mulang

Table of contents

1 1 Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

2 2 Running Code

Quarto files using the jupyter engine cannot mix languages. 1. Create one Quarto file for each language and one `main.qmd`: - `main.qmd` - does not have any code chunks, only prose and embed shortcodes - has the engine option set to knitr in its YAML header - the other Quarto files - do not have any prose, only code - have the YAML header engine option set to jupyter and the relevant language 2. Create Jupyter notebooks from the Quarto files 3. Keep them all in sync with one of the following options a. Convert the Quarto files into Jupyter notebooks and work in the notebooks b. Output the Quarto files to Jupyter notebooks 3. Embed cells from the Jupyter notebooks in `main.qmd` 4. Generate all output files from `main.qmd`

1. Set the engine option to knitr in the main Quarto file YAML header
2. Use Python, R, and bash code in their respective chunks
3. Eventually move code into scripts and Jupyter notebooks to keep the quarto file small

a. Julia code must be in Jupyter notebooks
b. Embed cells from the Jupyter notebooks in the Quarto file
c. Bring in scripts with the file chunk option Keep Python, R, and Julia code in separate scripts

1. Keep Python, R, and Julia code in separate scripts
2. Use Quarto notebooks that are sync'ed to Jupyter Notebooks with the correct kernel setting.
3. Embed cells from those notebooks in a combined Quarto notebook Use Jupyter for

▼ Code

```{r}
#| tags: [pairs]
ma <- as.matrix(iris[, 1:4])
pairs(ma, col = rainbow(3)[iris$Species])
```

Another test

▼ Code

````{r}
#| tags: [ggpairs]
# install.packages("GGally")
library(GGally)

ggpairs(iris,                   # Data frame
        columns = 1:4,          # Columns
        aes(color = Species,    # Color by group (cat. variable)
            alpha = 0.5))       # Transparency
````

Will this be deleted?

▼ Code

````{python}
#| tags: [pairplot]
import seaborn as sns
import matplotlib.pyplot as plt
iris = sns.load_dataset("iris")
sns.pairplot(iris, hue="species")
````

Test

▼ Code

````{julia}
#| tags: [corrplot]
#| eval: false
using StatsPlots, RDatasets
iris = RDatasets.dataset("datasets", "iris")
@df iris corrplot(cols(1:4), grid = false)
````