DTSC670: Foundations of Machine Learning Models

Module 1

Assignment 2: COVID-19 Data Wrangling

Name: Ejegu Smith

The purpose of this assignment is to hone your data wrangling skills. Your task for this assignment is to perform the data preparation as instructed in the `DTSC670_Assignment_2` pdf listed in Brightspace. After performing all the data preparation tasks outlined in the document, run the code in the "Prepare DataFrames for Grading" section.

You are supplied an Excel file called `BrazilCOVIDData.xlsx` - be sure to put the data file in the same directory as this Jupyter Notebook. *Please note that it may take around 5 minutes to read-in all of the data in this file.*

In [13]:
```
import pandas as pd

fileName = 'BrazilCOVIDData.xlsx'
xls = pd.ExcelFile(fileName)
BrazilCovid_df = pd.read_excel(xls, 'Brazil Covid-19 data')
TempByState_df = pd.read_excel(xls, 'Temperature by State')
#StateStats_df = pd.read_excel(xls, 'Brazil State Stats')
CityArea_df = pd.read_excel(xls, 'City area')
```

In [14]:
```
BrazilCovid_df.head()
```

Out[14]:

	Region	State	Municipality	State-code	Municipality-code	Health-region-code	Health-region-name	Date	Week #	Population as of 2019
0	Brasil	NaN	NaN	76	NaN	NaN	NaN	2020-02-25	9	210147125
1	Brasil	NaN	NaN	76	NaN	NaN	NaN	2020-02-26	9	210147125
2	Brasil	NaN	NaN	76	NaN	NaN	NaN	2020-02-27	9	210147125
3	Brasil	NaN	NaN	76	NaN	NaN	NaN	2020-02-28	9	210147125
4	Brasil	NaN	NaN	76	NaN	NaN	NaN	2020-02-29	9	210147125

In [15]:
```
TempByState_df.head()
```

Out[15]:

	STATE_ABBR	STATE	CITY	IS_CAPITOL	ANNUAL	JAN	FEB	MAR	APR	MAY
0	AC	Acre	NaN	NaN	77.2	78.0	77.9	77.4	77.1	76.0
1	AC	Acre	CRUSEIRO DO SUL, ACRE	NaN	76.8	77.5	77.2	77.2	76.8	76.5
2	AC	Acre	FLORESCENCIA, ACRE	NaN	80.0	80.0	80.0	80.0	79.0	78.0
3	AC	Acre	SENA MADUREIRA, ACRE	NaN	75.0	77.0	77.0	75.0	75.0	73.0
4	AC	Acre	TARAUACÁ, ACRE	NaN	76.8	77.5	77.5	77.5	77.4	76.5

In [16]:
```
CityArea_df.head()
```

Out[16]:

	ST	City	SQ_KM
0	AC	Rio Branco	8835.0
1	AL	Maceió	511.0
2	AP	Macapá	6407.0
3	AM	Manaus	11400.0
4	BA	Salvador	693.8

In [17]:
```
#renaming to appropriate column names
TempByState_df_2 = TempByState_df.rename(columns= {'ANNUAL': 'Temp', '
BrazilCovid_df_2 = BrazilCovid_df.rename(columns= {'Population as of 2
CityArea_df_2 = CityArea_df.rename(columns= {'ST': 'State'})
```

In [8]:
```
BrazilCovid_df
```

In [19]:
```
TempByState_df.head()
```

Out[19]:

	STATE_ABBR	STATE	CITY	IS_CAPITOL	ANNUAL	JAN	FEB	MAR	APR	MAY
0	AC	Acre	NaN	NaN	77.2	78.0	77.9	77.4	77.1	76.0
1	AC	Acre	CRUSEIRO DO SUL, ACRE	NaN	76.8	77.5	77.2	77.2	76.8	76.5
2	AC	Acre	FLORESCENCIA, ACRE	NaN	80.0	80.0	80.0	80.0	79.0	78.0
3	AC	Acre	SENA MADUREIRA, ACRE	NaN	75.0	77.0	77.0	75.0	75.0	73.0
4	AC	Acre	TARAUACÁ, ACRE	NaN	76.8	77.5	77.5	77.5	77.4	76.5

In [20]:
```
joined_df = BrazilCovid_df_2.merge(TempByState_df_2, on=['City', "Stat
joined_df
```

Out[20]:

	Region	State	City	State-code	Municipality-code	Health-region-code	Health-region-name	Date	Week #	Population	...
0	Norte	RO	NaN	11	NaN	NaN	NaN	2020-02-25	9	1777225	...
1	Norte	RO	NaN	11	NaN	NaN	NaN	2020-02-26	9	1777225	...
2	Norte	RO	NaN	11	NaN	NaN	NaN	2020-02-27	9	1777225	...
3	Norte	RO	NaN	11	NaN	NaN	NaN	2020-02-28	9	1777225	...
4	Norte	RO	NaN	11	NaN	NaN	NaN	2020-02-29	9	1777225	...
...	...	...	...	...	...	...	...	...	...	...	...
24	Centro-Oeste	DF	Brasília	53	530010.0	53001.0	DISTRITO FEDERAL	2020-08-20	34	3015268	...
25	Centro-Oeste	DF	Brasília	53	530010.0	53001.0	DISTRITO FEDERAL	2020-08-21	34	3015268	...
26	Centro-Oeste	DF	Brasília	53	530010.0	53001.0	DISTRITO FEDERAL	2020-08-22	34	3015268	...
27	Centro-Oeste	DF	Brasília	53	530010.0	53001.0	DISTRITO FEDERAL	2020-08-23	35	3015268	...
28	Centro-Oeste	DF	Brasília	53	530010.0	53001.0	DISTRITO FEDERAL	2020-08-24	35	3015268	...

29 rows × 34 columns

In [22]:
```
joined_df_2 = joined_df.merge(CityArea_df_2, on=['City', "State"])
joined_df_2
```

Out[22]:

	Region	State	City	State-code	Municipality-code	Health-region-code	Health-region-name	Date	Week #	Populatio
0	Norte	RO	Porto Velho	11	110020.0	11004.0	MADEIRA-MAMORE	2020-03-27	13	52954
1	Norte	RO	Porto Velho	11	110020.0	11004.0	MADEIRA-MAMORE	2020-03-28	13	52954
2	Norte	RO	Porto Velho	11	110020.0	11004.0	MADEIRA-MAMORE	2020-03-29	14	52954
3	Norte	RO	Porto Velho	11	110020.0	11004.0	MADEIRA-MAMORE	2020-03-30	14	52954
4	Norte	RO	Porto Velho	11	110020.0	11004.0	MADEIRA-MAMORE	2020-03-31	14	52954
...	...	...	...	...	...	...	...	...	...	
4072	Centro-Oeste	DF	Brasília	53	530010.0	53001.0	DISTRITO FEDERAL	2020-08-20	34	301526
4073	Centro-Oeste	DF	Brasília	53	530010.0	53001.0	DISTRITO FEDERAL	2020-08-21	34	301526
4074	Centro-Oeste	DF	Brasília	53	530010.0	53001.0	DISTRITO FEDERAL	2020-08-22	34	301526
4075	Centro-Oeste	DF	Brasília	53	530010.0	53001.0	DISTRITO FEDERAL	2020-08-23	35	301526
4076	Centro-Oeste	DF	Brasília	53	530010.0	53001.0	DISTRITO FEDERAL	2020-08-24	35	301526

4077 rows × 35 columns

In [24]:
```
#filtering down data
joined_df_2 = joined_df_2[['State', 'City', 'Population', 'SQ_KM', 'Ac
joined_df_2
```

Out[24]:

	State	City	Population	SQ_KM	Accumulated cases	Temp
0	RO	Porto Velho	529544	34091.0	0	78.1
1	RO	Porto Velho	529544	34091.0	5	78.1
2	RO	Porto Velho	529544	34091.0	5	78.1
3	RO	Porto Velho	529544	34091.0	5	78.1
4	RO	Porto Velho	529544	34091.0	6	78.1
...	...	...	...	...	...	...
4072	DF	Brasília	3015268	5802.0	143759	69.1
4073	DF	Brasília	3015268	5802.0	145452	69.1
4074	DF	Brasília	3015268	5802.0	147127	69.1
4075	DF	Brasília	3015268	5802.0	148998	69.1
4076	DF	Brasília	3015268	5802.0	150519	69.1

4077 rows × 6 columns

In [25]:
```
#checking the data types
joined_df_2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4077 entries, 0 to 4076
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   State              4077 non-null   object 
 1   City               4077 non-null   object 
 2   Population         4077 non-null   object 
 3   SQ_KM              4077 non-null   float64
 4   Accumulated cases  4077 non-null   int64  
 5   Temp               4077 non-null   float64
dtypes: float64(2), int64(1), object(3)
memory usage: 223.0+ KB
```

In [27]:
```
#changing pop to an int
joined_df_2[['Population']] = joined_df_2[['Population']].apply(pd.to_
```

```
/opt/anaconda3/lib/python3.8/site-packages/pandas/core/frame.py:2963:
SettingWithCopyWarning:
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: https://pandas.pydata.org/panda
s-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy)
  self[k1] = value[k2]
```

In [30]:
```
#creating population density column
joined_df_2['Pop_dense'] = joined_df_2['Population']/joined_df_2["SQ_K
joined_df_2
```

```
<ipython-input-30-025b0fc0ec34>:2: SettingWithCopyWarning:
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: https://pandas.pydata.org/panda
s-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy)
  joined_df_2['Pop_dense'] = joined_df_2['Population']/joined_df_2["S
Q_KM"]
```

Out[30]:

	State	City	Population	SQ_KM	Accumulated cases	Temp	Pop_dense
0	RO	Porto Velho	529544	34091.0	0	78.1	15.533249
1	RO	Porto Velho	529544	34091.0	5	78.1	15.533249
2	RO	Porto Velho	529544	34091.0	5	78.1	15.533249
3	RO	Porto Velho	529544	34091.0	5	78.1	15.533249
4	RO	Porto Velho	529544	34091.0	6	78.1	15.533249
...	...	...	...	...	...	...	...
4072	DF	Brasília	3015268	5802.0	143759	69.1	519.694588
4073	DF	Brasília	3015268	5802.0	145452	69.1	519.694588
4074	DF	Brasília	3015268	5802.0	147127	69.1	519.694588
4075	DF	Brasília	3015268	5802.0	148998	69.1	519.694588
4076	DF	Brasília	3015268	5802.0	150519	69.1	519.694588

4077 rows × 7 columns

In [31]:
```
#making a list to use as the days column
my_list = list(range(0,151)) * 27
```

In [32]:
```
# turning list into df to convert to days column
my_list = pd.DataFrame(my_list)
my_list
```

Out[32]:

	0
0	0
1	1
2	2
3	3
4	4
...	...
4072	146
4073	147
4074	148
4075	149
4076	150

4077 rows × 1 columns

In [34]:
```
#addiding the days column to the df using the list
new_df = joined_df_2.assign(Days = my_list)
```

In [35]:
```
new_df
```

Out[35]:

	State	City	Population	SQ_KM	Accumulated cases	Temp	Pop_dense	Days
0	RO	Porto Velho	529544	34091.0	0	78.1	15.533249	0
1	RO	Porto Velho	529544	34091.0	5	78.1	15.533249	1
2	RO	Porto Velho	529544	34091.0	5	78.1	15.533249	2
3	RO	Porto Velho	529544	34091.0	5	78.1	15.533249	3
4	RO	Porto Velho	529544	34091.0	6	78.1	15.533249	4
...	...	...	...	...	...	...	...	...
4072	DF	Brasília	3015268	5802.0	143759	69.1	519.694588	146
4073	DF	Brasília	3015268	5802.0	145452	69.1	519.694588	147
4074	DF	Brasília	3015268	5802.0	147127	69.1	519.694588	148
4075	DF	Brasília	3015268	5802.0	148998	69.1	519.694588	149
4076	DF	Brasília	3015268	5802.0	150519	69.1	519.694588	150

4077 rows × 8 columns

In [36]:
```
new_df.shape
```

Out[36]:
```
(4077, 8)
```

In [ ]:
```
#new_df.drop_duplicates(subset=['STATE_ABBR', 'Capitol City'])
```

In [38]:
```
#adding features
new_df['days_cube'] = new_df['Days']**3
new_df['days_sq'] = new_df['Days']**2
new_df['Pop_dens_sq'] = new_df['Pop_dense']**2
```

In [39]:
```
#checking it out with all the features
new_df
```

Out[39]:

	State	City	Population	SQ_KM	Accumulated cases	Temp	Pop_dense	Days	days_cube	
0	RO	Porto Velho	529544	34091.0	0	78.1	15.533249	0	0	
1	RO	Porto Velho	529544	34091.0	5	78.1	15.533249	1	1	
2	RO	Porto Velho	529544	34091.0	5	78.1	15.533249	2	8	
3	RO	Porto Velho	529544	34091.0	5	78.1	15.533249	3	27	
4	RO	Porto Velho	529544	34091.0	6	78.1	15.533249	4	64	
...	...	...	...	...	...	...	...	...	...	
4072	DF	Brasília	3015268	5802.0	143759	69.1	519.694588	146	3112136	
4073	DF	Brasília	3015268	5802.0	145452	69.1	519.694588	147	3176523	
4074	DF	Brasília	3015268	5802.0	147127	69.1	519.694588	148	3241792	
4075	DF	Brasília	3015268	5802.0	148998	69.1	519.694588	149	3307949	
4076	DF	Brasília	3015268	5802.0	150519	69.1	519.694588	150	3375000	

4077 rows × 11 columns

In [41]:
```
# Get Final Features DataFrame
features = new_df[['days_cube', 'days_sq', 'Days', 'Temp','Pop_dens_sq

# Get Final Response DataFrame
response = new_df[['Accumulated cases']]
```

In [45]:
```
import numpy as np
```

Prepare DataFrames for Grading

Do not make changes to the below code

After completing all data preparation tasks, run the following four cells to prepare your DataFrame for grading by:

1. Outputting the `features` and `response` DataFrames (you do not need to print).
2. Using the NumPy around() (https://numpy.org/doc/stable/reference/generated/numpy.around.html) function to round all the values in both DataFrames to ***ZERO decimal places***. You are calling these `features_round` and `response_round`, respectively.
3. Computing the sum of every column for both `features_round` and `response_round`, and saving those values as `features_final` and `response_final`.

   Finally, you are printing your final answer using the `print()` function.

   Be sure to run all cells of your notebook prior to submitting, so that all output is rendered, visible and there are no error messages.

In [42]:
```
features
```

Out[42]:

	days_cube	days_sq	Days	Temp	Pop_dens_sq	Pop_dense	Population
0	0	0	0	78.1	241.281832	15.533249	529544
1	1	1	1	78.1	241.281832	15.533249	529544
2	8	4	2	78.1	241.281832	15.533249	529544
3	27	9	3	78.1	241.281832	15.533249	529544
4	64	16	4	78.1	241.281832	15.533249	529544
...	...	...	...	...	...	...	...
4072	3112136	21316	146	69.1	270082.464872	519.694588	3015268
4073	3176523	21609	147	69.1	270082.464872	519.694588	3015268
4074	3241792	21904	148	69.1	270082.464872	519.694588	3015268
4075	3307949	22201	149	69.1	270082.464872	519.694588	3015268
4076	3375000	22500	150	69.1	270082.464872	519.694588	3015268

4077 rows × 7 columns

In [43]:
```
response
```

Out[43]:

	Accumulated cases
0	0
1	5
2	5
3	5
4	6
...	...
4072	143759
4073	145452
4074	147127
4075	148998
4076	150519

4077 rows × 1 columns

In [46]:
```
features_round = np.around(features, decimals=0)
features_final = features_round.sum(axis=0)
print(features_final)
```

```
days_cube      3.462902e+09
days_sq        3.067942e+07
Days           3.057750e+05
Temp           3.095500e+05
Pop_dens_sq    6.186755e+10
Pop_dense      1.127728e+07
Population     7.571160e+09
dtype: float64
```

In [47]:
```
response_round = np.around(response, decimals=0)
response_final = response_round.sum(axis=0)
print(response_final)
```

```
Accumulated cases    61281356
dtype: int64
```

In [ ]:
```

```