Future Sales Prediction

Kiuk Paeng

2024-01-03

Introduction

```
library('fpp3')
```

```
## -- Attaching packages --------------------------------------------- fpp3 0.5 --

## v tibble      3.2.1     v tsibble     1.1.3
## v dplyr       1.1.2     v tsibbledata 0.4.1
## v tidyr       1.3.0     v feasts      0.3.1
## v lubridate   1.9.3     v fable       0.3.3
## v ggplot2     3.4.4     v fabletools  0.3.4

## -- Conflicts ------------------------------------------------ fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```
library(tsibble)
library(dplyr)
library(forecast)
```

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

```
library(ggplot2)
```

Salesd dataset is provided by the Kaggle for the competition. Data fields are defined as below.

ID - an Id that represents a (Shop, Item) tuple within the test set shop_id - unique identifier of a shop item_id - unique identifier of a product item_category_id - unique identifier of item category item_cnt_day - number of products sold. You are predicting a monthly amount of this measure item_price - current price of an item date - date in format dd/mm/yyyy date_block_num - a consecutive month number, used for convenience. January 2013 is 0, February 2013 is 1,..., October 2015 is 33 item_name - name of item shop_name - name of shop item_category_name - name of item category

```
sales <- read.csv('/Users/kiukpaeng/Documents/1_Data Science/1. Personal Projects/predict-future-sales/
```

Let's review the first 6 row of the datset.

```
head(sales)
```

```
##         date date_block_num shop_id item_id item_price item_cnt_day
## 1 02.01.2013              0      59   22154     999.00            1
## 2 03.01.2013              0      25    2552     899.00            1
## 3 05.01.2013              0      25    2552     899.00           -1
## 4 06.01.2013              0      25    2554    1709.05            1
## 5 15.01.2013              0      25    2555    1099.00            1
## 6 10.01.2013              0      25    2564     349.00            1
```

Let review some summary statistics for the data set

```
summary(sales)
```

```
##      date           date_block_num     shop_id      item_id
##  Length:2935849     Min.   : 0.00   Min.   : 0   Min.   :    0
##  Class :character   1st Qu.: 7.00   1st Qu.:22   1st Qu.: 4476
##  Mode  :character   Median :14.00   Median :31   Median : 9343
##                     Mean   :14.57   Mean   :33   Mean   :10197
##                     3rd Qu.:23.00   3rd Qu.:47   3rd Qu.:15684
##                     Max.   :33.00   Max.   :59   Max.   :22169
##    item_price        item_cnt_day
##  Min.   :    -1.0   Min.   : -22.000
##  1st Qu.:   249.0   1st Qu.:   1.000
##  Median :   399.0   Median :   1.000
##  Mean   :   890.9   Mean   :   1.243
##  3rd Qu.:   999.0   3rd Qu.:   1.000
##  Max.   :307980.0   Max.   :2169.000
```

For the purpose of demonstrating the sales pattern and forecasting future sales, the original data is first grouped by sales data. Then, the number of items sold is summed as shown below.

```
sales_tsibble <- sales %>%
  group_by(date_block_num) %>%
  summarise(sales = sum(item_cnt_day)) %>%
  as_tsibble(index = date_block_num)
```

```
head(sales_tsibble)
```

```
## # A tsibble: 6 x 2 [1]
##   date_block_num  sales
##            <int>  <dbl>
## 1              0 131479
## 2              1 128090
## 3              2 147142
## 4              3 107190
## 5              4 106970
## 6              5 125381
```

```r
sales_tsibble_t <- sales_tsibble %>%
  mutate(`5-MA` = slider::slide_dbl(sales, mean,.before = 4, .after = 2, .complete = TRUE))
```

Moving Average Smoothing

In order to identify the trend, let's use the simple moving average moving smoothing. A moving average of order 5 is used.

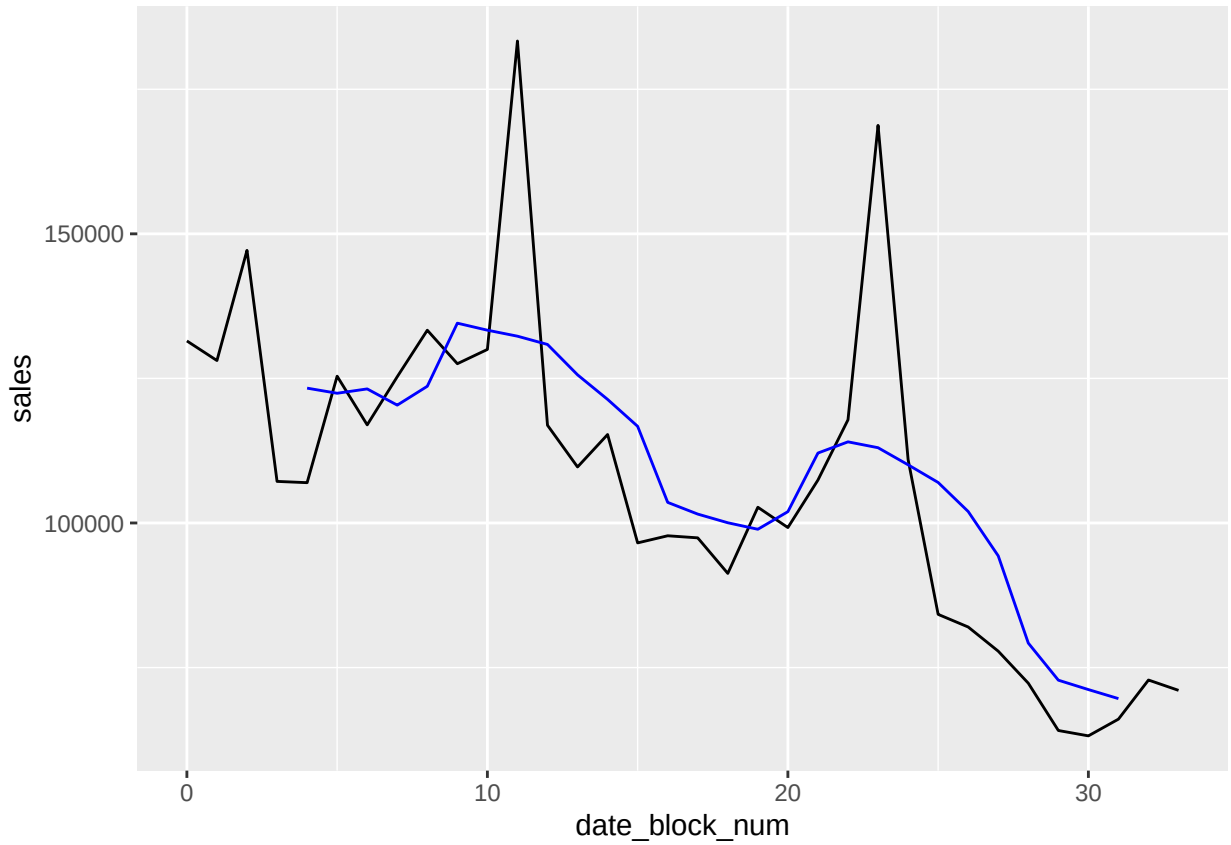
```r
head(sales_tsibble_t)
```

```
## # A tsibble: 6 x 3 [1]
##   date_block_num  sales  `5-MA`
##            <int>  <dbl>   <dbl>
## 1              0 131479     NA
## 2              1 128090     NA
## 3              2 147142     NA
## 4              3 107190     NA
## 5              4 106970 123317.
## 6              5 125381 122433.
```

We can observe that the trend-cycle (in blue) is smoother than the original data. It effectively captures the primary movement of the time series while filtering out the fluctuations. It is shown that the sales is on down trend with some fluctuation.

```r
ggplot(data = sales_tsibble_t, mapping = aes(x = date_block_num)) +
  geom_line(aes(y = sales)) +
  geom_line(aes(y = `5-MA`), color = "blue")
```

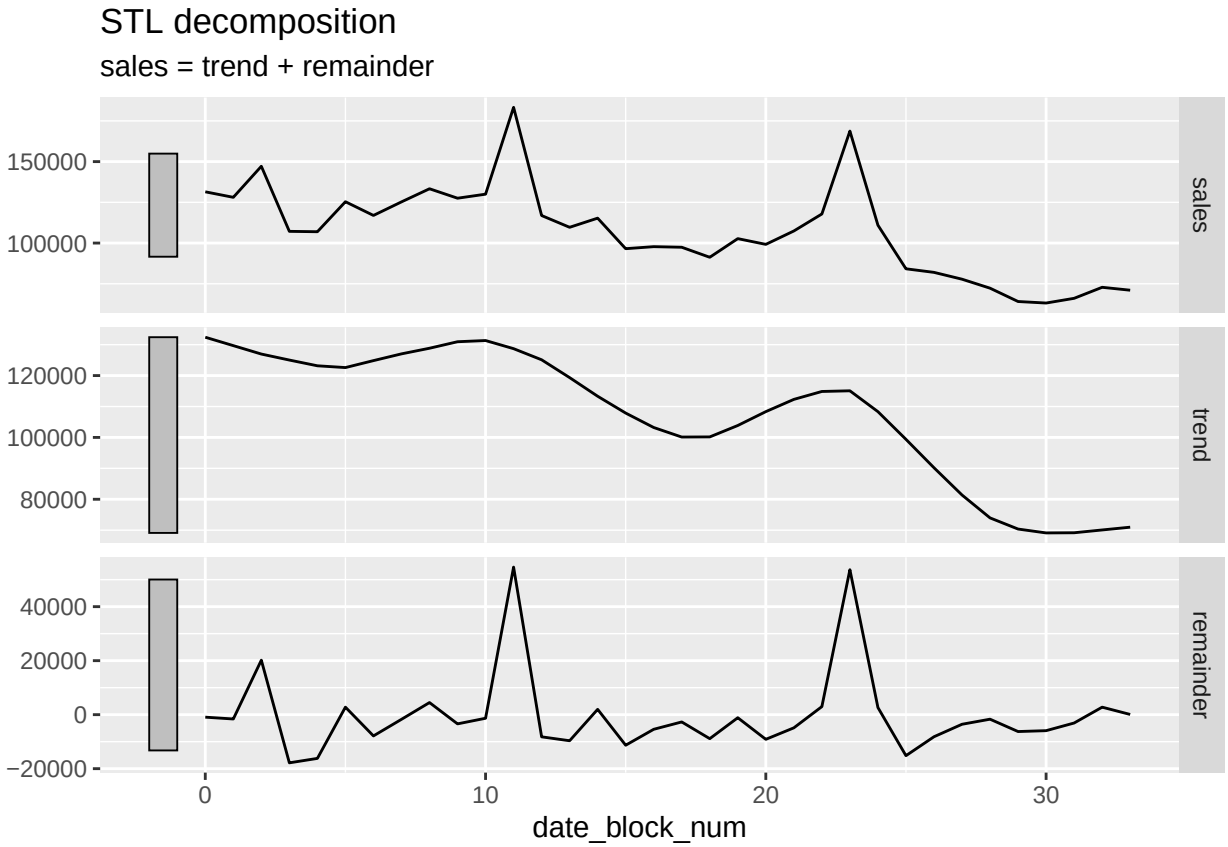
```
## Warning: Removed 6 rows containing missing values (`geom_line()`).
```



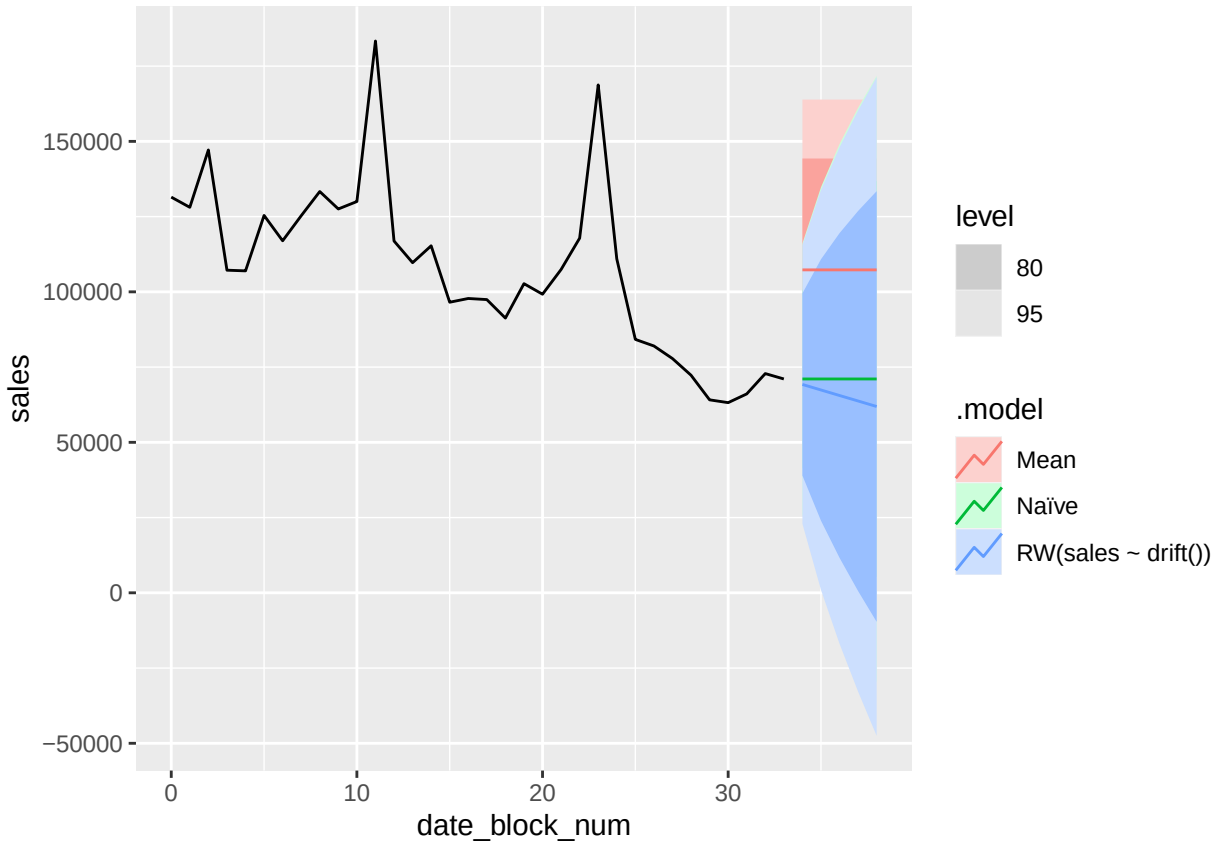
STL (Seasonal and Trend decomposition using Loess)

By employing time series decomposition, we can examine the components of the time series. As shown below it identifies a downtrend, and the remainder behaves like white noise, displaying no discernible patterns or trends. It is deemed that there is no strong seaonality in the time serise.

```
sales_tsibble_t %>%
  model(stl = STL(sales)) %>%
  components() %>%
  autoplot()
```




```
sales_tsibble%>%
  model(RW(sales ~ drift()),
        Mean = MEAN(sales),
        `Naïve` =  NAIVE(sales)) %>%
  forecast(h = 5) %>%
  autoplot(sales_tsibble)
```


150000
100000
50000
0
−50000
sales
0
10
20
30
date_block_num
level
80
95
.model
Mean
Naïve
RW(sales ~ drift())