

Week 2 Exercises

Steven Simonsen

March 17, 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)

Exercise 1

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

```
library(stringr)
library(lubridate)

##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##      date, intersect, setdiff, union
library(forcats)
# 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 data frame to Row.ID.

Note: You will need to assign the first element of colnames to a single character.

```
colnames(sales)[1] <- c("Row.ID")
#Showing slice so my document doesn't span too many pages. See first
#column name change
sales[1:10 , 1:6]
```

##	Row.ID	Order.ID	Order.Date	Ship.Date	Ship.Mode	Customer.ID
## 1	1	CA-2016-152156	11/8/2016	November 11 2016	Second Class	CG-12520
## 2	2	CA-2016-152156	11/8/2016	November 11 2016	Second Class	CG-12520
## 3	3	CA-2016-138688	6/12/2016	June 16 2016	Second Class	DV-13045
## 4	4	US-2015-108966	10/11/2015	October 18 2015	Standard Class	SO-20335
## 5	5	US-2015-108966	10/11/2015	October 18 2015	Standard Class	SO-20335

```
## 6      6 CA-2014-115812 6/9/2014      June 14 2014 Standard Class BH-11710
## 7      7 CA-2014-115812 6/9/2014      June 14 2014 Standard Class BH-11710
## 8      8 CA-2014-115812 6/9/2014      June 14 2014 Standard Class BH-11710
## 9      9 CA-2014-115812 6/9/2014      June 14 2014 Standard Class BH-11710
## 10     10 CA-2014-115812 6/9/2014      June 14 2014 Standard Class BH-11710
```

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

```
sales$Order.Date <- as.Date(sales$Order.Date,format="%M/%d/%Y")
sales$Ship.Date <- as.Date(sales$Ship.Date,format="%B %d %Y")

max_order_date <- max(sales$Order.Date)
min_order_date <- min(sales$Order.Date)

years_between_orders <- time_length(difftime(max_order_date, min_order_date), "years")

days_between_orders <- time_length(difftime(max_order_date, min_order_date), "days")

weeks_between_orders <- time_length(difftime(max_order_date, min_order_date), "weeks")

print(years_between_orders)
```

```
## [1] 3.08282
```

```
print(days_between_orders)
```

```
## [1] 1126
```

```
print(weeks_between_orders)
```

```
## [1] 160.8571
```

Exercise 4

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

```
ship_order_diff <- sales$Ship.Date - sales$Order.Date

mean_ship_days <- time_length(mean(ship_order_diff), "days")

print(mean_ship_days)
```

```
## [1] 152.7946
```

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.

```
#It was hard for me to incorporate the length() function into this
#This isn't the way I would normally write the code, but to answer the question
#I wrote it this way.
```

```
#Split customer name into two columns
split_custname <- str_split_fixed(sales$Customer.Name," ",n=2)

#pull out the length of customer first names
firstname_length <- str_length(split_custname[,1])

#If the first name is equal to 4, then count the total number matched to the
#Pattern Bill. Wrap this entire piece of code in a sum function.
sum(ifelse(firstname_length == 4, str_count(split_custname[,1],"Bill"),0))
```

```
## [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**

```
sum(str_count(sales$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.

```
sales$State <- factor(sales$State)
is.factor(sales$State)
```

```
## [1] TRUE
```

```
levels(sales$State)
```

```
## [1] "Alabama"      "Arizona"      "Arkansas"
## [4] "California"   "Colorado"     "Connecticut"
## [7] "Delaware"     "District of Columbia" "Florida"
## [10] "Georgia"      "Idaho"        "Illinois"
## [13] "Indiana"      "Iowa"         "Kansas"
## [16] "Kentucky"     "Louisiana"    "Maine"
## [19] "Maryland"     "Massachusetts" "Michigan"
## [22] "Minnesota"    "Mississippi"  "Missouri"
## [25] "Montana"      "Nebraska"     "Nevada"
## [28] "New Hampshire" "New Jersey"   "New Mexico"
## [31] "New York"     "North Carolina" "North Dakota"
## [34] "Ohio"         "Oklahoma"     "Oregon"
## [37] "Pennsylvania" "Rhode Island"  "South Carolina"
## [40] "South Dakota" "Tennessee"    "Texas"
## [43] "Utah"         "Vermont"      "Virginia"
## [46] "Washington"   "West Virginia" "Wisconsin"
## [49] "Wyoming"
```

```
table(sales$State)
```

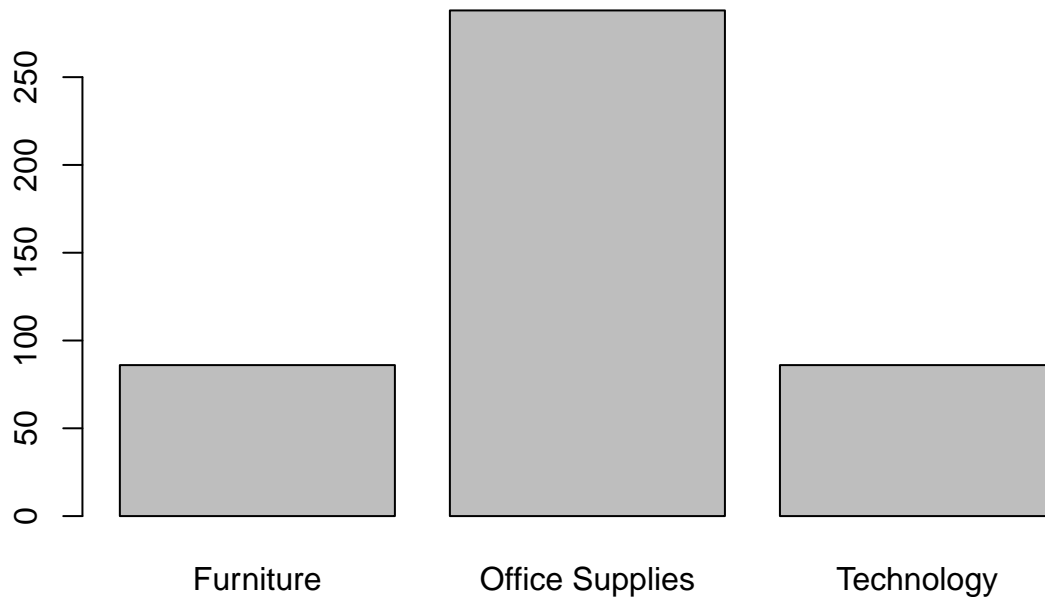
```
##
##      Alabama      Arizona      Arkansas
##      28          119          22
##      California    Colorado    Connecticut
##      993          90          50
##      Delaware District of Columbia    Florida
##      47           1          186
##      Georgia       Idaho       Illinois
##      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.

```
sales_texas <- subset(sales, State=="Texas")

barplot(table(sales_texas$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.**

```
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.**

```
order_year <- format.Date(sales$Order.Date, "%Y")
```

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

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

4 2017 21.31825