## include library

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

#### How We Can Create Series

```
temp=[4,5,6,78,2]
ser=pd.Series(temp)
print(f'The Table is {ser}')
print(f'The Data Type of The Df is {type(ser)}')
```

#### How We Can Create DataFrame

```
data={
    "Name":["Malek","Ali","Ahmed"],
    "Age": [23,25,28]
df=pd.DataFrame(data)
print(f'The Table is {df}')
print(f'The Data Type of The Df is {type(df)}')
# df.iloc[0]
# df.iloc[0,1]
The Table is
                Name Age
  Malek 23
1
     Ali
           25
2 Ahmed
           28
The Data Type of The Df is <class 'pandas.core.frame.DataFrame'>
```

#### Read Data From JSON File

```
myJsonFile=pd.read_json('output.json')
myJsonFile.head()

Name Age City
0 John 25 New York
1 Alice 30 London
2 Bob 35 Paris
```

## Read Data From XML File

```
myXMLFile=pd.read xml('output.xml')
myJsonFile.head()
    Name Age
                  City
0
    John
         25
             New York
1
  Alice
          30
              London
2
           35
                 Paris
     Bob
```

#### Read Data From Text File

```
myTxtFile = pd.read_fwf('data.txt')
print(myTxtFile)

John 25 170
0 Alice 28 165
1 Bob 30 180
```

include file

## Read Data From CSV File

	reda Bata i rom es vinte					
	<pre>dataset=pd.read_csv('students.csv') print(dataset)</pre>					
	sex	<pre>race_ethnicity</pre>	parental_level_of_education	lunch		
0	female	group B	bachelor's degree	standard		
1	female	group C	some college	standard		
2	female	NaN	master's degree	standard		
3	male	group A	associate's degree	free/reduced		
4	male	group C	NaN	standard		
20999	female	group E	master's degree	standard		
21000	male	group C	high school	free/reduced		
21001	female	group C	high school	free/reduced		
21002	female	group D	some college	standard		

21003	female	group D		some college	free/reduced
		J ,		J	
	test_prepa	ration_course	math digree	reading digree	e writing
score					-1
0		none	72	72.0	)
74			60	00 (	`
1 88		completed	69	90.0	J
2		none	90	95.0	)
93		Hone	30	3310	
3		none	47	57.0	)
44					
4		none	76	78.0	)
75					
20999		completed	58	99.0	<b>1</b>
95		compteted	30	99.0	,
21000		none	62	55.0	)
55			02	3310	
21001		completed	49	71.0	)
65					
21002		completed	68	48.0	)
77			77	00.7	
21003		none	77	89.0	)
86					
[21004	1 rows x 8	columnsl			

## **Know Your Data**

## Get The Shape

```
dataset.shape
(21004, 8)
```

#### Get The Head Of The Data Set

```
dataset.head()
     sex race_ethnicity parental_level_of_education
                                                            lunch \
  female
                group B
                                  bachelor's degree
                                                         standard
1
  female
                 group C
                                       some college
                                                         standard
2
  female
                    NaN
                                    master's degree
                                                         standard
                                 associate's degree free/reduced
     male
                 group A
```

4	male	group C				NaN	standa	ard
t	est_prep	paration_course	math	digree	reading	digree	writing	score
0		none		72		72.0		74
1		completed		69		90.0		88
2		none		90		95.0		93
3		none		47		57.0		44
4		none		76		78.0		75
dat	aset.hea	ad ( <mark>20</mark> )						
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19	sex female female male female female male female male female female female female female female female female	race_ethnicity group B group C NaN group A group B group B group B group D group B group C group B group C group C group C group C group A group C group C group C group C group C	parent	associ	nelor's or some conster's or some conster's or some constants or some constants or some conster's or some constants or s	degree ollege degree NaN degree ollege ollege school degree degree school school degree school degree degree	stand stand stand free/redu stand stand free/redu free/redu stand stand stand stand stand stand free/redu free/redu free/redu free/redu free/redu free/redu stand	dard dard dard dard dard dard dard dard
	test_pre	eparation_course	e math	n digree	reading	g digree		
0		none	9	72		72.6	)	74
1		completed	d	69		90.0	)	88
2		none	9	90		95.0	)	93
3		none	9	47		57.0	)	44
4		none	9	76		78.0	)	75
5		none	2	71		83.6	)	78

6	completed	88	95.0	92
7	none	40	43.0	39
8	completed	64	64.0	67
9	none	38	60.0	50
10	none	58	54.0	52
11	none	40	52.0	43
12	none	65	81.0	73
13	completed	78	72.0	70
14	NaN	50	53.0	58
15	none	69	75.0	78
16	none	88	NaN	86
17	none	18	32.0	28
18	completed	46	42.0	46
19	none	54	58.0	61

## Get The Data The Tail of The DataSet

datase	et.tail()				
	sex	race_ethnicity p	arental_level	_of_education	lunch
20999	female	group E	ma	ster's degree	standard
21000	male	group C		high school	free/reduced
21001	female	group C		high school	free/reduced
21002	female	group D		some college	standard
21003	female	group D		some college	free/reduced
	test_pre	paration_course	math digree	reading digre	e writing
score 20999		completed	58	99.	Θ
95 21000		none	62	55.	0

55					
21001	completed	49	71.0		
65					
21002	completed	68	48.0		
77					
21003	none	77	89.0		
86					
dataset.tail(20)					

#### Get The Columns Name

```
dataset.columns
Index(['sex', 'race_ethnicity', 'parental_level_of_education',
    'lunch',
        'test_preparation_course', 'math digree', 'reading digree',
        'writing score'],
        dtype='object')
```

#### Get The DataSet Information

```
dataset.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21004 entries, 0 to 21003
Data columns (total 8 columns):
#
     Column
                                  Non-Null Count Dtype
 0
                                  21002 non-null object
     sex
                                  20962 non-null object
     race ethnicity
                                  20962 non-null
 2
     parental level of education
                                                  object
 3
    lunch
                                  21004 non-null
                                                  object
4
     test_preparation_course
                                  20983 non-null
                                                  object
 5
     math digree
                                  21004 non-null
                                                  int64
     reading digree
                                  20983 non-null float64
                                  21004 non-null int64
 7
     writing score
dtypes: float64(1), int64(2), object(5)
memory usage: 1.3+ MB
```

#### Get The DataSet Describe

```
dataset.describe()
                      reading digree
        math digree
                                      writing score
       21004.000000
                        20983.000000
                                       21004.000000
count
          66.105408
                          69.151027
                                          68.068511
mean
          15.169625
                           14.598066
                                          15.190079
std
min
           0.000000
                           17.000000
                                          10.000000
```

25% 50%	57.000000 66.000000	59.000000 70.000000	58.000000 69.000000
75%	77.000000	79.000000	79.000000
max	100.000000	100.000000	100.000000

#### Select Columns From Table With DataType

```
dataset.select_dtypes('int64').columns
Index(['math digree', 'writing score'], dtype='object')
```

### Get All unique Value From The Column

```
dataset.sex.unique()
array(['female', 'male', 'ذكر', 'انثى', nan], dtype=object)
dataset.sex.value counts()
sex
female
          10875
male
          10119
6
       انثى
       ذکر
Name: count, dtype: int64
dataset.lunch.unique()
array(['standard', 'free/reduced'], dtype=object)
dataset.lunch.value counts()
lunch
standard
                13549
free/reduced
                 7455
Name: count, dtype: int64
```

#### ReName The Column Names

```
dataset.rename(columns={'math digree':'math_digree'}, inplace=True)
dataset.rename(columns={'reading digree':'reading_digree'},
inplace=True)
dataset.rename(columns={'writing score':'writing_score'},
inplace=True)
dataset

sex race_ethnicity parental_level_of_education lunch
```

0	female	group B	bachelor's degree	standard
1	female	group C	some college	standard
2	female	NaN	master's degree	standard
3	male	group A	associate's degree fr	ee/reduced
4	male	group C	NaN	standard
20999	female	group E	master's degree	standard
21000	male	group C	<u>-</u>	ee/reduced
21001	female	group C	-	ee/reduced
21002	female	group D	some college	standard
21003	female	group D	some college fr	ee/reduced
		aration_course	math_digree reading_digree	
writin 0	g_score	none	72 72.0	
74 1			69 90.0	
88		completed		
2 93		none	90 95.0	
3		none	47 57.0	
44 4		none	76 78.0	
75 				
20999 95		completed	58 99.0	
21000 55		none	62 55.0	
21001		completed	49 71.0	
65 21002		completed	68 48.0	
77 21003		none	77 89.0	
86				
[21004	rows x 8	columns]		

## **Data Cleaning**

## Missing Vlaues

#### Get The Number of Null Value

```
dataset.isnull().sum()
                                 2
sex
                                42
race ethnicity
parental_level_of_education
                                42
lunch
                                 0
test preparation course
                                21
math digree
                                 0
reading digree
                                21
writing score
                                 0
dtype: int64
```

#### Drop Rows With Missing Values

```
# define a dictionary with sample data which includes some missing
values
data = {
    'A': [1, 2, 3, None, 5],
    'B': [None, 2, 3, 4, 5],
    'C': [1, 2, None, None, 5]
}
df = pd.DataFrame(data)
print("Original Data:\n",df)
print()
# use dropna() to remove rows with any missing values
df cleaned = df.dropna()
print("Cleaned Data:\n",df_cleaned)
Original Data:
               C
     Α
          В
 1.0 NaN 1.0
1 2.0 2.0 2.0
2 3.0 3.0 NaN
3 NaN 4.0 NaN
4 5.0 5.0 5.0
Cleaned Data:
    A B C
```

2.0 2.0 2.0 4 5.0 5.0 5.0 myFirestCopy=dataset.copy() myFirestCopy sex race ethnicity parental level of education lunch bachelor's degree 0 female group B standard female group C some college standard 2 female master's degree standard NaN associate's degree free/reduced male group A male group C NaN standard 20999 female group E master's degree standard 21000 male group C high school free/reduced 21001 female high school free/reduced group C 21002 female some college standard group D 21003 female group D some college free/reduced test preparation course math digree reading digree writing score 72 72.0 0 none 74 69 90.0 1 completed 88 2 90 95.0 none 93 3 47 57.0 none 44 78.0 none 76 75 . . . 20999 completed 58 99.0 95 21000 55.0 none 62 55 21001 49 completed 71.0

65

21002 77		completed	68	48.	0
21003 86		none	77	89.	0
	_				
[21004	rows x 8	3 columns]			
		/FirestCopy.dropna() Data:\n",df_cleaned	)		
Cleane	d Data:	race ethnicity pare	ntal level of	education	
lunch	\				
0	female	group B		's degree	standard
1	female	group C	som	e college	standard
3	male	group A	associate	's degree	free/reduced
5	female	group B	associate	's degree	standard
6	female	group B	som	e college	standard
20999	female	group E	master	's degree	standard
21000	male	group C	hi	gh school	free/reduced
21001	female	group C	hi	gh school	free/reduced
21002	female	group D	som	e college	standard
21003	female	group D	som	e college	free/reduced
	test nrer	paration_course mat	h digree rea	ding digre	e writing
score	cest_prep	_	_		_
0 74		none	72	72.	Θ
1		completed	69	90.	0
88 3		none	47	57.	0
44 5		none	71	83.	Θ
78					
6 92		completed	88	95.	U
20999		completed	58	99.	0

```
95
21000
                                           62
                                                          55.0
                           none
55
21001
                                           49
                                                          71.0
                     completed
21002
                     completed
                                           68
                                                          48.0
77
21003
                                           77
                                                          89.0
                           none
86
[20897 rows x 8 columns]
df cleaned.isnull().sum()
                                 0
sex
race_ethnicity
                                 0
parental level of education
                                 0
lunch
                                 0
test preparation course
                                 0
math digree
                                 0
reading digree
                                 0
writing score
                                 0
dtype: int64
```

#### Fill Missing Values With Const Value

```
# define a dictionary with sample data which includes some missing
values
data = {
    'A': [1, 2, 3, None, 5],
    'B': [None, 2, 3, 4, 5],
    'C': [1, 2, None, None, 5]
df = pd.DataFrame(data)
print("Original Data:\n", df)
# filling NaN values with 0
df.fillna(0, inplace=True)
print("\nData after filling NaN with 0:\n", df)
Original Data:
          B C
     Α
  1.0 NaN 1.0
1 2.0 2.0 2.0
  3.0 3.0 NaN
3 NaN 4.0 NaN
4 5.0 5.0 5.0
Data after filling NaN with 0:
          В
0 1.0 0.0 1.0
```

```
2.0
        2.0 2.0
1
2
  3.0
       3.0 0.0
3
  0.0 4.0
             0.0
4 5.0
        5.0 5.0
dataset.isnull().sum()
                                 2
sex
race ethnicity
                                42
                                42
parental level of education
lunch
                                 0
                                21
test preparation course
math digree
                                 0
                                21
reading digree
writing score
                                 0
dtype: int64
mis=dataset[dataset['reading digree'].isna()]
mis
        sex race ethnicity parental level of education
                                                              lunch
16
       male
                    group C
                                             high school
                                                           standard
1016
                                             high school
       male
                    group C
                                                           standard
                                             high school
2016
       male
                    group C
                                                           standard
3016
                                             high school
       male
                    group C
                                                           standard
4016
       male
                                             high school
                                                           standard
                    group C
5016
       male
                    group C
                                             high school
                                                           standard
6016
       male
                    group C
                                             high school
                                                           standard
7016
       male
                    group C
                                             high school
                                                           standard
8016
                                             high school
                                                           standard
       male
                    group C
9016
       male
                    group C
                                             high school
                                                           standard
10016
       male
                    group C
                                             high school
                                                           standard
                    group C
                                             high school
                                                           standard
11016
       male
12016
       male
                    group C
                                             high school
                                                           standard
13016
                                             high school
                                                           standard
       male
                    group C
14016
       male
                    group C
                                             high school
                                                           standard
15016
                                             high school
       male
                    group C
                                                           standard
                                             high school
16016
       male
                    group C
                                                           standard
17016
       male
                    group C
                                             high school
                                                           standard
18016
       male
                                             high school
                    group C
                                                           standard
19018
       male
                    group C
                                             high school
                                                           standard
20018
       male
                    group C
                                             high school
                                                           standard
                                              reading digree writing
      test preparation course math digree
score
16
                                          88
                                                          NaN
                          none
86
1016
                                          88
                          none
                                                          NaN
86
                                          88
                                                          NaN
2016
                          none
```

```
86
3016
                                                           NaN
                                           88
                           none
86
4016
                                           88
                                                           NaN
                           none
86
                                           88
5016
                           none
                                                           NaN
86
6016
                                           88
                                                           NaN
                           none
86
7016
                                           88
                                                           NaN
                           none
86
8016
                                           88
                                                           NaN
                           none
86
                                           88
9016
                                                           NaN
                           none
86
                                                           NaN
10016
                                           88
                           none
86
11016
                                           88
                                                           NaN
                           none
86
12016
                                           88
                                                           NaN
                           none
86
13016
                                           88
                                                           NaN
                           none
86
                                           88
14016
                                                           NaN
                           none
86
                                           88
15016
                                                           NaN
                           none
86
                                           88
                                                           NaN
16016
                           none
86
17016
                                           88
                                                           NaN
                           none
86
                                           88
18016
                           none
                                                           NaN
86
19018
                                           88
                                                           NaN
                           none
86
20018
                                           88
                                                           NaN
                           none
86
mySecondCopy=dataset.copy()
mySecondCopy.loc[:, "reading digree"] = mySecondCopy["reading
digree"].fillna(20.0)
mySecondCopy.isnull().sum()
                                  2
sex
                                 42
race ethnicity
parental level_of_education
                                 42
                                  0
lunch
test_preparation_course
                                 21
math digree
                                  0
                                  0
reading digree
```

0 writing score dtype: int64 mis=mySecondCopy[mySecondCopy['reading digree']==20.0] mis sex race ethnicity parental level of education lunch 16 group C high school standard male 1016 male group C high school standard 2016 high school male group C standard 3016 male group C high school standard high school standard 4016 male group C 5016 high school standard male group C 6016 male high school standard group C 7016 male high school standard group C 8016 male group C high school standard high school 9016 male standard group C 10016 male group C high school standard 11016 male high school standard group C 12016 male high school standard group C 13016 male high school standard group C 14016 male high school standard group C 15016 male group C high school standard 16016 male high school standard group C 17016 male high school standard group C 18016 male high school standard group C 19018 group C high school standard male 20018 male group C high school standard reading digree writing test preparation course math digree score 16 88 20.0 none 86 88 20.0 1016 none 86 20.0 2016 88 none 86 20.0 3016 88 none 86 88 20.0 4016 none 86 88 20.0 5016 none 86 6016 88 20.0 none 86

88

88

88

none

none

none

20.0

20.0

20.0

7016

8016

86

86 9016

86			
10016	none	88	20.0
86			
11016	none	88	20.0
86 12016	nana	00	20.0
86	none	88	20.0
13016	none	88	20.0
86			
14016	none	88	20.0
86			
15016	none	88	20.0
86		0.0	20.0
16016 86	none	88	20.0
17016	none	88	20.0
86	Hone	00	2010
18016	none	88	20.0
86			
19018	none	88	20.0
86			
20018	none	88	20.0
86			

mySecondCopy.iloc[16,1]='group A'
mySecondCopy.head(20)

0 1 2 3 4	sex female female female male	race_ethnicity group B group C NaN group A group C	parental_level_of_education bachelor's degree some college master's degree associate's degree NaN	lunch standard standard standard free/reduced standard	\
5 6 7 8 9 10 11 12 13 14 15 16	female female male female male female female female female female	group B group B group D group B group C group D group B group A group A	associate's degree some college some college high school high school associate's degree associate's degree high school some college master's degree some high school high school	standard standard free/reduced free/reduced	
17 18 19	female male female	group B group C group C	some high school master's degree associate's degree	free/reduced free/reduced free/reduced	

test\_preparation\_course math digree reading digree writing score

0	none	72	72.0	74
1	completed	69	90.0	88
2	none	90	95.0	93
3	none	47	57.0	44
4	none	76	78.0	75
5	none	71	83.0	78
6	completed	88	95.0	92
7	none	40	43.0	39
8	completed	64	64.0	67
9	none	38	60.0	50
10	none	58	54.0	52
11	none	40	52.0	43
12	none	65	81.0	73
13	completed	78	72.0	70
14	NaN	50	53.0	58
15	none	69	75.0	78
16	none	88	20.0	86
17	none	18	32.0	28
18	completed	46	42.0	46
19	none	54	58.0	61

#### Replace Missing Values With Mean, Median and Mode

```
data = {
    'A': [1, 2, np.nan, 4, 5],
    'B': [np.nan, 2, 3, 4, 5],
    'C': [1, 2, 3, np.nan, 5],
    'D': [1, 2, 3, 4, 5]
}
df = pd.DataFrame(data)
df_copy=df.copy()
```

```
print(f'The Data Before \n{df}')
print('-----
# replace missing values with mean
# df copy['A']=df['A'].fillna(value=df['A'].mean() )
# # replace missing values with median
# df_copy['B']=df['B'].fillna(value=df['B'].median() )
# # replace missing values with mode
# df copy['C']=df['C'].fillna(value=df['C'].mode()[0] )
# print(df copy)
The Data Before
     Α
          В
               C
                  D
   1.0
             1.0
                  1
        NaN
       2.0
                  2
  2.0
             2.0
2
        3.0
             3.0
                  3
  NaN
3
  4.0
       4.0
             NaN
                  4
                  5
4
   5.0
        5.0
             5.0
dataset.describe()
        math digree
                      reading digree
                                      writing score
       21004.000000
                        20983.000000
                                       21004.000000
count
mean
          66.105408
                           69.151027
                                           68.068511
std
          15.169625
                           14.598066
                                           15.190079
                                           10.000000
min
           0.000000
                           17.000000
25%
          57.000000
                           59.000000
                                           58.000000
50%
                           70.000000
                                           69.000000
          66.000000
75%
          77,000000
                           79,000000
                                           79.000000
                          100.000000
         100.000000
                                         100.000000
max
mis=dataset[dataset["reading digree"].isna()]
mis
        sex race ethnicity parental level of education
                                                             lunch
                                             high school
16
       male
                    group C
                                                          standard
1016
       male
                    group C
                                             high school
                                                          standard
2016
       male
                    group C
                                             high school
                                                          standard
3016
       male
                                             high school
                                                          standard
                    group C
4016
       male
                    group C
                                             high school
                                                          standard
5016
                                             high school
                                                          standard
       male
                    group C
6016
       male
                    group C
                                             high school
                                                          standard
7016
       male
                   group C
                                             high school
                                                          standard
8016
       male
                    group C
                                             high school
                                                          standard
9016
       male
                    group C
                                             high school
                                                          standard
                                             high school
10016
       male
                    group C
                                                          standard
11016
       male
                    group C
                                             high school
                                                          standard
12016
       male
                                             high school
                                                          standard
                   group C
```

13016 14016 15016 16016 17016 18016 19018 20018	male male male male male male male male	group C		high school	standard standard standard standard standard standard
00000	test_prepara	ation_course	math digree	reading di	gree writing
score 16		none	88		NaN
86					
1016		none	88		NaN
86 2016		none	88		NaN
86		Horic	00		Nan
3016		none	88		NaN
86			0.0		N-N
4016 86		none	88		NaN
5016		none	88		NaN
86					
6016		none	88		NaN
86 7016		none	88		NaN
86					113.11
8016		none	88		NaN
86 9016		nono	88		NaN
86		none	00		NaN
10016		none	88		NaN
86					
11016 86		none	88		NaN
12016		none	88		NaN
86					
13016		none	88		NaN
86 14016		none	88		NaN
86		Hone	00		IVAIV
15016		none	88		NaN
86			0.0		N - N
16016 86		none	88		NaN
17016		none	88		NaN
86					
18016		none	88		NaN
86					

19018 86	no	one	88	NaN	
20018	no	one	88	NaN	
86					
mis=datas	<pre>et.copy() ing digree"]=datas</pre>	set["reading			
digree"].	fillna(value=datas		ligree"]. <mark>mi</mark> n()	)	
mis.head(	20)				
	x race_ethnicity p				\
0 femal 1 femal	<b>5</b> 1	bach	nelor's degree some college		
2 femal	<b>.</b> .	ma	ster's degree		
3 mal	<b>.</b> .		ciate's degree	free/reduced	
4 mal 5 femal		25506	NaN Nataka dagaa		
<pre>5 femal 6 femal</pre>		d5500	iate's degree some college		
7 mal	9 .		some college		
8 mal	<b>.</b> .		high school		
9 femal 10 mal	<b>.</b> .	25506	high school iate's degree		
10 mat	<b>.</b> .		ciate's degree		
12 femal	•	43500	high school		
13 mal	9 1		some college		
14 femal 15 femal	<b>.</b> .		nster's degree ne high school		
16 mal	<b>.</b> .	5011	high school		
17 femal	e group B		ne high school	free/reduced	
18 mal	<b>.</b> .		ster's degree		
19 femal	e group C	a5500	ciate's degree	free/reduced	
test_p	reparation_course	math digree	reading digr	ee writing scor	e
0	none	72	72	.0 7	4
1	completed	69	90	.0 8	8
2	none	90	95	.0 9	3
3	none	47	57	.0 4	4
4	none	76	78	.0 7	5
5	none	71	83	.0 78	8
6	completed	88	95	.0 9	2
7	none	40	43	.0 3	9
8	completed	64	64	.0 6	7

9	none	38	60.0	50
10	none	58	54.0	52
11	none	40	52.0	43
12	none	65	81.0	73
13	completed	78	72.0	70
14	NaN	50	53.0	58
15	none	69	75.0	78
16	none	88	17.0	86
17	none	18	32.0	28
18	completed	46	42.0	46
19	none	54	58.0	61

# Handle Duplicates Values

datase	t			
	sex	race_ethnicity	<pre>parental_level_of_education</pre>	lunch
0	female	group B	bachelor's degree	standard
1	female	group C	some college	standard
2	female	NaN	master's degree	standard
3	male	group A	associate's degree	free/reduced
4	male	group C	NaN	standard
20999	female	group E	master's degree	standard
21000	male	group C	high school	free/reduced
21001	female	group C	high school	free/reduced
21002	female	group D	some college	standard
21003	female	group D	some college	free/reduced

	test preparation course	math digree	reading digree	writing
	test_preparation_course	math digree	reading digree	wilting
score		70	72.0	
0	none	72	72.0	
74				
1	completed	69	90.0	
88				
2	none	90	95.0	
93				
3	none	47	57.0	
44				
4	none	76	78.0	
75				
20999	completed	58	99.0	
95	•			
21000	none	62	55.0	
55				
21001	completed	49	71.0	
65	00p 1010u	.5	, 2.0	
21002	completed	68	48.0	
77	comp te teu	00	1010	
21003	none	77	89.0	
86	lione	7.7	09.0	
80				

[21004 rows x 8 columns]

dup=dataset[dataset.duplicated()]
dup

	sex	race_ethnicity	parental_level_of_education	lunch
1000	female	group B	bachelor's degree	standard
1000	remate	group B	buchetor 5 degree	Standard
1001	female	group C	some college	standard
1002	female	NaN	master's degree	standard
1003	male	group A	associate's degree	free/reduced
1004	male	group C	NaN	standard
20984	male	group B	some high school	standard
20986	انثى	group A	some college	standard
20987	انثی	group A	some college	standard
20993	female	group D	some college	free/reduced

21000	male	group C		high school	free/reduced
21000	ma cc	group c		nigh senoce	Tree, reduced
	test_preparati	on_course	math digree	reading digree	e writing
score					
1000		none	72	72.0	)
74					
1001		completed	69	90.0	)
88		-			
1002		none	90	95.0	)
93					
1003		none	47	57.0	)
44					
1004		none	76	78.0	)
75					
20984		completed	79	85.0	)
86					
20986		completed	77	87.0	)
91		•			
20987		completed	77	87.0	)
91					
20993		completed	67	86.0	)
83					
21000		none	62	55.0	)
55					
[19858	3 rows x 8 colu	mns]			

dropDu=dataset.drop\_duplicates()
dropDu

	sex	race_ethnicity	<pre>parental_level_of_education</pre>	lunch
\				
0	female	group B	bachelor's degree	standard
1	female	anoun C	some college	c+ondond
1	relliace	group C	some college	standard
2	female	NaN	master's degree	standard
3	male	group A	associate's degree	free/reduced
	-		N. N.	
4	male	group C	NaN	standard
	_			
20998	male	group A	high school	standard
20000	female	arous E	master's deares	standard
20999	remate	group E	master's degree	standard

21001 female group C high school free/reduced 21002 female group D some college standard 21003 female group D some college free/reduced  test_preparation_course math digree reading digree writing score none 72 72.0  74 completed 69 90.0  88 2 none 90 95.0  93 none 47 57.0  44 none 76 78.0  75 20998 none 63 63.0  65 20999 completed 58 99.0  95 21001 completed 49 71.0  65 21002 completed 68 48.0  77 21003 none 77 89.0						
21003 female group D some college free/reduced  test_preparation_course math digree reading digree writing  score  none 72 72.0  74  completed 69 90.0  88  none 90 95.0  3 none 47 57.0  44  none 76 78.0   20998 none 63 63.0  65  20999 completed 58 99.0  95  1001 completed 49 71.0  65  21002 completed 68 48.0  77  21003 none 77 89.0	21001	female	group C		high school	free/reduced
test_preparation_course math digree reading digree writing  score 0	21002	female	group D		some college	standard
test_preparation_course math digree reading digree writing  score 0	21003	female	aroup D		some college	free/reduced
score       0       none       72       72.0         74       1       completed       69       90.0         88       2       none       90       95.0         93       3       none       47       57.0         44       none       76       78.0         75            20998       none       63       63.0         65       65       99.0         21001       completed       49       71.0         65       21002       completed       68       48.0         77       21003       none       77       89.0		· oma co	g. 0 ap 2		Jome Correge	,
0       none       72       72.0         74       1       completed       69       90.0         88       2       none       90       95.0         93       3       none       47       57.0         44       none       76       78.0         75            20998       none       63       63.0         65            20999       completed       58       99.0         95            21001       completed       49       71.0         65            21002       completed       68       48.0         77            21003       none       77       89.0         86		test_prepa	ration_course	math digree	reading digre	e writing
74 1						_
1 completed 69 90.0 888 2 none 90 95.0 93 3 none 47 57.0 44 4 none 76 78.0 75 20998 none 63 63.0 65 20999 completed 58 99.0 95 21001 completed 49 71.0 65 21002 completed 68 48.0 77 21003 none 77 89.0			none	72	72.	0
888 2			1	60	0.0	0
2 none 90 95.0 93 3 none 47 57.0 44 4 none 76 78.0 75 20998 none 63 63.0 65 20999 completed 58 99.0 95 21001 completed 49 71.0 65 21002 completed 68 48.0 77 21003 none 77 89.0			completed	69	90.	O
93 3			nono	00	ΛE	0
3       none       47       57.0         44       none       76       78.0         75            20998       none       63       63.0         65            20999       completed       58       99.0         95            21001       completed       49       71.0         65            21002       completed       68       48.0         77            21003       none       77       89.0         86			none	90	95.	U
44 4			none	17	57	O
4       none       76       78.0         75            20998       none       63       63.0         65            20999       completed       58       99.0         95            21001       completed       49       71.0         65            21002       completed       68       48.0         77            21003       none       77       89.0         86			Hone	7/	57.	O
75 20998 none 63 63.0 65 20999 completed 58 99.0 95 21001 completed 49 71.0 65 21002 completed 68 48.0 77 21003 none 77 89.0			none	76	78.	Θ
20998 none 63 63.0 65 20999 completed 58 99.0 95 21001 completed 49 71.0 65 21002 completed 68 48.0 77 21003 none 77 89.0						_
20998 none 63 63.0 65 20999 completed 58 99.0 95 21001 completed 49 71.0 65 21002 completed 68 48.0 77 21003 none 77 89.0						
65 20999 completed 58 99.0 95 21001 completed 49 71.0 65 21002 completed 68 48.0 77 21003 none 77 89.0			nono	63	63	Ω
20999 completed 58 99.0 95 21001 completed 49 71.0 65 21002 completed 68 48.0 77 21003 none 77 89.0			none	03	05.	U
95 21001 completed 49 71.0 65 21002 completed 68 48.0 77 21003 none 77 89.0 86			completed	58	99.	Θ
21001 completed 49 71.0 65 21002 completed 68 48.0 77 21003 none 77 89.0 86			00p 1010u	30	33.	
21002 completed 68 48.0 77 21003 none 77 89.0 86	21001		completed	49	71.	0
77 21003 none 77 89.0 86	65					
21003 none 77 89.0 86			completed	68	48.	0
86				77	00	0
			none	11	89.	Θ
[11/6 rous v 0 columns]	80					
[1146 rows x 8 columns]	[1146	rows x 8 c	olumns]			

## Pandas Handling Wrong Format

```
# create dataframe
data = {
    'Country': ['USA', 'Canada', 'Australia', 'Germany', 'Japan'],
    'Date': ['2023-07-20', '2023-07-21', '2023-07-22', '2023-07-23',
'2023-07-24'],
    'Temperature': [25.5, '28.0', 30.2, 22.8, 26.3]
}
df = pd.DataFrame(data)

print(df.Temperature.unique())
print('-----')
# convert temperature column to float
df['Temperature'] = df['Temperature'].astype(float)
```

```
print(df.Temperature.unique())
print('----')
print(df.info())
print('-----')
# calculate the mean temperature
mean temperature = df['Temperature'].mean()
print(mean temperature)
[25.5 '28.0' 30.2 22.8 26.3]
-----
[25.5 28. 30.2 22.8 26.3]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 3 columns):
    Column Non-Null Count Dtype
#
0 Country 5 non-null object
1 Date 5 non-null object
2 Temperature 5 non-null float64
dtypes: float64(1), object(2)
memory usage: 252.0+ bytes
None
26.560000000000000
# create a sample dataframe with mixed date formats
df = pd.DataFrame({'date': ['2022-12-01', '01/02/2022', '2022-03-23',
'03/02/2022', '3 4 2023', '2023.9.30']})
# convert the date column to datetime format
df['date'] = pd.to datetime(df['date'], format='mixed', dayfirst=True)
print(df)
       date
0 2022-12-01
1 2022-01-02
2 2022-03-23
3 2022-03-02
4 2023-03-04
5 2023-09-30
```

#### Pandas Handling Wrong Data

```
data = {
    'Name': ['John', 'Michael', 'Tom', 'Alex', 'Ryan'],
    'Age': [8, 9, 7, 80, 100],
    'Gender': ['M', 'M', 'F', 'M'],
```

```
'Standard': [3, 4, 12, 3, 5]
}
df = pd.DataFrame(data)
# replace F with M
df.loc[3, 'Gender'] = 'M'
print(df)
            Age Gender
      Name
                         Standard
0
      John
              8
                      М
                                 3
1
   Michael
              9
                      Μ
                                 4
2
              7
                                12
       Tom
                      Μ
3
      Alex
                                 3
             80
                      Μ
                                 5
4
      Ryan
           100
                      М
data = {
    'Name': ['John', 'Michael', 'Tom', 'Alex', 'Ryan'],
    'Age': [8, 9, 7, 80, 100],
    'Gender': ['M', 'M', 'M', 'M', 'M'],
    'Standard': [3, 4, 12, 3, 5]
}
df = pd.DataFrame(data)
# replace values based on conditions
for i in df.index:
    age val = df.loc[i, 'Age']
    if (age_val > 14) and (age_val 10 == 0):
        df.loc[i, 'Age'] = age_val/10
print(df)
      Name
            Age Gender
                         Standard
0
      John
                                 3
              8
                      М
1
   Michael
               9
                                 4
                      М
2
              7
                                12
       Tom
                      М
3
      Alex
              8
                      Μ
                                 3
      Ryan
             10
                      М
                                 5
len(dataset)
21004
dataset.sex.unique()
array(['female', 'male', 'ذكر', 'انثى', nan], dtype=object)
for i in range(0,len(dataset)):
    if dataset.loc[i, 'sex']==' : 'انثی'
         dataset.loc[i,'sex']='female'
```

```
[ ' انثی ' == [ ' sex ' ] dataSex=dataset
dataSex
      sex race_ethnicity parental_level_of_education
                                                           lunch \
ذکر 2081
            group B
                                    high school free/reduced
            group C
                                    high school free/reduced
ذکر 2091
    test preparation course math digree reading digree writing
score \
2081
                                     49
                                                   45.0
                       none
45
2091
                                                   34.0
                       none
                                     27
36
             18729
                    18731
                              20985
                                     20986
                                             20987
      18728
2081 female female
                     female female female
2091 female female female female female
dataSex=dataset[dataset['sex']==' نذكر'==[
dataSex
Empty DataFrame
Columns: [sex, race ethnicity, parental level of education, lunch,
test preparation course, math digree, reading digree, writing score,
18728, 18729, 18731, 20985, 20986, 20987]
Index: []
for i in range(0,len(dataset)):
   if dataset.loc[i, 'sex']=='ذكر':
        dataset.loc[i,'sex']='Male'
```

#### Write To New DataSets

dataset.to\_csv('datanew2.csv')