

EsqlabsR Reports Tutorial

`{esqlabs.Reports}`, provides Esqlabs' report template templates that match the company's quality standards and design with minimal effort. These templates relies on Quarto, which is multi-language, next-generation version of R Markdown from Posit. This tutorial will help getting started with Quarto, provide some tips and tricks and details on how the esqlabs templates work.

esqLABS GmbH

2023-09-20 14:10:25

Table of Content

1	About Quarto Documents	2
2	About Markdown	2
3	Report Header	2
3.1	Report Metadata	2
3.2	Report Parameters	2
3.3	Advanced Report Settings	3
4	Elements	3
4.1	Figures	3
4.1.1	Plot a figure	3
4.1.2	Plot all figures in a directory	3
4.2	Display Table	4
4.2.1	Using Markdown syntax	4
4.2.2	From a file	5
4.3	Callout boxes	5
5	Source External Elements	5
5.1	Source Code	5
5.2	Source Text	6

1 About Quarto Documents

Quarto documents allow to mix text and code to produce beautiful, reproducible reports.

This file is Quarto document with the extension `.qmd`. It set to be rendered as a `.pdf` file. But you could also choose to render it into other formats like HTML, MS Word or other...

All information about Quarto can be found [here](#).

2 About Markdown

Quarto is based on Pandoc and uses its variation of markdown as its underlying document syntax. Pandoc markdown is an extended and slightly revised version of John Gruber's Markdown syntax.

Markdown is a plain text format that is designed to be easy to write, and, even more importantly, easy to read:

A Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. – John Gruber

Using Markdown, you can:

- format text,
- insert images,
- build tables,
- write equations using LaTeX syntax,
- Draw diagrams,
- and many more ...

[This page](#) provides examples of the most commonly used markdown syntax.

3 Report Header

3.1 Report Metadata

An (optional) YAML header demarcated by three dashes (`---`) on either end at the top of the document. The basic syntax of YAML uses key-value pairs in the format **key: value**. Use that to customize the report's header including title, subtitle, author and date.

3.2 Report Parameters

Parameters defined in the header of the documents can be accessed by R in the `params` object.

It can be character, numeric, integer, and logical.

```
params:
  run_script: yes
  client_name: "Very Good Very Nice Inc"
  days_before_Xmax: !expr as.Date(paste0(format(Sys.Date(), "%Y"), "-12-25")) - Sys.Date()
```

```
params$client_name
```

```
[1] "Very Good Very Nice Inc."
```

It is also possible to include R code directly in parameters using the `!expr` tag.

Parameters can be used inline to produce dynamic text:

```
Only **`r as.integer(params$days_before_Xmax)`** days before Christmas !
```

Only **96** days before Christmas !

3.3 Advanced Report Settings

Other metadata can be accessed and modified in the `_quarto.yml` file.

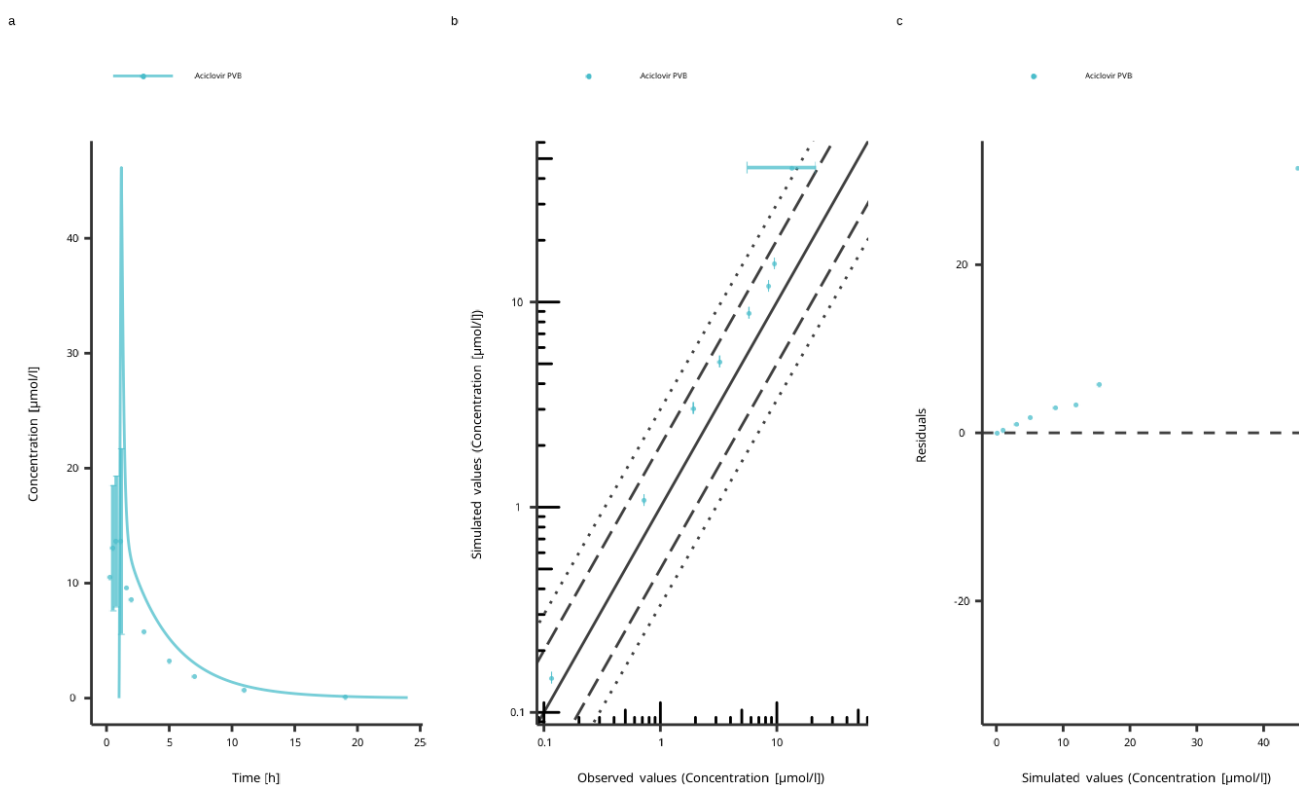
4 Elements

4.1 Figures

4.1.1 Plot a figure

```

```



4.1.2 Plot all figures in a directory

This needs to be setup in two different code chunks

```
# list files we want to display
figs <- list.files("figures/", pattern = "\\.[pj][pn]g", full.names = TRUE)
# Extract the name of the plot from the files names
figs_names <- fs::path_ext_remove(fs::path_file(figs))
```

```
#| fig-cap: !expr figs_names
knitr::include_graphics(figs)
```

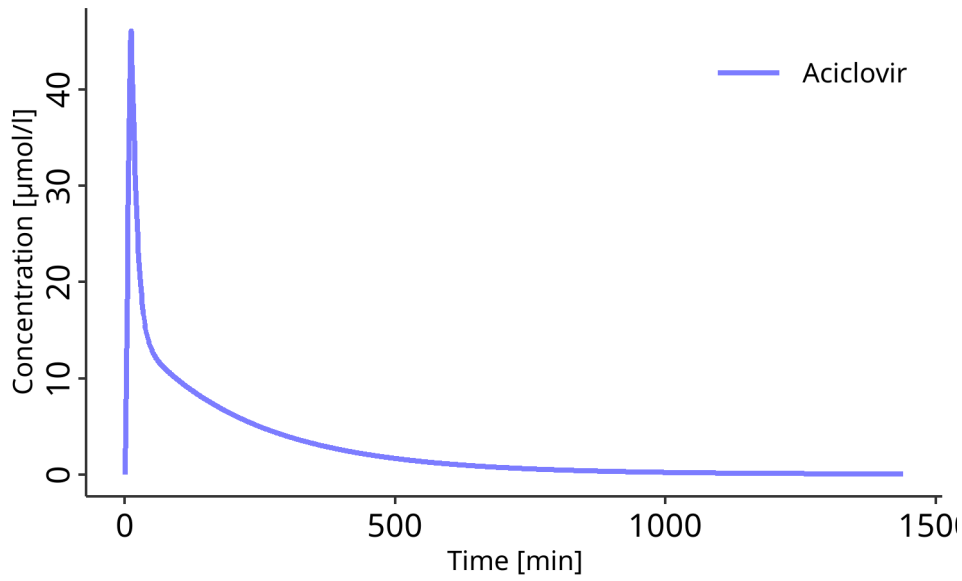


Figure 1: Aciclovir Concentration Time Profile

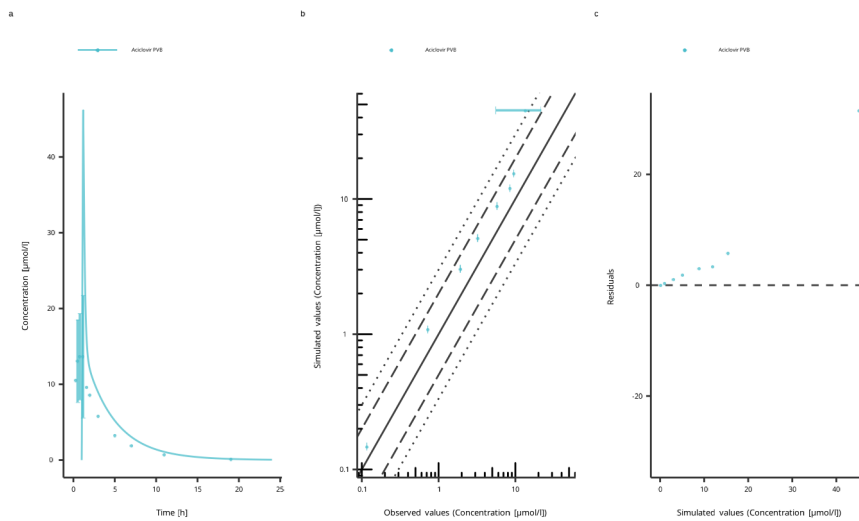


Figure 2: Aciclovir Plots

More information about figures [here](#)

4.2 Display Table

4.2.1 Using Markdown syntax

You can display simple tables by building directly using markdown:

Default	Left	Right	Center
12	12	12	12
123	123	123	123

```
| 1 | 1 | 1 | 1 |
```

: Demonstration of pipe table syntax

Default	Left	Right	Center
12	12	12	12
123	123	123	123
1	1	1	1

Table 1: Demonstration of pipe table syntax

4.2.2 From a file

IndividualId	Gender	Population	Organism.Weight	Organism.BMI
0	MALE	European_ICRP_2002	61.53480	0.2211044
1	MALE	European_ICRP_2002	62.96552	0.1965402

4.3 Callout boxes

```
 ::: {.callout-note}
Text can be displayed in callout boxes to draw reader's attention.
 :::
```

i Note

Like this

To generate callout grammatically use the following structure

```
if (params$run_script) {
  cat(
    " ::: {.callout-warning}
    `run_script` is set to `TRUE` !
    ::: "
  )
}
```

! Warning

run_script is set to TRUE !

More information about callouts [here](#)

5 Source External Elements

5.1 Source Code

It is often cleaner to define functions and logic in separate code files. for this, use the **scripts/** folder and source the files there

```
source("scripts/say-hello.R")
```

```
[1] "Hello !"
```

```
say_hello(params$client_name)
```

```
[1] "Hello Very Good Very Nice Inc. !"
```

5.2 Source Text

In some case, it can be useful to insert text from another file.

This text was written in `source_text.md` and yet, is included in the main document using `{{< include texts/source_text.md >}}`.