Statistics with Recitation: TA Session

Danny Po-Hsien Kang (康柏賢)

September 30, 2025

Today's agenda

1. Random Number Generator
   - `rnorm()`
   - `rbinom()`
2. Control Flow
   - `for-loop`
   - `if-else`
   - `replicate()`
   - `function()`
3. Assign Column Names
   - `colnames()`

Generate Normal Random Samples: `rnorm()`

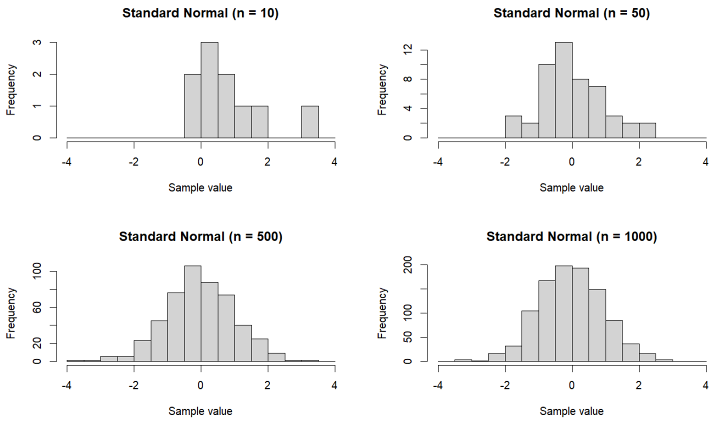

Figure: Random samples drawn from $\mathcal{N}(0,1)$ in different sizes

Generate Normal Random Samples: rnorm()

- **Syntax:**

```
num <- rnorm(n = ..., mean = ..., sd = ...)
```

- **Example:**

```
rdnum <- rnorm(10) # mean = 0, sd = 1
rdnum_2 <- rnorm(n = 10, mean = 60, sd = 15)
```

- **Output**

```
print(rdnum)
[1]  1.2040950  0.1933983 -0.4341151  1.9722344  0.3030853
[6]  1.6329367  0.3025881  1.1791016 -1.6924863 -0.6795203

print(rdnum_2)
[1]  53.90865 44.79792 68.35403 85.12336 52.85266
[6]  45.43769 49.58361 44.52621 83.63402 72.28533
```

Generate Binomial Random Samples: `rbinom()`

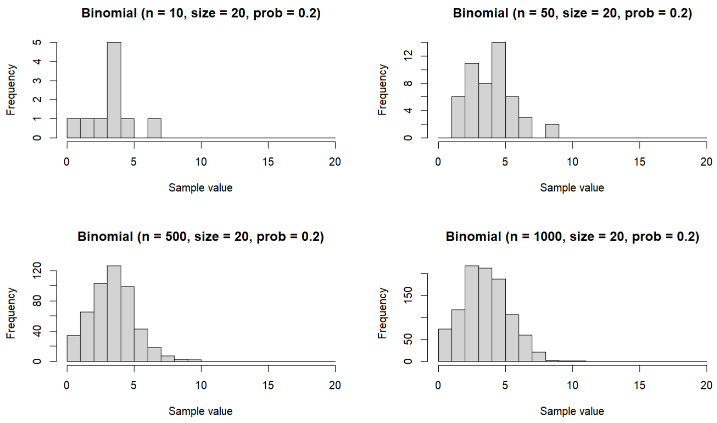

Figure: Random samples drawn from Bin$(20, 0.2)$ in different sizes

Generate Binomial Random Samples: rnorm()

- **Syntax:**

```
a <- rbinom(n = ..., size = ..., prob = ...)
```

- **Example:**

```
rdnum <- rbinom(n = 30, size = 20, prob = .5)
```

- **Output**

```
print(rdnum)
[1]   9  8 12 13 17 14 10  9 12 11
[11] 11 14  7  9 11 10 12  8 11 12
[21]  9 11 12 11 14  9 14  9 11 14

print(mean(rdnum))
[1] 11.13333
```

Repeated Work: for-loop

- **Syntax:**

```
for (var in seq){
  # do something
  }
```

- **Example:**

```
for (i in 1:5){
  print(i)
  }
```

- **Output:**

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

Different Works in Different Conditions: if-else

- **Syntax:**

```
if(a < b){
  # do work 1
  }else if(a == b){
    # do work 2
    }else{
      # do work 3
      }
```

Different Works in Different Conditions: if-else

- **Example:**

```
for (i in 55:65){
  if(i < 60){
    print("Fail.")
    }else if(i == 60){
      print("On the threshold.")
      }else{
        print("Pass the course!")
      }
  }
```

Different Works in Different Conditions: if-else

- **Output:**

```
[1] "Fail."
[1] "Fail."
[1] "Fail."
[1] "Fail."
[1] "Fail."
[1] "On the threshold."
[1] "Pass the course!"
[1] "Pass the course!"
[1] "Pass the course!"
[1] "Pass the course!"
[1] "Pass the course!"
```

Repeat Doing Something: replicate()

- **Syntax:**

```
replicate(n = ..., expr = ...)
```

- **Example:**

```
rep_value <- replicate(n = 10, "Hello!")
rep_sample <- replicate(n = 4, rnorm(2, mean = 0, sd = 1))
```

- **Output:**

```
print(rep_value)
[1] "Hello!" "Hello!" "Hello!" "Hello!" "Hello!"
[6] "Hello!" "Hello!" "Hello!" "Hello!" "Hello!"

print(rep_sample)
           [,1]      [,2]       [,3]       [,4]
[1,] 0.30578023 -1.695602  0.7769269 -0.6512941
[2,] 0.06849653  1.596402 -0.6807501  1.2318443
```

Self-Defined Function: function()

- **Syntax:**

```
function_name <- function(param1, param2, ...){
  # do something
  return(output)
  }
```

- **Example:**

```
add <- function(a, b){
 return(a+b)
 }
```

- **Output:**

```
print(add(50, 100)) # or use add(a = 50, b = 100)
[1] 150
```

Reasons to use `function()`

- Avoid copying and pasting the same code when repeated use.
  - Makes code cleaner.
- When you hope the parameter can be customized.
  - Sometimes the parameters in a project are undetermined.
  - Without a function, changing parameters can be troublesome and easy to make mistakes.
- Functions can be saved in a script.
  - Can be called by `source("myfunctions.R")`.
  - This allowed the functions to be portable.
  - Makes the code shorter and easier to understand.

Assign Column Names: colnames()

- **Syntax:**

```
colnames(df) <- new_names
```

- **Example:**

```
df <- data.frame(a = 1:3, b = 4:6)
colnames(df) <- c("height", "weight")
```

- **Output:**

```
  height weight
1      1      4
2      2      5
3      3      6
```

Assign Column Names: colnames()

- colnames() works on matrices too.
- **Example:**

```
m <- matrix(1:6, nrow = 3)
colnames(m) <- c("col1", "col2")
```

- **Output:**

```
     col1 col2
[1,]    1    4
[2,]    2    5
[3,]    3    6
```

About Today's Practices

- The last practice "Certification Exam" is a challenging question.
    - You have to make use of the tools taught in this TA session to solve it.
- Recommended to try it yourself before asking Chat-GPT
    - See how far you can go without any help.