

1. Contexto

Para la continuacion de la practica, se realizara la **Matriz de Correlacion**, este siendo graficada con un *Mapa de Calor*, asi mismo un grafica de **Disepercion** para las variables que tengan mas relacion.

2. Importando librerias/modulos

Para poder realizar el objetivo de esta practica, se es necesario las librerias de **Pandas,MatlPlotLib y Seaborn**

```
In [ ]: # Importando Liberias
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#import numpy as np
```

3. Lectura del dataset

```
In [ ]: # Lectura del dataset
ruta = "../results/dataset_penguins_clean.csv"
data = pd.read_csv(ruta)
print(data)

data.columns #ver las comlumnas del CSV
data.describe() #ver variables numericas
data.isna() #ver datos fatantes
data.isna().sum() #contar la cantidade datos faltantes#
```

	rowid	species	island	bill_length_mm	bill_depth_mm	\
0	1	Adelie	Torgersen	39.1	18.7	
1	2	Adelie	Torgersen	39.5	17.4	
2	3	Adelie	Torgersen	40.3	18.0	
3	5	Adelie	Torgersen	36.7	19.3	
4	6	Adelie	Torgersen	39.3	20.6	
..	
328	340	Chinstrap	Dream	55.8	19.8	
329	341	Chinstrap	Dream	43.5	18.1	
330	342	Chinstrap	Dream	49.6	18.2	
331	343	Chinstrap	Dream	50.8	19.0	
332	344	Chinstrap	Dream	50.2	18.7	

	flipper_length_mm	body_mass_g	sex	year	edad_actual
0	181.0	3750.0	male	2007	16
1	186.0	3800.0	female	2007	16
2	195.0	3250.0	female	2007	16
3	193.0	3450.0	female	2007	16
4	190.0	3650.0	male	2007	16
..
328	207.0	4000.0	male	2009	14
329	202.0	3400.0	female	2009	14
330	193.0	3775.0	male	2009	14
331	210.0	4100.0	male	2009	14
332	198.0	3775.0	female	2009	14

[333 rows x 10 columns]

```
Out[ ]: rowid      0
species      0
island       0
bill_length_mm 0
bill_depth_mm 0
flipper_length_mm 0
body_mass_g   0
sex           0
year          0
edad_actual   0
dtype: int64
```

4. Filtro de las columnas numericas

Para poder realizar nuestros graficos de Calor y Dispersion, se es necesario solo tener en cuenta las columnas numericas, para este caso tendremos cinco columnas numericas

```
In [ ]: #Filtrando comLumnas numericas
filtro = ['bill_length_mm','bill_depth_mm',
          'flipper_length_mm', 'body_mass_g', 'year']
data=data[filtro]

#Mostrando e nuevo DF
data
```

Out[]:

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	year
0	39.1	18.7	181.0	3750.0	2007
1	39.5	17.4	186.0	3800.0	2007
2	40.3	18.0	195.0	3250.0	2007
3	36.7	19.3	193.0	3450.0	2007
4	39.3	20.6	190.0	3650.0	2007
...
328	55.8	19.8	207.0	4000.0	2009
329	43.5	18.1	202.0	3400.0	2009
330	49.6	18.2	193.0	3775.0	2009
331	50.8	19.0	210.0	4100.0	2009
332	50.2	18.7	198.0	3775.0	2009

333 rows × 5 columns

5. Graficando

Una matriz de correlación es una herramienta estadística que muestra la intensidad y la dirección de la relación entre dos o más variables. Se utiliza mucho en campos como las finanzas, la economía, la psicología y la biología, porque ayuda a entender cómo se relacionan entre sí distintas cosas.

In []: `#Creando la mtriz de correlacion`
`print("Matriz de Correlacion:\n")`
`matriz_correlacion=data.corr().round(2)`
`matriz_correlacion`

Matriz de Correlacion:

Out[]:

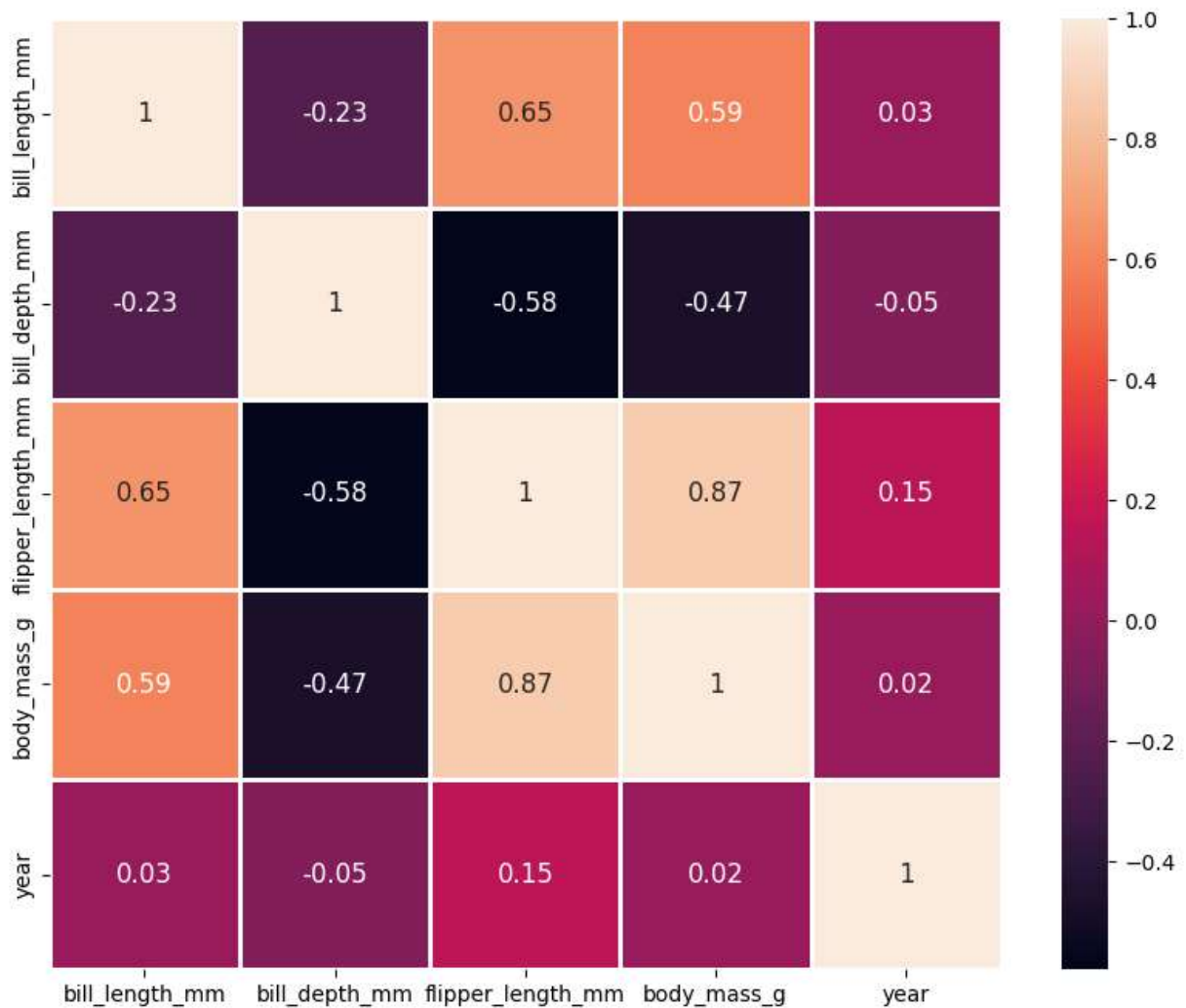
	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	year
bill_length_mm	1.00	-0.23	0.65	0.59	0.03
bill_depth_mm	-0.23	1.00	-0.58	-0.47	-0.05
flipper_length_mm	0.65	-0.58	1.00	0.87	0.15
body_mass_g	0.59	-0.47	0.87	1.00	0.02
year	0.03	-0.05	0.15	0.02	1.00



5.1 Mapa de Calor en Seaborn con heatmap (antes)

La función **heatmap** permite crear mapas de calor o gráficos de correlación en Python con seaborn. Puedes pasar un conjunto de datos 2D o un data frame de pandas. En caso de utilizar un data frame de pandas los nombres de las filas y las columnas serán usados como etiquetas para los ejes.

```
In [ ]: #ruta = "../results/dataset_penguins_clean.csv"
# Diagrama de Calor
plt.figure(figsize=(10, 8))
sns.heatmap(data=matriz_correlacion, # pasandoles los datos CSV con el nuevo DF
            annot=True, # Mostrar los datos dentro de los cuadros
            square=True, # Celdas cuadradas
            annot_kws={'size': 12}, # Tamaño de los datos en los cuadros
            linecolor='white', # Color del borde de los cuadros
            linewidths=0.75) # Grosor del borde de los cuadros
plt.show()
```



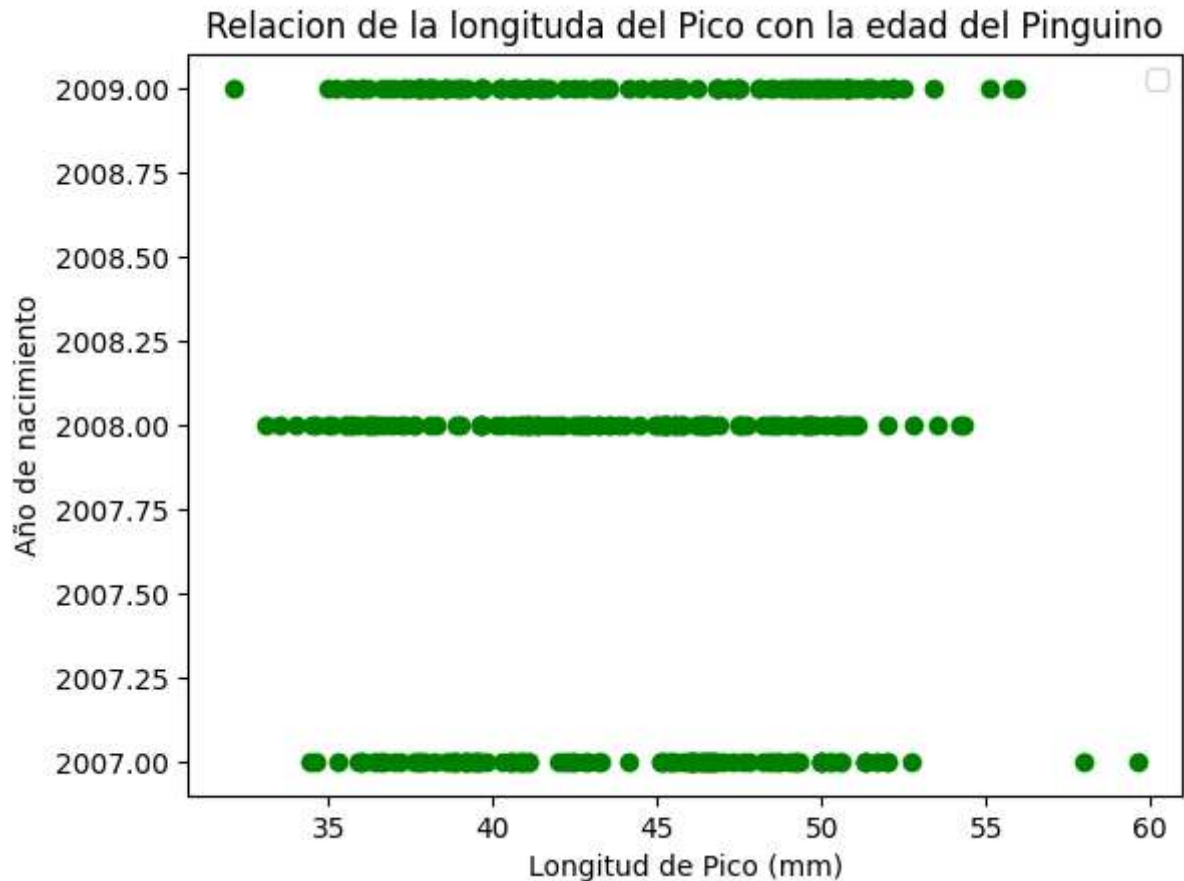
```
In [ ]: #Datos para el grafico de dispersion
x=data['bill_length_mm']
```

```

y=data['year']
plt.scatter(x,y, color="g")
plt.xlabel('Longitud de Pico (mm)')
plt.ylabel('Año de nacimiento')
plt.title("Relacion de la longitud del Pico con la edad del Pinguino")
plt.legend()
plt.show()

```

No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.



5.2 Mapa de Calor en Seaborn con heatmap (despues)

Anteriormente se hizo el mapa de calor sin la fecha calculada real contra la fecha actual, es decir, año en este mapa de calor, se vera la edad real del pinguino a fecha de hoy.

```

In [ ]: #ruta = "../results/dataset_penguins_clean.csv"
        #Filtrando comlumnas numericas
        filtro = ['bill_length_mm', 'bill_depth_mm',
                  'flipper_length_mm', 'body_mass_g', 'year', 'edad_actual']
        data=data[filtro]

```

```
#Mostrando e nuevo DF
data
```

```
Out[ ]:
```

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	year	edad_actual
0	39.1	18.7	181.0	3750.0	2007	16
1	39.5	17.4	186.0	3800.0	2007	16
2	40.3	18.0	195.0	3250.0	2007	16
3	36.7	19.3	193.0	3450.0	2007	16
4	39.3	20.6	190.0	3650.0	2007	16
...
328	55.8	19.8	207.0	4000.0	2009	14
329	43.5	18.1	202.0	3400.0	2009	14
330	49.6	18.2	193.0	3775.0	2009	14
331	50.8	19.0	210.0	4100.0	2009	14
332	50.2	18.7	198.0	3775.0	2009	14

333 rows × 6 columns

```
In [ ]: #Creando la mtriz de correlacion
print("Matriz de Correlacion:\n")
matriz_correlacion=data.corr().round(2)
matriz_correlacion
```

Matriz de Correlacion:

```
Out[ ]:
```

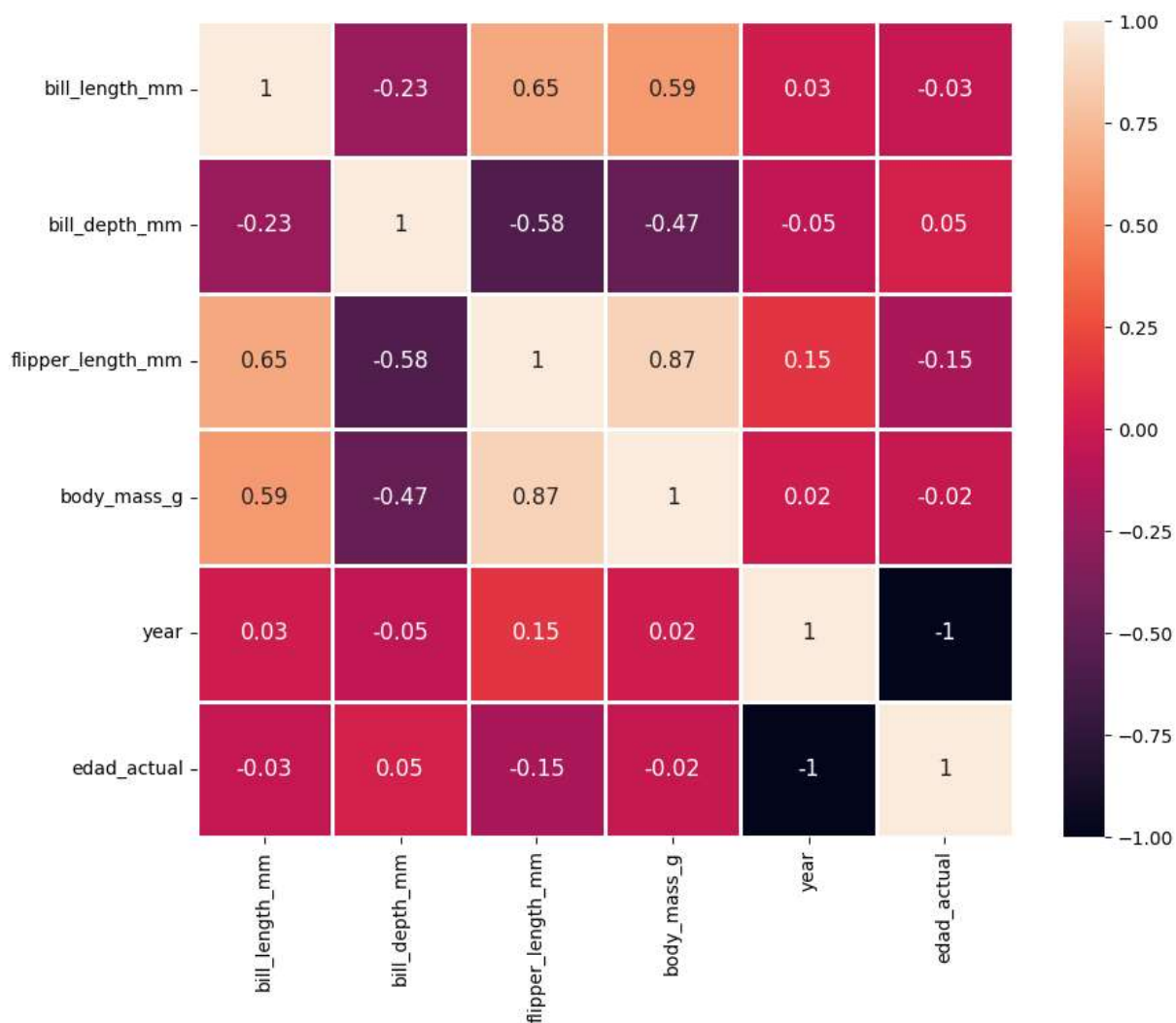
	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	year
bill_length_mm	1.00	-0.23	0.65	0.59	0.03
bill_depth_mm	-0.23	1.00	-0.58	-0.47	-0.05
flipper_length_mm	0.65	-0.58	1.00	0.87	0.15
body_mass_g	0.59	-0.47	0.87	1.00	0.02
year	0.03	-0.05	0.15	0.02	1.00
edad_actual	-0.03	0.05	-0.15	-0.02	-1.00

```
In [ ]: # Diagrama de Calor
plt.figure(figsize=(10, 8))
sns.heatmap(data=matriz_correlacion, # pasandoles los datos CSV con el nuevo DF
            annot=True, # Mostrar los datos dentro de los cuadros
            square=True, # Celdas cuadradas
            annot_kws={'size': 12}, # Tamaño de los datos en los cuadros)
```

```

linecolor='white', # Color del borde de Los cuadros
linewidths=0.75) # Grosor del borde de Los cuadros
plt.show()

```



```

In [ ]: #Datos para el grafico de dispersion
x=data['bill_length_mm']
y=data['edad_actual']
plt.scatter(x,y, color="g")
plt.xlabel('Longitud de Pico (mm)')
plt.ylabel('Edad Actual del Pinguino')
plt.title("Relacion de la longitud del Pico con la edad del Pinguino")
plt.legend()
plt.show()

```

No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.

