

Different Ways of Importing in Python

Below is the image showing the question from DataCamp:

The screenshot shows a web browser window displaying a DataCamp exercise page. The browser's address bar shows the URL: `campus.datacamp.com/courses/intro-to-python-for-data-science/chapter-3-functions-and-packages?ex=12`. The page title is "Different ways of importing". The exercise text states: "There are several ways to import packages and modules into Python. Depending on the import call, you'll have to use different Python code. Suppose you want to use the function `inv()`, which is in the `linalg` subpackage of the `scipy` package. You want to be able to use this function as follows: `my_inv([[1,2], [3,4]])`. Which `import` statement will you need in order to run the above code without an error?" The "Possible answers" section lists four options:
1. `import scipy` (selected)
2. `import scipy.linalg`
3. `from scipy.linalg import my_inv`
4. `from scipy.linalg import inv as my_inv`
A green "Submit Answer" button is visible. Below the answers is a "Take Hint (-15 XP)" button. The right side of the page shows an "IPython Shell" interface with a prompt "In [1]:". The bottom of the image shows a Windows taskbar with various application icons and a system clock showing 20:24 on 11/11/2024.

Explanation of the Question:

The question asks about which import statement is needed to use the function `inv()` from the `scipy.linalg` subpackage so that the given code can run without errors.

Answer:

```
from scipy.linalg import inv as my_inv
```

Explanation of the Answer:

The function `inv()` belongs to the `scipy.linalg` subpackage. To use it with the alias `my_inv`, the correct syntax is `'from scipy.linalg import inv as my_inv'`.

This allows the function to be referenced as 'my_inv' without having to import the entire scipy or scipy.linalg package.