Class 4 - Time Series Analysis

ARIMA

Assumption :

- Stationary Data : mean and variance does not change as a function of time
- Differencing technique:

$$y/_diff/_trend = y_{(t+1)} - y_t$$

$$y/_diff/_seasonal = y_{(t+lag)} - y_t$$

1. Load Dataset

```
cycle = read.csv("G:/My Drive/Master-Data-Science/Semester_1/Business_Analytics/Data/Ch6_ridership_data_
head(cycle)
```

```
##              datetime count
## 1 2011-01-01 00:00:00    16
## 2 2011-01-01 01:00:00    40
## 3 2011-01-01 02:00:00    32
## 4 2011-01-01 03:00:00    13
## 5 2011-01-01 04:00:00     1
## 6 2011-01-01 05:00:00     1
```

2. Aggregate by month

```
monthly_ride = as.data.frame(cycle %>%
                              group_by(year = year(datetime), month = month(datetime)) %>%
                              summarise(riders = sum(count)))
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
head(monthly_ride)
```

```
##   year month riders
## 1 2011     1  37727
## 2 2011     2  46396
## 3 2011     3  65109
## 4 2011     4  90332
## 5 2011     5 132580
## 6 2011     6 139674
```

```r
table(monthly_ride$year, monthly_ride$month)
```

```
##
##        1  2  3  4  5  6  7  8  9  10  11  12
##   2011 1  1  1  1  1  1  1  1  1   1   1   1
##   2012 1  1  1  1  1  1  1  1  1   1   1   1
```

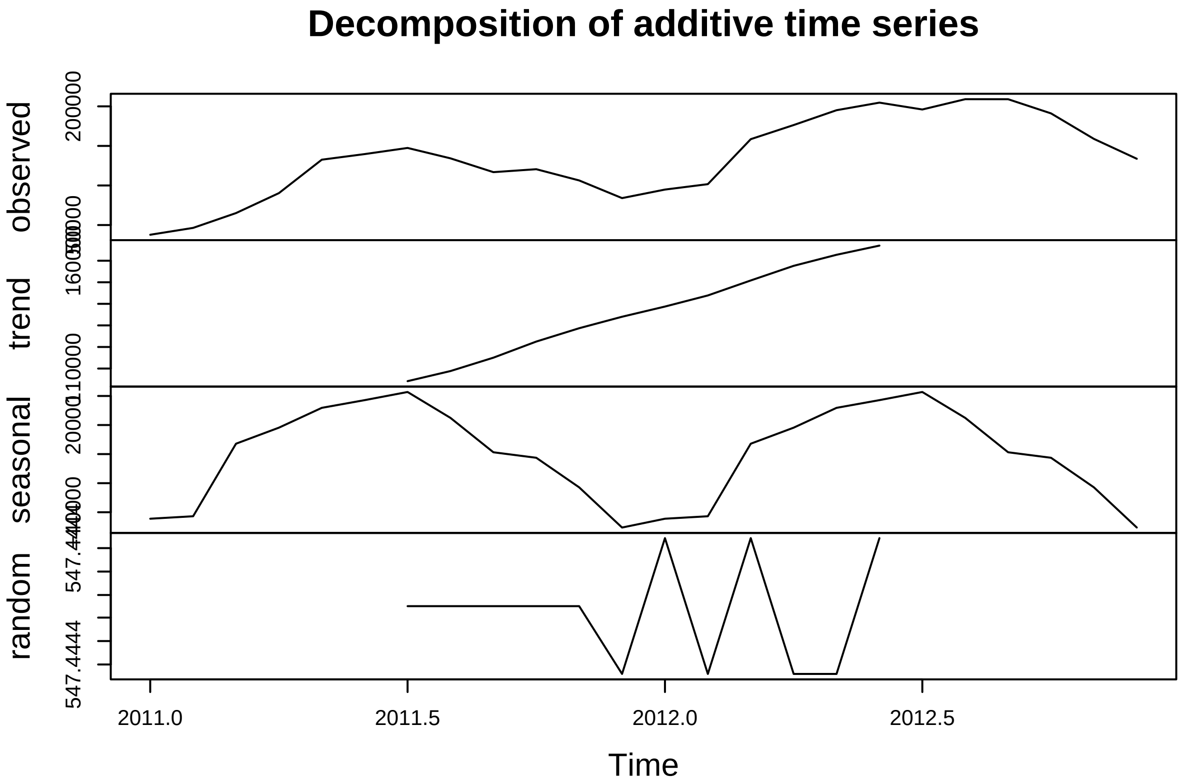
3. Convert to TS data

```r
riders = monthly_ride[,3]
monthly = ts(data = riders, start = c(2011,1), end = c(2012,12), frequency = 12)
monthly
```

```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2011  37727  46396  65109  90332 132580 139674 147426 134280 116825 120535
## 2012  94832 101668 158535 176349 195114 204683 196014 209024 208995 191108
##         Nov    Dec
## 2011 106361  84025
## 2012 158855 133735
```

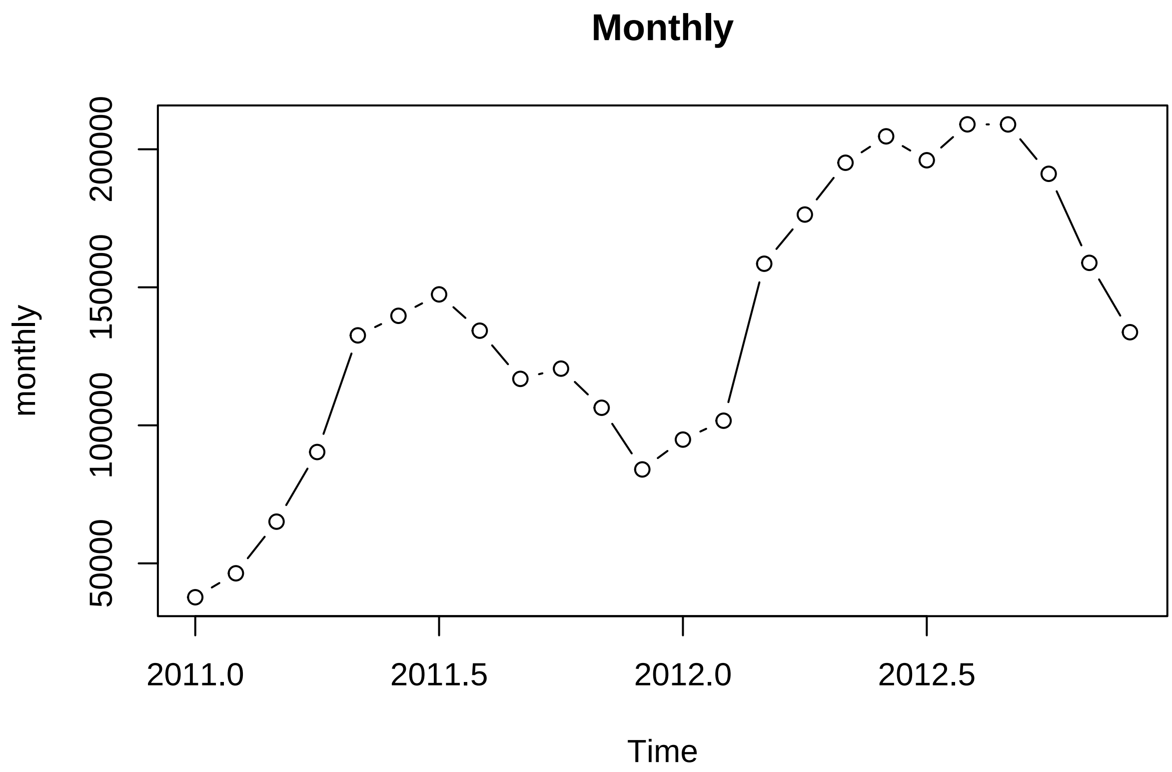
4. Decompose the data

```r
plot(decompose(monthly))
```

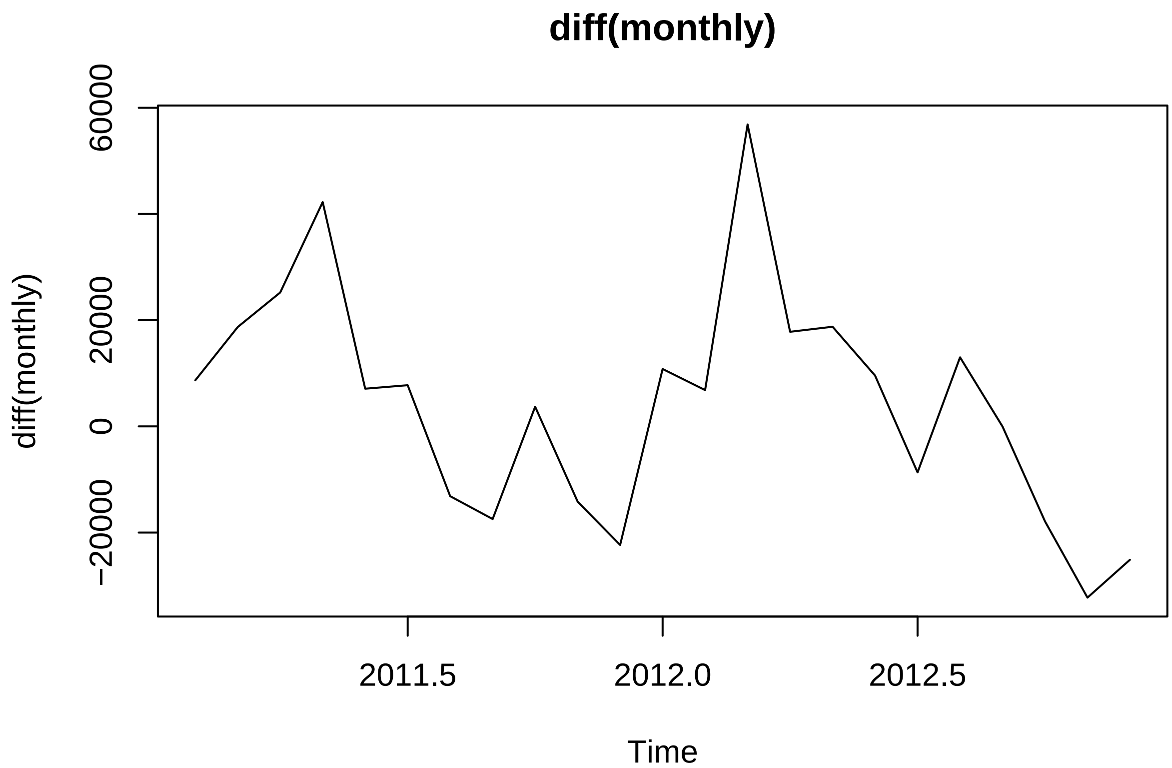


5. Difference the data (if needed)

```
plot(monthly , type = 'b', main='Monthly')
```

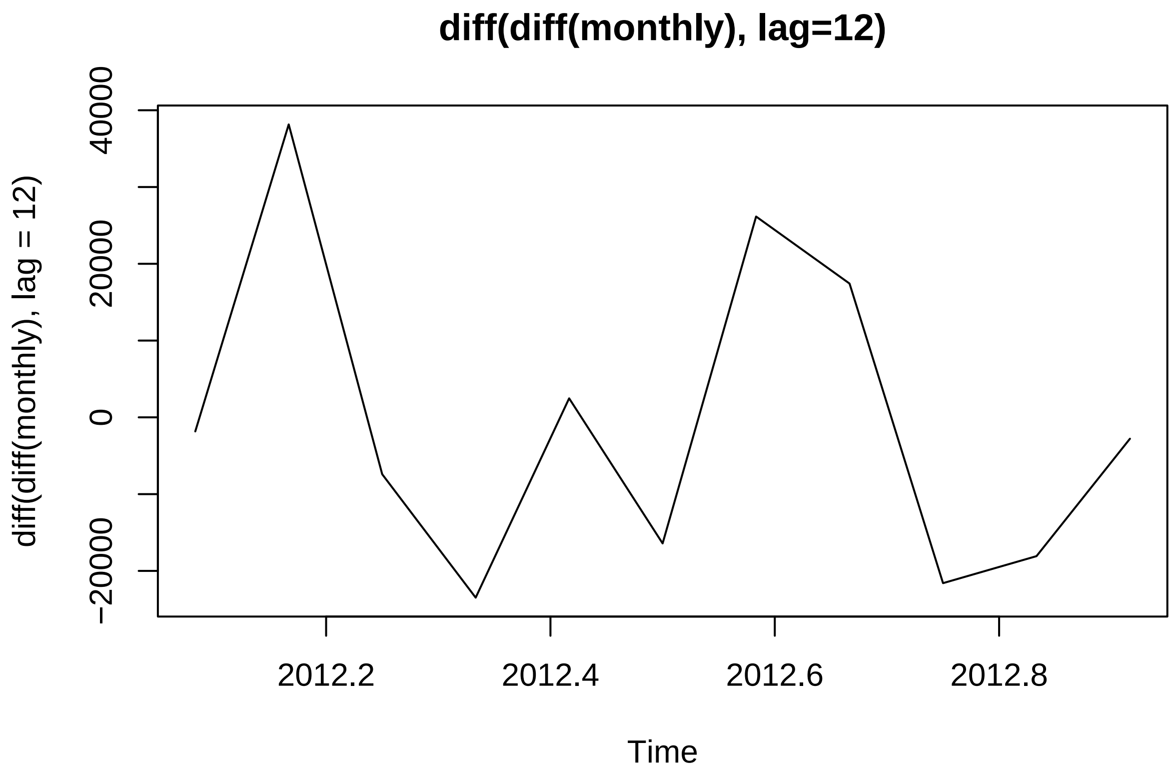



```r
plot(diff(monthly), main = 'diff(monthly)')
```




```r
# Remove trend pattern in data, remain seasonality+constant+randomness
```

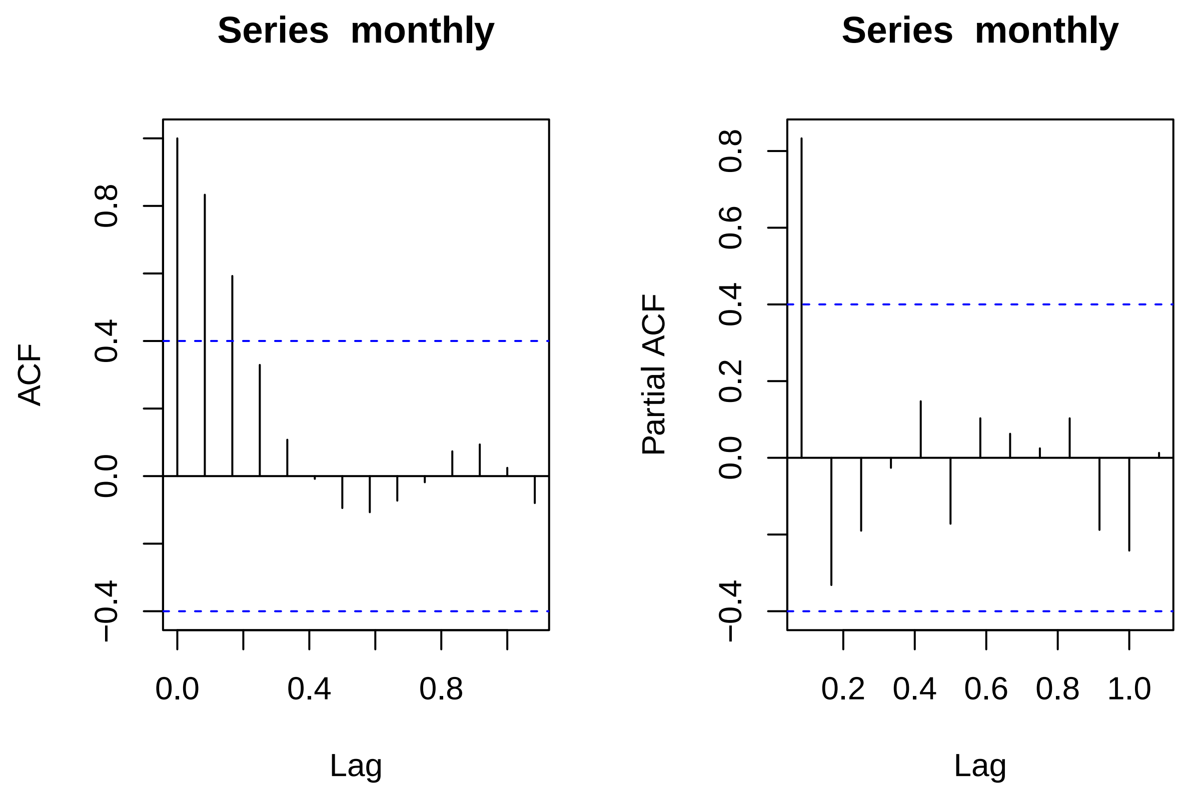
```r
plot(diff(diff(monthly), lag=12), main='diff(diff(monthly), lag=12)')
```




```
# Remoce trend and seasonal pattern in data, remain constant+randomness
```

6. Generate ACF and PACF

```
par(mfrow=c(1,2))
acf(monthly)
pacf(monthly)
```




```r
par(mfrow=c(1,1))
```

The values have to be whole numbers, therefore inadequate data to use for prediction. Possible values:

- p : 0,1,2
- d : 0,1
- q : 0,1

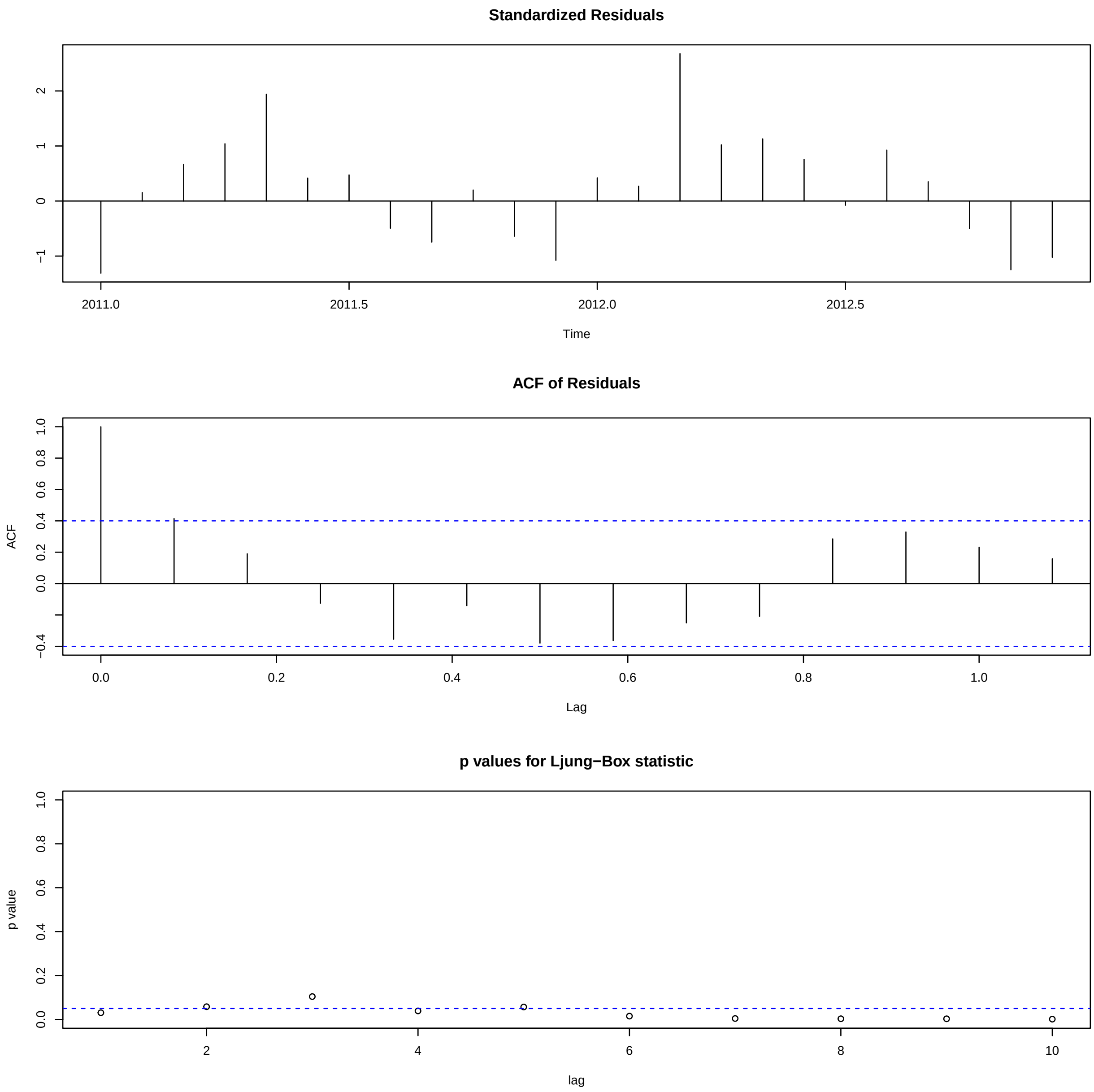
7. Run ARIMA Model

```r
arima100_000 = arima(monthly, c(1,0,0),
                    seasonal = list(order = c(0,0,0)))
arima100_000
```

```
##
## Call:
## arima(x = monthly, order = c(1, 0, 0), seasonal = list(order = c(0, 0, 0)))
##
## Coefficients:
##          ar1  intercept
##       0.9252  110276.60
```

```
## s.e.   0.0667   41914.62
##
## sigma^2 estimated as 440733047:  log likelihood = -273.87,  aic = 553.74
```

```r
tsdiag(arima100_000)
```

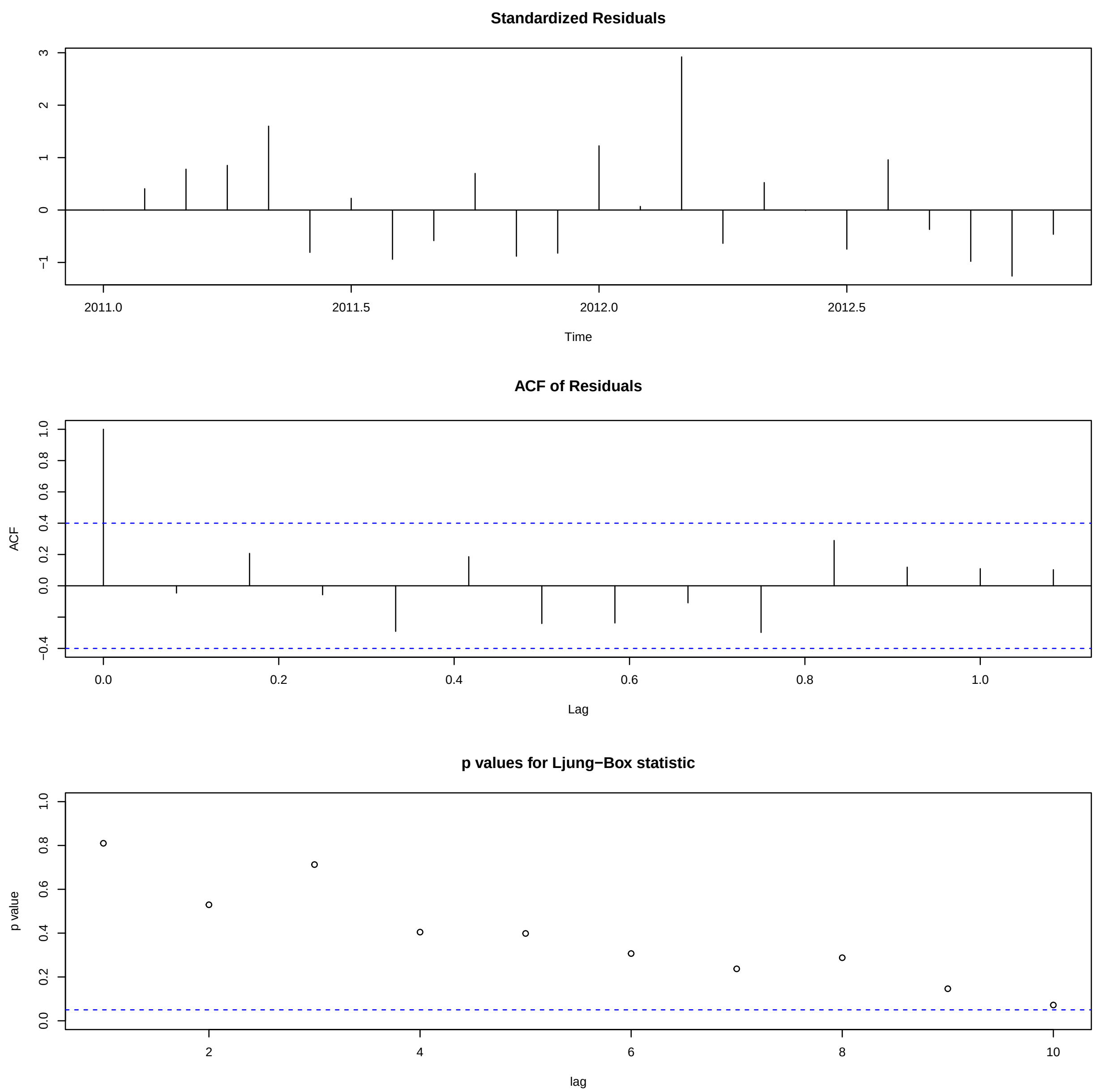



```r
arima110_000 = arima(monthly, c(1,1,0),
                     seasonal = list(order = c(0,0,0)))
arima110_000
```

```
##
## Call:
## arima(x = monthly, order = c(1, 1, 0), seasonal = list(order = c(0, 0, 0)))
```

```
##
## Coefficients:
##          ar1
##       0.5171
## s.e.  0.1772
##
## sigma^2 estimated as 333435859:  log likelihood = -258.48,  aic = 520.96
```

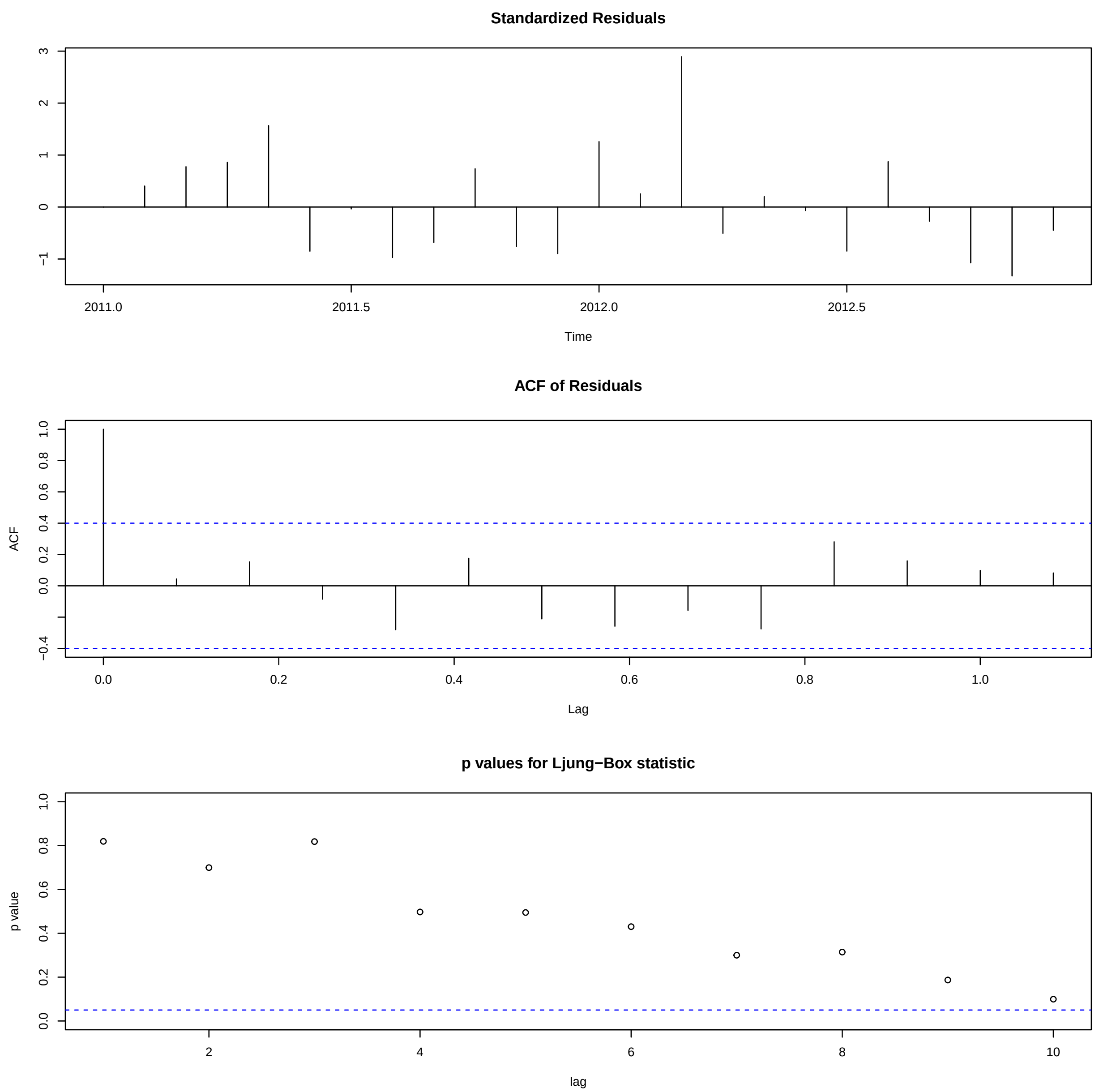
```
tsdiag(arima110_000)
```




```
arima210_000 = arima(monthly, c(2,1,0),
                     seasonal = list(order = c(0,0,0)))
arima210_000
```

```
##
## Call:
## arima(x = monthly, order = c(2, 1, 0), seasonal = list(order = c(0, 0, 0)))
##
## Coefficients:
##          ar1     ar2
##       0.4597  0.1219
## s.e.  0.2014  0.2127
##
## sigma^2 estimated as 328207179:  log likelihood = -258.32,  aic = 522.63
```

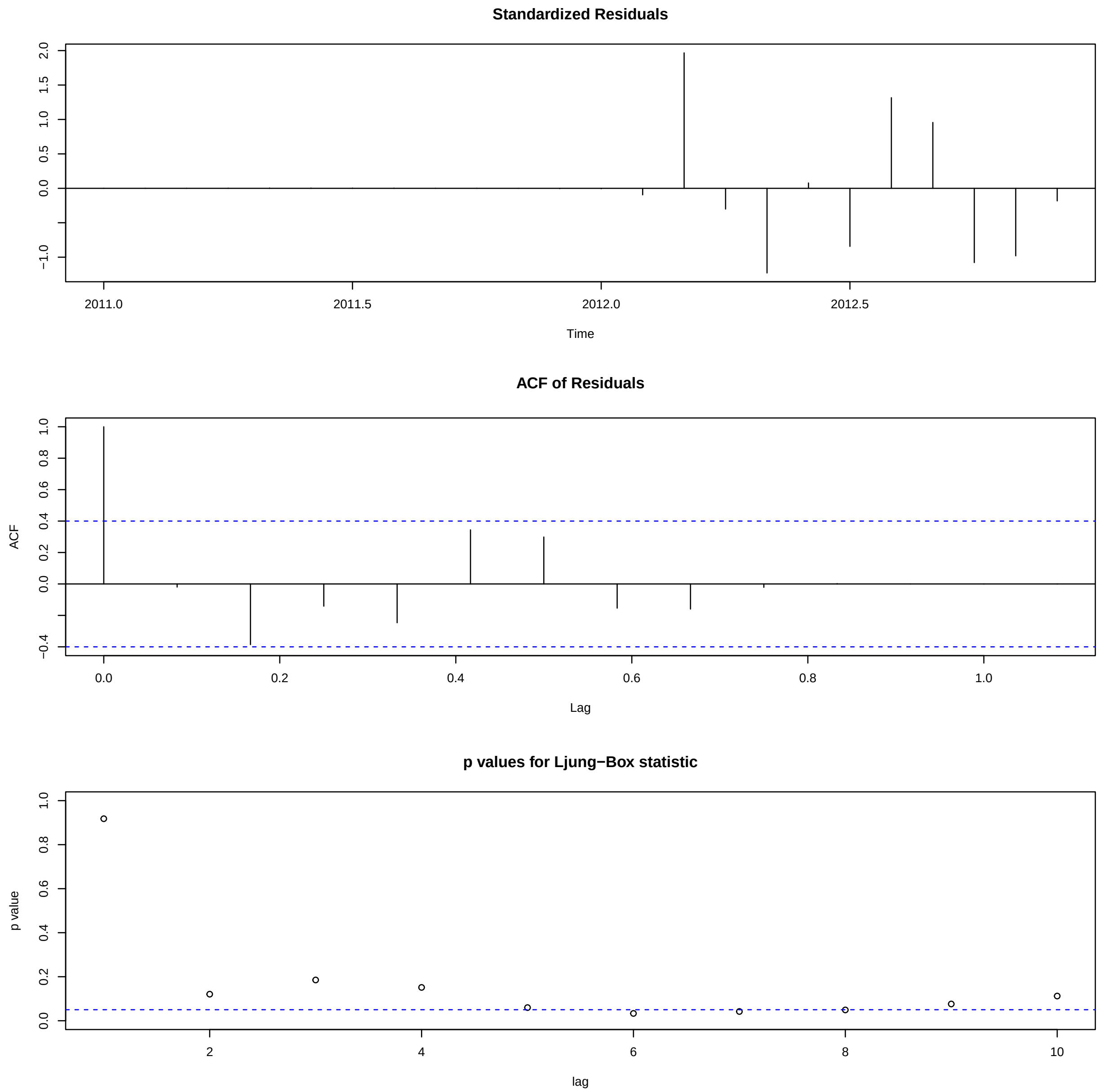
```
tsdiag(arima210_000)
```



```r
arima110_010 = arima(monthly, c(1,1,0),
                     seasonal = list(order = c(0,1,0)))
arima110_010
```

```
##
## Call:
## arima(x = monthly, order = c(1, 1, 0), seasonal = list(order = c(0, 1, 0)))
##
## Coefficients:
##           ar1
##       -0.0415
## s.e.   0.2887
##
## sigma^2 estimated as 374694447:  log likelihood = -124.19,  aic = 252.38
```

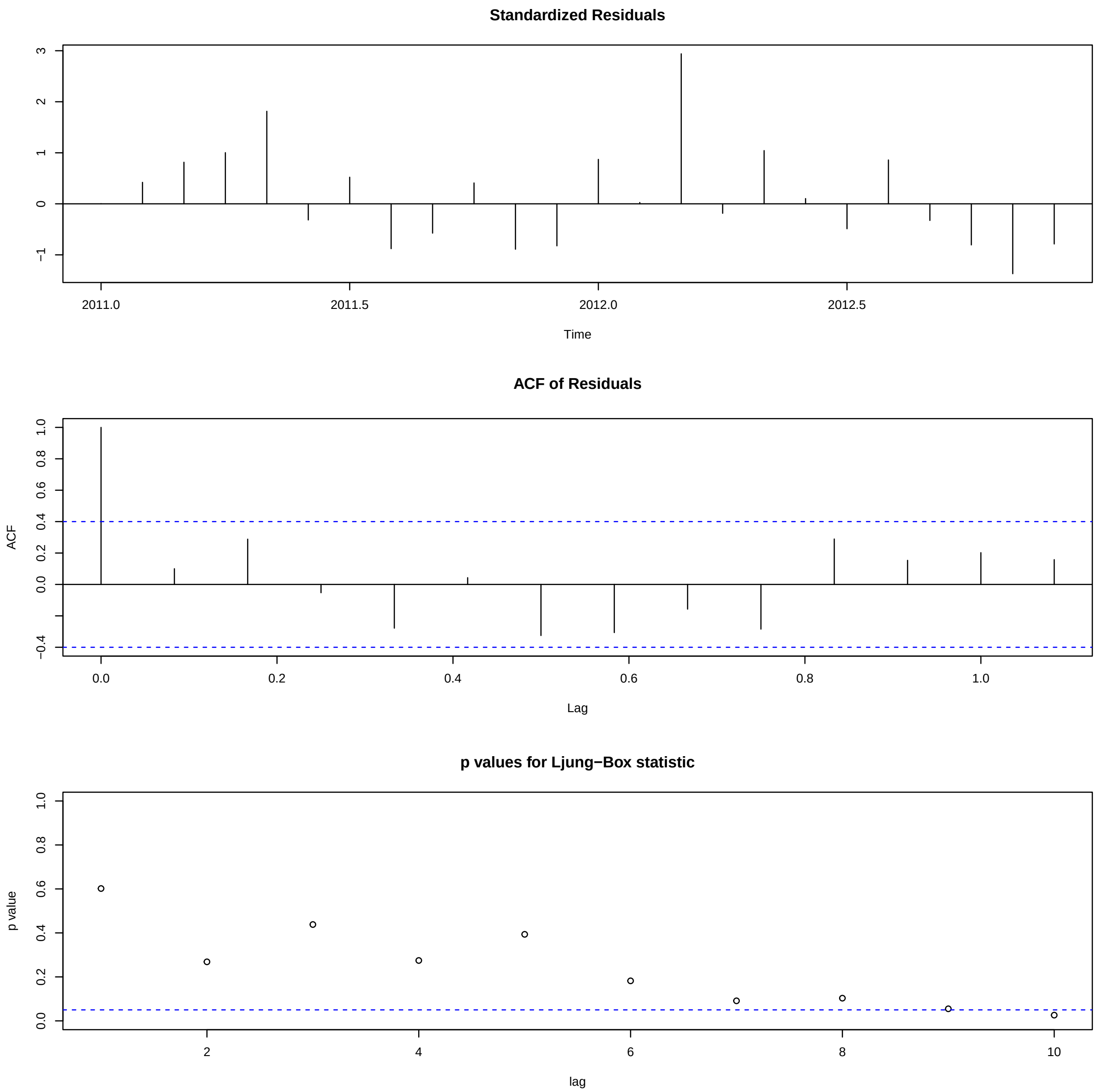
```r
tsdiag(arima110_010)
```




```
arima011_000 = arima(monthly, c(0,1,1),
                     seasonal = list(order = c(0,0,0)))
arima011_000
```

```
##
## Call:
## arima(x = monthly, order = c(0, 1, 1), seasonal = list(order = c(0, 0, 0)))
##
## Coefficients:
##          ma1
##       0.3772
## s.e.  0.1646
##
## sigma^2 estimated as 371716295:  log likelihood = -259.65,  aic = 523.3
```

```r
tsdiag(arima011_000)
```




```r
data.frame(
  Model = c("ARIMA(1,0,0)(0,0,0)",
            "ARIMA(1,1,0)(0,0,0)",
            "ARIMA(2,1,0)(0,0,0)",
            "ARIMA(1,1,0)(0,1,0)",
            "ARIMA(0,1,1)(0,0,0)"),
  AIC = c(arima100_000$aic,
          arima110_000$aic,
          arima210_000$aic,
          arima110_010$aic,
          arima011_000$aic),
```

```
  Diagnostics = c("Poor residuals and poor p-values",
                  "Good diagnostics",
                  "Good diagnostics",
                  "Poor p-values",
                  "Poor p-values")
)
```

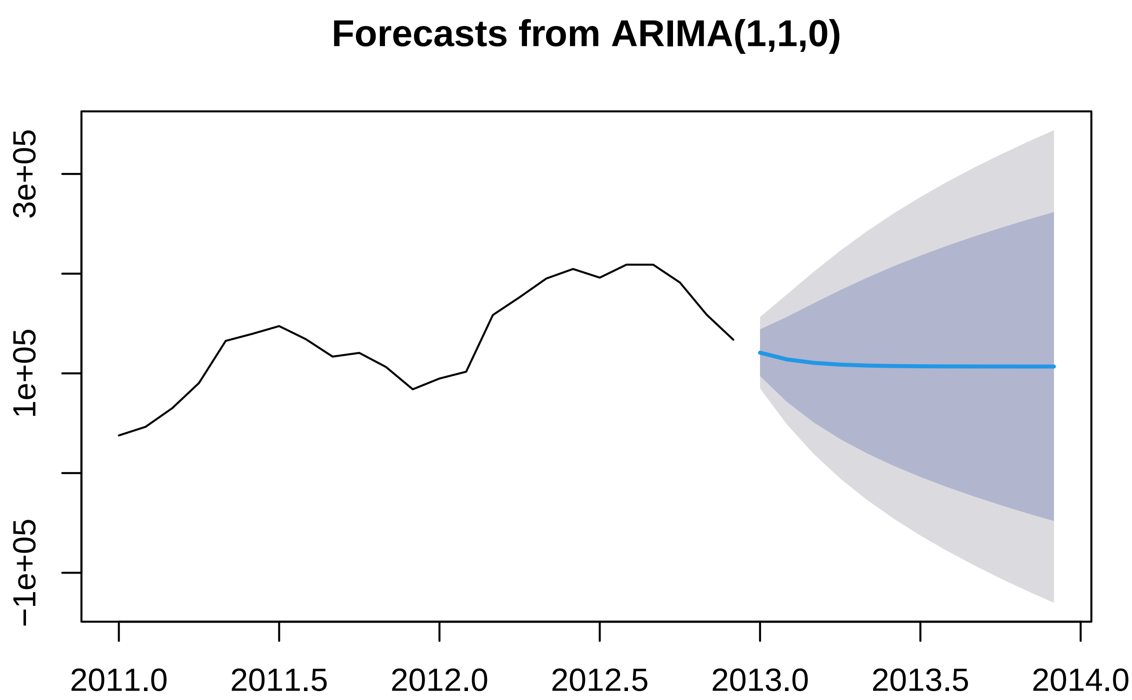
```
##                  Model      AIC                      Diagnostics
## 1 ARIMA(1,0,0)(0,0,0) 553.7413 Poor residuals and poor p-values
## 2 ARIMA(1,1,0)(0,0,0) 520.9564                 Good diagnostics
## 3 ARIMA(2,1,0)(0,0,0) 522.6320                 Good diagnostics
## 4 ARIMA(1,1,0)(0,1,0) 252.3762                    Poor p-values
## 5 ARIMA(0,1,1)(0,0,0) 523.2984                    Poor p-values
```

8. Run forecasting

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

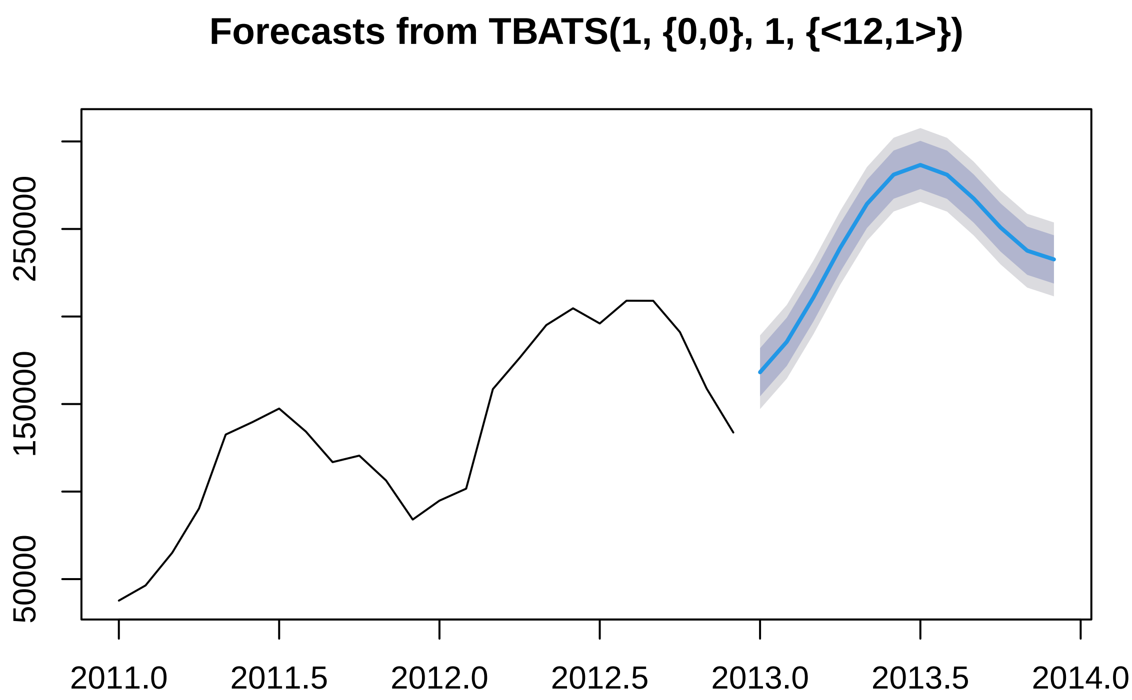
```
yr_forecast = forecast(arima110_000, h=12)
plot(yr_forecast)
```



TBATS

```r
monthly_data = tbats(monthly)

year_forecast = forecast(monthly_data, h=12)
plot(year_forecast)
```



Look at data

```r
summary(year_forecast$mean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  168164  227220  245052  242105  270736  286574
```

```r
summary(year_forecast$upper)
```

```
##        80%                95%
##  Min.   :181925   Min.   :189210
##  1st Qu.:241006   1st Qu.:248303
##  Median :258830   Median :266123
##  Mean   :255881   Mean   :263174
##  3rd Qu.:284519   3rd Qu.:291815
##  Max.   :300352   Max.   :307646
```

```r
summary(year_forecast$lower)
```

```
##       80%              95%
##  Min.   :154402   Min.   :147118
##  1st Qu.:213435   1st Qu.:206138
##  Median :231274   Median :223980
##  Mean   :228328   Mean   :221035
##  3rd Qu.:256953   3rd Qu.:249657
##  Max.   :272796   Max.   :265503
```

Create annotations

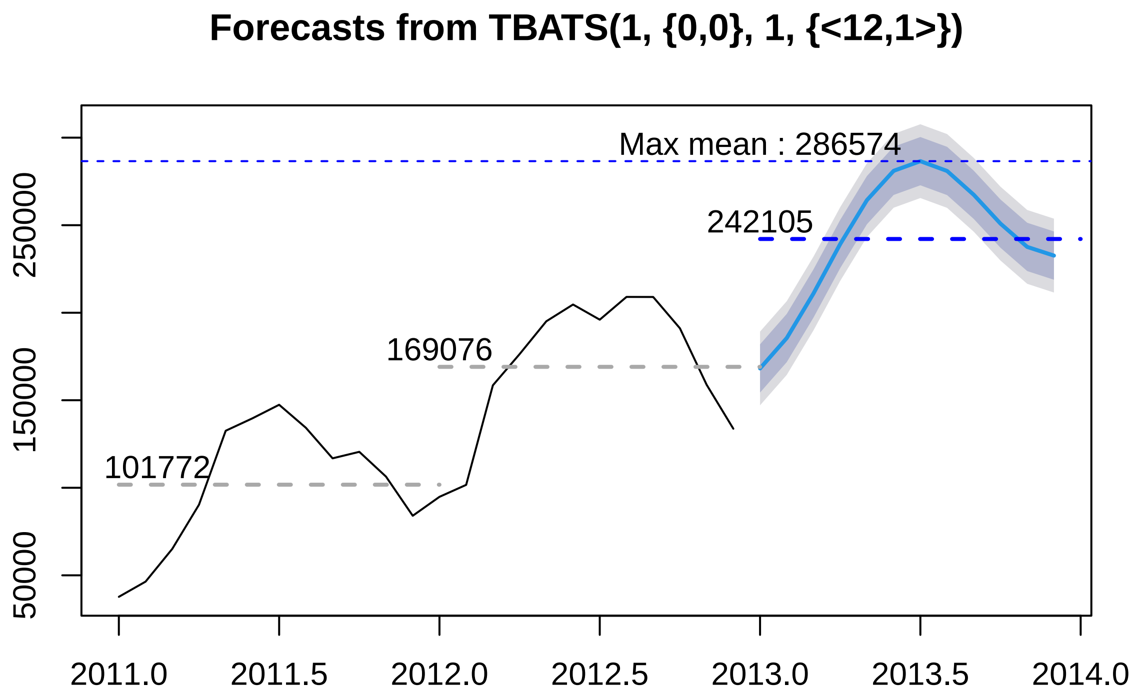
```r
mean_2011 = round(as.numeric(filter(monthly_ride, year == 2011) %>%
                               summarise(mean = mean(riders))),0)
mean_2012 = round(as.numeric(filter(monthly_ride, year == 2012) %>%
                               summarise(mean = mean(riders))),0)
mean_2013 = round(mean(year_forecast$mean))
max_mean_2013 = round(max(year_forecast$mean))
```

Create complete graph

```r
# Main plot
plot(year_forecast)

# line segments
abline(h = max(year_forecast$mean), lty=2, col='blue')
segments(2011, mean_2011, x1 = 2012, y1 = mean_2011, col = 'darkgrey', lty = 2, lwd = 2)
segments(2012, mean_2012, x1=2013, y1=mean_2012, col='darkgrey', lty=2,lwd=2)
segments(2013, mean_2013, x1=2014, y1=mean_2013, col='blue', lty=2, lwd=2)

# Annotation
text(2011.12, mean_2011 + 10000, mean_2011)
text(2012, mean_2012 + 10000, mean_2012)
text(2013, mean_2013 + 10000, mean_2013)
text(2013, max_mean_2013 + 10000, paste("Max mean :", max_mean_2013))
```



Forecasts from TBATS(1, {0,0}, 1, {<12,1>})
Max mean : 286574
242105
169076
101772
50000
150000
250000
2011.0
2011.5
2012.0
2012.5
2013.0
2013.5
2014.0