

Hola, Quarto

Pedro L. Luque

1. Uso de Quarto-Typst: Python y R

1.1. Markdown

Markdown es un formato de texto fácil de leer y escribir:

- Es *plain text* por lo que funciona bien con control de versiones
- Puede ser **rendered** en HTML, PDF y más
- Aprende más en: <https://quarto.org/docs/authoring/>

1.2. Celdas de código (chunks de código): Python y R

Desde Positron o VSCode, puede ejecutar celdas de código en Python y R, pulsando el botón “Run Cell” que está por encima de cada celda de código. No es necesario ejecutar todo el documento, solo la celda que le interesa. Además, ejecutará las celdas en el orden que desee, no es necesario seguir el orden del documento (habrá que tener en cuenta las dependencias entre celdas). Positron y VSCode mostrarán la salida de cada celda en la “Consola”, para que pueda ver los resultados inmediatamente.

Aquí hay una celda de código en Python:

```
import os

cpu_count = os.cpu_count()
print(f"Núcleos (cores) de CPU disponibles: {cpu_count}")
```

Núcleos (cores) de CPU disponibles: 12

Aquí hay una celda de código en R:

```
nuevafuncion <- function(x) {
  y = x^2
  y = y + 1
  y = y / 2
  y = log(y)
  return(y)
}
(p1 = nuevafuncion(mtcars$mpg))
```

```
[1] 5.398163 5.398163 5.562296 5.435816 5.166755 5.101724 4.632250 5.697697
[9] 5.562296 5.219382 5.068401 4.905127 5.011601 4.753763 3.999668 3.999668
[17] 4.687165 6.264122 6.136820 6.354553 5.445120 4.792687 4.753763 4.488018
```

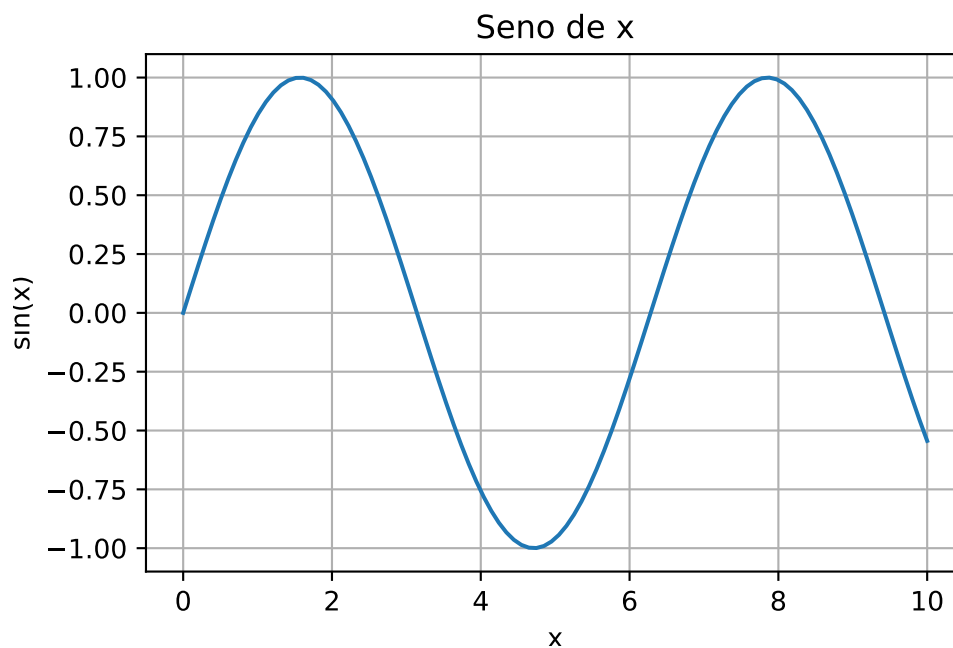
```
[25] 5.219382 5.921967 5.824524 6.136820 4.830870 5.270663 4.727388 5.435816
```

```
(p2 = nuevafuncion(mtcars$hp))
```

```
[1] 8.707896 8.707896 8.372167 8.707896 9.636457 8.614864 10.309386  
[8] 7.561382 8.414717 8.931288 8.931288 9.692797 9.692797 9.692797  
[15] 9.952897 10.048151 10.183030 7.686392 7.209710 7.655864 8.456381  
[22] 9.328168 9.328168 10.309386 9.636457 7.686392 8.328693 8.761707  
[29] 10.458765 9.636457 10.935123 8.689633
```

1.2.1. Ejemplos de códigos Python

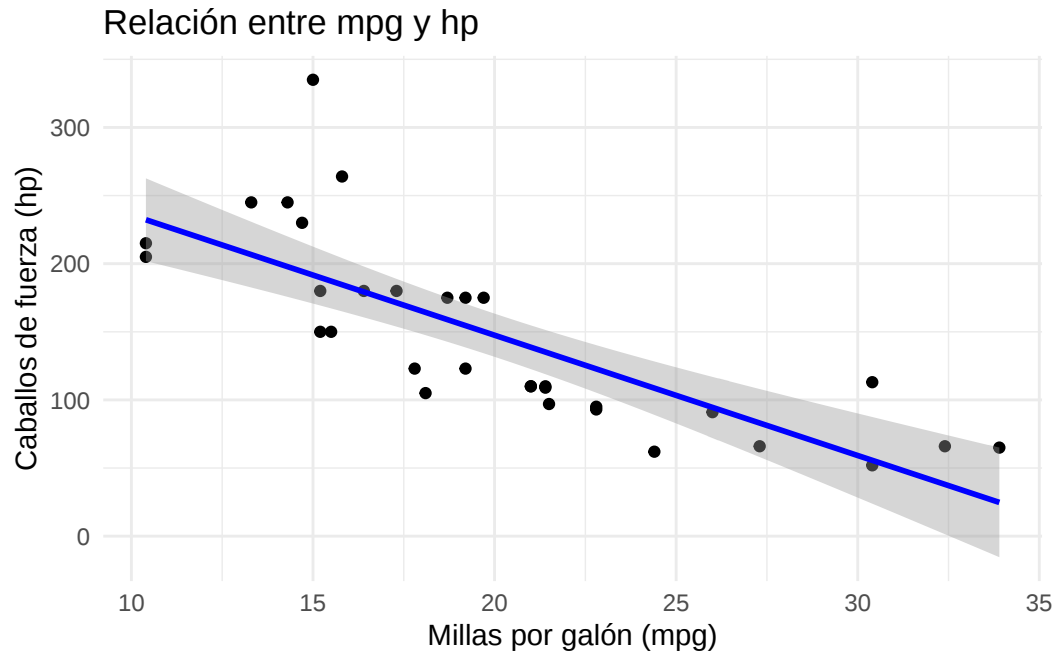
```
import numpy as np  
import matplotlib.pyplot as plt  
x = np.linspace(0, 10, 100)  
y = np.sin(x)  
plt.plot(x, y)  
plt.title("Seno de x")  
plt.xlabel("x")  
plt.ylabel("sin(x)")  
plt.grid()  
plt.show()
```



1.2.2. Ejemplos de códigos R

```
library(ggplot2)  
data(mtcars)  
ggplot(mtcars, aes(x=mpg, y=hp)) +  
  geom_point() +
```

```
geom_smooth(method="lm", col="blue") +
labs(title="Relación entre mpg y hp",
      x="Millas por galón (mpg)",
      y="Caballos de fuerza (hp)") +
theme_minimal()
```



1.3. Ecuaciones matemáticas con LaTeX

Uso de LaTeX para escribir ecuaciones, encerrando la expresión LaTeX entre dos dobles signos del dólar, \$\$, para ecuaciones centradas en una línea propia:

$$\chi' = \sum_{i=1}^n k_i s_i^2 = \int_0^{\infty} \hat{K}(u) dF(u)$$

```
print("Hola mundo")
```

```
print("Hola mundo")
```

```
print("Hola mundo")
```