W271 Section 3 Lab 4

Kiersten Henderson, Zhaoning Yu, Daghan Altas

12/09/2017

I. Introduction

II. Loading and cleaning up the data

We'll load the data and convert it to an xts object for easy subsetting

```
setwd("/Users/daghanaltas/Hacking/Berkeley/W271/Labs/w271_lab4")
df <- read.csv("./Lab4-series2.csv")
rbind(head(df), tail(df))
```

```
##       X     x
## 1     1 5.544
## 2     2 5.555
## 3     3 5.172
## 4     4 4.878
## 5     5 4.851
## 6     6 4.686
## 306 306 5.240
## 307 307 5.546
## 308 308 5.078
## 309 309 4.907
## 310 310 4.599
## 311 311 4.681
```

```
str(df)
```

```
## 'data.frame':    311 obs. of  2 variables:
##  $ X: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ x: num  5.54 5.55 5.17 4.88 4.85 ...
```

```
sum(is.na(df))  # check if there is any NA
```

```
## [1] 0
```

There are no missing variables and the first column is the index column, which can be discarded. We are going to convert the data to a (xts) based time seres

```
ms <- as.xts(ts(df$x, start = c(1990, 1), frequency = 12))
ms.training <- ms["/2014"]
rbind(head(ms.training), tail(ms.training))
```

```
##           [,1]
## Jan 1990 5.544
## Feb 1990 5.555
## Mar 1990 5.172
## Apr 1990 4.878
## May 1990 4.851
## Jun 1990 4.686
## Jul 2014 6.567
```

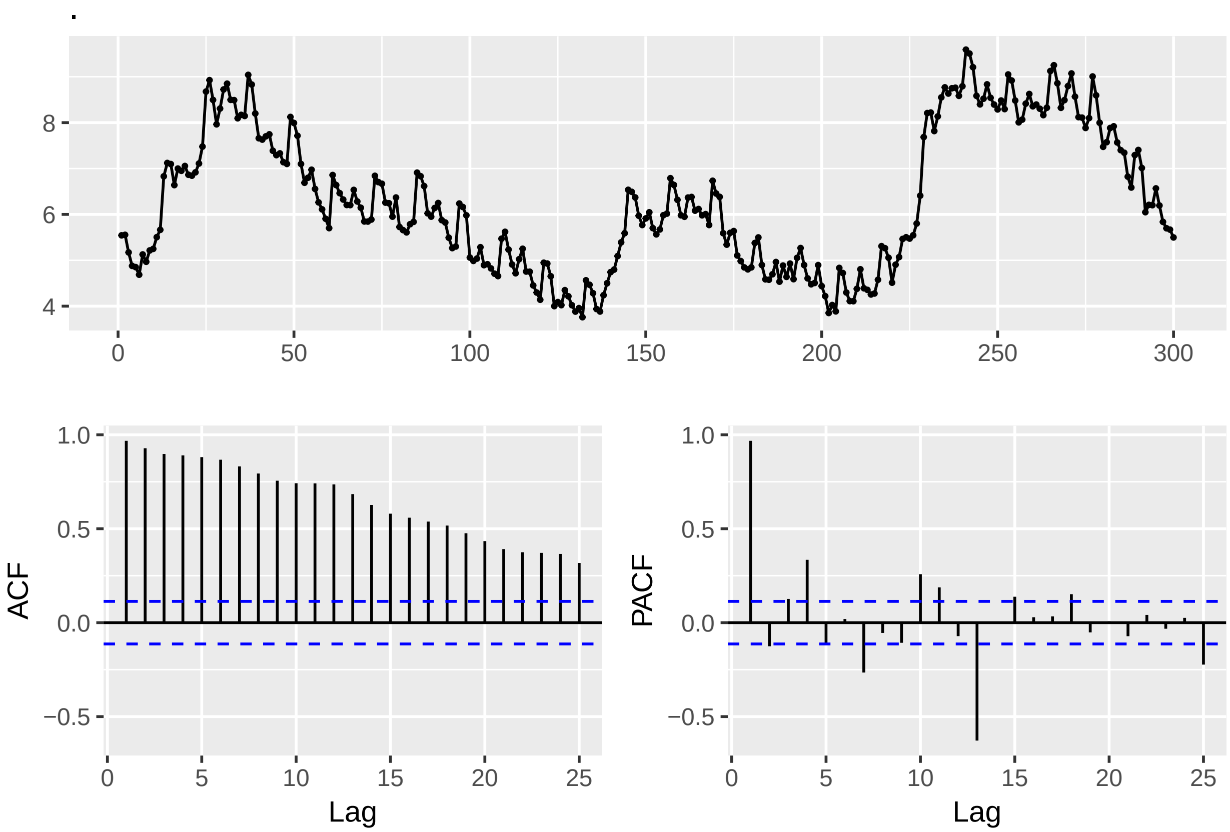
```
## Aug 2014 6.194
## Sep 2014 5.837
## Oct 2014 5.698
## Nov 2014 5.668
## Dec 2014 5.498
```

```
ms.test <- ms["2015/"]
```

III. EDA

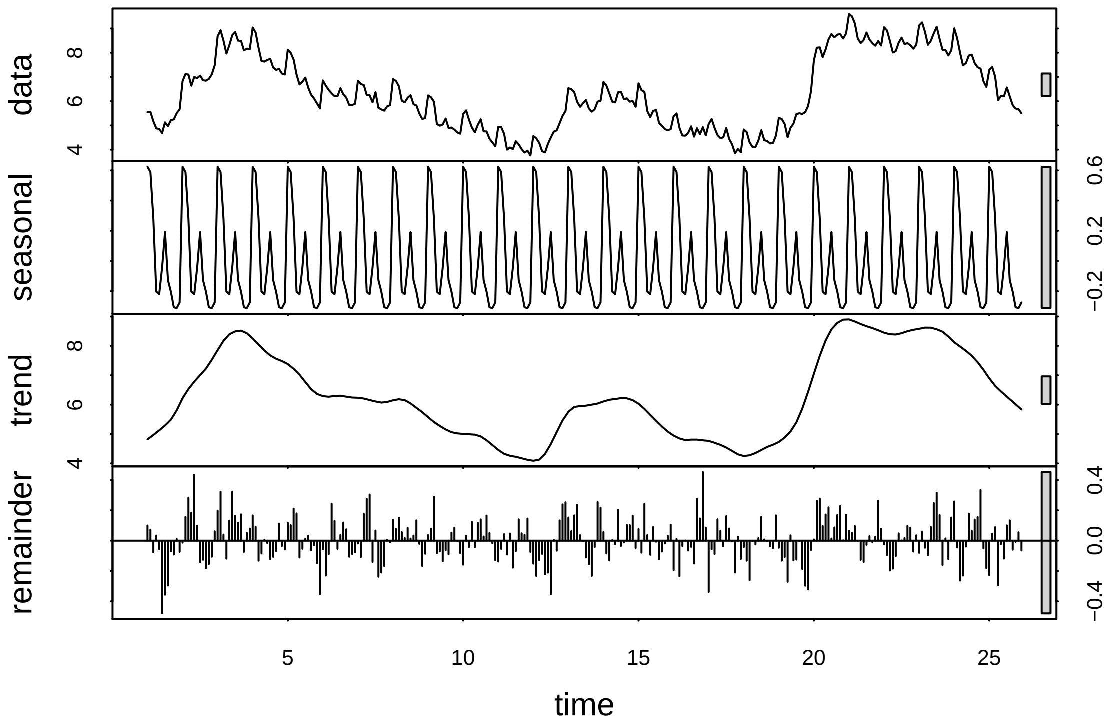
We first plot the time series together with its ACF and PACF.

```
ms.training %>% ggtsdisplay
```



We also use STL decomposition (HA ch6.5) to decompose the series into seasonal and trend components.

```
fit.stl <- stl(ms.training, t.window = 15, s.window = "periodic",
    robust = TRUE)
plot(fit.stl)
```

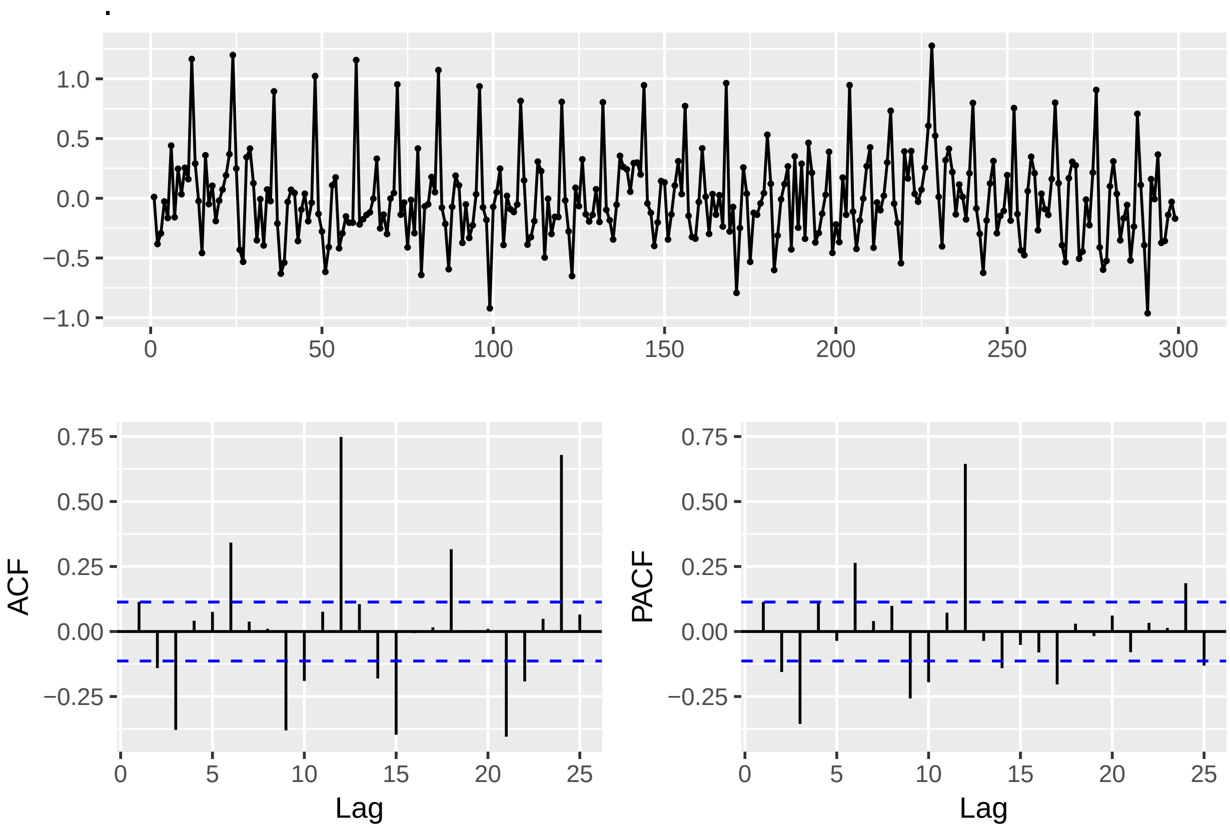


The series both show a trend and a seasonal component. It is not stationary in the mean. This indicates the need for differencing to stabilize the mean.

IV. Transformations

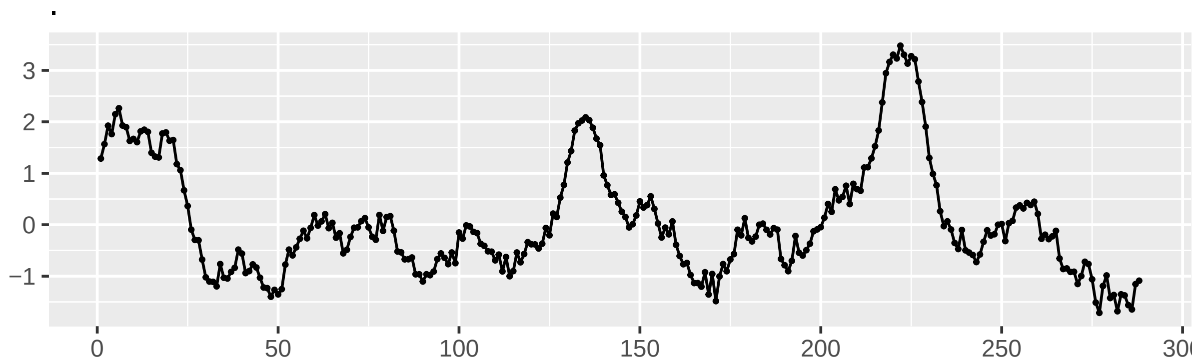
We try both the first and the seasonal differencing.

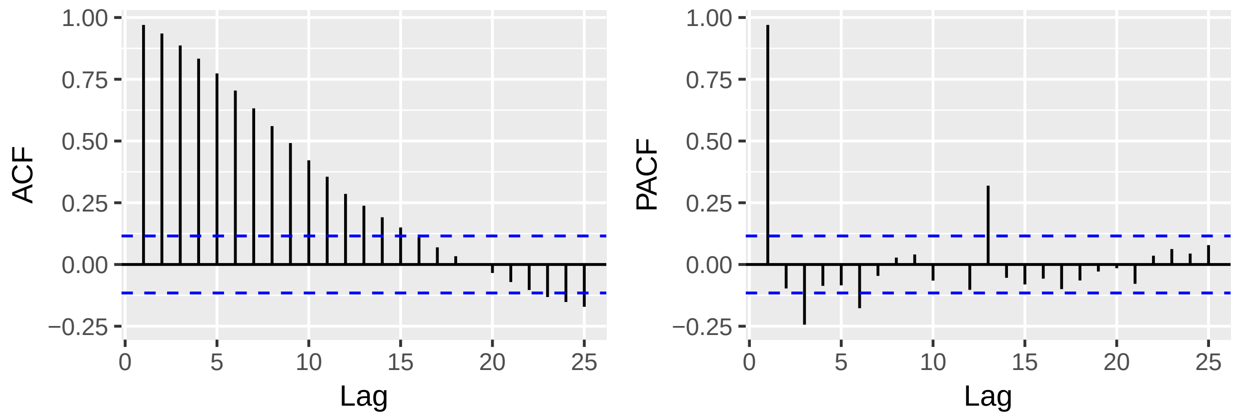
```
# First differencing only
ms.training.1d <- diff(ms.training, lag = 1)
ms.training.1d <- ms.training.1d[!is.na(ms.training.1d)]
ms.training.1d %>% ggtsdisplay
```



With only the first-differencing, the data are clearly non-stationary with strong seasonality.

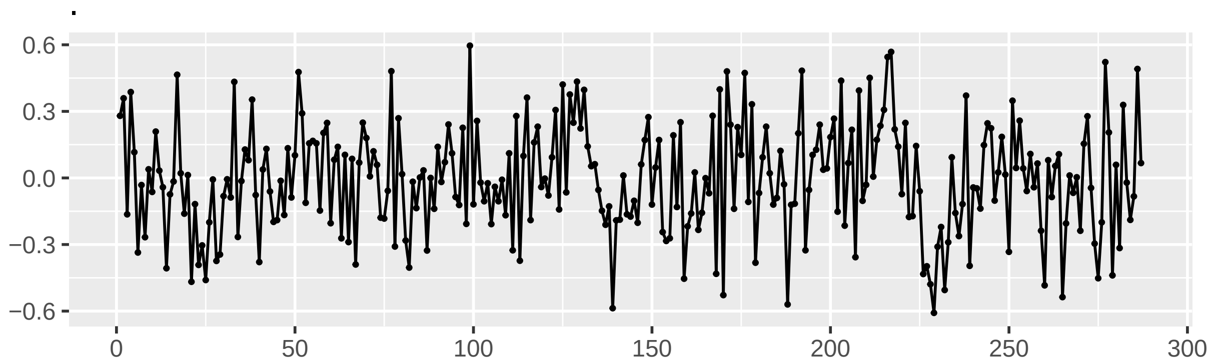
```r
# Seasonal differencing only
ms.training.12d <- diff(ms.training, lag = 12)
ms.training.12d <- ms.training.12d[!is.na(ms.training.12d)]
ms.training.12d %>% ggtsdisplay
```

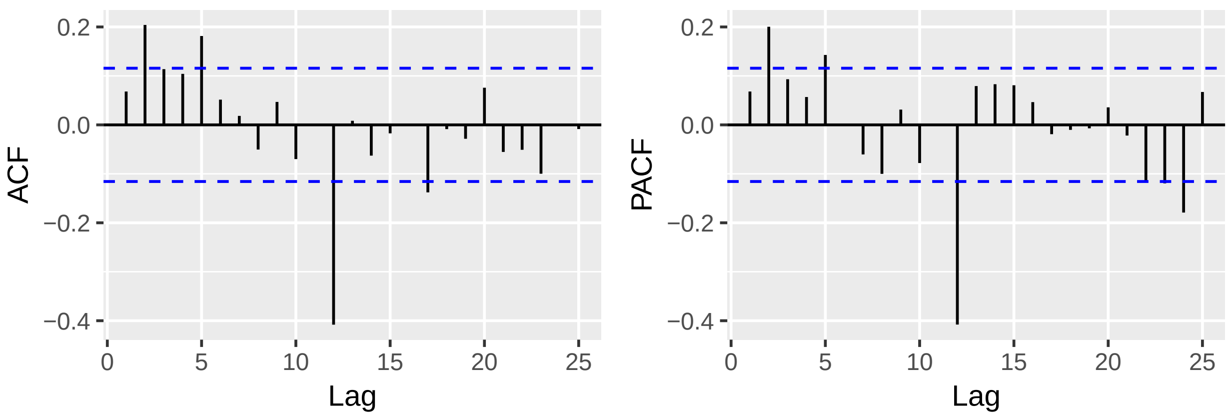




With only seasonal differencing, the data are clearly non-stationary.

```
# Both the first and the seasonal differencing
ms.training.1d.12d <- diff(diff(ms.training, lag = 1), lag = 12)
ms.training.1d.12d <- ms.training.1d.12d[!is.na(ms.training.1d.12d)]
ms.training.1d.12d %>% ggtsdisplay
```





We decide to do another difference after the seasonally difference, the data appear sufficiently stabilized.

V. Model search

In the plots of the differenced data, there are spikes in the PACF at lags 12, 24, 36 .. and a spike in ACF at lag 12, suggesting a seasonal MA(1) component.

There are significant spikes at lags 2, 5 in both the ACF and PACF, suggesting a possible MA(2) or AR(2) term, however, the choice is not obvious.

We decide to start with an ARIMA(0,1,2)(0,1,1)[12] and manually fit some variations on it to identify the models with the lowest AIC and AICc values. In addition, we also consider the out-of-sample performance (MAPE) on the testing data.

Define a function for model testing

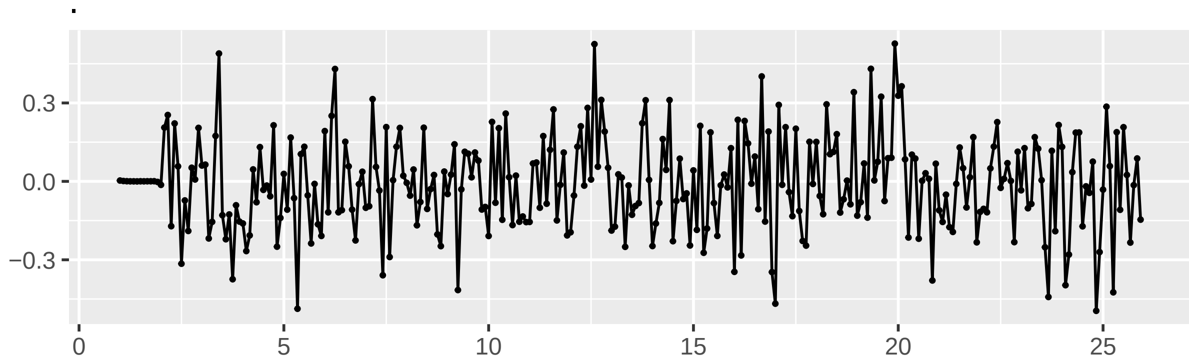
Since the procedure is repetitive, we define a function for model testing:

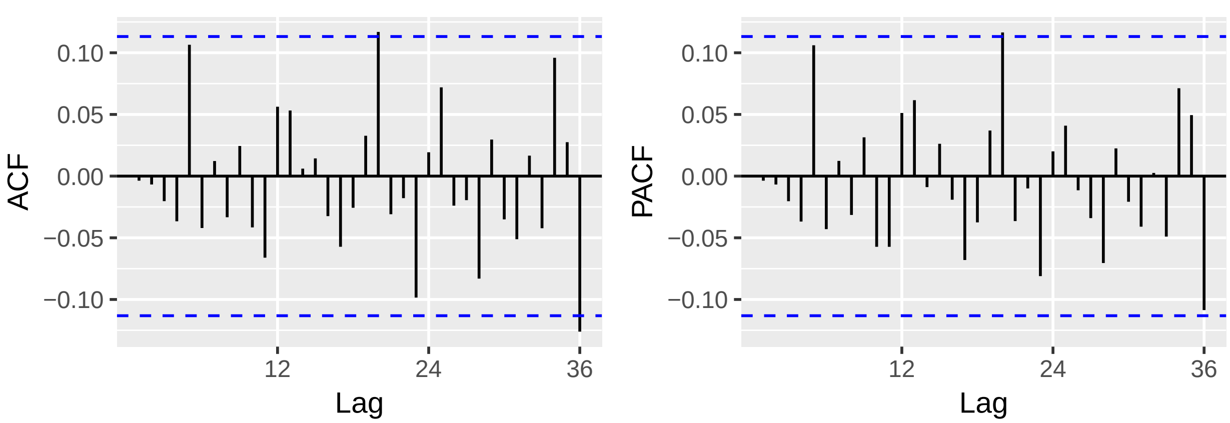
```
# Define a function for testing models
model.test <- function(ORDER, SEASONAL) {

    fit.test <- Arima(ms.training, order = ORDER, seasonal = SEASONAL)
    fit.test$residuals %>% ggtsdisplay  # residual plot

    # find MAPE
    f1 <- ms.training %>% Arima(order = ORDER, seasonal = list(order = SEASONAL,
        period = 12)) %>% forecast(h = 11)
```

```r
    # return AIC, AICc, BIC, MAPE.train, MAPE.test
    temp <- cbind(fit.test[6], fit.test[15], fit.test[16], accuracy(f1,
        ms.test)[1, 5], accuracy(f1, ms.test)[2, 5])
    colnames(temp) = c("AIC", "AICc", "BIC", "MAPE.train", "MAPE.test")
    rownames(temp) = NULL
    temp
}
```

Model testing

```r
# Define the model to be tested
Order = c(2, 1, 2)  # order
Seasonal = c(0, 1, 1)  # seasonal component

model.test(Order, Seasonal)
```






```
##      AIC       AICc      BIC      MAPE.train MAPE.test
## [1,] -123.8599 -123.5599 -101.903 2.364426   3.78801
```

Summary of Models

Candidate Models

ARIMA AIC AICc BIC MAPE.train MAPE.test

(0,1,5)(0,1,1)[12] -120.65 -120.25 -95.03 2.36 5.88

(0,1,6)(0,1,1)[12] -120.24 -119.72 -90.96 2.35 5.66

(1,1,1)(0,1,1)[12] -122.71 -122.57 -108.08 2.39 3.32

(1,1,2)(0,1,1)[12] -125.81 -125.60 -107.52 2.36 3.85

(1,1,3)(0,1,1)[12] -123.83 -123.53 -101.87 2.36 3.81

(1,1,1)(0,1,2)[12] -122.23 -122.01 -103.93 2.38 3.67

(2,1,1)(0,1,1)[12] -125.86 -125.64 -107.56 2.36 3.77

(3,1,1)(0,1,1)[12] -123.86 -123.56 -101.9 2.36 3.78

(2,1,1)(1,1,1)[12] -125.55 -125.25 -103.59 2.35 4.11

(1,1,1)(1,1,1)[12] -122.40 -122.19 -104.10 2.38 3.69

(2,1,1)(0,1,2)[12] -125.38 -125.08 -103.42 2.35 4.11

(2,1,2)(0,1,1)[12] -123.86 -123.56 -101.90 2.36 3.79

Grid Search

We'll now conduct a grid search to see if any other model provide an enhancement over these models.

```
results <- data.frame(p = 1:25, q = 1:25, AIC = 0, AICc = 0,
    BIC = 0)
for (p in 1:5) {
    for (q in 1:5) {
        m <- ms.training %>% Arima(order = c(p, 1, q), seasonal = list(order = c(0,
            1, 1), period = 12))
        index <- (p - 1) * 5 + q
        results[index, ] = c(p, q, m$aic, m$aicc, m$bic)
    }
}
results[which.min(results$AIC), ]
```

```
##   p q       AIC     AICc       BIC
## 6 2 1 -125.8566 -125.643 -107.5591
```

```
results[which.min(results$AICc), ]
```

```
##   p q       AIC     AICc       BIC
## 6 2 1 -125.8566 -125.643 -107.5591
```

```
results[which.min(results$BIC), ]
```

```
##   p q       AIC      AICc       BIC
## 1 1 1 -122.7133 -122.5715 -108.0754
```

Candidate models

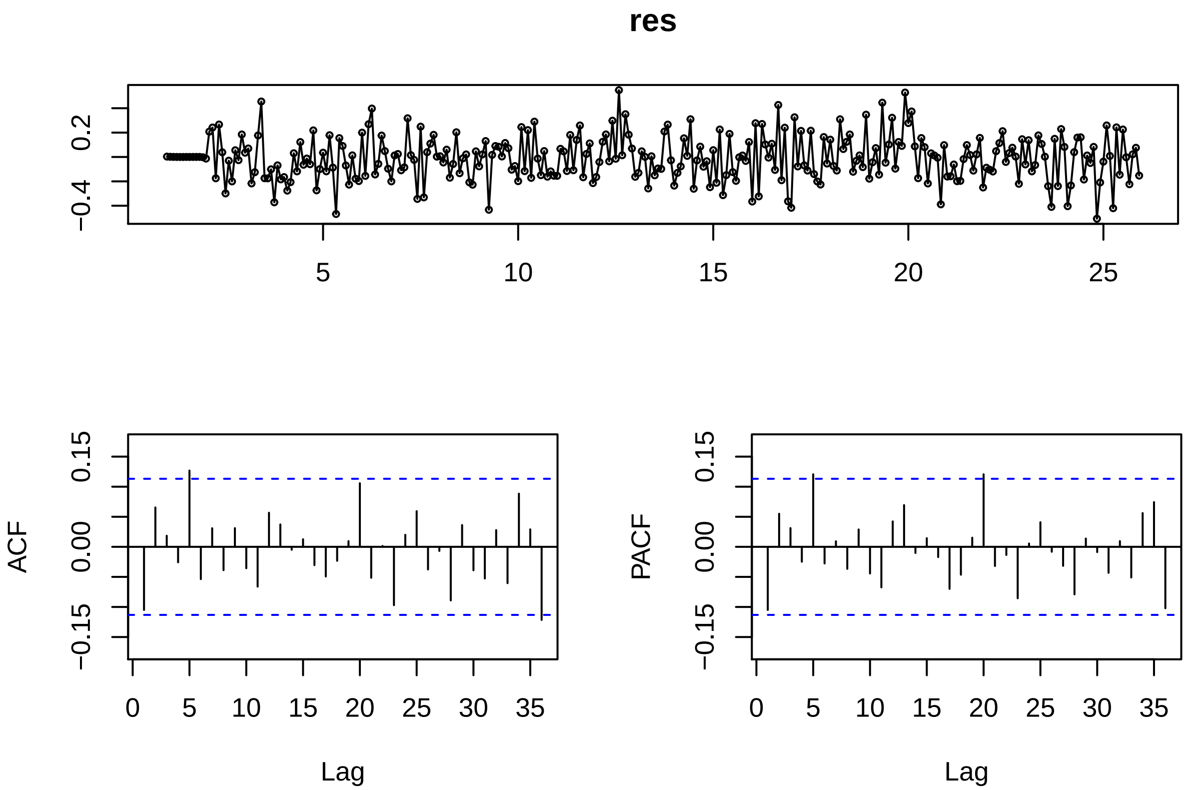
Based on the exploratory analysis and grid search, we are going to focus on the following models:

- ARIMA(1,1,1)(0,1,1)[12] (minimizes BIC, best out-of-sample performance)
- ARIMA(2,1,1)(0,1,1)[12] (minimizes AIC / AICc)

VI. Test the selected models

ARIMA(1,1,1)(0,1,1)[12]: lowest BIC and best out-of-sample performance

```
fit <- Arima(ms.training, order = c(1, 1, 1), seasonal = c(0,
    1, 1))
res <- residuals(fit)
tsdisplay(res)
```

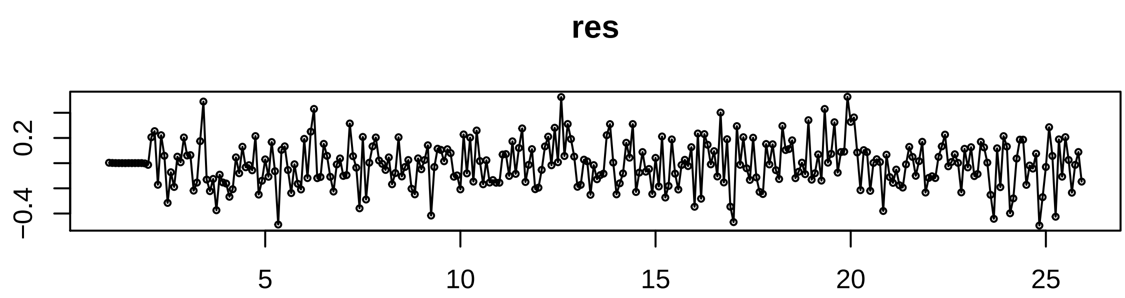


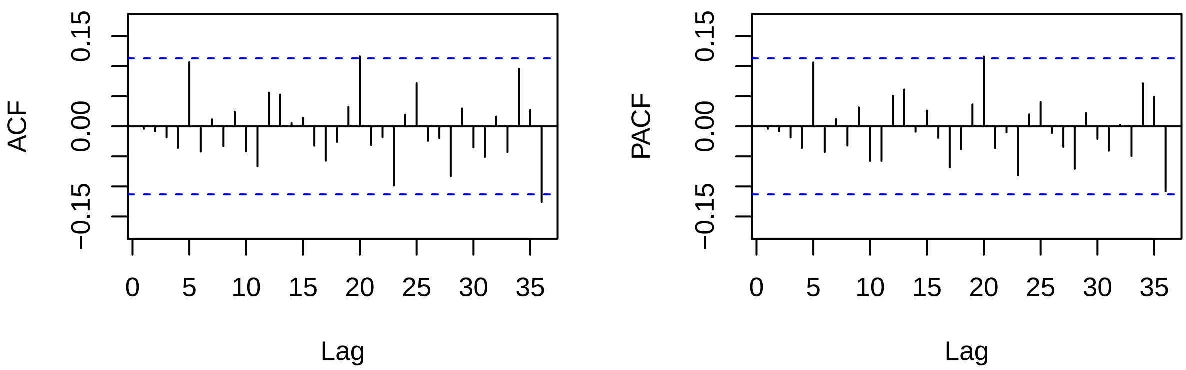

```
Box.test(res, lag = 16, fitdf = 4, type = "Ljung")  # p-value = 0.2188
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 15.428, df = 12, p-value = 0.2188
```

```
# Box.test(res, lag=36, fitdf=6, type='Ljung') # p-value =
# 0.1085

# QUESTION: what parameters to use for the Box.test????
```

```
## ARIMA(2,1,1)(0,1,1)[12]: lowest AIC, AICc
fit <- Arima(ms.training, order = c(2, 1, 1), seasonal = c(0,
    1, 1))
res <- residuals(fit)
tsdisplay(res)
```






```
Box.test(res, lag = 16, fitdf = 4, type = "Ljung")   # p-value = 0.6722
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 9.3568, df = 12, p-value = 0.6722
```

```
# Box.test(res, lag=36, fitdf=6, type='Ljung') # p-value =
# 0.2596

# QUESTION: what parameters to use for the Box.test????
```

The results for both models are similar:

- We can ignore the 2 spikes outside the 95% significant limits, the residuals appear to be white noise.
- A Ljung-Box test also shows that the residuals have no remaining auto-correlations.

VII. Forecast

We do a 11-month ahead forecast of the series in 2015 using both models.

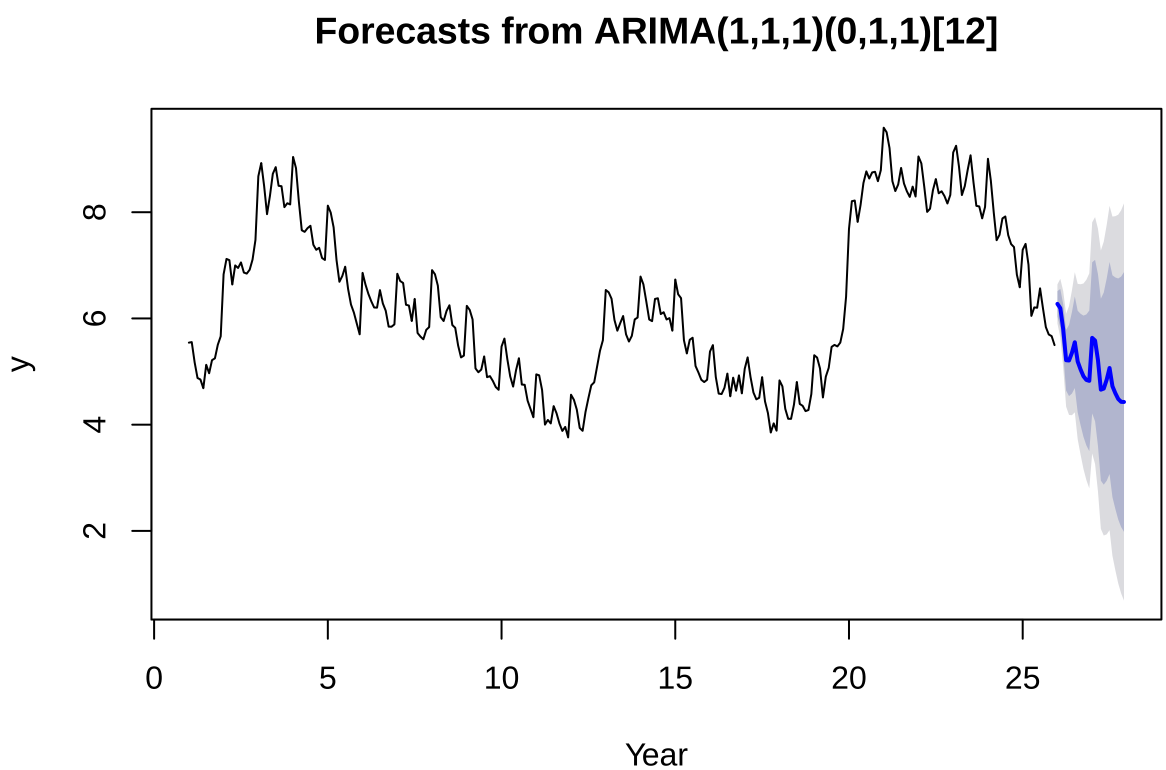
ARIMA(1,1,1)(0,1,1)[12]

This is the model with the lowest BIC and best out-of-sample performance.

```
fit <- Arima(ms.training, order = c(1, 1, 1), seasonal = c(0,
    1, 1))
fit
```

```
## Series: ms.training
## ARIMA(1,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.9311  -0.8047  -0.8909
## s.e.  0.0388   0.0560   0.0520
##
## sigma^2 estimated as 0.03519:  log likelihood=65.36
## AIC=-122.71   AICc=-122.57   BIC=-108.08
```

```
plot(forecast(fit), ylab = "y", xlab = "Year")
```




```
# PLEASE HELP TO FIX THE TICK VALUES FOR X-AXIS
```

ARIMA(2,1,1)(0,1,1)[12]

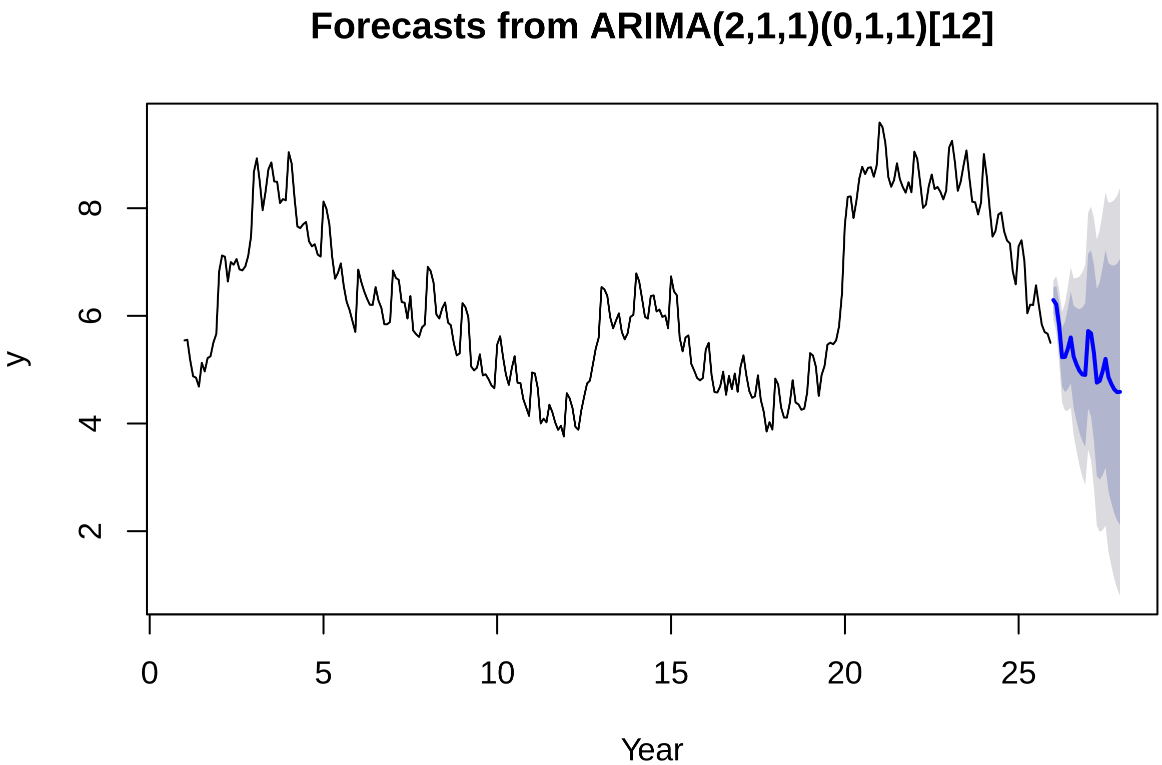
This is the model with the lowest AIC and AICc.

```
fit <- Arima(ms.training, order = c(2, 1, 1), seasonal = c(0,
    1, 1))
fit
```

```
## Series: ms.training
## ARIMA(2,1,1)(0,1,1)[12]
##
```

```
## Coefficients:
##          ar1     ar2      ma1     sma1
##       0.7289  0.1592  -0.7036  -0.8825
## s.e.  0.1030  0.0680   0.0900   0.0512
##
## sigma^2 estimated as 0.03476:  log likelihood=67.93
## AIC=-125.86   AICc=-125.64   BIC=-107.56
```

```
plot(forecast(fit), ylab = "y", xlab = "Year")
```




```
# PLEASE HELP TO FIX THE TICK VALUES FOR X-AXIS
```

VIII. Conclusions