STA 137 Project: US Monthly Prescription Costs

Chad Pickering, Sierra Tevlin, Janice Luong

March 8, 2016

```r
library(TSA)
```

Description of the data

```
## Loading required package: leaps
```

```
## Loading required package: locfit
```

```
## locfit 1.5-9.1    2013-03-22
```

```
## Loading required package: mgcv
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-9. For overview type 'help("mgcv-package")'.
```

```
## Loading required package: tseries
```

```
##
## Attaching package: 'TSA'
```

```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

```
## The following object is masked from 'package:utils':
##
##     tar
```

```r
library(forecast)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

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

```
## Loading required package: timeDate
```

```
##
## Attaching package: 'timeDate'
```

```
## The following objects are masked from 'package:TSA':
##
##     kurtosis, skewness
```

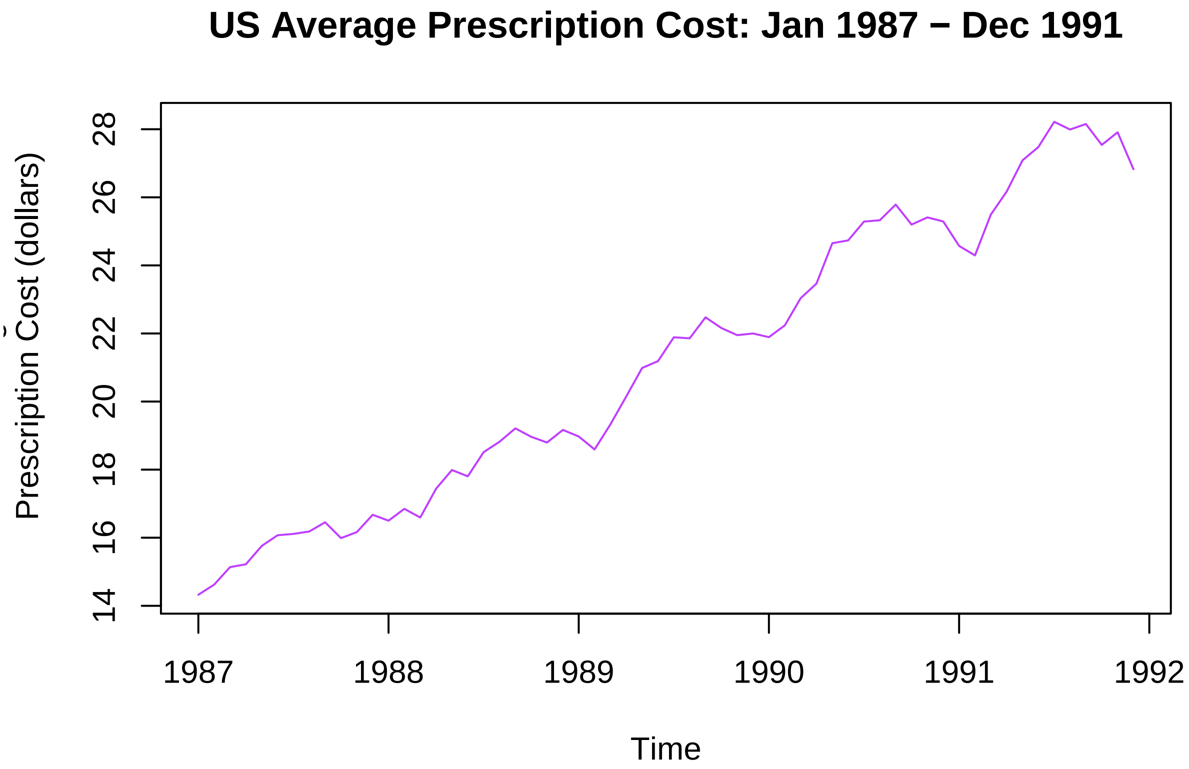
```
## This is forecast 6.2
```

```
##
## Attaching package: 'forecast'
```

```
## The following objects are masked from 'package:TSA':
##
##     fitted.Arima, plot.Arima
```

```
## The following object is masked from 'package:nlme':
##
##     getResponse
```

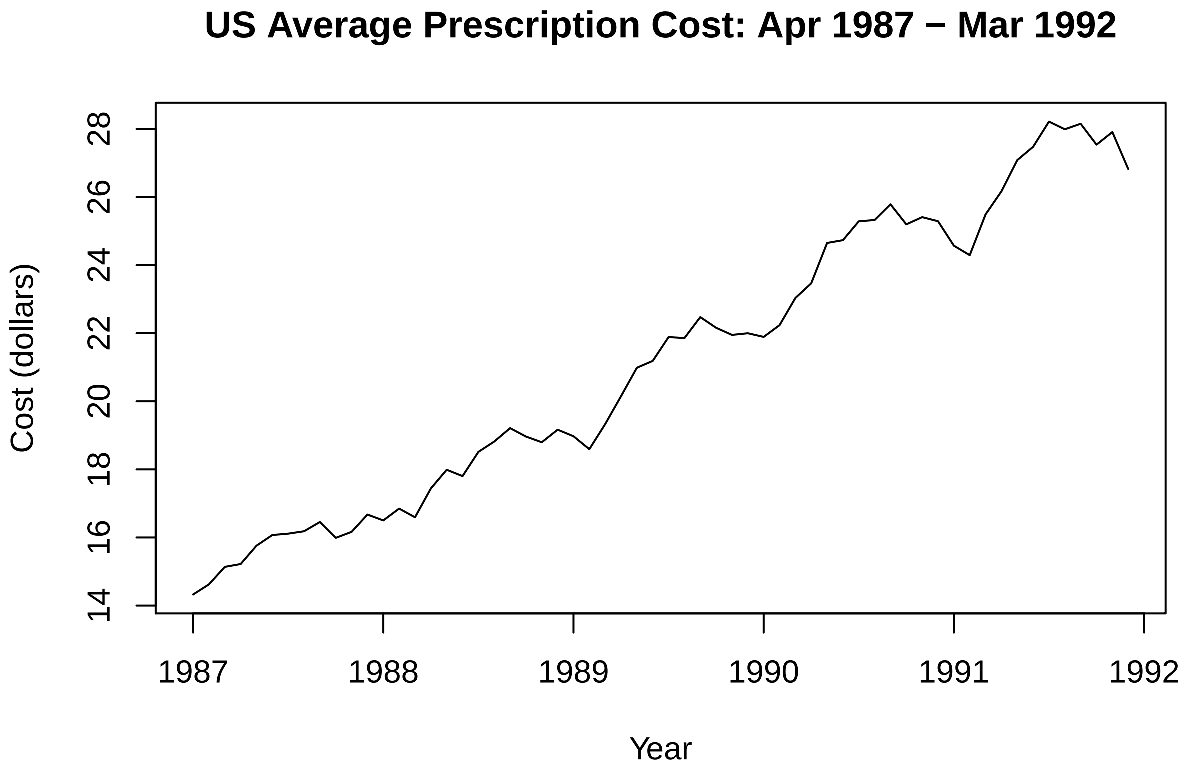
```r
data(prescrip)

prescrip.part = window(prescrip, start = c(1987,1), end = c(1991,12))

t <- as.vector(time(prescrip.part))
x <- as.vector(prescrip.part)

plot(t, x, type = "l",
     main = "US Average Prescription Cost: Jan 1987 - Dec 1991", xlab = "Time", ylab = "Average
     Prescription Cost (dollars)", col = "darkorchid1")
```



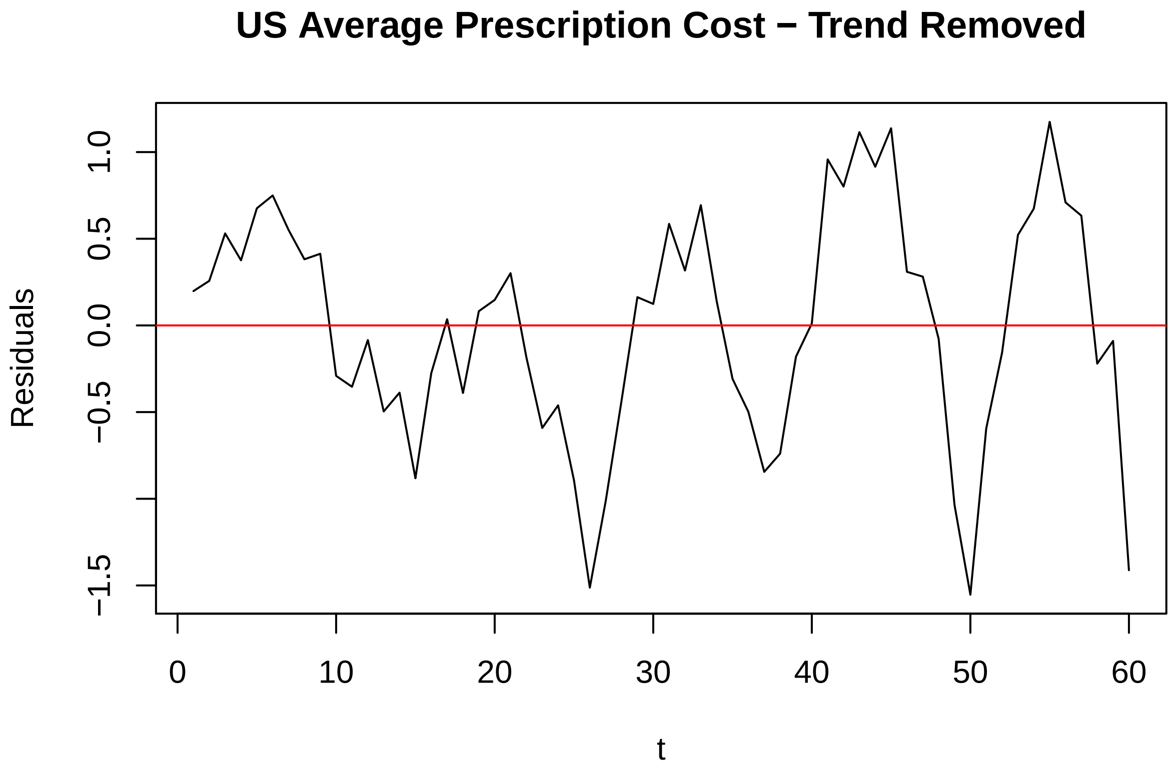
Our adjusted dataset (five complete years) contains the average prescription cost in dollars from January 1987 - December 1991 measured monthly - this was attained from the TSA library. Our adjustment excludes all measurements from 1986 and 1992. Through visual inspection of the data, we decided that a transformation was not necessary, and no outliers are present.

Deterministic components Pre-estimate m_t:

```r
n = 60
t = 1:n
trend.fit <- lm(x ~ t)

# Pre-estimate
plot(prescrip.part, main = "US Average Prescription Cost: Apr 1987 - Mar 1992",
     xlab = "Year", ylab = "Cost (dollars)")
lines(t, fitted(trend.fit), col="red")
```



```
# Trend removed
y <- residuals(trend.fit)
plot(t, y, type="l", main = "US Average Prescription Cost - Trend Removed",
     ylab = "Residuals")
abline(h=0, col="red")
```

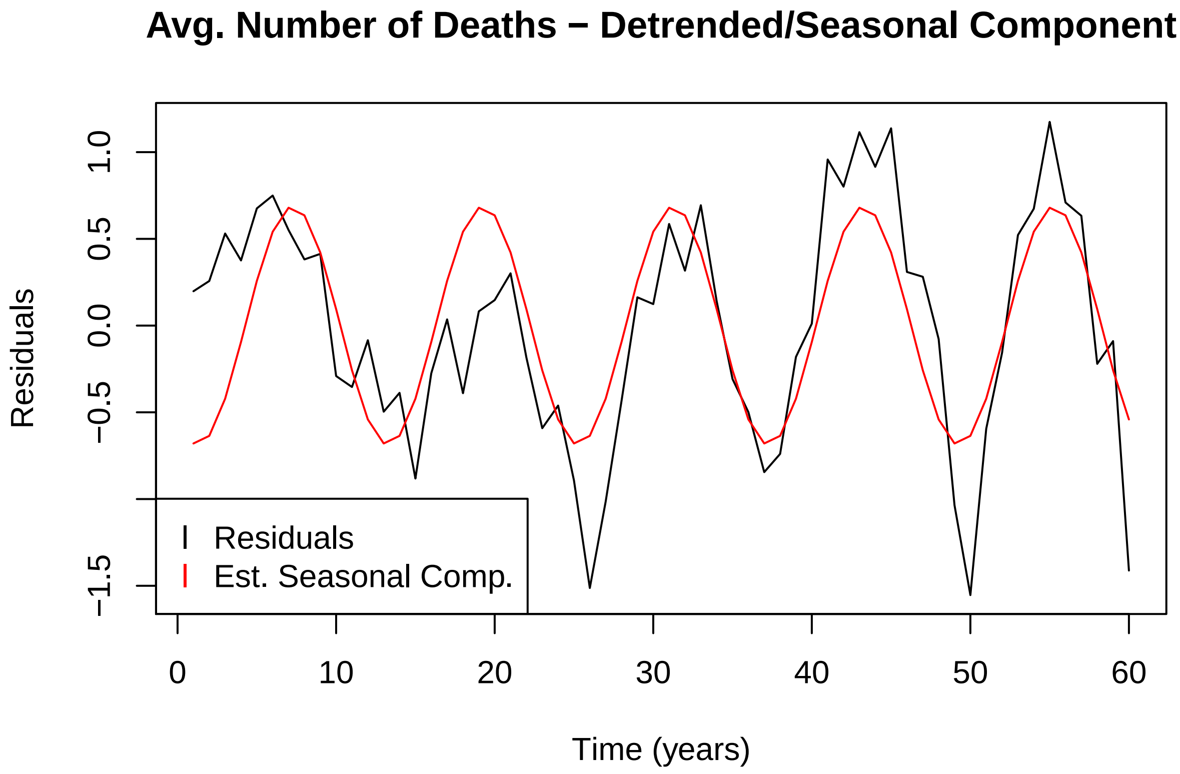


Seasonality component s_t:

```
n <- length(t)
t <- 1:length(y)
t <- (t) / n

d <- 12 # number of time points in each season
n.harm <- 5 # set to [d/2]
harm <- matrix(nrow = length(t), ncol = 2*n.harm)

for(i in 1:n.harm)
{
  harm[,i*2-1] = sin(n/d * i * 2*pi*t)
  harm[,i*2] = cos(n/d * i * 2*pi*t)
}

colnames(harm) <- paste0(c("sin", "cos"),
                         rep(1:n.harm, each = 2))

# fit sines and cosines
dat <- data.frame(y, harm)
fit <- lm(y ~ ., data = dat)

# setup the full model and the model with only an intercept
full <- lm(y ~ ., data = dat)
reduced <- lm(y ~ 1, data = dat)
```

```r
# stepwise regression starting with the full model
fit.back <- step(full,
                 scope = formula(reduced), direction = "both")
```

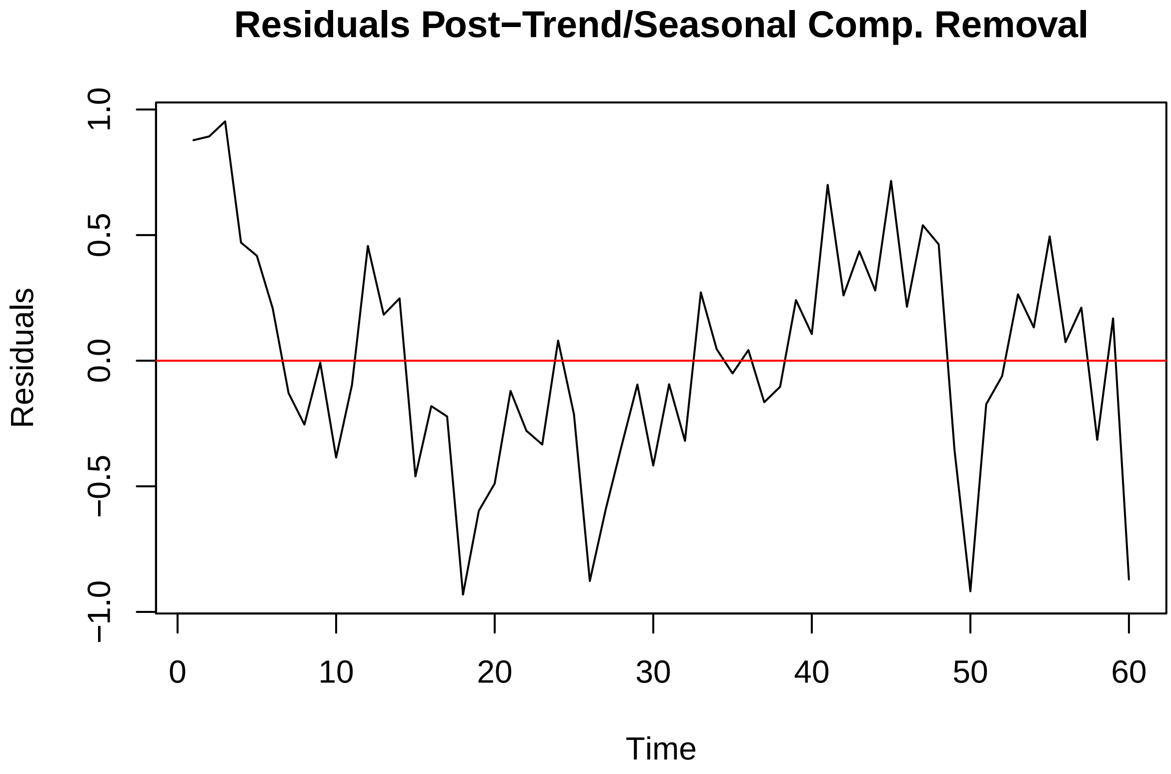
```r
t <- 1:n

# plot the estimated seasonal components
plot(t, y, type="l", col="black",
     ylab = "Residuals", xlab = "Time (years)",
     main = "Avg. Number of Deaths - Detrended/Seasonal Component")
lines(t, fitted(fit.back), col="red")
legend("bottomleft", c("Residuals", "Est. Seasonal Comp."),
       pch = "l", col = c("black", "red"))
```



We set n.harm equal to 5 because we only have 60 data points so any n.harm value higher than 5 causes the sin6 component of our harmonics to be poorly estimated. This caused our seasonal components to vary over the year. Looking at the actual estimate for sin6 after using stepwise regression, it is very large (8.154e+12). Setting n.harm = 5 will lead to less harmonic components, but it does an adequate job considering that we only have n = 60.

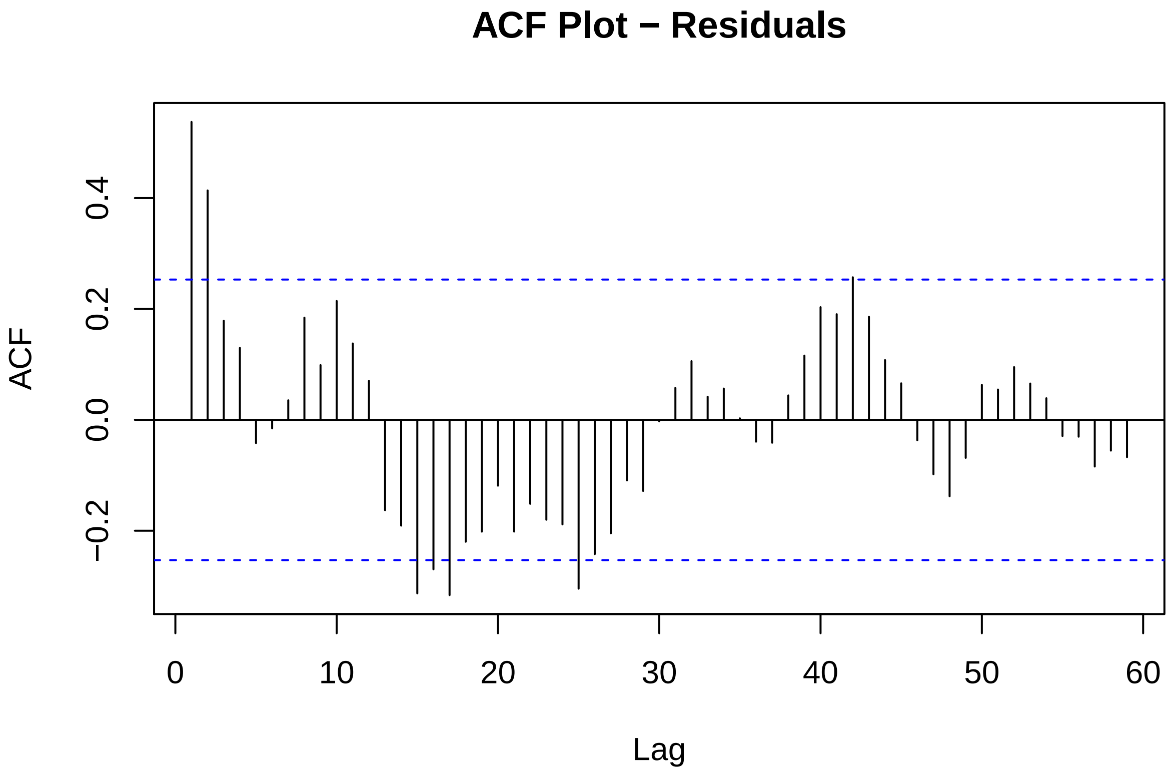
Residuals Y_t:

```r
# plot the residuals after seasonal component is removed
ts.plot(residuals(fit.back),
        main = "Residuals Post-Trend/Seasonal Comp. Removal",
        ylab = "Residuals", xlab = "Time")
abline(h=0, col="red")
```



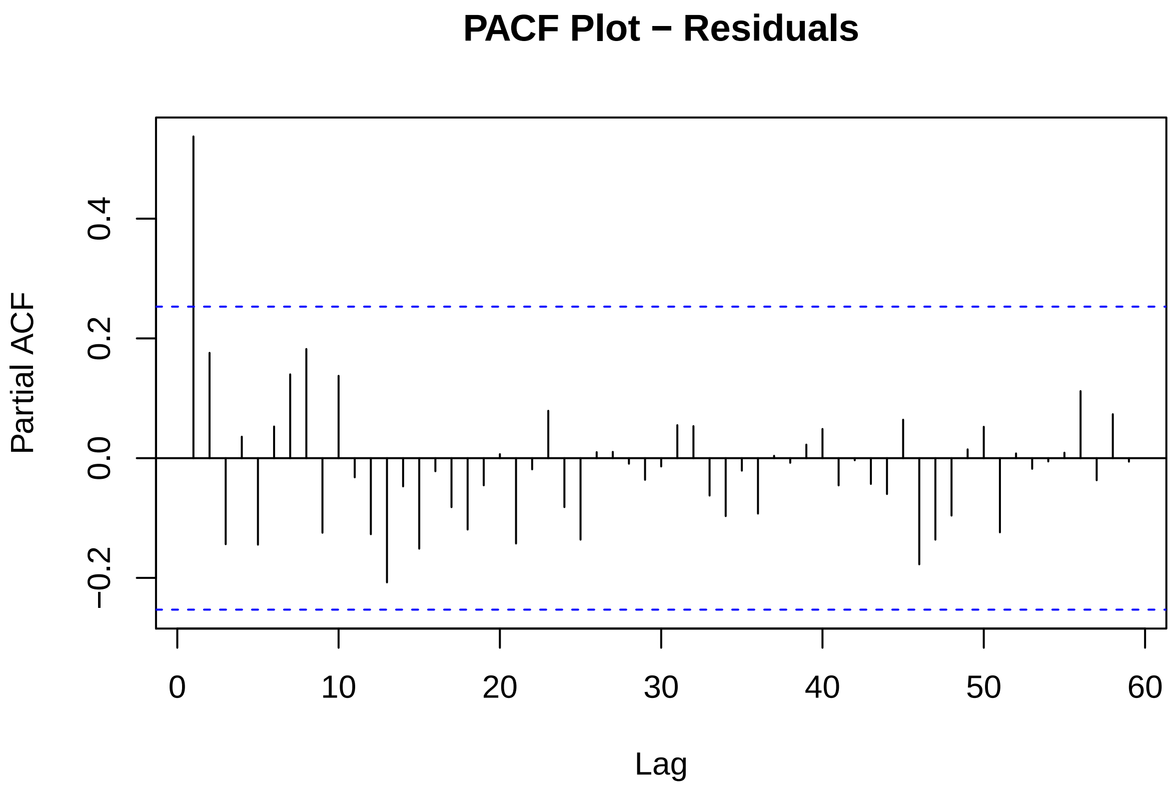
We used a first order polynomial to estimate the trend component. After removing the trend component, we observed the resulting residuals and concluded that there is a seasonal component that needs to be removed. We used sum of harmonics because using the moving average method did not completely remove the seasonal component.

```
res <- residuals(fit.back)
acf(res, lag.max = 60, main = "ACF Plot - Residuals")
```



Time series model

```
pacf(res, lag.max = 60, main = "PACF Plot - Residuals")
```



Based on the ACF and PACF plots, we should consider an AR(1) model because the ACF plot trails off to zero and the PACF plot drops off sharply after lag 1. We also noticed that our ACF plot does not start at 0, so it does not show that the autocorrelation at 0 is 1.

```
arima_fit <- auto.arima(res, stepwise = FALSE)
arima_fit
```

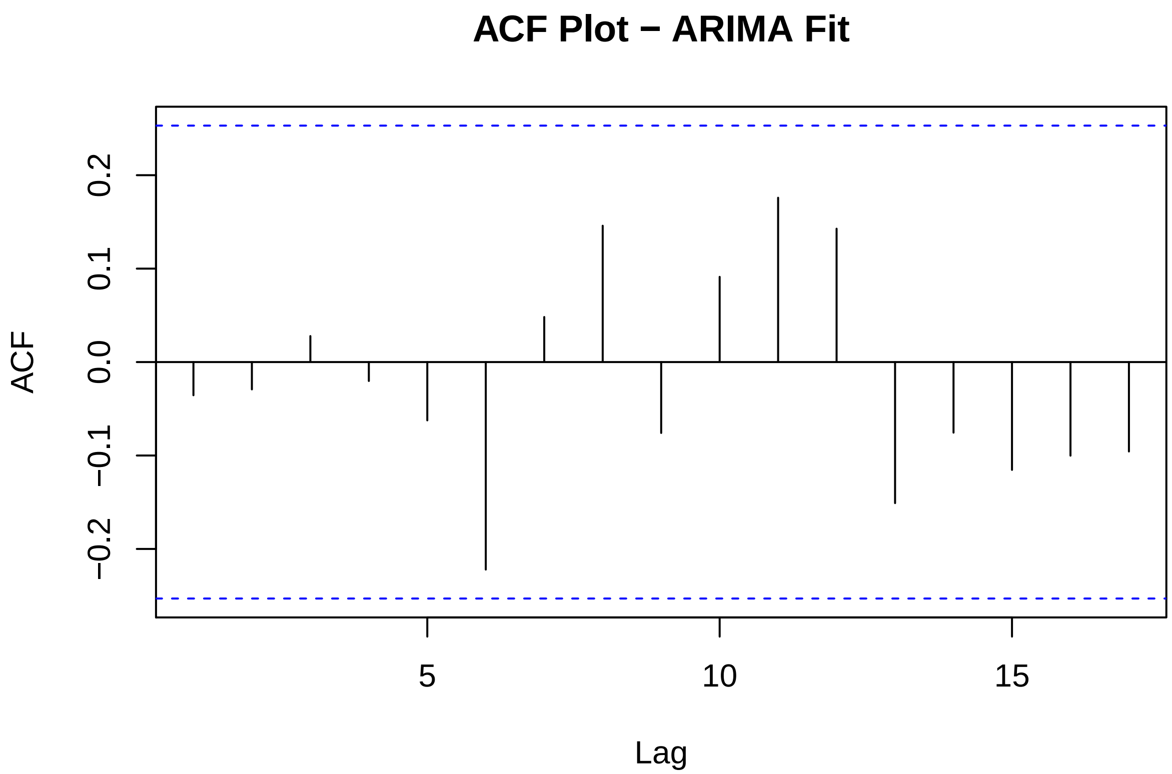
```
## Series: res
## ARIMA(2,0,1) with zero mean
##
## Coefficients:
##           ar1     ar2     ma1
##       -0.3064  0.6768  0.9210
## s.e.   0.1045  0.1019  0.0608
##
## sigma^2 estimated as 0.1091:  log likelihood=-19.45
## AIC=46.89   AICc=47.62   BIC=55.27
```

Based on the results of auto.arima, we will consider an ARMA(2,1) model instead because in the previous PACF plot, there are signs of a moving average component based on clear oscillations throughout.

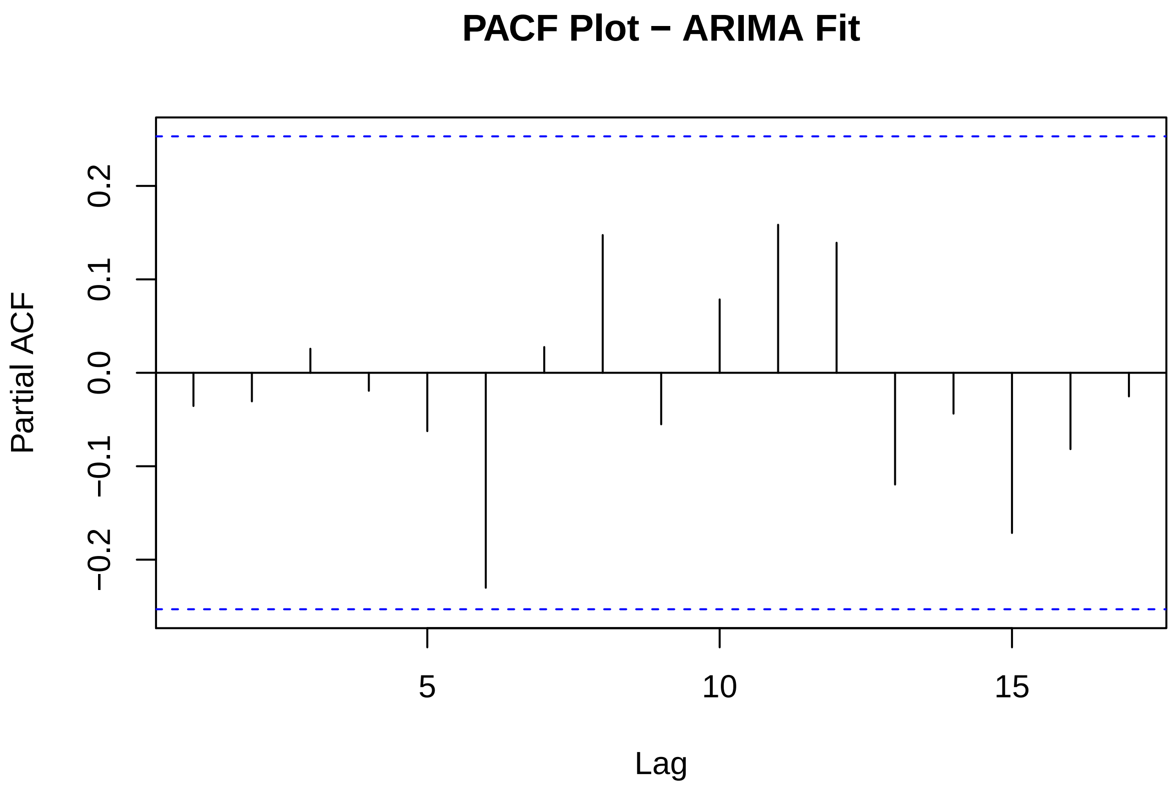
The auto.arima function is more reliable than the ACF and PACF plots because the auto.arima function checks the AIC value, so it will take the model with the smallest AIC value ("best" model). Using ACF and PACF is not as reliable because it is a visualization and we can only guess the model through intuition / eyeballing.

Our ARMA(2,1) model is: $X_t + 0.3064X_{t-1} - 0.6768X_{t-2} = Z_t + 0.9210Z_{t-1}$

```r
acf(resid(arima_fit), na.action = na.pass, main = "ACF Plot - ARIMA Fit")
```



```r
pacf(resid(arima_fit), na.action = na.pass, main = "PACF Plot - ARIMA Fit")
```

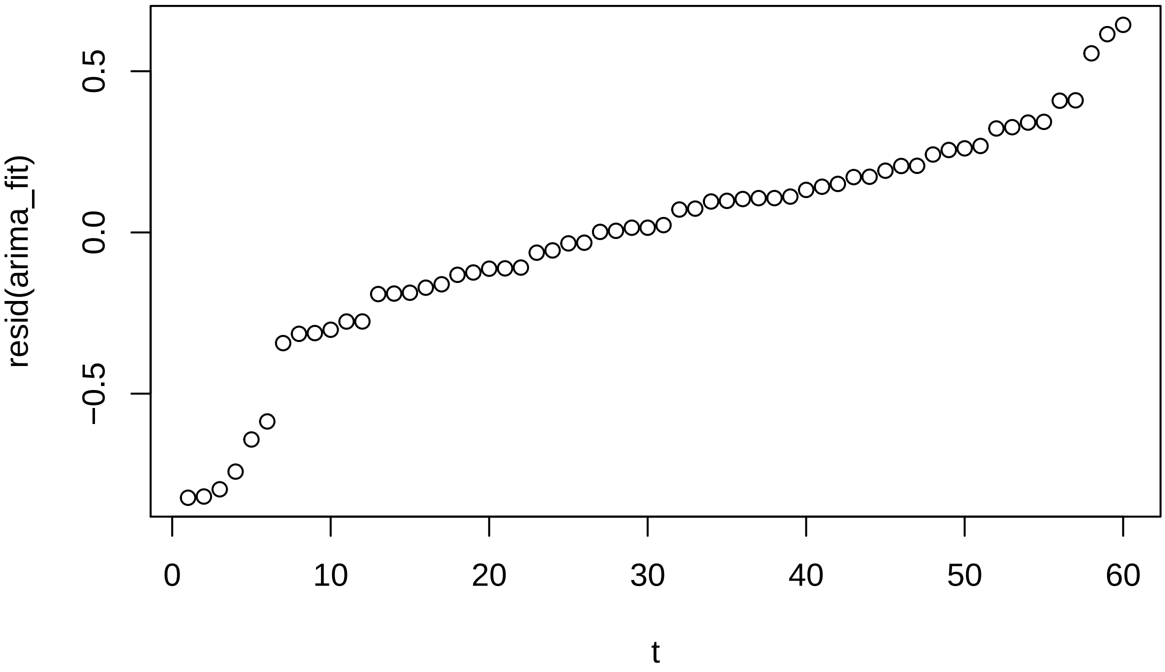


```
Box.test(resid(arima_fit), type="Ljung-Box", lag = min(2*d, floor(n/5)))
```

```
##
##  Box-Ljung test
##
## data:  resid(arima_fit)
## X-squared = 10.526, df = 12, p-value = 0.5699
```

Neither the ACF nor the PACF have any significant lags - there is likely no dependence structure remaining. To check this, we ran the Ljung-Box test, and got a p-value of 0.5699. This indicates that only white noise remains / there is no dependence structure remaining.

```
qqplot(t,resid(arima_fit))
```



Forecasting

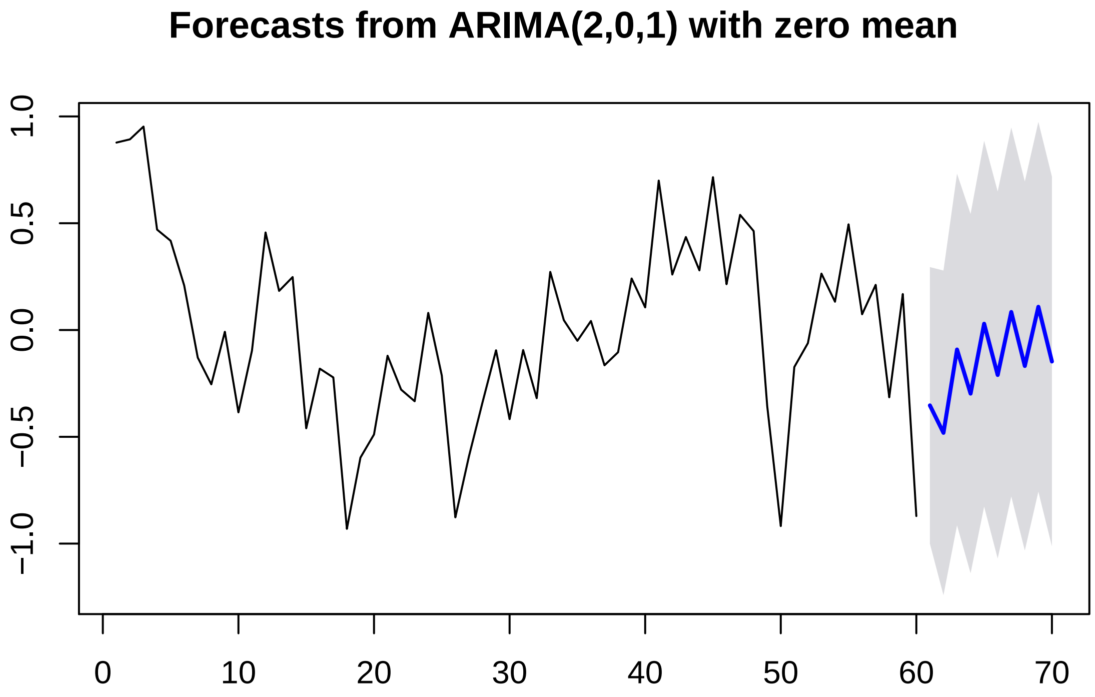
```
shapiro.test(resid(arima_fit))
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  resid(arima_fit)
## W = 0.95274, p-value = 0.021
```

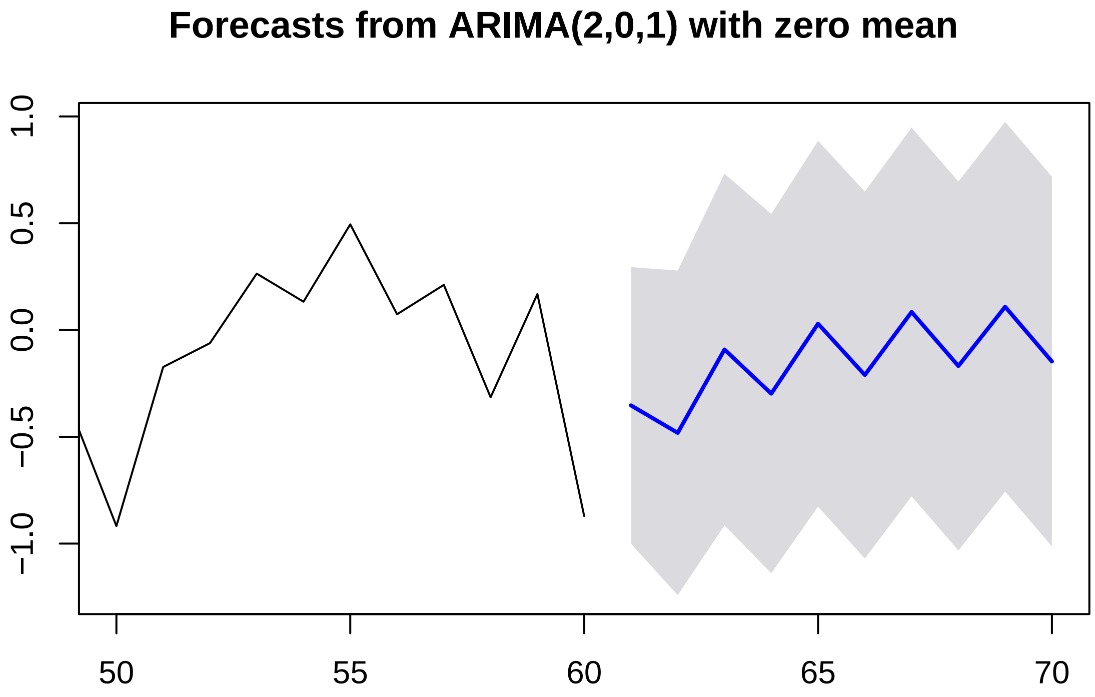
According to our Shapiro-Wilk test, our residuals are not normal. A small p-value of 0.021 rejects the null hypothesis that the residuals are normal at a significance level of 0.05. Since we are rejecting normality here, the forecast intervals may not be correct.

This phenomenon is also shown in our QQ-plot. Points stray from linearity at both ends of the diagonal line.

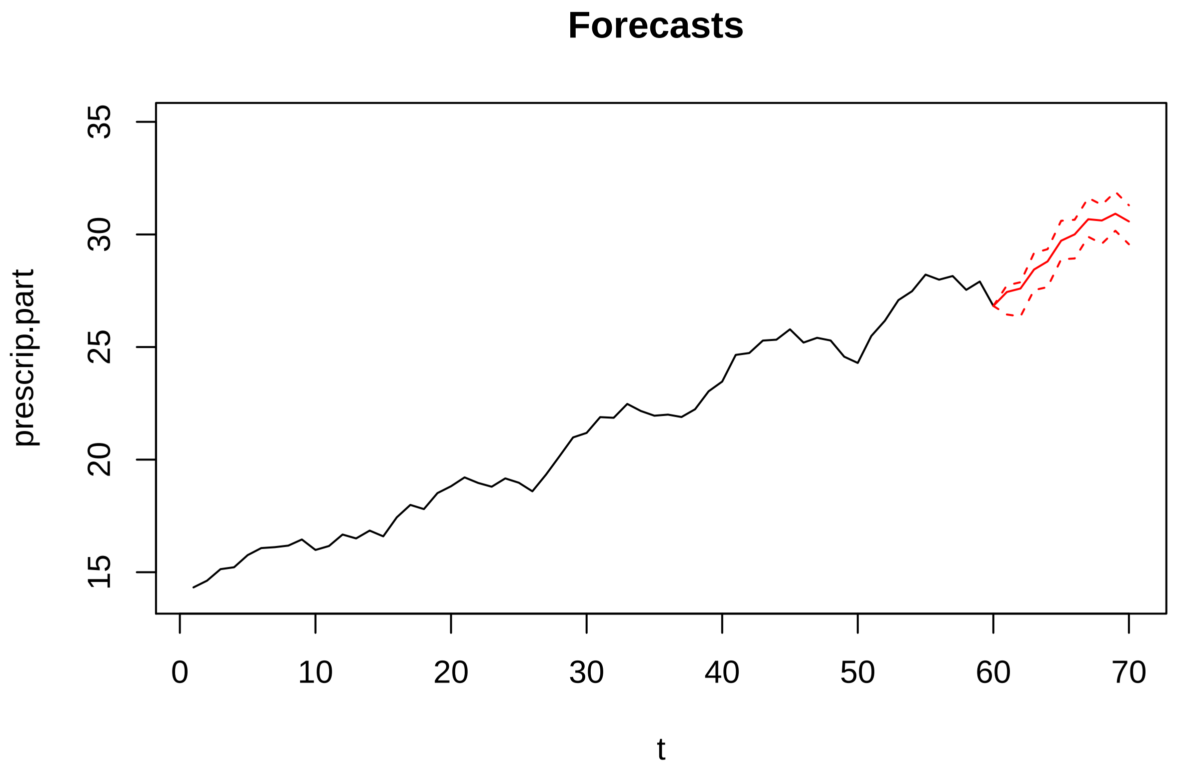
```
# forecast the noise
fc <- forecast(arima_fit, h=10, level = .95)

#plot the noise forecast
plot(fc)
```



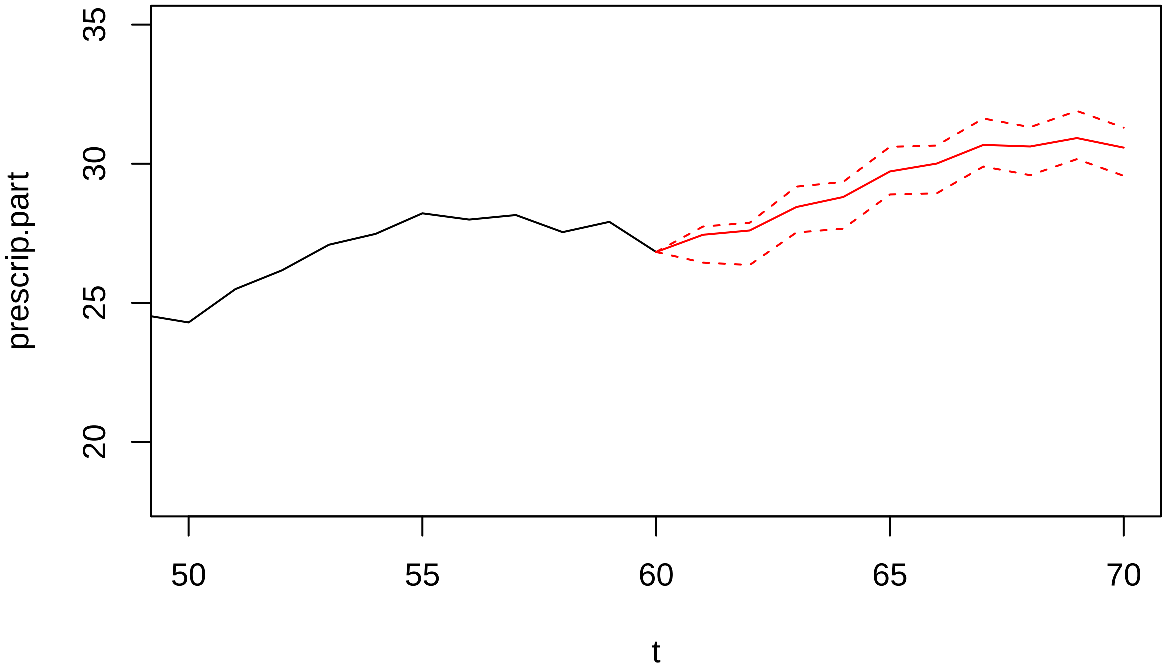
```
#zoom in on the forecast
plot(fc, xlim=c(50,70))
```



```r
#forecast the seasonal component and noise
season.fc = fit.back$fitted.values[1:10]+fc$mean

#forecast the trend
trend.fc = predict(trend.fit, newdata = data.frame(t=61:70))

#add the seasonal and noise forecasts
x.hat = season.fc+trend.fc


plot(t, prescrip.part, main = "Forecasts ", xlim = c(1,70), ylim = c(14,35), type="l")
lines(60:70, c(prescrip.part[60], x.hat), col="red")
#add the forecast intervals
lines(60:70, c(prescrip.part[60], x.hat+fc$lower), col="red", lty=2)
lines(60:70, c(prescrip.part[60], x.hat+fc$upper), col="red", lty=2)
```



```
plot(t, prescrip.part, xlim = c(50,70), ylim = c(18,35), type="l")
lines(60:70, c(prescrip.part[60], x.hat), col="red")
#add the forecast intervals
lines(60:70, c(prescrip.part[60], x.hat+fc$lower), col="red", lty=2)
lines(60:70, c(prescrip.part[60], x.hat+fc$upper), col="red", lty=2)
```



```
x.hat
```

```
## Time Series:
## Start = 61
## End = 70
## Frequency = 1
##        1        2        3        4        5        6        7        8
## 27.44464 27.59900 28.44239 28.80213 29.72107 30.00353 30.67611 30.61864
##        9       10
## 30.92087 30.57684
```

The values of the 10 forecasts are given by x.hat, which are: 27.44464, 27.59900, 28.44239, 28.80213, 29.72107, 30.00353, 30.67611, 30.61864, 30.92087 and 30.57684.

We would expect the seasonality to follow the same pattern and the trend to increase at the same rate as the first 60 points. The forecast does what we expect it to do based on our previous points. However, since our residuals are not normal (based on the Shapiro-Wilk test), our forcasting is not reliable.