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.

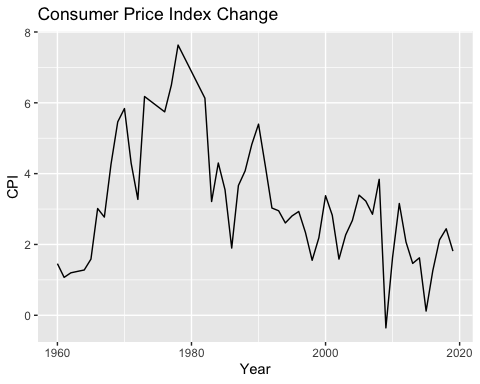
Rape statistics cannot be trusted, as prior to 2016, the FBI included only female-reported rapes, and from 2016 and forward, they included both male and female reported rapes. Therefore, we will not use it for our analysis, as it shows a massive jump in reports.

# 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)

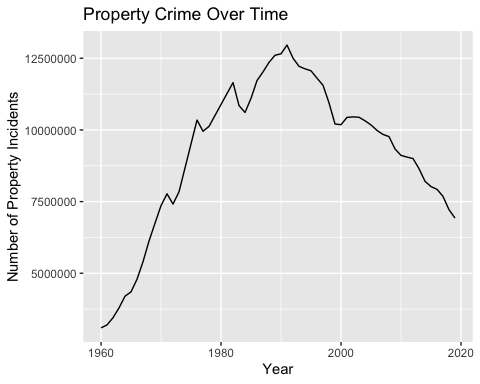
## Date CPI Tot\_Pop Total Violent Property Murder Rape Robbery assault  
## 1 1960 1.457976 180671000 3384200 288460 3095700 9110 17190 107840 154320  
## 2 1961 1.070724 183691000 3488000 289390 3198600 8740 17220 106670 156760  
## 3 1962 1.198773 186538000 3752200 301510 3450700 8530 17550 110860 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.

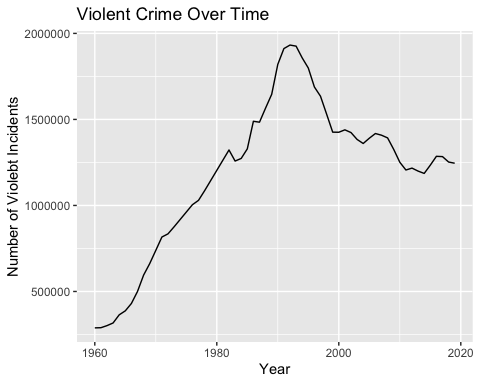
CPI\_Plot <- ggplot(dat, aes(x = Date, y = CPI)) +  
 geom\_line() +  
 labs(title = "Consumer Price Index Change", x = "Year", y = "CPI")  
  
  
Property\_Plot <- ggplot(dat, aes(x = Date, y = Property)) +  
 geom\_line() +  
 labs(title = "Property Crime Over Time", x = "Year", y = "Number of Property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



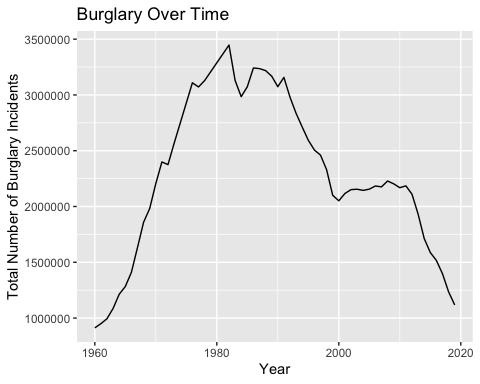
Violent\_Plot



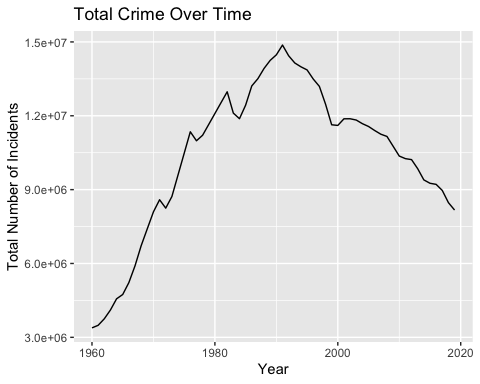
Murder\_Plot



Burglary\_Plot



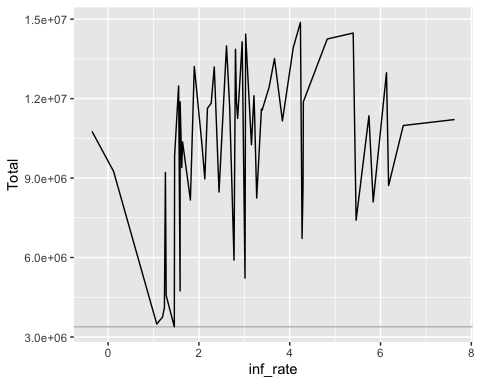
Total\_Plot



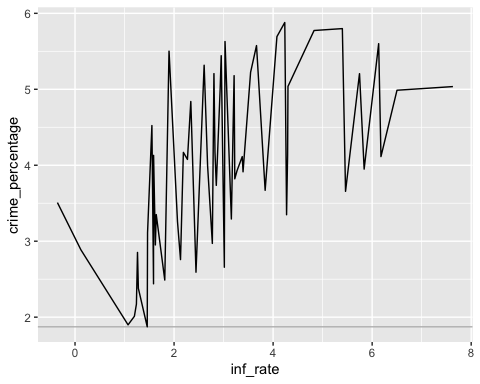
Data Visualization

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

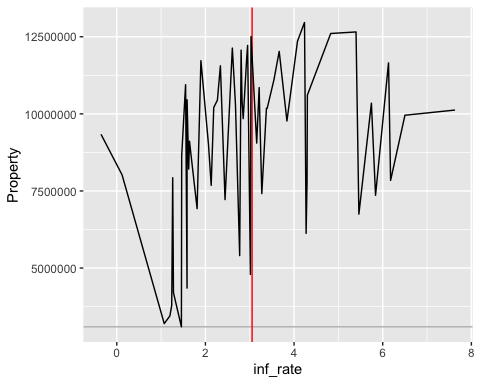
# With the previously obtained information, we can visualize tot\_crime vs inflation  
#ggplot(data= dat, mapping = aes(x = inf\_rate, y = Total))+  
# geom\_point()  
ggplot(data= dat, mapping = aes(x = inf\_rate, y = Total))+  
 geom\_hline(yintercept = min(dat$Total),color="gray")+  
 geom\_line()

 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()

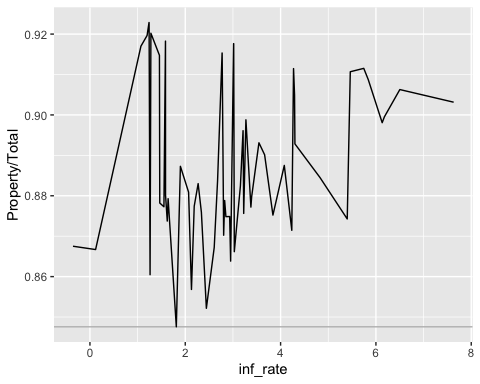
 We now should focus on crimes related to money and resources. In this dataset, these crimes would be: Property, Robbery, Burglary, Larceny theft and Vehicle theft, which are summarized by the Property type of crime.

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()



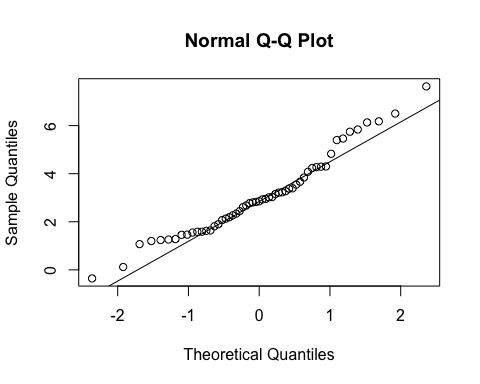
We can also try to see any relation with the percentage of financial-type crimes relative to the total amount of crimes per year vs inflation rates:

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

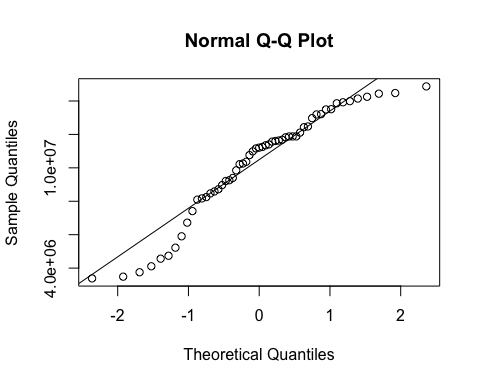


As we can see, there may be a relationship between inflation and crime; however, it is uncertain how much of a relationship there. This is why it would be useful to perform a series of correlation tests. Prior to performing these tests, let’s determine whether our data is normal using qqplots. If so,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

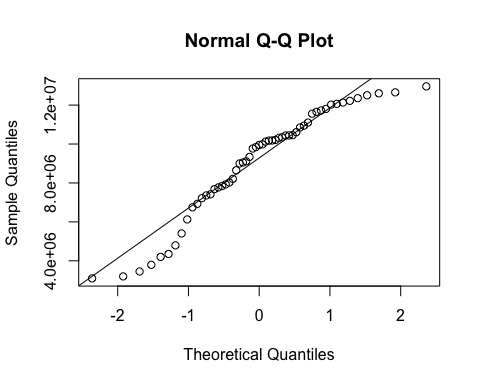
qqnorm(dat$CPI)  
qqline(dat$CPI)



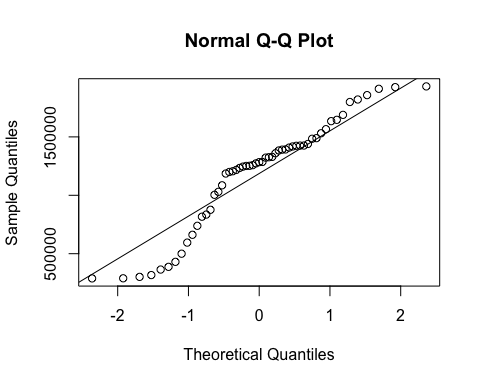
qqnorm(dat$Total)  
qqline(dat$Total)



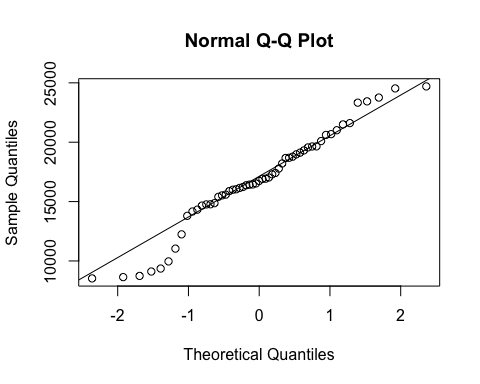
qqnorm(dat$Property)  
qqline(dat$Property)



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

## Date CPI Tot\_Pop Total Violent Property Murder Rape  
## 1 1960 1.4579760 180671000 1873.129 159.6604 1713.446 5.042314 9.514532  
## 2 1961 1.0707241 183691000 1898.841 157.5417 1741.294 4.757990 9.374439  
## 3 1962 1.1987733 186538000 2011.494 161.6346 1849.864 4.572795 9.408271  
## 4 1963 1.2396694 189242000 2171.558 167.4945 2004.048 4.565583 9.326682  
## 5 1964 1.2789116 191889000 2378.771 189.8076 2188.974 4.877820 11.162703  
## 6 1965 1.5851693 194303000 2439.180 199.3742 2239.801 5.126015 12.048193  
## 7 1966 3.0150754 196560000 2657.458 218.8543 2438.594 5.616606 13.135938  
## 8 1967 2.7727856 198712000 2970.832 251.5852 2719.262 6.159668 13.899513  
## 9 1968 4.2717962 200706000 3348.281 296.4585 3051.827 6.875729 15.779299  
## 10 1969 5.4623862 202677000 3656.508 326.5639 3329.929 7.282523 18.339525  
## 11 1970 5.8382553 205052000 3949.242 360.3086 3588.943 7.802899 18.527008  
## 12 1971 4.2927667 207661000 4135.683 393.1889 3742.494 8.562031 20.350475  
## 13 1972 3.2722782 209896000 3929.946 397.7684 3532.178 8.894881 22.320578  
## 14 1973 6.1777601 211909000 4114.077 413.3425 3700.739 9.268129 24.255695  
## 15 1976 5.7448126 218035000 5205.449 460.5728 4744.880 8.613296 26.179283  
## 16 1977 6.5016840 220239000 4987.536 467.4831 4520.090 8.681478 28.832314  
## 17 1978 7.6309638 222585000 5035.829 487.7013 4548.105 8.787654 30.374913  
## 18 1982 6.1314270 231664000 5600.525 570.8224 5029.698 9.069169 34.001830  
## 19 1983 3.2124352 233792000 5179.219 538.1236 4641.091 8.259479 33.756502  
## 20 1984 4.3005355 235825000 5038.397 539.9258 4498.463 7.925368 35.717163  
## 21 1985 3.5456442 237924000 5224.946 558.4977 4666.448 7.977337 37.268203  
## 22 1986 1.8980477 240133000 5501.896 620.1434 4881.753 8.583993 38.086810  
## 23 1987 3.6645632 242289000 5575.449 612.4913 4962.957 8.294227 37.603853  
## 24 1988 4.0777411 244499000 5694.543 640.5834 5053.968 8.458112 37.828376  
## 25 1989 4.8270030 246819000 5774.029 666.9017 5107.143 8.710837 38.287166  
## 26 1990 5.3979564 249623000 5798.985 729.1516 5069.845 9.390160 41.085958  
## 27 1991 4.2349640 252981000 5879.058 755.6971 5123.349 9.763579 42.133599  
## 28 1992 3.0288197 256514000 5628.621 753.2805 4875.328 9.262652 42.516198  
## 29 1993 2.9516570 259919000 5442.003 741.0078 4701.003 9.437556 40.785783  
## 30 1994 2.6074416 263126000 5316.654 706.0002 4610.681 8.866475 38.848308  
## 31 1995 2.8054197 266278000 5206.100 675.5308 4530.566 8.115578 36.604601  
## 32 1996 2.9312042 269394000 5008.969 626.7920 4382.169 7.294149 35.728338  
## 33 1997 2.3376899 272657000 4839.256 599.5702 4239.090 6.677987 35.265187  
## 34 1998 1.5522791 275854000 4522.550 555.0197 3967.530 6.131504 33.750825  
## 35 1999 2.1880272 279040000 4169.430 511.0536 3658.377 5.562643 32.042360  
## 36 2000 3.3768573 282162411 4113.968 505.2005 3608.768 5.523769 31.959608  
## 37 2001 2.8261711 284968955 4167.706 505.1357 3662.673 5.627631 31.885228  
## 38 2002 1.5860316 287625193 4130.012 494.9765 3635.035 5.642413 33.110799  
## 39 2003 2.2700950 290107933 4076.599 476.9521 3599.647 5.697190 32.361404  
## 40 2004 2.6772367 292805298 3988.819 464.5025 3524.317 5.514928 32.475164  
## 41 2005 3.3927468 295516599 3913.655 470.6149 3443.040 5.664656 31.926125  
## 42 2006 3.2259441 298379912 3821.139 475.2475 3345.925 5.707489 31.086878  
## 43 2007 2.8526725 301231207 3735.280 467.5269 3267.749 5.619936 30.019134  
## 44 2008 3.8391003 304093966 3670.097 457.9598 3212.137 5.406881 29.753632  
## 45 2009 -0.3555463 306771529 3508.460 432.2096 3043.653 5.019697 29.090379  
## 46 2010 1.6400434 309327143 3350.457 404.5064 2945.951 4.775527 27.670705  
## 47 2011 3.1568416 311583481 3292.464 387.0651 2905.399 4.705320 27.015232  
## 48 2012 2.0693373 313877662 3255.746 387.7520 2867.994 4.736240 27.125537  
## 49 2013 1.4648327 316059947 3116.638 379.5748 2737.063 4.530470 25.978932  
## 50 2014 1.6222230 318386329 2950.879 372.5615 2578.317 4.448683 26.654411  
## 51 2015 0.1186271 320738994 2886.552 384.7936 2501.758 4.952002 28.453354  
## 52 2016 1.2615832 323071755 2852.040 397.9320 2454.108 5.389824 40.985941  
## 53 2017 2.1301100 325122128 2757.998 394.8901 2363.108 5.319232 41.727704  
## 54 2018 2.4425833 326838199 2591.950 383.1862 2208.764 5.009818 43.986597  
## 55 2019 1.8122101 328329953 2488.682 379.3166 2109.365 5.002590 42.583687  
## Robbery assault Burglary Larceny theft Vehicle theft inf\_rate  
## 1 59.68861 85.41493 504.8403 1026.950 181.6562 8.069784e-04  
## 2 58.07035 85.33897 516.9551 1041.423 182.9159 5.828942e-04  
## 3 59.43025 88.22331 533.0281 1120.201 196.6355 6.426430e-04  
## 4 61.54553 92.05673 574.0797 1214.212 215.7555 6.550710e-04  
## 5 67.95074 105.81638 632.2405 1310.341 246.3924 6.664851e-04  
## 6 71.37821 110.82176 660.0516 1324.015 255.7346 8.158234e-04  
## 7 80.37749 119.72426 717.3891 1435.694 285.5108 1.533921e-03  
## 8 102.11261 129.41342 821.3394 1565.884 332.0383 1.395379e-03  
## 9 130.95772 142.84575 926.1806 1735.225 390.4218 2.128385e-03  
## 10 147.45136 153.49053 977.8613 1918.619 433.4483 2.695119e-03  
## 11 170.62014 163.35856 1075.3370 2060.843 452.7632 2.847207e-03  
## 12 186.69851 177.57788 1155.3927 2130.492 456.6096 2.067199e-03  
## 13 179.27450 187.27846 1131.7510 1977.741 422.6855 1.559000e-03  
## 14 181.31368 198.50502 1210.6612 2051.777 438.3013 2.915289e-03  
## 15 196.21162 229.56406 1425.7803 2876.052 443.0481 2.634812e-03  
## 16 187.34647 242.62279 1394.6213 2681.496 443.9268 2.952104e-03  
## 17 191.80538 256.73788 1405.4406 2691.556 451.1086 3.428337e-03  
## 18 238.76390 288.98750 1487.9740 3083.129 458.5952 2.646690e-03  
## 19 216.67551 279.43214 1338.7541 2871.270 431.1097 1.374057e-03  
## 20 205.66522 290.61804 1265.5147 2795.251 437.6974 1.823613e-03  
## 21 209.25590 303.98363 1291.7150 2911.182 463.5514 1.490242e-03  
## 22 226.03099 347.44163 1349.8395 3022.139 509.7746 7.904152e-04  
## 23 213.67210 352.92069 1335.6710 3095.436 531.8747 1.512476e-03  
## 24 222.07453 372.22647 1316.2017 3151.710 586.0556 1.667795e-03  
## 25 234.31340 385.59025 1283.6127 3189.544 633.9868 1.955685e-03  
## 26 256.09419 422.58125 1231.4170 3183.080 655.3483 2.162444e-03  
## 27 271.85046 431.94548 1247.9989 3218.503 656.8477 1.674025e-03  
## 28 262.16113 439.34054 1161.6910 3085.680 627.9579 1.180762e-03  
## 29 253.87525 436.90919 1090.6475 3008.976 601.3797 1.135606e-03  
## 30 235.22951 423.05967 1030.9890 2994.687 585.0049 9.909479e-04  
## 31 218.00900 412.80541 974.0947 3003.515 552.9559 1.053568e-03  
## 32 198.81289 384.95661 930.3845 2934.252 517.5319 1.088073e-03  
## 33 182.84291 375.27039 902.4254 2840.110 496.6639 8.573739e-04  
## 34 161.90630 353.23106 844.6316 2673.112 449.7865 5.627176e-04  
## 35 146.70692 326.74169 752.8451 2492.661 412.8709 7.841267e-04  
## 36 144.60324 323.11391 726.8835 2470.772 411.1115 1.196778e-03  
## 37 148.63268 318.99019 742.7234 2488.786 431.0613 9.917470e-04  
## 38 146.30360 309.91965 747.9359 2453.669 433.4273 5.514231e-04  
## 39 142.78651 296.10704 742.7698 2422.134 434.7437 7.825001e-04  
## 40 137.11159 289.40084 732.3795 2369.182 422.7557 9.143402e-04  
## 41 141.25704 291.76703 729.3831 2295.454 418.2029 1.148073e-03  
## 42 149.94408 288.50903 731.8676 2214.296 399.7618 1.081153e-03  
## 43 147.76855 284.11930 722.4152 2180.575 363.7634 9.470043e-04  
## 44 145.86741 276.93216 732.8241 2166.451 315.2410 1.262472e-03  
## 45 133.23987 264.85965 718.2260 2066.064 259.3631 -1.158994e-04  
## 46 119.31995 252.75635 701.0238 2005.838 239.0883 5.301971e-04  
## 47 113.86098 241.48360 701.3016 1974.140 229.9570 1.013161e-03  
## 48 113.11764 242.77261 672.2148 1965.375 230.4038 6.592815e-04  
## 49 109.18657 229.88519 611.2242 1904.269 221.5700 4.634667e-04  
## 50 101.41924 229.62324 538.0737 1824.530 215.7137 5.095140e-04  
## 51 102.29782 238.21768 494.9707 1784.469 222.3188 3.698557e-05  
## 52 103.01024 248.54602 469.3710 1747.239 237.4983 3.904963e-04  
## 53 98.60787 249.23527 429.6985 1695.670 237.7393 6.551723e-04  
## 54 86.06032 248.12950 377.8668 1600.843 230.0539 7.473372e-04  
## 55 81.62155 250.10877 340.4185 1549.081 219.8657 5.519478e-04  
## crime\_percentage  
## 1 0.0010367621  
## 2 0.0010337148  
## 3 0.0010783292  
## 4 0.0011475033  
## 5 0.0012396599  
## 6 0.0012553486  
## 7 0.0013519833  
## 8 0.0014950442  
## 9 0.0016682514  
## 10 0.0018041059  
## 11 0.0019259710  
## 12 0.0019915548  
## 13 0.0018723302  
## 14 0.0019414358  
## 15 0.0023874372  
## 16 0.0022646018  
## 17 0.0022624296  
## 18 0.0024175206  
## 19 0.0022153107  
## 20 0.0021364983  
## 21 0.0021960566  
## 22 0.0022911872  
## 23 0.0023011565  
## 24 0.0023290659  
## 25 0.0023393777  
## 26 0.0023230972  
## 27 0.0023239129  
## 28 0.0021942743  
## 29 0.0020937304  
## 30 0.0020205736  
## 31 0.0019551373  
## 32 0.0018593470  
## 33 0.0017748513  
## 34 0.0016394722  
## 35 0.0014942052  
## 36 0.0014580143  
## 37 0.0014625124  
## 38 0.0014359005  
## 39 0.0014052009  
## 40 0.0013622770  
## 41 0.0013243434  
## 42 0.0012806288  
## 43 0.0012400042  
## 44 0.0012068956  
## 45 0.0011436719  
## 46 0.0010831436  
## 47 0.0010566876  
## 48 0.0010372659  
## 49 0.0009860908  
## 50 0.0009268234  
## 51 0.0008999692  
## 52 0.0008827884  
## 53 0.0008482961  
## 54 0.0007930377  
## 55 0.0007579819

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 determing 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.

# Load the vars package  
library(lmtest)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(vars)

## Loading required package: MASS

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

## Loading required package: strucchange

## Loading required package: sandwich

##   
## Attaching package: 'strucchange'

## The following object is masked from 'package:stringr':  
##   
## boundary

## Loading required package: urca

# 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.

dat\_converted

## Date CPI Tot\_Pop Total Violent Property Murder Rape  
## 1 1960 1.4579760 180671000 1873.129 159.6604 1713.446 5.042314 9.514532  
## 2 1961 1.0707241 183691000 1898.841 157.5417 1741.294 4.757990 9.374439  
## 3 1962 1.1987733 186538000 2011.494 161.6346 1849.864 4.572795 9.408271  
## 4 1963 1.2396694 189242000 2171.558 167.4945 2004.048 4.565583 9.326682  
## 5 1964 1.2789116 191889000 2378.771 189.8076 2188.974 4.877820 11.162703  
## 6 1965 1.5851693 194303000 2439.180 199.3742 2239.801 5.126015 12.048193  
## 7 1966 3.0150754 196560000 2657.458 218.8543 2438.594 5.616606 13.135938  
## 8 1967 2.7727856 198712000 2970.832 251.5852 2719.262 6.159668 13.899513  
## 9 1968 4.2717962 200706000 3348.281 296.4585 3051.827 6.875729 15.779299  
## 10 1969 5.4623862 202677000 3656.508 326.5639 3329.929 7.282523 18.339525  
## 11 1970 5.8382553 205052000 3949.242 360.3086 3588.943 7.802899 18.527008  
## 12 1971 4.2927667 207661000 4135.683 393.1889 3742.494 8.562031 20.350475  
## 13 1972 3.2722782 209896000 3929.946 397.7684 3532.178 8.894881 22.320578  
## 14 1973 6.1777601 211909000 4114.077 413.3425 3700.739 9.268129 24.255695  
## 15 1976 5.7448126 218035000 5205.449 460.5728 4744.880 8.613296 26.179283  
## 16 1977 6.5016840 220239000 4987.536 467.4831 4520.090 8.681478 28.832314  
## 17 1978 7.6309638 222585000 5035.829 487.7013 4548.105 8.787654 30.374913  
## 18 1982 6.1314270 231664000 5600.525 570.8224 5029.698 9.069169 34.001830  
## 19 1983 3.2124352 233792000 5179.219 538.1236 4641.091 8.259479 33.756502  
## 20 1984 4.3005355 235825000 5038.397 539.9258 4498.463 7.925368 35.717163  
## 21 1985 3.5456442 237924000 5224.946 558.4977 4666.448 7.977337 37.268203  
## 22 1986 1.8980477 240133000 5501.896 620.1434 4881.753 8.583993 38.086810  
## 23 1987 3.6645632 242289000 5575.449 612.4913 4962.957 8.294227 37.603853  
## 24 1988 4.0777411 244499000 5694.543 640.5834 5053.968 8.458112 37.828376  
## 25 1989 4.8270030 246819000 5774.029 666.9017 5107.143 8.710837 38.287166  
## 26 1990 5.3979564 249623000 5798.985 729.1516 5069.845 9.390160 41.085958  
## 27 1991 4.2349640 252981000 5879.058 755.6971 5123.349 9.763579 42.133599  
## 28 1992 3.0288197 256514000 5628.621 753.2805 4875.328 9.262652 42.516198  
## 29 1993 2.9516570 259919000 5442.003 741.0078 4701.003 9.437556 40.785783  
## 30 1994 2.6074416 263126000 5316.654 706.0002 4610.681 8.866475 38.848308  
## 31 1995 2.8054197 266278000 5206.100 675.5308 4530.566 8.115578 36.604601  
## 32 1996 2.9312042 269394000 5008.969 626.7920 4382.169 7.294149 35.728338  
## 33 1997 2.3376899 272657000 4839.256 599.5702 4239.090 6.677987 35.265187  
## 34 1998 1.5522791 275854000 4522.550 555.0197 3967.530 6.131504 33.750825  
## 35 1999 2.1880272 279040000 4169.430 511.0536 3658.377 5.562643 32.042360  
## 36 2000 3.3768573 282162411 4113.968 505.2005 3608.768 5.523769 31.959608  
## 37 2001 2.8261711 284968955 4167.706 505.1357 3662.673 5.627631 31.885228  
## 38 2002 1.5860316 287625193 4130.012 494.9765 3635.035 5.642413 33.110799  
## 39 2003 2.2700950 290107933 4076.599 476.9521 3599.647 5.697190 32.361404  
## 40 2004 2.6772367 292805298 3988.819 464.5025 3524.317 5.514928 32.475164  
## 41 2005 3.3927468 295516599 3913.655 470.6149 3443.040 5.664656 31.926125  
## 42 2006 3.2259441 298379912 3821.139 475.2475 3345.925 5.707489 31.086878  
## 43 2007 2.8526725 301231207 3735.280 467.5269 3267.749 5.619936 30.019134  
## 44 2008 3.8391003 304093966 3670.097 457.9598 3212.137 5.406881 29.753632  
## 45 2009 -0.3555463 306771529 3508.460 432.2096 3043.653 5.019697 29.090379  
## 46 2010 1.6400434 309327143 3350.457 404.5064 2945.951 4.775527 27.670705  
## 47 2011 3.1568416 311583481 3292.464 387.0651 2905.399 4.705320 27.015232  
## 48 2012 2.0693373 313877662 3255.746 387.7520 2867.994 4.736240 27.125537  
## 49 2013 1.4648327 316059947 3116.638 379.5748 2737.063 4.530470 25.978932  
## 50 2014 1.6222230 318386329 2950.879 372.5615 2578.317 4.448683 26.654411  
## 51 2015 0.1186271 320738994 2886.552 384.7936 2501.758 4.952002 28.453354  
## 52 2016 1.2615832 323071755 2852.040 397.9320 2454.108 5.389824 40.985941  
## 53 2017 2.1301100 325122128 2757.998 394.8901 2363.108 5.319232 41.727704  
## 54 2018 2.4425833 326838199 2591.950 383.1862 2208.764 5.009818 43.986597  
## 55 2019 1.8122101 328329953 2488.682 379.3166 2109.365 5.002590 42.583687  
## Robbery assault Burglary Larceny theft Vehicle theft inf\_rate  
## 1 59.68861 85.41493 504.8403 1026.950 181.6562 8.069784e-04  
## 2 58.07035 85.33897 516.9551 1041.423 182.9159 5.828942e-04  
## 3 59.43025 88.22331 533.0281 1120.201 196.6355 6.426430e-04  
## 4 61.54553 92.05673 574.0797 1214.212 215.7555 6.550710e-04  
## 5 67.95074 105.81638 632.2405 1310.341 246.3924 6.664851e-04  
## 6 71.37821 110.82176 660.0516 1324.015 255.7346 8.158234e-04  
## 7 80.37749 119.72426 717.3891 1435.694 285.5108 1.533921e-03  
## 8 102.11261 129.41342 821.3394 1565.884 332.0383 1.395379e-03  
## 9 130.95772 142.84575 926.1806 1735.225 390.4218 2.128385e-03  
## 10 147.45136 153.49053 977.8613 1918.619 433.4483 2.695119e-03  
## 11 170.62014 163.35856 1075.3370 2060.843 452.7632 2.847207e-03  
## 12 186.69851 177.57788 1155.3927 2130.492 456.6096 2.067199e-03  
## 13 179.27450 187.27846 1131.7510 1977.741 422.6855 1.559000e-03  
## 14 181.31368 198.50502 1210.6612 2051.777 438.3013 2.915289e-03  
## 15 196.21162 229.56406 1425.7803 2876.052 443.0481 2.634812e-03  
## 16 187.34647 242.62279 1394.6213 2681.496 443.9268 2.952104e-03  
## 17 191.80538 256.73788 1405.4406 2691.556 451.1086 3.428337e-03  
## 18 238.76390 288.98750 1487.9740 3083.129 458.5952 2.646690e-03  
## 19 216.67551 279.43214 1338.7541 2871.270 431.1097 1.374057e-03  
## 20 205.66522 290.61804 1265.5147 2795.251 437.6974 1.823613e-03  
## 21 209.25590 303.98363 1291.7150 2911.182 463.5514 1.490242e-03  
## 22 226.03099 347.44163 1349.8395 3022.139 509.7746 7.904152e-04  
## 23 213.67210 352.92069 1335.6710 3095.436 531.8747 1.512476e-03  
## 24 222.07453 372.22647 1316.2017 3151.710 586.0556 1.667795e-03  
## 25 234.31340 385.59025 1283.6127 3189.544 633.9868 1.955685e-03  
## 26 256.09419 422.58125 1231.4170 3183.080 655.3483 2.162444e-03  
## 27 271.85046 431.94548 1247.9989 3218.503 656.8477 1.674025e-03  
## 28 262.16113 439.34054 1161.6910 3085.680 627.9579 1.180762e-03  
## 29 253.87525 436.90919 1090.6475 3008.976 601.3797 1.135606e-03  
## 30 235.22951 423.05967 1030.9890 2994.687 585.0049 9.909479e-04  
## 31 218.00900 412.80541 974.0947 3003.515 552.9559 1.053568e-03  
## 32 198.81289 384.95661 930.3845 2934.252 517.5319 1.088073e-03  
## 33 182.84291 375.27039 902.4254 2840.110 496.6639 8.573739e-04  
## 34 161.90630 353.23106 844.6316 2673.112 449.7865 5.627176e-04  
## 35 146.70692 326.74169 752.8451 2492.661 412.8709 7.841267e-04  
## 36 144.60324 323.11391 726.8835 2470.772 411.1115 1.196778e-03  
## 37 148.63268 318.99019 742.7234 2488.786 431.0613 9.917470e-04  
## 38 146.30360 309.91965 747.9359 2453.669 433.4273 5.514231e-04  
## 39 142.78651 296.10704 742.7698 2422.134 434.7437 7.825001e-04  
## 40 137.11159 289.40084 732.3795 2369.182 422.7557 9.143402e-04  
## 41 141.25704 291.76703 729.3831 2295.454 418.2029 1.148073e-03  
## 42 149.94408 288.50903 731.8676 2214.296 399.7618 1.081153e-03  
## 43 147.76855 284.11930 722.4152 2180.575 363.7634 9.470043e-04  
## 44 145.86741 276.93216 732.8241 2166.451 315.2410 1.262472e-03  
## 45 133.23987 264.85965 718.2260 2066.064 259.3631 -1.158994e-04  
## 46 119.31995 252.75635 701.0238 2005.838 239.0883 5.301971e-04  
## 47 113.86098 241.48360 701.3016 1974.140 229.9570 1.013161e-03  
## 48 113.11764 242.77261 672.2148 1965.375 230.4038 6.592815e-04  
## 49 109.18657 229.88519 611.2242 1904.269 221.5700 4.634667e-04  
## 50 101.41924 229.62324 538.0737 1824.530 215.7137 5.095140e-04  
## 51 102.29782 238.21768 494.9707 1784.469 222.3188 3.698557e-05  
## 52 103.01024 248.54602 469.3710 1747.239 237.4983 3.904963e-04  
## 53 98.60787 249.23527 429.6985 1695.670 237.7393 6.551723e-04  
## 54 86.06032 248.12950 377.8668 1600.843 230.0539 7.473372e-04  
## 55 81.62155 250.10877 340.4185 1549.081 219.8657 5.519478e-04  
## crime\_percentage  
## 1 0.0010367621  
## 2 0.0010337148  
## 3 0.0010783292  
## 4 0.0011475033  
## 5 0.0012396599  
## 6 0.0012553486  
## 7 0.0013519833  
## 8 0.0014950442  
## 9 0.0016682514  
## 10 0.0018041059  
## 11 0.0019259710  
## 12 0.0019915548  
## 13 0.0018723302  
## 14 0.0019414358  
## 15 0.0023874372  
## 16 0.0022646018  
## 17 0.0022624296  
## 18 0.0024175206  
## 19 0.0022153107  
## 20 0.0021364983  
## 21 0.0021960566  
## 22 0.0022911872  
## 23 0.0023011565  
## 24 0.0023290659  
## 25 0.0023393777  
## 26 0.0023230972  
## 27 0.0023239129  
## 28 0.0021942743  
## 29 0.0020937304  
## 30 0.0020205736  
## 31 0.0019551373  
## 32 0.0018593470  
## 33 0.0017748513  
## 34 0.0016394722  
## 35 0.0014942052  
## 36 0.0014580143  
## 37 0.0014625124  
## 38 0.0014359005  
## 39 0.0014052009  
## 40 0.0013622770  
## 41 0.0013243434  
## 42 0.0012806288  
## 43 0.0012400042  
## 44 0.0012068956  
## 45 0.0011436719  
## 46 0.0010831436  
## 47 0.0010566876  
## 48 0.0010372659  
## 49 0.0009860908  
## 50 0.0009268234  
## 51 0.0008999692  
## 52 0.0008827884  
## 53 0.0008482961  
## 54 0.0007930377  
## 55 0.0007579819

#Prophet requires the data to be formatted in a certain way.   
library(lubridate)

## Loading required package: timechange

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

#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")

library (prophet)

## Loading required package: Rcpp

## Loading required package: rlang

##   
## Attaching package: 'rlang'

## The following objects are masked from 'package:purrr':  
##   
## %@%, flatten, flatten\_chr, flatten\_dbl, flatten\_int, flatten\_lgl,  
## flatten\_raw, invoke, splice

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 I’m guessing there is an overfitting problem here but I’m not sure how to fix it.

model <- prophet()  
  
#Use CPI as a regressor  
model <- add\_regressor(model, 'CPI')  
model <- fit.prophet(model, df\_model1)

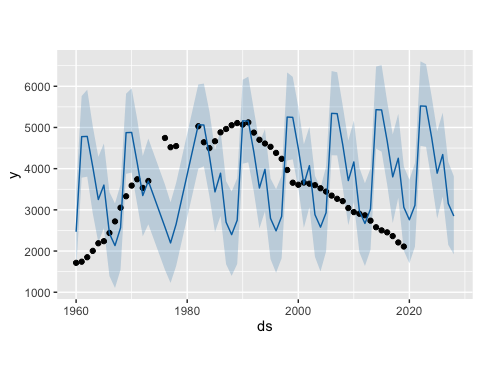
## Disabling weekly seasonality. Run prophet with weekly.seasonality=TRUE to override this.

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

future <- make\_future\_dataframe(model, periods = 9, freq = 'years')  
future$CPI <- c(1.2, 7.0, 6.9, 5.0, 3.0, 4.0, 1.0, 0.2)  
forecast <- predict(model, future)  
  
forecast

## ds trend additive\_terms additive\_terms\_lower  
## 1 1960-01-01 -350.5892659 2817.898 2817.898  
## 2 1961-01-01 -338.4667591 5115.915 5115.915  
## 3 1962-01-01 -326.3773738 5109.326 5109.326  
## 4 1963-01-01 -314.2879886 4356.930 4356.930  
## 5 1964-01-01 -302.1986034 3552.317 3552.317  
## 6 1965-01-01 -290.0760966 3891.883 3891.883  
## 7 1966-01-01 -277.9867114 2702.064 2702.064  
## 8 1967-01-01 -265.8973262 2398.480 2398.480  
## 9 1968-01-01 -253.8079410 2817.898 2817.898  
## 10 1969-01-01 -241.6854342 5115.915 5115.915  
## 11 1970-01-01 -229.5960490 5109.326 5109.326  
## 12 1971-01-01 -217.5066638 4356.930 4356.930  
## 13 1972-01-01 -205.4172786 3552.317 3552.317  
## 14 1973-01-01 -193.2947718 3891.883 3891.883  
## 15 1976-01-01 -157.0266162 2736.296 2736.296  
## 16 1977-01-01 -144.9066699 2341.443 2341.443  
## 17 1978-01-01 -132.8198383 2783.666 2783.666  
## 18 1982-01-01 -84.4393972 5150.127 5150.127  
## 19 1983-01-01 -72.4049534 5132.150 5132.150  
## 20 1984-01-01 -60.3705097 4368.338 4368.338  
## 21 1985-01-01 -48.3331329 3483.873 3483.873  
## 22 1986-01-01 -36.3286450 3926.095 3926.095  
## 23 1987-01-01 -24.3933465 2724.888 2724.888  
## 24 1988-01-01 -12.5396386 2409.888 2409.888  
## 25 1989-01-01 -0.6534547 2749.454 2749.454  
## 26 1990-01-01 11.0442338 5150.127 5150.127  
## 27 1991-01-01 22.7419224 5132.150 5132.150  
## 28 1992-01-01 34.2682165 4368.338 4368.338  
## 29 1993-01-01 45.8260894 3483.873 3483.873  
## 30 1994-01-01 57.2301153 3926.095 3926.095  
## 31 1995-01-01 68.5521028 2724.888 2724.888  
## 32 1996-01-01 79.8740903 2409.888 2409.888  
## 33 1997-01-01 91.1746057 2749.454 2749.454  
## 34 1998-01-01 102.4442455 5150.127 5150.127  
## 35 1999-01-01 113.7019620 5132.150 5132.150  
## 36 2000-01-01 124.9367602 4368.338 4368.338  
## 37 2001-01-01 136.2023386 3483.873 3483.873  
## 38 2002-01-01 147.4291519 3926.095 3926.095  
## 39 2003-01-01 158.6559652 2724.888 2724.888  
## 40 2004-01-01 169.8827681 2409.888 2409.888  
## 41 2005-01-01 181.1403293 2749.454 2749.454  
## 42 2006-01-01 192.3671322 5150.127 5150.127  
## 43 2007-01-01 203.5939350 5132.150 5132.150  
## 44 2008-01-01 214.8207379 4368.338 4368.338  
## 45 2009-01-01 226.0782991 3483.873 3483.873  
## 46 2010-01-01 237.3051020 3926.095 3926.095  
## 47 2011-01-01 248.5319048 2724.888 2724.888  
## 48 2012-01-01 259.7587077 2409.888 2409.888  
## 49 2013-01-01 271.0162689 2749.454 2749.454  
## 50 2014-01-01 282.2430718 5150.127 5150.127  
## 51 2015-01-01 293.4698746 5132.150 5132.150  
## 52 2016-01-01 304.6966775 4368.338 4368.338  
## 53 2017-01-01 315.9542387 3483.873 3483.873  
## 54 2018-01-01 327.1810416 3926.095 3926.095  
## 55 2019-01-01 338.4078444 2724.888 2724.888  
## 56 2020-01-01 349.6346473 2409.888 2409.888  
## 57 2021-01-01 360.8922085 2749.454 2749.454  
## 58 2022-01-01 372.1190114 5150.127 5150.127  
## 59 2023-01-01 383.3458142 5132.150 5132.150  
## 60 2024-01-01 394.5726171 4368.338 4368.338  
## 61 2025-01-01 405.8301783 3483.873 3483.873  
## 62 2026-01-01 417.0569812 3926.095 3926.095  
## 63 2027-01-01 428.2837840 2724.888 2724.888  
## 64 2028-01-01 439.5105869 2409.888 2409.888  
## additive\_terms\_upper CPI CPI\_lower CPI\_upper  
## 1 2817.898 -756.03864 -756.03864 -756.03864  
## 2 5115.915 1610.42241 1610.42241 1610.42241  
## 3 5109.326 1569.62136 1569.62136 1569.62136  
## 4 4356.930 794.40136 794.40136 794.40136  
## 5 3552.317 -21.61969 -21.61969 -21.61969  
## 6 3891.883 386.39083 386.39083 386.39083  
## 7 2702.064 -837.64074 -837.64074 -837.64074  
## 8 2398.480 -1164.04917 -1164.04917 -1164.04917  
## 9 2817.898 -756.03864 -756.03864 -756.03864  
## 10 5115.915 1610.42241 1610.42241 1610.42241  
## 11 5109.326 1569.62136 1569.62136 1569.62136  
## 12 4356.930 794.40136 794.40136 794.40136  
## 13 3552.317 -21.61969 -21.61969 -21.61969  
## 14 3891.883 386.39083 386.39083 386.39083  
## 15 2736.296 -837.64074 -837.64074 -837.64074  
## 16 2341.443 -1164.04917 -1164.04917 -1164.04917  
## 17 2783.666 -756.03864 -756.03864 -756.03864  
## 18 5150.127 1610.42241 1610.42241 1610.42241  
## 19 5132.150 1569.62136 1569.62136 1569.62136  
## 20 4368.338 794.40136 794.40136 794.40136  
## 21 3483.873 -21.61969 -21.61969 -21.61969  
## 22 3926.095 386.39083 386.39083 386.39083  
## 23 2724.888 -837.64074 -837.64074 -837.64074  
## 24 2409.888 -1164.04917 -1164.04917 -1164.04917  
## 25 2749.454 -756.03864 -756.03864 -756.03864  
## 26 5150.127 1610.42241 1610.42241 1610.42241  
## 27 5132.150 1569.62136 1569.62136 1569.62136  
## 28 4368.338 794.40136 794.40136 794.40136  
## 29 3483.873 -21.61969 -21.61969 -21.61969  
## 30 3926.095 386.39083 386.39083 386.39083  
## 31 2724.888 -837.64074 -837.64074 -837.64074  
## 32 2409.888 -1164.04917 -1164.04917 -1164.04917  
## 33 2749.454 -756.03864 -756.03864 -756.03864  
## 34 5150.127 1610.42241 1610.42241 1610.42241  
## 35 5132.150 1569.62136 1569.62136 1569.62136  
## 36 4368.338 794.40136 794.40136 794.40136  
## 37 3483.873 -21.61969 -21.61969 -21.61969  
## 38 3926.095 386.39083 386.39083 386.39083  
## 39 2724.888 -837.64074 -837.64074 -837.64074  
## 40 2409.888 -1164.04917 -1164.04917 -1164.04917  
## 41 2749.454 -756.03864 -756.03864 -756.03864  
## 42 5150.127 1610.42241 1610.42241 1610.42241  
## 43 5132.150 1569.62136 1569.62136 1569.62136  
## 44 4368.338 794.40136 794.40136 794.40136  
## 45 3483.873 -21.61969 -21.61969 -21.61969  
## 46 3926.095 386.39083 386.39083 386.39083  
## 47 2724.888 -837.64074 -837.64074 -837.64074  
## 48 2409.888 -1164.04917 -1164.04917 -1164.04917  
## 49 2749.454 -756.03864 -756.03864 -756.03864  
## 50 5150.127 1610.42241 1610.42241 1610.42241  
## 51 5132.150 1569.62136 1569.62136 1569.62136  
## 52 4368.338 794.40136 794.40136 794.40136  
## 53 3483.873 -21.61969 -21.61969 -21.61969  
## 54 3926.095 386.39083 386.39083 386.39083  
## 55 2724.888 -837.64074 -837.64074 -837.64074  
## 56 2409.888 -1164.04917 -1164.04917 -1164.04917  
## 57 2749.454 -756.03864 -756.03864 -756.03864  
## 58 5150.127 1610.42241 1610.42241 1610.42241  
## 59 5132.150 1569.62136 1569.62136 1569.62136  
## 60 4368.338 794.40136 794.40136 794.40136  
## 61 3483.873 -21.61969 -21.61969 -21.61969  
## 62 3926.095 386.39083 386.39083 386.39083  
## 63 2724.888 -837.64074 -837.64074 -837.64074  
## 64 2409.888 -1164.04917 -1164.04917 -1164.04917  
## extra\_regressors\_additive extra\_regressors\_additive\_lower  
## 1 -756.03864 -756.03864  
## 2 1610.42241 1610.42241  
## 3 1569.62136 1569.62136  
## 4 794.40136 794.40136  
## 5 -21.61969 -21.61969  
## 6 386.39083 386.39083  
## 7 -837.64074 -837.64074  
## 8 -1164.04917 -1164.04917  
## 9 -756.03864 -756.03864  
## 10 1610.42241 1610.42241  
## 11 1569.62136 1569.62136  
## 12 794.40136 794.40136  
## 13 -21.61969 -21.61969  
## 14 386.39083 386.39083  
## 15 -837.64074 -837.64074  
## 16 -1164.04917 -1164.04917  
## 17 -756.03864 -756.03864  
## 18 1610.42241 1610.42241  
## 19 1569.62136 1569.62136  
## 20 794.40136 794.40136  
## 21 -21.61969 -21.61969  
## 22 386.39083 386.39083  
## 23 -837.64074 -837.64074  
## 24 -1164.04917 -1164.04917  
## 25 -756.03864 -756.03864  
## 26 1610.42241 1610.42241  
## 27 1569.62136 1569.62136  
## 28 794.40136 794.40136  
## 29 -21.61969 -21.61969  
## 30 386.39083 386.39083  
## 31 -837.64074 -837.64074  
## 32 -1164.04917 -1164.04917  
## 33 -756.03864 -756.03864  
## 34 1610.42241 1610.42241  
## 35 1569.62136 1569.62136  
## 36 794.40136 794.40136  
## 37 -21.61969 -21.61969  
## 38 386.39083 386.39083  
## 39 -837.64074 -837.64074  
## 40 -1164.04917 -1164.04917  
## 41 -756.03864 -756.03864  
## 42 1610.42241 1610.42241  
## 43 1569.62136 1569.62136  
## 44 794.40136 794.40136  
## 45 -21.61969 -21.61969  
## 46 386.39083 386.39083  
## 47 -837.64074 -837.64074  
## 48 -1164.04917 -1164.04917  
## 49 -756.03864 -756.03864  
## 50 1610.42241 1610.42241  
## 51 1569.62136 1569.62136  
## 52 794.40136 794.40136  
## 53 -21.61969 -21.61969  
## 54 386.39083 386.39083  
## 55 -837.64074 -837.64074  
## 56 -1164.04917 -1164.04917  
## 57 -756.03864 -756.03864  
## 58 1610.42241 1610.42241  
## 59 1569.62136 1569.62136  
## 60 794.40136 794.40136  
## 61 -21.61969 -21.61969  
## 62 386.39083 386.39083  
## 63 -837.64074 -837.64074  
## 64 -1164.04917 -1164.04917  
## extra\_regressors\_additive\_upper yearly yearly\_lower yearly\_upper  
## 1 -756.03864 3573.937 3573.937 3573.937  
## 2 1610.42241 3505.493 3505.493 3505.493  
## 3 1569.62136 3539.704 3539.704 3539.704  
## 4 794.40136 3562.529 3562.529 3562.529  
## 5 -21.61969 3573.937 3573.937 3573.937  
## 6 386.39083 3505.493 3505.493 3505.493  
## 7 -837.64074 3539.704 3539.704 3539.704  
## 8 -1164.04917 3562.529 3562.529 3562.529  
## 9 -756.03864 3573.937 3573.937 3573.937  
## 10 1610.42241 3505.493 3505.493 3505.493  
## 11 1569.62136 3539.704 3539.704 3539.704  
## 12 794.40136 3562.529 3562.529 3562.529  
## 13 -21.61969 3573.937 3573.937 3573.937  
## 14 386.39083 3505.493 3505.493 3505.493  
## 15 -837.64074 3573.937 3573.937 3573.937  
## 16 -1164.04917 3505.493 3505.493 3505.493  
## 17 -756.03864 3539.704 3539.704 3539.704  
## 18 1610.42241 3539.704 3539.704 3539.704  
## 19 1569.62136 3562.529 3562.529 3562.529  
## 20 794.40136 3573.937 3573.937 3573.937  
## 21 -21.61969 3505.493 3505.493 3505.493  
## 22 386.39083 3539.704 3539.704 3539.704  
## 23 -837.64074 3562.529 3562.529 3562.529  
## 24 -1164.04917 3573.937 3573.937 3573.937  
## 25 -756.03864 3505.493 3505.493 3505.493  
## 26 1610.42241 3539.704 3539.704 3539.704  
## 27 1569.62136 3562.529 3562.529 3562.529  
## 28 794.40136 3573.937 3573.937 3573.937  
## 29 -21.61969 3505.493 3505.493 3505.493  
## 30 386.39083 3539.704 3539.704 3539.704  
## 31 -837.64074 3562.529 3562.529 3562.529  
## 32 -1164.04917 3573.937 3573.937 3573.937  
## 33 -756.03864 3505.493 3505.493 3505.493  
## 34 1610.42241 3539.704 3539.704 3539.704  
## 35 1569.62136 3562.529 3562.529 3562.529  
## 36 794.40136 3573.937 3573.937 3573.937  
## 37 -21.61969 3505.493 3505.493 3505.493  
## 38 386.39083 3539.704 3539.704 3539.704  
## 39 -837.64074 3562.529 3562.529 3562.529  
## 40 -1164.04917 3573.937 3573.937 3573.937  
## 41 -756.03864 3505.493 3505.493 3505.493  
## 42 1610.42241 3539.704 3539.704 3539.704  
## 43 1569.62136 3562.529 3562.529 3562.529  
## 44 794.40136 3573.937 3573.937 3573.937  
## 45 -21.61969 3505.493 3505.493 3505.493  
## 46 386.39083 3539.704 3539.704 3539.704  
## 47 -837.64074 3562.529 3562.529 3562.529  
## 48 -1164.04917 3573.937 3573.937 3573.937  
## 49 -756.03864 3505.493 3505.493 3505.493  
## 50 1610.42241 3539.704 3539.704 3539.704  
## 51 1569.62136 3562.529 3562.529 3562.529  
## 52 794.40136 3573.937 3573.937 3573.937  
## 53 -21.61969 3505.493 3505.493 3505.493  
## 54 386.39083 3539.704 3539.704 3539.704  
## 55 -837.64074 3562.529 3562.529 3562.529  
## 56 -1164.04917 3573.937 3573.937 3573.937  
## 57 -756.03864 3505.493 3505.493 3505.493  
## 58 1610.42241 3539.704 3539.704 3539.704  
## 59 1569.62136 3562.529 3562.529 3562.529  
## 60 794.40136 3573.937 3573.937 3573.937  
## 61 -21.61969 3505.493 3505.493 3505.493  
## 62 386.39083 3539.704 3539.704 3539.704  
## 63 -837.64074 3562.529 3562.529 3562.529  
## 64 -1164.04917 3573.937 3573.937 3573.937  
## multiplicative\_terms multiplicative\_terms\_lower multiplicative\_terms\_upper  
## 1 0 0 0  
## 2 0 0 0  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 0  
## 6 0 0 0  
## 7 0 0 0  
## 8 0 0 0  
## 9 0 0 0  
## 10 0 0 0  
## 11 0 0 0  
## 12 0 0 0  
## 13 0 0 0  
## 14 0 0 0  
## 15 0 0 0  
## 16 0 0 0  
## 17 0 0 0  
## 18 0 0 0  
## 19 0 0 0  
## 20 0 0 0  
## 21 0 0 0  
## 22 0 0 0  
## 23 0 0 0  
## 24 0 0 0  
## 25 0 0 0  
## 26 0 0 0  
## 27 0 0 0  
## 28 0 0 0  
## 29 0 0 0  
## 30 0 0 0  
## 31 0 0 0  
## 32 0 0 0  
## 33 0 0 0  
## 34 0 0 0  
## 35 0 0 0  
## 36 0 0 0  
## 37 0 0 0  
## 38 0 0 0  
## 39 0 0 0  
## 40 0 0 0  
## 41 0 0 0  
## 42 0 0 0  
## 43 0 0 0  
## 44 0 0 0  
## 45 0 0 0  
## 46 0 0 0  
## 47 0 0 0  
## 48 0 0 0  
## 49 0 0 0  
## 50 0 0 0  
## 51 0 0 0  
## 52 0 0 0  
## 53 0 0 0  
## 54 0 0 0  
## 55 0 0 0  
## 56 0 0 0  
## 57 0 0 0  
## 58 0 0 0  
## 59 0 0 0  
## 60 0 0 0  
## 61 0 0 0  
## 62 0 0 0  
## 63 0 0 0  
## 64 0 0 0  
## yhat\_lower yhat\_upper trend\_lower trend\_upper yhat  
## 1 1512.815 3492.716 -350.5892659 -350.5892659 2467.309  
## 2 3781.187 5763.487 -338.4667591 -338.4667591 4777.448  
## 3 3806.084 5916.507 -326.3773738 -326.3773738 4782.948  
## 4 2913.089 5028.748 -314.2879886 -314.2879886 4042.642  
## 5 2208.814 4285.309 -302.1986034 -302.1986034 3250.118  
## 6 2555.591 4611.642 -290.0760966 -290.0760966 3601.807  
## 7 1393.390 3431.067 -277.9867114 -277.9867114 2424.077  
## 8 1103.518 3155.393 -265.8973262 -265.8973262 2132.582  
## 9 1549.231 3614.403 -253.8079410 -253.8079410 2564.090  
## 10 3875.047 5820.533 -241.6854342 -241.6854342 4874.230  
## 11 3904.283 5949.274 -229.5960490 -229.5960490 4879.730  
## 12 3130.154 5199.809 -217.5066638 -217.5066638 4139.424  
## 13 2358.115 4305.944 -205.4172786 -205.4172786 3346.900  
## 14 2650.324 4730.162 -193.2947718 -193.2947718 3698.589  
## 15 1542.326 3612.104 -157.0266162 -157.0266162 2579.269  
## 16 1226.705 3182.959 -144.9066699 -144.9066699 2196.537  
## 17 1632.859 3649.864 -132.8198383 -132.8198383 2650.846  
## 18 3998.825 6043.839 -84.4393972 -84.4393972 5065.687  
## 19 4057.260 6065.612 -72.4049534 -72.4049534 5059.745  
## 20 3332.003 5387.798 -60.3705097 -60.3705097 4307.968  
## 21 2459.671 4452.639 -48.3331329 -48.3331329 3435.540  
## 22 2851.077 4931.087 -36.3286450 -36.3286450 3889.767  
## 23 1681.359 3695.790 -24.3933465 -24.3933465 2700.495  
## 24 1397.791 3444.848 -12.5396386 -12.5396386 2397.348  
## 25 1685.493 3768.033 -0.6534547 -0.6534547 2748.801  
## 26 4121.243 6159.590 11.0442338 11.0442338 5161.171  
## 27 4153.443 6233.101 22.7419224 22.7419224 5154.892  
## 28 3438.617 5429.355 34.2682165 34.2682165 4402.606  
## 29 2549.018 4626.196 45.8260894 45.8260894 3529.699  
## 30 2965.074 4957.358 57.2301153 57.2301153 3983.325  
## 31 1804.536 3762.959 68.5521028 68.5521028 2793.440  
## 32 1472.079 3502.585 79.8740903 79.8740903 2489.762  
## 33 1823.040 3825.857 91.1746057 91.1746057 2840.629  
## 34 4196.373 6335.757 102.4442455 102.4442455 5252.571  
## 35 4226.575 6228.706 113.7019620 113.7019620 5245.852  
## 36 3526.869 5500.352 124.9367602 124.9367602 4493.275  
## 37 2562.815 4590.171 136.2023386 136.2023386 3620.075  
## 38 3064.448 5014.252 147.4291519 147.4291519 4073.524  
## 39 1859.827 3827.457 158.6559652 158.6559652 2883.544  
## 40 1504.667 3531.589 169.8827681 169.8827681 2579.770  
## 41 1989.337 4048.586 181.1403293 181.1403293 2930.594  
## 42 4325.766 6369.418 192.3671322 192.3671322 5342.494  
## 43 4322.087 6340.127 203.5939350 203.5939350 5335.744  
## 44 3536.878 5529.932 214.8207379 214.8207379 4583.159  
## 45 2629.431 4678.584 226.0782991 226.0782991 3709.951  
## 46 3182.523 5167.927 237.3051020 237.3051020 4163.400  
## 47 1971.722 4022.043 248.5319048 248.5319048 2973.420  
## 48 1637.648 3645.634 259.7587077 259.7587077 2669.646  
## 49 2031.912 4005.624 271.0162689 271.0162689 3020.470  
## 50 4485.084 6478.284 282.2430718 282.2430718 5432.370  
## 51 4413.170 6514.193 293.4698746 293.4698746 5425.620  
## 52 3613.218 5632.305 304.6966775 304.6966775 4673.035  
## 53 2677.198 4828.508 315.9542387 315.9542387 3799.827  
## 54 3294.000 5332.105 327.1810416 327.1810416 4253.276  
## 55 2008.958 4052.601 338.4078444 338.4078444 3063.296  
## 56 1700.992 3731.247 349.6235621 349.6404616 2759.522  
## 57 2106.887 4132.415 360.8452745 360.9333501 3110.346  
## 58 4553.303 6605.063 372.0244254 372.2115009 5522.246  
## 59 4529.745 6536.006 383.1986495 383.4899105 5515.496  
## 60 3702.734 5760.472 394.3637486 394.7797839 4762.911  
## 61 2788.134 4906.563 405.5507411 406.1103644 3889.703  
## 62 3323.865 5359.228 416.6951326 417.4187198 4343.152  
## 63 2151.235 4163.697 427.8370631 428.7256236 3153.172  
## 64 1918.931 3816.391 438.9617515 440.0399833 2849.398

plot(model, forecast)

 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)

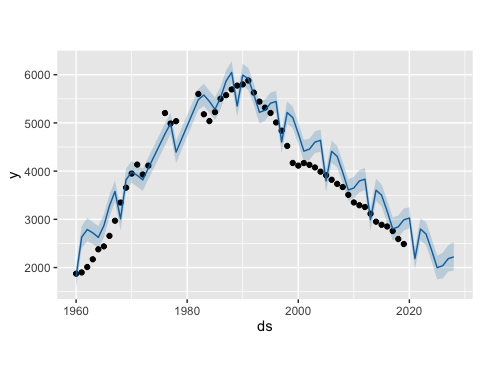
## Disabling weekly seasonality. Run prophet with weekly.seasonality=TRUE to override this.

## 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

## ds trend additive\_terms additive\_terms\_lower  
## 1 1960-01-01 -547.90597 2363.506 2363.506  
## 2 1961-01-01 -398.58431 3028.309 3028.309  
## 3 1962-01-01 -249.67062 3039.192 3039.192  
## 4 1963-01-01 -100.75693 2821.565 2821.565  
## 5 1964-01-01 48.15676 2577.817 2577.817  
## 6 1965-01-01 197.47843 2671.124 2671.124  
## 7 1966-01-01 346.39211 2932.037 2932.037  
## 8 1967-01-01 495.30580 3083.500 3083.500  
## 9 1968-01-01 644.21949 2363.506 2363.506  
## 10 1969-01-01 793.54116 3028.309 3028.309  
## 11 1970-01-01 942.45341 3039.192 3039.192  
## 12 1971-01-01 1091.36389 2821.565 2821.565  
## 13 1972-01-01 1240.27436 2577.817 2577.817  
## 14 1973-01-01 1389.56427 2671.124 2671.124  
## 15 1976-01-01 1836.21029 2935.002 2935.002  
## 16 1977-01-01 1935.63665 3052.121 3052.121  
## 17 1978-01-01 2034.77614 2360.541 2360.541  
## 18 1982-01-01 2431.60570 3051.098 3051.098  
## 19 1983-01-01 2530.71770 3047.782 3047.782  
## 20 1984-01-01 2629.82970 2815.940 2815.940  
## 21 1985-01-01 2729.14890 2552.063 2552.063  
## 22 1986-01-01 2828.19673 2693.913 2693.913  
## 23 1987-01-01 2927.20645 2940.626 2940.626  
## 24 1988-01-01 2969.79322 3077.876 3077.876  
## 25 1989-01-01 3012.49666 2337.752 2337.752  
## 26 1990-01-01 2941.13051 3051.098 3051.098  
## 27 1991-01-01 2869.76436 3047.782 3047.782  
## 28 1992-01-01 2769.17342 2815.940 2815.940  
## 29 1993-01-01 2668.30689 2552.063 2552.063  
## 30 1994-01-01 2567.66800 2693.913 2693.913  
## 31 1995-01-01 2467.02733 2940.626 2940.626  
## 32 1996-01-01 2366.38665 3077.876 3077.876  
## 33 1997-01-01 2265.47025 2337.752 2337.752  
## 34 1998-01-01 2164.82958 3051.098 3051.098  
## 35 1999-01-01 2064.18890 3047.782 3047.782  
## 36 2000-01-01 1963.54823 2815.940 2815.940  
## 37 2001-01-01 1862.63183 2552.063 2552.063  
## 38 2002-01-01 1761.99116 2693.913 2693.913  
## 39 2003-01-01 1661.35048 2940.626 2940.626  
## 40 2004-01-01 1560.70981 3077.876 3077.876  
## 41 2005-01-01 1459.79341 2337.752 2337.752  
## 42 2006-01-01 1359.15274 3051.098 3051.098  
## 43 2007-01-01 1258.51206 3047.782 3047.782  
## 44 2008-01-01 1157.87139 2815.940 2815.940  
## 45 2009-01-01 1056.95499 2552.063 2552.063  
## 46 2010-01-01 956.31432 2693.913 2693.913  
## 47 2011-01-01 855.67364 2940.626 2940.626  
## 48 2012-01-01 755.03297 3077.876 3077.876  
## 49 2013-01-01 654.11657 2337.752 2337.752  
## 50 2014-01-01 553.47590 3051.098 3051.098  
## 51 2015-01-01 452.83523 3047.782 3047.782  
## 52 2016-01-01 352.19455 2815.940 2815.940  
## 53 2017-01-01 251.27815 2552.063 2552.063  
## 54 2018-01-01 150.63748 2693.913 2693.913  
## 55 2019-01-01 49.99681 2940.626 2940.626  
## 56 2020-01-01 -50.64387 3077.876 3077.876  
## 57 2021-01-01 -151.56027 2337.752 2337.752  
## 58 2022-01-01 -252.20094 3051.098 3051.098  
## 59 2023-01-01 -352.84161 3047.782 3047.782  
## 60 2024-01-01 -453.48228 2815.940 2815.940  
## 61 2025-01-01 -554.39869 2552.063 2552.063  
## 62 2026-01-01 -655.03936 2693.913 2693.913  
## 63 2027-01-01 -755.68003 2940.626 2940.626  
## 64 2028-01-01 -856.32070 3077.876 3077.876  
## additive\_terms\_upper CPI CPI\_lower CPI\_upper  
## 1 2363.506 -220.619735 -220.619735 -220.619735  
## 2 3028.309 469.937574 469.937574 469.937574  
## 3 3039.192 458.031413 458.031413 458.031413  
## 4 2821.565 231.814364 231.814364 231.814364  
## 5 2577.817 -6.308846 -6.308846 -6.308846  
## 6 2671.124 112.752759 112.752759 112.752759  
## 7 2932.037 350.875969 350.875969 350.875969  
## 8 3083.500 493.749895 493.749895 493.749895  
## 9 2363.506 -220.619735 -220.619735 -220.619735  
## 10 3028.309 469.937574 469.937574 469.937574  
## 11 3039.192 458.031413 458.031413 458.031413  
## 12 2821.565 231.814364 231.814364 231.814364  
## 13 2577.817 -6.308846 -6.308846 -6.308846  
## 14 2671.124 112.752759 112.752759 112.752759  
## 15 2935.002 350.875969 350.875969 350.875969  
## 16 3052.121 493.749895 493.749895 493.749895  
## 17 2360.541 -220.619735 -220.619735 -220.619735  
## 18 3051.098 469.937574 469.937574 469.937574  
## 19 3047.782 458.031413 458.031413 458.031413  
## 20 2815.940 231.814364 231.814364 231.814364  
## 21 2552.063 -6.308846 -6.308846 -6.308846  
## 22 2693.913 112.752759 112.752759 112.752759  
## 23 2940.626 350.875969 350.875969 350.875969  
## 24 3077.876 493.749895 493.749895 493.749895  
## 25 2337.752 -220.619735 -220.619735 -220.619735  
## 26 3051.098 469.937574 469.937574 469.937574  
## 27 3047.782 458.031413 458.031413 458.031413  
## 28 2815.940 231.814364 231.814364 231.814364  
## 29 2552.063 -6.308846 -6.308846 -6.308846  
## 30 2693.913 112.752759 112.752759 112.752759  
## 31 2940.626 350.875969 350.875969 350.875969  
## 32 3077.876 493.749895 493.749895 493.749895  
## 33 2337.752 -220.619735 -220.619735 -220.619735  
## 34 3051.098 469.937574 469.937574 469.937574  
## 35 3047.782 458.031413 458.031413 458.031413  
## 36 2815.940 231.814364 231.814364 231.814364  
## 37 2552.063 -6.308846 -6.308846 -6.308846  
## 38 2693.913 112.752759 112.752759 112.752759  
## 39 2940.626 350.875969 350.875969 350.875969  
## 40 3077.876 493.749895 493.749895 493.749895  
## 41 2337.752 -220.619735 -220.619735 -220.619735  
## 42 3051.098 469.937574 469.937574 469.937574  
## 43 3047.782 458.031413 458.031413 458.031413  
## 44 2815.940 231.814364 231.814364 231.814364  
## 45 2552.063 -6.308846 -6.308846 -6.308846  
## 46 2693.913 112.752759 112.752759 112.752759  
## 47 2940.626 350.875969 350.875969 350.875969  
## 48 3077.876 493.749895 493.749895 493.749895  
## 49 2337.752 -220.619735 -220.619735 -220.619735  
## 50 3051.098 469.937574 469.937574 469.937574  
## 51 3047.782 458.031413 458.031413 458.031413  
## 52 2815.940 231.814364 231.814364 231.814364  
## 53 2552.063 -6.308846 -6.308846 -6.308846  
## 54 2693.913 112.752759 112.752759 112.752759  
## 55 2940.626 350.875969 350.875969 350.875969  
## 56 3077.876 493.749895 493.749895 493.749895  
## 57 2337.752 -220.619735 -220.619735 -220.619735  
## 58 3051.098 469.937574 469.937574 469.937574  
## 59 3047.782 458.031413 458.031413 458.031413  
## 60 2815.940 231.814364 231.814364 231.814364  
## 61 2552.063 -6.308846 -6.308846 -6.308846  
## 62 2693.913 112.752759 112.752759 112.752759  
## 63 2940.626 350.875969 350.875969 350.875969  
## 64 3077.876 493.749895 493.749895 493.749895  
## extra\_regressors\_additive extra\_regressors\_additive\_lower  
## 1 -220.619735 -220.619735  
## 2 469.937574 469.937574  
## 3 458.031413 458.031413  
## 4 231.814364 231.814364  
## 5 -6.308846 -6.308846  
## 6 112.752759 112.752759  
## 7 350.875969 350.875969  
## 8 493.749895 493.749895  
## 9 -220.619735 -220.619735  
## 10 469.937574 469.937574  
## 11 458.031413 458.031413  
## 12 231.814364 231.814364  
## 13 -6.308846 -6.308846  
## 14 112.752759 112.752759  
## 15 350.875969 350.875969  
## 16 493.749895 493.749895  
## 17 -220.619735 -220.619735  
## 18 469.937574 469.937574  
## 19 458.031413 458.031413  
## 20 231.814364 231.814364  
## 21 -6.308846 -6.308846  
## 22 112.752759 112.752759  
## 23 350.875969 350.875969  
## 24 493.749895 493.749895  
## 25 -220.619735 -220.619735  
## 26 469.937574 469.937574  
## 27 458.031413 458.031413  
## 28 231.814364 231.814364  
## 29 -6.308846 -6.308846  
## 30 112.752759 112.752759  
## 31 350.875969 350.875969  
## 32 493.749895 493.749895  
## 33 -220.619735 -220.619735  
## 34 469.937574 469.937574  
## 35 458.031413 458.031413  
## 36 231.814364 231.814364  
## 37 -6.308846 -6.308846  
## 38 112.752759 112.752759  
## 39 350.875969 350.875969  
## 40 493.749895 493.749895  
## 41 -220.619735 -220.619735  
## 42 469.937574 469.937574  
## 43 458.031413 458.031413  
## 44 231.814364 231.814364  
## 45 -6.308846 -6.308846  
## 46 112.752759 112.752759  
## 47 350.875969 350.875969  
## 48 493.749895 493.749895  
## 49 -220.619735 -220.619735  
## 50 469.937574 469.937574  
## 51 458.031413 458.031413  
## 52 231.814364 231.814364  
## 53 -6.308846 -6.308846  
## 54 112.752759 112.752759  
## 55 350.875969 350.875969  
## 56 493.749895 493.749895  
## 57 -220.619735 -220.619735  
## 58 469.937574 469.937574  
## 59 458.031413 458.031413  
## 60 231.814364 231.814364  
## 61 -6.308846 -6.308846  
## 62 112.752759 112.752759  
## 63 350.875969 350.875969  
## 64 493.749895 493.749895  
## extra\_regressors\_additive\_upper yearly yearly\_lower yearly\_upper  
## 1 -220.619735 2584.126 2584.126 2584.126  
## 2 469.937574 2558.371 2558.371 2558.371  
## 3 458.031413 2581.161 2581.161 2581.161  
## 4 231.814364 2589.750 2589.750 2589.750  
## 5 -6.308846 2584.126 2584.126 2584.126  
## 6 112.752759 2558.371 2558.371 2558.371  
## 7 350.875969 2581.161 2581.161 2581.161  
## 8 493.749895 2589.750 2589.750 2589.750  
## 9 -220.619735 2584.126 2584.126 2584.126  
## 10 469.937574 2558.371 2558.371 2558.371  
## 11 458.031413 2581.161 2581.161 2581.161  
## 12 231.814364 2589.750 2589.750 2589.750  
## 13 -6.308846 2584.126 2584.126 2584.126  
## 14 112.752759 2558.371 2558.371 2558.371  
## 15 350.875969 2584.126 2584.126 2584.126  
## 16 493.749895 2558.371 2558.371 2558.371  
## 17 -220.619735 2581.161 2581.161 2581.161  
## 18 469.937574 2581.161 2581.161 2581.161  
## 19 458.031413 2589.750 2589.750 2589.750  
## 20 231.814364 2584.126 2584.126 2584.126  
## 21 -6.308846 2558.371 2558.371 2558.371  
## 22 112.752759 2581.161 2581.161 2581.161  
## 23 350.875969 2589.750 2589.750 2589.750  
## 24 493.749895 2584.126 2584.126 2584.126  
## 25 -220.619735 2558.371 2558.371 2558.371  
## 26 469.937574 2581.161 2581.161 2581.161  
## 27 458.031413 2589.750 2589.750 2589.750  
## 28 231.814364 2584.126 2584.126 2584.126  
## 29 -6.308846 2558.371 2558.371 2558.371  
## 30 112.752759 2581.161 2581.161 2581.161  
## 31 350.875969 2589.750 2589.750 2589.750  
## 32 493.749895 2584.126 2584.126 2584.126  
## 33 -220.619735 2558.371 2558.371 2558.371  
## 34 469.937574 2581.161 2581.161 2581.161  
## 35 458.031413 2589.750 2589.750 2589.750  
## 36 231.814364 2584.126 2584.126 2584.126  
## 37 -6.308846 2558.371 2558.371 2558.371  
## 38 112.752759 2581.161 2581.161 2581.161  
## 39 350.875969 2589.750 2589.750 2589.750  
## 40 493.749895 2584.126 2584.126 2584.126  
## 41 -220.619735 2558.371 2558.371 2558.371  
## 42 469.937574 2581.161 2581.161 2581.161  
## 43 458.031413 2589.750 2589.750 2589.750  
## 44 231.814364 2584.126 2584.126 2584.126  
## 45 -6.308846 2558.371 2558.371 2558.371  
## 46 112.752759 2581.161 2581.161 2581.161  
## 47 350.875969 2589.750 2589.750 2589.750  
## 48 493.749895 2584.126 2584.126 2584.126  
## 49 -220.619735 2558.371 2558.371 2558.371  
## 50 469.937574 2581.161 2581.161 2581.161  
## 51 458.031413 2589.750 2589.750 2589.750  
## 52 231.814364 2584.126 2584.126 2584.126  
## 53 -6.308846 2558.371 2558.371 2558.371  
## 54 112.752759 2581.161 2581.161 2581.161  
## 55 350.875969 2589.750 2589.750 2589.750  
## 56 493.749895 2584.126 2584.126 2584.126  
## 57 -220.619735 2558.371 2558.371 2558.371  
## 58 469.937574 2581.161 2581.161 2581.161  
## 59 458.031413 2589.750 2589.750 2589.750  
## 60 231.814364 2584.126 2584.126 2584.126  
## 61 -6.308846 2558.371 2558.371 2558.371  
## 62 112.752759 2581.161 2581.161 2581.161  
## 63 350.875969 2589.750 2589.750 2589.750  
## 64 493.749895 2584.126 2584.126 2584.126  
## multiplicative\_terms multiplicative\_terms\_lower multiplicative\_terms\_upper  
## 1 0 0 0  
## 2 0 0 0  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 0  
## 6 0 0 0  
## 7 0 0 0  
## 8 0 0 0  
## 9 0 0 0  
## 10 0 0 0  
## 11 0 0 0  
## 12 0 0 0  
## 13 0 0 0  
## 14 0 0 0  
## 15 0 0 0  
## 16 0 0 0  
## 17 0 0 0  
## 18 0 0 0  
## 19 0 0 0  
## 20 0 0 0  
## 21 0 0 0  
## 22 0 0 0  
## 23 0 0 0  
## 24 0 0 0  
## 25 0 0 0  
## 26 0 0 0  
## 27 0 0 0  
## 28 0 0 0  
## 29 0 0 0  
## 30 0 0 0  
## 31 0 0 0  
## 32 0 0 0  
## 33 0 0 0  
## 34 0 0 0  
## 35 0 0 0  
## 36 0 0 0  
## 37 0 0 0  
## 38 0 0 0  
## 39 0 0 0  
## 40 0 0 0  
## 41 0 0 0  
## 42 0 0 0  
## 43 0 0 0  
## 44 0 0 0  
## 45 0 0 0  
## 46 0 0 0  
## 47 0 0 0  
## 48 0 0 0  
## 49 0 0 0  
## 50 0 0 0  
## 51 0 0 0  
## 52 0 0 0  
## 53 0 0 0  
## 54 0 0 0  
## 55 0 0 0  
## 56 0 0 0  
## 57 0 0 0  
## 58 0 0 0  
## 59 0 0 0  
## 60 0 0 0  
## 61 0 0 0  
## 62 0 0 0  
## 63 0 0 0  
## 64 0 0 0  
## yhat\_lower yhat\_upper trend\_lower trend\_upper yhat  
## 1 1576.719 2040.643 -547.90597 -547.90597 1815.600  
## 2 2398.508 2858.241 -398.58431 -398.58431 2629.725  
## 3 2547.035 3032.079 -249.67062 -249.67062 2789.521  
## 4 2489.530 2943.670 -100.75693 -100.75693 2720.808  
## 5 2400.974 2862.339 48.15676 48.15676 2625.974  
## 6 2622.015 3096.840 197.47843 197.47843 2868.603  
## 7 3040.167 3509.166 346.39211 346.39211 3278.429  
## 8 3365.298 3804.354 495.30580 495.30580 3578.806  
## 9 2769.177 3244.194 644.21949 644.21949 3007.726  
## 10 3588.605 4055.113 793.54116 793.54116 3821.850  
## 11 3760.956 4213.449 942.45341 942.45341 3981.645  
## 12 3674.508 4128.519 1091.36389 1091.36389 3912.929  
## 13 3591.406 4046.557 1240.27436 1240.27436 3818.091  
## 14 3828.390 4287.387 1389.56427 1389.56427 4060.688  
## 15 4527.615 4988.634 1836.21029 1836.21029 4771.212  
## 16 4768.865 5208.010 1935.63665 1935.63665 4987.758  
## 17 4166.066 4624.545 2034.77614 2034.77614 4395.317  
## 18 5269.263 5706.071 2431.60570 2431.60570 5482.704  
## 19 5345.982 5812.015 2530.71770 2530.71770 5578.500  
## 20 5211.834 5666.237 2629.82970 2629.82970 5445.770  
## 21 5058.168 5531.223 2729.14890 2729.14890 5281.211  
## 22 5283.420 5751.458 2828.19673 2828.19673 5522.110  
## 23 5625.979 6087.512 2927.20645 2927.20645 5867.833  
## 24 5820.907 6276.807 2969.79322 2969.79322 6047.669  
## 25 5119.658 5578.227 3012.49666 3012.49666 5350.248  
## 26 5758.396 6227.444 2941.13051 2941.13051 5992.229  
## 27 5687.361 6145.245 2869.76436 2869.76436 5917.546  
## 28 5370.571 5810.905 2769.17342 2769.17342 5585.114  
## 29 4985.597 5450.239 2668.30689 2668.30689 5220.369  
## 30 5044.492 5484.717 2567.66800 2567.66800 5261.581  
## 31 5195.475 5628.876 2467.02733 2467.02733 5407.654  
## 32 5214.317 5665.218 2366.38665 2366.38665 5444.262  
## 33 4369.910 4835.558 2265.47025 2265.47025 4603.222  
## 34 4975.420 5446.279 2164.82958 2164.82958 5215.928  
## 35 4896.875 5341.470 2064.18890 2064.18890 5111.971  
## 36 4561.676 5002.610 1963.54823 1963.54823 4779.488  
## 37 4196.664 4649.939 1862.63183 1862.63183 4414.694  
## 38 4230.466 4674.086 1761.99116 1761.99116 4455.905  
## 39 4375.611 4826.236 1661.35048 1661.35048 4601.977  
## 40 4400.011 4859.670 1560.70981 1560.70981 4638.586  
## 41 3562.447 4019.363 1459.79341 1459.79341 3797.545  
## 42 4177.409 4633.911 1359.15274 1359.15274 4410.251  
## 43 4077.264 4523.759 1258.51206 1258.51206 4306.294  
## 44 3738.297 4192.909 1157.87139 1157.87139 3973.812  
## 45 3379.001 3830.509 1056.95499 1056.95499 3609.017  
## 46 3442.068 3881.357 956.31432 956.31432 3650.228  
## 47 3559.996 4022.398 855.67364 855.67364 3796.300  
## 48 3606.889 4064.559 755.03297 755.03297 3832.909  
## 49 2754.395 3217.804 654.11657 654.11657 2991.868  
## 50 3382.235 3842.442 553.47590 553.47590 3604.574  
## 51 3264.970 3726.317 452.83523 452.83523 3500.617  
## 52 2926.356 3396.956 352.19455 352.19455 3168.135  
## 53 2561.339 3038.063 251.27815 251.27815 2803.341  
## 54 2617.973 3079.372 150.63748 150.63748 2844.551  
## 55 2767.963 3225.215 49.99681 49.99681 2990.623  
## 56 2811.322 3249.623 -52.01788 -48.23578 3027.232  
## 57 1966.635 2415.370 -162.69157 -138.10768 2186.191  
## 58 2562.148 3020.645 -275.82257 -222.77629 2798.897  
## 59 2463.740 2942.061 -389.91779 -302.30810 2694.940  
## 60 2128.743 2609.621 -506.92501 -383.93797 2362.458  
## 61 1754.549 2244.940 -629.06782 -459.75161 1997.664  
## 62 1775.983 2306.574 -752.57881 -533.81088 2038.874  
## 63 1915.434 2465.014 -876.87771 -603.61462 2184.946  
## 64 1936.843 2526.915 -1001.85647 -684.21240 2221.555

plot(model, forecast)

 Next we will perform a simple VAR autogression to this data.

# Load the vars package  
library(vars)  
library(lubridate)  
library(dplyr)  
library (forecast)

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

property\_inf <- dat\_converted[, c("Property", "CPI")]  
  
model\_inf <- VAR(property\_inf, p = 3, type = "const")  
  
forecast\_inf <- predict(model\_inf, n.ahead = 3, newdata = data.frame(inf\_rate = c(1.0, 3.1, 7)))  
  
  
forecast\_inf

## $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  
##   
## $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

### Model Evaluation / Model Selection / Model Comparison

### Conclusion