

```
In [1]: #install specific version of libraries used in lab
#! mamba install pandas==1.3.3 -y
#! mamba install numpy==1.21.2 -y

In [2]: # import pandas library
import pandas as pd
import numpy as np

In [11]: #This function will download the dataset into your browser

async def download(url, filename):
    response = await pyfetch(url)
    if response.status == 200:
        with open(filename, "wb") as f:
            f.write(await response.bytes())

In [12]: path = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DA0101EN-SkillsNetwork/labs/Data%20files/auto.csv"

In [15]: # Import pandas library
import pandas as pd

# Read the online file by the URL provides above, and assign it to variable "df"
df = pd.read_csv(path, header=None)
df

Out[15]:
   0  1  2  3  4  5  6  7  8  9 ... 16 17 18 19 20 21 22 23 24 25
0  0  3  ?  alfa-romero  gas  std  two  convertible  rwd  front  88.6  ...  130  mpfi  3.47  2.68  9.0  111  5000  21  27  13495
1  1  3  ?  alfa-romero  gas  std  two  convertible  rwd  front  88.6  ...  130  mpfi  3.47  2.68  9.0  111  5000  21  27  16500
2  2  1  ?  alfa-romero  gas  std  two  hatchback  rwd  front  94.5  ...  152  mpfi  2.68  3.47  9.0  154  5000  19  26  16500
3  3  2  164  audi  gas  std  four  sedan  fwd  front  99.8  ...  109  mpfi  3.19  3.40  10.0  102  5500  24  30  13950
4  4  2  164  audi  gas  std  four  sedan  4wd  front  99.4  ...  136  mpfi  3.19  3.40  8.0  115  5500  18  22  17450
... ..
200 -1  95  volvo  gas  std  four  sedan  rwd  front  109.1  ...  141  mpfi  3.78  3.15  9.5  114  5400  23  28  16845
201 -1  95  volvo  gas  turbo  four  sedan  rwd  front  109.1  ...  141  mpfi  3.78  3.15  8.7  160  5300  19  25  19045
202 -1  95  volvo  gas  std  four  sedan  rwd  front  109.1  ...  173  mpfi  3.58  2.87  8.8  134  5500  18  23  21485
203 -1  95  volvo  diesel  turbo  four  sedan  rwd  front  109.1  ...  145  idi  3.01  3.40  23.0  106  4800  26  27  22470
204 -1  95  volvo  gas  turbo  four  sedan  rwd  front  109.1  ...  141  mpfi  3.78  3.15  9.5  114  5400  19  25  22625

205 rows x 26 columns

In [16]: df.head(10)

Out[16]:
   0  1  2  3  4  5  6  7  8  9 ... 16 17 18 19 20 21 22 23 24 25
0  0  3  ?  alfa-romero  gas  std  two  convertible  rwd  front  88.6  ...  130  mpfi  3.47  2.68  9.0  111  5000  21  27  13495
1  1  3  ?  alfa-romero  gas  std  two  convertible  rwd  front  88.6  ...  130  mpfi  3.47  2.68  9.0  111  5000  21  27  16500
2  2  1  ?  alfa-romero  gas  std  two  hatchback  rwd  front  94.5  ...  152  mpfi  2.68  3.47  9.0  154  5000  19  26  16500
3  3  2  164  audi  gas  std  four  sedan  fwd  front  99.8  ...  109  mpfi  3.19  3.40  10.0  102  5500  24  30  13950
4  4  2  164  audi  gas  std  four  sedan  4wd  front  99.4  ...  136  mpfi  3.19  3.40  8.0  115  5500  18  22  17450
5  5  2  ?  audi  gas  std  two  sedan  fwd  front  99.8  ...  136  mpfi  3.19  3.40  8.5  110  5500  19  25  15250
6  6  1  158  audi  gas  std  four  sedan  fwd  front  105.8  ...  136  mpfi  3.19  3.40  8.5  110  5500  19  25  17710
7  7  1  ?  audi  gas  std  four  wagon  fwd  front  105.8  ...  136  mpfi  3.19  3.40  8.5  110  5500  19  25  18920
8  8  1  158  audi  gas  turbo  four  sedan  fwd  front  105.8  ...  131  mpfi  3.13  3.40  8.3  140  5500  17  20  23875
9  9  0  ?  audi  gas  turbo  two  hatchback  4wd  front  99.5  ...  131  mpfi  3.13  3.40  7.0  160  5500  16  22  ?

10 rows x 26 columns

In [17]: #Check the bottom 10 rows of data frame "df".
df.tail(10)

Out[17]:
   0  1  2  3  4  5  6  7  8  9 ... 16 17 18 19 20 21 22 23 24 25
195 -1  74  volvo  gas  std  four  wagon  rwd  front  104.3  ...  141  mpfi  3.78  3.15  9.5  114  5400  23  28  13415
196 -2  103  volvo  gas  std  four  sedan  rwd  front  104.3  ...  141  mpfi  3.78  3.15  9.5  114  5400  24  28  15985
197 -1  74  volvo  gas  std  four  wagon  rwd  front  104.3  ...  141  mpfi  3.78  3.15  9.5  114  5400  24  28  16515
198 -2  103  volvo  gas  turbo  four  sedan  rwd  front  104.3  ...  130  mpfi  3.62  3.15  7.5  162  5100  17  22  18420
199 -1  74  volvo  gas  turbo  four  wagon  rwd  front  104.3  ...  130  mpfi  3.62  3.15  7.5  162  5100  17  22  18950
200 -1  95  volvo  gas  std  four  sedan  rwd  front  109.1  ...  141  mpfi  3.78  3.15  9.5  114  5400  23  28  16845
201 -1  95  volvo  gas  turbo  four  sedan  rwd  front  109.1  ...  141  mpfi  3.78  3.15  8.7  160  5300  19  25  19045
202 -1  95  volvo  gas  std  four  sedan  rwd  front  109.1  ...  173  mpfi  3.58  2.87  8.8  134  5500  18  23  21485
203 -1  95  volvo  diesel  turbo  four  sedan  rwd  front  109.1  ...  145  idi  3.01  3.40  23.0  106  4800  26  27  22470
204 -1  95  volvo  gas  turbo  four  sedan  rwd  front  109.1  ...  141  mpfi  3.78  3.15  9.5  114  5400  19  25  22625

10 rows x 26 columns

In [18]: # create headers list
headers = ["symboling", "normalized-losses", "make", "fuel-type", "aspiration", "num-of-doors", "body-style",
           "drive-wheels", "engine-location", "wheel-base", "length", "width", "height", "curb-weight", "engine-type",
           "num-of-cylinders", "engine-size", "fuel-system", "bore", "stroke", "compression-ratio", "horsepower",
           "peak-rpm", "city-mpg", "highway-mpg", "price"]
print("headers\n", headers)

#We replace headers and recheck our dataframe:
df.columns = headers
df.head(10)

headers
['symboling', 'normalized-losses', 'make', 'fuel-type', 'aspiration', 'num-of-doors', 'body-style', 'drive-wheels', 'engine-location', 'wheel-base', 'length', 'width', 'height', 'curb-weight', 'engine-type', 'num-of-cylinders', 'engine-size', 'fuel-system', 'bore', 'stroke', 'compression-ratio', 'horsepower', 'peak-rpm', 'city-mpg', 'highway-mpg', 'price']

Out[18]:
   symboling  normalized-losses  make  fuel-type  aspiration  num-of-doors  body-style  drive-wheels  engine-location  wheel-base  ...  engine-size  fuel-system  bore  stroke  compression-ratio  horsepower  peak-rpm  city-mpg  highway-mpg  price
0           3                 ?  alfa-romero  gas         std         two  convertible  rwd         front  88.6  ...      130      mpfi  3.47  2.68  9.0  111  5000  21  27  13495
1           3                 ?  alfa-romero  gas         std         two  convertible  rwd         front  88.6  ...      130      mpfi  3.47  2.68  9.0  111  5000  21  27  16500
2           1                 ?  alfa-romero  gas         std         two  hatchback  rwd         front  94.5  ...      152      mpfi  2.68  3.47  9.0  154  5000  19  26  16500
3           2                 164  audi  gas         std         four  sedan  fwd         front  99.8  ...      109      mpfi  3.19  3.40  10.0  102  5500  24  30  13950
4           2                 164  audi  gas         std         four  sedan  4wd         front  99.4  ...      136      mpfi  3.19  3.40  8.0  115  5500  18  22  17450
5           2                 ?  audi  gas         std         two  sedan  fwd         front  99.8  ...      136      mpfi  3.19  3.40  8.5  110  5500  19  25  15250
6           1                 158  audi  gas         std         four  sedan  fwd         front  105.8  ...      136      mpfi  3.19  3.40  8.5  110  5500  19  25  17710
7           1                 ?  audi  gas         std         four  wagon  fwd         front  105.8  ...      136      mpfi  3.19  3.40  8.5  110  5500  19  25  18920
8           1                 158  audi  gas         turbo        four  sedan  fwd         front  105.8  ...      131      mpfi  3.13  3.40  8.3  140  5500  17  20  23875
9           0                 ?  audi  gas         turbo        two  hatchback  4wd         front  99.5  ...      131      mpfi  3.13  3.40  7.0  160  5500  16  22  ?

10 rows x 26 columns

In [19]: #We need to replace the "?" symbol with NaN so the dropna() can remove the missing values:
df1=df.replace('?', np.NaN)

In [20]: #We can drop missing values along the column "price" as follows:
df=df1.dropna(subset=["price"], axis=0)
df.head(20)

Out[20]:
   symboling  normalized-losses  make  fuel-type  aspiration  num-of-doors  body-style  drive-wheels  engine-location  wheel-base  ...  engine-size  fuel-system  bore  stroke  compression-ratio  horsepower  peak-rpm  city-mpg  highway-mpg  price
0           3                 NaN  alfa-romero  gas         std         two  convertible  rwd         front  88.6  ...      130      mpfi  3.47  2.68  9.0  111  5000  21  27  13495
1           3                 NaN  alfa-romero  gas         std         two  convertible  rwd         front  88.6  ...      130      mpfi  3.47  2.68  9.0  111  5000  21  27  16500
2           1                 NaN  alfa-romero  gas         std         two  hatchback  rwd         front  94.5  ...      152      mpfi  2.68  3.47  9.0  154  5000  19  26  16500
3           2                 164  audi  gas         std         four  sedan  fwd         front  99.8  ...      109      mpfi  3.19  3.40  10.0  102  5500  24  30  13950
4           2                 164  audi  gas         std         four  sedan  4wd         front  99.4  ...      136      mpfi  3.19  3.40  8.0  115  5500  18  22  17450
5           2                 NaN  audi  gas         std         two  sedan  fwd         front  99.8  ...      136      mpfi  3.19  3.40  8.5  110  5500  19  25  15250
6           1                 158  audi  gas         std         four  sedan  fwd         front  105.8  ...      136      mpfi  3.19  3.40  8.5  110  5500  19  25  17710
7           1                 NaN  audi  gas         std         four  wagon  fwd         front  105.8  ...      136      mpfi  3.19  3.40  8.5  110  5500  19  25  18920
8           1                 158  audi  gas         turbo        four  sedan  fwd         front  105.8  ...      131      mpfi  3.13  3.40  8.3  140  5500  17  20  23875
10          2                 192  bmw  gas         std         two  sedan  rwd         front  101.2  ...      108      mpfi  3.50  2.80  8.8  101  5800  23  29  16430
11          0                 192  bmw  gas         std         four  sedan  rwd         front  101.2  ...      108      mpfi  3.50  2.80  8.8  101  5800  23  29  16925
12          0                 188  bmw  gas         std         two  sedan  rwd         front  101.2  ...      164      mpfi  3.31  3.19  9.0  121  4250  21  28  20970
13          0                 188  bmw  gas         std         four  sedan  rwd         front  101.2  ...      164      mpfi  3.31  3.19  9.0  121  4250  21  28  21105
14          1                 NaN  bmw  gas         std         four  sedan  rwd         front  103.5  ...      164      mpfi  3.31  3.19  9.0  121  4250  20  25  24565
15          0                 NaN  bmw  gas         std         four  sedan  rwd         front  103.5  ...      209      mpfi  3.62  3.39  8.0  182  5400  16  22  30760
16          0                 NaN  bmw  gas         std         two  sedan  rwd         front  103.5  ...      209      mpfi  3.62  3.39  8.0  182  5400  16  22  41315
17          0                 NaN  bmw  gas         std         four  sedan  rwd         front  110.0  ...      209      mpfi  3.62  3.39  8.0  182  5400  15  20  36880
18          2                 121  chevrolet  gas         std         two  hatchback  fwd         front  88.4  ...      61  2bbl  2.91  3.03  9.5  48  5100  47  53  5151
19          1                 98  chevrolet  gas         std         two  hatchback  fwd         front  94.5  ...      90  2bbl  3.03  3.11  9.6  70  5400  38  43  6295
20          0                 81  chevrolet  gas         std         four  sedan  fwd         front  94.5  ...      90  2bbl  3.03  3.11  9.6  70  5400  38  43  6575

20 rows x 26 columns

In [21]: #Find the name of the columns of the dataframe.
df.columns

Out[21]:
Index(['symboling', 'normalized-losses', 'make', 'fuel-type', 'aspiration',
       'num-of-doors', 'body-style', 'drive-wheels', 'engine-location',
       'wheel-base', 'length', 'width', 'height', 'curb-weight', 'engine-type',
       'num-of-cylinders', 'engine-size', 'fuel-system', 'bore', 'stroke',
       'compression-ratio', 'horsepower', 'peak-rpm', 'city-mpg',
       'highway-mpg', 'price'],
      dtype='object')

In [22]: #Save Dataset
#Correspondingly, Pandas enables us to save the dataset to csv. By using the dataframe.to_csv() method,
#you can add the file path and name along with quotation marks in the brackets.

#For example, if you would save the dataframe df as automobile.csv to your local machine,
#you may use the syntax below, where index = False means the row names will not be written.
df.to_csv("automobile.csv", index=False)

In [31]: #Apply the method to ".describe()" to the columns 'length' and 'compression-ratio'.
path = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DA0101EN-SkillsNetwork/labs/Data%20files/auto.csv"

df = pd.read_csv(path, header=None)
df
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column  Non-Null Count  Dtype
---  -
0  0      205 non-null      int64
1  1      205 non-null      object
2  2      205 non-null      object
3  3      205 non-null      object
4  4      205 non-null      object
5  5      205 non-null      object
6  6      205 non-null      object
7  7      205 non-null      object
8  8      205 non-null      object
9  9      205 non-null      float64
10 10     205 non-null      float64
11 11     205 non-null      float64
12 12     205 non-null      float64
13 13     205 non-null      int64
14 14     205 non-null      object
15 15     205 non-null      object
16 16     205 non-null      int64
17 17     205 non-null      object
18 18     205 non-null      object
19 19     205 non-null      object
20 20     205 non-null      float64
21 21     205 non-null      object
22 22     205 non-null      object
23 23     205 non-null      int64
24 24     205 non-null      int64
25 25     205 non-null      object
dtypes: float64(5), int64(5), object(16)
memory usage: 41.8+ KB

In [32]: # look at the info of "df"
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column  Non-Null Count  Dtype
---  -
0  0      205 non-null      int64
1  1      205 non-null      object
2  2      205 non-null      object
3  3      205 non-null      object
4  4      205 non-null      object
5  5      205 non-null      object
6  6      205 non-null      object
7  7      205 non-null      object
8  8      205 non-null      object
9  9      205 non-null      float64
10 10     205 non-null      float64
11 11     205 non-null      float64
12 12     205 non-null      float64
13 13     205 non-null      int64
14 14     205 non-null      object
15 15     205 non-null      object
16 16     205 non-null      int64
17 17     205 non-null      object
18 18     205 non-null      object
19 19     205 non-null      object
20 20     205 non-null      float64
21 21     205 non-null      object
22 22     205 non-null      object
23 23     205 non-null      int64
24 24     205 non-null      int64
25 25     205 non-null      object
dtypes: float64(5), int64(5), object(16)
memory usage: 41.8+ KB

In [ ]:
```