Scientific Python QuickStart

Thomas J. Sargent and John Stachurski

Dec 11, 2023

CONTENTS

Aquí vamos a recopilar una serie de commando últiles para JupyterBooks.

Web de JupyterBooks

Web de MyST

Web de ExecutableBooks

Dec 11, 2023 | 20 words | 0 min read

CHAPTER

ONE

CODE BLOCKS Y ECUACIONES

Veamos como escrirbir bloques de código así como ecuaciones en JupyterBook usando MyST.

- *Code cells y code Blocks*
- *Ecuaciones*

Dec 11, 2023 | 266 words | 1 min read

1.1 Code cells y code Blocks

- *Code blocks and outputs*
- *glue para insertar variables en el texto*
- *Estadisticas de las ejecuciones*

En realidad no es necesario poner este indice local. Ya hay el indice de la izquierda y este te pone en azul los titulos de las secciones.

1.1.1 Code blocks and outputs

Los code block son un tipo de (directives)[./Roles_and_directives.md].

Jupyter Book will also embed your code blocks and output in your book. For example, here's some sample Matplotlib code:

```
from matplotlib import rcParams, cycler
import matplotlib.pyplot as plt
import numpy as np
plt.ion()
```

```
<contextlib.ExitStack at 0x7f58afab7820>
```

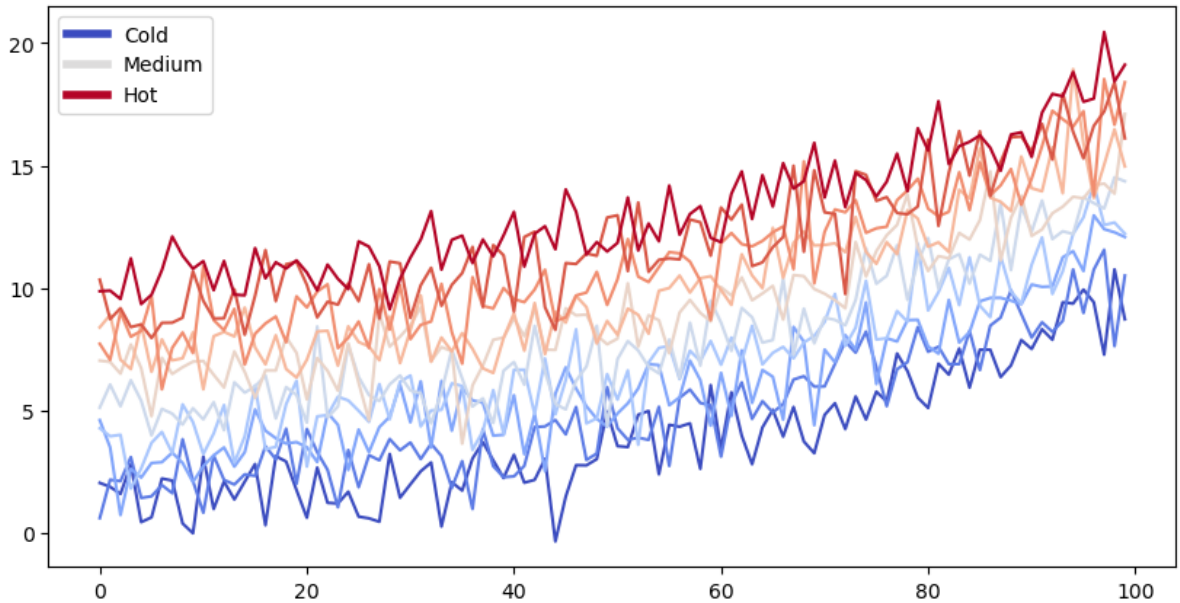
```
# Fixing random state for reproducibility
np.random.seed(19680801)

N = 10
data = [np.logspace(0, 1, 100) + np.random.randn(100) + ii for ii in range(N)]
data = np.array(data).T
```

(continues on next page)

(continued from previous page)

```
cmap = plt.cm.coolwarm
rcParams['axes.prop_cycle'] = cycler(color=cmap(np.linspace(0, 1, N)))


from matplotlib.lines import Line2D
custom_lines = [Line2D([0], [0], color=cmap(0.), lw=4),
                Line2D([0], [0], color=cmap(.5), lw=4),
                Line2D([0], [0], color=cmap(1.), lw=4)]

fig, ax = plt.subplots(figsize=(10, 5))
lines = ax.plot(data)
ax.legend(custom_lines, ['Cold', 'Medium', 'Hot'])
```

```
<matplotlib.legend.Legend at 0x7f57e4e9f940>
```



Celdas no ejecutables pero numeradas y con formato

Listing 1.1: This is my multi-line caption. It is *pretty nifty* ;-)

```
a = 2
print('my 1st line')
print(f'my {a}nd line')
```

Celdas con salida de error

Prueba de celda que da error

```
print(val_a)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[3], line 1
----> 1 print(val_a)

NameError: name 'val_a' is not defined
```

Celdas desplegables

Veamos una celda desplegable:

```
import numpy as np
import pandas as pd

np.random.seed(24)
df = pd.DataFrame({'A': np.linspace(1, 10, 10)})
df = pd.concat([df, pd.DataFrame(np.random.randn(10, 4), columns=list('BCDE'))],
            axis=1)
df.iloc[3, 3] = np.nan
df.iloc[0, 2] = np.nan

def color_negative_red(val):
    """
    Takes a scalar and returns a string with
    the css property `'color: red'` for negative
    strings, black otherwise.
    """
    color = 'red' if val < 0 else 'black'
    return 'color: %s' % color

def highlight_max(s):
    '''
    highlight the maximum in a Series yellow.
    '''
    is_max = s == s.max()
    return ['background-color: yellow' if v else '' for v in is_max]

df.style.\
    applymap(color_negative_red).\
    apply(highlight_max).\
    set_table_attributes('style="font-size: 10px"')
```

```
<pandas.io.formats.style.Styler at 0x7f57d0d84f70>
```

Celdas con scroll

Veamos una celca con scroll en la salida

```python
for ii in range(40):
    print(f"this is output line {ii}")
```

```
this is output line 0
this is output line 1
this is output line 2
this is output line 3
this is output line 4
this is output line 5
this is output line 6
this is output line 7
this is output line 8
this is output line 9
this is output line 10
this is output line 11
this is output line 12
this is output line 13
this is output line 14
this is output line 15
this is output line 16
this is output line 17
this is output line 18
this is output line 19
this is output line 20
this is output line 21
this is output line 22
this is output line 23
this is output line 24
this is output line 25
this is output line 26
this is output line 27
this is output line 28
this is output line 29
this is output line 30
this is output line 31
this is output line 32
this is output line 33
this is output line 34
this is output line 35
this is output line 36
this is output line 37
this is output line 38
this is output line 39
```

Colores en los print

Veamos ahora los colores que podemos poner en los print de python:

```
text = " XYZ "
formatstring = "\x1b[{}m" + text + "\x1b[m"

print(
    " " * 6
    + " " * len(text)
    + "".join("{:^{}}".format(bg, len(text)) for bg in range(40, 48))
)
for fg in range(30, 38):
    for bold in False, True:
        fg_code = ("1;" if bold else "") + str(fg)
        print(
            " {:>4} ".format(fg_code)
            + formatstring.format(fg_code)
            + "".join(
                formatstring.format(fg_code + ";" + str(bg)) for bg in range(40, 48)
            )
        )
```

```
            40   41   42   43   44   45   46   47
  30  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
1;30  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
  31  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
1;31  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
  32  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
1;32  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
  33  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
1;33  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
  34  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
1;34  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
  35  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
1;35  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
  36  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
1;36  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
  37  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
1;37  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ  XYZ
```

1.1.2 glue para insertar variables en el texto

"Gluing" variables en el notebook

Tenemos que importar la función `glue()` de la libreria `myst_nb`:

```
from myst_nb import glue
```

Veamos un ejemplo de como usarlo:

```
my_variable = "here is some text!"
glue("cool_text", my_variable,  display=False)
```

Para llamarla usamos `{glue:}`cool_text`` : 'here is some text!'

"Gluing" numeros, plots. math y tablas

```
# Simulate some data and bootstrap the mean of the data
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

n_points = 10000
n_boots = 1000
mean, sd = (3, .2)
data = sd*np.random.randn(n_points) + mean
bootstrap_indices = np.random.randint(0, n_points, n_points*n_boots).reshape((n_boots,
 ↪ n_points))

# Calculate the mean of a bunch of random samples
means = data[bootstrap_indices].mean(0)
# Calculate the 95% confidence interval for the mean
clo, chi = np.percentile(means, [2.5, 97.5])

# Visualize the histogram with the intervals
fig, ax = plt.subplots()
ax.hist(means)
for ln in [clo, chi]:
    ax.axvline(ln, ls='--', c='r')
ax.set_title("Bootstrap distribution and 95% CI")

# And a wider figure to show a timeseries
fig2, ax = plt.subplots(figsize=(6, 2))
ax.plot(np.sort(means), lw=3, c='r')
ax.set_axis_off()


# Store the values in our notebook

glue("boot_mean", means.mean(), display=False) # numero
glue("boot_clo", clo, display=False)           # numero
glue("boot_chi", chi, display=False)           # numero

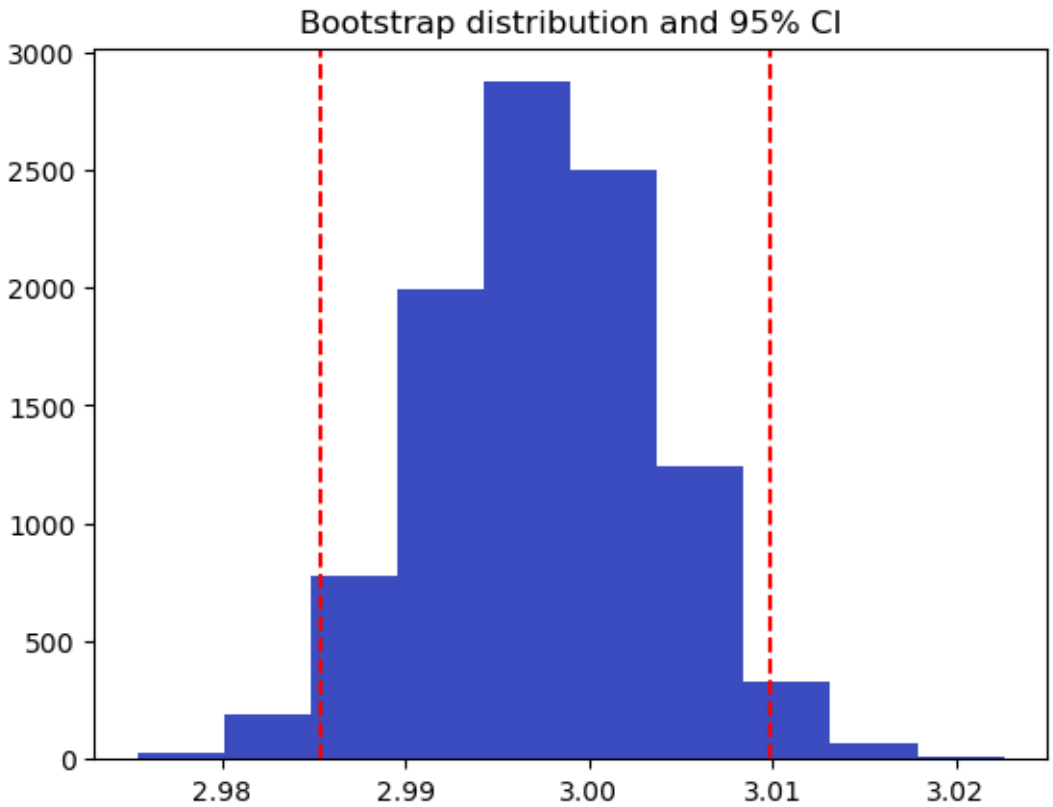
glue("boot_fig", fig, display=False)           # Plot
glue("sorted_means_fig", fig2, display=False)  # Plot

# Dataframes

bootstrap_subsets = data[bootstrap_indices][:3, :5].T
df = pd.DataFrame(bootstrap_subsets, columns=["first", "second", "third"])
display(df)

glue("df_tbl", df, display=False)
```

```
       first    second     third
0  2.864279  3.096170  3.040294
1  2.850520  3.582923  2.632284
2  3.182612  3.026727  3.013184
3  3.159771  2.793257  3.151598
4  3.087032  3.159038  3.132074
```



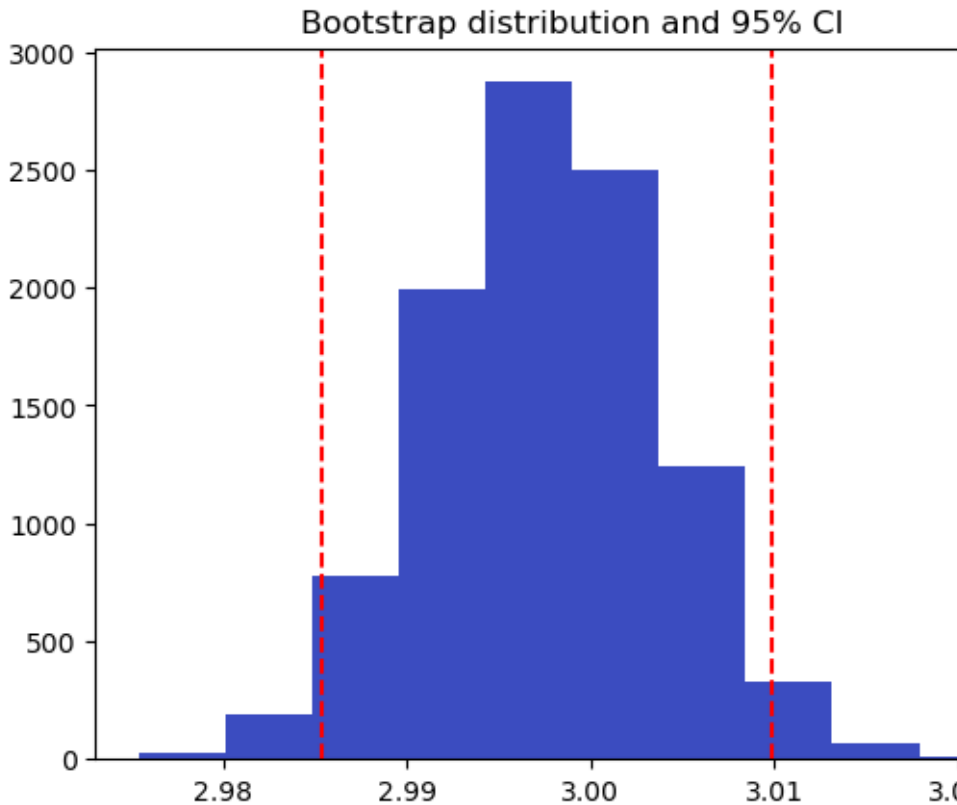
"Pasting" las variables

glue:any (sin formato)

Por defecto, al usar `{glue:}` estamos usando `{glue:any}`, que pega la salida "encolada" en línea o como bloque, respectivamente, sin formato adicional.

Veamos un ejemplo:

```
In-line text; {glue:}`boot_mean`, and a figure: {glue:}`boot_fig`.
```



In-line text; 2.99758724978736, and a figure:

glue:text

El `glue:text` es específico àra textos planos. Veamos un ejemplo:

```
The mean of the bootstrapped distribution was {glue:text}`boot_mean` (95%␣
↪confidence interval {glue:text}`boot_clo`/{glue:text}`boot_chi`).
```

The mean of the bootstrapped distribution was 2.99758724978736 (95% confidence interval 2.985312582852057/3.0098125309029817).

Podemos darle formato al output, como redondear números. La sintaxis es

- `{glue:text}`mykey:formatstring`

Por ejemplo:

```
Media: {glue:text}`boot_mean``

Media redondeada: {glue:text}`boot_mean:.2f`
```

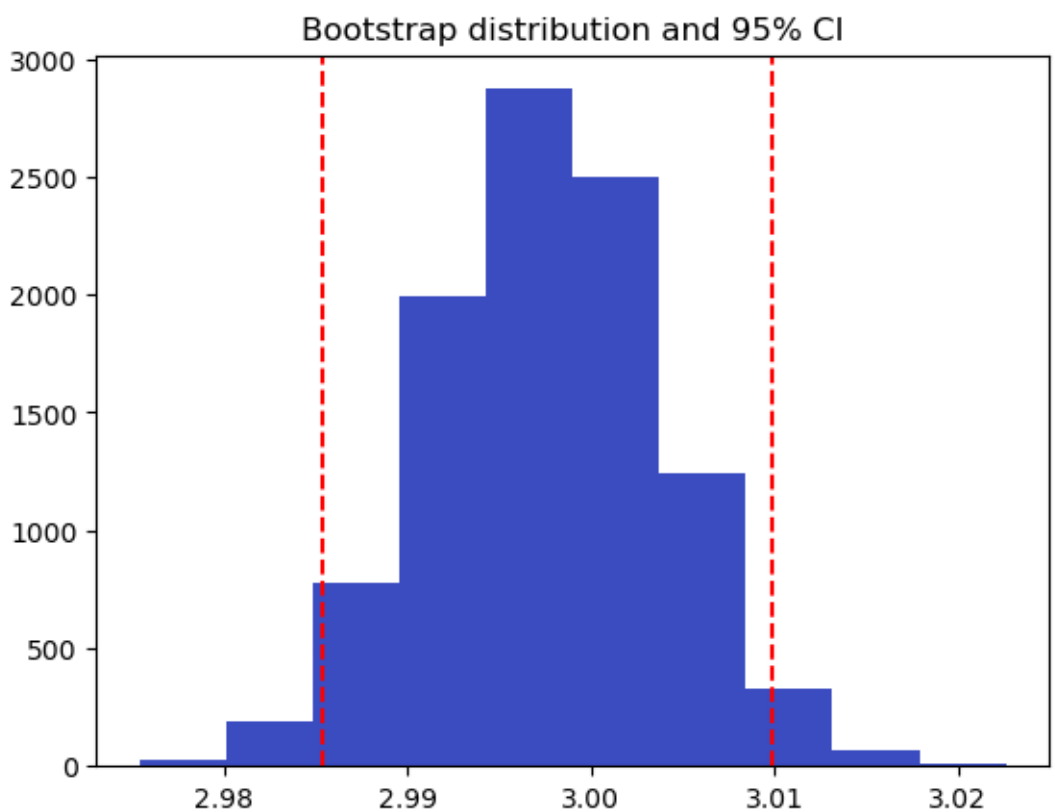
Media: 2.99758724978736

Media redondeada: 3.00

glue:figure

Sirve para figuras y tablas (dataframes).

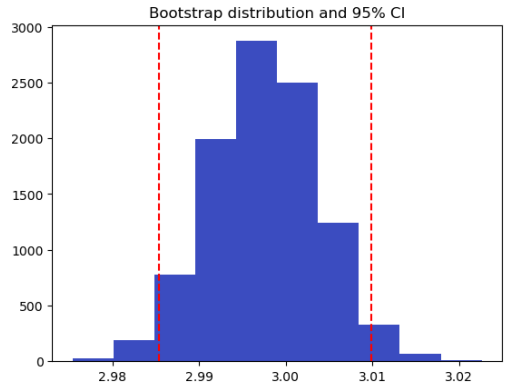
Figura:

```
```{glue:figure} boot_fig
:figwidth: 300px
:name: "fig-boot"

This is a **caption**, with an embedded `{glue:text}` element: {glue:text}
↪`boot_mean:.2f`!
```
```



Fig. 1.1: This is a **caption**, with an embedded `{glue:text}` element: 3.00!

```
Here is a {ref}`reference to the figure <fig-boot>`
```

Here is a *reference to the figure*

Dataframe:

```
```{glue:figure} df_tbl
:figwidth: 300px
:name: "tbl:df"
```
```

(continues on next page)

(continued from previous page)

```
A caption for a pandas table.
```

```
      first    second     third
0  2.864279  3.096170  3.040294
1  2.850520  3.582923  2.632284
2  3.182612  3.026727  3.013184
3  3.159771  2.793257  3.151598
4  3.087032  3.159038  3.132074
```

Fig. 1.2: A caption for a pandas table.

glue:math

```
import sympy as sym
f = sym.Function('f')
y = sym.Function('y')
n = sym.symbols(r'\alpha')
f = y(n)-2*y(n-1/sym.pi)-5*y(n-2)
glue("sym_eq", sym.rsolve(f,y(n),[1,4]) ,display=False)
```

```
```{glue:math} sym_eq
:label: eq-sym
``
```

$$\left(\sqrt{5}i\right)^{\alpha}\left(\frac{1}{2}-\frac{2\sqrt{5}i}{5}\right)+\left(-\sqrt{5}i\right)^{\alpha}\left(\frac{1}{2}+\frac{2\sqrt{5}i}{5}\right) \tag{1.1}$$

“Pasting” en tablas

```
| name                              |       plot                    | mean    ␣
↪                  | ci                                                |
|:---------------------------------:|:-----------------------------:|----------
↪-----------------|---------------------------------------------------|
| histogram and raw text            | {glue:}`boot_fig`             | {glue:}
↪`boot_mean`           | {glue:}`boot_clo`-{glue:}`boot_chi`               ␣
↪  |
| sorted means and formatted text | {glue:}`sorted_means_fig`     |
↪{glue:text}`boot_mean:.3f` | {glue:text}`boot_clo:.3f`-{glue:text}`boot_
↪chi:.3f` |
```
```

name	plot	mean	ci
histogram and raw text		2.9975872497873561	2.9853512582852057-3.0098125309029817
sorted means and formatted text		2.998	2.985-3.010

1.1.3 Estadisticas de las ejecuciones

Document	Modified	Method	Run Time (s)	Status
docs/01_00_Code_Blocks_y_ecuaciones	2023-11-30 10:00	cache	0.68	✅
docs/01_01_Code_Blocks	2023-12-11 09:16	cache	2.44	✅
docs/01_02_Ecuaciones	2023-11-30 10:00	cache	0.68	✅
docs/02_00_Roles_and_directives	2023-11-30 10:00	cache	0.68	✅
docs/02_01_Cuadros	2023-11-30 10:00	cache	0.68	✅
docs/02_02_Footnotes	2023-11-30 10:00	cache	0.68	✅
docs/02_03_Figuras	2023-11-30 10:00	cache	0.68	✅
docs/02_04_Mas_cosas	2023-11-30 10:00	cache	0.68	✅
docs/03_00_referenciar_cosas	2023-11-30 10:00	cache	0.68	✅
docs/04_00_Teoremas_pruebas_y_algoritmos	2023-11-30 10:00	cache	0.68	✅

Dec 11, 2023 | 25 words | 0 min read

1.2 Ecuaciones

$$f(x) = x^2 \tag{1.2}$$

$$w_{t+1} = (1 + r_{t+1})s(w_t) + y_{t+1} \tag{1.3}$$

A link to a dollar math block: (1.3)

$$e^{i\pi} + 1 = 0 \tag{1.4}$$

Refereciemos la ec. de euler (1.4).

$$y = ax^2 + bx + c \tag{1.5}$$
$$f(x) = x^2 + 2xy + y^2$$

Refereciemos la ec. (1.5).

$$y = ax^2 + bx + c \tag{1.5}$$
$$f(x) = x^2 + 2xy + y^2$$

Refereciemos la ec. (1.5).

$$\begin{aligned}(a+b)^2 &= (a+b)(a+b) \\ &= a^2 + 2ab + b^2\end{aligned} \tag{1.6}$$

Refereciemos la ec. (1.6).

Dec 11, 2023 | 22 words | 0 min read

CHAPTER

TWO

ROLES AND DIRECTIVES

Basicamente son funciones. https://myst-parser.readthedocs.io/en/latest/syntax/roles-and-directives.html#syntax-directives

Las **directives** son funciones de varias lineas. Los **roles** son de una línea.

Dec 11, 2023 | 284 words | 1 min read

2.1 Cuadros (admonitions)

2.1.1 Cuadros donde se puede cambiar el título

This is an admonition

This is an admonition

This is an admonition class note

This is an admonition class note

This is an admonition class important

This is an admonition class important

This is an admonition class warning

This is an admonition class warning

This is an admonition class tip

This is an admonition class tip

This is an admonition class tip

This is an admonition class tip

```
Esto es para escribir texto plano. Es decir, no se renderizan
los comandos estilo {numref}`Figure %s <fig-target_3>`
```

> Quote block
>
> Y vemos que puede ser de varias líneas

2.1.2 Cuadros rápidos (no se puede cambiar el título)

Note: This a note (no se puede cambiar el titulo)

Warning: This is a warning (no se puede cambiar el título)

Tip: This is a tip (no se puede cambiar el título)

See also:

This is a seealso (no se puede cambiar el título)

Note: The next info should be nested

Warning: Here's my **warning**

2.1.3 Cuadros (usando ::: :::)

Para entornos donde no se reconoce la sintaxis ``` puede usarse la sintaxis :::

Important:

Note: Esto es una nota

Cuadro de Warning

This is a **warning**

2.1.4 Cuadros con html

A drawback of admonition syntax is that it will not render in interfaces that do not support this syntax (e.g., GitHub). If you'd like to use admonitions that are defined purely with HTML, MyST can parse them via the html_admonitions extension.

This is the title

This is the *content*

During the Sphinx render, both the class and name attributes will be used by Sphinx, but any other attributes like style will be discarded.

There can be no empty lines in the block, otherwise they will be read as two separate blocks. If you want to use multiple paragraphs then they can be enclosed in `<p>`:

Note

Paragraph 1

Paragraph 2

Note

Some **content**

A *title*

Paragraph 1

Paragraph 2

Dec 11, 2023 | 66 words | 0 min read

2.2 Footnotes

This is a manually-numbered footnote reference.[3]

This is an auto-numbered footnote reference.[1]

A longer footnote definition.[2]

Como podemos ver en la nota a pie de página[?]

Dec 11, 2023 | 31 words | 0 min read

[3] This is a manually-numbered footnote definition.

[1] This is an auto-numbered footnote definition.

[2] This is the ***footnote definition***. That continues for all indented lines

- even other block elements

Plus any preceding unindented lines, that are not separated by a blank line

2.3 Figuras

Align options: “top”, “middle”, “bottom”, “left”, “center”, or “right”

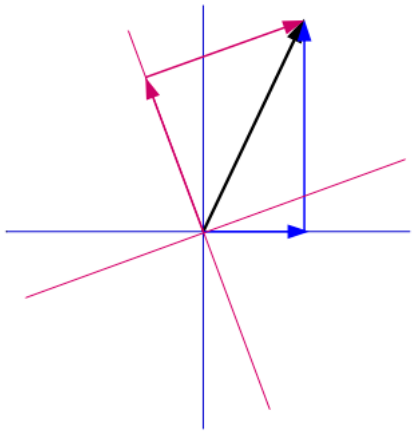


Fig. 2.1: This is the caption of the figure (a simple paragraph).

Referencia a la figura: Fig. 2.1

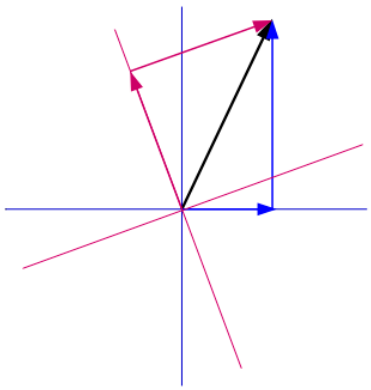


Fig. 2.2: This is a caption in **Markdown**

Referencia a la figura: Fig. 2.2

Dec 11, 2023 | 334 words | 2 min read

2.4 Más cosas

Presentamos primero la linea horizontal, pues la usaremos (pueden ser tres − o más):

> this is a quote. this is a quote. this is a quote. this is a quote. this is a quote. this is a quote
>
> this is a quote
>
> this is a quote

Veamos otra forma de hacer cuotas:

> Here is a cool quotation.
>
> ---Jo the Jovyan

2.4.1 Margin and sidebar

My sidebar title

My sidebar content

If you use a sidebar within your content, the sidebar will stay in-line with your page's content. However, it will be placed to the right, allowing your content to wrap around it. This prevents the sidebar from breaking up the flow of your content. This is particularly useful if you've got tall-and-long blocks of content or images that you would like to provide context to throughout your content.

Escribimos en el margen !!!

Note: Incluso escribimos notas!!!

2.4.2 hlist

Veamos una hlist:

- Elemento 1
- Elemento 2
- Elemento 3
- Elemento 4
- Elemento 5
- Elemento 6
- Elemento 7

2.4.3 rubric

Veamos un rubric:

This is a rubric (título chulo básicamente)

2.4.4 Centered

Veamos un centered:

Esto es un centered (negrita y centrado)

2.4.5 Sectionauthor

Autor (no se como funciona):

2.4.6 Glossary

Veamos glosarios:

Term one
An indented explanation of term 1

A second term
An indented explanation of term2

2.4.7 Sustitutions

Veamos las sustitution (hay que añadirlas al encabezado de la página)

Sustitution 1: I'm a **substitution**

Sustitution 2:

Note: I'm a **substitution**

You can also define book-level substitution variables with the following configuration:

```
parse:
  myst_substitutions:
    key: value
```

Sustituciones con formato: MyST substitutions use Jinja templates in order to substitute in key / values. This means that you can apply any standard Jinja formatting to your substitutions. For example, you can replace text in your substitutions like so:

The original key1: I'm a **substitution**

I'm a **substitution**

2.4.8 Grids

A

B

C

D

A

A2

B

B2

C

C2

D

D2

One! Here's the first card.

Two! Here's the second card.

Three! Here's the third card.

2.4.9 Dropdowns

Here's my dropdown

And here's my dropdown content

Click here!

Here's what's inside!

Note: The note body will be hidden!

2.4.10 Tab content

Tab 1 title

My first tab

Tab 2 title

My second tab with `some code`!

C++

```
int main(const int argc, const char **argv) {
  return 0;
}
```

Python

```
def main():
    return
```

Java

```
class Main {
    public static void main(String[] args) {
    }
}
```

2.4.11 Table

Table 2.1: My table title

header 1	header 2
3	4

Here is Table 2.1

Dec 11, 2023 | 73 words | 0 min read

CHAPTER

THREE

REFERENCIAR COSAS

Referencia a una sección usando el título: *Code Blocks y ecuaciones*. Se usa el **nombre del fichero**

```
{doc}`./01_00_sec_Code_Blocks_y_Ecuaciones`
```

Referenciamos una sección Section 1. Se usa una **label**

```
`(sec_Code_Blocks_y_Ecuaciones)=`
# Code Blocks y Ecuaciones

{numref}`sec_Code_Blocks_y_Ecuaciones`.
```

Referencias a ecuaciones: (1.3)

```
{eq}`my_other_label`
```

Referencias a figuras: Fig. 2.1

```
{numref}`fig-target`
```

Referencias a bloques de código Listing 1.1 (deben de tener tanto :caption: como :name:)

```
{numref}`label_codeblock`
```

3.1 Editar nombre en numref:

Podemos editar el nombre que apare en las numref antes del número. Por ejemplo, podemos pasar de Fig. 2.1 a Figura 2.1.

```
{numref}`Fig. %s <fig-target>`
```

Otro ejemplo: sec. 1 en vez de Section 1.

```
{numref}`sec. %s <sec_Code_Blocks_y_Ecuaciones>`
```

3.2 Referencias bibliográficas

Cita: `{cite}`guttag2016introduction`` : [1]

Dec 11, 2023 | 167 words | 1 min read

CHAPTER

FOUR

TEOREMAS, PRUEBAS, ALGORITMOS ...

Infrastructure to support items such as proof and algorithm style formatting is provided by the sphinx-proof extension.

Para ver todas las directivas: https://sphinx-proof.readthedocs.io/en/latest/syntax.html#collection-of-directives

4.1 Theorems

Theorem 4.1 (Titulo del teorema (opcional))

Esto sería un teorema

Referenciamos: *Theorem 4.1*

4.2 Lemmas

Lemma 4.1 (Titulo del lemma (opcional))

Esto sería un lemma

Referenciamos: *Lemma 4.1*

4.3 Corollaries

Corollary 4.1 (Titulo del corollary (opcional))

Esto sería un corollary

Referenciamos: *Corollary 4.1*

4.4 Proofs

Proof. Esto sería un proof. No se puede referenciar

4.5 Definitions

Definition 4.1 (Titulo del definition (opcional))

Esto sería un definition

Referenciamos: *Definition 4.1*

4.6 Examples

Example 4.1 (Titulo del example (opcional))

Esto sería un example

Referenciamos: *Example 4.1*

4.7 Axioms

Axiom 4.1 (Titulo del axiom (opcional))

Esto sería un axiom

Referenciamos: *Axiom 4.1*

4.8 Algoritms

Algorithm 4.1 (Titulo del algoritm (opcional))

Esto sería un algorithm

Referenciamos: *Algorithm 4.1*

4.9 Conjectures

Conjecture 4.1 (Titulo del conjetures (opcional))

Esto sería un conjetures

Referenciamos: *Conjecture 4.1*

4.10 Criteria

Criterion 4.1 (Titulo del criteria (opcional))

Esto sería un criteria

Referenciamos: *Criterion 4.1*

4.11 Observations

Observation 4.1 (Titulo del observation (opcional))

Esto sería un observation

Referenciamos: *Observation 4.1*

4.12 Properties

Property 4.1 (Titulo del property (opcional))

Esto sería un property

Referenciamos: *Property 4.1*

4.13 Propositions

Proposition 4.1 (Titulo del proposition (opcional))

Esto sería un proposition

Referenciamos: *Proposition 4.1*

4.14 Remarks

Remark 4.1 (Titulo del remark (opcional))

Esto sería un remarks

Referenciamos: *Remark 4.1*

BIBLIOGRAPHY

[1] John Guttag. *Introduction to computation and programming using Python: With application to understanding data*. MIT Press, 2016.

PROOF INDEX