

NOTEBOOKS

CODE NOTEBOOKS

In order to make analysis **practically reproducible**, one should strive to make analysis

1. easy to interact with
2. easy to understand

code notebooks help achieve these goals via a **literate programming** format that interweaves

1. text
2. code
3. output

all together.

POPULAR NOTEBOOKS AND SOFTWARE

Two most popular notebook formats/software:

1. “jupyter”: **jupyter lab** software and **.ipynb** format
2. “quarto”: **Rstudio** software and **.qmd** format

JUPYTER LAB EXAMPLE

An example of jupyter lab:

The screenshot shows the Jupyter Lab interface with several annotations:

- A red box labeled "rich text" highlights the toolbar icon for rich text.
- A red arrow points from the "rich text" annotation to the "rich text" icon in the toolbar.
- A red box labeled "code" highlights the code cell containing R code.
- A red arrow points from the "code" annotation to the "code" icon in the toolbar.
- A red box labeled "suppressed output" highlights the output cell containing the command `penguins %>% sample_n(3)`.
- A red arrow points from the "suppressed output" annotation to the "suppressed output" icon in the toolbar.
- A red box labeled "output" highlights the data frame displayed below the output cell.
- A red arrow points from the "output" annotation to the "output" icon in the toolbar.

The Jupyter Lab interface includes a sidebar with file, folder, and settings icons, a top bar with tabs, file menu, and zoom controls, and a bottom status bar showing mode, line, column, and file information.

```
[1]: library("palmerpenguins")
library('tidyverse')
...
[2]: penguins %>% sample_n(3)
```

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
<fct>	<fct>	<dbl>	<dbl>	<int>	<int>	<fct>	<int>
Adelie	Torgersen	42.5	20.7	197	4500	male	2007
Adelie	Biscoe	41.1	18.2	192	4050	male	2008

TEXT IN `markdown`

jupyter allows text to be written in `markdown` which is a light-weight markup language.

More-or-less: if you can display it on a webpage, you can write it in `markdown`. (Additionally, one can directly embed `html`)

One can also use extended markdown languages like `myst` which enables features like references, figures, bibliographies, ...

In any case, jupyter's `markdown` enables rich-text commentary on **both the code and the output**

(In fact, this presentation is written in `markdown`)

BASIC MARKDOWN

The screenshot shows a Jupyter Notebook interface with the title bar "intro_to_jupyter.ipynb" and the URL "localhost:8889/lab/tree/intro_to_jupyter.ipynb". The menu bar includes File, Edit, View, Run, Kernel, Tabs, Settings, and Help. The toolbar features icons for file operations like Open, Save, and New, along with Code, Git, and R buttons. A sidebar on the left contains icons for File, Cell, Cell Type, and Cell Kernel. The main content area displays the following text:

```
headings can be written using successive indentation by # , e.g.  
# heading level 1  
## heading level 2  
### heading level 3
```

Below this, three headings are shown:

- heading level 1
- heading level 2
- heading level 3

The bottom status bar shows "Simple" mode, 0 cells, R | Idle, Mode: Command, and the file path "intro_to_jupyter.ipynb".

BASIC MARKDOWN

The screenshot shows a Jupyter Notebook interface with the title bar "intro_to_jupyter.ipynb" and the URL "localhost:8889/lab/tree/intro_to_jupyter.ipynb". The menu bar includes File, Edit, View, Run, Kernel, Tabs, Settings, and Help. The toolbar features icons for file operations like Open, Save, and New, along with Raw, git, and R mode buttons. The main content area displays the following Markdown examples:

```
**bold text** displays bold text
*italic text* displays italic text
> block quote
    block quote
```

The bottom status bar shows "Simple" mode, cell count 0, and "R | Idle". It also indicates "Mode: Command", line 1, column 38, and the file name "intro_to_jupyter.ipynb".

BASIC MARKDOWN

The screenshot shows a Jupyter Notebook interface with the title bar "intro_to_jupyter.ipynb" and the URL "localhost:8889/lab/tree/intro_to_jupyter.ipynb". The menu bar includes File, Edit, View, Run, Kernel, Tabs, Settings, and Help. The toolbar features icons for file operations like save, new, cut, copy, paste, and run, along with raw mode, git integration, and cell type selection (code, text, etc.). The main content area displays the following Markdown code:

```
ordered lists:  
  
1. item 1  
1. item 2  
1. item 3  
  
1. item 1  
2. item 2  
3. item 3
```

The bottom status bar indicates "Mode: Command", "Ln 1, Col 38", and the file name "intro_to_jupyter.ipynb".

BASIC MARKDOWN

The screenshot shows a Jupyter Notebook interface with the title "intro_to_jupyter.ipynb". The notebook contains the following Markdown content:

```
unordered list:  
- item 1  
- item 2  
- item 3  
  
• item 1  
• item 2  
• item 3
```

The interface includes a toolbar with file operations like Save, New, Cut, Copy, Paste, Run Cell, Kernel, Settings, Help, and a git integration button. The bottom status bar shows "Mode: Command", "Ln 1, Col 38", and the file name "intro_to_jupyter.ipynb".

BASIC MARKDOWN

The screenshot shows a Jupyter Notebook interface with the title bar "intro_to_jupyter.ipynb" and the URL "localhost:8889/lab/tree/intro_to_jupyter.ipynb". The menu bar includes File, Edit, View, Run, Kernel, Tabs, Settings, and Help. The toolbar features icons for file operations like Open, Save, and New, along with Markdown, Cell, and Git buttons. The main content area displays the following Markdown code:

```
`code`  
displays code  
  
you can link to webpages like this:  
  
[link title](http://www.example.com/)  
link title
```

The bottom status bar shows "Simple" mode, 0 cells, R | Idle, Mode: Command, and the file path "intro_to_jupyter.ipynb".

BASIC MARKDOWN

The screenshot shows a Jupyter Notebook interface with the title bar "intro_to_jupyter.ipynb" and the URL "localhost:8889/lab/tree/intro_to_jupyter.ipynb". The menu bar includes File, Edit, View, Run, Kernel, Tabs, Settings, and Help. A toolbar below the menu has icons for file operations like save, new, cut, copy, paste, and run, followed by Raw, git, and other settings. On the left is a sidebar with icons for file, folder, and settings. The main content area contains the following text:

and embed images like this:

```
![alt](https://upload.wikimedia.org/wikipedia/commons/thumb/7/73/A_doing_aisatsu_kitten_%28Flickr%29.jpg/320px-A_doing_aisatsu_kitten_%28Flickr%29.jpg)
```

Below the text is an image of a white and orange kitten looking up.

At the bottom, the status bar shows "Simple" mode, cell count (0 \$ 2), kernel status (R | Idle), mode (Command), and file path (Ln 1, Col 1 intro_to_jupyter.ipynb).

BASIC MARKDOWN

The screenshot shows a Jupyter Notebook interface running in a browser window. The title bar indicates the file is titled "intro_to_jupyter.ipynb". The menu bar includes File, Edit, View, Run, Kernel, Tabs, Settings, and Help. The toolbar below the menu has icons for file operations like Open, Save, and New, along with a Raw button and a git icon. On the left, there's a sidebar with icons for file/folder, cell, list, and puzzle. The main content area displays the following Markdown code:

```
<div class="alert alert-block alert-info">this is an info box</div>
<div class="alert alert-block alert-warning">this is a warning box</div>
```

The rendered output consists of two boxes: a light blue box containing "this is an info box" and an orange box containing "this is a warning box".

At the bottom, the status bar shows "Simple" mode, 0 cells, R | Idle, Mode: Edit, and the file path "intro_to_jupyter.ipynb".

BASIC MARKDOWN

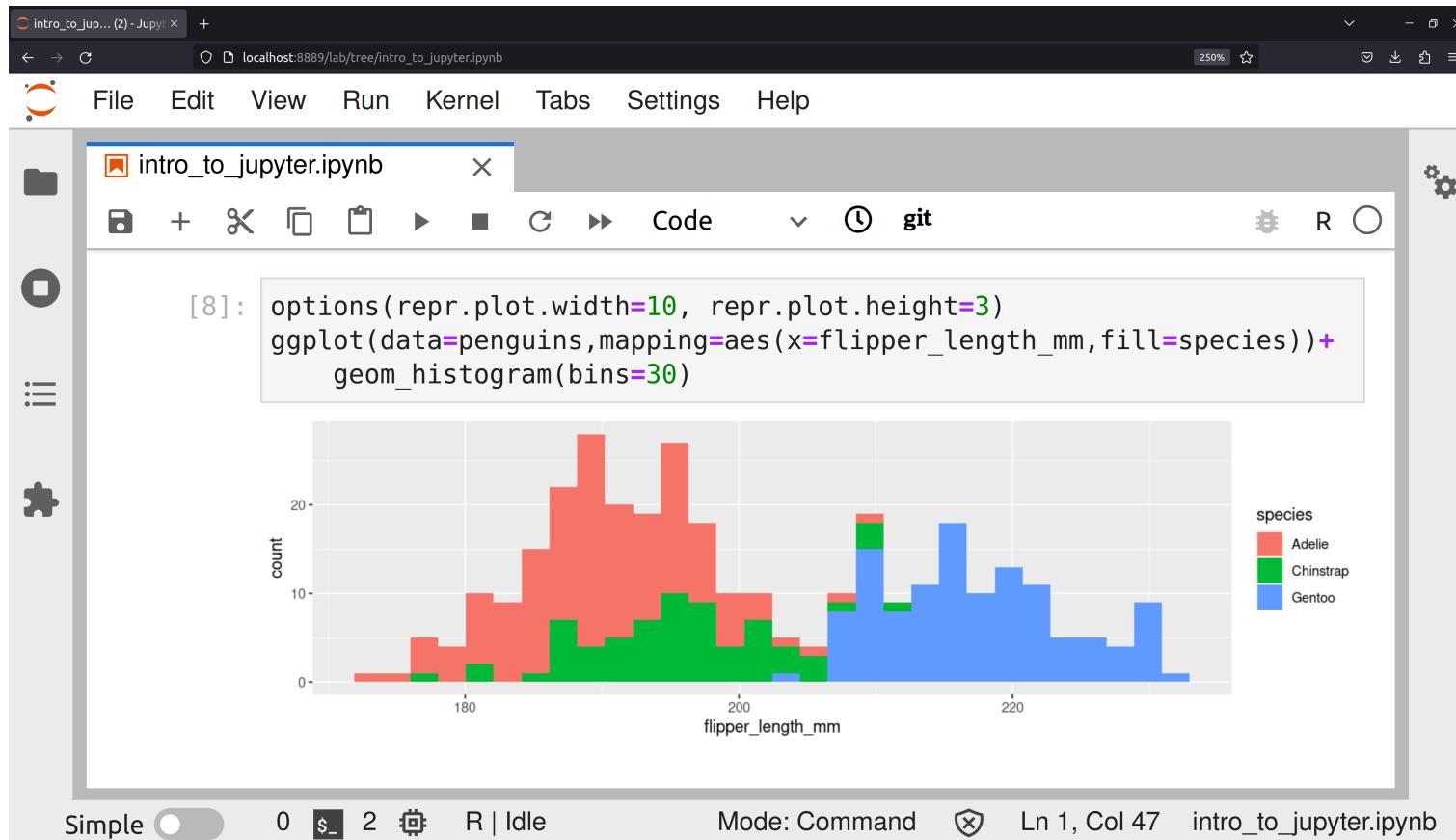
The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** The title bar displays "intro_to_jupyter.ipynb" and the URL "localhost:8889/lab/tree/intro_to_jupyter.ipynb".
- Toolbar:** The toolbar includes icons for File, Edit, View, Run, Kernel, Tabs, Settings, Help, and a Git icon.
- File List:** A sidebar on the left lists files, with "intro_to_jupyter.ipynb" currently selected.
- Text Cell:** A text cell contains the text "I can also embed *LATEX* mathematics type".
- Code Cell:** A code cell contains the following LaTeX code:

```
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$
```
- Output:** The output of the code cell is the mathematical formula
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$
- Bottom Bar:** The bottom bar shows "Simple" mode, 0 cells, R | Idle, Mode: Command, and the file path "intro_to_jupyter.ipynb".

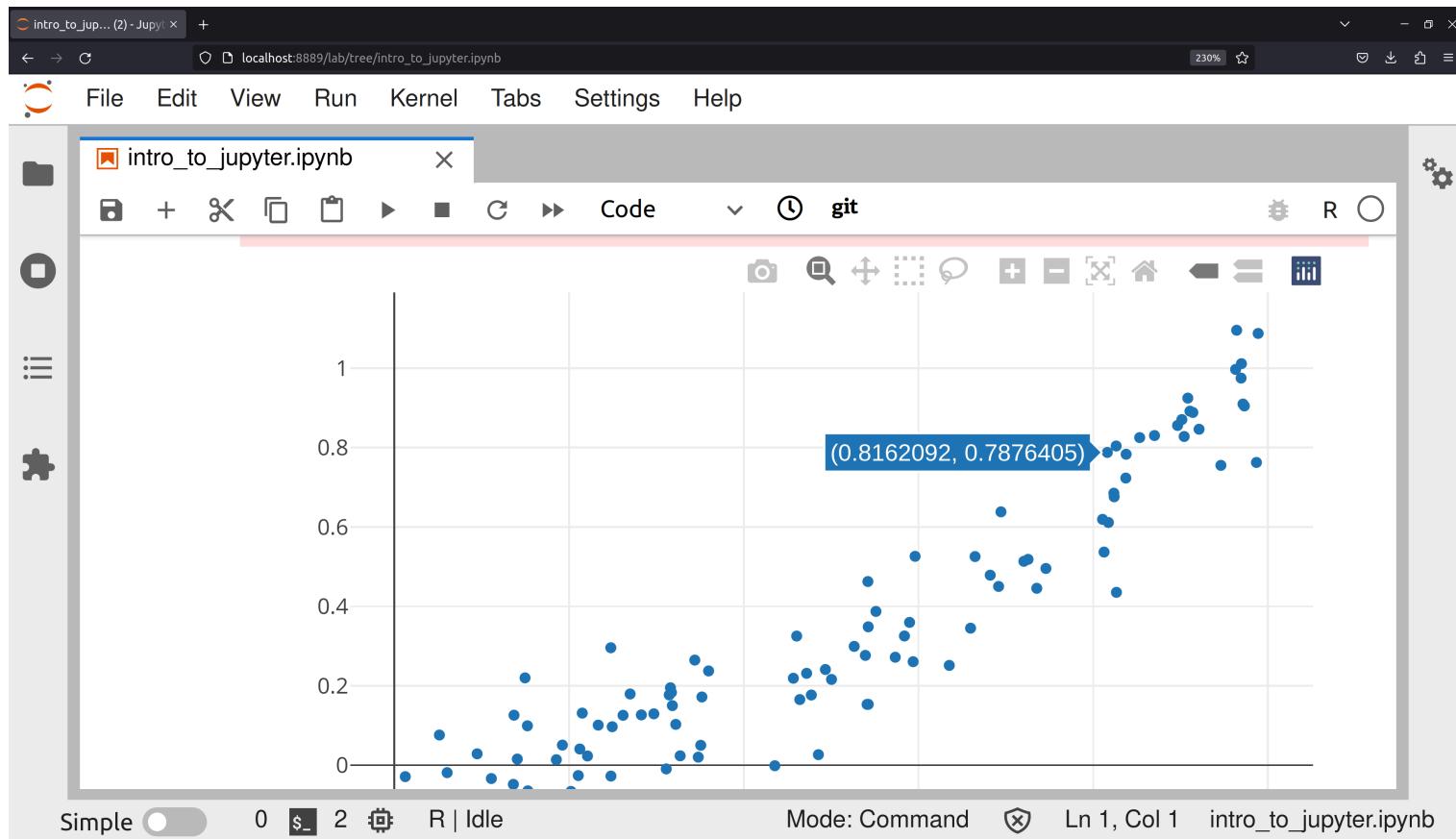
CODE: BASIC OUTPUT

Interweaved markdown commentary, intersperses **code** and **in-line output**, e.g.,



CODE: INTERACTIVE WIDGETS

Example: in R using plotly:



CODE: LANGUAGES

There are **many** language backends that jupyter can use. (These are called **kernels** in jupyter-speak). Jupyter lists well over 100 available kernels [here](#) including kernels for:

- python,
- r,
- julia,
- stata,
- octave,
- matlab,
- ...

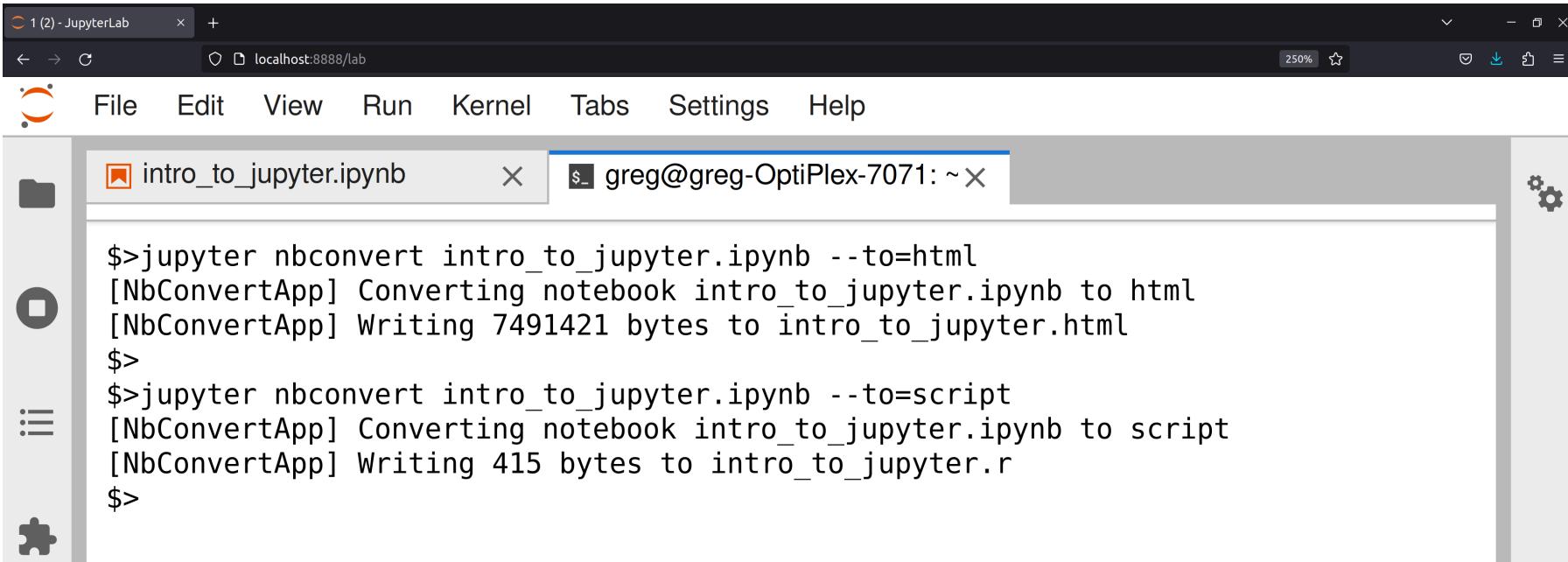
EXPORTING

One can **export** a **.ipynb** notebook using **jupyter** to many different formats like: `html`, `markdown` `pdf`, `reveal.js` `html slides`, ..., and even an executable script.

This can all be done using the command

```
jupyter nbconvert notebook.ipynb --to=format
```

EXPORTING

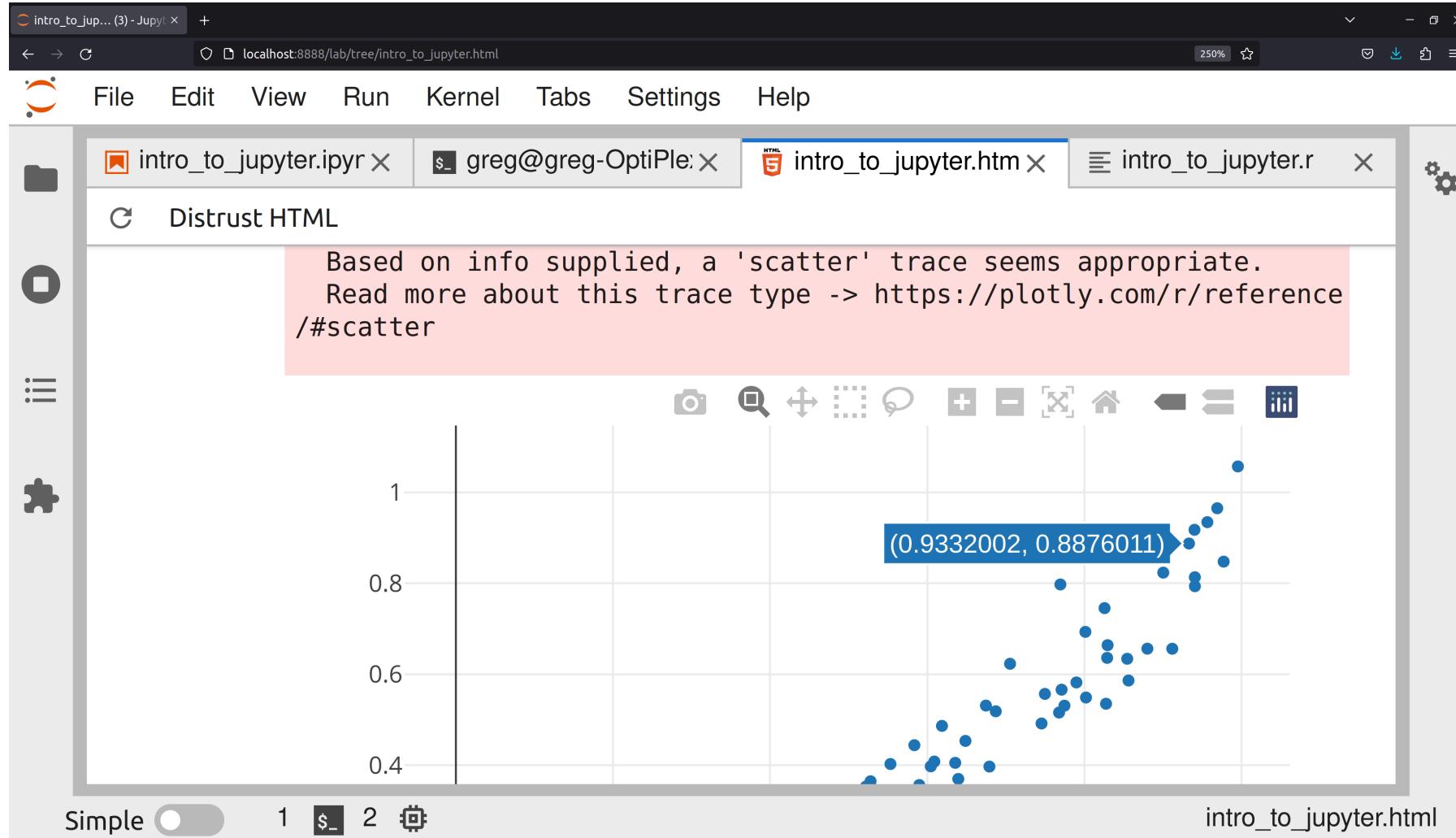


The screenshot shows the JupyterLab interface running in a web browser window. The title bar indicates "1 (2) - JupyterLab". The address bar shows "localhost:8888/lab". The menu bar includes File, Edit, View, Run, Kernel, Tabs, Settings, and Help. On the left is a sidebar with icons for file/folder, terminal, and settings. The main area has two tabs: "intro_to_jupyter.ipynb" and a terminal tab. The terminal tab displays the following command-line session:

```
$>jupyter nbconvert intro_to_jupyter.ipynb --to=html
[NbConvertApp] Converting notebook intro_to_jupyter.ipynb to html
[NbConvertApp] Writing 7491421 bytes to intro_to_jupyter.html
$>
$>jupyter nbconvert intro_to_jupyter.ipynb --to=script
[NbConvertApp] Converting notebook intro_to_jupyter.ipynb to script
[NbConvertApp] Writing 415 bytes to intro_to_jupyter.r
$>
```

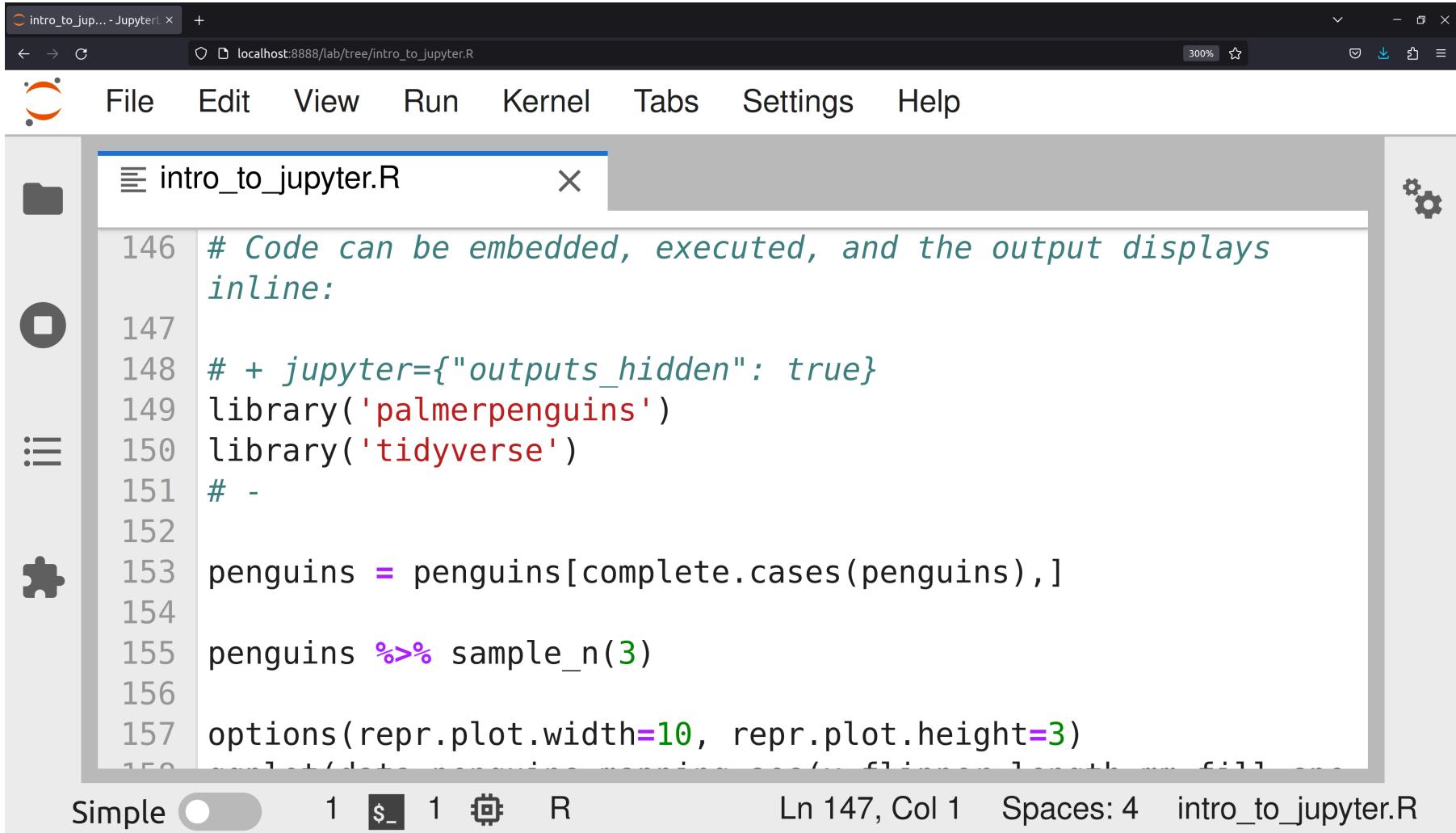
EXPORTING: HTML

Interactive widgets still work!



EXPORTING: SCRIPT

Exporting as a script gives us basic executable script



The screenshot shows a Jupyter Notebook interface with a single R script file open. The file is titled "intro_to_jupyter.R". The code in the file is as follows:

```
146 # Code can be embedded, executed, and the output displays
147 # inline:
148 # + jupyter={"outputs_hidden": true}
149 library('palmerpenguins')
150 library('tidyverse')
151 #
152
153 penguins = penguins[complete.cases(penguins),]
154
155 penguins %>% sample_n(3)
156
157 options(repr.plot.width=10, repr.plot.height=3)
158
#<code>#</code>
#<code>#</code>
```

The interface includes a toolbar at the top with File, Edit, View, Run, Kernel, Tabs, Settings, and Help. On the left, there's a sidebar with icons for file operations like New, Open, Save, and a gear icon for settings. The bottom of the screen shows the status bar with "Simple" mode selected, line 147, column 1, 4 spaces, and the file name "intro_to_jupyter.R".

NOTEBOOK INTEROPERABILITY

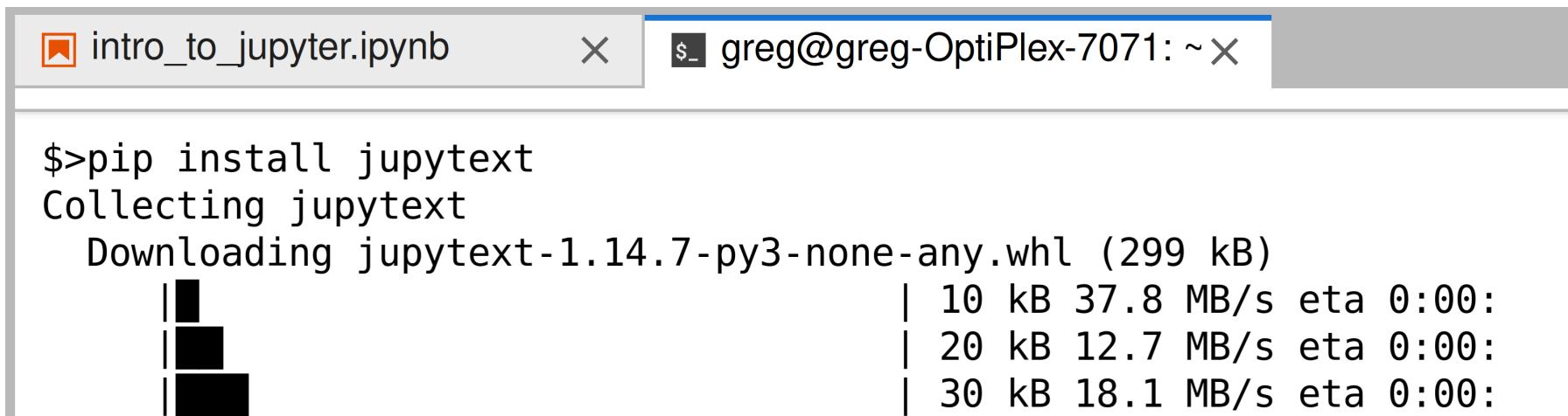
One can also use third-party tools to convert/maintain versions in **other notebook formats**.

For this, we find the tool `jupytext` to be invaluable.

We can install via

```
pip install jupytext
```

NOTEBOOK INTEROPERABILITY



A screenshot of a terminal window. The title bar shows two tabs: "intro_to_jupyter.ipynb" and "greg@greg-OptiPlex-7071: ~". The main pane displays the following command and its execution:

```
$>pip install jupytext
Collecting jupytext
  Downloading jupytext-1.14.7-py3-none-any.whl (299 kB)
    |██████████| 10 kB 37.8 MB/s eta 0:00:
    |██████████| 20 kB 12.7 MB/s eta 0:00:
    |██████████| 30 kB 18.1 MB/s eta 0:00:
```

The progress bar consists of three segments: a top segment with 10 kB, a middle segment with 20 kB, and a bottom segment with 30 kB. Each segment has a vertical bar on its left.

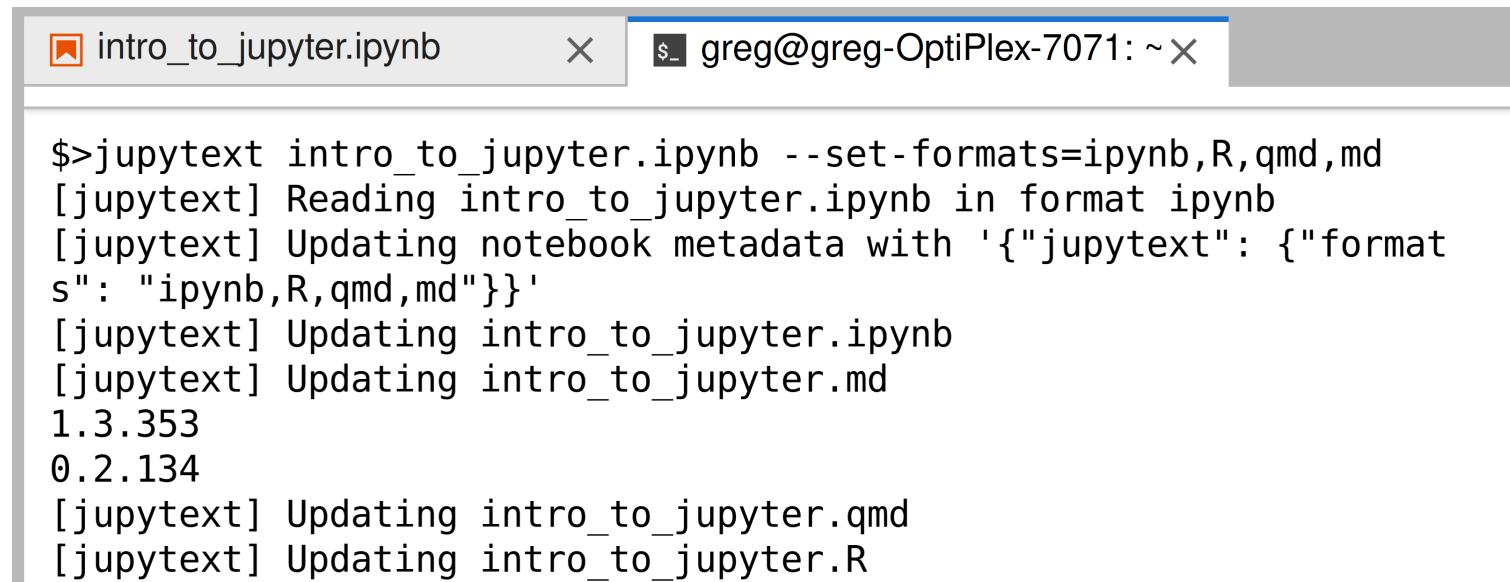
NOTEBOOK INTEROPERABILITY

jupytext **synchronously** propagates changes in the jupyter notebook
(when the .ipynb is saved) to other formats like markdown, annotated
scripts, rmarkdown, quarto, ...

via the command

```
jupytext notebook.ipynb --set-formats=format1,format2,...
```

NOTEBOOK INTEROPERABILITY



A screenshot of a terminal window with a light gray background. At the top, there are two tabs: one labeled "intro_to_jupyter.ipynb" with an orange icon, and another labeled "greg@greg-OptiPlex-7071: ~" with a blue icon. The main area of the terminal contains the following text:

```
$>jupytext intro_to_jupyter.ipynb --set-formats=ipynb,R,qmd,md
[jupytext] Reading intro_to_jupyter.ipynb in format ipynb
[jupytext] Updating notebook metadata with '{"jupytext": {"formats": "ipynb,R,qmd,md"}}'
[jupytext] Updating intro_to_jupyter.ipynb
[jupytext] Updating intro_to_jupyter.md
1.3.353
0.2.134
[jupytext] Updating intro_to_jupyter.qmd
[jupytext] Updating intro_to_jupyter.R
```

JUPYTEXT CHANGE

Changes are **synchronously** propagated to/from `.ipynb`. (Don't edit two files at once!)

The screenshot shows the JupyterLab interface with two tabs open:

- `intro_to_jupyter.ipynb`: An IPython notebook containing the following cell content:

```
ied, a 'scatter' trac
e seems appropriate.
Read more about thi
s trace type -> http
s://plotly.com/r/refer
ence/#scatter
```

This is a change
- `intro_to_jupyter.R`: An R script containing the following code:

```
166 aT =
167 data.frame(x=runif(100))
168 df$y =
169 df$x^2+rnorm(100,0,.1)
170 p <- plot_ly(x = df$x, y
171 = df$y, mode = "markers")
172 embed_notebook(p)
173 # This is a change
```

A pink callout box highlights the warning message in the notebook cell.

EXPORTING AND INTEROPERABILITY

- **Exporting** via `nbconvert` mostly immortalizes analysis for display e.g. as **HTML** or a **PDF**
- We can also export executable scripts:
 - **production**,
 - **non-interactive cluster**,
 - **version control**
- Exporting to display formats makes analysis more **sharable** and **reproducible**
 - same analysis, many formats

QUARTO: LATEST R STUDIO NOTEBOOK FORMAT

- `quarto` is the latest notebook format from Posit (RStudio).
- Largely, it supplants R notebooks and R markdown formats (and is back-compatable).

QUARTO: LATEST R STUDIO NOTEBOOK FORMAT

- Supports Python and Julia, with Jupyter back-end compatibility.
- Integrates well with RStudio, including a WYSIWYG editor.
- A solid alternative to Jupyter Lab/Notebooks.
- Quarto documents can be edited in Jupyter.

QUARTO AND JUPYTER

quarto and jupyter differences:

- quarto uses markdown-like `.qmd`, jupyter uses JSON `.ipynb`
- quarto geared towards publishing (e.g. figures, tables, references, ...),
use `myst` or `jupyter book` for jupyter
- `.qmd` doesn't store output, `.ipynb` does
- **A useful idiom:** do analysis in jupyter and mirror into `ipynb`, `md`, `R`, `qmd`, ...

NOTEBOOKS AND REPRODUCIBILITY

Why do notebooks help reproducibility:

- literate programming: interweaving code/commentary/output
 - allows rich commentary on code, output
 - develops a narrative that is easy to read
 - document diagnostic/exploratory/micro-decision analysis
- keeps commentary/output close to code
 - good tool for playing with code, immediately observing output

NOTEBOOKS AND REPRODUCIBILITY

Why do notebooks help reproducibility:

- good for showcasing results and interoperability
 - can be converted to many sharable formats (html, pdf, ...)
 - can convert **among** the notebook formats
- creates a reproducible record
 - code automatically generates results from data
 - this forces documentation on how the results were produced

NOTEBOOKS AND REPRODUCIBILITY

Why do notebooks help **reproducibility**:

- promotes good code organization via chunking
- software/formats not proprietary, easy to distribute

SOME POTENTIAL DOWNSIDES

While code notebooks can be great, there are some **potential issues**, including:

1. chunks can be run in non-sequential order, making them not reproducible (soln: re-run all analysis at the end sequentially)
2. saved format of notebook may make version control difficult (soln: jupytext)
3. not great for non-interactive environments (soln: jupytext)
4. conversion among various formats isn't 100% fool-proof

DISCUSSION

- Do you use notebooks regularly? Might you, now?
- Where do you find notebooks to be helpful?
- Where do you find notebooks **not** to be helpful?

EXERCISE

Create a notebook that does some simple EDA on some data. It should generate at least one plot or table.

- export the notebook to `.html`
- export the notebook to a script

Optional: Mirror this notebook into a script using `jupytext`.