Week-6: Code-along

NM2207: Computational Media Literacy

2023-09-18

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

A. `for` loop

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

```
# Enter code here
for (x in c(3, 6, 9)) { print(x)
}
```

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

```
# Left-hand side code: for loop for passing values
for (x in 1:8) {print(x)}
```

```
# Right-hand side code: for loop for passing indices
for (x in 1:8)
{y <- seq(from=100,to=200,by=5)
print(y[x])}
```

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

```
# Enter code here
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
```

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

```
# Example 3 for data_type=double
data_list <- vector("list", length = 5)
```

```
# Initialisation of data_list
sample_means <- double(5)
```

5. Review: Vectorized operations (Slide #18)

```
# Example: bad idea!
a <- 7:11
b <- 8:12
out <- a + b
out
```

```
# Taking advantage of vectorization
a <- 7:11
b <- 8:12
 out <- rep(0L, 5)
 for (i in seq_along(a)) { out[i] <- a[i] + b[i]}
out
```

B. Functionals

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

```
# Slide 23
sample_sizes <- c(5, 10, 15, 20, 25000)
sample_summary <- function(sample_sizes, fun) {
 sample_sizes
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)
```

```
# Slide 24
#Compute mean
sample_summary(sample_sizes,mean)
# Compute median
sample_summary(sample_sizes,median)
# Compute sd
sample_summary(sample_sizes,sd)
```

C. `while` loop

7. `while` loop (Slides #27)

```
# Left-hand side code: for loop
for(i in 1:5){ print(i)
}
```

```
# Right-hand side code: while loop
i <- 1
while (i <= 5) {
# body
print(i)
i <- i + 1 }
```