Data manipulation

In data science, it is important to know how to extract exact information that is useful for the user. In this course, we will learn how to import that from a .csv file, how to filter rows that are important for us, how to sort that data, how to group that data, and in the other part of the course how to plot that data.

First, we should learn the basics of pandas so we can manipulate data with larger databases. Pandas has DataFrame and that is a 2 dimensional data structure, like a 2 dimensional array, or a table with rows and columns.

Firstly we will learn how to make data and DataFreme. We will define our data through a hashmap. We will set three columns “Food”, “Calories” and “Proteins”. You can put food that you like with its calories and proteins. You can see the example below in Figure 1. After that, we have to make our DataFrame. To do that we will make an object, which is shown in figure 1. At the end, we will just print our data.

```
import pandas as pd

data = {
    "food": ["burger","pizza","fries"],
    "calories": [420, 380, 390],
    "proteins": [20, 25, 15]
}

#load data into a DataFrame object:
df = pd.DataFrame(data)

print(df)
```

Figure 1: example 1

	food	calories	proteins
0	burger	420	20
1	pizza	380	25
2	fries	390	15

Figure 2: results

The next step is to learn how to load this data from a .csv file. We can make an Excel table with the same columns and rows as in example 1. To import data from a .csv file we should use the read_csv function. As an argument, we should provide the name of our .csv file. This object is already DataFrame so we don’t have to add again to DataFrame. At the end, we can just print this data. The code sample is shown in Figure 3, result is the same as in Figure 2.

```
1   import pandas as pd
2
3   data = pd.read_csv('data.csv')
4
5   print(data)
```

Figure 3: example 2

Example 3: Now we want to manipulate this data. For example, we want to show only burger and fries. Also, we want to sort this table by calories ascending. To do this we need to specify what food we want to display. We will do this by making an array of food. Then we will filter our data. To do this we have to use the function isin to get a new DataFrame with data that we want to display. To sort data we will use the function sort_values. This code is shown in Figure 4 and the results are in Figure 5.

```
#Example 3
import pandas as pd

data = pd.read_csv('data.csv')

data_to_displey = ["burger","fries"]

data_fil = data[data["food"].isin(data_to_displey)]

print(data_fil.sort_values("calories",ascending=True))
```

Figure 4: example 3

```
     food  calories  proteins
2   fries       390        15
0  burger       420        20
```

Figure 5: results

Pandas allow us to do many other things with data. For example, we can get min or max values, or we want to count rows for specific data in tables. Some examples of this function are listed below.

- min() – Returns the minimum value of each column or row in a DataFrame.

```
df = pd.DataFrame({'A': [3, 6, 2], 'B': [8, 1, 9]})
print(df.min())  # Output: A=2, B=1
```

- max() – Returns the maximum value of each column or row in a DataFrame.

```
df = pd.DataFrame({'A': [3, 6, 2], 'B': [8, 1, 9]})
print(df.max())  # Output: A=6, B=9
```

- groupby() - Groups data based on a column or multiple columns and allows for aggregation and transformation of the grouped data.

```
df = pd.DataFrame({'Category': ['A', 'B', 'A', 'B'], 'Value': [1, 2, 3, 4]})
grouped = df.groupby('Category').sum()  # Sum values within each group
print(grouped)
# Output:
#           Value
# Category
# A             4
# B             6
```

- count() - Returns the number of non-null (non-missing) entries in each column or row.

```
df = pd.DataFrame({'A': [1, None, 3], 'B': [4, 5, None]})
print(df.count())  # Output: A=2, B=2
```

- mean() - Returns the mean (average) value of each column or row in a DataFrame.

```
df = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})
print(df.mean())  # Output: A=2, B=5
```