2021

Research Assistant Programming Assessment

UNIVERSITY OF OTTAWA
EUGENE TROSTIN

BANK OF CANADA | Selection Committee

Research Assistant Programming Assessment 2021-22

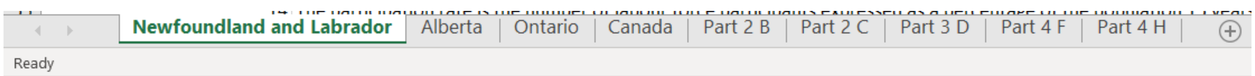
As a research assistant at the Bank of Canada, you will be required to assist economists and researchers with various data-related work. This includes obtaining datasets, checking their accuracy, and manipulating them to make it easier to perform statistical analysis. The following questions are examples of the kind of work that you will be expected to perform at the Bank.

Question 1:

For Question 1, you should show all your work and any external code used for part 4 in the same, clearly laid out, Excel document.

Part 1: Gathering Data

a) *Using StatsCan Table 14-10-0287-01, download total population, total employment, and total unemployment for both sexes (combined) over the age of 15 in Newfoundland and Labrador, Alberta, Ontario, and Canada from January 1977 until December 2019 (inclusive). All variables should be seasonally adjusted.*



For convenience all the requested data has been saved and imported into the submitted Excel file. Solutions for specific questions are indicated by its name.

Part 2: Transforming Data

b) *Convert total employment in Ontario to a quarterly frequency by taking the average monthly values. Which quarter resulted in the largest quarter-over-quarter percentage increase (not annualized) in employment?*

Ontario's	Frequency	Reference Period	**Value**	**Q/Q Change**
Total Employment	Quarterly	Q3 1983	4301.066667	2.26%

I found that Question 1 uses all the data request in part a) except for Canada. Just in case I've calculated Canada's Total Employment aswell.

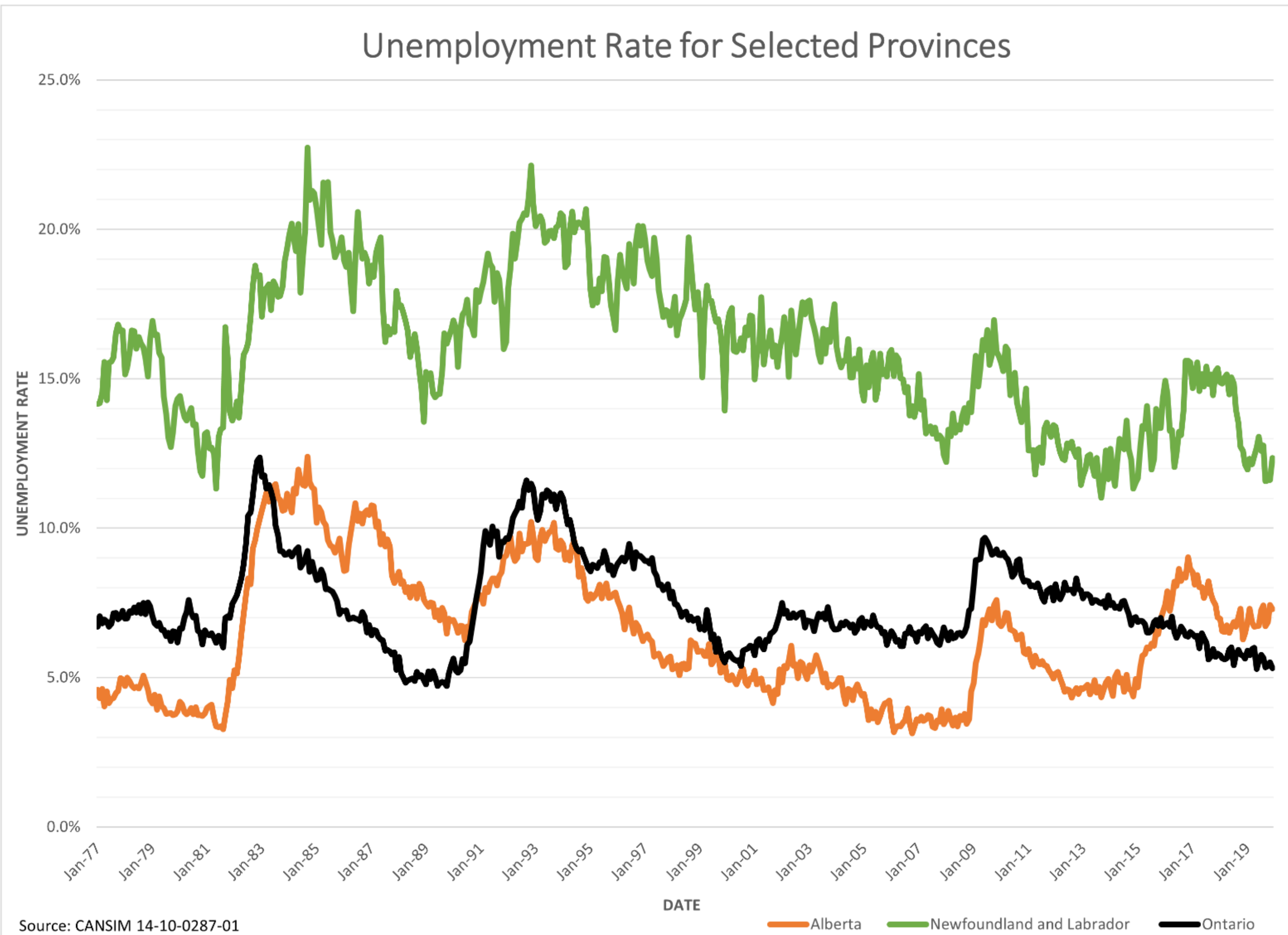
Canada's	Frequency	Reference Period	**Value**	**Q/Q Change**
Total Employment	Quarterly	Q3 1983	11124.7	1.42%

c) For each of the 3 provinces, calculate the monthly unemployment rate over the entire period

- It's unnecessary to calculate the monthly unemployment rate because its already has been calculated by Statistics Canada and can be easily copy pasted
- Just in case, I've calculated the monthly unemployment rate using this formula $\frac{Unemployed}{Total\ Labour\ \mathrm{Force}} = Unemployment\ Rate$ giving the same numbers

Part 3: Interpreting Data

d) For the 3 unemployment rates series you calculated in part 2c, create 1 chart showing this data that could be used in a report or presentation



e) *Using the chart you just created, answer the following questions:*

  i. *What pattern do you see in Alberta starting in late 2014? Name 1 reason why this may have happened.*

- In 2014, there has been a large rise in construction workers. Typically, these workers are seasonal, and these jobs have low worker's retention and go through seasons of no work, like the winter.

Industries

Construction Industry had the largest gain in employment

The Construction industry had the largest increase in employment in 2014, rising by 12,400. This increase accounted for 25.6% of all employment gains in Alberta in 2014. Transportation and Warehousing had the second largest employment increase of 10,100, which represents 20.9% of all the provincial gains in employment in 2014. Employment in the Professional, Scientific and Technical Services industry rose by 8,200, accounting for 16.9% of all employment increases in the province (See Figure 13).

Employment was lower in six industries in 2014: Retail Trade, down 10,100; Agriculture, down 2,800; Information, Culture and Recreation, down 2,300; Public Administration, down 1,600; Utilities, down 400; and Educational Services, down 200.

The three industries with the lowest unemployment rate in 2013 were: Health Care and Social Assistance, 1.6%; Finance, Insurance, Real Estate and Leasing, 1.8%; and Professional, Scientific and Technical Services, 2.1%.

Figure 13

Alberta Labour Force Statistics by Industry, 2014

Industry Group	2014 Employment	Change from 2013	Unemployment Rate
Construction	256,400	12,400	5.1%
Health Care and Social Assistance	240,600	7,400	1.6%
Retail Trade	234,400	-10,100	4.9%
Professional, Scientific and Technical Services	184,300	8,200	2.1%
Mining, Quarrying, and Oil and Gas Extraction	175,300	4,700	3.6%
Accommodation and Food Services	150,000	7,300	4.2%
Manufacturing	144,500	1,800	2.5%
Transportation and Warehousing	129,900	10,100	3.3%
Educational Services	124,700	-200	3.6%
Other Services[4]	122,100	7,600	2.9%
Finance, Insurance, Real Estate and Leasing	104,700	500	1.8%
Public Administration	88,100	-1,600	*
Wholesale Trade	86,100	4,600	*
Business, Building and Other Support Services	79,100	900	4.8%
Information, Culture and Recreation	72,500	-2,300	3.5%
Agriculture	60,600	-2,800	*
Utilities	18,300	-400	*
Forestry and Logging with Support Activities	3,000	300	*

[4]This sector comprises establishments not classified to any other sector, primarily engaged in repairing, or performing general or routine maintenance on motor vehicles, machinery, equipment, and other products to ensure that they work efficiently; providing personal care services, funeral services, laundry services, and other services to individuals, such as pet care services and photo finishing services; organizing and promoting religious activities; supporting various causes through grant-making, advocating (promoting) various social and political causes, and promoting and defending the interests of their members. Private households are also included.
**Insufficient Data*

ii. *Does one of the provinces seem different from the other 2? Name 2 reasons why this might be.*

- Unlike the other two provinces Ontario sees a gradual decline in unemployment rates
    - There's more labour participation
    - There's lower unemployment

Part 4: Computing Descriptive Statistics and Regressions

f) *Download series found at the below links from 1965Q1-2019Q4*

   i. *Seasonally Adjusted Quarterly Unemployment Rate, Canada*

   ii. *Seasonally Adjusted Quarterly Real (Chained 2012) Gross domestic product at market prices, Canada (v62305752)*

g) *Compute the quarter-over-quarter growth[1] of GDP and the quarterly percentage point change in the unemployment rate.*

Refer to Excel sheet

h) *What is the correlation between the two series calculated in part g)?*

```
. pwcorr GDP_Growth_Rate Unemployment_Rate

             | GDP_Gr~e Unempl~e
-------------+------------------
GDP_Growth~e |   1.0000
Unemployme~e |  -0.5128   1.0000
```

- Pearson's correlation test indicates that this is a negative correlation or considerable magnitude

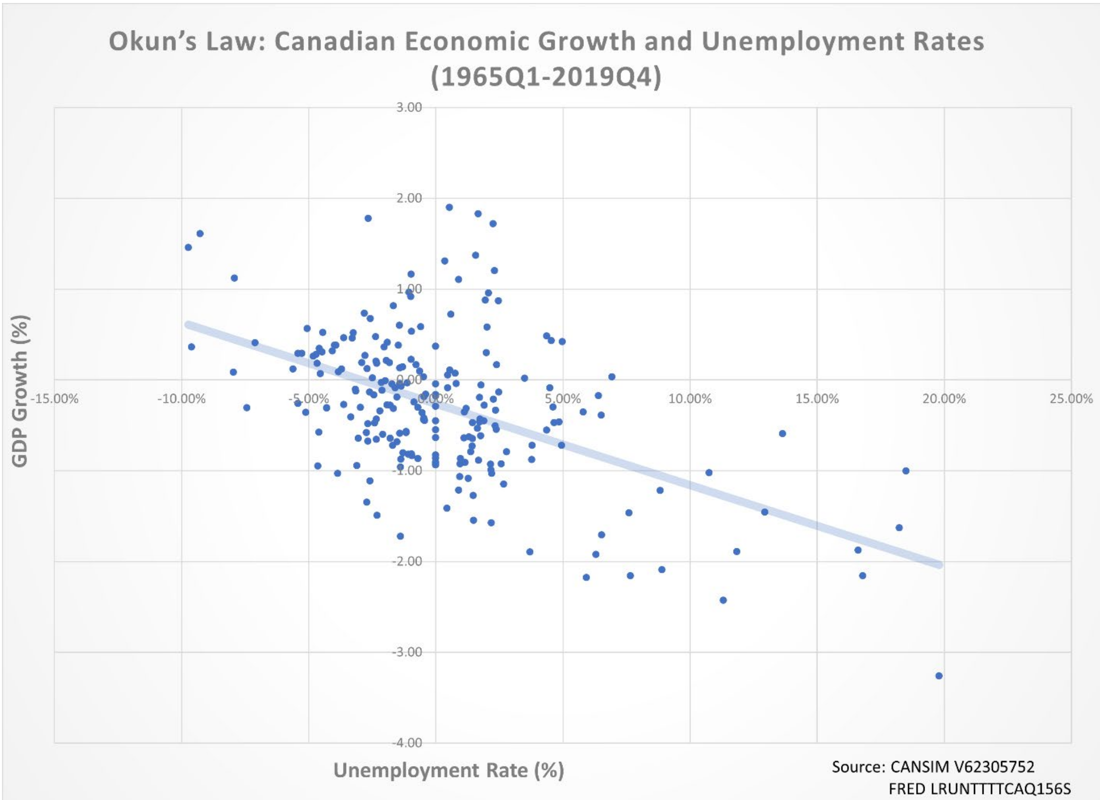
i) *Perform an ordinary least squares regression (of the form yt = a*xt + c) of quarter-over-quarter real GDP growth (dependent variable or yt) and the quarterly percentage point change in the unemployment rate (independent variable or xt).*

```
. reg GDP_Growth_Rate Unemployment_Rate,robust

Linear regression                               Number of obs     =        219
                                                F(1, 217)         =      73.26
                                                Prob > F          =     0.0000
                                                R-squared         =     0.2630
                                                Root MSE          =     .69506

------------------------------------------------------------------------------------
                  |               Robust
  GDP_Growth_Rate |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
------------------+-----------------------------------------------------------------
Unemployment_Rate |  -8.957177   1.046494    -8.56   0.000    -11.01977   -6.894582
            _cons |    99.7376    .046493  2145.22   0.000     99.64597    99.82924
------------------------------------------------------------------------------------
```

[1] The formula is $100*(x_t/x_{t}-1)-1$

i. What is this relationship usually called in economics?

This relationship between GDP growth and Unemployment rate describes Okun's law. Having more GDP, or in other words more production would lead to lower unemployment rates.



ii. Give a brief interpretation of the regression results.

- An increase in the unemployment rate by one 1% is associated with, on average, a decrease in GDP by approximately 8.96%
- Alternatively, when unemployment rates fall by 1%, on average, it is associated with an increase in GDP by approximately 8.96%

iii. Describe 2 ways the regression could be improved.

- Can pick a smaller sample, having noisy data led to a smaller R^2
- Controlling for recessions

Question 2:

a) *The .csv file attached to this assignment has temperature data for Ottawa going back until 1890. Read the .csv into your code and create a structure (e.g. dataframe, list, matrix, etc) with the following variables:*
   - *Year (LOCAL_YEAR)*
   - *Month (LOCAL_MONTH)*
   - *Day (LOCAL_DAY)*
   - *Mean Temperature (MEAN_TEMPERATURE) Maximum Temperature (MAX_TEMPERATURE) Minimum Temperature (MIN_TEMPERATURE) Total Rain (TOTAL_RAIN)*
   - *Total Snow (TOTAL_SNOW)*

b) *Create a code that sums the total amount of rain for all days in the dataset.*

```
In [9]: #Naming equation
column_name = "Total Rain"
#Get sum for equation
column_sum = df1[column_name].sum()
#Print out message
print(f"The sum of the total amount of rain for all days in the dataset is {round(column_sum)}")
```

```
The sum of the total amount of rain for all days in the dataset is 90527
```

c) *Create a code that counts the number of days that it snowed in the dataset.*

```
In [12]: #Naming equation
column_name1 = "Snow Day"
#Get sum for equation
column_sum1 = df1[column_name1].sum()
#Print out message
print(f"The sum of the total number of days that it snowed is {round(column_sum1)}")
```

```
The sum of the total number of days that it snowed is 6082
```

d) *Create a code that counts the number of days that it snowed in May, June, July, and August in the dataset.*

```
In [14]: #Naming equation
column_name2 = "Summer Snow Day"
#Get sum for equation
column_sum2 = df1[column_name2].sum()
#Print out message
print(f"The sum of the total number of days that it snowed in the summer is {round(column_sum2)}")
```

```
The sum of the total number of days that it snowed in the summer is 9
```

e) *Create a code that computes the annual values for the following variables and saves them in a new structure (e.g. dataframe, list, matrix, etc): Mean Temperature, Min Temperature, Max Temperature.*

f) *Create a code that will add the temperature range as a column to the structure in Part e)*

- The code for e) and f) be seen in the on pages 8-17 or replication .ipynb file
- The requested output data can be found in the excel file

***Bonus**: Display the numerical values from Question 2 parts b), c), d) in a PDF or Word document.*

Works Cited

Furhmann, Ryan. "Unemployment and Economic Growth: Okun's Law." *Investopedia*, 2019, www.investopedia.com/articles/economics/12/okuns-law.asp.

"Labour Force Characteristics, Monthly, Seasonally Adjusted and Trend-Cycle." *Statistica Canada*, 1977, www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1410028701&pickMembers%5B0%5D=1.1&pickMembers%5B1%5D=3.1&pickMembers%5B2%5D=4.1&pickMembers%5B3%5D=5.1&cubeTimeFrame.startMonth=01&cubeTimeFrame.startYear=1977&cubeTimeFrame.endMonth=12&cubeTimeFrame.endYear=2019&referencePeriods=19770101%2C20191201.

Labour Market Review. *Annual Alberta Labour Market Review Employment . Unemployment . Economic Regions Migration . Aboriginal People . Industries Occupations . Education . Demographics*. 2014, open.alberta.ca/dataset/591795c0-ac54-4692-81c4-9f1ee0f1bd27/resource/9fb0d414-c01c-42fe-a90c-7fa901bf2e08/download/2683515-2014-annual-ab-labour-market-review.pdf.

Organization for Economic Co-operation and Development. "Unemployment Rate: Aged 15 and Over: All Persons for Canada." *FRED, Federal Reserve Bank of St. Louis*, 1 Jan. 1955, fred.stlouisfed.org/series/LRUNTTTTCAQ156S.

Statistics Canada. "V62305752 Dataset." *Www150.Statcan.gc.ca*, Governement of Canada, www150.statcan.gc.ca/t1/tbl1/en/sbv.action?vectorNumbers=v62305752&searchOption=1&refPeriodStart=1965-01-01&refPeriodEnd=2019-10-30. Accessed 30 Oct. 2021.

Eugene_Trostin_Replication_Code

October 30, 2021

[1]:
```
#Importing needed data
import pandas as pd
import numpy as np
```

[2]:
```
#Setting address
filefolder = r"C:\Users\eugen\Desktop\BOC-Assigment".replace("\\","/")
#Storing file
Q2=''.join([filefolder,"/climate-daily.csv"])
#Loaded into memory
df=pd.read_csv(Q2)
```

```
C:\ProgramData\Anaconda3\lib\site-
packages\IPython\core\interactiveshell.py:3165: DtypeWarning: Columns (23) have
mixed types.Specify dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

[3]:
```
# Lists # of obs. and # of columns
print("The number obsevation and variables in this dataset " + str(df.shape))

#data colums for names, observations, and data type + print out
display(df.info())

#summary statistics
display(df.describe().round().T)
```

```
The number obsevation and variables in this dataset (47735, 36)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 47735 entries, 0 to 47734
Data columns (total 36 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   x                         47735 non-null  float64
 1   y                         47735 non-null  float64
 2   STATION_NAME              47735 non-null  object
 3   CLIMATE_IDENTIFIER        47735 non-null  int64
 4   ID                        47735 non-null  object
 5   LOCAL_DATE                47735 non-null  object
 6   PROVINCE_CODE             47735 non-null  object
```

```
 7   LOCAL_YEAR                 47735 non-null  int64
 8   LOCAL_MONTH                47735 non-null  int64
 9   LOCAL_DAY                  47735 non-null  int64
 10  MEAN_TEMPERATURE           47714 non-null  float64
 11  MEAN_TEMPERATURE_FLAG      39 non-null     object
 12  MIN_TEMPERATURE            47716 non-null  float64
 13  MIN_TEMPERATURE_FLAG       32 non-null     object
 14  MAX_TEMPERATURE            47714 non-null  float64
 15  MAX_TEMPERATURE_FLAG       39 non-null     object
 16  TOTAL_PRECIPITATION        47711 non-null  float64
 17  TOTAL_PRECIPITATION_FLAG   3878 non-null   object
 18  TOTAL_RAIN                 47725 non-null  float64
 19  TOTAL_RAIN_FLAG            2437 non-null   object
 20  TOTAL_SNOW                 47735 non-null  float64
 21  TOTAL_SNOW_FLAG            2329 non-null   object
 22  SNOW_ON_GROUND             21729 non-null  float64
 23  SNOW_ON_GROUND_FLAG        631 non-null    object
 24  DIRECTION_MAX_GUST         0 non-null      float64
 25  DIRECTION_MAX_GUST_FLAG    0 non-null      float64
 26  SPEED_MAX_GUST             0 non-null      float64
 27  SPEED_MAX_GUST_FLAG        0 non-null      float64
 28  COOLING_DEGREE_DAYS        47714 non-null  float64
 29  COOLING_DEGREE_DAYS_FLAG   39 non-null     object
 30  HEATING_DEGREE_DAYS        47714 non-null  float64
 31  HEATING_DEGREE_DAYS_FLAG   39 non-null     object
 32  MIN_REL_HUMIDITY           0 non-null      float64
 33  MIN_REL_HUMIDITY_FLAG      0 non-null      float64
 34  MAX_REL_HUMIDITY           0 non-null      float64
 35  MAX_REL_HUMIDITY_FLAG      0 non-null      float64
dtypes: float64(19), int64(4), object(13)
memory usage: 13.1+ MB

None

                           count       mean   std        min        25%  \
x                        47735.0      -76.0   0.0      -76.0      -76.0
y                        47735.0       45.0   0.0       45.0       45.0
CLIMATE_IDENTIFIER       47735.0  6105976.0   0.0  6105976.0  6105976.0
LOCAL_YEAR               47735.0     1955.0  38.0     1890.0     1922.0
LOCAL_MONTH              47735.0        7.0   3.0        1.0        4.0
LOCAL_DAY                47735.0       16.0   9.0        1.0        8.0
MEAN_TEMPERATURE         47714.0        6.0  12.0      -35.0       -3.0
MIN_TEMPERATURE          47716.0        1.0  12.0      -39.0       -7.0
MAX_TEMPERATURE          47714.0       11.0  13.0      -32.0        1.0
TOTAL_PRECIPITATION      47711.0        2.0   6.0        0.0        0.0
TOTAL_RAIN               47725.0        2.0   5.0        0.0        0.0
TOTAL_SNOW               47735.0        1.0   2.0        0.0        0.0
SNOW_ON_GROUND           21729.0        6.0  12.0        0.0        0.0
DIRECTION_MAX_GUST           0.0        NaN   NaN        NaN        NaN
```

```
DIRECTION_MAX_GUST_FLAG       0.0        NaN   NaN         NaN         NaN
SPEED_MAX_GUST                0.0        NaN   NaN         NaN         NaN
SPEED_MAX_GUST_FLAG           0.0        NaN   NaN         NaN         NaN
COOLING_DEGREE_DAYS       47714.0        1.0   2.0         0.0         0.0
HEATING_DEGREE_DAYS       47714.0       13.0  11.0         0.0         1.0
MIN_REL_HUMIDITY              0.0        NaN   NaN         NaN         NaN
MIN_REL_HUMIDITY_FLAG         0.0        NaN   NaN         NaN         NaN
MAX_REL_HUMIDITY              0.0        NaN   NaN         NaN         NaN
MAX_REL_HUMIDITY_FLAG         0.0        NaN   NaN         NaN         NaN

                               50%        75%        max
x                            -76.0      -76.0      -76.0
y                             45.0       45.0       45.0
CLIMATE_IDENTIFIER       6105976.0  6105976.0  6105976.0
LOCAL_YEAR                  1955.0     1988.0     2020.0
LOCAL_MONTH                    7.0       10.0       12.0
LOCAL_DAY                     16.0       23.0       31.0
MEAN_TEMPERATURE               7.0       17.0       31.0
MIN_TEMPERATURE                2.0       11.0       25.0
MAX_TEMPERATURE               12.0       22.0       38.0
TOTAL_PRECIPITATION            0.0        2.0      109.0
TOTAL_RAIN                     0.0        1.0      109.0
TOTAL_SNOW                     0.0        0.0       56.0
SNOW_ON_GROUND                 0.0        8.0       97.0
DIRECTION_MAX_GUST             NaN        NaN        NaN
DIRECTION_MAX_GUST_FLAG        NaN        NaN        NaN
SPEED_MAX_GUST                 NaN        NaN        NaN
SPEED_MAX_GUST_FLAG            NaN        NaN        NaN
COOLING_DEGREE_DAYS            0.0        0.0       13.0
HEATING_DEGREE_DAYS           11.0       21.0       53.0
MIN_REL_HUMIDITY               NaN        NaN        NaN
MIN_REL_HUMIDITY_FLAG          NaN        NaN        NaN
MAX_REL_HUMIDITY               NaN        NaN        NaN
MAX_REL_HUMIDITY_FLAG          NaN        NaN        NaN
```

[4]:

```
#Leaving these columns and deleting the rest
df.drop(df.columns.
 ↪difference(['LOCAL_YEAR','LOCAL_MONTH','LOCAL_DAY','MEAN_TEMPERATURE','MIN_TEMPERATURE','MAX_
 ↪1, inplace=True)
```

[5]:

```
#Renaming columns names to replicate layout from page 2
df.rename(columns = {'LOCAL_YEAR':'Year'}, inplace = True)
df.rename(columns = {'LOCAL_MONTH':'Month'}, inplace = True)
df.rename(columns = {'LOCAL_DAY':'Day'}, inplace = True)
df.rename(columns = {'MEAN_TEMPERATURE':'Mean Temperature'}, inplace = True)
df.rename(columns = {'MIN_TEMPERATURE':'Minimum Temperature'}, inplace = True)
df.rename(columns = {'MAX_TEMPERATURE':'Maximum Temperature'}, inplace = True)
```

```
df.rename(columns = {'TOTAL_RAIN':'Total Rain'}, inplace = True)
df.rename(columns = {'TOTAL_SNOW':'Total Snow'}, inplace = True)
```

[6]:
```
#creating copies of modifed dateset
df1=df.copy()
df2=df.copy()
```

[7]:
```
# Lists # of obs. and # of columns
print("The number obsevation and variables in this dataset " + str(df1.shape))

#data colums for names, observations, and data type + print out
display(df1.info())

#summary statistics
display(df1.describe().round().T)
```

```
The number obsevation and variables in this dataset (47735, 8)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 47735 entries, 0 to 47734
Data columns (total 8 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Year                 47735 non-null  int64  
 1   Month                47735 non-null  int64  
 2   Day                  47735 non-null  int64  
 3   Mean Temperature     47714 non-null  float64
 4   Minimum Temperature  47716 non-null  float64
 5   Maximum Temperature  47714 non-null  float64
 6   Total Rain           47725 non-null  float64
 7   Total Snow           47735 non-null  float64
dtypes: float64(5), int64(3)
memory usage: 2.9 MB

None

                       count    mean   std     min     25%     50%     75%  \
Year                 47735.0  1955.0  38.0  1890.0  1922.0  1955.0  1988.0
Month                47735.0     7.0   3.0     1.0     4.0     7.0    10.0
Day                  47735.0    16.0   9.0     1.0     8.0    16.0    23.0
Mean Temperature     47714.0     6.0  12.0   -35.0    -3.0     7.0    17.0
Minimum Temperature  47716.0     1.0  12.0   -39.0    -7.0     2.0    11.0
Maximum Temperature  47714.0    11.0  13.0   -32.0     1.0    12.0    22.0
Total Rain           47725.0     2.0   5.0     0.0     0.0     0.0     1.0
Total Snow           47735.0     1.0   2.0     0.0     0.0     0.0     0.0

                        max
Year                 2020.0
Month                  12.0
Day                    31.0
```

```
Mean Temperature        31.0
Minimum Temperature     25.0
Maximum Temperature     38.0
Total Rain             109.0
Total Snow              56.0
```

[8]:
```
#Show the first and last 5 observation
df1.head().append(df1.tail())
```

[8]:
```
       Year  Month  Day  Mean Temperature  Minimum Temperature  \
0      1890      1    1              -5.3                -14.4
1      1890      1    2               5.6                  2.8
2      1890      1    3              -4.8                -12.8
3      1890      1    4             -10.3                -13.9
4      1890      1    5              -7.0                 -8.9
47730  2020     12   27              -6.0                -10.0
47731  2020     12   28               0.0                 -5.0
47732  2020     12   29              -8.8                -11.5
47733  2020     12   30              -3.5                -11.0
47734  2020     12   31              -1.5                 -4.0

       Maximum Temperature  Total Rain  Total Snow
0                      3.9         0.0         0.0
1                      8.3        15.7         0.0
2                      3.3         2.8         0.0
3                     -6.7         0.0         0.0
4                     -5.0         0.0         5.1
47730                 -2.0         2.0         2.0
47731                  5.0         0.0         0.0
47732                 -6.0         0.0         0.0
47733                  4.0         1.8         2.0
47734                  1.0         0.0         0.0
```

[9]:
```
#Naming equation
column_name = "Total Rain"
#Get sum for equation
column_sum = df1[column_name].sum()
#Print out message
print(f"The sum of the total amount of rain for all days in the dataset is␣
 ↪{round(column_sum)}")
```

```
The sum of the total amount of rain for all days in the dataset is 90527
```

[10]:
```
# Making dummy variable
df1['Snow Day']= df1['Total Snow']
# If zero it didn't snow
df1.loc[df1['Snow Day']==0, 'Snow Day']= 0
# If not zero it snowed
```

```
df1.loc[df1['Snow Day']!=0, 'Snow Day']= 1
```

[11]:
```
df1.head().append(df1.tail())
```

[11]:
```
       Year  Month  Day  Mean Temperature  Minimum Temperature  \
0      1890      1    1              -5.3                -14.4
1      1890      1    2               5.6                  2.8
2      1890      1    3              -4.8                -12.8
3      1890      1    4             -10.3                -13.9
4      1890      1    5              -7.0                 -8.9
47730  2020     12   27              -6.0                -10.0
47731  2020     12   28               0.0                 -5.0
47732  2020     12   29              -8.8                -11.5
47733  2020     12   30              -3.5                -11.0
47734  2020     12   31              -1.5                 -4.0

       Maximum Temperature  Total Rain  Total Snow  Snow Day
0                      3.9         0.0         0.0       0.0
1                      8.3        15.7         0.0       0.0
2                      3.3         2.8         0.0       0.0
3                     -6.7         0.0         0.0       0.0
4                     -5.0         0.0         5.1       1.0
47730                 -2.0         2.0         2.0       1.0
47731                  5.0         0.0         0.0       0.0
47732                 -6.0         0.0         0.0       0.0
47733                  4.0         1.8         2.0       1.0
47734                  1.0         0.0         0.0       0.0
```

[12]:
```
#Naming equation
column_name1 = "Snow Day"
#Get sum for equation
column_sum1 = df1[column_name1].sum()
#Print out message
print(f"The sum of the total number of days that it snowed is␣
 ↪{round(column_sum1)}")
```

```
The sum of the total number of days that it snowed is 6082
```

[13]:
```
#Make dummy variables for May (5), June (6), July (7), and Augaust (8)
df1['Summer Snow Day'] = np.where((df1['Snow Day']==1) & (df1['Month']==8), 1, 0)
df1['Summer Snow Day'] = np.where((df1['Snow Day']==1) & (df1['Month']==7), 1, 0)
df1['Summer Snow Day'] = np.where((df1['Snow Day']==1) & (df1['Month']==6), 1, 0)
df1['Summer Snow Day'] = np.where((df1['Snow Day']==1) & (df1['Month']==5), 1, 0)
```

[14]:
```
#Naming equation
column_name2 = "Summer Snow Day"
#Get sum for equation
column_sum2 = df1[column_name2].sum()
```

```
#Print out message
print(f"The sum of the total number of days that it snowed in the summer is␣
 ↪{round(column_sum2)}")
```

```
The sum of the total number of days that it snowed in the summer is 9
```

[15]:
```
#Save dataframe as excel print out
file_name = 'df1.xlsx'
df1.to_excel(file_name)
print('DataFrame is written to Excel File successfully.')
```

```
DataFrame is written to Excel File successfully.
```

[16]:
```
# Lists # of obs. and # of columns
print("The number obsevation and variables in this dataset " + str(df2.shape))
#data colums for names, observations, and data type + print out
display(df2.info())
#summary statistics
display(df2.describe().round().T)
```

```
The number obsevation and variables in this dataset (47735, 8)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 47735 entries, 0 to 47734
Data columns (total 8 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Year                 47735 non-null  int64
 1   Month                47735 non-null  int64
 2   Day                  47735 non-null  int64
 3   Mean Temperature     47714 non-null  float64
 4   Minimum Temperature  47716 non-null  float64
 5   Maximum Temperature  47714 non-null  float64
 6   Total Rain           47725 non-null  float64
 7   Total Snow           47735 non-null  float64
dtypes: float64(5), int64(3)
memory usage: 2.9 MB
```

```
None
```

```
                       count    mean   std     min     25%     50%     75%  \
Year                 47735.0  1955.0  38.0  1890.0  1922.0  1955.0  1988.0
Month                47735.0     7.0   3.0     1.0     4.0     7.0    10.0
Day                  47735.0    16.0   9.0     1.0     8.0    16.0    23.0
Mean Temperature     47714.0     6.0  12.0   -35.0    -3.0     7.0    17.0
Minimum Temperature  47716.0     1.0  12.0   -39.0    -7.0     2.0    11.0
Maximum Temperature  47714.0    11.0  13.0   -32.0     1.0    12.0    22.0
Total Rain           47725.0     2.0   5.0     0.0     0.0     0.0     1.0
Total Snow           47735.0     1.0   2.0     0.0     0.0     0.0     0.0

                        max
```

```
Year                   2020.0
Month                    12.0
Day                      31.0
Mean Temperature         31.0
Minimum Temperature      25.0
Maximum Temperature      38.0
Total Rain              109.0
Total Snow               56.0
```

[17]:
```
#Dropping month and day columns
df2.drop(df2.columns.difference(['Year','Mean Temperature','Minimum␣
 ↪Temperature','Maximum Temperature']), 1, inplace=True)
```

[18]:
```
#Top 5 and bottom 5
df2.head().append(df2.tail())
```

[18]:

	Year	Mean Temperature	Minimum Temperature	Maximum Temperature
0	1890	-5.3	-14.4	3.9
1	1890	5.6	2.8	8.3
2	1890	-4.8	-12.8	3.3
3	1890	-10.3	-13.9	-6.7
4	1890	-7.0	-8.9	-5.0
47730	2020	-6.0	-10.0	-2.0
47731	2020	0.0	-5.0	5.0
47732	2020	-8.8	-11.5	-6.0
47733	2020	-3.5	-11.0	4.0
47734	2020	-1.5	-4.0	1.0

[19]:
```
#Copies of dataset for calculations
calc1=df2.copy()
calc2=df2.copy()
calc3=df2.copy()
```

[20]:
```
#Group by year and get average temperture
merge1=calc1.groupby(['Year'],as_index=False)['Mean Temperature'].mean()
```

[21]:
```
#Top 5 and bottom 5
merge1.head().append(merge1.tail())
```

[21]:

	Year	Mean Temperature
0	1890	4.707143
1	1891	6.194521
2	1892	5.110109
3	1893	4.354521
4	1894	6.108493
126	2016	7.602186
127	2017	7.485359
128	2018	7.324011

```
129  2019          5.989136
130  2020          8.595286
```

[22]:
```
#Group by year and get minimum observation for temp
merge2=calc2.groupby(['Year'],as_index=False)['Minimum Temperature'].min()
```

[23]:
```
#Top 5 and bottom 5
merge2.head().append(merge2.tail())
```

[23]:

	Year	Minimum Temperature
0	1890	-28.9
1	1891	-32.8
2	1892	-31.1
3	1893	-32.2
4	1894	-32.2
126	2016	-29.2
127	2017	-27.0
128	2018	-28.5
129	2019	-27.0
130	2020	-26.5

[24]:
```
#Group by year and get maximum obsercation for temp
merge3=calc3.groupby(['Year'],as_index=False)['Maximum Temperature'].max()
```

[25]:
```
#Top 5 and bottom 5
merge3.head().append(merge3.tail())
```

[25]:

	Year	Maximum Temperature
0	1890	33.9
1	1891	33.9
2	1892	36.1
3	1893	35.0
4	1894	33.9
126	2016	34.0
127	2017	33.0
128	2018	35.5
129	2019	33.5
130	2020	37.0

[26]:
```
#merge calculation 1 and 2
merged_df = pd.merge(merge1, merge2, on="Year")
#Merge the merged calculation
df3 = pd.merge(merged_df, merge3, on="Year")
```

[27]:
```
#top 5 and bottom 5
df3.head().append(df3.tail())
```

[27]:
```
      Year  Mean Temperature  Minimum Temperature  Maximum Temperature
0     1890          4.707143                -28.9                 33.9
1     1891          6.194521                -32.8                 33.9
2     1892          5.110109                -31.1                 36.1
3     1893          4.354521                -32.2                 35.0
4     1894          6.108493                -32.2                 33.9
126   2016          7.602186                -29.2                 34.0
127   2017          7.485359                -27.0                 33.0
128   2018          7.324011                -28.5                 35.5
129   2019          5.989136                -27.0                 33.5
130   2020          8.595286                -26.5                 37.0
```

[28]:
```
#Generate new column and calculate range
df3['Temperature Range']=df3['Maximum Temperature']-df3['Minimum Temperature']
```

[29]:
```
#top5bottom5
df3.head().append(df3.tail())
```

[29]:
```
      Year  Mean Temperature  Minimum Temperature  Maximum Temperature  \
0     1890          4.707143                -28.9                 33.9
1     1891          6.194521                -32.8                 33.9
2     1892          5.110109                -31.1                 36.1
3     1893          4.354521                -32.2                 35.0
4     1894          6.108493                -32.2                 33.9
126   2016          7.602186                -29.2                 34.0
127   2017          7.485359                -27.0                 33.0
128   2018          7.324011                -28.5                 35.5
129   2019          5.989136                -27.0                 33.5
130   2020          8.595286                -26.5                 37.0

      Temperature Range
0                  62.8
1                  66.7
2                  67.2
3                  67.2
4                  66.1
126                63.2
127                60.0
128                64.0
129                60.5
130                63.5
```

[30]:
```
#Output Excel file
file_name = 'df3.xlsx'
df3.to_excel(file_name)
print('DataFrame is written to Excel File successfully.')
```

```
DataFrame is written to Excel File successfully.
```