

Analysing columns

Problem 1: Import MPG dataset and store as the pandas dataframe with name *mpg*

```
import pandas as pd
```

```
mpg =  
pd.read_csv("https://github.com/YBI-Foundation/Dataset/raw/main/MPG.csv")
```

```
mpg
```

	mpg	cylinders	displacement	horsepower	weight	
acceleration \						
0	18.0	8	307.0	130.0	3504	12.0
1	15.0	8	350.0	165.0	3693	11.5
2	18.0	8	318.0	150.0	3436	11.0
3	16.0	8	304.0	150.0	3433	12.0
4	17.0	8	302.0	140.0	3449	10.5
..
393	27.0	4	140.0	86.0	2790	15.6
394	44.0	4	97.0	52.0	2130	24.6
395	32.0	4	135.0	84.0	2295	11.6
396	28.0	4	120.0	79.0	2625	18.6
397	31.0	4	119.0	82.0	2720	19.4

	model_year	origin	name
0	70	usa	chevrolet chevelle malibu
1	70	usa	buick skylark 320
2	70	usa	plymouth satellite
3	70	usa	amc rebel sst
4	70	usa	ford torino
..
393	82	usa	ford mustang gl
394	82	europa	vw pickup
395	82	usa	dodge rampage
396	82	usa	ford ranger
397	82	usa	chevy s-10

[398 rows x 9 columns]

Problem 2: Copy MPG dataframe as car

```
car = mpg.copy()
```

car

	mpg	cylinders	displacement	horsepower	weight	
0	18.0	8	307.0	130.0	3504	12.0
1	15.0	8	350.0	165.0	3693	11.5
2	18.0	8	318.0	150.0	3436	11.0
3	16.0	8	304.0	150.0	3433	12.0
4	17.0	8	302.0	140.0	3449	10.5
..
393	27.0	4	140.0	86.0	2790	15.6
394	44.0	4	97.0	52.0	2130	24.6
395	32.0	4	135.0	84.0	2295	11.6
396	28.0	4	120.0	79.0	2625	18.6
397	31.0	4	119.0	82.0	2720	19.4

	model_year	origin	name
0	70	usa	chevrolet chevelle malibu
1	70	usa	buick skylark 320
2	70	usa	plymouth satellite
3	70	usa	amc rebel sst
4	70	usa	ford torino
..
393	82	usa	ford mustang gl
394	82	europe	vw pickup
395	82	usa	dodge rampage
396	82	usa	ford ranger
397	82	usa	chevy s-10

[398 rows x 9 columns]

Problem 3: Drop column name cylinders from original dataframe (mpg) and inspect what happened to the copy(car).

```
mpg = mpg.drop("cylinders", axis = 1)

mpg.columns

Index(['mpg', 'displacement', 'horsepower', 'weight', 'acceleration',
      'model_year', 'origin', 'name'],
      dtype='object')

car.columns

Index(['mpg', 'cylinders', 'displacement', 'horsepower', 'weight',
      'acceleration', 'model_year', 'origin', 'name'],
      dtype='object')
```

Note :No changes in cars dataframe

Problem 4: Analysing car dataframe

```
car.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype  
---  -
 0   mpg             398 non-null   float64
 1   cylinders        398 non-null   int64   
 2   displacement     398 non-null   float64
 3   horsepower       392 non-null   float64
 4   weight           398 non-null   int64   
 5   acceleration     398 non-null   float64
 6   model_year       398 non-null   int64   
 7   origin           398 non-null   object  
 8   name             398 non-null   object  
dtypes: float64(4), int64(3), object(2)
memory usage: 28.1+ KB

car.describe()

           mpg  cylinders  displacement  horsepower
weight \
count  398.000000  398.000000    398.000000    392.000000    398.000000

mean     23.514573    5.454774    193.425879    104.469388    2970.424623

std       7.815984    1.701004    104.269838    38.491160    846.841774

min       9.000000    3.000000    68.000000    46.000000   1613.000000
```

25%	17.500000	4.000000	104.250000	75.000000	2223.750000
50%	23.000000	4.000000	148.500000	93.500000	2803.500000
75%	29.000000	8.000000	262.000000	126.000000	3608.000000
max	46.600000	8.000000	455.000000	230.000000	5140.000000

	acceleration	model_year
count	398.000000	398.000000
mean	15.568090	76.010050
std	2.757689	3.697627
min	8.000000	70.000000
25%	13.825000	73.000000
50%	15.500000	76.000000
75%	17.175000	79.000000
max	24.800000	82.000000

Problem 5: Provide unique values in each columns cylinders and origin

```
car[["cylinders","origin"]].value_counts()
```

cylinders	origin	
8	usa	103
6	usa	74
4	usa	72
	japan	69
	europa	63
6	japan	6
3	japan	4
6	europa	4
5	europa	3

dtype: int64

Problem 6: Provide unique values of column origin

```
car[["origin"]].value_counts()
```

origin	
usa	249
japan	79
europa	70

dtype: int64

```
car["origin"].unique()
```

```
array(['usa', 'japan', 'europa'], dtype=object)
```

```
car["origin"].nunique()
```

3

Problem 7: Sort value car dataframe as per displacement column

```
car.displacement
```

```
0      307.0
1      350.0
2      318.0
3      304.0
4      302.0
```

```
...
393    140.0
394     97.0
395    135.0
396    120.0
397    119.0
```

```
Name: displacement, Length: 398, dtype: float64
```

Now sorting the values

```
car.sort_values("displacement")
```

	mpg	cylinders	displacement	horsepower	weight	
acceleration \						
117	29.0	4	68.0	49.0	1867	19.5
71	19.0	3	70.0	97.0	2330	13.5
111	18.0	3	70.0	90.0	2124	13.5
334	23.7	3	70.0	100.0	2420	12.5
131	32.0	4	71.0	65.0	1836	21.0
..
94	13.0	8	440.0	215.0	4735	11.0
6	14.0	8	454.0	220.0	4354	9.0
95	12.0	8	455.0	225.0	4951	11.0
8	14.0	8	455.0	225.0	4425	10.0
13	14.0	8	455.0	225.0	3086	10.0
	model_year	origin		name		
117	73	europe		fiat 128		
71	72	japan		mazda rx2 coupe		
111	73	japan		maxda rx3		

```

334      80  japan      mazda rx-7 gs
131      74  japan      toyota corolla 1200
..      ...  ...
94       73  usa  chrysler new yorker brougham
6        70  usa      chevrolet impala
95       73  usa      buick electra 225 custom
8        70  usa      pontiac catalina
13       70  usa      buick estate wagon (sw)

```

[398 rows x 9 columns]

Problem 8: Sort value of car dataframe as per displacement column in descending order.

```
car.sort_values("displacement", ascending = False)
```

```

      mpg  cylinders  displacement  horsepower  weight
acceleration \
8      14.0          8          455.0          225.0    4425      10.0

95     12.0          8          455.0          225.0    4951      11.0

13     14.0          8          455.0          225.0    3086      10.0

6      14.0          8          454.0          220.0    4354       9.0

7      14.0          8          440.0          215.0    4312       8.5

..     ...          ...          ...          ...     ...     ...

131    32.0          4           71.0           65.0    1836      21.0

111    18.0          3           70.0           90.0    2124      13.5

71     19.0          3           70.0           97.0    2330      13.5

334    23.7          3           70.0          100.0    2420      12.5

117    29.0          4           68.0           49.0    1867      19.5

```

```

      model_year  origin      name
8             70     usa  pontiac catalina
95            73     usa  buick electra 225 custom
13            70     usa  buick estate wagon (sw)
6             70     usa    chevrolet impala
7             70     usa  plymouth fury iii
..           ...     ...
131           74  japan  toyota corolla 1200
111           73  japan      mazda rx3

```

71	72	japan	mazda rx2 coupe
334	80	japan	mazda rx-7 gs
117	73	europe	fiat 128

[398 rows x 9 columns]

Problem 9: Sort value of car dataframe as per displacement and weight columns in descending order

```
car.sort_values(["displacement", "weight"], ascending = False)
```

	mpg	acceleration \	cylinders	displacement	horsepower	weight	
95	12.0		8	455.0	225.0	4951	11.0
8	14.0		8	455.0	225.0	4425	10.0
13	14.0		8	455.0	225.0	3086	10.0
6	14.0		8	454.0	220.0	4354	9.0
94	13.0		8	440.0	215.0	4735	11.0
..
53	31.0		4	71.0	65.0	1773	19.0
334	23.7		3	70.0	100.0	2420	12.5
71	19.0		3	70.0	97.0	2330	13.5
111	18.0		3	70.0	90.0	2124	13.5
117	29.0		4	68.0	49.0	1867	19.5

	model_year	origin	name
95	73	usa	buick electra 225 custom
8	70	usa	pontiac catalina
13	70	usa	buick estate wagon (sw)
6	70	usa	chevrolet impala
94	73	usa	chrysler new yorker brougham
..
53	71	japan	toyota corolla 1200
334	80	japan	mazda rx-7 gs
71	72	japan	mazda rx2 coupe
111	73	japan	maxda rx3
117	73	europe	fiat 128

[398 rows x 9 columns]

Problem 10: Summary statistics of all columns

```
car.describe(include = "all")
```

	mpg	cylinders	displacement	horsepower	weight
\count	398.000000	398.000000	398.000000	392.000000	398.000000
unique	NaN	NaN	NaN	NaN	NaN
top	NaN	NaN	NaN	NaN	NaN
freq	NaN	NaN	NaN	NaN	NaN
mean	23.514573	5.454774	193.425879	104.469388	2970.424623
std	7.815984	1.701004	104.269838	38.491160	846.841774
min	9.000000	3.000000	68.000000	46.000000	1613.000000
25%	17.500000	4.000000	104.250000	75.000000	2223.750000
50%	23.000000	4.000000	148.500000	93.500000	2803.500000
75%	29.000000	8.000000	262.000000	126.000000	3608.000000
max	46.600000	8.000000	455.000000	230.000000	5140.000000

	acceleration	model_year	origin	name
count	398.000000	398.000000	398	398
unique	NaN	NaN	3	305
top	NaN	NaN	usa	ford pinto
freq	NaN	NaN	249	6
mean	15.568090	76.010050	NaN	NaN
std	2.757689	3.697627	NaN	NaN
min	8.000000	70.000000	NaN	NaN
25%	13.825000	73.000000	NaN	NaN
50%	15.500000	76.000000	NaN	NaN
75%	17.175000	79.000000	NaN	NaN
max	24.800000	82.000000	NaN	NaN

Problem 11: Transpose of dataframe

```
car.T
```


	0	1	\
mpg	18.0	15.0	
cylinders	8	8	
displacement	307.0	350.0	
horsepower	130.0	165.0	
weight	3504	3693	
acceleration	12.0	11.5	
model_year	70	70	
origin	usa	usa	
name	chevrolet chevelle malibu	buick skylark 320	

	2	3	4	\
mpg	18.0	16.0	17.0	
cylinders	8	8	8	
displacement	318.0	304.0	302.0	
horsepower	150.0	150.0	140.0	
weight	3436	3433	3449	
acceleration	11.0	12.0	10.5	
model_year	70	70	70	
origin	usa	usa	usa	
name	plymouth satellite	amc rebel sst	ford torino	

	5	6	7	\
mpg	15.0	14.0	14.0	
cylinders	8	8	8	
displacement	429.0	454.0	440.0	
horsepower	198.0	220.0	215.0	
weight	4341	4354	4312	
acceleration	10.0	9.0	8.5	
model_year	70	70	70	
origin	usa	usa	usa	
name	ford galaxie 500	chevrolet impala	plymouth fury iii	

	8	9	...	\
mpg	14.0	15.0	...	
cylinders	8	8	...	
displacement	455.0	390.0	...	
horsepower	225.0	190.0	...	
weight	4425	3850	...	
acceleration	10.0	8.5	...	
model_year	70	70	...	
origin	usa	usa	...	
name	pontiac catalina	amc ambassador dpl	...	

	388	389
390 \		
mpg	26.0	22.0
32.0		
cylinders	4	6
4		

displacement	156.0	232.0	
144.0			
horsepower	92.0	112.0	
96.0			
weight	2585	2835	
2665			
acceleration	14.5	14.7	
13.9			
model_year	82	82	
82			
origin	usa	usa	
japan			
name	chrysler lebaron medallion	ford granada l	toyota
celica gt			
	391	392	393
394 \			
mpg	36.0	27.0	27.0
44.0			
cylinders	4	4	4
4			
displacement	135.0	151.0	140.0
97.0			
horsepower	84.0	90.0	86.0
52.0			
weight	2370	2950	2790
2130			
acceleration	13.0	17.3	15.6
24.6			
model_year	82	82	82
82			
origin	usa	usa	usa
europa			
name	dodge charger 2.2	chevrolet camaro	ford mustang gl
pickup			vw
	395	396	397
mpg	32.0	28.0	31.0
cylinders	4	4	4
displacement	135.0	120.0	119.0
horsepower	84.0	79.0	82.0
weight	2295	2625	2720
acceleration	11.6	18.6	19.4
model_year	82	82	82
origin	usa	usa	usa
name	dodge rampage	ford ranger	chevy s-10

[9 rows x 398 columns]