Pandas Lab Exercise (Kaggle Automobile Dataset)

We shall now test your skills in using Pandas package. We will be using the [automobiles Datase](https://www.kaggle.com/nisargpatel/automobiles/data) Kaggle.

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

In [8]: **import** pandas **as** pd

# Load the Automobile dataset into variable "auto"

In [11]: auto**=**pd**.**read\_csv('Automobile.csv')

# Check the head of the DataFrame.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| In [16]: | auto**.**head() |  | | | | | |
| Out[16]: | **symboling** | **normalized\_losses** | **make** | **fuel\_type** | **aspiration** | **number\_of\_doors** | **body\_s** |

romero

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** 3 168 alfa- gas | | | | | std | two | conver |
| **1** 3 168 alfa- gas | | | | | std | two | conver |
| **2** 1 168 alfa- gas | | | | |  |  |  |
|  |  |  | romero |  |  |  |  |
| **3** | 2 | 164 | audi | gas | std | four | se |
| **4** | 2 | 164 | audi | gas | std | four | se |

romero

std two hatch

5 rows × 26 columns

# How many rows and columns are there?

In [20]: auto**.**shape

Out[20]: (201, 26)

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

In [22]: auto['price']**.**mean()

Out[22]: 13207.129353233831

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

In [44]: auto['price']**.**max(),auto['price']**.**idxmax(),

Out[44]: (45400, 71)

In [46]: auto['price']**.**min(),auto['price']**.**idxmin()

Out[46]: (5118, 134)

# How many cars have horsepower greater than 100?

In [50]: horse\_power**=**auto[auto['horsepower']**>**100] horse\_power

Out[50]: **symboling normalized\_losses make fuel\_type aspiration number\_of\_doors body**

**0** 3 168 alfa- romero

|  |  |  |  |
| --- | --- | --- | --- |
| gas | std | two | conv |
| gas | std | two | conv |

**1** 3 168 alfa- romero

hat

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2** | 1 | 168 | alfa- romero | gas | std | two |
| **3** | 2 | 164 | audi | gas | std | four |
| **4** | 2 | 164 | audi | gas | std | four |
| **...** | ... | ... | ... | ... | ... | ... |
| **196** | -1 | 95 | volvo | gas | std | four |
| **197** | -1 | 95 | volvo | gas | turbo | four |
| **198** | -1 | 95 | volvo | gas | std | four |
| **199** | -1 | 95 | volvo | diesel | turbo | four |
| **200** | -1 | 95 | volvo | gas | turbo | four |

90 rows × 26 columns

# How many hatchback cars are in the dataset ?

In [66]: hatchback\_count**=**auto[auto['body\_style']**==**'hatchback']**.**shape[0] hatchback\_count

Out[66]: 68

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

In [72]: most\_common\_cars**=**auto['make']**.**value\_counts()**.**head(3) most\_common\_cars

Out[72]: make

toyota 32

nissan 18

mazda 17

Name: count, dtype: int64

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

In [74]: make\_Car**=**auto[auto['price']**==**7099]['make'] make\_Car

Out[74]: 87 nissan

Name: make, dtype: object

# Which cars are priced greater than 40000?

In [90]: greater\_than\_40k**=**auto[auto['price']**>**40000][['make','price']] greater\_than\_40k

|  |  |  |  |
| --- | --- | --- | --- |
| Out[90]: | **15** | **make**  bmw | **price**  41315 |
|  | **70** | mercedes-benz | 40960 |
|  | **71** | mercedes-benz | 45400 |

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

In [104]: condition**=**auto[(auto['body\_style']**==** 'sedan') **&** (auto['price'] **<** 7000)][[ condition

|  |  |  |  |
| --- | --- | --- | --- |
| Out[104]: | **19** | **body\_style**  sedan | **price**  6575 |
|  | **24** | sedan | 6692 |
|  | **42** | sedan | 6785 |
|  | **50** | sedan | 6695 |
|  | **82** | sedan | 6989 |
|  | **86** | sedan | 5499 |
|  | **88** | sedan | 6649 |
|  | **89** | sedan | 6849 |
|  | **118** | sedan | 6692 |
|  | **152** | sedan | 6938 |

# Count the number of unique values in the fuel\_type column.

In [112]: unique\_values**=**auto['fuel\_type']**.**unique() unique\_values

Out[112]: array(['gas', 'diesel'], dtype=object)

# List all the cars that have a horsepower between 100 and 200, and display their make ,

**horsepower , and**  **price .**

In [158]: condition\_2**=** auto[(auto['horsepower']**>**100) **&** (auto['horsepower']**<=**200) ] condition\_2[['make','horsepower','price']]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Out[158]: | **0** | **make**  alfa-romero | **horsepower**  111 | **price**  13495 |
|  | **1** | alfa-romero | 111 | 16500 |
|  | **2** | alfa-romero | 154 | 16500 |
|  | **3** | audi | 102 | 13950 |
|  | **4** | audi | 115 | 17450 |
|  | **...**  **196** | ...  volvo | ...  114 | ...  16845 |
|  | **197** | volvo | 160 | 19045 |
|  | **198** | volvo | 134 | 21485 |
|  | **199** | volvo | 106 | 22470 |
|  | **200** | volvo | 114 | 22625 |

86 rows × 3 columns

# Find the average city\_mpg and highway\_mpg for each body\_style .

In [122]: avg\_mpg **=** auto**.**groupby('body\_style')[['city\_mpg','highway\_mpg']]**.**mean() avg\_mpg

|  |  |  |  |
| --- | --- | --- | --- |
| Out[122]: | **body\_style** | **city\_mpg** | **highway\_mpg** |
|  | **convertible** | 20.500000 | 26.000000 |
|  | **hardtop** | 21.625000 | 27.250000 |
|  | **hatchback** | 26.602941 | 32.382353 |
|  | **sedan** | 25.053191 | 30.574468 |
|  | **wagon** | 24.040000 | 28.720000 |

# What is the median price for each make ?

In [124]: median\_price **=** auto**.**groupby('make')['price']**.**mean() median\_price

|  |  |  |
| --- | --- | --- |
| Out[124]: | make |  |
|  | alfa-romero | 15498.333333 |
|  | audi | 17859.166667 |
|  | bmw | 26118.750000 |
|  | chevrolet | 6007.000000 |
|  | dodge | 7875.444444 |
|  | honda | 8184.692308 |
|  | isuzu | 8916.500000 |
|  | jaguar | 34600.000000 |
|  | mazda | 10652.882353 |
|  | mercedes-benz | 33647.000000 |
|  | mercury | 16503.000000 |
|  | mitsubishi | 9239.769231 |
|  | nissan | 10415.666667 |
|  | peugot | 15489.090909 |
|  | plymouth | 7963.428571 |
|  | porsche | 31400.500000 |
|  | renault | 9595.000000 |
|  | saab | 15223.333333 |
|  | subaru | 8541.250000 |
|  | toyota | 9885.812500 |
|  | volkswagen | 10077.500000 |
|  | volvo | 18063.181818 |
|  | Name: price, | dtype: float64 |

# List all cars that have a wheel\_base greater than 100 and a curb\_weight less than 25

In [128]: condition\_3 **=** auto[(auto['wheel\_base'] **>**100 ) **&** (auto['curb\_weight']**<**2500 condition\_3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Out[128]: | **symboling** | **normalized\_losses** | **make** | **fuel\_type** | **aspiration** | **number\_of\_doors** | **body\_** |
| **9** | 2 | 192 | bmw | gas | std | two |  |
| **10** | 0 | 192 | bmw | gas | std | four |  |
| **169** | -1 | 65 | toyota | gas | std | four |  |
| **170** | -1 | 65 | toyota | diesel | turbo | four |  |
| **171** | -1 | 65 | toyota | gas | std | four | hatc |
| **172** | -1 | 65 | toyota | gas | std | four |  |
| **173** | -1 | 65 | toyota | gas | std | four | hatc |

7 rows × 26 columns

# Create a new column price\_per\_hp that calculates the price of the car per horsepower.

In [134]: auto['price\_per\_hp']**=** auto['price']**/**auto['horsepower'] auto**.**head()

Out[134]: **symboling normalized\_losses make fuel\_type aspiration number\_of\_doors body\_s**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 3 | 168 | alfa- romero | gas | std | two | conver |
| **1** | 3 | 168 | alfa- romero | gas | std | two | conver |
| **2** | 1 | 168 | alfa- romero | gas | std | two | hatch |
| **3** | 2 | 164 | audi | gas | std | four | se |
| **4** | 2 | 164 | audi | gas | std | four | se |

5 rows × 27 columns

# Count how many cars have a number\_of\_doors as four .

In [136]: four\_number\_of\_doors**=**auto[auto['number\_of\_doors']**==**'four']**.**shape[0] four\_number\_of\_doors

Out[136]: 114

# Find the top 5 cars based on their highway\_mpg and price .

In [144]: top\_5\_cars**=**auto**.**sort\_values(['highway\_mpg','price'],ascending**=**[**False**,**Fals** top\_5\_cars

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Out[144]: | **symboling** | **normalized\_losses** | **make** | **fuel\_type** | **aspiration** | **number\_of\_doors** | **bo** |
| **29** | 2 | 137 | honda | gas | std | two | h |
| **17** | 2 | 121 | chevrolet | gas | std | two | h |
| **87** | 1 | 128 | nissan | diesel | std | two |  |
| **155** | 0 | 91 | toyota | diesel | std | four | h |
| **156** | 0 | 91 | toyota | gas | std | four |  |

5 rows × 27 columns

# How many cars have missing values in the normalized\_losses column?

In [150]: missing\_values **=** auto['normalized\_losses']**.**isnull()**.**sum() missing\_values

Out[150]: 0

# Create a new column car\_age that calculates the age of the car based on the

**year\_of\_manufacture (assume the current year is 2025).**

In [ ]: current\_year **=** 2025

auto['car\_age'] **=** current\_year **-** auto['year\_of\_manufacture'] auto**.**head()

The END