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).

A data dictionary is available at this article (https://www.sciencedirect.com/science/article/pii/S2352340918307182) under *Table 1. Variable descriptions*.

Cleaning and EDA (Exploratory Data Analysis)

- 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/) 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

This open-ended project explores a dataset containing information about major power outages in the continental U.S., ranging in a time period starting from January 2000 and ending in July 2016. In this context, a major power outage would be "a power outage that impacted at least 50,000 customers or caused an unplanned firm load loss of at least 300 MW". In my analysis of data, I try to find different correlations and probable causal relationships within our data. Specifically, our line of inquiry is: in what climate and location do higher severity outages tend to occur, and what are probable causes? While we will explore interesting trends found within the data, we will also assess reasons for missingness, performing univariate and bivariate analyses, as well as performing hypothesis tests that will allow us a better understanding of our data.

Cleaning and EDA

As with any other dataset, we must go through the process of data cleaning, meaning the process of preparing our data by removing unnecessary factors and modifying it until it's ready for analysis. When reading in the outage data from an excel spreadsheet into Pandas, we are faced with admittedly very messy data. Since the column names were pushed down to the 5th row, I turned that row into the column names and dropped the first five rows and the first two columns to get a much cleaner representation of our dataset. I then combined OUTAGE.START.DATE & OUTAGE.START.TIME columns to form OUTAGE.START, as well as combining OUTAGE.RESTORATION.DATE & OUTAGE.RESTORATION.TIME columns to form OUTAGE.RESTORATION, and then dropped the original columns. I then decided that a highly severe power outage would be categorized by having the longer outage durations, and that being in the 75th percentile and above would warrant a "higher severity" rating. I made a new boolean column named HIGHER.SEVERITY and added it to my dataframe, so that I could filter my search based on severity later. I did not impute missing values because I will assess reasons for missingness in the next section.

After the cleaning and preparing process, it was time to explore the data more in-depth. I created multiple visualizations to explore the relationships between severe and non-severe power outages in different climates, regions, U.S. states, as well as higstograms that grouped the duration of power outages together, and different causes of power outages for severe vs non-severe outages. Upon viewing the charts and graphs, we can observe that the severity of an outage is roughly proportional among all three climate categories. Looking at severity plotted against region, there is a bit more variability in the severe outages versus the overall amount of outages in each region. The Northeast has the largest counts of outages overall, followed by the South and West. Looking at severe outages by U.S. state, California has the most amount of outages overall, followed by Texas. After gathering these aggregations, I became interested in the reason why the Northeast

had the highest counts of severe outages and outages overall, when it seemed as if areas in the West/South like California, Washington, and Texas had a much larger volume of outages in general.

Assessment of Missingness

When looking for columns that are NMAR, I believe that CAUSE.CATEGORY.DETAIL is an NMAR column. The related column is CAUSE.CATEGORY, and I believe that CAUSE.CATEGORY.DETAIL is not dependent on CAUSE.CATEGORY because while CAUSE.CATEGORY has values that explains the overarching reason why an outage occurred, CAUSE.CATEGORY.DETAIL further explains the specifics of why an outage occurred. However, if the recorder of the data does not know the specifics, then they are less likely to fill in the column, leaving CAUSE.CATEGORY.DETAIL to have NaN values. The reason for missingness would not dependent on another column, but rather the actual missing value.

The outage dataset has many missing values, and I want to categorize the types of missingness of certain columns. I chose to analyze the columns CAUSE.CATEGORY.DETAIL and HURRICANE.NAMES together, as well as CAUSE.CATEGORY.DETAIL and OUTAGE.DURATION together.

Was the missingness of HURRICANE.NAMES MAR dependent on CAUSE.CATEGORY.DETAIL? I chose to set a significance level of 0.05 in order to see if it was. (Any p-value below 0.05 would mean the two columns come from the same distribution, meaning MAR dependency). I performed a permutation test with TVD as my test statistic, performing 1000 simulations and found that my p-value was 0.002, meaning HURRICANE.NAMES was MAR dependent on CAUSE.CATEGORY.DETAIL.

Was the missingness of CAUSE.CATEGORY.DETAIL dependent on OUTAGE.DURATION? I performed a permutation test with TVD as my test statistic, performing 1000 simulations and found that my p-value was 0.287, meaning HURRICANE.NAMES was MAR not dependent on CAUSE.CATEGORY.DETAIL.

Hypothesis Test

For my hypothesis test, I want to find if there is a difference between the "WEST" climate region's proportion of severe outages to all "WEST" outages, and the proportion of severe outages in all U.S. regions to all outages in all U.S. regions. Since I am comparing categorical distributions, the test statistic to be used should be the total variation distance (TVD), and we will set a significance level of 0.05.

Null Hypothesis: The distribution of causes for severe outages between the Northeast and Southwest regions are the same. The alternative hypothesis is that the distribution of causes for severe outages between Northeast and West regions are difference.

Alternative Hypothesis: The alternative hypothesis is that the distribution of causes for severe outages between Northeast and Southwest regions are difference.

After running the permutation test with 1000 simulations, we get a p-value of 0.046, which is less than the significance level. Since it is less than the significance level, it means our results are statistically significant. When results are statistically significant, it means that the results are not

likely to occur by chance but instead can be attributable to a specific cause. We can then reject the null hypothesis. Relating back to our line of inquiry, we can attribute the proportion of severe/non-severe outages to differ based on location, and for causes to differ based on those locations.

Code

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

Cleaning and EDA

In [2]:
```
pd.set_option('display.max_columns', None)
# reading in excel file to dataframe: outages
outages_excel = os.path.join('outage.xlsx')
outages = pd.read_excel(outages_excel)
outages.head(10)
```

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	CLIM/
5	Units	NaN	NaN	NaN	NaN	NaN	NaN	
6	NaN	1	2011	7	Minnesota	MN	MRO	East
7	NaN	2	2014	5	Minnesota	MN	MRO	East
8	NaN	3	2010	10	Minnesota	MN	MRO	East
9	NaN	4	2012	6	Minnesota	MN	MRO	East

Looking at our read-in dataset, we need to clean and organize the rows and columns. Currently, there are some unneeded rows and columns contained within this dataframe. Looking at the differences between the excel data and our Pandas dataframe, our dataframe has the column names listed on the fourth row as values. We can drop the first four rows of the dataframe, and replace the fifth row to be the column names. The first two columns can be ommitted as well.

In [3]:
```
# messy data: drop unneeded columns and rows
outages = outages.drop([0, 1, 2, 3]).drop(columns='Major power outage events \
in the continental U.S.').reset_index(drop=True)
outages.columns = outages.iloc[0]
outages = outages.drop([0, 1]).reset_index(drop=True).drop(columns='OBS')

outages.head(10)
```

Out[3]:

	YEAR	MONTH	U.S._STATE	POSTAL.CODE	NERC.REGION	CLIMATE.REGION	ANOMALY.LEVEL
0	2011	7	Minnesota	MN	MRO	East North Central	-0.3
1	2014	5	Minnesota	MN	MRO	East North Central	-0.1
2	2010	10	Minnesota	MN	MRO	East North Central	-1.5
3	2012	6	Minnesota	MN	MRO	East North Central	-0.1
4	2015	7	Minnesota	MN	MRO	East North Central	1.2
5	2010	11	Minnesota	MN	MRO	East North Central	-1.4
6	2010	7	Minnesota	MN	MRO	East North Central	-0.9
7	2005	6	Minnesota	MN	MRO	East North Central	0.2
8	2015	3	Minnesota	MN	MRO	East North Central	0.6
9	2013	6	Minnesota	MN	MRO	East North Central	-0.2

We can combine OUTAGE.START.DATE and OUTAGE.START.TIME to become another column, OUTAGE.START. We can do the same with OUTAGE.RESTORATION.DATE and OUTAGE.RESTORATION.TIME and combine them into OUTAGE.RESTORATION. To do so, we can typecast the two 'date' and 'time' columns to strings and combine them, and then cast that new column to a datetime column. Then, we can drop the aforementioned columns.

In [4]:

```
# making OUTAGE.START
outage_start = (outages['OUTAGE.START.DATE'].astype(str).str[:-8] + ' ' +
                outages['OUTAGE.START.TIME'].astype(str))
outages['OUTAGE.START'] = pd.to_datetime(outage_start, errors='coerce')

# making RESTORATION.START
outage_restore = (outages['OUTAGE.RESTORATION.DATE'].astype(str).str[:-8] +
                  ' ' + outages['OUTAGE.RESTORATION.TIME'].dropna().astype(str))
outages['OUTAGE.RESTORATION'] = pd.to_datetime(outage_restore, errors='coerce')

# drop columns that have been replaced
outages = outages.drop(columns=['OUTAGE.START.DATE', 'OUTAGE.START.TIME', 'OUTAGE

outages.head(10)
```

Out[4]:

POPDEN_UC	POPDEN_RURAL	AREAPCT_URBAN	AREAPCT_UC	PCT_LAND	PCT_WATER_TOT	PCT
1700.5	18.2	2.14	0.6	91.5927	8.40733	
1700.5	18.2	2.14	0.6	91.5927	8.40733	
1700.5	18.2	2.14	0.6	91.5927	8.40733	
1700.5	18.2	2.14	0.6	91.5927	8.40733	
1700.5	18.2	2.14	0.6	91.5927	8.40733	
1700.5	18.2	2.14	0.6	91.5927	8.40733	
1700.5	18.2	2.14	0.6	91.5927	8.40733	
1700.5	18.2	2.14	0.6	91.5927	8.40733	
1700.5	18.2	2.14	0.6	91.5927	8.40733	
1700.5	18.2	2.14	0.6	91.5927	8.40733	

We have a much more clean and organized dataset. Now, we can take a closer look at the question at hand. We need to self-define power outages with "higher severity". To simplify things and avoid convolution, we can take its meaning to be all power outages that are at or above the 75th percentile of outage duration. We can make a column called 'TOTAL.TIME' that better represents the amount of days and hours that the outage lasted, and we can also make a boolean column called "higher severity" that checks if each entry is severe enough or not to be classified as such.

In [5]:
```
# filter dataset to outages that corresponds to higher severity

outages['TOTAL.TIME'] = outages['OUTAGE.RESTORATION'] - outages['OUTAGE.START']

duration = (outages['TOTAL.TIME'] >= outages['TOTAL.TIME'].quantile(0.75))

outages['HIGHER.SEVERITY'] = duration

outages.head(10)
```

Out[5]:

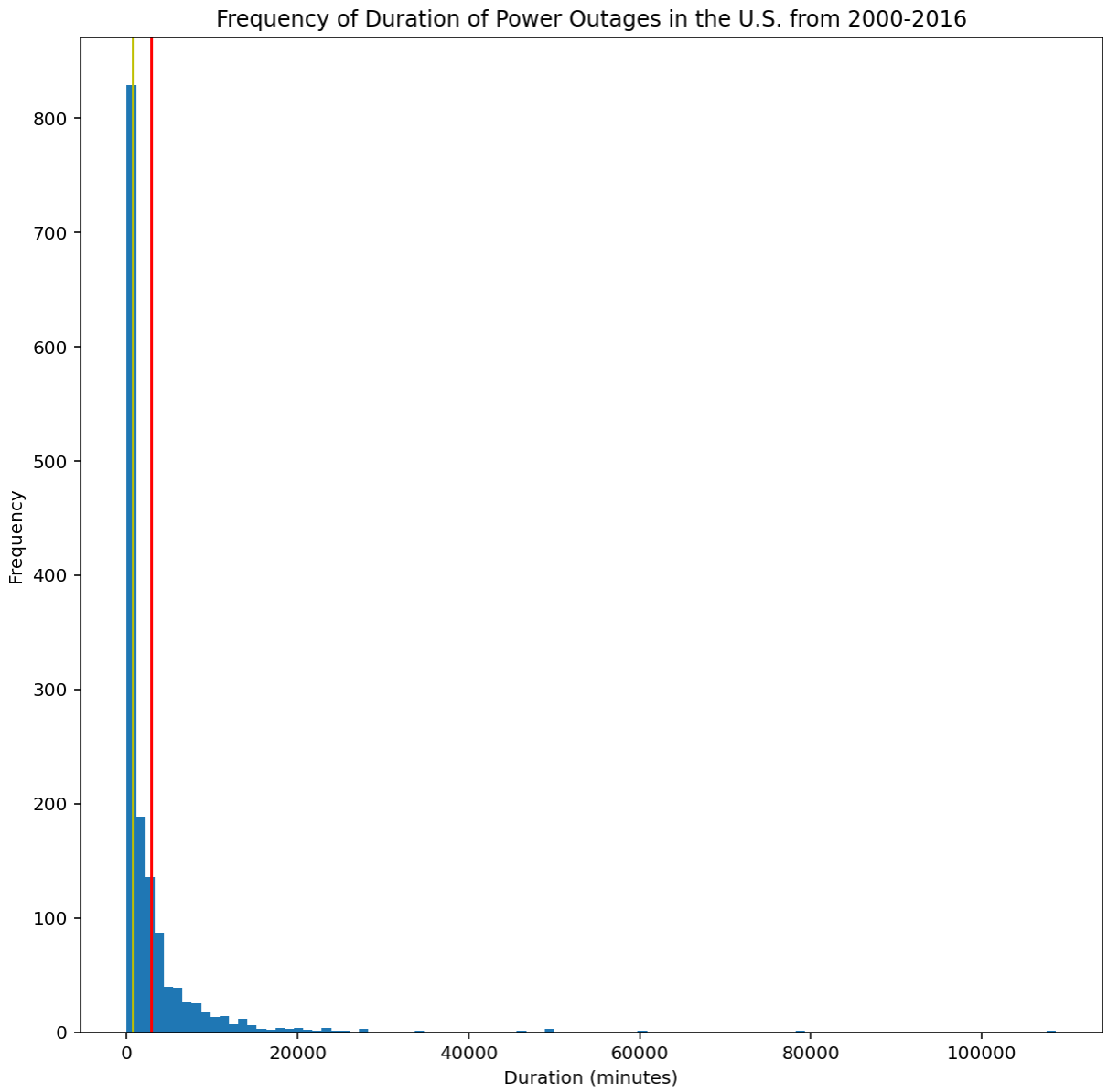
	YEAR	MONTH	U.S._STATE	POSTAL.CODE	NERC.REGION	CLIMATE.REGION	ANOMALY.LEVEL
0	2011	7	Minnesota	MN	MRO	East North Central	-0.3
1	2014	5	Minnesota	MN	MRO	East North Central	-0.1
2	2010	10	Minnesota	MN	MRO	East North Central	-1.5
3	2012	6	Minnesota	MN	MRO	East North Central	-0.1
4	2015	7	Minnesota	MN	MRO	East North Central	1.2
5	2010	11	Minnesota	MN	MRO	East North Central	-1.4
6	2010	7	Minnesota	MN	MRO	East North Central	-0.9
7	2005	6	Minnesota	MN	MRO	East North Central	0.2
8	2015	3	Minnesota	MN	MRO	East North Central	0.6
9	2013	6	Minnesota	MN	MRO	East North Central	-0.2

Now it's time for some exploratory data analysis.

By plotting the values below, we can get a better visualization of the distribution of power outages in the U.S. based on severity (duration). We can see that the bulk of the data lies closer to zero, and the data is skewed to the right. This distribution is far from normal and looks almost like an exponential distribution. We are focused on the data points that lie within the tail, as those will be categorized as severe. The yellow line denotes the median of the data, and the red line denotes the 75th percentile. Everything to the right of the red line is what we are categorizing as "severe".

In [6]:
```
# histogram of outage duration, positive skew with median at 701
# and 75th percentile at 2880

hist = (
    outages['OUTAGE.DURATION']
    .plot(kind = 'hist', title='Frequency of Duration of Power Outages in the U.S
    bins=100, figsize=(10, 10))
)
hist.set_xlabel('Duration (minutes)')

perc_75 = outages['OUTAGE.DURATION'].quantile(0.75)
perc_50 = outages['OUTAGE.DURATION'].quantile(0.50)
plt.axvline(x=perc_75, color='r')
plt.axvline(x=perc_50, color='y')

print('The 75th percentile is when power outages last for: ' + str(perc_75) + ' m
```

```
The 75th percentile is when power outages last for: 2880.0 minutes
```



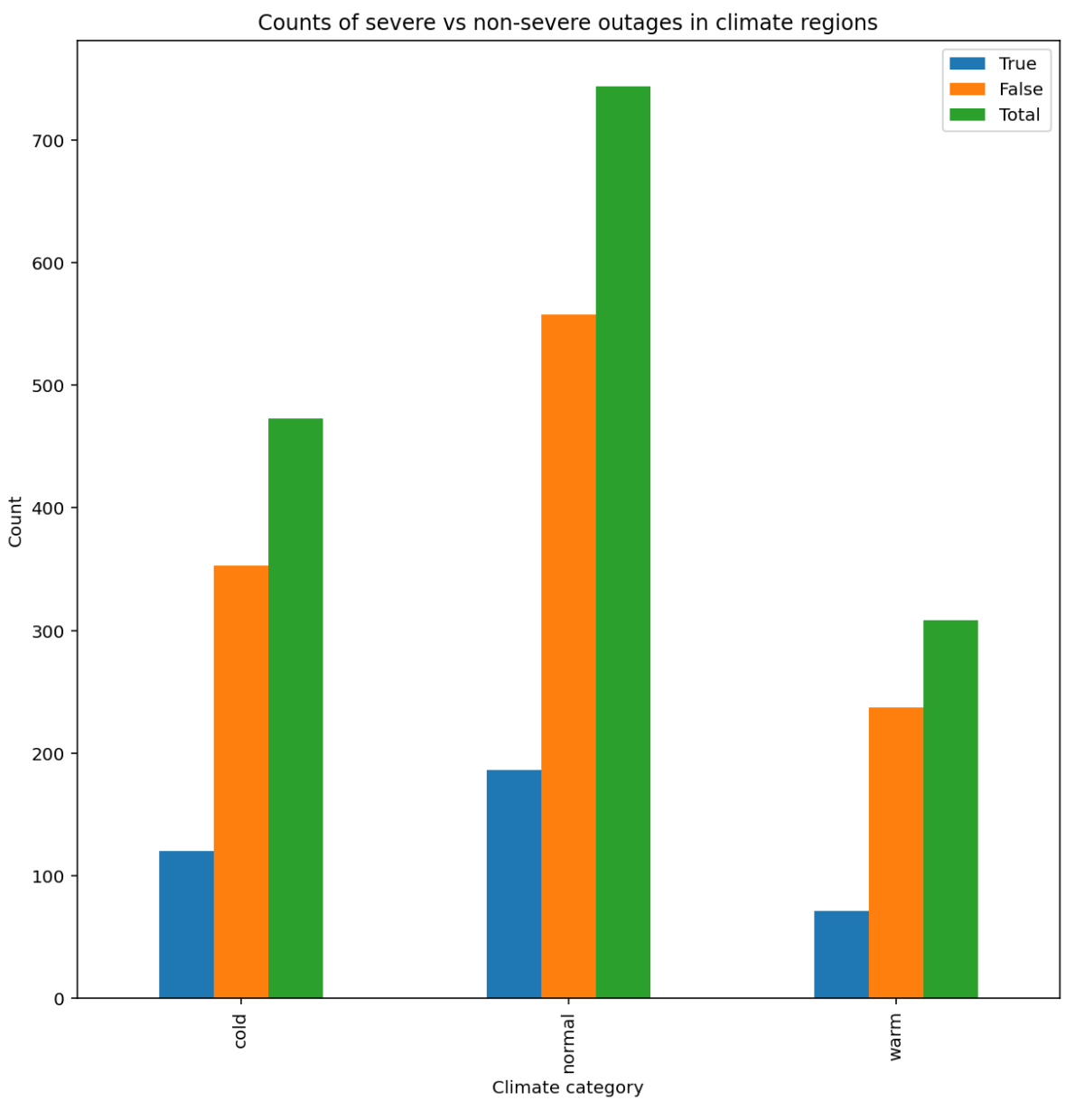
Here I create a grouped bar chart to visualize the proportion of severe and non-severe power outages, against total outages per climate category. Upon a cursory glance, the data appears to be proportional in each climate category, with severe outages corresponding to roughly a fourth of all outages.

In [37]:
```
# comparing the amount of severe power outages to the amount of
# non-severe power outages in different climate categories

severity_true = outages['HIGHER.SEVERITY'] == True
severity_false = outages['HIGHER.SEVERITY'] == False
severity_total = severity_true+severity_false
severity = pd.DataFrame({'True': severity_true, 'False': severity_false,
                         'Total': severity_total,
                         'climate_cat': outages['CLIMATE.CATEGORY']})

plot = (severity.groupby('climate_cat').sum().reset_index()
        .plot.bar(x='climate_cat', figsize=(10, 10),
                  title='Counts of severe vs non-severe outages in climate regior
plot.set_xlabel('Climate category')
plot.set_ylabel('Count')

# pivot table representation of data
category_pt = pd.pivot_table(outages, index=outages['CLIMATE.CATEGORY'],
                             columns=outages['HIGHER.SEVERITY'], aggfunc=len)['YE
category_pt['Total'] = category_pt[False] + category_pt[True]
category_pt
```

```
C:\Users\felic\anaconda3\lib\site-packages\pandas\core\computation\expressions.
py:203: UserWarning: evaluating in Python space because the '+' operator is not
supported by numexpr for the bool dtype, use '|' instead
  warnings.warn(
```

Out[37]:

HIGHER.SEVERITY	False	True	Total
CLIMATE.CATEGORY			
cold	353	120	473
normal	558	186	744
warm	237	71	308

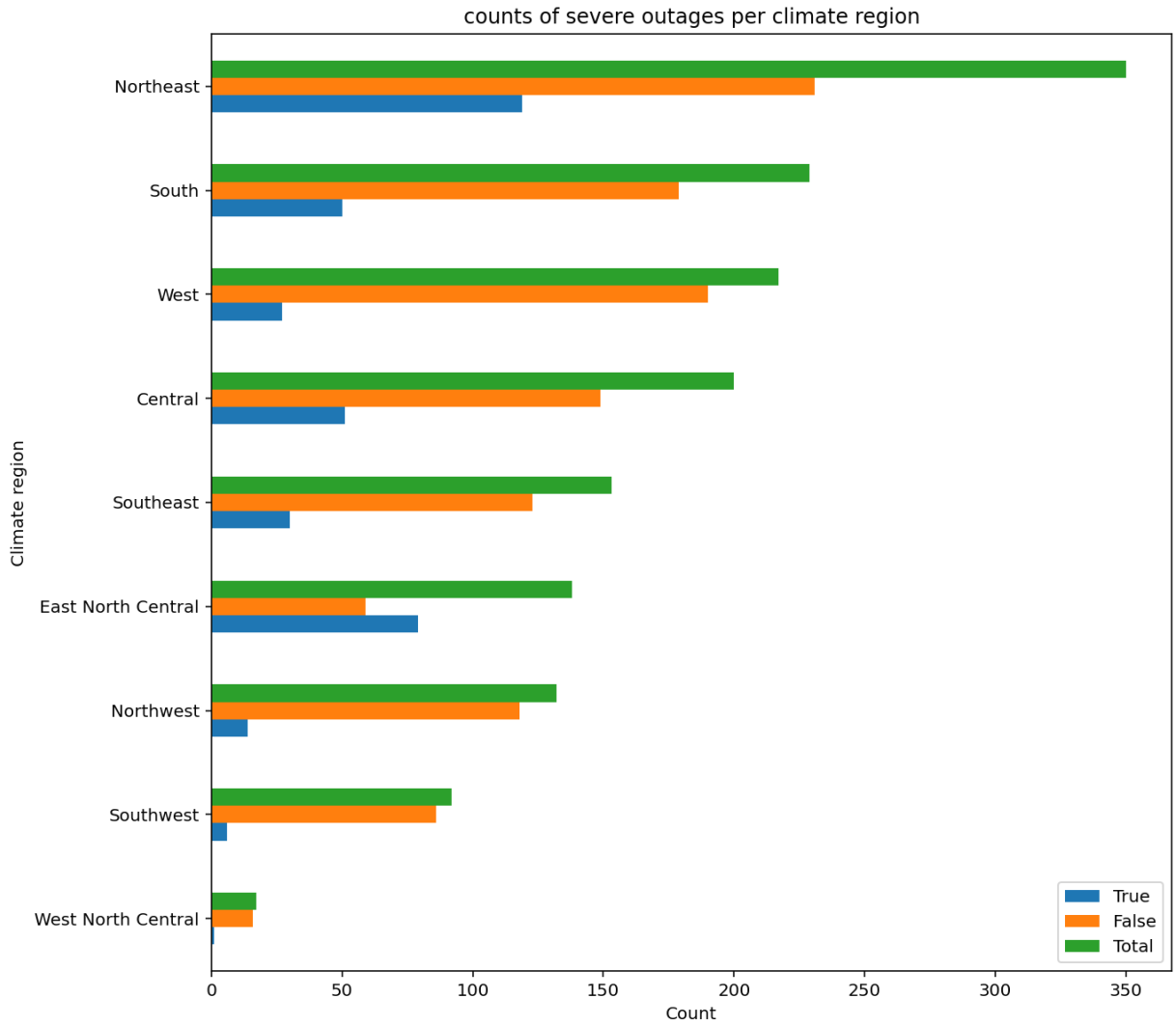


Since we also want to know the location of severe vs. non-severe power outages, I plot another grouped bar chart and sort by the total amount of outages per climate region.

In [38]:
```
he amount of severe power outages to the amount of
 power outages in different climate regions

.DataFrame({'True': severity_true, 'False': severity_false,
            'Total': severity_total,
            'climate_region': outages['CLIMATE.REGION']})


groupby('climate_region').sum().reset_index()
ues(['Total'], ascending=True).plot
climate_region', figsize=(10, 10), title='counts of severe outages per climate re

el('Count')
el('Climate region')

 representation of data
d.pivot_table(outages, index=outages['CLIMATE.REGION'], columns=outages['HIGHER.S
tal'] = region_pt[False] + region_pt[True]
d(10)
```

Out[38]:

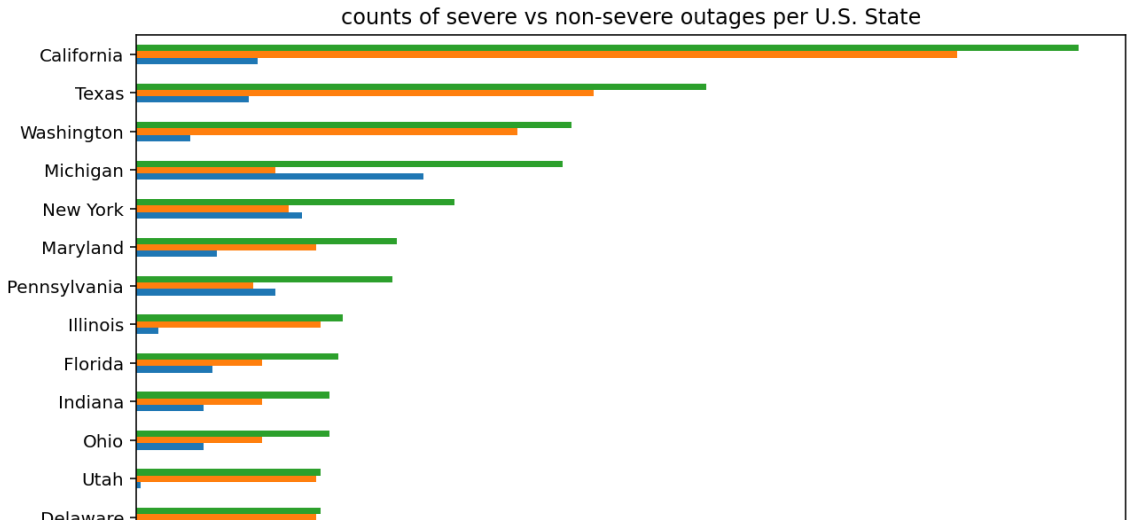
HIGHER.SEVERITY	False	True	Total
CLIMATE.REGION			
Central	149	51	200
East North Central	59	79	138
Northeast	231	119	350
Northwest	118	14	132
South	179	50	229
Southeast	123	30	153
Southwest	86	6	92
West	190	27	217
West North Central	16	1	17

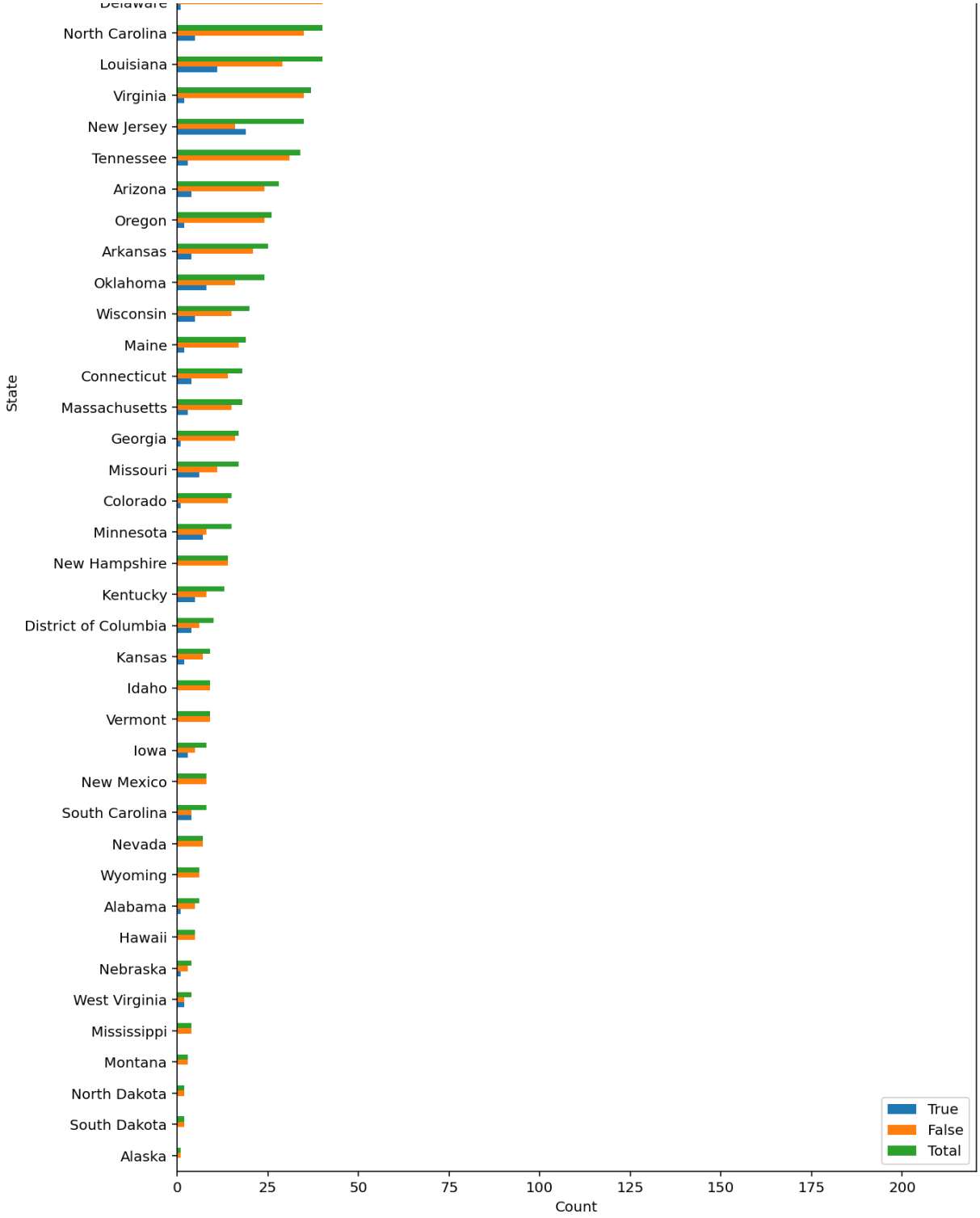

counts of severe outages per climate region
Northeast
South
West
Central
Southeast
East North Central
Northwest
Southwest
West North Central
Climate region
0
50
100
150
200
250
300
350
Count
True
False
Total

In [39]:
```
# comparing the amount of severe power outages to the amount of
# non-severe power outages in different U.S. States

severity = pd.DataFrame({'True': severity_true, 'False': severity_false, 'Total':
                         'state': outages['U.S._STATE']})

plot = (
    severity.groupby('state').sum().reset_index()
    .sort_values(['Total'], ascending=True).plot
    .barh(x='state', figsize=(10, 20), title='counts of severe vs non-severe outa
)
plot.set_xlabel('Count')
plot.set_ylabel('State')


# pivot table representation of data, filling NaN with 0 to get total outage
state_pt = pd.pivot_table(outages, index=outages['U.S._STATE'], columns=outages['
                          aggfunc=len)['YEAR'].fillna(0)
state_pt['Total'] = state_pt[False] + state_pt[True]
state_pt.head(10)
```

Out[39]:

HIGHER.SEVERITY	False	True	Total
U.S._STATE			
Alabama	5.0	1.0	6.0
Alaska	1.0	0.0	1.0
Arizona	24.0	4.0	28.0
Arkansas	21.0	4.0	25.0
California	183.0	27.0	210.0
Colorado	14.0	1.0	15.0
Connecticut	14.0	4.0	18.0
Delaware	40.0	1.0	41.0
District of Columbia	6.0	4.0	10.0
Florida	28.0	17.0	45.0

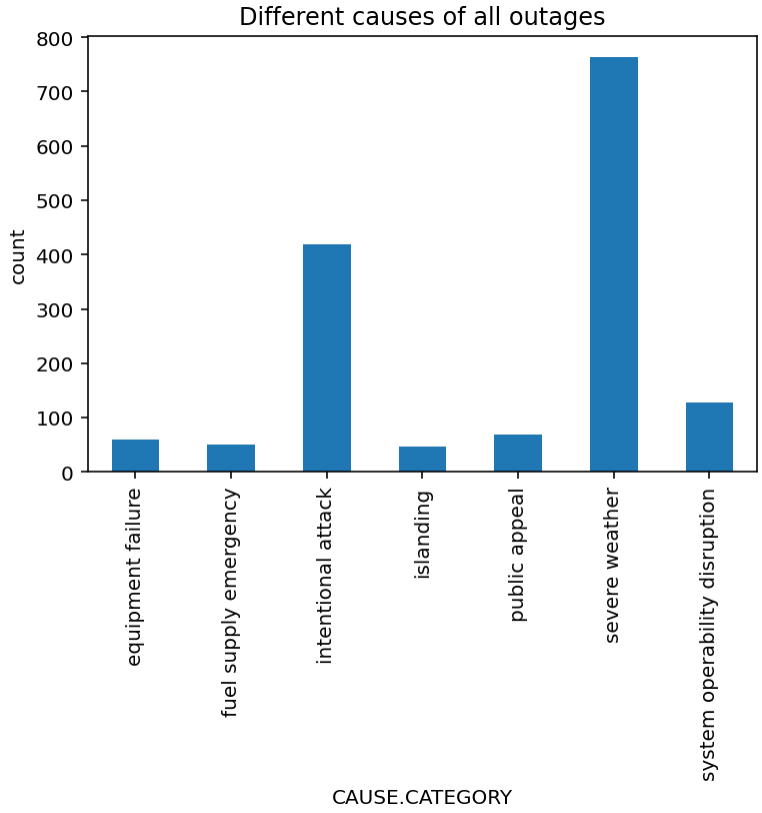


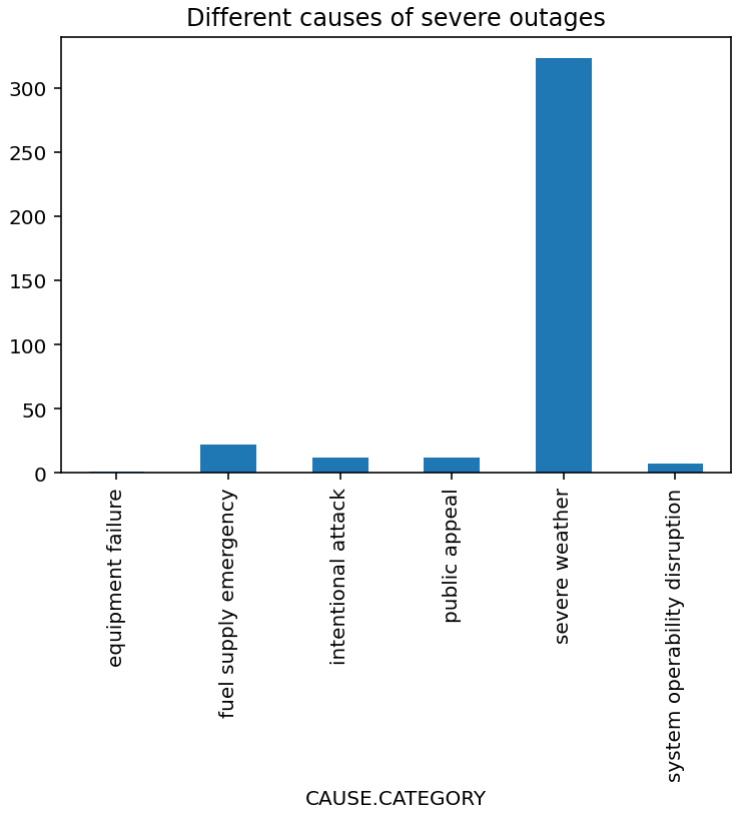


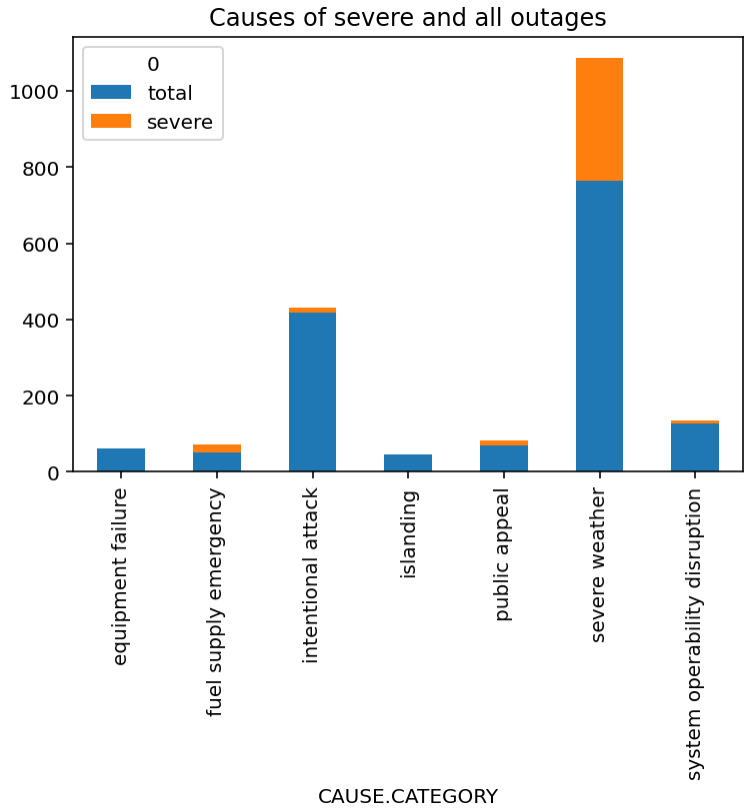
When plotting the causes of all outages, severe weather and intentional attacks are the top two leading causes. However, when only graphing severe outages, severe weather is the leading cause, and the second leading cause, 'fuel supply emergency', falls much farther than the first. With the stacked bar chart it's easier to visualize the difference between the leading causes.

In [40]:
```
# graph of causes of all outages
all_causes = (
    outages.groupby('CAUSE.CATEGORY').count()
    .reset_index().plot.bar(x='CAUSE.CATEGORY', y='YEAR', legend=False,
    title='Different causes of all outages')
)
all_causes.set_ylabel('count')

# graph of causes of severe outages
severe_causes = (
    outages[outages['HIGHER.SEVERITY']==True]
    .groupby('CAUSE.CATEGORY').count().reset_index()
    .plot.bar(x='CAUSE.CATEGORY', y='YEAR', legend=False,
    title='Different causes of severe outages')
)
all_causes.set_ylabel('count')

# creating a stacked bar chart
severe = outages.groupby('CAUSE.CATEGORY').sum().reset_index()
every = outages.groupby('CAUSE.CATEGORY').count()[['HIGHER.SEVERITY']].reset_inde
both = every.merge(severe, on='CAUSE.CATEGORY')
both = both.rename(columns={'HIGHER.SEVERITY_x': 'total', 'HIGHER.SEVERITY_y': 's
both.plot.bar(stacked=True, title='Causes of severe and all outages')
```

Out[40]:
```
<AxesSubplot:title={'center':'Causes of severe and all outages'}, xlabel='CAUS
E.CATEGORY'>
```




Different causes of severe outages
300
250
200
150
100
50
0
equipment failure
fuel supply emergency
intentional attack
public appeal
severe weather
system operability disruption
CAUSE.CATEGORY



Assessment of Missingness

Looking at our dataset, we have a lot of missing values in our columns, some more so than others. We can get the percentages of non-missing to total values by running the cell below.

In [17]:
```
percentages = outages.copy()
percentages.loc['percentages'] = (outages.count()/outages.shape[0]*100)
percentages[percentages['CLIMATE.REGION'].isna()]

percentages.iloc[[-1]]
```

Out[17]:

	YEAR	MONTH	U.S._STATE	POSTAL.CODE	NERC.REGION	CLIMATE.REGION	ANOM
percentages	100	99.4133	100	100	100	99.6089	

Since I am interested in the causes of power outages, I am also interested in CAUSE.CATEGORY.DETAIL, and I want to see if HURRICANE.NAMES is MAR dependent on CAUSE.CATEGORY.DETAIL. Since this is a categorical comparison, I will use TVD in my permutation tests.

In [18]:
```
# create a dataframe with only wanted columns
df1 = outages[['CAUSE.CATEGORY.DETAIL', 'HURRICANE.NAMES']]
df1 = df1[df1['CAUSE.CATEGORY.DETAIL'].notnull()].reset_index(drop=True)
df1['is_null'] = df1['HURRICANE.NAMES'].isnull()
df1
```

Out[18]:

	CAUSE.CATEGORY.DETAIL	HURRICANE.NAMES	is_null
0	vandalism	NaN	True
1	heavy wind	NaN	True
2	thunderstorm	NaN	True
3	winter storm	NaN	True
4	tornadoes	NaN	True
...	...	...	...
1058	sabotage	NaN	True
1059	vandalism	NaN	True
1060	uncontrolled loss	NaN	True
1061	Coal	NaN	True
1062	failure	NaN	True

1063 rows × 3 columns

Setting a significance level of 0.95.

After running the permutation test with 1000 simulations, the p-value comes out to be 0.996. Since this permutation test's purpose was to find if the missingness of HURRICANE.NAME and CAUSE.CATEGORY.DETAIL came from the same distribution or not, having a high p-value means it is highly likely they come from the same distribution, meaning we can accept that HURRICANE.NAMES is MAR dependent on CAUSE.CATEGORY.DETAIL.

In [19]:
```
# helper function to find the total variation distance
def tvd1(df):
    cnts = (
        pd.pivot_table(df, index='is_null', columns='CAUSE.CATEGORY.DETAIL',
                       aggfunc='size').fillna(0)
    )
    distr = cnts / cnts.sum()
    return distr.diff(axis=1).iloc[-1].abs().sum() / 2

# create observed test statistic
obs = tvd1(df1)

# run simulation 1000 times
tvds = []
n_repetitions = 1000

for _ in range(n_repetitions):
    # sample and shuffle within dataframe, then append calculated tvd to list
    s = df1['is_null'].sample(frac=1, replace=False).reset_index(drop=True)
    shuffled = df1.assign(**{'is_null': s})
    tvds.append(tvd1(shuffled))

tvds = pd.Series(tvds)

# find p-value
p_val = (tvds <= obs).sum() / n_repetitions
p_val
```

Out[19]:
```
0.002
```

Now I want to know if CAUSE.CATEGORY.DETAIL is MAR dependent on OUTAGE.DURATION.

In [20]:
```
df2 = outages[['OUTAGE.DURATION', 'CAUSE.CATEGORY.DETAIL']]
df2 = df2[df2['OUTAGE.DURATION'].notnull()].reset_index(drop=True)
df2['is_null'] = df2['CAUSE.CATEGORY.DETAIL'].isnull()
df2
```

Out[20]:

	OUTAGE.DURATION	CAUSE.CATEGORY.DETAIL	is_null
0	3060	NaN	True
1	1	vandalism	False
2	3000	heavy wind	False
3	2550	thunderstorm	False
4	1740	NaN	True
...	...	...	...
1471	0	sabotage	False
1472	220	uncontrolled loss	False
1473	720	NaN	True
1474	59	NaN	True
1475	181	NaN	True

1476 rows × 3 columns

Setting a significance level of 0.95 After running the permutation test with 1000 simulations, the p-value comes out to be 0.7. Since this permutation test's purpose was to find if the missingness of OUTAGE.DURATION and CAUSE.CATEGORY.DETAIL came from the same distribution or not, having a lower p-value than our significance level means it is likely that they did not come from the same distribution, meaning we cannot accept that OUTAGE.DURATION is MAR dependent on CAUSE.CATEGORY.DETAIL.

In [21]:
```
# helper function to find the total variation distance
def tvd2(df):
    cnts = (
        pd.pivot_table(df, index='is_null', columns='OUTAGE.DURATION',
                       aggfunc='size').fillna(0)
    )
    distr = cnts / cnts.sum()
    return distr.diff(axis=1).iloc[-1].abs().sum() / 2

# create observed test statistic
obs2 = tvd2(df2)

# run simulation 1000 times
tvds = []
n_repetitions = 1000

for _ in range(n_repetitions):
    # sample and shuffle within dataframe, then append calculated tvd to list
    s = df2['is_null'].sample(frac=1, replace=False).reset_index(drop=True)
    shuffled = df2.assign(**{'is_null': s})
    tvds.append(tvd2(shuffled))

tvds = pd.Series(tvds)

# find p-value
p_val = (tvds <= obs2).sum() / n_repetitions
p_val
```

Out[21]: 0.287

Hypothesis Test

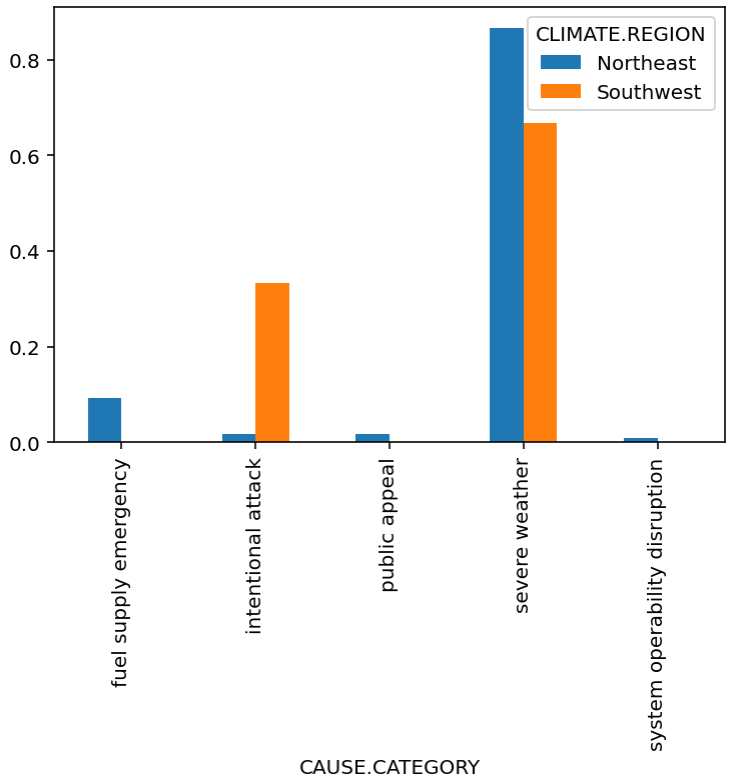
After looking at different distributions of categorical data that correspond to location, climate, and cause, I am interested in digging deeper into differences between climate regions, specifically the Northeast and West regions. I'm choosing these regions because the Northeast has the highest amount of outages out of all regions, and the Southwest region may be interesting to compare against because it's location is geographically opposite from the Northeast. I want to know if the distributions for the causes of severe outages in the Northeast and Southwest regions are the same.

My null hypothesis is that the distribution of causes for severe outages between the Northeast and West regions are the same. The alternative hypothesis is that the distribution of causes for severe outages between Northeast and West regions are difference. In my permutation test, I will be using the TVD as a test statistic since we are comparing categorical variables.

After filtering the dataset, calculating my observed statistic and running a permutation test with 1000 simulations, we get a p-value of 0.04. Since this is lower than the significance level of 0.05 that we set, we can reject the null hypothesis.

In [32]:
```
# filtering dataset to only account for severe outages in NE and SW
severe_outages = outages[outages['HIGHER.SEVERITY'] == True]
severe_outages = (severe_outages[(severe_outages['CLIMATE.REGION'] == 'Northeast'
                                  (severe_outages['CLIMATE.REGION'] == 'Southwest'
severe_outages = (severe_outages[['CLIMATE.REGION', 'CAUSE.CATEGORY']]
                  .reset_index(drop=True))

cnts = pd.pivot_table(severe_outages,
            index = 'CAUSE.CATEGORY',
            columns = 'CLIMATE.REGION',
            aggfunc = 'size',
            fill_value = 0)

normalized = cnts / cnts.sum()

# visualization
normalized.plot.bar()
```

Out[32]: <AxesSubplot:xlabel='CAUSE.CATEGORY'>



In [33]:
```
# creating a helper function to calculate TVD
def total_variation_distance(df):
    cnts = (
        pd.pivot_table(
            df, index = 'CAUSE.CATEGORY',
            columns = 'CLIMATE.REGION',
            aggfunc = 'size', fill_value = 0
        )
    )
    normalized = cnts / cnts.sum()
    return normalized.diff(axis=1).iloc[:,-1].abs().sum() / 2
```

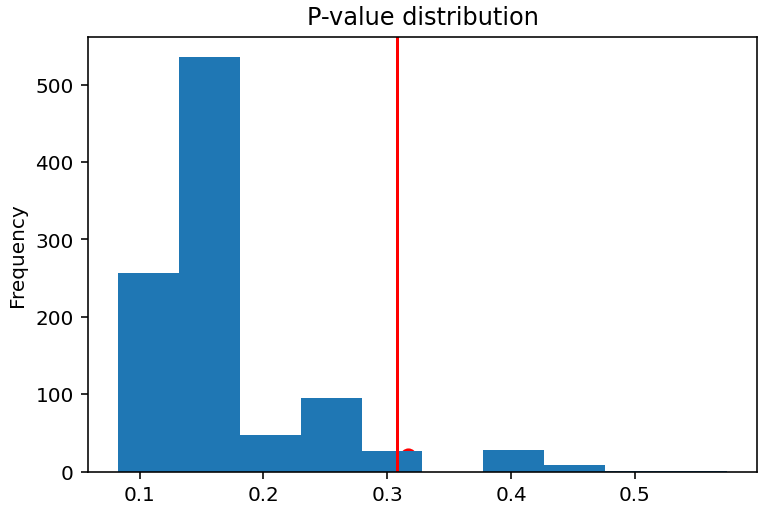
In [34]:
```
# observed statistic
observed = total_variation_distance(severe_outages)

# permutation test, simulating 1000 times
n_repetitions = 1000
tvds = []

for _ in range(n_repetitions):
    s = (severe_outages['CLIMATE.REGION']
         .sample(frac=1, replace=False).reset_index(drop=True))
    shuffled = severe_outages[['CAUSE.CATEGORY']].assign(**{'CLIMATE.REGION': s})
    calculated = total_variation_distance(shuffled)
    tvds.append(calculated)

tvds = pd.Series(tvds)
```

In [35]:
```
# p-value = 0.046, we will reject the null hypothesis.
p_val = (tvds >= observed).sum() / n_repetitions
p_val
```

Out[35]: 0.046

In [36]:
```
# distribution of tvds and observed
tvds.plot(kind='hist', title='P-value distribution')
plt.scatter([observed], [20], s=50, color='r')
percentile = np.percentile(tvds, 95)
plt.axvline(x=percentile, color='r')
```

Out[36]: <matplotlib.lines.Line2D at 0x17d6da229d0>



Conclusion

Although much was covered in this project, I would like to perhaps redefine my definition of "severity" in the future, as duration of the outage does not seem like enough to justify it being severe. In the future, I would try to incorporate the amount of customers affected and demand loss to create a better definition. I would also like to explore more about the relationship between location and cause of outage to further answer the line of inquiry.