Pandas Lab Exercise (kaggle automobile dataset) - Solutions

July 6, 2024

0.1 Pandas Lab Exercise (Kaggle Automobile Dataset)

We shall now test your skills in using Pandas package. We will be using the automobiles Dataset from Kaggle.

Answer each question asked below wrt the automobiles dataset. Load pandas as pd and upload the Automobile.csv file as auto

[1]:
```
import pandas as pd
```

[2]:
```
auto = pd.read_csv('Automobile.csv')
```

Check the head of the DataFrame.

[3]:
```
auto.head()
```

[3]:

	symboling	normalized_losses	make	fuel_type	aspiration	…
0	3	168	alfa-romero	gas	std	…
1	3	168	alfa-romero	gas	std	…
2	1	168	alfa-romero	gas	std	…
3	2	164	audi	gas	std	…
4	2	164	audi	gas	std	…

	number_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	…

	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	3.40	10.0	102
4	136	mpfi	3.19	3.40	8.0	115

	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

[5 rows x 26 columns]
```

** How many rows and columns are there? **

[4]:
```
auto.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 201 entries, 0 to 200
Data columns (total 26 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   symboling            201 non-null    int64
 1   normalized_losses    201 non-null    int64
 2   make                 201 non-null    object
 3   fuel_type            201 non-null    object
 4   aspiration           201 non-null    object
 5   number_of_doors      201 non-null    object
 6   body_style           201 non-null    object
 7   drive_wheels         201 non-null    object
 8   engine_location      201 non-null    object
 9   wheel_base           201 non-null    float64
 10  length               201 non-null    float64
 11  width                201 non-null    float64
 12  height               201 non-null    float64
 13  curb_weight          201 non-null    int64
 14  engine_type          201 non-null    object
 15  number_of_cylinders  201 non-null    object
 16  engine_size          201 non-null    int64
 17  fuel_system          201 non-null    object
 18  bore                 201 non-null    float64
 19  stroke               201 non-null    float64
 20  compression_ratio    201 non-null    float64
 21  horsepower           201 non-null    int64
 22  peak_rpm             201 non-null    int64
 23  city_mpg             201 non-null    int64
 24  highway_mpg          201 non-null    int64
 25  price                201 non-null    int64
dtypes: float64(7), int64(9), object(10)
memory usage: 41.0+ KB
```

** What is the average Price of all cars in the dataset? **

[5]:
```
auto['price'].mean()
```

[5]:
```
13207.129353233831
```

** Which is the cheapest make and costliest make of car in the lot? **

[6]:
```
auto[auto['price']==auto['price'].max()]
```

[6]:
```
    symboling  normalized_losses          make fuel_type aspiration  \
71          1                140  mercedes-benz       gas        std

   number_of_doors body_style drive_wheels engine_location  wheel_base  …  \
71             two    hardtop          rwd           front       112.0  …

    engine_size  fuel_system  bore  stroke compression_ratio horsepower  \
71          304         mpfi   3.8    3.35               8.0        184

    peak_rpm city_mpg  highway_mpg  price
71      4500       14           16  45400

[1 rows x 26 columns]
```

[7]:
```
auto[auto['price']==auto['price'].min()]
```

[7]:
```
     symboling  normalized_losses    make fuel_type aspiration  \
134          2                 83  subaru       gas        std

    number_of_doors body_style drive_wheels engine_location  wheel_base  …  \
134             two  hatchback          fwd           front        93.7  …

     engine_size  fuel_system  bore  stroke compression_ratio horsepower  \
134           97         2bbl  3.62    2.36               9.0         69

     peak_rpm city_mpg  highway_mpg  price
134      4900       31           36   5118

[1 rows x 26 columns]
```

** How many cars have horsepower greater than 100? **

[8]:
```
auto[auto['horsepower']>100].count()
```

[8]:
```
symboling             90
normalized_losses     90
make                  90
fuel_type             90
aspiration            90
number_of_doors       90
body_style            90
drive_wheels          90
engine_location       90
```

```
wheel_base            90
length                90
width                 90
height                90
curb_weight           90
engine_type           90
number_of_cylinders   90
engine_size           90
fuel_system           90
bore                  90
stroke                90
compression_ratio     90
horsepower            90
peak_rpm              90
city_mpg              90
highway_mpg           90
price                 90
dtype: int64
```

** How many hatchback cars are in the dataset ? **

[9]:
```
auto[auto['body_style'] == 'hatchback'].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 68 entries, 2 to 186
Data columns (total 26 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   symboling            68 non-null     int64
 1   normalized_losses    68 non-null     int64
 2   make                 68 non-null     object
 3   fuel_type            68 non-null     object
 4   aspiration           68 non-null     object
 5   number_of_doors      68 non-null     object
 6   body_style           68 non-null     object
 7   drive_wheels         68 non-null     object
 8   engine_location      68 non-null     object
 9   wheel_base           68 non-null     float64
 10  length               68 non-null     float64
 11  width                68 non-null     float64
 12  height               68 non-null     float64
 13  curb_weight          68 non-null     int64
 14  engine_type          68 non-null     object
 15  number_of_cylinders  68 non-null     object
 16  engine_size          68 non-null     int64
 17  fuel_system          68 non-null     object
 18  bore                 68 non-null     float64
 19  stroke               68 non-null     float64
```

```
 20  compression_ratio    68 non-null     float64
 21  horsepower           68 non-null     int64
 22  peak_rpm             68 non-null     int64
 23  city_mpg             68 non-null     int64
 24  highway_mpg          68 non-null     int64
 25  price                68 non-null     int64
dtypes: float64(7), int64(9), object(10)
memory usage: 14.3+ KB
```

** What are the 3 most commonly found cars in the dataset? **

[10]:
```
auto['make'].value_counts().head(3)
```

[10]:
```
make
toyota    32
nissan    18
mazda     17
Name: count, dtype: int64
```

** Someone purchased a car for 7099, what is the make of the car? **

[11]:
```
auto[auto['price']==7099]['make']
```

[11]:
```
87    nissan
Name: make, dtype: object
```

*** Which cars are priced greater than 40000? **

[12]:
```
auto[auto["price"] >40000]
```

[12]:

	symboling	normalized_losses	make	fuel_type	aspiration	\
15	0	149	bmw	gas	std	
70	0	140	mercedes-benz	gas	std	
71	1	140	mercedes-benz	gas	std	

	number_of_doors	body_style	drive_wheels	engine_location	wheel_base	…	\
15	two	sedan	rwd	front	103.5	…	
70	four	sedan	rwd	front	120.9	…	
71	two	hardtop	rwd	front	112.0	…	

	engine_size	fuel_system	bore	stroke	compression_ratio	horsepower	\
15	209	mpfi	3.62	3.39	8.0	182	
70	308	mpfi	3.80	3.35	8.0	184	
71	304	mpfi	3.80	3.35	8.0	184	

	peak_rpm	city_mpg	highway_mpg	price
15	5400	16	22	41315
70	4500	14	16	40960
71	4500	14	16	45400

[3 rows x 26 columns]

** Which are the cars that are both a sedan and priced less than 7000? **

[13]:
```
auto[(auto['body_style']=='sedan') & (auto['price']<7000)]
```

[13]:

	symboling	normalized_losses	make	fuel_type	aspiration	\
19	0	81	chevrolet	gas	std	
24	1	148	dodge	gas	std	
42	0	110	isuzu	gas	std	
50	1	113	mazda	gas	std	
82	1	125	mitsubishi	gas	std	
86	1	128	nissan	gas	std	
88	1	128	nissan	gas	std	
89	1	122	nissan	gas	std	
118	1	154	plymouth	gas	std	
152	0	91	toyota	gas	std	

	number_of_doors	body_style	drive_wheels	engine_location	wheel_base	…	\
19	four	sedan	fwd	front	94.5	…	
24	four	sedan	fwd	front	93.7	…	
42	four	sedan	rwd	front	94.3	…	
50	four	sedan	fwd	front	93.1	…	
82	four	sedan	fwd	front	96.3	…	
86	two	sedan	fwd	front	94.5	…	
88	two	sedan	fwd	front	94.5	…	
89	four	sedan	fwd	front	94.5	…	
118	four	sedan	fwd	front	93.7	…	
152	four	sedan	fwd	front	95.7	…	

	engine_size	fuel_system	bore	stroke	compression_ratio	horsepower	\
19	90	2bbl	3.03	3.11	9.6	70	
24	90	2bbl	2.97	3.23	9.4	68	
42	111	2bbl	3.31	3.23	8.5	78	
50	91	2bbl	3.03	3.15	9.0	68	
82	122	2bbl	3.35	3.46	8.5	88	
86	97	2bbl	3.15	3.29	9.4	69	
88	97	2bbl	3.15	3.29	9.4	69	
89	97	2bbl	3.15	3.29	9.4	69	
118	90	2bbl	2.97	3.23	9.4	68	
152	98	2bbl	3.19	3.03	9.0	70	

	peak_rpm	city_mpg	highway_mpg	price
19	5400	38	43	6575
24	5500	31	38	6692
42	4800	24	29	6785
50	5000	31	38	6695

```
82       5000       25          32   6989
86       5200       31          37   5499
88       5200       31          37   6649
89       5200       31          37   6849
118      5500       31          38   6692
152      4800       30          37   6938

[10 rows x 26 columns]
```

0.1.1 The END