**# PandasAssignment**

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

Ans- Following are the syntax to load CSV file in Pandas:

import pandas as pd

df = pd.read\_csv('Path where the CSV file is stored\File name.csv')

print(df)

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

Ans- To check the data type of column we can use “dtype” attribute. Syntax is shown below:

df.dtypes

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

Ans-

record = {

'Name': ['Ankit', 'Amit', 'Aishwarya', 'Priyanka', 'Priya', 'Shaurya' ],

'Stream': ['Math', 'Commerce', 'Science', 'Math', 'Math', 'Science'],

'Percentage': [88, 92, 95, 70, 65, 78] }

# creating a dataframe

df = pd.DataFrame(record)

print("Given Dataframe:\n",df)

res\_df = df[df['Percentage']>80]

print("\nResult Dataframe:\n",res\_df)

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

Ans- We can use rename() function to rename columns and indexes

record = {

'Nme': ['Ankit', 'Amit', 'Aishwarya', 'Priyanka', 'Priya', 'Shaurya' ],

'Stream': ['Math', 'Commerce', 'Science', 'Math', 'Math', 'Science']}

# converting dictionary into dataframe

df = pd.DataFrame(record)

print(df)

# renaming the columns

new = df.rename(columns = {'Nme':'Name'})

print("\n",new)

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

Ans- By using drop() method you can drop columns in pandas dataframe. For that you need to write syntax as below:

df.drop('column\_name', axis=1, inplace=True)

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

Ans- By using unique() method you can find the unique values in column of pandas Dataframe. You need to write syntax as below:

df['col\_name'].unique()

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

Ans- We can use “isnull()” function to find out all field which have missing values. This will return True if field has missing values and false if the field does not have missing values. To get how many missing values are in each column we use sum() along with isnull() which is shown below.

df.isnull().sum()

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

Ans- You can fill missing values with fillna() method.

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

df

# filling missing values with zero

new\_df = df.fillna(0)

new\_df

**Q9. How do you concatenate two Pandas DataFrames?**

Ans- For concatenate we need to pass two dataframes in pd.concat() method is shown below:

data1 = {'Name' : ['Jai', 'Anuj', 'Jai', 'Prince'],

'Age' : [27, 24, 22, 32],

'Address' : ['Nagpur', 'Kanpur', 'Allahabad', 'Kannuaj'],

'Qualification' : ['Msc', 'MA', 'MCA', 'Phd'],

'Salary': [1000, 2000, 3000, 4000]

}

data2 = {'Name' : ['Gaurav', 'Anuj', 'Prince', 'Abhi','Gaurav'],

'Age' : [33, 36, 27, 32, 28],

'Address' : ['Jaunpur', 'Kanpur', 'Allahabad', 'Aligarh', 'Varansi'],

'Qualification' : ['B.tech', 'B.com', 'Msc', 'MA', 'B.tech'],

'Salary': [5000, 6000, 7000, 8000, 9000]

}

# converting dict to df and gives index number

df1 = pd.DataFrame(data1, index=[0,1,2,3])

df2 = pd.DataFrame(data2, index=[4,5,6,7,8])

# Merging the dataframes

df\_concat = [df1, df2]

res= pd.concat(df\_concat)

res

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

Ans-

# creating a dataframe

df1 = pd.DataFrame({'Name':['Raju', 'Rani', 'Geeta', 'Sita', 'Sohit'],

'Marks':[80, 90, 75, 88, 59]})

# creating another dataframe with different data

df2 = pd.DataFrame({'Name':['Raju', 'Divya', 'Geeta', 'Sita'],

'Grade':['A', 'A', 'B', 'A'],

'Rank':[3, 1, 4, 2 ],

'Gender':['Male', 'Female', 'Female', 'Female']})

# display df1

print(df1)

# merging the data on

mer\_df = df1.merge(df2[['Name','Rank','Gender']])

print('\n',mer\_df)

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

Ans-

data = {"Team": ["Red Sox", "Red Sox", "Red Sox", "Red Sox", "Red Sox", "Red Sox", "Yankees", "Yankees", "Yankees", "Yankees", "Yankees", "Yankees"],

"Pos": ["Pitcher", "Pitcher", "Pitcher", "Not Pitcher", "Not Pitcher", "Not Pitcher", "Pitcher", "Pitcher", "Pitcher", "Not Pitcher", "Not Pitcher", "Not Pitcher"],

"Age": [24, 28, 40, 22, 29, 33, 31, 26, 21, 36, 25, 31]}

# create dataframe

df = pd.DataFrame(data)

# using aggregate function

df1 = df.groupby('Team').aggregate({'Age':['mean', 'min', 'max']})

df1

**Q12. How do you pivot a Pandas DataFrame?**

Ans-

data1 = {'Name' : ['Jai', 'Anuj', 'Prince', 'Gaurav','Abhi'],

'Age' : [27, 24, 22, 32, 33],

'Address' : ['Nagpur', 'Kanpur', 'Allahabad', 'Kanpur', 'Allahabad'],

'Qualification' : ['Msc', 'MA', 'MCA', 'Phd', 'B.tech'],

'Salary': [1000, 2000, 3000, 4000, 5000]

}

df=pd.DataFrame(data1)

df1 = df.pivot(index='Name',columns='Address',values='Salary')

df1

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

Ans- You can change the data type of column by using DataFrame.astype() method.

dict = {

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

'B': ['55', '60', '70', '80', '90']}

df=pd.DataFrame(dict)

df1=df.astype(float)

df1

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

Ans- We can use sort\_values() to sort Pandas Dataframe by specific column.

data1 = {'Name' : ['Jai', 'Anuj', 'Prince', 'Gaurav','Abhi'],

'Age' : [27, 24, 22, 32, 33],

'Address' : ['Nagpur', 'Kanpur', 'Allahabad', 'Kanpur', 'Allahabad'],

'Qualification' : ['Msc', 'MA', 'MCA', 'Phd', 'B.tech'],

'Salary': [1000, 2000, 3000, 4000, 5000]

}

df = pd.DataFrame(data1)

print('Before Sort:\n', df)

df1 = df.sort\_values('Name')

print('\nAfter Sort:\n',df1)

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

Ans- By using copy() method you can create a copy of a Pandas DataFrame.

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

Ans-

data = {'Name' : ['Jai', 'Anuj', 'Prince', 'Gaurav','Abhi'],

'Age' : [35, 24, 22, 29, 33],

'Address' : ['Nagpur', 'Kanpur', 'Allahabad', 'Kanpur', 'Allahabad'],

'Qualification' : ['Msc', 'MA', 'MCA', 'Phd', 'B.tech'],

'Salary': [1000, 2000, 3000, 4000, 5000]

}

df = pd.DataFrame(data)

# filter dataframe

df1 = df.loc[(df['Age']>30) & (df['Salary']>1000), ['Name', 'Qualification']]

df1

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

Ans- By using mean() and describe() method to calculate the mean of column in Pandas Dataframe.

technologies = {

'Courses':["Spark","PySpark","Python","pandas"],

'Fee' :[20000,25000,22000,30000],

'Duration':['30days','40days','35days','50days'],

'Discount':[1000,2300,1200,2000]

}

df = pd.DataFrame(technologies, index=['r1','r2','r3','r4'])

# calculate mean

df1 = df['Fee'].mean()

df1

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

Ans- By using std() method you can calculate standard deviation of column in Pandas Dataframe is shown as below:

d = {

'Name':['Alisa','Bobby','Cathrine','Madonna','Rocky','Sebastian','Jaqluine',

'Rahul','David','Andrew','Ajay','Teresa'],

'Score1':[62,47,55,74,31,77,85,63,42,32,71,57],

'Score2':[89,87,67,55,47,72,76,79,44,92,99,69],

'Score3':[56,86,77,45,73,62,74,89,71,67,97,68]}

df = pd.DataFrame(d)

# Calculate the standard deviation

answer= df.std()

print(answer)

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

Ans- By using corr() function we can get the correlation between two columns in the dataframe is shown as below:

data = {

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

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

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

# display dataframe

df = pd.DataFrame(data)

print(df)

# correlation between column 1 and column2

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

# correlation between column 2 and column3

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

# correlation between column 1 and column3

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

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

Ans-

employees = [('Stuti', 28, 'Varanasi', 20000),

('Saumya', 32, 'Delhi', 25000),

('Aaditya', 25, 'Mumbai', 40000),

('Saumya', 32, 'Delhi', 35000),

('Saumya', 32, 'Delhi', 30000),

('Saumya', 32, 'Mumbai', 20000),

('Aaditya', 40, 'Dehradun', 24000),

('Seema', 32, 'Delhi', 70000)]

# Create a DataFrame from list

df = pd.DataFrame(employees, columns=['Name','Age','City','Salary'])

print('Given Dataframe:\n',df)

# display name and age column

res = df[['Name','Age']]

print('\nDisplay column:\n',res)

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

Ans-

employees = [('Stuti', 28, 'Varanasi', 20000),

('Saumya', 32, 'Delhi', 25000),

('Aaditya', 25, 'Mumbai', 40000),

('Saumya', 32, 'Delhi', 35000),

('Saumya', 32, 'Delhi', 30000),

('Saumya', 32, 'Mumbai', 20000),

('Aaditya', 40, 'Dehradun', 24000),

('Seema', 32, 'Delhi', 70000)]

# Create a DataFrame from list

df = pd.DataFrame(employees, columns=['Name','Age','City','Salary'])

# print('Given Dataframe:\n',df)

# set name column as index

df.set\_index('Name',inplace=True)

# Select rows using index

res = df.loc['Seema']

print(res)

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

Ans- We can use sort\_values() to sort Pandas Dataframe by specific column.

data1 = {'Name' : ['Jai', 'Anuj', 'Prince', 'Gaurav','Abhi'],

'Age' : [27, 24, 22, 32, 33],

'Address' : ['Nagpur', 'Kanpur', 'Allahabad', 'Kanpur', 'Allahabad'],

'Qualification' : ['Msc', 'MA', 'MCA', 'Phd', 'B.tech'],

'Salary': [1000, 2000, 3000, 4000, 5000]

}

df = pd.DataFrame(data1)

print('Before Sort:\n', df)

df1 = df.sort\_values('Name')

print('\nAfter Sort:\n',df1)

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

Ans-

programs = {

'Date':['10/2/2011', '11/2/2011', '12/2/2011', '13/2/2011'],

'Event':['Music', 'Poetry', 'Theatre', 'Comedy'],

'Cost':[10000, 5000, 15000, 2000]}

# crate dataframe

df = pd.DataFrame(programs)

print(df)

# Calculate 10% discounted price in new column

df['Discounted Price'] = df['Cost'] - (0.1 \* df['Cost'])

print('\n',df)

**Q24. How do you remove duplicates from a DataFrame?**

Ans- The drop\_duplicates() method removes duplicate rows.

students = {

'Name' : ['Jai', 'Anuj', 'Jai', 'Prince', 'Gaurav', 'Anuj'],

'Age' : [27, 24, 27, 32, 33, 24],

'Address' : ['Nagpur', 'Kanpur', 'Nagpur', 'Kannuaj', 'Jaunpur', 'Kanpur']}

# Create dataframe

df = pd.DataFrame(students)

# remove duplicates

df.drop\_duplicates(inplace=True)

df

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

Ans- For accessing rows dataframe.iloc() consider the system generated index. And dataframe.loc() consider the user index.