FS21 STT 180 Homework 1

Kaitlyn Watson

Sept 20-Oct 3, 2020

This homework assignment consists of two sections. The first section deals with data structures and the second section is a small data analysis project. You will use the data wrangling and tidying knowledge for this section.

General Instructions:

- This is an individual assignment. You may consult with others as you work on the assignment, but each student should write up a separate set of solutions.
- Rather than creating a new Rmd file, just add your solutions to the supplied Rmd file. Submit both the Rmd file and the resulting HTML file to D2L.
- Except for questions, or parts of questions, that ask for your commentary, use R in a code chunk to answer the questions.
- The code chunk option `echo = TRUE` is specified in the setup code chunk at the beginning of the document. Please do not override this in your code chunks.
- A solution will lose points if the Rmd file does not compile. If one of your code chunks is causing your Rmd file to not compile, you can use the `eval = FALSE` option. Another possibility is to use the `error = TRUE` option in the code chunk.
- This Homework is due on **Saturday, OCtober 3rd, 2020 on or before 11 pm.**

Section 1

This section focuses on some basic manipulations of vectors in R.

Question 1

Create three vectors in R: One called `evennums` which contains the even integers from 1 through 15. One called `charnums` which contains character representations of the numbers 4 through 8, namely, "4", "5", "6", "7", "8". And one called `mixed` which contains the same values as in `charnums` but which also contains the letters "a", "b" and "c". **No commentary or explanations are necessary.**

```
evennums <-c(2, 4, 6, 8, 10, 12, 14)
charnums<-c("4", "5", "6", "7", "8")
mixed<-c("4", "5", "6", "7", "8", "a", "b", "c")
```

Question 2

Investigate what happens when you try to convert `evennums` to character and to logical. Investigate what happens when you convert `charnums` to numeric. Investigate what happens when you convert `mixed` to numeric. **Comment on each of these conversions.**

```
as.character(evennums)
```

```
[1] "2"  "4"  "6"  "8"  "10" "12" "14"
```

```
#The numbers now have quotations around them representing that they are characters and no longer numeri
as.logical(evennums)
```

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

```
#All of the integers turn into TRUE statements
as.numeric(charnums)
```

```
[1] 4 5 6 7 8
```

```
#The quotation marks disappear and the values are now numbers
as.numeric(mixed)
```

```
[1]  4  5  6  7  8 NA NA NA
```

```
#The last three values are characters that cannot be converted to numeric and therefore are considered
```

Question 3

No commentary is necessary on this part.

a. Show how to extract the first element of `evennums`.

```
evennums[1]
```

```
[1] 2
```

b. Show how to extract the last element of `evennums`. In this case you are NOT allowed to use the fact that `evennums` has seven elements, rather, you must give code which would work no matter how many elements `nums` has.

```
evennums[length(evennums)]
```

```
[1] 14
```

c. Show how to extract all but the first element of `evennums`.

```
evennums[-1]
```

```
[1]  4  6  8 10 12 14
```

d. Show how to extract all but the first two and last two elements of `evennums`.

```
evennums[c(3,4,5)]
```

```
[1]  6  8 10
```

Question 4

a. Generate a sequence "y" of 50 evenly spaced values between 0 and 1.

```
y<-seq(0, 1, length=50)
y
```

```
 [1] 0.00000000 0.02040816 0.04081633 0.06122449 0.08163265 0.10204082
 [7] 0.12244898 0.14285714 0.16326531 0.18367347 0.20408163 0.22448980
[13] 0.24489796 0.26530612 0.28571429 0.30612245 0.32653061 0.34693878
[19] 0.36734694 0.38775510 0.40816327 0.42857143 0.44897959 0.46938776
[25] 0.48979592 0.51020408 0.53061224 0.55102041 0.57142857 0.59183673
[31] 0.61224490 0.63265306 0.65306122 0.67346939 0.69387755 0.71428571
[37] 0.73469388 0.75510204 0.77551020 0.79591837 0.81632653 0.83673469
[43] 0.85714286 0.87755102 0.89795918 0.91836735 0.93877551 0.95918367
[49] 0.97959184 1.00000000
```

b. Calculate the mean of the sequence.

```
mean(y)
```

```
[1] 0.5
```

Section 2

The dataset contains information about births in the United States. The full data set is from the Centers for Disease Control. The data for this homework assignment is a "small" sample (chosen at random) of slightly over one million records from the full data set. The data for this homework assignment also only contain a subset of the variables in the full data set. ## Setting up

Load `tidyverse`, which includes `dplyr`, `tidyr`, and other packages, and the load 'knitr.

```
library(tidyverse)
library(knitr)
```

Read in the data and convert the data frame to a tibble.

```
birth_data <- read.csv("BirthData.csv", header = TRUE)
birth_data <- as_tibble(birth_data)
```

A glimpse of the data:

```
glimpse(birth_data)
```

```
Rows: 1,103,629
Columns: 8
$ year          <int> 1969, 1969, 1969, 1969, 1969, 1969, 1969, 1969, 1969, 19~
$ month         <int> 9, 8, 9, 2, 3, 5, 5, 5, 6, 8, 8, 11, 11, 11, 1, 12, 3, 3~
$ state         <chr> "AL", "AZ", "AZ", "CA", "CA", "CA", "CA", "CA", "CA", "C~
$ is_male       <lgl> FALSE, TRUE, TRUE, TRUE, FALSE, TRUE, FALSE, TRUE, TRUE,~
$ weight_pounds <dbl> 1.624807, 7.500126, 8.937540, 6.999677, 6.876218, 7.1870~
$ mother_age    <int> 20, 35, 17, 20, 25, 30, 17, 22, 26, 26, 19, 25, 26, 26, ~
$ child_race    <int> 2, 1, 1, 1, 2, 1, 1, 4, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1,~
$ plurality     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
```

The variables in the data set are:

Variable	Description
`year`	the year of the birth
`month`	the month of the birth
`state`	the state where the birth occurred, including "DC" for Washington D.C.
`is_male`	which is `TRUE` if the child is male, `FALSE` otherwise
`weight_pounds`	the child's birth weight in pounds
`mother_age`	the age of the mother
`child_race`	race of the child.
`plurality`	the number of children born as a result of the pregnancy, with 1 representing a single birth, 2 representing twins, etc.

For both of Questions 1 and 2 you should show the R code used and the output of the `str` and`glimpse` functions applied to the data frame. Use of dplyr functions and the pipe operator is highly recommended.

Question 1

Create a variable called `region` in the data frame `birth_data` which takes the values `Northeast`, `Midwest`, `South`, and `West`. The first two Steps have been done for you.

Here are the states in each region:

Northeast Region: Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island and Vermont, New Jersey, New York, and Pennsylvania

Midwest Region: Illinois, Indiana, Michigan, Ohio and Wisconsin, Iowa, Kansas, Minnesota, Missouri, Nebraska, North Dakota, and South Dakota

South Region: Delaware, District of Columbia, Florida, Georgia, Maryland, North Carolina, South Carolina, Virginia, and West Virginia, Alabama, Kentucky, Mississippi, and Tennessee, Arkansas, Louisiana, Oklahoma, and Texas

West Region: Arizona, Colorado, Idaho, Montana, Nevada, New Mexico, Utah and Wyoming, Alaska, California, Hawaii, Oregon and Washington

```
#Step 1: Assign the regions.
NE <- c("CT", "ME", "MA", "NH", "RI", "VT", "NJ", "NY", "PA")
MW <- c("IL", "IN", "MI", "OH", "WI", "IA", "KS", "MN", "MO", "NE", "ND", "SD")
SO <- c("DE", "DC", "FL", "GA", "MD", "NC", "SC", "VA", "WV", "AL", "KY", "MS", "TN", "AR", "LA", "OK",
WE <- c("AZ", "CO", "ID", "MT", "NV", "NM", "UT", "WY", "AK", "CA", "HI", "OR", "WA")
## Step 2 Create a blank vector
birth_data$region <- rep(NA, length(birth_data$state))

##Step 3
birth_data$region[birth_data$state%in%NE]<-"Northeast"
birth_data$region[birth_data$state%in% MW]<- "Midwest"
birth_data$region[birth_data$state%in% SO]<-"South"
birth_data$region[birth_data$state%in% WE ]<-"West"

glimpse(birth_data$region)
```

```
 chr [1:1103629] "South" "West" "West" "West" "West" "West" "West" "West" ...
```

```
## Hint use if-else and %in% to create the regions.
```

Question 2

Create a variable in `birth_data` called `state_color` which takes the values `red`, `blue`, and `purple`, using the following divisions.

Red: Alaska, Idaho, Kansas, Nebraska, North Dakota, Oklahoma, South Dakota, Utah, Wyoming, Texas, Alabama, Mississippi, South Carolina, Montana, Georgia, Missouri, Louisiana, Tennessee, Arkansas, Kentucky, Arizona, West Virginia.

Purple: North Carolina, Virginia, Florida, Ohio, Colorado, Nevada, Indiana, Iowa, New Mexico.

Blue: New Hampshire, Pennsylvania, California, Michigan, Illinois, Maryland, Delaware, New Jersey, Connecticut, Vermont, Maine, Washington, Oregon, Wisconsin, New York, Massachusetts, Rhode Island, Hawaii, Minnesota, District of Columbia.

```
RED <- c("AK", "ID", "KS", "NE", "ND", "OK", "SD", "UT", "WY", "TX", "AL", "MS", "SC", "MT", "GA", "MO"
PURPLE <- c("NC", "VA", "FL", "OH", "CO", "NV", "IN", "IA", "NM")
BLUE <- c("NH", "PA", "CA", "MI", "IL", "MD", "DE", "NJ", "CT", "VT", "ME", "WA", "OR", "WI", "NY", "MA


state_color_data<-birth_data%>%
  mutate(state_color= ifelse(state%in%RED,"red", ifelse(state%in%BLUE,"blue", ifelse(state%in%PURPLE,"p

state_color_data
```

```
# A tibble: 1,103,629 x 10
    year month state is_male weight_pounds mother_age child_race plurality
   <int> <int> <chr> <lgl>           <dbl>      <int>      <int>     <int>
 1  1969     9 AL    FALSE            1.62         20          2        NA
```

```
 2  1969     8 AZ    TRUE            7.50         35          1        NA
 3  1969     9 AZ    TRUE            8.94         17          1        NA
 4  1969     2 CA    TRUE            7.00         20          1        NA
 5  1969     3 CA    FALSE           6.88         25          2        NA
 6  1969     5 CA    TRUE            7.19         30          1        NA
 7  1969     5 CA    FALSE           7.75         17          1        NA
 8  1969     5 CA    TRUE            7.37         22          4        NA
 9  1969     6 CA    TRUE            9.63         26          2        NA
10  1969     8 CA    FALSE           7.94         26          1        NA
# ... with 1,103,619 more rows, and 2 more variables: region <chr>,
#   state_color <chr>
```

```
## try using mutate
```

Question 3

Create two new objects `perc_male` and `perc_female` that calculates the percentile ranking of a baby's weight with respect to the baby's sex.

```
## The dataset to find the male percentiles
birth_data_male<-birth_data%>%
             filter(is_male== TRUE)

 perc_male<-quantile(birth_data_male$weight_pounds,probs=c(0,0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9

 perc_male
```

```
       0%        10%        20%        30%        40%        50%        60%
 0.5004493  5.8753193  6.5322968  6.9379474  7.2510038  7.5001262  7.8131826
      70%        80%        90%       100%
 8.1262390  8.4767740  8.9992695 17.9897206
```

```
## Do the same steps for the female population.

birth_data_female<-birth_data%>%
             filter(is_male== FALSE)
perc_female<-quantile(birth_data_female$weight_pounds, probs=c(0,0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8

perc_female
```

```
       0%        10%        20%        30%        40%        50%        60%
 0.5004493  5.7496558  6.3118346  6.6866204  6.9996768  7.2510038  7.5001262
      70%        80%        90%       100%
 7.8131826  8.1571037  8.6244837 17.1453501
```

```
## Hint: use the quantile function to find the percentiles.
```

Question 4

Create another new variable that that stores the percentile cut-points for a baby's weight with respect to the baby's plurality (i.e., whether it was a single child, twin, triplet, etc.)

[For example, if a baby is a twin (plurality = 2), the variable should record the percentile ranking of the baby's weight relative only to all other twins.]

```
## The dataset for plurality == 1 ; do the same for the other pluralities
birth_data1<-birth_data%>%
             filter(plurality == 1)
perc_1<-quantile(birth_data1$weight_pounds, na.rm=TRUE)

perc_1
```

```
       0%        25%        50%        75%       100%
 0.5004493  6.6866204  7.4383967  8.1791499 17.9897206
```

```
birth_data2<-birth_data%>%
             filter(plurality == 2)
perc_2<-quantile(birth_data2$weight_pounds, na.rm=TRUE)

perc_2
```

```
       0%        25%        50%        75%       100%
 0.5004493  4.4379053  5.4079393  6.1883757 13.4658350
```

```
birth_data3<-birth_data%>%
             filter(plurality == 3)
perc_3<-quantile(birth_data2$weight_pounds, na.rm=TRUE)

perc_3
```

```
       0%        25%        50%        75%       100%
 0.5004493  4.4379053  5.4079393  6.1883757 13.4658350
```

```
birth_data4<-birth_data%>%
             filter(plurality == 4)
perc_4<-quantile(birth_data2$weight_pounds, na.rm=TRUE)

perc_4
```

```
       0%        25%        50%        75%       100%
 0.5004493  4.4379053  5.4079393  6.1883757 13.4658350
```

```
birth_data5<-birth_data%>%
             filter(plurality == 5)
perc_5<-quantile(birth_data5$weight_pounds, na.rm=TRUE)

perc_5
```

```
       0%       25%       50%       75%      100%
0.6856376 1.0262518 2.1076192 3.0396234 3.6243996
```

```
birth_data6<-birth_data%>%
              filter(plurality == 6)
perc_6<-quantile(birth_data5$weight_pounds, na.rm=TRUE)

perc_6
```

```
       0%       25%       50%       75%      100%
0.6856376 1.0262518 2.1076192 3.0396234 3.6243996
```

```
## Hint: use the quantile function to find the percentiles.
```

Question 5

Provide an example case in which these two percentile rankings in Question 3 and Question 4 (gender vs plurality) would be quite similar. Provide another example case in which these two percentile rankings would be quite different.

Male babies and baby's plurality=1 are quite similar in regards to their percentile rank using weight in pounds

However, female babies and and baby's plurality=5 have a quite different percentile rank using weight in pounds.

Question 6

Agree or disagree with this claim. **If you agree, provide a rationale for why it is correct. If you disagree, provide a counter-example that reveals the error in its thinking:**

"If these two percentile rankings are very different from one another, we should suspect that the baby in question is more likely to be a twin/triplet/etc., than a single-birth."

I agree with this claim. Both male and female percentile ranks in regard to weight are similar to one another.When looking at the plurality percentile ranks, the only similar percentile to gender is if a single-birth occurred. Therefore, for multiple births, it is evident that the percentile ranks are quite different which supports the statement above.

Some of the variables have missing values, and these may be related to different data collection choices during different years. For example, possibly plurality wasn't recorded during some years, or state of birth wasn't recorded during some years. In this exercise we investigate using some `dplyr` functions. Hint: The `group_by` and `summarize` functions will help.

Question 7

Count the number of missing values in each variable in the data frame.

```
state_color_data %>%
  summarise_each(funs(sum(is.na(.))))
```

```
# A tibble: 1 x 10
   year month  state is_male weight_pounds mother_age child_race plurality
  <int> <int>  <int>   <int>         <int>      <int>      <int>     <int>
1     0     0 135937       0          1660          0     201636     29088
# ... with 2 more variables: region <int>, state_color <int>
```

Question 8

Use `group_by` and `summarize` to count the number of missing values of the two variables, `state` and `child_race`, for each year, and to also count the total number of observations per year.

Are there particular years when these two variables are either not available, or of limited availability?

```
birth_data %>%
  group_by(year) %>%
  summarise(missing_state=sum(is.na(state)),child_race=sum(is.na(child_race)),num_obs=length(state))
```

```
# A tibble: 40 x 4
    year missing_state child_race num_obs
   <int>         <int>      <int>   <int>
 1  1969             0          0   14280
 2  1970             0          0   14808
 3  1971             0          0   14209
 4  1972             0          0   14106
 5  1973             0          0   14840
 6  1974             0          0   16432
 7  1975             0          0   18194
 8  1976             0          0   19537
 9  1977             0          0   22036
10  1978             0          0   23064
# ... with 30 more rows
```

```
#2005-2008 lacks data for state and child race. 2003 and 2004 also lacks child race data.
```

Question 9

Create the following data frame which gives the counts, the mean weight of babies and the mean age of mothers for the six levels of `plurality`. Comment on what you notice about the relationship of plurality and birth weight, and the relationship of plurality and age of the mother.

```
birth_data %>%
  group_by(plurality) %>%
  summarise(count=n(),mean_weight_pounds = mean(weight_pounds, na.rm=TRUE), mean_age_mothers = mean (mo
```

```
# A tibble: 6 x 4
  plurality   count mean_weight_pounds mean_age_mothers
      <int>   <int>              <dbl>            <dbl>
1         1 1046856               7.37             26.3
2         2   26582               5.22             28.1
3         3    1018               3.74             30.7
4         4      75               2.81             31.3
5         5      10               2.05             30.9
6        NA   29088               7.21             24.6
```

Question 10

Create a data frame which gives the counts, the mean weight of babies and the mean age of mothers for each combination of the four levels of `state_color` and the two levels of `is_male`.

```
state_color_data%>%
  group_by(state_color, is_male) %>%
  summarise(count=n(), mean_weight_pounds = mean(weight_pounds, na.rm=TRUE), mean_age_mothers = mean (m
```

```
# A tibble: 8 x 5
# Groups:   state_color [4]
  state_color is_male  count mean_weight_pounds mean_age_mothers
  <chr>       <lgl>    <int>              <dbl>            <dbl>
1 blue        FALSE   227900               7.24             26.8
2 blue        TRUE    241398               7.50             26.7
3 NA          FALSE    66384               7.08             27.4
4 NA          TRUE     69553               7.30             27.4
5 purple      FALSE    94073               7.16             25.9
6 purple      TRUE     99079               7.43             25.9
7 red         FALSE   148892               7.15             25.3
8 red         TRUE    156350               7.41             25.3
```

Essential details

Deadline and submission

- The deadline to submit Homework 1 is **11:00pm on Saturday, October 2nd, 2021.** This is a individual assignment.
- Submit your work by uploading your RMD and HTML/PDF files through D2L. Kindly **double check your submission to note whether the everything is displayed in the uploaded version of the output in D2L or not.** If submitting HTML outputs, please zip the .rmd and html files for submission.
- Please enable the `echo=TRUE` option in your code chunks and name each code chunk.
- **Late work will not be accepted except under certain extraordinary circumstances.**

Help

- Post general questions in the Teams HW 1 channel. If you are trying to get help on a code error, explain your error in detail
- Feel free to visit us in during our virtual office hours or make an appointment.
- Communicate with your classmates, but do not share snippets of code.
- The instructional team **will not answer any questions within the first 24 hours of this homework being assigned, and we will not answer any questions after 6 P.M of the due date**.

Academic integrity

This is an individual assignment.You may discuss ideas, how to debug code, and how to approach a problem with your classmates in the discussion board forum. You may not copy-and-paste another's code from this class. As a reminder, below is the policy on sharing and using other's code.

Similar reproducible examples (reprex) exist online that will help you answer many of the questions posed on group assignments, and homework assignments. Use of these resources is allowed unless it is written explicitly on the assignment. You must always cite any code you copy or use as inspiration. Copied code without citation is plagiarism and will result in a 0 for the assignment.

Grading

You must use R Markdown. Formatting is at your discretion but is graded. Use the in-class assignments and resources available online for inspiration. Another useful resource for R Markdown formatting is available at: https://holtzy.github.io/Pimp-my-rmd/

Topic	**Points**
Questions 1-4 (Sec 1) and 1-10 (Sec 2)	84
R Markdown formatting	5
Rmd file compilation	5
Code style and named code chunks	6
Total	100