**PandasAssignment**

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

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?

df['DataFrame Column'].dtypes

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

df.loc[df['column\_name'] == some\_value]

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

df = df.rename(columns={'oldName1': 'newName1', 'oldName2': 'newName2'})

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

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

where 1 is the *axis* number (0 for rows and 1 for columns.)

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

We can find unique values by unique function in two formats:

* series.unique() : In this we have to add the unique function after the series(column) in which we want to find the unique values.
* pd.unique() : In this we have to pass the series as a parameter to find the unique values.

We have used both functions for better understanding. print(df.name.unique()) print(pd.unique(df['year']))

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

df.isnull**()**.sum**()**

It gives you pandas series of column names along with the sum of missing values in each column.

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

DataFrame.fillna(value=None, method=None, axis=None, inplace=False, limit=None, downcast=None, \*\*kwargs)

Q9. How do you concatenate two Pandas DataFrames?

# concatenating df1 and df2 along rows

vertical\_concat **=** pd.concat([df1, df2], axis**=**0)

# concatenating df3 and df4 along columns

horizontal\_concat **=** pd.concat([df3, df4], axis**=**1)

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

df1.merge(df2[['Name', 'Grade', 'Rank']])

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

# applying groupby() function to

# group the data on team value.

gk **=** df.groupby('Team')

Q12. How do you pivot a Pandas DataFrame?

# values can be an object or a list

df.pivot('A', 'B', 'C')

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

DataFrame.astype() casts this DataFrame to a specified datatype. Following is the syntax of astype() method.

astype(dtype, copy=True, errors='raise', \*\*kwargs)

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

# Sort the rows of dataframe by 'Name' column

rslt\_df **=** details.sort\_values(by **=** 'Name')

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

#Make a copy:  
newdf = df.copy()

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

As an input to label you can give a single label or it’s index or a list of array of labels.

Enter all the conditions and with & as a logical operator between them.

df.loc[(df['Salary\_in\_1000']>=100) & (df['Age']< 60) &(df['FT\_Team'].str.startswith('S')),['Name','FT\_Team']]

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

df["weight"].mean()

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

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.

df = pd.DataFrame(d)

answer= df.std()

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

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

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

The following code shows how to create a pandas DataFrame and use .loc to select the column with an index label of ‘rebounds’:

df = pd.DataFrame({'team': ['A', 'A', 'A', 'B', 'B', 'B'],

'points': [11, 7, 8, 10, 13, 13],

'assists': [5, 7, 7, 9, 12, 9],

'rebounds': [11, 8, 10, 6, 6, 5]})

#view DataFrame

df

team points assists rebounds

0 A 11 5 11

1 A 7 7 8

2 A 8 7 10

3 B 10 9 6

4 B 13 12 6

5 B 13 9 5

#select column with index label 'rebounds'

df.loc[:, 'rebounds']

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

#select column with index position 3

df.iloc[:, 3]

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

We can sort pandas dataframe based on the values of a single column by specifying the column name wwe want to sort as input argument to sort\_values(). For example, we can sort by the values of “lifeExp” column in the gapminder data like:

sort\_by\_life **=** gapminder.sort\_values('lifeExp')

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

We can use DataFrame.apply() function to achieve this task. Also We can achieve the same result by directly performing the required operation on the desired column element-wise.

Q24. How do you remove duplicates from a DataFrame?

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

# dropping ALL duplicate values

data.drop\_duplicates(subset="First Name",

                     keep=False, inplace=True)

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

The[loc() function](https://www.geeksforgeeks.org/python-pandas-dataframe-loc/) is label based data selecting method which means that we have to pass the name of the row or column which we want to select. This method includes the last element of the range passed in it, unlike iloc(). loc() can accept the boolean data unlike iloc().

The[iloc() function](https://www.geeksforgeeks.org/python-extracting-rows-using-pandas-iloc/) is an indexed-based selecting method which means that we have to pass an integer index in the method to select a specific row/column. This method does not include the last element of the range passed in it unlike loc(). iloc() does not accept the boolean data unlike loc().