



Go, así como lo viste en el curso tiene características muy particulares que lo hace encajar muy bien en el mundo del Data Science. Especialmente en estas tres características:

- Manejar tareas pesadas con múltiples hilos de ejecución a través de la concurrencia es muy sencillo.
- Es compilado, lo que lo hace mucho más eficiente.
- Optimiza muy bien los recursos del hardware.

Por tal motivo, esta lectura está enfocada en mostrar el estado del arte actual de Go en Data Science.

Jupyter

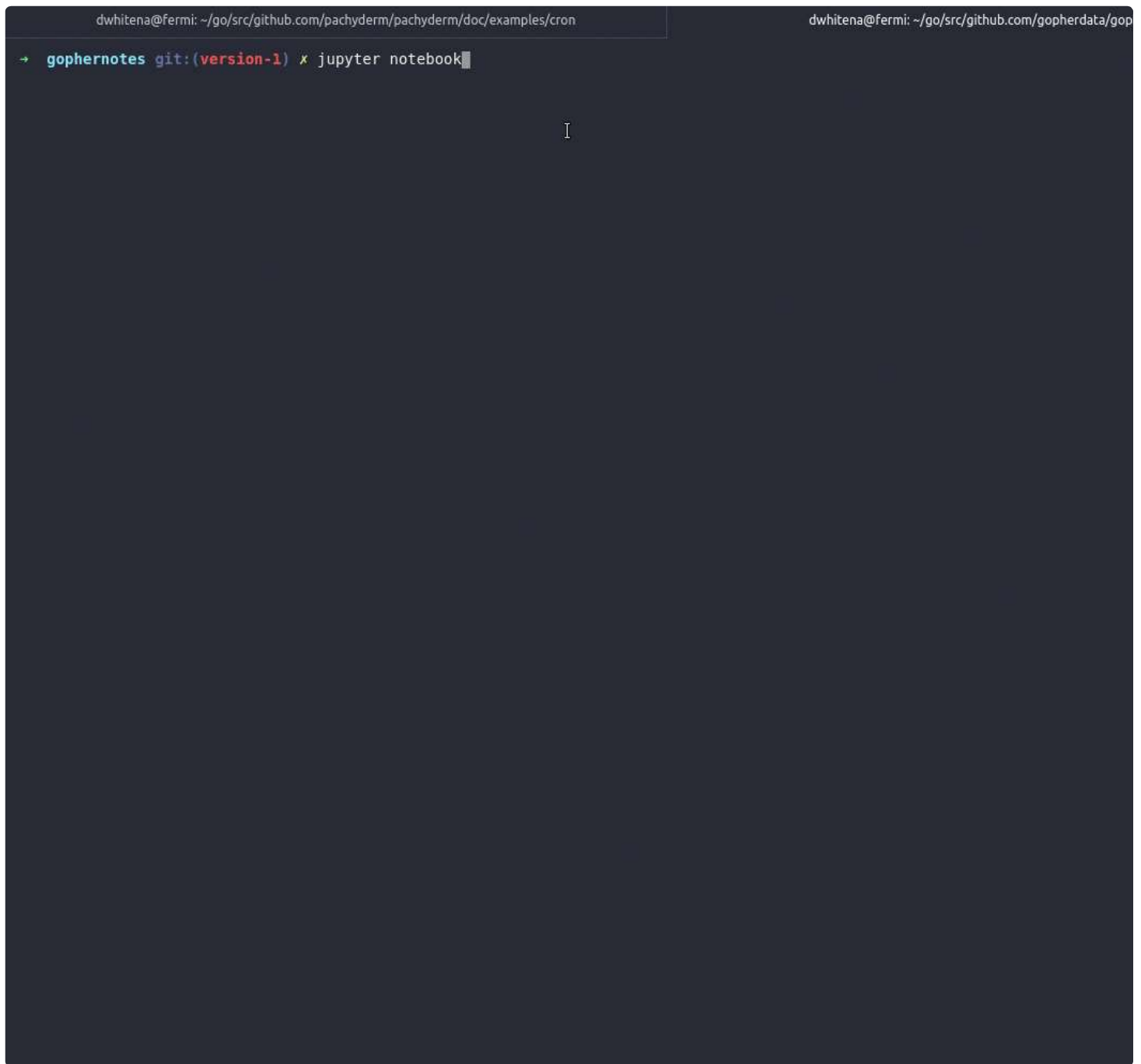


Jupyter es una de las principales herramientas que utilizamos los Data Scientists en el día a día ya que nos permite ejecutar código de manera fácil e iterativa pudiendo reciclar variables en cualquier momento.

Al momento de instalarlo vía pip (manejador de paquetes para Python) viene instalado con el kernel de Python listo para ejecutar código Python.

A pesar, que Go es un lenguaje compilado, la comunidad ha creado un intérprete de Go llamado **gomacro** para ejecutar código sin compilar. Partiendo de ello, crearon también un

kernel de Go para usarlo en Jupyter Notebook llamado [gophernotes](#)



The screenshot shows a terminal window with two tabs. The left tab is titled 'dwhitena@fermi: ~/go/src/github.com/pachyderm/pachyderm/doc/examples/cron' and the right tab is titled 'dwhitena@fermi: ~/go/src/github.com/gopherdata/gop'. The terminal content shows a prompt followed by the command 'gophernotes git:(version-1) x jupyter notebook' and a cursor. The background is dark blue.

De los métodos de instalación, mi sugerencia es que utilices la vía de Docker.

Manejo de DataFrames

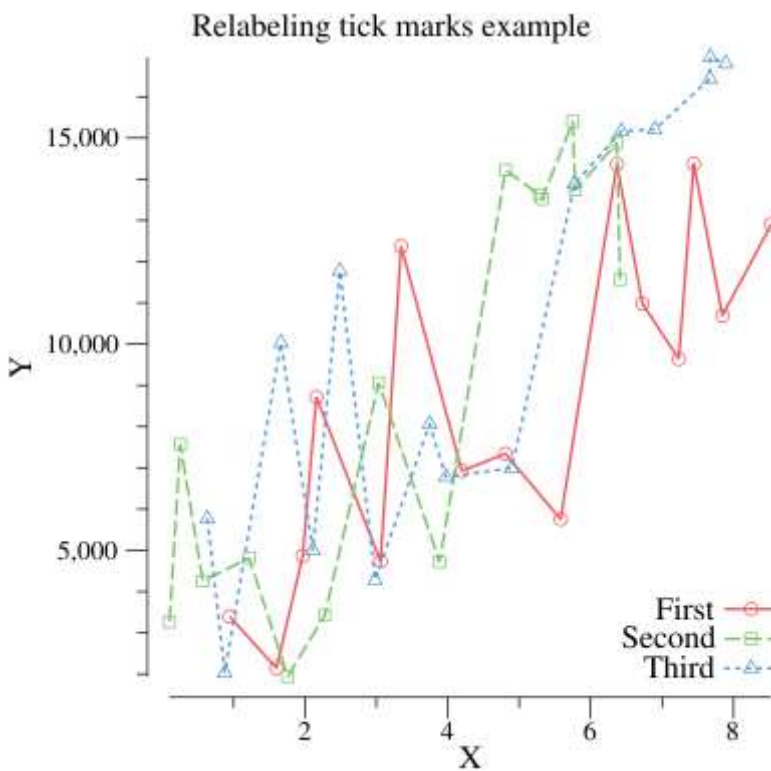
En este punto los más populares son: [qframe](#), [gota](#) y [dataframe-go](#).

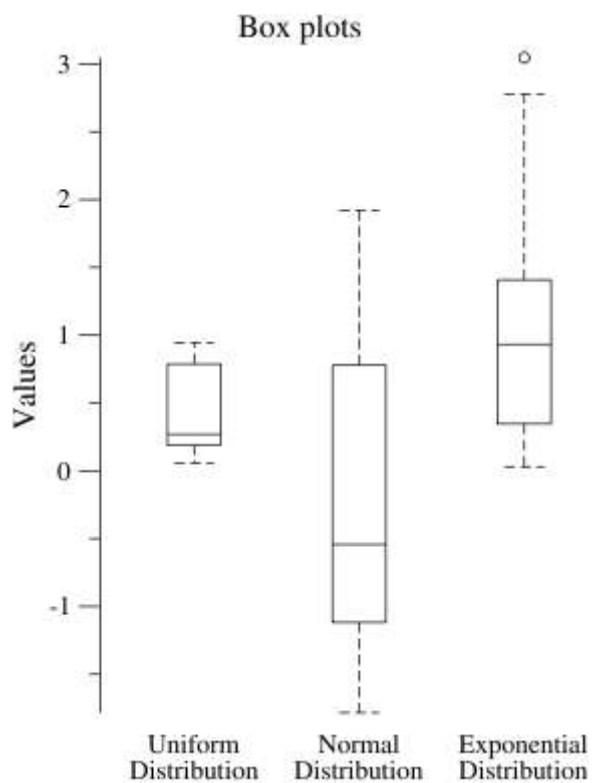
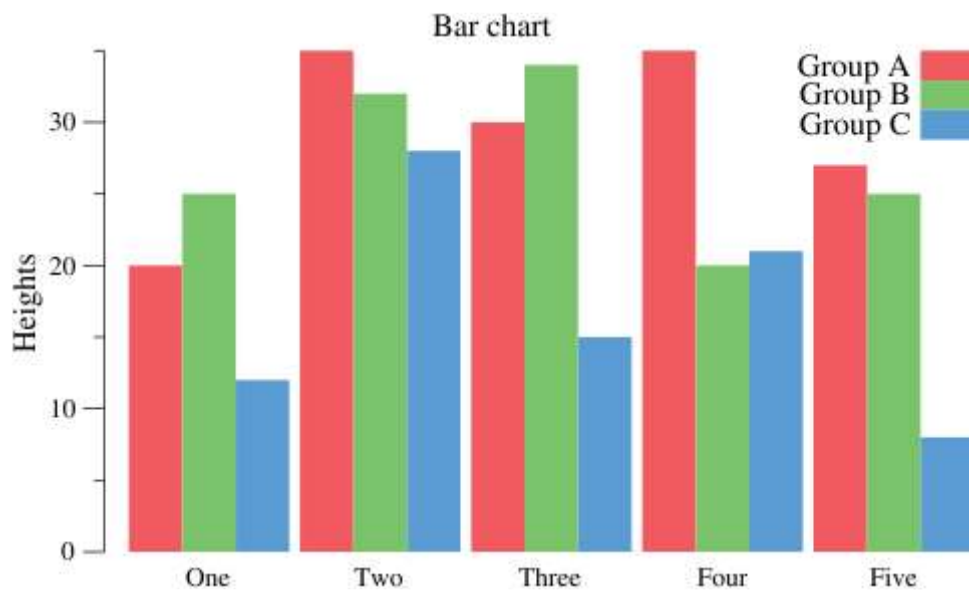
Hasta la fecha, ninguno está en su versión estable, pero todos están haciendo un gran trabajo porque como lo notaste en el curso, en Go las variables vacías no son nulas sino que tienen un valor por defecto. Y esto en el mundo de los datos es todo un reto ya que tener datos nulos es el pan de cada día.

Visualizaciones

En este apartado destacan dos:

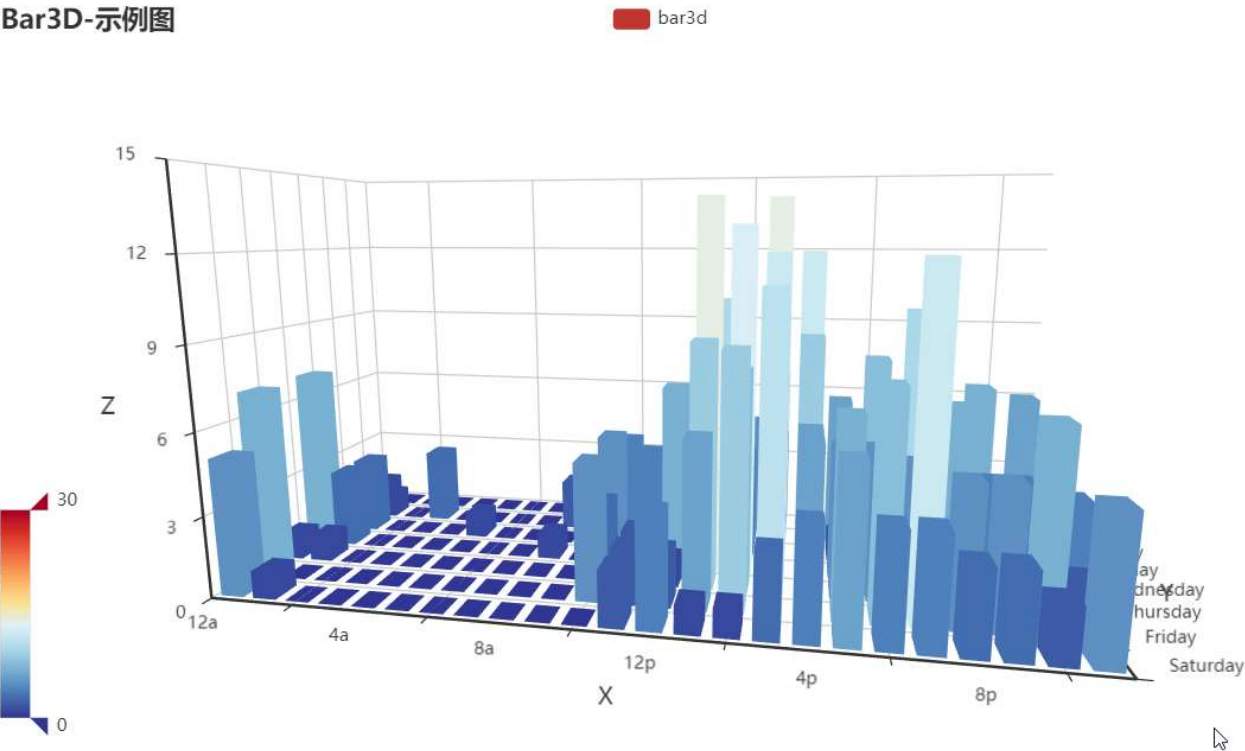
- [gonum/plot](#): Gonum no solo es el [Numpy](#) en Go sino que además tiene su propio código de visualización. En este caso gonum/plot te permite hacer visualizaciones estáticas. Te comparto algunos ejemplos:



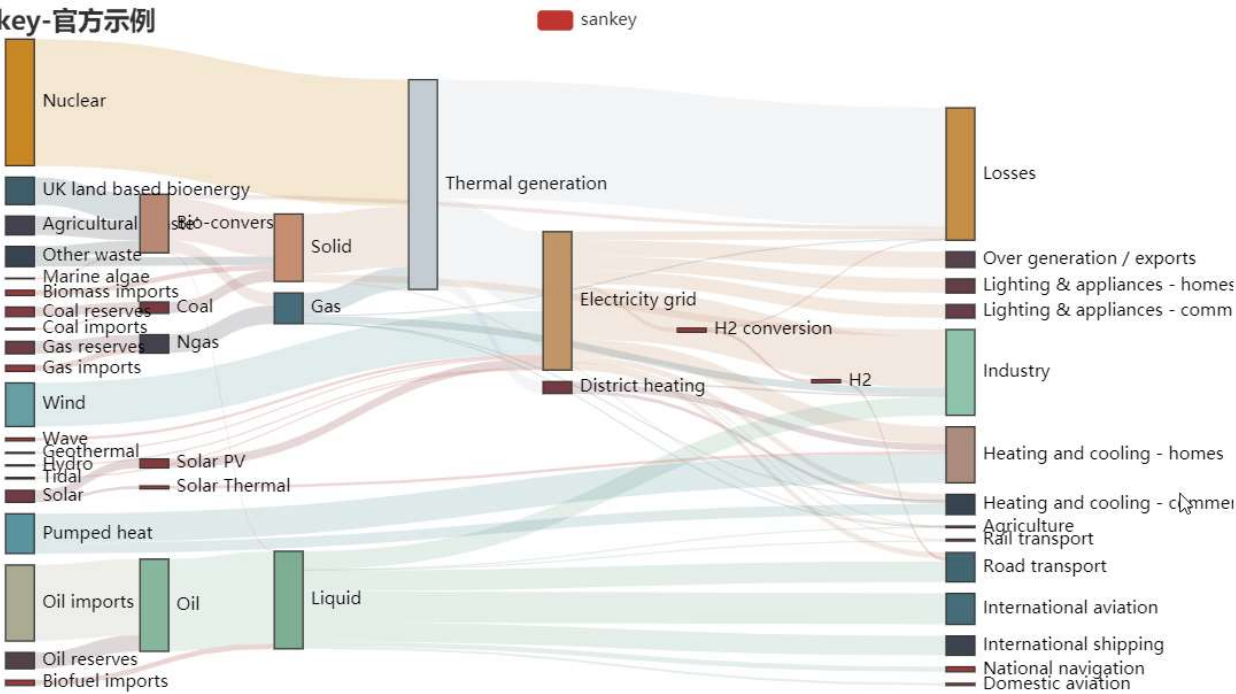


- [go-echarts](#): Para el caso de gráficos interactivos esta es una de las mejores opciones ya que además de los gráficos puedes crear tu propio dashboard con la misma librería.

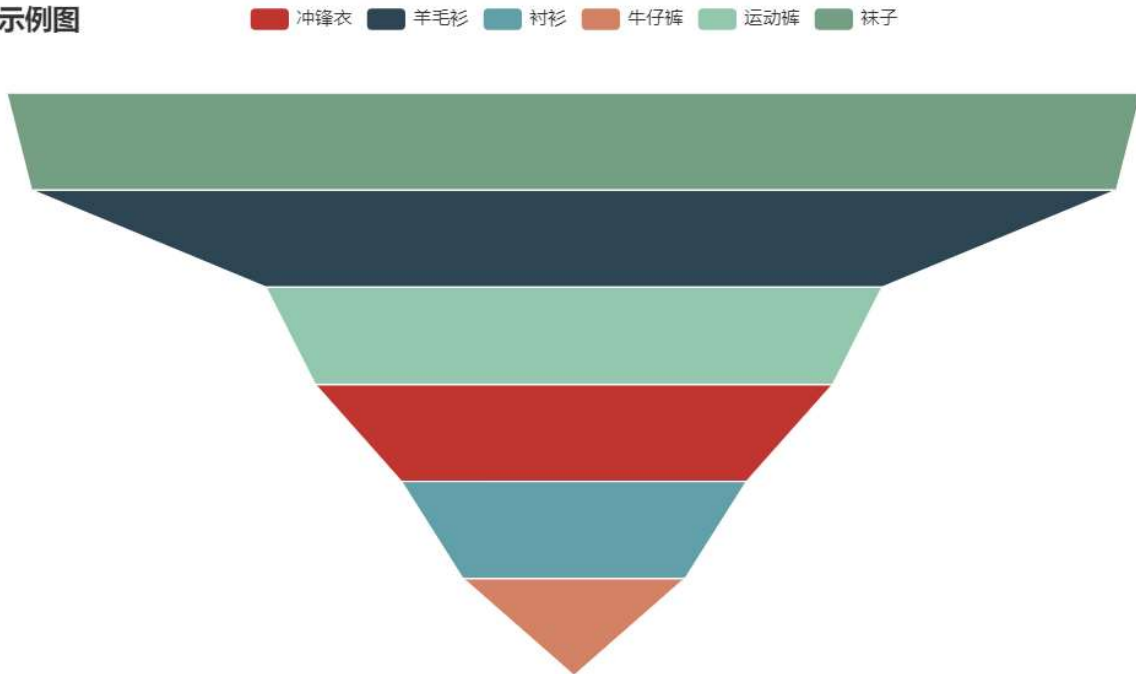
Bar3D-示例图



Sankey-官方示例



Funnel-示例图



Machine Learning

Hay muchas librerías este ámbito, sin embargo, destacaré solo dos:

- **GoLearn**: Tiene diferentes modelos, entre ellos lineales, regresiones y clasificación.
- **Gorgonia**: Es el más popular cuando requerimos implementar **Deep Learning** sino que además tiene la opción de implementar **CUDA** (hacer modelos de Deep Learning usando la tarjeta gráfica Nvidia)

La gran mayoría de estas librerías apenas están en sus inicios, pero tienen lo mínimo para iniciar en Data Science aplicando Go. Mi manera de apoyar a la comunidad es creando una librería para manejo de dataframes con Go llamada **Higor** usando mi experiencia como Data Engineer y haciendo que su implementación sea lo más amigable posible.

