SOK-1005-Assignment4

Casper Andresen

```
rm (list = ls())
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.4.0      v purrr   0.3.4
v tibble  3.1.8      v dplyr   1.0.10
v tidyr   1.2.1      v stringr 1.4.1
v readr   2.1.2      v forcats 0.5.2

Warning: package 'ggplot2' was built under R version 4.2.2

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

```
library(janitor)
```

```

Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```
library(lubridate)
```

```
Attaching package: 'lubridate'

The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```
library(zoo)
```

```
Warning: package 'zoo' was built under R version 4.2.2
```

```
Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

```
library(quantmod)
```

```
Loading required package: xts

Warning: package 'xts' was built under R version 4.2.3
```

```
####################################### WARNING ########################################
# We noticed you have dplyr installed. The dplyr lag() function breaks how    #
# base R's lag() function is supposed to work, which breaks lag(my_xts).      #
#                                                                             #
# Calls to lag(my_xts) that you enter or source() into this session won't     #
# work correctly.                                                             #
#                                                                             #
# All package code is unaffected because it is protected by the R namespace   #
# mechanism.                                                                  #
#                                                                             #
# Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
#                                                                             #
# You can use stats::lag() to make sure you're not using dplyr::lag(), or you #
# can add conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop   #
```

```
# dplyr from breaking base R's lag() function.                                 #
################################### WARNING ###################################

Attaching package: 'xts'

The following objects are masked from 'package:dplyr':

    first, last

Loading required package: TTR

Warning: package 'TTR' was built under R version 4.2.3

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

```
options(dplyr.summarise.inform = FALSE)
```

```
df <- read.csv("https://raw.githubusercontent.com/uit-sok-1005-v23/uit-sok-1005-v23.github
  clean_names()
```

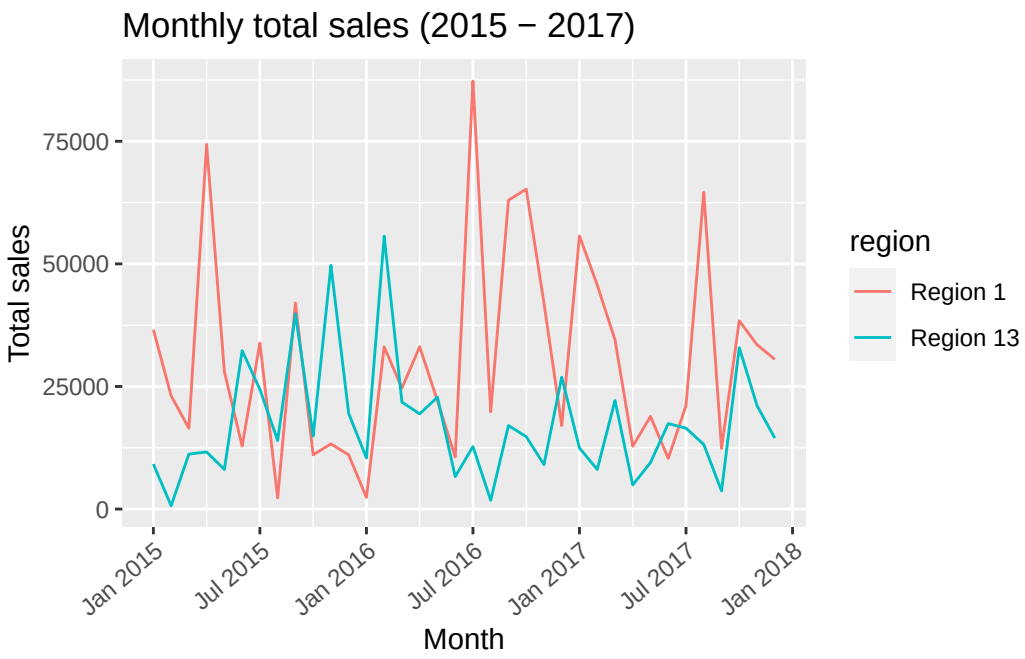
Task 1

```
df$order_date <- as.Date(df$order_date, "%Y-%m-%d")

table1 <- df %>%
  filter(between(order_date, as.Date('2017-10-1'), as.Date('2017-12-31'))) %>%
  filter(region %in% c("Region 1", "Region 9")) %>%
  filter(customer_segment %in% c("Corporate", "Consumer")) %>%
  mutate(year = year(order_date)) %>%
  mutate(month = month(order_date)) %>%
  group_by(region, month, customer_segment) %>%
  summarize(total_sales = sum(sales))

table1
```

```
# A tibble: 11 x 4
# Groups:   region, month [6]
```

```
   region   month customer_segment total_sales
   <chr>    <dbl> <chr>                   <dbl>
 1 Region 1    10 Consumer                816.
 2 Region 1    10 Corporate               295.
 3 Region 1    11 Consumer               9480.
 4 Region 1    11 Corporate              8565.
 5 Region 1    12 Consumer                210.
 6 Region 1    12 Corporate             13261.
 7 Region 9    10 Consumer               5909.
 8 Region 9    10 Corporate             16781.
 9 Region 9    11 Consumer                192.
10 Region 9    11 Corporate              5463.
11 Region 9    12 Corporate              9378.
```

```r
figure1 <- df %>%
  mutate(year = year(order_date)) %>%
  mutate(month = month(order_date)) %>%
  filter(region %in% c("Region 1", "Region 13")) %>%
  filter(year %in% c("2015", "2016", "2017")) %>%
  group_by(region, month, year) %>%
  summarize(total_sales = sum(sales)) %>%
  mutate(date = zoo::as.yearmon(paste(year, month), "%Y %m"))

figure1 <- subset(figure1, select = -c(month, year))

figure1 %>%
  ggplot(aes(x=date, y=total_sales, color=region)) +
  labs(title = "Monthly total sales (2015 - 2017)", x = "Month",
  y = "Total sales") +
  theme(axis.text.x = element_text(angle = 40, hjust = 1)) +
  geom_line()
```




```
table2 <- figure1 %>%
  pivot_wider(names_from = region, values_from = total_sales) %>%
  filter(`Region 13` > `Region 1`)

table2
```

```
# A tibble: 10 x 3
   date      `Region 1` `Region 13`
   <yearmon>      <dbl>       <dbl>
 1 Jan 2016       2362.      10408.
 2 Feb 2016      33085.      55632.
 3 May 2016      22069.      22822.
 4 Jun 2015      12845.      32307.
 5 Jun 2017      10335.      17430.
 6 Aug 2015       2267.      13985.
 7 Oct 2015      11058.      14885.
 8 Nov 2015      13290.      49686.
 9 Dec 2015      11048.      19515.
10 Dec 2016      17020.      26890.
```

```r
table3 <- df %>%
  mutate(year = year(order_date)) %>%
  mutate(month = month(order_date)) %>%
  filter(year == 2017) %>%
  filter(!region %in% c("Region 3", "Region 5", "Region 8")) %>%
  group_by(region, customer_segment, product_category) %>%
  summarize(avg_profit = mean(profit))

table3
```

```
# A tibble: 119 x 4
# Groups:   region, customer_segment [40]
   region   customer_segment product_category avg_profit
   <chr>    <chr>            <chr>                 <dbl>
 1 Region 1 Consumer         Furniture              375.
 2 Region 1 Consumer         Office Supplies        -37.1
 3 Region 1 Consumer         Technology             963.
 4 Region 1 Corporate        Furniture             -223.
 5 Region 1 Corporate        Office Supplies         38.5
 6 Region 1 Corporate        Technology             241.
 7 Region 1 Home Office      Furniture             -319.
 8 Region 1 Home Office      Office Supplies        -10.5
 9 Region 1 Home Office      Technology            -582.
10 Region 1 Small Business   Furniture             -307.
# ... with 109 more rows
```

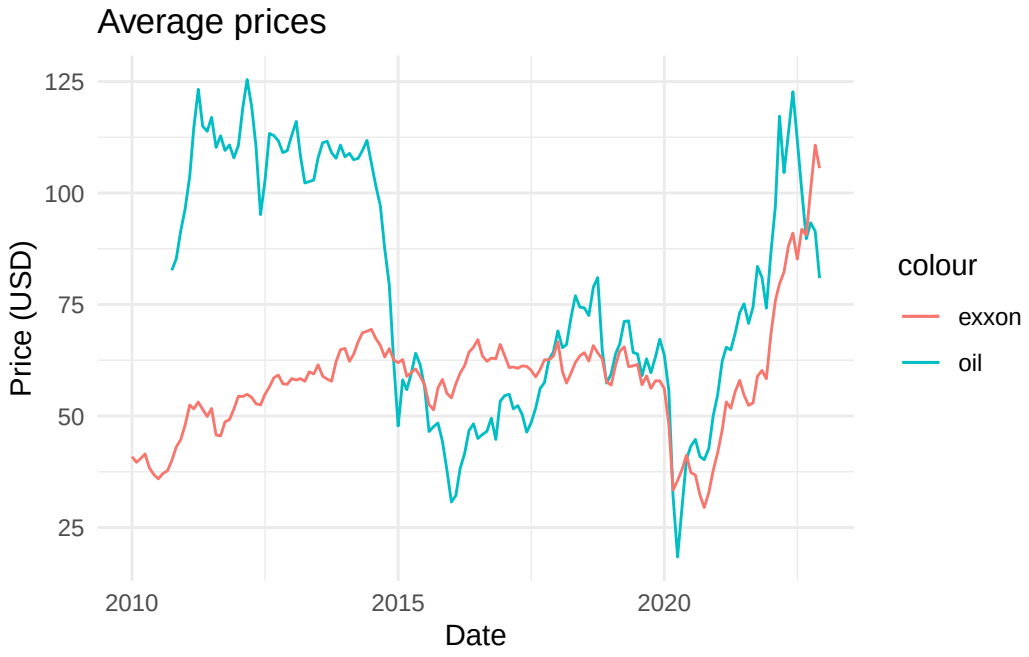
Task 2

```r
xom <- data.frame(getSymbols("XOM", src = "yahoo", from = "2010-1-04", to = "2022-12-31",
  clean_names() %>%
  rownames_to_column(., var = 'date')

xom$date <- as.Date(xom$date, "%Y-%m-%d")

xom <- xom %>%
  mutate(year = year(date)) %>%
  mutate(month = month(date)) %>%
  group_by(year, month) %>%
  summarise(exxon = weighted.mean(xom_adjusted, xom_volume)) %>%
```

```
  mutate(date = make_date(year, month))
```

```
brent <- data.frame(getSymbols("DCOILBRENTEU", src = "FRED",from = "2010-1-04", to = "2022
  clean_names() %>%
  rownames_to_column(., var = 'date')

brent <- brent %>%
  mutate(year = year(date)) %>%
  mutate(month = month(date)) %>%
  group_by(year, month) %>%
  summarise(oil= mean(dcoilbrenteu, na.rm=TRUE)) %>%
  mutate(date = make_date(year, month))
```

```
ggplot() +
  geom_line(data = brent, aes(x=date,y=oil, col = "oil")) +
  geom_line(data = xom, aes(x=date,y=exxon,col="exxon")) +
  labs(title = "Average prices", x = "Date",
  y = "Price (USD)") +
  theme_minimal()
```



On the figure we can clearly see the 2014/15 drop in oil prices. The drop didn't have much

effect on Exxons stock prices.