Time Series Models

```
## # A tibble: 1,726 x 20
##    TaxID   Outbreak SRA_release_date SRA_Center AMR_genotypes_co~ Contigs    N50
##    <fct>      <dbl> <date>           <fct>      <fct>               <dbl>  <dbl>
##  1 1399004        0 2013-09-10       CFSAN      fosX=COMPLETE,li~      18 535981
##  2 1399005        0 2013-09-10       CFSAN      fosX=COMPLETE,li~      16 584558
##  3 1639           0 2013-10-25       CFSAN      fosX=COMPLETE,li~      17 545164
##  4 1639           0 2018-07-23       CFSAN      fosX=COMPLETE,li~      14 527852
##  5 1639           0 2014-01-24       CFSAN      fosX=COMPLETE,li~      22 410100
##  6 1639           0 2014-01-24       CFSAN      fosX=COMPLETE,li~      25 438054
##  7 1639           0 2014-01-24       CFSAN      fosX=COMPLETE,li~      19 437998
##  8 1639           0 2014-01-24       CFSAN      fosX=COMPLETE,li~      21 545215
##  9 1639           0 2014-01-24       CFSAN      fosX=COMPLETE,li~      21 545164
## 10 1639           0 2014-01-24       CFSAN      fosX=COMPLETE,li~      37 545164
## # ... with 1,716 more rows, and 13 more variables: Length <dbl>,
## #   BioProject <fct>, Collection_date <fct>, Collected_by <fct>,
## #   Scientific_name <fct>, Create_date <date>, Location <fct>,
## #   Isolation_source <fct>, Isolation_type <fct>, SNP_cluster <fct>,
## #   `Min-same` <dbl>, `Min-diff` <dbl>, AMR_genotypes <fct>
```

We will build time series models using cluster PDS000000366.488 data.

```
##        Date Frequency
## 1   2013-11         2
## 2   2013-12         0
## 3   2014-01        20
## 4   2014-02         0
## 5   2014-03         4
## 6   2014-04         0
## 7   2014-05         7
## 8   2014-06         9
## 9   2014-07         8
## 10  2014-08        34
## 11  2014-09         2
## 12  2014-10         9
## 13  2014-11         3
## 14  2014-12         6
## 15  2015-01         4
## 16  2015-02         3
## 17  2015-03         2
## 18  2015-04         4
## 19  2015-05        22
## 20  2015-06         6
## 21  2015-07        18
## 22  2015-08         4
## 23  2015-09         1
## 24  2015-10        44
## 25  2015-11        25
## 26  2015-12        30
```

```
## 27  2016-01        34
## 28  2016-02        32
## 29  2016-03        16
## 30  2016-04        35
## 31  2016-05        19
## 32  2016-06        13
## 33  2016-07        22
## 34  2016-08        38
## 35  2016-09        10
## 36  2016-10        11
## 37  2016-11        26
## 38  2016-12        11
## 39  2017-01         8
## 40  2017-02        14
## 41  2017-03        36
## 42  2017-04        54
## 43  2017-05        18
## 44  2017-06        52
## 45  2017-07        18
## 46  2017-08         7
## 47  2017-09        12
## 48  2017-10        35
## 49  2017-11        16
## 50  2017-12        14
## 51  2018-01        21
## 52  2018-02        29
## 53  2018-03        29
## 54  2018-04        31
## 55  2018-05        35
## 56  2018-06        41
## 57  2018-07         5
## 58  2018-08         9
## 59  2018-09        36
## 60  2018-10        14
## 61  2018-11        10
## 62  2018-12        21
## 63  2019-01        12
## 64  2019-02        19
## 65  2019-03        32
## 66  2019-04        13
## 67  2019-05        11
## 68  2019-06         2
## 69  2019-07        17
## 70  2019-08        15
## 71  2019-09        11
## 72  2019-10         6
## 73  2019-11         8
## 74  2019-12        43
## 75  2020-01       174
## 76  2020-02         8
## 77  2020-03         2
## 78  2020-04        10
## 79  2020-05         1
## 80  2020-06         2
```

```
## 81  2020-07          7
## 82  2020-08          2
## 83  2020-09          6
## 84  2020-10         44
## 85  2020-11          1
## 86  2020-12          6
## 87  2021-01          6
## 88  2021-02          4
## 89  2021-03          0
## 90  2021-04         28
## 91  2021-05          7
## 92  2021-06          8
## 93  2021-07          4
## 94  2021-08         10
## 95  2021-09         14
## 96  2021-10          4
## 97  2021-11          7
## 98  2021-12          5
## 99  2022-01          9
## 100 2022-02         16
## 101 2022-03         15
## 102 2022-04          3
## 103 2022-05         21
## 104 2022-06         14
```

We construct a dataframe that are suitable for the time series models.

```
##          Date Frequency
## Nov 2013    1         2
## Dec 2013    2         0
## Jan 2014    3        20
## Feb 2014    4         0
## Mar 2014    5         4
## Apr 2014    6         0
## May 2014    7         7
## Jun 2014    8         9
## Jul 2014    9         8
## Aug 2014   10        34
## Sep 2014   11         2
## Oct 2014   12         9
## Nov 2014   13         3
## Dec 2014   14         6
## Jan 2015   15         4
## Feb 2015   16         3
## Mar 2015   17         2
## Apr 2015   18         4
## May 2015   19        22
## Jun 2015   20         6
## Jul 2015   21        18
## Aug 2015   22         4
## Sep 2015   23         1
## Oct 2015   24        44
## Nov 2015   25        25
## Dec 2015   26        30
## Jan 2016   27        34
```

```
## Feb 2016   28          32
## Mar 2016   29          16
## Apr 2016   30          35
## May 2016   31          19
## Jun 2016   32          13
## Jul 2016   33          22
## Aug 2016   34          38
## Sep 2016   35          10
## Oct 2016   36          11
## Nov 2016   37          26
## Dec 2016   38          11
## Jan 2017   39           8
## Feb 2017   40          14
## Mar 2017   41          36
## Apr 2017   42          54
## May 2017   43          18
## Jun 2017   44          52
## Jul 2017   45          18
## Aug 2017   46           7
## Sep 2017   47          12
## Oct 2017   48          35
## Nov 2017   49          16
## Dec 2017   50          14
## Jan 2018   51          21
## Feb 2018   52          29
## Mar 2018   53          29
## Apr 2018   54          31
## May 2018   55          35
## Jun 2018   56          41
## Jul 2018   57           5
## Aug 2018   58           9
## Sep 2018   59          36
## Oct 2018   60          14
## Nov 2018   61          10
## Dec 2018   62          21
## Jan 2019   63          12
## Feb 2019   64          19
## Mar 2019   65          32
## Apr 2019   66          13
## May 2019   67          11
## Jun 2019   68           2
## Jul 2019   69          17
## Aug 2019   70          15
## Sep 2019   71          11
## Oct 2019   72           6
## Nov 2019   73           8
## Dec 2019   74          43
## Jan 2020   75         174
## Feb 2020   76           8
## Mar 2020   77           2
## Apr 2020   78          10
## May 2020   79           1
## Jun 2020   80           2
## Jul 2020   81           7
```

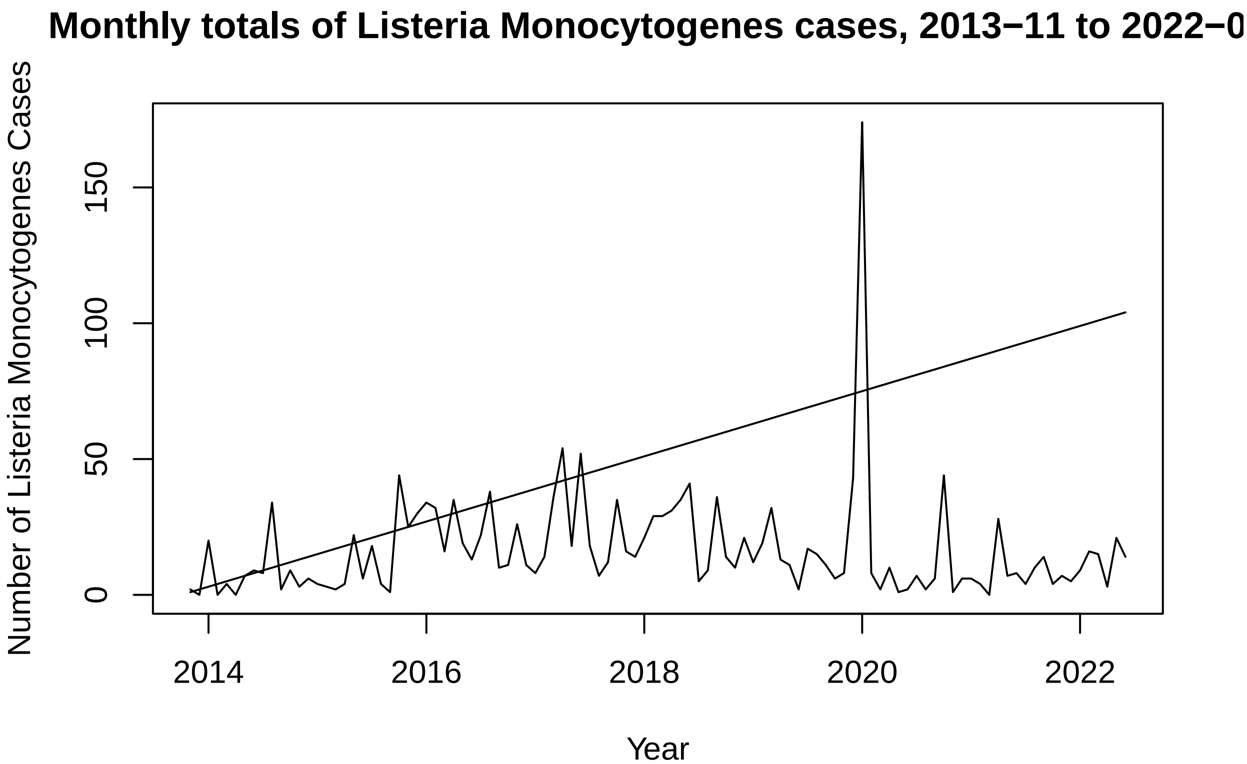
```
## Aug 2020   82          2
## Sep 2020   83          6
## Oct 2020   84         44
## Nov 2020   85          1
## Dec 2020   86          6
## Jan 2021   87          6
## Feb 2021   88          4
## Mar 2021   89          0
## Apr 2021   90         28
## May 2021   91          7
## Jun 2021   92          8
## Jul 2021   93          4
## Aug 2021   94         10
## Sep 2021   95         14
## Oct 2021   96          4
## Nov 2021   97          7
## Dec 2021   98          5
## Jan 2022   99          9
## Feb 2022  100         16
## Mar 2022  101         15
## Apr 2022  102          3
## May 2022  103         21
## Jun 2022  104         14
```

We convert above dataframe to time series format.

```
## [1] TRUE
```

```
##       Date          Frequency
##  Min.   :  1.00   Min.   :  0.00
##  1st Qu.: 26.75   1st Qu.:  5.75
##  Median : 52.50   Median : 11.00
##  Mean   : 52.50   Mean   : 16.60
##  3rd Qu.: 78.25   3rd Qu.: 21.25
##  Max.   :104.00   Max.   :174.00
```

Above table is the summary table for the time series formatted dataset.



We visualize the trend for the time series and we can see that there is a huge spike for the cases of the disease around 2020.

```
## [1] 2013   11
```

The start date of our time series dataset is 2013-11.

```
## [1] 2022    6
```

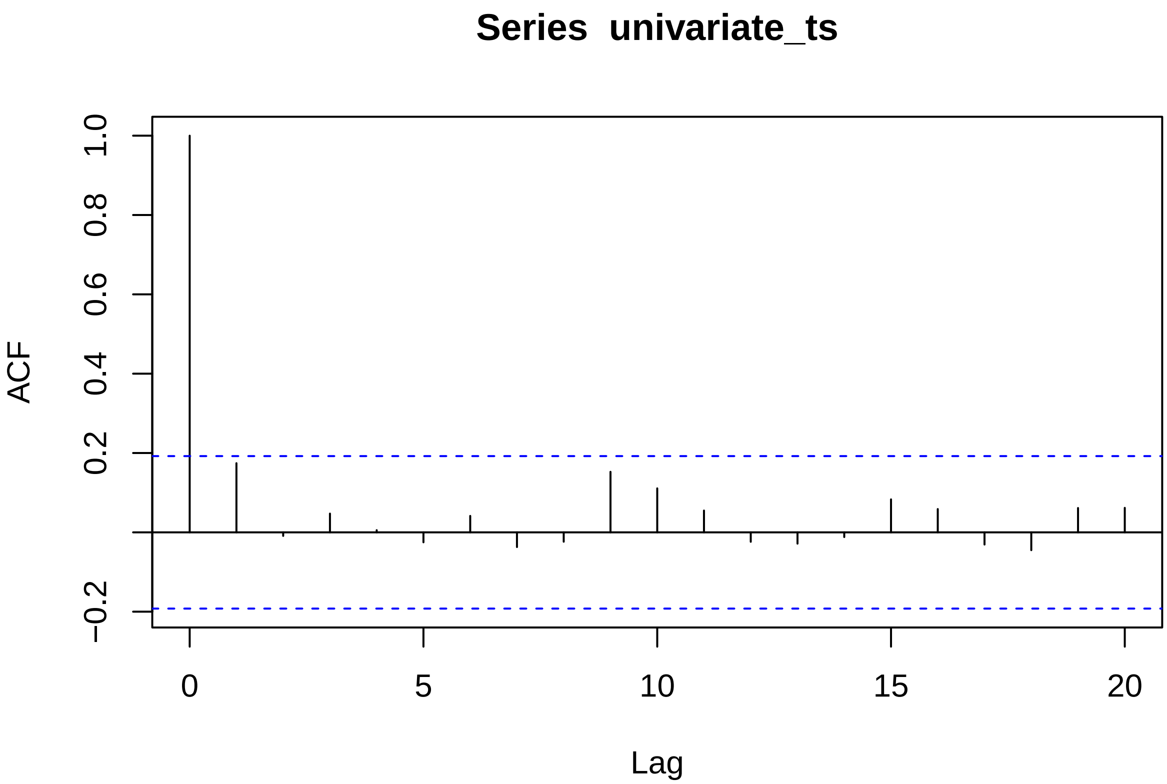
The end date of our time series dataset is 2022-06.

```
## [1] 12
```

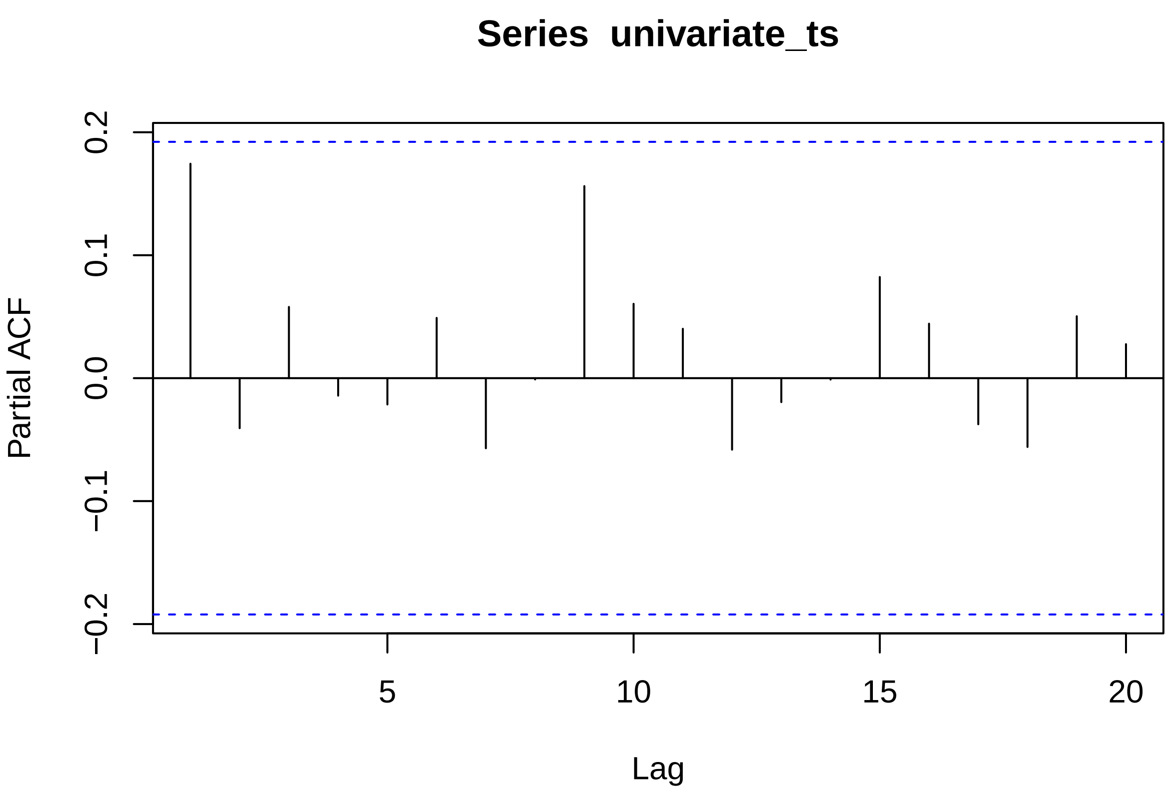
The cycle of our time series dataset is 12.

```
## Time Series:
## Start = 1
## End = 104
## Frequency = 1
##    [1]   2   0  20   0   4   0   7   9   8  34   2   9   3   6   4   3   2   4
##   [19]  22   6  18   4   1  44  25  30  34  32  16  35  19  13  22  38  10  11
##   [37]  26  11   8  14  36  54  18  52  18   7  12  35  16  14  21  29  29  31
##   [55]  35  41   5   9  36  14  10  21  12  19  32  13  11   2  17  15  11   6
##   [73]   8  43 174   8   2  10   1   2   7   2   6  44   1   6   6   4   0  28
##   [91]   7   8   4  10  14   4   7   5   9  16  15   3  21  14
```

We convert above time series dataframe to a univariate time series dataset.



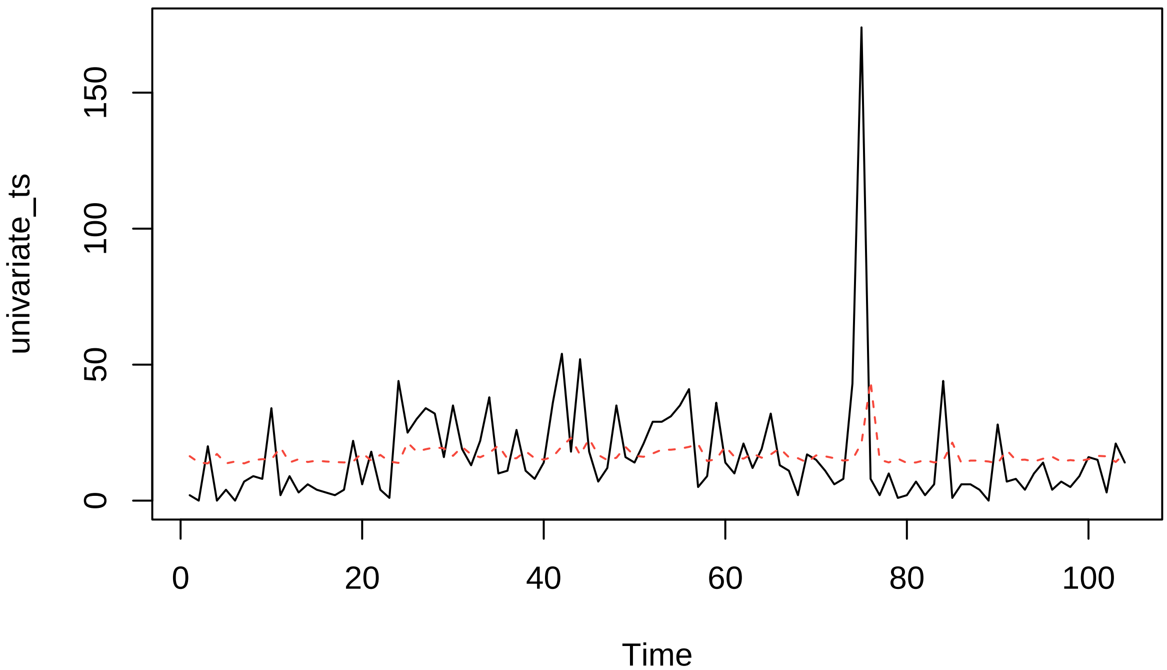
From the ACF graph, we should choose the order of the MA model to be 1.



From the PACF graph, we should choose the order of the AR model to be 0 since no band is significant.

AR Model

```
##
## Call:
## arima(x = univariate_ts, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.1737    16.5616
## s.e.  0.0963     2.3450
##
## sigma^2 estimated as 392:  log likelihood = -458.1,  aic = 922.19
```



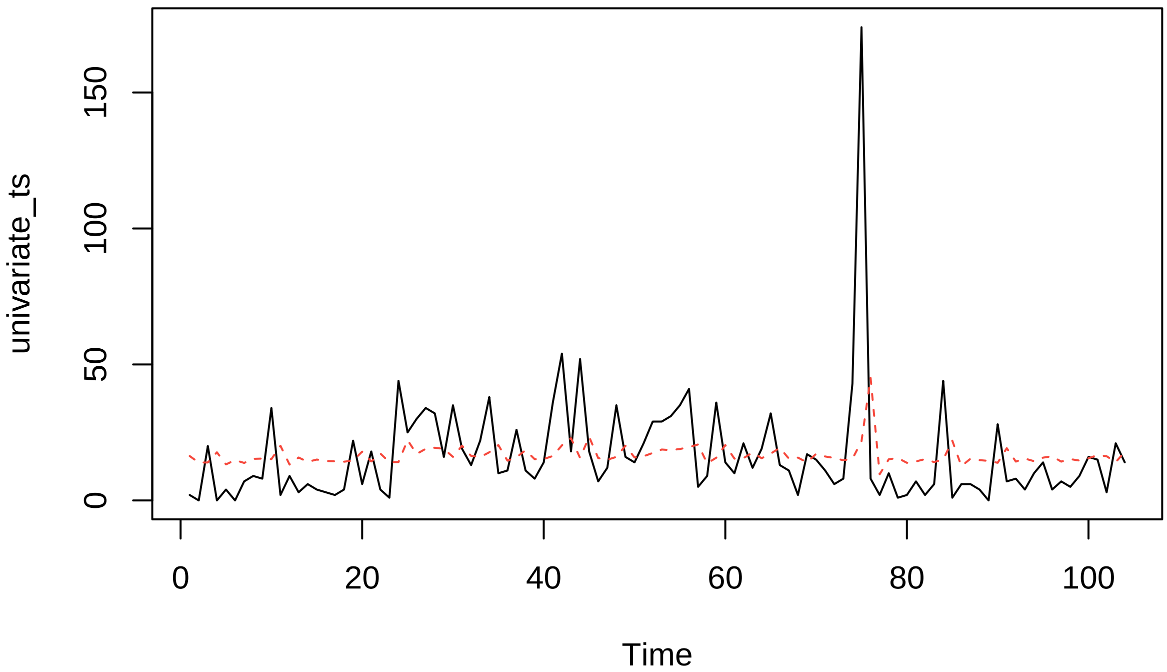

```
## [1] 922.1941
```

```
## [1] 930.1272
```

The AIC for the AR model is 922 and the BIC for the AR model is 930.

MA Model

```
##
## Call:
## arima(x = univariate_ts, order = c(0, 0, 1))
##
## Coefficients:
##          ma1  intercept
##       0.1873    16.5688
## s.e.  0.0992     2.2991
##
## sigma^2 estimated as 391.2:  log likelihood = -457.98,  aic = 921.97
```



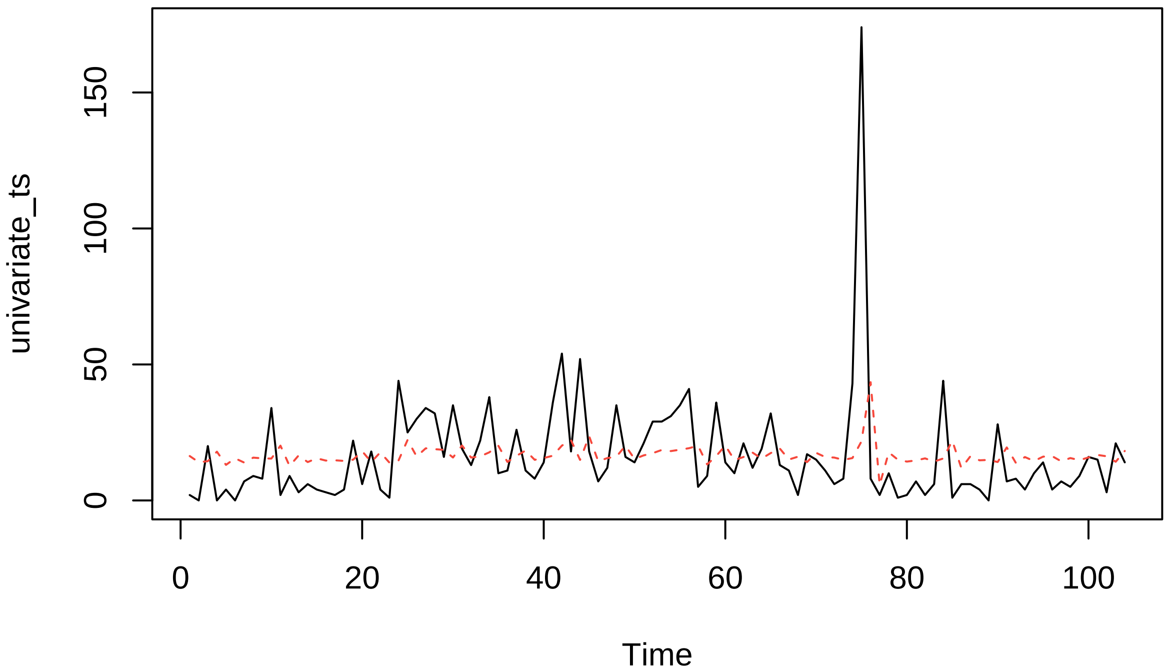

```
## [1] 921.9661
```

```
## [1] 929.8993
```

The AIC for the MA model is 921 and the BIC for the MA model is 929.

ARMA Model

```
##
## Call:
## arima(x = univariate_ts, order = c(1, 0, 1))
##
## Coefficients:
##           ar1     ma1  intercept
##       -0.1634  0.3457    16.5742
## s.e.   0.5014  0.4767     2.2400
##
## sigma^2 estimated as 390.8:  log likelihood = -457.94,  aic = 923.88
```



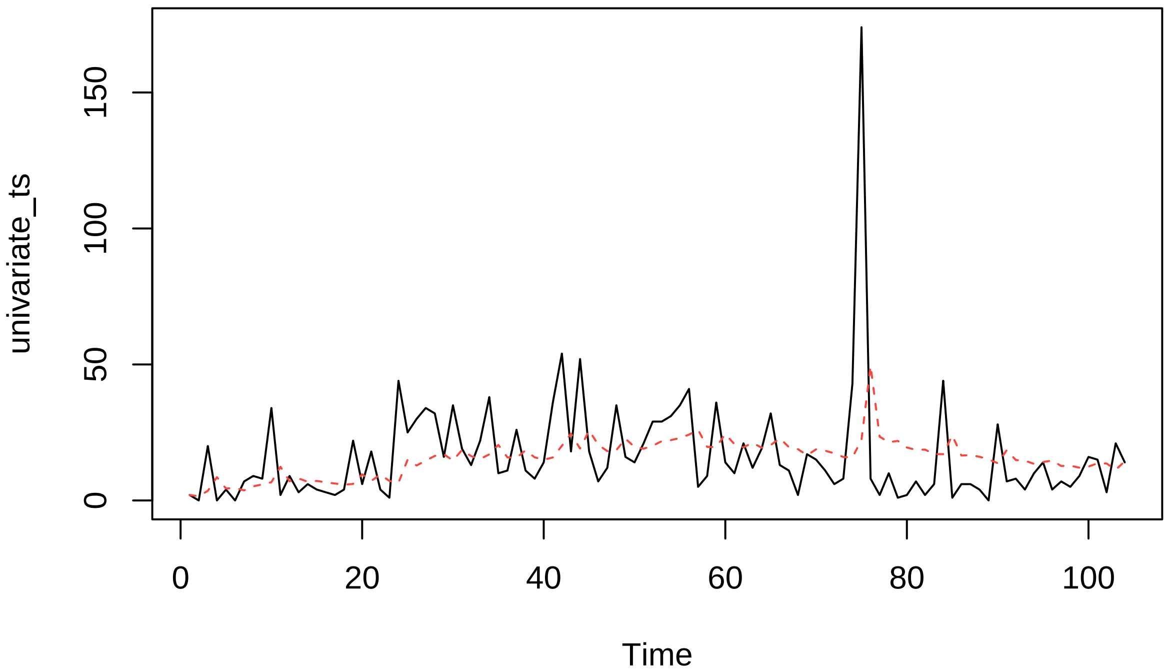

```
## [1] 923.8775
```

```
## [1] 934.455
```

The AIC for the ARMA model is 923 and the BIC for the ARMA model is 934.

ARIMA Model

```
##
## Call:
## arima(x = univariate_ts, order = c(1, 1, 1))
##
## Coefficients:
##          ar1      ma1
##       0.1432  -0.9445
## s.e.  0.1079   0.0481
##
## sigma^2 estimated as 403.2:  log likelihood = -456.1,  aic = 918.21
```




```
## [1] 918.2055
```

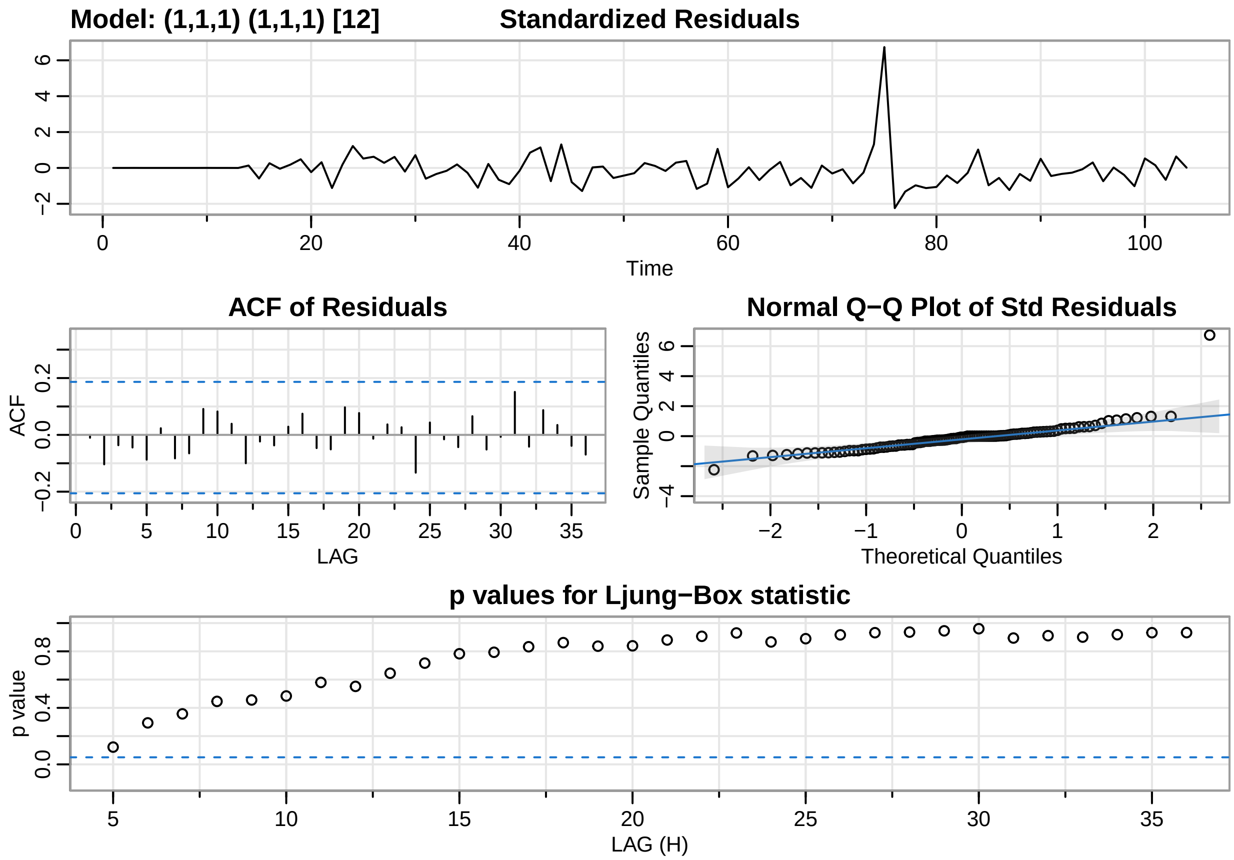
```
## [1] 926.1097
```

The AIC for the ARIMA model is 918 and the BIC for the ARIMA model is 926.

SARIMA Model

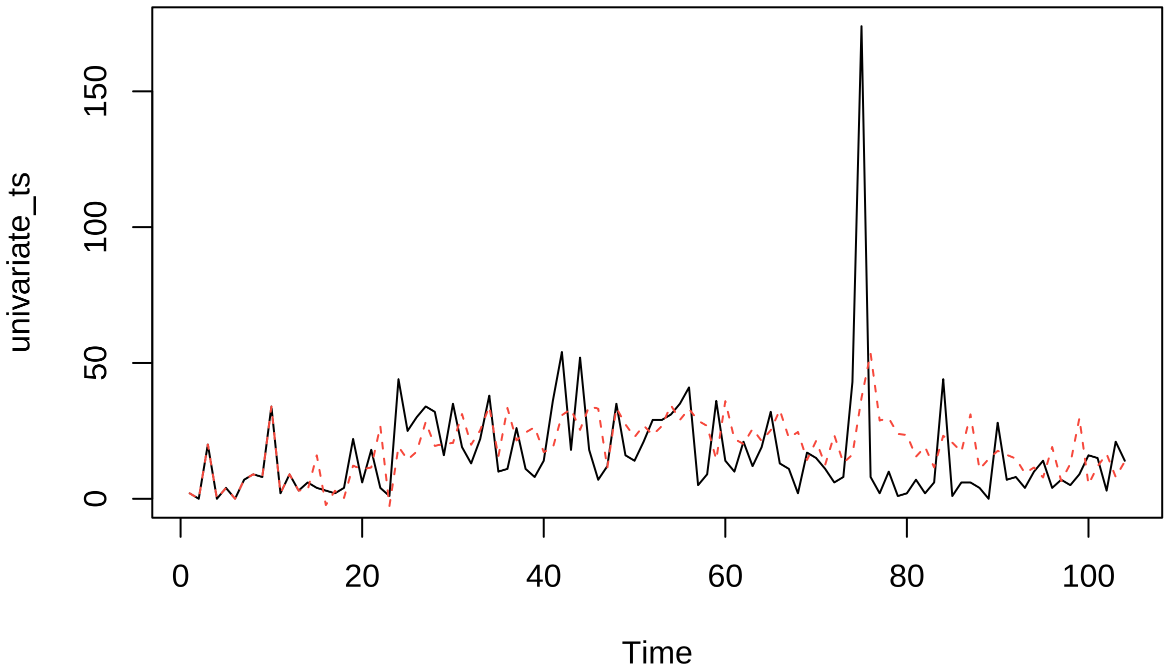
```
## initial  value 3.731718
## iter   2 value 3.395655
## iter   3 value 3.311211
## iter   4 value 3.303048
## iter   5 value 3.250697
## iter   6 value 3.233034
## iter   7 value 3.230637
## iter   8 value 3.230459
## iter   9 value 3.230257
## iter  10 value 3.230253
## iter  11 value 3.230253
## iter  11 value 3.230253
## iter  11 value 3.230253
## final  value 3.230253
## converged
## initial  value 3.204693
## iter   2 value 3.197900
## iter   3 value 3.176433
## iter   4 value 3.172718
## iter   5 value 3.171409
## iter   6 value 3.171303
## iter   7 value 3.171275
## iter   8 value 3.171275
## iter   8 value 3.171275
## iter   8 value 3.171275
```

```
## final  value 3.171275
## converged
```




```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ma1     sar1     sma1
##       0.1525  -0.9196  -0.0441  -1.0000
## s.e.  0.1114   0.0477   0.1093   0.1716
##
## sigma^2 estimated as 412.8:  log likelihood = -417.71,  aic = 845.42
##
## $degrees_of_freedom
## [1] 87
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1    0.1525 0.1114   1.3681  0.1748
## ma1   -0.9196 0.0477 -19.2934  0.0000
## sar1  -0.0441 0.1093  -0.4039  0.6873
## sma1  -1.0000 0.1716  -5.8277  0.0000
##
```

```
## $AIC
## [1] 9.290317
##
## $AICc
## [1] 9.295428
##
## $BIC
## [1] 9.428277
```



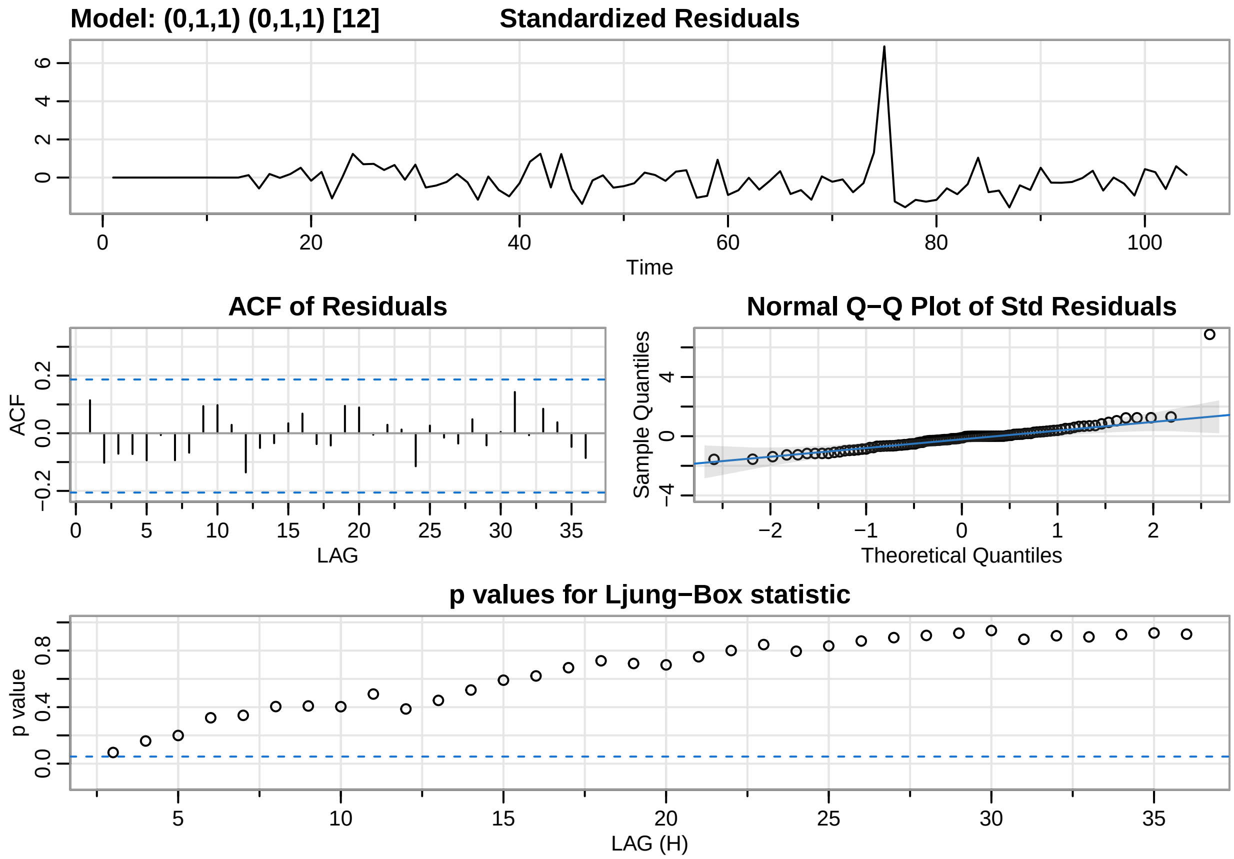

```
## [1] 9.290317
```

```
## [1] 9.428277
```

The AIC for the SARIMA model is 9.3 and the BIC for the SARIMA model is 9.4.

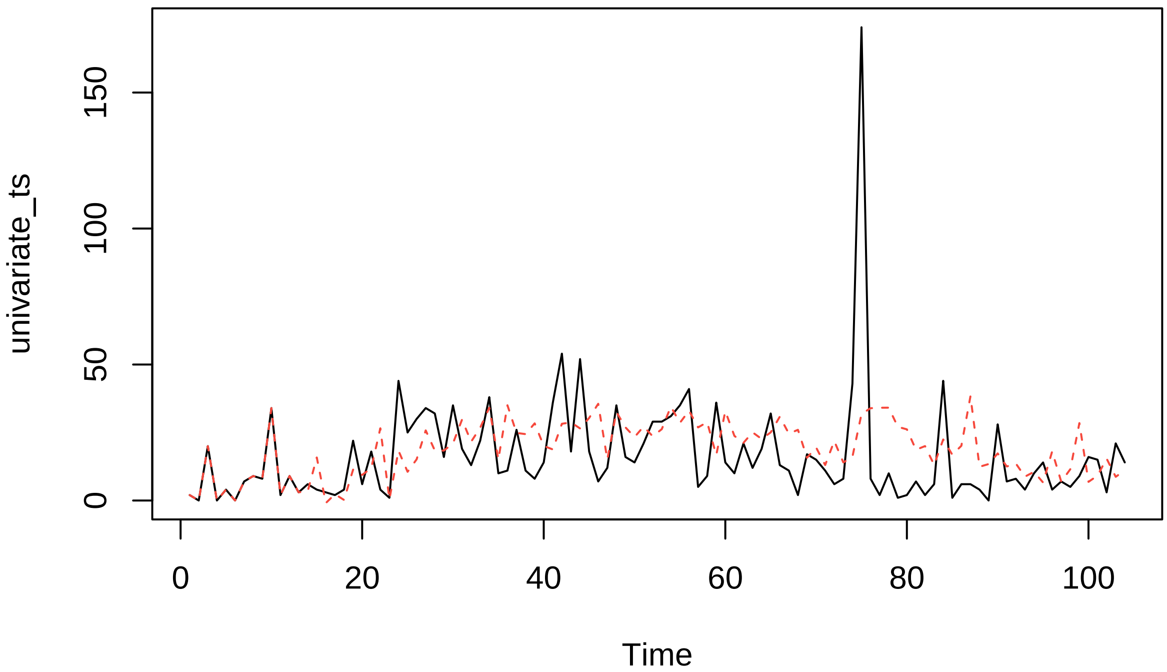
```
## initial  value 3.674311
## iter   2 value 3.366923
## iter   3 value 3.216442
## iter   4 value 3.159545
## iter   5 value 3.146283
## iter   6 value 3.135101
## iter   7 value 3.131381
## iter   8 value 3.126402
## iter   9 value 3.123548
## iter  10 value 3.123372
## iter  11 value 3.123148
## iter  12 value 3.123082
## iter  13 value 3.123081
## iter  14 value 3.123081
## iter  14 value 3.123081
## iter  14 value 3.123081
## final  value 3.123081
## converged
## initial  value 3.186360
```

```
## iter   2 value 3.183769
## iter   3 value 3.182835
## iter   4 value 3.182590
## iter   5 value 3.182587
## iter   5 value 3.182587
## iter   5 value 3.182587
## final  value 3.182587
## converged
```




```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     sma1
##       -0.8993  -1.0000
## s.e.   0.0502   0.1548
##
## sigma^2 estimated as 427.4:  log likelihood = -418.74,  aic = 843.48
##
## $degrees_of_freedom
## [1] 89
##
## $ttable
```

```
##      Estimate     SE  t.value p.value
## ma1   -0.8993 0.0502 -17.9182       0
## sma1  -1.0000 0.1548  -6.4616       0
##
## $AIC
## [1] 9.268985
##
## $AICc
## [1] 9.270483
##
## $BIC
## [1] 9.35176
```



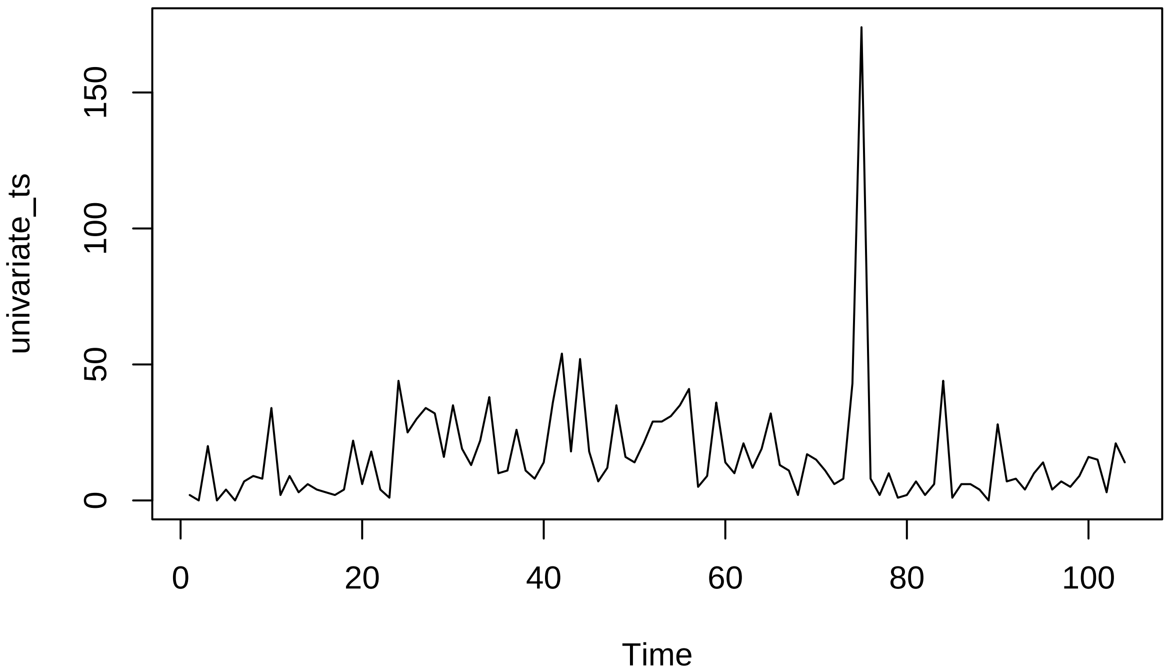

```
## [1] 9.268985
```

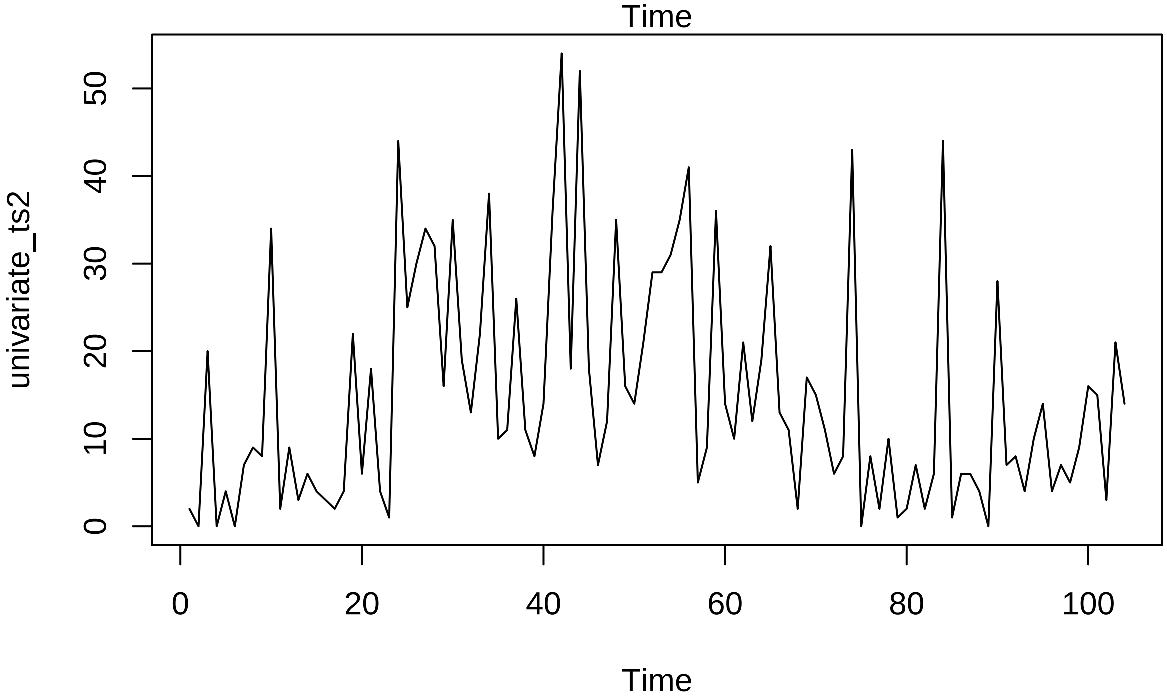
```
## [1] 9.35176
```

The AIC for the SARIMA model without AR part is 9.27. The BIC for the SARIMA model without AR part is 9.35.

SARIMA Model without outliers

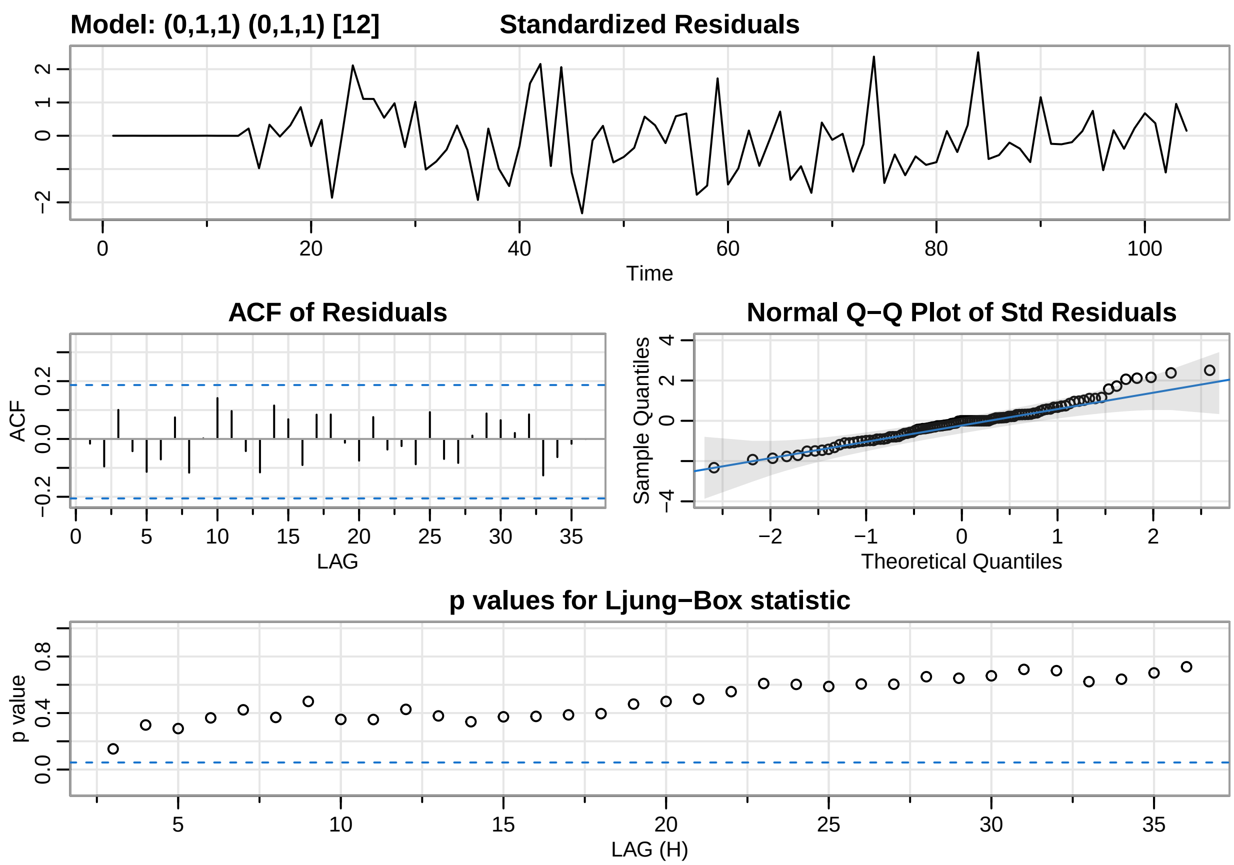
1. Replace the outlier with 0






```
## initial  value 3.163762
## iter   2 value 2.875234
## iter   3 value 2.696893
## iter   4 value 2.687958
## iter   5 value 2.686859
## iter   6 value 2.685899
## iter   7 value 2.685890
## iter   8 value 2.685888
## iter   9 value 2.685887
## iter   9 value 2.685887
## iter   9 value 2.685887
## final  value 2.685887
```

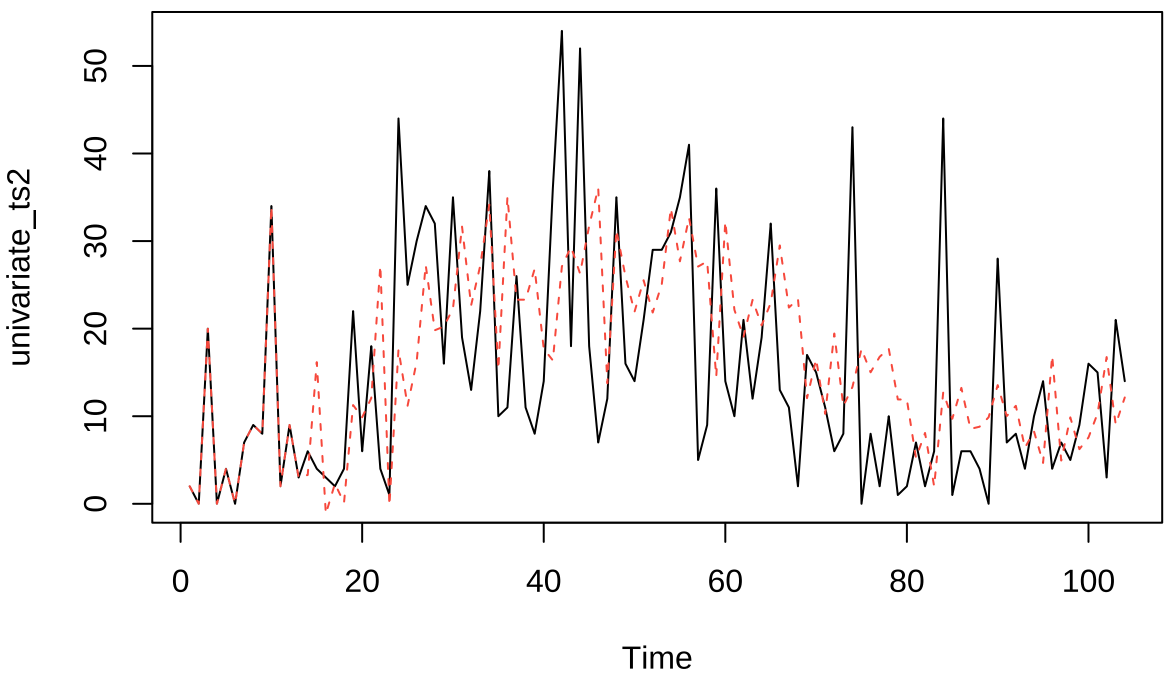
```
## converged
## initial  value 2.690780
## iter   2 value 2.685994
## iter   3 value 2.673721
## iter   4 value 2.673440
## iter   5 value 2.672888
## iter   6 value 2.672869
## iter   7 value 2.672869
## iter   8 value 2.672868
## iter   8 value 2.672868
## iter   8 value 2.672868
## final  value 2.672868
## converged
```



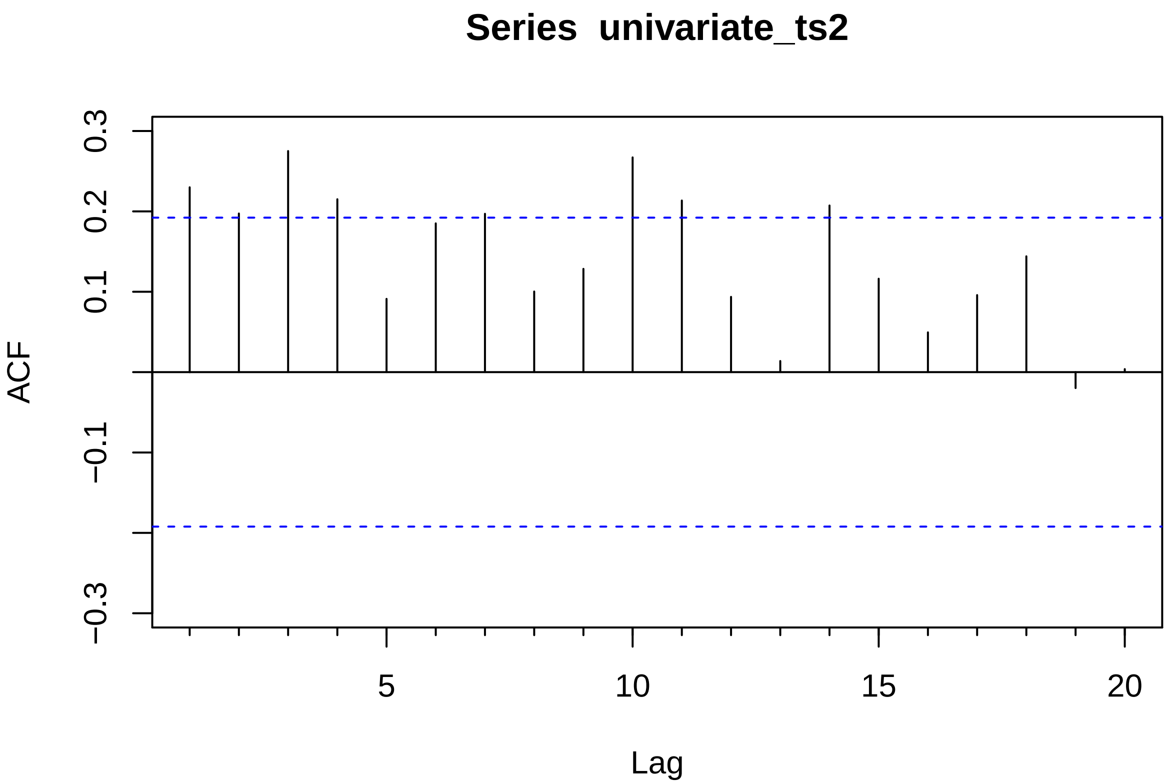

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     sma1
##       -0.8365  -1.0000
## s.e.   0.0545   0.1675
##
## sigma^2 estimated as 155.5:  log likelihood = -372.35,  aic = 750.71
```

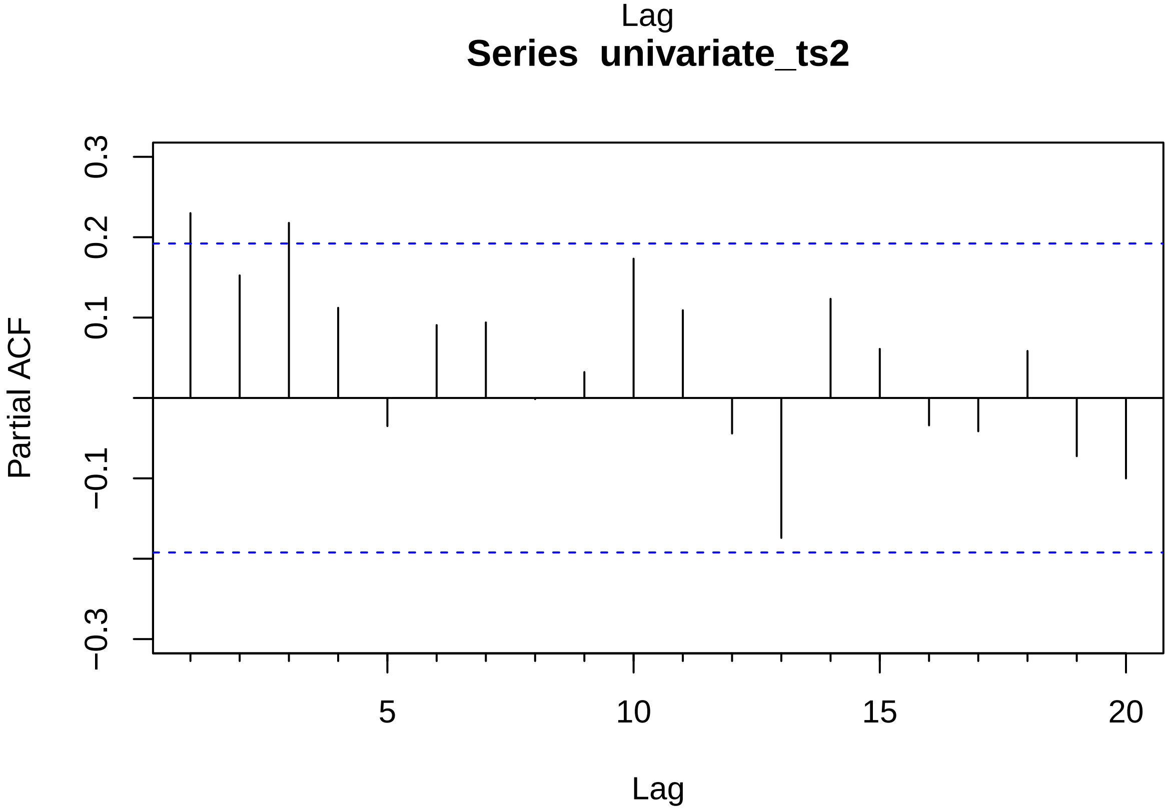
```
##
## $degrees_of_freedom
## [1] 89
##
## $ttable
##      Estimate     SE  t.value p.value
## ma1   -0.8365 0.0545 -15.3586       0
## sma1  -1.0000 0.1675  -5.9688       0
##
## $AIC
## [1] 8.249548
##
## $AICc
## [1] 8.251046
##
## $BIC
## [1] 8.332324

## [1] 8.249548

## [1] 8.332324
```

The AIC for this SARIMA model is 8.25. The BIC for this SARIMA model is 8.33.

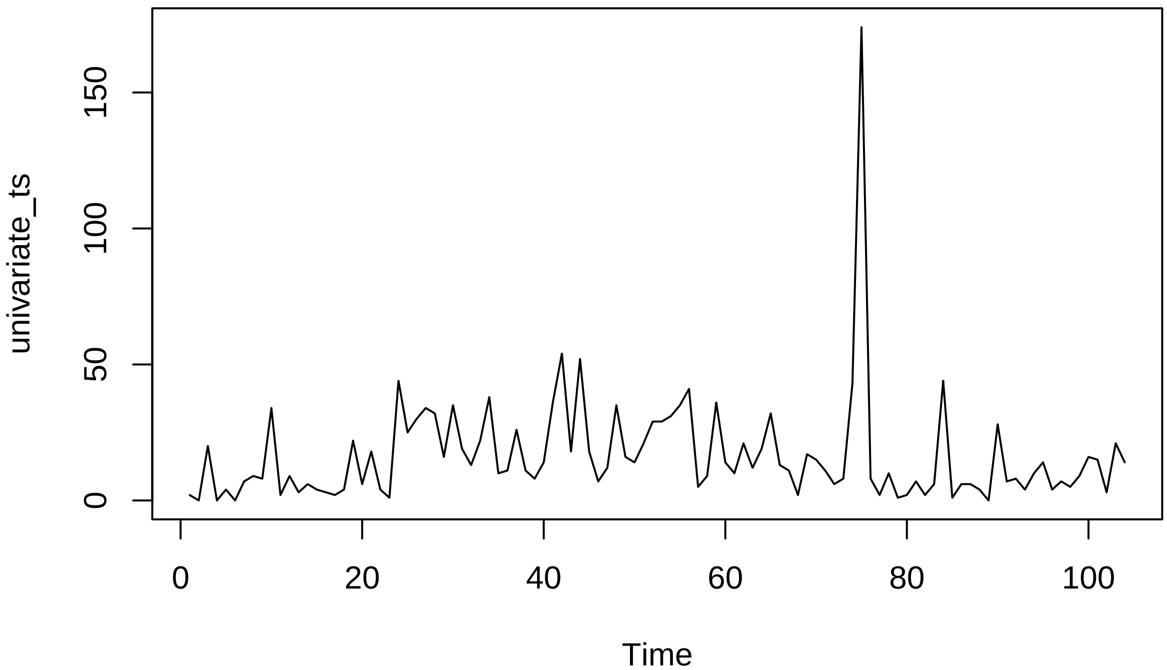


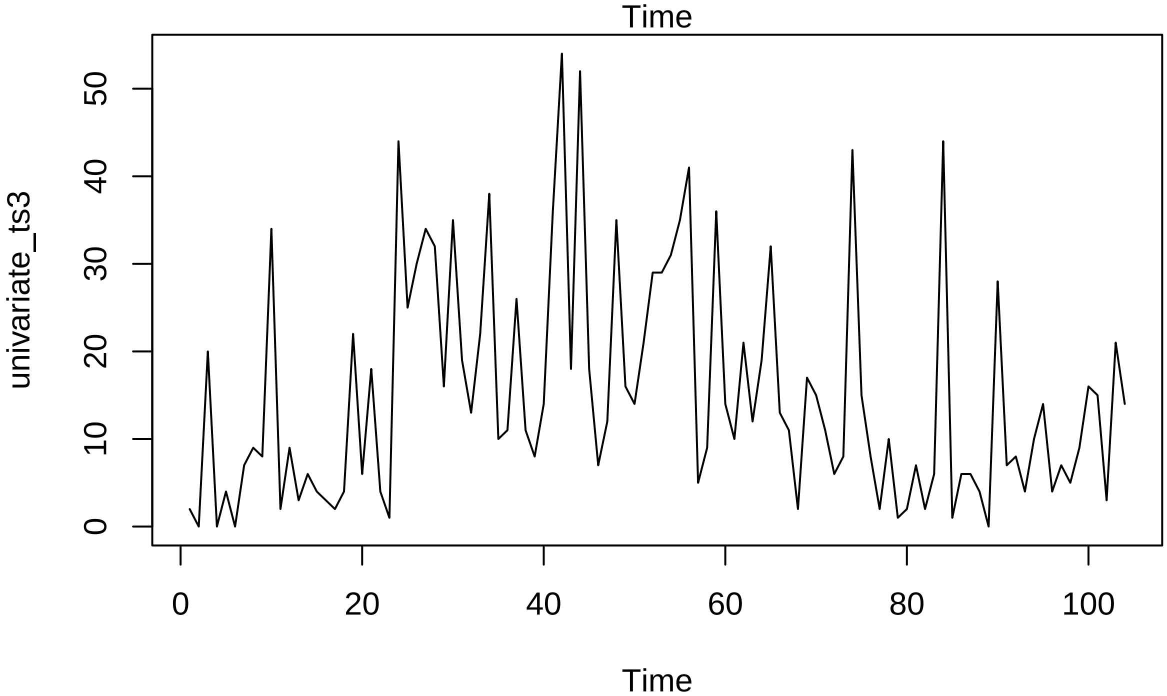
Above graph represents the time series along with the fitted values





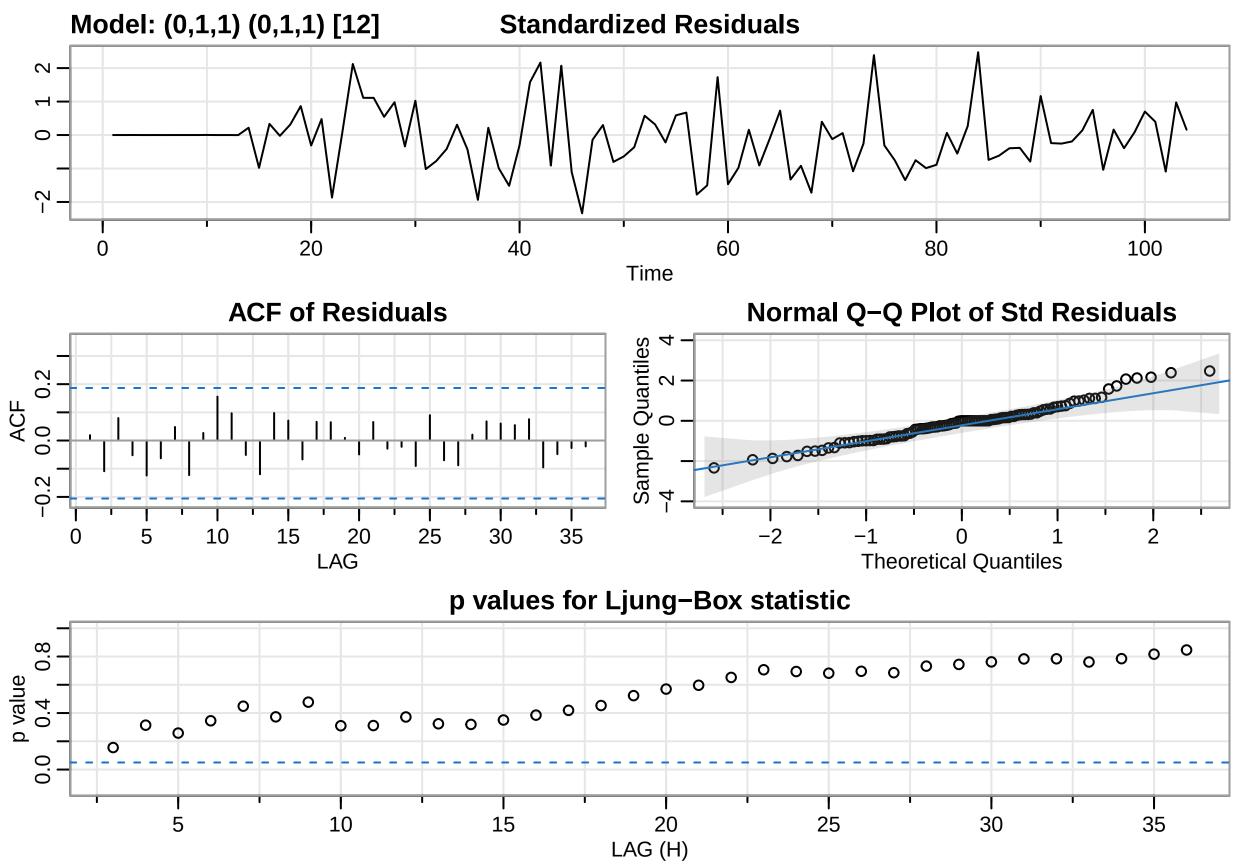
2. Replace the outlier with the mean value of frequency.






```
## initial  value 3.146385
## iter   2 value 2.876232
## iter   3 value 2.700753
## iter   4 value 2.687846
## iter   5 value 2.684698
## iter   6 value 2.682128
## iter   7 value 2.682120
## iter   8 value 2.682119
## iter   9 value 2.682119
## iter   9 value 2.682119
## iter   9 value 2.682119
## final  value 2.682119
```

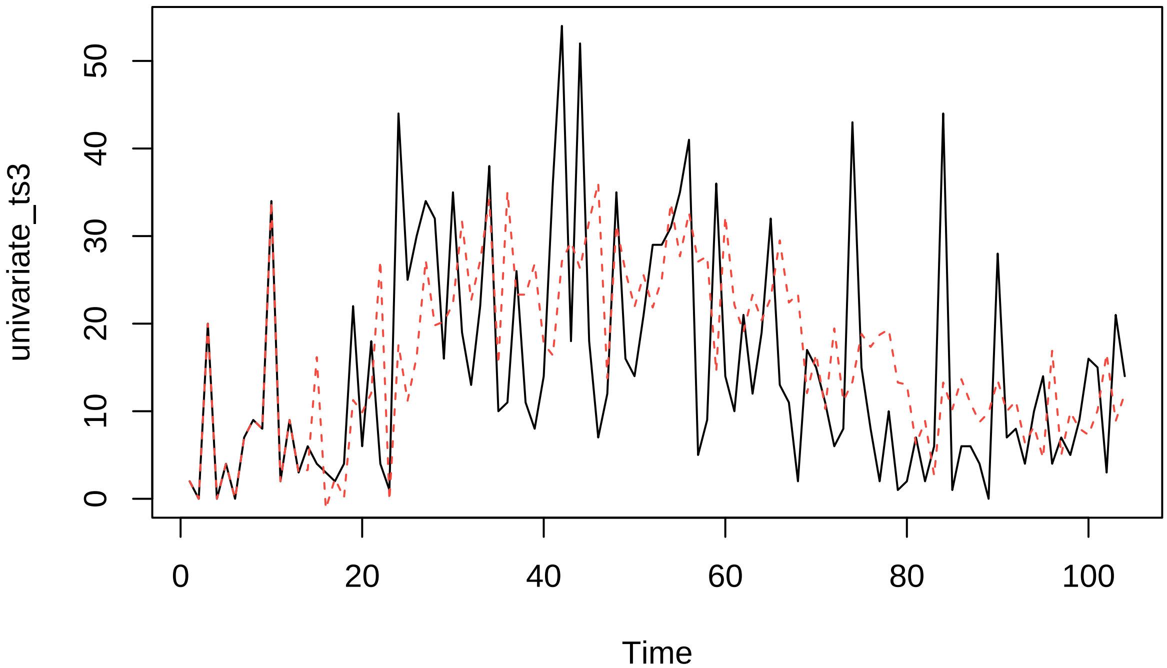
```
## converged
## initial  value 2.687041
## iter   2 value 2.681647
## iter   3 value 2.669783
## iter   4 value 2.669598
## iter   5 value 2.668827
## iter   6 value 2.668792
## iter   7 value 2.668791
## iter   8 value 2.668791
## iter   8 value 2.668791
## iter   8 value 2.668791
## final  value 2.668791
## converged
```



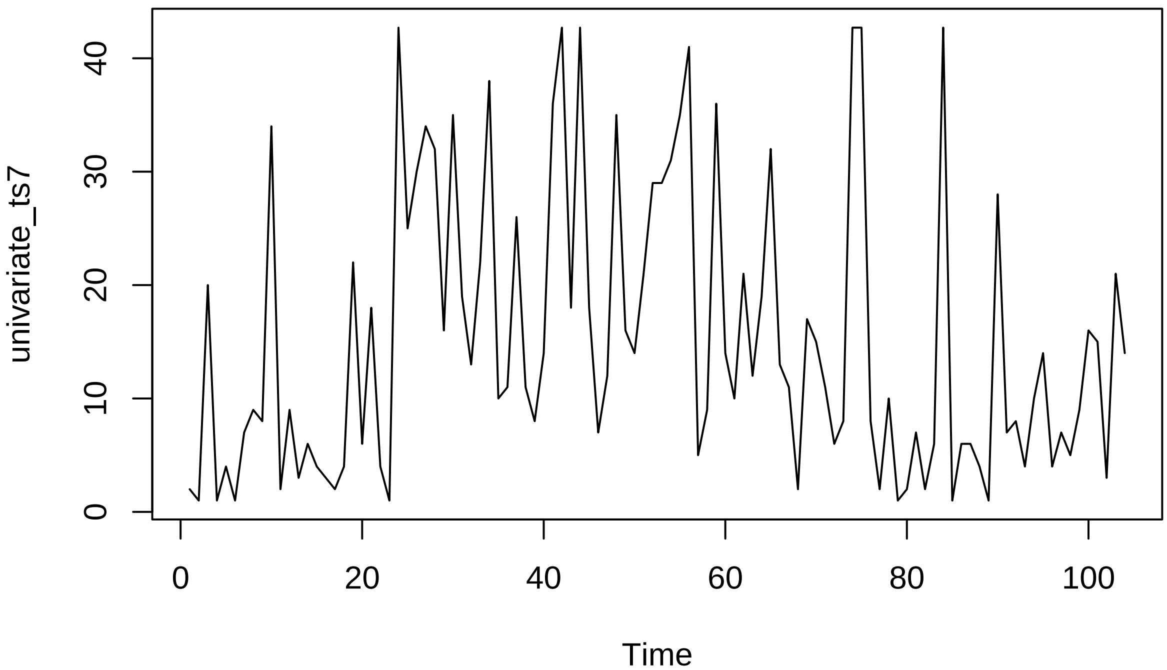

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1    sma1
##       -0.8367  -1.000
## s.e.   0.0555   0.158
##
## sigma^2 estimated as 154.3:  log likelihood = -371.98,  aic = 749.97
```

```
##
## $degrees_of_freedom
## [1] 89
##
## $ttable
##      Estimate     SE  t.value p.value
## ma1   -0.8367 0.0555 -15.0644       0
## sma1  -1.0000 0.1580  -6.3294       0
##
## $AIC
## [1] 8.241393
##
## $AICc
## [1] 8.242891
##
## $BIC
## [1] 8.324168

## [1] 8.241393

## [1] 8.324168
```

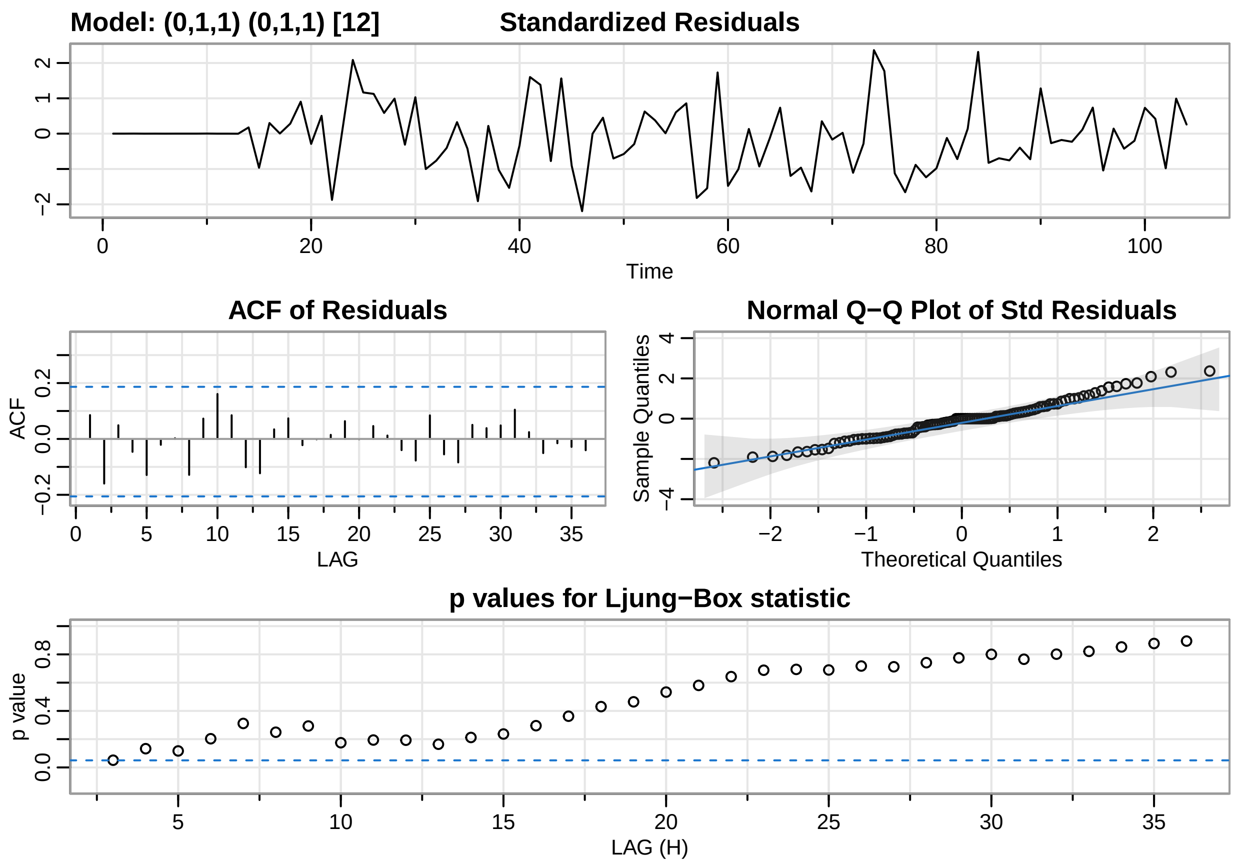
The AIC for this SARIMA model is 8.24. The BIC for this SARIMA model is 8.32.



Winsorization for monthly data




```
## initial  value 3.121770
## iter   2 value 2.863193
## iter   3 value 2.709352
## iter   4 value 2.693686
## iter   5 value 2.671281
## iter   6 value 2.665684
## iter   7 value 2.665145
## iter   8 value 2.665129
## iter   9 value 2.665123
## iter   9 value 2.665123
## iter   9 value 2.665123
## final  value 2.665123
## converged
## initial  value 2.672845
## iter   2 value 2.667456
## iter   3 value 2.656637
## iter   4 value 2.655891
## iter   5 value 2.655435
## iter   6 value 2.655377
## iter   7 value 2.655376
## iter   8 value 2.655375
## iter   8 value 2.655375
## iter   8 value 2.655375
## final  value 2.655375
## converged
```



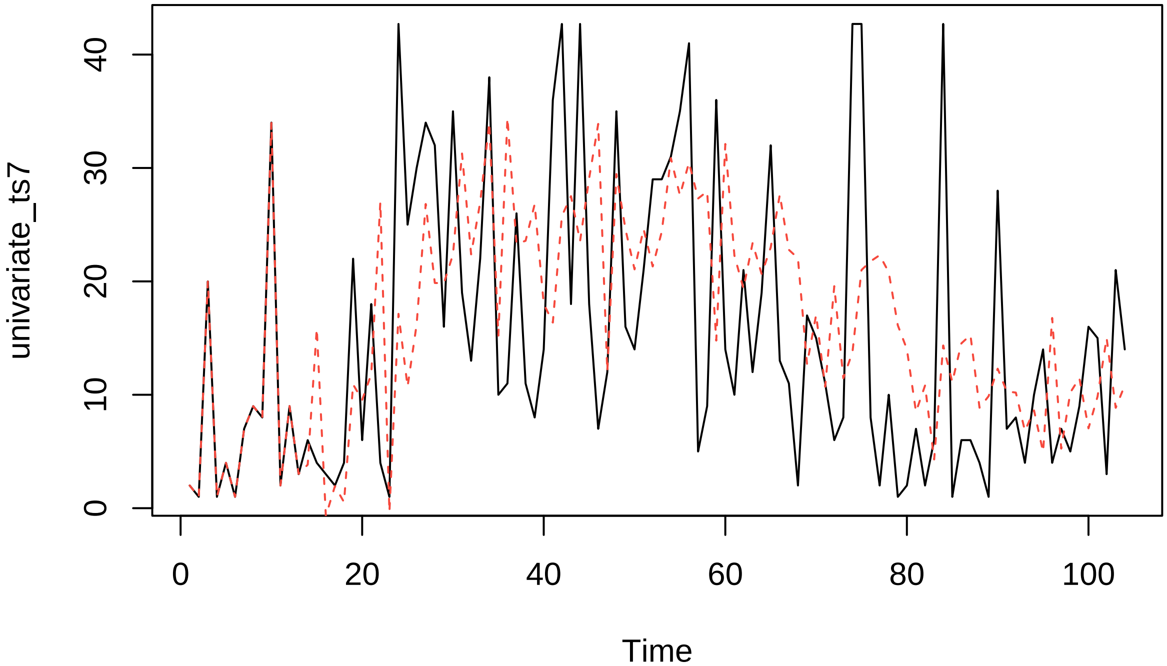

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     sma1
##       -0.8399  -1.0000
## s.e.   0.0581   0.1457
##
## sigma^2 estimated as 150.1:  log likelihood = -370.76,  aic = 747.53
##
## $degrees_of_freedom
## [1] 89
##
## $ttable
##      Estimate     SE  t.value p.value
## ma1   -0.8399 0.0581 -14.4434       0
## sma1  -1.0000 0.1457  -6.8650       0
##
## $AIC
## [1] 8.214562
##
## $AICc
```

```
## [1] 8.21606
##
## $BIC
## [1] 8.297337
```

```
## [1] 8.214562
```

```
## [1] 8.297337
```

The AIC for this SARIMA model is 8.2. The BIC for this SARIMA model is 8.3.



Above graph represents the time series along with the fitted values

Using Weekly Data to Model the Time Series

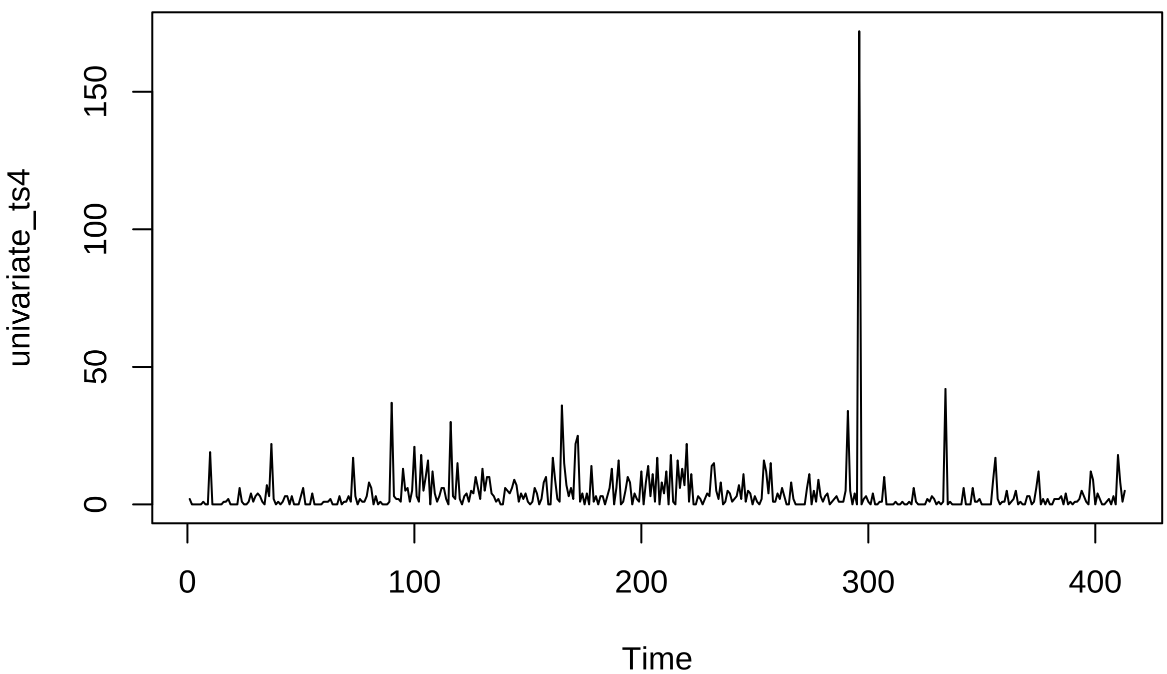
```
##    Date(YMW) Frequency
## 1  2013-11-3         2
## 2  2013-11-4         0
## 3  2013-12-1         0
## 4  2013-12-2         0
## 5  2013-12-3         0
## 6  2013-12-4         0
## 7  2014-01-1         1
## 8  2014-01-2         0
## 9  2014-01-3         0
## 10 2014-01-4        19
```

For each month, I coded date 1 to date 7 as the first week; date 8 to date 14 as the second week; date 15 to date 21 as the third week; and the rest of the days within each month as the fourth week.

Model with the outlier

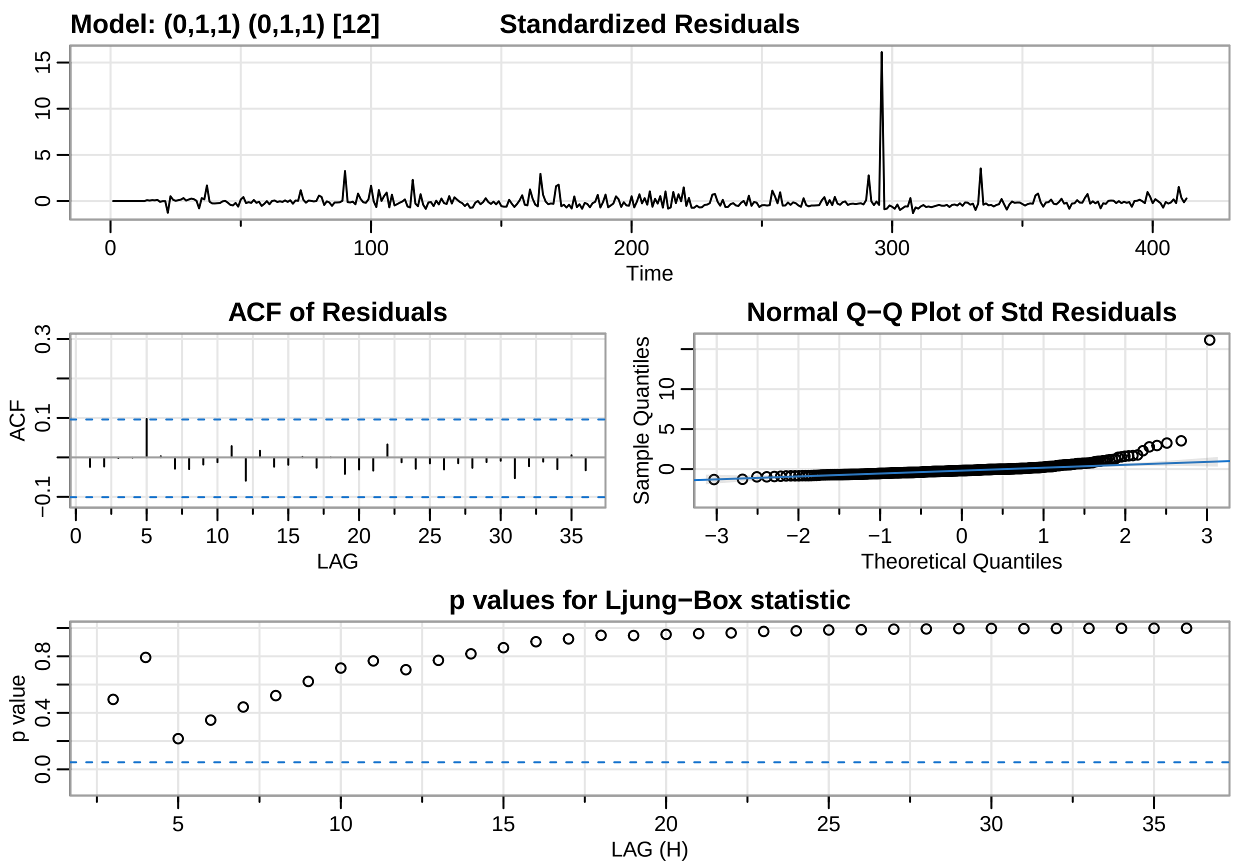
```
## Time Series:
## Start = 1
```

```
## End = 413
## Frequency = 1
##   [1]   2   0   0   0   0   0   1   0   0  19   0   0   0   0   0   1   1   2
##  [19]   0   0   0   0   6   1   0   0   1   4   1   3   4   3   1   0   7   3
##  [37]  22   2   0   1   0   1   3   3   0   3   0   0   0   3   6   0   0   0
##  [55]   4   0   0   0   0   1   1   1   2   0   0   0   3   0   1   1   3   1
##  [73]  17   3   0   2   1   1   3   8   6   0   3   0   1   0   0   0   1  37
##  [91]   3   2   2   1  13   5   6   1   5  21   3   1  18   5  10  16   0  12
## [109]   4   1   3   6   6   2   0  30   3   2  15   2   0   3   4   1   5   4
## [127]  10   6   2  13   5  10  10   4   3   1   2   0   0   6   5   4   6   9
## [145]   7   1   4   2   4   1   0   1   6   4   0   2   8  10   0   0  17   9
## [163]   2   1  36  15   7   3   6   2  22  25   1   4   0   4   0  14   1   3
## [181]   0   3   3   0   3   6  13   0   6  16   0   1   5  10   8   0   4   2
## [199]   1  12   0   8  14   3  11   1  17   0   8   4  12   0  18   1   0  16
## [217]   6  13   7  22   1  11   0   0   3   2   0   2   4   3  14  15   5   2
## [235]   8   0   1   5   4   1   2   3   7   2  11   1   5   4   0   3   1   0
## [253]   2  16  12   4  15   1   1   4   2   6   3   0   0   8   2   0   0   0
## [271]   0   0   6  11   0   5   1   9   3   1   3   4   0   1   2   3   1   1
## [289]   1   5  34   5   0   4   0 172   0   2   3   1   0   4   0   0   1   1
## [307]  10   0   0   0   0   1   0   0   1   0   0   1   0   6   1   0   0   0
## [325]   0   2   1   3   2   0   1   0   1  42   0   1   0   0   0   0   0   6
## [343]   0   0   0   6   1   1   2   0   0   0   0   0   9  17   2   0   1   1
## [361]   5   0   1   2   5   0   1   0   0   3   3   0   1   6  12   0   2   0
## [379]   2   0   0   2   2   2   3   0   4   0   1   0   1   1   2   5   3   1
## [397]   0  12   9   0   4   2   0   0   1   2   0   3   0  18   8   1   5
```




```
## initial  value 3.050683
## iter   2 value 2.609056
## iter   3 value 2.480297
## iter   4 value 2.387615
## iter   5 value 2.386751
## iter   6 value 2.386708
## iter   7 value 2.386680
```

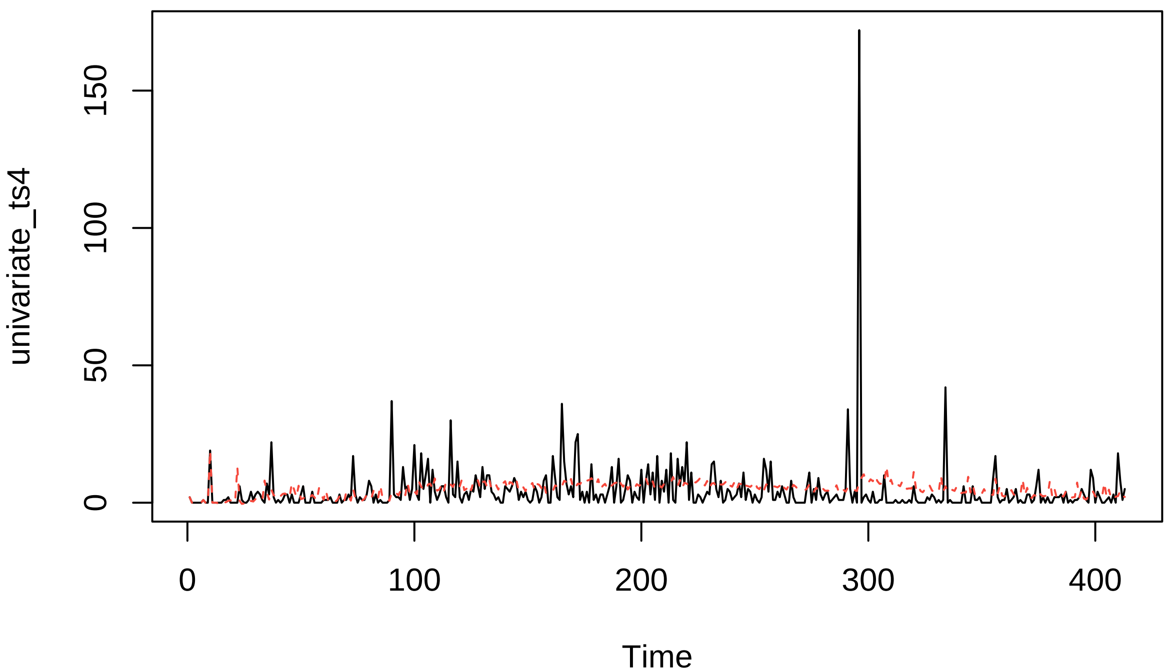
```
## iter   8 value 2.386679
## iter   9 value 2.386672
## iter  10 value 2.386672
## iter  10 value 2.386672
## iter  10 value 2.386672
## final  value 2.386672
## converged
## initial  value 2.390980
## iter   2 value 2.382503
## iter   3 value 2.381522
## iter   4 value 2.381042
## iter   5 value 2.380764
## iter   6 value 2.380744
## iter   7 value 2.380744
## iter   8 value 2.380744
## iter   8 value 2.380744
## iter   8 value 2.380744
## final  value 2.380744
## converged
```




```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
```

```
## Coefficients:
##           ma1     sma1
##       -0.9762  -1.0000
## s.e.   0.0126   0.0533
##
## sigma^2 estimated as 103.7:  log likelihood = -1519.87,  aic = 3045.75
##
## $degrees_of_freedom
## [1] 398
##
## $ttable
##      Estimate     SE t.value p.value
## ma1   -0.9762 0.0126 -77.304       0
## sma1  -1.0000 0.0533 -18.773       0
##
## $AIC
## [1] 7.614364
##
## $AICc
## [1] 7.61444
##
## $BIC
## [1] 7.6443

## [1] 7.614364

## [1] 7.6443
```

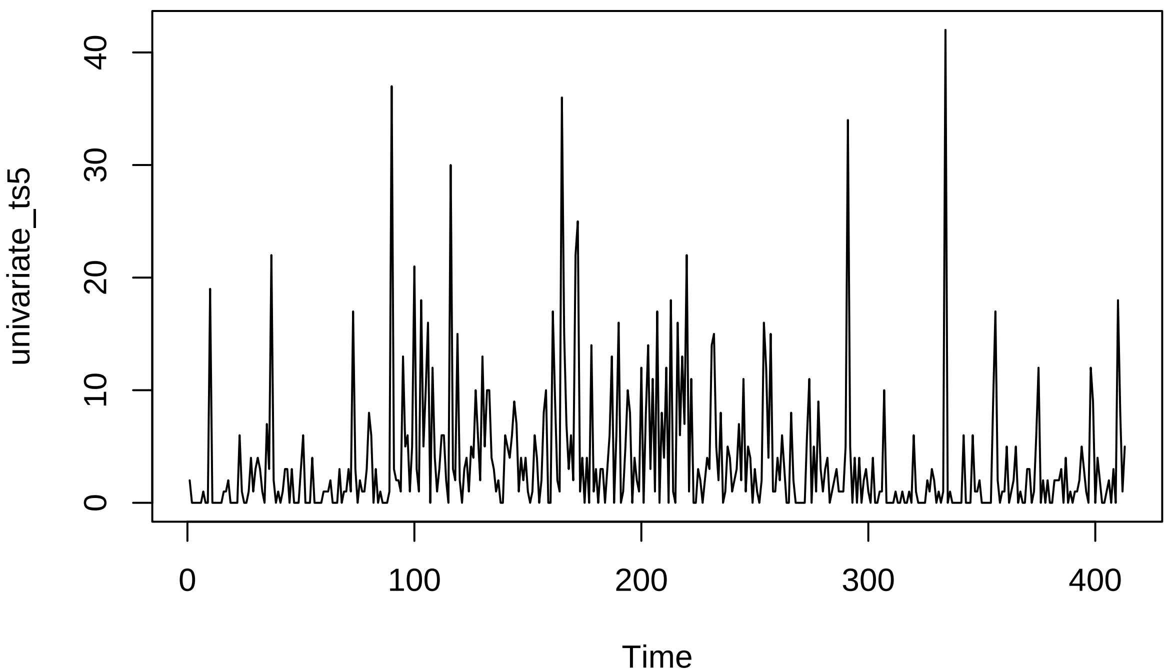
The AIC for this SARIMA model is 7.61. The BIC for this SARIMA model is 7.64.



Model without the outlier (replace with the mean value)

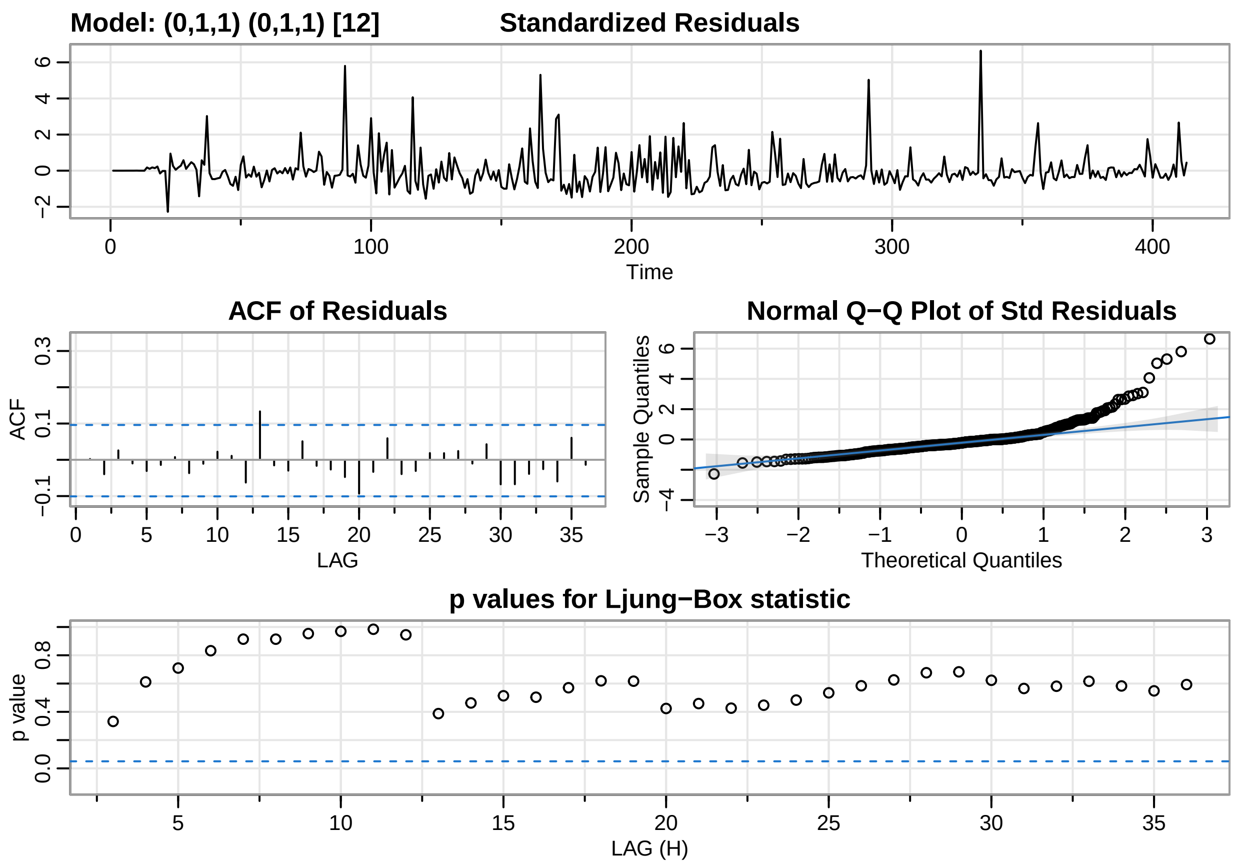
```
## Time Series:
## Start = 1
```

```
## End = 413
## Frequency = 1
##   [1]  2  0  0  0  0  0  1  0  0 19  0  0  0  0  0  1  1  2  0  0  0  0  6  1  0
##  [26]  0  1  4  1  3  4  3  1  0  7  3 22  2  0  1  0  1  3  3  0  3  0  0  0  3
##  [51]  6  0  0  0  4  0  0  0  0  1  1  1  2  0  0  0  3  0  1  1  3  1 17  3  0
##  [76]  2  1  1  3  8  6  0  3  0  1  0  0  0  1 37  3  2  2  1 13  5  6  1  5 21
## [101]  3  1 18  5 10 16  0 12  4  1  3  6  6  2  0 30  3  2 15  2  0  3  4  1  5
## [126]  4 10  6  2 13  5 10 10  4  3  1  2  0  0  6  5  4  6  9  7  1  4  2  4  1
## [151]  0  1  6  4  0  2  8 10  0  0 17  9  2  1 36 15  7  3  6  2 22 25  1  4  0
## [176]  4  0 14  1  3  0  3  3  0  3  6 13  0  6 16  0  1  5 10  8  0  4  2  1 12
## [201]  0  8 14  3 11  1 17  0  8  4 12  0 18  1  0 16  6 13  7 22  1 11  0  0  3
## [226]  2  0  2  4  3 14 15  5  2  8  0  1  5  4  1  2  3  7  2 11  1  5  4  0  3
## [251]  1  0  2 16 12  4 15  1  1  4  2  6  3  0  0  8  2  0  0  0  0  0  6 11  0
## [276]  5  1  9  3  1  3  4  0  1  2  3  1  1  1  5 34  5  0  4  0  4  0  2  3  1
## [301]  0  4  0  0  1  1 10  0  0  0  0  1  0  0  1  0  0  1  0  6  1  0  0  0  0
## [326]  2  1  3  2  0  1  0  1 42  0  1  0  0  0  0  0  6  0  0  0  6  1  1  2  0
## [351]  0  0  0  0  9 17  2  0  1  1  5  0  1  2  5  0  1  0  0  3  3  0  1  6 12
## [376]  0  2  0  2  0  0  2  2  2  3  0  4  0  1  0  1  1  2  5  3  1  0 12  9  0
## [401]  4  2  0  0  1  2  0  3  0 18  8  1  5
```




```
## initial  value 2.479276
## iter   2 value 2.031936
## iter   3 value 1.915223
## iter   4 value 1.850342
## iter   5 value 1.847233
## iter   6 value 1.845933
## iter   7 value 1.843551
## iter   8 value 1.843486
## iter   9 value 1.843472
## iter  10 value 1.843445
## iter  11 value 1.843442
## iter  11 value 1.843442
## final  value 1.843442
```

```
## converged
## initial  value 1.824096
## iter   2 value 1.815507
## iter   3 value 1.812237
## iter   4 value 1.808218
## iter   5 value 1.807888
## iter   6 value 1.807873
## iter   7 value 1.807872
## iter   8 value 1.807871
## iter   9 value 1.807870
## iter   9 value 1.807870
## iter   9 value 1.807870
## final  value 1.807870
## converged
```



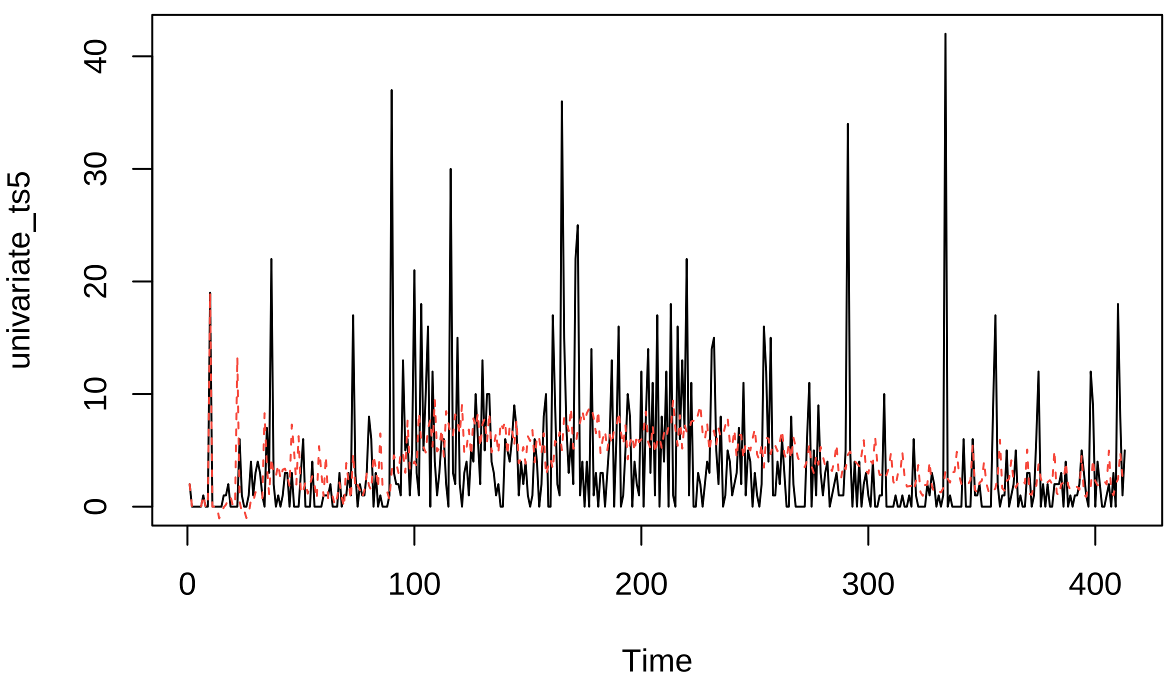

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     sma1
##       -0.9571  -0.9638
## s.e.   0.0150   0.0465
##
```

```
## sigma^2 estimated as 34.08:  log likelihood = -1290.72,  aic = 2587.45
##
## $degrees_of_freedom
## [1] 398
##
## $ttable
##      Estimate     SE  t.value p.value
## ma1   -0.9571 0.0150 -63.6395       0
## sma1  -0.9638 0.0465 -20.7387       0
##
## $AIC
## [1] 6.468617
##
## $AICc
## [1] 6.468693
##
## $BIC
## [1] 6.498553
```

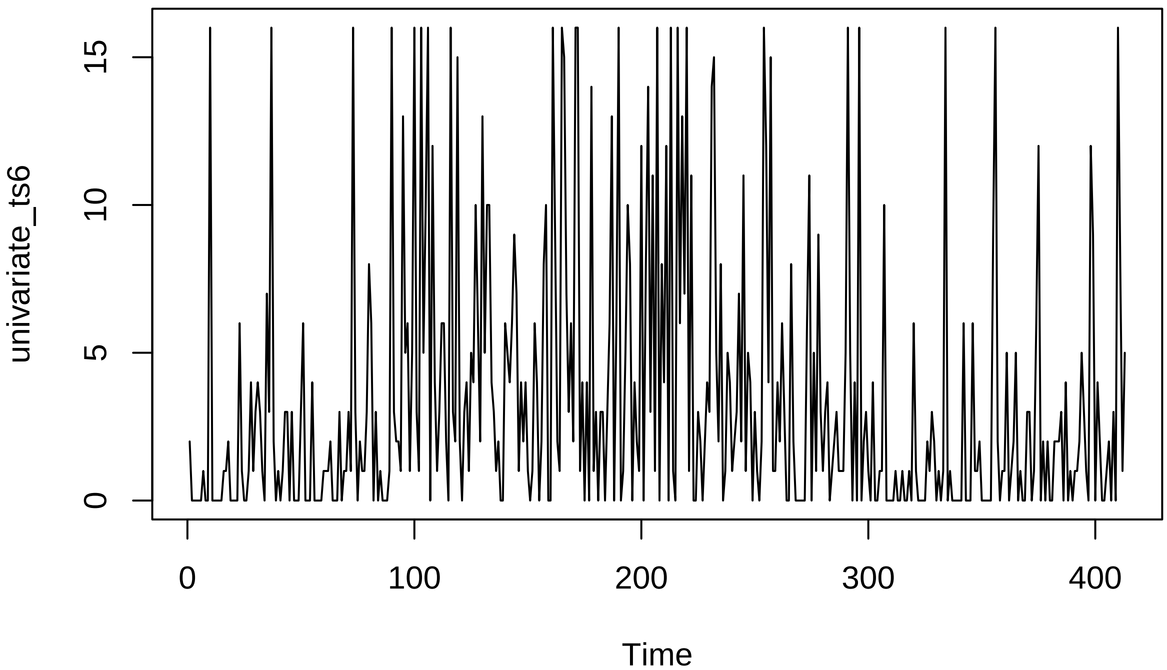
```
## [1] 6.468617
```

```
## [1] 6.498553
```

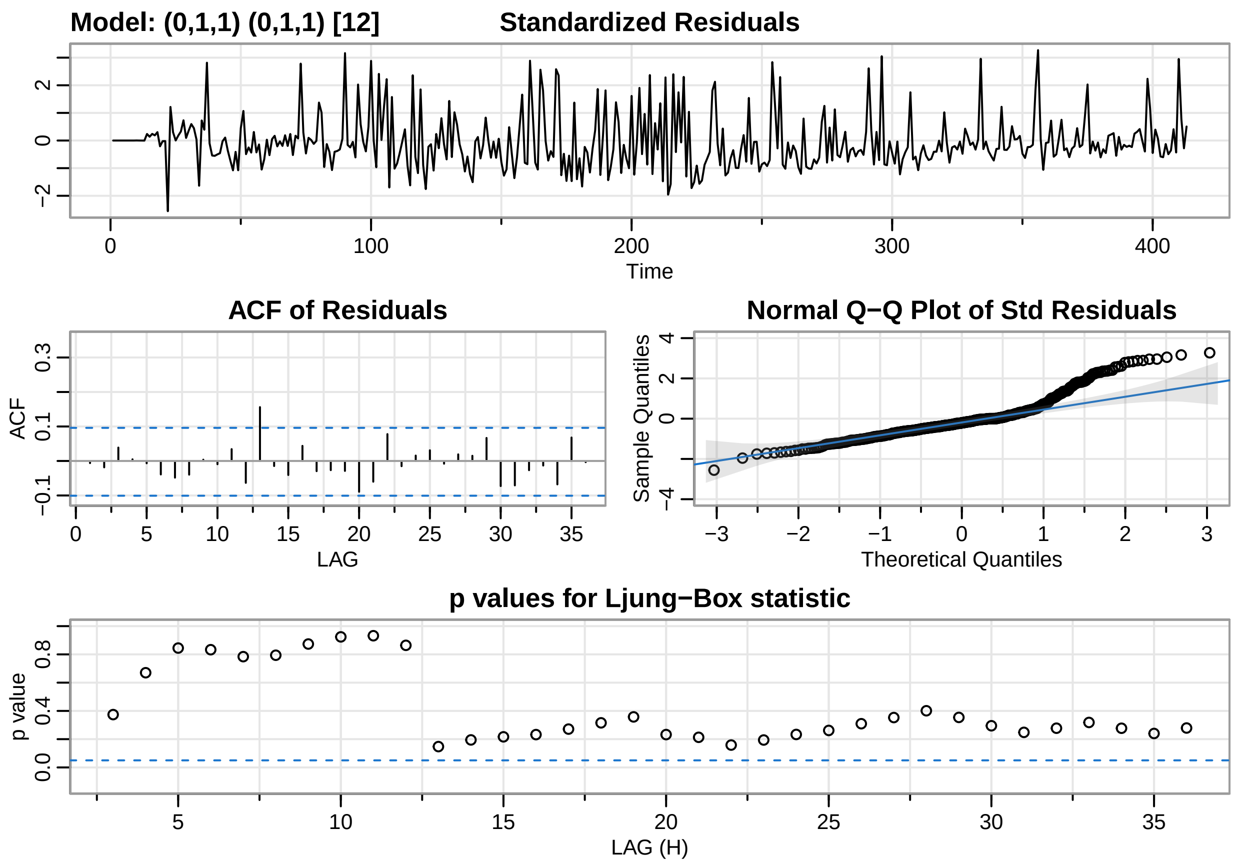
The AIC for this SARIMA model is 6.47. The BIC for this SARIMA model is 6.5.



Winsorization for weekly data




```
## initial  value 2.206445
## iter   2 value 1.727027
## iter   3 value 1.630805
## iter   4 value 1.582020
## iter   5 value 1.571189
## iter   6 value 1.570868
## iter   7 value 1.570807
## iter   8 value 1.570780
## iter   9 value 1.570770
## iter   9 value 1.570770
## iter   9 value 1.570770
## final  value 1.570770
## converged
## initial  value 1.546623
## iter   2 value 1.531170
## iter   3 value 1.529688
## iter   4 value 1.529372
## iter   5 value 1.529162
## iter   6 value 1.529131
## iter   7 value 1.529129
## iter   8 value 1.529129
## iter   9 value 1.529129
## iter   9 value 1.529129
## iter   9 value 1.529129
## final  value 1.529129
## converged
```



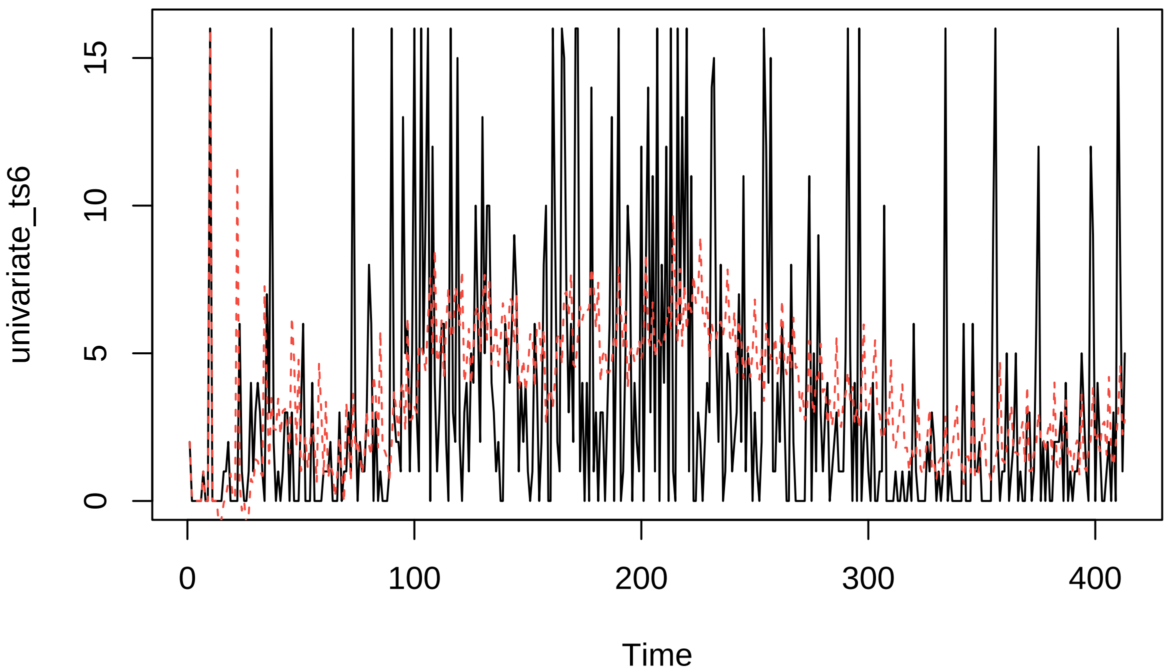

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     sma1
##       -0.9474  -0.9452
## s.e.   0.0166   0.0349
##
## sigma^2 estimated as 19.74:  log likelihood = -1179.23,  aic = 2364.45
##
## $degrees_of_freedom
## [1] 398
##
## $ttable
##      Estimate     SE  t.value p.value
## ma1   -0.9474 0.0166 -57.0612       0
## sma1  -0.9452 0.0349 -27.0550       0
##
## $AIC
## [1] 5.911134
##
## $AICc
```

```
## [1] 5.91121
##
## $BIC
## [1] 5.94107
```

```
## [1] 5.911134
```

```
## [1] 5.94107
```

The AIC for this SARIMA model is 5.91. The BIC for this SARIMA model is 5.94.



Above graph represents the time series along with the fitted values

Perform cross-validation on ARIMA model

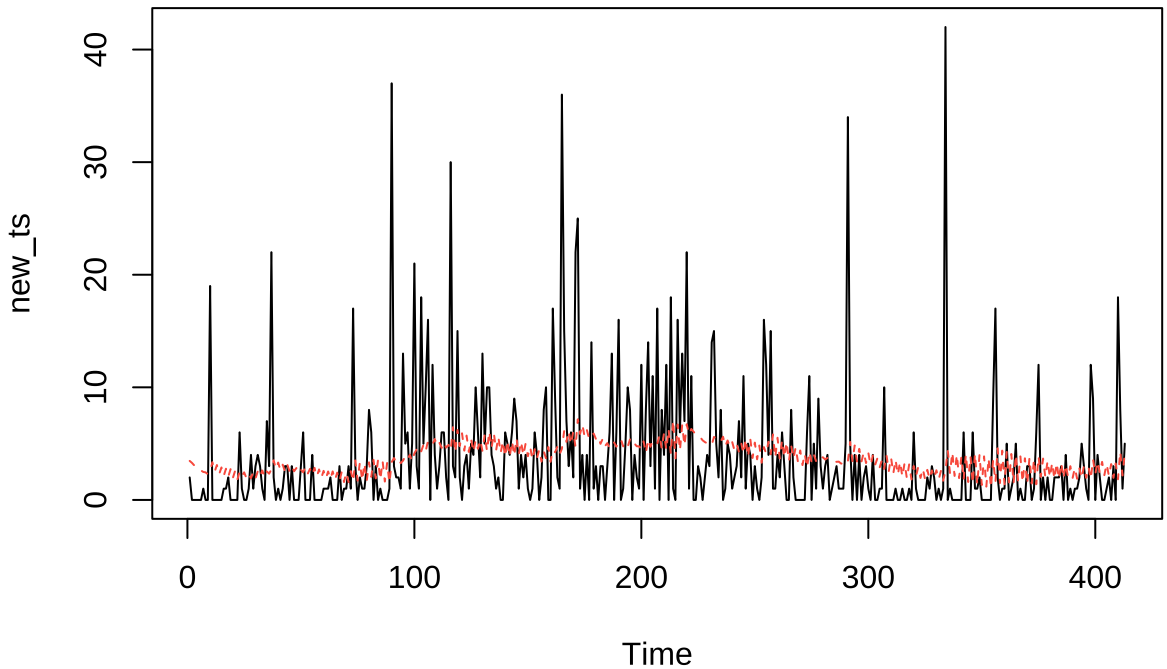
```
##  [1] 3.329902e+01 3.327590e+01 3.325912e+01 3.320812e+01 3.898795e+01
##  [6] 3.265190e+01 3.266926e+01 3.283018e+01 9.846892e+03 4.046007e+01
## [11] 5.555527e+01 5.524690e+01 2.833033e+06 9.846960e+03 6.914001e+01
## [16] 5.178956e+01 3.326945e+01 3.180089e+01 3.181764e+01 3.190395e+01
## [21] 3.589847e+01 3.266809e+01 3.223645e+01 3.238243e+01 6.433158e+03
## [26] 3.628773e+01 3.916790e+01 4.888327e+01 9.217366e+06 6.737546e+03
## [31] 4.588331e+01 4.380248e+01 3.325047e+01 3.181265e+01 3.139406e+01
## [36] 3.145390e+01 3.377326e+01 3.283749e+01 3.237512e+01 3.246927e+01
## [41] 1.551510e+03 3.369907e+01 5.225147e+01 4.900463e+01 2.579588e+06
## [46] 1.547552e+03 4.826784e+01 3.497173e+01 3.324736e+01 3.190548e+01
## [51] 3.145122e+01 3.240938e+01 3.373361e+01 3.266247e+01 3.270203e+01
## [56] 3.282743e+01 6.827320e+02 3.355034e+01 4.306066e+01 5.473041e+01
## [61] 1.777735e+06 6.687179e+02 4.446475e+01 4.063132e+01
```

```
## [1] 31.39406
```

```
## [1] 35
```

```
##
## Call:
## arima(x = new_ts, order = c(new_p, new_d, new_q))
```

```
##
## Coefficients:
##          ar1     ar2     ma1      ma2  intercept
##       0.0214  0.9501  0.0482  -0.9517     3.5089
## s.e.  0.0196  0.0188  0.0250   0.0248     0.8887
##
## sigma^2 estimated as 32.34:  log likelihood = -1305.1,  aic = 2622.2
```




```
## [1] 2622.195
```

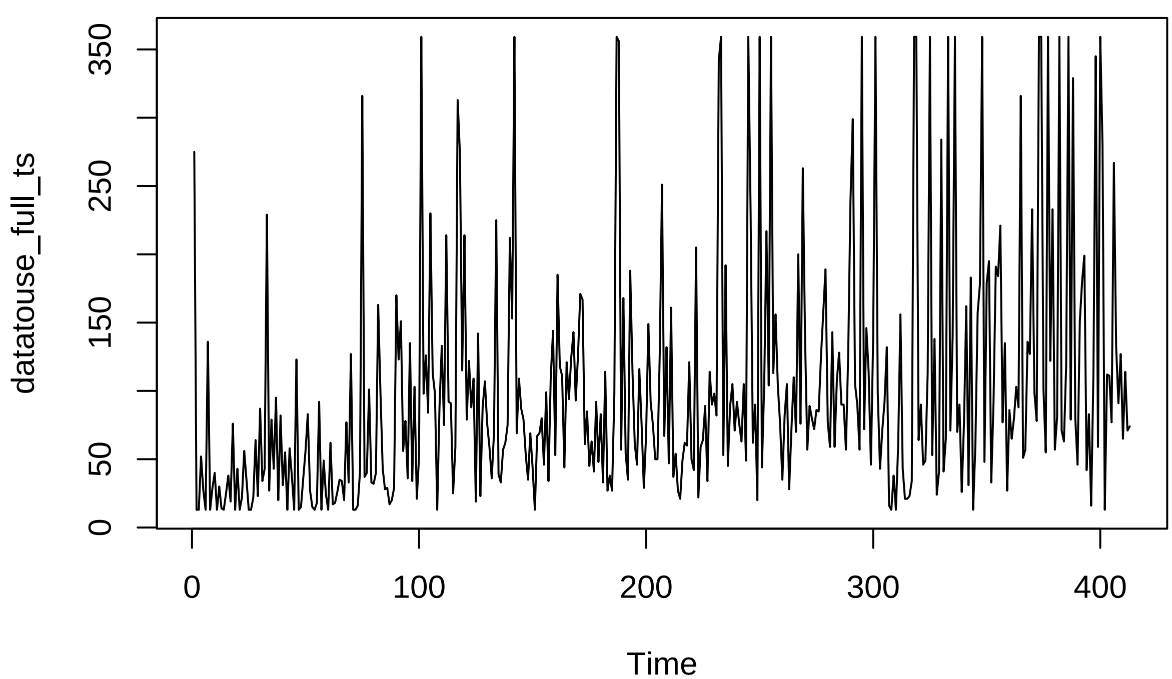
```
## [1] 2646.336
```

Built time series models on the full dataset.

```
##    Date(YMW) Frequency
## 1  2013-11-3       275
## 2  2013-11-4        13
## 3  2013-12-1        13
## 4  2013-12-2        52
## 5  2013-12-3        27
## 6  2013-12-4        13
## 7  2014-01-1       136
## 8  2014-01-2        13
## 9  2014-01-3        29
## 10 2014-01-4        40
```

```
## Time Series:
## Start = 1
## End = 413
## Frequency = 1
##   [1] 275.0  13.0  13.0  52.0  27.0  13.0 136.0  13.0  29.0  40.0  13.0  30.0
##  [13]  14.0  13.0  25.0  38.0  19.0  76.0  13.0  43.0  13.0  22.0  56.0  35.0
##  [25]  13.0  13.0  22.0  64.0  23.0  87.0  34.0  43.0 229.0  27.0  79.0  43.0
```

```
##  [37]  95.0  20.0  82.0  31.0  55.0  13.0  58.0  37.0  13.0 123.0  13.0  15.0
##  [49]  36.0  56.0  83.0  27.0  15.0  13.0  18.0  92.0  13.0  49.0  24.0  13.0
##  [61]  62.0  17.0  18.0  26.0  35.0  34.0  20.0  77.0  33.0 127.0  13.0  13.0
##  [73]  16.0  41.0 316.0  37.0  40.0 101.0  33.0  32.0  40.0 163.0  99.0  43.0
##  [85]  28.0  29.0  17.0  20.0  29.0 170.0 123.0 151.0  56.0  78.0  36.0 135.0
##  [97]  34.0 103.0  21.0  52.0 359.2  98.0 126.0  84.0 230.0 113.0  98.0  13.0
## [109]  88.0 133.0  75.0 214.0  92.0  91.0  25.0  59.0 313.0 274.0 115.0 214.0
## [121]  79.0 122.0  88.0 109.0  19.0 142.0  23.0  88.0 107.0  76.0  59.0  36.0
## [133]  69.0 225.0  39.0  33.0  57.0  62.0  75.0 212.0 153.0 359.2  69.0 109.0
## [145]  87.0  79.0  53.0  35.0  69.0  44.0  13.0  67.0  69.0  80.0  46.0  99.0
## [157]  34.0 110.0 144.0  53.0 185.0 118.0 111.0  44.0 121.0  94.0 124.0 143.0
## [169]  93.0 126.0 171.0 167.0  61.0  85.0  45.0  63.0  41.0  92.0  48.0  83.0
## [181]  33.0 114.0  27.0  38.0  27.0  97.0 359.2 356.0  57.0 168.0  54.0  35.0
## [193] 188.0 114.0  61.0  46.0 116.0  76.0  29.0  68.0 149.0  92.0  75.0  50.0
## [205]  50.0 127.0 251.0  67.0 132.0  47.0 161.0  37.0  54.0  27.0  21.0  49.0
## [217]  62.0  60.0 121.0  50.0  42.0 205.0  22.0  59.0  64.0  89.0  34.0 114.0
## [229]  90.0  98.0  82.0 342.0 359.2  53.0 192.0  45.0  89.0 105.0  71.0  92.0
## [241]  76.0  63.0 105.0  49.0 359.2 234.0  62.0  90.0  20.0 359.2  44.0 104.0
## [253] 217.0 104.0 359.2 113.0 156.0 104.0  76.0  35.0  82.0 105.0  28.0  72.0
## [265] 110.0  70.0 200.0  76.0 263.0 138.0  57.0  89.0  80.0  72.0  86.0  85.0
## [277] 125.0 157.0 189.0  77.0  59.0 143.0  59.0 108.0 128.0  90.0  90.0  57.0
## [289] 127.0 239.0 299.0 105.0  90.0  57.0 359.2  72.0 146.0 114.0  46.0 141.0
## [301] 359.2  99.0  43.0  71.0  92.0 132.0  16.0  13.0  38.0  13.0  60.0 156.0
## [313]  43.0  21.0  21.0  23.0  34.0 359.2 359.2  64.0  90.0  46.0  49.0 113.0
## [325] 359.2  53.0 138.0  24.0  41.0 284.0  41.0  65.0 359.2  71.0 140.0 359.2
## [337]  70.0  90.0  26.0  77.0 162.0  31.0 183.0  13.0  60.0 157.0 178.0 359.2
## [349]  48.0 179.0 195.0  33.0  92.0 191.0 184.0 221.0  77.0 135.0  27.0  86.0
## [361]  65.0  80.0 103.0  88.0 316.0  51.0  57.0 136.0 127.0 233.0  99.0  78.0
## [373] 359.2 359.2 101.0  55.0 359.2 122.0 233.0  57.0  81.0 359.2  71.0  63.0
## [385] 119.0 359.2  79.0 329.0  80.0  46.0 151.0 180.0 199.0  42.0  83.0  16.0
## [397] 114.0 345.0  59.0 359.2 280.0  13.0 112.0 111.0  77.0 267.0 132.0  91.0
## [409] 127.0  65.0 114.0  71.0  74.0
```

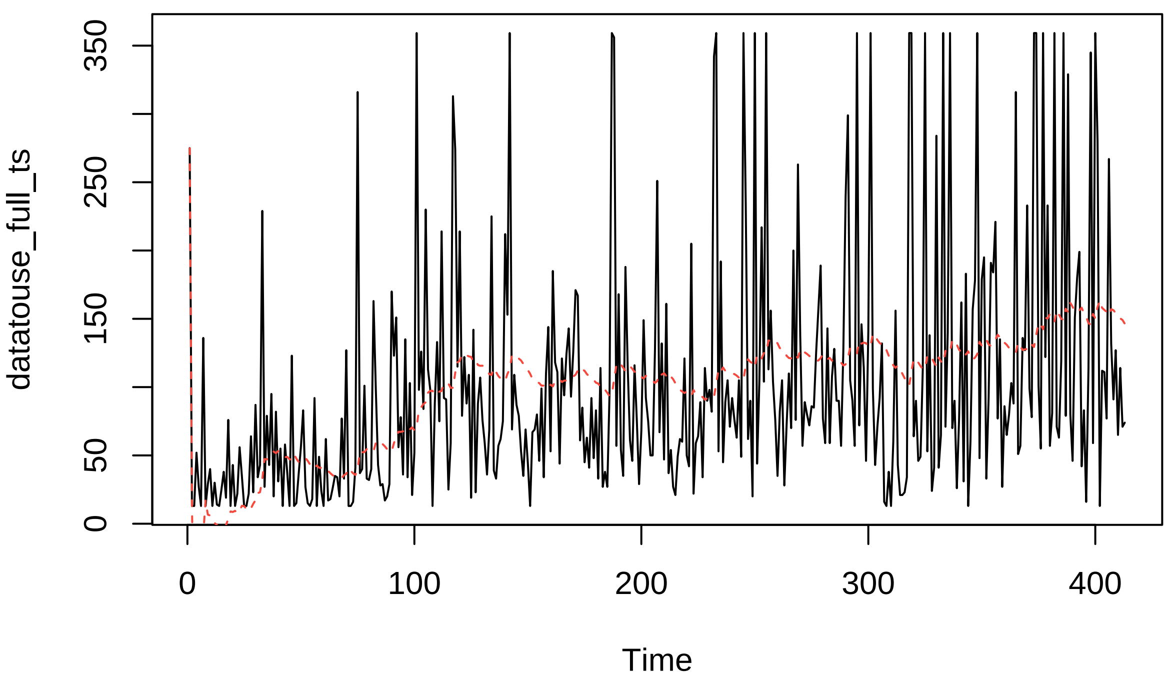


```
##  [1] 1.319852e+04 1.323291e+04 1.323780e+04 1.319868e+04 3.140040e+04
##  [6] 1.139364e+04 1.142786e+04 1.143820e+04 1.409871e+08 3.172855e+04
## [11] 1.086333e+04 1.090663e+04 1.370353e+11 2.031356e+08 9.778919e+05
## [16] 2.813356e+04 1.323070e+04 1.224684e+04 1.217776e+04 1.321470e+04
## [21] 1.540449e+04 1.143063e+04 1.143000e+04 1.144530e+04 5.681044e+07
## [26] 1.569247e+04 1.090730e+04 1.087660e+04 9.300909e+10 8.166685e+07
## [31] 2.614455e+05 4.635361e+04 1.318881e+04 1.216698e+04 1.327003e+04
## [36] 1.320240e+04 1.151650e+04 1.142973e+04 1.143386e+04 1.144148e+04
## [41] 1.771426e+07 1.166139e+04 1.088004e+04 1.089400e+04 2.480627e+10
## [46] 2.629325e+07 6.710611e+04 1.137739e+04
```

```
## [1] 10863.33
```

```
## [1] 11
```

```
##
## Call:
## arima(x = datatouse_full_ts, order = c(0, 2, 2))
##
## Coefficients:
##           ma1     ma2
##       -1.9625  0.9626
## s.e.   0.0172  0.0169
##
## sigma^2 estimated as 7180:  log likelihood = -2414.9,  aic = 4835.79
```




```
## [1] 4835.795
```

```
## [1] 4847.85
```

Cross-validation for SARIMA models using full dataset

```
##    [1] 1.319852e+04 1.319852e+04 1.323291e+04 1.327162e+04 3.140040e+04
##    [6] 2.886391e+04 1.139364e+04 1.436352e+04 1.323070e+04 1.329452e+04
```

```
##  [11] 1.224684e+04 1.342943e+04 1.540449e+04 2.367627e+04 1.143063e+04
##  [16] 1.476771e+04 1.323291e+04 1.323291e+04 1.323461e+04 1.328928e+04
##  [21] 1.139364e+04 2.858923e+04 1.072867e+04 1.415454e+04 1.224684e+04
##  [26] 1.330581e+04 1.315641e+04 1.339758e+04 1.143063e+04 2.373340e+04
##  [31] 1.139310e+04 1.451154e+04 1.323780e+04 1.323780e+04 1.319619e+04
##  [36] 1.328675e+04 1.142786e+04 2.863583e+04 1.142911e+04 1.398388e+04
##  [41] 1.217776e+04 1.329953e+04 1.321549e+04 1.335702e+04 1.143000e+04
##  [46] 2.372303e+04 1.142756e+04 1.428484e+04 3.140040e+04 3.140040e+04
##  [51] 1.139364e+04 3.127774e+04 1.409871e+08 1.458639e+05 3.172855e+04
##  [56] 3.187410e+04 1.540449e+04 3.125740e+04 1.143063e+04 3.146768e+04
##  [61] 5.681044e+07 3.498611e+04 1.569247e+04 3.645144e+04 1.139364e+04
##  [66] 1.139364e+04 1.072867e+04 1.140546e+04 3.172855e+04 2.880797e+04
##  [71] 1.094313e+04 1.200696e+04 1.143063e+04 1.140846e+04 1.139310e+04
##  [76] 1.159801e+04 1.569247e+04 2.305725e+04 1.089641e+04 1.224547e+04
##  [81] 1.142786e+04 1.142786e+04 1.142911e+04 1.144503e+04 1.086333e+04
##  [86] 2.849861e+04 1.086756e+04 1.197437e+04 1.143000e+04 1.144901e+04
##  [91] 1.142756e+04 1.165617e+04 1.090730e+04 2.312623e+04 1.088658e+04
##  [96] 1.220021e+04 1.409871e+08 1.409871e+08 3.172855e+04 1.439857e+08
## [101] 1.370353e+11 1.700710e+08 2.031356e+08 1.955894e+08 5.681044e+07
## [106] 1.445909e+08 1.569247e+04 1.487391e+08 9.300909e+10 6.826199e+07
## [111] 8.166685e+07 2.197100e+08 3.172855e+04 3.172855e+04 1.094313e+04
## [116] 3.143941e+04 2.031356e+08 9.411829e+04 1.196374e+05 9.615548e+04
## [121] 1.569247e+04 3.138977e+04 1.089641e+04 3.141893e+04 8.166685e+07
## [126] 7.174315e+04 4.359106e+04 9.761939e+04 1.086333e+04 1.086333e+04
## [131] 1.086756e+04 1.091000e+04 9.778919e+05 2.878702e+04 1.169444e+04
## [136] 1.288410e+04 1.090730e+04 1.090318e+04 1.088658e+04 1.090465e+04
## [141] 2.614432e+05 2.359008e+04 1.154499e+04 1.276781e+04 1.540449e+04
## [146] 1.540449e+04 1.143063e+04 1.517920e+04 5.681044e+07 7.950021e+04
## [151] 1.569247e+04 1.530447e+04 1.343015e+04 1.515189e+04 1.143060e+04
## [156] 1.564298e+04 3.174767e+07 2.399229e+04 1.366992e+04 1.562238e+04
## [161] 1.143063e+04 1.143063e+04 1.139310e+04 1.144931e+04 1.569247e+04
## [166] 2.847066e+04 1.089641e+04 1.196345e+04 1.143060e+04 1.145365e+04
## [171] 1.129460e+04 1.166204e+04 1.366992e+04 2.312537e+04 1.089934e+04
## [176] 1.217818e+04 1.143000e+04 1.143000e+04 1.142756e+04 1.144925e+04
## [181] 1.090730e+04 2.864706e+04 1.088658e+04 1.195912e+04 1.143337e+04
## [186] 1.145380e+04 1.143007e+04 1.139619e+04 1.094411e+04 2.312007e+04
## [191] 1.086513e+04 1.216724e+04 1.151650e+04 1.151650e+04 1.142973e+04
## [196] 1.141354e+04 1.771426e+07 5.819413e+04 1.166139e+04 1.244029e+04
## [201] 1.110799e+04 1.139892e+04 1.147799e+04 1.163882e+04 1.011349e+07
## [206] 2.267954e+04 1.121595e+04 1.238769e+04 1.142973e+04 1.142973e+04
## [211] 1.144791e+04 1.144983e+04 1.166139e+04 2.854566e+04 1.088317e+04
## [216] 1.195350e+04 1.147799e+04 1.145451e+04 1.145754e+04 1.166127e+04
## [221] 1.121595e+04 2.311888e+04 1.088401e+04 1.215162e+04 1.143386e+04
## [226] 1.143386e+04 1.143919e+04 1.144968e+04 1.088004e+04 2.866038e+04
## [231] 1.086784e+04 1.194786e+04 1.143704e+04 1.145356e+04 1.144141e+04
## [236] 1.166463e+04 1.090695e+04 2.312570e+04 1.087098e+04 1.214632e+04
```

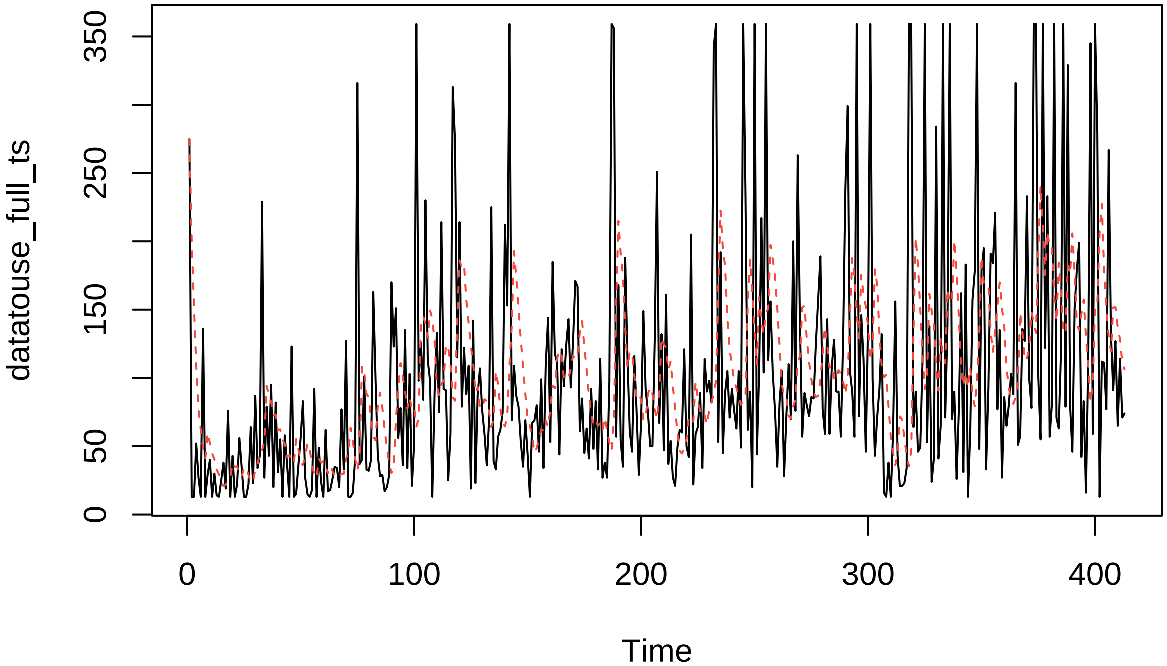
```
## [1] 10728.67
```

```
## [1] 23 67
```

```
## [1] 240
```

```
## Series: datatouse_full_ts
## ARIMA(0,0,1)
##
```

```
## Coefficients:
##           ma1     sma1
##       -0.4797  -0.4797
## s.e.      NaN      NaN
##
## sigma^2 = 8397:  log likelihood = -2445.45
## AIC=4896.9   AICc=4896.96   BIC=4908.97
```



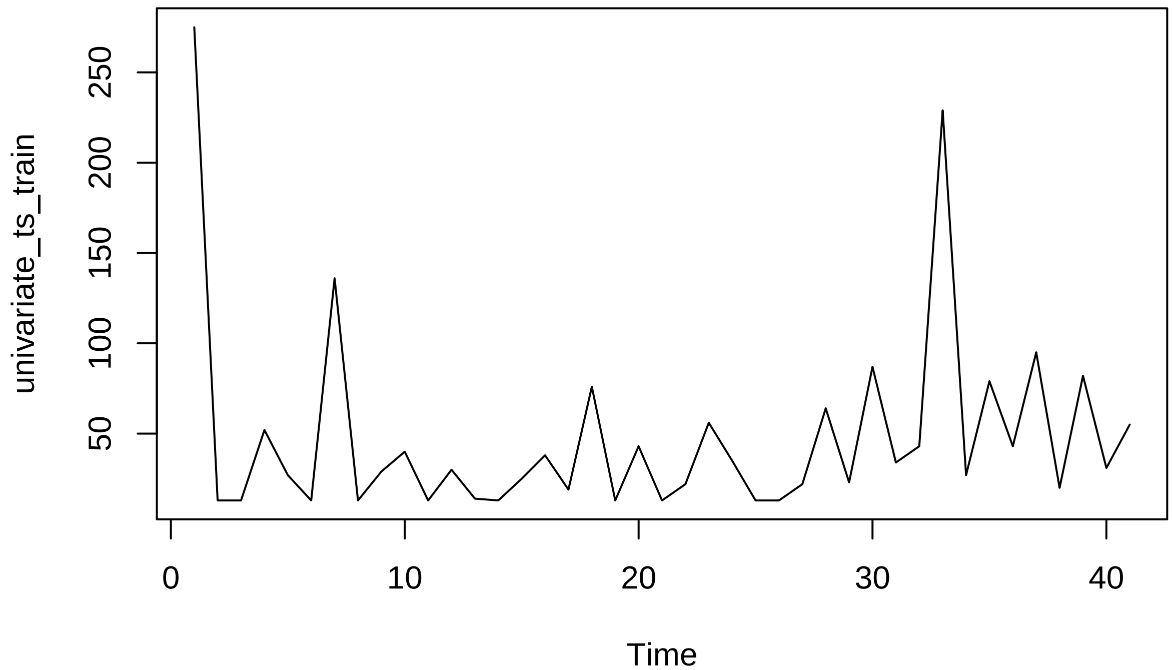

```
## [1] 4896.905
```

```
## [1] 4908.968
```

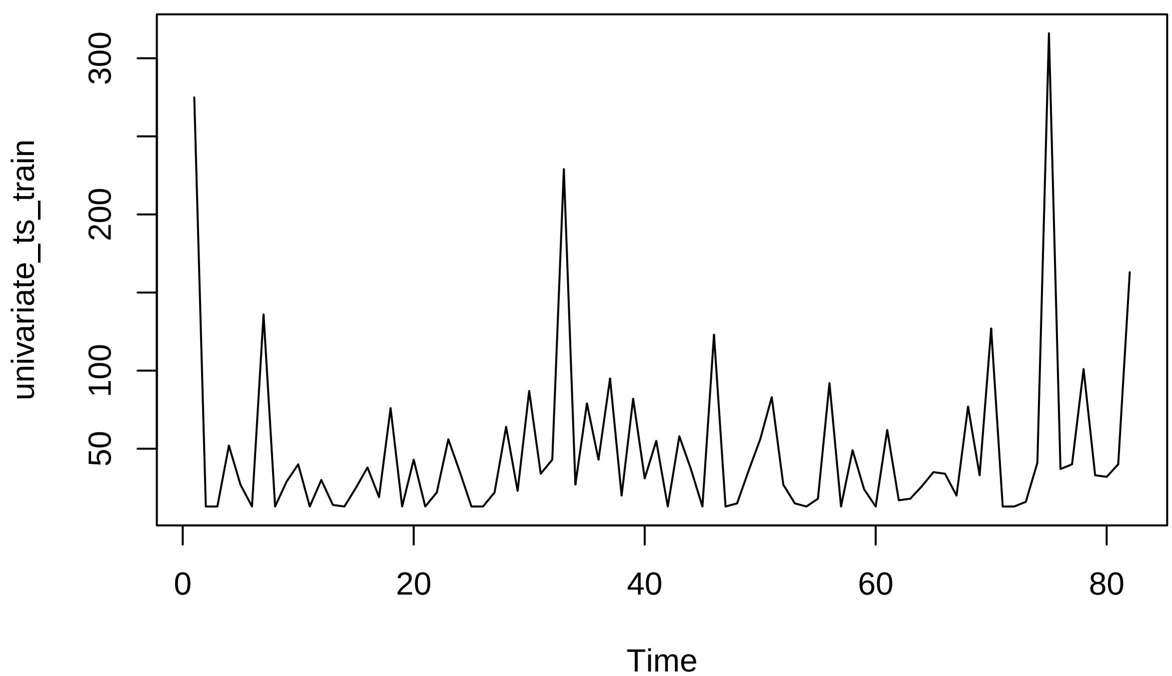
Perform Nested Cross-Validation on the full dataset

Nested cross-validation is performed on weekly data with winsorization to tune the SARIMA model's hyperparameters.

```
## [1] "this is the number of rows for the training dataset 1: 41"
## [1] "this is the number of rows for the testing dataset 1: 41"
```

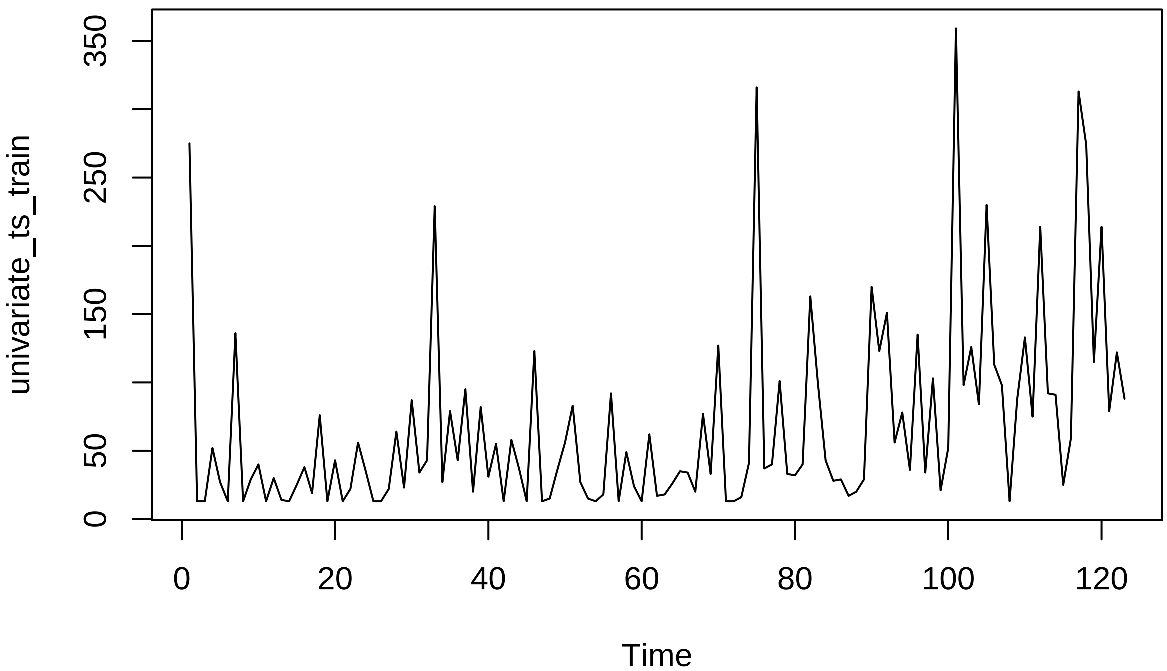



```
## [1] "this is the currect values: 0, 1, 1, 1, 0, 0, 12"
## [1] "this is the currect values: 0, 1, 1, 1, 0, 0, 12"
## [1] "this is total count: 240"
## [1] "there are total 144 different models"
## [1] "this is the number of rows for the training dataset 2: 82"
## [1] "this is the number of rows for the testing dataset 2: 41"
```

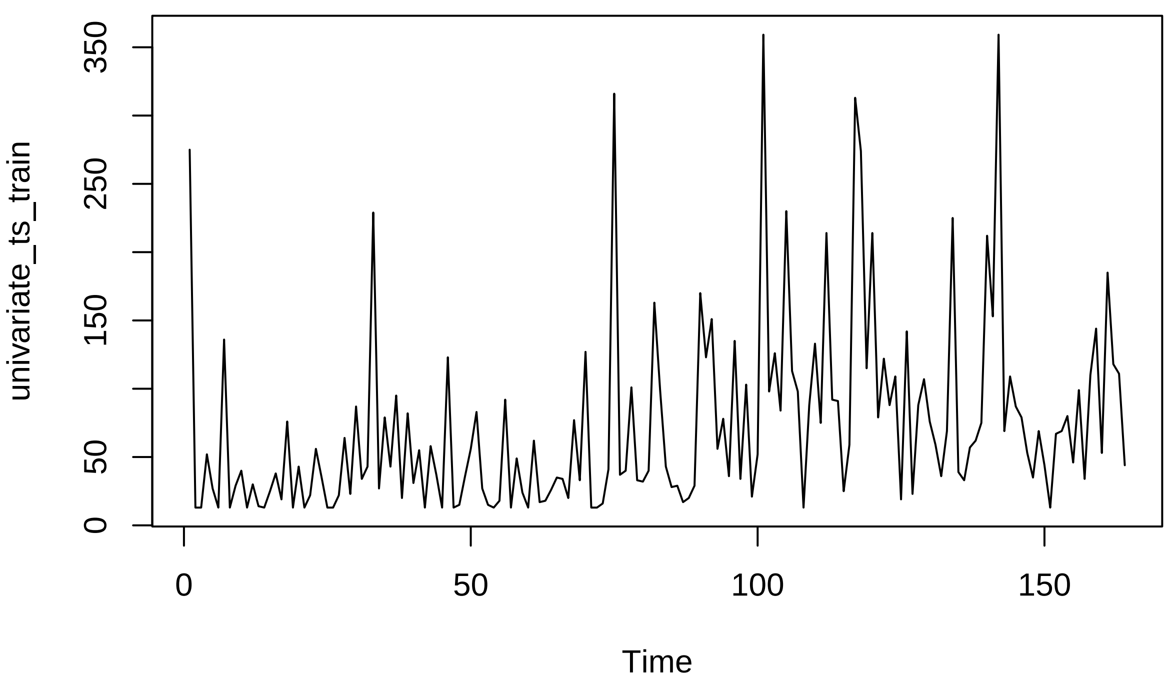



```
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is total count: 144"
## [1] "there are total 144 different models"
```

```
## [1] "this is the number of rows for the training dataset 3: 123"
## [1] "this is the number of rows for the testing dataset 3: 41"
```

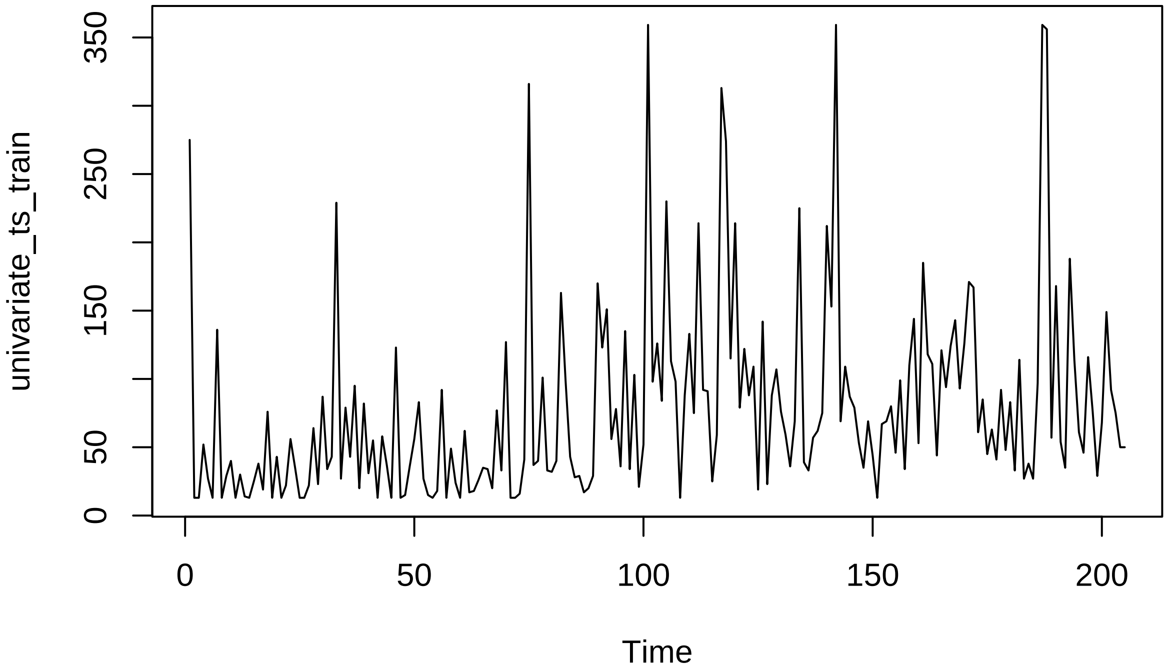



```
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is total count: 144"
## [1] "there are total 144 different models"
## [1] "this is the number of rows for the training dataset 4: 164"
## [1] "this is the number of rows for the testing dataset 4: 41"
```

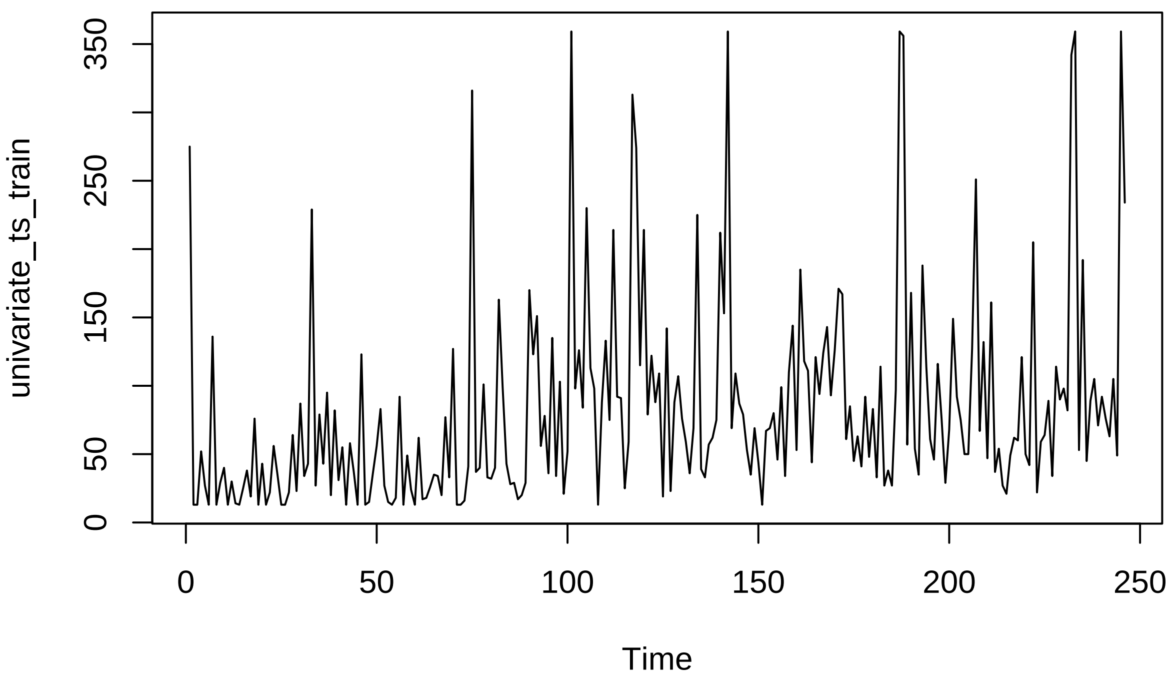



```
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
```

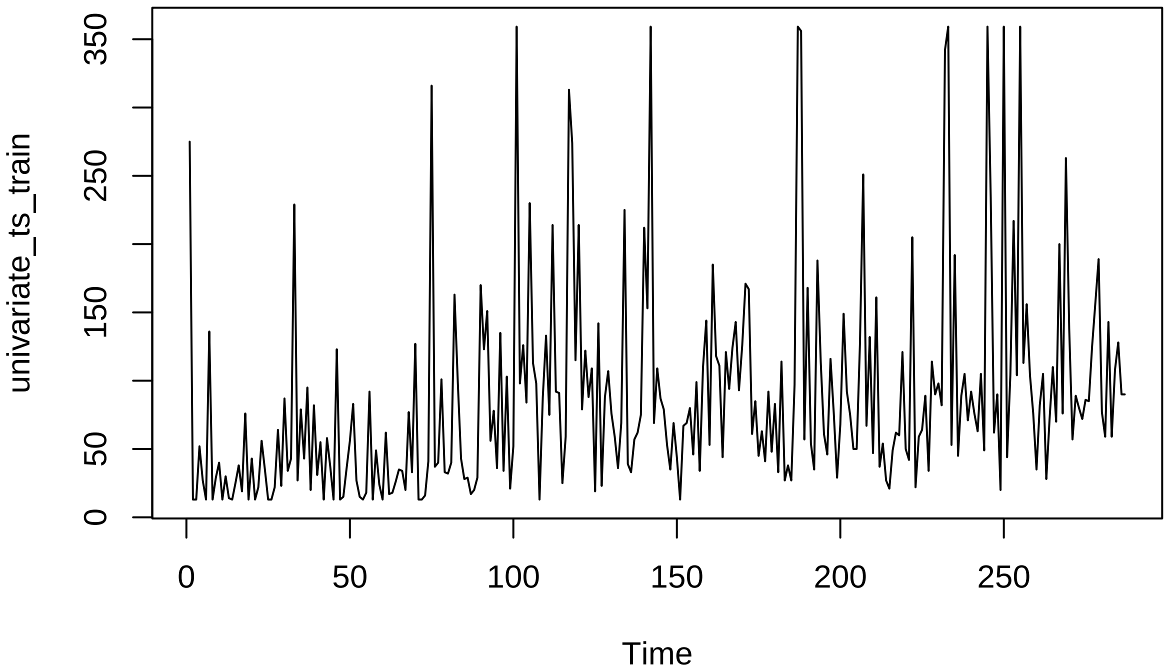
```
## [1] "this is total count: 144"
## [1] "there are total 144 different models"
## [1] "this is the number of rows for the training dataset 5: 205"
## [1] "this is the number of rows for the testing dataset 5: 41"
```



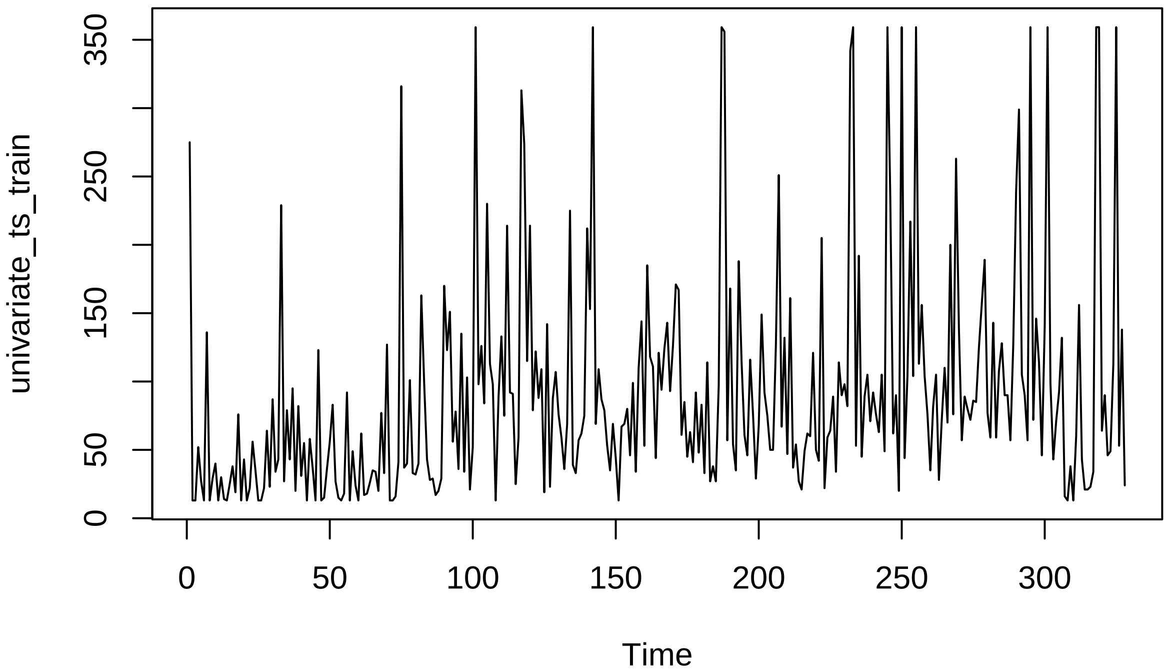

```
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is total count: 144"
## [1] "there are total 144 different models"
## [1] "this is the number of rows for the training dataset 6: 246"
## [1] "this is the number of rows for the testing dataset 6: 41"
```



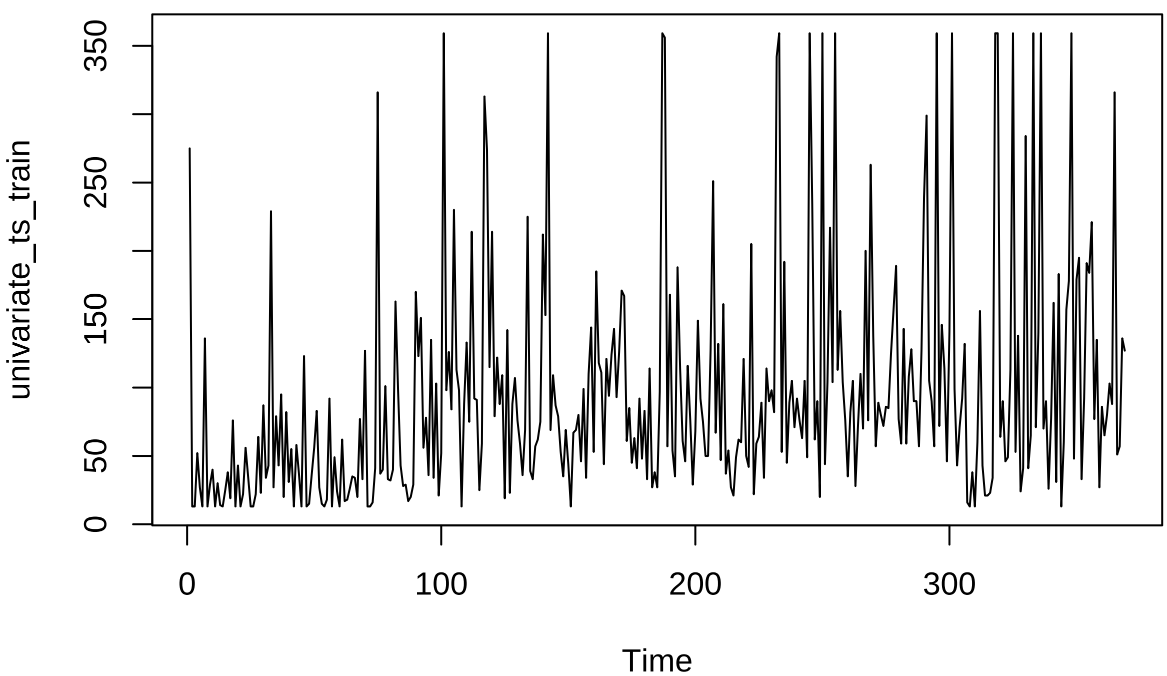
```
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is total count: 144"
## [1] "there are total 144 different models"
## [1] "this is the number of rows for the training dataset 7: 287"
## [1] "this is the number of rows for the testing dataset 7: 41"
```




```
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is total count: 144"
## [1] "there are total 144 different models"
## [1] "this is the number of rows for the training dataset 8: 328"
## [1] "this is the number of rows for the testing dataset 8: 41"
```




```
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is total count: 144"
## [1] "there are total 144 different models"
## [1] "this is the number of rows for the training dataset 9: 369"
## [1] "this is the number of rows for the testing dataset 9: 41"
```



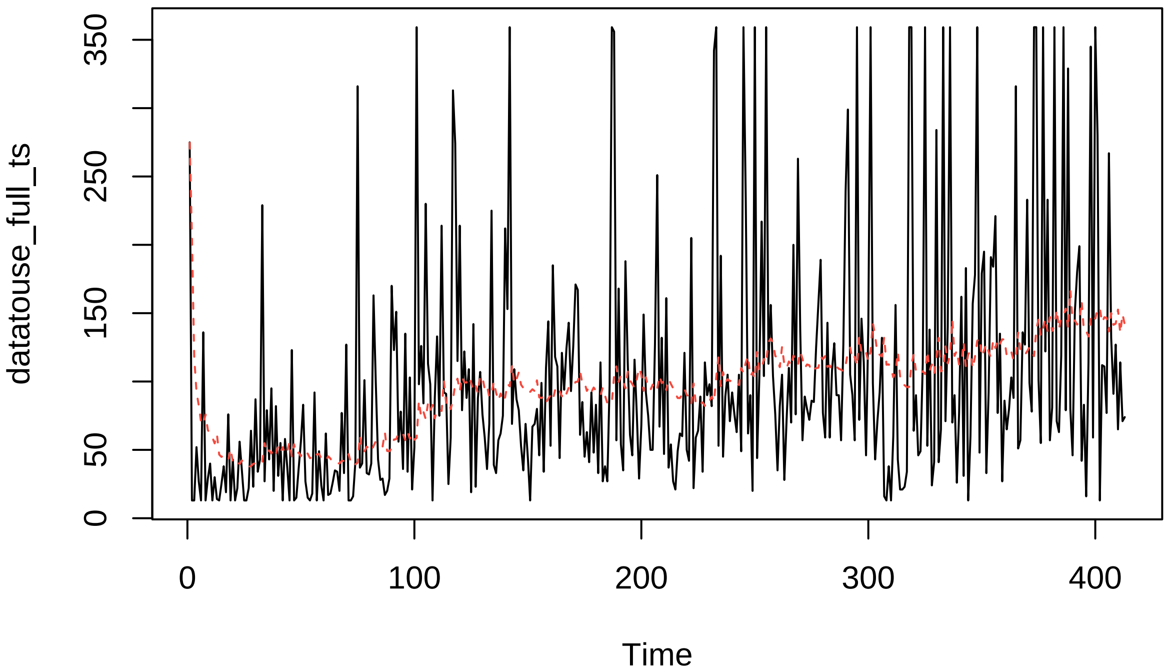

```
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is the currect values: 1, 1, 1, 1, 0, 0, 12"
## [1] "this is total count: 144"
## [1] "there are total 144 different models"
```

```
##   [1] 1.128392e+04 1.128392e+04 8.198454e+03 1.087620e+04 1.293202e+07
##   [6] 1.404357e+05 3.682282e+05 1.968539e+04 1.129526e+04 1.091487e+04
##  [11] 7.858167e+03 1.092590e+04 3.454564e+06 4.956831e+04 6.849556e+04
##  [16] 1.374233e+04 8.198454e+03 8.198454e+03 9.286062e+03 8.162108e+03
##  [21] 3.682282e+05 4.246354e+04 7.665819e+05 1.425913e+04 7.858167e+03
##  [26] 7.824566e+03 8.720505e+03 7.814593e+03 6.849556e+04 1.774032e+04
##  [31] 8.197647e+03 8.473089e+03 8.033121e+03 8.033121e+03 8.026636e+03
##  [36] 7.998168e+03 1.190873e+06 3.988090e+04 1.034653e+06 1.398261e+04
##  [41] 7.830409e+03 7.833936e+03 7.845705e+03 7.827804e+03 9.195394e+03
##  [46] 1.708683e+04 8.032651e+03 8.515889e+03 1.129720e+04 1.129720e+04
##  [51] 7.856331e+03 1.100968e+04 3.604891e+06 7.629489e+04 6.866494e+04
##  [56] 1.802852e+04 1.096384e+04 1.106541e+04 7.789643e+03 1.106158e+04
##  [61] 3.741438e+06 4.061315e+04 2.140244e+04 1.403055e+04 7.856331e+03
##  [66] 7.856331e+03 8.718040e+03 7.856312e+03 6.866494e+04 1.928787e+04
##  [71] 8.076444e+03 9.750512e+03 7.789643e+03 7.788949e+03 8.002322e+03
##  [76] 7.951615e+03 2.140244e+04 1.617902e+04 3.185026e+04 1.018890e+04
##  [81] 7.827282e+03 7.827282e+03 7.842906e+03 7.808908e+03 9.183916e+03
##  [86] 1.881151e+04 9.079561e+03 9.579784e+03 7.881219e+03 7.797544e+03
##  [91] 7.852359e+03 7.823420e+03 3.311344e+04 1.627252e+04 8.981281e+03
##  [96] 1.013853e+04 1.606841e+05 1.606841e+05 7.983368e+03 4.995420e+04
## [101] 1.608302e+09 1.159674e+07 1.704248e+07 7.442988e+06 1.760895e+04
## [106] 6.565089e+04 8.171361e+03 1.693867e+05 1.696237e+07 7.339437e+06
## [111] 1.953191e+04 5.814849e+06 7.983368e+03 7.983368e+03 8.611744e+03
## [116] 7.848570e+03 1.704248e+07 1.053711e+05 9.027133e+06 1.038907e+05
## [121] 8.170552e+03 7.850187e+03 7.996706e+03 7.927471e+03 1.952322e+04
## [126] 2.171871e+05 6.422873e+05 2.312944e+05 7.868707e+03 7.868707e+03
## [131] 9.382571e+03 8.356923e+03 2.738219e+06 2.178744e+04 8.221281e+05
## [136] 1.135466e+04 7.937095e+03 7.894734e+03 8.065520e+03 8.353233e+03
## [141] 4.868022e+05 4.112813e+04 5.188009e+06 3.102363e+04
```

```
## [1] 7788.949
```

```
## [1] 74
```

```
## Series: datatouse_full_ts
## ARIMA(1,1,1)(1,0,0)[12]
##
## Coefficients:
##          ar1      ma1    sar1
##       0.0394  -0.9689  0.0410
## s.e.  0.0513   0.0121  0.0505
##
## sigma^2 = 7216:  log likelihood = -2414.55
## AIC=4837.11   AICc=4837.2   BIC=4853.19
```

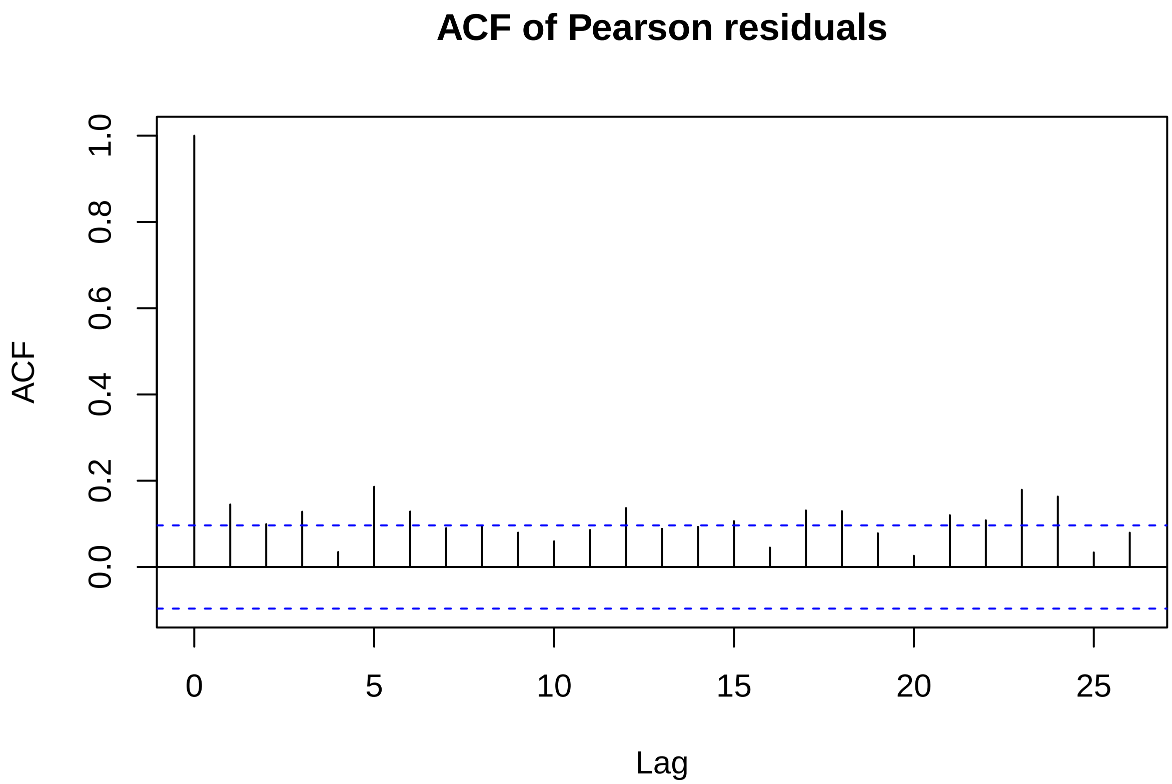


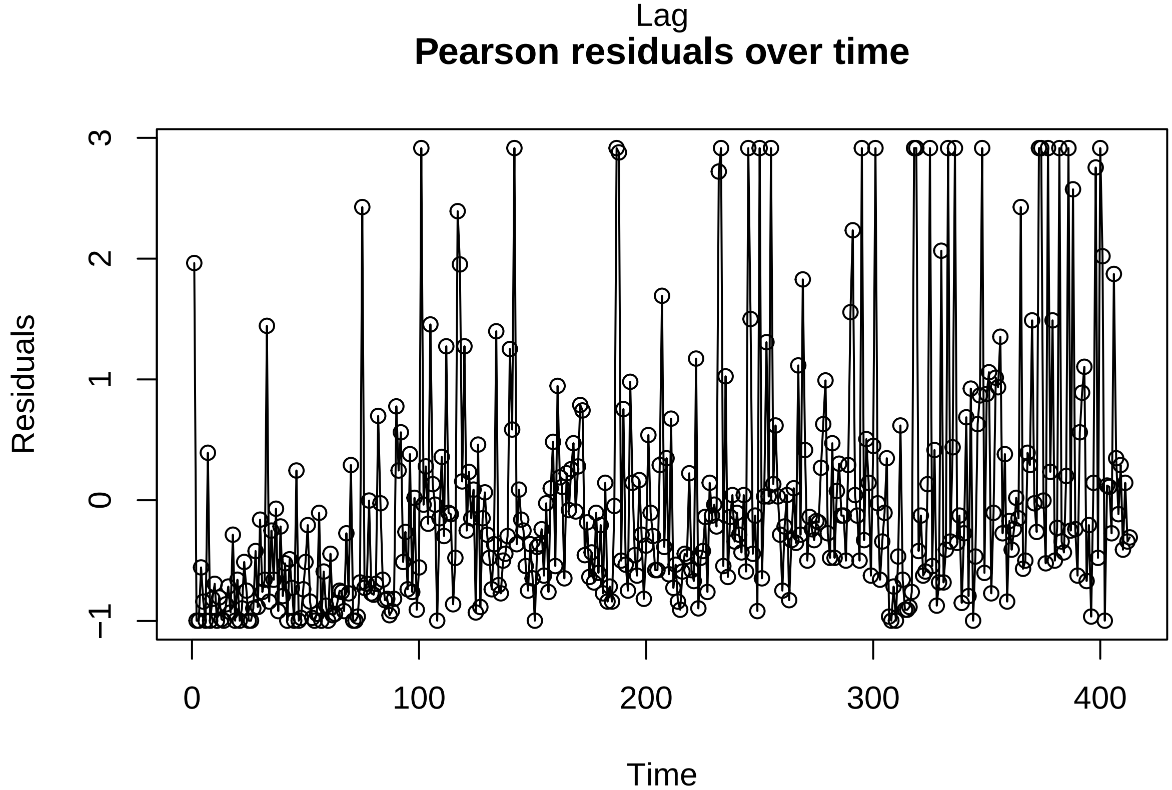

```
## [1] 4837.106
```

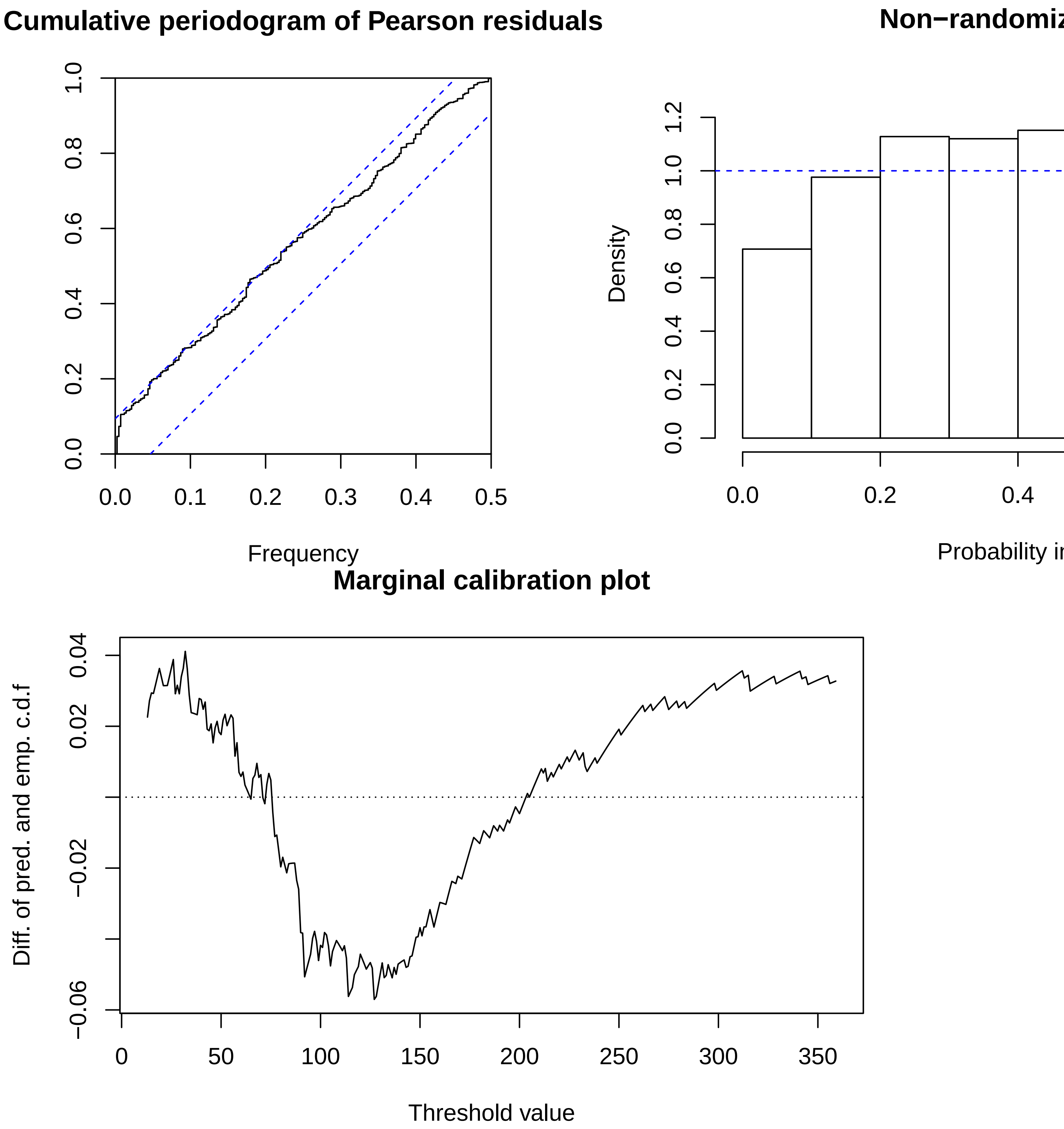
```
## [1] 4853.19
```

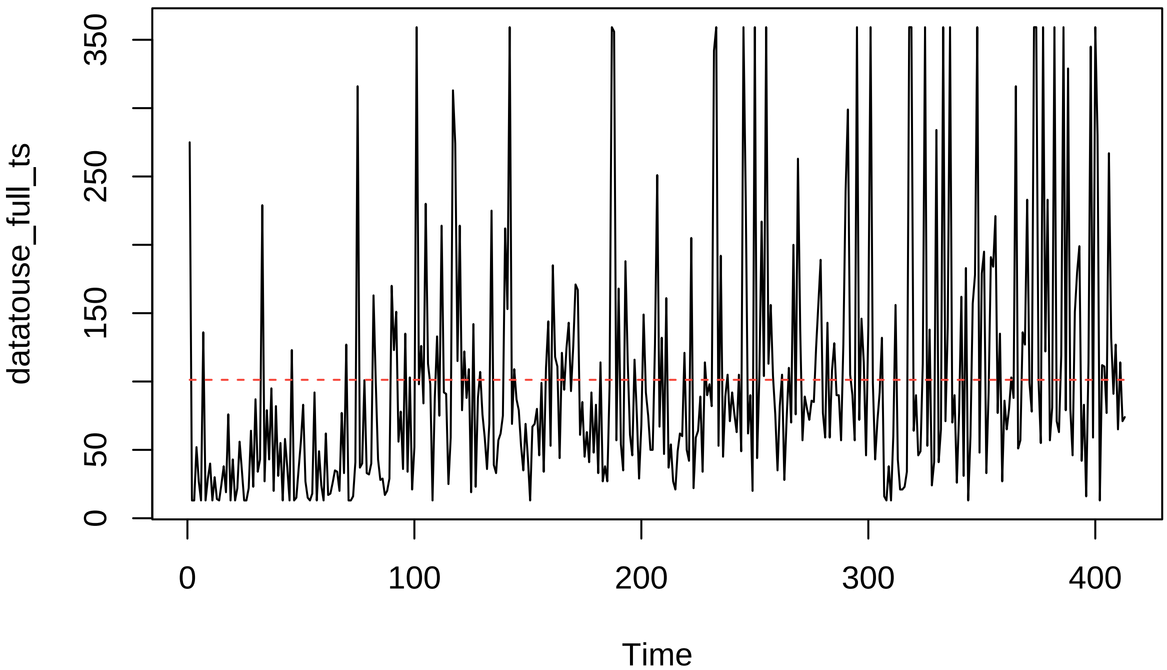
GSARIMA Model

```
##
## Call:
## tsglm(ts = datatouse_full_ts, link = "log", distr = "nbinom")
##
## Coefficients:
##              Estimate  Std.Error  CI(lower)  CI(upper)
## (Intercept)     4.618      0.043       4.53        4.7
## sigmasq         0.754         NA         NA         NA
## Standard errors and confidence intervals (level =  95 %) obtained
## by normal approximation.
##
## Link function: log
## Distribution family: nbinom (with overdispersion coefficient 'sigmasq')
## Number of coefficients: 2
## Log-likelihood: -Inf
## AIC: Inf
## BIC: Inf
## QIC: Inf
```


ACF of Pearson residuals
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25
Lag


Pearson residuals over time
Residuals
−1
0
1
2
3
0
100
200
300
400
Time


Cumulative periodogram of Pearson residuals
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.1
0.2
0.3
0.4
0.5
Frequency
Non−randomiz
Density
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
Probability in
Marginal calibration plot
Diff. of pred. and emp. c.d.f
0.04
0.02
−0.02
−0.06
0
50
100
150
200
250
300
350
Threshold value




```
## [1] Inf
## [1] Inf
```

Code Appendix:

```
knitr::opts_chunk$set(echo = TRUE)
library(naniar)
library(readr)
library(dplyr)
library(ggplot2)
library(gsarima)
library(forecast)
library(caret)
library(zoo)
library(astsa)
library(DescTools)
library(tscount)
setwd("~/Desktop")
isolates <- read_csv("isolates.csv")
isolates = isolates %>%
  select(-c(Computed_types, Virulence_genotypes, AST_phenotypes))

isolates = isolates %>%
  select(-c(Host_disease, PFGE_secondary_enzyme_pattern, PFGE_primary_enzyme_pattern, Stress_genotypes,

isolates = isolates %>%
  select(-c(Species_TaxID, `K-mer_group`, Organism_group))

isolates = isolates %>%
  select(-c(WGS_accession, WGS_prefix, Run, Isolate, Assembly))
```

```
isolates = isolates %>%
  select(-c(AMRFinderPlus_version, PD_Ref_Gene_Catalog_version, Level))

isolates <- isolates %>%
    mutate(across(.cols=c(Library_layout, Method, SRA_Center, Platform, AMR_genotypes_core, BioProject,

isolates <- isolates %>%
    mutate(across(.cols=c(SRA_release_date, Create_date), .fns = as.Date))


isolates = isolates %>%
  select(-c(Library_layout, Method, Platform, AMRFinderPlus_analysis_type, Isolate_identifiers, BioSampl

isolates = isolates %>%
  select(-Strain)

isolates$Outbreak = ifelse(is.na(isolates$Outbreak), 0, 1)
count_SNP = as.data.frame(table(isolates$SNP_cluster))
colnames(count_SNP)[colnames(count_SNP) == "Var1"] <- "SNP_cluster"
colnames(count_SNP)[colnames(count_SNP) == "Freq"] <- "Frequency"
count_SNP =count_SNP[order(-count_SNP$Frequency),]

count_SNP_20 = count_SNP[1:20,]
SNP_percentage = numeric(20)
for (i in 1:20){
  SNP_percentage[i] = (count_SNP$Frequency[i]/sum(count_SNP$Frequency))*100
}
count_SNP_20['SNP_percentage'] <- SNP_percentage
cluster_1 = isolates %>%
  filter((SNP_cluster == count_SNP_20[1,1]))
cluster_1
cluster_1$Create_date_YM = format(as.Date(cluster_1$Create_date), "%Y-%m")
datatouse = as.data.frame(table(cluster_1$Create_date_YM))
colnames(datatouse)[colnames(datatouse) == "Var1"] <- "Date"
colnames(datatouse)[colnames(datatouse) == "Freq"] <- "Frequency"
datatouse$Date = as.character(datatouse$Date)
datatouse[nrow(datatouse)+1,] = c("2013-12", 0)
datatouse[nrow(datatouse)+1,] = c("2014-02", 0)
datatouse[nrow(datatouse)+1,] = c("2014-04", 0)
datatouse[nrow(datatouse)+1,] = c("2021-03", 0)
datatouse = datatouse[order(datatouse$Date),]
rownames(datatouse) <- NULL
datatouse$Frequency = as.numeric(datatouse$Frequency)
datatouse
tsData = ts(datatouse, start = c(2013, 11), end = c(2022, 6), frequency = 12)
tsData
is.ts(tsData)
summary(tsData)
ts.plot(tsData, xlab="Year", ylab="Number of Listeria Monocytogenes Cases", main="Monthly totals of Lis
start(tsData)
end(tsData)
frequency(tsData)
univariate_ts = as.ts(datatouse$Frequency)
```

```r
univariate_ts
acf(univariate_ts)
pacf(univariate_ts)
AR1 <- arima(univariate_ts, order = c(1,0,0))
print(AR1)
ts.plot(univariate_ts)
AR1_fit <- univariate_ts - residuals(AR1)
points(AR1_fit, type = "l", col = 2, lty = 2)  # type = "l" means lines
AIC(AR1)
BIC(AR1)
AR2 <- arima(univariate_ts, order = c(0,0,1))
print(AR2)
ts.plot(univariate_ts)
AR2_fit <- univariate_ts - residuals(AR2)
points(AR2_fit, type = "l", col = 2, lty = 2)
AIC(AR2)
BIC(AR2)
AR3 <- arima(univariate_ts, order = c(1,0,1))
print(AR3)
ts.plot(univariate_ts)
AR3_fit <- univariate_ts - residuals(AR3)
points(AR3_fit, type = "l", col = 2, lty = 2)
AIC(AR3)
BIC(AR3)
AR4 <- arima(univariate_ts, order = c(1,1,1))
print(AR4)
ts.plot(univariate_ts)
AR4_fit <- univariate_ts - residuals(AR4)
points(AR4_fit, type = "l", col = 2, lty = 2)
AIC(AR4)
BIC(AR4)
AR5 <- sarima(univariate_ts,1,1,1,1,1,1,12)
print(AR5)
ts.plot(univariate_ts)
AR5_fit <- univariate_ts - resid(AR5$fit)
points(AR5_fit, type = "l", col = 2, lty = 2)
AR5$AIC
AR5$BIC
AR6 <- sarima(univariate_ts,0,1,1,0,1,1,12)
print(AR6)
ts.plot(univariate_ts)
AR6_fit <- univariate_ts - resid(AR6$fit)
points(AR6_fit, type = "l", col = 2, lty = 2)
AR6$AIC
AR6$BIC
datatouse[datatouse$Date == '2020-01',]$Frequency <- 0
univariate_ts2 = as.ts(datatouse$Frequency)
ts.plot(univariate_ts)
ts.plot(univariate_ts2)
AR7 <- sarima(univariate_ts2,0,1,1,0,1,1,12)
print(AR7)
AR7$AIC
AR7$BIC
```

```r
ts.plot(univariate_ts2)
AR7_fit <- univariate_ts2 - resid(AR7$fit)
points(AR7_fit, type = "l", col = 2, lty = 2)
Acf(univariate_ts2)
Pacf(univariate_ts2)
datatouse[datatouse$Date == '2020-01',]$Frequency <- round(mean(datatouse$Frequency))
univariate_ts3 = as.ts(datatouse$Frequency)
ts.plot(univariate_ts)
ts.plot(univariate_ts3)
AR8 <- sarima(univariate_ts3,0,1,1,0,1,1,12)
print(AR8)
AR8$AIC
AR8$BIC
ts.plot(univariate_ts3)
AR8_fit <- univariate_ts3 - resid(AR8$fit)
points(AR8_fit, type = "l", col = 2, lty = 2)
univariate_ts7 = Winsorize(univariate_ts)
ts.plot(univariate_ts7)
AR12 <- sarima(univariate_ts7,0,1,1,0,1,1,12)
print(AR12)
AR12$AIC
AR12$BIC
ts.plot(univariate_ts7)
AR12_fit <- univariate_ts7 - resid(AR12$fit)
points(AR12_fit, type = "l", col = 2, lty = 2)
cluster_1$Create_date = format(as.Date(cluster_1$Create_date), "%Y-%m-%d")
cluster_1$Create_date_YM = format(as.Date(cluster_1$Create_date), "%Y-%m")
  for (i in 1:dim(cluster_1[1])){
  date = as.numeric(format(as.Date(cluster_1$Create_date[i]), "%d"))
  cluster_1$week[i] = if(date >= 1 && date <= 7){
    1
  } else if(date >= 8 && date <= 14){
    2
  } else if(date >= 15 && date <= 21){
    3
  } else {
    4
  }
  cluster_1$Create_date_YMW[i] = sprintf("%s-%s", cluster_1$Create_date_YM[i], cluster_1$week[i])
  }
datatouse2 = as.data.frame(table(cluster_1$Create_date_YMW))
colnames(datatouse2)[colnames(datatouse2) == "Var1"] <- "Date(YMW)"
colnames(datatouse2)[colnames(datatouse2) == "Freq"] <- "Frequency"
datatouse2$`Date(YMW)` = as.character(datatouse2$Date)
datatouse2[nrow(datatouse2)+1,] = c("2013-11-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2013-12-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2013-12-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2013-12-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2013-12-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-01-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-01-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-02-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-02-2", 0)
```

```
datatouse2[nrow(datatouse2)+1,] = c("2014-02-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-02-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-03-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-04-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-04-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-04-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-04-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-05-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-05-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-07-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-09-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-09-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-10-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-11-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-11-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-11-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-12-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-12-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2014-12-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2015-01-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2015-01-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2015-01-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2015-02-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2015-03-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2015-03-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2015-04-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2015-04-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2015-06-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2015-08-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2015-08-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2015-09-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2015-09-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2015-09-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2016-02-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2016-04-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2016-05-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2016-10-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2016-10-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2017-01-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2017-02-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2017-03-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2017-03-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2017-07-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2017-07-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2017-08-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2017-09-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2017-10-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2017-11-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2017-12-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2018-01-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2018-03-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2018-04-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2018-05-1", 0)
```

```
datatouse2[nrow(datatouse2)+1,] = c("2018-07-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2018-07-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2018-08-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2018-10-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2019-01-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2019-02-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2019-05-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2019-05-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2019-06-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2019-06-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2019-06-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2019-07-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2019-07-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2019-08-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2019-10-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2019-12-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-01-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-01-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-02-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-03-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-03-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-04-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-04-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-04-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-05-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-05-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-05-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-06-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-06-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-07-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-07-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-08-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-08-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-08-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-09-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-10-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-11-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-11-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-11-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-12-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-12-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2020-12-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-01-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-01-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-01-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-02-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-03-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-03-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-03-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-03-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-04-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-05-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-06-4", 0)
```

```r
datatouse2[nrow(datatouse2)+1,] = c("2021-07-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-07-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-08-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-09-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-09-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-10-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-10-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-11-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-12-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2021-12-4", 0)
datatouse2[nrow(datatouse2)+1,] = c("2022-02-3", 0)
datatouse2[nrow(datatouse2)+1,] = c("2022-03-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2022-04-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2022-04-2", 0)
datatouse2[nrow(datatouse2)+1,] = c("2022-05-1", 0)
datatouse2[nrow(datatouse2)+1,] = c("2022-05-3", 0)


datatouse2 = datatouse2[order(datatouse2$Date),]
rownames(datatouse2) <- NULL
datatouse2$Frequency = as.numeric(datatouse2$Frequency)
datatouse2[1:10,]
univariate_ts4 = as.ts(datatouse2$Frequency)
univariate_ts4
ts.plot(univariate_ts4)
AR9 <- sarima(univariate_ts4,0,1,1,0,1,1,12)
print(AR9)
AR9$AIC
AR9$BIC
ts.plot(univariate_ts4)
AR9_fit <- univariate_ts4 - resid(AR9$fit)
points(AR9_fit, type = "l", col = 2, lty = 2)
datatouse2[datatouse2$Date == '2020-01-2',]$Frequency <- 0
datatouse2[datatouse2$Date == '2020-01-2',]$Frequency <- round(mean(datatouse2$Frequency))
univariate_ts5 = as.ts(datatouse2$Frequency)
univariate_ts5
ts.plot(univariate_ts5)
AR10 <- sarima(univariate_ts5,0,1,1,0,1,1,12)
print(AR10)
AR10$AIC
AR10$BIC
ts.plot(univariate_ts5)
AR10_fit <- univariate_ts5 - resid(AR10$fit)
points(AR10_fit, type = "l", col = 2, lty = 2)
univariate_ts6 = Winsorize(univariate_ts4)
ts.plot(univariate_ts6)
AR11 <- sarima(univariate_ts6,0,1,1,0,1,1,12)
print(AR11)
AR11$AIC
AR11$BIC
ts.plot(univariate_ts6)
AR11_fit <- univariate_ts6 - resid(AR11$fit)
points(AR11_fit, type = "l", col = 2, lty = 2)
```

```
datatouse2_train = datatouse2[1:round(4*(nrow(datatouse2)/5)),]
datatouse2_test = datatouse2[-(1:round(4*(nrow(datatouse2)/5))),]
univariate_ts_train = as.ts(datatouse2_train$Frequency)

p = c(0,1,2,3)
d = c(0,1,2,3)
q = c(0,1,2,3)

mse_list = c()

for(a in p){
  for (b in d){
    for (c in q){
      arima_models = arima(univariate_ts_train, order = c(a,b,c))
      predict_arima_models = predict(arima_models, n.ahead = 83)
      mse_value = mean((predict_arima_models$pred - datatouse2_test$Frequency)^2)
      mse_list = append(mse_list, mse_value)
      }
    }
}

print(mse_list)
print(min(mse_list))
print(which(mse_list == min(mse_list)))

new_p = 2
new_d = 0
new_q = 2
new_ts = as.ts(datatouse2$Frequency)
ARIMA_model_cv = arima(new_ts, order = c(new_p, new_d, new_q))
print(ARIMA_model_cv)
ts.plot(new_ts)
ARIMA_model_cv_fit = new_ts - residuals(ARIMA_model_cv)
points(ARIMA_model_cv_fit, type = "l", col=2, lty = 2)
AIC(ARIMA_model_cv)
BIC(ARIMA_model_cv)
isolates$Create_date = format(as.Date(isolates$Create_date), "%Y-%m-%d")
isolates$Create_date_YM = format(as.Date(isolates$Create_date), "%Y-%m")
for (i in 1:dim(isolates[1])){
  date = as.numeric(format(as.Date(isolates$Create_date[i]), "%d"))
  isolates$week[i] = if(date >= 1 && date <= 7){
    1
  } else if(date >= 8 && date <= 14){
    2
  } else if(date >= 15 && date <= 21){
    3
  } else {
    4
  }
  isolates$Create_date_YMW[i] = sprintf("%s-%s", isolates$Create_date_YM[i], isolates$week[i])
  }
datatouse_full = as.data.frame(table(isolates$Create_date_YMW))
colnames(datatouse_full)[colnames(datatouse_full) == "Var1"] <- "Date(YMW)"
```

```
colnames(datatouse_full)[colnames(datatouse_full) == "Freq"] <- "Frequency"
datatouse_full$`Date(YMW)` = as.character(datatouse_full$Date)
datatouse_full = datatouse_full[order(datatouse_full$`Date(YMW)`),]
datatouse_full = datatouse_full[17:nrow(datatouse_full),]
datatouse_full[nrow(datatouse_full)+1,] = c("2013-11-4", 0)
datatouse_full[nrow(datatouse_full)+1,] = c("2013-12-1", 0)
datatouse_full = datatouse_full[order(datatouse_full$`Date(YMW)`),]
rownames(datatouse_full) <- NULL
datatouse_full$Frequency = as.numeric(datatouse_full$Frequency)
datatouse_full$Frequency = Winsorize(datatouse_full$Frequency)
datatouse_full[1:10,]
datatouse_full_ts = as.ts(datatouse_full$Frequency)
datatouse_full_ts
ts.plot(datatouse_full_ts)
datatouse_full_train = datatouse_full[1:round(4*(nrow(datatouse_full)/5)),]
datatouse_full_test = datatouse_full[-(1:round(4*(nrow(datatouse_full)/5))),]
univariate_ts_train_full = as.ts(datatouse_full_train$Frequency)


p = c(0,1,2)
d = c(0,1,2,3)
q = c(0,1,2,3)

mse_list = c()

for(a in p){
  for (b in d){
    for (c in q){
      arima_models = arima(univariate_ts_train_full, order = c(a,b,c))
      predict_arima_models = predict(arima_models, n.ahead = 83)
      mse_value = mean((predict_arima_models$pred - datatouse_full_test$Frequency)^2)
      mse_list = append(mse_list, mse_value)
      }
    }
}

print(mse_list)
print(min(mse_list))
print(which(mse_list == min(mse_list)))

new_p = 0
new_d = 2
new_q = 2
arima_full = arima(datatouse_full_ts, order = c(0,2,2))
print(arima_full)
ts.plot(datatouse_full_ts)
arima_full_cv_fit = datatouse_full_ts - residuals(arima_full)
points(arima_full_cv_fit, type = "l", col=2, lty = 2)
AIC(arima_full)
BIC(arima_full)
datatouse_full_train = datatouse_full[1:round(4*(nrow(datatouse_full)/5)),]
datatouse_full_test = datatouse_full[-(1:round(4*(nrow(datatouse_full)/5))),]
univariate_ts_train_full = as.ts(datatouse_full_train$Frequency)
```

```r
p = c(0,1,2)
d = c(0,1,2)
q = c(0,1,2)
p2 = c(0,1)
d2 = c(0,1)
q2 = c(0,1)
s = c(0,12)

mse_list2 = c()
count = 0

for(a in p){
  for (b in d){
    for (c in q){
      for (d in p2){
        for (e in d2){
          for (f in q2){
            for (g in s){
              sarima_models = Arima(univariate_ts_train_full, order = c(a,b,c), seasonal = list(order=c
              predict_sarima_models = predict(sarima_models, n.ahead = 83)
              mse_value = mean((predict_sarima_models$pred - datatouse_full_test$Frequency)^2)
              mse_list2 = append(mse_list2, mse_value)
              # print(sprintf("this is the currect values: %s, %s, %s, %s, %s, %s, %s", a,b,c,d,e,f,g))
              count = count + 1
            }
          }
        }
      }
      }
    }
}

print(mse_list2)
print(min(mse_list2))
print(which(mse_list2 == min(mse_list2)))

new_p = 0
new_d = 0
new_q = 1
new_p2 = 0
new_d2 = 1
new_q2 = 1
s = 0

print(count)
sarima_model1 = Arima(datatouse_full_ts, order = c(0,0,1), seasonal = list(order=c(0,1,1), period=0))
print(sarima_model1)
ts.plot(datatouse_full_ts)
sarima_full_cv_fit = datatouse_full_ts - residuals(sarima_model1)
points(sarima_full_cv_fit, type = "l", col=2, lty = 2)
AIC(sarima_model1)
BIC(sarima_model1)
p = c(0,1,2)
```

```
d = c(0,1,2)
q = c(0,1,2)
p2 = c(0,1)
d2 = c(0,1)
q2 = c(0,1)
s = c(0,12)

mse_list_full = rep(0,144)

for (i in 1:9){
  start = 1
  end = i * round(nrow(datatouse_full)/10)
  end2 = (i+1) * round(nrow(datatouse_full)/10)
  if (end2 > nrow(datatouse_full)){
    end2 = dim(datatouse_full)[1]
  }
  datatouse_full_train = datatouse_full[start:end,]
  datatouse_full_test = datatouse_full[(end+1):end2,]
  print(sprintf("this is the number of rows for the training dataset %s: %s", i, nrow(datatouse_full_tra
  print(sprintf("this is the number of rows for the testing dataset %s: %s", i, nrow(datatouse_full_tes

  univariate_ts_train = as.ts(datatouse_full_train$Frequency)
  ts.plot(univariate_ts_train)

  mse_list_nested = c()

  count = 0
  for(a in p){
    for (b in d){
      for (c in q){
        for (d in p2){
          for (e in d2){
            for (f in q2){
              for (g in s){
                sarima_models = Arima(univariate_ts_train, order = c(a,b,c), seasonal = list(order=c(d,
                predict_sarima_models = predict(sarima_models, n.ahead = 83)
                mse_value = mean((predict_sarima_models$pred - datatouse_full_test$Frequency)^2)
                mse_list_nested = append(mse_list_nested, mse_value)
                count = count + 1
                if (count == 74){
                  print(print(sprintf("this is the currect values: %s, %s, %s, %s, %s, %s, %s", a,b,c,d
                }
              }
            }
          }
        }
      }
    }
  }
  print(sprintf("this is total count: %s", count))
  mse_list_nested = mse_list_nested[1:144]
  mse_list_full = mse_list_full + mse_list_nested
  print(sprintf("there are total %s different models", length(mse_list_nested)))
```

```
}

mse_list_full = mse_list_full/9
print(mse_list_full)
print(min(mse_list_full))
print(which(mse_list_full == min(mse_list_full)))


new_p = 1
new_d = 1
new_q = 1
new_p2 = 1
new_d2 = 0
new_q2 = 0
s = 12
sarima_model2 = Arima(datatouse_full_ts, order = c(1,1,1), seasonal = list(order=c(1,0,0), period=12))
print(sarima_model2)
ts.plot(datatouse_full_ts)
sarima_full_cv_fit_2 = datatouse_full_ts - residuals(sarima_model2)
points(sarima_full_cv_fit_2, type = "l", col=2, lty = 2)
AIC(sarima_model2)
BIC(sarima_model2)
glmts = tsglm(datatouse_full_ts, distr = "nbinom", link = "log")
summary(glmts)
plot(glmts)
ts.plot(datatouse_full_ts)
gsarima_model_fit = datatouse_full_ts - residuals(glmts)
points(gsarima_model_fit, type = "l", col=2, lty=2)
AIC(glmts)
BIC(glmts)
```