In [1]:

```
import warnings
warnings.filterwarnings(action='ignore')
```

Q: 1. How to import pandas? Show the example with Python

In [2]:

```
import pandas as pd
```

Q: 2. How to create a series from a list, numpy array and dict? Show the example with Python

In [3]:

```
import numpy as np

mylist = list('abc')
myarr = np.arange(3)
mydict = dict(zip(mylist, myarr))

series1 = pd.Series(mylist)
series2 = pd.Series(myarr)
series3 = pd.Series(mydict)

print(series1, '₩n')
print(series2, '₩n')
print(series3, '₩n')
```

```
0    a
1    b
2    c
dtype: object

0    0
1    1
2    2
dtype: int32

a    0
b    1
c    2
dtype: int64
```

Q: 3. How to convert the index of a series into a column of a data frame? Show the example with Python

In [4]:

```
mylist = list('abc')
myarr = np.arange(3)
mydict = dict(zip(mylist, myarr))
series = pd.Series(mydict)

df = series.to_frame().reset_index()

print(df)
```

```
  index  0
0     a  0
1     b  1
2     c  2
```

Q: 4. How to get the items not common to both series A and series B? Show the example with Python

In [5]:

```
series1 = pd.Series([3, 4, 5])
series2 = pd.Series([4, 5, 6, 7])

series_union = pd.Series(np.union1d(series1, series2))
series_intersect = pd.Series(np.intersect1d(series1, series2))
series_union[~series_union.isin(series_intersect)]
```

Out[5]:

```
0    3
3    6
4    7
dtype: int64
```

1. When input is as follows:
   Input = pd.Series(np.random(10,5, 20))
   Computer the minimum, 25th percentile, median, 75th and maximum of Input

In [6]:

```
state = np.random.RandomState(100)
series = pd.Series(state.normal(10, 5, 20))
np.percentile(series, q=[0, 25, 50, 75, 100])
```

Out[6]:

```
array([ 1.25117263,  7.61829833, 10.29192131, 12.76922166, 15.76517901])
```

1. When input is as follows:
   Input = pd.Series(np.take(list('abcdefgh'), np.random.randint(8, size=30)))
   Compute to get frequency counts of unique items of the series

In [7]:

```
Input = pd.Series(np.take(list('abcdefgh'), np.random.randint(8, size=30)))

Input.value_counts()
```

Out[7]:

```
g    6
b    5
a    4
d    4
f    3
h    3
c    3
e    2
dtype: int64
```

1. When input & position is as follows:
   Input = pd.Series(list('abcdefghijklmnopqrstuvwxyz'))
   pos = [0, 4, 8, 14, 20]
   Compute how to extract items at given positions from ‘pos’

In [8]:

```
Input = pd.Series(list('abcdefghijklmnopqrstuvwxyz'))
pos = [0, 4, 8, 14, 20]

Input.take(pos)
```

Out[8]:

```
0     a
4     e
8     i
14    o
20    u
dtype: object
```

1. When both truth and predicted are as follows:
   truth = pd.Series(range(10))
   pred = pd.Series(range(10)) + np.random.random(10)
   Compute the mean squared error

In [9]:

```
truth = pd.Series(range(10))
pred = pd.Series(range(10)) + np.random.random(10)

np.mean((truth-pred)**2)
```

Out[9]:

```
0.3465686714577367
```

In [ ]: