

# 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

Our task was to analyze a time series entitled “Lab4-series2”. Because the time series is “unidentified” we did not know the domain of the data-generating process. Domain-specific knowledge often guides modeling choices.

Nonetheless, we were able to estimate a valid model (based on model diagnostics) with high forecast accuracy (measured by mean absolute percent error (MAPE) using out-of-sample fit) when compared to a naive forecast.

Of the many models we estimated, we favor an ARIMA(1,1,1)(0,1,1)<sub>12</sub>.

*latex*

## II. Data Loading and Cleaning

We began our analysis by loading the data, inspecting its structure, and checking for 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 values and the first column of the dataframe is the index column, which we chose to discard. We proceeded by converting the data to a (xts) based time series due to the ease of subsetting xts-based objects.

```
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
```

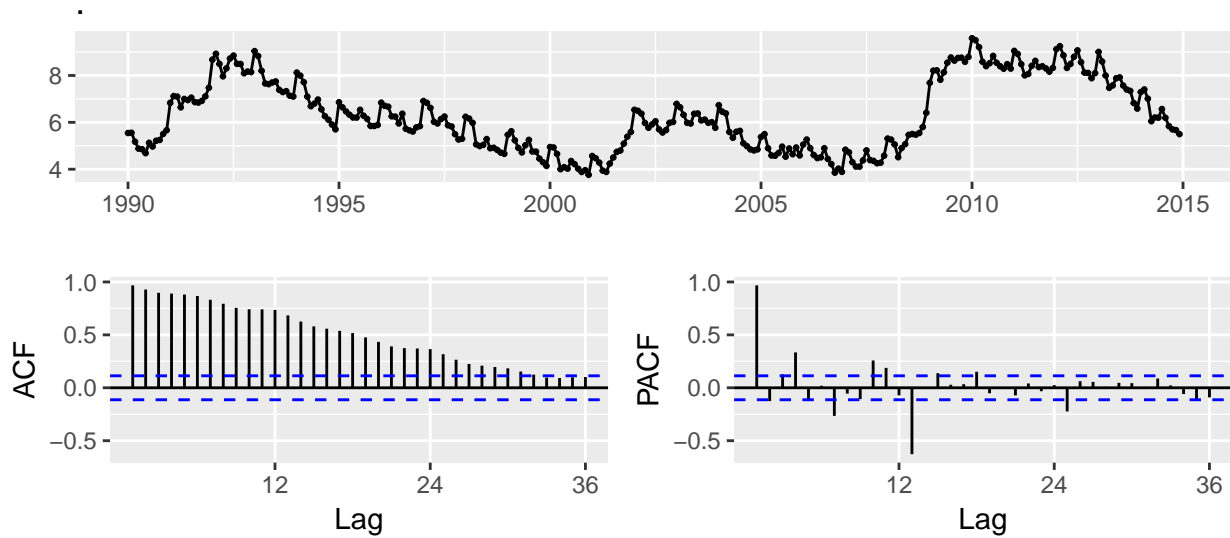
```
ms.test
```

```
##           [,1]
## Jan 2015 6.449
## Feb 2015 6.425
## Mar 2015 5.929
## Apr 2015 5.536
## May 2015 5.472
## Jun 2015 5.240
## Jul 2015 5.546
## Aug 2015 5.078
## Sep 2015 4.907
## Oct 2015 4.599
## Nov 2015 4.681
```

### III. Exploratory Time Series Data Analysis

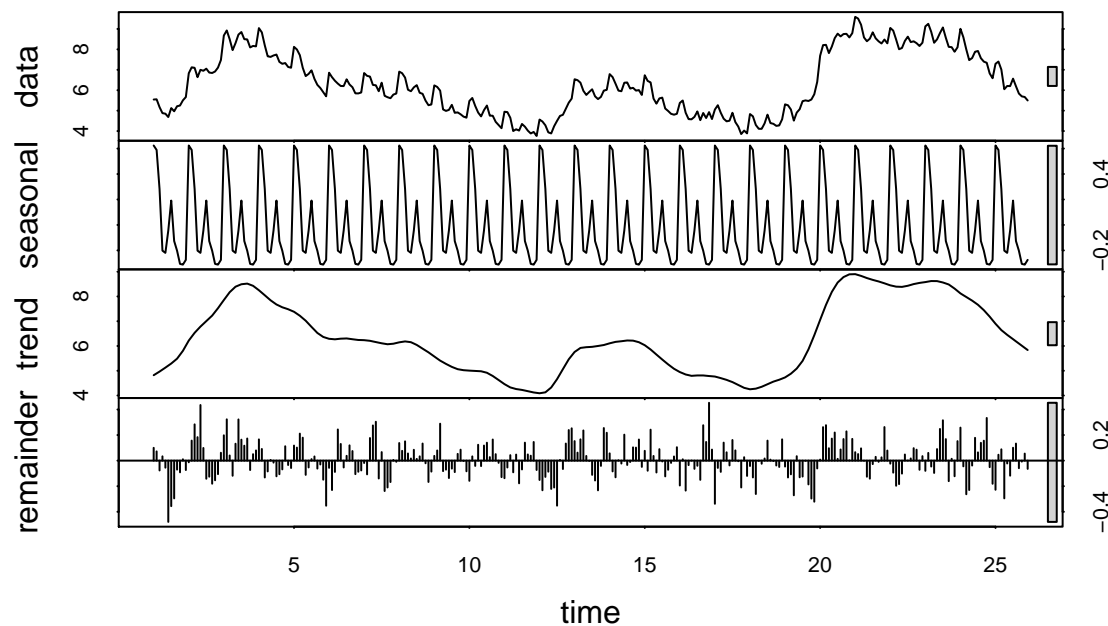
We initially plotted the untransformed time series with its ACF and PACF.

```
# 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 made use of STL decomposition 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)
```



We observed that the untransformed series has both a trend and a seasonal component. Because the series is not stationary in the mean this immediately suggested the need for differencing.

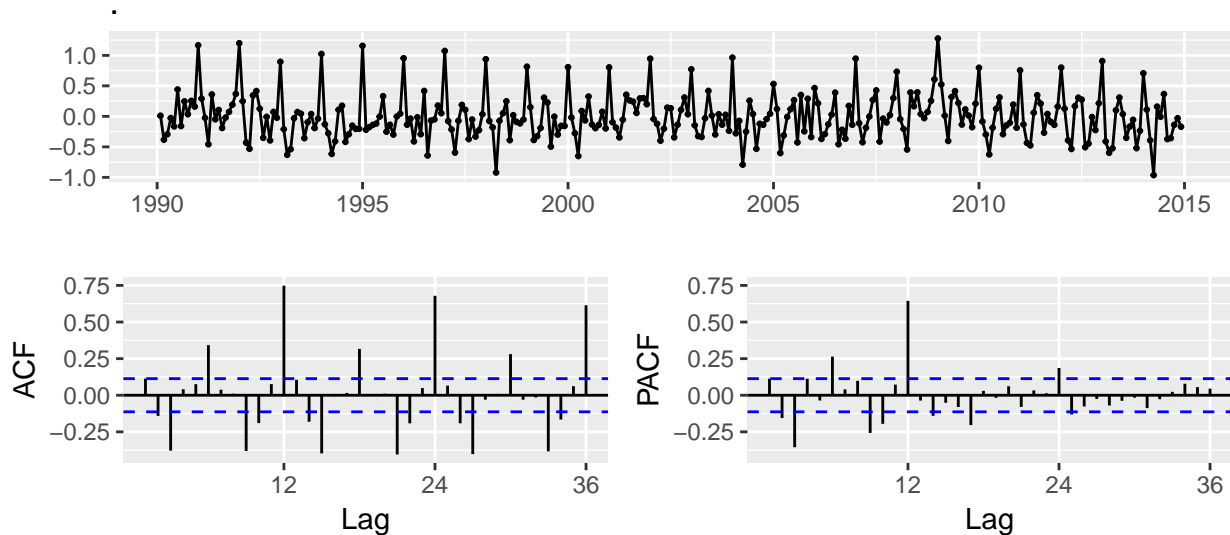
## Transformations

### Differencing for Trend

We identified a trend component in the untransformed time series, so we performed a first difference of the time series.

```
# First differencing only
ms.training.1d <- diff(ms.training, lag = 1)
```

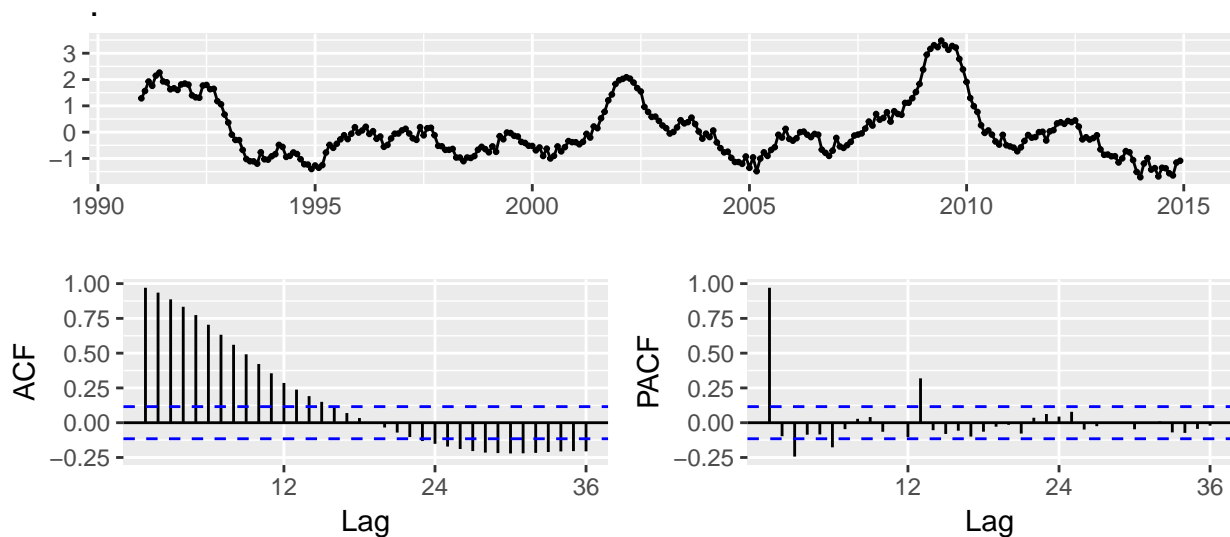
```
# We'll filter out the first value (since we have a 1 lag differencing)
ms.training.1d <- ms.training.1d[!is.na(ms.training.1d)]
as.ts(ms.training.1d, start = head(index(ms.training.1d),1), end = tail(index(ms.training.1d),1)) %>% g
```



With only the first-differencing, we observed that the series appears somewhat stationary. However, when we examined the time series plot as well as the PACF chart, we observed a strong yearly seasonal component (at lag 12). This yearly seasonal component warrants attention and so we proceeded to explore the seasonality of our time series.

\*\* add adf test and interpretation

```
# 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)) %>% g
```



## Differencing for Yearly Seasonality

We observed that the seasonal differencing significantly smoothed the time series. Furthermore, the effect of seasonal differencing is also apparent in the ACF / PACF graphs.

*\*\*in what way?*

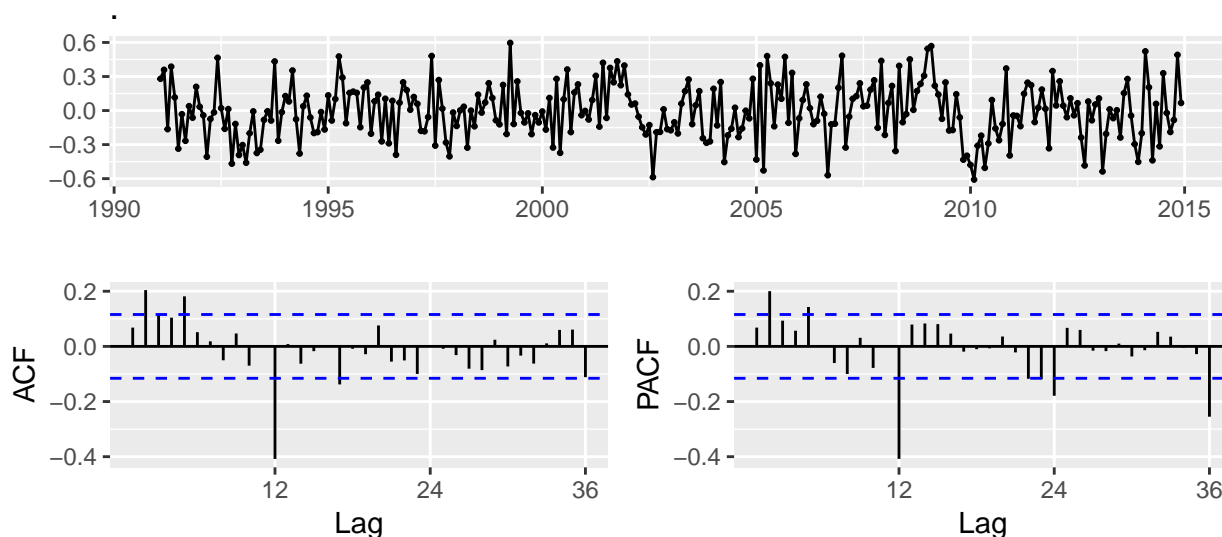
However, there is still obvious trend in the series and it is therefore, not stationary.

*\*\*adf and discussion*

We will combine the seasonal and non-seasonal components for our next exploratory graph

## Differencing for Trend and Yearly Seasonality

```
# Both the trend and seasonal differencing
ms.training.1d.12d <- diff(diff(ms.training, lag = 1), lag = 12)
# We filtered 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))
```



We found that the combination of first-difference for trend plus differencing for lag-12 seasonality produces a time series that appears much more stationary than did differencing for either component alone. The augmented Dickey-Fuller test supported our intuition. Using the test, we were able to reject the null hypothesis that a unit root existed for the series ( $p = 0.01$ ) and thus proceed with the alternate hypothesis that the differenced series is stationary.

```
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 were thus able to examine and interpret the ACF and PACF plots to identify potential *auto-regressive* and *moving-average* components for model estimation.

There are statistically significant ACF/PACF components 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 an AR(2) model. Both graphs hint at a lag(5) component, in addition to the seasonal MA(1) component.

When examining the PACF, we observe a strong component at lag 12, \*\*expand discussion to ACF which suggests a seasonal MA(1) component.

## EDA Summary

- summarize rationale for differencing decision
- Our exploratory analysis of the time series suggests an  $ARIMA(p, 1, q)(P, 1, Q)_{12}$  model
- Based on our interpretation of the ACF and PACF plots, our exploration to identify non-seasonal AR/MA order (p and q) considered order up to 5.
- However, to avoid overfitting, the goal of our model estimation process was to identify appropriate models where the order of  $p \in (1, 2)$  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 defined 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

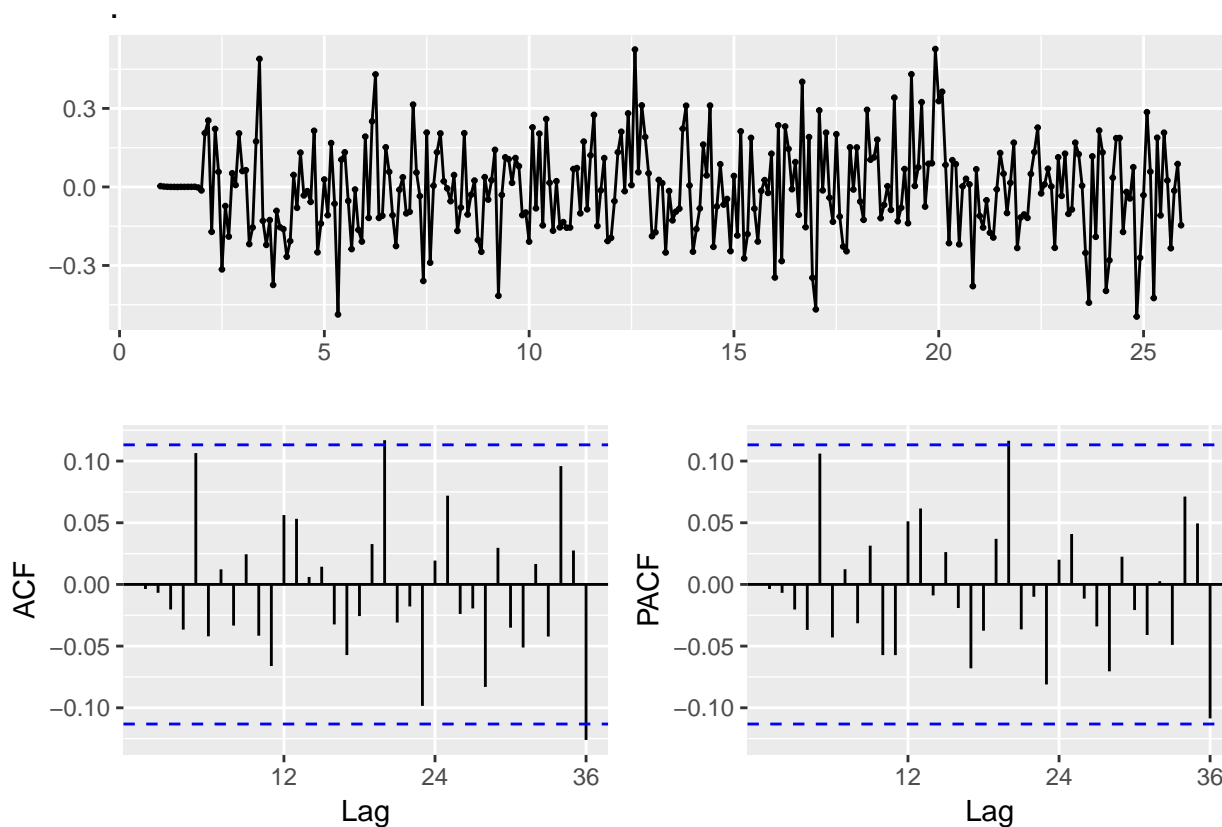
  # find MAPE
  f1 <- ms.training %>% Arima(order = ORDER, seasonal=list(order=SEASONAL,period=12)) %>% forecast(h = 1)

  # 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)[1,6])
  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 corroborates 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:

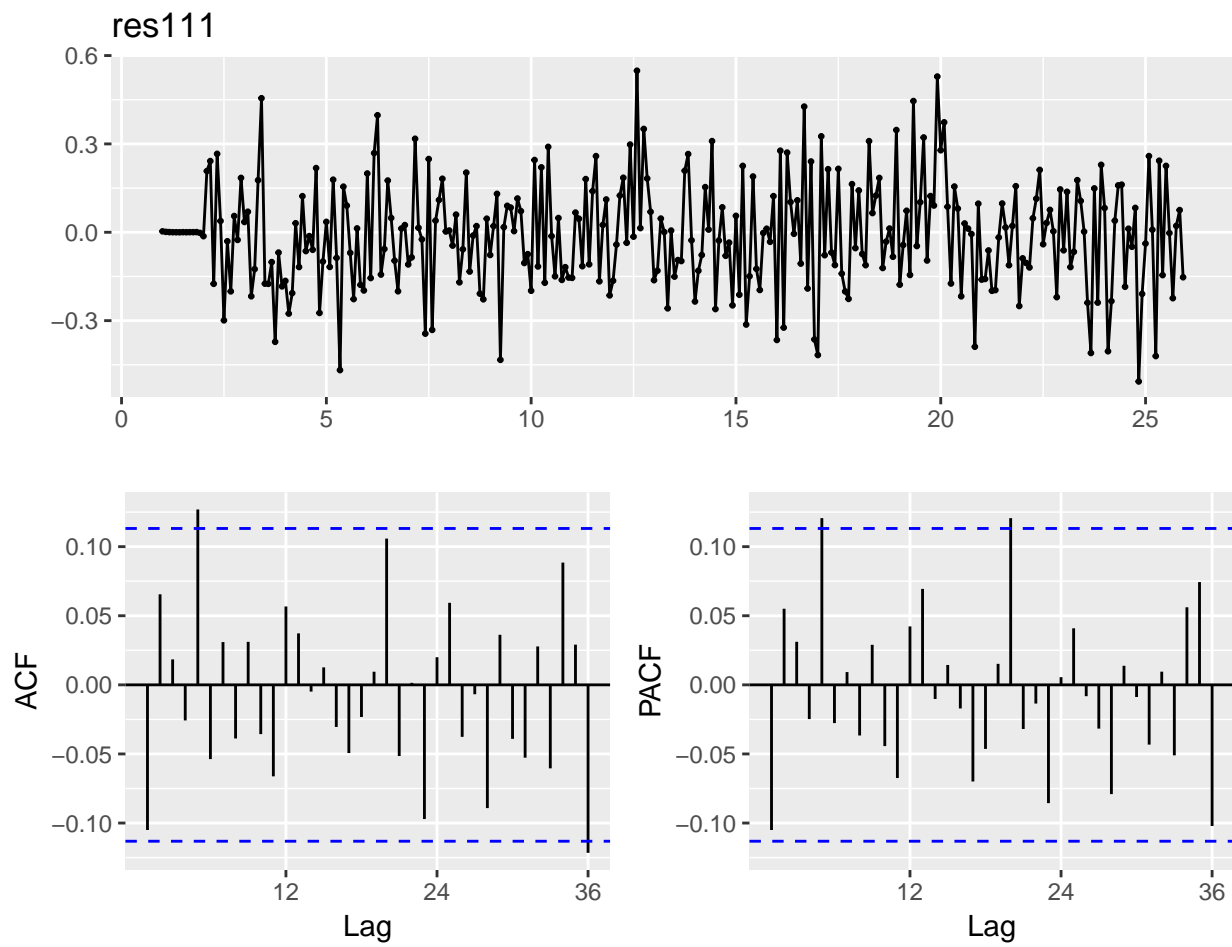
- ARIMA(1,1,1)(0,1,1)[12] (minimizes BIC)
- ARIMA(2,1,1)(0,1,1)[12] (minimizes AIC / AICc)

## VI. Model validation

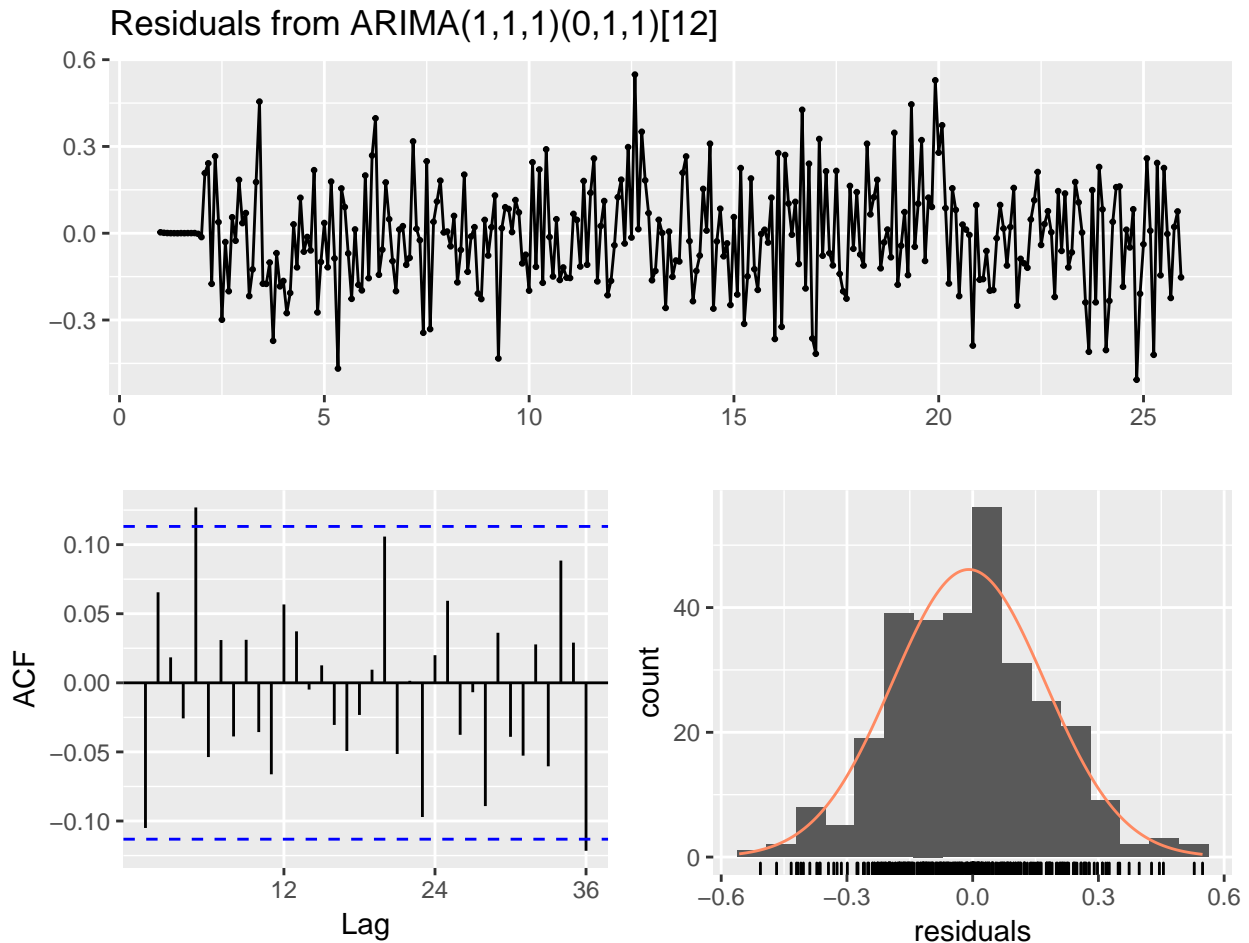
### ARIMA(1,1,1)(0,1,1)[12]: lowest BIC

```
fit111 <- ms.training %>% Arima(order = c(1, 1, 1), seasonal=list(order=c(0,1,1),period=12))
res111 <- residuals(fit111)
ggtsdisplay(res111)
```





```
checkresiduals(fit111)
```



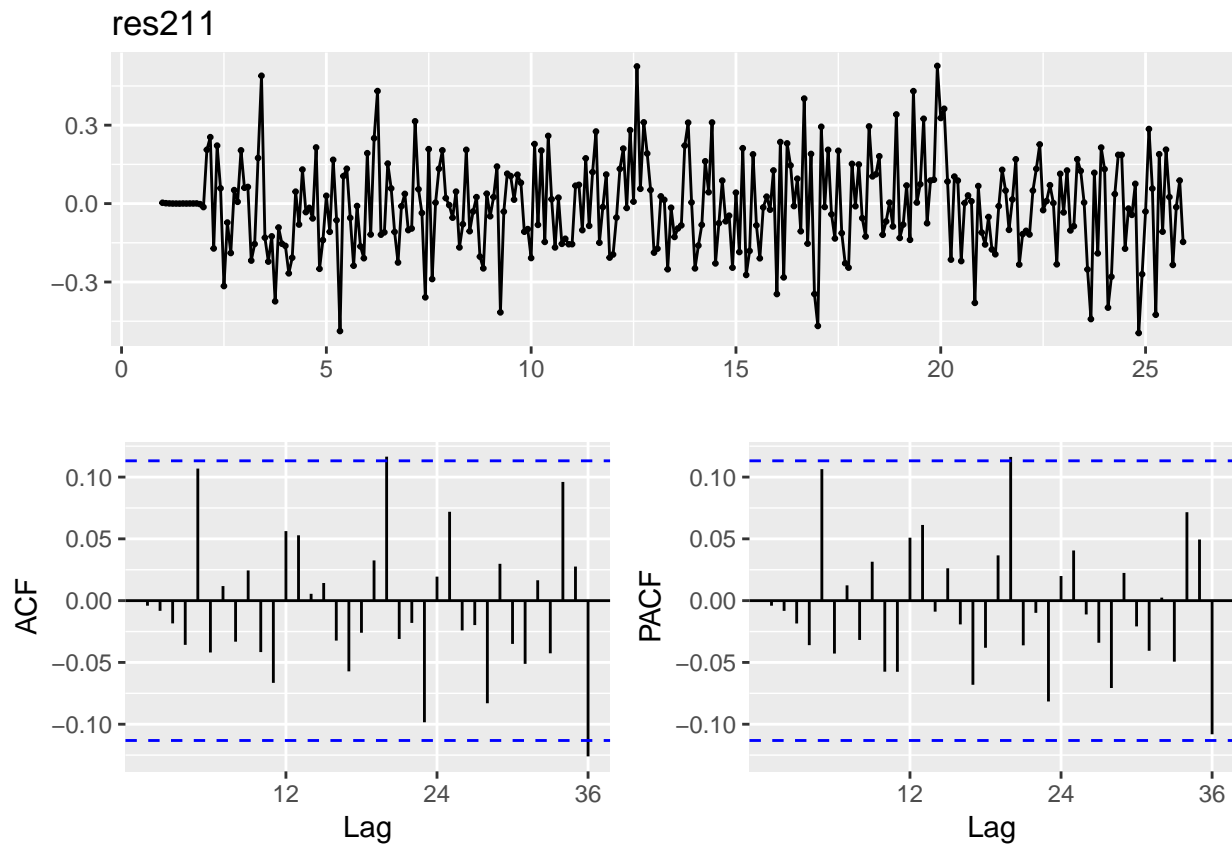
```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)(0,1,1)[12]
## Q* = 24.11, df = 21, p-value = 0.2878
##
## Model df: 3.   Total lags used: 24
Box.test(res111, lag=16, fitdf=4, type="Ljung") # p-value = 0.2188
```

```
##
##  Box-Ljung test
##
## data:  res111
## 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

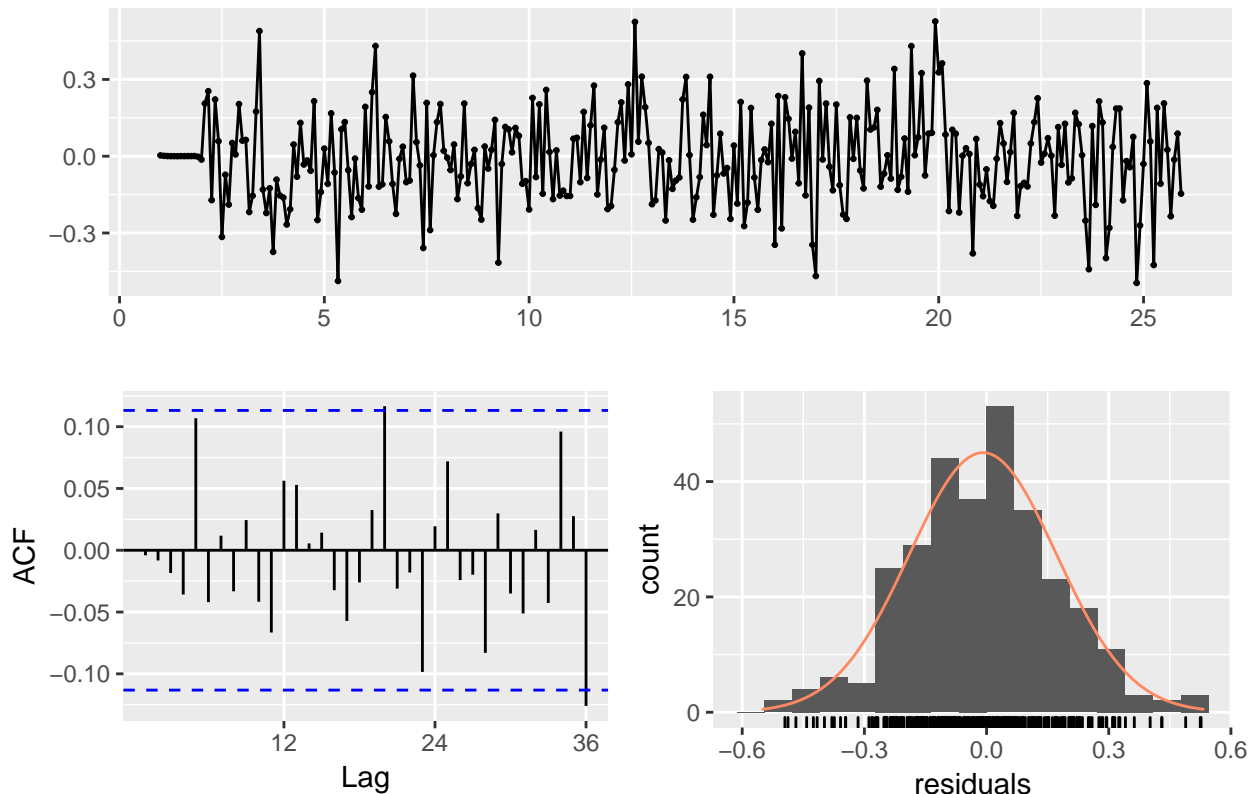
```
fit211 <- ms.training %>% Arima(order = c(2, 1, 1), seasonal=list(order=c(0,1,1),period=12))
res211 <- residuals(fit211)
```

```
ggtsdisplay(res211)
```



```
checkresiduals(fit211)
```

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



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,1)(0,1,1)[12]
## Q* = 19.069, df = 20, p-value = 0.5174
##
## Model df: 4.    Total lags used: 24
Box.test(res211, lag=16, fitdf=4, type="Ljung") # p-value = 0.6722

##
##  Box-Ljung test
##
## data:  res211
## 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. Forecasting

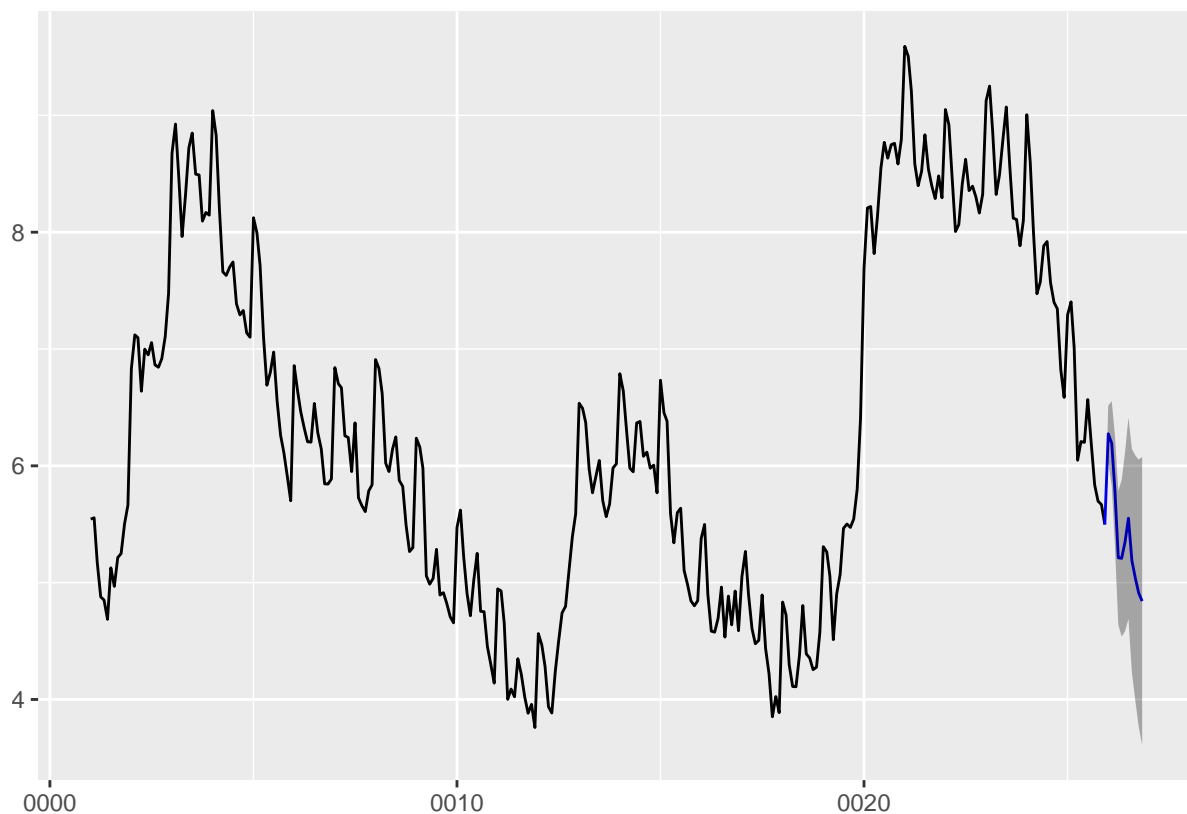
We performed an 11-month ahead forecast of the series in 2015 using both models.

```
forecast111 <- fit111 %>% forecast(h=11)
forecast211 <- fit211 %>% forecast(h=11)
(results <- rbind( accuracy(forecast111,ms.test),
                        accuracy(forecast211,ms.test))[,1:5])
```

	ME	RMSE	MAE	MPE	MAPE
## Training set	-0.009339336	0.1825311	0.1427566	-0.1059365	2.389875
## Test set	0.026623361	0.2002727	0.1779535	0.1710571	3.319050
## Training set	-0.008926836	0.1810804	0.1416233	-0.1017597	2.364688
## Test set	-0.012986097	0.2163455	0.1986888	-0.6007220	3.770132

We note that the  $ARIMA(1,1,1)(0,1,1)_{12}$  has the lowest MAPE score for the test set (2015). Therefore we performed forecasting with that model.

```
forecast111 %>% autoplot()
```



\*Describe the forecasted graph

## VIII. Conclusions

Based on our EDA we proceeded to...

We identified several appropriate models. Our criteria for choosing our favored model was... -good model diagnostics - the model residuals are statistically indistinguishable from white noise and are normally distributed -good in sample fit as measured by low values of AIC, AICc, BIC -low out of sample forecast as measured by out of sample MAPE forecast

Our favored model is: `ARIMA()`

Interpret the model in words: The value for  $y$  at any given time is dependent on a combination of the error from the previous time point (one lag previous), the value of the previous time point, and the error in the measurement from one year ago.

Describe what the forecast says. . . .

The

Why we are happy with the model : the lowest out of sample accuracy that we identified for all the valid models we estimated. Only 3.2% error when compared to actual values realized in the time series. Compare this to xx% for the naive seasonal forecast, seems like an excellent forecast.