**Panda Assignment**

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

Ans. To upload a CSV file into a Pandas DataFrame first, we have to import the Pandas and then we will read the file through that.

import pandas as pd  
  
df = pd.read\_csv('data.csv')  
  
print(df)

**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. If any column has mixed data types are stored then the data type of the entire column is indicated as object dtype.

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

Ans. There are several methods through which we can apply the conditions on the columns/rows:

Method 1: Create a new df and then write the condition in square brackets with existing df and then print the new df.

Ex:

df

new\_df = df[df[column name]>5] // df[condition]

new\_df

Method 2: Using loc

new\_df = df.loc[ df[column name]>5]

New\_df

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

Ans. Using rename() function we can rename any specific column providing it’s information and to rename all the columns in single shot we can use the list method i.e using df.column = [column names] method

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

Ans. drop.([column name], axis=1)

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

Ans. df.column\_name.unique()

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

Ans. df.isnull().sum().sum()

df.isnull() // this will give us the Boolean values like true and false for the places where the null value is present and where not.

df.isnull().sum() // this will give the sum of all the null values present in each column.

df.isnull().sum().sum() this will give you the total sum or total number of null values in the complete dataset.

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

Ans. To fill the missing values in a pandas Dataframe with a specific value we can use

df.fillna(value = any number)

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

Ans. Concatenation can be done using concat() with pd.

* Concatenate two rows

New\_df = pd.concate ([df, new\_rows], axis = 0, ignore\_index = true)

* Concatenate two columns

New\_df = pd.concate([df, new\_columns], axis = 1)

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

Ans. To merge two data frames on a specific column

df1, df2 are two dataframes then we can merge them using merge function with “by” parameter.

>> merge(df1, df2, by = “column name”)

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

Ans. To do the grouping you can use the groupby() function and followed by that you can use the aggregate functions.

The syntax - df.groupby().aggregate()

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

Ans. Pivot allows you to reshape and transform the dataframe and it takes three arguments that are index, columns, values if needed.

Where index takes the value of the rows you want to have in df and column takes the column names and value can take the parameter of whatever values you want to have in your df.

Ex: df.pivot(index= ‘ ’, columns =’ ’ , values =’ ’ ) //where values is optional

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

Ans. We can use the astype() method for changing the data type of a column in a Panda data frame.

Ex: df[“column name”]= df[“column n”].astype(“data type”)

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

Ans: We can sort a Pandas data frame by specific column using sort\_values() function.

Ex: df.sort\_values(“Column name”)

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

.Ans A copy of a dataframe can be created with .copy() in two ways shallow copy and deep copy.

1. Shallow copy : in this the copy and the original dataframe has different different memory location.

Lst1=[ 1,3,4,5]

Lst2 =Lst1.copy()

Or

Lst2 = copy.copy(Lst1)

1. Deep copy: When we try to copy the nested list df then they both are refering to the same object location.
2. Lst1=[[ 1,3,4,5], [6,7,8,9]]

Lst[1][0] = 100

Then in both the copy and the original df, the value will be changed to 100.

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

Ans. We can use the loc and iloc methods for filtering the rows by multiple conditions.

For ex:

* first we will write the conditions using logical operators like & or etc. like- df[condition one] == true & df[condition 2] == false
* After that I can use the loc function outside it to get the exact location of the values satisfying the conditions like –
* df.loc [(df[condition one] == true & df[condition 2)] == false]
* For using iloc we will have to get the indexes first in the form of and to get the indexes we will use “np.where” like-

np.where(df[condition one] == true & df[condition 2] == false)

df.iloc[np.where(df[condition one] == true & df[condition 2] == false)

]

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

Ans: To calculate the mean of the column in the pandas dataframe we can use the mean() function. The syntax for calculating the mean of the specific column-

Syntax- df[‘column name’].mean()

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

Ans: To calculate the standard deviation of a column in a pandas dataframe we can use .std() function, like df[‘column name’].std()

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

Ans: To calculate the correlation between two columns in a pandas df we can first assign the variable names to the columns and then can use the corr() function for finding the correlation between them**.**

The syntax for it is - df[‘col1’].corr([‘col2’])

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

Ans: We can select the specific columns in a data frame using their labels by simply writing the column name in the square brackets. For ex: df [‘column name’].

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

Ans: To select the rows, we can use the slicing method, and the syntax for is –

df. loc[start: stop: step]; where the start is the name of the first-row label to take, stop is the name of the last-row label to take, and step is the number of indices to advance after each extraction.

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

Ans: we can use the .sort\_values() method; ex: df.sort\_values(“Column name”)

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

Ans: To create a new column using the existing column we can use the square brackets with the column name written inside them along the df.

Ex: df[“new\_column”] = df[“existing\_column”]+ df[“any\_other\_column”]

**Q24. How do you remove duplicates from a data frame?**

Ans: To remove the duplicates from a data frame we can use the drop\_duplicates() Method.

Ex: df

New\_df= df.drop\_duplicates()

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

Ans: The major difference between the .loc and .iloc is majorly of indexing i.e. loc is typically used for label indexing and can access multiple columns, while iloc is used for integer indexing.