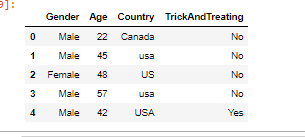
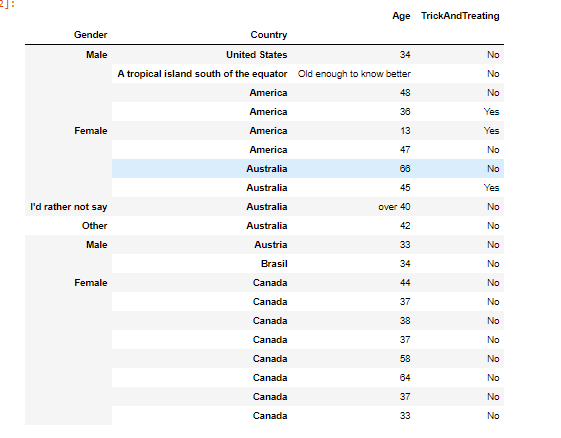
Traditionally, when we store data in a database table, we define one or more indices that have unique value across the tables. For example, we could have a table with social security number and employees and the rest of the dataset has information about date of hire, salary, region, etc. Another table could have the same indices, but he information in this table is related to perhaps medical history. With indexing we can ‘link’ tables together and augment the information the exists in those tables.

In some cases the table is index on something that does not a have a unique value (like social security), but can have a finite number of values such as ‘Male’,’Female’,’Other’. In this case we can have multiple records with these values as shown below. Now, let’s say we have a second index, ‘Country’; which can have a finite number of values.



If we index on Gender and Country, we get this:



This is called Hierarchical indexing. With this method, we can index in as many levels as it makes sense. For example, we could index further with setting ‘TrickAndTreating’ as the 3rd index. We can then sort on Gender(level=1), Country(level=2), or TrickAndTreating’(level=3).

Hierarchical indexing is a powerful tool in data wrangling. In Python and pandas, there are numerous techniques and features we can use to manipulate and extract data from and otherwise messy set od data.

References

1. Hierarchical Indexing - <https://jakevdp.github.io/PythonDataScienceHandbook/03.05-hierarchical-indexing.html>