R Quiz Practice Instructions
18.05, Spring 2022

Overview

These are practice questions for the quiz. The format is similiar to the R studios and will be identical to the R Quiz.

This is much longer than the actual quiz. As usual, you should read and follow all instructions.

You should be sure to look at problem 7, which is about looking up how to do something on Google.

Download the zip file

- You should have downloaded the file `mit18_05_s22_RQuizPractice.zip` from our MITx site.
- Unzip it in your 18.05 studio folder.
- You should see the following R files
  `mit18_05_s22_RQuizPractice.r`
  `mit18_05_s22_RQuizPractice-test.r`

  and the following other files

  `mit18_05_s22_RQuizPractice-test-answers.html`

General instructions

- Using the Session menu, set the working directory to source file location. (This is a good habit to develop!)
- Answer the questions in the detailed instructions just below. Your answers should be put in `mit18_05_s22_RQuizPractice.r`
- For each question, you will complete the code for the corresponding function.
- As usual, use the the function's arguments and any other code given in the function.
- Do not print out things that are not asked for.
- Solution code is posted alongside the zip file.
- For the real quiz, before uploading your code, soure the code file and run all the tests.

Detailed instructions for this practice

0. Clean your space

The two lines here will clear the environment and console every time you source the file.

Problem 1

Problem 1 will cover some basic R: pseudo-random sampling, plotting, basic sample statistics.

Problem 1a. Here you will finish the code for the function
`rquiz_practice_problem_1a(n_samples, mu, sigma)`

Arguments:
`n_samples` = size of the sample to generate
`mu` = mean of the underlying normal distribution
`sigma` = standard deviation of the underlying normal distribution

The function should generate and print a sample of size `n_samples` from a normal distribution with mean mu and standard deviation sigma.

Problem 1b. Here you will finish the code for the function
`rquiz_practice_problem_1b = function(n_samples, size, theta)`

Arguments:
`n_samples` = size of the sample to generate
`size` = number of Bernoulli trials
`theta` = probability of success in each Bernoulli trial

The function should generate and print a sample of size `n_samples` from a Binomial(size, theta) distribution.

Problem 1c. Here you will finish the code for the function
`rquiz_practice_problem_1c(sample_space, n_samples)`

Arguments:
`sample_space` = the list of outcomes to sample from
`n_samples` = number of the sample to generate

Generate `n_samples` with replacement from the list given by `sample_space`.

Print the result.

Problem 1d. Here you will finish the code for the function
`rquiz_practice_problem_1d(n)`

Arguments:
`n`: the problem asks you to generate a permutation of the numbers 1:n

Use the sample function to generate a permutation of the numbers 1 to n.

Print the result

Problem 1e. Here you will finish the code for the function
`rquiz_practice_problem_1e(w, x_1)`

Arguments:
`w` = angular frequency (for sin(w*x) and cos(w*x))
`x_1`: plot from 0 to `x_1`

Plot the graph of $\sin(w * x)$ over the interval 0 to `x_1`

Add the graph of $\cos(w * x)$ in a different color.

Add a vertical line at $x = 0$ and a horizontal line at $y = 0$.

Problem 1f. Here you will finish the code for the function
`rquiz_practice_problem_1f(n_samples, a, b)`

Arguments:
`n_samples` = number of samples to generate from a Uniform(a,b) distribution.
`a` = left endpoint for the Uniform(a,b) distribution.
`b` = right endpoint for the Uniform(a,b) distribution.

Generate `n_samples` independent values from a Uniform(`a, b`) distribution.

Compute and print the sample mean, median, variance, standard deviation and first and third quartiles of this data. (For this, you may need to get help on the `quantile` function.)

Problem 1g. Here you will finish the code for the function
`rquiz_practice_problem_1g(n_samples, a, b)`

`n_samples` = number of samples to generate from a Uniform(a,b) distribution.
`a` = left endpoint for the Uniform(a,b) distribution.
`b` = right endpoint for the Uniform(a,b) distribution.

- Generate arrays x and y: each of them should have `n_samples` independent values from a Uniform(a, b) distribution.
- Let z = (x + y)/2.
- Compute the sample covariance and correlation of x and z.
- Print the results.
- Make a scatter plot of y vs x.

Problem 1h. Here you will finish the code for the function
`rquiz_practice_problem_1h(n_samples, mu1, sigma1, mu2, sigma2, alpha, mu0, sigma0)`

Arguments:
`n_samples` = number of samples to generate from each normal distribution.
`mu1` = mean for the first normal distribution.
`sigm1` = standard deviation for the first normal distribution.
`mu2` = mean for the second normal distribution.
`sigm2` = standard deviation for the second normal distribution.
`alpha` = significance level for tests.
`mu0, sigma0` are the mean and standard deviation for H0. (sigma0 is not used in the t-test comparing means).

Generate `x1` and `x2`: both samples of size `n_samples` from normal distributions.

`x_1` should be from a Norm(`mu1, sigma1`) and `x_2` should be from a Norm(`mu2, sigma2`).

Print out all the arguments to this function.

1h(i) Run a z-test with null hypothesis H0: `x_1` is drawn from a normal distribution with mean `mu0` and known variance `sigma0`.

Make the test two-sided.
Use signifcance level `alpha`.

Print out the z-statistic, p-value and the conclusion of the test.

1h(ii) Run a t-test with null hypothesis H0: `x_1` is drawn from a normal distribution with mean `mu0` and unknown variance.

Make the test two-sided.
Use signifcance level `alpha`.

You can use the R function `t.test()`.

Print the t-statistic, p-value and the conclusion of the test.

1h(iii) Test whether `x_1` and `x_2` are drawn from normal distributions with the same mean.

Make the test two-sided.
Assume equal variances.
Use significance level `alpha`.

Print the t-statistic, p-value and the conclusion of the test.

1h(iv) Test whether `x_1` and `x_2` are drawn from normal distributions with the same variance.

Use significance level `alpha`.

Print the test statistic, p-value and the conclusion of the test.

Problem 2

Problem 2. Here you will finish the code for the function

`rquiz_practice_problem_2(prior, next_roll)`

Arguments:
`prior` = prior probability for the dice choices (5 numbers, usual order)
`next_roll` = the problem asks for the posterior predictive probability of rolling `next_roll` on the next roll.

Run a complete simulation of randomly choosing a die from our standard Platonic dice (4, 6, 8, 12, and 20 sided) based on the given prior. Then rolling it once and updating to a posterior probability for the type of die chosen.

Print out the prior, chosen die, roll and posterior.

Compute and print out the posterior predictive probability of rolling `next_roll` on the next roll.

You can use standard_likelihood_table loaded by the code given in the `mit18_05_s22_RQuizPractice.r` file.

Problem 3

The MIT Administration has decided that MIT students should be able to vote for their school mascot. They are now able to vote between Tim the Beaver, Hack the Hawk, and Punt the Platypus.

Problem 3a. Here you will finish the code for the function
`rquiz_practice_problem_3a(n_rats)`

Arguments:
`n_rats` = number of Brass Rats Tim has

On the morning of his platform speech, Tim the Beaver is deciding which Brass Rats to wear. They will wear 2 Brass Rats, one on their left hand, and one on their right hand. They have `n_rats` Brass Rats to choose from. How many ways can Tim wear two of their Brass Rats? (Order matters)

Compute and print out the number of ways Tim can wear their rings.

Problem 3b. Here you will finish the code for the function
`rquiz_practice_problem_3b(n_ways)`

Arguments:
`n_ways` = minimum number of ways Tim wants to be able to wear their Brass Rats.

How many Brass Rats would Tim hypothetically need for the number of possibilities to be over `n_ways`? Use a for loop or any other efficient means.

Problem 4

Continuing problem 3. In past years fraction of students supporting Tim has been `theta_tim`. An 18.05 student runs a poll to see if that support has dropped. They poll `n_students` MIT students.

Problem 4a. Here you will finish the code for the function
`rquiz_practice_problem_4a(n_students, theta_tim)`

Arguments:
`n_students` = the number of students polled
`theta_tim` = true fraction of the student population that supports Tim

The number of students who support Tim follows a certain distribution. Plot this distribution.

Problem 4b. Here you will finish the code for the function
`rquiz_practice_problem_4b(n_students, n_support_tim, theta_H0, alpha)`

Arguments:
`n_students` = the number of students polled.
`n_support_tim` = the number of who support Tim in the poll.
`theta_H0` = For H0, the fraction of the student population that supports Tim
`alpha` = significance level for the NHST

To test the conjecture that Tim's support has dropped below `theta_H0`, they run a signifi-

cance test.

For the test, use the data given: `n_support` and significance level `alpha`.

Print the null and alternative hypotheses.

Print the rejection region, p-value and the conclusion of the test.

Problem 5

Hack the Hawk takes the Number 1 Bus to commute to MIT's campus. The 1 Bus is known to be late by Y minutes, where Y follows an exponential distribution with lamba = 0.2 (units of 1/minutes).

Problem 5a. Here you will finish the code for the function

`rquiz_practice_problem_5a(lambda)`

Arguments:
`lambda` = rate parameter for the exponential distribution

Plot the pdf of Exp(lambda). Use the range 0 to 4/lambda.

Problem 5b. Here you will finish the code for the function

`rquiz_practice_problem_5b(lambda, n_bus_trips)`

Arguments:
`lambda` = rate parameter for the exponential distribution
`n_bus_trips` = the number of Hack's bus trips to simulate

Hack the Hawk has taken the bus to MIT `n_bus_trips` times. Simulate the lateness of the bus over `n_bus_trips` days.

5b(i) Plot a frequency histogram of the simulated data. Use bin_width of 1.0.

5b(i) Plot a density histogram of the simulated data. Use bin_width of 1.0. Add a plot of the pdf for Exponential(lambda).

Problem 6

Work is stressful. To relax, Punt plays roulette at a nearby casino. She plays 10 times every day, for 7 days. She bets on red each time. (The roulette wheel has 18 red, 18 black, and 2 green slots)

If she loses, her loss is $1. If she wins her gain is $1.

Problem 6a. Here you will finish the code for the function

`rquiz_practice_problem_6a(n_bets_per_day, n_days)`

Arguments:
`n_bets_per_day` = number of bets Punt makes each day.
`n_days` = number of days Punt plays

Compute and print the expected value and variance of one bet.

Compute and print out the expected value and variance one day's winnings.

Compute and print out the expected value and variance `n_days` winnings.

Problem 6b. Here you will finish the code for the function
`rquiz_practice_problem_6b(n_bets_per_day, n_days, n_trials)`

Arguments:
`n_bets_per_day` = number of bets Punt makes each day.
`n_days` = number of days Punt plays `n_trials` = number of simulations of n_days of betting to simulate.

6b(i) Write R code to simulate Punt's total earnings for `n_days`. Run `n_trials` trials and plot the density histogram of the results. Add a graph of the normal curve which approximates this histogram.

6b(ii) Explain why the normal distribution used in 6b(i) approximates the density.

Problem 7

This problem is about using Google and R help to solve a problem.

Problem 7. Here you will finish the code for the function
`rquiz_practice_problem_7(mit_times, harvard_times, alpha)`

Arguments:
`mit_times` = list of times for the MIT students
`harvard_times` = list of times for the Harvard students
`alpha` = significance level for NHST

MIT challenged Harvard to a race to determine which school had, on average, faster runners. Both schools chose some random students and excused them from finals to run the race.

After the race you were called in to analyze the data. Unfortunately, the statistics professors at both schools were gone and the only instruction they had left was to use the Wilcoxon Rank Sum test to see if the mean ranks were the same.

Figure out how to do this in R and write code that expects two vectors of data `mit_times` and `harvard_times` containing the finishing time for the students who raced.

Your code should run the Wilcoxon Rank Sum test and print:

1. the test statistic,
2. which if either school appears faster on average,
3. the p_value of the test.

You may use any function you can find in R.

Testing your code

For each problem, we ran the problem function with certain parameters. You can see the function call and the output in `mit18_05_s22_RQuizPractice-test-answers.html`. If you call the same function with the same parameters, you should get the same results as in `mit18_05_s22_RQuizPractice-test-answers.html` – if there is randomness involved the answers should be close but not identical.

For your convenience, the file `mit18_05_s22_RQuizPractice-test.r` contains all the function calls used to make `mit18_05_s22_RQuizPractice-test-answers.html`.

Before uploading your code

You won't upload this code. But you should still practice what you should do before uploading!

1. Make sure all your code is in your R file. Also make sure it is all inside the functions for the problems.
2. Clean the environment and plots window.
3. Source the file.
4. Call each of the problem functions with the same parameters as the test file `mit18_05_s22_RQuizPractice-test-answers.html`.
5. Make sure it runs without error and outputs just the answers asked for in the questions.
6. Compare the output to the answers given in `mit18_05_s22_RQuizPractice-test-answers.html`.

18.05 Introduction to Probability and Statistics
Spring 2022