week5a_lab

Naga Soundari Balamurugan

10/23/2018

Collaborators: Jayashree Raman

Instructions:

1. Download the `week5a_lab.Rmd` file from Canvas. Open `week4b_lab.Rmd` in RStudio and supply your solutions to the assignment by editing `week5a_lab.Rmd`.
2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text. If you are using more than just a standard function that you found from another source, please credit the source in the comments.
4. Collaboration on labs is encouraged, but students must turn in an individual assignments. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF` or `Knit Word`, rename the R Markdown file to `YourLastName_YourFirstName_Lab5a.Rmd`, knit a PDF or DOC and submit both the PDF/DOC and the Rmd file on Canvas.

Setup:

```r
# Load any specific libraries you need here
library(bit64)
```

```
## Loading required package: bit
```

```
## Attaching package bit
```

```
## package:bit (c) 2008-2012 Jens Oehlschlaegel (GPL-2)
```

```
## creators: bit bitwhich
```

```
## coercion: as.logical as.integer as.bit as.bitwhich which
```

```
## operator: ! & | xor != ==
```

```
## querying: print length any all min max range sum summary
```

```
## bit access: length<- [ [<- [[ [[<-
```

```
## for more help type ?bit
```

```
##
## Attaching package: 'bit'
```

```
## The following object is masked from 'package:base':
##
##     xor
## Attaching package bit64

## package:bit64 (c) 2011-2012 Jens Oehlschlaegel

## creators: integer64 seq :

## coercion: as.integer64 as.vector as.logical as.integer as.double as.character as.bin

## logical operator: ! & | xor != == < <= >= >

## arithmetic operator: + - * / %/% %% ^

## math: sign abs sqrt log log2 log10

## math: floor ceiling trunc round

## querying: is.integer64 is.vector [is.atomic} [length] format print str

## values: is.na is.nan is.finite is.infinite

## aggregation: any all min max range sum prod

## cumulation: diff cummin cummax cumsum cumprod

## access: length<- [ [<- [[ [[<-

## combine: c rep cbind rbind as.data.frame

## WARNING don't use as subscripts

## WARNING semantics differ from integer

## for more help type ?bit64

##
## Attaching package: 'bit64'

## The following object is masked from 'package:bit':
##
##     still.identical

## The following objects are masked from 'package:base':
##
##     %in%, :, is.double, match, order, rank
```

Problem 1a: What is the sample space of tossing a coin twice? Use S = {. . . } notation.

Sample space S = {HH, HT, TH, TT}. Number of elements in S is 4.

Problem 1b: Given the Sample Space S = {aqua, red, green, purple, hazel, sandstone, lime, slate} and the event A = {red, green, lime}; what is A^c (the complement of A)?

A^{c} = {aqua, purple, hazel, sandstone, slate}

Problem 2: Exam Questions

A teacher assigns a True/False exam with 12 questions. What is the probability that a test taker who guesses at random answers 8 correctly?

```
# Feel free to perform calculations here
SampleSpace <- 2^12
answers8right <- factorial(12)/(factorial(8) * factorial(12 - 8))

probRightAns <- answers8right/SampleSpace
probRightAns
```

```
## [1] 0.1208496
```

Describe your answer here: Since we have 12 questions with 2 options, the sample space would be 2^12. Now we need to look for the probability of guessing 8 right answers. The number of answers that contain exactly 8 right answers will be equal to the number of combinations that can be found with 8 right answers and 12 ??? 8 wrong answers. This gives the number of combinations to guess 8 right answers. Inorder to calculate the probability, lets divide no of combinations by sample space.

Problem 3: Fair Die Tossed 100 times

A fair die is tossed 100 times. Find the probability that there is at least one 3.

```
# Feel free to perform calculations here
inAllRolls <- (5/6) ^ 100
atLeastOne <- 1 - inAllRolls
atLeastOne
```

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

Describe your answer here: Probability of getting 3 in one roll = 1/6 Probability of not getting 3 = 5/6 Probability of not getting 3 in all 100 rolls = 5/(6 ^ 100) Thus by the probability rule, probability of not getting 3 in all 100 rolls + probability of getting at least one 3 = 1 Hence, probability of getting at least one 3 = 1 - probability of not getting 3 in all 100 rolls which makes probability of getting at least one 3 = 1 - (5/(6 ^ 100))

Additional Learning: Mathematical Notation in R Markdown

Take a look at this link: https://www.calvin.edu/~rpruim/courses/s341/S17/from-class/MathinRmd.html to see some examples of how to create mathematical notation within your .Rmd document.