Week 2 Exercises

Larry Taylor

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: library(stringr) library(lubridate) 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 = "|"
                    )
```

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

```
library(lubridate)
```

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

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

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


sales$Ship.Date <- mdy(sales$Ship.Date)
```

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

num_of_weeks <- days(num_of_days)/weeks(1)

## estimate only: convert to intervals for accuracy
num_of_years <- days(num_of_days)/years(1)

## estimate only: convert to intervals for accuracy
print(num_of_days)

## Time difference of 1457 days
print(num_of_weeks)

## [1] 208.1429
print(num_of_years)

## [1] 3.989049
```

Exercise 4

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

```r
sales$days_to_ship_order <- difftime(sales$Ship.Date, sales$Order.Date, units='days')

mean(sales$days_to_ship_order)

## 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.

```r
library(stringr)


sales$first_name <- str_split_fixed(string=sales$Customer.Name, pattern=" ", n=2)

length(which(sales$first_name=="Bill"))

## [1] 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**

```r
library(stringr)

length(str_which(sales$Product.Name, "table", negate=FALSE))

## [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
library(forcats)



fct_count(sales$State, sort=FALSE,prop=FALSE)
```

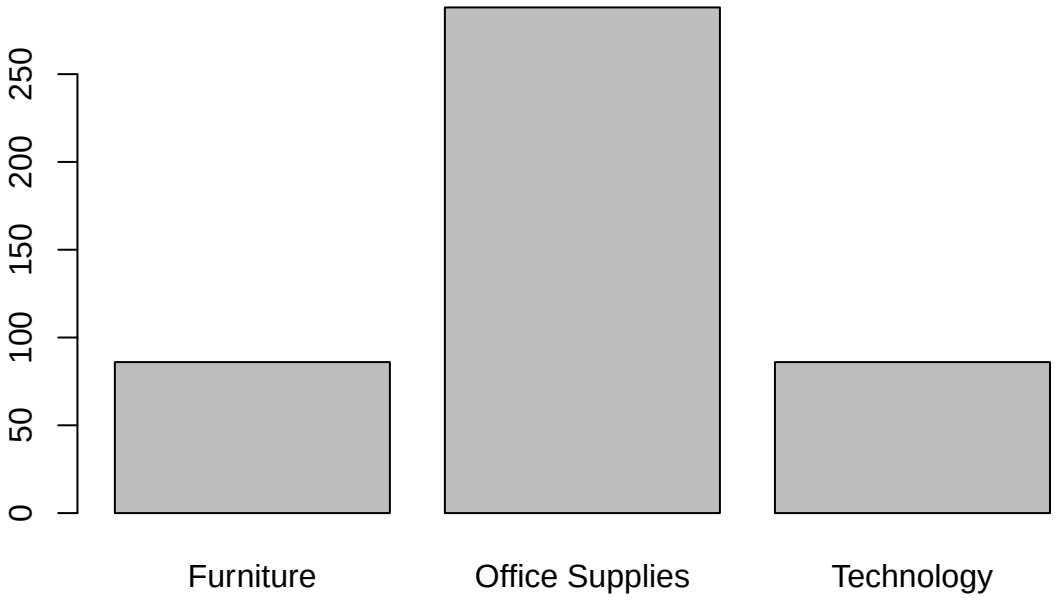
```
## # A tibble: 49 x 2
##    f                        n
##    <fct>                <int>
##  1 Alabama                 28
##  2 Arizona                119
##  3 Arkansas                22
##  4 California             993
##  5 Colorado                90
##  6 Connecticut             50
##  7 Delaware                47
##  8 District of Columbia     1
##  9 Florida                186
## 10 Georgia                 79
## # ... with 39 more rows
```

Exercise 8

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

```
# Your code here
texas_only <- sales[sales$State=="Texas",]

barplot(table(factor(texas_only$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$Profit, list(Region = sales$Region), mean)
```

```
##    Region        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
aggregate(sales$Profit, list(Region = year(sales$Order.Date)), mean)
```

```
##   Region        x
## 1   2014 32.24582
## 2   2015 21.58676
## 3   2016 30.10960
```

```
## 4   2017 21.31825
```