Here, The Data Of Different Cars Is Given With Their Specifications In [1]: import pandas as pd # We will analyze datab In [2]: cars = pd.read_csv(r"C:\Users\LENOVO\Downloads\2. Cars Data1.csv") cars MSRP Invoice EngineSize Cylinders Horsepower MPG_City MPG_Highway Weight Wheelbase Length Make Origin DriveTrain Out[2]: Model Type MDX SUV All \$36,945 \$33,337 3.5 6.0 265.0 17.0 23.0 4451.0 106.0 189.0 0 Acura Asia 1 Acura RSX Type S 2dr Sedan Asia Front \$23,820 \$21,761 2.0 4.0 200.0 24.0 31.0 2778.0 101.0 172.0 2 Acura TSX 4dr Sedan Front \$26,990 \$24,647 2.4 4.0 200.0 22.0 29.0 3230.0 105.0 183.0 Asia 3 Acura TL 4dr Sedan Asia Front \$33,195 \$30,299 3.2 6.0 270.0 20.0 28.0 3575.0 108.0 186.0 3.5 RL 4dr Sedan Front \$43,755 \$39,014 3.5 6.0 225.0 18.0 24.0 3880.0 115.0 197.0 4 Acura Asia 427 Volvo C70 LPT convertible 2dr Sedan Europe Front \$40,565 \$38,203 2.4 5.0 197.0 21.0 28.0 3450.0 105.0 186.0 Volvo C70 HPT convertible 2dr Sedan Europe Front \$42,565 \$40,083 2.3 5.0 242.0 20.0 26.0 3450.0 105.0 186.0 429 Volvo Front \$45,210 \$42,573 2.9 6.0 268.0 19.0 26.0 3653.0 110.0 190.0 S80 T6 4dr Sedan Europe 170.0 29.0 2822.0 180.0 430 Volvo V40 Wagon Europe Front \$26,135 \$24,641 1.9 4.0 22.0 101.0 431 Volvo XC70 Wagon Europe All \$35,145 \$33,112 2.5 5.0 208.0 20.0 27.0 3823.0 109.0 186.0 432 rows × 15 columns In [3]: cars.shape # Checking rows and coloumn Out[3]: (432, 15) # Q.1 Find All The Null Value In The Dataset. If There Is Any Null Value In Any Coloumn, Then Fill It With The Mean Of That Coloumn In [6]: cars.isnull().sum() # we can clearly see that cylinders coloumn has two null value Make Out[6]: Model 4 Type 4 4 Origin DriveTrain **MSRP** Invoice EngineSize Cylinders Horsepower MPG_City 4 4 MPG_Highway Weight Wheelbase 4 Length dtype: int64 In [7]: cars.head(2) MSRP Invoice EngineSize Cylinders Horsepower MPG_City MPG_Highway Weight Wheelbase Length Out[7]: Make Model Type Origin DriveTrain MDX 189.0 **0** Acura SUV All \$36,945 \$33,337 3.5 6.0 265.0 17.0 23.0 4451.0 106.0 Asia Front \$23,820 \$21,761 1 Acura RSX Type S 2dr Sedan Asia 2.0 4.0 200.0 24.0 31.0 2778.0 101.0 172.0 In [10] cars['Cylinders'].fillna(cars['Cylinders'].mean(), inplace = True) In [11]: cars.isnull().sum() # we can see that now cylinder coloumn has no null value 4 Make Out[11]: Model 4 Type 4 **Origin** 4 DriveTrain **MSRP** Invoice EngineSize 4 0 Cylinders Horsepower 4 MPG_City 4 MPG_Highway Weight 4 Wheelbase Length 4 dtype: int64 In []: # Q.2 Check What Are The Different Types Of Make Are There In Our Dataset And What Is The Count(Occurence) Of Each Make In The Data??? In [12]: cars.head(2) Type Origin DriveTrain Make **MSRP** Invoice EngineSize Cylinders Horsepower MPG_City MPG_Highway Weight Wheelbase Length Out[12]: Model **0** Acura MDX SUV Asia All \$36,945 \$33,337 3.5 6.0 265.0 17.0 23.0 4451.0 106.0 189.0 2.0 4.0 200.0 31.0 2778.0 101.0 172.0 1 Acura RSX Type S 2dr Sedan Asia Front \$23,820 \$21,761 24.0 In [13] cars['Make'].value_counts() # we can see that which car has how many records(rows) in this data 28 Toyota Out[13]: Chevrolet 27 Mercedes-Benz 26 23 Ford 20 BMW Audi 19 17 Nissan Honda 17 Volkswagen 15 Chrysler 15 Mitsubishi 13 Dodge 13 Hyundai 12 Jaguar 12 Volvo 12 Lexus 11 11 Kia Subaru 11 Mazda 11 Pontiac 11 9 Lincoln 9 Mercury Buick 9 Infiniti 8 8 Cadillac Saturn 8 GMC 8 8 Suzuki 7 Saab 7 Porsche 7 Acura 3 Jeep 3 Land Rover Oldsmobile 3 MINI 2 2 Scion 2 Isuzu Hummer 1 Name: Make, dtype: int64 # Q.3 Show All The Records Where Origin Is Asia Or Europe?? In [14]: cars.head(2) Out[14]: Invoice EngineSize Cylinders Horsepower MPG_City MPG_Highway Weight Wheelbase Length Make Model Type Origin DriveTrain MSRP **0** Acura All \$36,945 265.0 4451.0 189.0 MDX SUV Asia \$33,337 3.5 6.0 17.0 23.0 106.0 Front \$23,820 \$21,761 1 Acura RSX Type S 2dr Sedan Asia 2.0 200.0 24.0 31.0 2778.0 101.0 172.0 In [16]: cars[cars['Origin'].isin(['Asia', 'Europe'])] # by using filteration method we can see the data Origin DriveTrain MSRP Invoice EngineSize Cylinders Horsepower MPG_City MPG_Highway Weight Wheelbase Length Out[16]: Make Model Type **0** Acura MDX SUV All \$36,945 \$33,337 3.5 6.0 265.0 17.0 23.0 4451.0 106.0 189.0 Asia Front \$23,820 \$21,761 200.0 31.0 2778.0 172.0 1 Acura RSX Type S 2dr Sedan Asia 2.0 4.0 24.0 101.0 183.0 2 Acura TSX 4dr Sedan Front \$26,990 \$24,647 2.4 4.0 200.0 22.0 29.0 3230.0 105.0 Asia 3 Acura Front \$33,195 \$30,299 3.2 6.0 270.0 20.0 28.0 3575.0 186.0 TL 4dr Sedan Asia 108.0 3.5 RL 4dr Sedan Front \$43,755 \$39,014 3.5 6.0 225.0 18.0 24.0 3880.0 115.0 197.0 4 Acura Asia Europe **427** Volvo C70 LPT convertible 2dr Sedan Front \$40,565 \$38,203 2.4 5.0 197.0 21.0 28.0 3450.0 105.0 186.0 Front \$42,565 \$40,083 242.0 26.0 3450.0 186.0 428 Volvo C70 HPT convertible 2dr Sedan Europe 2.3 5.0 20.0 105.0 Front \$45,210 \$42,573 268.0 190.0 429 Volvo S80 T6 4dr Sedan Europe 2.9 6.0 19.0 26.0 3653.0 110.0 430 Volvo Front \$26,135 \$24,641 170.0 29.0 2822.0 180.0 V40 Wagon Europe 1.9 4.0 22.0 101.0 27.0 3823.0 431 Volvo XC70 Wagon Europe All \$35,145 \$33,112 2.5 5.0 208.0 20.0 109.0 186.0 281 rows × 15 columns In []: # Q.4 Remove All The Records(Rows) Where Weight Is Above 4000??? (Removing Unwanted Rows From The Dataset) In [17]: cars.head(2) Make Type Origin DriveTrain **MSRP** Invoice EngineSize Cylinders Horsepower MPG_City MPG_Highway Weight Wheelbase Length Out[17]: Model 0 Acura MDX SUV Asia All \$36,945 \$33,337 3.5 6.0 265.0 17.0 23.0 4451.0 106.0 189.0 1 Acura RSX Type S 2dr Sedan Front \$23,820 \$21,761 2.0 4.0 200.0 24.0 31.0 2778.0 172.0 Asia 101.0 In [18]: cars[cars['Weight'] > 4000] # we can see all the weight rows which is greater than 4000 MSRP Invoice EngineSize Cylinders Horsepower MPG_City MPG_Highway Weight Wheelbase Length Out[18]: Make Model Type Origin DriveTrain 0 MDX SUV Asia \$36,945 \$33,337 3.5 6.0 265.0 17.0 4451.0 106.0 189.0 Acura ΑII 23.0 15 Audi A4 3.0 Quattro convertible 2dr Sedan Europe All \$44,240 \$40,075 3.0 6.0 220.0 18.0 25.0 4013.0 105.0 180.0 193.0 17 Sedan Europe All \$49,690 \$44,936 4024.0 Audi A6 4.2 Quattro 4dr 4.2 8.0 300.0 17.0 24.0 109.0 18 A8 L Quattro 4dr Sedan Europe \$69,190 \$64,740 4.2 8.0 330.0 17.0 4399.0 121.0 204.0 Audi 24.0 20 Front \$84,600 \$76,417 8.0 450.0 15.0 4024.0 109.0 191.0 Audi RS 6 4dr Sports Europe 4.2 22.0 ••• 405 Volkswagen Touareg V6 SUV Europe All \$35,515 \$32,243 220.0 5086.0 112.0 187.0 3.2 6.0 15.0 20.0 Phaeton 4dr Sedan Europe Front \$65,000 \$59,912 335.0 16.0 22.0 5194.0 204.0 415 Volkswagen 118.0 Front \$75,000 \$69,130 12.0 420.0 12.0 5399.0 204.0 **416** Volkswagen Phaeton W12 4dr Sedan Europe 6.0 19.0 118.0 419 Volkswagen Front \$40,235 \$36,956 8.0 270.0 18.0 4067.0 184.0 Passat W8 Wagon Europe 4.0 25.0 106.0 420 Volvo XC90 T6 SUV Europe All \$41,250 \$38,851 2.9 6.0 268.0 15.0 20.0 4638.0 113.0 189.0 103 rows × 15 columns In [19]: cars[~(cars['Weight'] > 4000)] # we cann see in the data that all the rows greater than 4000 has been removed Make Origin DriveTrain MSRP Invoice EngineSize Cylinders Horsepower MPG_City MPG_Highway Weight Wheelbase Length Out[19]: Model Type 172.0 1 Acura RSX Type S 2dr Sedan Asia Front \$23,820 \$21,761 2.0 4.0 200.0 24.0 31.0 2778.0 101.0 2 Acura TSX 4dr Sedan Asia Front \$26,990 \$24,647 2.4 4.0 200.0 22.0 29.0 3230.0 105.0 183.0 28.0 3575.0 Front \$33,195 \$30,299 270.0 186.0 **3** Acura TL 4dr Sedan 3.2 6.0 20.0 108.0 Asia 3.5 RL 4dr Sedan 225.0 18.0 24.0 3880.0 197.0 4 Acura Asia Front \$43,755 \$39,014 3.5 6.0 115.0 225.0 197.0 5 Acura 3.5 RL w/Navigation 4dr Sedan Asia Front \$46,100 \$41,100 3.5 6.0 18.0 24.0 3893.0 115.0 427 Volvo C70 LPT convertible 2dr Sedan Europe Front \$40,565 \$38,203 28.0 3450.0 186.0 2.4 5.0 197.0 21.0 105.0 Front \$42,565 \$40,083 2.3 242.0 20.0 26.0 3450.0 186.0 428 Volvo C70 HPT convertible 2dr Sedan Europe 5.0 105.0 **429** Volvo Front \$45,210 \$42,573 2.9 6.0 268.0 19.0 26.0 3653.0 110.0 190.0 S80 T6 4dr Sedan Europe 1.9 4.0 170.0 22.0 29.0 2822.0 180.0 430 Volvo V40 Wagon Europe Front \$26,135 \$24,641 101.0 27.0 3823.0 **431** Volvo XC70 Wagon Europe All \$35,145 \$33,112 2.5 5.0 208.0 20.0 109.0 186.0 329 rows × 15 columns # Q.5 Increase All The Values Of 'MPG_City' Coloumn By 3??? (Applying Function On Coloumn) In [20] cars.head(2) Out[20]: Make Type Origin DriveTrain MSRP Invoice EngineSize Cylinders Horsepower MPG_City MPG_Highway Weight Wheelbase Length Model **0** Acura MDX SUV All \$36,945 \$33,337 3.5 6.0 265.0 17.0 23.0 4451.0 106.0 189.0 Asia Front \$23,820 \$21,761 1 Acura RSX Type S 2dr Sedan Asia 2.0 4.0 200.0 24.0 31.0 2778.0 101.0 172.0 In [21]: cars['MPG_City'] = cars['MPG_City'].apply(lambda x:x+3) In [22]: cars # we can clearly see that 'MPG_City' coloumn has been increased by 3 Out[22]: Make Model Type Origin DriveTrain MSRP Invoice EngineSize Cylinders Horsepower MPG_City MPG_Highway Weight Wheelbase Length **0** Acura MDX SUV Asia All \$36,945 \$33,337 3.5 6.0 265.0 20.0 23.0 4451.0 106.0 189.0 1 Acura RSX Type S 2dr Sedan Front \$23,820 \$21,761 4.0 200.0 27.0 31.0 2778.0 172.0 Asia 2.0 101.0 Front \$26,990 \$24,647 200.0 25.0 29.0 3230.0 183.0 **2** Acura TSX 4dr Sedan 2.4 4.0 105.0 Asia **3** Acura Front \$33,195 \$30,299 3.2 270.0 23.0 TL 4dr Sedan Asia 6.0 28.0 3575.0 108.0 186.0 4 Acura 3.5 RL 4dr Sedan Asia Front \$43,755 \$39,014 3.5 6.0 225.0 21.0 24.0 3880.0 115.0 197.0 427 Volvo C70 LPT convertible 2dr Sedan Europe Front \$40,565 \$38,203 24.0 28.0 3450.0 186.0 2.4 5.0 197.0 105.0 2.3 5.0 242.0 23.0 26.0 3450.0 186.0 428 Volvo C70 HPT convertible 2dr Sedan Europe Front \$42,565 \$40,083 105.0 S80 T6 4dr Sedan Europe 190.0 **429** Volvo Front \$45,210 \$42,573 2.9 6.0 268.0 22.0 26.0 3653.0 110.0 170.0 25.0 29.0 2822.0 180.0 430 Volvo V40 Wagon Europe Front \$26,135 \$24,641 1.9 4.0 101.0 **431** Volvo XC70 Wagon Europe All \$35,145 \$33,112 2.5 5.0 208.0 23.0 27.0 3823.0 109.0 186.0 432 rows × 15 columns