Assignment 2

April 21, 2020

[11]:

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

1 Assignment 2

1.1 Question 1

Generate a Series with the first 10 prime numbers

[2]:

```
arr = []
i = 2

while len(arr) < 10:
    prime = True
    for j in range(2,i):
        if (i % j) == 0:
            prime = False
            break
    if prime:
        arr.append(i)
        if len(arr) == 10:
            break
    i += 1
ans1 = pd.Series(arr)
print(ans1)
```

```
0     2
1     3
2     5
3     7
4    11
5    13
6    17
7    19
8    23
9    29
dtype: int64
```

1.2 Question 2

Select the prime numbers at the odd positions of the Series, using `iloc`

[3]:
```
ans2 = ans1.iloc[1:10:2]
print(ans2)
```

```
1     3
3     7
5    13
7    19
9    29
dtype: int64
```

1.3 Question 3

Change the index of the Series to be the characters *a* through *j*

[8]:
```
ans3 = ans1
ans3.index = ['a','b','c','d','e','f','g','h','i','j']
print(ans3)
```

```
a     2
b     3
c     5
d     7
e    11
f    13
g    17
h    19
i    23
j    29
dtype: int64
```

1.4 Question 4

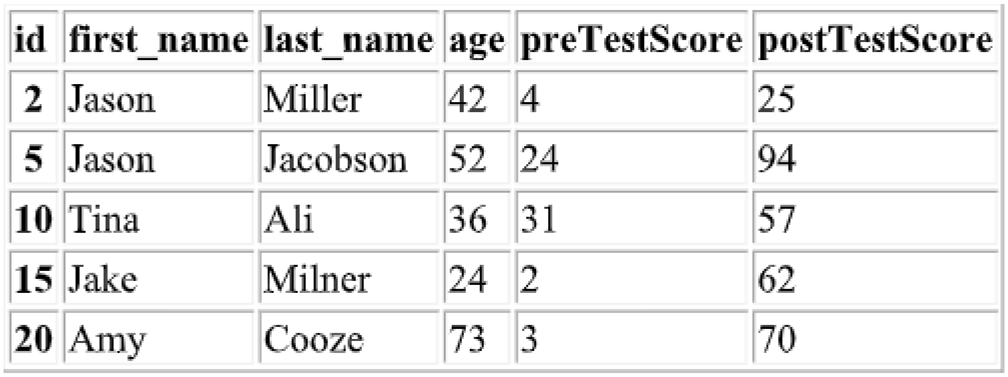
Write code that selects the numbers at the odd positions of Series using the `loc` method

[9]:
```
ans4 = ans1.loc['b':'j':2]
print(ans4)
```

```
b     3
d     7
f    13
h    19
j    29
dtype: int64
```

1.5 Question 5

Create the following DataFrame (index not shown):



id	first_name	last_name	age	preTestScore	postTestScore
2	Jason	Miller	42	4	25
5	Jason	Jacobson	52	24	94
10	Tina	Ali	36	31	57
15	Jake	Milner	24	2	62
20	Amy	Cooze	73	3	70

```
[17]: row1 = [2,'Jason','Miller',42,4,25]
      row2 = [5,'Jason','Jacobson',52,24,94]
      row3 = [10,'Tina','Ali',36,31,57]
      row4 = [15,'Jake','Milner',24,2,62]
      row5 = [20,'Amy','Cooze',73,3,70]
      columns = ['id','first_name','last_name','age','preTestScore','postTestScore']
      ans5 = pd.DataFrame([row1,row2,row3,row4,row5],columns=columns)
      display(ans5)
```

```
   id first_name last_name  age  preTestScore  postTestScore
0   2      Jason    Miller   42             4             25
1   5      Jason  Jacobson   52            24             94
2  10       Tina       Ali   36            31             57
3  15       Jake    Milner   24             2             62
4  20        Amy     Cooze   73             3             70
```

1.6 Question 6

Make id the index attribute

```
[21]: ans5.index = [2,5,10,15,20]
      ans6 = ans5.drop(columns=['id'])
      ans6.index.name = 'id' # Also naming the index 'id'
      display(ans6)
```

```
   first_name last_name  age  preTestScore  postTestScore
id
2       Jason    Miller   42             4             25
5       Jason  Jacobson   52            24             94
10       Tina       Ali   36            31             57
15       Jake    Milner   24             2             62
20        Amy     Cooze   73             3             70
```

1.7 Question 7

Show the first names in the DataFrame

[28]:

```
ans7 = ans6['first_name']
display(ans7)
```

```
id
2     Jason
5     Jason
10     Tina
15     Jake
20      Amy
Name: first_name, dtype: object
```

1.8 Question 8

Show the age of Tina

[26]:

```
ans8 = ans6[ans6['first_name'] == 'Tina']['age'].iloc[0]
print(ans8)
```

```
36
```

1.9 Question 9

What is the mean difference between the preTestScore and postTestScore?

[31]:

```
ans6['diffTestScore'] = ans6['postTestScore'] - ans6['preTestScore']
ans9 = ans6['diffTestScore'].mean()
print(ans9)
```

```
48.8
```

1.10 Question 10

Make the postTestScore of Amy and Jake to be `NaN`

[37]:

```
ans10 = ans6
ans10.loc[ans10['first_name'] == 'Amy','postTestScore'] = np.nan
ans10.loc[ans10['first_name'] == 'Jake','postTestScore'] = np.nan
display(ans10)
```

```
   first_name last_name  age  preTestScore  postTestScore  diffTestScore
id
2       Jason    Miller   42             4           25.0             21
5       Jason  Jacobson   52            24           94.0             70
```

```
10        Tina       Ali   36            31          57.0            26
15        Jake    Milner   24             2           NaN            60
20         Amy     Cooze   73             3           NaN            67
```

1.11 Question 11

Show all rows in the DataFrame, where all values are different from `NaN`

```
[40]: ans11 = ans10.dropna(axis=0)
      display(ans11)
```

```
   first_name last_name  age  preTestScore  postTestScore  diffTestScore
id
2       Jason    Miller   42             4           25.0             21
5       Jason  Jacobson   52            24           94.0             70
10       Tina       Ali   36            31           57.0             26
```

1.12 Question 12

Reset the index, then make the first name and last name together to be the index

```
[42]: ans12 = ans11.reset_index()
      ans12 = ans12.set_index(['first_name','last_name'])
      display(ans12)
```

```
                      id  age  preTestScore  postTestScore  diffTestScore
first_name last_name
Jason      Miller      2   42             4           25.0             21
           Jacobson    5   52            24           94.0             70
Tina       Ali        10   36            31           57.0             26
```

1.13 Question 13

Show the age of Tina using the DataFrame

```
[45]: ans13 = ans12.loc['Tina']['age'].iloc[0]
      print(ans13)
```

```
36
```

```
[ ]:
```