# Week 2 Exercises

James Clark

March 2022

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)
```

```
## Loading required package: timechange

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

## 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_df)[1] <- "Row.ID"
colnames(sales_df)
```

```
##  [1] "Row.ID"       "Order.ID"      "Order.Date"    "Ship.Date"
##  [5] "Ship.Mode"     "Customer.ID"   "Customer.Name" "Segment"
##  [9] "Country"       "City"          "State"         "Postal.Code"
## [13] "Region"        "Product.ID"    "Category"      "Sub.Category"
## [17] "Product.Name"  "Sales"         "Quantity"      "Discount"
## [21] "Profit"
```

# Exercise 3

Convert both Order.ID 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_df$Order.Date <- mdy(sales_df$Order.Date)

# max(sales_df$Order.Date)-min(sales_df$Order.Date) #number
# of days between most recent and oldest orders

order_minmax_length_days <- interval(min(sales_df$Order.Date),
    max(sales_df$Order.Date))/days(1)  #number of days between most recent and oldest
    ↪  orders again
order_minmax_length_years <- interval(min(sales_df$Order.Date),
    max(sales_df$Order.Date))/years(1)  #exact number of years between most recent and
    ↪  oldest orders
order_minmax_length_weeks <- interval(min(sales_df$Order.Date),
    max(sales_df$Order.Date))/weeks(1)  #exact number of weeks between most recent and
    ↪  oldest orders

approx_order_minmax_length_years <- round(order_minmax_length_years)  #approximate number
↪  of years between most recent and oldest orders
approx_order_minmax_length_weeks <- round(order_minmax_length_weeks)  #approximate number
↪  of weeks between most recent and oldest orders

cat("The number of days between the most recent and oldest order is",
    order_minmax_length_days, "days.\nThe number of years between the most recent and
↪   oldest order is",
    order_minmax_length_years, "years (or approximately",
↪   approx_order_minmax_length_years,
    "years).\nThe number of weeks between the most recent and oldest order is",
    order_minmax_length_weeks, "weeks (or approximately",
↪   approx_order_minmax_length_weeks,
    "weeks).")
```

```
## The number of days between the most recent and oldest order is 1457 days.
## The number of years between the most recent and oldest order is 3.989041 years (or
↪  approximately 4 years).
## The number of weeks between the most recent and oldest order is 208.1429 weeks (or
↪  approximately 208 weeks).
```

# Exercise 4

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

```
# Your code here
sales_df$Ship.Date <- mdy(sales_df$Ship.Date)

mean_order_to_ship_length <- mean(interval(sales_df$Order.Date,
    sales_df$Ship.Date)/days(1))

cat("The average number of days it takes to ship an order is",
    mean_order_to_ship_length, "(or roughly", round(mean_order_to_ship_length),
    "days).")
```

```
## The average number of days it takes to ship an order is 3.908482 (or roughly 4 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
temp_char <- str_split_fixed(string = sales_df$Customer.Name,
    pattern = " ", n = 2)
sales_df$Customer.First.Name <- temp_char[, 1]
sales_df$Customer.Last.Name <- temp_char[, 2]

length(unique(sales_df$Customer.Name[sales_df$Customer.First.Name ==
    "Bill"]))
```

```
## [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
sum(str_count(sales_df$Product.Name, "table"))
```

```
## [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.

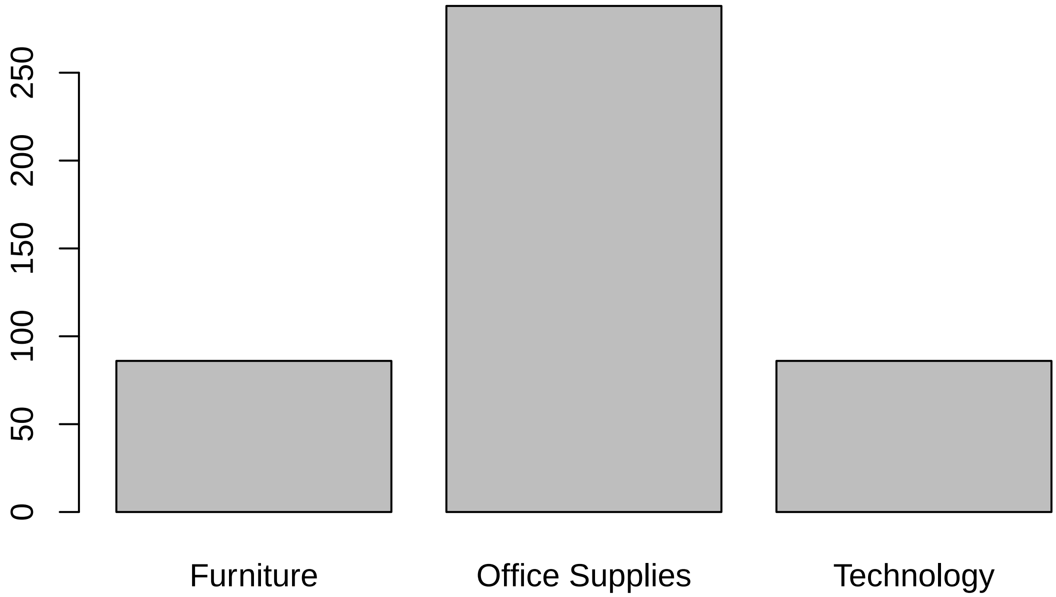
```
# Your code here
table(sales_df$State)
```

```
## 
##              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.

```
# Your code here
barplot(table(sales_df$Category[sales_df$State == "Texas"]))
```



## 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_df$Profit, list(sales_df$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
aggregate(sales_df$Profit, list(year(sales_df$Order.Date)), mean)
```

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