Case Study: Adventure Works

Adejumo Ridwan Suleiman

2022-09-27

Install Packages

```
#install remotes
#install.packages("remotes")

#install and load adventure works database
remotes::install_github("adejumoridwan/adventureWorks")
```

```
## Skipping install of 'adventureWorks' from a github remote, the SHA1 (04d1c470) has not changed since
##   Use `force = TRUE` to force installation
```

```
library(adventureWorks)

#install tidyverse
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.1

## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.3.6     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

## Warning: package 'ggplot2' was built under R version 4.2.1

## Warning: package 'tibble' was built under R version 4.2.1

## Warning: package 'readr' was built under R version 4.2.1

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

Data in the database

- sales
- customers
- products
- budget
- calendar
- territory

Total Sales Made in each product sub category

```
inner_join(x = sales,
           y = products,
           by = "ProductKey") |>
  group_by(Category,SubCategory) |>
  summarize(Total_Sales = round(sum(SalesAmount))) |>
  arrange(desc(Total_Sales))
```

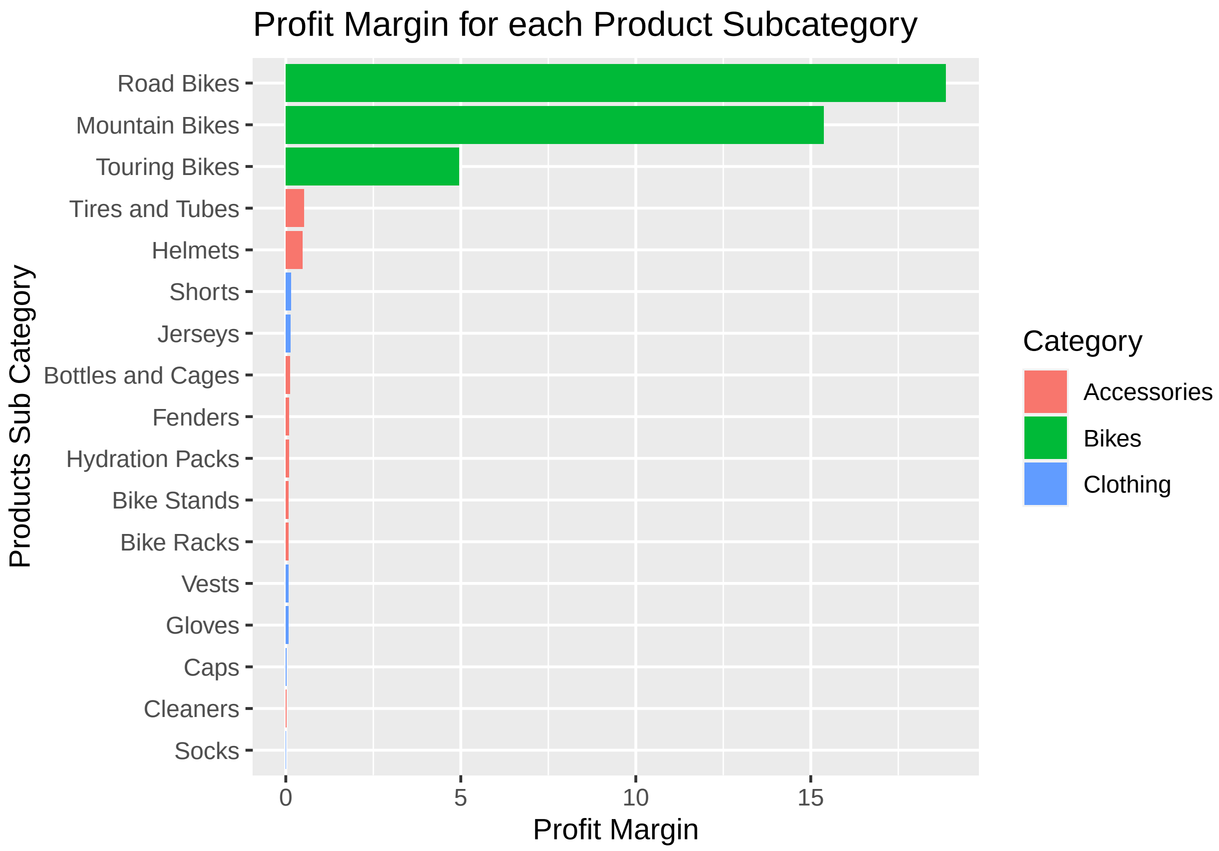
```
## ‘summarise()‘ has grouped output by ’Category’. You can override using the
## ‘.groups‘ argument.
```

```
## # A tibble: 17 x 3
## # Groups:   Category [3]
##    Category    SubCategory       Total_Sales
##    <chr>       <chr>                   <dbl>
##  1 Bikes       Road Bikes           14520584
##  2 Bikes       Mountain Bikes        9952760
##  3 Bikes       Touring Bikes         3844801
##  4 Accessories Tires and Tubes        245529
##  5 Accessories Helmets                225336
##  6 Clothing    Jerseys                172951
##  7 Clothing    Shorts                  71320
##  8 Accessories Bottles and Cages       56798
##  9 Accessories Fenders                 46620
## 10 Accessories Hydration Packs         40308
## 11 Accessories Bike Stands             39591
## 12 Accessories Bike Racks              39360
## 13 Clothing    Vests                   35687
## 14 Clothing    Gloves                  35021
## 15 Clothing    Caps                    19688
## 16 Accessories Cleaners                 7219
## 17 Clothing    Socks                    5106
```

Profit Margin made in each Product category

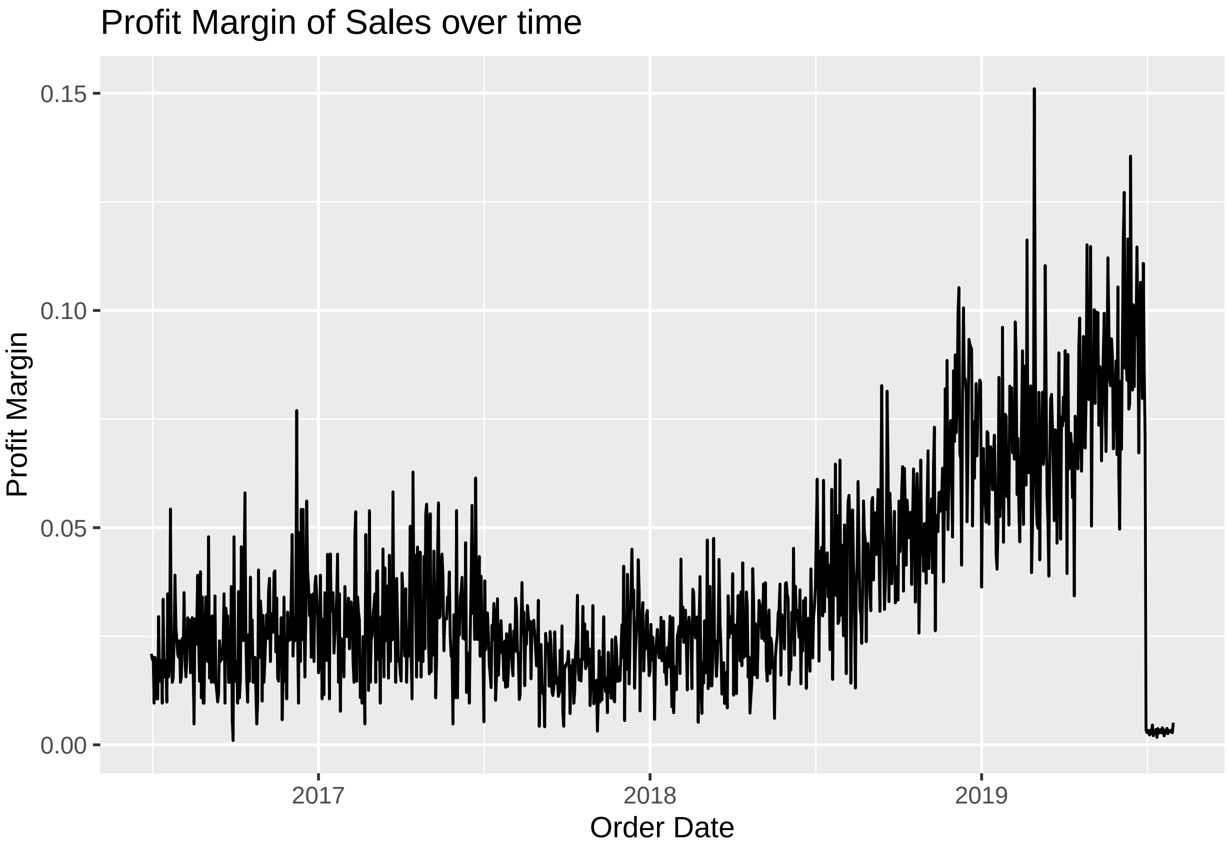
```
inner_join(x = sales,
           y = products,
          by = "ProductKey") |>
  mutate(Profit_Margin =
           (SalesAmount - ProductStandardCost)/sum(SalesAmount)*100)|>
  group_by(Category,SubCategory) |>
  summarize(Total_Profit_Margin = sum(Profit_Margin)) |>
  arrange(desc(Total_Profit_Margin)) |>
  ggplot(aes(x = Total_Profit_Margin,
             y = reorder(SubCategory, Total_Profit_Margin),
             fill = Category)) +
  geom_col() +
  labs(x = "Profit Margin",
       y = "Products Sub Category",
       title = "Profit Margin for each Product Subcategory")
```

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



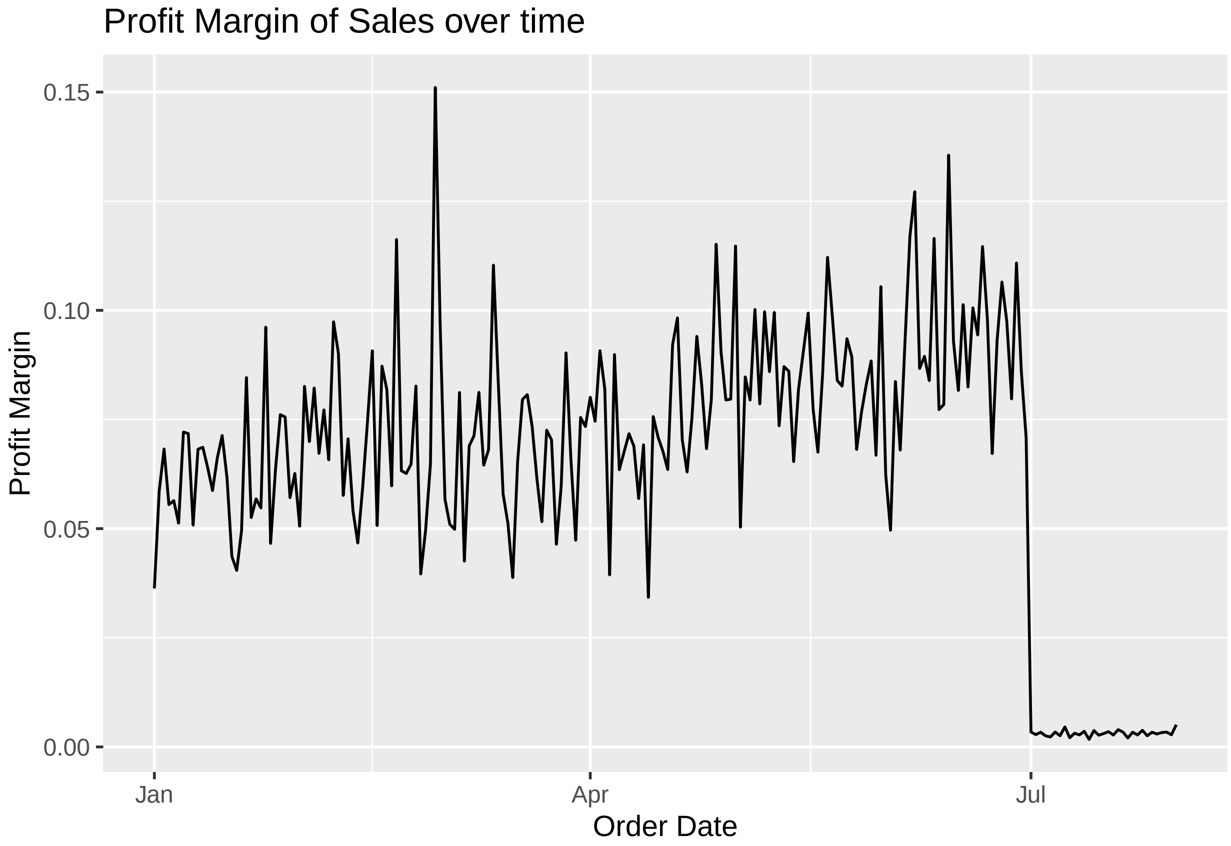
Profit Margin of Sales over time

```
sales |>
  mutate(Profit_Margin =
           (SalesAmount - ProductStandardCost)/sum(SalesAmount)*100,
         OrderDate = lubridate::dmy(OrderDate)) |>
  group_by(OrderDate) |>
  summarize(Profit_Margin = sum(Profit_Margin)) |>
  ggplot(aes(x = OrderDate,
             y = Profit_Margin)) +
  geom_line() +
  labs(x = "Order Date",
       y = "Profit Margin",
       title = "Profit Margin of Sales over time")
```



What happened in 2019

```
sales |>
  mutate(Profit_Margin =
           (SalesAmount - ProductStandardCost)/sum(SalesAmount)*100,
         OrderDate = lubridate::dmy(OrderDate)) |>
  group_by(OrderDate) |>
  summarize(Profit_Margin = sum(Profit_Margin)) |>
  filter(OrderDate >= "2019-01-01") |>
  ggplot(aes(x = OrderDate,
             y = Profit_Margin)) +
  geom_line() +
  labs(x = "Order Date",
       y = "Profit Margin",
       title = "Profit Margin of Sales over time")
```



What happened after July

```
inner_join(x = sales,
           y = products,
           by = "ProductKey") |>
  mutate(Profit_Margin =
           (SalesAmount - ProductStandardCost)/sum(SalesAmount)*100,
         OrderDate = lubridate::dmy(OrderDate)) |>
  filter(OrderDate >= "2019-07-01") |>
  group_by(Category) |>
  summarize(Profit_Margin = sum(Profit_Margin))
```

```
## # A tibble: 2 x 2
##   Category    Profit_Margin
##   <chr>               <dbl>
## 1 Accessories        0.0720
## 2 Clothing           0.0247
```

So products belonging to bike categories were not sold, no wonder sales dropped

Customers where most profit were made from

```
inner_join(x = sales,
           y = customers,
           by = "CustomerKey") |>
  mutate(Profit_Margin =
           (SalesAmount - ProductStandardCost)/sum(SalesAmount)*100) |>
  group_by(Occupation, MaritalStatus, Gender) |>
  summarize(Total_Profit = sum(Profit_Margin)) |>
  arrange(desc(Total_Profit))
```

```
## ‘summarise()‘ has grouped output by ’Occupation’, ’MaritalStatus’. You can
## override using the ‘.groups‘ argument.
```

```
## # A tibble: 20 x 4
## # Groups:   Occupation, MaritalStatus [10]
##    Occupation     MaritalStatus Gender Total_Profit
##    <chr>          <chr>         <chr>         <dbl>
##  1 Professional   S             F             3.90
##  2 Professional   M             M             3.55
##  3 Professional   M             F             3.34
##  4 Professional   S             M             3.17
##  5 Skilled Manual M             M             2.79
##  6 Management     M             M             2.41
##  7 Skilled Manual M             F             2.36
##  8 Skilled Manual S             F             2.31
##  9 Management     M             F             2.13
## 10 Clerical       M             M             1.83
## 11 Clerical       M             F             1.81
## 12 Management     S             F             1.63
## 13 Skilled Manual S             M             1.54
## 14 Clerical       S             M             1.53
## 15 Management     S             M             1.53
## 16 Manual         S             M             1.49
## 17 Manual         S             F             1.38
## 18 Clerical       S             F             1.34
## 19 Manual         M             F             0.565
## 20 Manual         M             M             0.550
```

Exercises

Question 1

Which sub category was sales made most from for single female.
1. Touring Bikes 2. Mountain Bikes 3. Helmets 4. None of the Above - Ans

```
inner_join(x = sales,
           y = customers,
           by = "CustomerKey") |>
```

```
  inner_join(y = products,
             by = "ProductKey") |>
  filter(Occupation == "Professional",
         MaritalStatus == "S",
         Gender == "F") |>
  group_by(Category, SubCategory) |>
  summarize(Total_Sales = sum(ExtendedAmount)) |>
  arrange(desc(Total_Sales))
```

```
## ‘summarise()‘ has grouped output by ’Category’. You can override using the
## ‘.groups‘ argument.
```

```
## # A tibble: 17 x 3
## # Groups:   Category [3]
##    Category    SubCategory       Total_Sales
##    <chr>       <chr>                   <dbl>
##  1 Bikes       Road Bikes           1220218.
##  2 Bikes       Mountain Bikes       1033551.
##  3 Bikes       Touring Bikes         454426.
##  4 Accessories Helmets                16235.
##  5 Accessories Tires and Tubes        15735.
##  6 Clothing    Jerseys                12422.
##  7 Clothing    Shorts                  4059.
##  8 Accessories Bottles and Cages       3850.
##  9 Accessories Hydration Packs         3464.
## 10 Accessories Bike Stands             3339
## 11 Accessories Fenders                 3121.
## 12 Accessories Bike Racks              2880
## 13 Clothing    Vests                   2476.
## 14 Clothing    Gloves                  2376.
## 15 Clothing    Caps                    1492.
## 16 Accessories Cleaners                 485.
## 17 Clothing    Socks                    342.
```

Question 2

Calculate the total tax on the product category Bikes.
1. 2,265,451.62 - Ans 2. 56,060.80 3. 27,181.81 4. None of the Above

```
inner_join(x = sales,
           y = products,
           by = "ProductKey") |>
  group_by(Category) |>
  summarize(Total_Tax = sum(TaxAmt)) |>
  arrange(desc(Total_Tax))
```

```
## # A tibble: 3 x 2
##   Category    Total_Tax
##   <chr>           <dbl>
## 1 Bikes        2265452.
## 2 Accessories    56061.
## 3 Clothing       27182.
```

Question 3

What is the total profit on bikes made from customers who are Professionals. Note: Profit = Selling Price - Cost Price.
1. 3,916,901
2. 3,912,902 3. 3,716,902 4. None of the Above - Ans

```
inner_join(x = sales,
           y = customers,
           by = "CustomerKey") |>
  inner_join(y = products,
              by = "ProductKey") |>
  mutate(Profit = SalesAmount - ProductStandardCost) |>
  filter(Occupation == "Professional") |>
  group_by(Category) |>
  summarize(Profit = sum(Profit)) |>
  filter(Category == "Bikes")
```

```
## # A tibble: 1 x 2
##   Category   Profit
##   <chr>       <dbl>
## 1 Bikes    3916902.
```