

Time Series 413, Assignment 9

Non-linear Modeling and Model Monitoring (TS9)

Reed Ballesteros

Northwestern University SPS, Fall 2022

MSDS-413-DL

Instructor: Dr. Jamie D. Riggs, Ph.D

2022-11-21

The following list defines the data sets and their respective variables.

- AustralianWine.csv (volumes are 1000x Kiloliters)
 - Month
 - Fortified
 - Red
 - Rose
 - sparkling
 - Sweet.white
 - Dry.white
- Bike data (Washington, D.C.): <https://www.kaggle.com/c/bike-sharing-demand/dataset?select=train.csv>, Filename: bikes.csv
 - date-time: YYYY-MM-DD HH:MM:SS
 - season: 1, 2, 3, or 4
 - holiday: 0 = not a holiday, 1 = holiday
 - workingday: 0 = not a workday, 1 = workday
 - weather: 4 classes, 1, 2, 3, 4
 - temp: temperature, degrees Celsius
 - atemp: apparent temperature, degrees Celsius
 - humidity: mm Hg
 - windspeed: km/hr
 - casual: 0 = not casual, 1 = casual
 - registered: 0 = not registered, 1 = registered
 - count: total number of bike rentals

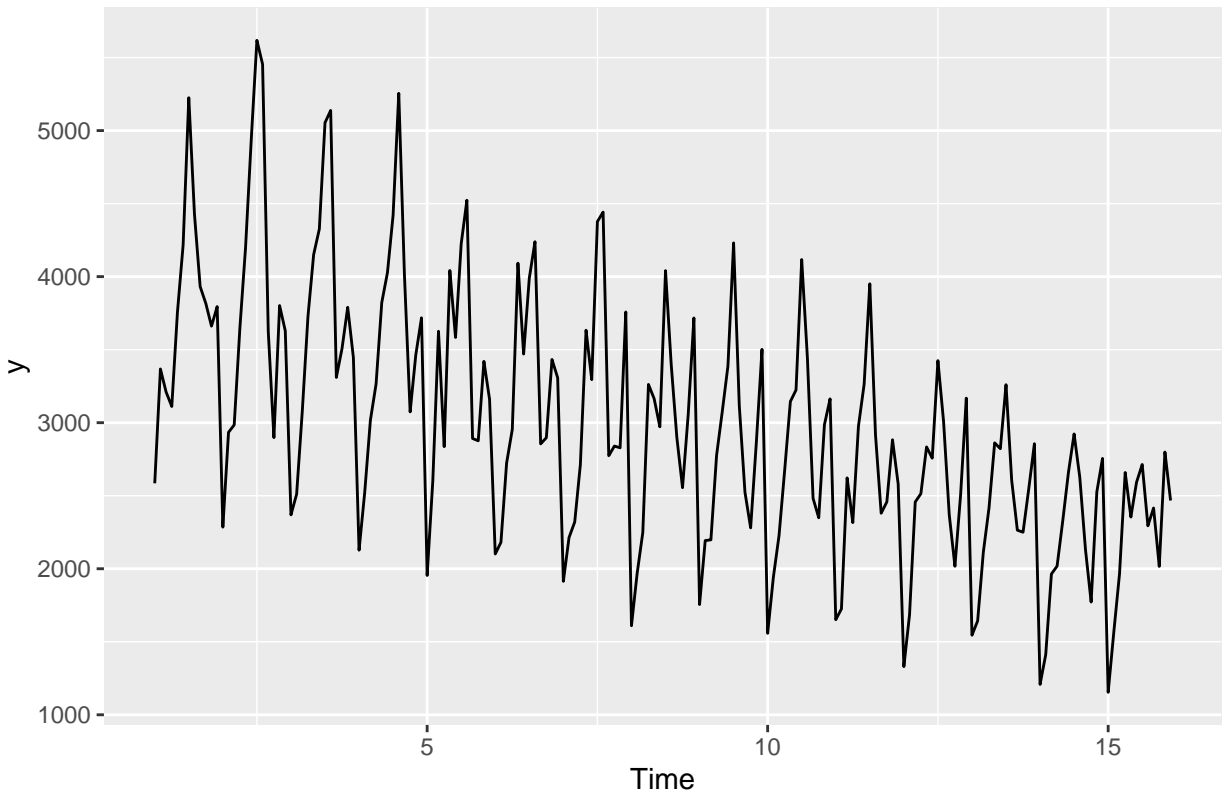
The objective is to explore the time series behavior of these data sets including EDA, modeling, model diagnostics, and interpretation.

1. Exponential Smoothing (50 points). For this exercise, use the monthly Australian wine

sales Fortified data. (Data set: AustralianWine.csv.)

1.1. Perform EDA. Why is a multiplicative seasonal model necessary?

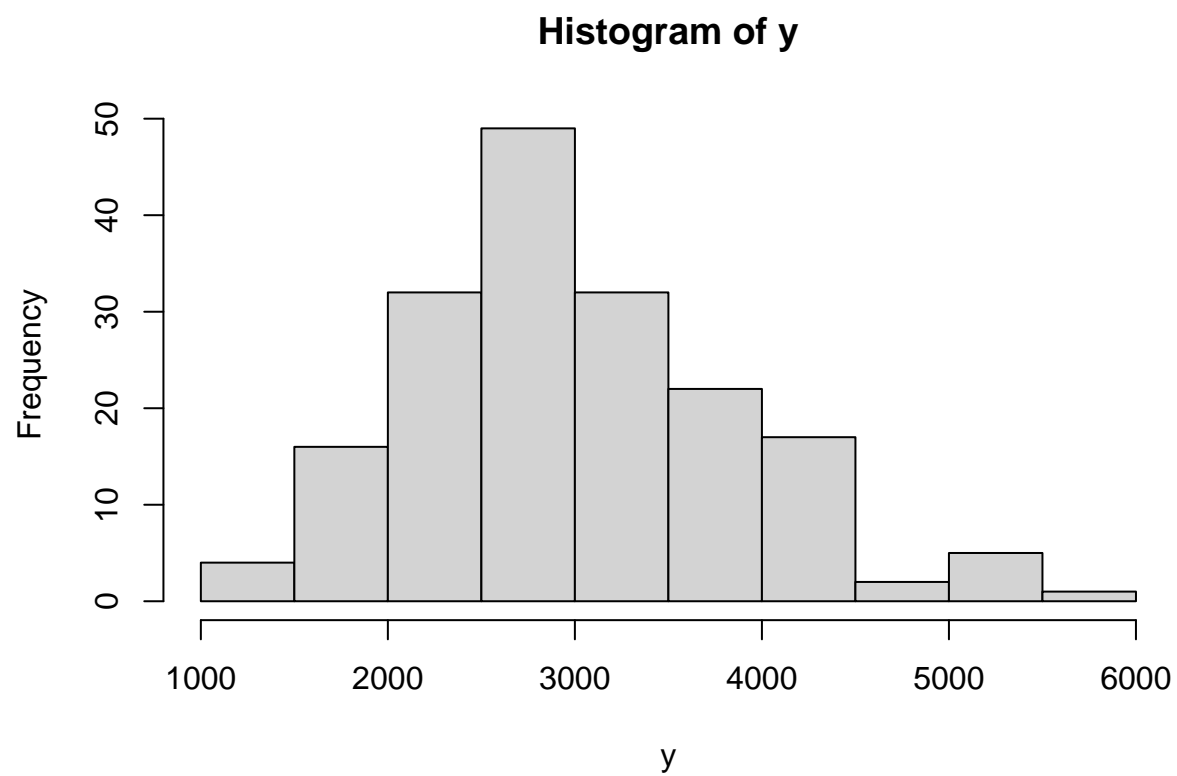
Let us plot the Australian Wines sales Fortified time series data:



We notice a downward trend, with the seasonality getting smaller as the plot trends downwards over time.

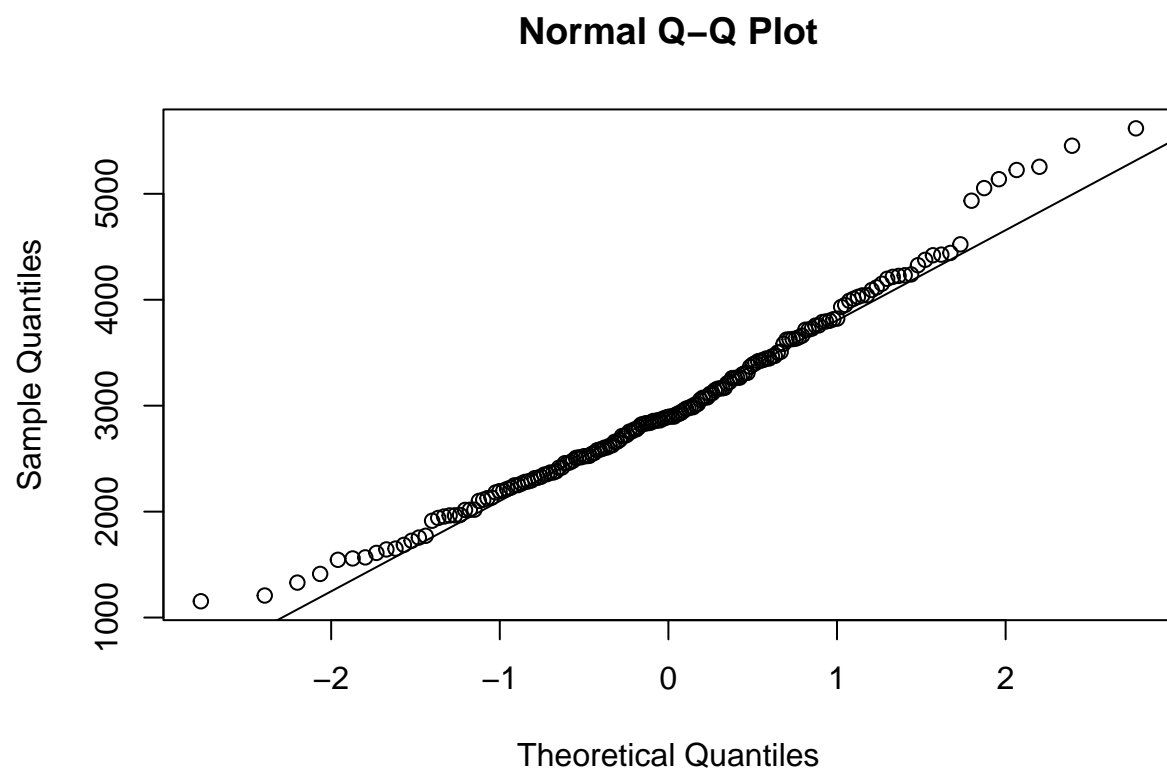
In regards to normalcy, we do not observe a mean 0, nor constant variance. We will not transform the data to make it fit a normal form as this exponential exercise does not require it.

Histogram: Fortified Sales Data



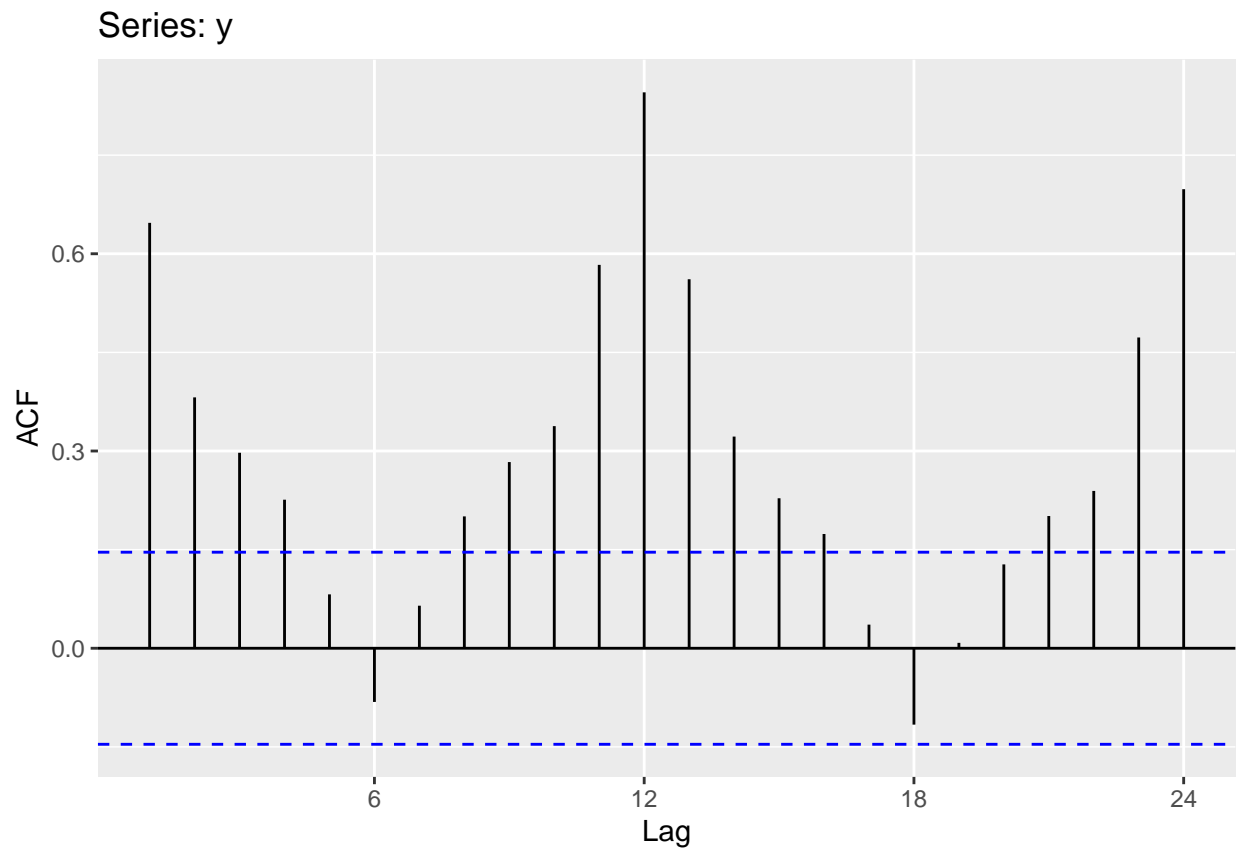
We see a right-skewed, somewhat tall histogram. The data is not normal based on a Gaussian PDF.

Q-Q Plot: Fortified Sales Data



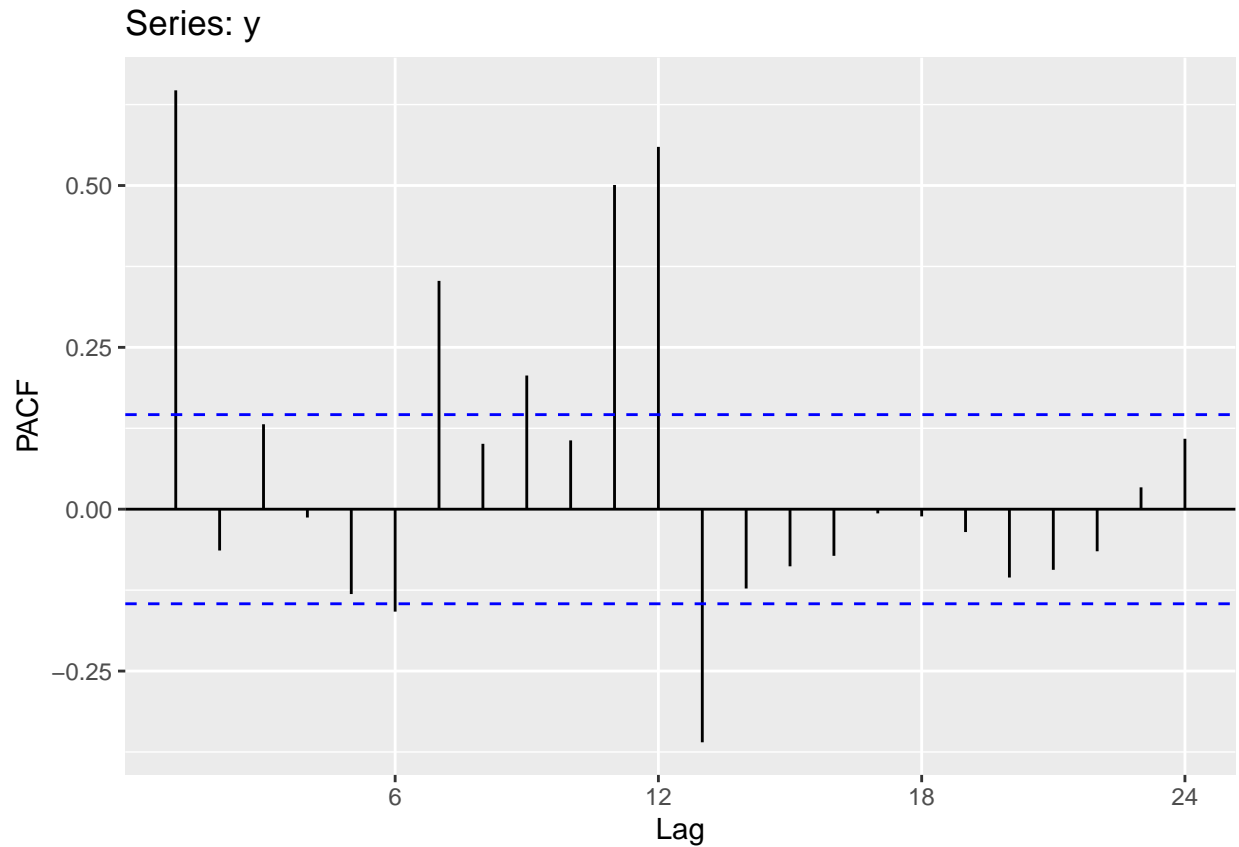
Most data is on the ideal normal line, but the tails flare upward.

ACF Plot: Fortified Sales Data



We can observe a season pattern of 12 months peak-to-peak.

PACF Plot: Fortified Sales Data

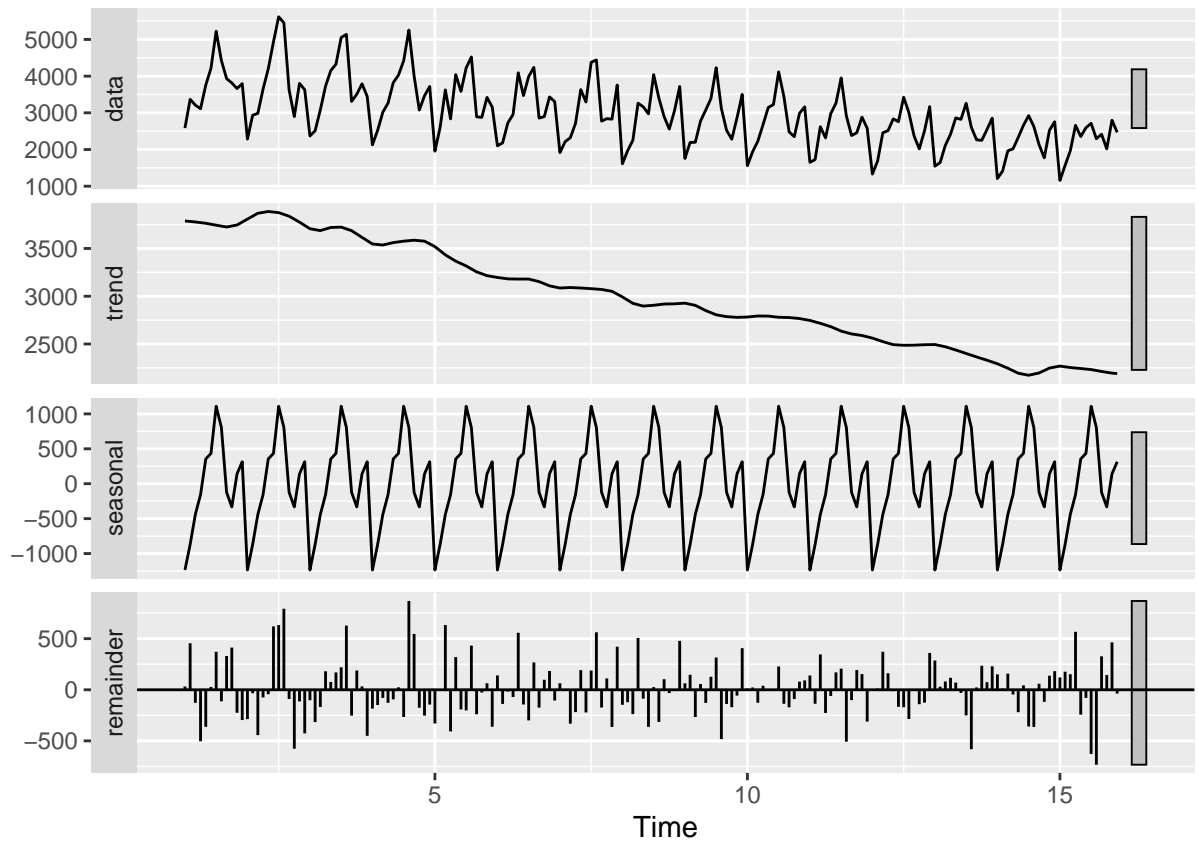


We find the PACF plot to be fairly stationary, with few lags outside of the 95% confidence interval (CI) threshold.

We will not conduct other tests for normalcy under a Gaussian PDF as exponential smoothing does not require times series data that fits a normal form.

A multiplicative model is necessary for this particular time series due to that the seasonal component is not constant; we can see from the plot that the seasonality is decreasing based on the downward trend over time.

From the decomposition:



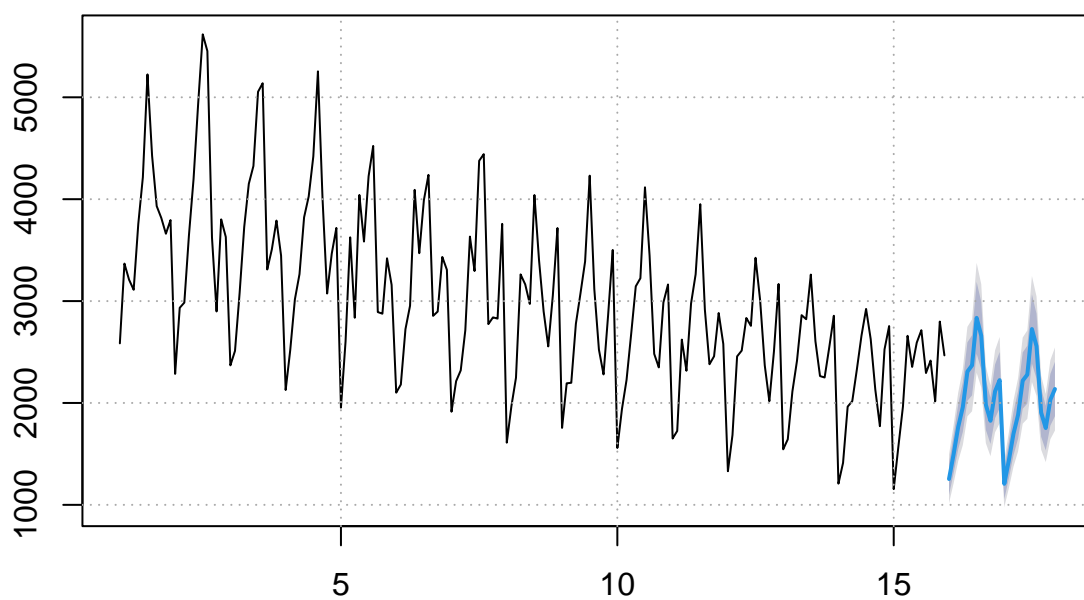
While we are able to decompose constant seasonal component and the trend separately, we can see with the downward level trend influence the seasonal component in the data over time in the data, thus the need for a multiplicative model over an additive one.

1.2. Forecast the next two years using Holt-Winters multiplicative method.

Forecast: Multiplicative Method

```
fit_multi <- hw(y,seasonal="multiplicative")
```

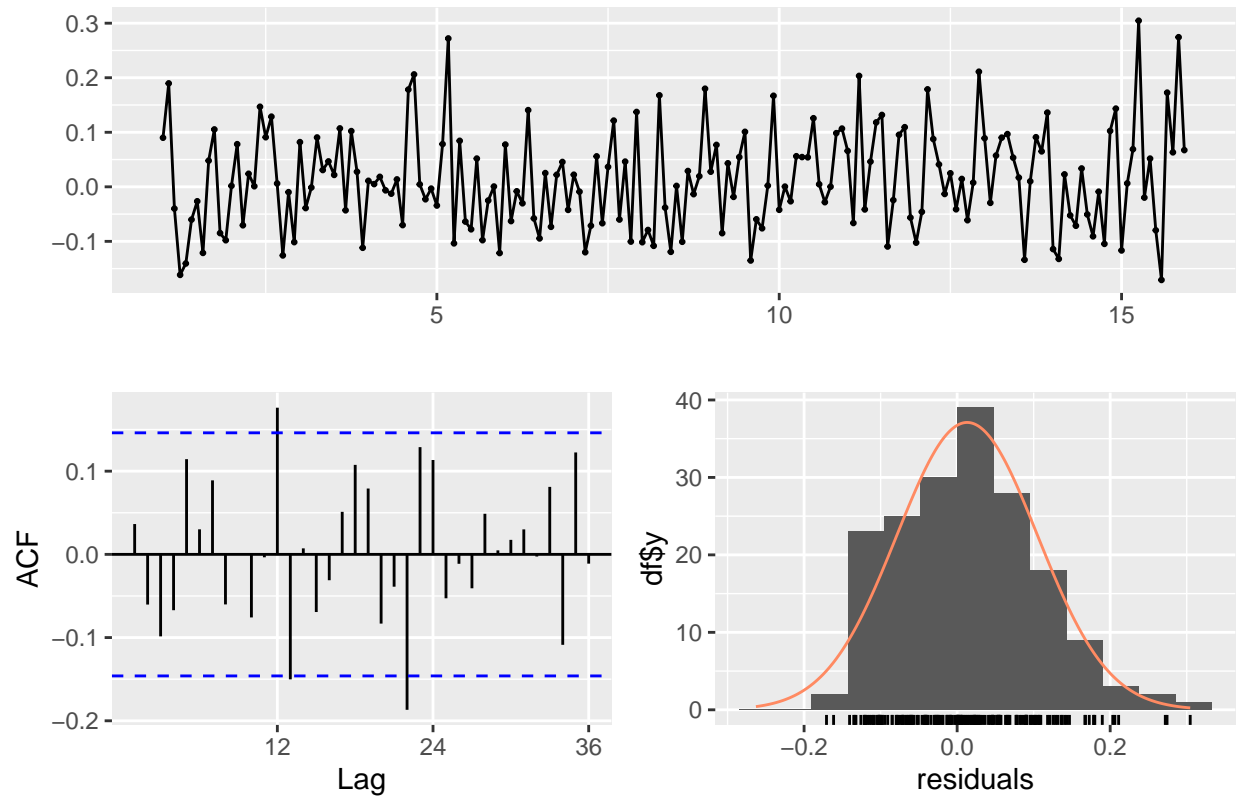

Forecasts from Holt–Winters' multiplicative method



We observe a 2-year forecast continuing the downward trend, with the seasonal component getting smaller as well.

Residuals: Multiplicative Method

Residuals from Holt–Winters' multiplicative method



```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' multiplicative method
## Q* = 40.37, df = 8, p-value = 2.733e-06
##
## Model df: 16.    Total lags used: 24
```

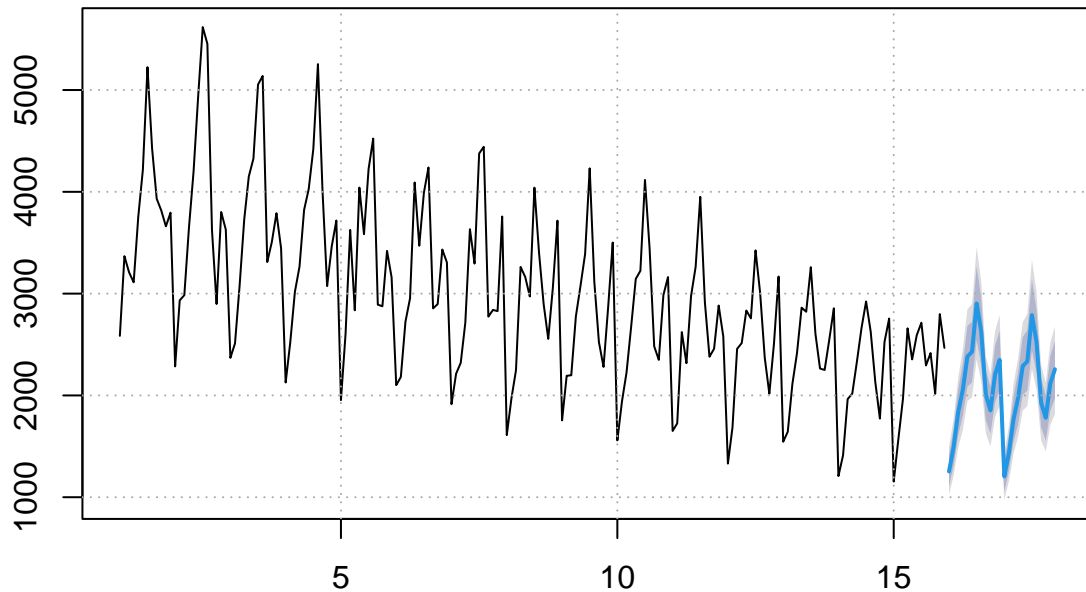
The histogram of the residuals does not have the most ideal symmetry (somewhat right-skewed) but the plot shows close to mean 0 with signs of constant variance.

1.3. Experiment with making the trend exponential and/or damped.

Forecast: Exponential Trend

```
fit_multi_exp <- hw(y,seasonal="multiplicative",exponential=TRUE)
```

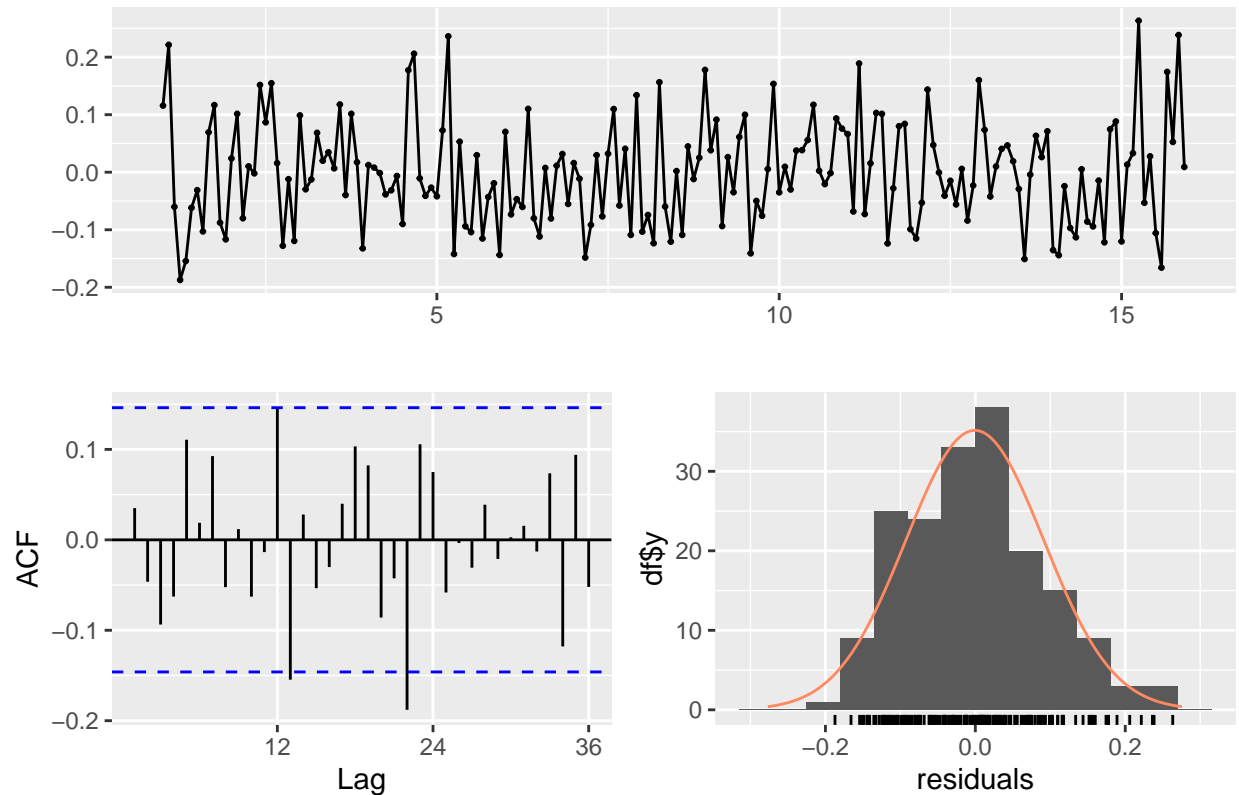
Forecasts from Holt–Winters' multiplicative method with exponential tr



We observe a 2-year forecast continuing the downward trend, with the seasonal component getting smaller as well. The exponential downward curve is subtle in this respect.

Residuals: Multiplicative Method

Residuals from Holt–Winters' multiplicative method with exponential trend



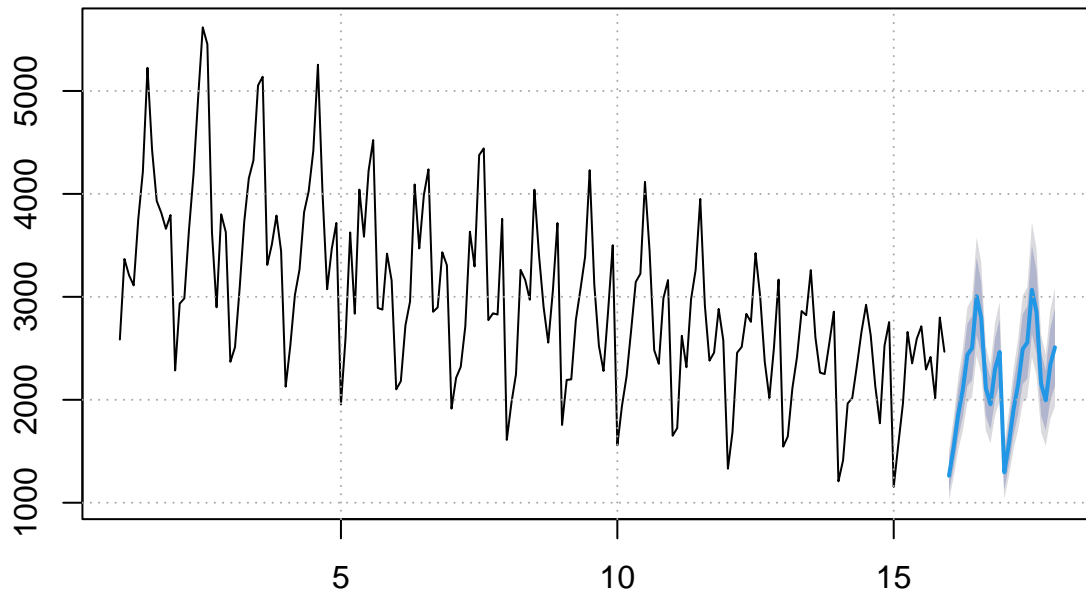
```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' multiplicative method with exponential trend
## Q* = 34.611, df = 8, p-value = 3.147e-05
##
## Model df: 16.    Total lags used: 24
```

The histogram of the residuals has slightly better symmetry compared to the multiplicative model, and the plot shows close to mean 0 with signs of constant variance.

Forecast: Multiplicative Method, Damped

```
fit_multi_damped <- hw(y,seasonal="multiplicative",damped=TRUE)
```

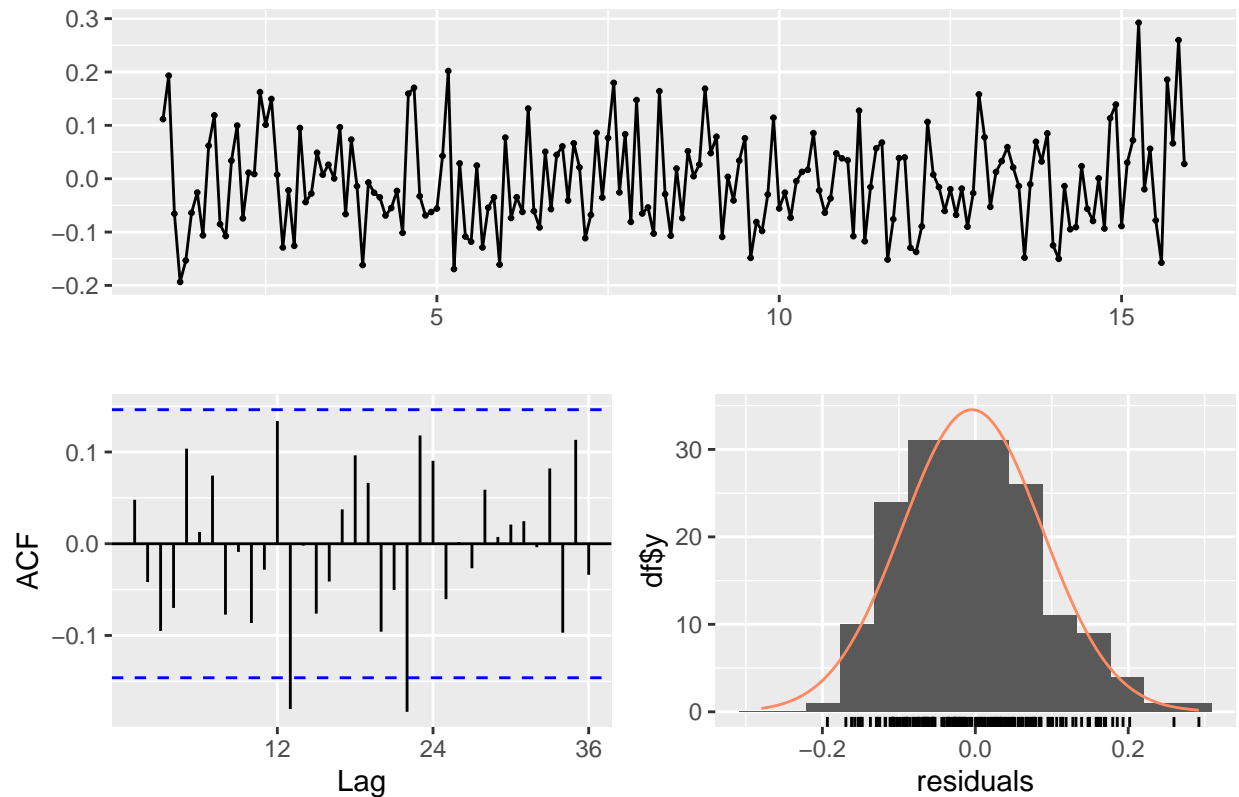
Forecasts from Damped Holt–Winters' multiplicative method



We observe the trend to slightly rise and plateau, which is the usual effect of a damping forecast.

Residuals: Multiplicative Method, Damped

Residuals from Damped Holt–Winters' multiplicative method



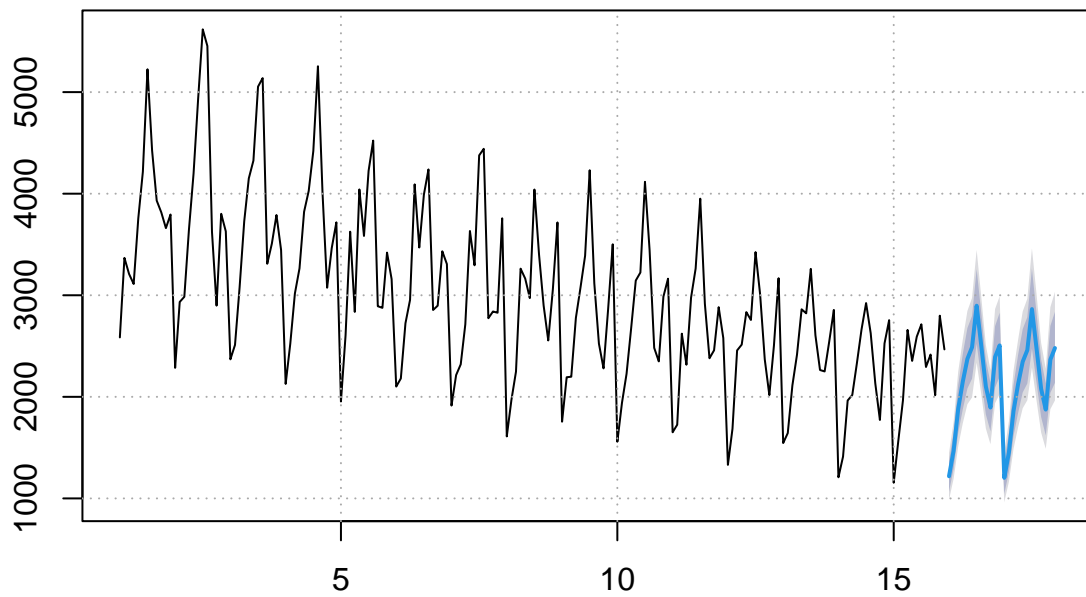
```
##
##  Ljung-Box test
##
## data:  Residuals from Damped Holt-Winters' multiplicative method
## Q* = 37.56, df = 7, p-value = 3.673e-06
##
## Model df: 17.    Total lags used: 24
```

The histogram of the residuals display slightly better symmetry the plot also shows mean 0 with signs of constant variance.

Forecast: Multiplicative Method, Exponential and Damped

```
fit_multi_exp_damped <- hw(y, seasonal="multiplicative", exponential=TRUE, damped=TRUE)
```

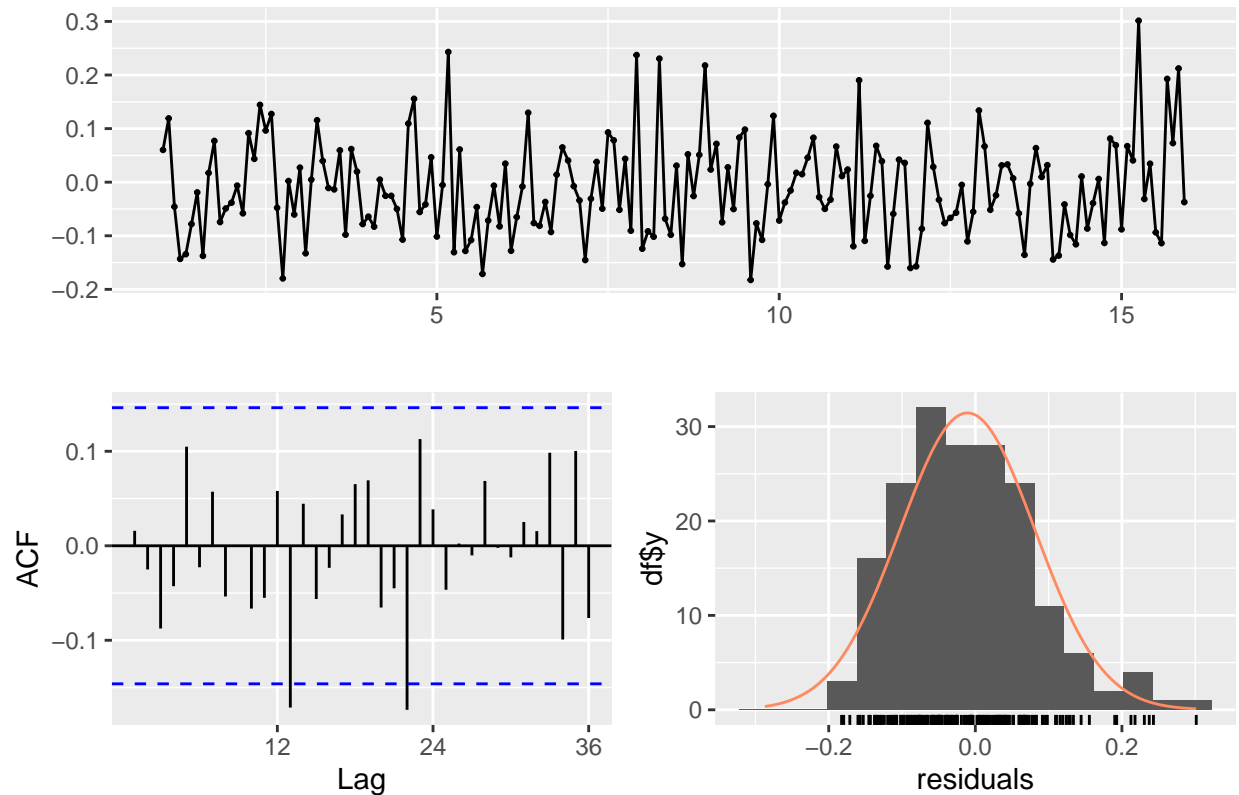
casts from Damped Holt–Winters' multiplicative method with exponent



We observe the trend to to slightly flatten, the result of integrating both exponential and damped effects.

Residuals: Multiplicative Method, Exponential and Damped

Residuals from Damped Holt–Winters' multiplicative method with exponent



```
##
##  Ljung-Box test
##
## data:  Residuals from Damped Holt-Winters' multiplicative method with exponential trend
## Q* = 26.782, df = 7, p-value = 0.0003648
##
## Model df: 17.    Total lags used: 24
```

The histogram of the residuals display better symmetry than the other models, but still right-skewed, with the plot also showing mean 0 with signs of constant variance.

1.4. Compare the RMSE of the one-step forecasts from the various methods. Which do you prefer?

RMSE: Multiplicative Method

```
##           ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 33.35988 281.1189 217.9141 0.4996233 7.174729 0.7961545 0.06051479
```

RMSE: Multiplicative with Exponential Trend

```
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -6.877928 288.9563 223.3168 -0.8944623 7.383183 0.8158933
##           ACF1
## Training set 0.0726839
```


RMSE: Multiplicative with Damped Trends

```
##           ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set -18.84945 290.4883 227.8256 -1.276021 7.56037 0.8323665 0.08025566
```

RMSE: Multiplicative with Exponential and Damped Trends

```
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -34.45949 286.6223 226.6717 -1.930774 7.616595 0.8281504
##           ACF1
## Training set 0.06101945
```

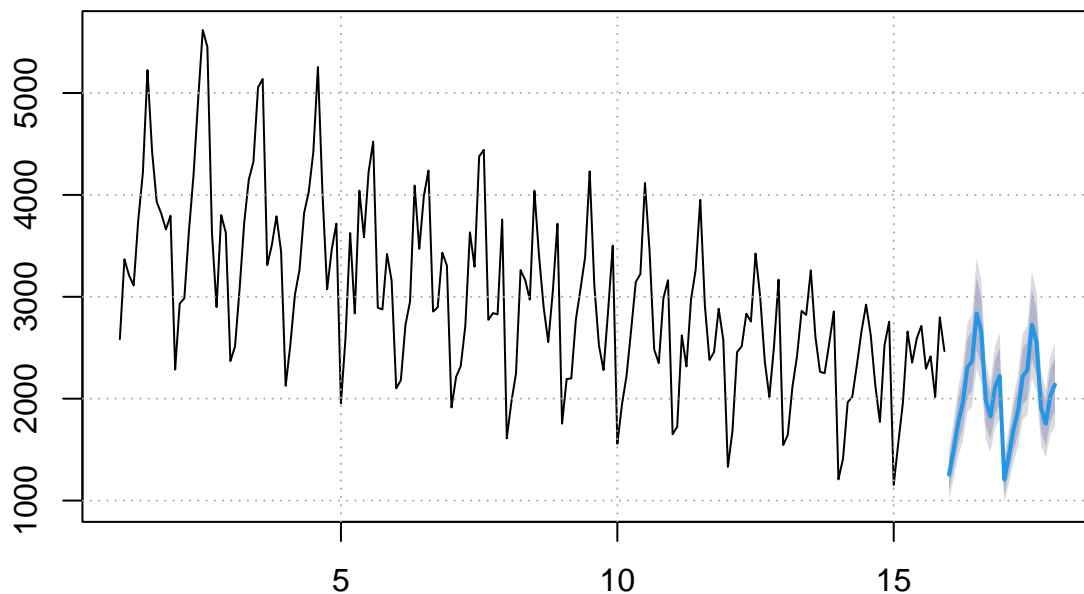
The model fitted with only the multiplicative seasonal trend yields the lowest RMSE of 281.1189, thus being the best-fitting model of the four we tested. It also follows the general multiplicative seasonal trend of the time series data, as we can observe and compare from the respective plots.

1.5. Now fit each of the following models to the same data using ets:

1.5.1. a multiplicative Holt-Winters method Forecast: Multiplicative Method

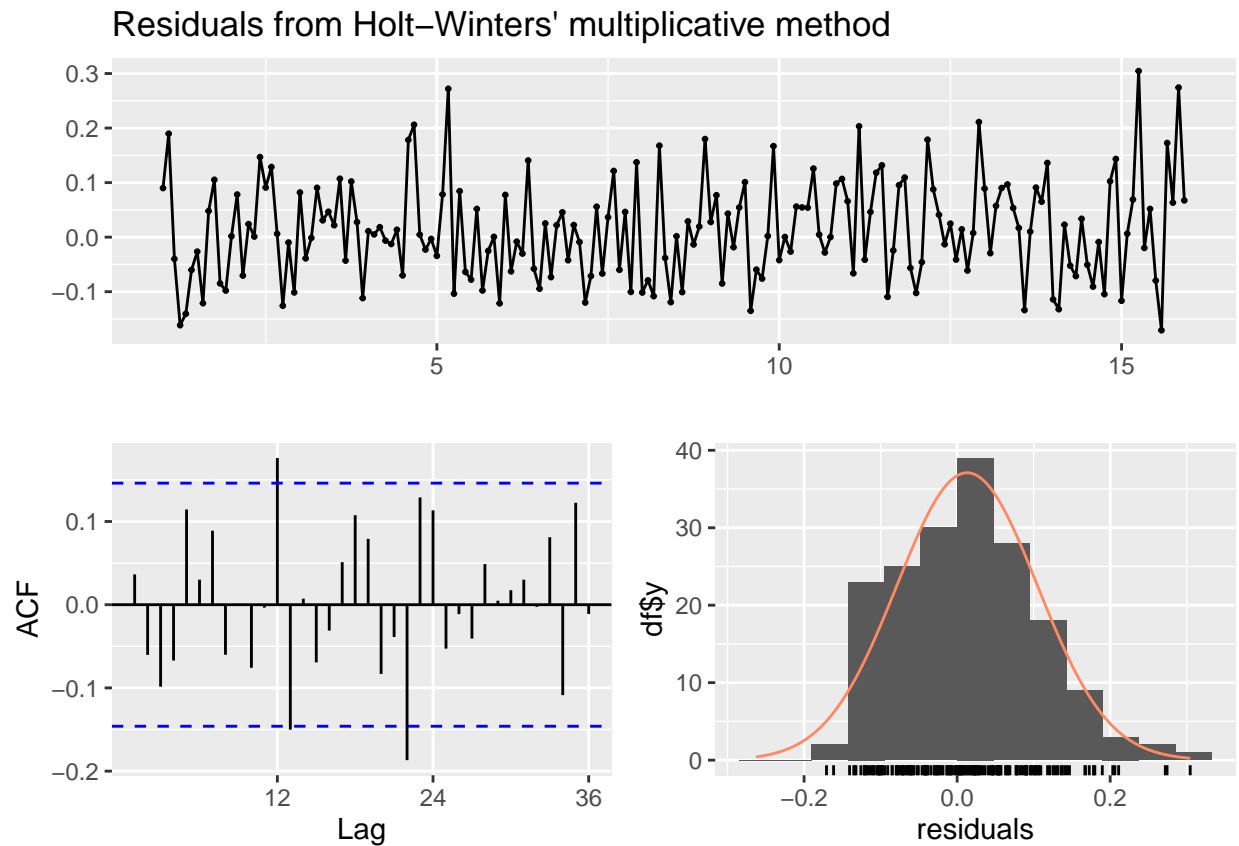
```
fit_multi <- hw(y,seasonal="multiplicative")
```

Forecasts from Holt-Winters' multiplicative method



We observe a 2-year forecast continuing the downward trend, with the seasonal component getting smaller as well.

Residuals: Multiplicative Method



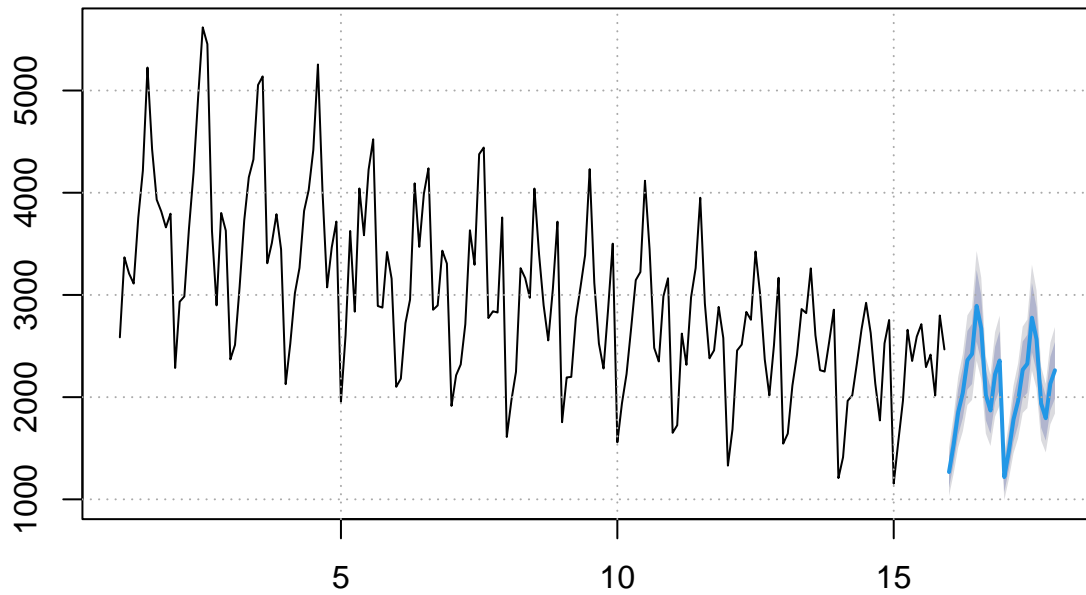
```
##  
## Ljung-Box test  
##  
## data: Residuals from Holt-Winters' multiplicative method  
## Q* = 40.37, df = 8, p-value = 2.733e-06  
##  
## Model df: 16. Total lags used: 24
```

The histogram of the residuals does not have the most ideal symmetry (somewhat right-skewed) but the plot shows close to mean 0 with signs of constant variance.

1.5.2. an ETS model Forecast: ETS Model

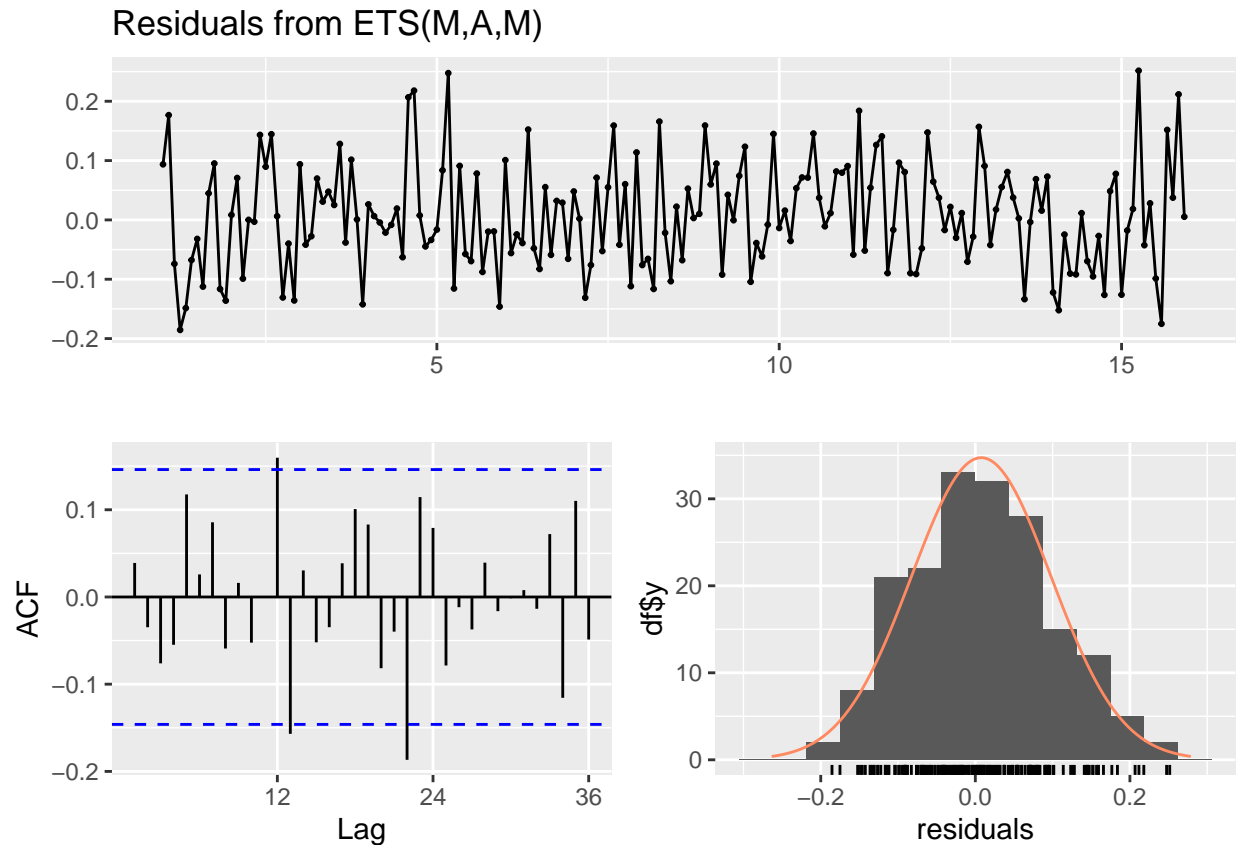
```
# ETS model  
fit_mam <- ets(y, model="MAM")
```

Forecasts from ETS(M,A,M)



We also observe a 2-year forecast continuing the downward trend, similar to the multiplicative model, with the seasonal component getting smaller as well.

Residuals: ETS Model



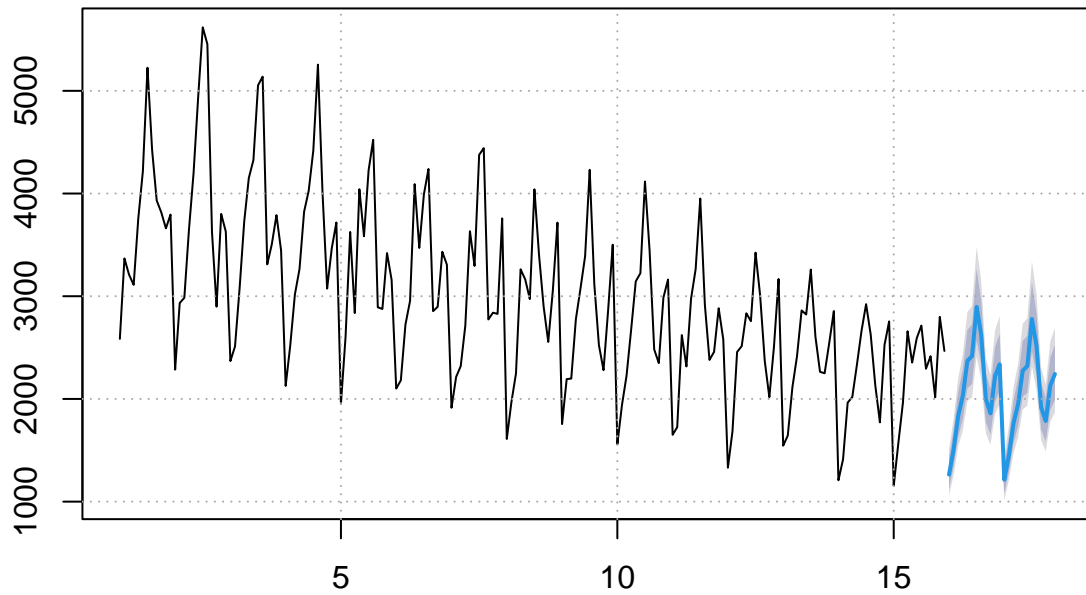
```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,A,M)
## Q* = 34.964, df = 8, p-value = 2.714e-05
##
## Model df: 16.    Total lags used: 24
```

The histogram of the residuals of the ETS model has better symmetry and the plot shows close to mean 0 with signs of constant variance.

1.5.3. an additive ETS model applied to a Box-Cox transformed series Forecast: Additive ETS Model with Box-Cox Transformation

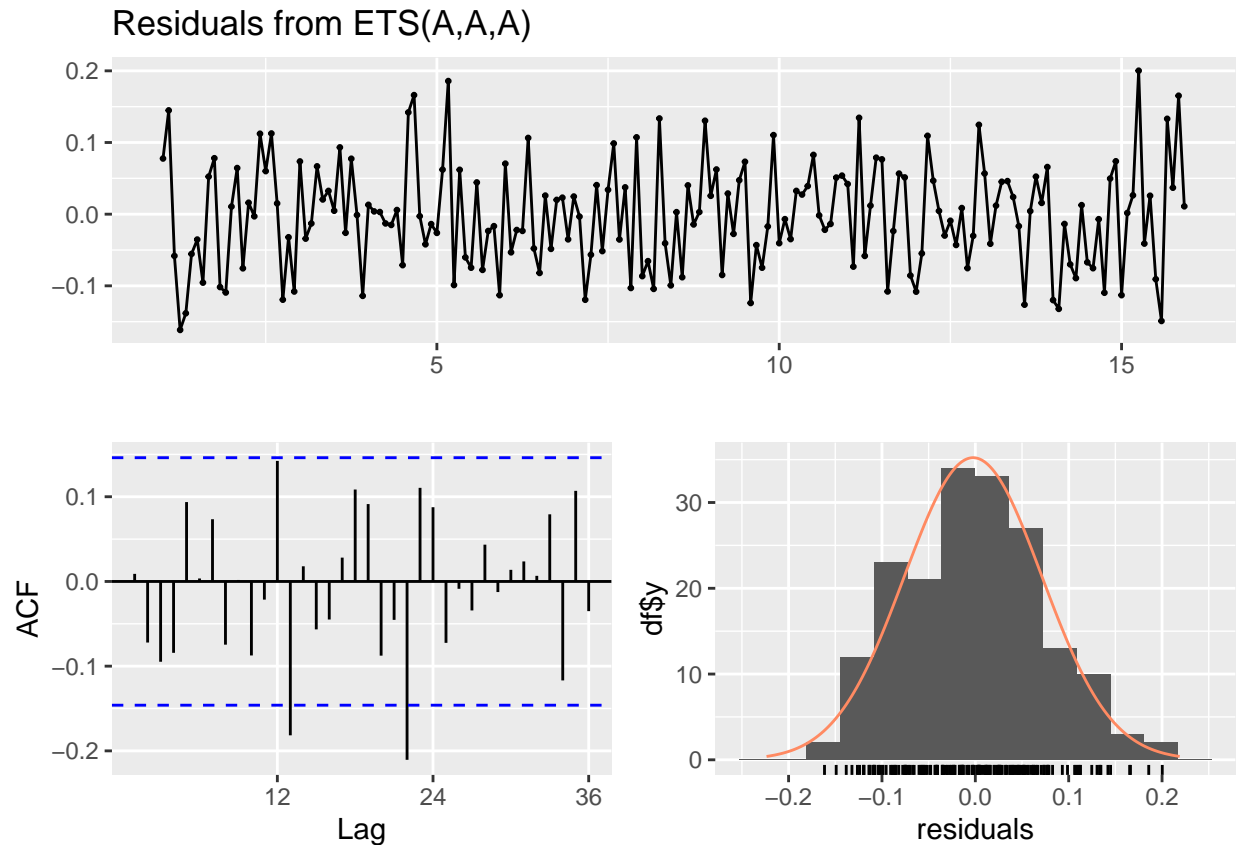
```
# Additive ETS model with Box-Cox transformation
fit_ana_box <- ets(y,additive.only=TRUE,lambda="auto")
```

Forecasts from ETS(A,A,A)



We observe a 2-year forecast continuing the downward trend, with the seasonal component getting smaller as well. It is not dramatically different from the previous two models.

Residuals: Additive ETS Model with Box-Cox Transformation



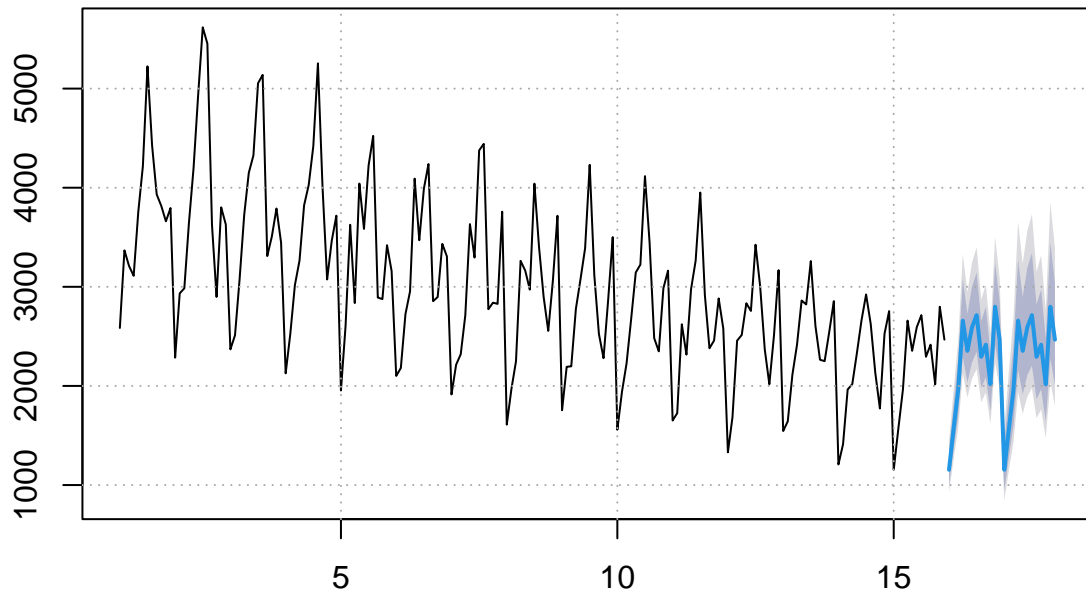
```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,A,A)
## Q* = 40.361, df = 8, p-value = 2.744e-06
##
## Model df: 16.    Total lags used: 24
```

The histogram of the residuals of the ETS Box-Cox model has improved symmetry and the plot shows close to mean 0 with signs of constant variance.

1.5.4. a seasonal naive method applied to the Box-Cox transformed series Forecast: Seasonal Naive Model with Box-Cox Transformation

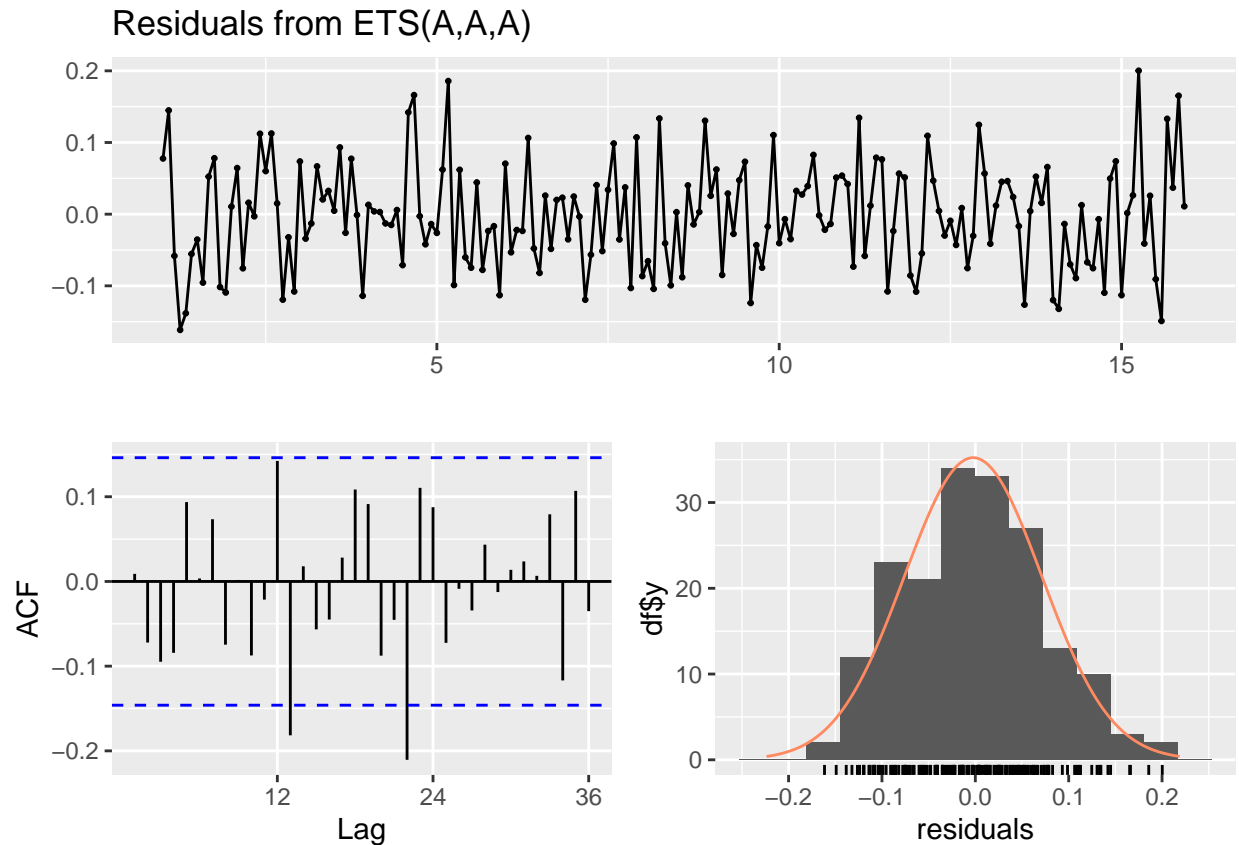
```
# Seasonal naive with Box-Cox transformation
fit_naive <- snaive(y,lambda="auto")
```

Forecasts from Seasonal naive method



The two-year forecast is very interesting as it seems to mimic the last seasonal time series data, with the trend starting to flatten. We also notice the confidence interval noted by the grey areas to get wider in the second year of the forecast compared to the first year.

Residuals: Seasonal Naive Model with Box-Cox Transformation



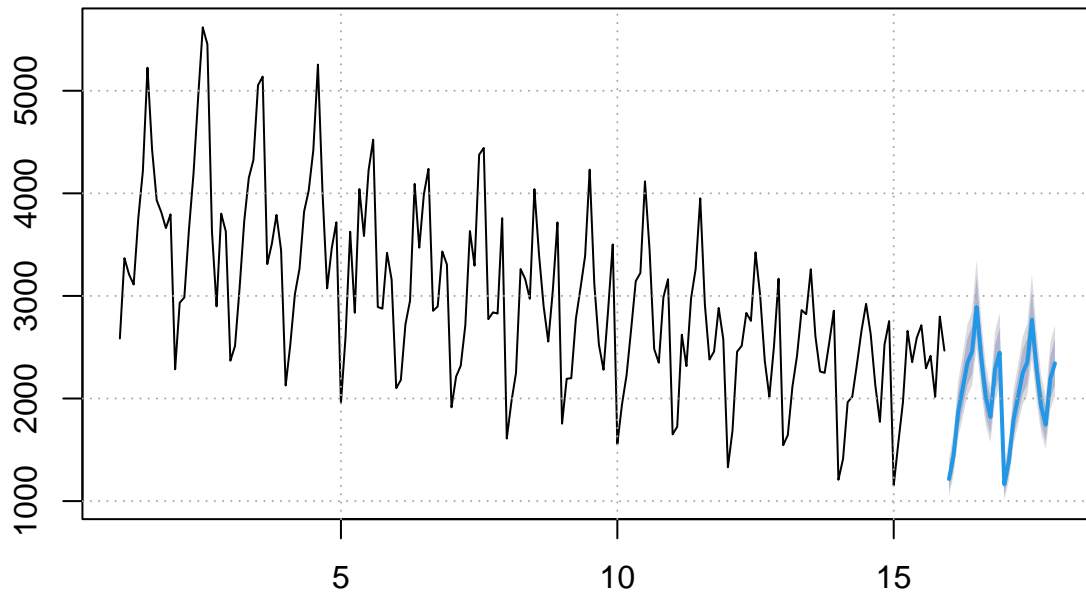
```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,A,A)
## Q* = 40.361, df = 8, p-value = 2.744e-06
##
## Model df: 16.    Total lags used: 24
```

The histogram of the residuals of the seasonal naive model has decent symmetry and the plot shows close to mean 0 with signs of constant variance.

1.5.5. a STL decomposition applied to the Box-Cox transformed data plus ETS model applied to the seasonally adjusted (transformed) data. Forecast: STL decomposition with Box-Cox transformation on ETS

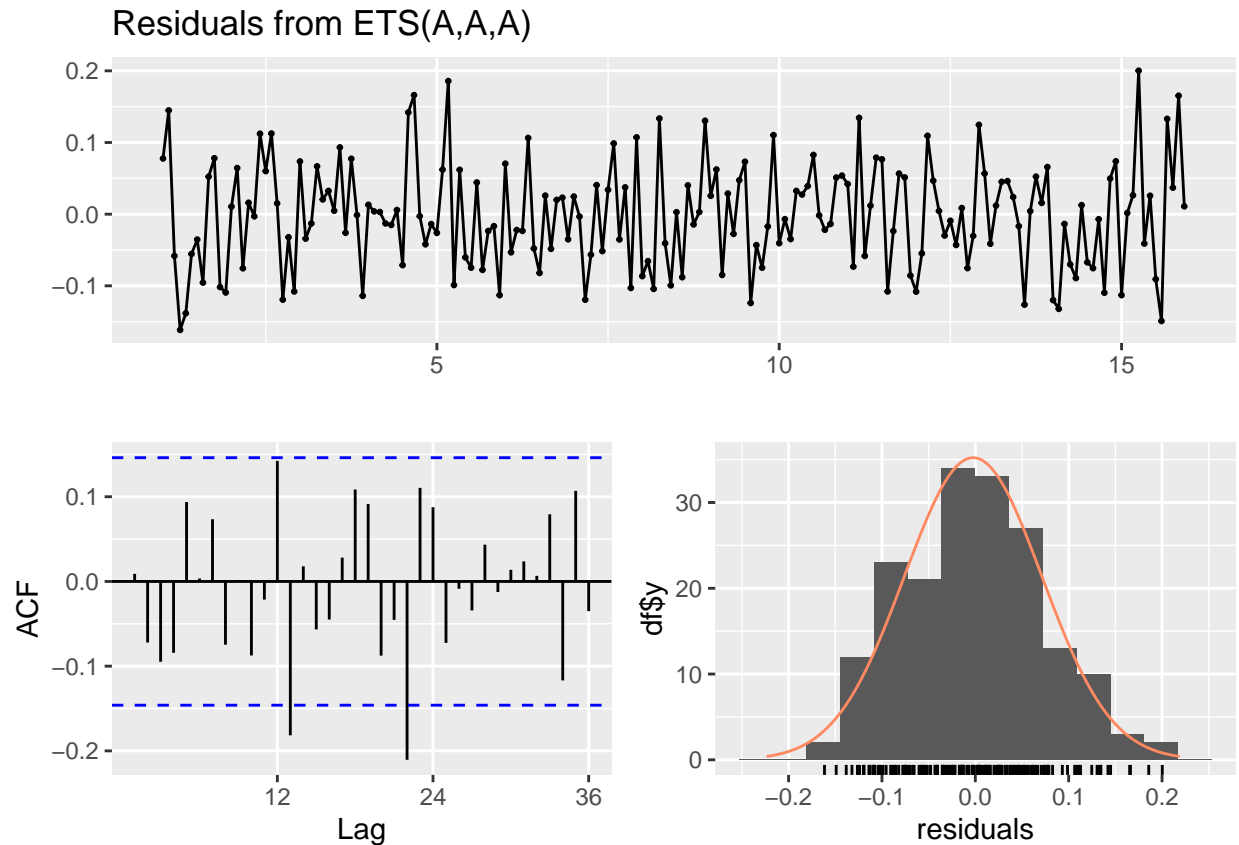
```
# STL decomposition with Box-Cox transformation on ETS
fit_stld <- stlf(y,method="ets",lambda="auto")
```


Forecasts from STL + ETS(A,A,N)



We observe a 2-year forecast continuing the downward trend, with the seasonal component getting smaller as well. It is not dramatically different from the multiplicative or default ETS model.

Residuals: STL decomposition with Box-Cox transformation on ETS



```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,A,A)
## Q* = 40.361, df = 8, p-value = 2.744e-06
##
## Model df: 16.    Total lags used: 24
```

The histogram of the residuals of the ETS model has good symmetry and the plot shows close to mean 0 with signs of constant variance.

1.6. For each model, look at the residual diagnostics and compare the forecasts for the next two years. Which do you prefer?

RMSE: Multiplicative Method

```
##           ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 33.35988 281.1189 217.9141 0.4996233 7.174729 0.7961545 0.06051479
```

RMSE: ETS Model

```
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 19.99363 287.4642 221.5388 -0.02963343 7.222422 0.8093973
##           ACF1
## Training set 0.08533949
```

RMSE: ETS Model with Box-Cox Transformation

```
##                ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.9057893 284.5196 219.3023 -0.6744277 7.222296 0.8012263
##                ACF1
## Training set 0.05499586
```

RMSE: Seasonal Naive Model with Box-Cox Transformation

```
##                ME      RMSE      MAE      MPE      MAPE  MASE      ACF1
## Training set -107.756 350.2644 273.7083 -4.432429 9.567052    1 0.1442344
```

RMSE: STL decomposition with Box-Cox transformation on ETS

```
##                ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.5990637 236.7522 177.6653 -0.556021 5.880365 0.6491045
##                ACF1
## Training set 0.08056433
```

The model with STL decomposition and Box-Cox transformation on ETS from section 1.5.5 yields the lowest RMSE of 236.7522, thus being the best fitting model of the five we tested.

2. Prophet 1 (25 points) Use the bikes data to forecast bike rentals using Prophet.

2.1. EDA. Then partition the data using the 2011 data as a training data. Partition the first 6 months of the 2012 data as a validation data set. Use the last 6 months of 2012 as a test data set.

We will aggregate the data from hourly counts to daily counts:

```
XAgg <- aggregate(X$count, list(X$datetime), sum)
names(XAgg) <- c('ds', 'y')
```

After aggregation, we want to check to see if the data is a valid time series dataset.

Let's check for unique dates:

```
## [1] 456
```

```
## [1] 456
```

There are 456 unique dates for all the 456 observations, indicating there are no repeated dates.

Let's check to see if all dates have the same time spans (in days) between them:

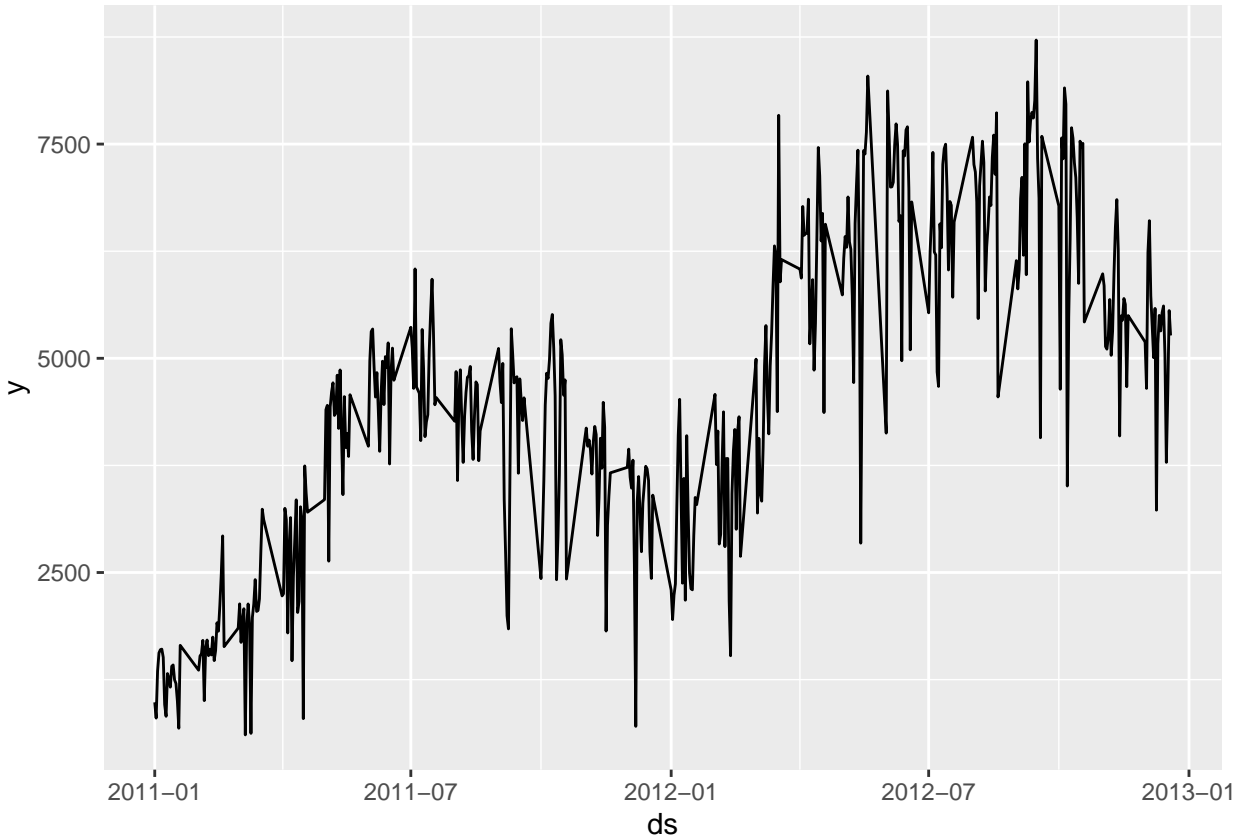
```
## [1] 456
```

```
## difX
##    1  10  11  12  13
## 432   1   1   8  13
```

From the test above we see varying lengths of time spans in the dataset, thus the time series data is not a valid dataset. That being said, Facebook Prophet can still create forecasts using time series data that is not fully validated.

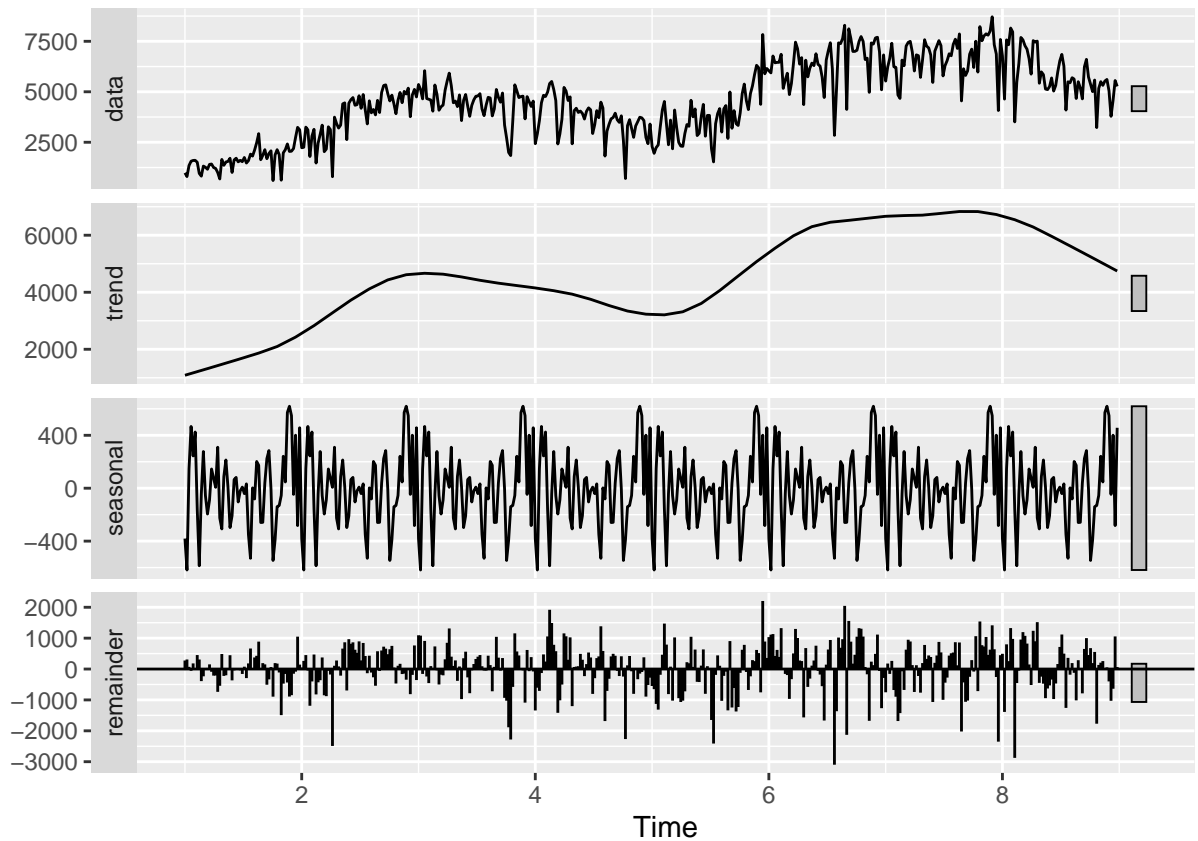
That being said, we will plot and decompose the data.

Plot: Aggregated bike data



We notice higher counts of bike rides during the summer season and lower counts of bike rides during the winter season. Overall though, year-by-year, we notice an increase of bike activity in the second year (2012) compared to the first year (2011).

We will attempt to decompose the data into four seasons of each year:

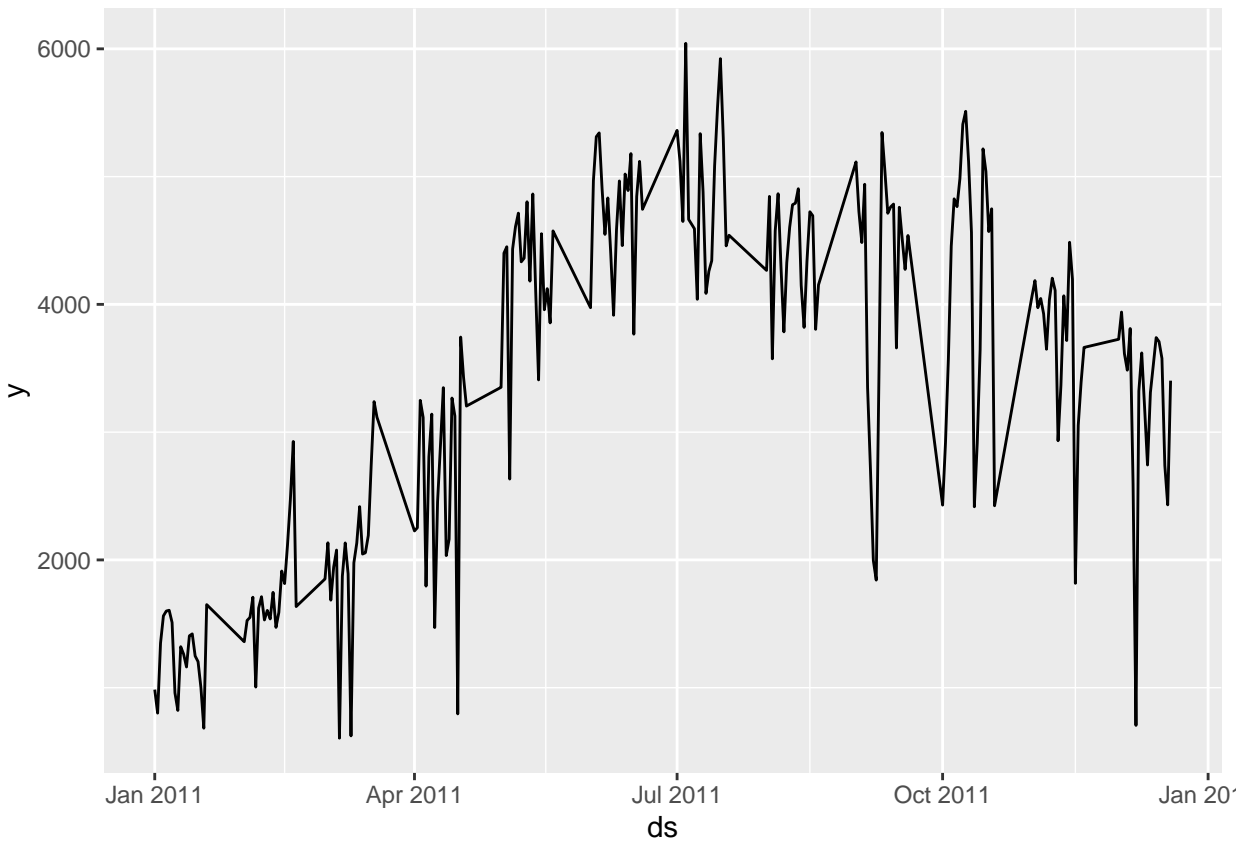


Based on the high frequency of 57 we are able to capture winter spring, summer and fall seasonal activity and get a generalized trend which shoes disctint seasonal activit and overall year-by-year growth.

We will create an aggregated training data set using the first year of the time series data:

```
# Training data, 1 year
A <- X[X$datetime < ymd("2012-01-01"), c(1,12)]
Tr <- aggregate(A$count, list(A$datetime), sum)
names(Tr) <- c('ds', 'y')
```

Plotted training data for the first year:

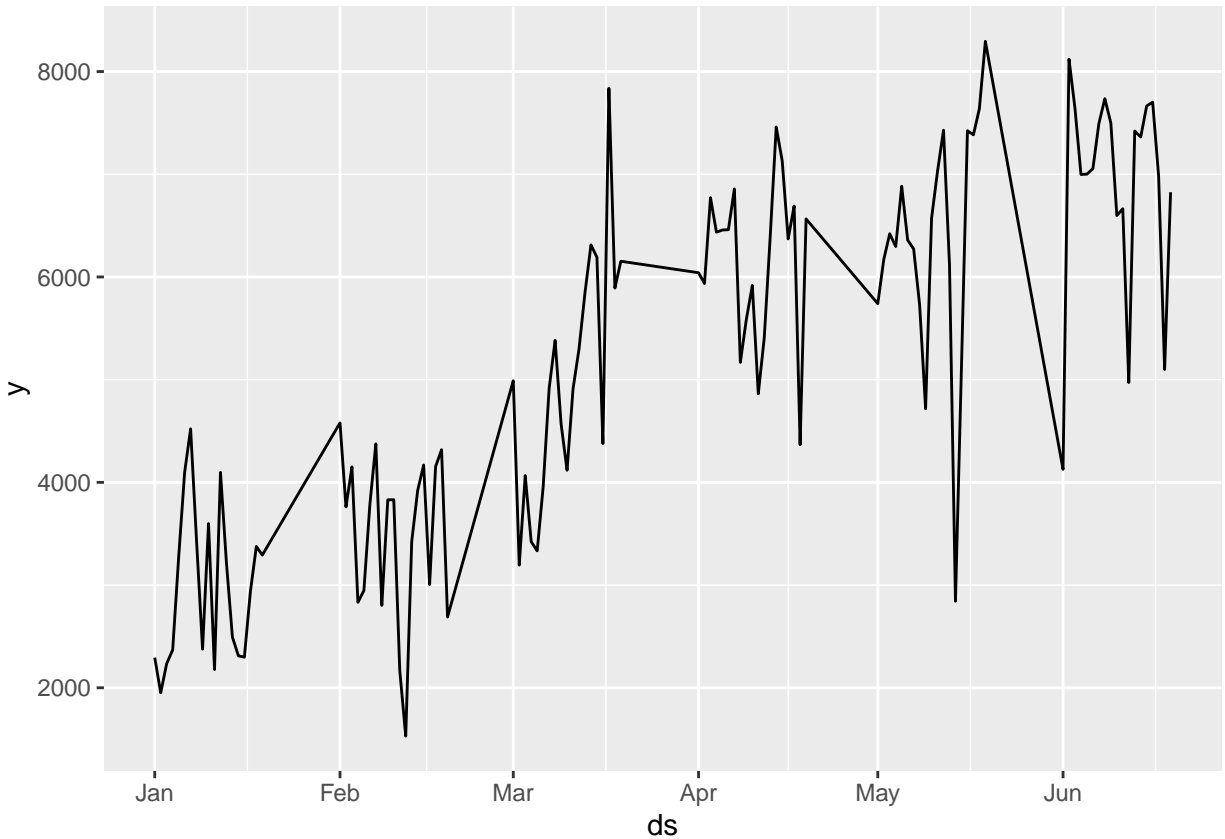


We see a rise in bike activity as the year goes into the warmer months, and a decrease when going into colder months. The end of the year ends higher than the beginning of the year showing overall growth going into the next year.

We will create an aggregated validation data set using the first half of the second year of the time series data:

```
# Validation data, 6 months
A <- X[X$datetime >= ymd("2012-01-01") & X$datetime <= ymd("2012-06-30"), c(1,12)]
V <- aggregate(A$count, list(A$datetime), sum)
names(V) <- c('ds', 'y')
rm(A)
```

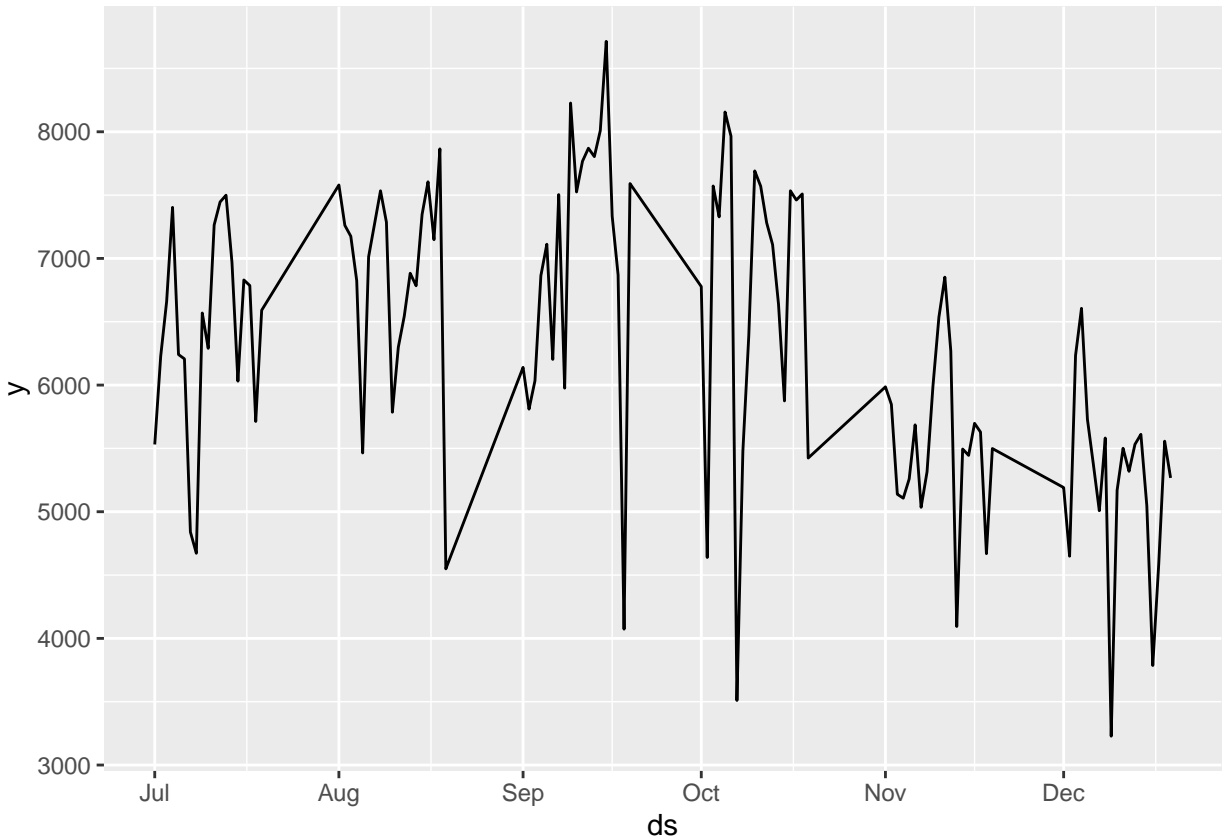
Plot:



From the validation data we see a rise in bike activity as the the first half of 2012 transitions from the winter to the summer.

We will create an aggregated test data set using the second half of the second year of the time series data:

```
# Test data, 6 months
A <- X[X$datetime >= ymd("2012-07-01"), c(1,12)] # X$datetime, X$count
Te <- aggregate(A$count, list(A$datetime), sum) # training set
names(Te) <- c('ds', 'y') # Prophet uses only these names
rm(A)
```



From the testing data we again see a drop in bike activity in the second half of the year as it transitions from summer to winter.

2.2. Define the marked holidays by name.

The original time series data has the following holidays:

```
## # A tibble: 13 x 2
##   ds      holiday
##   <date>   <chr>
## 1 2011-01-17 Martin Luther King
## 2 2011-04-15 Emancipation Day
## 3 2011-07-04 Independence Day
## 4 2011-09-05 Labor Day
## 5 2011-10-10 Columbus Day
## 6 2011-11-11 Veterans Day
## 7 2012-01-02 New Year
## 8 2012-01-16 Martin Luther King
## 9 2012-04-16 Emancipation Day
## 10 2012-07-04 Independence Day
## 11 2012-09-03 Labor Day
## 12 2012-10-08 Columbus Day
## 13 2012-11-12 Veterans Day
```

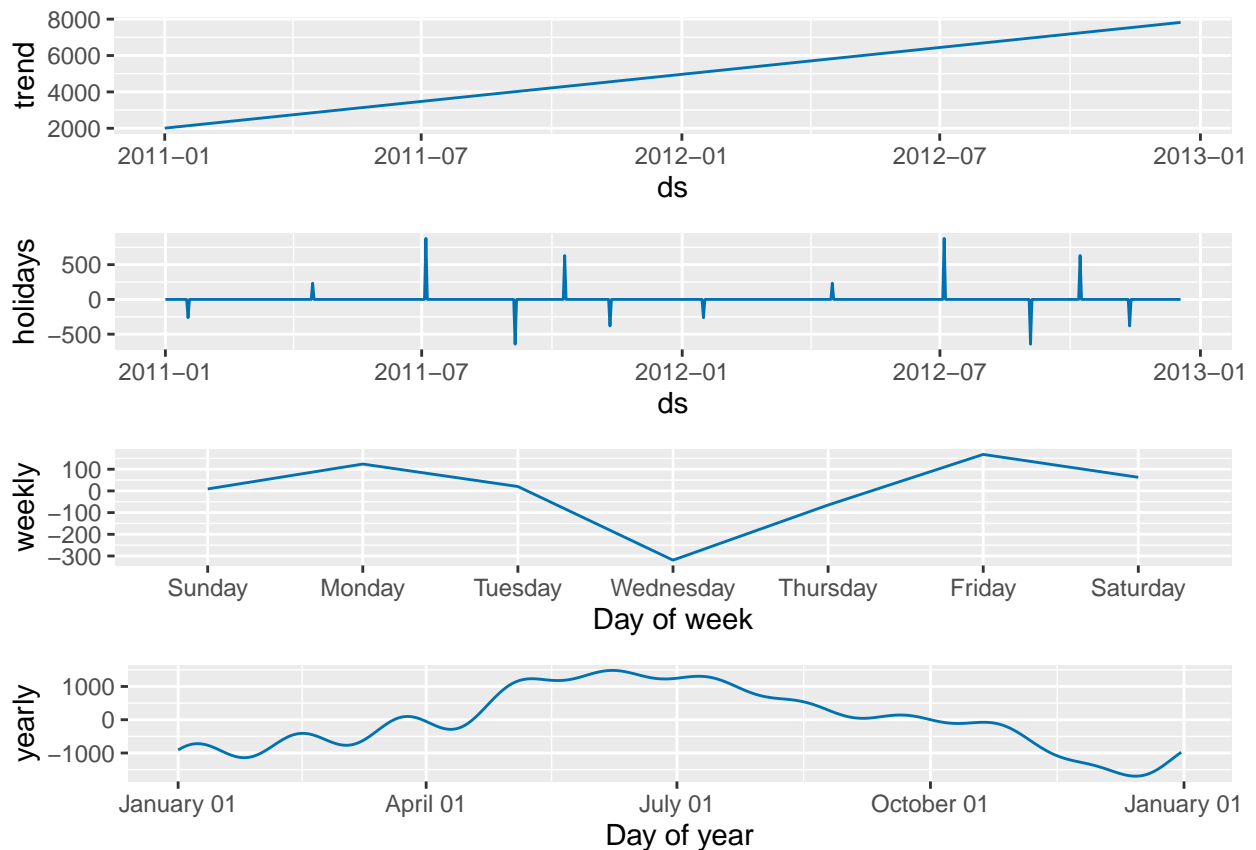

2.3. Construct a Prophet model using the default values for seasonality and changepoints using the training data. Use the holidays as defined above. Describe the model components.

We create the following default Facebook Prophet model only using the holidays defined in section 2.2:

```
m <- prophet(Tr, holidays = h, yearly.seasonality = TRUE)
m1 <- m
```

We have the following components from the default Prophet model:

```
prophet_plot_components(m, Fc)
```



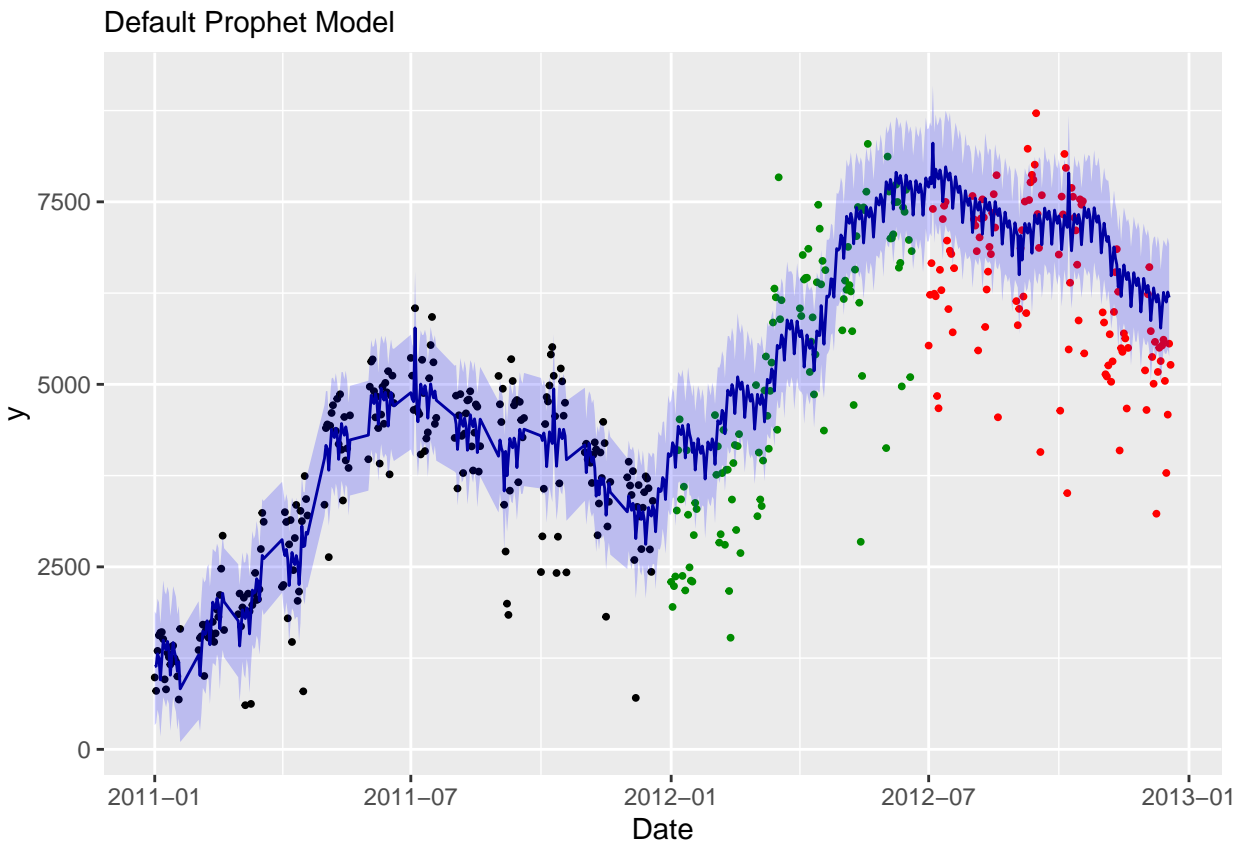
The Prophet model has an overall upward trend component. The holiday component shows Emancipation Day, Independence Day, and Columbus Day influence a positive spike in bike ride counts, while other holidays such as Martin Luther King Day, Labor Day, and Veterans Day have a negative impact on bike rides. Interestingly, New Year's day does not seem to influence bike rides, even though it is included in the list of holidays. The weekly component shows that Wednesdays have the lowest amount of bike rides in the week, while the ends of the work week such as Monday and Friday yield the most bike rides. The yearly component shows that summer days have a more positive impact on bike rides as opposed to days in the winter seasons.

2.4. Forecast the next 6 months from the default model. Plot the training data followed by the forecasted data. Describe the plot.

We predict and plot the bike count activity for 2012 based on the training data which covers 2011:

The default Prophet model predicts a rise in bike counts as it transitions from the colder to warmer seasons, then drops back down as the year gets colder. Overall it anticipates more bike activity in 2012 compared to 2011.

We find that the default Prophet prediction for 2012 looks slightly higher compared to the validation (green dots) and testing (red dots) data that covers 2012, but is able to cover the general trend well, especially with anticipating an overall rise in bike usage for the year.



2.5. Produce forecast accuracy measures based on the validation data, RMSE and MAPE. What do they tell us?

```
##      RMSE      MAPE
## 1292.8868  27.5712
```

The default Prophet model gives us an RMSE score of 1292.8868 and MAPE score of 27.5712, giving us a baseline to compare with tuned models we will develop in section 3.

3. Prophet 2 (25 points) Use the bike data to refine the Prophet model as follows:

3.1. Create a matrix of changepoint flexibility values, seasonal strength values, holiday strength values, set growth as logistic, and set capacity levels. Describe how the column values in the matrix change relative to each other.

We create a matrix below based on various combinations of changepoint, seasonality, holidays, capacity, and logistic growth.

```

##      cps sps hps capacity  growth
## 1  0.05  1  1    7000 logistic
## 2  0.25  1  1    7000 logistic
## 3  0.50  1  1    7000 logistic
## 4  0.80  1  1    7000 logistic
## 5  0.05 10  1    7000 logistic
## 6  0.25 10  1    7000 logistic
## 7  0.50 10  1    7000 logistic
## 8  0.80 10  1    7000 logistic
## 9  0.05  1 10    7000 logistic
## 10 0.25  1 10    7000 logistic
## 11 0.50  1 10    7000 logistic
## 12 0.80  1 10    7000 logistic
## 13 0.05 10 10    7000 logistic
## 14 0.25 10 10    7000 logistic
## 15 0.50 10 10    7000 logistic
## 16 0.80 10 10    7000 logistic
## 17 0.05  1  1    8000 logistic
## 18 0.25  1  1    8000 logistic
## 19 0.50  1  1    8000 logistic
## 20 0.80  1  1    8000 logistic
## 21 0.05 10  1    8000 logistic
## 22 0.25 10  1    8000 logistic
## 23 0.50 10  1    8000 logistic
## 24 0.80 10  1    8000 logistic
## 25 0.05  1 10    8000 logistic
## 26 0.25  1 10    8000 logistic
## 27 0.50  1 10    8000 logistic
## 28 0.80  1 10    8000 logistic
## 29 0.05 10 10    8000 logistic
## 30 0.25 10 10    8000 logistic
## 31 0.50 10 10    8000 logistic
## 32 0.80 10 10    8000 logistic

```

3.2. Using MAPE as the selection criterion, find and list the optimum parameter set from the matrix constructed above.

Using the matrix in the section above, along with the training and validation data, and holidays list, we are able to find the following parameters from the matrix to build a more optimal Prophet model:

```

##      cps sps hps capacity  growth    RMSE    MAPE
## 8  0.8  10  1    7000 logistic 1195.272 24.99066

```

Prophet's suggestion of the optimal parameters above would yield a model with an RMSE of 1195.272 and an MAPE of 24.99066, which are lower values than the default Prophet model.

3.3. Retrain the Prophet model using the training data, the validation data, and the optimum parameters from above. Compare the optimum model components with the default model components.

We develop the following Prophet model using the optimized parameters above along with the training and validation data:

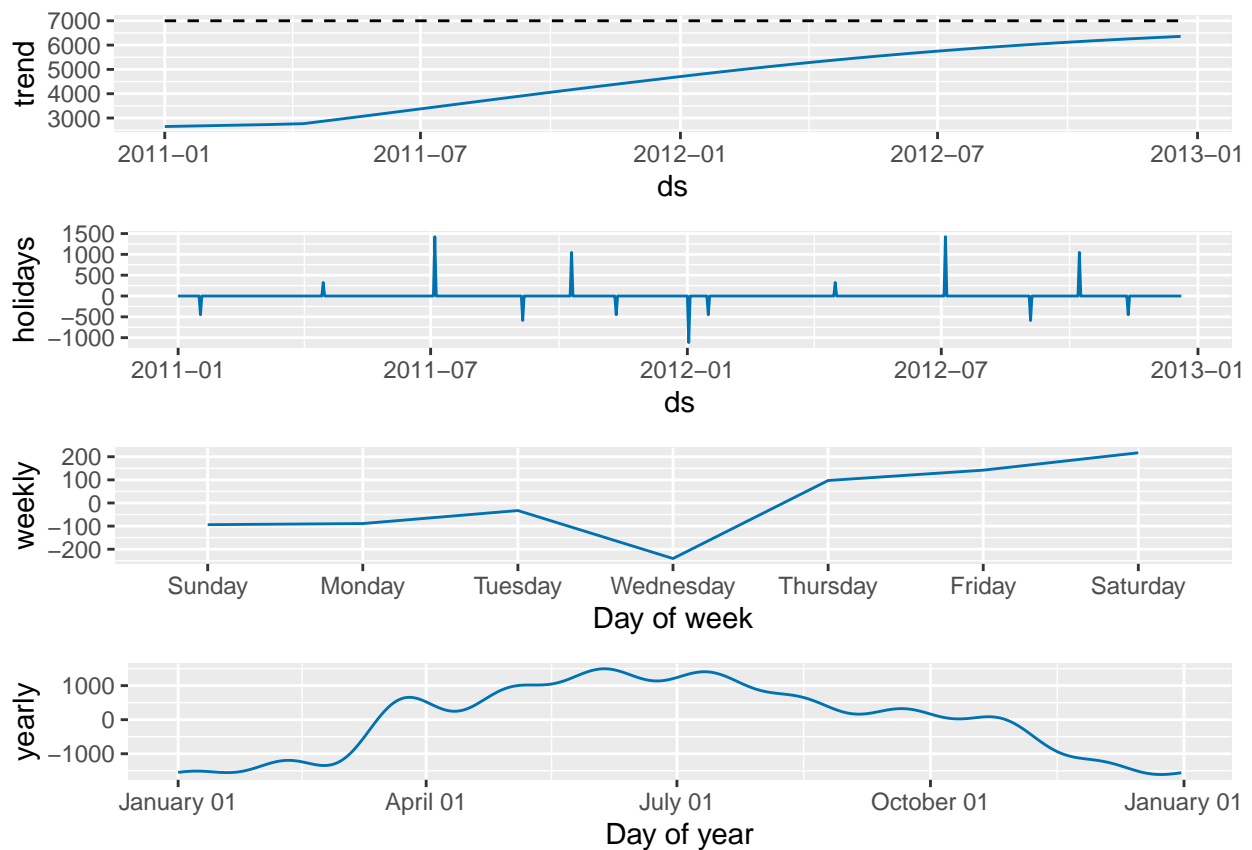
```

reTr <- bind_rows(Tr, V)
reTr$cap <- MAPE.min$capacity

m <- prophet(reTr, growth=MAPE.min$growth, holidays = h,
             seasonality.prior.scale = MAPE.min$sps,
             changepoint.prior.scale = MAPE.min$cps,
             holidays.prior.scale = MAPE.min$hps,
             yearly.seasonality = TRUE)
m2 <- m

```

The new model provides the following components:

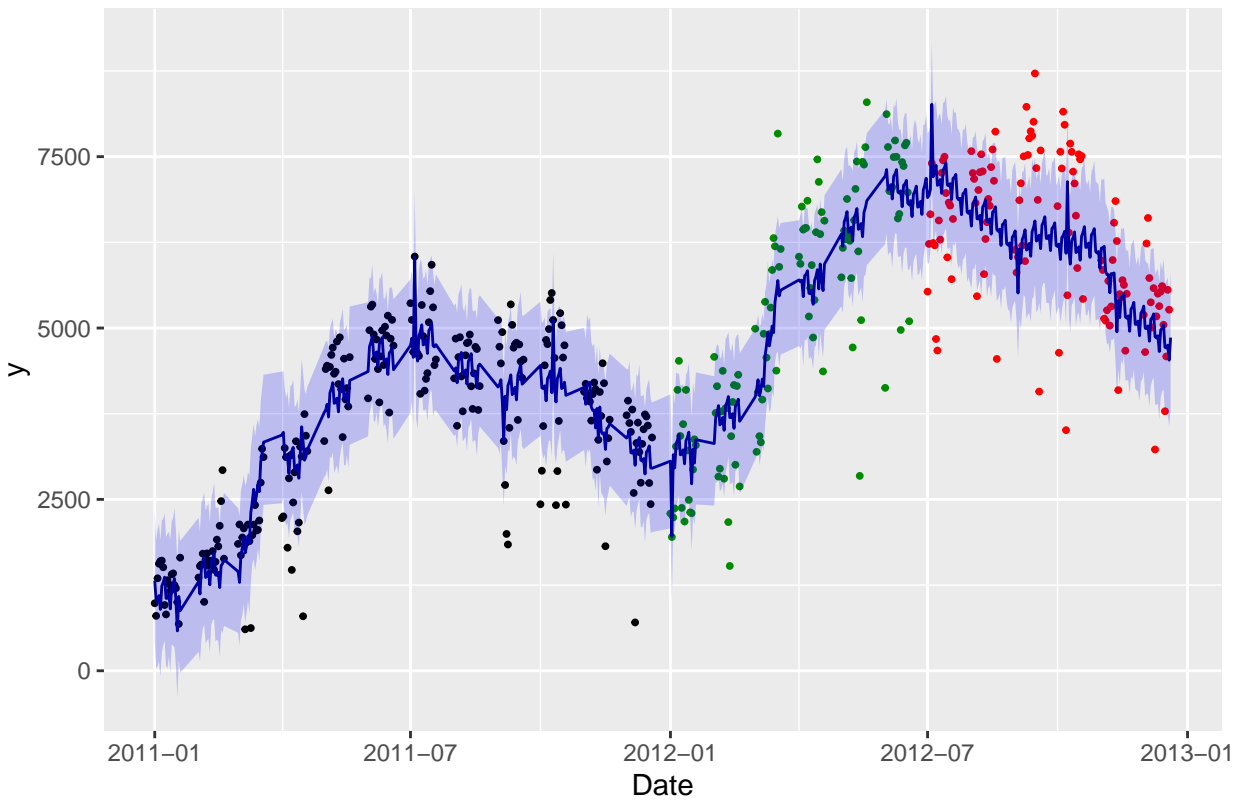


The trend component shows a slight damp in the prediction, with some plateauing into a flat. Holiday impacts are similar to the default model but also denotes a negative spike in bike rides on New Year's day, which wasn't shown on the default model. While Wednesdays in this model also yield the lowest amount of bike rides in the week, it differs such that the second half of the week yields more bike rides than the first half. The yearly component is very similar to the default model, where summer days positively impact rides while winter days have a negative impact.

3.4. Forecast the next 6 months from the optimum model. Plot the data followed by the forecasted data. Describe the plot and compare it to the default model plot.

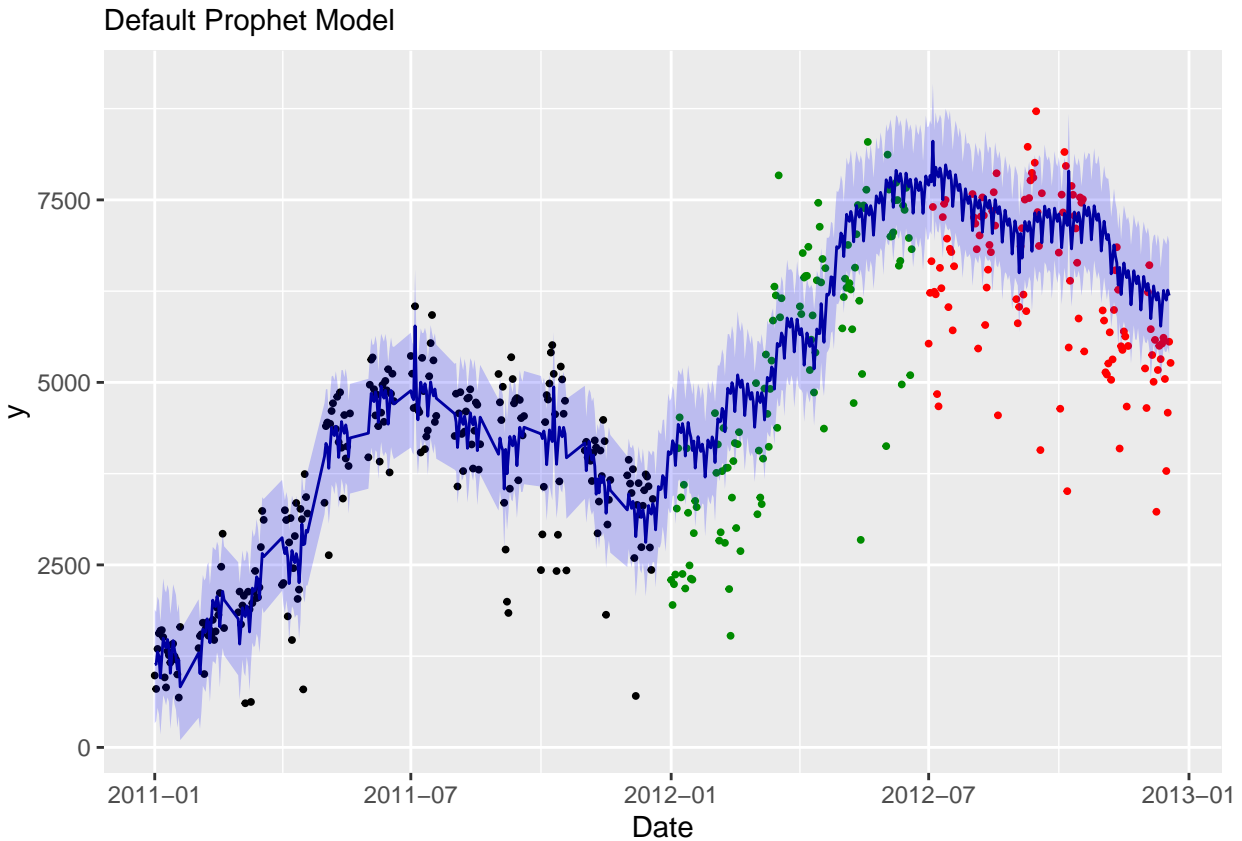
We create the following plot for the 2nd half of 2012 using the optimized Prophet model fitted with the training (black dots) and validation (green dots) data, with the test data (red dots) shown:

Optimized Prophet Model



The predicted data from 2012-07 to 2012-12 lines up quite well with the test data, accounting for slight rise in activity at the beginning of the fall season and the drop as it heads into the colder end of the year.

Comparing with the default Prophet model:



We find the default model to slightly overestimate its prediction. That being said, the default model was fitted using the training data which only covers 2011.

3.5. Produce, describe, and compare forecast accuracy measures based on the test data, RMSE and MAPE with those of the default model.

RMSE and MAPE: Optimized Prophet Model

```
##      RMSE      MAPE
## 1040.97173  14.10718
```

RMSE and MAPE: Default Prophet Model

```
##      RMSE      MAPE
## 1292.8868  27.5712
```

We find the optimized Prophet model to have notably lower RMSE (1040.97173) and MAPE (14.10718) scores compared to the default model, quantitatively making the optimized model the better-fitting one.