A Jupyter Notebook with my incredible R code

José R. Ferrer-Paris

2023-12-08

[Project Jupyter](https://jupyter.org/) develops open-source software, open standards, and services for interactive computing across multiple programming languages.

Jupyter supports execution environments (called “kernels”) in several dozen languages, including Julia, R, Haskell, Ruby, and Python.

[Quarto](https://quarto.org) is an open-source scientific and technical publishing system. It enables you to weave together content and executable code into a finished document.

Quarto can use Jupyter notebooks to create dynamic content or reproducible documents.

When you render a Jupyter notebook with Quarto, the contents of the notebook (code, markdown, and outputs) are converted to plain markdown and then processed by Pandoc, which creates the finished format.

|  |
| --- |
| Quarto workflow with jupyter notebook |

## This is my great R code

### Load the libraries

We need magick to read the images, ggplot2 for plotting and and cowplot for arranging the grid of plots and images.

library(ggplot2)  
library(cowplot)  
library(magick)

Linking to ImageMagick 6.9.12.3  
Enabled features: cairo, fontconfig, freetype, heic, lcms, pango, raw, rsvg, webp  
Disabled features: fftw, ghostscript, x11

### Locate the images

We use the functions from package here to establish our project root, and then use the relative paths to the images

here::i\_am("presentation/showcase-your-research.qmd")  
yes\_img <- here::here("images/yes.png")  
no\_img <- here::here("images/no.png")

here() starts at /Users/z3529065/proyectos/codeRs/showcase-your-research

We use functions from cowplot to set up drawing layers in top of blank ggplot2 canvas. This requires the magick package to read and manipulate the images internally.

yes\_plot <- ggdraw() + draw\_image(yes\_img, scale = 1)  
no\_plot <- ggdraw() + draw\_image(no\_img, scale = 1)

### This is a bad plot

Sometimes we do silly mistakes when we are learning to use a package. This is one example:

(bad\_plot <-   
 ggplot(mpg, aes(x = displ, y = hwy, colour = "blue")) +  
 geom\_point())

|  |
| --- |
| This is not the way to use the colour aesthetic |

### This is a good plot

Adding layers of ggplot2 functions allow us to customise plots and make them ready for publication.

(good\_plot <-   
 ggplot(mpg, aes(x = displ, y = hwy)) +  
 geom\_point(colour = "blue") +  
 geom\_smooth(colour = "red") +   
 theme\_half\_open(12) +  
 labs(x = "Engine power (litres displ.)",  
 y = "Fuel Efficiency (miles/gallon)"))

`geom\_smooth()` using method = 'loess' and formula = 'y ~ x'

|  |
| --- |
| This plot is so nice! |

### Conclusion

plot\_grid(no\_plot, bad\_plot,  
 yes\_plot, good\_plot)

`geom\_smooth()` using method = 'loess' and formula = 'y ~ x'

|  |
| --- |
| This grid shows the progress from novice to expert in ggplot2 |

## Render this document

Open a terminal in the same directory as this file and then type:

* quarto render Example.ipynb --to html for html output
* quarto render Example.ipynb --to docx for docx output