Zapisz kopię pliku na dysku iz mień jego nazwę *Python_czyszczenieDanych_ImieNazwisko.jpynb*

Wczytaj biblioteki:

- numpy
- pandas

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

Wczytaj plik "flavors_of_cacao.csv" i wyświetl 5 pierwszych wierszy

```
data = pd. read_csv ('flavors_of_cacao.csv')
```

```
data.head(5)
```

	Company \n(Maker-if known)	Specific Bean Origin\nor Bar Name	REF	Review\nDate	Cocoa\nPercent	Company\nLocation
0	A. Morin	Agua Grande	1876	2016	63%	France
1	A. Morin	Kpime	1676	2015	70%	France
2	A. Morin	Atsane	1676	2015	70%	France

Next steps: View recommended plots

Zmień nazwy kolumn na:

'Company', 'Specific_Bean_Origin_or_Bar_Name', 'REF','Review_Date', 'Cocoa_Percentage', 'Company_Location','Rating', 'Bean_Type', 'Broad_Bean_Origin'

i wyświetl 3 pierwsze wiersze

```
data.columns=['Company', 'Specific_Bean_Origin_or_Bar_Name', 'REF','Review_Date', 'Cocoa_Percentage', 'Company_Location','Rating', 'Bean_Ty
data.head(3)
```

	Company	Specific_Bean_Origin_or_Bar_Name	REF	Review_Date	Cocoa_Percentage	Comp
0	A. Morin	Agua Grande	1876	2016	63%	
1	A. Morin	Kpime	1676	2015	70%	
2	A. Morin	Atsane	1676	2015	70%	

Next steps: View recommended plots

Wyfiltruj 3 wiersz (komórkę) dla zmiennej *Review_Date*

```
data.iloc[2]['Review_Date']
```

```
2015
```

Wyfiltruj komórki obejmujące kolumny od 2 do 4 i wiersze od 1 do 5.

```
data.iloc[1:6, 1:4]
```

	Specific_Bean_Origin_or_Bar_Name	REF	Review_Date
1	Kpime	1676	2015
2	Atsane	1676	2015
3	Akata	1680	2015
4	Quilla	1704	2015
5	Carenero	1315	2014

Wyświtl "kształt" zbioru - liczba wierszy i kolumn

```
data.shape
```

```
(1795, 9)
```

Wyświetl informcje o zbiorze danych

Czy wyświetlają się onformacje o "brakach" w zmiennej *Bean_Type* widocznych po użyciu funkcji head?

Dlaczego zmienna *Cocoa_Percentage* jest typu object?

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1795 entries, 0 to 1794
Data columns (total 9 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Company                         1795 non-null   object 
 1   Specific_Bean_Origin_or_Bar_Name  1795 non-null   object 
 2   REF                             1795 non-null   int64  
 3   Review_Date                     1795 non-null   int64  
 4   Cocoa_Percentage                1795 non-null   object 
 5   Company_Location                1795 non-null   object 
 6   Rating                          1795 non-null   float64
 7   Bean_Type                       1794 non-null   object 
 8   Broad_Bean_Origin               1794 non-null   object 
dtypes: float64(1), int64(2), object(6)
memory usage: 126.3+ KB
```

Usuń znak "%" w zmiennej *Cocoa_Percentage* - ustaw odpowiednio typ danej i wyświetl 3 pierwsze wiersze

```
data['Cocoa_Percentage'] = data['Cocoa_Percentage'].str.replace('%', '').astype(float)
```

```
data.head(5)
```

	Company	Specific_Bean_Origin_or_Bar_Name	REF	Review_Date	Cocoa_Percentage	Comp
0	A. Morin	Agua Grande	1876	2016	63.0	
1	A. Morin	Kpime	1676	2015	70.0	
2	A. Morin	Atsane	1676	2015	70.0	
3	A. Morin	Akata	1680	2015	70.0	
4	A. Morin	Quilla	1704	2015	70.0	

Next steps: View recommended plots

Wyfiltruj w zmiennych *Bean_Type* i *Broad_Bean_Origin* "niewidoczne braki danych" i zamień je na NA

```
data['Bean_Type'] = data['Bean_Type'].replace({'\xa0': pd.NA})
data['Broad_Bean_Origin'] = data['Broad_Bean_Origin'].replace({'\xa0': pd.NA})
```

```
data[['Bean_Type', 'Broad_Bean_Origin']].head()
```

	Bean_Type	Broad_Bean_Origin
0	<NA>	Sao Tome
1	<NA>	Togo
2	<NA>	Togo
3	<NA>	Togo
4	<NA>	Peru

Sprawdź czy puste komórki występują jednoszcześnie w zmiennej *Bean_Type* i *Broad_Bean_Origin*. Jesli tak, to policz ile jest takich wierszy i usuń całe wiersze dla tych przypadków

```
empty_rows = data[(data['Bean_Type'].isna()) & (data['Broad_Bean_Origin'].isna())]
empty_rows_count = len(empty_rows)
print("Liczba wierszy:", empty_rows_count)
```

```
Liczba wierszy: 51
```

```
data.dropna(subset=['Bean_Type', 'Broad_Bean_Origin'], inplace=True)
print(data.head())
```

```
     Company Specific_Bean_Origin_or_Bar_Name   REF  Review_Date  \
5   A. Morin                         Carenero  1315         2014
7   A. Morin                     Sur del Lago  1315         2014
8   A. Morin                   Puerto Cabello  1319         2014
11  A. Morin                       Madagascar  1011         2013
17  A. Morin                            Chuao  1015         2013

    Cocoa_Percentage Company_Location  Rating   Bean_Type Broad_Bean_Origin
5               70.0           France    2.75     Criollo         Venezuela
7               70.0           France    3.50     Criollo         Venezuela
8               70.0           France    3.75     Criollo         Venezuela
11              70.0           France    3.00     Criollo        Madagascar
17              70.0           France    4.00  Trinitario         Venezuela
```

```
empty_rows = data[(data['Bean_Type'].isna()) & (data['Broad_Bean_Origin'].isna())]
empty_rows_count = len(empty_rows)
print("Liczba wierszy:", empty_rows_count)
```

```
Liczba wierszy: 0
```

Wyświetl ponownie kształ zbioru.

```
data.head()
```

	Company	Specific_Bean_Origin_or_Bar_Name	REF	Review_Date	Cocoa_Percentage	Company_Location	Rating	Bean_Type	Broad_Bean_Orig
5	A. Morin	Carenero	1315	2014	70.0	France	2.75	Criollo	Venezu
7	A. Morin	Sur del Lago	1315	2014	70.0	France	3.50	Criollo	Venezu
8	A. Morin	Puerto Cabello	1319	2014	70.0	France	3.75	Criollo	Venezu
11	A. Morin	Madagascar	1011	2013	70.0	France	3.00	Criollo	Madagas
17	A. Morin	Chuao	1015	2013	70.0	France	4.00	Trinitario	Venezu

Next steps: View recommended plots

```
data.shape
```

```
(884, 9)
```

Dodaj do ramki danych nową kolumnę *Chocolate_Type*, która będzie zawierała informajce czy dana czekolada jest gorzka czy nie? W tym celu wykorzystaj zmienną *Cocoa_Percentage*.

```
data['Chocolate_Type'] = data['Cocoa_Percentage'].apply(lambda x: 'Gorzka' if x >= 75 else 'Słodka')
```

```
data.head(10)
```

	Company	Specific_Bean_Origin_or_Bar_Name	REF	Review_Date	Cocoa_Percentage	Company_Location	Rating	Bean_Type	Broad_Bean_Orig
5	A. Morin	Carenero	1315	2014	70.0	France	2.75	Criollo	Venezu
7	A. Morin	Sur del Lago	1315	2014	70.0	France	3.50	Criollo	Venezu
8	A. Morin	Puerto Cabello	1319	2014	70.0	France	3.75	Criollo	Venezu
11	A. Morin	Madagascar	1011	2013	70.0	France	3.00	Criollo	Madagas
17	A. Morin	Chuao	1015	2013	70.0	France	4.00	Trinitario	Venezu
24	Acalli	Tumbes, Norandino	1470	2015	70.0	U.S.A.	3.75	Criollo	P
25	Adi	Vanua Levu	705	2011	60.0	Fiji	2.75	Trinitario	
26	Adi	Vanua Levu, Toto-A	705	2011	80.0	Fiji	3.25	Trinitario	
27	Adi	Vanua Levu	705	2011	88.0	Fiji	3.50	Trinitario	
28	Adi	Vanua Levu, Ami-Ami-CA	705	2011	72.0	Fiji	3.50	Trinitario	

Next steps: View recommended plots

Dodaj zmienną kategoryczną *Rating_Grade* opisującą ocenę czekolady. Przyjmij następujące przedziały:

System oceny smaków kakao:

- 5 - Elite (Przekraczanie poza zwykłe granice)
- $<4;5)$ - Premium (Doskonały rozwój smaku, charakteru i stylu)
- $<3;4)$ - Satisfactory (3,0) do godnego pochwały (3,75) (dobrze wykonany, o specjalnych właściwościach)
- $<2;3)$ - Disappointing (Zadowalający, ale zawiera co najmniej jedną istotną wadę)
- $<1;2)$ - Unpleasant (przeważnie niesmaczny)

```
data['Rating_Grade']=pd.cut(data['Rating'], bins=[1,2,3,4,5,6], labels=['Unpleasant', 'Disappointing',' Satisfactory', 'Premium','Elite'])
```

```
data.head(10)
```

	Company	Specific_Bean_Origin_or_Bar_Name	REF	Review_Date	Cocoa_Percentage	Com
5	A. Morin	Carenero	1315	2014	70.0	
7	A. Morin	Sur del Lago	1315	2014	70.0	
8	A. Morin	Puerto Cabello	1319	2014	70.0	
11	A. Morin	Madagascar	1011	2013	70.0	
17	A. Morin	Chuao	1015	2013	70.0	
24	Acalli	Tumbes, Norandino	1470	2015	70.0	
25	Adi	Vanua Levu	705	2011	60.0	
26	Adi	Vanua Levu, Toto-A	705	2011	80.0	
27	Adi	Vanua Levu	705	2011	88.0	
28	Adi	Vanua Levu, Ami-Ami-CA	705	2011	72.0	

Next steps: View recommended plots

Dla zmiennych numerycznych oblicz statystyki opisowe

```
data.describe()
```

	REF	Review_Date	Cocoa_Percentage	Rating
count	884.000000	884.000000	884.000000	884.000000
mean	937.572398	2011.777149	71.895362	3.227658
std	566.936444	3.079376	6.124101	0.465389
min	5.000000	2006.000000	50.000000	1.000000
25%	411.500000	2009.000000	70.000000	3.000000
50%	951.000000	2012.000000	70.000000	3.250000
75%	1426.000000	2014.000000	75.000000	3.500000
max	1944.000000	2017.000000	100.000000	5.000000

Dla zmiennych kategorycznych zlicz unikalne wartości

```
grade = data['Rating_Grade'].value_counts()
print(grade)
```

```
Rating_Grade
 Satisfactory    502
Disappointing    363
Unpleasant        17
Premium            1
Elite              0
Name: count, dtype: int64
```

Policz średnią ocen dla czekolady mlecznej i gorzkiej (wykonaj pivot table lub tabelę grupującą)

```
srednia = data.groupby('Chocolate_Type')['Rating'].mean()
print(srednia)
```

```
Chocolate_Type
Gorzka    3.089844
Słodka    3.283838
Name: Rating, dtype: float64
```

Przygoyuj tablę raportującą która zliczy ilość unikalnych typów ocen () w poszczególnych latah dla czekolady mlecznej i gorzkiej

		Rating_Grade				
Review_Date	Chocolate_Type	**Disappointing**	**Elite**	**Premium**	...	...
2006	Dark					
	Milk					
2007	Dark					
	Milk					
...	...					

```
data.pivot_table(index=[data['Review_Date'], 'Chocolate_Type'], columns='Rating_Grade', values='Rating', aggfunc='count')
```

Review_Date	Chocolate_Type	Rating_Grade	Unpleasant	Disappointing	Satisfactory	Premium	Elite
2006	**Gorzka**	0	6	15	0	0
	Słodka	5	14	12	0	0
2007	**Gorzka**	3	10	5	0	0
	Słodka	0	12	26	1	0
2008	**Gorzka**	3	11	3	0	0
	Słodka	0	23	19	0	0
2009	**Gorzka**	2	16	5	0	0
	Słodka	0	27	28	0	0
2010	**Gorzka**	0	9	8	0	0
	Słodka	0	15	21	0	0
2011	**Gorzka**	0	10	10	0	0
	Słodka	1	19	40	0	0
2012	**Gorzka**	1	13	12	0	0
	Słodka	0	20	44	0	0
2013	**Gorzka**	1	9	15	0	0
	Słodka	0	19	34	0	0
2014	**Gorzka**	0	18	15	0	0
	Słodka	0	31	55	0	0
2015	**Gorzka**	0	18	17	0	0
	Słodka	0	29	64	0	0
2016	**Gorzka**	0	10	10	0	0
	Słodka	1	24	40	0	0
2017	**Gorzka**	0	0	0	0	0
	Słodka	0	0	4	0	0

- Zapisz plik do pdf (Plik -> Drukuj -> PDF) i umieść go na upelu.