

R forecasting EDA

Mattia

2022-12-15

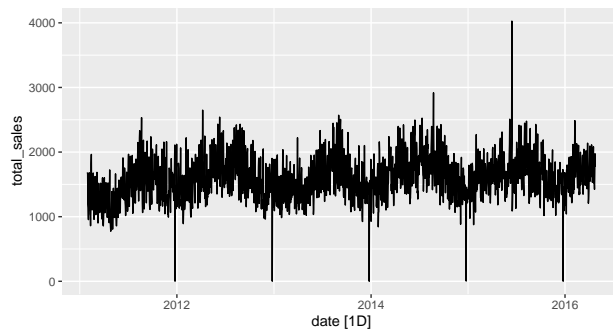
Import data

```
full_dataset <- read.csv("/Users/mattiacintioli/UvA_DS/Forecasting/Project R/full_dataset.csv")
```

plot sum across all products in time

```
full_dataset$date <- as.Date(full_dataset$date, format="%m-%d-%Y")
walmart <- as_tsibble(full_dataset, index = date, key = id)
walmart %>%
  summarise(total_sales = sum(value)) %>%
  autoplot()
```

Plot variable not specified, automatically selected '.vars = total_sales'



Top 8 sold products

```
product_1 <- full_dataset %>% filter(id == "FOODS_3_586_TX_3_validation")
product_1$date <- as.Date(product_1$date, format="%m-%d-%Y")
product_1_ts <- as_tsibble(product_1, index = date)

#product_2 <- full_dataset %>% filter(id == "FOODS_3_090_TX_3_validation")
product_2 <- full_dataset %>% filter(id == "FOODS_3_372_TX_3_validation")
product_2$date <- as.Date(product_2$date, format="%m-%d-%Y")
product_2_ts <- as_tsibble(product_2, index = date)

#product_3 <- full_dataset %>% filter(id == "FOODS_3_252_TX_3_validation")
product_3 <- full_dataset %>% filter(id == "FOODS_3_464_TX_3_validation")
product_3$date <- as.Date(product_3$date, format="%m-%d-%Y")
product_3_ts <- as_tsibble(product_3, index = date)
```

```

#product_4 <- full_dataset %>% filter(id == "FOODS_3_555_TX_3_validation")
product_4 <- full_dataset %>% filter(id == "FOODS_3_166_TX_3_validation")
product_4$date <- as.Date(product_4$date, format="%m-%d-%Y")
product_4_ts <- as_tsibble(product_4, index = date)

#product_5 <- full_dataset %>% filter(id == "FOODS_3_377_TX_3_validation")
product_5 <- full_dataset %>% filter(id == "FOODS_3_171_TX_3_validation")
product_5$date <- as.Date(product_5$date, format="%m-%d-%Y")
product_5_ts <- as_tsibble(product_5, index = date)

#product_6 <- full_dataset %>% filter(id == "FOODS_3_030_TX_3_validation")
#product_6$date <- as.Date(product_6$date, format="%m-%d-%Y")
#product_6_ts <- as_tsibble(product_6, index = date)

#product_7 <- full_dataset %>% filter(id == "FOODS_3_202_TX_3_validation")
#product_7$date <- as.Date(product_7$date, format="%m-%d-%Y")
#product_7_ts <- as_tsibble(product_7, index = date)

#product_8 <- full_dataset %>% filter(id == "FOODS_3_714_TX_3_validation")
#product_8$date <- as.Date(product_8$date, format="%m-%d-%Y")
#product_8_ts <- as_tsibble(product_8, index = date)

```

5 representative products

```

# install and load cowplot package
#install.packages("cowplot")
library(cowplot)

```

```

##
## Attaching package: 'cowplot'

```

```

## The following object is masked from 'package:lubridate':

```

```

##
## stamp

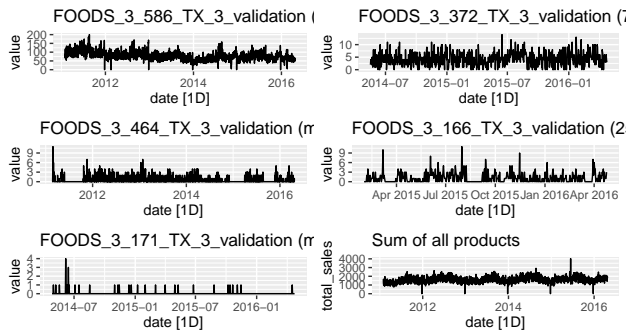
```

```

# create a list of the plots
plots <- list(
  product_1_ts %>% autoplot(value) + ggtitle(paste(product_1_ts$id[1], "(max)")),
  product_2_ts %>% autoplot(value) + ggtitle(paste(product_2_ts$id[1], "(75th percentile)")),
  product_3_ts %>% autoplot(value) + ggtitle(paste(product_3_ts$id[1], "(median)")),
  product_4_ts %>% autoplot(value) + ggtitle(paste(product_4_ts$id[1], "(25th percentile)")),
  product_5_ts %>% autoplot(value) + ggtitle(paste(product_5_ts$id[1], "(min)")),
  walmart %>% summarise(total_sales = sum(value)) %>% autoplot(total_sales) + ggtitle("Sum of all products")
)

# use plot_grid to specify the layout of the plots
plot_grid(plotlist = plots, nrow = 3, ncol = 2)

```



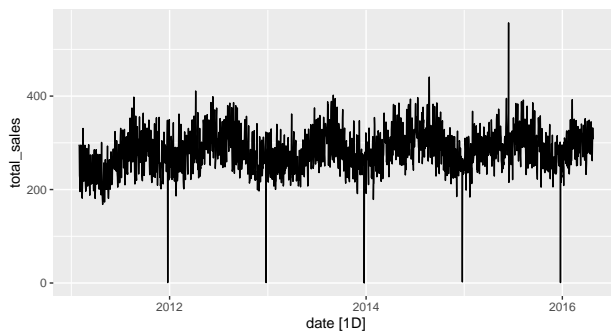
boxcox sales

```
lambda <- walmart %>%
  summarise(total_sales = sum(value)) %>%
  features (total_sales, features = guerrero) %>%
  pull(lambda_guerrero)

box_walmart <- walmart %>%
  summarise(total_sales = sum(value)) %>%
  mutate(total_sales = box_cox(total_sales, lambda))

box_walmart %>%
  autoplot()
```

Plot variable not specified, automatically selected '.vars = total_sales'

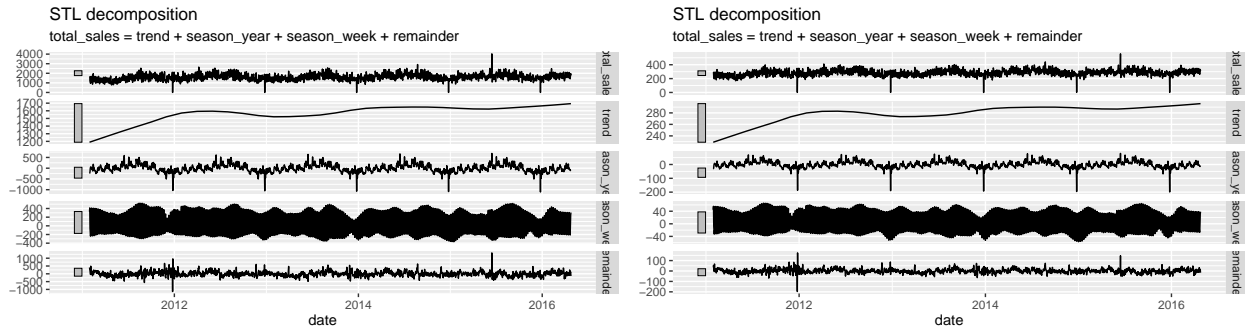


STL decomposition of non-transformed and trans-

formed data

```
dcmp <- walmart %>%
  summarise(total_sales = sum(value)) %>%
  #model(stl =STL(total_sales, s.window = "periodic"))
  model(stl =STL(total_sales))
components(dcmp) %>%
  autoplot()
#components(dcmp)

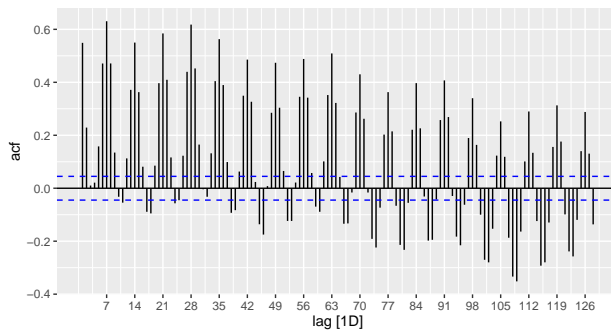
dcmp1 <- box_walmart %>%
  model(stl =STL(total_sales))
components(dcmp1) %>%
  autoplot()
```



ACF plot

```
walmart %>%
  summarise(total_sales = sum(value)) %>%
  ACF(lag_max = 128) %>%
  autoplot()
```

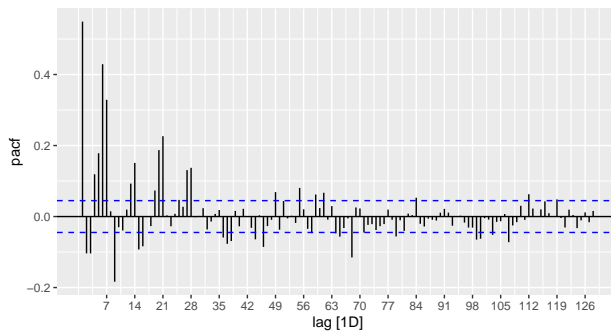
Response variable not specified, automatically selected 'var = total_sales'



PACF plot

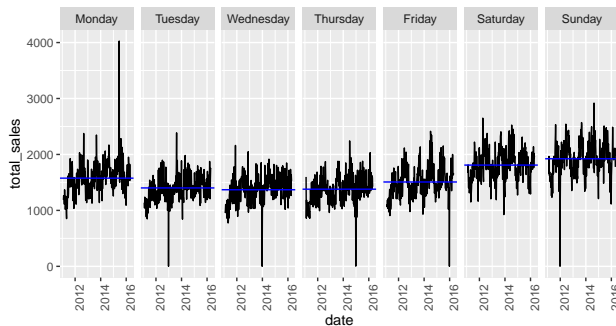
```
walmart %>%
  summarise(total_sales = sum(value)) %>%
  PACF(lag_max = 128) %>%
  autoplot()
```

Response variable not specified, automatically selected 'var = total_sales'



weekly seasonality

```
walmart %>%
  summarise(total_sales = sum(value)) %>%
  gg_subseries(total_sales, period = "week")
```



weekly seasonality of 5 products

```
# install and load cowplot package
#install.packages("cowplot")
library(cowplot)

# create a list of the plots
plots <- list(
  product_1_ts %>% gg_subseries(value, period = "week") + ggtitle(paste(product_1_ts$id[1], "(max)")) +
  product_2_ts %>% gg_subseries(value, period = "week") + ggtitle(paste(product_2_ts$id[1], "(75th perc")) +
  product_3_ts %>% gg_subseries(value, period = "week") + ggtitle(paste(product_3_ts$id[1], "(median)")) +
  product_4_ts %>% gg_subseries(value, period = "week") + ggtitle(paste(product_4_ts$id[1], "(25th perc")) +
  product_5_ts %>% gg_subseries(value, period = "week") + ggtitle(paste(product_5_ts$id[1], "(min)")) +
  walmart %>% summarise(total_sales = sum(value)) %>%
  gg_subseries(total_sales, period = "week") + ggtitle("Sum of all products") + scale_x_date(date_breaks = "1 week")
)
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

```
# use plot_grid to specify the layout of the plots
plot_grid(plotlist = plots, nrow = 3, ncol = 2)
```

