Liam Moyers - PSTAT 174 Final Project

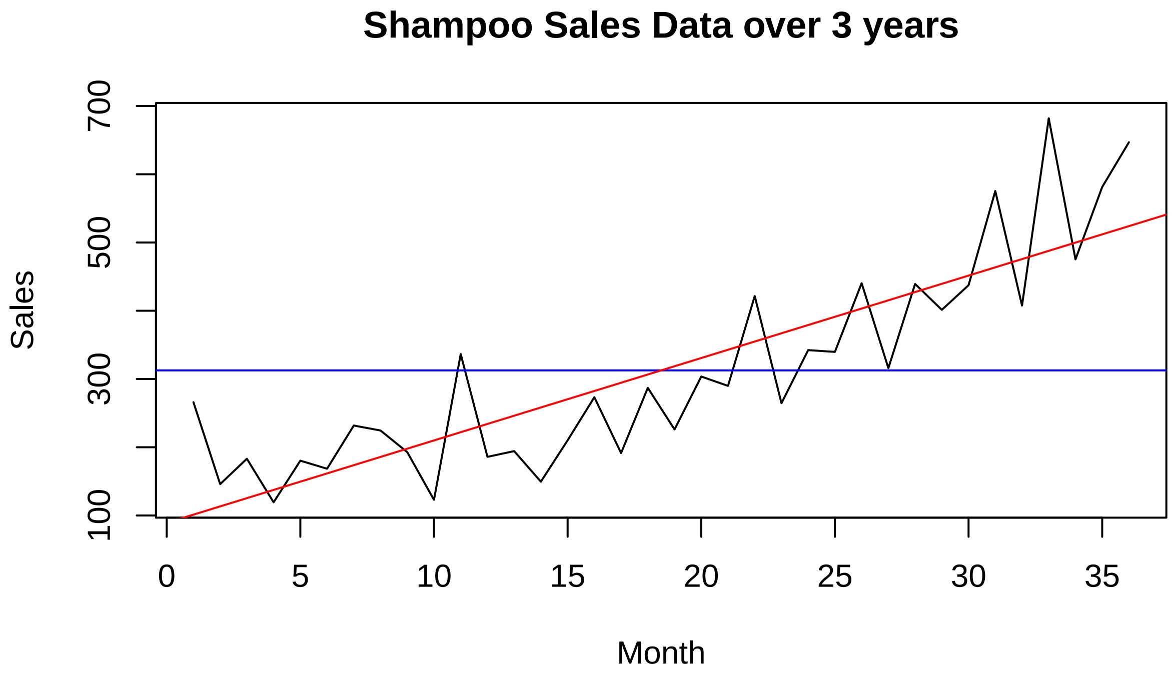
Abstract

This project takes 3 years worth of shampoo sales data from 1998-2001 and forecasts future sales. Forecasting future sales will allow this business to better manage its inventory of shampoo as well as allow them to introduce promotional offers when shampoo sales are forecasted to be low.

The data used for this project contains 36 months of shampoo sales data starting from January 1998 - December 2001. This project used a log transformation, differenced to remove trend, and forecasted future sales. ARIMA(1,0,1) proved to be the best model and passed all diagnostic tests. ARIMA(0,0,2) model was second best and also passed all diagnostic tests. This project concluded that sales would go up in the next few months with 90% confidence. Therefore, conclude that more shampoo should be stocked and no promotions should be used until a later date.

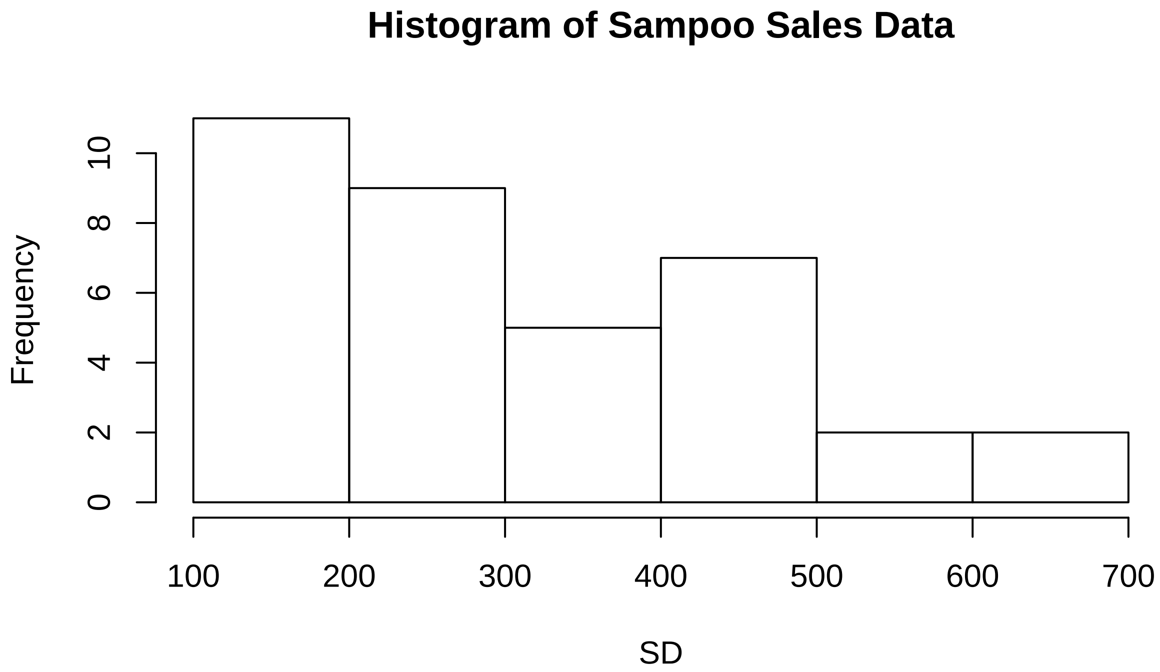
Introduction

The problem this project aims to solve is the uncertainty of shampoo sales. We solve this problem through forecasting. Forecasting sales allows the business to more accurately stock the store, run promotions if forecasted sales are low in the coming months, and hire an appropriate ammount of employees. This dataset is important because shampoo sales are this business's only mode of making profit! This project found that in the coming months of 2002 we are to see an increase of sales from December 2001 and should prepare accordingly. Unfortunately, this project was unable to forecast deep into the future. This is most likely because lack of data from the store. The model is likely to continue improving as more months go by and the model is updated. The data comes from Makridakis, Wheelwright and Hyndman (1998) and can be downloaded at https://raw.githubusercontent.com/jbrownlee/Datasets/master/shampoo.csv. This project used R.Studio.



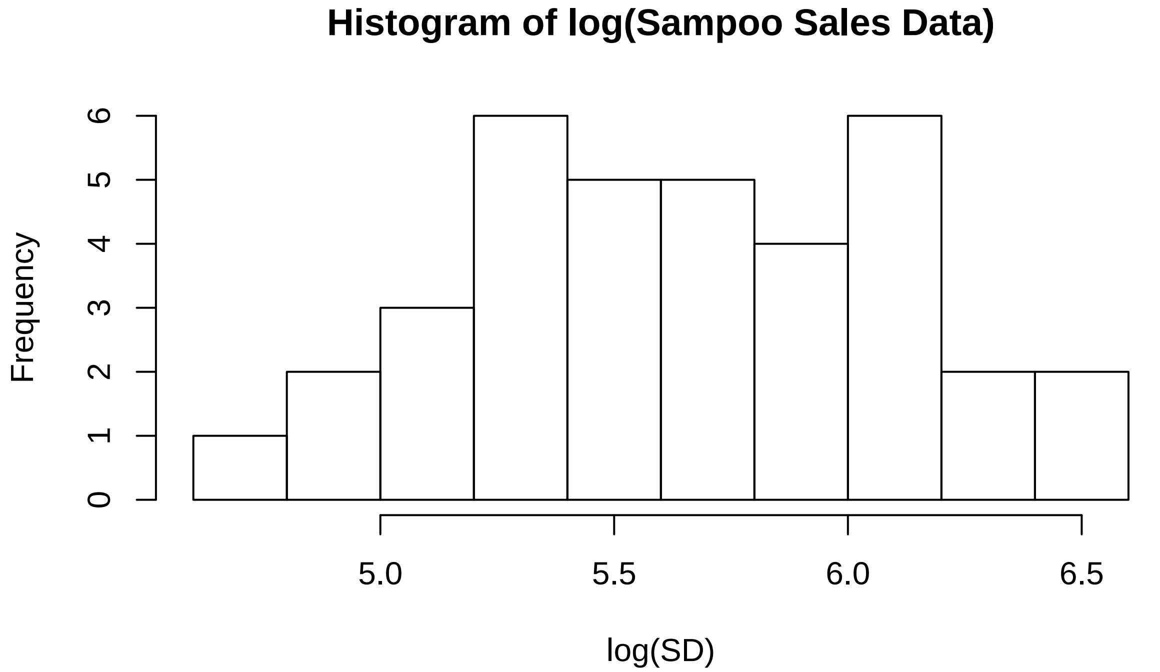
From the graph we immediately see that there is upward trend and potentially seasonality. This is important for later because we will need to difference the time series in order to have a stationarity. We also see that we have a nonconstant mean and variance. This means we should investigate the histogram and see if we need to transform our time series.

```
hist(SD, main = "Histogram of Sampoo Sales Data")
```



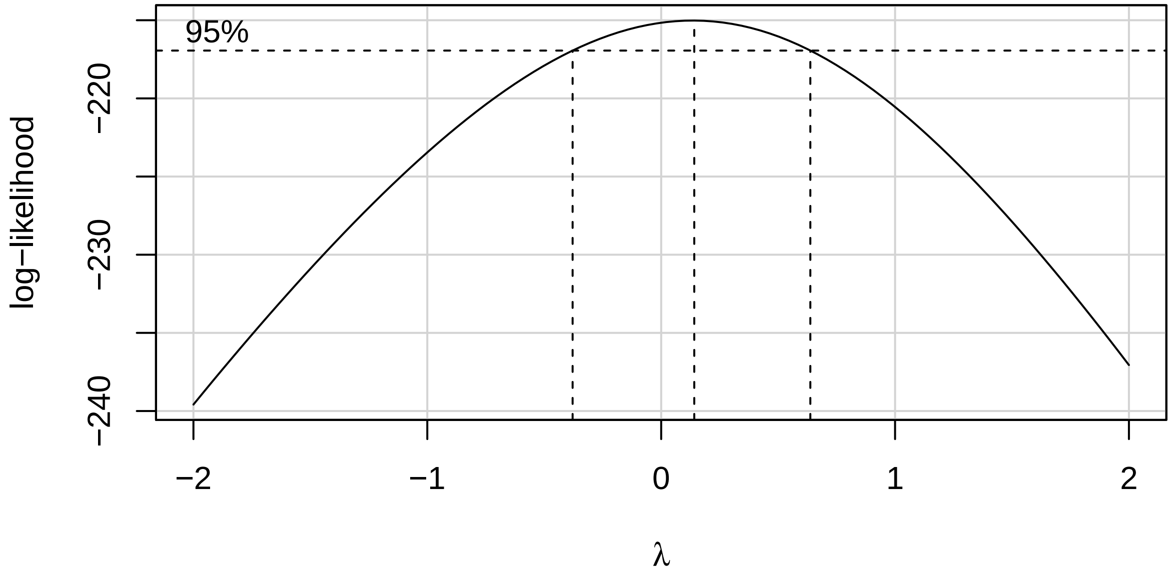
Unfortunately our data is very skewed, so we'll need to transform our data in some way. We will need to take a look at a potential log() or boxCox() transformation in order to minimize variance and make our data more symmetrical.

```
hist(log(SD), main = "Histogram of log(Sampoo Sales Data)")
```



The log transform is much more symmetrical than the original histogram.

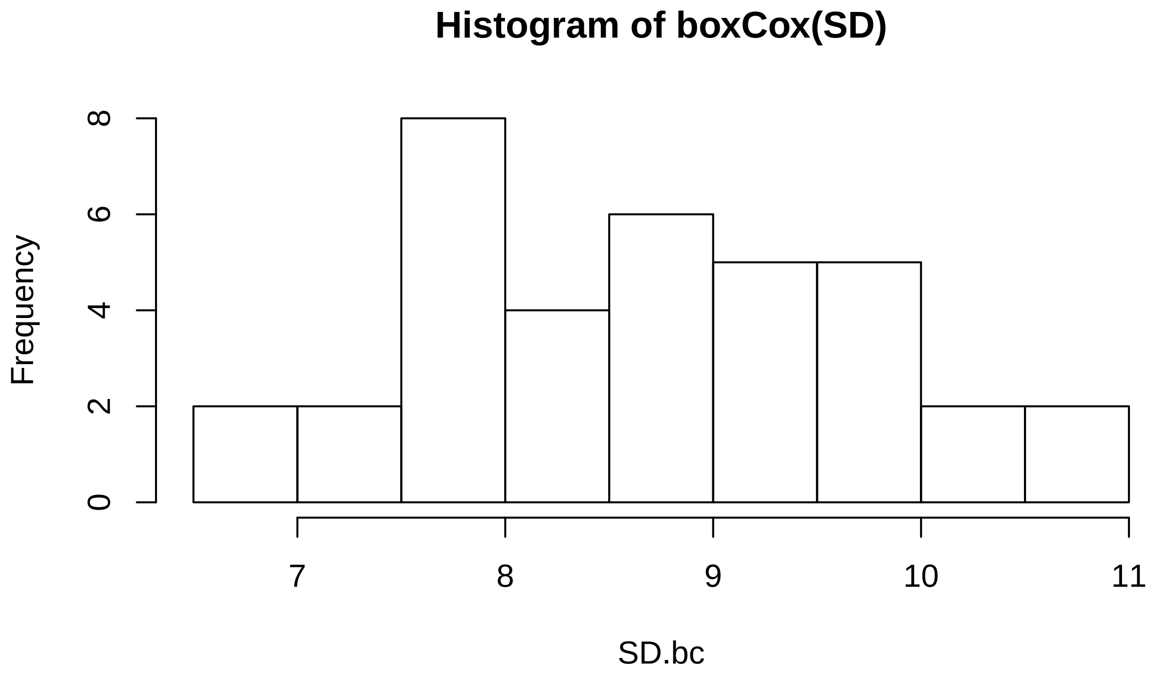
```
bcTransform <- boxCox(SD~ as.numeric(1:length(SD)))
```



```
lambda <- bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
```

From the bcTransform command we see that lambda = .14141414 and 0 is contained in the confidence interval. This suggests that a log transform will be best.

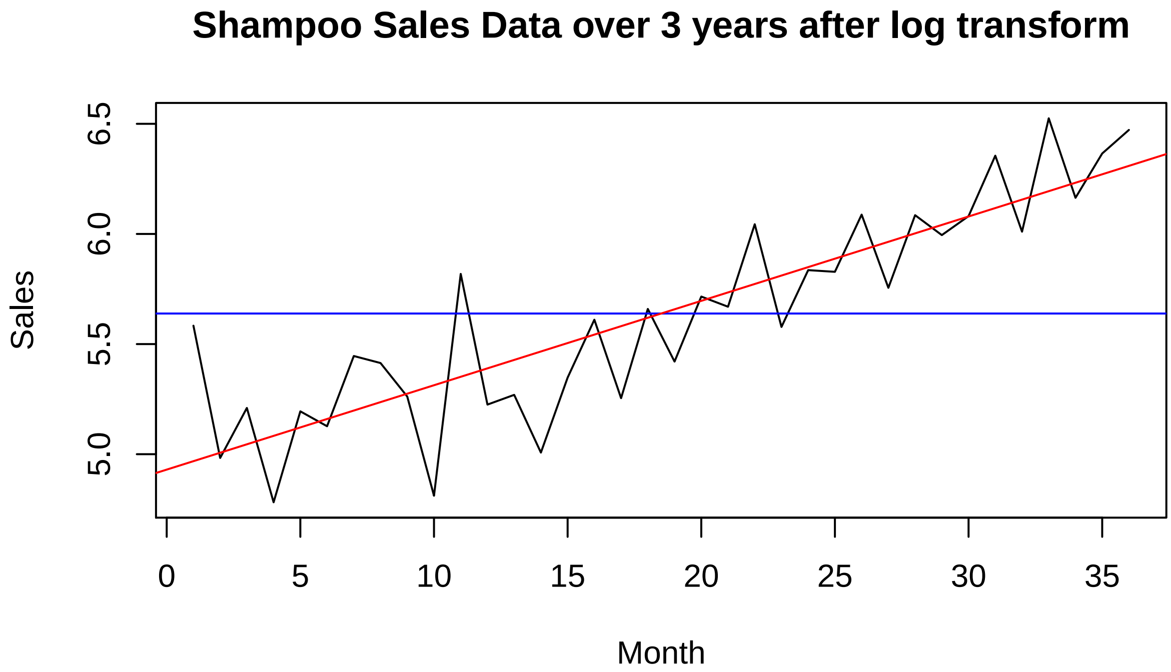
```
SD.bc <- (1/lambda)*(SD^lambda-1)
hist(SD.bc, main = "Histogram of boxCox(SD)")
```



Although the boxCox transformation is better than the original, it appears to be less symmetrical than the log transform. Therefore, we conclude that log transform is best for our time series.

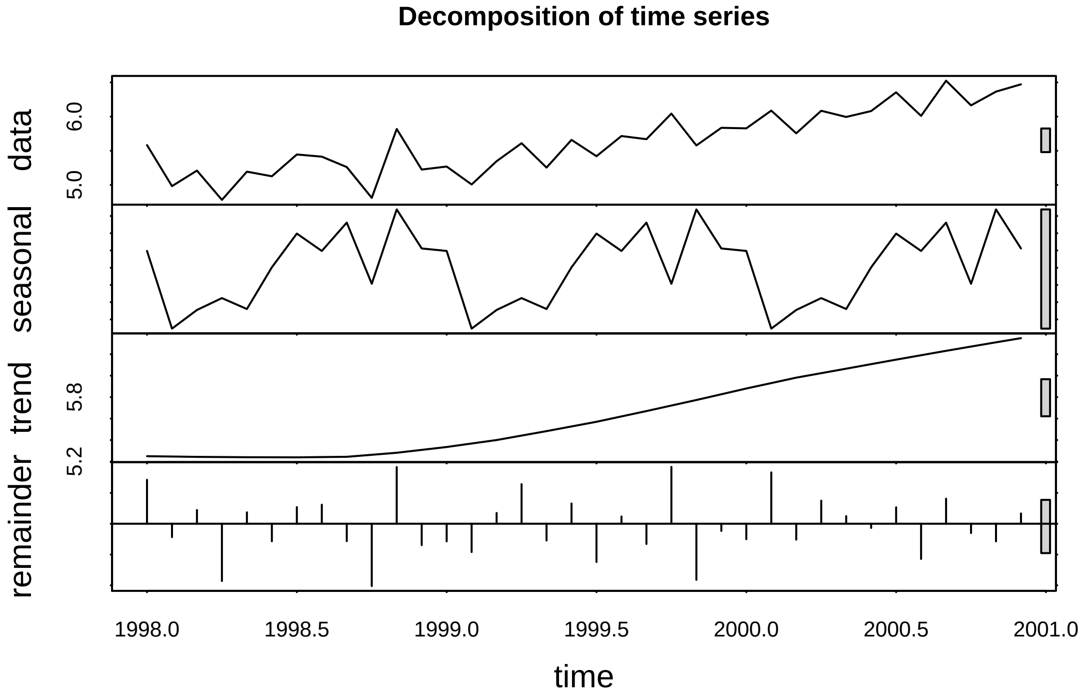
```
SD.log <- log(SD)

plot.ts(SD.log, main = "Shampoo Sales Data over 3 years after log transform", xlab = "Month", ylab = "Sa
abline(h = mean(SD.log), col ="blue")
fit2 <- lm(SD.log ~ c(1:length(SD.log)))
abline(fit2, col = "red")
```



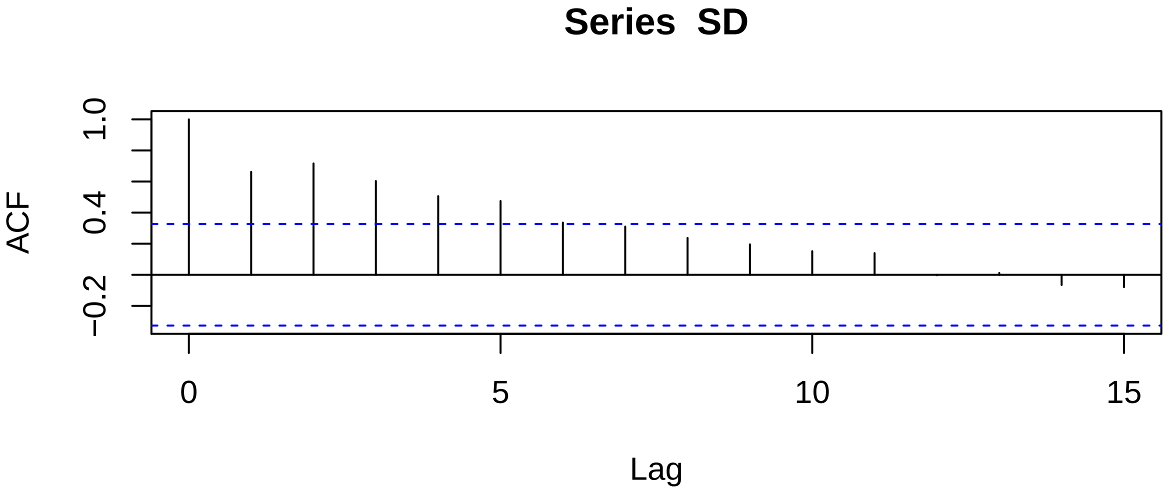
After the log transform we see that we still have an upward trend and potential seasonality. We now need to run a decomposition and look at the trend and seasonality to confirm our suspicions.

```
SD_ts.log <- ts(SD.log, start=c(1998,1), frequency = 12)
decomp <- stl(SD_ts.log, s.window = "periodic")
plot(decomp, main = "Decomposition of time series")
```

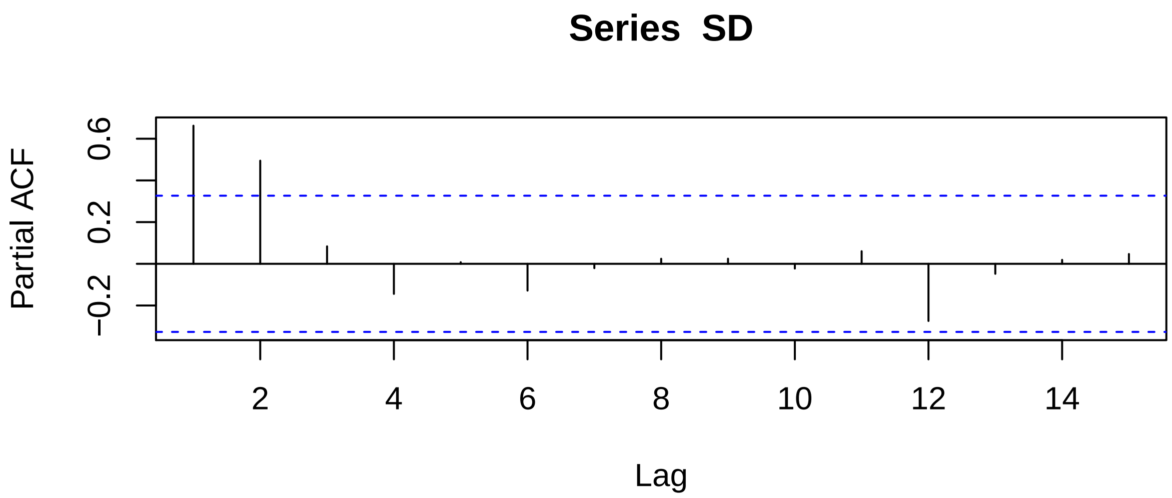


Looking at the decomposition we can see that we have linear trend. We also see a possibility for a seasonality component. However, we'll need to further test for seasonality because the decomp function in R purposefully smoothes the data, which can be misleading. We'll take a look at the acf and pacf of the time series and see if they help us determine the existence of seasonality.

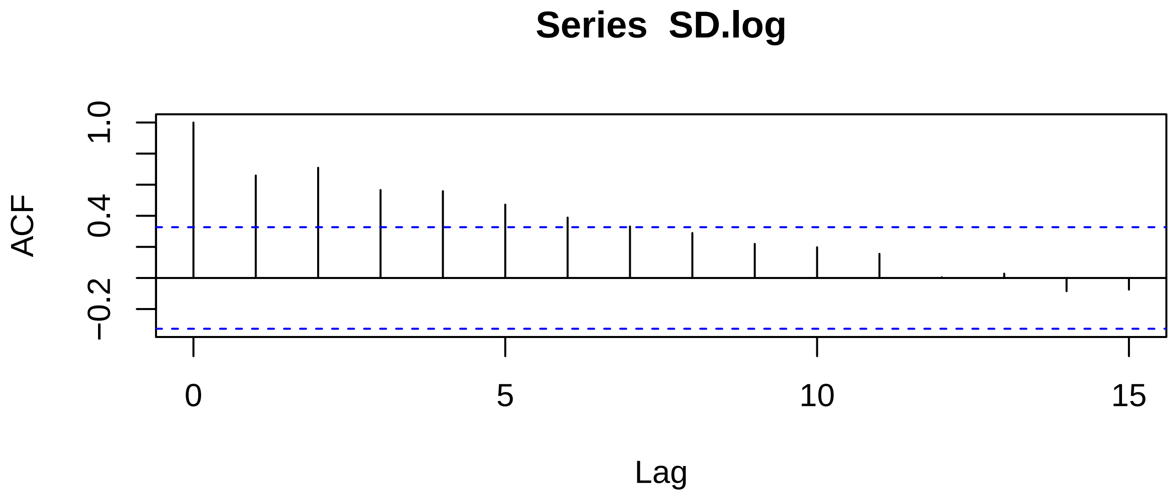
```
acf(SD)
```



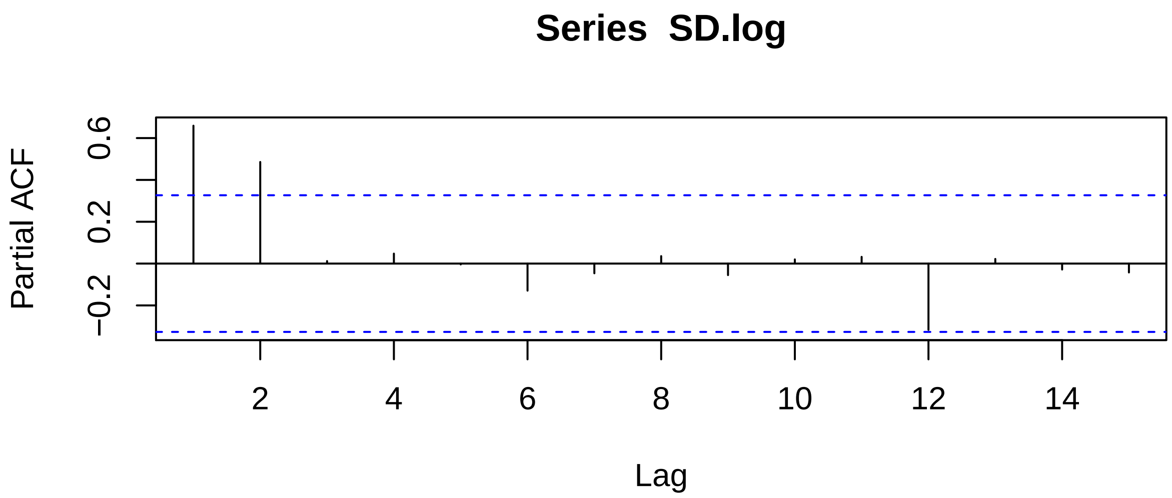
```
pacf(SD)
```



```r
acf(SD.log)
```

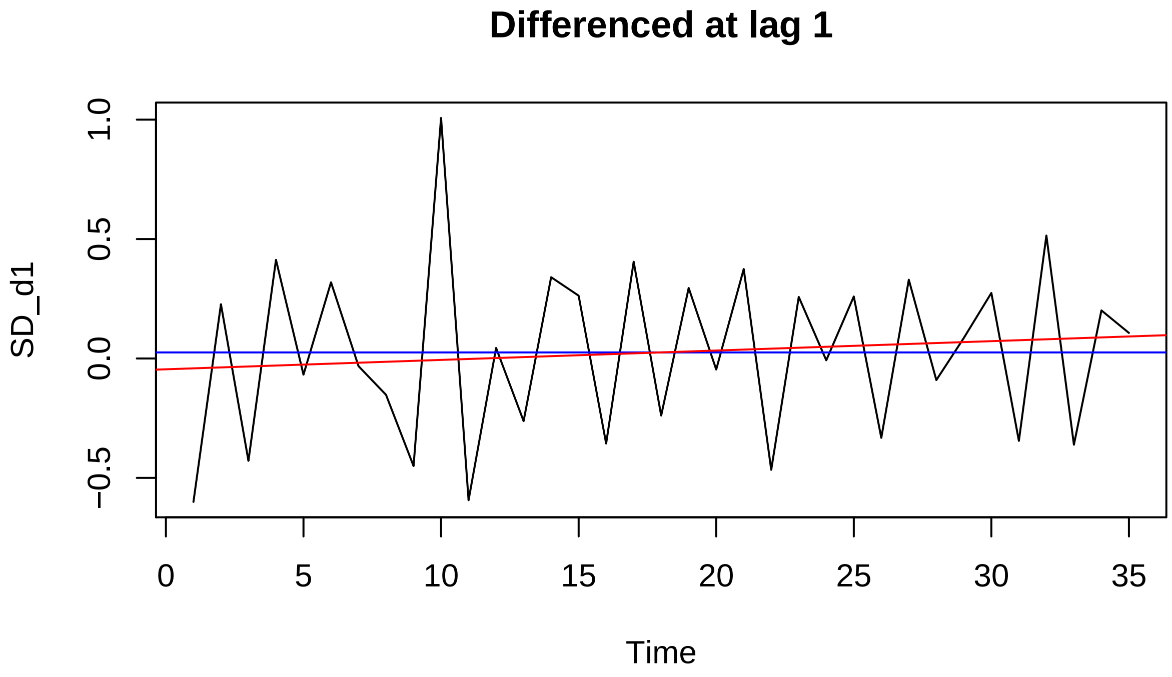


```r
pacf(SD.log)
```



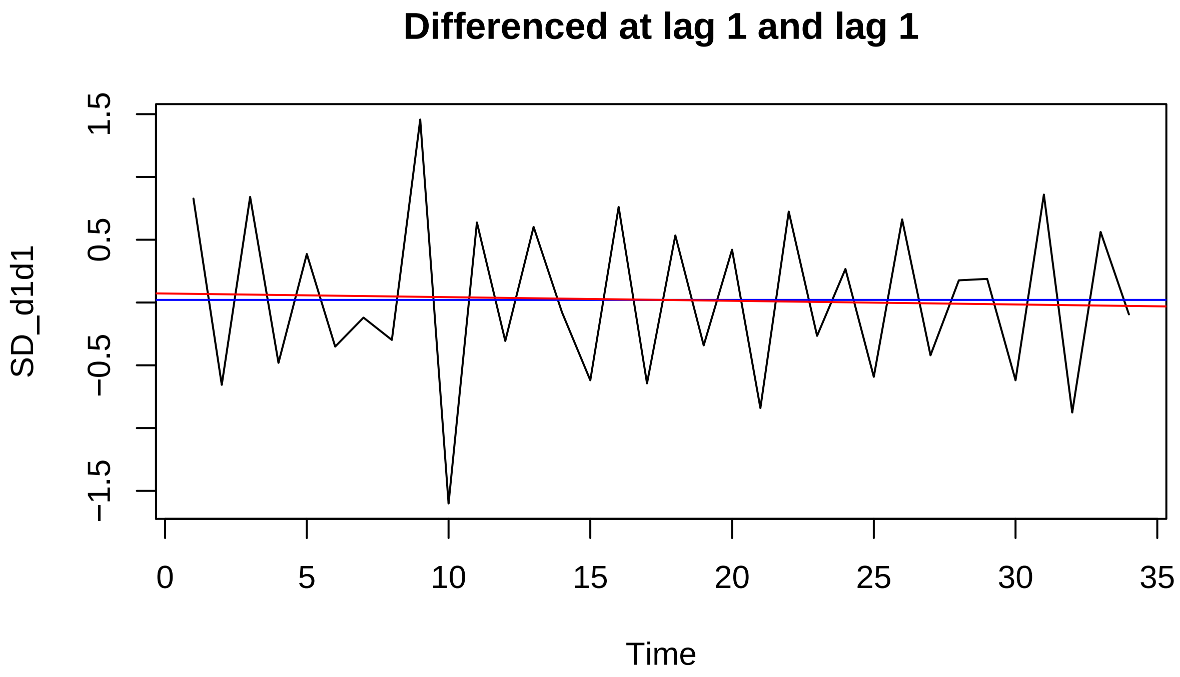
After investigating the acf and pacf of SD and SD.log we found no indication of seasonality and will continue onto differencing for trend starting at lag = 1. We need to difference at lag = 1 because doing so should make the time series stationary. If it does make it stationary we can move onto identifying a model to forcast our shampoo sales.

```
SD_d1 <- diff(SD.log, lag = 1)
plot.ts(SD_d1, main = "Differenced at lag 1")
abline(h = mean(SD_d1), col ="blue")
fit3 <- lm(SD_d1 ~ c(1:length(SD_d1)))
abline(fit3, col = "red")
```



After differencing at lag = 1 we still see a slight upwards trend so we will difference again and compare variances. If the variance goes up then it's a sign the original model is better.

```
SD_d1d1 <- diff(SD_d1, lag = 1)
plot.ts(SD_d1d1, main = "Differenced at lag 1 and lag 1")
abline(h = mean(SD_d1d1), col ="blue")
fit4 <- lm(SD_d1d1 ~ c(1:length(SD_d1d1)))
abline(fit4, col = "red")
```



We now see that our trend line has a negative slope, indicating a slight over differencing. Comparing respective variances we see that differencing once at lag 1 gives us a better variance = 0.1324857, as opposed to the variance of differencing at lag 1 and 1, which equals 0.4408112. Therefore, we will stick with the first difference at lag 1 and discard the differencing at lag 1 and 1. We now assume that we have a stationary time series but lets run a few tests to make sure. First we'll run kpss.test which has a null hypothesis that the series is stationary. If kpss returns a value larger than .05 we'll be able to confirm that our time series is indeed stationary. We will also run adf.test (Augmented Dickey-Fuller), which has the opposite null hypothesis of the time series being non-stationary. If adf returns a p value $<$ .05 then we will be able to confirm our time series stationarity.

```
kpss.test(SD_d1)
```

```
## Warning in kpss.test(SD_d1): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  SD_d1
## KPSS Level = 0.2373, Truncation lag parameter = 3, p-value = 0.1
```

p value = .1 $>$ .05 so kpss confirms stationarity

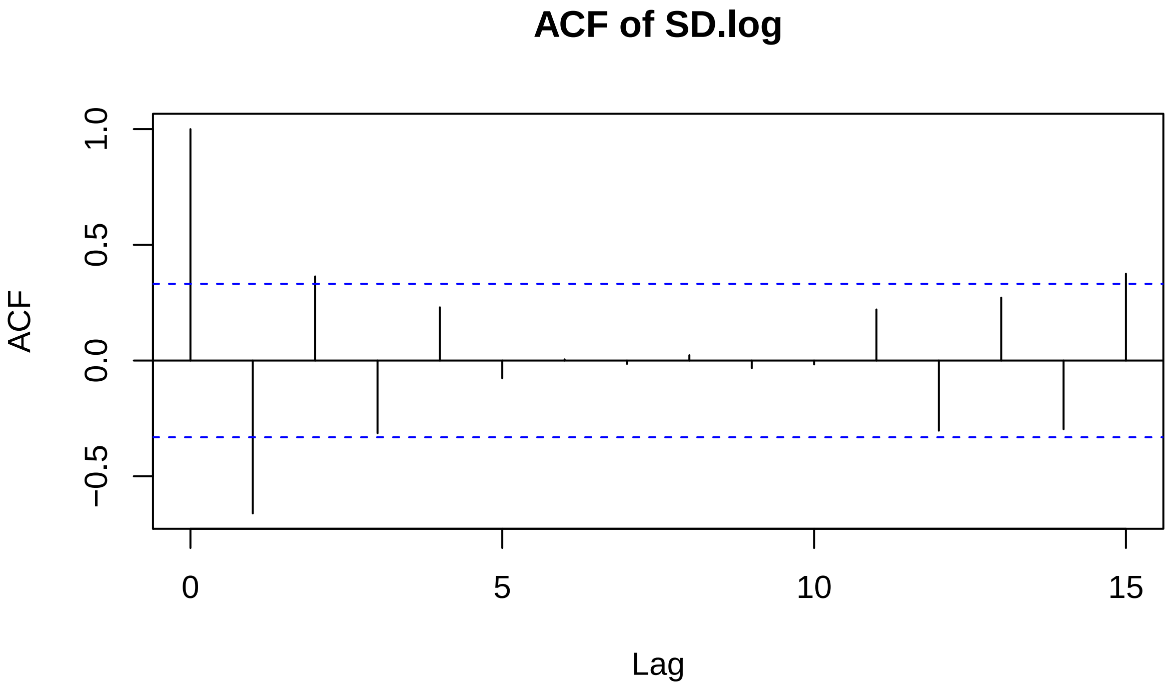
```
adf.test(SD_d1)
```

```
## Warning in adf.test(SD_d1): p-value smaller than printed p-value
```

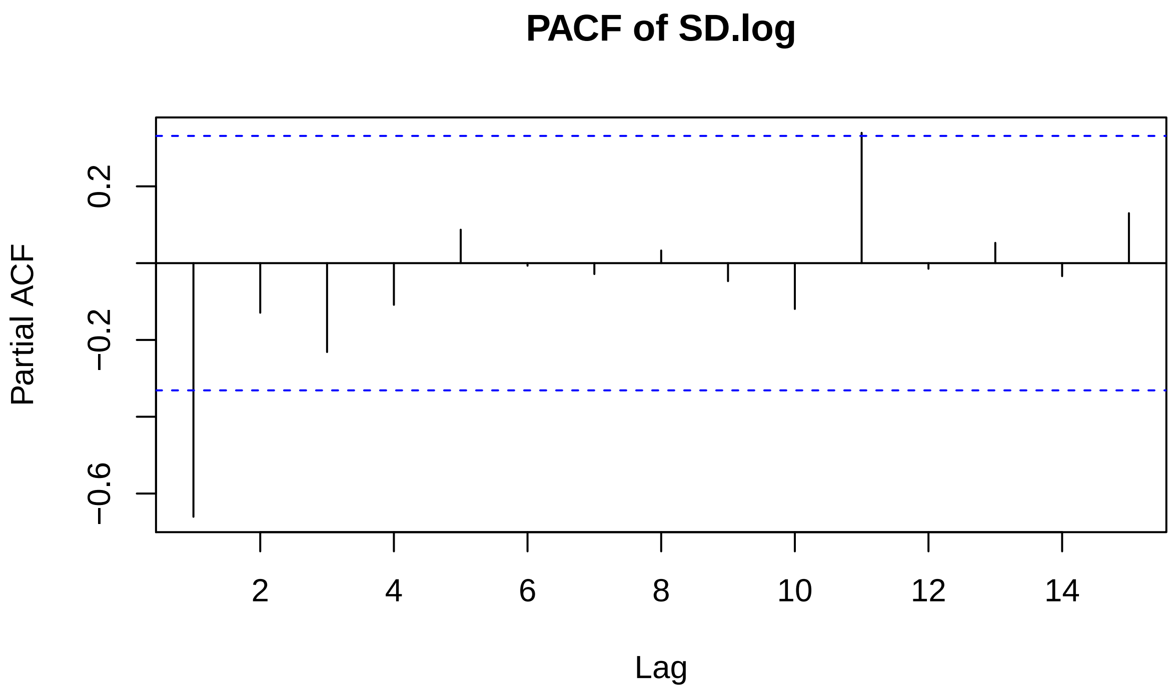
```
##
##  Augmented Dickey-Fuller Test
##
## data:  SD_d1
## Dickey-Fuller = -6.1103, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

p value = .01 < .05, so the adf confirms stationarity as well. Now that we have confirmed stationarity we will move onto disecting the acf and pacf in order to build our model.

```
acf(SD_d1, main = "ACF of SD.log")
```



```
pacf(SD_d1,  main = "PACF of SD.log")
```



Plot of ACF differenced at lag 1 shows decay which corresponds to stationarity. We also see that the ACF is outside the confidence interval at lags 1, 2, and 15. Therefore we suspect MA, q = 1, 2 or 15. Our PACF is outside confidence intervals at lag 1. Therefore we suspect AR, p = 1. This is because values outside the confidence intervals are statistically significant.

Now we test each of the potential models to see which has the lowest AIC and AICc. Whichever has the lowest AIC and AICc will be the best fit model for our sales data.

```
arima(SD_d1, order=c(1,1,1), method="ML")
```

```
##
## Call:
## arima(x = SD_d1, order = c(1, 1, 1), method = "ML")
##
## Coefficients:
##           ar1      ma1
##       -0.6939  -1.0000
## s.e.   0.1248   0.1253
##
## sigma^2 estimated as 0.06906:  log likelihood = -5.43,  aic = 16.86
```

```
AICc(arima(SD_d1, order=c(1,1,1), method="ML"))
```

```
## [1] 17.65552
```

AIC = 16.86 #3rd Lowest AIC AICc = 17.65552 #3rd Lowest AICc

```r
arima(SD_d1, order=c(1,1,15), method="ML")
```

```
##
## Call:
## arima(x = SD_d1, order = c(1, 1, 15), method = "ML")
##
## Coefficients:
##            ar1      ma1      ma2     ma3      ma4     ma5      ma6      ma7
##        -0.8600  -1.2769  -0.0620  0.7382  -0.2613  0.0275  -0.2656  -0.1901
## s.e.   0.2182   0.8290   1.6055  1.3753   0.9457  0.9999   1.0736   1.0904
##          ma8      ma9    ma10     ma11     ma12    ma13     ma14     ma15
##        0.3982  -0.0401  0.4098  -0.2038  -1.0297  1.3083  -0.1403  -0.4121
## s.e.  0.9407   0.9251  0.9362   1.3417   1.5297  0.9348   0.6816   0.4989
##
## sigma^2 estimated as 0.02325:  log likelihood = 5.17,  aic = 23.66
```

```r
AICc(arima(SD_d1, order=c(1,1,15), method="ML"))
```

```
## [1] 61.90806
```

AIC = 23.66 #4th lowest AICc = 61.90806 #4th lowest

```r
arima(SD_d1, order=c(1,1,0), method="ML")
```

```
##
## Call:
## arima(x = SD_d1, order = c(1, 1, 0), method = "ML")
##
## Coefficients:
##            ar1
##        -0.8107
## s.e.   0.0954
##
## sigma^2 estimated as 0.144:  log likelihood = -15.83,  aic = 35.67
```

```r
AICc(arima(SD_d1, order=c(1,1,0), method="ML"))
```

```
## [1] 36.05485
```

AIC = 35.67 #5th lowest AIC AICc = 36.05485 #5th lowest AICc

```r
arima(SD_d1, order=c(1,1,2), method="ML")
```

```
##
## Call:
## arima(x = SD_d1, order = c(1, 1, 2), method = "ML")
##
## Coefficients:
##            ar1      ma1     ma2
##        -0.3467  -1.7896  0.8427
## s.e.   0.1943   0.1557  0.1622
##
## sigma^2 estimated as 0.05397:  log likelihood = -1.89,  aic = 11.79
```

```
AICc(arima(SD_d1, order=c(1,1,2), method="ML"))
```

```
## [1] 13.16461
```

AIC = 11.79 # lowest AIC AICc = 13.16461 # lowest AICc

```
arima(SD_d1, order=c(0,1,2), method="ML")
```

```
##
## Call:
## arima(x = SD_d1, order = c(0, 1, 2), method = "ML")
##
## Coefficients:
##           ma1     ma2
##       -1.9347  1.0000
## s.e.   0.1847  0.1887
##
## sigma^2 estimated as 0.05532:  log likelihood = -3.3,  aic = 12.6
```

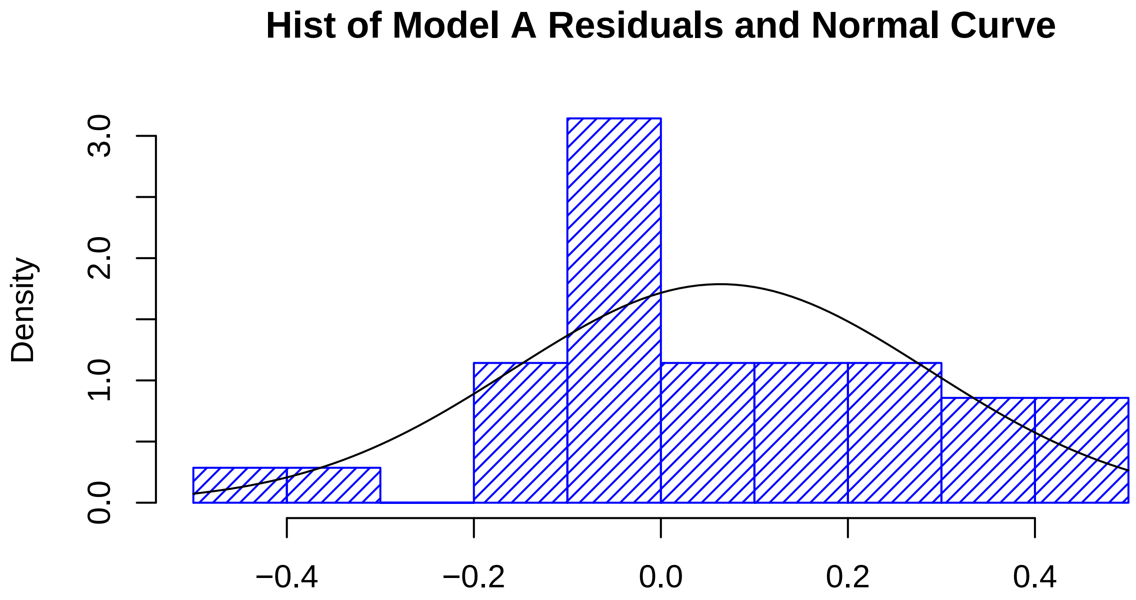
```
AICc(arima(SD_d1, order=c(0,1,2), method="ML"))
```

```
## [1] 13.39696
```

AIC = 12.6 #2nd lowest AICc = 13.39696 #2nd lowest

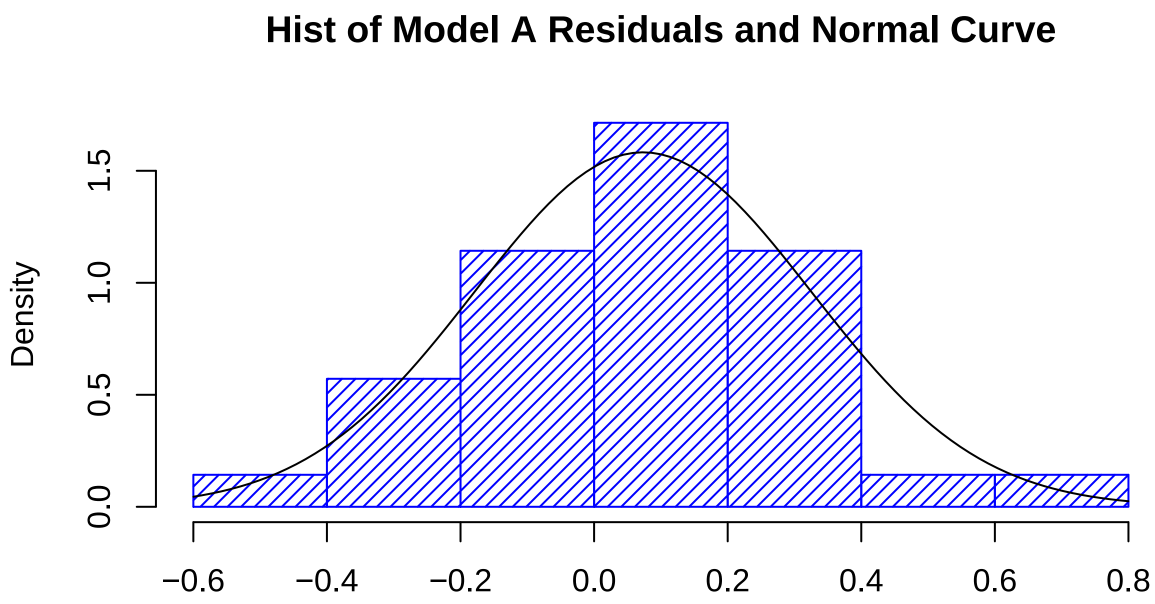
After testing multiple models we conclude that ARIMA(1,1,2) and ARIMA(0,1,2) are the two best because they have the lowest AIC and AICc. We will now run diagnostic testing on both models starting with:

```
fit5 <- arima(SD_d1, order=c(1,1,2), method="ML")
res <- residuals(fit5)
hist(res,density=20, col="blue", xlab="", prob=TRUE, main = "Hist of Model A Residuals and Normal Curve
m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )
```



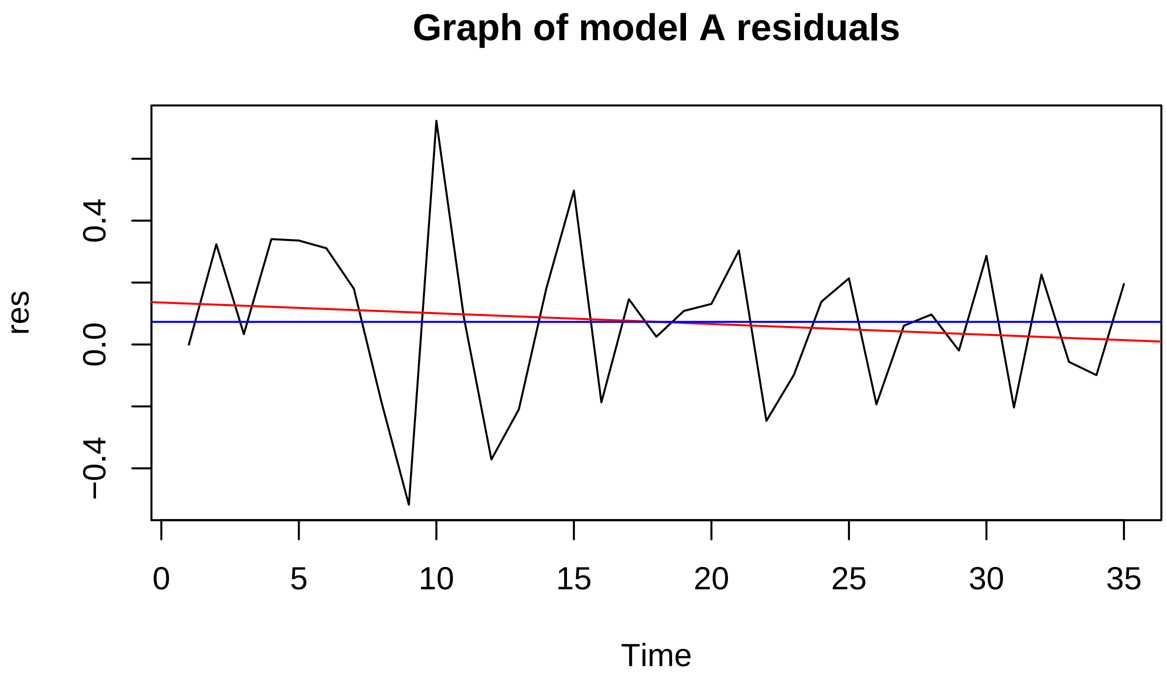
Doesn't resemble Gaussian which is bad. Let's try ARIMA(1,1,1) instead lets take the 2nd best and 3rd best AIC.

```
fit5 <- arima(SD_d1, order=c(1,1,1), method="ML")
res <- residuals(fit5)
hist(res,density=20, col="blue", xlab="", prob=TRUE, main = "Hist of Model A Residuals and Normal Curve
m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )
```



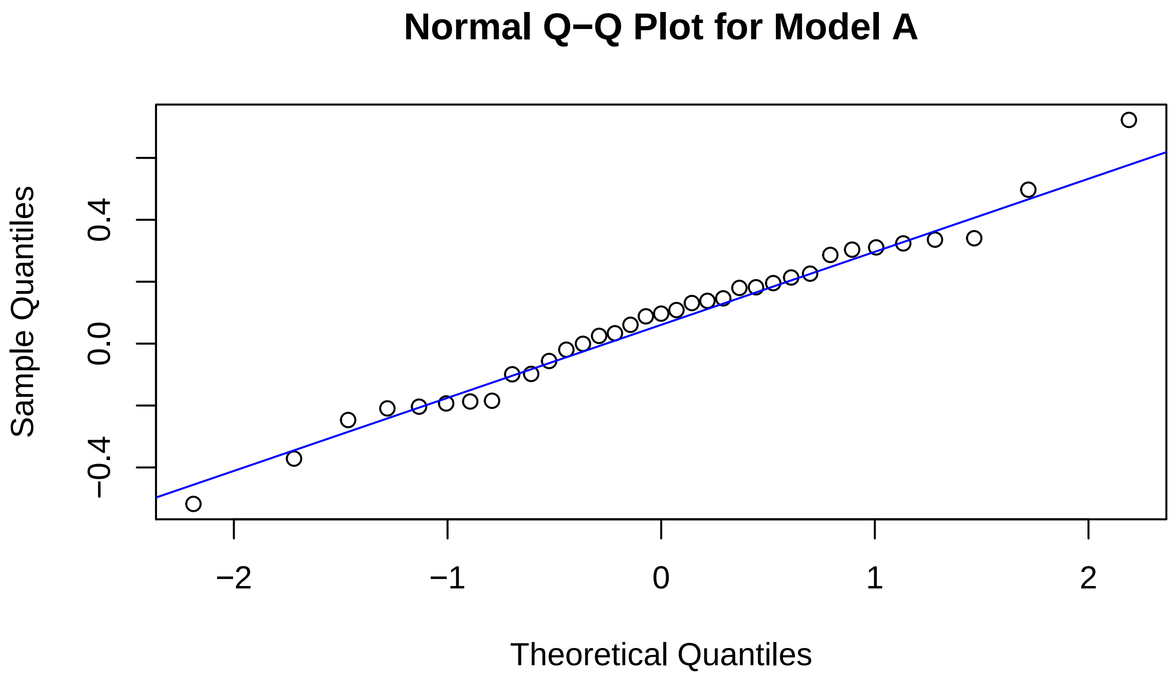
Much better.

```r
plot.ts(res, main = "Graph of model A residuals")
fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
```



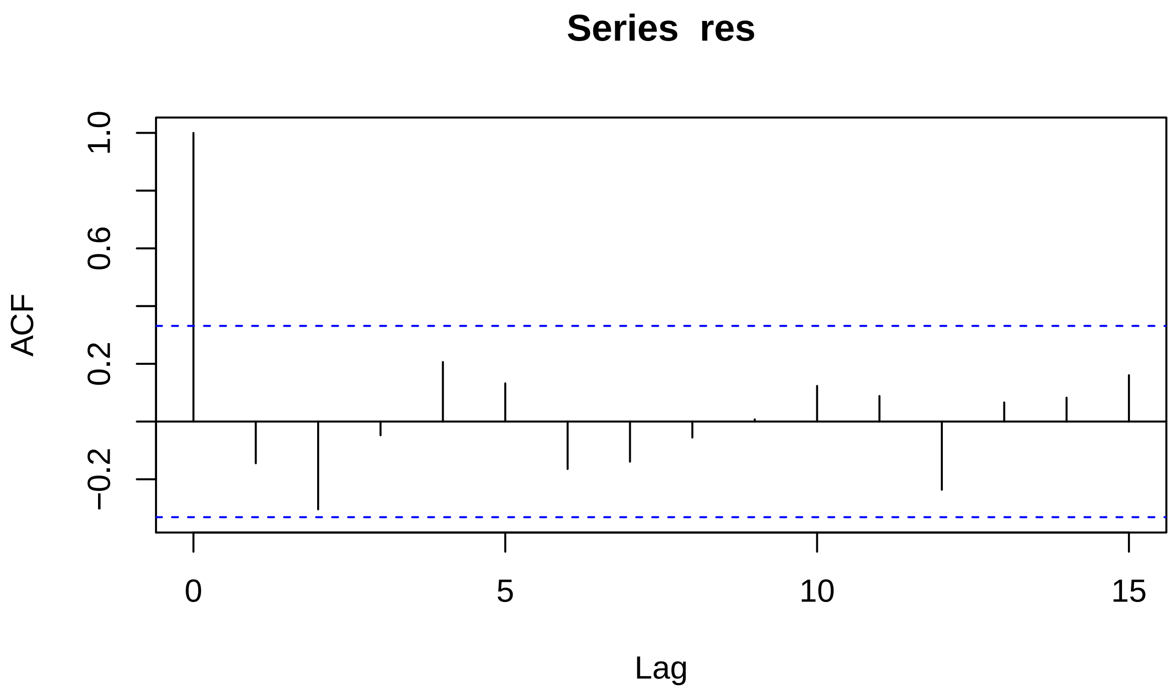
Slightly negative trend unfortunately, but lets continue.

```
qqnorm(res,main= "Normal Q-Q Plot for Model A")
qqline(res,col="blue")
```

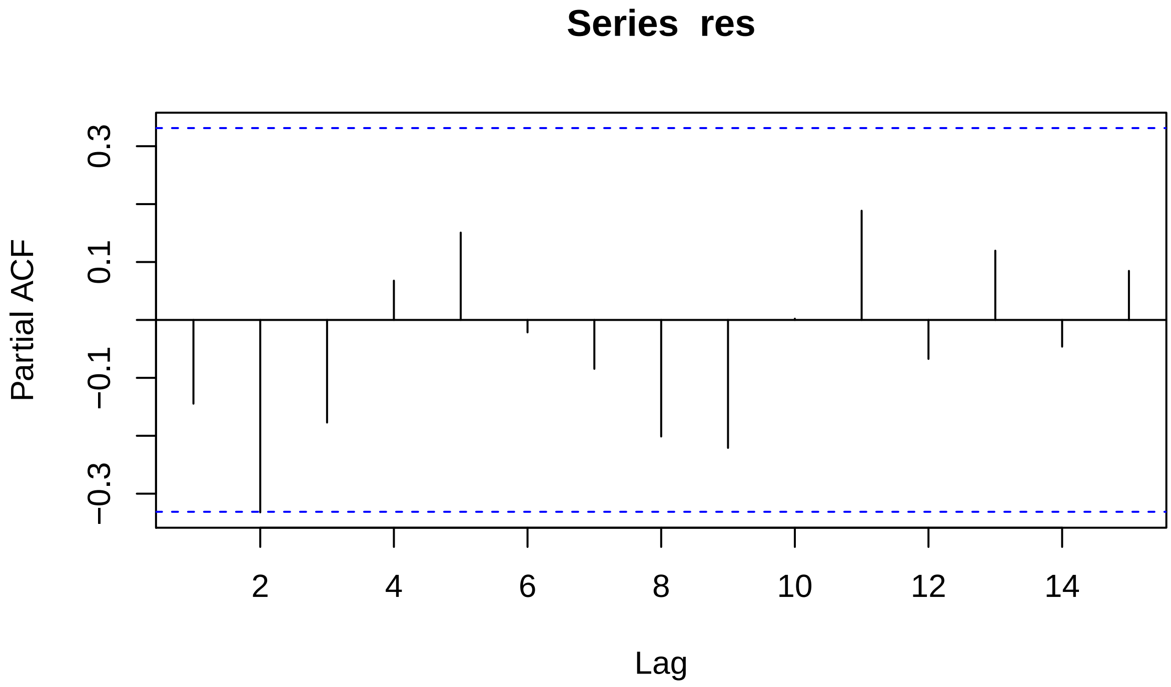


Close to straight line. Good!

```
acf(res)
```



```
pacf(res)
```



All sample acf/pacf inside confidence interval. Good!

```
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.98514, p-value = 0.9072
```

Reject when values of W are too small or p value < .05. W is large and p-value = .9072 > .05. Passes Shapiro-wilk test!

```
Box.test(res, lag = 6, type = c("Box-Pierce"), fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 7.1059, df = 4, p-value = 0.1304
```

```
Box.test(res, lag = 6, type = c("Ljung-Box"), fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 8.2689, df = 4, p-value = 0.08221
```

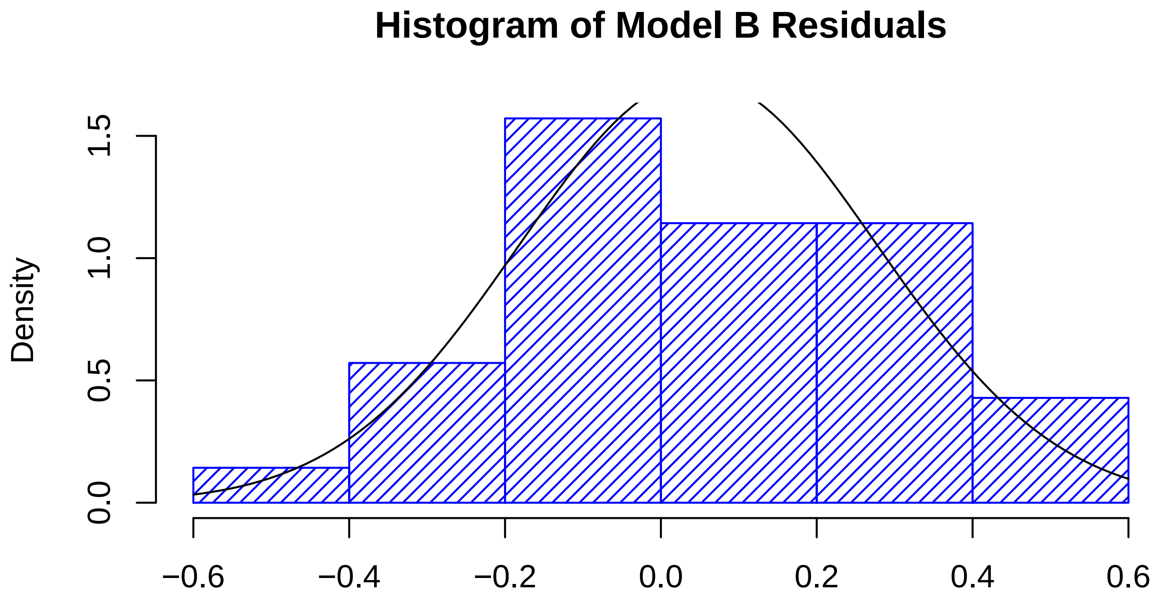
```
Box.test(res^2, lag = 6, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  res^2
## X-squared = 7.4125, df = 6, p-value = 0.2844
```

All p-values are larger than .05!

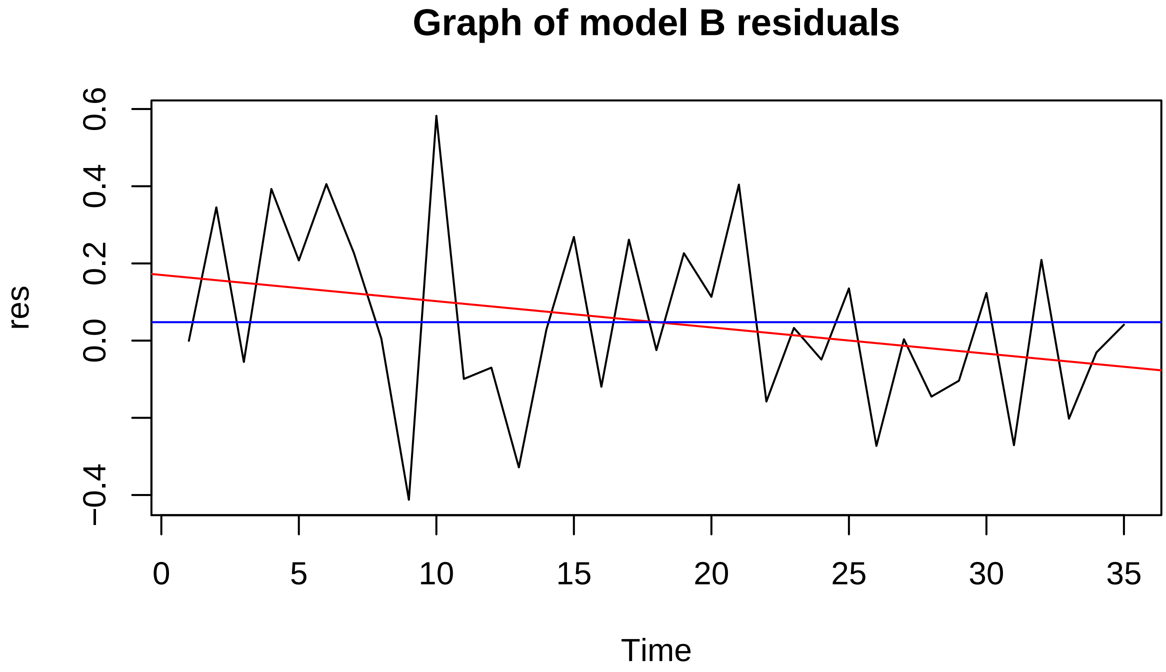
Now that we passed all diagnostic testing we can run diagnostics on model B.

```
fit6 <- arima(SD_d1, order=c(0,1,2), method="ML")
res <- residuals(fit6)
hist(res,density=20, col="blue", xlab="", prob=TRUE, main = "Histogram of Model B Residuals")
m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )
```



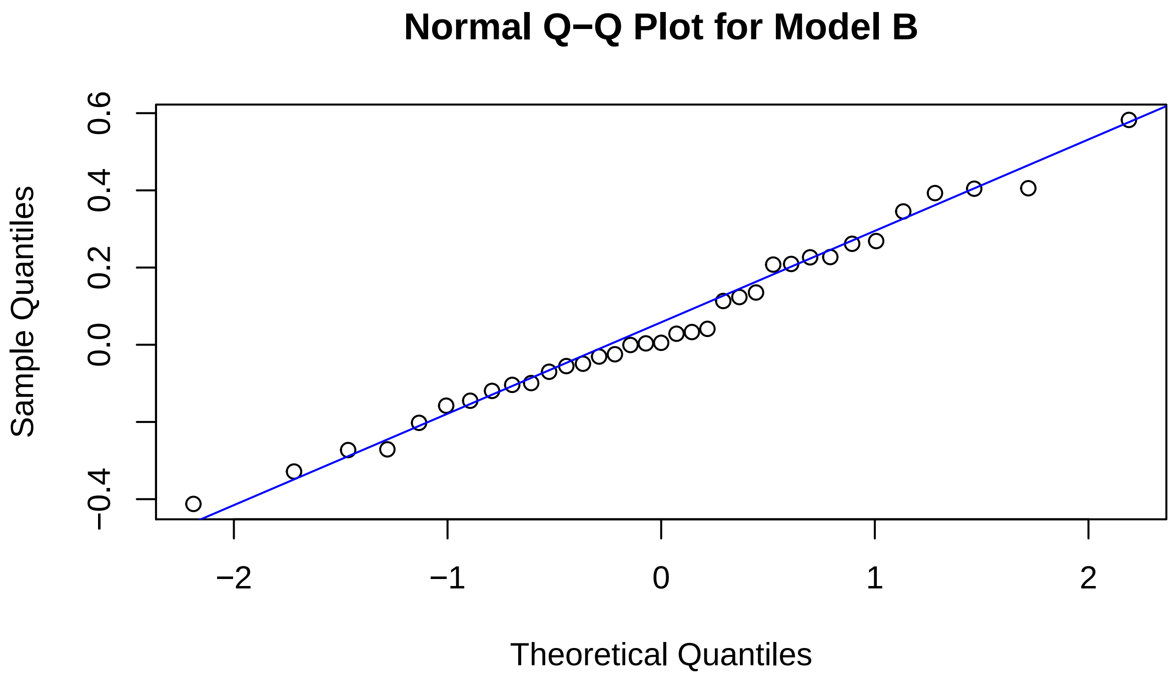
Not nearly as close to gaussian as model A, but lets continue.

```
plot.ts(res, main = "Graph of model B residuals")
fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
```



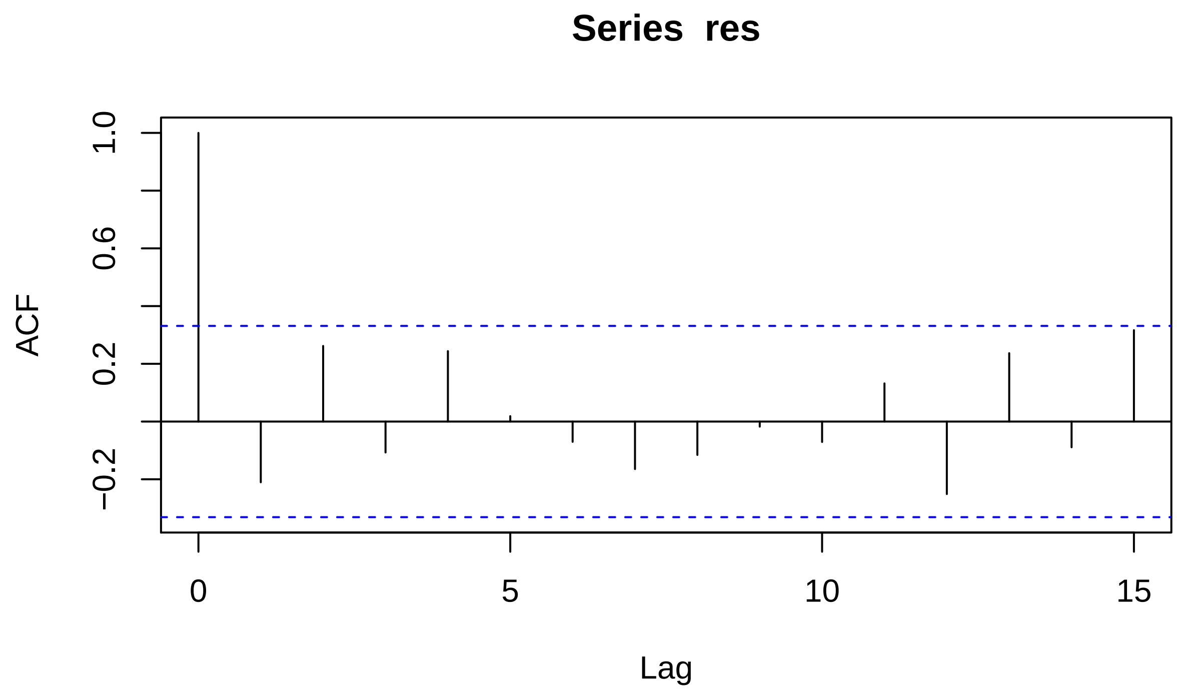
More negative trend than A unfortunately.

```
qqnorm(res,main= "Normal Q-Q Plot for Model B")
qqline(res,col="blue")
```

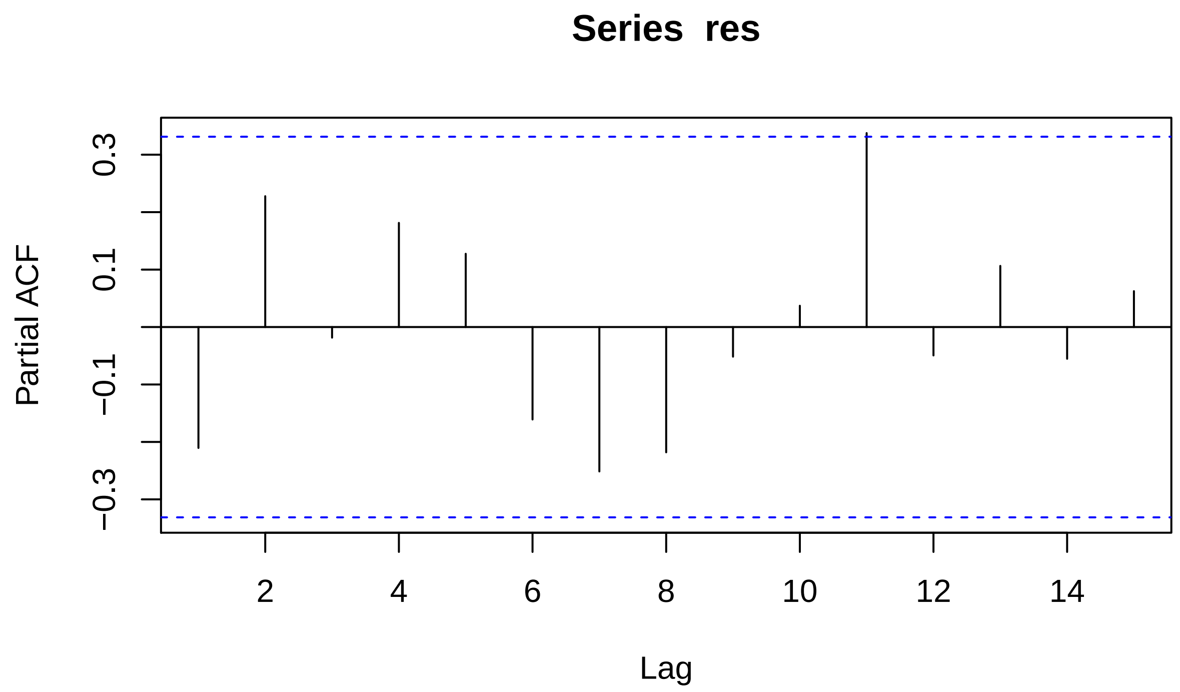


Fairly close to straight line. NormalQQ plot looks good.

```
acf(res)
```



```
pacf(res)
```



ACF and PACF are good, neither has residuals which are statistically significant.

```r
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.98497, p-value = 0.9033
```

Reject when values of W are too small or p value $< .05$. W is large and p-value $= .1861 > .05$. Passes Shapiro-wilk test!

```r
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
##
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.05297
```

Order selected $= 0$, residuals fitted for AR(0)

```r
Box.test(res, lag = 6, type = c("Box-Pierce"), fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 6.6233, df = 4, p-value = 0.1572
```

```r
Box.test(res, lag = 6, type = c("Ljung-Box"), fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 7.5679, df = 4, p-value = 0.1088
```

```r
Box.test(res^2, lag = 6, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  res^2
## X-squared = 2.8259, df = 6, p-value = 0.8304
```

All p-values are larger than .05!

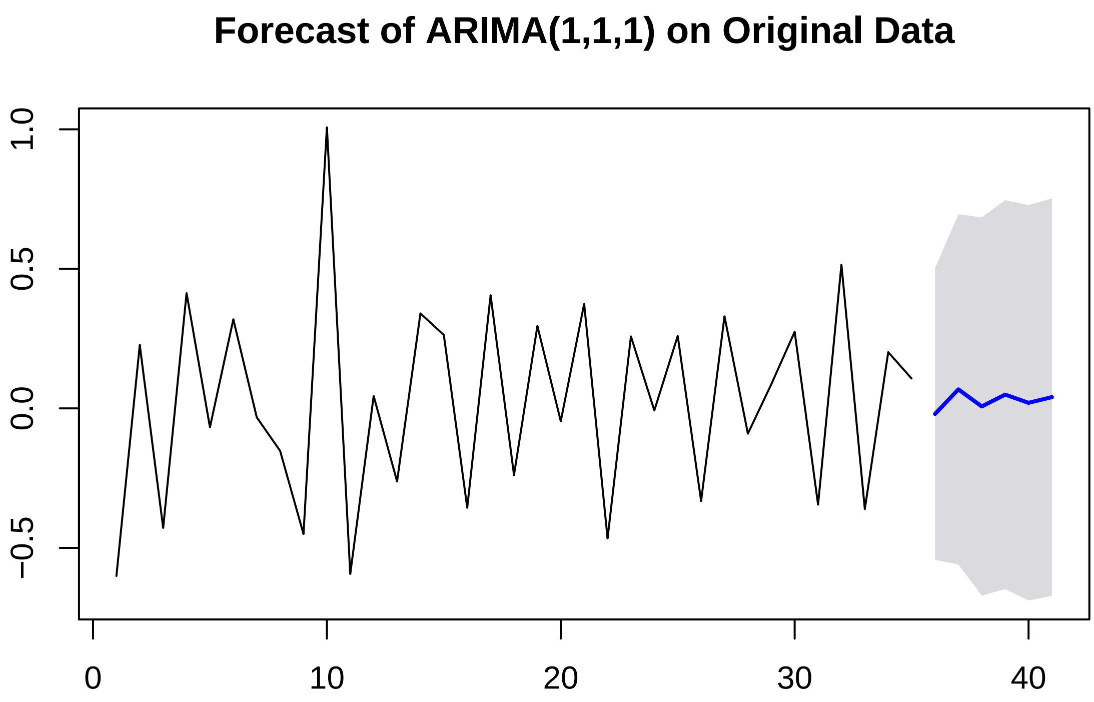
Choose model A because it passed all tests but returned less trend than model B in the graph of residuals. Final model ARIMA(1,1,1)

Coefficients: ar1 ma1 -0.6939 -1.0000 s.e. 0.1248 0.1253

```
fit.A <-  arima(SD_d1, order=c(1,1,1), method="ML")
predict(fit.A, n.ahead = 6)
```

```
## $pred
## Time Series:
## Start = 36
## End = 41
## Frequency = 1
## [1] -0.020129847  0.068028589  0.006858744  0.049302214  0.019852276
## [6]  0.040286488
##
## $se
## Time Series:
## Start = 36
## End = 41
## Frequency = 1
## [1] 0.2666084 0.3201522 0.3457869 0.3555806 0.3614986 0.3634205
```

```
futurVal <- forecast(fit.A,h=6, level=c(95))
plot(futurVal, main = "Forecast of ARIMA(1,1,1) on Original Data")
```



From the graph we can see the forecast, which is the blue line. We also see that with 95% certainty our sales will be inside the dark grey area.

Conclusion: Overall, this project was successful in forecasting future sales for the shampoo business. We can now properly hire employees, stock inventory, and prepare promotions for shampoo. However, this model is far from perfect and would benefit from more shampoo sales data. More data would allow this project to better forecast future sales and maximize profits. Another thing that I think could have helped was if the data was more evenly distributed. It was difficult to find a model that fit the data due to it having large spikes in sales with no relation to trend or season. My best AIC model also didnt have residuals that resembled Gaussian distribution so I had to take my 2nd and 3rd best options, which definitely effected the end result.

References:

Shampoo Data Set: https://raw.githubusercontent.com/jbrownlee/Datasets/master/shampoo.csv