
Exploración y visualización de datos con Python

Métodos Numéricos y Estadísticos
Grado en Ingeniería Informática / Mecánica

Curso 2022-2023

Eva María Mazcuñán Navarro



Eva María Mazcuñán Navarro
Departamento de Matemáticas
Universidad de León
E-mail: emmazn@unileon.es



Esta obra está bajo una [licencia de Creative Commons Reconocimiento 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Contenidos

	Página
Inicio	1
Introducción	1
Los pingüinos del archipiélago Palmer	1
Objetivos	1
1 Librerías	2
1.1 <code>pandas</code>	2
1.2 <code>seaborn</code>	2
2 Datos	3
2.1 Importar los datos	3
2.2 Dimensiones	4
2.3 Primeras y últimas filas	4
2.4 Estructura	5
2.5 Variables	6
2.6 Índice	9
3 Seleccionar variables	10
3.1 Seleccionar una variable	11
3.2 Seleccionar una lista de variables	11
4 Una variable categórica	12
4.1 El método <code>describe()</code>	13
4.2 Tabla de recuentos	14
4.3 Diagrama de recuentos	14
4.4 Personalización de los gráficos (opcional)	16
5 Una variable numérica	19
5.1 El método <code>describe()</code>	21
5.2 Histograma	22
5.3 Diagrama de caja y bigotes	23
6 Continuará . . .	24
Bibliografía	25

Introducción

En esta práctica aprenderás las técnicas básicas para explorar y visualizar un conjunto de datos con Python.

Los pingüinos del archipiélago Palmer

Presentaremos las diferentes técnicas a través de ejemplos trabajando con un conjunto de datos relativos a diferentes características de tres especies de pingüinos del archipiélago Palmer en la Antártida.

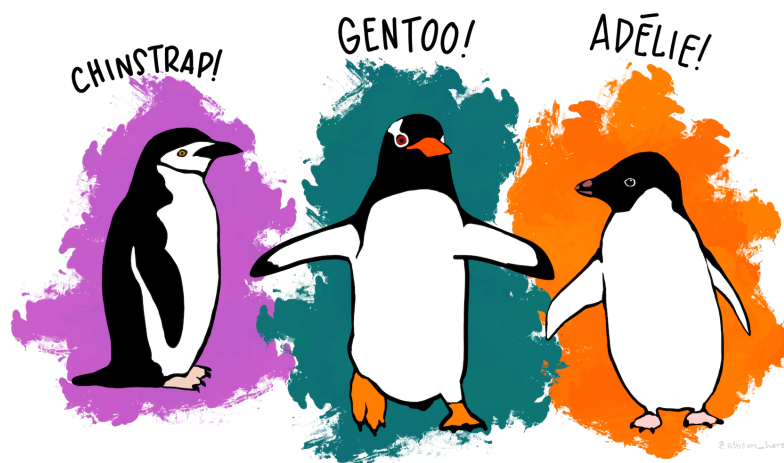


Figura 1: Ilustración de las tres especies de pingüinos del archipiélago Palmer (Artista @allison_horst)

Los datos fueron originalmente publicados en Gorman, Williams y Fraser [1]. Este conjunto de datos se hizo popular a partir de la creación del paquete [palmerpenguins](#) de [R](#). Hoy en día los datos de los pingüinos del archipiélago Palmer se usan de forma extendida para ilustrar las técnicas de exploración y visualización de datos no solo en R, sino en muchos otros lenguajes de programación para estadística y ciencia de datos, como Python. Nosotros accederemos a los datos a través de [este enlace](#), que proporciona los datos en formato CSV (*comma separated values*).

Objetivos

Aprenderás en concreto a calcular las medidas descriptivas más representativas de las características de interés y a crear diferentes tipos de gráficos o

visualizaciones.

1. Librerías

Lo primero que necesitamos hacer es importar las librerías de Python que usaremos a lo largo de práctica, que son **pandas** y **seaborn**:

```
import pandas as pd
import seaborn as sns
```

1.1. pandas



pandas es una librería que permite leer datos almacenados en estructuras similares a una tabla, como las hojas de cálculo o los archivos CSV, y proporciona métodos para explorar y describir esos datos.

Usando esta librería podremos calcular por ejemplo el peso medio de los pingüinos de cada especie. Obtendremos esta tabla:

body_mass_g	
species	
Adelie	3700.662252
Chinstrap	3733.088235
Gentoo	5076.016260

Puedes consultar la documentación oficial de **pandas** [aquí](#).

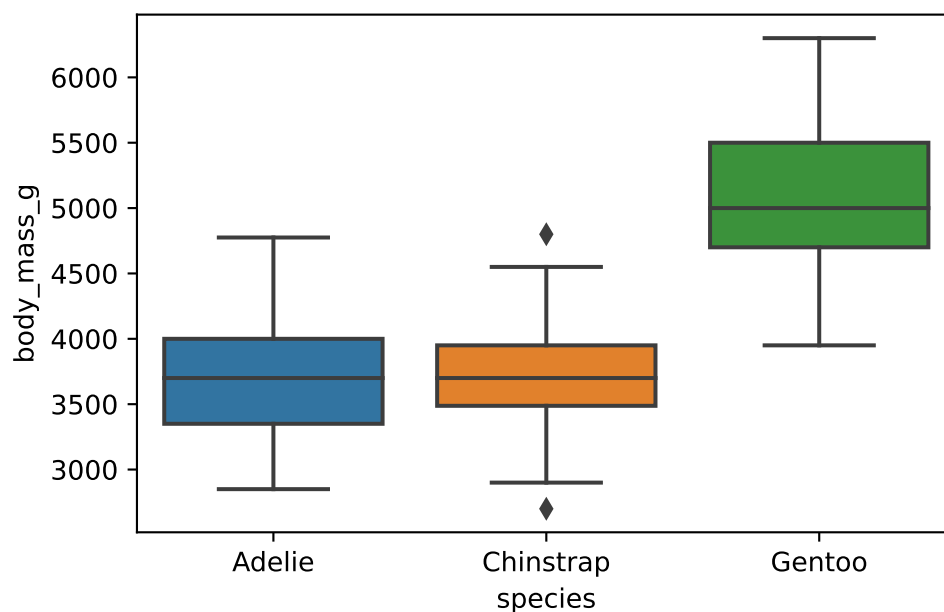
1.2. seaborn



seaborn es una librería para visualización de datos. Permite crear gráficos estadísticos muy informativos y visualmente atractivos con pocas líneas de código. Y está diseñado para trabajar con las estructuras de datos creadas con **pandas**.

Usaremos **seaborn** para realizar diferentes tipos de gráficos como diagramas de barras, histogramas, diagramas de caja y bigotes etc.

Aprenderemos por ejemplo a crear el siguiente gráfico, con los diagramas de caja y bigotes para el peso de los pingüinos de cada especie.



Puedes consultar la documentación oficial de **seaborn** [aquí](#).

2. Datos

En esta sección importarás los datos sobre los pingüinos del archipiélago Palmer presentados en la introducción y conocerás la información que contienen.

2.1. Importar los datos

Como se indicó en la introducción, los datos con los que vamos a trabajar están disponibles en la web en un fichero de formato CSV.

Ejecuta las instrucciones a continuación para importar el archivo usando la función `read_csv()` y guardar el resultado en una variable de nombre `penguins`:

```
url =  
↪ "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/penguins.csv"  
penguins = pd.read_csv(url)
```

El objeto `penguins` que acabas de crear es una **hoja de datos**, representada en `pandas` con la clase `DataFrame`.

```
type(penguins)
```

```
pandas.core.frame.DataFrame
```

En los siguientes apartados aprenderás a realizar una exploración inicial de la hoja de datos `penguins` que acabas de crear para conocer su estructura y la información que contiene.

2.2. Dimensiones

Una hoja de datos es una estructura matricial o tabular que contiene datos organizados por filas y columnas.

Para saber las dimensiones de nuestra hoja de datos `penguins` consulta su propiedad `shape`:

```
penguins.shape
```

```
(344, 7)
```

Vemos que nuestra hoja de datos tiene 344 filas y 7 columnas.

2.3. Primeras y últimas filas

Con las siguientes instrucciones puedes visualizar las cinco primeras y últimas filas de la hoja de datos `penguins` que acabas de crear.

```
penguins.head(5)
```


	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	Adelie	Torgersen	39.1	18.7	181.0	3750
1	Adelie	Torgersen	39.5	17.4	186.0	3800
2	Adelie	Torgersen	40.3	18.0	195.0	3250
3	Adelie	Torgersen	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450

```
penguins.tail(5)
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
339	Gentoo	Biscoe	NaN	NaN	NaN	NaN
340	Gentoo	Biscoe	46.8	14.3	215.0	4850
341	Gentoo	Biscoe	50.4	15.7	222.0	5750
342	Gentoo	Biscoe	45.2	14.8	212.0	5200
343	Gentoo	Biscoe	49.9	16.1	213.0	5400

2.4. Estructura

En nuestra hoja de datos `penguins`:

- Cada columna representa una variable asociada a una propiedad o característica de los pingüinos. Por ejemplo, la primera columna, de nombre `species` indica la especie (Chinstrap, Adélie o Gentoo) de pingüino. En el siguiente apartado se describen las otras seis variables.
- Cada fila se corresponde con un pingüino concreto de los 344 seleccionados en el estudio.
- Cada celda contiene el valor de la característica del pingüino en la correspondiente fila.

Por ejemplo, mirando la primera fila de la hoja de datos

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	Adelie	Torgersen	39.1	18.7	181.0	3750

vemos en la primera celda que el primer pingüino del listado es de la especie Adelie.

Unos mismos datos pueden organizarse o presentarse de diferentes maneras en diferentes hojas de datos. Para que sea sencillo trabajar con una hoja de datos es conveniente que haya una relación clara entre su significado y su estructura. Se considera que la hoja de datos está *ordenada* o *limpia* (en inglés se habla de *tidy data*) si está organizada de acuerdo con los siguientes principios:

- Cada **columna** representa una **variable** o característica de interés.
- Cada **fila** representa una **observación**, caso o unidad experimental.
- Cada **celda** contiene un **valor**, el de la variable en la correspondiente columna para la observación en la correspondiente fila.

De acuerdo con la descripción inicial, nuestra hoja de datos cumple con los principios anteriores.

2.5. Variables

Ejecuta la siguiente instrucción:

```
penguins.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   species               344 non-null   object
1   island                344 non-null   object
2   bill_length_mm        342 non-null   float64
3   bill_depth_mm         342 non-null   float64
4   flipper_length_mm     342 non-null   float64
5   body_mass_g           342 non-null   float64
6   sex                   333 non-null   object
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

La salida del método `info()` nos da una tabla con información sobre las siete variables de nuestra hoja de datos.

2.5.1. Descripción

En la columna de la tabla de nombre `Column` se lista el nombre de las siete variables en `penguins`. El significado de las variables es el siguiente:

Nombre	Descripción
<code>species</code>	Especie de pingüinos (Chinstrap, Adélie o Gentoo)
<code>island</code>	Nombre de la isla del archipiélago Palmer (Dream, Torgersen o Biscoe)
<code>bill_length_mm</code>	Longitud del pico, en milímetros (ver Figura 2)
<code>bill_depth_mm</code>	Anchura del pico, en milímetros (ver Figura 2)
<code>flipper_length_mm</code>	Longitud de las alas
<code>body_mass_g</code>	Peso en gramos
<code>sex</code>	Sexo (MALE o FEMALE)

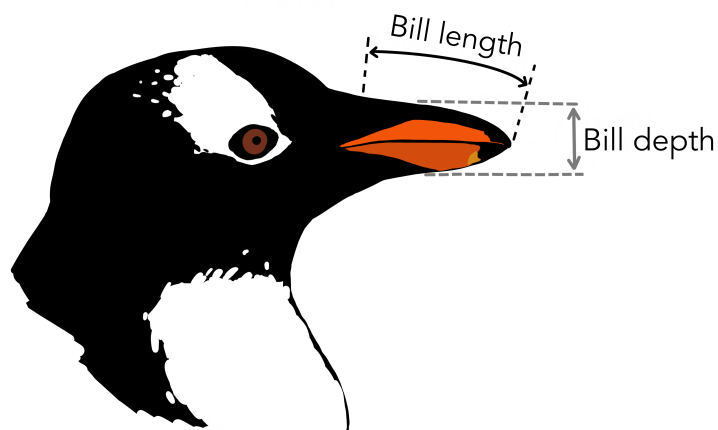


Figura 2: Ilustración de las variables `bill_length_mm` y `bill_depth_mm` (Artista @allison_horst)

Volviendo a mirar la primera fila de nuestra hoja de datos

	<code>species</code>	<code>island</code>	<code>bill_length_mm</code>	<code>bill_depth_mm</code>	<code>flipper_length_mm</code>	<code>body_mass_g</code>
0	Adelie	Torgersen	39.1	18.7	181.0	3750

ahora sabes que el primer pingüino es de la especie Adelie, vive en la isla Torgensen, las dimensiones de su pico son 39.1×18.7 milímetros, sus alas miden 181 milímetros, pesa 3 kilos y 750 gramos, y es un macho.

Problema 1

Describe las características del tercer pingüino del estudio (índice 2).

2.5.2. Valores nulos

Fíjate ahora en la columna Non-Null Count de la salida del método `info()`:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   species               344 non-null   object
1   island                344 non-null   object
2   bill_length_mm        342 non-null   float64
3   bill_depth_mm         342 non-null   float64
4   flipper_length_mm     342 non-null   float64
5   body_mass_g           342 non-null   float64
6   sex                   333 non-null   object
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

Los valores nulos o perdidos son valores no disponibles, que no han podido registrarse, se representan con el símbolo **NaN** (iniciales de *Not A Number*).

Si vuelves a mirar las cinco primeras filas de la hoja de datos verás que para el cuarto pingüino sólo sabemos que es de la especie Adelie y vive en la isla Torgensen, y se desconocen las otras cinco variables.

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
3	Adelie	Torgensen	NaN	NaN	NaN	NaN

Los diferentes métodos de la librería **pandas** contemplan la posibilidad de que las hojas de datos tengan valores nulos y los tratan de una forma predeterminada (por ejemplo, la media los ignora por defecto).

2.5.3. Tipos de variables

Las siete variables de nuestra hoja de datos se dividen en dos clases:

1. **species**, **island** y **sex** son **variables categóricas**. Este tipo de variables representan una característica cualitativa que puede tomar un número finito y fijo de valores, denominados categorías o niveles.
2. Las cuatro restantes, **bill_length_mm**, **bill_depth_mm**, **flipper_length_mm** y **body_mass_g** son **variables numéricas**, que representan características cuantitativas que se describen con valores numéricos (números enteros o reales).

pandas asigna un tipo a cada variable de una hoja de datos en función de los valores que presenta, como puede verse en la columna **Dtype** de la salida del método **info()**:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   species                344 non-null   object
1   island                 344 non-null   object
2   bill_length_mm         342 non-null   float64
3   bill_depth_mm          342 non-null   float64
4   flipper_length_mm      342 non-null   float64
5   body_mass_g            342 non-null   float64
6   sex                    333 non-null   object
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

2.6. Índice

Igual que cada columna (variable) en una hoja de datos tiene un nombre, cada fila (observación) también tiene una etiqueta identificativa. En nuestra hoja de datos cada uno de los 344 pingüinos se identifica con un número entero de la secuencia 0, 1, ..., 333.

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass
0	Adelie	Torgersen	39.1	18.7	181.0	375
1	Adelie	Torgersen	39.5	17.4	186.0	380
2	Adelie	Torgersen	40.3	18.0	195.0	323

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass
341	Gentoo	Biscoe	50.4	15.7	222.0	575
342	Gentoo	Biscoe	45.2	14.8	212.0	520
343	Gentoo	Biscoe	49.9	16.1	213.0	540

Las etiquetas identificativas de las filas de una hoja de datos forman su **índice**. El índice de una hoja de datos de **pandas** se registra en su propiedad **index**.

```
penguins.index
```

```
RangeIndex(start=0, stop=344, step=1)
```

Si cada pingüino estuviera identificado por un código, podríamos haber indicado esa variable como índice en el momento de la importación de los datos. Cuando no se indica el índice de una hoja de datos de forma explícita, **pandas** asigna una secuencia de números enteros comenzando en 0, como ha ocurrido en nuestro caso.

3. Seleccionar variables

Al estudiar un conjunto de datos, es frecuente tener que seleccionar los datos relevantes para responder a las diferentes cuestiones planteadas.

Si por ejemplo queremos saber cuál es el peso máximo de todos los pingüinos del estudio, seleccionaremos la variable **body_mass_g** y después calcularemos su máximo.

En esta sección aprenderás los métodos para seleccionar variables de una hoja de datos.

3.1. Seleccionar una variable

Utiliza la siguiente instrucción para seleccionar la variable `body_mass_g`:

```
mass = penguins["body_mass_g"]
```

i Para seleccionar una sola variable, usa corchetes `[]` e indica el nombre de la columna de interés.

Ahora podemos aplicar la función `max()` para obtener el peso máximo:

```
mass.max()
```

6300.0

Vemos que el pingüino más pesado del estudio pesa 6 kilos y 300 gramos.

Podemos realizar las dos operaciones anteriores, seleccionar la variable `body_mass_g`, y calcular su máximo con una sola instrucción:

```
penguins["body_mass_g"].max()
```

6300.0

Obtenemos el mismo resultado de antes.

Problema 2

Calcula el peso medio de todos los pingüinos (función `mean()`).

Problema 3

Calcula el valor mínimo para la longitud de las alas de todos los pingüinos (función `min()`).

3.2. Seleccionar una lista de variables

Para seleccionar las dos variables relativas a las dimensiones del pico, `bill_length_mm` y `bill_depth_mm`, ejecuta la siguiente instrucción:

```
bill = penguins[["bill_length_mm", "bill_depth_mm"]]
```

i Para seleccionar una lista de variables, usa corchetes `[]` adicionales para crear la lista con los nombres de las columnas de interés (los corchetes exteriores indican que se van a seleccionar datos y los interiores crean la lista).

Ahora podemos calcular la media para ambas variables con

```
bill.mean()
```

	0
bill_length_mm	43.92193
bill_depth_mm	17.15117

Vemos que los picos de los pingüinos tiene una longitud media de 43.92 milímetros y una anchura media de 17.15 milímetros.

Problema 4

Calcula el número de observaciones no nulas (función `count()`) para las variables `species` y `body_mass_g` con una sola línea de código.

4. Una variable categórica

En este apartado se describen los métodos básicos para explorar y analizar una variable categórica.

Analizaremos en concreto la variable `species` de nuestra hoja de datos, para conocer la distribución de los pingüinos por especies.

Comenzamos seleccionando la variable en nuestra hoja de datos. Almacenamos el resultado en un nuevo objeto de nombre `species`.

```
species = penguins["species"]
```


4.1. El método `describe()`

Para obtener la información más relevante sobre la distribución de las especies de pingüinos aplicamos a `species` el método `describe()`:

```
species.describe()
```

species	
count	344
unique	3
top	Adelie
freq	152

De la salida anterior obtenemos la siguiente información sobre la variable `species`:

Fragmento de la salida		Significado
count	344	Hay 344 valores no nulos, así que se conoce la especie de todos los pingüinos del estudio.
unique	3	La variable toma tres valores diferentes, es decir, hay tres categorías (las tres especie Adelie, Chinstrap y Gentto).
top	Adelie	La categoría <code>top</code> o más frecuente, es decir, la especie más numerosa es la especie Adelie
freq	152	El número de pingüinos de la especie Adelie (la categoría <code>top</code>) es 152

Problema 5

Utiliza el método `describe()` para obtener información sobre la distribución de los pingüinos por islas.

4.2. Tabla de recuentos

Ya sabemos que la especie más numerosa es Adelie, con 152 pingüinos, pero aún no sabemos cuántos pingüinos hay de las otras dos especies, Chinstrap y Gentoo. Para obtener una tabla con el número de pingüinos de cada especie usamos la función `value_counts()`:

```
species.value_counts()
```

species	
Adelie	152
Gentoo	124
Chinstrap	68

Ahora sabemos que la segunda especie más numerosa es Gentoo, con 124 ejemplares, y que solo hay 68 pingüinos de la especie Chinstrap.

Problema 6

Determina el número total de hembras y de machos.

4.3. Diagrama de recuentos

Un **diagrama de recuentos** asocia a cada categoría una barra de longitud igual al número de observaciones en esa categoría.

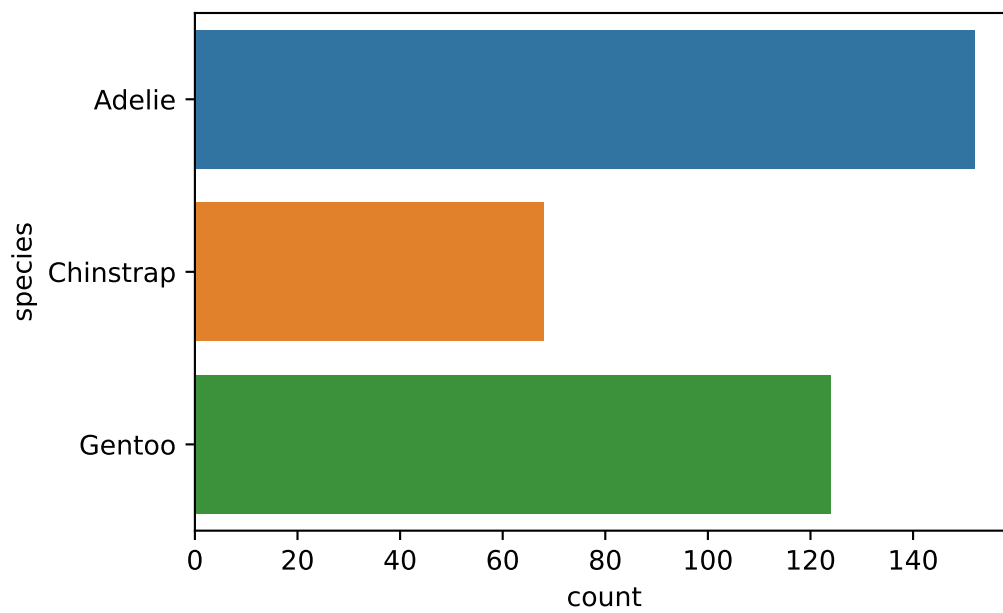
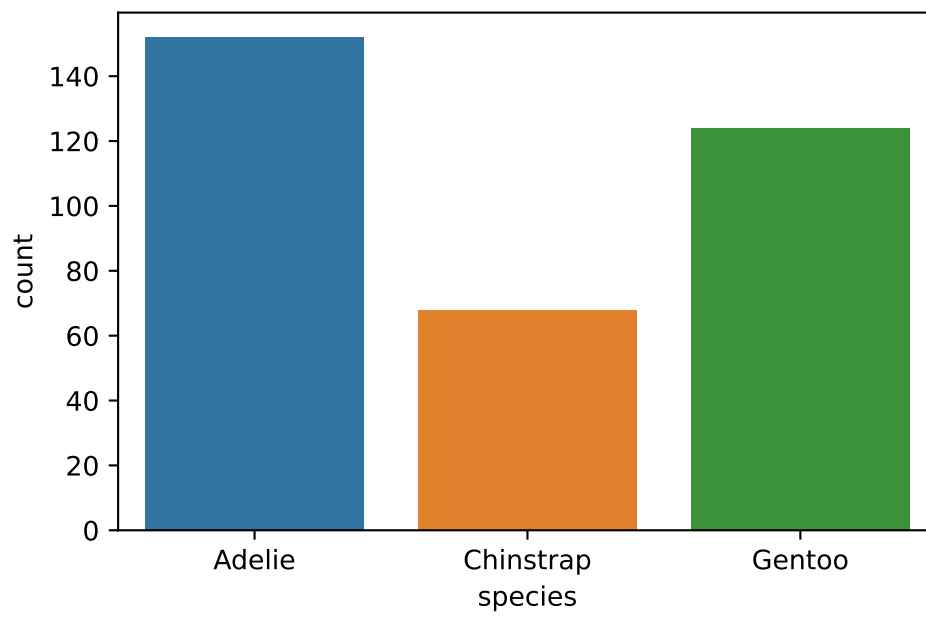
Para realizar un diagrama del recuento de pingüinos por especie usamos la función `countplot()` del paquete `seaborn`

```
sns.countplot(data=penguins, x="species");
```

Nota las alturas de las barras en el gráfico anterior coincidan con los recuentos que hemos calculado en el apartado anterior con `value_counts()`.

A veces es preferible usar barras horizontales, porque se tiene más espacio para las etiquetas de las categorías.

```
sns.countplot(data=penguins, y="species");
```



- i** Para realizar un diagrama de recuentos de las categorías de una variable categórica, usa la función `countplot()` de `seaborn`. E indica:
- El nombre de la hoja de datos como valor del argumento `data`.
 - El nombre de la variable categórica como valor del argumento `x` si quieres barras verticales, o como valor del argumento `y` si quieres barras horizontales.

- i** Si quitas el punto y coma `;` al final de las instrucciones anteriores aparecerá el mensaje `<AxesSubplot:xlabel='count', ylabel='species'>` en la salida antes del gráfico. Al terminar una instrucción con punto y coma `;` se inhibe la impresión de la salida.

Problema 7

Crea un diagrama de recuentos para el número de hembras y machos, con barras verticales.

Problema 8

Crea un diagrama de recuentos para el número de pingüinos en cada isla, con barras horizontales.

4.4. Personalización de los gráficos (opcional)

Los gráficos de la librería `seaborn` admiten muchas opciones para personalizar su aspecto cambiando por ejemplo los colores, los rótulos de los ejes, etc.

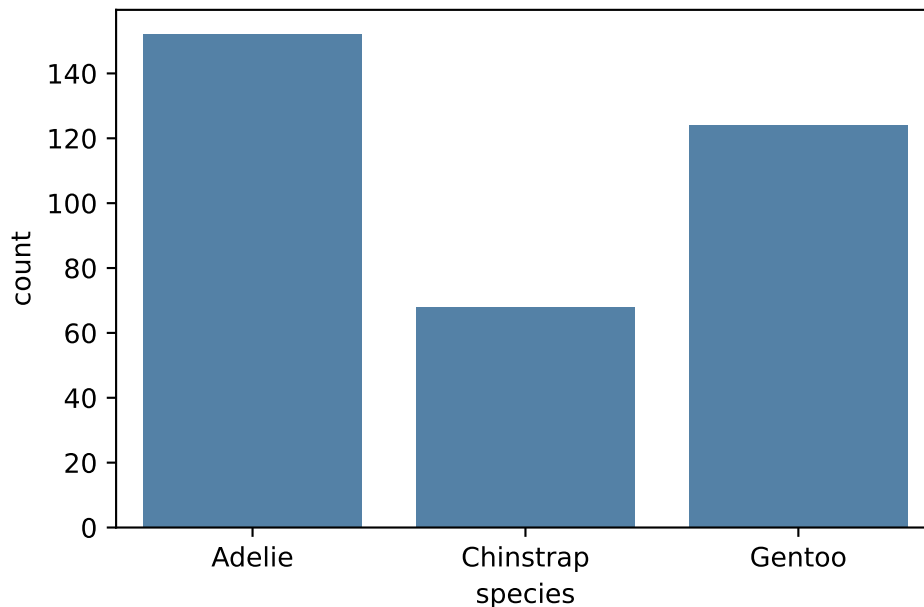
La personalización de los gráficos no carece de importancia, siendo especialmente relevante dar títulos descriptivos a los ejes. Pero en esta práctica nos centraremos en los procedimientos para realizar los gráficos y en la mayoría de ocasiones omitiremos los detalles de personalización de los mismos, que pueden consultarse en la documentación de `seaborn`.

En este apartado se da una muestra de las opciones para personalizar los diagramas de recuentos que se han presentado en el apartado anterior. Se trata de un apartado opcional y puedes por ahora omitir su lectura y volver a él al final de la sesión.

4.4.1. Colores

Si quieres un mismo color para todas las barras usa en argumento `color`:

```
sns.countplot(data=penguins, x="species",  
↪ color="steelblue");
```



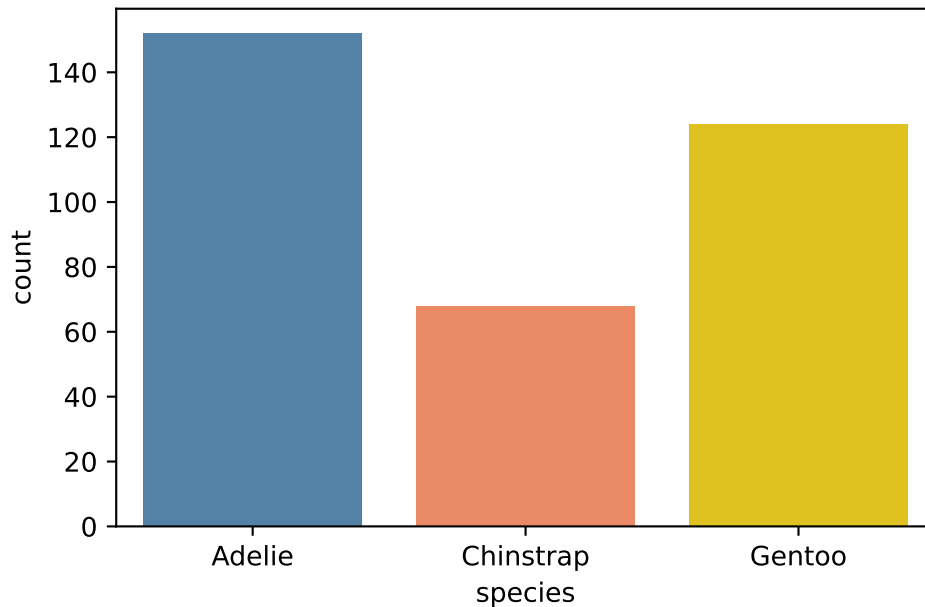
Puedes ver los colores disponibles [aquí](#).

También se puede indicar una secuencia de colores usando el argumento `palette`:

```
sns.countplot(data=penguins, x="species",  
↪ palette=["steelblue", "coral", "gold"]);
```

Como valor para `palette` se puede indicar una lista de colores, como en el ejemplo anterior, o el nombre de una de las paletas predefinidas (`deep`, `muted`, `pastel`, `bright`, `dark` y `colorblind`), como en el siguiente ejemplo:

```
sns.countplot(data=penguins, x="species",  
↪ palette="colorblind");
```



4.4.2. Rótulos

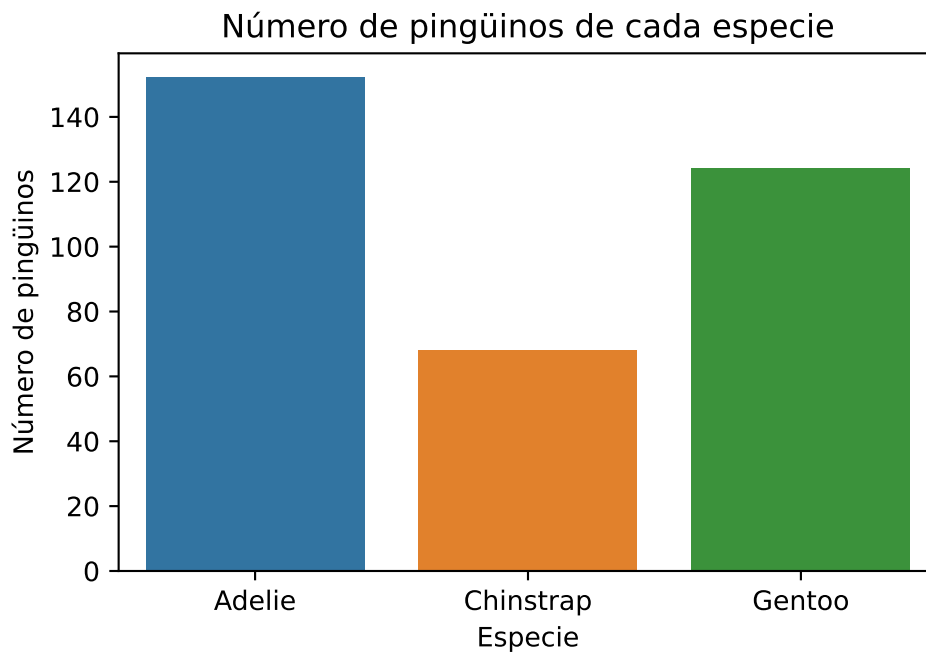
Con el siguiente código personalizamos los títulos de los ejes y damos un título global al gráfico

```
ax = sns.countplot(data=penguins, x="species")
ax.set(
    title="Número de pingüinos de cada especie",
    xlabel="Especie",
    ylabel="Número de pingüinos"
);
```

4.4.3. Orden de los niveles

Si te fijas en los gráficos que has hecho hasta ahora, apreciarás que las categorías aparecen en los gráficos en el mismo orden en el que aparecen en las filas de la hoja de datos.

Si quieres un orden personalizado para las categorías puedes usar el argumento `order`, como en el siguiente ejemplo:



```
sns.countplot(data=penguins, x="species", order =  
↳ ['Chinstrap', 'Adelie', 'Gentoo']);
```

Con el siguiente código ordenamos las categorías de mayor a menor frecuencia.

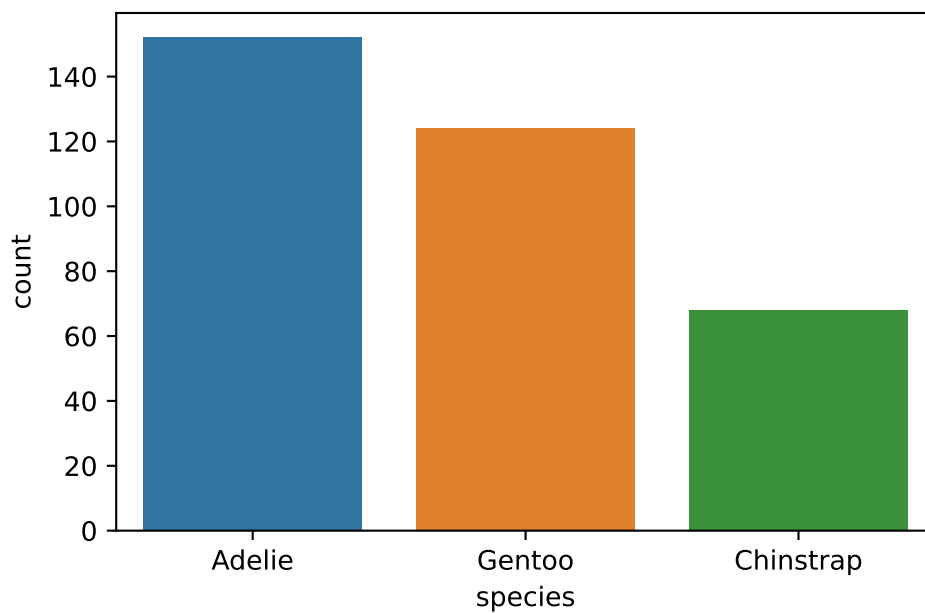
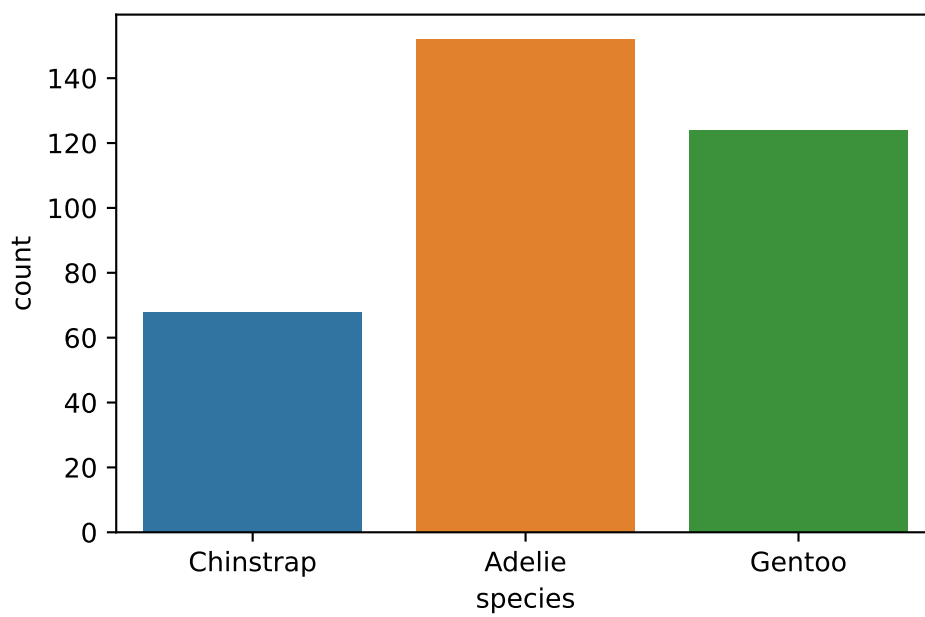
```
sns.countplot(data=penguins, x="species", order =  
↳ species.value_counts().index);
```

5. Una variable numérica

Igual que en el apartado anterior estudiamos una variable categórica, en este apartado se describen los métodos básicos para explorar y analizar una variable numérica.

Analizaremos en concreto la variable `body_mass_g` de nuestra hoja de datos, para conocer la distribución de los pesos de los pingüinos.

Empezamos seleccionando la variable en nuestra hoja de datos y almacenando el resultado en un nuevo objeto de nombre `mass`.




```
mass = penguins["body_mass_g"]
```

5.1. El método `describe()`

El método `describe()` que usamos en la sección anterior para una variable categórica, también puede aplicarse a variables numéricas.

Si lo aplicamos a `mass` obtenemos el siguiente resultado:

```
mass.describe()
```

	body_mass_g
count	342.000000
mean	4201.754386
std	801.954536
min	2700.000000
25%	3550.000000
50%	4050.000000
75%	4750.000000
max	6300.000000

La información que obtenemos de la salida anterior sobre la distribución del peso de los pingüinos es la siguiente:

Fragmento de la salida		Significado
count	342.000000	Hay 342 valores no nulos, así que no se conoce el peso de dos pingüinos.
mean	4201.754386	El peso medio de los pingüinos es 4 kilogramos y 201 gramos
std	801.954536	La desviación estandar del peso de los pinguinos es 801 gramos
min	2700.000000	El pingüino que menos pesa pesa 2 kilos y 700 gramos.

Fragmento de la salida		Significado
25%	3550.000000	El 25% de los pingüinos pesa menos de 3 kilos y 550 gramos (y el 75% restante más). Este valor se conoce como cuantil 0.25 o primer cuartil .
50%	4050.000000	El 50% de los pingüinos pesa menos de 4 kilos y 50 gramos (y el 50% restante más). Este valor se llama cuantil 0.5 , segundo cuartil o mediana .
75%	4750.000000	El 75% de los pingüinos pesa menos de 4 kilos y 750 gramos (y el 25% restante más). Este valor se llama cuantil 0.75 o tercer cuartil .
max	6300.000000	El pingüino que más pesa pesa 6 kilos y 300 gramos.

Problema 9

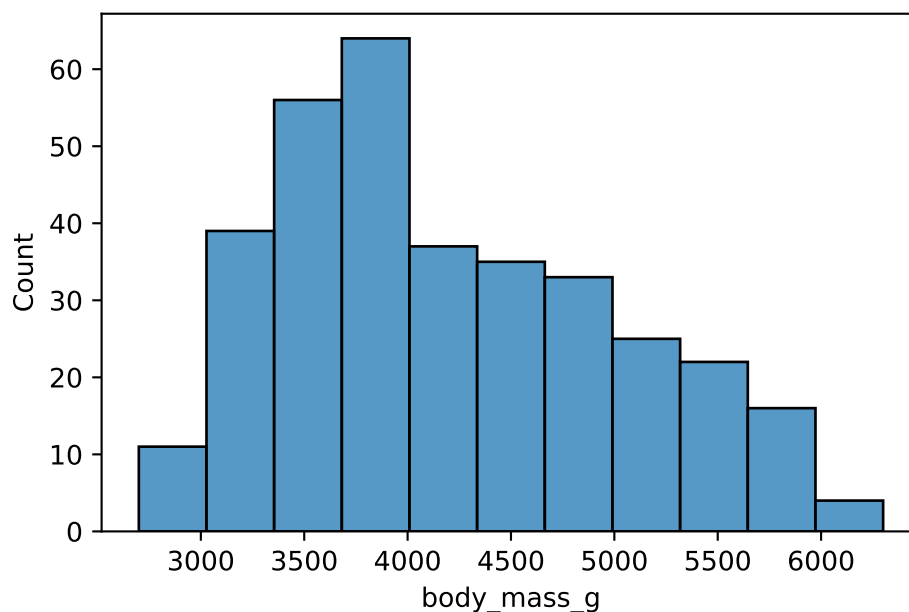
Utiliza el método `describe()` para obtener información sobre la distribución de la longitud de las alas de los pingüinos.

5.2. Histograma

Los **histogramas** son uno de los gráficos más comunes e informativos para describir la distribución de una variable continua. Para crear un histograma se representa en el eje x el rango de valores de la variable, se divide ese rango en intervalos de la misma longitud, y se dibuja para cada intervalo una barra de altura igual al número de valores que caen en ese intervalo.

El siguiente código crea un histograma para el peso de los pingüinos:

```
sns.histplot(data=penguins, x="body_mass_g");
```



- i** Para realizar un histograma usa la función `histplot()` del paquete `seaborn`, e indica:
- El nombre de la hoja de datos como valor del argumento `data`.
 - El nombre de la variable como valor del argumento `x`.

Problema 10

Realiza un histograma para la longitud de las alas de los pingüinos.

5.3. Diagrama de caja y bigotes

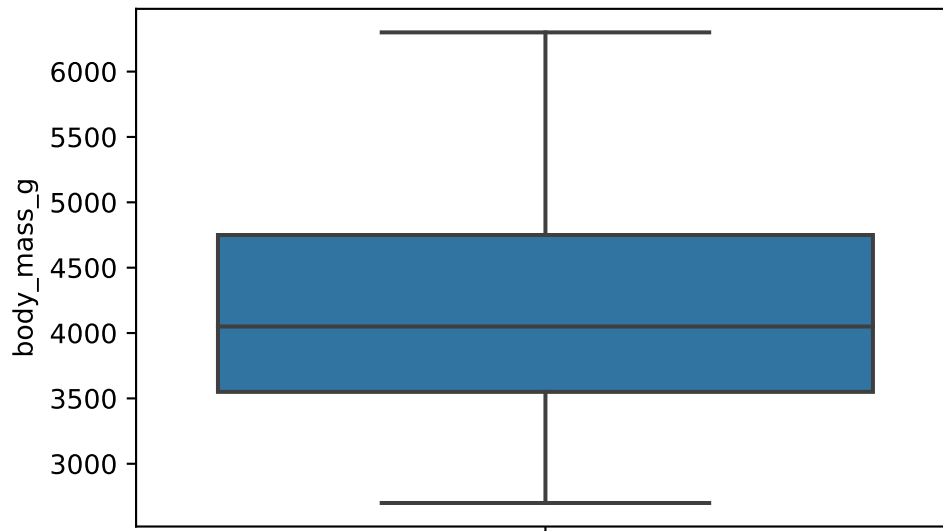
Otro tipo de gráficos para variables numéricas son los llamados **diagramas de caja** o **de caja y bigotes**.

La caja se construye usando los cuartiles de la variable.

Y los bigotes se extienden desde los extremos de la caja hasta los valores mínimo y máximo, exceptuando los valores que se clasifican como **outliers**.

Ejecuta el siguiente código para crear un diagrama de caja y bigotes para el peso de los pingüinos.

```
sns.boxplot(data=penguins, y="body_mass_g");
```



- i** Para realizar un diagrama de caja y bigotes usa la función `histplot()` del paquete `seaborn` e indica:
- El nombre de la hoja de datos como valor del argumento `data`.
 - El nombre de la variable como valor del argumento `y`.

Problema 11

Realiza un diagrama de caja y bigotes para la longitud de las alas de los pingüinos.

6. Continuará ...

Bibliografía

- [1] KB Gorman, TD Williams y WR Fraser. “Ecological Sexual Dimorphism and Environmental Variability within a Community of Antarctic Penguins (Genus *Pygoscelis*)”. en. En: *PLoS ONE* 9.3 (2014). DOI: [10.1371/journal.pone.0090081](https://doi.org/10.1371/journal.pone.0090081).
- [2] Wes McKinney. “Data Structures for Statistical Computing in Python”. En: *Proceedings of the 9th Python in Science Conference*. Ed. por Stéfan van der Walt y Jarrod Millman. 2010, págs. 56-61. DOI: [10.25080/Majora-92bf1922-00a](https://doi.org/10.25080/Majora-92bf1922-00a).
- [3] The pandas development team. *pandas-dev/pandas: Pandas*. Ver. latest. Feb. de 2020. DOI: [10.5281/zenodo.3509134](https://doi.org/10.5281/zenodo.3509134). URL: [10.5281/zenodo.3509134](https://zenodo.org/record/3509134).
- [4] Michael L. Waskom. “seaborn: statistical data visualization”. En: *Journal of Open Source Software* 6.60 (2021), pag. 3021. DOI: [10.21105/joss.03021](https://doi.org/10.21105/joss.03021). URL: <https://doi.org/10.21105/joss.03021>.

