Pandas case study

Titanic

we will check data from ship data set

In [ ]:
```
# import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
ship = sns.load_dataset("titanic")
ship
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True
...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True

891 rows × 15 columns

In [ ]:
```
ship.head()
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN

In [ ]:
```
# saving data fram einto csv file
ship.to_csv("ship.csv")
```

In [ ]:
```
# basic statistics / summary
ship.describe()
```

Out[ ]:

	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [ ]:
```
ship.head()
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN

In [ ]:
```
# droping few column and make new dataset
new_ship=ship.drop(["sibsp","parch"], axis=1)
new_ship.head()
```

Out[ ]:

	survived	pclass	sex	age	fare	embarked	class	who	adult_male	deck	embark_town
0	0	3	male	22.0	7.2500	S	Third	man	True	NaN	Southampton
1	1	1	female	38.0	71.2833	C	First	woman	False	C	Cherbourg
2	1	3	female	26.0	7.9250	S	Third	woman	False	NaN	Southampton
3	1	1	female	35.0	53.1000	S	First	woman	False	C	Southampton
4	0	3	male	35.0	8.0500	S	Third	man	True	NaN	Southampton

In [ ]:
```
ship.mean()
```

```
C:\Users\m s\AppData\Local\Temp\ipykernel_18268\2490984261.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  ship.mean()
```

Out[ ]:
```
survived       0.383838
pclass         2.308642
age           29.699118
sibsp          0.523008
parch          0.381594
fare          32.204208
adult_male     0.602694
alone          0.602694
dtype: float64
```

In [ ]:
```
ship.groupby(["survived","age"]).mean()
```

Out[ ]:

		pclass	sibsp	parch	fare	adult_male	alone
survived	**age**						
0	**1.0**	3.000000	4.500000	1.500000	43.293750	0.0	0.000000
	2.0	2.714286	2.714286	1.428571	44.439286	0.0	0.000000
	3.0	3.000000	3.000000	1.000000	21.075000	0.0	0.000000
	4.0	3.000000	3.666667	1.666667	29.433333	0.0	0.000000
	6.0	3.000000	4.000000	2.000000	31.275000	0.0	0.000000
...	**...**	...	...	...	...	...	...
1	**58.0**	1.000000	0.000000	0.333333	108.844433	0.0	0.666667
	60.0	1.000000	1.000000	0.500000	77.225000	0.5	0.000000
	62.0	1.500000	0.000000	0.000000	45.250000	0.5	1.000000
	63.0	2.000000	0.500000	0.000000	43.772900	0.0	0.500000
	80.0	1.000000	0.000000	0.000000	30.000000	1.0	1.000000

142 rows × 6 columns

In [ ]:
```
ship.groupby(["sex","class"]).mean()
```

Out[ ]:

		survived	pclass	age	sibsp	parch	fare	adult_male	alone
sex	**class**								
female	**First**	0.968085	1.0	34.611765	0.553191	0.457447	106.125798	0.000000	0.361702
	Second	0.921053	2.0	28.722973	0.486842	0.605263	21.970121	0.000000	0.421053
	Third	0.500000	3.0	21.750000	0.895833	0.798611	16.118810	0.000000	0.416667
male	**First**	0.368852	1.0	41.281386	0.311475	0.278689	67.226127	0.975410	0.614754
	Second	0.157407	2.0	30.740707	0.342593	0.222222	19.741782	0.916667	0.666667
	Third	0.135447	3.0	26.507589	0.498559	0.224784	12.661633	0.919308	0.760807

In [ ]:
```
ship.value_counts(["survived"])
```

Out[ ]:
```
survived
0           549
1           342
dtype: int64
```

In [ ]:
```
ship.groupby(["sex"]).mean()
```

Out[ ]:

sex	survived	pclass	age	sibsp	parch	fare	adult_male	alone
female	0.742038	2.159236	27.915709	0.694268	0.649682	44.479818	0.000000	0.401274
male	0.188908	2.389948	30.726645	0.429809	0.235702	25.523893	0.930676	0.712305

In [ ]:
```
ship[ship['age']<16].groupby(["sex", "class"]).mean()
```

Out[ ]:

sex	class	survived	pclass	age	sibsp	parch	fare	adult_male	alone
female	**First**	0.666667	1.0	10.333333	0.666667	1.666667	160.962500	0.0	0.000000
	Second	1.000000	2.0	6.600000	0.700000	1.300000	29.240000	0.0	0.000000
	Third	0.533333	3.0	7.100000	1.533333	1.100000	19.023753	0.0	0.166667
male	**First**	1.000000	1.0	5.306667	0.666667	2.000000	117.802767	0.0	0.000000
	Second	1.000000	2.0	2.258889	0.888889	1.222222	27.306022	0.0	0.000000
	Third	0.321429	3.0	6.515000	2.821429	1.321429	27.716371	0.0	0.035714

Malta Country

we will check from malta land data set

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
malta = pd.read_csv("malta_land.csv")
malta.head()
```

Out[ ]:

	Domain Code	Domain	Area Code	Area	Element Code	Element	Item Code	Item	Year Code	Year	Unit	Value	F
0	RL	Land Use	134	Malta	5110	Area	6610	Agricultural land	1961	1961	1000 ha	18.0	
1	RL	Land Use	134	Malta	5110	Area	6610	Agricultural land	1962	1962	1000 ha	17.0	
2	RL	Land Use	134	Malta	5110	Area	6610	Agricultural land	1963	1963	1000 ha	16.0	
3	RL	Land Use	134	Malta	5110	Area	6610	Agricultural land	1964	1964	1000 ha	15.0	
4	RL	Land Use	134	Malta	5110	Area	6610	Agricultural land	1965	1965	1000 ha	14.0	

In [ ]:
```
malta.describe()
```

Out[ ]:

	Area Code	Element Code	Item Code	Year Code	Year	Value
count	89.0	89.0	89.000000	89.000000	89.000000	89.000000
mean	134.0	5110.0	6630.224719	1994.887640	1994.887640	15.204270
std	0.0	0.0	28.523417	16.355037	16.355037	4.678391
min	134.0	5110.0	6610.000000	1961.000000	1961.000000	9.000000
25%	134.0	5110.0	6610.000000	1983.000000	1983.000000	10.400000
50%	134.0	5110.0	6610.000000	1997.000000	1997.000000	14.000000
75%	134.0	5110.0	6670.000000	2008.000000	2008.000000	21.160000
max	134.0	5110.0	6670.000000	2019.000000	2019.000000	22.650000

In [ ]:
```
# droping few column and make new dataset
new_malta=malta.drop(["Area","Element"], axis=1)
new_malta.head()
```

Out[ ]:

	Domain Code	Domain	Area Code	Element Code	Item Code	Item	Year Code	Year	Unit	Value	Flag	Fla Descriptic
0	RL	Land Use	134	5110	6610	Agricultural land	1961	1961	1000 ha	18.0	F	FAO estima
1	RL	Land Use	134	5110	6610	Agricultural land	1962	1962	1000 ha	17.0	Q	Official da reported c FA Questionnair .
2	RL	Land Use	134	5110	6610	Agricultural land	1963	1963	1000 ha	16.0	Q	Official da reported c FA Questionnair .
3	RL	Land Use	134	5110	6610	Agricultural land	1964	1964	1000 ha	15.0	Q	Official da reported c FA Questionnair .
4	RL	Land Use	134	5110	6610	Agricultural land	1965	1965	1000 ha	14.0	Q	Official da reported c FA Questionnair .

In [ ]:
```
malta.mean()
```

```
C:\Users\m s\AppData\Local\Temp\ipykernel_18268\3880934829.py:1: FutureWarning: Dropp
ing of nuisance columns in DataFrame reductions (with 'numeric_only=None') is depreca
ted; in a future version this will raise TypeError.  Select only valid columns before
calling the reduction.
  malta.mean()
```

Out[ ]:
```
Area Code        134.000000
Element Code    5110.000000
Item Code       6630.224719
Year Code       1994.887640
Year            1994.887640
Value             15.204270
dtype: float64
```

In [ ]:
```
malta.groupby(["Value", "Flag"]).mean().head()
```

Out[ ]:

		Area Code	Element Code	Item Code	Year Code	Year
Value	**Flag**					
9.0	**F**	134.0	5110.0	6610.0	2000.00	2000.00
	Q	134.0	5110.0	6610.0	1998.50	1998.50
9.2	**Fm**	134.0	5110.0	6610.0	2006.00	2006.00
9.3	**Q**	134.0	5110.0	6610.0	2007.25	2007.25
10.0	**F**	134.0	5110.0	6610.0	1997.00	1997.00

In [ ]:
```
malta.value_counts(['Year Code']).head()
```

Out[ ]:
```
Year Code
1990         2
2005         2
1993         2
1994         2
1995         2
dtype: int64
```

In [ ]:
```
malta[malta['Value']<14].groupby(["Year", "Unit"]).mean().head(6)
```

Out[ ]:

		Area Code	Element Code	Item Code	Year Code	Value
Year	**Unit**					
1975	**1000 ha**	134.0	5110.0	6610.0	1975.0	13.0
1976	**1000 ha**	134.0	5110.0	6610.0	1976.0	13.0
1980	**1000 ha**	134.0	5110.0	6610.0	1980.0	13.0
1981	**1000 ha**	134.0	5110.0	6610.0	1981.0	13.0
1982	**1000 ha**	134.0	5110.0	6610.0	1982.0	13.0
1983	**1000 ha**	134.0	5110.0	6610.0	1983.0	13.0

In [ ]: