

Q.1. Display first 5 rows of dataset

=> ds.head()

ds.head()								
	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

Q.2. Display the last 5 rows of dataset

=> ds.tail()

ds.tail()								
	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

Q.3. What is the shape of the DataSet (number of rows and columns)?

=> ds.shape

```
ds.shape

(1000, 8)
```

Q.4. Display summary information about the DataSet, including data types and non-null counts.

=> ds.info()

```
ds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   gender                                1000 non-null   object
 1   race/ethnicity                        1000 non-null   object
 2   parental level of education           1000 non-null   object
 3   lunch                                 1000 non-null   object
 4   test preparation course               1000 non-null   object
 5   math score                           1000 non-null   int64
 6   reading score                        1000 non-null   int64
 7   writing score                         1000 non-null   int64
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

**Q.5. Get descriptive statistics for numerical columns in the DataSet.**

=> ds.describe()

```
ds.describe()
```

	math score	reading score	writing score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

**Q.6. Select the "gender" column from the DataSet.**

=> ds['gender']

```
ds['gender']

0      female
1      female
2      female
3        male
4        male
...
995    female
996      male
997    female
998    female
999    female
Name: gender, Length: 1000, dtype: object
```

Q.7. Select the "gender" and "race/ethnicity" columns from the DataSet.

=> ds[['gender','race/ethnicity']]

ds[['gender','race/ethnicity']]

	gender	race/ethnicity
0	female	group B
1	female	group C
2	female	group B
3	male	group A
4	male	group C
...	...	...
995	female	group E
996	male	group C
997	female	group C
998	female	group D
999	female	group D

1000 rows × 2 columns

Q.8. Select the first row from the DataSet by index.

=> ds.iloc[0]

```
ds.iloc[0]
```

```
gender          female
race/ethnicity   group B
parental level of education  bachelor's degree
lunch            standard
test preparation course  none
math score       72
reading score    72
writing score    74
Name: 0, dtype: object
```

**Q.9. Select the row with label/index 0 from the DataSet.**

=> ds.loc[0]

```
ds.loc[0]
```

```
gender          female
race/ethnicity   group B
parental level of education  bachelor's degree
lunch            standard
test preparation course  none
math score       72
reading score    72
writing score    74
Name: 0, dtype: object
```

**Q.10. Select rows from 1 to 5 and columns from 2 to 3**

=> ds.iloc[1:5, 2:4]

```
ds.iloc[1:5, 2:4]
```

	parental level of education	lunch
1	some college	standard
2	master's degree	standard
3	associate's degree	free/reduced
4	some college	standard

**Q.11. Filter the DataSet for rows where "math score" is greater than 90.**

=> ds[ds['math score']>90]

ds[ds['math score']>90]								
	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
34	male	group E	some college	standard	none	97	87	82
104	male	group C	some college	standard	completed	98	86	90
114	female	group E	bachelor's degree	standard	completed	99	100	100
121	male	group B	associate's degree	standard	completed	91	89	92
149	male	group E	associate's degree	free/reduced	completed	100	100	93
165	female	group C	bachelor's degree	standard	completed	96	100	100
171	male	group E	some high school	standard	none	94	88	78
179	female	group D	some high school	standard	completed	97	100	100
233	male	group E	some high school	standard	none	92	87	78

Q.12. Use the query method to filter rows where "math score" is greater than 90.

=> ds.query("`math score` > 90")

ds.query("`math score` > 90")								
	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
34	male	group E	some college	standard	none	97	87	82
104	male	group C	some college	standard	completed	98	86	90
114	female	group E	bachelor's degree	standard	completed	99	100	100
121	male	group B	associate's degree	standard	completed	91	89	92
149	male	group E	associate's degree	free/reduced	completed	100	100	93
165	female	group C	bachelor's degree	standard	completed	96	100	100
171	male	group E	some high school	standard	none	94	88	78
179	female	group D	some high school	standard	completed	97	100	100
233	male	group E	some high school	standard	none	92	87	78

Q.13. Sort the DataSet by the "math score" column in ascending order.

=> ds.sort\_values('math score')



```
ds.sort_values('math score')
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
59	female	group C	some high school	free/reduced	none	0	17	10
980	female	group B	high school	free/reduced	none	8	24	23
17	female	group B	some high school	free/reduced	none	18	32	28
787	female	group B	some college	standard	none	19	38	32
145	female	group C	some college	free/reduced	none	22	39	33
...	...	...	...	...	...	...	...	...
625	male	group D	some college	standard	completed	100	97	99
623	male	group A	some college	standard	completed	100	96	86
451	female	group E	some college	standard	none	100	92	97
962	female	group E	associate's degree	standard	none	100	100	100
149	male	group E	associate's degree	free/reduced	completed	100	100	93

1000 rows × 8 columns

Q.14. Check if there are any missing values in the DataSet.

=> ds.isnull()

```
ds.isnull()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
995	False	False	False	False	False	False	False	False
996	False	False	False	False	False	False	False	False
997	False	False	False	False	False	False	False	False
998	False	False	False	False	False	False	False	False
999	False	False	False	False	False	False	False	False

1000 rows × 8 columns

Q.15. Count the number of missing values in each column.

=> ds.isnull().sum()

```
ds.isnull().sum()
```

```
gender          0
race/ethnicity   0
parental level of education  0
lunch            0
test preparation course  0
math score       0
reading score    0
writing score    0
dtype: int64
```

**Q.16. Fill missing values with 0.**

=> ds.fillna(0, inplace= True)

```
ds.isnull().sum()
```

```
ds.fillna(0, inplace= True)
```

```
ds.isnull().sum()
```

```
gender          0
race/ethnicity   0
parental level of education  0
lunch            0
test preparation course  0
math score       0
reading score    0
writing score    0
dtype: int64
```

**Q.17. Sort the dataSet based on a specific column.**

=> ds.sort\_values(by='math score', ascending=False)

```
ds.sort_values(by='math score', ascending=False)
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
962	female	group E	associate's degree	standard	none	100	100	100
625	male	group D	some college	standard	completed	100	97	99
458	female	group E	bachelor's degree	standard	none	100	100	100
623	male	group A	some college	standard	completed	100	96	86
451	female	group E	some college	standard	none	100	92	97
...	...	...	...	...	...	...	...	...
145	female	group C	some college	free/reduced	none	22	39	33
787	female	group B	some college	standard	none	19	38	32
17	female	group B	some high school	free/reduced	none	18	32	28

**Q.18. Filter rows where the lunch is of standard type.**

=> ds[ds['lunch']=='standard']

ds[ds['lunch']=='standard']

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
4	male	group C	some college	standard	none	76	78	75
5	female	group B	associate's degree	standard	none	71	83	78
...	...	...	...	...	...	...	...	...
987	male	group E	some high school	standard	completed	81	75	76
991	female	group B	some high school	standard	completed	65	82	78
994	male	group A	high school	standard	none	63	63	62
995	female	group E	master's degree	standard	completed	88	99	95
998	female	group D	some college	standard	completed	68	78	77

645 rows × 8 columns

Q.19. Convert 'male'/'female' responses to 'm'/'f'

=> ds['gender'].map({'male':'m','female':'f'})

ds['gender'].map({'male':'m','female':'f'})

0f

1f

2f

3m

4m

...

995f

996m

997f

998f

999f

Name: gender, Length: 1000, dtype: object

Q.20. Selects the element in the first row and first column

=> ds.iloc[0, 0]

ds.iloc[0, 0]

'female'

Q.21. Find all rows where gender is "male" and race/ethnicity is “group B”

=> ds[(ds['gender']=='male')&(ds['race/ethnicity']=='group B')]



```
ds[(ds['gender']=='male')&(ds['race/ethnicity']=='group B')]
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
7	male	group B	some college	free/reduced	none	40	43	39
26	male	group B	some college	standard	none	69	54	55
39	male	group B	associate's degree	free/reduced	none	57	56	57
43	male	group B	some college	free/reduced	completed	59	65	66
45	male	group B	associate's degree	standard	none	65	54	57
...	...	...	...	...	...	...	...	...
919	male	group B	some college	standard	completed	91	96	91
946	male	group B	high school	standard	none	82	82	80
948	male	group B	some high school	free/reduced	completed	49	50	52
976	male	group B	some college	free/reduced	completed	60	62	60
982	male	group B	some high school	standard	completed	79	85	86

86 rows × 8 columns

**Q.22. Find unique values in the 'parental level of education' column.**

=> ds['parental level of education'].unique()

```
ds['parental level of education'].unique()
```

```
array(["bachelor's degree", 'some college', "master's degree",  
      "associate's degree", 'high school', 'some high school'],  
      dtype=object)
```

**Q.23. Count unique values in the 'parental level of education' column.**

=> ds['parental level of education'].nunique()

```
ds['parental level of education'].nunique()
```

6

**Q.24. Calculate the average reading score.**

=> ds['reading score'].mean()

```
ds['reading score'].mean()
```

69.169

**Q.25. Select the first 3 rows of the 'test preparation course' and 'writing score' columns.**

=> ds.loc[:2,['test preparation course','writing score']]

```
ds.loc[:2,['test preparation course','writing score']]
```

	test preparation course	writing score
0	none	74
1	completed	88
2	none	93

Q.26. Select the first 3 rows of the 'reading score'.

=> ds.loc[:2,['reading score']]

```
ds.loc[:2,['reading score']]
```

	reading score
0	72
1	90
2	95

Q.27. Change 'test preparation course' for the student with 'writing score' >=90 to "completed".

=> ds.loc[ds['writing score'] >=90, 'test preparation course'] = "completed"  
print(ds.loc[ds['writing score'] >=90])

```
ds.loc[ds['writing score'] >=90, 'test preparation course'] = "completed"  
print(ds.loc[ds['writing score'] >=90])
```

	gender	race/ethnicity	parental level of education	lunch	\
2	female	group B	master's degree	standard	
6	female	group B	some college	standard	
94	female	group B	some college	standard	
104	male	group C	some college	standard	
106	female	group D	master's degree	standard	
..	...	...	...	...	
962	female	group E	associate's degree	standard	
970	female	group D	bachelor's degree	standard	
979	female	group C	associate's degree	standard	
983	female	group A	some college	standard	
995	female	group E	master's degree	standard	
	test preparation course	math score	reading score	writing score	
2	completed	90	95	93	
6	completed	88	95	92	
94	completed	79	86	92	
104	completed	98	86	90	
106	completed	87	100	100	
..	...	...	...	...	
962	completed	100	100	100	
970	completed	89	100	100	
979	completed	91	95	94	
983	completed	78	87	91	
995	completed	88	99	95	

[78 rows x 8 columns]

Q.28. Check if 'race/ethnicity' == 'group F' exists

```
=> if (ds['race/ethnicity'] == 'group F').any():  
    ds.loc[ds['race/ethnicity'] == 'group F', 'lunch'] = "standard"  
    print(ds.loc[ds['race/ethnicity'] == 'group F'])  
else:  
    print("No row with race/ethnicity == group F found.")
```

```
if (ds['race/ethnicity'] == 'group F').any():  
    ds.loc[ds['race/ethnicity'] == 'group F', 'lunch'] = "standard"  
    print(ds.loc[ds['race/ethnicity'] == 'group F']) # Print updated row  
else:  
    print("No row with race/ethnicity == group F found.")
```

No row with race/ethnicity == group F found.

Q.29.

```
=> ds.iloc[[0,1]]
```

ds.iloc[[0,1]]								
	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88

Q.30.

```
=> ds.iloc[[0,5],2]
```

```
ds.iloc[[0,5],2]  
  
0    bachelor's degree  
5    associate's degree  
Name: parental level of education, dtype: object
```

Q.31.

```
=> ds.loc[[0,5],['parental level of education','lunch']]
```

ds.loc[[0,5],['parental level of education','lunch']]		
	parental level of education	lunch
0	bachelor's degree	standard
5	associate's degree	standard

Q.32.

=> ds['lunch'].value\_counts()

```
ds['lunch'].value_counts()
```

lunch  
standard 645  
free/reduced 355  
Name: count, dtype: int64

Q.33.

=> ds.loc[[1,4,7],['gender','test preparation course']]

```
ds.loc[[1,4,7],['gender','test preparation course']]
```

	gender	test preparation course
1	female	completed
4	male	none
7	male	none

Q.34.

=> ds.loc[1:8,['gender','test preparation course']]

```
ds.loc[1:8,['gender','test preparation course']]
```

	gender	test preparation course
1	female	completed
2	female	completed
3	male	none
4	male	none
5	female	none
6	female	completed
7	male	none
8	male	completed

Q.35.

=> ds.loc[1:8:2,['gender','test preparation course']]



```
ds.loc[1:8:2,['gender','test preparation course']]
```

	gender	test preparation course
1	female	completed
3	male	none
5	female	none
7	male	none

Q.36.

=> ds.loc[1:8:2,'gender':'test preparation course']

```
ds.loc[1:8:2,'gender':'test preparation course']
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course
1	female	group C	some college	standard	completed
3	male	group A	associate's degree	free/reduced	none
5	female	group B	associate's degree	standard	none
7	male	group B	some college	free/reduced	none

Q.37.

=> ds.set\_index('parental level of education')

```
ds.set_index('parental level of education')
```

	gender	race/ethnicity	lunch	test preparation course	math score	reading score	writing score
parental level of education							
bachelor's degree	female	group B	standard	none	72	72	74
some college	female	group C	standard	completed	69	90	88
master's degree	female	group B	standard	completed	90	95	93
associate's degree	male	group A	free/reduced	none	47	57	44
some college	male	group C	standard	none	76	78	75
...	...	...	...	...	...	...	...
master's degree	female	group E	standard	completed	88	99	95
high school	male	group C	free/reduced	none	62	55	55
high school	female	group C	free/reduced	completed	59	71	65
some college	female	group D	standard	completed	68	78	77
some college	female	group D	free/reduced	none	77	86	86

1000 rows × 7 columns

Q.38.



```
=> ds.set_index('gender',inplace=True)
ds.index
```

```
ds.set_index('gender',inplace=True)
```

```
ds.index
```

```
Index(['female', 'female', 'female', 'male', 'male', 'female', 'female',
      'male', 'male', 'female',
      ...
      'male', 'female', 'female', 'female', 'male', 'female', 'male',
      'female', 'female', 'female'],
      dtype='object', name='gender', length=1000)
```

### Q.39.

```
=> ds.reset_index(inplace=True)
ds.index
```

```
ds.reset_index(inplace=True)
```

```
ds.index
```

```
RangeIndex(start=0, stop=1000, step=1)
```

### Q.40.

```
=> ds.loc[[2],['lunch','parental level of education']]
```

```
ds.loc[[2],['lunch','parental level of education']]
```

	<b>lunch</b>	<b>parental level of education</b>
<b>2</b>	standard	master's degree

### Q.41.

```
=> ds['gender']=='male'
```

```
ds['gender']=='male'
```

```
0      False
1      False
2      False
3       True
4       True
...
995     False
996      True
997     False
998     False
999     False
Name: gender, Length: 1000, dtype: bool
```

### Q.42.

```
=> ds.query('gender == "male"')
```

```
ds.query('gender == "male"')
```

	gender	level_0	index	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
3	male	3	3	group A	associate's degree	free/reduced	none	47	57	44
4	male	4	4	group C	some college	standard	none	76	78	75
7	male	7	7	group B	some college	free/reduced	none	40	43	39
8	male	8	8	group D	high school	free/reduced	completed	64	64	67
10	male	10	10	group C	associate's degree	standard	none	58	54	52
...	...	...	...	...	...	...	...	...	...	...
985	male	985	985	group A	high school	standard	none	57	51	54
987	male	987	987	group E	some high school	standard	completed	81	75	76
990	male	990	990	group E	high school	free/reduced	completed	86	81	75
994	male	994	994	group A	high school	standard	none	63	63	62
996	male	996	996	group C	high school	free/reduced	none	62	55	55

482 rows × 10 columns

Q.43.

=> ds.query('gender == "male"')[['math score','reading score']]

```
ds.query('gender == "male"')[['math score','reading score']]
```

	math score	reading score
3	47	57
4	76	78
7	40	43
8	64	64
10	58	54
...	...	...
985	57	51
987	81	75
990	86	81
994	63	63
996	62	55

482 rows × 2 columns

Q.44.

=> ds[(ds['race/ethnicity'] == "group A") & (ds['lunch'] == "free/reduced")][['writing score']]

```
ds[(ds['race/ethnicity'] == "group A") & (ds['lunch'] == "free/reduced")][['writing score']]
```

writing score	
3	44
25	72
61	34
62	55
72	48
82	54
228	64
300	81
327	19
365	60
368	58
384	43
395	41
402	55
428	53
433	50

**Q.45.**

```
=> race_ethnicity = ['group A','group D']
    filter = ds['race/ethnicity'].isin(race_ethnicity)
    ds.loc[filter,'race/ethnicity']
```

```
race_ethnicity = ['group A','group D']
filter = ds['race/ethnicity'].isin(race_ethnicity)
ds.loc[filter,'race/ethnicity']
```

```
3      group A
8      group D
11     group D
13     group A
14     group A
...
992    group D
993    group D
994    group A
998    group D
999    group D
Name: race/ethnicity, Length: 351, dtype: object
```

**Q.46.**

```
=> m= ds['parental level of education'].str.contains("associate's degree",na=False)
    ds.loc[m,'parental level of education']
```

```
m= ds['parental level of education'].str.contains("associate's degree",na=False)
ds.loc[m,'parental level of education']
```

```
3      associate's degree
5      associate's degree
10     associate's degree
11     associate's degree
19     associate's degree
...
968    associate's degree
977    associate's degree
979    associate's degree
986    associate's degree
992    associate's degree
Name: parental level of education, Length: 222, dtype: object
```