1. Contexto

En este conjunto de datos relacionados con los pingüinos implica examinar y visualizar los patrones y relaciones presentes en los datos. A continuacion, se busca respuesta a las siguientes preguntas:

1. ¿Cual es la vida promedio de un pinguino?
2. ¿Viven as las hembras o los machos?
3. ¿La altura es un rasgo distintivo del sexo?
4. ¿Cual es la proporción altura/ancho de los picos?
5. ¿Qué tipo de datos son las variables del conjunto de datos?
6. ¿Cuantas variables de cada tipo de dato tenemos en el conjunto de datos?
7. ¿Cuantas observaciones y variables tenemos en el conjunto de datos?
8. ¿Existen valores nulos explícitos en el conjunto de datos?
9. ¿De tener observaciones con valores nulos? ¿Cuántas tenemos por cada variable?
10. ¿Cuantos valores nulos tenemos en el total en el conjunto de datos?

2. El dataset

La informacion recolectada se encuentra en un archivo CSV (penguins), contiene 344 filas y 9 columnas. Despues de la limpieza veremos si se reducen las filas y/o la columnas de nuestro dataset.

3. Primer vistazo al dataset

In [ ]:
```
# Importamos librerias/modulos
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Leyendo el dataset
ruta = "../data/penguins.csv"
data = pd.read_csv(ruta)

# Mostra el dataset
print(data.shape)
data.head()
```

4. Limpieza del dataset

Se realizara el proceso de limpieza teniendo en cuenta las siguietnes situaciones comunes:

1. Datos faltantes en algunas celdas.
2. Columnas irrelevantes (ques no corresponden al problema que queremos resolver)
3. Registros (filas) repetidos.
4. Valores extremos (outliers) en el caso de las variables numericas. Se deben analizar en detalle pues no necesariamente la soulucion es eliminarlos.
5. Errores tipograficos en el caso de las variabe categoriccas.

Se supone que, al final de este proceso de limpieza deberiamos tener un set de datos integro, listo para la fase de Análisis Exploratorio.

4.1 Datos faltantes

Aqui damos comienzo a los puntos anteriormente mencionados. El numero total de filas son 344 y 9 columnas hasta el momento.

In [ ]:
```
data.dropna(inplace=True)
data.info()
```

4.2 Columnas irrelevantes

Una columnas irrelenvante contiene:

1. No contienen informcion relevante para el problema que queremos resolver.
2. Una columnas categoria pero con un solo nivel.
3. Una columna numerica pero con un solo valor.
4. Columnas con informacion redundante.

Pero si se tiene dudas sobre una columnas puede ser relevante o no, lo mejor es dejarla, y mas adelante en las sigueintes etapas, podremos darnos cuenta de si se preserva o no.

Todas las columnas categoricas, tienen que tener mas de un subnivel. Local no se elemina ninguna.

In [ ]:
```
# Conteo de niveles en las diferentes columnas categoricas
columnas_categoricas = ['species', 'island', 'sex']

for col in columnas_categoricas:
    print(f"Columna {col}: {data[col].nunique()} subniveles")
```

In [ ]:
```
# Veamos que ocurren con las columnas numericas
data.describe()
```

Como se muestra, hay 6 columnas que tienen desviacion estandar (std) diferentes de cero, lo que indica que no tienen un unico valor.

4.3 Filas repetidas

In [ ]:
```
print(f"Volumen del dataset antes de eliminar filas repetidas: {data.shape}")
data.drop_duplicates(inplace=True)
print(f"Volumen del dataset despues de eliminar filas repetidas: {data.shape}")
```

Como vemos, no hay filas repetidas, por ende, se mantienen las 344 filas y las 9 columas intactas.

4.4 Outliers (valores extremos) en variables numericas

No siempre se deben eliminar los Outliers porque dependiendo de la variable numérica analizada estos pueden contener informacion importante. Aqui pedemos usar algo asi como el sentido comun, dependiendo de nuestras varaibles.

In [ ]:
```
# Generamos graficas individuales para las variable numericas, pues estas
# tienen rangos diferentes
columnas_numericas = ['rowid', 'bill_length_mm',
                      'bill_depth_mm', 'flipper_length_mm', 'body_mass_g', 'year']

fig, ax = plt.subplots(nrows=6, ncols=1, figsize=(10, 25))
fig.subplots_adjust(hspace=0.5)

for i, col in enumerate(columnas_numericas):
    sns.boxplot(x=col, data=data, ax=ax[i])
    ax[i].set_title(col)
    ax[i].set_xticklabels(ax[i].get_xticklabels(), rotation=45)
```

Observaciones: Por lo que vemos es un dataset que no contiene outliers, por ende, no se elimina ninguna fila. Pero en la variable numerica *year* tiene datos erroneos que podemos eliminar.

4.5 Errores tipograficos en variables categoricas

En una variable categorica pueden aparecer subniveles que sen lo mismo, una palabra completa, mayusculas y/o vrebiaturas, y que pueden ser lo mismo, por ejemplo, div y division.

In [ ]:
```
# Graficamos los subniveles de cada variable categorica
columnas_categoricas = ['species', 'island', 'sex']

fir, ax = plt.subplots(nrows=3, ncols=1, figsize=(10, 25))
fig.subplots_adjust(hspace=1)

for i, col in enumerate(columnas_categoricas):
```

```
    sns.countplot(x=col, data=data, ax=ax[i])
    ax[i].set_xticklabels(ax[i].get_xticklabels(), rotation=30)
```

Observaciones: En caso que hubieran salido dos subniveles o mas similares con el tittulo, estos se deberia unificar

Si en la columna categorica **sex** hubieran dos subniveles que significaran lo mismo o que estuvieran mal escritos.

Por ejemplo: male y *males*, su unificacion y/o reemplazo seria de la siguiente manera:

print(data['sex'].unique()) >> *Mostrar la columna en cuestion antes de la unificacion/reemplazo.*

data['sex']=data['sex'].str.replace('males','male', regex=false) >> *Unificando/Reemplazando los subniveles.*

print(data['sex'].unique()) >> *Mostrar la columna en cuestion despues de la unificacion/reemplazo.*

Los resultados serian asi:

['males' 'male' 'female']

['male' 'female']

4.6 Exportado resultados

Listo, ya se ha completado la fase de limpieza del dataset, que originalmente tenia 344 filas y 9 columnas. El dataset resultante tiene 333 filas y 9 columnas.

El dataset ya esta listo para el Análisis Exploratorio.

In [ ]:
```
# Exportando los resultados en un nuevo archivo CSV
ruta = "../results/dataset_penguins_clean.csv"
data.to_csv(ruta, index=False)
```