Which parameters are most important in cancer classification?

Have you ever thought what may cause malignant cancer?

Is this connected with area, maximum radius or irregularity of cancer?

The main aim of his project is to find most important features causes malignancy, but there is another value. I would like to make a model which will be able to predict if the cancer might be malignant witch accuracy at least of 0.9 and stability less than 3%.

So, let's go!

I would like to split project into parts:

1. My own data analysis based on data visualization
2. Taking care of missing values
3. Adding and removing columns
4. Getting dummies values for 'object' data
5. Fitting classifiers:
    - 5.1. Checking scores for basic data
    - 5.2. Find the most important features
6. Checking scores for final model
7. Evaluation

1. **My analysis:**

I would like to start with basic things connected with dataset – types of values, properties of dataset, skew of each column etc.

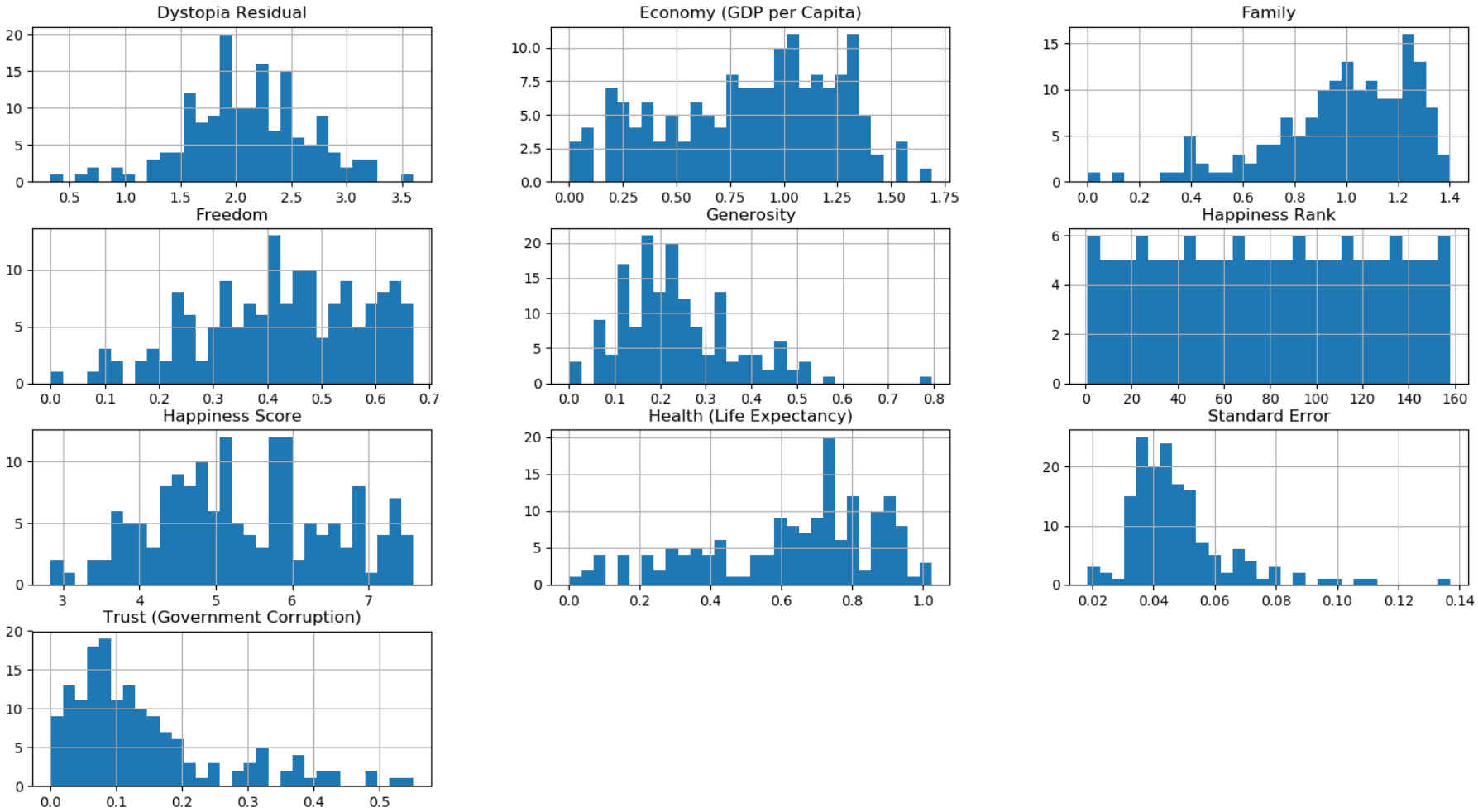
I would check it by functions:

df.dtypes -> it shows me types of each column
I know that there is only one object column -> 'diagnosis'. There are only Boolean values so I will map this column into 0 (normal cancer) and 1 (malignant).

df.describe() -> helps me to get the view of the data from mathematical side
Here, I may see that there is no missing values. Lucky day :)

df.skew() -> it count the skew of each numeric column, the closer to 0 the more resemble the normal distribution

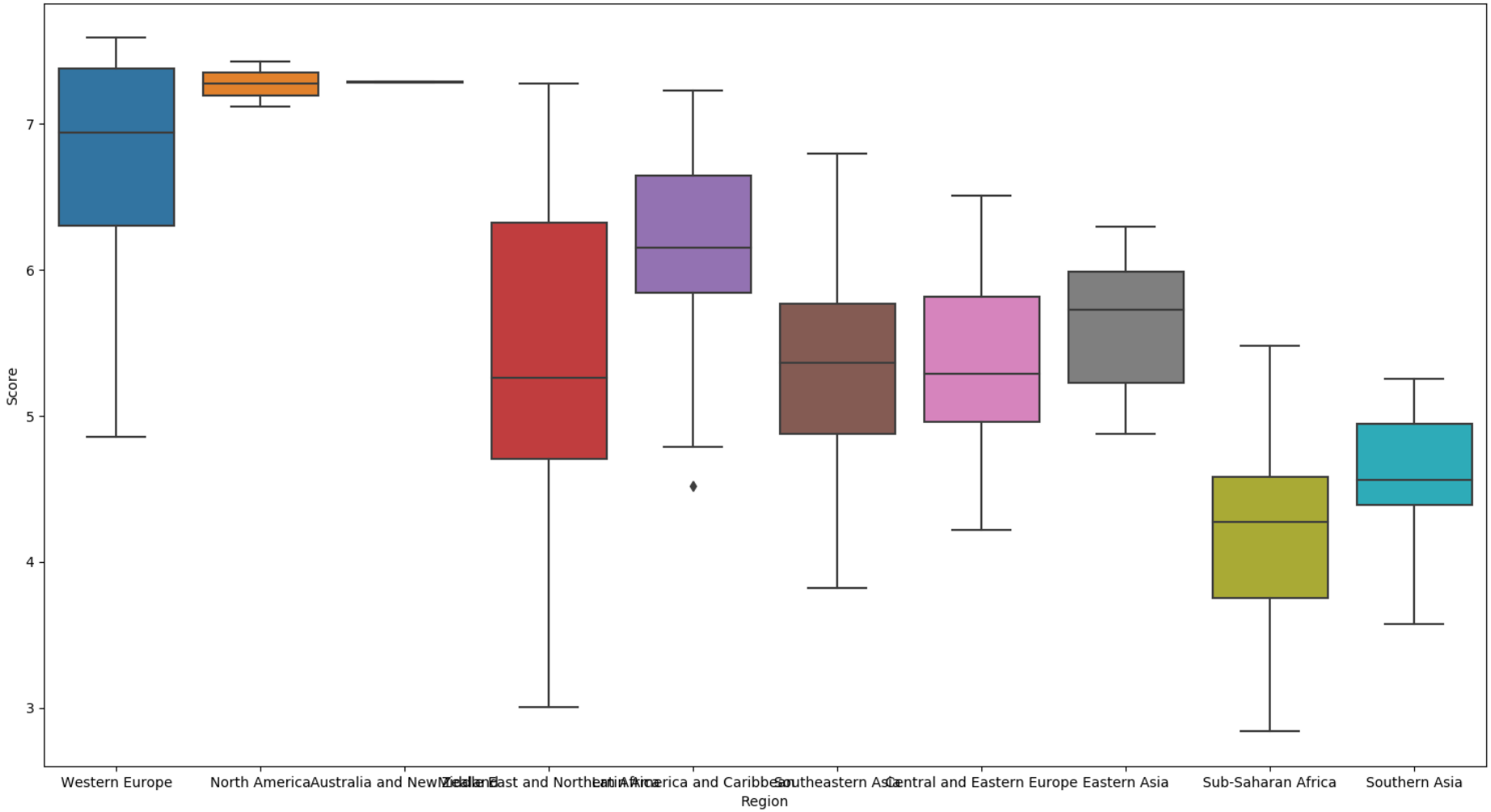
df.hist(bins = 30) -> shows histograms of each numerical data



In two cases (**'Economy', 'Health'**) we can see huge dispersion. We can observe large skew in majority of histograms. My aim will be Happiness Score so I should remove Happiness Rank column Looking at these histograms I may state that each NaN value I should replace by median value of right column.

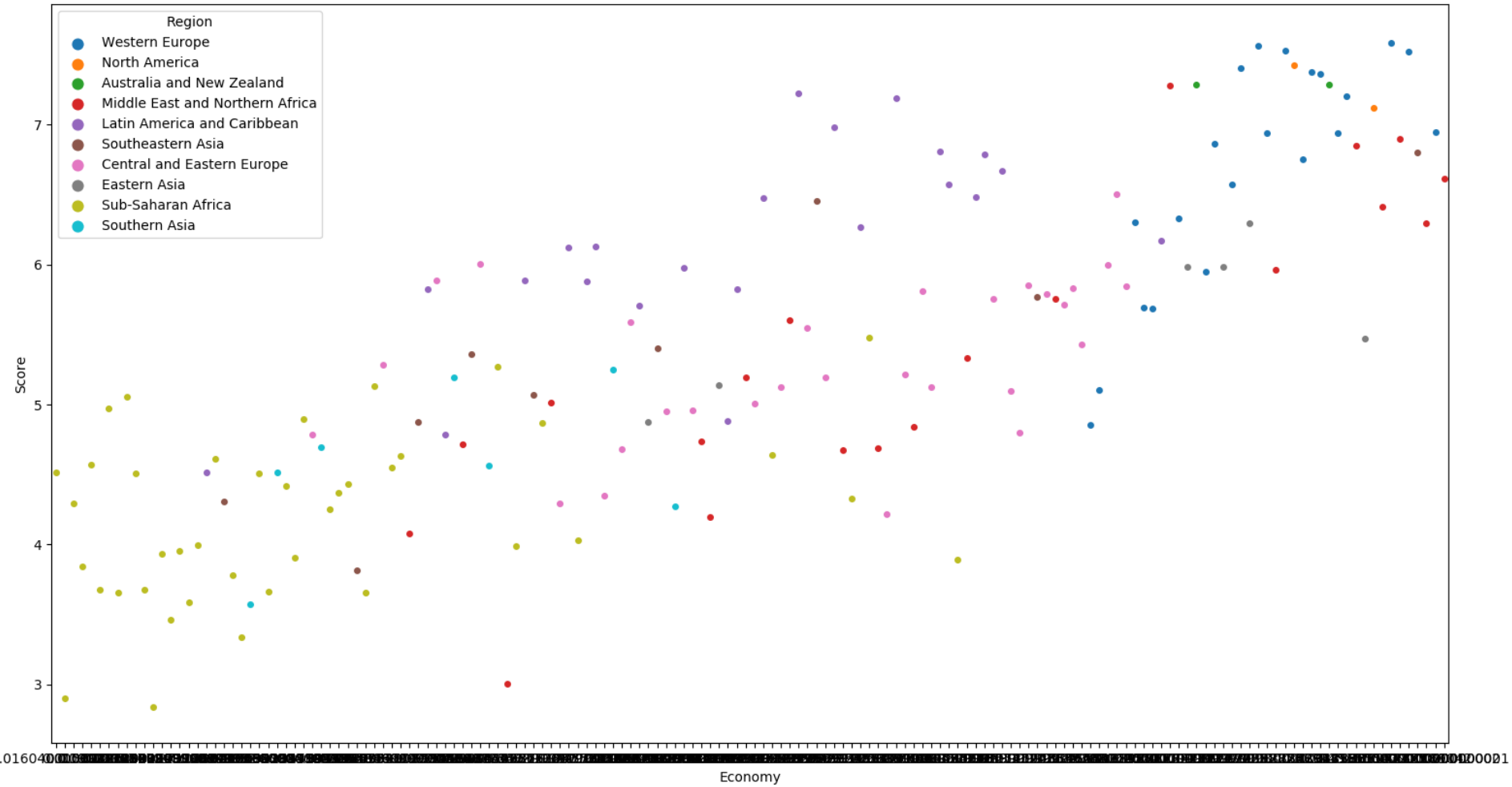
Visualizations:

Relation between **Region** and **Score**



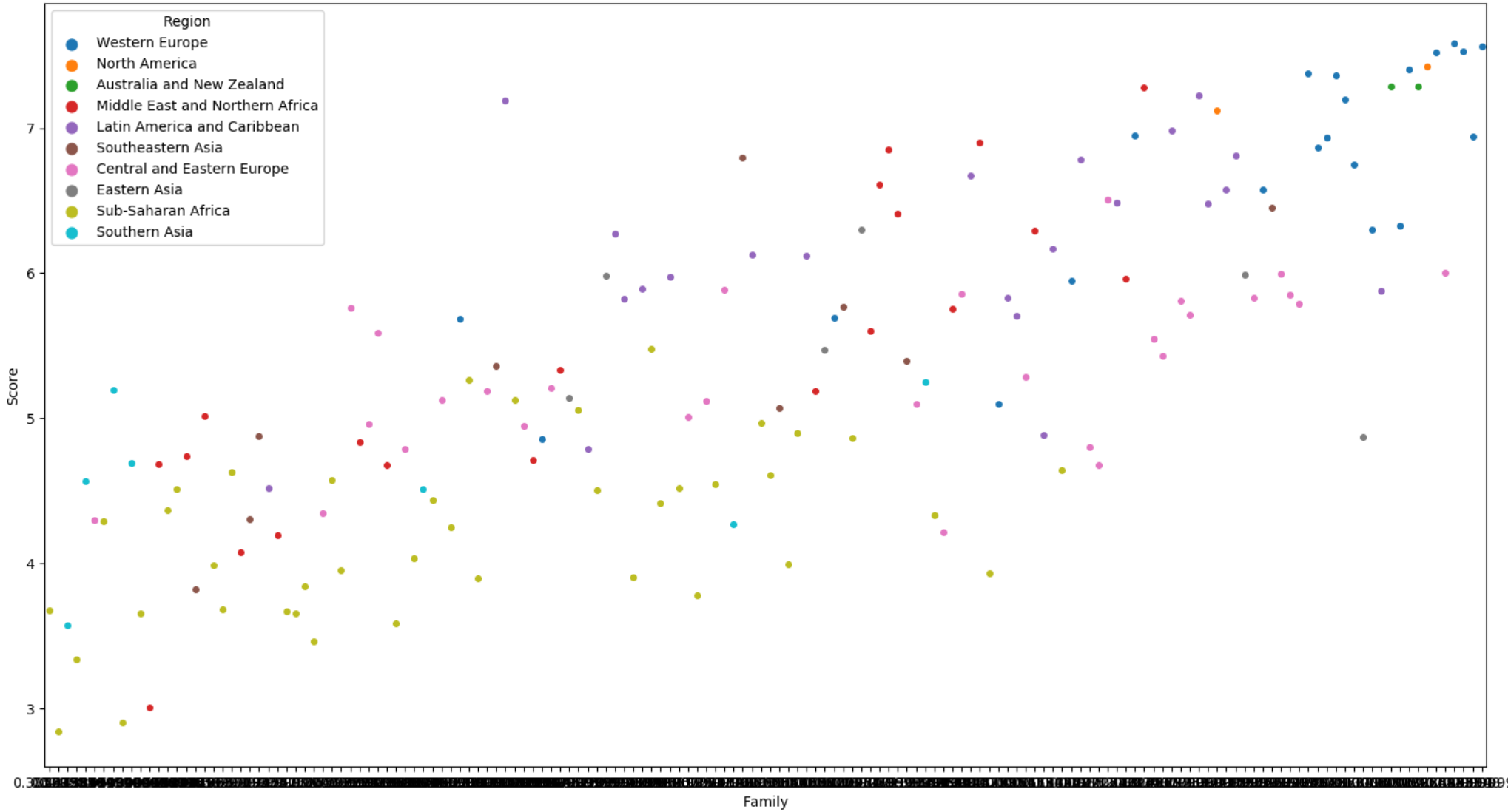
As we can see there is correlation between **'Region'** and **'Score'**. In next steps of project I will check connection between each variable in relation to **'Score'**. Each graph will be grouped by **'Region'.**

Relation between **Economy** and **Score**



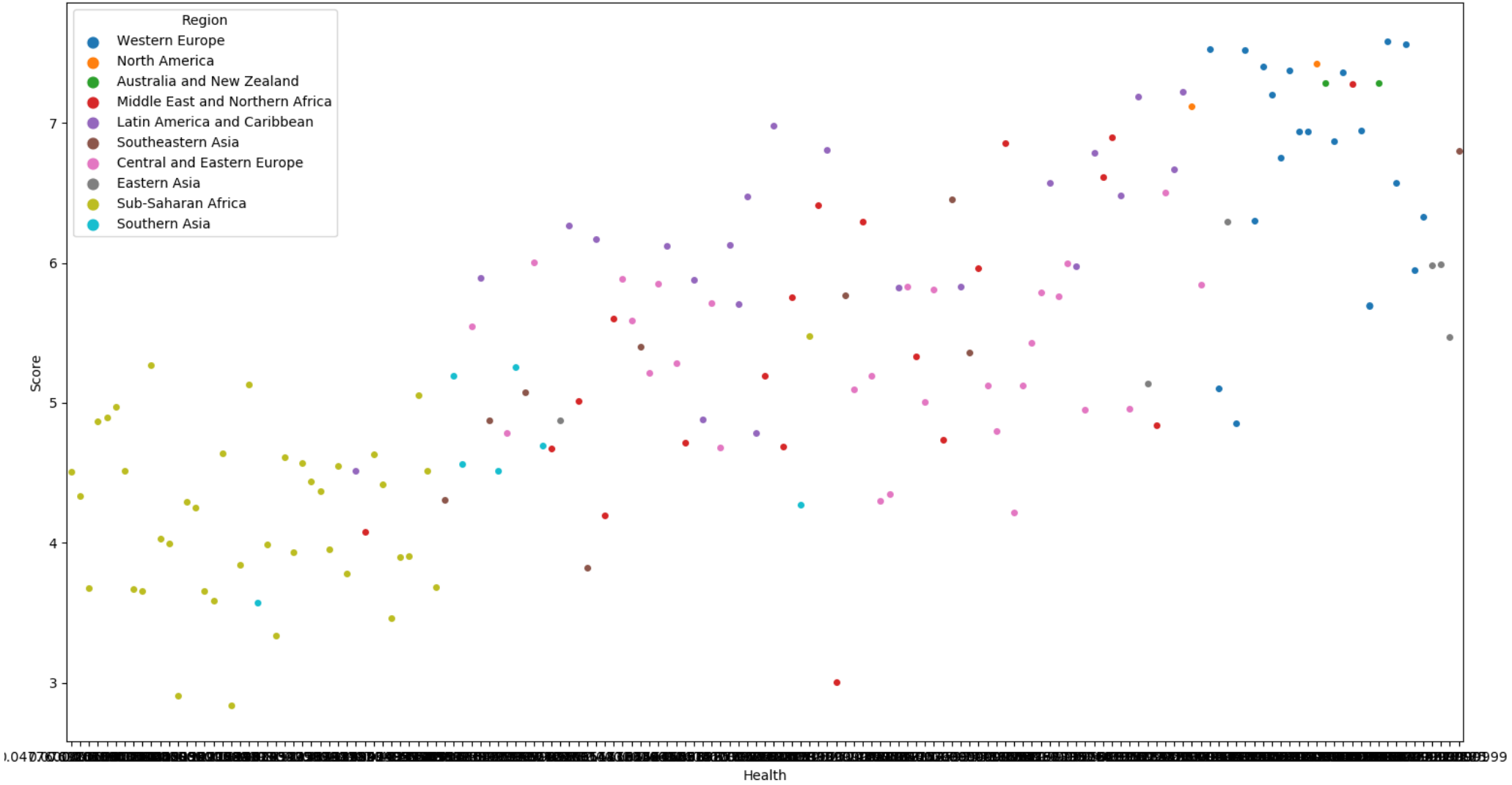
It seems to be linear relation: higher **Economy** -> higher **Score**. When we look at each region, we can see that only few countries are strongly grouped. From this graph we can draw conclusions that '**Economy'** value is more important than '**Region'**

Relation between **Family** and **Score**



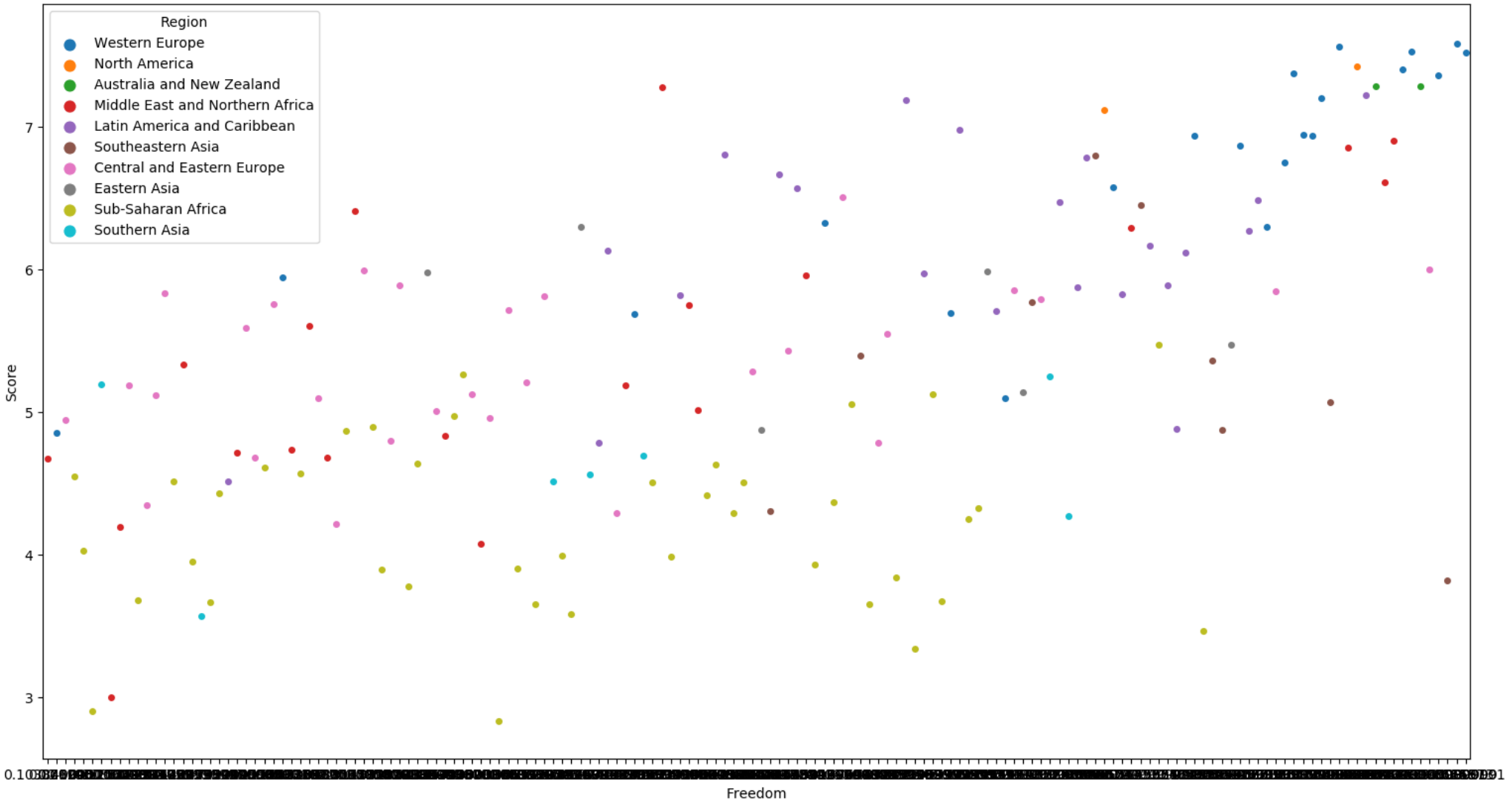
The same situation than before. Quite linear correlation between numerical values and graph seems that there is no strongly effect on '**Score'** caused by '**Region'**

Relation between **Health** and **Score**



Weak correlation. We can see larger dispersion than before but still it may be linear correlation. It should be useful value in final model.

Relation between **Freedom** and **Score**



Practical no correlation. The largest dispersion between dots to this time. It may be caused by difficulties in reckoning value of freedom. Probably it would not be useful in final model.

I have made some research among internet. I have found few more parameters which may describe happiness level:

- Purchasing power
- Possibilities of self-development
- Stability

I think I should add new column: **Maslow Ratio** -> level from Maslow pyramid for each country. It should describe country's level of stride. Unfortunately I have not found data like this.

2. Taking care of missing values:

Fortunately there is no missing values.

3. Adding and removing columns:

I am making predictions for basic dataset.

4. Getting dummies:

There is only one categorical column which I will transform into dummies -> **'Region'**

There is no reason why use **'Country'** value in model (each country has a different name).

5. Fitting classifiers:

5.1 Checking scores for basic data

I will split dataset into train, validation and test with **test_size = 0.2** for test set

I will use two different algorithms: **GradientBoostingRegressor** and **LinearRegression.** After first fitting I will check the scores and choose the best. Next I will use RFE to choose most important features, next I will use GridSearchCV to get best parameters. At the end I will check accuracy on a test set.

Metrics: **nrse** -> used in features selection as a negative mean squared error
Root mean squared error (RMSE)-> main metric of accuracy

In first fitting I have used completely basic classifiers, and scores are:

	RMSE
LinearRegression	**0,0002**
GradientBoostingRegressor	**0,0333**

It is clear to see that better scores are with **LinearRegression** so I will take it into next steps.

5.2 Find the most important features

To find the most important features I will use RFE method. I will take 20 iteration of calculations for different numbers of variables (from 5 to 114). Each calculation will be made with using cross validation (splitting dataset into 10 subsets). I will measure negative mean squared error and take result which is the closest to 0.

Results:

	5	6	7	8	9	10	11	12	13	14
nrse	**-0,0606**	**-0,0140**	**-8,0348 *10^-8**	**-7,6947 *10^-8**	**-7,8973 *10^-8**	**-8,2738 *10^-8**	**-8,2814 *10^-8**	**-8,3003 *10^-8**	**-8,3956 *10^-8**	**-8,4616 *10^-8**

We will get the best results if we take 8 best values. And these are:

- **'Economy'**
- **'Family'**
- **'Health'**
- **'Freedom'**
- **'Trust'**
- **'Generosity'**
- **'DystopiaResidual'**
- **'SouthernAsia'**

As we can see, my variables practical each basic value is in most important group. Another interesting thing is **nrse** value. It is very low, I think that the **Happiness Score** was created as a result of linear equation using all basic variables.

6. Checking scores for final model:

For last classifier I will use parameters form **5.2** to predict test values.

After fitting classifier I have got results:

	RMSE
XGBClassifier	**0,0003**

7. Evaluation:

In my opinion I have been working on splendid data. This was complete, looked-after dataset with no NaN values. That was a pleasure to make this project. I think, that I have found algorithm which was used do count **Happiness Score**.

I have reached my aim but I am not satisfied. I think if we want to count the happiness score of each person, we should done research based on human psychology and feelings. This is great topic and I would like to expand it in bigger project.