Time Serie Exam S20

Mamadi.Fofana

11/1/2020

Loading library

```r
library(fpp2)
```

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

```
## -- Attaching packages --------------------------------------------- fpp2 2.4 --
```

```
## v ggplot2   3.3.2     v fma       2.4
## v forecast  8.13      v expsmooth 2.3
```

```
##
```

```r
library(forecast)
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(xts)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'xts'
```

```
## The following objects are masked from 'package:dplyr':
##
##     first, last
```

Loading data set

```r
my_data = read.csv("/cloud/project/Elec-train.csv")
```

Summary of data

```r
str(my_data)
```

```
## 'data.frame':    4603 obs. of  3 variables:
##  $ Timestamp : chr  "1/1/2010 1:15" "1/1/2010 1:30" "1/1/2010 1:45" "1/1/2010 2:00" ...
##  $ Power..kW.: num  165 152 147 154 154 ...
##  $ Temp..C.. : num  10.6 10.6 10.6 10.6 10.6 10.6 10.6 10.6 10 10 ...
```

```r
summary(my_data)
```

```
##   Timestamp           Power..kW.      Temp..C..
##  Length:4603        Min.   :134.1   Min.   : 3.90
##  Class :character   1st Qu.:163.3   1st Qu.: 8.90
##  Mode  :character   Median :253.7   Median :11.10
##                     Mean   :231.6   Mean   :10.89
##                     3rd Qu.:277.5   3rd Qu.:12.80
##                     Max.   :355.1   Max.   :19.40
##                     NA's   :96
```

```r
head(my_data)
```

```
##       Timestamp Power..kW. Temp..C..
## 1 1/1/2010 1:15      165.1      10.6
## 2 1/1/2010 1:30      151.6      10.6
## 3 1/1/2010 1:45      146.9      10.6
## 4 1/1/2010 2:00      153.7      10.6
## 5 1/1/2010 2:15      153.8      10.6
## 6 1/1/2010 2:30      159.0      10.6
```

```r
dim(my_data)
```

```
## [1] 4603    3
```

Data wrangling

Conversion field timeStamp into date format

```r
start.date <- strptime("2010-1-1 1:15", "%Y-%m-%d %H:%M")
start.date <- format(start.date, "%Y-%m-%d %H:%M")

end.date <- strptime("2010-2-17 23:45", "%Y-%m-%d %H:%M")
end.date <- format(end.date, "%Y-%m-%d %H:%M")

my_date = seq.POSIXt(as.POSIXct(start.date), as.POSIXct(end.date), by = "15 min")

my_data$Timestamp = my_date
str(my_data)
```

```
## 'data.frame':    4603 obs. of  3 variables:
##  $ Timestamp : POSIXct, format: "2010-01-01 01:15:00" "2010-01-01 01:30:00" ...
##  $ Power..kW.: num  165 152 147 154 154 ...
```

```
##  $ Temp..C.. : num  10.6 10.6 10.6 10.6 10.6 10.6 10.6 10.6 10 10 ...
```

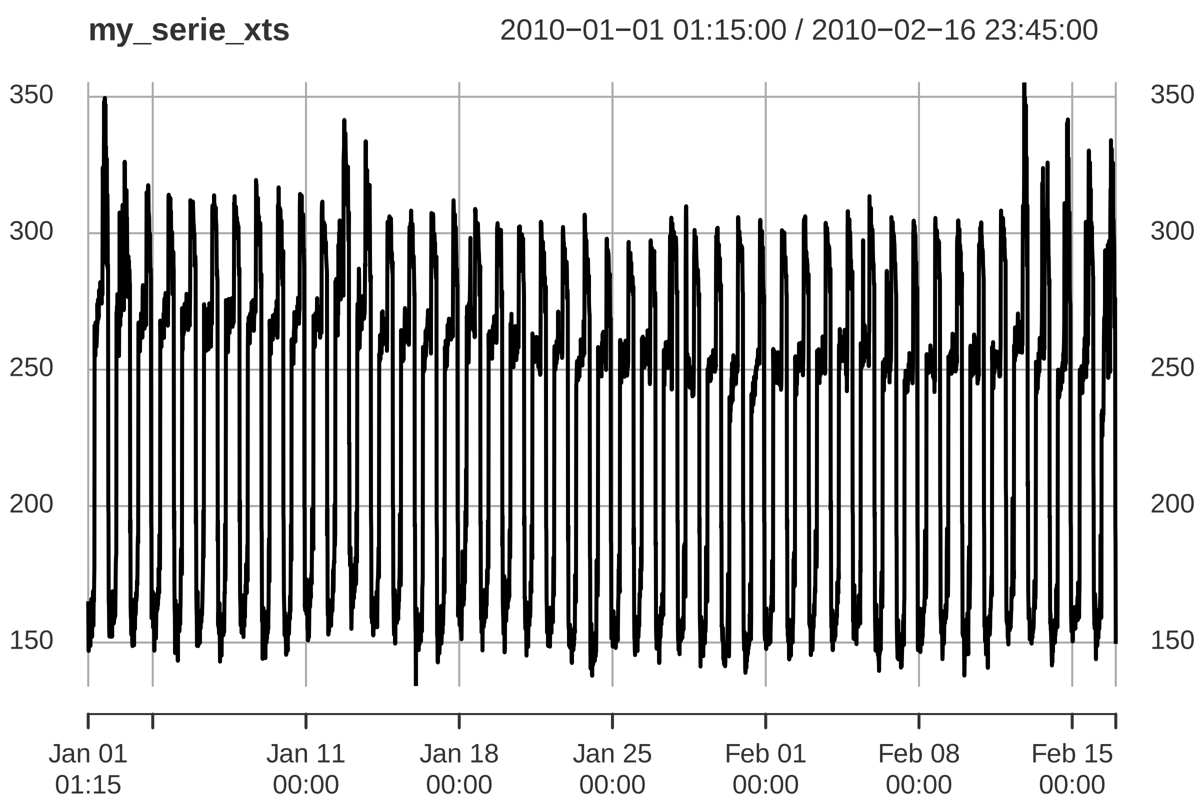
Serie analysis

Data splitting : between my_data1 used to train and test and my_data2 used for forecasting

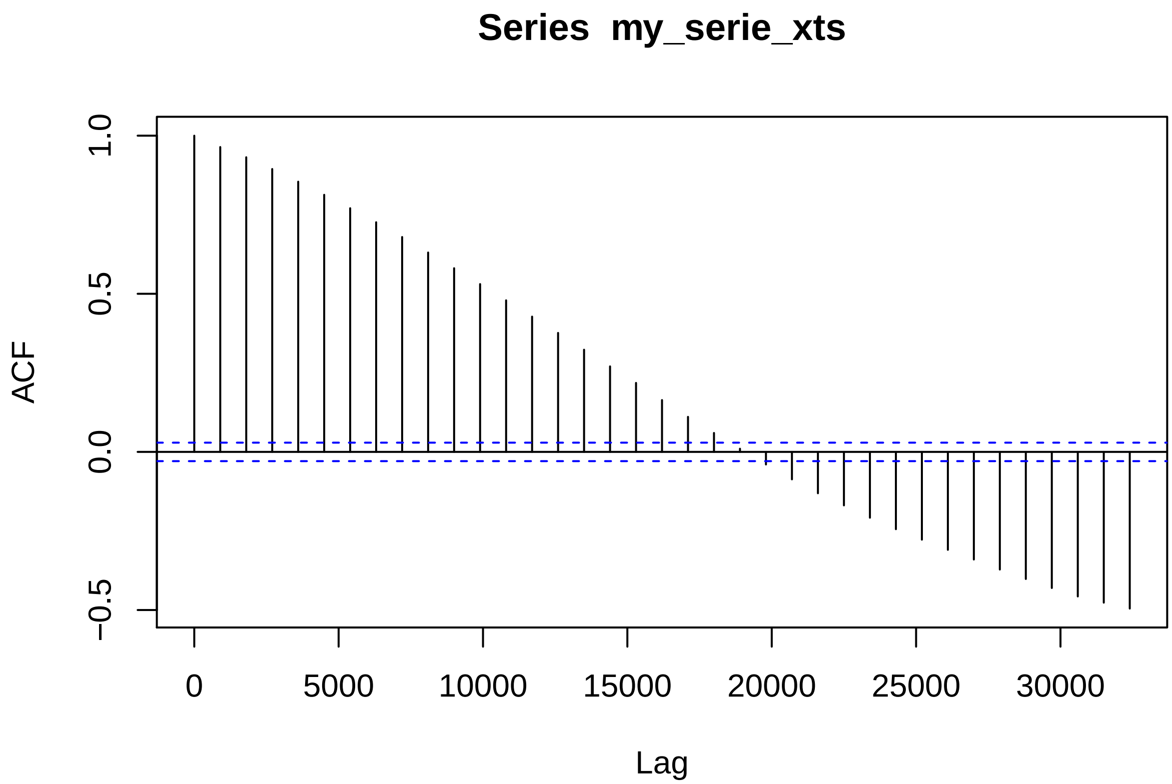
```
my_data1 = na.omit(my_data)
my_data2 = my_data %>% filter(is.na(Power..kW.))
```

Time serie creation and visualization. Serie seems to have a period of 24 hours

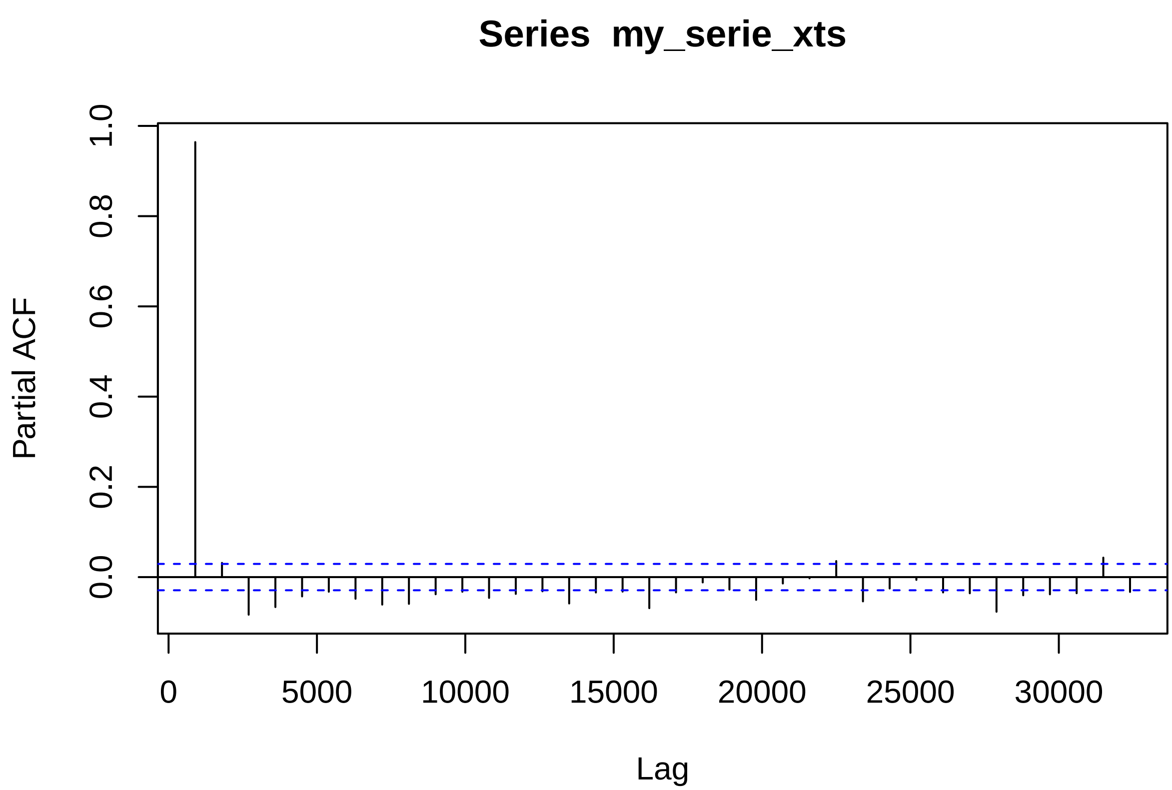
```
my_serie = ts(my_data1$Power..kW.,start= min(my_data1$Timestamp), freq=24*60/15)

# creating serie my_serie_xts for better visualization purpose
my_serie_xts = xts(my_data1$Power..kW., order.by =my_data1$Timestamp)

plot(my_serie_xts)
```



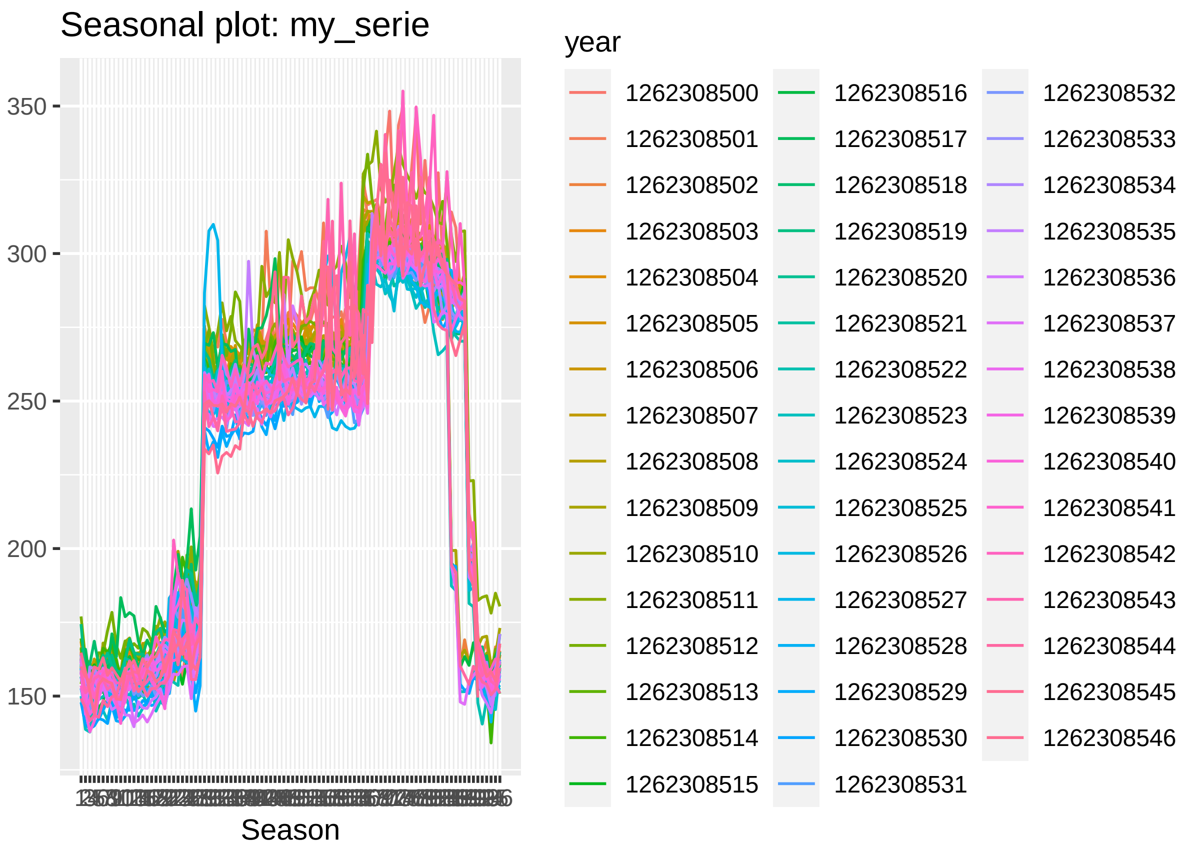

```
acf(my_serie_xts)
```




```r
pacf(my_serie_xts)
```




```r
ggseasonplot(my_serie)
```




```r
#Some  statistic for time serie
mean(my_serie)
```

```
## [1] 231.5873
```

```r
var(my_serie)
```

```
## [1] 3312.599
```

```r
my_acf = acf(my_serie,type="cor", plot = FALSE)
my_acf$acf[1:10,1,1]
```

```
##  [1] 1.0000000 0.9641389 0.9317837 0.8945770 0.8545477 0.8130328 0.7706198
##  [8] 0.7263123 0.6795421 0.6305817
```

It seems that there is no long-term (linear) trend. But it seems that there is a seasonal pattern, we can see it with the seasonal plot and by seeing auto-correlation table We guess a 24 hours period for the serie We noticed that there are some period of pic consumption (16h to 23 h) and period of less consumption (23h to 6h) We also noticed some cyclical pattern which can be more detailed using spectral analysis with Fast Fourier Analysis

Splitting between train and test We will use about 80% for the training data, to estimate model parameter and the 20% most recent, to evaluate forecast accuracy

```r
my_serie_test = tail(my_serie,901)
my_serie_train  = head(my_serie, 3606)
```

Building and evaluating models

We will use several models and compare them to select the best one

```
fit1=holt(my_serie_train,h=901, damped=FALSE)
fit2=holt(my_serie_train,h=901, damped=TRUE)
fit3=auto.arima(my_serie_train)
prev3=forecast(fit3,h=901)
fit4=HoltWinters(my_serie_train,  seasonal = "additive")
prev4 = predict(fit4,n.ahead=901)
```

We encounter issue with freq > 24 for seasonal hw damped We use seasonal HolWinters as workaround

RMSE

```
cat('Holt: ',sqrt(mean((fit1$mean-my_serie_test)^2)),'\n')
```

```
## Holt:  66.46902
```

```
cat('Damped Holt: ',sqrt(mean((fit2$mean-my_serie_test)^2)),'\n')
```

```
## Damped Holt:  61.81559
```

```
cat('auto.arima: ',sqrt(mean((prev3$mean-my_serie_test)^2)),'\n')
```

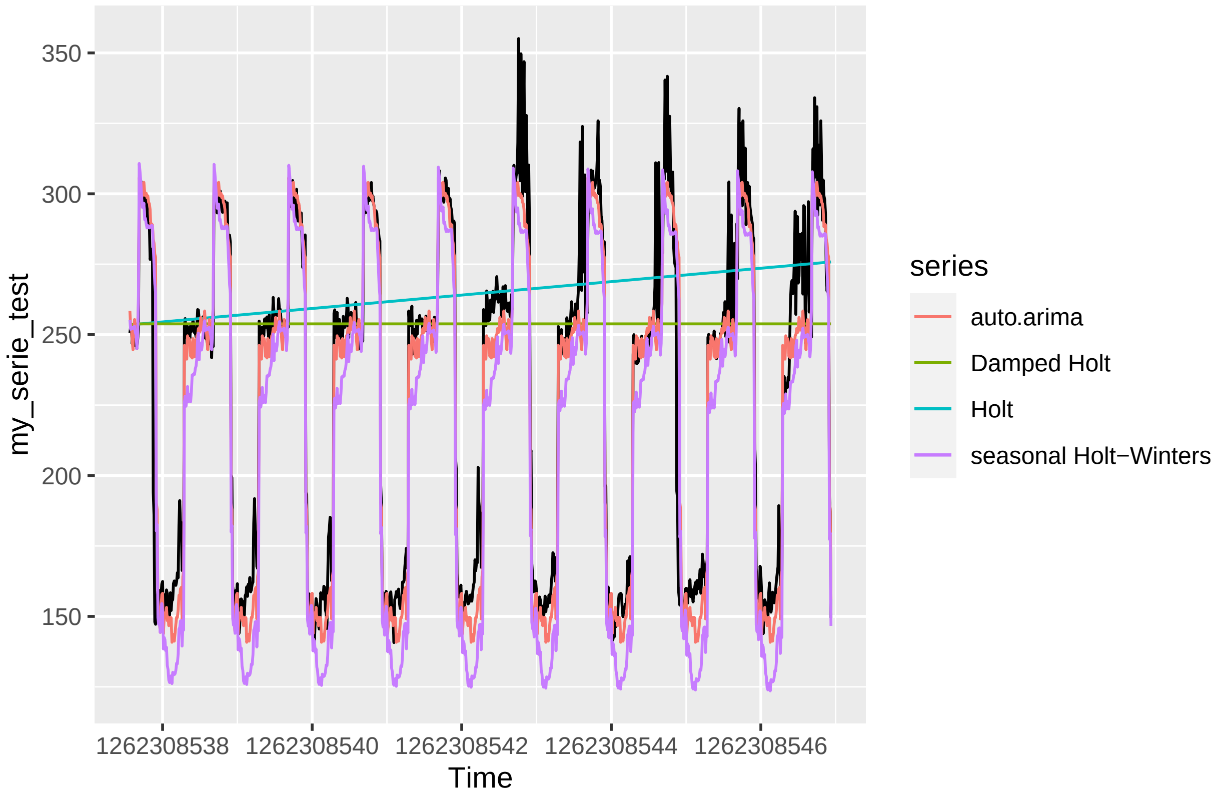
```
## auto.arima:  16.70526
```

```
cat('seasonal Holt-Winters: ',sqrt(mean((prev4-my_serie_test)^2)),'\n')
```

```
## seasonal Holt-Winters:  23.11928
```

We notice that the better model is auto.arima with perform a RMSE=16.70526 Cross validation It could better to use cross validation for better model selection

```
autoplot(my_serie_test) +
  autolayer(fit1$mean,series="Holt") +
  autolayer(fit2$mean,series="Damped Holt")+
  autolayer(prev3$mean,series="auto.arima")+
  autolayer(prev4,series="seasonal Holt-Winters")
```



Forecasting

Let's forecast 96 data using auto.arima Now we will use all data1 set as training ()

```
fit6=auto.arima(my_serie)
prev6=forecast(fit6,h=96)

#Checking model
summary(fit6)
```

```
## Series: my_serie
## ARIMA(5,0,0)(0,1,0)[96]
##
## Coefficients:
##          ar1     ar2     ar3      ar4     ar5
##       0.6920  0.0662  0.1451  -0.2963  0.1424
## s.e.  0.0149  0.0176  0.0175   0.0176  0.0149
##
## sigma^2 estimated as 118.4:  log likelihood=-16785.64
## AIC=33583.29   AICc=33583.31   BIC=33621.64
##
## Training set error measures:
##                       ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.04360809 10.75736 6.272848 -0.1250109 2.839516 0.7394969
##                      ACF1
## Training set 0.0008991053
```

```
#checkresiduals(fit6)
prev6$mean
```

```
## Time Series:
```

```
## Start = c(1262308546, 92)
## End = c(1262308547, 91)
## Frequency = 96
##  [1] 146.1747 146.6443 150.0067 149.2703 163.9390 161.1339 155.7695 149.2982
##  [9] 142.1431 152.6909 154.8982 154.1044 153.7430 148.5244 148.4431 157.6337
## [17] 161.6748 155.7363 151.3711 153.2953 158.5241 157.8383 154.7519 156.2643
## [25] 159.3707 162.8779 165.3830 167.6863 189.2897 174.3919 161.4936 158.9952
## [33] 163.0962 233.7970 232.0977 235.0982 225.5986 231.2989 232.5992 231.1994
## [41] 234.8995 233.6996 246.7997 261.9998 267.9998 269.1999 264.4999 267.5999
## [49] 277.4999 293.7999 257.2000 292.0000 291.9000 268.3000 270.8000 285.6000
## [57] 276.6000 279.0000 286.0000 264.4000 295.8000 295.4000 247.0000 263.3000
## [65] 269.8000 271.9000 297.2000 258.0000 269.3000 272.3000 249.1000 304.0000
## [73] 316.0000 299.2000 334.1000 308.1000 305.9000 330.9000 292.4000 317.3000
## [81] 316.0000 296.0000 325.9000 308.7000 290.6000 304.9000 299.2000 297.9000
## [89] 292.7000 270.6000 265.4000 270.9000 276.2000 192.7000 187.1000 149.5000
```

Forecasting with covariates

Here we will use dynamic regression by using variable temperature

```
fit7=auto.arima(my_serie, ,xreg=my_data1$Temp..C..)
prev7=forecast(fit7,h=96,xreg=my_data2$Temp..C..)


summary(fit7)
```

```
## Series: my_serie
## Regression with ARIMA(5,0,4)(0,1,0)[96] errors
##
## Coefficients:
##          ar1      ar2     ar3     ar4     ar5     ma1     ma2     ma3     ma4
##       0.0714  -0.1656  0.0328  -0.097  0.3285  0.6171  0.6555  0.6117  0.3554
## s.e.  0.1110   0.0725  0.0511   0.044  0.0547  0.1115  0.0876  0.0752  0.0675
##         xreg
##       0.5761
## s.e.  0.2270
##
## sigma^2 estimated as 117.8:  log likelihood=-16771.81
## AIC=33565.62   AICc=33565.68   BIC=33635.93
##
## Training set error measures:
##                       ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.04701593 10.72361 6.261653 -0.1267378 2.834427 0.7381771
##                    ACF1
## Training set 0.00216454
```

```
#checkresiduals(fit7)
prev7$mean
```

```
## Time Series:
## Start = c(1262308546, 92)
## End = c(1262308547, 91)
## Frequency = 96
##  [1] 144.3204 145.2309 150.6391 150.7399 164.6575 159.7480 154.7153 148.8979
##  [9] 142.2931 152.3392 153.8499 153.1107 152.9345 147.8288 147.7572 156.6620
```

```
## [17] 160.6778 155.1815 150.8351 152.7744 157.9079 157.5339 154.4915 156.0105
## [25] 159.1210 163.1766 165.6620 167.9818 189.5902 174.4044 161.4985 158.9913
## [33] 163.0975 233.4558 231.7562 234.7545 225.2515 231.0108 232.3127 230.9127
## [41] 234.6122 233.0654 246.1658 261.3666 267.3666 268.5664 263.8660 266.9661
## [49] 276.8664 294.0882 257.4881 292.2880 292.1880 268.0120 270.5120 285.3120
## [57] 276.3119 279.0000 286.0000 264.4000 295.8000 295.4000 247.0000 263.3000
## [65] 269.8000 272.1881 297.4880 258.2880 269.5880 272.6457 249.4457 304.3457
## [73] 316.3457 299.2000 334.1000 308.1000 305.9000 331.2457 292.7457 317.6457
## [81] 316.3457 297.2674 327.1674 309.9674 291.8674 306.8011 301.1011 299.8011
## [89] 294.6011 271.8674 266.6674 272.1674 277.4674 193.3337 187.7337 150.1337
```