

Accounting Project

Study Group 6

10/03/2022

```
library(tidyverse)
library(here)
library(janitor)
library(lubridate)
library(patchwork)
```

Load data

```
# import data
overheads <- readxl::read_excel(here("Data", "Overheads.xlsx"))
transactions <- read.delim(here("Data", "Transactions.txt"))
```

Data Cleaning

```
unique(transactions$region)
```

```
## [1] "Midwest"      "Northwest"    "West"         "Northeast"
## [5] "East coast"   "Central"      "South"        "International"
## [9] "Centrall"     "Soouth"
```

```
transactions[transactions$region == "Centrall", "region"] <-"Central"
transactions[transactions$region == "Soouth", "region"] <-"South"
```

```
# clean data
transactions$quantity_sold <- as.numeric(transactions$quantity_sold)
transactions$date_of_sale <- lubridate::as_date(transactions$date_of_sale, format = '%d/%m/%Y')
transactions <- transactions %>%
  mutate(across(c(list_price, cost), ~gsub("\\$", "", .) %>% as.numeric))
head(transactions)
```

```
##   customer_number   region date_of_sale   item      brand collection
## 1         20943   Midwest  2015-01-01  918DP Jeffrey Alexander  Prestige
## 2         126101 Northwest  2015-01-01  2981AB      Elements  Florence
## 3         161675     West   2015-01-01  910-128PC Jeffrey Alexander  Modena
## 4         175749     West   2015-01-01  351-128PC      Elements  Calloway
## 5         216582     West   2015-01-01  S271-3PB      Elements  Torino
## 6         272896 Northeast  2015-01-01  293-160PC Jeffrey Alexander    Zane
##   description list_price  cost quantity_sold
## 1      Knob      14.14  8.62           434
## 2    3" pull       6.83  4.27           54
## 3 128 mm CC pull     17.68 11.08          450
```

```
## 4 128" CC pull      7.63  4.85      467
## 5   3" CC pull      2.52  1.60      380
## 6 160 mm CC pull    17.15 10.77      689
```

```
overheads<-rename(overheads, category = `For the year 2018`)
```

Question 1

```
# calculate overhead costs in 2018
overheads %>% pivot_longer(2:9,names_to = "region", values_to = "overheads") %>%
  pivot_wider(names_from = "category", values_from = "overheads") %>% #separate production and non-prod
  clean_names() %>%
  summarise(production_overheads = sum(production_overheads),
            non_production_overheads = sum(non_production_overheads)) #summing up
```

```
## # A tibble: 1 x 2
##   production_overheads non_production_overheads
##           <dbl>             <dbl>
## 1         48205754             9580000
```

```
# calculate sales revenue and costs
transactions <-transactions %>%
  mutate(sales_revenue = list_price*quantity_sold,
         costs = cost*quantity_sold,
         year = year(date_of_sale))

# calculate operating profits and percentage in sales
transactions %>% filter(year == 2018) %>%
  summarise(sales_revenue = sum(sales_revenue),
            costs = sum(costs)) %>%
  mutate(gross_profits = sales_revenue-costs-48205754) %>%
  mutate(operating_profit = gross_profits-9580000) %>% # subtract non-production overhead costs
  mutate(profit_as_percentage_of_sales = operating_profit/sales_revenue)
```

```
##   sales_revenue    costs gross_profits operating_profit
## 1  154984988 94618794    12160440      2580440
##   profit_as_percentage_of_sales
## 1                0.01664961
```

Question 2

By calculating and plotting the operating profits of each region, we found that the Eastcoast region has the highest operating profits, as shown by the bar chart in figure 1. The West and South regions are suffering negative operating profits. By looking at figure 2, we see that the West region has a positive gross profits, which suggests that its non-production overheads must be higher than gross profits so that it brings the operating profits to a negative number.

By breaking down the costs, we can see that the non-production overhead costs are relatively low when compared with the other cost categories, as shown by figure 2. The largest of costs still come from product direct and variable costs, followed by production overheads.

```
# calculate overhead costs in 2018
overheads_by_region <- overheads %>%
  pivot_longer(2:9,names_to = "region", values_to = "overheads") %>%
  pivot_wider(names_from = "category", values_from = "overheads") %>% #separate production and non-prod
```

```

clean_names()
overheads_by_region

## # A tibble: 8 x 3
##   region      production_overheads non_production_overheads
##   <chr>          <dbl>          <dbl>
## 1 Central          5278312          1219464
## 2 East coast       7513545          1589821
## 3 International    690587           217290
## 4 Midwest          9874330          2018055
## 5 Northeast        3965187           814835
## 6 Northwest        4968774          1075209
## 7 South            6849541           980076
## 8 West             9065478          1665250

transactions_by_region <- transactions %>%
  filter(year == 2018) %>%
  group_by(region) %>%
  summarise(sales_revenue = sum(sales_revenue),
            costs = sum(costs))
transactions_by_region

## # A tibble: 8 x 3
##   region      sales_revenue      costs
##   <chr>          <dbl>      <dbl>
## 1 Central      19728444.  12526495
## 2 East coast   25720078.  15615559.
## 3 International 3515301.   2139543.
## 4 Midwest      32648041.  19821850.
## 5 Northeast    13182393.   8020092.
## 6 Northwest    17394723.  10510174.
## 7 South        15855649.   9661517.
## 8 West         26940358.  16323564.

joined_by_region <- full_join(overheads_by_region, transactions_by_region, by = "region")
joined_by_region <- joined_by_region %>%
  mutate(gross_profits = sales_revenue-costs-production_overheads) %>%
  mutate(operating_profits = sales_revenue-costs-production_overheads-non_production_overheads)
head(joined_by_region)

## # A tibble: 6 x 7
##   region production_overh~ non_production_o~ sales_revenue costs gross_profits
##   <chr>          <dbl>          <dbl>          <dbl> <dbl>      <dbl>
## 1 Central          5278312          1219464      19728444.  1.25e7    1923638.
## 2 East c~         7513545          1589821      25720078.  1.56e7    2590974.
## 3 Intern~         690587           217290       3515301.  2.14e6     685172.
## 4 Midwest          9874330          2018055      32648041.  1.98e7    2951861.
## 5 Northe~         3965187           814835      13182393.  8.02e6    1197114.
## 6 Northw~         4968774          1075209      17394723.  1.05e7    1915774.
## # ... with 1 more variable: operating_profits <dbl>

```

Figure 1- Operating Profits by Region

```

(joined_by_region %>%
  ggplot(aes(x = fct_reorder(region, -operating_profits), y = operating_profits)) +

```

```
geom_col(fill = "darkred") +
labs(x = "", y = "Absolute Operating Profits") +
theme_minimal()) / (joined_by_region %>%
ggplot(aes(x = fct_reorder(region, -operating_profits), y = operating_profits/sales_revenue)) +
geom_col(fill = "darkblue") +
scale_y_continuous(labels = scales::percent) +
labs(x = "", y = "% Operating Profits") +
theme_minimal())
```

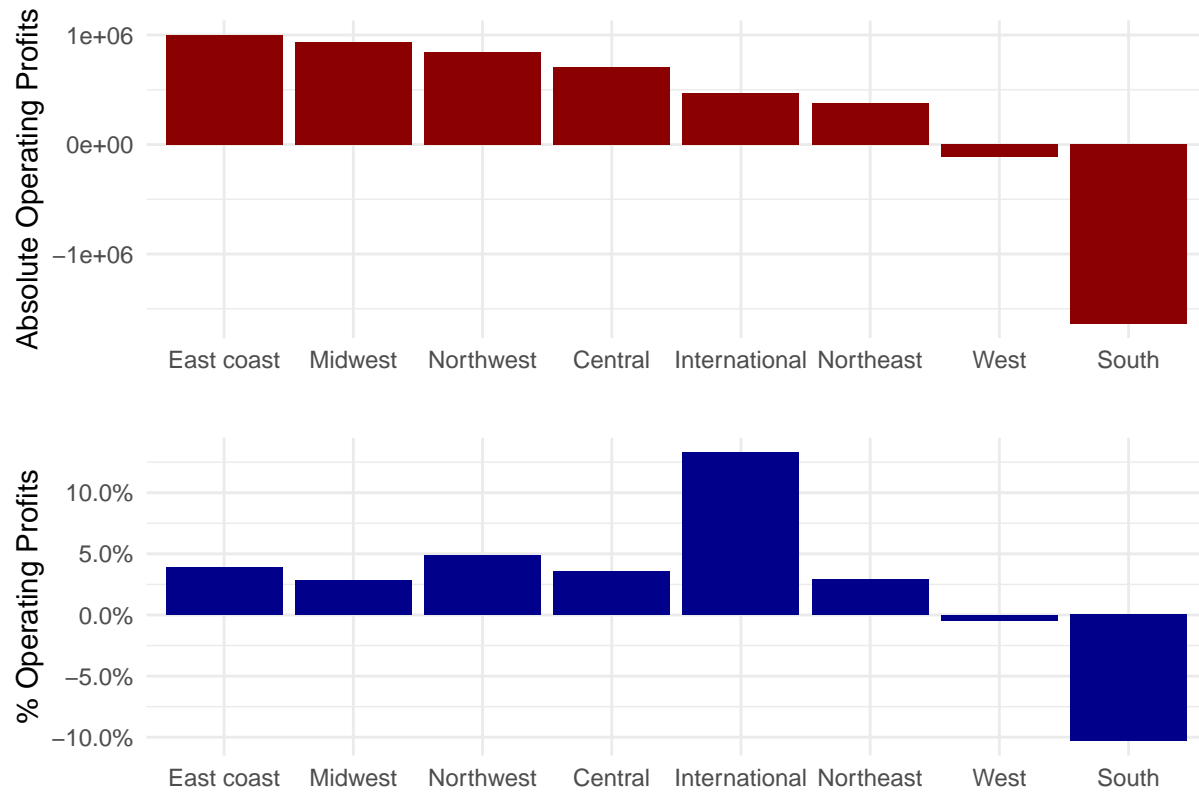


Figure 2- Gross Profits by Region

```
(joined_by_region %>%
ggplot(aes(x = fct_reorder(region, -gross_profits), y = gross_profits)) +
geom_col(fill = "darkred") +
labs(x = "", y = "Absolute Gross Profits") +
theme_minimal()) / (joined_by_region %>%
ggplot(aes(x = fct_reorder(region, -gross_profits), y = gross_profits/sales_revenue)) +
geom_col(fill = "darkblue") +
scale_y_continuous(labels = scales::percent) +
labs(x = "", y = "% Gross Profits") +
theme_minimal())
```

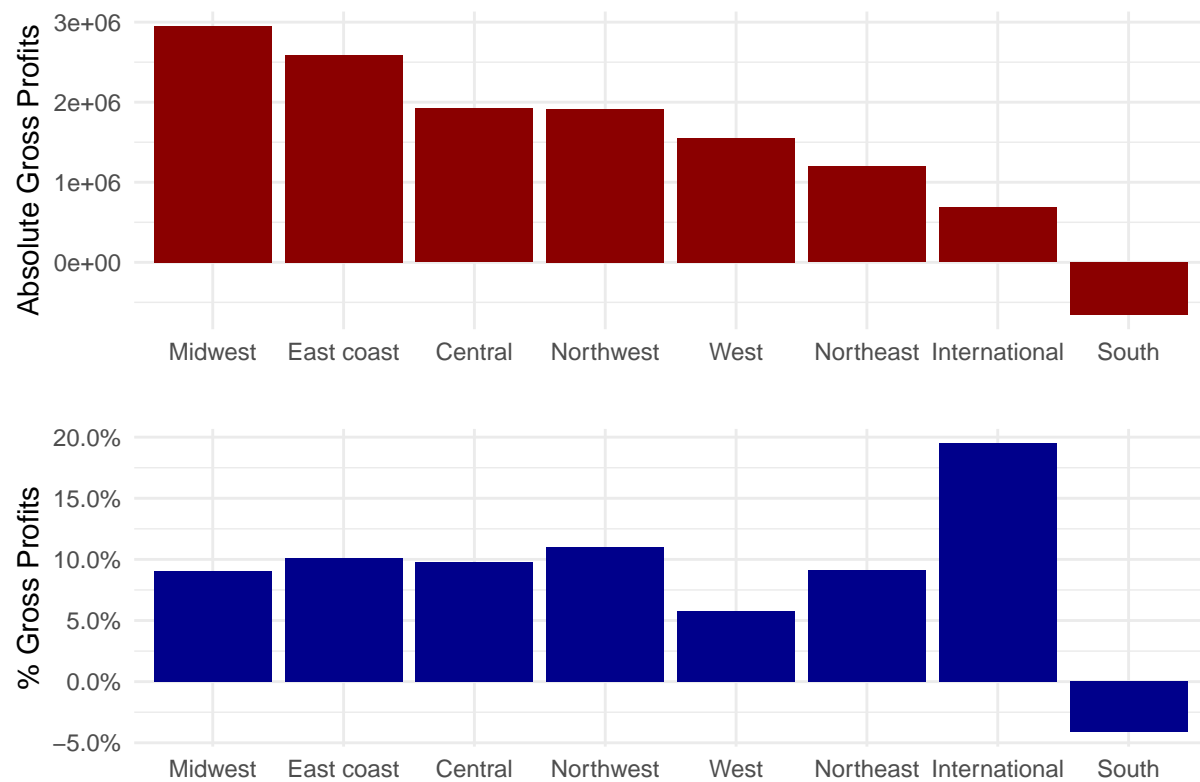


Figure 3- Cost Breakdown by Region

```
joined_by_region %>% pivot_longer(c(2,3,5), names_to = "components", values_to = "value") %>%
ggplot(aes(x = fct_rev(fct_reorder(region, sales_revenue)), y = value, fill = components)) +
  geom_bar(stat = "identity") +
  labs(x = "Region", y = "Components")+
  theme_minimal() +
  scale_fill_brewer(palette = "Blues")
```

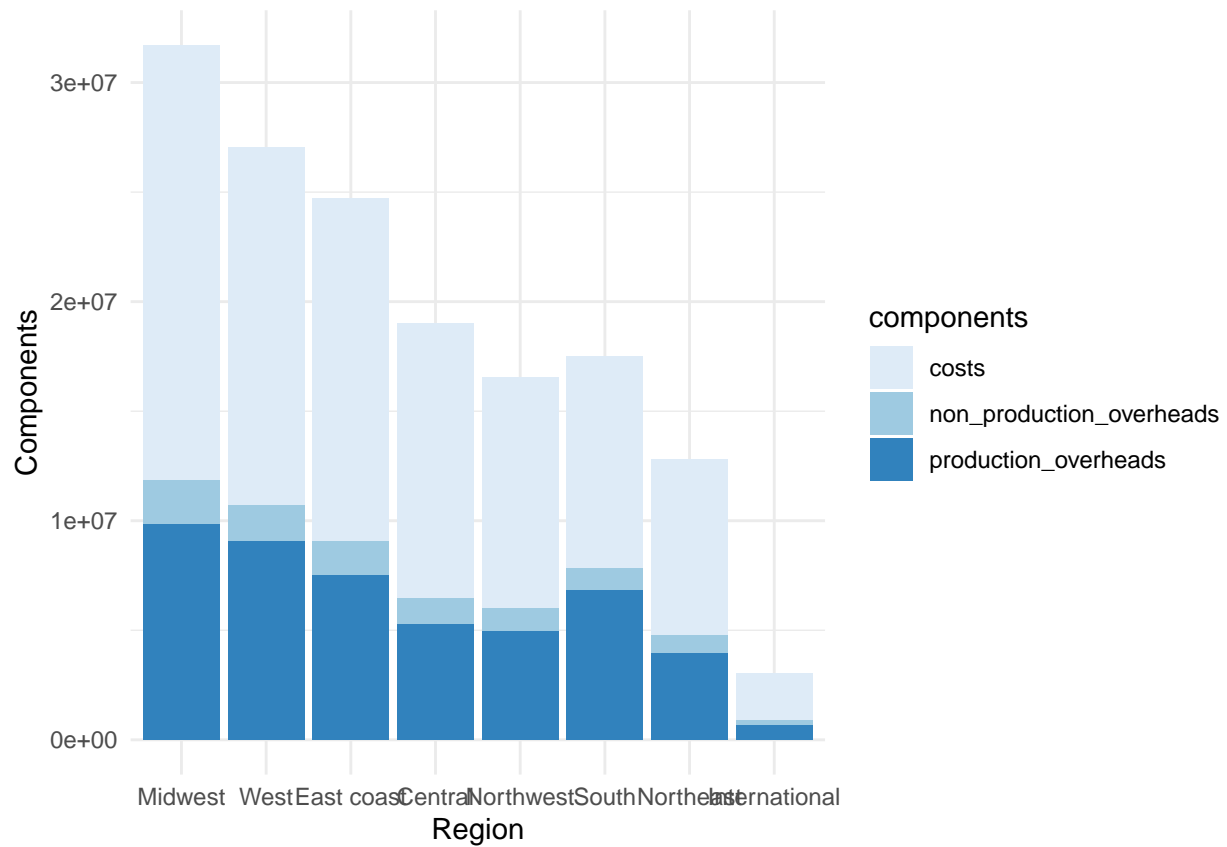
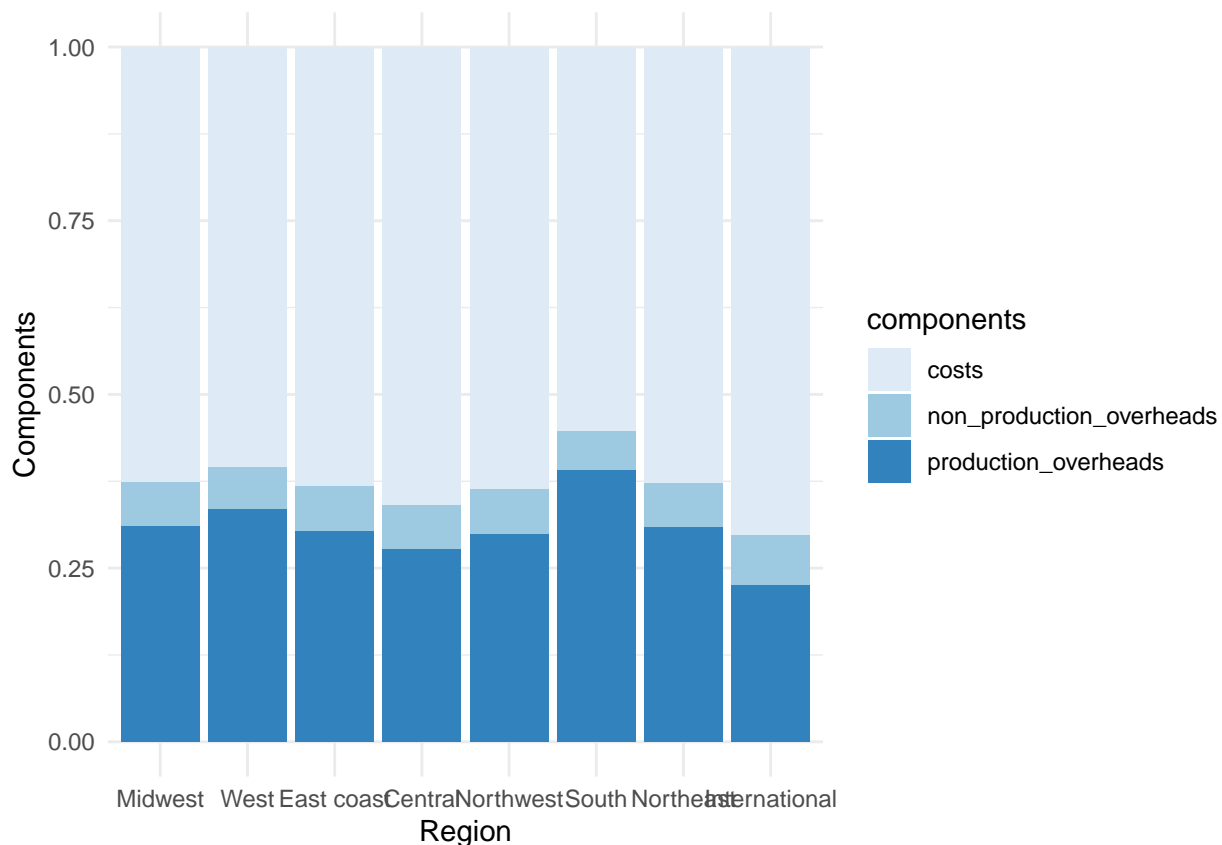


Figure 4- Cost Breakdown in Percentage by Region

```
joined_by_region %>% pivot_longer(c(2,3,5), names_to = "components", values_to = "value") %>%
ggplot(aes(x = fct_rev(fct_reorder(region, sales_revenue)), y = value, fill = components)) +
  geom_bar(stat = "identity", position = "fill") +
  labs(x = "Region", y = "Components") +
  theme_minimal() +
  scale_fill_brewer(palette = "Blues")
```



Question 3

As shown in figure 1 and 2, both sales revenue and contribution margin are higher for Jefferey Alexander brand. The difference in sales revenue is very large, which means that Jefferey Alexander is much more competitive in market than Elements. From figure 3, we see that for different collections, there is a positive relationship between sales revenue and contribution margin, which holds true for both brands.

From figure 4, we see that Belcastel contributed the most sales revenue and contribution margin to the Jefferey Alexander brand, while Madison contributed the most to the Elements brand.

```
joined_by_region <- joined_by_region %>%
  mutate(pro_overhead_per_sales = production_overheads/sales_revenue)

transactions_by_brand <- transactions %>%
  left_join(joined_by_region %>% select(region, pro_overhead_per_sales), by = "region") %>%
  mutate(production_overheads = pro_overhead_per_sales*sales_revenue,
         total_variable_cost = production_overheads+costs) %>%
  group_by(brand) %>%
  summarize(sales_revenue = sum(sales_revenue),
            total_variable_cost = sum(total_variable_cost)) %>%
  mutate(contribution_margin_perc = 1- (total_variable_cost/sales_revenue),
         contribution_margin = sales_revenue-total_variable_cost)
transactions_by_brand
```

```
## # A tibble: 2 x 5
##   brand      sales_revenue total_variable_c~ contribution_margi~ contribution_ma~
##   <chr>          <dbl>          <dbl>          <dbl>          <dbl>
```

```
## 1 Elements      81739400.      77737786.      0.0490      4001614.
## 2 Jeffrey ~    403337381.      370438371.      0.0816      32899010.

transactions_by_collection <- transactions %>%
  left_join(joined_by_region %>% select(region, pro_overhead_per_sales), by = "region") %>%
  mutate(production_overheads = pro_overhead_per_sales*sales_revenue,
         total_variable_cost = production_overheads+costs) %>%
  group_by(collection) %>%
  summarize(sales_revenue = sum(sales_revenue),
            total_variable_cost = sum(total_variable_cost)) %>%
  mutate(contribution_margin_perc = 1- (total_variable_cost/sales_revenue),
         contribution_margin = sales_revenue-total_variable_cost)
head(transactions_by_collection)
```

```
## # A tibble: 6 x 5
##   collection sales_revenue total_variable_cost contribution_ma~ contribution_ma~
##   <chr>      <dbl>          <dbl>          <dbl>          <dbl>
## 1 Aberdeen    4959652.      4550959.      0.0824      408693.
## 2 Aiden        330647.      1074964.     -2.25      -744317.
## 3 Alvar        2117913.      1916725.      0.0950      201188.
## 4 Amsden       2502762.      2298731.      0.0815      204031.
## 5 Annadale     4869955.      4528855.      0.0700      341100.
## 6 Anwick       6022229.      5634367.      0.0644      387863.

transactions_by_brand_collection <- transactions %>%
  left_join(joined_by_region %>% select(region, pro_overhead_per_sales), by = "region") %>%
  mutate(production_overheads = pro_overhead_per_sales*sales_revenue,
         total_variable_cost = production_overheads+costs) %>%
  group_by(brand,collection) %>%
  summarize(sales_revenue = sum(sales_revenue),
            total_variable_cost = sum(total_variable_cost)) %>%
  mutate(contribution_margin_perc = 1- (total_variable_cost/sales_revenue),
         contribution_margin = sales_revenue-total_variable_cost)
```

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

head(transactions_by_brand_collection)
```

```
## # A tibble: 6 x 6
## # Groups:   brand [1]
##   brand      collection sales_revenue total_variable_cost contribution_margin_perc
##   <chr>      <chr>          <dbl>          <dbl>          <dbl>
## 1 Elements Aiden        330647.      1074964.     -2.25
## 2 Elements Arcadia    1219833.      1160922.      0.0483
## 3 Elements Asher      3016359.      2859747.      0.0519
## 4 Elements Belfast   2407136.      2313362.      0.0390
## 5 Elements Brenton    3784927.      3547444.      0.0627
## 6 Elements Calloway   2989323.      2809301.      0.0602
## # ... with 1 more variable: contribution_margin <dbl>
```

Figure 1

```
transactions_by_brand %>% ggplot(aes(x = contribution_margin/1000000, y = brand, fill = brand)) +
  geom_col() +
  geom_text(aes(label = round(contribution_margin/1000000)), hjust = 1.5, color = '#ffffff') +
```



```
scale_fill_manual(values = c("darkred", "darkblue"))+
labs(x = "Contribution Margin (Millions $)", y = "Brand") +
theme_minimal()
```

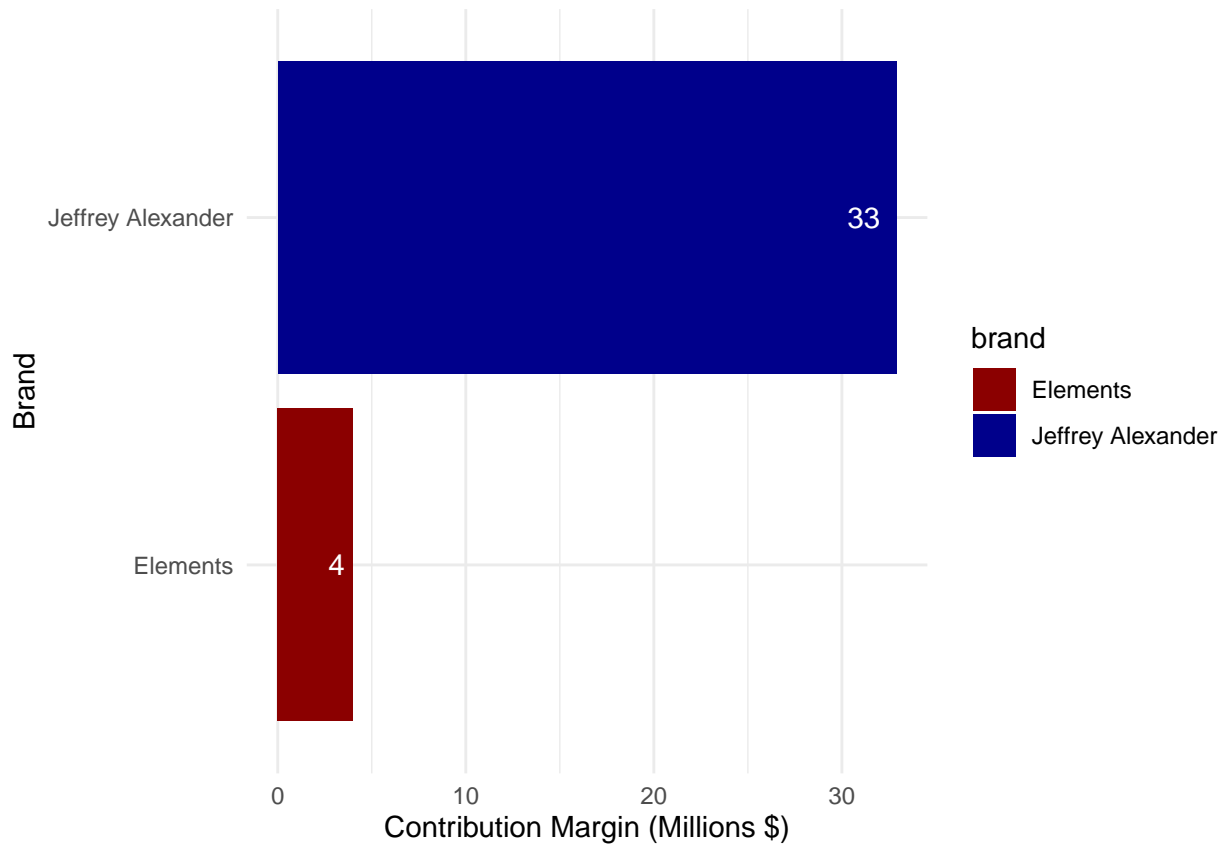


Figure 2

```
transactions_by_brand %>% ggplot(aes(x = sales_revenue/1000000, y = brand, fill = brand)) +
  geom_col() +
  geom_text(aes(label = round(sales_revenue/1000000)), hjust = 1.5, color = '#ffffff') +
  scale_fill_manual(values = c("darkred", "darkblue"))+
  labs(x = "Sales Revenue (Million $)", y = "Brand") +
  theme_minimal()
```

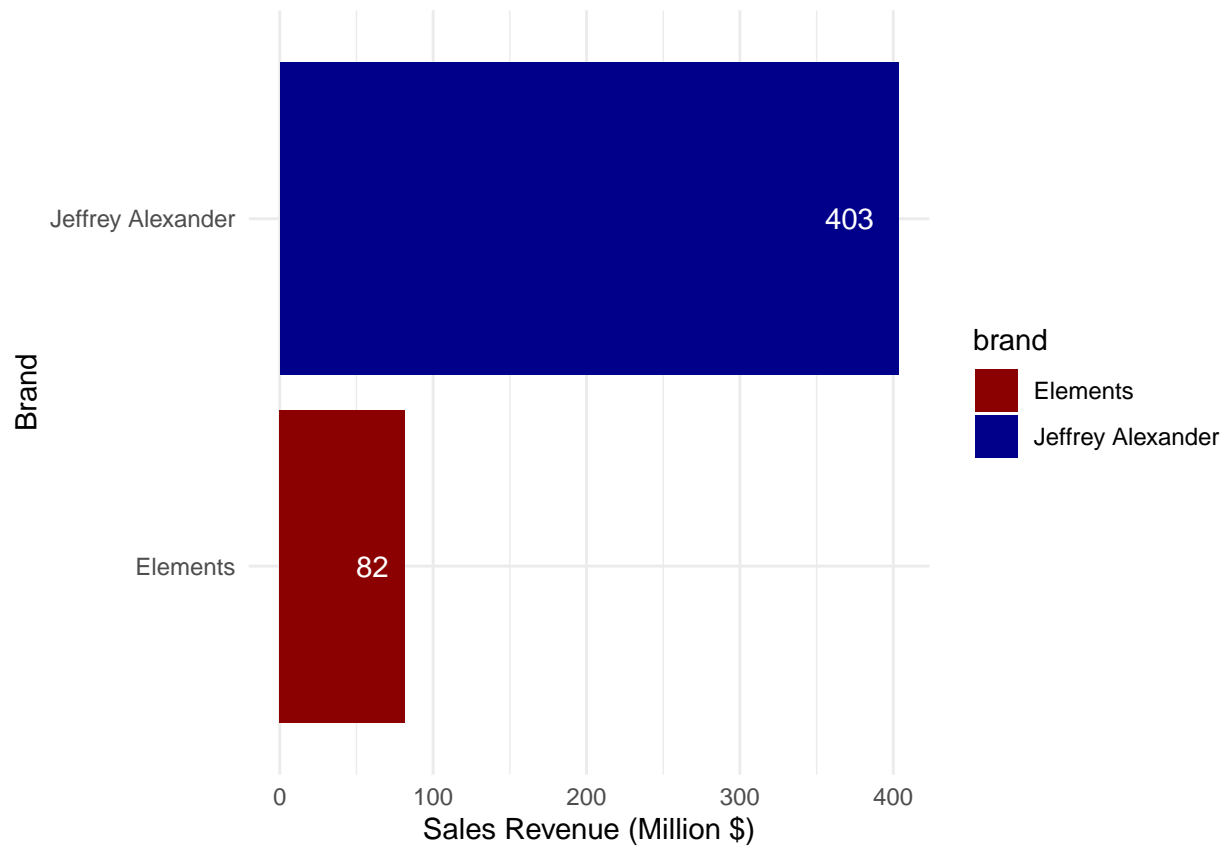
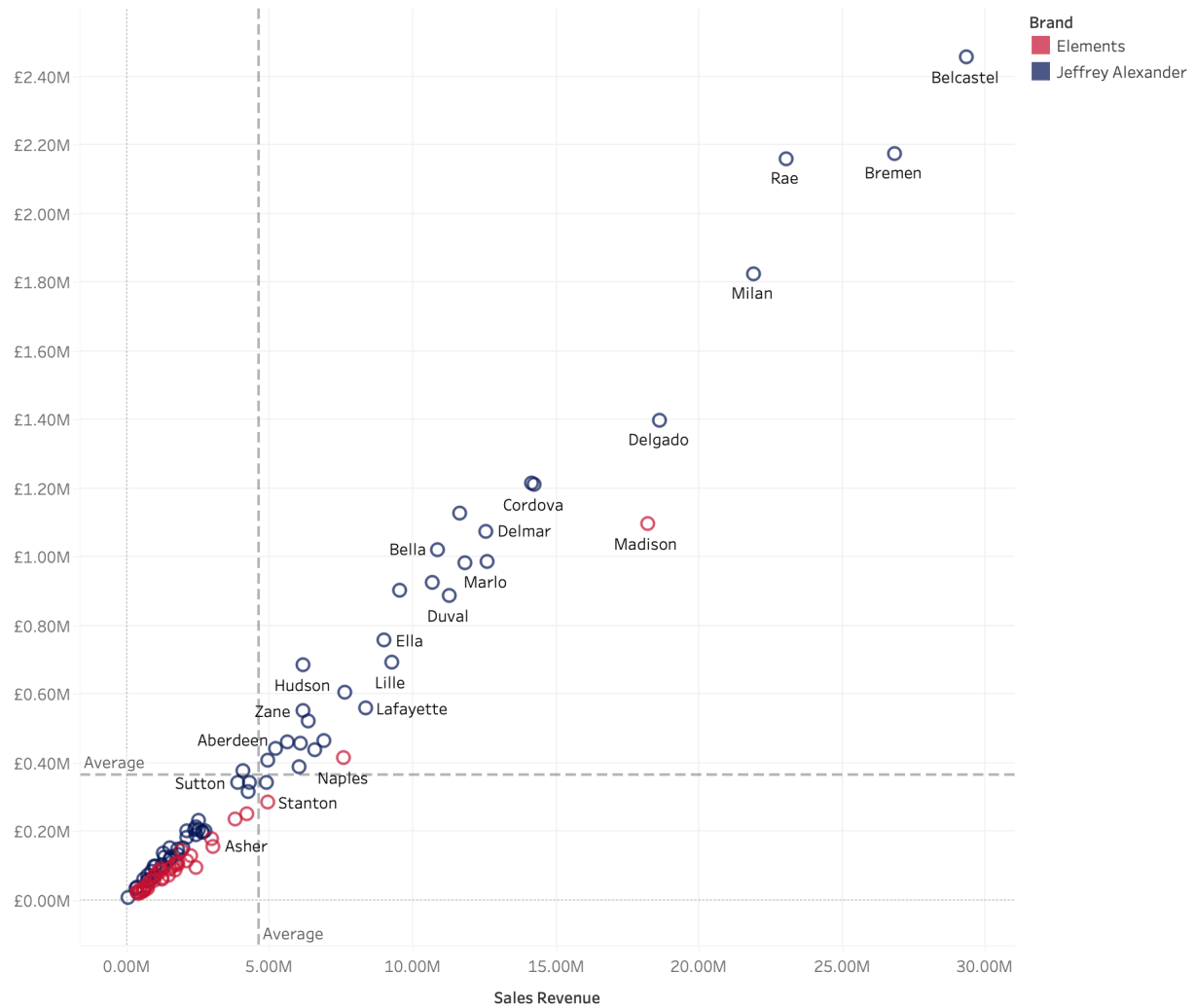


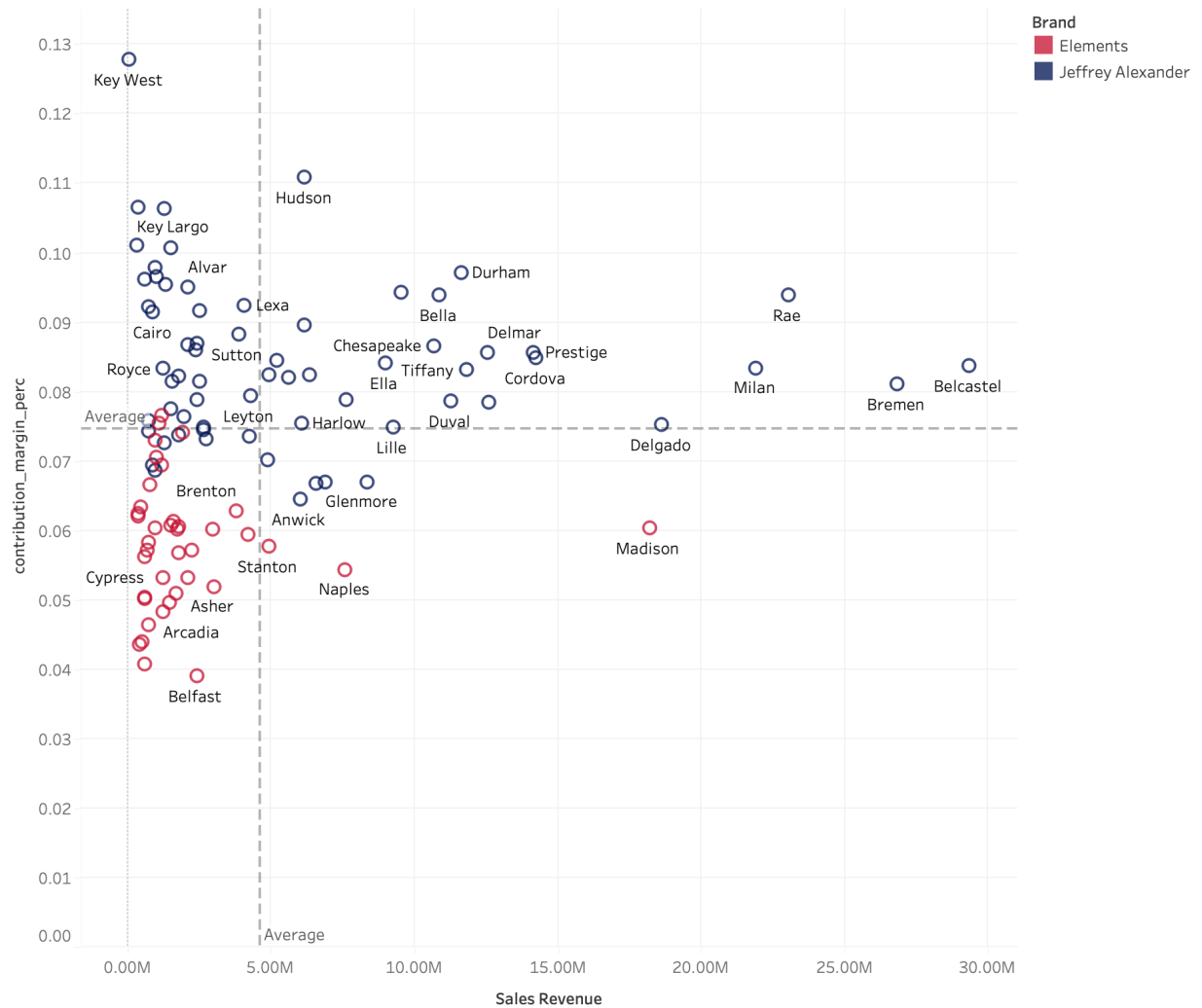
Figure 3

There is Positive Linear Relationship Between Sales and Contribution Margin

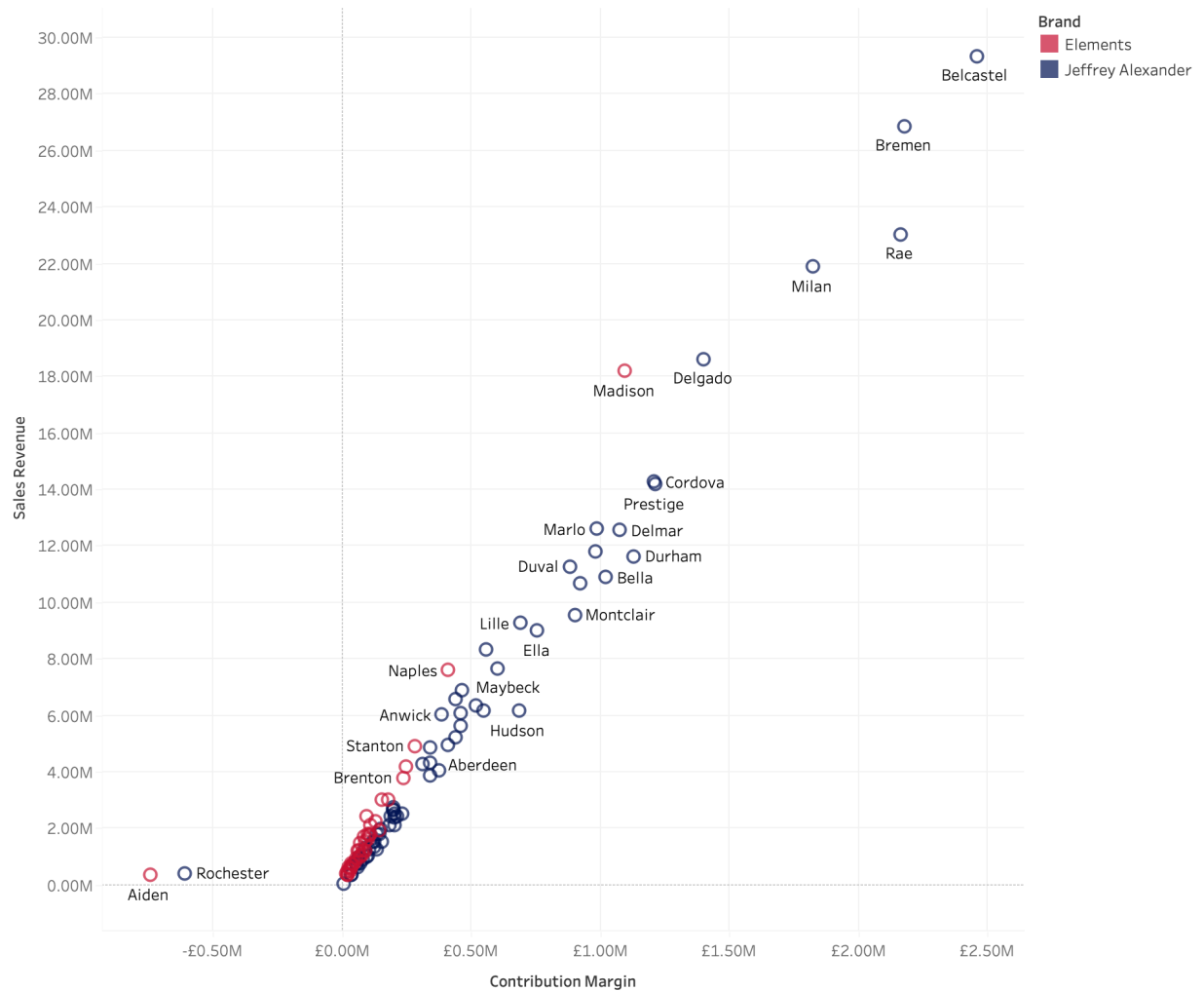
Contribution Margin and Sales Revenue by Brand and Collection



Contribution Margin % and Sales Revenue by Brand and Collection



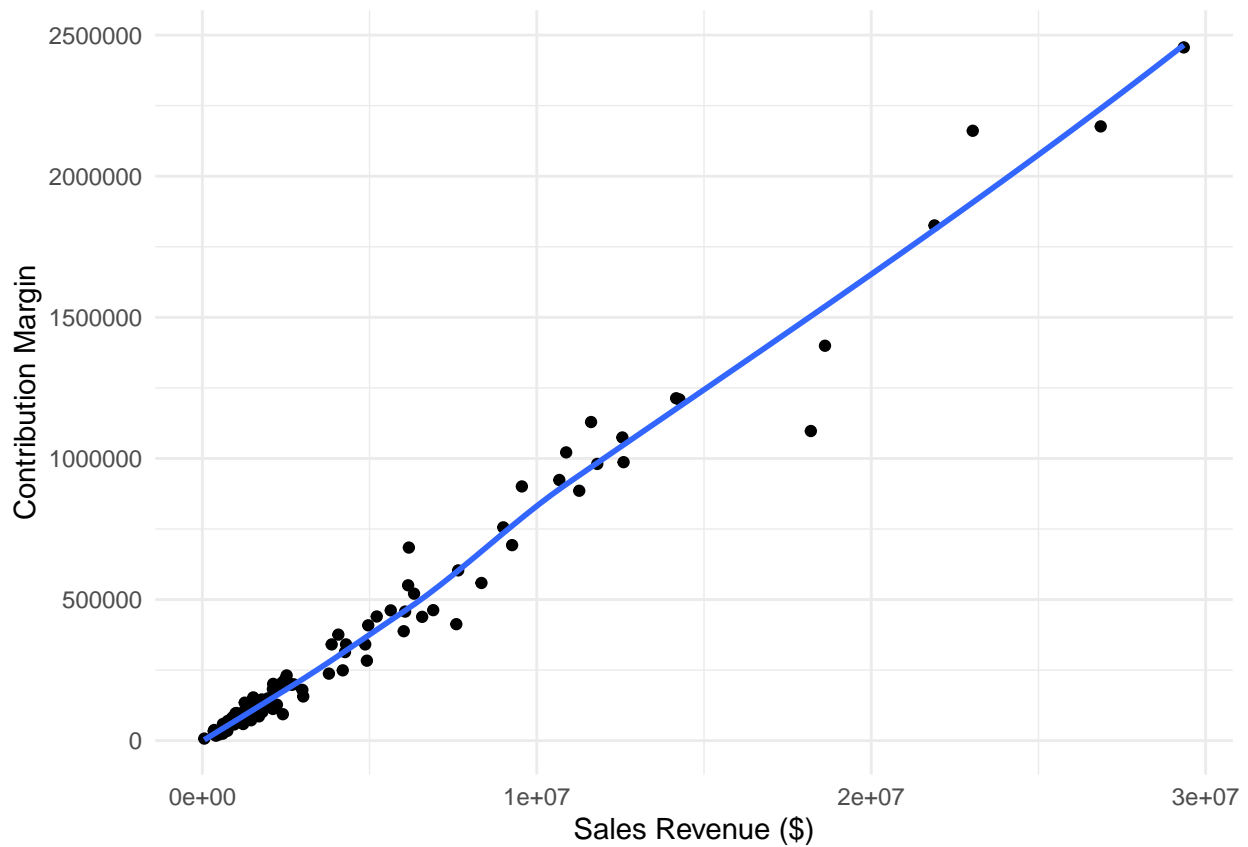
Outliers with Negative Contribution Margin



```
# transactions_by_collection %>% ggplot(aes(x = sales_revenue, y = contribution_margin)) +
#   geom_point()
```

```
transactions_by_brand_collection %>% filter(contribution_margin > 0) %>% ggplot(aes(x = sales_revenue, y = contribution_margin)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  scale_fill_manual(values = c("darkred", "darkblue")) +
  labs(x = "Sales Revenue ($)", y = "Contribution Margin") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
transactions_by_brand_collection %>% filter(contribution_margin > 0) %>% ggplot(aes(x = sales_revenue, y = contribution_margin)) +  
  geom_point() +  
  geom_smooth(se = FALSE) +  
  facet_wrap(~brand) +  
  labs(x = "Sales Revenue ($)", y = "Contribution Margin") +  
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

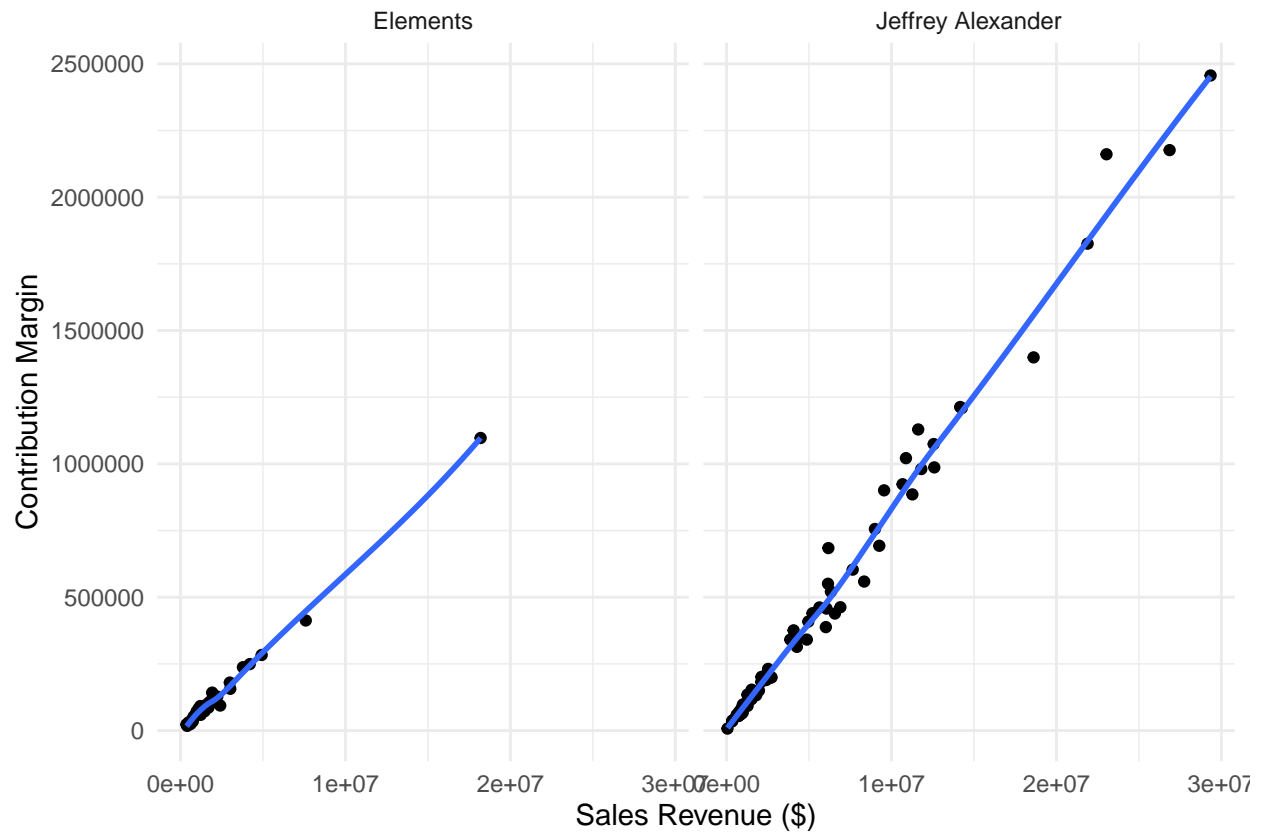
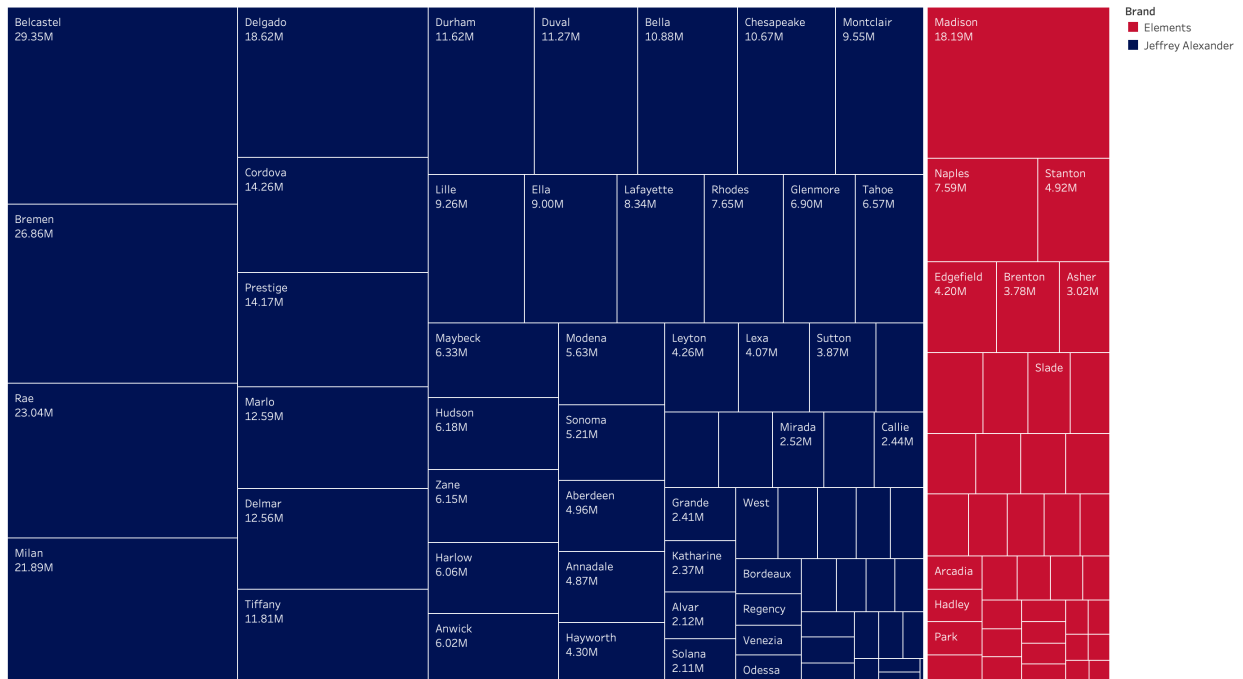
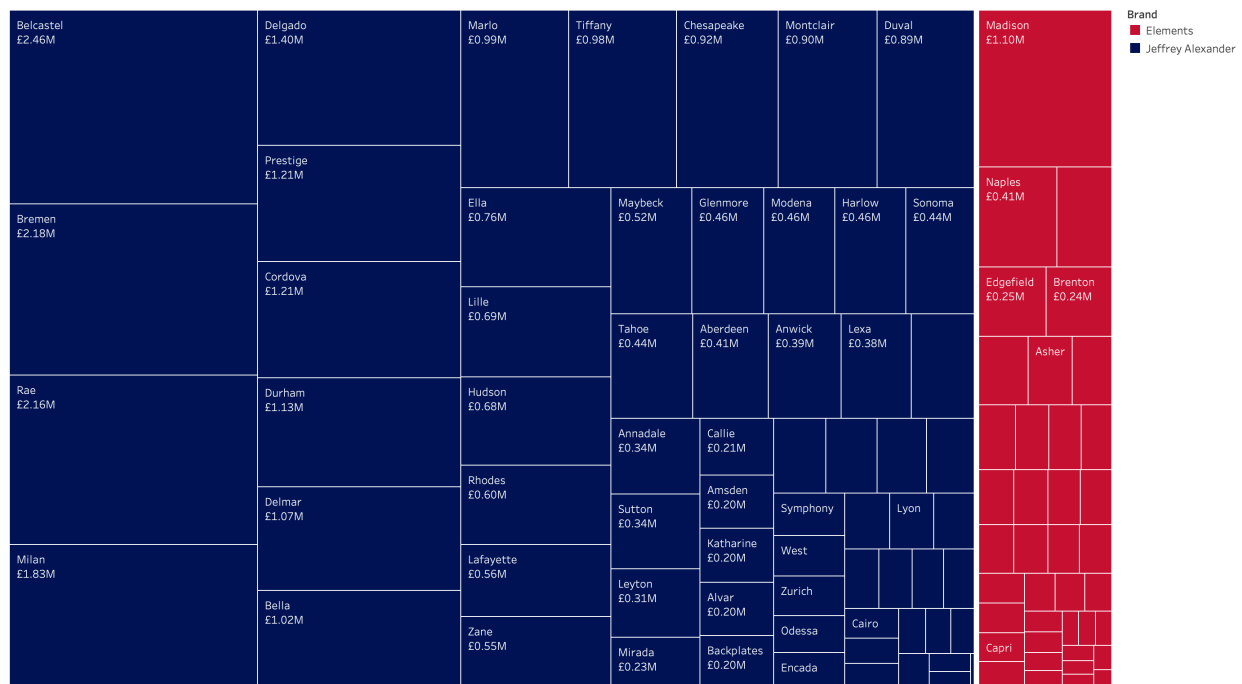


Figure 4

Sales Revenue by Brand and Collection



Contribution Margin by Brand and Collection



Question 4

We choose to find the breakeven sales and volume by chronological order of transactions.

Overall

```
transactions_variableCost <- transactions %>%
  filter(year == 2018) %>%
  arrange(date_of_sale) %>%
  left_join(joined_by_region %>% select(region, pro_overhead_per_sales), by = "region") %>%
  mutate(total_variable_cost = costs + pro_overhead_per_sales*sales_revenue)

break_even <- data.frame(total_cost = cumsum(transactions_variableCost$total_variable_cost) + sum(joined_by_region$pro_overhead_per_sales),
  sales_revenue = cumsum(transactions_variableCost$sales_revenue),
  volume = cumsum(transactions_variableCost$quantity_sold),
  date = transactions_variableCost$date_of_sale)

break_even[which(break_even$total_cost <= break_even$sales_revenue)[1],]
```

```
##      total_cost sales_revenue  volume      date
## 12361  122790564    122793343 6750493 2018-10-21
```

Overall, the company manage to breakeven at 2018-10-21 with a breakeven sales revenue of 122.79 million and a breakeven sales volume of 6.75 million.

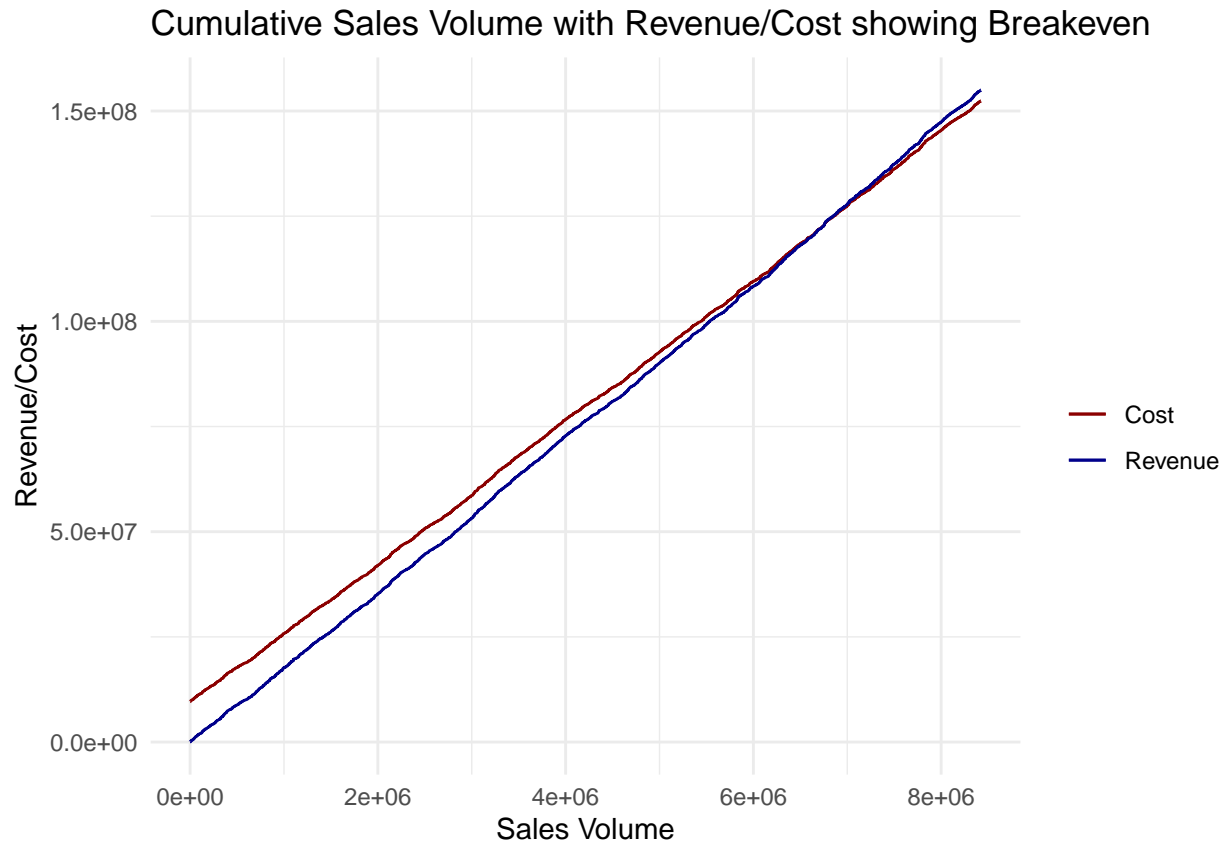
```
ggplot(break_even, aes(x = volume)) +
  geom_line(aes(y = total_cost, color = "Cost")) +
  geom_line(aes(y = sales_revenue, color = "Revenue")) +
  scale_colour_manual("",
    breaks = c("Cost", "Revenue"),
```



```

        values = c("darkred", "darkblue")) +
theme_minimal() +
labs(title = "Cumulative Sales Volume with Revenue/Cost showing Breakeven",
     x = "Sales Volume",
     y = "Revenue/Cost")

```



Regional

```
unique(transactions_variableCost$region)
```

```
## [1] "Northeast"    "East coast"   "Midwest"      "Central"
## [5] "West"         "South"        "Northwest"    "International"
```

```

transactions_variableCost_Northeast <- transactions_variableCost %>% filter(region == "Northeast")
break_even_Northeast <- data.frame(total_cost = cumsum(transactions_variableCost_Northeast$total_variable_cost),
                                   sales_revenue = cumsum(transactions_variableCost_Northeast$sales_revenue),
                                   volume = cumsum(transactions_variableCost_Northeast$quantity_sold),
                                   date = transactions_variableCost_Northeast$date_of_sale)

```

```

transactions_variableCost_Eastcoast <- transactions_variableCost %>% filter(region == "East coast")
break_even_Eastcoast <- data.frame(total_cost = cumsum(transactions_variableCost_Eastcoast$total_variable_cost),
                                   sales_revenue = cumsum(transactions_variableCost_Eastcoast$sales_revenue),
                                   volume = cumsum(transactions_variableCost_Eastcoast$quantity_sold),
                                   date = transactions_variableCost_Eastcoast$date_of_sale)

```

```

transactions_variableCost_Midwest <- transactions_variableCost %>% filter(region == "Midwest")
break_even_Midwest <- data.frame(total_cost = cumsum(transactions_variableCost_Midwest$total_variable_cost),
                                   sales_revenue = cumsum(transactions_variableCost_Midwest$sales_revenue),
                                   volume = cumsum(transactions_variableCost_Midwest$quantity_sold),
                                   date = transactions_variableCost_Midwest$date_of_sale)

```

```

        sales_revenue = cumsum(transactions_variableCost_Midwest$sales_revenue),
        volume = cumsum(transactions_variableCost_Midwest$quantity_sold),
        date = transactions_variableCost_Midwest$date_of_sale)

transactions_variableCost_Central <- transactions_variableCost %>% filter(region == "Central")
break_even_Central <- data.frame(total_cost = cumsum(transactions_variableCost_Central$total_variable_cost),
        sales_revenue = cumsum(transactions_variableCost_Central$sales_revenue),
        volume = cumsum(transactions_variableCost_Central$quantity_sold),
        date = transactions_variableCost_Central$date_of_sale)

transactions_variableCost_West <- transactions_variableCost %>% filter(region == "West")
break_even_West <- data.frame(total_cost = cumsum(transactions_variableCost_West$total_variable_cost) +
        sales_revenue = cumsum(transactions_variableCost_West$sales_revenue),
        volume = cumsum(transactions_variableCost_West$quantity_sold),
        date = transactions_variableCost_West$date_of_sale)

transactions_variableCost_South <- transactions_variableCost %>% filter(region == "South")
break_even_South <- data.frame(total_cost = cumsum(transactions_variableCost_South$total_variable_cost),
        sales_revenue = cumsum(transactions_variableCost_South$sales_revenue),
        volume = cumsum(transactions_variableCost_South$quantity_sold),
        date = transactions_variableCost_South$date_of_sale)

transactions_variableCost_Northwest <- transactions_variableCost %>% filter(region == "Northwest")
break_even_Northwest <- data.frame(total_cost = cumsum(transactions_variableCost_Northwest$total_variable_cost),
        sales_revenue = cumsum(transactions_variableCost_Northwest$sales_revenue),
        volume = cumsum(transactions_variableCost_Northwest$quantity_sold),
        date = transactions_variableCost_Northwest$date_of_sale)

transactions_variableCost_International <- transactions_variableCost %>% filter(region == "International")
break_even_International <- data.frame(total_cost = cumsum(transactions_variableCost_International$total_variable_cost),
        sales_revenue = cumsum(transactions_variableCost_International$sales_revenue),
        volume = cumsum(transactions_variableCost_International$quantity_sold),
        date = transactions_variableCost_International$date_of_sale)

break_even_region <- data.frame(
    region = unique(transactions_variableCost$region),
    break_even_date = c(break_even_Northeast[which(break_even_Northeast$total_cost <= break_even_Northeast$total_cost),
        break_even_Eastcoast[which(break_even_Eastcoast$total_cost <= break_even_Eastcoast$total_cost),
        break_even_Midwest[which(break_even_Midwest$total_cost <= break_even_Midwest$total_cost),
        break_even_Central[which(break_even_Central$total_cost <= break_even_Central$total_cost),
        break_even_West[which(break_even_West$total_cost <= break_even_West$total_cost),
        break_even_South[which(break_even_South$total_cost <= break_even_South$total_cost),
        break_even_Northwest[which(break_even_Northwest$total_cost <= break_even_Northwest$total_cost),
        break_even_International[which(break_even_International$total_cost <= break_even_International$total_cost),
    break_even_sales = c(break_even_Northeast[which(break_even_Northeast$total_cost <= break_even_Northeast$total_cost),
        break_even_Eastcoast[which(break_even_Eastcoast$total_cost <= break_even_Eastcoast$total_cost),
        break_even_Midwest[which(break_even_Midwest$total_cost <= break_even_Midwest$total_cost),
        break_even_Central[which(break_even_Central$total_cost <= break_even_Central$total_cost),
        break_even_West[which(break_even_West$total_cost <= break_even_West$total_cost),
        break_even_South[which(break_even_South$total_cost <= break_even_South$total_cost),
        break_even_Northwest[which(break_even_Northwest$total_cost <= break_even_Northwest$total_cost),
        break_even_International[which(break_even_International$total_cost <= break_even_International$total_cost),
    break_even_quantity = c(break_even_Northeast[which(break_even_Northeast$total_cost <= break_even_Northeast$total_cost),

```

```

break_even_Eastcoast[which(break_even_Eastcoast$total_cost <= break_even_Eastcoast$break_even_date)]
break_even_Midwest[which(break_even_Midwest$total_cost <= break_even_Midwest$break_even_date)]
break_even_Central[which(break_even_Central$total_cost <= break_even_Central$break_even_date)]
break_even_West[which(break_even_West$total_cost <= break_even_West$break_even_date)]
break_even_South[which(break_even_South$total_cost <= break_even_South$break_even_date)]
break_even_Northwest[which(break_even_Northwest$total_cost <= break_even_Northwest$break_even_date)]
break_even_International[which(break_even_International$total_cost <= break_even_International$break_even_date)]
)
break_even_region

```

```

##           region break_even_date break_even_sales break_even_quantity
## 1 Northeast    2018-09-15      8903799          431546
## 2 East coast   2018-08-26     15702736          794605
## 3 Midwest     2018-09-10     22069107          1130497
## 4 Central     2018-09-01     12686386          1266387
## 5 West        <NA>          NA              NA
## 6 South       <NA>          NA              NA
## 7 Northwest   2018-08-05     9650867          470358
## 8 International 2018-04-08     1083576          46049

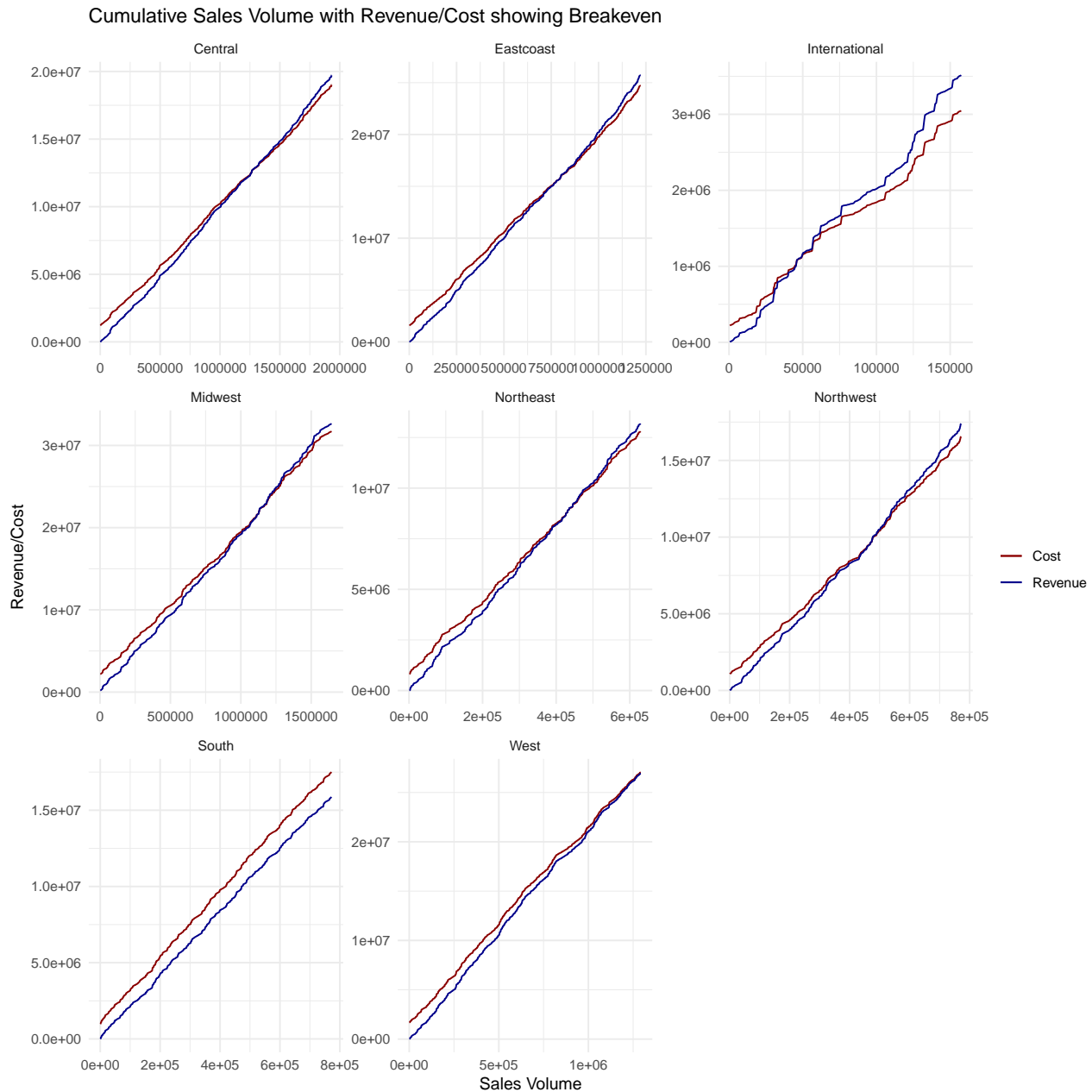
```

As we can see, most regions break even in between August and September but only two regions “West” and “South” did not break even. The region “International” broke even very early in the year only at April.

```

break_even_Northeast %>%
  mutate(region = "Northeast") %>%
  rbind(break_even_Eastcoast %>% mutate(region = "Eastcoast")) %>%
  rbind(break_even_Midwest %>% mutate(region = "Midwest")) %>%
  rbind(break_even_Central %>% mutate(region = "Central")) %>%
  rbind(break_even_West %>% mutate(region = "West")) %>%
  rbind(break_even_South %>% mutate(region = "South")) %>%
  rbind(break_even_Northwest %>% mutate(region = "Northwest")) %>%
  rbind(break_even_International %>% mutate(region = "International")) %>%
  select(-date) %>%
  ggplot(aes(x = volume)) +
    geom_line(aes(y = total_cost, color = "Cost")) +
    geom_line(aes(y = sales_revenue, color = "Revenue")) +
    scale_colour_manual("",
                        breaks = c("Cost", "Revenue"),
                        values = c("darkred", "darkblue")) +
    facet_wrap(~region, scales = "free") +
    theme_minimal() +
    labs(title = "Cumulative Sales Volume with Revenue/Cost showing Breakeven",
         x = "Sales Volume",
         y = "Revenue/Cost")

```



Question 5

a)

Cost absorption is an approach of allocating fixed overhead costs to each unit of a product produced in the same period when the fixed overhead cost is made. This disregards when the produced good is actually sold.

With the data provided, we only know when products are sold but has no information on when it is produced. However, we can make an assumption that the company uses made to order policy and hence the sales is made when the product is produced. Similar to above, we will split the regional overhead costs by regional sales revenue of each transaction.

```
joined_by_region <- joined_by_region %>%
  mutate(overhead_per_sales = (production_overheads+non_production_overheads)/sales_revenue)
```

```

transactions_by_brand_collection <- transactions %>%
  filter(year == 2018) %>%
  left_join(joined_by_region %>% select(region, overhead_per_sales), by = "region") %>%
  mutate(overheads = overhead_per_sales*sales_revenue,
         total_cost = overheads+costs) %>%
  group_by(brand,collection) %>%
  summarize(total_cost = sum(total_cost))

```

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

```
transactions_by_brand_collection
```

```

## # A tibble: 107 x 3
## # Groups:   brand [2]
##   brand    collection total_cost
##   <chr>    <chr>         <dbl>
## 1 Elements Aiden          328333.
## 2 Elements Arcadia        384968.
## 3 Elements Asher          808282.
## 4 Elements Belfast       885453.
## 5 Elements Brenton       1191983.
## 6 Elements Calloway       870604.
## 7 Elements Capri          272849.
## 8 Elements Cosgrove       691912.
## 9 Elements Cypress       145605.
## 10 Elements Drake         496568.
## # ... with 97 more rows

```

b)

Our full cost estimations could be improved if we can know about when the products of each transactions are produced and hence allocate the fixed cost according to their production date instead of date of sales.

If detailed information about each products are hard to gather, mean inventory holding period of each product or turnover rate could also be used to make a rough estimate of when each product is produced.