Time Series Analysis of Capital Bikeshare Users - Xuhui Ying

Load Libraries

```
library(tidyverse)
library(tidymodels)
library(janitor)
library(skimr)
library(kableExtra)
library(GGally)
library(vip)
library(fastshap)
library(MASS)
library(ISLR)
library(tree)
library(ggplot2)
library(dplyr)
library(lubridate)
library(imputeTS)
library(forecast)
library(urca)
library(pracma)
library(astsa)
library(fpp2)
```

Load Data

Import your data with read_csv()

```
bike_data <- read_csv("bikedata_2012.csv") %>% clean_names()

head(bike_data)
```

date <chr>	y… <dbl>	mo… <dbl>	sea… <dbl>	h… <dbl>	holiday <dbl>	day_of_the_week <dbl>	working_day <dbl>	weather_type <dbl>	te
1/1/2011	2011	1	4	0	0	6	0	1	
1/1/2011	2011	1	4	1	0	6	0	1	
1/1/2011	2011	1	4	2	0	6	0	1	
1/1/2011	2011	1	4	3	0	6	0	1	
1/1/2011	2011	1	4	4	0	6	0	1	
1/1/2011	2011	1	4	5	0	6	0	2	

6 rows | 1-10 of 16 columns

```
tail(bike_data)
```

date <chr>	y… <dbl>	mo… <dbl>	sea… <dbl>	h… <dbl>	holiday <dbl>	day_of_the_week <dbl>	working_day <dbl>	weather_type <dbl>
12/31/2012	2012	12	4	18	0	1	1	2
12/31/2012	2012	12	4	19	0	1	1	2
12/31/2012	2012	12	4	20	0	1	1	2
12/31/2012	2012	12	4	21	0	1	1	1
12/31/2012	2012	12	4	22	0	1	1	1
12/31/2012	2012	12	4	23	0	1	1	1

6 rows | 1-10 of 16 columns

```
skim(bike_data)
```

Data summary

Name	bike_data
Number of rows	17379
Number of columns	16
_____	
Column type frequency:	
character	1
numeric	15
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
date	0	1	8	10	0	731	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
year	0	1	2011.50	0.50	2011.0	2011.0	2012.0	2012.0	2012.0	▇▁▁▁▇
month	0	1	6.54	3.44	1.0	4.0	7.0	10.0	12.0	▆▆▃▅▇
season	0	1	2.49	1.12	1.0	1.0	2.0	3.0	4.0	▇▂▂▂▇
hour	0	1	11.55	6.91	0.0	6.0	12.0	18.0	23.0	▇▆▆▆▇
holiday	0	1	0.03	0.17	0.0	0.0	0.0	0.0	1.0	▇▁▁▁▁
day_of_the_week	0	1	3.00	2.01	0.0	1.0	3.0	5.0	6.0	▇▃▃▃▇
working_day	0	1	0.68	0.47	0.0	0.0	1.0	1.0	1.0	▃▁▁▁▇
weather_type	0	1	1.43	0.64	1.0	1.0	1.0	2.0	4.0	▇▃▁▁▁
temperature_f	0	1	58.78	16.62	17.6	45.2	59.0	72.8	102.2	▂▆▇▅▁

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
temperature_feels_f	0	1	59.72	20.42	3.2	42.8	60.8	77.0	122.0	▁▆▇▁
humidity	0	1	62.72	19.29	0.0	48.0	63.0	78.0	100.0	▁▅▇▁
wind_speed	0	1	12.74	8.20	0.0	7.0	13.0	17.0	57.0	▇▇▂▁
casual_users	0	1	35.68	49.31	0.0	4.0	17.0	48.0	367.0	▇▁▁▁
registered_users	0	1	153.79	151.36	0.0	34.0	115.0	220.0	886.0	▇▃▁▁
total_users	0	1	189.46	181.39	1.0	40.0	142.0	281.0	977.0	▇▃▁▁

Create time series object and plot time series

```
bike_month <- bike_data %>%
  group_by(year, month) %>%
  summarize(total_users = sum(total_users))

bike_month
```

year <dbl>	month <dbl>	total_users <dbl>
2011	1	38189
2011	2	48215
2011	3	64045
2011	4	94870
2011	5	135821
2011	6	143512
2011	7	141341
2011	8	136691
2011	9	127418
2011	10	123511

1-10 of 24 rows Previous **1** 2 3 Next

```
bike_month <- subset(bike_month, select=-c(year, month))
bike_month
```

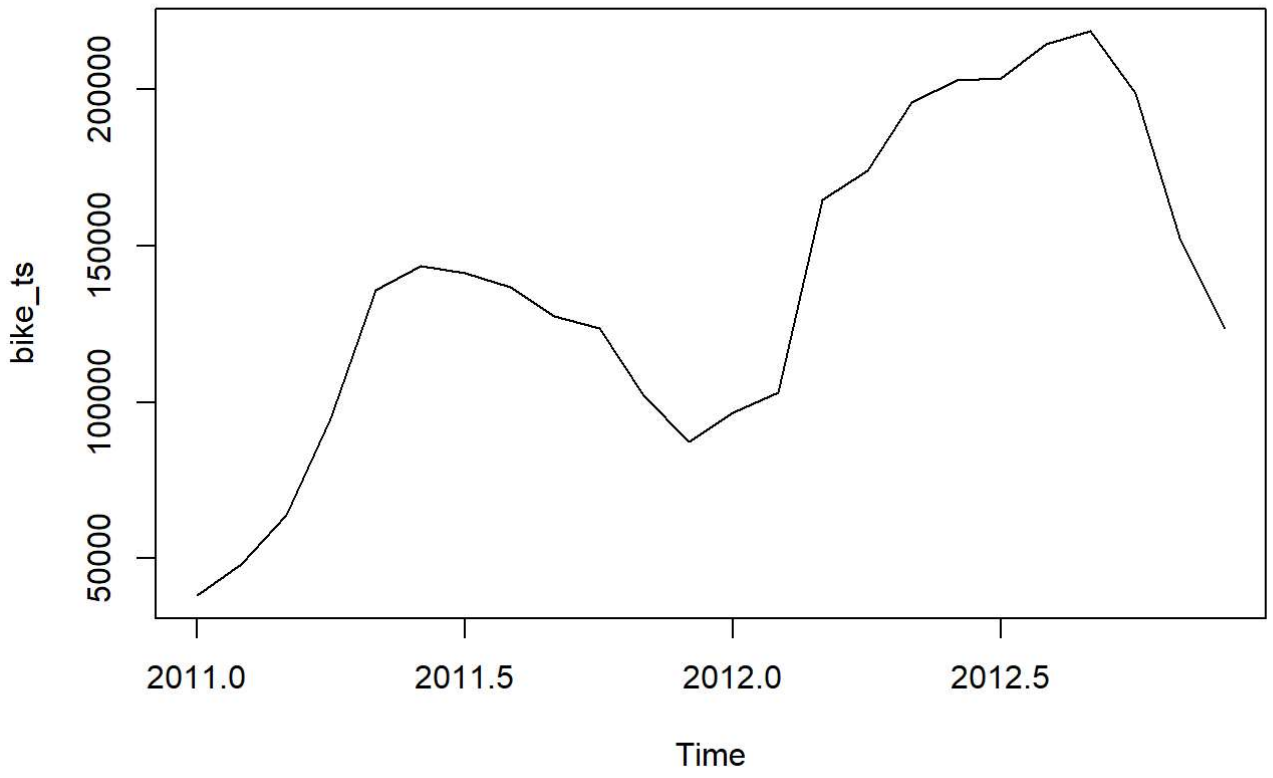
total_users <dbl>
38189
48215
64045
94870
135821
143512

total_users <dbl>
141341
136691
127418
123511

1-10 of 24 rows Previous **1** 2 3 Next

```
# Create time series object and plot time series

bike_ts <- ts(bike_month, start=c(2011,1), frequency = 12)
ts.plot(bike_ts)
```



Is this series white noise?

```
Box.test(bike_ts, lag=8, fitdf=0, type="Lj")
```

```
##
##  Box-Ljung test
##
## data:  bike_ts
## X-squared = 32.377, df = 8, p-value = 7.972e-05
```

ADF test for stationarity

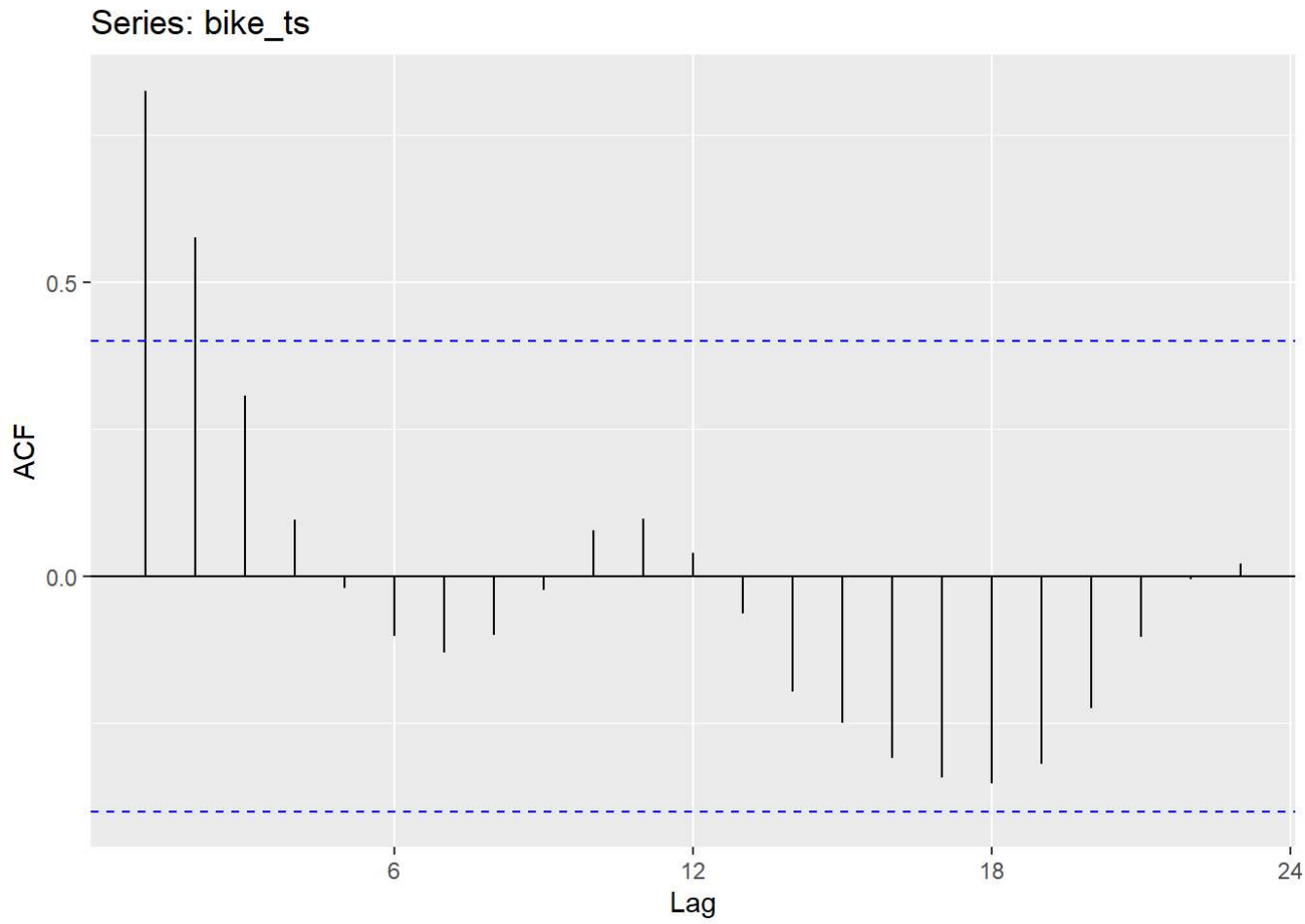
Use the Single Mean Version of the Test

```
bike_df <- ur.df(bike_ts, type = "drift")
summary(bike_df)
```

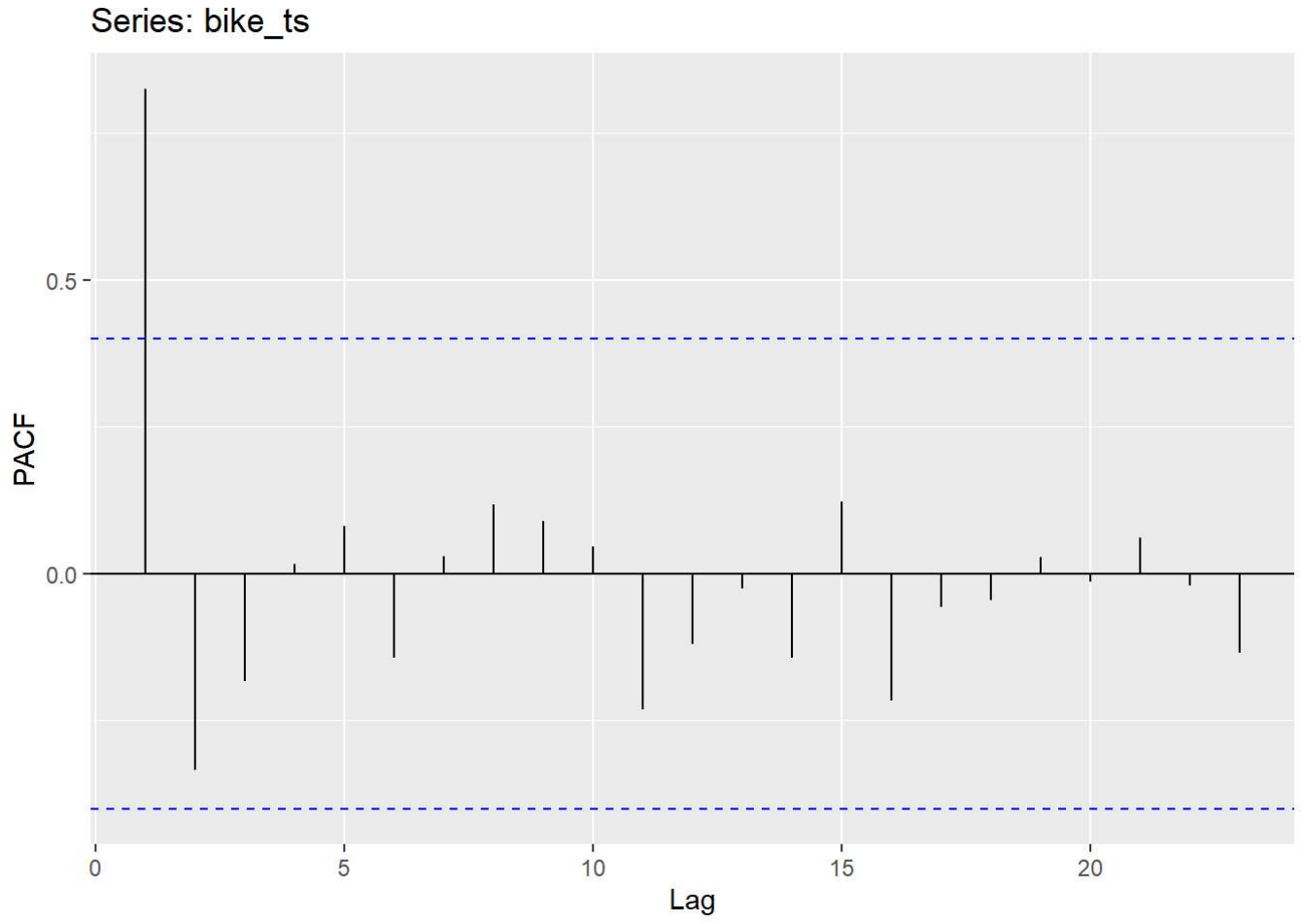
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -24689  -8605  -3191   7751  49979
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.839e+04  1.204e+04   2.359  0.02920 *
## z.lag.1     -1.957e-01  7.982e-02  -2.452  0.02405 *
## z.diff.lag   5.557e-01  1.785e-01   3.112  0.00574 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18250 on 19 degrees of freedom
## Multiple R-squared:  0.4508, Adjusted R-squared:  0.393
## F-statistic: 7.799 on 2 and 19 DF,  p-value: 0.003367
##
##
## Value of test-statistic is: -2.4519 3.0157
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.75 -3.00 -2.63
## phi1  7.88  5.18  4.12
```

Plot ACF and PACF

```
ggAcf(bike_ts)
```



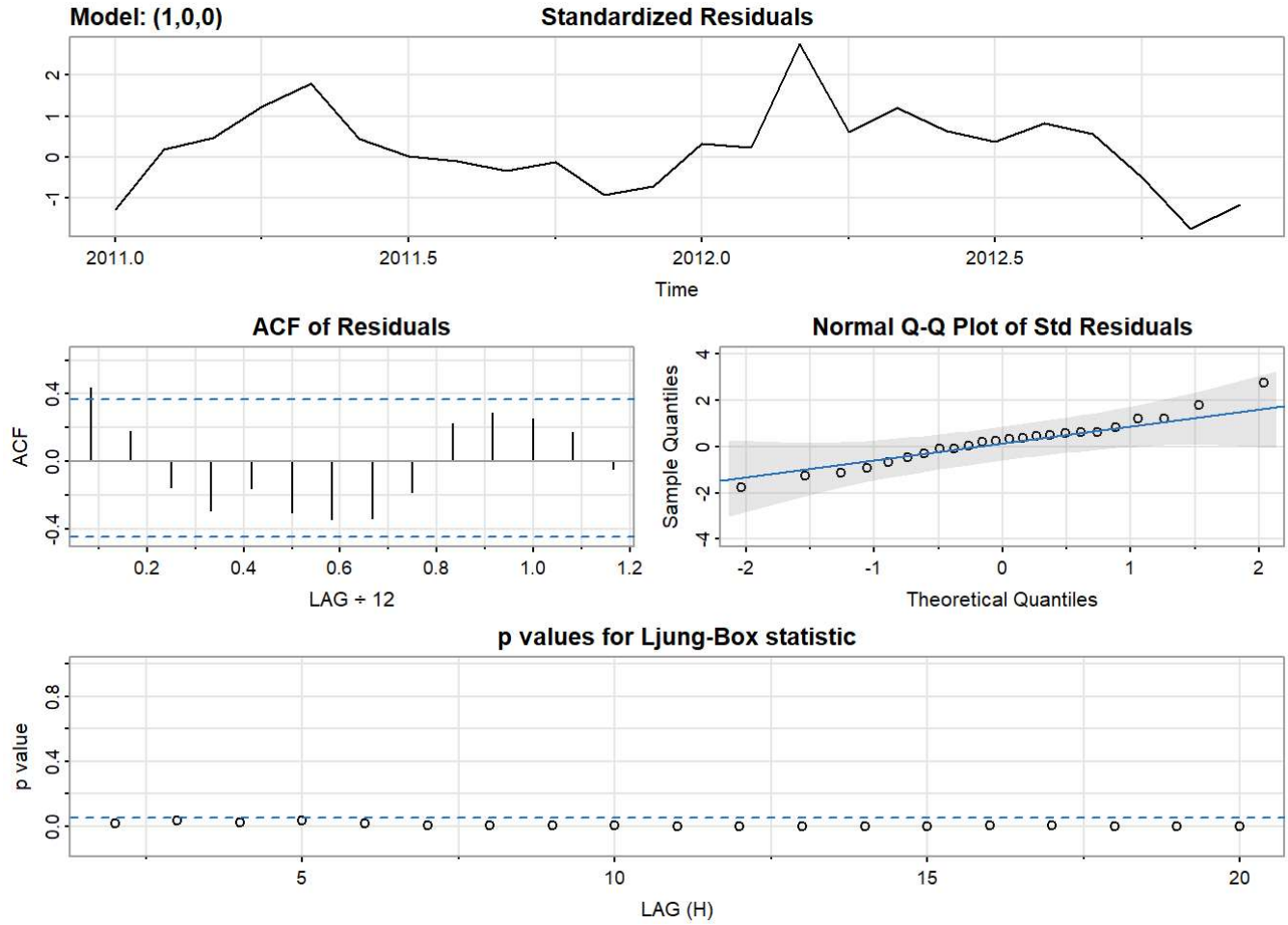

```
ggPacf(bike_ts)
```



Fit an ARIMA model (1, 0, 0)

```
fit_data_1 <- sarima(bike_ts, 1, 0, 0)
```

```
## initial  value 10.779319
## iter   2 value 10.009771
## iter   3 value 9.941103
## iter   4 value 9.933020
## iter   5 value 9.930092
## iter   6 value 9.930036
## iter   7 value 9.930034
## iter   8 value 9.930033
## iter   8 value 9.930032
## iter   8 value 9.930032
## final  value 9.930032
## converged
## initial  value 10.127748
## iter   2 value 10.054962
## iter   3 value 10.049884
## iter   4 value 10.047757
## iter   5 value 10.047453
## iter   6 value 10.047401
## iter   7 value 10.047394
## iter   8 value 10.047388
## iter   8 value 10.047388
## iter   8 value 10.047388
## final  value 10.047388
## converged
```




```
summary(fit_data_1)
```

```
##                   Length Class  Mode
## fit               14     Arima  list
## degrees_of_freedom  1     -none- numeric
## ttable              8     -none- numeric
## AIC                 1     -none- numeric
## AICc                1     -none- numeric
## BIC                 1     -none- numeric
```

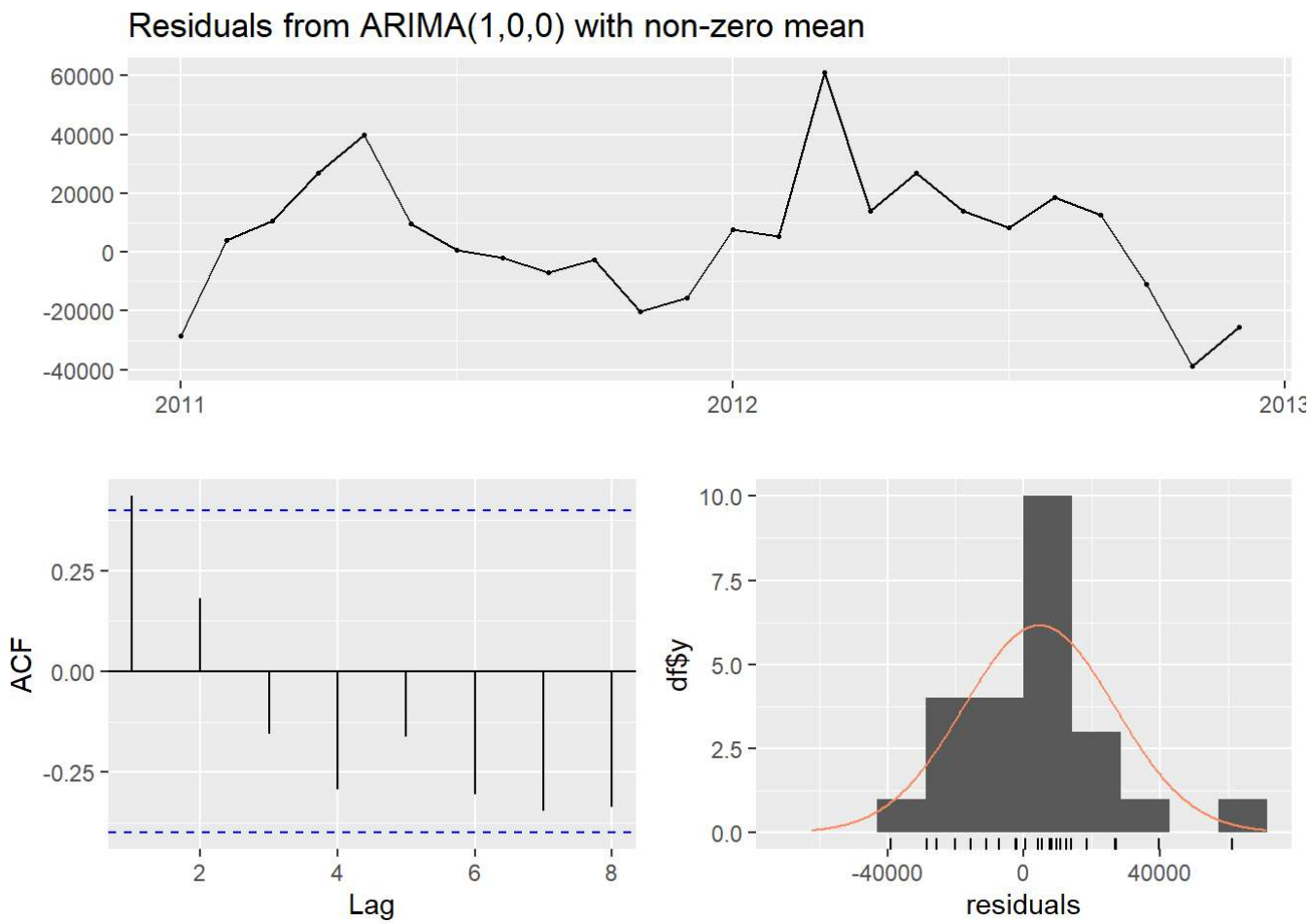
```
fit_data_1
```

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      xmean
##       0.9180  110087.01
## s.e.  0.0703   41893.55
##
## sigma^2 estimated as 493819101:  log likelihood = -275.19,  aic = 556.38
##
## $degrees_of_freedom
## [1] 22
##
## $ttable
##         Estimate         SE t.value p.value
## ar1        0.918     0.0703 13.0676  0.0000
## xmean 110087.010 41893.5459  2.6278  0.0154
##
## $AIC
## [1] 23.18265
##
## $AICc
## [1] 23.20646
##
## $BIC
## [1] 23.32991
```

```
fit_data_1 <- Arima(bike_ts, order=c(1, 0, 0))
summary(fit_data_1)
```

```
## Series: bike_ts
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1       mean
##       0.9180  110087.01
## s.e.  0.0703   41893.55
##
## sigma^2 = 538711747:  log likelihood = -275.19
## AIC=556.38   AICc=557.58   BIC=559.92
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE     MASE      ACF1
## Training set 4550.782 22222.04 17143.87 0.6122979 15.01429 0.255094 0.4353424
```

```
checkresiduals(fit_data_1)
```

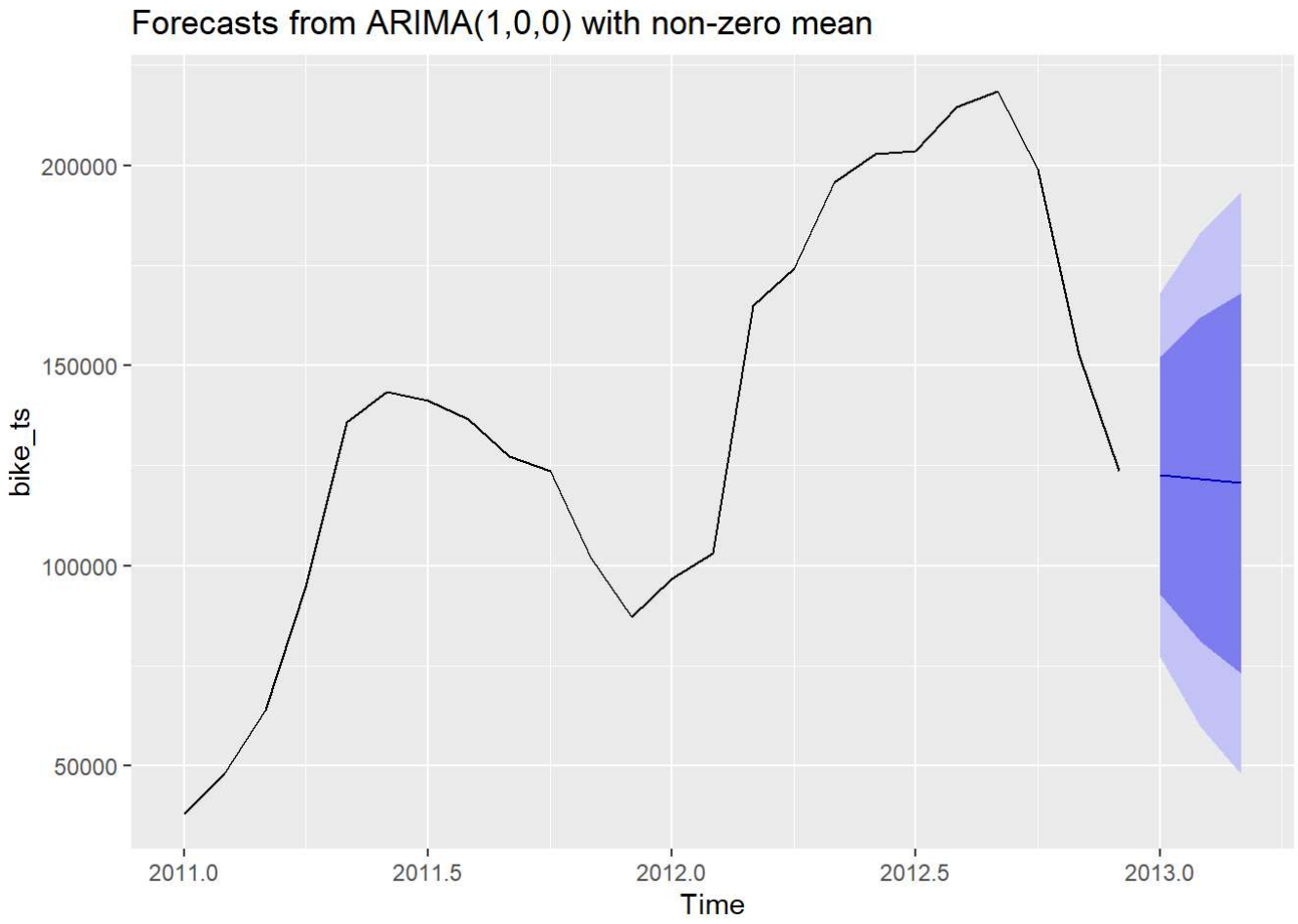



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with non-zero mean
## Q* = 10.315, df = 4, p-value = 0.03545
##
## Model df: 1.   Total lags used: 5
```

```
forecast(fit_data_1, h=3)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2013       122596.3 92851.31 152341.4 77105.25 168087.4
## Feb 2013       121571.2 81192.23 161950.1 59816.91 183325.5
## Mar 2013       120630.0 73101.73 168158.4 47941.76 193318.3
```

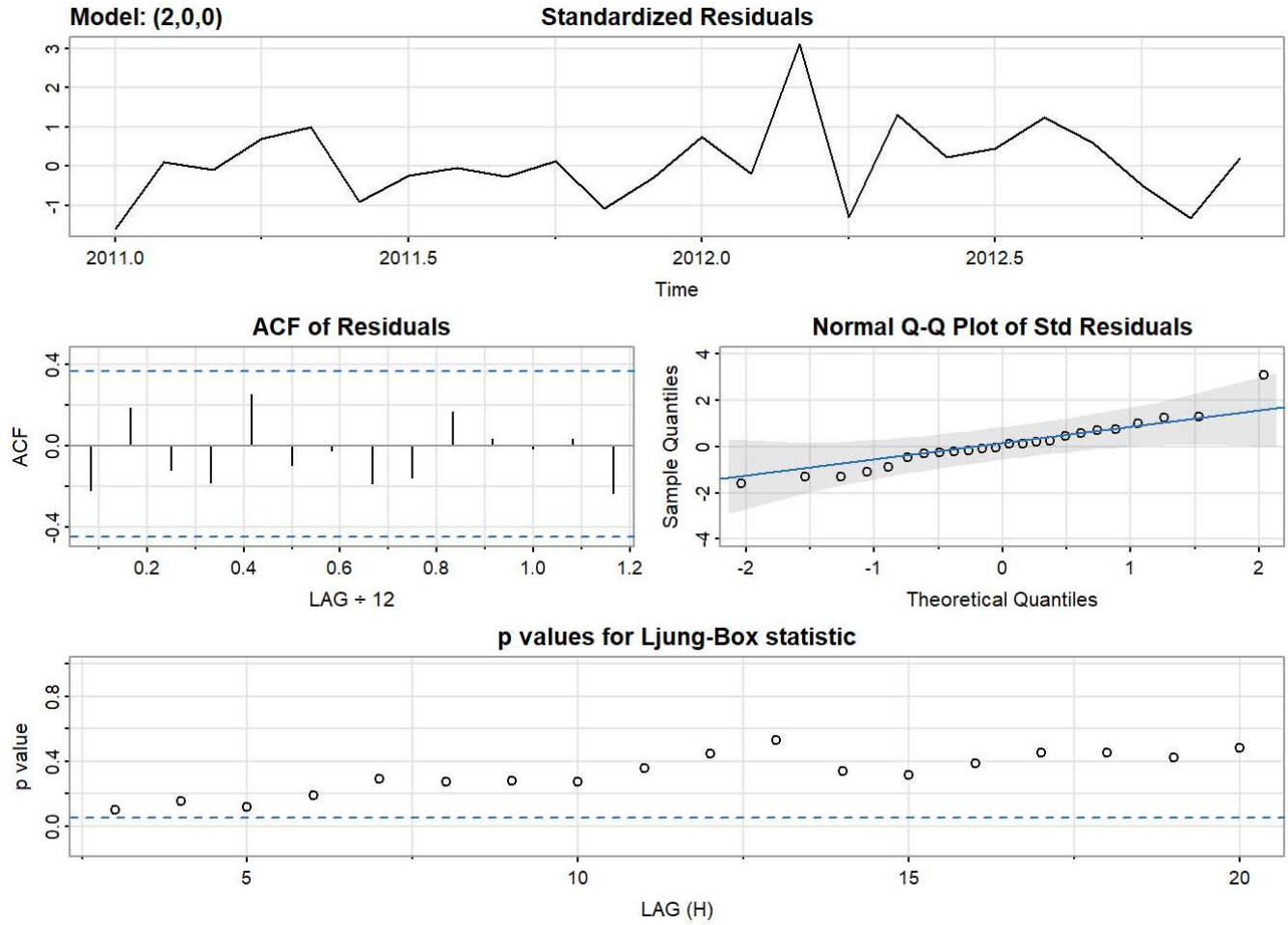
```
autoplot(forecast(fit_data_1, h=3))
```



Fit an ARIMA model (2, 0, 0)

```
fit_data_2 <- sarima(bike_ts, 2, 0, 0)
```

```
## initial  value 10.720697
## iter   2 value 10.373992
## iter   3 value 10.216281
## iter   4 value 10.110755
## iter   5 value 9.956252
## iter   6 value 9.823893
## iter   7 value 9.801890
## iter   8 value 9.795941
## iter   9 value 9.792613
## iter  10 value 9.756769
## iter  11 value 9.744934
## iter  12 value 9.740188
## iter  13 value 9.738704
## iter  14 value 9.738602
## iter  15 value 9.738594
## iter  16 value 9.738594
## iter  17 value 9.738594
## iter  17 value 9.738594
## iter  17 value 9.738594
## final  value 9.738594
## converged
## initial  value 9.870023
## iter   2 value 9.835384
## iter   3 value 9.834758
## iter   4 value 9.834722
## iter   5 value 9.834707
## iter   5 value 9.834707
## iter   5 value 9.834707
## final  value 9.834707
## converged
```




```
summary(fit_data_2)
```

```
##                   Length Class  Mode
## fit               14     Arima  list
## degrees_of_freedom  1     -none- numeric
## ttable             12     -none- numeric
## AIC                 1     -none- numeric
## AICc                1     -none- numeric
## BIC                 1     -none- numeric
```

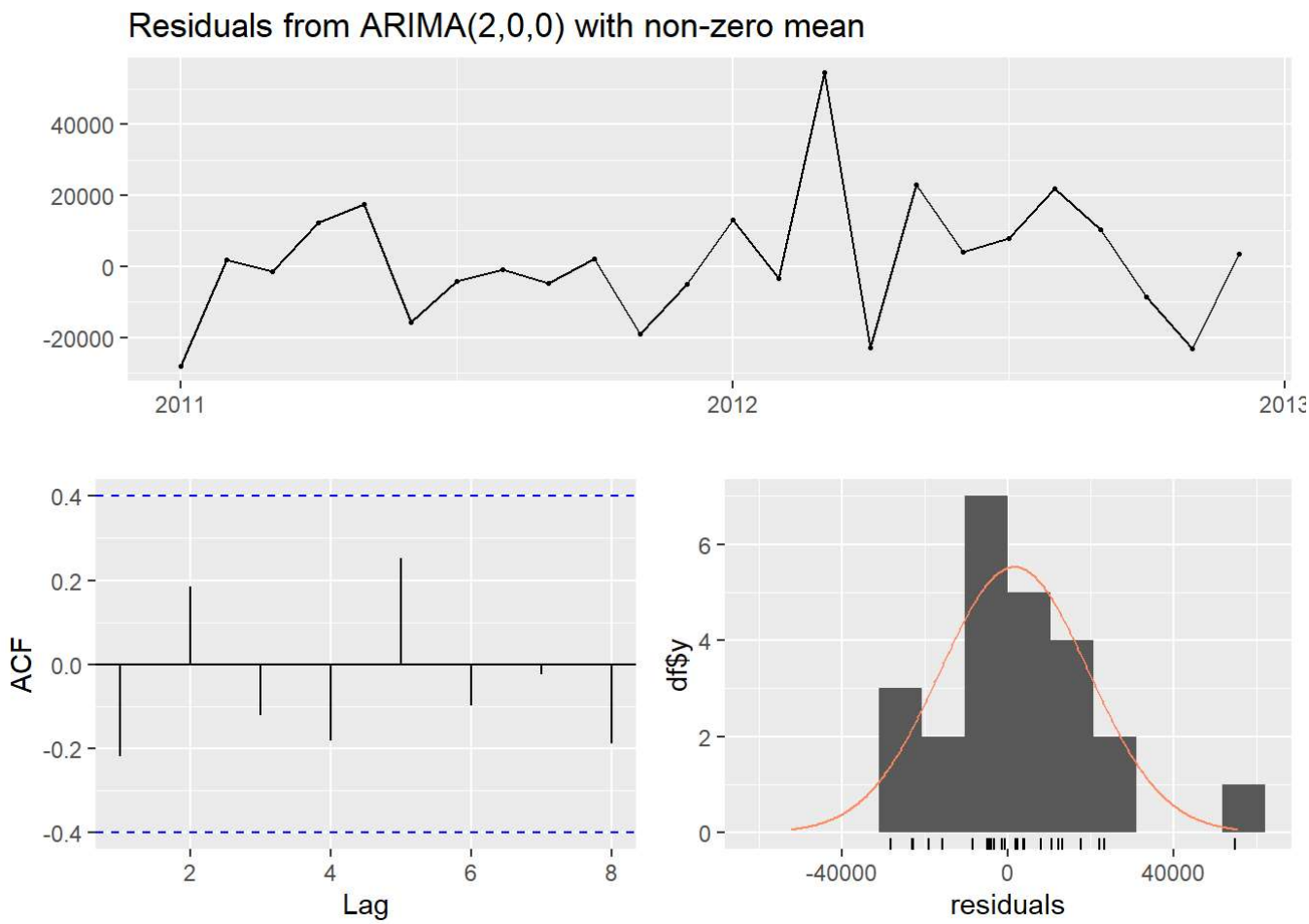
```
fit_data_2
```

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2      xmean
##       1.4699  -0.6156  124399.04
## s.e.  0.1569   0.1648   24664.58
##
## sigma^2 estimated as 311349057:  log likelihood = -270.09,  aic = 548.17
##
## $degrees_of_freedom
## [1] 21
##
## $ttable
##          Estimate         SE t.value p.value
## ar1        1.4699     0.1569  9.3707  0.0000
## ar2       -0.6156     0.1648 -3.7345  0.0012
## xmean 124399.0388 24664.5847  5.0436  0.0001
##
## $AIC
## [1] 22.84062
##
## $AICc
## [1] 22.89062
##
## $BIC
## [1] 23.03697
```

```
fit_data_2 <- Arima(bike_ts, order=c(2, 0, 0))
summary(fit_data_2)
```

```
## Series: bike_ts
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2       mean
##       1.4699  -0.6156  124399.04
## s.e.  0.1569   0.1648   24664.58
##
## sigma^2 = 355827493:  log likelihood = -270.09
## AIC=548.17   AICc=550.28   BIC=552.89
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
## Training set 1500.392 17645.09 12915.86 -1.691429 11.18161 0.1921829 -0.2194434
```

```
checkresiduals(fit_data_2)
```

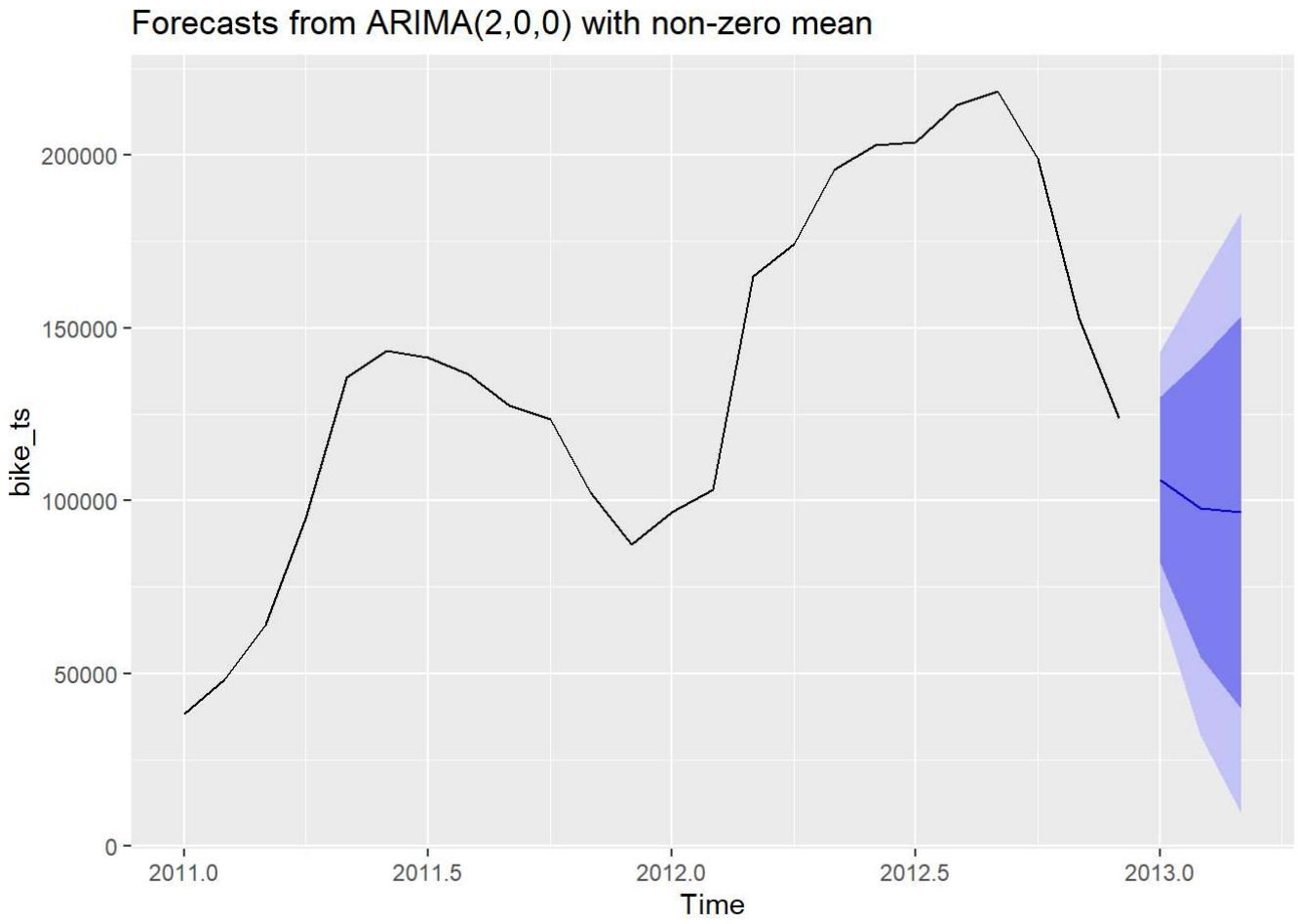



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,0) with non-zero mean
## Q* = 5.8338, df = 3, p-value = 0.12
##
## Model df: 2.   Total lags used: 5
```

```
forecast(fit_data_2, h=3)
```

```
##          Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## Jan 2013      105990.20 81815.79 130164.6 69018.632 142961.8
## Feb 2013       97761.83 54783.91 140739.7 32032.776 163490.9
## Mar 2013       96577.24 39637.14 153517.3  9494.866 183659.6
```

```
autoplot(forecast(fit_data_2, h=3))
```

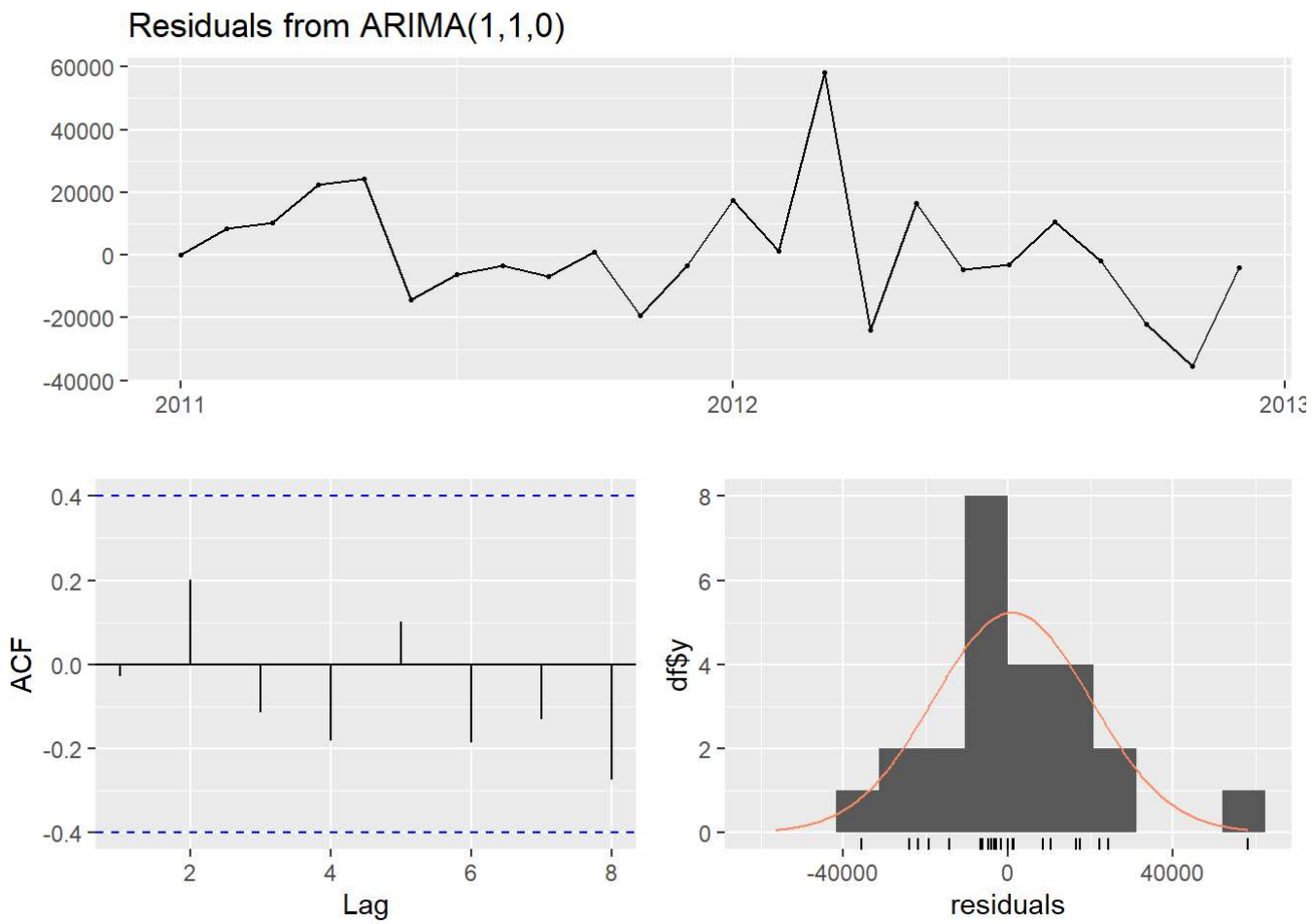


Use auto.arima (final model)

```
fitauto <- auto.arima(bike_ts)
summary(fitauto)
```

```
## Series: bike_ts
## ARIMA(1,1,0)
##
## Coefficients:
##          ar1
##       0.5383
## s.e.  0.1755
##
## sigma^2 = 382385344:  log likelihood = -259.56
## AIC=523.12   AICc=523.72   BIC=525.39
##
## Training set error measures:
##                    ME     RMSE      MAE      MPE     MAPE    MASE         ACF1
## Training set 931.7514 18722.18 13300.08 1.804659 10.21507 0.1979 -0.02802576
```

```
checkresiduals(fitauto)
```

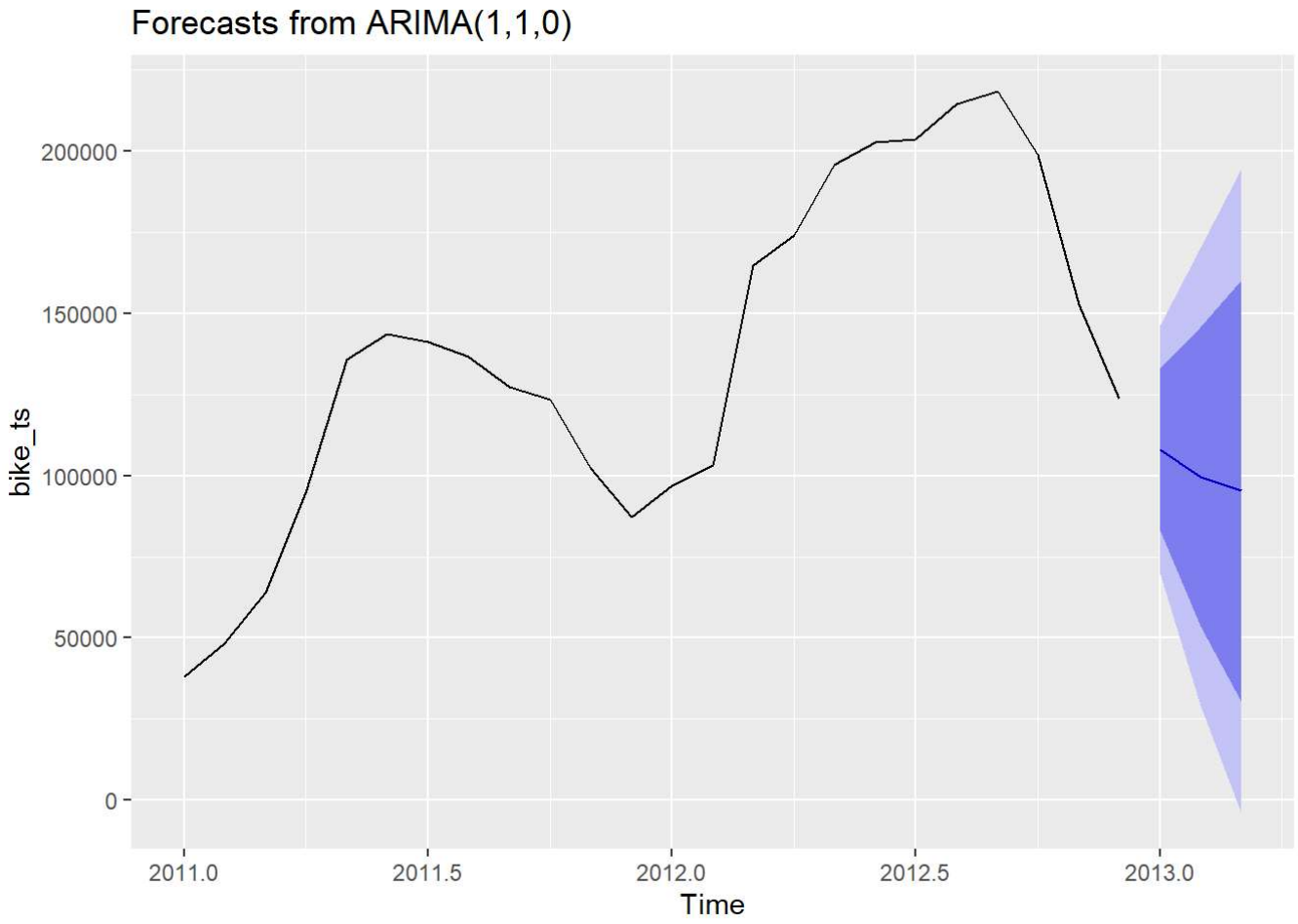



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)
## Q* = 2.9228, df = 4, p-value = 0.5708
##
## Model df: 1.   Total lags used: 5
```

```
fitauto %>% forecast(h=3)
```

```
##          Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## Jan 2013      108128.27 83067.95 133188.6 69801.813 146454.7
## Feb 2013       99738.80 53758.67 145718.9 29418.257 170059.3
## Mar 2013       95222.63 30315.20 160130.0 -4044.712 194490.0
```

```
fitauto %>% forecast(h=3) %>% autoplot()
```


Forecasts from ARIMA(1,1,0)
200000
150000
100000
50000
0
bike_ts
2011.0
2011.5
2012.0
2012.5
2013.0
Time