**Fundamentals of data science**  
Outlier detection

**Introduction**

Outliers in data science can be understood as atypical value of one variable or atypical combination of few features. As you already know it can be a consequence of measurement error, missing values or a presence of „information nugget”

Outlier detection methods:

* Univariate outlier detection
* Multivariate outlier detection

Your task is to implement one method of univariate and multivariate detection. All the theory needed is provided in the course slides.

**Task description**

Please load Haberman’s survival dataset (use first link from Google). Data file (haberman.data) contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer.

There are three attributes present

Age of patient at time of operation (numerical)

Patient's year of operation (year - 1900, numerical)

Number of positive axillary nodes detected (numerical)

And a class attribute:

1 = the patient survived 5 years or longer

2 = the patient died within 5 year

which we’re not using for outlier detection (!).

Please follow the list of tasks:

1. Draw three dimensional (so called group *scatter plot*) with classes indicated by different color of data points.
2. Please use one dimensional outlier detection using typical mean and standard deviation (3-sigma) method for all attributes separately. Note the number of outliers as **p**. Note: it should be done without the last column (class label).
3. Please find outliers using distance-based multivariate method, with k=5, **p** found in point II. Namely: outliers are **p** cases for which distance to the 5th nearest neighbor is the largest.
4. Are the results for both methods the same?

**Technical aspects**

* You can choose any software you like to implement your solution. MATLAB and Java compiler are already installed. If you prefer Python it is also available and portable distribution with Jupyter Notebook can be also downloaded (see WinPython).

C++ is strongly discouraged due to time needed to implement a working solution (unless you’re really a hardcore fan of this language)

You can also set up free Kaggle account and fork notebook:  
<https://www.kaggle.com/srikanthdasam/haberman-s-survival>

Google Colab can be used as well.

Useful Python functions:

* pandas.read\_csv
* pandas.DataFrame.mean
* pandas.DataFrame.var
* sklearn.neighbors.NearestNeighbors
* pandas.DataFrame.sort\_values