Week 3 Tutorial: Control Flow in R

POP77001 Computer Programming for Social Scientists

Note on Code Formatting

- Use consistent style and indentation (RStudio indents by 2 whitespaces, Jupyter Notebook by 4)
- Even though it does not affect how programs are executed in R

```r
# Good style
is_positive <- function(num) {
  if (num > 0) {
    res <- TRUE
  } else {
    res <- FALSE
  }
  return(res)
}
```

```r
# Bad style
is_positive <- function(num) {
if (num > 0) {
res <- TRUE
}
else {
res <- FALSE
}
return(res)
}
```

Exercise 1: Conditional Statements

- Below you will find a code snippet for finding the maximum value in vector `v` using exhaustive enumeration.
- Modify it in such a way that it finds the minimum (rather than maximum) value.
- Check that your code works correctly by applying the built-in function `min()`.

```r
set.seed(2025)
v <- sample(1:1000, 50)
max_val <- v[1]
for (i in v[2:length(v)]) {
  if (i > max_val) {
    max_val <- i
  }
}
max_val
```

```
[1] 983
```

- Now let's make this code more robust.
- Re-write the code above so that it can handle vectors that contain NAs in them.
- Test your code on the vector below.

```
set.seed(2025)
v <- sample(c(1:750, rep(NA, 250)), 50)
v
```

```
 [1]  NA 460  NA  NA  NA 279 510 187 266 461  NA  NA  NA 539  NA  NA 373 407 571
[20]  NA 159 142 549  NA 575  NA  NA  NA  NA 630  88 195 686  NA 371 306  59 317
[39] 528  NA 564 186 144  NA  49 399  50 626 185 107
```

Exercise 2: Iteration

- Below you see a matrix of random 30 observations of 5 variables
- Inspect visually the matrix
- Which variable(s) do you think has(ve) the highest standard deviation?
- First, try subsetting individual rows and columns from this matrix
- Check the dimensionality of the matrix using `dim()`, `nrow()` and `ncol()` functions
- Write a loop that goes over each variable and calculates its standard deviation
- You can use `sd()` function to calculate the standard deviation
- Save these calculated standard deviations in a vector
- Find the variable with the maximum standard deviation using `max()` or `which.max()` functions
- Is it the one you thought it would be?

```r
# When dealing with random number generation it's always a good idea to mak
# by setting the seed with set.seed(function)
set.seed(2025)
# Here we create a matrix of 30 observations of 5 variables
# where each variable is a random draw from a normal distribution with mean
# and standard deviation drawn from a uniform distribution between 0 and 10
mat <- mapply(
  function(x) cbind(rnorm(n = 30, mean = 0, sd = x)),
  runif(n = 5, min = 0, max = 10)
)
```

```r
mat
```

```
             [,1]        [,2]       [,3]       [,4]         [,5]
 [1,]   0.0780885  3.48089343  2.9553545  3.2248615  -4.66712911
 [2,]  -8.3269637 -2.05680059  6.5670977  8.3727181 -13.86628154
 [3,]   1.2834069 -4.25078791  3.0311714 -3.6736945   2.20685092
 [4,]  12.2189488 12.51048821  2.2500211  5.8992348  12.81071523
 [5,] -14.5079803 -0.83591818 -0.8329266  1.0321989   5.73323413
 [6,]   7.8277431 -0.04931062  1.4101378  6.7058639   2.63359979
 [7,]  -2.5172557  1.27351635 -5.6927903 -12.9850971  -8.19349948
 [8,]  -9.0808601  1.07779111  3.0050245 -4.1673004   0.51438396
 [9,]   6.5582896 -3.70133731  7.1498934 -8.8572673   0.05485376
[10,]   4.1685004 -6.50775660  9.0180872 -4.5555447  -4.32601307
[11,]  10.2156894 -1.84773737  2.4000462  6.2044559  -6.12311135
[12,]  -6.1880010 -4.12393197 -2.1740754 -0.4008144   3.40181041
```

Week 3: Assignment 1

- Practice subsetting, conditional statements and iterations in R
- Due by 12:00 on Monday, 6th October