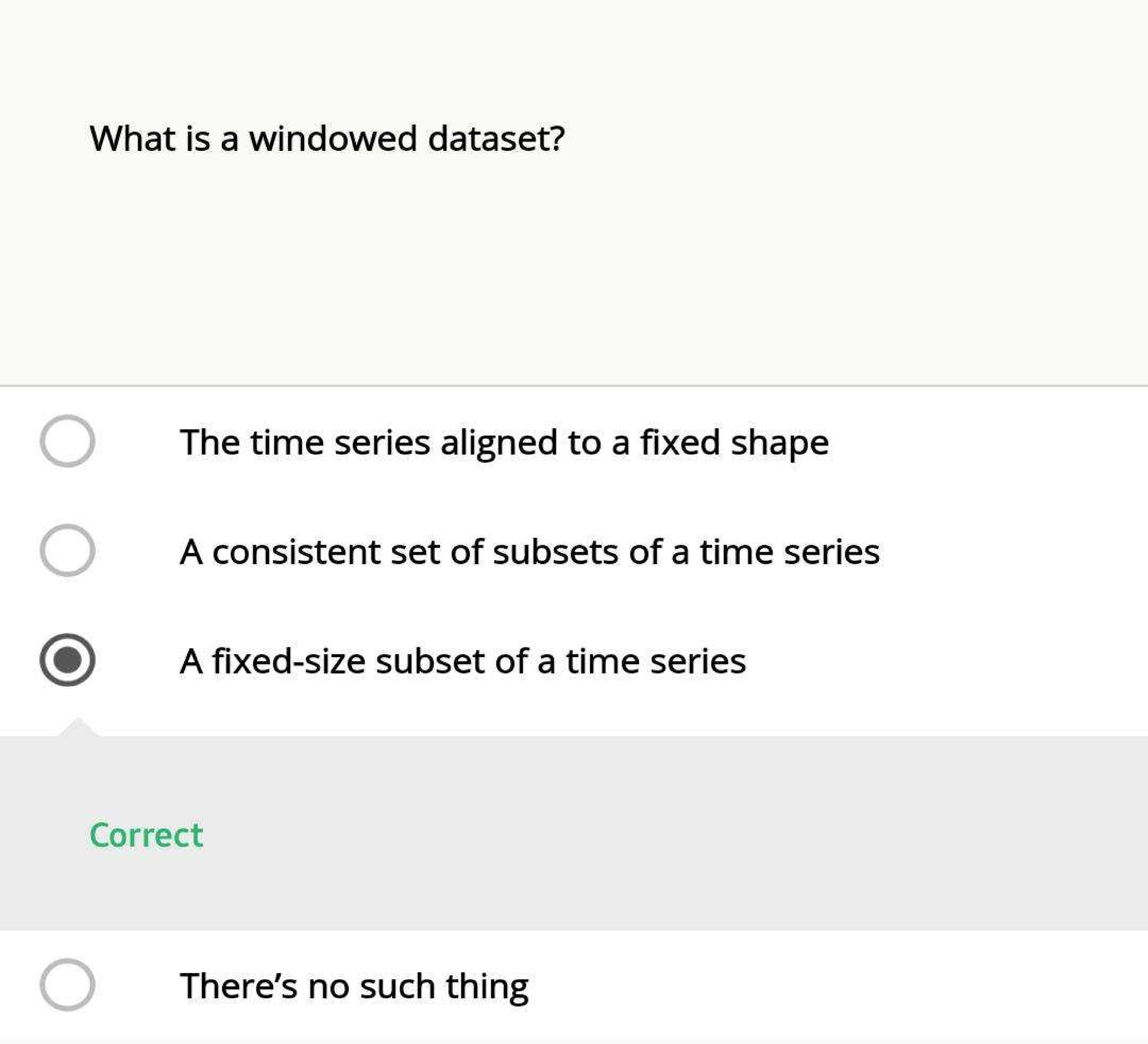


What is a windowed dataset?

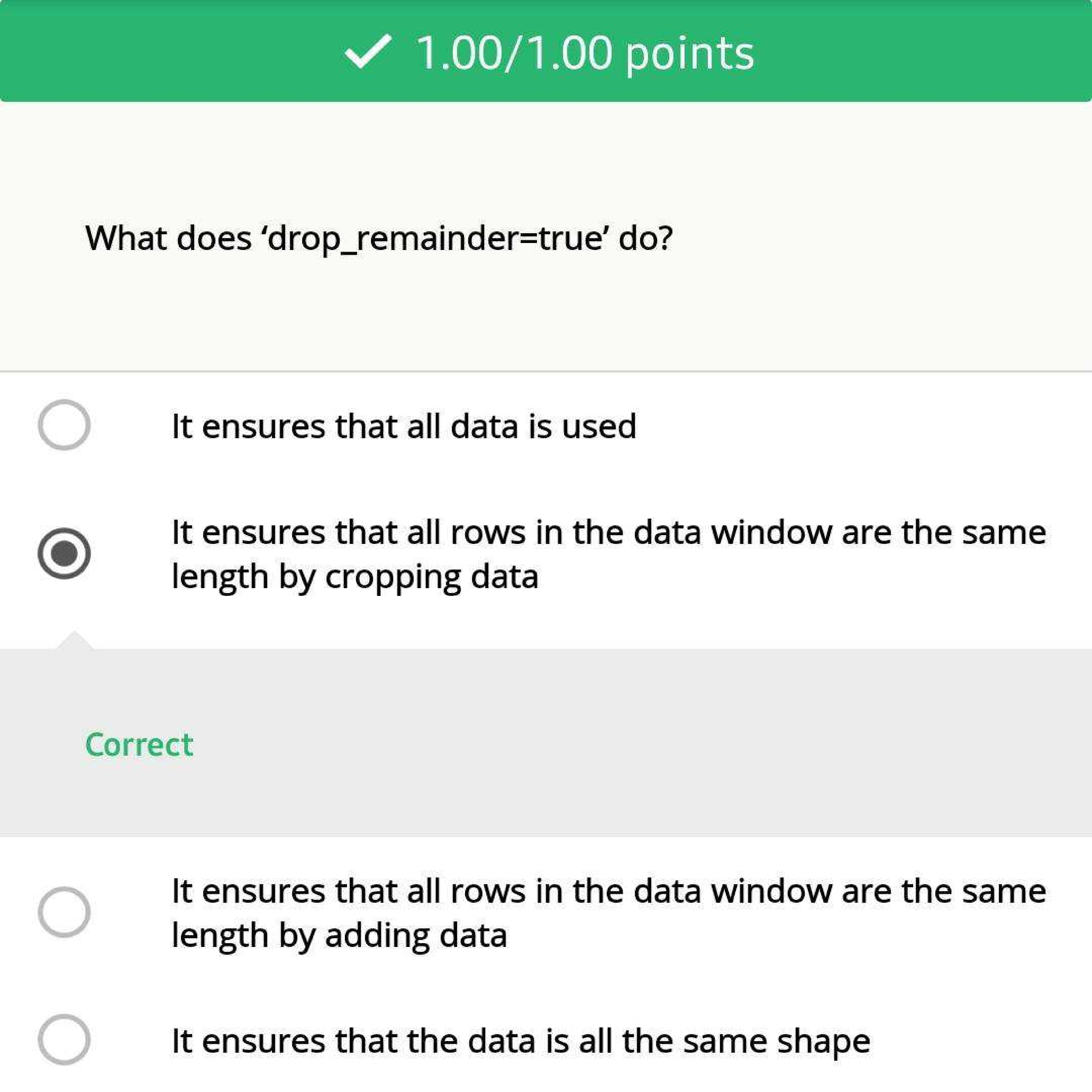
○ The time series aligned to a fixed shape

○ A consistent set of subsets of a time series

● A fixed-size subset of a time series

Correct

○ There’s no such thing



✓ 1.00/1.00 points

What does ‘drop_remainder=true’ do?

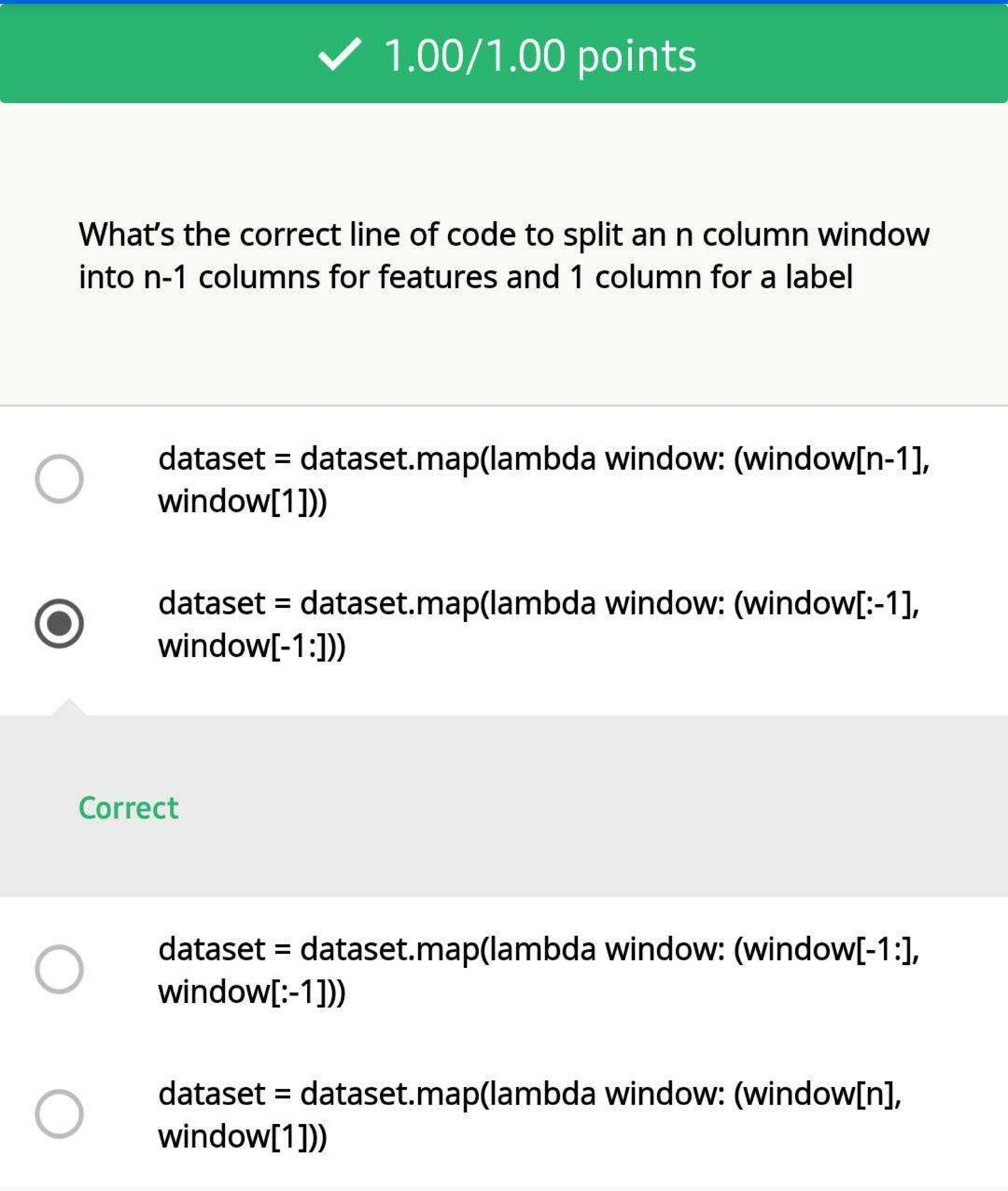
○ It ensures that all data is used

● It ensures that all rows in the data window are the same length by cropping data

Correct

○ It ensures that all rows in the data window are the same length by adding data

○ It ensures that the data is all the same shape



✓ 1.00/1.00 points

What’s the correct line of code to split an n column window into n-1 columns for features and 1 column for a label

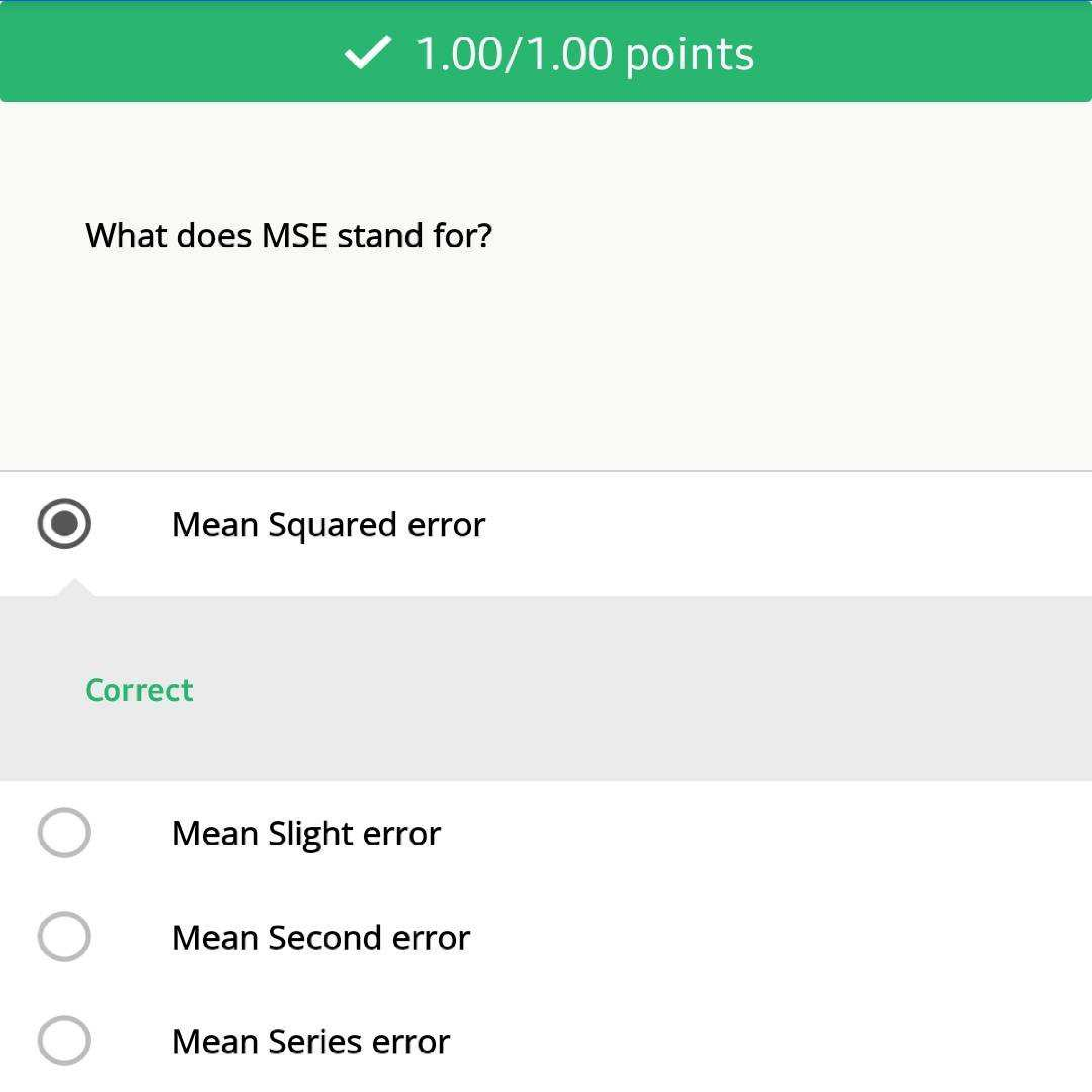
○ dataset = dataset.map(lambda window: (window[n-1], window[1]))

● dataset = dataset.map(lambda window: (window[:-1], window[-1:]))

Correct

○ dataset = dataset.map(lambda window: (window[-1:], window[:-1]))

○ dataset = dataset.map(lambda window: (window[n], window[1]))



✓ 1.00/1.00 points

What does MSE stand for?

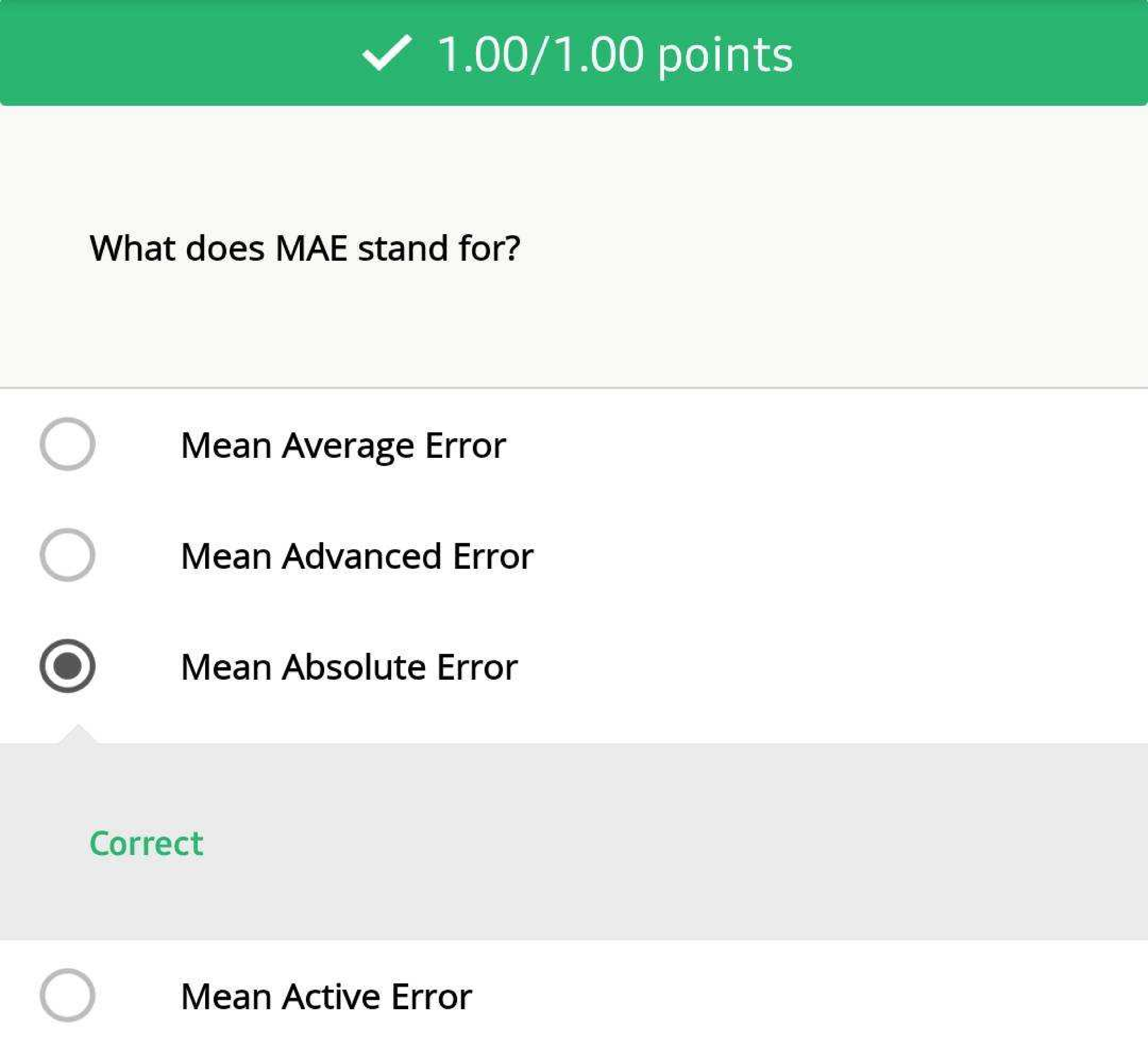
● Mean Squared error

Correct

○ Mean Slight error

○ Mean Second error

○ Mean Series error



✓ 1.00/1.00 points

What does MAE stand for?

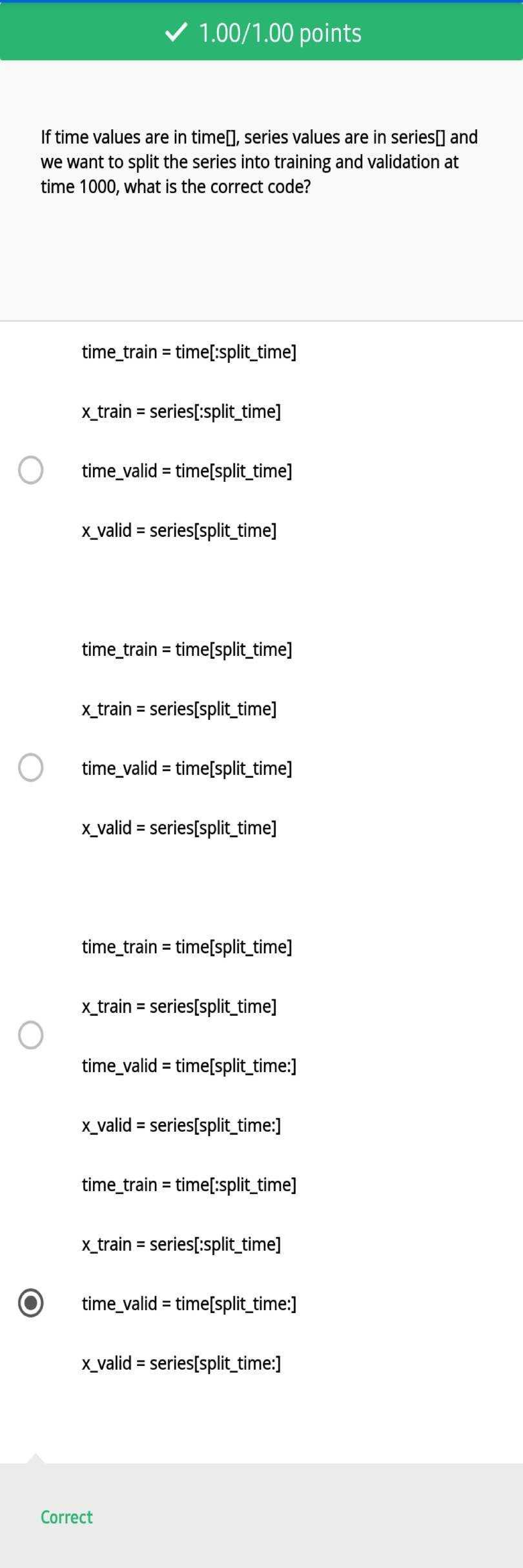
○ Mean Average Error

○ Mean Advanced Error

● Mean Absolute Error

Correct

○ Mean Active Error



✓ 1.00/1.00 points

If time values are in time[], series values are in series[] and we want to split the series into training and validation at time 1000, what is the correct code?

○

```
time_train = time[:split_time]

x_train = series[:split_time]

time_valid = time[split_time]

x_valid = series[split_time]
```

○

```
time_train = time[split_time]

x_train = series[split_time]

time_valid = time[split_time]

x_valid = series[split_time]
```

○

```
time_train = time[split_time]

x_train = series[split_time]

time_valid = time[split_time:]

x_valid = series[split_time:]
```

●

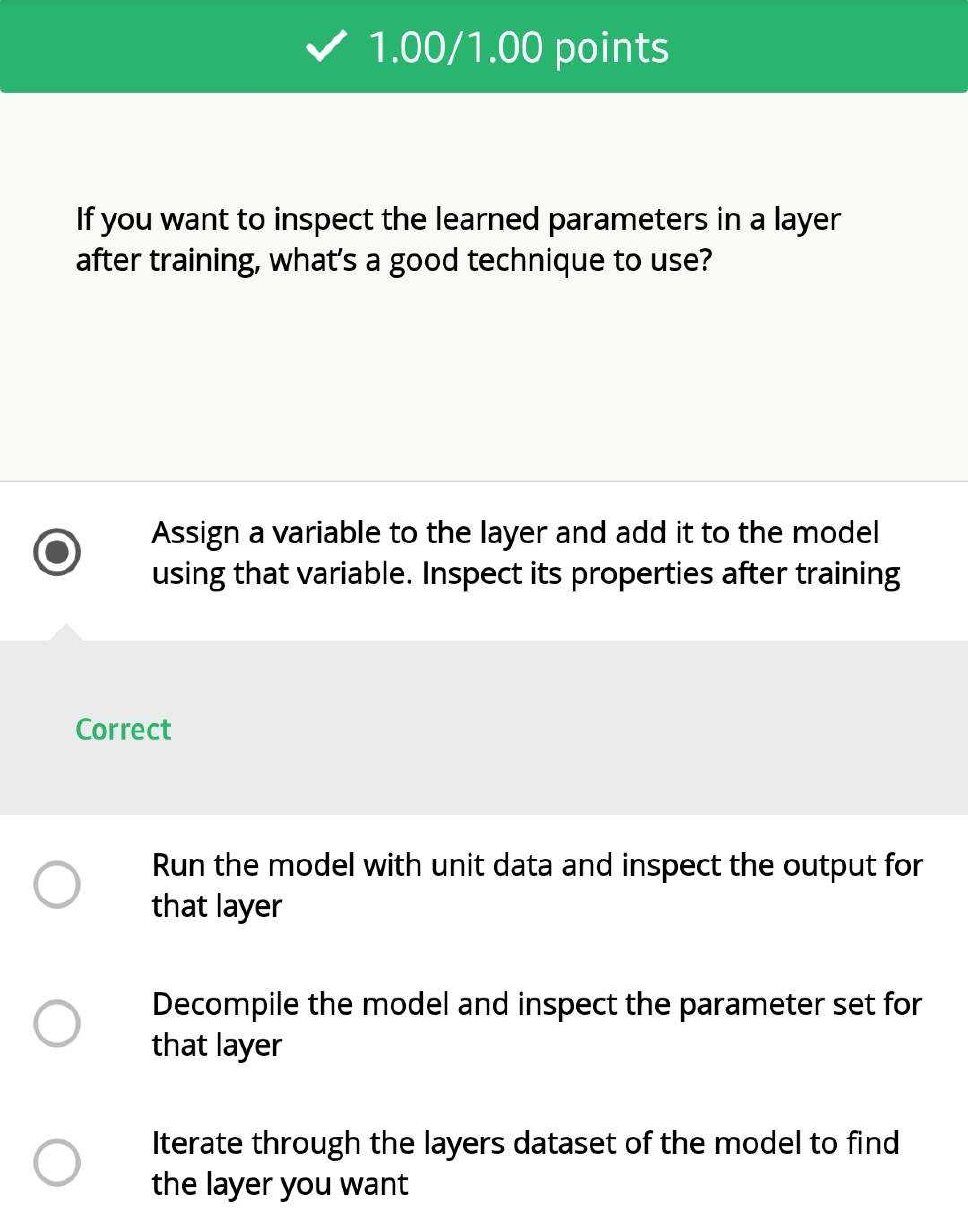
```
time_train = time[:split_time]

x_train = series[:split_time]

time_valid = time[split_time:]

x_valid = series[split_time:]
```

Correct



✓ 1.00/1.00 points

If you want to inspect the learned parameters in a layer after training, what's a good technique to use?

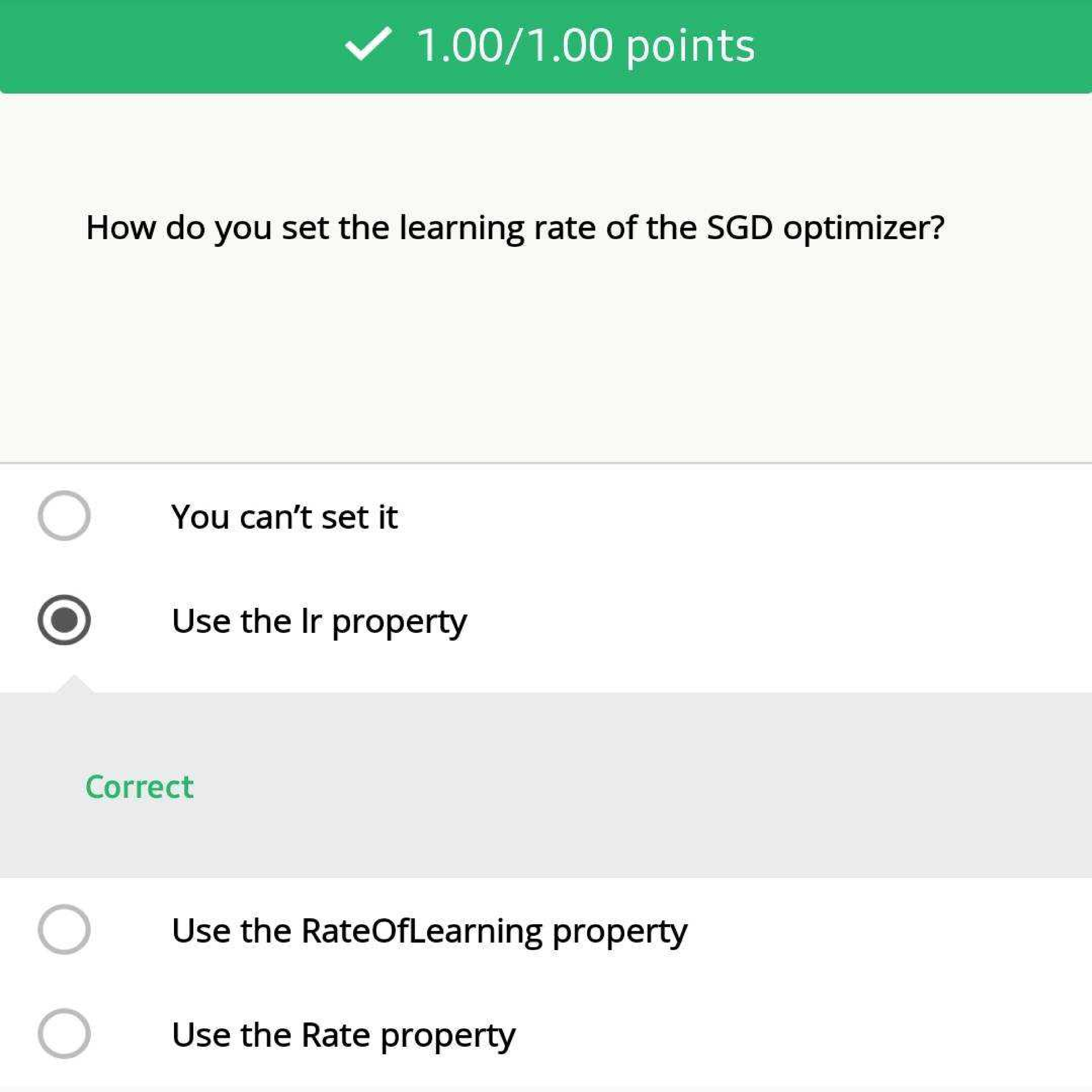
● Assign a variable to the layer and add it to the model using that variable. Inspect its properties after training

Correct

○ Run the model with unit data and inspect the output for that layer

○ Decompile the model and inspect the parameter set for that layer

○ Iterate through the layers dataset of the model to find the layer you want



✓ 1.00/1.00 points

How do you set the learning rate of the SGD optimizer?

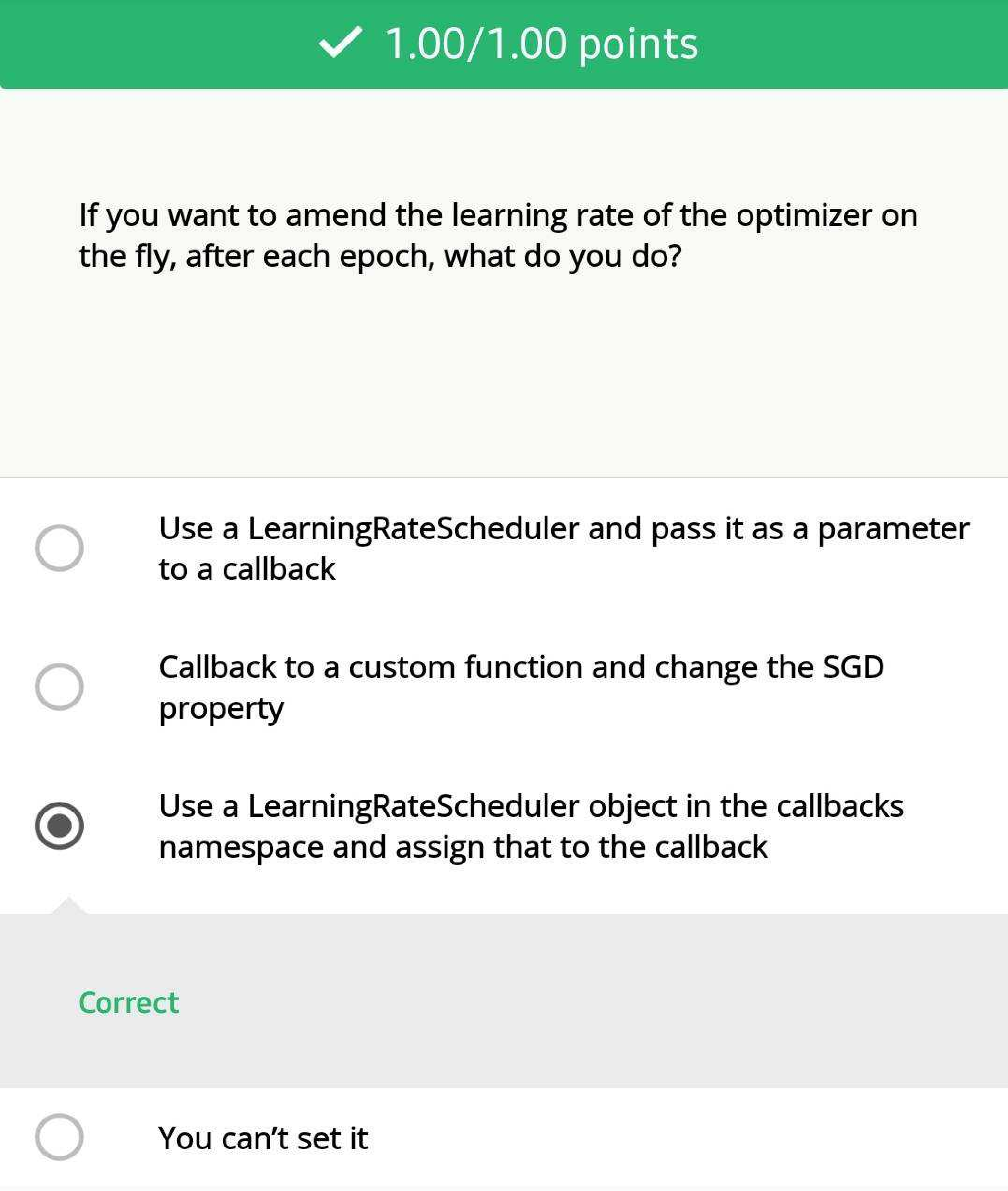
○ You can’t set it

● Use the lr property

Correct

○ Use the RateOfLearning property

○ Use the Rate property



✓ 1.00/1.00 points

If you want to amend the learning rate of the optimizer on the fly, after each epoch, what do you do?

○ Use a LearningRateScheduler and pass it as a parameter to a callback

○ Callback to a custom function and change the SGD property

● Use a LearningRateScheduler object in the callbacks namespace and assign that to the callback

Correct

○ You can't set it