Introduction to Econometrics
Sessions 1 and 2 – Basics of Programming in R

September 2025

Before everything: create a RStudio project entitled `econometrics_intro`. Create a Rmarkdown document or a Quarto document in this project entitled `R_basics`. Use this document for the following exercises.

1 Exercise 1

Write a function that takes as input a positive integer and returns the factorial of that integer, i.e., the product of all positive integers less than or equal to it.

2 Exercise 2

Write a function that takes as input a floating-point number and determines whether it is positive, negative, or zero. The function should print "The number is positive", "The number is negative", or "The number is zero" depending on the case.

3 Exercise 3

Write a function that takes as input a year, and returns `TRUE` if this year is a leap year, and `FALSE` otherwise. A year is a leap year if it is divisible by 4 but not by 100, unless it is divisible by 400.

4 Exercise 4

Write a function that takes as input a score between 0 and 100 and returns:

- A for scores from 90 to 100
- B for scores from 80 to 89
- C for scores from 70 to 79

- D for scores from 60 to 69
- F below 60.

The function should return a warning if the score is not between 0 and 100.

5 Exercise 5

Write a function that takes as argument a natural number n and returns the sum of the squares from 1 to n.

6 Exercise 6

Write a function that takes three arguments: `temperature`, `weather`, and `day`. The function should return "Good for outdoor activities" if all the following conditions are met:

- the temperature is greater than 20 and less than 30
- the weather is "sunny"
- the day is not "Monday" or "Sunday"

If the temperature is below 20 or above 30, but the weather is still "sunny" and the day is neither "Monday" nor "Sunday", the function should return "Not the best day for outdoor activities, but still fine".

In all other cases, the function should return "Not a good day for outdoor activities".

7 Exercise 7

Write a function that takes as argument a strictly positive integer, and returns the first n terms of the following sequence. The first term is 1. The next term is 2, then $u_n = u_{n-1}u_{n-2}$ if u_{n-1} is odd, and $u_n = 2u_{n-1}$ if u_{n-1} is even. To check if a number A is even, you can use the condition `A%%2==0`.

8 Exercise 8

Write a function that takes a temperature and a string as input. The string must be either "FtoC" or "CtoF". If the string is "FtoC", the function should interpret the temperature as Fahrenheit and return the temperature in Celsius. If the string is "CtoF", the function should interpret the temperature as Celsius and return the temperature in Fahrenheit. The formula for converting Fahrenheit to Celsius is

$$(F - 32)\frac{5}{9}$$

and the formula for converting Celsius to Fahrenheit is

$$\frac{9}{5}C + 32$$

The function should return a warning if the string is not “FtoC” or “CtoF”.

9 Exercise 9

Write a function that takes as input a natural number and checks if it is a perfect square (i.e., its square root is an integer). The function should print “The number is a perfect square” if true, and “The number is not a perfect square” otherwise.

10 Exercise 10

The following function takes a number as input and returns all its divisors.

```
divisors <- function(n) {
     potential_divisors <- seq(1, n/2)
     actual_divisors <-
         potential_divisors[which(n %% potential_divisors == 0)]
     return(actual_divisors)
 }
```

Use this function to write another function that takes a non-zero natural number as input and checks if it is a perfect number. A perfect number is a non-zero natural number equal to the sum of its proper positive divisors, excluding itself. For example, the first perfect number is 6, since 1, 2, and 3 are its proper divisors, and 1 + 2 + 3 = 6. The function should return `TRUE` if the number is perfect and `FALSE` otherwise.

11 Exercise 11

Write a function that takes a positive integer as input and returns the sum of its digits.

12 Exercise 12

Write a function that takes a vector of numbers and returns the largest element in the vector without using the built-in `max()` function.

13 Exercise 13

Write a function that takes a string as input and returns the number of vowels (a, e, i, o, u) it contains.

14 Exercise 14

Write a function that takes a natural number n as input and returns a vector containing the first n prime numbers.

15 Exercise 15

Write a function that takes a vector of numbers and returns `TRUE` if the vector is sorted in increasing order and `FALSE` otherwise.

16 Exercise 16

Write a function that simulates rolling two six-sided dice. The function should return both dice results and their sum.

17 Exercise 17

Write a function that generates n random numbers between 0 and 1, and returns their mean. The function should use `set.seed()` to make results reproducible.