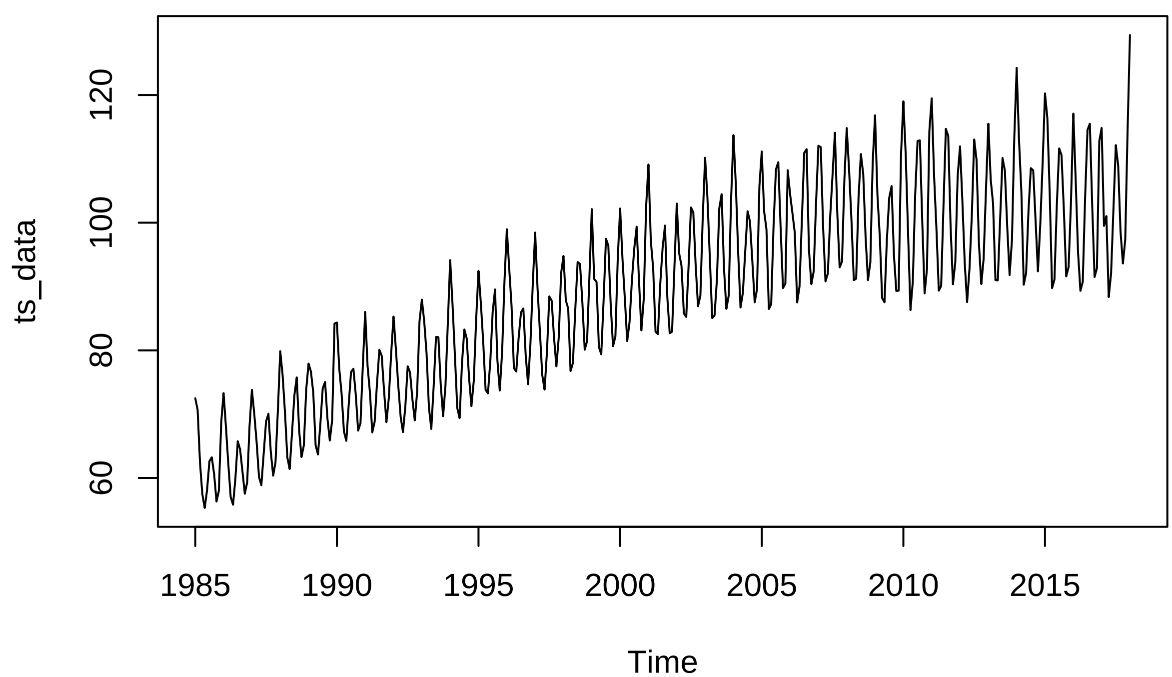
Project Step 5

Kislay

2024-04-27

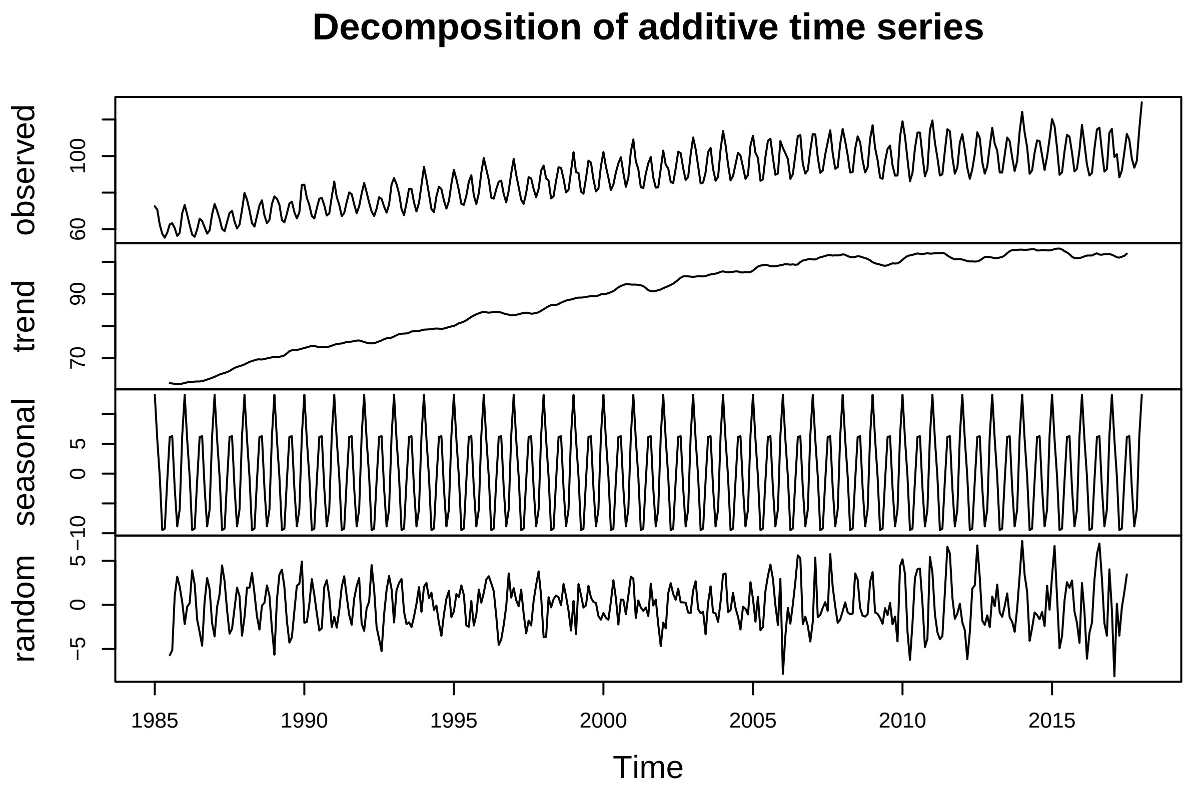
```
data <- read.csv("/Users/kislaynandan/Desktop/MA 641/Electric_Production.csv")
df = data
setDT(data)
df$DateTime <- as.POSIXct(paste(df$DATE), format="%Y-%m-%d")
df$Month = month(df$DateTime)

df$Year = year(df$DateTime)
```

```
ts_data <- ts(df$Value, start = min(df$Year), end = max(df$Year), frequency = 12)
plot(ts_data)
```



Decomposed Data

```
decomp_result <- decompose(ts_data)
plot(decomp_result)
```



#Stationarity Test

```
result <- adf.test(ts_data)
```

```
## Warning in adf.test(ts_data): p-value smaller than printed p-value
```

```
result
```

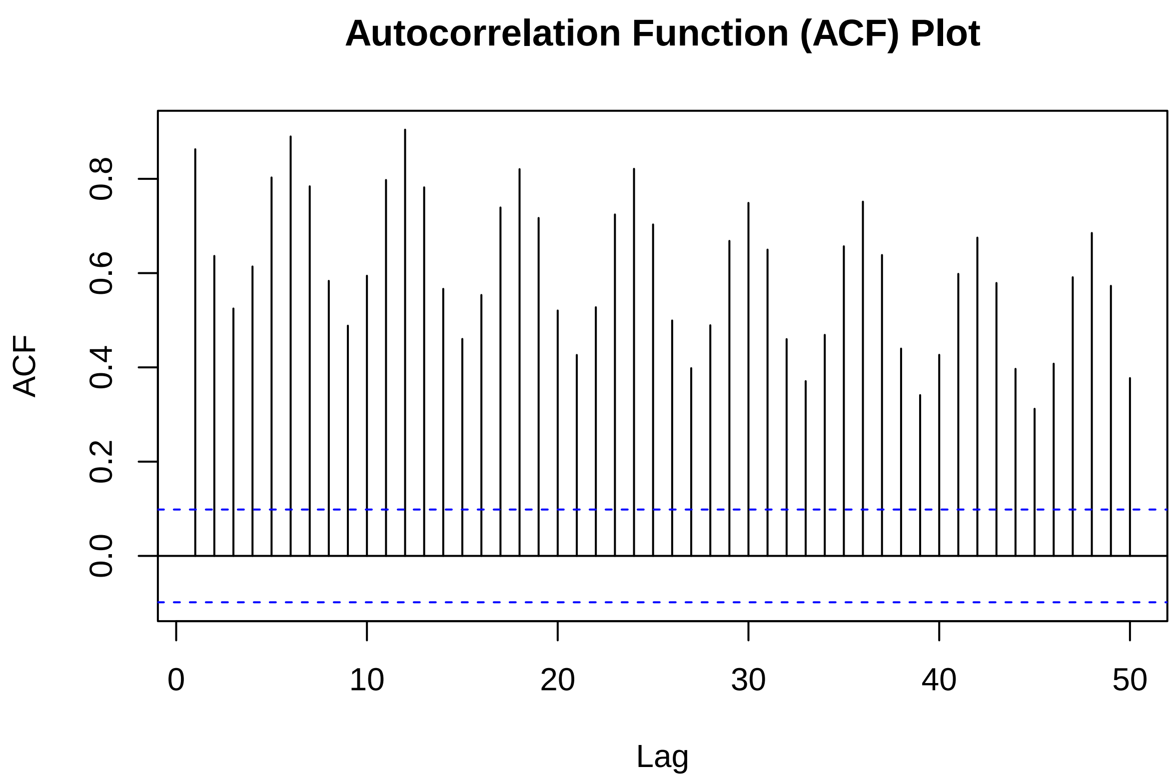
```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_data
## Dickey-Fuller = -5.139, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
cat("p-value:", result$p.value)
```

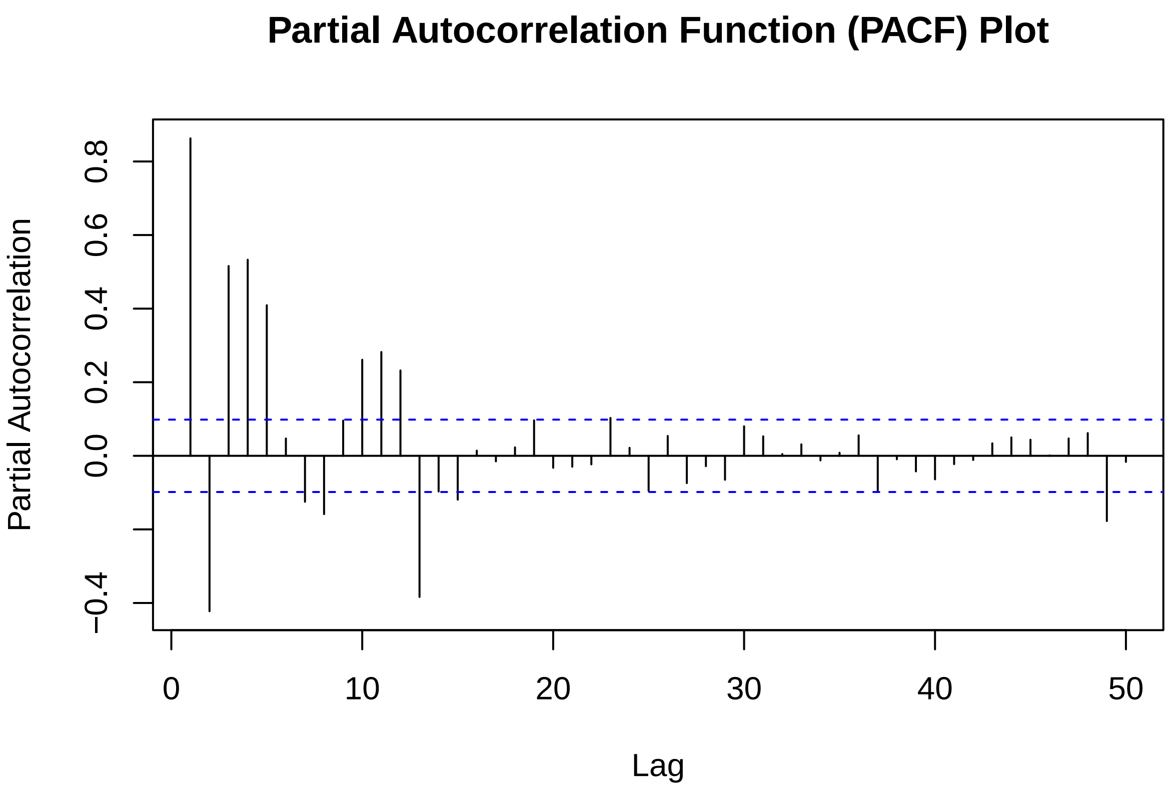
```
## p-value: 0.01
```

Data Visualisation

```r
acf(df$Value, lag.max = 50,
    main = "Autocorrelation Function (ACF) Plot",
    xlab = "Lag", ylab = "ACF")
```




```r
pacf(df$Value, lag.max = 50,
    main = "Partial Autocorrelation Function (PACF) Plot",
    xlab = "Lag", ylab = "Partial Autocorrelation")
```




```
eacf(df$Value)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 x x x x x x x x x x x  x  x  x
## 2 x x x x x x x x x x x  x  x  x
## 3 x x o x o x o o o o x  x  x  x
## 4 x x x x o x o o o o o  x  x  x
## 5 x x x x o o o o o o o  x  o  o
## 6 x x o x o x o o o o o  x  x  o
## 7 x x o x x o o o o o o  x  x  o
```

Model fitting

SARIMA MODEL FITTING

- Because the series is seasonal, SARIMA (Seasonal ARIMA) will be used instead of ARIMA. From ACF and PACF plot below models are chosen to fit to the data:
- Fit 1: SARIMA(5,0,0)(1,0,0)[12]
- Fit 2: SARIMA(4,0,0)(1,0,0)[12]
- Fit 3: SARIMA(3,0,0)(1,0,0)[12]

Since the data is daily, seasonality is 12.

```r
fit <- auto.arima(ts_data)
fit
```

```
## Series: ts_data
## ARIMA(2,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ar2      ma1     sma1
##       0.5503  -0.0683  -0.9477  -0.7635
## s.e.  0.0544   0.0549   0.0193   0.0331
##
## sigma^2 = 5.838:  log likelihood = -888.05
## AIC=1786.11   AICc=1786.27   BIC=1805.86
```

```r
sarima_model <- Arima(df$Value, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12))

sarima_model
```

```
## Series: df$Value
## ARIMA(2,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ar2      ma1     sma1
##       0.5503  -0.0683  -0.9477  -0.7635
## s.e.  0.0544   0.0549   0.0193   0.0331
##
## sigma^2 = 5.838:  log likelihood = -888.05
## AIC=1786.11   AICc=1786.27   BIC=1805.86
```

```r
Arima(df$Value, order = c(5, 0, 0), seasonal = list(order = c(1, 0, 0), period = 12))
```

```
## Series: df$Value
## ARIMA(5,0,0)(1,0,0)[12] with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ar4     ar5    sar1     mean
##       0.6461  -0.1252  0.2403  -0.0944  0.0688  0.9414  86.7475
## s.e.  0.0541   0.0623  0.0604   0.0611  0.0568  0.0183   6.6373
##
## sigma^2 = 8.452:  log likelihood = -996.96
## AIC=2009.92   AICc=2010.3   BIC=2041.8
```

```r
Arima(df$Value, order = c(4, 0, 0), seasonal = list(order = c(1, 0, 0), period = 12))
```

```
## Series: df$Value
## ARIMA(4,0,0)(1,0,0)[12] with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ar4    sar1     mean
##       0.6369  -0.1072  0.2350  -0.0572  0.9495  86.4348
## s.e.  0.0533   0.0602  0.0602   0.0529  0.0149   6.6934
```

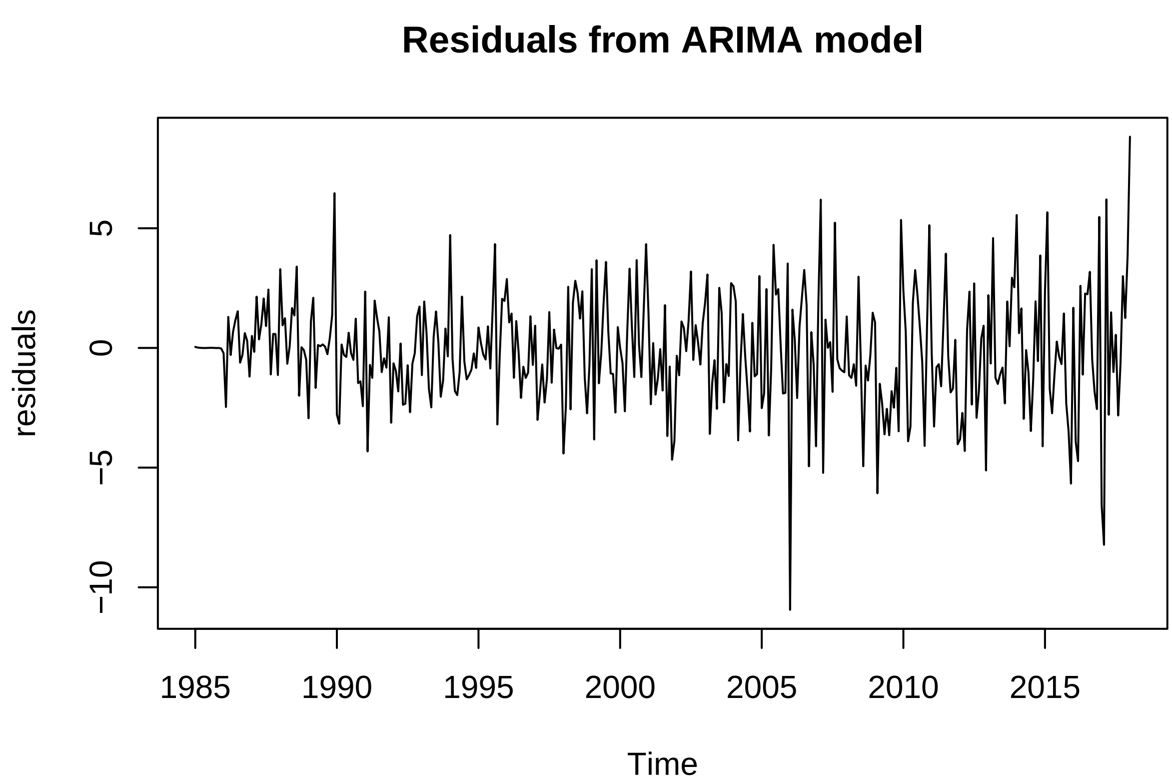
```
##
## sigma^2 = 8.428:  log likelihood = -997.71
## AIC=2009.42   AICc=2009.71   BIC=2037.31
```

```
Arima(df$Value, order = c(3, 0, 0), seasonal = list(order = c(1, 0, 0), period = 12))
```

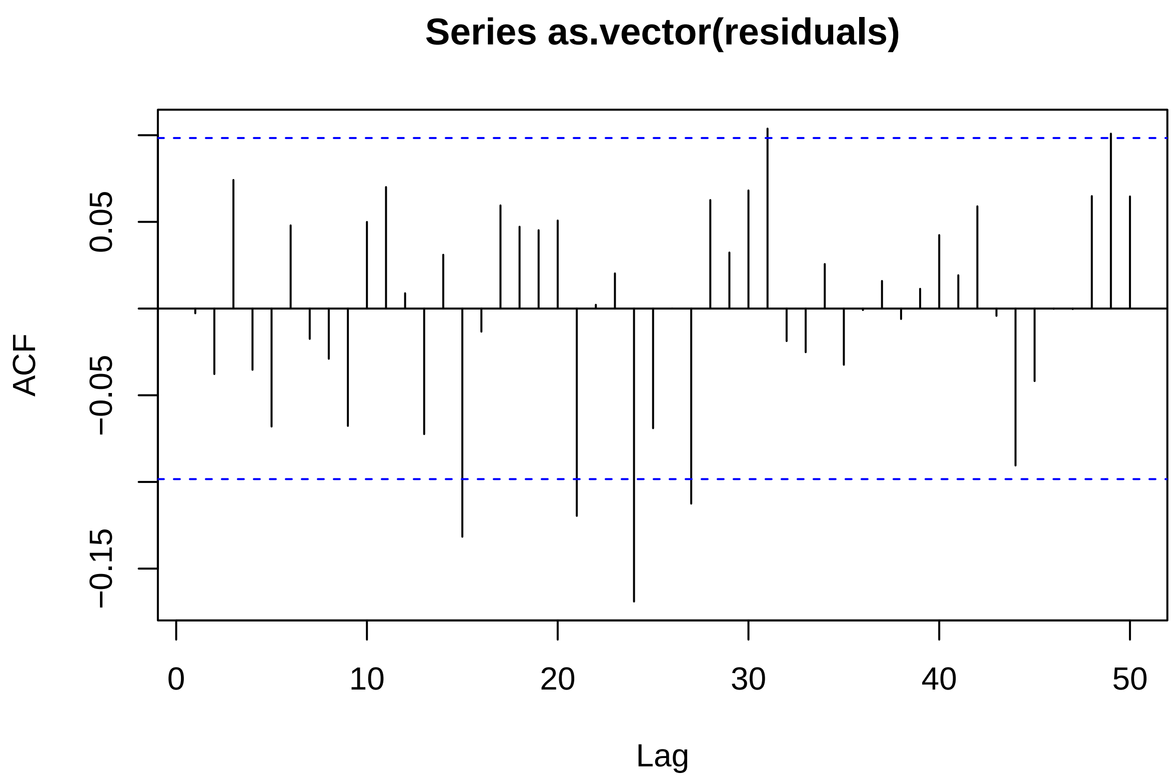
```
## Series: df$Value
## ARIMA(3,0,0)(1,0,0)[12] with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3    sar1     mean
##       0.6293  -0.1021  0.1997  0.9454  86.7786
## s.e.  0.0532   0.0602  0.0506  0.0153   6.7813
##
## sigma^2 = 8.449:  log likelihood = -998.29
## AIC=2008.58   AICc=2008.8   BIC=2032.49
```

Residual Analysis

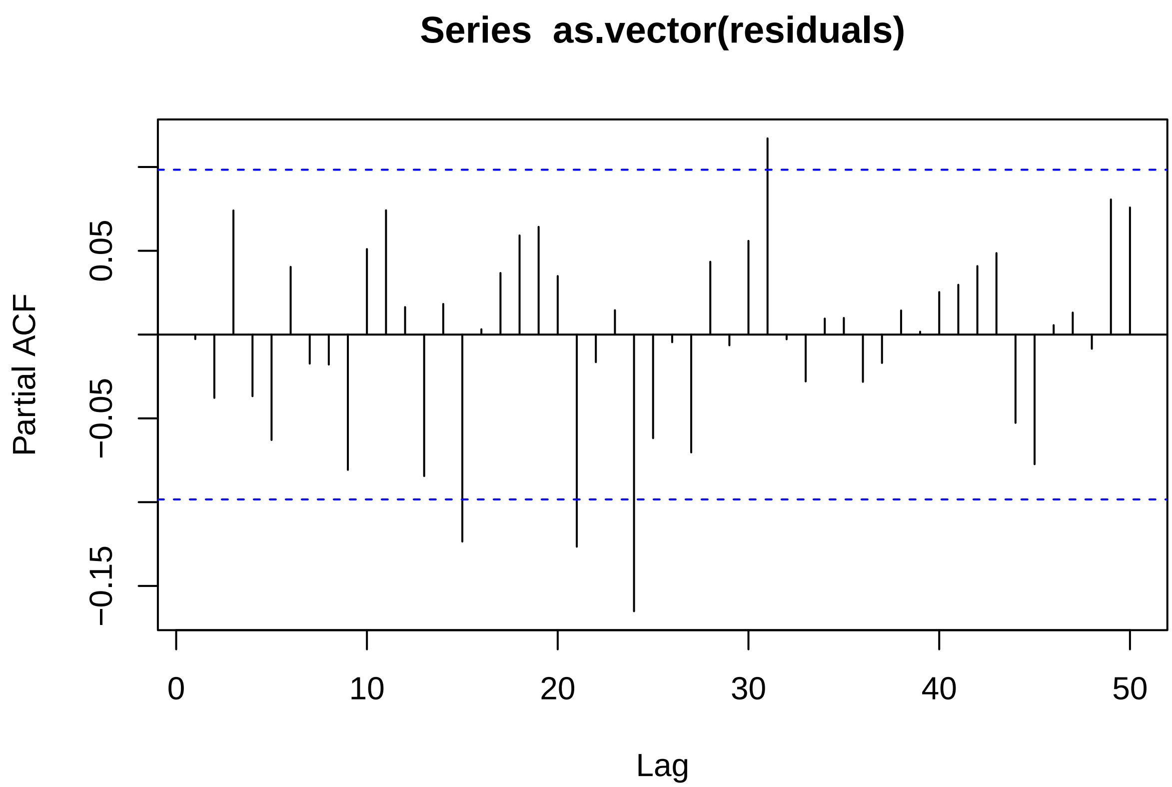
```
residuals <- residuals(fit)
plot(residuals, main="Residuals from ARIMA model")
```



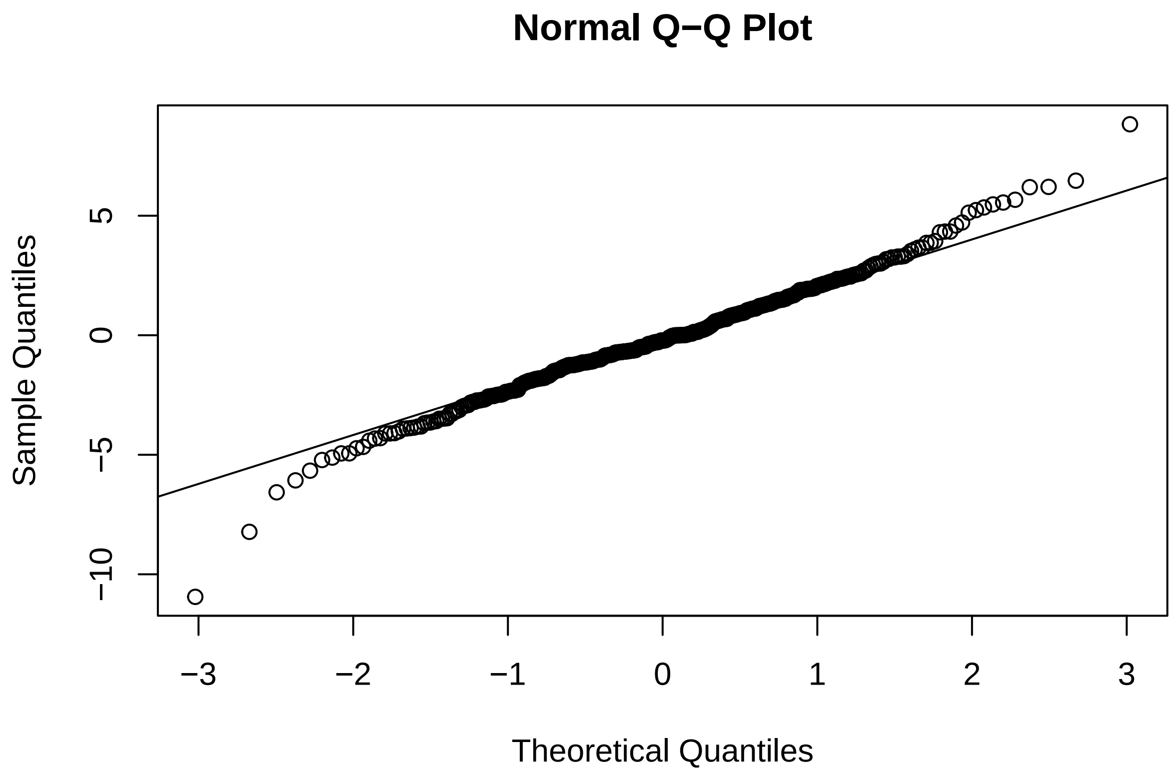
```r
acf(as.vector(residuals), lag.max = 50)
```



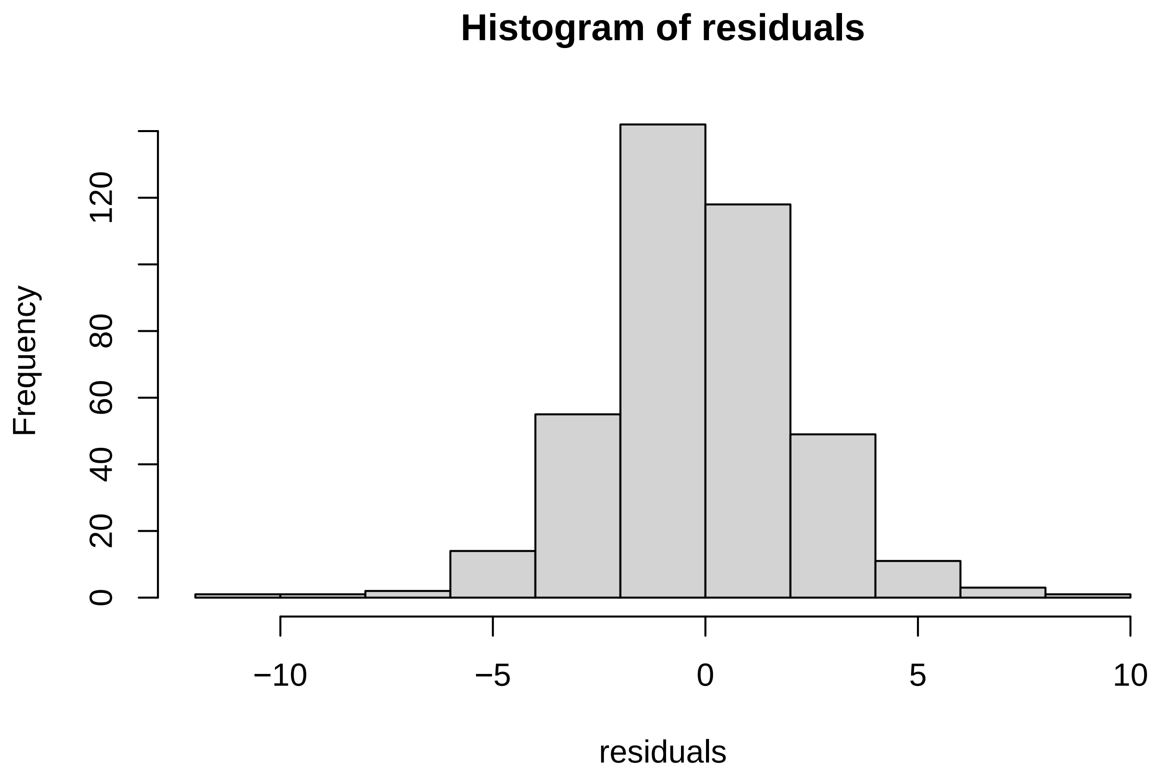

```r
pacf(as.vector(residuals), lag.max = 50)
```




```
qqnorm(residuals)
qqline(residuals)
```




```
hist(residuals)
```




```r
shapiro.test(residuals)
```

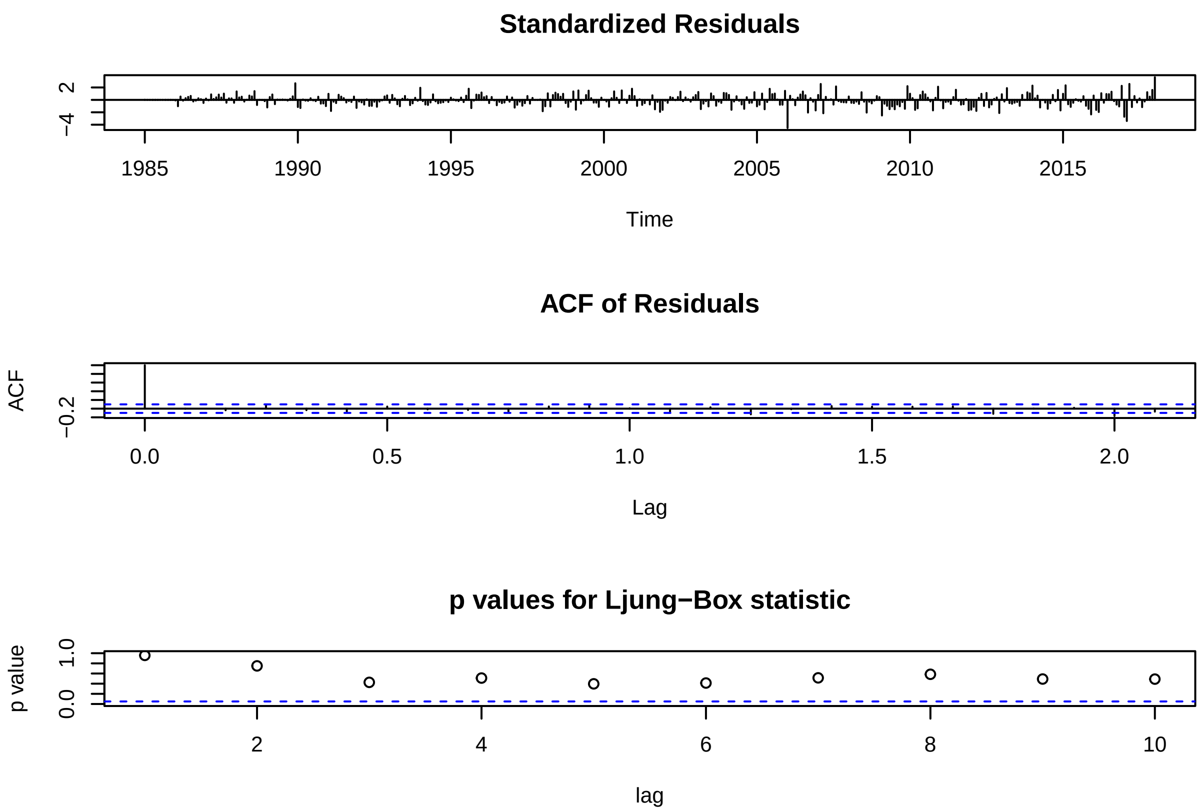
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.98648, p-value = 0.0009324
```

```r
Box.test(residuals,lag=10, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals
## X-squared = 9.4504, df = 10, p-value = 0.49
```

The ACF plot of the residuals shows that most autocorrelations are within the confidence bounds (the blue dotted lines), which is a good indication that the residuals are white noise. The plot shows most points lie close to the reference line, suggesting that the residuals are approximately normally distributed. The histogram shows a relatively bell-shaped curve, but it is not perfectly symmetric, and there appears to be a slight skew to the right. With a p-value of 0.49, which is above the alpha level of 0.05, we fail to reject the null hypothesis that the residuals are independently distributed, meaning there is no autocorrelation.

```
tsdiag(fit)
```

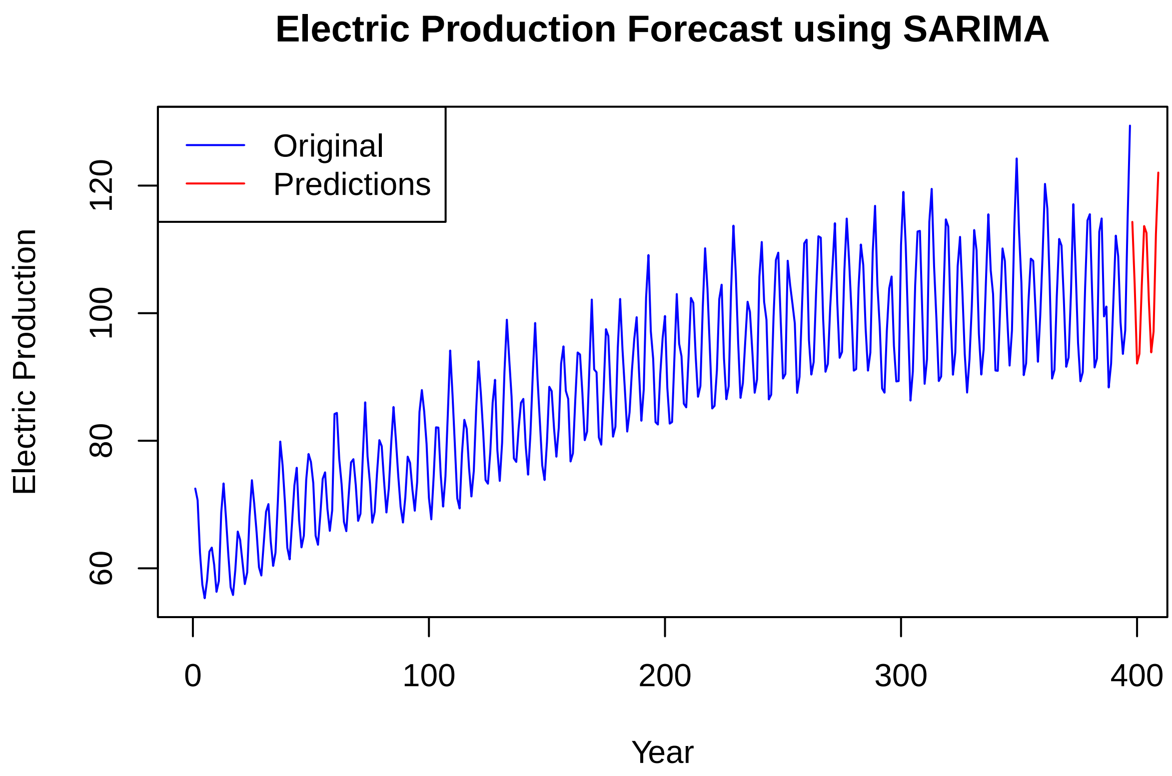


Prediction

```
predictions <- forecast(sarima_model, h = 12)
predictions
```

```
##     Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## 398      114.31111 111.21470 117.40753 109.57555 119.04667
## 399      104.45857 100.84328 108.07386  98.92946 109.98768
## 400       92.09910  88.35393  95.84426  86.37136  97.82683
## 401       93.63140  89.84270  97.42011  87.83708  99.42572
## 402      104.31821 100.50733 108.12909  98.48997 110.14645
## 403      113.65996 109.83314 117.48678 107.80734 119.51258
## 404      112.58325 108.74256 116.42394 106.70943 118.45708
## 405      101.93541  98.08160 105.78922  96.04152 107.82930
## 406       93.85642  89.98978  97.72305  87.94291  99.76992
## 407       97.12217  93.24285 101.00150  91.18925 103.05509
## 408      112.41629 108.52434 116.30823 106.46407 118.36850
## 409      122.04284 118.13833 125.94735 116.07140 128.01428
```

```
plot(df$Value, type = "l", col = "blue", xlab = "Year", ylab = "Electric Production", main = "Electric I
lines(predictions$mean, col = "red")
legend("topleft", legend = c("Original", "Predictions"), col = c("blue", "red"), lty = c(1, 1))
```



Non Seasonal Data

```
library(data.table)
library(forecast)
library(tseries)
library(lubridate)
```

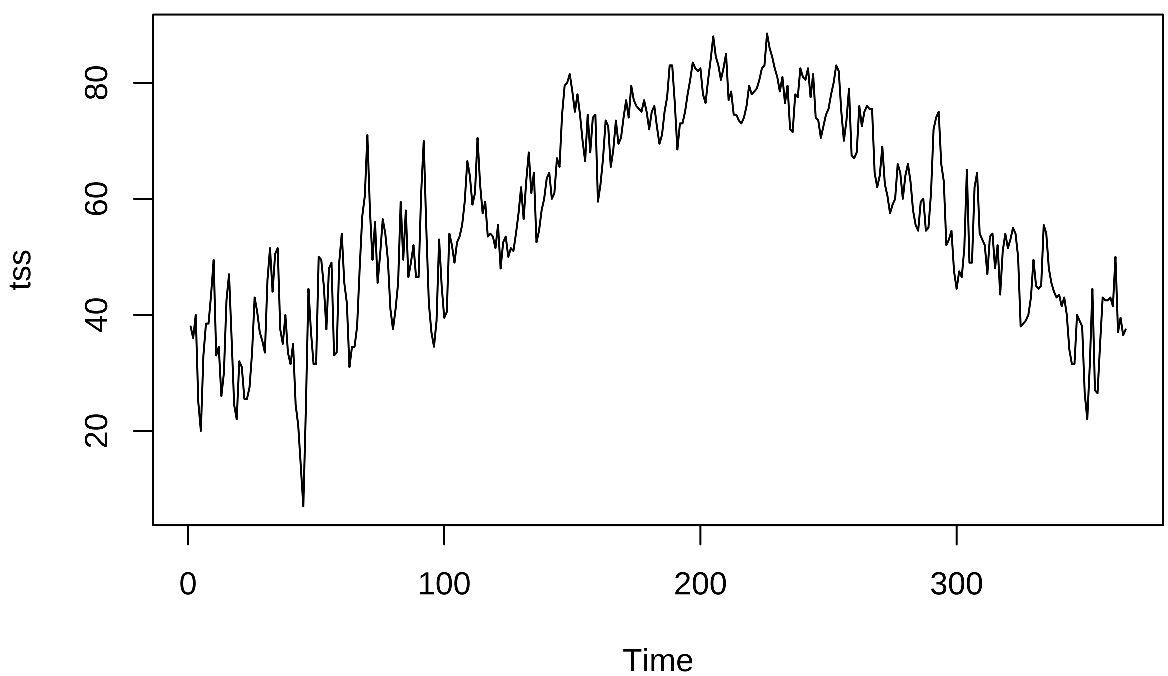
```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

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

```r
library(ggplot2)
library(MASS)
```

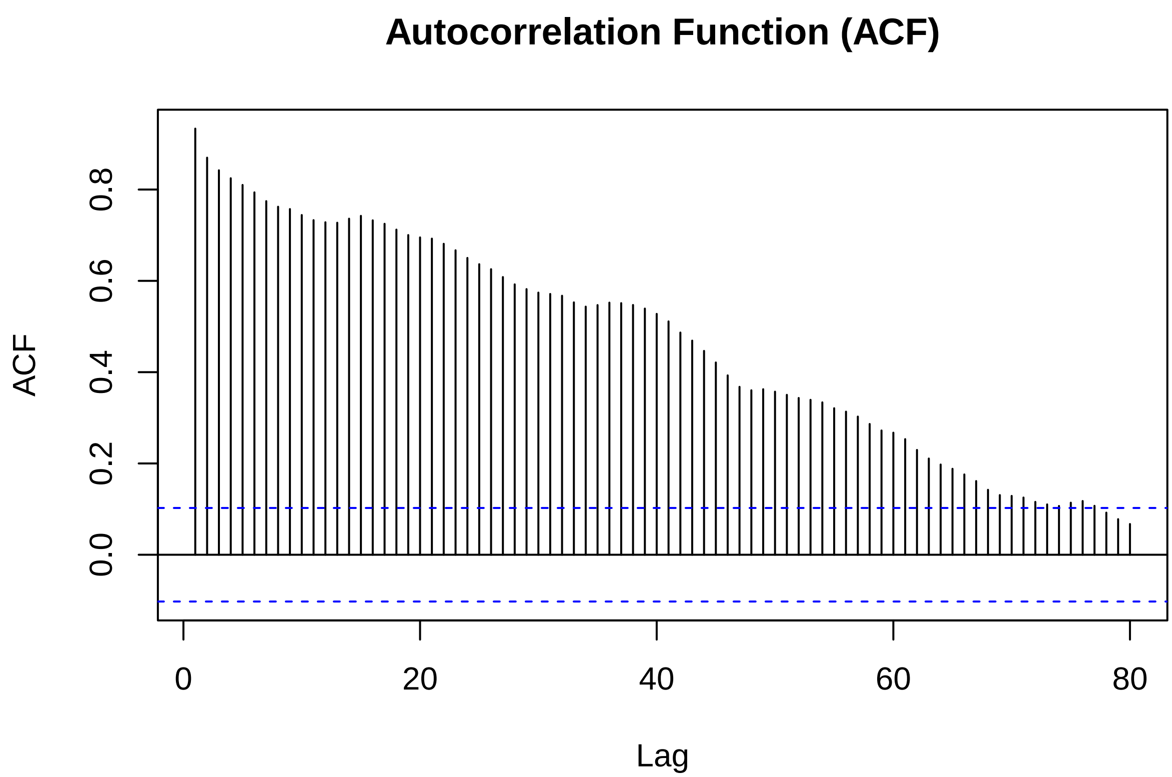
```
##
## Attaching package: 'MASS'

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

```r
library(TSA)
```

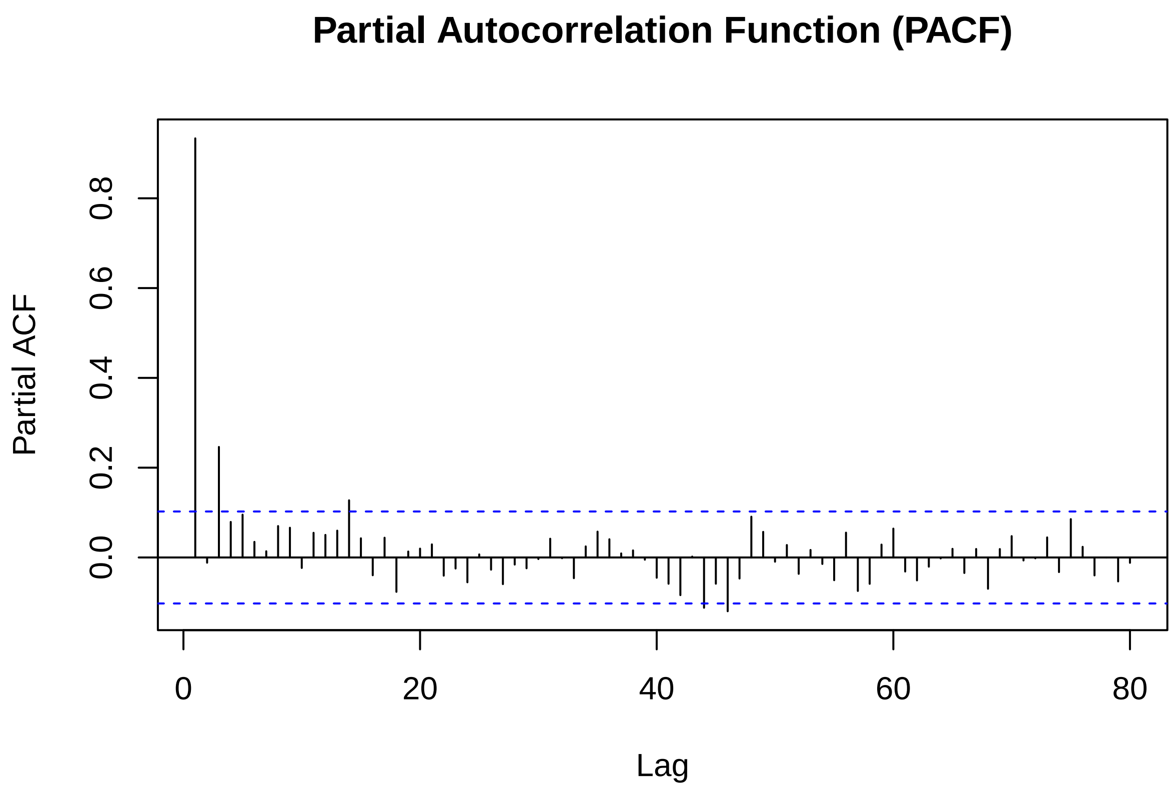
```r
plot(tss)
```



```r
acf(tss, main = "Autocorrelation Function (ACF)", lag.max = 80)
```




```r
pacf(tss, main = "Partial Autocorrelation Function (PACF)",lag.max = 80)
```



The above ACF is decaying/decreasing, very slowly, and remains well above the significance range (dotted blue lines). This is indicative of a non-stationary series. # Stationarity Test

```
g = as.numeric(tss)
adf.test(g)
```

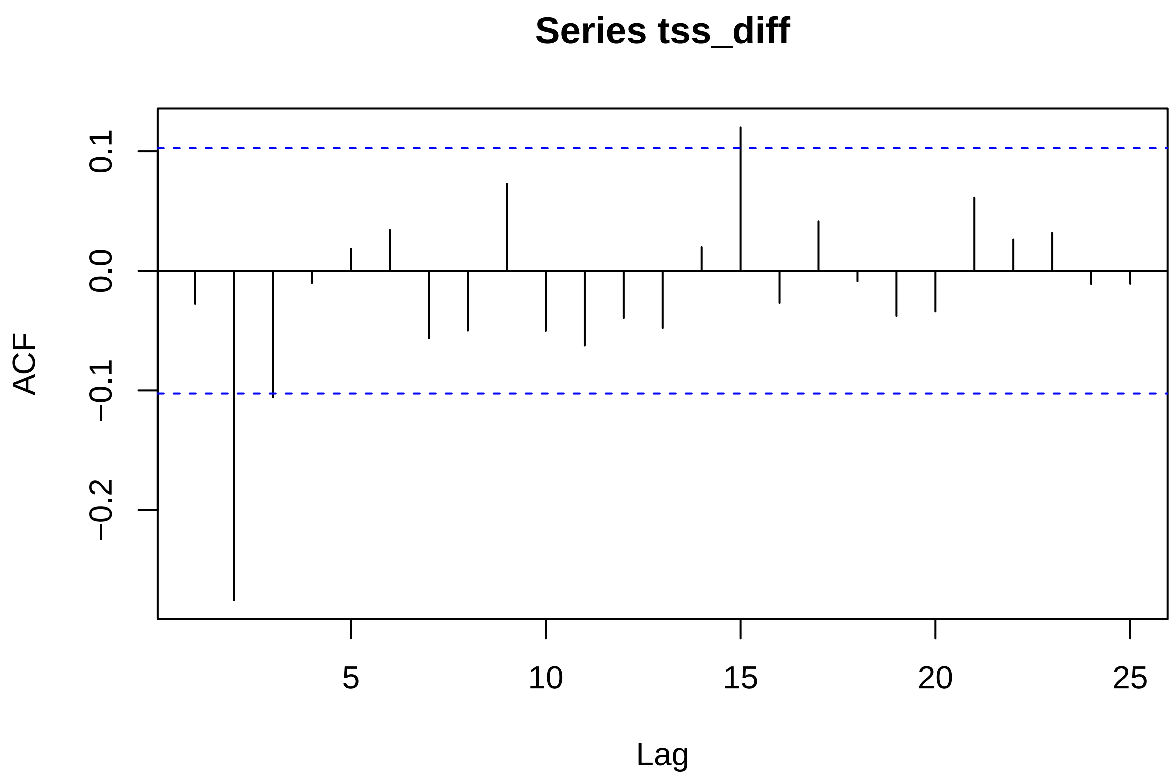
```
##
##  Augmented Dickey-Fuller Test
##
## data:  g
## Dickey-Fuller = -1.5185, Lag order = 7, p-value = 0.7802
## alternative hypothesis: stationary
```

```
tss_diff = diff(tss)
# Stationarity after differencing
h = as.numeric(tss_diff)
adf.test(h)
```

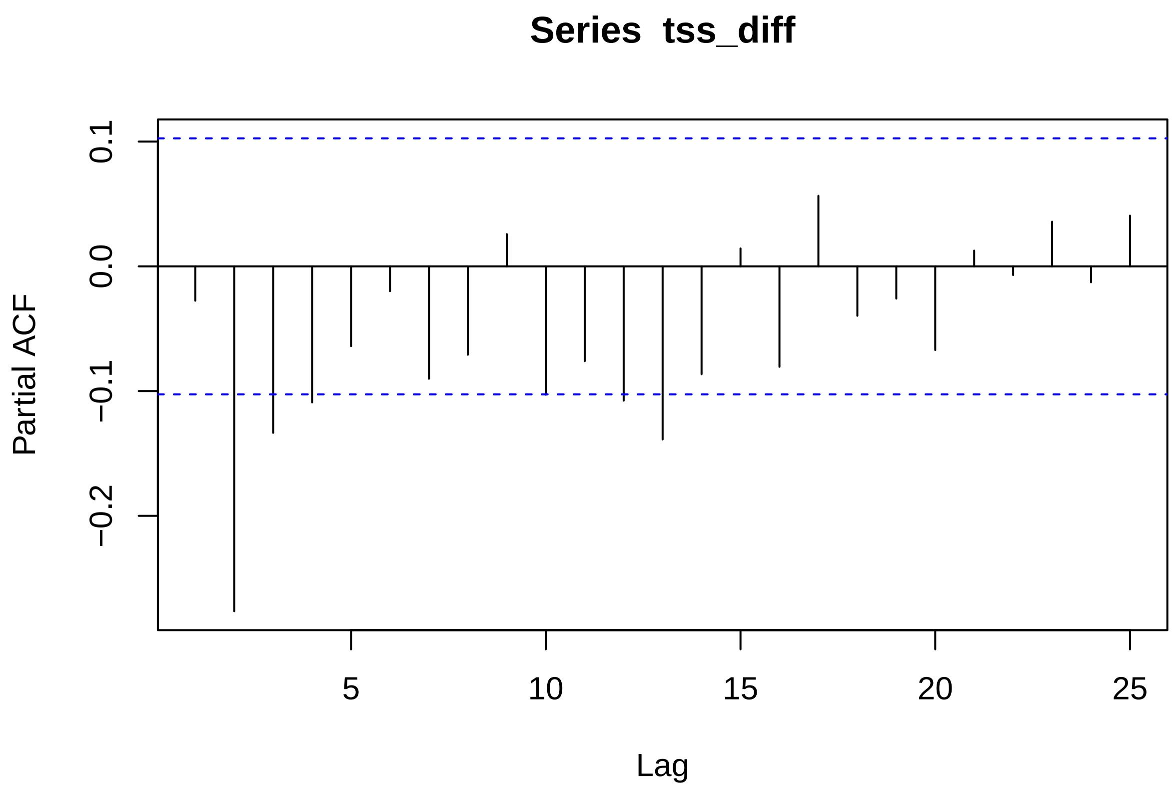
```
## Warning in adf.test(h): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  h
## Dickey-Fuller = -9.2043, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
acf(tss_diff)
```




```
pacf(tss_diff)
```




```
eacf(tss_diff)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o x x o o o o o o o o  o  o  o
## 1 o x o o o o o o o o o  o  o  o
## 2 x x o o o o o o o o o  o  o  o
## 3 x x o o o o o o o o o  o  o  o
## 4 x x x o x o o o o o o  o  o  o
## 5 x o x x o o o o o o o  o  o  o
## 6 x x x o o o o o o o o  o  o  o
## 7 x x x x o o o o o o o  o  o  o
```

Model fitting

- Model 1: ARIMA(3,1,2)
- Model 2: ARIMA(2,1,2)
- Model 3: ARIMA(1,1,1)
- Model 4: ARIMA(0,1,1)

```
arimafit <- auto.arima(tss)
arimafit
```

```
## Series: tss
## ARIMA(1,1,2)
##
## Coefficients:
##          ar1      ma1      ma2
##       0.4199  -0.5435  -0.2880
## s.e.  0.0957   0.0964   0.0669
##
## sigma^2 = 32.26:  log likelihood = -1150.7
## AIC=2309.4   AICc=2309.51   BIC=2325
```

```r
Arima(tss, order = c(3, 1, 2))
```

```
## Series: tss
## ARIMA(3,1,2)
##
## Coefficients:
##          ar1      ar2     ar3      ma1     ma2
##       0.8679  -0.3260  0.1226  -0.9952  0.0916
## s.e.  0.4978   0.3643  0.1010   0.5012  0.4379
##
## sigma^2 = 32.38:  log likelihood = -1150.3
## AIC=2312.61   AICc=2312.84   BIC=2336.01
```

```r
Arima(tss, order = c(2, 1, 2))
```

```
## Series: tss
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1     ar2      ma1      ma2
##       0.3719  0.0402  -0.4973  -0.3330
## s.e.  0.2056  0.1548   0.1976   0.1821
##
## sigma^2 = 32.35:  log likelihood = -1150.66
## AIC=2311.33   AICc=2311.5   BIC=2330.83
```

```r
Arima(tss, order = c(1, 1, 2))
```

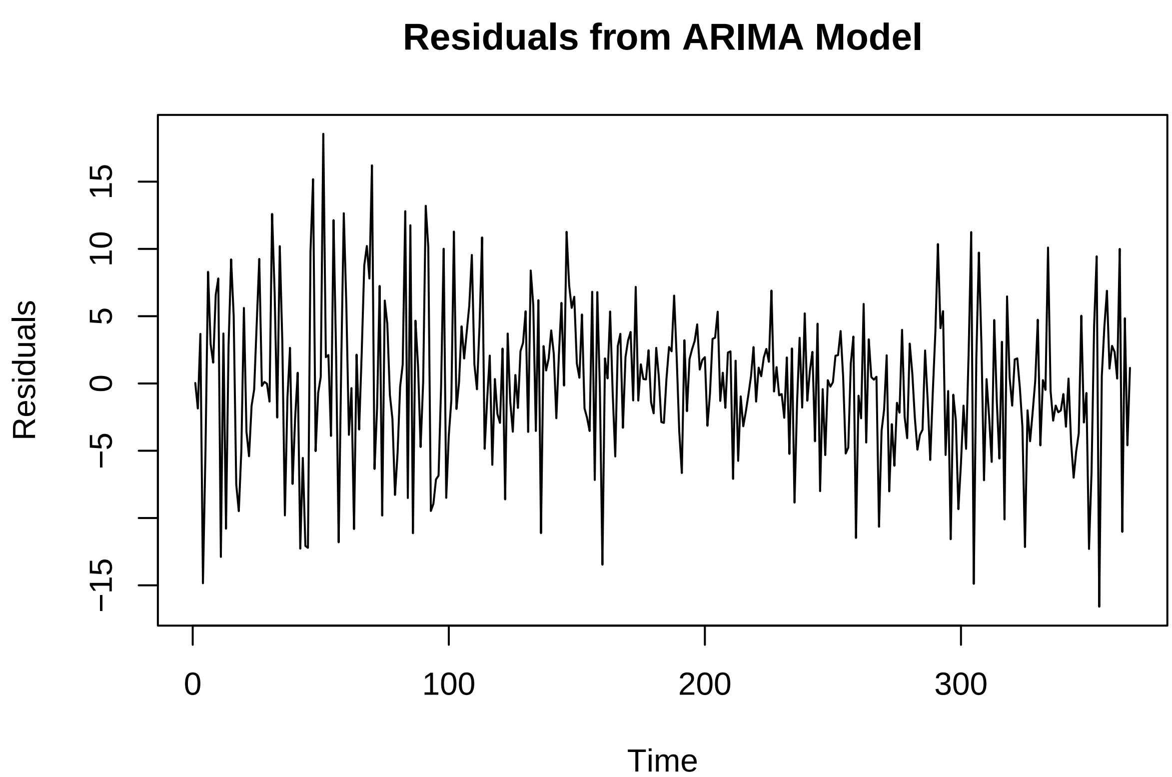
```
## Series: tss
## ARIMA(1,1,2)
##
## Coefficients:
##          ar1      ma1      ma2
##       0.4199  -0.5435  -0.2880
## s.e.  0.0957   0.0964   0.0669
##
## sigma^2 = 32.26:  log likelihood = -1150.7
## AIC=2309.4   AICc=2309.51   BIC=2325
```

```r
model <- Arima(tss, order = c(1, 1, 2))
model
```

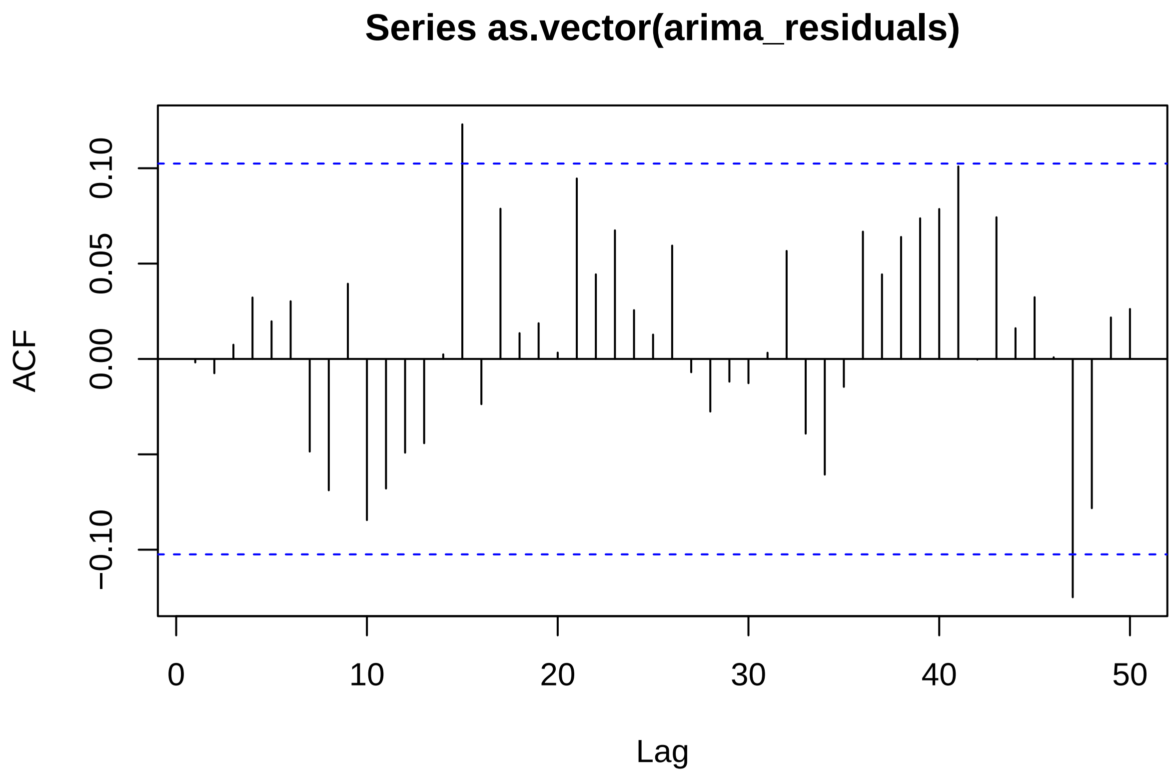
```
## Series: tss
## ARIMA(1,1,2)
##
## Coefficients:
##          ar1      ma1      ma2
##       0.4199  -0.5435  -0.2880
## s.e.  0.0957   0.0964   0.0669
##
## sigma^2 = 32.26:  log likelihood = -1150.7
## AIC=2309.4   AICc=2309.51   BIC=2325
```

Residual Analysis

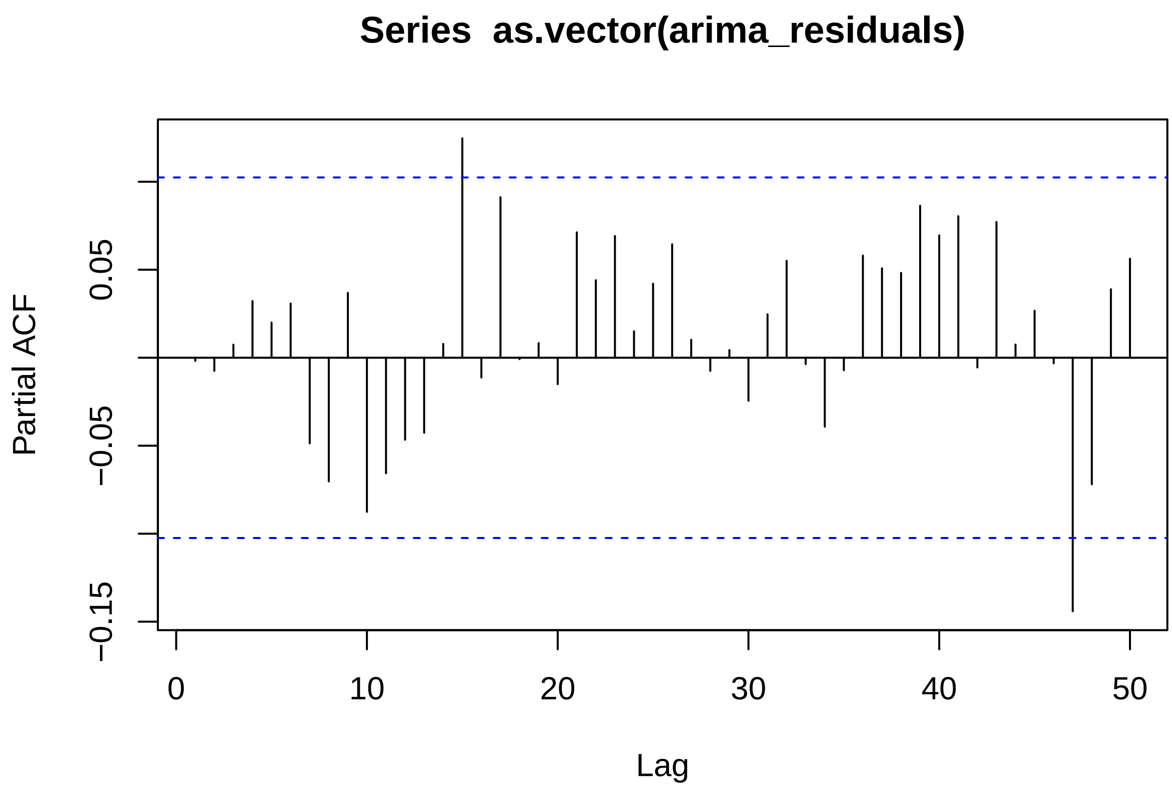
```r
arima_residuals <- residuals(model)
plot(arima_residuals, main = "Residuals from ARIMA Model", ylab = "Residuals")
```



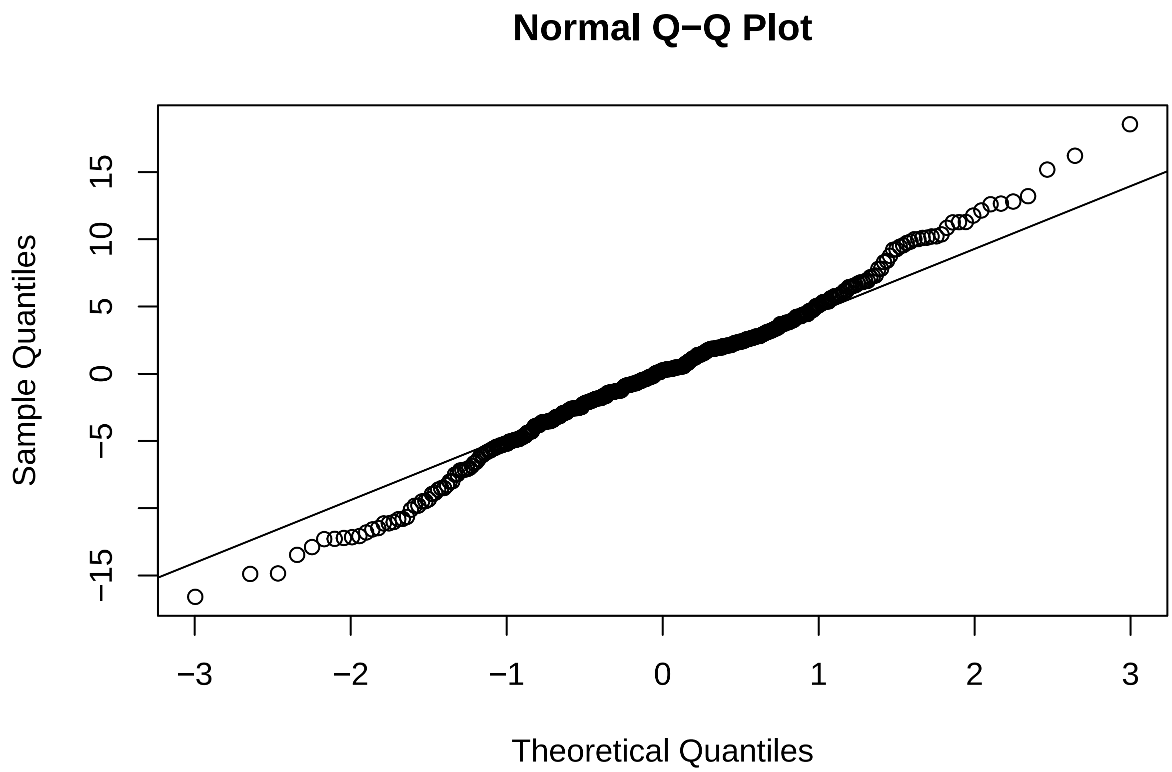

```r
acf(as.vector(arima_residuals), lag.max = 50)
```



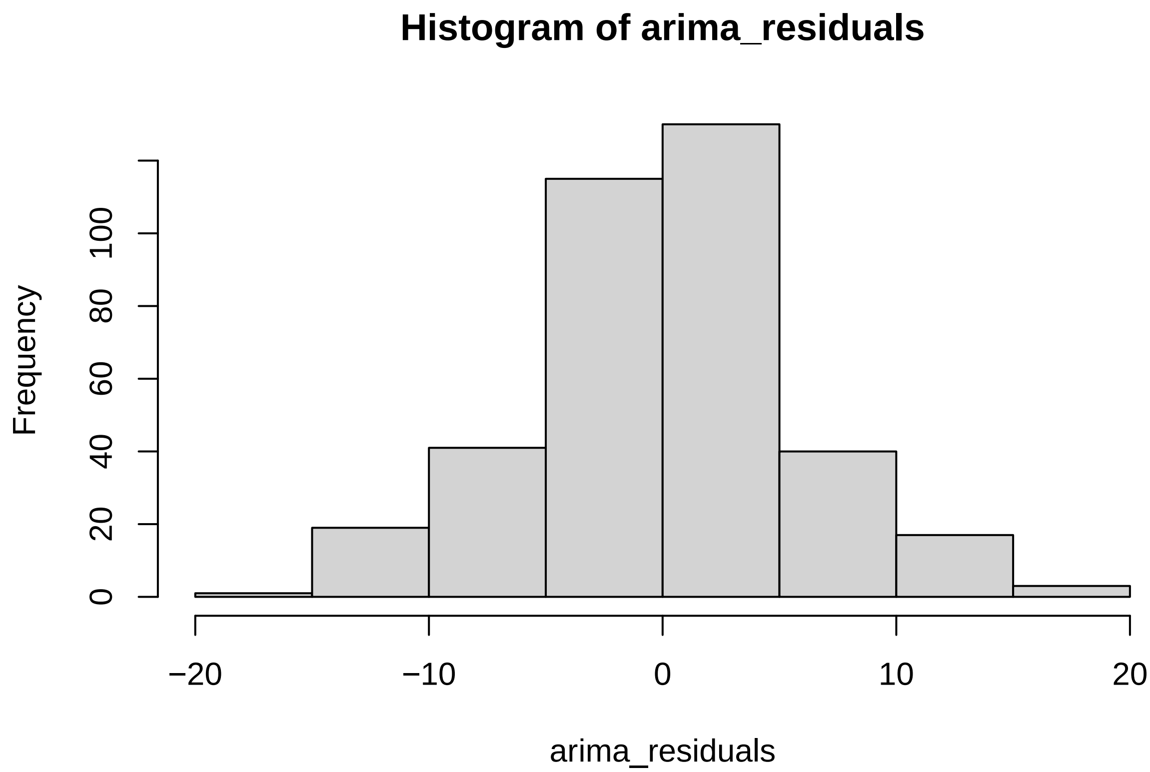

```
pacf(as.vector(arima_residuals), lag.max = 50)
```




```
qqnorm(arima_residuals)
qqline(arima_residuals)
```




```
hist(arima_residuals)
```



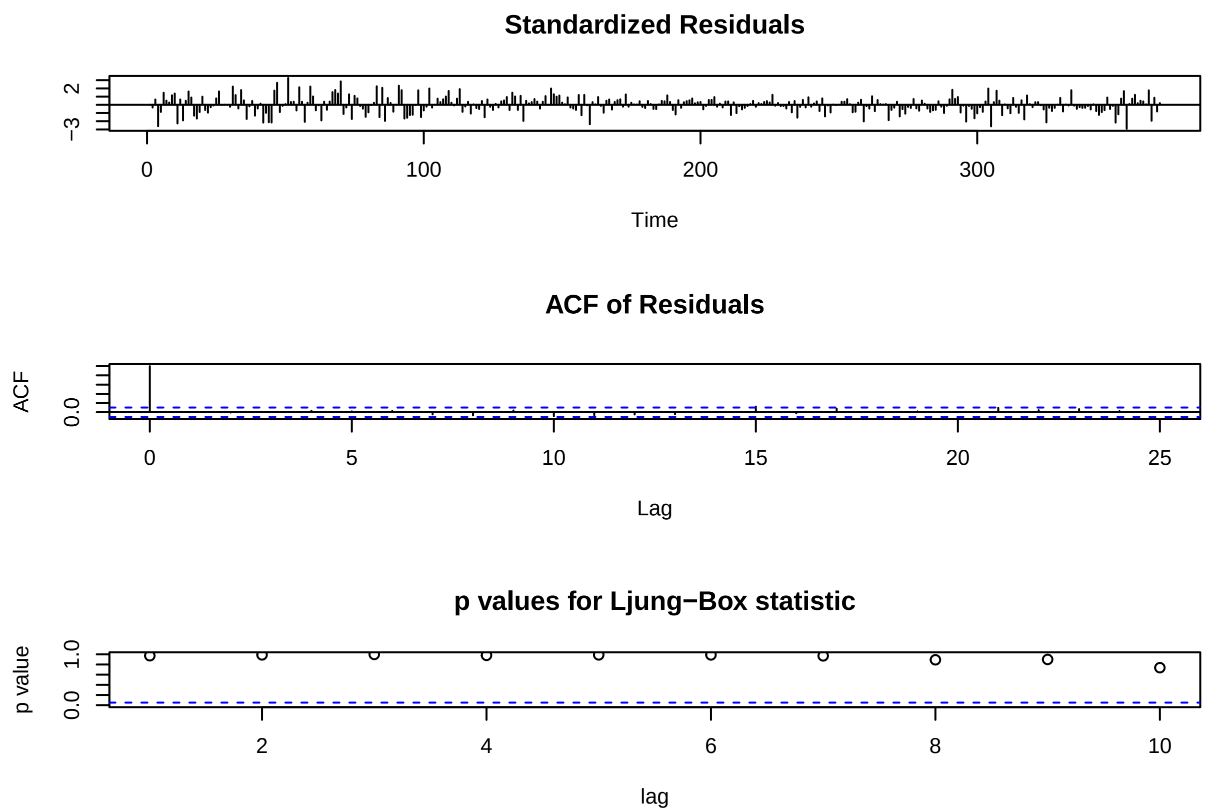

```r
print(shapiro.test(arima_residuals))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  arima_residuals
## W = 0.99101, p-value = 0.02506
```

```r
ljung_box_test <- Box.test(arima_residuals, lag = 10, type = "Ljung-Box")
ljung_box_test
```

```
##
##  Box-Ljung test
##
## data:  arima_residuals
## X-squared = 6.8724, df = 10, p-value = 0.7374
```

```r
tsdiag(model)
```

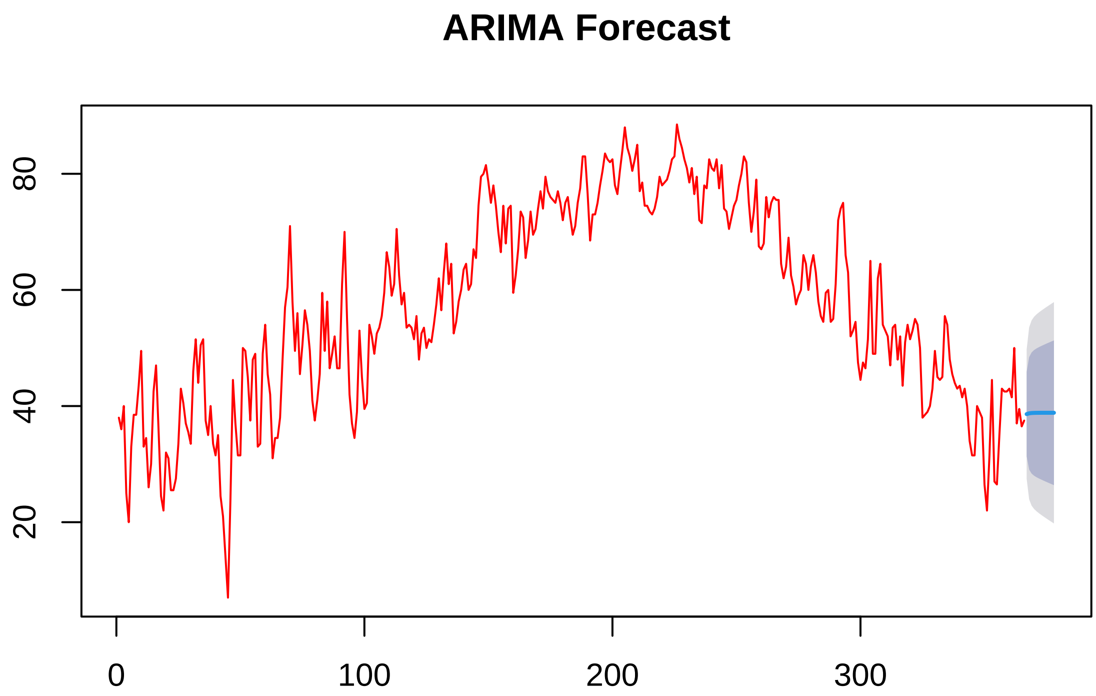


Prediction

```r
forecast_best_model <- forecast(model, h = 12)
forecast_best_model
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 367       38.61261 31.33317 45.89205 27.47967 49.74556
## 368       38.74589 29.06627 48.42550 23.94220 53.54958
## 369       38.80185 28.36396 49.23974 22.83848 54.76523
## 370       38.82535 28.00095 49.64975 22.27086 55.37985
## 371       38.83522 27.74132 49.92912 21.86857 55.80188
## 372       38.83937 27.52015 50.15858 21.52812 56.15061
## 373       38.84111 27.31571 50.36650 21.21453 56.46768
## 374       38.84184 27.11982 50.56386 20.91456 56.76912
## 375       38.84214 26.92917 50.75512 20.62283 57.06146
## 376       38.84227 26.74236 50.94218 20.33706 57.34749
## 377       38.84233 26.55873 51.12592 20.05619 57.62846
## 378       38.84235 26.37795 51.30675 19.77969 57.90501
```

```r
plot(forecast_best_model, col="red", main = "ARIMA Forecast")
```


ARIMA Forecast
80
60
40
20
0
100
200
300