Project 1: Ramenphile

Remarque: each record in the datasets is a single product review

On a verifié que le tableau s’affiche correctement data : data.head(10) *#show the first 10 rows of the data*

Et la taille du tableau e utilisant data.shape

Alright, we know that we got 7 columns in our data. Which are:

*Review* : unique numbers that inform the review order from the latest

1. *Brand* : Ramen brand
2. *Variety*: variation of ramen
3. *Style* : style of ramen
4. *Country*: Where the ramen is available
5. *Stars* : Ramen ratings
6. *Top Ten*: ramen achievement
7. Detect missing values for an array-like object.

data.isna()

This function takes a scalar or array-like object and indicates whether values are missing (NaN in numeric arrays, None or NaN in object arrays, NaT in datetimelike).

Drop the missing data

data = data.dropna(subset=['Style'])

print(data["Style"].isna().sum())

les fonctions utiles:

import numpy as np #linear algebra

import pandas as pd #data processing

import matplotlib.pyplot as plt #data visualisation

import seaborn as sns #data visualisation

|  |  |  |
| --- | --- | --- |
| **Fonction** | | **L’utilisation** |
| df = pd.read\_csv('ramen-good.txt',sep=";")  df.head() | | Lire les données et afficher les premiers cinqs ligne |
| df.isna().sum() | | Sommer le nombre de cases vides pour chaque colonnes |
| df=df.dropna(subset=['Style']) | | Supprimer les lignes où les cases sont vides |
| df['Style'].unique() | | Prélever les éléments constituents la colonnes sans répétition |
| df['Style'].value\_counts() | | Compter combine de fois chaques types est répété |
| top10=df.dropna() | |  |
| top10 = top10[top10['Top Ten'] != ' '] | | Redéfinir le data en exluant les lignes où “top ten” est vide |
| top10 = top10.sort\_values('Top Ten' ) | Trier le data en utilisant les données de la colonnes “top ten” | |
| df['Brand'].value\_counts()[:10] | | La même function d’avant mais on affiche que ls dix premiers de la liste |
| for s in df['Stars']:    try:      s=float(s)    except:      print(s) | | On cherche si il y a des valeurs qui sont pas valides |
| brands = list(df['Brand'].unique())  counter = [0.0]\*355  brands\_cnt = dict(zip(brands, counter)) #create dictionary to count all ratings and then save the averages  #print(brands\_cnt['Nissin'])  for brand in brands:      brands\_data = df[df['Brand'] == brand]      for star in brands\_data['Stars']:          brands\_cnt[brand] += float(star) #count all ratings      brands\_cnt[brand] /= len(brands\_data) #average  print(brands\_cnt) | | On affiche la moyenne des notes de chaque type de ramen “Brand” en utilisant un dictionnaire pour afficher à la fin |
| top50ratings = [] #list for saving the brand name and its average rating  for key, values in brands\_cnt.items():      top50ratings.append([key,values]) | | Convertir le dictionnaire en une liste |
| top50ratings = sorted(top50ratings, key = lambda x : x[1], reverse = True) #sorting values in descending order  output1=[]  output2=[]  output3=[]  for i in range(50):      output2+=[top50ratings[i][0]]      output3+=[round(top50ratings[i][1],2)]  d={'top50rating':output2,'top50ratinground':output3}  df\_topratings = pd.DataFrame(d)  df\_topratings.index += 1  df\_topratings.head() | | On trie la liste avec la clé qui est le deuxième élément de chaque mini liste.  Créer un tableau en utilisant data frame |
| #Data visualisation  #count Plot  sns.set(style = 'darkgrid')  f, ax = plt.subplots(1,1,figsize = (15,5))  sns.countplot(x = 'Country', data = df)  plt.xticks(rotation=90)  plt.show() | | Afficher les éléments de la colonne «country » |
| labels = 'Pack', 'Bowl', 'Cup' , 'Tray', 'Box' #We can't include 'Bar' and 'Can' because they only appear once in our data.  size = [1531, 481, 450, 108, 6]  f, ax = plt.subplots(1,1, figsize= (10,10))  ax.pie(size, labels = labels,autopct = '%1.2f%%', startangle = 180)  ax.axis('equal')  ax.set\_title("Style", size = 20)  plt.show() | | Afficher le pourcentage de charque type de ramen. |