Week 2 Exercises

Christopher Rodgers

March 22 2024

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)

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 4.3.3
```

```
library(forcats)
```

```
## Warning: package 'forcats' was built under R version 4.3.3
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.3.3
```

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

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

Exercise 1

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

```
# Your code here
sales <- read.delim("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"
```

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")
```

```
diff_days <- max(sales$Order.Date) - min(sales$Order.Date)
```

```
print(diff_days)
```

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

```
print(paste(as.duration(diff_days) %/% as.duration(years(1)), "years"))
```

```
## [1] "3 years"
```

```
(paste(as.duration(diff_days) %/% as.duration(weeks(1)), "weeks"))
```

```
## [1] "208 weeks"
```

Exercise 4

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

```
# Your code here
sales$difference <- (sales$Ship.Date - sales$Order.Date)
result = as.numeric(mean(sales$difference))
print(result)
```

```
## [1] 3.908482
```

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
names = unique(sales$Customer.Name)
length(grep("Bill", names))
```

```
## [1] 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**

```
# Your code here
length(grep("table", sales$Product.Name))
```

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

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
as.data.frame(table(sales$State))
```

```
##                    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

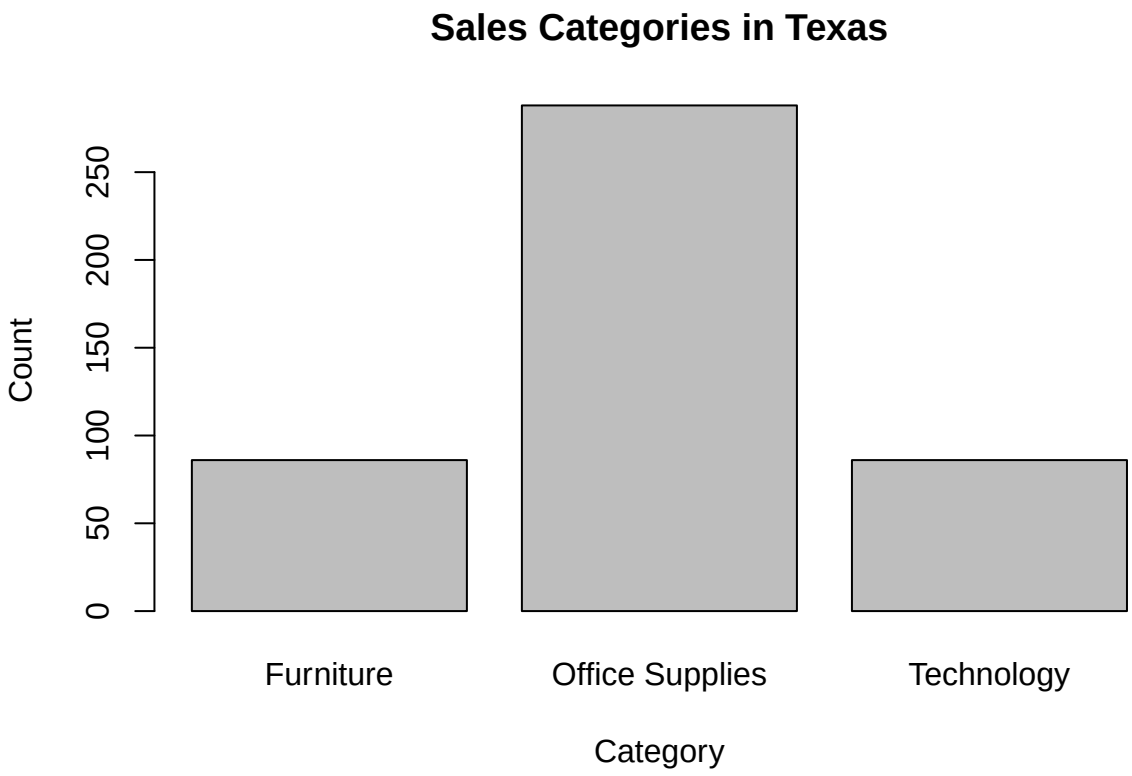
Create an alphabetically ordered barplot for each sales Category in the State of Texas.

```
# Your code here
sales_in_texas <- subset(sales, State == "Texas")
category_counts <- table(sales_in_texas$Category)
sales_in_texas$Category <- fct_reorder(factor(sales_in_texas$Category), as.numeric(sales_in_texas$Categ
```

```
## Warning in stopifnot(length(f) == length(.x)): NAs introduced by coercion
```

```
## Warning: `fct_reorder()` removing 460 missing values.
## i Use `.na_rm = TRUE` to silence this message.
## i Use `.na_rm = FALSE` to preserve NAs.
```

```
barplot(category_counts, main = "Sales Categories in Texas",
        xlab = "Category",
        ylab = "Count")
```



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$Profit ~ sales$Region, sales, mean)
```

```
##   sales$Region sales$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.**

```
# Your code here
sales$Order.Year = format(sales$Order.Date, format = "%Y")
```

```r
aggregate(sales$Profit ~ sales$Order.Year, sales, mean)
```

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