

```
In [1]: #imports

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

Homework 1: CSE 708

```
In [2]: #Question 1

ten = pd.Series(x for x in range(0, 10))
ten
```

```
Out[2]: 0    0
        1    1
        2    2
        3    3
        4    4
        5    5
        6    6
        7    7
        8    8
        9    9
dtype: int64
```

```
In [3]: #Question 2

ten.to_list()
```

```
Out[3]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [4]: #Question 3

dic = {'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 800}

print("dic: ", dic)
ser = pd.Series(dic)
print(ser)
```

```
dic: {'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 800}
a    100
b    200
c    300
d    400
e    800
dtype: int64
```

Dataframe

```
In [5]: d = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
                    'score': [12.5, 9.0, 16.5, np.nan, 9.0, 20.0, 14.0, np.nan, 8.0, 19.0],
                    'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
                    'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
df = pd.DataFrame(d)
df
```

Out[5]:

	name	score	attempts	qualify
0	Anastasia	12.5	1	yes
1	Dima	9.0	3	no
2	Katherine	16.5	2	yes
3	James	NaN	3	no
4	Emily	9.0	2	no
5	Michael	20.0	3	yes
6	Matthew	14.0	1	yes
7	Laura	NaN	1	no
8	Kevin	8.0	2	no
9	Jonas	19.0	1	yes

In [6]:

```
#Question 4
name = df.sort_values(by=['name', 'score'], ascending=[False, True])
name
```

Out[6]:

	name	score	attempts	qualify
5	Michael	20.0	3	yes
6	Matthew	14.0	1	yes
7	Laura	NaN	1	no
8	Kevin	8.0	2	no
2	Katherine	16.5	2	yes
9	Jonas	19.0	1	yes
3	James	NaN	3	no
4	Emily	9.0	2	no
1	Dima	9.0	3	no
0	Anastasia	12.5	1	yes

In [7]:

```
#Question 5
not_recorded = df[df['score'].isnull()]['name']
not_recorded
```

Out[7]:

```
3    James
7    Laura
Name: name, dtype: object
```

In [8]:

```
#Question 6
attempts = df[df['attempt'] >= 2]['name']
attempts
```

```

-----
KeyError                                Traceback (most recent call last)
~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method,
tolerance)
    3360             try:
-> 3361                 return self._engine.get_loc(casted_key)
    3362             except KeyError as err:

~\anaconda3\lib\site-packages\pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.g
et_loc()

~\anaconda3\lib\site-packages\pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.g
et_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_
item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_
item()

KeyError: 'attempt'

```

The above exception was the direct cause of the following exception:

```

KeyError                                Traceback (most recent call last)
<ipython-input-8-98c97e81adb8> in <module>
      1 #Question 6
----> 2 attempts = df[df['attempt']>= 2]['name']
      3 attempts

~\anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
    3453         if self.columns.nlevels > 1:
    3454             return self._getitem_multilevel(key)
-> 3455         indexer = self.columns.get_loc(key)
    3456         if is_integer(indexer):
    3457             indexer = [indexer]

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method,
tolerance)
    3361             return self._engine.get_loc(casted_key)
    3362         except KeyError as err:
-> 3363             raise KeyError(key) from err
    3364
    3365         if is_scalar(key) and isna(key) and not self.hasnans:

KeyError: 'attempt'

```

```

In [ ]: #Question 7
qualified = df[df['qualify'] == 'yes']['name'] & df[df['attempt'] == 1]['name']
qualified

```

In []:

In []:

