

# Tasca S8.01. Visualitzacions en Python

## 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.

Pas 1. Importar les llibreries necessàries per fer les visualitzacions i funcions específiques que sen's demana a l'sprint.

In [1]:

```
# Importem les llibreries i mòduls
import matplotlib.pyplot as plt    # Per crear gràfics
import numpy as np                # Per realitzar operacions estadístiques
import pandas as pd              # Per treballar amb dades estructurades
import seaborn as sns            # Per la visualització de dades estadístiques
import mysql.connector           # Per establir la connexió i realitzar operacions amb la base de dades MySQL
```

Pas 2. Establir la connexió entre la base de dades de MySQL Workbench i Python

In [2]:

```
def connection():
    # Connexió amb la base de dades MySQL mitjançant les credencials
    connection = mysql.connector.connect(
        host="localhost",
        user="maria",
        password="marmota",
        database="test"
    )
    return connection
```

Pas 3. Realitzar una consulta que obtingui informació de l'estructura de cada taula per tal de verificar que les dades són correctes.

El següent codi es connecta a la base de dades 'test' i mitjançant la creació d'un objecte «cursor» interactuem amb la BBDD, executant consultes i recuperant les dades. D'aquesta manera, obtenim la llista de taules utilitzant la consulta SQL SHOW TABLES, i a continuació iterem sobre cada taula per obtenir la seva estructura mitjançant DESCRIBE. Finalment, imprimim la informació de totes les columnes de cada taula. D'aquesta manera obtenim una descripció detallada de l'estructura de les taules, els noms de les columnes i el tipus de dades.

In [3]:

```
# Funció per obtenir les taules de la base de dades
def get_tables(cursor):
    cursor.execute("SHOW TABLES")
    return cursor.fetchall()
```

```

# Funció per obtenir la informació d'una taula
def get_table_info(cursor, table_name):
    cursor.execute(f"DESCRIBE {table_name}")
    return cursor.fetchall()

# Establir la connexió
conn = connection()

# Crear un objecte cursor
cursor = conn.cursor()

# Obtenir la llista de taules de la base de dades
tables = get_tables(cursor)

# Iterar sobre cada taula i imprimir la seva informació
for table in tables:
    table_name = table[0]
    print(f"Taula: {table_name}")
    table_info = get_table_info(cursor, table_name)
    for column_info in table_info:
        print(column_info)

# Tancar el cursor i la connexió
cursor.close()
conn.close()

```

Taula: companies

```

('company_id', 'varchar(15)', 'NO', 'PRI', None, "")
('company_name', 'varchar(255)', 'YES', "", None, "")
('phone', 'varchar(15)', 'YES', "", None, "")
('email', 'varchar(100)', 'YES', "", None, "")
('country', 'varchar(100)', 'YES', "", None, "")
('website', 'varchar(255)', 'YES', "", None, "")

```

Taula: credit\_cards

```

('id', 'varchar(15)', 'NO', 'PRI', None, "")
('user_id', 'int(11)', 'YES', "", None, "")
('iban', 'varchar(100)', 'YES', "", None, "")
('pan', 'varchar(100)', 'YES', "", None, "")
('pin', 'varchar(100)', 'YES', "", None, "")
('cvv', 'varchar(15)', 'YES', "", None, "")
('track1', 'varchar(255)', 'YES', "", None, "")
('track2', 'varchar(255)', 'YES', "", None, "")
('expiring_date', 'varchar(25)', 'YES', "", None, "")

```

Taula: products

```

('id', 'int(11)', 'NO', 'PRI', None, 'auto_increment')
('product_name', 'varchar(100)', 'YES', "", None, "")
('price', 'decimal(10,2)', 'YES', "", None, "")
('colour', 'varchar(50)', 'YES', "", None, "")
('weight', 'decimal(10,2)', 'YES', "", None, "")
('warehouse_id', 'varchar(10)', 'YES', "", None, "")

```

Taula: transactions

```

('id', 'varchar(255)', 'NO', 'PRI', None, "")
('card_id', 'varchar(15)', 'YES', 'MUL', None, "")
('business_id', 'varchar(15)', 'YES', 'MUL', None, "")
('timestamp', 'timestamp', 'YES', "", None, "")
('amount', 'decimal(10,2)', 'YES', "", None, "")
('declined', 'tinyint(1)', 'YES', "", None, "")
('product_ids', 'varchar(255)', 'YES', "", None, "")
('user_id', 'int(11)', 'YES', 'MUL', None, "")
('lat', 'float', 'YES', "", None, "")
('longitude', 'float', 'YES', "", None, "")
Taula: transactions_products
('transaction_id', 'varchar(255)', 'YES', 'MUL', None, "")
('product_id', 'int(11)', 'YES', 'MUL', None, "")
Taula: users
('id', 'int(11)', 'NO', 'PRI', None, "")
('name', 'varchar(255)', 'YES', "", None, "")
('surname', 'varchar(255)', 'YES', "", None, "")
('phone', 'varchar(20)', 'YES', "", None, "")
('email', 'varchar(255)', 'YES', "", None, "")
('birth_date', 'varchar(20)', 'YES', "", None, "")
('country', 'varchar(255)', 'YES', "", None, "")
('city', 'varchar(255)', 'YES', "", None, "")
('postal_code', 'varchar(50)', 'YES', "", None, "")
('address', 'varchar(255)', 'YES', "", None, "")

```

## Exercici 1

### Una variable numèrica

Seleccionem el pes dels productes com a variable numèrica i l'histograma com objecte visual. Triem l'histograma perquè ens permet representar com es distribueix el pes comptant el nombre d'observacions que es troben dins de cada bloc. Aquest nombre d'observacions per contenidors o 'bins' es representa per l'alçada de la barra.

A més a més, aprofitarem per visualitzar estadístics descriptius com la mitjana, la moda, la mediana, la desviació estàndard i l'estimació de densitat de Kernel (KDE).

Vegem-ho:

In [399]:

```

# Establir la connexió
conn = connection()

# Cursor per executar consultes
cursor = conn.cursor()

# Definir la consulta per obtenir la llista de taules de la base de dades
cursor.execute("SELECT weight FROM products")
rows = cursor.fetchall()

# Convertir les dades en DataFrame de Pandas per facilitar l'anàlisi
dataframe = pd.DataFrame(rows, columns=['weight'])

```

```

# Tancar el cursor
cursor.close()

# Tancar connexió
conn.close()

# Convertir la columna 'weight' a float
dataframe['weight'] = dataframe['weight'].astype(float)

# Obtenir estadístiques descriptives
stats = dataframe['weight'].describe().round(2)
mean_weight = stats['mean']
std_weight = stats['std']

# Calcular la moda, la mediana i el rang
mode_weight = dataframe['weight'].mode()[0] # La moda pot tenir més d'un valor, seleccionem el primer
median_weight = dataframe['weight'].median()
range_weight = dataframe['weight'].max() - dataframe['weight'].min()

# Agregar la moda, mediana i rang a les estadístiques descriptives
stats['mode'] = round(mode_weight, 2)
stats['median'] = round(median_weight, 2)
stats['range'] = round(range_weight, 2)

# Seleccionar un estil de Matplotlib
plt.style.use('bmh')

# Comprova si la columna 'weight' existeix
if 'weight' in dataframe.columns:

    # Crear la figura
    plt.figure(figsize=(12, 5))

    # Crear l'histograma
    sns.histplot(data=dataframe, x="weight", bins=30, kde=True, color='royalblue', edgecolor='black')

    # Afegir línies verticals per a la mitjana i la desviació estàndard
    plt.axvline(mean_weight, color='red', linestyle='dashed', linewidth=2, label=f'Mitjana: {mean_weight:.2f}')
    plt.axvline(mean_weight + std_weight, color='orange', linestyle='dashed', linewidth=2, label=f'+1 STD: {mean_weight + std_weight:.2f}')
    plt.axvline(mean_weight - std_weight, color='orange', linestyle='dashed', linewidth=2, label=f'-1 STD: {mean_weight - std_weight:.2f}')

    # Configurar les etiquetes i el títol dels eixos
    plt.title('Distribució del pes dels productes', fontsize=16)
    plt.xlabel('Pes (Kilograms)')
    plt.ylabel('Quantitat')

    # Modificar les etiquetes de l'eix Y per intervals específics
    plt.yticks([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

```

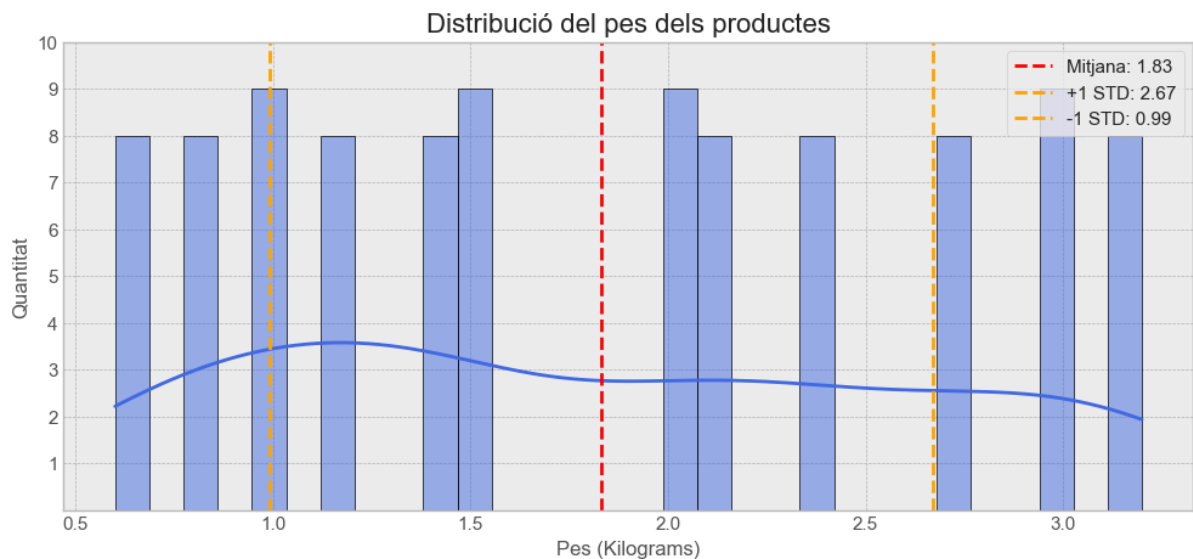
```
# Col·locar la llegenda a la part superior dreta  
plt.legend(loc='upper right')
```

```
# Mostrar la visualització  
plt.show()
```

```
print(stats)
```

**else:**

```
print("La columna 'weight' no existeix a la taula 'products'.")
```



```
count    100.00  
mean      1.83  
std       0.84  
min       0.60  
25%      1.15  
50%      1.75  
75%      2.48  
max       3.20  
mode      1.00  
median    1.75  
range     2.60  
Name: weight, dtype: float64
```

Obtenim un histograma amb la distribució de la variable analitzada i un 'print' amb els principals estadístics descriptius. Les estadístiques ens ajuden a entendre millor la distribució del pes dels productes, indicant on es troben la majoria en relació amb la mitjana i com es dispersen aquests pesos.

Observem que el pes es mou entre els 0.60 kg i els 3.20 kg (hem suposat que la mesura de pes és en Kilograms). El rang, que és la diferència entre el valor màxim i mínim, és de **2.60 kg**. Tenim 100 productes i la mitjana de pes és de **1.83 kg** (marcat en el gràfic amb una línia vermella) en comparació a la mediana que és de **1.75 kg**. Recordem que un dels avantatges de la mediana és que es veu menys afectada pels valors atípics i les dades esbiaixades. Les línies de color taronja ens

marquen la Desviació Estàndard (STD), que mesura la variabilitat en els pesos. La STD és de 0.84 kg, la +1 STD representa la mitjana més la STD ( $1.83+0.84$ ) i la -1 STD se li resta la STD ( $1.83-0.84$ ). Així doncs, la majoria dels productes tenen un pes que està entre 0.99 kg i 2.67 kg. En el gràfic, les línies verticals a 1.83 kg (mitjana), 2.67 kg (+1 STD) i 0.99 kg (-1 STD) ajuden a visualitzar on es troba la major part de la distribució dels pesos.

Les barres dels gràfics ens ensenyen com el pes es distribueix de manera força uniforme, **tenim entre 8 i 9 pproductes per cada rang de pes**. La línia de color blau és l'estimació de Densitat de Kernel (KDE) i ens mostra precisament aquesta tendència. Representa la distribució del pes dels productes mitjançant una corba suau. Així doncs, veiem que els pesos es distribueixen de manera força uniforme, tenint una major concentració on la corba és més alta i menys concentració on és més baixa.

He trobat interessant analitzar aquesta variable, ja que el pes és molt rellevant a l'hora de tenir en compte els costos de producció i la logística. Conèixer com influeixen els diferents pesos en el cost de producció i tenir-los en compte en els costos d'enviament i emmagatzematge, permet a l'empresa prendre decisions molt més informades a l'hora de triar estratègies de preus i rendibilitat de cada producte. També és important a l'hora d'optimitzar inventaris per la capacitat d'emmagatzematge i a l'hora de planificar la reposició d'estoc, i per conèixer la rendibilitat dels productes en relació amb el seu pes. Ens pot ser de gran ajuda, a l'hora de refinar estratègies d'inventari. Conèixer la distribució dels pesos per optimitzar la gestió d'inventaris, assegurant una disponibilitat de manera equilibrada dels productes de cada rang de pes.

Vist això, si volem analitzar més aquesta variable, podríem considerar combinar l'anàlisi del pes amb altres dades com el seu preu i la quantitat de transaccions per tenir una imatge més completa. Com veurem més endavant.

## Exercici 2

### Dues variables numèriques

En aquest cas, mantenim la variable pes i li afegim una segona variable numèrica, el preu. Ho visualitzarem amb un scatterplot o diagrama de dispersió per tal de mostra la relació entre aquestes dues variables numèriques. També mostrarem el càlcul del coeficient de correlació, que mesura la relació lineal.

Ho farem mitjançant el següent codi:

In [400]:

```
#Establir la connexió
conn = connection()

# Cursor per executar la consulta
cursor = conn.cursor()

# Definir la consulta SQL que obté les dades preu i pes del productes
cursor.execute("SELECT price, weight FROM products")

# Obtenir els resultats de la consulta
rows = cursor.fetchall()
dataframe = pd.DataFrame(rows, columns=['price', 'weight'])

# Calcular la correlació
correlation = dataframe['weight'].corr(dataframe['price'])
```

```

# Crear una nova columna que classifica el pes per intervals
dataframe['weight_range'] = pd.cut(dataframe['weight'], bins=[0.6, 1.0, 1.5, 2.0, 2.5, 3.0, 3.2])

# Només es mostren les combinacions de categories que tenen dades
grouped = dataframe.groupby('weight_range', observed=True)['price'].mean()

# Tancar el cursor
cursor.close()

# Tancar la connexió
conn.close()

# Verificar si les columnes 'price' i 'weight' existeixen en el DataFrame
if 'price' in dataframe.columns and 'weight' in dataframe.columns:

    # Convertir les columnes a float
    dataframe['price'] = dataframe['price'].astype(float)
    dataframe['weight'] = dataframe['weight'].astype(float)

    # Agregar una columna pels rangs de pes
    bins = [0.60, 1.00, 1.50, 2.00, 2.50, 3.00, 3.20]
    labels = ['0.60-1.00', '1.01-1.50', '1.51-2.00', '2.01-2.50', '2.51-3.00', '3.01-3.20']
    dataframe['weight_range'] = pd.cut(dataframe['weight'], bins=bins, labels=labels, right=False)

    # Seleccionar un estil de Matplotlib
    plt.style.use('bmh')

    # Crear la figura
    plt.figure(figsize=(12, 8))

    # Crear el gràfic de dispersió amb una línia de regressió
    ax = sns.scatterplot(x=dataframe['price'], y=dataframe['weight'], hue=dataframe['weight_range'],
palette='viridis')
    sns.regplot(x='price', y='weight', data=dataframe, scatter=False, ax=ax, color='royalblue')

    # Configurar les etiquetes dels eixos i el títol
    plt.title('Relació entre Preu i Pes dels Productes', fontsize=16)
    plt.xlabel('Preu (€)')
    plt.ylabel('Pes (kg)')

    # Establir els límits de l'eix Y perquè comenci a 0
    plt.ylim(0, dataframe['weight'].max() + 0.5)
    plt.xlim(0)

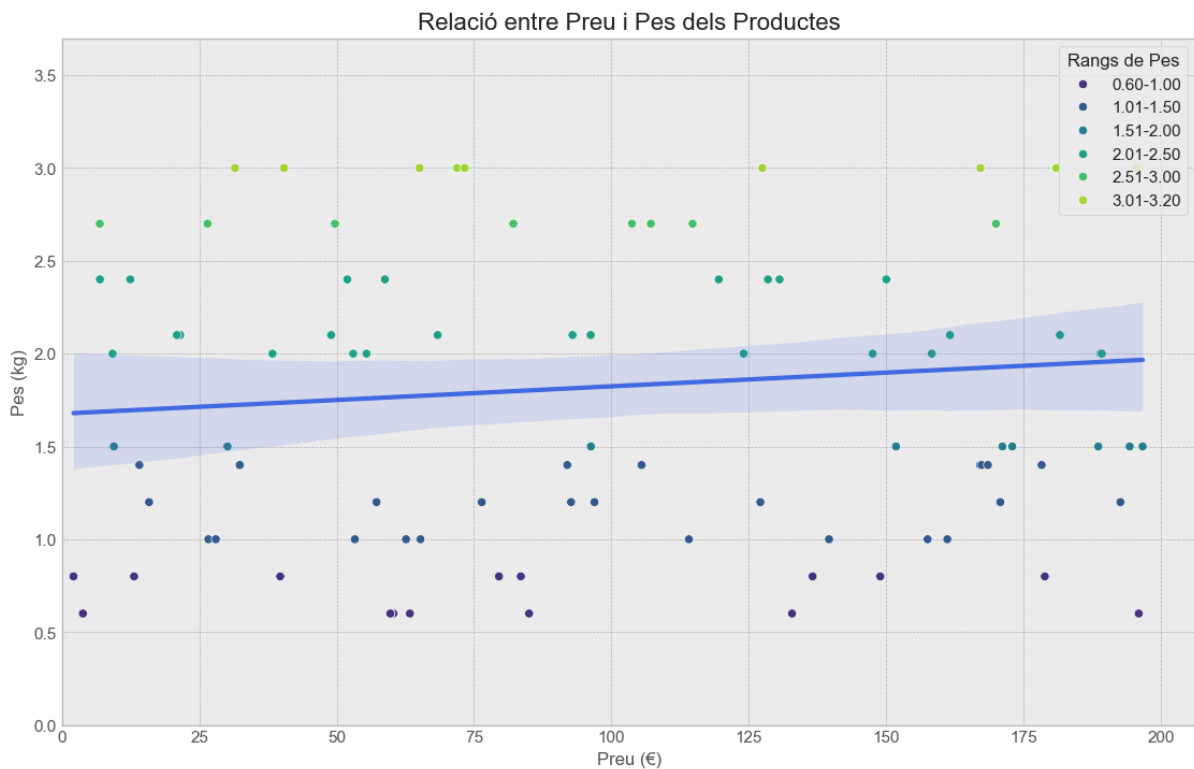
    # Col·locar la llegenda
    plt.legend(title='Rangs de Pes', loc='upper right')

    # Afegir un marge al voltant de la figura
    plt.gcf().set_facecolor('white')
    plt.gcf().tight_layout(pad=3.5)

```

```
# Mostrar el gràfic
plt.show()

print(f"Coefficient de correlació de Pearson: {correlation:.2f}")
print("Mitjana de pes per rang de pes:")
print(grouped)
else:
    print("Les columnes 'price' i/o 'weight' no existeixen a la taula 'products'.")
```



Coefficient de correlació de Pearson: 0.11

Mitjana de pes per rang de pes:

weight\_range

(0.6, 1.0] 86.0572

(1.0, 1.5] 118.6344

(1.5, 2.0] 107.113333

(2.0, 2.5] 84.42875

(2.5, 3.0] 94.954706

(3.0, 3.2] 144.29

Name: price, dtype: object

El coeficient de correlació de Pearson, pintat just a sota del gràfic, mostra un valor de **0.11**, el que indica que hi ha una lleu correlació lineal positiva entre el pes i el preu. Com que es tracta d'una correlació tan dèbil no podem considerar que sigui rellevant. Cada rang de pes és representat per un color diferent.

També mostrem la mitjana dels preus dels productes agrupats per rangs de pes. Això és important perquè reforça que no hi ha una relació lineal clara entre les variables, ja que en alguns intervals, com el de (1.0, 1.5] veiem que la mitjana de preus és més alta (118.63 €) en comparació amb un rang de pes més gran. Si la mitjana de preus varia significativament entre els diferents intervals de pes,



suggereix que altres factors, a més del pes, poden estar influint en la determinació dels preus dels productes.

Aquesta interpretació de les dades pot indicar que probablement tenen una major influència en el preu dels productes altres factors. Aquests podrien ser la marca o bé les funcionalitats i característiques específiques, per exemple. Amb això sabem que el pes no té un impacte important en l'increment del producte, per la qual cosa hauríem de continuar explorant quins factors són els factors que hi influeixen.

Com anàlisi addicional podríem explorar més a fons la distribució de preus dins de cada rang de pes i examinar si hi ha correlacions amb altres variables, com ara la categoria del producte o la marca, per obtenir una comprensió més completa dels determinants del preu dels productes.

## Exercici 3

### Una variable categòrica

Trio com a variable categòrica el color dels productes i per la visualització un diagrama de barres per tal de veure la quantitat de productes de cada color. Amb diagrama de barres, cada entitat de la variable categòrica es representa com una columna, l'alçada de la barra representa el seu valor numèric.

Analitzo la variable 'colour' perquè penso que ens pots ajudar a comprendre millor el tipus de producte venem i pot impactar positivament diverses àrees del negoci, des de la gestió d'inventari fins a les estratègies de màrqueting i l'experiència del client.

In [402]:

```
# Establir la connexió
```

```
conn = connection()
```

```
# Crear un objecte cursor per executar consultes SQL
```

```
cursor = conn.cursor()
```

```
# Definir la consulta SQL per obtenir els colors i les seves ocurrences
```

```
cursor.execute("""
```

```
SELECT DISTINCT colour, count(*) as count
```

```
FROM products
```

```
GROUP BY colour
```

```
ORDER BY count DESC;
```

```
""")
```

```
# Obtenir tots els resultats de la consulta
```

```
results = cursor.fetchall()
```

```
# Tancar el cursor després d'obtenir els resultats
```

```
cursor.close()
```

```
conn.close()
```

```
# Establir l'estil del gràfic
```

```
plt.style.use('default')
```

```
# Crear un DataFrame a partir dels resultats obtinguts
```

```
df = pd.DataFrame(results, columns=['colour', 'count'])
```

```
# Seleccionar un estil de Matplotlib
plt.style.use('bmh')

# Crear el gràfic de barres amb colors personalitzats
plt.figure(figsize=(16, 10))
bars = plt.bar(df['colour'], df['count'], color=df['colour'], edgecolor='black')

# Afegir etiquetes i títol
plt.xlabel('Gamma de Colors')
plt.ylabel('Quantitat')
plt.title('Quantitat de Productes per Color', fontsize=16)

# Configurar les etiquetes de l'eix Y sense decimals
plt.yticks(range(0, int(df['count'].max()) + 1))

# Ocultar les etiquetes de l'eix X
plt.xticks([])

# Mostrar el gràfic
plt.show()

# Calcular estadístics descriptius
stats = df['count'].describe().round(2)

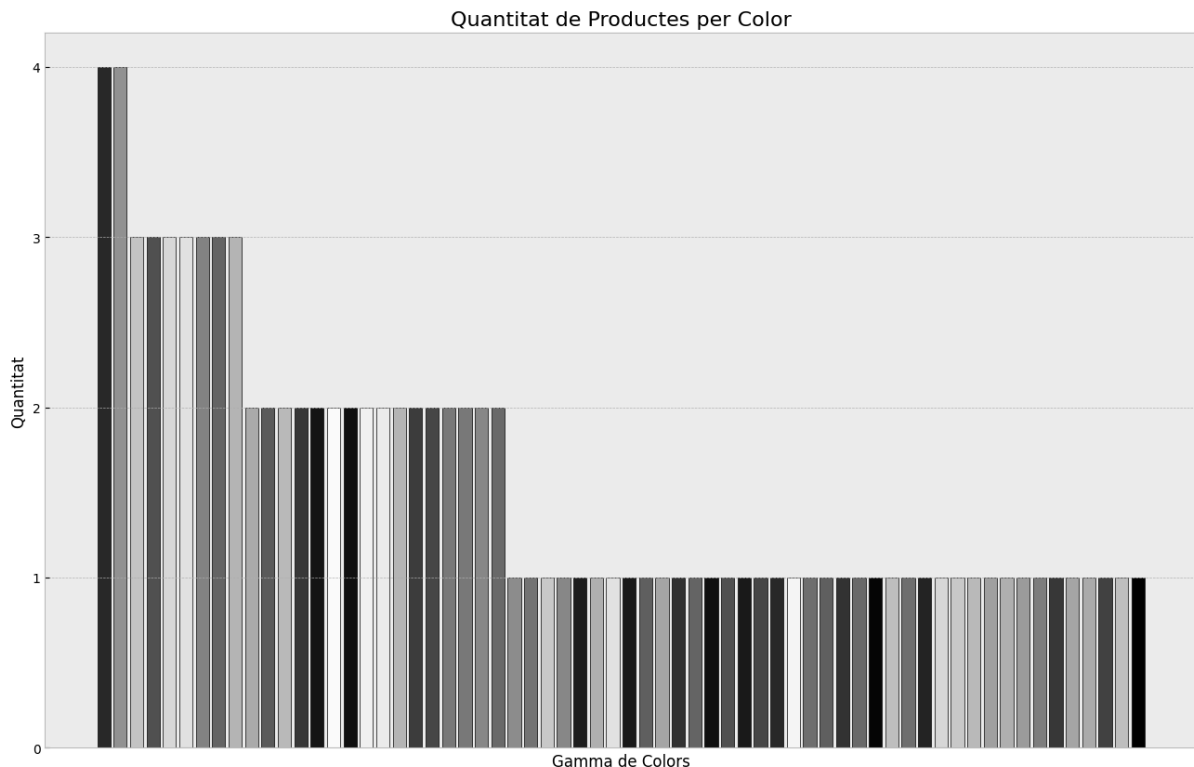
# Calcular la mediana
median_count = df['count'].median()

# Calcular la desviació estàndard
std_count = df['count'].std().round(2)

# Calcular el màxim i el mínim
max_count = df['count'].max()
min_count = df['count'].min()

# Calcular alguns quantils específics (25%, 50%, 75%)
quantiles = df['count'].quantile([0.25, 0.5, 0.75]).round(2)

# Mostrar els estadístics descriptius
print("Estadístics descriptius:")
print(stats)
```



Estadístics descriptius:

```
count    64.00
mean      1.56
std       0.81
min       1.00
25%       1.00
50%       1.00
75%       2.00
max       4.00
Name: count, dtype: float64
```

El primer que observem i que ens crida més l'atenció, és que els diferents codis de color hexadecimals de la taula 'products' pertanyen a 64 tons de gris. Si esperàvem veure l'arc de Sant Martí no és el cas. Això indica una diversitat considerable de tonalitats de gris.

Una anàlisi descriptiva més detallada ens mostra que la mitjana del nombre de productes per color gris és d'1.56, amb una desviació estàndard de 0.81. Això suggereix que hi ha una dispersió moderada respecte a la mitjana, indicant que alguns tons de gris poden ser més populars o freqüents que d'altres. La quantitat màxima de productes d'un mateix color de gris és de 4, mentre que la mínima és d'1. Podem cloure, doncs, que hi ha certa variabilitat.

Finalment, els quantils ens diuen que el 25% dels tons de gris tenen una quantitat d'1 producte, el 50% (mediana) té 1 producte i el 75% tenen 2 productes. Això reflecteix una distribució que es concentra principalment en tons de gris amb quantitats baixes, mentre que uns pocs colors de gris tenen quantitats més elevades.

A tall de conclusió, oferim una gamma de productes amb una àmplia varietat de tonalitats de gris per escollir. Encara que observem una mitjana d'1,56 productes per color, hi ha una dispersió significativa amb alguns tons de gris que tenen molts productes, fins a 4, i altres que en tenen només 1. El que pot reflectir tendències de demanda diverses o preferències dels clients per certes tonalitats.

Amb aquesta informació de base, podríem analitzar quins tons de gris són els que es venen més i quins se'n venen menys, per tal d'ampliar la gamma de productes en aquestes tonalitats específiques o desenvolupar variants de productes que incorporen aquests colors més demandats. De la mateixa manera, es podria ajudar a detectar patrons específics en les preferències de color entre segments de clients per regions geogràfiques i edat (informació de la qual disposem), per personalitzar les ofertes i atendre millor les necessitats de cada 'customer persona'. En qualsevol cas, tindria molta més utilitat analitzar la variable color, creuant-la amb altres, i aplicar-ho a estratègies de màrqueting o de planificació de la producció, per exemple.

## Exercici 4

### Una variable categòrica i una numèrica

Agafarem com a variable el país de les empreses i com a variable numèrica l'import (amount) de les transaccions. L'objectiu és poder comparar la distribució de les transaccions per país, i utilitzar un diagrama de caixes o 'boxplot' és una bona manera. Aquest tipus de diagrama ens facilita comparar de manera ràpida la distribució de l'import de les transaccions entre molts grups de dades (categories), en aquest cas entre 15 països, percebent d'una manera intuïtiva la mitjana, la dispersió de les dades i la seva simetria. En cas d'observar-se un biaix a les dades també podrem identificar en quina direcció es produeix aquest biaix.

Ho veiem:

In [403]:

```
# Establir la connexió
conn = connection()

# Crear un objecte cursor per executar consultes SQL
cursor = conn.cursor()

# Executar la consulta SQL
cursor.execute("""
SELECT t.amount, c.country
FROM transactions t
INNER JOIN companies c ON t.business_id = c.company_id
ORDER BY country
""")

# Obtenir tots els resultats de la consulta
results = cursor.fetchall()

# Tancar el cursor després d'obtenir els resultats
cursor.close()
conn.close()

# Establir l'estil del gràfic
plt.style.use('ggplot')

# Convertir els resultats a un DataFrame de pandas
df = pd.DataFrame(results, columns=['amount', 'country'])

# Imprimir el recompte de transaccions per país
transaction_counts = df.groupby('country').size()
print(transaction_counts)
```

```

# Calcular la desviació estàndard per país
std_devs = df.groupby('country')['amount'].std()

# Definir els llindars per categoritzar la variabilitat
low_threshold = std_devs.quantile(0.33)
high_threshold = std_devs.quantile(0.66)

# Crear una nova columna per a les categories de variabilitat
df['variability'] = df['country'].apply(lambda x: 'Baixa' if std_devs[x] <= low_threshold else ('Mitja' if
std_devs[x] <= high_threshold else 'Alta'))

# Definir la paleta de colors
palette = {'Baixa': 'lightgreen', 'Mitja': 'khaki', 'Alta': 'salmon'}

# Crear el boxplot amb colors personalitzats
plt.figure(figsize=(15, 8))

# Assegurar-se que els països s'ordenen correctament en el DataFrame
order = df['country'].unique()
sns.boxplot(x='country', y='amount', hue='variability', data=df, order=order, palette=palette,
dodge=False)

# Afegir etiquetes de mediana al boxplot
medians = df.groupby('country')['amount'].median()
for tick, label in enumerate(order):
    plt.text(tick, medians[label] + 0.5, f"{medians[label]:.2f}",
             horizontalalignment='center', size='medium', color='black')

# Configurar etiquetes dels eixos i títol
plt.title('Import de les Transaccions per País', fontsize=16)
plt.xlabel('País')
plt.ylabel('Import de les transaccions (€)')

# Rotar les etiquetes de l'eix X per millorar la llegibilitat
plt.xticks(rotation=45)

# Afegir la llegenda
handles, labels = ax.get_legend_handles_labels()
plt.legend(handles=handles, labels=['Baixa', 'Mitja', 'Alta'], title='Variabilitat', loc='upper left')

# Mostrar el gràfic
plt.show()

# Imprimir les mitjanes de l'import
print(medians)

# Mostrar la desviació estàndard per cada país
print("Desviació estàndard per país:")
print(std_devs)

```

country

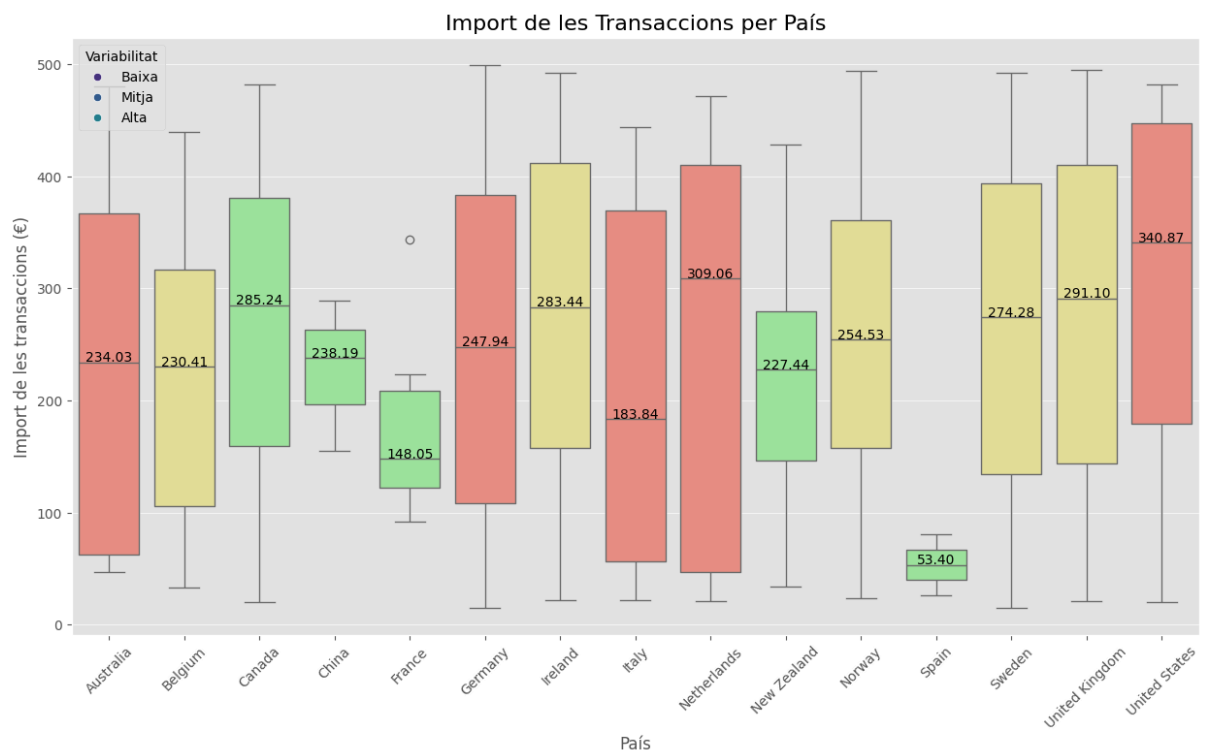
Australia	12
Belgium	13
Canada	61
China	3
France	6
Germany	118
Ireland	62
Italy	17
Netherlands	18
New Zealand	11
Norway	68
Spain	2
Sweden	79
United Kingdom	100
United States	17

dtype: int64

```

/var/folders/bq/gwwghf997q74ydx2r6mhgd2h0000gp/T/ipykernel_40835/438842331.py:68:
UserWarning: Mismatched number of handles and labels: len(handles) = 6 len(labels) = 3
plt.legend(handles=handles, labels=['Baixa', 'Mitja', 'Alta'], title='Variabilitat', loc='upper left')

```



country	
Australia	234.035
Belgium	230.41
Canada	285.24
China	238.19
France	148.045
Germany	247.935
Ireland	283.445

Italy	183.84
Netherlands	309.065
New Zealand	227.44
Norway	254.53
Spain	53.4
Sweden	274.28
United Kingdom	291.1
United States	340.87

Name: amount, dtype: object

Desviació estàndard per país:

country	
Australia	162.789664
Belgium	141.983942
Canada	131.335128
China	67.431749
France	92.285542
Germany	149.658844
Ireland	139.819171
Italy	154.779715
Netherlands	179.317567
New Zealand	127.830270
Norway	133.421060
Spain	38.438325
Sweden	143.587291
United Kingdom	145.882836
United States	160.292693

Name: amount, dtype: float64

Els diagrames 'boxplot' o diagrames de caixes ens mostren la distribució de l'import de les transaccions mitjançant quartils. El número que apareix just a sobre de la línia del centre de la caixa és la mediana, que divideix les dades en dues parts iguals. Les línies fora de la caixa, indiquen la variabilitat fora dels quartils.

Adicionalment, hem creat una nova columna anomenada 'variability' que basant-se en la desviació estàndard, classifica els països en 3 categories: Variabilitat baixa (color salmó), mitja (color groc) i alta (color verd).

Recordem que la desviació estàndard és una mesura d'extensió o variabilitat a l'estadística descriptiva i s'utilitza per calcular la variació o dispersió en què els punts de dades individuals difereixen de la mitjana. Una desviació baixa indica que els punts de dades estan molt a prop de la mitjana, mentre que una desviació alta mostra que les dades estan disperses en un rang més gran de valors. Així doncs, segons la nostra classificació observem que:

- Països amb Alta variabilitat: Austràlia, Alemanya, Itàlia, Països Baixos i els Estats Units.
- Països amb Mitja variabilitat: Bèlgica, Irlanda, Noruega, Suècia i Regne Unit.
- Països amb Baixa variabilitat: Canadà, Xina, França, Nova Zelanda i Espanya.

Pel que fa a la simetria de distribució, observem que ho és molt simètrica en el cas d'Alemanya, Noruega i Espanya. En canvi, hi ha una distribució asimètrica negativa a Austràlia, Itàlia i Països Baixos. Fent que la mitjana sigui menor que la mediana. I una distribució asimètrica positiva en la resta de països.

En l'àmbit del màrqueting, la desviació pot ajudar a tenir en compte la gran variació dels costos o les vendes. La desviació estàndard també ajuda a determinar la dispersió dels preus dels actius respecte al seu preu mitjà i la volatilitat al mercat.

Troblem únicament un punt de dades únic o valor atípic a França. En mostres petites, cada dada té més pes, per tant, els valors extrems poden aparèixer com a 'outliers', distorsionant la representació visual del 'boxplot'.

Considerar, però, que no podem extreure conclusions precipitades de la interpretació d'aquests resultats, ja que tenim una distribució desproporcionada de les dades originades per l'asimetria de la mostra: països com Alemanya i Regne Unit disposem d'unes 100 transaccions registrades i d'altres com Espanya i la Xina menys de 5. Aquesta diferència és molt gran. El perill de representar un nombre tan petit de transaccions pot representar incorrectament la distribució real de les dades. Els països amb poques transaccions tindran intervals de confiança més amplis, la qual cosa dificulta la confiança en les mitjanes, medianes o altres mesures estadístiques presentades. Amb tot això, ens hem de prendre les dades com una aproximació i no prendre decisions precipitades sense abans poder assegurar que les dades ens aporten un nivell suficient de significació estadística.

Recomanaria augmentar les transaccions on les transaccions es troben per sota les 50 i en futures anàlisis, aplicar la normalització de dades, per tal d'igualar la representació visual o filtrar les dades per excloure els països amb menys transaccions per a anàlisis més robustes.

## Exercici 5

### Dues variables categòriques

Agafem com a variables categòriques: 'country' de companyies i 'declined' de transaccions. Mostrarem mitjançant un diagrama de xarxa o 'spider chart' les transaccions declinades per país, ja que ens permet comparar fàcilment el total de declinacions de cada país de manera simultània, identificant quins països tenen més transaccions declinades en relació amb altres.

L'objectiu és detectar patrons o anomalies per facilitar la presa de decisions davant d'aquests resultats.

In [405]:

```
# Connectem
```

```
conn = connection()
```

```
# Cursor per executar consultes SQL
```

```
cursor = conn.cursor()
```

```
# Executar la consulta
```

```
cursor.execute("""
```

```
SELECT c.country, COUNT(t.declined) AS declined_count
```

```
FROM transactions t
```

```
INNER JOIN companies c ON t.business_id = c.company_id
```

```
WHERE t.declined = 1
```

```
GROUP BY c.country
```

```
""")
```

```
# Obtenir resultats
```

```
results = cursor.fetchall()
```

```
# Tancar el cursor i la connexió
```



```

cursor.close()
conn.close()

# Estil del gràfic
plt.style.use('ggplot')

# Convertir els resultats en un DataFrame de Pandas
df = pd.DataFrame(results, columns=['country', 'declined_count'])

# Calcular el valor mitjà de transaccions declinades
mean_declined = df['declined_count'].mean()

# Ordenar els països per número de transaccions declinades
country_counts_sorted = df.sort_values(by='declined_count',
ascending=False).reset_index(drop=True)

# Crear el gràfic
fig, ax = plt.subplots(figsize=(10, 8), subplot_kw={'projection': 'polar'})

# Calcular l'angle per cada país
angles = np.linspace(0, 2 * np.pi, len(country_counts_sorted), endpoint=False).tolist()

# Plotejar les línies del gràfic
values = country_counts_sorted['declined_count'].tolist()
values += values[:1]
angles += angles[:1]

ax.plot(angles, values, 'o-', linewidth=2)
ax.fill(angles, values, alpha=0.25)

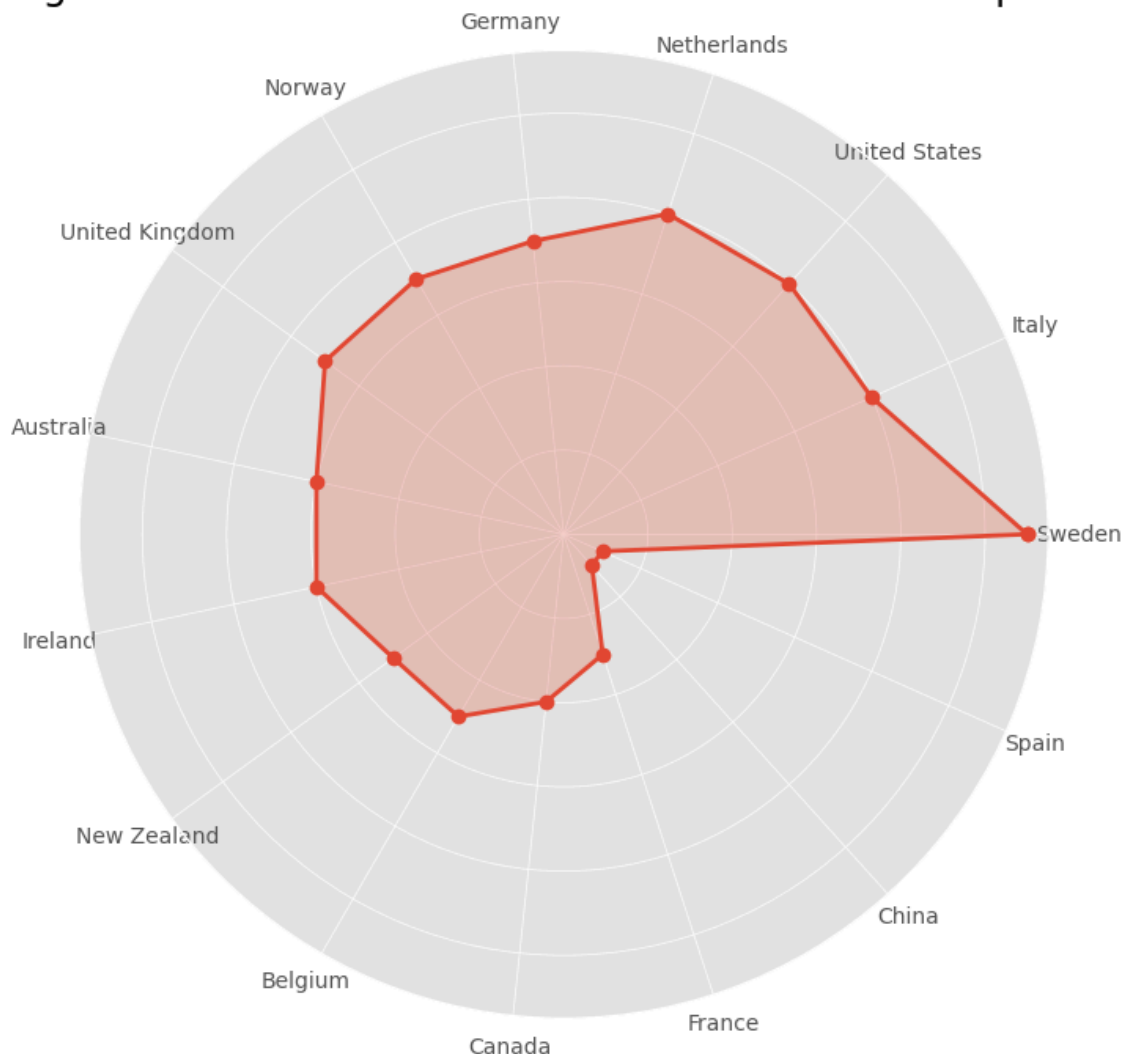
# Etiquetar els països
ax.set_yticklabels([])
ax.set_xticks(angles[:-1])
ax.set_xticklabels(country_counts_sorted['country'], size=10)

# Configurar el títol i mostrar el gràfic
plt.title('Diagrama de Xarxa de les Transaccions Declinades per País', size=18)
plt.show()

# Imprimir la mitjana de transacciones declinadas
print(f"Mitjana de transacciones declinadas: {mean_declined:.1f}")

```

## Diagrama de Xarxa de les Transaccions Declinades per País



Mitjana de transacciones declinades: 5.8

El primer que observem és que bona part dels països segueixen un mateix patró en les declinacions, 11 de 15 països es troben propers a la mitjana de transaccions declinades (5.8). Mentre que a Suècia veiem com creixen desproporcionadament fins arriba a les 11 declinacions. Per contra, Espanya, Xina i França en tenen menys que el patró general, per sota de 3.

Aquestes dades ens poden portar a fer les següents suposicions:

- On la distribució és uniforme, pot indicar una bona implementació global dels sistemes de pagament de l'empresa.
- On hi ha un baix nombre de declinacions pot suggerir que aquests països tenen sistemes de pagament més fiables o polítiques menys restrictives.
- On hi ha un alt nombre de declinacions, com és Suècia, pot indicar problemes amb el processador de pagaments local o polítiques estrictes de les institucions financeres.

Seria recomanable, doncs, investigar quins són els factors que realment influeixen en la quantitat de transaccions declinades de Suècia, i tenir en compte el context complet del mercat financer i les polítiques locals, a l'hora de prendre decisions d'expansió del negoci a nous països.

## Exercici 6

### Tres variables

Per aquest exercici hem seleccionat dues variables numèriques i una categòrica. Les variables són: 'amount' de transactions, 'country' i 'age' de users. Aquesta última variable l'haurem de calcular prèviament.

L'objectiu és observar si hi ha cap relació entre aquestes tres variables mitjançant la visualització d'un gràfic d'àrees apilades. Esperem veure com es distribueix la mitjana de l'amount en funció de l'edat dels usuaris i el país de residència, per identificar patrons i comparar tendències entre diferents grups demogràfics i geogràfics.

In [406]:

```
# Importar el mòdul per manipular dates i hores
from datetime import datetime

conn = connection()

cursor = conn.cursor()

cursor.execute("""
SELECT t.amount, u.country, u.birth_date
FROM transactions t
JOIN users u ON u.id = t.user_id
""")

results = cursor.fetchall()

cursor.close()
conn.close()

df = pd.DataFrame(results, columns=['amount', 'country', 'birth_date'])

# Calcular l'edat a partir de la data de naixement
df['birth_date'] = pd.to_datetime(df['birth_date'])
df['age'] = df['birth_date'].apply(lambda x: (datetime.now() - x).days // 365)

# Agrupar per país i edat, i calcular la mitjana de l'amount
grouped_df = df.groupby(['country', 'age']).agg({'amount': 'mean'}).reset_index()

# Crear una taula pívota pel heatmap
pivot_df = grouped_df.pivot(index='age', columns='country', values='amount').fillna(0)

# Crear el heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(pivot_df, cmap="YlGnBu", annot=True, fmt=".2f", linewidths=.5)

# Afegir el símbol de € a les anotacions
for text in plt.gca().texts:
    value = text.get_text()
    if value != 'nan':
        text.set_text(f'{value}€')
```

```
plt.title('Mitjana Import de les Transaccions per Edat i País')
plt.xlabel('País')
plt.ylabel('Edat')
plt.show()
```

*# Calcular i imprimir el total d'usuaris per país, la mitjana de l'edat i l'amount*

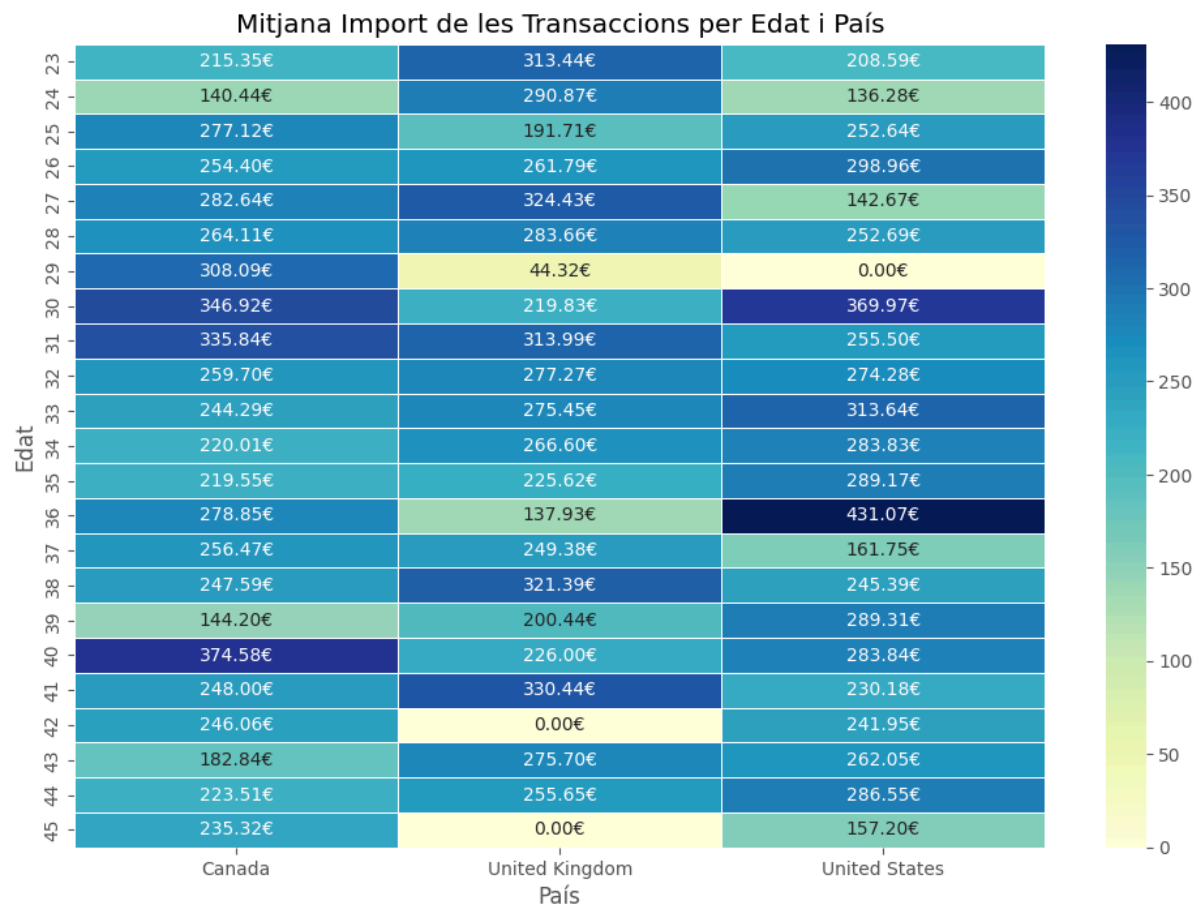
```
user_stats = df.groupby('country').agg({
    'amount': 'mean',
    'age': 'mean',
    'country': 'count'
}).rename(columns={'country': 'total_users', 'amount': 'average_amount', 'age':
'average_age'}).reset_index()
```

```
print(user_stats)
```

/var/folders/bq/gwwghf997q74ydx2r6mhgd2h0000gp/T/ipykernel\_40835/726336177.py:29:

FutureWarning: Downcasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and will change in a future version. Call result.infer\_objects(copy=False) instead. To opt-in to the future behavior, set `pd.set\_option('future.no\_silent\_downcasting', True)`

```
pivot_df = grouped_df.pivot(index='age', columns='country', values='amount').fillna(0)
```



```
country average_amount average_age total_users
0 Canada 253.559375 33.795455 352
1 United Kingdom 245.424032 32.887097 62
```

Aquest mapa de calor s'ha d'interpretar tenint en compte que a l'eix X trobem representats els 3 països: Canadà, Estats Units i Regne Unit. I en l'eix Y tenim les diferents edats dels usuaris, que van dels 23 als 45 anys. Al costat dret del gràfic tenim la llegenda per interpretar els diferents colors. El color canvia en funció de la mitjana de l'amount, els tons més foscos (verd i blau) apunten a una mitjana més alta, i els tons més clars (groc) són els que tenen una mitjana més baixa. Finalment, com es pot intuir a cada cel·la hi podem llegir la mitjana de l'amount per cada combinació de país i edat.

Pel que fa a la mitjana d'edat dels usuaris de cada país, tenim que els Estats Units té la mitjana d'edat més alta amb 34,7 anys, després ve el Canadà amb 33,7 i finalment Regne Unit amb 32,8. No observem grans diferències entre els països. Hem afegit el nombre total d'usuaris per país perquè no és el mateix extreure conclusions amb una mostra més gran o més petita.

Dit això, no observem una tendència general, on certs grups d'edat tendixin a fer transaccions amb un amount més alt o més baix, ni grans diferències entre països. De fet ho podem veure en el print de 'users\_stats' on la mitjana de l'import no varia molt significativament.

D'altra banda, sí que podem intuir patrons més específics, on certs grups d'edat en determinats països tendeixen a tenir transaccions més elevades i en d'altres més baixes. Vegem-ho:

- El Canadà troba la seva mitjana més alta en els 40 anys amb 374,58 € i la franja de 29 a 31 anys també es manté alta, per sobre del 300 €. Per contra, troba la seva mitjana més baixa amb 61,12 € just un any abans del seu màxim, als 39 anys. Els 24 anys també té una mitjana baixa de 140,44 €.
- A Regne Unit, els usuaris que gasten més de mitjana són els de 33 anys amb 398,34 € i el segueixen amb mitjanes també altes els grups de 41, 38, 31, 27 i 23 anys. Tenim un 'gap' amb el grup de 42 i 45 anys, ja que no han fet cap transacció. Els grups d'edat que gasta menys són els de 29 anys amb 44,32 €, i els de 34 i 36 tenen una mitjana de 137 € aprox.
- Als Estats Units hi trobem els usuaris amb una mitjana general més alta i el grup d'edat estrella és el de 36 anys amb una mitjana 431,07 € i els de 30 amb 369,97 €. No es té mercat amb el grup de 29 anys i una mitjana baixa d'amount amb els de 25, 45, 27 i 37 per sota del 200 €.

Amb aquestes dades podem concloure que s'han identificat diferents patrons de despesa. Hi ha grups amb alt i baix rendiment pel que fa a la mitjana de transaccions, per la qual cosa utilitzar la segmentació per edat i país a l'hora d'enfocar les estratègies de màrqueting podria ser molt rendible. També ens podríem plantejar implementar programes d'afiliació i retenció per als grups d'edat que mostren les mitges més altes de despesa i oferir promocions especials en moments clau (per exemple, aniversaris, aniversaris de registre, etc.). En definitiva, aplicar qualsevol estratègia que ajudi a maximitzar el valor del client, augmentar l'afiliació i retenció, i obtenir el màxim retorn de la inversió.

## Exercici 7

### Graficar un Pairplot

El Pairplot de Seaborn és una eina de visualització de dades que crea una matriu de diagrames de dispersió que mostra relacions per parells entre variables en un conjunt de dades, cosa que ajuda a visualitzar correlacions i distribucions.

Nosaltres volem observar quines relacions trobem entre la mitjana de l'import, el pes i els preus dels productes pels diferents països. Amb aquestes variables, podem explorar si hi ha alguna relació entre el preu i el pes dels productes, així com veure si aquestes característiques afecten a la quantitat de la transacció.

```

# Establir la connexió
conn = connection()

# Crear un objecte cursor per executar consultes SQL
cursor = conn.cursor()

# Consulta SQL original amb la selecció de dades
query = """
SELECT u.country, t.amount, p.weight, p.price
FROM users u
JOIN transactions t ON u.id = t.user_id
JOIN products p ON t.product_ids = p.id
"""

# Executar la consulta SQL
cursor.execute(query)

# Obtenir tots els resultats de la consulta
results = cursor.fetchall()

# Tancar el cursor i la connexió després d'obtenir els resultats
cursor.close()
conn.close()

# Convertir els resultats en un DataFrame de Pandas
df = pd.DataFrame(results, columns=['country', 'amount', 'weight', 'price'])

# Calcular la mitjana de l'amount per país
df_avg_amount = df.groupby('country')['amount'].mean().reset_index()
df_avg_amount.rename(columns={'amount': 'average_amount'}, inplace=True)

# Funció per obtenir una mostra aleatòria de 62 registres per país
def get_random_sample(df, sample_size):
    return df.sample(min(len(df), sample_size))

# Aplicar la funció per obtenir una mostra aleatòria amb 62 registres per país
df_sampled = df.groupby('country', group_keys=False).apply(lambda x: get_random_sample(x, 62))

# Unir la mitjana de l'amount amb les altres dades
df_sampled = df_sampled.merge(df_avg_amount, on='country')

# Triar les variables seleccionades per al pairplot
selected_vars = ['average_amount', 'weight', 'price']

# Configurar el gràfic
plt.figure(figsize=(14, 10)) # Ajustar la mida de la figura

# Crear el pairplot amb la tercera variable
pairplot = sns.pairplot(df_sampled, vars=selected_vars, height=4, aspect=1.5, hue='country',
palette='husl')

```

```
# Afegir un títol al gràfic
pairplot.fig.suptitle("Mitjana de l'import, el pes i els preus dels productes pels diferents països",
y=1.02, fontsize=24)
```

```
# Ajustar la posició de la llegenda a la part superior
pairplot.fig.subplots_adjust(top=0.95)
```

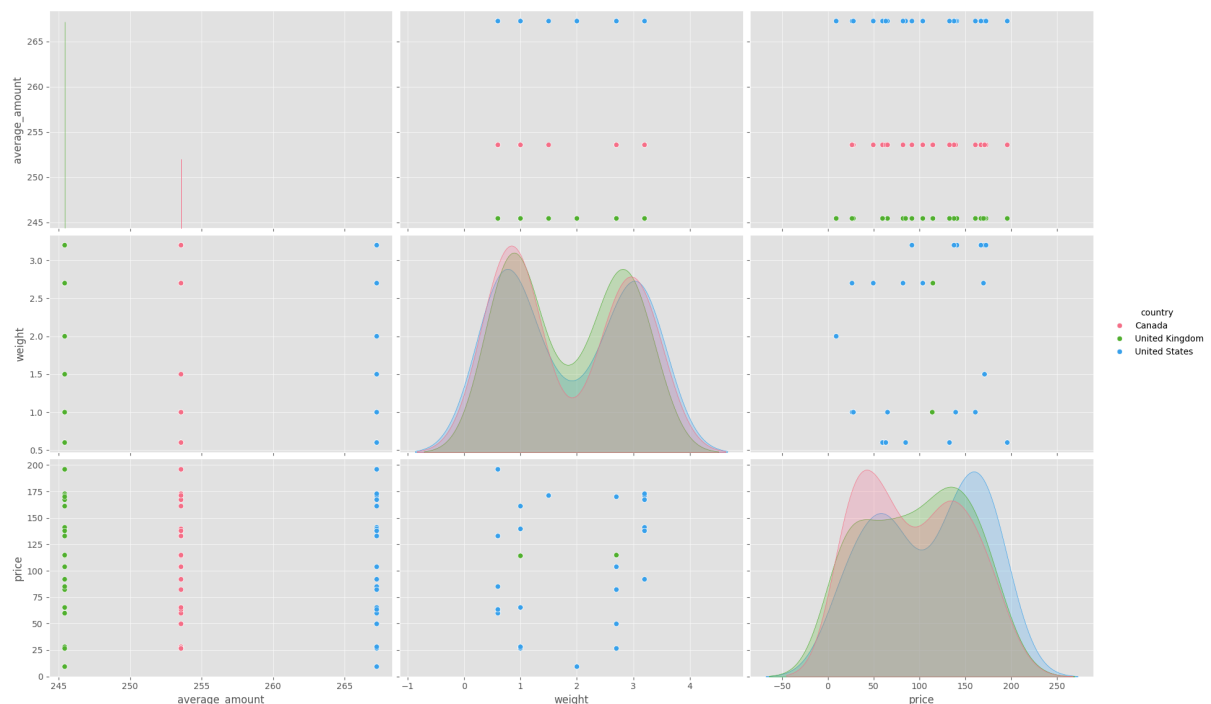
```
# Mostrar el gràfic
plt.show()
```

```
# Mostrar la taula de mitjana de l'amount per país
print(df_avg_amount)
```

```
/var/folders/bq/gwwghf997q74ydx2r6mhgd2h0000gp/T/ipykernel_40835/2355277550.py:37:
DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is
deprecated, and in a future version of pandas the grouping columns will be excluded from the
operation. Either pass `include_groups=False` to exclude the groupings or explicitly select the
grouping columns after groupby to silence this warning.
df_sampled = df.groupby('country', group_keys=False).apply(lambda x: get_random_sample(x, 62))
```

<Figure size 1400x1000 with 0 Axes>

Mitjana de l'import, el pes i els preus dels productes pels diferents països



```
country average_amount
0    Canada    253.559375
1  United Kingdom    245.424032
2    United States    267.251792
```

Els diferents colors identifiquen el país dels usuaris, tal com es mostra en la llegenda situada a la dreta. Per interpretar el pairplot ho hem de fer la següent manera: els gràfics en diagonal mostren la

distribució de cada variable individualment. Així doncs, el gràfic superior esquerre representa la mitjana de l'import dels tres països (tot i que per algun motiu la línia blava dels Estats Units no es visualitza). El Canadà amb una mitjana de 253,55 €, Regne Unit 245,24 € i els Estats Units la més alta amb 267,25 €. El gràfic del centre mostra la distribució del pes, que ja havíem analitzat a l'exercici 1, però fent servir un gràfic de barres. Com s'observa, en agafar una mostra amb una grandària exactament igual per a cada país, la distribució del pes és molt similar entre els països. Hem decidit obtenir una mostra aleatòria de 62 registres per país, perquè el nombre de transaccions per cada país té proporcions molt diferents: Canadà 352, USA 173 i UK 62 registres.

Hi ha dos pics per sota d'un quilo i pròxims als tres quilos. Tot i que els Estats Units té un pic més baix en el primer i més alt en el segon. Encara que la tendència és molt similar, sembla que Regne Unit prefereix productes de pes inferior i els Estats Units superior. Per acabar amb la diagonal tenim el gràfic inferior dret amb la distribució de l'última variable, el preu. Aquí observem que els usuaris dels Estats Units, en proporció a la resta, trien de manera clara productes més cars. A més a més, trobem dos repunts en productes del voltant de 60 €, fa una davallada en els de 110 € i torna a assolir el seu màxim prop del 170 €. Pel que fa als usuaris de Regne Unit genera unes corbes més suaus, començant abans i assolint el primer pic als 30 € i a partir d'aquí va pujant més lleugerament fins als 140 €. Pel que fa al Canadà fa un primer pic passats els 50 € i el segon abans del 150 €. La distribució d'aquest país és molt més compacta. Per acabar, podem concloure que hi ha més similitud entre els usuaris de Regne Unit i el Canadà. Els Estats Units té un comportament de preus que es diferencia.

Passem a la resta dels gràfics, els que es mostren fora de la diagonal. Aquests ens ensenyen la relació entre dos variables:

- Relació mitjana de l'import i pes (gràfics superior-centre i centre-esquerra): no hi ha cap relació.
- Relació mitjana de l'import i preu (gràfics superior-dreta i inferior-esquerra): no hi ha cap relació.
- Relació pes i preu (gràfics centre-dreta i inferior-centre): no hi ha cap relació.

## Conclusió

Les dades ens han mostrat que no hi ha cap correlació lineal evident entre aquestes tres variables. Pel que un canvi en una de les variables no té per què afectar l'altre. En definitiva, aquest resultat pot suggerir que la interacció entre les variables sigui molt més complexa o es vegi afectada per altres factors que no considerem aquí. Tot i que per mi, la hipòtesi més plausible, és que el nostre negoci disposa de poques dades per fer evidents aquest tipus de relacions.

# NIVELL 2

## Exercici 1

### Correlació de totes les variables numèriques

Primer localitzem totes les variables numèriques, sense transformar, de la nostra base de dades 'test'. Aquestes són el preu, pes i import de les transaccions. Com que són només tres variables una bona manera d'observar la correlació pot ser utilitzant un gràfic de dispersió tridimensional de la llibreria matplotlib.

Aquest gràfic podrem visualitzar com estan distribuïdes les variables weight, price i amount en un espai tridimensional, proporcionant una perspectiva visual de la seva relació.

Vegem-ho:



In [417]:

```
# Importar la llibreria
from mpl_toolkits.mplot3d import Axes3D

# Establir la connexió i obtenir les dades (assumint que `connection()` retorna la connexió)
conn = connection()

# Crear un objecte cursor per executar consultes SQL
cursor = conn.cursor()

# Consulta SQL original amb la selecció de dades
query = """
SELECT p.weight, p.price, t.amount
FROM products p
JOIN transactions t ON t.product_ids = p.id;
"""

# Executar la consulta SQL
cursor.execute(query)

# Obtenir tots els resultats de la consulta
results = cursor.fetchall()

# Tancar el cursor i la connexió després d'obtenir els resultats
cursor.close()
conn.close()

# Convertir els resultats en un DataFrame de Pandas
df = pd.DataFrame(results, columns=['weight', 'price', 'amount'])

# Convertir les columnes numèriques a tipus float per evitar errors
df['weight'] = df['weight'].astype(float)
df['price'] = df['price'].astype(float)
df['amount'] = df['amount'].astype(float)

# Crear la figura 3D
fig = plt.figure(figsize=(12, 9))
ax = fig.add_subplot(111, projection='3d')

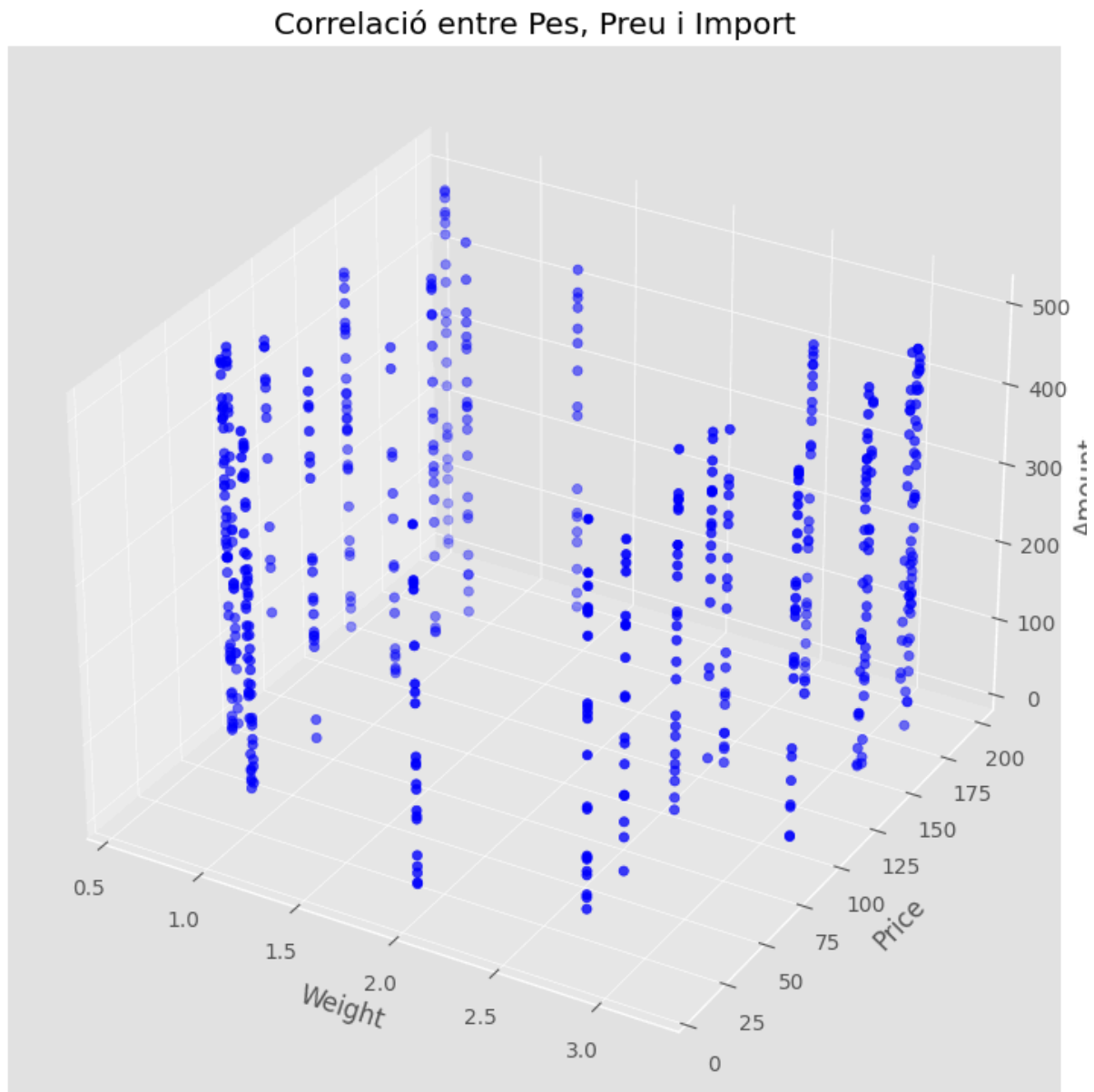
# Extreure les variables per a l'eix x, y, z
x = df['weight']
y = df['price']
z = df['amount']

# Crear el gràfic de dispersió 3D
sc = ax.scatter(x, y, z, c='blue', marker='o')

# Etiquetar els eixos
ax.set_xlabel('Weight')
ax.set_ylabel('Price')
ax.set_zlabel('Amount')
```

```
# Afegir un títol
ax.set_title('Correlació entre Pes, Preu i Import')

# Mostrar el gràfic
plt.show()
```



Hem creat un gràfic de dispersió tridimensional amb els punts que representen cada observació. Com ja hem vist amb anterioritat, ja que hem analitzat aquestes variables en altres exercicis, no evidenciem cap mena de correlació entre aquestes variables. Intentant analitzar el motiu pel qual això pot succeir, trobem que a vegades les variables poden tenir escales molt diferents entre si, cosa que pot fer que la relació no sigui evident visualment. Alternativament, podríem normalitzar o estandaritzar les variables i veure si ens mostra uns resultats diferents o potser explorar altres models de regressió no lineal que mostrin millor la relació entre variables.

## Exercici 2

## Implementa un jointplot

El jointplot de la llibreria Seaborn, permet dibuixar una gràfica de dues variables amb gràfics bivariants i univariants. Permet explorar la relació entre dos variables i obtenir les seves distribucions de manera individual. Així doncs, seleccionem les variables pertinents per mostrar la relació entre la quantitat de productes adquirits en cada transacció, per l'import total de la transacció, agrupat per país. L'objectiu és explorar simultàniament aquestes variables i les seves relacions, per identificar patrons i tendències en les dades de transaccions agrupades per país.

Mirem què surt...

In [428]:

```
# Establir la connexió i obtenir les dades
conn = connection()

# Crear un objecte cursor per executar consultes SQL
cursor = conn.cursor()

# Consulta SQL per obtenir les dades necessàries
query = """
SELECT u.country, t.amount, COUNT(tp.product_id) AS quantity
FROM users u
JOIN transactions t ON t.user_id = u.id
JOIN transactions_products tp ON tp.transaction_id = t.id
GROUP BY u.country, t.amount;
"""

# Executar la consulta SQL
cursor.execute(query)

# Obtenir tots els resultats de la consulta
results = cursor.fetchall()

# Tancar el cursor i la connexió després d'obtenir els resultats
cursor.close()
conn.close()

# Convertir els resultats en un DataFrame de Pandas
df = pd.DataFrame(results, columns=['country', 'amount', 'quantity'])

# Convertir les columnes numèriques a tipus float/int per evitar errors
df['amount'] = df['amount'].astype(float)
df['quantity'] = df['quantity'].astype(int)

# Configurar el gràfic de Seaborn
plt.figure(figsize=(12, 8))

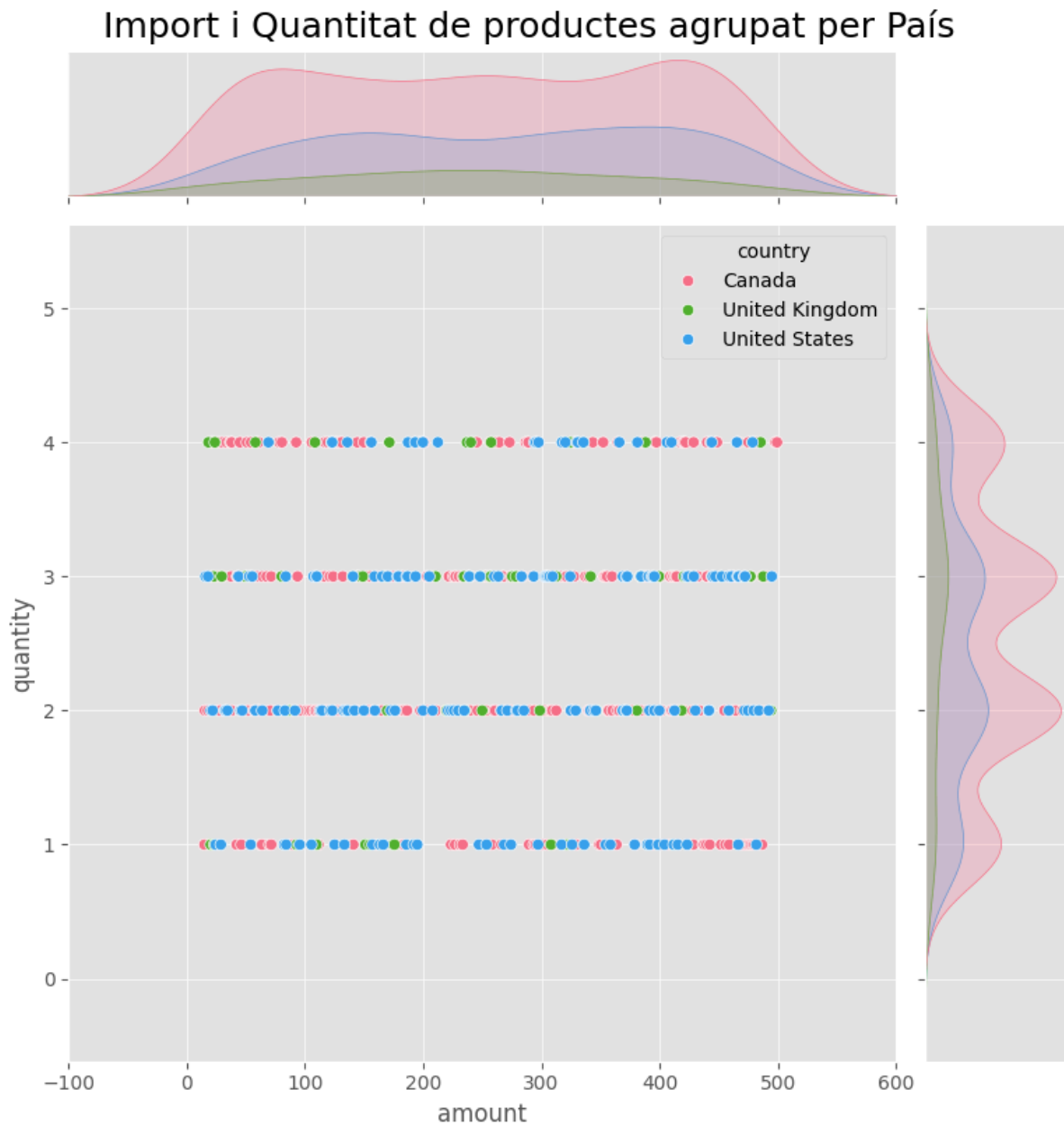
# Crear el jointplot amb les variables seleccionades
joint_plot = sns.jointplot(x='amount', y='quantity', data=df, kind='scatter', hue='country', height=8,
ratio=5, palette='husl')

# Ajustar els límits de l'eix X
joint_plot.ax_joint.set_xlim(-100, 600)
```

```
# Afegir títol
plt.suptitle('Import i Quantitat de productes agrupat per País', y=1.02, fontsize=18)

# Mostrar el gràfic
plt.show()
```

<Figure size 1200x800 with 0 Axes>



El jointplot ens mostra la relació entre la quantitat de productes obtinguts en una transacció i la suma total de la transacció, amb un scatterplot central i gràfics de distribució als laterals.

Observem la distribució de les variables:

### Import de Transaccions

- Canadà (color vermell): Té una distribució més típicament de campana on els usuaris tenen tres pics definits, un al voltant dels 60 €, un molt suau entre els 200-300 € i l'últim als 450 €.

- Regne Unit (color verd): Mostra una distribució força plana i uniforme, amb més transaccions entre 200-300 €. Una distribució amb cues menys pesades pot indicar menys risc de valors extrems.
- Estats Units (color blau): La seva distribució es troba entremig dels dos altres països, mostrant dos pics suaus als rangs 100-200 € i 400 €.

### Quantitat de Productes

- Canadà: La majoria d'usuaris adquireixen 2 i 3 productes per transacció, i en menys mesura 1 i 4.
- Regne Unit: Adquireixen 3 productes per transacció.
- Estats Units: Té similituds amb el Canadà. Veiem que la majoria adquireix 2 i 3 productes per transacció.

### Relació entre la Quantitat de Productes i l'Import de Transacció

La suma de l'amount es distribueix al llarg de tot el rang d'import, independentment de la quantitat de productes comprats. Observem molt pocs 'gaps'.

No s'observa una tendència clara segons el país de l'usuari. Sí que notem força menys presència del Regne Unit (punts de color verd), segurament degut a una menor mostra en comparació amb els altres dos països.

### Conclusió

Aquesta anàlisi no revela una relació clara entre la quantitat de productes adquirits i l'import de la transacció. Contràriament a la lògica que esperaria que comprar més productes comporti en una despesa més gran, no es veu aquesta tendència en les dades observades.

Aquest fet pot suggerir una major influència d'altres factors, com ara la diversitat de preus: comprar més productes barats pot resultar en un menor import total que comprar menys productes però més cars. Un altre factor podria ser l'aplicació de descomptes per a compres més voluminoses. Aquest podria estar distorsionant la relació esperada entre quantitat i import total.

## NIVELL 3

### Exercici 1

#### Implementa un violinplot combinat amb un altre tipus de gràfic

In [440]:

```
# Establir la connexió i obtenir les dades
conn = connection()

# Crear un objecte cursor per executar consultes SQL
cursor = conn.cursor()

# Consulta SQL per obtenir les dades necessàries
query = """
SELECT u.country, t.amount, COUNT(tp.product_id) AS quantity
FROM users u
JOIN transactions t ON t.user_id = u.id
JOIN transactions_products tp ON tp.transaction_id = t.id
GROUP BY u.country, t.amount;
```

"""

*# Executar la consulta SQL*

cursor.execute(query)

*# Obtenir tots els resultats de la consulta*

results = cursor.fetchall()

*# Tancar el cursor i la connexió després d'obtenir els resultats*

cursor.close()

conn.close()

*# Convertir els resultats en un DataFrame de Pandas*

df = pd.DataFrame(results, columns=['country', 'amount', 'quantity'])

*# Configurar la mida de la figura*

plt.figure(figsize=(12, 6))

*# Crear un grid de subplots para combinar els gràfics*

plt.subplot(1, 2, 1) *# Primer subplot pel Violinplot*

sns.violinplot(x='country', y='quantity', data=df, palette='husl')

plt.title('Quantitat de Productes per País')

plt.xlabel('País')

plt.ylabel('Quantitat')

plt.subplot(1, 2, 2) *# Segon subplot pel gràfic de barres*

sns.barplot(x='country', y='amount', data=df, estimator=sum, ci=None, palette='husl')

plt.title('Suma Import per País')

plt.xlabel('País')

plt.ylabel('Suma Import (€)')

*# Ajustar els límits de l'eix Y*

plt.ylim(0, df['amount'].sum() + 100) *# Ajustar el límit de l'eix Y*

*# Ajustar el layout dels subplots*

plt.tight\_layout()

*# Mostrar el gràfic*

plt.show()

/var/folders/bq/gwwghf997q74ydx2r6mhgd2h0000gp/T/ipykernel\_40835/1596411550.py:34:

FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

sns.violinplot(x='country', y='quantity', data=df, palette='husl')

/var/folders/bq/gwwghf997q74ydx2r6mhgd2h0000gp/T/ipykernel\_40835/1596411550.py:40:

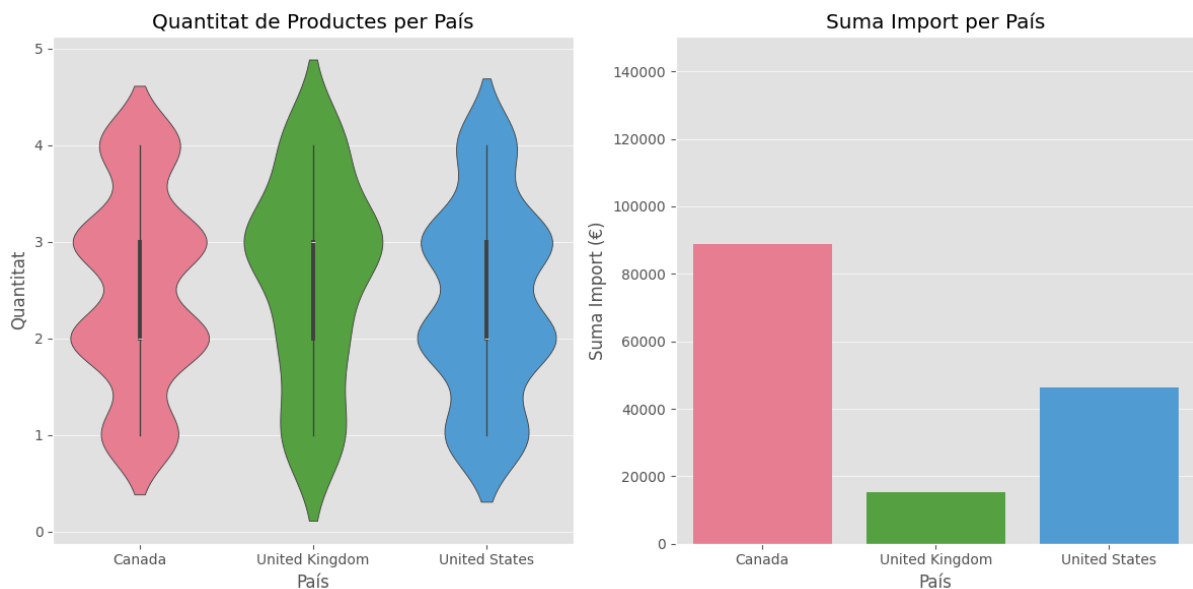
FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

```
sns.barplot(x='country', y='amount', data=df, estimator=sum, ci=None, palette='husl')  
/var/folders/bq/gwwghf997q74ydx2r6mhgd2h0000gp/T/ipykernel_40835/1596411550.py:40:  
FutureWarning:
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x='country', y='amount', data=df, estimator=sum, ci=None, palette='husl')
```



En aquesta ocasió hem combinat el violinplot de les quantitats de productes adquirits per país amb un gràfic de barres típic, que ens mostra el total del amount també per país.

### Violinplot: Quantitat de Productes per País

- Regne Unit:

La distribució mostra una forma allargada amb un pic destacat en la quantitat de 3 productes. Això indica que una gran part dels compradors del Regne Unit adquireixen exactament 3 productes per transacció. Aquesta preferència pot suggerir que els clients d'aquest país tendeixen a fer compres més planificades, possiblement aprofitant ofertes o promocions que impliquin la compra de múltiples productes. Tot i el pic en 3 productes, hi ha una variabilitat significativa en les quantitats de productes comprats, amb alguns clients que compren més o menys de 3 productes.

- Canadà:

La seva distribució és més simètrica, amb preferència per 2 i 3 productes. Els clients canadencs tendeixen a comprar entre 2 i 3 productes per transacció, amb una distribució més equilibrada entre aquestes dues quantitats. Aquesta consistència pot indicar una estabilitat en el comportament de compra, amb menys variació en les quantitats comprades per transacció en comparació amb el Regne Unit.

- Estats Units:

Els Estats Units també mostren una distribució simètrica amb preferència per 2 i 3 productes, com ja hem vist amb anterioritat. Hi ha una lleugera inclinació cap a la compra de 2 productes, amb un pic

menys pronunciat en 3 productes. El patró és similar al del Canadà, la qual cosa pot suggerir similituds culturals o d'hàbits de compra entre els dos països.

### Bar Chart: Suma Total de l'Import per País

- Canadà:

El Canadà té la xifra més alta amb 221.111,87 €. Això indica que, tot i tenir una preferència per comprar 2 o 3 productes, les transaccions totals són significativament més altes que en els altres països. Això podria ser degut a diversos factors com preus més alts dels productes, major poder adquisitiu, o una major freqüència de compra.

- Estats Units:

Els Estats Units tenen una suma total de 113.499,34 €, situant-se en una posició intermèdia. La preferència per 2 productes pot suggerir que els clients nord-americans realitzen compres més freqüents però amb una menor quantitat per transacció en comparació amb el Canadà.

- Regne Unit:

El Regne Unit té la suma total més baixa amb 40.531,86 €. Aquesta discrepància pot reflectir un comportament de compra diferent, amb una preferència més clara per una quantitat específica de productes però amb un valor total menor. Pot ser degut a preus més baixos dels productes, menys freqüència de compra, o altres factors econòmics i culturals que influeixin en les decisions de compra.

### Conclusió

Combinar l'anàlisi del violinplot i del gràfic de barres proporciona una visió més completa dels patrons de compra per país. El violinplot revela preferències específiques en la quantitat de productes comprats, mentre que el bar chart il·lustra les diferències en la suma total de les transaccions. Aquests resultats suggereixen que els compradors del Canadà tendeixen a gastar més globalment, tot i tenir una preferència similar per la quantitat de productes que els compradors dels Estats Units. El Regne Unit, tot i tenir una preferència clara per 3 productes, té una suma total de transaccions menor, suggerint possibles diferències en els preus dels productes o en el poder adquisitiu.

## Exercici 2

### Genera un FacetGrid per a visualitzar múltiples aspectes de les dades simultàniament

Un FacetGrid és un objecte de Seaborn que permet mapejar un conjunt de dades en diversos eixos ordenats en una graella de files i columnes que corresponen als nivells de variables del conjunt de dades. Així doncs, permet crear una matriu de subgràfics, cadascun mostrant una part diferent del conjunt de dades segons els valors d'una o més variables categòriques. Resulta molt útil per explorar com una variable depèn d'altres.

Nosaltres volem representar la quantitat de productes venuts en funció del seu identificador per cada país. D'aquesta manera podem comparar la distribució de les vendes de productes entre els diferents països.

Creem el Facetgrid:

In [395]:

```
# Establir la connexió i obtenir les dades (assumint que `connection()` retorna la connexió)
conn = connection()
cursor = conn.cursor()
```



```

# Consulta SQL per obtenir les dades necessàries
query = """
SELECT u.country, p.product_name, tp.product_id, COUNT(product_id) AS quantity
FROM transactions_products tp
JOIN transactions t ON t.id = tp.transaction_id
JOIN users u ON u.id = t.user_id
JOIN products p ON tp.product_id = p.id
GROUP BY product_id, u.country;
"""

# Executar la consulta SQL
cursor.execute(query)

# Obtenir tots els resultats de la consulta
results = cursor.fetchall()

# Tancar el cursor i la connexió després d'obtenir els resultats
cursor.close()
conn.close()

# Convertir els resultats en un DataFrame de Pandas
df = pd.DataFrame(results, columns=['country', 'product_name', 'product_id', 'quantity'])

# Ordenar el DataFrame per product_id
df = df.sort_values(by='product_id')

# Configurar la mida global de la figura
plt.figure(figsize=(18, 12))

# Configurar el gràfic de Seaborn
g = sns.FacetGrid(df, col='country', col_wrap=3, height=6) # Augmentar l'altura de cada subplot

# Funció per pintar les barres
def barplot_with_highlight(*args, **kwargs):
    data = kwargs.pop('data')
    ax = sns.barplot(*args, **kwargs, data=data, errorbar=None) # Canviat ci=None per
errorbar=None
    # Identificar els 5 productes més venuts
    top5 = data.nlargest(5, 'quantity')
    most_sold = data.nlargest(1, 'quantity')
    least_sold = data.nsmallest(1, 'quantity')
    # Pintar les barres amb els colors destacats
    for bar, product_id in zip(ax.patches, data['product_id']):
        if product_id in most_sold['product_id'].values:
            bar.set_color('seagreen')
        elif product_id in least_sold['product_id'].values:
            bar.set_color('r')
        elif product_id in top5['product_id'].values:
            bar.set_color('sandybrown')
        else:
            bar.set_color('dodgerblue')
    # Afegir la línia de la mitjana

```

```

mean_value = data['quantity'].mean()
ax.axhline(mean_value, color='black', linestyle='--')
ax.text(0, mean_value, f'Mitjana: {mean_value:.2f}', color='black', ha='right')
return ax

# Mapejar el gràfic de barres per a cada subplot amb barplot_with_highlight
g.map_dataframe(barplot_with_highlight, x='product_id', y='quantity', order=df['product_id'].unique())

# Ajustar l'escala de l'eix y
for ax in g.axes.flat:
    ax.set_ylim(0, 50) # Establir els límits de l'eix y

# Rotar les etiquetes de l'eix x
for ax in g.axes.flat:
    ax.set_xticks(np.arange(len(df['product_id'].unique())))
    ax.set_xticklabels(df['product_id'].unique(), rotation=90)

# Afegir títol als subplots
g.set_titles('{col_name}')

# Afegir títol general
g.fig.suptitle("Vendes de producte per país d'usuari", y=1.02, fontsize=20)

# Afegir etiquetes als eixos
g.set_axis_labels(x_var='ID Productes', y_var='Unitats venudes')

# Ajustar l'espai entre els subplots
plt.tight_layout()

# Mostrar el gràfic
plt.show()

# Imprimir una llista amb els noms dels productes
print("Llista de Productes:")
for product_id, product_name in df[['product_id',
'product_name']].drop_duplicates().sort_values(by='product_id').itertuples(index=False):
    print(f"Producte ID: {product_id} - Nom: {product_name}")

# Identificar els productes més venuts per cada país
most_sold_products = df.loc[df.groupby('country')['quantity'].nlargest(6).reset_index(level=0,
drop=True).index]
least_sold_products = df.loc[df.groupby('country')['quantity'].nsmallest(1).reset_index(level=0,
drop=True).index]
most_sold_product_per_country = df.loc[df.groupby('country')['quantity'].idxmax()]

print("\nEls 6 productes més venuts per país:")
for country, product_name, quantity in most_sold_products[['country', 'product_name',
'quantity']].itertuples(index=False):
    print(f"País: {country} - Producte: {product_name} - Quantitat: {quantity}")

print("\nProducte més venut per país:")

```

```

for country, product_name, quantity in most_sold_product_per_country[['country', 'product_name',
'quantity']].itertuples(index=False):
    print(f"País: {country} - Producte més venut: {product_name} - Quantitat: {quantity}")

```

```

print("\nProducte menys venut per país:")

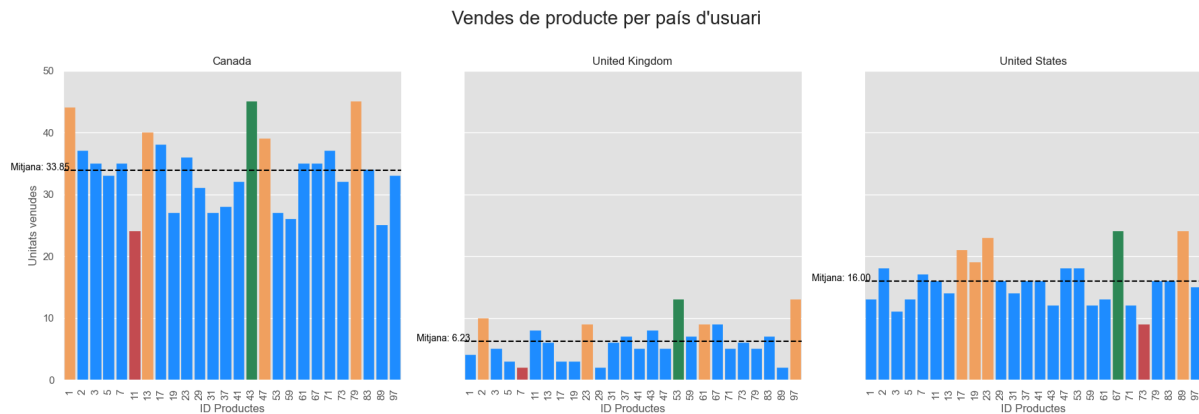
```

```

for country, product_name, quantity in least_sold_products[['country', 'product_name',
'quantity']].itertuples(index=False):
    print(f"País: {country} - Producte menys venut: {product_name} - Quantitat: {quantity}")

```

<Figure size 1800x1200 with 0 Axes>



Llista de Productes:

Producte ID: 1 - Nom: Direwolf Stannis  
 Producte ID: 2 - Nom: Tarly Stark  
 Producte ID: 3 - Nom: duel tourney Lannister  
 Producte ID: 5 - Nom: skywalker ewok  
 Producte ID: 7 - Nom: north of Casterly  
 Producte ID: 11 - Nom: Karstark Dorne  
 Producte ID: 13 - Nom: palpatine chewbacca  
 Producte ID: 17 - Nom: skywalker ewok sith  
 Producte ID: 19 - Nom: dooku solo  
 Producte ID: 23 - Nom: riverlands north  
 Producte ID: 29 - Nom: Tully maester Tarly  
 Producte ID: 31 - Nom: Lannister  
 Producte ID: 37 - Nom: Direwolf Littlefinger  
 Producte ID: 41 - Nom: Lannister Barratheon Direwolf  
 Producte ID: 43 - Nom: duel  
 Producte ID: 47 - Nom: Tully  
 Producte ID: 53 - Nom: kingsblood Littlefinger the  
 Producte ID: 59 - Nom: Direwolf Stannis  
 Producte ID: 61 - Nom: Winterfell Lannister  
 Producte ID: 67 - Nom: Winterfell  
 Producte ID: 71 - Nom: Tully Dorne  
 Producte ID: 73 - Nom: Dorne bastard  
 Producte ID: 79 - Nom: Direwolf riverlands the  
 Producte ID: 83 - Nom: duel tourney  
 Producte ID: 89 - Nom: skywalker ewok  
 Producte ID: 97 - Nom: jinn Winterfell

Els 6 productes més venuts per país:

País: Canada - Producte: duel - Quantitat: 45  
País: Canada - Producte: Direwolf riverlands the - Quantitat: 45  
País: Canada - Producte: Direwolf Stannis - Quantitat: 44  
País: Canada - Producte: palpatine chewbacca - Quantitat: 40  
País: Canada - Producte: Tully - Quantitat: 39  
País: Canada - Producte: skywalker ewok sith - Quantitat: 38  
País: United Kingdom - Producte: kingsblood Littlefinger the - Quantitat: 13  
País: United Kingdom - Producte: jinn Winterfell - Quantitat: 13  
País: United Kingdom - Producte: Tally Stark - Quantitat: 10  
País: United Kingdom - Producte: riverlands north - Quantitat: 9  
País: United Kingdom - Producte: Winterfell Lannister - Quantitat: 9  
País: United Kingdom - Producte: Winterfell - Quantitat: 9  
País: United States - Producte: Winterfell - Quantitat: 24  
País: United States - Producte: skywalker ewok - Quantitat: 24  
País: United States - Producte: riverlands north - Quantitat: 23  
País: United States - Producte: skywalker ewok sith - Quantitat: 21  
País: United States - Producte: dooku solo - Quantitat: 19  
País: United States - Producte: Tally Stark - Quantitat: 18

Producte més venut per país:

País: Canada - Producte més venut: duel - Quantitat: 45  
País: United Kingdom - Producte més venut: kingsblood Littlefinger the - Quantitat: 13  
País: United States - Producte més venut: Winterfell - Quantitat: 24

Producte menys venut per país:

País: Canada - Producte menys venut: Karstark Dorne - Quantitat: 24  
País: United Kingdom - Producte menys venut: north of Casterly - Quantitat: 2  
País: United States - Producte menys venut: Dorne bastard - Quantitat: 9

En primer lloc, observem que s'han creat tres subplots amb les vendes de productes dels tres països. En l'eix horitzontal tenim representat l'identificador de producte, volíem posar el nom del producte, però ocupava massa espai, per la qual cosa hem optat per imprimir un llistat amb l'identificador i el nom del producte després del gràfic. Les barres verticals informen de la quantitat de cada producte.

De manera general, observem que el Canadà és el país que ven, amb diferència, major quantitat de productes, la mitjana d'unitats venudes és de 33,85. Després ve els Estats Units amb una mitjana força inferior de 16 unitats venudes i per últim Regne Unit amb 6,23. A simple vista, la distribució dels productes entre països sembla tenir certes similituds, però ara ho analitzarem amb més detall.

**Producte més venut per país (barra de color verd):**

- Canadà - Producte més venut: *duel* - Quantitat: 45
- Els Estats Units - Producte més venut: *Winterfell* - Quantitat: 24
- Regne Unit - Producte més venut: *kingsblood Littlefinger the* - Quantitat: 13

**Producte menys venut per país (barra de color vermell):**

- Canadà - Producte menys venut: *Karstark Dorne* - Quantitat: 24
- Els Estats Units - Producte menys venut: *Dorne bastard* - Quantitat: 9
- Regne Unit - Producte menys venut: *north of Casterly* - Quantitat: 2

Aquestes observacions ens ajuden a veure les diferències i similituds en les preferències de productes entre països. També observem que al Canadà el top 5 de productes amb èxit són l'1, 13, 43, 47 i 79 (barres de color verd i taronja). No en comparteix cap amb els Estats Units i Regne Unit.

Trobem una única coincidència amb el producte amb identificador 23, *riverlands north*, dins del top 5 dels Estats Units i Regne Unit.

Aquestes observacions permeten iniciar una anàlisi sobre les vendes i investigar sobre quins poden ser els factors que fan que els productes més venuts varien entre països. Aquest coneixement és crucial per a l'empresa per poder adaptar les seves estratègies de mercat per satisfer millor les necessitats dels consumidors en cada regió.