Week-6: Code-along

Wong Wei Qi

2023-09-20

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]  0.3754313457 -0.0491798179 -0.1366351137  0.2395072688 -0.0002305616
```

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

```
sample_means <- vector("double", length = 5)
sample_means <- double(5)
sample_means <- rep(0, length(sample_sizes))
```

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

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)
# Create a functional- function inside a function
sample_summary <- function(sample_sizes, fun) {
# Initialise a vector of the same size as sample_sizes
out <- vector("double", length(sample_sizes))
# Run the for loop for as long as the length of sample_sizes
for (i in seq_along(sample_sizes)) {
    # Perform operations indicated fun
    out[i] <- fun(rnorm(sample_sizes[i]))
  }
return(out) }
```

```
# Slide 24
sample_summary(sample_sizes,mean)
```

```
## [1]  0.238889702 -0.505100692  0.524354857 -0.197684411 -0.006297173
```

```
sample_summary(sample_sizes,median)
```

```
## [1]  0.158193536 -0.248131960  0.174344341  0.346583324 -0.004565616
```

```
sample_summary(sample_sizes,sd)
```

```
## [1] 1.3054513 1.0813627 0.9894230 0.9274310 0.9988177
```

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) {
# body
print(i)
i <- i + 1 }
```

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