Ex4: Data Wrangling

Wassnaa Al-Mawee

01/31/2024

cran\_mirror <- "https://cran.rstudio.com/"  
options(repos = c(CRAN = cran\_mirror))  
install.packages("tidyr")

##   
## The downloaded binary packages are in  
## /var/folders/q\_/sg2ykfgs57z4jr3mr195l43c0000gn/T//RtmpiaSUhI/downloaded\_packages

## Exploring data sets: using dplyr on external data

* An important part about being a data scientist is asking questions. Write a question you may be interested in about the data set data/nba\_teams\_2016.csv, and then use dplyr to figure out the answer!
* Load the dplyr library

install.packages("dplyr")

##   
## The downloaded binary packages are in  
## /var/folders/q\_/sg2ykfgs57z4jr3mr195l43c0000gn/T//RtmpiaSUhI/downloaded\_packages

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

* Use the read.csv() function to read in the included data set. Remember to save it as a variable.

nba\_data <- read.csv("nba\_teams\_2016.csv")

* View the data frame you loaded, and get some basic information about the number of rows/columns.Note the “X” preceding some of the column titles as well as the “\*” following the names of teams that made it to the playoffs that year.

str(nba\_data)

## 'data.frame': 30 obs. of 26 variables:  
## $ Rk : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Team: chr "Golden State Warriors\*" "Oklahoma City Thunder\*" "Sacramento Kings" "Houston Rockets\*" ...  
## $ G : int 82 82 82 82 82 82 82 82 82 82 ...  
## $ MP : int 19880 19830 19805 19830 19780 19805 19830 19855 19755 19705 ...  
## $ FG : int 3489 3372 3283 3094 3216 3167 3141 3171 3238 3289 ...  
## $ FGA : int 7159 7082 7083 6847 7318 7040 6759 6888 7033 6797 ...  
## $ FG. : num 0.487 0.476 0.464 0.452 0.439 0.45 0.465 0.46 0.46 0.484 ...  
## $ X3P : int 1077 678 660 878 717 864 797 880 709 570 ...  
## $ X3PA: int 2592 1945 1839 2533 2142 2336 2190 2428 1983 1518 ...  
## $ X3P.: num 0.416 0.349 0.359 0.347 0.335 0.37 0.364 0.362 0.358 0.375 ...  
## $ X2P : int 2412 2694 2623 2216 2499 2303 2344 2291 2529 2719 ...  
## $ X2PA: int 4567 5137 5244 4314 5176 4704 4569 4460 5050 5279 ...  
## $ X2P.: num 0.528 0.524 0.5 0.514 0.483 0.49 0.513 0.514 0.501 0.515 ...  
## $ FT : int 1366 1616 1514 1671 1520 1424 1490 1333 1349 1342 ...  
## $ FTA : int 1790 2067 2089 2407 1929 1889 2152 1783 1849 1672 ...  
## $ FT. : num 0.763 0.782 0.725 0.694 0.788 0.754 0.692 0.748 0.73 0.803 ...  
## $ ORB : int 816 1071 868 930 950 948 721 873 743 770 ...  
## $ DRB : int 2972 2916 2760 2601 2733 2782 2727 2777 2688 2831 ...  
## $ TRB : int 3788 3987 3628 3531 3683 3730 3448 3650 3431 3601 ...  
## $ AST : int 2373 1883 2009 1821 1981 1748 1873 1861 2005 2010 ...  
## $ STL : int 689 603 733 821 752 562 709 551 708 677 ...  
## $ BLK : int 498 487 368 430 348 380 460 317 323 485 ...  
## $ TOV : int 1245 1305 1326 1307 1127 1200 1063 1114 1186 1071 ...  
## $ PF : int 1701 1691 1676 1790 1796 1782 1746 1666 1708 1433 ...  
## $ PTS : int 9421 9038 8740 8737 8669 8622 8569 8555 8534 8490 ...  
## $ PS.G: num 115 110 107 106 106 ...

* Add a column that gives the turnovers to steals ratio (TOV / STL) for each team.

nba\_data <- nba\_data %>%  
 mutate(TOV\_STL\_Ratio = TOV / STL)

* Sort the teams from lowest turnover/steal ratio to highest. Which team has the lowest turnover/steal ratio?

lowest\_ratio\_team <- nba\_data %>%  
 arrange(TOV\_STL\_Ratio) %>%  
 slice(1)

* Using the pipe operator, create a new column of assists per game (AST / G), AND sort the data.frame by this new column in descending order.

nba\_data <- nba\_data %>%  
 mutate(AST\_per\_Game = AST / G) %>%  
 arrange(desc(AST\_per\_Game))

* Create a data frame called good\_offense of teams that scored more than 8700 points (PTS) in the season

good\_offense <- nba\_data %>%  
 filter(PTS > 8700)

* Create a data frame called good\_defense of teams that had more than 470 blocks (BLK)

good\_defense <- nba\_data %>%  
 filter(BLK > 470)

* Create a data frame called offense\_stats that only shows offensive rebounds (ORB), field-goal % (FG.), and assists (AST) along with the team name.

offense\_stats <- nba\_data %>%  
 select(Team, ORB, FG, AST)

* Create a data frame called defense\_stats that only shows defensive rebounds (DRB), steals (STL), and blocks (BLK) along with the team name.

defense\_stats <- nba\_data %>%  
 select(Team, DRB, STL, BLK)

* Create a function called better\_shooters that takes in two teams and returns a data frame of the team with the better field-goal percentage. Include the team name, field-goal percentage, and total points in your resulting data frame.

better\_shooters <- function(team1, team2) {  
 teams <- nba\_data %>%  
 filter(Team %in% c(team1, team2)) %>%  
 select(Team, FG, PTS)  
   
 better\_team <- teams %>%  
 arrange(desc(FG)) %>%  
 slice(1)  
   
 return(better\_team)  
}

* Call the function on two teams to compare them (remember the \* if needed)

comparison\_result <- better\_shooters("Team1\*", "Team2\*")

## Analyzing data with the tidyr package

* Load necessary packages (tidyr, dplyr, and ggplot2)

install.packages("tidyr")

##   
## The downloaded binary packages are in  
## /var/folders/q\_/sg2ykfgs57z4jr3mr195l43c0000gn/T//RtmpiaSUhI/downloaded\_packages

install.packages("ggplot2")

##   
## The downloaded binary packages are in  
## /var/folders/q\_/sg2ykfgs57z4jr3mr195l43c0000gn/T//RtmpiaSUhI/downloaded\_packages

library(tidyr)  
library(ggplot2)

* Load the data/avocado.csv file into a variable avocados. Make sure strings are *not* read in as factors

avocados <- read.csv("avocado.csv", stringsAsFactors = FALSE)

* To tell R to treat the Date column as a date (not just a string). Redefine that column as a date using the as.Date() function (hint: use the mutate function)

avocados <- avocados %>%  
 mutate(Date = as.Date(Date))

* The file had some uninformative column names, so rename these columns: – X4046 to small\_haas – X4225 to large\_haas – X4770 to xlarge\_haas

avocados <- avocados %>%  
 rename(small\_haas = X4046, large\_haas = X4225, xlarge\_haas = X4770)

* The data only has sales for haas avocados. Create a new column other\_avos that is the Total. Volume minus all haas avocados (small, large, xlarge)

avocados <- avocados %>%  
 mutate(other\_avos = Total.Volume - small\_haas - large\_haas - xlarge\_haas)

* To perform analysis by avocado size, create a dataframe by\_size that has only Date, other\_avos, small\_haas, large\_haas, xlarge\_haas

by\_size <- avocados %>%  
 select(Date, other\_avos, small\_haas, large\_haas, xlarge\_haas)

* In order to visualize this data, it needs to be reshaped. The four columns other\_avos, small\_haas, large\_haas, xlarge\_haas need to be **gathered** together into a single column called size. The volume of sales (currently stored in each column) should be stored in a new column called volume. Create a new dataframe size\_gathered by passing the by\_size data frame to the gather() function. size\_gathered will only have 3 columns: Date, size, and volume.

size\_gathered <- by\_size %>%  
 gather(key = "size", value = "volume", -Date)

* Using size\_gathered, compute the average sales volume of each size (hint, first group\_by size, then compute using summarize)

avg\_sales\_by\_size <- size\_gathered %>%  
 group\_by(size) %>%  
 summarize(avg\_volume = mean(volume))

* Investigate sales by avocado type (conventional, organic). Create a new data frame by\_type by grouping the avocados dataframe by Date and type, and calculating the sum of the Total.Volume for that type in that week (resulting in a data frame with 2 rows per week).

by\_type <- avocados %>%  
 group\_by(Date, type) %>%  
 summarize(total\_volume = sum(Total.Volume), .groups = "drop")