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

```
df=pd.read_csv("https://github.com/HamoyeHQ/HDSC-Introduction-to-Python-for-machine-learning/files/7768140/FoodBalanceSheets_E_Africa_NOFLAG.
```

```
df
```

	Area Code	Area	Item Code	Item	Element Code	Element	Unit	
0	4	Algeria	2501	Population	511	Total Population - Both sexes	1000 persons	3
1	4	Algeria	2501	Population	5301	Domestic supply quantity	1000 tonnes	
2	4	Algeria	2901	Grand Total	664	Food supply (kcal/capita/day)	kcal/capita/day	
3	4	Algeria	2901	Grand Total	674	Protein supply quantity (g/capita/day)	g/capita/day	
4	4	Algeria	2901	Grand Total	684	Fat supply quantity (g/capita/day)	g/capita/day	
...	...	...	...	...	...	...	...	
60938	181	Zimbabwe	2899	Miscellaneous	5142	Food	1000 tonnes	

```
df.groupby('Element')['Y2017'].sum()

# Grouping by 'Element' and calculating the sum of 'Processing' in 2017
#result = df['Y2017'].groupby('Element')['Processing'].sum()

# Printing the result
#print(result['Processing'])
```

```
Element
Domestic supply quantity                  2088198.10
Export Quantity                            182338.80
Fat supply quantity (g/capita/day)          10253.84
Feed                                       223705.68
Food                                      1258888.28
Food supply (kcal/capita/day)              454681.00
Food supply quantity (kg/capita/yr)         48690.04
Import Quantity                            294559.09
Losses                                     160614.00
Other uses (non-food)                       91645.97
Processing                                 292836.00
Production                                2030056.89
Protein supply quantity (g/capita/day)      11842.45
Residuals                                   35500.00
Seed                                        24870.14
Stock Variation                             54316.91
Total Population - Both sexes             1112641.00
Tourist consumption                            91.00
Name: Y2017, dtype: float64
```

```
df_2014 = df['Y2014']
df_2014.isnull().sum()
```

```
1589
```

```
df_2014.shape
```

```
(60943,)
```

```
df_2017=df['Y2017']
df_2017.mean()
```

```
140.9177648602722
```

```
df_2017.std()
```

```
1671.8623590567995
```

```
selected_columns = df[['Y2017', 'Area']]
grouped_data= selected_columns.groupby('Area')['Y2017'].sum()
highest= grouped_data.idxmax()
highest
```

```
'Nigeria'
```

```
sorted_data = grouped_data.sort_values(ascending=True)
seventh= sorted_data.index[6]
seventh
```

```
'Guinea-Bissau'
```

```
# Grouping by 'Element' and calculating the sum of 'Stock Variation' for each year
grouped_data = df.groupby('Element').sum()

grouped_data
```

```
<ipython-input-73-f7249345c24f>:2: FutureWarning: The default value of numeric_onl
  grouped_data = df.groupby('Element').sum()
```

Element	Area Code	Item Code	Element Code	Y2014	Y2015	Y2016	
Domestic supply quantity	708993	14197445	28068795	1996716.35	2021493.55	2044842.70	208
Export Quantity	599910	11840553	26026133	150020.64	157614.47	151920.46	18
Fat supply quantity (g/capita/day)	675050	13535000	3435732	10225.56	10235.74	10102.77	1
Feed	176272	3538507	7282199	216927.89	225050.22	228958.65	22
Food	663295	13285035	25406622	1212332.49	1232361.10	1247022.17	125
Food supply (kcal/capita/day)	674057	13511060	3329296	454257.00	453383.00	451810.00	45
Food supply quantity (kg/capita/yr)	658446	13185401	3163725	49650.63	49345.13	48985.28	4
Import Quantity	688174	13795966	28834929	274144.48	267018.46	286582.78	29
Losses	274353	5424803	10292107	153223.00	155439.00	157787.00	16
Other uses (non-food)	235554	4729749	8926728	78718.13	66254.41	69563.68	9
Processing	271940	5350416	10313310	282923.00	287929.00	280631.00	29
Production	526751	10450053	21388191	1931287.75	1947019.39	1943537.15	203
Protein supply quantity (g/capita/day)	675050	13535000	3385502	11836.46	11833.95	11779.69	1

```
grouped_data = df.groupby('Item').sum()

print(grouped_data)
```

```
                      Area Code  Item Code  Element Code      Y2014      Y2015  \
Item
Alcohol, Non-Food         38447     752497       1515090    2403.00    2180.00
Alcoholic Beverages       75016    1649136       2171381  102410.11   98783.72
Animal Products           18060     397035         90990   11935.65   11811.73
Animal fats               85279    1882494       2567303  209460.54  200675.72
Apples and products       69280    1352989       1920418    9499.23   10559.15
...                         ...        ...           ...        ...        ...
Vegetables, Other         77779    1518715       2275409  155038.96  158104.08
Vegetal Products          18060     391905         90990  107145.19  107064.17
Wheat and products        87073    1639683       2654934  232670.13  234710.51
Wine                      66206    1319535       1818328    4497.36    4251.81
```

```
Yams                     50798     970905       1525745  200396.96  203151.78

                         Y2016      Y2017      Y2018
Item
Alcohol, Non-Food      2223.00    2348.00    2293.00
Alcoholic Beverages   96958.75   95581.06   97847.27
Animal Products       11661.69   11547.65   11578.61
Animal fats          183314.22  269617.53  269648.27
Apples and products    9853.26   10198.90    9640.51
...                        ...        ...        ...
Vegetables, Other    156218.90  157752.59  163987.21
Vegetal Products     106792.72  107655.20  107775.39
Wheat and products   239377.07  240047.62  242645.19
Wine                   3872.09    4178.02    4039.32
Yams                 220626.20  229174.59  221272.09

[119 rows x 8 columns]
<ipython-input-74-061f83835ee2>:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future v
  grouped_data = df.groupby('Item').sum()
```

```
filtered_data = df[(df['Area'] == 'Madagascar') & (df['Y2015'])]
filtered_data.groupby('Element').sum()
```

```
# Extracting the Protein supply quantity for Madagascar in 2015
#protein_supply_madagascar_2015 = filtered_data['Protein Supply Quantity'].sum()

# Printing the total Protein supply quantity in Madagascar in 2015
#print("Total Protein supply quantity in Madagascar in 2015:", protein_supply_madagascar_2015)
```

```
<ipython-input-75-0014ed6da114>:2: FutureWarning: The default value of numeric_onl
  filtered_data.groupby('Element').sum()
```

Element	Area Code	Item Code	Element Code	Y2014	Y2015	Y2016	Y2017	Y
Domestic supply quantity	11481	240991	471789	31577.76	31214.98	31730.08	31926.87	3259
Export Quantity	6321	134352	289639	425.48	494.75	535.06	566.81	53
Fat supply quantity (g/capita/day)	8772	185508	46512	88.71	91.85	95.29	101.01	9
Feed	1806	37888	77294	2177.76	2070.22	2117.40	2056.60	213
Food	10449	220084	416502	21404.42	21120.65	21433.61	21675.21	2211
Food supply (kcal/capita/day)	9159	193424	47144	7942.00	7685.00	7666.00	7765.00	774
Food supply quantity (kg/capita/yr)	11223	235799	56115	907.58	871.59	861.17	847.75	84
Import Quantity	5160	109447	224440	2008.68	1721.80	1890.46	3132.79	32(
Losses	5418	113563	215166	2252.00	2176.00	2215.00	2153.00	225
Other uses (non-food)	2322	49301	92772	4346.58	3957.12	4112.08	4170.06	422
Processing	1548	32155	61572	1262.00	1803.00	1770.00	1826.00	181
Production	10965	230006	468435	30337.51	29482.89	29889.65	29267.86	3021
Protein supply	8901	180220	46506	177.77	172.05	172.27	175.72	17

```
unique_countries = df['Area'].nunique()
unique_countries
```

```
49
```

```
correlation_data = df.groupby('Element Code').corr()

# Finding the year with the least correlation with 'Element Code'
correlation_data
```

```
<ipython-input-77-0ca4a6042639>:1: FutureWarning: The default value of numeric_only in D
  correlation_data = df.groupby('Element Code').corr()
```

		Area Code	Item Code	Y2014	Y2015	Y2016	Y2017	Y2018
Element Code								
511	**Area Code**	1.000000	NaN	0.149348	0.149792	0.150292	0.150840	0.151415
	Item Code	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Y2014	0.149348	NaN	1.000000	0.999988	0.999951	0.999885	0.999791
	Y2015	0.149792	NaN	0.999988	1.000000	0.999987	0.999947	0.999878
	Y2016	0.150292	NaN	0.999951	0.999987	1.000000	0.999986	0.999944
...	**...**	...	...	...	...	...	...	...
5911	**Y2014**	0.005901	0.056301	1.000000	0.976942	0.992694	0.941934	0.942024
	Y2015	0.007883	0.050526	0.976942	1.000000	0.965195	0.863665	0.863140
	Y2016	0.006203	0.062306	0.992694	0.965195	1.000000	0.963790	0.963378
	Y2017	0.005156	0.063080	0.941934	0.863665	0.963790	1.000000	0.999409

```
my_tuppy = (1,2,5,8)

my_tuppy[2] = 6
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-78-ce1543b3be47> in <cell line: 3>()
      1 my_tuppy = (1,2,5,8)
      2
----> 3 my_tuppy[2] = 6

TypeError: 'tuple' object does not support item assignment
```

SEARCH STACK OVERFLOW

```
lst = [[35, 'Portugal', 94], [33, 'Argentina', 93], [30 , 'Brazil', 92]]

col = ['Age','Nationality','Overall']


pd.DataFrame(lst, columns=col, index=[i for i in range(1,4)])
```

	Age	Nationality	Overall
1	35	Portugal	94
2	33	Argentina	93
3	30	Brazil	92

✓ 0s completed at 2:17 PM