Week 4_ Quiz #4 (Working)

March 13, 2021

1 Week 4 Quiz 4 (Solutions)

1.1 Q1

a is the array of numbers 0 to 7. b is the array of 4 to 5. All elements of b are changed to 40. a[4] and a[5] are hence changed. Hence, a[4] + a[6] = 40 + 6 = 46

[1]:
```
import numpy as np

a = np.arange(8)
b = a[4:6]
b[:] = 40
c = a[4] + a[6]
c
```

[1]:
```
46
```

1.2 Q2

There is a difference between the re.search() and re.match(). Both return the first match of a substring found in the string, but re. match() searches only from the beginning of the string and return **match object** if found. But if a match of substring is found somewhere in the middle of the string, it returns none. So re.match() == True will return false, but the boolean applied to re.match() is true.

However, re.search returns a match object, which does not have a len property.

[2]:
```
import re
s = 'ABCAC'
```

[3]:
```
bool(re.match('A',s)) == True
```

[3]:
```
True
```

1.3 Q3

[4]:
```
def result():
    s = 'ACAABAACAAABACDBADDDFSDDDFFSSSASDAFAAACBAAAFASD'

    result = []
```

```
    # compete the pattern below
    pattern = "(\w)(?=AAA)"
    for item in re.finditer(pattern, s):
      # identify the group number below.
      result.append(item.group())

    return result
```

[5]:
```
result()
```

[5]:
```
['C', 'F', 'B']
```

1.4 Q4

df.index[0] will return the value 'd'.

[6]:
```
df = pd.Series({'d': 4, 'b':7, 'a':-5, 'c':3})
df.index[0]
```

```
␣
↪---------------------------------------------------------------------------

NameError                                 Traceback (most recent call␣
↪last)

<ipython-input-6-7eac2daecabc> in <module>
----> 1 df = pd.Series({'d': 4, 'b':7, 'a':-5, 'c':3})
      2 df.index[0]


NameError: name 'pd' is not defined
```

1.5 Q5

s3: Mango -> 40, Strawberry ->35, Blueberry->18, Vanilla-> 61, Plain->20, Banana- >15

[7]:
```
s1 = pd.Series([20,15,18,31],index = ['Mango','Strawberry', 'Blueberry',␣
↪'Vanilla'])
s2 = pd.Series([20,30,15,20,20],index = ['Strawberry', 'Vanilla','Banana',␣
↪'Mango','Plain'])
s3 = s1.add(s2)
```

```
␣
↪---------------------------------------------------------------------------
```

```
NameError                                 Traceback (most recent call␣
↪last)

<ipython-input-7-9ec8f8ab340c> in <module>
----> 1 s1 = pd.Series([20,15,18,31],index = ['Mango','Strawberry',␣
↪'Blueberry', 'Vanilla'])
      2 s2 = pd.Series([20,30,15,20,20],index = ['Strawberry',␣
↪'Vanilla','Banana', 'Mango','Plain'])
      3 s3 = s1.add(s2)


NameError: name 'pd' is not defined
```

[8]:
```
s3 = s1.add(s2, fill_value = 0)
## Notice the change in datatype!! This results in this expression to not be␣
 ↪able to hold:
# s3['Blueberry'] == s1.add(s2, fill_value = 0)['Blueberry'] due to the␣
 ↪difference in type.
```

```
␣
↪---------------------------------------------------------------------------

NameError                                 Traceback (most recent call␣
↪last)

<ipython-input-8-4eb4efc6970d> in <module>
----> 1 s3 = s1.add(s2, fill_value = 0)
      2 ## Notice the change in datatype!! This results in this expression␣
↪to not be able to hold:
      3 # s3['Blueberry'] == s1.add(s2, fill_value = 0)['Blueberry'] due to␣
↪the difference in type.


NameError: name 's1' is not defined
```

[9]:
```
s3['Mango'] >=  s1.add(s2, fill_value = 0)['Mango']
```

```
␣
↪---------------------------------------------------------------------------

NameError                                 Traceback (most recent call␣
↪last)
```

```
    <ipython-input-9-eae82d2e455b> in <module>
----> 1 s3['Mango'] >=  s1.add(s2, fill_value = 0)['Mango']


NameError: name 's3' is not defined
```

1.6 Q6

- Every time we call df.set_index(), the old index will be discarded.
- Every time we call df.reset_index(), the old index will be set as a new column.

[10]:
```
df = pd.DataFrame({'month': [1, 4, 7, 10],
                   'year': [2012, 2014, 2013, 2014],
                   'sale': [55, 40, 84, 31]})
```

```
␣
↪---------------------------------------------------------------------------

NameError                                 Traceback (most recent call␣
↪last)

<ipython-input-10-bc84512d2c99> in <module>
----> 1 df = pd.DataFrame({'month': [1, 4, 7, 10],
      2                    'year': [2012, 2014, 2013, 2014],
      3                    'sale': [55, 40, 84, 31]})


NameError: name 'pd' is not defined
```

[11]:
```
df.set_index('month')
```

```
␣
↪---------------------------------------------------------------------------

NameError                                 Traceback (most recent call␣
↪last)

<ipython-input-11-745de75fcb61> in <module>
----> 1 df.set_index('month')


NameError: name 'df' is not defined
```

[12]:
```
df = df.reset_index()
df
```

```
      ␣
↪---------------------------------------------------------------------------

       NameError                                 Traceback (most recent call␣
↪last)

       <ipython-input-12-d5b3a80c472f> in <module>
   ----> 1 df = df.reset_index()
         2 df


       NameError: name 'df' is not defined
```

1.7 Q7

Apparently, S['b':'e'] is wrong as the 2nd parameter is included.

[13]:
```
S = pd.Series(np.arange(5), index=['a', 'b', 'c', 'd', 'e'])
S['b':'e']
```

```
      ␣
↪---------------------------------------------------------------------------

       NameError                                 Traceback (most recent call␣
↪last)

       <ipython-input-13-b85964d24c7a> in <module>
   ----> 1 S = pd.Series(np.arange(5), index=['a', 'b', 'c', 'd', 'e'])
         2 S['b':'e']


       NameError: name 'pd' is not defined
```

1.8 Q8

Since, no axis parameter is given, by default the function will apply down the column. Since the resulting structure is a series, df_new[1] will have the value of 82 + 6 = **88**

[14]:
```
df = pd.DataFrame({'a': [5, 5 ,71, 67],
                   'b': [6,82,31,37],
                   'c': [20,28,92,49]})
f = lambda x: x.max() + x.min()
```

```
df_new = df.apply(f)
df_new
```

```
␣
↪---------------------------------------------------------------------------

NameError                                 Traceback (most recent call␣
↪last)

<ipython-input-14-ac4a7facdcf5> in <module>
----> 1 df = pd.DataFrame({'a': [5, 5 ,71, 67],
      2                     'b': [6,82,31,37],
      3                     'c': [20,28,92,49]})
      4 f = lambda x: x.max() + x.min()
      5 df_new = df.apply(f)

NameError: name 'pd' is not defined
```

[15]:
```
df_new[1]
```

```
␣
↪---------------------------------------------------------------------------

NameError                                 Traceback (most recent call␣
↪last)

<ipython-input-15-70912f7dde75> in <module>
----> 1 df_new[1]

NameError: name 'df_new' is not defined
```

1.9 Q9

My guess is to new_df.stack().stack().unstack() but there is no such option. So I guess the sensible option is to stack() and then call the transpose of the dataframe.

[16]:
```
df = pd.DataFrame({'a': [5, 5 ,71, 67],
                   'b': [6,82,31,37],
                   'c': [20,28,92,49]})
```

```
      ␣
↪---------------------------------------------------------------------------

      NameError                                 Traceback (most recent call␣
↪last)

      <ipython-input-16-ff1f23594239> in <module>
----> 1 df = pd.DataFrame({'a': [5, 5 ,71, 67],
      2                    'b': [6,82,31,37],
      3                    'c': [20,28,92,49]})


      NameError: name 'pd' is not defined
```

[ ]:

1.10 Q10

Sum does not take into account the NaN values. So item1 willl have a value of NaN.

Wrong: Groupby will create two groups from this df which correspond to item_1 and item_2. Calling sum() will add the quantities sold for each item across all 3 stores (A, B and C). iloc[0] will get the first row, which corresponds to item_1 and the sum of 'Quantity sold' values for item_1 will be calculated ignoring NaN, which comes out to be 30.

[17]:
```
sum does not take i
```

```
      File "<ipython-input-17-f98d68ea4457>", line 1
    sum does not take i
           ^
SyntaxError: invalid syntax
```

[ ]:
```
wow! This is soooo tricky!
```