Assignment 4

“[8]”

```r
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   1.0.9
v tidyr   1.2.0     v stringr 1.4.0
v readr   2.1.2     v forcats 0.5.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```r
library(lubridate)
```

```

Attaching package: 'lubridate'

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

    date, intersect, setdiff, union
```

```r
library(quantmod)
```

```
Loading required package: xts
Loading required package: zoo

Attaching package: 'zoo'
```

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

    as.Date, as.Date.numeric


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

```r
library(janitor)
```

```

Attaching package: 'janitor'

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

    chisq.test, fisher.test
```

```
library(plotly)
```

```
Attaching package: 'plotly'

The following object is masked from 'package:ggplot2':

    last_plot

The following object is masked from 'package:stats':

    filter

The following object is masked from 'package:graphics':

    layout
```

```
library(knitr)
library(rvest)
```

```
Attaching package: 'rvest'

The following object is masked from 'package:readr':

    guess_encoding
```

Task 1

1)

For the last 3 months of 2017, calculate the total Sales by month, for Region 1 and Region 9 in the Customer_Segment, Corporate, and Consumer. This output is Table 1.

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

```
#total sale

table1 <- df %>%
  select(order_date, sales, customer_segment, region)

#select variabls
```

```
#calculate the total sale by month


table1 <- table1 %>%
  filter(customer_segment == "Corporate" | customer_segment == "Consumer") %>%
  filter(region == "Region 1" | region == "Region 9")

#filter out variables and region
```

```
table1$order_date <- as.Date(table1$order_date)

#set date as date
```

```
#select the date

table1 <- table1[table1$order_date > "2017-09-01" &
             table1$order_date < "2017-12-31", ]

#source: https://www.google.com/search?q=how+to+filter+out+date+in+rstudio&sxsrf=APwXEdej_
```

2)

Make a plot of the monthly total Sales in Region 1 and Region 13 in 2015, 2016, and 2017. This output is Figure 1.

```
figur1 <- df %>%
  select(order_date, sales, region)

figur1 <- figur1 %>%
  filter(region == "Region 1" | region == "Region 13")
```

```r
figur1$order_date <- as.Date(figur1$order_date)

figur1 <- figur1[figur1$order_date > "2015-01-01" &
             figur1$order_date < "2017-12-31", ]
```

```r
figur1$year <- strftime(figur1$order_date, "%Y")
figur1$month <- strftime(figur1$order_date, "%m")
#source: https://www.google.com/search?q=how+to+change+from+monthly+to+yearly+in+rstudi&oq
```

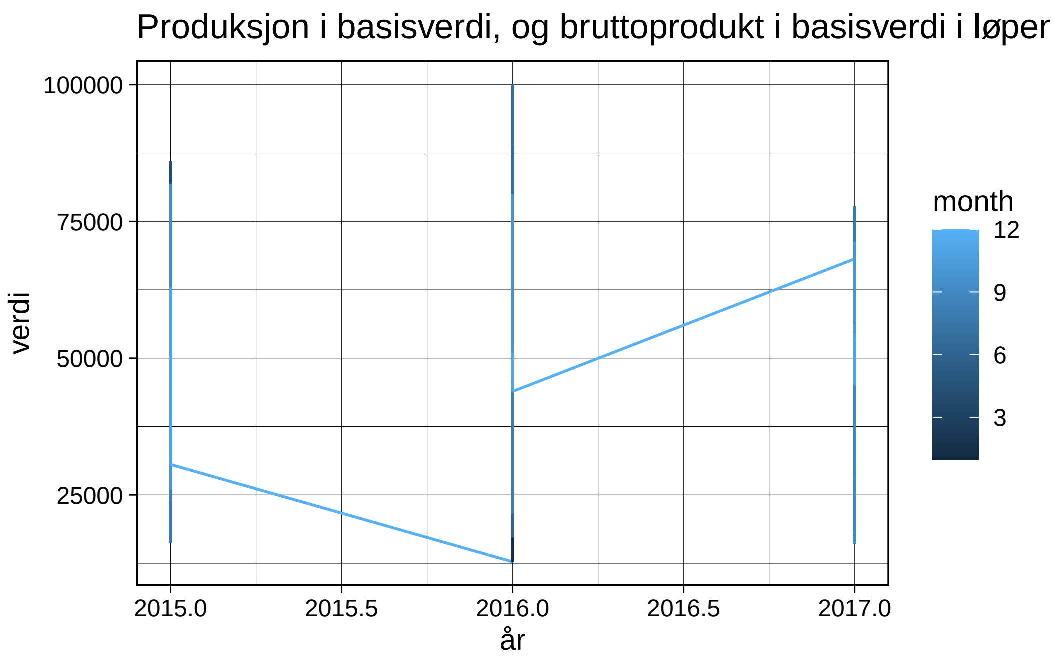
```r
figur1 <- figur1 %>%
  aggregate(sales ~ month + year,
            FUN = sum)

#source: https://www.google.com/search?q=how+to+change+from+monthly+to+yearly+in+rstudi&oq
```

```r
figur1$month <- as.numeric(figur1$month)
figur1$year <- as.numeric(figur1$year)
```

```r
figur1 <- figur1 %>%
  group_by(year,month) %>%
summarise(total = sum(sales))
```

```
`summarise()` has grouped output by 'year'. You can override using the
`.groups` argument.
```

```r
figur1 %>%
  ggplot(aes(x=year, y=total, col=month)) +
  geom_line()+
  ylab(expression("verdi")) +
  xlab("år") +
  labs(title = "Produksjon i basisverdi, og bruttoprodukt i basisverdi i løpende priser")+
  theme_linedraw()
```



3)

In Figure 1, identify the months where the total Sales in Region 13 is greater than the total Sales in Region 1. This output is Table 2.

4)

Find the average Profit per Customer_Segment and Product_Category in 2017, for all regions except Region 3, 5 and 8. What segment produced the highest average profit? This output is Table 3.

```
table3 <- df %>%
  select(order_date, profit, customer_segment, region, product_category)
```

```
table3 <- table3 %>%
 filter(region == "Region 1" | region == "Region 2" | region == "Region 4" | region == "Re
```

```
table3$order_date <- as.Date(table3$order_date)

table3 <- table3[table3$order_date > "2017-01-01" &
```

```
               table3$order_date < "2017-12-31", ]

#source: https://www.google.com/search?q=how+to+filter+out+date+in+rstudio&sxsrf=APwXEdej_
```

```
table3$year <- strftime(table3$order_date, "%Y")
table3$month <- strftime(table3$order_date, "%m")
#source: https://www.google.com/search?q=how+to+change+from+monthly+to+yearly+in+rstudi&oq
```

```
table3 <- table3 %>%
  group_by(month) %>% summarise(Average=mean(profit))
```

Task 2

```
getSymbols("XOM", src = "yahoo")
```

```
[1] "XOM"
```

```
XOM <- as.data.frame(XOM)
XOM <- tibble::rownames_to_column(XOM, var = "Date")
```

```
XOM <- data.frame(getSymbols("XOM", src = "yahoo", from = "2010-1-04", to = "2022-12-31",
XOM <- tibble::rownames_to_column(XOM, var = "Date")

oil_price <- data.frame(getSymbols("DCOILBRENTEU", src = "FRED", from = "2010-1-04", to =

oil_price <- tibble::rownames_to_column(oil_price, var = "Date")
```