R Notebook

Code

Hide

```
library("dplyr")
```

```

Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

Hide

```
library("plyr")
```

```
------------------------------------------------------------------------------
------------------------------
You have loaded plyr after dplyr - this is likely to cause problems.
If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
library(plyr); library(dplyr)
------------------------------------------------------------------------------
------------------------------

Attaching package: ‘plyr’

The following objects are masked from ‘package:dplyr’:

    arrange, count, desc, failwith, id, mutate, rename, summarise, summarize
```

Hide

```
library("readr")
library("readxl")
library("data.table")
```

```
Registered S3 method overwritten by 'data.table':
  method           from
  print.data.table
data.table 1.14.2 using 1 threads (see ?getDTthreads).  Latest news: r-datatable.com
**********
This installation of data.table has not detected OpenMP support. It should still work
but in single-threaded mode.
This is a Mac. Please read https://mac.r-project.org/openmp/. Please engage with Appl
e and ask them for support. Check r-datatable.com for updates, and our Mac instructio
ns here: https://github.com/Rdatatable/data.table/wiki/Installation. After several ye
ars of many reports of installation problems on Mac, it's time to gingerly point out
that there have been no similar problems on Windows or Linux.
**********

Attaching package: ‘data.table’

The following objects are masked from ‘package:dplyr’:

    between, first, last
```

Hide

```
library(fpp)
```

```
Loading required package: forecast
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
Loading required package: fma

Attaching package: ‘fma’

The following object is masked from ‘package:plyr’:

    ozone

Loading required package: expsmooth
Loading required package: lmtest
Loading required package: zoo

Attaching package: ‘zoo’

The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric

Loading required package: tseries

    ‘tseries’ version: 0.10-50

    ‘tseries’ is a package for time series analysis and computational finance.

    See ‘library(help="tseries")’ for details.
```

Hide

```
library(fpp2)
```

```
── Attaching packages ──────────────────────────────────────────────────────────
──────────────────── fpp2 2.4 ──
✓ ggplot2 3.3.5


Attaching package: ‘fpp2’

The following objects are masked from ‘package:fpp’:

    ausair, ausbeer, austa, austourists, debitcards, departures, elecequip, euretail,
guinearice, oil,
    sunspotarea, usmelec
```

Hide

```
library(tseries)
library(forecast)
library(ggplot2)
library(stats)
library(TSA)
```

```
Registered S3 methods overwritten by 'TSA':
  method       from
  fitted.Arima forecast
  plot.Arima   forecast

Attaching package: ‘TSA’

The following object is masked from ‘package:readr’:

    spec

The following objects are masked from ‘package:stats’:

    acf, arima

The following object is masked from ‘package:utils’:

    tar
```

Hide

```
path <- "/Users/shulundong/Desktop/Time Series Final Project/date.csv"
path2 <- "/Users/shulundong/Desktop/Time Series Final Project/weather_data_24hr.csv"
df <- fread(path, select = c("CRASH_DATE"))
df2 <- fread(path2, select = c("date", "totalprecipIn", "visibilityMiles"))
```

#Seperation of Date from 2016-2022 on Weather Dataset

Hide

```
df2_wk_seg <- df2[df2$date >= "2016-01-01"]
df2_wk_seg <- df2_wk_seg[df2_wk_seg$date < "2022-01-03"]
df2_wk_seg_fc <- df2[df2$date >= "2022-01-03"]
df2_wk_seg_fc <- df2_wk_seg_fc[df2_wk_seg_fc$date < "2022-05-01"]
```

Hide

```
df2_wk_seg <- df2_wk_seg[order(df2_wk_seg$date)]
df2_wk_seg$week <- as.Date(cut(df2_wk_seg$date, "week"))
df2_wk_seg_fc <- df2_wk_seg_fc[order(df2_wk_seg_fc$date)]
df2_wk_seg_fc$week <- as.Date(cut(df2_wk_seg_fc$date, "week"))
```

Hide

```
df2_wk_gp <- df2_wk_seg %>% group_by(week) %>%dplyr:: summarise(across(everything(), mean))
df2_wk_gp_fc <- df2_wk_seg_fc %>% group_by(week) %>%dplyr:: summarise(across(everything(), mean))
```

#Seperation of Date from 2016-2022 on Crashes Dataset

Hide

```
df_wk_seg <- df[df$CRASH_DATE >= "2016-01-01"]
df_wk_seg <- df_wk_seg[df_wk_seg$CRASH_DATE < "2022-01-03"]

df_wk_seg_fc <- df[df$CRASH_DATE >= "2022-01-03"]
df_wk_seg_fc <- df_wk_seg_fc[df_wk_seg_fc$CRASH_DATE < "2022-05-01"]
```

Hide

```
df_wk_seg <- df_wk_seg[order(df_wk_seg$CRASH_DATE)]
```

Hide

```
df_wk_seg$week <- as.Date(cut(df_wk_seg$CRASH_DATE, "week"))
```

Hide

Hide

```
df_wk_gp <- df_wk_seg %>% group_by(week) %>% dplyr::summarize(crashes = n())
```

#Validation set h=17

Hide

```
df_wk_seg_fc <- df_wk_seg_fc[order(df_wk_seg_fc$CRASH_DATE)]
df_wk_seg_fc$week <- as.Date(cut(df_wk_seg_fc$CRASH_DATE, "week"))
```

Hide

```
df_wk_gp_fc <- df_wk_seg_fc %>% group_by(week) %>% dplyr::summarize(crashes = n())
```

#Joining DF1 & Df2

Hide

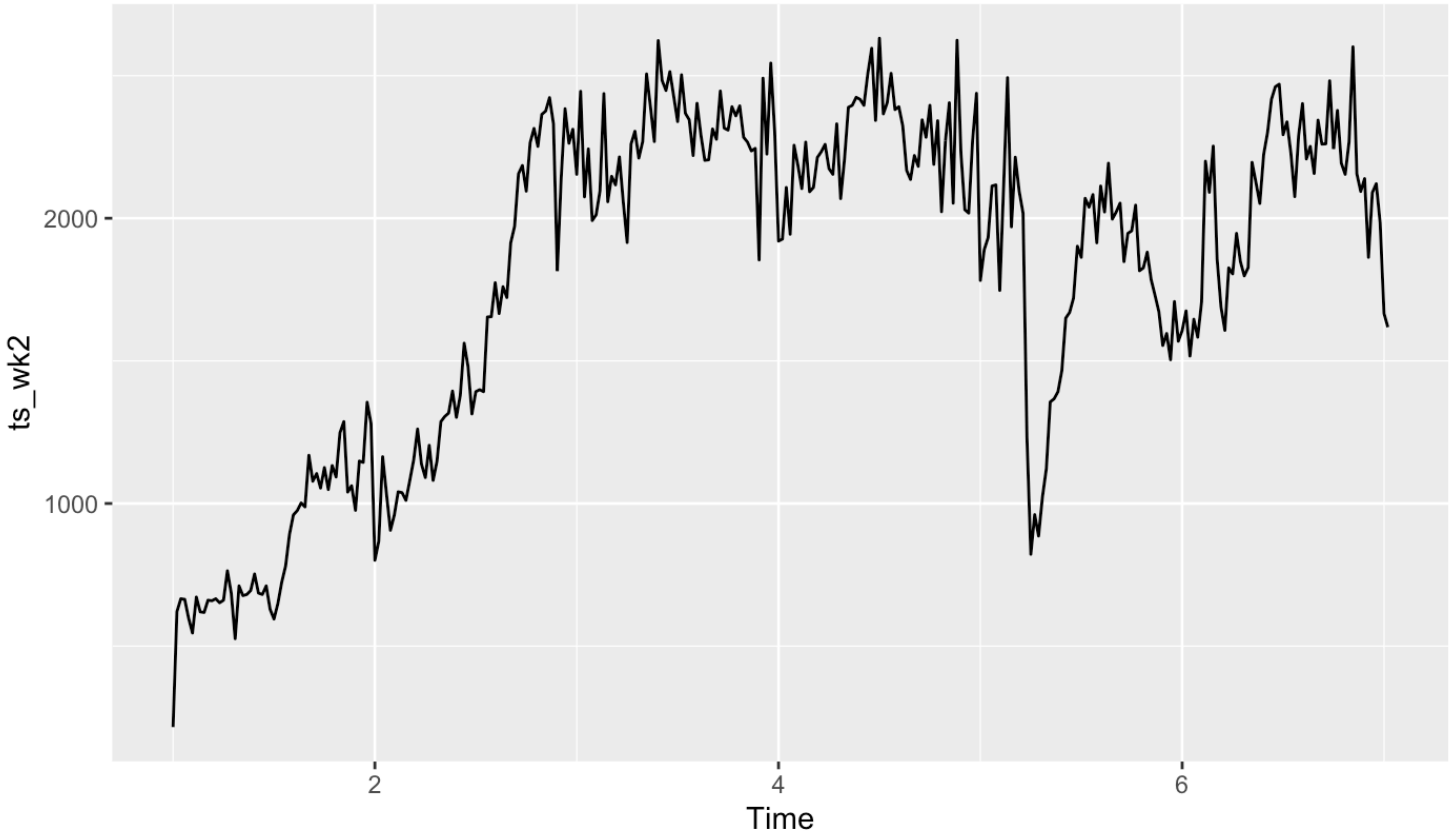
```
df_combine <- merge(df2_wk_gp,df_wk_gp,by="week")
df_combine <- subset(df_combine, select = c('week', 'totalprecipIn', 'crashes', 'visi
bilityMiles'))
```

Hide

```
ts_wk2 <- ts(df_combine$crashes, frequency = 52)
ts_wk3 <- ts(df_combine$totalprecipIn, frequency = 52)
ts_wk4 <- ts(df_combine$visibilityMiles, frequency = 52)
```

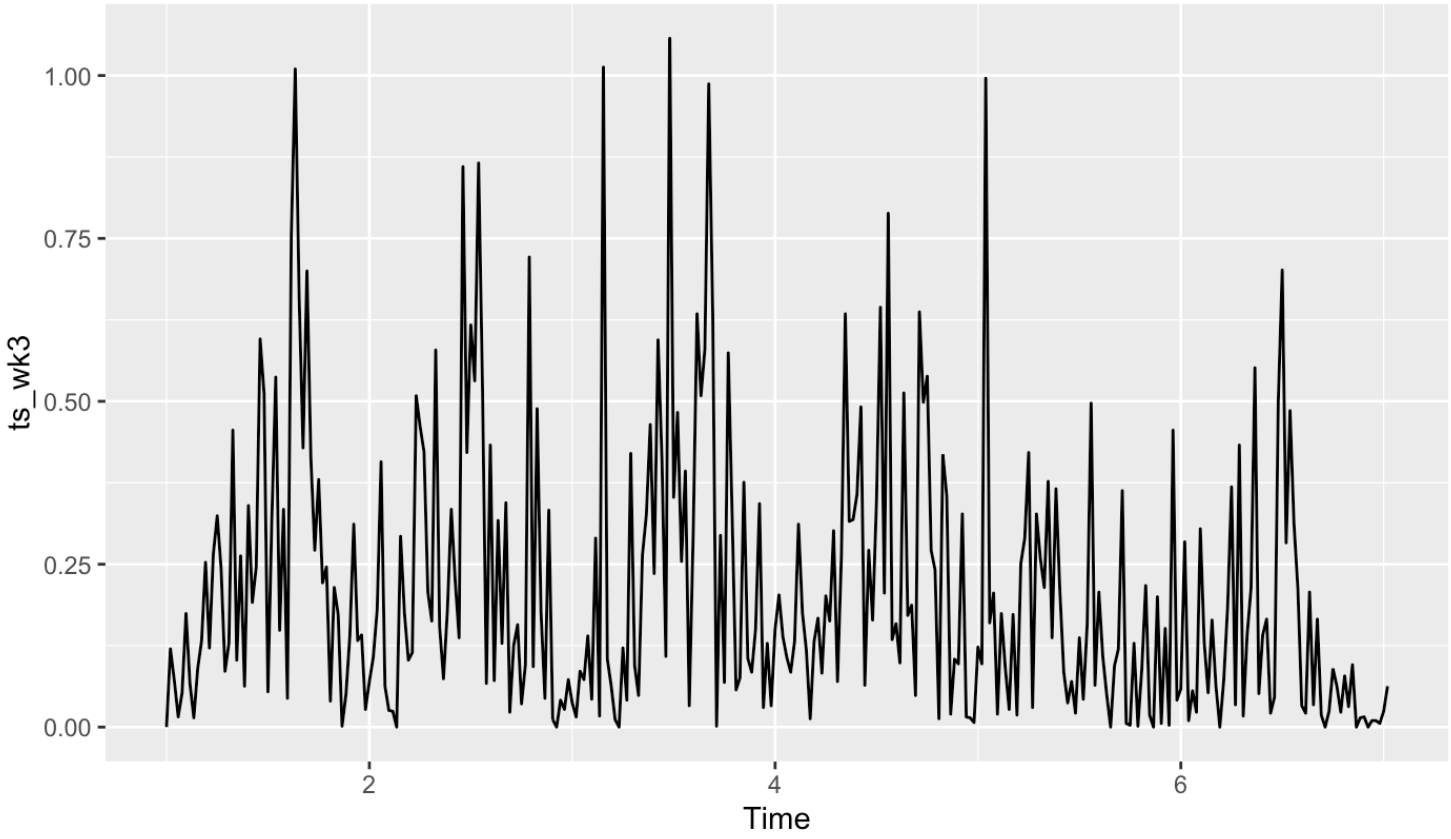
Hide

```
autoplot(ts_wk2)
```



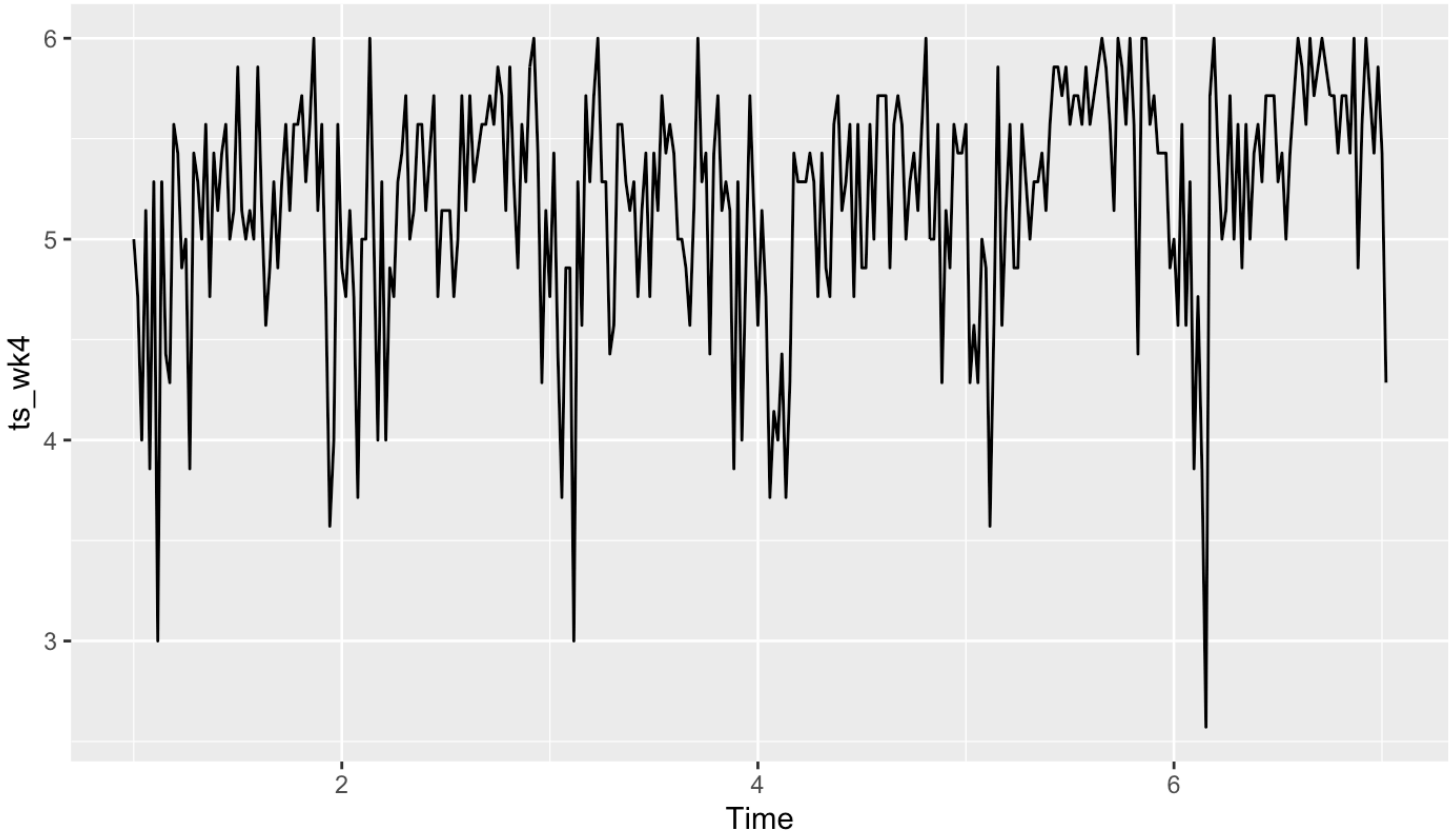
Hide

```
autoplot(ts_wk3)
```



Hide

```
autoplot(ts_wk4)
```



It looks like the series has a trend and some seasonality. There is a significant drop which is most likely caused by external event COVID-19.

Hide

```
fit_naive <- snaive(ts_wk2,lambda = 1.555455)
fit_naive
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
7.038462	1517	365.5403	2284.677	-789.64396	2636.270
7.057692	1646	613.6169	2388.528	-584.49317	2732.418
7.076923	1583	499.7549	2337.542	-690.09996	2685.157
7.096154	1707	715.6234	2438.351	-468.35544	2778.705
7.115385	2200	1409.6653	2854.674	869.33880	3168.888
7.134615	2091	1268.0134	2760.803	677.07955	3080.432
7.153846	2253	1477.0937	2900.634	956.31592	3212.288

7.173077	1857	945.0386	2562.618	65.03806	2894.562
7.192308	1689	686.1977	2423.604	-504.40397	2764.994
7.211538	1607	544.3878	2356.907	-651.26601	2703.094

1-10 of 104 rows Previous **1** 2 3 4 5 6 … 11 Next

Hide

```
fc_naive <- forecast(fit_naive, h=17)
```

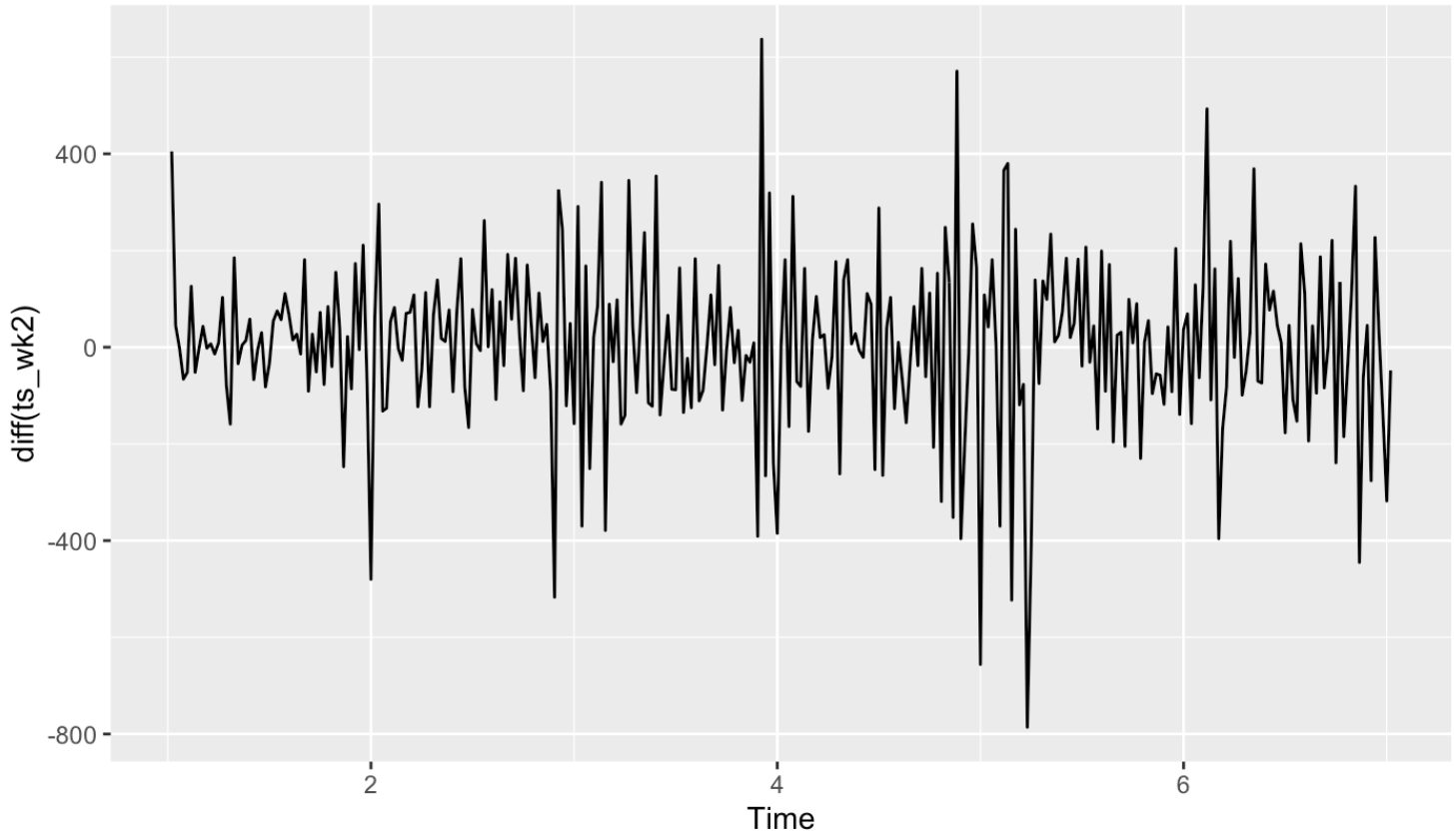
Hide

```
err_naive <- sqrt(mean((fc_naive$mean - df_wk_gp_fc$crashes)^2))
err_naive
```

```
[1] 205.5478
```

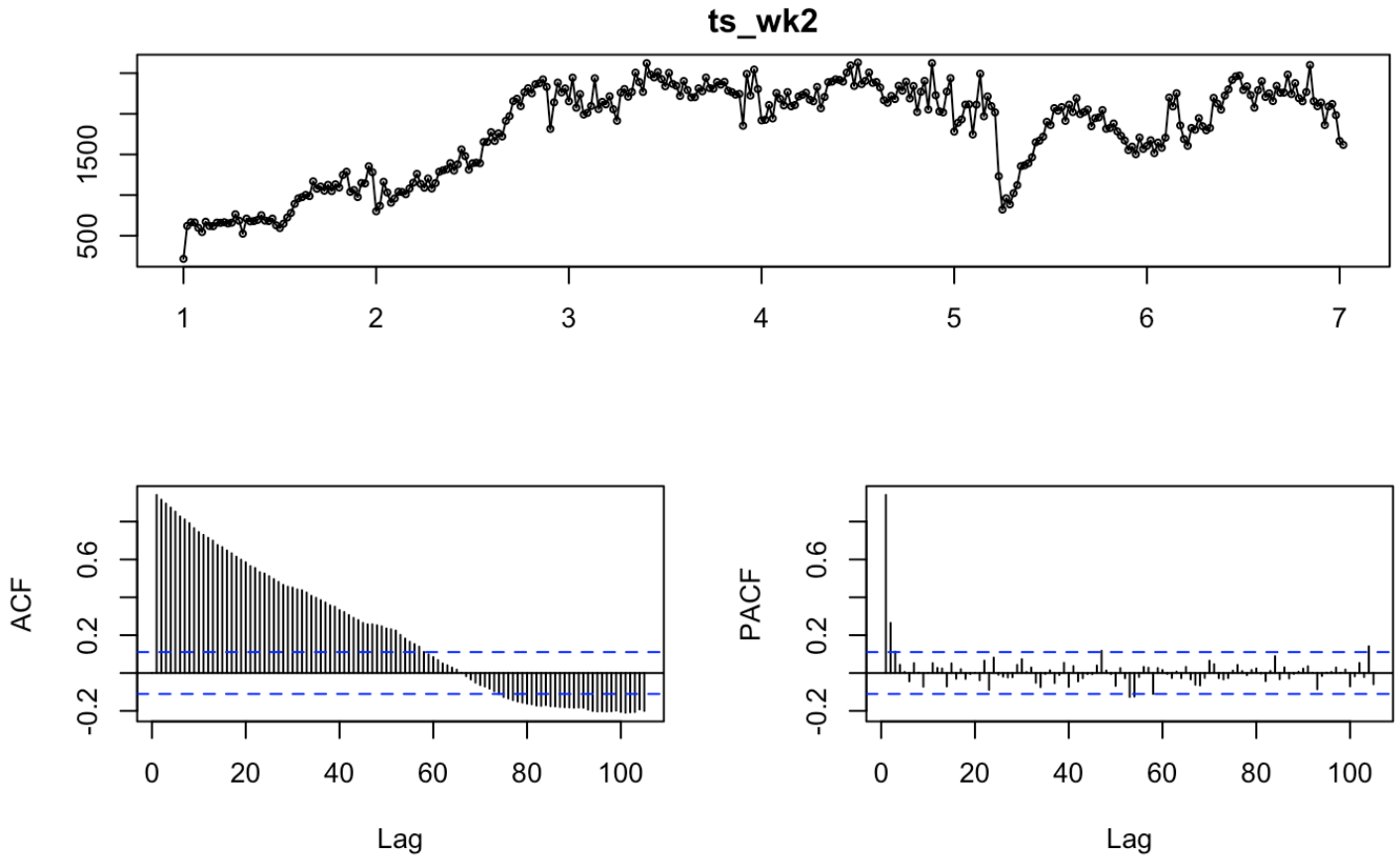
Hide

```
autoplot(diff(ts_wk2))
```



Hide

```
tsdisplay(ts_wk2)
```



Hide

```
BoxCox.lambda(ts_wk2)
```

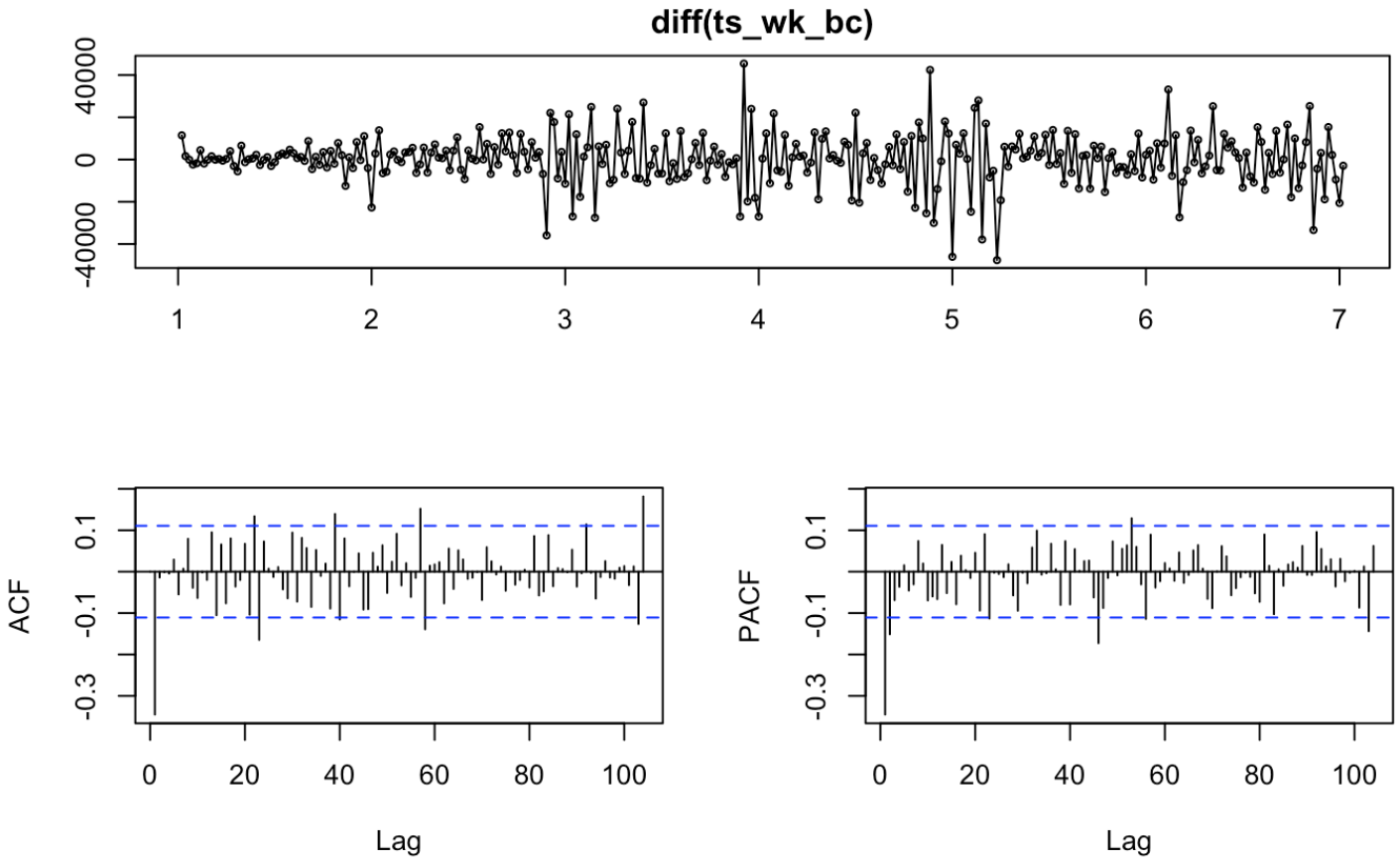
```
[1] 1.555455
```

Hide

```
ts_wk_bc <- BoxCox(ts_wk2, lambda = 1.555455)
```

Hide

```
tsdisplay(diff(ts_wk_bc))
```



Hide

```
kpss.test(ts_wk2)
```

```
Warning in kpss.test(ts_wk2) : p-value smaller than printed p-value
```

```

    KPSS Test for Level Stationarity

data:  ts_wk2
KPSS Level = 2.3714, Truncation lag parameter = 5, p-value = 0.01
```

Hide

```
adf.test(ts_wk2)
```

```
    Augmented Dickey-Fuller Test

data:  ts_wk2
Dickey-Fuller = -1.9024, Lag order = 6, p-value = 0.6176
alternative hypothesis: stationary
```

Both tests show that the original series is not stationary.

Hide

```
kpss.test(diff(ts_wk_bc))
```

```
Warning in kpss.test(diff(ts_wk_bc)) :
  p-value greater than printed p-value
```

```
    KPSS Test for Level Stationarity

data:  diff(ts_wk_bc)
KPSS Level = 0.15216, Truncation lag parameter = 5, p-value = 0.1
```

Hide

```
adf.test(diff(ts_wk_bc))
```

```
Warning in adf.test(diff(ts_wk_bc)) :
  p-value smaller than printed p-value
```

```
    Augmented Dickey-Fuller Test

data:  diff(ts_wk_bc)
Dickey-Fuller = -7.7343, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary
```

After first order differencing, we get a stationary series.

We firstly tried the auto arima to model the data.

Hide

```
fit_wk_auto <- auto.arima(ts_wk2, lambda = 1.555455, seasonal=TRUE)
```

Hide

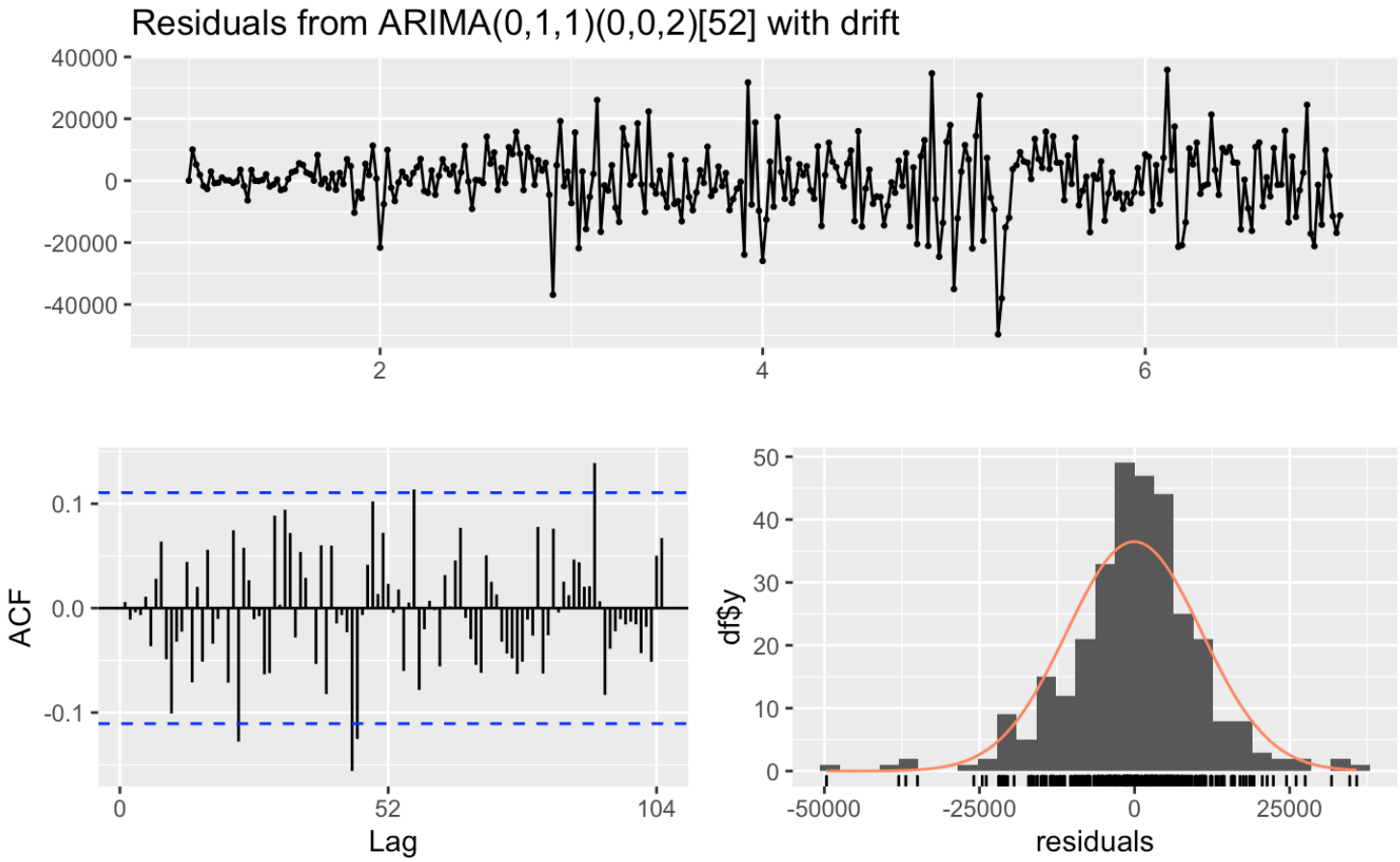
```
fit_wk_auto
```

```
Series: ts_wk2
ARIMA(0,1,1)(0,0,2)[52] with drift
Box Cox transformation: lambda= 1.555455

Coefficients:
          ma1    sma1    sma2     drift
      -0.4243  0.0773  0.2260  236.0806
s.e.   0.0525  0.0579  0.0683  433.6952

sigma^2 = 119664852:  log likelihood = -3355.98
AIC=6721.95   AICc=6722.15   BIC=6740.68
```

Hide

```
checkresiduals(fit_wk_auto)
```

```

    Ljung-Box test

data:  Residuals from ARIMA(0,1,1)(0,0,2)[52] with drift
Q* = 72.928, df = 59, p-value = 0.105

Model df: 4.   Total lags used: 63
```



The ACF shows several significant spikes within the lags. However, the Ljung-Box test gave p value > 0.05 and therefore, the residual looks much like the white noise.

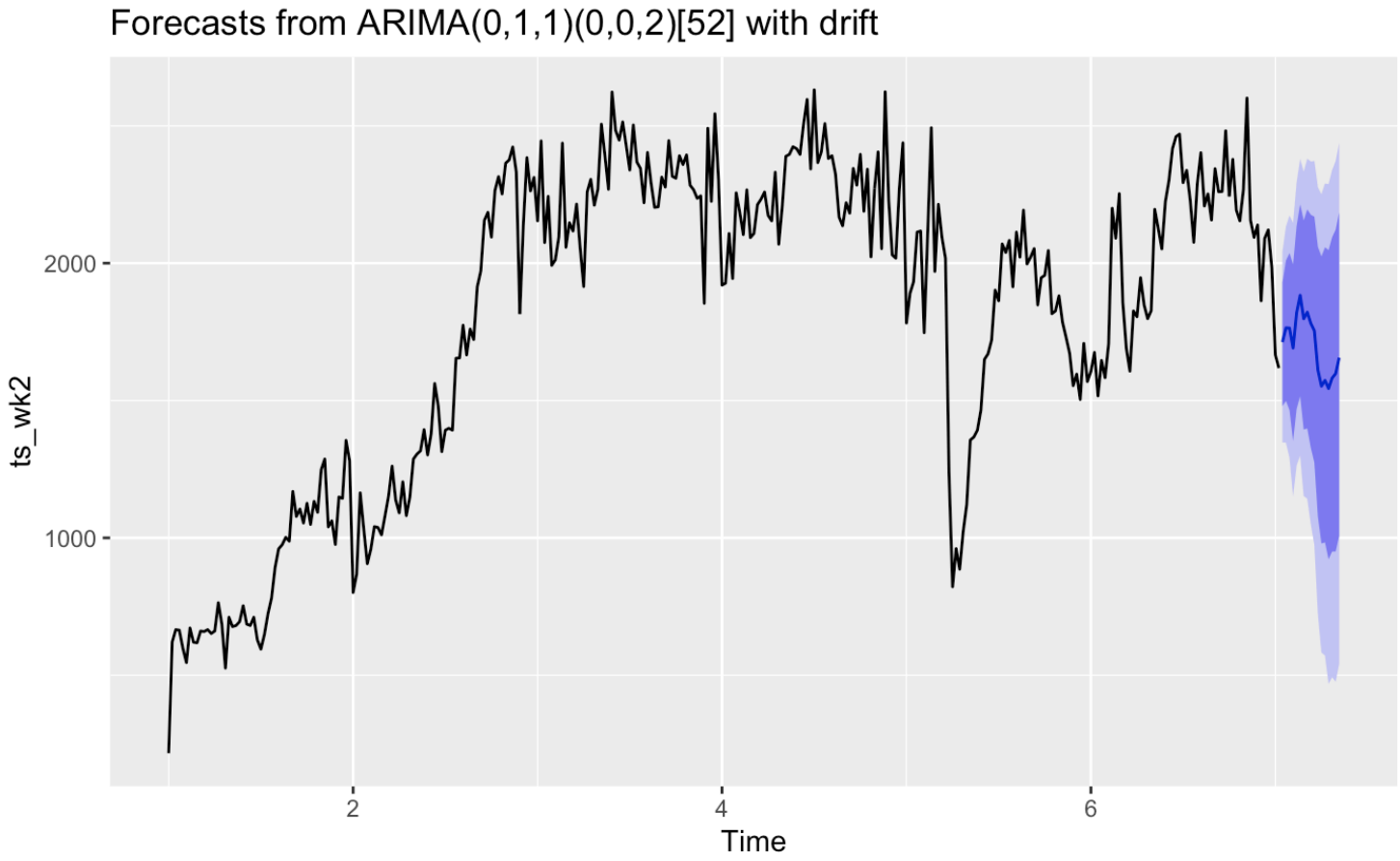
Forecast next 16 weeks crash number with auto arima.

Hide

```
fc_wk2_auto <- forecast(fit_wk_auto, h=17)
```

Hide

```
autoplot(fc_wk2_auto)
```



Evaluation - RMSE

Hide

```
err_auto <- sqrt(mean((fc_wk2_auto$mean - df_wk_gp_fc$crashes)^2))
err_auto
```

```
[1] 276.825
```

We also want to try Exponential Smoothing method. As you have already found ets() does not work for high seasonal periods. The reason is that there are too many degrees of freedom associated with the seasonality — with period 52, there would be 51 degrees of freedom just on the seasonal component which makes little sense.

The 'smoothing' part of this method brushes over high and low variations. As the forecast graph shows a smooth line of data, it's important to note that spikes in data aren't necessarily represented.

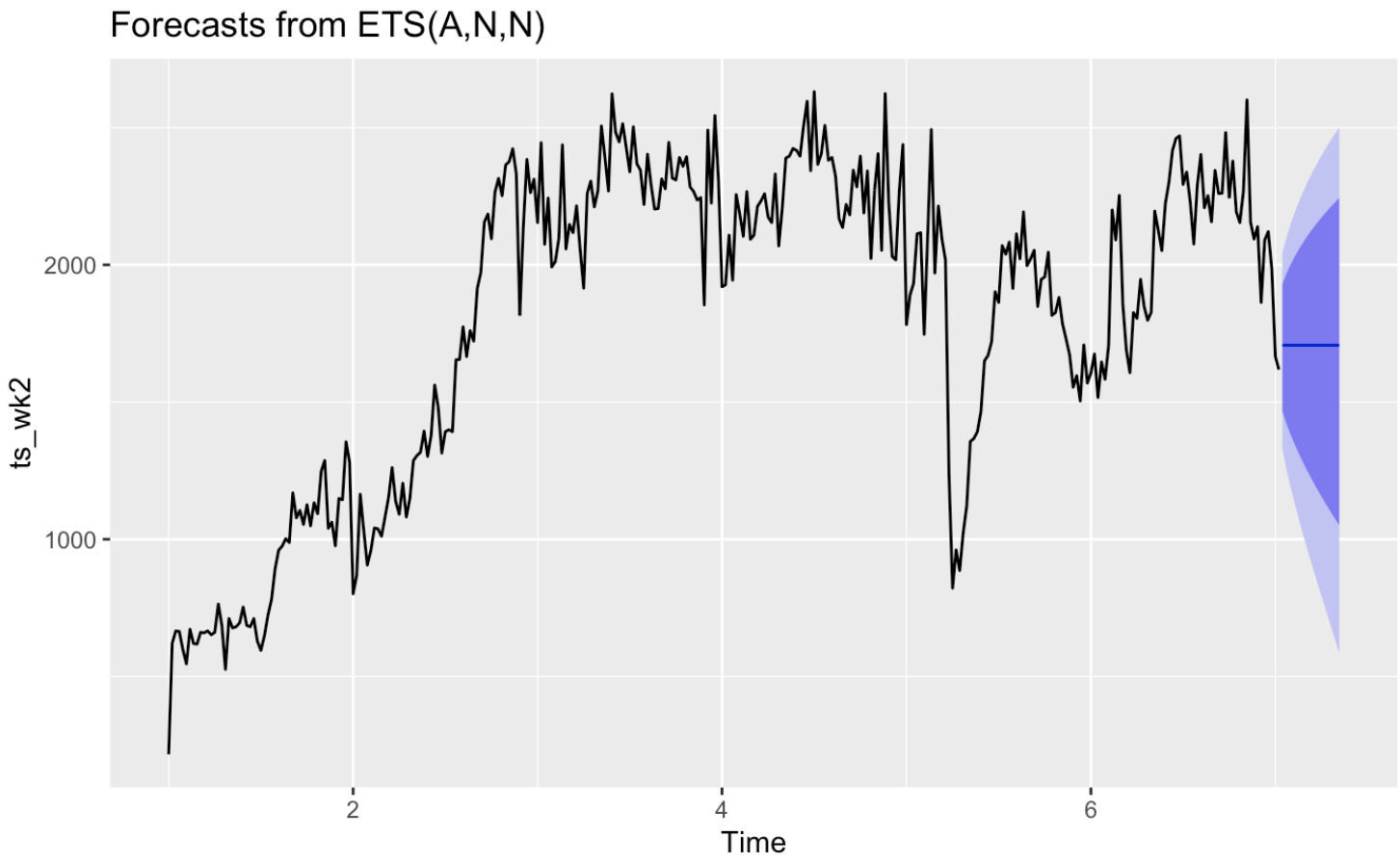
Hide

```
fit_ets <- ets(ts_wk2, lambda = "auto")
```

```
Warning in ets(ts_wk2, lambda = "auto") :
  I can't handle data with frequency greater than 24. Seasonality will be ignored. Tr
y stlf() if you need seasonal forecasts.
```

Hide

```
autoplot(forecast(fit_ets, h=17))
```



This seasonality rules out the use of ets models which can't handle data with frequency greater than 24, unless used in combination with STL (Seasonal and Trend decomposition using Loess)

Holt Winter model

Hide

```
fit_hw_add <- hw(ts_wk2, damped=TRUE, seasonal='additive', h=17)
```

```
Error in ets(x, "AAA", alpha = alpha, beta = beta, gamma = gamma, phi = phi,  :
  Frequency too high
```

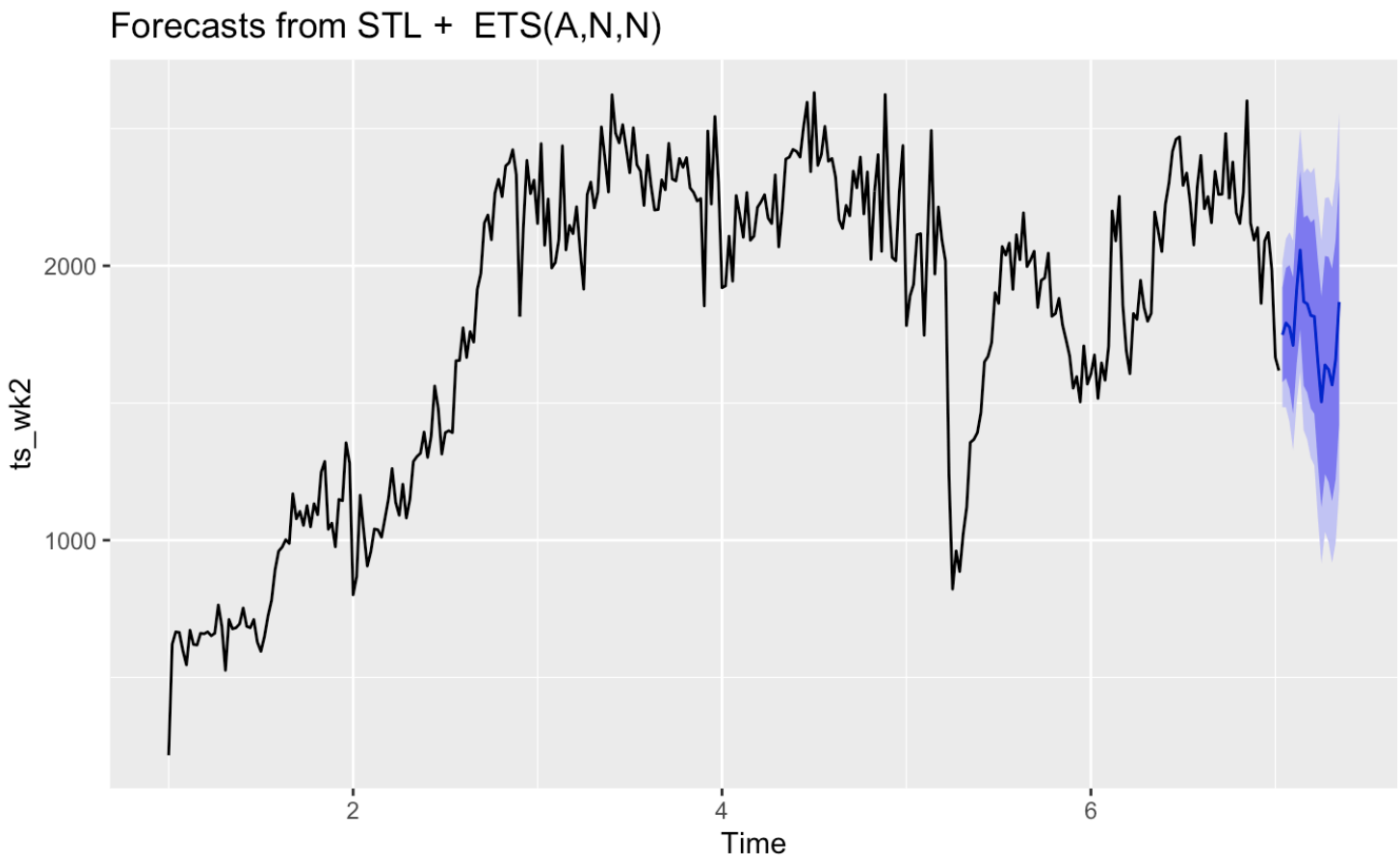
STLF can be defined as Seasonal and Trend decomposition using Loess Forecasting model. The following code will decompose the time series using STL, and forecast the seasonally adjusted series with ETS, and return the reseasonalised forecasts.

Hide

```
fit_stlf <- stlf(ts_wk2, method='ets', h=17)
```

Hide

```
autoplot(fit_stlf)
```



Hide

```
err_stlf <- sqrt(mean((fit_stlf$mean- df_wk_gp_fc$crashes)^2))
err_stlf
```

```
[1] 243.2953
```

We will try to model the series with TBATS.

The variance has changed a lot over time, so it needs a transformation. • The seasonality has also changed shape over time, • and there is a strong trend. This makes it an ideal series to test the tbats() function which is designed to handle these features.

Hide

```
fit_wk2_tbats <- tbats(ts_wk2)
```

Hide

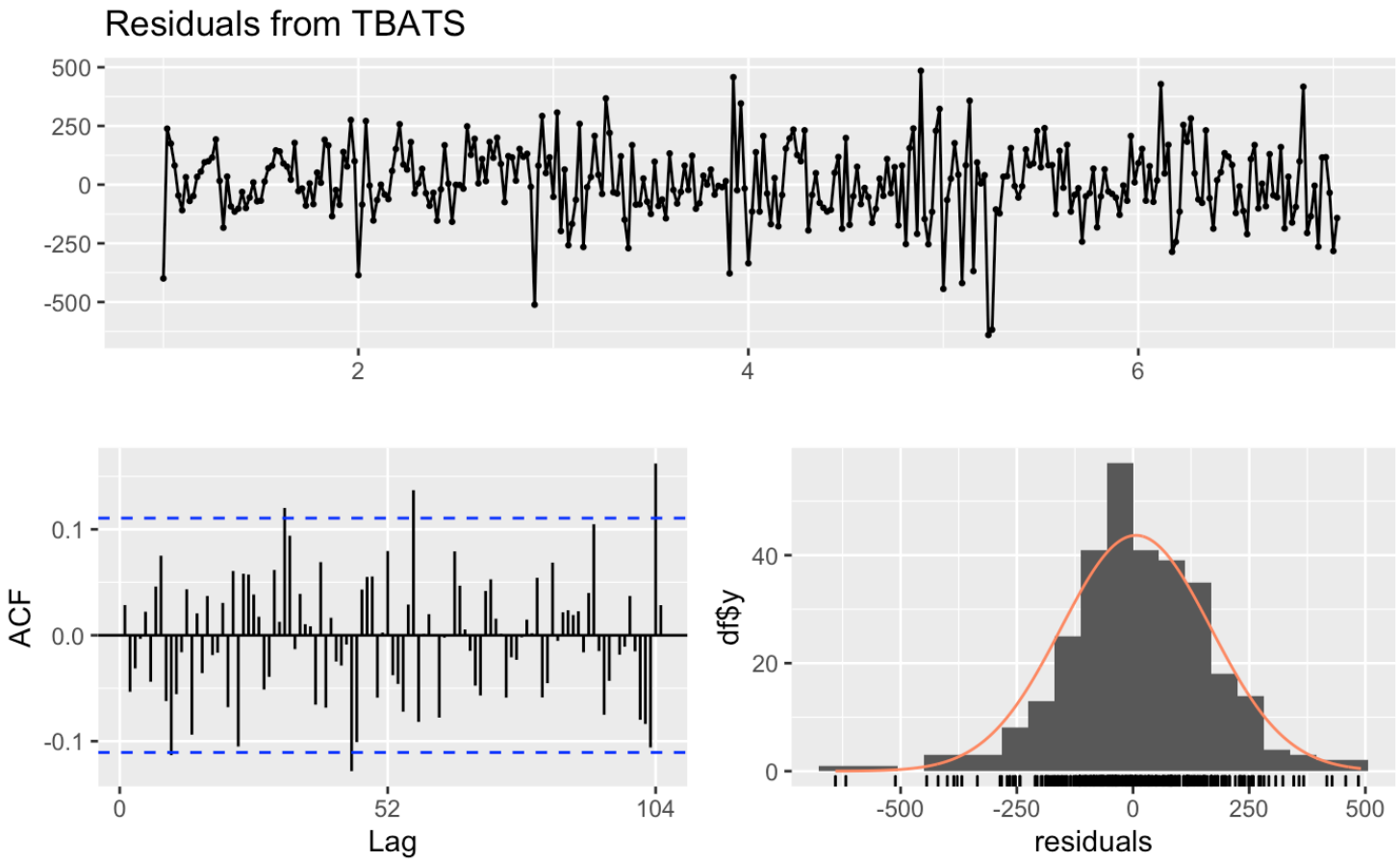
```
fit_wk2_tbats
```

```
TBATS(1, {0,0}, -, {<52,5>})

Call: tbats(y = ts_wk2)

Parameters
  Alpha: 0.5480914
  Gamma-1 Values: -0.008646206
  Gamma-2 Values: -0.007217537

Seed States:
             [,1]
 [1,]  727.285960
 [2,]  -47.979778
 [3,]    3.329315
 [4,]  -56.997318
 [5,]  -27.286547
 [6,]   17.268421
 [7,] -149.014345
 [8,]    4.959360
 [9,]   38.502313
[10,]  -32.771923
[11,]  -29.996358

Sigma: 161.2386
AIC: 5025.361
```

Hide

```
checkresiduals(fit_wk2_tbats)
```

```
    Ljung-Box test

data:  Residuals from TBATS
Q* = 75.769, df = 49, p-value = 0.0084

Model df: 14.   Total lags used: 63
```

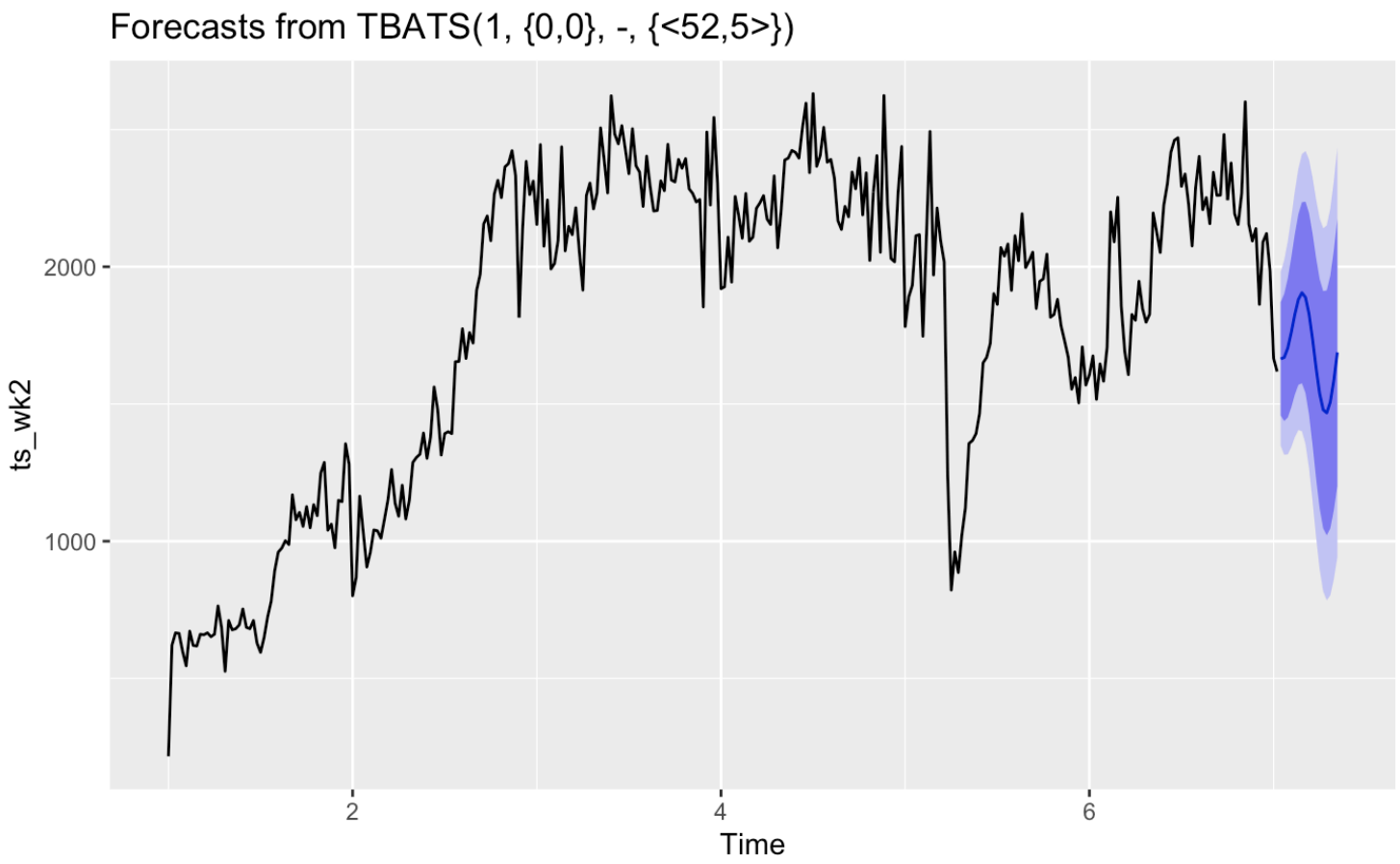


Hide

```
fc_wk2_tbats <- forecast(fit_wk2_tbats, h=17)
```

Hide

```
autoplot(fc_wk2_tbats)
```



Hide

```
err_tbats <- sqrt(mean((fc_wk2_tbats$mean- df_wk_gp_fc$crashes)^2))
err_tbats
```

```
[1] 290.4389
```

We used Fourier transformation to transform a time-domain representation to a frequency- domain representation, we will use the Fourier transform to decompose the series since it got a dynamic seasonality.

We firstly did the spectrum analysis. Spectral analysis will identify the correlation of sine and cosine functions of different frequency with the observed data. If a large correlation (sine or cosine coefficient) is identified, you can conclude that there is a strong periodicity of the respective frequency (or period) in the data.

Spectral analysis is appropriate for the analysis of stationary time series and for identifying periodic signals that are corrupted by noise. However,spectral analysis is not suitable for non-stationary applications. One goal of spectral analysis is to identify the important frequencies (or periods) in the observed series.

Hide

```
kpss.test(ts_wk2)
```

```
Warning in kpss.test(ts_wk2) : p-value smaller than printed p-value
```

```

    KPSS Test for Level Stationarity

data:  ts_wk2
KPSS Level = 2.3714, Truncation lag parameter = 5, p-value = 0.01
```

Hide

```
ts_wk_bc_diff <- diff(ts_wk_bc)
```

Hide

```
kpss.test(ts_wk_bc_diff)
```

```
Warning in kpss.test(ts_wk_bc_diff) :
  p-value greater than printed p-value
```

```

    KPSS Test for Level Stationarity

data:  ts_wk_bc_diff
KPSS Level = 0.15216, Truncation lag parameter = 5, p-value = 0.1
```

Hide

```
adf.test(ts_wk_bc_diff)
```

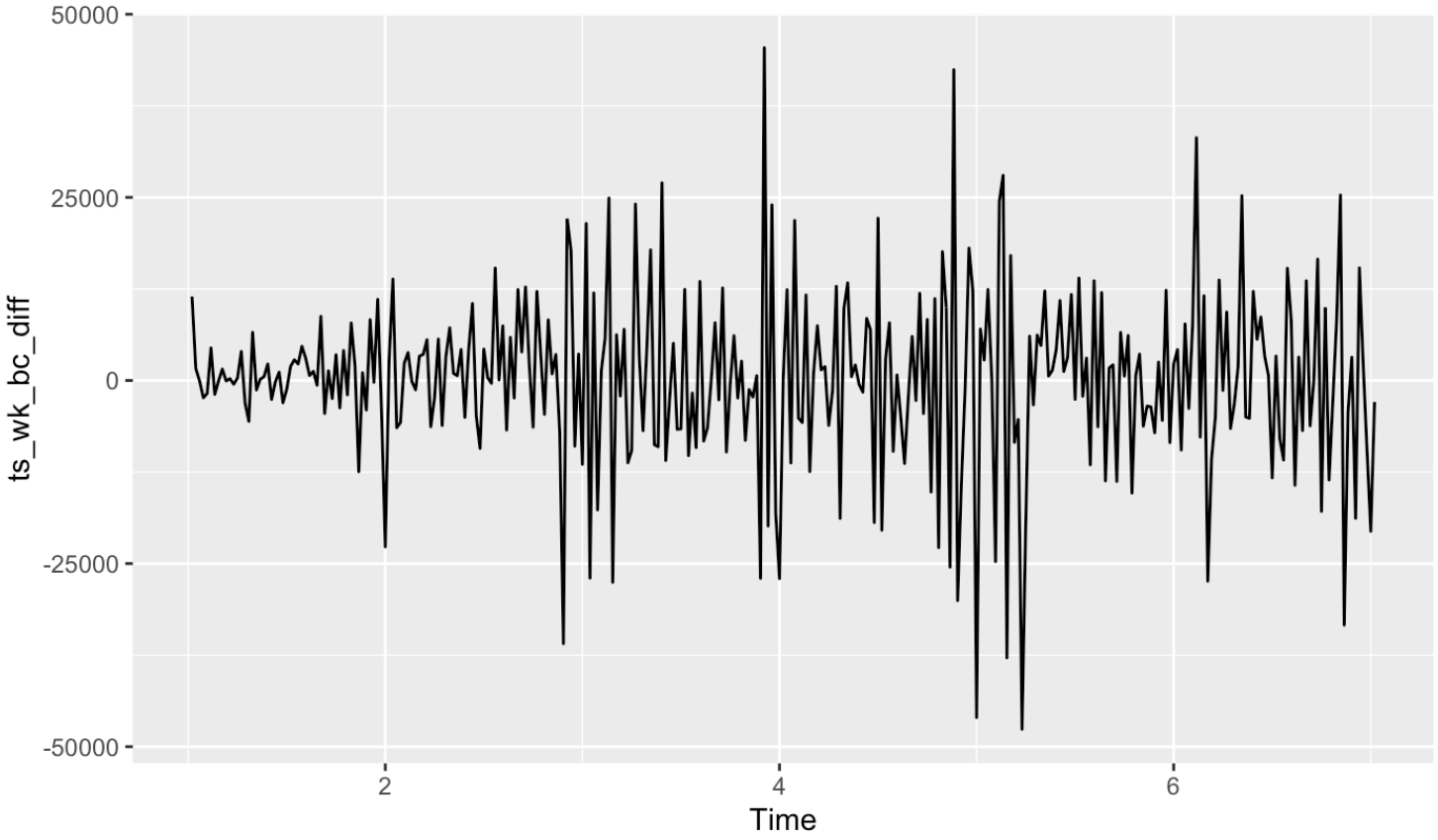
```
Warning in adf.test(ts_wk_bc_diff) :
  p-value smaller than printed p-value
```

```

    Augmented Dickey-Fuller Test

data:  ts_wk_bc_diff
Dickey-Fuller = -7.7343, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary
```

***Not sure if this is right.

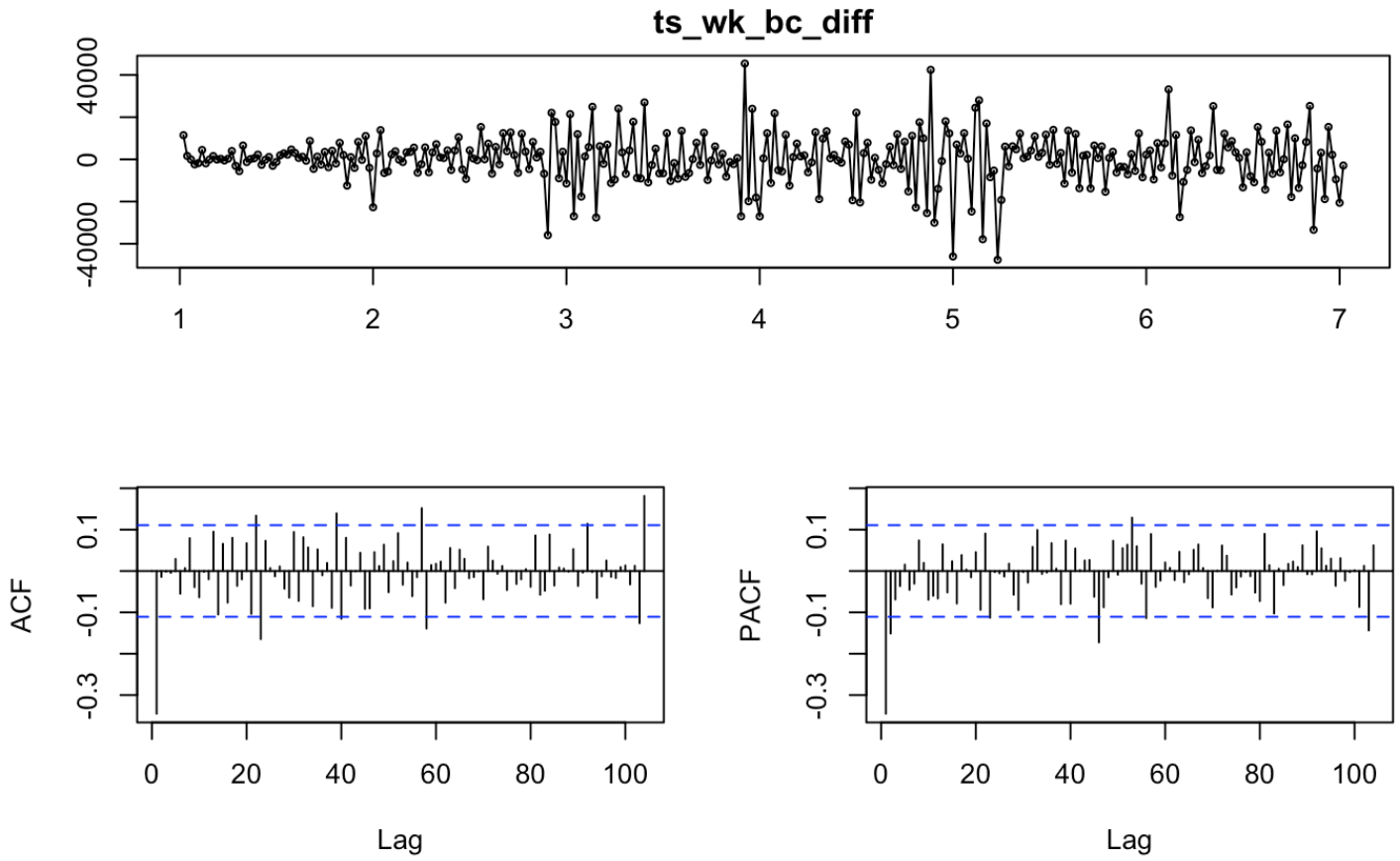
Hide

```
autoplot(ts_wk_bc_diff)
```



Hide

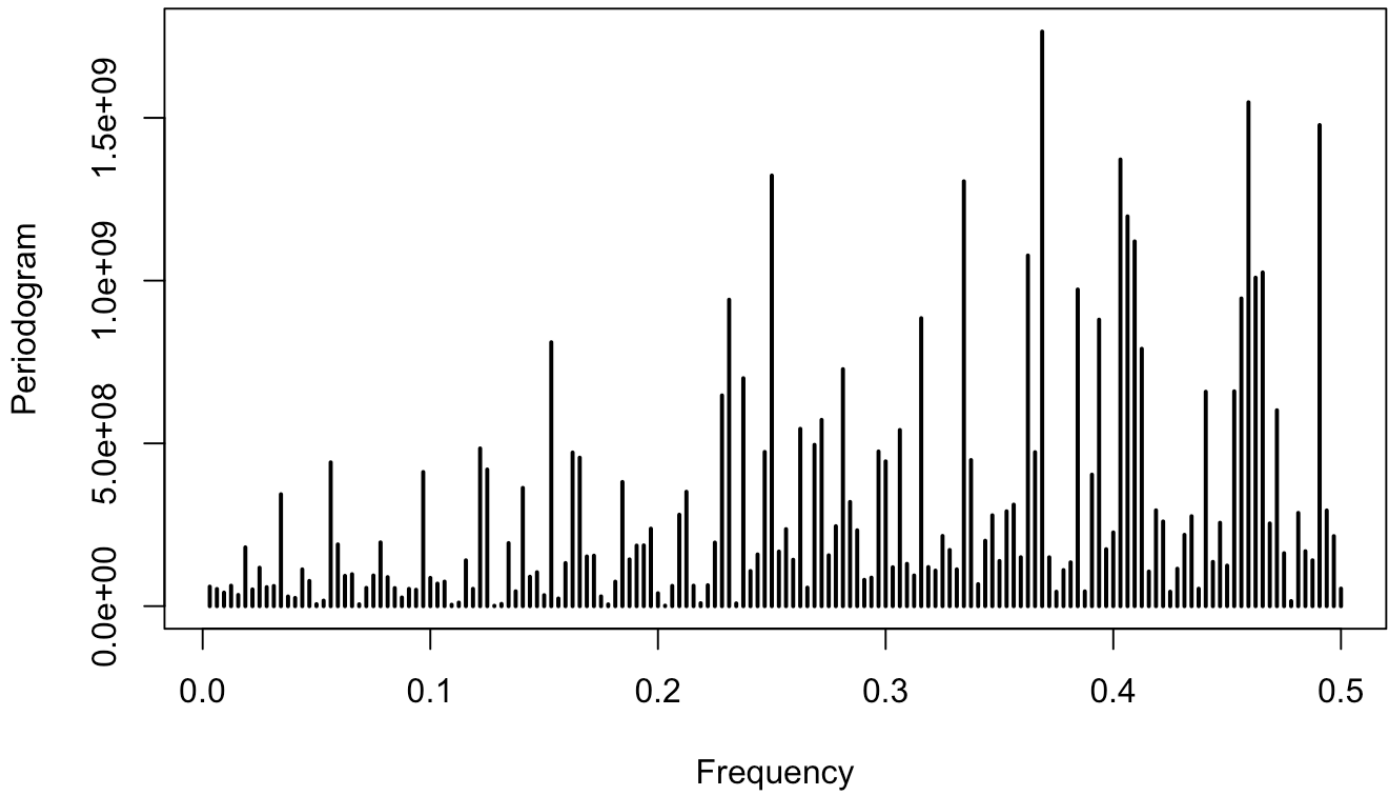
```
tsdisplay(ts_wk_bc_diff)
```



Too much noise, periodogram. Investigate one significant frequency.

Hide

```
temp <- periodogram(ts_wk_bc_diff)
```



Hide

```
max_freq <- temp$freq[which.max(temp$spec)]
max_freq
```

```
[1] 0.36875
```

Hide

```
seasonality <- 1/max_freq
seasonality
```

```
[1] 2.711864
```

Hide

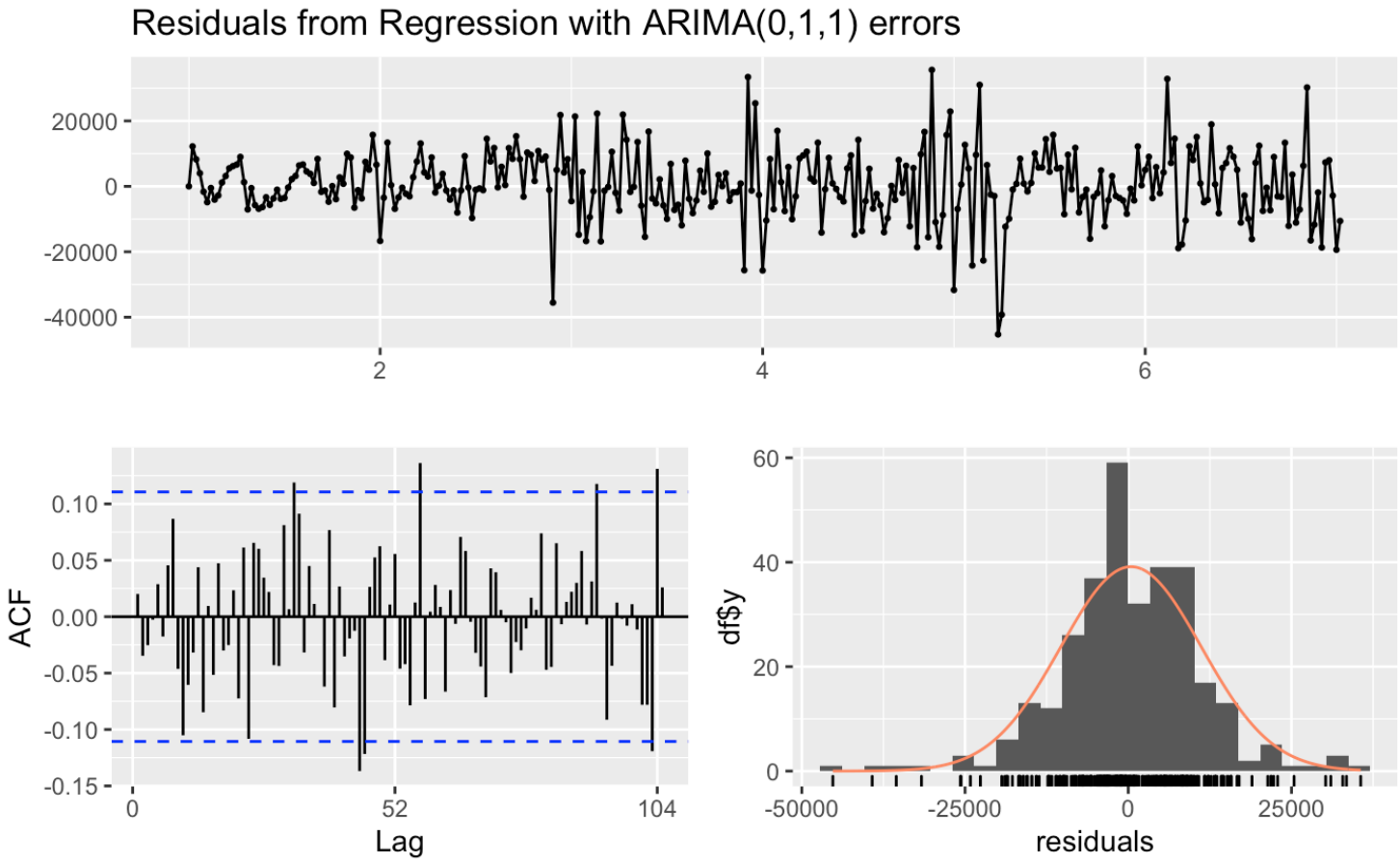
```
fit_wk2_fourier <- auto.arima(ts_wk2, xreg=fourier(ts_wk2, 5), seasonal = FALSE, lamb
da = 1.555455)
fit_wk2_fourier
```

```
Series: ts_wk2
Regression with ARIMA(0,1,1) errors
Box Cox transformation: lambda= 1.555455

Coefficients:
          ma1      S1-52      C1-52      S2-52      C2-52     S3-52      C3-52       S
4-52      C4-52       S5-52
      -0.5107  -9870.436  -2053.814  -104.0771   661.7946   938.974  -4286.944  -2426
.718  -216.6967  -1055.5990
s.e.   0.0503   3566.090   3554.647  1863.4394  1859.0888  1326.881   1325.249   1077
.830  1077.5819    940.6372
          C5-52
      2004.9948
s.e.   941.0624

sigma^2 = 120654680:  log likelihood = -3350.9
AIC=6725.8   AICc=6726.84   BIC=6770.76
```

Hide

```
checkresiduals(fit_wk2_fourier)
```

```


    Ljung-Box test

data:  Residuals from Regression with ARIMA(0,1,1) errors
Q* = 77.537, df = 52, p-value = 0.01237

Model df: 11.   Total lags used: 63
```

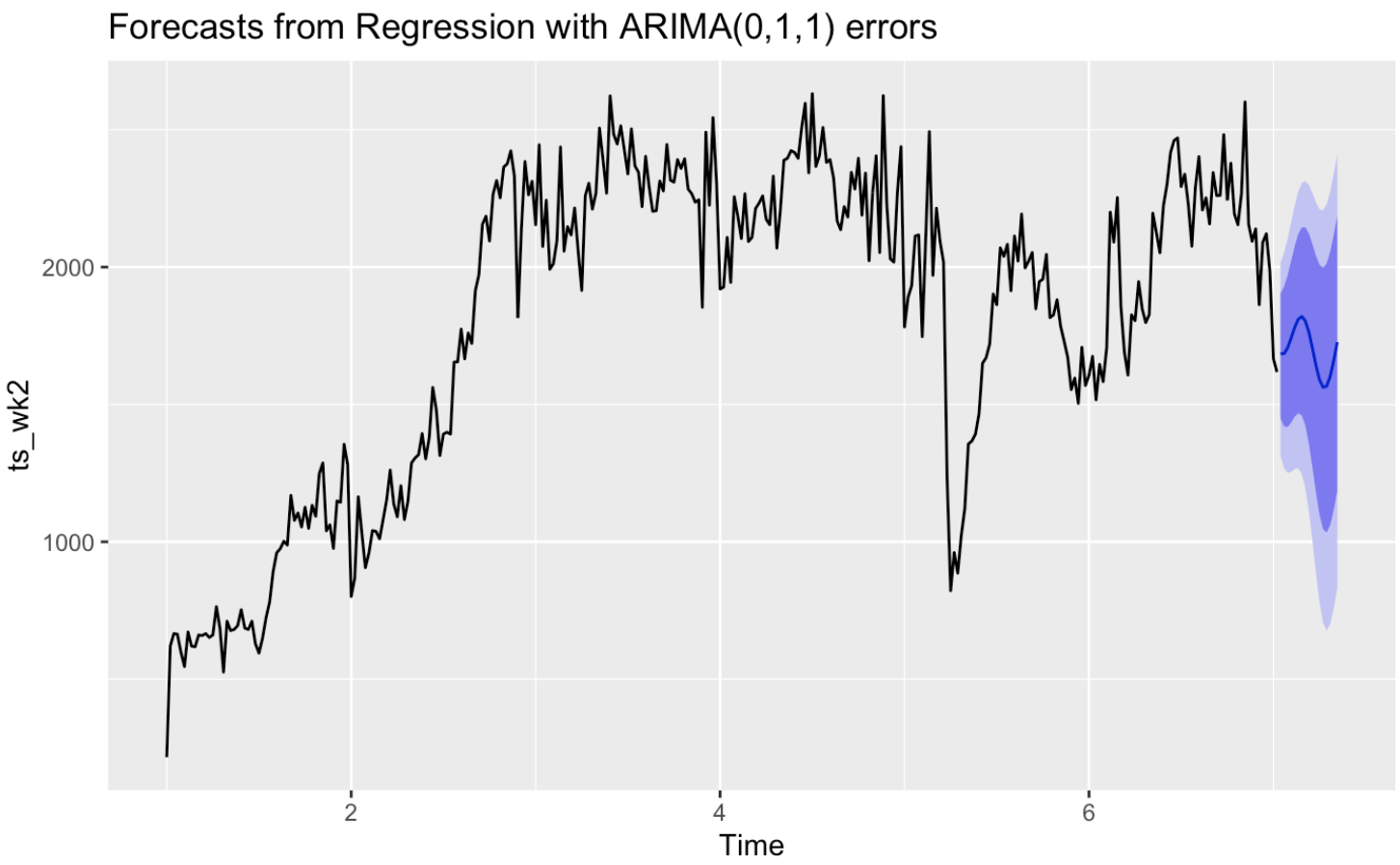


Hide

```
fc_wk2_fourier <- forecast(fit_wk2_fourier, xreg = fourier(ts_wk2,K = 5,h=17))
```

Hide

```
autoplot(fc_wk2_fourier)
```



Hide

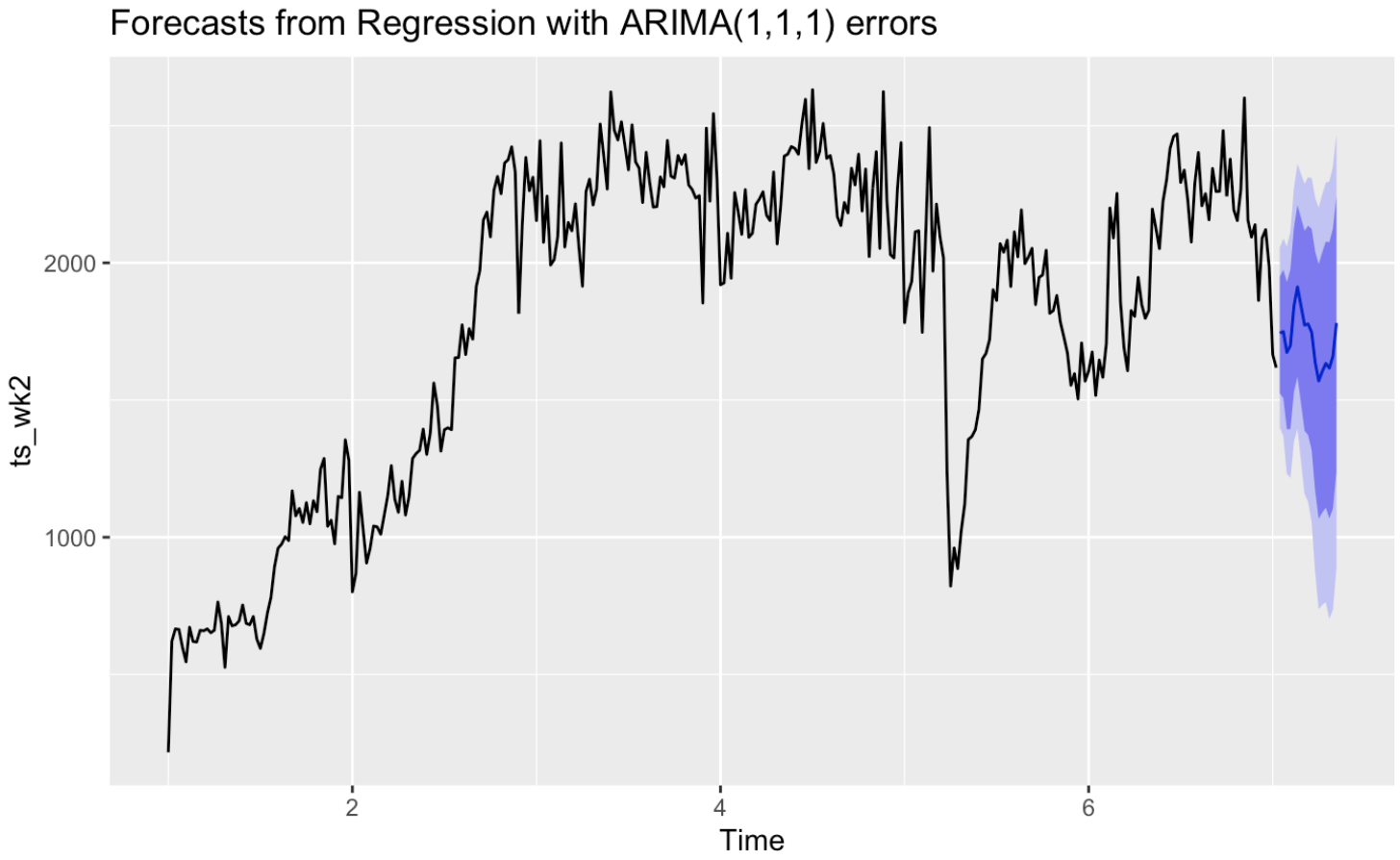
```
bestfit <- list(aicc=Inf)
k = 0
for(i in 1:25)
{
  fit_fourier_best <-auto.arima(ts_wk2, xreg=fourier(ts_wk2, i), seasonal = FALSE, la
mbda = 1.555455)

  if(fit_fourier_best$aicc < bestfit$aicc){
    bestfit <- fit_fourier_best
    k = i
  }
}
```

Hide

```
fc_wk2_fourier_best <- forecast(bestfit, xreg=fourier(ts_wk2, K=13, h=17))
```

Hide

```
autoplot(fc_wk2_fourier_best)
```



Hide

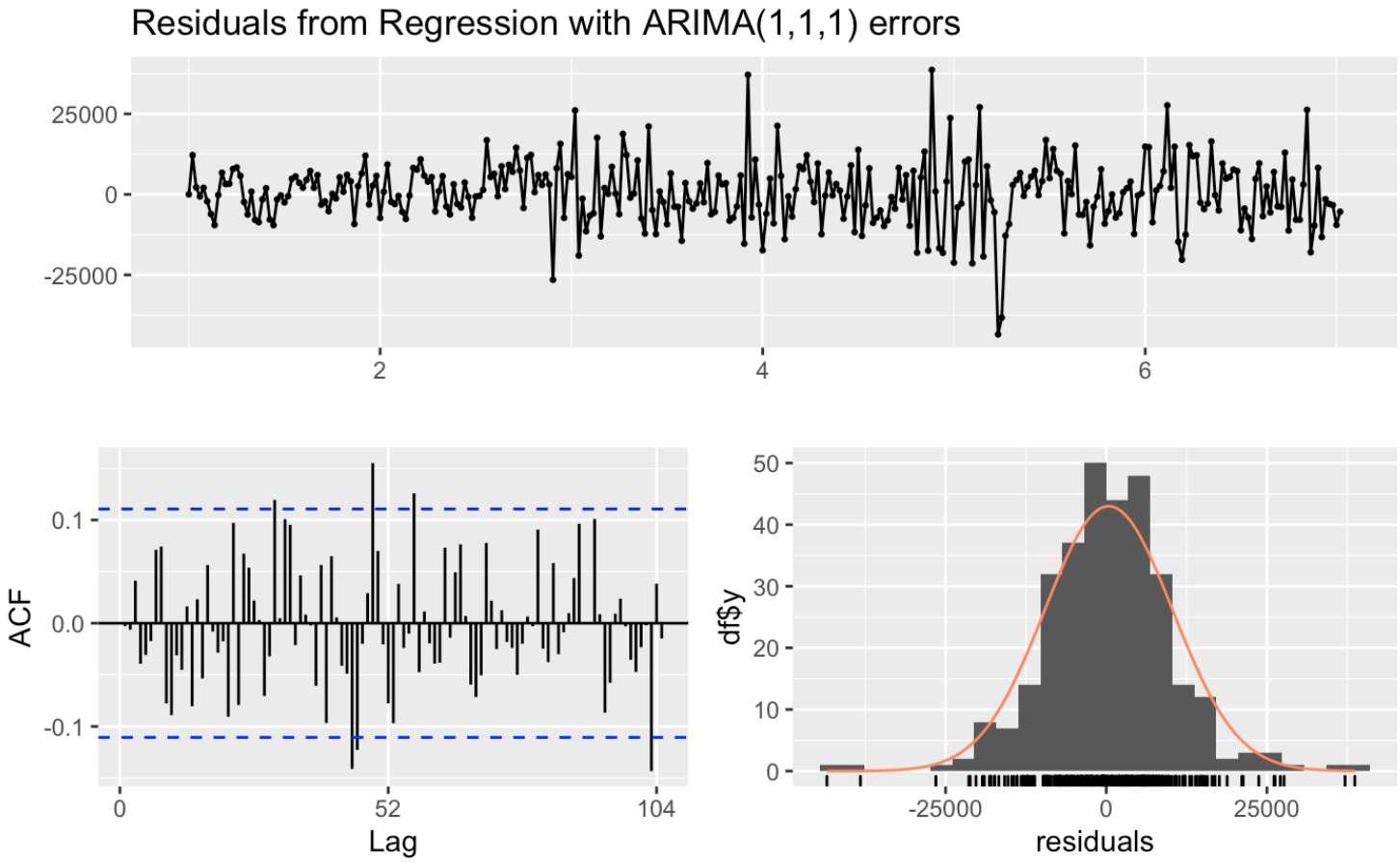
```
bestfit
```

```
Series: ts_wk2
Regression with ARIMA(1,1,1) errors
Box Cox transformation: lambda= 1.555455

Coefficients:
          ar1      ma1      S1-52      C1-52     S2-52      C2-52      S3-52      C3-
52      S4-52      C4-52
      -0.1680  -0.3727  -9948.558  -1981.271  -133.413   716.6996   934.6597  -4236.8
77  -2421.367  -176.1832
s.e.   0.1075   0.1006   3586.074   3580.949  1836.836  1834.8779  1272.3847   1271.6
72   1003.376  1003.2829
           S5-52      C5-52      S6-52     C6-52     S7-52      C7-52       S8-52
C8-52       S9-52     C9-52
      -1037.0726  2042.7483  -119.8836  1302.395  642.4525  1142.2846  -1234.1802  17
43.877  -1063.8719   25.3034
s.e.    851.8568   852.0661   758.5380   758.831  698.0701   698.2922    657.8543   6
57.909    630.9506  630.8152
         S10-52     C10-52      S11-52    C11-52      S12-52     C12-52     S13-52
C13-52
      -1734.638  -109.8238  -1652.6035  -35.5724  -1672.6484  -917.4102  -1167.566  1
741.9501
s.e.    613.237   612.9466    602.1258  601.7811    595.9155   595.6520    593.417
593.4143

sigma^2 = 109337835:  log likelihood = -3326.42
AIC=6710.84   AICc=6716.99   BIC=6819.48
```

Hide

```
checkresiduals(bestfit)
```

```


    Ljung-Box test

data:  Residuals from Regression with ARIMA(1,1,1) errors
Q* = 91.313, df = 35, p-value = 6.426e-07

Model df: 28.   Total lags used: 63
```



Hide

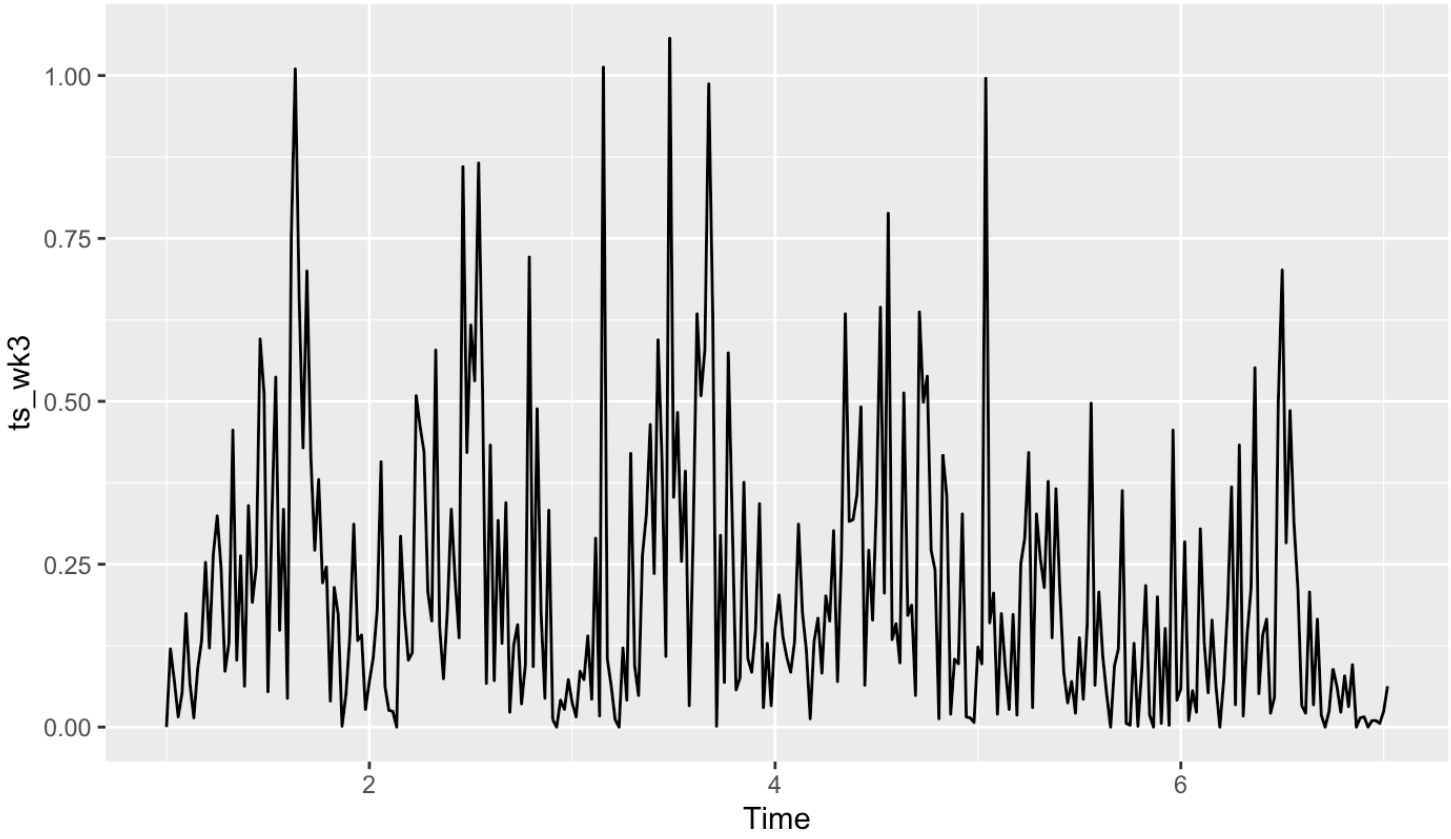
```
err_fourier_best <- sqrt(mean((fc_wk2_fourier_best$mean - df_wk_gp_fc$crashes)^2))
err_fourier_best
```

```
[1] 250.393
```

#Running ARIMA on two variables for car crashes (Percipitation)

Hide

```
autoplot(ts_wk3)
```



Hide

```
kpss.test(ts_wk3)
```

```
    KPSS Test for Level Stationarity

data:  ts_wk3
KPSS Level = 0.64392, Truncation lag parameter = 5, p-value = 0.01864
```

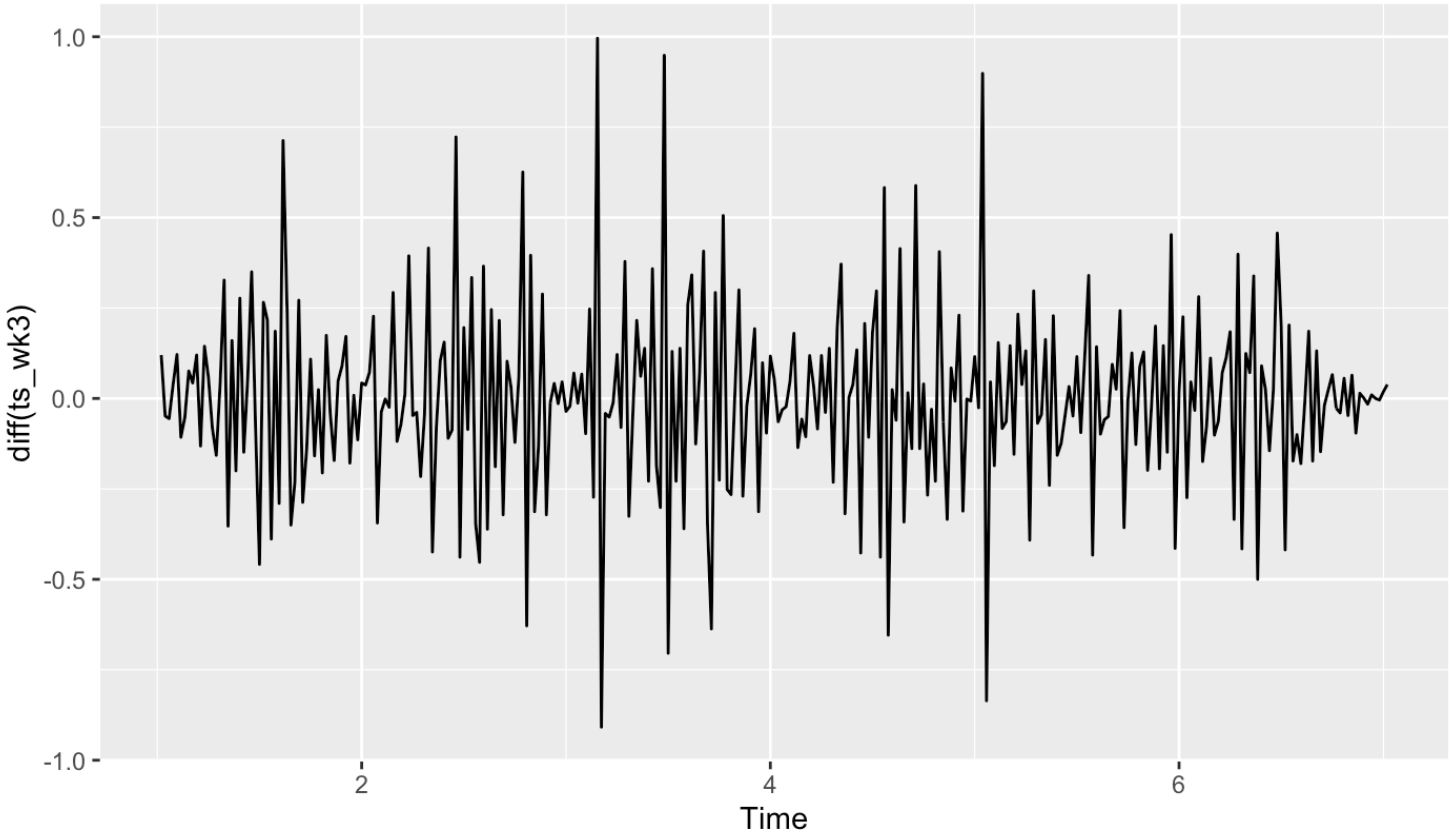
Hide

```
kpss.test(ts_wk4)
```

```
    KPSS Test for Level Stationarity

data:  ts_wk4
KPSS Level = 0.72275, Truncation lag parameter = 5, p-value = 0.01148
```

Hide

```
autoplot(diff(ts_wk3))
```



Hide

```
kpss.test(diff(ts_wk3))
```

```
Warning in kpss.test(diff(ts_wk3)) :
  p-value greater than printed p-value
```

```

    KPSS Test for Level Stationarity

data:  diff(ts_wk3)
KPSS Level = 0.018856, Truncation lag parameter = 5, p-value = 0.1
```

Hide

```
length(diff(ts_wk3))
```

```
[1] 313
```

Hide

```
fit_wk_auto_percip <- auto.arima(ts_wk2, xreg = ts_wk3, lambda = 1.368373, seasonal=TRUE)
```

Hide

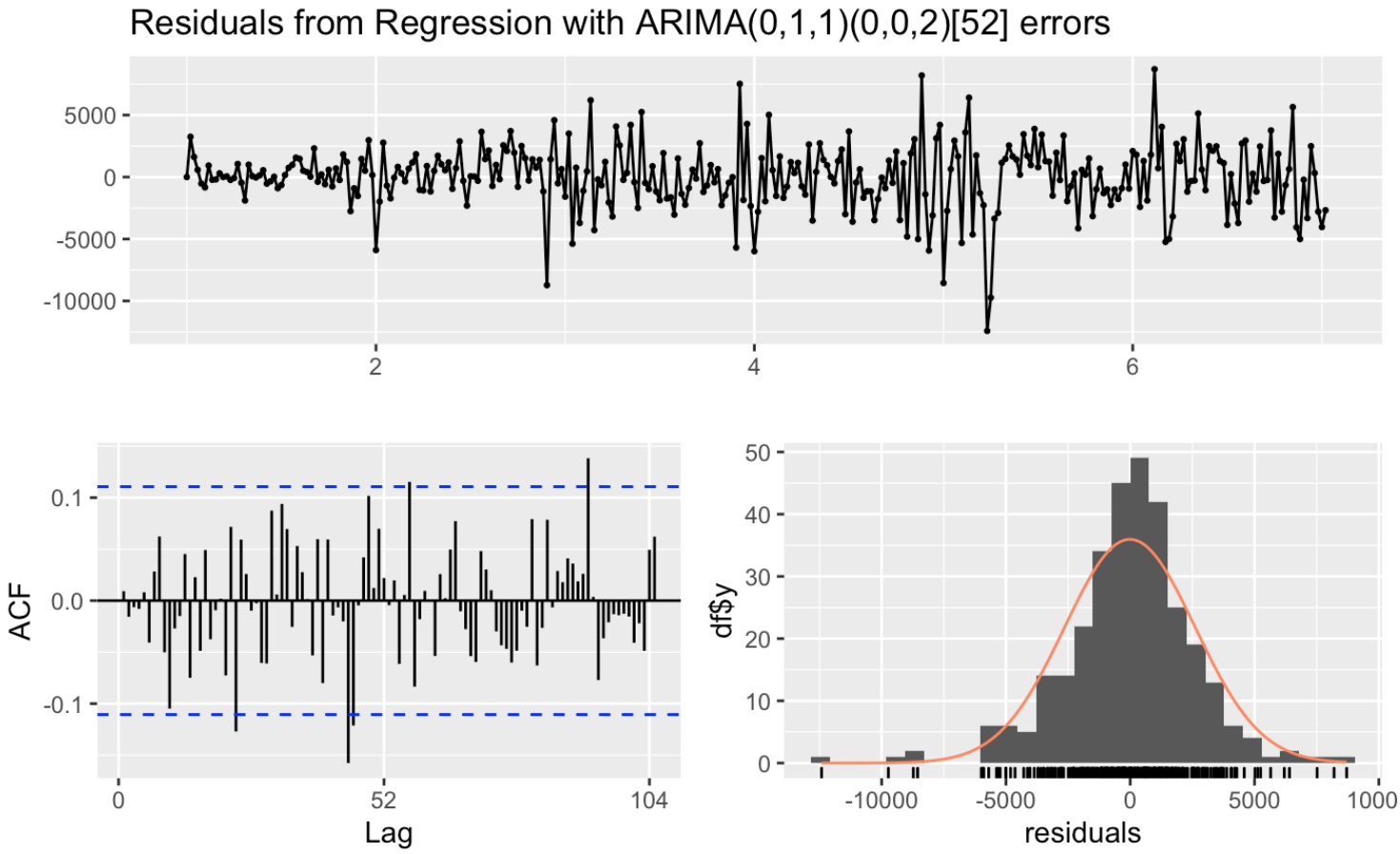
```
fit_wk_auto_percip
```

```
Series: ts_wk2
Regression with ARIMA(0,1,1)(0,0,2)[52] errors
Box Cox transformation: lambda= 1.368373

Coefficients:
          ma1    sma1    sma2     drift      xreg
      -0.4034  0.0780  0.2355   64.2193  291.1018
s.e.   0.0536  0.0582  0.0696  109.3153  687.1749

sigma^2 = 7021318:  log likelihood = -2911.9
AIC=5835.81   AICc=5836.08   BIC=5858.28
```

Hide

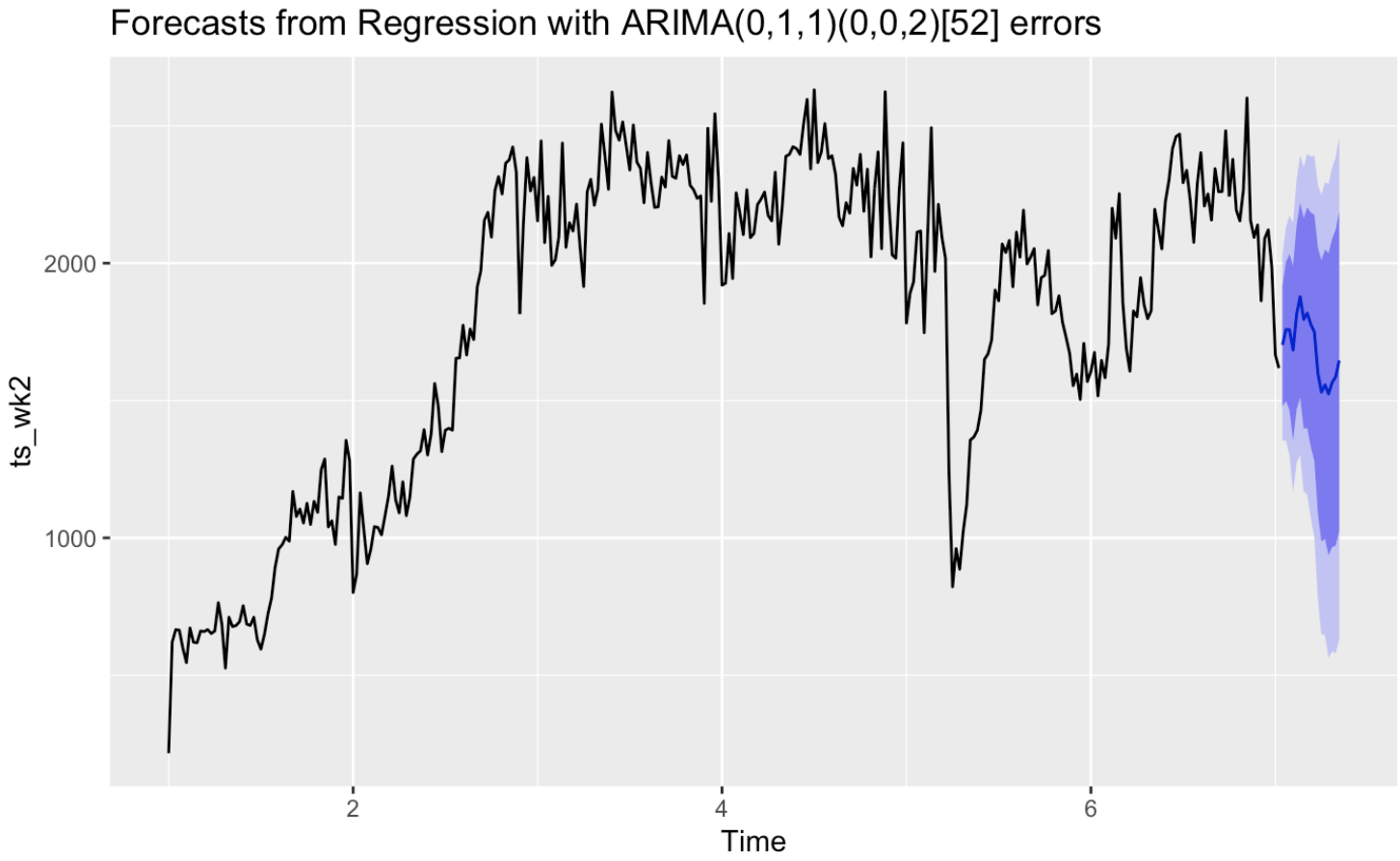
```
checkresiduals(fit_wk_auto_percip)
```

```

    Ljung-Box test

data:  Residuals from Regression with ARIMA(0,1,1)(0,0,2)[52] errors
Q* = 72.175, df = 58, p-value = 0.09979

Model df: 5.   Total lags used: 63
```



Hide

```
fc_wk_auto_percip <- forecast(fit_wk_auto_percip, xreg = df2_wk_gp_fc$totalprecipIn , h=17)
autoplot(fc_wk_auto_percip)
```



If differencing is specified, then the differencing is applied to all variables in the regression model before the model is estimated.If differencing is required, then all variables are differenced during the estimation process, although the final model will be expressed in terms of the original variables.

#Running ARIMA MAX on two variables for car crashes (Visibility)

Hide

```
fit_wk_auto_visibility <- auto.arima(ts_wk2, xreg = ts_wk4, lambda = 1.368373, season
al=TRUE)
```

Hide

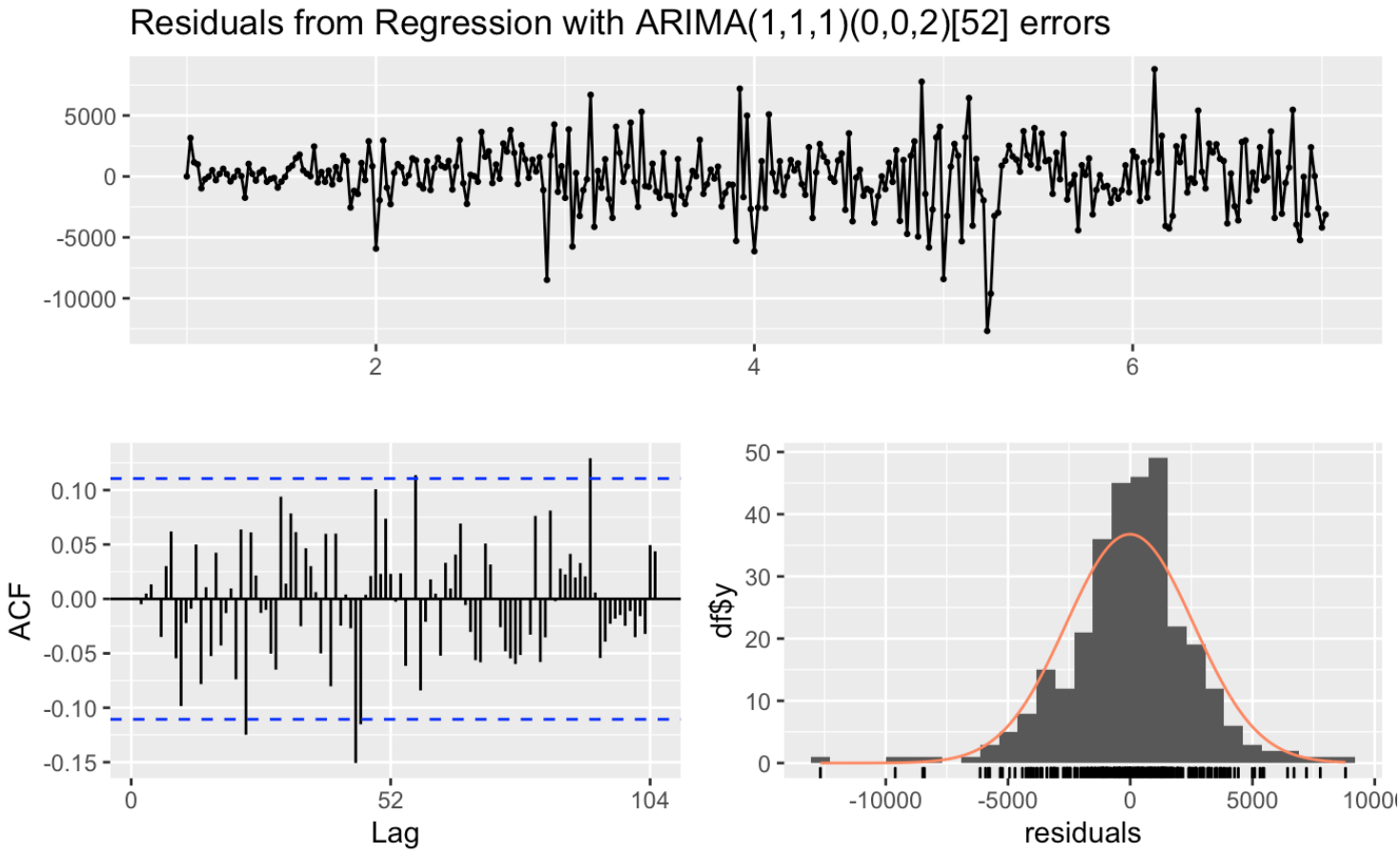
```
fit_wk_auto_visibility
```

```
Series: ts_wk2
Regression with ARIMA(1,1,1)(0,0,2)[52] errors
Box Cox transformation: lambda= 1.368373

Coefficients:
         ar1      ma1    sma1    sma2     drift       xreg
      0.0589  -0.4496  0.0781  0.2348   64.5611  -511.5991
s.e.  0.1393   0.1251  0.0579  0.0691  106.5097   259.8333

sigma^2 = 6960542:  log likelihood = -2910.02
AIC=5834.03   AICc=5834.4   BIC=5860.26
```

Hide

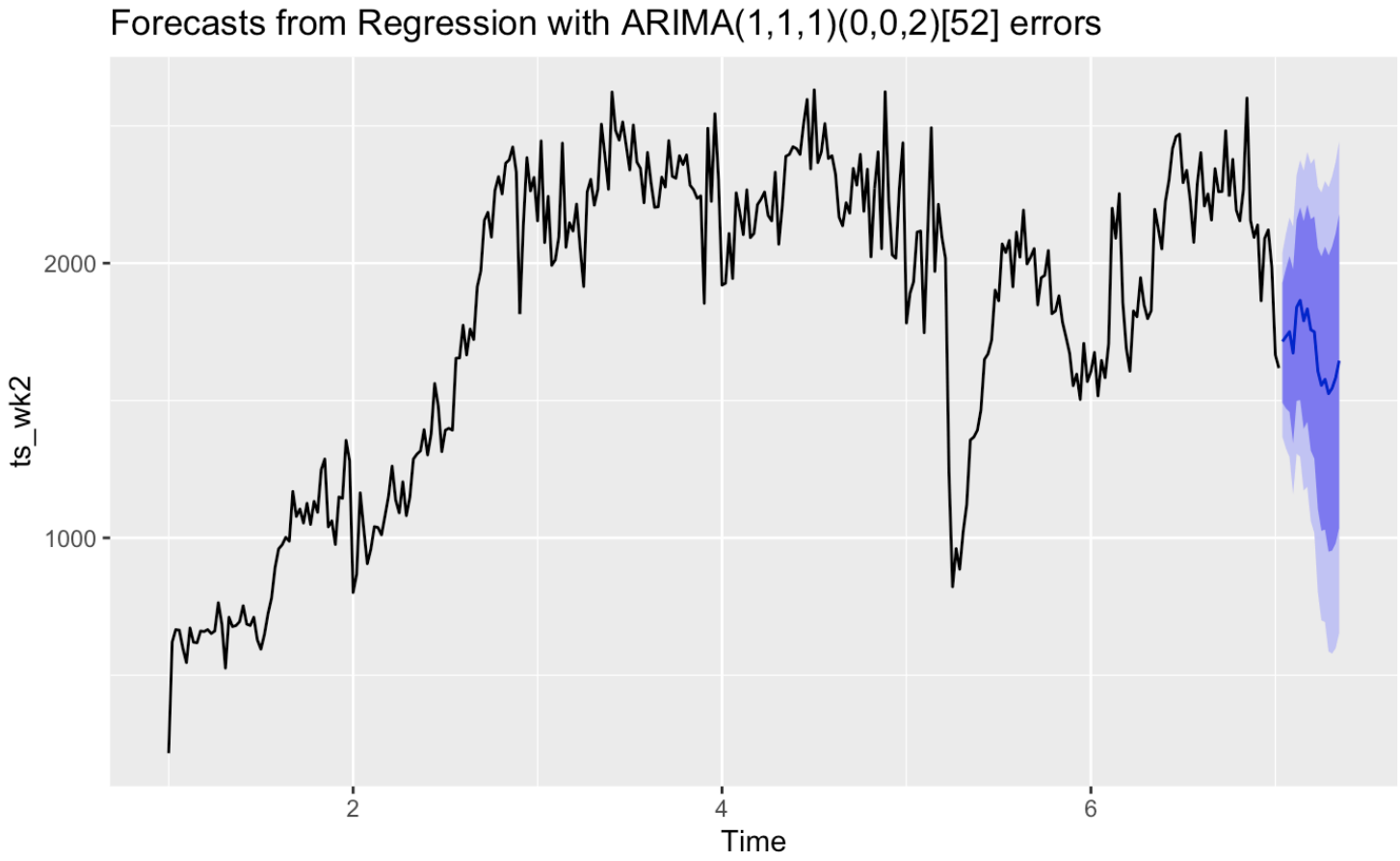
```
checkresiduals(fit_wk_auto_visibility)
```

```


    Ljung-Box test

data:  Residuals from Regression with ARIMA(1,1,1)(0,0,2)[52] errors
Q* = 69.157, df = 57, p-value = 0.1297

Model df: 6.   Total lags used: 63
```



Hide

```
fc_wk_auto_visibility <- forecast(fit_wk_auto_visibility, xreg = df2_wk_gp_fc$visibil
ityMiles ,h=17)
autoplot(fc_wk_auto_visibility)
```



#Running ARIMA MAX on two variables for car crashes (totalprecipIn & visibility)

Hide

```
xreg = cbind(Percipitation = ts_wk3, Visibility = ts_wk4)
fit_wk_auto_both <- auto.arima(ts_wk2, xreg = xreg, lambda = 1.368373, seasonal=TRUE)
```

Hide

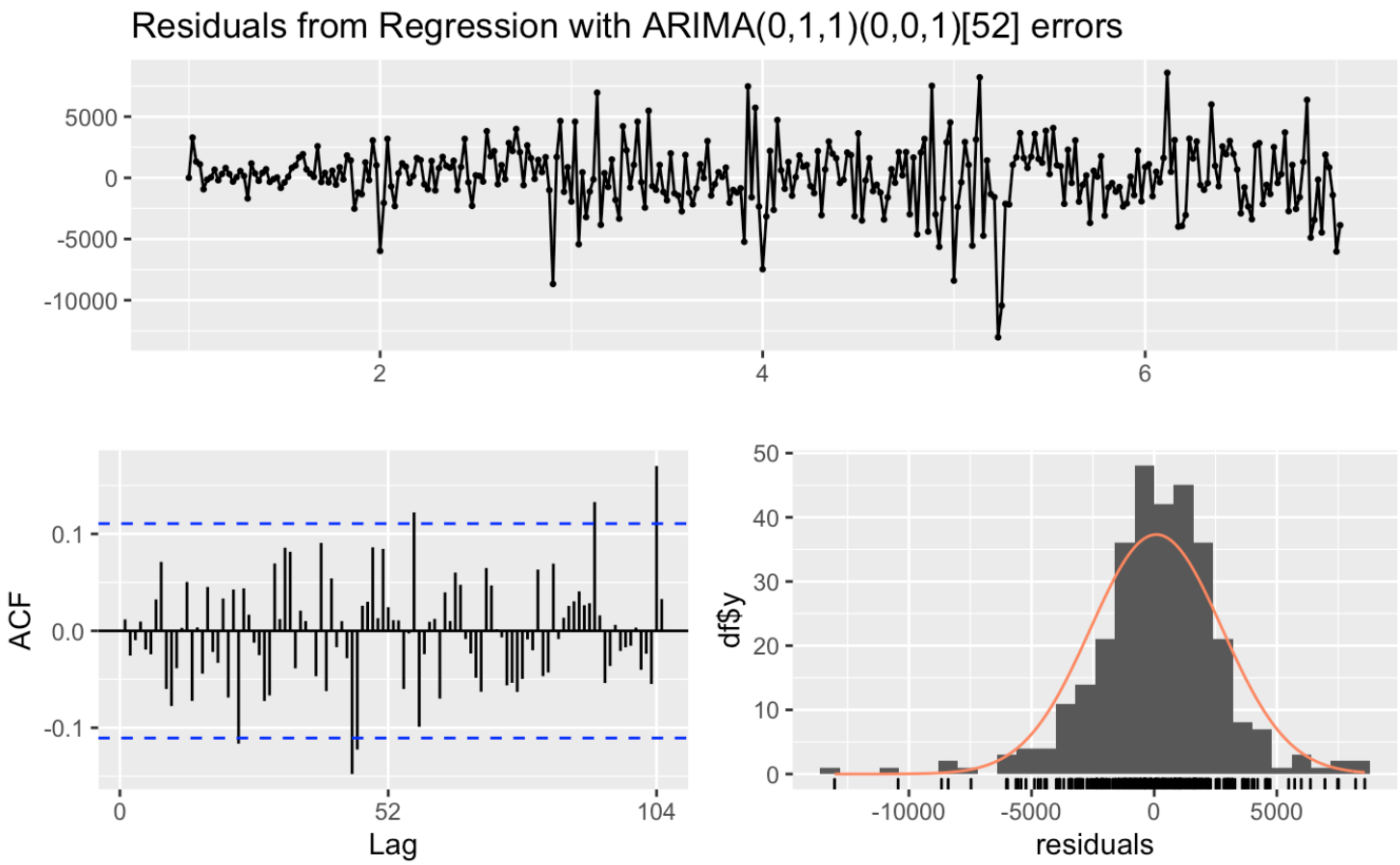
```
fit_wk_auto_both
```

```
Series: ts_wk2
Regression with ARIMA(0,1,1)(0,0,1)[52] errors
Box Cox transformation: lambda= 1.368373

Coefficients:
          ma1    sma1  Percipitation  Visibility
      -0.4069  0.1007      -134.4997   -534.2200
s.e.   0.0540  0.0512       774.4471    287.7701

sigma^2 = 7326247:  log likelihood = -2916.26
AIC=5842.52   AICc=5842.72   BIC=5861.25
```

Hide

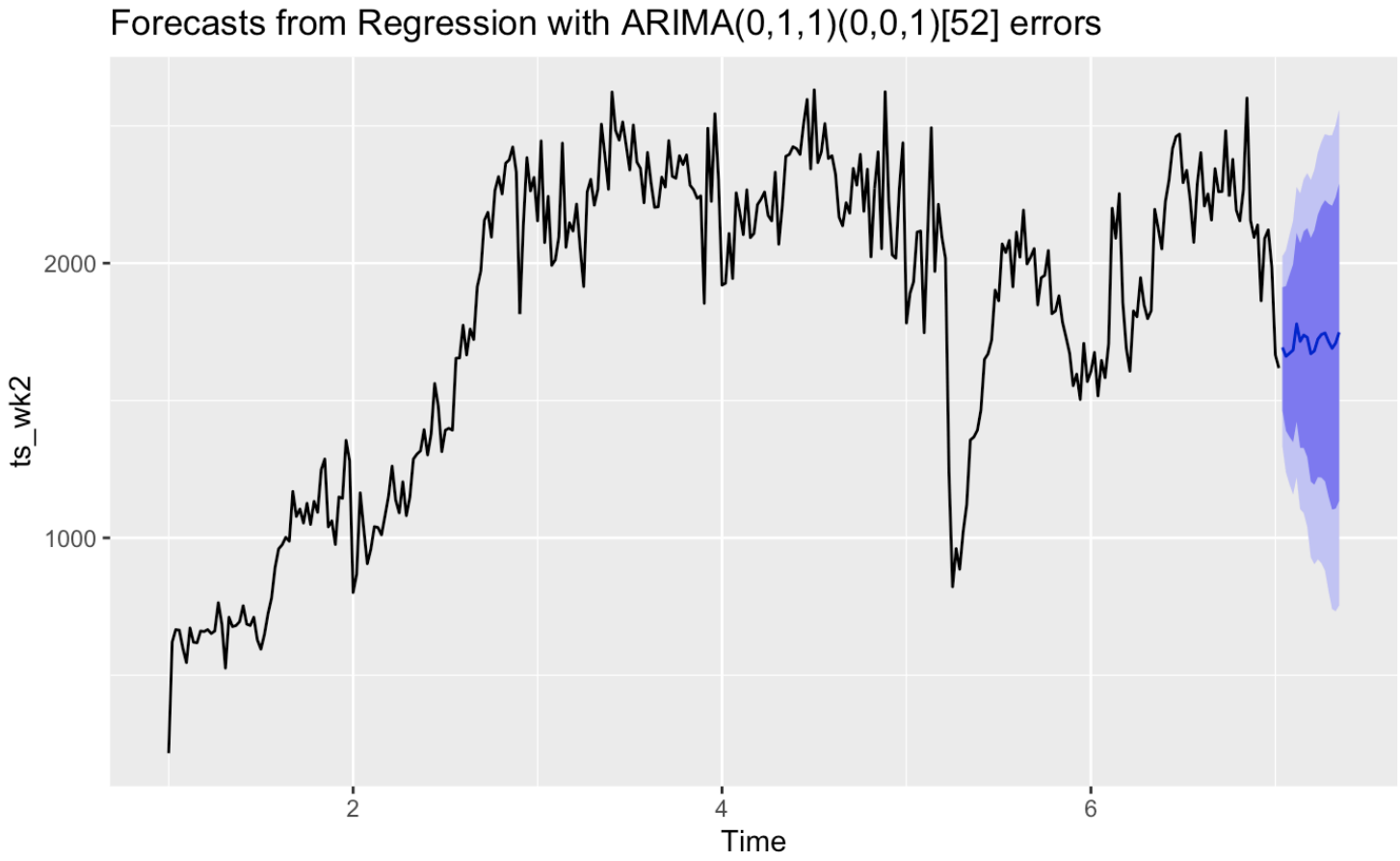
```
checkresiduals(fit_wk_auto_both)
```

```
    Ljung-Box test

data:  Residuals from Regression with ARIMA(0,1,1)(0,0,1)[52] errors
Q* = 70.201, df = 59, p-value = 0.1508

Model df: 4.   Total lags used: 63
```



Hide

```
xreg_fc = cbind(Percipitation = df2_wk_gp_fc$totalprecipIn, Visibility = df2_wk_gp_fc
$visibilityMiles)
fc_wk_auto_both <- forecast(fit_wk_auto_both, xreg = xreg_fc ,h=17)
autoplot(fc_wk_auto_both)
```



Evaluation of ARIMAX model

Hide

```
fc_wk_auto_both
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
7.038462	1693.249	1463.022	1912.444	1335.6429	2024.768
7.057692	1660.946	1389.866	1916.552	1238.2892	2046.830
7.076923	1672.070	1367.306	1957.504	1195.5515	2102.466
7.096154	1683.626	1348.550	1995.598	1158.3421	2153.558
7.115385	1778.762	1423.351	2109.574	1221.5302	2277.049
7.134615	1715.551	1327.800	2073.139	1104.9545	2253.352
7.153846	1738.245	1327.602	2115.581	1090.4344	2305.422
7.173077	1729.565	1294.056	2127.557	1040.5578	2327.285

7.192308	1670.505	1204.757	2092.094	929.7087	2302.774
7.211538	1680.634	1193.781	2119.635	904.4851	2338.658

1-10 of 17 rows Previous **1** 2 Next

RMSE

Hide

```
err_arimax_both <- sqrt(mean((fc_wk_auto_both$mean - df_wk_gp_fc$crashes)^2))
err_arimax_both
```

```
[1] 236.3234
```

#Cross Validation This method has the benefit of providing a much more robust estimation of how the chosen modeling method and parameters will perform in practice.

#Intervention Analysis It looks like that starting from the week of 2020-03-16, the weekly crash number has a significant drop. We assumed that this drop is very likely caused by the external event - COVID-19. Due to the quarantine, there were less traffic on the road and therefore, led to less car accidents.

It appears that there is a drastic shift in the series, that slowly increased and eventually returns to previous levels. This may indicate an intervention model with a pulse function which is typically employed if the effects are expected to be only temporary, and decay over time.

Auto Arima: ARIMA(0,1,1)(0,0,1)[52]

Analysis with Pulse function

Hide

```
covid <- 1*(ts_wk2 == 1232)
```

Hide

```
fit_intv_model <- arimax(log(ts_wk2), order=c(0,1,1), seasonal=list(order=c(0,0,1),pe
riod=52), xreg=xreg, xtransf=covid, transfer=list(c(1,0)), method = 'ML')
```

Hide

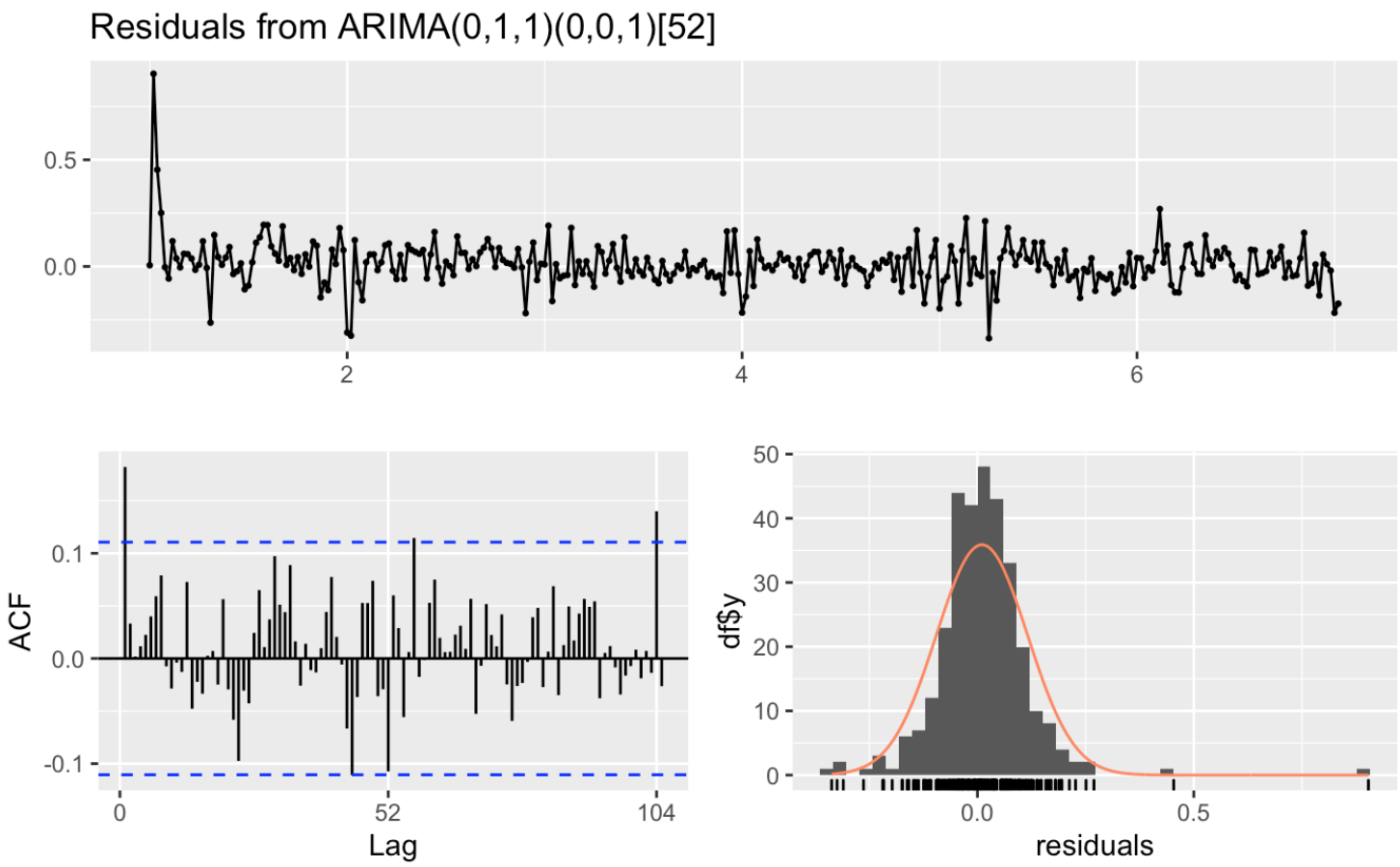
```
fit_intv_model
```

```
Call:
arimax(x = log(ts_wk2), order = c(0, 1, 1), seasonal = list(order = c(0, 0,
    1), period = 52), xreg = xreg, method = "ML", xtransf = covid, transfer = list(c(
1,
    0)))

Coefficients:
          ma1    sma1  Percipitation  Visibility  T1-AR1   T1-MA0
      -0.5170  0.2495         0.0061     -0.0245  0.9397  -0.7534
s.e.   0.0638  0.0667         0.0308      0.0115  0.0293   0.0933

sigma^2 estimated as 0.01126:  log likelihood = 256.15,  aic = -500.3
```

Hide

```
checkresiduals(fit_intv_model)
```

```
    Ljung-Box test

data:  Residuals from ARIMA(0,1,1)(0,0,1)[52]
Q* = 66.045, df = 57, p-value = 0.1928

Model df: 6.   Total lags used: 63
```



Hide

```
fit_intv_model$coef
```

```
        ma1         sma1 Percipitation   Visibility       T1-AR1       T1-MA0
-0.517029810  0.249475933  0.006148913 -0.024489411  0.939704695 -0.753388853
```

Hide

```
covidx <- c(covid, rep(0,17))
```

Hide

```
intv_pulse <- fit_intv_model$coef["T1-MA0"]+stats::filter(covidx, filter=fit_intv_mod
el$coef["T1-AR1"], method="recursive", side=1)
```

Hide

```
intv_pulse_x <- Arima(log(ts_wk2), order=c(0,1,1), seasonal=list(order=c(0,0,1),perio
d=52), xreg=intv_pulse[1:length(ts_wk2)])
```

Hide

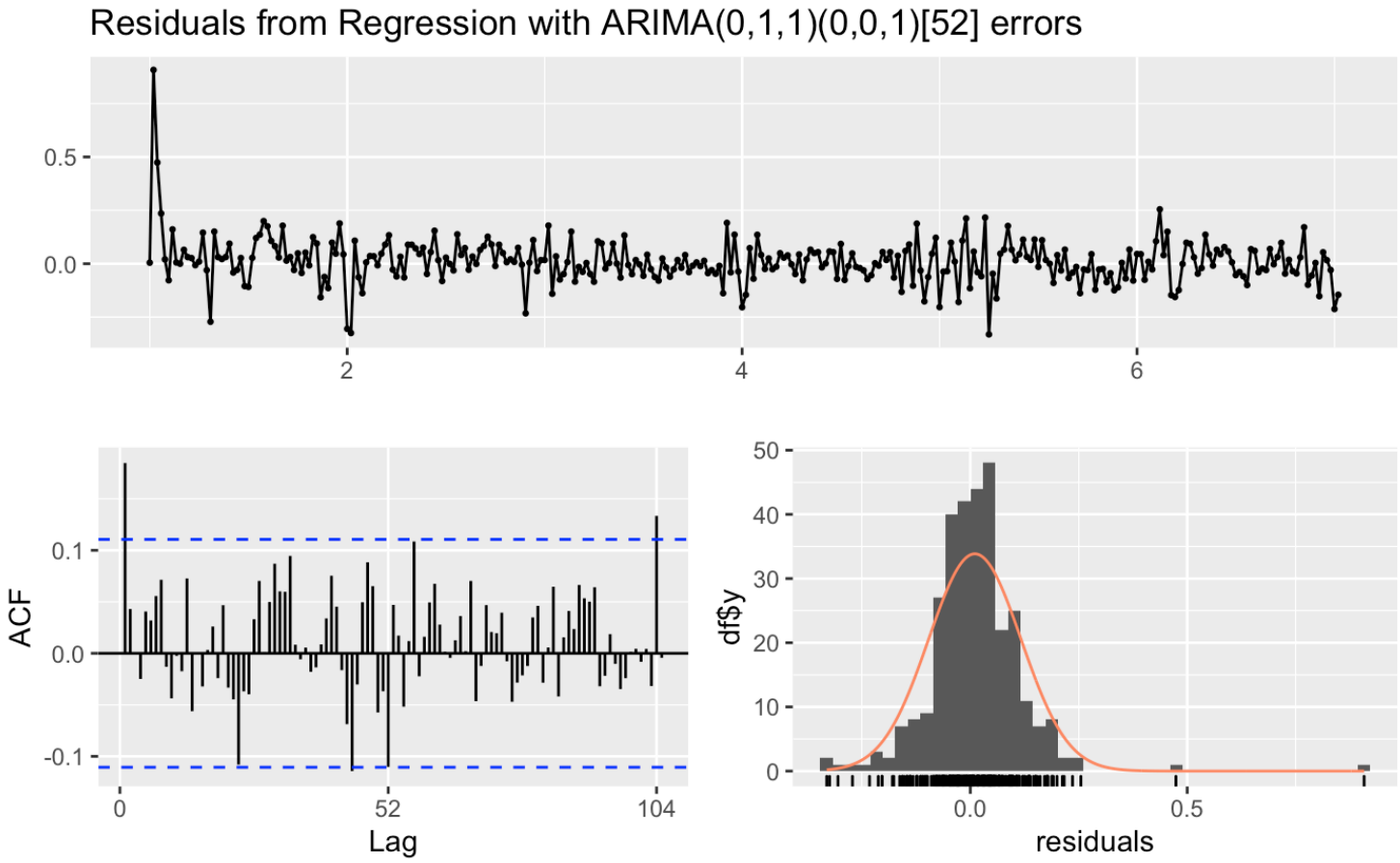
```
intv_pulse_x
```

```
Series: log(ts_wk2)
Regression with ARIMA(0,1,1)(0,0,1)[52] errors

Coefficients:
          ma1    sma1     xreg
      -0.5214  0.2532  -0.7467
s.e.   0.0639  0.0667   0.0935

sigma^2 = 0.01159:  log likelihood = 253.17
AIC=-498.34   AICc=-498.21   BIC=-483.35
```

Hide

```
checkresiduals(intv_pulse_x)
```

```


    Ljung-Box test

data:  Residuals from Regression with ARIMA(0,1,1)(0,0,1)[52] errors
Q* = 69.706, df = 60, p-value = 0.1834

Model df: 3.   Total lags used: 63
```



Hide

```
intv_pulse_fc <- forecast(intv_pulse_x, h=17, xreg=intv_pulse[(length(ts_wk2)+1):(len
gth(ts_wk2)+17)] )
```

Hide

```
intv_pulse_fc
```

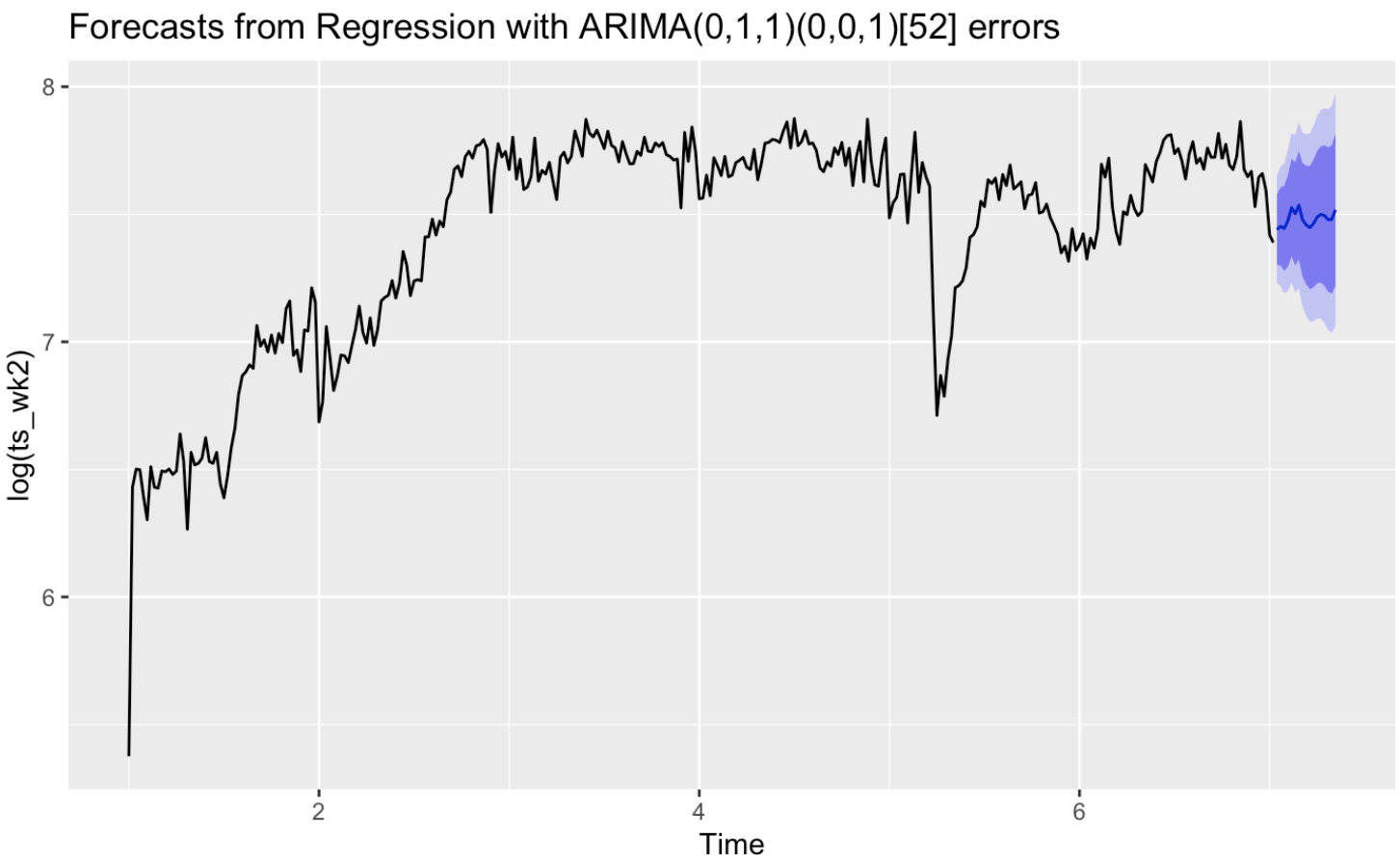
	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
7.038462	7.439904	7.301960	7.577849	7.228936	7.650872
7.057692	7.452362	7.299436	7.605289	7.218481	7.686244
7.076923	7.444491	7.277924	7.611058	7.189749	7.699233
7.096154	7.474796	7.295624	7.653967	7.200777	7.748815
7.115385	7.525507	7.334561	7.716453	7.233481	7.817534
7.134615	7.502099	7.300064	7.704134	7.193113	7.811085

7.153846	7.534938	7.322392	7.747485	7.209876	7.860000
7.173077	7.478022	7.255459	7.700584	7.137642	7.818401
7.192308	7.458012	7.225866	7.690159	7.102975	7.813049
7.211538	7.447433	7.206083	7.688783	7.078320	7.816546

1-10 of 17 rows Previous **1** 2 Next

Hide

```
autoplot(intv_pulse_fc)
```



Hide

```
err_intv_pulse <- sqrt(mean((exp(intv_pulse_fc$mean) - df_wk_gp_fc$crashes)^2))
err_intv_pulse
```

```
[1] 179.2215
```

Analysis with Step function

Hide

```
covid_step <- 1*(seq(ts_wk2) >= 221)
```

Hide

```
fit_intv_step <- arimax(log(ts_wk2), order=c(0,1,1), seasonal=list(order=c(0,0,1),per
iod=52), xreg=xreg, xtransf=covid_step, transfer=list(c(1,0)), method = 'ML')
```

Hide

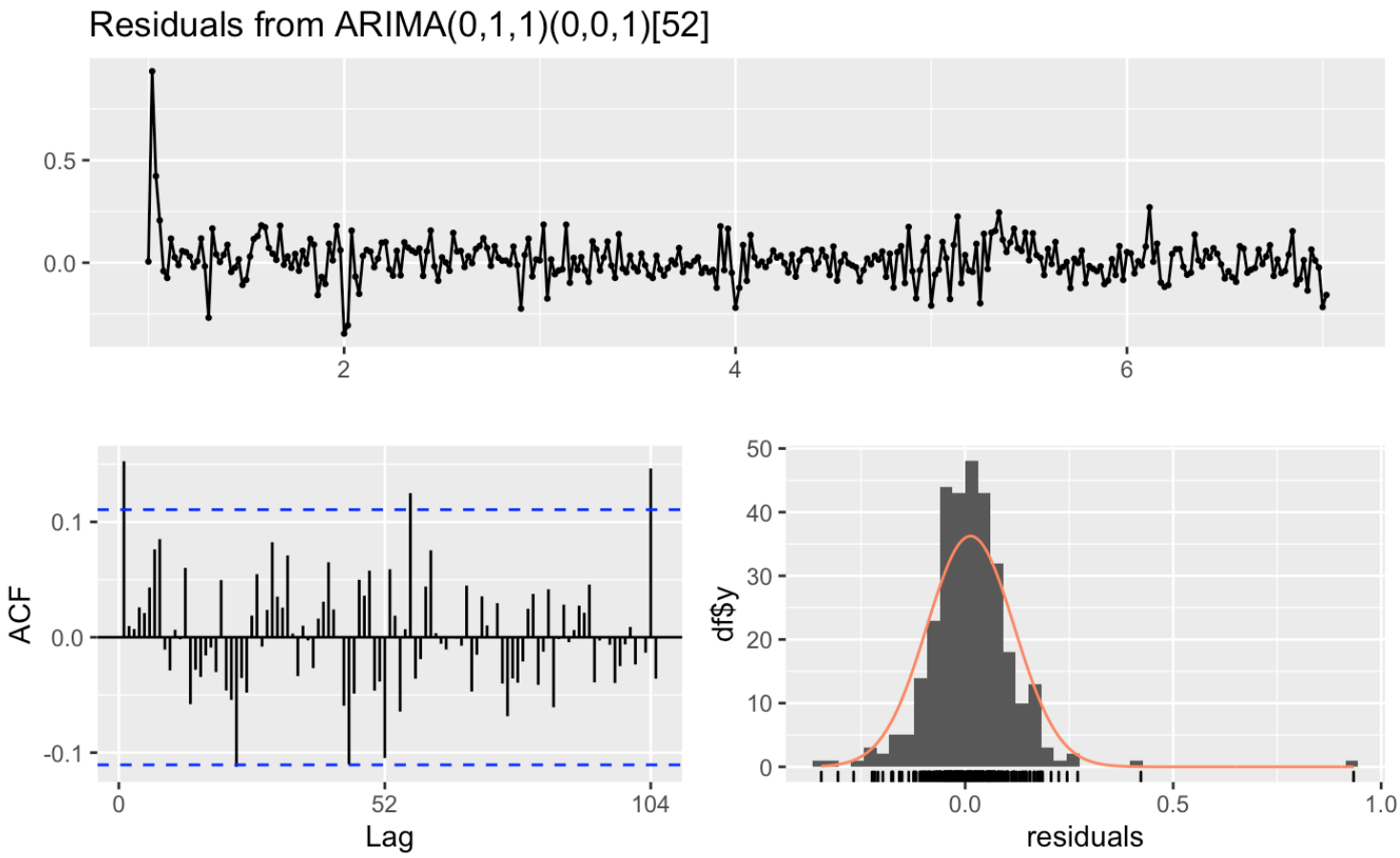
```
fit_intv_step
```

```

Call:
arimax(x = log(ts_wk2), order = c(0, 1, 1), seasonal = list(order = c(0, 0,
    1), period = 52), xreg = xreg, method = "ML", xtransf = covid_step, transfer = li
st(c(1,
    0)))

Coefficients:
          ma1    sma1  Percipitation  Visibility  T1-AR1   T1-MA0
      -0.4415  0.2360         0.0046     -0.0236  0.2591  -0.6278
s.e.   0.0636  0.0655         0.0303      0.0113  0.0979   0.0914

sigma^2 estimated as 0.01126:  log likelihood = 256.45,  aic = -500.9
```

Hide

```
checkresiduals(fit_intv_step)
```

```

    Ljung-Box test

data:  Residuals from ARIMA(0,1,1)(0,0,1)[52]
Q* = 61.051, df = 57, p-value = 0.3325

Model df: 6.   Total lags used: 63
```



Hide

```
covidx_step <- c(covid_step,  rep(1,18))
```

Hide

```
intv_step <- fit_intv_step$coef["T1-MA0"]+stats::filter(covidx_step, filter=fit_intv_
step$coef["T1-AR1"], method="recursive", side=1)
```

Hide

```
intv_step_x <- Arima(log(ts_wk2), order=c(0,1,1), seasonal=list(order=c(0,0,1),period
=52), xreg=intv_step[1:length(ts_wk2)])
```

Hide

```
intv_step_x
```

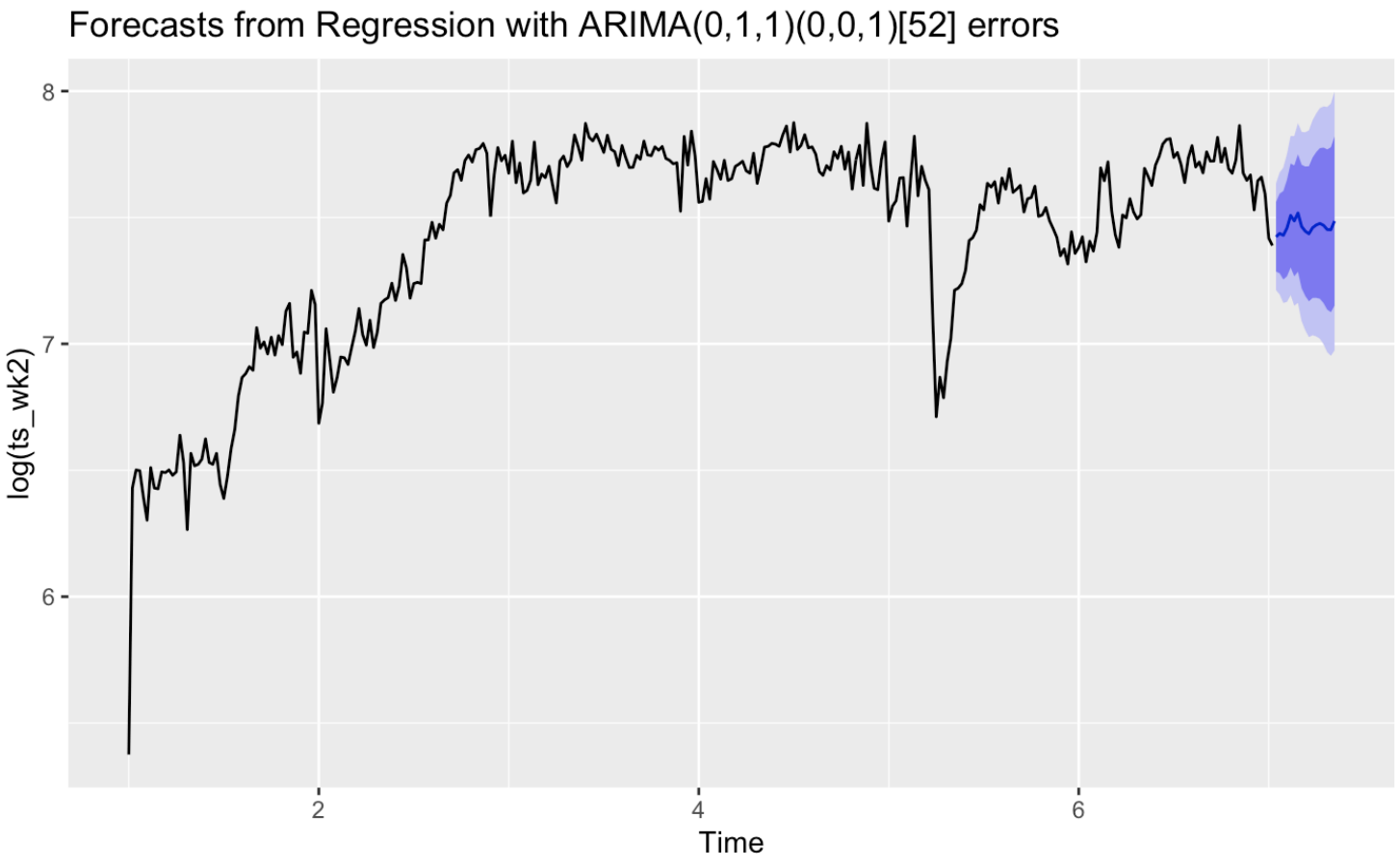
```
Series: log(ts_wk2)
Regression with ARIMA(0,1,1)(0,0,1)[52] errors

Coefficients:
          ma1    sma1     xreg
      -0.4468  0.2405  -0.6225
s.e.   0.0627  0.0653   0.0802

sigma^2 = 0.01157:  log likelihood = 253.63
AIC=-499.26   AICc=-499.13   BIC=-484.27
```

Hide

```
intv_step_fc <- forecast(intv_step_x, h=17, xreg=intv_step[(length(ts_wk2)+1):(length
(ts_wk2)+17)] )
```

Hide

```
autoplot(intv_step_fc)
```



Hide

```
exp(intv_step_fc$mean)
```

```
Time Series:
Start = c(7, 3)
End = c(7, 19)
Frequency = 52
 [1] 1675.332 1697.196 1685.166 1734.467 1822.278 1783.975 1840.151 1744.727 1712.171
1695.541 1736.030 1755.169
[13] 1765.959 1752.744 1724.059 1722.883 1782.578
```

Hide

```
err_intv_step <- sqrt(mean((exp(intv_step_fc$mean) - df_wk_gp_fc$crashes)^2))
err_intv_step
```

```
[1] 205.1181
```