Question 1

Correct

Question text

Which of the following expressions will get and display all rows in the dataframe where WindDirDegrees is greater than or equal to 180 and Mean TemperatureC is less than 14?

Select one:



london[(london['WindDirDegrees'] >= 180)

      & (london['Mean TemperatureC'] < 14)]

The expression will display all rows in the dataframe where `WindDirDegrees` is greater than or equal to 180 and `Mean TemperatureC` is less than 14



london[(london['WindDirDegrees'] == 180)

      & (london['Mean TemperatureC'] > 14)]



london[(london['WindDirDegrees'] <= 180)

      & (london['Mean TemperatureC'] < 14)]



london[(london['WindDirDegrees'] > 180)  
& (london['Mean TemperatureC'] <= 14)]

Feedback

Your answer is correct.

Question 2

Correct

Question text

How would you group a dataframe with three columns named 'City', 'Country' and Continent', by continent and country?

Select one:



df.groupby('Continent','Country')



df.groupBy(['Continent','Country'])



df.groupby(['Continent','Country'])

The method you need is groupby() and the two columns you are grouping on need to be added as a list.



df.grouped(['Continent','Country'])

Feedback

Your answer is correct.

Question 3

Correct

Question text

**Given the following dataframe:**

|  | **Route** | **Region** | **Hemisphere** | **Volume** |
| --- | --- | --- | --- | --- |
| **0** | Sea | Europe | North | 10 |
| **1** | Sea | Africa | South | 15 |
| **2** | Sea | Asia | North | 5 |
| **3** | Sea | Australia | South | 20 |
| **4** | Air | Europe | North | 10 |
| **5** | Air | Africa | South | 10 |
| **6** | Air | Asia | North | 5 |
| **7** | Land | Europe | North | 20 |
| **8** | Land | Africa | South | 30 |

**which expression will generate the following report?**

|  | **Hemisphere** | **Region** | **Volume** |
| --- | --- | --- | --- |
| **0** | North | Europe | 40 |
| **1** | North | Asia | 10 |
| **2** | South | Africa | 55 |
| **3** | South | Australia | 20 |

You are advised to turn your device to landscape orientation or use a desktop before you attempt this question as the table is large.

Select one:



df.groupby(['Hemisphere','Region']).aggregate(sum).reset\_index()

You need to group by the 'Hemisphere' and 'Region' elements, find the sum, and then reset the index to produce the flat table.



df.groupby(['Hemisphere','Region']).aggregate(sum)



df.groupby(['Region','Hemisphere']).aggregate(sum).reset\_index()



df[['Hemisphere','Region']].aggregate(sum).reset\_index()

Feedback

Your answer is correct.

Question 4

Correct

Question text

If there are anomalous values in a column the data type may not be recognised correctly when the file is read. If the constant LOCATION holds the name of the csv file to be read, which of the following would correctly set the  type of column 'Product Name' to type string?

Select one:



df = read\_csv(LOCATION)

df = df['Product Code'].coalesce(str)



df = read\_csv(LOCATION, dtype={'Product Code':string })



df = read\_csv(LOCATION)

df = df['Product Code'].coalesce(string)



df = read\_csv(LOCATION, dtype={'Product Code':str})

Feedback

Your answer is correct.

Correct answer:

df = read\_csv(LOCATION, dtype={'Product Code':str})

Setting the data type to str is the correct strategy using dtype

Question 5

Correct

Question text

Data from a dataframe df may be saved to a csv file named data\_example.csv using which one of the following statements?

Select one:



df.to\_csv('data\_example.csv', index=false)



df.to\_csv('data\_example.csv', index:False)



df.to\_csv\_file('data\_example.csv', index=False)



df.to\_csv('data\_example.csv', index=False)

Feedback

Your answer is correct.

Correct answer:

df.to\_csv('data\_example.csv', index=False)

This is the correct method to save the data as a csv file and the index column will not be included.