

## loc and iloc (3)

Exercise

### loc and iloc (3)

It's also possible to select only columns with `loc` and `iloc`. In both cases, you simply put a slice going from beginning to end in front of the comma:

```
cars.loc[:, 'country']
cars.iloc[:, 1]
cars.loc[:, ['country', 'drives_right']]
cars.iloc[:, [1, 2]]
```

Instructions

- Print out the `drives_right` column as a Series using `loc` or `iloc`.
- Print out the `drives_right` column as a DataFrame using `loc` or `iloc`.
- Print out both the `cars_per_cap` and `drives_right` column as a DataFrame using `loc` or `iloc`.

Take Hint (-50 XP)

```
script.py
1 # Import cars data
2 import pandas as pd
3 cars = pd.read_csv('cars.csv', index_col = 0)
4
5 # Print out drives_right column as Series
6
7 # Print out drives_right column as DataFrame
8
9 # Print out cars_per_cap and drives_right as DataFrame
10
11
12
13
```

Run Code Submit Answer

Python Shell

```
In [1]:
```

### Question:

1. Print out the `drives_right` column as a Series using `loc` or `iloc`.
2. Print out the `drives_right` column as a DataFrame using `loc` or `iloc`.
3. Print out both the `cars_per_cap` and `drives_right` columns as a DataFrame using `loc` or `iloc`.

### Answer:

```
# Import cars data
import pandas as pd
cars = pd.read_csv('cars.csv', index_col=0)
```

```
# Print out drives_right column as Series
print(cars['drives_right']) # or cars.loc[:, 'drives_right']
```

```
# Print out drives_right column as DataFrame
print(cars[['drives_right']]) # or cars.loc[:, ['drives_right']]
```

```
# Print out cars_per_cap and drives_right columns as DataFrame
print(cars[['cars_per_cap', 'drives_right']]) # or cars.loc[:, ['cars_per_cap',
```

```
'drives_right']]
```

## Code Explanation:

1. ``cars['drives_right']``: Accesses the 'drives\_right' column directly as a Series.

Alternatively, ``cars.loc[:, 'drives_right']`` does the same using the loc method, where the colon selects all rows and the column name selects the column.

2. ``cars[['drives_right']]``: Accesses the 'drives\_right' column as a DataFrame.

Alternatively, ``cars.loc[:, ['drives_right']]`` does the same using loc, where the colon selects all rows and the column name in a list selects the column as a DataFrame.

3. ``cars[['cars_per_cap', 'drives_right']]``: Selects the 'cars\_per\_cap' and 'drives\_right' columns as a DataFrame.

Alternatively, ``cars.loc[:, ['cars_per_cap', 'drives_right']]`` does the same using loc, where the colon selects all rows and the column names in a list select the columns as a DataFrame.