Week 2 Exercises

Hunter Huberdeau

October 27, 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)
```

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

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

```
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"
                    ,stringsAsFactors=FALSE
                    ,sep = "|"
                    ,fileEncoding="WINDOWS-1252"
 )
```

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.

```
# Your code here
colnames(sales)[1] <- "Row.ID"
colnames(sales)[1]
```

```
## [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

```
# Your code here
sales$Ship.Date <- as.Date(sales$Ship.Date,format='%B %d %Y')
sales$Order.Date <- as.Date(sales$Order.Date,format='%m/%d/%Y')

recent_order<- max(sales$Order.Date)
oldest_order<-min(sales$Order.Date)
order_diff<-recent_order-oldest_order
order_diff
```

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

```
week_diff<- as.numeric(difftime(recent_order, oldest_order, units = 'weeks'))
print(paste('Time difference of', week_diff, 'weeks'))
```

```
## [1] "Time difference of 208.142857142857 weeks"
```

```
year_diff <- as.period(interval(recent_order, oldest_order))/years(-1)
```

```
## estimate only: convert to intervals for accuracy
```

```
print(paste('Time difference of', year_diff, 'years'))
```

```
## [1] "Time difference of 3.99058863791923 years"
```

Exercise 4

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

```
# Your code here
mean(sales$Ship.Date-sales$Order.Date)
```

```
## Time difference of 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 full name count 'bill' in new column
split_name <- stringr::str_split_fixed(string=sales$Customer.Name,pattern=' ',n=2)
sales$first_name <- split_name[,1]
name_table<-table(sales$first_name)
bill_count <- name_table["Bill"]
bill_count
```

```
## Bill
##   37
```

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
table_count <- sum(str_count(sales$Product.Name,"table"))
table_count
```

```
## [1] 240
```

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
state_table <- table(sales$State)
state_table
```

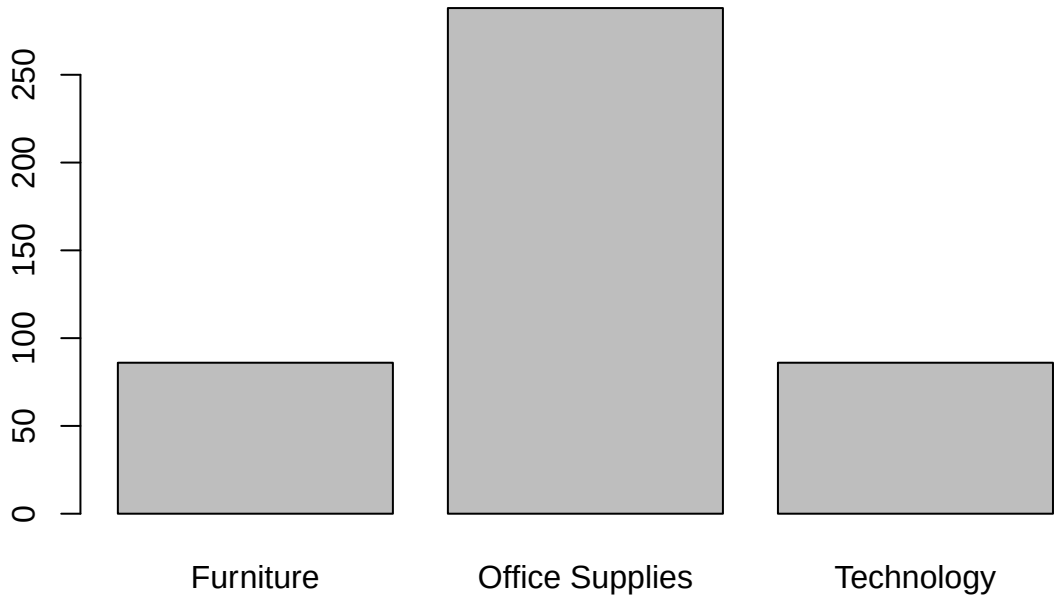
```
##
##              Alabama              Arizona             Arkansas
##                   28                  119                   22
##           California             Colorado          Connecticut
##                  993                   90                   50
##             Delaware District 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.

```
# Your code here

Texas_sales_df = sales[(sales$State=='Texas'), ]
barplot(table(Texas_sales_df$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.**

```
# Your code here
aggregate(sales$Profit, list(sales$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.**

```
# Your code here
order_year <- stringr::str_split_fixed(string=sales$Order.Date,pattern='-',n=3)
```

```r
sales$order_year <- order_year[,1]

aggregate(sales$Profit, list(sales$order_year), FUN = mean)
```

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