

# Week-6: Code-along

Annette

2023-09-17

```
knitr::opts_chunk$set(echo = TRUE)
```

## II. Code to edit and execute using the Code-along-6.Rmd file

### A. for loop

#### 1. Simple for loop (Slide #6)

```
for (x in c(3, 6, 9)) {  
  print(x)  
}
```

```
## [1] 3  
## [1] 6  
## [1] 9
```

#### 2. for loops structure (Slide #7)

```
for (x in 1:8) {print(x)}
```

```
## [1] 1  
## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5  
## [1] 6  
## [1] 7  
## [1] 8
```

```
for (x in 1:8)  
{y <- seq(from=100, to=200, by=5)  
print(y[x])}
```

```
## [1] 100  
## [1] 105  
## [1] 110
```

```
## [1] 115
## [1] 120
## [1] 125
## [1] 130
## [1] 135
```

### 3. Example: find sample means (Slide #9)

```
sample_sizes <- c(5, 10, 15, 20, 25000)

sample_means <- double(length(sample_sizes))

for (i in seq_along(sample_sizes)) {
  sample_means[[i]] <- mean(rnorm(sample_sizes[[i]]))
}

sample_means

## [1] -0.109335858  0.714445765  0.231756020 -0.121306350  0.007322002
```

### 4. Alternate ways to pre-allocate space (Slide #12)

```
sample_means <- rep(0, length(sample_sizes))
sample_means

## [1] 0 0 0 0 0
```

```
data_list <- vector("list", length = 5)
data_list

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

## 5. Review: Vectorized operations (Slide #18)

```
a <- 7:11
b <- 8:12

out <- rep(0L, 5)
for (i in seq_along(a)) {
  out[i] <- a[i] + b[i]
}

out

## [1] 15 17 19 21 23
```

```
a <- 7:11
b <- 8:12
out <- a + b
out

## [1] 15 17 19 21 23
```

## B. Functionals

### 6. for loops vs Functionals (Slides #23 and #24)

```
sample_sizes <- c(5, 10, 15, 20, 25000)

sample_summary <- function(sample_sizes, fun) {

  out <- vector("double", length(sample_sizes))
  for (i in seq_along(sample_sizes)) {
    out[i] <- fun(rnorm(sample_sizes[i]))
  }
  return(out)
}
```

```
sample_summary(sample_sizes, mean)

## [1] -0.421971237 -0.588724964 -0.221535598  0.068624333 -0.001418015

sample_summary(sample_sizes, median)

## [1]  0.6713089339  0.1614167733  0.0006341392 -0.2390643662  0.0075166562

sample_summary(sample_sizes, sd)

## [1] 1.0808812 0.5643126 0.8550431 1.3157016 0.9981776
```

## C. while loop

### 7. while loop (Slides #27)

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

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

```
i <- 1  
while (i<=5) {  
  print(i)  
  i <- i + 1  
}
```

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