Day 01 - Exercise 01 - Intro Exercises

Victor Calderon

15 August, 2018

This are the responses/answers to the problems posed by the exercises in the `Intro.R` file.

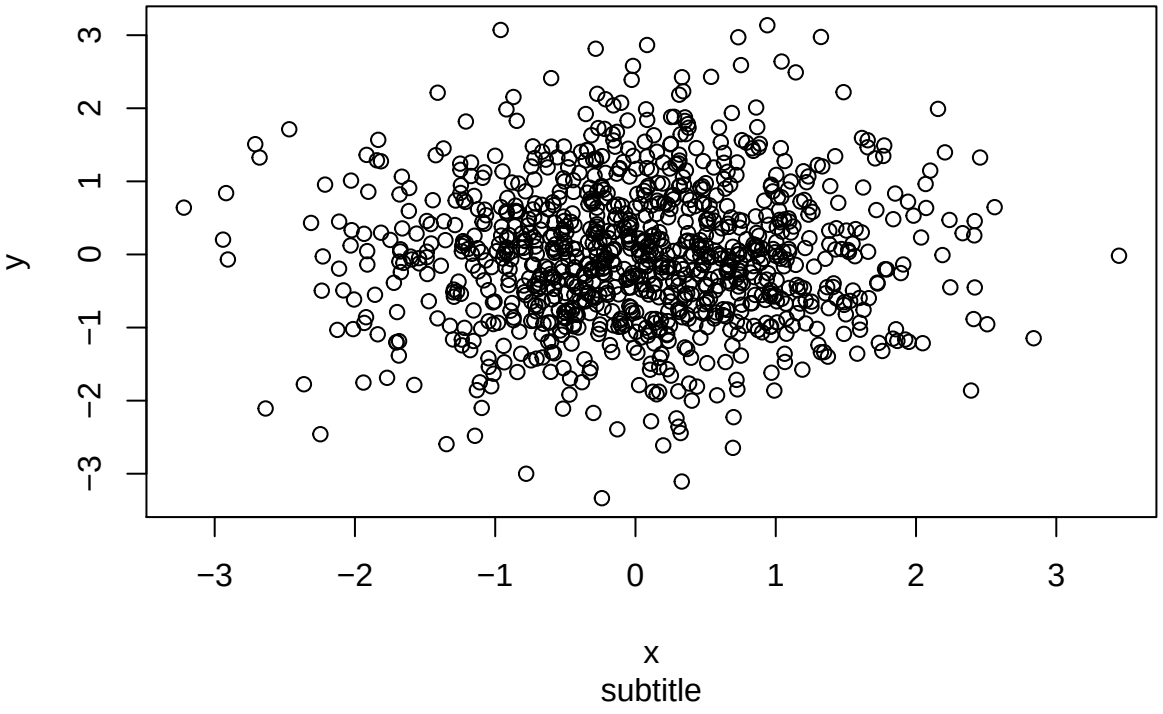
Exercise 1.1

> Look through the help file for `plot`, then recreate any figure above and set the subtitle to `subtitle`.

```
help(plot)
```

Now we can recreate the figure as in `intro.R`:

```
x = rnorm(1000)
y = rnorm(1000)
plot(x, y, sub = 'subtitle')
```



Exercise 1.2

> Find two different expressions to create a 3 x 3 (row x col) matrix with the values 2, 4, and 6 in the rows.

The first method:

```r
matrix(c(2, 4, 6), nrow = 3, ncol = 3, byrow = TRUE)
```

```
##      [,1] [,2] [,3]
## [1,]    2    4    6
## [2,]    2    4    6
## [3,]    2    4    6
```

and the 2nd version:

```r
matrix(rep(c(2,4,6), 3), nrow = 3, ncol = 3, byrow = TRUE)
```

```
##      [,1] [,2] [,3]
## [1,]    2    4    6
## [2,]    2    4    6
## [3,]    2    4    6
```

and the rest of forms:

```r
matrix(rep(seq(2, 6, 2), 3), nrow = 3, ncol = 3, byrow = TRUE)
```

```
##      [,1] [,2] [,3]
## [1,]    2    4    6
## [2,]    2    4    6
## [3,]    2    4    6
```

```r
matrix(rep(seq(2, 6, 2), c(3, 3, 3)), nrow = 3, ncol = 3)
```

```
##      [,1] [,2] [,3]
## [1,]    2    4    6
## [2,]    2    4    6
## [3,]    2    4    6
```

```r
t(matrix(rep(c(2, 4, 6), 3), 3, 3))
```

```
##      [,1] [,2] [,3]
## [1,]    2    4    6
## [2,]    2    4    6
## [3,]    2    4    6
```

```r
cbind(rep(2, 3), rep(4, 3), rep(6, 3))
```

```
##      [,1] [,2] [,3]
## [1,]    2    4    6
## [2,]    2    4    6
## [3,]    2    4    6
```

```r
rbind(c(2, 4, 6), c(2, 4, 6), c(2, 4, 6))
```

```
##      [,1] [,2] [,3]
## [1,]    2    4    6
## [2,]    2    4    6
## [3,]    2    4    6
```

These are just a few ways of determining a matrix of shape (3, 3), with the elements (2, 4, 6) in each row.

Exercise 1.3

What happens when you collate (`c()`) a list and a vector? List and a list?

Collating a list and a vector

```
# Defining my list and vector
list1 = list(c(1, 2, 3))
vec1  = c(5, 6)
c(list1, vec1)
```

```
## [[1]]
## [1] 1 2 3
##
## [[2]]
## [1] 5
##
## [[3]]
## [1] 6
```

Now collating a list with a list

```
list2 <- list(c( 1, 2, 3))
list3 <- list(c( 4, 5, 6))
c(list2, list3)
```

```
## [[1]]
## [1] 1 2 3
##
## [[2]]
## [1] 4 5 6
```

or another example:

```
c(list(1,2), c(1,2))
```

```
## [[1]]
## [1] 1
##
## [[2]]
## [1] 2
##
## [[3]]
## [1] 1
##
## [[4]]
## [1] 2
```

```
c(list(1,2), list(1,2))
```

```
## [[1]]
## [1] 1
##
## [[2]]
## [1] 2
##
## [[3]]
## [1] 1
##
```

```
## [[4]]
## [1] 2
```

Exercise 1.4

Install and load the `manipulate` package.

Installing the package `manipulate`

with `install.packages("manipulate")`

and loading the package with

`library("manipulate")`

Exercise 1.5

Use the `manipulate` function to interactively vary the `phi` argument to `persp` in the above example. Use `persp(x,y,fa,theta=30,phi=phi_slider)` as the first argument (see the `manipulate` help file and examples).

We will vary `phi`:

```
x = seq(-pi, pi, length = 50)
y = x

# Creating function
f = outer(x, y, function(x, y)cos(y)/(1 + x^2))
fa = (f - t(f)) / 2

# Manipulating figure

#manipulate(persp(x,y,fa,theta=30,phi=phi_slider),
#           phi_slider=slider(0, 90))
```

Exercise 1.6

Generate 10 values from the normal distribution with mean 5 and sd 3 and compute their sample mean.

Generating 10 values from a **normal** distribution:

```
mu = 5
sd = 4
random_values = rnorm(10, mean = mu, sd = sd)
head(random_values)
```

```
## [1]  7.7665463  5.3151400 -0.1929489 11.2728328 -1.9785986  4.1538108
```

```r
# Mean value
mean_random = mean(random_values)
head(mean_random)
```

```
## [1] 4.481353
```

Exercise 1.7

Use the `replicate` function to repeat item 6. 1000 times.

We will now use the `replicate` function:

```r
head(replicate(1000, mean(rnorm(10, mean = mu, sd = sd))))
```

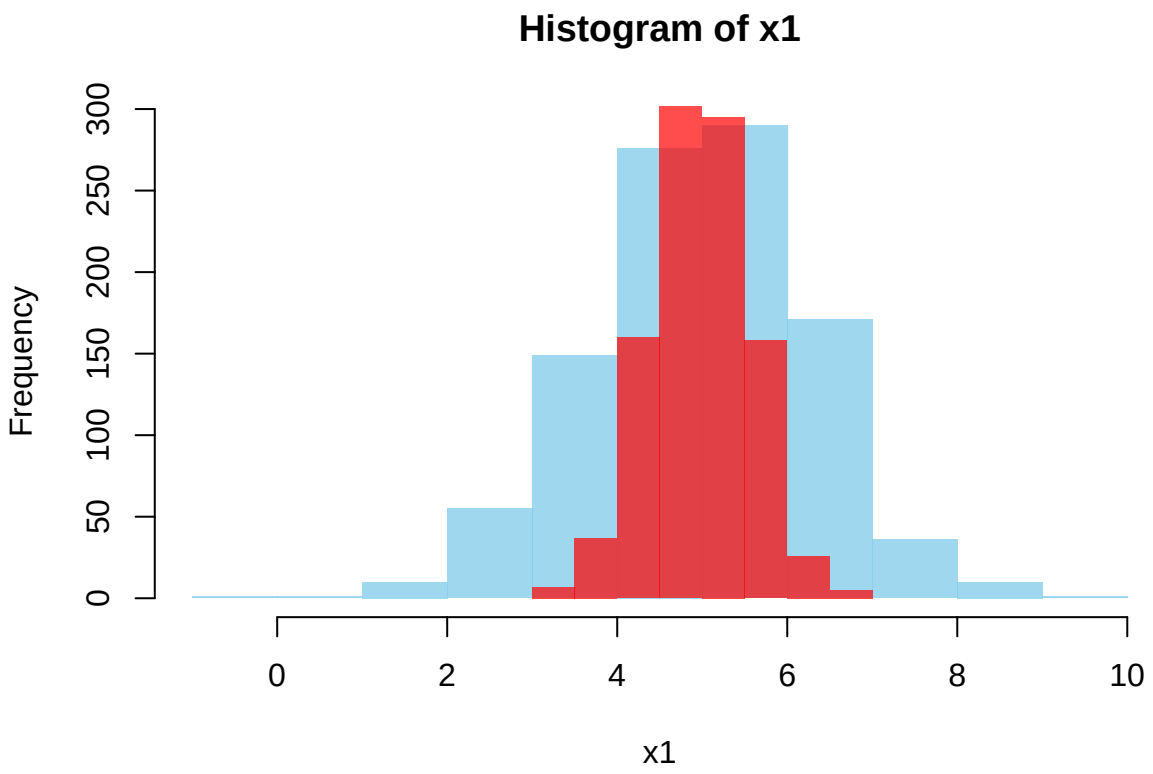
```
## [1] 4.711441 3.933092 7.202014 4.637932 5.461638 5.908382
```

Exercise 1.8

Use the `hist` function to plot a histogram of the sample means from item 7. Repeat where `N = 50` instead of `10`. Use the `add = TRUE` and `col = "red"` arguments to the 'hist' function to add the second histogram to the first for for comparison.

```r
library(scales)
## 1st histogram
x1 = replicate(1000, mean(rnorm(10, mean = mu, sd = sd)))
hist(x1, col=scales::alpha('skyblue',.8), border = F)

## 2nd histogram
x2 = replicate(1000, mean(rnorm(50, mean = mu, sd = sd)))
hist(x2, add = TRUE, col=scales::alpha('red',.7), border = F)
```


Histogram of x1
Frequency
0
50
100
150
200
250
300
0
2
4
6
8
10
x1