R help session 02/09/2018 - object, function

- Han Jiang, Sai Aravindh Ravi

Topics for today

- Object
- Function
- Simulating rolling die
- Define your own function (optional)

R object

- A name to store the data
  - a <- 1 (a is the box, 1 is the mail)
  - Try

```
a = 1
a = a + 1
a?
```

```
> a = 1
> a = a + 1
> a
[1] 2
```

- Rules for naming
  - Cannot start with number
  - Cannot use some special symbols
    - +, - , $, ^, !, @
  - Case-sensitive
    - 'Name' and 'name' are different

```
Name <- 1
name <- 0
```

- List all object names
  - ls()

Good names	Names that cause errors
a	1trial
b	$
FOO	^mean
my_var	2nd
.day	!bad

R objects

- Example:
  - define a die object
    - Create a vector from 1 to 6
    - Assign the vector to a object named ‘die’

```
> die = 1:6
> die
[1] 1 2 3 4 5 6
```

R objects

- Example:
  - Operations
    - Multiplication: * (element-wise)

```
> die * die
[1]  1  4  9 16 25 36
```

```
> c(1*1, 2*2, 3*3, 4*4,5*5,6*6)
[1]  1  4  9 16 25 36
```

    - Try: Inner multiplication: %*%
    - Try: Outer multiplication %o% (solution on next page)

R objects (solution)

- Example:
  - Operations
    - Inner multiplication: %*%
    - Outer multiplication %o%

%*%

```
> die %*% die
     [,1]
[1,]   91
> 6*6+5*5+4*4+3*3+2*2+1*1
[1] 91
```

%o%

```
> die_1 = 1:5
> die %o% die_1
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]    2    4    6    8   10
[3,]    3    6    9   12   15
[4,]    4    8   12   16   20
[5,]    5   10   15   20   25
[6,]    6   12   18   24   30
> # col vector die with shape (6*1)
```

Functions

- R has many built-in functions
  - round
  - mean
    - Try mean of die
  - Link functions together
    - From inner to outer

```
round(3.1415)
## 3
```

```
> # mean
> mean(die)
[1] 3.5
```

```
> # two functions
> round(mean(die))
[1] 4
```

```
round(mean(die))
↓
round(mean(1:6))
↓
round(3.5)
↓
4
```

Functions

- Roll the die using sample()
  - Argument 1: a vector named x
  - Argument 2: a number named size

```
> sample(x = die, size = 3)
[1] 2 6 1
> sample(x = die, size = 3)
[1] 3 6 2
```

- Argument order can be switched

```
> sample(size = 3, x = die)
[1] 5 3 6
```

- A concise , less clear way
  - Align the order of the arguments

```
> sample(die,3)
[1] 1 2 3
```

  - try

```
> sample(3,die)
```

Rolling with replacement

- The rolling is without replacement
  - try 1
  - try 2

```
> sample(x=die, size=6)
[1] 6 5 1 2 3 4
```

```
> sample(x=die, size=7)
```

- Check the help of sample()
  - required argument vs. optional argument

sample(x, size, replace = FALSE, prob = NULL)

- Rolling with replacement
  - Define function: rolling die 20 times with replacement
  - solution

```
> sample(die, size = 20, replace = TRUE)
```

Rolling with replacement

- Calculate the expectation of rolling a die
  - Analytical way

```
> expecation = 1*1/6 + 2*1/6 + 3*1/6 + 4*1/6 + 5*1/6 + 6*1/6
> expecation
[1] 3.5
```

  - Simulation way
    - Rolling die multiple times and calculate the mean
    - Rolling 10 times

```
> result = sample(die, size=10, replace=TRUE)
> mean(result)
```

    - Try 100, 10000 times

Rolling die function

- Define a biased die
  - P(x=1) = 0.9, P(x=2) = 0.1, rest are 0
  - Hint: use prob argument
  - Solution

```
> sample(die, size=20, replace=TRUE, prob=c(0.9,0.1,0,0,0,0))
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1
```

  - The expectation?

Your own function (optional)

- So far,

```
> die <- 1:6
> dice <- sample(die, size = 1000, replace = TRUE)
> mean(dice)
```

- Wrap the code into a roll() function

Function constructor

Constructor:

```
my_function <- function() {}
```

Define function name: roll

{}: put code

return (optional): to send computed value back

Define the roll function and return the expectation

Run the roll function by > roll()

```
# roll() function
roll <- function(){
  die <- 1:6
  dice <- sample(die, size = 1000, replace = TRUE)
  expectation = mean(dice)

  return(expectation)
}
```

Function constructor

- Argument
- Define a object in ()

```
roll_1 <- function(die){
  dice <- sample(die, size=10,replace=TRUE)
}
```

- Try

```
> roll_1()
```

  – Why error ?
- Need to give the argument a value

```
> result = roll_1(die)
> result
```

summary

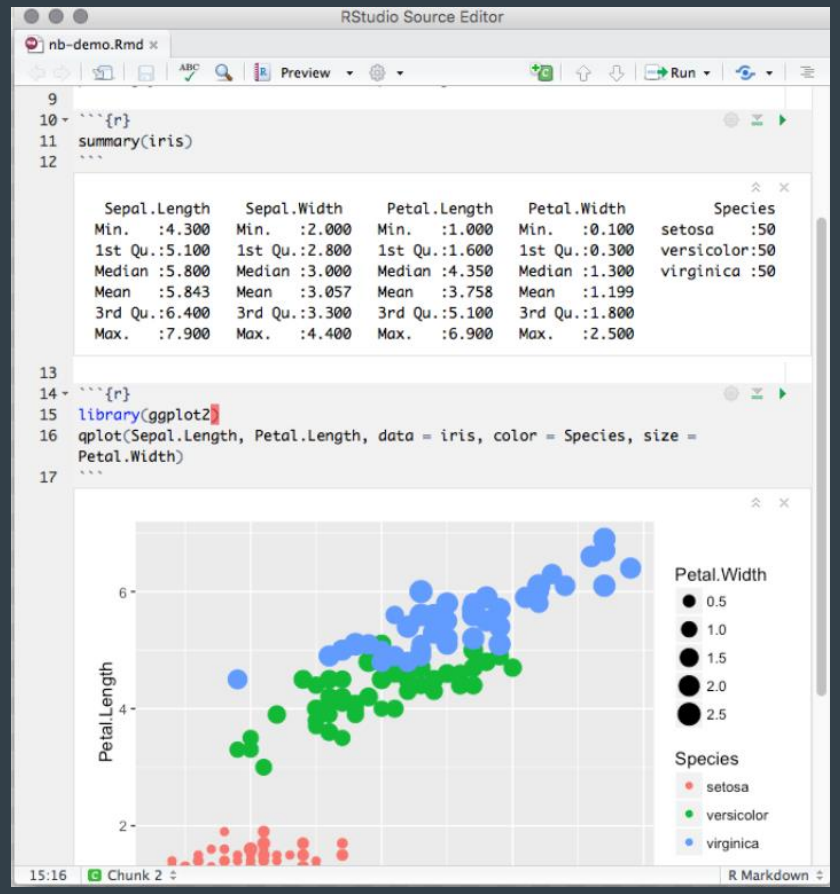
- R object, basic math operation
- Basic function
- Required argument vs. optional argument
- Design your own function

R notebook

- R notebook
  - an R markdown document with executable codes
  - output visible beneath the input
  - Interactive document with R
  - Publication quality output
- Create notebook

  > open Rstudio

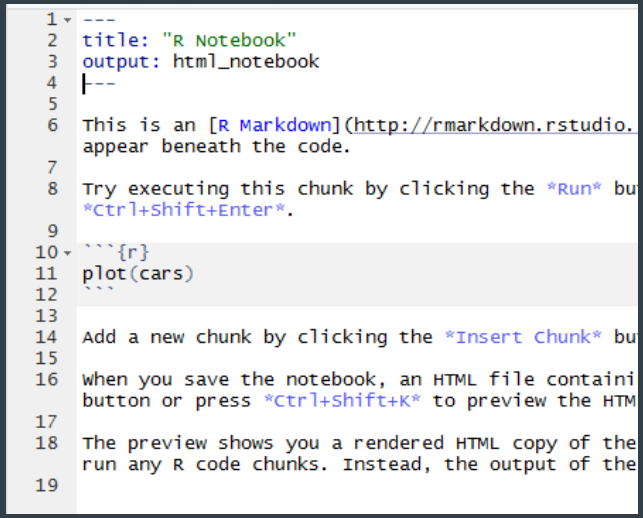
  > File > New file > R notebook



R notebook

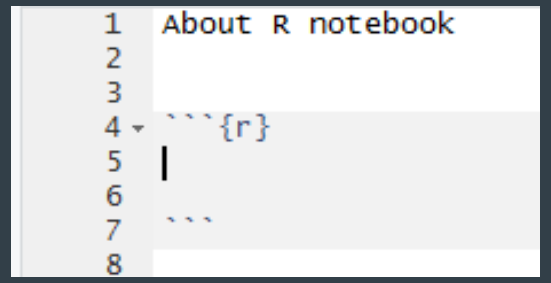
documentation

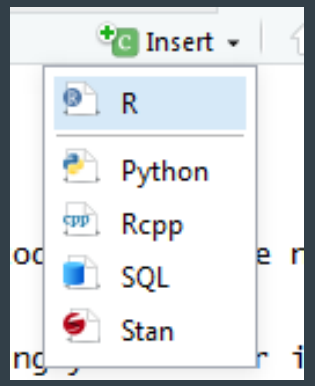
R code



- A text editor

- Add R code
  - 1. Click insert >> Insert a R chunk
  - 2. Direct insert R chunk by typing





- Run code
  - Select code chunk and Ctrl + Enter
  - Click Run and select run current chunk