Question 1

Correct

Question text

The computer will report an error when executing the following code. Why?

headings = ['Population (2013)', 'Country']

table = [

   ['UK', 64.1, 'Europe'],

   ['Tuvalu', 9876, 'Pacific'],

   ['India', 1252.1, 'Asia']

]

DataFrame(data=table, columns=headings)

Select one:



The dataframe is not stored in a variable.



The headings are in the wrong order.



The population values are inconsistent.



There are only two headings, but three values in each row.

This is the structural error that the computer will detect and report: the headings imply the table has two columns, but the row data implies there are three columns. All other errors listed relate to the meaning of the data or the purpose of the code, which are beyond the computer's understanding.

Feedback

Your answer is correct.

Question 2

Correct

Question text

If you wish to convert miles to kilometers (1 mile is roughly 1.61 kilometers), which function should you program ?

Select one:



def miToKm (miles):

     return miles / 1.61



def miToKm (miles)

    result = miles \* 1.61

    return result



def convert (value):

    return value / 1.61



def miToKm (miles):

    return miles \* 1.61

This is the best of all options: the names are self-explanatory, the function is succinct and is correct.

Feedback

Your answer is correct.

Question 3

Correct

Question text

All but one of these functions do the same action. Which is not equivalent to the other three?

Select one:



def f4 (number):

    if number < 0:

        return -1

    if number > 0:

        return 1

    else:

        return 0



def f2 (number):

    if number < 0:

        return -1

    elif number == 0:

        return 0

    else:

        return 1



def f3 (number):

    if number < 0:

        return -1

    else:

        return 1

    if number == 0:

        return 0

The first if returns -1 for a negative number and 1 for any other number (including zero), so the second if statement is never executed. The other functions return 0 if the number is zero, but this returns 1.



def f1 (number):

    if number < 0:

        return -1

    if number > 0:

        return 1

    return 0

Feedback

Your answer is correct.

Question 4

Correct

Question text

A dataframe gdp has 2 columns, 'Country' and 'GDP (US$)'.

Suppose you have written two functions to convert the GDP into Euros and round to the nearest whole number of thousands. The functions are called usdToEuro() and roundToThousands().

Which of the following lines of code would express the GDP of each country correctly in thousands of Euros to the nearest whole number?

Select one:



roundToThousands(usdToEuro(gdp['GDP(US$)']))



gdp['GDP(US$)'].apply(roundToThousands).apply(usdToEuro)



roundToThousands(gdp['GDP(US$)']).apply(usdToEuro)



gdp['GDP(US$)'].apply(usdToEuro).apply(roundToThousands)

Feedback

Your answer is correct.

Correct answer:

gdp['GDP(US$)'].apply(usdToEuro).apply(roundToThousands)

This is correct as the amount in US$ must be converted to Euros before it is rounded to the nearest whole number of thousands.

Question 5

Correct

Question text

A dataframe gdp has 4 columns: 'Country', 'GDP (US$)', 'GDP (£m)' and 'Euros (1000s)'. Assume functions roundToThousands() and roundToMillions() have been written.

Which of the following would replace gdp with a new dataframe consisting of just 2 columns: 'Country' and 'Euros (1000000s)'?

Select one:



gdp['Country','Euros (1000000s)'] = gdp['Euros (1000s)'].apply(roundToThousands)



gdp['Euros (1000000s)'] = gdp['Euros (1000s)'].apply(roundToMillions)  
gdp = gdp[['Country', 'Euros (1000000s)']]



gdp['Euros (1000000s)'] = gdp['Euros (1000s)'].apply(roundToThousands)  
gdp = gdp['Country', 'Euros (1000000s)']



gdp['Euros (1000000s)'] = gdp['Euros (1000s)'].apply(roundToThousands)  
gdp = gdp[['Country', 'Euros (1000000s)']]

Feedback

Your answer is correct.

Correct answer:

gdp['Euros (1000000s)'] = gdp['Euros (1000s)'].apply(roundToThousands)  
gdp = gdp[['Country', 'Euros (1000000s)']]

This rounds by dividing thousands by 1000.  Doing this to a value which is already in thousands will result in a value in millions, as the original value will have been divided by 1000, twice.  This also drops the unwanted columns.

Question 1

Correct

Question text

Enter the missing words to complete these statements about the Spearman rank correlation coefficient.

The gap in each statement only needs one word to complete it.

a) A value of 1 for the Spearman rank correlation coefficient means that if I rank (sort) the data from smallest to largest value in one indicator, data for the other indicator will be in Answer 1 order.

b) A value of -1 for the Spearman rank correlation coefficient means that if I rank (sort) the data from smallest to largest value in one indicator, data for the other indicator will be in Answer 2 order.

Feedback

Your answer is correct.

Correct answer:

a) A value of 1 for the Spearman rank correlation coefficient means that if I rank (sort) the data from smallest to largest value in one indicator, data for the other indicator will be in **ascending** order.

) A value of -1 for the Spearman rank correlation coefficient means that if I rank (sort) the data from smallest to largest value in one indicator, data for the other indicator will be in **descending** order.

Question 2

Correct

Question text

Which of the following statements are true about constants in Python?

Select one:



If you define a column heading using a constant then that will be printed out when the dataframe is displayed.



Using constants means the code is always consistent throughout.



A variable spelled with capital letters and underscores cannot have its value changed once it is assigned a value.



Use of constants should be avoided as it's clearer to use the value instead.

Feedback

Your answer is correct.

Correct answer:

Using constants means the code is always consistent throughout.

You can be sure if you use a constant that everywhere it's used the same value is being used. This helps to avoid forgetting to change a value in one of several places.

It's up to the programmer not to change the value of a constant as there's no way of enforcing this in Python. Using capital letters for the name acts as a reminder though.

Defining a column name using constants has no effect on what will be printed out. The name stored by the constant will be used for the column name, not the name of the constant.

If constants are named sensibly they can help to make code much clearer.

Question 3

Correct

Question text

Which of the following would retain just the last 50 rows from a dataframe df with 100 rows?

Select one:



df = df[51:]



df = df[51:100]



df = df[50:]



df = df[50:99]

Feedback

Your answer is correct.

Correct answer:

df = df[50:]

The format of a splice is [m:n] where m is the index of the first row required and n-1 is the index of the last row required. So this returns the rows 50-99. The index of the last row is 99 as indexing starts from 0.

Question 4

Correct

Question text

The Spearman correlation coefficient of two sets of values is -0.82. Which of these statements is true?

Select one:



It is an inverse correlation.

A negative correlation means that values tend to go in opposite directions (increasing values in one set, decreasing values in the other).



It is a statistically significant correlation.



It is a direct correlation.



It is a very weak correlation, close to the minimum of -1.

Feedback

Your answer is correct.

Question 5

Correct

Question text

Which of the following attributes of plot() would you use to make a chart larger?

Select one:



logx



kind



figsize



grid

Feedback

Your answer is correct.

Correct answer:

figsize

figsize: takes a tuple value of (width, height) in inches.

kind: changes the type of chart. Valid values for kind include "scatter", "line", "bar", "hist", "pie".

grid: causes the grid lines to be displayed.

logx: causes the scale on the x axis to be logarithmic.