

24MDT0184_10_jan_EDA

January 10, 2025

1 10 Jan EDA Lab

```
[1]: import numpy as np
import pandas as pd
#loading the dataset
df = pd.read_csv(r'C:\Users\Batch1\Downloads\24MDT0184\Automobile_data.csv')
df
```

```
[1]:      symboling normalized-losses      make fuel-type aspiration \
0          3          ?  alfa-romero    gas    std
1          3          ?  alfa-romero    gas    std
2          1          ?  alfa-romero    gas    std
3          2        164      audi    gas    std
4          2        164      audi    gas    std
..      ...      ...      ...      ...      ...
200        -1          95    volvo    gas    std
201        -1          95    volvo    gas  turbo
202        -1          95    volvo    gas    std
203        -1          95    volvo  diesel  turbo
204        -1          95    volvo    gas  turbo

      num-of-doors  body-style drive-wheels engine-location  wheel-base ... \
0          two  convertible      rwd      front      88.6 ...
1          two  convertible      rwd      front      88.6 ...
2          two   hatchback      rwd      front      94.5 ...
3          four     sedan      fwd      front      99.8 ...
4          four     sedan      4wd      front      99.4 ...
..      ...      ...      ...      ...      ...
200        four     sedan      rwd      front      109.1 ...
201        four     sedan      rwd      front      109.1 ...
202        four     sedan      rwd      front      109.1 ...
203        four     sedan      rwd      front      109.1 ...
204        four     sedan      rwd      front      109.1 ...

      engine-size  fuel-system  bore  stroke  compression-ratio  horsepower \
0          130      mpfi  3.47    2.68          9.0          111
1          130      mpfi  3.47    2.68          9.0          111
```

2	152	mpfi	2.68	3.47	9.0	154
3	109	mpfi	3.19	3.4	10.0	102
4	136	mpfi	3.19	3.4	8.0	115
..
200	141	mpfi	3.78	3.15	9.5	114
201	141	mpfi	3.78	3.15	8.7	160
202	173	mpfi	3.58	2.87	8.8	134
203	145	idi	3.01	3.4	23.0	106
204	141	mpfi	3.78	3.15	9.5	114

	peak-rpm	city-mpg	highway-mpg	price
0	5000	21	27	13495
1	5000	21	27	16500
2	5000	19	26	16500
3	5500	24	30	13950
4	5500	18	22	17450
..
200	5400	23	28	16845
201	5300	19	25	19045
202	5500	18	23	21485
203	4800	26	27	22470
204	5400	19	25	22625

[205 rows x 26 columns]

```
[2]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
#   Column                Non-Null Count  Dtype
---  -
0   symboling              205 non-null    int64
1   normalized-losses      205 non-null    object
2   make                   205 non-null    object
3   fuel-type              205 non-null    object
4   aspiration              205 non-null    object
5   num-of-doors           205 non-null    object
6   body-style             205 non-null    object
7   drive-wheels           205 non-null    object
8   engine-location        205 non-null    object
9   wheel-base            205 non-null    float64
10  length                 205 non-null    float64
11  width                  205 non-null    float64
12  height                 205 non-null    float64
13  curb-weight            205 non-null    int64
14  engine-type            205 non-null    object
15  num-of-cylinders       205 non-null    object
```

```

16 engine-size      205 non-null    int64
17 fuel-system     205 non-null    object
18 bore            205 non-null    object
19 stroke          205 non-null    object
20 compression-ratio 205 non-null    float64
21 horsepower      205 non-null    object
22 peak-rpm        205 non-null    object
23 city-mpg        205 non-null    int64
24 highway-mpg     205 non-null    int64
25 price           205 non-null    object
dtypes: float64(5), int64(5), object(16)
memory usage: 41.8+ KB

```

```
[3]: df.replace('?', np.NaN, inplace=True)
```

```
[4]: df['normalized-losses']
```

```

[4]: 0      NaN
     1      NaN
     2      NaN
     3     164
     4     164
     ...
    200     95
    201     95
    202     95
    203     95
    204     95
Name: normalized-losses, Length: 205, dtype: object

```

```
[5]: df['normalized-losses'].isna().sum()
```

```
[5]: 41
```

```
[6]: df.dropna(inplace = True)
```

```
[7]: df['normalized-losses'] = df['normalized-losses'].astype('int')
```

```
[8]: df['price'] = df['price'].astype('int')
```

```
[9]: df['normalized-losses']
```

```

[9]: 3      164
     4      164
     6     158
     8     158
    10     192
     ...

```

```

200      95
201      95
202      95
203      95
204      95
Name: normalized-losses, Length: 159, dtype: int32

```

```
[10]: # group the dataframe according to bodystyle and print the key of groups
df.groupby('body-style').groups.keys()
```

```
[10]: dict_keys(['convertible', 'hardtop', 'hatchback', 'sedan', 'wagon'])
```

```
[11]: df['body-style'].value_counts()
```

```

[11]: body-style
sedan      79
hatchback  56
wagon      17
hardtop     5
convertible 2
Name: count, dtype: int64

```

1.1 Filtering dataframe where body type is sedan

```
[12]: df[df['body-style']=='sedan']
```

```

[12]:      symboling  normalized-losses  make fuel-type aspiration num-of-doors \
3          2          164  audi      gas      std      four
4          2          164  audi      gas      std      four
6          1          158  audi      gas      std      four
8          1          158  audi      gas     turbo      four
10         2          192  bmw      gas      std      two
..         ...          ...    ...    ...    ...    ...
200        -1           95  volvo      gas      std      four
201        -1           95  volvo      gas     turbo      four
202        -1           95  volvo      gas      std      four
203        -1           95  volvo  diesel     turbo      four
204        -1           95  volvo      gas     turbo      four

      body-style drive-wheels engine-location  wheel-base  ...  engine-size  \
3      sedan      fwd      front      99.8  ...      109
4      sedan      4wd      front      99.4  ...      136
6      sedan      fwd      front     105.8  ...      136
8      sedan      fwd      front     105.8  ...      131
10     sedan      rwd      front     101.2  ...      108
..         ...          ...    ...    ...    ...    ...
200     sedan      rwd      front     109.1  ...      141

```

201	sedan	rwd	front	109.1	...	141
202	sedan	rwd	front	109.1	...	173
203	sedan	rwd	front	109.1	...	145
204	sedan	rwd	front	109.1	...	141

	fuel-system	bore	stroke	compression-ratio	horsepower	peak-rpm	\
3	mpfi	3.19	3.4	10.0	102	5500	
4	mpfi	3.19	3.4	8.0	115	5500	
6	mpfi	3.19	3.4	8.5	110	5500	
8	mpfi	3.13	3.4	8.3	140	5500	
10	mpfi	3.5	2.8	8.8	101	5800	
..	
200	mpfi	3.78	3.15	9.5	114	5400	
201	mpfi	3.78	3.15	8.7	160	5300	
202	mpfi	3.58	2.87	8.8	134	5500	
203	idi	3.01	3.4	23.0	106	4800	
204	mpfi	3.78	3.15	9.5	114	5400	

	city-mpg	highway-mpg	price
3	24	30	13950
4	18	22	17450
6	19	25	17710
8	17	20	23875
10	23	29	16430
..
200	23	28	16845
201	19	25	19045
202	18	23	21485
203	26	27	22470
204	19	25	22625

[79 rows x 26 columns]

```
[13]: df[df['body-style']=='wagon']
```

```
[13]:
```

	symboling	normalized-losses	make	fuel-type	aspiration	\
28	-1	110	dodge	gas	std	
36	0	78	honda	gas	std	
68	-1	93	mercedes-benz	diesel	turbo	
93	1	103	nissan	gas	std	
97	1	103	nissan	gas	std	
102	0	108	nissan	gas	std	
123	-1	74	plymouth	gas	std	
146	0	89	subaru	gas	std	
147	0	89	subaru	gas	std	
148	0	85	subaru	gas	std	
149	0	85	subaru	gas	turbo	

153	0	77	toyota	gas	std
154	0	81	toyota	gas	std
155	0	91	toyota	gas	std
195	-1	74	volvo	gas	std
197	-1	74	volvo	gas	std
199	-1	74	volvo	gas	turbo

	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	\
28	four	wagon	fwd	front	103.3	...	
36	four	wagon	fwd	front	96.5	...	
68	four	wagon	rwd	front	110.0	...	
93	four	wagon	fwd	front	94.5	...	
97	four	wagon	fwd	front	94.5	...	
102	four	wagon	fwd	front	100.4	...	
123	four	wagon	fwd	front	103.3	...	
146	four	wagon	fwd	front	97.0	...	
147	four	wagon	fwd	front	97.0	...	
148	four	wagon	4wd	front	96.9	...	
149	four	wagon	4wd	front	96.9	...	
153	four	wagon	fwd	front	95.7	...	
154	four	wagon	4wd	front	95.7	...	
155	four	wagon	4wd	front	95.7	...	
195	four	wagon	rwd	front	104.3	...	
197	four	wagon	rwd	front	104.3	...	
199	four	wagon	rwd	front	104.3	...	

	engine-size	fuel-system	bore	stroke	compression-ratio	horsepower	\
28	122	2bbl	3.34	3.46	8.5	88	
36	92	1bbl	2.92	3.41	9.2	76	
68	183	idi	3.58	3.64	21.5	123	
93	97	2bbl	3.15	3.29	9.4	69	
97	97	2bbl	3.15	3.29	9.4	69	
102	181	mpfi	3.43	3.27	9.0	152	
123	122	2bbl	3.35	3.46	8.5	88	
146	108	2bbl	3.62	2.64	9.0	82	
147	108	mpfi	3.62	2.64	9.0	94	
148	108	2bbl	3.62	2.64	9.0	82	
149	108	mpfi	3.62	2.64	7.7	111	
153	92	2bbl	3.05	3.03	9.0	62	
154	92	2bbl	3.05	3.03	9.0	62	
155	92	2bbl	3.05	3.03	9.0	62	
195	141	mpfi	3.78	3.15	9.5	114	
197	141	mpfi	3.78	3.15	9.5	114	
199	130	mpfi	3.62	3.15	7.5	162	

	peak-rpm	city-mpg	highway-mpg	price
28	5000	24	30	8921

36	6000	30	34	7295
68	4350	22	25	28248
93	5200	31	37	7349
97	5200	31	37	7999
102	5200	17	22	14399
123	5000	24	30	8921
146	4800	28	32	7463
147	5200	25	31	10198
148	4800	23	29	8013
149	4800	23	23	11694
153	4800	31	37	6918
154	4800	27	32	7898
155	4800	27	32	8778
195	5400	23	28	13415
197	5400	24	28	16515
199	5100	17	22	18950

[17 rows x 26 columns]

1.2 filtering the records of convertible body type

```
[14]: style = df.groupby('body-style')
style.get_group("convertible")
```

```
[14]:      symboling  normalized-losses      make fuel-type aspiration \
72          3          142  mercedes-benz      gas      std
172         2          134      toyota      gas      std

      num-of-doors  body-style drive-wheels engine-location  wheel-base  ... \
72          two  convertible          rwd          front      96.6  ...
172         two  convertible          rwd          front      98.4  ...

      engine-size  fuel-system  bore  stroke  compression-ratio  horsepower  \
72          234          mpfi  3.46    3.1              8.3          155
172          146          mpfi  3.62    3.5              9.3          116

      peak-rpm  city-mpg  highway-mpg  price
72          4750      16          18  35056
172          4800      24          30  17669
```

[2 rows x 26 columns]

```
[15]: style.first()
```

```
[15]:      symboling  normalized-losses      make fuel-type aspiration \
body-style
convertible          3          142  mercedes-benz      gas      std
```

hardtop	0	93	mercedes-benz	diesel	turbo
hatchback	2	121	chevrolet	gas	std
sedan	2	164	audi	gas	std
wagon	-1	110	dodge	gas	std

	num-of-doors	drive-wheels	engine-location	wheel-base	length	\
body-style						
convertible	two	rwd	front	96.6	180.3	
hardtop	two	rwd	front	106.7	187.5	
hatchback	two	fwd	front	88.4	141.1	
sedan	four	fwd	front	99.8	176.6	
wagon	four	fwd	front	103.3	174.6	

	...	engine-size	fuel-system	bore	stroke	compression-ratio	\
body-style	...						
convertible	...	234	mpfi	3.46	3.1	8.3	
hardtop	...	183	idi	3.58	3.64	21.5	
hatchback	...	61	2bbl	2.91	3.03	9.5	
sedan	...	109	mpfi	3.19	3.4	10.0	
wagon	...	122	2bbl	3.34	3.46	8.5	

	horsepower	peak-rpm	city-mpg	highway-mpg	price
body-style					
convertible	155	4750	16	18	35056
hardtop	123	4350	22	25	28176
hatchback	48	5100	47	53	5151
sedan	102	5500	24	30	13950
wagon	88	5000	24	30	8921

[5 rows x 25 columns]

1.3 grouping by multiple features

```
[16]: double_grouping = df.groupby(['body-style', 'drive-wheels'])
```

```
[17]: double_grouping.first()
```

```
[17]:
```

		symboling	normalized-losses	make	\
body-style	drive-wheels				
convertible	rwd	3	142	mercedes-benz	
hardtop	fwd	2	168	nissan	
	rwd	0	93	mercedes-benz	
hatchback	4wd	2	83	subaru	
	fwd	2	121	chevrolet	
	rwd	3	194	nissan	
sedan	4wd	2	164	audi	
	fwd	2	164	audi	

	rwd	2	192	bmw
wagon	4wd	0	85	subaru
	fwd	-1	110	dodge
	rwd	-1	93	mercedes-benz

		fuel-type	aspiration	num-of-doors	engine-location	\
body-style	drive-wheels					
convertible	rwd	gas	std	two	front	
hardtop	fwd	gas	std	two	front	
	rwd	diesel	turbo	two	front	
hatchback	4wd	gas	std	two	front	
	fwd	gas	std	two	front	
	rwd	gas	std	two	front	
sedan	4wd	gas	std	four	front	
	fwd	gas	std	four	front	
	rwd	gas	std	two	front	
wagon	4wd	gas	std	four	front	
	fwd	gas	std	four	front	
	rwd	diesel	turbo	four	front	

		wheel-base	length	width	...	engine-size	\
body-style	drive-wheels				...		
convertible	rwd	96.6	180.3	70.5	...	234	
hardtop	fwd	95.1	162.4	63.8	...	97	
	rwd	106.7	187.5	70.3	...	183	
hatchback	4wd	93.3	157.3	63.8	...	108	
	fwd	88.4	141.1	60.3	...	61	
	rwd	91.3	170.7	67.9	...	181	
sedan	4wd	99.4	176.6	66.4	...	136	
	fwd	99.8	176.6	66.2	...	109	
	rwd	101.2	176.8	64.8	...	108	
wagon	4wd	96.9	173.6	65.4	...	108	
	fwd	103.3	174.6	64.6	...	122	
	rwd	110.0	190.9	70.3	...	183	

		fuel-system	bore	stroke	compression-ratio	\
body-style	drive-wheels					
convertible	rwd	mpfi	3.46	3.1	8.3	
hardtop	fwd	2bbl	3.15	3.29	9.4	
	rwd	idi	3.58	3.64	21.5	
hatchback	4wd	2bbl	3.62	2.64	8.7	
	fwd	2bbl	2.91	3.03	9.5	
	rwd	mpfi	3.43	3.27	9.0	
sedan	4wd	mpfi	3.19	3.4	8.0	
	fwd	mpfi	3.19	3.4	10.0	
	rwd	mpfi	3.5	2.8	8.8	
wagon	4wd	2bbl	3.62	2.64	9.0	

	fwd	2bbl	3.34	3.46		8.5
	rwd	idi	3.58	3.64		21.5

		horsepower	peak-rpm	city-mpg	highway-mpg	price
body-style	drive-wheels					
convertible	rwd	155	4750	16	18	35056
hardtop	fwd	69	5200	31	37	8249
	rwd	123	4350	22	25	28176
hatchback	4wd	73	4400	26	31	7603
	fwd	48	5100	47	53	5151
	rwd	160	5200	19	25	17199
sedan	4wd	115	5500	18	22	17450
	fwd	102	5500	24	30	13950
	rwd	101	5800	23	29	16430
wagon	4wd	82	4800	23	29	8013
	fwd	88	5000	24	30	8921
	rwd	123	4350	22	25	28248

[12 rows x 24 columns]

```
[18]: style[['normalized-losses']].max()
```

```
[18]:          normalized-losses
body-style
convertible          142
hardtop             168
hatchback           256
sedan                192
wagon               110
```

```
[19]: style['normalized-losses'].min()
```

```
[19]: body-style
convertible    134
hardtop        93
hatchback      65
sedan          65
wagon          74
Name: normalized-losses, dtype: int32
```

```
[20]: df['normalized-losses'].max()
```

```
[20]: 256
```

```
[21]: df['normalized-losses'].min()
```

```
[21]: 65
```

```
[22]: style.mean(numeric_only=True)
```

```
[22]:
```

	symboling	normalized-losses	wheel-base	length	width \
body-style					
convertible	2.500000	138.000000	97.500000	178.250000	68.050000
hardtop	1.600000	132.600000	99.400000	175.700000	66.180000
hatchback	1.428571	130.803571	95.276786	164.996429	64.889286
sedan	0.354430	120.354430	100.078481	176.545570	66.067089
wagon	-0.235294	87.529412	99.429412	175.994118	65.382353

	height	curb-weight	engine-size	compression-ratio \
body-style				
convertible	51.900000	3330.000000	190.000000	8.800000
hardtop	52.840000	2651.600000	143.600000	11.760000
hatchback	52.442857	2247.928571	111.017857	9.186071
sedan	54.429114	2539.113924	121.873418	10.900000
wagon	56.782353	2642.882353	118.470588	9.629412

	city-mpg	highway-mpg	price
body-style			
convertible	20.000000	24.000000	26362.500000
hardtop	25.000000	30.400000	13142.400000
hatchback	28.232143	34.000000	9220.160714
sedan	25.873418	31.493671	12558.620253
wagon	25.117647	29.941176	11351.411765

```
[23]: style['price'].sum()
```

```
[23]: body-style
convertible    52725
hardtop        65712
hatchback     516329
sedan          992131
wagon         192974
Name: price, dtype: int32
```

1.4 Titanic dataset

```
[25]: import seaborn as sns
      ## loading the dataset
      titanic = pd.read_csv(r'C:\Users\Batch1\Downloads\24MDT0184\titanic.csv')
      titanic
```

```
[25]:
```

	Unnamed: 0	survived	pclass	sex	age	sibsp	parch	fare \
0	0	0	3	male	22.0	1	0	7.2500
1	1	1	1	female	38.0	1	0	71.2833
2	2	1	3	female	26.0	0	0	7.9250

3	3	1	1	female	35.0	1	0	53.1000
4	4	0	3	male	35.0	0	0	8.0500
..
886	886	0	2	male	27.0	0	0	13.0000
887	887	1	1	female	19.0	0	0	30.0000
888	888	0	3	female	NaN	1	2	23.4500
889	889	1	1	male	26.0	0	0	30.0000
890	890	0	3	male	32.0	0	0	7.7500

	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	S	Third	man	True	NaN	Southampton	no	False
1	C	First	woman	False	C	Cherbourg	yes	False
2	S	Third	woman	False	NaN	Southampton	yes	True
3	S	First	woman	False	C	Southampton	yes	False
4	S	Third	man	True	NaN	Southampton	no	True
..
886	S	Second	man	True	NaN	Southampton	no	True
887	S	First	woman	False	B	Southampton	yes	True
888	S	Third	woman	False	NaN	Southampton	no	False
889	C	First	man	True	C	Cherbourg	yes	True
890	Q	Third	man	True	NaN	Queenstown	no	True

[891 rows x 16 columns]

```
[26]: titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 16 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Unnamed: 0      891 non-null   int64
1   survived        891 non-null   int64
2   pclass          891 non-null   int64
3   sex             891 non-null   object
4   age             714 non-null   float64
5   sibsp           891 non-null   int64
6   parch           891 non-null   int64
7   fare            891 non-null   float64
8   embarked        889 non-null   object
9   class           891 non-null   object
10  who             891 non-null   object
11  adult_male      891 non-null   bool
12  deck            203 non-null   object
13  embark_town     889 non-null   object
14  alive           891 non-null   object
15  alone           891 non-null   bool
dtypes: bool(2), float64(2), int64(5), object(7)
```

memory usage: 99.3+ KB

```
[27]: titanic.groupby('embark_town').groups.keys()
```

```
[27]: dict_keys(['Cherbourg', 'Queenstown', 'Southampton'])
```

```
[34]: style = titanic.groupby('embark_town')
style.first()
```

```
[34]:
```

	Unnamed: 0	survived	pclass	sex	age	sibsp	parch	\
embark_town								
Cherbourg	1	1	1	female	38.0	1	0	
Queenstown	5	0	3	male	2.0	0	0	
Southampton	0	0	3	male	22.0	1	0	

	fare	embarked	class	who	adult_male	deck	alive	alone
embark_town								
Cherbourg	71.2833	C	First	woman	False	C	yes	False
Queenstown	8.4583	Q	Third	man	True	C	no	True
Southampton	7.2500	S	Third	man	True	C	no	False

```
[35]: style.get_group("Cherbourg")
```

```
[35]:
```

	Unnamed: 0	survived	pclass	sex	age	sibsp	parch	fare	\
1	1	1	1	female	38.0	1	0	71.2833	
9	9	1	2	female	14.0	1	0	30.0708	
19	19	1	3	female	NaN	0	0	7.2250	
26	26	0	3	male	NaN	0	0	7.2250	
30	30	0	1	male	40.0	0	0	27.7208	
..	
866	866	1	2	female	27.0	1	0	13.8583	
874	874	1	2	female	28.0	1	0	24.0000	
875	875	1	3	female	15.0	0	0	7.2250	
879	879	1	1	female	56.0	0	1	83.1583	
889	889	1	1	male	26.0	0	0	30.0000	

	embarked	class	who	adult_male	deck	embark_town	alive	alone
1	C	First	woman	False	C	Cherbourg	yes	False
9	C	Second	child	False	NaN	Cherbourg	yes	False
19	C	Third	woman	False	NaN	Cherbourg	yes	True
26	C	Third	man	True	NaN	Cherbourg	no	True
30	C	First	man	True	NaN	Cherbourg	no	True
..
866	C	Second	woman	False	NaN	Cherbourg	yes	False
874	C	Second	woman	False	NaN	Cherbourg	yes	False
875	C	Third	child	False	NaN	Cherbourg	yes	True
879	C	First	woman	False	C	Cherbourg	yes	False
889	C	First	man	True	C	Cherbourg	yes	True

[168 rows x 16 columns]

```
[36]: style['age'].max()
```

```
[36]: embark_town
Cherbourg      71.0
Queenstown     70.5
Southampton    80.0
Name: age, dtype: float64
```

```
[37]: style['age'].min()
```

```
[37]: embark_town
Cherbourg      0.42
Queenstown     2.00
Southampton    0.67
Name: age, dtype: float64
```

```
[38]: style['age'].mean()
```

```
[38]: embark_town
Cherbourg      30.814769
Queenstown     28.089286
Southampton    29.445397
Name: age, dtype: float64
```

```
[40]: style.mean(numeric_only=True)
```

```
[40]:      Unnamed: 0  survived  pclass      age  sibsp  parch  \
embark_town
Cherbourg    444.357143  0.553571  1.886905  30.814769  0.386905  0.363095
Queenstown   416.896104  0.389610  2.909091  28.089286  0.428571  0.168831
Southampton  448.527950  0.336957  2.350932  29.445397  0.571429  0.413043

      fare  adult_male  alone
embark_town
Cherbourg    59.954144  0.535714  0.505952
Queenstown   13.276030  0.480519  0.740260
Southampton   27.079812  0.636646  0.610248
```

```
[41]: style['fare'].sum()
```

```
[41]: embark_town
Cherbourg      10072.2962
Queenstown      1022.2543
Southampton     17439.3988
Name: fare, dtype: float64
```

```
[47]: double_grouping = titanic.groupby(['embark_town', 'sex'])
double_grouping.first()
```

```
[47]:
```

		Unnamed: 0	survived	pclass	age	sibsp	parch	fare	\
embark_town	sex								
Cherbourg	female	1	1	1	38.0	1	0	71.2833	
	male	26	0	3	40.0	0	0	7.2250	
Queenstown	female	22	1	3	15.0	0	0	8.0292	
	male	5	0	3	2.0	0	0	8.4583	
Southampton	female	2	1	3	26.0	0	0	7.9250	
	male	0	0	3	22.0	1	0	7.2500	

		embarked	class	who	adult_male	deck	alive	alone
embark_town	sex							
Cherbourg	female	C	First	woman	False	C	yes	False
	male	C	Third	man	True	B	no	True
Queenstown	female	Q	Third	child	False	E	yes	True
	male	Q	Third	man	True	C	no	True
Southampton	female	S	Third	woman	False	C	yes	True
	male	S	Third	man	True	E	no	False

2 Iris dataset

```
[43]: iris = pd.read_csv(r'C:\Users\Batch1\Downloads\24MDT0184\iris.csv')
iris
```

```
[43]:
```

	Unnamed: 0	sepal_length	sepal_width	petal_length	petal_width	\
0	0	5.1	3.5	1.4	0.2	
1	1	4.9	3.0	1.4	0.2	
2	2	4.7	3.2	1.3	0.2	
3	3	4.6	3.1	1.5	0.2	
4	4	5.0	3.6	1.4	0.2	
..	
145	145	6.7	3.0	5.2	2.3	
146	146	6.3	2.5	5.0	1.9	
147	147	6.5	3.0	5.2	2.0	
148	148	6.2	3.4	5.4	2.3	
149	149	5.9	3.0	5.1	1.8	

	species
0	setosa
1	setosa
2	setosa
3	setosa
4	setosa
..	...
145	virginica

```

146 virginica
147 virginica
148 virginica
149 virginica

```

```
[150 rows x 6 columns]
```

```
[44]: style = iris.groupby('species')
      style.first()
```

```
[44]:
```

	Unnamed: 0	sepal_length	sepal_width	petal_length	petal_width
species					
setosa	0	5.1	3.5	1.4	0.2
versicolor	50	7.0	3.2	4.7	1.4
virginica	100	6.3	3.3	6.0	2.5

```
[49]: style.get_group("virginica")
```

```
[49]:
```

	Unnamed: 0	sepal_length	sepal_width	petal_length	petal_width	\
100	100	6.3	3.3	6.0	2.5	
101	101	5.8	2.7	5.1	1.9	
102	102	7.1	3.0	5.9	2.1	
103	103	6.3	2.9	5.6	1.8	
104	104	6.5	3.0	5.8	2.2	
105	105	7.6	3.0	6.6	2.1	
106	106	4.9	2.5	4.5	1.7	
107	107	7.3	2.9	6.3	1.8	
108	108	6.7	2.5	5.8	1.8	
109	109	7.2	3.6	6.1	2.5	
110	110	6.5	3.2	5.1	2.0	
111	111	6.4	2.7	5.3	1.9	
112	112	6.8	3.0	5.5	2.1	
113	113	5.7	2.5	5.0	2.0	
114	114	5.8	2.8	5.1	2.4	
115	115	6.4	3.2	5.3	2.3	
116	116	6.5	3.0	5.5	1.8	
117	117	7.7	3.8	6.7	2.2	
118	118	7.7	2.6	6.9	2.3	
119	119	6.0	2.2	5.0	1.5	
120	120	6.9	3.2	5.7	2.3	
121	121	5.6	2.8	4.9	2.0	
122	122	7.7	2.8	6.7	2.0	
123	123	6.3	2.7	4.9	1.8	
124	124	6.7	3.3	5.7	2.1	
125	125	7.2	3.2	6.0	1.8	
126	126	6.2	2.8	4.8	1.8	
127	127	6.1	3.0	4.9	1.8	

128	128	6.4	2.8	5.6	2.1
129	129	7.2	3.0	5.8	1.6
130	130	7.4	2.8	6.1	1.9
131	131	7.9	3.8	6.4	2.0
132	132	6.4	2.8	5.6	2.2
133	133	6.3	2.8	5.1	1.5
134	134	6.1	2.6	5.6	1.4
135	135	7.7	3.0	6.1	2.3
136	136	6.3	3.4	5.6	2.4
137	137	6.4	3.1	5.5	1.8
138	138	6.0	3.0	4.8	1.8
139	139	6.9	3.1	5.4	2.1
140	140	6.7	3.1	5.6	2.4
141	141	6.9	3.1	5.1	2.3
142	142	5.8	2.7	5.1	1.9
143	143	6.8	3.2	5.9	2.3
144	144	6.7	3.3	5.7	2.5
145	145	6.7	3.0	5.2	2.3
146	146	6.3	2.5	5.0	1.9
147	147	6.5	3.0	5.2	2.0
148	148	6.2	3.4	5.4	2.3
149	149	5.9	3.0	5.1	1.8

	species
100	virginica
101	virginica
102	virginica
103	virginica
104	virginica
105	virginica
106	virginica
107	virginica
108	virginica
109	virginica
110	virginica
111	virginica
112	virginica
113	virginica
114	virginica
115	virginica
116	virginica
117	virginica
118	virginica
119	virginica
120	virginica
121	virginica
122	virginica

```
123 virginica
124 virginica
125 virginica
126 virginica
127 virginica
128 virginica
129 virginica
130 virginica
131 virginica
132 virginica
133 virginica
134 virginica
135 virginica
136 virginica
137 virginica
138 virginica
139 virginica
140 virginica
141 virginica
142 virginica
143 virginica
144 virginica
145 virginica
146 virginica
147 virginica
148 virginica
149 virginica
```

```
[50]: style['sepal_length'].max()
```

```
[50]: species
      setosa      5.8
      versicolor  7.0
      virginica   7.9
      Name: sepal_length, dtype: float64
```

```
[51]: style['sepal_length'].min()
```

```
[51]: species
      setosa      4.3
      versicolor  4.9
      virginica   4.9
      Name: sepal_length, dtype: float64
```

```
[53]: style['sepal_length'].max()
```

```
[53]: species
      setosa      5.8
      versicolor  7.0
      virginica   7.9
      Name: sepal_length, dtype: float64
```

```
[52]: style['sepal_length'].min()
```

```
[52]: species
      setosa      4.3
      versicolor  4.9
      virginica   4.9
      Name: sepal_length, dtype: float64
```

```
[54]: style['sepal_width'].max()
```

```
[54]: species
      setosa      4.4
      versicolor  3.4
      virginica   3.8
      Name: sepal_width, dtype: float64
```

```
[55]: style['sepal_width'].min()
```

```
[55]: species
      setosa      2.3
      versicolor  2.0
      virginica   2.2
      Name: sepal_width, dtype: float64
```

```
[56]: style['petal_length'].max()
```

```
[56]: species
      setosa      1.9
      versicolor  5.1
      virginica   6.9
      Name: petal_length, dtype: float64
```

```
[57]: style['petal_length'].min()
```

```
[57]: species
      setosa      1.0
      versicolor  3.0
      virginica   4.5
      Name: petal_length, dtype: float64
```

```
[58]: style['petal_width'].max()
```

```
[58]: species
      setosa      0.6
      versicolor  1.8
      virginica   2.5
      Name: petal_width, dtype: float64
```

```
[59]: style['petal_width'].min()
```

```
[59]: species
      setosa      0.1
      versicolor  1.0
      virginica   1.4
      Name: petal_width, dtype: float64
```

```
[60]: style.mean(numeric_only=True)
```

```
[60]:      Unnamed: 0  sepal_length  sepal_width  petal_length  petal_width
      species
      setosa      24.5          5.006          3.428          1.462          0.246
      versicolor  74.5          5.936          2.770          4.260          1.326
      virginica  124.5          6.588          2.974          5.552          2.026
```

```
[ ]:
```