Trendlines

Examples

We are able to assign each user a certain score about *Interest in Sport*.

Possibly, we compute this score using data about his/her navigation style. Say, the score is the ratio

$$Interest\ in\ Sport = \frac{number\ of\ accesses\ to\ pages\ about\ sport}{number\ of\ accesses\ to\ pages\ of\ every\ kind}$$

Defined in this way, the score is a number in the range (0, 1). With some manipulation, we ensure it is not 0, nor 1. We know, extremum values are to be avoided.

Possibly, we compute the score with some more sophisticated algorithm, e.g. Naïve Bayes, Logistic Regression or many others.

Possibly, we purchase a database where this user has a pre-assigned score.

Interest in Sport	Spending in €
0,31	242
0,35	260
0,44	310
0,46	374
0,50	348
0,52	392
0,55	380
0,59	396
0,61	312
0,64	408
0,67	384
0,70	436
0,71	448
0,72	406
0,78	418
0,87	452

This is a sample of 16 users.

We want to generate an equation of the form

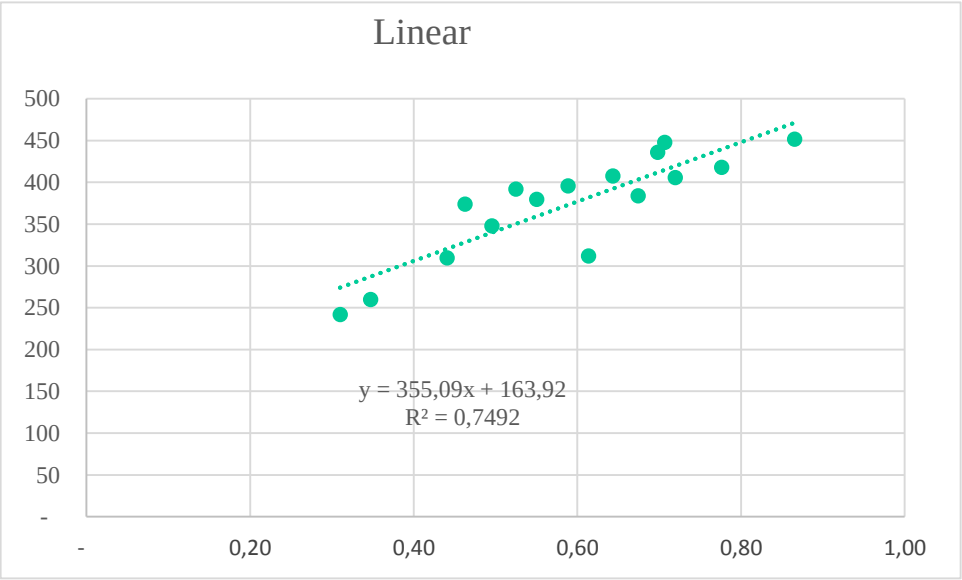
$$\hat{y} = f(x)$$

where $\hat{y}$ is the forecasted spending, *x* is the user's *Sport* score and *f* is a function.

It is up to us (the data scientists) to choose a functional form for *f*. In principle, any kind of functional form can be used. In daily practice, a few kinds cover normal needs.

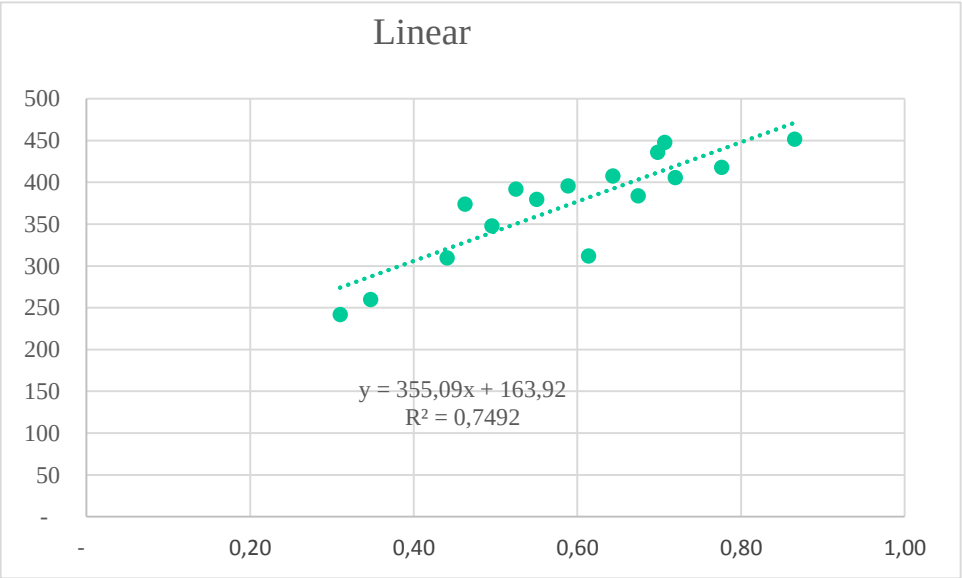
The most common functional structures for the forecasting function *f* are: linear, polynomial, power, logarithmic, exponential.
Applying *f* to the data table we obtain a *trendline*, i.e. a curve describing a supposed *baseline tendency* of the phenomenon to be predicted (here *Spending*).

Interest in Sport	Spending in €
0,31	242
0,35	260
0,44	310
0,46	374
0,50	348
0,52	392
0,55	380
0,59	396
0,61	312
0,64	408
0,67	384
0,70	436
0,71	448
0,72	406
0,78	418
0,87	452



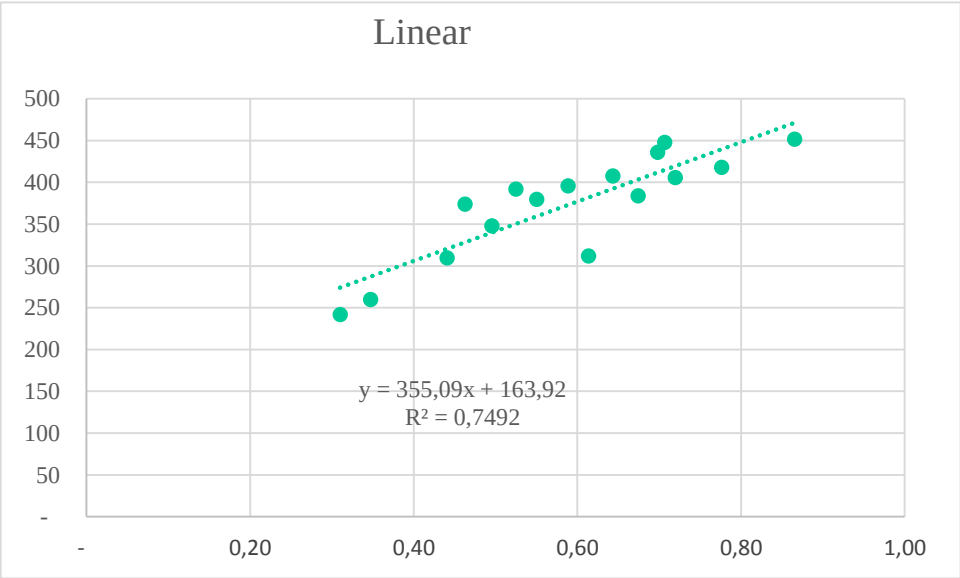
We assume that observed spending amounts are caused by the predictor *Interest in Sport* together with some *error* or *noise*.
The mathematical model is $\hat{y} = f(x) + \varepsilon$, i.e. the answer is the effect of the predictor plus a noise (noise is whichever factor not included in the model).
If an user with score 0.68 enters our site, we predict his/her spending will be $\hat{y} = 355.99 \times Score + 163{,}92 = 405.99$

Interest in Sport	Spending in €
0,31	242
0,35	260
0,44	310
0,46	374
0,50	348
0,52	392
0,55	380
0,59	396
0,61	312
0,64	408
0,67	384
0,70	436
0,71	448
0,72	406
0,78	418
0,87	452



The predicted spending for users with score 0.68 is 406€.
This is the actual spending of an observed user with score 0.72.
This can sound strange, but observe the two users with 0.67 and 0.70 scores. You will note that 406€ is a reasonable *interpolation* of their spending, i.e. 384€ and 436€.
We see that the trend line approximates empirical data, but not exactly. It extracts some *inner logic* out of the data table.

Interest in Sport	Spending in €
0,31	242
0,35	260
0,44	310
0,46	374
0,50	348
0,52	392
0,55	380
0,59	396
0,61	312
0,64	408
0,67	384
0,70	436
0,71	448
0,72	406
0,78	418
0,87	452



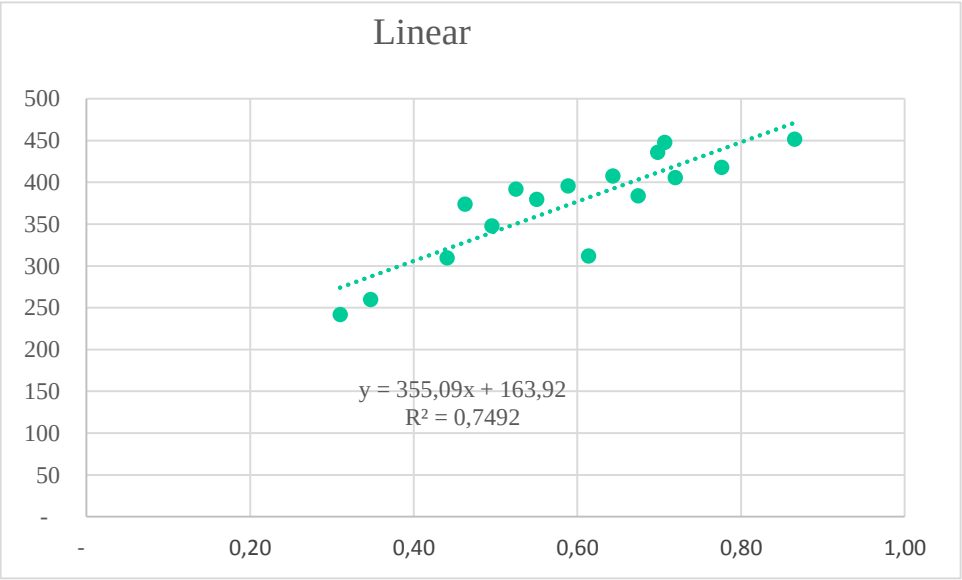
This model is *linear*, i.e. with form

$$\hat{y} = ax + b$$

It is up to a data analysis tool (Microsoft Excel® here) to generate the *parameters* of the model, here a and b.

The parameters are *optimal*, in the sense the resulting equation forecasts past observed spending in the best possible way *given the linear functional form*.

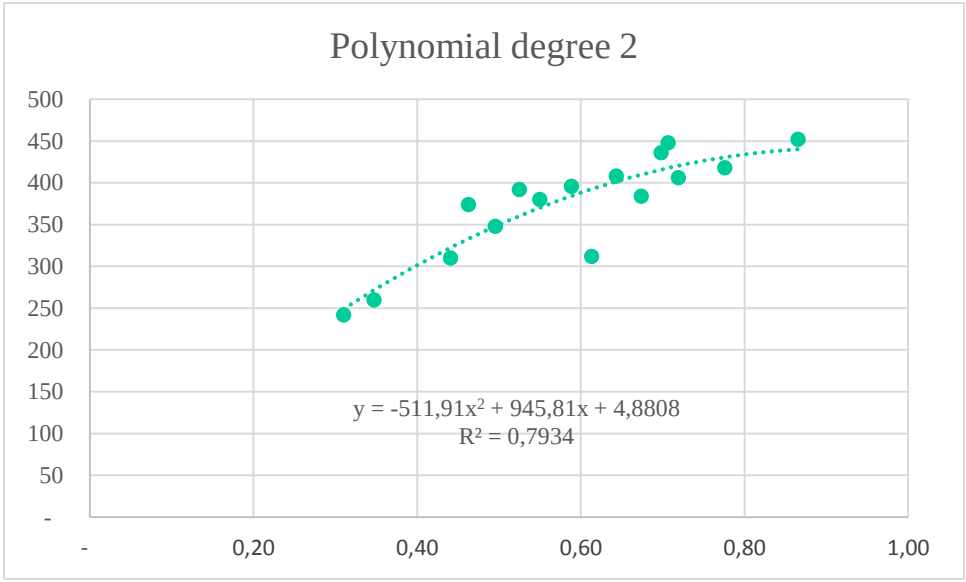
Interest in Sport	Spending in €
0,31	242
0,35	260
0,44	310
0,46	374
0,50	348
0,52	392
0,55	380
0,59	396
0,61	312
0,64	408
0,67	384
0,70	436
0,71	448
0,72	406
0,78	418
0,87	452



The number R^2 is named *linear determinant*. It measures the goodness of the trendline. You can think of it as the percentage of error you save when using this equation as predictor instead of the naïve predictor "the average".
The error is measured as the sum of squared differences between empirical values y_i and corresponding forecasted values $\hat{y}_i$.

See the Excel worksheet in lecture notes for details.

Interest in Sport	Spending in €
0,31	242
0,35	260
0,44	310
0,46	374
0,50	348
0,52	392
0,55	380
0,59	396
0,61	312
0,64	408
0,67	384
0,70	436
0,71	448
0,72	406
0,78	418
0,87	452



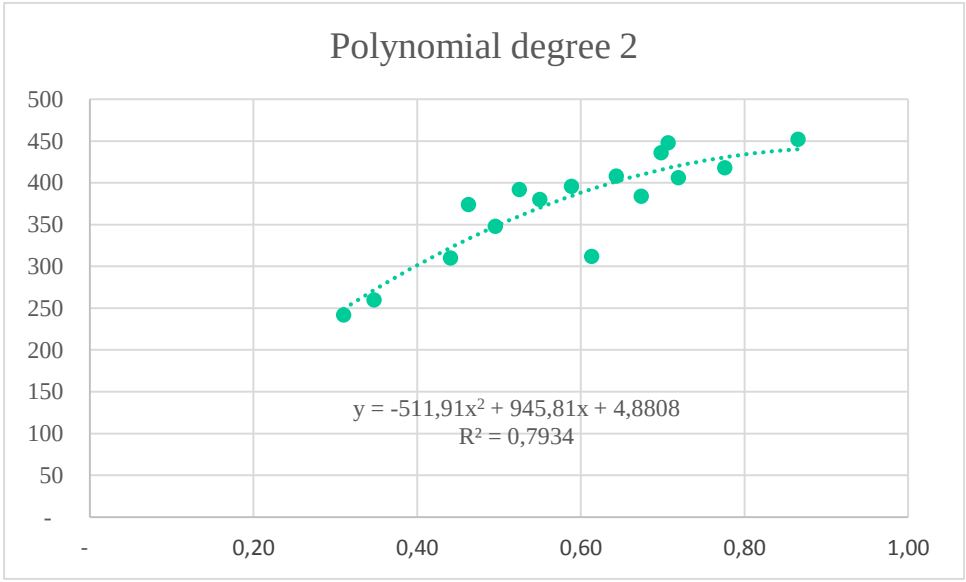
Here we have a *quadratic* (i.e. second degree polynomial) equation, i.e. with form

$$\hat{y} = ax^2 + bx + c$$

In this case, Excel finds the best three parameters which make the equation to be the best fitting one among the infinite possible second degree equations.

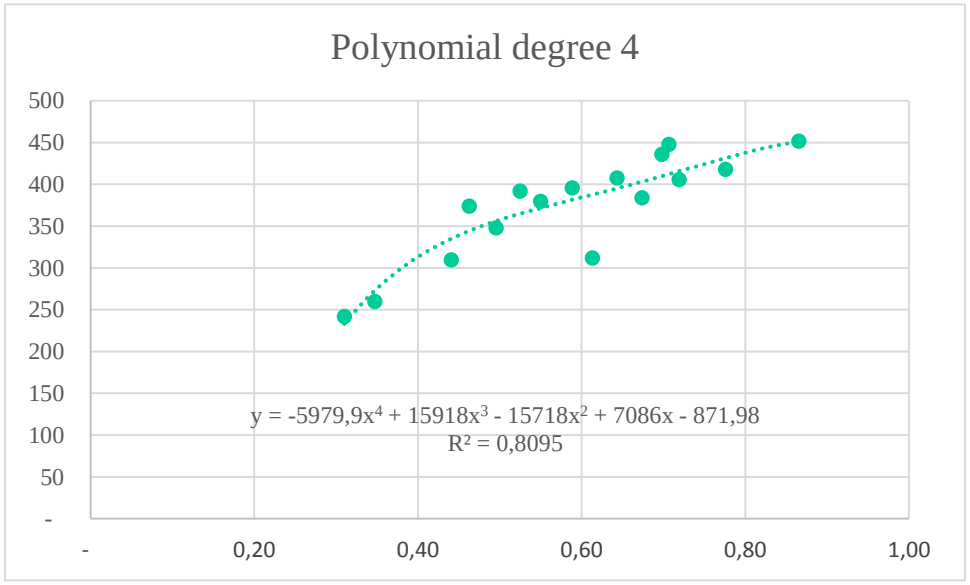
When encountering a new user, fill his/her score into *x* and you have a prediction for the spending.

Interest in Sport	Spending in €
0,31	242
0,35	260
0,44	310
0,46	374
0,50	348
0,52	392
0,55	380
0,59	396
0,61	312
0,64	408
0,67	384
0,70	436
0,71	448
0,72	406
0,78	418
0,87	452



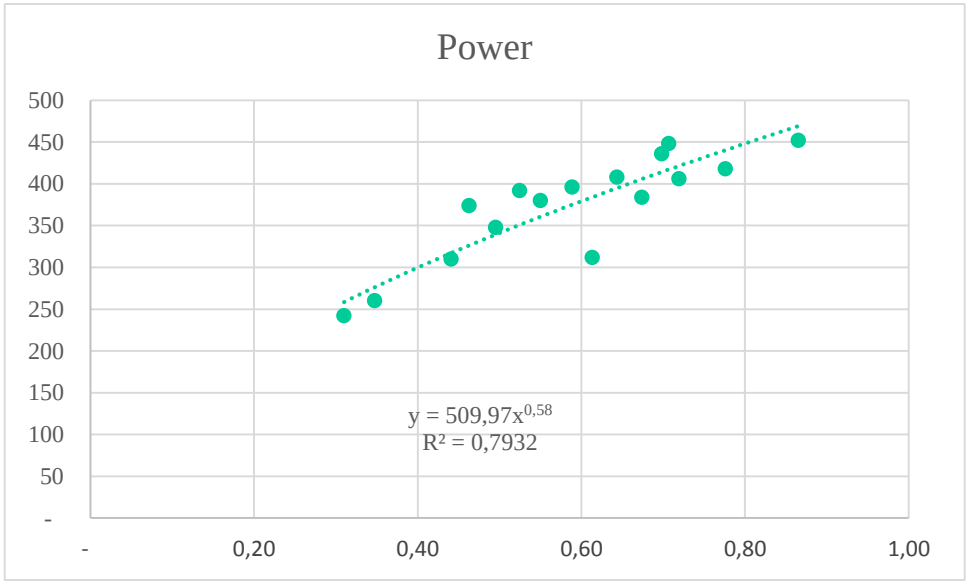
The linear determinant R^2 is better than for the linear model.
The "absorbed error" in comparison with the naïve predictor "the next user spending will be equal to the average in the past" is now 79% versus 75%.
The squared error with this trendline is 21% of the naïve prediction error (25% in the linear model).

Interest in Sport	Spending in €
0,31	242
0,35	260
0,44	310
0,46	374
0,50	348
0,52	392
0,55	380
0,59	396
0,61	312
0,64	408
0,67	384
0,70	436
0,71	448
0,72	406
0,78	418
0,87	452



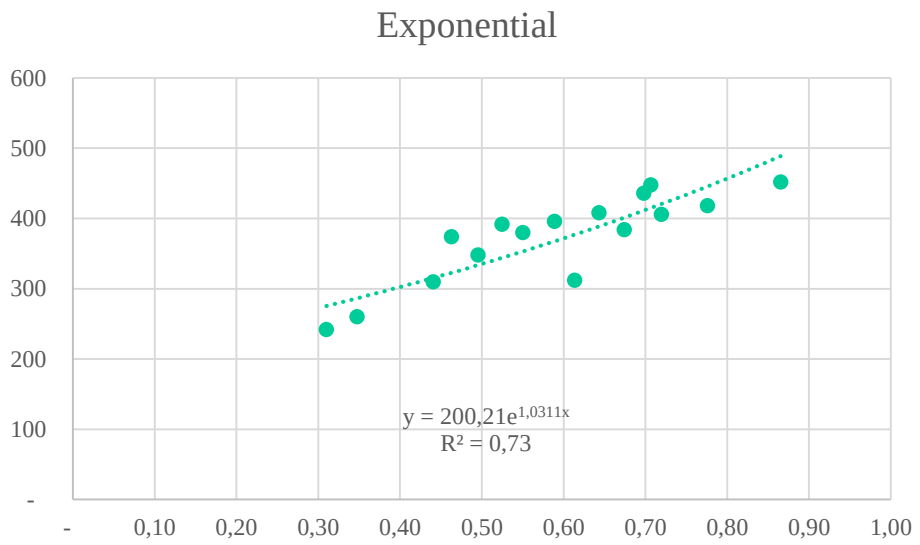
The *quartic* model, i.e. the polynomial model of degree 4, fits data points better than the second degree model, which in turn was better than the first degree model (the linear model is indeed the polynomial model of degree 1).
Now the R-squared is 81%, an improvement on 79%.
Is this improvement for free?
Unluckily, it is not. We will see why when discussing overfitting.

Interest in Sport	Spending in €
0,31	242
0,35	260
0,44	310
0,46	374
0,50	348
0,52	392
0,55	380
0,59	396
0,61	312
0,64	408
0,67	384
0,70	436
0,71	448
0,72	406
0,78	418
0,87	452



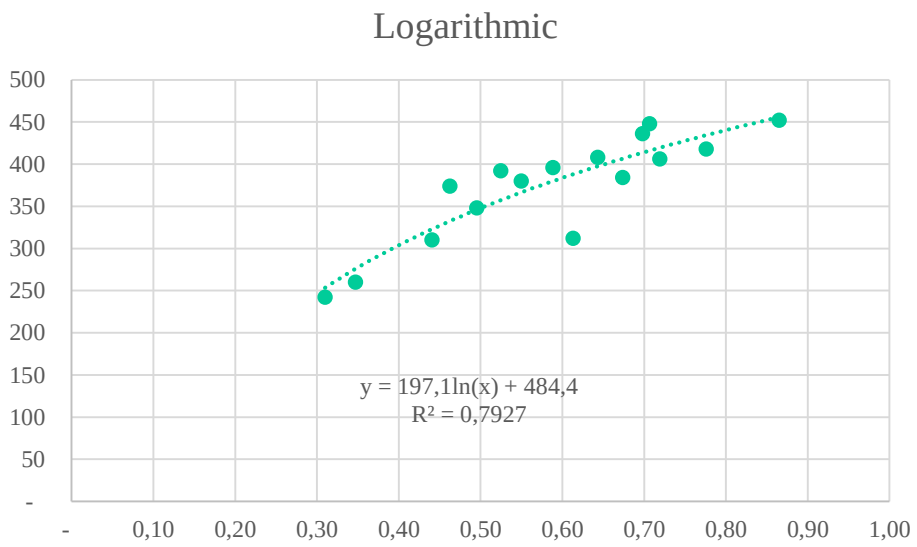
The *power* model represent *y* as *x* raised at a certain power, and multiplied for a certain coefficient. So, it has two parameters determined by the data analysis tool.
The R-squared is close to 79%, similar to the quadratic model.

Interest in Sport	Spending in €
0,31	242
0,35	260
0,44	310
0,46	374
0,50	348
0,52	392
0,55	380
0,59	396
0,61	312
0,64	408
0,67	384
0,70	436
0,71	448
0,72	406
0,78	418
0,87	452



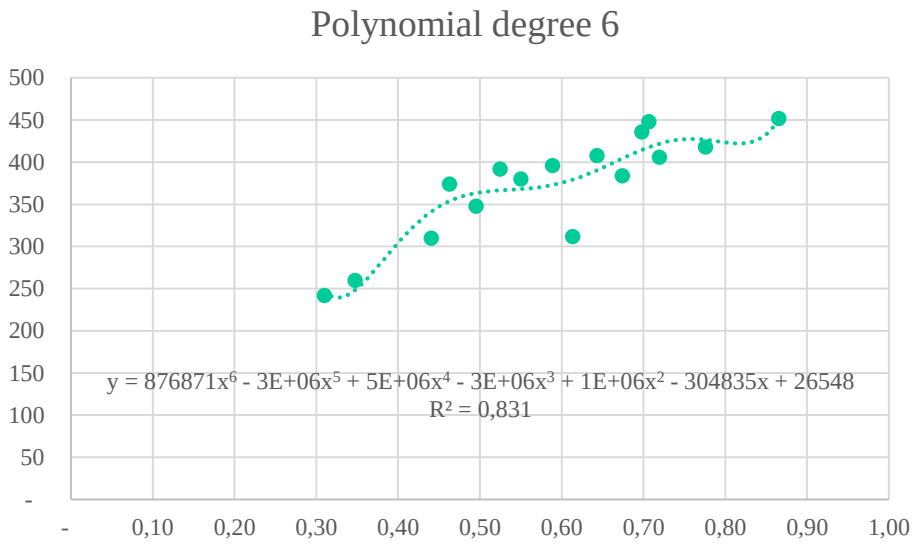
The *exponential model* is not particularly effective with this data table. R-squared is 73%.

Interest in Sport	Spending in €
0,31	242
0,35	260
0,44	310
0,46	374
0,50	348
0,52	392
0,55	380
0,59	396
0,61	312
0,64	408
0,67	384
0,70	436
0,71	448
0,72	406
0,78	418
0,87	452



The *logarithmic* model fits better than the exponential one. R-squared is 79%.
This suggests that when the score (the variable *x*) raises, the spending (the variable *y*) responds in a decreasing manner.
(Of course, this toy sample is too little to really support this kind of statements. Let us use it as a stimulus for intuition).

Interest in Sport	Spending in €
0,31	242
0,35	260
0,44	310
0,46	374
0,50	348
0,52	392
0,55	380
0,59	396
0,61	312
0,64	408
0,67	384
0,70	436
0,71	448
0,72	406
0,78	418
0,87	452



The polynomial model of degree 6fits better than others already seen (R-squared 83%), though the difference is not huge.
Note how the higher degree allows the curve to "adapt" to data with more inflection points.

Overfitting

We saw examples of data approximation with curves expressed by several equational forms.

One can easily think that it is possible to find an equation *perfectly* approximating data points and it is a good thing.

Really, it is possible but not so good.

If you have *n* data points you can always build a polynomial curve of degree *n*, named *interpolation polynomial*, that perfectly reproduces data points. This is very useful for other applications, but not in forecasting.

Indeed, the interpolation polynomial perfectly predicts the past: yet, we want to predict the future!

Predicting the past is how we *learn to predict the future*. If our equation fits well to past data points, hopefully it will work well with future data points.

It is not possible to treat overfitting here, though it is an extremely important topic in data science.

The reader is invited to remember that we need models which are both *accurate* (i.e. fitting well with data) and *simple* (i.e. with a small number of parameters).

Unfortunately, in general we have to cope with a trade-off: few parameters give us simpler but less accurate models, many parameters gives us high accuracy at the price of increasing complexity.

Though, we run a risk: *learning too much and too well what happened.*

If we learn perfectly the sample observed in the past, we can easily learn something which is a characteristic of the sample, not of the population the sample is taken off.

Therefore, we will have a too strong bias (prior assumptions) when predicting future cases (i.e. future users with certain scores for interest in sport).

We want to learn the inner logic of data, not data itself.

This means we want to compress observed data and get an equation which reproduces it well, not perfectly. We want to get the *signal*, not the *noise*.

Indeed, the interpolation polynomial of degree 16 is able to reproduce our 16 past cases just because it has exactly 16 parameters: it is simply the data table expressed in another way! For this reason it is not useful to predict the future.