WEEK_2 R PROJECT

Nancy Wangare

2025-06-07

IMPORT LIBRARIES

```
pacman::p_load(

 #data importation
 tidyverse, readxl, tidylog, data.table, janitor, tidyr, dplyr,

 #Data Analysis
 skimr,summarytools,psych,Hmisc,

 #Data Visualisation
 ggpubr, plotly, GGally, factoextra)
```

LOAD DATA

```
df <- read_excel("C:/Users/Nancy/Documents/Mysql project/Week2_R_ProjectData_5000_Rows.xlsx")
```

```
str(df)
```

```
## tibble [5,000 x 6] (S3: tbl_df/tbl/data.frame)
##  $ CustomerID: num [1:5000] 1001 1002 1003 1004 1005 ...
##  $ Region    : chr [1:5000] "North" "South" "East" "North" ...
##  $ Product   : chr [1:5000] "Widget C" "Widget C" "Widget C" "Widget C" ...
##  $ Quantity  : num [1:5000] 5 10 10 10 8 9 2 5 7 6 ...
##  $ Price     : num [1:5000] 30 30 30 30 30 20 30 30 20 20 ...
##  $ Date      : POSIXct[1:5000], format: "2024-01-01" "2024-01-02" ...
```

```
df
```

```
## # A tibble: 5,000 x 6
##    CustomerID Region Product  Quantity Price Date
##         <dbl> <chr>  <chr>       <dbl> <dbl> <dttm>
##  1       1001 North  Widget C        5    30 2024-01-01 00:00:00
##  2       1002 South  Widget C       10    30 2024-01-02 00:00:00
##  3       1003 East   Widget C       10    30 2024-01-03 00:00:00
##  4       1004 North  Widget C       10    30 2024-01-04 00:00:00
##  5       1005 North  Widget C        8    30 2024-01-05 00:00:00
##  6       1006 South  Widget A        9    20 2024-01-06 00:00:00
##  7       1007 East   Widget C        2    30 2024-01-07 00:00:00
##  8       1008 East   Widget C        5    30 2024-01-08 00:00:00
##  9       1009 North  Widget A        7    20 2024-01-09 00:00:00
## 10       1010 West   Widget A        6    20 2024-01-10 00:00:00
## # i 4,990 more rows
```

Understanding the Dataset

1. What does each column in the dataset represent?

```
colnames(df)
```

```
## [1] "CustomerID" "Region"     "Product"    "Quantity"   "Price"
## [6] "Date"
```

```
    library(dplyr)
    glimpse(df)
```

```
## Rows: 5,000
## Columns: 6
## $ CustomerID <dbl> 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010,~
## $ Region     <chr> "North", "South", "East", "North", "North", "South", "East"~
## $ Product    <chr> "Widget C", "Widget C", "Widget C", "Widget C", "Widget C",~
## $ Quantity   <dbl> 5, 10, 10, 10, 8, 9, 2, 5, 7, 6, 6, 7, 3, 3, 10, 9, 8, 5, 6~
## $ Price      <dbl> 30, 30, 30, 30, 30, 20, 30, 30, 20, 20, 20, 20, 20, 20, 30,~
## $ Date       <dttm> 2024-01-01, 2024-01-02, 2024-01-03, 2024-01-04, 2024-01-05~
```

2. Are there any missing or inconsistent values in the dataset?

```
colSums(is.na(df))
```

```
## CustomerID     Region    Product   Quantity      Price       Date
##          0          0          0          0          0          0
```

3. What is the range of dates in the dataset?

```
range(df$Date)
```

```
## [1] "2024-01-01 UTC" "2037-09-08 UTC"
```

Data Cleaning with dplyr

4. How can I remove rows with missing values?

```r
sales_data_cleaned <- na.omit(df)
```

```r
data_inspection <- function(df) {
  cat("Column types:\n")
  str(df)

  cat("\nMissing values per column:\n")
  print(colSums(is.na(df)))
}

# Run the function
data_inspection(df)
```

```
## Column types:
## tibble [5,000 x 6] (S3: tbl_df/tbl/data.frame)
##  $ CustomerID: num [1:5000] 1001 1002 1003 1004 1005 ...
##  $ Region    : chr [1:5000] "North" "South" "East" "North" ...
##  $ Product   : chr [1:5000] "Widget C" "Widget C" "Widget C" "Widget C" ...
##  $ Quantity  : num [1:5000] 5 10 10 10 8 9 2 5 7 6 ...
##  $ Price     : num [1:5000] 30 30 30 30 30 20 30 30 20 20 ...
##  $ Date      : POSIXct[1:5000], format: "2024-01-01" "2024-01-02" ...
##
## Missing values per column:
## CustomerID     Region    Product   Quantity      Price       Date
##          0          0          0          0          0          0
```

5. Do any columns have incorrect or unnecessary values?

```r
clean_df <- df %>%
drop_na()
```

```
## drop_na: no rows removed
```

6. Are there duplicate rows?

```r
clean_df$Date <- as.Date(clean_df$Date)
```

```r
# View duplicate rows only
clean_df[duplicated(clean_df), ]
```

```
## # A tibble: 0 x 6
## # i 6 variables: CustomerID <dbl>, Region <chr>, Product <chr>, Quantity <dbl>,
## #   Price <dbl>, Date <date>
```

Data Grouping and Summarizing

7. How can I group the data by Region and Product?

```
group_data <-  clean_df %>%
   group_by(Region , Product) %>%
   summarise(Total_quantity = sum( Quantity ),
             average_price = mean( Price ))
```

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

```
head(group_data)
```

```
## # A tibble: 6 x 4
## # Groups:   Region [2]
##   Region Product  Total_quantity average_price
##   <chr>  <chr>             <dbl>         <dbl>
## 1 East   Widget A           2450            20
## 2 East   Widget B           2290            15
## 3 East   Widget C           2459            30
## 4 North  Widget A           2345            20
## 5 North  Widget B           2199            15
## 6 North  Widget C           2349            30
```

8. How do I calculate total quantity and total revenue for each group?

```
clean_df <- clean_df %>%
  mutate(Revenue = Quantity * Price)
head(clean_df)
```

```
## # A tibble: 6 x 7
##   CustomerID Region Product  Quantity Price Date       Revenue
##        <dbl> <chr>  <chr>       <dbl> <dbl> <date>       <dbl>
## 1       1001 North  Widget C        5    30 2024-01-01     150
## 2       1002 South  Widget C       10    30 2024-01-02     300
## 3       1003 East   Widget C       10    30 2024-01-03     300
## 4       1004 North  Widget C       10    30 2024-01-04     300
## 5       1005 North  Widget C        8    30 2024-01-05     240
## 6       1006 South  Widget A        9    20 2024-01-06     180
```

```
group_revenue <- clean_df %>%
  group_by(Region , Product) %>%
  summarise(Total_Revenue = sum(Revenue))
```

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

```
group_revenue
```

```
## # A tibble: 12 x 3
## # Groups:   Region [4]
##    Region Product  Total_Revenue
##    <chr>  <chr>            <dbl>
##  1 East   Widget A         49000
##  2 East   Widget B         34350
##  3 East   Widget C         73770
##  4 North  Widget A         46900
##  5 North  Widget B         32985
##  6 North  Widget C         70470
##  7 South  Widget A         48860
##  8 South  Widget B         35955
##  9 South  Widget C         64680
## 10 West   Widget A         40800
## 11 West   Widget B         36240
## 12 West   Widget C         65370
```

9. Can I sort the summarized results in descending order of total revenue?

```
group_revenue %>%
   arrange(desc(Total_Revenue))
```

```
## # A tibble: 12 x 3
## # Groups:   Region [4]
##    Region Product  Total_Revenue
##    <chr>  <chr>            <dbl>
##  1 East   Widget C         73770
##  2 North  Widget C         70470
##  3 West   Widget C         65370
##  4 South  Widget C         64680
##  5 East   Widget A         49000
##  6 South  Widget A         48860
##  7 North  Widget A         46900
##  8 West   Widget A         40800
##  9 West   Widget B         36240
## 10 South  Widget B         35955
## 11 East   Widget B         34350
## 12 North  Widget B         32985
```

Saving Output

10. How can I export the summarized data to a CSV file?

```


file.create("file.output.csv")
```

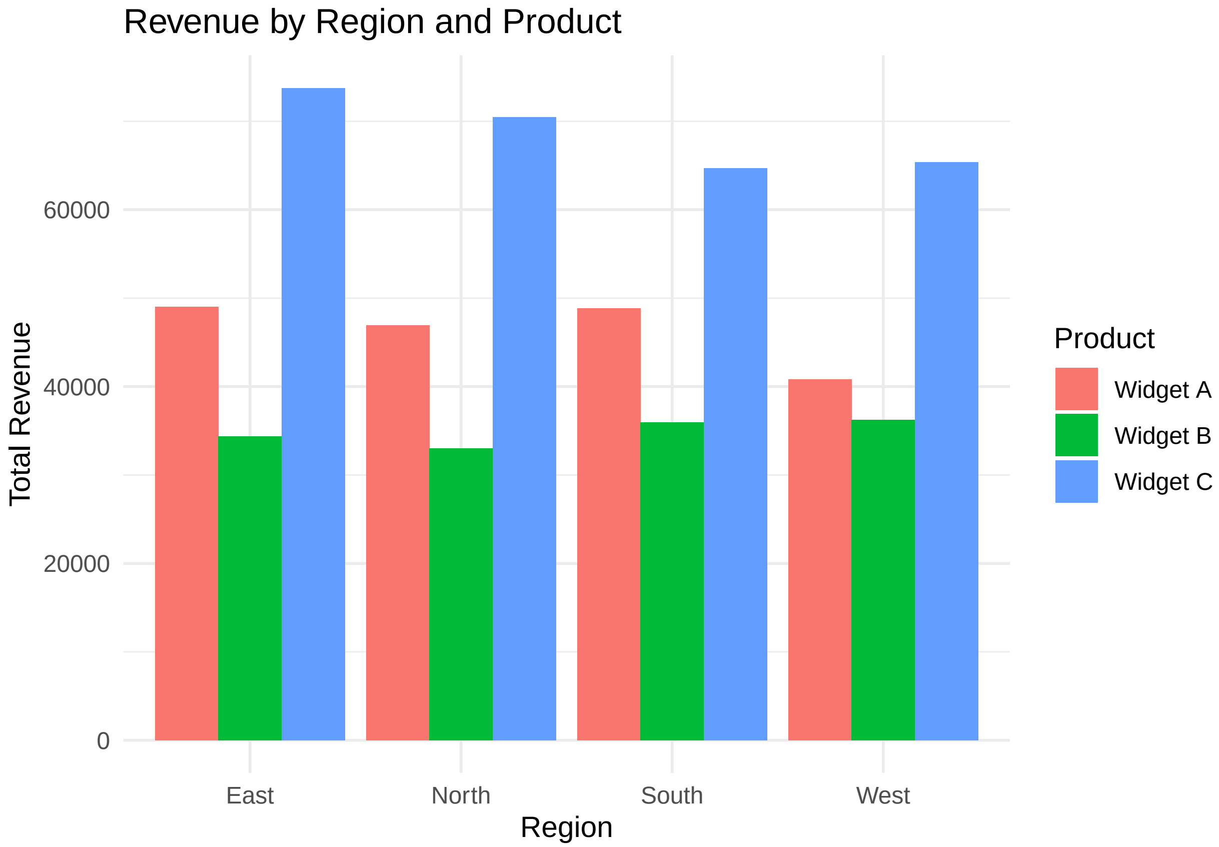
```
## [1] TRUE
```

```
write.csv(group_revenue, "C:/Users/Nancy/Desktop/New folder (6)/file.output.csv", row.names = FALSE)
```

Extension/Reflection Questions

12. What insights can you draw from the summarized data?

```
ggplot(group_revenue, aes(x = Region, y = Total_Revenue, fill = Product)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Revenue by Region and Product",
       x = "Region",
       y = "Total Revenue") +
  theme_minimal()
```



13. How would the analysis change if we added customer demographics (e.g., age, gender)?

Adding **customer demographics** like **age** and **gender** to your analysis can significantly enhance insights by enabling more detailed segmentation and understanding of customer behavior.

14. How can this process be reused for future sales datasets?

To reuse your sales analysis for future datasets:

1. **Create a reusable R script** that automates:

```
Loading data

Summarizing sales

Exporting results

Creating visualizations
```

2. **Make the script modular** using functions so you only need to change the file path.
3. **Organize your files** into folders like `data/`, `output/`, and `scripts/`.
4. **Document the process** to ensure consistency for future datasets