Coding Assignment 4

Overview

For this assignment, you will continue to work with the sleep study data. As a reminder, the documentation for the data can be found here, but the data are on Github. (Note that the documentation is not entirely accurate, but it should still provide sufficient guidance for our purposes.)

As before, run the following chunk to load `SleepStudy.Rda`. This will load the R object `SleepStudy` (a data frame). For this code chunk to run, be sure that the data file is in a subfolder of your DSC201 folder called "data."

```
load("data/SleepStudy.Rda")
```

(1) In the following code chunk, keep the library statement. This loads the `dplyr` package. In this code chunk, convert the data frame to a tibble. Then get a summary description of the data using `glimpse()`. [1 POINT]

```
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
```

```
SleepTibble <- as_tibble(SleepStudy)
glimpse(SleepTibble)
```

```
## Rows: 253
## Columns: 30
## $ Gender           <int> 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1,~
## $ ClassYear        <int> 4, 4, 4, 1, 4, 4, 2, 2, 1, 4, 2, 2, 1, 3, 3, 3, 2, 2,~
## $ LarkOwl          <fct> Neither, Neither, Owl, Lark, Owl, Neither, Lark, Lark~
## $ NumEarlyClass    <int> 0, 2, 0, 5, 0, 0, 2, 0, 2, 2, 1, 0, 4, 2, 5, 0, 2, 5,~
## $ EarlyClass       <int> 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1,~
## $ GPA              <dbl> 3.60, 3.24, 2.97, 3.76, 3.20, 3.50, 3.35, 3.00, 4.00,~
## $ ClassesMissed    <int> 0, 0, 12, 0, 4, 0, 2, 0, 0, 0, 0, 0, 0, 6, 1, 0, 5, 0~
## $ CognitionZscore  <dbl> -0.26, 1.39, 0.38, 1.39, 1.22, -0.04, 0.41, -0.59, 1.~
## $ PoorSleepQuality <int> 4, 6, 18, 9, 9, 6, 2, 10, 5, 2, 11, 8, 3, 7, 4, 4, 8,~
## $ DepressionScore  <int> 4, 1, 18, 1, 7, 14, 1, 2, 12, 6, 2, 2, 10, 3, 2, 3, 1~
## $ AnxietyScore     <int> 3, 0, 18, 4, 25, 8, 0, 2, 16, 11, 12, 8, 13, 3, 0, 1,~
## $ StressScore      <int> 8, 3, 9, 6, 14, 28, 1, 3, 20, 31, 13, 11, 18, 2, 3, 1~
## $ DepressionStatus <fct> normal, normal, moderate, normal, normal, moderate, n~
## $ AnxietyStatus    <fct> normal, normal, severe, normal, severe, moderate, nor~
## $ Stress           <fct> normal, normal, normal, normal, normal, high, normal,~
## $ DASScore         <int> 15, 4, 45, 11, 46, 50, 2, 7, 48, 48, 27, 21, 41, 8, 5~
```

```
## $ Happiness        <int> 28, 25, 17, 32, 15, 22, 25, 29, 29, 30, 14, 24, 21, 2~
## $ AlcoholUse       <fct> Moderate, Moderate, Light, Light, Moderate, Abstain, ~
## $ Drinks           <int> 10, 6, 3, 2, 4, 0, 6, 3, 3, 6, 10, 10, 4, 5, 0, 2, 4,~
## $ WeekdayBed       <dbl> 25.75, 25.70, 27.44, 23.50, 25.90, 23.80, 25.35, 23.9~
## $ WeekdayRise      <dbl> 8.70, 8.20, 6.55, 7.17, 8.67, 8.95, 8.48, 9.07, 8.75,~
## $ WeekdaySleep     <dbl> 7.70, 6.80, 3.00, 6.77, 6.09, 9.05, 7.73, 9.02, 8.25,~
## $ WeekendBed       <dbl> 25.75, 26.00, 28.00, 27.00, 23.75, 26.00, 25.63, 25.1~
## $ WeekendRise      <dbl> 9.50, 10.00, 12.59, 8.00, 9.50, 10.75, 10.13, 9.75, 9~
## $ WeekendSleep     <dbl> 5.88, 7.25, 10.09, 7.25, 7.00, 9.00, 7.00, 9.00, 9.25~
## $ AverageSleep     <dbl> 7.18, 6.93, 5.02, 6.90, 6.35, 9.04, 7.52, 9.01, 8.54,~
## $ AllNighter       <int> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Sex              <fct> Female, Female, Female, Female, Female, Male, Male, F~
## $ allNighter       <fct> No, No, No, No, No, No, Yes, No, No, No, No, No, No, ~
## $ earlyClass       <fct> No, Yes, No, Yes, No, No, Yes, No, Yes, Yes, Yes, No,~
```

Data Moves

"Data moves" are a data science framework introduced by Erickson et. al. (2019). They define a data move as an action that alters a dataset's contents, structure, or values. They define six core data moves: (1) filtering, (2) grouping, (3) summarizing, (4) calculating, (5) merging/joining, and (5) making hierarchy. We will cover the first four data moves in this assignment, while practicing `dplyr` functionality.

Filtering

Filtering produces a subset of data. It serves at least two important purposes. First, if a dataset includes extraneous cases, filtering removes the irrelevant ones. This is sometimes called scoping—reducing the scope of the investigation—or focusing. Also, filtering may be used in order to reduce the complexity or quantity of data in order to gain insight. (Largely copied from Erickson et. al., 2019)

In R-Notebook3.Rmd, we learned several `dplyr` functions/verbs that do filtering. Note that filtering does operations on rows.

(2) Filter the sleep study tibble in two different ways. That is, use two different `dplyr` functions/verbs or use the same function/verb twice to reduce your rows according to two different criteria. For each of the two ways: [3 POINTS FOR EACH WAY]

- Write 1-3 sentences describing the filtering you are doing and how such filtering may be helpful in gaining insights from the data. For example, "I am creating a reduced data set that includes only students who have a normal stress level (for whom the variable `Stress` is equal to"normal"). This would allow me to analyze sleep patterns among students who have normal stress-levels." You do not need to implement what the filtering would allow you to do - just talk about it like in my example. [1 POINT]
- Create a code chunk and use a `dplyr` function/verb to do the filtering. Create a data frame (name as you choose) to hold the new, filtered data (the revised version of `SleepStudy`). If you are not sure what code to write or cannot get it to work, partial credit will be give for writing text of what coding you are attempting to do. [1 POINT]
- Write 1-2 lines of code that will display results that help you check if your filtering worked as you expected. This may be a print or some other display. If you decide to print data, limit your print to a subset of rows and columns so that you can view the entire display on your screen. You have an example of this in the last homework. You do not have to use `dplyr` for the check. If you are not sure what code to write, partial credit will be give for writing text of what you think would be helpful. [1 POINT]

Grouping

Grouping is typically used to set up a comparison among different subgroups of a dataset. Just as filtering restricts analysis to a single subset, grouping divides a dataset into multiple subsets. This division is guided by the available value(s) of some variable or variables so that, among the observations within each resulting group, the values of these "grouping" variables are the same. Note that "binning" is a special type of grouping that uses ranges of continuous values (bins or classes) to determine group membership. (Largely copied from Erickson et. al., 2019) Note that we learned about group in R-Notebook3.Rmd and remember that grouping does operations on rows.

(3) Group the sleep study tibble in two different ways. That is, use two different `dplyr` functions/verbs or use the same function/verb twice to group your data according to two different criteria. For each of the two ways: [3 POINTS FOR EACH WAY]

- Write 1-3 sentences describing the grouping you are doing and how such grouping may be helpful in gaining insights from the data. For example, "I am grouping students by their gender. This would allow me to compare summary statistics by gender." You do not need to implement what the grouping would allow you to do (e.g. compare summary statistics by gender) - just talk about it like in my example. [1 POINT]
- Create a code chunk and use a `dplyr` function/verb to do the grouping. Create a data frame (name as you choose) to hold the new, grouped data (the revised version of `SleepStudy`). If you are not sure what code to write or cannot get it to work, partial credit will be give for writing text of what coding you are attempting to do. [1 POINT]
- Write 1-2 lines of code that will display some results that help you check if your grouping worked as you expected. This may be a print or some other display. But limit your print to a subset of rows and columns so that you can view the entire display on your screen. You do not have to use `dplyr` for the check. If you are not sure what code to write, partial credit will be give for writing text of what you think would be helpful. [1 POINT]

Summarizing

Analysts often compute values that summarize a group (even if the group is the entire data set). Summarizing is the process of producing and recording a summary or aggregate value, i.e., a statistic. There are a wide variety of summary measures, and "summary" does not necessarily mean "numerical" or "typical." Often, the point of summarizing is not even the chosen aggregate measure, or the results of that measure across groups. The purpose may be deeper: The value of an aggregate measure summarizes a group, and that summary value can then be used as data in further analysis.

Grouping and summarizing work together to help an analyst get a simpler display or dataset—many fewer points!—that more clearly shows an overall pattern. Note, though, that consolidation into simpler distinct categories leads to a reduction of information. For example, when a display shows only measures of center, variability is lost. (Largely copied from Erickson et. al., 2019)

(4) Summarize the sleep study tibble in two different ways. For one of them, combine summarizing with grouping. For each of the two ways: [3 POINTS FOR EACH WAY]

- Write 1-3 sentences describing the summarizing you are doing and how such summarizing may be helpful in gaining insights from the data. For example, when grouping and summarizing, "I grouping students by whether they have an early class (whether the variable `EarlyClass==1`) and then computing the mean time that students go to bed on weekdays, by group (the mean of the variable `WeekdayBed`). This allows me to see if students who have an early class go to bed earlier on average." [1 POINT]
- Create a code chunk and use a `dplyr` function/verb to do the summarizing. Create an object (name as you choose) to hold the summarized data (remember that the summary results are a small data frame, as we discussed in class). If you are not sure what code to write or cannot get it to work, partial credit will be give for writing text of what coding you are attempting to do. [1 POINT]

- Write a sentence or two interpreting your results. [1 POINT]

Calculating

Another data move is to create a new variable, often represented by a new column in a data table. Because this typically involves calculating the values in this new variable, this data move is called calculating by Erickson et. al. Others refer to calculating as "mutating" or "transforming." Many new variables are calculated using the values from one or more existing variables, which is what you will do below.

(5) Create two new variables in the data frame `SleepStudy`. Each new variable should be created as a calculation or transformation performed on one or more existing variables in `SleepStudy`. [6 POINTS TOTAL]

- For each new variable, write 1-3 sentences defining it and how it may be helpful for other analyses or research questions about the data. For example, "I am creating a variable called `HighHappiness` which is equal to one if `Happiness` is greater than 26. This would allow me to group students by whether they have high happiness levels or not." You do not need to implement what the new variable would allow you to do (e.g. group students by their happiness) - just talk about it like in my example. [2 POINTS]
- Use `dplyr` to create both new variables in one code chunk. Create a new data frame (name as you choose) to hold the new, expanded data (identical to `SleepStudy` but with two new variables). If you are not sure what code to write or cannot get it to work, partial credit will be give for writing text of what coding you are attempting to do. [2 POINTS]
- Write 1-2 lines of code that will display some results that help you check if your grouping worked as you expected. This may be a print or some other display. But limit your print to a subset of rows and columns so that you can view the entire display on your screen. You do not have to use `dplyr` for the check. If you are not sure what code to write, partial credit will be give for writing text of what you think would be helpful. [2 POINTS]

Extra Credit

In the last homework (Coding-Assignment3.Rmd Problem #6), you explored the question: Do students with insufficient sleep have lower GPA's than students with sufficient sleep? You used base R to follow steps to answer this question in a code chunk named `compareGPA`. For extra credit, create a code chunk and use `dplyr` to carry out the same analysis. That is, use `dplyr` to compute the GPA for students with insufficent sleep and the GPA for students with sufficient sleep. [2 POINTS]

References

Erickson, T., Wilkerson, M., Finzer, W., & Reichsman, F. (2019). Data Moves. Technology Innovations in Statistics Education, 12(1). http://dx.doi.org/10.5070/T5121038001 Retrieved from https://escholarship.org /uc/item/0mg8m7g6