ISTA 116: Lab Assignment #1 (50 pts)

SOLUTION

Due by the end of lab on Aug. 30-31 (as appropriate)

1 (Textbook 1.4) Calculation (8 pts)

Use `R` as you would a calculator to find numeric answers to the following (each lettered subpart worth 2 pts):

a. $1 + 2(3 + 4)$

```
>  1 + 2 * (3 + 4)
[1] 15
```

b. $4^3 + 3^{2+1}$

```
> 4 ^ 3 + 3 ^ (2 + 1)
[1] 91
```

c. $\sqrt{(4 + 3)(2 + 1)}$

```
> sqrt((4 + 3) * (2 + 1))
[1] 4.582576
```

d. $(\frac{1+2}{3+4})^2$

```
> ((1 + 2) / (3 + 4)) ^ 2
[1] 0.1836735
```

2 (Text 1.8) Arithmetic With Vectors (8 pts)

Let our small data set be:

```
2 5 4 10 8
```

a. Enter this data into a data vector `x`.

```
> x = c(2, 5, 4, 10, 8)
```

b. Find the square of each number.

```
> x ^ 2
[1]   4  25  16 100  64
```

c. Subtract 6 from each number.

```
> x - 6
[1] -4 -1 -2  4  2
```

d. Subtract 9 from each number and then square the resulting values.

```
> (x - 9) ^ 2
[1] 49 16 25  1  1
```

Use a single line of code for each letter (2 pts each).

3 (Text 1.14) Summarizing a Vector (9 pts)

You track your commute times for ten days, recording the following times in minutes:

```
17 16 20 24 22 15 21 15 17 22
```

a. (4 pts) Enter these into `R`. Use the function `max()` to find the longest commute time, the function `mean()` to find the average, and the function `min()` to find the minimum.

```
> time = c(17, 16, 20, 24, 22, 15, 21, 15, 17, 22)
> max(time)
[1] 24
> mean(time)
[1] 18.9
> min(time)
[1] 15
```

b. (5 pts) How many times was your commute 20 minutes or more? What percentage of your commutes are less than 18 minutes long? Use `R` to find these answers (Hint: Extract logical subsets, and use the `length()` function)

```
> gte20 = time[time >= 20]
> gte20
[1] 20 24 22 21 22
> length(gte20)
[1] 5
```

```
> lt18 = time[time < 18]
> lt18
[1] 17 16 15 15 17
> (length(lt18) / length(time)) * 100
[1] 50
```

4 (Modified from Text 1.25) Accessing Parts of a Data Frame (14 pts)

The data set `nym.2002` in the `UsingR` package contains data about participants in the 2002 New York City Marathon. Use `R` commands to answer the following questions (include your code as well as the answer, where applicable).

a. (2 pts) Load the `UsingR` library into the workspace.

```
> install.packages("UsingR") # will only need to do this once per R installation
```

Then do:

```
> data(nym.2002,package="UsingR")
```

or:

```
> library("UsingR")
> data(nym.2002)
```

b. (2 pts) How many participants are recorded in this data set (Hint: you can either use `length()` with an individual variable, or use the `nrow()` function on the entire data frame)?

```
> nrow(nym.2002)
[1] 1000
```

c. (3 pts) Create a variable called `time.hrs` that contains times converted to hours (you can leave the result as a decimal value).

```
> time.hrs = nym.2002$time / 60.0  # preferred
```

or:

```
> time.hrs = nym.2002["time"] / 60.0
```

d. (4 pts) Create a new data frame that contains the data for only those runners from New York State (Hint: select a subset of rows using the `home` variable).

```
> ny = nym.2002[nym.2002$home=="NY",]  # preferred
```

or:

```
> ny = nym.2002[nym.2002$"home"=="NY",]
```

or:

```
> ny = nym.2002[nym.2002["home"]=="NY",]
```

e. (3 pts) What percentage of the runners came from within the state?

```
> (nrow(ny) / nrow(nym.2002)) * 100.0
[1] 28.4
```

5 Reading in Data from a File (11 pts)

There is a data set available on d2l called `BrainBodyWeight.csv`, containing brain weights and body weights for various terrestrial (land) mammals.

a. (2 pts) Create a directory somewhere on your computer for `R` data sets in this class (you don't need to use `R` to do this). Download the data file from d2l into that folder, and in `R`, set your working directory there.

b. (2 pts) Read the file into `R` using `read.csv()`, and save it as a data frame (i.e., assign the output of `read.csv()` to a variable). Be sure to use the option `header = TRUE` so that `R` interprets the first row of the data as variable names.

```
> bbw = read.csv("/mypath/Labs/datasets/BrainBodyWeight.csv", header=TRUE)
```

c. (2 pts) Have `R` print out the variable names.

```
> names(bbw)
[1] "Terrestrial.mammal"    "body.weight.kilograms"
[3] "brain.weight.grams"    "log.body."
[5] "log.brain."
```

d. (2 pts) Use `attach()` to make the individual variables directly accessible.

```
> attach(bbw)
```

e. (2 pts) How much does the heaviest brain weigh?

```
> max(brain.weight.grams)  # preferred, this works since we attached bbw
[1] 5712
```

or:

```
> max(bbw$brain.weight.grams)  # preferred
[1] 5712
```

or:

```
> max(bbw["brain.weight.grams"])
[1] 5712
```

f. (1 pt) When you're finished, `detach()` the data frame, to keep the workspace clean.

```
> detach(bbw)
```