# 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 (https://engineering.purdue.edu/LASCI/research-data/outages/outagerisks)](https://engineering.purdue.edu/LASCI/research-data/outages/outagerisks).

A data dictionary is available at this [article (https://www.sciencedirect.com/science/article/pii/S2352340918307182)](https://www.sciencedirect.com/science/article/pii/S2352340918307182) 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 (https://python-visualization.github.io/folium/)](https://python-visualization.github.io/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

For this project, we are doing analysis towards outage dataset. The dataset contains a time period from January 2000 to July 2016, providing a lot of outage cases with different climate region, causes and factors. Since there are too many factors, only a part of them are useful to make hypothesis test. Thus, we do the following steps to clean up data and filter these important factors to make hypothesis test, finding the relationships between certain factors.

### Cleaning and EDA

We cleaned up the data to make entire dataset more readable by deleting useless data columns, reform 2 columns contains the information about OUTAGE time and filter out unrelated rows. Then based on cleaned dataset, we did Univariate Analysis, Bivariate Analysis and Interesting Aggregates to find out the relationships between certain variables. For example, we did univariate analysis to 'YEAR', 'MONTH', 'U.S._STATE', 'CLIMATE REGION' and 'CAUSE CATEGORY', plotting them using methods. Then we did bivariate analysis to 'CLIMATE.REGION' with'U.S._STATE', 'CLIMATE.REGION' with 'CAUSE CATEGORY', and 'CLIMATE.REGION' with 'OUTAGE DURATION', trying to use plotting for find their relationships. At last, we did interesting aggregates to 'CLIMATE.REGION','OUTAGE.DURATION','YEAR'.

### Assessment of Missingness

For this part, We identify some nmar columns and study the rest of the data, we pick the column "CAUSE.CATEOGORY" specifically to anaylsis its dependency with other columns. More specifically, we pick columns ANOMALY.LEVEL, ANOMALY.LEVEL, POPDEN_UC and POPDEN_RURAL to study their relationship which means determine if the missingness in "CAUSE.CATEGORY" depends on these four columns. The conclusion we make in here is that the missingness of CAUSE.CATEGORY is dependents with ANOMALY.LEVEL, POPDEN_UC and POPDEN_RURAL but not with ANOMALY.LEVEL

### Hypothesis Test

Our hypothesis is that Null hypothesis: The probability that an outage occurs in the South during 2011 is equal to the probability that it occurs in the West during 2011 with a significant value of 0.05. After we do the hypothesis testing, we find out that we need to reject the null hypothesis since the p_value we get is around 0.012 which is lower than the p value

# Code

In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
import seaborn as sns
%matplotlib inline
%config InlineBackend.figure_format = 'retina'  # Higher resolution figures
```

## Cleaning and EDA

Read the original table, reform it and then clean up all the useless info stored in columns and rows. Combine 4 required columns.

In [2]:
```
# read the table
df = pd.read_excel('outage.xlsx')
df
```

Out[2]:

| | Major power outage events in the continental U.S. | Unnamed: 1 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 | Unnamed: 5 | Unnamed: 6 |
|---|---|---|---|---|---|---|---|
| **0** | Time period: January 2000 - July 2016 | NaN | NaN | NaN | NaN | NaN | NaN |
| **1** | Regions affected: Outages reported in this dat... | NaN | NaN | NaN | NaN | NaN | NaN |
| **2** | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **3** | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **4** | variables | OBS | YEAR | MONTH | U.S._STATE | POSTAL.CODE | NERC.REGION |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **1535** | NaN | 1530 | 2011 | 12 | North Dakota | ND | MRO |
| **1536** | NaN | 1531 | 2006 | NaN | North Dakota | ND | MRO |
| **1537** | NaN | 1532 | 2009 | 8 | South Dakota | SD | RFC |
| **1538** | NaN | 1533 | 2009 | 8 | South Dakota | SD | MRO |
| **1539** | NaN | 1534 | 2000 | NaN | Alaska | AK | ASCC |

1540 rows × 57 columns

In [3]:
```
#Combine OUTAGE.START.DATE and OUTAGE.START.TIME into a new datetime column ca
lled OUTAGE.START
#Make a copy from original table
out = df.copy()
#Remove unrelated rows and reset the index
out = out.drop(index = [0,1,2,3,5]).reset_index(drop=True)
#Rename the columns
out.columns = out.iloc[0]
#Remove unrelated rows
out = out.drop(index = [0])
#Remove unrelated columns
out = out.drop(columns=['variables'])
#Combine OUTAGE.START.DATE and OUTAGE.START.TIME into a new datetime column ca
lled OUTAGE.START
out['OUTAGE.START'] = pd.to_datetime(out['OUTAGE.START.DATE']).astype(str) + "
" +out['OUTAGE.START.TIME'].astype(str)
out['OUTAGE.START'] = out['OUTAGE.START'].replace('nannan',np.NaN)
out['OUTAGE.START'] = out['OUTAGE.START'].replace('NaT nan',np.NaN)
out['OUTAGE.START'] = pd.to_datetime(out['OUTAGE.START'])

#combine OUTAGE.RESTORATION.DATE and OUTAGE.RESTORATION.TIME into a new dateti
me column called OUTAGE.RESTORATION.
out['OUTAGE.RESTORATION'] = pd.to_datetime(out['OUTAGE.RESTORATION.DATE']).ast
ype(str) + " " +out['OUTAGE.RESTORATION.TIME'].astype(str)
out['OUTAGE.RESTORATION'] = out['OUTAGE.RESTORATION'].replace('nannan',np.NaN)
out['OUTAGE.RESTORATION'] = out['OUTAGE.RESTORATION'].replace('NaT nan',np.NaN
)
out['OUTAGE.RESTORATION'] = pd.to_datetime(out['OUTAGE.RESTORATION'])
out
```

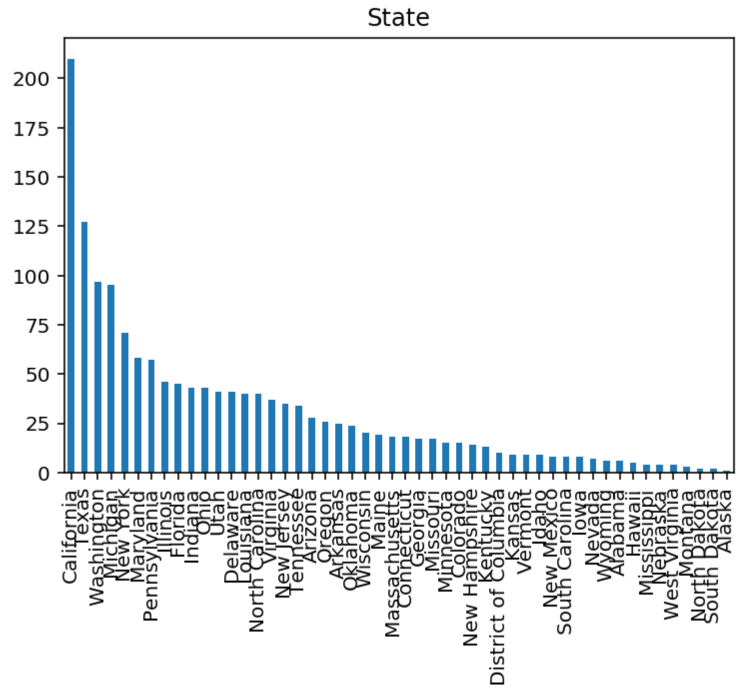
Out[3]:

| | OBS | YEAR | MONTH | U.S._STATE | POSTAL.CODE | NERC.REGION | CLIMATE.REGION | ANO |
|---|---|---|---|---|---|---|---|---|
| **1** | 1 | 2011 | 7 | Minnesota | MN | MRO | East North Central | |
| **2** | 2 | 2014 | 5 | Minnesota | MN | MRO | East North Central | |
| **3** | 3 | 2010 | 10 | Minnesota | MN | MRO | East North Central | |
| **4** | 4 | 2012 | 6 | Minnesota | MN | MRO | East North Central | |
| **5** | 5 | 2015 | 7 | Minnesota | MN | MRO | East North Central | |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **1530** | 1530 | 2011 | 12 | North Dakota | ND | MRO | West North Central | |
| **1531** | 1531 | 2006 | NaN | North Dakota | ND | MRO | West North Central | |
| **1532** | 1532 | 2009 | 8 | South Dakota | SD | RFC | West North Central | |
| **1533** | 1533 | 2009 | 8 | South Dakota | SD | MRO | West North Central | |
| **1534** | 1534 | 2000 | NaN | Alaska | AK | ASCC | NaN | |

1534 rows × 58 columns

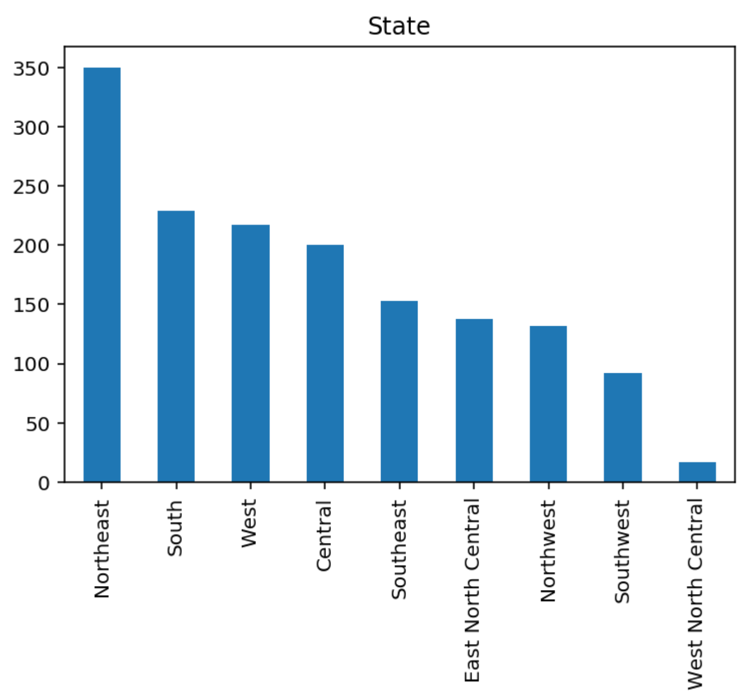
Making a plot to find out the cases happened in each state

In [4]:
```
#Univariate Analysis
#Plot of State
plot = out['U.S._STATE'].value_counts().plot(kind='bar',title='State')
plt.show()
```



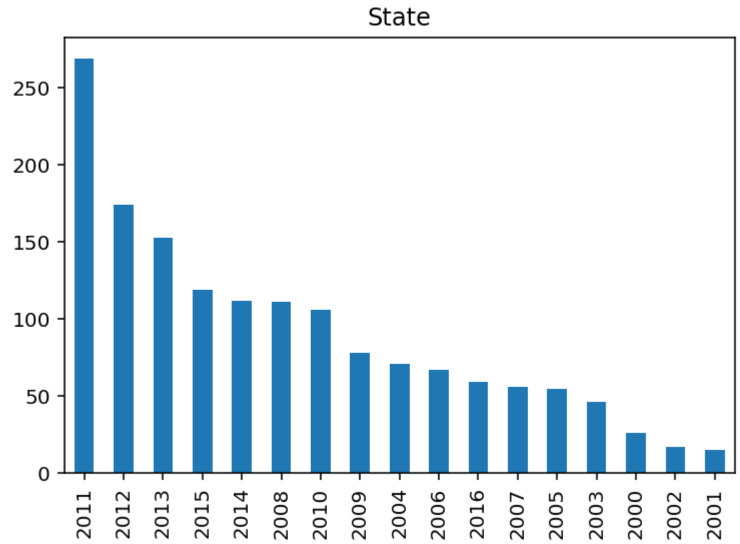

Making a plot to find out the cases happened in each climate region

In [5]:
```
#Plot of Climate Region
plot = out['CLIMATE.REGION'].value_counts().plot(kind='bar',title='State')
plt.show()
```



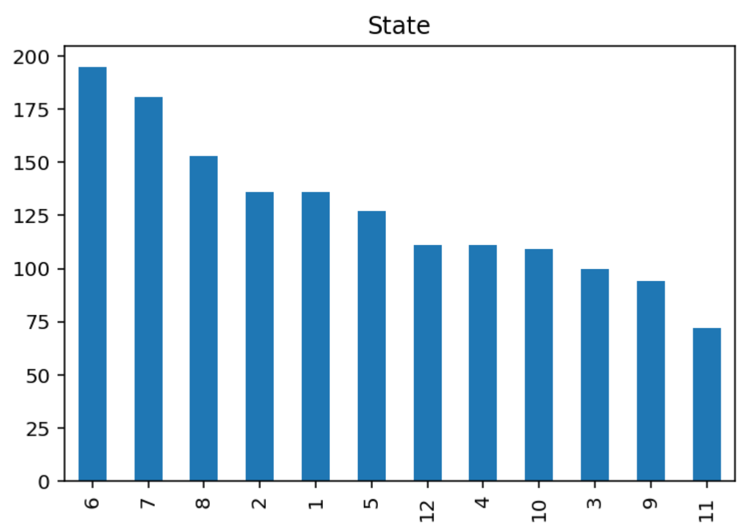
Making a plot to find out the cases happened in each year

In [6]:
```
#Plot of Year
plot = out['YEAR'].value_counts().plot(kind='bar',title='State')
plt.show()
```



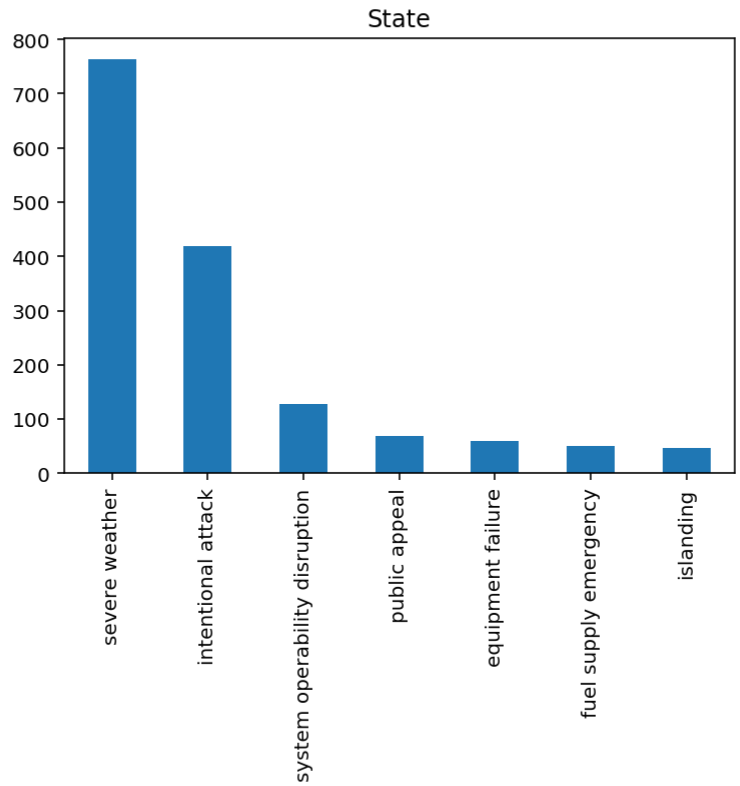

Making a plot to find out the cases happened in each month

In [7]:
```
#Plot of Month
plot = out['MONTH'].value_counts().plot(kind='bar',title='State')
plt.show()
```




Making a plot to find out the cases happened in each cause category

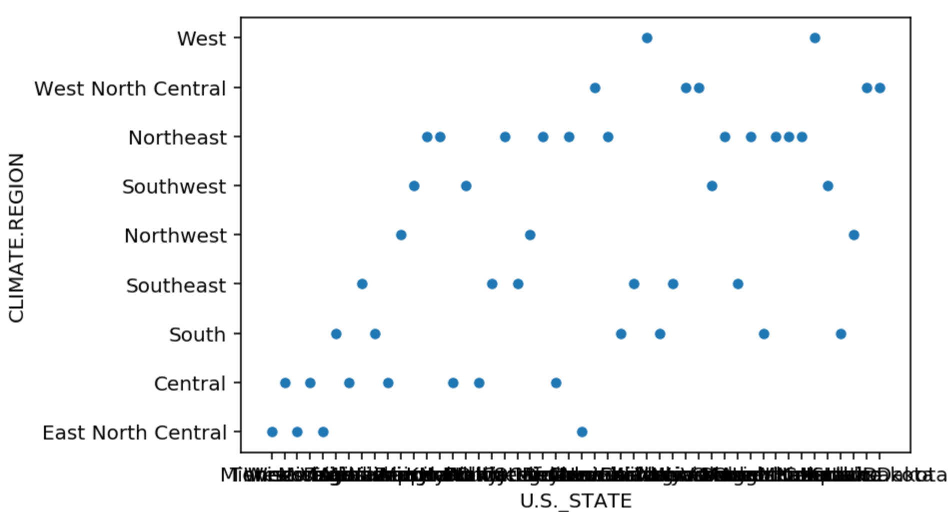
In [8]:
```
#Plot of Cause Category
plot = out['CAUSE.CATEGORY'].value_counts().plot(kind='bar',title='State')
plt.show()
```




Making a plot to find out the relationship between 2 variables. Find out that each state has totally different climate.

In [9]:
```
#Bivariate Analysis
#Check the relationship between Climate Region and State
cleaned = out[['CLIMATE.REGION','U.S._STATE']].dropna()
#Plot 2 variables using scatter plot
sns.scatterplot(y = cleaned['CLIMATE.REGION'], x = cleaned['U.S._STATE'])
```

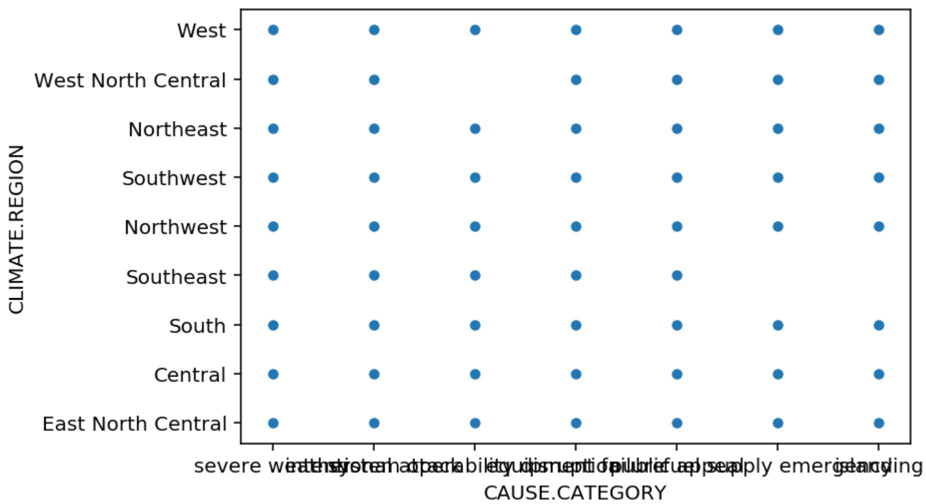
Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2cc567edbc8>
```



Making a plot to find out the relationships between climate region and cause category. The plot shows that cause category in each state equally happened.

In [10]:
```
#Check the relationship between Climate Region and Cause Category
new_cleaned = out[['CLIMATE.REGION','CAUSE.CATEGORY']].dropna()
#Plot these 2 variables using scatter plots
sns.scatterplot(y = new_cleaned['CLIMATE.REGION'], x = new_cleaned['CAUSE.CATE
GORY'])
```

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x2cc56740848>




Making aggregate analysis. Count the OUTAGE.DURATION in each climate region.

In [11]:
```
#Interesting Aggregates
#the relationship between CLIMATE REGION and OUTAGE DURATION
out[['CLIMATE.REGION','OUTAGE.DURATION']].dropna().groupby('CLIMATE.REGION').s
um()
```

Out[11]:

| | OUTAGE.DURATION |
|---|---|
| **CLIMATE.REGION** | |
| **Central** | 515916 |
| **East North Central** | 733230 |
| **Northeast** | 1029130 |
| **Northwest** | 156709 |
| **South** | 620450 |
| **Southeast** | 332653 |
| **Southwest** | 137820 |
| **West** | 333808 |
| **West North Central** | 11145 |

Making a pivot table to show that in each year, the sum up duration in each climate region. Values are so fluctuant.

In [12]:
```
#Selecting 3 variables and make a pivot table
out[['CLIMATE.REGION','OUTAGE.DURATION','YEAR']].dropna().pivot_table(index =
['CLIMATE.REGION'],values = 'OUTAGE.DURATION',columns=['YEAR'],aggfunc=np.sum)
```

Out[12]:

| YEAR | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 20 |
|---|---|---|---|---|---|---|---|---|---|
| **CLIMATE.REGION** | | | | | | | | | |
| **Central** | 1200.0 | NaN | 15420.0 | 11057.0 | 17940.0 | 42139.0 | 11375.0 | 12126.0 | 8873 |
| **East North Central** | NaN | NaN | 3600.0 | 73785.0 | 27260.0 | 56129.0 | 13500.0 | 32406.0 | 5401 |
| **Northeast** | 681.0 | 597.0 | 9390.0 | 61525.0 | 16843.0 | 19616.0 | 80993.0 | 23475.0 | 7193 |
| **Northwest** | NaN | NaN | NaN | 8028.0 | 9720.0 | NaN | 72593.0 | 8316.0 | 44 |
| **South** | 2709.0 | 11747.0 | 20040.0 | 10429.0 | 40548.0 | 58832.0 | 3265.0 | 33566.0 | 18281 |
| **Southeast** | 32304.0 | 241.0 | 2921.0 | 5993.0 | 87898.0 | 94825.0 | 3015.0 | 1201.0 | 1878 |
| **Southwest** | 66.0 | NaN | NaN | 135.0 | 99058.0 | NaN | 2579.0 | 283.0 | 87 |
| **West** | NaN | 5224.0 | 15143.0 | 43060.0 | 10913.0 | 14062.0 | 20143.0 | 14807.0 | 4121 |
| **West North Central** | NaN | NaN | NaN | NaN | 4.0 | NaN | 9600.0 | NaN | 6 |

## Assessment of Missingness

below is our dataframe after cleaning up

In [26]:
```
pd.set_option('display.max_columns', None)
out
```

Out[26]:

| | OBS | YEAR | MONTH | U.S._STATE | POSTAL.CODE | NERC.REGION | CLIMATE.REGION | ANO |
|---|---|---|---|---|---|---|---|---|
| **1** | 1 | 2011 | 7 | Minnesota | MN | MRO | East North Central | |
| **2** | 2 | 2014 | 5 | Minnesota | MN | MRO | East North Central | |
| **3** | 3 | 2010 | 10 | Minnesota | MN | MRO | East North Central | |
| **4** | 4 | 2012 | 6 | Minnesota | MN | MRO | East North Central | |
| **5** | 5 | 2015 | 7 | Minnesota | MN | MRO | East North Central | |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **1530** | 1530 | 2011 | 12 | North Dakota | ND | MRO | West North Central | |
| **1531** | 1531 | 2006 | NaN | North Dakota | ND | MRO | West North Central | |
| **1532** | 1532 | 2009 | 8 | South Dakota | SD | RFC | West North Central | |
| **1533** | 1533 | 2009 | 8 | South Dakota | SD | MRO | West North Central | |
| **1534** | 1534 | 2000 | NaN | Alaska | AK | ASCC | NaN | |

1534 rows × 58 columns

Q1. Nmar columns has month since CLIMATE.REGION, CLIMATE.CATEGORY, MONTH.

CLIMATE.REGION, since some area is hard to determine which region it belongs to, mightbe at the edge of two region, hence it does not shows in the data to make it mar, we need more accurate geographic location.

MONTH is nmar since this missing in it relates to much more details about the weather, the climate, the global politic machine warrentship and other relationship hence it's undetermine by all of the data given in the table

CLIMATE.CATEGORY is nmar since the missingness is not determined by other columns, missingness in this column might be due to hard to determined whethere the weather is hot or normal when it's only a little hotter than usual.

In [ ]:

## Two column analysis

For Assessment of Missingness question 2, I will pick the column CAUSE.CATEOGORY to study their behavior and analysis them with permutation test below is my permutation test

In [28]:
```
def per(outage,col,check_dep):#permutation test method using tvds
    distr = (
        outage
        .assign(is_null=outage[check_dep].isnull())
        .pivot_table(index='is_null', columns=col, aggfunc='size',fill_value =
0)
        .apply(lambda x:x / x.sum(), axis=1)
    )
    #determine the obeservation result
    obs = distr.iloc[-1].abs().sum() / 2
    #setting up with 500 repetition
    n_repetitions = 500
    tvds = []
    for i in range(n_repetitions):
        shuffled_col = (
            outage[col]
            .sample(replace=False, frac=1)
            .reset_index(drop=True)
        )
        #shuffled the column we are trying to study,

        shuffled = (
            outage
            .assign(**{
                col: shuffled_col,
                'is_null': outage[check_dep].isnull()
            })
        )
        #insert a column of 'is_null'
        shuffled = (
            shuffled
            .pivot_table(index='is_null', columns=col, aggfunc='size',fill_val
ue=0)
            .apply(lambda x:x / x.sum(), axis=1)
        )
        #get the tvds for each shuffled and append it into a list
        tvd = shuffled.diff().iloc[-1].abs().sum() / 2
        tvds.append(tvd)
    #return p value
    p_value = np.mean(tvds>obs)
    return p_value
```

In [32]:
```
outage = out.copy()
a=per(outage,'CAUSE.CATEGORY', 'ANOMALY.LEVEL')
b=per(outage,'CAUSE.CATEGORY', 'CLIMATE.REGION')
c=per(outage,'CAUSE.CATEGORY', 'POPDEN_UC')
d=per(outage,'CAUSE.CATEGORY', 'POPDEN_RURAL')
[a,b,c,d]
```

Out[32]:
```
[0.014, 0.096, 0.012, 0.006]
```

In this case, when we set the significant level to be the most common 0.05 since a<0.05 and b >0.05,c<0.05 and d<0.05 We need to reject the hypothesis that cause.category is not dependent on ANOMALY.LEVEL, POPDEN_UC and POPDEN_RURAL which means that they are dependent with each other But we failed to reject that cause.category is not dependent on ANOMALY.LEVEL which means that they are not dependent with each other.

## Hypothesis Test

Null hypothesis: The probability that an outage occurs in the South during 2011 is equal to the probability that it occurs in the West during 2011 p(South|2011)=p(West|2011)

Alternative hypothesis: The probability that an outage occurs in the South during 2011 is not equal to the probability that it occurs in the West during 2011 p(South|2011)!=p(West|2011)

we use a significant level of 0.05

In [80]:
```
table_2011 = out[out['YEAR']==2011]
obs_S2011 = len(table_2011[table_2011['CLIMATE.REGION']=='South']) # obtain th
e observe value of south 2011
obs_W2011 = len(table_2011[table_2011['CLIMATE.REGION']=='West']) # obtain the
observe value of south 2012
total_occurance = obs_S2011+obs_W2011
total_occurance
```

Out[80]:
```
51
```

In [81]:
```
N = 10000

# choose a place evenly for total_occurance time
results = []
for _ in range(N):
    simulation = np.random.choice(['S', 'W'], p = [0.5, 0.5], size = total_occ
urance) #array of Ne or W stands for Northeast or west
    sim_South = (simulation == 'S').sum()  # test stastistic
    results.append(sim_South)
p_value = (pd.Series(results)>obs_N2011).mean() #obtain p value
p_value
```

Out[81]:
```
0.0117
```

since the p_value is lower than the significant level, we can reject the null hypothesis that:

The probability that an outage occurs in the South during 2011 is equal to the probability that it occurs in the West during 2011 p(South|2011)=p(West|2011)