

TRABAJOS PRÁCTICOS

Librerías útiles en Python y consejos presentación

TABLA DE CONTENIDOS

01. LIBRERÍAS ÚTILES

02. GRÁFICOS

03. CONFIGURACIÓN

04. PRESENTACIONES

05. INDUSTRIA

01

LIBRERÍAS ÚTILES



Operaciones rápidas en arrays

```
pip install numpy
```

```
import numpy as np
```

[Numpy Documentation](https://numpy.org/doc/stable/)



Librería para manejo de datos.

```
pip install pandas
```

```
import pandas as pd
```

[Pandas Documentation](https://pandas.pydata.org/pandas-docs/stable/10min.html)

De gran ayuda para el TP2 en el que hay manejar datasets grandes.



Realizar gráficos

```
pip install matplotlib
```

```
import matplotlib.pyplot as plt
```

[Matplotlib Documentation](https://matplotlib.org/)



Librería para realizar gráficos interactivos

```
pip install plotly
```

```
import plotly.express as px  
import plotly.graph_objects as go
```

[Plotly Documentación](#)



Librería para análisis de datos.

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
```

[Sklearn Documentation](#)

Para el TP4 de Aprendizaje No Supervisado el modulo de decomposition para Principal component analysis (PCA) y el de preprocessing para estandarizar features.



Librería de optimización.

```
from scipy.optimize import minimize
```

[Scipy Documentation](#)

Ofrece varios métodos como Powell, BFGS, Newton, entre otros.
En particular, usamos el de Powell para el TP5 sobre Deep Learning.



ARCADE

Librería para crear videojuegos en 2D.

```
pip install arcade
```

```
import arcade
```

[Arcade Documentation](https://arcade.academy/)

Sirve para diseñar el tablero del Sokoban en el TP1 de Métodos de Búsqueda. Es fácil de aprender y de usar



Librería para crear videojuegos.

```
pip install pygame
```

```
import pygame
```

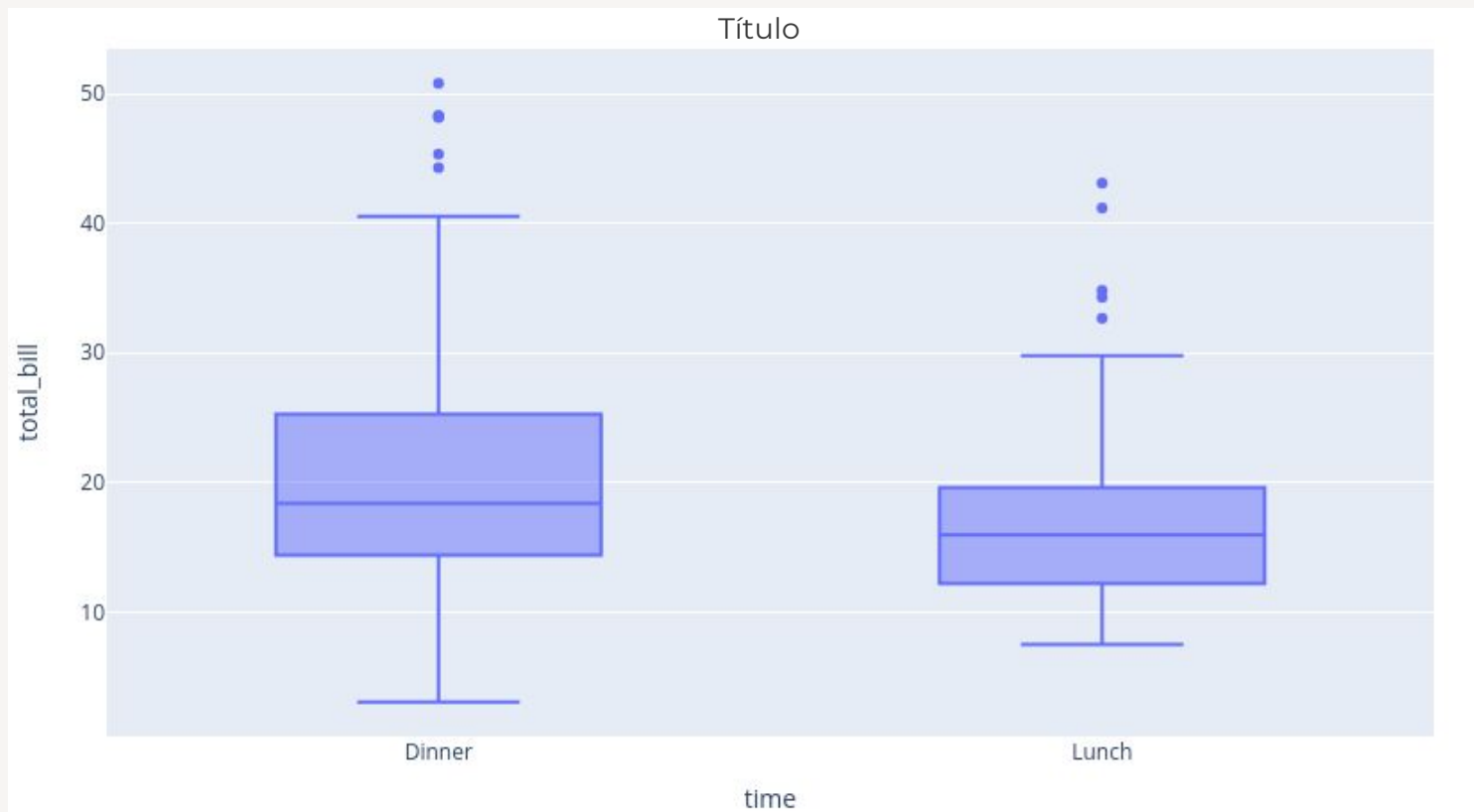
[Pygame Documentation](https://www.pygame.org/docs/)

Sirve para diseñar el tablero del Sokoban en el TP1 de Métodos de Búsqueda. Es fácil de aprender y de usar

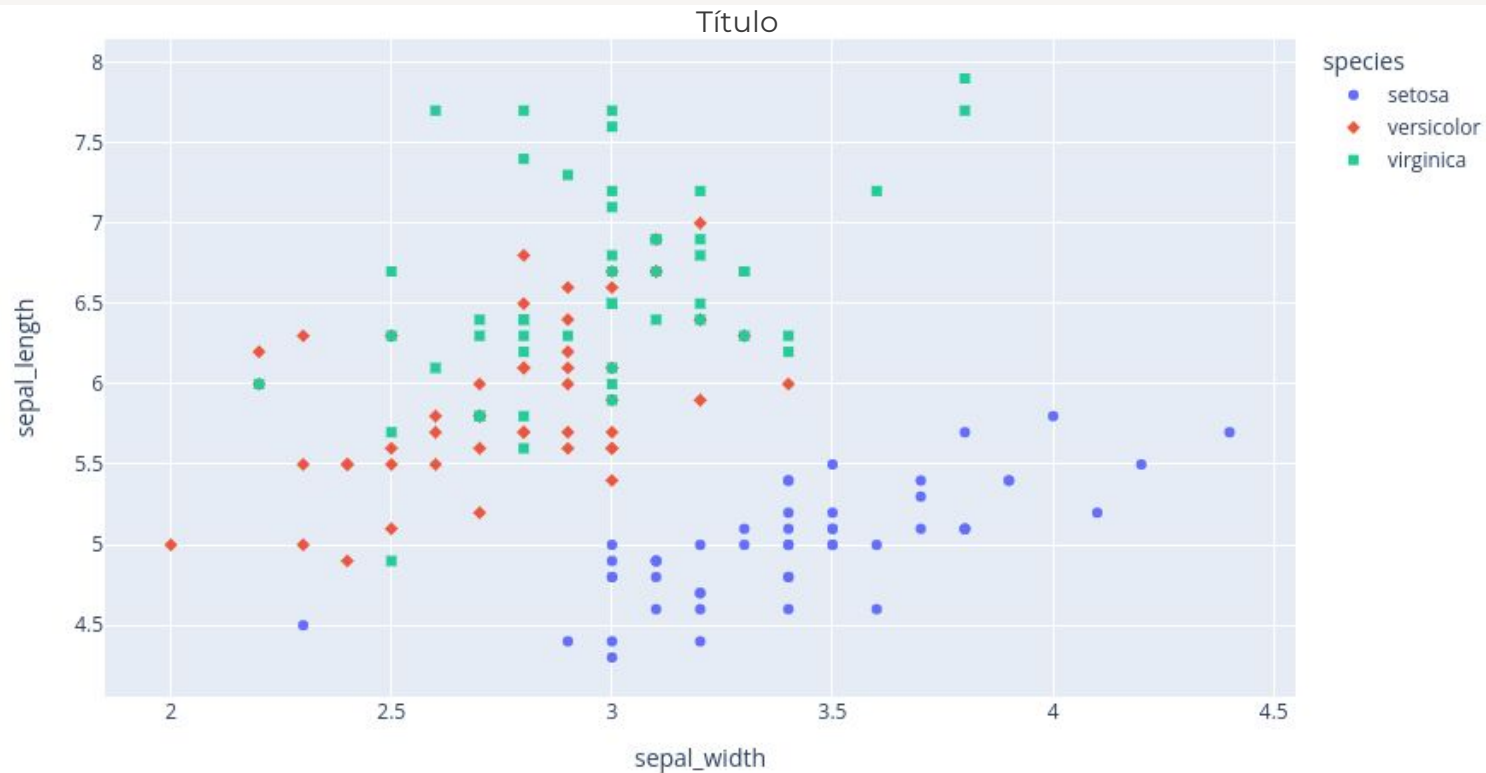
02

GRÁFICOS

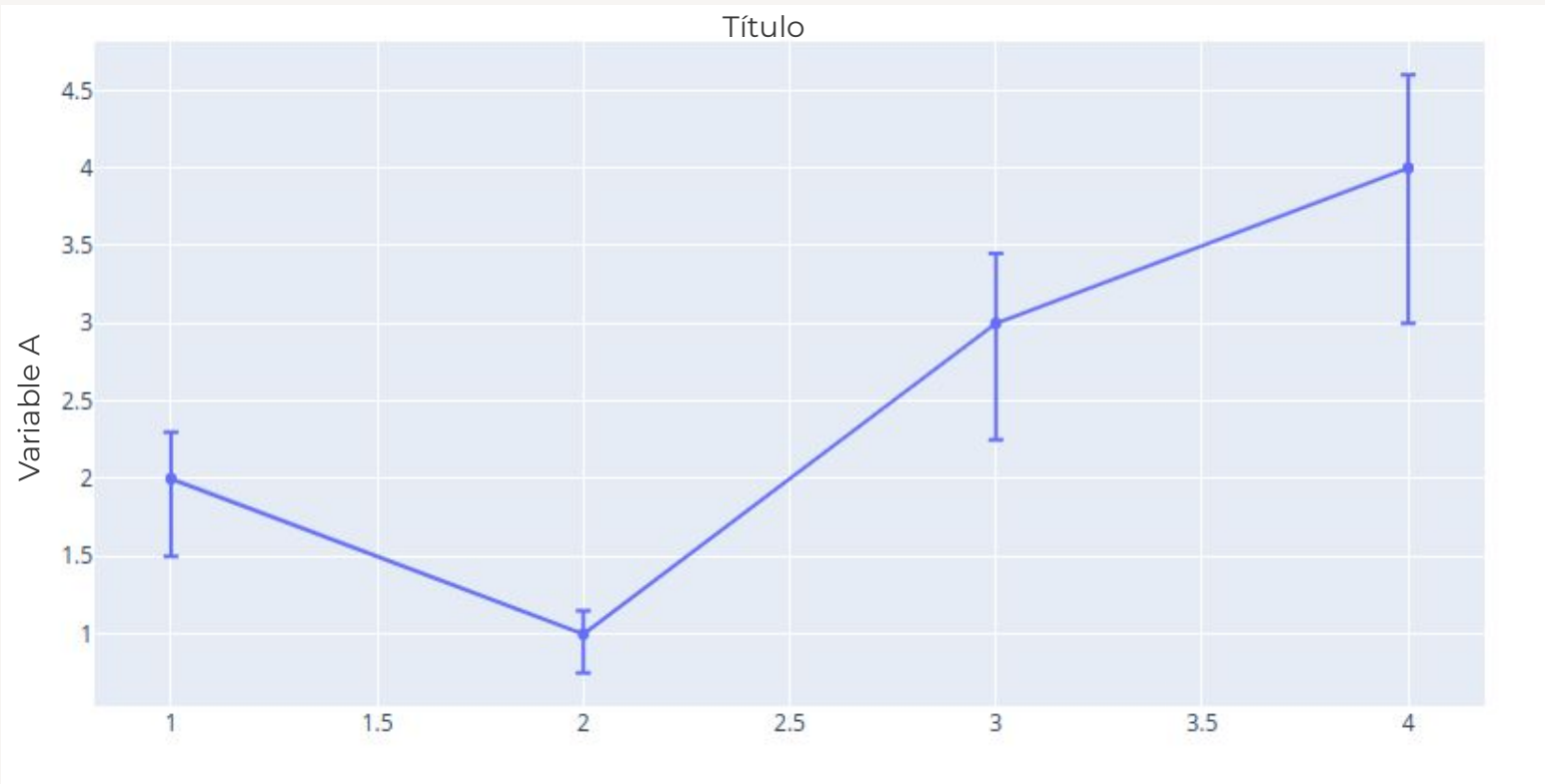
BOXPLOT



SCATTER PLOT



BARRAS DE ERROR



BARRAS DE ERROR

Gráfico de líneas con desvío estándar sombreado

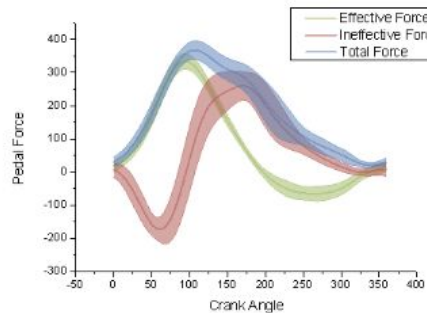


Gráfico de líneas con barras de error asimétricas

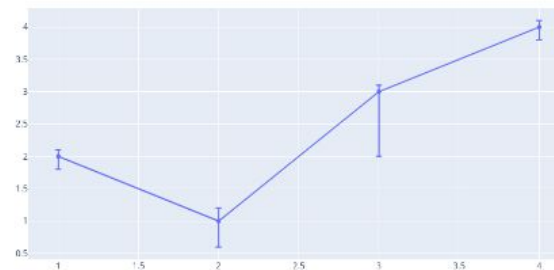
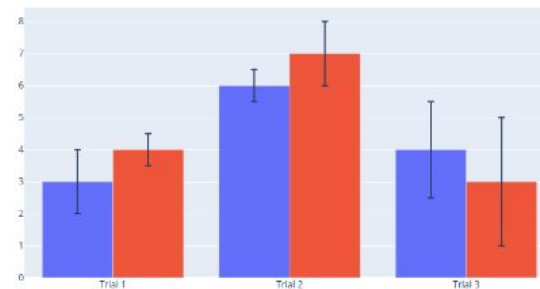


Gráfico de barras con barras de error



[PLOTLY](#)

[Seaborn](#)

03

ARCHIVO DE CONFIGURACIÓN

config.json

Example configurations: `config.json`

```
{
  "corpus": {
    "build": true,
    "name": "popular",
    "path": "../corpus/",
    "text_start": 0,
    "text_end": -1,
    "paging": 1
  },
  "postprocessing": {
    "word_cloud": true,
    "count_vectorizer": false,
    "tfidf": false,
    "word2vec": true
  }
}
```

Nota: Similar a config.yml

PRESENTACIONES

[Github SIA Tips](#)

- Presentación de 25 minutos + tiempo de preguntas individuales
- Orientada a los resultados
- Evaluación de parámetros
- Gráficos representativos de los resultados
- Promedio de ejecuciones
- Fijar un *seed* particular para reproducir resultados
- Conclusiones
- Entrega: repositorio incluyendo ppt, readme.md , config.json