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)_12.

$$x_t = \phi x_{t-1} + \theta w_{t-1} + \Theta w_{t-12} + w_t$$

and

$$\hat{x_t} = 0.93_{(0.038)}\hat{x}_{t-1} - 0.80_{(0.056)}\hat{w}_{t-1} - 0.89_{(0.052)}\hat{w}_{t-12} + \hat{w}_t$$

expand this below Interpret the model in words: The value for x 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.

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 seres due to the ease of subseting 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
```

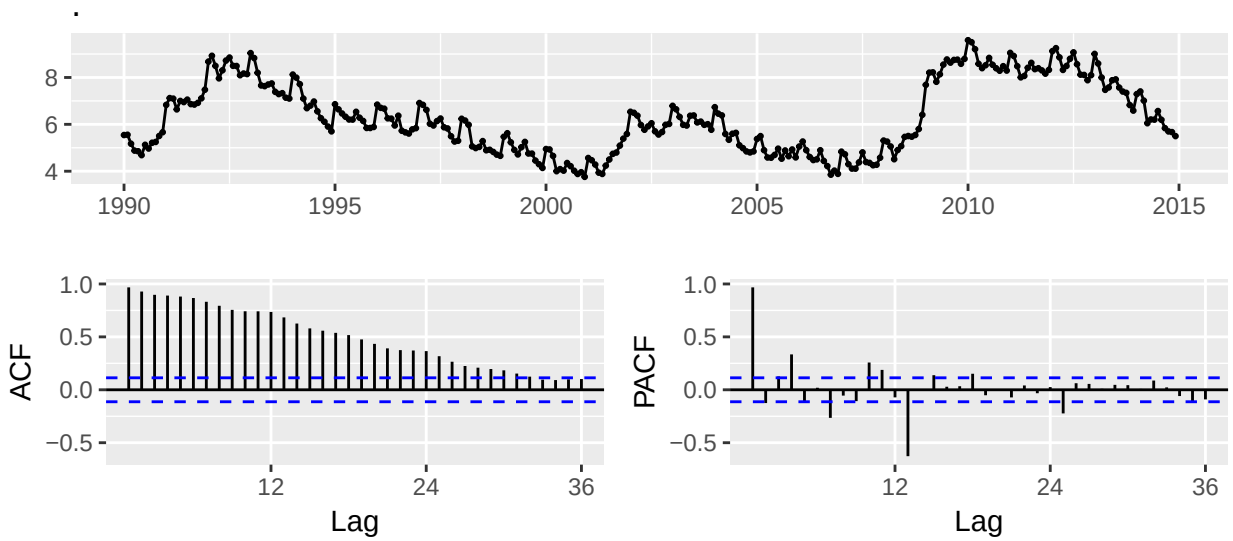
```
ms.test
```

```
##           [,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

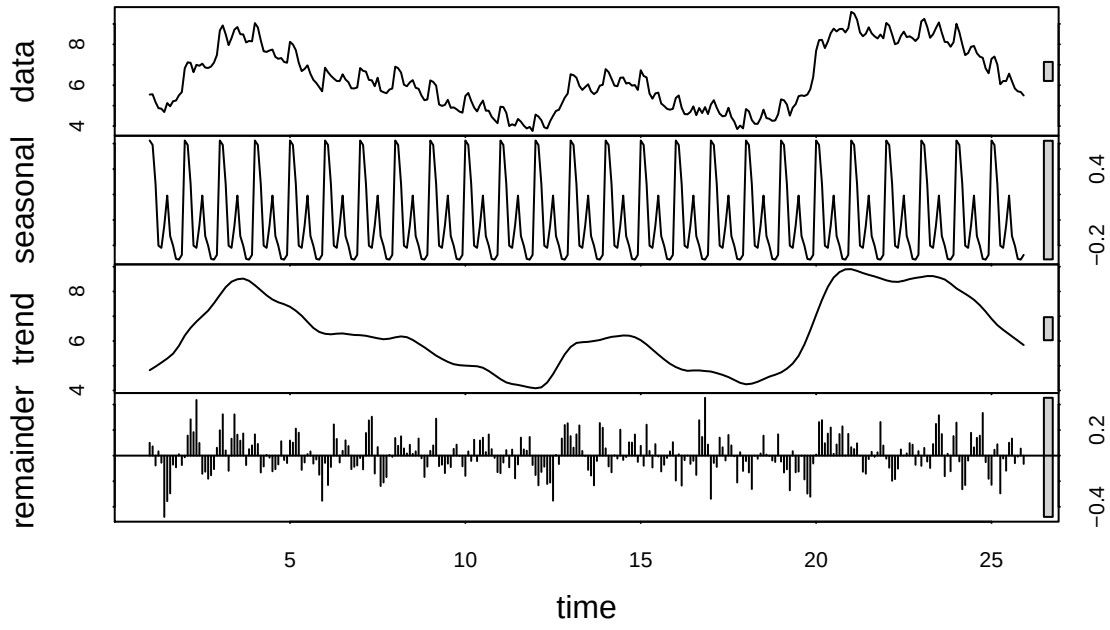
We initially plotted the untransformed time series with its ACF and PACF.

```
# 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.

The time series appears to have time-constant variance and our intuition is supported by the White test.

```
#white test
```

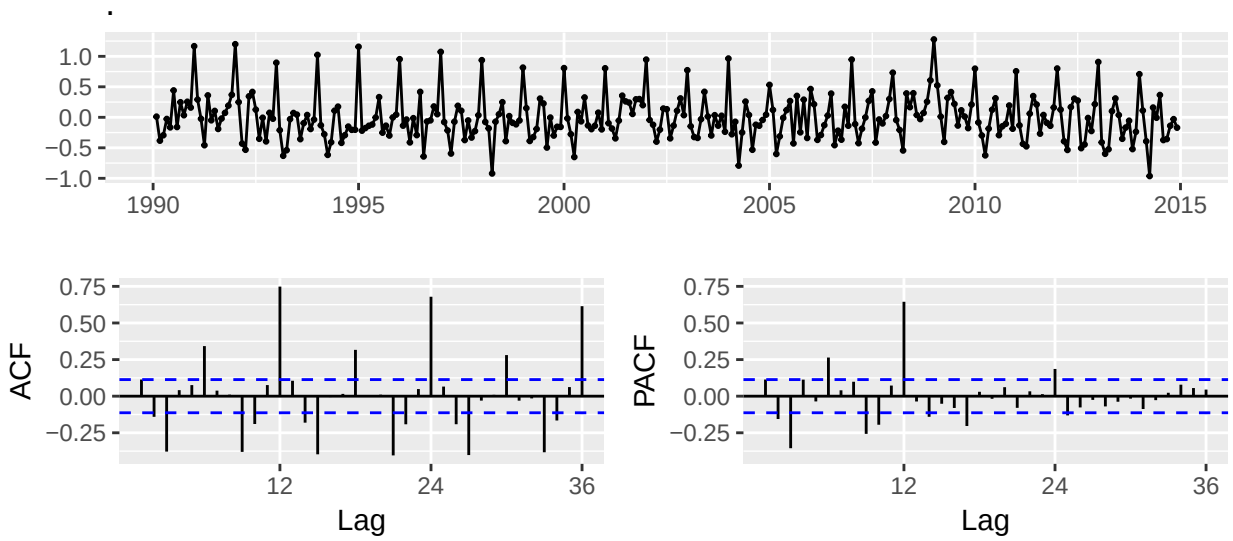
Thus, we do not see any need to stabilize the variance by transforming the series.

Transformations

Differencing for Trend

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

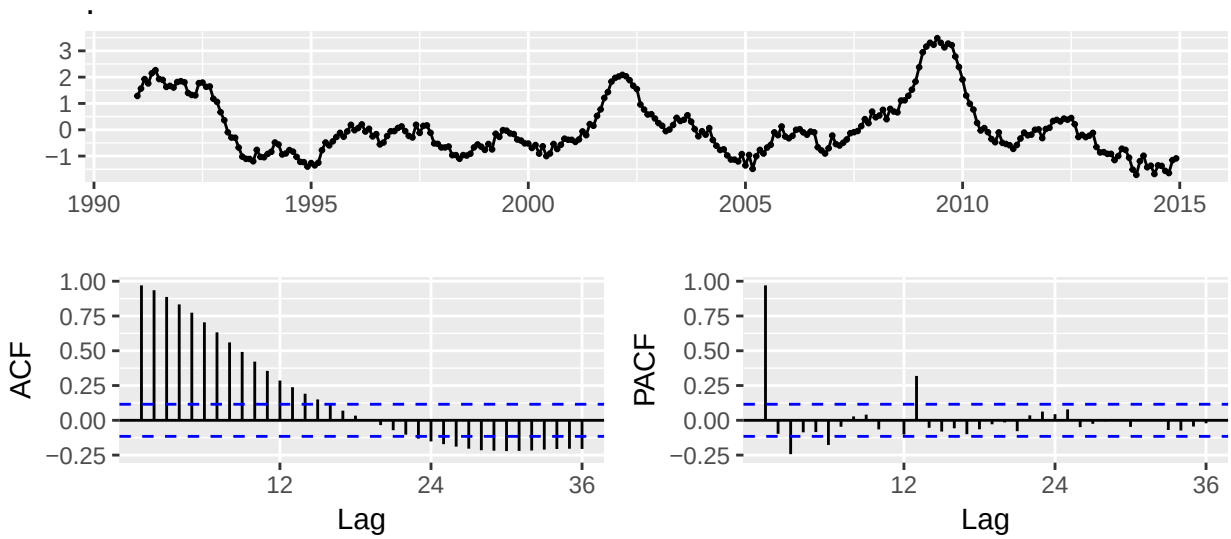
```
# First differencing only
ms.training.1d <- diff(ms.training, lag = 1)
# We'll filter out the first value (since we have a 1 lag differencing)
ms.training.1d <- ms.training.1d[!is.na(ms.training.1d)]
as.ts(ms.training.1d, start = head(index(ms.training.1d),1), end = tail(index(ms.training.1d),1))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



With only the first-differencing, we observed that the series appears somewhat statinary. 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)) %>%
```



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 and PACF graphs.

**in what way?

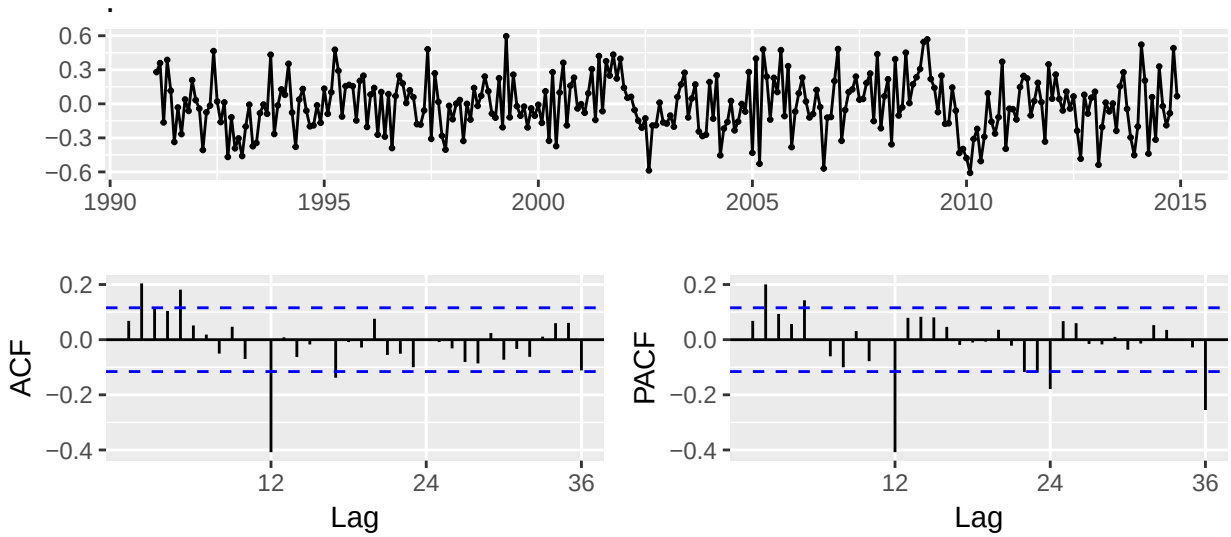
However, there is still obvious trend in the series and it is therefore, not stationary.

**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.12
```



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 weakly 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 and 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 disciussion to ACF

which suggests a seasonal MA(1) component.

Summary of Exploratory Time Series Data Analysis

- 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 approapriate models where the order of $p \in (1, 2)$ and $q \in (1, 2)$.

IV. Identifying the Dependence Orders for Estimated Models

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 decided 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.

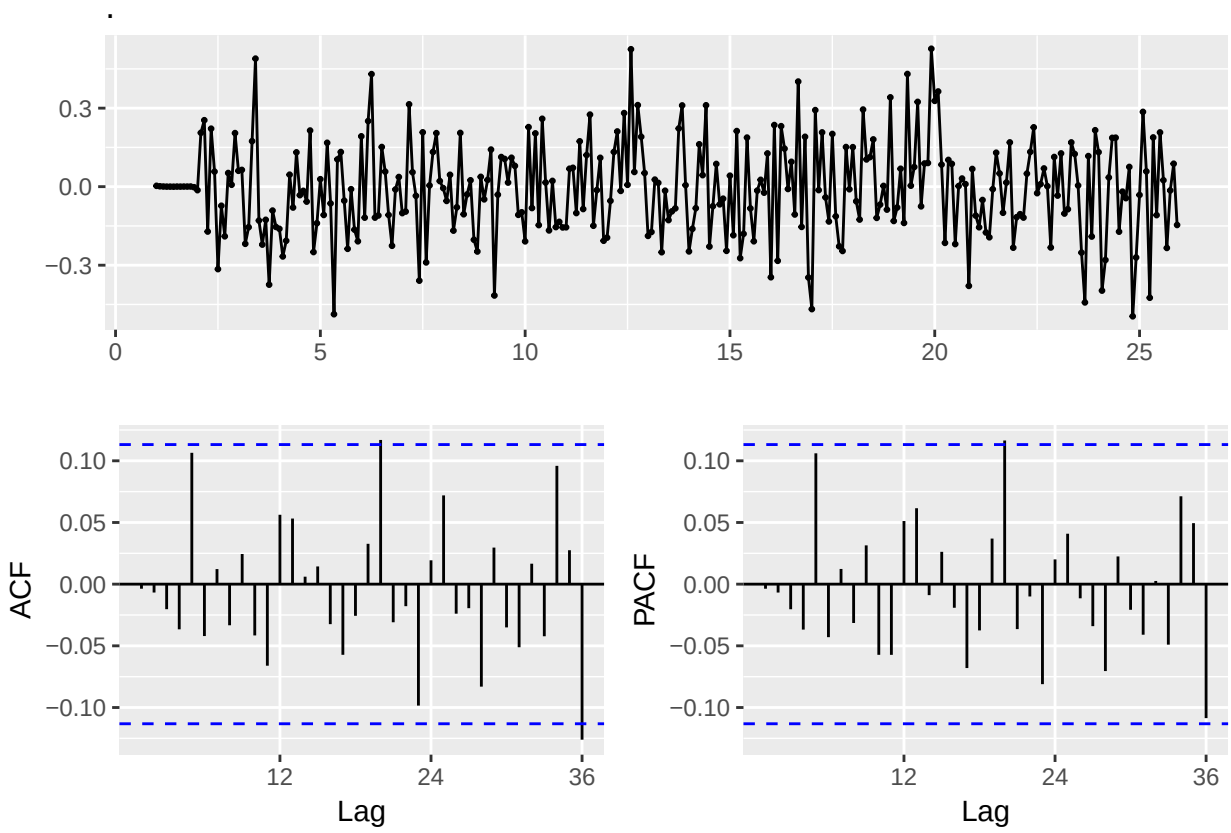
Defining a Function for Optimizing the Dependence Orders

Since the procedure was 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 =

  # 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
  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 to Optimize Model Dependence Orders

To expand the search space and identify any models that might surpass those we already estimated, we conducted a grid search.

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

```
##    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

We found that the grid search corraborated 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 decided to focus on the following models:

- ARIMA(1,1,1)(0,1,1)[12] because it minimized the BIC.
- ARIMA(2,1,1)(0,1,1)[12] because it minimized the AIC and AICc.

Model Parameter Estimation

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

```
## Series: .
## 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
##
## Training set error measures:
##                        ME      RMSE       MAE        MPE     MAPE
## Training set -0.009339336 0.1825311 0.1427566 -0.1059365 2.389875
##                    MASE       ACF1
## Training set 0.1763078 -0.1050292
```

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

```
## Series: .
## 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
##
## Training set error measures:
##                        ME      RMSE       MAE        MPE     MAPE
## Training set -0.008926836 0.1810804 0.1416233 -0.1017597 2.364688
##                    MASE         ACF1
## Training set 0.1749081 -0.004085812
```

The model parameters we estimated for both our candidates are all statistically significant (criterion = 2*SE<Parameter).

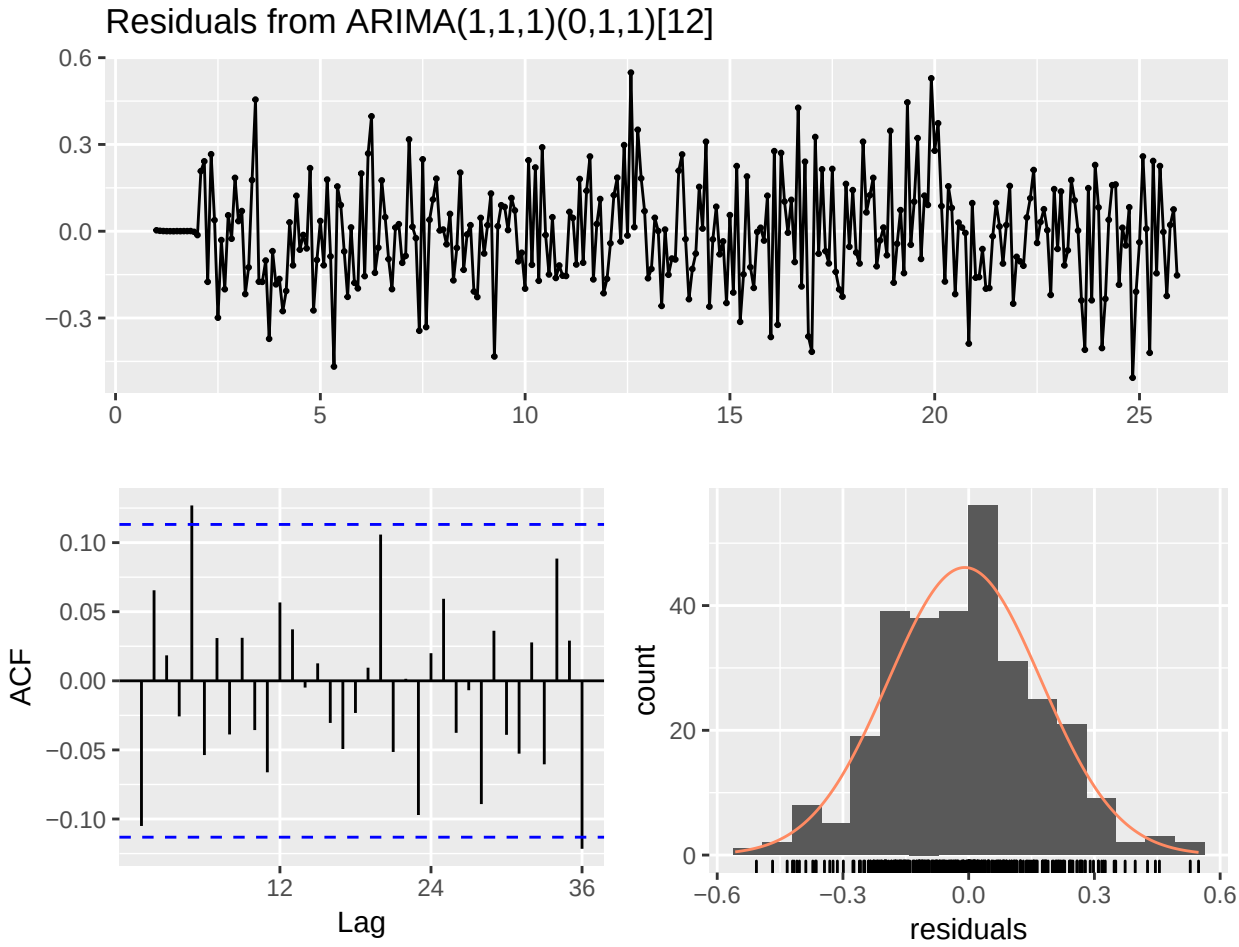
VI. Model Validation with Model Diagnostics

We performed and examined the model diagnostics for the two models we favored.

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

**we need a hist of residuals here. instead of ggtsdisplay(res111), we could do #checkresiduals(fit111) - this also performs the Ljung-box test

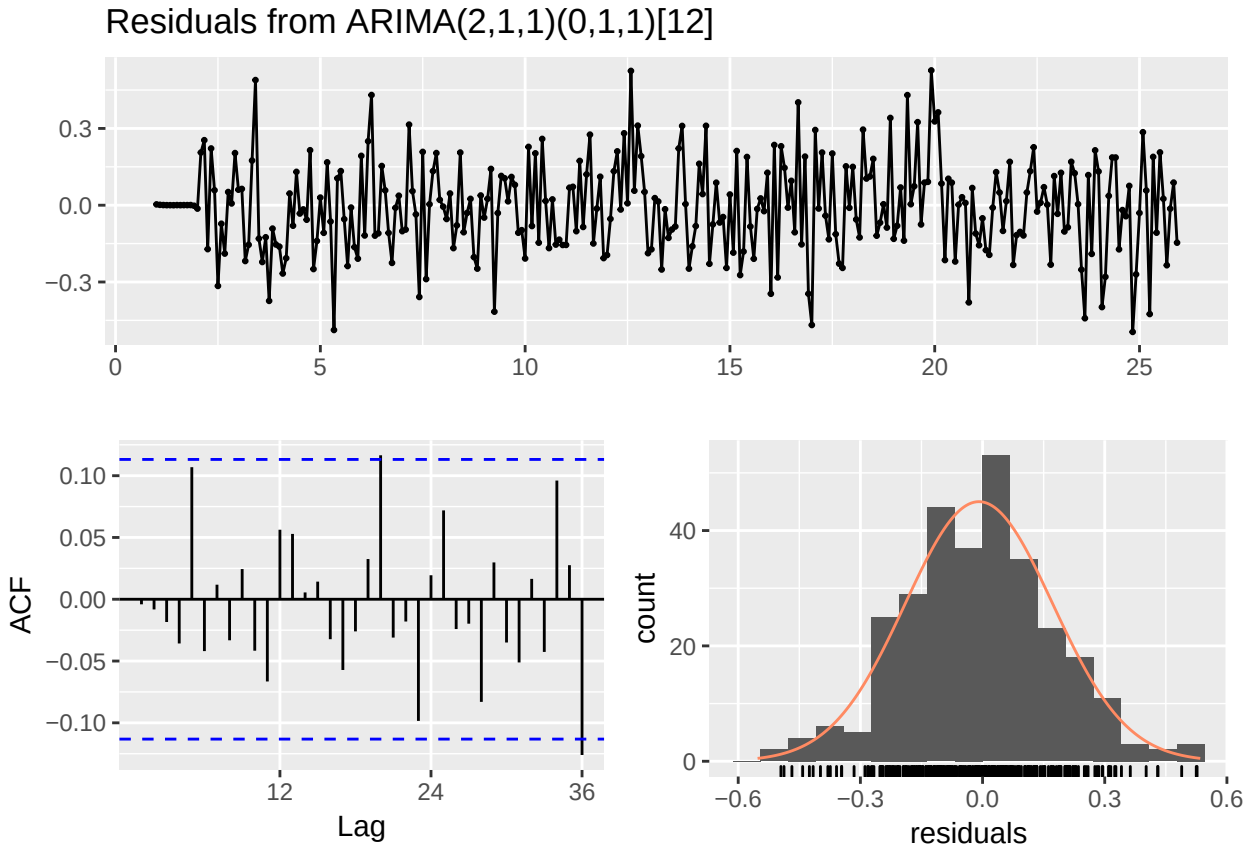
```r
res111 <- residuals(fit111)
#ggtsdisplay(res111)
checkresiduals(fit111)
```




```
##
##  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
```

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

```
res211 <- residuals(fit211)
#ggtsdisplay(res211)
checkresiduals(fit211)
```




```
##
##  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
```

The interpretation of the model diagnostics for both of our candidate models is similar. There are very few significant lags for the model residuals in ACF plots, and the Box-Ljung tests cannot reject the null hypothesis that the residuals have remaining autocorrelations (p=0.29, and p=0.52, respectively). Thus, we conclude that the model residuals sufficiently resemble white noise. In addition, the residuals from both models follow a normal distribution.

These diagnostic tests provide support for accepting the model assumptions required for valid time series modeling and forecasting. We can therefore proceed to forecasting with confidence.

VII. Forecasting

Standard for Evaluating our Model

We wanted to have some kind of "standard" for model performance, in order to evaluate the performance of our model accuracy. Naive forecast is typically used for as a standard against which forecasts are compared (ref?). One hopes that thier model forecast is more accurate than simply using what has either just happened or happened at the same time period last year to forecast. In our case, we wanted to know if our favored model would outperform simply using the same values for last year as the current year's forecast.

```
#insert naive seasonal forcast calculation of MAPE from Kiersten V1.
```

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

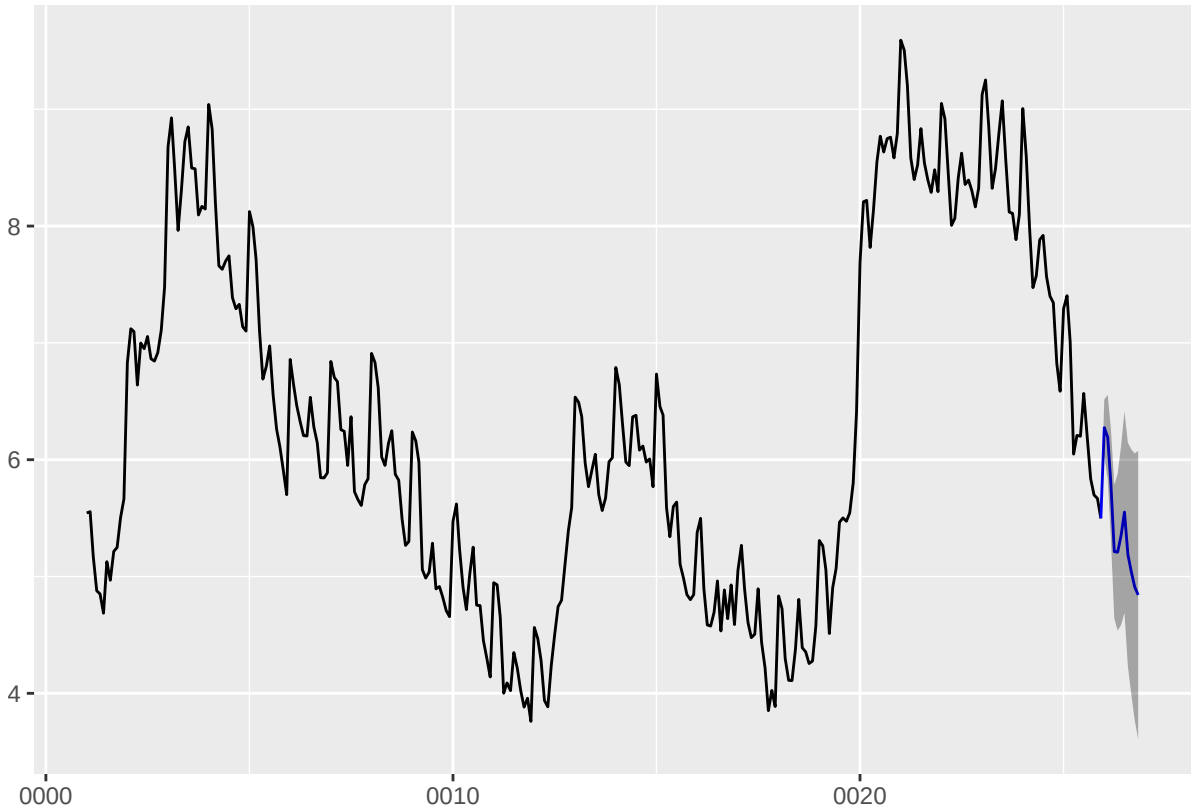
```
forecast111 <- fit111 %>% forecast(h=11)
forecast211 <- fit211 %>% forecast(h=11)
(results <- rbind( accuracy(forecast111,ms.test),
                   accuracy(forecast211,ms.test))[,1:5])
```

```
##                         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 compared our forecast fit to our reserved test data using the MAPE score. We note that the $ARIMA(1,1,1)(0,1,1)_{12}$ has not only a lower score than the $ARIMA(2,1,1)(0,1,1)_{12}$, (3.2 versus 3.8), but also the lowest out of sample MAPE score for any of the models we estimated. We therefore performed forecasting with the $ARIMA(1,1,1)(0,1,1)_{12}$ model.

*__expand__ In addition, our favored model forecast substantially outperformed the seasonal naive forecast in terms of MAPE. We are therefore confident that we have developed a valid and accurate forecast, which can be seen below.

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



The 11 month-ahead seasonal ARIMA model forecast follows the recent trend in the data due to the combination of seasonal and trend differencing. The confidence intervals are quite small for the first third of the forecast, however the confidence intervals rapidly increase during the latter two-thirds of the forecast. The point forecasts initially increase and then trend downward. The prediction interval also initially increases

but allows for the data to either trend upwards or downwards during the latter two-thirds of the forecast period. We conclude that, as expected, our model is better at predicting the near-future.

Based on our forecast we expect the value of the dependent variable to continue to show seasonality over the next year and to mimic the downward trend present in the most recently observed values. We must however, be mindful of the possibility that the series will experience one of its unpredictable sudden upward shifts during the forecast period. If an upward shift occured, it would lead to considerably different and higher point values than our forecast predicts.

?Can you overlay the actual values on the forecast plot???

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 statisitcally 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 x 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.

Briefly summarize forecast predictins again...

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.