Week 2 Exercises

Rebecca Hawthorne

July 8, 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.

```
sales_pipe <- read.delim("Data/sales_pipe.txt"
                         ,stringsAsFactors=FALSE
                         ,sep = "|"
                         )
```

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.

```
#create a vector of the current columnnames
sales_pipe_colnames <- colnames(sales_pipe)

#change name of first column
sales_pipe_colnames[1] <- 'Row.ID'

#replace column names in sales_pipe df
colnames(sales_pipe) <- sales_pipe_colnames
```

Exercise 3

Convert both Ship.Date and Order.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

```
library(lubridate)
```

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

```
## The following object is masked from 'package:base':
##
##     date
```

```
#convert Ship.Date and Order.Date to date types

sales_pipe$Ship.Date <- as.Date(sales_pipe$Ship.Date
```

```
                                ,format='%B %d %Y'
                                )

sales_pipe$Order.Date <- as.Date(sales_pipe$Order.Date
                                ,format='%m/%d/%Y'
                                )

#find the oldest and newest order dates
oldest_order <- min(sales_pipe$Order.Date)

most_recent_order <- max(sales_pipe$Order.Date)

#find the difference between the oldest and newest order dates in days and weeks
days_between_oldest_and_youngest_orders <- difftime(most_recent_order, oldest_order,units = "days")

weeks_between_oldest_and_youngest_orders <- difftime(most_recent_order, oldest_order,
                                                     units =     "weeks")

#convert the number of days to numeric (from difftime) so that I can calculate number of years assuming


numeric_days_between <- as.numeric(days_between_oldest_and_youngest_orders)

years_between_oldest_and_youngest_orders <- numeric_days_between / 365
```

Exercise 4

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

```
#create vector of the number of days it takes to ship
days_to_ship <- (sales_pipe$Ship.Date - sales_pipe$Order.Date)

#calculate mean of the number of days to ship
avg_days_to_ship <- mean(days_to_ship)
```

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.

```
library(stringr)

#break customer name into to columns-first and last name
customer_name_first_last <- str_split_fixed(string=sales_pipe$Customer.Name,pattern='\\s',n=2)

#slice the vector where the first names are Bill
first_name_Bill <- customer_name_first_last[customer_name_first_last[,1]=="Bill"]

#count the length of the Bill vector
number_of_Bills <- length(first_name_Bill)
```

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**

```
library(stringr)
#use sum to add all the times table was counted by str.count in each element
number_of_table <- sum(str_count(sales_pipe$Product.Name, pattern = '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.

```
#convert State to factor type
sales_pipe$State <- factor(sales_pipe$State)

#create a table of the factor counts
table(sales_pipe$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

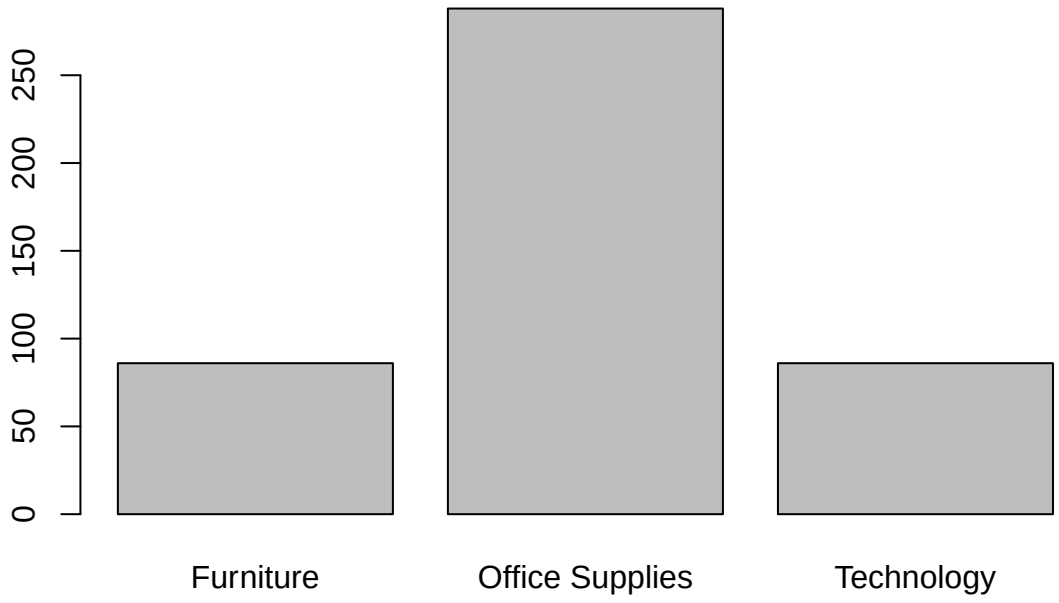
Create an alphabetically ordered barplot for each sales Category in the State of Texas.

```
# slice the data where the state is Texas

Texas_sales <- sales_pipe[sales_pipe$State == "Texas", ]

#change the category to factor type
Texas_sales$Category <- factor(Texas_sales$Category)

#create barplot of factor type category data
barplot(table(Texas_sales$Category))
```



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.**

```
#use aggregate to calculate the mean profit using region as the grouping elements
aggregate(sales_pipe$Profit, list(sales_pipe$Region), FUN=mean)
```

```
##   Group.1        x
## 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.**

```
# break the order.date into pieces and paste jsut the order year into the sales_pipe df
order_date_broken_out <- stringr::str_split_fixed(string=sales_pipe$Order.Date,pattern='-',n=3)

sales_pipe$order_year <- paste(order_date_broken_out[,1])

#use aggregate to calculate the mean profit using order year as the grouping elements
aggregate(sales_pipe$Profit, list(sales_pipe$order_year), FUN=mean)
```

```
##   Group.1        x
## 1    2014 32.24582
## 2    2015 21.58676
## 3    2016 30.10960
## 4    2017 21.31825
```