Appendix

Data summary

```
## 'data.frame':    2935849 obs. of  6 variables:
##  $ date          : Factor w/ 1034 levels "01.01.2013","01.01.2014",..: 35 69 137 171 477 30
##  $ date_block_num: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ shop_id       : int  59 25 25 25 25 25 25 25 25 25 ...
##  $ item_id       : int  22154 2552 2552 2554 2555 2564 2565 2572 2572 2573 ...
##  $ item_price    : num  999 899 899 1709 1099 ...
##  $ item_cnt_day  : num  1 1 -1 1 1 1 1 1 1 3 ...
```

```
## [1] 0
```

```
##       date             date_block_num     shop_id      item_id
##  Min.   :2013-01-01   Min.   : 0.00   Min.   : 0   Min.   :    0
##  1st Qu.:2013-08-01   1st Qu.: 7.00   1st Qu.:22   1st Qu.: 4476
##  Median :2014-03-04   Median :14.00   Median :31   Median : 9343
##  Mean   :2014-04-03   Mean   :14.57   Mean   :33   Mean   :10197
##  3rd Qu.:2014-12-05   3rd Qu.:23.00   3rd Qu.:47   3rd Qu.:15684
##  Max.   :2015-10-31   Max.   :33.00   Max.   :59   Max.   :22169
##
##    item_price        item_cnt_day        year             month
##  Min.   :    -1.0   Min.   : -22.000   2013:1267562   1      : 303561
##  1st Qu.:   249.0   1st Qu.:   1.000   2014:1055861   3      : 284057
##  Median :   399.0   Median :   1.000   2015: 612426   12     : 274032
##  Mean   :   890.9   Mean   :   1.243                  2      : 270251
##  3rd Qu.:   999.0   3rd Qu.:   1.000                  8      : 248415
##  Max.   :307980.0   Max.   :2169.000                  6      : 237428
##                                                       (Other):1318105
##       day
##  2      : 103372
##  7      : 102273
##  22     : 101345
##  23     : 101339
##  8      : 100986
##  21     : 100208
##  (Other):2326326
```

Group by month sales

```
## # A tibble: 34 x 3
## # Groups:   year [3]
##    year  month total_sales_month
##    <fct> <fct>             <dbl>
##  1 2013  1                131479
##  2 2013  2                128090
##  3 2013  3                147142
```

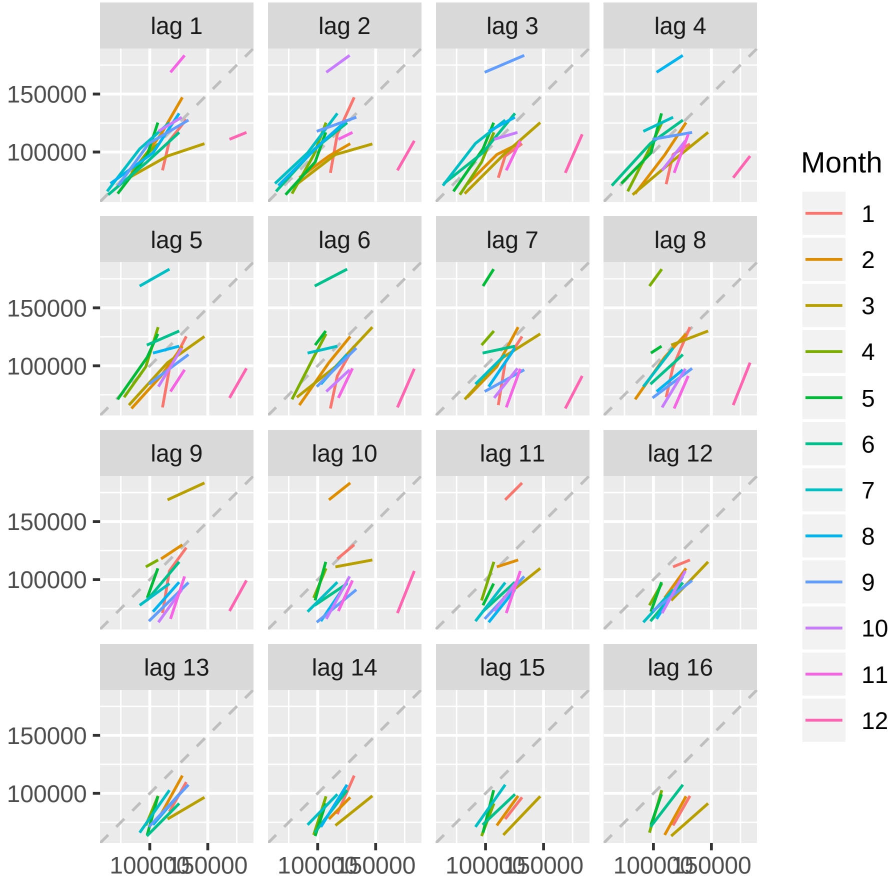
```
## 4 2013  4                107190
## 5 2013  5                106970
## 6 2013  6                125381
## 7 2013  7                116966
## 8 2013  8                125291
## 9 2013  9                133332
## 10 2013 10               127541
## # ... with 24 more rows
```

```
## # A tibble: 2,935,849 x 9
## # Groups:   year, month [34]
##    date       date_block_num shop_id item_id item_price item_cnt_day year  month
##    <date>              <int>   <int>   <int>      <dbl>        <dbl> <fct> <fct>
##  1 2013-01-02              0      59   22154       999             1 2013  1
##  2 2013-01-03              0      25    2552       899             1 2013  1
##  3 2013-01-05              0      25    2552       899            -1 2013  1
##  4 2013-01-06              0      25    2554      1709.            1 2013  1
##  5 2013-01-15              0      25    2555      1099             1 2013  1
##  6 2013-01-10              0      25    2564       349             1 2013  1
##  7 2013-01-02              0      25    2565       549             1 2013  1
##  8 2013-01-04              0      25    2572       239             1 2013  1
##  9 2013-01-11              0      25    2572       299             1 2013  1
## 10 2013-01-03              0      25    2573       299             3 2013  1
## # ... with 2,935,839 more rows, and 1 more variable: day <fct>
```

Summary of ts object data

```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2013 131479 128090 147142 107190 106970 125381 116966 125291 133332 127541
## 2014 116899 109687 115297  96556  97790  97429  91280 102721  99208 107422
## 2015 110971  84198  82014  77827  72295  64114  63187  66079  72843  71056
##         Nov    Dec
## 2013 130009 183342
## 2014 117845 168755
## 2015
```

Scatterplot for lag



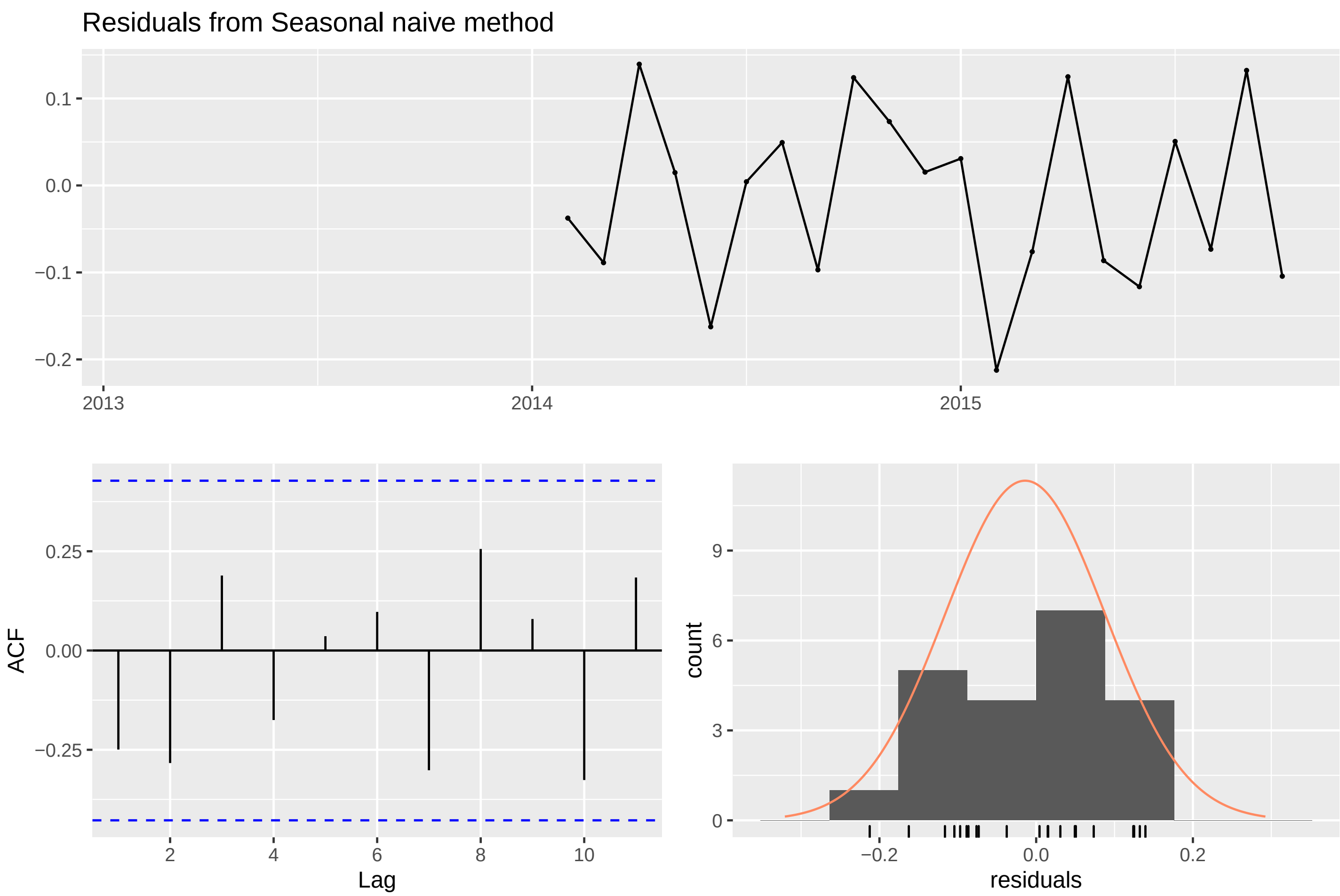
Dicky-Fuller test

```
##
##  Augmented Dickey-Fuller Test
##
## data:  sales_monthly_ts
## Dickey-Fuller = -0.32986, Lag order = 12, p-value = 0.9835
## alternative hypothesis: stationary
```

Seasonal naive output

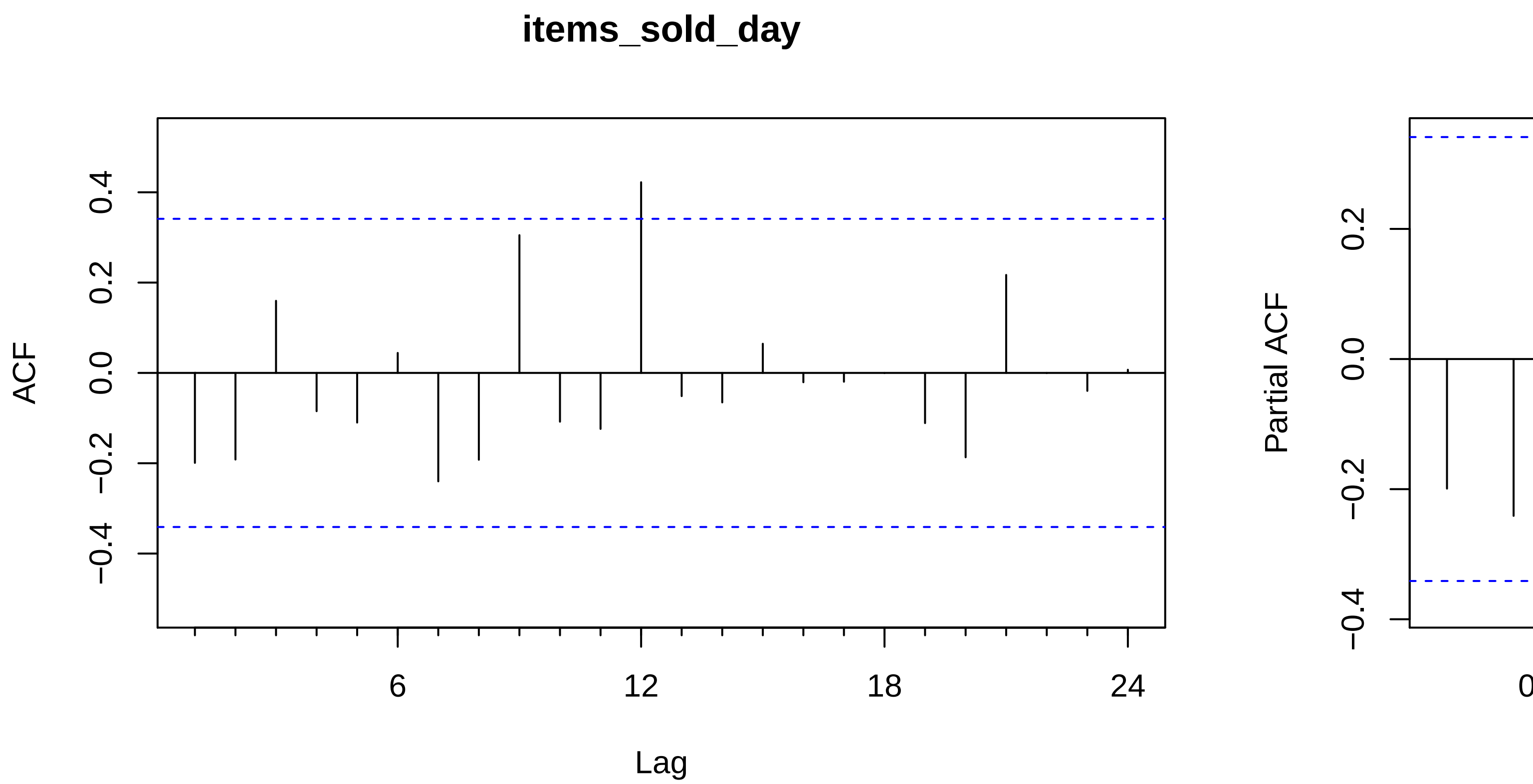
```
##
## Forecast method: Seasonal naive method
##
## Model Information:
## Call: snaive(y = changeofitemsales)
##
## Residual sd: 0.1022
##
## Error measures:
```

```
##                      ME      RMSE        MAE      MPE     MAPE MASE       ACF1
## Training set -0.0140838 0.1007235 0.08639063 249.7803 346.9721    1 -0.2493608
##
## Forecasts:
##          Point Forecast       Lo 80        Hi 80       Lo 95        Hi 95
## Nov 2015     0.09260520 -0.03647718  0.221687583 -0.10480927  0.290019669
## Dec 2015     0.35907776  0.22999537  0.488160141  0.16166329  0.556492227
## Jan 2016    -0.41917905 -0.54826144 -0.290096669 -0.61659352 -0.221764583
## Feb 2016    -0.27609774 -0.40518012 -0.147015354 -0.47351221 -0.078683268
## Mar 2016    -0.02628120 -0.15536359  0.102801180 -0.22369567  0.171133267
## Apr 2016    -0.05240155 -0.18148393  0.076680834 -0.24981602  0.145012920
## May 2016    -0.07373344 -0.20281583  0.055348940 -0.27114791  0.123681026
## Jun 2016    -0.12009222 -0.24917461  0.008990162 -0.31750669  0.077322248
## Jul 2016    -0.01456417 -0.14364655  0.114518219 -0.21197864  0.182850305
## Aug 2016     0.04475241 -0.08432997  0.173834796 -0.15266206  0.242166882
## Sep 2016     0.09745544 -0.03162694  0.226537828 -0.09995903  0.294869914
## Oct 2016    -0.02483814 -0.15392053  0.104244242 -0.22225261  0.172576328
## Nov 2016     0.09260520 -0.08994486  0.275155257 -0.18658102  0.371791420
## Dec 2016     0.35907776  0.17652770  0.541627815  0.07989154  0.638263978
## Jan 2017    -0.41917905 -0.60172911 -0.236628995 -0.69836527 -0.139992832
## Feb 2017    -0.27609774 -0.45864780 -0.093547680 -0.55528396  0.003088483
## Mar 2017    -0.02628120 -0.20883126  0.156268854 -0.30546742  0.252905017
## Apr 2017    -0.05240155 -0.23495161  0.130148508 -0.33158777  0.226784671
## May 2017    -0.07373344 -0.25628350  0.108816614 -0.35291967  0.205452777
## Jun 2017    -0.12009222 -0.30264228  0.062457836 -0.39927844  0.159093999
## Jul 2017    -0.01456417 -0.19711422  0.167985893 -0.29375039  0.264622056
## Aug 2017     0.04475241 -0.13779765  0.227302470 -0.23443381  0.323938633
## Sep 2017     0.09745544 -0.08509461  0.280005502 -0.18173078  0.376641665
## Oct 2017    -0.02483814 -0.20738820  0.157711916 -0.30402436  0.254348079
```



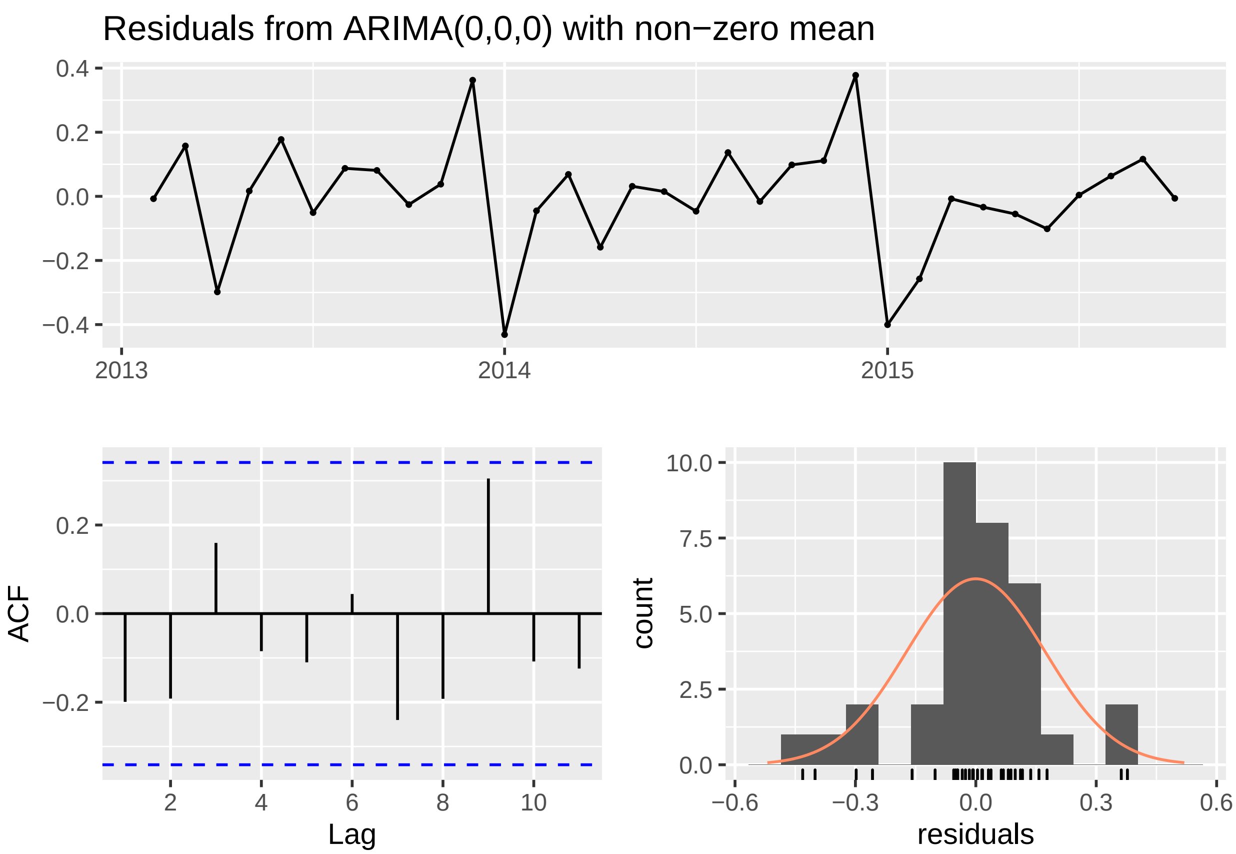

```
##
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 8.8521, df = 7, p-value = 0.2635
##
## Model df: 0.   Total lags used: 7
```

ACF and PACF for ARIMA



ARIMA (0,0,0)

```
##
## Call:
## arima(x = monthly_stationary, order = c(0, 0, 0), seasonal = list(order = c(0,
##     0, 0), period = 12))
##
## Coefficients:
##       intercept
##         -0.0186
## s.e.     0.0297
##
## sigma^2 estimated as 0.02908:  log likelihood = 11.55,  aic = -19.1
##
## Training set error measures:
##                         ME      RMSE       MAE      MPE     MAPE      MASE
## Training set -8.426951e-19 0.1705142 0.1177058 52.99428 128.1393 0.6187436
##                    ACF1
## Training set -0.1992047
```

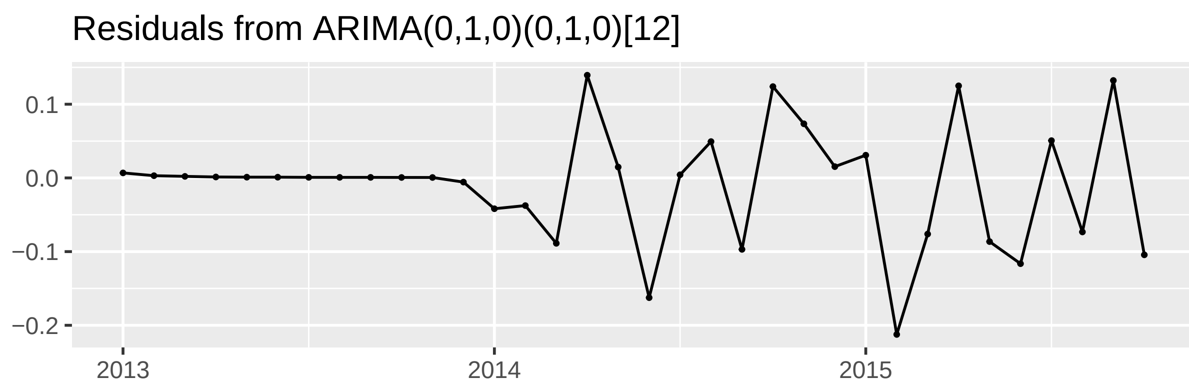


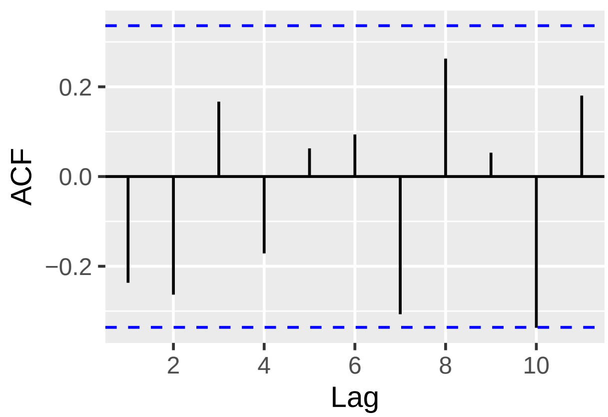

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,0) with non-zero mean
## Q* = 7.2141, df = 6, p-value = 0.3015
##
## Model df: 1.   Total lags used: 7
```

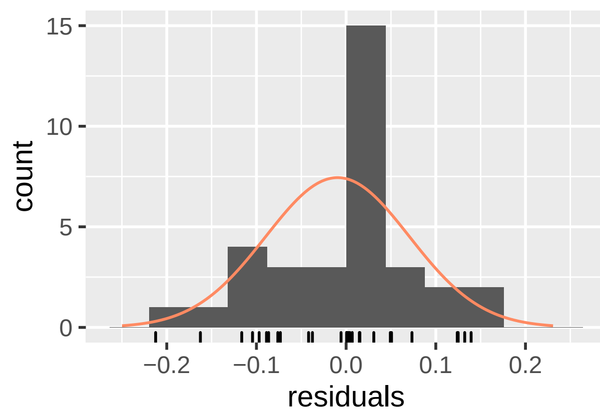
```
## [1] 0.1705286
```

ARIMA (0,1,0)

```
## Series: log(sales_monthly_ts)
## ARIMA(0,1,0)(0,1,0)[12]
##
## sigma^2 estimated as 0.01023:  log likelihood=18.41
## AIC=-34.81   AICc=-34.6   BIC=-33.77
##
## Training set error measures:
##                        ME       RMSE        MAE         MPE      MAPE     MASE
## Training set -0.009545926 0.07950166 0.05530518 -0.08547149 0.4859299 0.234775
##                    ACF1
## Training set -0.2367404
```








```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,0)(0,1,0)[12]
## Q* = 11.845, df = 7, p-value = 0.1058
##
## Model df: 0.   Total lags used: 7
```

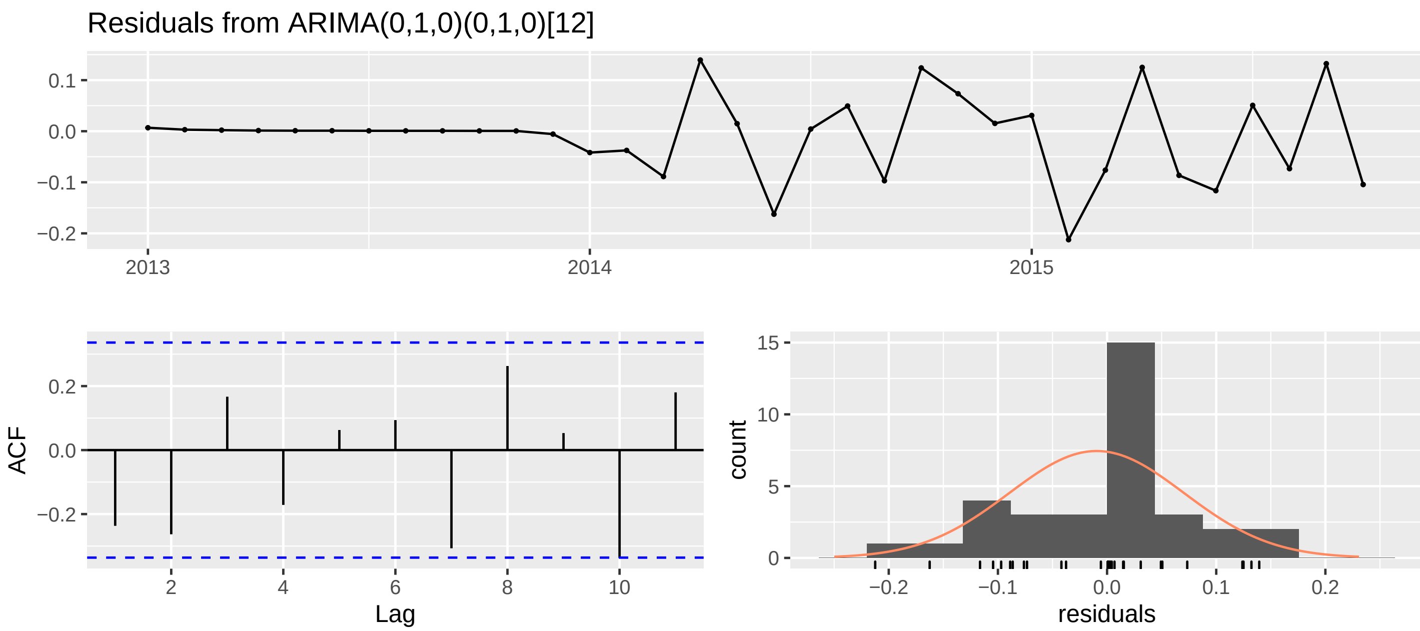
```
## [1] 0.1011435
```

ARIMA (0,1,0) fitting

```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2013 131479 128090 147142 107190 106970 125381 116966 125291 133332 127541
## 2014 116899 109687 115297  96556  97790  97429  91280 102721  99208 107422
##         Nov    Dec
## 2013 130009 183342
## 2014 117845 168755
```

```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2015 141604 130143 124927 122475 121304 120740 120468 120336 120273 120242
## 2016 120216 120214 120214 120213 120213 120213 120213 120213 120213 120213
##         Nov    Dec
## 2015 120227 120220
## 2016 120213 120213
```

Forecasting



Reference

[1] Coursera (2018). Predict Future Sales. Retrieved April 10, 2020 from https://www.kaggle.com/c/competitive-data-science-predict-future-sales/data.

[2] Brownlee, J. (2016). How to Check if Time Series Data is Stationary with Python. Retrieved April 10, 2020 from https://machinelearningmastery.com/time-series-data-stationary-python

[3] Schneider, O. (2020). Seminar 27: Time series, lecture notes, Statistical Methods for Data Analytics MSCI 718, University of Waterloo, delivered in Mar 2020.

[4] Rob J Hyndman and George Athanasopoulos (2018). Forecasting principles and practice. Retrieved April 11, 2020 from https://otexts.com/fpp2.