

Lecture 5: More functions

Last time

```
1 assess_coverage <- function(n, nsim, beta0, beta1, noise_dist){
2   results <- rep(NA, nsim)
3
4   for(i in 1:nsim){
5     x <- runif(n, min=0, max=1)
6     noise <- noise_dist(n)
7     y <- beta0 + beta1*x + noise
8
9     lm_mod <- lm(y ~ x)
10    ci <- confint(lm_mod, "x", level = 0.95)
11    results[i] <- ci[1] < beta1 & ci[2] > beta1
12  }
13  return(mean(results))
14 }
```

Function defaults

```
1 assess_coverage <- function(n, nsim, beta0, beta1, noise_dist){  
2   ...  
3 }
```

```
1 assess_coverage(n = 100, nsim = 1000, beta0 = 0.5, beta1 = 1,  
2                 noise_dist = rexp)
```

```
[1] 0.951
```

What if I wanted `noise_dist` to be the $N(0, 1)$ distribution instead?

Function defaults

```
1 assess_coverage <- function(n, nsim, beta0, beta1, noise_dist){  
2   ...  
3 }
```

```
1 assess_coverage(n = 100, nsim = 1000, beta0 = 0.5, beta1 = 1,  
2                 noise_dist = rnorm)
```

```
[1] 0.951
```

What if I wanted `noise_dist` to be the χ_1^2 distribution?

Function defaults

```
1 assess_coverage <- function(n, nsim, beta0, beta1, noise_dist){  
2   ...  
3 }
```

```
1 assess_coverage(n = 100, nsim = 1000, beta0 = 0.5, beta1 = 1,  
2                 noise_dist = rchisq)
```

Error in noise_dist(n): argument "df" is missing, with no default

Function defaults

```
1 ?rexp
```

Description

Density, distribution function, quantile function and random generation for the exponential distribution with rate `rate` (i.e., mean $1/\text{rate}$).

Usage

```
rexp(n, rate = 1)
```

- The *default* value of `rate` is 1!

Function defaults

```
rexp(n, rate = 1)
```

The *default* value of rate is 1!

Same results:

```
1 set.seed(93)
2 rexp(n=1)
```

```
[1] 1.188317
```

```
1 set.seed(93)
2 rexp(n=1, rate=1)
```

```
[1] 1.188317
```

Different result:

```
1 set.seed(93)
2 rexp(n=1, rate=2)
```

```
[1] 0.5941585
```

Function defaults

```
1 ?rnorm
```

Description

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to `mean` and standard deviation equal to `sd`.

Usage

```
rnorm(n, mean = 0, sd = 1)
```


Function defaults

```
1 ?rchisq
```

Usage

```
rchisq(n, df, ncp = 0)
```

- There is no default for `df` in the `rchisq` function!

```
1 assess_coverage(n = 100, nsim = 1000, beta0 = 0.5, beta1 = 1,  
2                 noise_dist = rchisq)
```

Error in `noise_dist(n)`: argument "df" is missing, with no default

- How can we use a χ_1^2 for `noise_dist`?

Two options

Option 1: Create a new function

```
1 set.seed(73)
2
3 chisq_1 <- function(m){
4   return(rchisq(m, df=1))
5 }
6 assess_coverage(n = 100, nsim = 1000, beta0 = 0.5, beta1 = 1,
7               noise_dist = chisq_1)
```

```
[1] 0.962
```

Option 2: Anonymous functions

```
1 set.seed(73)
2
3 assess_coverage(n = 100, nsim = 1000, beta0 = 0.5, beta1 = 1,
4               noise_dist = function(m) {return(rchisq(m, df=1))})
```

```
[1] 0.962
```

Anonymous functions

We use anonymous functions when we don't need a name for them.

Example:

```
1 integrate(function(x) {sin(x)^2}, 0, pi)
```

```
1.570796 with absolute error < 1.7e-14
```

Function scoping

What value will the following code return?

```
1 g01 <- function(x = 10) {  
2   return(x)  
3 }  
4  
5 g01()
```

Function scoping

What value will the following code return?

```
1 g01 <- function(x = 10) {  
2   return(x)  
3 }  
4  
5 g01()
```

```
[1] 10
```

What if I try to look at x?

```
1 x
```

Function scoping

What value will the following code return?

```
1 g01 <- function(x = 10) {  
2   return(x)  
3 }  
4  
5 g01()
```

```
[1] 10
```

What if I try to look at x?

```
1 x
```

```
Error in eval(expr, envir, enclos): object 'x' not found
```

- Variables created within functions don't exist outside the function!

Function scoping

Variables created within functions don't exist outside the function!

```
1 g01 <- function() {  
2   x <- 10  
3   return(x)  
4 }  
5  
6 g01()
```

```
[1] 10
```

```
1 x
```

```
Error in eval(expr, envir, enclos): object 'x' not found
```

Function scoping

What will the following code return?

```
1 x <- 10
2
3 g01 <- function(){
4   return(x)
5 }
6
7 g01()
```


Function scoping

```
1 x <- 10
2
3 g01 <- function(){
4   return(x)
5 }
6
7 g01()
```

```
[1] 10
```

```
1 x
```

```
[1] 10
```

- If a variable is not defined in a function, R looks outside the function (the *global environment*)

Name masking

What value will the following code return?

```
1 x <- 10
2 g01 <- function() {
3   x <- 20
4   return(x)
5 }
6
7 g01()
8 x
```

Name masking

What value will the following code return?

```
1 x <- 10
2 g01 <- function() {
3   x <- 20
4   return(x)
5 }
6
7 g01()
```

```
[1] 20
```

```
1 x
```

```
[1] 10
```

- Names defined inside a function *mask* names defined outside a function
- Variables created within a function don't exist outside

Summary

- Anonymous functions can be used if we don't need to name them
- Variables created within a function don't exist outside
- If a variable is not defined in a function, R looks outside the function
- Names defined inside a function *mask* names defined outside a function

Class activity

https://sta279-f23.github.io/class_activities/ca_lecture_5.html

- If finished early, you may work on homework
- Solutions will be posted on course website

