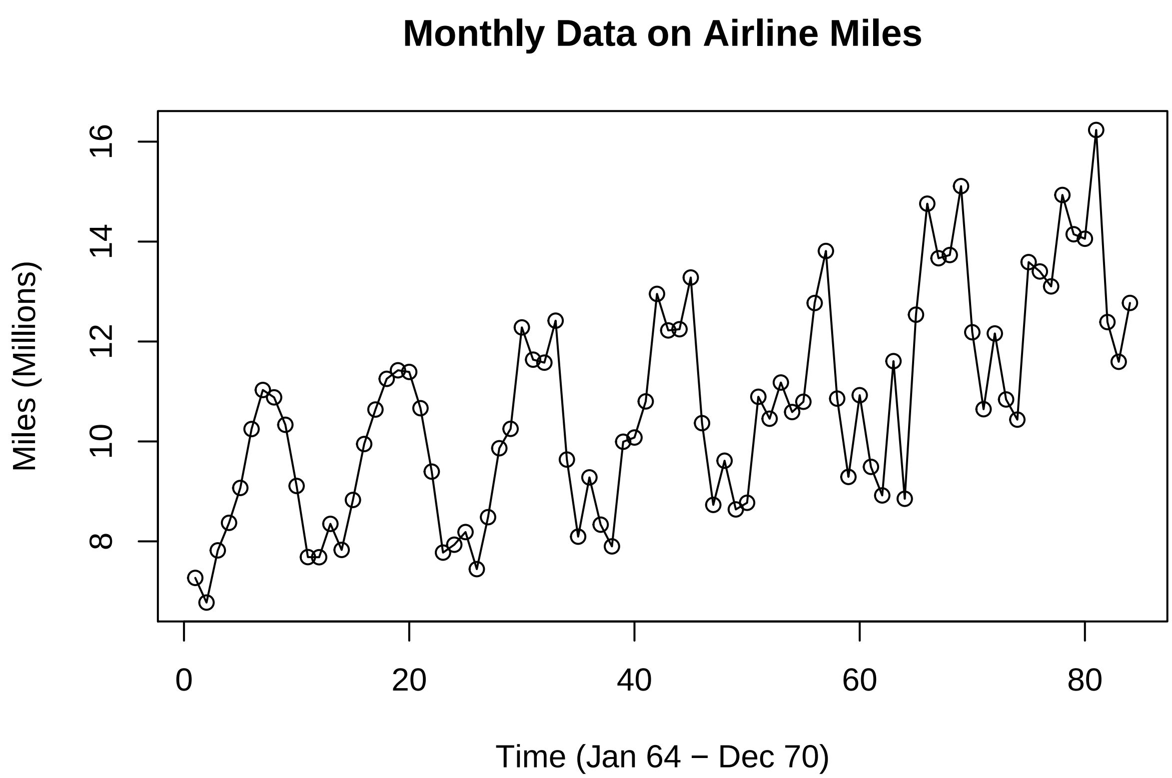
DSC 475 - Project 1

Hao Ma

Fall 2020

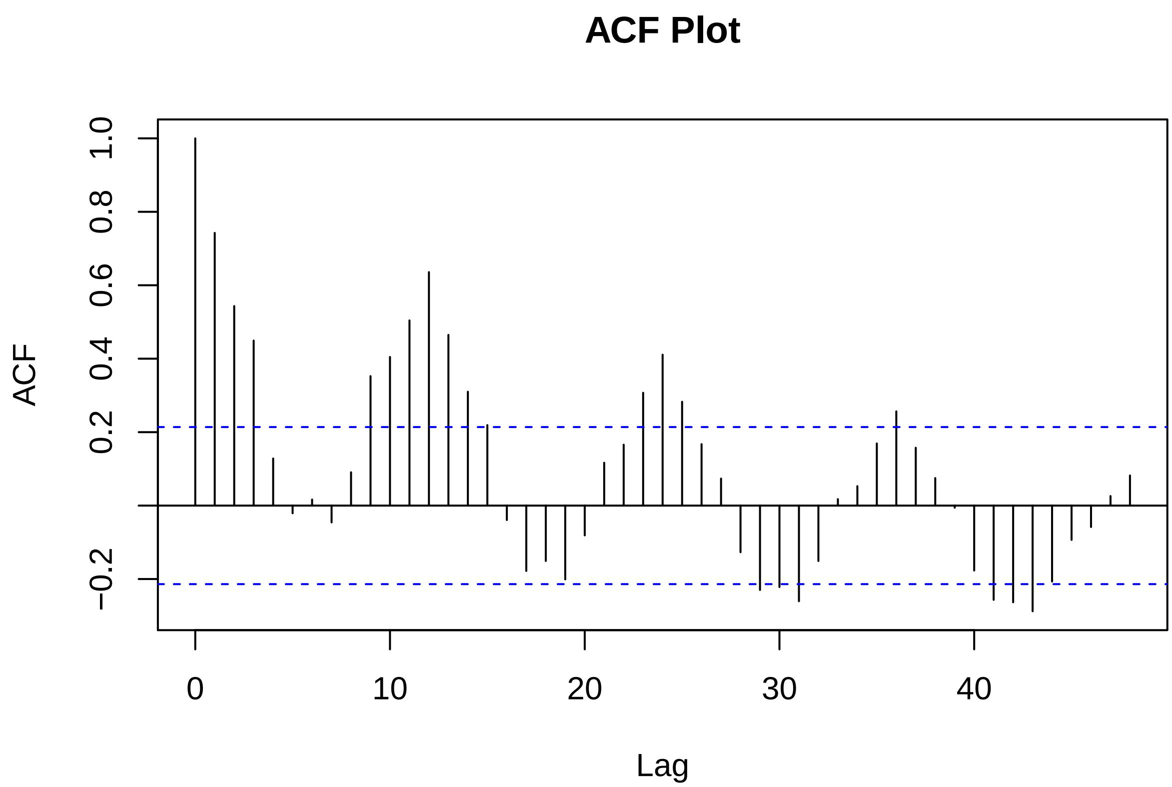
1.

```r
library(readxl)
data <- read_excel("Project1_DataSet.xlsx")
plot(data$`Miles, in Millions`, main="Monthly Data on Airline Miles",
     xlab="Time (Jan 64 - Dec 70)", ylab="Miles (Millions)")
lines(data$`Miles, in Millions`)
```



2.

```r
acf(data$`Miles, in Millions`, main="ACF Plot", lag.max=48)
```

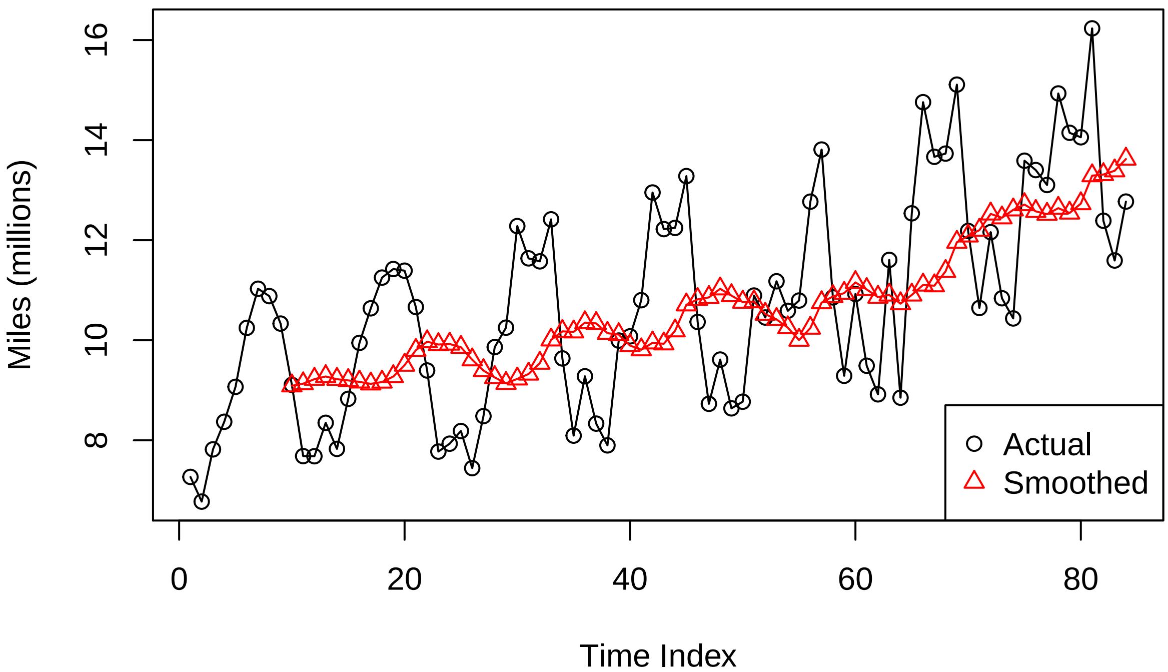


The seasonal period is 12 months.

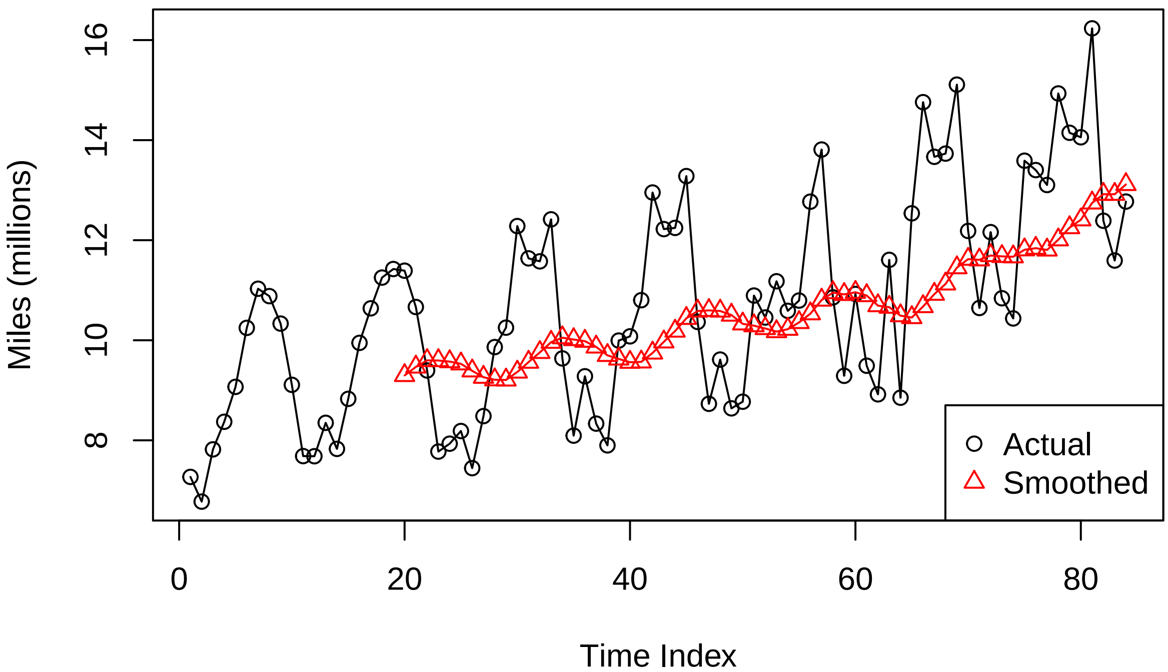
3.

```r
# A function to compute simple moving average with a given window length
# and a univariate dataset. Then plot the data along with the SMA results.
SMA <- function(N, data){
  T_ <- nrow(data)
  t <- N:T_
  M <- c()
  for(i in t){
    M <- c(M, sum(data[,2][(i-N+1):i])/N)
  }
  plot(data[,2], xlab="Time Index", ylab="Miles (millions)")
  lines(data[,2])
  points(t, M, pch=2, col="red")
  lines(t, M, pch=2, col="red")
  legend("bottomright", c("Actual","Smoothed"),
         col=c("black","red"), pch=c(1,2))
}
```

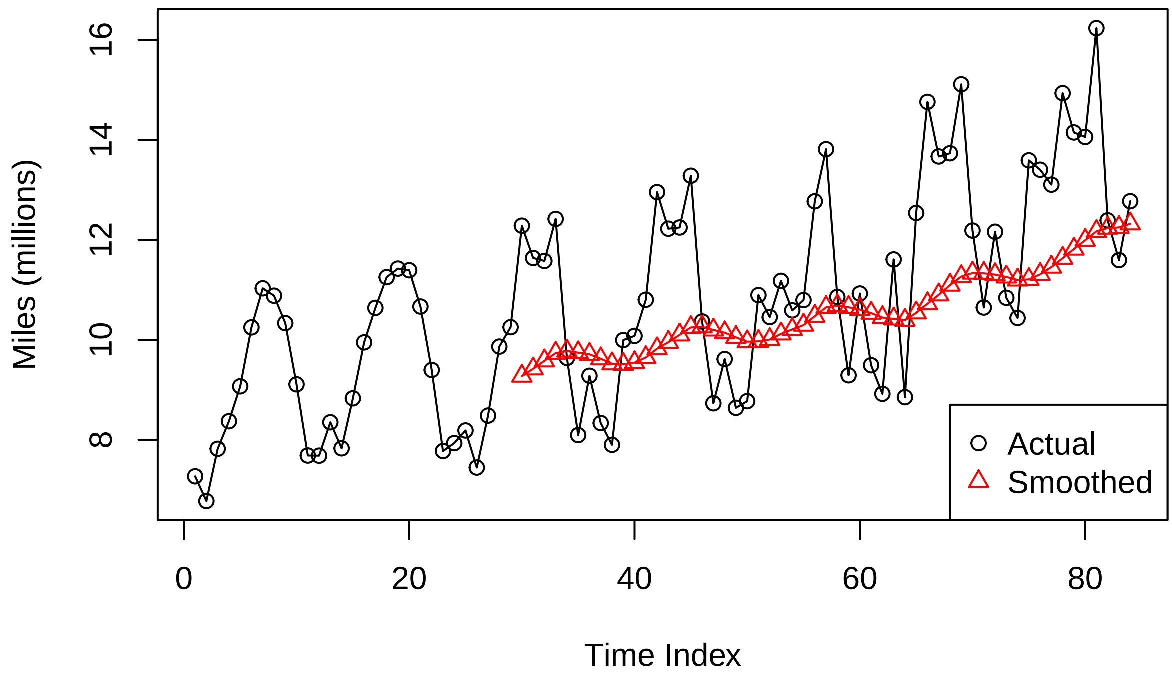
```r
SMA(10, as.data.frame(data))
```



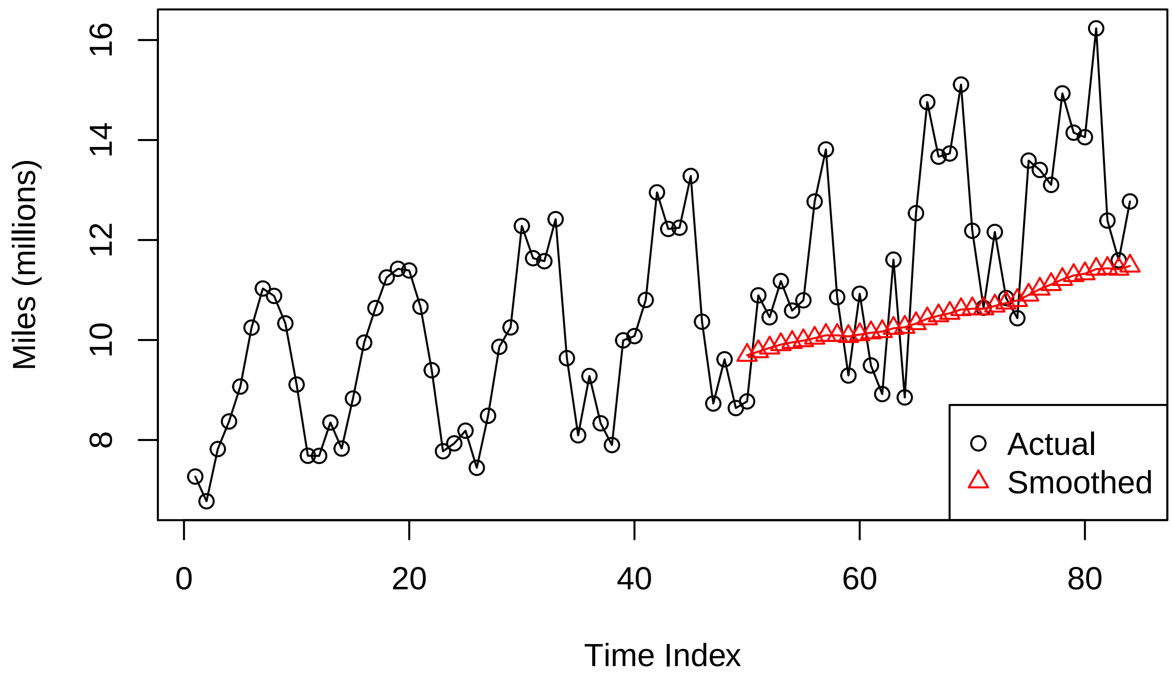

```
SMA(20, as.data.frame(data))
```




```r
SMA(30, as.data.frame(data))
```




```
SMA(50, as.data.frame(data))
```



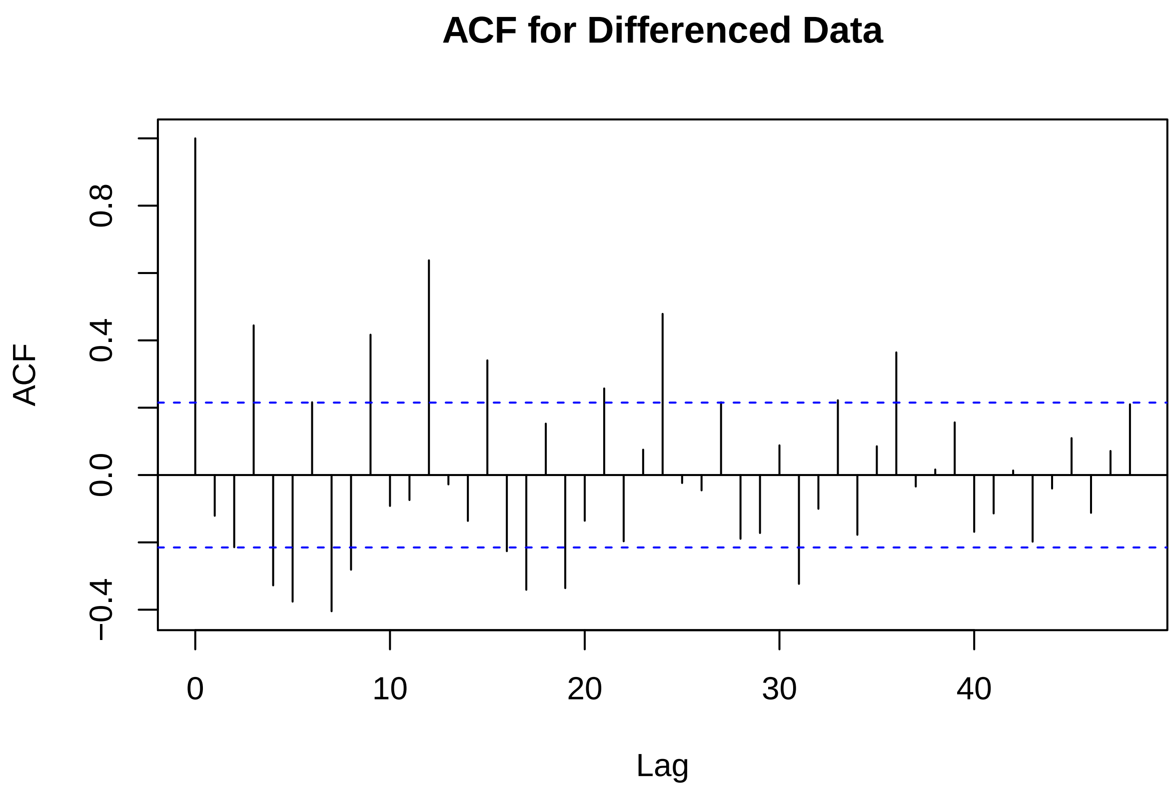
I chose 20 as the window length based on the plots shown above. We do not want the smoothed curve be "too smooth" like a straight line or "too accurate" like the original data. Somewhere in the middle is sufficient to discover the trend.

4.

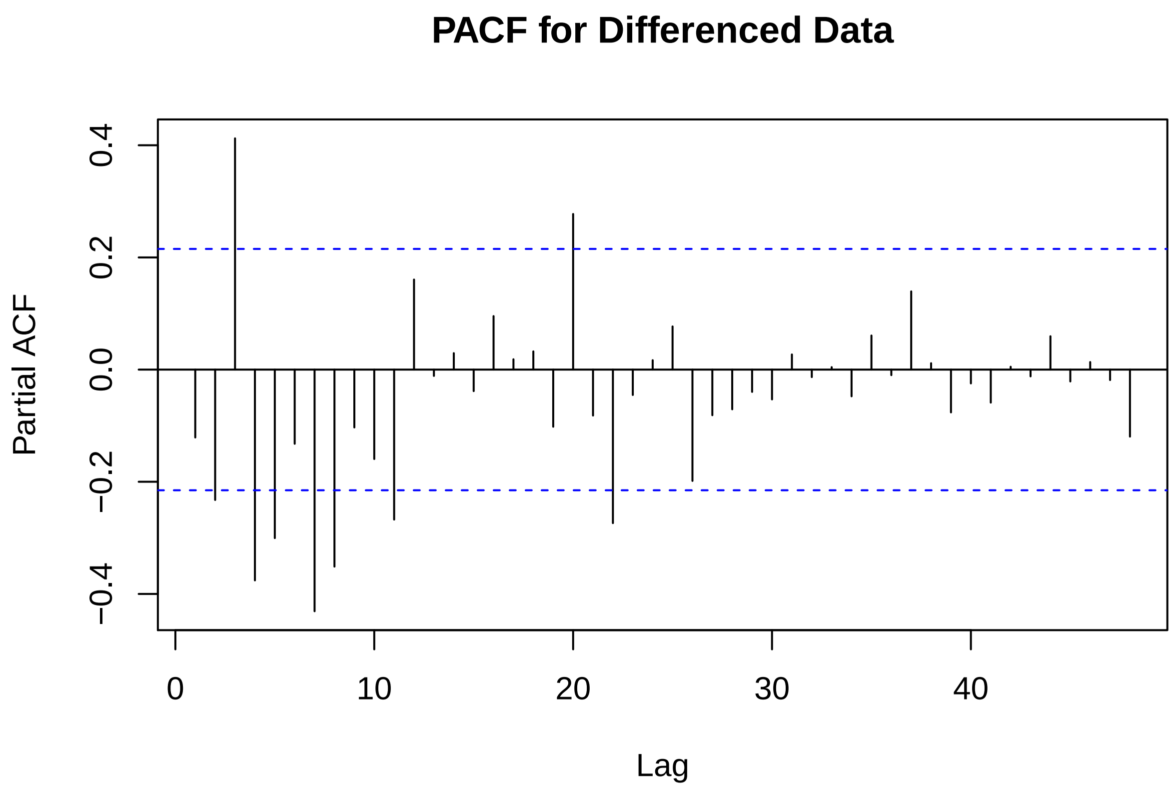
It has an increasing trend.

5.

```r
# Then compute first difference to remove trend
diffed <- diff(data$`Miles, in Millions`, differences=1)
acf(diffed, lag.max=48, main="ACF for Differenced Data")
```



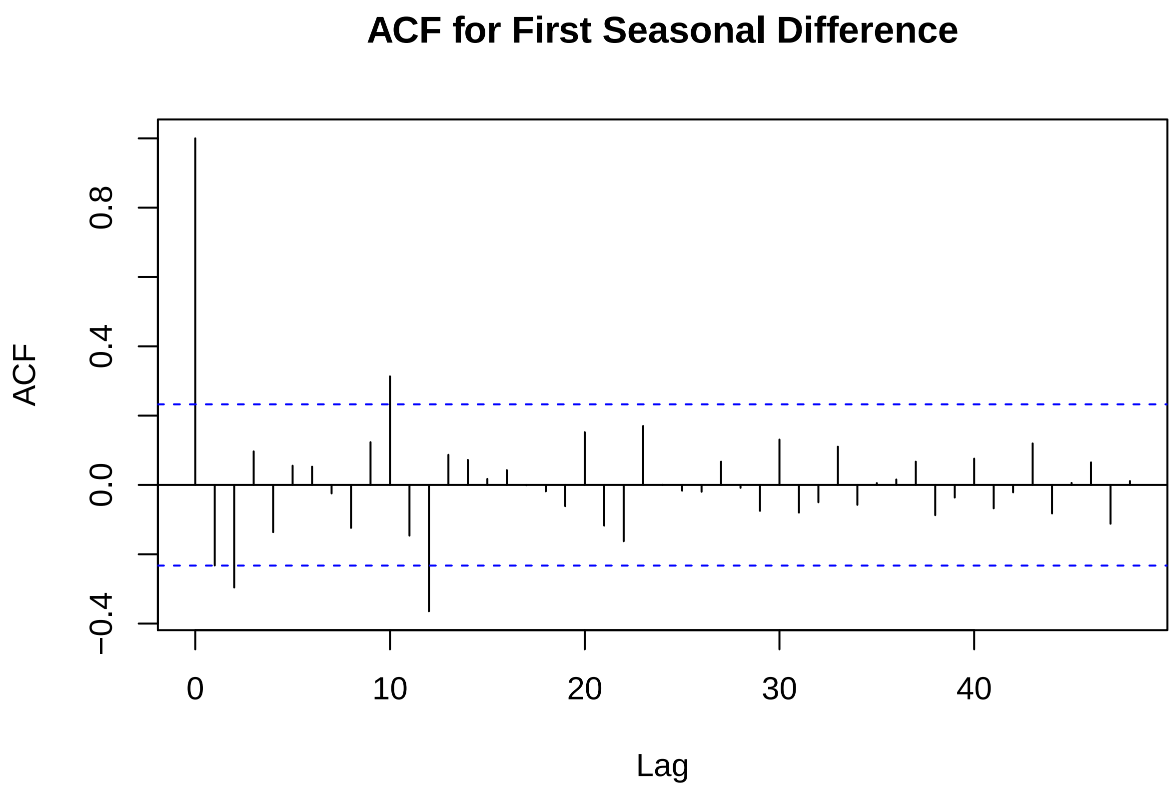

```
pacf(diffed, lag.max=48, main="PACF for Differenced Data")
```



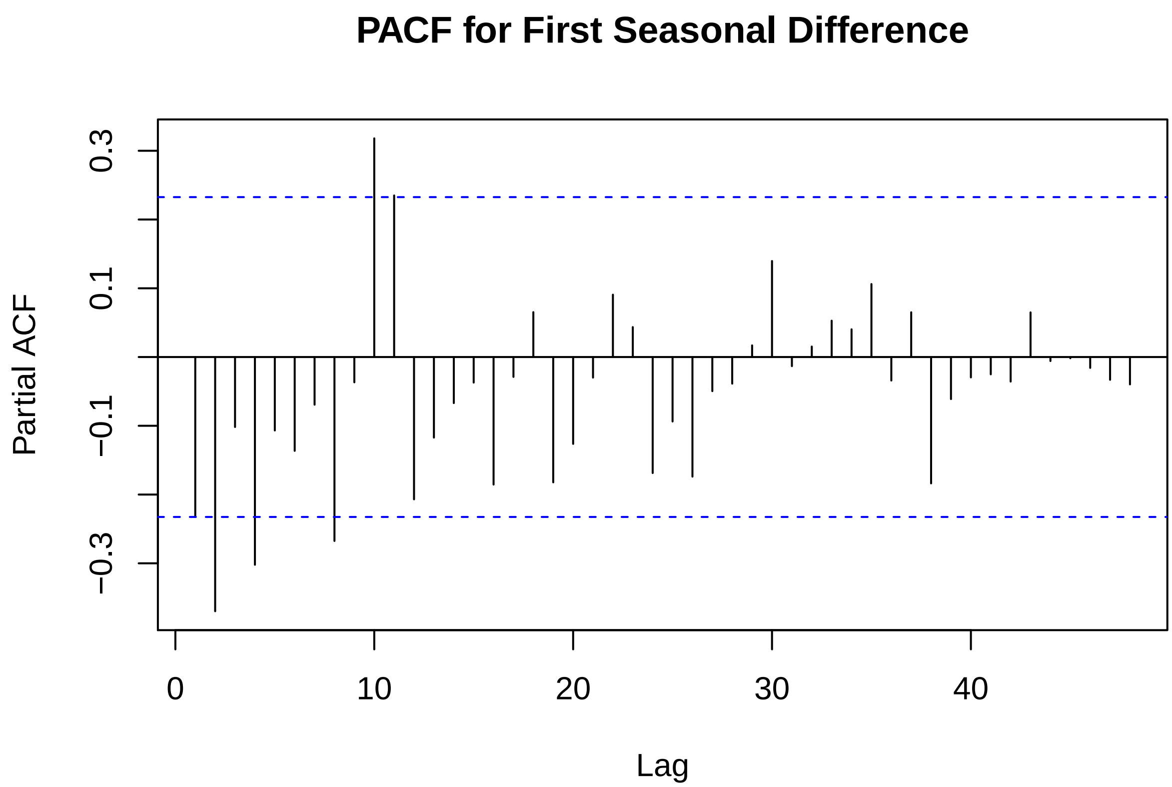
There are many significant lags due to the seasonality within each year (non-stationarity).

6.

```
# Compute first seasonal difference to remove seasonality as well with 12 as
# the seasonal period
seasonal_diff <- diff(diffed, lag=12, differneces=1)
acf(seasonal_diff, lag.max=48, main="ACF for First Seasonal Difference")
```




```r
pacf(seasonal_diff, lag.max=48, main="PACF for First Seasonal Difference")
```



The number of significant lags decreases and they are all within the first year.

7.

Based on the trend, seasonality and auto-correlation plots, we will develop a SARIMA model using the `auto.arima()` function in the `forecast` library. Set $d = 1$, $D = 1$ and vary p, q, P, Q each over the range 0 to 3 to find the best model based on BIC.

```
library(forecast)
```

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

```
# Use the first 6 years of monthly data to create a time series object
training <- ts(data[1:72,2], start=c(1964, 1), frequency=12)
# Then search for the best combination of parameters using `auto.arima()`
model <- auto.arima(training, d=1, D=1, max.p=3, max.q=3, max.P=3, max.Q=3,
           start.p=0, start.q=0, start.P=0, start.Q=0, ic="bic", trace=T)
```

```
##
##  ARIMA(0,1,0)(0,1,0)[12]                    : 168.9797
##  ARIMA(0,1,0)(0,1,0)[12]                    : 168.9797
##  ARIMA(1,1,0)(1,1,0)[12]                    : 169.5404
##  ARIMA(0,1,1)(0,1,1)[12]                    : 158.9514
##  ARIMA(0,1,1)(0,1,0)[12]                    : 164.4505
##  ARIMA(0,1,1)(1,1,1)[12]                    : 162.3086
##  ARIMA(0,1,1)(0,1,2)[12]                    : 162.0796
```

```
##  ARIMA(0,1,1)(1,1,0)[12]                    : 158.6684
##  ARIMA(0,1,1)(2,1,0)[12]                    : 162.1385
##  ARIMA(0,1,1)(2,1,1)[12]                    : 165.9725
##  ARIMA(0,1,0)(1,1,0)[12]                    : 169.4882
##  ARIMA(1,1,1)(1,1,0)[12]                    : 159.3504
##  ARIMA(0,1,2)(1,1,0)[12]                    : 157.279
##  ARIMA(0,1,2)(0,1,0)[12]                    : Inf
##  ARIMA(0,1,2)(2,1,0)[12]                    : 161.3522
##  ARIMA(0,1,2)(1,1,1)[12]                    : 161.3535
##  ARIMA(0,1,2)(0,1,1)[12]                    : 157.5353
##  ARIMA(0,1,2)(2,1,1)[12]                    : Inf
##  ARIMA(1,1,2)(1,1,0)[12]                    : 160.2461
##  ARIMA(0,1,3)(1,1,0)[12]                    : 160.4903
##  ARIMA(1,1,3)(1,1,0)[12]                    : 164.3235
##
##  Best model: ARIMA(0,1,2)(1,1,0)[12]
```

The best model is: $ARIMA(0,1,2)(1,1,0)_{12}$.

8.

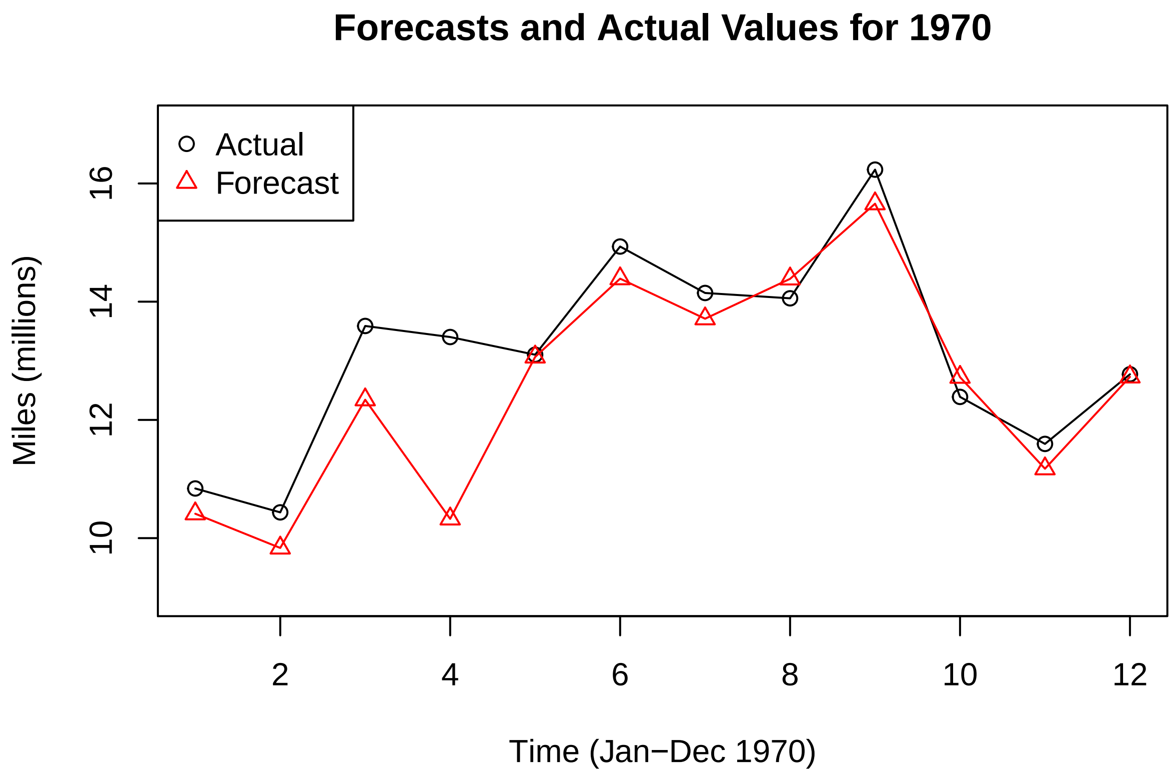
Use the model above to forecast for the year 1970 (12 forecasts) using the function `forecast()`.

```
my_forecast <- forecast(model, h=12)
my_forecast
```

```
##          Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## Jan 1970      10.412957  9.391576 11.43434  8.850890 11.97502
## Feb 1970       9.834015  8.666566 11.00146  8.048556 11.61947
## Mar 1970      12.341312 11.171186 13.51144 10.551758 14.13087
## Apr 1970      10.325723  9.152924 11.49852  8.532082 12.11936
## May 1970      13.065388 11.889924 14.24085 11.267671 14.86311
## Jun 1970      14.389305 13.211180 15.56743 12.587519 16.19109
## Jul 1970      13.711707 12.530928 14.89248 11.905862 15.51755
## Aug 1970      14.386135 13.202709 15.56956 12.576241 16.19603
## Sep 1970      15.657544 14.471476 16.84361 13.843609 17.47148
## Oct 1970      12.722966 11.534262 13.91167 10.905000 14.54093
## Nov 1970      11.174346  9.983011 12.36568  9.352357 12.99634
## Dec 1970      12.728338 11.534379 13.92230 10.902335 14.55434
```

The forecasts and prediction intervals are shown above.

```
# Compare the mean forecasts with the actual data
actual <- data$`Miles, in Millions`[73:84]
forecasts <- as.numeric(my_forecast$mean)
plot(actual, main="Forecasts and Actual Values for 1970",
     ylim=c(9,17), xlab="Time (Jan-Dec 1970)", ylab="Miles (millions)")
lines(actual)
points(forecasts, col="red", pch=2)
lines(forecasts, col="red", pch=2)
legend("topleft", c("Actual","Forecast"),
       col=c("black","red"), pch=c(1,2))
```



The monthly trend and values of the forecast are close to the actual data, except for the time period of April 1970 where the decrease is too steep. This should be acceptable.