

# ISyE 6402 Project

Siddharth Sen

11/04/2022

## Part II - ARIMA Modelling

```
library(zoo)
library(xts)
library(lubridate)
library(mgcv)
library(lmtest)
```

### Load data

```
## Atlanta
atl.v.df <- read.csv("atl_violent_final.csv", head = TRUE)
atl.p.df <- read.csv("atl_prop_final.csv", head = TRUE)
colnames(atl.v.df) <- c("Date", "violentCrime")
colnames(atl.p.df) <- c("Date", "propertyCrime")

atl.df <- merge(atl.v.df, atl.p.df)
atl.df$Date <- as.Date(atl.df$Date, "%Y-%m-%d")
atl.df <- atl.df[atl.df$Date <= "2020-12-31", ]
atl.df <- na.locf(atl.df)

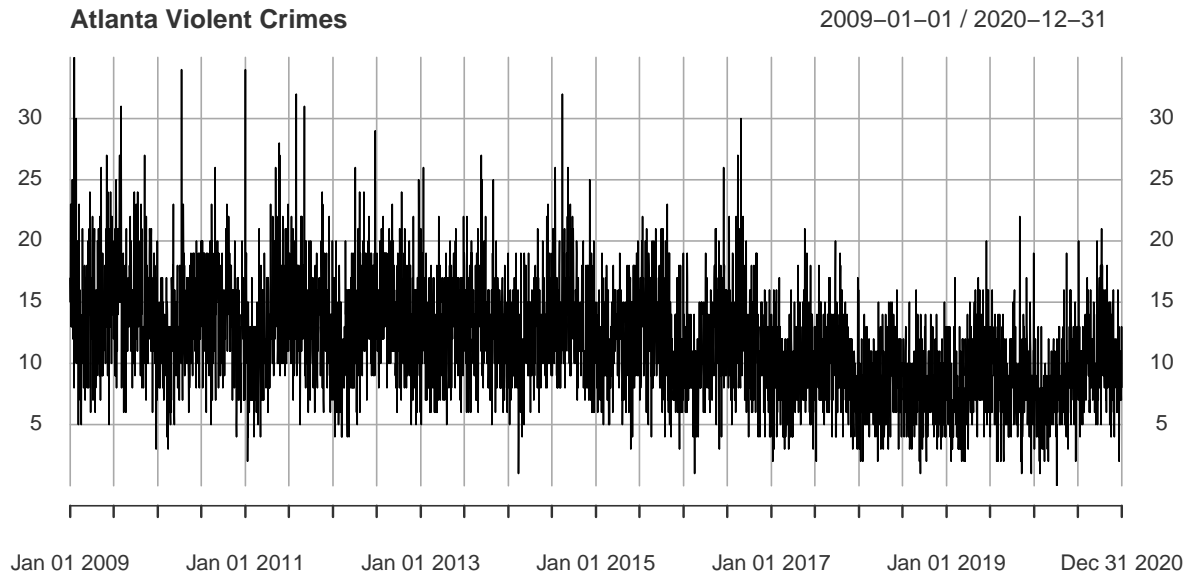
## New York City
nyc.v.df <- read.csv("nyc_violent_final.csv", head = TRUE)
nyc.p.df <- read.csv("nyc_prop_final.csv", head = TRUE)
nyc.q.df <- read.csv("nyc_QOL_final.csv", head = TRUE)
colnames(nyc.v.df) <- c("Date", "violentCrime")
colnames(nyc.p.df) <- c("Date", "propertyCrime")
colnames(nyc.q.df) <- c("Date", "qolCrime")

nyc.df <- merge(merge(nyc.v.df, nyc.p.df), nyc.q.df)
nyc.df$Date <- as.Date(nyc.df$Date, "%Y-%m-%d")
nyc.df <- nyc.df[nyc.df$Date >= "2009-01-01", ]
nyc.df <- na.locf(nyc.df)
```

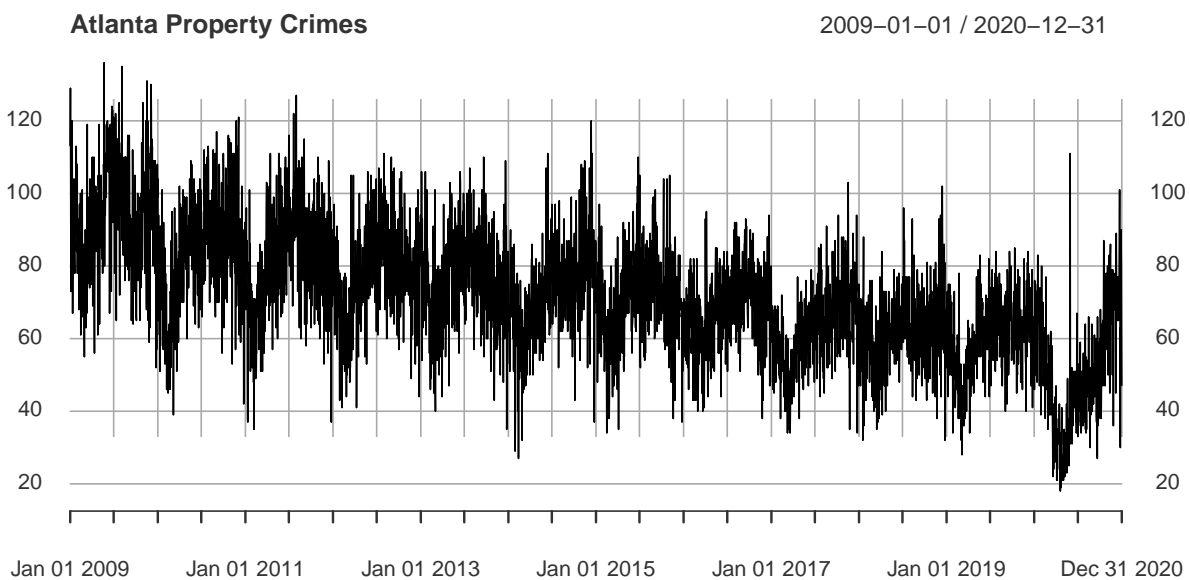
## Plot Time Series

```
## ATL TS
atl.v.ts <- ts(atl.df$violentCrime, start = 2009, freq = 365.25)
atl.p.ts <- ts(atl.df$propertyCrime, start = 2009, freq = 365.25)

plot(xts(atl.df$violentCrime, atl.df$Date), main="Atlanta Violent Crimes", lwd=1)
```



```
plot(xts(atl.df$propertyCrime, atl.df$Date), main="Atlanta Property Crimes", lwd=1)
```



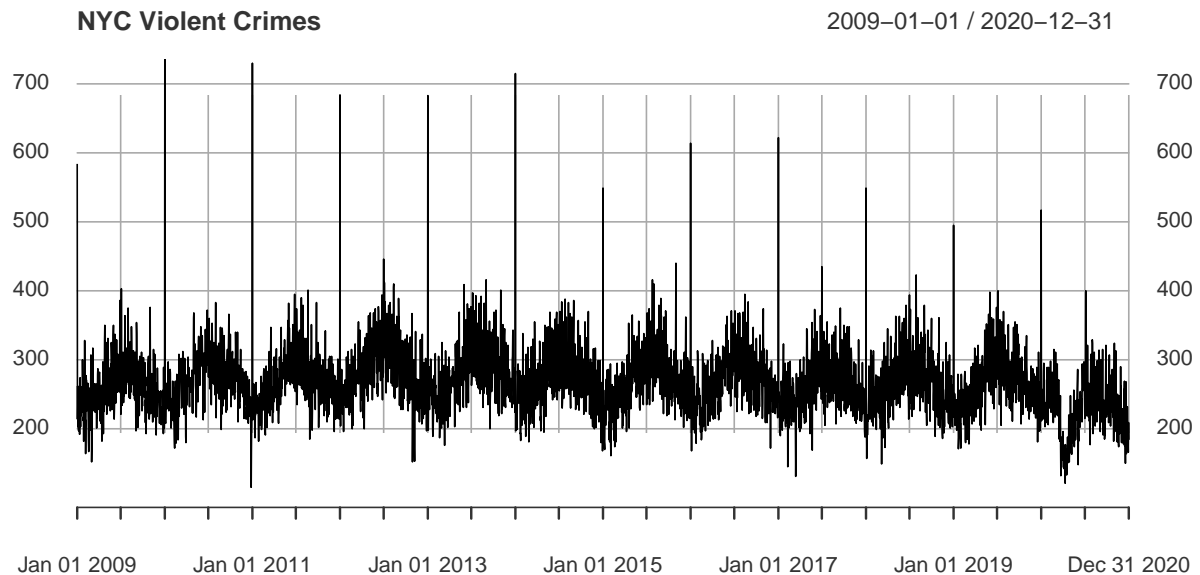
```
## NYC TS
```

```
nyc.v.ts <- ts(nyc.df$violentCrime, start = 2009, freq = 365.25)
```

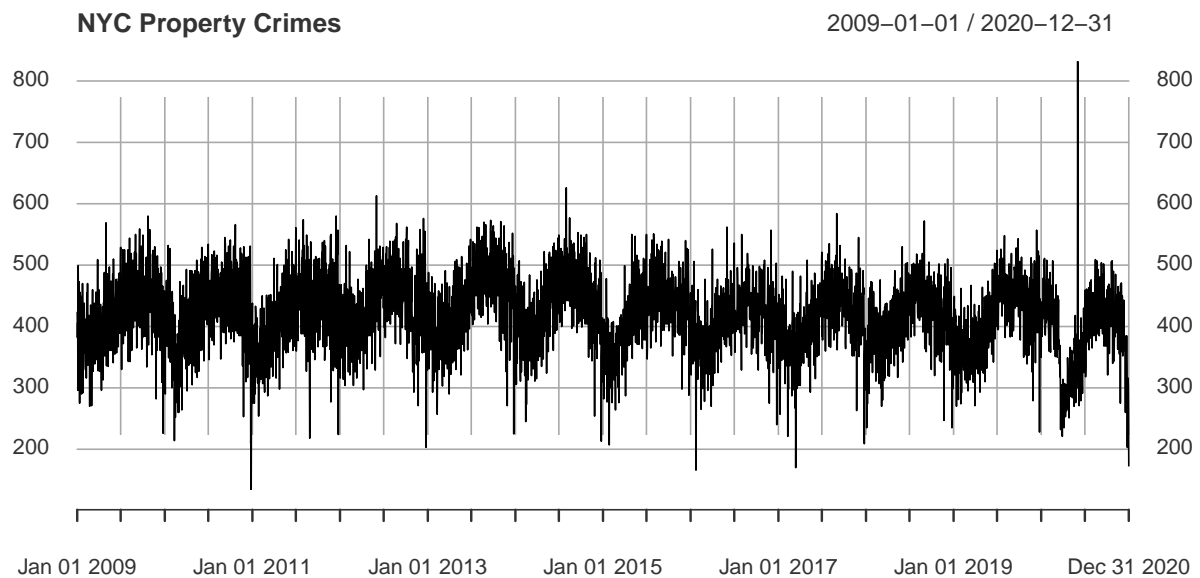
```
nyc.p.ts <- ts(nyc.df$propertyCrime, start = 2009, freq = 365.25)
```

```
nyc.q.ts <- ts(nyc.df$qolCrime, start = 2009, freq = 365.25)
```

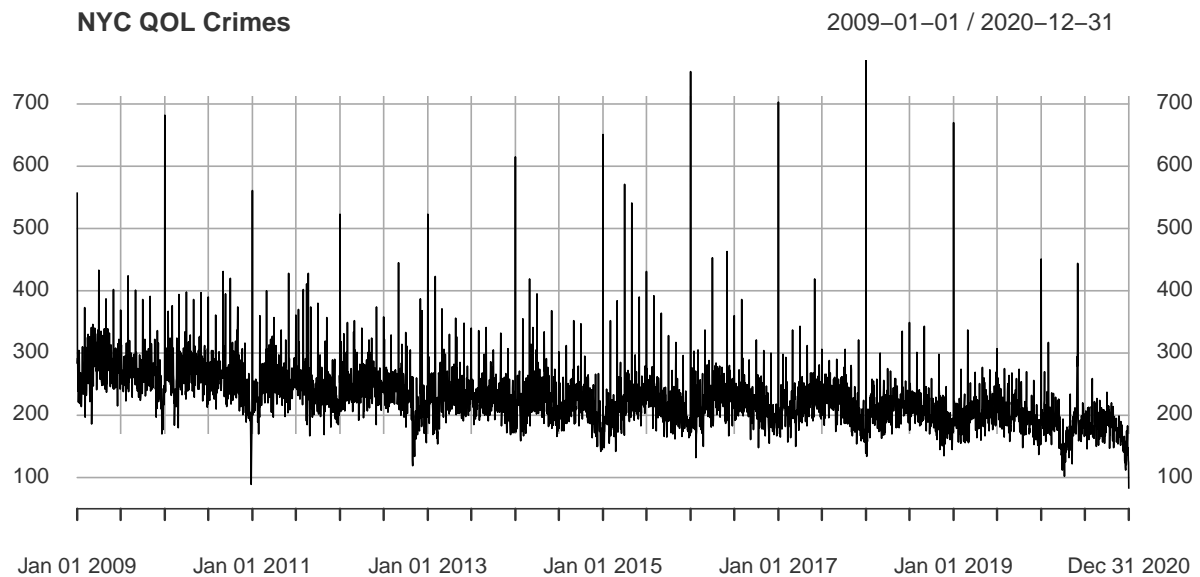
```
plot(xts(nyc.df$violentCrime, nyc.df$Date), main="NYC Violent Crimes", lwd=1)
```



```
plot(xts(nyc.df$propertyCrime, nyc.df$Date), main="NYC Property Crimes", lwd=1)
```



```
plot(xts(nyc.df$qolCrime, nyc.df$Date), main="NYC QOL Crimes", lwd=1)
```



## ARIMA Fitting and Forecasting

Test set = 2022 data (from Jan 1 2022 - end) Training Set = All previous data

### Test Train Split

```
## X-axis points converted to 0-1 scale, common in nonparametric regression
scaler <- function(ts) {
  ts.pts = c(1:length(ts))
  ts.pts = c(ts.pts - min(ts.pts))/max(ts.pts)
  return(ts.pts)
}

train.ind = c(1:which(atl.df$Date == "2020-12-24"))

atl.train <- atl.df[train.ind, ]
atl.test <- atl.df[-train.ind, ]
nyc.train <- nyc.df[train.ind, ]
nyc.test <- nyc.df[-train.ind, ]
```

```
atl.v.train <- ts(atl.train$violentCrime, start = 2009, freq = 365.25)

# Function to train ARIMA (p, d, q) Model
test_modelA <- function(ts, p, d, q) {
  mod = arima(ts, order = c(p, d, q), method = "ML")
  current.aic = AIC(mod)
  df = data.frame(p, d, q, current.aic)
  names(df) <- c("p", "d", "q", "AIC")
}
```

```

# print(paste(p,d,q,current.aic,sep=" "))
return(df)
}

# Daily TS ARIMA (p, d, q) Fitting
atl.v.orders = data.frame(Inf, Inf, Inf, Inf)
names(atl.v.orders) <- c("p", "d", "q", "AIC")

for (p in 0:6) {
  for (d in 1:2) {
    for (q in 0:6) {
      possibleError <- tryCatch(
        atl.v.orders <- rbind(atl.v.orders, test_modelA(atl.v.train,p,d,q)),
        error = function(e) {e}
      )
      if (inherits(possibleError, "error"))
        next
    }
  }
}

atl.v.orders <- atl.v.orders[order(-atl.v.orders$AIC), ]
atl.v.ord <- atl.v.orders[nrow(atl.v.orders), ]
atl.v.orders[(nrow(atl.v.orders)-3):nrow(atl.v.orders), ]

```

#### a) Atlanta Violent Crime

```

##      p d q      AIC
## 64 4 1 6 24141.52
## 77 5 1 5 24140.22
## 78 5 1 6 24139.15
## 63 4 1 5 24138.43

```

```

# (4, 1, 5)

# ARIMA Fitted Model
atl.v.arima = arima(atl.v.train, order = c(atl.v.ord$p, atl.v.ord$d, atl.v.ord$q), method='ML')
atl.v.arima

```

```

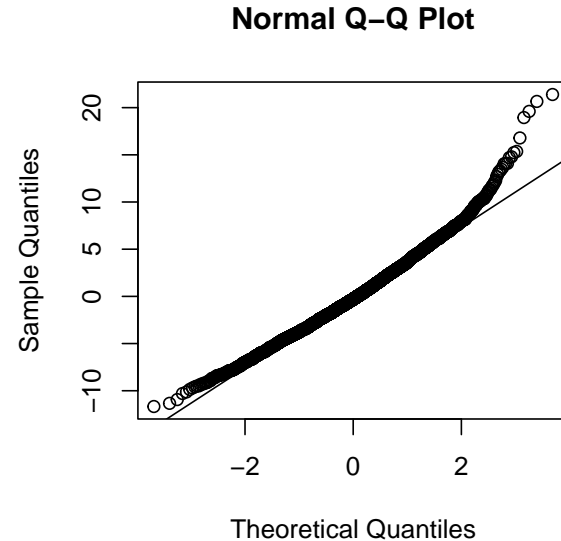
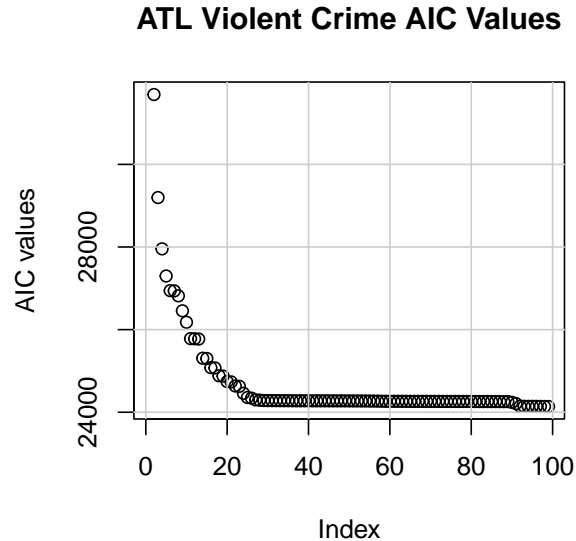
##
## Call:
## arima(x = atl.v.train, order = c(atl.v.ord$p, atl.v.ord$d, atl.v.ord$q), method = "ML")
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ma1      ma2      ma3      ma4
##      -0.4066  0.0706 -0.4172 -0.9895 -0.5383 -0.4581  0.4844  0.5873
## s.e.   0.0077  0.0070  0.0055  0.0063  0.0107  0.0167  0.0138  0.0111
##          ma5
##      -0.9264
## s.e.   0.0106
##
## sigma^2 estimated as 14.5:  log likelihood = -12059.22,  aic = 24138.43

```

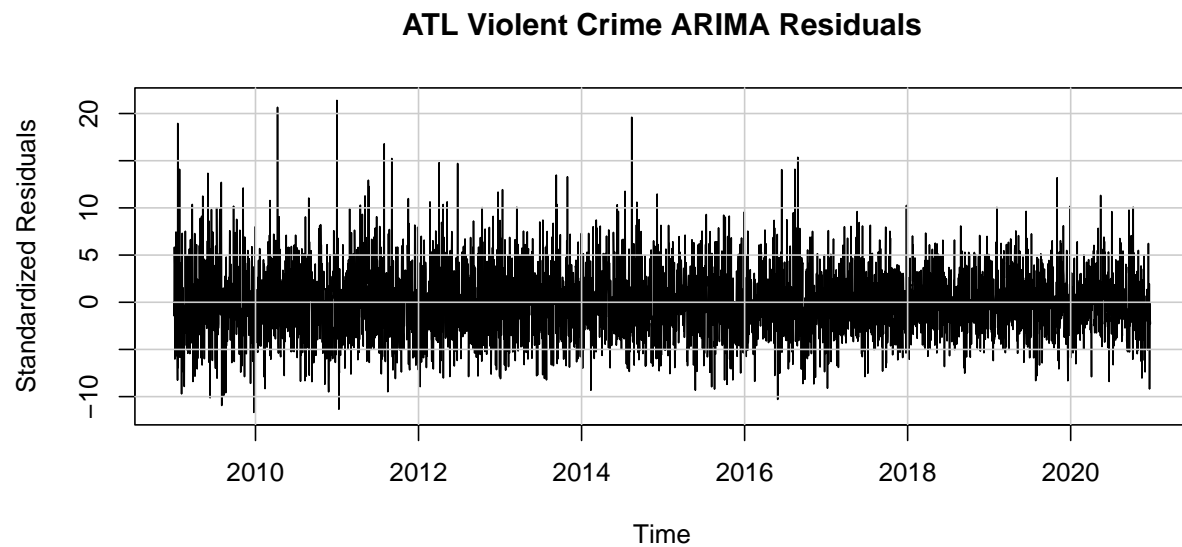
```
coeftest(atl.v.arima)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -0.4066173  0.0076901  -52.876 < 2.2e-16 ***
## ar2  0.0706332  0.0069641   10.143 < 2.2e-16 ***
## ar3 -0.4171808  0.0054937  -75.937 < 2.2e-16 ***
## ar4 -0.9894605  0.0063201 -156.558 < 2.2e-16 ***
## ma1 -0.5383125  0.0107060  -50.282 < 2.2e-16 ***
## ma2 -0.4580922  0.0167166  -27.404 < 2.2e-16 ***
## ma3  0.4844409  0.0137791   35.158 < 2.2e-16 ***
## ma4  0.5872852  0.0111161   52.832 < 2.2e-16 ***
## ma5 -0.9263718  0.0106095  -87.315 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

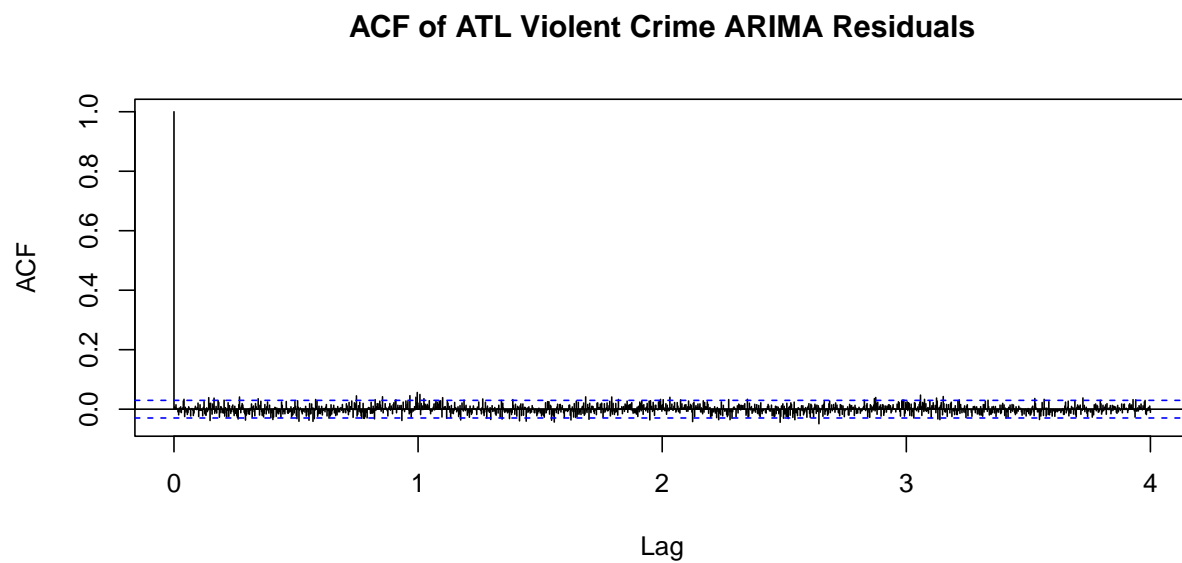
```
par(mfrow=c(1,2))
plot(atl.v.orders$AIC, ylab="AIC values", main="ATL Violent Crime AIC Values")
grid(lty=1, col=gray(0.8))
qqnorm(resid(atl.v.arima))
qqline(resid(atl.v.arima))
```



```
par(mfrow=c(1,1))
plot(residuals(atl.v.arima), ylab='Standardized Residuals', main="ATL Violent Crime ARIMA Residuals")
grid(lty=1, col=gray(0.8))
```



```
acf(residuals(atl.v.arima), lag.max = 365.25*4, main="ACF of ATL Violent Crime ARIMA Residuals")
```



```
atl.p.train <- ts(atl.train$propertyCrime, start = 2009, freq = 365.25)

# Daily TS ARIMA (p, d, q) Fitting
atl.p.orders = data.frame(Inf, Inf, Inf, Inf)
names(atl.p.orders) <- c("p", "d", "q", "AIC")

for (p in 0:6) {
```

```

for (d in 1:2) {
  for (q in 0:6) {
    possibleError <- tryCatch(
      atl.p.orders <- rbind(atl.p.orders, test_modelA(atl.p.train,p,d,q)),
      error = function(e) {e}
    )
    if (inherits(possibleError, "error"))
      next
  }
}

atl.p.orders <- atl.p.orders[order(-atl.p.orders$AIC), ]
atl.p.ord <- atl.p.orders[nrow(atl.p.orders), ]
atl.p.orders[(nrow(atl.p.orders)-3):nrow(atl.p.orders), ]

```

## b) Atlanta Property Crime

```

##      p d q      AIC
## 90 6 1 4 33576.15
## 89 6 1 3 33573.50
## 92 6 1 6 33445.99
## 78 5 1 6 33371.86

```

```
# (5, 1, 6)
```

```
# ARIMA Fitted Model
```

```
atl.p.arima = arima(atl.p.train, order = c(atl.p.ord$p, atl.p.ord$d, atl.p.ord$q), method='ML')
atl.p.arima

```

```

##
## Call:
## arima(x = atl.p.train, order = c(atl.p.ord$p, atl.p.ord$d, atl.p.ord$q), method = "ML")
##
## Coefficients:
##          ar1          ar2          ar3          ar4          ar5          ma1          ma2          ma3
##      -0.1968  -0.6420  -0.6417  -0.1970  -0.9993  -0.6726  0.4640  0.0859
## s.e.   0.0005   0.0005   0.0006   0.0005   0.0005   0.0083  0.0035  0.0054
##          ma4          ma5          ma6
##      -0.3675  0.8238  -0.8593
## s.e.   0.0058  0.0033  0.0084
##
## sigma^2 estimated as 119.4:  log likelihood = -16673.93,  aic = 33371.86

```

```
coeftest(atl.p.arima)
```

```

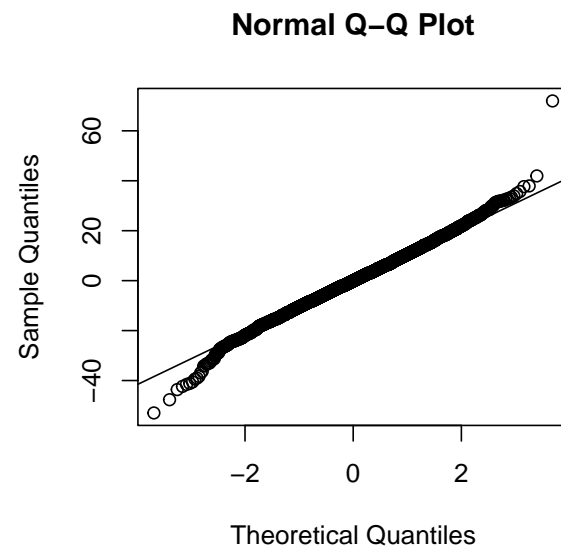
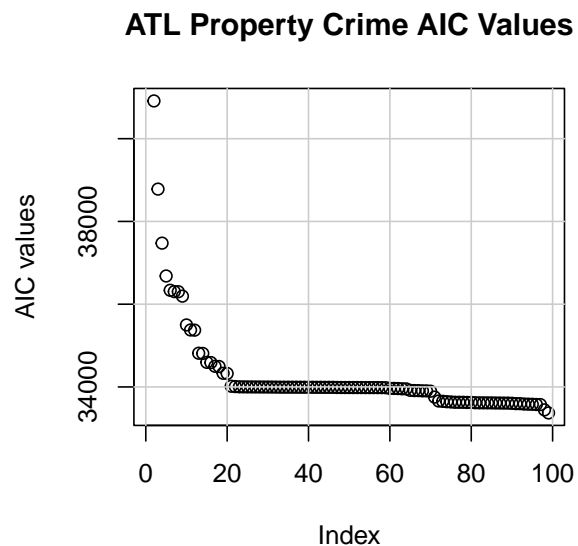
##
## z test of coefficients:
##
##      Estimate Std. Error  z value Pr(>|z|)
## ar1 -0.19684539  0.00048673 -404.422 < 2.2e-16 ***
## ar2 -0.64198218  0.00054440 -1179.242 < 2.2e-16 ***

```

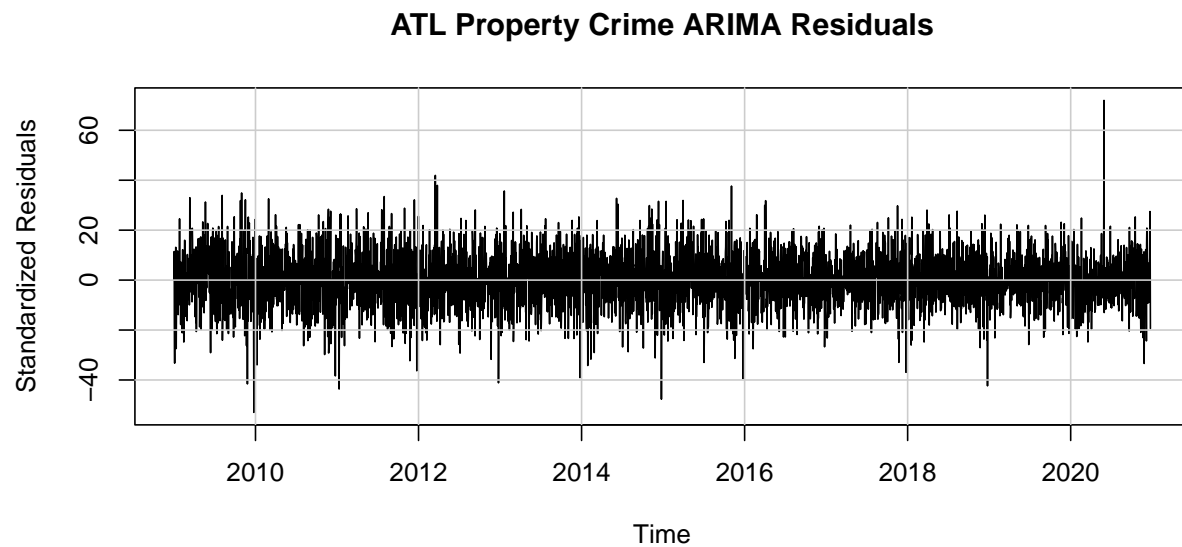


```
## ar3 -0.64166764 0.00060612 -1058.651 < 2.2e-16 ***
## ar4 -0.19701187 0.00047048 -418.748 < 2.2e-16 ***
## ar5 -0.99929418 0.00049669 -2011.892 < 2.2e-16 ***
## ma1 -0.67256340 0.00834158 -80.628 < 2.2e-16 ***
## ma2 0.46398799 0.00347634 133.470 < 2.2e-16 ***
## ma3 0.08585226 0.00542108 15.837 < 2.2e-16 ***
## ma4 -0.36750983 0.00577484 -63.640 < 2.2e-16 ***
## ma5 0.82384339 0.00326033 252.687 < 2.2e-16 ***
## ma6 -0.85933018 0.00839259 -102.391 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

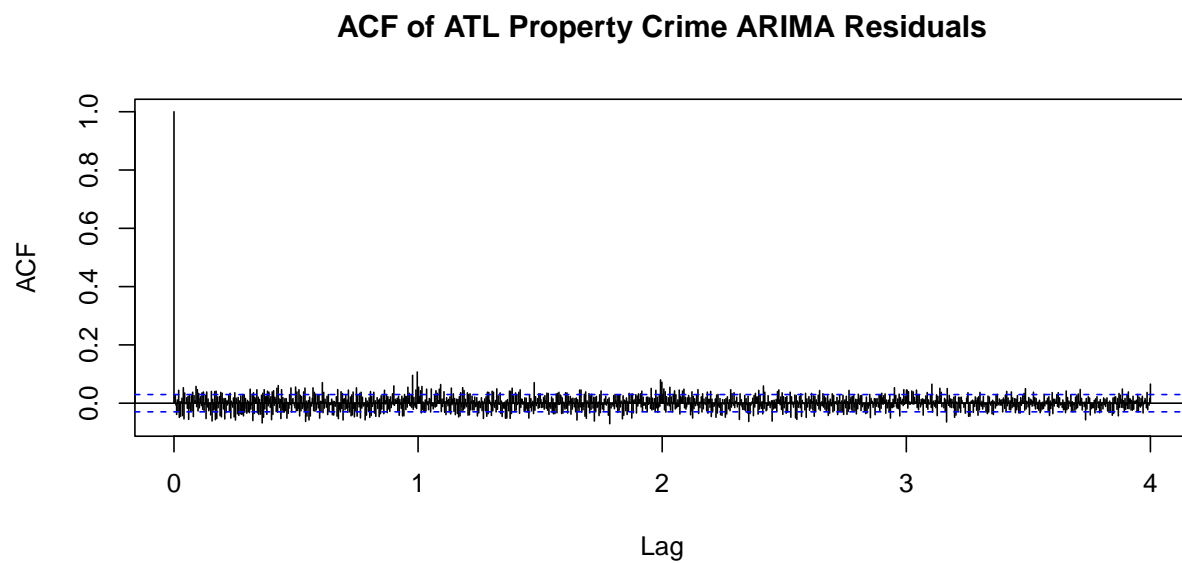
```
par(mfrow=c(1,2))
plot(atl.p.orders$AIC, ylab="AIC values", main="ATL Property Crime AIC Values")
grid(lty=1, col=gray(0.8))
qqnorm(resid(atl.p.arima))
qqline(resid(atl.p.arima))
```



```
par(mfrow=c(1,1))
plot(residuals(atl.p.arima), ylab='Standardized Residuals', main="ATL Property Crime ARIMA Residuals")
grid(lty=1, col=gray(0.8))
```



```
acf(residuals(atl.p.arima), lag.max = 365.25*4, main="ACF of ATL Property Crime ARIMA Residuals")
```



```
nyc.v.train <- ts(nyc.train$violentCrime, start = 2009, freq = 365.25)

# Daily TS ARIMA (p, d, q) Fitting
nyc.v.orders = data.frame(Inf, Inf, Inf, Inf)
names(nyc.v.orders) <- c("p", "d", "q", "AIC")

for (p in 0:6) {
```

```

for (d in 1:2) {
  for (q in 0:6) {
    possibleError <- tryCatch(
      nyc.v.orders <- rbind(nyc.v.orders, test_modelA(nyc.v.train,p,d,q)),
      error = function(e) {e}
    )
    if (inherits(possibleError, "error"))
      next
  }
}

nyc.v.orders <- nyc.v.orders[order(-nyc.v.orders$AIC), ]
nyc.v.ord <- nyc.v.orders[nrow(nyc.v.orders), ]
nyc.v.orders[(nrow(nyc.v.orders)-3):nrow(nyc.v.orders), ]

```

### c) NYC Violent Crime

```

##      p d q      AIC
## 50 3 1 6 44018.68
## 91 6 1 5 44018.09
## 75 5 1 3 44005.77
## 78 5 1 6 43915.95

```

```

# (5, 1, 6)

# ARIMA Fitted Model
nyc.v.arima = arima(nyc.v.train, order = c(nyc.v.ord$p, nyc.v.ord$d, nyc.v.ord$q), method='ML')
nyc.v.arima

```

```

##
## Call:
## arima(x = nyc.v.train, order = c(nyc.v.ord$p, nyc.v.ord$d, nyc.v.ord$q), method = "ML")
##
## Coefficients:
##          ar1          ar2          ar3          ar4          ar5          ma1          ma2          ma3
##      -0.1942   -0.6045   -0.6525   -0.2048   -0.9596   -0.7061    0.3822    0.1495
## s.e.    0.0064    0.0099    0.0027    0.0070    0.0099    0.0159    0.0253    0.0166
##          ma4          ma5          ma6
##      -0.3843    0.726   -0.8270
## s.e.    0.0182    0.024    0.0142
##
## sigma^2 estimated as 1330:  log likelihood = -21945.98,  aic = 43915.95

```

```
coeftest(nyc.v.arima)
```

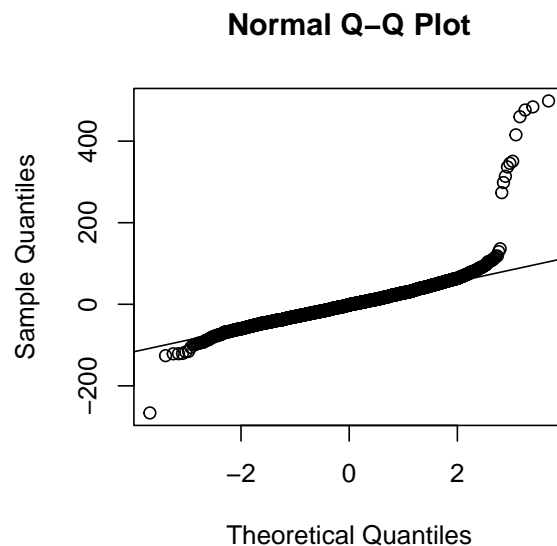
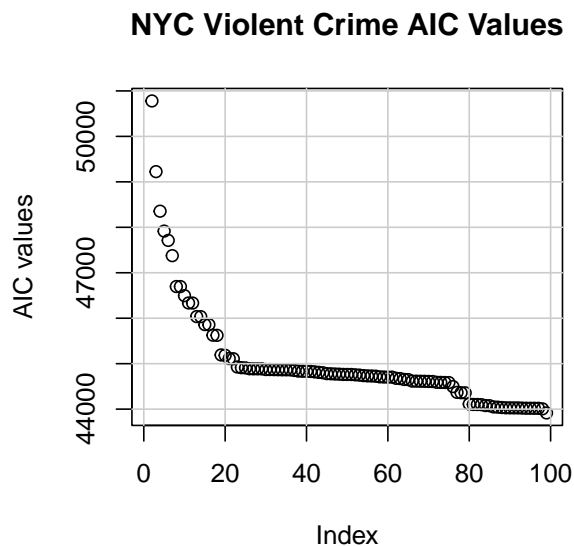
```

##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.1942055  0.0064024 -30.333 < 2.2e-16 ***
## ar2 -0.6045441  0.0099253 -60.910 < 2.2e-16 ***

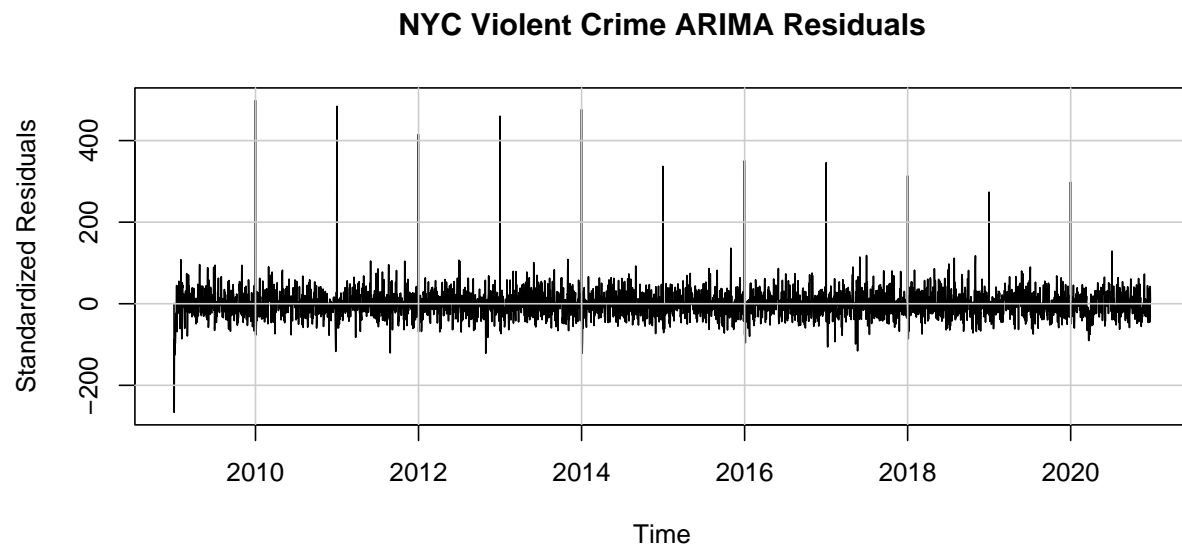
```

```
## ar3 -0.6525132 0.0026580 -245.491 < 2.2e-16 ***
## ar4 -0.2048112 0.0070461 -29.067 < 2.2e-16 ***
## ar5 -0.9595771 0.0098731 -97.191 < 2.2e-16 ***
## ma1 -0.7060965 0.0158833 -44.455 < 2.2e-16 ***
## ma2 0.3822363 0.0253117 15.101 < 2.2e-16 ***
## ma3 0.1494532 0.0165783 9.015 < 2.2e-16 ***
## ma4 -0.3842738 0.0182305 -21.079 < 2.2e-16 ***
## ma5 0.7259711 0.0240376 30.201 < 2.2e-16 ***
## ma6 -0.8270375 0.0142342 -58.102 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

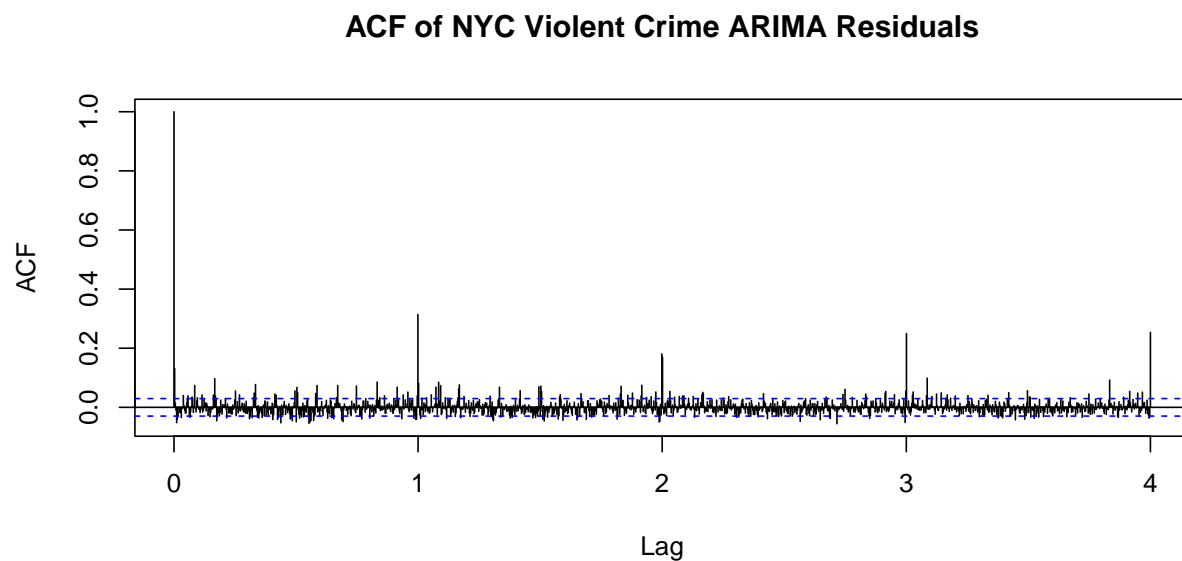
```
par(mfrow=c(1,2))
plot(nyc.v.orders$AIC, ylab="AIC values", main="NYC Violent Crime AIC Values")
grid(lty=1, col=gray(0.8))
qqnorm(resid(nyc.v.arima))
qqline(resid(nyc.v.arima))
```



```
par(mfrow=c(1,1))
plot(residuals(nyc.v.arima), ylab='Standardized Residuals', main="NYC Violent Crime ARIMA Residuals")
grid(lty=1, col=gray(0.8))
```



```
acf(residuals(nyc.v.arima), lag.max = 365.25*4, main="ACF of NYC Violent Crime ARIMA Residuals")
```



```
nyc.p.train <- ts(nyc.train$propertyCrime, start = 2009, freq = 365.25)

# Daily TS ARIMA (p, d, q) Fitting
nyc.p.orders = data.frame(Inf, Inf, Inf, Inf)
names(nyc.p.orders) <- c("p", "d", "q", "AIC")

for (p in 0:6) {
```

```

for (d in 1:2) {
  for (q in 0:6) {
    possibleError <- tryCatch(
      nyc.p.orders <- rbind(nyc.p.orders, test_modelA(nyc.p.train,p,d,q)),
      error = function(e) {e}
    )
    if (inherits(possibleError, "error"))
      next
  }
}
}

nyc.p.orders <- nyc.p.orders[order(-nyc.p.orders$AIC), ]
nyc.p.ord <- nyc.p.orders[nrow(nyc.p.orders), ]
nyc.p.orders[(nrow(nyc.p.orders)-3):nrow(nyc.p.orders), ]

```

#### d) NYC Property Crime

```

##      p d q      AIC
## 64 4 1 6 44637.15
## 92 6 1 6 44631.83
## 63 4 1 5 44630.61
## 77 5 1 5 44628.74

```

```

# (5, 1, 5)

# ARIMA Fitted Model
nyc.p.arima = arima(nyc.p.train, order = c(nyc.p.ord$p, nyc.p.ord$d, nyc.p.ord$q), method='ML')
nyc.p.arima

```

```

##
## Call:
## arima(x = nyc.p.train, order = c(nyc.p.ord$p, nyc.p.ord$d, nyc.p.ord$q), method = "ML")
##
## Coefficients:
##          ar1          ar2          ar3          ar4          ar5          ma1          ma2          ma3
##          0.8422   -1.4773   0.8604   -1.0321   0.0406   -1.6608   2.1228   -2.0412
## s.e.    0.0183    0.0147   0.0265    0.0147   0.0183    0.0100   0.0089   0.0141
##          ma4          ma5
##          1.6793   -0.8591
## s.e.    0.0092    0.0099
##
## sigma^2 estimated as 1563:  log likelihood = -22303.37,  aic = 44628.74

```

```
coeftest(nyc.p.arima)
```

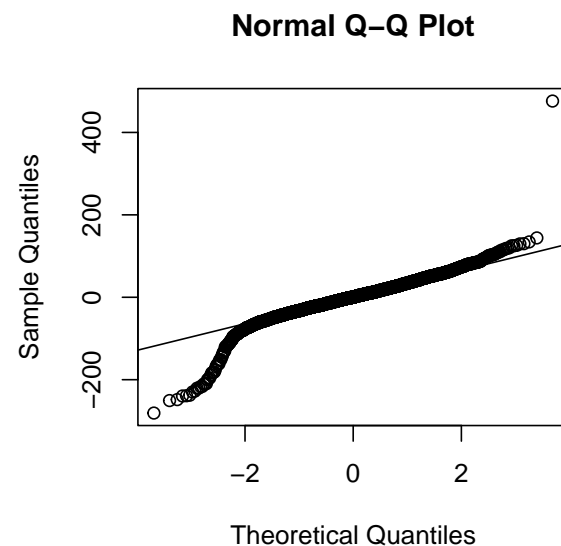
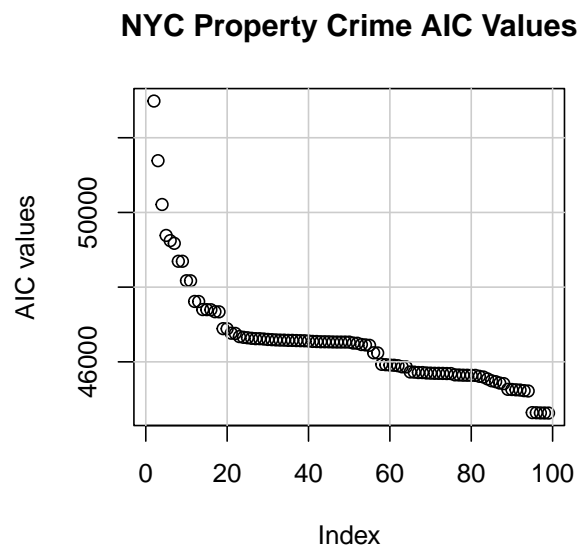
```

##
## z test of coefficients:
##
##      Estimate Std. Error  z value Pr(>|z|)
## ar1  0.8421669  0.0183329  45.9374 < 2e-16 ***
## ar2 -1.4773179  0.0146969 -100.5190 < 2e-16 ***

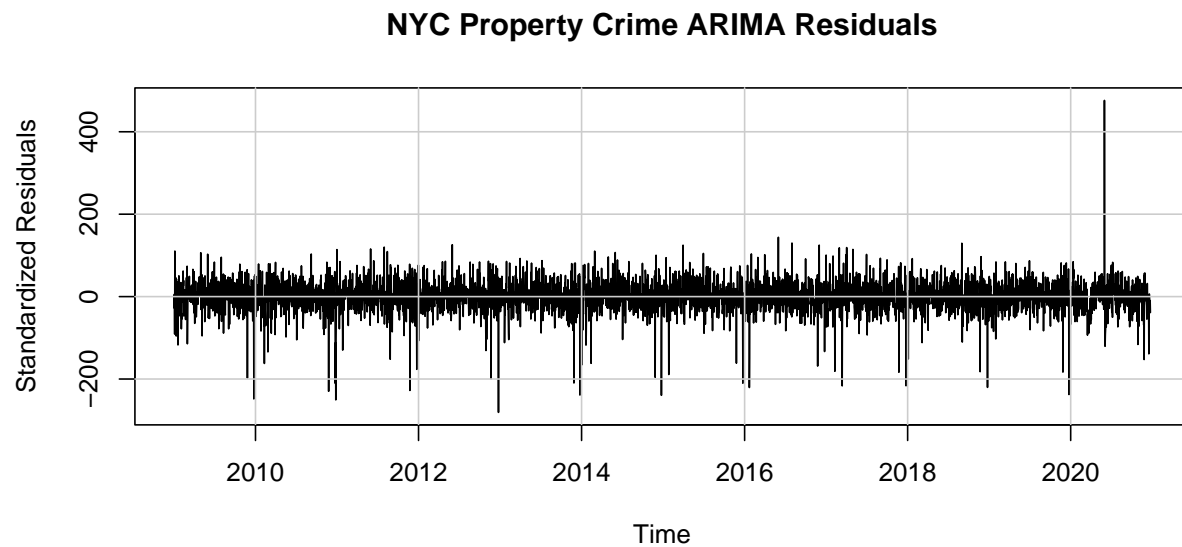
```

```
## ar3  0.8603678  0.0264784   32.4931 < 2e-16 ***
## ar4 -1.0321186  0.0147035  -70.1954 < 2e-16 ***
## ar5  0.0405528  0.0183155   2.2141 0.02682 *
## ma1 -1.6608287  0.0099977 -166.1216 < 2e-16 ***
## ma2  2.1227808  0.0088966  238.6049 < 2e-16 ***
## ma3 -2.0412180  0.0141320 -144.4398 < 2e-16 ***
## ma4  1.6792755  0.0092460  181.6216 < 2e-16 ***
## ma5 -0.8590554  0.0099030  -86.7473 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

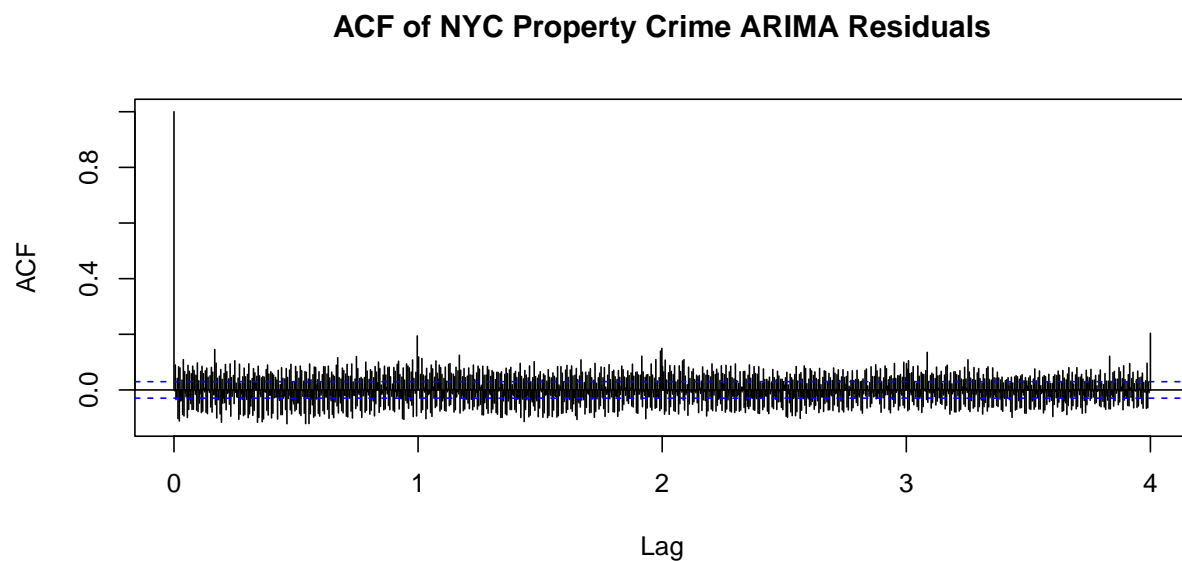
```
par(mfrow=c(1,2))
plot(nyc.p.orders$AIC, ylab="AIC values", main="NYC Property Crime AIC Values")
grid(lty=1, col=gray(0.8))
qqnorm(resid(nyc.p.arima))
qqline(resid(nyc.p.arima))
```



```
par(mfrow=c(1,1))
plot(residuals(nyc.p.arima), ylab='Standardized Residuals', main="NYC Property Crime ARIMA Residuals")
grid(lty=1, col=gray(0.8))
```



```
acf(residuals(nyc.p.arima), lag.max = 365.25*4, main="ACF of NYC Property Crime ARIMA Residuals")
```



```
nyc.q.train <- ts(nyc.train$qolCrime, start = 2009, freq = 365.25)

# Daily TS ARIMA (p, d, q) Fitting
nyc.q.orders = data.frame(Inf, Inf, Inf, Inf)
names(nyc.q.orders) <- c("p", "d", "q", "AIC")

for (p in 0:6) {
```



```

for (d in 1:2) {
  for (q in 0:6) {
    possibleError <- tryCatch(
      nyc.q.orders <- rbind(nyc.q.orders, test_modelA(nyc.q.train,p,d,q)),
      error = function(e) {e}
    )
    if (inherits(possibleError, "error"))
      next
  }
}

nyc.q.orders <- nyc.q.orders[order(-nyc.q.orders$AIC), ]
nyc.q.ord <- nyc.q.orders[nrow(nyc.q.orders), ]
nyc.q.orders[(nrow(nyc.q.orders)-3):nrow(nyc.q.orders), ]

```

#### e) NYC QOL Crime

```

##      p d q      AIC
## 63 4 1 5 44620.36
## 78 5 1 6 44610.26
## 91 6 1 5 44607.88
## 92 6 1 6 44584.47

```

```
# (6, 1, 6)
```

```
# ARIMA Fitted Model
```

```
nyc.q.arima = arima(nyc.q.train, order = c(nyc.q.ord$p, nyc.q.ord$d, nyc.q.ord$q), method='ML')
nyc.q.arima
```

```

##
## Call:
## arima(x = nyc.q.train, order = c(nyc.q.ord$p, nyc.q.ord$d, nyc.q.ord$q), method = "ML")
##
## Coefficients:
##          ar1          ar2          ar3          ar4          ar5          ar6          ma1          ma2
##        -1.0642   -0.1357   -0.3707   -1.2558   -0.6258    0.0405    0.1756   -0.8648
## s.e.    0.1238    0.0407    0.0212    0.0593    0.1125    0.0232    0.1225    0.0730
##          ma3          ma4          ma5          ma6
##         0.1547    0.9200   -0.4958   -0.6871
## s.e.    0.0543    0.0657    0.0664    0.1139
##
## sigma^2 estimated as 1550:  log likelihood = -22279.23,  aic = 44584.47

```

```
coeftest(nyc.q.arima)
```

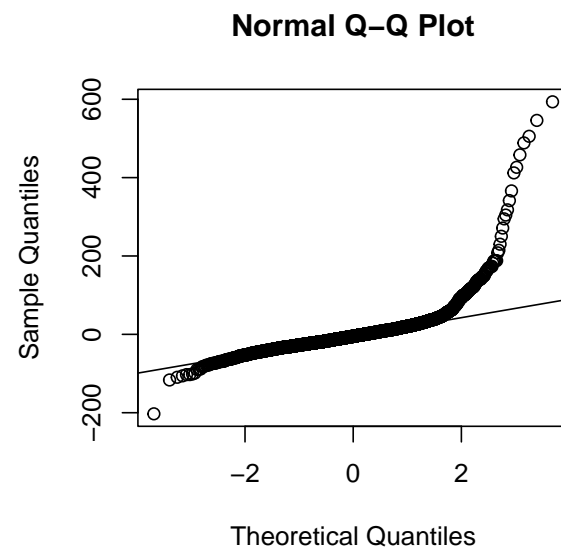
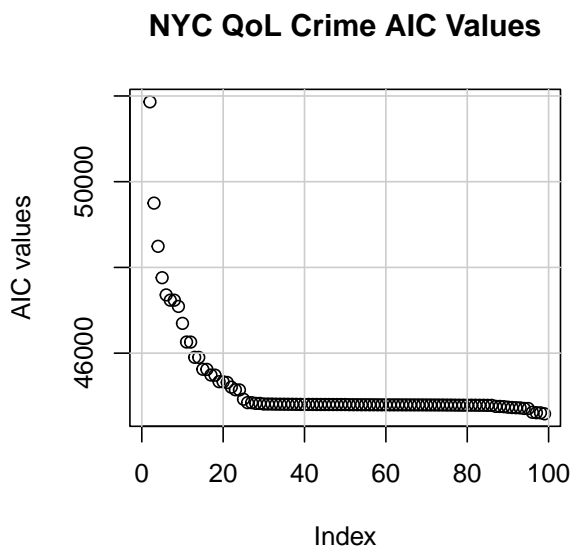
```

##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -1.064228    0.123813  -8.5955 < 2.2e-16 ***
## ar2 -0.135716    0.040739  -3.3314 0.0008642 ***

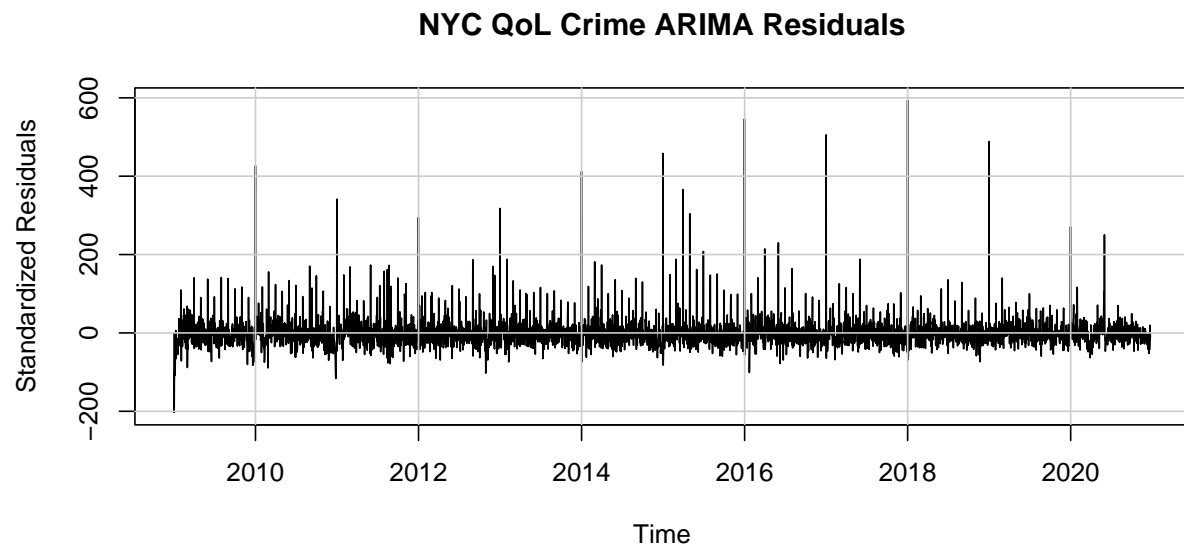
```

```
## ar3 -0.370684    0.021164 -17.5146 < 2.2e-16 ***
## ar4 -1.255817    0.059349 -21.1597 < 2.2e-16 ***
## ar5 -0.625842    0.112452  -5.5654 2.615e-08 ***
## ar6  0.040486    0.023177   1.7468 0.0806696 .
## ma1  0.175558    0.122508   1.4330 0.1518473
## ma2 -0.864804    0.073014 -11.8443 < 2.2e-16 ***
## ma3  0.154691    0.054317   2.8479 0.0044003 **
## ma4  0.920008    0.065745  13.9937 < 2.2e-16 ***
## ma5 -0.495806    0.066438  -7.4627 8.479e-14 ***
## ma6 -0.687102    0.113908  -6.0321 1.619e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

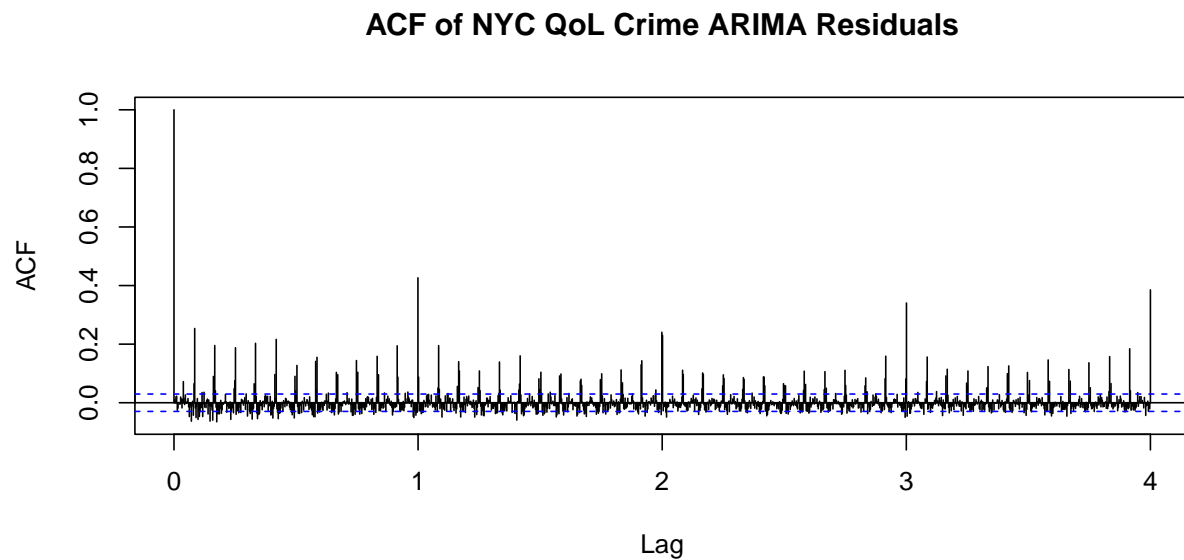
```
par(mfrow=c(1,2))
plot(nyc.q.orders$AIC, ylab="AIC values", main="NYC QoL Crime AIC Values")
grid(lty=1, col=gray(0.8))
qqnorm(resid(nyc.q.arima))
qqline(resid(nyc.q.arima))
```



```
par(mfrow=c(1,1))
plot(residuals(nyc.q.arima), ylab='Standardized Residuals', main="NYC QoL Crime ARIMA Residuals")
grid(lty=1, col=gray(0.8))
```



```
acf(residuals(nyc.q.arima), lag.max = 365.25*4, main="ACF of NYC QoL Crime ARIMA Residuals")
```



## Residual Analysis

```
## Test for Uncorrelated Residuals for the final model

# ATL
Box.test(atl.v.arima$resid, lag = (atl.v.ord$p+atl.v.ord$q+1),
        type = "Ljung-Box", fitdf = (atl.v.ord$p+atl.v.ord$q))

##
```

```
## Box-Ljung test
##
## data:  atl.v.arima$resid
## X-squared = 4.0175, df = 1, p-value = 0.04503
```

```
Box.test(atl.p.arima$resid, lag = (atl.p.ord$p+atl.p.ord$q+1),
         type = "Ljung-Box", fitdf = (atl.p.ord$p+atl.p.ord$q))
```

```
##
## Box-Ljung test
##
## data:  atl.p.arima$resid
## X-squared = 46.327, df = 1, p-value = 1.001e-11
```

```
# NYC
Box.test(nyc.v.arima$resid, lag = (nyc.v.ord$p+nyc.v.ord$q+1),
         type = "Ljung-Box", fitdf = (nyc.v.ord$p+nyc.v.ord$q))
```

```
##
## Box-Ljung test
##
## data:  nyc.v.arima$resid
## X-squared = 107.9, df = 1, p-value < 2.2e-16
```

```
Box.test(nyc.p.arima$resid, lag = (nyc.p.ord$p+nyc.p.ord$q+1),
         type = "Ljung-Box", fitdf = (nyc.p.ord$p+nyc.p.ord$q))
```

```
##
## Box-Ljung test
##
## data:  nyc.p.arima$resid
## X-squared = 222.03, df = 1, p-value < 2.2e-16
```

```
Box.test(nyc.q.arima$resid, lag = (nyc.q.ord$p+nyc.q.ord$q+1),
         type = "Ljung-Box", fitdf = (nyc.q.ord$p+nyc.q.ord$q))
```

```
##
## Box-Ljung test
##
## data:  nyc.q.arima$resid
## X-squared = 13.413, df = 1, p-value = 0.0002499
```

## Forecast

```
plot_forecast <- function(ts, out_pred, days_ahead, plot_title, conf) {
  n = length(ts)
  nfit = n-days_ahead

  timevol=time(ts)
```

```

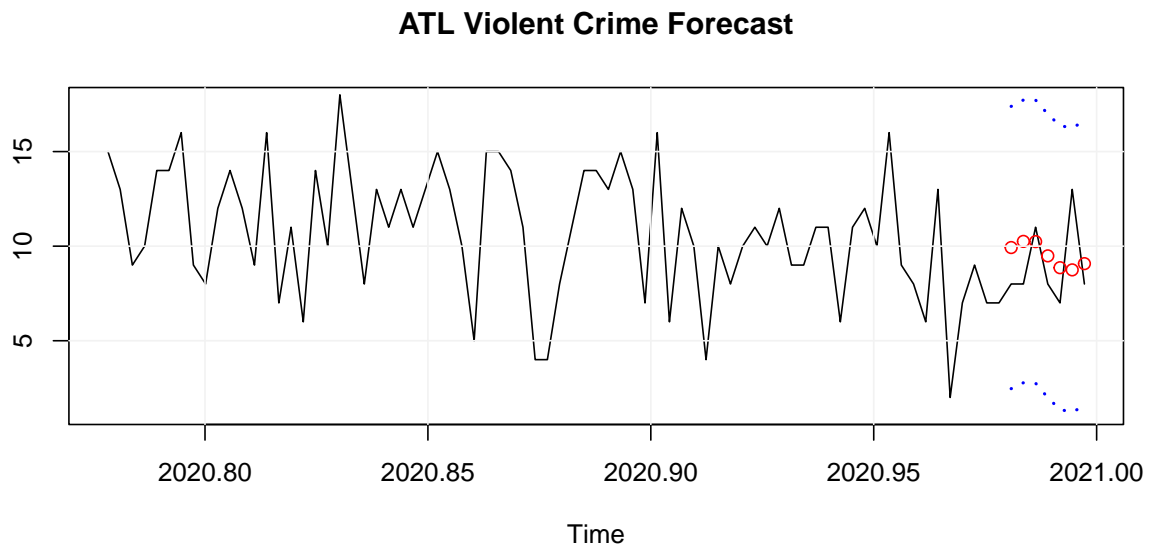
ubound = out_pred$pred+conf*out_pred$se
lbound = out_pred$pred-conf*out_pred$se
ymin = min(lbound, min(out_pred$pred))
ymax = max(ubound, max(out_pred$pred))

par(mfrow=c(1,1))
plot(timevol[(n-80):n],ts[(n-80):n],type="l", ylim=c(ymin, ymax), xlab="Time",
      ylab="", main=plot_title)
points(timevol[(nfit+1):n],out_pred$pred,col="red")
lines(timevol[(nfit+1):n],ubound,lty=3,lwd= 2, col="blue")
lines(timevol[(nfit+1):n],lbound,lty=3,lwd= 2, col="blue")
}

n.ahead <- nrow(atl.test)

## Forecast ATL Violent Crime
atl.v.pred <- as.vector(predict(atl.v.arima, n.ahead=n.ahead))
plot_forecast(atl.v.ts, atl.v.pred, n.ahead, conf=1.96, plot_title = "ATL Violent Crime Forecast")
grid(lty=1, col=gray(0.95))

```

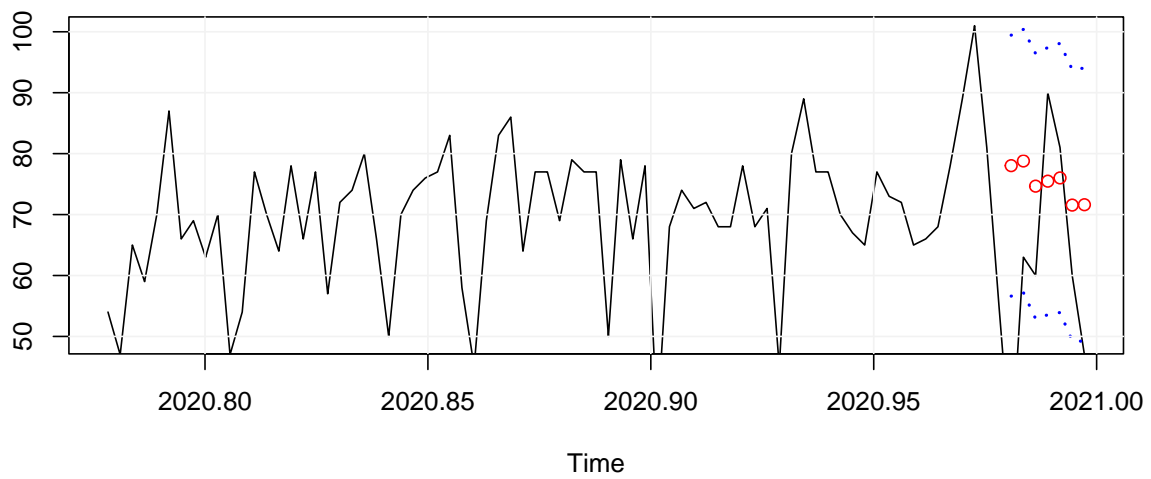


```

## Forecast ATL Property Crime
atl.p.pred <- as.vector(predict(atl.p.arima, n.ahead=n.ahead))
plot_forecast(atl.p.ts, atl.p.pred, n.ahead, conf=1.96, plot_title = "ATL Property Crime Forecast")
grid(lty=1, col=gray(0.95))

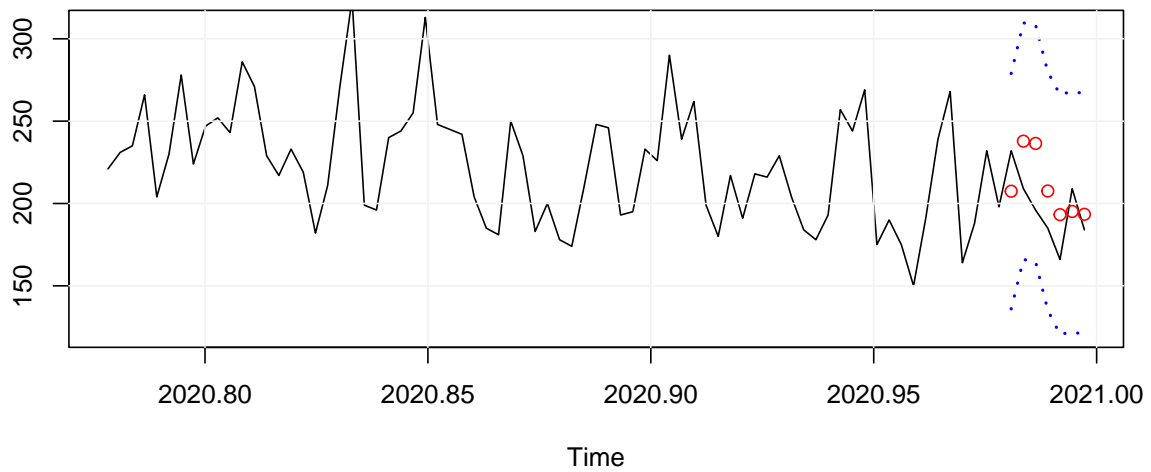
```

### ATL Property Crime Forecast



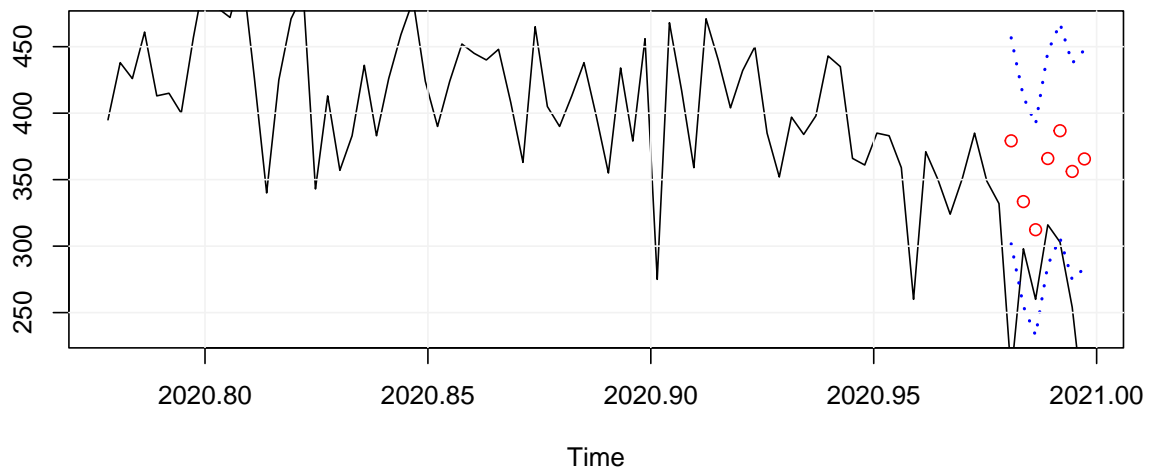
```
## Forecast NYC Violent Crime
nyc.v.pred <- as.vector(predict(nyc.v.arima, n.ahead=n.ahead))
plot_forecast(nyc.v.ts, nyc.v.pred, n.ahead, conf=1.96, plot_title = "NYC Violent Crime Forecast")
grid(lty=1, col=gray(0.95))
```

### NYC Violent Crime Forecast



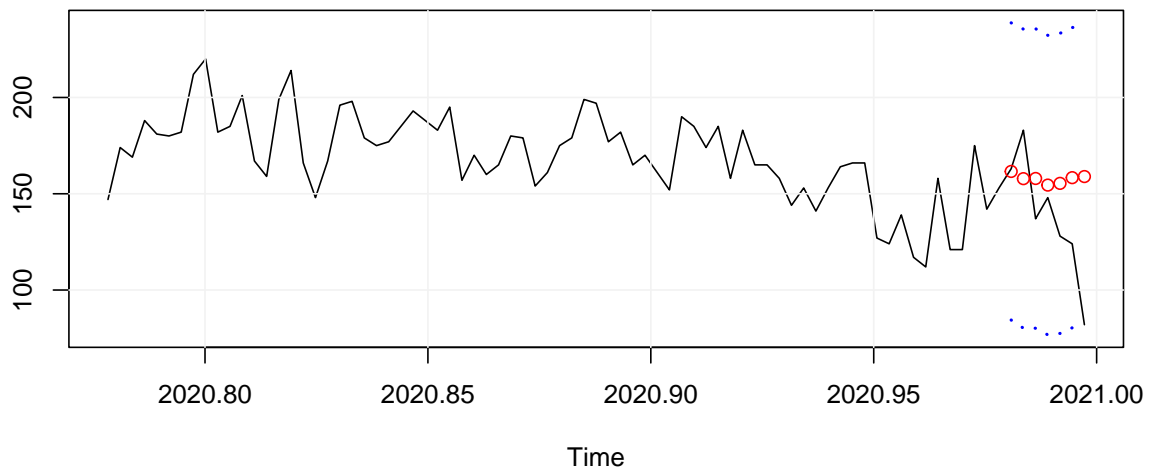
```
## Forecast NYC Property Crime
nyc.p.pred <- as.vector(predict(nyc.p.arima, n.ahead=n.ahead))
plot_forecast(nyc.p.ts, nyc.p.pred, n.ahead, conf=1.96, plot_title = "NYC Property Crime Forecast")
grid(lty=1, col=gray(0.95))
```

## NYC Property Crime Forecast



```
## Forecast NYC QoL Crime
nyc.q.pred <- as.vector(predict(nyc.q.arima, n.ahead=n.ahead))
plot_forecast(nyc.q.ts, nyc.q.pred, n.ahead, conf=1.96, plot_title = "NYC QoL Crime Forecast")
grid(lty=1, col=gray(0.95))
```

## NYC QoL Crime Forecast



## Prediction Evaluation

```
mape <- function(y, y_pred) {
  mape <- mean(abs((y-y_pred)/y))
  return(mape)
```

```

}

pm <- function(obs, pred) {
  pm <- sum((pred-obs)^2)/sum((obs-mean(obs))^2)
  return(pm)
}

atl.v.mape <- mape(atl.test$violentCrime, atl.v.pred$pred)
atl.p.mape <- mape(atl.test$propertyCrime, atl.p.pred$pred)
atl.v.pm <- pm(atl.test$violentCrime, atl.v.pred$pred)
atl.p.pm <- pm(atl.test$propertyCrime, atl.p.pred$pred)

nyc.v.mape <- mape(nyc.test$violentCrime, nyc.v.pred$pred)
nyc.p.mape <- mape(nyc.test$propertyCrime, nyc.p.pred$pred)
nyc.q.mape <- mape(nyc.test$qolCrime, nyc.q.pred$pred)
nyc.v.pm <- pm(nyc.test$violentCrime, nyc.v.pred$pred)
nyc.p.pm <- pm(nyc.test$propertyCrime, nyc.p.pred$pred)
nyc.q.pm <- pm(nyc.test$qolCrime, nyc.q.pred$pred)

cat("ATL Violent:\nMAPE =", atl.v.mape, "\nPM =", atl.v.pm,
    "\n\nATL Property:\nMAPE =", atl.p.mape, "\nPM =", atl.p.pm,
    "\n\nNYC Violent:\nMAPE =", nyc.v.mape, "\nPM =", nyc.v.pm,
    "\n\nNYC Property:\nMAPE =", atl.p.mape, "\nPM =", atl.p.pm,
    "\n\nNYC QoL:\nMAPE =", nyc.q.mape, "\nPM =", nyc.q.pm)

```

```

## ATL Violent:
## MAPE = 0.2149906
## PM = 1.224809
##
## ATL Property:
## MAPE = 0.433607
## PM = 1.559845
##
## NYC Violent:
## MAPE = 0.1219765
## PM = 1.651684
##
## NYC Property:
## MAPE = 0.433607
## PM = 1.559845
##
## NYC QoL:
## MAPE = 0.253166
## PM = 1.449972

```

## SARIMA Fitting and Forecasting

```

atl.v.sarima = arima(atl.v.train, order = c(4,1,5),seasonal = list(order =c(1,0,1), period=30), method=
atl.v.sarima

```

### a) ATL Violent Crime

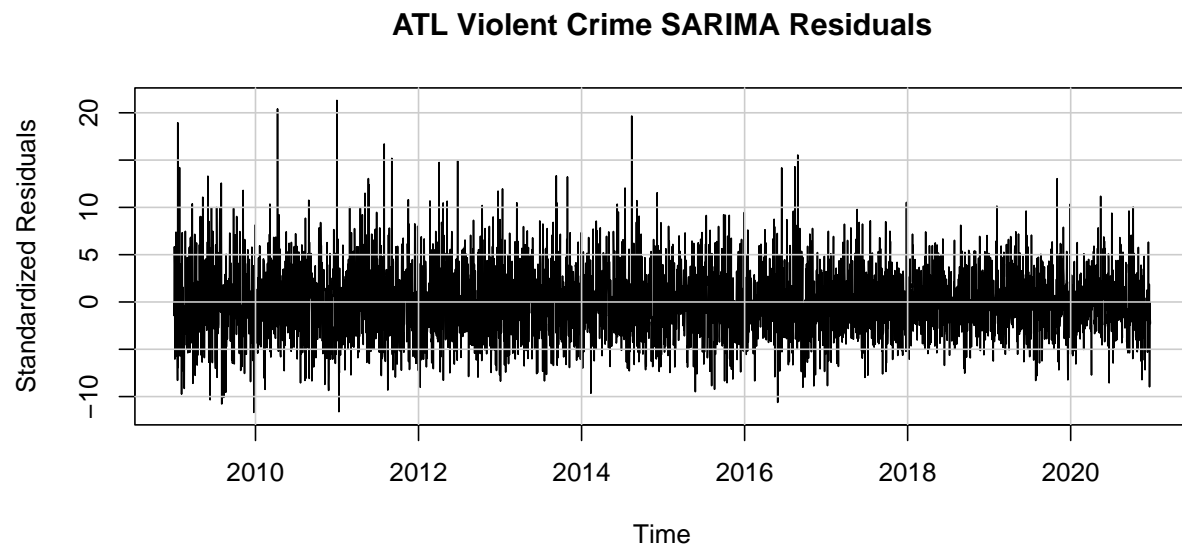


```
##
## Call:
## arima(x = atl.v.train, order = c(4, 1, 5), seasonal = list(order = c(1, 0, 1),
##     period = 30), method = "ML")
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ma1      ma2      ma3      ma4
##      -0.3676  0.0162 -0.3712 -0.9961 -0.5776 -0.3636  0.3831  0.6423
## s.e.   0.0031  0.0033  0.0032  0.0041  0.0063  0.0069  0.0074  0.0092
##          ma5      sar1      sma1
##      -0.9356 -0.2662  0.246
## s.e.   0.0089      NaN      NaN
##
## sigma^2 estimated as 14.5:  log likelihood = -12059.09,  aic = 24142.17
```

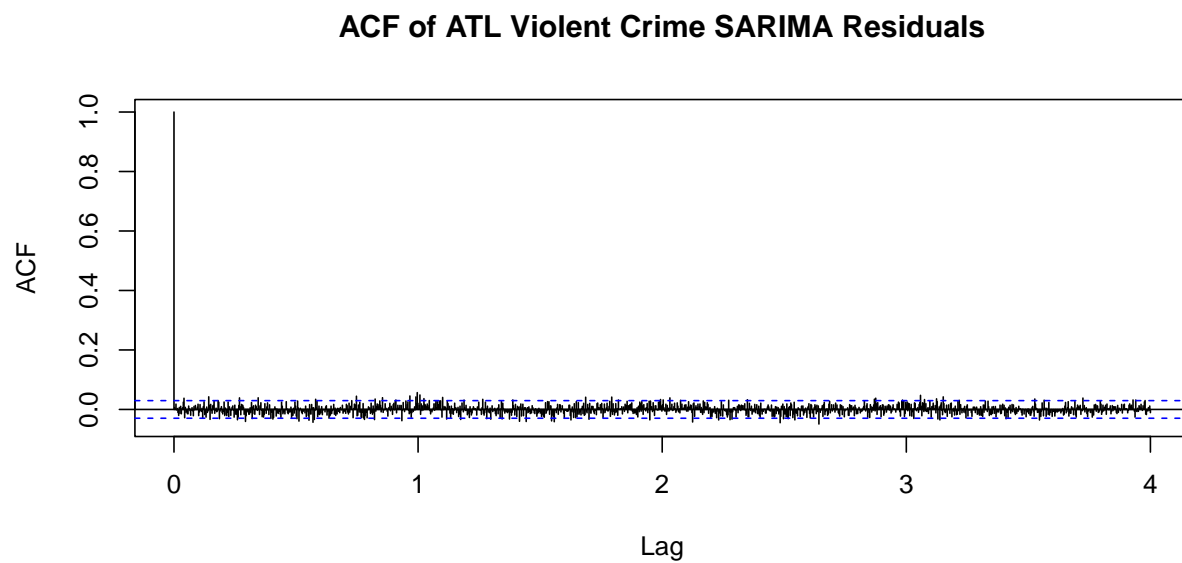
```
coeftest(atl.v.sarima)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -0.3676457  0.0030797 -119.3757 < 2.2e-16 ***
## ar2  0.0161737  0.0033271   4.8611 1.167e-06 ***
## ar3 -0.3711793  0.0032169 -115.3856 < 2.2e-16 ***
## ar4 -0.9960878  0.0041013 -242.8708 < 2.2e-16 ***
## ma1 -0.5775813  0.0063188  -91.4062 < 2.2e-16 ***
## ma2 -0.3635749  0.0068935  -52.7417 < 2.2e-16 ***
## ma3  0.3830643  0.0073768   51.9285 < 2.2e-16 ***
## ma4  0.6423401  0.0091529   70.1791 < 2.2e-16 ***
## ma5 -0.9355964  0.0089466 -104.5754 < 2.2e-16 ***
## sar1 -0.2661653      NaN      NaN      NaN
## sma1  0.2459751      NaN      NaN      NaN
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow=c(1,1))
plot(residuals(atl.v.sarima), ylab='Standardized Residuals', main="ATL Violent Crime SARIMA Residuals")
grid(lty=1, col=gray(0.8))
```



```
acf(residuals(atl.v.sarima), lag.max = 365.25*4, main="ACF of ATL Violent Crime SARIMA Residuals")
```



```
atl.p.sarima = arima(atl.p.train, order = c(5,1,6),seasonal = list(order =c(1,0,1), period=30), method=
atl.p.sarima
```

#### b) ATL Property Crime

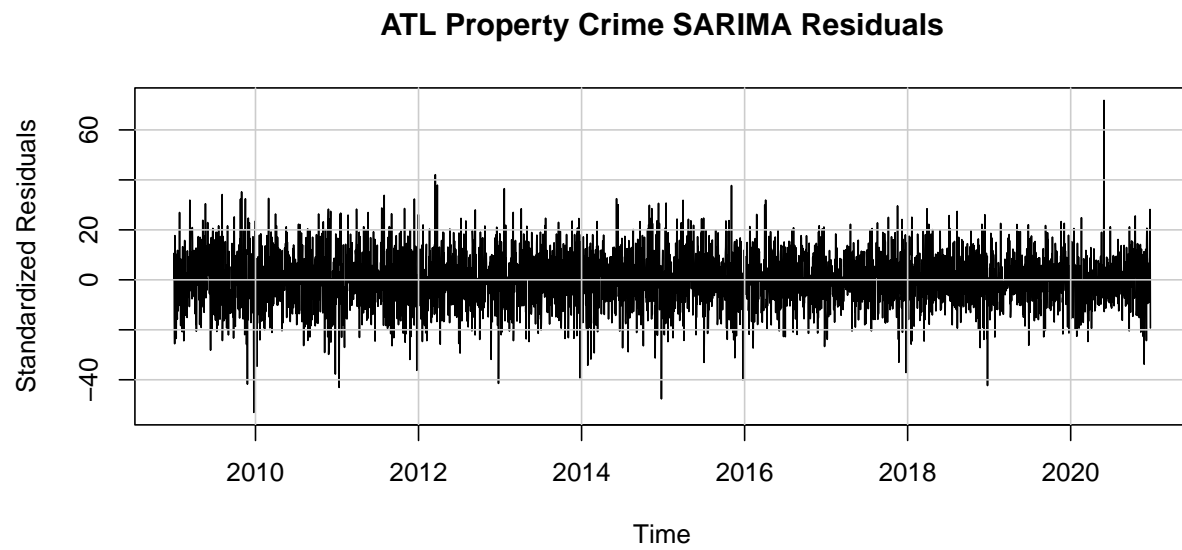
```
##
## Call:
```

```
## arima(x = atl.p.train, order = c(5, 1, 6), seasonal = list(order = c(1, 0, 1),
##      period = 30), method = "ML")
##
## Coefficients:
##      ar1      ar2      ar3      ar4      ar5      ma1      ma2      ma3
##    -0.1970 -0.6420 -0.6420 -0.1971 -0.9997 -0.6728  0.4641  0.0857
## s.e.   0.0005   0.0005   0.0005   0.0005   0.0002   0.0083   0.0033   0.0054
##      ma4      ma5      ma6      sar1      sma1
##    -0.3669  0.8237 -0.8606 -0.0056  0.0317
## s.e.   0.0057  0.0031   0.0083   0.2988  0.3043
##
## sigma^2 estimated as 119.3:  log likelihood = -16673.45,  aic = 33374.91
```

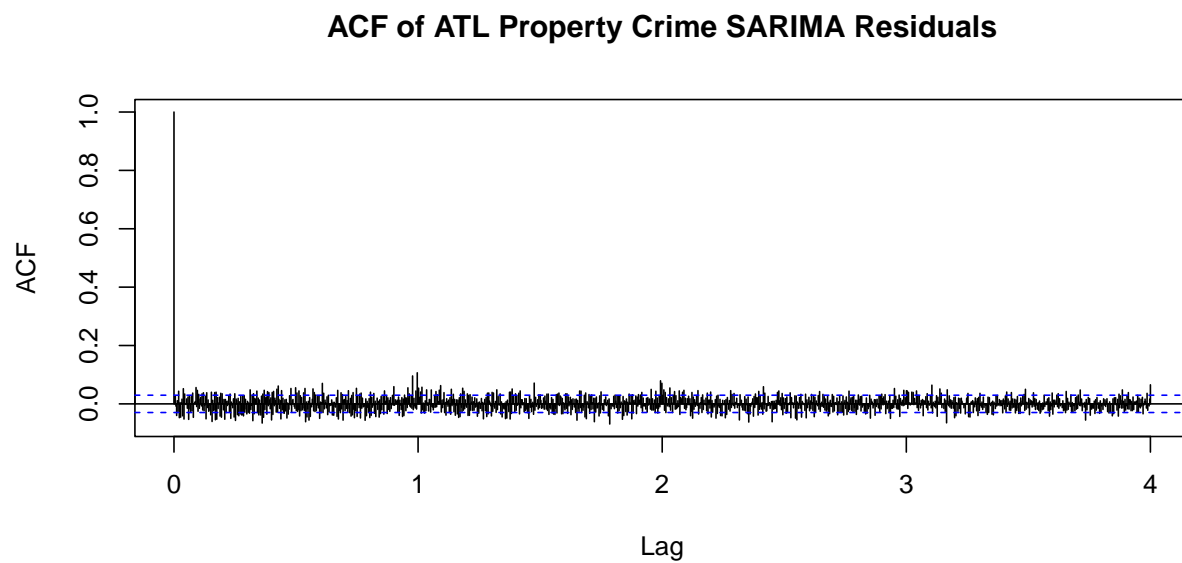
```
coeftest(atl.p.sarima)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error   z value Pr(>|z|)
## ar1 -0.19695669  0.00047301  -416.3883  <2e-16 ***
## ar2 -0.64195460  0.00048821 -1314.9212  <2e-16 ***
## ar3 -0.64199058  0.00048693 -1318.4534  <2e-16 ***
## ar4 -0.19708564  0.00046506  -423.7871  <2e-16 ***
## ar5 -0.99970588  0.00023530 -4248.5803  <2e-16 ***
## ma1 -0.67282557  0.00834266  -80.6489   <2e-16 ***
## ma2  0.46407711  0.00334660   138.6712  <2e-16 ***
## ma3  0.08573301  0.00544659   15.7407   <2e-16 ***
## ma4 -0.36686116  0.00570193  -64.3398   <2e-16 ***
## ma5  0.82368380  0.00306074   269.1130  <2e-16 ***
## ma6 -0.86060172  0.00826927 -104.0723  <2e-16 ***
## sar1 -0.00562442  0.29876478   -0.0188    0.985
## sma1  0.03170224  0.30429015    0.1042    0.917
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow=c(1,1))
plot(residuals(atl.p.sarima), ylab='Standardized Residuals', main="ATL Property Crime SARIMA Residuals")
grid(lty=1, col=gray(0.8))
```



```
acf(residuals(atl.p.sarima), lag.max = 365.25*4, main="ACF of ATL Property Crime SARIMA Residuals")
```



```
nyc.v.sarima = arima(nyc.v.train, order = c(5,1,6),seasonal = list(order =c(1,0,1), period=30), method=
nyc.v.sarima
```

c) NYC Violent Crime

```
##
## Call:
```

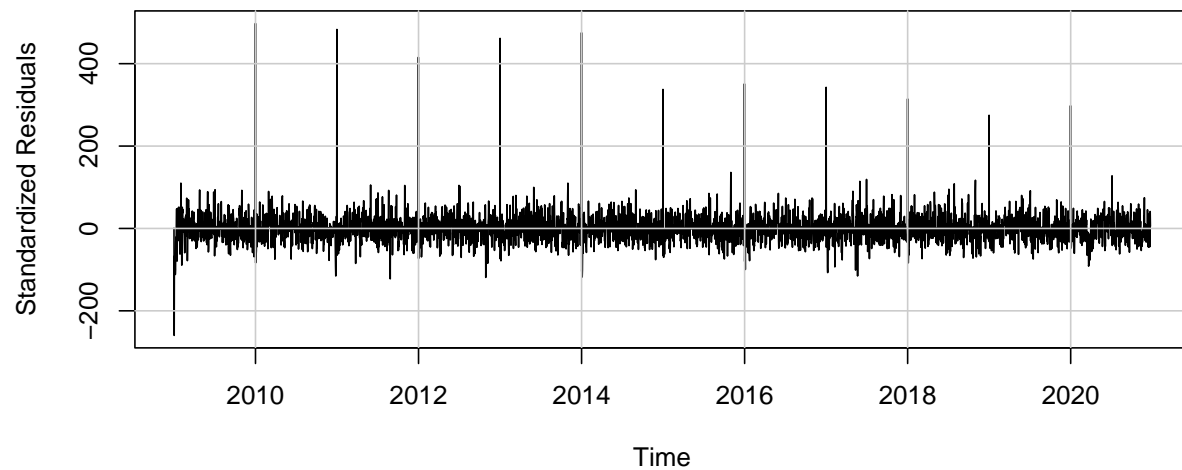
```
## arima(x = nyc.v.train, order = c(5, 1, 6), seasonal = list(order = c(1, 0, 1),
##      period = 30), method = "ML")
##
## Coefficients:
##      ar1      ar2      ar3      ar4      ar5      ma1      ma2      ma3
##    -0.1947 -0.6117 -0.6512 -0.203 -0.9675 -0.6954  0.3782  0.1436
## s.e.   0.0052   0.0150   0.0040   0.006   0.0154   0.0144   0.0347   0.0260
##      ma4      ma5      ma6      sar1      sma1
##    -0.3706  0.7248 -0.8316 -0.3152  0.3137
## s.e.   0.0168  0.0371   0.0220      NaN      NaN
##
## sigma^2 estimated as 1325:  log likelihood = -21939.2,  aic = 43906.4
```

```
coeftest(nyc.v.sarima)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error   z value Pr(>|z|)
## ar1 -0.1947074  0.0052324 -37.2121 < 2.2e-16 ***
## ar2 -0.6116805  0.0150202 -40.7238 < 2.2e-16 ***
## ar3 -0.6512457  0.0040175 -162.1006 < 2.2e-16 ***
## ar4 -0.2030236  0.0059510 -34.1160 < 2.2e-16 ***
## ar5 -0.9674917  0.0153952 -62.8436 < 2.2e-16 ***
## ma1 -0.6953527  0.0144455 -48.1362 < 2.2e-16 ***
## ma2  0.3781931  0.0347072  10.8967 < 2.2e-16 ***
## ma3  0.1435506  0.0259575   5.5302 3.198e-08 ***
## ma4 -0.3706086  0.0168145 -22.0410 < 2.2e-16 ***
## ma5  0.7247722  0.0371346  19.5174 < 2.2e-16 ***
## ma6 -0.8316189  0.0219617 -37.8669 < 2.2e-16 ***
## sar1 -0.3152318      NaN      NaN      NaN
## sma1  0.3136630      NaN      NaN      NaN
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

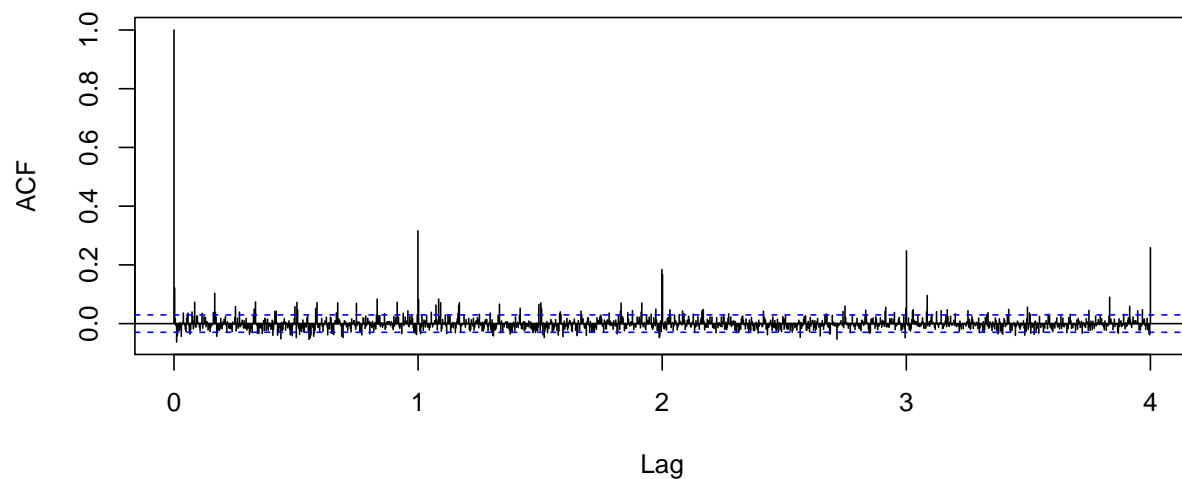
```
par(mfrow=c(1,1))
plot(residuals(nyc.v.sarima), ylab='Standardized Residuals', main="NYC Violent Crime SARIMA Residuals")
grid(lty=1, col=gray(0.8))
```

### NYC Violent Crime SARIMA Residuals



```
acf(residuals(nyc.v.sarima), lag.max = 365.25*4, main="ACF of NYC Violent Crime SARIMA Residuals")
```

### ACF of NYC Violent Crime SARIMA Residuals



```
# nyc.p.sarima = arima(nyc.p.train, order = c(5,1,5),seasonal = list(order =c(1,0,1), period=30), metho
# nyc.p.sarima
# coeftest(nyc.p.sarima)
#
# par(mfrow=c(1,1))
# plot(residuals(nyc.p.sarima), ylab='Standardized Residuals', main="NYC Property Crime SARIMA Residual
# grid(lty=1, col=gray(0.8))
# acf(residuals(nyc.p.sarima), lag.max = 365.25*4, main="ACF of NYC Property Crime SARIMA Residuals")
```

#### d) NYC Property Crime

```
nyc.q.sarima = arima(nyc.q.train, order = c(6,1,6), seasonal = list(order = c(1,0,1), period=30), method="ML")
nyc.q.sarima
```

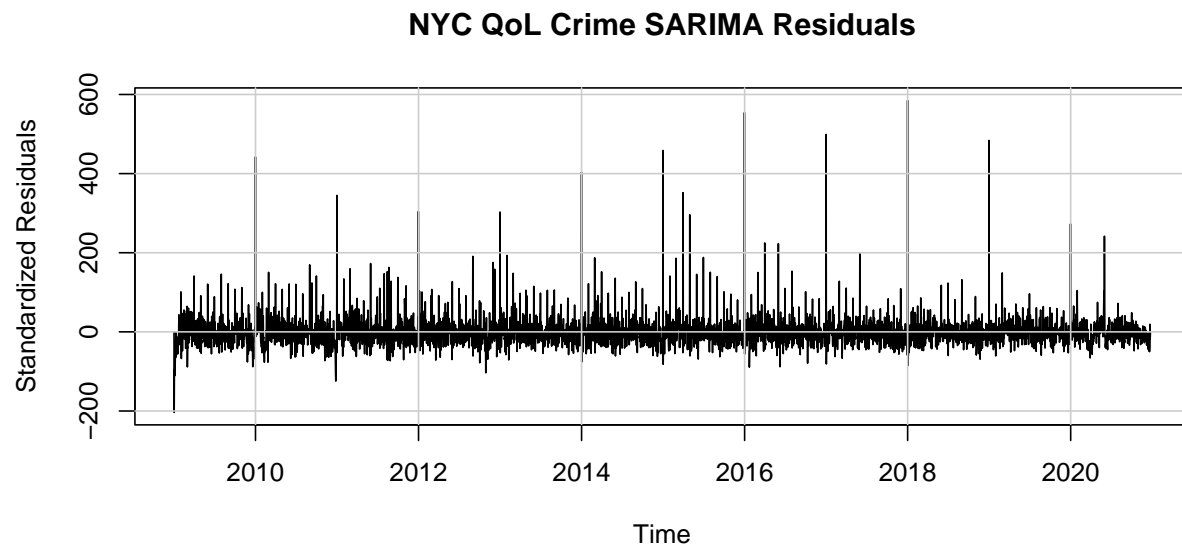
#### e) NYC QoL Crime

```
##
## Call:
## arima(x = nyc.q.train, order = c(6, 1, 6), seasonal = list(order = c(1, 0, 1),
##     period = 30), method = "ML")
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ar5      ar6      ma1      ma2
##      -0.4995 -0.2883 -0.6058 -0.7789  0.0920 -0.0838 -0.3734 -0.2243
## s.e.   0.1164  0.0721  0.0446  0.0781  0.0998  0.0201  0.1165  0.0400
##          ma3      ma4      ma5      ma6      sar1      sma1
##      0.2855  0.2487 -0.8863  0.094  -0.4094  0.4742
## s.e.   0.0354  0.0382  0.0373  0.107  0.1203  0.1162
##
## sigma^2 estimated as 1583:  log likelihood = -22324.74,  aic = 44679.48
```

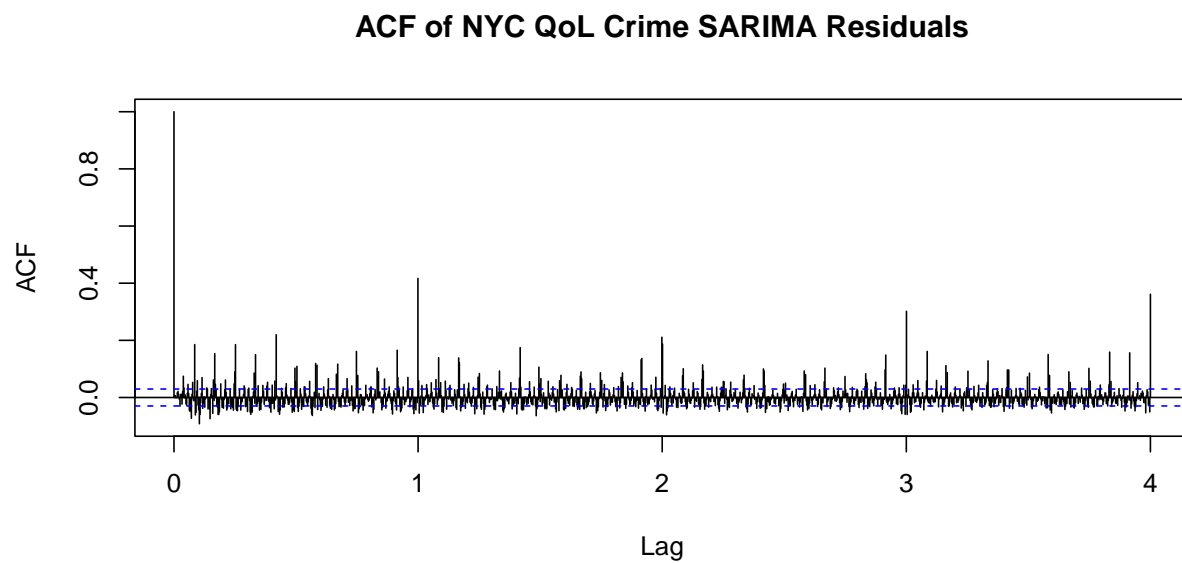
```
coeftest(nyc.q.sarima)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.499539   0.116430 -4.2905 1.783e-05 ***
## ar2 -0.288309   0.072107 -3.9983 6.379e-05 ***
## ar3 -0.605760   0.044637 -13.5709 < 2.2e-16 ***
## ar4 -0.778947   0.078050 -9.9800 < 2.2e-16 ***
## ar5  0.091971   0.099792  0.9216  0.356722
## ar6 -0.083839   0.020058 -4.1798 2.917e-05 ***
## ma1 -0.373361   0.116474 -3.2055  0.001348 **
## ma2 -0.224299   0.040024 -5.6041 2.093e-08 ***
## ma3  0.285500   0.035380  8.0696 7.052e-16 ***
## ma4  0.248706   0.038212  6.5085 7.589e-11 ***
## ma5 -0.886287   0.037305 -23.7578 < 2.2e-16 ***
## ma6  0.094042   0.107019  0.8787  0.379539
## sar1 -0.409445   0.120326 -3.4028  0.000667 ***
## sma1 0.474246   0.116199  4.0813 4.478e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow=c(1,1))
plot(residuals(nyc.q.sarima), ylab='Standardized Residuals', main="NYC QoL Crime SARIMA Residuals")
grid(lty=1, col=gray(0.8))
```



```
acf(residuals(nyc.q.sarima), lag.max = 365.25*4, main="ACF of NYC QoL Crime SARIMA Residuals")
```

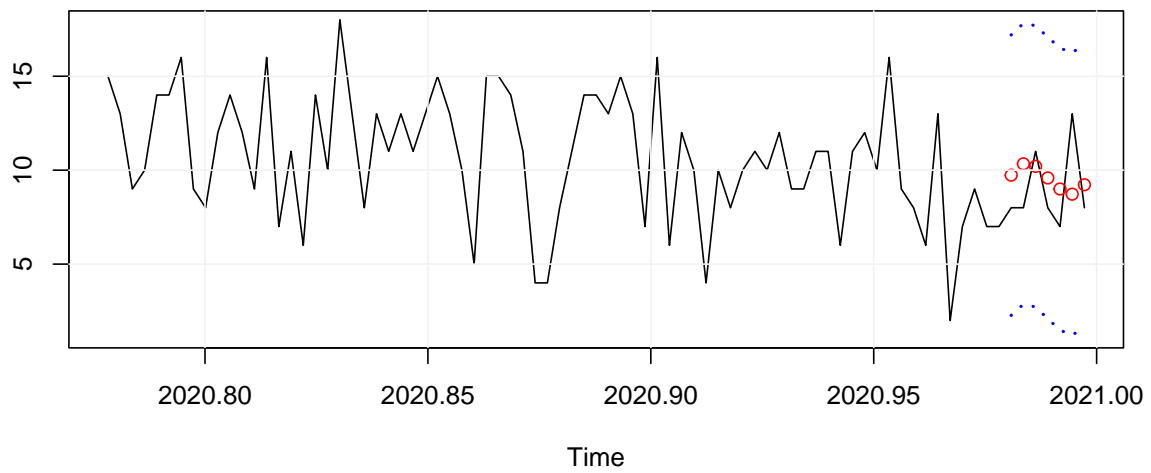


## Forecasting Analysis

```
## Forecast ATL Violent Crime
atl.v.spred <- as.vector(predict(atl.v.sarima, n.ahead=n.ahead))
plot_forecast(atl.v.ts, atl.v.spred, n.ahead, conf=1.96, plot_title = "ATL Violent Crime Forecast")
grid(lty=1, col=gray(0.95))
```

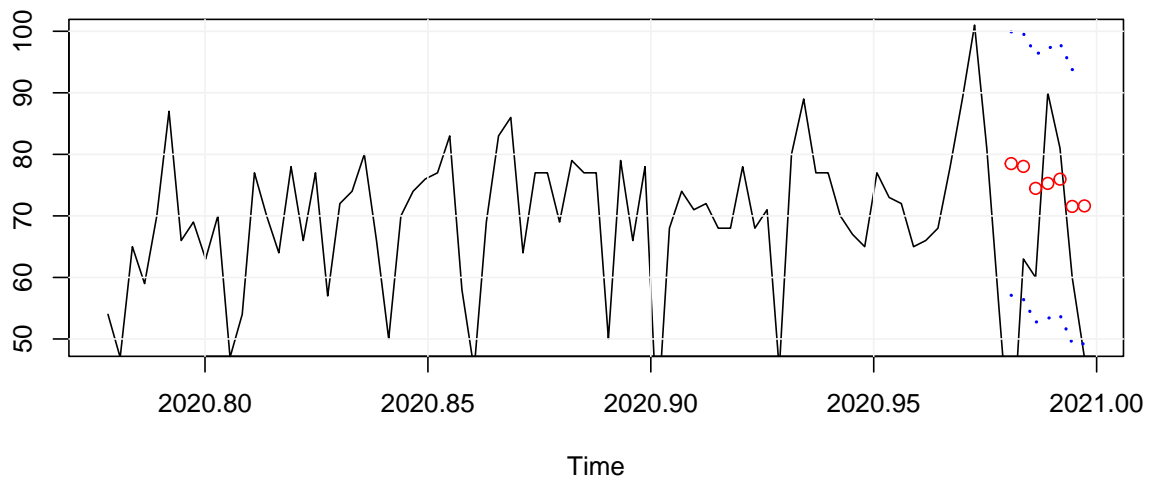


### ATL Violent Crime Forecast



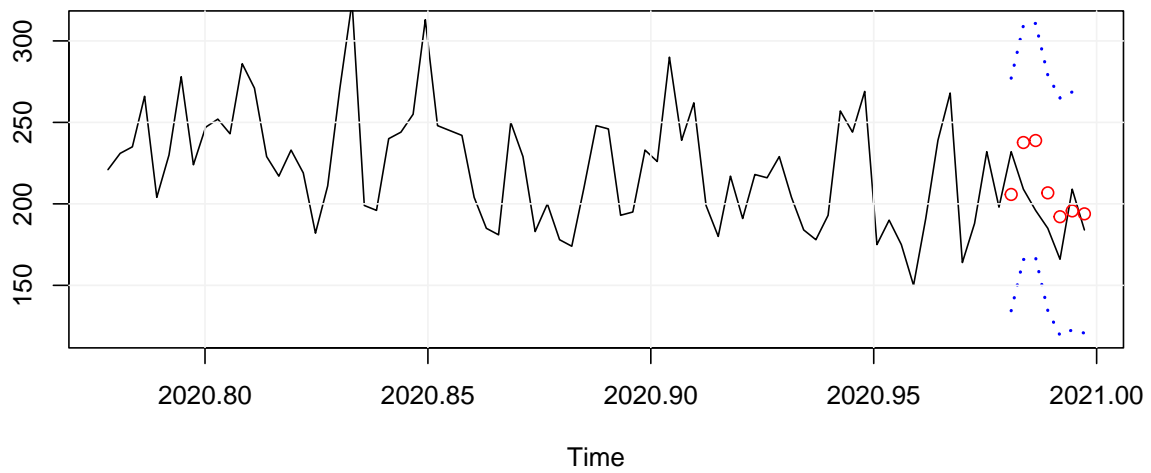
```
## Forecast ATL Property Crime
atl.p.spred <- as.vector(predict(atl.p.sarima, n.ahead=n.ahead))
plot_forecast(atl.p.ts, atl.p.spred, n.ahead, conf=1.96, plot_title = "ATL Property Crime Forecast")
grid(lty=1, col=gray(0.95))
```

### ATL Property Crime Forecast



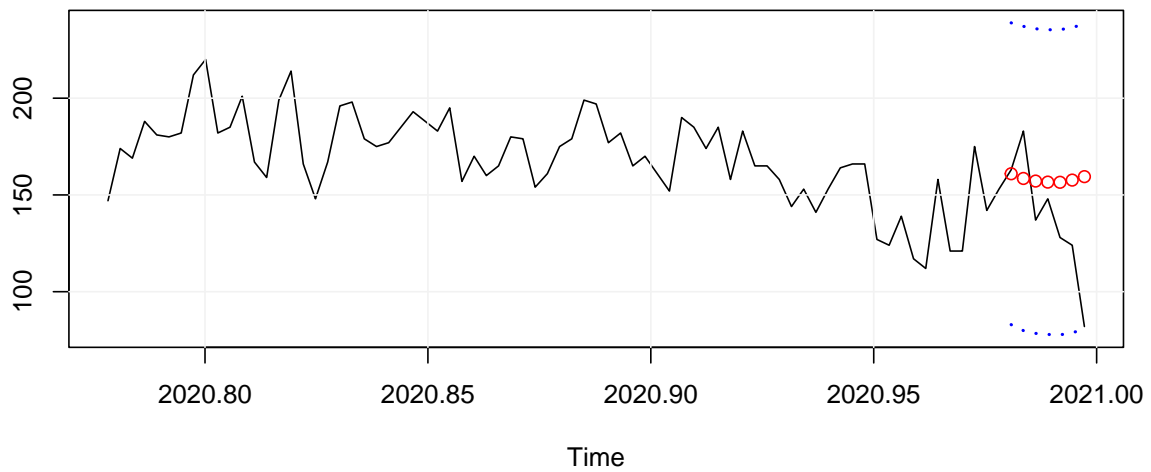
```
## Forecast NYC Violent Crime
nyc.v.spred <- as.vector(predict(nyc.v.sarima, n.ahead=n.ahead))
plot_forecast(nyc.v.ts, nyc.v.spred, n.ahead, conf=1.96, plot_title = "NYC Violent Crime Forecast")
grid(lty=1, col=gray(0.95))
```

### NYC Violent Crime Forecast



```
## Forecast NYC QoL Crime
nyc.q.spred <- as.vector(predict(nyc.q.sarima, n.ahead=n.ahead))
plot_forecast(nyc.q.ts, nyc.q.spred, n.ahead, conf=1.96, plot_title = "NYC QoL Crime Forecast")
grid(lty=1, col=gray(0.95))
```

### NYC QoL Crime Forecast



### Evaluation

```
atl.v.smape <- mape(atl.test$violentCrime, atl.v.spred$pred)
atl.p.smape <- mape(atl.test$propertyCrime, atl.p.spred$pred)
atl.v.spm <- pm(atl.test$violentCrime, atl.v.spred$pred)
```

```

atl.p.spm <- pm(atl.test$propertyCrime, atl.p.spred$pred)

nyc.v.smape <- mape(nyc.test$violentCrime, nyc.v.spred$pred)
nyc.q.smape <- mape(nyc.test$qolCrime, nyc.q.spred$pred)
nyc.v.spm <- pm(nyc.test$violentCrime, nyc.v.spred$pred)
nyc.q.spm <- pm(nyc.test$qolCrime, nyc.q.spred$pred)

cat("ATL Violent:\nMAPE =", atl.v.smape, "\nPM =", atl.v.spm,
    "\n\nATL Property:\nMAPE =", atl.p.smape, "\nPM =", atl.p.spm,
    "\n\nNYC Violent:\nMAPE =", nyc.v.smape, "\nPM =", nyc.v.spm,
    "\n\nNYC QoL:\nMAPE =", nyc.q.smape, "\nPM =", nyc.q.spm)

```

```

## ATL Violent:
## MAPE = 0.2210196
## PM = 1.264791
##
## ATL Property:
## MAPE = 0.4340857
## PM = 1.569532
##
## NYC Violent:
## MAPE = 0.1230118
## PM = 1.71343
##
## NYC QoL:
## MAPE = 0.2558379
## PM = 1.460691

```