Zurich University
of Applied Sciences

School of
Engineering

P08 – Basic Data Analysis

1. Hockey statistics

The website **`nationalleague.ch`** provides standings and team statistics of the Swiss National Hockey League A. At the time of writing, SC Bern was leading, followed by HC Davos; whereas Fribourg and Lakers were second last and last, respectively.

As a Data Scientist you would like to know what the differences between high and low ranked teams are. In order to do that, team statistics of HC Davos (placed 2nd) and Fribourg (placed 2nd last) were extracted from the website and stored in an excel-sheet **`p08_hockey_stats.xlsx`**.

Explore these statistics using a combination of the external Python modules **`xlrd`**, **`numpy`** and **`matplotlib`**. More specifically we would like to know:

- What are **total** and **average** (per player) shots on goal?
- What are **total** and **average** (per player) goals scored?
- Display a **histogram** of the goals scored, i.e. how many players scored [0,1,2,…15] goals?
- Using a **pie chart**, visualize the percentage of goals scored by forward and defense players.
- How effective are the players?
    - What is the **correlation** between shots on goal and goals scored for each team (do players shooting a lot score a lot)?
    - What are the **linear regression** parameters (**a**,**b**) to predict number of goals scored from number of shots on goal: **`goals = a*shots +b`**?
    - Visualize the relationship between shots on goal and goals scored using a **scatter plot** and visualizing linear regression.

Create a script that for each team (i.e., work sheet in the Excel file):

1. Reads for each player the position he plays, number of shots on goal and the number of goals scored using **`xlrd`**
2. Analyze the data using **numpy** functions **`np.mean()`**, **`np.std()`**, **`np.sum()`**.
3. Plot a histogram, pie chart and scatter plot with the corresponding information

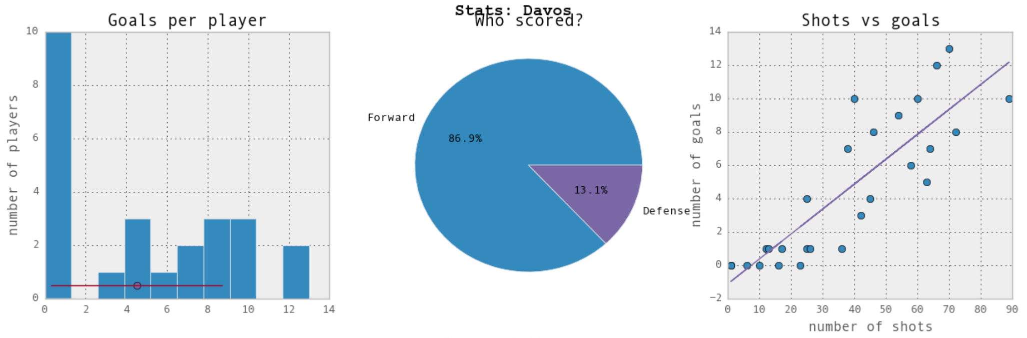
Using **numpy** (imported as **np**), the un-centered correlation coefficient **`r`** of two 1D vectors **`x`** and **`y`** can be calculated using the following lines of code:

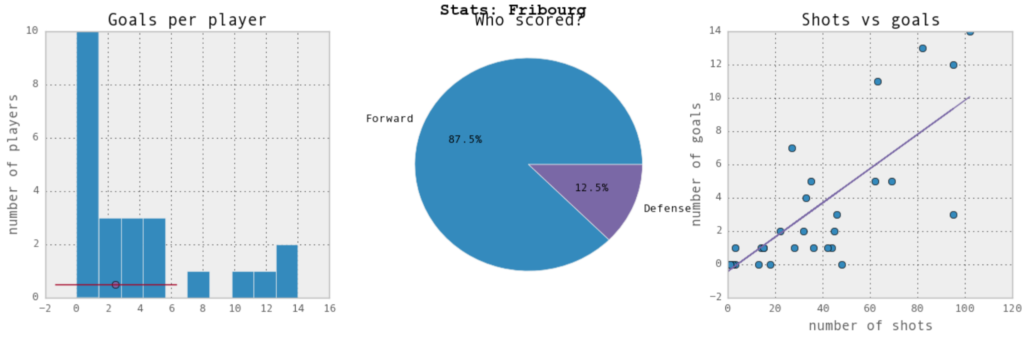
```
def get_correlation(x,y):
    r = x.dot(y) / (np.sqrt(x.dot(x) * y.dot(y)))
    return r
```

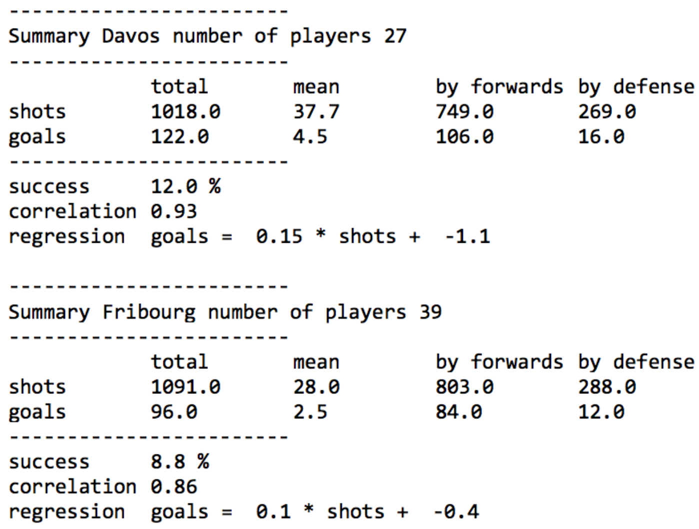
Using **numpy** (imported as **np**), linear regression parameters (**a, b**) for relating 1D vectors **xi** and **y** such that **y = a*xi+b**, can be calculated using the following function:

```
def get_linear_regression(xi,y):
    A = np.array([ xi, np.ones(len(xi))])
    a, b = np.linalg.lstsq(A.T,y)[0] # obtaining the parameters
    return a, b #return as a tuple
```

The output of your final script could look like this:







```
------------------------
Summary Davos number of players 27
------------------------
            total       mean        by forwards by defense
shots       1018.0      37.7        749.0       269.0
goals       122.0       4.5         106.0       16.0
------------------------
success     12.0 %
correlation 0.93
regression  goals =  0.15 * shots +  -1.1

------------------------
Summary Fribourg number of players 39
------------------------
            total       mean        by forwards by defense
shots       1091.0      28.0        803.0       288.0
goals       96.0        2.5         84.0        12.0
------------------------
success     8.8 %
correlation 0.86
regression  goals =  0.1 * shots +  -0.4
```

Hints:

- Pseudo code for this script could look like this:

```
• open the Excel file
• for each worksheet in Excel #this is: for each team
    o for each row except the first one #this is: for each player
        ▪ add number of goals to a list of goals
        ▪ add number of shots to a list of shots
        ▪ add number of penalties to a list of penalties
        ▪ add position to a list of positions #positions and goals/shots/penalties per player have corresponding indices
        ▪ compute correlation of goals and shots, using get_correlation()
        ▪ compute linear regression parameters of shots and goals, using get_linear_regression()
        ▪ create new lists of shots and goals per defense and forward
        ▪ print the text summary as displayed above, using the np-functions stated above and round()
        ▪ create a new figure (plot)
            • add subplot for histogram
            • add subplot for pie chart
            • add subplot for correlation
        ▪ show the plot and exit
```

- Create the histogram using **`plt.hist()`**, **`plt.ylim()`** and **`plt.plot()`**. Look up the first 6 parameters of the latter for the subtleties seen in the above figures.

- Create the pie chart using **`plt.pie()`**. Look up the parameters called **`labels`**, **`autopct`** and **`colors`** to recreate the above figure.
- Create the scatter plot using **`plt.plot()`** again, using an array containing the linear regression line (as computed by **`line = a*shots + b`**) as its 5th parameter