In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as st
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn import linear_model
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.svm import SVC
from scipy.cluster.hierarchy import linkage, fcluster
from sklearn.cluster import KMeans, DBSCAN
from sklearn.neighbors import KNeighborsClassifier
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.graphics.tsaplots import plot_pacf, plot_acf
from sklearn.metrics import mean_squared_error
```

In [2]:

```
data = pd.read_csv("COVID-19_update.csv")
```

In [3]:

```
data.sort_values(by="Week Number")
```

Out[3]:

	ZIP Code	Week Number	Week Start	Week End	Cases - Weekly	Cases - Cumulative	Case Rate - Weekly	Case Rate - Cumulative	Tests - Weekly	Tests - Cumulative	...	Test Rate - Cumulative	Percent Tested Positive - Weekly
1593	60636	10	03/01/2020	03/07/2020	NaN	NaN	NaN	NaN	0.0	0	...	0.0	0.0
1328	60628	10	03/01/2020	03/07/2020	NaN	NaN	NaN	NaN	2.0	2	...	3.0	0.0
781	60612	10	03/01/2020	03/07/2020	NaN	NaN	NaN	NaN	1.0	1	...	2.9	0.0
1949	60646	10	03/01/2020	03/07/2020	NaN	NaN	NaN	NaN	3.0	3	...	10.7	0.0
1984	60647	10	03/01/2020	03/07/2020	NaN	NaN	NaN	NaN	0.0	0	...	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1712	60606	47	11/15/2020	11/21/2020	17.0	166.0	548.0	5353.1	228.0	4035	...	130119.3	0.1
1773	60616	47	11/15/2020	11/21/2020	167.0	1807.0	307.0	3317.8	2360.0	35412	...	65019.1	0.1
1772	60605	47	11/15/2020	11/21/2020	136.0	978.0	494.0	3553.9	1513.0	19020	...	69115.9	0.1
1714	60615	47	11/15/2020	11/21/2020	115.0	1199.0	277.0	2884.8	2871.0	33286	...	80085.7	0.0
158	60636	47	11/15/2020	11/21/2020	139.0	1770.0	432.0	5496.4	891.0	15655	...	48613.5	0.2

2280 rows × 21 columns

data preparation

data.dtypes

```
In [4]: data.isnull().sum()
```

```
Out[4]: ZIP Code                                0
        Week Number                             0
        Week Start                              0
        Week End                                0
        Cases - Weekly                        175
        Cases - Cumulative                    175
        Case Rate - Weekly                    175
        Case Rate - Cumulative                175
        Tests - Weekly                         30
        Tests - Cumulative                      0
        Test Rate - Weekly                      0
        Test Rate - Cumulative                  0
        Percent Tested Positive - Weekly        0
        Percent Tested Positive - Cumulative    0
        Deaths - Weekly                         0
        Deaths - Cumulative                     0
        Death Rate - Weekly                     0
        Death Rate - Cumulative                 0
        Population                              0
        Row ID                                  0
        ZIP Code Location                      38
        dtype: int64
```

In [5]:

```
data = data.fillna(0)
data
```

Out[5]:

	ZIP Code	Week Number	Week Start	Week End	Cases - Weekly	Cases - Cumulative	Case Rate - Weekly	Case Rate - Cumulative	Tests - Weekly	Tests - Cumulative	...	Test Rate - Cumulative	Percer Teste Positiv Weekl
0	60603	13	03/22/2020	03/28/2020	0.0	0.0	0.0	0.0	3.0	4	...	340.7	0.
1	60603	14	03/29/2020	04/04/2020	0.0	0.0	0.0	0.0	6.0	10	...	851.8	0.
2	60603	15	04/05/2020	04/11/2020	0.0	0.0	0.0	0.0	6.0	16	...	1362.9	0.
3	60603	22	05/24/2020	05/30/2020	0.0	6.0	0.0	511.1	7.0	71	...	6047.7	0.
4	60603	23	05/31/2020	06/06/2020	0.0	6.0	0.0	511.1	19.0	90	...	7666.1	0.
...	...	...	...	...	...	...	...	...	...	...	...	...	.
2275	60655	28	07/05/2020	07/11/2020	31.0	323.0	108.0	1121.4	453.0	4795	...	16647.0	0.
2276	60655	29	07/12/2020	07/18/2020	26.0	349.0	90.0	1211.6	403.0	5198	...	18046.1	0.
2277	60655	30	07/19/2020	07/25/2020	23.0	372.0	80.0	1291.5	382.0	5580	...	19372.3	0.
2278	60655	31	07/26/2020	08/01/2020	26.0	398.0	90.0	1381.8	560.0	6140	...	21316.5	0.
2279	Unknown	36	08/30/2020	09/05/2020	2.0	147.0	0.0	0.0	1562.0	49987	...	0.0	0.

2280 rows × 21 columns

==> 21 variables, int64(6), float64(10), object(5), also we changed Nan value to 0, because when "week start" on March 1st, the value is Nan. It does not proper aggregate so we changed to start 0.

```
In [6]: data = data.drop(['Cases - Weekly', 'Cases - Cumulative', 'Tests - Weekly', 'Tests - Cumulative', 'Deaths - Weekly', 'Deaths - Cumulative'], axis = 1)
```

==> (should find irrelevant for redundant variables), I think should use "Rate case or not", so delete 2 of 4 like (Case-weekly and Cumulative) or (Case Rate - Weekly or Cummulative). But in my opinion to delete totals, not rates as like our project1.

Data Exploration

==> I think we can change every weeks by month

```
In [7]: data.dtypes
```

```
Out[7]: ZIP Code                                object
        Week Number                              int64
        Week Start                              object
        Week End                                object
        Case Rate - Weekly                     float64
        Case Rate - Cumulative                 float64
        Test Rate - Weekly                       int64
        Test Rate - Cumulative                 float64
        Percent Tested Positive - Weekly       float64
        Percent Tested Positive - Cumulative   float64
        Death Rate - Weekly                    float64
        Death Rate - Cumulative                float64
        Population                               int64
        Row ID                                  object
        ZIP Code Location                       object
        dtype: object
```

In [8]:

```
data['Month'] = pd.DatetimeIndex(data['Week Start']).month
data.sort_values(by='Month')
```

Out[8]:

	ZIP Code	Week Number	Week Start	Week End	Case Rate - Weekly	Case Rate - Cumulative	Test Rate - Weekly	Test Rate - Cumulative	Percent Tested Positive - Weekly	Percent Tested Positive - Cumulative	Death Rate - Weekly	Death Rate - Cumulative	Pop
0	60603	13	03/22/2020	03/28/2020	0.0	0.0	256	340.7	0.0	0.0	0.0	0.0	
1232	60623	10	03/01/2020	03/07/2020	0.0	0.0	4	3.5	0.0	0.0	0.0	0.0	
1236	60623	14	03/29/2020	04/04/2020	121.0	210.5	248	508.3	0.4	0.4	5.8	10.5	
398	60643	14	03/29/2020	04/04/2020	204.0	391.0	487	1036.7	0.4	0.3	6.0	6.0	
397	60643	13	03/22/2020	03/28/2020	142.0	186.5	387	549.4	0.3	0.3	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
1740	60613	47	11/15/2020	11/21/2020	341.0	3755.5	5895	73675.5	0.1	0.1	0.0	67.8	
299	60604	45	11/01/2020	11/07/2020	1151.0	6138.1	7033	90792.8	0.2	0.1	0.0	0.0	
1741	60629	47	11/15/2020	11/21/2020	857.0	9344.7	3704	54128.7	0.3	0.2	3.6	130.5	
1502	60666	46	11/08/2020	11/14/2020	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	
1787	60637	45	11/01/2020	11/07/2020	322.0	2811.1	9285	95267.0	0.0	0.0	6.3	65.3	

2280 rows × 16 columns

In [9]:

```
dt = data.groupby("Month")
dt = dt.aggregate(np.mean)
```

In [10]:

```
dt.info()
dt
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9 entries, 3 to 11
Data columns (total 10 columns):
 #   Column                            Non-Null Count  Dtype  
---  ------                            --------------  -----  
 0   Week Number                       9 non-null      float64
 1   Case Rate - Weekly                9 non-null      float64
 2   Case Rate - Cumulative            9 non-null      float64
 3   Test Rate - Weekly                9 non-null      float64
 4   Test Rate - Cumulative            9 non-null      float64
 5   Percent Tested Positive - Weekly  9 non-null      float64
 6   Percent Tested Positive - Cumulative  9 non-null      float64
 7   Death Rate - Weekly               9 non-null      float64
 8   Death Rate - Cumulative           9 non-null      float64
 9   Population                        9 non-null      float64
dtypes: float64(10)
memory usage: 792.0 bytes
```

Out[10]:

	Week Number	Case Rate - Weekly	Case Rate - Cumulative	Test Rate - Weekly	Test Rate - Cumulative	Percent Tested Positive - Weekly	Percent Tested Positive - Cumulative	Death Rate - Weekly	Death Rate - Cumulative	Population
Month										
3	12.0	40.036667	70.231667	160.083333	294.595667	0.127333	0.110333	1.057000	1.304000	46230.216667
4	16.5	161.366667	578.600000	625.075000	2164.183750	0.231250	0.235417	7.789167	23.430000	46230.216667
5	21.0	123.286667	1299.659000	1085.080000	6540.155333	0.128000	0.193333	6.966333	60.476333	46230.216667
6	25.5	44.137500	1594.243333	1417.670833	12076.105000	0.033750	0.141250	2.509167	78.409583	46230.216667
7	29.5	64.758333	1824.582083	1824.520833	18972.335833	0.034167	0.105000	0.773750	83.393333	46230.216667
8	34.0	74.950000	2146.974333	1918.960000	27376.599333	0.038667	0.088333	0.682667	86.475667	46230.216667
9	38.5	68.020833	2461.683333	2058.754167	36224.480417	0.027917	0.077083	0.590417	89.183333	46230.216667
10	42.5	210.579167	3008.524583	3078.204167	46858.276667	0.076250	0.073333	0.983333	92.429167	46230.216667
11	46.0	485.350000	4396.642778	4131.605556	59958.565556	0.131111	0.090000	2.437778	98.848889	46230.216667

==> suppose 3-5 is spring, 6-8 is summer, 9-11 is fall So picked April, July, and October

In [11]:

```
spring = dt.iloc[0:3,:]
summer = dt.iloc[3:6,:]
fall = dt.iloc[6:,:]
```

In [12]:

```
[statistic, pvalue] = st.ttest_ind(spring['Case Rate - Weekly'],summer['Case Rate - Weekly'],equal_var = False)
print(pvalue*2)
```

```
0.6384281788527542
```

In [13]:

```
[statistic, pvalue] = st.ttest_ind(summer['Case Rate - Weekly'],fall['Case Rate - Weekly'],equal_var = False)
print(pvalue*2)
```

```
0.5094128255454738
```

In [14]:

```
[statistic, pvalue] = st.ttest_ind(spring['Case Rate - Weekly'],fall['Case Rate - Weekly'],equal_var = False)
print(2*(pvalue))
```

```
0.7098481982179874
```

==> October is high. p-value is less than 0.05 which means there is sufficient data to reject the null hypothesis that Case Rate by Week of Spring and Fall are equal.

Making Plot

In [15]:

```
data = [[np.average(spring['Case Rate - Weekly']),np.average(spring['Test Rate - Weekly']),np.average(spring[
'Death Rate - Weekly']),'Spring'],
       [np.average(summer['Case Rate - Weekly']),np.average(summer['Test Rate - Weekly']),np.average(summer[
'Death Rate - Weekly']),'summer'],
       [np.average(fall['Case Rate - Weekly']),np.average(fall['Test Rate - Weekly']),np.average(fall['Death
 Rate - Weekly']),'fall']]


data_case = pd.DataFrame(data, columns = ['Case Rate - Weekly', 'Test Rate - Weekly','Death Rate - Weekly','S
easons'])
```

In [16]:

```
data_case
```

Out[16]:

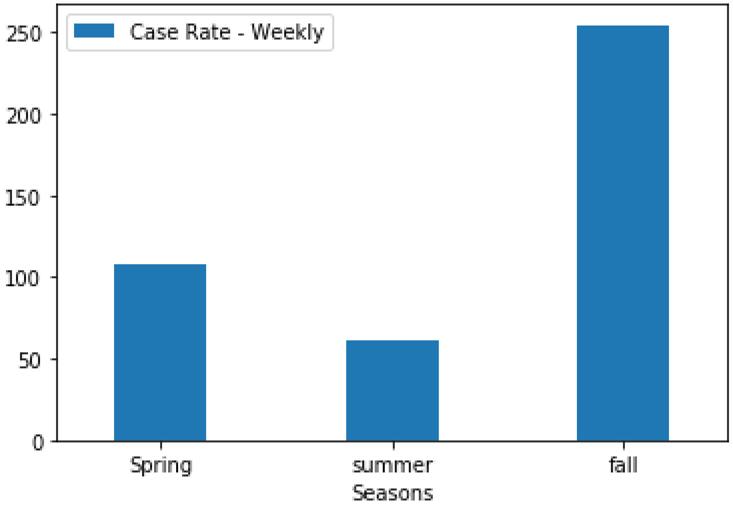
	Case Rate - Weekly	Test Rate - Weekly	Death Rate - Weekly	Seasons
0	108.230000	623.412778	5.270833	Spring
1	61.281944	1720.383889	1.321861	summer
2	254.650000	3089.521296	1.337176	fall

In [17]:

```
data_case.plot.bar(x='Seasons', y = ['Case Rate - Weekly'], width = 0.4, rot = 0)
```

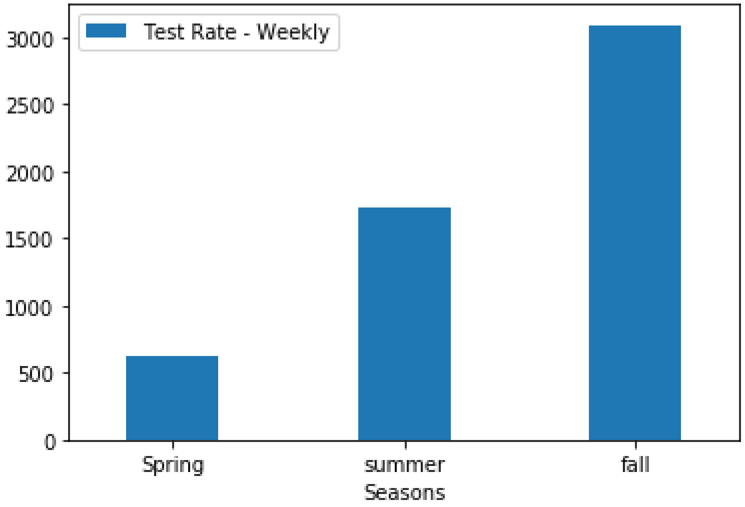
Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x14077135908>
```



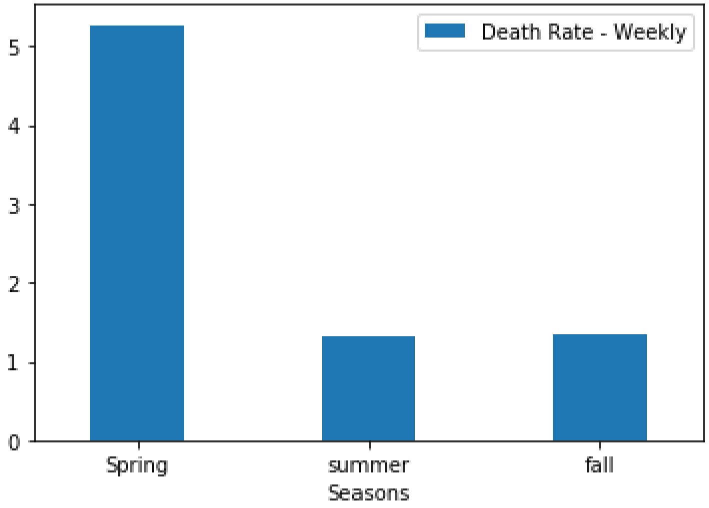
```
In [18]: data_case.plot.bar(x='Seasons', y = ['Test Rate - Weekly'], width = 0.4, rot = 0)
```

```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x140774b95c8>
```



```
In [19]: data_case.plot.bar(x='Seasons', y = ['Death Rate - Weekly'], width = 0.4, rot = 0)
```

```
Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x1407751cc08>
```


Death Rate - Weekly
Spring
summer
fall
Seasons