pandas_solution

August 20, 2023

[ ]:

[ ]:
```
Q1. How do you load csv file into pandas Dataframe?
Ans-1. using the read.csv() function from pandas package, we can load csv file.
```

[ ]:
```
Q2. how do you check datatype of a column in a pandasDataframe?
Ans-2. we can use "dtype" attribute.
```

[ ]:
```
Q3. How do you select rows from a pandas Dataframe based on a condition?
Ans-3. we can use df.[row name]
     for ex-
         result_df = Dataframe[dataframe[percentage]>80]
```

[ ]:
```
Q4. how do you rename column in a pandas Dataframe?
Ans-4. rankings_pd.rename(columns = {'test':'TEST', 'odi':'ODI','t20':'T20'},␣
 ↪inplace = True)
```

[ ]:
```
Q5. How do you drop column in a pandas Dataframe?
Ans-5. df.drop('new column',axis=inplace=True)
```

[ ]:
```
Q6. How do you find the unique values in a column?
Ans-6.by using unique method().
for ex-
   df['species'].unique()
```

[ ]:
```
Q7. how do you fill the missing values in a pandas Dataframe with a specific␣
 ↪values?
Ans-7.By using fillna()method, we can fill missing values.
for ex-
df.fillna(method='ffill', inplace=True)
```

[ ]:
```
Q8. How do you concatenate two pandas Dataframe?
Ans-8.we can pass two Dataframe to pd.concate()method in the form of list and␣
 ↪mention in which axis you want to concate.
for ex-
vertical_concat = pd.concat([df1, df2], axis=0
```

[ ]:
```
Q9. how do you find the number of misssing values in each column of pandas␣
 ↪Dataframe?
ans-9 we can pass df.isna()and df.isnull() and we can find the missing values.
for ex-
print(df.isna()),print(df.isnull())
```

[ ]:
```
Q10. how do yu merge two Dataframe in Pandas?
Ans-10.By using the merge() function and set on parameter as the column name.
for ex-
(df2[['Name', 'Grade', 'Rank']])
```

[ ]:
```
Q11. How do you group data in a Pandas DataFrame by a specific column and apply␣
 ↪an aggregation function?
Ans-11.
To do grouping use DataFrame.groupby() function. This function returns the␣
 ↪DataFrameGroupBy object and use aggregate()function
to calculate the sum.
for ex -
result = df[['Fee','Discount']].aggregte('sum')
```

[ ]:
```
Q12.How do you pivot a Pandas DataFrame?
Ans-12.you can pivot table by adding .pivot() to the end of your pivot table␣
 ↪code will create a plot of the data.
for ex-
pivot = np.round(pd.pivot_table(data, values='price',
                                index='num-of-doors',
                                columns='fuel-type',
                                aggfunc=np.mean),2)
pivot
```

[ ]:
```
Q13. How do you change the data type of a column in a Pandas DataFrame?
Ans-13.
BY using DataFrame.astype() method.it 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.
for ex-
df = df.astype(str)
```

[ ]:
```
Q14.How do you sort a Pandas DataFrame by a specific column?
Ans-14.To sort the DataFrame based on the values in a single column, you'll use␣
 ↪. sort_values() .
for ex-
 df.sort_values("city08")
```

[ ]:
```
Q15.How do you create a copy of a Pandas DataFrame?
```

```
Ans-15The copy method is used to make a copy of the given DataFrame. There are␣
 ↪two ways a DataFrame is copied:

 1. Deep copy: It creates a new DataFrame with a copy of the data and indices␣
 ↪of the given DataFrame.
Changes to the copy's data or indices will not be reflected in the original␣
 ↪DataFrame.
for ex-
     df_deep_copy =df.copy(deep=True)
 2. Shallow copy: It creat the data and index are copied).
    Any modifications to the original's data will be mirrored in the copy (and␣
 ↪vice versa).
for ex-
     df_shallow_copy =df.copy(deep=false)
```

[ ]:
```
Q16.How do you filter rows of a Pandas DataFrame by multiple conditions?
Ans-16. using loc to filter with multiple condition.
for ex-
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-17.mean() function returns the mean of the values for the requested axis.
If the method is applied on a pandas series object, then the method returns a␣
 ↪scalar value which is the mean value of
all the observations in the Pandas Dataframe.
for ex-
df.mean(axis = 0)
```

[ ]:
```
Q18.How do you calculate the standard deviation of a column in a Pandas␣
 ↪DataFrame?
Ans-18.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 .
for ex-
print(my_df['Age'].std())
```

[ ]:
```
Q19.How do you calculate the correlation between two columns in a Pandas␣
 ↪DataFrame?
Ans-19.Print the input DataFrame, df. Initialize two variables, col1 and col2,␣
 ↪and assign them the columns that you want to find the correlation of.
Find the correlation between col1 and col2 by using df[col1]. corr(df[col2])␣
 ↪and save the correlation value in a variable,
corr.
for ex-
```

```
corr = df[col1].corr(df[col2])
```

[ ]:
```
Q20.How do you select specific columns in a DataFrame using their labels?
Ans-20. To access specific columns of a DataFrame with their columns labels,␣
 ↪directly use DataFrame[~] or use the
DataFrame. loc property.
for ex-
df.loc[:,"A"]
```

[ ]:
```
Q21.How do you select specific rows in a DataFrame using their indexes?
Ans-21.You can select a single row from pandas DataFrame by integer index using␣
 ↪df. iloc[n] . Replace n with a position you
wanted to select.
for ex-
df2 = df.loc['r2']
```

[ ]:
```
Q22.How do you sort a DataFrame by a specific column?
Ans-22. By using sort_values()method.
for ex-
rslt_df = details.sort_values(by = 'Name')
```

[ ]:
```
Q23.How do you create a new column in a DataFrame based on the values of␣
 ↪another column?
Ans-23.You can add/append a new column to the DataFrame based on the values of␣
 ↪another
column using df. assign() , df. apply() , and, np. where() functions and return␣
 ↪a new Dataframe after adding a new
column.
for ex-
df['Discounted_Price'] = df.apply(lambda row: row.Cost -
                                  (row.Cost * 0.1), axis = 1)
```

[ ]:
```
Q24.How do you remove duplicates from a DataFrame?
Ans-24.Pandas DataFrame drop_duplicates() Method
The drop_duplicates() method removes duplicate rows. Use the subset parameter␣
 ↪if only some specified columns
should be considered when looking for duplicates.
for ex-
df = pd.DataFrame(data)
newdf = df.drop_duplicates()
```

[ ]:
```
Q25. What is the difference between .loc and .iloc in Pandas?
Ans-25
Python loc() function
```

```
The loc() function 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(). Many operations␣
 ↪can be performed using the loc() method
like
# selecting cars with brand 'Maruti' and Mileage > 25
display(data.loc[(data.Brand == 'Maruti') & (data.Mileage > 25)])
Python iloc() function
The iloc() function 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(). Operations performed using iloc()
are:
        # selecting 0th, 2nd, 4th, and 7th index rows
display(data.iloc[[0, 2, 4, 7]])
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: