Week 2 Exercises

Joseph Annand

July 15, 2023

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

Place this at the top of your script:

Libraries

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

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

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

```
library(forcats)
```

Exercise 1

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

```
# Your code here
sales <- read.delim("Data/sales_pipe.txt"
                    ,fileEncoding="WINDOWS-1252"
                    ,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.

```r
# Your code here

# Create character vector of column names in sales
sales_colnames <- colnames(sales)
# Rename first column of sales data frame
sales[,1] = "Row.ID"
```

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

```r
# Your code here

# Convert ship dates to date-times using mdy()
sales$Ship.Date = mdy(sales[["Ship.Date"]])
# Convert order dates to date-times using mdy()
sales$Order.Date = mdy(sales[["Order.Date"]])

# Find difference between most recent order and oldest order in days
diff_days <- difftime(max(sales$Order.Date), min(sales$Order.Date), units="days")
diff_days
```

```
## Time difference of 1457 days
```

```r
# Calculate difference in years
diff_days / 365.25
```

```
## Time difference of 3.989049 days
```

```r
# Calculate difference in week
diff_weeks <- difftime(max(sales$Order.Date), min(sales$Order.Date), units="weeks")
diff_weeks
```

```
## Time difference of 208.1429 weeks
```

The number of days between the most recent order and the oldest order is 1457 days. The difference in years in 3.989049 years. The difference in weeks is 208.1429 weeks.

Exercise 4

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

```r
# Your code here
# Create column indicating amount of days between Ship.Date and Order.Date
sales$Ship.Time <- (sales$Ship.Date - sales$Order.Date)

# Calculate the average number of days to ship an order
mean(sales$Ship.Time)
```

```
## Time difference of 3.908482 days
```

The average number of days it takes to ship an order is 3.908482 days.

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.

```
# Your code here
# Split Customer.Name column into two columns for first and last name
sales[c("First.Name", "Last.Name")] <- stringr::str_split_fixed(string=sales$Customer.Name,pattern=' ',n

# Create a vector of all indices with the first name Bill
customers_bill <- grep("Bill", sales$First.Name)

# Return the length of the vector with all customers named Bill
length(customers_bill)
```

```
## [1] 37
```

There are 37 customers with the 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**

```
# Your code here
# Return the length of the vector of all product names that mention "table"
length(grep("table", sales$Product.Name))
```

```
## [1] 197
```

There are 197 mentions of the word 'table' in Product.Name column.

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.

```
# Your code here
# Create a frequency table for State column and convert it to a data frame
states_df <- as.data.frame(table(sales$State))
states_df
```

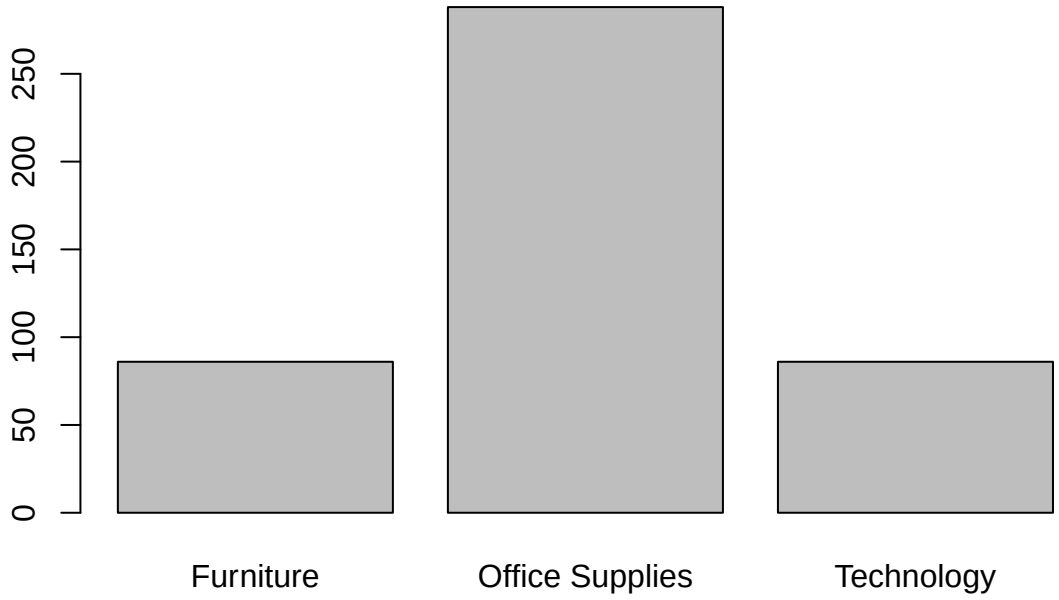
```
##                   Var1 Freq
## 1              Alabama   28
## 2              Arizona  119
## 3             Arkansas   22
## 4           California  993
## 5             Colorado   90
## 6          Connecticut   50
## 7             Delaware   47
## 8  District of Columbia    1
## 9              Florida  186
## 10             Georgia   79
## 11               Idaho    9
## 12            Illinois  286
## 13             Indiana   74
## 14                Iowa   11
## 15              Kansas   16
## 16            Kentucky   64
## 17           Louisiana   18
## 18               Maine    4
## 19            Maryland   63
## 20       Massachusetts   71
## 21            Michigan  142
## 22           Minnesota   41
## 23         Mississippi   27
## 24            Missouri   37
## 25             Montana    2
## 26            Nebraska   26
## 27              Nevada   24
## 28       New Hampshire    9
## 29          New Jersey   58
## 30          New Mexico   11
## 31            New York  555
## 32      North Carolina  117
## 33        North Dakota    7
## 34                Ohio  211
## 35            Oklahoma   38
## 36              Oregon   56
## 37        Pennsylvania  312
## 38        Rhode Island   25
## 39      South Carolina   28
## 40        South Dakota    9
## 41           Tennessee   88
## 42               Texas  460
## 43                Utah   27
## 44             Vermont   10
## 45            Virginia   80
## 46          Washington  254
## 47       West Virginia    4
## 48           Wisconsin   38
## 49             Wyoming    1
```

Exercise 8

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

```
# Your code here
# Create data frame with all observations where state is Texas
texas_df <- subset(sales, State=="Texas")

# Convert values of Category column for texas_df to factors and create frequency table
texas_categories <- table(factor(texas_df$Category))
# Plot texas_categories frequency table in a bar chart
barplot(texas_categories)
```



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.**

```
# Your code here
# Determine the mean of profits grouped by region
aggregate(sales$Profit, list(sales$Region), 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.**

```
# Your code here
# Create a column representing the year the order was placed
sales$Order.Year <- year(sales$Order.Date)
# Determine the mean of profits by order year
aggregate(sales$Profit, list(sales$Order.Year), mean)
```

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