

ABHILASHA CHATTERJEE
DATA ANALYTICS & BUSINESS INTELLIGENCE BATCH 9

ASSIGNMENT 2

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

```
[2]: df = pd.read_csv(r"C:\Users\abc\Desktop\ASSINGMENTS\grade data.csv")
```

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

```
[4]: df1 = pd.read_excel(r"C:\Users\abc\Desktop\ASSINGMENTS\Grade Data.xlsx")
```

```
[5]: print(df1)
```

	S.No	NAME	GRADE	GENDER
0	1	Rubel	4	M
1	2	Ankit	info	M
2	3	Jiya	6	F
3	4	Suraj	7	M
4	5	Tanya	10	F

```
[6]: new_column_name = 'FAMILY INCOME'
df1[new_column_name] = None
```

```
[7]: new_row_data = [6, 'Naina', 3, 'F', None]
df1.loc[len(df1)] = new_row_data
```

```
[8]: print(df1.head(3))
```

	S.No	NAME	GRADE	GENDER	FAMILY INCOME
0	1	Rubel	4	M	None
1	2	Ankit	info	M	None
2	3	Jiya	6	F	None

```
[9]: print(df1.tail(3))
```

	S.No	NAME	GRADE	GENDER	FAMILY INCOME
3	4	Suraj	7	M	None

4	5	Tanya	10	F	None
5	6	Naina	3	F	None

```
[10]: df1.describe()
```

```
[10]:          S.No
count    6.000000
mean     3.500000
std      1.870829
min      1.000000
25%      2.250000
50%      3.500000
75%      4.750000
max      6.000000
```

```
[11]: df1.shape
```

```
[11]: (6, 5)
```

```
[12]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6 entries, 0 to 5
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   S.No            6 non-null      int64
1   NAME            6 non-null      object
2   GRADE           6 non-null      object
3   GENDER          6 non-null      object
4   FAMILY INCOME   0 non-null      object
dtypes: int64(1), object(4)
memory usage: 288.0+ bytes
```

```
[13]: missing_values = df1.isnull()
```

```
[14]: missing_values_count = missing_values.sum()
```

```
[15]: print(missing_values)
```

	S.No	NAME	GRADE	GENDER	FAMILY INCOME
0	False	False	False	False	True
1	False	False	False	False	True
2	False	False	False	False	True
3	False	False	False	False	True
4	False	False	False	False	True
5	False	False	False	False	True

```
[16]: print(missing_values_count)
```

```
S.No          0
NAME          0
GRADE         0
GENDER        0
FAMILY INCOME 6
dtype: int64
```

```
[17]: first_column = df1['S.No']
```

```
[18]: first_column = df['S.No']
second_column = df['NAME']
```

```
[19]: first_row = df1.iloc[0]
```

```
[20]: first_and_second_row = df1.iloc[0:2]
```

```
[21]: df2 = df1.dropna()
```

```
[22]: df1_filled = df1.fillna(df1.mean())
```

C:\Users\abc\AppData\Local\Temp\ipykernel_5488\1078566346.py:1: FutureWarning:
The default value of numeric_only in DataFrame.mean is deprecated. In a future
version, it will default to False. In addition, specifying 'numeric_only=None'
is deprecated. Select only valid columns or specify the value of numeric_only to
silence this warning.

```
df1_filled = df1.fillna(df1.mean())
```

```
[23]: subset_male = df1[df1['GENDER'] == 'Male']
```

```
[32]: subset_female = df1[df1['GENDER'] == 'Female']
```

```
[33]: subset_female = df1[df1['GRADE'] >= 7]
```

TypeError

Traceback (most recent call last)

Cell In[33], line 1

```
----> 1 subset_female = df1[df1['GRADE'] >= 7]
```

File E:\Users\abhilasha\jupyter\Lib\site-packages\pandas\core\ops\common.py:72,
in _unpack_zerodim_and_defer.<locals>.new_method(self, other)

```
68         return NotImplemented
```

```
70 other = item_from_zerodim(other)
```

```
---> 72 return method(self, other)
```

```

File E:\Users\abhilasha\jupyter\Lib\site-packages\pandas\core\arraylike.py:62, in OpsMixin.__ge__(self, other)
    60 @unpack_zerodim_and_defer("__ge__")
    61 def __ge__(self, other):
--> 62     return self._cmp_method(other, operator.ge)

File E:\Users\abhilasha\jupyter\Lib\site-packages\pandas\core\series.py:6243, in Series._cmp_method(self, other, op)
    6240 rvalues = extract_array(other, extract_numpy=True, extract_range=True)
    6242 with np.errstate(all="ignore"):
-> 6243     res_values = ops.comparison_op(lvalues, rvalues, op)
    6245 return self._construct_result(res_values, name=res_name)

File E:\Users\abhilasha\jupyter\Lib\site-packages\pandas\core\ops\array_ops.py:
287, in comparison_op(left, right, op)
    284     return invalid_comparison(lvalues, rvalues, op)
    286 elif is_object_dtype(lvalues.dtype) or isinstance(rvalues, str):
--> 287     res_values = comp_method_OBJECT_ARRAY(op, lvalues, rvalues)
    289 else:
    290     res_values = _na_arithmetic_op(lvalues, rvalues, op, is_cmp=True)

File E:\Users\abhilasha\jupyter\Lib\site-packages\pandas\core\ops\array_ops.py:
75, in comp_method_OBJECT_ARRAY(op, x, y)
    73     result = libops.vec_compare(x.ravel(), y.ravel(), op)
    74 else:
--> 75     result = libops.scalar_compare(x.ravel(), y, op)
    76 return result.reshape(x.shape)

File E:\Users\abhilasha\jupyter\Lib\site-packages\pandas\_libs\ops.pyx:107, in pandas._libs.ops.scalar_compare()

TypeError: '>=' not supported between instances of 'str' and 'int'

```

```
[34]: df.mean()
```

```

C:\Users\abc\AppData\Local\Temp\ipykernel_5488\3698961737.py:1: FutureWarning:
The default value of numeric_only in DataFrame.mean is deprecated. In a future
version, it will default to False. In addition, specifying 'numeric_only=None'
is deprecated. Select only valid columns or specify the value of numeric_only to
silence this warning.

```

```
df.mean()
```

```

[34]: S.No      3.0
      dtype: float64

```