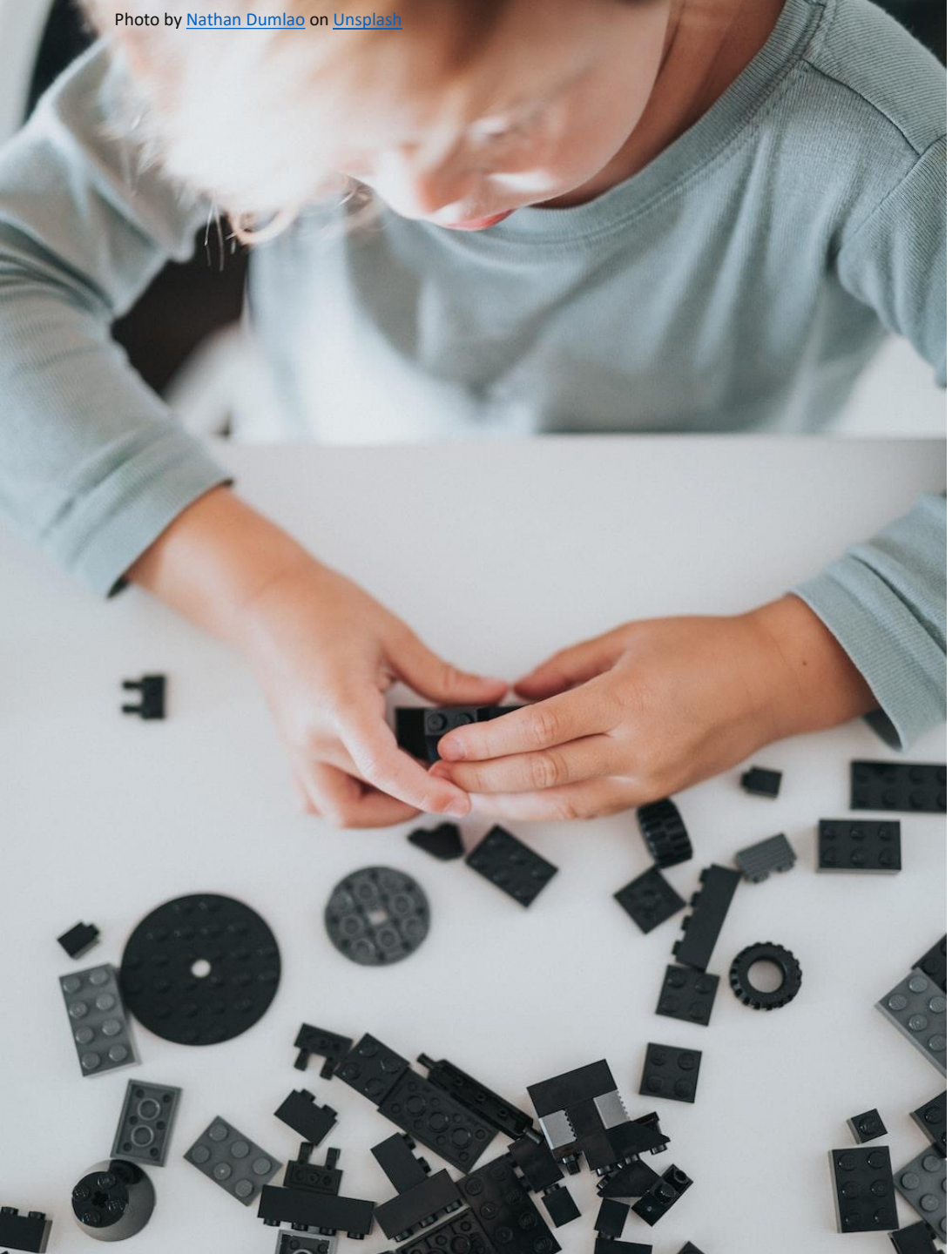


Photo by Nathan Dumlao on Unsplash

Barriers to reproducible research in safe haven settings - and how to overcome them

Andreas Höhn
RSS Glasgow Local Group
14/06/2023

Photo by Iván Díaz on Unsplash

Introduction

Current Roles:

- System Science in Public Health Team (@ MRC/CSO, UofG)
- SIPHER (@UKPRP Consortium)
- MigrantLife (@ERC Project, St. Andrews)

Experience of Data Safe Haven Settings:

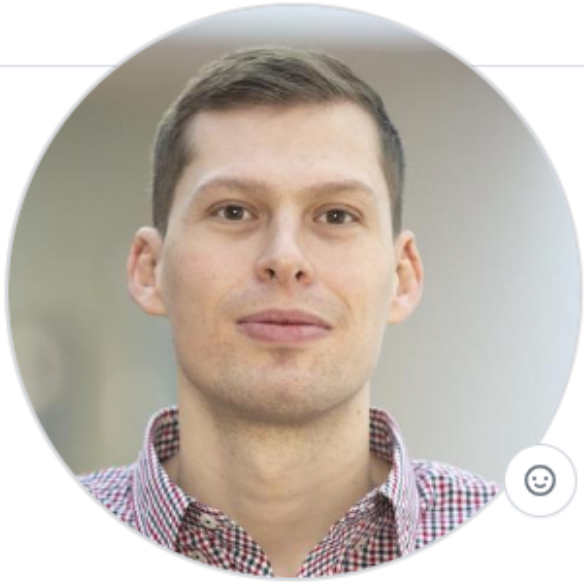
- Sweden
- Denmark
- Scotland

Outline

1. Introduction: Making a Case
2. Demonstration: The Magic of Reproducibility
3. Overcoming Barriers to Reproducibility
4. Q & A

Photo by Apolo Photographer on Unsplash

Introduction



Andreas Höhn

AndreasxHoehn

(Health) Data Science and Demography

Edit profile

13 followers · 17 following

University of Glasgow - MRC/CSO Social and Public Health Science Unit

Overview | Repositories 11 | Projects | Packages | Stars

Pinned

Customize your pins

QALE_Exemplar Public

Estimating Quality-Adjusted Life Expectancy - alongside other population health metrics such as Life Expectancy and Lifespan Variation - for UK Local Authority Districts using publicly available da...

R ☆ 2 2

Imputation_TimeSeries Public

minimum working example of how to impute gaps in time series data N times and stack results (as mean or median) in 1 final dataset

R

MSM_Working_LifeExpectancy Public

Estimating Working Life Expectancy from a Parametric Multistate Model using "flexsurv" and Validating the Obtained Results using Chiang's 1984 Life Tables

HTML

Parametric_LifeExpectancy Public

estimating life expectancy straight from individual-level data using a parametric survival model and validating the results with Chiang 1984

R

PhD_Thesis Public

PhD Thesis Gender, Hospitalization, and Mortality, University of Southern Denmark, Defended 03/2020

TeX

Research_Pipeline_Example Public

This is the example used for a presentation to the Royal Statistical Society Glasgow, 06/2023

R ☆ 2

https://github.com/AndreasxHoehn/Research_Pipeline_Example

Introduction

Data Safe Haven Settings:

- Secure access
- Limited cross-checking
- Often large-scale data
- Often uncleaned data
- Often complex process
- Increasingly batch mode and HPC processes

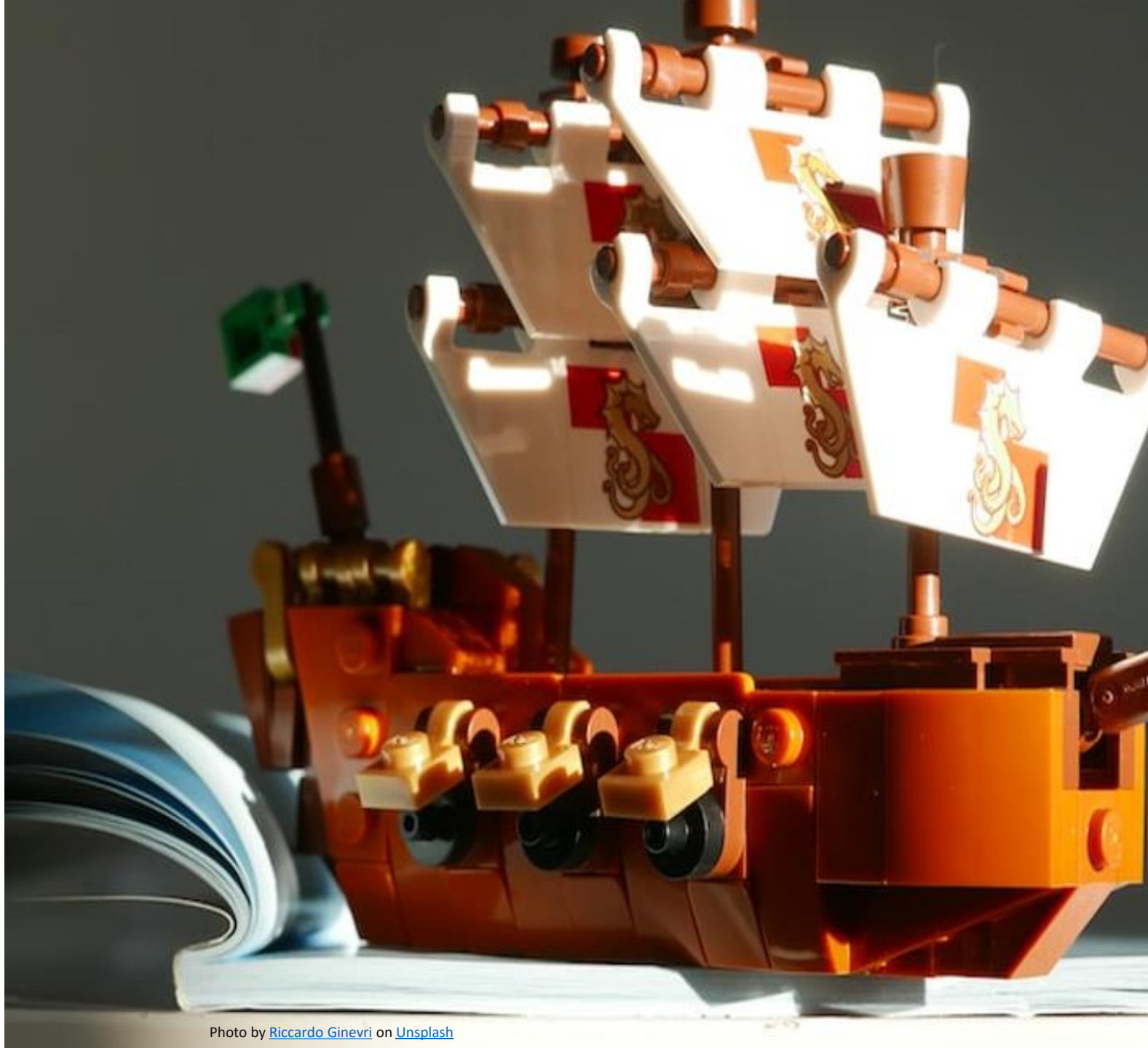


Photo by Riccardo Ginevri on Unsplash

Introduction

Reproducibility in Safe Haven Settings?

Let's not bother?

Photo by Xavi Cabrera on Unsplash

Introduction

estimated separate models by cause of admission to hospital to investigate whether the female advantage in survival following hospitalisation varies across different causes of admission. While the data preparation and the merging of registries was carried out with STATA (V.15), all statistical analyses were performed in R (V.3.3.2).

Source: Höhn et al. 2018, doi: 10.1136/bmjopen-2018-021813

Introduction

Motivation:

- Efficiency
- Integrity
- Credibility
- Safety
- Teamworking
- Funder’s Guidelines

Photo by Xavi Cabrera on Unsplash

Demonstration: The Magic of Reproducible Research

Barriers to reproducible research - and how to overcome them

1. Flow is Disrupted
2. Code is not Modular
3. Poor Readability of Code
4. Slow Performance

Barrier: I end up mixing up results and numbers when running the code.

Solution: Identify important definitions and rules!

Barrier: I end up mixing up results and numbers when running code.

Solution: identify important definitions and rules!

```
46
47  # ------------------------------------------------------------------------ #
48
49  ### [B] Define Study Outline, Cut-Offs etc. ###
50
51  # define global variables DONT HARD CODE THEM LATER!
52  # in this example we want to be able to quickly switch between males females
53  # subset variable "subset_sex":
54  # change the value here and you get the entire paper based on this change
55  definitions <- list()
56  definitions$subset_sex <- "female" # either: male/female
57  # definitions$subset_sex <- "male" # either: male/female
58
59  # ------------------------------------------------------------------------ #
60
```

Barrier: My code files get messy and too long.

Solution: Make it modular!

Photo by Mourizal Zativa on Unsplash

```r
# ------------------------------------------------------------------

### [1] Source File: Build and Specify the Study Population ###
source("RCode/01_data_preparation.R")

# ------------------------------------------------------------------

### [2] Source File: Run Analysis ###
# source analysis file - means run what's in the file.
# object x shall appear in the work space
source("RCode/02_data_analysis.R")

# ------------------------------------------------------------------

### [3] Source File: Built the Paper ###
# csl files from: https://www.zotero.org/styles
# source bib files as usual
# rmarkdown might requires here:: file paths due to location of pandoc
rmarkdown::render(input = "RCode/03_manuscript.Rmd",
      output_file = here::here("ROutput/manuscript_.docx"),
      output_format = "word_document")
```

Barrier: My code files get messy and too long.

Solution: Make it modular!

Barrier: I often struggle to read the code which I and my team have written.

Solution: keep it easy, keep it tidy!

Barrier: I often struggle to keep on top of code and objects in my workspace

Solution: keep it easy, keep it tidy!

```
14  # creating a table
15
16  # using a function that just depends on a data object we pass in
17  # we do this in a function to make sure we don't get a messed up work space
18  # the table will look ugly and won't be formatted for open office
19  # however, it looks kind of okay in MS Word
20
21  .MakeTable <- function(data_input) {
22    tab_N   <- data.frame(
23      Summary = "Individuals",
24      Value   = length(unique(data_input$ID,2)))
25    tab_age <- data.frame(
26      Summary = "Mean Age",
27      Value   = sprintf("%.2f", round(mean(data_input$age), 2)))
28    tab_inc <- data.frame(
29      Summary = "Mean Income",
30      Value   = sprintf("%.2f", round(mean(data_input$income), 2)))
31
32    table <- rbind(tab_N, tab_age, tab_inc)
33    return(table)
34  }
```

Barrier: My code takes too long to run, and it limits me.

Solution: make it fly!

Photo by Nik on Unsplash

Example #1: Memory Use

```
 3  # create 100 males
 4  V1 <- rep(as.character("Question_1_Sex_Male"),
 5            times = 100)
 6  V2 <- rep(as.character("1"), times = 100)
 7  V3 <- rep(as.factor(1), times = 100)
 8  V4 <- rep(as.numeric(1), times = 100)
 9  V5 <- rep(as.integer(1), times = 100)
10
11  # compare 1000 males
12  object.size(V1) # 928 bytes
13  object.size(V2) # 904 bytes
14  object.size(V3) # 896 bytes
15  object.size(V4) # 848 bytes
16  object.size(V5) # 448 bytes
```

Barrier: My code takes too long to run, and it limits me.

Solution: make it fly!

Example #2: Dialect

```r
# base R way
read_base <- microbenchmark::microbenchmark(
  read.csv("RData/A_Large_File.csv",skip = 0, header = TRUE),
  times = 1)

# tidyverse
read_tidy <- microbenchmark::microbenchmark(
  readr::read_csv("RData/A_Large_File.csv"),
  times = 1)

# data.table
read_dt <- microbenchmark::microbenchmark(
  data.table::fread("RData/A_Large_File.csv"),
  times = 1)

# which is the fastest: seconds
read_base$time / 10e8  # 6.4 sec
read_tidy$time / 10e8  # 2.8 sec
read_dt$time   / 10e8  # 1.1 sec
```

Barrier: My code takes too long to run, and it limits me.

Solution: make it fly!

Summary

Find your motivation for reproducibility

Find the sustainable changes you can make

Identify good return-on-investments

Photo by Vlad Hilitanu on Unsplash

Andreas Höhn

Andreas.hoehn@glasgow.ac.uk

0000-0002-7170-1205

AndreasXHoehn