Problem 2: Least squares, curve-fitting, residuals, and variances

Suppose someone had presented you a x, y – dataset that looks like Figure 1, and your only job was to create a least-squares polynomial fit for $y(x)$.

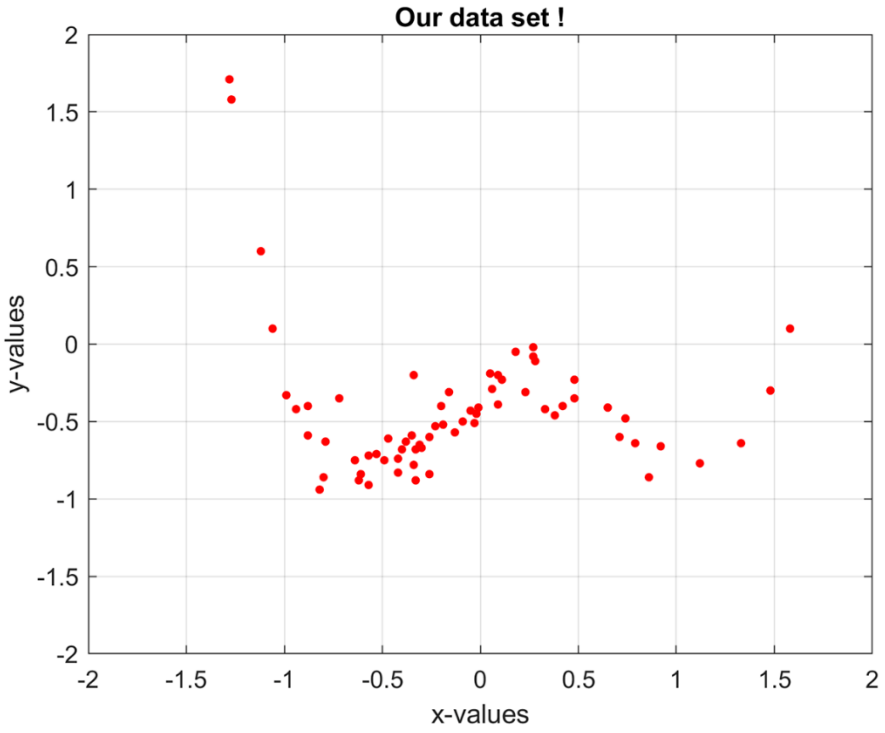

Figure 1: Your job is to fit the y-values as polynomial of the x-values !

Your homework tasks: (turn in the parts highlighted in yellow… plots and/or diary echoes)

Part 1: Choosing the right polynomial model

1) Using your intuition, which of the following polynomial models would you choose for your least-squares fitting routine ? Using the *disp* function in matlab, echo your choice in the diary file.

$$Linear: \quad y = c_0 + c_1 x$$

$$Quadratic: \quad y = c_0 + c_1 x + c_2 x^2$$

$$Cubic: \quad y = c_0 + c_1 x + c_2 x^2 + c_3 x^3$$

$$Quartic: \quad y = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4$$

Part 2: Find the least-squares coefficients for your model

2 columns of floating-point data in our file

One line of annotations to ignore

1) Load the raw data using the matlab command:

```
[x, y] = textread('Problem2_polynomial_data.txt', '%f%f', 'headerlines', 1)
```

2) Using the least-squares formula:

$$X^T X\ c\ =\ X^T\ y$$

find the least-squares coefficients $c_0, c_1, \cdots, etc$ and echo them in your diary

3) Using matlab's *plot* function, plot the raw data points "y" as small circular dots.

4) Overlay the previous graph by plotting the least-squares approximation curve as a line with a different color

5) Label the plots properly (add proper legends to the figure)

Part 3: Analyzing your residuals (data deviations from your model curve)

1) Using your model curve and the 70 raw data points, calculate the residual vector $\vec{r}$.

Note: You don't have to echo this vector in your diary.
Hint: You should be able to calculate this using 1 line of code in matlab !! Think about it before you type it… =)

2) Now, calculate and echo the sum of the $(residuals)^2$ for your 70 data points, where:

$$|\vec{r}|^2\ =\ \sum_{i=1}^{70} {r_i}^2$$

More tasks on the next page

3) Then, using matlab's *histogram* function (it's a really easy function to use… Google it ! =)) , create a histogram of the residual values stored within vector $\vec{r}$.

- For the calculations, set your histogram bin edges to be = [-1 : 0.025 : 1]
- Set the x −axis limits of your histogram plot to be from -1 to +1
- Set the y −axis limits of your histogram plot (# of occurrences) be from 0 to 10

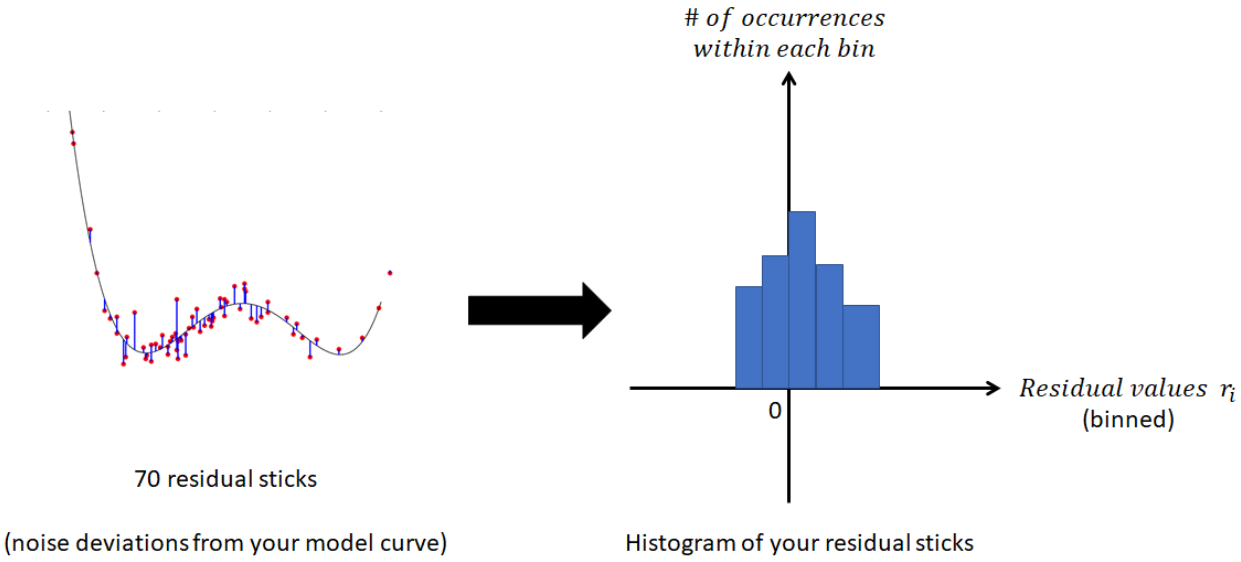

Figure 2: Your residual histogram plot may look something like this !! =)

4) You will see that the noise of your original data set is shaped like a normal-distributed (Gaussian) bell curve ! Using matlab, the 2 big questions we would like to ask are:

- What is the *mean* μ (average) of the residuals ?
- What is the *sample variance* σ^2 of the residuals (use the "$N-1$" formula in the web link below)
- Multiply your variance by $(N-1)$ and echo your answer. You should see a cool result !

http://www.visiondummy.com/2014/03/divide-variance-n-1/

Moral of the story

$$|\vec{r}|^2 = \sum_{i=1}^{N} r_i^2 = \sum_{i=1}^{N} (r_i - \mu)^2 = (N-1)\cdot \sigma^2 \Big|_{with\ \mu=0}$$

$Our\ least-squares\ (residual)^2\ sum$

$Is\ direct\ measure\ of$

$(if\ zero\ mean-Gaussian)$

$The\ variance\ of\ our\ data\ noise\ !!!!$