# PandasAssignment

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

Ans. A simple way to store big data sets is to use CSV files (comma separated files).CSV files contain plain text and is a well known format that can be read by everyone including PandasIn our examples we will be using a CSV file called 'data.csv'.

import pandas as pd

df = pd.read\_csv('data.csv')

print(df.to\_string())

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.

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

Ans. Selecting rows based on a particular column value using '>', '=', '=', '<=', '!=' operator.

# importing pandas

import pandas as pd

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

'Age': [21, 19, 20, 18, 17, 21],

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

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

# create a dataframe

dataframe = pd.DataFrame(record, columns = ['Name', 'Age', 'Stream', 'Percentage'])

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

# selecting rows based on condition

rslt\_df = dataframe[dataframe['Percentage'] > 80]

print('\nResult dataframe :\n', rslt\_df)

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

Ans. One way of renaming the columns in a Pandas Dataframe is by using the rename() function. This method is quite useful when we need to rename some selected columns because we need to specify information only for the columns which are to be renamed.

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

Ans. The **drop()** method removes the specified row or column.By specifying the column axis (axis='columns'), the drop() method removes the specified column.By specifying the row axis (axis='index'), the drop() method removes the specified row.

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

Ans. pandas.DataFrame().unique() method is used when we deal with a single column of a DataFrame and returns all unique elements of a column. The method returns a DataFrame containing the unique elements of a column, along with their corresponding index labels.

Syntax:

Series.unique(self)

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

Ans. We will use Pandas’s isna() function to find if an element in Pandas dataframe is missing value or not and then use the results to get counts of missing values in the dataframe.

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.

Replace NULL values with the number 222111:

import pandas as pd

df = pd.read\_csv('data.csv')

new df = df.fillna(222111)

print(newdf.to\_string())

#Note that we use the to\_string() method to return the entire DataFrame.

Q9. How do you concatenate two Pandas DataFrames?

Ans. 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.

import pandas as pd

# First DataFrame

df1 = pd.DataFrame({'id': ['A01', 'A02', 'A03', 'A04'],

'Name': ['ABC', 'PQR', 'DEF', 'GHI']})

# Second DataFrame

df2 = pd.DataFrame({'id': ['B05', 'B06', 'B07', 'B08'],

'Name': ['XYZ', 'TUV', 'MNO', 'JKL']})

frames = [df1, df2]

result = pd.concat(frames)

display(result)

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.

# importing modules

import pandas as pd

# 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

display(df1)

# display df2

display(df2)

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

Ans. [DataFrame.groupby()](https://sparkbyexamples.com/pandas/pandas-groupby-explained-with-examples/) function is used to collect the identical data into groups and perform aggregate functions on the grouped data. This function returns DataFrame GroupBy object where several aggregate functions are defined.

By default, it calculates specified aggregation functions on all numeric columns.

# Using groupby() and aggregate()

result = df.groupby('Courses').aggregate('sum')

print(result)

# Outputs

Fee Discount

Courses

Hadoop 26000 1200

PySpark 49000 4300

Python 22000 2500

Spark 55000 3000

Q12. How do you pivot a Pandas DataFrame?

Ans. The pivot() function is used to reshape 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.

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

Ans. Change column type into string object using DataFrame.astype()

[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.

# importing pandas as pd

import pandas as pd

# sample dataframe

df = pd.DataFrame({

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

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

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

# converting all columns to string type

df = df.astype(str)

print(df.dtypes)

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

Ans. To sort the rows of a DataFrame by a column, use pandas.DataFrame.sort\_values() method with the argument by=column\_name. The sort\_values() method does not modify the original DataFrame, but returns the sorted DataFrame.

import pandas as pd

data = {'name': ['prasuna', 'vamsi', 'puji', 'sneha'],

'physics': [68, 74, 77, 78],

'chemistry': [84, 56, 73, 69],

'algebra': [78, 88, 82, 87]}

#create dataframe

df\_marks = pd.DataFrame(data)

#sort dataframe

sorted\_df = df\_marks.sort\_values(by='algebra')

print(sorted\_df)

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

Ans. The copy() method returns a copy of the DataFrame.By default, the copy is a "deep copy" meaning that any changes made in the original DataFrame will NOT be reflected in the copy.

import pandas as pd

data = {"name": ["Sally", "Mary", "John"],

"qualified": [True, False, False]}

df = pd.DataFrame(data)

#Make a copy:

newdf = df.copy()

print(newdf)

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

Ans.By using df[], loc[], query(), eval() and numpy.where() we can filter Pandas DataFrame by multiple conditions.The process of applying multiple filter conditions in Pandas DataFrame is one of the most frequently performed tasks while manipulating data.

# Example 1: Use DataFrame.loc[] to filter by multiple conditions

df2 = df.loc[(df['Fee']>=24000) &

(df['Discount']< 2000) &

(df['Courses'].str.startswith('P')),

['Courses','Duration']]

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

Ans. To get column average or mean from pandas DataFrame use either mean() and describe() method. The [DataFrame.mean()](https://sparkbyexamples.com/pandas/pandas-dataframe-mean-examples/) method is used to return the mean of the values for the requested axis.

# Using DataFrame.mean() to get entire column mean

df2 = df.mean()

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

Ans. Standard deviation is calculated using the function .std(). However, the Pandas library creates the Dataframe object and then the function .std() is applied on that Dataframe.

import pandas as pd

import numpy as np

#Create a DataFrame

d = { 'Name':['valli','prasuna','sneha','gayathri','Rocky','vamsi','abhi',

'Rahul','David','vijay','Ajay','bhargavi'],

'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)

answer= df.std()

print("The standard deviations of the 3 columns are:")

print (answer)

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

Ans. By using [corr()](https://www.geeksforgeeks.org/python-pandas-dataframe-corr/) function we can get the correlation between two columns in the dataframe.

# import pandas module

import pandas as pd

# create dataframe with 3 columns

data = pd.DataFrame({

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

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

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

}

)

# display dataframe

print(data)

# correlation between column 1 and column2

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

# correlation between column 2 and column3

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

# correlation between column 1 and column3

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

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

Ans. To access specific columns of a DataFrame with their columns labels, directly use DataFrame[~] or use the [DataFrame.loc](https://www.skytowner.com/explore/pandas_dataframe_loc_property) property.

df = pd.DataFrame({"A":[3,4],"B":[5,6]}, index=["a","b"])

df

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

Ans. Indexing in Pandas means selecting rows and columns of data from a [Dataframe](https://www.geeksforgeeks.org/python-pandas-dataframe/).

# import pandas

import pandas as pd

# List of Tuples

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 object from list

df = pd.DataFrame(employees,

columns =['Name', 'Age',

'City', 'Salary'])

# Show the dataframe

df

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

Ans. To sort the rows of a DataFrame by a column, use pandas.DataFrame.sort\_values() method with the argument by=column\_name. The sort\_values() method does not modify the original DataFrame, but returns the sorted DataFrame.

import pandas as pd

data = {'name': ['Somu', 'Kiku', 'Amol', 'Lini'],

'physics': [68, 74, 77, 78],

'chemistry': [84, 56, 73, 69],

'algebra': [78, 88, 82, 87]}

#create dataframe

df\_marks = pd.DataFrame(data)

#sort dataframe

sorted\_df = df\_marks.sort\_values(by='algebra')

print(sorted\_df)

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

Ans.# importing pandas as pd

import pandas as pd

# Creating the DataFrame

df = pd.DataFrame({'Date':['10/2/2011', '11/2/2011', '12/2/2011', '13/2/2011'],

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

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

# Print the data frame

print(df)

Q24. How do you remove duplicates from a DataFrame?

Ans. Pandas drop\_duplicates() method helps in removing duplicates from the [Pandas Dataframe](https://www.geeksforgeeks.org/python-pandas-dataframe/) In [Python](https://www.geeksforgeeks.org/python-programming-language/).

import pandas as pd

data = {"A": ["A", "B", "B", "C", "A"],

"B": [50, 40, 40, 30, 50],

"C": [True, False, False, False, True]

}

df = pd.DataFrame(data)

display(df.drop\_duplicates())

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

Ans. The main difference between pandas .loc vs .iloc is that loc gets DataFrame rows & columns by labels/names and iloc[] gets by integer Index/position. For loc[], if the label is not present it gives a key error. For iloc[], if the position is not present it gives an index error.