

# pandas\_introduction\_notebook

December 20, 2025

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

```
[2]: # ML Data Manipulation
```

## 1 2 main Datatypes

```
[3]: # series
```

```
[4]: series = pd.Series(["BMW", "Toyota", "Honda"])
```

```
[5]: series
```

```
[5]: 0      BMW  
1      Toyota  
2      Honda  
dtype: object
```

```
[6]: # series = 1-dimensional
```

```
[7]: colours = pd.Series(["RED", "BLUE", "WHITE"])  
colours
```

```
[7]: 0      RED  
1      BLUE  
2      WHITE  
dtype: object
```

```
[8]: # DataFrame = 2-dimensional
```

```
[9]: car_data = pd.DataFrame({"Car make": series, "Colour": colours})  
car_data
```

```
[9]:   Car make Colour  
0      BMW      RED  
1      Toyota    BLUE  
2      Honda    WHITE
```

```
[10]: # Import data
```

```
[11]: car_sales = pd.read_csv("car-sales.csv")
```

```
[12]: car_sales
```

```
[12]:      Make Colour  Odometer (KM)  Doors      Price
 0  Toyota  White       150043       4  $4,000.00
 1  Honda   Red        87899       4  $5,000.00
 2  Toyota  Blue       32549       3  $7,000.00
 3    BMW  Black       11179       5 $22,000.00
 4  Nissan  White      213095       4  $3,500.00
 5  Toyota  Green      99213       4  $4,500.00
 6  Honda   Blue       45698       4  $7,500.00
 7  Honda   Blue       54738       4  $7,000.00
 8  Toyota  White      60000       4  $6,250.00
 9  Nissan  White      31600       4  $9,700.00
```

```
[13]: # Anatomy of a DataFrame
```

## Anatomy of a DataFrame

	Column (axis = 1)	Make	Colour	Odometer	Doors	Price	Column name
Index number (starts at 0 by default)	0	Toyota	White	150043	4	\$4,000	
	1	Honda	Red	87899	4	\$5,000	
	2	Toyota	Blue	32549	3	\$7,000	
Row (axis = 0)	3	BMW	Black	11179	5	\$22,000	
	4	Nissan	White	213095	4	\$3,500	

```
[14]: # Exporting a DataFrame
```

```
[15]: # Setting `index=False` prevents pandas from writing the DataFrame index as an additional column in the CSV file.
```

```
[16]: car_sales.to_csv("exported-car-sales.csv", index = False)
```

```
[17]: exported_car_sales = pd.read_csv("exported-car-sales.csv")
```

```
[18]: exported_car_sales
```

```
[18]:    Make Colour Odometer (KM)  Doors      Price
  0  Toyota  White       150043      4   $4,000.00
  1  Honda   Red        87899       4   $5,000.00
  2  Toyota  Blue       32549       3   $7,000.00
  3    BMW  Black       11179       5  $22,000.00
  4  Nissan  White     213095      4   $3,500.00
  5  Toyota  Green      99213       4   $4,500.00
  6  Honda   Blue      45698       4   $7,500.00
  7  Honda   Blue      54738       4   $7,000.00
  8  Toyota  White     60000       4   $6,250.00
  9  Nissan  White     31600       4   $9,700.00
```

## 1.1 Describe Data

```
[19]: # Attribute
# get the datatypes of the columns
car_sales.dtypes

# Function
# car_sales.to_csv()
```

```
[19]: Make          object
Colour         object
Odometer (KM)  int64
Doors          int64
Price          object
dtype: object
```

```
[20]: car_sales.columns
```

```
[20]: Index(['Make', 'Colour', 'Odometer (KM)', 'Doors', 'Price'], dtype='object')
```

```
[21]: car_columns = car_sales.columns
```

```
[22]: car_columns
```

```
[22]: Index(['Make', 'Colour', 'Odometer (KM)', 'Doors', 'Price'], dtype='object')
```

```
[23]: car_sales.index
```

```
[23]: RangeIndex(start=0, stop=10, step=1)
```

```
[24]: car_sales
```

```
[24]:    Make Colour Odometer (KM)  Doors      Price
  0  Toyota  White       150043      4   $4,000.00
```

```
1   Honda    Red        87899      4   $5,000.00
2   Toyota   Blue       32549      3   $7,000.00
3     BMW    Black      11179      5   $22,000.00
4   Nissan   White     213095     4   $3,500.00
5   Toyota   Green      99213      4   $4,500.00
6   Honda    Blue       45698      4   $7,500.00
7   Honda    Blue       54738      4   $7,000.00
8   Toyota   White      60000      4   $6,250.00
9   Nissan   White     31600      4   $9,700.00
```

```
[25]: car_sales.describe() # getting a general overview of the numeric columns
```

```
[25]:          Odometer (KM)      Doors
count      10.000000  10.000000
mean      78601.400000  4.000000
std       61983.471735  0.471405
min      11179.000000  3.000000
25%      35836.250000  4.000000
50%      57369.000000  4.000000
75%      96384.500000  4.000000
max      213095.000000 5.000000
```

```
[26]: car_sales.info() # getting general informations abozt the columns
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Make            10 non-null    object  
 1   Colour          10 non-null    object  
 2   Odometer (KM)   10 non-null    int64   
 3   Doors           10 non-null    int64   
 4   Price           10 non-null    object  
dtypes: int64(2), object(3)
memory usage: 532.0+ bytes
```

```
[27]: car_sales.mean(numeric_only = True) #parameter is necessary
```

```
[27]: Odometer (KM)    78601.4
Doors                  4.0
dtype: float64
```

```
[28]: car_prices = pd.Series([200,1700,1222,])
```

```
[29]: car_prices.mean()
```

```
[29]: np.float64(1040.666666666667)
```

```
[30]: car_sales.sum() # get the sum of each column
```

```
[30]: Make          ToyotaHondaToyotaBMWNNissanToyotaHondaHondaToyo...
      Colour         WhiteRedBlueBlackWhiteGreenBlueBlueWhiteWhite
      Odometer (KM)           786014
      Doors             40
      Price        $4,000.00$5,000.00$7,000.00$22,000.00$3,500.00...
      dtype: object
```

```
[31]: car_sales["Doors"].sum() #Access a specific column
```

```
[31]: np.int64(40)
```

```
[32]: len(car_sales) # get the count of rows
```

```
[32]: 10
```

```
[33]: car_sales
```

```
[33]:   Make Colour  Odometer (KM)  Doors      Price
  0  Toyota  White       150043      4  $4,000.00
  1  Honda   Red        87899       4  $5,000.00
  2  Toyota  Blue        32549       3  $7,000.00
  3    BMW  Black        11179       5  $22,000.00
  4  Nissan  White       213095      4  $3,500.00
  5  Toyota  Green       99213       4  $4,500.00
  6  Honda   Blue        45698       4  $7,500.00
  7  Honda   Blue        54738       4  $7,000.00
  8  Toyota  White       60000       4  $6,250.00
  9  Nissan  White       31600       4  $9,700.00
```

## 1.2 View and selectin data

```
[34]: car_sales.head() #top 5 rows quick snapshot
```

```
[34]:   Make Colour  Odometer (KM)  Doors      Price
  0  Toyota  White       150043      4  $4,000.00
  1  Honda   Red        87899       4  $5,000.00
  2  Toyota  Blue        32549       3  $7,000.00
  3    BMW  Black        11179       5  $22,000.00
  4  Nissan  White       213095      4  $3,500.00
```

```
[35]: car_sales.head(7)
```

```
[35]:   Make Colour  Odometer (KM)  Doors      Price
  0  Toyota  White       150043      4  $4,000.00
  1  Honda   Red        87899       4  $5,000.00
  2  Toyota  Blue        32549       3  $7,000.00
```

```
3      BMW  Black      11179      5  $22,000.00
4  Nissan  White     213095      4  $3,500.00
5  Toyota  Green     99213      4  $4,500.00
6   Honda  Blue      45698      4  $7,500.00
```

```
[36]: car_sales.tail() # bottom 5 rows
```

```
[36]:    Make Colour  Odometer (KM)  Doors      Price
5  Toyota  Green     99213      4  $4,500.00
6   Honda  Blue      45698      4  $7,500.00
7   Honda  Blue      54738      4  $7,000.00
8  Toyota  White     60000      4  $6,250.00
9  Nissan  White     31600      4  $9,700.00
```

```
[37]: # .loc & .iloc
```

```
[38]: animals = pd.Series(["cat", "dog", "bird", "panda", "snake"], index = [0,3,9,8,3])
```

```
[39]: animals
```

```
[39]: 0      cat
3      dog
9      bird
8      panda
3      snake
dtype: object
```

```
[40]: animals.loc[3] # getting the index numbers
```

```
[40]: 3      dog
3      snake
dtype: object
```

```
[41]: car_sales.loc[9]
```

```
[41]: Make          Nissan
Colour        White
Odometer (KM)  31600
Doors          4
Price          $9,700.00
Name: 9, dtype: object
```

```
[42]: #iloc refers to position the third element
```

```
[43]: animals.iloc[3]
```

```
[43]: 'panda'
```

```
[44]: car_sales.iloc[2]
```

```
[44]: Make          Toyota
      Colour        Blue
      Odometer (KM) 32549
      Doors          3
      Price         $7,000.00
      Name: 2, dtype: object
```

```
[45]: animals.iloc[:3] # gives us the elements up to position 3
```

```
[45]: 0    cat
      3    dog
      9    bird
      dtype: object
```

```
[46]: car_sales.loc[:3]
```

```
[46]:   Make Colour  Odometer (KM)  Doors      Price
      0  Toyota  White       150043      4  $4,000.00
      1  Honda   Red        87899       4  $5,000.00
      2  Toyota  Blue       32549       3  $7,000.00
      3    BMW   Black      11179       5 $22,000.00
```

```
[47]: car_sales["Make"]
```

```
[47]: 0    Toyota
      1    Honda
      2    Toyota
      3    BMW
      4    Nissan
      5    Toyota
      6    Honda
      7    Honda
      8    Toyota
      9    Nissan
      Name: Make, dtype: object
```

```
[48]: car_sales.Make # works only when in the name is no space
```

```
[48]: 0    Toyota
      1    Honda
      2    Toyota
      3    BMW
      4    Nissan
      5    Toyota
      6    Honda
      7    Honda
```

```
8    Toyota
9    Nissan
Name: Make, dtype: object
```

```
[49]: car_sales[car_sales["Make"] == "Toyota"] # Filtering Data
```

```
[49]:   Make Colour  Odometer (KM)  Doors      Price
0  Toyota  White        150043      4  $4,000.00
2  Toyota  Blue         32549       3  $7,000.00
5  Toyota  Green        99213      4  $4,500.00
8  Toyota  White        60000      4  $6,250.00
```

```
[50]: car_sales[car_sales["Odometer (KM)"] > 100000]
```

```
[50]:   Make Colour  Odometer (KM)  Doors      Price
0  Toyota  White        150043      4  $4,000.00
4  Nissan  White        213095      4  $3,500.00
```

```
[51]: pd.crosstab(car_sales["Make"], car_sales["Doors"]) # Cross Tab columns
```

```
[51]: Doors    3    4    5
      Make
      BMW     0    0    1
      Honda   0    3    0
      Nissan  0    2    0
      Toyota  1    3    0
```

```
[52]: #Group by
```

```
[53]: car_sales.groupby(["Make"]).mean(numeric_only = True)
```

```
[53]:   Odometer (KM)  Doors
      Make
      BMW     11179.000000  5.00
      Honda   62778.333333  4.00
      Nissan  122347.500000  4.00
      Toyota  85451.250000  3.75
```

```
[54]: car_sales.groupby(["Colour"])
```

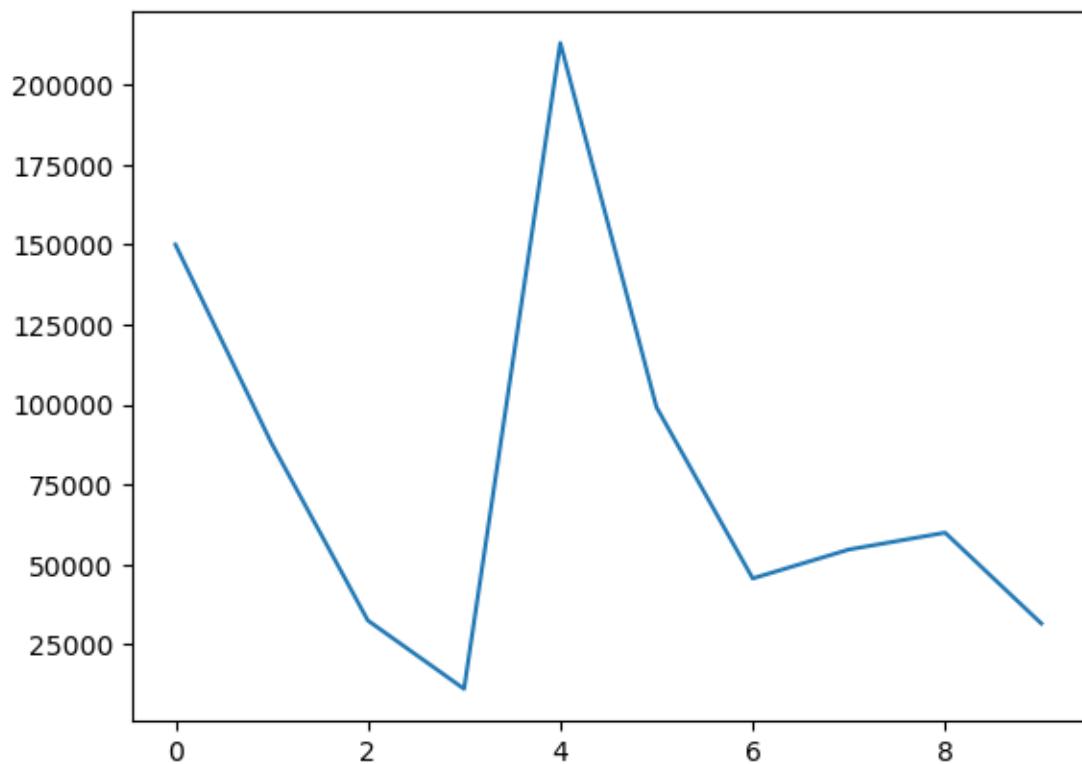
```
[54]: <pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001BF82065310>
```

```
[55]: # Visualization
```

```
[56]: import matplotlib.pyplot as plt
```

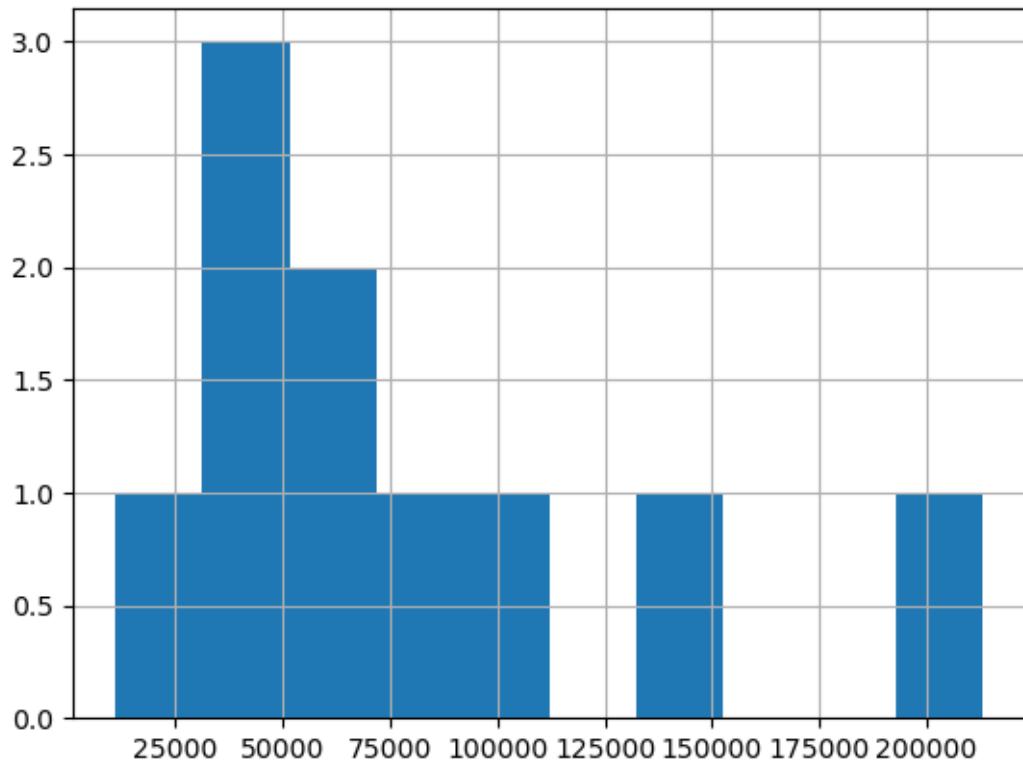
```
[57]: car_sales["Odometer (KM)"].plot()
```

```
[57]: <Axes: >
```



```
[58]: car_sales["Odometer (KM)"].hist()
```

```
[58]: <Axes: >
```



```
[59]: car_sales["Price"].dtype
```

```
[59]: dtype('O')
```

```
[60]: # How to change the datatype of a column
```

```
[61]: car_sales["Price"] = car_sales["Price"].str.replace("\$\,\.\]", "", regex = True).  
      ↪astype(int)
```

```
[62]: ## ignore the cents!
```

```
[63]: car_sales["Price"] = car_sales["Price"].astype(str).str[:-2].astype(int)
```

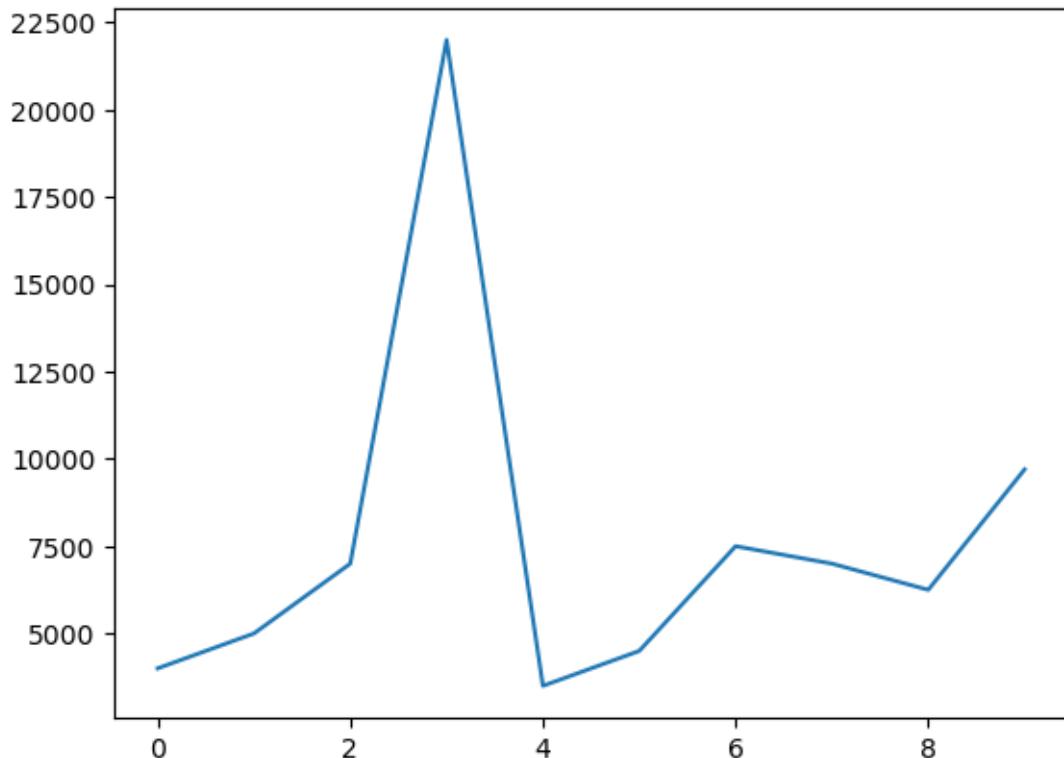
```
[64]: car_sales["Price"]
```

```
[64]: 0    4000  
1    5000  
2    7000  
3   22000  
4   3500  
5   4500  
6   7500
```

```
7      7000
8      6250
9      9700
Name: Price, dtype: int64
```

```
[65]: car_sales["Price"].plot()
```

```
[65]: <Axes: >
```



## 2 Manipulating Data

```
[66]: car_sales["Make"].str.lower() # access the str value of a column with the
    ↪property str
```

```
[66]: 0      toyota
1      honda
2      toyota
3      bmw
4      nissan
5      toyota
6      honda
```

```
7      honda
8      toyota
9      nissan
Name: Make, dtype: object
```

```
[67]: car_sales["Make"] = car_sales["Make"].str.lower() # assign new value the column
```

```
[68]: car_sales
```

```
[68]:   Make Colour  Odometer (KM)  Doors  Price
0  toyota  White       150043        4    4000
1  honda    Red        87899        4    5000
2  toyota   Blue       32549         3    7000
3    bmw  Black       11179         5   22000
4  nissan  White      213095        4    3500
5  toyota  Green       99213        4    4500
6  honda   Blue       45698        4    7500
7  honda   Blue       54738        4    7000
8  toyota  White      60000        4    6250
9  nissan  White      31600        4    9700
```

```
[69]: #Working with missing Data NaN
```

```
[70]: car_sales_missing = pd.read_csv("car-sales-missing-data.csv")
```

```
[71]: car_sales_missing
```

```
[71]:   Make Colour  Odometer  Doors  Price
0  Toyota  White  150043.0    4.0 $4,000
1  Honda   Red   87899.0    4.0 $5,000
2  Toyota   Blue     NaN     3.0 $7,000
3    BMW  Black  11179.0    5.0 $22,000
4  Nissan  White  213095.0   4.0 $3,500
5  Toyota  Green     NaN    4.0 $4,500
6  Honda    NaN     NaN    4.0 $7,500
7  Honda   Blue     NaN    4.0     NaN
8  Toyota  White  60000.0     NaN     NaN
9    NaN  White  31600.0    4.0 $9,700
```

```
[72]: car_sales_missing["Odometer"] = car_sales_missing["Odometer"].
    ↪fillna(car_sales_missing["Odometer"].mean(numeric_only=True)) # fill NAN
    ↪values
```

```
[73]: car_sales_missing
```

```
[73]:   Make Colour  Odometer  Doors  Price
0  Toyota  White  150043.000000    4.0 $4,000
1  Honda   Red   87899.000000    4.0 $5,000
```

```

2 Toyota Blue 92302.666667 3.0 $7,000
3 BMW Black 11179.000000 5.0 $22,000
4 Nissan White 213095.000000 4.0 $3,500
5 Toyota Green 92302.666667 4.0 $4,500
6 Honda NaN 92302.666667 4.0 $7,500
7 Honda Blue 92302.666667 4.0 NaN
8 Toyota White 60000.000000 NaN NaN
9 NaN White 31600.000000 4.0 $9,700

```

```
[74]: car_sales_missing = car_sales_missing.dropna() #Drop all NAN Rows
```

```
[75]: car_sales_missing
```

```
[75]:      Make Colour        Odometer  Doors   Price
0  Toyota  White  150043.000000  4.0  $4,000
1  Honda   Red   87899.000000  4.0  $5,000
2  Toyota  Blue  92302.666667  3.0  $7,000
3  BMW    Black  11179.000000  5.0  $22,000
4  Nissan  White 213095.000000  4.0  $3,500
5  Toyota  Green 92302.666667  4.0  $4,500
```

```
[76]: car_sales_missing.to_csv("car_sales_missing_dropped")
```

```
[77]: #Column from series
seats_column = pd.Series([5,5,5,5,5])

# New Column calles seats
car_sales["Seats"] = seats_column
car_sales
```

```
[77]:      Make Colour  Odometer (KM)  Doors  Price  Seats
0  toyota  White     150043       4  4000   5.0
1  honda   Red      87899        4  5000   5.0
2  toyota  Blue     32549        3  7000   5.0
3  bmw    Black     11179        5  22000  5.0
4  nissan  White    213095       4  3500   5.0
5  toyota  Green    99213        4  4500   NaN
6  honda   Blue     45698        4  7500   NaN
7  honda   Blue     54738        4  7000   NaN
8  toyota  White    60000        4  6250   NaN
9  nissan  White    31600        4  9700   NaN
```

```
[78]: car_sales["Seats"] = car_sales["Seats"].fillna(5)
```

```
[79]: car_sales
```

```
[79]:      Make Colour Odometer (KM) Doors Price Seats
0  toyota  White     150043     4  4000   5.0
1  honda   Red       87899     4  5000   5.0
2  toyota  Blue      32549     3  7000   5.0
3  bmw    Black      11179     5  22000  5.0
4  nissan  White     213095    4  3500   5.0
5  toyota  Green     99213     4  4500   5.0
6  honda   Blue      45698     4  7500   5.0
7  honda   Blue      54738     4  7000   5.0
8  toyota  White     60000     4  6250   5.0
9  nissan  White     31600     4  9700   5.0
```

```
[80]: #Column from Python List
fuel_economy = [7.5,9.5,5.6,7.8,9.8,8.8,7.5,5.5,8.7,6] #the amount of the values
           ↪must match the amount of the rows!
car_sales["Fuel per 100KM"] = fuel_economy
```

```
[81]: car_sales
```

```
[81]:      Make Colour Odometer (KM) Doors Price Seats Fuel per 100KM
0  toyota  White     150043     4  4000   5.0        7.5
1  honda   Red       87899     4  5000   5.0        9.5
2  toyota  Blue      32549     3  7000   5.0        5.6
3  bmw    Black      11179     5  22000  5.0        7.8
4  nissan  White     213095    4  3500   5.0        9.8
5  toyota  Green     99213     4  4500   5.0        8.8
6  honda   Blue      45698     4  7500   5.0        7.5
7  honda   Blue      54738     4  7000   5.0        5.5
8  toyota  White     60000     4  6250   5.0        8.7
9  nissan  White     31600     4  9700   5.0        6.0
```

```
[82]: car_sales["Total fuel used"] = car_sales["Odometer (KM)"]/100 * car_sales["Fuel per 100KM"]
           ↪per 100KM" ] # generate a new column with a calculation
car_sales
```

```
[82]:      Make Colour Odometer (KM) Doors Price Seats Fuel per 100KM \
0  toyota  White     150043     4  4000   5.0        7.5
1  honda   Red       87899     4  5000   5.0        9.5
2  toyota  Blue      32549     3  7000   5.0        5.6
3  bmw    Black      11179     5  22000  5.0        7.8
4  nissan  White     213095    4  3500   5.0        9.8
5  toyota  Green     99213     4  4500   5.0        8.8
6  honda   Blue      45698     4  7500   5.0        7.5
7  honda   Blue      54738     4  7000   5.0        5.5
8  toyota  White     60000     4  6250   5.0        8.7
9  nissan  White     31600     4  9700   5.0        6.0
```

```
Total fuel used
0      11253.225
1      8350.405
2      1822.744
3      871.962
4      20883.310
5      8730.744
6      3427.350
7      3010.590
8      5220.000
9      1896.000
```

```
[83]: # create a column from a single value
car_sales["Number of Wheels"] = 4
car_sales
```

```
[83]:      Make Colour Odometer (KM)  Doors  Price  Seats  Fuel per 100KM \
0  toyota  White       150043      4    4000    5.0      7.5
1  honda   Red        87899       4    5000    5.0      9.5
2  toyota  Blue       32549       3    7000    5.0      5.6
3  bmw    Black       11179       5   22000    5.0      7.8
4  nissan  White      213095      4    3500    5.0      9.8
5  toyota  Green      99213       4    4500    5.0      8.8
6  honda   Blue       45698       4    7500    5.0      7.5
7  honda   Blue       54738       4    7000    5.0      5.5
8  toyota  White      60000       4   6250    5.0      8.7
9  nissan  White      31600       4   9700    5.0      6.0
```

```
Total fuel used  Number of Wheels
0      11253.225      4
1      8350.405      4
2      1822.744      4
3      871.962      4
4      20883.310      4
5      8730.744      4
6      3427.350      4
7      3010.590      4
8      5220.000      4
9      1896.000      4
```

```
[84]: car_sales["Passed the road safety"] = True
car_sales
```

```
[84]:      Make Colour Odometer (KM)  Doors  Price  Seats  Fuel per 100KM \
0  toyota  White       150043      4    4000    5.0      7.5
1  honda   Red        87899       4    5000    5.0      9.5
2  toyota  Blue       32549       3    7000    5.0      5.6
```

```

3     bmw  Black      11179      5  22000      5.0       7.8
4  nissan  White     213095      4   3500      5.0       9.8
5  toyota  Green     99213       4   4500      5.0       8.8
6   honda  Blue      45698       4   7500      5.0       7.5
7   honda  Blue      54738       4  7000      5.0       5.5
8  toyota  White     60000       4   6250      5.0       8.7
9  nissan  White     31600       4   9700      5.0       6.0

```

	Total fuel used	Number of Wheels	Passed the road safety
0	11253.225	4	True
1	8350.405	4	True
2	1822.744	4	True
3	871.962	4	True
4	20883.310	4	True
5	8730.744	4	True
6	3427.350	4	True
7	3010.590	4	True
8	5220.000	4	True
9	1896.000	4	True

[85]: car\_sales.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Make             10 non-null    object 
 1   Colour           10 non-null    object 
 2   Odometer (KM)    10 non-null    int64  
 3   Doors            10 non-null    int64  
 4   Price            10 non-null    int64  
 5   Seats            10 non-null    float64
 6   Fuel per 100KM  10 non-null    float64
 7   Total fuel used 10 non-null    float64
 8   Number of Wheels 10 non-null    int64  
 9   Passed the road safety 10 non-null    bool  
dtypes: bool(1), float64(3), int64(4), object(2)
memory usage: 862.0+ bytes

```

[86]: # Remove a column

[87]: car\_sales = car\_sales.drop("Total fuel used", axis = 1) # axis 1 because of the ↵anatomy

[88]: car\_sales

```
[88]:      Make Colour  Odometer (KM)  Doors  Price  Seats  Fuel per 100KM \
0  toyota  White       150043       4    4000    5.0      7.5
1  honda   Red        87899        4    5000    5.0      9.5
2  toyota  Blue       32549        3    7000    5.0      5.6
3  bmw    Black       11179        5   22000    5.0      7.8
4  nissan  White      213095       4    3500    5.0      9.8
5  toyota  Green      99213        4    4500    5.0      8.8
6  honda   Blue       45698        4    7500    5.0      7.5
7  honda   Blue       54738        4    7000    5.0      5.5
8  toyota  White      60000        4   6250    5.0      8.7
9  nissan  White      31600        4   9700    5.0      6.0
```

	Number of Wheels	Passed the road safety
0	4	True
1	4	True
2	4	True
3	4	True
4	4	True
5	4	True
6	4	True
7	4	True
8	4	True
9	4	True

```
[89]: car_sales=car_sales.sample(frac=1) # mix the order of the columns, frac is the
     ↪fraction 1 = all 0.5 the half
```

```
[90]: car_sales
```

```
[90]:      Make Colour  Odometer (KM)  Doors  Price  Seats  Fuel per 100KM \
4  nissan  White      213095       4    3500    5.0      9.8
2  toyota  Blue       32549        3    7000    5.0      5.6
7  honda   Blue       54738        4    7000    5.0      5.5
8  toyota  White      60000        4   6250    5.0      8.7
6  honda   Blue       45698        4    7500    5.0      7.5
9  nissan  White      31600        4   9700    5.0      6.0
5  toyota  Green      99213        4    4500    5.0      8.8
1  honda   Red        87899        4    5000    5.0      9.5
3  bmw    Black       11179        5   22000    5.0      7.8
0  toyota  White       150043       4    4000    5.0      7.5
```

	Number of Wheels	Passed the road safety
4	4	True
2	4	True
7	4	True
8	4	True
6	4	True

```

9          4           True
5          4           True
1          4           True
3          4           True
0          4           True

```

```
[91]: # Only select 20% of the data
car_sales.sample(frac = 0.2)
```

```
[91]:      Make Colour Odometer (KM)  Doors  Price  Seats  Fuel per 100KM \
2  toyota  Blue        32549       3    7000   5.0      5.6
8  toyota  White       60000       4    6250   5.0      8.7

Number of Wheels  Passed the road safety
2                  4           True
8                  4           True
```

```
[92]: car_sales=car_sales.reset_index(drop = True) # Order by index
```

```
[93]: car_sales
```

```
[93]:      Make Colour Odometer (KM)  Doors  Price  Seats  Fuel per 100KM \
0  nissan  White       213095      4    3500   5.0      9.8
1  toyota  Blue        32549       3    7000   5.0      5.6
2  honda   Blue        54738       4    7000   5.0      5.5
3  toyota  White       60000       4    6250   5.0      8.7
4  honda   Blue        45698       4    7500   5.0      7.5
5  nissan  White       31600       4    9700   5.0      6.0
6  toyota  Green       99213       4    4500   5.0      8.8
7  honda   Red         87899       4    5000   5.0      9.5
8  bmw    Black        11179       5   22000   5.0      7.8
9  toyota  White       150043      4    4000   5.0      7.5

Number of Wheels  Passed the road safety
0                  4           True
1                  4           True
2                  4           True
3                  4           True
4                  4           True
5                  4           True
6                  4           True
7                  4           True
8                  4           True
9                  4           True
```

```
[96]: car_sales["Odometer (KM)"] = car_sales["Odometer (KM)"].apply(lambda x: x/1.6) # ↵
      ↵Change km to Miles
```

```
[97]: car_sales
```

```
[97]:      Make Colour  Odometer (KM)  Doors  Price  Seats  Fuel per 100KM \
0  nissan  White       1331.84375     4    3500    5.0        9.8
1  toyota   Blue        203.43125     3    7000    5.0        5.6
2  honda    Blue        342.11250     4    7000    5.0        5.5
3  toyota   White       375.00000     4    6250    5.0        8.7
4  honda    Blue        285.61250     4    7500    5.0        7.5
5  nissan  White       197.50000     4    9700    5.0        6.0
6  toyota   Green       620.08125     4    4500    5.0        8.8
7  honda    Red         549.36875     4    5000    5.0        9.5
8  bmw     Black        69.86875      5   22000    5.0        7.8
9  toyota   White       937.76875     4    4000    5.0        7.5

      Number of Wheels  Passed the road safety
0                      4                  True
1                      4                  True
2                      4                  True
3                      4                  True
4                      4                  True
5                      4                  True
6                      4                  True
7                      4                  True
8                      4                  True
9                      4                  True
```

```
[103]: car_sales = car_sales.rename(columns={"Odometer (KM)": "Odometer (Miles)"}) #  
      ↪Change the name of a column
```

```
[104]: car_sales
```

```
[104]:      Make Colour  Odometer (Miles)  Doors  Price  Seats  Fuel per 100KM \
0  nissan  White       1331.84375     4    3500    5.0        9.8
1  toyota   Blue        203.43125     3    7000    5.0        5.6
2  honda    Blue        342.11250     4    7000    5.0        5.5
3  toyota   White       375.00000     4    6250    5.0        8.7
4  honda    Blue        285.61250     4    7500    5.0        7.5
5  nissan  White       197.50000     4    9700    5.0        6.0
6  toyota   Green       620.08125     4    4500    5.0        8.8
7  honda    Red         549.36875     4    5000    5.0        9.5
8  bmw     Black        69.86875      5   22000    5.0        7.8
9  toyota   White       937.76875     4    4000    5.0        7.5

      Number of Wheels  Passed the road safety
0                      4                  True
1                      4                  True
2                      4                  True
```

3	4	True
4	4	True
5	4	True
6	4	True
7	4	True
8	4	True
9	4	True

[ ]: