Project 2, Dataset 2

Alice Ding

2023-02-27

Overview

For this dataset, I'll be using the first one posted by Waheeb and it represents sales data for different product lines based on a specific date. This data has one row per date and includes the following columns:

- Date
- Product Line 1
- Product Line 2
- Product Line 3

The last three columns are dollar values.

```
product_data <- read.csv("https://raw.githubusercontent.com/addsding/data607/main/project2/product-lines
head(product_data)
```

```
##      Date Product.Line.1 Product.Line.2 Product.Line.3
## 1 1/17/23           2500           1250           5000
## 2  2/4/23           1000           1000           4500
## 3  4/8/23            980           2000            850
## 4  5/7/23            990           3000            976
## 5 6/17/23           3000           5000           1500
```

Our goal is to flatten this table to be one row per date and product line number combination before beginning analysis.

Tidying the Data

To clean this data frame, we'll be pivoting it.

```
product_data_pivot <- pivot_longer(product_data, cols=2:4, names_to="product_line", values_to="sales")
head(product_data_pivot)
```

```
## # A tibble: 6 x 3
##   Date    product_line   sales
##   <chr>   <chr>          <int>
## 1 1/17/23 Product.Line.1  2500
## 2 1/17/23 Product.Line.2  1250
## 3 1/17/23 Product.Line.3  5000
## 4 2/4/23  Product.Line.1  1000
## 5 2/4/23  Product.Line.2  1000
## 6 2/4/23  Product.Line.3  4500
```

The pivot has worked, but I'll want to reformat the `product_line` column to just be an int to represent each product line. The `Date` column also should be changed into an actual date.

```
product_data_pivot$product_line <- gsub("\\.", " ", product_data_pivot$product_line)

product_data_pivot$Date <- as.Date(product_data_pivot$Date,
  format = "%m/%d/%y")

head(product_data_pivot)
```

```
## # A tibble: 6 x 3
##   Date       product_line   sales
##   <date>     <chr>          <int>
## 1 2023-01-17 Product Line 1  2500
## 2 2023-01-17 Product Line 2  1250
## 3 2023-01-17 Product Line 3  5000
## 4 2023-02-04 Product Line 1  1000
## 5 2023-02-04 Product Line 2  1000
## 6 2023-02-04 Product Line 3  4500
```

Looks good, time to analyze!

Analysis

To begin, we can find the stats for sales for each product line.

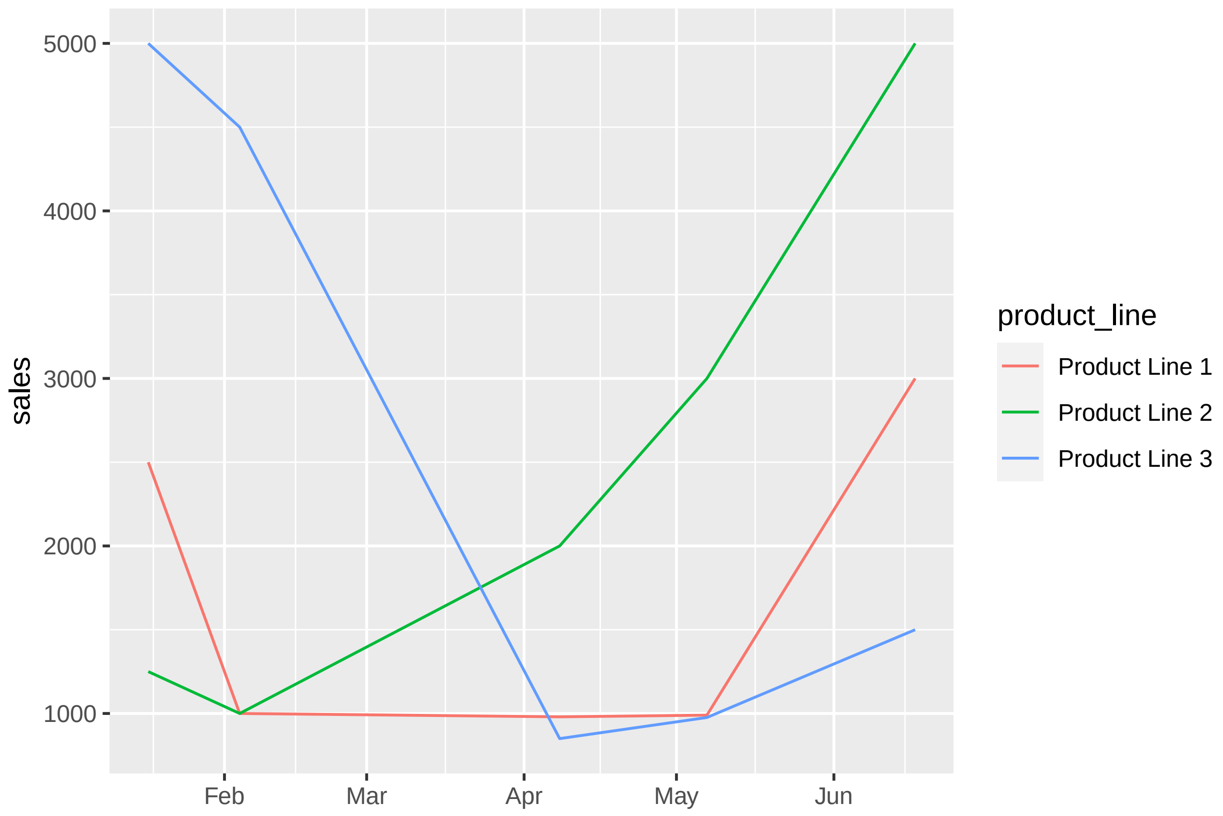
```
sales <- product_data_pivot |>
  group_by(product_line) |>
  summarise(mean_sales = mean(sales),
            median_sales = median(sales),
            min_sales = min(sales),
            max_sales = max(sales),
            .groups = 'drop')

sales
```

```
## # A tibble: 3 x 5
##   product_line   mean_sales median_sales min_sales max_sales
##   <chr>               <dbl>        <int>     <int>     <int>
## 1 Product Line 1      1694          1000       980      3000
## 2 Product Line 2      2450          2000      1000      5000
## 3 Product Line 3      2565.         1500       850      5000
```

On average, product line 3 seems to be doing the best as on average, it has sales of $2,500+. Product line 2 isn't that far behind at $2,450, while product line 1 seems to trail behind at only $1,700. We can see though that product line 3 has a pretty large range of sales from $850 to $5,000 – could this be due to seasonality?

```
product_time_series <- ggplot(product_data_pivot, aes(x=Date, y=sales, color=product_line)) +
  geom_line() +
  xlab("")
product_time_series
```



When looking at this data over time, it tells a very different story. Product line 2 seems to be growing a lot while product line 3 has not been doing so well, really tanking in sales for the first quarter of the year. Product line 1 was relatively consistent after a pretty huge drop from January to February, however seems to be bouncing back as of June.

Conclusion

Overall, this data was relatively simple to clean and the findings were pretty straight-forward. One piece of information that I think is super important here though is supply as well as overall price of each product line – we can't really compare the performance of each product line without knowing how much each unit costs as well as how much was actually produced. If for example, product line 1 just is priced at a lower tier and had less units made, its performance would actually be more impressive if product line 3 was over-produced and was at a much higher price point. To continue with this analysis, we'd definitely need more data points to fully gage how well each product line is performing.