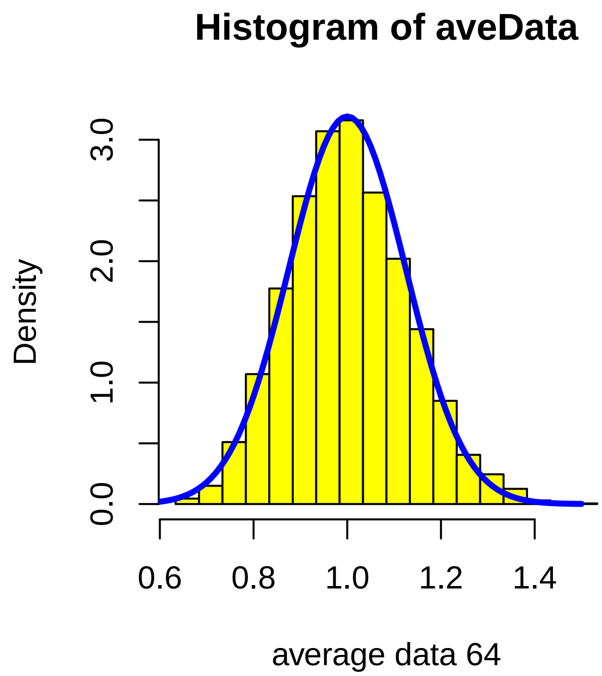
Studio 3 Histograms
18.05, Spring 2022



Overview of the studio

This studio is about making histograms of data. We will use the function `hist()` to do most of the work. When we expect the histogram to match a known pdf, we will add a graph of the pdf on top.

R introduced in this studio

The R needed is all introduced in `mit18_05_s22_studio3-samplecode.r`. We will use

1. Pseudo-random generators for various distributions: `rnorm(), rexp(), runif()`.
2. `hist(...)` for making histograms
3. lines() for adding graphs to existing plots

Download the zip file

- You should have downloaded the file `mit18_05_s22_studio3.zip` from our MITx site.
- Unzip it in your 18.05 studio folder.

- You should see the following R files
  `mit18_05_s22_studio3.r`
  `mit18_05_s22_studio3-samplecode.r`
  `mit18_05_s22_studio3-test.r`

  and the following other files

  `mit18_05_s22_studio3-test-answers.html`

Prepping R Studio

- In R studio, open `mit18_05_s22_studio3-samplecode.r` and `mit18_05_s22_studio3.r`
- 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_studio3.r`
- Solution code will be posted tomorrow at 10 pm

Detailed instructions for the studio

• Go through **mit18_05_s22_studio3-samplecode.r** as a tutorial. It is relatively short.

Problem 1. Frequency and density histograms.

After doing this problem, look at the frequency and density histograms. Note especially, the scales on the vertical axis.

Problem 1a. Here you will finish the code for the function

```
studio3_problem_1a(rate, nsamples)
```

This function should draw a *frequency* histogram of (simulated) exponential data.

The arguments to this function are:

```
rate = rate parameter for an exponential distribution
nsamples = the size of the sample to use for the histogram
```

The function should do the following.

1. Generate `nsamples` from an exponential distribution with the given rate parameter.
2. Draw a *frequency* histogram of the (simulated) data.

Follow the sample code in explicitly setting up the bins for the histogram. We set bin_width for you in our part of the code.

You should use the built in R function `rexp()` to generate the random sample.

Problem 1b. Here you will finish the code for the function

```
studio3_problem_1b(rate, nsamples)
```

This is very similar to 1a.

The arguments to this function are:
```
rate = rate parameter for an exponential distribution
nsamples = the size of the exponential sample
```
As in 1(a), we define bin_width for you in our part of the code.

The function should do the following.

1. Generate `nsamples` from an exponential distribution with the given rate parameter.

2. Draw a *density* histogram of the (simulated) data.

3. Add a graph of the probability density (pdf) of the underlying exponential distribution to the plot.

Problem 2. In this problem we will draw density histograms of the average of many exponential samples. The goal is to see how the density of the averages changes as the number averaged increases. We will see graphically how it approaches a normal distribution as the number of terms in the average increases.

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

`studio3_problem_2a(rate, nsamples)`

The arguments to this function are:
```
rate = rate parameter for an exponential distribution
nsamples = the size of the exponential sample
```
As before, we define bin_width for you in our part of the code.

The function should do the following.

1. Simulate the average of two independent samples of size `nsamples` from an exponential distribution with the given rate parameter. That is, it should simulate sampling from a random variable $Y = \frac{X_1 + X_2}{2}$, where each X_1, X_2 is an exponential random variable.

2.Draw a *density* histogram of the (simulated) average.

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

`studio3_problem_2b(rate, nsamples, n_to_average, bin_width)`

The arguments to this function are:
```
rate = rate parameter for an exponential distribution
nsamples = the size of the exponential sample
n_to_average = the number of independent random variables to average
bin_width = the bin width used in the histogram
```
Note, in this function bin_width is specified by the user.

The function should do the following.

1. Draw a *density* histogram of the (simulated) average of `n_to_average` independent exponential samples. That is, it should simulate sampling from a random variable

$$Y = \frac{X_1 + X_2 + ... + X_n}{n},$$

where each X_j is an exponential random variables with the given rate parameter

2. The mean and variance of the average are given in our part of the code. Use R to add the graph of a normal density with this mean and variance to your plot. The graph should extend from 4 standard deviations below the mean to 4 above.

Be careful, in R the arguments to `dnorm` are mean and *standard deviation.*

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_studio3-test-answers.html`. If you call the same function with the same parameters, you should get the same results as in `mit18_05_s22_studio3-test-answers.html` – if there is randomness involved the answers should be close but not identical.

For your convenience, the file `mit18_05_s22_studio3-test.r` contains all the function calls used to make `mit18_05_s22_studio3-test-answers.html`.

Before uploading your code

1. Make sure all your code is in `mit18_05_s22_studio3.r`. 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_studio3-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_studio3-test-answers.html`.

Upload your code

Use the upload link on our MITx site to upload your code for grading.

Leave the file name as `mit18_05_s22_studio3.r`. (The upload script will automatically add your name and a timestamp to the file.)

You can upload more than once. We will grade the last file you upload.

Due date

Due date: The goal is to upload your work by the end of class.

If you need extra time, you can upload your work any time before 10 PM ET the day after the studio.

Solutions uploaded: Solution code will be posted on MITx at 10 PM the day after the studio.