PROBLEM SET # 1
EC/ACM/CS 112: Bayesian Statistics
Caltech

Submission instructions	>> Create a pdf of the R Notebook with your solutions (details below) >> Submit in Canvas
Additional files included in the problem set package	>> Solutions template (.Rmd file) >> R basic cheat sheet >> R markdown cheat sheet

BACKGROUND

- In this course we will use the R software, which is one of the core tools used by practitioners of Bayesian statistics.
- Although of you this will entail learning the basics of a new programming language, exit surveys from previous years show that most students were happy to have acquired this new toolkit by the end of the course, and that they planned to use it in their own work.

GOALS

- Provide you with an introduction and basic practice with R
- The skills acquired in this problem set will be required in all later work in the course.

POTENTIALLY USEFUL MATERIALS	
R cheat sheets	https://www.rstudio.com/resources/cheatsheets/
Guide to R	https://cran.r-project.org/doc/contrib/Owen-TheRGuide.pdf
RStudio Keyboard Shortcuts	https://support.rstudio.com/hc/en-us/articles/200711853-Keyboard-Shortcuts
Free on-line books on R for data science	https://r4ds.had.co.nz/ https://rstudio-education.github.io/hopr/

PART 1: GETTING STARTED WITH R

Step 1. Install R in your own computer

- Download the latest version of R from official repository (https://cran.r-project.org/) and install it in your machine.

Step 2. Install RStudio in your own computer

- Download the latest free version of RStudio (https://www.rstudio.com/) and install it in your machine.
- Important: RStudio must be installed after Step 1 has been completed.
- RStudio is not necessary to use R, but is an extremely useful and popular integrated development environment in data science. It supports debugging, workspace management, plotting and much more.

Step 3. Complete an R tutorial

- If you are new to R, or have very limited experience, I strongly recommend that you spend 4-6 hours completing a good on-line tutorial on R
- There are a large number of good free tutorials online
- My recommendation is that you work through most of the outstanding book ***Hands-On Programming with R***, by Garret Grolemund. The book is freely available on- line (https://rstudio-education.github.io/hopr/).
- If you end up liking R and want to learn more, the book ***R for Data Science*** (https://r4ds.had.co.nz/index.html), by the same author, is a more advanced invaluable resource
- If you prefer to look for alternative tutorials, the following blogpost provides some useful suggestions: https://www.r-bloggers.com/how-to-learn-r-2/

Step 4. Learn how to install and import R packages

- Your initial installation of R will come with a set of base packages.
- However, what makes R so popular and powerful is the large number of packages that are constantly being developed to improve data science with R.
- To take advantage of this, you need to learn how to install packages (using the **install.packages** function) and to import packages (using the **library** function).
- The following blog posts provide an introduction to both functions: https://www.r-bloggers.com/installing-r-packages/

Step 5. Learn how to use R notebooks

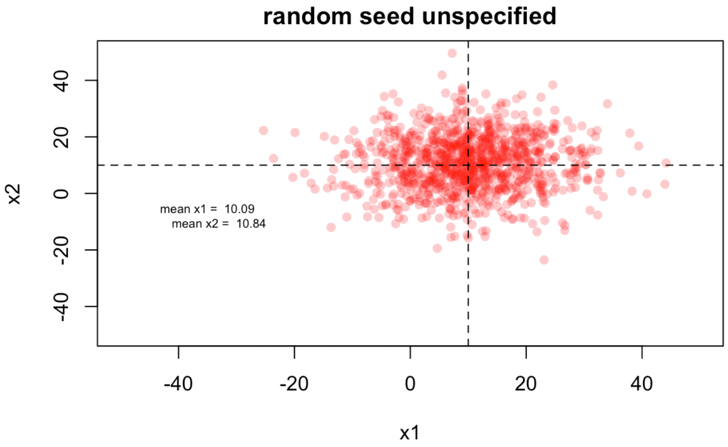
- A good practice in statistical science is to R notebooks (or an equivalent took like Jupyter Notebooks) to document your work. These notebooks are useful because they will allow you to keep track and share your code and results in a single file, thus increasing the transparency and replicability of your work.
- In this course we will use this practice by using a tool called R Markdown, which allows you to produce R notebooks using R Studio in a user friendly way.
- All problem sets will provide you with an R Markdown template (.Rmd extension) that you should load into R Studio and edit to complete your work.
- Read the short chapter 27 in the R for Data Science book (https://r4ds.had.co.nz/r-markdown.html) to learn the basics.

PART 2: BASIC R PRACTICE

Step 1. Programming replicable stochastic simulations

1.A.

- In this section, do not specify the seed for the random number generator.
- Simulate 1,000 independent draws for a Gaussian random variable with mean 10 and standard deviation 10, and store it in a vector x1.
- Simulate the same random variable again and store it in a vector x2.
- Generate the following plot using your own data. The plotting parameters should match (e.g., colors, transparency of symbols, symbol type, axis size, …)



1.B.

- Repeat all of the steps in exercise in 1.A., but in this case set the seed of the random number generator to 2021, before generating each random sample.

- The plot should include a 45-degree line, and the title should be “random seed specified”

Step 2. Simulating the hot-hand in basketball

In this problem you are going to simulate the scoring streaks of basketball players. You will simulate data from 10,000 different NBA players. Each of these simulates the scoring patterns of a given player in a given game, under the assumption that players always take 25 shots per game. In your simulate data, a basket is coded as a 1, and a miss as a 0.

Your simulations should assume that all shots are independent.

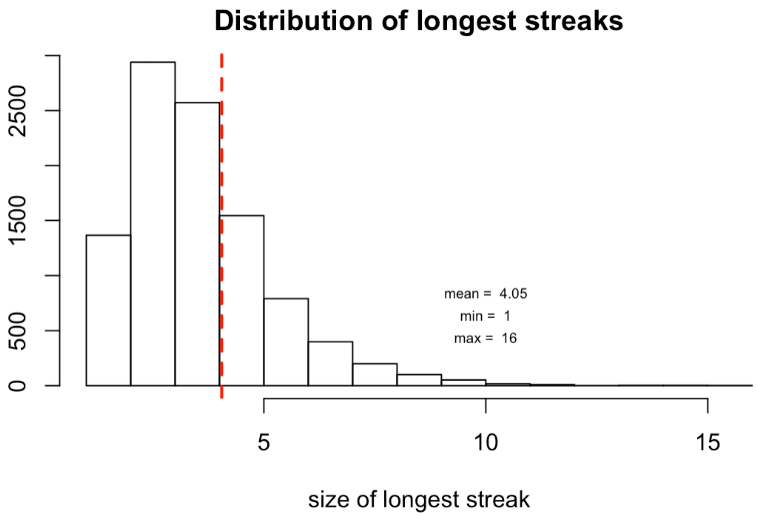
At the outset of your simulations, set the random seed to 2021.

2.A.

- Write the following two functions, which you will then use in the simulations below:
    - A function **simulate_player**, which takes as input the probability of scoring each basket, and returns a list of the hits and misses for the 25 shots taken in a game.
    - A function **count_sequence**, which takes as input a list of hits and misses, and returns the size of the longest streak of baskets in the list.
- Be careful and double check that your functions work well in all cases.

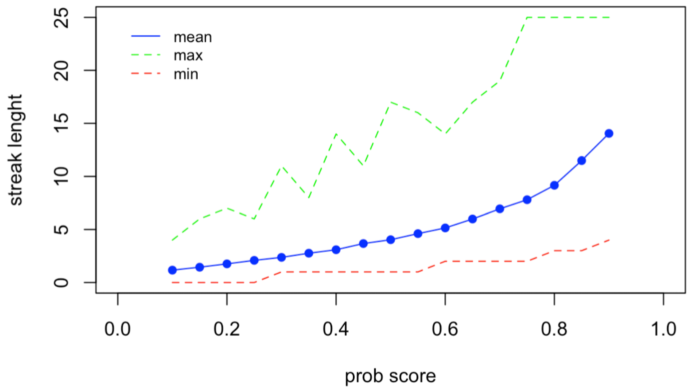
2.B.

- Using the functions in 2.A, simulate a dataset with 10,000 players assuming that the probability of scoring each shot is 0.5.
- Use the results of your simulation to generate the following plot. As before, try to match as many details as possible.



2.C.

- Now investigate how the results change with the probability of scoring each shot.
- To do this, repeat the simulations in 1.B for values of *pHit = 0.1, 0.15, …, 0.9*, where *pHit* denotes the probability of scoring in any given shot.
- Use the results of your simulation to generate the following plot. As before, try to match as many details as possible.


mean
max
min
streak lenght
prob score