Homework #1

Gbemisola Talabi

16 September 2022

Instructions

1. Please submit your knitted `.pdf` file to the assignment drop box on eLC. If you are still having trouble knitting your file you can submit your `.Rmd` file.
2. All assignments are due by September 20, 2022 by 7:00pm EST. This assignment will be graded for accuracy. Please reach out to us if you need help before this time!
3. Please add your name as "author" to the YAML header above.
4. Below each question is a `r` code chunk that can be used to explore the question. Use the space below the code chunk to directly answer the question.

```
## you can add more, or change...these are suggestions
library(tidyverse)
library(readr)
library(dplyr)
```

Problem Set

1. (a) Make an object "age". Assign it your age in years. (b) Make another object "name". Assign it your name. Make sure to use quotation marks for anything with text!

```
age <- 24
name <- "Gbemisola Talabi"
```

2. Make an object "me" that is "age" and "name" combined.

```
me <- c(age, name)
```

3. Determine the data class for "me".

```
class(me)
```

```
## [1] "character"
```

4. If I want to do `me / 2` I get the following error: `Error in me/2 : non-numeric argument to binary operator`. Why? Write your answer as a comment inside the R chunk below.

```
# because me is a character, it is not numeric
```

The following questions involve an outside dataset.

We will be working with a dataset from the "Kaggle" website, which hosts competitions for prediction and machine learning. More details on this dataset are here: https://www.kaggle.com/c/DontGetKicked/overview/background.

5. Import the dataset into R. The dataset is located on eLC, "kaggleCarAuction.csv" Once you get the file, read the dataset in using `read_csv()` and assign it the name "cars".

```
cars <- read_csv("../data/kaggleCarAuction.csv")
```

```
## Rows: 72983 Columns: 34
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (24): PurchDate, Auction, Make, Model, Trim, SubModel, Color, Transmissi...
## dbl (10): RefId, IsBadBuy, VehYear, VehicleAge, VehOdo, BYRNO, VNZIP1, VehBC...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

6. Download the data dictionary from eLC, "Carvana_Data_Dictionary.txt" Open the file and determine the deliminator. Use the `read_delim()` function to import the file into R and assign it the name "key".

```
key <- read_delim("../data/Carvana_Data_Dictionary.txt", delim=";")
```

```
## Rows: 36 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ";"
## chr (2): Field Name, Definition
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

7. R can save individual variables as `.rds` files that can be imported again later. Save the "cars" data in an .rds file using the `write_rds()` function. You can choose what the `file=` argument is.

```
write_rds(cars, file ="my_cars.rds")
```

8. You should now be ready to work with the "cars" dataset.

(a) Preview the data so that you can see the names of the columns. There are several possible functions to do this.
(b) What is the class (data type) of the first three columns using `spec()` or `glimpse()`.

```
colnames(cars)
```

```
##  [1] "RefId"                           "IsBadBuy"
##  [3] "PurchDate"                       "Auction"
##  [5] "VehYear"                         "VehicleAge"
```

```
##  [7] "Make"                                "Model"
##  [9] "Trim"                                "SubModel"
## [11] "Color"                               "Transmission"
## [13] "WheelTypeID"                         "WheelType"
## [15] "VehOdo"                              "Nationality"
## [17] "Size"                                "TopThreeAmericanName"
## [19] "MMRAcquisitionAuctionAveragePrice" "MMRAcquisitionAuctionCleanPrice"
## [21] "MMRAcquisitionRetailAveragePrice"  "MMRAcquisitonRetailCleanPrice"
## [23] "MMRCurrentAuctionAveragePrice"     "MMRCurrentAuctionCleanPrice"
## [25] "MMRCurrentRetailAveragePrice"      "MMRCurrentRetailCleanPrice"
## [27] "PRIMEUNIT"                           "AUCGUART"
## [29] "BYRNO"                               "VNZIP1"
## [31] "VNST"                                "VehBCost"
## [33] "IsOnlineSale"                        "WarrantyCost"
```

```
spec(cars)
```

```
## cols(
##   RefId = col_double(),
##   IsBadBuy = col_double(),
##   PurchDate = col_character(),
##   Auction = col_character(),
##   VehYear = col_double(),
##   VehicleAge = col_double(),
##   Make = col_character(),
##   Model = col_character(),
##   Trim = col_character(),
##   SubModel = col_character(),
##   Color = col_character(),
##   Transmission = col_character(),
##   WheelTypeID = col_character(),
##   WheelType = col_character(),
##   VehOdo = col_double(),
##   Nationality = col_character(),
##   Size = col_character(),
##   TopThreeAmericanName = col_character(),
##   MMRAcquisitionAuctionAveragePrice = col_character(),
##   MMRAcquisitionAuctionCleanPrice = col_character(),
##   MMRAcquisitionRetailAveragePrice = col_character(),
##   MMRAcquisitonRetailCleanPrice = col_character(),
##   MMRCurrentAuctionAveragePrice = col_character(),
##   MMRCurrentAuctionCleanPrice = col_character(),
##   MMRCurrentRetailAveragePrice = col_character(),
##   MMRCurrentRetailCleanPrice = col_character(),
##   PRIMEUNIT = col_character(),
##   AUCGUART = col_character(),
##   BYRNO = col_double(),
##   VNZIP1 = col_double(),
##   VNST = col_character(),
##   VehBCost = col_double(),
##   IsOnlineSale = col_double(),
##   WarrantyCost = col_double()
## )
```

```
# RefId = col_double()
# IsBadBuy = col_double()
# PurchDate = col_character()
```

9. How many cars (rows) are in the dataset? How many variables (columns) are recorded for each car?

```
dim(cars)
```

```
## [1] 72983    34
```

10. Let's reduce the number of variables in the dataset to only those that will be used for the remainder of the exercises in order to make it slightly easier to work with. Keep the following variables (Model, Make, Color, VehOdo, VehicleAge, VehYear, VehBCost, Transmission) and reassign the new dataset to "cars". How many variables (columns) are left for each car?

```
cars <- select(cars, c(Model, Make, Color, VehOdo, VehicleAge, VehYear, VehBCost, Transmission))
ncol(cars)
```

```
## [1] 8
```

11. Remove any vehicles that at the time of purchase cost less than or equal to $5000. To do this first identify the variable using "key" that represents the acquisition cost paid for the vehicle at time of purchase, then filter based on this variable. Reassign the new filtered dataset to "cars". How many vehicles are left after filtering?

```
key
```

```
## # A tibble: 36 x 2
##    `Field Name` Definition
##    <chr>        <chr>
##  1 RefId        Unique (sequential) number assigned to vehicles
##  2 IsBadBuy     Identifies if the kicked vehicle was an avoidable purchase
##  3 PurchDate    The Date the vehicle was Purchased at Auction
##  4 Auction      Auction provider at which the  vehicle was purchased
##  5 VehYear      The manufacturer's year of the vehicle
##  6 VehicleAge   The Years elapsed since the manufacturer's year
##  7 Make         Vehicle Manufacturer
##  8 Model        Vehicle Model
##  9 Trim         Vehicle Trim Level
## 10 SubModel     Vehicle Submodel
## # ... with 26 more rows
## # i Use `print(n = ...)` to see more rows
```

```
cars <- filter(cars, VehBCost > 5000)
glimpse(cars)
```

```
## Rows: 59,957
## Columns: 8
## $ Model        <chr> "MAZDA3", "1500 RAM PICKUP 2WD", "GALANT 4C", "SPECTRA", ~
## $ Make         <chr> "MAZDA", "DODGE", "MITSUBISHI", "KIA", "FORD", "GMC", "FO~
```

```
## $ Color        <chr> "RED", "WHITE", "WHITE", "BLACK", "RED", "SILVER", "WHITE~
## $ VehOdo       <dbl> 89046, 93593, 81054, 49921, 84872, 80080, 75419, 79315, 7~
## $ VehicleAge   <dbl> 3, 5, 5, 2, 2, 4, 8, 4, 4, 3, 4, 4, 5, 7, 3, 2, 7, 5, 5, ~
## $ VehYear      <dbl> 2006, 2004, 2004, 2007, 2007, 2005, 2001, 2005, 2005, 200~
## $ VehBCost     <dbl> 7100, 7600, 5600, 5600, 7700, 5500, 5300, 5400, 7800, 690~
## $ Transmission <chr> "AUTO", "AUTO", "AUTO", "AUTO", "AUTO", "AUTO", "MANUAL",~
```

```
nrow(cars)
```

```
## [1] 59957
```

12. From this point on, work with the filtered "cars" dataset from the above question. Given the average car loan today is 70 months, create a new variable (column) called "MonthlyPrice" that shows the monthly cost for each car. Check to make sure the new column is there.

- Divide "VehBCost" by 70
- use the `mutate()` function

```
cars
```

```
## # A tibble: 59,957 x 8
##    Model                Make       Color  VehOdo Vehic~1 VehYear VehBC~2 Trans~3
##    <chr>                <chr>      <chr>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
##  1 MAZDA3               MAZDA      RED     89046       3    2006    7100 AUTO
##  2 1500 RAM PICKUP 2WD  DODGE      WHITE   93593       5    2004    7600 AUTO
##  3 GALANT 4C            MITSUBISHI WHITE   81054       5    2004    5600 AUTO
##  4 SPECTRA              KIA        BLACK   49921       2    2007    5600 AUTO
##  5 FIVE HUNDRED         FORD       RED     84872       2    2007    7700 AUTO
##  6 1500 SIERRA PICKUP 2 GMC        SILVER  80080       4    2005    5500 AUTO
##  7 F150 PICKUP 2WD V6   FORD       WHITE   75419       8    2001    5300 MANUAL
##  8 CARAVAN GRAND FWD V6 DODGE      RED     79315       4    2005    5400 AUTO
##  9 ALTIMA               NISSAN     WHITE   71254       4    2005    7800 AUTO
## 10 CARAVAN GRAND FWD V6 DODGE      GOLD    74722       3    2006    6900 AUTO
## # ... with 59,947 more rows, and abbreviated variable names 1: VehicleAge,
## #   2: VehBCost, 3: Transmission
## # i Use 'print(n = ...)' to see more rows
```

```
cars <- cars %>%
mutate(Monthlyprice = VehBCost/70)
colnames(cars)
```

```
## [1] "Model"        "Make"         "Color"        "VehOdo"       "VehicleAge"
## [6] "VehYear"      "VehBCost"     "Transmission" "Monthlyprice"
```

13. What is the range of the manufacture year of the vehicles?

```
range(cars$VehYear)
```

```
## [1] 2001 2010
```

14. How many cars were from before 2004? What percent/proportion of all cars do these represent?

- use`filter()` and `nrow()` or
- `group_by()` and `summarize()` or
- `sum()`

```
sum(cars$VehYear < 2004)
```

```
## [1] 6132
```

15. How many different vehicle manufacturers are there?

- use `length()` with `unique()` or `table()`. Remember to `pull()` the right column.

```
length(table(cars$Make))
```

```
## [1] 32
```

16. How many different vehicle models are there?

```
length(table(cars$Model))
```

```
## [1] 985
```

17. Which vehicle color group had the highest mean acquisition cost paid for the vehicle at time of purchase, and what was this cost?

- use `group_by()` with `summarize()` and use the `arrange()` function to sort the output by mean acquisition cost.

```
cars %>%
  group_by(Color) %>%
summarize(mean = mean(VehBCost)) %>%
  arrange(desc(mean))
```

```
## # A tibble: 17 x 2
##    Color      mean
##    <chr>     <dbl>
##  1 GREY      7551.
##  2 BLACK     7538.
##  3 BROWN     7509.
##  4 OTHER     7429.
##  5 BEIGE     7317.
##  6 RED       7279.
##  7 MAROON    7220.
##  8 WHITE     7201.
##  9 BLUE      7182.
## 10 SILVER    7175.
## 11 NOT AVAIL 7151.
## 12 ORANGE    7135.
## 13 GREEN     7089.
## 14 GOLD      7052.
## 15 YELLOW    6922.
## 16 PURPLE    6889.
## 17 NULL      5860
```

18. How many vehicles were red and have fewer than 30,000 miles? To determine the column that corresponds to mileage (odometer reading), check the "key" corresponding to the data dictionary that you imported above.

- use `filter()` and `count()` or
- `filter()` and `tally()` or
- `sum()`

```
key
```

```
## # A tibble: 36 x 2
##    `Field Name` Definition
##    <chr>        <chr>
##  1 RefId        Unique (sequential) number assigned to vehicles
##  2 IsBadBuy     Identifies if the kicked vehicle was an avoidable purchase
##  3 PurchDate    The Date the vehicle was Purchased at Auction
##  4 Auction      Auction provider at which the  vehicle was purchased
##  5 VehYear      The manufacturer's year of the vehicle
##  6 VehicleAge   The Years elapsed since the manufacturer's year
##  7 Make         Vehicle Manufacturer
##  8 Model        Vehicle Model
##  9 Trim         Vehicle Trim Level
## 10 SubModel     Vehicle Submodel
## # ... with 26 more rows
## # i Use `print(n = ...)` to see more rows
```

```
cars %>%
  filter(Color == "RED" & VehOdo < 30000) %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    24
```

19. How many vehicles are blue or red?

- use `filter()` and `count()` or
- `filter()` and `tally()` or
- `sum()`

```
cars %>%
  filter(Color == "BLUE" | Color == "RED") %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1 13777
```

20. Select all columns in "cars" where the column names starts with "Veh" (using `select()` and `starts_with()`. Then, use `colMeans()` to summarize across these columns.

```
cars %>%
  select(starts_with("Veh")) %>%
  colMeans()
```

```
##       VehOdo   VehicleAge      VehYear     VehBCost
## 70336.967210     3.896176  2005.654252  7264.971979
```

21. Using "cars", create a new binary (TRUEs and FALSEs) column to indicate if the car has an automatic transmission. Call the new column "is_automatic".

```
cars <- cars %>%
  mutate(is_automatic = (Transmission == "AUTOMATIC"))
```

22. What is the average (mean) vehicle odometer reading for cars that are both RED and NISSANs? How does this compare with vehicles that do NOT fit this criteria?

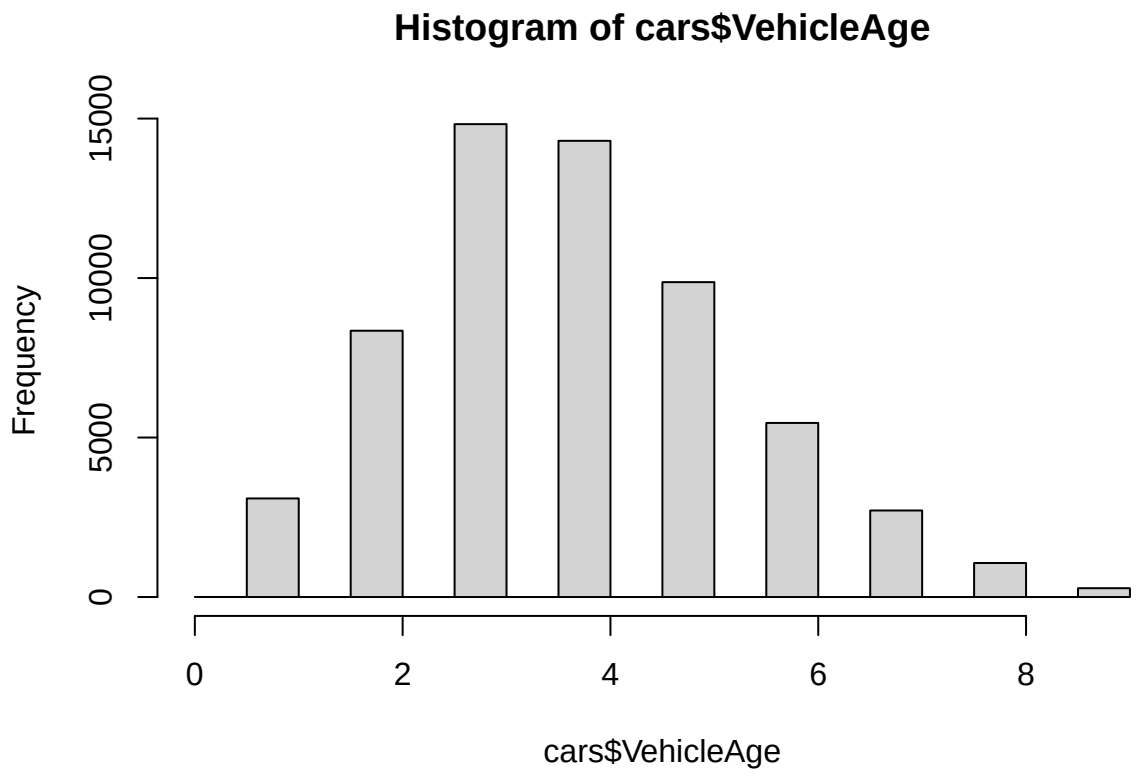
```
mean_redandnissan <- cars %>%
  filter(Color == "RED" & Make == "NISSAN") %>%
  summarize(mean = mean(VehOdo)) %>%
  pull()
mean_notredandnissan <- cars %>%
  filter(Color != "RED" | Make != "NISSAN") %>%
  summarize(mean = mean(VehOdo)) %>%
  pull()

# The average(mean) of the vehicle odometer for cars that are both RED and NISSANS is 75117.32
# and the mean for the vehicles that do not fit this criteria is 70324.34
```

23. Among red Nissans, what is the distribution of vehicle ages? To do this describe the distribution by evaluating the number of cars by vehicle age. Also, plot the distribution of vehicle with `hist()` function. Note: You have not been asked to use this function before, so first explore the help page of this function to learn the arguments needed.

```
red_nissan <- cars %>% filter(Color == "RED" & Make == "NISSAN")
red_nissan %>%
  group_by(VehicleAge) %>%
  count()
```

```
## # A tibble: 8 x 2
## # Groups:   VehicleAge [8]
##   VehicleAge     n
##        <dbl> <int>
## 1          2     8
## 2          3    35
## 3          4    48
## 4          5    34
## 5          6    13
## 6          7    14
## 7          8     4
## 8          9     2
```

```
hist(cars$VehicleAge)
```



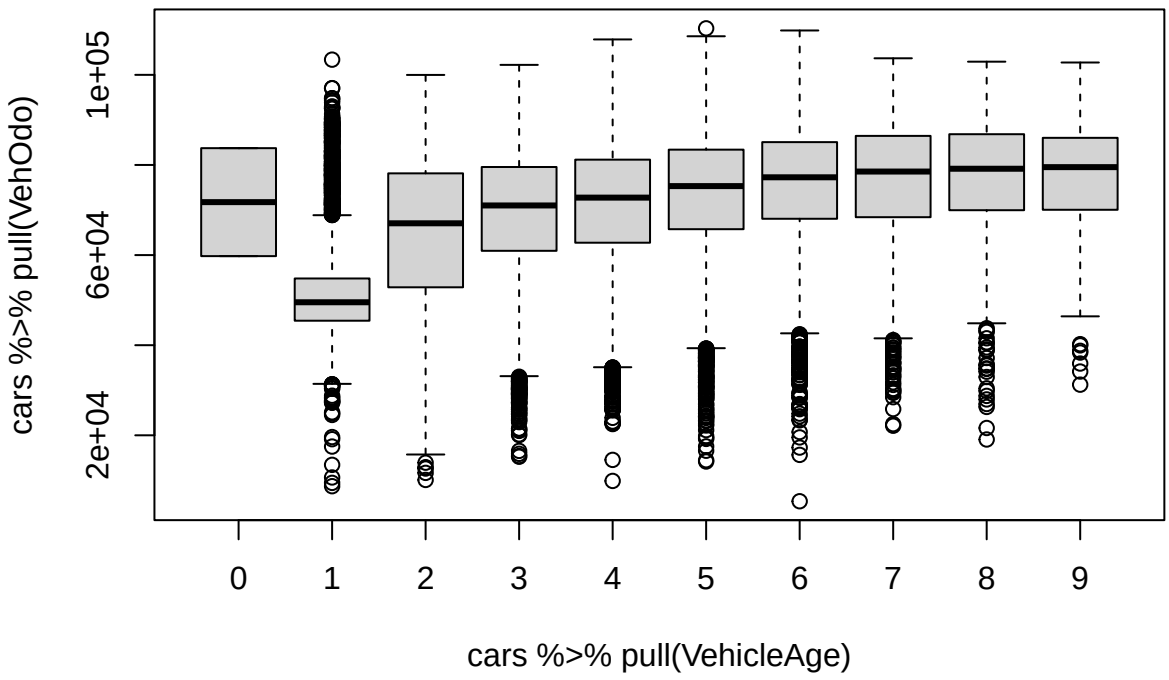
24. How many vehicles (using `filter()` or `sum()`) are made by Chrysler or Nissan and are white or silver?

```
sum((cars$Make == "CHRYSLER" | cars$Make == "NISSAN") & (cars$Color == "WHITE" | cars$Color == "SILVER"
```

```
## [1] 3718
```

25. Make a boxplot (`boxplot()`) that looks at vehicle age ("VehicleAge") on the x-axis and odometer reading ("VehOdo") on the y-axis. Note: You have not been asked to use this function before, so first explore the help page of this function to learn the arguments needed.

```
boxplot(cars %>% pull(VehOdo) ~ cars %>% pull(VehicleAge))
```



26. Knit your document into a PDF report.

You use the knit button to do this. Make sure all your code is working first!