Week-6: Code-along

Ho Wei Ni

2023-09-17

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] -0.493523199  0.233167449  0.103907779 -0.298553831 -0.005252361
```

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

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

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

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)
fsd <- function(sample_sizes) {
  sample_sds <- rep(0, length(sample_sizes))
  for (i in seq_along(sample_sizes)) {
    sample_sds[1] <- sd(rnorm(sample_sizes[i]))
  }
}
```

```
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)
}
#Compute mean
sample_summary(sample_sizes, mean)
```

```
## [1]  0.21873298 -0.44883364 -0.17164703  0.04389004  0.00301603
```

```
# Compute median
sample_summary(sample_sizes, median)
```

```
## [1] -0.402053299 -0.116213721  0.097900732 -0.139868611 -0.001110745
```

```
# Compute sd
sample_summary(sample_sizes, sd)
```

```
## [1] 0.7066761 1.3640308 0.7774635 0.9305551 1.0038029
```

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
```

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

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