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

Summary of Findings

Introduction

The data that I decided to analyze was the set out outages and information on their cause, location, and population. The question that I investigated throughout this project is if there is a correlation between urban and non-urban regions and the duration of outages. While cleaning the data, there were different parts of data that revealed potential topics to investigate. The part that interested me the most was the urban/non-urban regions and I wanted to take a look at outage durations. That way I can work with pandas dataframe and performe a permutation test, and see if I could reject or fail to reject the null hypothesis.

Cleaning and EDA

My data-cleaning process first started with reading the excel sheet into a pandas dataframe.

Firstly, I noticed that the given dataset, reading the excel sheet as a dataframe, includes the title and extra empty cells as the first couple rows. I will be cleaning the data by setting up the correct indices, and starting row. The indices in the original dataframe I see is in row 4. The columns seem to begin from the third one, because the second column indicates the observation number, which would just be default indices for each row.

- Taking the column names into a list
- Dropping correct number of rows and columns

I thought it was important to understand each column and its data types before exploring different things I could investigate given the data. Now that the dataframe is set up, we'll be making sure that the types of input values are correct. The following is the list of columns in order and their data type as well as whether they are categorical(nominal/ordinal), or quantitative.

- Year -- ordinal
- Month -- nominal
- US State -- nominal
- Postal Code (of states) -- nominal
- NERC Region (North American Electric Reliability Corporation) -- nominal
- Climate Region -- nominal
- Anomaly Level (cold and warm episodes by season) -- quantitative
- Climate Category (warm/cold/normal) -- nominal
- Outage Start Date -- ordinal
- Outage Start Time -- ordinal
- Outage Restoration Date -- ordinal
- Outage Restoration Time -- ordinal
- Cause Category -- nominal
- Cause Category Detail -- nominal
- Hurricane Names (if outage due to hurricane, its name) -- nominal
- Outage Duration (in minutes) -- quantitative
- Demand Loss (in MW - amount peak demand lost) -- quantitative
- Customers Affected -- quantitative
- Res Price (monthly price in residential sector) -- quantitative
- Com Price (commercial sector) -- quantitative

- Ind Price (industrial sector) -- quantitative
- Total Price (in state) -- quantitative
- Res Sales (electricity consumption in residential sector) -- quantitative
- Com Sales (commercial sector) -- quantitative
- Ind Sales (industrial sector) -- quantitative
- Total Sales (in state) -- quantitative
- Res Percentage Consumption (percentage compared to total in state in residential sector) -- quantitative
- Com Percentage Consumption (commercial sector) -- quantitative
- Ind Percentage Consumption (industrial sector) -- quantitative
- Res Customers (annual number customers served in residential sector) -- quantitative
- Com Customers (commercial sector) -- quantitative
- Ind Customers (industrial sector) -- quantitative
- Total Customers (in state) -- quantitative
- Res Customers Percentage (percent residential customers served in state) -- quantitative
- Com Customers Percentage (commercial customers) -- quantitative
- Ind Customers Percentage (industrial customers) -- quantitative
- Per Capita Real Gross State Product in State -- quantitative
- PC Real GSP in US -- quantitative
- PC Real GSP Relative (state & US) -- quantitative
- PC Real GSP Change (% change from prev year) -- quantitative
- Utility Real GSP (contributed by utility industry) -- quantitative
- Total Real GSP -- quantitative
- Utility Contribution (% in state) -- quantitative
- Percentage Income Utility of USA (% total earnings of US utility sector) -- quantitative
- Population -- quantitative
- Population Percentage Urban -- quantitative
- Population Percentage Urban Clusters -- quantitative
- Population Density Urban Areas -- quantitative
- Population Density Urban Clusters -- quantitative
- Population Density Rural Areas -- quantitative
- Area Percentage Urban Areas -- quantitative
- Area Percentage Urban Clusters -- quantitative
- Percentage Land -- quantitative
- Percentage Water Total -- quantitative
- Percentage Water Inland -- quantitative

The part that struck my interest is where I decided to make plots and see how the outage durations could display different distributions.

Assessment of Missingness

In the Cleaning and EDA portion of my project, I noticed that I had come across a problem of not being able to convert some columns into datetime. This could be due to the missingness.

Looking at the outage start and restoration date and times, there are some missing data. This would be ignorable missing data, and MCAR (missing completely at random) because the missing data on start or restoration times of the outages is not conditional / associated to the actual value. There seems to be no particular reason why certain values are missing, thus we can remove those rows while investigating the duration of outages. This also applies to the column 'OUTAGE.DURATION', essentially difference between start and restoration of outage.

Another set of missing values would be related to cause of outage. The missing values in columns 'CAUSE.CATEGORY.DETAIL' and 'HURRICANE.NAMES'. 'CAUSE.CATEGORY.DETAIL' is a column that is MCAR since the missing value in this column is not associated with other fields. Adding the detail is random, not directly related to cause of outage. 'HURRICANE.NAMES' is a column that is MAR (missing at random) where the missing value depends on values of other fields, not its own. If the cause is due to a hurricane, there would be a hurricane name, but if not caused by a hurricane, value would be null.

Hypothesis Test

From the Cleaning and EDA part, we can develop a hypothesis test and further investigate how urban regions could be related to outage durations.

- **Null hypothesis**: In the reported outage durations, the urban regions and non-urban regions have the same distribution.
- **Alternative hypothesis**: In the reported outage durations, the urban regions typically have longer outage durations than non-urban regions.
- **Test statistic**: Differences in means (this is the best way we can compare the distributions)

As noted earlier, I decided to differentiate between urban and non-urban regions by taking the mean of percentage of urban population, and regions that are greater than the mean are considered urban regions, and regions that are equal or less than the mean percentage urban population, are considered to be non-urban regions.

Now that I had the observed difference in means, I tested through simulation using a permutation test. A permutation test is ideal because a null hypothesis will only be sampling from the same distribution. On the other hand, using a permutation test will allow me to shuffle the groups and randomy assign the outage durations to observe the distributions. I wanted to see if the difference in means is based on random chance in the assignment.

The purpose of conducting simulations is to compute difference in means and compute the test statistics shuffling multiple times. I used 500 repetitions because I think it is sufficient enough to get random distributions to compare with the observed difference.

Hypothesis Test Summary

In our permutation test, we realize that the urban regions and non-urban regions and their outage durations have the same distribution through taking the difference in means. This is through interpreting the p-value, being 0.095, which is above the level of significance which i set to be 0.05. Thus, we fail to reject the null hypothesis. In conclusion, I found that even though it could look like there is an association between urban regions and the length of outage durations, it is random.

For future investigations, I believe that there are different parts of the data that could be investigated besides the population and outages, but possibly taking a look at the cause of outages and its duration, or how the weather

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 figu
res
```

In [2]:
```
filepath_outage = os.path.join('data', 'outage.xlsx')
outage_data = pd.read_excel(filepath_outage)
```

In [3]:
```
outage_data.head(10)
```

Out[3]:

	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 I
7	NaN	2	2014	5	Minnesota	MN	MRO	East I
8	NaN	3	2010	10	Minnesota	MN	MRO	East I
9	NaN	4	2012	6	Minnesota	MN	MRO	East I

10 rows × 57 columns

Cleaning and EDA

In [4]:
```
column_names_lst = outage_data.iloc[4].values.tolist()[2:]
```

In [5]:
```
outage_data_adjusted = outage_data.iloc[6:,2:]
```

In [6]:
```
outage_data_adjusted.shape[0]
```

Out[6]: 1534

In [7]:
```
# reassigning index to default
outage_data_adjusted.reset_index(inplace=True, drop=True)
```

In [8]:
```
outage_data_adjusted.columns = column_names_lst
```

In [9]:
```
outage_data_adjusted.sample(3)
```

Out[9]:

	YEAR	MONTH	U.S._STATE	POSTAL.CODE	NERC.REGION	CLIMATE.REGION	ANOMALY.LE
545	2003	9	Maryland	MD	RFC	Northeast	
184	2007	9	Texas	TX	SPP	South	
419	2011	11	Washington	WA	WECC	Northwest	

3 rows × 55 columns

One area that we can investigate, looking at the types of columns, is the duration of the power outage. We have the outage start date and time, as well as outage restoration date and time. I think it would be best to combine the date and time columns into one, to get the exact date-time together.

In [81]:
```
types_df_outages = outage_data_adjusted.dtypes
```

In [15]:
```
outage_data_adjusted['OUTAGE.START.DATE']= outage_data_adjusted['OUTAGE.
START.DATE'].apply(pd.to_datetime)
```

In [20]:
```
#outage_data_adjusted['OUTAGE.START.TIME']= outage_data_adjusted['OUTAG
E.START.TIME'].apply(pd.to_datetime)
```

I noticed that I could be looking at the column 'OUTAGE.DURATION' because that would be the difference between outage start and restoration date and times.

In [18]:
```
#outage_data_adjusted.dtypes
outage_data_adjusted.head(2)
```

Out[18]:

	YEAR	MONTH	U.S._STATE	POSTAL.CODE	NERC.REGION	CLIMATE.REGION	ANOMALY.LEVE
0	2011	7	Minnesota	MN	MRO	East North Central	-0.
1	2014	5	Minnesota	MN	MRO	East North Central	-0.

2 rows × 55 columns

We can try to merge the rows that have the same climate region and find the average outage duration to see if there are any patterns or correlation with region and the outage.

In [23]:
```
outage_duration_nona = outage_data_adjusted[outage_data_adjusted['OUTAG
E.DURATION'].notna()]
```

In [29]:
```
outage_duration_nona['OUTAGE.DURATION'].astype(str).astype(int)
```

Out[29]:
```
0       3060
1          1
2       3000
3       2550
4       1740
        ...
1526       0
1528     220
1529     720
1531      59
1532     181
Name: OUTAGE.DURATION, Length: 1476, dtype: int64
```

In [80]:
```
typesindf_outage = outage_duration_nona.dtypes
```

In [31]:
```
outage_duration_nona.groupby(['CLIMATE.REGION'])['OUTAGE.DURATION'].mean
().sort_values(ascending=False)
```

Out[31]:
```
CLIMATE.REGION
East North Central    5352.043796
Northeast             2991.656977
South                 2846.100917
Central               2701.130890
Southeast             2217.686667
West                  1628.331707
Southwest             1566.136364
Northwest             1284.500000
West North Central     696.562500
Name: OUTAGE.DURATION, dtype: float64
```

Observing the series I got above, we see that the average outage duration is the longest in the East North Central region in the US, and the shortest average outage duration is in the West North Central region. It is slightly difficult to tell between the Northeast, South, and Central regions, as well as for West and Southwest regions.

Another thing we can try investigating is to see if the urban areas or rural areas have more outage duration.

In [33]:
```
outage_duration_nona.head(2)
```

Out[33]:

	YEAR	MONTH	U.S._STATE	POSTAL.CODE	NERC.REGION	CLIMATE.REGION	ANOMALY.LEVE
0	2011	7	Minnesota	MN	MRO	East North Central	-0.
1	2014	5	Minnesota	MN	MRO	East North Central	-0.

2 rows × 55 columns

In [36]:
```
outage_duration_nona['POPPCT_URBAN'].astype(str).astype(float)
```

Out[36]:
```
0       73.27
1       73.27
2       73.27
3       73.27
4       73.27
        ...
1526    70.58
1528    70.58
1529    59.90
1531    56.65
1532    56.65
Name: POPPCT_URBAN, Length: 1476, dtype: float64
```

In [65]:
```
# when I took the dataframe and used .groupby(), it was not a good way o
f making the observation.
# the numbers were hard to interpret in a series, so instead, taking a l
ook at distributions would be better.
grouped_series_urban = outage_duration_nona.groupby(['POPPCT_URBAN'])['O
UTAGE.DURATION'].mean().sort_values(ascending=True)
```

In [38]:
```
mean_urban_pop = outage_duration_nona['POPPCT_URBAN'].mean()
mean_urban_pop
```

Out[38]:
```
80.94572493224909
```

In order to see any patterns or make observations based on urban and non-urban regions, I needed to find a way to distinguish between what is an urban region and what is not.

I decided to separate the regions by taking the mean percentage population that is urban and make that a threshold and split into urban or non-urban. An urban region would be regions with percentage population of urban region greater than the mean, and anything equal to or below is a non-urban region.

In [41]:
```
outage_duration_nona['IS_URBAN'] = outage_duration_nona['POPPCT_URBAN'].
apply(lambda x: True if x>mean_urban_pop else False)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
```

In [45]:
```
outage_duration_nona.head(5)
```

Out[45]:

	YEAR	MONTH	U.S._STATE	POSTAL.CODE	NERC.REGION	CLIMATE.REGION	ANOMALY.LEVE
0	2011	7	Minnesota	MN	MRO	East North Central	-0.
1	2014	5	Minnesota	MN	MRO	East North Central	-0.
2	2010	10	Minnesota	MN	MRO	East North Central	-1.
3	2012	6	Minnesota	MN	MRO	East North Central	-0.
4	2015	7	Minnesota	MN	MRO	East North Central	1.

5 rows × 56 columns

In [46]:
```
df_urban_outagedur = outage_duration_nona[['OUTAGE.DURATION', 'IS_URBAN'
]]
```

In [51]:
```
df_urban_outagedur.groupby('IS_URBAN').mean()
```

Out[51]:

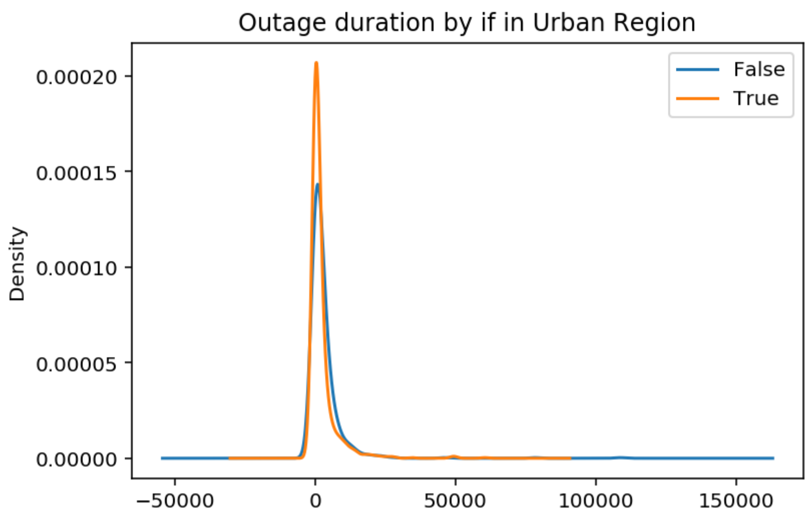
	OUTAGE.DURATION
IS_URBAN	
False	3047.684553
True	2323.765389

In [53]:
```
df_urban_outagedur['OUTAGE.DURATION']
```

Out[53]:
```
0       3060
1          1
2       3000
3       2550
4       1740
        ...
1526       0
1528     220
1529     720
1531      59
1532     181
Name: OUTAGE.DURATION, Length: 1476, dtype: int64
```

In [52]:
```
title = 'Outage duration by if in Urban Region'
(
    df_urban_outagedur
    .groupby('IS_URBAN')['OUTAGE.DURATION']
    .plot(kind='kde',legend=True, subplots=False, title=title)
)
```

Out[52]:
```
IS_URBAN
False    AxesSubplot(0.125,0.125;0.775x0.755)
True     AxesSubplot(0.125,0.125;0.775x0.755)
Name: OUTAGE.DURATION, dtype: object
```

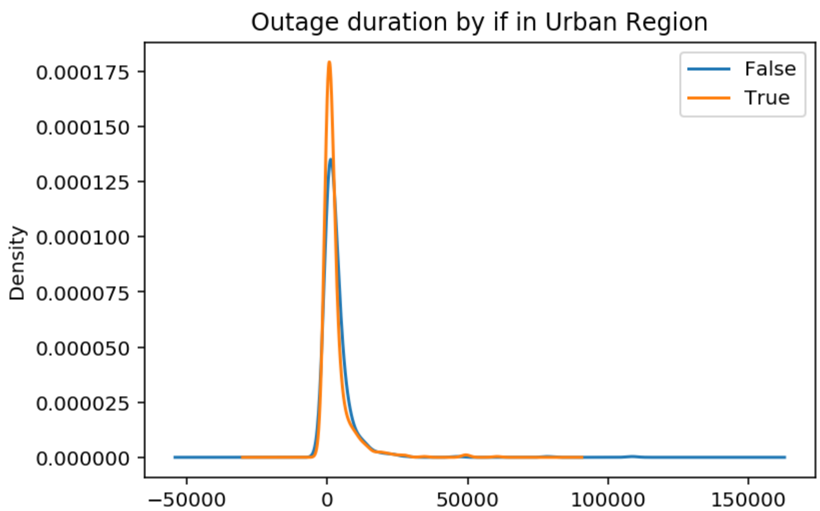


Plotting the outage duration by region if urban or not, we see that there is a peak around 0, as outage duration, and we are interested in outages that have occured, so we want to remove data that includes outage duration of 0 because that means there was no outage.

In [61]:
```
df_urban_outagedur = df_urban_outagedur[df_urban_outagedur['OUTAGE.DURAT
ION']!=0]
```

In [62]:
```
title = 'Outage duration by if in Urban Region'
(
    df_urban_outagedur
    .groupby('IS_URBAN')['OUTAGE.DURATION']
    .plot(kind='kde',legend=True, subplots=False, title=title, )
)
```

Out[62]:
```
IS_URBAN
False    AxesSubplot(0.125,0.125;0.775x0.755)
True     AxesSubplot(0.125,0.125;0.775x0.755)
Name: OUTAGE.DURATION, dtype: object
```



Looking at the plot above, we can formulate a hypothesis test to investigate this correlation between outage duration and urban regions.

With our data, we could also take a look at the cause of outage, and see if there is an association to duration of outage.

In [79]:
```
outage_duration_nona.groupby('CAUSE.CATEGORY')['OUTAGE.DURATION'].mean()
.sort_values(ascending=False)
```

Out[79]:
```
CAUSE.CATEGORY
fuel supply emergency            13484.026316
severe weather                    3883.985215
equipment failure                 1816.909091
public appeal                     1468.449275
system operability disruption      728.869919
intentional attack                 429.980149
islanding                          200.545455
Name: OUTAGE.DURATION, dtype: float64
```

From the series shown above, it is possible to see how the cause of outage relates to the length of duration of outage. We can observe that the fuel supply emergency has the greatest average outage duration, and islanding seems to have a short average outage duration.

Assessment of Missingness

In the Cleaning and EDA portion of my project, I dealt with the missingness and went ahead and removed rows with null values that I could ignore when analyzing my data.

Hypothesis Test

First thing to do in a hypothesis test is to get the observed difference in means to compare the results with.

In [67]:
```
means_outages_urban_table = df_urban_outagedur.groupby('IS_URBAN').mean
()
means_outages_urban_table
```

Out[67]:

	OUTAGE.DURATION
IS_URBAN	
False	3340.918004
True	2801.980392

In [68]:
```
observed_difference_means = means_outages_urban_table.diff().iloc[-1,0]
observed_difference_means
```

Out[68]:
```
-538.9376114081997
```

In [69]:
```
df_urban_outagedur
```

Out[69]:

	OUTAGE.DURATION	IS_URBAN
0	3060	False
2	3000	False
3	2550	False
4	1740	False
5	1860	False
...	...	...
1525	870	False
1528	220	False
1529	720	False
1531	59	False
1532	181	False

1275 rows × 2 columns

In [70]:
```
# the column that we want to shuffle is the outage durations
shuffled_durations = (
    df_urban_outagedur['OUTAGE.DURATION']
    .sample(replace=False, frac=1)
    .reset_index(drop=True)
)

df_with_shuffled = (
    df_urban_outagedur
    .assign(**{'Shuffled Outage Durations': shuffled_durations})
)
# we are adding a column with shuffled durations, this is one simulation
df_with_shuffled.head(5)
```

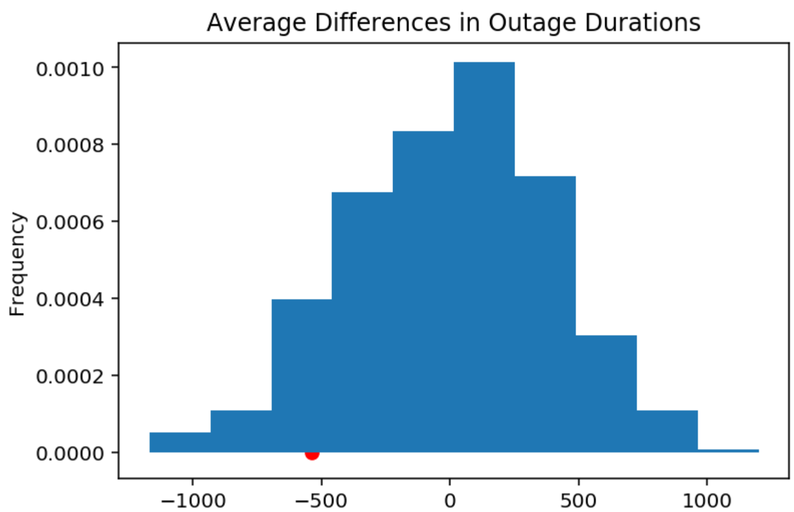
Out[70]:

	OUTAGE.DURATION	IS_URBAN	Shuffled Outage Durations
0	3060	False	2775.0
2	3000	False	1135.0
3	2550	False	618.0
4	1740	False	5820.0
5	1860	False	4260.0

In [71]:
```
# we know that one shuffle is not enough, so we will be doing this 500 t
imes
num_repetitions = 500
# we will calculate the difference in means and putting them into a list
difference_means_lst = []
for i in range(num_repetitions):
    # shuffling outage durations
    shuffled_durations = (
        df_urban_outagedur['OUTAGE.DURATION']
        .sample(replace=False, frac=1)
        .reset_index(drop=True)
    )

    df_with_shuffled = (
        df_urban_outagedur
        .assign(**{'Shuffled Outage Durations': shuffled_durations})
    )
    # compute difference in means
    group_means = (
        df_with_shuffled
        .groupby('IS_URBAN')
        .mean()
        .loc[:,'Shuffled Outage Durations']
    )
    difference_means = group_means.diff().iloc[-1]
    difference_means_lst.append(difference_means)
```

In [73]:
```
title_diff = 'Average Differences in Outage Durations'
pd.Series(difference_means_lst).plot(kind='hist',density=True,title=titl
e_diff)
plt.scatter(observed_difference_means,0,color='red',s=40);
```



Interpreting the plot above, we can see that the difference in means between the observed and sampled is not too significant. We can try to calculate the p-value before making the conclusion, where the p-value is the probability of seeing difference of means being at least as extreme as observed, under the null hypothesis.

In [76]:
```
p_val_outages = np.count_nonzero(difference_means_lst<=observed_differen
ce_means) / num_repetitions
p_val_outages
```

Out[76]: 0.094

Above, I calculated the p-value of our permutation test and what I got was 0.094. I would consider p-value less than 0.05 to be an indication to reject the null hypothesis.

In my result, I got a large p-value, 0.094, thus we can conclude that we fail to reject the null hypothesis

In [ ]:
```

```