Forecasting Models

Haya Naviwala

5/12/2022

R Markdown

```
library(tidyverse)
```

```
## Registered S3 methods overwritten by 'tibble':
##   method     from
##   format.tbl pillar
##   print.tbl  pillar
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   1.0.7
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```
## Warning: package 'dplyr' was built under R version 3.6.2
```

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

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.6.2
```

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

```
library(xts)
```

```
## Warning: package 'xts' was built under R version 3.6.2
```

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

```
## Warning: package 'zoo' was built under R version 3.6.2
```

```
##
## 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
```

```
dt <- read_csv('google.csv')
```

```
## Parsed with column specification:
## cols(
##   gvkey = col_double(),
##   datadate = col_double(),
##   fyearq = col_double(),
##   fqtr = col_double(),
##   tic = col_character(),
##   datafqtr = col_character(),
##   saleq = col_double()
## )
```

```
dt %>% glimpse()
```

```
## Observations: 72
## Variables: 7
## $ gvkey    <dbl> 160329, 160329, 160329, 160329, 160329, 160329, 16032...
## $ datadate <dbl> 20020331, 20020630, 20020930, 20021231, 20030331, 200...
## $ fyearq   <dbl> 2002, 2002, 2002, 2002, 2003, 2003, 2003, 2003, 2004,...
## $ fqtr     <dbl> 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2,...
## $ tic      <chr> "GOOGL", "GOOGL", "GOOGL", "GOOGL", "GOOGL", "GOOGL",...
## $ datafqtr <chr> "2002Q1", "2002Q2", "2002Q3", "2002Q4", "2003Q1", "20...
## $ saleq    <dbl> 42.285, 78.525, 130.787, 187.911, 248.618, 311.199, 3...
```

```
dt %>% head(2)
```

```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 2 x 7
##    gvkey datadate fyearq  fqtr tic   datafqtr saleq
##    <dbl>    <dbl>  <dbl> <dbl> <chr> <chr>    <dbl>
## 1 160329 20020331   2002     1 GOOGL 2002Q1    42.3
## 2 160329 20020630   2002     2 GOOGL 2002Q2    78.5
```

```
dt %>% tail(2)
```

```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 2 x 7
```

```
##    gvkey datadate fyearq  fqtr tic   datafqtr saleq
##    <dbl>    <dbl>  <dbl> <dbl> <chr> <chr>    <dbl>
## 1 160329 20190930   2019     3 GOOGL 2019Q3   40499
## 2 160329 20191231   2019     4 GOOGL 2019Q4   46075
```

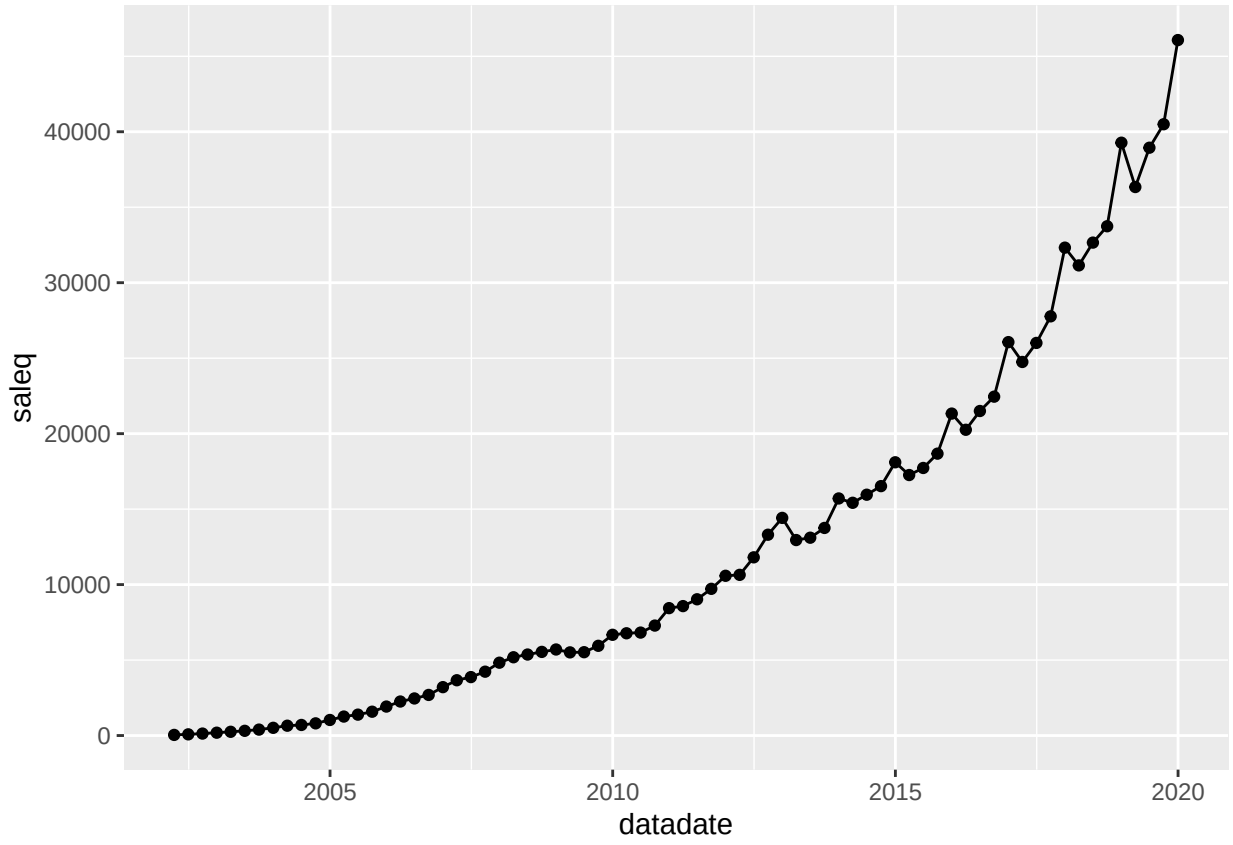
```
dt1 <- dt %>%
  mutate(datadate = ymd(datadate)) %>%
  arrange(datadate)

dt1 %>% head(2)
```

```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 2 x 7
##    gvkey datadate   fyearq  fqtr tic   datafqtr saleq
##    <dbl> <date>      <dbl> <dbl> <chr> <chr>    <dbl>
## 1 160329 2002-03-31   2002     1 GOOGL 2002Q1    42.3
## 2 160329 2002-06-30   2002     2 GOOGL 2002Q2    78.5
```

```
options(scipen=123)
dt1 %>%
  ggplot(aes(x=datadate, y=saleq)) + geom_point() + geom_line()
```



```
dtTest <- dt1 %>% arrange(desc(datadate)) %>% top_n(4, datadate)
dtTest
```

```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 4 x 7
##    gvkey datadate   fyearq  fqtr tic   datafqtr saleq
##    <dbl> <date>      <dbl> <dbl> <chr> <chr>    <dbl>
## 1 160329 2019-12-31   2019     4 GOOGL 2019Q4   46075
## 2 160329 2019-09-30   2019     3 GOOGL 2019Q3   40499
## 3 160329 2019-06-30   2019     2 GOOGL 2019Q2   38944
## 4 160329 2019-03-31   2019     1 GOOGL 2019Q1   36339
```

```
dtTrain <- dt1 %>%
  filter(!(datadate %in% dtTest$datadate)) %>% arrange(desc(datadate))

head(dtTrain, 2)
```

```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 2 x 7
##    gvkey datadate   fyearq  fqtr tic   datafqtr saleq
##    <dbl> <date>      <dbl> <dbl> <chr> <chr>    <dbl>
## 1 160329 2018-12-31   2018     4 GOOGL 2018Q4   39276
## 2 160329 2018-09-30   2018     3 GOOGL 2018Q3   33740
```

```
tail(dtTrain, 2)
```

```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 2 x 7
##    gvkey datadate   fyearq  fqtr tic   datafqtr saleq
##    <dbl> <date>      <dbl> <dbl> <chr> <chr>    <dbl>
## 1 160329 2002-06-30   2002     2 GOOGL 2002Q2    78.5
## 2 160329 2002-03-31   2002     1 GOOGL 2002Q1    42.3
```

```
#NOTE: the xts data set shown below, the date is not considered a variable. It is just a indicator of t
dtxts_Train <- xts(dtTrain$saleq,
                   order.by = dtTrain$datadate)
```

```r
head(dtxts_Train, 4)
```

```
##               [,1]
## 2002-03-31  42.285
## 2002-06-30  78.525
## 2002-09-30 130.787
## 2002-12-31 187.911
```

```r
dtxts <- xts(dt1$saleq,
                  order.by = dt1$datadate)
head(dtxts_Train, 4)
```

```
##               [,1]
## 2002-03-31  42.285
## 2002-06-30  78.525
## 2002-09-30 130.787
## 2002-12-31 187.911
```

```r
dtxts_Test <- xts(dtTest$saleq,
                  order.by = dtTest$datadate)
head(dtxts_Test)
```

```
##             [,1]
## 2019-03-31 36339
## 2019-06-30 38944
## 2019-09-30 40499
## 2019-12-31 46075
```

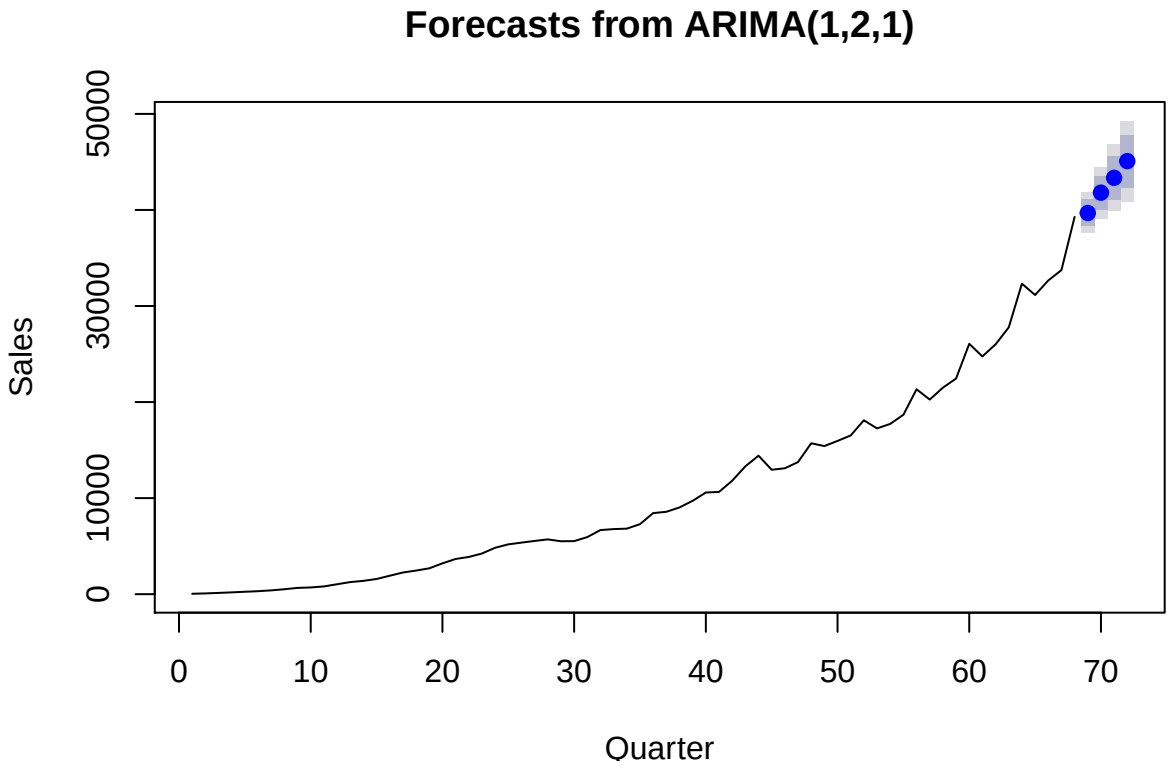
```r
Model1 <- auto.arima(dtxts_Train)
summary(Model1)
```

```
## Series: dtxts_Train
## ARIMA(1,2,1)
##
## Coefficients:
##           ar1      ma1
##       -0.3330  -0.8418
## s.e.   0.1373   0.0622
##
## sigma^2 estimated as 1132994:  log likelihood=-553.59
## AIC=1113.18   AICc=1113.57   BIC=1119.75
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 202.0644 1032.641 573.0415 2.701254 5.407492 0.7391444
##                      ACF1
## Training set -0.08618544
```

```r
forecastedModel1 <- forecast(Model1,4)
forecastedModel1
```

```
##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 69       39683.92 38319.81 41048.03 37597.69 41770.15
## 70       41799.38 40030.79 43567.98 39094.55 44504.21
## 71       43346.27 41077.64 45614.90 39876.69 46815.84
## 72       45082.48 42341.47 47823.49 40890.47 49274.50
```

```
plot(forecastedModel1, xlab = 'Quarter', ylab = 'Sales')
```




```
accuracy(forecastedModel1, dtxts_Test)
```

```
##                      ME     RMSE       MAE       MPE     MAPE      MASE
## Training set   202.0644 1032.641  573.0415  2.701254 5.407492 0.7391444
## Test set     -2013.7626 2666.164 2510.0219 -5.353278 6.430346 3.2375818
##                     ACF1
## Training set -0.08618544
## Test set              NA
```

```
#forecasting 2020
dt1xts <- xts(dt1$saleq, order.by = dt1$datadate)
head(dt1xts,2)
```

```
##              [,1]
## 2002-03-31 42.285
## 2002-06-30 78.525
```

```
tail(dt1xts,2)
```

```
##             [,1]
## 2019-09-30 40499
## 2019-12-31 46075
```

```
Model2 <- auto.arima(dt1xts)
summary(Model2)
```

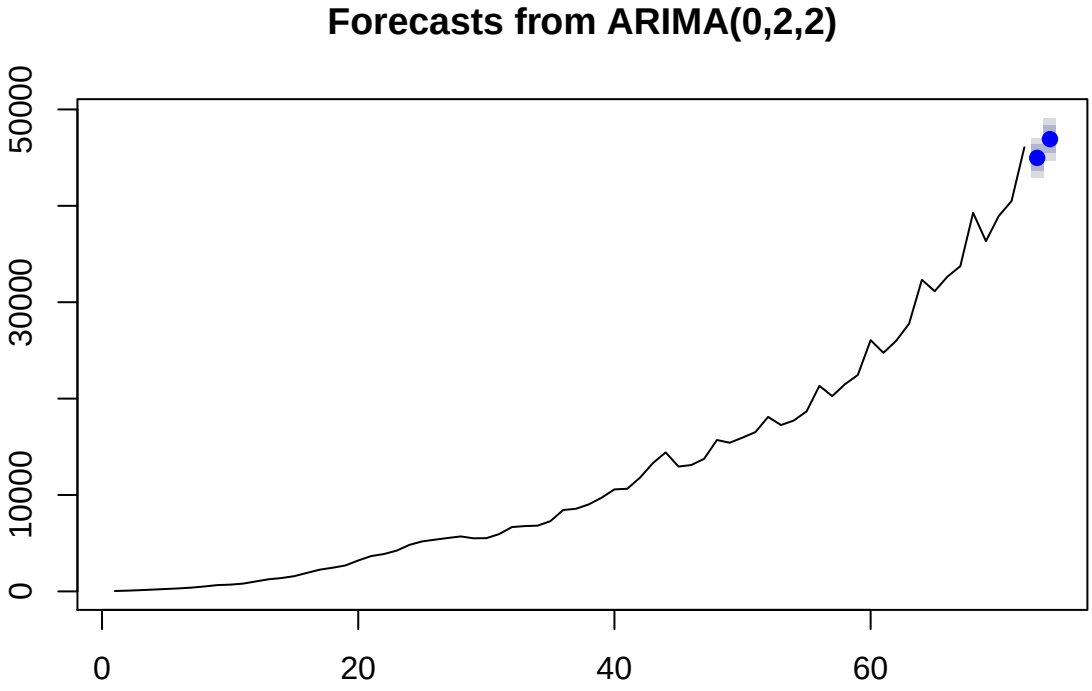
```
## Series: dt1xts
## ARIMA(0,2,2)
##
## Coefficients:
```

```
##           ma1     ma2
##       -1.6625  0.8511
## s.e.   0.0665  0.0703
##
## sigma^2 estimated as 1112485:  log likelihood=-587.69
## AIC=1181.39   AICc=1181.75   BIC=1188.14
##
## Training set error measures:
##                     ME     RMSE      MAE     MPE     MAPE      MASE
## Training set 139.2192 1025.027 616.9538 2.45016 5.961645 0.6779025
##                   ACF1
## Training set -0.03335
```

```
forecastedModel2 <- forecast(Model2, 2)
forecastedModel2
```

```
##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 73       44977.57 43625.86 46329.28 42910.30 47044.83
## 74       46921.92 45495.29 48348.56 44740.07 49103.78
```

```
plot(forecastedModel2)
```



Including Plots

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.