Python Pandas

1. Differentiate pivot() and pivot_table().
2. Name the function used to sort the data in a dataframe.
3. Give syntax for sort_values() and explain its arguments.
4. What is the default value for 'inplace' argument in sort_values()?
5. What is the default value for 'na_position' argument in sort_values()?
6. Write Program to sort the above dataframe by Rupees.
7. Write program to sort the above dataframe by 'Company' in ascending order and USD in descending order.
8. Which of the following is/are correct?
   (i) df.sort_values(by=['Item','Company'])
   (ii) df.sort_values(['Item','Company'])
   (iii) df.sort_values(by=['Item','Company'],ascending=[False])
   (iv) df.sort_values(by=['Item','Company'], ascending=False)
9. What is the use of sort_index() function?

1. List out the aggregate functions that we used in the dataframe.
2. Expand MAD.
3. What is MAD? What function can be used to calculate it?
4. Give syntax for mad() and explain its arguments.
5. Write program to find the Mean Absolute Deviation for Rupees column in the above DataFrame. Also write the output.

1. Write program to find the standard deviation for USD column in the above DataFrame.
2. Find the output:

```
import pandas as pd
d={'scor':[2,5,8,7]}
df=DataFrame(d)
print(df)
print(df.std())
```

3. What is a histogram?
4. What is the difference between bar graph and histogram?
5. Why there is no gap between bars in the histogram?
6. What function is used to create a histogram?

1. Does the hist() creates histogram for all the columns of a dataframe?
2. Write code to create histogram for USD column of above DataFrame.
3. How many histograms does each of the following code generate?
   (i)
```
import pandas as pd
d={'score':[2,5,8,7],'age':[10,10,20,25]}
df=pd.DataFrame(d)
print(df.hist())
```
   (ii)
```
import pandas as pd
d={'name':['AA','BB','CC','DD'],'age':[10,10,20,25]}
```

```
df=pd.DataFrame(d)
print(df.hist())
```

4. What is function application?
5. Mention the different ways in which a function may be applied on a dataframe using function application.
6. Name the 3 functions used for function application and write their uses.

1. What is a pipe technique? What function is used for the same in Pandas? Can we do the same without using pipe()? If yes, what is the name of that style?
2. Give an example for sandwiching style of using functions.
3. What is the need for pipe() function when we can do the same using sandwiching style?
4. Name the function used to chain the functions in the order they should execute.
5. Give an example for pipe() function.
6. Rewrite the following code using pipe() function:
   df.add(div(power(sqrt(),2),3),100)
7. Find the error in the following code:
   df. pipe(add(),3).pipe(power(),2)

1. Give the syntax for pipe() function and explain its arguments.
2. What will be returned by the pipe() function?
3. Can we use numpy functions on pipes()?
4. df:

	2016	**2017**	**2020**
Q1	34	21	40
Q2	35	28	45
Q3	38	30	47

Find the output for the following code w.r.t the above dataframe:

```
print(df.pipe(np.add,45.37).pipe(np.floor,))
```

1. Which is the Series function among the following:
   (a) apply (b) applymap()
2. Which is the element function among the following:
   (b) apply (b) applymap()
3. Give the syntax for the functions apply() and applymap() and explain its arguments.

1.Give an example for each: (i) apply() (ii) applymap()
2. Find the output with reference to above dataframe:

```
print(df.apply(np.mean,axis=1))
print(df.applymap(np.mean))
```

1. In what situation the function apply() will behave like applymap()? Give example.
2. Will there be any difference in the output of the following:
   (i) print(df.apply(np.sqrt))
   (ii) print(df.applymap(np.sqrt))

1. Write the use of groupby() function.

2. Give the syntax for groupby() and explain its arguments.
3. What does the groupby() function returns.
4. What does the following attributes or functions does when used with grouped data:
   (i) groups (ii) get_group() (iii) size() (iv) count (v)[<columnname>].head()

1. Can we create groups on multiple columns. If yes, give example.
2. df

Classes	Country	Quarter	Tutor
28	USA	1	Tahira
36	UK	1	Jacob
41	Japan	2	Venkat
32	USA	2	Tahira
40	USA	3	Venkat
40	UK	3	Tahira

Find the Output:

```
df1=df.groupby(['Tutor','Country'])
print(df1.groups)
print(df1.get_group(('Tahira','USA')))
print(df1.size())
print(df1.count())
print(df1['classes'].head())
print(df1.get_group(('Jacob','USA')))
```

3. Give syntax for agg() method and explain its basic arguments.
4. Can we combine groupby() and agg() in single command. Give example.
5. Find the output:

```
print(df1.agg([np.sum,np.mean]))
```

1. Write the use of transform() function.
2. Find the output:

```
print(df1.groupby('Tutor').transform(np.mean))
```

1. Differentiate agg() and transform()
2. Find output:

```
print(df2.groupby('Tutor')['Classes'].transform(np.mean))
df2['Classmean']=df1.groupby('Tutor') ['Classes'].transform(np.mean)
print(df2)
```

1. Write the uses of the following functions:
   (i) rename() (ii) reindex() (iii) reindex_like()

1. Give the 2 different syntax for rename() function and explain its arguments.
2. Find output:

df:

	2016	2017	2020
Q1	34	27	51
Q2	25	28	52
Q4	29	40	75

```
print(df.rename(index={'Q1':10,'Q2':20,'Q4':40})
```

print(df)
print(df.rename(columns={2016:16,2017:17,2020:20},inplace=True))
print(df)
print(df.rename({'Q1':10,'Q2':20,'Q4':40},axis=0))

3. What will happen if we give extra labels while using rename().
4. Give the 2 different syntax for reindex() and explain its arguments.
5. Write the output:
   print(df.reindex(['Q4','Q1','Q2'],axis=0))
   print(df.reindex(index=['Q4','Q1','Q2'],axis=0))
   print(df.reindex(index=['Q1','Q2','Q3','Q4']))
   print(df.reindex(index=[' Q1','Q2','Q3','Q4'],fill_value=1000))
6. Explain the working of reindex_like() function.
7. Find the output:
   df1:

	2016	**2017**	**2018**
Q1	40	70	20
Q2	50	80	30
Q3	60	90	40

print(df.reindex_like(df1))