## **Pandas Assignment**

Q1. How do you load a CSV file into a Pandas DataFrame?

Ans) Using the pd.read\_csv() method we can load CSV file into a Pandas Data Frame. Load the CSV into a DataFrame: import pandas as pd.

df=pd.read\_csv('https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv')

Q2. How do you check the data type of a column in a Pandas DataFrame?

Ans) To check the data type in pandas DataFrame we can use the “dtype” attribute. The attribute returns a series with the data type of each column. And the column names of the DataFrame are represented as the index of the resultant series object and the corresponding data types are returned as values of the series object.

Syntax1: df.dtypes

Syntax2:

lst = [1,2,3,4]

ser = pd.Series(lst)

print(type(ser)) #output: <class 'pandas.core.series.Series'>

Q3. How do you select rows from a Pandas DataFrame based on a condition?

Ans) df.loc[30:41] #Display rows from 30 to 40 in a dataframe.

Q4. How do you rename columns in a Pandas DataFrame?

Ans) Using the rename function we can rename the columns in a Pandas DataFrame.

Syntax: df2.rename(columns={'Name': 'Name1', 'Address': 'Location'})

Q5. How do you drop columns in a Pandas DataFrame?

Ans) Using drop function we can drop columns in a Pandas DataFrame.

Syntax: df2.drop('Age', axis=1, inplace = True)

Q6. How do you find the unique values in a column of a Pandas DataFrame?

Ans) By using unique() function we can get values from a Pandas DataFrame.

Syntax: print(df2['Age'].unique())

Q7. How do you find the number of missing values in each column of a Pandas DataFrame?

Ans) If the total number of missing values is not zero, it means pandas. DataFrame contains at least one missing value.

Syntax: df.isnull()

Q8. How do you fill missing values in a Pandas DataFrame with a specific value?

Ans) The fillna() method replaces the NULL values with a specified value.

The fillna() method returns a new DataFrame object unless the inplace parameter is set to True, in that case the fillna() method does the replacing in the original DataFrame instead.

Syntax: df = pd.read\_csv(‘data.csv’)

Newdf = df.fillna(2222)

Q9. How do you concatenate two Pandas DataFrames?

Ans) To concatenate an arbitrary number of pandas objects ( DataFrame or Series ), use concat. The concat() function in pandas is used to append either columns or rows from one DataFrame to another. The concat() function does all the heavy lifting of performing concatenation operations along an axis while performing optional set logic (union or intersection) of the indexes (if any) on the other axes.

Syntax:

frames = [df, df1]

res1 = pd.concat(frames)

res1

Q10. How do you merge two Pandas DataFrames on a specific column?

Ans) We can merge two Pandas DataFrames on certain columns using the merge function by simply specifying the certain columns for merge.

Syntax:

DataFrame.merge(right, how=’inner’, on=None, left\_on=None, right\_on=None, left\_index=False, right\_index=False, sort=False, copy=True, indicator=False, validate=None)

Q11. How do you group data in a Pandas DataFrame by a specific column and apply an aggregation function?

Ans) The groupby() function returns a GroupBy object, but essentially describes how the rows of the original data set has been split.

Syntax:

data.groupby('month')['duration'].sum()

Q12. How do you pivot a Pandas DataFrame?

Ans) The pivot() function is used to reshaped a given DataFrame organized by given index / column values. This function does not support data aggregation, multiple values will result in a MultiIndex in the columns.

Syntax:

DataFrame.pivot(self, index=None, columns=None, values=None)

Q13. How do you change the data type of a column in a Pandas DataFrame?

Ans) [DataFrame.astype()](https://www.geeksforgeeks.org/python-pandas-dataframe-astype/) method is used to cast pandas object to a specified dtype. This function also provides the capability to convert any suitable existing column to a categorical type.

Syntax:

import pandas as pd

df = pd.DataFrame({

'A': [1, 2, 3, 4, 5],

'B': ['a', 'b', 'c', 'd', 'e'],

'C': [1.1, '1.0', '1.3', 2, 5]})

df = df.astype(str)

print(df.dtypes)

Q14. How do you sort a Pandas DataFrame by a specific column?

Ans) By using the sort\_values() function, we can sort the values of a specific column.

Syntax:

df.sort\_values("city08")

Q15. How do you create a copy of a Pandas DataFrame?

Ans) By using the copy() function, we can copy a Pandas DataFrame.

Syntax:

df = pd.DataFrame(data)  
newdf = df.copy()  
print(newdf)

Q16. How do you filter rows of a Pandas DataFrame by multiple conditions?

Ans) The loc function in pandas can be used to access groups of rows or columns by label.

Syntax:

import pandas as pd

dataFrame = pd.DataFrame({'Name': [' RACHEL ', ' MONICA ', ' PHOEBE ',

' ROSS ', 'CHANDLER', ' JOEY '],

'Age': [30, 35, 37, 33, 34, 30],

'Salary': [100000, 93000, 88000, 120000, 94000, 95000],

'JOB': ['DESIGNER', 'CHEF', 'MASUS', 'PALENTOLOGY',

'IT', 'ARTIST']})

display(dataFrame.loc[(dataFrame['Salary']>=100000) & (dataFrame['Age']< 40) & (dataFrame['JOB'].str.startswith('D')),

['Name','JOB']])

Q17. How do you calculate the mean of a column in a Pandas DataFrame?

Ans) Using mean() or describe() function, we can calculate the mean of a column.

Syntax:

Df2 = df[“Fee”].mean()

Q18. How do you calculate the standard deviation of a column in a Pandas DataFrame?

Ans) Using std() or describe() function, we can calculate the standard deviation of a column.

Syntax:

df = pd.DataFrame(d)

answer= df.std()

Q19. How do you calculate the correlation between two columns in a Pandas DataFrame?

Ans) By using corr() function, we can be able to calculate the correlation between two columns.

Syntax:

import pandas as pd

data = pd.DataFrame({

"column1": [12, 23, 45, 67],

"column2": [67, 54, 32, 1],

"column3": [34, 23, 56, 23]

}

)

print(data)

print(data['column1'].corr(data['column2']))

print(data['column2'].corr(data['column3']))

print(data['column1'].corr(data['column3']))

Q20. How do you select specific columns in a DataFrame using their labels?

Ans) Passing a list in the brackets lets you select multiple columns at the same time.

Syntax:

df[['alcohol','hue']]

Q21. How do you select specific rows in a DataFrame using their indexes?

Ans) We can select specific rows in a df by using an operator separated by colon(:)

Syntax:

df[15:31]

Q22. How do you sort a DataFrame by a specific column?

Ans) Using sort\_values() function, we can sort values in a DF by a specific column.

Syntax:

df.sort\_values("Ticket")

Q23. How do you create a new column in a DataFrame based on the values of another column?

Ans) We can create a new column by assigning some values to a variable.

Syntax:

import pandas as pd  
  
df = pd.DataFrame(  
 [  
 (1, 'Hello', 158, True, 12.8),  
 (2, 'Hey', 567, False, 74.2),  
 (3, 'Hi', 123, False, 1.1),  
 (4, 'Howdy', 578, True, 45.8),  
 (5, 'Hello', 418, True, 21.1),  
 (6, 'Hi', 98, False, 98.1),  
 ],  
 columns=['colA', 'colB', 'colC', 'colD', 'colE']  
)  
print(df)

Q24. How do you remove duplicates from a DataFrame?

Ans) The drop\_duplicates() method removes duplicate rows. Use the subset parameter if only some specified columns should be considered when looking for duplicates.

Q25. What is the difference between .loc and .iloc in Pandas?

Ans) The main distinction between loc and iloc is: loc is label-based, which means that you have to specify rows and columns based on their row and column labels. iloc is integer position-based, so you have to specify rows and columns by their integer position values (0-based integer position).