Lab 4

Samir Datta

December 5, 2017

```
library(ggplot2)
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.4.3
```

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

```
## The following object is masked from 'package:ggplot2':
##
##     autolayer
```

```
library(reshape2)
library(xts)
```

```
## Warning: package 'xts' was built under R version 3.4.3
```

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

```
## Warning: package 'zoo' was built under R version 3.4.1
```

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

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

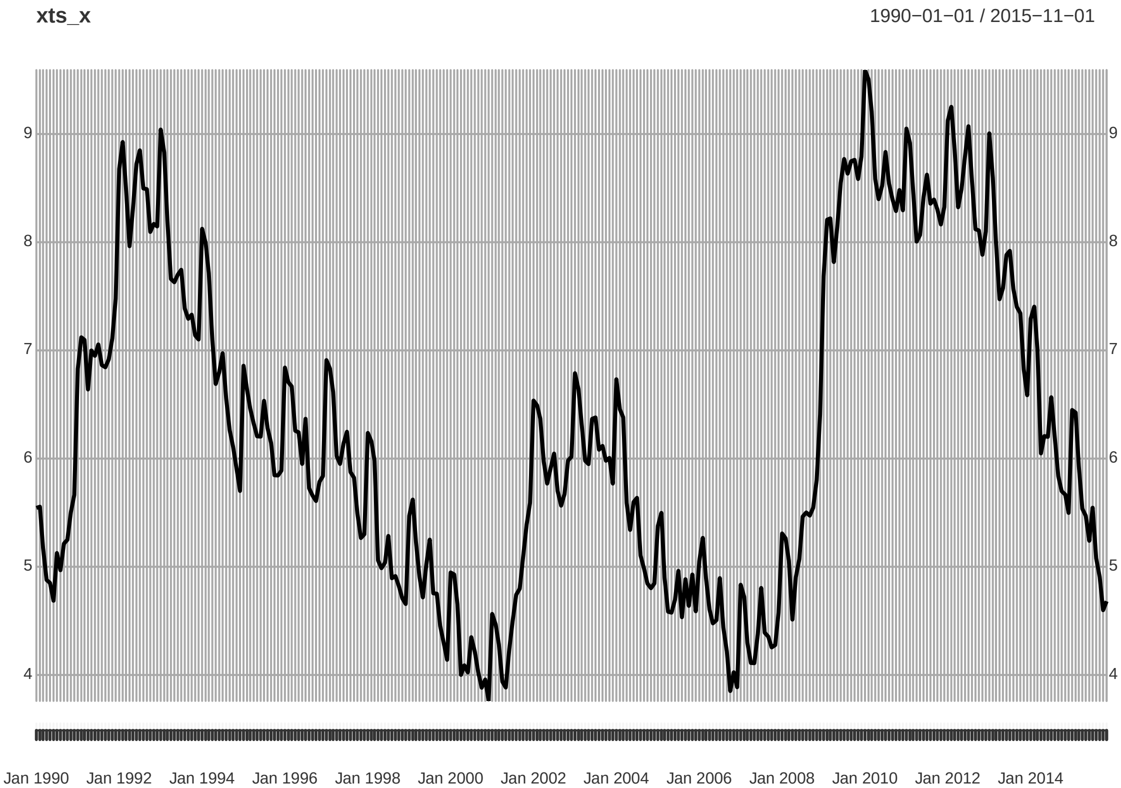
```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 3.4.3
```

```
setwd('C:/Users/Samir/Documents/MIDS/StatsF17/lab 4/')
lab4data <- read.csv('Lab4-series2.csv')
```

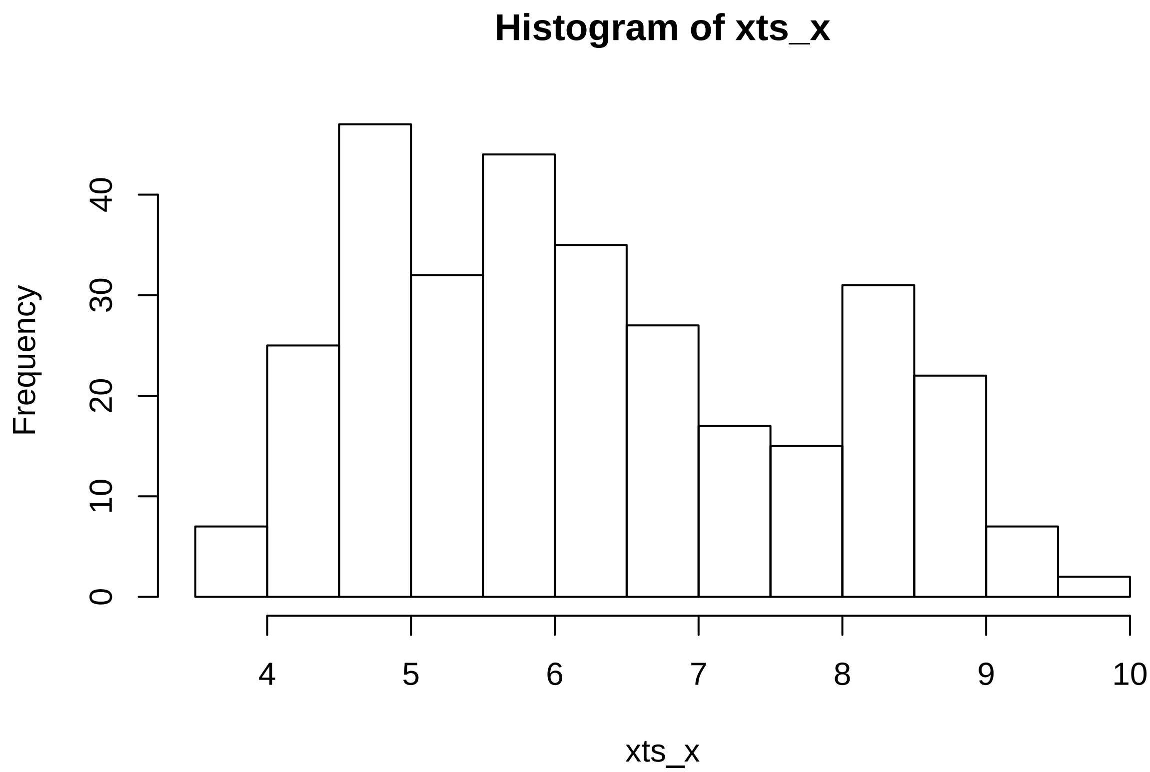
EDA

```
months <- seq.Date(from=as.Date("1990-1-1"),
                   to=as.Date("2015-11-1"), by="month")
xts_x <- xts(lab4data$x, order.by=months)
plot(xts_x)
```



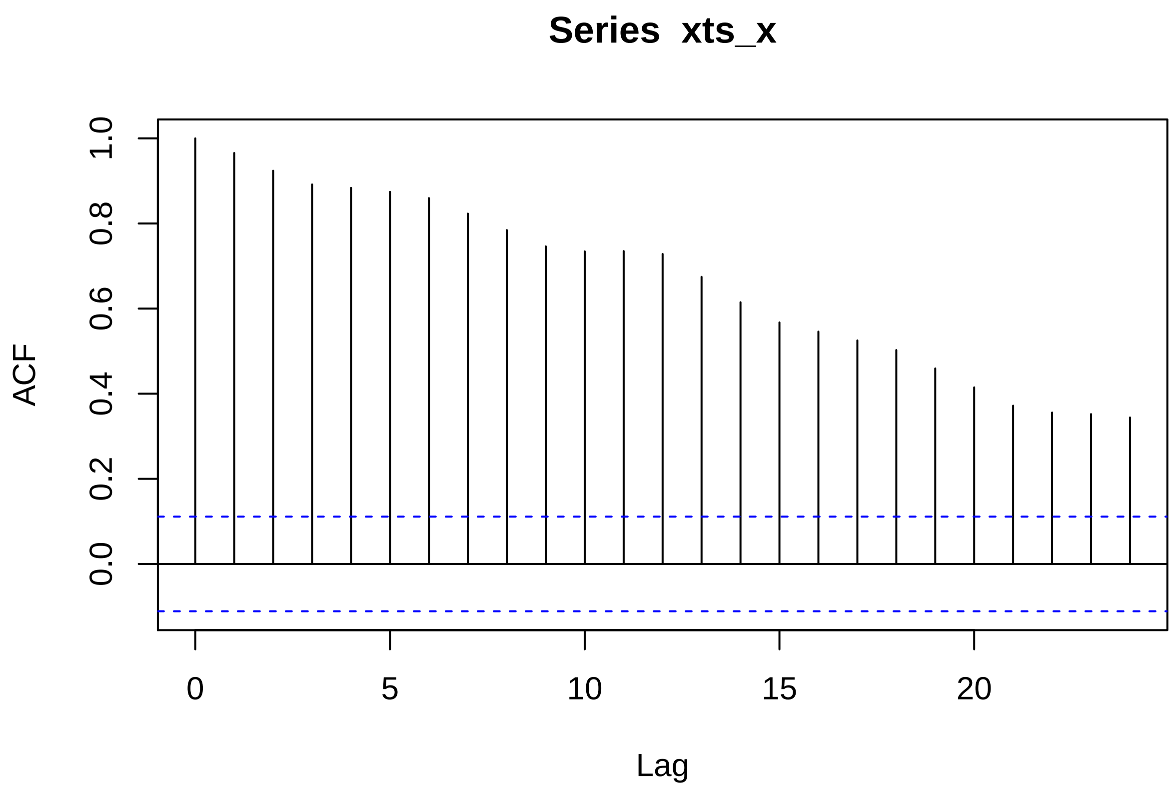
At first glance, appears to be very non-stationary with strong seasonal trends. Seasonal trends appear to be yearly.

```r
hist(xts_x)
```



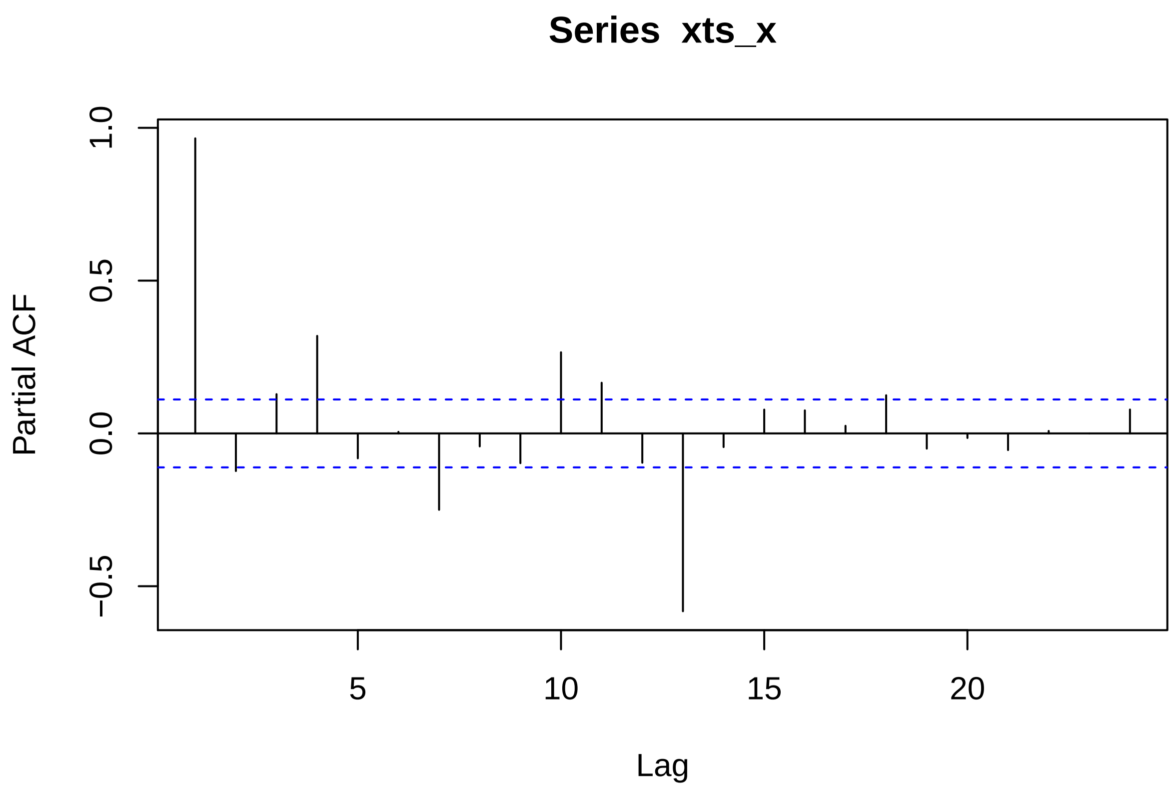
Histogram doesn't seem to suggest any log transformations are necessary.

```
acf(xts_x)
```



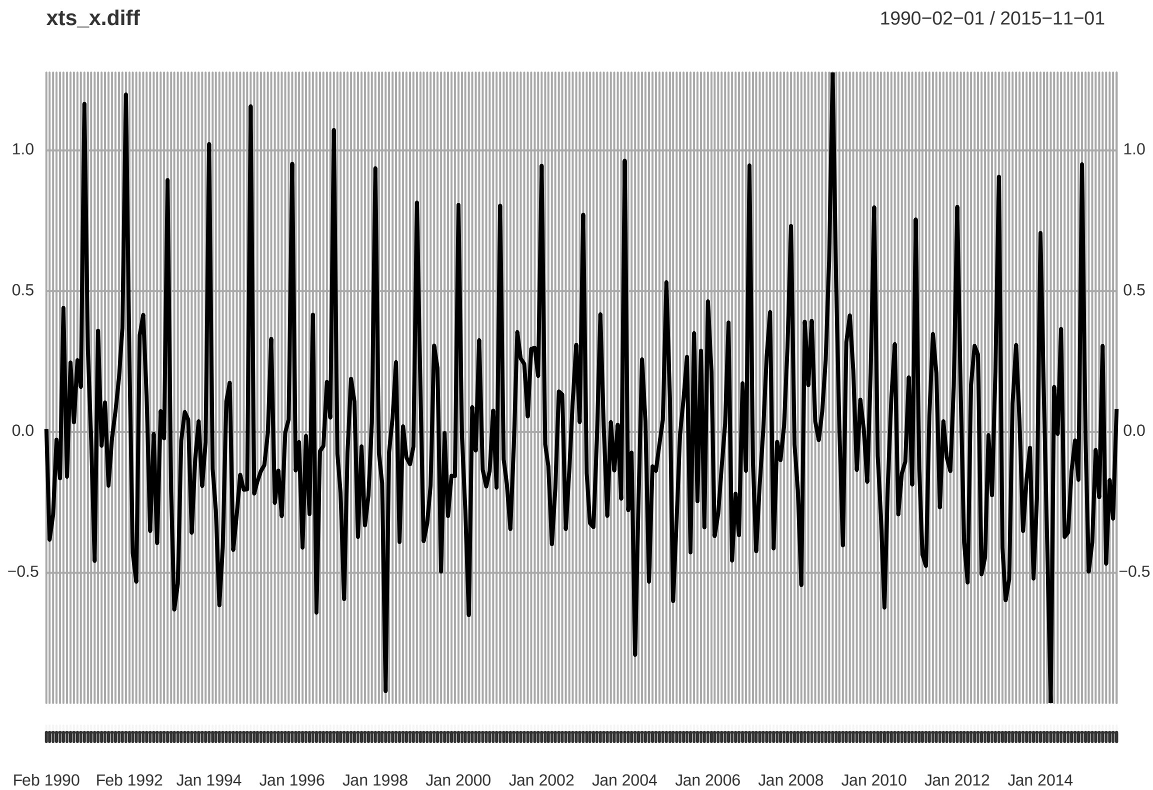
ACF has a gradual decline suggesting an AR model with p of at least 1 would be usful.

```
pacf(xts_x)
```



The strongest PACF occurs at lag 13. (Does this confirm the yearly seasonal trend? I would have expected the significance at lag = 12) There are quite a few other significant PACFs at different lags which suggest non-seasonal , so we will have to test many different parameters.

```
xts_x.diff = diff(xts_x)
xts_x.diff <- xts_x.diff[!is.na(xts_x.diff)]
plot(xts_x.diff)
```



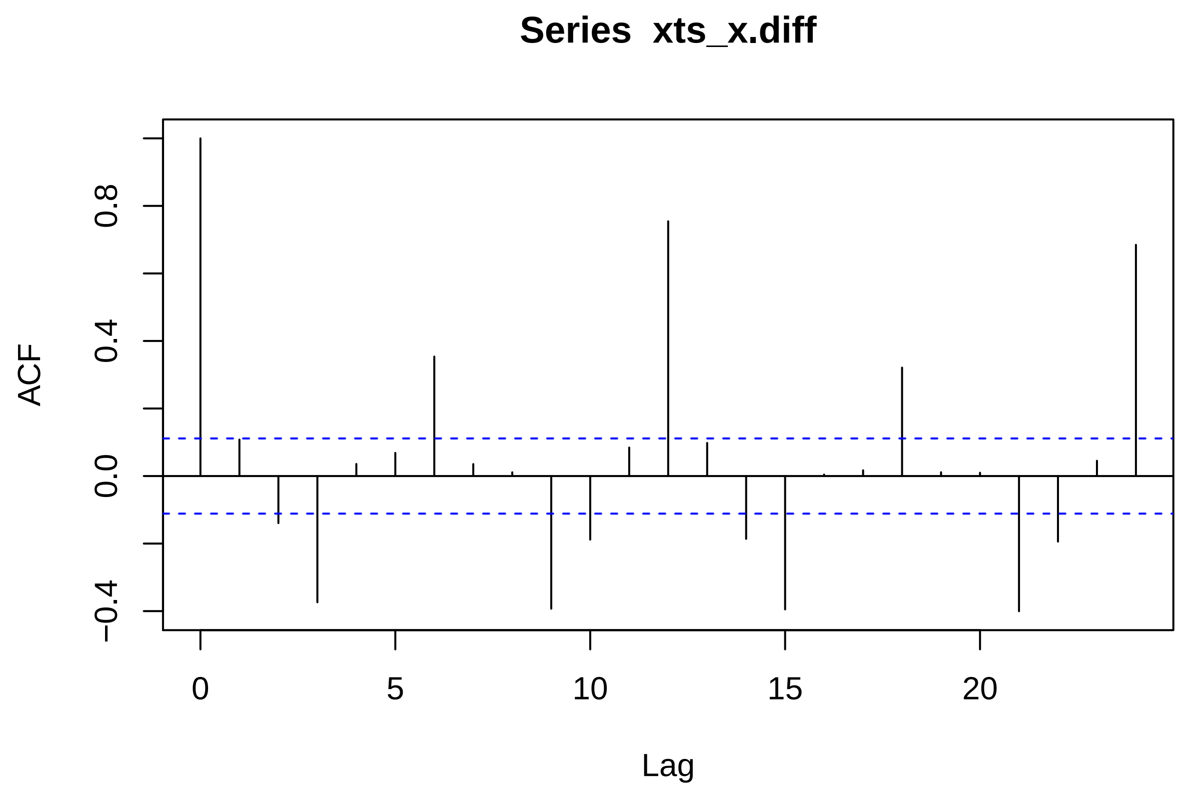

```
adf.test(xts_x.diff)
```

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

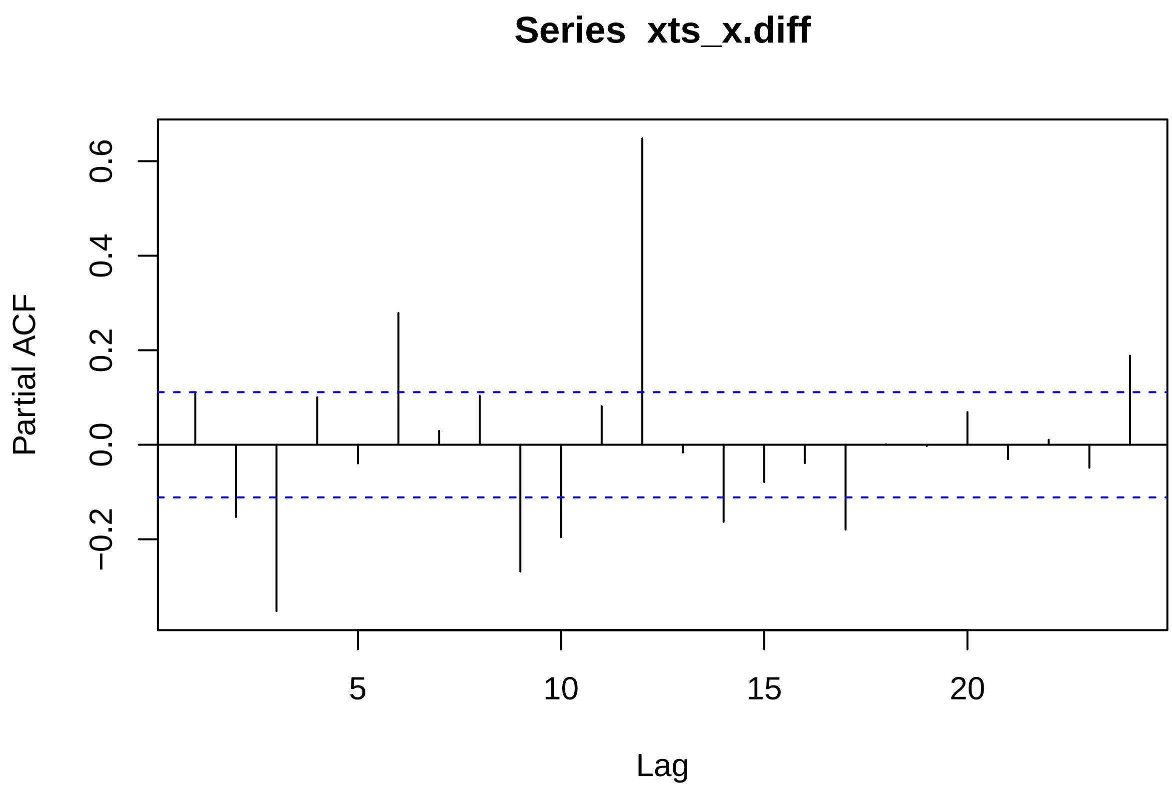
```
##
##  Augmented Dickey-Fuller Test
##
## data:  xts_x.diff
## Dickey-Fuller = -5.6045, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

Plotting the first difference makes it stationary, which is seen from the plot and confirmed by the significance ADF test. But there are clear spikes from the seasonal trend we have to account for.

```
acf(xts_x.diff)
```




```
pacf(xts_x.diff)
```



Both the ACF and PACF for the differences series show a huge spike at lag 12 confirming the yearly seasonal trend.

Loop to find optimal ARIMA

```
x_train <- lab4data[1:300,]$x
x_test <- lab4data[301:311,]$x

results = data.frame(p=NA, d=NA, q=NA, P=NA, D=NA, Q=NA, AIC=NA, RMSE=NA, MAPE=NA)
upperLimit = 2
start = Sys.time()
for(p in 0:upperLimit){
  for(d in 1:1){
    for(q in 0:upperLimit){
      for(P in 0:upperLimit){
        for(D in 1:1){
          for(Q in 0:upperLimit){
tryCatch({Arima.out <- Arima(x_train,
                  order = c(p,d,q),
                   seasonal = list(order=c(P,D,Q), period=12))
AIC <- Arima.out$aic
s <- as.data.frame(summary(Arima.out))
RMSE <- s$RMSE
MAPE <- s$MAPE
```

```
}, warning = function(w){
  AIC <- NA
  RMSE <- NA
  MAPE <- NA
  }, error = function(e){
  AIC <- NA
  RMSE <- NA
  MAPE <- NA
})

result <- data.frame(p=p, d=d, q=q, P=P, D=D, Q=Q, AIC=AIC, RMSE=RMSE, MAPE=MAPE)
results <- rbind(results, result)
          }
        }
      }
    }
  }
}
results <- results[2:nrow(results),]
end = Sys.time()
end-start
```

```
results[results$AIC==min(results$AIC, na.rm=T),]
```

```
##    p d q P D Q       AIC      RMSE    MAPE
## 73 2 1 1 2 1 2 -129.3362 0.1753747 2.28454
```

```
results[results$RMSE==min(results$RMSE, na.rm=T),]
```

```
##    p d q P D Q       AIC      RMSE     MAPE
## 64 2 1 0 2 1 2 -121.4514 0.1743903 2.291954
```

```
results[results$MAPE==min(results$MAPE, na.rm=T),]
```

```
##    p d q P D Q      AIC      RMSE     MAPE
## 55 1 1 2 2 1 2 -129.156 0.1755049 2.283458
```

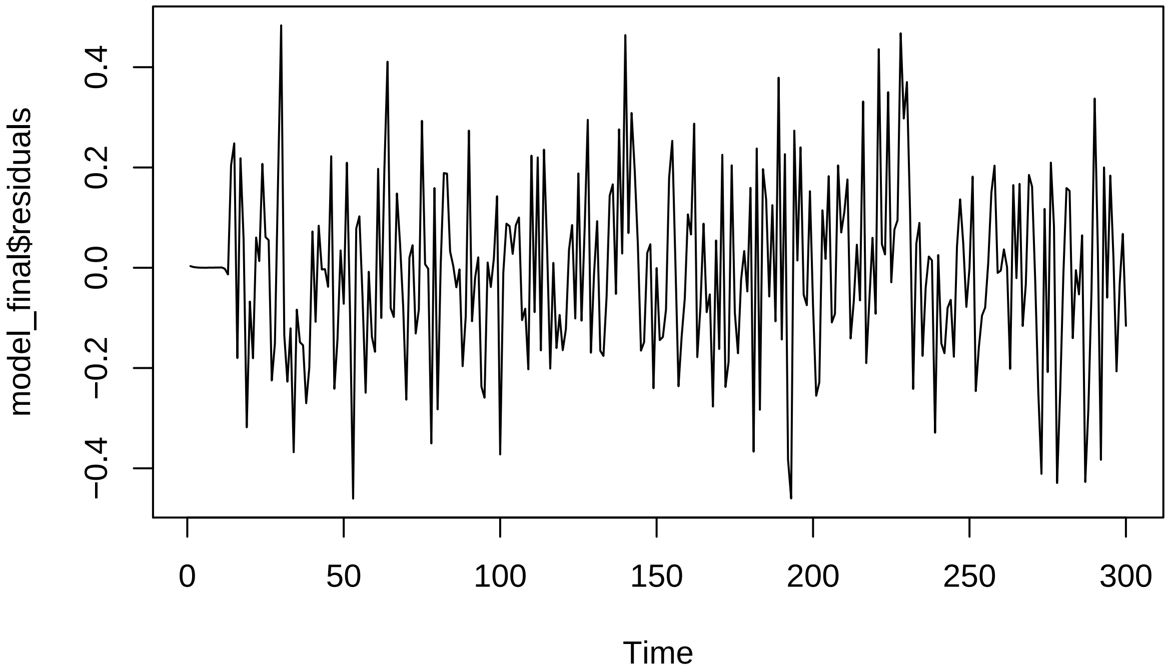
Using minimum AIC/RMSE/MAPE all gives different answers…

```
model_final <- Arima.out <- Arima(lab4data[1:300,]$x,
                  order = c(2,1,1),
                   seasonal = list(order=c(2,1,2), period=12))
summary(model_final)
```

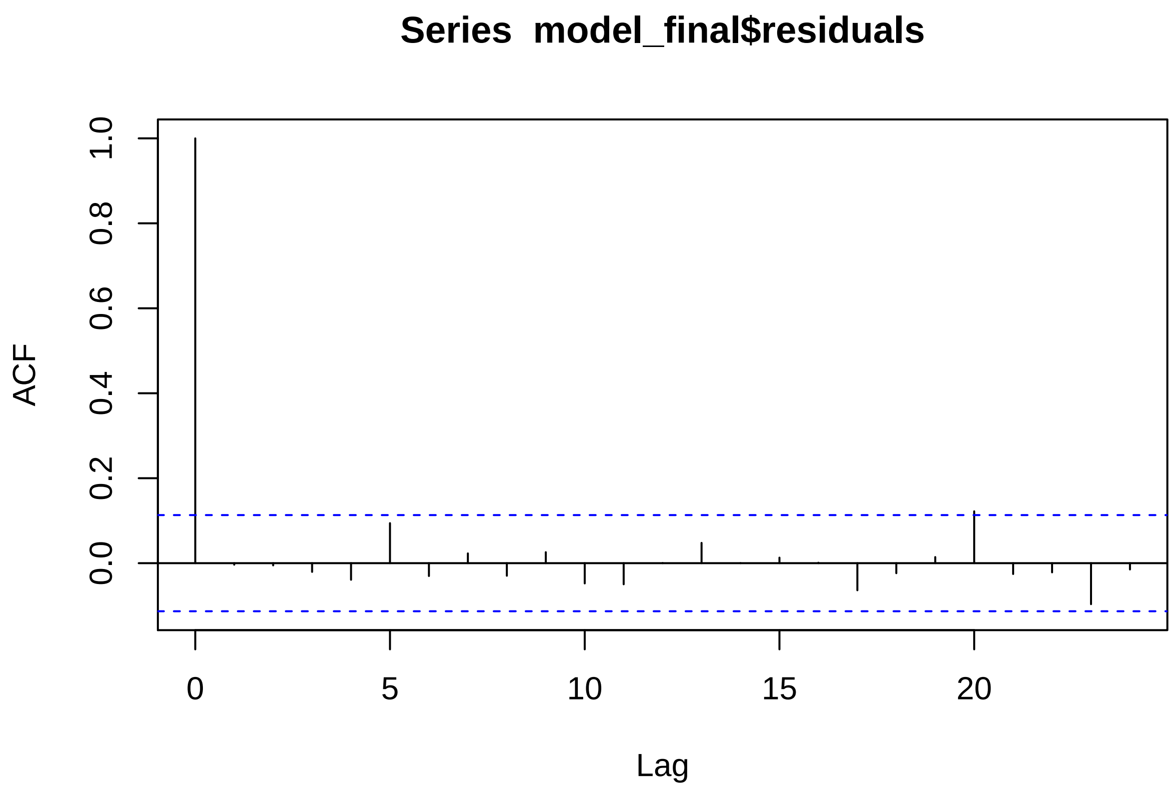
```
## Series: lab4data[1:300, ]$x
## ARIMA(2,1,1)(2,1,2)[12]
##
## Coefficients:
##          ar1     ar2      ma1     sar1    sar2     sma1     sma2
##       0.6849  0.1652  -0.6501  -0.8035  0.1778  -0.0008  -0.8962
## s.e.  0.0902  0.0633   0.0844   0.0755  0.0739   0.0839   0.0751
##
## sigma^2 estimated as 0.03295:  log likelihood=72.67
## AIC=-129.34   AICc=-128.82   BIC=-100.06
##
## Training set error measures:
##                         ME      RMSE       MAE         MPE    MAPE     MASE
```

```
## Training set -0.008473527 0.1753747 0.1367246 -0.09983992 2.28454 0.485703
##                        ACF1
## Training set -0.003564555
```

```r
plot(model_final$residuals)
```




```r
acf(model_final$residuals)
```



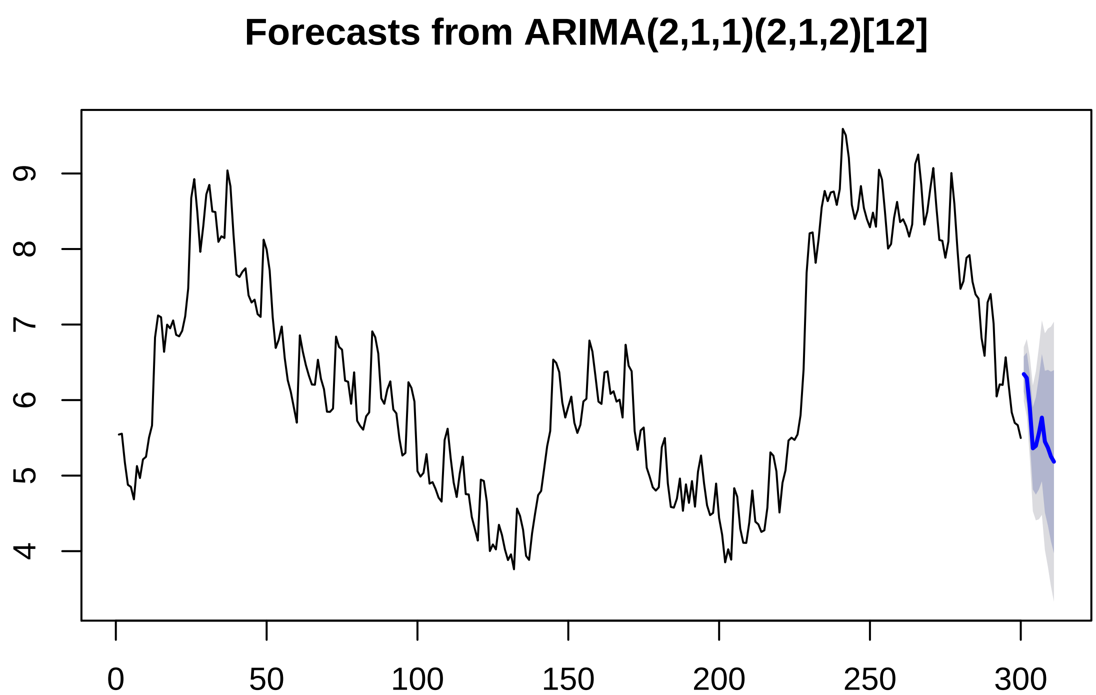

```r
adf.test(model_final$residuals)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  model_final$residuals
## Dickey-Fuller = -6.1841, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

The residuals of the final model, based on the plot, ACF, and ADF test, are stationary.

```r
model_forecast <- forecast(model_final, h = 11)

plot(model_forecast)
```




```r
predicted <- as.numeric(model_forecast$mean)
actual <- lab4data[301:311,]$x

mean(abs((actual-predicted)/actual) * 100)
```

```
## [1] 5.504884
```

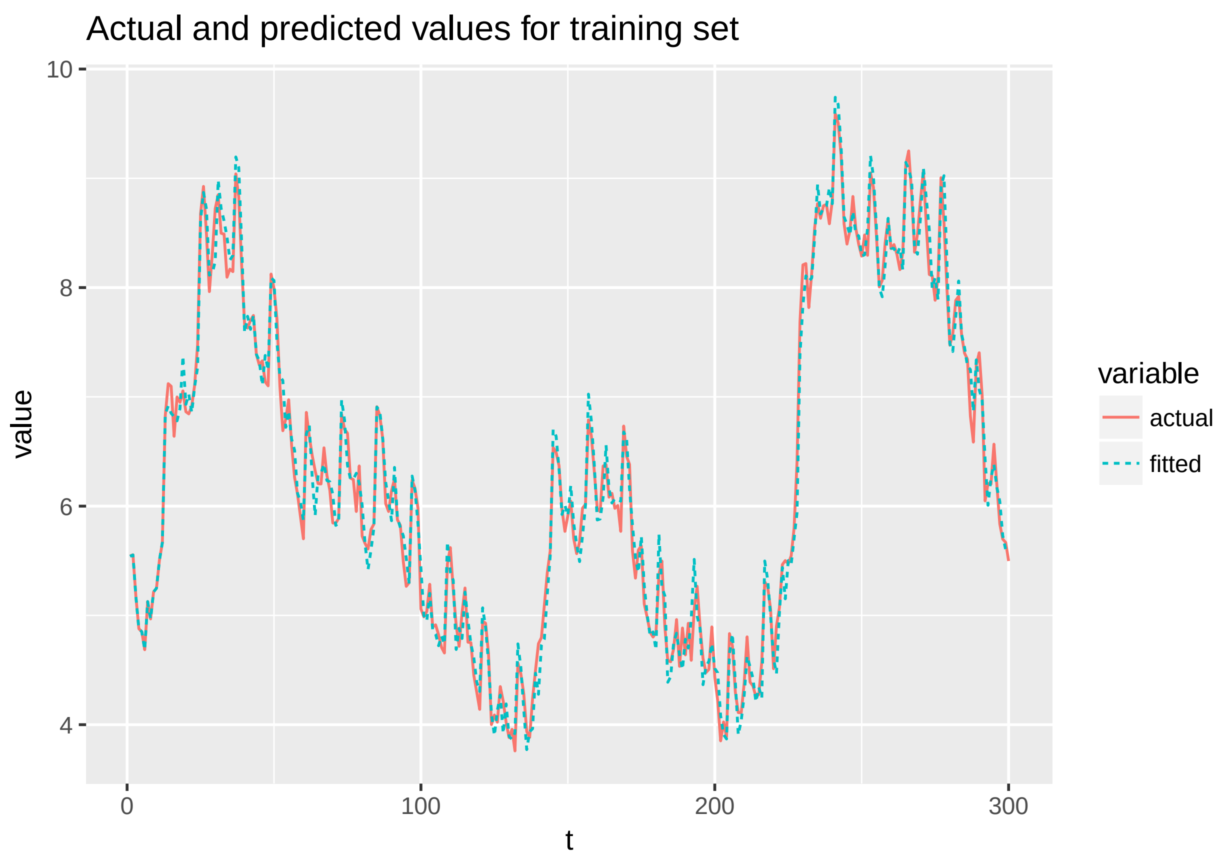
```r
model_final_df <- data.frame(actual = x_train, fitted = as.numeric(model_final$fitted))
df_melt <- cbind(melt(model_final_df), rbind(cbind(1:300), cbind(1:300)))
```

```
## No id variables; using all as measure variables
```

```r
head(df_melt)
```

```
##   variable value rbind(cbind(1:300), cbind(1:300))
## 1   actual 5.544                                 1
## 2   actual 5.555                                 2
## 3   actual 5.172                                 3
## 4   actual 4.878                                 4
## 5   actual 4.851                                 5
## 6   actual 4.686                                 6
```

```r
colnames(df_melt) <- c("variable", "value", "t")
df_melt$variable <- factor(df_melt$variable, levels=c("actual", "fitted"))

ggp <- ggplot(df_melt, aes(x=t, y=value, group=variable, color=variable))
ggp+geom_line(aes(linetype=variable))+
  ggtitle("Actual and predicted values for training set")
```




```
model_forecast_df <- data.frame(actual = x_test,
                                forecasted = as.numeric(model_forecast$mean))
df_melt <- cbind(melt(model_forecast_df), rbind(cbind(1:11), cbind(1:11)))
```

```
## No id variables; using all as measure variables
```

```
head(df_melt)
```

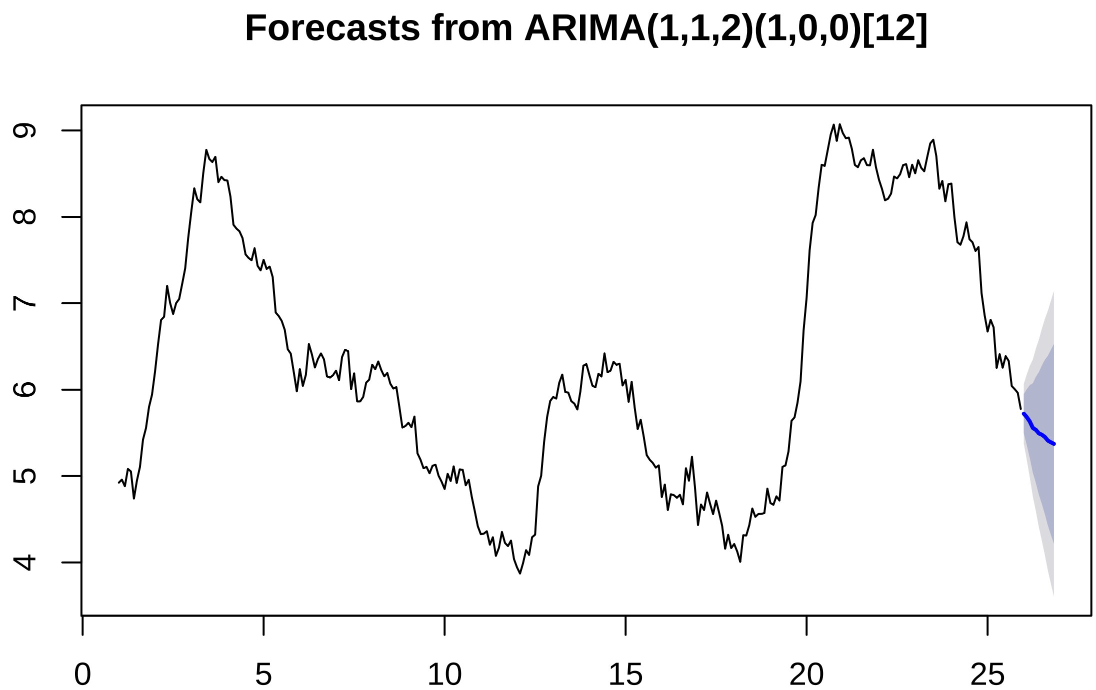
```
##   variable value rbind(cbind(1:11), cbind(1:11))
## 1   actual 6.449                               1
## 2   actual 6.425                               2
## 3   actual 5.929                               3
## 4   actual 5.536                               4
## 5   actual 5.472                               5
## 6   actual 5.240                               6
```

```
colnames(df_melt) <- c("variable", "value", "t")
df_melt$variable <- factor(df_melt$variable, levels=c("actual", "forecasted"))

ggp <- ggplot(df_melt, aes(x=t, y=value, group=variable, color=variable))
ggp+geom_line(aes(linetype=variable))+
  ggtitle("Actual and forecasted values for test set")
```


```
ts2 <- ts(lab4data[0:300,]$x, frequency=12)
dsts2 <- seasadj(stl(ts2, s.window="periodic"))
auto.arima.out <- auto.arima(dsts2, seasonal=T)
summary(auto.arima.out)
```

```
## Series: dsts2
## ARIMA(1,1,2)(1,0,0)[12]
##
## Coefficients:
##          ar1      ma1     ma2    sar1
##       0.8714  -0.8421  0.1377  0.0777
## s.e.  0.0604   0.0813  0.0591  0.0602
##
## sigma^2 estimated as 0.03117:  log likelihood=96.08
## AIC=-182.16   AICc=-181.95   BIC=-163.66
##
## Training set error measures:
##                        ME      RMSE       MAE        MPE     MAPE     MASE
## Training set 0.0003683998 0.1750703 0.1388201 0.01981815 2.326253 0.171446
##                        ACF1
## Training set -0.0006323352
```

```
model_forecast <- forecast(auto.arima.out, h = 11)

plot(model_forecast)
```



The autoarima function with seasonal decomposition has a much lower AIC, suggesting better in-sample fit, but the forecast looks bad. The coefficient of the seasonal component in the arima output is very small and not significant which probably explains the bad fit... not worth it?