Untitled

October 21, 2023

[9]:

```
import pandas as pd
import numpy as np
```

[10]:

```
import numpy as np

A = np.array([1, 2, 3])
B = np.array([4, 5, 6])

# Stack A and B vertically
result_vertical = np.vstack((A, B))

# Stack A and B horizontally
result_horizontal = np.hstack((A, B))

print("Stacked Vertically:")
print(result_vertical)

print("\nStacked Horizontally:")
print(result_horizontal)
```

```
Stacked Vertically:
[[1 2 3]
 [4 5 6]]

Stacked Horizontally:
[1 2 3 4 5 6]
```

[11]:

```
common_elements = np.intersect1d(A, B)
print("Common elements between A and B:", common_elements)
```

```
Common elements between A and B: []
```

[12]:

```
A = np.array([1, 6, 7, 3, 8, 12, 9, 4])
result = A[(A >= 5) & (A <= 10)]
print("Numbers between 5 and 10 in A:", result)
```

```
Numbers between 5 and 10 in A: [6 7 8 9]
```

[13]:

```
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
iris_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0, 1, 2, 3])

filtered_rows = iris_2d[(iris_2d[:, 2] > 1.5) & (iris_2d[:, 0] < 5.0)]
print("Filtered Rows:")
print(filtered_rows)
```

```
Filtered Rows:
[[4.8 3.4 1.6 0.2]
 [4.8 3.4 1.9 0.2]
 [4.7 3.2 1.6 0.2]
 [4.8 3.1 1.6 0.2]
 [4.9 2.4 3.3 1. ]
 [4.9 2.5 4.5 1.7]]
```

[14]:

```
url = 'https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.
 ↪csv'
df = pd.read_csv(url)

filtered_data = df.loc[::20, ['Manufacturer', 'Model', 'Type']]
print(filtered_data)
```

```
   Manufacturer    Model     Type
0         Acura  Integra    Small
20     Chrysler  LeBaron  Compact
40        Honda  Prelude   Sporty
60      Mercury   Cougar  Midsize
80       Subaru   Loyale    Small
```

[15]:

```
url = 'https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.
 ↪csv'
df = pd.read_csv(url)

# Calculate the mean for Min.Price and Max.Price, and replace missing values
mean_min_price = df['Min.Price'].mean()
mean_max_price = df['Max.Price'].mean()

df['Min.Price'].fillna(mean_min_price, inplace=True)
df['Max.Price'].fillna(mean_max_price, inplace=True)

print("DataFrame with Missing Values Replaced:")
print(df)
```

```
DataFrame with Missing Values Replaced:
   Manufacturer    Model     Type  Min.Price  Price  Max.Price  MPG.city  \
0         Acura  Integra    Small  12.900000   15.9  18.800000      25.0
1           NaN   Legend  Midsize  29.200000   33.9  38.700000      18.0
2          Audi       90  Compact  25.900000   29.1  32.300000      20.0
```

```
3          Audi      100  Midsize  17.118605   37.7  44.600000      19.0
4           BMW     535i  Midsize  17.118605   30.0  21.459091      22.0
..          …      …      …        …    …        …       …
88   Volkswagen  Eurovan      Van  16.600000   19.7  22.700000      17.0
89   Volkswagen   Passat  Compact  17.600000   20.0  22.400000      21.0
90   Volkswagen  Corrado   Sporty  22.900000   23.3  23.700000      18.0
91        Volvo      240  Compact  21.800000   22.7  23.500000      21.0
92          NaN      850  Midsize  24.800000   26.7  28.500000      20.0

    MPG.highway             AirBags DriveTrain  … Passengers  Length  \
0          31.0                 NaN      Front  …        5.0   177.0
1          25.0  Driver & Passenger      Front  …        5.0   195.0
2          26.0         Driver only      Front  …        5.0   180.0
3          26.0  Driver & Passenger        NaN  …        6.0   193.0
4          30.0                 NaN       Rear  …        4.0   186.0
..          …                 …        …  …        …     …
88         21.0                 NaN      Front  …        7.0   187.0
89         30.0                 NaN      Front  …        5.0   180.0
90         25.0                 NaN      Front  …        4.0   159.0
91         28.0         Driver only       Rear  …        5.0   190.0
92         28.0  Driver & Passenger      Front  …        5.0   184.0

    Wheelbase  Width  Turn.circle Rear.seat.room  Luggage.room  Weight  \
0       102.0   68.0         37.0           26.5           NaN  2705.0
1       115.0   71.0         38.0           30.0          15.0  3560.0
2       102.0   67.0         37.0           28.0          14.0  3375.0
3       106.0    NaN         37.0           31.0          17.0  3405.0
4       109.0   69.0         39.0           27.0          13.0  3640.0
..        …    …          …            …           …     …
88      115.0   72.0         38.0           34.0           NaN  3960.0
89      103.0   67.0         35.0           31.5          14.0  2985.0
90       97.0   66.0         36.0           26.0          15.0  2810.0
91      104.0   67.0         37.0           29.5          14.0  2985.0
92      105.0   69.0         38.0           30.0          15.0  3245.0

     Origin                Make
0   non-USA       Acura Integra
1   non-USA        Acura Legend
2   non-USA             Audi 90
3   non-USA            Audi 100
4   non-USA            BMW 535i
..      …                 …
88      NaN  Volkswagen Eurovan
89  non-USA   Volkswagen Passat
90  non-USA  Volkswagen Corrado
91  non-USA           Volvo 240
92  non-USA           Volvo 850
```

```
[93 rows x 27 columns]
```

[16]:

```
data = np.random.randint(10, 40, 60).reshape(-1, 4)
df = pd.DataFrame(data)

row_sums = df.sum(axis=1)
result = df[row_sums > 100]

print("Rows with Row Sum > 100:")
print(result)
```

```
Rows with Row Sum > 100:
     0   1   2   3
1   39  32  25  11
2   31  36  19  18
3   16  37  26  22
4   37  22  25  39
6   32  25  32  24
9   31  27  18  34
12  21  28  25  30
14  35  32  10  27
```