universidad de león

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

Contenidos

Introducción

En esta práctica aprenderás las técnicas básicas para explorar 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 características morfológicas de tres especies de pingüinos del archipiélago Palmer en la Antártida.

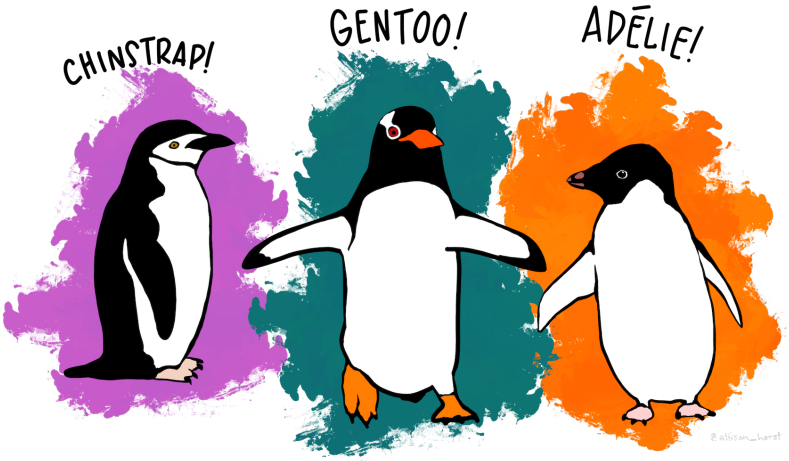


Figura 1: Ilustración de las tres especies de pingüinos del archipiélago Palmer (Artista @allison_horst)

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.

- ☐ Añadir un ejemplo de tabla y un ejemplo de gráfico. Por ejemplo, peso por especies.

1. Librerías

```
import pandas as pd
import numpy as np
```

1.1. `pandas`

1.1.1. sub

1.2. `numpy`

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`.

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. Visualización

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)
```

```
/home/eva/.local/lib/python3.10/site-packages/IPython/core/formatters.py:342: Futu

In future versions `DataFrame.to_latex` is expected to utilise the base implementa
```

	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	325
3	Adelie	Torgersen	NaN	NaN	NaN	N
4	Adelie	Torgersen	36.7	19.3	193.0	345

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

```
/home/eva/.local/lib/python3.10/site-packages/IPython/core/formatters.py:342: Fu

In future versions `DataFrame.to_latex` is expected to utilise the base implemen
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass
339	Gentoo	Biscoe	NaN	NaN	NaN	N
340	Gentoo	Biscoe	46.8	14.3	215.0	485
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

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, en la primera celda de la hoja de datos

```
/home/eva/.local/lib/python3.10/site-packages/IPython/core/formatters.py:342: Fu
```

```
In future versions `DataFrame.to_latex` is expected to utilise the base implementa
```

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

vemos 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 anterior con información sobre las siete variables de nuestra hoja de datos.

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
`species`	Especie de pingüinos (Chinstrap, Adélie o Gentoo)
`island`	Nombre de la isla del archipíelago Palmer (Dream, Torgersen o Biscoe)
`bill_length_mm`	Longitud del pico, en milímetros
`bill_depth_mm`	Anchura del pico, en milímetros
`flipper_length_mm`	Longitud de las alas
`body_mass_g`	Peso en gramos
`sex`	Sexo (MALE o FEMALE)

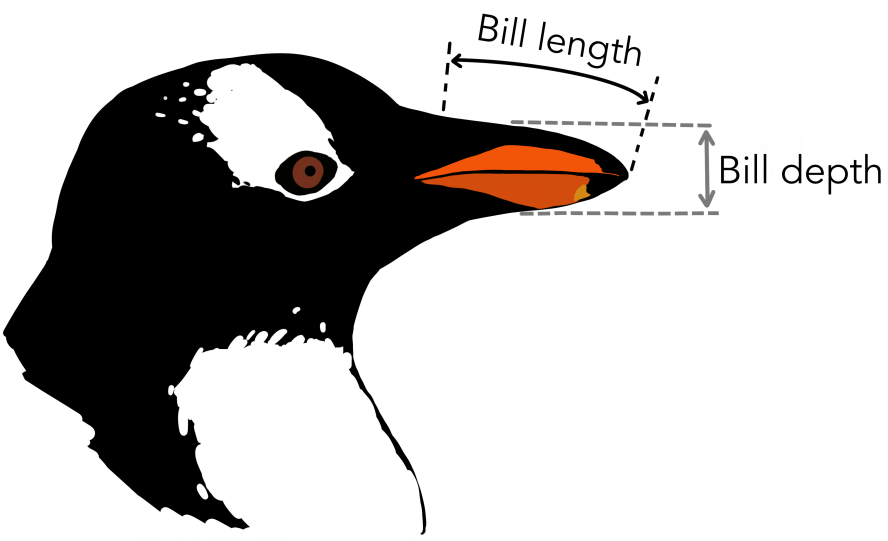


Figura 2: Ilustración de las variables `bill_length_mm` y `bill_depth_mm` (Artista @allison_horst)

Volviendo a mirar la primera fila de nuestra hoja de datos

```
/home/eva/.local/lib/python3.10/site-packages/IPython/core/formatters.py:342: Futu

In future versions `DataFrame.to_latex` is expected to utilise the base implementa
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_
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 3750 gramos, y es una hembra.

2.6. 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
```

- ☐ TODO

2.7. Tipos de variables

Fíjate ahora 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
```

- ☐ Hay `dtype` y `dtypes`
- ☐ Numéricas vs categóricas
- ☐ ¿Hablar aquí de conversión a categórica (`astype("category")`)?
- ☐ Mirar si `describe` se comporta diferente para numerica, `object` y `category`.

2.8. Índice de una hoja de datos

Igual que cada columna tiene un nombre, cada fila 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.

```
/home/eva/.local/lib/python3.10/site-packages/IPython/core/formatters.py:342: Fu

In future versions `DataFrame.to_latex` is expected to utilise the base implemen
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mas
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	325

```
/home/eva/.local/lib/python3.10/site-packages/IPython/core/formatters.py:342: Fu

In future versions `DataFrame.to_latex` is expected to utilise the base implemen
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_
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

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 utilizar 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.

Problema 1
Describe las características del tercer pingüino del estudio (índice 2).

3. Una variable categórica

3.1. El método `describe()`

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

```
species.describe()
```

```
C:\Users\Usuario\AppData\Local\Programs\Python\Python310\lib\site-packages\IPytho
```

```
In future versions `DataFrame.to_latex` is expected to utilise the base implement
```

	species
count	344
unique	3
top	Adelie
freq	152

- ☐ Decir las funciones individuales.
- ☐ ¿Hay diferencia entre convertirla a categórica (`astype("category")`) o no ?

3.2. Tabla de frecuencias

Tabla de frecuencias absolutas:

```
species_counts = penguins.value_counts(subset="species")
species_counts
```

```
C:\Users\Usuario\AppData\Local\Programs\Python\Python310\lib\site-packages\IPyth
```

```
In future versions `DataFrame.to_latex` is expected to utilise the base implemen
```

	0
species	
Adelie	152
Gentoo	124
Chinstrap	68

```
type(species_counts)
```

```
pandas.core.series.Series
```

```
species_counts.index
```

```
Index(['Adelie', 'Gentoo', 'Chinstrap'], dtype='object', name='species')
```

Problema 2
Determina el número total de hembras y de machos. Almacena el resultado en una variable de nombre `sex_counts`.

Tabla de frecuencias relativas (proporciones):

```
species_props = penguins.value_counts(
    subset="species",
    normalize=True
)
species_props
```

```
C:\Users\Usuario\AppData\Local\Programs\Python\Python310\lib\site-packages\IPytho

In future versions `DataFrame.to_latex` is expected to utilise the base implement
```

	0
species	
Adelie	0.441860
Gentoo	0.360465
Chinstrap	0.197674

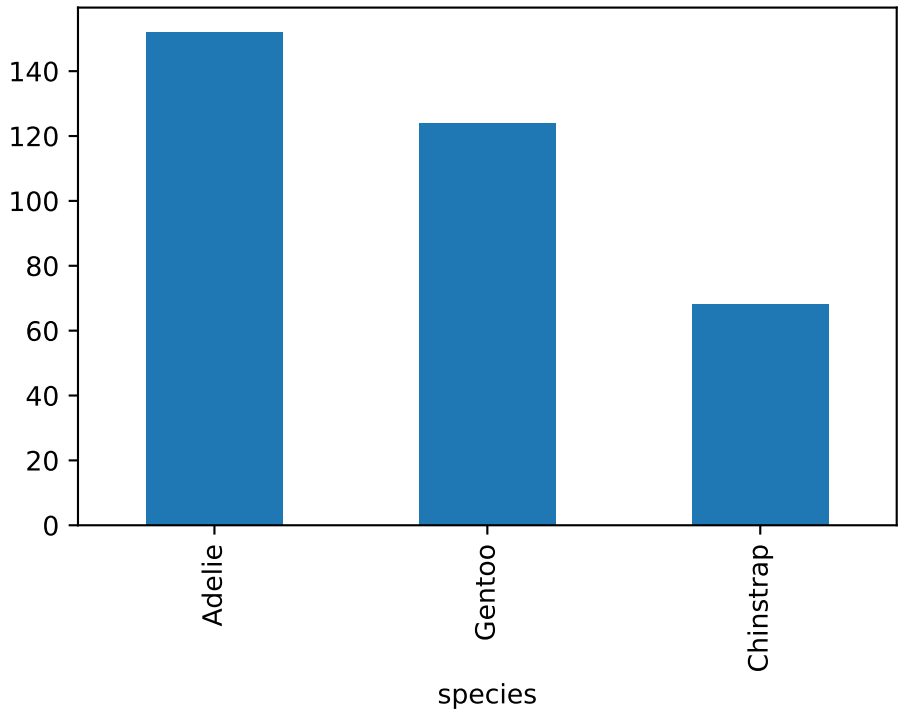
Tabla de porcentajes:

```
100*species_props
```

	0
species	
Adelie	44.186047
Gentoo	36.046512
Chinstrap	19.767442

3.3. Diagrama de barras con `pandas`

```
species_counts.plot.bar();
```



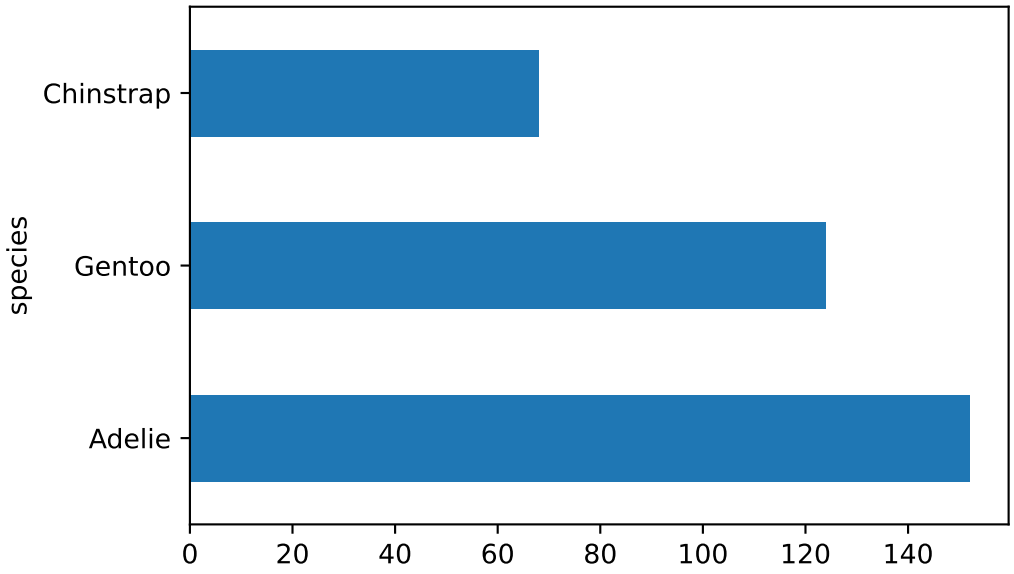

```
species_counts.plot.barh();
```

Problema 3

Usa la variable `sex_counts` creada en el Problema 2 para crear un diagrama de barras mostrando el número total de hembras y de machos.

Problema 4

Determina cuántos pingüinos hay en cada isla y dibuja un diagrama de barras con los resultados.



3.4. Diagrama de barras con `seaborn`

```
sns.barplot(x=species_counts.index, y=species_counts);
```

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

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

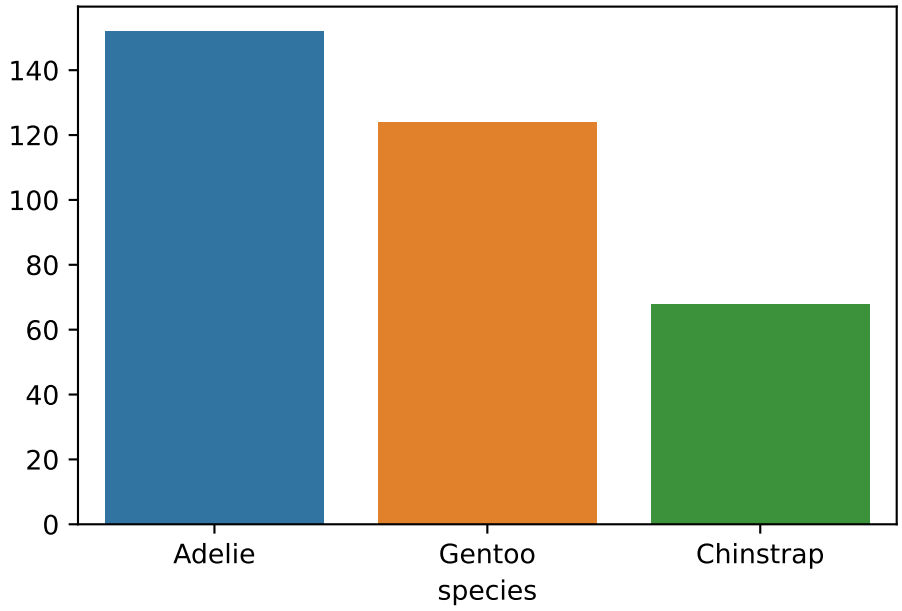
```
sns.countplot(data=penguins, x="species", order =
↪  species_counts.index);
```

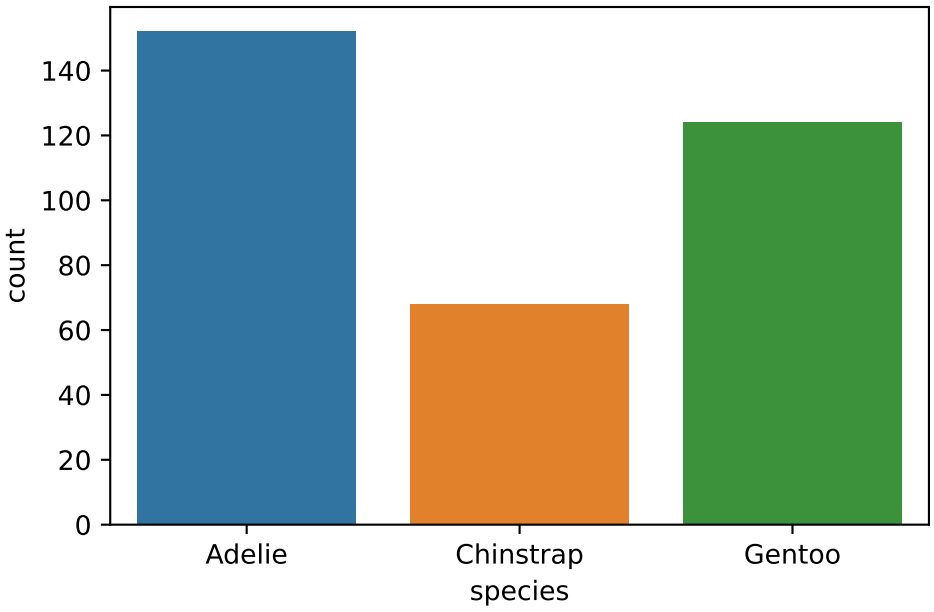
```
sns.countplot(data=penguins, y="species");
```

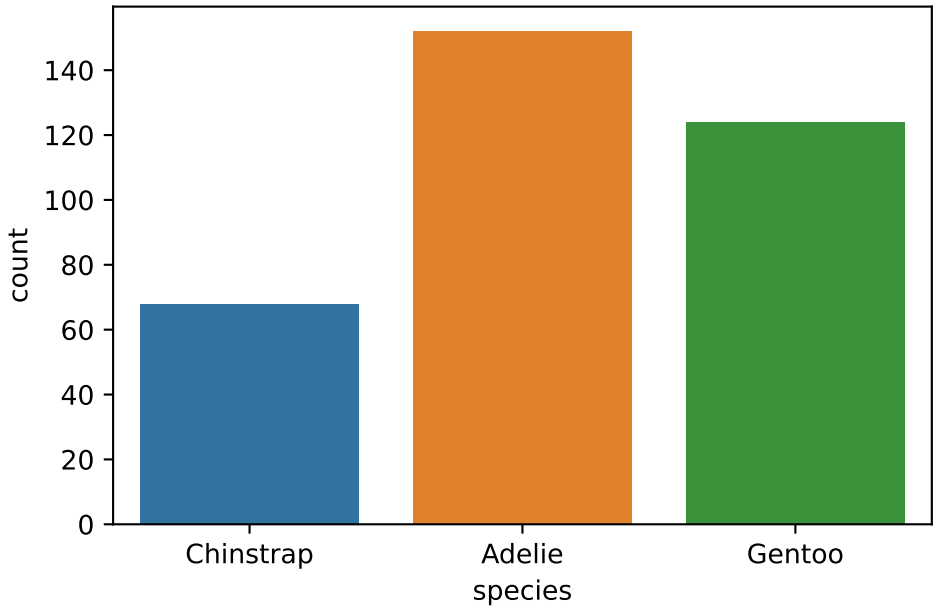
Problema 5

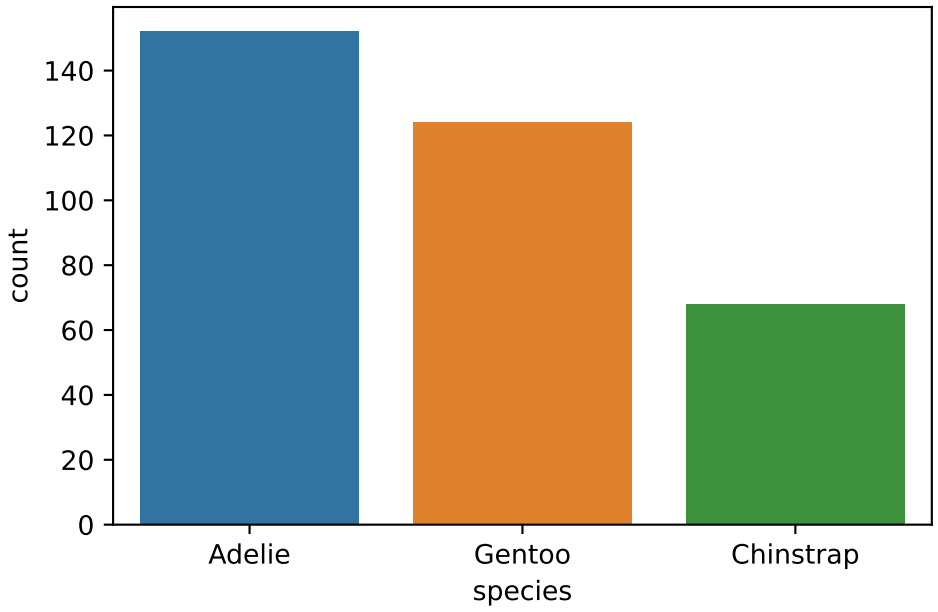
Utiliza la función `countplot()` de la librería `seaborn` para crear diagramas de barras para

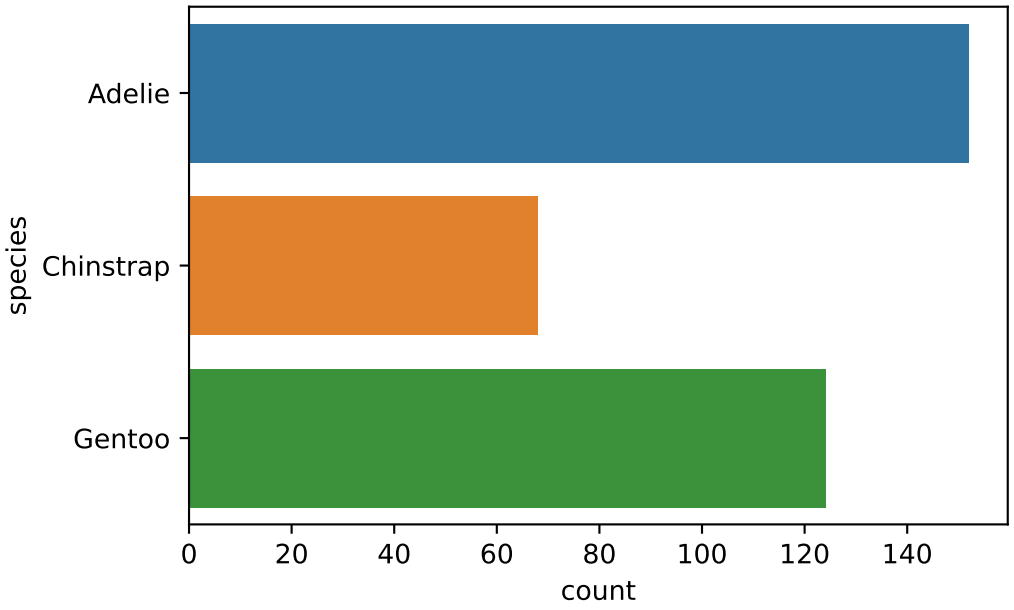
- el número de hembras y machos


140
120
100
80
60
40
20
0
Adelie
Gentoo
Chinstrap
species


count
140
120
100
80
60
40
20
0
Adelie
Chinstrap
Gentoo
species


140
120
100
80
60
40
20
0
count
Chinstrap
Adelie
Gentoo
species


140
120
100
80
60
40
20
0
count
Adelie
Gentoo
Chinstrap
species



- el número de pingüinos en cada isla

sin crear previamente tablas de recuentos.

3.4.1. Personalización de los gráficos

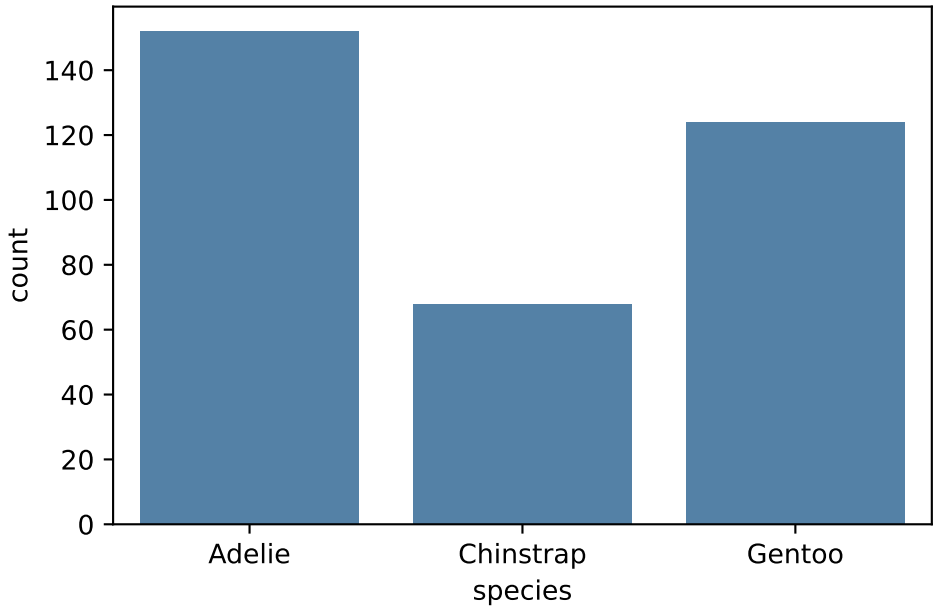
No es difícil personalizar los gráficos indicando títulos y colores [1]. Por ejemplo:

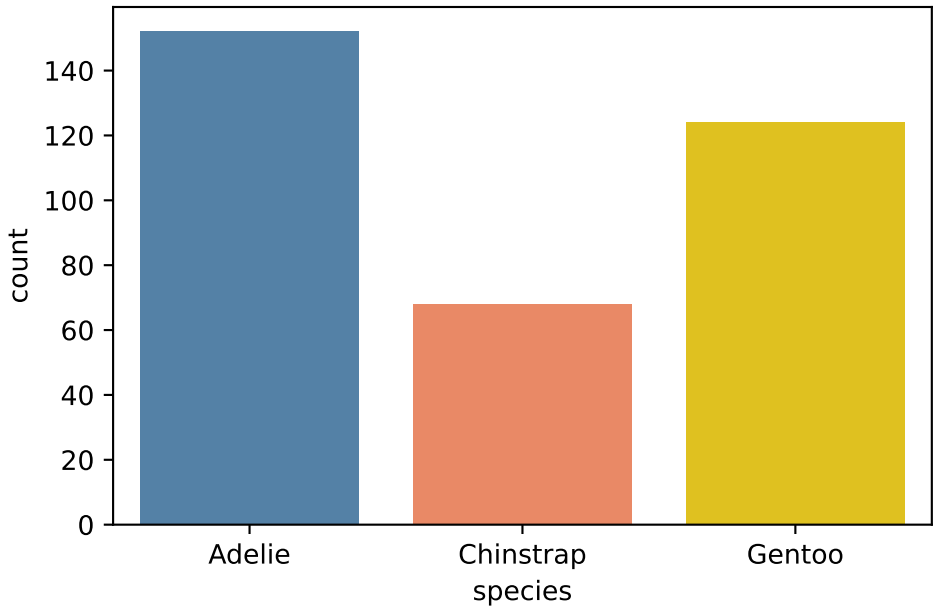
```
sns.countplot(data=penguins, x="species",
↪  color="steelblue");
```

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

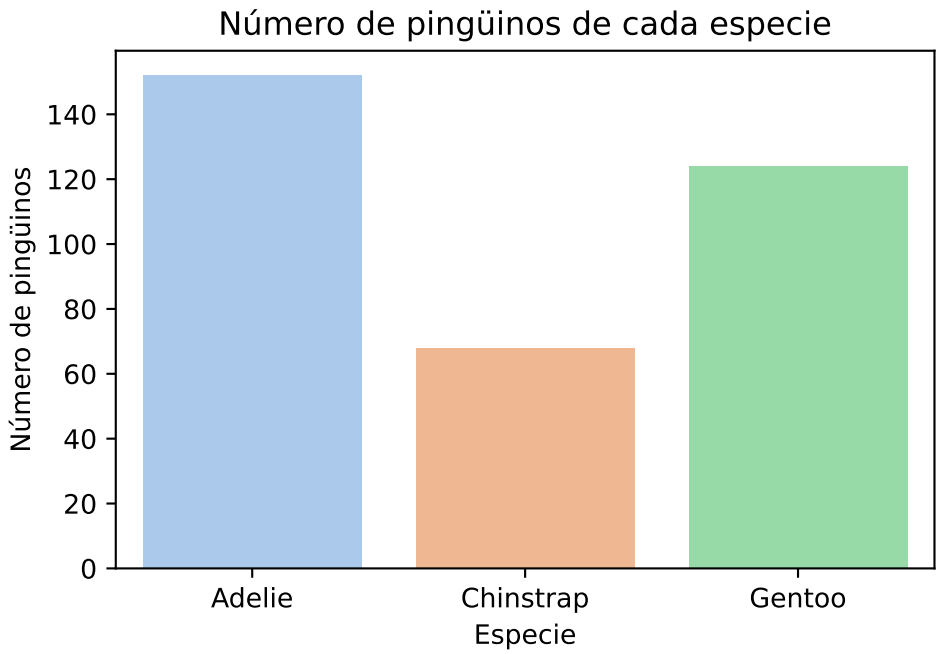
```
ax = sns.countplot(data=penguins, x="species",
↪  palette="pastel")
ax.set(
    title="Número de pingüinos de cada especie",
    xlabel="Especie",
```

[1] Puedes ver los colores disponibles aquí.


140
120
100
80
60
40
20
0
count
Adelie
Chinstrap
Gentoo
species


140
120
100
80
60
40
20
0
count
Adelie
Chinstrap
Gentoo
species

```
    ylabel="Número de pingüinos"
);
```



La personalización de los gráficos no carece de importancia, siendo especialmente relevante dar títulos descriptivos a los ejes. No obstante, 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 las librerías usadas.

4. Una variable numérica

4.1. El método `describe()`

4.2. Histograma

☐ Están `Dataframe.plot.hist()` y `Dataframe.hist()`

4.3. Diagrama de caja y bigotes

☐ Están `Dataframe.plot.box()` y `Dataframe.boxplot()`

5. Agrupación de hojas de datos

- ☐ Habiendo descubierto el argumento `subset` de `value_counts()` y el método `unstack()` (ver en mis apuntes de pandas `Gráficos > Diagramas de barras > Dos variables` y `Gráficos > Diagramas de barras > Tres variables`) puede que esta sección no haga falta hasta analizar variable numérica por categorías.

- Split
- Apply
- Combine

Un ejemplo sencillo como aplicar `sum()`.

O también tamaño muestral:

```
df.groupby("grade3").size()
```

6. Dos variables categóricas

- ☐ Ver lo que tengo explorado en mi manual de pandas en apartados `Gráficos > Diagramas de barras > Dos variables` y `Gráficos > Diagramas de barras > Tres variables`.

6.1. Tablas de frecuencias

6.2. Tablas de contingencia

```
pd.crosstab(df2["factor1"], df2["factor2"])
```

```
pd.crosstab(
    df2["factor1"],
    [df2["factor2"], df2["factor3"]]
)
```

7. Variable numérica por categorías

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: https://doi.org/10.1371/journal.pone.0090081.