

# Tasca S8.01. Visualitzacions en Python

## Descripció

Aquesta pràctica permetrà explorar com es poden utilitzar eines de visualització avançades en Python per a interpretar i presentar dades de manera efectiva, proporcionant una oportunitat valuosa per a millorar les habilitats analítiques i tècniques.

## Nivell 1

Realitza la connexió en Python amb el MySQL Workbench per a carregar tota la informació que tens en les taules.

Realitzaràs una visualització per a cada exercici. Comenta el que et crida l'atenció de graficar aquesta variable, justifica l'elecció del gràfic i interpreta els resultats en funció de les teves dades.

## SQL ALCHEMY

### Importar pandas i establir connexió

In [96]:

```
import pandas as pd
import sqlalchemy
from sqlalchemy import create_engine, inspect

engine = create_engine("mysql+pymysql://root@localhost/ecommerce")
cnx = engine.connect()
```

### Generar Dataframes

In [98]:

```
inspector = inspect(engine)
nombre_tablas = inspector.get_table_names()

for table in nombre_tablas:
    query = f"select * from {table}"
    globals()[f"df_{table}"] = pd.read_sql(query, cnx)
```

### Mostrar taules

In [100]:

```
#print(nombre_tablas)
nombre_tablas
#df_products[df_products["weight"] == 0.5]
df_companies
```

Out[100]:

	id	company_name	phone	email	country	website
0	b-2 222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.tellus@yahoo.net	Germany	https://instagram.com/site
1	b-2 226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia	https://whatsapp.com/group/9
2	b-2 230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	https://pinterest.com/sub/cars
3	b-2 234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.couk	Germany	https://cnn.com/user/110
4	b-2 238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	https://netflix.com/settings
.	...	...	...	...	...	...
9 5	b-2 602	Placerat LLP	05 43 67 24 41	tellus.suspendisse@hotmail.org	Netherlands	https://zoom.us/sub
9 6	b-2 606	Sed Est Corp.	04 58 02 37 91	tempor.bibendum@google.couk	Canada	https://instagram.com/sub/cars
9 7	b-2 610	Egestas Nunc Sed Limited	06 01 02 70 47	vitae@hotmail.edu	Italy	https://walmart.com/one
9 8	b-2 614	Rutrum Non Inc.	02 66 31 61 09	neque@protonmail.net	Germany	https://netflix.com/site
9 9	b-2 618	Non Institute	06 77 15 31 14	amet.metus@protonmail.couk	United Kingdom	https://guardian.co.uk/settings

100 rows × 6 columns

## Importar lliberies

In [102]:

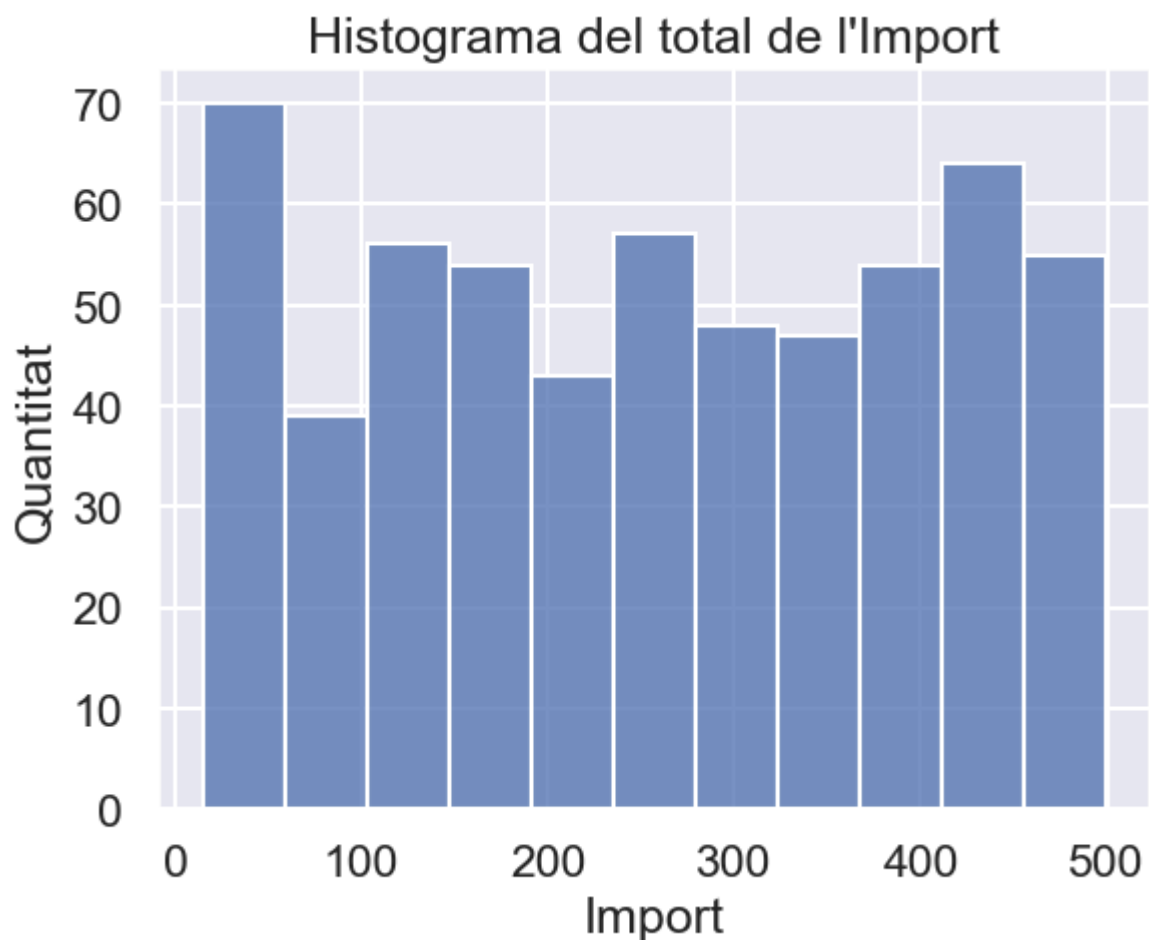
```
import matplotlib.pyplot as plt #genera gràfics a partir de continguts en llistes, vector amb py i amb  
l'extensió NumPy  
import seaborn as sns #llibreria especialitzada en visualització de dades per py desenvolupada amb  
matplotlib i ofereix interfície d'alt nivell  
#import squarify  
#import plotly.express as px  
import numpy as np  
import warnings
```

## Exercici 1

Una variable numèrica.

In [34]:

```
sns.set_theme(style="darkgrid", context="talk")  
sns.histplot(df_transactions["amount"])  
plt.title("Histograma del total de l'Import")  
plt.ylabel("Quantitat de registres")  
plt.xlabel("Import")  
plt.show()
```



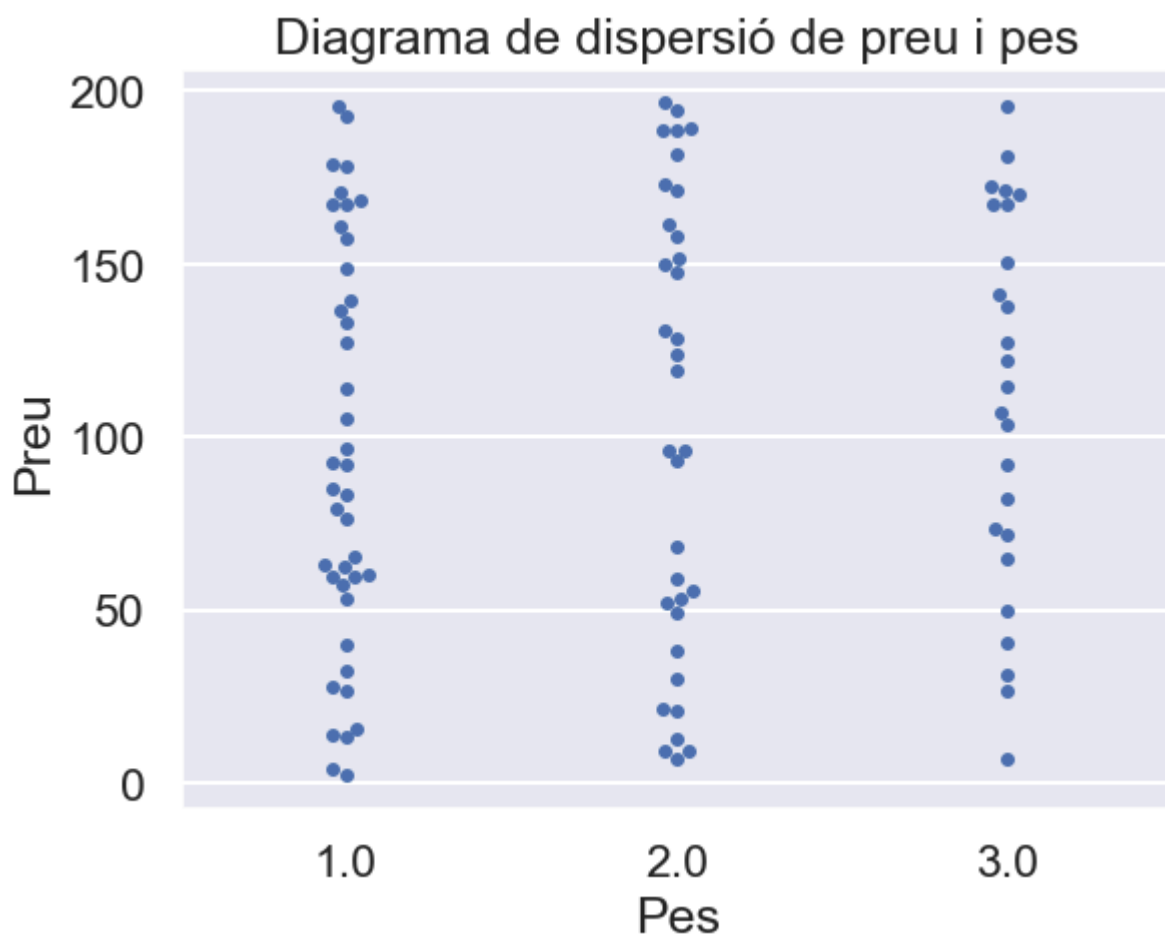
Estem veient que el major nombre d'amounts són els que van de 0 a 50€ aproximadament. I les que tenen menys, estan entre 50 i 100€.

## Exercici 2

Dues variables numèriques.

In [37]:

```
sns.set_theme(style="darkgrid", context="talk")
sns.swarmplot(df_products, y="precio", x="weight")
plt.title("Diagrama de dispersió de preu i pes")
plt.ylabel("Preu")
plt.xlabel("Pes")
plt.show()
```



En la visualització, podem veure que no hi ha diferència relativa entre pes i preu. És a dir, el fet que un producte sigui major pes, no vol dir que sigui més car.

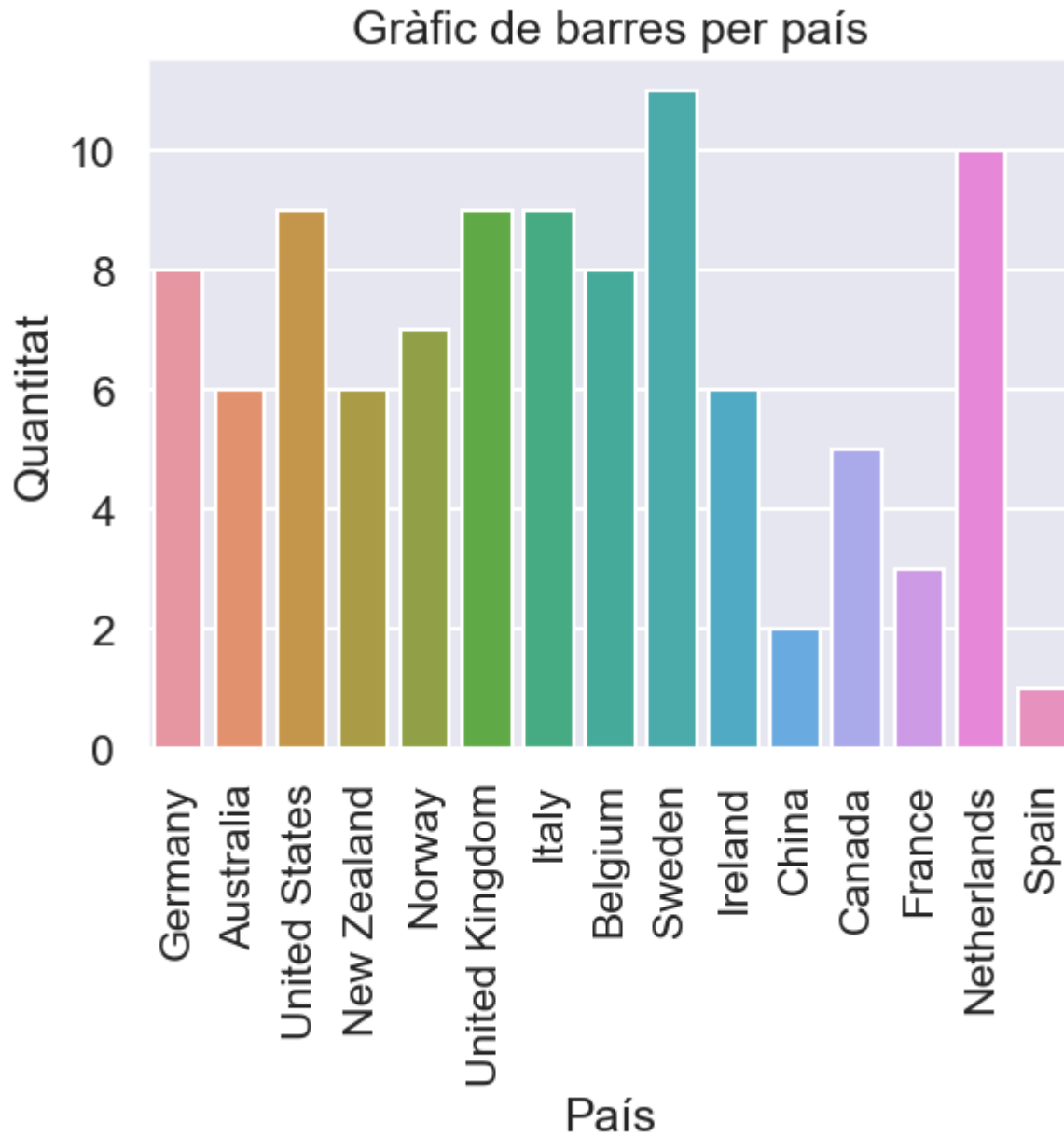
## Exercici 3

Una variable categòrica.

In [39]:

```
sns.set_theme(style="darkgrid", context="talk")
sns.countplot(df_companies, x="country")
plt.title("Gràfic de barres per país")
```

```
plt.ylabel("Quantitat")
plt.xlabel("País")
plt.xticks(rotation=90)
plt.show()
```



La variable categòrica escollida és country per companyies; i a la visualització podem veure que els països que tenen major nombre de companyies són Suècia i Països Baixos; i les que menys Espanya i Xina.

#### Exercici 4

Una variable categòrica i una numèrica.

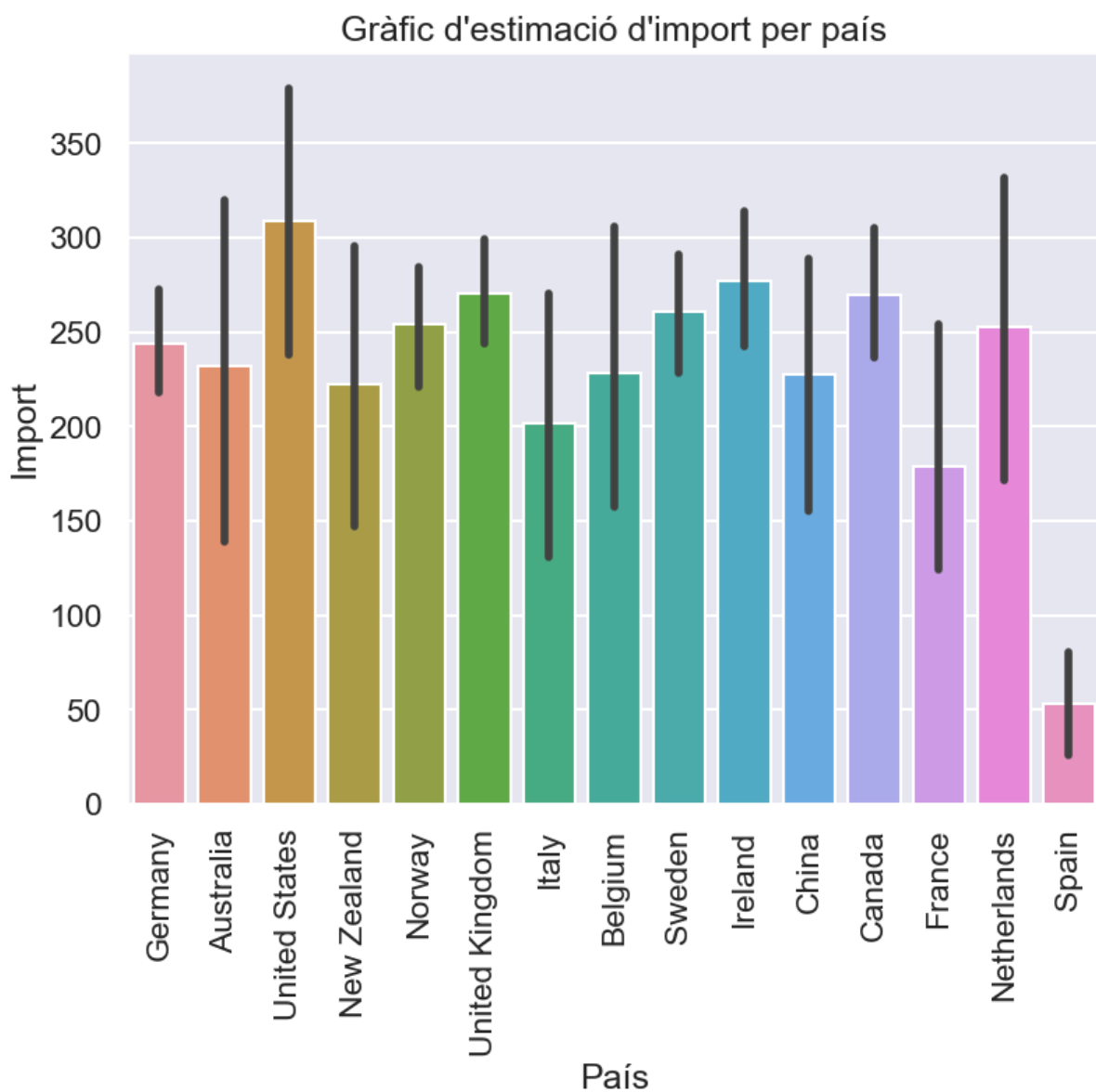
In [112]:

```
sns.set_theme(style="darkgrid", context="talk")
#fem un merge
```

```

df_companies_transactions = df_companies.merge(df_transactions, how = "inner", left_on="id",
right_on="company_id")
plt.figure(figsize=(9,7))
sns.barplot(df_companies_transactions, x="country", y="amount")
plt.title("Gràfic d'estimació d'import per país")
plt.ylabel("Import")
plt.xlabel("País")
plt.xticks(rotation=90)
plt.show()
#df_companies_transactions

```



Hem triat una visualització de barplot per fer una variable categòrica de país i una variable numèrica amount. La conclusió que podem treure és que el promig més alt seria el que té Estats Units, i Espanya està en darrer lloc.

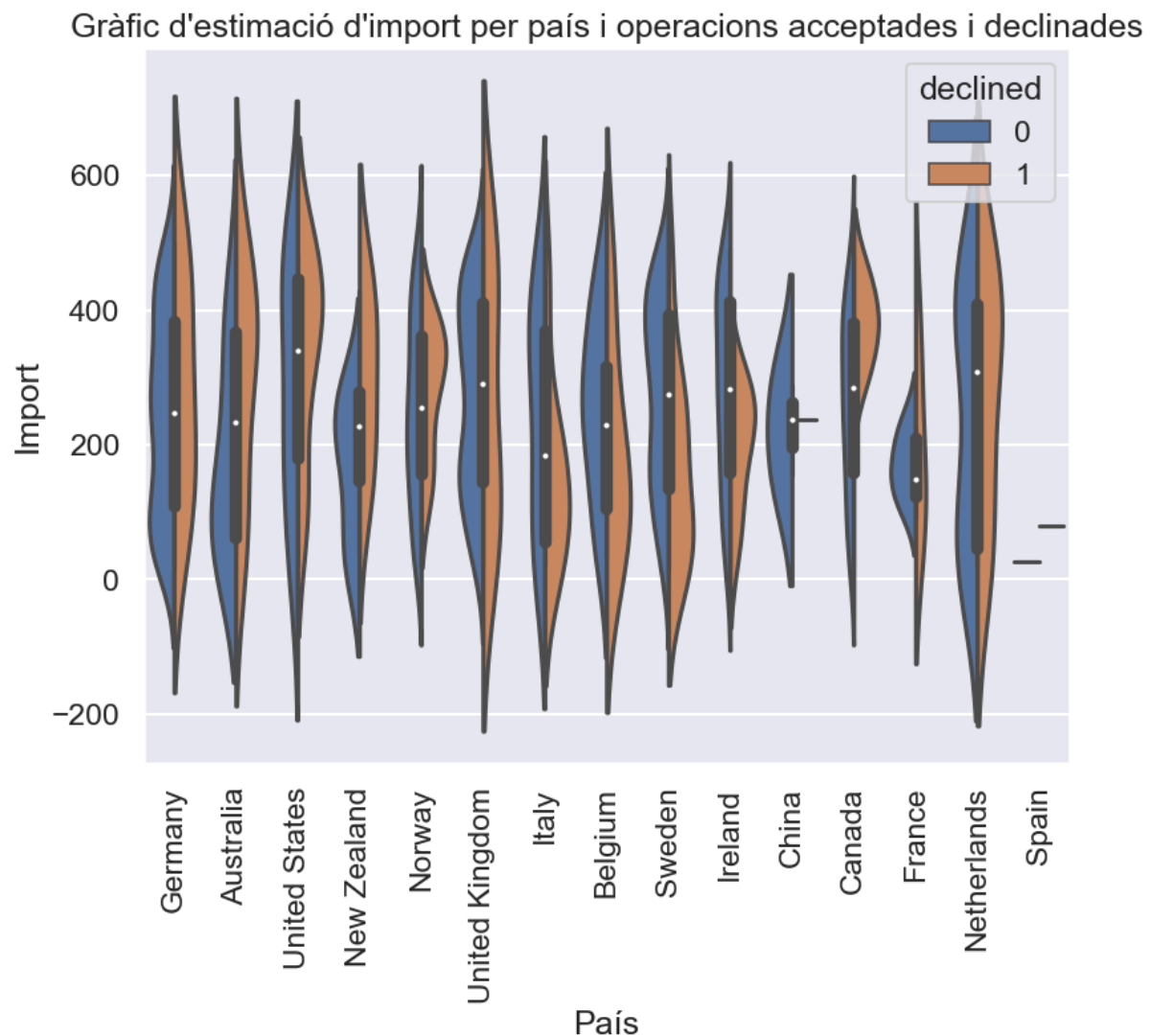
A més, també podem veure les línies amb l'interval de confiança de cada país, que ens mostra el valor més probable d'entrada del proper registre per cada país.

## Exercici 5 i 6

Dues variables categòriques Tres variables.

In [110]:

```
sns.set_theme(style="darkgrid", context="talk")
plt.figure(figsize=(9,7))
sns.violinplot(df_companies_transactions, x="country", y="amount", hue="declined", split = True)
plt.title("Gràfic d'estimació d'import per país i operacions acceptades i declinades")
plt.ylabel("Import")
plt.xlabel("País")
plt.xticks(rotation=90)
plt.show()
```



En aquest cas, hem optat per ajuntar els exercicis 5 i 6, i per tant, tenim 2 variables categòriques (declined i country) i una tercera variable que és l'amount. Hem utilitzat violinplot, que fa el promig d'amount agrupant per país i declined. En aquest sentit, podem veure d'una manera molt clara que en la majoria de països, el valor promig de les transaccions declinades és major que el promig de les transaccions acceptades.

## Exercici 7

Graficar un Pairplot.

In [72]:

```
sns.set_theme(style="darkgrid", context="talk")
# al segon merge, el join left és la taula intermitja, és a dir en aquest cas, product_transaction
df_transactions_products_tr = df_transactions.merge(df_product_transaction, how = "inner",
left_on="id", right_on="order_id").merge(df_products, how = "inner", left_on = "product_id", right_on =
"id")
df_transactions_products_tr
```

*#El pairplot només treballa amb variables numèriques, ja que compara variables numèriques.*

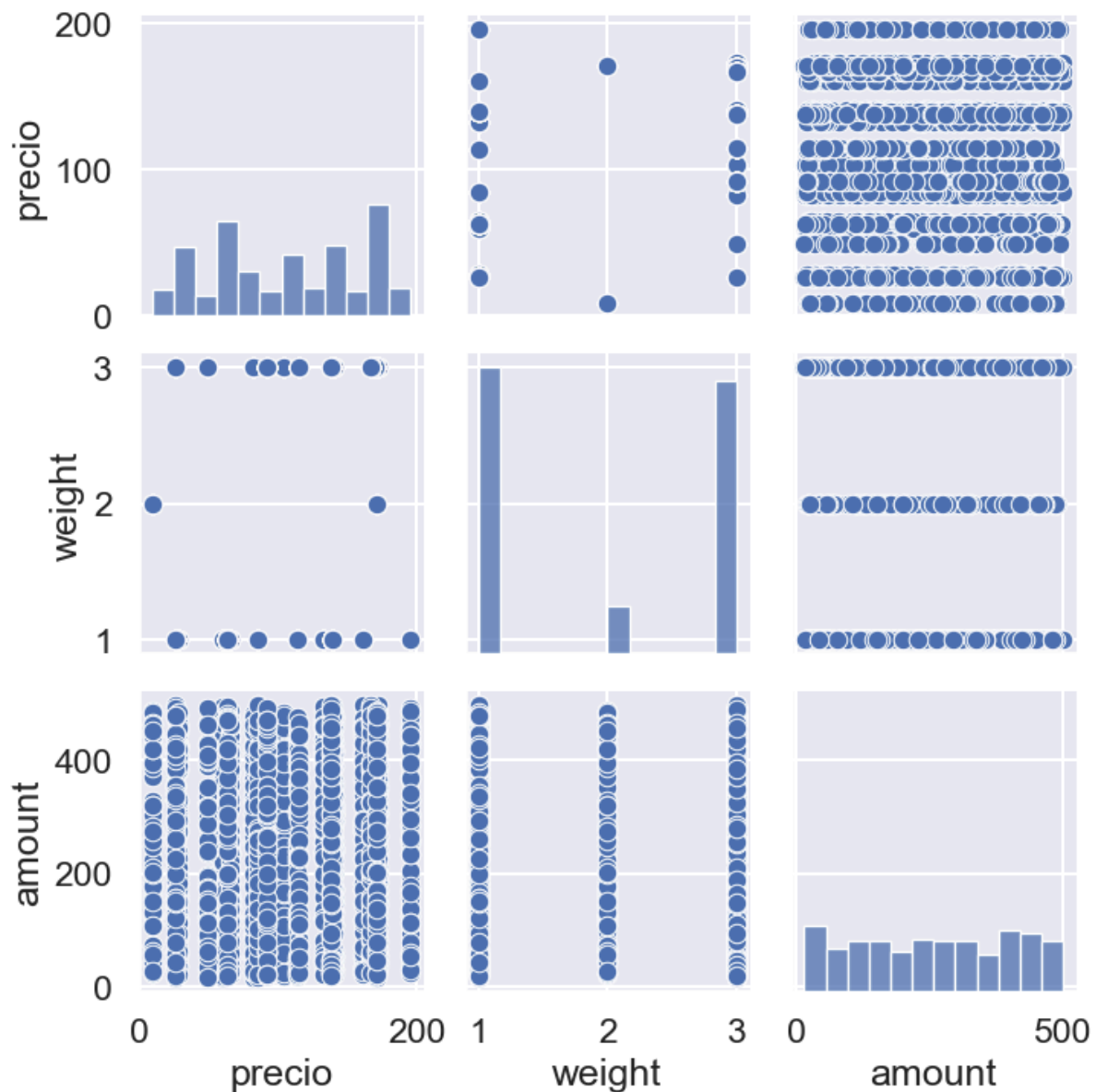
```
sns.pairplot(df_transactions_products_tr, vars= ["precio", "weight", "amount"])
```

```
plt.show()
```

C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight

```
self._figure.tight_layout(*args, **kwargs)
```





Tal i com podem veure amb la utilització del pairplot, aquest funciona amb varialbes numèriques, i per això hem triat com a variables, amount de la taula transactions, i precio i weight de la taula products.

El pairplot ens mostra gràfics d'histograma i gràfics de dispersió; en aquests, podem observar que a la distribució entre amount i precio hi ha una relació entre preu i total de l'amount, i que aquest per tant, no té perquè ser major si el preu del producte és més elevat o més baix.

En canvi, en la realació weith i precio, veiem que tenim molts productes a la venda que pesen 1 i 3kg, mentre que hi ha procs productes que pesin 2kg. A més, també observem que la distribució ens els productes de 2 kgs el preu o és molt car o molt baix, i que per tant, segurament faltaria que tinguéssim productes de 2 kgs amb un preu intermig.

## Tancar connexió

```
cnx.close()
```

In [118]:

### **Comprovar que la connexió estigui tancada (resposta TRUE)**

In [120]:

```
cnx.closed
```

Out[120]:

```
True
```

### **Desactivar engine**

In [122]:

```
engine.dispose()
```