Stat 5050: Introduction to R

Name: Peter Kurtz

Homework Assignment 02

40 Points

General Instructions

For this second homework assignment, you have to work with RMarkdown or knitr/Sweave. You can create your own RMarkdown (.Rmd) file, based on files from class and from Homework 1, copy the question numbers and the answer options into your .Rmd file, and knit that file into a pdf file. **Alternatively** (and much easier!!!), use this .Rnw file as a template, just fill in the answers into the provided spaces, and knit into a pdf file.

Only the final resulting pdf file (from .Rmd or .Rnw) has to be submitted via Canvas. As previously stated, I would like to encourage potential and current MS and PhD students to work with .Rnw and LaTeX instead of .Rmd.

This homework will be graded on completeness. You will learn more if you are honest when you check the boxes. No need to correct your initial answer if the answer from R is different. Just explain why it is different and what you missed initially.

Do not forget to put your name at the top!

(i) (16 Points) **What will R do?!?**

Based on the classroom notes and without using R, mark your best guess of the answer to each of the following questions with an [x]. Then check your answers with R. If you got any answer wrong, comment on what you learned when you ran the code in R, i.e., what did you initially misinterpret or overlook in the R code?

We want to assign the value -5 to a variable. Assume that we *sequentially* execute the following expressions at the start of an R session with no previous workspace loaded (or the workspace being erased — recall how we did this in class). Which of the following expressions will do such an assignment of -5? Think of valid variable names, assignment operators, and breaks. Read carefully!

```
# 1 The following expression will
#      [ ] make the correct assignment
#      [ ] do something (but not the correct assignment)
#      [x] result in an error

x1y10 == -5; x1y10
```

The actual result in R is:

```
Error: object 'x1y10' not found
```

Mark one of these two options with an x, i.e., [x]:
[x] My answer and the answer from R are matching.
[ ] My answer and the answer from R are **not** matching. I made this mistake /
overlooked this detail: **Explain!**

```
# 2 The following expression will
#      [x] make the correct assignment
#      [ ] do something (but not the correct assignment)
#      [ ] result in an error

x1y10 = - 5; x1y10
```

The actual result in R is:

```
[1] -5
```

Mark one of these two options with an x, i.e., [x]:
[x] My answer and the answer from R are matching.
[ ] My answer and the answer from R are **not** matching. I made this mistake /
overlooked this detail: **Explain!**

```
# 3 The following expression will
#      [ ] make the correct assignment
#      [ ] do something (but not the correct assignment)
#      [x] result in an error

efg99 < - 5; efg99
```

The actual result in R is:

```
Error: object 'efg99' not found
```

Mark one of these two options with an x, i.e., [x]:
[x] My answer and the answer from R are matching.
[ ] My answer and the answer from R are **not** matching. I made this mistake /
overlooked this detail: **Explain!**

```
# 4 The following expression will
#       [x] make the correct assignment
#       [ ] do something (but not the correct assignment)
#       [ ] result in an error


_myXyz = -5; _myXyz
```

The actual result in R is:

```
Error: unexpected symbol in "_myXyz"
```

Mark one of these two options with an x, i.e., [x]:
[ ] My answer and the answer from R are matching.
[x] My answer and the answer from R are **not** matching. I made this mistake /
overlooked this detail: I did not realize that underscores could not start a variable
name. I assumed that since underscores are allowed in r variables the r variables
could start with an underscore.

```
# 5 The following expression will
#       [x] make the correct assignment
#       [ ] do something (but not the correct assignment)
#       [ ] result in an error


x999_y1001 = -5; x999_y1001
```

The actual result in R is:

```
[1] -5
```

Mark one of these two options with an x, i.e., [x]:

[x] My answer and the answer from R are matching.

[ ] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: **Explain!**

```
# 6 The following expression will
#      [x] make the correct assignment
#      [ ] do something (but not the correct assignment)
#      [ ] result in an error


x5_999<--5;x5_999
```

The actual result in R is:

```
[1] -5
```

Mark one of these two options with an x, i.e., [x]:

[x] My answer and the answer from R are matching.

[ ] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: **Explain!**

```
# 7 The following expression will
#      [x] make the correct assignment
#      [ ] do something (but not the correct assignment)
#      [ ] result in an error


jjj <- -
5; jjj
```

The actual result in R is:

```
[1] -5
```

Mark one of these two options with an x, i.e., [x]:

[x] My answer and the answer from R are matching.

[ ] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: **Explain!**

```
# 8 The following expression will
#       [ ] make the correct assignment
#       [ ] do something (but not the correct assignment)
#       [x] result in an error

jjj < - - 5
```

The actual result in R is:

```
[1] TRUE
```

Mark one of these two options with an x, i.e., [x]:

[ ] My answer and the answer from R are matching.

[x] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: I assumed that r would recognize that it should have been an assignment. The extra space would result in an error. Now I see that double negatives are allowed in r.

(ii) (24 Points) **Random Numbers from a Binomial Distribution**

In R, look at the documentation of the function `rbinom`, using `?rbinom`. Suppose we want to generate 5 pseudo-random values from a binomial distribution with `size = 12` and `prob = 0.5`. Assume that we execute these expressions at the start of an R session with no previous workspace loaded. Which of the following commands is/are suitable for this task? **Mark those with an [x].** As in question (i), first answer this without using R, then correct your answers using R and explain briefly what you learned.

```
# 1 The following expression will
#      [x] create the correct 5 pseudo-random values
#      [ ] do something (but not the correct thing)
#      [ ] result in an error


dbinom(5, 12, 0.5)
```

The actual result in R is:

```
[1] 0.1933594
```

Mark one of these two options with an x, i.e., [x]:
[ ] My answer and the answer from R are matching.
[x] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: I did not realize the d in the front.

```
# 2 The following expression will
#      [x] create the correct 5 pseudo-random values
#      [ ] do something (but not the correct thing)
#      [ ] result in an error


rbinom(size = 12, prob = 0.5, n = 5)
```

The actual result in R is:

```
[1] 9 7 3 5 6
```

Mark one of these two options with an x, i.e., [x]:

[x] My answer and the answer from R are matching.

[ ] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: **Explain!**

```
# 3 The following expression will
#       [ ] create the correct 5 pseudo-random values
#       [x] do something (but not the correct thing)
#       [ ] result in an error


rbinom(5, 6, 0.5)
```

The actual result in R is:

```
[1] 2 1 1 3 2
```

Mark one of these two options with an x, i.e., [x]:

[x] My answer and the answer from R are matching.

[ ] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: **Explain!**

```
# 4 The following expression will
#       [ ] create the correct 5 pseudo-random values
#       [x] do something (but not the correct thing)
#       [ ] result in an error


rbinom(5, prob = 0.5, n = 12)
```

The actual result in R is:

```
[1] 3 2 3 3 4 3 0 2 4 3 4 3
```

Mark one of these two options with an x, i.e., [x]:

[x] My answer and the answer from R are matching.

[ ] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: **Explain!**

```
# 5 The following expression will
#      [x] create the correct 5 pseudo-random values
#      [ ] do something (but not the correct thing)
#      [ ] result in an error


rbinom(5, 0.5, 12)
```

The actual result in R is:

```
Warning message:
In rbinom(5, 0.5, 12) : NAs produced
```

Mark one of these two options with an x, i.e., [x]:

[ ] My answer and the answer from R are matching.

[x] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: I thought R would recognize that the decimal should go under prob.

```
# 6 The following expression will
#      [x] create the correct 5 pseudo-random values
#      [ ] do something (but not the correct thing)
#      [ ] result in an error


rbinom(5, 12, 0.5)
```

The actual result in R is:

```
[1] 3 2 7 6 8
```

Mark one of these two options with an x, i.e., [x]:

[x] My answer and the answer from R are matching.

[ ] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: **Explain!**

```
# 7 The following expression will
#       [x] create the correct 5 pseudo-random values
#       [ ] do something (but not the correct thing)
#       [ ] result in an error

rbinom(5, prob = 0.5, size = 12)
```

The actual result in R is:

```
[1] 8 8 5 5 4
```

Mark one of these two options with an x, i.e., [x]:

[x] My answer and the answer from R are matching.

[ ] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: **Explain!**

```
# 8 The following expression will
#       [ ] create the correct 5 pseudo-random values
#       [x] do something (but not the correct thing)
#       [ ] result in an error

rbinom(n = 12, prob = 0.5, size = 5)
```

The actual result in R is:

```
[1] 4 0 4 3 2 2 2 1 2 2 2 2
```

Mark one of these two options with an x, i.e., [x]:

[x] My answer and the answer from R are matching.

[ ] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: **Explain!**

```
# 9 The following expression will
#       [ ] create the correct 5 pseudo-random values
#       [ ] do something (but not the correct thing)
#       [x] result in an error

rbinom(s = 12, p = 5, n = .5)
```

The actual result in R is:

```
integer(0)
```

Mark one of these two options with an x, i.e., [x]:
[ ] My answer and the answer from R are matching.
[x] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: I thought the s would cause an error. Now I remember that the inputs can have abbreviations.

```
# 10 The following expression will
#      [ ] create the correct 5 pseudo-random values
#      [x] do something (but not the correct thing)
#      [ ] result in an error


rbinom(12, prob = 0.5, 5)
```

The actual result in R is:

```
[1] 4 3 4 1 1 3 3 3 4 3 3 1
```

Mark one of these two options with an x, i.e., [x]:
[x] My answer and the answer from R are matching.
[ ] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: **Explain!**

```
# 11 The following expression will
#      [x] create the correct 5 pseudo-random values
#      [ ] do something (but not the correct thing)
#      [ ] result in an error


rbinom(5, prob = 0.5, 12)
```

The actual result in R is:

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

Mark one of these two options with an x, i.e., [x]:

[x] My answer and the answer from R are matching.

[ ] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: **Explain!**

```
# 12 The following expression will
#       [ ] create the correct 5 pseudo-random values
#       [ ] do something (but not the correct thing)
#       [x] result in an error

rbinom(n = 5, prob = 0.5, n = 12)
```

The actual result in R is:

```
Error in rbinom(n = 5, prob = 0.5, n = 12) :
  formal argument "n" matched by multiple actual arguments
```

Mark one of these two options with an x, i.e., [x]:

[x] My answer and the answer from R are matching.

[ ] My answer and the answer from R are **not** matching. I made this mistake / overlooked this detail: **Explain!**