Learning Objectives

In the second module of this course, you will learn how to import Data Frames and select information from a Data Frame.

Learning Objectives:

1. Selecting columns
2. Selecting rows
3. Selecting rows and columns based on values

Selecting Data

### Selecting Columns

We can use `.info()` in pandas to get our column names. Once you have that information, there are two main ways to print information for a specific column.

1. `data.ColumnName`
2. `data['Column Name']`

The first `.column`option is used more frequently, when your column name has special characters or spaces you can use the bracket method. We will return to our movie CSV and print the individual rows.

First, we can use the `.info()` function to acquire the row names.

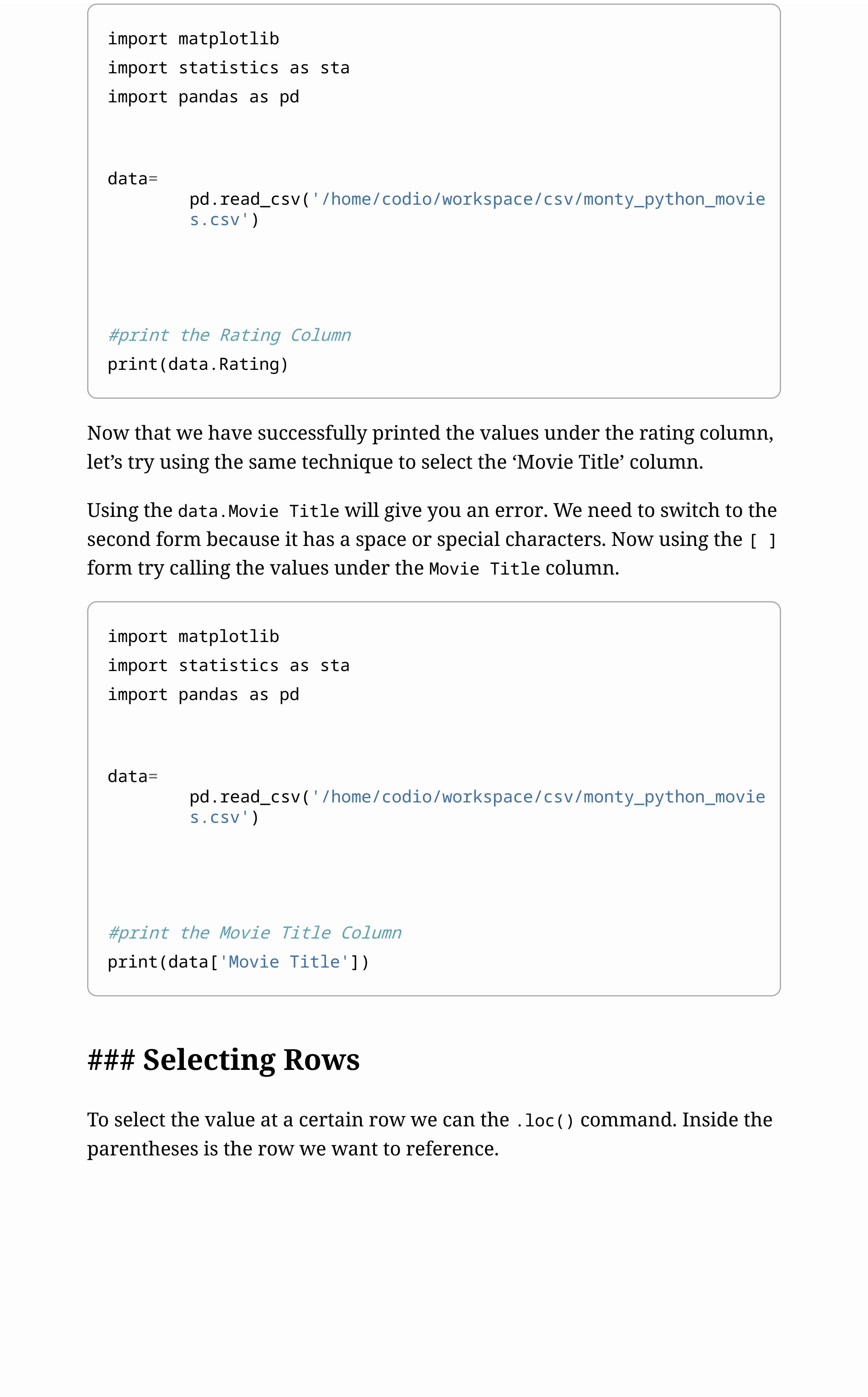
```
import matplotlib
import statistics as sta
import pandas as pd


data=
        pd.read_csv('/home/codio/workspace/csv/monty_python_movies.csv')


# printing the info in order to get the data column names

print(data.info())
```

Now that we know our movie has 2 columns, let’s first try printing the value in the column rating.



```
import matplotlib
import statistics as sta
import pandas as pd


data=
        pd.read_csv('/home/codio/workspace/csv/monty_python_movie
        s.csv')



#print the Rating Column
print(data.Rating)
```

Now that we have successfully printed the values under the rating column, let’s try using the same technique to select the ‘Movie Title’ column.

Using the `data.Movie Title` will give you an error. We need to switch to the second form because it has a space or special characters. Now using the `[ ]` form try calling the values under the `Movie Title` column.

```
import matplotlib
import statistics as sta
import pandas as pd


data=
        pd.read_csv('/home/codio/workspace/csv/monty_python_movie
        s.csv')



#print the Movie Title Column
print(data['Movie Title'])
```

### Selecting Rows

To select the value at a certain row we can the `.loc()` command. Inside the parentheses is the row we want to reference.

```
import matplotlib
import statistics as sta
import pandas as pd


data=
        pd.read_csv('/home/codio/workspace/csv/monty_python_movies.csv')



#print row 2


print(data.loc[2])
```

In order for us to select multiple rows based on value, we will use logical operators. Logical operators are used on conditional statements (either True or False). We can use arithmetic operators or boolean operators which return True or False.
Using movie rating, return all the valueS that are equal to 5.

```
import matplotlib
import statistics as sta
import pandas as pd


data=
        pd.read_csv('/home/codio/workspace/csv/monty_python_movies.csv')



#Using boolean show whether the value of rating is equal to 5
print(data.Rating==5)
```

The example above returns a column of True or False based on the logical operator, which in this case is whether the Rating in that row is equal to 5. To return the row where logical operator evaluates to True, we must put the logical operator inside brackets.

```
import matplotlib
import statistics as sta
import pandas as pd



data=
        pd.read_csv('/home/codio/workspace/csv/monty_python_movie
        s.csv')




#prints out all rows with a data rating of 5
print(data[data.Rating==5])
```

Statistics of selected data

Descriptive statistics of DataFrames

We previously used descriptive statistics to find the **mean**, **median**, **mode**,and **range** of a list. Now we are going to apply descriptive statistics to our Data Frames.

First, let's find the most common given rating from our CSV. For that we have `sta.mode()` and inside our parentheses we will specify the portion of our data we want THE selected. In this case, we have to use the rating columns. Replace the `print` statement from the last line of the example on the previous page with the following:

```
print(sta.mode(data.Rating))
```

Create the following variables.

- **data_mode** which stores the mode of our rating columns
- **data_mean** which stores the mean of our rating columns
- **data_median** which stores the median of our rating columns
- **data_range** which stores the range of our rating columns

Often we will need the sum of all the values in a column. For that we use `.sum()` function after selecting which column. For example,

```


print(data.Rating.sum())
```