# Quarto Analysis Book Template

Author 1 Author 2

2023-08-16

# Table of contents

# Quarto Analysis Book Template

The Quarto Analysis Book template is designed to streamline the preparation and publication of scientific manuscripts and reports. This platform facilitates collaboration among multiple users and integrates various data science tools, including R and Python, within a single environment. The organized file management structure ensures easy navigation and readability, allowing users to efficiently transition between preliminary analyses, testing, and the final manuscript.

See README for instructions on use.

# Part I

# Loose Analyses

## 1 Feature Analysis

## Part II

# Manuscripts

# 2 Manuscript Example

see Quarto basics for examples

# Part III

# Abstracts

# 3 Abstract Example

see Quarto basics for examples

# Part IV

# Grants

# 4 Grant Example

see Quarto basics for examples

## References

# A Python Environment

This guide will help you set up a Python environment for your project using micromamba and configure Jupyter Notebook to work with the created environment.

## A.1 Project-specific information

```
PREFIX=/srv/Analysis/quarto_analyses
ENV_NAME=forge
```

## A.2 Quick Start

1. After logging in, go to the `setup` subfolder in the Quarto book folder.
2. Run:

```
./setup_mm.sh
```

3. To start Jupyter Lab run and open a browser:

```
./start_lab.shell
```

- a link called `srv` will exist in the default notebook folder that links to all server files.
- To add python modules, edit `conda.yml` and rerun `./setup_mm.sh`

4. To use python in RStudio, open RStudio and create a new python file. A variable `RETICULATE_PYTHON` was added to the `.Renviron` to automatically load when the python interpreter activates.

## A.3 Setting up a new project

1. After creating your quarto book directories, edit the project.sh to modify the variable names. The script will setup a new virtual environment in the PREFIX directory with the same name as the Quarto book project.

## A.4 File information

1. `project.sh`: This file specifies the `PREFIX` and `ENV_NAME` variables. `PREFIX` is the root directory of the Python environments, and `ENV_NAME` is the name of the environment.
2. `conda.yml`: This file can be modified to specify the name of the environment to place the packages into and what packages to install.

### A.4.1 Creating the Python environment

1. **Micromamba setup script**: A script called `setup_mm.sh` is located in the `project_name/setup` directory. This script is responsible for downloading and installing a fresh copy of micromamba and setting up a Python environment.
2. **Environment specifications**: The `conda.yml` file contains the specifications for the Python environment, including the packages to be installed.
3. **Setting the RETICULATE_PYTHON environment variable**: The `setup_mm.sh` script also sets an environment variable for R's `reticulate` package to know which Python environment to use. You can run this command anytime you log in to a shell to ensure the environment is up to date.

# B Useful tools

Software installation and configuration routines that do not run by default. They may be used when setting up a new system or repairing a configuration.

Modify `#| eval: false` tag to `#| eval: true` to activate a section. These are `bash` blocks that should execute on Linux systems. You must use `Render` action, not `Run: Code Block`.

## B.1 Install certificates for internet access

```
sudo cp /srv/licenses/*.crt /usr/local/share/ca-certificates/
sudo update-ca-certificates
```

## B.2 GPT Studio

https://github.com/MichelNivard/gptstudio

```
pacman::p_load("gptstudio")
```

## B.3 Midnight Commander

Commandline tool for file management.

```
# echo 'Acquire::https::Verify-Peer "false";' | sudo tee /etc/apt/apt.conf.d/99disable-cert-
sudo apt-get update && sudo apt-get install mc
```

## B.4 To use the Jetbrains Mono font in RStudio, follow these steps:

1. Download the font from the JetBrains website (https://www.jetbrains.com/lp/mono/).
2. Install the font on your operating system by double-clicking the downloaded file and following the installation instructions.
3. Open RStudio.
4. Go to "Tools" -> "Global Options" -> "Appearance".
5. In the "Editor font" section, click on the "…" button.
6. Select "JetBrains Mono" from the font list.
7. Adjust the font size and other preferences if needed.
8. Click "OK" to save the changes.

After following these steps, RStudio will use the Jetbrains Mono font in the script editor, console, and other parts of the IDE. The ligatures of the font should also work automatically.

```
cd /tmp
wget https://download.jetbrains.com/fonts/JetBrainsMono-2.304.zip
unzip JetBrainsMono-2.304.zip
mkdir -p ~/.local/share/fonts
mv fonts/JetBrainsMono-*.ttf ~/.local/share/fonts/
fc-cache -f -v
rm -rf JetBrainsMono-2.304.zip fonts OFL.txt AUTHORS.txt
```

## B.5 GitKraken

Linux graphical client for git

```
cd /tmp

# Download GitKraken .deb package
wget -O gitkraken.deb https://release.gitkraken.com/linux/gitkraken-amd64.deb

# Install GitKraken using the downloaded .deb package
sudo dpkg -i gitkraken.deb
sudo apt-get install -f

# Remove the downloaded .deb package
rm gitkraken.deb

# Define the content of the .desktop file
desktop_file_content='[Desktop Entry]
```

```
Name=GitKraken
Comment=GitKraken Git Client
GenericName=GitKraken
Exec=gitkraken --no-sandbox
Icon=gitkraken
Type=Application
StartupNotify=true
Categories=GNOME;GTK;Utility;Development;'

# Create the .desktop file on the Desktop
echo "$desktop_file_content" > ~/Desktop/GitKraken.desktop

# Make the .desktop file executable
chmod +x ~/Desktop/GitKraken.desktop
```