FREQUENTLY ASKED QUESTIONS:

PYTHON: PANDAS

PART 2

FAQ

1) Define the different ways a DataFrame can be created in pandas?

We can create a DataFrame using following ways:

- Lists
- Dict of ndarrays

Example 1 - Create a DataFrame using List:

```
import pandas as pd
# a list of strings
a = ['Workearly', 'Pandas']
# Calling DataFrame constructor on list
info = pd.DataFrame(a)
print(info)

```

Output:

```
           0
0  Workearly
1     Pandas
```

Example 2 - Create a DataFrame from dict of ndarrays:

```
import pandas as pd
info = {'ID' :[101, 102, 103],'Department' :['B.Sc','B.Tech','M.Tech',]}
info = pd.DataFrame(info)
print (info)

```

Output:

```
    ID Department
0  101       B.Sc
1  102    B.Tech
2  103    M.Tech
```

FAQ

2) How will you create a series from dict in Pandas?

A Series is defined as a one-dimensional array that is capable of storing various data types.
We can create a Pandas Series from Dictionary:

Create a Series from dict:

We can also create a Series from dict. If the dictionary object is being passed as an input and the index is not specified, then the dictionary keys are taken in a sorted order to construct the index.
If index is passed, then values correspond to a particular label in the index will be extracted from the dictionary.
You can iterate over the rows of the DataFrame by using for loop in combination with an iterrows() call on the DataFrame.

```
import pandas as pd

info = {'x' : 0., 'y' : 1., 'z' : 2.}
a = pd.Series(info)
print (a)

```

Output:

```
x    0.0
y    1.0
z    2.0
dtype: float64
```

FAQ

3) How will you add a column to a pandas DataFrame?

We can add any new column to an existing DataFrame. The below code demonstrates how to add any new column to an existing DataFrame:

```
import pandas as pd
info = {'one' : pd.Series([1, 2, 3, 4, 5], index=['a', 'b', 'c', 'd', 'e']),
             'two' : pd.Series([1, 2, 3, 4, 5, 6], index=['a', 'b', 'c', 'd', 'e', 'f'])}

info = pd.DataFrame(info)

# Add a new column to an existing DataFrame object

print ("Add new column by passing series")
info['three']=pd.Series([20,40,60],index=['a','b','c'])
print (info)
print ("Add new column using existing DataFrame columns")
info['four']=info['one']+info['three']
print (info)

```

Output:

```
Add new column by passing series
   one  two  three
a  1.0    1   20.0
b  2.0    2   40.0
c  3.0    3   60.0
d  4.0    4    NaN
e  5.0    5    NaN
f  NaN    6    NaN
Add new column using existing DataFrame columns
   one  two  three  four
a  1.0    1   20.0  21.0
b  2.0    2   40.0  42.0
c  3.0    3   60.0  63.0
d  4.0    4    NaN   NaN
e  5.0    5    NaN   NaN
f  NaN    6    NaN   NaN
```

4) How to get the items of series A not present in series B?

We can remove items present in p2 from p1 using isin() method.

```
import pandas as pd
p1 = pd.Series([2, 4, 6, 8, 10])
p2 = pd.Series([8, 10, 12, 14, 16])
p1[~p1.isin(p2)]

```

FAQ

Solution:

```
0    2
1    4
2    6
dtype: int64
```

5) How to get the items not common to both series A and series B?

We get all the items of p1 and p2 not common to both using below example:

```
import pandas as pd
import numpy as np
p1 = pd.Series([2, 4, 6, 8, 10])
p2 = pd.Series([8, 10, 12, 14, 16])
p1[~p1.isin(p2)]
p_u = pd.Series(np.union1d(p1, p2))  # union
p_i = pd.Series(np.intersect1d(p1, p2))  # intersect
p_u[~p_u.isin(p_i)]
```

Output:

```
0     2
1     4
2     6
5    12
6    14
7    16
dtype: int64
```

6) How to get the minimum, 25th percentile, median, 75th, and max of a numeric series?

We can compute the minimum, 25th percentile, median, 75th, and maximum of p as below example:

FAQ

```
import pandas as pd
import numpy as np
p = pd.Series(np.random.normal(14, 6, 22))
state = np.random.RandomState(120)
p = pd.Series(state.normal(14, 6, 22))
np.percentile(p, q=[0, 25, 50, 75, 100])

```

Output:

```
array([ 4.61498692, 12.15572753, 14.67780756, 17.58054104, 33.24975515])
```

7) How to get frequency counts of unique items of a series?

We can calculate the frequency counts of each unique value p as below example:

```
import pandas as pd
import numpy as np
p= pd.Series(np.take(list('pqrstu'), np.random.randint(6, size=17)))
p = pd.Series(np.take(list('pqrstu'), np.random.randint(6, size=17)))
p.value_counts()

```

Output:

```
s    4
r    4
q    3
p    3
u    3
```

FAQ

8) How to convert a numpy array to a dataframe of given shape?

We can reshape the series p into a dataframe with 6 rows and 2 columns as below example:

```
import pandas as pd
import numpy as np
p = pd.Series(np.random.randint(1, 7, 35))
# Input
p = pd.Series(np.random.randint(1, 7, 35))
info = pd.DataFrame(p.values.reshape(7,5))
print(info)

```

Output:

```
   0  1  2  3  4
0  5  3  6  1  2
1  3  5  1  3  5
2  5  5  4  4  4
3  5  1  4  1  6
4  6  3  2  4  1
5  4  2  5  1  1
6  1  1  6  6  3
```

9) How can we convert a Series to DataFrame?

The Pandas Series.to_frame() function is used to convert the series object to the DataFrame.

- Series.to_frame(name=None)

name: Refers to the object. Its Default value is None. If it has one value, the passed name will be substituted for the series name.

```
import pandas as pd
s = pd.Series(["a", "b", "c"],name="vals")
s.to_frame()

```

FAQ

Output:

```
   vals
0     a
1     b
2     c
```