PS1 Solution

BSS Stat 20

2022-06-15

Question 1

part a

This subproblem is meant to get students using the help files.

```
?iris
```

A species of iris (a flower).

part b

`Sepal.Length`, `Sepal.Width`, `Petal.Length`, `Petal.Width` are continuous, numerical variables.

`Species` is a nominal, categorical variable.

part c

I used the `sort()` function here but this is not necessary. Scrolling through the data frame with the Viewer is probably enough.

```
sort(iris$Sepal.Length)
```

Answers will vary (in terms of what categories are made), but one possible choice could be a variable with 4 levels: <5.0, <6.0, <7.0, 7.0+. In any case, we will have an ordinal, categorical variable.

Question 2

The purpose of this question is to get students using help files and also thinking a bit about the data sets they might work with.

part a

"Subcompact" has been fabricated. This is not a level of the `Type` category, which can be seen by checking the help file for Cars93.

part b

"17" has been fabricated. `Min.Price` is a continuous, numerical variable; therefore, "17.0" is an accurate recording but "17" is not.

Question 3

part a

The variables are (identified) gender (which is categorical and nominal) and admission status (which is categorical and nominal).

part b

There are 512 + 89 admitted students. Divide this by the total number of students (512 + 89 + 313 + 19) to get

```
(512 + 89)/(512 + 89 + 313 + 19)
```

```
## [1] 0.6441586
```

part c

There are 89 + 19 students reported as female. Divide this by the total number of students (512 + 89 + 313 + 19) to get

```
(19 + 89)/(512 + 89 + 313 + 19)
```

```
## [1] 0.1157556
```

part d

We are only concerned now with the `Male` column of the table. We see the correct proportion of `Admitted` students is

```
512/(512 + 313)
```

```
## [1] 0.6206061
```

part e

We are only concerned now with the `Rejected` row of the table. We see the correct proportion of students reported as `Female` is

```
19/(313 + 19)
```

```
## [1] 0.05722892
```

Question 4

The purpose of this question is solely to get students in the habit of setting the random seed value when performing random generation. It also makes sure they know how to use comments.

```r
set.seed(1)
my_nums <- rnorm(n = 100)
```

part a

A vector of numbers (numeric vector).

part b

Answers will vary.

```r
sum(my_nums)
```

```
## [1] 10.88874
```

part c

Answers will vary.

```r
my_nums2 <- rnorm(n = 100)
sum(my_nums2)
```

```
## [1] -3.780808
```

part d

Answers will vary (first object's sum is **smaller** or **larger** than that of the second). In my case, the sum of the first object is **larger** than that of the second.

part e

I set the random seed. This ensures upon knitting my document that the computer gives me the same results I achieved earlier.

part f

```r
#This line of code ensured my correctness

set.seed(1)
my_nums <- rnorm(n = 100)
```

Question 5

This is more of a lab-style question. It is the Spring 2022 Lab 10, albeit condensed.

part a

The unit of observation would be a student.

We would need to collect information on whether the students' section was led by a male or female-identifying instructor, and we'd need to know the student's evaluation results for that instructor.

part b

Each row corresponds to a student and the evaluations of their teacher. The data frame has dimensions 47 rows by 20 columns.

part c

`student_gender` is a categorical, nominal variable, despite the numeric coding.

part d

In this case, it would be wise to let `overall` be categorical, ordinal.

part e

In this case, it would be wise to let `overall` be numerical, discrete.

part f

Accessing the data landing page that was linked earlier in the question, we see that the data was published in 2015.

Therefore, we need to use `mutate()` to create the correct age variable.

```r
library(tidyverse)
SET <- read_csv("https://www.dropbox.com/s/jog3lnqjinabe9s/set.csv?dl=1", show_col_types = FALSE)

SET <- mutate(SET, new_age = 2015 - student_age)

#Sanity check
head(select(SET, student_age, new_age))
```

```
## # A tibble: 6 x 2
##   student_age new_age
##         <dbl>   <dbl>
## 1        1990      25
## 2        1992      23
## 3        1991      24
## 4        1991      24
## 5        1992      23
## 6        1991      24
```