W271 Section 3 Lab 4

Kiersten Henderson, Zhaoning Yu, Daghan Altas

12/09/2017

```
knitr::opts_chunk$set(cache=TRUE)

library(easypackages)
packages("knitr","xts","forecast","ggfortify","ggplot2", "dplyr","plotly", "Hmisc",
         "tseries","stats","fpp", "forcats")
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
rm(list=ls())
```

I. Introduction

II. Loading and cleaning up the data

We'll load the data and inspect the structure. We will also check to see if there are any missing values.

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

```
cbind(head(df), tail(df))
```

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

```
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"]
ms.test <- ms["2015/"]
rbind(head(ms.training, 3), tail(ms.training, 3))
```

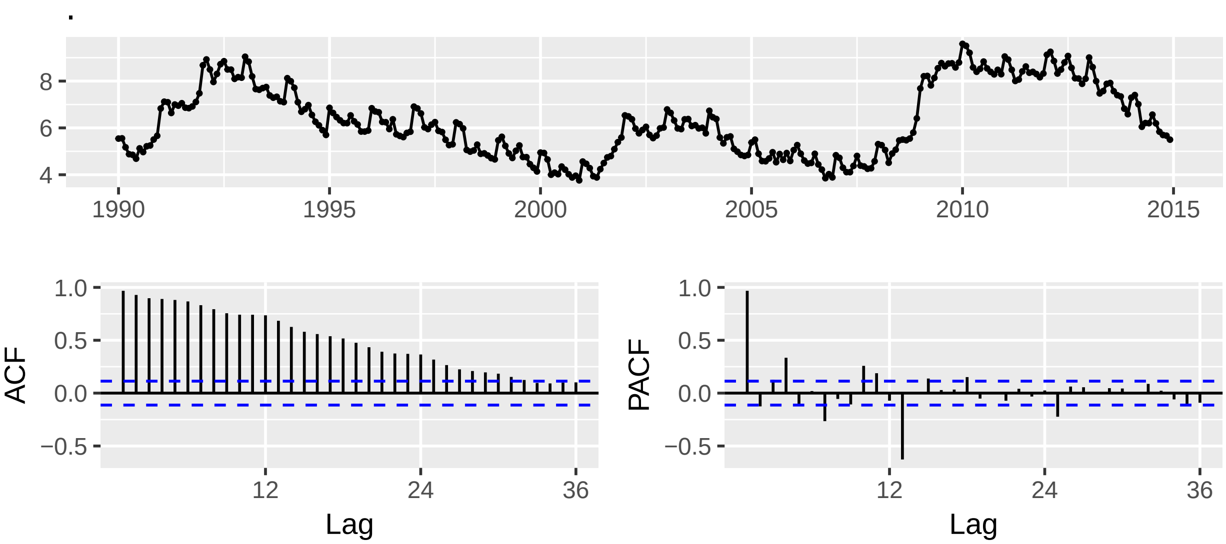
```
##           [,1]
## Jan 1990 5.544
## Feb 1990 5.555
## Mar 1990 5.172
```

```
## Oct 2014 5.698
## Nov 2014 5.668
## Dec 2014 5.498
```

III. EDA

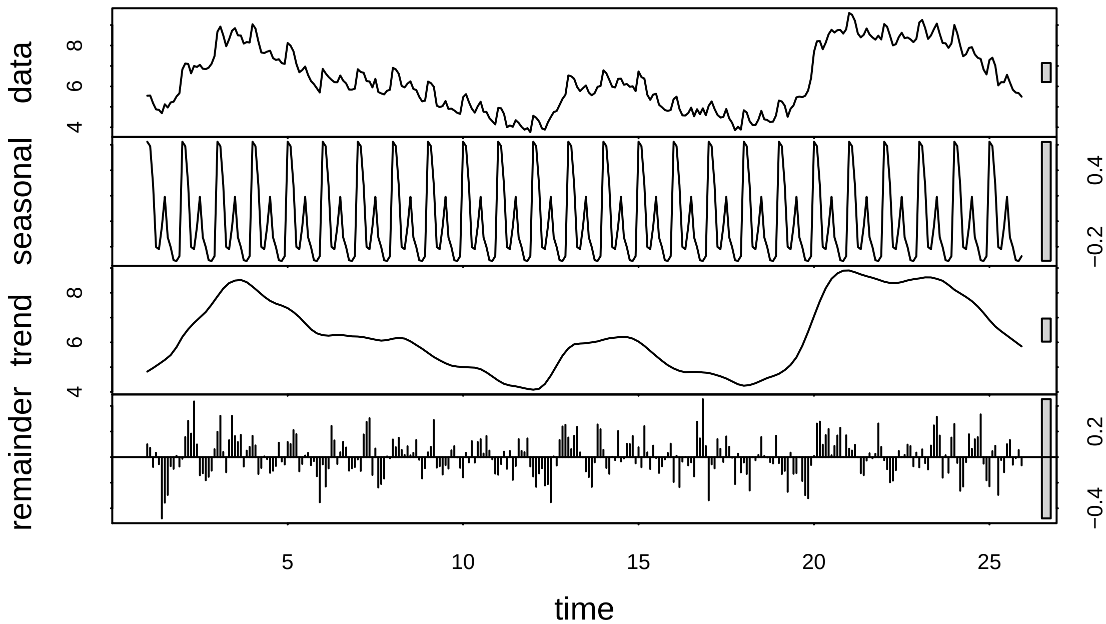
We first plot the time series together with its ACF and PACF.

```
# there is an issue with X axis when plotting xts objects,
# converting to ts for plotting
as.ts(ms.training, start = head(index(ms.training), 1), end = tail(index(ms.training),
    1)) %>% 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)
```

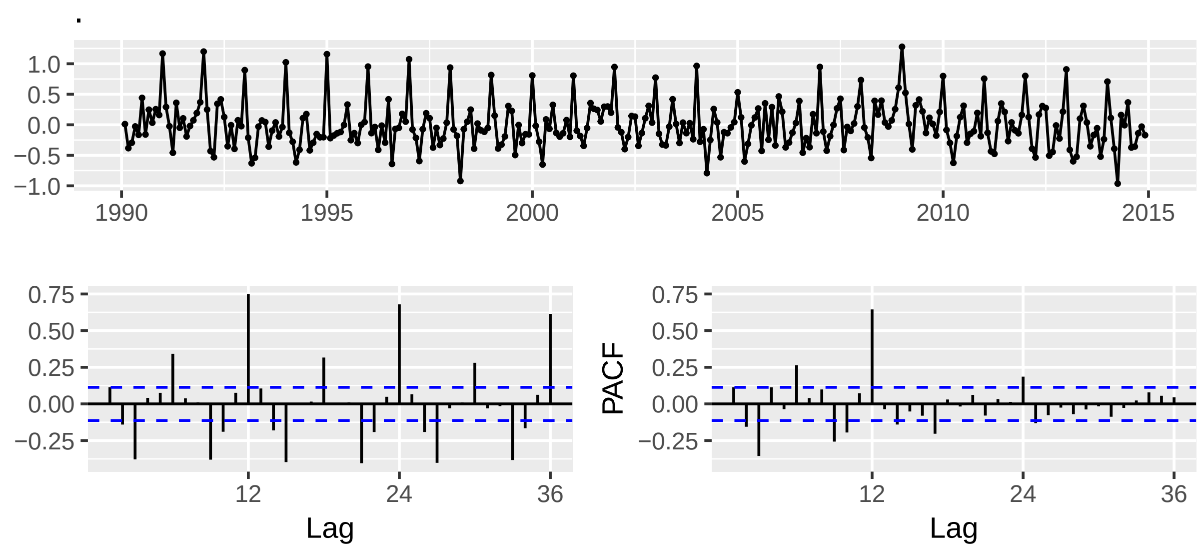


The series both show a trend and a seasonal component. It is not stationary in the mean. This indicates the need for differencing to stabilize the mean.

Transformations

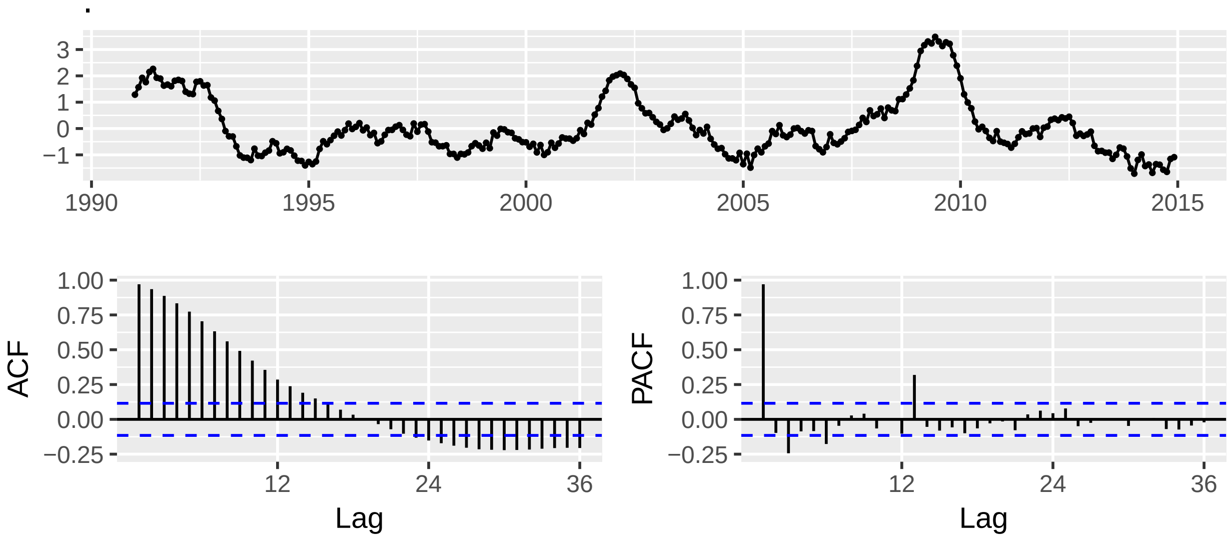
We have trend so we start with taking a first difference of the time series.

```
# First differencing only
ms.training.1d <- diff(ms.training, lag = 1)
# We'll filter out the first value (since we have a 1 lag
# differencing)
ms.training.1d <- ms.training.1d[!is.na(ms.training.1d)]
as.ts(ms.training.1d, start = head(index(ms.training.1d), 1),
    end = tail(index(ms.training.1d), 1)) %>% ggtsdisplay
```



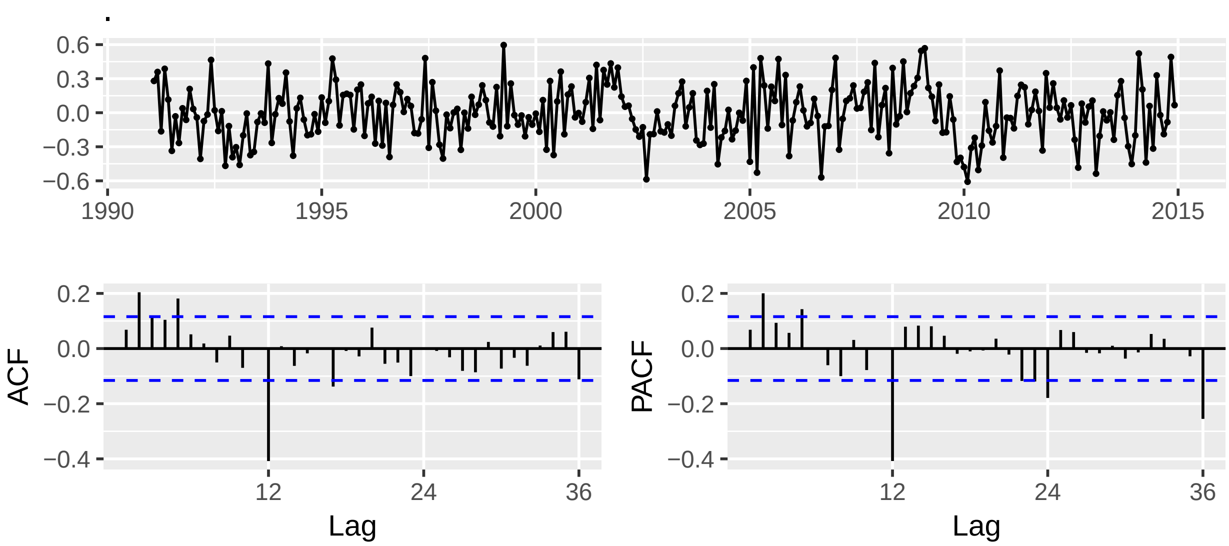
With only the first-differencing, the series appear to be somewhat statinary. But looking at the time domain as well as the PACF graph, it is clear that there is a strong yearly (at lag 12) component that needs to be addressed. Next we are going to explore the seasonal effects.

```
# Seasonal differencing only
ms.training.12d <- diff(ms.training, lag = 12)
# We'll filter out the first 12 values (since we have a
# 12-lag differencing)
ms.training.12d <- ms.training.12d[!is.na(ms.training.12d)]
as.ts(ms.training.12d, start = head(index(ms.training.12d), 1),
    end = tail(index(ms.training.12d), 1)) %>% ggtsdisplay
```



We observe that the seasonal differencing has significantly smoothed the time domain graph and we further observe that effect on the ACF / PACF graphs. However, the trend is obvious and the series are not stationary. We will combine the seasonal and non-seasonal components for our next exploratory graph

```
# Both the first and the seasonal differencing
ms.training.1d.12d <- diff(diff(ms.training, lag = 1), lag = 12)
# We'll filter out the first 12 values (since we have a
# 12-lag differencing)
ms.training.1d.12d <- ms.training.1d.12d[!is.na(ms.training.1d.12d)]
as.ts(ms.training.1d.12d, start = head(index(ms.training.1d.12d),
    1), end = tail(index(ms.training.1d.12d), 1)) %>% ggtsdisplay
```



We note that our first-difference / lag-12 seasonal differenced model appear much more stationary and allow us to start conducting ACF / PACF analysis to find the *auto-regressive* and *moving-average* components. We will further strengthen our argument with an augmented Dickey Fuller test between the 2 potential series (first-difference vs. first-difference/seasonal-difference).

```
adf.test(ms.training.1d.12d)
```

```
## Warning in adf.test(ms.training.1d.12d): p-value smaller than printed p-
## value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ms.training.1d.12d
## Dickey-Fuller = -4.8865, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

We observe that there is emprical evidence to consider our first-difference / 12-lag seasonal difference model to be stationary. In addition, we see a PACF strong component at lag 12, which suggests a seasonal MA(1) component. There are statistically significant ACF/PACF componnents at lag 2 and 5 without a clear pattern pointing in either MA or AR direction. ACF graph suggests an MA(2) model, whereas PACF graph suggests and AR(2) model. Both graphs hint at a lag(5) component, in addition to the seasonal MA(1) component.

EDA Summary

- Our analysis points to an ARIMA$(p,1,q)(P,1,Q)_{12}$ model
- Our non-seasonal AR/MA search for p/q should go up to lag(5)
- Our expectation is to find an appropriate model with $p \in (1,2)$ and $q \in (1,2)$

IV. 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

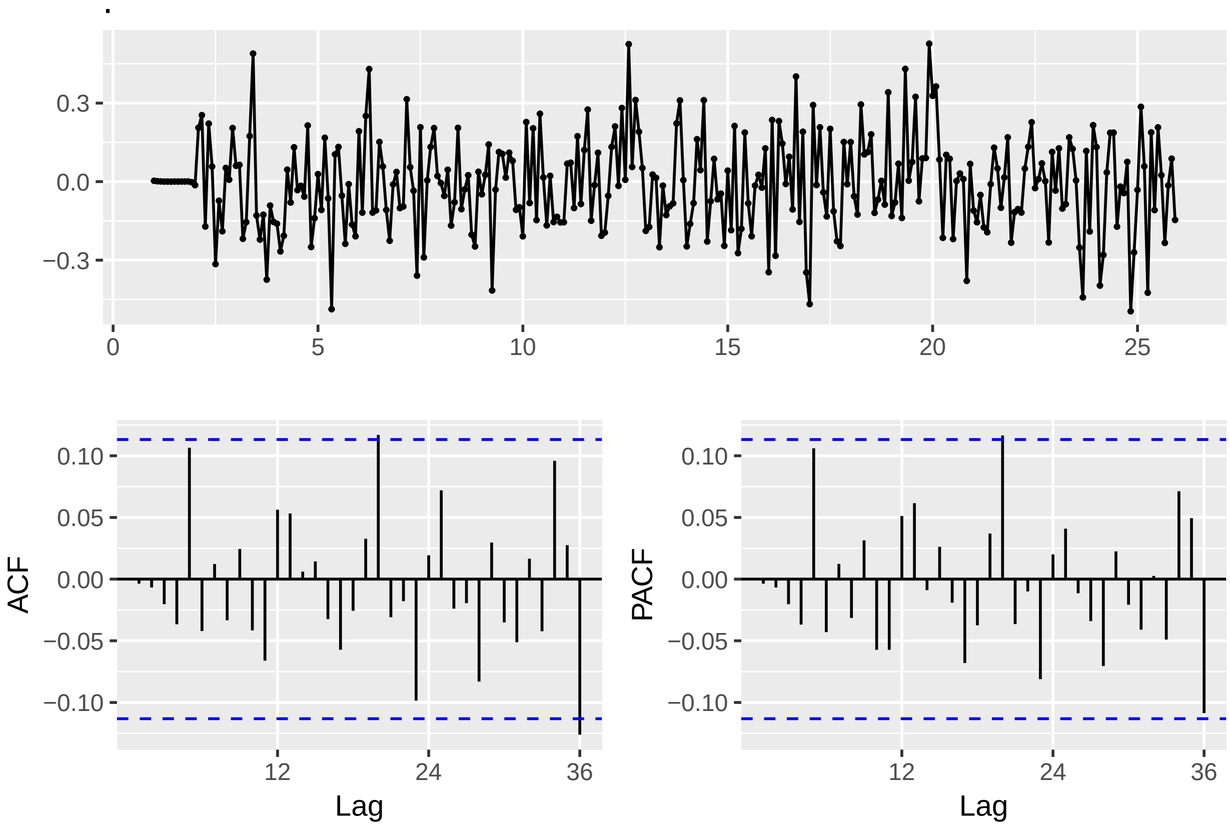
Since the procedure is repetitive, we define a function for model testing:

```
# 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)
```

```
    # 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

```
# 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$bic)
    }
}
rbind(results[which.min(results$AIC), ], results[which.min(results$AICc),
    ], results[which.min(results$BIC), ])
```

```
##    p q       AIC      AICc       BIC
## 6  2 1 -125.8566 -125.6430 -107.5591
## 61 2 1 -125.8566 -125.6430 -107.5591
## 1  1 1 -122.7133 -122.5715 -108.0754
```

Candidate models

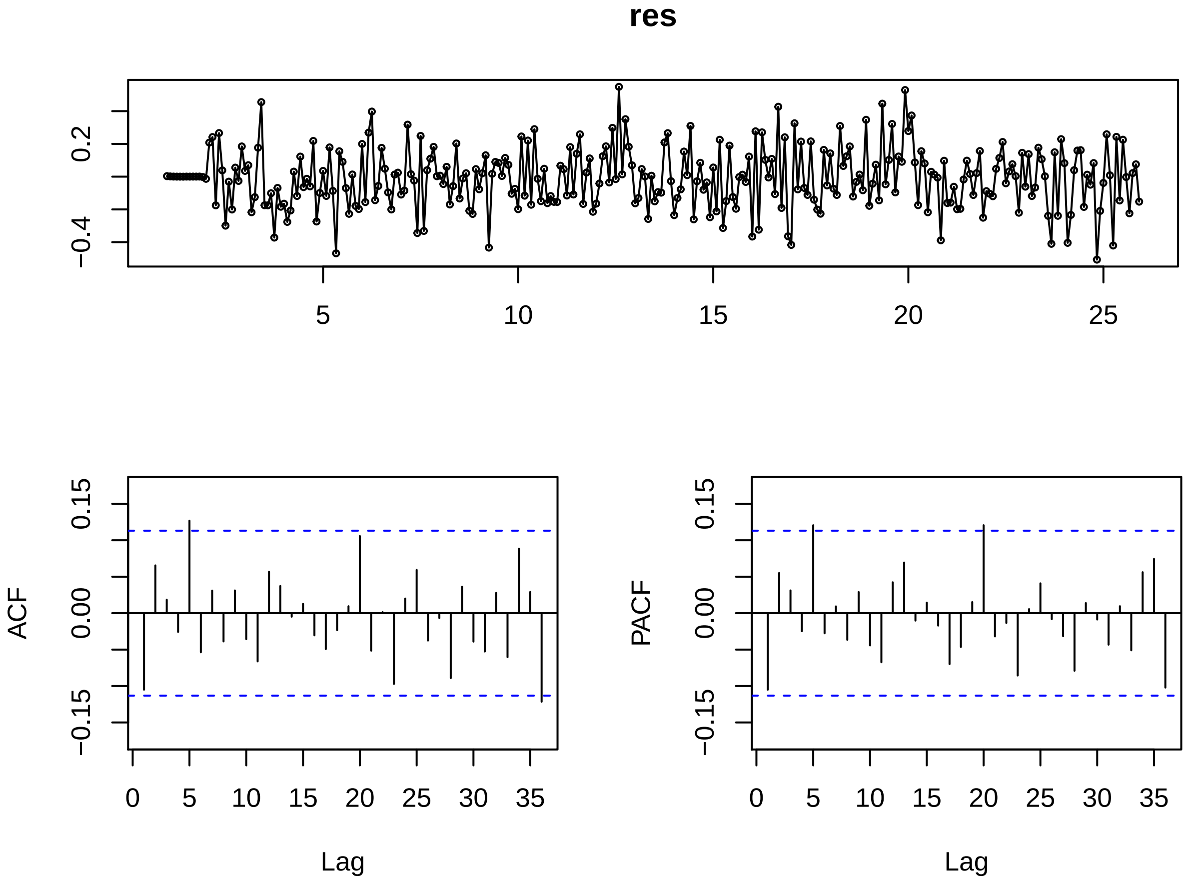
The grid search corraborates our exploratory analysis. We have looked for an optimal p,q combination within the $p \in 1,5$ and $q \in 1,5$. Based on the information criteria optimization, we are going to focus on the following models:

- ARIMA(1,1,1)(0,1,1)[12] (minimizes BIC)
- 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

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



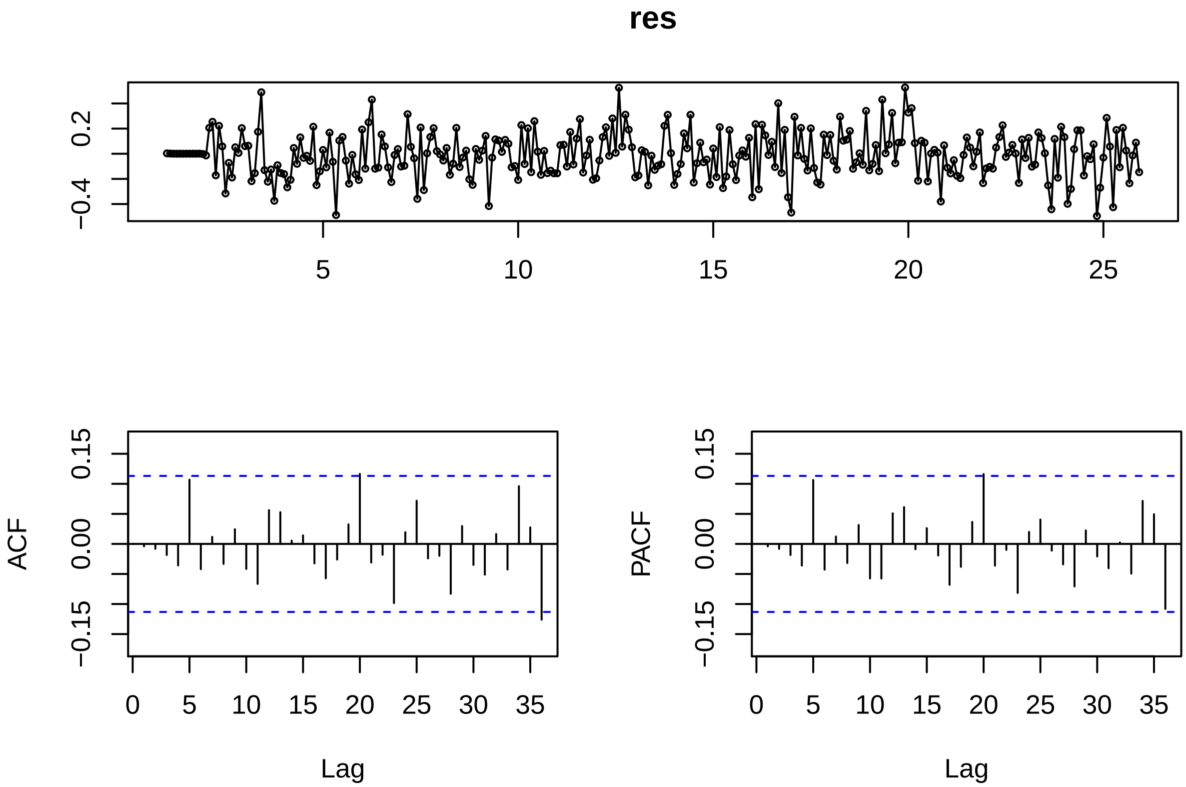

```
Box.test(res, lag = 16, fitdf = 4, type = "Ljung")  # p-value = 0.2188
```

```
##
##  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)
```

```r
tsdisplay(res)
```




```r
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
```

```r
# 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.

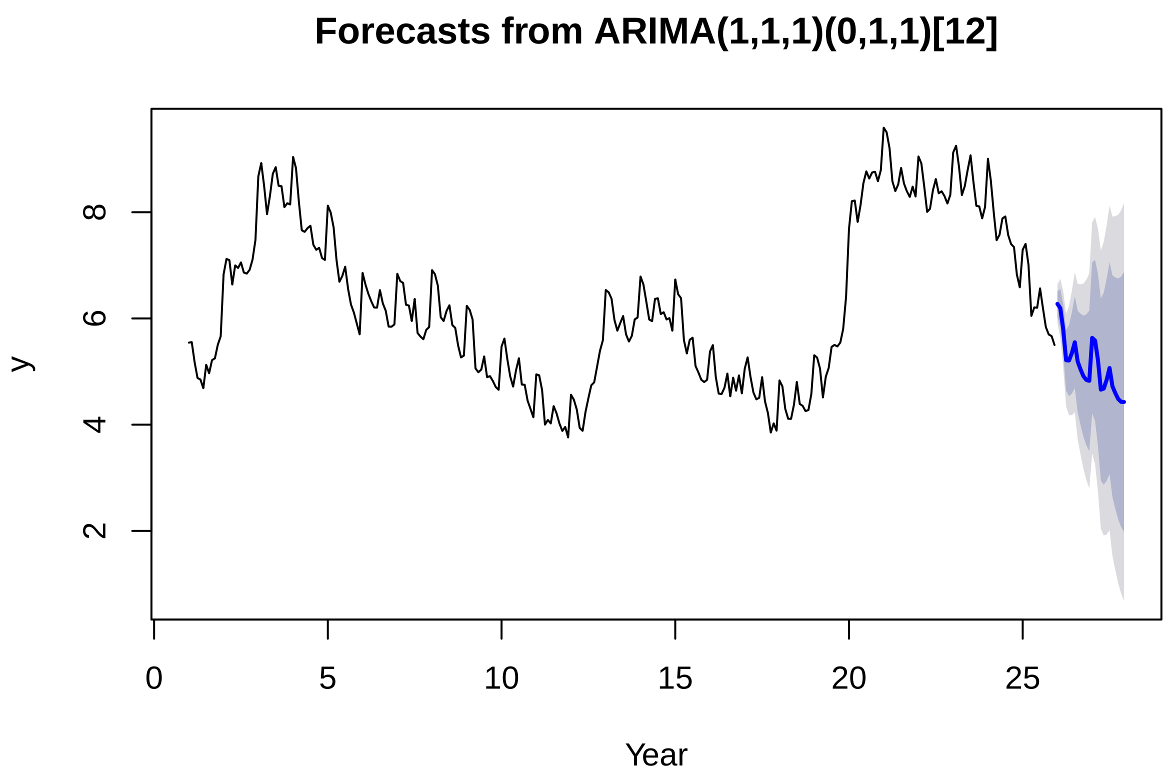
ARIMA(1,1,1)(0,1,1)[12]

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

```r
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
```

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




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

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

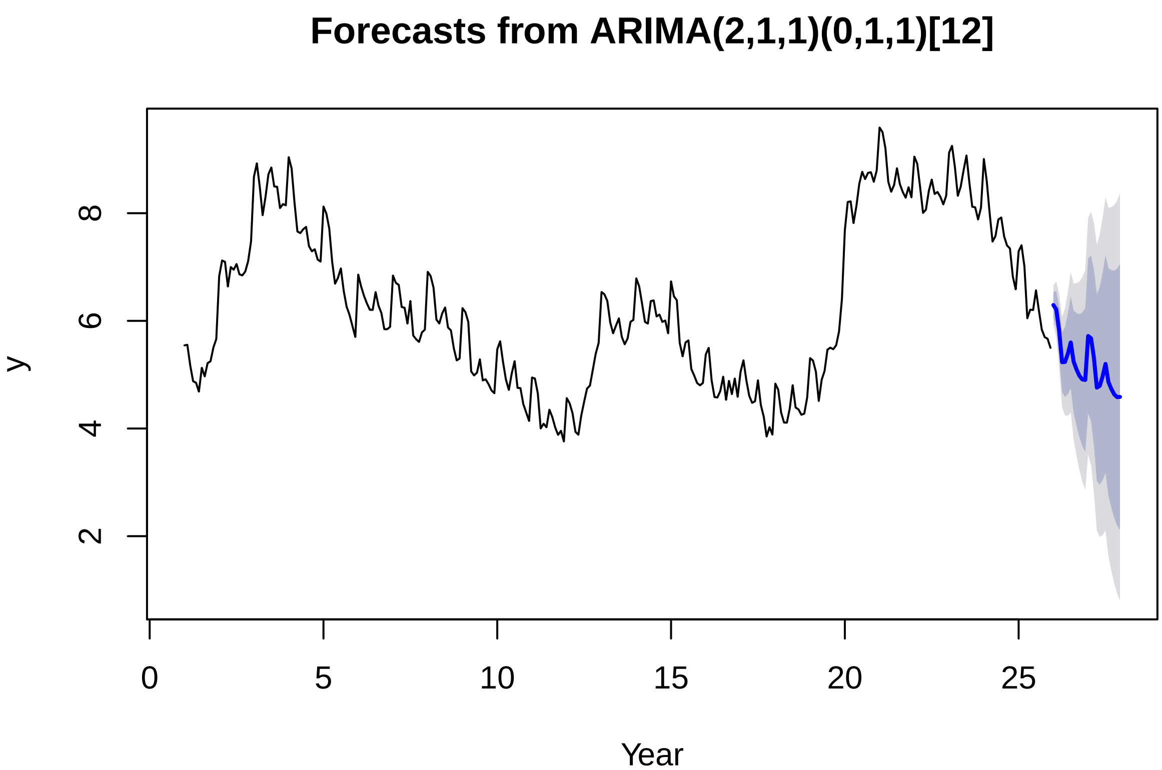
This is the model with the lowest AIC and AICc.

```r
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