# Unit 1. Statistical software tools

Statistical analysis is nothing more than a set of calculations, they could be fairly simple (like computing average or standard deviation, or complex as running a cluster analysis). In any case at the end of the day those are just a set of calculations (math formulas computed following a certain algorithm) that convert an input data set into some output (a numerical value, score, labels etc.).

Any of that calculation may in fact be done with primitive tools (on paper or with a calculator).

As much as we would love to see students doing it manually in order to better understand, there is no need to do so. Our goal to be able to understand the situation that one or another method should be applied, apply it and interpret the results.

There are many tools on the market that can perform those methods.

Our intention is to build a tool-agnostic course, however, we need to provide some guidance. Here is our take on that.

As we will learn not all predictive techniques is a part of standard statistic and therefore they may not be included in the statistical software.

Depending on the complexity of the method various tool can be used (ranging from Excel to machine learning libraries for Python or R). We will recommend a tool to use if you are equally unfamiliar with one or another. However, if you know another feasible tool (e.g. SAS or Stata), you may use it. However, the faculty help can be limited to his or her knowledge of those tools.

Here is a cheat sheet of the tools that can be used for methods covered in the class:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unit | Major Topics Covered | Excel | SPSS | Python or R |
| 1 | Descriptive statistics. | X | X | X |
| 2 | Hypothesis testing |  | X | X |
| 3 | Regression analysis |  | X (model not for predction) | X |
| 4 | Factor analysis |  | X | X |
| 5 | Classification and clusterization |  |  | X |
| 6 | Decision trees and decision-making with predictive analytics |  |  | X |
| 7 | Predictive analytics applications: marketing and product launch. |  | X | X |
| 8 | Final project |  | X | X |

Below you can find a short notes, on how to start with each of the tool, an in the later modules there will be more detailed instruction relevant to the specific methods.

# Python

By now you would be expected to have some basic understanding of the Python workflow, but if you don’t have here are few tips that you may need to know.

Python may come in various forms from a wed-based environment (PythonFiddle of Python Everywhere) to an installed version on your computer. We recommend to install Anaconda, which is Python set up for data analysis that comes with multiple additional tools (<https://www.anaconda.com/download/>).

Anaconda’s default Python editor is called Spyder, which you can run from Anaconda’s start screen.

Then you can create projects or individual python files. Read more on Spyder here: <https://docs.spyder-ide.org/>

When it comes to code, we will need to use Python’s statistical modules. You can read about the whole set up here: <https://www.scipy-lectures.org/packages/statistics/index.html>

For simple statistics you may use numpy with SciPy, while for more advance statistics you will need to use Pandas. The difference is in the methods that are supported by those libraries.

Here is a pandas sample for mean and standard deviation. If a csv file with a data is given the following simple code:

import pandas as pd

df = pd.read\_csv("test.csv")

**print**(df['VariableA'].mean())

**print**(df['VariableA'].std())

Press “Run”

Result:

0.47033200821052645

0.2274416582872558

See descriptive.py and test.csv files.

## R

R (<https://cran.r-project.org>) is an environment/language designed specifically for statistics and data analysis.

Working with R requires either install it first on your computer or use a university provided computer / virtual desktop with R preinstalled.

When you open R, you can wor in the R console or run scripts.

Let’s consider an example with a console:

If you are working with csv files, first, you need to make sure that you selected the correct directory (File->Change Dir) which matches the one where your data file is.

Then type:

dat <- read.csv("test.csv", header = TRUE)

And hit ‘Enter’

You can output your data:

dat

VariableA

1 0.56071392

2 0.51409902

…

Then you can calculate mean and standard deviation:

mean(dat[['VariableA']])

[1] 0.470332

sd(dat[['VariableA']])

[1] 0.2274417

## SPSS

SPSS is the most common statistical software which allows but does not require coding. It is designed to provide user-friendly graphic user interface.

SPSS can import Excel files or csv (<https://libguides.library.kent.edu/SPSS/ImportData>)

To import test.csv file use File->Open->Data

Once your data is loaded you can call command from the menu. For instance to get descriptive statistics you can use Analyze->Descriptive statistics->Descriptives

Then mean and standard deviation will be a part of the SPSS output. There are plenty of documentations and videos on how to do different things in SPSS.

As we do not encourage to limit your toolkit to **Excel** we do not include detailed instruction on using it for statistics which can be easily found online.

## Conclusion

Whichever tool you will chose to complete the assignment it is important to understand that they essentially do the same math, just use a different syntax or interface.