

```
# !pip install pandas # oruvela pandas install pannalena indha command use pan:
```

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

Pandas = Work with tables (rows & columns) easily

Used for:

CSV / Excel data

Cleaning data

ML preprocessing

Analysis

NumPy	Pandas
Arrays	Tables
Fast math	Data handling
No column names	Column names
Fixed type	Mixed types

```
# pandas use panti, table mari structures namalala manipulate panamudium ,
# columns names , rows ku names irukum
```

Series (1D – single column)

-> single column mattume irukum indha table la , idhuku peru dha series

```
s = pd.Series([10, 20, 30])
print(s) # oreya oru column , adhula irukara elements 10,20,30 , left pakathula
```

```
0    10
1    20
2    30
dtype: int64
```

2D Table

DataFrame nu soluvanga

rows and columns with column names and row names irukum

Dataframes create pandradhuku naraya ways iriki, sila pakalam ipo

1st way of creating dataframe

pudhu data use paniti dataframe create panaporom

```
chem_df = pd.read_csv("https://raw.githubusercontent.com/rames4498/workshop_t...")
chem_df
```

	ID	Name	InChI	InChIKey
0	A-3	N,N,N-trimethyloctadecan-1-aminium bromide	InChI=1S/C21H46N.BrH/c1-5-6-7-8-9-10-11-12-13-...	SZEMGTQCPRNXI UHFFFAOYSA
1	A-4	Benzo[cd]indol-2(1H)-one	InChI=1S/C11H7NO/c13-11-8-5-1-3-7-4-2-6-9(12-1...	GPYLCFQEKPUWI UHFFFAOYSA
2	A-5	4-chlorobenzaldehyde	InChI=1S/C7H5ClO/c8-7-3-1-6(5-9)2-4-7/h1-5H	AVPYQKSLYISFF UHFFFAOYSA
3	A-8	zinc bis[2-hydroxy-3,5-bis(1-phenylethyl)benzo...	InChI=1S/2C23H22O3.Zn/c2*1-15(17-9-5-3-6-10-17...	XTUPUYCJWKHGS UHFFFAOYSA
4	A-9	4-({4-[bis(oxiran-2-ylmethyl)amino]phenyl)meth...	InChI=1S/C25H30N2O4/c1-5-20(26(10-22-14-28-22)...	FAUAZXVRLVIAI UHFFFAOYSA
...	...	...	...	...
9977	I-84	tetracaine	InChI=1S/C15H24N2O2/c1-4-5-10-16-14-8-6-13(7-9...	GKCBAGFKIBET UHFFFAOYSA
9978	I-85	tetracycline	InChI=1S/C22H24N2O8/c1-21(31)8-5-4-6-11(25)12(...	OFVLGDICTFRJM WESIUVDSSE
9979	I-86	thymol	InChI=1S/C10H14O/c1-7(2)9-5-4-8(3)6-10(9)11/h4...	MGSRCZKZVOBK UHFFFAOYSA
9980	I-93	verapamil	InChI=1S/C27H38N2O4/c1-20(2)27(19-28,22-10-12-...	SGTNSNPWRIOYI UHFFFAOYSA
9981	I-94	warfarin	InChI=1S/C19H16O4/c1-12(20)11-15(13-7-3-2-4-8-...	PJVWKTQKMONH UHFFFAOYSA

9982 rows x 26 columns

```
data = {
    "Name": ["Ram", "Sam", "Bahubali", "RRR", "Samantha", "Ramesh"],
    "Age": [25, 30, 22, 67, 89, 78],
}
```

```
"Salary": [50000, 60000, 45000, 99000,89000,25000]
}
```

```
# inge Name , Age , Salary moonu column names ,
```

```
#pd.DataFrame use paniti mela irukara object , oru data object table ah matham
df = pd.DataFrame(data)
print(df) # inge df andradhu namaloda dataframe name
```

```
# output la oru table create ayindhirkum
```

	Name	Age	Salary
0	Ram	25	50000
1	Sam	30	60000
2	Bahubali	22	45000
3	RRR	67	99000
4	Samantha	89	89000
5	Ramesh	78	25000

```
df.head()      # first 5 rows mattume varum
```

	Name	Age	Salary
<b>0</b>	Ram	25	50000
<b>1</b>	Sam	30	60000
<b>2</b>	Bahubali	22	45000
<b>3</b>	RRR	67	99000
<b>4</b>	Samantha	89	89000

```
df.head(2)
```

	Name	Age	Salary
<b>0</b>	Ram	25	50000
<b>1</b>	Sam	30	60000

```
df.tail()      # last 5 rows
```

	Name	Age	Salary
1	Sam	30	60000
2	Bahubali	22	45000
3	RRR	67	99000
4	Samantha	89	89000
5	Ramesh	78	25000

```
df.tail(3) # last 3 rows
```

	Name	Age	Salary
3	RRR	67	99000
4	Samantha	89	89000
5	Ramesh	78	25000

```
df.shape      # (rows, columns)
# 6 rows, 3 columns
```

```
(6, 3)
```

```
df.columns     # column names
```

```
Index(['Name', 'Age', 'Salary'], dtype='object')
```

```
df.info()      # data types, dataframe pathi yella information info function na
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0   Name    6 non-null         object
1   Age     6 non-null         int64
2   Salary  6 non-null         int64
dtypes: int64(2), object(1)
memory usage: 276.0+ bytes
```

2nd way of creating dataframe

pudhu data use paniti dataframe create panaporom

```
from numpy.random import randn
np.random.seed(103)      # nama ipo random decimal values 5 rows 5 columns use paniti
                        # A B C D namaloda dataframe rows peru,,,,, namaloda col
dframe=pd.DataFrame(randn(5,5),['A','B','C','D','E'],['U','V','W','X','Y'])
dframe
```

	U	V	W	X	Y
A	-1.249278	-0.260331	0.383793	-0.385461	-1.085137
B	2.327219	0.430793	0.432316	-0.980011	-0.631965
C	0.577442	-0.124758	0.978948	1.594922	-1.201945
D	-1.376369	1.054346	-0.038853	0.680286	1.329175
E	1.283450	-1.758254	0.614306	1.516358	-0.195977

dframe

	U	V	W	X	Y
A	-1.249278	-0.260331	0.383793	-0.385461	-1.085137
B	2.327219	0.430793	0.432316	-0.980011	-0.631965
C	0.577442	-0.124758	0.978948	1.594922	-1.201945
D	-1.376369	1.054346	-0.038853	0.680286	1.329175
E	1.283450	-1.758254	0.614306	1.516358	-0.195977

```
dframe>0      # dframe andra dataframe la values greater than zero yengalam irul
```

	U	V	W	X	Y
A	False	False	True	False	False
B	True	True	True	False	False
C	True	False	True	True	False
D	False	True	False	True	True
E	True	False	True	True	False

```
dframe['X']>0      # X column la values greater than zero yengalam irukardho anga
```

**X**

**A** False

**B** False

**C** True

**D** True

**E** True

**dtype:** bool

```
dframe[dframe['X']>0]      # X column la yengalam grater than 0 irukardho andha w
```

	U	V	W	X	Y
C	0.577442	-0.124758	0.978948	1.594922	-1.201945
D	-1.376369	1.054346	-0.038853	0.680286	1.329175
E	1.283450	-1.758254	0.614306	1.516358	-0.195977

dframe

	U	V	W	X	Y
A	-1.249278	-0.260331	0.383793	-0.385461	-1.085137
B	2.327219	0.430793	0.432316	-0.980011	-0.631965
C	0.577442	-0.124758	0.978948	1.594922	-1.201945
D	-1.376369	1.054346	-0.038853	0.680286	1.329175
E	1.283450	-1.758254	0.614306	1.516358	-0.195977

```
dframe['X']+dframe['Y']      # rendu colums add pandrom
```

```
0
A -1.470598
B -1.611977
C 0.392977
D 2.009461
E 1.320381
```

```
dtype: float64
```

```
dframe['new'] = dframe['X']+dframe['Y'] # X and Y columns add paniti pudhu col
dframe
```

	U	V	W	X	Y	new
A	-1.249278	-0.260331	0.383793	-0.385461	-1.085137	-1.470598
B	2.327219	0.430793	0.432316	-0.980011	-0.631965	-1.611977
C	0.577442	-0.124758	0.978948	1.594922	-1.201945	0.392977
D	-1.376369	1.054346	-0.038853	0.680286	1.329175	2.009461
E	1.283450	-1.758254	0.614306	1.516358	-0.195977	1.320381

```
sherlock=dframe[dframe['U']>0]
```

```
sherlock
```

	U	V	W	X	Y	new
B	2.327219	0.430793	0.432316	-0.980011	-0.631965	-1.611977
C	0.577442	-0.124758	0.978948	1.594922	-1.201945	0.392977
E	1.283450	-1.758254	0.614306	1.516358	-0.195977	1.320381

```
sherlock['X']
```

	X
<b>B</b>	-0.980011
<b>C</b>	1.594922
<b>E</b>	1.516358

**dtype:** float64

dframe[dframe['U']>0]['X']

	X
<b>B</b>	-0.980011
<b>C</b>	1.594922
<b>E</b>	1.516358

**dtype:** float64

dframe[dframe['U']>0][['X','Y']]

	X	Y
<b>B</b>	-0.980011	-0.631965
<b>C</b>	1.594922	-1.201945
<b>E</b>	1.516358	-0.195977

dframe.reset\_index()

	index	U	V	W	X	Y	new
<b>0</b>	A	-1.249278	-0.260331	0.383793	-0.385461	-1.085137	-1.470598
<b>1</b>	B	2.327219	0.430793	0.432316	-0.980011	-0.631965	-1.611977
<b>2</b>	C	0.577442	-0.124758	0.978948	1.594922	-1.201945	0.392977
<b>3</b>	D	-1.376369	1.054346	-0.038853	0.680286	1.329175	2.009461
<b>4</b>	E	1.283450	-1.758254	0.614306	1.516358	-0.195977	1.320381

dframe

	U	V	W	X	Y	new
A	-1.249278	-0.260331	0.383793	-0.385461	-1.085137	-1.470598
B	2.327219	0.430793	0.432316	-0.980011	-0.631965	-1.611977
C	0.577442	-0.124758	0.978948	1.594922	-1.201945	0.392977
D	-1.376369	1.054346	-0.038853	0.680286	1.329175	2.009461
E	1.283450	-1.758254	0.614306	1.516358	-0.195977	1.320381

```
dframe[(dframe['U']>0) | (dframe['W']>0)]
# U column and W column rendula yedhavadhu greater than 0 irundha , andha frame
```

	U	V	W	X	Y	new
A	-1.249278	-0.260331	0.383793	-0.385461	-1.085137	-1.470598
B	2.327219	0.430793	0.432316	-0.980011	-0.631965	-1.611977
C	0.577442	-0.124758	0.978948	1.594922	-1.201945	0.392977
E	1.283450	-1.758254	0.614306	1.516358	-0.195977	1.320381

```
dframe[(dframe['U']>0) & (dframe['W']>0)]
# U column and W column rendume greater than 0 irundha , andha frame output va
```

	U	V	W	X	Y	new
B	2.327219	0.430793	0.432316	-0.980011	-0.631965	-1.611977
C	0.577442	-0.124758	0.978948	1.594922	-1.201945	0.392977
E	1.283450	-1.758254	0.614306	1.516358	-0.195977	1.320381

```
# df andra dataframe la 50000 ku adhigama salary yaru vangurangalo kadupudikal
df['Salary']>50000
```

### Salary

0 False  
1 True  
2 False  
3 True  
4 True  
5 False

**dtype:** bool

```
df[df['Salary']>50000] # ivana moonu peru 50000 ku mela salary vanguranga
```

	Name	Age	Salary
1	Sam	30	60000
3	RRR	67	99000
4	Samantha	89	89000

```
#highest salary yaru vangurangalo kanudpidukalam  
df['Salary'].max() # highest salary 99000 ,  
#avangaloda peru ipo nama kandupidukanum
```

99000

```
df[df['Salary'].max() == df['Salary']] # yengalam adhigama salary match agudho
```

	Name	Age	Salary
3	RRR	67	99000

```
df[df['Salary'].min() == df['Salary']] # yengalam kammiana salary match agudho
```

	Name	Age	Salary
5	Ramesh	78	25000

```
# oldest person yaro kandupudikalam  
df[df['Age'].max() == df['Age']]
```

	Name	Age	Salary
4	Samantha	89	89000

```
# youngest person yaro kandupudikalam  
df[df['Age'].min() == df['Age']]
```

	Name	Age	Salary
2	Bahubali	22	45000

```
df['Age']<50
```

[Show hidden output](#)

```
df[df['Age']<50]['Salary'].min()
```

```
45000
```

```
df['Age']<50
```

[Show hidden output](#)

```
dummy = df[df['Age']<50]  
dummy
```

	Name	Age	Salary
0	Ram	25	50000
1	Sam	30	60000
2	Bahubali	22	45000

```
dummy['Salary'] == dummy['Salary'].min()
```

	Salary
0	False
1	False
2	True

```
dtype: bool
```

```
dummy[dummy['Salary'] == dummy['Salary'].min()]
```

	Name	Age	Salary
2	Bahubali	22	45000

```
df[df['Age']<50][['Name','Age']]
```

	Name	Age
0	Ram	25
1	Sam	30
2	Bahubali	22

```
new_df = df[df['Age']<50]  
new_df
```

	Name	Age	Salary
0	Ram	25	50000
1	Sam	30	60000
2	Bahubali	22	45000

```
new_df[new_df['Salary'].min() == new_df['Salary']]
```

	Name	Age	Salary
2	Bahubali	22	45000

```
# Creating simple dataset  
data = {  
    "Customer_ID": [101,102,103,104,105,106,107,108,109,110],  
    "Product_Category": ["Fruits","Vegetables","Fruits","Dairy","Snacks",  
                          "Dairy","Snacks","Fruits","Vegetables","Snacks"],  
    "Payment_Method": ["Cash","Card","Card","Cash","UPI",  
                       "UPI","Cash","Card","Cash","UPI"],  
    "Quantity": [5,3,6,2,8,1,7,4,2,9],  
    "Total_Bill": [250,150,300,120,400,80,350,220,130,450]  
}  
  
df = pd.DataFrame(data)  
  
df
```

	Customer_ID	Product_Category	Payment_Method	Quantity	Total_Bill
0	101	Fruits	Cash	5	250
1	102	Vegetables	Card	3	150
2	103	Fruits	Card	6	300
3	104	Dairy	Cash	2	120
4	105	Snacks	UPI	8	400
5	106	Dairy	UPI	1	80
6	107	Snacks	Cash	7	350
7	108	Fruits	Card	4	220
8	109	Vegetables	Cash	2	130
9	110	Snacks	UPI	9	450

```
df.head()
```

	Customer_ID	Product_Category	Payment_Method	Quantity	Total_Bill
0	101	Fruits	Cash	5	250
1	102	Vegetables	Card	3	150
2	103	Fruits	Card	6	300
3	104	Dairy	Cash	2	120
4	105	Snacks	UPI	8	400

```
df.info()
```

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

```
df["Product_Category"].value_counts()
```

	count
Product_Category	
Fruits	3
Snacks	3
Vegetables	2
Dairy	2

**dtype:** int64

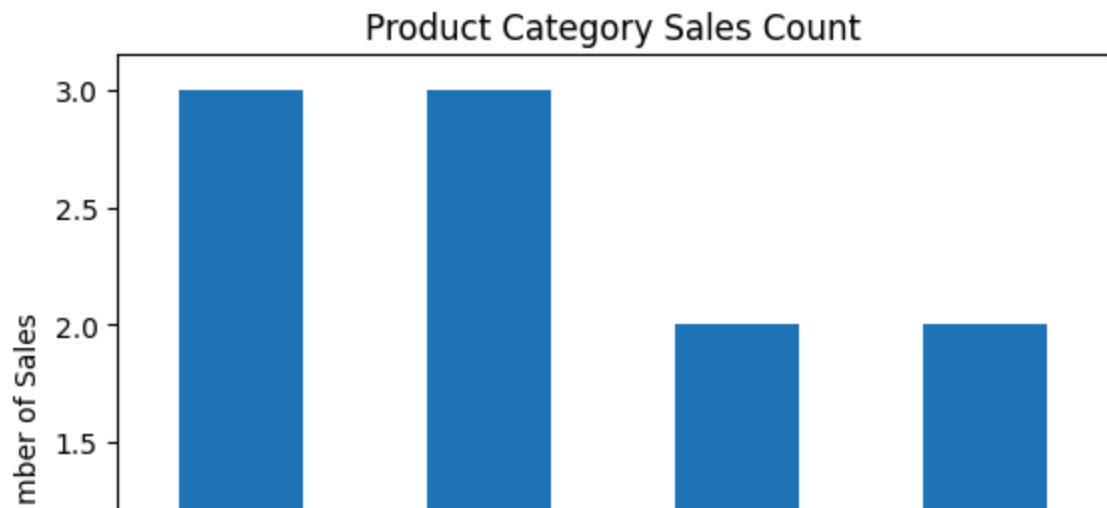
`df.describe()`

	Customer_ID	Quantity	Total_Bill
<b>count</b>	10.00000	10.000000	10.000000
<b>mean</b>	105.50000	4.700000	245.000000
<b>std</b>	3.02765	2.750757	127.301043
<b>min</b>	101.00000	1.000000	80.000000
<b>25%</b>	103.25000	2.250000	135.000000
<b>50%</b>	105.50000	4.500000	235.000000
<b>75%</b>	107.75000	6.750000	337.500000
<b>max</b>	110.00000	9.000000	450.000000

```
import matplotlib.pyplot as plt

df["Product_Category"].value_counts().plot(kind="bar")

plt.title("Product Category Sales Count")
plt.xlabel("Category")
plt.ylabel("Number of Sales")
plt.show()
```



```
data2 = np.array([
    45, 47, 50, 52, 46, 49, 51, 48, 47, 46,
    300,
    -120,
    53, 54, 48, 49, 50
])
print(data2)
```

```
[ 45  47  50  52  46  49  51  48  47  46 300 -120  53  54
  48  49  50]
```

```
data2.sort()
data2
```

```
array([-120,  45,  46,  46,  47,  47,  48,  48,  49,  49,  50,
         50,  51,  52,  53,  54, 300])
```

```
q3 = np.quantile(data2, 0.75)
q3
```

```
np.float64(51.0)
```

```
q1 = np.quantile(data2, 0.25)
q1
```

```
np.float64(47.0)
```

```
act_q1 = (25/100)
act_q1 = act_q1 *(len(data2)+1)
act_q1 #meaning 5th value
```

```
4.5
```

```
act_q3 = (75/100)
act_q3 = act_q3 *(len(data2)+1)
act_q3 #meaning 14th value
```

```
13.5
```

```
data2
```

```
array([-120,  45,  46,  46,  47,  47,  48,  48,  49,  49,  50,
        50,  51,  52,  53,  54, 300])
```

```
q1_final = data2[4]
```

```
q3_final =data2[13]
```

```
iqr = q3_final -q1_final
iqr
```

```
np.int64(5)
```

```
lower = q1_final - (1.5*iqr)
lower
```

```
np.float64(39.5)
```

```
upper = q3_final + (1.5*iqr)
upper
```

```
np.float64(59.5)
```

```
data2
```

[Show hidden output](#)

```
final_data = []
```

```
for i in data2:
    #print(i)
    if (lower <i < upper ):
        final_data.append(i)
```

```
for i in data2:
    #print(i)
    if (i > lower ) and (i < upper):
        final_data.append(i)
```

final\_data

[Show hidden output](#)

[https://github.com/rames4498/workshop\\_tasks/blob/master/employee\\_data.xlsx](https://github.com/rames4498/workshop_tasks/blob/master/employee_data.xlsx)

[https://raw.githubusercontent.com/rames4498/workshop\\_tasks/master/employee\\_data.xlsx](https://raw.githubusercontent.com/rames4498/workshop_tasks/master/employee_data.xlsx)

```
new_df =pd.read_excel('https://raw.githubusercontent.com/rames4498/workshop_tasks/master/employee_data.xlsx')
```

new\_df.head()

	Employee_ID	Age	Gender	Department	Experience_Years	Salary	Workload
0	1	37	Female	Sales	14.713222	65528.844811	1.0
1	2	34	Female	Marketing	0.502125	67033.267257	1.0
2	3	38	Female	HR	14.443281	41722.069554	1.0
3	4	44	Male	IT	10.111842	250000.000000	1.0
4	5	33	Female	Marketing	6.528949	64507.056816	1.0

```
chem_df = pd.read_csv("https://raw.githubusercontent.com/rames4498/workshop_tasks/master/employee_data.csv")
```

chem\_df.head()

	ID	Name	InChI	InChIKey
0	A-3	N,N,N-trimethyloctadecan-1-aminium bromide	InChI=1S/C21H46N.BrH/c1-5-6-7-8-9-10-11-12-13-...	SZEMGTQCPRNXEG-UHFFFAOYSA-M
1	A-4	Benzo[cd]indol-2(1H)-one	InChI=1S/C11H7NO/c13-11-8-5-1-3-7-4-2-6-9(12-1...	GPYLCFQEKPUWLD-UHFFFAOYSA-N
2	A-5	4-chlorobenzaldehyde	InChI=1S/C7H5ClO/c8-7-3-1-6(5-9)2-4-7/h1-5H	AVPYQKSPLYISFPO-UHFFFAOYSA-N
3	A-8	zinc bis[2-hydroxy-3,5-bis(1-phenylethyl)benzo...	InChI=1S/2C23H22O3.Zn/c2*1-15(17-9-5-3-6-10-17...	XTUPUYCJWKHGSW-UHFFFAOYSA-L
4	A-9	4-({4-[bis(oxiran-2-ylmethyl)amino]phenyl}meth...	InChI=1S/C25H30N2O4/c1-5-20(26(10-22-14-28-22)...	FAUAZXVRLVIARB-UHFFFAOYSA-N

5 rows × 26 columns

```
df = pd.read_excel('/content/employee_data.xlsx')
df.head()
```

	Employee_ID	Age	Gender	Department	Experience_Years	Salary	Wo
0	1	37	Female	Sales	14.713222	65528.844811	
1	2	34	Female	Marketing	0.502125	67033.267257	
2	3	38	Female	HR	14.443281	41722.069554	
3	4	44	Male	IT	10.111842	250000.000000	
4	5	33	Female	Marketing	6.528949	64507.056816	

```
Q1 = np.percentile(data2, 25)
Q3 = np.percentile(data2, 75)
IQR = Q3 - Q1

lower = Q1 - 1.5 * IQR
upper = Q3 + 1.5 * IQR

outliers = data2[(data2 < lower) | (data2 > upper)]
print(outliers)
```

```
[-120  300]
```

```
df.head()
```

	Employee_ID	Age	Gender	Department	Experience_Years	Salary	Wo
0	1	37	Female	Sales	14.713222	65528.844811	
1	2	34	Female	Marketing	0.502125	67033.267257	
2	3	38	Female	HR	14.443281	41722.069554	
3	4	44	Male	IT	10.111842	250000.000000	
4	5	33	Female	Marketing	6.528949	64507.056816	

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Employee_ID                          150 non-null    int64
1   Age                                  150 non-null    int64
2   Gender                               150 non-null    object
3   Department                           150 non-null    object
4   Experience_Years                      150 non-null    float64
5   Salary                               150 non-null    float64
6   Work_Hours_Per_Week                  150 non-null    float64
7   Performance_Rating                   150 non-null    int64
8   Projects_Handled                     150 non-null    int64
dtypes: float64(3), int64(4), object(2)
memory usage: 10.7+ KB
```

```
df.isnull().sum()
```

	0
<b>Employee_ID</b>	0
<b>Age</b>	0
<b>Gender</b>	0
<b>Department</b>	0
<b>Experience_Years</b>	0
<b>Salary</b>	0
<b>Work_Hours_Per_Week</b>	0
<b>Performance_Rating</b>	0
<b>Projects_Handled</b>	0

**dtype:** int64

```
import numpy as np
import pandas as pd

np.random.seed(0)

data = {
    "Age": np.random.normal(30, 5, 120),
    "Salary": np.random.exponential(40000, 120),
    "Study_Hours": np.random.randint(1, 12, 120),
    "Attendance": np.random.randint(50, 100, 120),
    "Marks": np.random.normal(70, 12, 120),
    "Experience_Years": np.random.randint(0, 10, 120),
    "Department": np.random.choice(
        ["CSE", "ECE", "MECH", "CIVIL"], 120
    )
}

df = pd.DataFrame(data)

# Inject outliers
df.loc[4, "Salary"] = 300000
df.loc[15, "Marks"] = 5

# Inject missing values
df.loc[3, "Age"] = np.nan
df.loc[8, "Marks"] = np.nan
```

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 120 entries, 0 to 119
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Age                   119 non-null   float64
```

```
1  Salary      120 non-null  float64
2  Study_Hours 120 non-null  int64
3  Attendance  120 non-null  int64
4  Marks       119 non-null  float64
5  Experience_Years 120 non-null int64
6  Department  120 non-null  object
dtypes: float64(3), int64(3), object(1)
memory usage: 6.7+ KB
```

```
df.isnull().sum()
```

	<b>0</b>
<hr/>	
<b>Age</b>	1
<b>Salary</b>	0
<b>Study_Hours</b>	0
<b>Attendance</b>	0
<b>Marks</b>	1
<b>Experience_Years</b>	0
<b>Department</b>	0