task 4

John-William Johnsen

4/1/23

Task 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.

```
library(tidyverse)
library(lubridate)
library(quantmod)
library(janitor)
library(knitr)
```

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

```
table_1 <- data %>%
  mutate(order_date = as.Date(order_date)) %>%
  mutate(year = year(order_date), month = month(order_date), day = day(order_date)) %>%
  filter(year=="2017", month >= 10, customer_segment %in% c("Corporate", "Consumer"),regio
  group_by(region, month, customer_segment) %>%
  summarize(total_sales = sum(sales)) %>%
  rename("Region" = "region", "Month" = "month", "Customer segment" = "customer_segment",
  arrange(Region, Month)
```

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

```
kable(table_1)
```

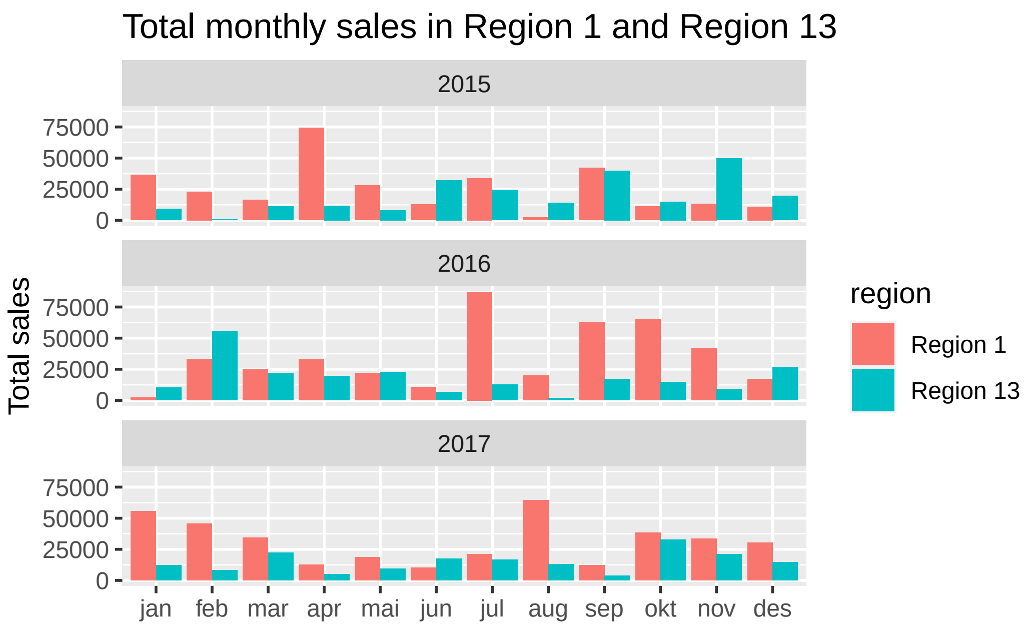
Region	Month	Customer segment	Total sales
Region 1	10	Consumer	815.790
Region 1	10	Corporate	295.060
Region 1	11	Consumer	9480.470
Region 1	11	Corporate	8564.840
Region 1	12	Consumer	210.060
Region 1	12	Corporate	13260.838
Region 9	10	Consumer	5908.890
Region 9	10	Corporate	16781.431
Region 9	11	Consumer	192.330
Region 9	11	Corporate	5463.130
Region 9	12	Corporate	9377.647

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

```r
Fig_1 <- data %>%
  mutate(order_date = as.Date(order_date)) %>%
  mutate(year = year(order_date),
         month = month(order_date, label = TRUE),
         day = day(order_date)) %>%
  filter(year %in% c("2015", "2016", "2017"),
         region %in% c("Region 1", "Region 13")) %>%
  group_by(region, month, year) %>%
  summarize(total_sales = sum(sales)) %>%
  mutate(date = make_date(year,month))
```

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

```r
Fig_1 %>%
  ggplot(aes(x=month, y=total_sales))+
  geom_col(aes(fill=region), position="dodge")+
  labs(x="", y="Total sales", title="Total monthly sales in Region 1 and Region 13", color
  facet_wrap(~year, nrow=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.

```
table_2 <- Fig_1 %>%
  pivot_wider(names_from = region, values_from = total_sales) %>%
  filter(`Region 13` > `Region 1`)

kable(table_2)
```

month	year	date	Region 1	Region 13
jan	2016	2016-01-01	2362.43	10407.92
feb	2016	2016-02-01	33085.29	55631.95
mai	2016	2016-05-01	22068.67	22821.70
jun	2015	2015-06-01	12844.97	32306.88
jun	2017	2017-06-01	10335.31	17430.46
aug	2015	2015-08-01	2266.96	13985.22
okt	2015	2015-10-01	11058.13	14885.24
nov	2015	2015-11-01	13290.26	49685.99
des	2015	2015-12-01	11048.17	19514.99
des	2016	2016-12-01	17020.27	26889.71

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.

```r
table_3 <- data %>%
  mutate(year = year(order_date),
         month = month(order_date),
         day = day(order_date)) %>%
  filter(year == 2017, !region %in% c("Region 3", "Region 5", "Region 8")) %>%
  group_by(customer_segment, product_category) %>%
  summarise(average_profit = mean(profit)) %>%
  arrange(desc(average_profit))
```

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

```r
kable(table_3)
```

customer_segment	product_category	average_profit
Small Business	Technology	544.442933
Corporate	Technology	413.915696
Home Office	Technology	271.119368
Consumer	Technology	223.123188
Corporate	Office Supplies	164.827620
Consumer	Office Supplies	107.459323
Small Business	Office Supplies	93.995230
Home Office	Office Supplies	71.364500
Consumer	Furniture	19.551912
Small Business	Furniture	16.874722
Home Office	Furniture	7.511408
Corporate	Furniture	-88.222121

Task. 2

In this task, feel free to use any API or package/library that downloads the data to your session. Use code and download daily stock prices for Exxon Mobil Corporation (XOM), traded at NYSE. The Yahoo! Finance site is a convenient place to find the data . Use the Adjusted closing price from January 4th 2010 as the starting date. And calculate the monthly average using trading volume as a weight, and save this variable as ”exxon”. Use code to download the daily Brent Crude Oil Price from FRED from January 4th 2010 as the starting date. And calculate the monthly arithmetic average. Save the monthly arithmetic average values as ”oil”. In both variables, take December 2022, or 2022:12 for shorthand as a last data point.

```
xom <- data.frame(getSymbols("XOM", src = "yahoo", auto.assign = FALSE)) %>%
  clean_names() %>%
  rownames_to_column(var="date") %>%
  rename("adj_close" = "xom_adjusted", "volume" = "xom_volume") %>%
  select(date, adj_close, volume) %>%
  mutate(date = as.Date(date)) %>%
  filter(date >= "2010-01-04" & date <= "2022-12-31") %>%
  mutate(year = year(date),
         month = month(date),
         day = day(date)) %>%
  group_by(year, month) %>%
  summarise(exxon = weighted.mean(adj_close, volume)) %>%
  mutate(date = make_date(year,month)) %>%
  ungroup() %>%
  select(date, exxon)
```

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

```
fredoil <- data.frame(getSymbols("DCOILBRENTEU", src = "FRED", auto.assign = FALSE))

fredoil <- fredoil %>%
  mutate(date = ymd(rownames(fredoil))) %>%
  rename(price = 1) %>%
  select(date, price) %>%
  filter(date >="2010-01-04",date<"2022-12-31")

fredoil <- fredoil %>%
  mutate(year = year(date),
         month = month(date),
         day = day(date)) %>%
  group_by(year,month) %>%
  summarise(oil= mean(price, na.rm=TRUE)) %>%
  mutate(date = make_date(year,month)) %>%
  as_tibble() %>%
  ungroup() %>%
  select(date, oil)
```

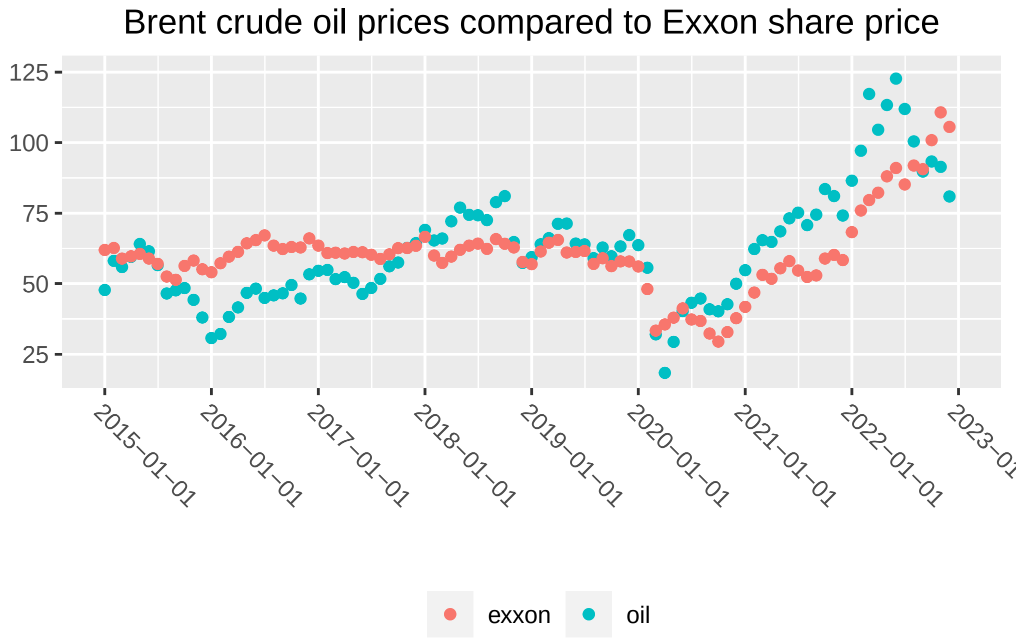
```
`summarise()` has grouped output by 'year'. You can override using the
`.groups` argument.
```

```
data_2 <- xom %>%
  cbind(oil = fredoil$oil)
```

Plot both variables, i.e., exxon and oil ” in the same plot window. Here, the x-axis should be the ”date” variable. Comment on the plots.

```
fig1 <- ggplot(data_2, aes(x=date)) +
  geom_point(aes(y = oil, color = "oil")) +
  geom_point(aes(y = exxon, color = "exxon")) +
  ggtitle("Brent crude oil prices compared to Exxon share price") +
  xlab("") +
  ylab("") +
  scale_x_date(limits = c(as.Date("2015-01-01"), as.Date("2022-12-31")), date_breaks = "1
  theme(plot.title = element_text(hjust = 0.5),
        legend.title = element_blank(),
        legend.position = "bottom",
        axis.text.x = element_text(angle = -45, hjust = 0))
fig1
```

```
Warning: Removed 60 rows containing missing values (`geom_point()`).
Removed 60 rows containing missing values (`geom_point()`).
```



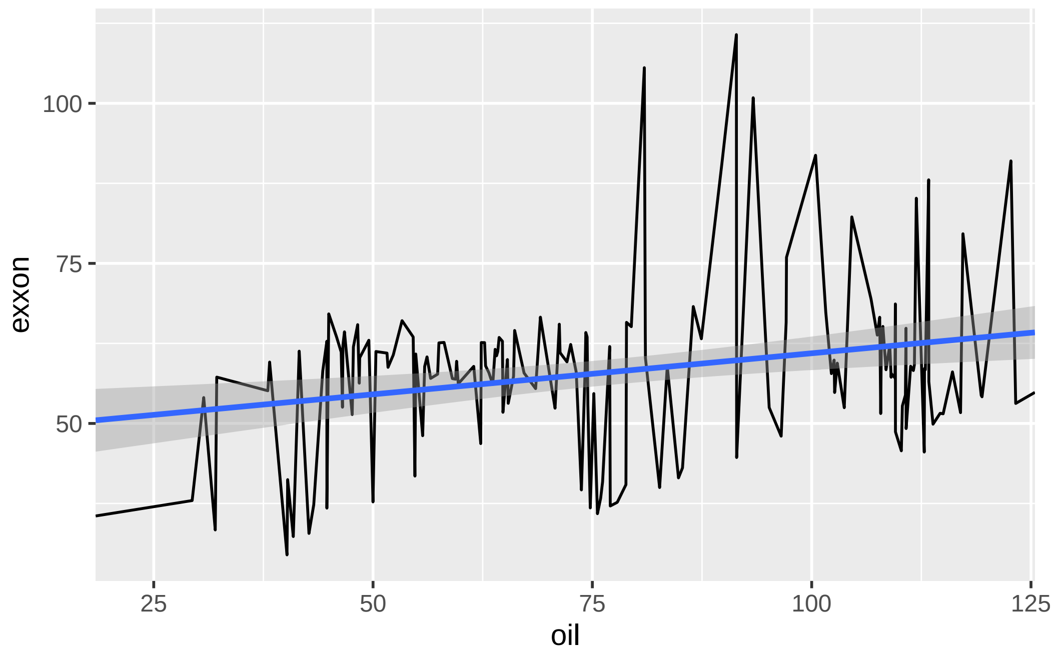
Now take ”exxon” as a y-variable and ”oil” as an x-variable.

Use R's lm() function. Set the variable on the y-axis and x-axis, and specify the data set.

LM(<Y variable name> ~ <X variable>, data=<dataset name>)

```
data_2 %>%
  ggplot(aes(x=oil, y=exxon)) +
  geom_line()+
  scale_x_continuous(expand = c(0,0)) +
  geom_smooth(method=lm)
```

```
`geom_smooth()` using formula = 'y ~ x'
```



After “running” the code, how do you interpret the estimated coefficients?

```
lm(exxon ~ oil, data = data_2)
```

```
Call:
lm(formula = exxon ~ oil, data = data_2)

Coefficients:
(Intercept)          oil
    48.1373       0.1282
```