PROJECT DOCUMENTATION:

OBJECTIVE:

Currently the city of Buffalo receives reports from the Crime Lab in Albany in the form of hotspot maps and regular daily crime forecasting reports. The current system is not automated and requires human intervention to produce report.

This model has the objective of predicting crime numbers, then predict time and location autonomously with the goal of helping police departments introduce micro improvement that could have an impact in better allocating resources and improve response time.

CONTEXT:

Buffalo is the second largest city in New York state with population estimate of 255,284, with median household income of $37,354, and per capita income of $24,40. The city of Buffalo's civilian labor force total is 59.8% of the total population and (Bureau, 2021).

According to city-data.com Crime in Buffalo is 450 per 100,000 , 1.7 times greater than the US average and higher than 93.9% of U.S. cities (City_data, 2021). Compared to neighboring towns and cities, Buffalo has at least double the crime rate of the closest city/town.

Labor data indicates that early in the 1990's Buffalo started witnessing higher than usual unemployment rate as well as crime rate, the data also showed a consistent shrinkage of the labor pool , indicating a correlation between crime and employment. (Ajimotokin, 2015) establishes a clear correlation between crime and unemployment as well as the number of police officer and crime.

THE PROBLEM:

How to forecast and predict crime in buffalo?

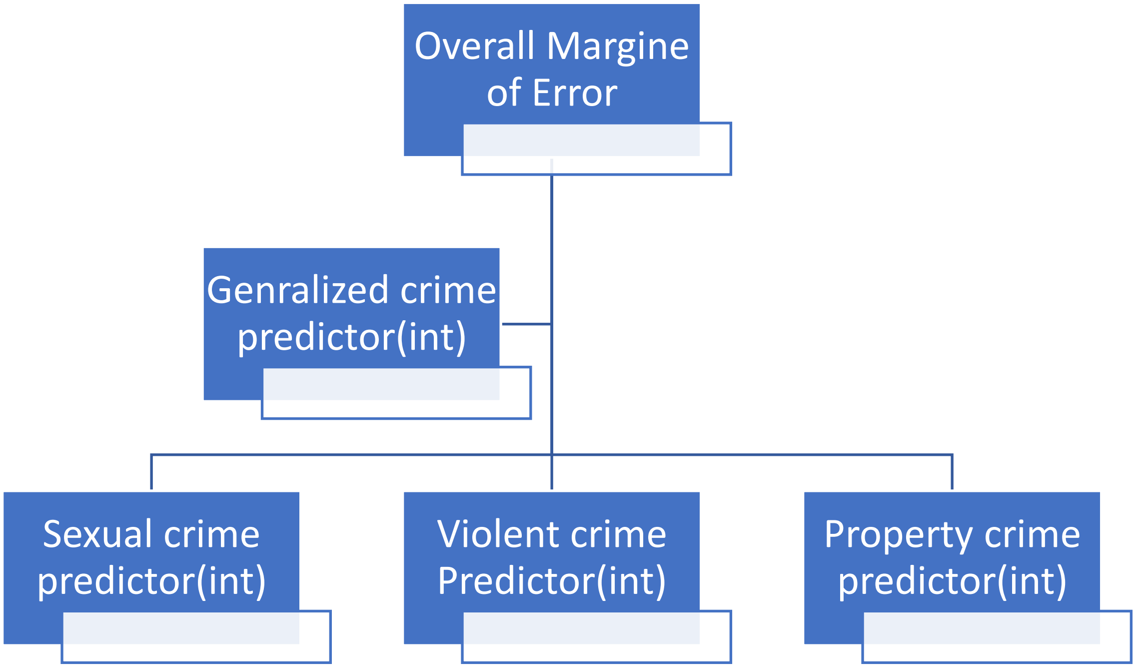
The hurdle around forecasting and predicting crime is the large number of variables that contribute to individuals committing crime. Yet, for the purpose of making a functioning and adequate predictor the use of variables should be broken and too many steps. I concluded that the first step is to build a generalized predictor based on employment data from New York State, this predictor will act as the guiding predictor other micro predictors.

DESIGN:

The relationship between all predictors is going to work as cross-referencing system, where the generalized predictor will predict crime total and micro predictors will predict subsets of crimes such as sexual crimes, violent crimes, and property crimes. After fine tuning all the parameters for all four predictors we can proceed to work on establishing a predictor that could use the input from all four predictors to predict locations and times of where and when a crime might happen.

it is true that the three micro predictors that are concerned in generating numbers will output different numbers that might not align with the generalized predictor ,this issue could be perceived as a weakness or as strength ,in my opinion it is going to be strength Because we will be using their wisdom of crowds to establish better predictions as well as better margin of error to fine tune the overarching problem.

The following is a schema of the system:



CONSTRAINTS:

There are many constraints facing this model the first is government reporting ,police department's tend to report every incident in a sheet where it needs to be aggregated in order for better clarity and understanding of the data as well as generating numerical predictions. The reporting Takes the form of incident reporting meaning every incident it is reported individually and not aggregated to a daily total or monthly total.

nt_id	case_number	incident_datetime	incident_type_primary	incident_description	clearance_type	address_1	address_2	city	state	...	location	hour_of
799.0	20-0050070	01/05/2020 01:30:00 AM	LARCENY/THEFT	Buffalo Police are investigating this report o...	NaN	200 Block ONEIDA AV	NaN	Buffalo	NY	...	POINT (-78.843 42.876)	
739.0	20-0050096	01/05/2020 02:07:00 AM	MURDER	Buffalo Police are investigating this report o...	NaN	PADEREWSKI DR & SHUMWAY ST	NaN	Buffalo	NY	...	POINT (-78.845 42.89)	
680.0	20-0050120	01/04/2020 11:00:48 PM	ASSAULT	Buffalo Police are investigating this report o...	NaN	400 Block GRIDER ST	NaN	Buffalo	NY	...	POINT (-78.829 42.918)	
541.0	20-0050158	01/05/2020 03:57:41 AM	ROBBERY	Buffalo Police are investigating this report o...	NaN	400 Block PEARL ST	NaN	Buffalo	NY	...	POINT (-78.873 42.891)	
NaN	20-0050162	01/05/2020 04:00:00 AM	ASSAULT	Buffalo Police are investigating this report o...	NaN	0 Block POPLAR AV	NaN	Buffalo	NY	...	POINT (-78.805 42.907)	

26 columns

(Data.gov, 2021)

Meanwhile employment data as reported by New York State all monthly basis , in order for the data to be used as exogenous variables to support the predictors crime data must be aggregated to monthly totals.

	Area	Year	Month	Labor Force	Employed	Unemployed	Unemployment Rate
0	Albany City	2021	1	46,800	43,100	3,700	7.9
1	Albany City	2020	12	47,200	43,500	3,700	7.8
2	Albany City	2020	11	47,400	43,800	3,600	7.7
3	Albany City	2020	10	47,200	43,500	3,800	8.0
4	Albany City	2020	9	47,200	43,200	3,900	8.4

Data source: (NYS, n.d.)

DATA WRANGLING:

Crime Data:

As mentioned above we needed to transform the data from an individual reporting to monthly reporting.

we first start by dropping unwanted columns ,in this case it is almost every column aside from the incident type and the date, we then Unify incident types into three distinct categories sexual, property, and violent .Once the and certain types unified we add a new column corresponding to incidents with number 1 to help aggregate the incidents to a total number. This could have been easier if the SQL API corresponding with New York State reporting worked, but after many tries I concluded that it was easier to manipulate the data on my own.

	incident_datetime	incident_type_primary	hour_of_day	day_of_week
0	01/05/2020 01:30:00 AM	LARCENY/THEFT	2	SUNDAY
1	01/05/2020 02:07:00 AM	MURDER	2	SUNDAY
2	01/04/2020 11:00:48 PM	ASSAULT	2	SUNDAY
3	01/05/2020 03:57:41 AM	ROBBERY	20	SATURDAY
4	01/05/2020 04:00:00 AM	ASSAULT	4	SUNDAY

we now have the data in the following form: date, incident type, hour, day, count .

	type	hour	day	count
date				
2020-01-05 01:30:00	prperty	2	sunday	1
2020-01-05 02:07:00	violent	2	sunday	1
2020-01-04 23:00:48	violent	2	sunday	1
2020-01-05 03:57:41	prperty	20	saturday	1
2020-01-05 04:00:00	violent	4	sunday	1

We then aggregate every single type on its own forming a data frame for the type indexed with the date column:

	date	violent
860	2005-06-30	2
957	2013-07-31	482
942	2012-04-30	384
888	2007-10-31	571
723	1994-01-31	0
865	2005-11-30	5
752	1996-06-30	0
952	2013-02-28	359
819	2002-01-31	2
945	2012-07-31	571
966	2014-04-30	350
1005	2017-07-31	448
822	2002-04-30	0
879	2007-01-31	407

Then we merge all three dataframes to form a unified data frame with is monthly totals:

	sexual	violent	property
date			
2006-01-31	45	466	972
2006-02-28	18	301	532
2006-03-31	6	114	215
2006-04-30	18	347	688
2006-05-31	43	712	1315
...	...	...	...
2020-09-30	5	382	710
2020-10-31	3	417	739
2020-11-30	11	400	691
2020-12-31	19	356	647
2021-01-31	3	66	76

Employment Data:

The employment data is reported in monthly fashion yet to crop out the Buffalo city data from the data frame few transformations must be made.

	Area	Year	Month	Labor Force	Employed	Unemployed	Unemployment Rate
0	Albany City	2021	1	46,800	43,100	3,700	7.9
1	Albany City	2020	12	47,200	43,500	3,700	7.8
2	Albany City	2020	11	47,400	43,800	3,600	7.7
3	Albany City	2020	10	47,200	43,500	3,800	8.0
4	Albany City	2020	9	47,200	43,200	3,900	8.4

the first transformation is establishing a datetime column By joining year and month columns and adding a day.

The following code was used :

```
df3=df2[['Year','Month']]

df3['day']=1

df3=pd.to_datetime(df3,yearfirst=True,errors='coerce',format='%m-%Y')
```

df2['Date']=df3

The product:

	Area	Year	Month	Labor Force	Employed	Unemployed	Unemployment Rate	Date
0	Albany City	2021	1	46,800	43,100	3,700	7.9	2021-01-01
1	Albany City	2020	12	47,200	43,500	3,700	7.8	2020-12-01
2	Albany City	2020	11	47,400	43,800	3,600	7.7	2020-11-01
3	Albany City	2020	10	47,200	43,500	3,800	8.0	2020-10-01
4	Albany City	2020	9	47,200	43,200	3,900	8.4	2020-09-01

We needed to transform all the column data types to the miracle data types which required to rid of commas from reported numbers then transforming the numbers to integers.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74423 entries, 0 to 74422
Data columns (total 6 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   Area
74423 non-null  object
 1   Labor Force
74423 non-null  int32
 2   Employed
74423 non-null  int32
 3   Unemployed
74423 non-null  int32
 4   Unemployment Rate
74423 non-null  float64
 5   Date
74423 non-null  datetime64[ns]
dtypes: datetime64[ns](1), float64(1), int32(3), object(1)
memory usage: 2.6+ MB
```

Now that the dataframe is ready, we have to crop Buffalo out of it and merge it with the crime data frame.

Code:

Buff=df2[df2['Area']=='Buffalo City']

	Area	Labor Force	Employed	Unemployed	Unemployment Rate	Date
6510	Buffalo City	109400	97400	12000	11.0	2021-01-01
6511	Buffalo City	110600	98300	12300	11.1	2020-12-01
6512	Buffalo City	110000	98800	11100	10.1	2020-11-01
6513	Buffalo City	110200	98800	11400	10.3	2020-10-01
6514	Buffalo City	110100	97900	12300	11.1	2020-09-01

Code :

```
final=pd.merge(df,Buff,how='left',on='date')
```

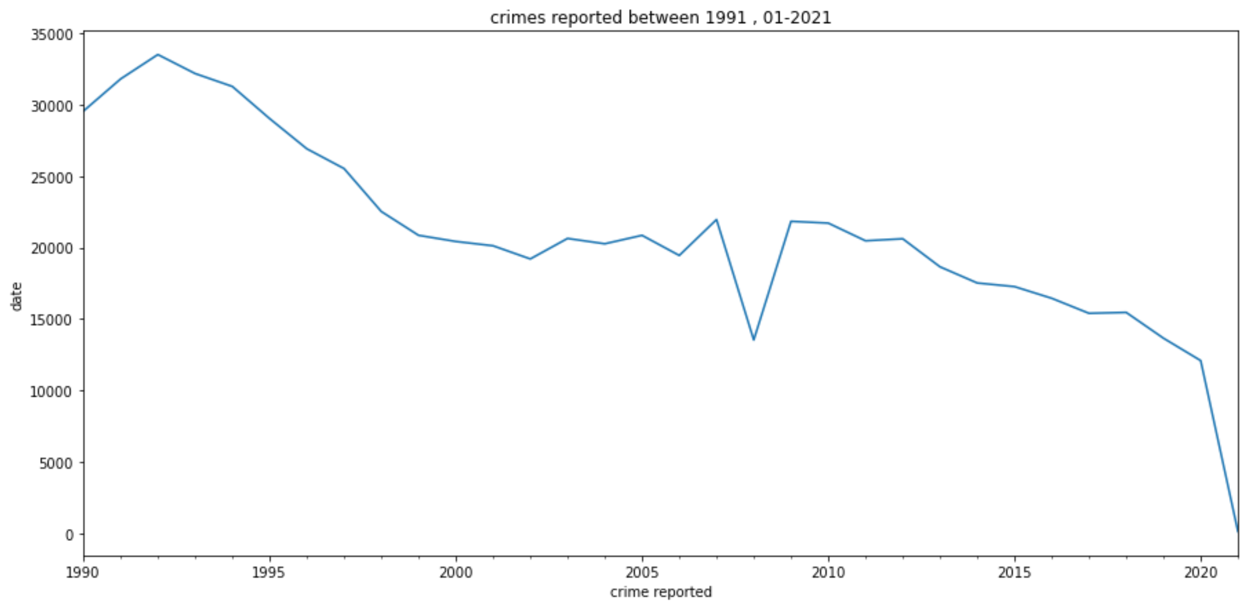
	sexual	violent	property	date	Area	Labor Force	Employed	Unemployed	Unemployment Rate
108	30	373	919	2015-01-01	Buffalo City	110300	101700	8600	7.8
97	17	284	767	2014-02-01	Buffalo City	111400	101400	10000	8.9
16	37	600	1219	2007-05-01	Buffalo City	119800	113100	6700	5.6
196	355	5276	23906	1990-12-01	Buffalo City	147500	133700	13900	9.4
67	31	506	1559	2011-08-01	Buffalo City	117100	104400	12700	10.9
161	21	402	821	2019-06-01	Buffalo City	108800	103000	5800	5.3
57	28	505	1381	2010-10-01	Buffalo City	118200	105900	12400	10.4
122	28	361	787	2016-03-01	Buffalo City	109600	102600	6900	6.3
120	32	355	880	2016-01-01	Buffalo City	109600	102500	7100	6.5
104	19	361	1153	2014-09-01	Buffalo City	111100	102600	8500	7.7
154	26	323	773	2018-11-01	Buffalo City	107400	102400	5000	4.6
172	3	326	614	2020-05-01	Buffalo City	109200	88100	21100	19.3
45	34	552	1415	2009-10-01	Buffalo City	121900	109800	12100	9.9
140	26	433	935	2017-09-01	Buffalo City	110600	103600	7000	6.3
153	28	373	910	2018-10-01	Buffalo City	108600	103600	5000	4.6

The data frame is now ready for the exploratory data analysis.

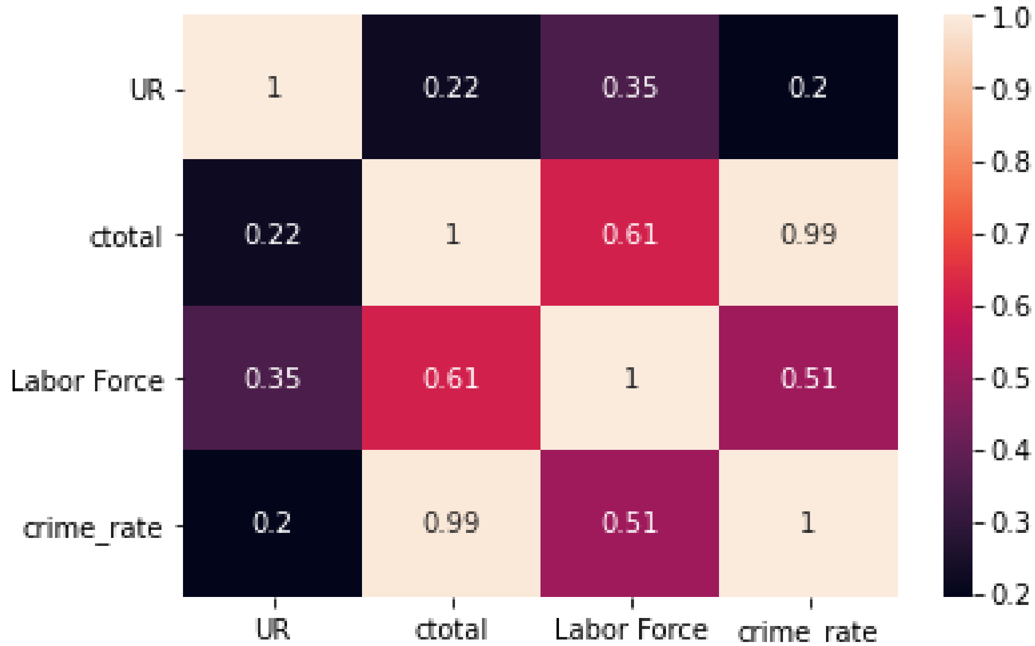
EDA:

Since we are trying to predict the number of crimes that are ought to be committed on a monthly basis ,we need to understand our data from that perspective .

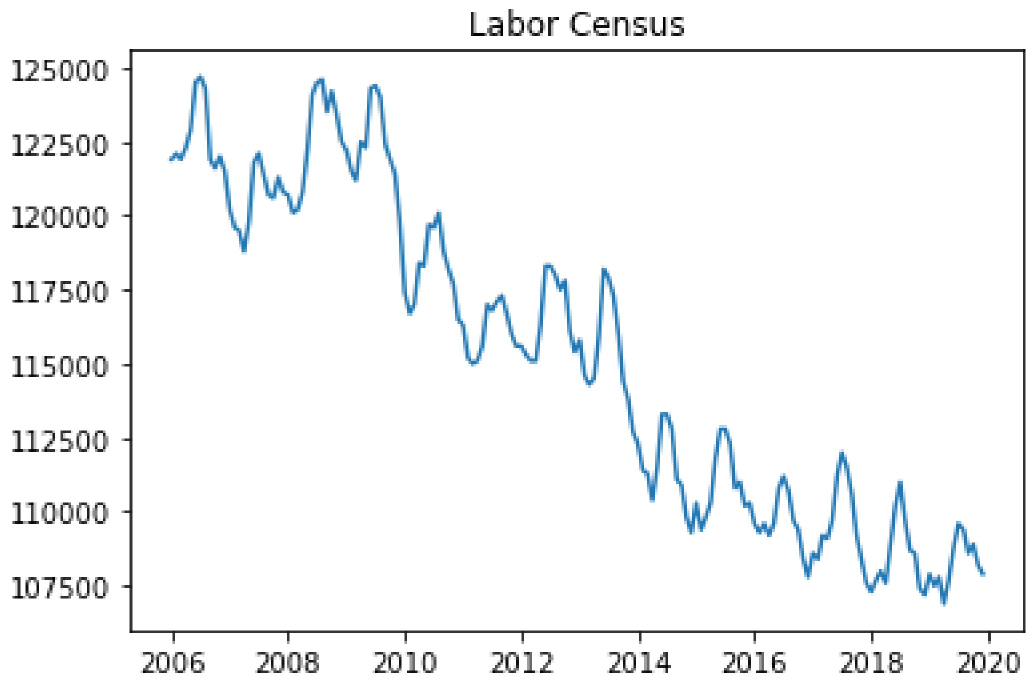
Time series analysis heavily emphasize few points first is the trend, is our data showing any sort of trend ?



Plotting the crime data shows that crime is trending downwards, this could be attributed to many factors, therefore a correlation map it is warranted.



The correlation map shows that crime total and labor force have the highest correlation and unemployment rate has somewhat of decent correlation.



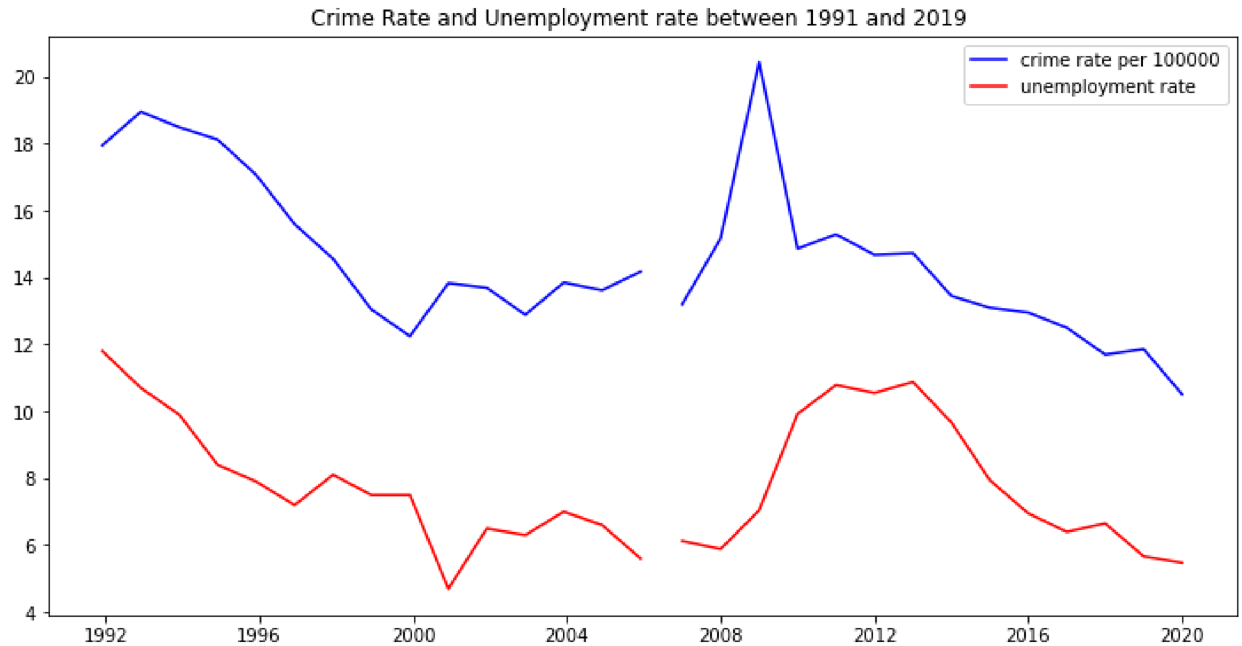
From the labor census provided by the New York State department Of Labor force we could conclude that the trend down in crimes committed is heavily influenced by their reduction of labor force and overall population using the logic less people that commit crimes results in less crimes committed.

Now that the trend is somewhat explained we need to explore the relationship between the unemployment rate and crime ,but first for us to be able to plot the unemployment rate with crime we need to transform the crime total to crime rate.

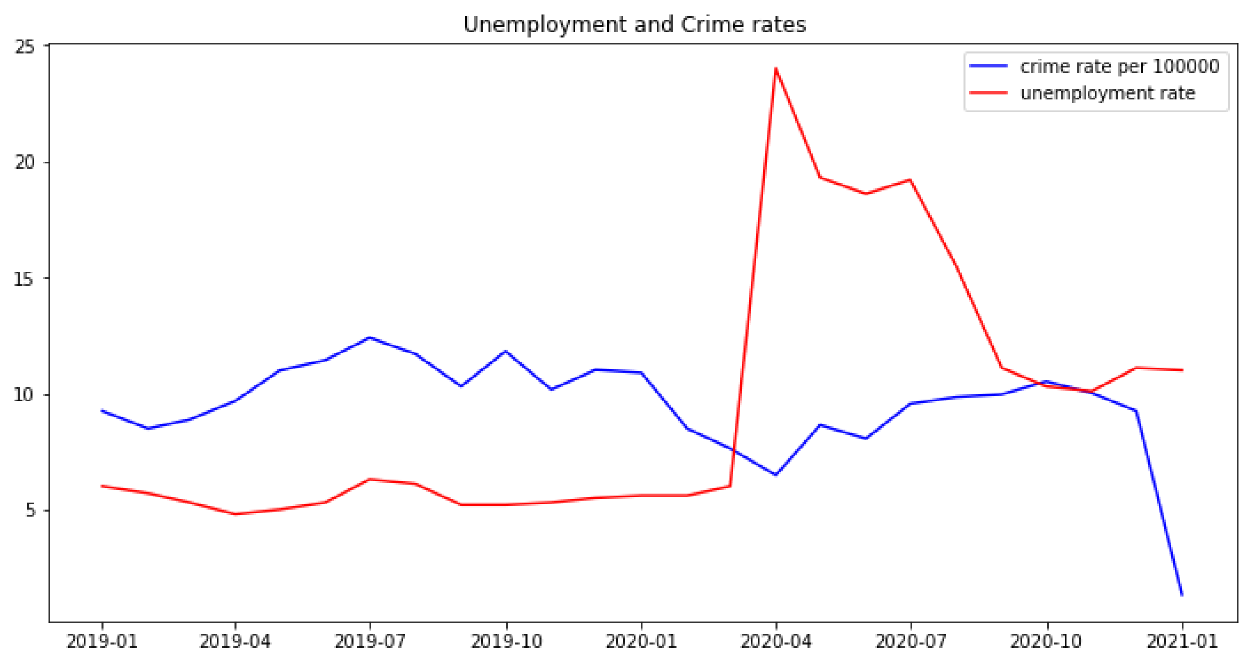
the crime rate his calculated as percentage per 100,000 , therefore we need population figures, a problem arise when trying to find monthly reported population figures since census are carried out once every 10 years , so in order to make the crime rate per 100,000 we need to make the assumption that labor force as a percentage is 50% of the population, it is highly unlikely that the labor force is 50% of the population since the labor force is calculated as the number of individuals between 16 and 65 ,yet for the purpose of data exploration we can make the exception of making an assumption about the labor force being half of the population.

Using the university of Arkansas's method and assuming that the population is likely double the labor force We derive the following equation :

Crime rate = ((crime total/ labor force*2)*100,000)/100

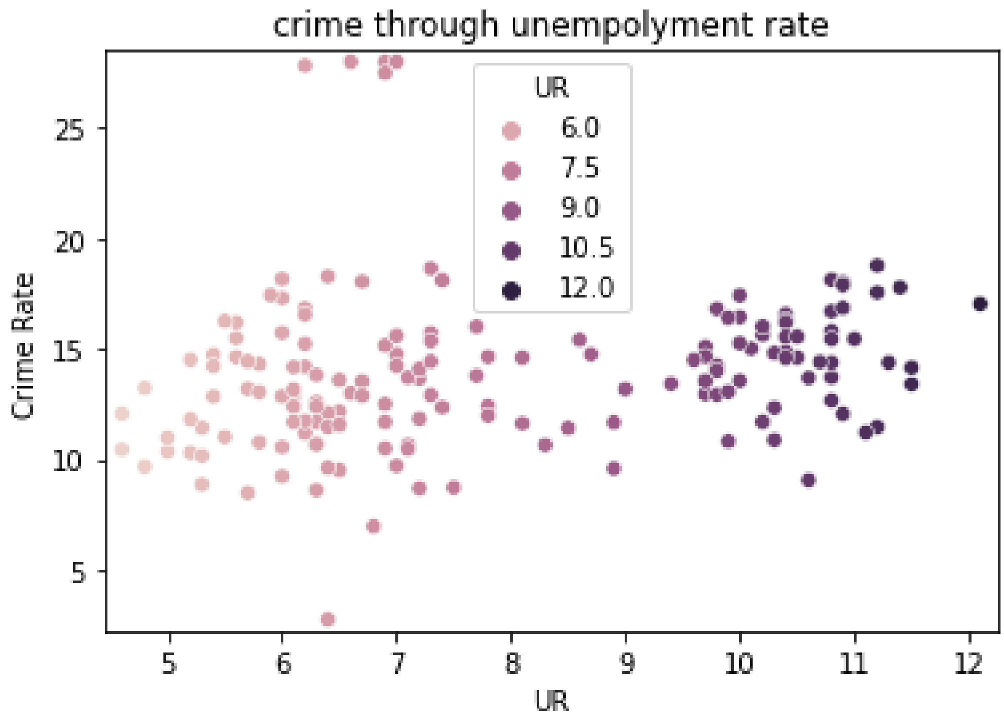


From the following plot it's fairly assumes that the correlation between the crime and unemployment is significant since they both follow the same trajectory for the past 30 years. Yet, while exploring the data an anomaly was detected which was in 2020 the unemployment rate reached up to 24% meanwhile the total crimes committed dropped significantly.



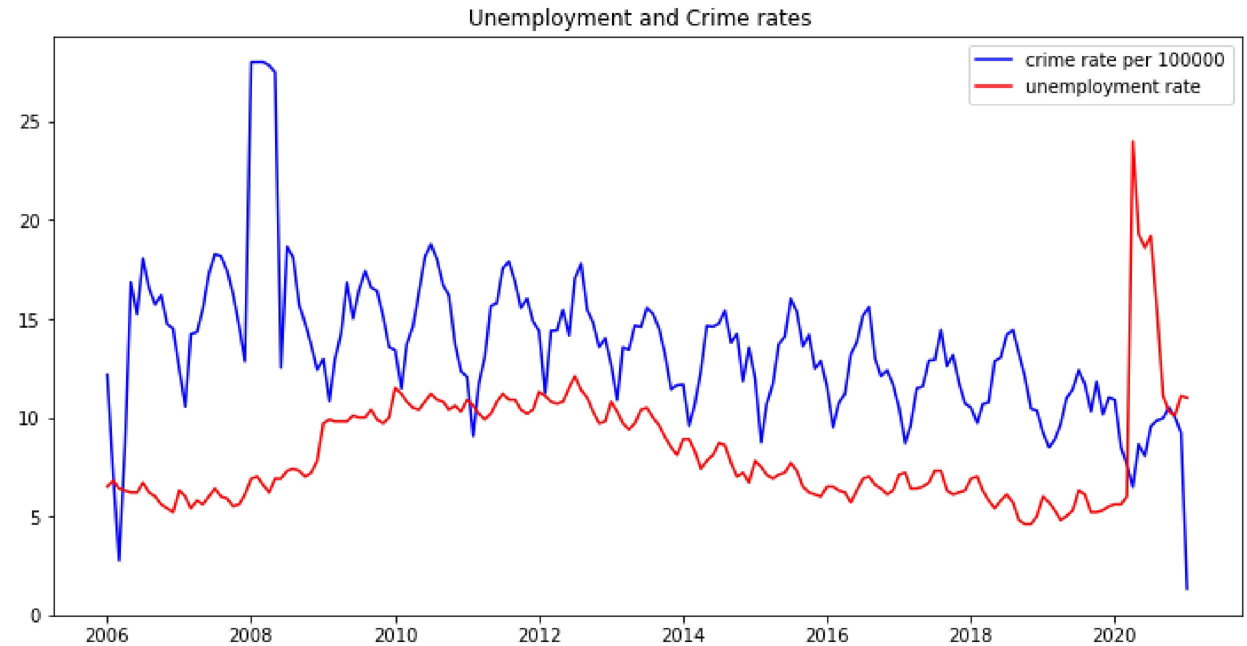
This could be justified with the advent of the coronavirus pandemic and the restrictions imposed on businesses Such as lockdown and limited capacity, requires us to generate new exogeneous variables to justify the divergence.

To explore the crime rate's correlation with the unemployment rate we plotted a scatter plot coloring the crime rate with unemployment rates.



This scatter plot shows minimal correlation between crime rate and unemployment rate pushing us to rid of the unemployment rate as the main exogeneous variable to help predict crime.

Aside from Trend and exogeneous variables exploration time series analysis requires us to see if the variable we would like to predict has any seasonality, a simple plot of the crime rate shows that every year the crime spikes and dips indicating seasonality in the data.



Pre-Processing:

In the preprocessing stage we added context variables to explain a huge drop in crimes committed, the context variables are extraordinary government interventions used to stabilize the economy and provide citizens with income due to employment loss resulting of lockdowns and business restrictions.

The United States government in March 2020 passed a stimulus package that includes unemployment supplement paid to workers that lost their jobs due to the coronavirus pandemic and issued stimulus checks for the entire population.

we translated the government intervention to machine comprehendible variables that followed a machine logic of the number one indicating the existence of the variable and the number 0 indicating the absence of the variable. Note we could not include all the variables contributing to crime therefore we focused on macroeconomic changes.

Here is a snapshot of the resulting data frame:

	sexual	violent	property	Labor Force	Employed	Unemployed	UR	ctotal	lockdown	e_stimulus	crime_rate
date											
2008-01-01	27	787	2592	120700	112300	8400	6.9	262	0	0	1.085336
2008-02-01	27	787	2592	120100	111700	8400	7.0	17	0	0	0.070774
2008-03-01	27	787	2592	120200	112300	7900	6.6	37	0	1	0.153910
2008-04-01	27	787	2592	120800	113400	7500	6.2	30	0	0	0.124172
2008-05-01	27	787	2592	122200	113800	8400	6.9	75	0	0	0.306874
2008-06-01	32	429	1092	124100	115600	8600	6.9	1553	0	0	6.257051
2008-07-01	41	679	1602	124500	115400	9100	7.3	2322	0	0	9.325301
2008-08-01	49	638	1570	124600	115400	9200	7.4	2257	0	0	9.056982
2008-09-01	33	548	1360	123500	114500	9000	7.3	1941	0	0	7.858300
2008-10-01	30	483	1321	124200	115600	8600	7.0	1834	0	0	7.383253
2008-11-01	39	487	1160	123400	114500	8900	7.2	1686	0	0	6.831442
2008-12-01	31	437	1053	122500	113000	9600	7.8	1521	0	0	6.208163

MODELING:

Since the data is seasonal we have many options to forecast crime one of them as Arima model , Arima stands for auto regressive moving average model, this model will forecast the moving average of future crimes committed , Arima models does not forecast seasonality or include any exogenous variables , therefore if we need and accurate forecasting we need to add seasonality to the forecast that could be achieved by using Sarimax model which is seasonal other aggressive moving average (Rob J Hyndman, 2016).

As mentioned above Sarimax is based On the Arima model and requires seven parameters to be found for the model to operate correctly , the parameters are Arima order which requires three parameters P, D, Q. seasonal order of P, D, Q, S . P is the auto regression order, D is the differencing order, Q is the moving average order , S is the season order (Rob J Hyndman, 2016).

Time series analysis requires the data to be stationary, stationarity means that the data should not have any trend to it. There are many ways to achieve stationarity , one way to achieve stationarity is by shifting the data ,Shifting the data was chosen because we are dealing with consequential data meaning The number of employed could take time to affect the crime number at a later date . To test for stationarity, we can use the Adfuller test, Code:

```
def adfuller_test(data):
        result=adfuller(data)
        labels=['ADF test statistic','P-value','#lags used','Number of Observations Used']
                for value,label in zip(result,labels):
        print (label+':'+str(value))
```

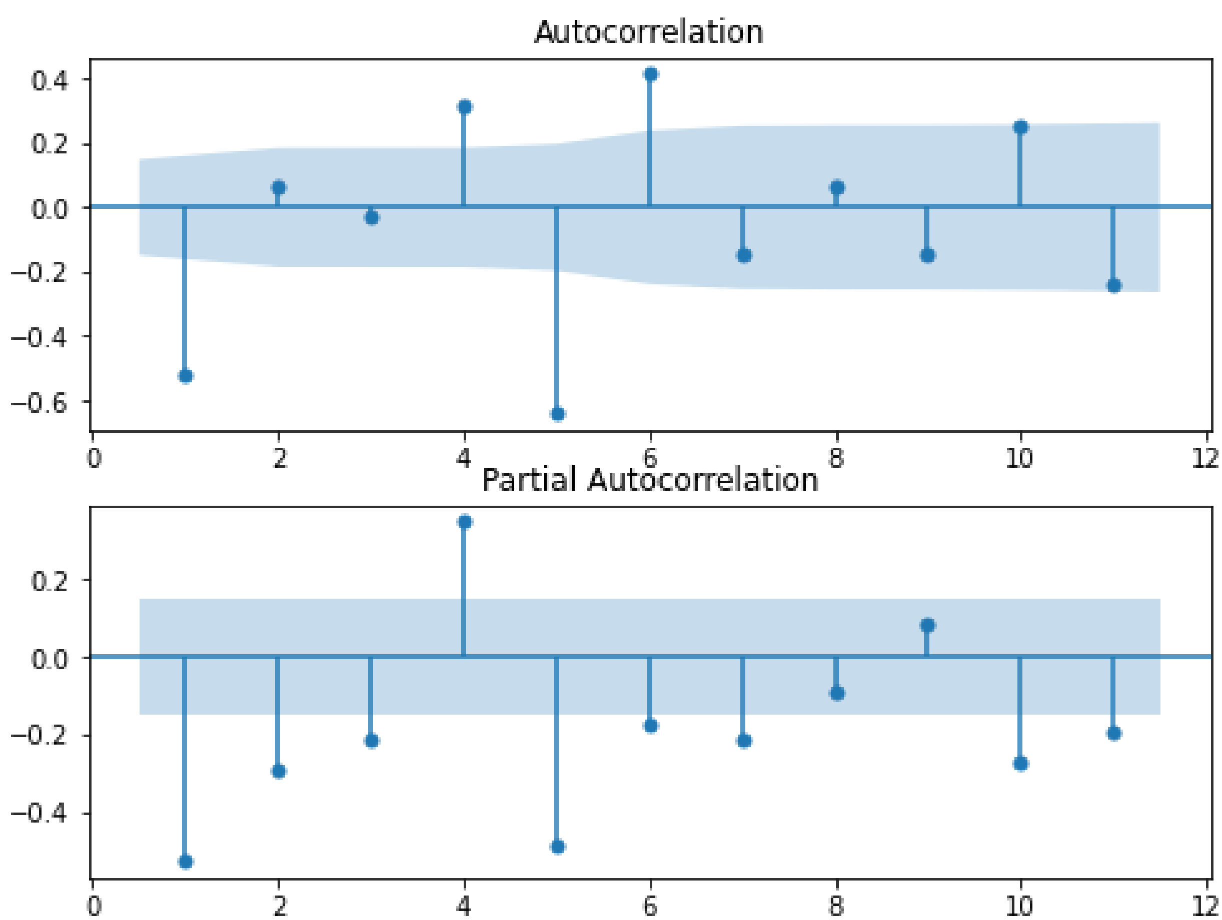
```
    if result [1]<=0.05 :
        print ('strong evidence against null hypothesis (h0), reject Null hypothesis, Data is stationary')
    else:
        print ('weak evidence against alternative hypothesis, time series has a unit root, indicating data is not stationary')
```

Adfuller test outputs variables such as ADF test statistic , p_value , number of lags used , and number of observations , the most significant out of all is the p_value which indicates if data is stationary or not , if the p_value is higher than 0.5 the data is stationary.

Shifting the data works in the following way: subtracting the data to itself while being moved one month forward, example: crime total – (crime total +1month). Note the differencing number in which the shift was made to achieve data Stationary is a part of the model parameters (d).

Once the D parameter is found we need to find P and Q, AIC and BIC tests are mainly the ways to find such parameters, also they could be found manually using Auto Correlation and Partial Auto Correlation.

Auto correlation, Partial Auto correlation Example:



With Auto Correlation we could find out if the model an AR or MA model, and Partial Autocorrelation helps us finding P and Q. Significant values in Auto Correlation are the values not covered by the shade , if the AutoCorrelation plot shows significant lags in the beginning then trails off within the shade it means the model is an AR model , if the AutoCorrelation plot exhibits another significant lags after it dips in the non-significant zone then the model is ARMA .

AutoCorrelation code:

```
plot_acf(diff,lags=11)
```

Partial AutoCorrelation code :

```
plot_pacf(diff,lags=11)
```

For our Model we used an AIC BIC test to find the best parameters with the following code:

```
def aic_bic(p,q):
    order_aic_bic=[]
    for p in range(p):
      for q in range(q):
            model = SARIMAX(X_train, order=(p,0,q))
            results = model.fit()
            order_aic_bic.append((p,q,results.aic, results.bic))
    order_df = pd.DataFrame(order_aic_bic, columns=['p', 'q', 'AIC', 'BIC'])
    print(order_df.sort_values('AIC').head())
    print(order_df.sort_values('BIC').head())
```

the aic_bic function returns a table of AIC and BIC test results:

```
    p  q          AIC          BIC
19  2  5  1874.958986  1898.021401
39  5  4  1875.783644  1904.611663
26  3  5  1875.965290  1901.910507
33  4  5  1876.162164  1904.990183
40  5  5  1877.322247  1909.033068
    p  q          AIC          BIC
5   0  5  1878.599296  1895.896107
19  2  5  1874.958986  1898.021401
8   1  1  1889.966374  1898.614780
12  1  5  1878.939695  1899.119308
18  2  4  1880.483898  1900.663511
```

As a rule of thumb when using AIC and BIC test lower AIC means better predictions and lower BIC means better training or at least that is what I learned from DataCamp course on time series analysis.

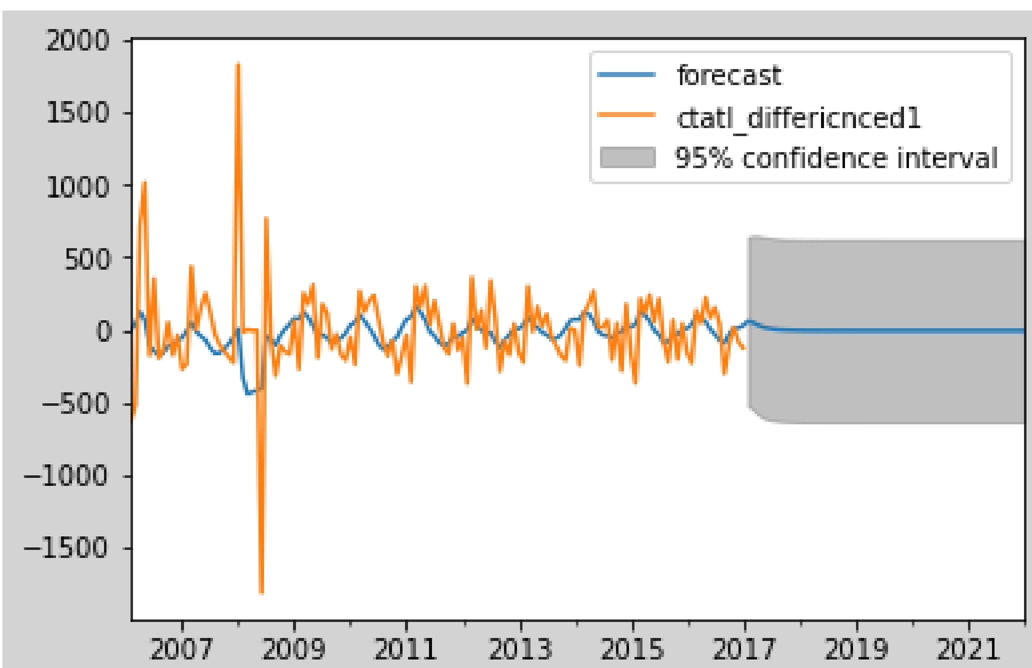
The best Parameters showing are 2 for p and 5 for q.

Now that we have all three parameters p,d,q we can test them on an Arima model using the code

```
# we first make the train test split
X=df.ctotal
y=dfmms.drop("ctatl_differicnced1",axis=1)

y_train=y['2006-08-01':'2017-01-01']
y_test=y['2017-01-01':'2020-01-01']
X_train=X['2006-08-01':'2017-01-01']
X_test=X['2017-01-01':'2019-12-01']
model=ARMA(X_train.dropna(),order=(2,1,5))
res=model.fit()
res.plot_predict(start=0,end='2022')
```

which resulted in the following plot:



The resulting plot does not seem like the best prediction plot we need something more tangible and useable.

So far we looked for three different variables P, D, Q , now it's time to find our seasonal order for the data , there are few ways to find seasonal order, the first is using seasonal decomposition Then applying partial autocorrelation to find the significant lags.

Code:

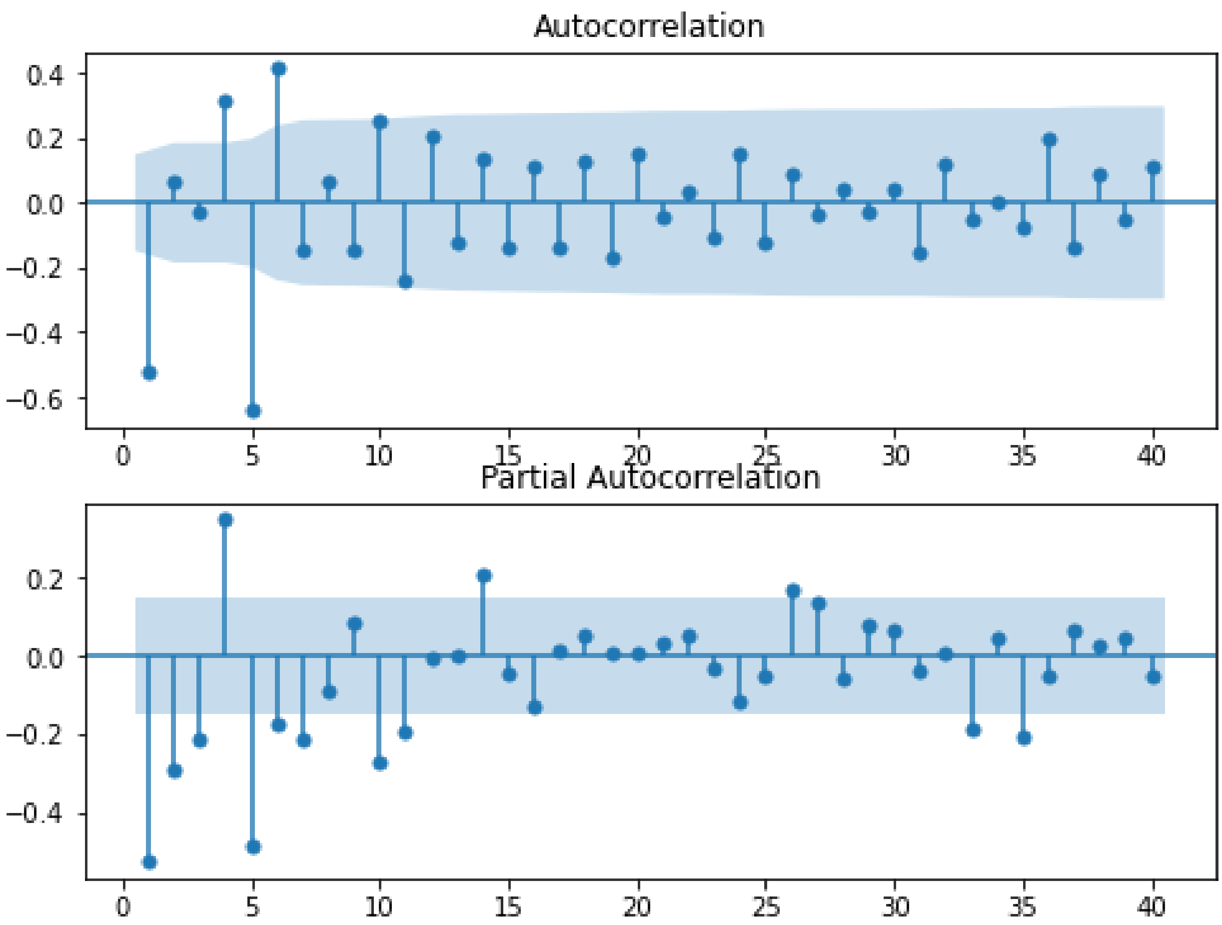
```
fig, (ax1, ax2) = plt.subplots(2,1,figsize=(8,6))

plot_acf(diff,lags=40,zero=False, ax=ax1)

plot_pacf(diff,lags=40,zero=False, ax=ax2)

plt.show()
```

the return should be close to this graph:



The auto correlation is showing significant lags at 3 ,5, 6, while PCA is showing 4,5,10,14.

SARIMAX:

To Implement SARIMAX on our variables we first must split the data to test and train data, then carry out the testing. You have seen an example of crating the split in previous code, the following the code I used to train, visualize, and test:

```
def Sarimax( endog,train_exog,test_exog,order,seasonal_order)
    mod=SARIMAX(endog,exog=train_exog,order=order,seasonal_order=seasonal_order,time_varying_regression=True,mle_regression=False)
    res=mod.fit()
    pred=res.predict(start='01-01-2017',end='01-02-2020',exog=test_exog,dynamic=True)
    results=results.join(pred)
    results.rename(columns={"predicted_mean":'Sarimax[seasonal_order]'},inplace=True)
```

for plotting results:

```
def plot(sarimax_order):
    fig = plt.figure(figsize=(12, 6))
    ax1=fig.add_subplot(111)
    ax22=fig.add_subplot(111)
    ax1.plot(results[sarimax_order],color='blue',label='Pred')
    ax1.set_title('preds vs True')
    ax22.plot(results['ctotal'],color='red',label='True ')
    plt.legend()
```

testing:

```
def mape(actual, pred):
    actual, pred = np.array(actual), np.array(pred)
    return np.mean(np.abs((actual - pred) / actual)) * 100
```

after testing a quite handful of Sarimax orders I settled on **SARIMAX(X_train,exog=y_train,order=(2,1,5),seasonal_order=(2,0,5,6),time_varying_regression=False)**

that had MAPE of 8.5%, and the residual plot was all bound between 100 to – 200 on predictions. A result that I'm willing to accept for the time being until I further develop other parts of the model.

Bibliography

Ajimotokin, S. H. (2015). The effects of unemployment on crime rates in the US.

Bureau, U. S. (2021, 08 02). *Quick Facts, Buffalo city, New York* . Retrieved from census.gov: https://www.census.gov/quickfacts/buffalocitynewyork

City_data. (2021). *Crime rate in Buffalo*. Retrieved 8 2, 2021, from city-data.com: http://www.city-data.com/crime/crime-Buffalo-New-York.html

Data.gov. (2021). *Data Lens*. Retrieved from Open Data Buffalo : https://data.buffalony.gov/Public-Safety/Crime-Incidents-Data-Lens-/vhp3-62vz

NYS. (n.d.). *NY state employment data (county based):*. Retrieved from Data.NY.Gov: https://data.ny.gov/Economic-Development/Local-Area-Unemployment-Statistics-Beginning-1976/5hyu-bdh8