Data Viz Home Work Batch 11

Tanaban Kob

2025-05-09

My Project

My ggplot2 project 5 Dashboard

I use dataset from "Global Fashion Retail Sales" 'https://www.kaggle.com/datasets/ricgomes/global-fashion-retail-stores-dataset/data?select=transactions.csv'

Install and Import data

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(tinytex)
customers <- read_csv("customers.csv")
```

```
## Rows: 1643306 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): Name, Email, Telephone, City, Country, Gender, Job Title
## dbl  (1): Customer ID
## date (1): Date Of Birth
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
customers <- select(customers, !matches("Email|Telephone"))

transaction <- read_csv("transactions.csv")
```

```
## Rows: 6416827 Columns: 19
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (8): Invoice ID, Size, Color, Currency, Currency Symbol, SKU, Transact...
## dbl  (10): Line, Customer ID, Product ID, Unit Price, Quantity, Discount, Li...
## dttm  (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
trans_usd <- transaction %>%
  filter(Currency == "USD")

products <- read_csv("products.csv")
```

```
## Rows: 17940 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): Category, Sub Category, Description PT, Description DE, Descriptio...
## dbl  (2): Product ID, Production Cost
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

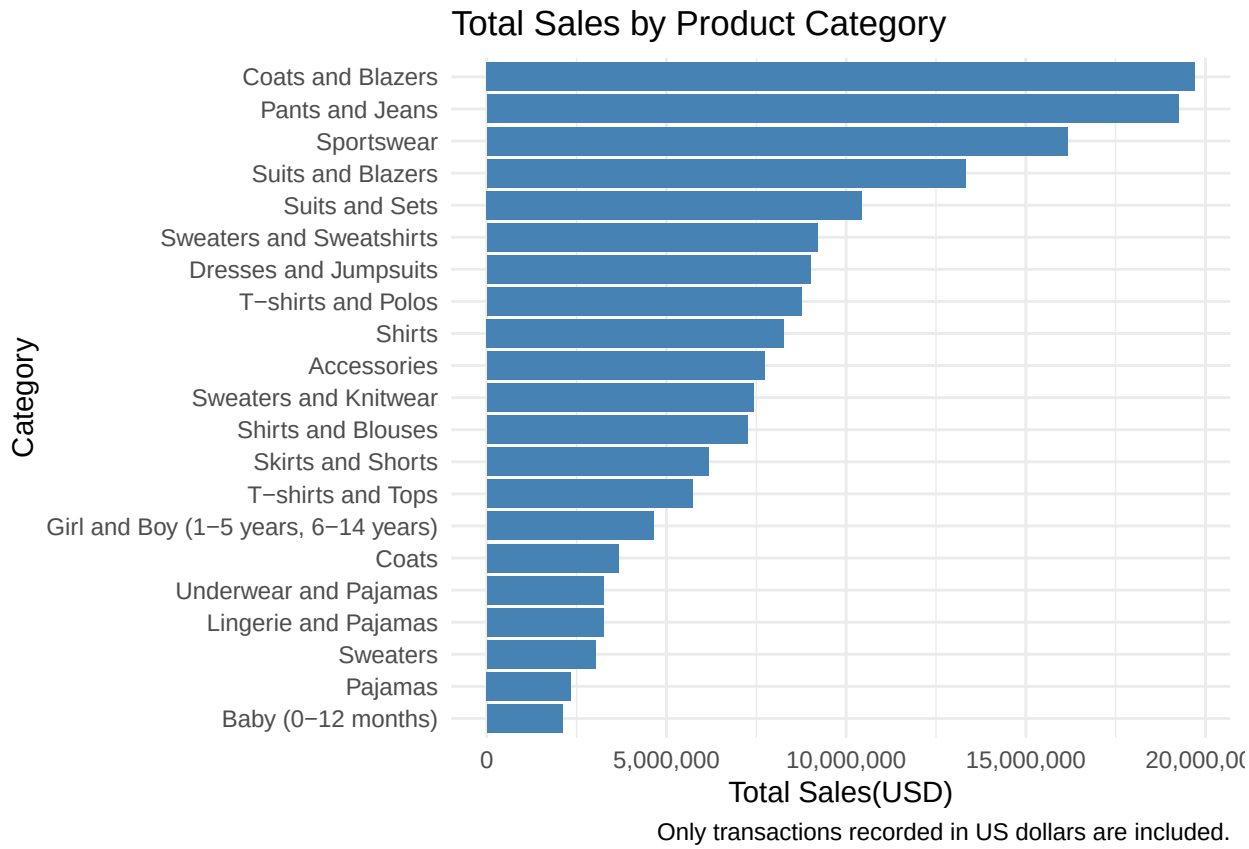
```
products <- products %>%
              select(-contains("Description"))

store <- read_csv("prep_stores.csv")
```

```
## Rows: 35 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): Country, City, Store Name, ZIP Code
## dbl (2): Store ID, Number of Employees
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

1.Total Sales by Product Category

```
trans_usd %>%
  left_join(products, by = "Product ID") %>%
  filter(`Invoice Total` > 0) %>%
  group_by(`Sub Category`) %>%
  summarise(total_sales = sum(`Invoice Total`, na.rm = TRUE)) %>%
```

```r
  arrange(desc(total_sales)) %>%
  ggplot(mapping = aes(x = reorder(`Sub Category`,total_sales),
                       y = total_sales)) +
  scale_y_continuous(labels = scales::comma) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(title = "Total Sales by Product Category",
       caption = "Only transactions recorded in US dollars are included.",
       x = "Category",
       y = "Total Sales(USD)") +
  theme_minimal()
```



This chart shows the total sales for each product category. The sales values are calculated from invoices. Categories with higher bars had more total sales. Only transactions using US dollars are included.

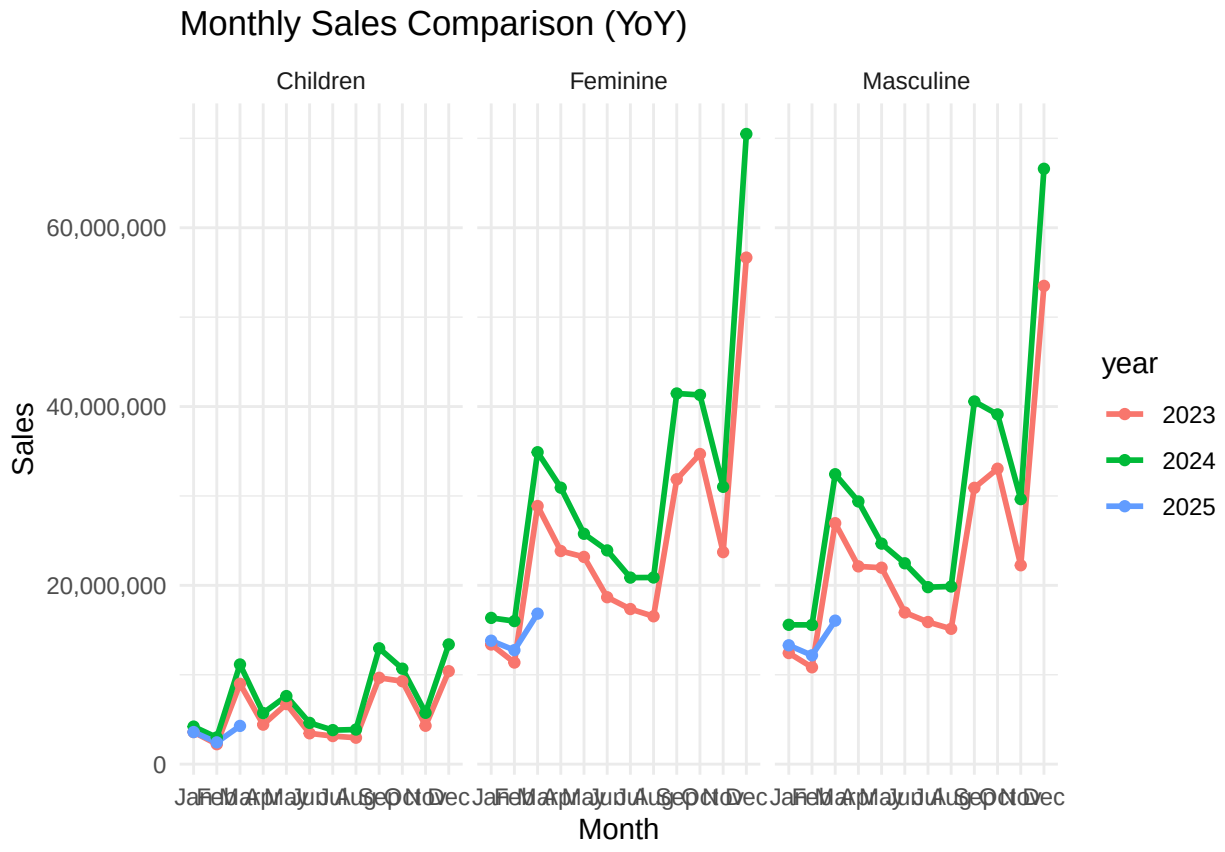
2. Monthly Sales Comparison (YoY) per Category

```r
transaction %>%
  left_join(products, by = "Product ID") %>%
  mutate(date = as.Date(Date),
         month_num = format(date, "%m"),
         month_name = format(date, "%b"),
         year = format(date, "%Y")) %>%
```

```r
  group_by(Category, year,month_name, month_num) %>%
  summarise(monthly_sales = sum(`Invoice Total`,na.rm = TRUE)) %>%
  mutate(month_name = factor(month_name, levels = month.abb)) %>%
  ggplot(aes(x = month_name,
             y = monthly_sales,
             color = year,
             group = year)) +
  geom_line(size = 1) +
  geom_point() +
  facet_wrap(~ Category) +
  scale_y_continuous(labels = scales::comma) +
  labs(title = "Monthly Sales Comparison (YoY)",
       x = "Month",
       y = "Sales") +
  theme_minimal()
```

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

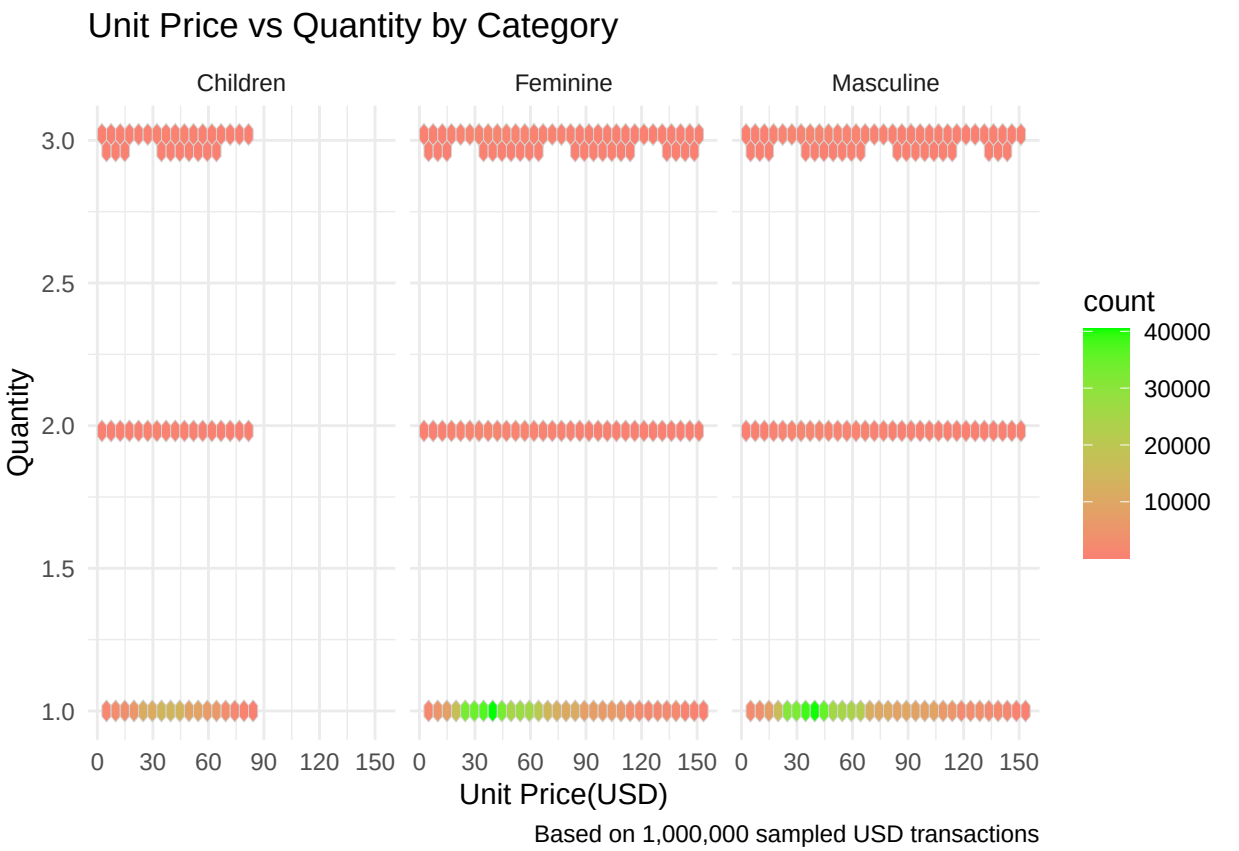
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Monthly sales comparison across 2023–2025 for each product category. Lines show how sales changed each month in each year.

3.Unit Price vs Quantity by Category

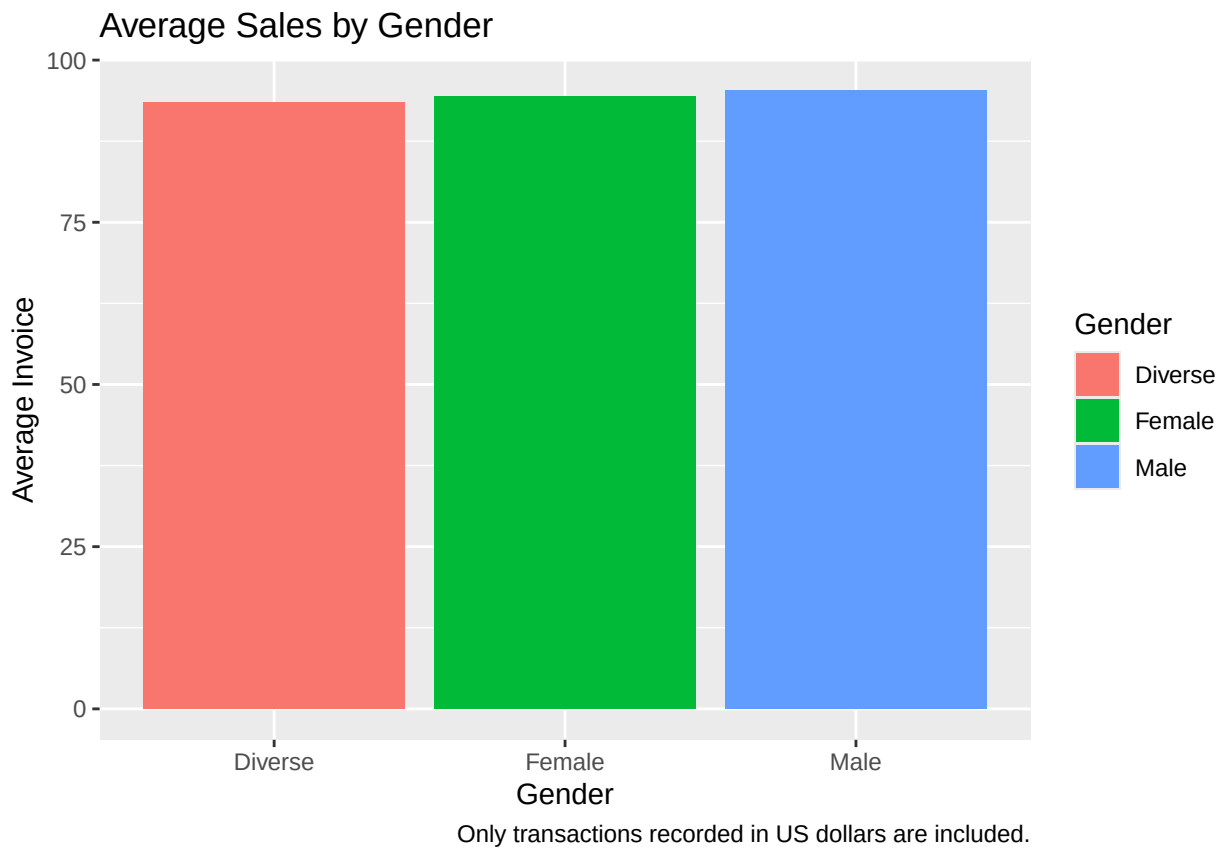
```
prep_df <- trans_usd %>%
  left_join(products, by = "Product ID") %>%
  filter(if_all(c(`Unit Price`, Quantity, Category, `Invoice Total`), ~!is.na(.))) %>%
  select(unit_price = `Unit Price`,
         quantity = Quantity,
         category = Category,
         invoice_total = `Invoice Total`)

prep_df %>%
  filter(invoice_total > 0) %>%
  sample_n(1000000) %>%
  ggplot(mapping = aes(x = unit_price,
                       y = quantity)) +
  geom_hex(colour = "lightgray",
           size = 0.15) +
  facet_wrap(~ category) +
  scale_fill_gradient(low = "salmon",
                      high = "green") +
  scale_y_continuous(breaks = seq(1 ,5 ,by = 0.5 )) +
  scale_x_continuous(breaks = seq(0, 200, by = 30)) +
  labs(title = "Unit Price vs Quantity by Category",
       caption = "Based on 1,000,000 sampled USD transactions",
       x = "Unit Price(USD)",
       y = "Quantity") +
  theme_minimal()
```



This chart shows how many units customers buy at different price levels. Most customers buy only one unit, especially for products under $60.

4.Average Sales by Gender

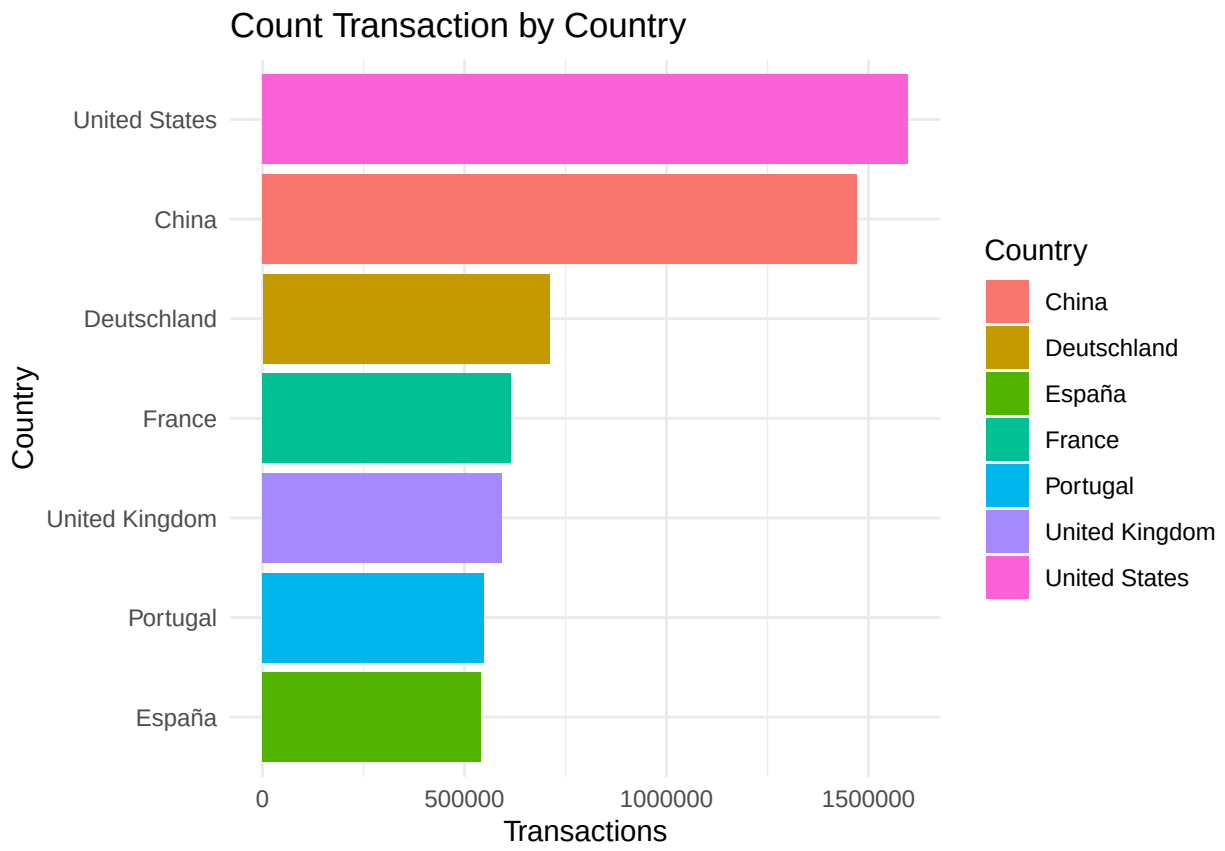
```
trans_usd %>%
  left_join(customers, by = "Customer ID") %>%
  mutate(Gender = case_when(
    Gender == "F" ~ "Female",
    Gender == "M" ~ "Male",
    Gender == "D" ~ "Diverse",
    TRUE ~ "Unknow")) %>%
  group_by(Gender) %>%
  summarise(avg_sales = mean(`Invoice Total`, na.rm = TRUE)) %>%
  ggplot(mapping = aes(x = Gender,
                       y = avg_sales,
                       fill = Gender)) +
  geom_col() +
  labs(title = "Average Sales by Gender",
       caption = "Only transactions recorded in US dollars are included.",
       x = "Gender",
       y = "Average Invoice")
```



Average amount spent per transaction by gender. Data includes only US dollar transactions.

5.Count Transaction by Country

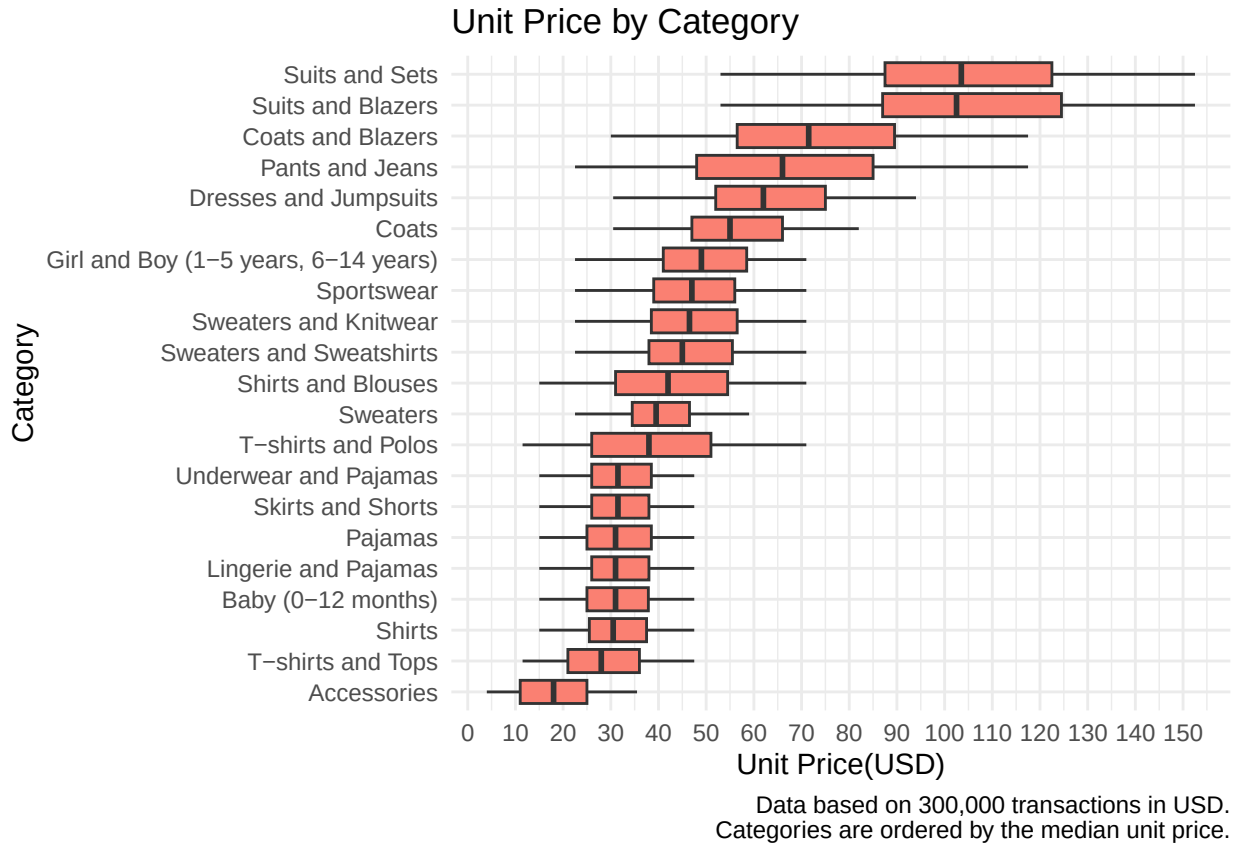
```
plot1 <- transaction %>%
  left_join(store, by = "Store ID") %>%
  filter(!is.na(`Invoice Total`), !is.na(Country), `Invoice Total` > 0) %>%
  select(`Invoice ID`, Country) %>%
  group_by(Country) %>%
  summarise(n = n())

plot1 %>%
  ggplot(mapping = aes(x = reorder(Country, n),
                       y = n,
                       fill = Country)) +
  geom_col() +
  coord_flip() +
  labs(title = "Count Transaction by Country",
       x = "Country",
       y = "Transactions") +
  theme_minimal()
```



Number of customer transactions by country. The chart helps compare which countries have more or fewer purchases.

6.Unit Price by Category with boxplot

```
trans_usd %>%
  left_join(products, by = "Product ID") %>%
  filter(!is.na(`Unit Price`), !is.na(`Sub Category`), Currency == "USD") %>%
  sample_n(300000) %>%
  select(`Unit Price`, `Sub Category`) %>%
  ggplot(mapping = aes(x = reorder(`Sub Category`, `Unit Price`, FUN = median),
                       y = `Unit Price`)) +
  geom_boxplot(fill = "salmon") +
  scale_y_continuous(breaks = seq(0, 1000, by = 10)) +
  coord_flip() +
  labs(title = "Unit Price by Category",
       y = "Unit Price(USD)",
       x = "Category",
       caption = "Data based on 300,000 transactions in USD.\n Categories are ordered by the median uni
  theme_minimal()
```

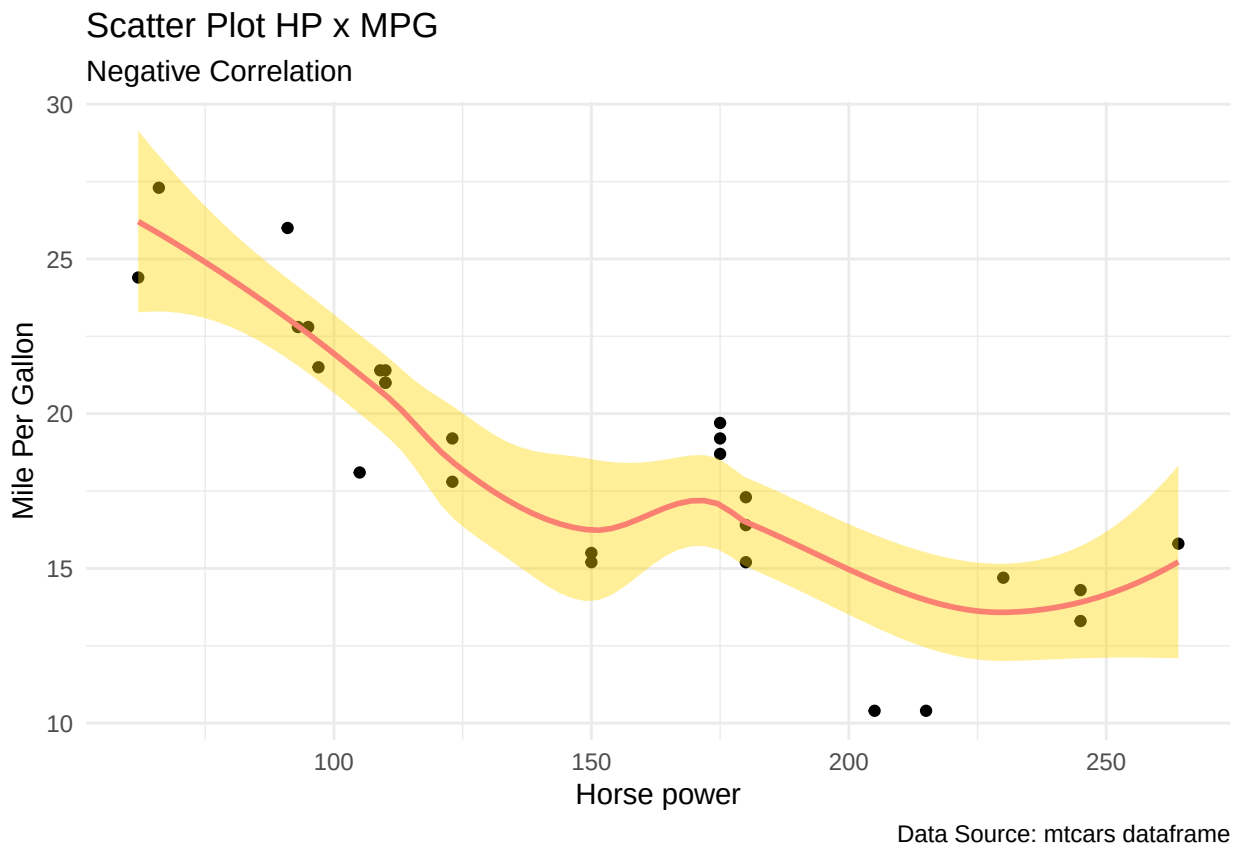


This chart shows the unit price range for each product category. Prices are shown in USD and sorted by the median price.

7.Scatter Plot with dataset "mtcars"

```
ggplot(mtcars %>%
         filter(hp < 300 & mpg < 30),
       aes(hp, mpg)) +
  geom_point() +
  geom_smooth(method = loess,
              se = T,
              color = "salmon",
              fill = "gold") +
  theme_minimal() +
  labs(title = "Scatter Plot HP x MPG",
       subtitle = "Negative Correlation",
       caption = "Data Source: mtcars dataframe",
       x = "Horse power",
       y = "Mile Per Gallon")
```

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



This chart shows the relationship between horsepower and fuel efficiency. When horsepower increases, cars usually use more fuel and get fewer miles per gallon.