Reproducible Data Science for Public Health

A practical guide for researchers

Hélène Langet Samwel Lwambura
Fenella Beynon Silvia Cicconi
Gillian Levine

2024-04-17

Table of contents

Preface

Introduction

Data science and artificial intelligence have the potential to generate fundamentally new insights on global health policies in Africa, but the full realization of this potential depends on the availability of a critical mass of highly trained health data scientists on the continent.

This electronic book was originally created by Hélène Langet, Fenella Beynon, Silvia Cicconi and Gillian Levine from the Swiss Tropical & Public Health Institute (Swiss TPH) and Samwel Lwambura from the Ifakara Health Institute (IHI) to accompany the **Data Science for Public Health** workshop which was held in Dar-es-Salaam from Monday,September 26th to Wednesday, September 28th 2022. The goal of this workshop was to enable IHI researchers to strengthen their expertise in the area and to lay the foundations for the development of a data science curriculum that is adapted to IHI's needs.

The content of this electronic book is currently being updated and restructured. This revision aims to incorporate some of the latest technical developments and to refine explanations of key reproducible research and data science concepts to better suit a public health graduate audience.

License

- Share — copy and redistribute the material in any medium or format
- Adapt — remix, transform, and build upon the material for any purpose, even commercially.

Under the following terms:

- Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Update History

- September 25, 2022: First edition.
- November 1st, 2022: Updated resources.
- Q2 2024: planned release of the second edition, reflecting insights gained from teaching at Swiss TPH.

Acknowledgement

This work was funded by the Leading House Africa (LHA) which promotes and fosters bilateral collaboration with partner institutions in Africa.

This work was not led in isolation and is the product of interdisciplinary discussions with the following research and public health professionals in Europe and in Sub-Saharan Africa: * Ms Moniek Bresser, Dr Aurelio Di Pasquale, Prof Manuel Hetzel, Dr Fabian Schär from Swiss TPH (Switzerland) * M. Ibrahim Mtebene, Dr Abdallah Mkopi, Dr Grace Mhalu, and Dr Robert Moshiro from IHI, Charles Festo, Hajirani Msuya and Martine Masonda from IHI (Tanzania) * Prof Jean Augustin Tine from the Cheikh Anta Diop University of Dakar (UCAD, Senegal) * M. Francis Njiri from University of Nairobi (UoN, Kenya)

1 Introduction

1.1 Learning objectives

Two complementary aspects of moving into data science are:

1. the mindset about how scientists think and collaborate about data, and
2. the skillsets which is composed of an ecosystem of tools (mostly open-source) and practices.

Upon completing the workshop, participants will have gained:

- exposure to data science approach, tools and collaborative practices
- hands-on experience on how to interface between Stata and R, learned the basics of working with data in R/RStudio, and how to incrementally incorporate R into your existing data analysis workflows in Stata. The idea is not to replace everything you do in Stata into R but that you can continue your learning after this workshop at your own pace.

1.2 Is this workshop for me?

This workshop is relevant for individuals who answer yes to the following questions:

- ☐ Do you who want to develop data science projects in public health?
- ☐ Do you wants to learn more about how open and reproducible science approaches can be used in your daily practice?

- ☐ Are you a Stata user (or any other data analysis language) who would like to expand your data analysis skillset with R?
- ☐ Do you want to bridge analyses between data analysis tools (Stata, R or Python) and to more easily collaborate with other researchers who use another of these tools?

1.3 Schedule

September 26-28, 2022
09:00 - 17:00
Dar-es-Salaam, Tanzania (Protea Hotel by Marriott Dar es Salaam Courtyard)

1.3.1 Before the workshop

1. Fill out the online pre-workshop questionnaire
2. Install on your laptop the (free) data science software that will be used during the workshop. If you have any difficulties with the installation, support can be provided on the first day of the workshop before the first session or during breaks.

1.3.2 Day 1

Table 1.1: Schedule Day 1

Time	Session
08.30 - 09.00	Welcome Support for software installation
09.00 - 09.15	Introduction to data science tools Overview of objectives for Day 1
09.15 - 10.30	Version control with Git
10.30 - 11.00	Break
11.00 -12.00	Introduction to dynamic documents and Quarto
12.00 - 13.00	Use Quarto with Stata

Time	Session
13.00 - 14.00	Lunch break
14.00 - 15.00	Import and manipulate external data (1)
15.00 - 15.30	Import and manipulate external data (2)
15.30 - 16.00	Break
16.00 - 17.00	Share code and Collaborate with Git

1.3.3 Day 2

Table 1.2: Schedule Day 2

Time	Session (all)
08.30 - 09.00	Welcome
09.00 - 09.15	Introduction to Data Science for Public Health Overview of objectives for Day 2
09.15 - 10.30	Discussion on concepts related to to health data for decision-making
10.30 - 11.00	Break
11:00- 11:15	Malaria use case - Presentation of the data
11.15 - 11.45	Malaria use case - Interdisciplinary discussion
11.45 - 12.30	Malaria use case - Data practicals by interdisciplinary groups
12.30 - 13.00	Malaria use case - Feedback on findings from practicals
13.00 - 14.00	Lunch break
14.00 - 14.30	Malaria use case - Interdisciplinary discussion
14.00 - 15.30	Malaria use case Analysis: data practicals Interpretation: discussion on data sources and interpretation

Time	Session (all)
15.30 - 16.00	Break
16.00 - 17.00	Malaria use case - Feedback on praticals

1.3.4 Day 3

Table 1.3: Schedule Day 3

Time	Session (all)
08.30 - 09.00	Welcome
09.00 - 09.15	Interdisciplinary introduction to big data and machine Learning Overview of objectives for Day 3
09.15 - 10.00	Discussion on secondary data sources (Public Datasets, e.g. DHS, Facebook, facilities, etc) Benefits and drawbacks between primary and secondary data sources
10.00 - 10.30	Break
11.00 - 13.00	Analysis: Introduction to machine learning Interpretation: Critically discuss data surveys/reports
13:00-14:00	Lunch break
14:00-14:30	Speed talks - research presentations
14:30-15:30	Feedback on findings from practicals
15:30-16:30	Feedback on workshop - Wrap-up

1.3.5 After the workshop

1. Fill out the online post-workshop questionnaire

1.4 Scope

This workshop aims to accompany researchers to progress on the following development axes:

1.4.1 Data science mindset

- Use of reproducible research practices in public health
- Data provenance
  - Use of distinct data sources for the development of public health indicators
  - Research data vs. real world evidence data
- Ethical data science
- Data papers

1.4.2 Data science skillset

- Programming tools
  - Move from Stata to R (prerequisite: Stata)
  - R programming
    - dplyr
  - Python programming
    - pandas
    - scikit-learn (prerequisite: independent Python user)
- Coding with best practices (R/RStudio/tidyverse)
  - Versioning using GitHub (all)
  - Using targets (prerequisite: independent R user)
- Reporting and publishing: Dynamic report generation
- Reproducible data
  - Use APIs (prerequisite: IT programming basics)
  - Open access data (all)
- Statistical methods for reproducible research (advanced)

1.4.3 What is not covered

- Reproducible workflows (targets)
- Reproducible environments (Binder, Docker, renv, etc)

1.5 Conventions

Discussion activity

Reflection activity

Coding activity

Part I

——— DAY 1

2 Tools for reproducible quantitative research

2.1 Introduction

2.1.1 Overview

When reporting scientific results, researchers must document the steps they followed so that independent researchers within the broader research community are able to trust and build upon their findings (*cumulative knowledge*).

> **!** Important
>
> Reporting should include negative results as they contribute to the development of a cumulative knowledge as much as positive results do and avoid wasting resources.

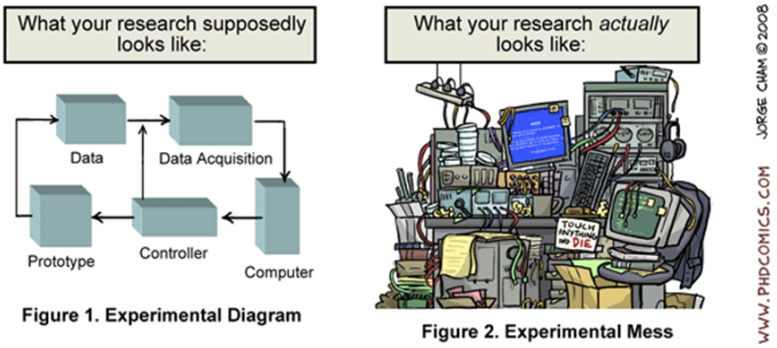


Figure 2.1: Research Diagram vs. Research Reality. "Piled Higher and Deeper" by Jorge Cham www.phdcomics.com

There have been recurrent calls in the recent decades for the scientific community to embrace practices to support research reproducibility and many software tools are now available to facilitate this process.

2.1.2 Learning objectives

- What is meant by *reproducible quantitative research* and why does it matters?
- What should be documented for ensuring the reproducibility of quantitative analyses?
- What software tools are available to support reproducible quantitative research?

2.2 Reproducible quantitative research

A quantitative analysis is said to be (*computationally*) **reproducible** when the **same analysis steps** performed on the **same dataset** consistently produce the **same quantitative results** (1). Given the deterministic nature of computer programmes, a quantitative analysis must be reproducible to be **credible**.

In addition to (computational) reproducibility, **replicability**, **robustness** and **generalisability** are key to the generation of strong quantitative evidence.

2.2.1 Replicability

The **same analysis** steps are performed on **different datasets** and produce **qualitatively similar answers** (1).

2.2.2 Robustness

Different analysis steps are performed on the **same dataset** to answer the **same research question** and produce **qualitatively similar or identical answers** (1).

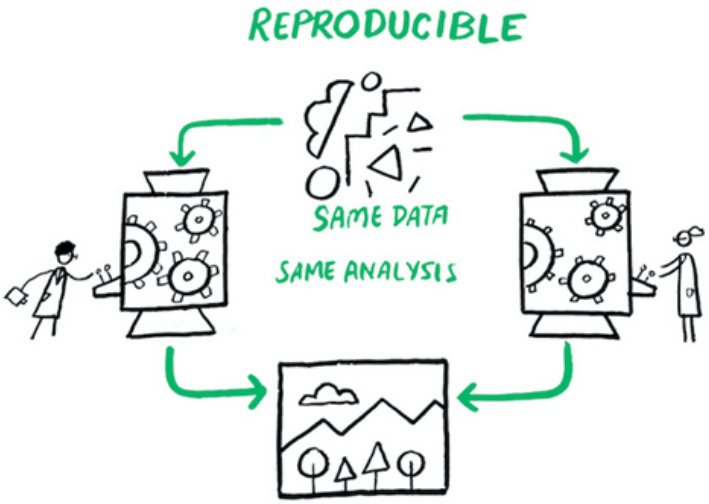


Figure 2.2: This image was created by Scriberia for The Turing Way community and is used under a CC-BY licence. DOI: 10.5281/zenodo.3332807

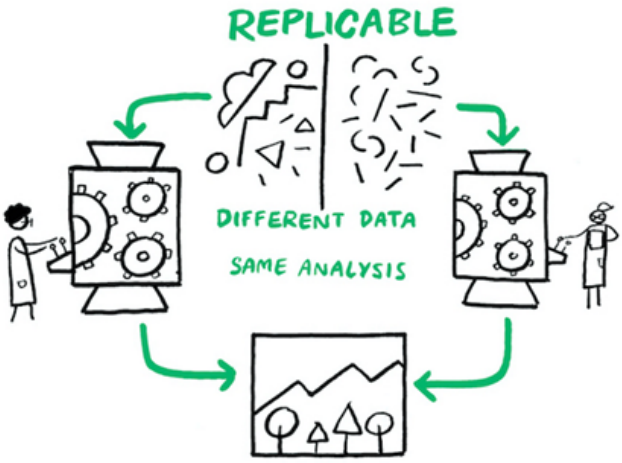


Figure 2.3: This image was created by Scriberia for The Turing Way community and is used under a CC-BY licence. DOI: 10.5281/zenodo.3332807

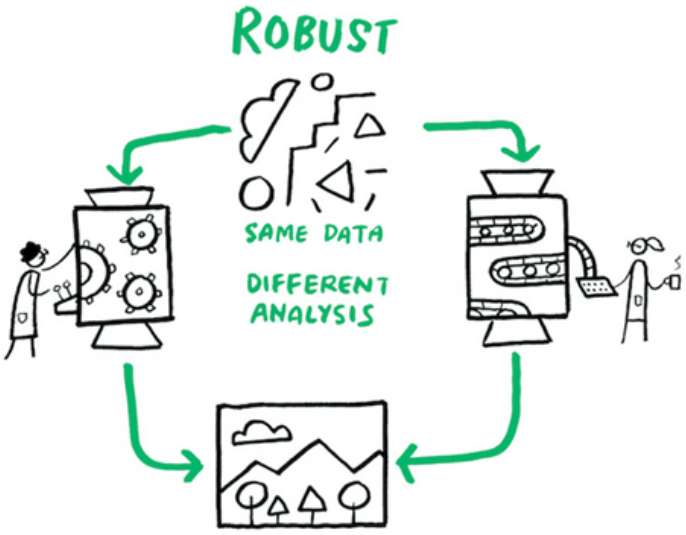


Figure 2.4: This image was created by Scriberia for The Turing Way community and is used under a CC-BY licence. DOI: 10.5281/zenodo.3332807

2.2.3 Generalisability

Different analysis steps are performed on the **different datasets** to know how well the work applies to all the different aspects of the research question and produce **generalisable answers** (1).

2.3 Good documentation for reproducible analyses

To guarantee that any other researcher can reproduce your analysis, you would need to document and share the full **computational environment**, **tools**, **data** and **code** that were used to generate your results.

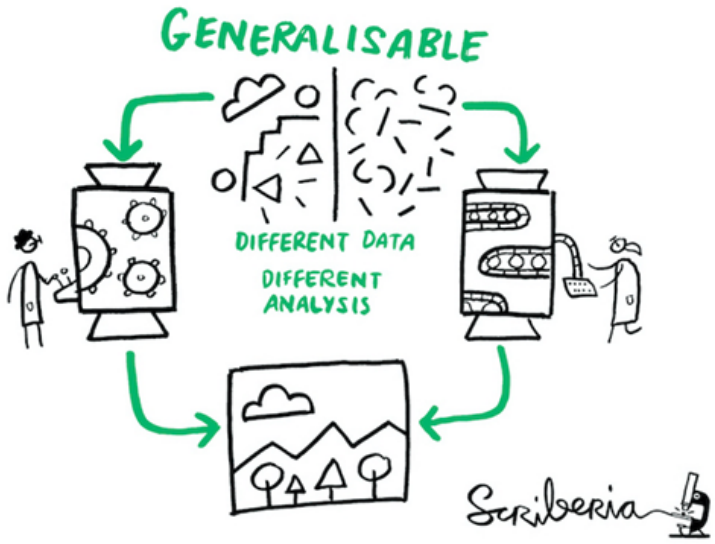


Figure 2.5: This image was created by Scriberia for The Turing Way community and is used under a CC-BY licence. DOI: 10.5281/zenodo.3332807

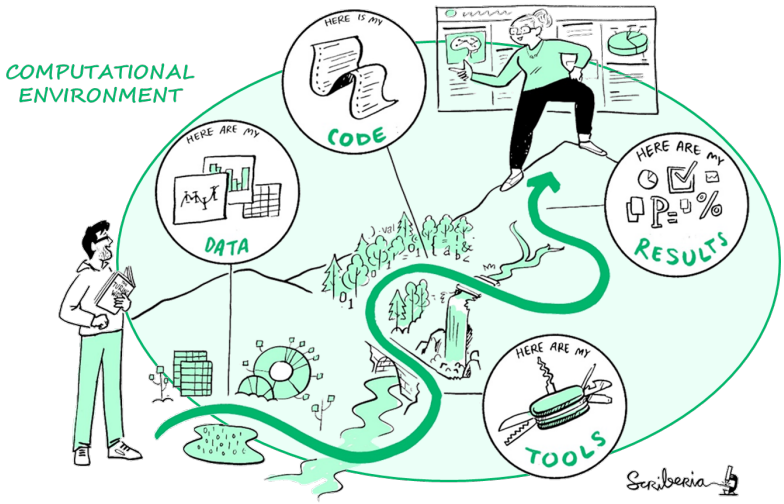


Figure 2.6: This image was adapted from an original image created by Scriberia for The Turing Way community (CC-BY licence. DOI: 10.5281/zenodo.3332807).

2.4 Opportunities and challenges of reproducible research

2.4.1 Discussion

1. How do you think reproducible quantitative analyses can improve your research?
2. Recent investigations have shown that a significant percentage of scientific studies cannot be reproduced, thus contributing to growing mistrust in scientific results (2,3). What barriers and challenges to reproducible research do you see in your daily practice?

5 minutes

2.4.2 Opportunities

Figure 2.7: This image was created by Scriberia for The Turing Way community and is used under a CC-BY licence. DOI: 10.5281/zenodo.3332807

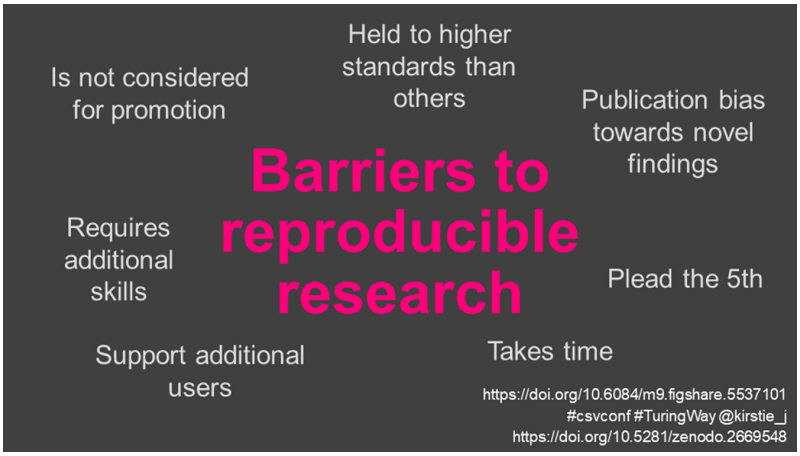
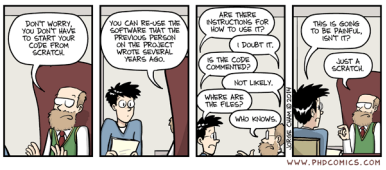


Figure 2.8: Example of barriers to reproducibility

2.4.3 Barriers and challenges

2.4.4 Challenges

In practice, reproducibility is challenging, even for trained data scientists equipped with an arsenal of software tools. Quantitative analyses can often not be fully reproduced because of complexities in how software tools are packaged, installed, and executed and because of limitations associated with how scientists document analysis steps.



2.5 Software tools across the research data lifecycle

As illustrated in Figure **??**, there is now a whole set of free or open source software tools that are available to help you automate your processes and overcome reproducibility challenges across the research data lifecycle.

In this workshop, you will be introduced to following software tools that will help make your quantitative data processing, study and analysis more reproducible:

- **Git/GitHub** allow you to keep track of various versions of your code, share your code and collaborate with others on code development;
- **R** is a programming language for statistical computing and graphics and one of main programming language used for data science (with other programming languages such as **Python** and Julia). The ecosystem around R provides an interactive environment for data science science workflows, thus making R is a great start for your data science journey.
- **Rstudio** is an integrated development environment (IDE) for R that enables an easier use of R.
- **Quarto** allow you to generate (reproducible) dynamic reports to document your data analyses. We will use Quarto within the **R/RStudio** environment.

> **!** Important
>
> In practice, most data scientists use a mix of languages, often at least R and Python. You will be slightly exposed to R, but the goal of this workshop is

> **i** Note
>
> Although qualitative research contributes as significantly as quantitative research to knowledge generation, the validation of qualitative research findings is a much more complex and debated concept as qualitative analysis is by essence subjective and contextual. This explains the lower availability of software tools dedicated to qualitative research compared to what is available for quantitative research.

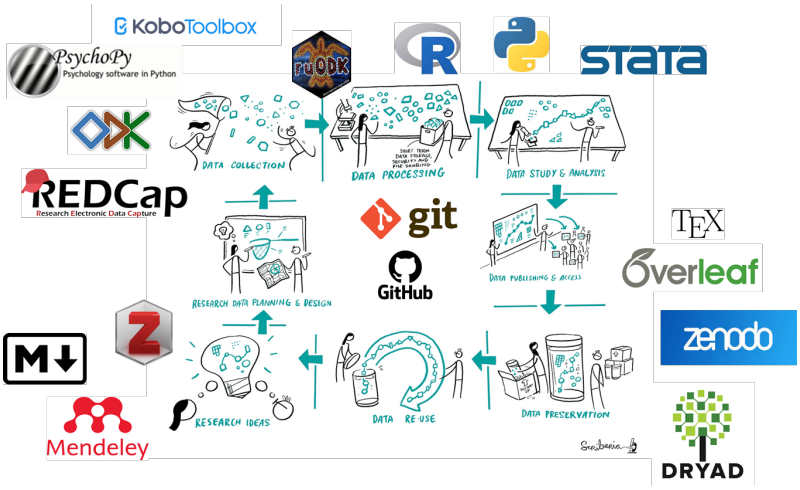


Figure 2.9: Example of free / open source software tools across the research data lifecycle. This image was adapted from an original image created by Scriberia for The Turing Way community (CC-BY licence. DOI: 10.5281/zenodo.3332807).

2.6 To go further with reproducibility

- Binder/Docker: reproducible environments
- Targets: reproducible workflows

3 Version control with Git

3.1 Introduction

3.1.1 Overview

Does the following situation seem sadly familiar? The challenge here is not only that you have no idea which draft is actually the latest version of the document, but also that it is almost impossible to understand what decision on the document content was taken when.

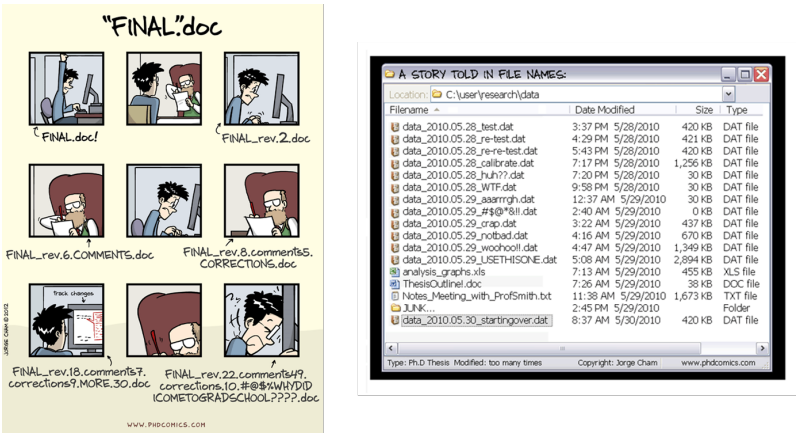


Figure 3.1: "Piled Higher and Deeper" by Jorge Cham www.phdcomics.com

Version control is the process by which the development of a document is clearly identified. It provides huge benefits to organization, archiving, and being able to find your files easily when you need them. Git is a tool that

3.1.2 Learning objectives

In this chapter, we introduce the basic elements of version control. We will learn the terminology and practice version control on a *need-to-know* basis across the workshop.

- What is version control?
- What tools are available to support version control?
- How to set up Git version control for a project?

3.2 Version control

Version control generally applies at a level of a project. It tracks and manages different drafts and versions for each document in the project.

> **!** Important
>
> With version control, you will only see a single file, which is the most recent version (*final* version). This helps avoiding confusion.

Version control provides an audit trail for the revisions and updates of final versions.

> Tip
>
> Version control tracks **what** changes have been made, **by whom** and **when**, so that you do no longer need to save a copy of your documents with your name or the date in the filename.

Version control allows you to discard recent updates and restore an earlier version of our project if needed.

3.3 Git, GitHub and GitHub Desktop

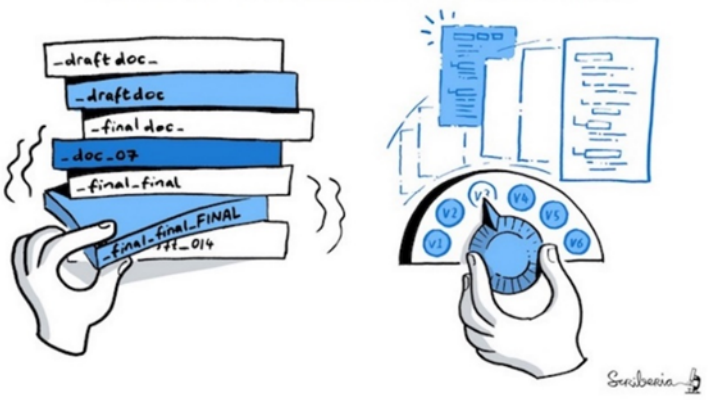


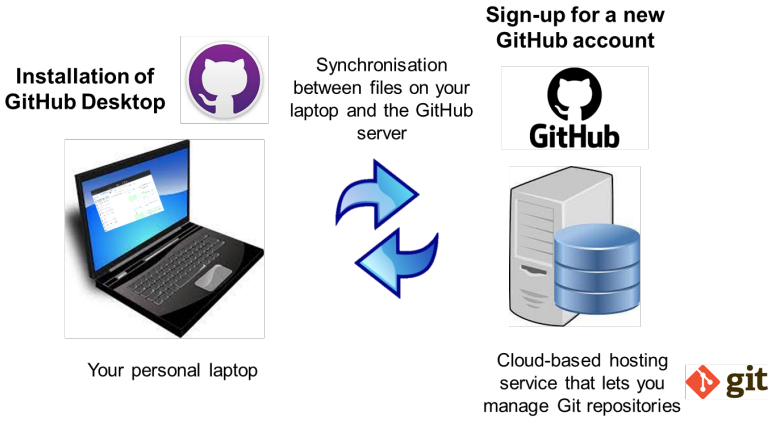
Figure 3.2: This image was created by Scriberia for The Turing Way community and is used under a CC-BY licence. DOI: 10.5281/zenodo.3332807

git	**Git**	Free and open programme that tracks changes to your files over time.
GitHub	**GitHub**	Cloud-based hosting platform that lets you host and manage Git repositories. GitHub synchronizes your local files online and enables you to collaborate with others (and yourself).
	GitHub Desktop	Application that allows to use Git commands seamlessly through a visual interface instead of using the command line (for software developers) or GitHub's website.

We will always use these three tools together.

Please refer to the following sections for instructions on creation/installation steps:

- Create a GitHub account (Section **??**)
- Install GitHub Desktop (Section **??**)

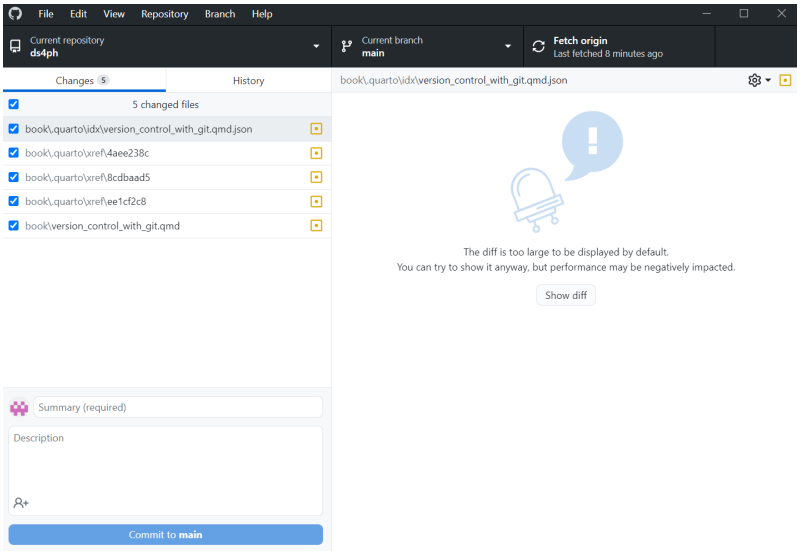


3.4 Set up Git version control for a project

> **Note**
>
> A **repository** is a database of all changes in your project. You will have a personal copy of all the final versions of the documents in the project (*working copy*) which will appear on your computer as a folder.

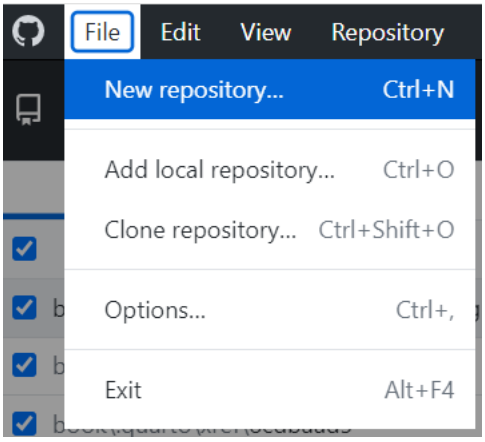
3.4.1 Step 1

Open GitHub Desktop



3.4.2 Step 2

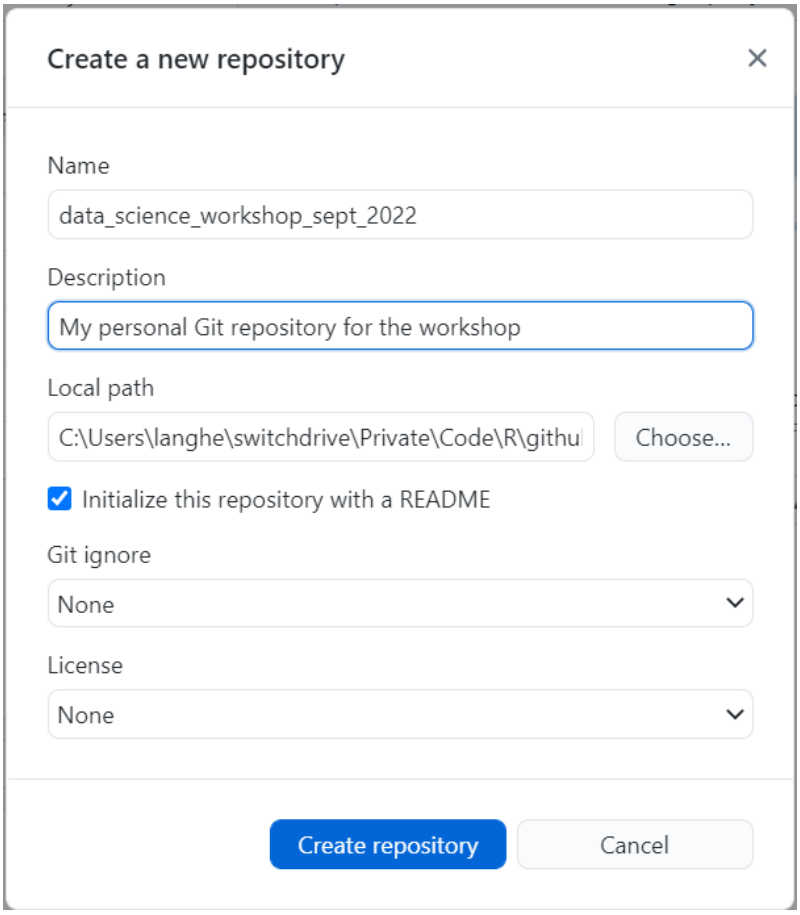
In GitHub Desktop, you can create a new repository by selecting **File** > **New repository**.



3.4.3 Step 3

1. Enter **data_science_workshop_sept_2022** as the name of your new repository.

2. Click **Choose...** to select the local directory in which your new repository will be created. Using Windows Explorer, navigate to the local repository of your choice.
3. Check **Initialize this repository with a README** to create a README file in your new repository. This is *optional* and you will be able to create a README file later if you wish to do so.
4. Click on **Create repository**.



3.5 To go further

To learn more about Git, we refer you to the resources listed in Section **??**.

4 Getting started with RStudio

4.1 Introduction

4.1.1 Overview

Please review the following sections for instructions on installation steps:

- Downloading R (Section **??**)
- Downloading and configuring RStudio (Section **??**)

4.1.2 Learning objectives

1. Familiarise with RStudio
2. Get set up with not storing the RStudio workspace

4.2 Orientation to the RStudio interface

Open RStudio

By default RStudio displays four rectangle panes.

> Tip
>
> If your RStudio displays only one left pane it is because you have no scripts open yet.

4.2.1 R Console Pane

The R Console, by default the left or lower-left pane in R Studio, is the home of the R "engine". This is where the commands are actually run and non-graphic outputs and error/warning messages appear. You can directly enter and run commands in the R Console, but realize that these commands are not saved as they are when running commands from a script.

> Tip
>
> This pane is similar to the Stata Command and the Results windows.

4.2.2 Source Pane

This pane, by default in the upper-left, is space to edit and run your scripts, including the RMarkdown outbreak and survey templates. This pane can also display datasets (data frames) for viewing.

> Tip
>
> This pane is similar to the Stata Do-file and Data Editor windows.

4.2.3 Environment Pane

This pane, by default the upper-right, is most often used to see brief summaries of objects in the R Environment in the current session. These objects could include imported, modified, or created datasets, parameters you have defined (e.g. a specific epi week for the analysis), or vectors or lists you have defined during analysis (e.g. names of regions). Click on the arrow next to a dataframe name to see its variables.

> Tip
>
> This pane is similar to the Stata Variables Manager window.

4.2.4 Lower-right pane

The lower-right pane includes several tabs:

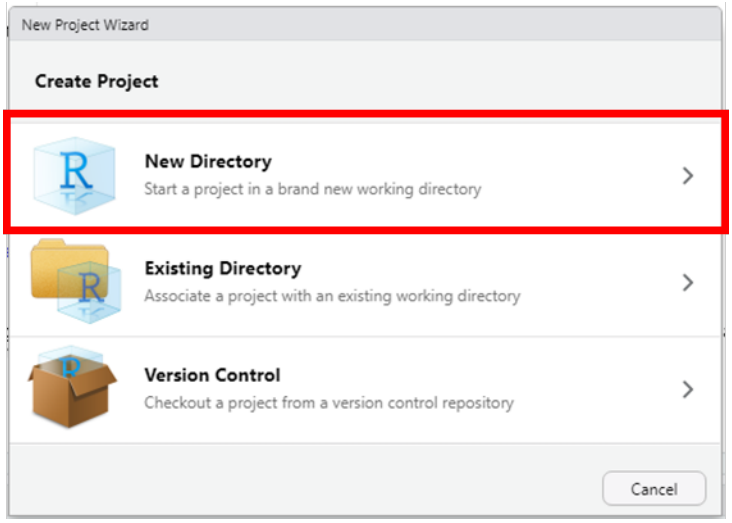
- Files (library of files)
- Plots (display of graphics including maps)
- Packages (available R packages including installation/update options)
- Help
- Viewer
- Presentation.

> Tip
>
> This pane contains the Stata equivalents of the Plots Manager and Project Manager windows.

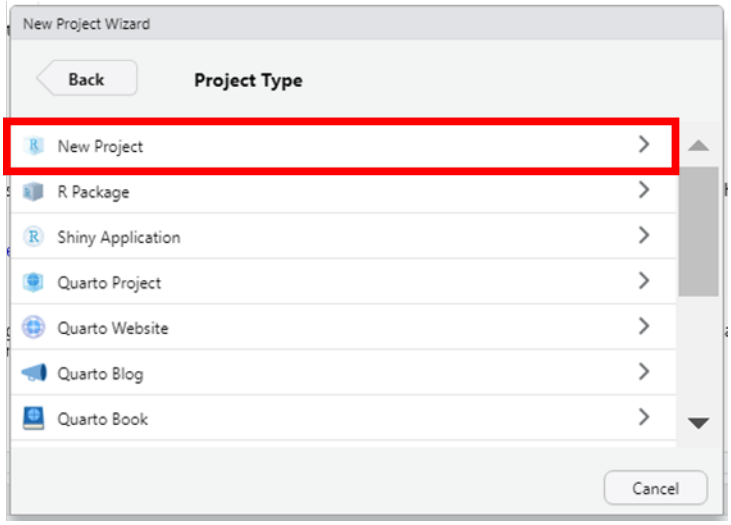
4.3 Open a new R project

In RStudio, you can create a new project by selecting **File** > **New Project...**

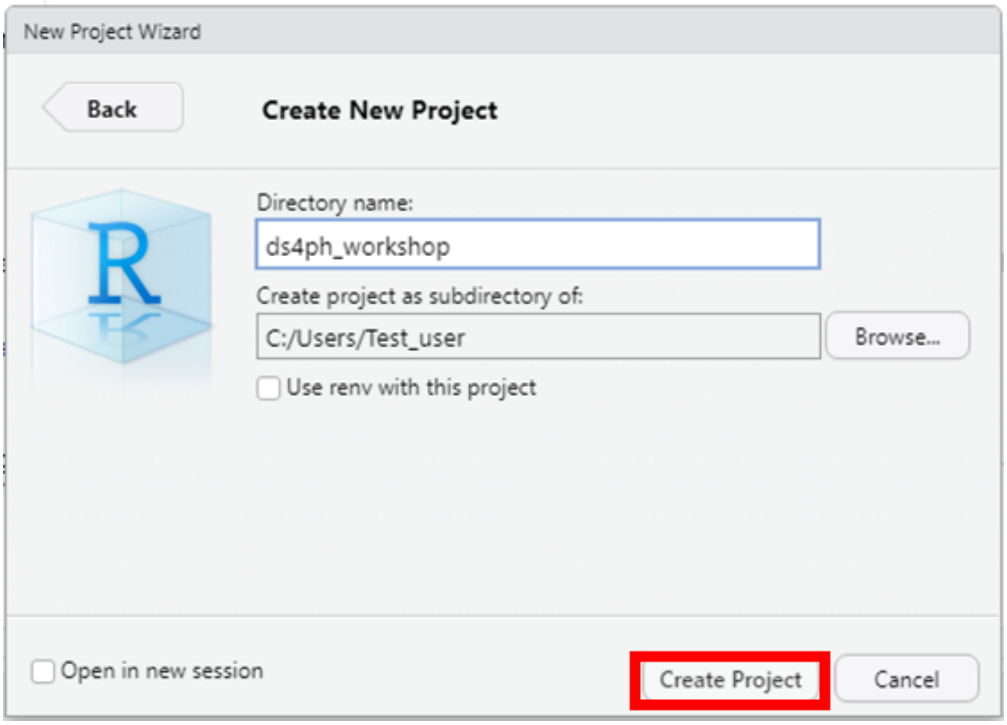
Select New Directory



Select New Project



Select a location to save the new R project (this creates a new folder).



Save all relevant data files into this new R project folder.

We will use this project for the duration of the workshop.

5 Dynamic documents

5.1 Introduction

5.1.1 Overview

The final product of a quantitative research is a report (e.g., scientific publications), i.e. a textual description of your research findings along with figures and tables resulting from your analysis. summary tables and figures. Based on this data, you discuss findings and give recommendations while using the data as evidence that backs up your discussion.

Imagine the following situations

1. you are informed that you were given the wrong data set just when you have finalised your article for submission to a journal. You are sent a new one and you are asked to run the same analysis with this new data set.
2. you realize that a mistake was made and need to re-examine the code, fix the error, and re-run the analysis 3. someone you are training wants to see the code and be able to reproduce the results to learn about your approach?

Situations like the ones just described are actually quite common for a data scientist.

It is actually possible to keep your data science projects organized with RStudio so that re-running an analysis and recreating reports is straightforward and can be done with minimal effort.Dynamic documents can be produced to update on a routine basis (e.g. daily surveillance reports) and/or run on subsets of data (e.g. reports for each jurisdiction).

5.1.2 Learning objectives

The goal of this section is to briefly discuss why we want to learn quarto, the benefits, and the barriers to using it.

- What is a dynamic report?
- What is Quarto?
- Think about why you want to use Quarto

5.2 Background to R Markdown

This is possible due to the fact that Quarto documents enable code and textual descriptions to be combined into the same document, and the figures and tables produced by the code are automatically added to the document.

Quarto is a tool that allows you integrate your code, text and figures in a single file in order to make high quality, reproducible reports. A paper published with an included quarto file and data sets can be reproduced by anyone with a computer. R Markdown integrates code and natural language in a way that is called "literate programming" (4).

To explain some of the concepts and packages involved:

5.2.1 Markdown

Markdown is a "language" that allows you to write a document using plain text, that can be converted to html and other formats. It is not specific to R.

Markdown files have a `md` extension.

5.2.2 R Markdown

It is a variation on markdown that is specific to R - it allows you to write a document using markdown to produce text and to embed R code and display their outputs. which was a variant of Markdown specifically designed to allow R code chunks to be included.

R Markdown is a widely-used tool for creating automated, reproducible, and share-worthy outputs, such as reports. It can generate static or interactive outputs, in Word, pdf, html, powerpoint, and other formats.

R Markdown files have `.Rmd` extension.

5.2.3 rmarkdown

It is the R package: This is used by R to render the .Rmd file into the desired output.

5.2.4 Quarto

Quarto is the successor to R Markdown. As a R Markdown document, a Quarto document intersperces code and text such that the script actually becomes your output document. You can create an entire formatted document, including narrative text (can be dynamic to change based on your data), tables, figures, bullets/numbers, bibliographies, etc.

Quatro uses a mark-up language similar to HyperText Markup Language (HTML) or LaTeX, in comparison to a "What You See Is What You Get" (WYSIWYG) language, such as Microsoft Word. This means that all the aspects are consistent, for instance, all top-level heading will look the same. But it means that we use symbols to designate how we would like certain aspects to appear. And it is only when we build the mark-up that we get to see what it looks like. A visual editor option can also be used which hides the need for the user to do this mark-up themselves.

Quarto is not tied to the R language.

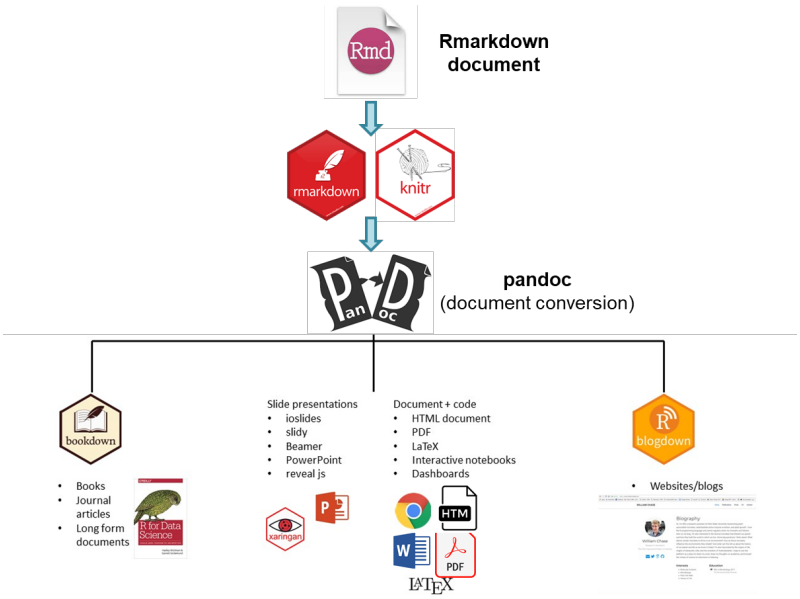
Quarto files have a `.Qmd` extension.

5.2.5 knitr

This R package will read the code chunks, execute it, and 'knit' it back into the document. This is how tables and graphs are included alongside the text.

5.2.6 Pandoc

Pandoc actually convert the output into word/pdf/powerpoint etc. It is a software separate from R but is installed automatically with RStudio.



The process that happens in the background involves feeding the .Rmd file to knitr, which executes the R code chunks and creates a new .md (markdown) file which includes the R code and its rendered output. The .md file is then processed by Pandoc to create the final product: a Microsoft Word document, HTML file, PowerPoint document, PDF, etc.

5.2.7 Discussion

Form small groups of 2-4 with your neighbours and discuss how you expect learning Quarto might benefit you.

5 minutes

5.3 References

- The Epidemiologist R Handbook (https://epirhandbook.com)

6 Getting started with Quarto

6.1 Introduction

6.1.1 Overview

Please review the following sections for instructions on installation steps:

- Downloading Quarto (Section **??**)
- Installing the rmarkdown package

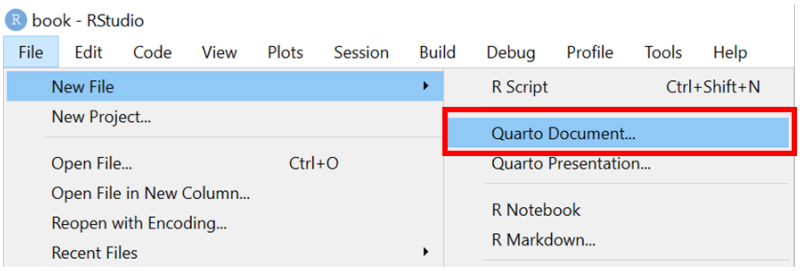
6.1.2 Learning objectives

1. Learn how to use Quarto

6.2 Create a new Quarto document

While it makes sense to use Quarto going forward, there are still a lot of resources written for and in R Markdown. For this reason we provide the R Markdown equivalents for this section in Appendix.

In RStudio, you can create a new Quarto document by selecting **File > New File > Quarto Document...**

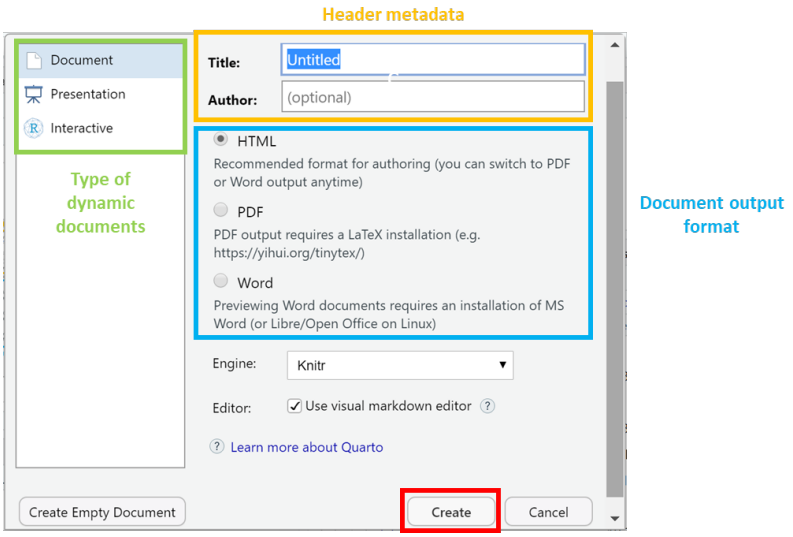


When you create a new Quarto document, RStudio tries to be helpful by allowing you to select a template which explains the different section of an R Markdown script. R Studio will enable you select options to pick from to generate a template Quarto document to start from.

The title and the author names are not important. If the output document type you want is not one of these, do not worry - you can just pick any one and change it later.

Let us select *HTML* to create an html document.

Click on create to open up a new Quarto (.Qmd) document.



6.3 Visual Editor

The RStudio Visual Editor is quite new and has features that improve your writing experience. Working in the Visual Editor feels a bit like working in a Google Doc.

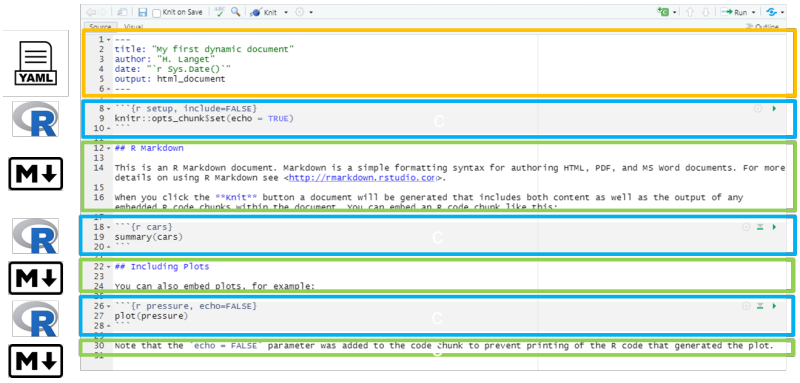
Here's an example showing the same file in the original Source Editor with content in markdown format and in the Visual Editor with content that looks more like it will appear in a live site. You can switch freely between these modes.

6.4 Quarto document structure

An R Markdown document can be edited in RStudio.

There are three basic components to a Quarto document, similar to the components of a R Markdown document:

- metadata (YAML header)
- text (markdown formatting)
- code (R code formatting)



6.4.1 YAML header

The very top of the document consists of a (YAML) header surrounded by — lines. Here you may want to edit the title of your document. The other settings in the header define the default document type produced (Microsoft Word) when the RMarkdown is "knit". the information intended to produce an html output.

6.4.2 Text

In WHITE background areas, any text will appear as regular text in the final report. Can have formatting such as headings, italics, bold, numbers, and bullets. See the second page of this RMarkdown cheatsheet for more detail. Can display parameters derived from your data via in-line code (such as epi week of the outbreak peak, as in the example above).

6.4.3 Code chunks

In gray background "code chunks", RMarkdown is running R commands. These commands perform data processing and cleaning steps, or could produce visual outputs in the document.

6.5 Quarto render

When you click the **Render** button a document will be generated that includes both content and the output of embedded code.

Code options

You can embed code like this:

```
1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).

6.6 Multilanguage

To write and execute code in Quarto, you will use **code chunks**.

As the number of programming languages used for scientific discourse is very broad, Quarto was developed to be multilingual, beginning with R, Python, Javascript, and Julia. building on

the RStudio (R) and Jupyter (Python, Julia) ecosystems which are very popular.

In this section, we will see how to use and mix R, Stata and Python within Quarto so that you can make the most out of it.

6.7 Create code chunks

> **Tip**
>
> Here are some tips for creating code chunks in RStudio
>
> - **Backticks**: use three backticks to start and end a code chunk.
> - **Toolbar icon**: you can also start a code chunk by clicking the appropriate icon in the toolbar.
> - **Keyboard shortcut**: for a quicker method, use the keyboard shortcut `Ctrl` + `Alt` + `I`

6.8 Write R code chunks

> **Note**
>
> A **data frame** is a two-dimensional array-like structure that contains rows and columns.

To store the content of the *iris* dataset into a data frame named `df`, you can use `=` (*equal*) but good practice recommend using `<-` (*back arrow*) to indicate the direction of your allocation.

````
```{r}
df <- iris
```
````

How to display data

```{r}
#| df-print: kable
library(dplyr)
```

```
Attache Paket: 'dplyr'
```

```
Die folgenden Objekte sind maskiert von 'package:stats':
```

```
    filter, lag
```

```
Die folgenden Objekte sind maskiert von 'package:base':
```

```
    intersect, setdiff, setequal, union
```

```{r}
#| df-print: kable
knitr::kable(head(df, 10))
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

The pipe operator

```{r}
#| df-print: kable
library(dplyr)
df %>%
  head(10) %>%
  knitr::kable()
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

You can cross-reference your table Table **??**

```{r}
#| label: tbl-iris
#| tbl-cap: Iris data set
#| df-print: kable
df %>%
  knitr::kable()
```

Table 6.3: Iris data set

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa
5.4	3.4	1.7	0.2	setosa
5.1	3.7	1.5	0.4	setosa
4.6	3.6	1.0	0.2	setosa
5.1	3.3	1.7	0.5	setosa
4.8	3.4	1.9	0.2	setosa
5.0	3.0	1.6	0.2	setosa
5.0	3.4	1.6	0.4	setosa
5.2	3.5	1.5	0.2	setosa
5.2	3.4	1.4	0.2	setosa
4.7	3.2	1.6	0.2	setosa
4.8	3.1	1.6	0.2	setosa
5.4	3.4	1.5	0.4	setosa
5.2	4.1	1.5	0.1	setosa
5.5	4.2	1.4	0.2	setosa
4.9	3.1	1.5	0.2	setosa
5.0	3.2	1.2	0.2	setosa
5.5	3.5	1.3	0.2	setosa
4.9	3.6	1.4	0.1	setosa
4.4	3.0	1.3	0.2	setosa
5.1	3.4	1.5	0.2	setosa
5.0	3.5	1.3	0.3	setosa
4.5	2.3	1.3	0.3	setosa
4.4	3.2	1.3	0.2	setosa
5.0	3.5	1.6	0.6	setosa

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.8	1.9	0.4	setosa
4.8	3.0	1.4	0.3	setosa
5.1	3.8	1.6	0.2	setosa
4.6	3.2	1.4	0.2	setosa
5.3	3.7	1.5	0.2	setosa
5.0	3.3	1.4	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
5.5	2.3	4.0	1.3	versicolor
6.5	2.8	4.6	1.5	versicolor
5.7	2.8	4.5	1.3	versicolor
6.3	3.3	4.7	1.6	versicolor
4.9	2.4	3.3	1.0	versicolor
6.6	2.9	4.6	1.3	versicolor
5.2	2.7	3.9	1.4	versicolor
5.0	2.0	3.5	1.0	versicolor
5.9	3.0	4.2	1.5	versicolor
6.0	2.2	4.0	1.0	versicolor
6.1	2.9	4.7	1.4	versicolor
5.6	2.9	3.6	1.3	versicolor
6.7	3.1	4.4	1.4	versicolor
5.6	3.0	4.5	1.5	versicolor
5.8	2.7	4.1	1.0	versicolor
6.2	2.2	4.5	1.5	versicolor
5.6	2.5	3.9	1.1	versicolor
5.9	3.2	4.8	1.8	versicolor
6.1	2.8	4.0	1.3	versicolor
6.3	2.5	4.9	1.5	versicolor
6.1	2.8	4.7	1.2	versicolor
6.4	2.9	4.3	1.3	versicolor
6.6	3.0	4.4	1.4	versicolor
6.8	2.8	4.8	1.4	versicolor
6.7	3.0	5.0	1.7	versicolor
6.0	2.9	4.5	1.5	versicolor
5.7	2.6	3.5	1.0	versicolor
5.5	2.4	3.8	1.1	versicolor
5.5	2.4	3.7	1.0	versicolor
5.8	2.7	3.9	1.2	versicolor

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
6.0	2.7	5.1	1.6	versicolor
5.4	3.0	4.5	1.5	versicolor
6.0	3.4	4.5	1.6	versicolor
6.7	3.1	4.7	1.5	versicolor
6.3	2.3	4.4	1.3	versicolor
5.6	3.0	4.1	1.3	versicolor
5.5	2.5	4.0	1.3	versicolor
5.5	2.6	4.4	1.2	versicolor
6.1	3.0	4.6	1.4	versicolor
5.8	2.6	4.0	1.2	versicolor
5.0	2.3	3.3	1.0	versicolor
5.6	2.7	4.2	1.3	versicolor
5.7	3.0	4.2	1.2	versicolor
5.7	2.9	4.2	1.3	versicolor
6.2	2.9	4.3	1.3	versicolor
5.1	2.5	3.0	1.1	versicolor
5.7	2.8	4.1	1.3	versicolor
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica
6.3	2.9	5.6	1.8	virginica
6.5	3.0	5.8	2.2	virginica
7.6	3.0	6.6	2.1	virginica
4.9	2.5	4.5	1.7	virginica
7.3	2.9	6.3	1.8	virginica
6.7	2.5	5.8	1.8	virginica
7.2	3.6	6.1	2.5	virginica
6.5	3.2	5.1	2.0	virginica
6.4	2.7	5.3	1.9	virginica
6.8	3.0	5.5	2.1	virginica
5.7	2.5	5.0	2.0	virginica
5.8	2.8	5.1	2.4	virginica
6.4	3.2	5.3	2.3	virginica
6.5	3.0	5.5	1.8	virginica
7.7	3.8	6.7	2.2	virginica
7.7	2.6	6.9	2.3	virginica
6.0	2.2	5.0	1.5	virginica
6.9	3.2	5.7	2.3	virginica
5.6	2.8	4.9	2.0	virginica

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
7.7	2.8	6.7	2.0	virginica
6.3	2.7	4.9	1.8	virginica
6.7	3.3	5.7	2.1	virginica
7.2	3.2	6.0	1.8	virginica
6.2	2.8	4.8	1.8	virginica
6.1	3.0	4.9	1.8	virginica
6.4	2.8	5.6	2.1	virginica
7.2	3.0	5.8	1.6	virginica
7.4	2.8	6.1	1.9	virginica
7.9	3.8	6.4	2.0	virginica
6.4	2.8	5.6	2.2	virginica
6.3	2.8	5.1	1.5	virginica
6.1	2.6	5.6	1.4	virginica
7.7	3.0	6.1	2.3	virginica
6.3	3.4	5.6	2.4	virginica
6.4	3.1	5.5	1.8	virginica
6.0	3.0	4.8	1.8	virginica
6.9	3.1	5.4	2.1	virginica
6.7	3.1	5.6	2.4	virginica
6.9	3.1	5.1	2.3	virginica
5.8	2.7	5.1	1.9	virginica
6.8	3.2	5.9	2.3	virginica
6.7	3.3	5.7	2.5	virginica
6.7	3.0	5.2	2.3	virginica
6.3	2.5	5.0	1.9	virginica
6.5	3.0	5.2	2.0	virginica
6.2	3.4	5.4	2.3	virginica
5.9	3.0	5.1	1.8	virginica

6.9 References

- The Epidemiologist R Handbook (https://epirhandbook.com)

7 Commit your changes with Git

7.1 Commit

Version control uses a *working copy* where you do your work.

> **Note**
>
> You can `update` your working copy to incorporate any new edits or versions that have been added to the repository since the last time you updated.

You make arbitrary edits to this copy, without affecting your teammates. When you are happy with your edits, you `commit` your changes to a *repository*.

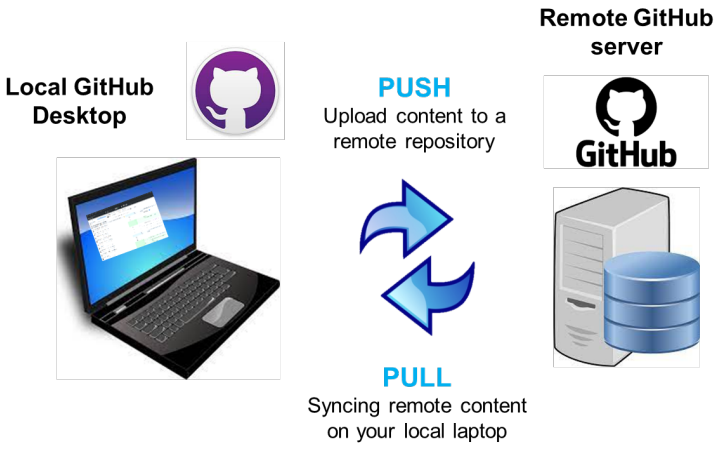
Snapshot of your entire repository at a specific time.

Over time, commits should tell a story of the history of your repository and how it came to be the way that it currently is.

Commits include lots of metadata in addition to the contents and message, like the author, timestamp, and more.

It also requires that you write something human-readable that will be a breadcrumb for you in the future. be easy to compare versions, and you can easily revert to previous versions.

7.2 Push and Pull



8 Transition from Stata to R

8.1 Execute Stata commands within R code chunks

Stata is not a language supported by Quarto. To use Stata within Quarto, you have to use the RStata R package, which is a simple R / Stata interface that enables you to:

- execute Stata commands (inline or from a `.do` file) from R;
- pass a data frame to Stata;
- return Stata outputs (including modified data frames) to R.

````
```{r}
if ( !require(RStata) ) {
  install.packages('RStata')
}
library(RStata)
```
````

8.1.1 Configure RStudio to execute Stata

8.1.1.1 Find your Stata binary path

The function `chooseStataBin` from the `RStata` library allows you to browse and set the path to your Stata binary executable.

````
```{r}
stata_bin_path <- RStata::chooseStataBin()
print(stata_bin_path)
```
````

```
```

When you run this code, you should normally get a path which is of the format: **“”C:\Program Files\Stata16\StataIC-64””**. Note that the `.exe` extension has been removed from this path. It is important that you keep the format as is, as otherwise your Stata engine will not be recognised.

8.1.1.2 Add your Stata binary path to your .Rprofile

The Stata binary path setting we just created is just for your current RStudio session and it will be lost once your RStudio is closed. To keep this setting each time you are using RStudio and avoid havin to reconfigure RStata each time you are restarting RStudio, let us add the Stata binary path as an option to `.Rprofile`, which is your user-specific R configuration file.

```
library(usethis)
```

`Usethis` is a workflow package: it automates repetitive tasks that arise during project setup and development, both for R packages and non-package projects.

The function `edit_r_profile` from the `usethis` library allows to open your configuration file `.Rprofile`.

````
```{r}
usethis::edit_r_profile("user")
```
````

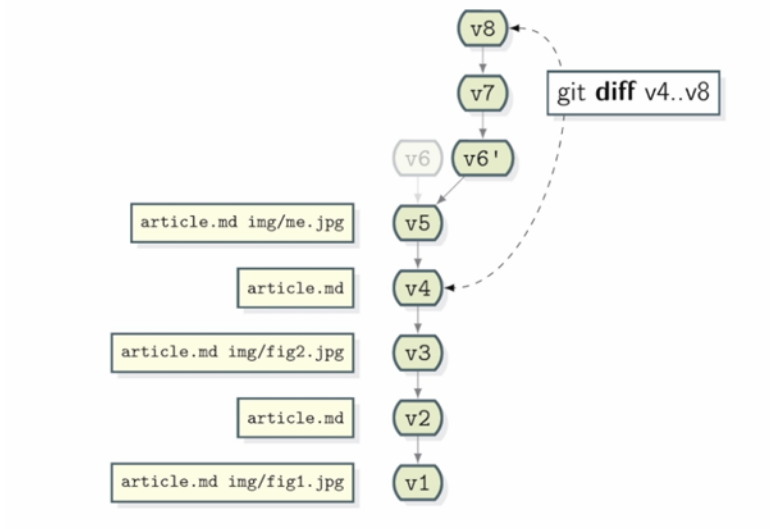
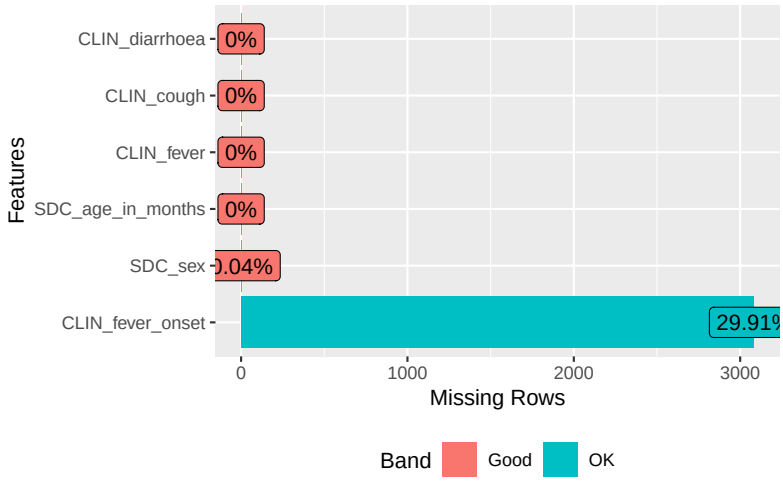
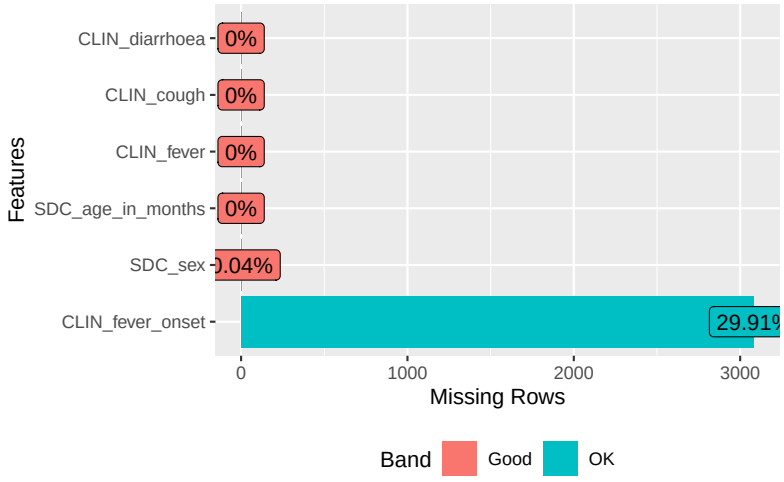
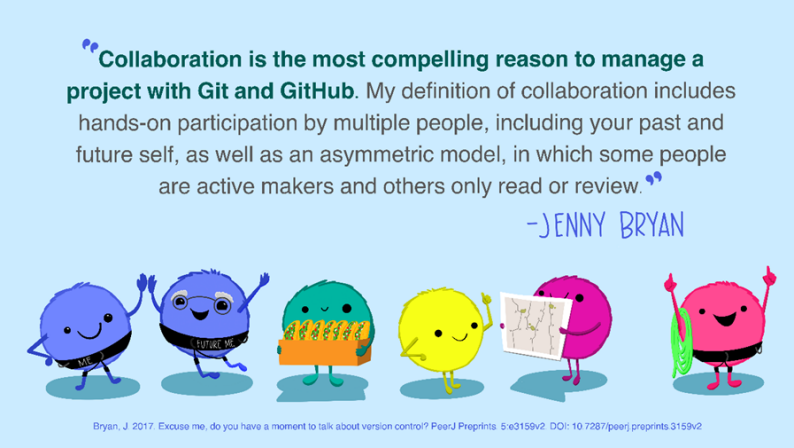
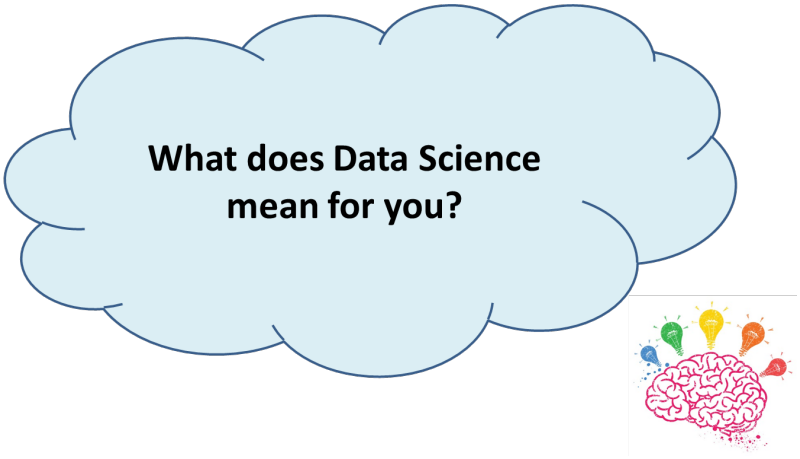
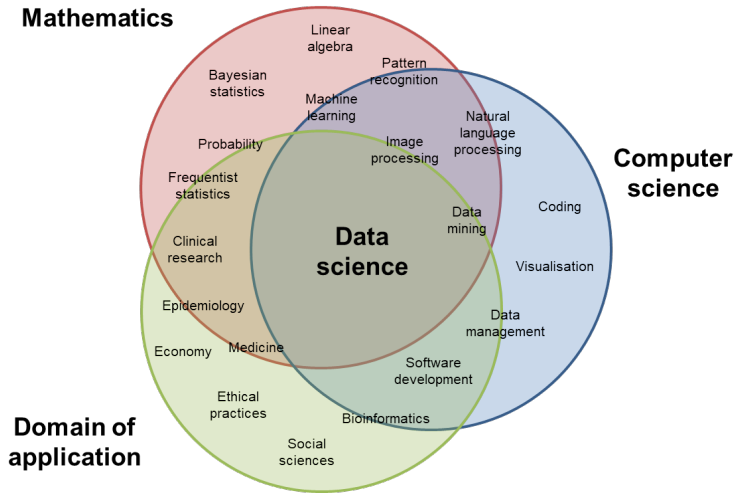
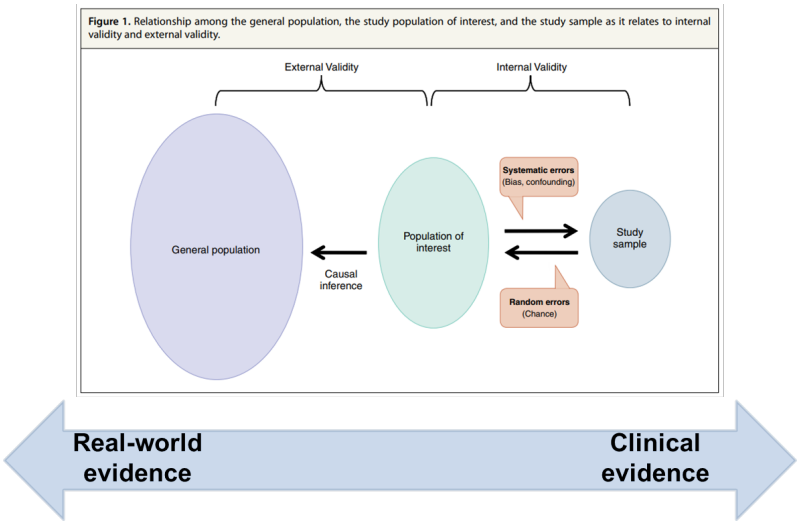
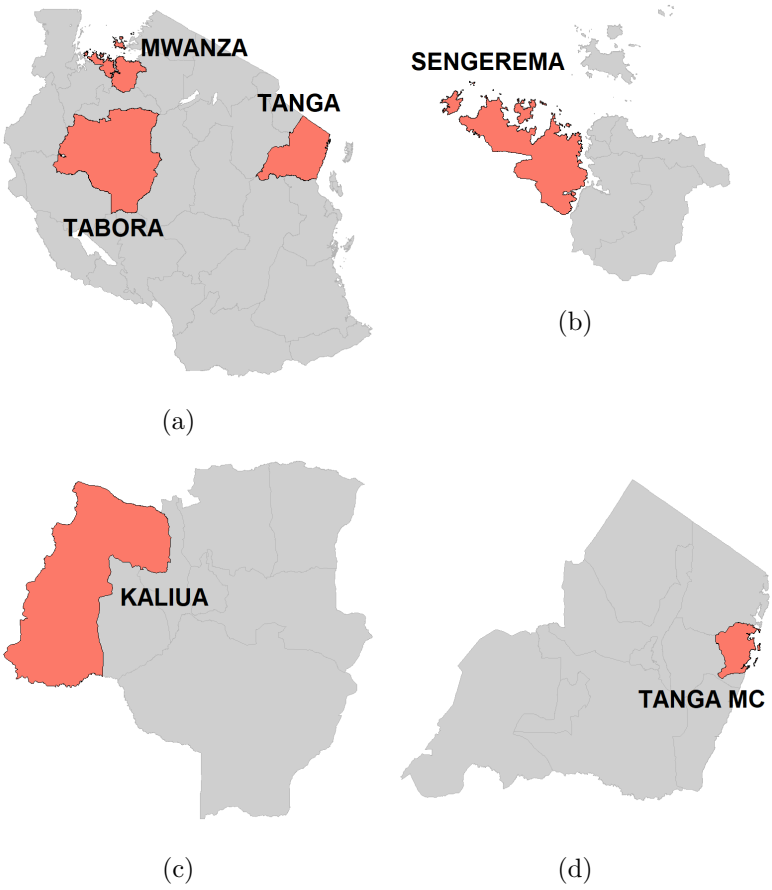
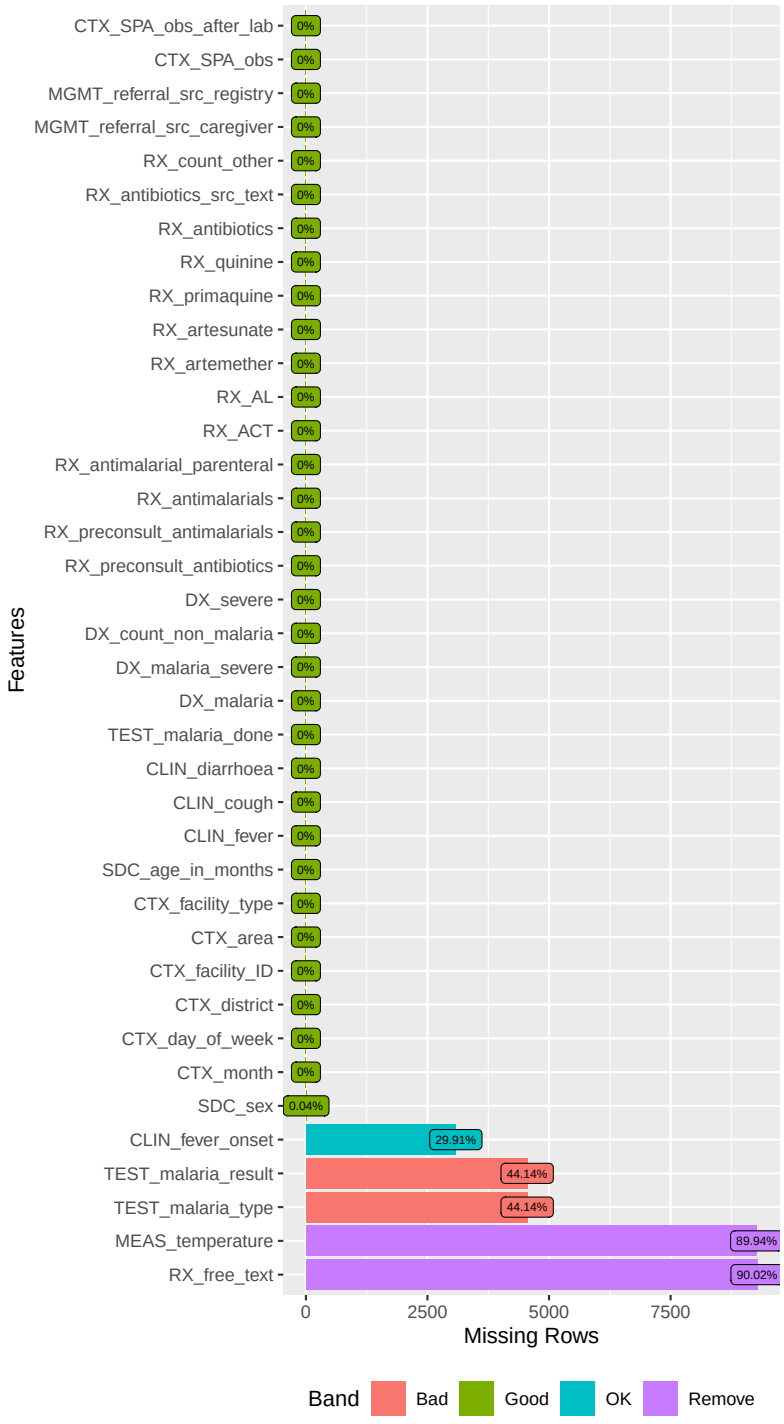
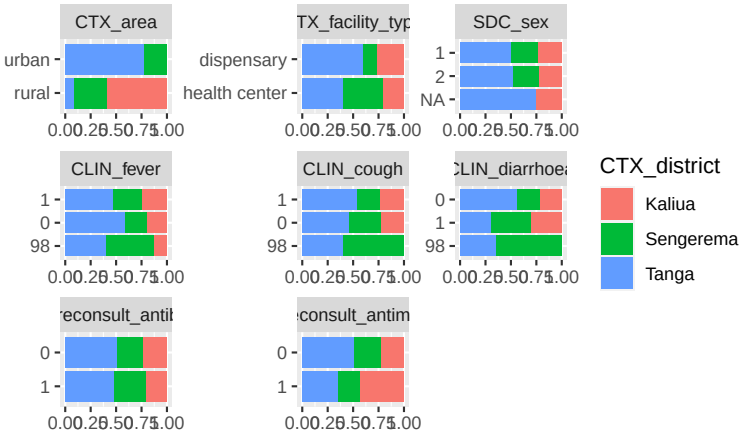
Add the following two lines in the Rprofile file that you have just opened:

````
```{r}
options("RStata.StataPath" = ...)
options("RStata.StataVersion" = ...)
```
````

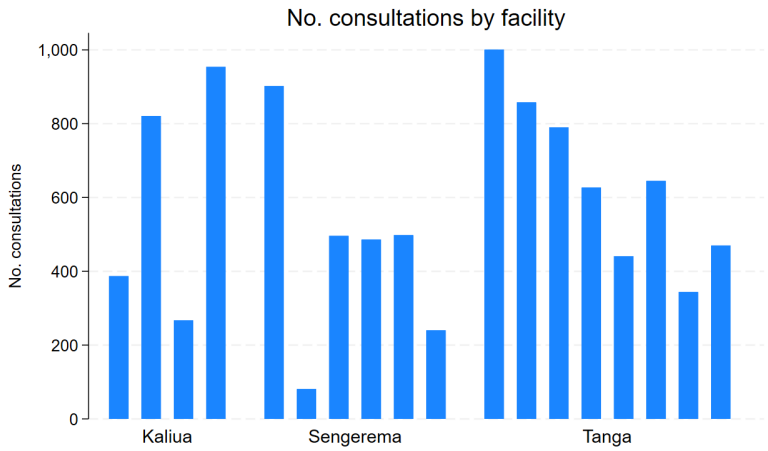
You need to indicate the path to your Stata binary executable in the `RStata.StataPath` option (e.g., “”C:\Program

Files\Stata16\StataIC-64"") and the version of your Stata (e.g., 16) in the `RStata.StataVersion` option.

after indicating the path and version your code should look as specified below

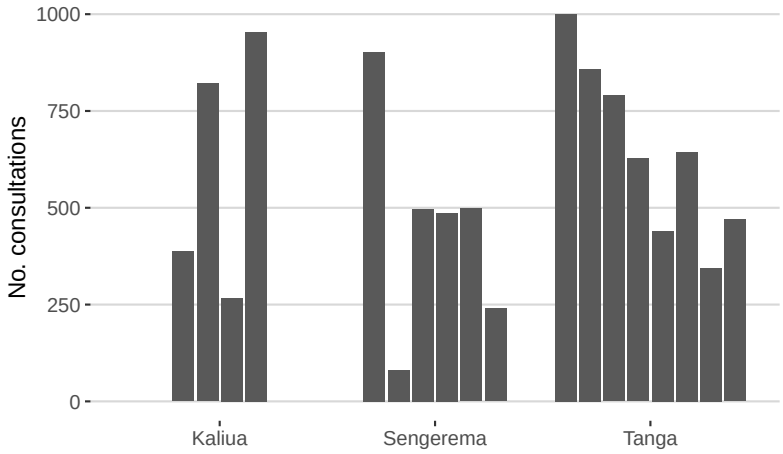
```
```{r}
options("RStata.StataPath" = "\"C:\\Program Files (x86)\\Stata15\\Stata-64\"")
options("RStata.StataVersion" = 15)
```
```

Once you are done, save and close your `.Rprofile`.

8.2 Python code chunks

> **Warning**
>
> This section is for (advanced) Python/R Markdown users only

You still need to import the R package `reticulate` if you want to use the Knitr engine and manipulate Python objects within R code chunks.

```
```{r}
library(reticulate)
```
```

8.2.1 Call R objects in Python code chunks

R objects can be manipulated in Python code chunks by referring to them as `r.`

```
```{r}
val <- 10
```
```

```
```{python}
val = 5
print(r.val)
```
```

```
10.0
```

8.2.2 Call Python objects in R code chunks

Python objects can be manipulated in R code chunks by referring to them as `py$`

```
```{r}
print(val)
print(py$val)
```
```

```
[1] 10
[1] 5
```

9 Project history with Git

9.1 Project history

The version control capabilities of Git permit us to keep track of changes we make to our code. We can also revert back to previous versions of files. Git also permits us to create branches in which we can test out ideas, then decide if we merge the new branch with the original.

In the simplest case, the database contains a linear history: each change is made after the previous one., but you can have a more complex history. We will see this later.



10 Import external data

10.1 Introduction

Most of the time you will want to generate *Quarto* documents using your own data. To this aim, you will have to import data from external sources such as files, URLs, or server data (e.g., ODK Central data). There is a dedicated importing function in R and Python for almost every data format. In this section we show you how to import Stata (`.dta`), Excel (`.xlsx`) and comma-separated values (CSV, `.csv`) data formats from files and from URLs, as well as how to import ODK data directly from an ODK Central server.

> **!** Important
>
> As you import data to further process / analyse them, you have to store the imported data in a data frame.

10.2 Import data from files

All Python and R functions only require as input the path where the file you want to import is stored. This path has to be passed as a sequence of characters (*character*) within double (" ") or single (' ') quotes.

The path can be either:

- relative to your Quarto document

```
```{r}
relative_path <- "./data/mydata.csv"
```
```

- absolute

```
```{r}
absolute_path <- "C:/Users/myuser/Documents/mydata.csv"
```
```

Functions in general have additional optional arguments.

10.2.1 Import Excel data

10.2.1.1 Exercise 1

Import the Excel data set **dataset1.xlsx** and store it into a data frame called **df1**.

> Tip
>
> - Stata: use the import excel Stata command with the stata function from the `RStata` package.
> - R: use the read.xlsx function from the `openxlsx` package.
> - Python: use the read_excel function from the `pandas` package.

```
```{r}
# Write your code here
```
```

10.2.1.2 Stata

Use the import excel Stata command with the stata function from the `RStata` package.

```
```{r}
#| df-print: kable
library(RStata)

df1_stata <- RStata::stata("import excel ./data/dataset1a.xlsx",
```
```

```
                        data.out = TRUE)
df1_stata_sub <- head(df1_stata, 5)
knitr::kable(df1_stata_sub)
```

```
. import excel ./data/dataset1a.xlsx
(7 vars, 10,309 obs)
```

A	B	C	D	E	F	G
child_ID	SDC_sex	SDC_age_in_months	CLIN_fever	CLIN_onset	CLIN_cough	CLIN_diarrhoea
1	2	10	0		1	1
2	2	6	0		0	1
3	1	6	0		0	0
4	1	11	1	3	1	1

10.2.1.3 R

Use the read.xlsx function from the `openxlsx` package.

```
```{r}
library(openxlsx)

df1_r <- openxlsx::read.xlsx("./data/dataset1a.xlsx")
df1_r_sub <- head(df1_r, 5)
knitr::kable(df1_r_sub)
```
```

child_ID	SDC_sex	SDC_age_in_months	CLIN_fever	CLIN_onset	CLIN_cough	CLIN_diarrhoea
1	2	10	0	NA	1	1
2	2	6	0	NA	0	1
3	1	6	0	NA	0	0
4	1	11	1	3	1	1
5	2	21	1	2	0	0

10.2.1.4 Python

Use the read_excel function from the `pandas` package.

```
```{python}
import pandas

df1_python = pandas.read_excel('./data/dataset1a.xlsx')
df1_python_sub = df1_python.head(5)
```
```

```
C:\Users\langhe\AppData\Local\R\win-library\4.3\reticulate\python\rpytools\loader.py:117: UserW
  return _find_and_load(name, import_)
```

```
library(reticulate)
```

```
Warning: Paket 'reticulate' wurde unter R Version 4.3.3 erstellt
```

```
knitr::kable(py$df1_python_sub)
```

child_ID	SDC_sex	SDC_age_in_months	CLIN_fever	CLIN_cough	CLIN_diarrhoea
1	2	10	0	NaN	1	1
2	2	6	0	NaN	0	1
3	1	6	0	NaN	0	0
4	1	11	1	3	1	1
5	2	21	1	2	0	0

10.2.2 Import CSV data

10.2.2.1 Exercise 2

Read the CSV data set **dataset1.csv** and store it into a data frame called **df2**.

Tip

- Stata: use the import delimited Stata command with the stata function from the `RStata` package.
- R: use the read.csv function from the `haven` package.
- Python: use the read_csv function from the `pandas` package.

```
```{r}
# Write your code here
```
```

10.2.2.2 Stata

```
```{r}
library(RStata)

df2 <- RStata::stata("import delimited ./data/dataset1b.csv",
                     data.out = TRUE)
```
```

```
. import delimited ./data/dataset1b.csv
(encoding automatically selected: ISO-8859-1)
(7 vars, 10,308 obs)
```

10.2.2.3 R

```
```{r}
df2_r <- read.csv("./data/dataset1b.csv")
df2_r_sub <- head(df2_r, 5)
knitr::kable(df2_r_sub)
```
```

child_ID	SDC_sex	SDC_age_in_months	CLIN_fever	CLIN_onset_cough	CLIN_diarrhoea
1	2	10	0	NA	1	1
2	2	6	0	NA	0	1

child_ID	SDC_sex	SDC_age_in_months	CLIN_fever	CLIN_cough	CLIN_diarrhoea
3	1	6	NA	0	0
4	1	11	3	1	1
5	2	21	2	0	0

10.2.2.4 Python

Use the read_csv function from the `pandas` package.

```
```{python}
import pandas

df2 = pandas.read_csv('./data/dataset1b.csv')
```
```

10.2.3 Import Stata data

10.2.3.1 Exercise 3

Read the Stata data set **dataset1.dta** and store it into a data frame called **df3**.

> Tip
>
> - Stata: use the `use` Stata command with the stata function from the `RStata` package.
> - R: use the read_dta function from the `haven` package. This package supports SAS, STATA and SPSS software.
> - Python: use the read_stata function from the `pandas` package.

```
```{r}
# Write your code here
```
```

10.2.3.2 Stata

```
```{r}
library(RStata)

df3 <- RStata::stata("use ./data/dataset1c.dta",
                     data.out = TRUE)
```
```

```
. use ./data/dataset1c.dta
```

10.2.3.3 R

```
```{r}
library(haven)

df3 <- haven::read_dta("./data/dataset1c.dta")
```
```

10.2.3.4 Python

Use the read_stata function from the `pandas` package.

```
```{python}
import pandas

df3 = pandas.read_stata('./data/dataset1c.dta')
```
```

10.3 Import data from URLs

All functions can accept URLs as well instead of the path to a specific file.

10.3.0.1 Exercise 4

Import the CSV data set that contains a comprehensive spatial inventory of 98,745 public health facilities in Sub Saharan Africa directly from the following url and store it into a data frame called **df4**.

To learn more about how this data set was assembled, please refer to (5)

> Tip
>
> - Stata: use the import excel Stata command with the `stata` function from the `RStata` package.
> - R: use the read.xlsx function from the `openxlsx` package.
> - Python: use the read_excel function from the `pandas` package. In the latest version of pandas (0.19.2) you can directly pass the url

```
```{r}
# Write your code here
```
```

10.3.0.2 Stata

Here, because quotes are already used for the Stata command, you need to use the other type of quotes for indicating the URL.

```
```{r}
library(RStata)

#df2 <- RStata::stata('import delimited "https://open.africa/dataset/d7335980-29d5-476c-bf7a
                     # data.out = TRUE)
```
```

10.3.0.3 R

Use the read.xlsx function from the `openxlsx` package.

````
```{r}
#csv_url <- "https://open.africa/dataset/d7335980-29d5-476c-bf7a-feb4e22cf631/resource/e2432
#df4 <- read.csv(csv_url)
```
````

10.3.0.4 Python

Use the read_excel function from the `pandas` package.

````
```{python}
import pandas

#csv_url = "https://open.africa/dataset/d7335980-29d5-476c-bf7a-feb4e22cf631/resource/e2432e
#df1 = pandas.read_csv(csv_url)
```
````

11 Manipulate data

11.1 Introduction

> **!** Important
>
> Remember that with Quarto you can store multiple data sets in memory (stored in different data frames `df1`, `df2`, `df3`, etc) and work in parallel on all these data sets.

11.2 Import data

11.2.0.1 Exercise 1

Import **dataset1.xlsx** using Stata and store it in `df1`

````
```{r}
# Write your code here
```
````

11.2.0.2 Solution

````
```{r}
df1 <- openxlsx::read.xlsx("./data/dataset1a.xlsx")
```
````

11.3 Piping

Package `dplyr`

Pipes are a powerful tool for clearly expressing a sequence of multiple operations. So far, you've been using them without knowing how they work, or what the alternatives are. Now, in this chapter, it's time to explore the pipe in more detail. You'll learn the alternatives to the pipe, when you shouldn't use the pipe, and some useful related tools.

11.4 Structure of the data

11.4.1 Inspect the structure of the data

```{r}
library(skimr)
```

```{r}
df1 <- df1 %>%
  tibble::remove_rownames() %>%
  tibble::column_to_rownames(var="child_ID")
```

```{r}
df1 %>%
  skimr::skim()
```

Table 11.1: Data summary

Name	Piped data
Number of rows	10308
Number of columns	6
_____	
Column type frequency:	
numeric	6

Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
SDC_sex	4	1.0	1.49	0.50	1	1	1	2	2	
SDC_age_in_months	0	1.0	18.75	14.90	0	7	15	27	59	
CLIN_fever	0	1.0	0.84	3.74	0	0	1	1	98	
CLIN_fever_onset	3083	0.7	2.50	1.93	0	1	2	3	14	
CLIN_cough	0	1.0	0.69	3.75	0	0	1	1	98	
CLIN_diarrhoea	0	1.0	0.41	4.32	0	0	0	0	98	

A categorical variable is:

a) a variable with only two different possible values
b) a variable with continuous numerical values
c) a variable with a finite set of possible values

Select a single answer

11.4.2 Convert

Function mutate across

11.4.3 Add new columns

Function mutate

11.5 Descriptive statistics

Package `gtsummary`

````
```{r}
install.packages("gtsummary")
```
````

```{r}
library(gtsummary)
```

```{r}
df1 %>%
  gtsummary::tbl_summary()
```

Characteristic	N = 10,308
SDC_sex	
1	5,229 (51%)
2	5,075 (49%)
Unknown	4
SDC_age_in_months	15 (7, 27)
CLIN_fever	
0	3,068 (30%)
1	7,225 (70%)
98	15 (0.1%)
CLIN_fever_onset	2.00 (1.00, 3.00)
Unknown	3,083
CLIN_cough	
0	4,658 (45%)
1	5,635 (55%)
98	15 (0.1%)
CLIN_diarrhoea	
0	7,982 (77%)
1	2,306 (22%)
98	20 (0.2%)

11.6 Filter data

11.7 Concatenate data

cbind rbind

11.8 Visualise data

11.9 Plot

Package `ggplot2`

```
```{r}
install.packages("ggplot2")
```
```

11.10 Explore data

Package `DataExplorer`

```
```{r}
install.packages("DataExplorer")
```
```

```
library(DataExplorer)
```

```
DataExplorer::plot_missing(df1)
```



```r
DataExplorer::plot_missing(df1)
```



11.11 Manipulate Python and R data

11.11.0.1 Exercise 1

1. Import **dataset1.xlsx** using Stata and store it in `df1`

````
```{r}
# Write your code here
```
````

2. Import **dataset1.csv** using Python and store it in `df2`

````
```{python}
# Write your code here
```
````

3. Compare `df1`and `df2`. Can you indicate what variable has been modified in **dataset1** between df1 and df2?

Tip

Use the R function **comparedf**

```
```{r}
# Write your code here
```
```

11.11.0.2 Solution

1. Import **dataset1.xlsx** using Stata and store it in `df1`

```
```{r}
library(RStata)
df1 <- RStata::stata("import excel ./data/dataset1a.xlsx",
                     data.out = TRUE)
```
```

```
. import excel ./data/dataset1a.xlsx
(7 vars, 10,309 obs)
```

2. Import **dataset1.csv** using Python and store it in `df2`

```
```{python}
import pandas as pd
df2 = pd.read_csv('./data/dataset1b.csv')
```
```

3. Compare `df1`and `df2`.

```
```{r}
library(reticulate)
arsenal::comparedf(df1, py$df2)
```
```

```
Compare Object

Function Call:
```

```
arsenal::comparedf(x = df1, y = py$df2)

Shared: 0 non-by variables and 10308 observations.
Not shared: 14 variables and 1 observations.

Differences found in 0/0 variables compared.
0 variables compared have non-identical attributes.
```

12 Share code and collaborate with Git

12.1 Introduction

12.1.1 Overview

12.1.2 Learning objectives

12.2 Sharing your code with Git

Even if we do not take advantage of the advanced and powerful version control functionality, we can still use Git and GitHub to share our code.

12.3 Access a remote GitHub repository

12.4 Collaborating with Git

Once you set up a central repository, you can have multiple people make changes to code and keep versions synced. GitHub provides a free service for centralized repositories. GitHub also has a special utility, called a pull request, that can be used by anybody to suggest changes to your code. You can easily either accept or deny the request.


"Collaboration is the most compelling reason to manage a project with Git and GitHub. My definition of collaboration includes hands-on participation by multiple people, including your past and future self, as well as an asymmetric model, in which some people are active makers and others only read or review."
-JENNY BRYAN
ME
FUTURE ME
Bryan, J. 2017. Excuse me, do you have a moment to talk about version control? PeerJ Preprints. 5:e3159v2. DOI: 10.7287/peerj.preprints.3159v2

Part II

——— DAY 2

———

13 Data Science for Public Health

13.1 What is Data Science

13.2 Data Science

13.2.1 Activity 1

3 minutes: You have probably already heard about data science as this is a main Data Science is today one of the main buzzwords, but what does **Data Science** mean for you?

13.2.2 Quote 1

The statistics profession faces a choice in its future research between continuing concentration on traditional topics – based

largely on data analysis supported by mathematical statistics – and a broader viewpoint – based on an inclusive concept of **learning from data**. The latter course present severe challenges as well as exciting opportunities. The former risks seeing statistics become increasingly marginal... (6)

John Chambers (PhD in Statistics) is the creator of the S programming language, and a core member of the R programming language project.

13.2.3 Quote 2

Data science is the science of **learning from data**; it studies the methods involved in the analysis and processing of data and proposes technology to **improve methods in an evidence-based manner**. The scope and impact of this science will expand enormously in coming decades as scientific data and data about science itself become ubiquitously available. (7).

David Donoho is a Professor of Statistics at Stanford University.

13.2.4 Quote 1

- Growing field with computational power and data generation
- Set of collective evidence-based processes, theories, concepts, tools and technologies
- Extraction of valuable knowledge and information from data (learning from data)

Qualitative thinking currently does not receive nearly as much attention as quantitative thinking in data science.

13.2.5 Discussion

What does Data Science mean for you?



13.2.6 From literature

Had you thought of any of those words? Word cloud generated by MonkeyLearn from (8)

13.2.7 Where do you see yourself on this Venn diagram



13.3 What do experts say about Data Science?

Public health data science is the study of formulating and rigorously answering questions in order to advance health and well-being using a data-centric process that emphasizes clarity, reproducibility, effective communication, and ethical practices. [...] Data science implies a perspective that is shaped by [...] interdisciplinary work. (8).

13.4 A tentative definition

NIH defines data science as "the interdisciplinary field of inquiry in which quantitative and analytical approaches, processes, and systems are developed and used to extract knowledge and insights from increasingly large and/or complex sets of data"

In this book, we define data science as

Data science is an interdisciplinary collaborative field that tries to answer rigorously formulated public health questions by relying on a set of collective evidence-based processes, theories, concepts, tools and technologies that enable the extraction of valuable knowledge and information from data to generate actionable insights that effectively communicated to decision-makers can advance health and well-being.

- Interdisciplinary collaborative field
- Rigorous public health question formulation
- Set of collective evidence-based processes, theories, concepts, tools and technologies
- Extraction of valuable knowledge and information from data
- Generation of actionable insights
- Effective communication to decision-makers

13.5 Uncertainty

There are many reasons why a data analysis might still leave us in a position of uncertainty.

Our data consists of a sample, and we want to generalize from facts about that sample to facts about the wider population from which it was drawn. This process of generalizing from a sample to a population is inherently uncertain, because we haven't sampled everyone.



Our data come from a randomized experiment. Historically, the tools of statistical inference have been designed to address concerns just like this.

We want to use our data to make a prediction about the future, and we expect the future to be similar to the past.

Our observations are subject to measurement or reporting error. Most measurements are subject to at least some kind of error, and sometimes those errors are large enough to matter.

Our data arise from an intrinsically random or variable process.

14 Malaria case study

14.1 Introduction

14.1.1 Overview

These pages will demonstrate how to use Quarto to data from Tanzania.

14.1.2 Learning objectives

- Apply what you have learnt on Day 1 on real data

14.2 Getting started

14.2.1 Access the Quarto template

Download the Quarto template used for this case study (add link) using GitHub.

Please review previous sections on Quarto, data import and manipulation.

14.2.2 Install packages

````
```{r}
install.packages("ggplot2")
install.packages("ggthemes")
install.packages("networkD3")
install.packages("apyramid")
```
````

```{r}
library(openxlsx)
library(dplyr)
library(skimr)
library(gtsummary)
library(finalfit)
library(ggplot2)
library(ggthemes)
library(networkD3) # For alluvial/Sankey diagrams
library(tidyverse)
```

14.2.3 Dataset description

We will be using data and examples from a real consultation data which occurred in Tanzania between **2021-07-29** and **2021-12-17** within the Integrated Management of Childhood Illness (TIMCI) project.

> **!** Important
>
> Data are made available by the Ifakara Health Institute (IHI) for training purposes only. Please note, that some data has been adapted in order to best achieve training objectives. No personally indentifiable information have been kept in this dataset.

Information about the consultations of **10,308 children** [**1 day - 59 months**] from **18 facilities** (dispensaries and health centres) in **Kaliua** District, **Sengerema** District and **Tanga** District, Tanzania.

14.2.4 Data collection

Data were collected using *ODK* (ODK Collect, ODK Central) between **2021-07-29** and **2021-12-17**. Research assistants recorded the following information from different sources.



(a) (b)

(c) (d)

Figure 14.1: Study area

Table 14.1: Types and sources of information

Information	Prefix	Source
Context	CTX	Metadata
Sociodemographics	SDC	Caregiver
Clinical presentation	CLIN	Caregiver
Laboratory investigations	TEST	Child booklet or facility MTUHA book
Diagnoses	DX	Child booklet or facility MTUHA book
Treatments	RX	
• before consultation		Caregiver
• as a result of the consultation		Child booklet or facility MTUHA book
Referral advice	MGMT	
		Caregiver
		Child booklet or facility MTUHA book

14.2.5 Data preparation

Data cleaning and data de-identification

Personally identifiable information (PII) were removed.

14.3 Population characteristics

14.3.1 Codebook

Variable	Coding
SDC_age_in_month	
SDC_sex	1: male
	2: female
	98: unknown
CLIN_fever	0: no
	1: yes
	98: not sure

Variable	Coding
CLIN_fever_onset	
CLIN_cough	0: no
	1: yes
	98: not sure
CLIN_diarrhoea	0: no
	1: yes
	98: not sure
RX_preconsult_antibiotics	
RX_preconsult_antimalarials	
CONSULT_district	Kaliua
	Sengerema
	Tanga
CONSULT_area	urban
	rural
CONSULT_facility_type	dispensary
	health centre

14.3.1.1 Exercise 1

How many variables are numerical? How many features are categorical?

> **Note**
>
> A **numerical variable** is a quantity represented by a real or integer number.
> A **categorical variable** has discrete values, typically represented by string labels (but not only) taken from a finite list of possible choices.

For instance, the variable native-country in our dataset is a categorical variable because it encodes the data using a finite list of possible countries (along with the ? symbol when this information is missing)

15 Malaria case study - Part 1

15.1 Introduction

15.1.1 Overview

These pages will demonstrate how to use Quarto to data from Tanzania.

15.1.2 Learning objectives

- Apply what you have learnt on Day 1 on real data

15.2 Getting started

15.2.1 Access the Quarto template

Download the Quarto template used for this case study (add link) using GitHub.

Please review previous sections on Quarto, data import and manipulation.

15.2.2 Install packages

```
```{r}
install.packages("ggplot2")
install.packages("ggthemes")
install.packages("networkD3")
```
```

```
install.packages("apyramid")
```

```{r}
library(openxlsx)
library(tidyverse)
library(skimr)
library(DataExplorer)
library(gtsummary)
library(finalfit)
library(ggplot2)
library(ggthemes)
library(networkD3) # For alluvial/Sankey diagrams
```

15.2.3 Import the data

15.2.3.1 Exercise 1

Import the dataset and store it into a dataframe called **df**. Display the first 5 rows and columns `child_ID`, `CTX_month`, `CTX_district`, `SDC_age_in_months`

> **Tip**
>
> Refer to Section **??**

```{r}
# Write your code here
```

15.2.3.2 Stata

```{r}
stata_df <- haven::read_dta("./data/dataset2.dta")
```

```{r}
#| df-print: kable
stata_df %>%
  head(5) %>%
  dplyr::select(child_ID,
                CTX_month,
                CTX_district,
                SDC_age_in_months) %>%
  knitr::kable()
```

child_ID	CTX_month	CTX_district	SDC_age_in_months
1	0	1	10
2	0	1	6
3	0	1	6
4	0	1	11
5	1	1	21

15.2.3.3 R

```{r}
df <- openxlsx::read.xlsx("./data/dataset2.xlsx")
```

```{r}
#| df-print: kable
df %>%
  head(5) %>%
  dplyr::select(child_ID,
                CTX_month,
                CTX_district,
                SDC_age_in_months) %>%
  knitr::kable()
```

child_ID	CTX_month	CTX_district	SDC_age_in_months
1	7	Kaliua	10
2	7	Kaliua	6
3	7	Kaliua	6
4	7	Kaliua	11
5	8	Kaliua	21

15.3 Population characteristics

15.3.1 Codebook

Variable	Coding
SDC_age_in_months	
SDC_sex	1: male
	2: female
	98: unknown
CLIN_fever	0: no
	1: yes
	98: not sure
CLIN_fever_onset	
CLIN_cough	0: no
	1: yes
	98: not sure
CLIN_diarrhoea	0: no
	1: yes
	98: not sure
RX_preconsult_antibiotics	
RX_preconsult_antimalarials	
CTX_district	Kaliua
	Sengerema
	Tanga
CTX_area	Urban
	rural
CTX_facility_type	Dispensary
	Health centre

15.3.2 Structure of the data

15.3.2.1 Exercise 2

Examine the structure of the data, including variable names, labels.

> Tip
>
> - Stata: use the codebook command
> - R: use the skim function from the `skimr` package

````
```{r}
# Write your code here
```
````

15.3.2.2 Stata

````
```{r}
RStata::stata("codebook SDC_age_in_months SDC_sex CLIN_fever CLIN_fever_onset CLIN_diarrhoea
              data.in = stata_df)
```
````

```
. codebook SDC_age_in_months SDC_sex CLIN_fever CLIN_fever_onset CLIN_diarrhoea
>  CLIN_cough RX_preconsult_antibiotics RX_preconsult_antimalarials CTX_distric
> t CTX_area CTX_facility_type

-------------------------------------------------------------------------------
SDC_age_in_months                                           SDC_age_in_months
-------------------------------------------------------------------------------

                  Type: Numeric (double)

                 Range: [0,59]                        Units: 1
         Unique values: 60                       Missing .: 0/10,308

                  Mean: 18.7498
             Std. dev.: 14.8998
```

```
          Percentiles:     10%       25%       50%       75%       90%
                             3         7        15        27        43

-------------------------------------------------------------------------------
SDC_sex                                                             SDC_sex
-------------------------------------------------------------------------------

                  Type: Numeric (double)

                 Range: [0,1]                         Units: 1
         Unique values: 2                         Missing .: 4/10,308

            Tabulation: Freq.  Value
                        5,229  0
                        5,075  1
                            4  .

-------------------------------------------------------------------------------
CLIN_fever                                                       CLIN_fever
-------------------------------------------------------------------------------

                  Type: Numeric (double)

                 Range: [0,2]                         Units: 1
         Unique values: 3                         Missing .: 0/10,308

            Tabulation: Freq.  Value
                        3,068  0
                        7,225  1
                           15  2

-------------------------------------------------------------------------------
CLIN_fever_onset                                           CLIN_fever_onset
-------------------------------------------------------------------------------

                  Type: Numeric (double)

                 Range: [0,14]                        Units: 1
         Unique values: 15                        Missing .: 3,083/10,308
```

```
                  Mean: 2.50145
             Std. dev.: 1.93099

           Percentiles:     10%       25%       50%       75%       90%
                              1         1         2         3         4
```

CLIN_diarrhoea CLIN_diarrhoea

```
                  Type: Numeric (double)

                 Range: [0,2]                         Units: 1
          Unique values: 3                        Missing .: 0/10,308

             Tabulation: Freq.  Value
                         7,982  0
                         2,306  1
                            20  2
```

CLIN_cough CLIN_cough

```
                  Type: Numeric (double)

                 Range: [0,2]                         Units: 1
          Unique values: 3                        Missing .: 0/10,308

             Tabulation: Freq.  Value
                         4,658  0
                         5,635  1
                            15  2
```

RX_preconsult_antibiotics RX_preconsult_antibiotics

```
                  Type: Numeric (double)

                 Range: [0,1]                         Units: 1
```

```
         Unique values: 2                        Missing .: 0/10,308

            Tabulation: Freq.  Value
                        8,573  0
                        1,735  1

------------------------------------------------------------------------------
RX_preconsult_antimalarials                       RX_preconsult_antimalarials
------------------------------------------------------------------------------

                  Type: Numeric (double)

                 Range: [0,1]                         Units: 1
         Unique values: 2                        Missing .: 0/10,308

            Tabulation: Freq.  Value
                        9,866  0
                          442  1

------------------------------------------------------------------------------
CTX_district                                                     CTX_district
------------------------------------------------------------------------------

                  Type: Numeric (double)

                 Range: [1,3]                         Units: 1
         Unique values: 3                        Missing .: 0/10,308

            Tabulation: Freq.  Value
                        2,429  1
                        2,703  2
                        5,176  3

------------------------------------------------------------------------------
CTX_area                                                         CTX_area
------------------------------------------------------------------------------

                  Type: Numeric (double)

                 Range: [1,2]                         Units: 1
         Unique values: 2                        Missing .: 0/10,308
```

```
            Tabulation: Freq.  Value
                        4,088  1
                        6,220  2

------------------------------------------------------------------------------
CTX_facility_type                                         CTX_facility_type
------------------------------------------------------------------------------

                  Type: Numeric (double)

                 Range: [1,2]                          Units: 1
        Unique values: 2                          Missing .: 0/10,308

            Tabulation: Freq.  Value
                        5,599  1
                        4,709  2
```

15.3.2.3 R

````
```{r}
df %>%
  skim(SDC_age_in_months,
       SDC_sex,
       CLIN_fever,
       CLIN_fever_onset,
       CLIN_diarrhoea,
       CLIN_cough,
       RX_preconsult_antibiotics,
       RX_preconsult_antimalarials,
       CTX_district,
       CTX_area,
       CTX_facility_type)
```
````

Table 15.4: Data summary

Name	Piped data
Number of rows	10308

Number of columns	39
Column type frequency:	
character	3
numeric	8
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
CTX_district	0	1	5	9	0	3	0
CTX_area	0	1	5	5	0	2	0
CTX_facility_type	0	1	10	13	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
SDC_age_in_months	0	1.0	18.75	14.90	0	7	15	27	59	
SDC_sex	4	1.0	1.49	0.50	1	1	1	2	2	
CLIN_fever	0	1.0	0.84	3.74	0	0	1	1	98	
CLIN_fever_onset	3083	0.7	2.50	1.93	0	1	2	3	14	
CLIN_diarrhoea	0	1.0	0.41	4.32	0	0	0	0	98	
CLIN_cough	0	1.0	0.69	3.75	0	0	1	1	98	
RX_preconsult_antibiotics	0	1.0	0.17	0.37	0	0	0	0	1	
RX_preconsult_antimalarials	0	1.0	0.04	0.20	0	0	0	0	1	

```
```{r}
df <- df %>%
  tibble::remove_rownames() %>%
  tibble::column_to_rownames(var="child_ID") %>%
  dplyr::mutate(across(c(SDC_sex,
                         CLIN_fever,
                         CLIN_cough,
                         CLIN_diarrhoea,
                         RX_preconsult_antibiotics,
```
```

```
                          RX_preconsult_antimalarials,
                          CTX_district,
                          CTX_area,
                          CTX_facility_type),
                        factor))
```

15.3.2.4 Identify missing values

Identify missing values in each variable

```
DataExplorer::plot_missing(df,
                           geom_label_args = list(size = 2, label.padding = unit(0.2, "lines
```


CTX_SPA_obs_after_lab
CTX_SPA_obs
MGMT_referral_src_registry
MGMT_referral_src_caregiver
RX_count_other
RX_antibiotics_src_text
RX_antibiotics
RX_quinine
RX_primaquine
RX_artesunate
RX_artemether
RX_AL
RX_ACT
RX_antimalarial_parenteral
RX_antimalarials
RX_preconsult_antimalarials
RX_preconsult_antibiotics
DX_severe
DX_count_non_malaria
DX_malaria_severe
DX_malaria
TEST_malaria_done
CLIN_diarrhoea
CLIN_cough
CLIN_fever
SDC_age_in_months
CTX_facility_type
CTX_area
CTX_facility_ID
CTX_district
CTX_day_of_week
CTX_month
SDC_sex
CLIN_fever_onset
TEST_malaria_result
TEST_malaria_type
MEAS_temperature
RX_free_text
0%
0.04%
29.91%
44.14%
89.94%
90.02%
Features
0
2500
5000
7500
Missing Rows
Band
Bad
Good
OK
Remove

```
DataExplorer::plot_bar(df %>%
                         dplyr::select(-CTX_facility_ID),
                       by = "CTX_district")
```



Page 1

Page 2

15.3.2.5 Exercise 3

Add the following two new variables to data frame `df`

Variable	Coding
SDC_age_category	<2 months
	2-11 months
	12-23 months
	24-35 months
	36-47 months
	48-59 months
CLIN_fever_onset_category	<2 days
	2-3 days
	4-6 days
	7 days

Tip

- Stata: use the gen command
- R: use the mutate and case_when functions from the `dplyr` package

```
```{r}
# Write your code here
```
```

15.3.2.6 Stata

15.3.2.7 R

```
```{r}
df <- df %>%
  dplyr::mutate(
    SDC_age_category = dplyr::case_when(
      SDC_age_in_months < 2 ~ "<2 months",
      SDC_age_in_months >= 2 & SDC_age_in_months < 12 ~ "02-11 months",
      SDC_age_in_months >= 12 & SDC_age_in_months < 24 ~ "12-23 months",
      SDC_age_in_months >= 24 & SDC_age_in_months < 36 ~ "24-35 months",
      SDC_age_in_months >= 36 & SDC_age_in_months < 48 ~ "36-47 months",
      SDC_age_in_months >= 48 & SDC_age_in_months < 60 ~ "48-59 months",
      TRUE ~ ""
    )
  ) %>%
  dplyr::mutate(
    CLIN_fever_onset_category = dplyr::case_when(
      CLIN_fever_onset < 2 ~ "<2 days",
      CLIN_fever_onset >= 2 & CLIN_fever_onset < 4 ~ "2-3 days",
      CLIN_fever_onset >= 4 & CLIN_fever_onset < 7 ~ "4-6 days",
      CLIN_fever_onset >= 7 ~ ">= 7 days",
      TRUE ~ ""
    )
  )
```
```

15.3.2.8 Python

15.3.2.9 Exercise 4

Display descriptive statistics for the following population characteristics:

> Tip
>
> - Stata
> - R: use the tbl_summary function from the `gtsummary` package

```
```{r}
# Write your code here
```
```

15.3.2.10 Stata

```
```{r}
RStata::stata('tabulate SDC_age_category
               tabulate SDC_sex
               tabulate CLIN_fever
               tabulate CLIN_fever_onset_category
               tabulate CLIN_diarrhoea
               tabulate CLIN_cough
               tabulate RX_preconsult_antibiotics
               tabulate RX_preconsult_antimalarials
               tabulate CTX_district
               tabulate CTX_area
               tabulate CTX_facility_type',
               data.in = stata_df)
```
```

```
. tabulate SDC_age_category
variable SDC_age_category not found
r(111);
.               tabulate SDC_sex
```

```
.               tabulate CLIN_fever
.               tabulate CLIN_fever_onset_category
.               tabulate CLIN_diarrhoea
.               tabulate CLIN_cough
.               tabulate RX_preconsult_antibiotics
.               tabulate RX_preconsult_antimalarials
.               tabulate CTX_district
.               tabulate CTX_area
.               tabulate CTX_facility_type
```

15.3.2.11 R

```
```{r}
df %>%
  dplyr::select(SDC_age_category,
                SDC_sex,
                CLIN_fever,
                CLIN_fever_onset_category,
                CLIN_diarrhoea,
                CLIN_cough,
                RX_preconsult_antibiotics,
                RX_preconsult_antimalarials,
                CTX_district,
                CTX_area,
                CTX_facility_type) %>%
  gtsummary::tbl_summary(missing_text = "(Missing)")
```
```

Characteristic	N = 10,308
SDC_age_category	
<2 months	597 (5.8%)
02-11 months	3,576 (35%)
12-23 months	2,947 (29%)
24-35 months	1,529 (15%)
36-47 months	980 (9.5%)
48-59 months	679 (6.6%)
SDC_sex	
1	5,229 (51%)
2	5,075 (49%)

Characteristic	N = 10,308
(Missing)	4
CLIN_fever	
0	3,068 (30%)
1	7,225 (70%)
98	15 (0.1%)
CLIN_fever_onset_category	
	3,083 (30%)
<2 days	1,998 (19%)
>= 7 days	343 (3.3%)
2-3 days	4,386 (43%)
4-6 days	498 (4.8%)
CLIN_diarrhoea	
0	7,982 (77%)
1	2,306 (22%)
98	20 (0.2%)
CLIN_cough	
0	4,658 (45%)
1	5,635 (55%)
98	15 (0.1%)
RX_preconsult_antibiotics	
0	8,573 (83%)
1	1,735 (17%)
RX_preconsult_antimalarials	
0	9,866 (96%)
1	442 (4.3%)
CTX_district	
Kaliua	2,429 (24%)
Sengerema	2,703 (26%)
Tanga	5,176 (50%)
CTX_area	
rural	4,088 (40%)
urban	6,220 (60%)
CTX_facility_type	
dispensary	5,599 (54%)
health center	4,709 (46%)

15.4 Healthcare provider actions

15.4.1 Codebook

- Temperature measured
  - Fever measured
- Fever (temp or history)
- Malaria test
- Any severe diagnosis
- Malaria diagnosis
- Malaria treatment
- Referral

Variable	Coding
MEAS_temperature	
TEST_malaria_result	0: negative 1: positive 2: indeterminate 95: unreadable result 98: not sure
DX_malaria	0: no 1: yes
RX_antimalarials	0: no 1: yes
MGMT_referral_src_caregiver	
MGMT_referral_src_registry	

15.4.2 Structure of the data

15.4.2.1 Exercise 5

Examine the structure of the data, including variable names, labels.

Tip

- Stata: use the codebook command
- R: use the skim function from the `skimr` package

```
```{r}
# Write your code here
```
```

15.4.2.2 Stata

```
RStata::stata("codebook MEAS_temperature TEST_malaria_result TEST_malaria_result DX_malaria
              data.in = stata_df)
```

```
. codebook MEAS_temperature TEST_malaria_result TEST_malaria_result DX_malaria
> RX_antimalarials MGMT_referral_src_caregiver MGMT_referral_src_registry

-------------------------------------------------------------------------------
MEAS_temperature                                                MEAS_temperature
-------------------------------------------------------------------------------

                  Type: Numeric (double)

                 Range: [34.5,42.5]                   Units: .1
         Unique values: 16                         Missing .: 9,271/10,308

                  Mean: 37.0781
             Std. dev.: .977139

           Percentiles:     10%       25%       50%       75%       90%
                             36      36.5        37      37.5      38.5

-------------------------------------------------------------------------------
TEST_malaria_result                                      TEST_malaria_result
-------------------------------------------------------------------------------

                  Type: Numeric (double)
```

```
        Range: [0,98]                        Units: 1
Unique values: 5                          Missing .: 4,550/10,308

   Tabulation: Freq.  Value
               4,665  0
               1,032  1
                   1  2
                   3  95
                  57  98
               4,550  .
```

TEST_malaria_result TEST_malaria_result

```
        Type: Numeric (double)

        Range: [0,98]                        Units: 1
Unique values: 5                          Missing .: 4,550/10,308

   Tabulation: Freq.  Value
               4,665  0
               1,032  1
                   1  2
                   3  95
                  57  98
               4,550  .
```

DX_malaria DX_malaria

```
        Type: Numeric (double)

        Range: [0,1]                         Units: 1
Unique values: 2                          Missing .: 0/10,308

   Tabulation: Freq.  Value
               8,508  0
               1,800  1
```

```
-------------------------------------------------------------------------------
RX_antimalarials                                              RX_antimalarials
-------------------------------------------------------------------------------

                  Type: Numeric (double)

                 Range: [0,1]                         Units: 1
         Unique values: 2                         Missing .: 0/10,308

            Tabulation: Freq.  Value
                        9,018  0
                        1,290  1

-------------------------------------------------------------------------------
MGMT_referral_src_caregiver                     MGMT_referral_src_caregiver
-------------------------------------------------------------------------------

                  Type: Numeric (double)

                 Range: [0,98]                        Units: 1
         Unique values: 4                         Missing .: 0/10,308

            Tabulation: Freq.  Value
                       10,122  0
                          164  1
                            9  97
                           13  98

-------------------------------------------------------------------------------
MGMT_referral_src_registry                       MGMT_referral_src_registry
-------------------------------------------------------------------------------

                  Type: Numeric (double)

                 Range: [0,1]                         Units: 1
         Unique values: 2                         Missing .: 0/10,308

            Tabulation: Freq.  Value
                       10,194  0
                          114  1
```

15.4.2.3 R

```
df %>%
  skimr::skim(MEAS_temperature,
              TEST_malaria_result,
              TEST_malaria_result,
              DX_malaria,
              RX_antimalarials,
              MGMT_referral_src_caregiver,
              MGMT_referral_src_registry)
```

Table 15.10: Data summary

Name	Piped data
Number of rows	10308
Number of columns	40
_____	
Column type frequency:	
numeric	6
_____	
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
MEAS_temperature	9271	0.10	37.08	0.98	34.5	36.5	37	37.5	42.5	
TEST_malaria_result	4550	0.56	1.20	9.93	0.0	0.0	0	0.0	98.0	
DX_malaria	0	1.00	0.17	0.38	0.0	0.0	0	0.0	1.0	
RX_antimalarials	0	1.00	0.13	0.33	0.0	0.0	0	0.0	1.0	
MGMT_referral_src_caregiver	0	1.00	0.22	4.50	0.0	0.0	0	0.0	98.0	
MGMT_referral_src_registry	0	1.00	0.01	0.10	0.0	0.0	0	0.0	1.0	

15.4.2.4 Exercise 3

Add the following two new variables to data frame `df`

- MEAS_fever
- Fever (temp or history)

Tip

- Stata: use the gen command
- R: use the mutate function from the `dplyr` package

```
```{r}
# Write your code here
```
```

15.4.2.5 Stata

15.4.2.6 R

```
```{r}
df <- df %>%
  dplyr::mutate(CALC_temperature_measured = !is.na(MEAS_temperature)) %>%
  dplyr::mutate(CALC_fever = MEAS_temperature >= 37.5) %>%
  dplyr::mutate(CALC_fever_or_temp = (CLIN_fever == 1) | (CALC_fever == 1))
```
```

15.4.2.7 Python

15.4.2.8 Exercise 6

Display descriptive statistics for the following healthcare provider actions:

Tip

- R: use the tbl_summary function from the `gtsummary` package

```
```{r}
# Write your code here
```
```

15.4.2.9 Stata

15.4.2.10 R

```
```{r}
df %>%
  dplyr::select(CALC_temperature_measured,
                CALC_fever,
                CALC_fever_or_temp,
                TEST_malaria_result,
                TEST_malaria_result,
                DX_malaria,
                RX_antimalarials,
                MGMT_referral_src_caregiver,
                MGMT_referral_src_registry) %>%
  gtsummary::tbl_summary(missing_text = "(Missing)")
```
```

Characteristic	**N = 10,308**
CALC_temperature_measured	1,037 (10%)
CALC_fever	326 (31%)
(Missing)	9,271
CALC_fever_or_temp	7,252 (97%)
(Missing)	2,842
TEST_malaria_result	
0	4,665 (81%)
1	1,032 (18%)
2	1 (<0.1%)
95	3 (<0.1%)
98	57 (1.0%)
(Missing)	4,550
DX_malaria	1,800 (17%)
RX_antimalarials	1,290 (13%)
MGMT_referral_src_caregiver	
0	10,122 (98%)
1	164 (1.6%)
97	9 (<0.1%)
98	13 (0.1%)
MGMT_referral_src_registry	114 (1.1%)

15.5 Number of consultations by facility

15.5.1 Exercise 6

Plot the number of consultations by facility in bars, grouped by district.

> Tip
>
> - Stata:
> - R:

15.5.2 Stata

````
```{r}
stata_cmd <- '
graph bar (count) child_ID, over(CTX_facility_ID, axis(off)) over(CTX_district, relabel(1 "K
graph export ./images/day02_stata_plot.png, replace
'
RStata::stata(stata_cmd,
              data.in = stata_df)
```
````

```
.
. graph bar (count) child_ID, over(CTX_facility_ID, axis(off)) over(CTX_distric
> t, relabel(1 "Kaliua" 2 "Sengerema" 3 "Tanga")) nofill ytitle(No. consultatio
> ns) title(No. consultations by facility)
. graph export ./images/day02_stata_plot.png, replace
file ./images/day02_stata_plot.png saved as PNG format
```



15.5.3 R

```
```{r}
df %>%
  dplyr::group_by(CTX_district,
                  CTX_facility_ID) %>%
  count() %>%
  ggplot2::ggplot(aes(x = haven::as_factor(CTX_district),
                      y = n)) +
  ggplot2::geom_bar(position = position_dodge2(preserve = "single"),
                    stat="identity") +
  ggplot2::labs(x = "", y = "No. consultations") +
  ggthemes::theme_hc()
```
```



15.6 Fever assessment

- temp measurement by reported fever; by facility

```
```{r}
# Write here
```
```

```
chisq.test(df$CLIN_fever, df$TEST_malaria_result)
```

```
    Pearson's Chi-squared test

data:  df$CLIN_fever and df$TEST_malaria_result
X-squared = 61.449, df = 8, p-value = 2.42e-10
```

```
```{r}
df$CTX_facility_ID <- haven::as_factor(df$CTX_facility_ID)
df %>%
  dplyr::filter(CLIN_fever == 1) %>%
  dplyr::select(MEAS_temperature,
                CTX_facility_ID) %>%
```
```

```
  gtsummary::tbl_summary(by = CTX_facility_ID) %>%
  add_p()
```

Characteristic	N = [illegible]	p-value
MEASURED_temperature	[illegible]	0.001
Unknown	[illegible]	

- also showing 'prevalence' of fever when of whole clinic vs of those who measure to indicate bias

```
```{r}
# Write here
```
```

15.7 Malaria tests

- malaria tests of those with history or measured fever

```
```{r}
# Write here
```
```

```
```{r}
table(df$CALC_fever_or_temp, df$TEST_malaria_result)
```
```

```
         0    1    2   95   98
FALSE   70    2    0    1    1
TRUE  3952  966    1    2   47
```

```r
chisq.test(df$CALC_fever_or_temp, df$TEST_malaria_result)
```

```
    Pearson's Chi-squared test

data:  df$CALC_fever_or_temp and df$TEST_malaria_result
X-squared = 33.92, df = 4, p-value = 7.74e-07
```

```r
```{r}
df$CTX_facility_ID <- haven::as_factor(df$CTX_facility_ID)
df %>%
  dplyr::filter(CALC_fever_or_temp == 1) %>%
  dplyr::select(TEST_malaria_result,
                CTX_facility_ID) %>%
  gtsummary::tbl_summary(by = CTX_facility_ID)
```
```

Characteristic	F8[illegible], N = 199	[illegible], N = 78	[illegible], N = 97	[illegible], N = 40	[illegible], N = 62	[illegible], N = 93	[illegible], N = 58	[illegible], N = 46	[illegible], N = 75	[illegible], N = 130	[illegible], N = 155	[illegible], N = 60	[illegible], N = 327	[illegible], N = 62	[illegible], N = 358	[illegible], N = 330	[illegible], N = 683	[illegible], N = 416
TEST_malaria_result																		
0	83 (67%)	151 (53%)	319 (54%)	125 (82%)	285 (75%)	135 (98%)	159 (95%)	37 (93%)	122 (95%)	62 (96%)	134 (97%)	46 (97%)	221 (97%)	25 (57%)	242 (92%)	225 (91%)	511 (97%)	287 (97%)
1	38 (31%)	130 (46%)	270 (46%)	28 (18%)	95 (25%)	0 (0%)	2 (1.2%)	2 (6.3%)	3 (2.9%)	4 (2.8%)	6 (4.1%)	2 (4.3%)	6 (2.7%)	18 (41%)	20 (7.6%)	20 (8.1%)	14 (2.7%)	10 (3.4%)
2	0 (0%)	0 (0%)	1 (0.2%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
95	1 (0.8%)	0 (0%)	1 (0.2%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
98	2 (1.6%)	5 (1.8%)	1 (0.2%)	0 (0%)	0 (0%)	3 (2.2%)	6 (3.6%)	2 (6.3%)	6 (5.1%)	6 (4.2%)	0 (0%)	3 (4.3%)	5 (2.3%)	4 (9.3%)	0 (0%)	1 (0.4%)	3 (0.6%)	0 (0%)
Unknown	66	12	518	844	26	71	191	69	252	78	3	23	248	18	56	83	114	119

15.8 Malaria treatments

- malaria diagnoses vs. positive tests vs. treatment.

```{r}
# Write here
```

```{r}
df <- df %>%
  dplyr::mutate(TEST_malaria_positive = 1* (TEST_malaria_result == 1))
table(df$TEST_malaria_result, df$DX_malaria, df$RX_antimalarials) %>% knitr::kable()
```

Var1	Var2	Var3	Freq
0	0	0	3890
1	0	0	15
2	0	0	1
95	0	0	2
98	0	0	31
0	1	0	645
1	1	0	49
2	1	0	0
95	1	0	1
98	1	0	17
0	0	1	99
1	0	1	50
2	0	1	0
95	0	1	0
98	0	1	2
0	1	1	31
1	1	1	918
2	1	1	0
95	1	1	0
98	1	1	7

```{r}
chisq.test(df$TEST_malaria_result, df$DX_malaria)
```

```
    Pearson's Chi-squared test

data:  df$TEST_malaria_result and df$DX_malaria
```

```
X-squared = 2582, df = 4, p-value < 2.2e-16
```

```r
chisq.test(df$TEST_malaria_result, df$RX_antimalarials)
```

```
    Pearson's Chi-squared test

data:  df$TEST_malaria_result and df$RX_antimalarials
X-squared = 4508.8, df = 4, p-value < 2.2e-16
```

```r
```{r}
# df$CTX_facility_ID <- haven::as_factor(df$CTX_facility_ID)
# df %>%
#   dplyr::filter(TEST_malaria_positive == 1) %>%
#   dplyr::select(DX_malaria,
#                 RX_antimalarials,
#                 CTX_facility_ID) %>%
#   gtsummary::tbl_summary(by = CTX_facility_ID)
```
```

16 Malaria case study - Report #1

16.1 Introduction

16.1.1 Instructions

The data set used for this report is stored in **dataset2.dta** or **dataset2.xlsx** or **dataset2.csv**.

1. Derive new variables to data frame `df`
2. Describe and critically discuss the distribution of consultations by facilities depending on their characteristics (urban vs. rural, health centre vs. dispensary, district)
3. Describe and critically discuss the demographics, clinical presentation and clinical history of patients.
4. Describe and critically discuss the clinical assessments (fever assessment, malaria test) conducted during the consultation.

Table 16.1: Extract of the database

child_ID	CTX_month	CTX_district	SDC_age_in_months
1	7	Kaliua	10
2	7	Kaliua	6
3	7	Kaliua	6
4	7	Kaliua	11
5	8	Kaliua	21

16.1.2 Codebook

16.1.2.1 Numerical variables

Variable
SDC_age_in_months
CLIN_fever_onset
MEAS_temperature

16.1.2.2 Categorical variables

Variable	Coding
SDC_sex	1: male
	2: female
	98: unknown
CLIN_fever	0: no
	1: yes
	98: not sure
CLIN_cough	0: no
	1: yes
	98: not sure
CLIN_diarrhoea	0: no
	1: yes
	98: not sure
RX_preconsult_antibiotics	0: no
	1: yes
RX_preconsult_antimalarials	0: no
	1: yes
CTX_district	Kaliua
	Sengerema
	Tanga
CTX_area	urban
	rural
CTX_facility_type	dispensary
	health centre
TEST_malaria_result	0: negative
	1: positive
	2: indeterminate
	95: unreadable result
	98: not sure

Variable	Coding
DX_malaria	0: no
	1: yes
DX_malaria_severe	0: no
	1: yes
RX_antimalarials	0: no
	1: yes
RX_artemether	0: no
	1: yes
RX_antibiotics	0: no
	1: yes
MGMT_referral_src_caregiver	
MGMT_referral_src_registry	

16.1.3 Derive new variables

Add the following two new variables to data frame `df`

Variable	Coding
SDC_age_category	<2 months
	2-11 months
	12-23 months
	24-35 months
	36-47 months
	48-59 months
CLIN_fever_onset_category	<2 days
	2-3 days
	4-6 days
	7 days

16.1.4 Structure of the data

Table 16.5: Data summary

Name	Piped data
Number of rows	10308
Number of columns	40
Column type frequency:	
factor	9
numeric	2
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
SDC_sex	4	1	FALSE	2	1: 5229, 2: 5075
CLIN_fever	0	1	FALSE	3	1: 7225, 0: 3068, 98: 15
CLIN_diarrhoea	0	1	FALSE	3	0: 7982, 1: 2306, 98: 20
CLIN_cough	0	1	FALSE	3	1: 5635, 0: 4658, 98: 15
RX_preconsult_antibiotics	0	1	FALSE	2	0: 8573, 1: 1735
RX_preconsult_antimalarials	0	1	FALSE	2	0: 9866, 1: 442
CTX_district	0	1	FALSE	3	Tan: 5176, Sen: 2703, Kal: 2429
CTX_area	0	1	FALSE	2	urb: 6220, rur: 4088
CTX_facility_type	0	1	FALSE	2	dis: 5599, hea: 4709

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
SDC_age_in_months	0	1.0	18.75	14.90	0	7	15	27	59	
CLIN_fever_onset	3083	0.7	2.50	1.93	0	1	2	3	14	

Table 16.8: Data summary

Name	Piped data
Number of rows	10308
Number of columns	43
Column type frequency:	
factor	12
numeric	1
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
TEST_malaria_done	0	1.00	FALSE	2	1: 5763, 0: 4545
TEST_malaria_type	4550	0.56	FALSE	4	1: 5371, 2: 340, 98: 45, 95: 2
TEST_malaria_result	4550	0.56	FALSE	5	0: 4665, 1: 1032, 98: 57, 95: 3
DX_severe	0	1.00	FALSE	2	0: 10054, 1: 254
DX_malaria	0	1.00	FALSE	2	0: 8508, 1: 1800
DX_malaria_severe	0	1.00	FALSE	2	0: 10160, 1: 148
RX_antimalarials	0	1.00	FALSE	2	0: 9018, 1: 1290
RX_antimalarial_parenteral	0	1.00	FALSE	2	0: 10059, 1: 249
RX_artemether	0	1.00	FALSE	2	0: 10248, 1: 60
RX_antibiotics	0	1.00	FALSE	2	1: 5430, 0: 4878
MGMT_referral_src	0	1.00	FALSE	4	0: 10122, 1: 164, 98: 13, 97: 9

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
MGMT_referral_src	0	1.00	FALSE	2	0: 10194, 1: 114

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
MEAS_temperature	927	0.1	37.08	0.98	34.5	36.5	37	37.5	42.5	

16.2 Facility characteristics

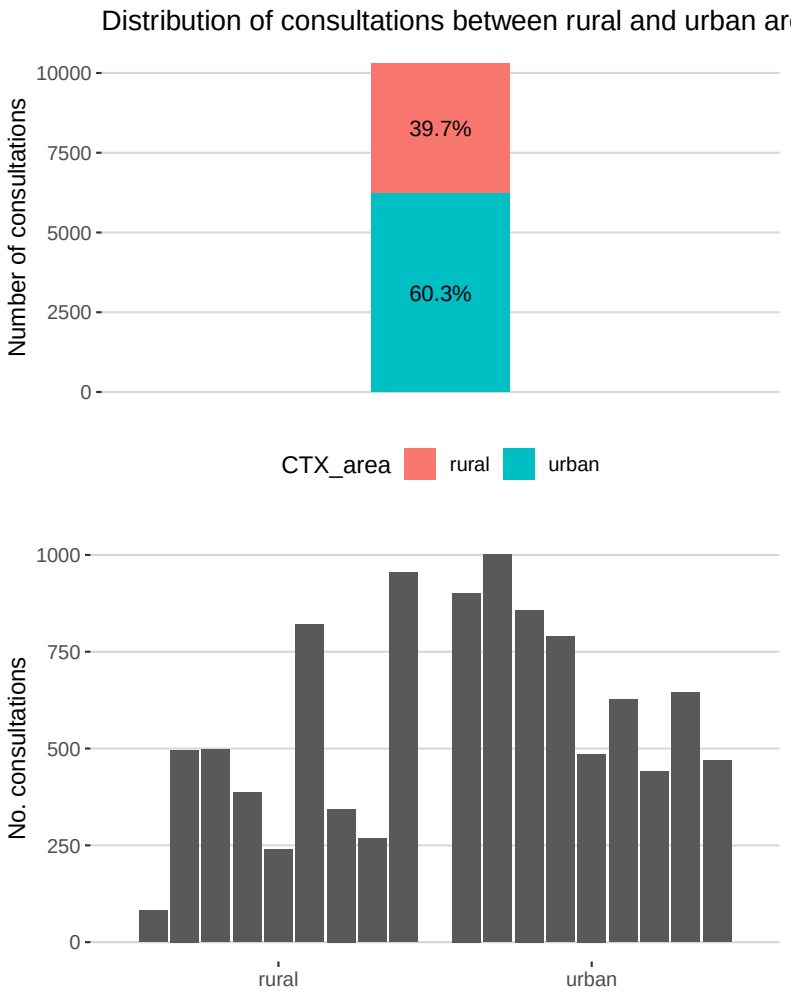
Describe and critically discuss the distribution of consultations by facilities depending on their characteristics (urban vs. rural, health centre vs. dispensary, district)

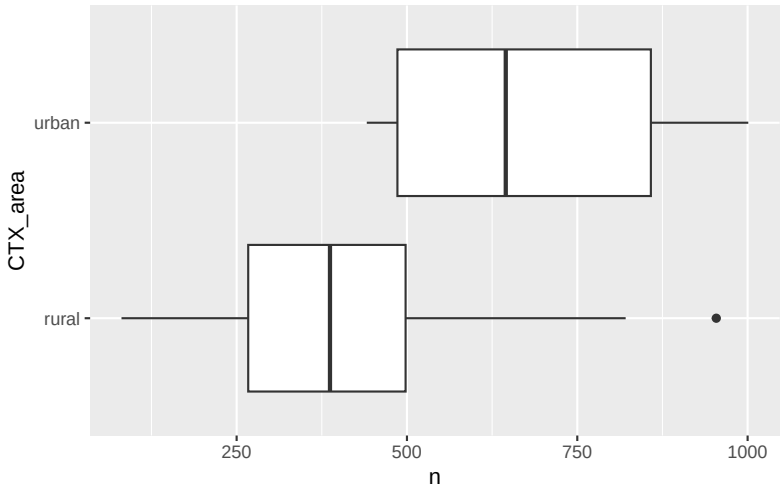
Characteristic	**N = 10,308**
CTX_facility_type	
dispensary	5,599 (54%)
health center	4,709 (46%)
CTX_area	
rural	4,088 (40%)
urban	6,220 (60%)
CTX_district	
Kaliua	2,429 (24%)
Sengerema	2,703 (26%)
Tanga	5,176 (50%)

16.2.1 Urban / rural consultations

Describe the distribution of consultations in urban vs. rural facilities

Area	Freq
rural	0.3965852
urban	0.6034148

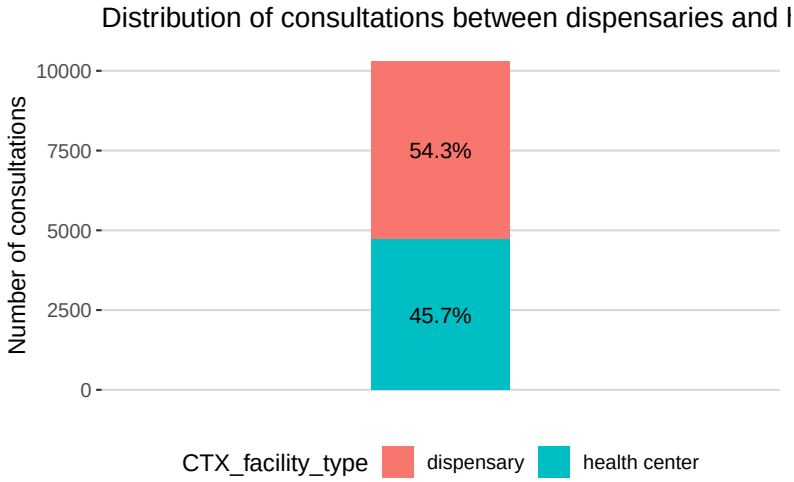

Distribution of consultations between rural and urban ar
Number of consultations
10000
7500
5000
2500
0
39.7%
60.3%
CTX_area
rural
urban
No. consultations
1000
750
500
250
0
rural
urban

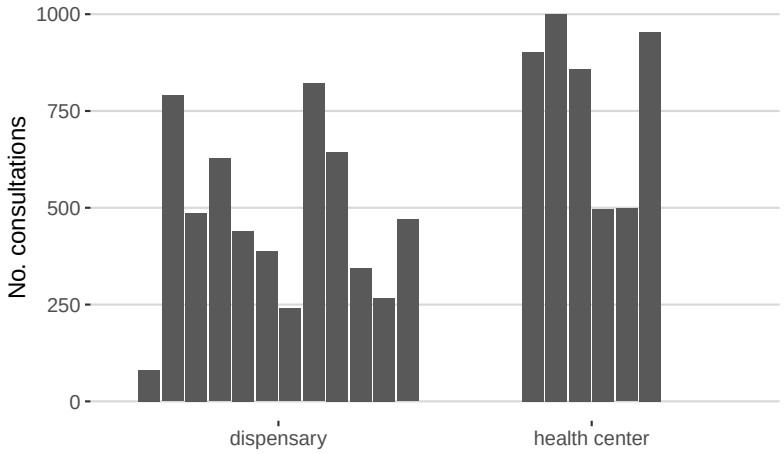


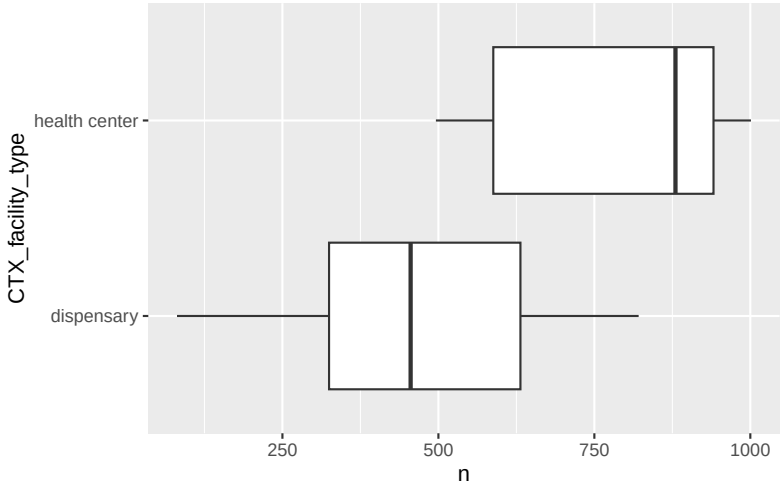
16.2.2 Consultations by type of facility

Describe the distribution of consultations in dispensaries vs. health centres.

Type	Freq
dispensary	0.5431704
health center	0.4568296



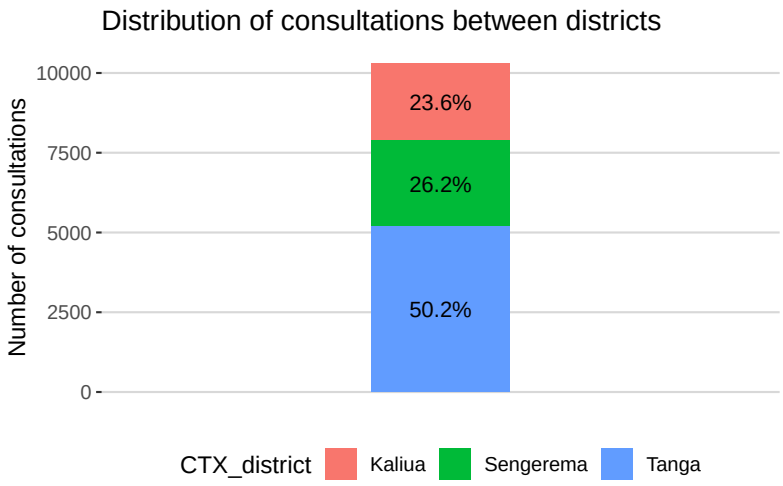




16.2.3 Consultations by district

Describe the distribution of consultations by district (Kaliua vs. Sengerema vs. tanga)

District	Freq
Kaliua	0.2356422
Sengerema	0.2622235
Tanga	0.5021343



16.2.3.1 Number of consultations by facility within each district

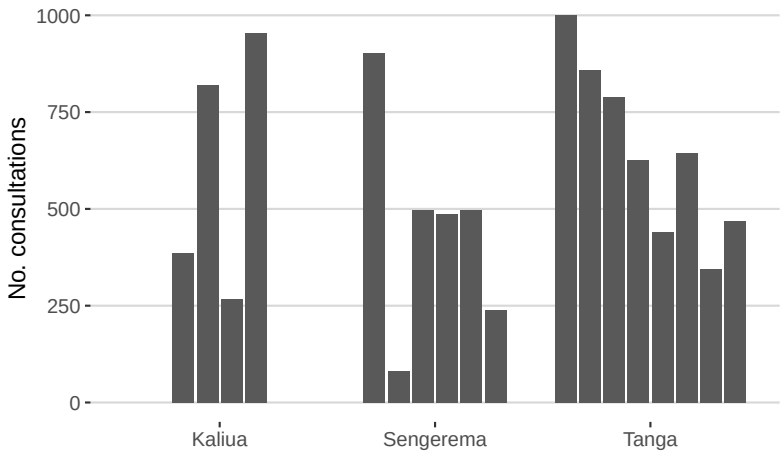
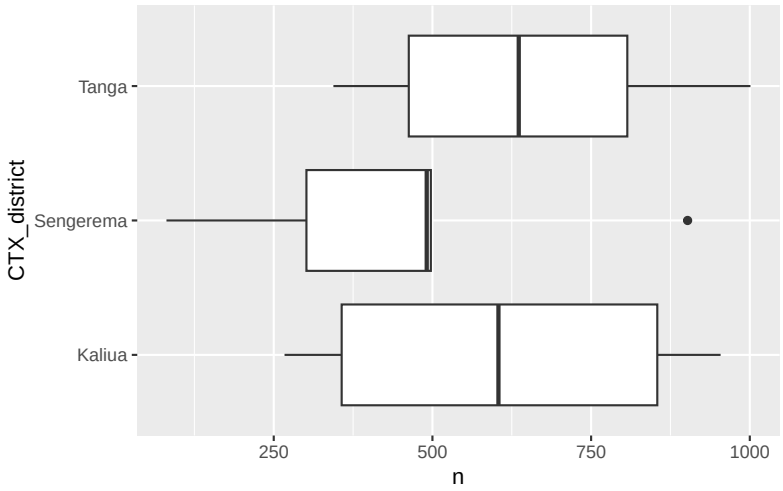


Figure 16.1: **?(caption)**



16.2.4 Consultations by area, type of facility and district

Area	Type	District	Freq
rural	dispensary	Kaliua	1475
urban	dispensary	Kaliua	0
rural	health center	Kaliua	954
urban	health center	Kaliua	0
rural	dispensary	Sengerema	321
urban	dispensary	Sengerema	486
rural	health center	Sengerema	994
urban	health center	Sengerema	902
rural	dispensary	Tanga	344
urban	dispensary	Tanga	2973
rural	health center	Tanga	0
urban	health center	Tanga	1859

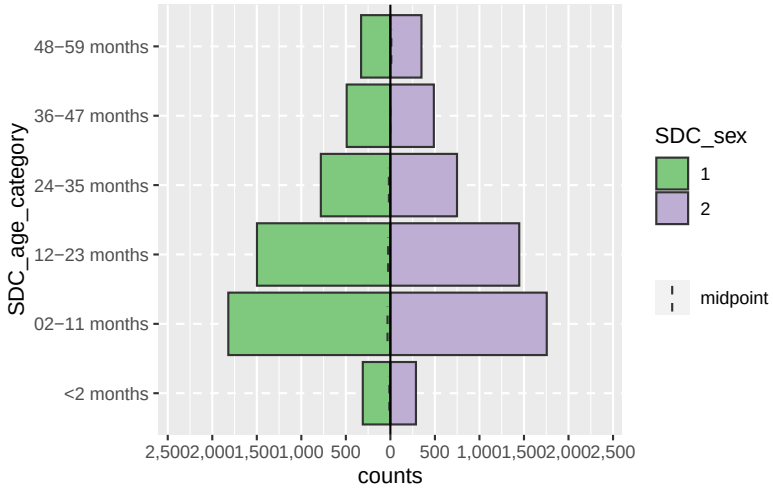
16.3 Patient characteristics

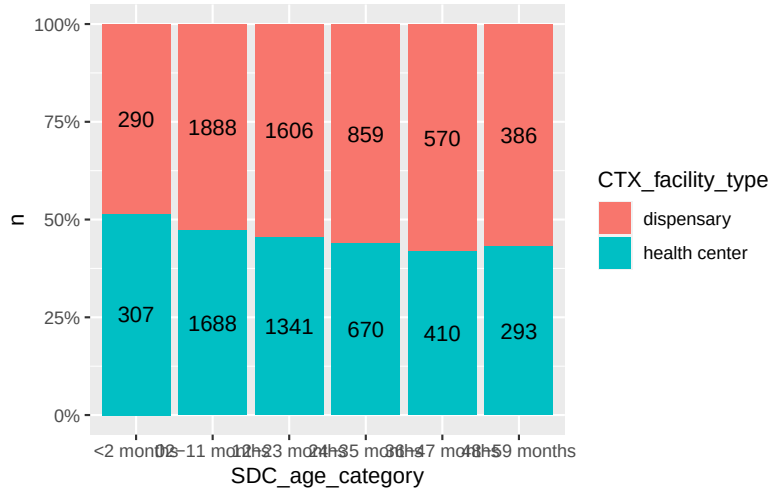
Describe and critically discuss the demographics, clinical presentation and clinical history of patients.

Characteristic	N = 10,308
SDC_age_category	
<2 months	597 (5.8%)
02-11 months	3,576 (35%)
12-23 months	2,947 (29%)
24-35 months	1,529 (15%)
36-47 months	980 (9.5%)
48-59 months	679 (6.6%)
SDC_sex	
1	5,229 (51%)
2	5,075 (49%)
(Missing)	4
CLIN_fever	
0	3,068 (30%)
1	7,225 (70%)
98	15 (0.1%)
CLIN_fever_onset_category	
	3,083 (30%)
< 2 days	1,998 (19%)
>= 7 days	343 (3.3%)
2-3 days	4,386 (43%)
4-6 days	498 (4.8%)
CLIN_diarrhoea	
0	7,982 (77%)
1	2,306 (22%)
98	20 (0.2%)
CLIN_cough	
0	4,658 (45%)
1	5,635 (55%)
98	15 (0.1%)
RX_preconsult_antibiotics	
0	8,573 (83%)
1	1,735 (17%)
RX_preconsult_antimalarials	
0	9,866 (96%)
1	442 (4.3%)

Characteristic	**dispensary**, N = 5,599	**health center**, N = 4,709
SDC_age_category		
<2 months	290 (5.2%)	307 (6.5%)
02-11 months	1,888 (34%)	1,688 (36%)
12-23 months	1,606 (29%)	1,341 (28%)
24-35 months	859 (15%)	670 (14%)
36-47 months	570 (10%)	410 (8.7%)
48-59 months	386 (6.9%)	293 (6.2%)
SDC_sex		
1	2,810 (50%)	2,419 (51%)
2	2,786 (50%)	2,289 (49%)
(Missing)	3	1
CLIN_fever		
0	1,710 (31%)	1,358 (29%)
1	3,881 (69%)	3,344 (71%)
98	8 (0.1%)	7 (0.1%)
CLIN_fever_onset_category		
	1,718 (31%)	1,365 (29%)
< 2 days	1,167 (21%)	831 (18%)
>= 7 days	149 (2.7%)	194 (4.1%)
2-3 days	2,333 (42%)	2,053 (44%)
4-6 days	232 (4.1%)	266 (5.6%)
CLIN_diarrhoea		
0	4,457 (80%)	3,525 (75%)
1	1,135 (20%)	1,171 (25%)
98	7 (0.1%)	13 (0.3%)
CLIN_cough		
0	2,365 (42%)	2,293 (49%)
1	3,226 (58%)	2,409 (51%)
98	8 (0.1%)	7 (0.1%)
RX_preconsult_antibiotics		
0	4,828 (86%)	3,745 (80%)
1	771 (14%)	964 (20%)
RX_preconsult_antimalarials		
0	5,468 (98%)	4,398 (93%)
1	131 (2.3%)	311 (6.6%)

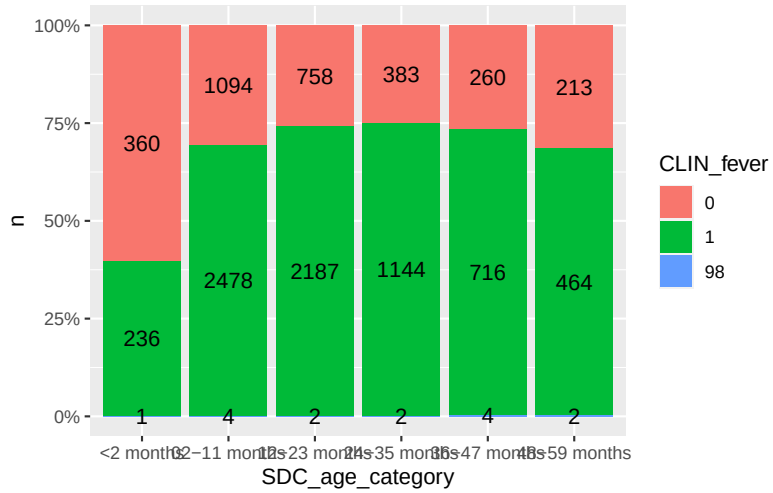
16.3.1 Demographics

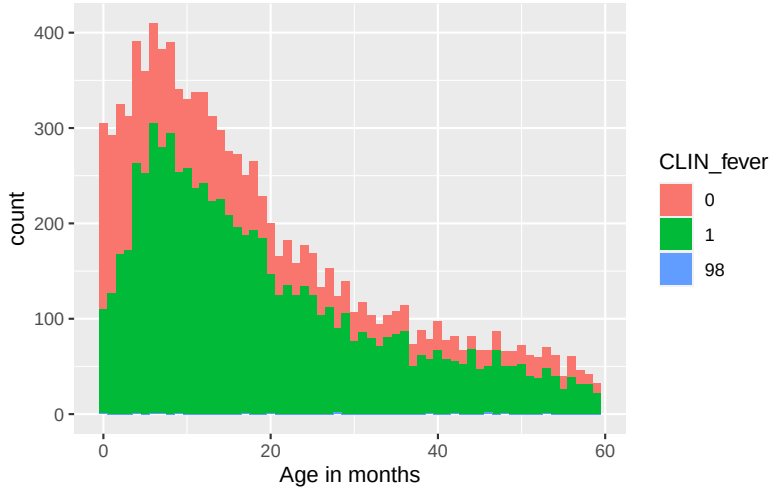




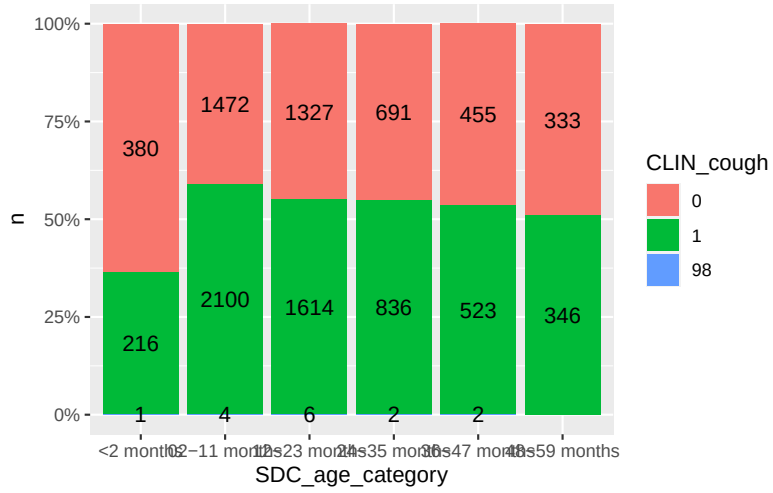
16.3.2 Clinical presentation

16.3.2.1 Fever



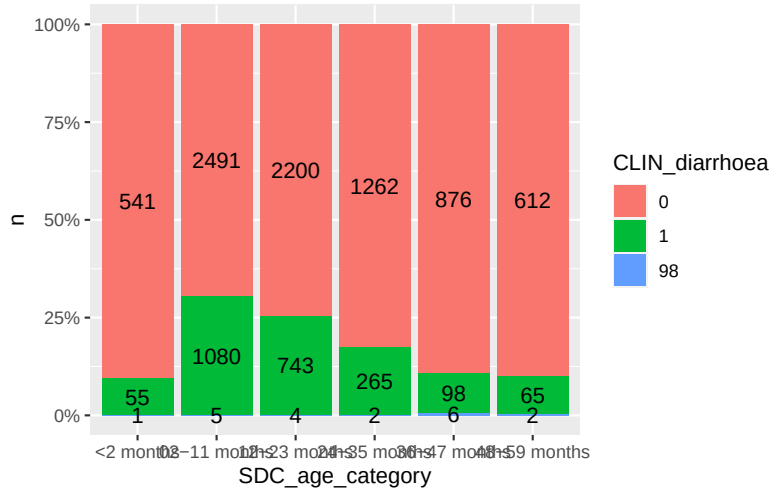


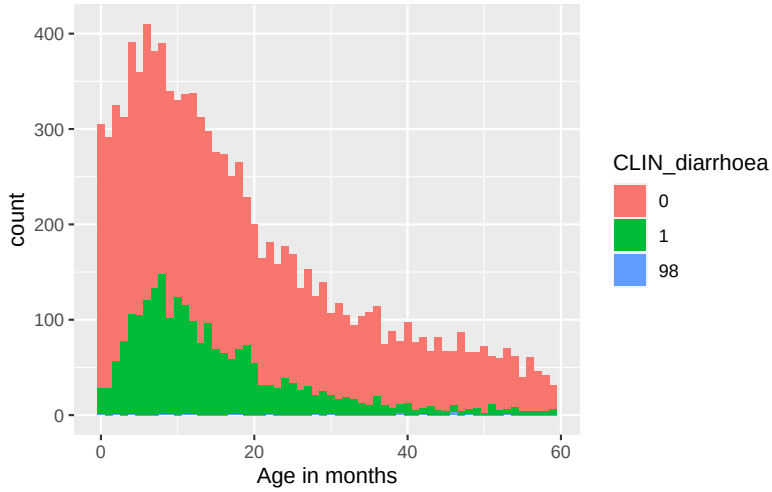
16.3.2.2 Cough


100%
75%
50%
25%
0%
n
380
216
1472
2100
1327
1614
691
836
455
523
333
346
1
4
6
2
2
<2 months
02−11 months
12−23 months
24−35 months
36−47 months
48−59 months
SDC_age_category
CLIN_cough
0
1
98

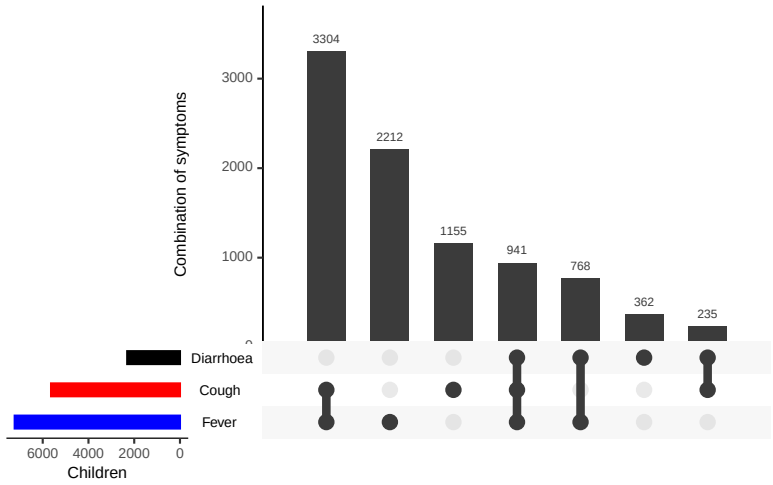

400
300
200
100
0
count
0
20
40
60
Age in months
CLIN_cough
0
1
98

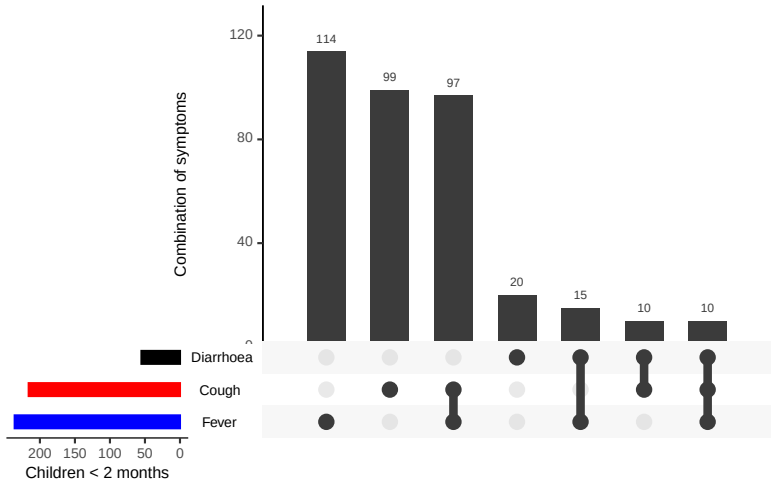
16.3.2.3 Diarrhoea

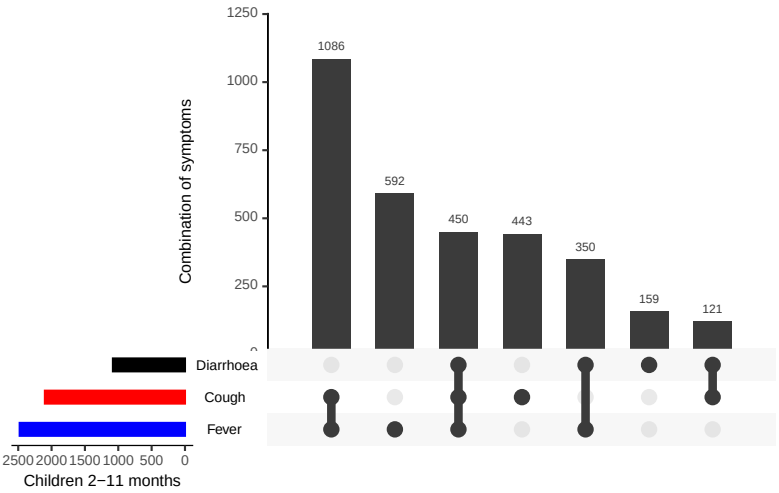


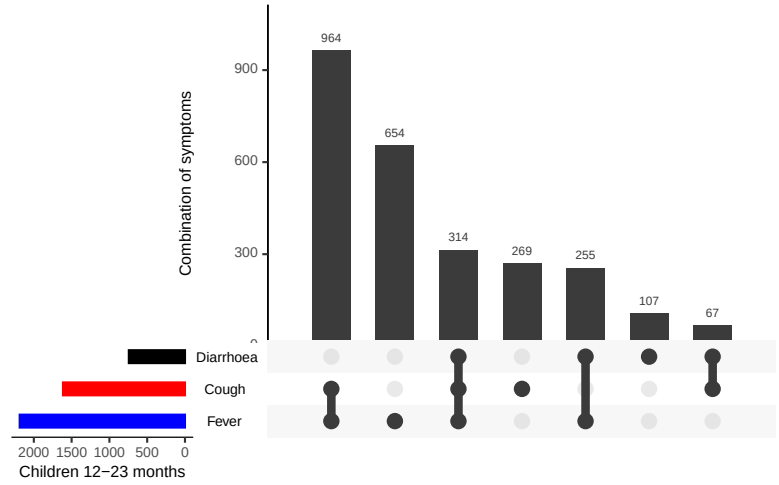


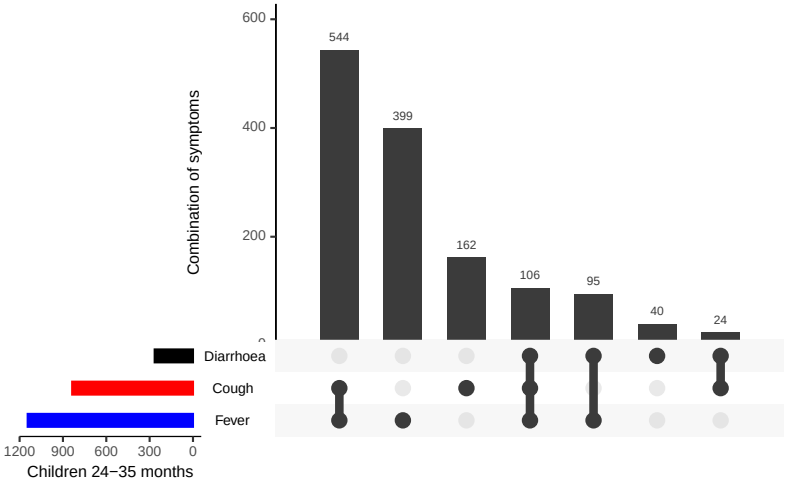
16.3.2.4 Combination of symptoms

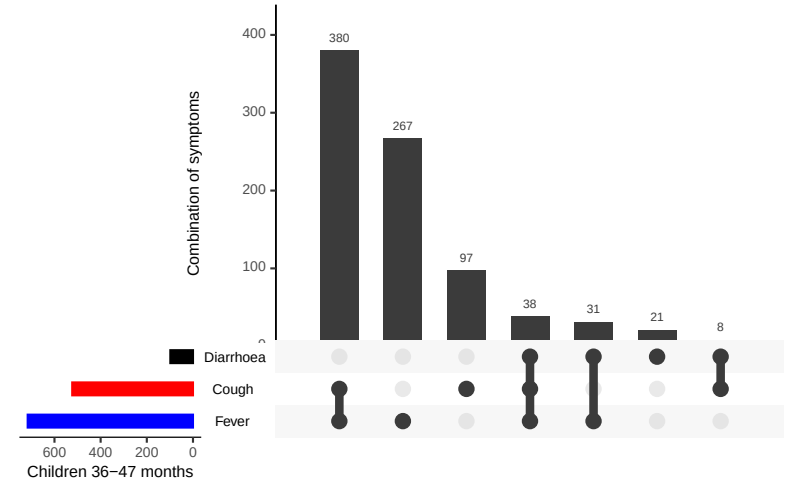


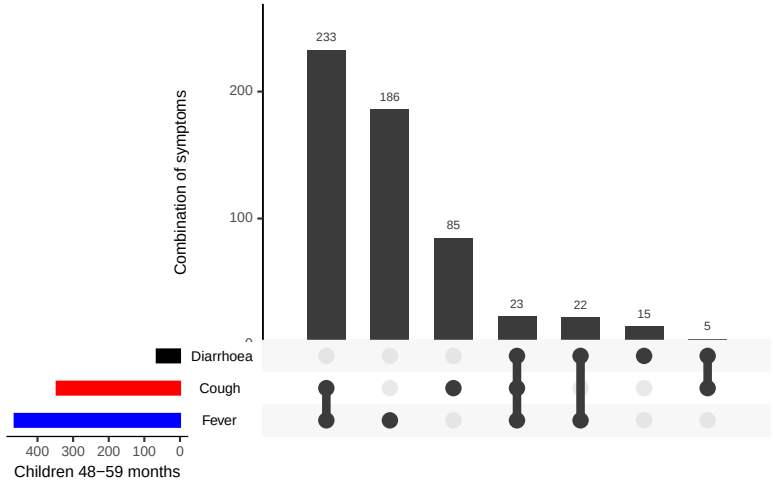



1250
1000
750
500
250
0
Combination of symptoms
1086
592
450
443
350
159
121
Diarrhoea
Cough
Fever
2500 2000 1500 1000 500 0
Children 2−11 months


900
600
300
0
Combination of symptoms
964
654
314
269
255
107
67
Diarrhoea
Cough
Fever
2000 1500 1000 500 0
Children 12−23 months


600
400
200
0
544
399
162
106
95
40
24
Combination of symptoms
Diarrhoea
Cough
Fever
1200
900
600
300
0
Children 24−35 months


400
300
200
100
0
380
267
97
38
31
21
8
Combination of symptoms
Diarrhoea
Cough
Fever
600
400
200
0
Children 36−47 months



16.4 Clinical assessments conducted during the consultation

Describe and critically discuss the clinical assessments (fever assessment, malaria test) conducted during the consultation.

Characteristic	N = 10,308
CALC_temperature_measured	
0	9,271 (90%)
1	1,037 (10%)
CALC_fever_measured	
0	711 (69%)
1	326 (31%)
(Missing)	9,271
CALC_fever_all	
0	3,056 (30%)
1	7,252 (70%)
TEST_malaria_done	
0	4,545 (44%)
1	5,763 (56%)
TEST_malaria_type	
1	5,371 (93%)
2	340 (5.9%)

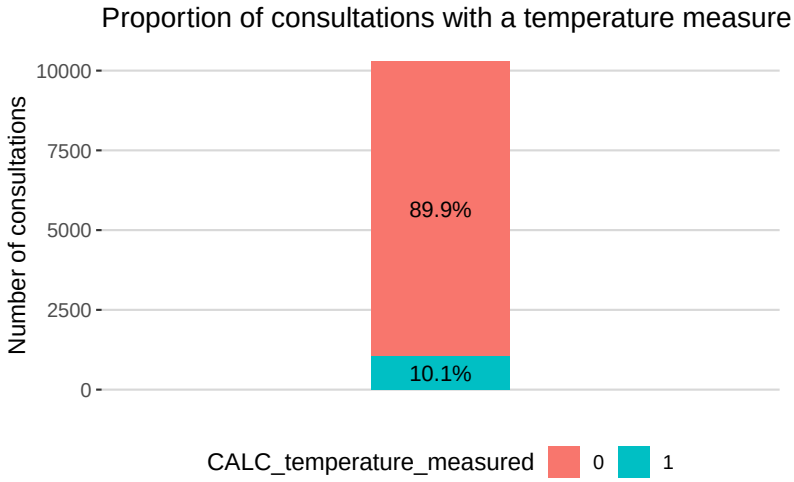
Characteristic	N = 10,308
95	2 (<0.1%)
98	45 (0.8%)
(Missing)	4,550
TEST_malaria_result	
0	4,665 (81%)
1	1,032 (18%)
2	1 (<0.1%)
95	3 (<0.1%)
98	57 (1.0%)
(Missing)	4,550
DX_severe	
0	10,054 (98%)
1	254 (2.5%)
DX_malaria	
0	8,508 (83%)
1	1,800 (17%)
DX_malaria_severe	
0	10,160 (99%)
1	148 (1.4%)
RX_antimalarials	
0	9,018 (87%)
1	1,290 (13%)
RX_antimalarial_parenteral	
0	10,059 (98%)
1	249 (2.4%)
RX_antibiotics	
0	4,878 (47%)
1	5,430 (53%)
MGMT_referral_src_caregiver	
0	10,122 (98%)
1	164 (1.6%)
97	9 (<0.1%)
98	13 (0.1%)
MGMT_referral_src_registry	
0	10,194 (99%)
1	114 (1.1%)

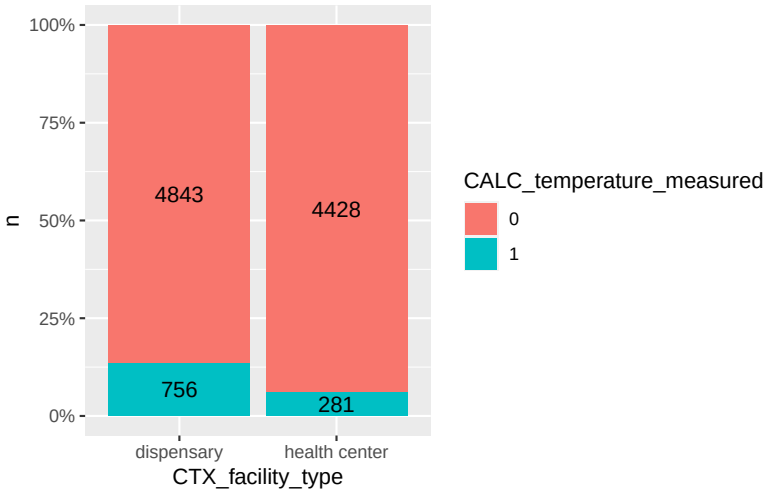
Characteristic	**dispensary**, N = 5,599	**health center**, N = 4,709
CALC_temperature_measured		
0	4,843 (86%)	4,428 (94%)
1	756 (14%)	281 (6.0%)
CALC_fever_measured		
0	507 (67%)	204 (73%)
1	249 (33%)	77 (27%)
(Missing)	4,843	4,428
CALC_fever_all		
0	1,699 (30%)	1,357 (29%)
1	3,900 (70%)	3,352 (71%)
TEST_malaria_done		
0	2,589 (46%)	1,956 (42%)
1	3,010 (54%)	2,753 (58%)
TEST_malaria_type		
1	2,836 (94%)	2,535 (92%)
2	134 (4.5%)	206 (7.5%)
95	1 (<0.1%)	1 (<0.1%)
98	34 (1.1%)	11 (0.4%)
(Missing)	2,594	1,956
TEST_malaria_result		
0	2,414 (80%)	2,251 (82%)
1	544 (18%)	488 (18%)
2	0 (0%)	1 (<0.1%)
95	2 (<0.1%)	1 (<0.1%)
98	45 (1.5%)	12 (0.4%)
(Missing)	2,594	1,956
DX_severe		
0	5,540 (99%)	4,514 (96%)
1	59 (1.1%)	195 (4.1%)
DX_malaria		
0	4,405 (79%)	4,103 (87%)
1	1,194 (21%)	606 (13%)
DX_malaria_severe		
0	5,579 (100%)	4,581 (97%)
1	20 (0.4%)	128 (2.7%)
RX_antimalarials		
0	4,923 (88%)	4,095 (87%)
1	676 (12%)	614 (13%)

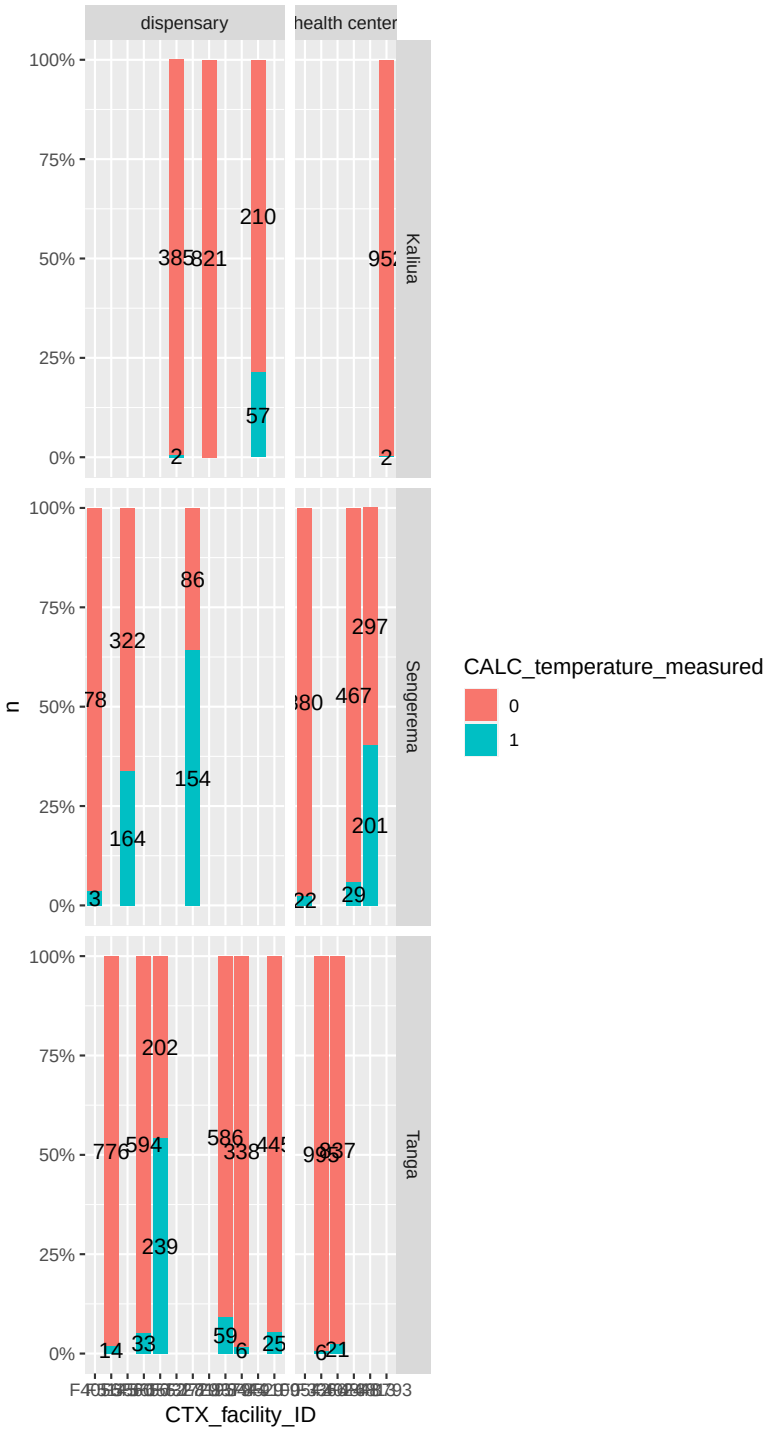
Characteristic	**dispensary**, N = 5,599	**health center**, N = 4,709
RX_antimalarial_parenteral		
0	5,562 (99%)	4,497 (95%)
1	37 (0.7%)	212 (4.5%)
RX_antibiotics		
0	2,407 (43%)	2,471 (52%)
1	3,192 (57%)	2,238 (48%)
MGMT_referral_src_caregiver		
0	5,528 (99%)	4,594 (98%)
1	60 (1.1%)	104 (2.2%)
97	6 (0.1%)	3 (<0.1%)
98	5 (<0.1%)	8 (0.2%)
MGMT_referral_src_registry		
0	5,578 (100%)	4,616 (98%)
1	21 (0.4%)	93 (2.0%)

16.4.1 Temperature measurement

Distribution of temperature measurements and associated factors




100%
75%
50%
25%
0%
n
4843
4428
756
281
dispensary
health center
CTX_facility_type
CALC_temperature_measured
0
1


dispensary
health center
Kaliua
Sengerema
Tanga
n
100%
75%
50%
25%
0%
CALC_temperature_measured
0
1
CTX_facility_ID

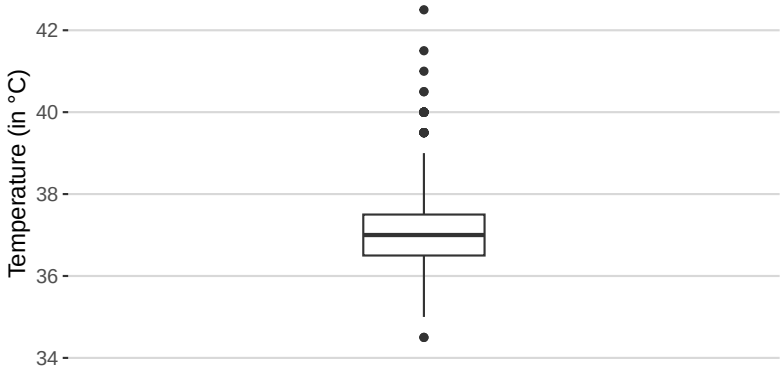


Figure 16.2: Distribution of temperature measurements

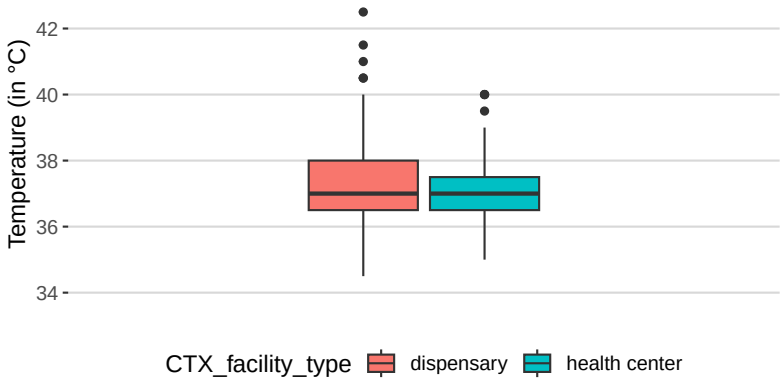


Figure 16.3: Distribution of temperature measurements by type of facility

16.4.2 Fever assessment

Distribution of measured fever

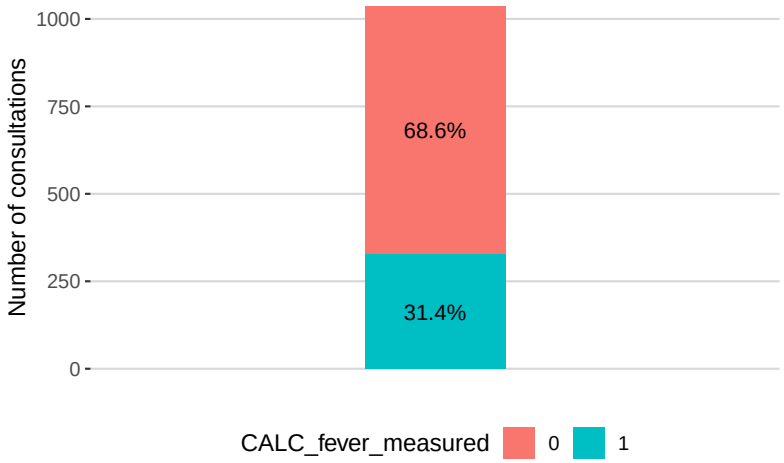
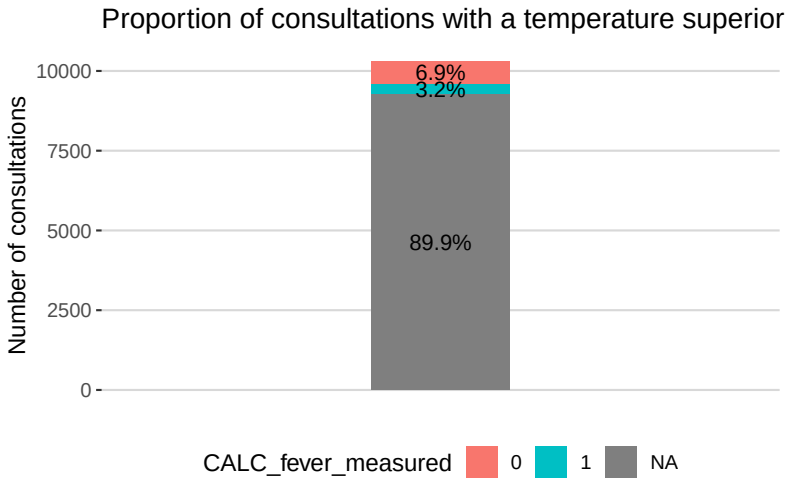


Figure 16.4: Proportion of consultations with a temperature superior or equal to 37.5°C among children with a temperature measurement



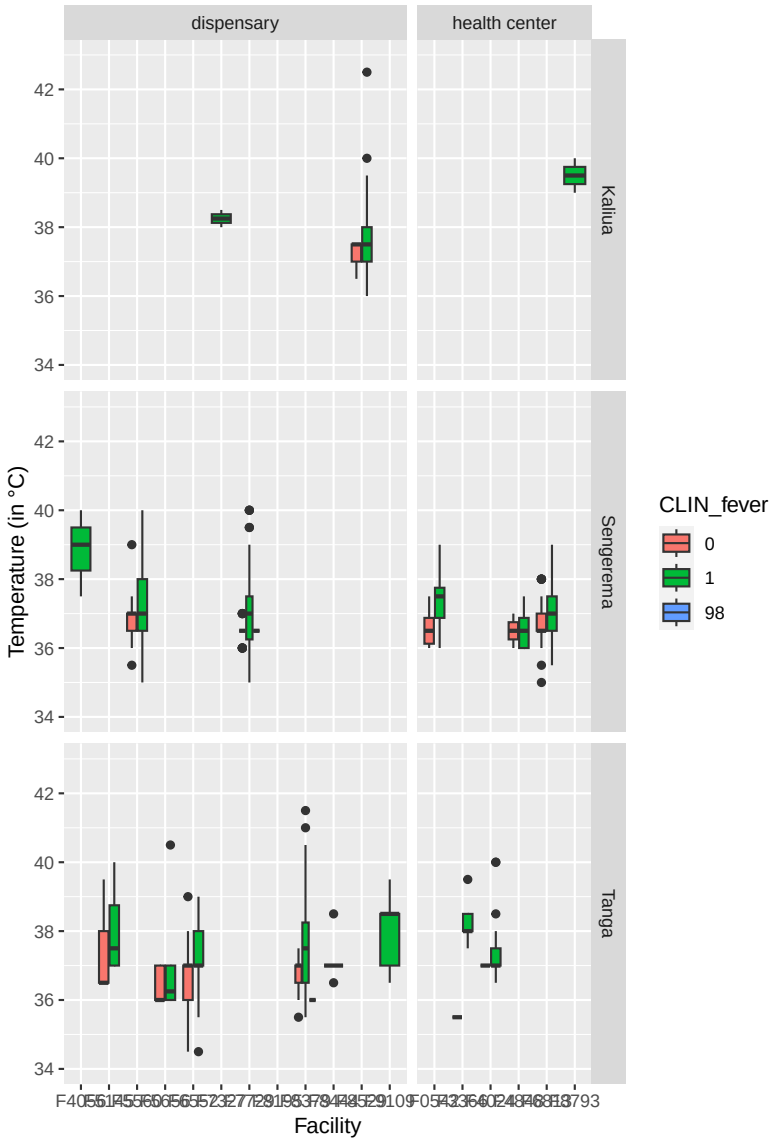
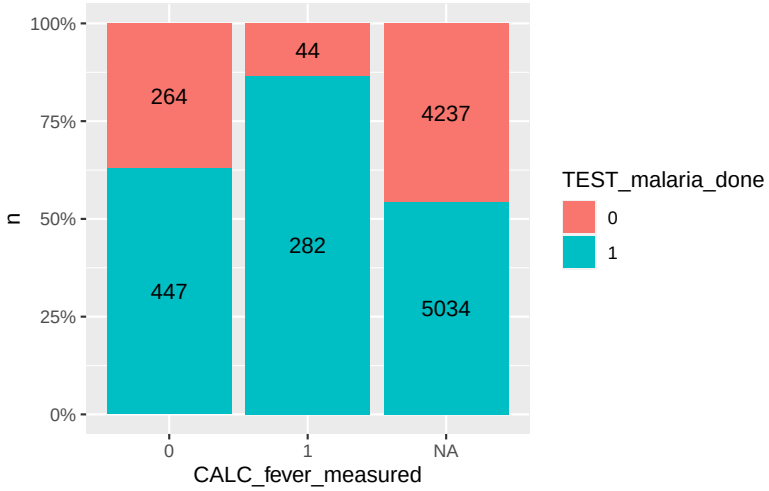


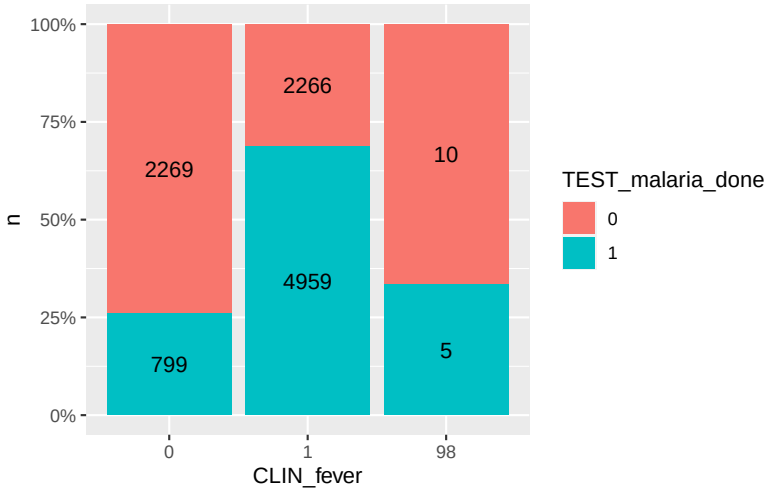
Figure 16.5: Range of temperature measurements for children with fever (1), without fever (2) and with unkwnown fever presentation (98) by facility

Missing data analysis: MEAS_temperature		Not missing	Missing	p
SDC_age_category	<2 months	56 (9.4)	541 (90.6)	0.108
	02-11 months	338 (9.5)	3238 (90.5)	
	12-23 months	293 (9.9)	2654 (90.1)	
	24-35 months	181 (11.8)	1348 (88.2)	
	36-47 months	108 (11.0)	872 (89.0)	
	48-59 months	61 (9.0)	618 (91.0)	
CTX_district	Kaliua	61 (2.5)	2368 (97.5)	<0.001
	Sengerema	573 (21.2)	2130 (78.8)	
	Tanga	403 (7.8)	4773 (92.2)	
CTX_area	rural	454 (11.1)	3634 (88.9)	0.005
	urban	583 (9.4)	5637 (90.6)	
CTX_facility_type	dispensary	756 (13.5)	4843 (86.5)	<0.001
	health center	281 (6.0)	4428 (94.0)	
CLIN_fever	0	238 (7.8)	2830 (92.2)	<0.001
	1	796 (11.0)	6429 (89.0)	
	98	3 (20.0)	12 (80.0)	

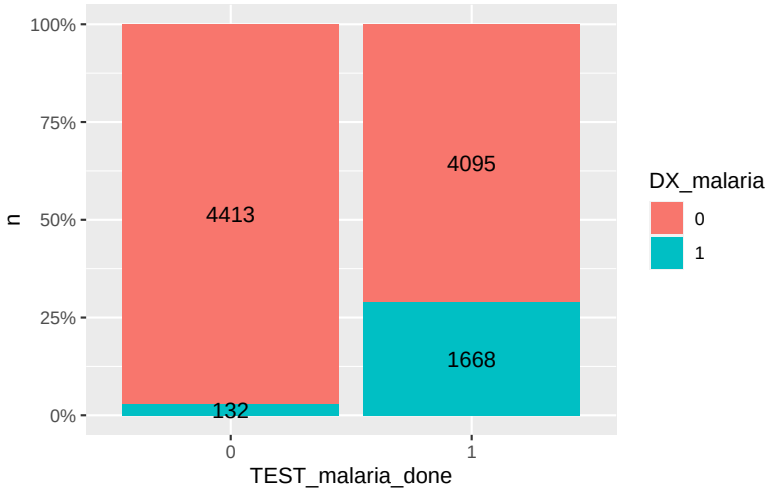
16.4.3 Malaria tests

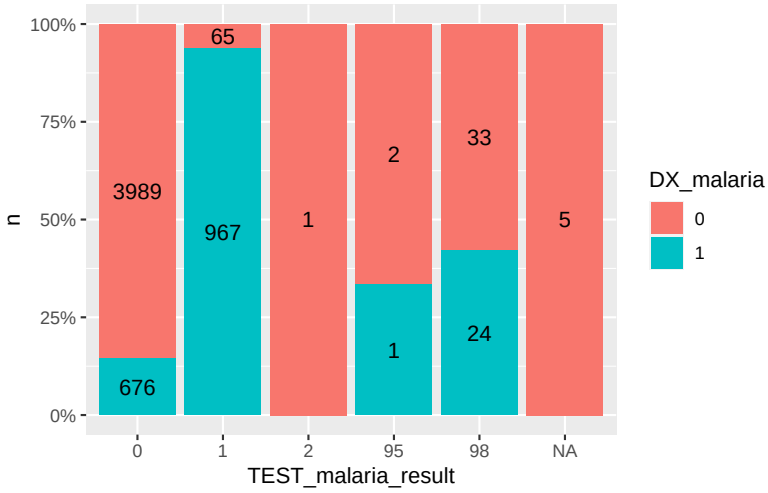
	0	1
0	264	447
1	44	282



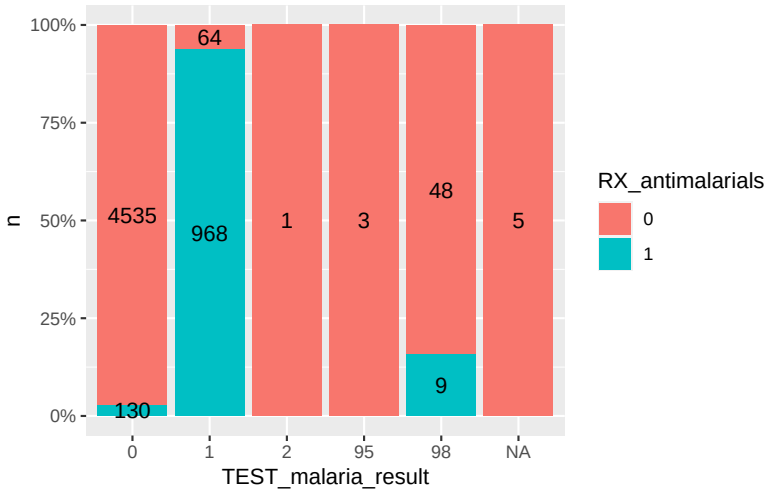


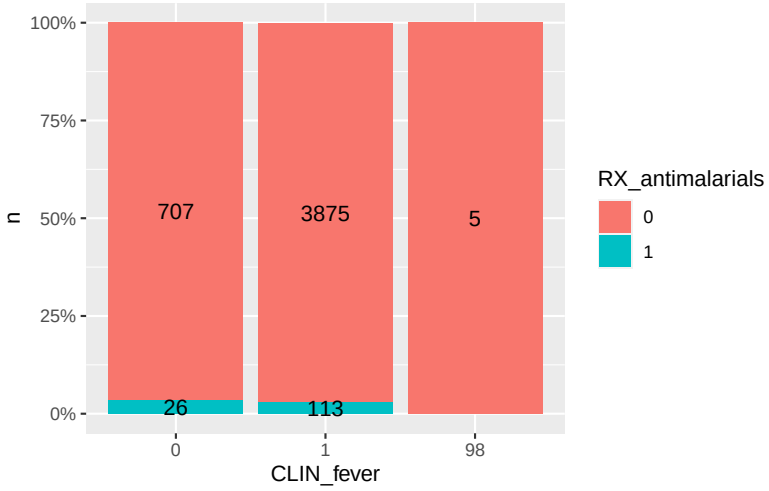
16.4.4 Malaria diagnoses





16.4.5 Malaria treatments

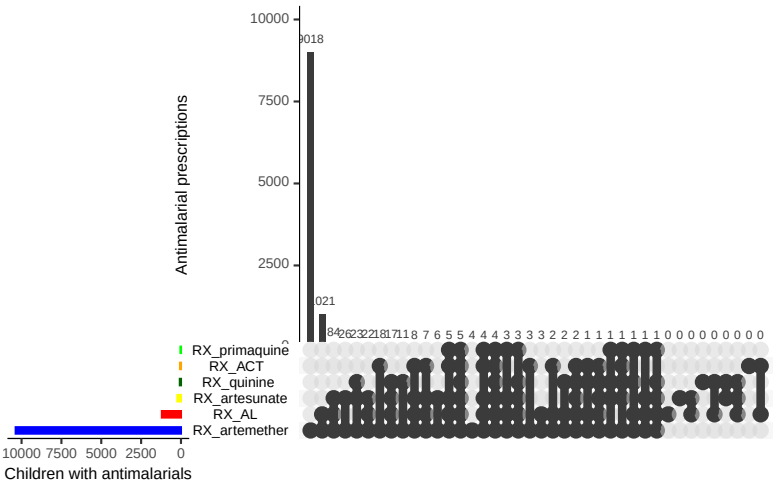




Result	Diagnosis	Antimalarials	Freq
0	0	0	3890
1	0	0	15
2	0	0	1
95	0	0	2
98	0	0	31
0	1	0	645
1	1	0	49
2	1	0	0
95	1	0	1
98	1	0	17

Result	Diagnosis	Antimalarials	Freq
0	0	1	99
1	0	1	50
2	0	1	0
95	0	1	0
98	0	1	2
0	1	1	31
1	1	1	918
2	1	1	0
95	1	1	0
98	1	1	7

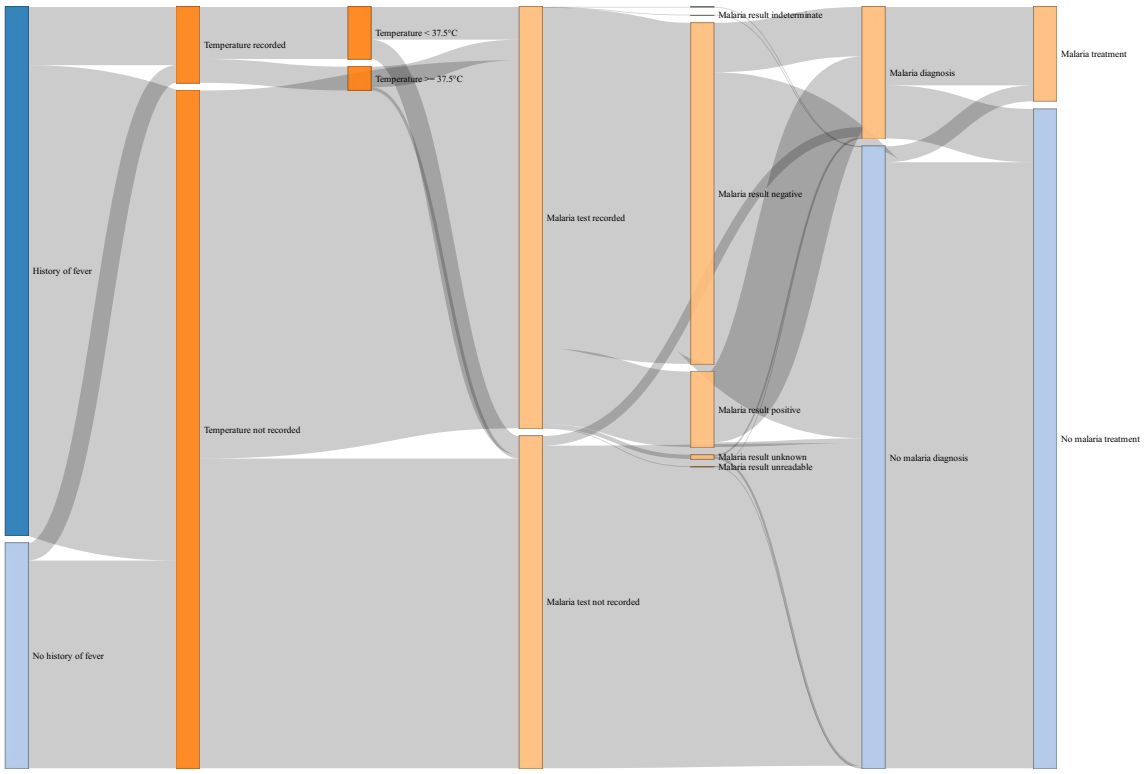
16.4.5.1 Combination of treatments

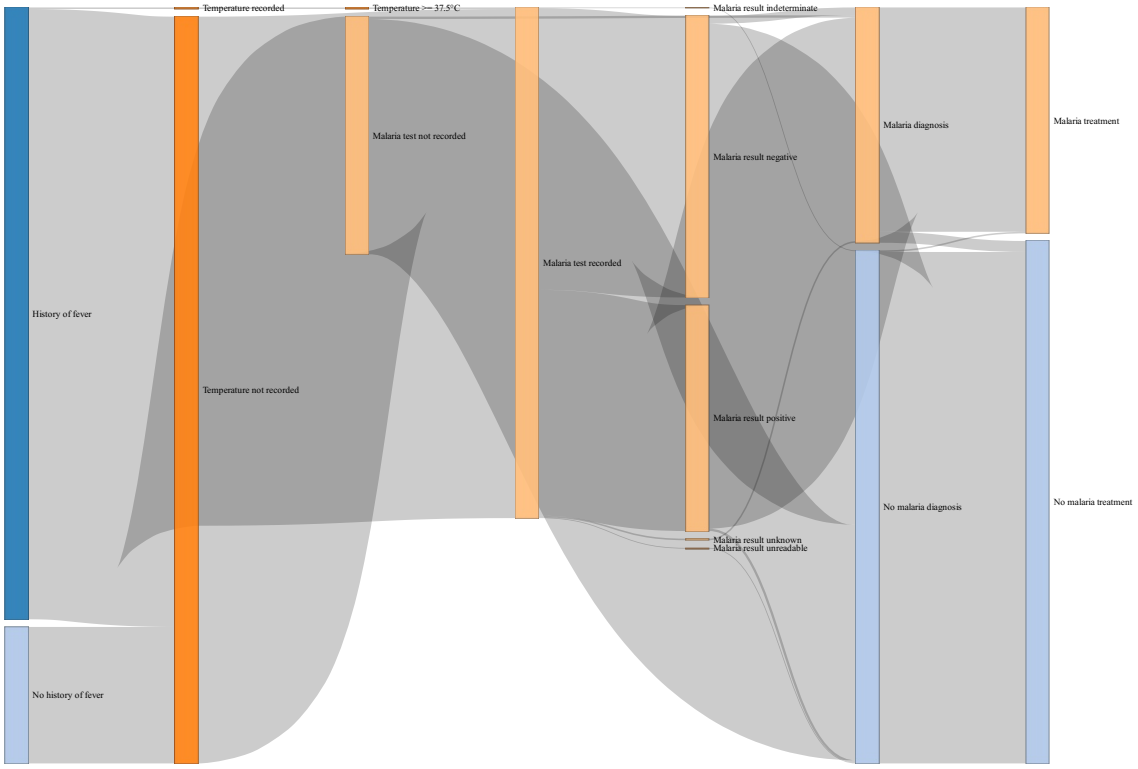


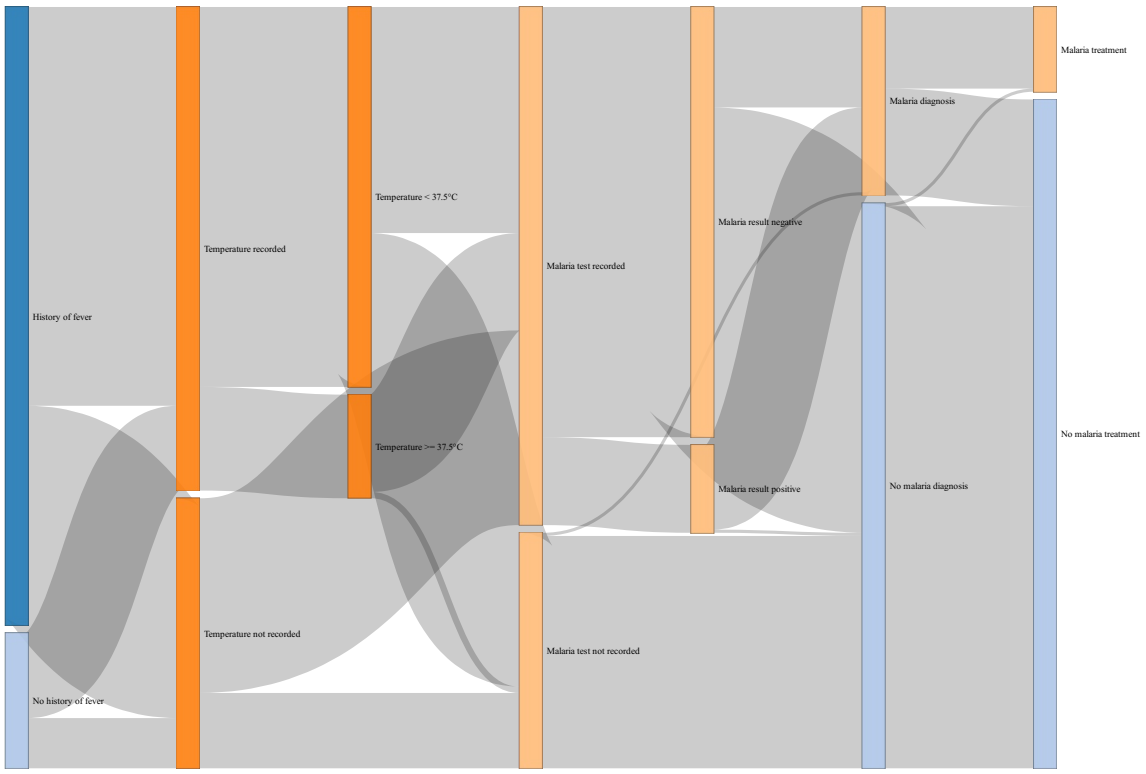
16.5 Data flow from fever assessment to malaria treatment (Sankey diagrams)

16.5.1 Facility F8793

16.5.2 Facility F7729


History of fever
No history of fever
Temperature recorded
Temperature not recorded
Temperature < 37.5°C
Temperature >= 37.5°C
Malaria test recorded
Malaria test not recorded
Malaria result indeterminate
Malaria result negative
Malaria result positive
Malaria result unknown
Malaria result unreadable
Malaria diagnosis
No malaria diagnosis
Malaria treatment
No malaria treatment


Temperature recorded
Temperature >= 37.5°C
Malaria result indeterminate
History of fever
No history of fever
Temperature not recorded
Malaria test not recorded
Malaria test recorded
Malaria result negative
Malaria result positive
Malaria result unknown
Malaria result unreadable
Malaria diagnosis
No malaria diagnosis
Malaria treatment
No malaria treatment


History of fever
No history of fever
Temperature recorded
Temperature not recorded
Temperature < 37.5°C
Temperature >= 37.5°C
Malaria test recorded
Malaria test not recorded
Malaria result negative
Malaria result positive
Malaria diagnosis
No malaria diagnosis
Malaria treatment
No malaria treatment

17 Malaria case study - Report #2

17.1 Introduction

In this section we will explore the impact of the presence of research staff observing consultations as part of a Service Provision Assessment (SPA) on RCT data.

> **Note**
>
> Add sensitivity analysis definition

The data set used for this report is stored in **dataset2.dta**.

17.2 Descriptive statistics

17.2.1 Population characteristics

Characteristic	**All**, N = 10,308	**Observed**, N = 656	**p-value**
SDC_age_category			0.4
<2 months	597 (5.8%)	40 (6.1%)	
02-11 months	3,576 (35%)	206 (31%)	
12-23 months	2,947 (29%)	209 (32%)	
24-35 months	1,529 (15%)	96 (15%)	
36-47 months	980 (9.5%)	67 (10%)	
48-59 months	679 (6.6%)	38 (5.8%)	
SDC_sex			0.3
1	5,229 (51%)	318 (49%)	
2	5,075 (49%)	337 (51%)	

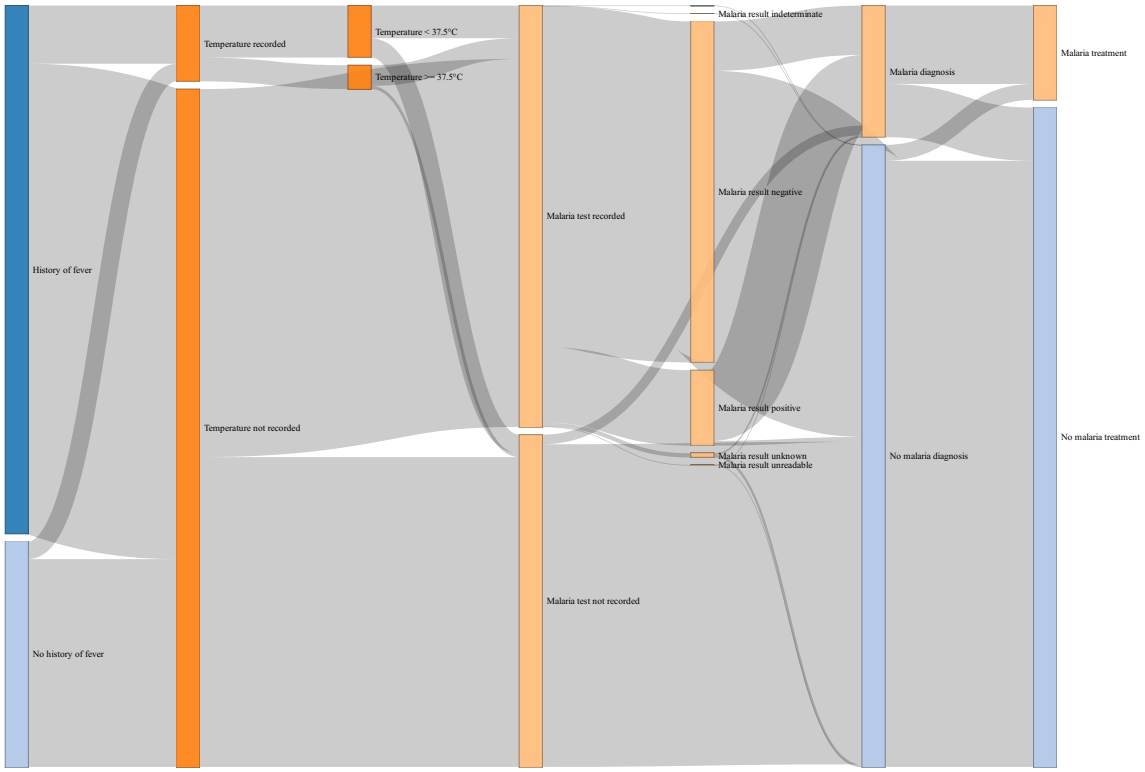
Characteristic	**All**, N = 10,308	**Observed**, N = 656	**p-value**
(Missing)	4	1	
CLIN_fever			>0.9
0	3,068 (30%)	193 (29%)	
1	7,225 (70%)	462 (70%)	
98	15 (0.1%)	1 (0.2%)	
CLIN_fever_onset_category			0.034
	3,083 (30%)	194 (30%)	
<2 days	1,998 (19%)	157 (24%)	
>= 7 days	343 (3.3%)	21 (3.2%)	
2-3 days	4,386 (43%)	248 (38%)	
4-6 days	498 (4.8%)	36 (5.5%)	
CLIN_diarrhoea			0.019
0	7,982 (77%)	477 (73%)	
1	2,306 (22%)	178 (27%)	
98	20 (0.2%)	1 (0.2%)	
CLIN_cough			0.055
0	4,658 (45%)	270 (41%)	
1	5,635 (55%)	384 (59%)	
98	15 (0.1%)	2 (0.3%)	
RX_preconsult_antibiotics			0.049
0	8,573 (83%)	565 (86%)	
1	1,735 (17%)	91 (14%)	
RX_preconsult_antimalarials			0.6
0	9,866 (96%)	631 (96%)	
1	442 (4.3%)	25 (3.8%)	
CTX_district			0.001
Kaliua	2,429 (24%)	149 (23%)	
Sengerema	2,703 (26%)	214 (33%)	
Tanga	5,176 (50%)	293 (45%)	
CTX_area			<0.001
rural	4,088 (40%)	309 (47%)	
urban	6,220 (60%)	347 (53%)	
CTX_facility_type			<0.001
dispensary	5,599 (54%)	412 (63%)	
health center	4,709 (46%)	244 (37%)	

17.2.2 Clinical management

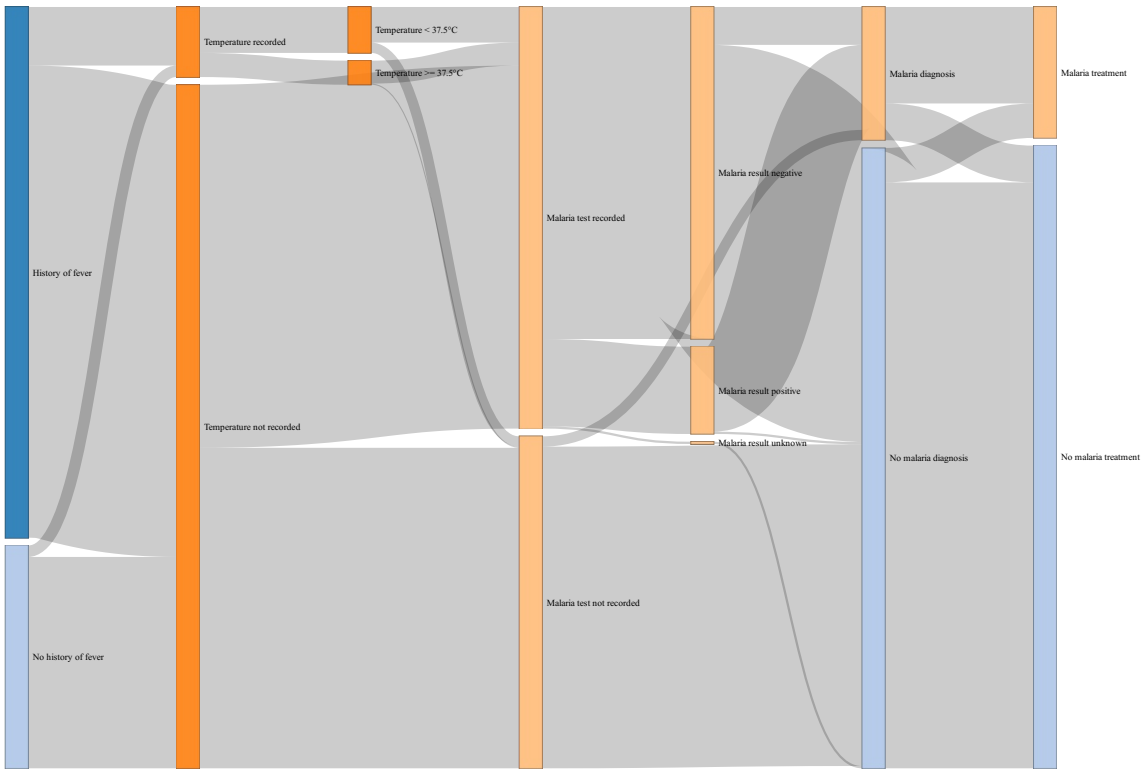
Characteristic	**All**, N = 10,308	**Observed**, N = 656	**p-value**
CALC_temperature_measured			0.5
0	9,271 (90%)	595 (91%)	
1	1,037 (10%)	61 (9.3%)	
CALC_fever_measured			0.6
0	711 (69%)	40 (66%)	
1	326 (31%)	21 (34%)	
(Missing)	9,271	595	
CALC_fever_all			>0.9
0	3,056 (30%)	193 (29%)	
1	7,252 (70%)	463 (71%)	
TEST_malaria_done	5,763 (56%)	367 (56%)	>0.9
TEST_malaria_type			0.4
1	5,371 (93%)	339 (92%)	
2	340 (5.9%)	23 (6.3%)	
95	2 (<0.1%)	0 (0%)	
98	45 (0.8%)	5 (1.4%)	
(Missing)	4,550	289	
TEST_malaria_result			0.5
0	4,665 (81%)	289 (79%)	
1	1,032 (18%)	76 (21%)	
2	1 (<0.1%)	0 (0%)	
95	3 (<0.1%)	0 (0%)	
98	57 (1.0%)	2 (0.5%)	
(Missing)	4,550	289	
DX_severe	254 (2.5%)	10 (1.5%)	0.13
DX_malaria	1,800 (17%)	116 (18%)	0.9
DX_malaria_severe	148 (1.4%)	6 (0.9%)	0.3
RX_antimalarials	1,290 (13%)	114 (17%)	<0.001
RX_antimalarial_parenteral	249 (2.4%)	29 (4.4%)	0.002
RX_antibiotics	5,430 (53%)	351 (54%)	0.7
MGMT_referral_src_caregiver			0.2
0	10,122 (98%)	640 (98%)	
1	164 (1.6%)	14 (2.1%)	
97	9 (<0.1%)	2 (0.3%)	
98	13 (0.1%)	0 (0%)	
MGMT_referral_src_registry	114 (1.1%)	8 (1.2%)	0.8

17.3 From fever assessment to malaria treatment

17.3.1 RCT population



17.3.2 SPA (sub)-population


History of fever
No history of fever
Temperature recorded
Temperature not recorded
Temperature < 37.5°C
Temperature >= 37.5°C
Malaria test recorded
Malaria test not recorded
Malaria result negative
Malaria result positive
Malaria result unknown
Malaria diagnosis
No malaria diagnosis
Malaria treatment
No malaria treatment

18 Malaria case study - Report #3

18.1 Introduction

Data are stored in **dataset3.dta**.

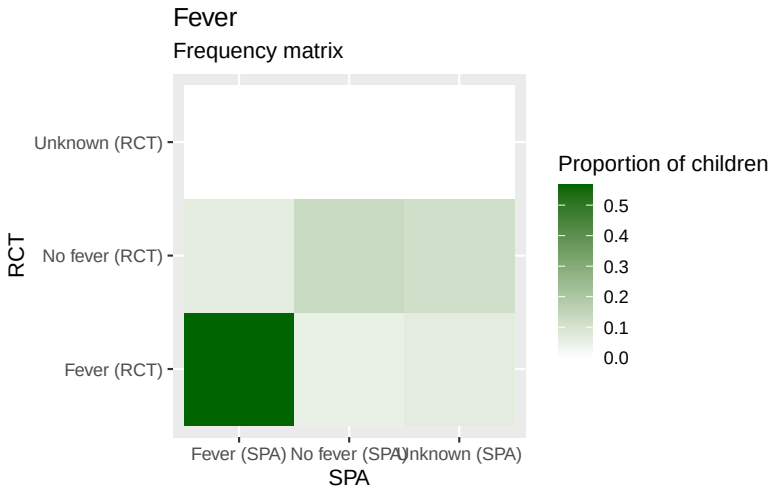
Table 18.1: Extract of the database

RCT_child_ID	RCT_CLIN_fever	SPA_CLIN_fever_reported	CLIN_fever_investigated
1	1	1	NA
2	1	0	0
3	1	1	NA
4	0	0	1
5	1	1	NA
6	0	1	NA
7	1	1	NA
8	0	0	1
9	1	1	NA
10	1	1	NA

18.1.1 Clinical presentation

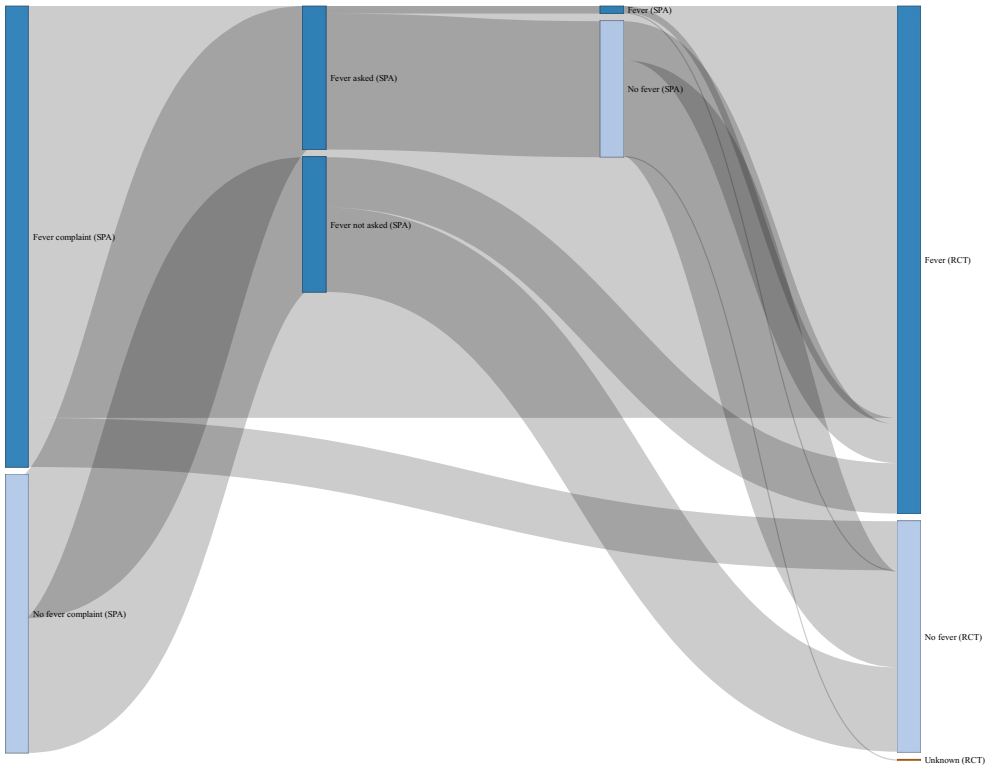
	Fever complaint (SPA)	No fever complaint (SPA)	
Fever (RCT)	327		76
No fever (RCT)	39		144
Unknown (RCT)	0		1

	Fever (SPA)	No fever (SPA)	Unknown (SPA)	
Fever (RCT)	332	31		40
No fever (RCT)	40	76		67
Unknown (RCT)	0	1		0



18.1.2 Temperature measurement

18.1.3 Malaria tests


Fever complaint (SPA)
No fever complaint (SPA)
Fever asked (SPA)
Fever not asked (SPA)
Fever (SPA)
No fever (SPA)
Fever (RCT)
No fever (RCT)
Unknown (RCT)

Part III

——————— DAY 3

19 Big data and machine learning

19.1 Introduction

19.1.1 Overview

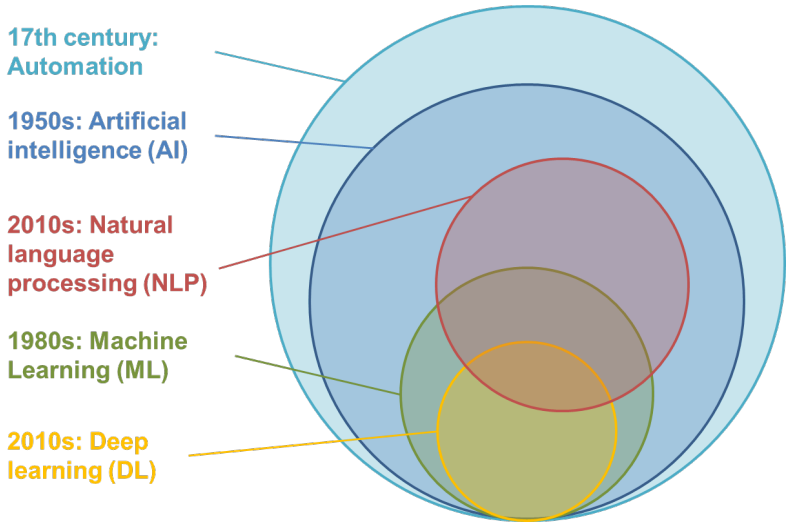
19.1.2 Learning objectives

- key dates and vocabular from the Artificial Intelligence community
- machine learning vs. statistics
- trade-off between accuracy and generalisability

19.2 Big data

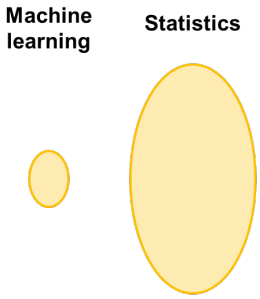
Big data is a combination of structured, semistructured and unstructured data collected by organizations that can be mined for information and used in Big data refers to data sets that are too large or complex to be dealt with by traditional data-processing application software.

19.3 Development of Artificial Intelligence



19.4 Machine Learning vs. Statistics

19.4.1 Inference

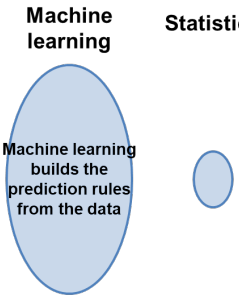


Inference: Use models to learn about the data generation process.

Understand relationship between (observed) input X and (observed) output Y, while adjusting for confounders

Example
What is the main determinant of U5 mortality:
- maternal education level?
- duration of breastfeeding?
- household size?
- child's immunization status?

19.4.2 Prediction



Prediction: Use models to predict outcomes for new observations.

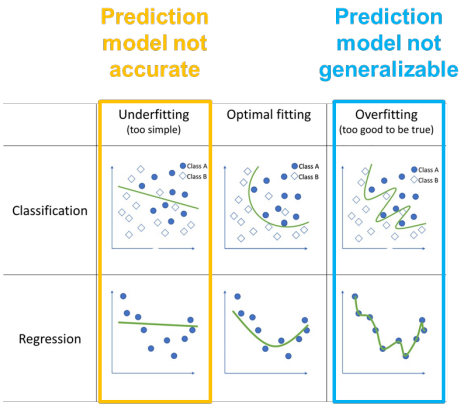
Predict observed output Y from observed input X (**supervised learning**)

Predict unobserved output Y from observed input X (**unsupervised learning**)

Example (supervised)
What is the U5 mortality risk given a certain maternal education level, duration of breastfeeding, household size, child's immunization status?

Example (unsupervised)
Can we discover groups with homogeneous social and economic characteristics ("*cluster*") among which U5 mortality rate is high (resp. low)?

19.5 Learning From Data: Accuracy or Generalizability?



Accuracy = capacity to get accurate (useful) predictions

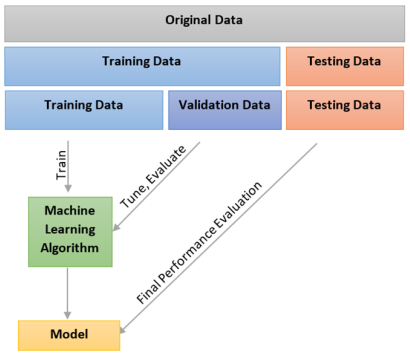
Generalizability = capacity to do predictions on new instances

Many sources of variability + noise (i.e. unexplainable variance)

Solanes Aleix, Radua Joaquim. **Advances in Using MRI to Estimate the Risk of Future Outcomes in Mental Health - Are We Getting There?** Front. Psychiatry, 12 April 2022. Sec. Neuroimaging and Stimulation. https://doi.org/10.3389/fpsyt.2022.826111

19.5.1 Train / test split

We can split the original data into a training set and a testing set.



Reproduced from https://medium.com/analytics-vidhya/only-train-and-test-set-is-not-enough-for-generalizing-ml-model-significance-of-validation-set-cf68bb26881a

Generalizing ≠ Memorizing

"test" data ≠ "train" data

Data on which the predictive model is applied

Data used to learn the predictive model

- Different sampling of noise
- Unobserved combination of features

20 Practical

20.1 Introduction

20.1.1 Overview

This tutorial is adapted from the excellent Machine learning in Python with scikit-learn

20.1.2 Learning objectives

- explore data
- prepare data
- fit a **k-nearest neighbors** model on a training dataset
- evaluate its generalization performance on the testing data

20.2 Question

We are interested in predicting the age of the child based on height and weight measured during the consultation.

- **MEAS_weight_in_kg** and
- **MEAS_height_in_cm**.

```
```{r}
library(tidyverse) # includes dplyr and tibble
library(skimr)
library(ggplot2)
library(DataExplorer)
library(reticulate)
```
```

20.3 Load the data

The dataset is stored in **dataset4.xlsx**.

Read the dataset and store it into a dataframe called **df**.

```
```{r}
df <- openxlsx::read.xlsx("./data/dataset4.xlsx")
```
```

```
```{r}
df <- df %>%
  dplyr::mutate(
    SDC_age_category = dplyr::case_when(
      SDC_age_in_months < 12 ~ "<11 months",
      SDC_age_in_months >= 12 & SDC_age_in_months < 36 ~ "12-35 months",
      SDC_age_in_months >= 36 & SDC_age_in_months < 60 ~ "36-59 months",
      TRUE ~ ""
    )
  ) %>%
  tibble::remove_rownames() %>%
  tibble::column_to_rownames(var="child_ID") %>%
  dplyr::mutate(across(c(SDC_sex,
                         SDC_age_category), factor))
```
```

20.4 Data exploration

We want to do some data exploration to get an initial understanding of the data. Before building a predictive model, it is always a good idea to look at the data:

- maybe the task you are trying to achieve can be solved without machine learning;
- you need to check that the information you need for your task is actually present in the dataset;
- inspecting the data is a good way to find peculiarities. These can arise during data collection (for example, mal-

functioning sensor or missing values), or from the way the data is processed afterwards (for example capped values).

20.4.1 Data structure

20.4.1.1 Exercise 1

Examine the structure of the data, including variable names, labels.

1. How many features are numerical?
2. How many features are categorical?

Display the variables/features **child_id**, **MEAS_weight_in_kg** and **MEAS_height_in_cm** for the **10 first samples** in the data.

```
```{r}
# Write your code here
```
```

20.4.1.2 R

```
```{r}
df %>%
  skimr::skim()
```
```

Table 20.1: Data summary

Name	Piped data
Number of rows	2003
Number of columns	5
_____	
Column type frequency:	
factor	2
numeric	3
_____	
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
SDC_sex	0	1	FALSE	2	1: 1037, 2: 966
SDC_age_category	0	1	FALSE	3	12-: 901, <11: 780, 36-: 322

Variable type: numeric

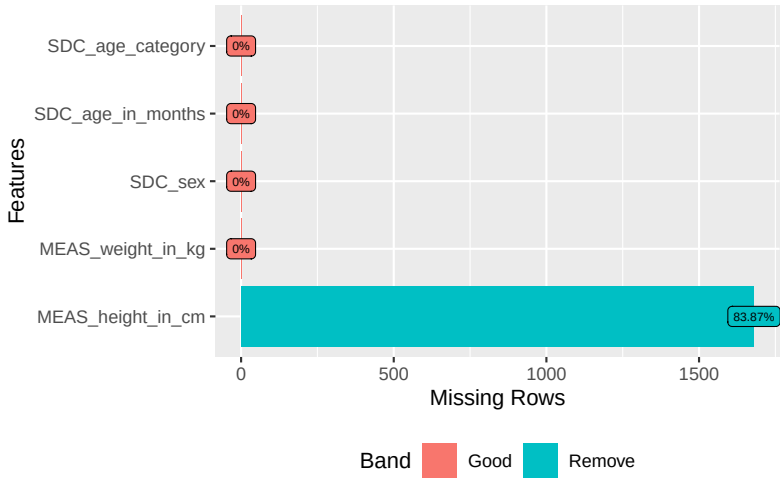
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
MEAS_weight_in_kg	0	1.00	10.25	3.40	2	8	10	12	38.2	
MEAS_height_in_cm	168	0.16	75.61	17.42	10	65	77	89	109.0	
SDC_age_in_months	0	1.00	19.41	14.76	0	8	16	29	59.0	

Numerical variables can be naturally handled by machine learning algorithms that are typically composed of a sequence of arithmetic instructions such as additions and multiplications.

20.4.2 Data preparation

20.4.2.1 Missing data

```
```{r}
DataExplorer::plot_missing(df,
                           geom_label_args = list(size = 2, label.padding = unit(0.2, "lines
```
```



20.4.2.2 Encoding of categorical data

```
sex_df <- df %>%
  dplyr::select(SDC_sex)
```

```
```{python}
from sklearn.preprocessing import OneHotEncoder

encoder = OneHotEncoder(sparse = False)
sex_encoded = encoder.fit_transform(r.sex_df)
```
```

```
df <- cbind(df,
            data.frame(py$sex_encoded)) %>%
  dplyr::rename(sex_male = X1,
                sex_female = X2)
df %>%
  head(10) %>%
  knitr::kable()
```

MEAS_weight	MEAS_height	SDC_sex	SDC_age_in_months	SDC_age_in_months_categories	sex_male	sex_female
150	3	NA	1	1	<11 months	1	0
162	14	NA	1	36	36-59 months	1	0
1177	3	NA	2	0	<11 months	0	1
1245	8	NA	2	19	12-35 months	0	1
1262	16	NA	2	55	36-59 months	0	1
1264	9	NA	1	7	<11 months	1	0
1265	9	NA	1	21	12-35 months	1	0
1266	8	NA	1	6	<11 months	1	0
1268	10	NA	1	13	12-35 months	1	0
1269	13	NA	1	49	36-59 months	1	0

20.4.3 Target classes

20.4.3.1 Exercise 2

What are the different age categories available in the dataset and how many observations/samples of each types are there?

> **Tip**
>
> - R: use `table`
> - Python: select the right column and use the `value_counts` method.

```
```{r}
# Write your code here
```
```

20.4.3.2 R

```
```{r}
table(df$SDC_age_category)
```
```

```
<11 months 12-35 months 36-59 months
       780          901          322
```

20.4.4 Feature distribution

20.4.4.1 Exercise 3

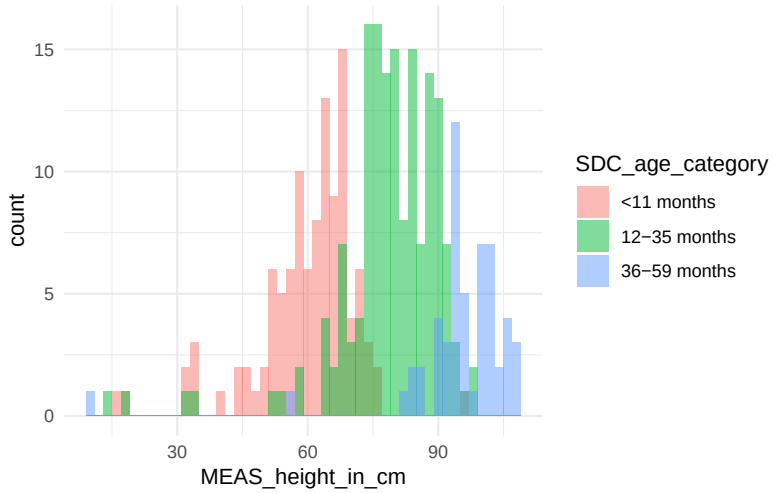
Let us now look at the distribution of individual features, to get more insights about the data.

```
```{r}
# Write your code here
```
```

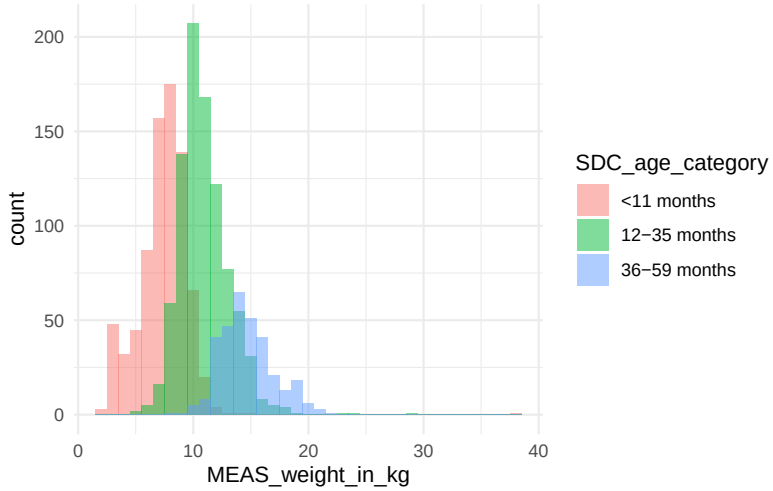
20.4.4.2 R

We can start by plotting histograms, note that this only works for features containing numerical values

```
```{r}
df %>%  ggplot2::ggplot(aes(x = MEAS_height_in_cm,
                            fill = SDC_age_category)) +
  geom_histogram(binwidth = 2, alpha = 0.5, position = "identity") +
  theme_minimal()
```
```

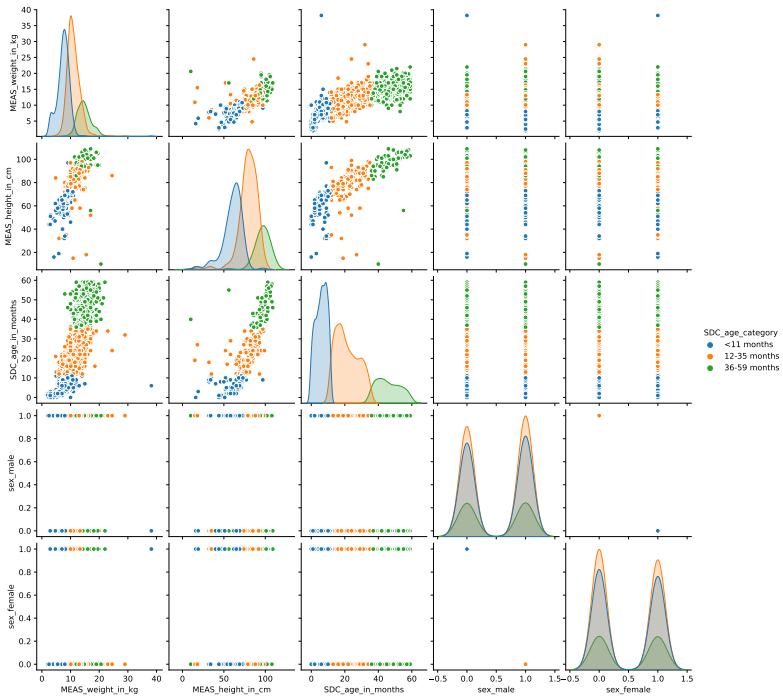



```{r}
df %>%  ggplot2::ggplot(aes(x = MEAS_weight_in_kg,
                            fill = SDC_age_category)) +
  geom_histogram(binwidth = 1, alpha = 0.5, position = "identity") +
  theme_minimal()
```



20.4.4.3 Python

```
```{python}
import seaborn
import matplotlib.pyplot
pairplot_figure = seaborn.pairplot(r.df, hue = "SDC_age_category")
matplotlib.pyplot.show()
```
```



20.4.5 Exercise 3

Show variable/feature distribution for each age category.

Looking at these distributions, how hard do you think it will be to classify the age category only using height and weight?

```
```{r}
# Write your code here
```
```

```
```

Looking at the previous scatter-plot showing height and weight, the age categories are reasonably well separated.

There is some small overlap between the age categories, so we can expect a statistical model to perform well on this dataset but not perfectly.

20.4.6 R

```
```{r}
# Write your code here
```
```

20.5 Train-test data split

When building a machine learning model, it is important to evaluate the trained model on data that was not used to fit it, as generalization is more than memorization (meaning we want a rule that generalizes to new data, without evaluating on data we memorized). The data used to fit a model is called **training** data.

Correct evaluation is easily done by leaving out a subset of the data when training the model and using it afterwards for model evaluation. The data used to assess a model is called **testing** data.

20.5.1 Remove missing data

```
```{r}
df <- df[!is.na(df$MEAS_height_in_cm), ]
df %>%
  head(5) %>%
  knitr::kable()
```
```

	MEAS_weight	MEAS_height	SDC_sex	SDC_age	SDC_age_in_months	SDC_age_category	sex_male	sex_female
5014	11.9	96	1	46	36-59 months	1	0	
5035	9.0	77	2	9	<11 months	0	1	
5037	6.0	65	2	6	<11 months	0	1	
5058	10.0	75	1	17	12-35 months	1	0	
7499	6.0	32	2	18	12-35 months	0	1	

20.5.2 Generate the training / test sampling

Use a seed to make the sampling reproducible (i.e. the same sampling will be generated each time we run this code)

```
```{r}
set.seed(1)
```
```

Create ID column

```
```{r}
df$id <- 1:nrow(df)
```
```

Use 70% of dataset as training set and 30% as test set

```
```{r}
train_df <- df %>%
  dplyr::sample_frac(0.70)
test_df  <- dplyr::anti_join(df,
                             train_df,
                             by = 'id')
```
```

20.5.3 Separate the data and the target

Create the target.

```
```{r}
train_target <- train_df["SDC_age_category"]
test_target <- test_df["SDC_age_category"]
```
```

Remove the target from the training and test dataset to create the data matrix.

```
```{r}
train_data_matrix <- train_df %>%
  dplyr::select(-SDC_age_category,
                -SDC_age_in_months)
test_data_matrix <- test_df %>%
  dplyr::select(-SDC_age_category,
                -SDC_age_in_months)
```
```

20.6 Train the classifier

Let us now use a nearest neighbour approach for learning the target from the training data matrix of weights and heights. The principle behind **nearest neighbor methods** is to find a predefined number of training samples closest in distance to the new point, and predict the label from these.

Despite its simplicity, nearest neighbors has been successful in a large number of classification and regression problems. Supervised neighbors-based learning comes in two flavors: * classification for data with discrete labels * regression for data with continuous labels.

```
```{python}
from sklearn.neighbors import KNeighborsClassifier

model = KNeighborsClassifier()
```
```

```
_ = model.fit(r.train_data_matrix, r.train_target.values.ravel())
```

20.7 Evaluate the performance of the classifier

```{python}
target_predicted = model.predict(r.test_data_matrix)
```

Confusion matrix

```{r}
#| df-print: kable
combined <- cbind(test_target, py$target_predicted)
colnames(combined) <- c("target", "prediction")
combined <- combined %>%
  dplyr::mutate(correct = 1 * (target == prediction))
combined %>%  head(5)
```

	target	prediction	correct
5035	<11 months	12-35 months	0
5037	<11 months	12-35 months	0
7499	12-35 months	<11 months	0
7994	36-59 months	12-35 months	0
7995	36-59 months	12-35 months	0

```{r}
perf <- mean(combined$correct)
```

Number of correct prediction: 0.7216495

```{python}
accuracy = model.score(r.test_data_matrix, r.test_target)
model_name = model.__class__.__name__
```