In this set of exercises, we will be working with the Wine Reviews DataSet.

The first step in most data analytics projects is reading the data file. In this exercise, you’ll create Series and DataFrame objects, both by hand and by reading data files.

# Exercises 1:

Select the description column from the reviews and assign the result the variable desc.

|  |
| --- |
| desc = wine\_rev.loc[:, "description"]  print(desc) |

# Exercises 2:

Select the first value from the description column of reviews, assigning it to variable first\_description.

|  |
| --- |
| first\_description = wine\_rev.loc[0, "description"]  print(first\_description) |

# Exercises 3:

Select the first row of data (the first record) from reviews, assigning it to the variable first\_row.

|  |
| --- |
| first\_row = wine\_rev.iloc[1]  print(first\_row) |

# Exercises 4:

Select the first 10 values from the description column in reviews, assigning the result to variable first\_description.

|  |
| --- |
| first\_description **=** wine\_rev["description"].*head*(10)  *print*(first\_description) |

# Exercise 5

Select the records with index labels 1, 2, 3, 5, and 8, assigning the result to variable sample\_reviews.

In other words, generate the following data frame:

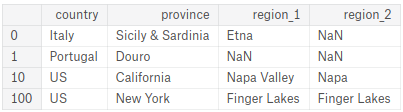


|  |
| --- |
| sample\_reviews **=** wine\_rev.iloc[[1,2,3,5,8]]  *print*(sample\_reviews) |

# Exercise 6

Create a variable df containing the country, province, region\_1 and region\_2 columns of the records with the index labels 0, 1, 10, 100.

In other words, generate the following DataFrame:



Hint: Use the loc operator. (Note that it is also possible to solve this problem using the iloc operator, but this would require extra effort to convert each column name to a corresponding integer-valued index.)

|  |
| --- |
| df **=** wine\_rev.loc[[0,1,10,100], ["country", "province", "region\_1", "region\_2"]]  *print*(df) |

# Exercise 7.

Create a variable df containing the country and variety columns of the first 100 records.

Hint: you may use loc or iloc. When working on the answer this question and the several of the ones that follow, keep the following "gotcha" described in the tutorial:

iloc uses the Python stdlib indexing scheme, where the first element of the range is included and the last one excluded. loc, meanwhile, indexes inclusively.

This is particularly confusing when the DataFrame index is a simple numerical list, e.g. 0,...,1000. In this case df.iloc[0:1000] will return 1000 entries, while df.loc[0:1000] return 1001 of them! To get 1000 elements using loc, you will need to go one lower and ask for df.iloc[0:999].

|  |
| --- |
| df **=** wine\_rev.loc[:99, ["country", "variety"]]  *print*(df) |

# Exercise 8.

Create a DataFrame italian\_wines containing reviews of wines made in Italy.

|  |
| --- |
| italian\_wines **=** wine\_rev.loc[wine\_rev["country"] **==** "Italy"]  *print*(italian\_wines) |

# Exercise 9.

Create a DataFrame top\_oceania\_wines containing all reviews with at least 95 points (out of 100) for wines from Australia or New Zealand.

|  |
| --- |
| top\_oceanic\_wines **=** wine\_rev.loc[(wine\_rev["points"] **>=** 95) **&** (wine\_rev["country"].*isin*(["Australia", "New Zealand"]))]  *print*(top\_oceanic\_wines) |

### Good Job.