To extract bottom records.

d2.tail() d2 → var-name

d2.sample(6) → To extract random function.

d2.columns → To know the column names.

d2.dtypes → column names with data types.

d2.isna().sum → To know how many null Values.

<09/10/2025, Thursday>.

d2.info()

*var_name.col_name [d2.manufacturer]

It is in series format means having either single row or column.

* To get in dataframe format, which include column name also.

d2 [['col_name']]

d2 [['col_name1', 'col_name2', 'col_name3']]

→To extract one (or) more columns.

To create a series data manually

1) pd.Series([1, 2, 3, 4])

2) S = {'id':[1, 2, 3, 4], 'name':['Hardu', 'midhun', 'viji', 'Lekha'],
'Sal':['50k', '20k', '40k', '30k']}

S

3) To print in dataframe format.

data = pd.DataFrame(S)

data.

Accessing a particular range of columns.

If i want to extract all columns except one column means, i'll go with delete that one colm temporarily.

Ex: d2.drop('col_name', axis = 1)

(or) d2.iloc[:, : -1]

(or) d2.loc[:, 'manufa/col_name from': 'col_name upto']

To permanently delete the features means, assigning to the same variable.

d2 = d2.drop(['wt', 'hp'], axis = 1)

d2.

To delete particular rows.

d2 = d2.drop([1, 3])

d2.

<iloc, loc dataframe slicing>

var.iloc[rows pos, Col pos]

var.iloc[row start val : row stop val : row inc/dec, col start val : col stop val : col inc/dec]

1) WAPT fetch all the rows with first 5 columns.

d2.iloc[:, :5]

2) print entire table using iloc

d2.iloc[:, :]

3) odd indexed rows with even indexed columns.

d2.iloc[1::2, ::2]