Week 2 Exercises

Adeline Casali

July 14, 2023

Please complete all exercises below. You may use stringr, lubridate, or the forcats library.

Place this at the top of your script: library(stringr) library(lubridate) library(forcats)

Exercise 1

Read the sales_pipe.txt file into an R data frame as sales.

```
library(stringr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(forcats)

sales_df <- read.delim("Data/sales_pipe.txt",
                       stringsAsFactors = FALSE,
                       sep = "|",
                       fileEncoding = "ISO-8859-1")
```

Exercise 2

You can extract a vector of columns names from a data frame using the colnames() function. Notice the first column has some odd characters. Change the column name for the FIRST column in the sales date frame to Row.ID.

Note: You will need to assign the first element of colnames to a single character.

```
colnames(sales_df)[1] <- "Row.ID"
colnames(sales_df)
```

```
##  [1] "Row.ID"        "Order.ID"      "Order.Date"    "Ship.Date"
##  [5] "Ship.Mode"     "Customer.ID"   "Customer.Name" "Segment"
##  [9] "Country"       "City"          "State"         "Postal.Code"
## [13] "Region"        "Product.ID"    "Category"      "Sub.Category"
## [17] "Product.Name"  "Sales"         "Quantity"      "Discount"
## [21] "Profit"
```

Exercise 3

Convert both Order.Date and Ship.Date to date vectors within the sales data frame. What is the number of days between the most recent order and the oldest order? How many years is that? How many weeks?

Note: Use lubridate

```
# Convert Order.Date and Ship.Date to date vectors
sales_df$Order.Date <- mdy(sales_df$Order.Date)
sales_df$Ship.Date <- mdy(sales_df$Ship.Date)

# Number of days between most recent and oldest order
num_days_min_max_order <- as.integer(max(sales_df$Order.Date) - min(sales_df$Order.Date))

# Conversion to years
num_years_min_max_order <- round(num_days_min_max_order / 365, digits = 2)

# Conversion to weeks
num_weeks_min_max_order <- round(num_days_min_max_order / 7, digits = 1)

# Results
cat("The number of days between the most recent order and oldest order is", num_days_min_max_order, ".\
```

```
## The number of days between the most recent order and oldest order is 1457 .
```

```
cat("The number of years between the most recent order and oldest order is", num_years_min_max_order, "
```

```
## The number of years between the most recent order and oldest order is 3.99 .
```

```
cat("The number of weeks between the most recent order and oldest order is", num_weeks_min_max_order, "
```

```
## The number of weeks between the most recent order and oldest order is 208.1 .
```

Exercise 4

What is the average number of days it takes to ship an order?

```
# Calculate the shipping duration
ship_duration <- sales_df$Ship.Date - sales_df$Order.Date

# Calculate the average ship time
avg_ship_time <- round(mean(as.numeric(ship_duration), na.rm = T), digits = 2)
```

```r
# Result
cat("The average number of days to ship an order is", avg_ship_time, ".\n")
```

```
## The average number of days to ship an order is 3.91 .
```

Exercise 5

How many customers have the first name Bill? You will need to split the customer name into first and last name segments and then use a regular expression to match the first name bill. Use the length() function to determine the number of customers with the first name Bill in the sales data.

```r
# Split first and last names by space
sales_names <- str_split_fixed(string = sales_df$Customer.Name, pattern = " ", n = 2)

# Add first and last name columns to df
sales_df$Customer.First.Name = sales_names[ , 1]
sales_df$Customer.Last.Name = sales_names[ , 2]

# Filter for the first name Bill
customers_named_bill <- subset(sales_df, sales_df$Customer.First.Name == "Bill")

# Count the number of customers with the first name Bill and unique last names
bill_unique <- length(unique(customers_named_bill$Customer.Last.Name))

# Result
cat("The number of unique customers with the first name Bill is", bill_unique, ".\n")
```

```
## The number of unique customers with the first name Bill is 6 .
```

Exercise 6

How many mentions of the word 'table' are there in the Product.Name column? **Note you can do this in one line of code**

```r
# Count number of mentions of table
count_table <- sum(str_count(sales_df$Product.Name, pattern = "table"))

# Result
cat("There are", count_table, "mentions of the word table.\n")
```

```
## There are 240 mentions of the word table.
```

Exercise 7

Create a table of counts for each state in the sales data. The counts table should be ordered alphabetically from A to Z.

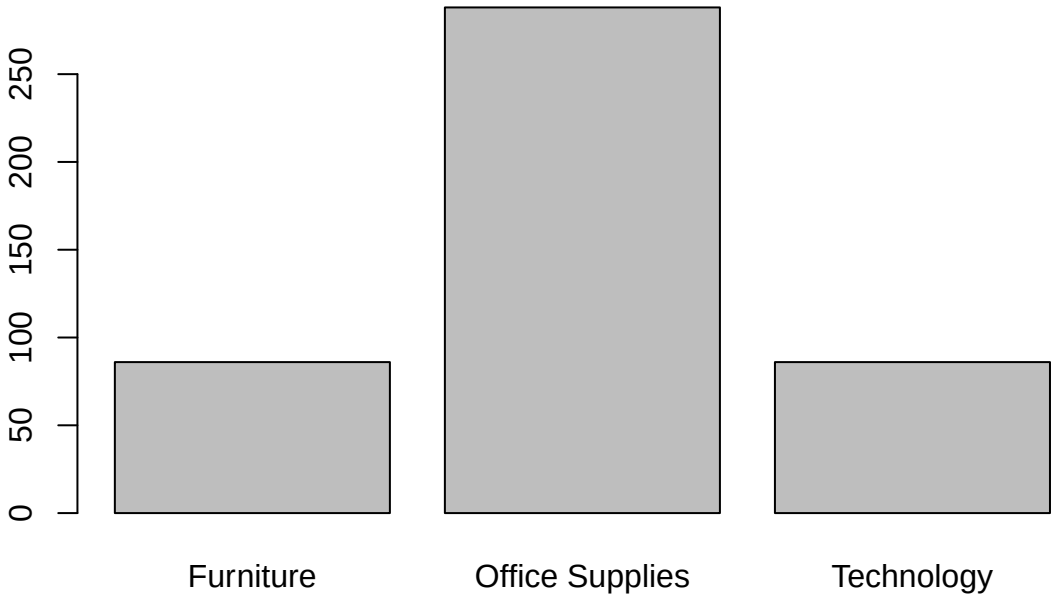
```r
table(sales_df$State)
```

```
##
##              Alabama              Arizona             Arkansas
##                   28                  119                   22
##           California             Colorado          Connecticut
##                  993                   90                   50
##             Delaware District of Columbia              Florida
##                   47                    1                  186
##              Georgia                Idaho             Illinois
##                   79                    9                  286
##              Indiana                 Iowa               Kansas
##                   74                   11                   16
##             Kentucky            Louisiana                Maine
##                   64                   18                    4
##             Maryland        Massachusetts             Michigan
##                   63                   71                  142
##            Minnesota          Mississippi             Missouri
##                   41                   27                   37
##              Montana             Nebraska               Nevada
##                    2                   26                   24
##        New Hampshire           New Jersey           New Mexico
##                    9                   58                   11
##             New York       North Carolina         North Dakota
##                  555                  117                    7
##                 Ohio             Oklahoma               Oregon
##                  211                   38                   56
##         Pennsylvania         Rhode Island       South Carolina
##                  312                   25                   28
##         South Dakota            Tennessee                Texas
##                    9                   88                  460
##                 Utah              Vermont             Virginia
##                   27                   10                   80
##           Washington        West Virginia            Wisconsin
##                  254                    4                   38
##              Wyoming
##                    1
```

Exercise 8

Create an alphabetically ordered barplot for each sales Category in the State of Texas.

```r
barplot(table(sales_df$Category[sales_df$State == "Texas"]))
```



Exercise 9

Find the average profit by region. **Note: You will need to use the aggregate() function to do this. To understand how the function works type ?aggregate in the console.**

```
aggregate(Profit ~ Region, data = sales_df, FUN = mean)
```

```
##    Region   Profit
## 1 Central 20.46822
## 2    East 29.91937
## 3   South 11.27720
## 4    West 32.77000
```

Exercise 10

Find the average profit by order year. **Note: You will need to use the aggregate() function to do this. To understand how the function works type ?aggregate in the console.**

```
# Extract the year
sales_df$Order.Year <- year(sales_df$Order.Date)

# Aggregate
aggregate(Profit ~ Order.Year, data = sales_df, FUN = mean)
```

```
##   Order.Year   Profit
## 1       2014 32.24582
## 2       2015 21.58676
## 3       2016 30.10960
## 4       2017 21.31825
```