Week-6: Code-along

NM2207: Computational Media Literacy

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)

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

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

```
# 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)

```r
# determine what to loop over
sample_sizes <- c(5, 10, 15, 20, 25000)
# pre-allocate space to store output
sample_means <- double(length(sample_sizes))

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

print(sample_means)
```

```
## [1] -0.70725237  0.49690461 -0.31187077 -0.12408008  0.00337916
```

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

```r
# Example 1 for data_type=double
sample_means <- vector("double", length = 5)

# Example 2 for data_type=double
sample_means <- double(5)

# Example 3 for data_type=double
sample_means <- rep(0, length(sample_sizes))
```

```r
# Initialisation of data_list
data_list <- vector("list", length = 5)
```

5. Review: Vectorized operations (Slide #18)

```r
# Example: bad idea!
# Vector with # from 7 to 11
a <- 7:11
# Vector with # from 8 to 12
b <- 8:12
# Vector of all zeros of length 5
out <- rep(0L, 5)
# Loop along the length of vector a
for (i in seq_along(a)) {
  # Each entry of out is the sum of the corresponding elements
  out[i] <- a[i] + b[i]
}

out
```

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

```r
# Taking advantage of vectorization
# Vector from # from 7 to 11
a <- 7:11
# Vector from # from 8 to 12
b <- 8:12
out <- a + b
out
```

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

B. Functionals

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

```r
# Initialise a vector with the size of 5 different samples
sample_sizes <- c(5, 10, 15, 20, 25000)

# Define a function that wraps the for loop
fmean <- function(sample_sizes){

  # Initialise an empty vector of the same length as semple_sizes
  sample_means <- rep(0, length(sample_sizes))

  # Compute the mean of each sample
  for (i in seq_along(samples_sizes)) {

    samples_means[i] <- mean(rnorm(sample_sizes[i]))
  }
}
```

```r
# Slide 24
#Compute mean
# Initialise a vector with the size of 5 different samples
sample_sizes <- c(5, 10, 15, 20, 25000)

# Define a function that wraps the for loop
fmean <- function(sample_sizes){

  # Initialise an empty vector of the same length as semple_sizes
  sample_means <- rep(0, length(sample_sizes))

  # Compute the mean of each sample
  for (i in seq_along(samples_sizes)) {

    samples_means[i] <- mean(rnorm(sample_sizes[i]))
  }
}

# Compute median
# Initialise a vector with the size of 5 different samples
sample_sizes <- c(5, 10, 15, 20, 25000)
```

```r
# Define a function that wraps the for loop
fmean <- function(sample_sizes){

  # Initialise an empty vector of the same length as semple_sizes
  sample_medians <- rep(0, length(sample_sizes))

  # Compute the mean of each sample
  for (i in seq_along(samples_sizes)) {

    samples_medians <- median(rnorm(sample_sizes[i]))
  }
}

# Compute sd
# Initialise a vector with the size of 5 different samples
sample_sizes <- c(5, 10, 15, 20, 25000)

# Define a function that wraps the for loop
fmean <- function(sample_sizes){

  # Initialise an empty vector of the same length as semple_sizes
  sample_sds <- rep(0, length(sample_sizes))

  # Compute the mean of each sample
  for (i in seq_along(samples_sizes)) {

    samples_sds[i] <- sd(rnorm(sample_sizes[i]))
  }
}
```

C. `while` loop

7. `while` loop (Slides #27)

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

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

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

```
i <- i + 1
}
```

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