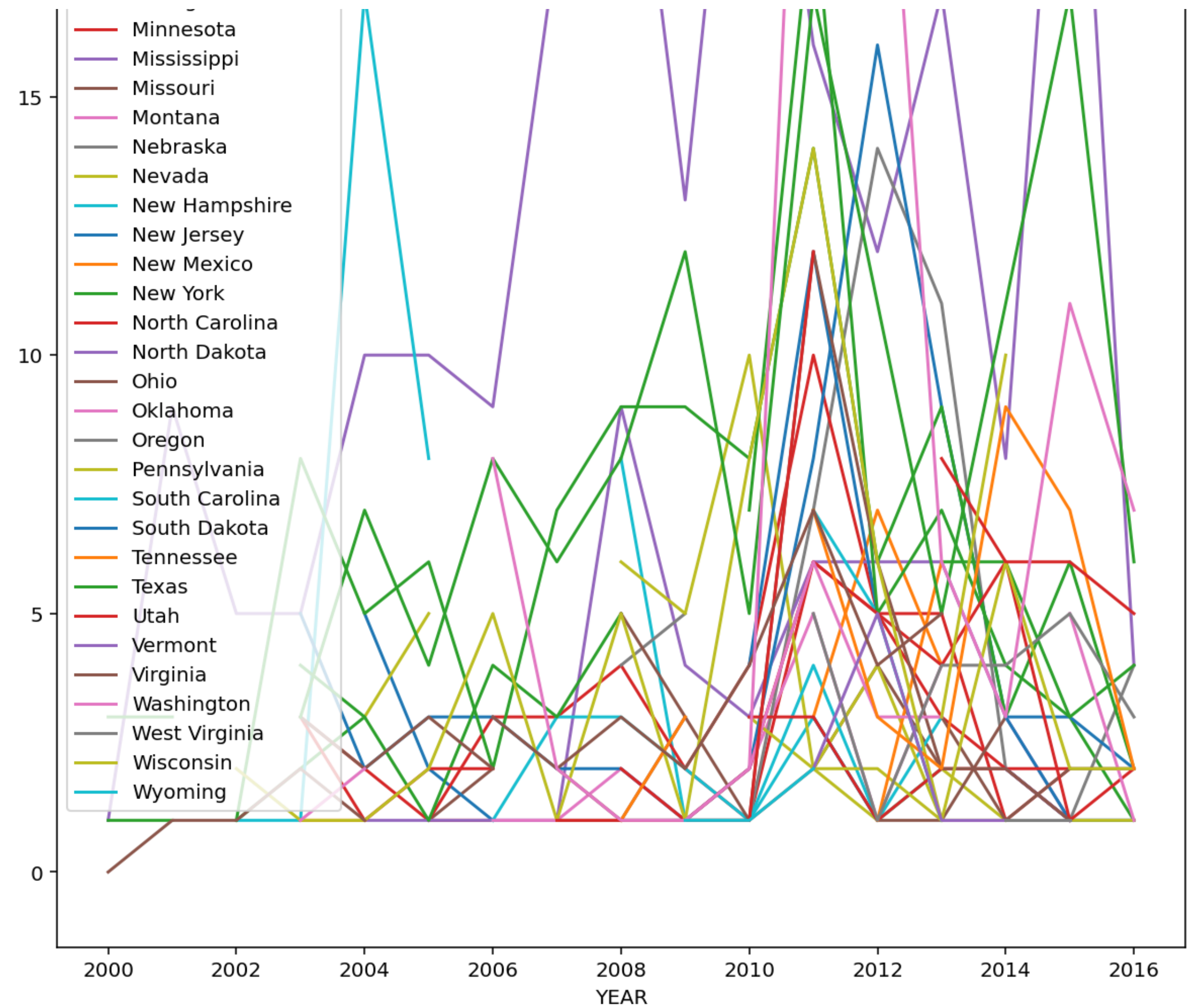
Now that our data is cleaned, let's move on to EDA!

Power Outages Per Year Per State - Bivariate

In [ ]:

```
ct = outage.groupby(['YEAR', 'U.S._STATE']).count() # group by year, state
c = outage.pivot_table(index='YEAR', columns='U.S._STATE', values='MONTH', ag
fig = c.plot(kind='line', figsize=(10, 15));
t = 'Number of Major Outages Per Year Per U.S. State'
fig.set_title(t);
```





There seem to be quite a few instances where states have more than 10 major outages in a year. Let's see if we can identify these.

In [ ]:
```
amt = outage.groupby(['YEAR', 'U.S._STATE']).count()[['MONTH']].sort_values(b
am = amt[amt > 10] # amt outages > 10
am[am['MONTH'].notna()].sort_index() # sort by year, then my state
```

Out[ ]:

YEAR	U.S._STATE	MONTH
2004	Florida	17.0
2007	California	18.0
2008	California	23.0
2009	California	13.0
	Texas	12.0
2010	California	25.0
2011	California	16.0
	Michigan	14.0
	New Jersey	12.0
	New York	19.0
	Ohio	12.0
	Pennsylvania	14.0
	Texas	17.0
	Utah	12.0
	Washington	29.0
2012	California	12.0
	Delaware	14.0
	Maryland	16.0
	Texas	11.0
	Washington	24.0
2013	California	17.0
	Delaware	11.0
2014	Texas	11.0
2015	California	25.0
	Texas	17.0
	Washington	11.0

We can see the top outage states with greater than 10 major outages per year are Washington, California, New York, Texas, Florida, Maryland, Michigan, Delaware, Pennsylvania, New Jersey, Ohio, and Utah.

California shows up the most, 8 times in a 9 year period, with Texas showing up 5 times and Washington showing up 3 times. Washington has had the most major power outages in a single year, with 29 in 2011.

In fact, in 2011, 9 out of 50 states had more than 10 major power outages!

Are There Certain Months Where Outages Are More Likely to Happen? - Interesting Aggregates

In [ ]:
```
# are there certain months outages are more likely to happen?
# sort values from most outages to least by month
outage.groupby(['U.S._STATE', 'MONTH']).count().pivot_table(index='MONTH', co
```

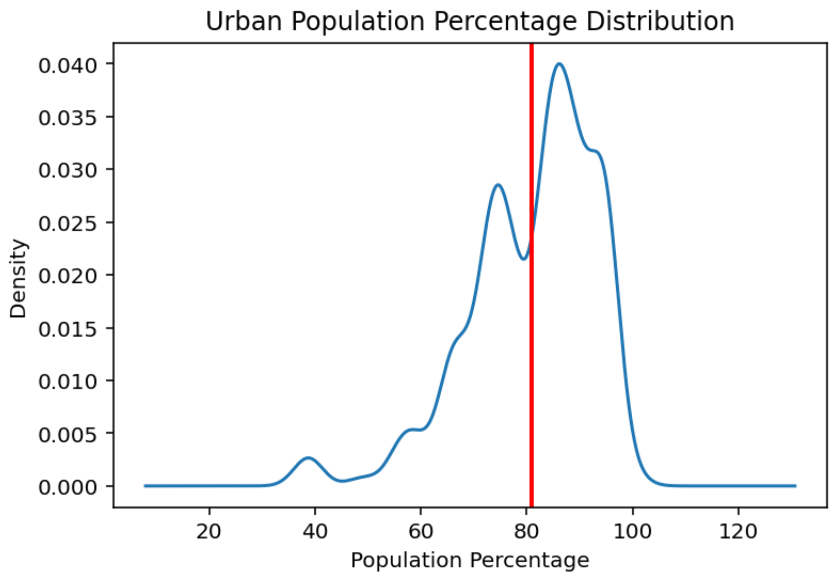
Out[ ]:
```
MONTH
6      195
7      181
8      153
2      136
1      136
5      127
12     111
4      111
10     109
3      100
9       94
11      72
dtype: int64
```

June, July and August are the months with the most power outages! This could have been predicted, because during the hottest months people use their A/C a lot more, putting more strain on power grids and causing more widespread power outages. The increased use of power also can cause wildfires, which also cause power outages.

Urban Population Percentage - Univariate

What does the distribution of urban population percentages look like across all states?

In [ ]:
```
# plot kde of urban pop percent
fig = outage['POPPCT_URBAN'].plot(kind='kde');
plt.axvline(x=outage['POPPCT_URBAN'].mean(), color='red', linewidth=2, label=
fig.set_title('Urban Population Percentage Distribution');
fig.set_xlabel('Population Percentage');
```



The plot of urban population percentage looks bimodal as there are two peaks, one around about 72 and one around 89. The average is just above 80 and the graph as a whole seems to skew to the left. The drop off at 100 makes sense as there is no such thing as population percentage being over 100%.

Assessment of Missingness

In [ ]:

```
# TODO
oo = outage.copy()
oo[oo['MONTH'].isna()].head() # look at rows where month is missing
```

Out[ ]:

OBS	YEAR	MONTH	U.S._STATE	POSTAL.CODE	NERC.REGION	CLIMATE.REGION	ANOMALY.L
240	2000	NaN	Texas	TX	FRCC	South	
340	2000	NaN	Alabama	AL	SERC	Southeast	
366	2000	NaN	Illinois	IL	SERC	Central	
767	2000	NaN	North Carolina	NC	SERC	Southeast	
888	2000	NaN	Delaware	DE	RFC	Northeast	

5 rows × 59 columns

In [ ]:
```
from scipy.stats import ks_2samp
def missing(out): # use permutation test to test if mcar or mar
    df = out[['YEAR','MONTH', 'POSTAL.CODE']].copy()
    pval = []
    pval2 = []
    df['missing'] = df['MONTH'].isna() # make missing col

    for _ in range(500):
        little = df[['YEAR','missing']]
        little2 = df[['POSTAL.CODE','missing']]

        #  p-value

        pv = ks_2samp(little[little['missing'] == False]['YEAR'],little[littl
        pv2 = ks_2samp(little2[little2['missing'] == False]['POSTAL.CODE'],li
        pval.append(pv)
        pval2.append(pv2)

    return [np.mean(pval), np.mean(pval2)]

m = missing(outage)
if m[0] < 0.01:
    print(f'YEAR: MAR, p-val: {m[0]}')
else:
    print(f'YEAR: MCAR, p-val: {m[0]}')

if m[1] < 0.01:
    print(f'POSTAL.CODE: MAR, p-val: {m[1]}')
else:
    print(f'POSTAL.CODE: MCAR, p-val: {m[1]}')
```

```
YEAR: MAR, p-val: 6.181061884547034e-08
POSTAL.CODE: MCAR, p-val: 0.7064996657865272
```

According to the missingness permuation test, the data is MAR, so the `MONTH` column missingness is in fact impacted by the `YEAR` column. Since a large majority of the missing `MONTH` values appear in the year 2000, we can safely attirbute the missingness to the year and not missing completely at random.

In [ ]:
```
outage[['YEAR', 'MONTH', 'U.S._STATE', 'OUTAGE.START.DATE', 'OUTAGE.START.TIM
```

Out[ ]:

OBS	YEAR	MONTH	U.S._STATE	OUTAGE.START.DATE	OUTAGE.START.TIME	OUTAGE.START
240	2000	NaN	Texas	NaN	NaN	NaN
340	2000	NaN	Alabama	NaN	NaN	NaN
366	2000	NaN	Illinois	NaN	NaN	NaN
767	2000	NaN	North Carolina	NaN	NaN	NaN
888	2000	NaN	Delaware	NaN	NaN	NaN

If we take a look at the `OUTAGE.START.DATE` and `OUTAGE.START.TIME`, both pieces of data are not available. We can see that `MONTH` is also missing, but not `YEAR`, so it makes sense that if `MONTH` is missing from the date, the `OUTAGE.START.DATE` will also be empty because the full date is unavailable. And with the date being unavailable, `OUTAGE.START` will also be missing. Therefore the missing data in `OUTAGE.START` is NMAR because the missingness relies on not having the full date information (this could be argued MAR as well, but I feel that my explanation for NMAR fits best).

Hypothesis Test

In [ ]:
```
# TODO
# Null Hypothesis: Urban population percentage does not affect amount of powe
# Alt Hypothesis: States with higher urban population have more power outages

urban = outage.groupby('U.S._STATE')[['POPPCT_URBAN']].median()
ct = outage.groupby('U.S._STATE')[['YEAR']].count()

merged = urban.merge(ct, left_index=True, right_index=True)
merged.columns = ['POPPCT_URBAN', 'NUM.OUTAGES']
avg_urban = np.round(outage['POPPCT_URBAN'].mean() / 1)

# 2 distributions
# below avg -- lower urban pop
# above avg -- higher urban pop

def test(data):
    n_repetitions = 3000
    mean_differences = []

    to_shuffle = data.copy()
    num_outages = to_shuffle['NUM.OUTAGES'].values

    observed = data.groupby('HIGHER.URBAN')['NUM.OUTAGES'].mean().iloc[-1]

    for _ in range(n_repetitions):

        # Step 1: Shuffle the weights
        shuffled = np.random.permutation(num_outages)
        to_shuffle['Shuffled Num Outages'] = shuffled

        # test statistic
        group_means = (
            to_shuffle
            .groupby('HIGHER.URBAN')
            .mean()
            .loc[:, 'Shuffled Num Outages']
        )
        difference = group_means.diff().iloc[-1]
        mean_differences.append(difference)

    means = pd.Series(mean_differences)
    pval = (means >= observed).sum() / n_repetitions
    return pval


merged['HIGHER.URBAN'] = merged['POPPCT_URBAN'] > avg_urban # make higher urb

# check p-value with results against cutoff
if test(merged) < 0.01:
    print('Reject the Null, ', f'pvalue: {test(merged)}')
else:
    print('Accept the Null, ', f'pvalue: {test(merged)}')
```

```
Reject the Null,  pvalue: 0.0
```

In [ ]: