Final

Kurt Meyer & Antonio Dehesa

## Introduction

Our research focuses on the relationship between the inflation rate and crime rates, as well as the type of crime and its rate relative to the inflation. With this information, we would like to know if it is possible to predict an increase in crime and type of crime based on inflation rates.

To measure the inflation rates, we used the Consumer Price Index (CPI) as our main metric, which is the measure of the average change in the prices paid by urban consumers for a market basket of consumer goods and services. The way CPI is calculated is: Value of Basket in the current year over the value of the basket in the prior year, times 100.

Data Sources and Definitions Explained

We based our research on data obtained from the St. Louis Federal Reserve [1](https://fred.stlouisfed.org/) for the inflation rates data. Specifically, a dataset which includes data from the 1960s to the current era. As for crime rates, as well as types of crime, we obtained this information from the FBI Uniform Crime Reporting [2](https://cde.ucr.cjis.gov/) dataset, which was supplemented with data obtained through Statista [3](https://www.statista.com), for which we have access thanks to our student access through the University of Denver. Unfortunately, the FBI datasets are split by year, which would take too much time to put everything together. Fortunately, we could use disastercenter [4](https://www.disastercenter.com/), which presents the same information already gathered.

Main Features of the Data Sets

We will first have to load the datasets and perform data cleaning.

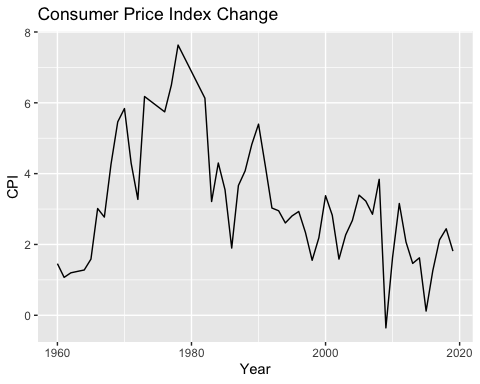
Rape statistics cannot be trusted, as prior to 2016, the FBI included only female-reported rapes, and from 2016 and forward, they included both male and female reported rapes. Therefore, we will not use it for our analysis, as it shows a massive jump in reports.

# We load our datasets, clean them and join them into a single dataframe, for easier handling  
Tot\_Pop <- read.csv("./TotalPopulation1960to2020.csv")  
Tot\_Pop$DATE <- as.numeric(format(as.Date(Tot\_Pop$DATE), "%Y"))  
CPI <- read.csv("./FPCPITOTLZGUSA.csv")  
CPI$DATE <- as.numeric(format(as.Date(CPI$DATE), "%Y"))  
  
crimes <- read.csv("./crimes.csv")  
names(crimes)[11] = "Larceny theft"  
names(crimes)[12] = "Vehicle theft"  
crimes <- filter(crimes,Year <= 2019)  
Tot\_Pop <- filter(Tot\_Pop,DATE <= 2019)  
CPI <- filter(CPI,DATE <= 2019)  
  
dat <- cbind(CPI, Tot\_Pop$POPTOTUSA647NWDB, crimes[,3:ncol(crimes)])  
  
# Now we can remove the original datasets to avoid memory issues  
rm(CPI)  
rm(Tot\_Pop)  
rm(crimes)  
# We rename some columns for easier handling  
names(dat)[1:3] = c("Date", "CPI", "Tot\_Pop")  
# Now, we get additional derived metrics which might be helpful   
  
  
inf\_rate <- (dat$CPI)   
  
  
crime\_percentage <- dat$Total/dat$Tot\_Pop \* 100  
dat <- cbind(dat, inf\_rate, crime\_percentage)  
  
# Finally, we should remove outliers. For example, inflation for most years stays within 100%, except for some years, in which inflation reached over 500%.   
mean\_inf\_rate <- mean(dat$inf\_rate)  
sd\_inf\_rate <- sd(dat$inf\_rate)  
# We will use only data that is within 3 standard deviations for the inflation rate  
dat <- filter(dat, between(inf\_rate, -3\*sd\_inf\_rate, 3\*sd\_inf\_rate))  
  
head(dat, n = 5)

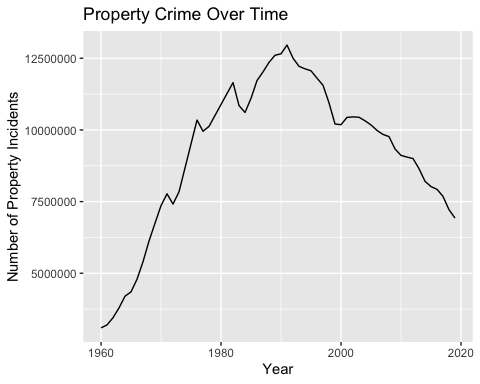
## Date CPI Tot\_Pop Total Violent Property Murder Rape Robbery assault  
## 1 1960 1.457976 180671000 3384200 288460 3095700 9110 17190 107840 154320  
## 2 1961 1.070724 183691000 3488000 289390 3198600 8740 17220 106670 156760  
## 3 1962 1.198773 186538000 3752200 301510 3450700 8530 17550 110860 164570  
## 4 1963 1.239669 189242000 4109500 316970 3792500 8640 17650 116470 174210  
## 5 1964 1.278912 191889000 4564600 364220 4200400 9360 21420 130390 203050  
## Burglary Larceny theft Vehicle theft inf\_rate crime\_percentage  
## 1 912100 1855400 328200 1.457976 1.873129  
## 2 949600 1913000 336000 1.070724 1.898841  
## 3 994300 2089600 366800 1.198773 2.011494  
## 4 1086400 2297800 408300 1.239669 2.171558  
## 5 1213200 2514400 472800 1.278912 2.378771

Below are several graphical depictions of the data set. They do seem to all follow roughly the same shape, an indication of normality.

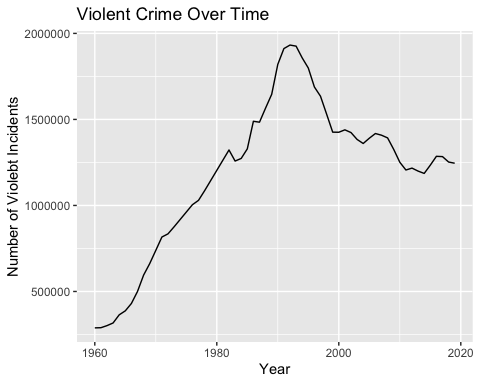
CPI\_Plot <- ggplot(dat, aes(x = Date, y = CPI)) +  
 geom\_line() +  
 labs(title = "Consumer Price Index Change", x = "Year", y = "CPI")  
  
  
Property\_Plot <- ggplot(dat, aes(x = Date, y = Property)) +  
 geom\_line() +  
 labs(title = "Property Crime Over Time", x = "Year", y = "Number of Property Incidents")  
  
Murder\_Plot <- ggplot(dat, aes(x = Date, y = Murder)) +  
 geom\_line() +  
 labs(title = "Murder Over Time", x = "Year", y = "Number of Murder Incidents")  
  
Violent\_Plot <- ggplot(dat, aes(x = Date, y = Violent)) +  
 geom\_line() +  
 labs(title = "Violent Crime Over Time", x = "Year", y = "Number of Violebt Incidents")  
  
Burglary\_Plot <- ggplot(dat, aes(x = Date, y = Burglary)) +  
 geom\_line() +  
 labs(title = "Burglary Over Time", x = "Year", y = "Total Number of Burglary Incidents")  
  
Total\_Plot <- ggplot(dat, aes(x = Date, y = Total)) +  
 geom\_line() +  
 labs(title = "Total Crime Over Time", x = "Year", y = "Total Number of Incidents")  
  
  
  
CPI\_Plot



Property\_Plot



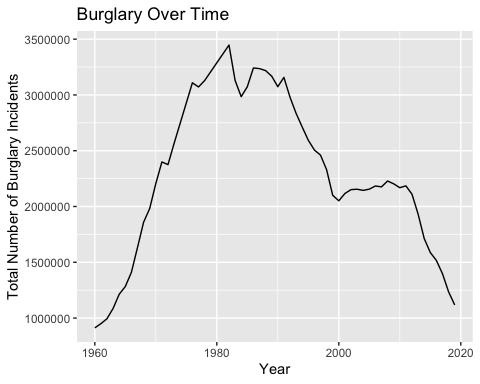
Violent\_Plot



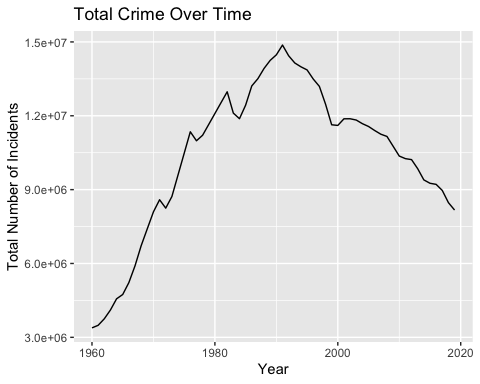
Murder\_Plot



Burglary\_Plot



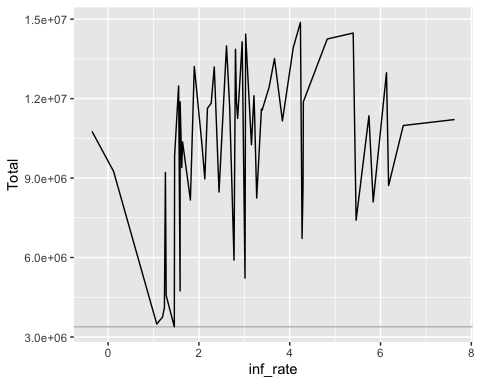
Total\_Plot



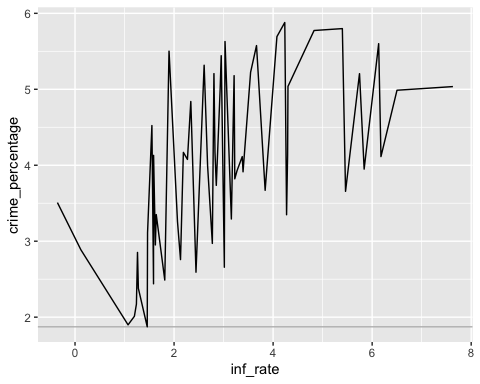
Data Visualization

In this section we will plot our inflation data on the x axis against different crime data on the y axis.

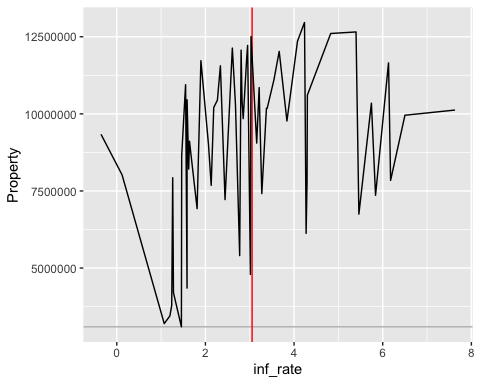
# With the previously obtained information, we can visualize tot\_crime vs inflation  
  
ggplot(data= dat, mapping = aes(x = inf\_rate, y = Total))+  
 geom\_hline(yintercept = min(dat$Total),color="gray")+  
 geom\_line()

 We visualize the rate of crime per year by population vs inflation rate to see if the difference in population may be related

ggplot(data= dat, mapping = aes(x = inf\_rate, y = crime\_percentage))+  
 geom\_hline(yintercept = min(dat$crime\_percentage),color="gray")+  
 geom\_line()

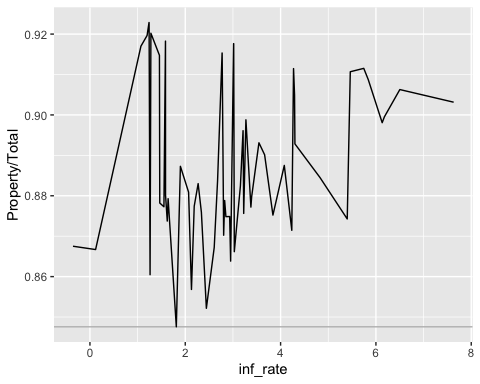
 We now should focus on crimes related to money and resources. In this dataset, these crimes would be: Property, Robbery, Burglary, Larceny theft and Vehicle theft, which are summarized by the Property type of crime.

ggplot(data= dat, mapping = aes(x = inf\_rate, y = Property))+  
 geom\_hline(yintercept = min(dat$Property),color="gray")+  
 geom\_vline(xintercept = mean(dat$inf\_rate),color="red")+  
 geom\_line()



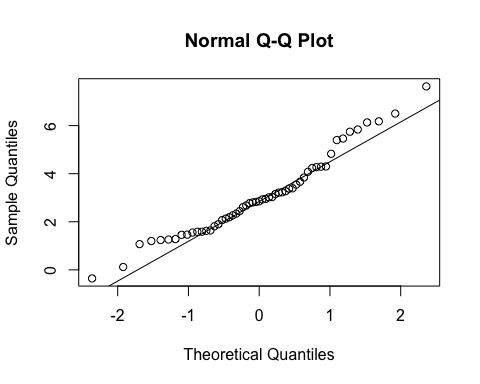
We can also try to see any relation with the percentage of financial-type crimes relative to the total amount of crimes per year vs inflation rates:

ggplot(data= dat, mapping = aes(x = inf\_rate, y = Property/Total))+  
 geom\_hline(yintercept = min(dat$Property / dat$Total),color="gray")+  
 geom\_line()

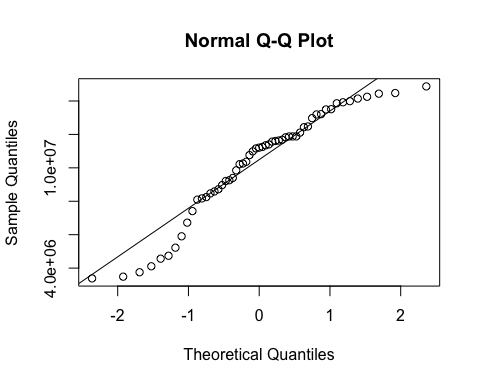


As we can see, there may be a relationship between inflation and crime; however, it is uncertain how much of a relationship there is. This is why it would be useful to perform a series of correlation tests. Prior to performing these tests, let’s determine whether our data is normal using qqplots. If so, we will use the Pearson Correlation Test.

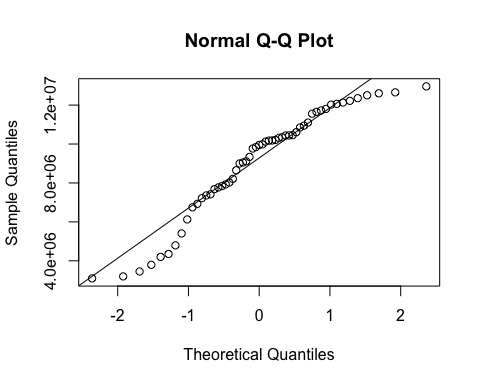
qqnorm(dat$CPI)  
qqline(dat$CPI)



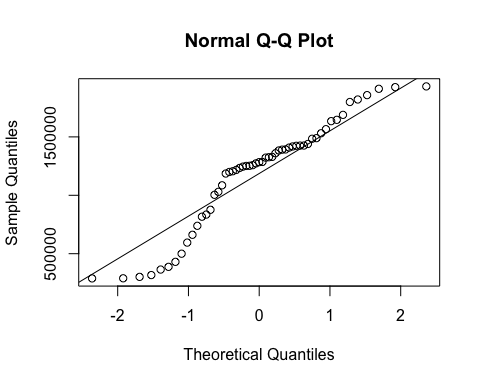
qqnorm(dat$Total)  
qqline(dat$Total)



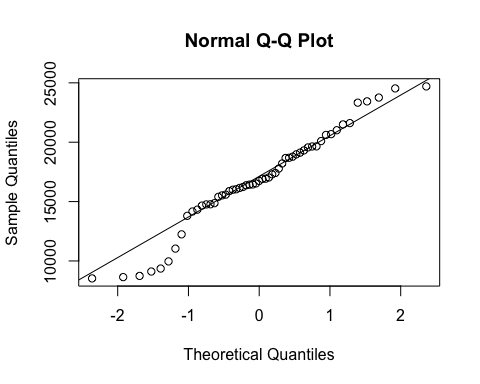
qqnorm(dat$Property)  
qqline(dat$Property)



qqnorm(dat$Violent)  
qqline(dat$Violent)



qqnorm(dat$Murder)  
qqline(dat$Murder)



Now we will perform a series of correlation tests, using the Pearson method. First we will use the raw data and then utilize data converted to crimes per 100,000.

#Pearson Correlations with Raw Data  
  
  
cor.test(dat$inf\_rate, dat$Total)

##   
## Pearson's product-moment correlation  
##   
## data: dat$inf\_rate and dat$Total  
## t = 2.3012, df = 53, p-value = 0.02534  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.0392393 0.5247379  
## sample estimates:  
## cor   
## 0.3013989

cor.test(dat$inf\_rate, dat$Property)

##   
## Pearson's product-moment correlation  
##   
## data: dat$inf\_rate and dat$Property  
## t = 2.5844, df = 53, p-value = 0.01254  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.07599138 0.55094306  
## sample estimates:  
## cor   
## 0.3345441

cor.test(dat$inf\_rate, dat$Murder)

##   
## Pearson's product-moment correlation  
##   
## data: dat$inf\_rate and dat$Murder  
## t = 3.9865, df = 53, p-value = 0.0002064  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.2463938 0.6613257  
## sample estimates:  
## cor   
## 0.4802953

cor.test(dat$inf\_rate, dat$Violent)

##   
## Pearson's product-moment correlation  
##   
## data: dat$inf\_rate and dat$Violent  
## t = 0.69648, df = 53, p-value = 0.4892  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.1744718 0.3516462  
## sample estimates:  
## cor   
## 0.09523339

cor.test(dat$inf\_rate, dat$Burglary)

##   
## Pearson's product-moment correlation  
##   
## data: dat$inf\_rate and dat$Burglary  
## t = 5.7786, df = 53, p-value = 4.066e-07  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.4268047 0.7614174  
## sample estimates:  
## cor   
## 0.6217061

The correlation tests in most cases show p-values less than .05 (the exception being violent crime) with varying correlation coefficients often showing a weak or moderate relationship. The strongest correlations seem to be between inflation and property crimes, specifically burglary. Let’s convert out data to crimes per 100,000. And run the correlation tests again on property crimes.

convert\_to\_per\_capita <- function(data) {  
 # Find the column index for the "Tot\_Pop" column  
 pop\_col <- grep("Tot\_Pop", colnames(data))  
 # If no "Tot\_Pop" column is found, return the original data  
 if (length(pop\_col) == 0) {  
 return(data)  
 }  
 # Divide all columns after "Tot\_Pop" by population and multiply by 100,000  
 for (i in (pop\_col+1):ncol(data)) {  
 data[, i] <- (data[, i] / data[, pop\_col]) \* 100000  
 }  
   
 return(data)  
}  
  
dat\_converted <- convert\_to\_per\_capita(dat)  
#dat\_converted

cor.test(dat$inf\_rate, dat\_converted$Total)

##   
## Pearson's product-moment correlation  
##   
## data: dat$inf\_rate and dat\_converted$Total  
## t = 4.9922, df = 53, p-value = 6.825e-06  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.3532408 0.7224438  
## sample estimates:  
## cor   
## 0.5655392

cor.test(dat$inf\_rate, dat\_converted$Property)

##   
## Pearson's product-moment correlation  
##   
## data: dat$inf\_rate and dat\_converted$Property  
## t = 5.4013, df = 53, p-value = 1.593e-06  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.3926037 0.7435940  
## sample estimates:  
## cor   
## 0.5958402

cor.test(dat$inf\_rate, dat\_converted$Murder)

##   
## Pearson's product-moment correlation  
##   
## data: dat$inf\_rate and dat\_converted$Murder  
## t = 7.3379, df = 53, p-value = 1.29e-09  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.5477594 0.8206389  
## sample estimates:  
## cor   
## 0.7098953

cor.test(dat$inf\_rate, dat\_converted$Violent)

##   
## Pearson's product-moment correlation  
##   
## data: dat$inf\_rate and dat\_converted$Violent  
## t = 2.4641, df = 53, p-value = 0.01701  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.06044965 0.53997453  
## sample estimates:  
## cor   
## 0.3206052

cor.test(dat$inf\_rate, dat\_converted$Burglary)

##   
## Pearson's product-moment correlation  
##   
## data: dat$inf\_rate and dat\_converted$Burglary  
## t = 8.1602, df = 53, p-value = 6.213e-11  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.5996050 0.8443472  
## sample estimates:  
## cor   
## 0.7462013

After converting the data to account for population changes we see even stronger correlations using the Pearson test. The Pearson test seems to be the most appropriate test given the normality of the data.

Determining Methods for Forecasting

Thus far our results have been promising. After working with this data we have determined that a time series analysis would be useful for predicting future crime rates based on inflation. We came across the Granger test in our research which can be helpful in terms of determining the predictive value of these data and whether it is worth performing a time series analysis. Based on the p-values we generated we believe there is enough predictive values to perform a time series on this data.

# Create a dataframe with the two variables  
data <- data.frame(inf\_rate = c(1.5, 2.0, 2.5, 2.2, 2.8),  
 property\_crime = c(100, 120, 110, 115, 130))  
  
# Specify the variables as a VAR model  
model <- VAR(dat\_converted, p = 2, type = "const")  
grangertest(inf\_rate ~ Property, order = 3, data = dat\_converted)

## Granger causality test  
##   
## Model 1: inf\_rate ~ Lags(inf\_rate, 1:3) + Lags(Property, 1:3)  
## Model 2: inf\_rate ~ Lags(inf\_rate, 1:3)  
## Res.Df Df F Pr(>F)   
## 1 45   
## 2 48 -3 5.3064 0.003226 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

grangertest(inf\_rate ~ Burglary, order = 3, data = dat\_converted)

## Granger causality test  
##   
## Model 1: inf\_rate ~ Lags(inf\_rate, 1:3) + Lags(Burglary, 1:3)  
## Model 2: inf\_rate ~ Lags(inf\_rate, 1:3)  
## Res.Df Df F Pr(>F)   
## 1 45   
## 2 48 -3 9.303 6.7e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

grangertest(inf\_rate ~ Total, order = 3, data = dat)

## Granger causality test  
##   
## Model 1: inf\_rate ~ Lags(inf\_rate, 1:3) + Lags(Total, 1:3)  
## Model 2: inf\_rate ~ Lags(inf\_rate, 1:3)  
## Res.Df Df F Pr(>F)   
## 1 45   
## 2 48 -3 3.841 0.01567 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

### Description of what our model does and how the algorith works

### Major Data analysis and modeling

Now that we have established a strong correlation we can experiment with predictive models. First, we will use Facebook prophet which is a forecasting package developed to improve upon existing prediction tools commonly found in R and Python. Due to Prophet’s requirements, this process requires some data preparation.

#Prophet requires the data to be formatted in a certain way.   
#Create formatted dataframes for making predictions  
df\_model1 <- dat\_converted[, c('Date', 'CPI', 'Property')]  
colnames(df\_model1) <- c('ds', 'CPI', 'y')  
df\_model1$ds <- as.Date(paste0(df\_model1$ds, "-01-01"), format = "%Y-%m-%d")  
df\_model2 <- dat\_converted[, c('Date', 'CPI', 'Total')]  
colnames(df\_model2) <- c('ds', 'CPI', 'y')  
df\_model2$ds <- as.Date(paste0(df\_model2$ds, "-01-01"), format = "%Y-%m-%d")

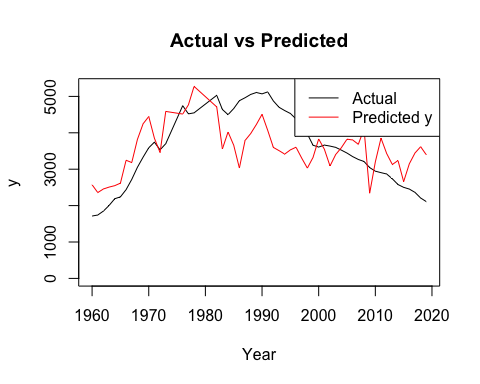
This is a prophet model based on “df\_model1”, which uses CPI as a regressor to predict Property crimes. Prophet uses a variety of methods but essentially is a high-powered Bayesian tool.Based on the output, this model is not going to work very well for forecasting. The model is erratic compared to real world data.

model <- prophet()  
model <- add\_regressor(model, 'CPI')  
model <- fit.prophet(model, df\_model1)

## Disabling weekly seasonality. Run prophet with weekly.seasonality=TRUE to override this.

## Disabling daily seasonality. Run prophet with daily.seasonality=TRUE to override this.

future <- data.frame(CPI = df\_model1$CPI)  
future$ds <- df\_model1$ds  
  
  
forecast <- predict(model, future)  
  
  
predicted\_y <- forecast$yhat  
plot(df\_model1$ds, df\_model1$y, type = 'l', ylim = c(0, max(df\_model1$y, predicted\_y)),   
 xlab = 'Year', ylab = 'y', main = 'Actual vs Predicted')  
lines(df\_model1$ds, predicted\_y, col = 'red')  
legend('topright', legend = c('Actual', 'Predicted y'), col = c('black', 'red'), lty = 1)

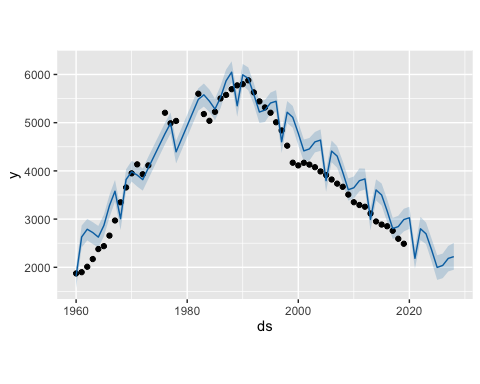
 Model 2 attempts to predict based on total crime. The model seems to track observed data pretty well. It predicts crime will spike with inflation then go back down with inflation. Here we are entering inflation values based on unofficial inflation numbers. The model predicts that total crime will increase as inflation increases.

model <- prophet()  
  
# add CPI as a regressor  
model <- add\_regressor(model, 'CPI')  
model <- fit.prophet(model, df\_model2)

## Disabling weekly seasonality. Run prophet with weekly.seasonality=TRUE to override this.

## Disabling daily seasonality. Run prophet with daily.seasonality=TRUE to override this.

future <- make\_future\_dataframe(model, periods = 9, freq = 'years')  
future$CPI <- c(1.2, 7.0, 6.9, 5.0, 3.0, 4.0, 6.0, 7.2)  
forecast <- predict(model, future)  
  
#forecast  
plot(model, forecast)

 Next we will perform simple VAR autogressions to this data.

property\_inf <- dat\_converted[, c("Property", "CPI")]  
model\_inf1 <- VAR(property\_inf, p = 3, type = "const")  
forecast\_inf1 <- predict(model\_inf1)  
forecast\_inf1

## $Property  
## fcst lower upper CI  
## [1,] 2128.638 1745.258 2512.018 383.3799  
## [2,] 2183.210 1535.830 2830.591 647.3805  
## [3,] 2260.055 1447.290 3072.819 812.7649  
## [4,] 2349.433 1368.161 3330.705 981.2717  
## [5,] 2432.998 1293.412 3572.584 1139.5860  
## [6,] 2520.616 1253.770 3787.462 1266.8461  
## [7,] 2614.957 1231.527 3998.387 1383.4297  
## [8,] 2706.862 1217.925 4195.798 1488.9365  
## [9,] 2796.357 1220.357 4372.357 1575.9999  
## [10,] 2883.811 1233.527 4534.094 1650.2834  
##   
## $CPI  
## fcst lower upper CI  
## [1,] 1.453184 -0.6128145 3.519183 2.065999  
## [2,] 1.690547 -0.8274777 4.208571 2.518024  
## [3,] 2.178307 -0.7448270 5.101440 2.923134  
## [4,] 2.471682 -0.7544808 5.697845 3.226163  
## [5,] 2.636982 -0.6798538 5.953817 3.316836  
## [6,] 2.787631 -0.5846716 6.159933 3.372302  
## [7,] 2.893231 -0.5383915 6.324854 3.431623  
## [8,] 2.981917 -0.4778286 6.441663 3.459746  
## [9,] 3.072105 -0.4030557 6.547266 3.475161  
## [10,] 3.136143 -0.3517835 6.624069 3.487926

burglary\_inf <- dat\_converted[, c("Burglary", "CPI")]  
model\_inf2 <- VAR(burglary\_inf, p = 3, type = "const")  
forecast\_inf2 <- predict(model\_inf2)  
forecast\_inf2

## $Burglary  
## fcst lower upper CI  
## [1,] 317.6093 215.17695 420.0416 102.4323  
## [2,] 298.6157 110.63126 486.6002 187.9845  
## [3,] 297.1539 40.38731 553.9205 256.7666  
## [4,] 308.0177 -22.19979 638.2352 330.2175  
## [5,] 321.8775 -74.58955 718.3445 396.4670  
## [6,] 339.1121 -112.23806 790.4622 451.3502  
## [7,] 357.8758 -142.15928 857.9108 500.0351  
## [8,] 376.0210 -166.47897 918.5209 542.4999  
## [9,] 394.2131 -184.74509 973.1713 578.9582  
## [10,] 412.3752 -198.67532 1023.4257 611.0505  
##   
## $CPI  
## fcst lower upper CI  
## [1,] 0.7489605 -1.124219 2.622140 1.873179  
## [2,] 0.4292780 -1.701579 2.560134 2.130856  
## [3,] 0.7119196 -1.840680 3.264519 2.552599  
## [4,] 0.9943012 -2.015969 4.004571 3.010270  
## [5,] 1.2632208 -1.930545 4.456987 3.193766  
## [6,] 1.4773808 -1.853034 4.807796 3.330415  
## [7,] 1.5696212 -1.855801 4.995044 3.425423  
## [8,] 1.6257998 -1.853865 5.105465 3.479665  
## [9,] 1.6848072 -1.843725 5.213339 3.528532  
## [10,] 1.7348989 -1.839958 5.309755 3.574856

total\_inf <- dat\_converted[, c("Total", "CPI")]  
model\_inf3 <- VAR(total\_inf, p = 3, type = "const")  
forecast\_inf3 <- predict(model\_inf3)  
forecast\_inf3

## $Total  
## fcst lower upper CI  
## [1,] 2499.804 2093.033 2906.576 406.7715  
## [2,] 2553.132 1857.864 3248.400 695.2679  
## [3,] 2633.682 1752.188 3515.176 881.4942  
## [4,] 2729.125 1657.225 3801.026 1071.9008  
## [5,] 2823.231 1570.553 4075.910 1252.6786  
## [6,] 2922.977 1524.185 4321.769 1398.7917  
## [7,] 3029.139 1497.795 4560.482 1531.3433  
## [8,] 3133.698 1482.682 4784.714 1651.0157  
## [9,] 3236.034 1486.181 4985.888 1749.8538  
## [10,] 3335.692 1502.122 5169.263 1833.5706  
##   
## $CPI  
## fcst lower upper CI  
## [1,] 1.499973 -0.5692372 3.569183 2.069210  
## [2,] 1.751473 -0.7668336 4.269780 2.518307  
## [3,] 2.211814 -0.6975543 5.121182 2.909368  
## [4,] 2.510380 -0.7059784 5.726738 3.216358  
## [5,] 2.695718 -0.6163684 6.007805 3.312086  
## [6,] 2.852894 -0.5146941 6.220483 3.367588  
## [7,] 2.965740 -0.4590521 6.390532 3.424792  
## [8,] 3.059858 -0.3915392 6.511256 3.451397  
## [9,] 3.148461 -0.3162334 6.613155 3.464694  
## [10,] 3.211914 -0.2633375 6.687165 3.475251

### Model Evaluation / Model Selection / Model Comparison

We performed forecasts using both a VAR and Facebook Prophet model. Both models seemed to reflect the results we saw from the correlation tests we performed but there were problems. Prophet’s predictions using real-world inflation data intuitively made sense with Total Crime numbers but there is no way to know whether they are more accurate until more real-world crime data observations are released. Current crime statistics only go as far as 2022 Q1. The Prophet model we used for property crime numbers did seem to have some fitting issues and currently would not be accurate for forecasts.

The math behind the model is described in detail in Taylor SJ, Letham B. 2017. Forecasting at scale. PeerJ Preprints 5:e3190v2 <https://doi.org/10.7287/peerj.preprints.3190v2>. This article describes how Prophet incorporates a variety of linear models and smoothers to create a polished time-series forecasting tool.

Another issue that could be addressed with further analysis is that of lag. The Granger test we performed said that lag would be an issue with the analysis. Prophet has features that allow for lag adjustments so it might be the better of the two models if we were to futher develop our research.

Last, one of Prophet’s strengths is how it handles seasonal data. Since our data was not seasonal this may have been a weakness.

### Conclusion

Some academic research has shown a curvilinear relationship between inflation and crime rates. This may explain why we struggled with the models and tests we used. A more advanced analysis may reveal more informaton about the relationship between crime and inflation. Therefore, we conclude that there is a relationship between crime and inflation but it is fairly complicated.

Additional information can be found by reading Richard Rosenfeld’s scholarship on this topic. A representative article can be found using the following citation: Journal of Quantitative Criminology (2016) 32:427–447, DOI 10.1007/s10940-016-9279-8.