R: A Hitchhikers Guide to Reproducible Research

- Don't stop me now

Brendan Palmer,
Clinical Research Facility - Cork &
School of Public Health
@B_A_Palmer

CRF-C
HRB Clinical Research Facility Cork

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh
School of Public Health

Functions: when, why and how

- If you find you are copying and pasting your code a number of times in the script, the time has arrived to start learning how to write functions

- If the requirements change you only need to update the code in one location instead of many

- Incidental mistakes can be avoided
    - i.e. a change in one location not mirrored in another

- The code can be easier to read overall

Reducing duplication in your code

```
function_name <- function(arg1, arg2,...){

    text body of function / outputs
}
```

Conditionals

- An "if" statement allows you to conditionally execute code

- The condition **must** evaluate to TRUE or FALSE

```
if (y < 0) {
  print("y is negative")
} else {
    print("y is positive")
}
```

- 01_functions.R

Loops

- Note: Once you have written and tested the for loop, you'll be ready to dive into the purrr package that contains some powerful (yet easy to implement) programming tools

3 main components of for loops:

```
output <- vector("numeric", ncol(df))

for (i in 1:ncol(df)){

  output[[i]] <- mean(df[[i]]) operation

}
```

- 02_loops.R

Reproducible examples

- Open Day_3/scripts/a_reproducible_example.R

- Open Day_3/scripts/first_reprex.R