Chocolate Bar Ratings Dataset

In this assignment, you will use a dataset with the ratings of several chocolate bars produced by companies located in different parts of the world. The dataset is a pre-processed version of the original dataset that can be found following this link (https://www.kaggle.com/rtatman/chocolate-bar-ratings).

Your focus will be on the ratings of chocolate bars produced in the UK and Switzerland. The ratings are in the range of 1-5; the higher the better.

KATE expects your code to define variables with specific names that correspond to certain things we are interested in.

KATE will run your notebook from top to bottom and check the latest value of those variables, so make sure you don't overwrite them.

- Remember to uncomment the line assigning the variable to your answer and don't change the variable or function names.
- Use copies of the original or previous DataFrames to make sure you do not overwrite them by mistake.

You will find instructions below about how to define each variable.

Once you're happy with your code, upload your notebook to KATE to check your feedback.

KATE expects your code to define variables with specific names that correspond to certain things we are interested in.

KATE will run your notebook from top to bottom and check the latest value of those variables, so make sure you don't overwrite them.

- Remember to uncomment the line assigning the variable to your answer and don't change the variable or function names.
- Use copies of the original or previous DataFrames to make sure you do not overwrite them by mistake.

You will find instructions below about how to define each variable.

Once you're happy with your code, upload your notebook to KATE to check your feedback.

Importing Libraries

Before proceeding to the questions, ensure that you run the code cell to import the necessary libraries.

In [5]:
```
# Import the usual suspects for data manipulation, statistics, and visua
import pandas as pd # used to "tidy" up and manipulate our data
import numpy as np # used for matrix and numerical calculations; the fou
from scipy import stats # contains stats functions and is used to visual
import matplotlib.pyplot as plt # used for visualisations
import seaborn as sns # a more user-friendly library used for visualisat
```

Dataset and variables

1. Load the dataset `flavors_cacao.csv` into a DataFrame called `choco_df` using the file path `'data/flavors_cacao.csv'` and display the first 5 rows of the DataFrame using the `.head()` method.

See below code syntax for some guidance:

```
choco_df = pd.read_csv(<file_path>)
choco_df.head()
```

In [7]:
```
#add your code below
#choco_df = ...
choco_df = pd.read_csv('data/flavors_cacao.csv')
choco_df.head()
```

Out[7]:

	company	species	REF	review_year	cocoa_p	company_location	rating	cou
0	Akesson's (Pralus)	Bali (west), Sukrama Family, Melaya area	636	2011	0.75	Switzerland	3.75	Indo
1	Akesson's (Pralus)	Madagascar, Ambolikapiky P.	502	2010	0.75	Switzerland	2.75	Madag
2	Akesson's (Pralus)	Monte Alegre, D. Badero	508	2010	0.75	Switzerland	2.75	E
3	Artisan du Chocolat	Trinidad, Heritage, Limited ed.	1193	2013	0.72	U.K.	3.25	Trin
4	Artisan du Chocolat	Colombia, Casa Luker	947	2012	0.72	U.K.	3.75	Colo

2. Using the `.loc` method select the column named `rating` from the DataFrame `choco_df`, Then filter the rows where the value in the column `company_location` is equal to `'U.K.'`. Store the resulting data in a variable called `uk_ratings`.

You can use `choco_df['company_location'] == 'U.K.'` as the filtering criteria while filtering rows.

See below code syntax for some guidance:

```
uk_ratings = choco_df.loc[<filtering_criteria>,<column_name>]
uk_ratings
```

Your answer should be a Pandas Series.

In [23]:
```
#add your code below
#uk_ratings = ...

uk_ratings = choco_df.loc[choco_df["company_location"] == "U.K.", "ratin
uk_ratings
```

Out[23]:
```
3      3.25
4      3.75
5      4.00
6      2.75
7      1.75
       ...
129    1.50
130    3.50
131    5.00
132    3.00
133    3.25
Name: rating, Length: 96, dtype: float64
```

3. Using the `.loc` method select the column named `rating` from the DataFrame `choco_df`, Then filter the rows where the value in the column `company_location` is equal to `'Switzerland'`. Store the resulting data in a variable called `swiss_ratings`.

You can use `choco_df['company_location'] == 'Switzerland'` as the filtering criteria while filtering rows.

See below code syntax for some guidance:

```
swiss_ratings = choco_df.loc[<filtering_criteria>,<column_name>]
swiss_ratings
```

Your answer should be a Pandas Series.

In [24]:
```
#add your code below
#swiss_ratings = ...

swiss_ratings = choco_df.loc[choco_df["company_location"] == "Switzerlan
swiss_ratings
```

Out[24]:
```
0      3.75
1      2.75
2      2.75
22     3.00
23     3.00
24     3.50
25     4.00
26     3.25
27     3.25
28     3.50
29     3.50
60     3.50
61     2.00
62     3.00
63     3.00
64     3.00
65     3.50
66     4.00
91     3.75
92     3.75
93     3.75
94     3.25
95     3.75
96     4.00
97     3.50
98     4.00
99     4.00
100    4.00
101    3.00
102    2.75
103    4.50
104    3.75
105    3.00
106    3.00
107    3.25
123    3.00
124    2.75
125    4.25
Name: rating, dtype: float64
```

4. How many rows are in `uk_ratings` ?

To determine the number of rows in the `uk_ratings` Pandas Series, you can use the `.shape[0]` attribute or `len()` function.

Store your answer in a variable called `uk_rows`

In [29]:
```
#add your code below
#uk_rows = ...

uk_rows = uk_ratings.shape[0]
uk_rows = len(uk_ratings)
uk_rows
```

Out[29]: 96

5. What is the mean rating of the chocolate produced by companies in the UK?

Refer to the `uk_ratings` Pandas Series. To calculate the mean rating of the chocolate produced by companies in the UK, you can use the NumPy `np.mean()` function.

See below code syntax for some guidance:

```
np.mean(uk_ratings)
```

Store your answer in a variable called `uk_mean_rating`

In [31]:
```
#add your code below
#uk_mean_rating = ...

uk_mean_rating = np.mean(uk_ratings)
uk_mean_rating
```

Out[31]: 3.0729166666666665

6. What is the median rating of the chocolate produced by companies in the UK?

Refer to the `uk_ratings` Pandas Series. To calculate the median rating of the chocolate produced by companies in the UK, you can use the NumPy `np.median()` function.

See below code syntax for some guidance:

```
np.median(uk_ratings)
```

Store your answer in a variable called `uk_median_rating`

In [52]:
```
#add your code below
#uk_median_rating = ...

uk_median_rating= np.median(uk_ratings)
uk_median_rating
```

Out[52]: 3.0

7. What is the Standard Error of the Mean (SEM) of the ratings of the chocolates produced by UK companies?

Refer to the `uk_ratings` Pandas Series. To calculate the `Standard Error of the Mean (SEM)` of the ratings of the chocolates produced by UK companies using the scipy library, you can utilize the sem() function from the stats module: `stats.sem()` .

See below code syntax for some guidance:

```
stats.sem(uk_ratings)
```

Store your answer in a variable called `uk_ratings_sem`

In [33]:
```
#add your code below
#uk_ratings_sem = ...

uk_ratings_sem = stats.sem(uk_ratings)
uk_ratings_sem
```

Out[33]: 0.05887708663316619

8. How many rows are in `swiss_ratings` ?

To determine the number of rows in the `swiss_ratings` Pandas Series, you can use the `.shape[0]` attribute or `len()` function.

Store your answer in a variable called `swiss_rows`

In [34]:
```
#add your code below
#swiss_rows = ...

swiss_rows = swiss_ratings.shape[0]
swiss_rows = len(swiss_ratings)
swiss_rows
```

Out[34]: 38

```
9. What is the mean rating of the chocolate produced by Swiss companies?**

Refer to the `swiss_ratings` Pandas Series. To calculate the mean rating of
the chocolate produced by  Swiss companies, you can use the NumPy
`np.mean()` function.

See below code syntax for some guidance:
```python
np.mean(swiss_ratings)
```
Store your answer in a variable called `swiss_mean_rating`
```

In [35]:
```
#add your code below
#swiss_mean_rating = ...

swiss_mean_rating = np.mean(swiss_ratings)
swiss_mean_rating
```

Out[35]: 3.401315789473684

10. What is the median rating of the chocolate produced by Swiss companies?

Refer to the `swiss_ratings` Pandas Series. To calculate the median rating of the chocolate produced by Swiss companies, you can use the NumPy `np.median()` function.

See below code syntax for some guidance:

```
np.median(swiss_ratings)
```

Store your answer in a variable called `swiss_median_rating`

In [37]:
```
#add your code below
#swiss_median_rating = ...

swiss_median_rating = np.median(swiss_ratings)
swiss_median_rating
```

Out[37]: 3.5

11. What is the Standard Error of the Mean (SEM) of the ratings of the chocolate produced by Swiss companies?

Refer to the `swiss_ratings` Pandas Series. To calculate the `Standard Error of the Mean (SEM)` of the ratings of the chocolates produced by Swiss companies using the scipy library, you can utilize the sem() function from the stats module: `stats.sem()` .

See below code syntax for some guidance:

```
stats.sem(swiss_ratings)
```

Store your answer in a variable called `swiss_ratings_sem`

In [55]:
```
#add your code below
#swiss_ratings_sem = ...

swiss_ratings_sem = stats.sem(swiss_ratings)
```

12. Use box plots to compare ratings from the UK and Switzerland

Refer to the `choco_df` DataFrame. Use `sns.boxplot()` to plot `rating` against `company_location` in order to compare national ratings.

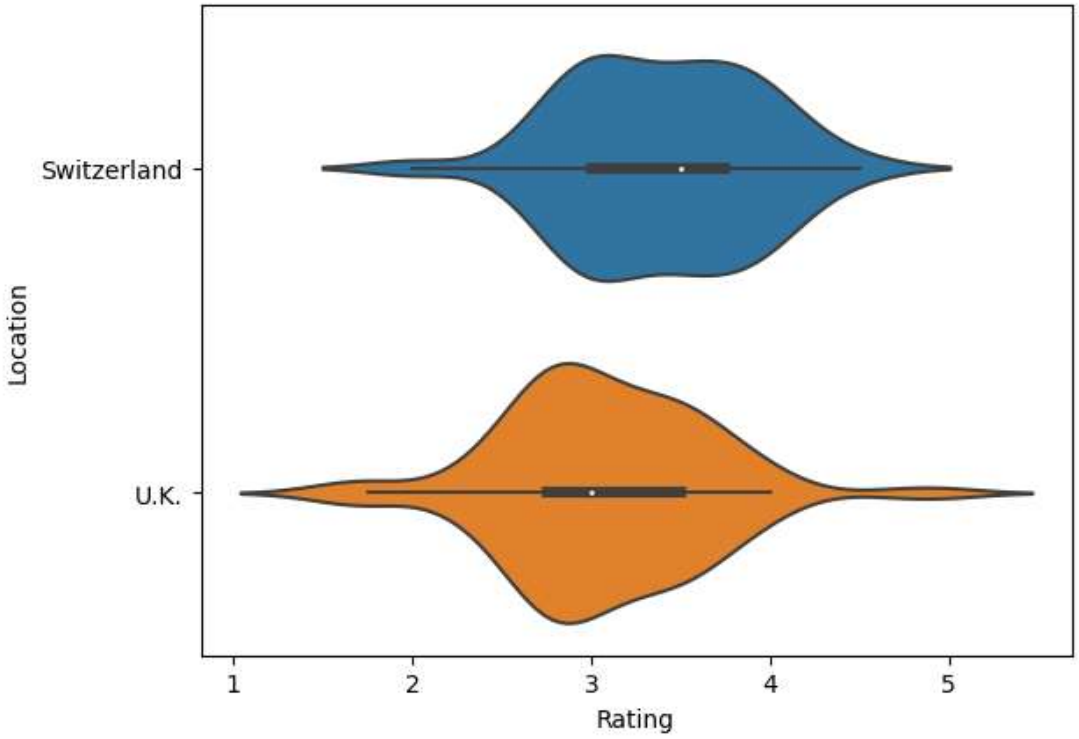
- Ensure you save your plot to the variable called `ratings_comp` .
- Ensure the plot has the x-label 'Rating'
- Ensure the plot has the y-label 'Location'

See below code syntax for some guidance:

```
ratings_comp = sns.violinplot(x=..., y=..., data=...)
ratings_comp.set(xlabel=..., ylabel=...)
```

In [56]:
```
#add your code below
import seaborn as sns
ratings_comp = sns.violinplot(x="rating", y="company_location", data=cho
ratings_comp.set(xlabel="Rating", ylabel="Location")
```

Out[56]: [Text(0.5, 0, 'Rating'), Text(0, 0.5, 'Location')]



Outlier detection

Are there any outliers in our data that might be skewing our statistics?

An outlier is an extreme value that lies outside the overall pattern of the data.

There are several different ways of determining outliers, depending on the nature of the data and the calculations that you are asked to carry out.

Here we will use the definition that an outlier is any value that is:

- either greater than $Q_3 + 1.5(Q_3 - Q_1)$
- or less than $Q_1 - 1.5(Q_3 - Q_1)$

where Q_3 is the upper quartile, and Q_1 is the lower quartile.

> Note: This is the default criteria in Seaborn used to mark outliers on boxplot diagrams like the one you created in the previous question

13 Create a function called `find_outliers` that accepts a DataFrame as input. The function will determine outliers in the `rating` column of the DataFrame using the specified criteria.

Upon execution, the function will return a list named `outliers` containing the identified outliers. It is assumed that the DataFrame has the same format as `choco_df` and includes a `rating` column.

See below code syntax for some guidance:

```
q1 = df['rating'].quantile(0.25)
q3 = df['rating'].quantile(0.75)
upper = q3 + 1.5*(q3-q1)
```

In [53]:
```
#add your code below
#def find_outliers(df):

df = choco_df

def find_outliers(df):
    q1 = df["rating"].quantile(0.25)
    q3 = df["rating"].quantile(0.75)

    upper = q3 + 1.5*(q3-q1)
    lower = q1 - 1.5*(q3-q1)

    outliers = df[(df["rating"] < lower) | (df["rating"] > upper)]["rati
    return outliers
```

14. Use your function `find_outliers` to determine the outliers in the `choco_df` data, first filtering the dataframe so it contains only ratings from the UK

Store your answer (the output of the `find_outliers` function) in a variable called `uk_outliers`

See below code syntax for some guidance:

```
uk_df = choco_df[choco_df['company_location']=='U.K.']
uk_outliers = find_outliers(uk_df)
uk_outliers
```

In [51]:
```
#add your code below
#uk_outliers = ...

uk_df = choco_df[choco_df["company_location"] == "U.K."]
uk_outliers = find_outliers(uk_df)
uk_outliers
```

Out[51]:
```
[4.75, 1.5, 5.0]
```

Well done for completing the assignment! As well as being able to calculate statistics, review distributions and graphs, it is also important to develop an understanding of what information we can draw from all of this. Post a comment to the Knowledge Base about a conclusion that you have been able to draw from the data as part of your work above.