# Programming with R/Advanced R

Dries Debeer & Benjamin Becker

18. and 19. March 2021

FDZ Spring Academy

# Introduction

**Who are we?**

Dries Debeer

Researcher at KU Leuven

scDIFtest, permimp, eatATA

dries.debeer@kuleuven.be

Benjamin Becker

Researcher at IQB (Statistics Department)

eatGADS, eatDB, eatATA, pisaRT

b.becker@iqb.hu-berlin.de

# Introduction

**Who are you?**

1. Specific interests/motivation for this workshop?
2. Previous knowledge and experience?
   - with R
   - with other statistic software
   - with other programming languages

# Motivation

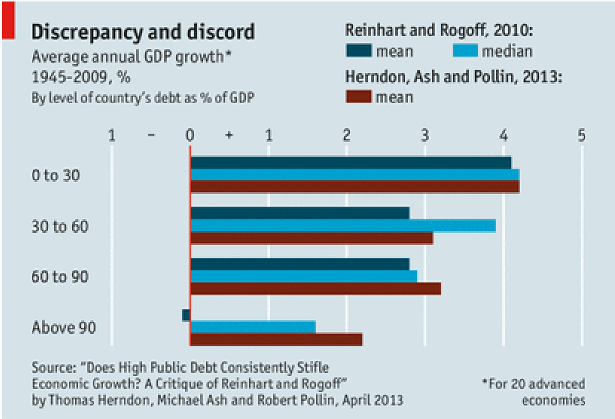
- Being more efficient in your research
    - Save time and nerves
    - Avoid errors and bugs
    - High transfer effect to all projects (which use data)
- Successful collaborations (with your future self?)
- Syntaxes as part of paper submissions

# Motivation

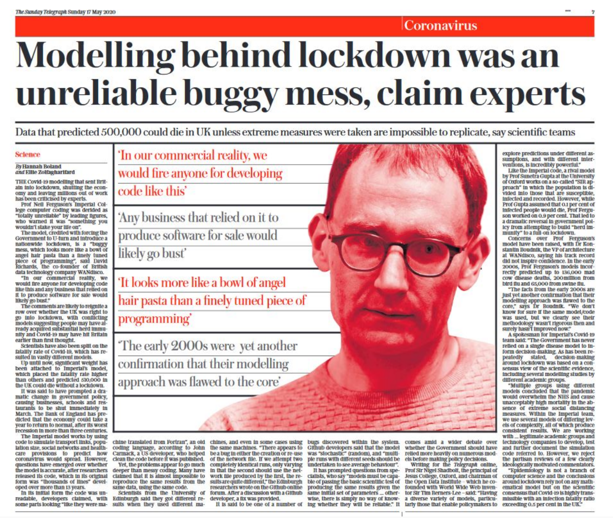
Two of your worst enemies

- Past Self
  - Is the biggest messy in existance
  - Did not document anything
  - Uses a completely different style of writing code than yourself
- Future Self
  - Has the memory of a goldfish
  - Will have zero understanding for your current brilliance

# Motivation


Discrepancy and discord
Average annual GDP growth*
1945-2009, %
By level of country's debt as % of GDP
Reinhart and Rogoff, 2010:
mean
median
Herndon, Ash and Pollin, 2013:
mean
1
-
0
+
1
2
3
4
5
0 to 30
30 to 60
60 to 90
Above 90
Source: "Does High Public Debt Consistently Stifle Economic Growth? A Critique of Reinhart and Rogoff" by Thomas Herndon, Michael Ash and Robert Pollin, April 2013
*For 20 advanced economies

# Motivation



*The Sunday Telegraph* Sunday 17 May 2020 *** 7

Coronavirus

## Modelling behind lockdown was an unreliable buggy mess, claim experts

Data that predicted 500,000 could die in UK unless extreme measures were taken are impossible to replicate, say scientific teams

'In our commercial reality, we would fire anyone for developing code like this'

'Any business that relied on it to produce software for sale would likely go bust'

'It looks more like a bowl of angel hair pasta than a finely tuned piece of programming'

'The early 2000s were yet another confirmation that their modelling approach was flawed to the core'

Science

*By* Hannah Boland *and* Ellie Zolfagharifard

THE Covid-19 modelling that sent Britain into lockdown, shutting the economy and leaving millions out of work has been criticised by experts.

Prof Neil Ferguson's Imperial College computer coding was derided as "totally unreliable" by leading figures, who warned it was "something you wouldn't stake your life on".

The model, credited with forcing the Government to U-turn and introduce a nationwide lockdown, is a "buggy mess, which looks more like a bowl of angel hair pasta than a finely tuned piece of programming", said David Richards, the co-founder of British data technology company WANdisco.

"In our commercial reality, we would fire anyone for developing code like this and any business that relied on it to produce software for sale would likely go bust."

The comments are likely to reignite a row over whether the UK was right to go into lockdown, with conflicting models suggesting people may have already acquired substantial herd immunity and Covid-19 may have hit Britain earlier than first thought.

Scientists have also been split on the fatality rate of Covid-19, which has resulted in vastly different models.

Up until now, significant weight has been attached to Imperial's model, which placed the fatality rate higher than others and predicted 510,000 in the UK could die without a lockdown.

It was said to have prompted a dramatic change in government policy, causing businesses, schools and restaurants to be shut immediately in March. The Bank of England has predicted that the economy could take a year to return to normal, after its worst recession in more than three centuries.

The Imperial model works by using code to simulate transport links, population size, social networks and healthcare provisions to predict how coronavirus would spread. However, questions have emerged over whether the model is accurate, after researchers released its code, which in its original form was "thousands of lines" developed over more than 13 years.

In its initial form the code was unreadable, developers claimed, with some parts looking "like they were machine translated from Fortran", an old coding language, according to John Carmack, a US developer, who helped clean the code before it was published.

Yet, the problems appear to go much deeper than messy coding. Many have claimed that it is almost impossible to reproduce the same results from the same data, using the same code.

Scientists from the University of Edinburgh said they got different results when they used different machines, and even in some cases using the same machines. "There appears to be a bug in either the creation or re-use of the network file. If we attempt two completely identical runs, only varying in that the second should use the network file produced by the first, the results are quite different," the Edinburgh researchers wrote on the Github online forum. After a discussion with a Github developer, a fix was provided.

It is said to be one of a number of bugs discovered within the system. Github developers said that the model was "stochastic" (random), and "multiple runs with different seeds should be undertaken to see average behaviour".

It has prompted questions from specialists, who say "models must be capable of passing the basic scientific test of producing the same results given the same initial set of parameters ... otherwise, there is simply no way of knowing whether they will be reliable." It comes amid a wider debate over whether the Government should have relied more heavily on numerous models before making policy decisions.

Writing for the *Telegraph* online, Prof Sir Nigel Shadbolt, the principal of Jesus College, Oxford, and chairman of the Open Data Institute – which he co-founded with World Wide Web inventor Sir Tim Berners-Lee – said: "Having a diverse variety of models, particularly those that enable policymakers to explore predictions under different assumptions, and with different interventions, is incredibly powerful."

Like the Imperial code, a rival model by Prof Sunetra Gupta at the University of Oxford works on a so-called "SIR approach" in which the population is divided into those that are susceptible, infected and recorded. However, while Prof Gupta assumed that 0.1 per cent of infected people would die, Prof Ferguson worked on 0.9 per cent. That led to a dramatic reversal in government policy from attempting to build "herd immunity" to a full-on lockdown.

Concerns over Prof Ferguson's model have been raised, with Dr Konstantin Boudnik, the VP of architecture at WANdisco, saying his track record did not inspire confidence. In the early 2000s, Prof Ferguson's models incorrectly predicted up to 136,000 mad cow disease deaths, 200 million from bird flu and 65,000 from swine flu.

"The facts from the early 2000s are just yet another confirmation that their modelling approach was flawed to the core," says Dr Boudnik. "We don't know for sure if the same model/code was used, but we clearly see their methodology wasn't rigorous then and surely hasn't improved now."

A spokesman for Imperial's Covid-19 team said: "The Government has never relied on a single disease model to inform decision-making. As has been repeatedly stated, decision-making around lockdown was based on a consensus view of the scientific evidence, including several modelling studies by different academic groups.

"Multiple groups using different models concluded that the pandemic would overwhelm the NHS and cause unacceptably high mortality in the absence of extreme social distancing measures. Within the Imperial team, we use several models of differing levels of complexity, all of which produce consistent results. We are working with ... legitimate academic groups and technology companies to develop, test and further document the simulation code referred to. However, we reject the partisan reviews of a few clearly ideologically motivated commentators.

"Epidemiology is not a branch of computer science and the conclusions around lockdown rely not on any mathematical model but on the scientific consensus that Covid-19 is highly transmissible with an infection fatality ratio exceeding 0.5 per cent in the UK."

# Motivation

Concept of Technical Debt

- We write (messy) code for data cleaning/analyses
- We decide on data sets/models/graphs/tables/...
- We try to publish it, get a major revision
- We need to rerun some analyses
- Modifying/extending our code is more difficult than it should be

Solutions

- Refactor/rewrite your could before submitting
- **Write better R code**

# Goals of this workshop

- Better practical R skills
- Better theoretical understanding of R (and programming)
- Different framing: R as a programming language

# R Objects (Recap)

# Clean Code

# Iteration

# Functions I

# Why?

- Readability
  - Shorter
  - Easier understanding
  - Removes distractions, like references in a paper
- Transferability
  - Other use cases
  - Other projects
  - Other persons

# Why?

```
mean(mtcars$mpg)
```

```
[1] 20.09062
```

```
# vs.
sum(mtcars$mpg)/dim(mtcars)[1]
```

```
[1] 20.09062
```

# Why?

```
summary(mtcars$mpg)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.40   15.43   19.20   20.09   22.80   33.90

# vs.
round(c("Min." = min(mtcars$mpg),
  "1st Qu." = as.numeric(quantile(mtcars$mpg)[2]),
  "Median" = median(mtcars$mpg),
  "Mean" = mean(mtcars$mpg),
  "3rd Qu." = as.numeric(quantile(mtcars$mpg)[4]),
  "Max." = max(mtcars$mpg)), 2)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.40   15.43   19.20   20.09   22.80   33.90
```

# How?

```r
countNA <- function(x) {    # Name, Arguments/Formals
  out <- sum(is.na(x))      # Body
  out                       # Output
}
```

# How?

- Before
    - What should my function do?
    - Input (Arguments)
    - Output
- Process
    - Write function
    - Test it
    - Add input validation
    - Document it

# Functions II

# What is a good function?

- pure functions
  - no sideeffects
  - no dependency on global environment
  - easier understanding, easier transfer!

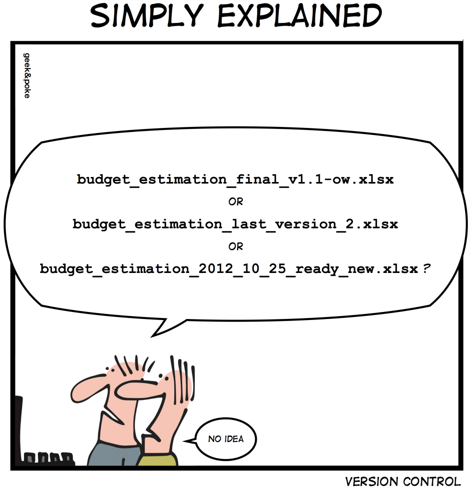
# Debugging

- browser()
- traceback()
- options(error = recover)

# Object Oriented Programming (S3)

# Version Controlling (Git + Github)

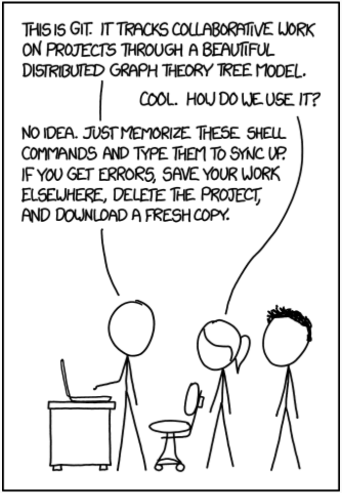
# Motivation


SIMPLY EXPLAINED
geek&poke
budget_estimation_final_v1.1-ow.xlsx
OR
budget_estimation_last_version_2.xlsx
OR
budget_estimation_2012_10_25_ready_new.xlsx ?
NO IDEA
VERSION CONTROL

# Motivation

- Implementation of long term change history
    - No ridiculous file names
    - No archive subfolder
    - Always perfect overview of file history and changes
- Collaborations
    - What has changed?
    - Who has changed it?
    - Documentation of changes
    - Parallel working possible (merging)

# But...


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

# Requirements

- Install git
- Install User Interface for git (RStudio, Gitkraken, ...)
- Setup account for Github/Bitbucket/Gitlab/...

# Workflow

## Creating a repository

- Create an online repository (e.g. on Github)
- Use an R specific .gitignore
- Initialize with a short readme
- Clone the repository to your local machine
- (optional) Place an R project in the existing repository

# Workflow

### Working with a repository

- Before working: Synch your local repo (pull)
- Perform changes in your local repository
- Stage your changes
- Commit your changes
- Push your changes

# Recommendations

- Keep it simple!
    - No branches/forks/pull requests
    -
- Have meaningful commits
- Keep it lean (no big files)

# Resources

Git (+ R) Resources

- Small Intro (`https://r-bio.github.io/intro-git-rstudio/`)
- Happy Git with R (`https://happygitwithr.com/`)
- R Packages and Git (`https://r-pkgs.org/git.html`)
- Git Book (`http://git-scm.com/book/en/v2`)

# Literature Recommendations

R Resources

- Avanced R Ed. 1 (`http://adv-r.had.co.nz/`)
- Avanced R Ed. 2 (`https://adv-r.hadley.nz/`)
- R Inferno (`https://www.burns-stat.com/pages/Tutor/R_inferno.pdf`)
- R Packages (`https://r-pkgs.org/`)
- Clean Code (`https://enos.itcollege.ee/~jpoial/oop/naited/Clean%20Code.pdf)`)

Thank you for your attention!

Thank you for your attention!

Questions? Remarks?