Week 2 Exercises

Auriana Anderson

October 28, 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)
```

```
## Warning: package 'stringr' was built under R version 4.4.1
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.4.1
```

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

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

```
library(forcats)
```

```
## Warning: package 'forcats' was built under R version 4.4.1
```

```
# setwd("C:/Users/aanderson/Desktop/Intro to R and Python/DES5002 Intro to R and Python")

sales <- read.delim("Week_2/Data/sales_pipe.txt",
                    stringsAsFactors=FALSE,
                     sep = "|", fileEncoding = "latin1")
```

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.

```
colnames(sales)
```

```
##  [1] "ï..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"
```

```
names(sales)[names(sales) == "ï..Row.ID"] <- "Row.ID"

head(sales)
```

```
##   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
##     Customer.Name   Segment       Country            City      State
## 1     Claire Gute  Consumer United States       Henderson   Kentucky
## 2     Claire Gute  Consumer United States       Henderson   Kentucky
## 3 Darrin Van Huff Corporate United States     Los Angeles California
## 4  Sean O'Donnell  Consumer United States Fort Lauderdale    Florida
## 5  Sean O'Donnell  Consumer United States Fort Lauderdale    Florida
## 6 Brosina Hoffman  Consumer United States     Los Angeles California
##   Postal.Code Region      Product.ID        Category Sub.Category
## 1       42420  South FUR-BO-10001798       Furniture    Bookcases
## 2       42420  South FUR-CH-10000454       Furniture       Chairs
## 3       90036   West OFF-LA-10000240 Office Supplies       Labels
## 4       33311  South FUR-TA-10000577       Furniture       Tables
## 5       33311  South OFF-ST-10000760 Office Supplies      Storage
## 6       90032   West FUR-FU-10001487       Furniture  Furnishings
##                                                    Product.Name    Sales
## 1                              Bush Somerset Collection Bookcase 261.9600
## 2     Hon Deluxe Fabric Upholstered Stacking Chairs, Rounded Back 731.9400
## 3       Self-Adhesive Address Labels for Typewriters by Universal  14.6200
## 4                  Bretford CR4500 Series Slim Rectangular Table 957.5775
## 5                                Eldon Fold 'N Roll Cart System  22.3680
## 6 Eldon Expressions Wood and Plastic Desk Accessories, Cherry Wood  48.8600
##   Quantity Discount    Profit
## 1        2     0.00   41.9136
## 2        3     0.00  219.5820
## 3        2     0.00    6.8714
```

```
## 4        5     0.45 -383.0310
## 5        2     0.20    2.5164
## 6        7     0.00   14.1694
```

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

sales$Order.Date <- mdy(sales$Order.Date)

oldest_order <- min(sales$Order.Date)

recent_order <-max(sales$Order.Date)

#What is the number of days between the most recent order and the oldest order?
days_between <- as.numeric(recent_order-oldest_order)

#How many years is that?
years_between <- days_between/365.25

#How many weeks?
weeks_between <- days_between/7

oldest_order
```

```
## [1] "2014-01-03"
```

```
recent_order
```

```
## [1] "2017-12-30"
```

```
days_between
```

```
## [1] 1457
```

```
years_between
```

```
## [1] 3.989049
```

```
weeks_between
```

```
## [1] 208.1429
```

Exercise 4

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

```
shiptime <-as.numeric(sales$Ship.Date - sales$Order.Date)

avg_shiptime <- mean(shiptime)
avg_shiptime
```

```
## [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.

```
name_split <- str_split(string = sales$Customer.Name, pattern = " ")

sales$first_name <- sapply(name_split, "[", 1)

number_of_bills <- sum(length(which(sales$first_name == "Bill")))
number_of_bills
```

```
## [1] 37
```

Exercise 6

How many mentions of the word 'table' are there in the Product.Name column?

There are zero occurrences of table.

There are 230 occurrences of Table.

capitalization seems to make a difference here.

Note you can do this in one line of code

```
#I looked this up and found in order to find just the word table without
#it being apart of a longer word and having exact matches you must use the
#regex \\b on either side of the word

sum(str_count(sales$Product.Name, "\\bTable\\b"))
```

```
## [1] 230
```

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.

```
count_state <-as.data.frame(table(sales$State))

colnames(count_state) <- c("State","Count")

count_state <- count_state[order(count_state$State),]

head(count_state)
```

```
##         State Count
## 1     Alabama    28
## 2     Arizona   119
## 3    Arkansas    22
## 4  California   993
## 5    Colorado    90
## 6 Connecticut    50
```

Exercise 8

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

```
texas <- sales[sales$State == "Texas",]

head(texas)
```

```
##     Row.ID       Order.ID Order.Date  Ship.Date      Ship.Mode Customer.ID
## 15      15 US-2015-118983 2015-11-22 2015-11-26 Standard Class    HP-14815
## 16      16 US-2015-118983 2015-11-22 2015-11-26 Standard Class    HP-14815
## 78      78 US-2017-118038 2017-12-09 2017-12-11    First Class    KB-16600
## 79      79 US-2014-147606 2014-11-26 2014-12-01   Second Class    JE-15745
## 89      89 CA-2016-159695 2016-04-05 2016-04-10   Second Class    GM-14455
## 345    345 US-2015-120712 2015-12-20 2015-12-24 Standard Class    CS-12130
##     Customer.Name     Segment       Country       City State Postal.Code
## 15  Harold Pawlan Home Office United States Fort Worth Texas       76106
## 16  Harold Pawlan Home Office United States Fort Worth Texas       76106
## 78    Ken Brennan   Corporate United States    Houston Texas       77041
## 79     Joel Eaton    Consumer United States    Houston Texas       77070
## 89   Gary Mitchum Home Office United States    Houston Texas       77095
## 345  Chad Sievert    Consumer United States     Austin Texas       78745
##      Region      Product.ID        Category Sub.Category
## 15  Central OFF-AP-10002311 Office Supplies   Appliances
## 16  Central OFF-BI-10000756 Office Supplies      Binders
## 78  Central OFF-ST-10000615 Office Supplies      Storage
## 79  Central FUR-FU-10003194       Furniture  Furnishings
## 89  Central OFF-ST-10003442 Office Supplies      Storage
## 345 Central OFF-ST-10000107 Office Supplies      Storage
##                                                                Product.Name
```

```
## 15  Holmes Replacement Filter for HEPA Air Cleaner, Very Large Room, HEPA Filter
## 16                              Storex DuraTech Recycled Plastic Frosted Binders
## 78            SimpliFile Personal File, Black Granite, 15w x 6-15/16d x 11-1/4h
## 79                    Eldon Expressions Desk Accessory, Wood Pencil Holder, Oak
## 89                                                 Eldon Portable Mobile Manager
## 345                                                   Fellowes Super Stor/Drawer
##       Sales Quantity Discount    Profit first_name
## 15   68.810        5      0.8 -123.8580     Harold
## 16    2.544        3      0.8   -3.8160     Harold
## 78   27.240        3      0.2    2.7240        Ken
## 79   19.300        5      0.6  -14.4750       Joel
## 89  158.368        7      0.2   13.8572       Gary
## 345  88.800        4      0.2   -2.2200       Chad
```

```
texas_count <- table(texas$Category)

texas_count <- as.data.frame(texas_count)
```

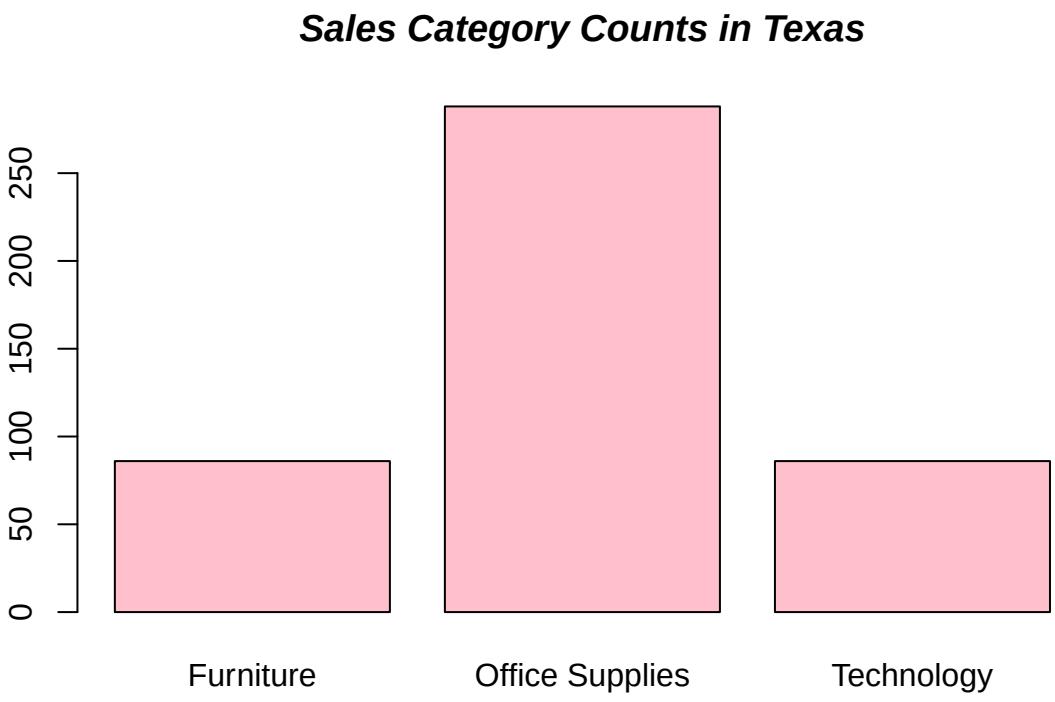
```
colnames(texas_count) <- c("Category","Count")

texas_count <- texas_count[order(names(texas_count))]

texas_count
```

```
##          Category Count
## 1       Furniture    86
## 2 Office Supplies   288
## 3      Technology    86
```

```
barplot(texas_count$Count,
        names.arg = texas_count$Category,
        col = "pink",
        main = "Sales Category Counts in Texas"
        ,font.main = 4)
```



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(x = list(Avg_Profit = sales$Profit), by = list(Region = sales$Region), FUN = "mean")
```

```
##    Region Avg_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.**

```
date_split <-  str_split(string = sales$Order.Date, pattern = "-")

sales$order_year <- sapply(date_split, "[", 1)

aggregate(x = list(Avg_Profit = sales$Profit), by = list(order_year = sales$order_year), FUN = "mean")
```

```
##   order_year Avg_Profit
## 1       2014   32.24582
## 2       2015   21.58676
```

```
## 3       2016   30.10960
## 4       2017   21.31825
```