outages

November 14, 2019

1 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?

1.0.1 Getting the Data

The data is downloadable here.

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

1.0.2 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.

1.0.3 Assessment of Missingness

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

1.0.4 Hypothesis Test

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

2 Summary of Findings

2.0.1 Introduction

This dataset, as stated above, revolves around major outages in the US, where major is defined as affecting at least 50,000 people or by an unplanned firm loss of at least 300 Megawatts, between Janurary 2000 to July 2016.The main question I'm planning on addressing is how abnormal weather affects the cause of outages compared to normal weather. Relating this question to the dataset, I would primarily focus on the column, CAUSE.CATEGORY, as well as the CLIMATE column based off the column CLIMATE.CATEGORY, which will state if the climate during an outage was abnormal or not.

2.0.2 Cleaning and EDA

I begin the cleaning process by reading in the excel file and dropping the first 3 rows of the dataset since those lines are only there to explain the dataset. The next step of cleaning the data was combining the columns OUTAGE.START.DATE and OUTAGE.START.TIME into one new single column OUTAGE.START, since its unneccessary to have two seperate columns solely for the date and time of an outage. I also applied this to the OUTAGE.RESTORATION.DATE and OUTAGE.RESTORATION.TIME columns, combining them into a new column OUTAGE.RESTORATION for the same purpose. I also create the CLIMATE column that I stated previously, where both 'warm' and 'cold' values in the column CLIMATE.CATEGORY are merged into a single string, 'abnormal'.

2.0.3 Assessment of Missingness

For my assessments of missingness the two questions I addressed were 'Is the cause category detail column missing at random or dependent on the climate category?' and 'Is the customers affected column missing at random or dependent on climate column?' Both processes started off the same way, using the total variation distance as the test statistic, a significance level of 0.05, and stating that the null hypothesis was that differences were due to chance and that the alternative hypothesis was indicating dependency, then I would get the empirical distribution of both questions and plot them. For the first question, I noticed that the distributions of the plots weren't very similar,

making me believe already that the missingness was dependent between the columns, and getting a p-value of 0 also proved the dependency or the NMAR relation. For the second question, the distributions of the plots were very similar to one another, so I assumed they were going to be independent and solving for the p-value gave me a value larger than our significance level (0.324) , justifying the independency of the columns or them being MAR to one another.

2.0.4 Hypothesis Test

Staying on the same topic of the main question I wanted to address on this dataset but mainly focusing on distributions up to this point, I tweaked the question to fit a similar theme ending up with 'Is the distribution of the cause of outages different for normal climates than it is for abnormal climates?' My null hypothesis would be that there is no significant difference and that any differences were primarily due to chance, while my alternative hypothesis is that the distributions due to weather is different, with a test statistic of total variation distance (again) and a significance level of 0.05 (also again). After going through the process of finding the p-value once again I ended up with a p-value of 0.364, making my conclusion fail to reject the null hypothesis, meaning that the differences were most likely due to chance.

3 Code

[2]:
```
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
```

[3]:
```
#Read in the Excel file
df = pd.read_excel('outage.xlsx')
```

3.0.1 Cleaning and EDA

[4]:
```
#Set the correct columns in the Excel file, ignoring the previous statements␣
↪explaining the dataset
new_col = df.loc[4:].loc[4].values
df.columns = new_col
df = df.loc[6:].set_index('OBS').drop('variables', axis = 1)
```

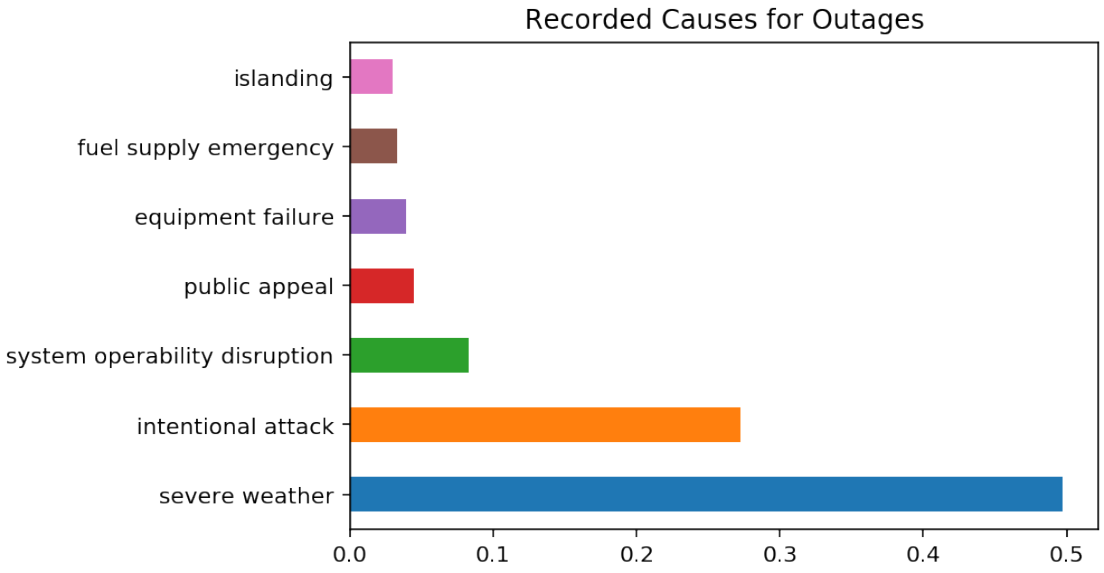
[5]:
```
#Cleaning the Dataset

#Combine OUTAGE.START.DATE and OUTAGE.START.TIME into a new column OUTAGE.START
#Drop the previous columns
df['OUTAGE.START.DATE'] = pd.to_datetime(df['OUTAGE.START.DATE'])
```

```
df['OUTAGE.START.TIME'] = pd.to_timedelta(df['OUTAGE.START.TIME'].astype(str))
df['OUTAGE.START'] = df['OUTAGE.START.DATE'] + df['OUTAGE.START.TIME']
df = df.drop('OUTAGE.START.DATE', axis = 1)
df = df.drop('OUTAGE.START.TIME', axis = 1)

#Combine OUTAGE.RESTORATION.DATE and OUTAGE.RESTORATION.TIME into a new column␣
↪OUTAGE.RESTORATION
#Drop the previous columns
df['OUTAGE.RESTORATION.DATE'] = pd.to_datetime(df['OUTAGE.RESTORATION.DATE'])
df['OUTAGE.RESTORATION.TIME'] = pd.to_timedelta(df['OUTAGE.RESTORATION.TIME'].
↪astype(str))
df['OUTAGE.RESTORATION'] = df['OUTAGE.RESTORATION.DATE'] + df['OUTAGE.
↪RESTORATION.TIME']
df = df.drop('OUTAGE.RESTORATION.DATE', axis = 1)
df = df.drop('OUTAGE.RESTORATION.TIME', axis = 1)
df.rename(columns = {'U.S._STATE': 'STATE'}, inplace = True) #Rename the␣
↪atrocious given state
```

[6]:
```
#Create a new column based off one given in the dataset focusing on normal vs␣
↪abnormal climates
df['CLIMATE'] = df['CLIMATE.CATEGORY'].replace({'warm': 'abnormal', 'cold':␣
↪'abnormal'})
```

[7]:
```
#Univariate Analysis
df['CAUSE.CATEGORY'].value_counts(normalize = True).plot(kind = 'barh', title =␣
↪'Recorded Causes for Outages')
```

[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f35c4cbb748>



[8]:
```
#The normalized distribution of cold climates on the cause of outages
df.loc[df['CLIMATE.CATEGORY'] == 'cold']['CAUSE.CATEGORY'].
 ↪value_counts(normalize = True)
```

[8]:
```
severe weather                  0.505285
intentional attack              0.257928
system operability disruption   0.078224
public appeal                   0.046512
equipment failure               0.040169
fuel supply emergency           0.040169
islanding                       0.031712
Name: CAUSE.CATEGORY, dtype: float64
```

[9]:
```
#The normalized distribution of warm climates on the cause of outages
df.loc[df['CLIMATE.CATEGORY'] == 'warm']['CAUSE.CATEGORY'].
 ↪value_counts(normalize = True)
```

[9]:
```
severe weather                  0.538961
intentional attack              0.227273
system operability disruption   0.097403
islanding                       0.045455
public appeal                   0.042208
equipment failure               0.032468
fuel supply emergency           0.016234
Name: CAUSE.CATEGORY, dtype: float64
```

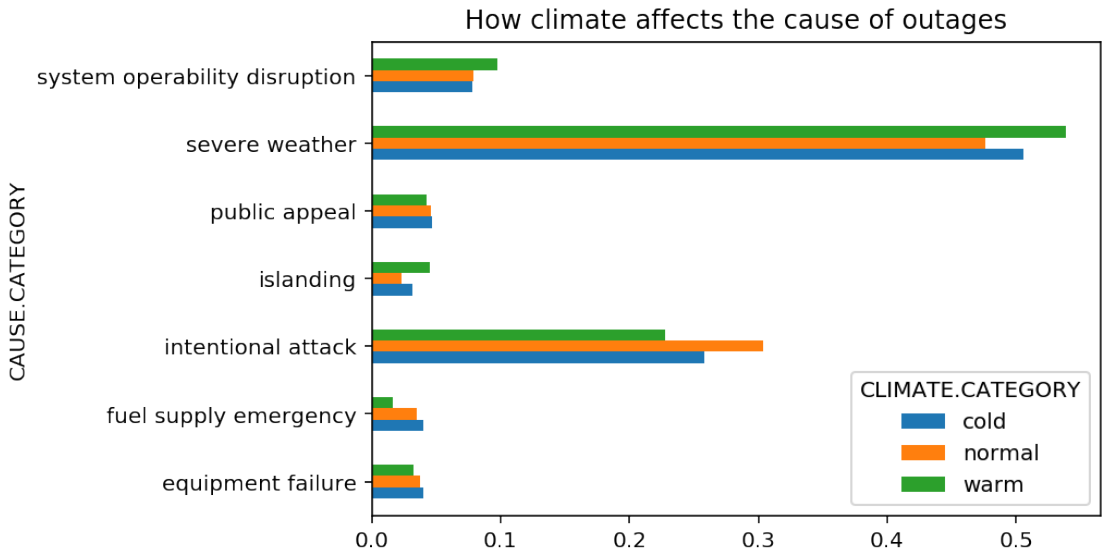
[10]:
```
#The normalized distribution of normal climates on the cause of outages
df.loc[df['CLIMATE.CATEGORY'] == 'normal']['CAUSE.CATEGORY'].
 ↪value_counts(normalize = True)
```

[10]:
```
severe weather                  0.475806
intentional attack              0.303763
system operability disruption   0.079301
public appeal                   0.045699
equipment failure               0.037634
fuel supply emergency           0.034946
islanding                       0.022849
Name: CAUSE.CATEGORY, dtype: float64
```

[11]:
```
#Bivariate Analysis on cause category and climate category

#Create a dataframe with cause category and climate category with the␣
 ↪aggregation of size
causal_counts = df.pivot_table(index = 'CAUSE.CATEGORY', columns = 'CLIMATE.
 ↪CATEGORY', aggfunc = 'size')
```

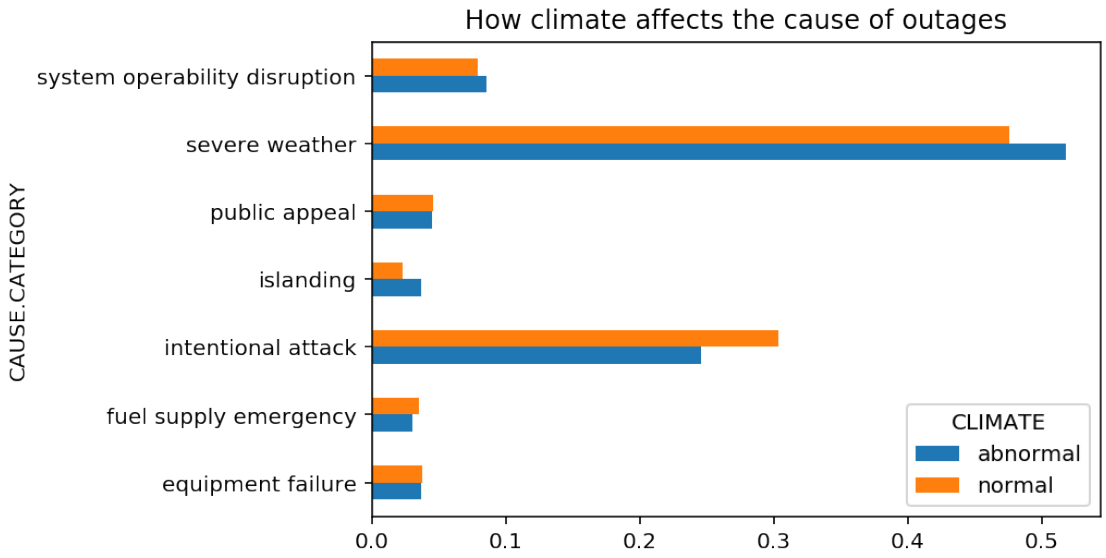
```
#Apply conditional distribution on the newly created dataframe
cond_dist = causal_counts.apply(lambda x: x / x.sum())

#Plot the distribution with a horizontal bar graph
cond_dist.plot(kind = 'barh', title = 'How climate affects the cause of␣
↪outages')
```

[11]: <matplotlib.axes._subplots.AxesSubplot at 0x7f35c4bb06a0>



[12]:
```
#Bivariate Analysis on cause category and the new climate column

#Create a dataframe with cause category and climate category with the␣
↪aggregation of size
causal_counts = df.pivot_table(index = 'CAUSE.CATEGORY', columns = 'CLIMATE',␣
↪aggfunc = 'size')

#Apply conditional distribution on the newly created dataframe
cond_dist = causal_counts.apply(lambda x: x / x.sum())

#Plot the distribution with a horizontal bar graph
cond_dist.plot(kind = 'barh', title = 'How climate affects the cause of␣
↪outages')
```

[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f35c4b46390>



[13]:
```
#Interesting aggregates
```

[14]:
```
#Total amount of people affected by outages based off the cause
df.groupby('CAUSE.CATEGORY')['CUSTOMERS.AFFECTED'].agg(np.sum)
```

[14]:
```
CAUSE.CATEGORY
equipment failure                 3058067
fuel supply emergency                   1
intentional attack                 356315
islanding                          209749
public appeal                      159994
severe weather                  135208133
system operability disruption    17518480
Name: CUSTOMERS.AFFECTED, dtype: int64
```

[15]:
```
#Amount of people affected by the cause of an outage by state
df.groupby(['STATE', 'CAUSE.CATEGORY'])['CUSTOMERS.AFFECTED'].agg(np.sum).head()
```

[15]:
```
STATE    CAUSE.CATEGORY
Alabama  intentional attack         0
         severe weather        471644
Alaska   equipment failure      14273
Arizona  equipment failure     167000
         intentional attack      2713
Name: CUSTOMERS.AFFECTED, dtype: int64
```

[16]:
```
#Number of outages per state
df.groupby('STATE')['CAUSE.CATEGORY'].agg('count').head()
```

[16]:
```
STATE
Alabama          6
Alaska           1
Arizona         28
Arkansas        25
California     210
Name: CAUSE.CATEGORY, dtype: int64
```

[17]:
```
#Number of outages affected by climate and a specific cause by state
df.groupby(['STATE', 'CAUSE.CATEGORY', 'CLIMATE'])['CLIMATE'].agg('count').
↪head()
```

[17]:
```
STATE    CAUSE.CATEGORY      CLIMATE
Alabama  intentional attack  normal      1
         severe weather      abnormal    3
                             normal      1
Arizona  equipment failure   abnormal    1
                             normal      3
Name: CLIMATE, dtype: int64
```

3.0.2 Assessment of Missingness

[111]:
```
#Find the missingness of each column in the dataset and sort from greatest to␣
↪least

missing_vals = pd.isnull(df).sum()
missing_vals.sort_values(ascending = False).head()
```

[111]:
```
HURRICANE.NAMES          1462
DEMAND.LOSS.MW            705
CAUSE.CATEGORY.DETAIL     471
CUSTOMERS.AFFECTED        443
OUTAGE.RESTORATION         58
dtype: int64
```

I decide to choose the 3rd and 4th columns with the greatest amount of missingness since the column with the greatest amount of missingness is Missing by Design, only being non-null when the cause category of an outage is due specifically to a Hurricane, while the 2nd column with the most missingness didn't seem to have as much relevance to my main question (granted the amount of people do not have that much relevance either but its a much more interesting column to address in my opinion)

Is the cause category detail column missing at random or dependent on the climate category?

Null hypothesis: The missingness of the cause category detail column is not dependent on the climate category. Any substantial difference is due to chance.

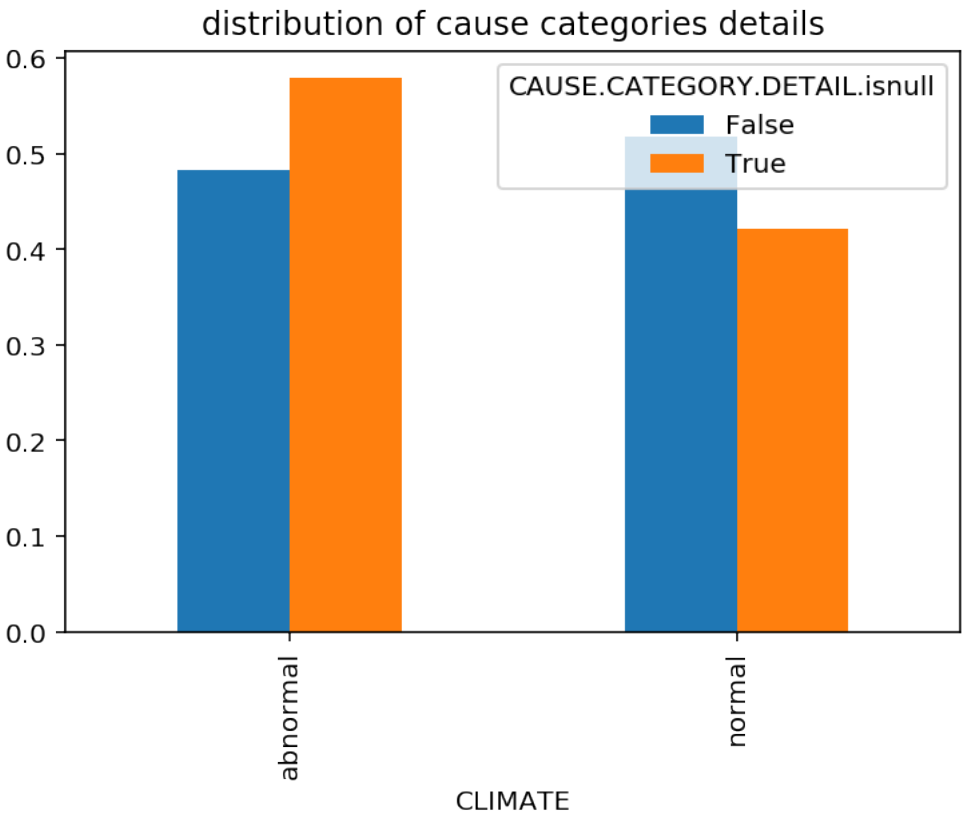
Alternative hypothesis: The missingness of the cause category detail column is dependent on the climate column.

Test statistic -> Total Variation Distance

Significance level -> 0.05

[105]:
```
#Append a new column to our dataframe where true values is where CLIMATE.
 ↪CATEGORY is null
df['CAUSE.CATEGORY.DETAIL.isnull'] = df['CAUSE.CATEGORY.DETAIL'].isnull()
```

[106]:
```
#Get the empirical distribution using total variation distance

emp_distributions = (
    df
    .pivot_table(columns='CAUSE.CATEGORY.DETAIL.isnull', index='CLIMATE',␣
 ↪values=None, aggfunc='size')
    .fillna(0)
    .apply(lambda x:x/x.sum())
)

#Plot the empirical distribution with a bar graph
emp_distributions.plot(kind='bar', title='distribution of cause categories␣
 ↪details');
```



[107]:
```
#Get the observed total variation distance from the data
observed_tvd = np.sum(np.abs(emp_distributions.diff(axis=1).iloc[:,-1])) / 2
observed_tvd
```

[107]: 0.09625894927901582

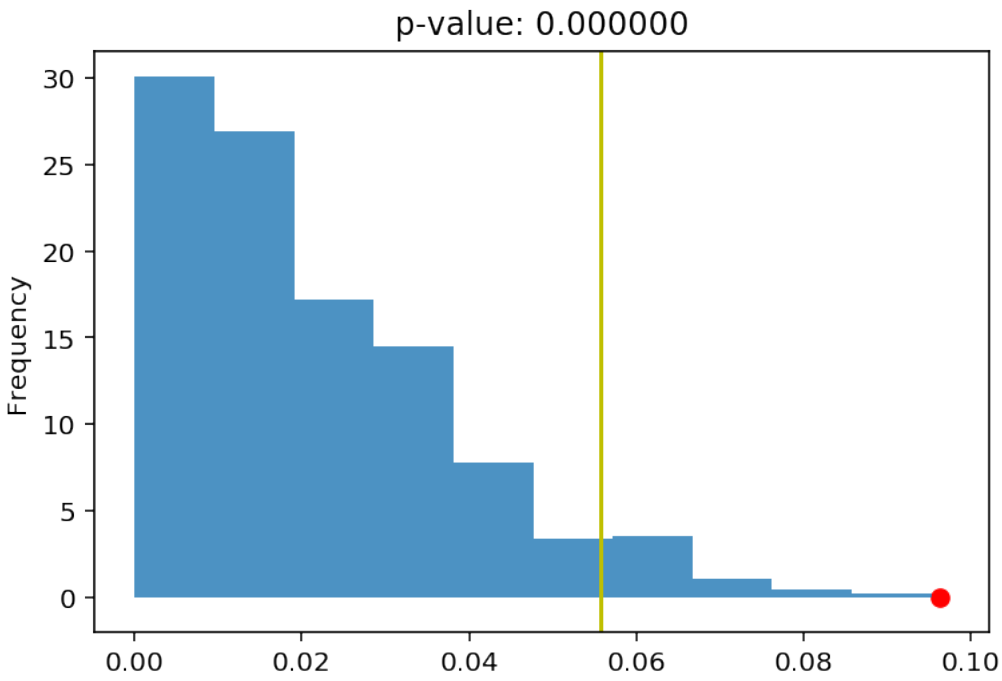
[108]:
```
n_repetitions = 500

#Copy the main dataset but only have the two listed columns
df_cli = df.copy()[['CLIMATE', 'CAUSE.CATEGORY.DETAIL.isnull']]
tvds = []
for _ in range(n_repetitions):

    # shuffle the climates
    shuffled_cli = (
        df_cli['CLIMATE']
        .sample(replace=False, frac=1)
        .reset_index(drop=True)
    )

    # put them in a table
```

```
    shuffled = (
        df_cli
        .assign(**{'Shuffled Climates': shuffled_cli})
    )

    # compute the tvd
    shuffled_emp_distributions = (
        shuffled
        .pivot_table(columns='CAUSE.CATEGORY.DETAIL.isnull', index='Shuffled␣
↪Climates', values=None, aggfunc='size')
        .fillna(0)
        .apply(lambda x:x/x.sum())
    )

    tvd = np.sum(np.abs(shuffled_emp_distributions.diff(axis=1).iloc[:,-1])) / 2

    #add it to the list of results
    tvds.append(tvd)
```

[109]:
```
#Get the p-value

pval = np.count_nonzero(tvds >= observed_tvd) / len(tvds)
pval
```

[109]: 0.0

[110]:
```
#Plot the findings into a histogram

pd.Series(tvds).plot(kind='hist', density=True, alpha=0.8, title='p-value: %f'␣
↪% pval)
plt.scatter(observed_tvd, 0, color='red', s=40);

perc = np.percentile(tvds, 95) # 5% significance level
plt.axvline(x=perc, color='y');
```



Conclusion: We reject the null hypothesis and conclude these columns are dependent, or NMAR.

Solving for this conclusion reveals that choosing the column CAUSE.CATEGORY.DETAIL instead of CAUSE.CATEGORY for answering my primary question would probably yield a similar p-value. This is useful information since I know that CAUSE.CATEGORY has much fewer missing values and I'd rather use as much of the given data as possible, making CAUSE.CATEGORY a better choice.

Is the customers affected column missing at random or dependent on climate column?

Null hypothesis: The missingness of the customers affected column is not dependent on the climate column. Any substantial difference is due to chance.

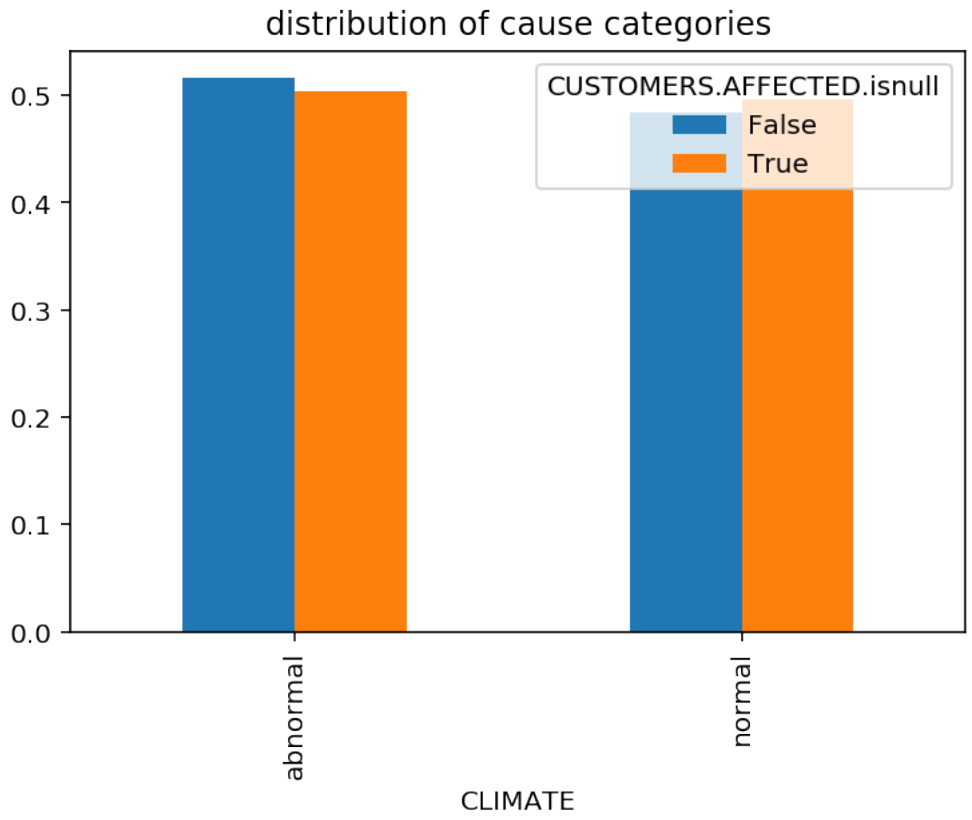
Alternative hypothesis: The missingness of the customers affected column is dependent on the climate column.

Test statistic -> Total Variation Distance

Significance level -> 0.05

[46]:
```
#Append a new column to our dataframe where true values is where CUSTOMERS.
 ↪AFFECTED is null
df['CUSTOMERS.AFFECTED.isnull'] = df['CUSTOMERS.AFFECTED'].isnull()
```

[61]:
```
#Get the empirical distribution using total variation distance

emp_distributions = (
    df
    .pivot_table(columns='CUSTOMERS.AFFECTED.isnull', index='CLIMATE',␣
 ↪values=None, aggfunc='size')
    .fillna(0)
    .apply(lambda x:x/x.sum())
)

#Plot the empirical distribution with a bar graph
emp_distributions.plot(kind='bar', title='distribution of customers affected');
```



[63]:
```
#Get the observed total variation distance from the data
```

```
observed_tvd = np.sum(np.abs(emp_distributions.diff(axis=1).iloc[:,-1])) / 2
observed_tvd
```

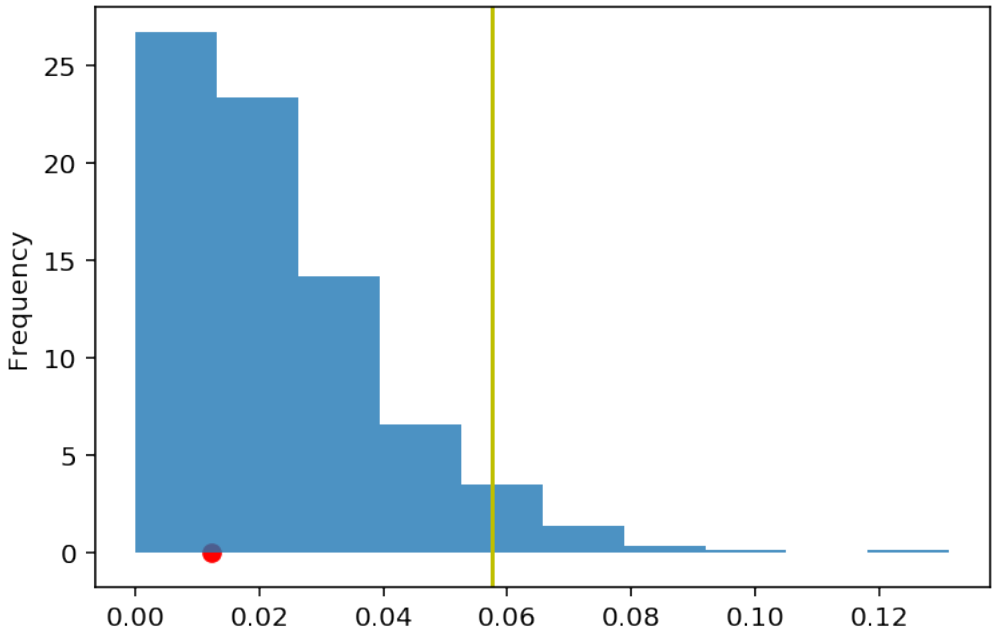
[63]: 0.012281296282350557

[104]:
```
n_repetitions = 500

#Copy the main dataset but only have the two listed columns
df_cli = df.copy()[['CLIMATE', 'CUSTOMERS.AFFECTED.isnull']]
tvds = []
for _ in range(n_repetitions):

    # shuffle the climates
    shuffled_cli = (
        df_cli['CLIMATE']
        .sample(replace=False, frac=1)
        .reset_index(drop=True)
    )

    # put them in a table
    shuffled = (
        df_cli
        .assign(**{'Shuffled Climates': shuffled_cli})
    )

    # compute the tvd
    shuffed_emp_distributions = (
        shuffled
        .pivot_table(columns='CUSTOMERS.AFFECTED.isnull', index='Shuffled␣
↪Climates', values=None, aggfunc='size')
        .fillna(0)
        .apply(lambda x:x/x.sum())
    )

    tvd = np.sum(np.abs(shuffed_emp_distributions.diff(axis=1).iloc[:,-1])) / 2

    #add it to the list of results
    tvds.append(tvd)
```

[69]:
```
#Get our p-value

pval = np.count_nonzero(tvds <= observed_tvd) / len(tvds)
pval
```

[69]: 0.324

[70]:
```
#Plot the findings into a histogram

pd.Series(tvds).plot(kind='hist', density=True, alpha=0.8, title='p-value: %f'␣
↪% pval)
plt.scatter(observed_tvd, 0, color='red', s=40);

perc = np.percentile(tvds, 95) # 5% significance level
plt.axvline(x=perc, color='y');
```



Conclusion: We fail to reject the null hypothesis making these two columns independent or in other words, MAR

I wasn't planning on using the amount of people affected by outages in regards to my primary question, but the conclusion from this permutation test gives me some interesting new information which is that climate during outages has no connection to whether or not the population affected by an outage is recorded, still not being relevant in the question i'm trying to answer but still interesting nonetheless.

3.0.3 Hypothesis / Permutation Test

Is the distribution of the cause of outages different for normal climates than it is for abnormal climates?

Null hypothesis: The distribution of outage causes during normal weather is the same compared to outage causes during abnormal weather. Any abnormalities between the two samples is due to chance.

Alternative hypothesis: The distributions of outage causes based on the weather is different.

Test statistic -> Total Variation Distance

Significance level -> 0.05

[34]:
```
#Helper method to calculate the total variation distance

def tvd_formula(df):
    cnts = df.pivot_table(index='CAUSE.CATEGORY', columns='CLIMATE',␣
 ↪aggfunc='size')
    distr = cnts.apply(lambda x: x / x.sum())
    return distr.diff(axis=1).iloc[-1].abs().sum() / 2
```

[35]:
```
#Calculate the observed total variation distance from the data

obs = tvd_formula(df)
obs
```

[35]:
```
0.0032431883579089396
```

[36]:
```
#One test example
s = df['CLIMATE'].sample(frac=1, replace=False).reset_index(drop=True)
shuffled = df.loc[:, ['CAUSE.CATEGORY']].assign(CLIMATE=s)

tvd_formula(shuffled)
```

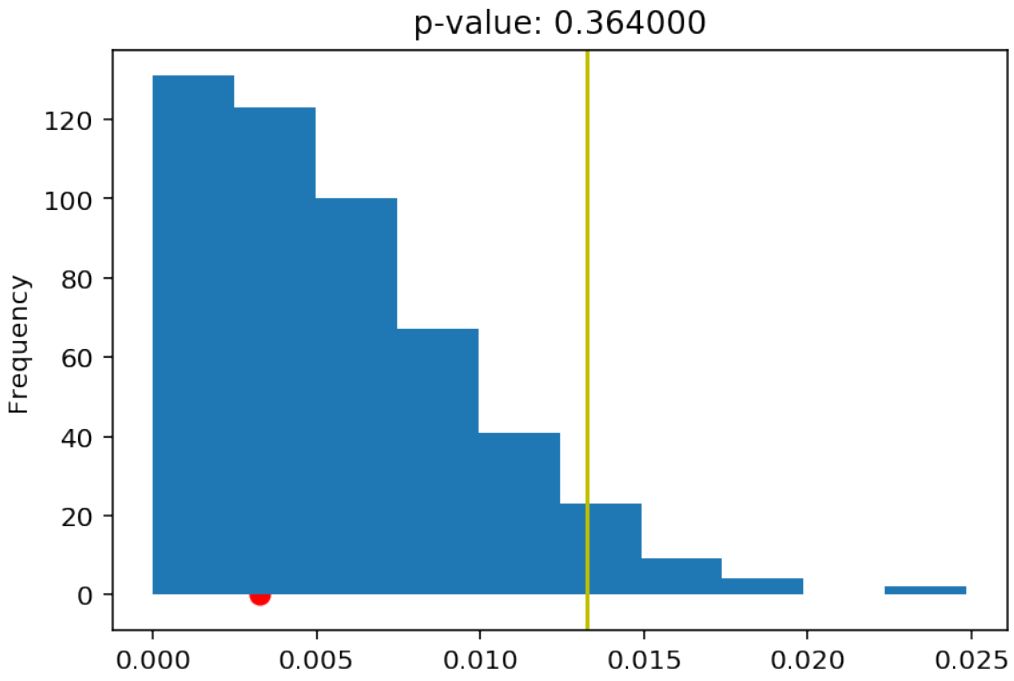
[36]:
```
0.001985111662531021
```

[37]:
```
N = 500
tvds = []
for _ in range(N):

    #Sample, find the TVD and append
    s = df['CLIMATE'].sample(frac=1, replace=False).reset_index(drop=True)
    shuffled = df.loc[:, ['CAUSE.CATEGORY']].assign(CLIMATE=s)

    tvds.append(tvd_formula(shuffled))

tvds = pd.Series(tvds)
```

[38]:
```
pval = (tvds <= obs).sum() / N
pval
```

[38]: 0.364

[43]:
```
tvds.plot(kind='hist', title='p-value: %f' % pval)
plt.scatter([obs], [0], s=50, color='r')

perc = np.percentile(tvds, 95) # 5% significance level
plt.axvline(x=perc, color='y');
```



Conclusion: We fail to reject our null hypothesis, the difference our distribution is most likely due to chance

With this conclusion, we discover that the climate during an outage does not have a significant effect on the cause of it.

[ ]: