STAT420 Homework 2

Alok K. Shukla

09/05/2016

Contents

Assignment Solutions

Exercise 1 (Writing Simple Functions)

Available vectors

```
a = 1:10
b = 10:1
c = rep(1, times = 10)
d = 2^(1:10)
```

(a) Write a function called `sum_of_squares`.

- Arguments:
  - A vector of numeric data `x`.
- Output:
  - The sum of the squares of the elements of the vector. $\sum_{i=1}^{n} x_i^2$

Provide the function, as well as the result of running the following code:

```
sum_of_squares(x = a)
sum_of_squares(x = c(c, d))
```

Solution

```
sum_of_squares = function(x) {
    ssq = as.numeric(crossprod(x))
}
```

```
print(sum_of_squares(x = a))
```

```
## [1] 385
```

```
print(sum_of_squares(x = c(c, d)))
```

```
## [1] 1398110
```

(b) Write a function called `sum_of_power`.

- Arguments:
  - A vector of numeric data `x`.
  - `p` which should have the default value of `2`.
- Output:
  - $\sum_{i=1}^{n} x_i^p$

Provide the function, as well as the result of running the following code:

```
sum_of_power(x = a)
sum_of_power(x = a, p = 3)
sum_of_power(x = a, p = a)
sum_of_power(x = a, p = c(1, 2))
```

Solution

```
sum_of_power = function(x, p = 2) {
    sp = sum(x^p)
}
```

```
print(sum_of_power(x = a))
```

```
## [1] 385
```

```
print(sum_of_power(x = a, p = 3))
```

```
## [1] 3025
```

```
print(sum_of_power(x = a, p = a))
```

```
## [1] 10405071317
```

```
print(sum_of_power(x = a, p = c(1, 2)))
```

```
## [1] 245
```

(c) Write a function called `rms_diff`.

- Arguments:
  - A vector of numeric data `x`.
  - A vector of numeric data `y`.
- Output:
  - $\sum_{i=1}^{n}(x_i - y_i)^2$

Provide the function, as well as the result of running the following code:

```
rms_diff(x = a, y = b)
rms_diff(x = d, y = c)
rms_diff(x = d, y = 1)
rms_diff(x = a, y = 0) ^ 2 * length(a)
```

Solution

```
rms_diff = function(x, y) {
    rms = sum((x - y)^2)
}
```

```
print(rms_diff(x = a, y = b))
```

```
## [1] 330
```

```
print(rms_diff(x = d, y = c))
```

```
## [1] 1394018
```

```
print(rms_diff(x = d, y = 1))
```

```
## [1] 1394018
```

```
print(rms_diff(x = a, y = 0)^2 * length(a))
```

```
## [1] 1482250
```

Exercise 2 (Plotting, Testing)

For this exercise we will use the data that is stored in `intelligence.csv` which records IQs of a random sample of residents of Pawnee and Eagleton, Indiana.

(a) Load the data from `intelligence.csv` into a variable in `R` called `intelligence`. Show the code used to do this.

Solution

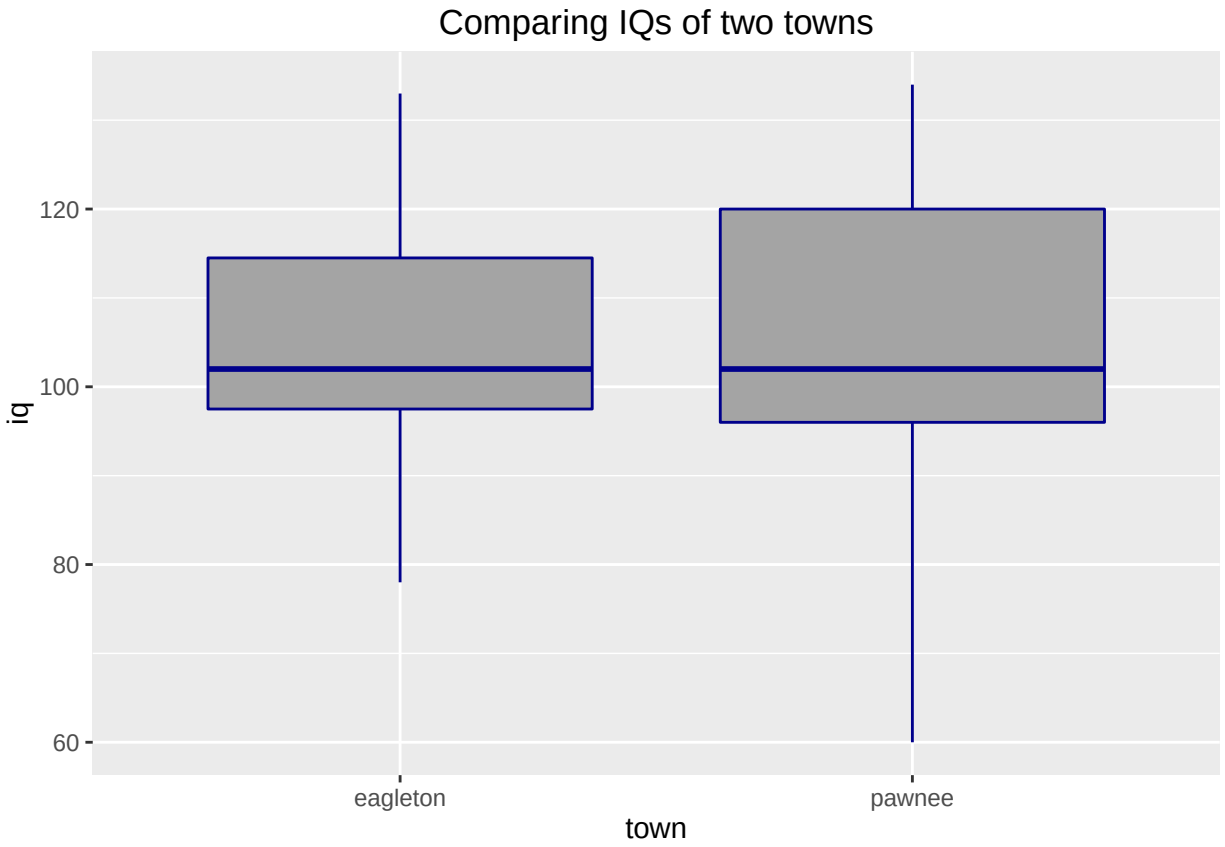
```
intelligence <- read.csv("intelligence.csv")
```

(b) Create a side-by-side boxplot that compares the IQs across the two towns. Be sure to give the plot a title and label the axes appropriately.

Solution

```
library(ggplot2)

ggplot(intelligence, aes(x = town, y = iq)) + geom_boxplot(fill = "#A4A4A4",
    color = "darkblue") + ggtitle("Comparing IQs of two towns")
```



(c) Are people from Eagleton smarter than people from Pawnee? Perform an appropriate statistical test using the given sample data. That is, test $H_0 : \mu_E = \mu_P$ vs $H_1 : \mu_E > \mu_P$, where

- μ_E is the mean IQ of a resident of Eagleton.
- μ_P is the mean IQ of a resident of Pawnee.

Explicitly state the p-value of the test and the resulting statistical decision at a significance level $\alpha = 0.10$. Interpret the results in the context of the problem.

Solution The null hypothesis

$H_0 : \mu_E = \mu_P$

Alternate hypothesis

$H_1 : \mu_E > \mu_P$

```
pawnee = intelligence[intelligence$town == "pawnee", 1]
eagleton = intelligence[intelligence$town == "eagleton", 1]
res = t.test(pawnee, eagleton, alternative = c("greater"), var.equal = TRUE)
res$p.value
```

```
## [1] 0.652316
```

We now have the p-value of our test, which is greater than our signi cance level (0.10), so we fail to reject the null hypothesis.

(d) Repeat **(c)** using a two-sided alternative hypothesis. What changes?

Solution

```
res2 = t.test(pawnee, eagleton, alternative = c("two.sided"),
    var.equal = TRUE)
res2$p.value
```

```
## [1] 0.6953681
```

The p-value increases, we still fail to reject null hypothesis.

Exercise 3 (Writing More Functions)

In this exercise we will write our own functions related to performing a one-sample t test. That is $H_0 : \mu = \mu_0$ versus $H_1 : \mu \neq \mu_0$, where μ_0 is the hypothesized value of μ.

Throughout this exercise you may **not** use the `t.test()` function inside your functions. You may use it to check your work separately, but no such double-checks should appear in your final report.

Some built in `R` functions that may be useful to you when writing your functions include: `c()`, `ifelse()`, `mean()`, `sd()`, `abs()`, `length()`, `sqrt()`, and `pt()`.

(a) Write a function called `do_t_test` which takes two inputs:

- `x`: A vector which stores observations.
- `mu`: The hypothesized value of μ which defaults to `0`.

The function should output:

- The value of the test statistic, t.
- The p-value of the test. The function only needs to be able to handle a two-sided alternative.

In order to output both, consider using `c(t, pval)` as the last line of your function, and store those two values elsewhere in the body of your function.

Solution

```
do_t_test = function(x, mu = 0) {
    x_bar = mean(x)
    s = sd(x)
    mu_0 = mu
    n = length(x)
    t = (x_bar - mu_0)/(s/sqrt(n))
    pval = 2 * pt(t, df = n - 1)
    c(t, pval)
}
```

(b) Write a function called `make_decision` which takes two inputs:

- `pval`: The p-value of a test.

- `alpha`: The significance level of a test. Set a default value of `0.05`.

The function should output `"Reject!"` or `"Fail to Reject."` based on the comparison of `pval` to `alpha`.

Solution

```
make_decision = function(pval, alpha) {
    ifelse(pval >= alpha, "Fail to Reject", "Reject")
}
```

(c) Now we will test the quality of your functions from parts **(a)** and **(b)**. Run the following code:

```
set.seed(42)
y = rnorm(25, 1.4, 1)
pval = do_t_test(y, mu = 2)[2]
pval
make_decision(pval, alpha = 0.10)
```

If your `do_t_test()` and `make_decision()` functions are correct, you should obtain a decision of `"Fail to Reject."` You will also be evaluated on whether the numeric p-value you obtain is correct.

Solution

```
set.seed(42)
y = rnorm(25, 1.4, 1)
pval = do_t_test(y, mu = 2)[2]
pval
```

```
## [1] 0.1275027
```

```
make_decision(pval, alpha = 0.1)
```

```
## [1] "Fail to Reject"
```

Exercise 4 (CLT Simulation)

For this exercise we will simulate from the exponential distribution. If a random variable X has an exponential distribution with rate parameter λ, the pdf of X can be written

$$f(x; \lambda) = \lambda e^{-\lambda x}$$

for $x \geq 0$.

Also recall,

$$\mu = E[X] = \frac{1}{\lambda}$$
$$\sigma^2 = Var[X] = \frac{1}{\lambda^2}$$

(a) This exercise relies heavily on generating random observations. To make this reproducible we will set a seed for the randomization. Alter the following code to make `birthday` store your birthday in the format: `yyyymmdd`. For example, William Gosset, better known as *Student*, was born on June 13, 1876, so he would use:

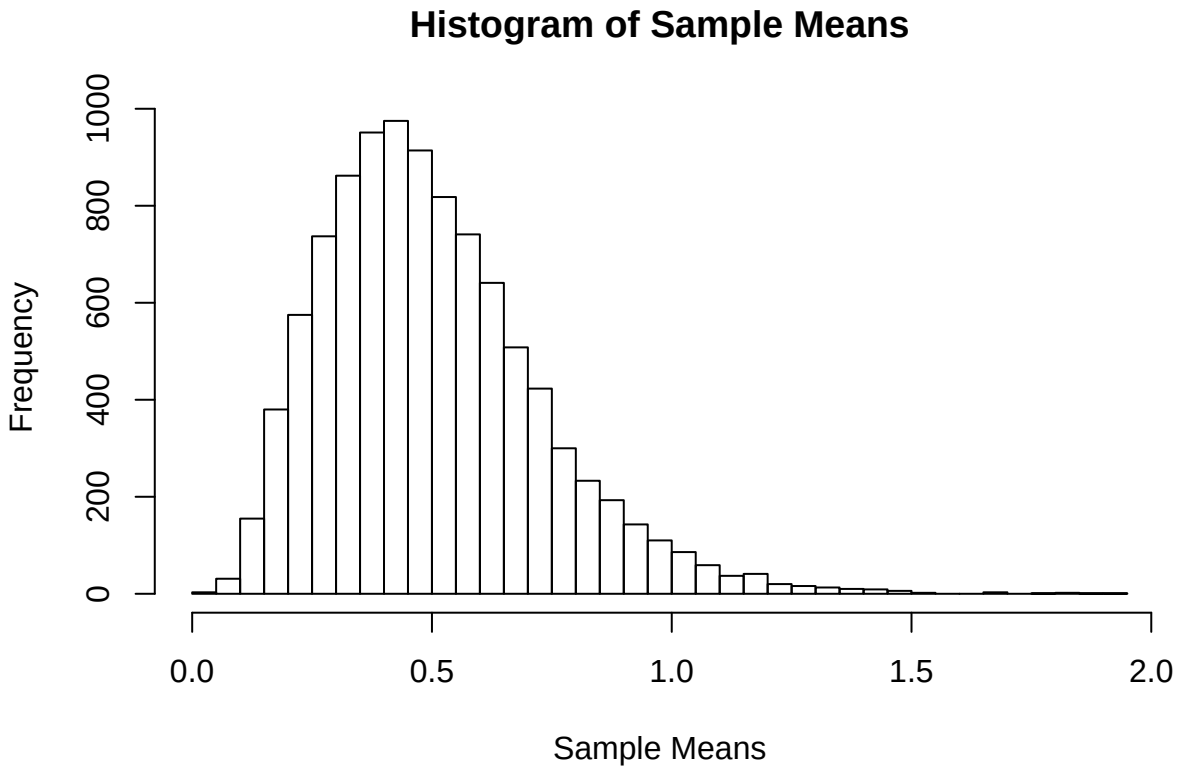
```
birthday = 18760613
set.seed(birthday)
```

(b) Simulate 10000 samples of size **5** from an exponential distribution with $\lambda = 2$. Store the mean of each sample in a vector. Plot a histogram of these sample means. (Be sure to give it a title, and label the axes appropriately.) Based on the histogram, do you think the central limit theorem applies here?

Solution

```
x = data.frame(rep(0, 10000))
x_bars = apply(x, 1, function(x) mean(rexp(5, 2)))


x_bar_hist = hist(x_bars, breaks = 50, main = "Histogram of Sample Means",
    xlab = "Sample Means")
```



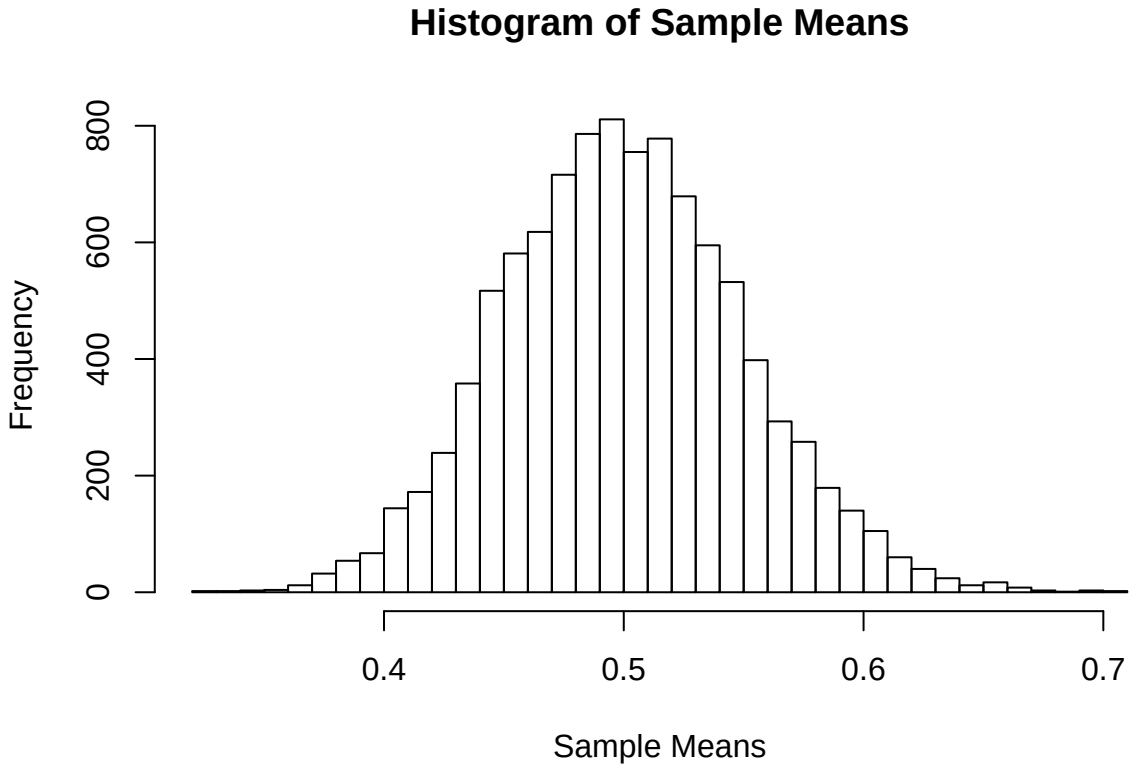
We can see from the histogram that the distribution of sample means is not perfectly normal; its right tail is longer. Thus, the Central Limit Theorem doesn't hold.

(c) Simulate 10000 samples of size **100** from an exponential distribution with $\lambda = 2$. Store the mean of each sample in a vector. Plot a histogram of these sample means. (Be sure to give it a title, and label the axes appropriately.) Based on the histogram, do you think the central limit theorem applies here?

Solution

```
x = data.frame(rep(0, 10000))
x_bars = apply(x, 1, function(x) mean(rexp(100, 2)))


x_bar_hist = hist(x_bars, breaks = 50, main = "Histogram of Sample Means",
    xlab = "Sample Means")
```



We can see from the histogram that the distribution of sample means is perfectly normal; there is no skewness. Thus, the Central Limit Theorem holds.

(d) We just repeated ourselves, so that means we probably should be writing a function. Write a function called `sim_xbars_exp` which takes three inputs:

- The number of samples to simulate.
- The sample size.
- The rate parameter of an exponential distribution.

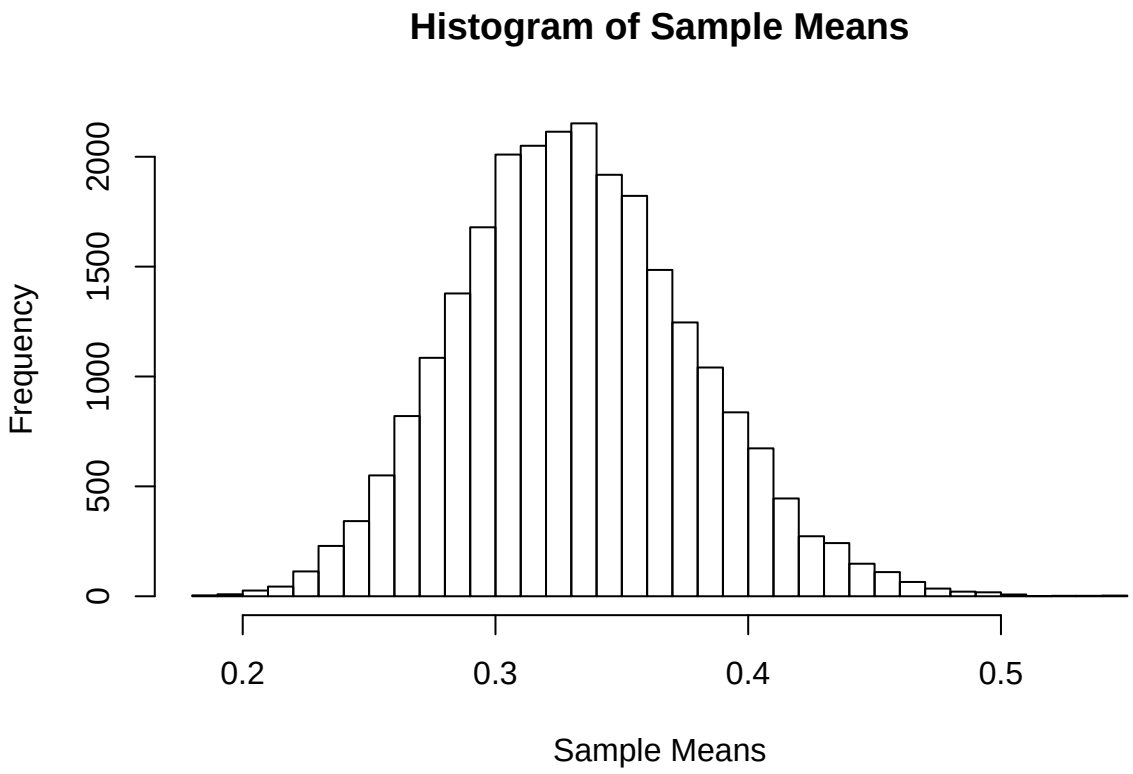
The function should output a vector of sample means which are the result of sampling from an exponential distribution as specified by the inputs.

Use your function to simulate 25000 samples of size **50** from an exponential distribution with $\lambda = 3$. Store the mean of each sample in a vector. Plot a histogram of these sample means. (Be sure to give it a title, and label the axes appropriately.)

Solution

```
sim_xbars_exp = function(no_of_iter, sample_size, rate_param) {
    x = data.frame(rep(0, no_of_iter))
    apply(x, 1, function(x) mean(rexp(sample_size, rate_param)))
}


x_bar_hist = hist(sim_xbars_exp(25000, 50, 3), breaks = 50, main = "Histogram of Sample Means",
    xlab = "Sample Means")
```



Exercise 5 (More Simulation)

Let X follow an exponential distribution with rate parameter $\lambda_X = 2$. Let Y follow a Poisson distribution with rate parameter $\lambda_Y = 3$.

We write $sd(X)$ for the true standard deviation of X and $m(Y)$ for the true median of Y.

Let s_x be the sample standard deviation of X which is an estimate of $sd(X)$. Also let m_y be the sample median which is an estimate of $m(Y)$.

Suppose we take samples of size $n_x = 10$ from X and take samples of size $n_y = 5$. Consider the statistic

$$\frac{s_x}{m_y}.$$

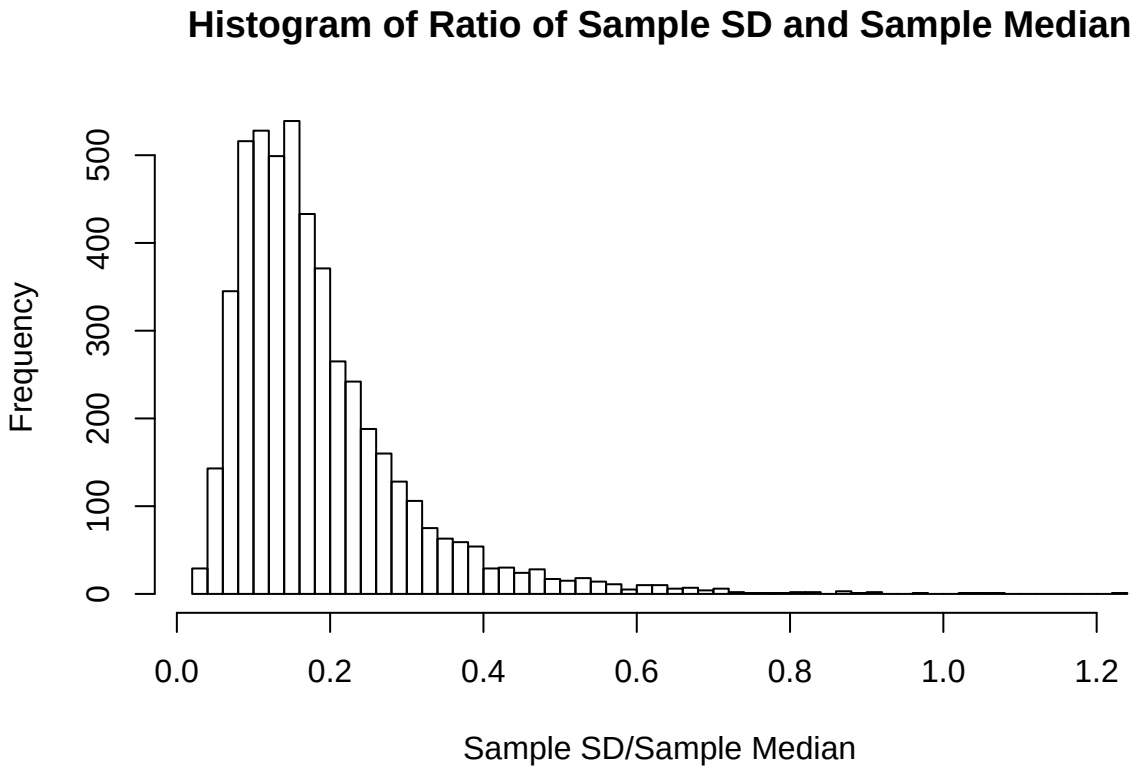
What is the (sampling) distribution of $\frac{s_x}{m_y}$? Who knows? Ask a statistician interested in theory. Instead of using mathematics, simulate $\frac{s_x}{m_y}$ 5000 times and store the results. Plot a histogram of the observed values of $\frac{s_x}{m_y}$. Comment on the shape of the histogram and empirical distribution of $\frac{s_x}{m_y}$. Before running your code, set the same seed used for the previous exercise. For full credit, do **not** use a `for` loop.

Solution

```
birthday = 18760613
set.seed(birthday)

x = data.frame(rep(0, 5000))
x_bars = apply(x, 1, function(x) sd(rexp(10, 2)))
y_bars = apply(x, 1, function(x) median(rpois(5, 3)))
z_bars = x_bars/y_bars
```

```r
z_bar_hist = hist(z_bars, breaks = 50, main = "Histogram of Ratio of Sample SD and Sample Median",
    xlab = "Sample SD/Sample Median")
```



The histogram doesnt