



Portfolio curso programación con Python

Mayo'22 en CIFO – La Violeta

INGRID BAULÉS GIMÉNEZ

Temario (100h):

1. Paquets informàtics

Anaconda : Jupyter, Synder

2. Anàlisi de Dades

Notació matemàtica

Càlculs estadístics (math)

Tipus de gràfics estadístics (Plotly)

Més gràfics : MathPlotLib

Càlculs Matricials : NumPy

Anàlisi de dades: Pandas

3. Jocs

Jocs Arcade: PyGame

4. GUI (Interface gràfica d'usuari)

Tkinter

5. Admin. Bases de Dades

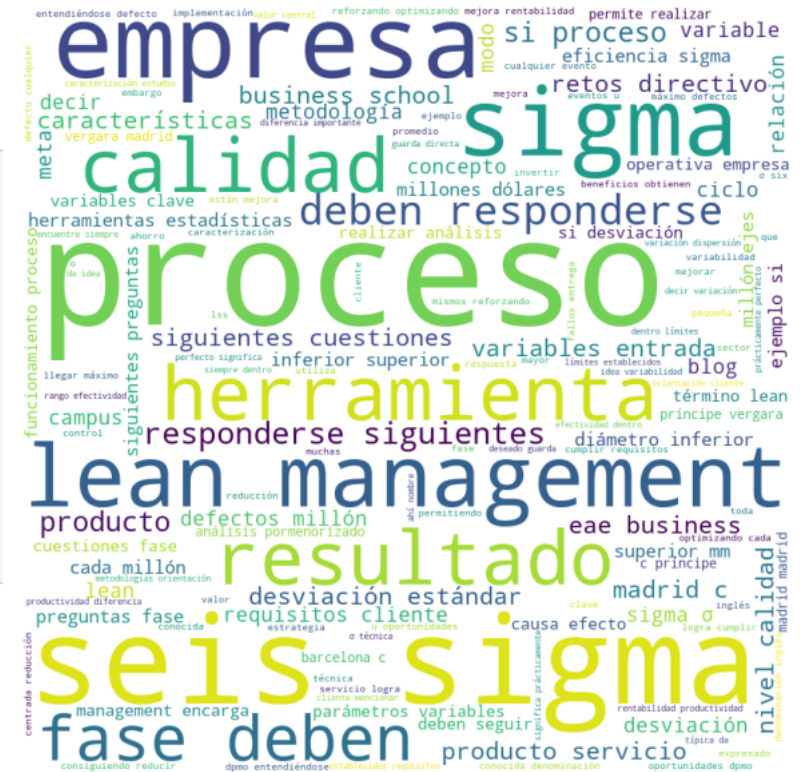
SQLite Studio

6. Web

Frameworks : web2py

Análisis de textos

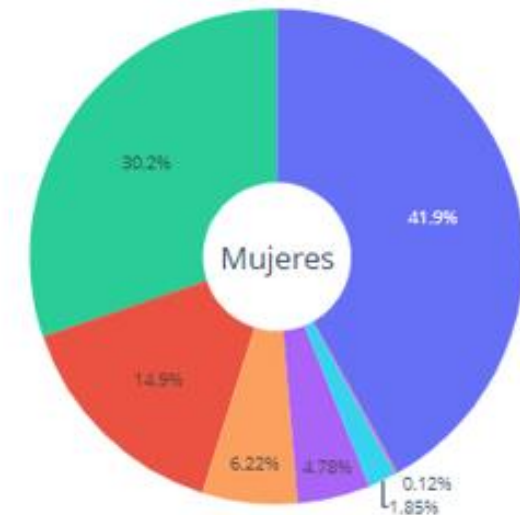
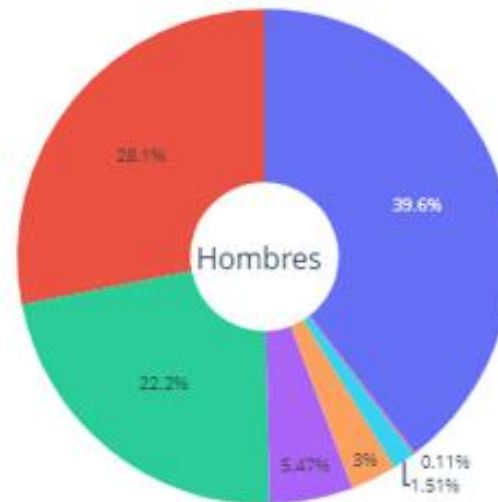
```
1 import numpy as np
2 import pandas as pd
3
4 word_cloud = WordCloud(height=800, width=800, background_color='white', max_words=150, min_font_size=5,
5                        collocation_threshold=10).generate(clean_texto)
6
7 word_cloud.to_file("./img/ejemplo_sencillo.png") #Guardamos la imagen generada
8
9 plt.figure(figsize=(10,8))
10 plt.imshow(word_cloud)
11 plt.axis('off')
12 plt.tight_layout(pad=0)
13 plt.show()
```



Análisis numéricos con gráficos

```
1 import plotly.graph_objects as go
2 from plotly.subplots import make_subplots
3
4 ##Labels = ["US", "China", "European Union", "Russian Federation", "Brazil", "India", "Rest of World"]
5 zona = ['Europa', 'África', 'América Nort', 'América Cent', 'América Sur', 'Asia', 'Oceanía']
6
7 # Create subplots: use 'domain' type for Pie subplot
8 fig = make_subplots(rows=1, cols=2, specs=[[{'type':'domain'}, {'type':'domain'}]])
9 fig.add_trace(go.Pie(labels=zona, values=[39.57, 28.11, 3, 1.51, 22.22, 5.47, 0.11], name="Hombres"),
10              1, 1)
11 fig.add_trace(go.Pie(labels=zona, values=[41.9, 14.89, 6.22, 1.85, 30.23, 4.78, 0.12], name="Mujeres"),
12              1, 2)
13
14 # Use `hole` to create a donut-like pie chart
15 fig.update_traces(hole=.3, hoverinfo="label+percent+name")
16
17 fig.update_layout(
18     title_text="Residentes por Zona origen datos 2001",
19     # Add annotations in the center of the donut pies.
20     annotations=[dict(text='Hombres', x=0.16, y=0.5, font_size=20, showarrow=False),
21                  dict(text='Mujeres', x=0.83, y=0.5, font_size=21, showarrow=False)])
22 fig.show()
```

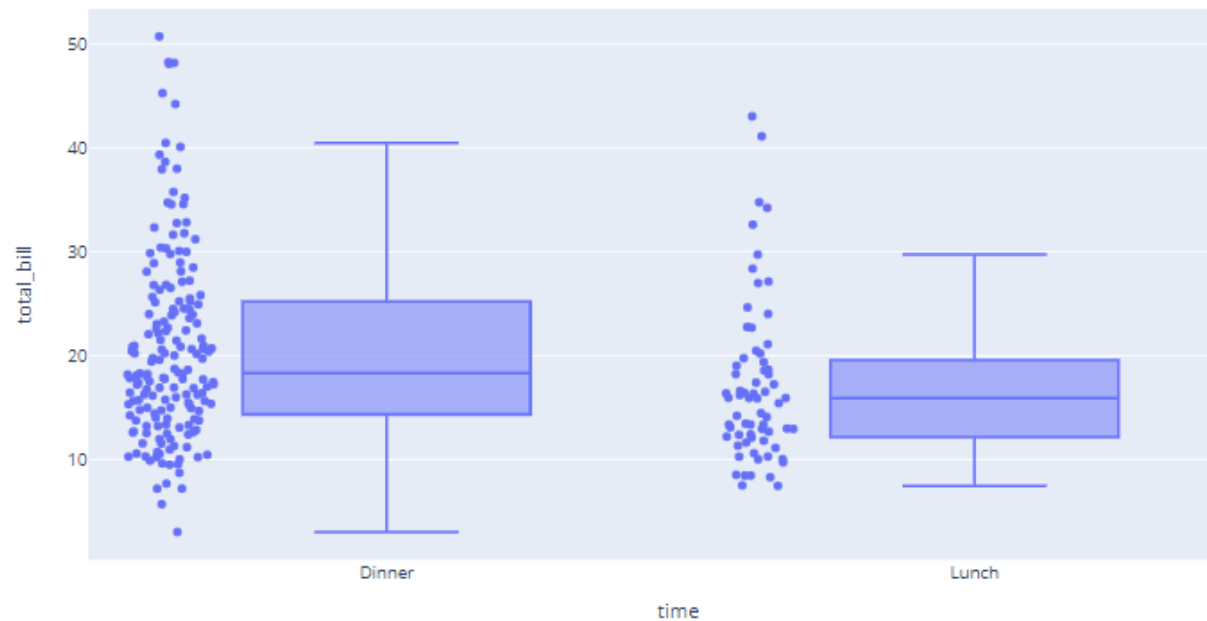
Residentes por Zona origen datos 2001



- Europa
- África
- América Sur
- Asia
- América Nort
- América Cent
- Oceanía

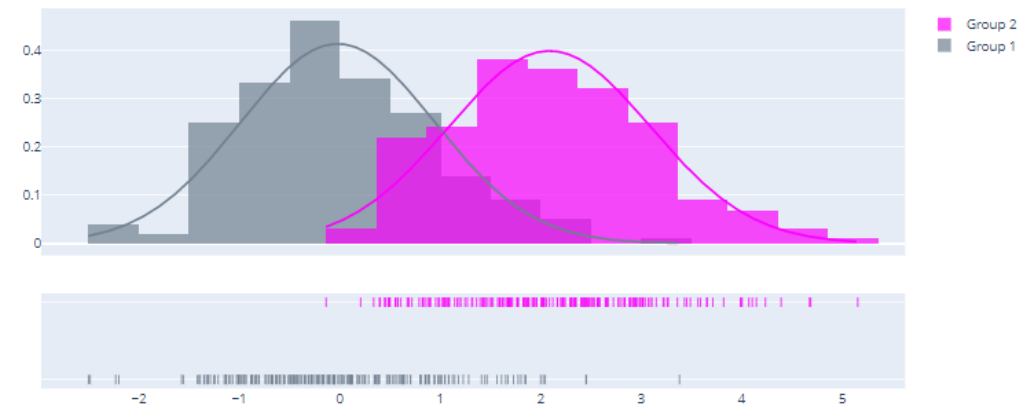
Análisis numéricos estadísticos

```
In [44]: 1 import plotly.express as px
2 df = px.data.tips()
3 fig = px.box(df, x="time", y="total_bill", points="all")
4 fig.show()
```



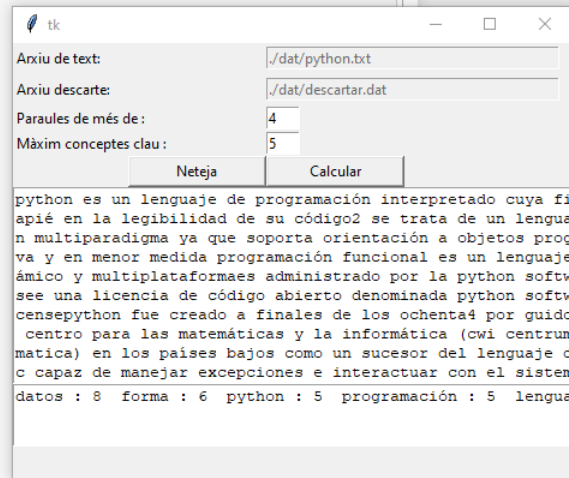
```
In [45]: 1 import plotly.figure_factory as ff
2 import numpy as np
3
4 x1 = np.random.randn(200)
5 x2 = np.random.randn(200) + 2
6
7 group_labels = ['Group 1', 'Group 2']
8
9 colors = ['slategray', 'magenta']
10
11 # Create distplot with curve_type set to 'normal'
12 fig = ff.create_distplot([x1, x2], group_labels, bin_size=.5,
13                          curve_type='normal', # override default 'kde'
14                          colors=colors)
15
16 # Add title
17 fig.update_layout(title_text='Distplot with Normal Distribution')
18 fig.show()
19
```

Distplot with Normal Distribution



GUI Interfaz Usuario: Windows, Frames, Labels, Entries, Buttons

```
In [*]: 1 from tkinter import *
2 from tkinter import filedialog as FileDialog
3
4 class LabEntry():
5
6     def __init__(self, window, text, row, **kwargs):
7
8         if text != "None":
9             Label(window, text = text).grid(row=row, column=0, sticky="w")
10
11         Entry(window, textvariable=vars[row-1], **kwargs).grid(row=row, column=1, sticky="w")
12
13
14
15 def llegir_arxiu(num):
16
17     contingut = ""
18
19     try:
20         nom_arxiu = vars[num].get()
21         f = open(nom_arxiu, "r", encoding="UTF-8")
22         contingut = f.read()
23         f.close()
24
25     except Exception as e:
26         print("Error: Arxiu no existeix", e)
27
28     return(contingut)
29
30 def netejar():
31     print(len(vars), vars)
32     for var in vars:
33         var.set("")
34
35 def calcular():
36
37     frame = Frame(width=800, height=600)
38     frame.place(in_=root, relx=.05, rely=.07)
39     frame.grid(row = 8, columnspan=3)
40
41     # Crear Zona de resultados res1 y res2 : texto bruto y frecuencias
42     res1 = Text(frame, width=75, height=10)
43     res1.grid(row=7, columnspan=3)
44
45     res2 = Text(frame, width=75, height=3)
46     res2.grid(row=8, columnspan=3)
47
48     # Leer fichero de texto en la variable texto
49
50     contingut = llegir_arxiu(0)
51     # pasarlo a minúsculas
52     contingut = contingut.lower().replace('\n', '')
53
54     # Quitar signos de puntuación
55     #
56     for i in list(".,;:-"):
57         contingut = contingut.replace(i, '')
58
59
```



```
57 # mostrar texto en la caja
58 res1.delete(1.0, 'end') # Nos aseguramos de que esté vacío
59 res1.insert('insert', contingut)
60
61 # Leer fichero de palabras a descartar
62 paraules_descarte = llegir_arxiu(1).split("\n")
63
64 # Calcular palabras diferentes y restar las palabras de descarte
65 paraules = contingut.split()
66 diferentes = list(set(paraules) - set(paraules_descarte))
67
68 # Calcular frecuencias y ordenarlas
69 veces = []
70
71 if vars[2].get().isnumeric():
72     lenmax = int(vars[2].get())
73 else:
74     lenmax = 3
75
76 for palabra in diferentes:
77     if len(palabra) > lenmax:
78         veces.append((contingut.count(palabra), palabra))
79 veces.sort(reverse=True)
80
81 # Seleccionar las n primeras
82 if vars[3].get().isnumeric():
83     numrank = int(vars[3].get())
84 else:
85     numrank = 8
86
87 resultado = ""
88 for i in range(0, numrank):
89     resultado += f"{veces[i][1]} : {veces[i][0]} \t"
90     print(veces[i])
91
92 res2.delete(1.0, 'end') # Nos aseguramos de que esté vacío
93 res2.insert('insert', resultado)
94
95 def navega_arxiu1():
96     fichero = FileDialog.askopenfilename( title="Arxiu de text")
97     vars[0].set(fichero)
98
99 def navega_arxiu2():
100     fichero = FileDialog.askopenfilename( title="Arxiu a descartar")
101     vars[1].set(fichero)
102
103 #----- Inicio
104 root = Tk()
105 root.geometry('800x600')
106
107 #----- crea un array de StringVar para las casillas de Entry
108 height = 4
109 vars = []
110 for i in range(height):
111     var = StringVar()
112     vars.append(var)
113
114 LabEntry(root, "Arxiu de text: ", 1, state=DISABLED, width=40)
115 LabEntry(root, "Arxiu descarte: ", 2, state=DISABLED, width=40)
116 LabEntry(root, "Paraules de més de : ", 3, width=4)
117 LabEntry(root, "Màxim conceptes clau : ", 4, width=4)
118 Button(root, text="...", command=navega_arxiu1, width=3).grid(row=1, column=3, sticky="e")
119 Button(root, text="...", command=navega_arxiu2, width=3).grid(row=2, column=3, sticky="e")
120 Button(root, text="Neteja", command=netejar, width=15).grid(row=6, column=0, sticky="e")
121 Button(root, text="Calcular", command=calcular, width=15).grid(row=6, column=1, sticky="w")
122
```