L24: Least-Squares Linear Regression

Discussion and Applications; Lab 09; Other Topics

Lucas A. J. Bastien

E7 Spring 2017, University of California at Berkeley

March 17, 2017

Version: release

Announcements

Lab 09 is due on March 24 at 12 pm (noon)

Today:

- Least-squares linear regression: discussion and applications
- Introduction to lab 09
- E7 programming project
  - A first glance
  - Forming teams

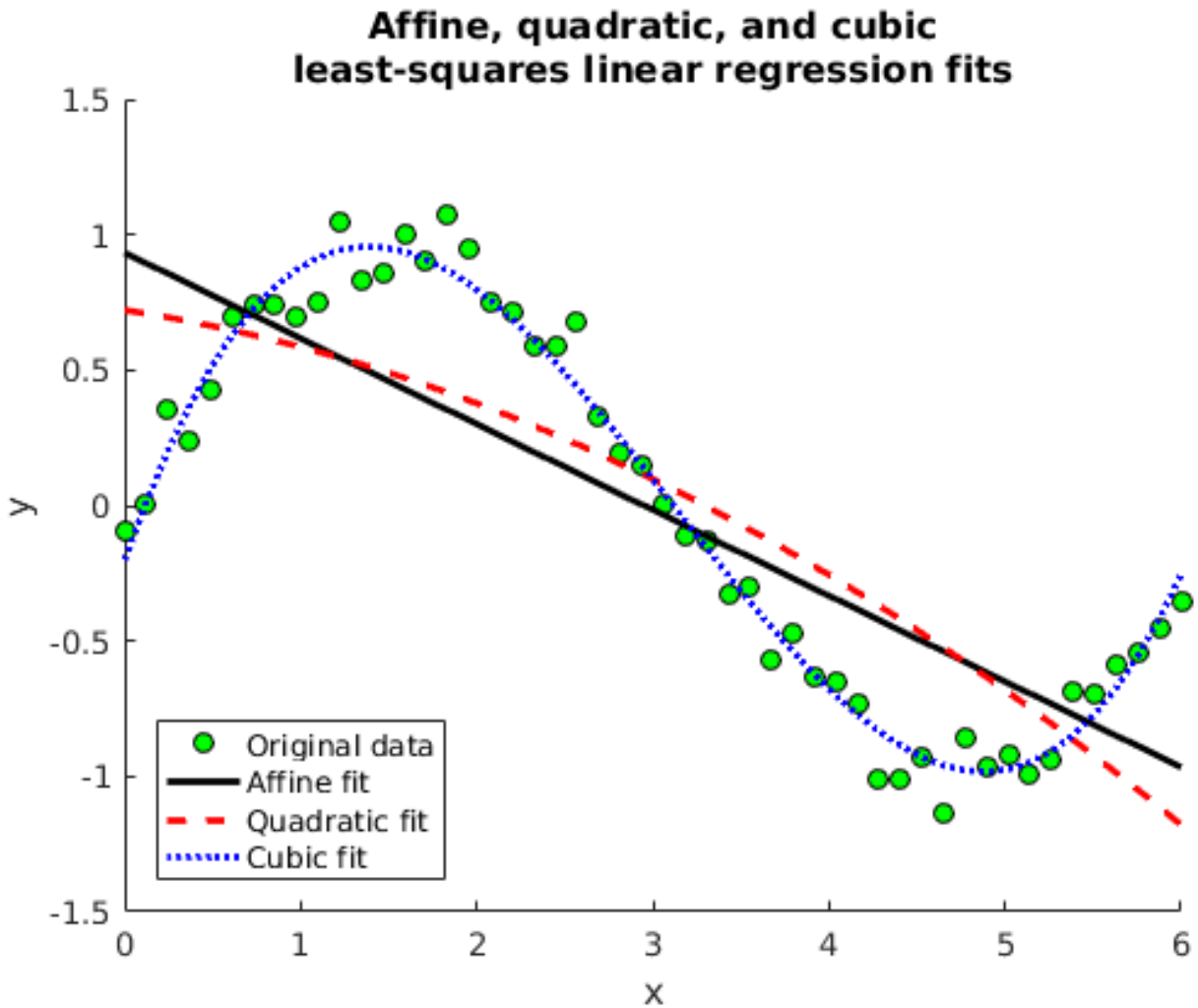
Next week:

- Monday: Interpolation (chapter 14)
- Wednesday: Series (chapter 15)
- Friday: Discussion
- Wednesday or Friday: presentation of the final programming project

One more example of least-squares linear regression

See function `my_multiple_regressions`

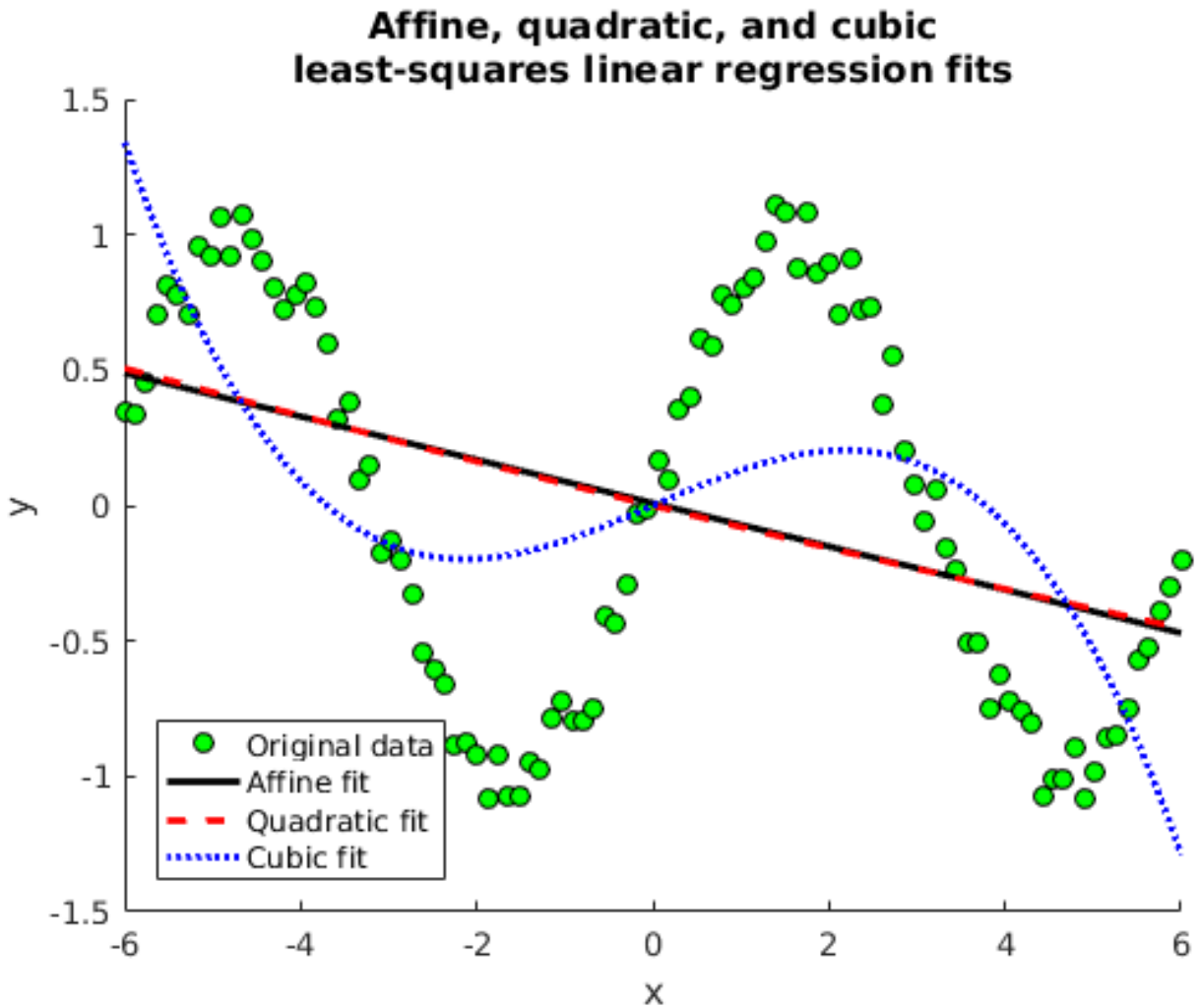
Here: the cubic line fits the original data nicely, but the other lines do not



One more example of least-squares linear regression

See function `my_multiple_regressions`

Here: none of the lines fit the original data nicely



“Unfittable” data

Sometimes, data just cannot be fitted in a way that makes sense...

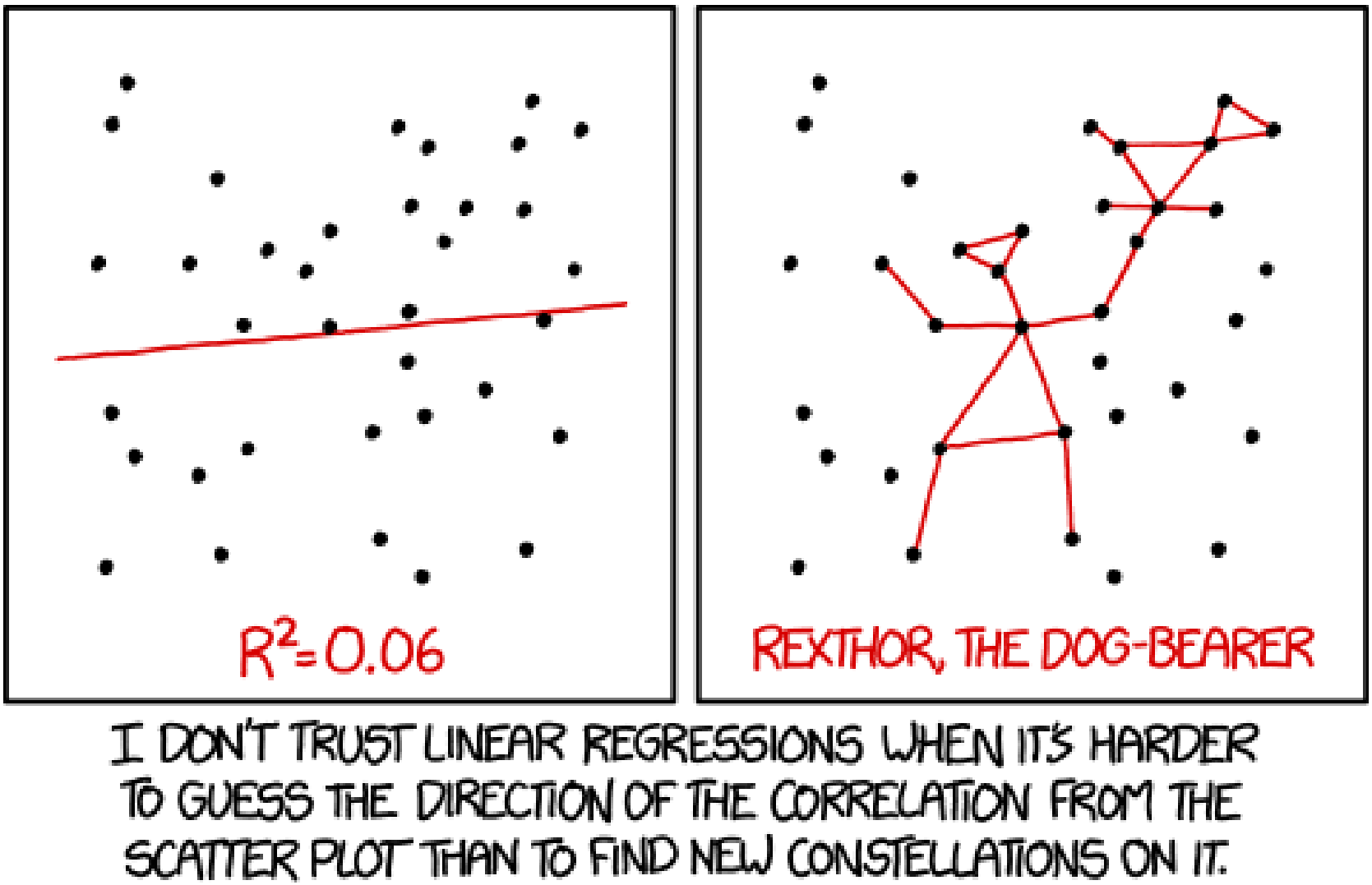


Image retrieved from https://xkcd.com/1725/ on March 17$^{\text{th}}$ 2017

Introduction to lab 09

Topic:

Least-squares linear regression

Introduction to lab 09

Topic:

Least-squares linear regression

Before working on the lab:

Review the material from lectures L22 and L23

Introduction to lab 09

Topic:

Least-squares linear regression

Before working on the lab:

Review the material from lectures L22 and L23

As you work on the lab, think about what the results mean

Look at the figures in the instructions

(note: your functions should **not** create figures)

Q3: Fitting with sines and cosines

Objective: given a set of x- and y-data, fit a function of the form:

$$f(x) = k + \sum_{i=1}^{n} a_i \sin(ix) + \sum_{i=1}^{n} b_i \cos(ix)$$

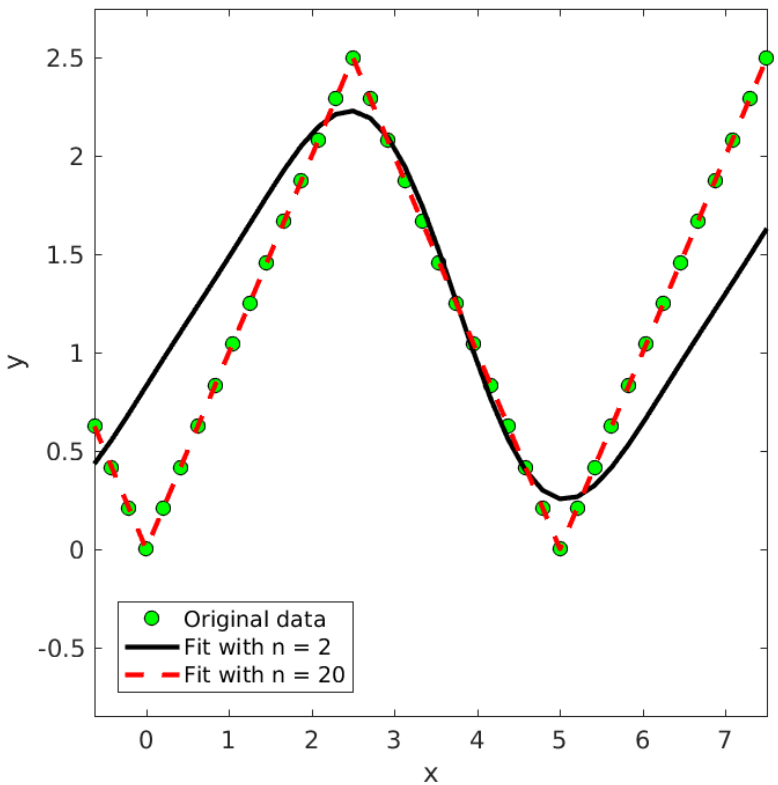
where n is an integer that is greater than zero, and where the coefficients to fit are k, $a_i, i = \{1, 2, \ldots, n\}$, and $b_i, i = \{1, 2, \ldots, n\}$

Q3: Fitting with sines and cosines

Objective: given a set of x- and y-data, fit a function of the form:

$$f(x) = k + \sum_{i=1}^{n} a_i \sin(ix) + \sum_{i=1}^{n} b_i \cos(ix)$$

where n is an integer that is greater than zero, and where the coefficients to fit are k, $a_i, i = \{1, 2, \ldots, n\}$, and $b_i, i = \{1, 2, \ldots, n\}$

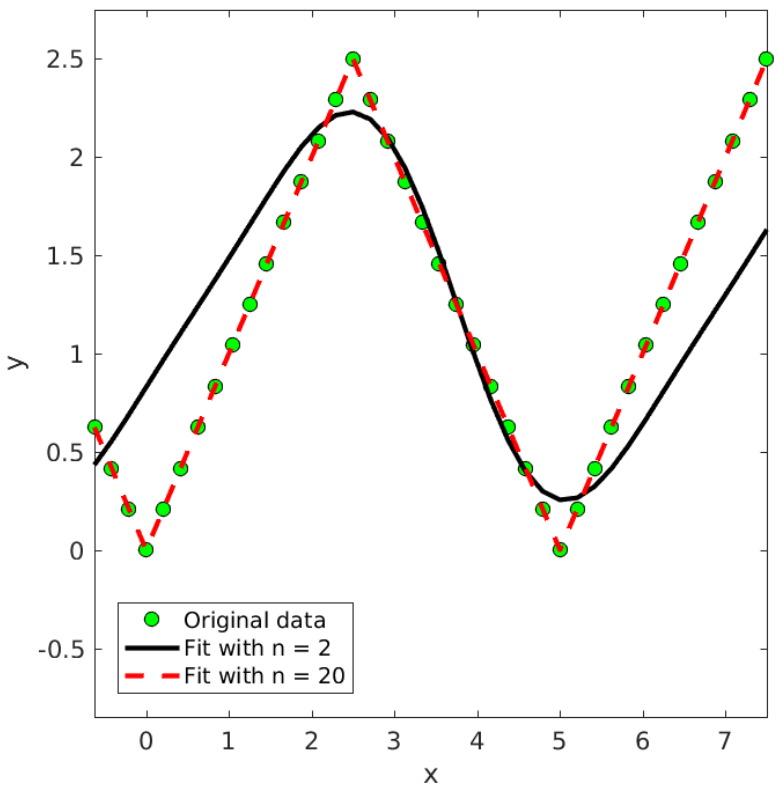


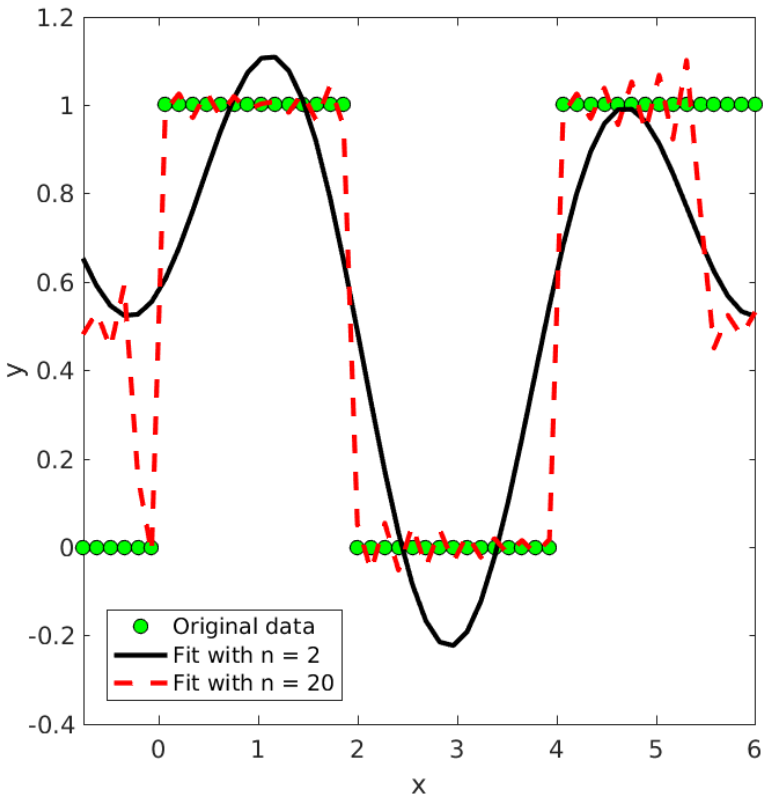
Q3: Fitting with sines and cosines

Objective: given a set of x- and y-data, fit a function of the form:

$$f(x) = k + \sum_{i=1}^{n} a_i \sin(ix) + \sum_{i=1}^{n} b_i \cos(ix)$$

where n is an integer that is greater than zero, and where the coefficients to fit are k, $a_i, i = \{1, 2, \ldots, n\}$, and $b_i, i = \{1, 2, \ldots, n\}$





Q5: Fitting polynomials

A polynomial P of degree p is a function of the form (with $a_p \neq 0$):

$$P(x) = a_p x^p + a_{p-1} x^{p-1} + \cdots + a_1 x + a_0$$

Q5: Fitting polynomials

A polynomial P of degree p is a function of the form (with $a_p \neq 0$):

$$P(x) = a_p x^p + a_{p-1} x^{p-1} + \cdots + a_1 x + a_0$$

Objective: fit a polynomial of degree p such that (see lab 09 for details):

- `degree_min` $\leqslant$ `p` $\leqslant$ `degree_max`
- p is reasonably small
  (we don't want to make it bigger than "necessary")
- The square error is small

Q5: Fitting polynomials

A polynomial P of degree p is a function of the form (with $a_p \neq 0$):

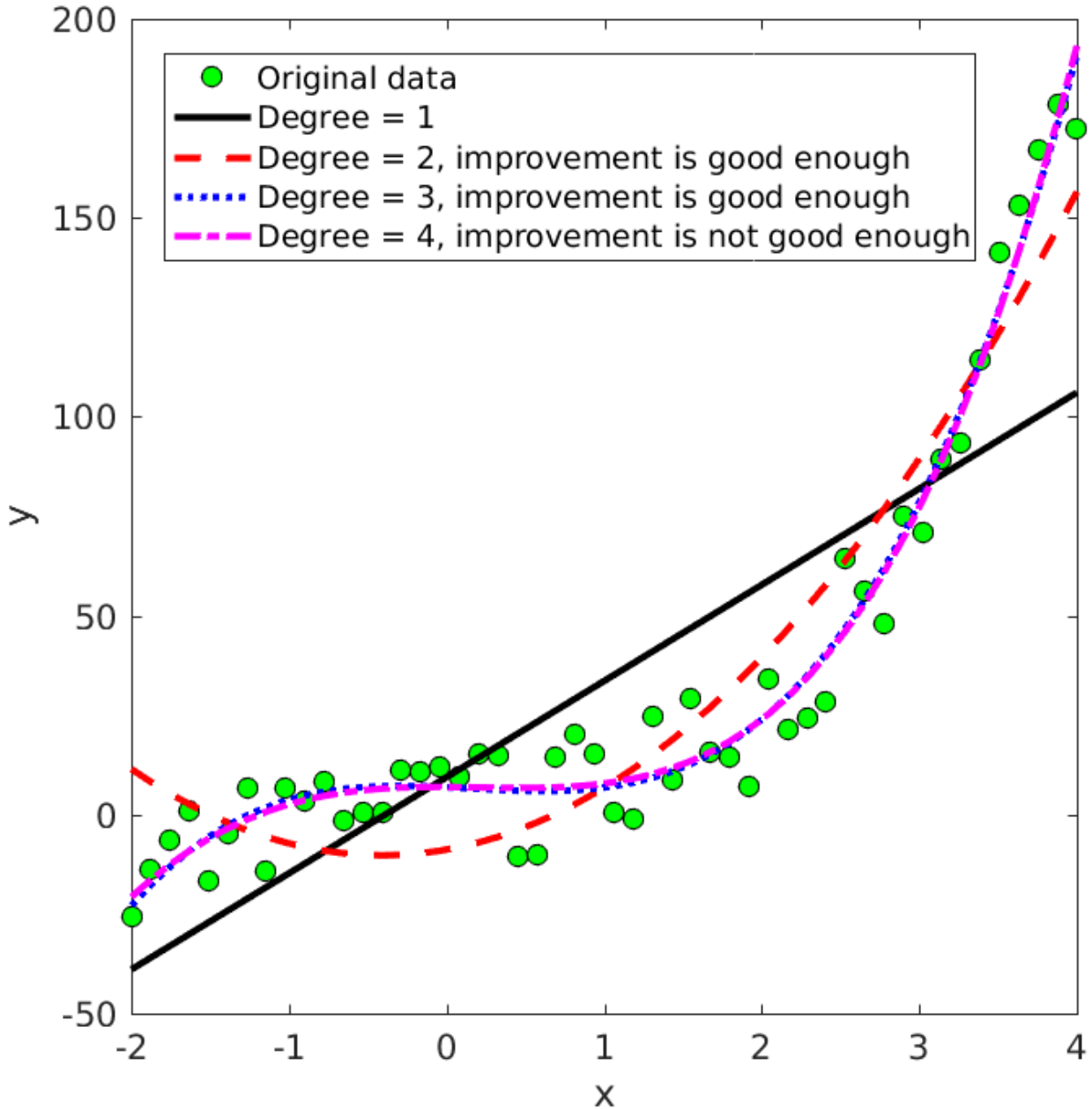
$$P(x) = a_p x^p + a_{p-1} x^{p-1} + \cdots + a_1 x + a_0$$

Objective: fit a polynomial of degree p such that (see lab 09 for details):

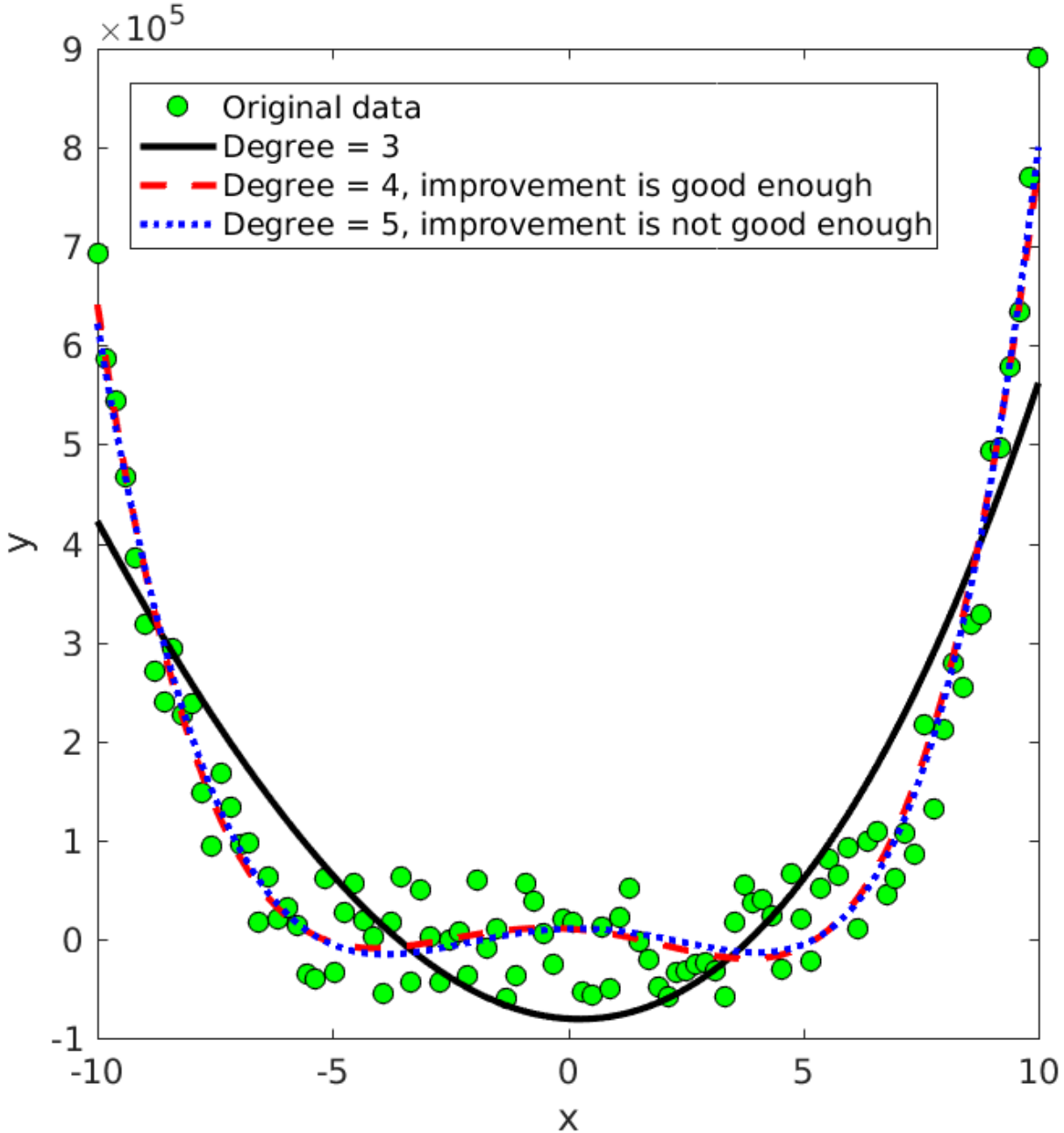
- `degree_min` $\leqslant$ `p` $\leqslant$ `degree_max`
- p is reasonably small
  (we don't want to make it bigger than "necessary")
- The square error is small

Note:
Fitting polynomial of higher and higher degrees reduces the square error

Q5: Fitting polynomials



Q5: Fitting polynomials



Q6: Fitting $y = ax$ for different error definitions

Consider a set of x- and y-data through which you want to fit a line of equation $y = ax$ (the coefficient to fit is a)

Q6: Fitting $y = ax$ for different error definitions

Consider a set of x- and y-data through which you want to fit a line of equation $y = ax$ (the coefficient to fit is a)

Least-squares regression: minimize the square error E_2:

$$E_2(a) = \sum_{i=1}^{m} (ax_i - y_i)^2$$

Q6: Fitting $y = ax$ for different error definitions

Consider a set of x- and y-data through which you want to fit a line of equation $y = ax$ (the coefficient to fit is a)

Least-squares regression: minimize the square error E_2:

$$E_2(a) = \sum_{i=1}^{m} (ax_i - y_i)^2$$

Objective:

In this question, fit a such that $y = ax$ minimizes the "p-error" E_p:

$$E_p(a) = \sum_{i=1}^{m} (ax_i - y_i)^p \qquad (p \text{ is positive and even})$$

Q6: Fitting $y = ax$ for different error definitions

Consider a set of x- and y-data through which you want to fit a line of equation $y = ax$ (the coefficient to fit is a)

Least-squares regression: minimize the square error E_2:

$$E_2(a) = \sum_{i=1}^{m} (ax_i - y_i)^2$$

Objective:

In this question, fit a such that $y = ax$ minimizes the "p-error" E_p:

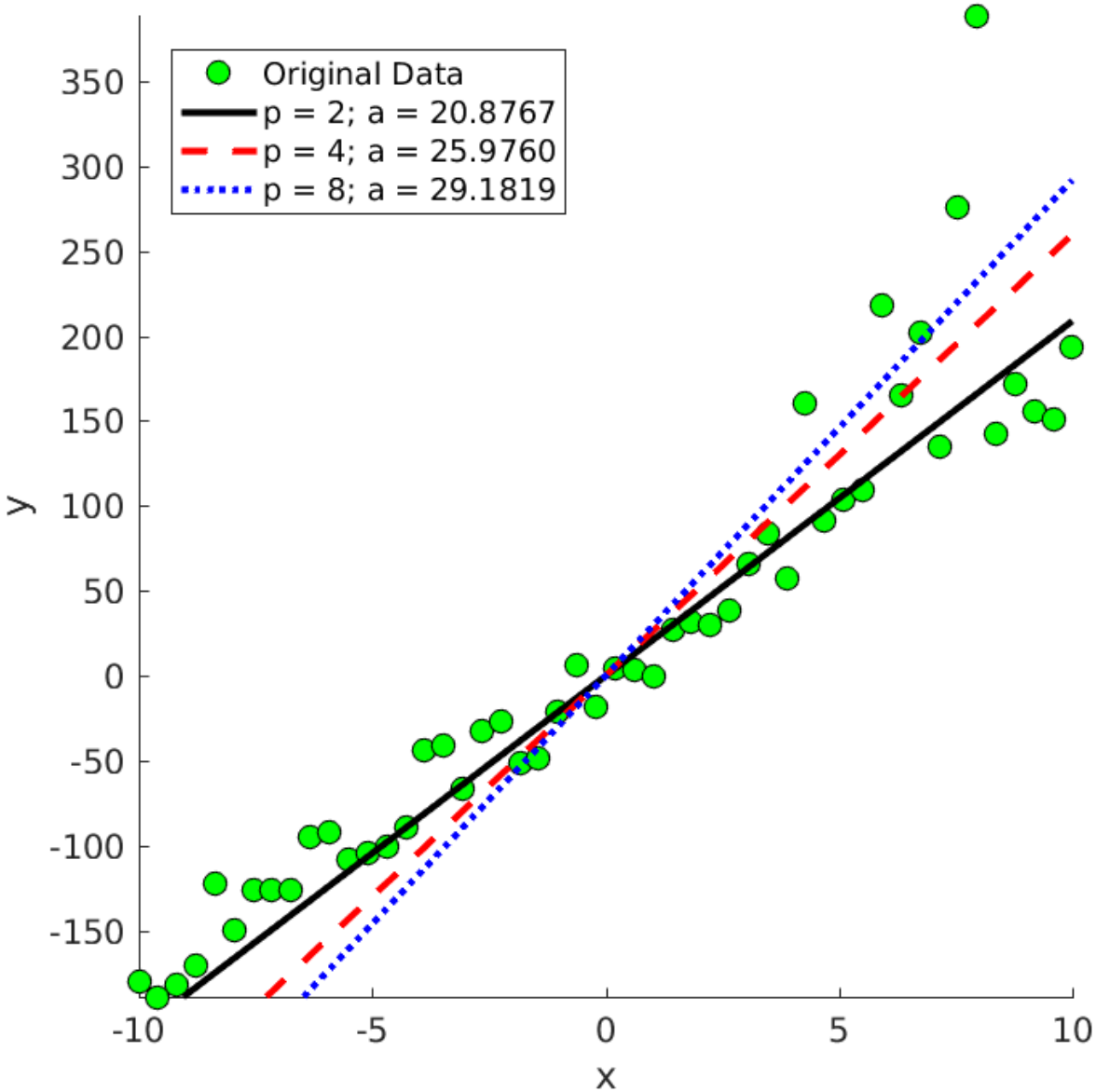
$$E_p(a) = \sum_{i=1}^{m} (ax_i - y_i)^p \qquad (p \text{ is positive and even})$$

by finding a such that:

$$E_p'(a) = \sum_{i=1}^{m} p x_i (ax_i - y_i)^{p-1} = 0$$

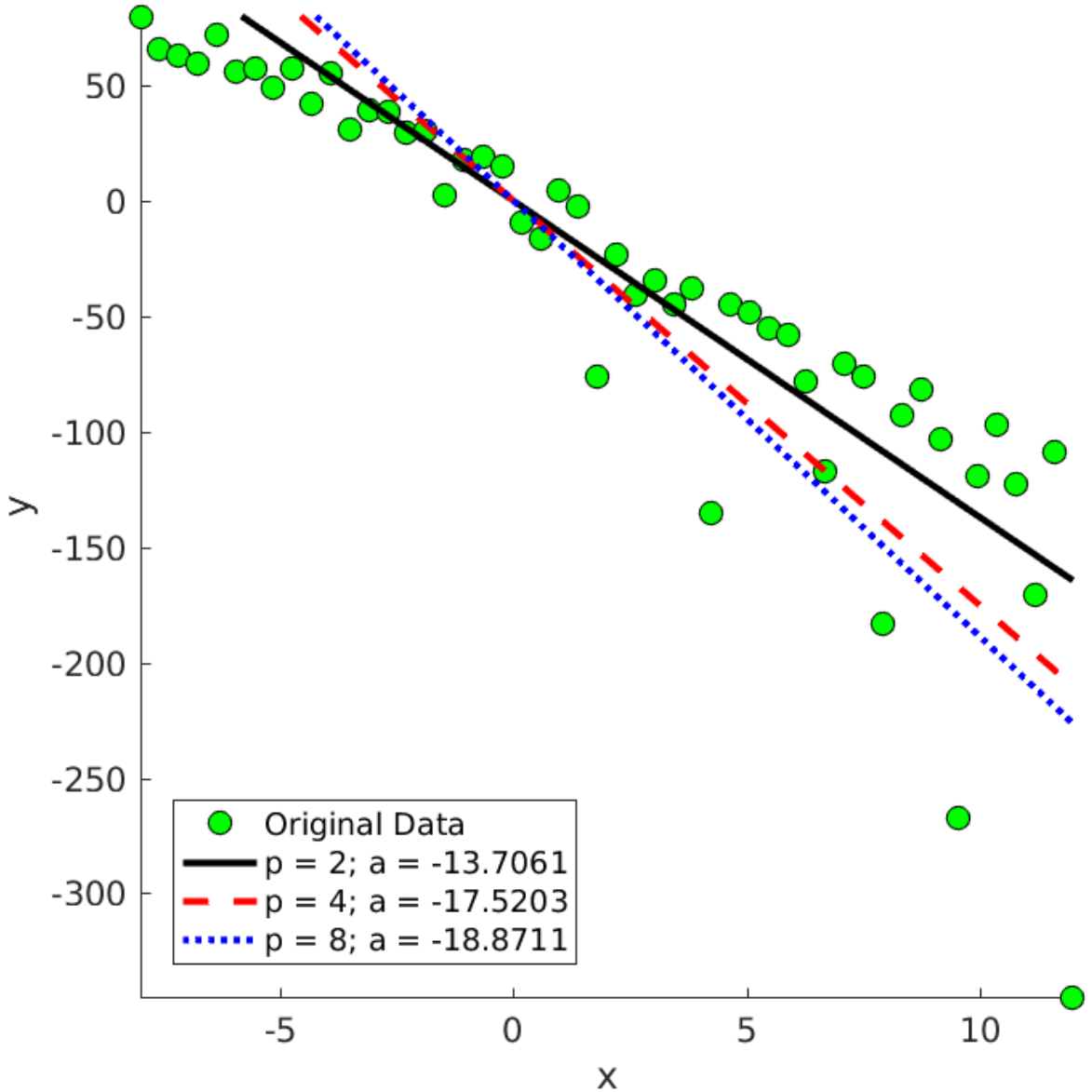
Q6: Fitting $y = ax$ for different error definitions

The larger the p, the most importance the "outliers" have



Q6: Fitting $y = ax$ for different error definitions

The larger the p, the most importance the "outliers" have



Useful functions: find and arrayfun

`find`: find the indices of all non-zero elements of an array

`arrayfun`: apply a function to all elements of an array

Useful functions: find and arrayfun

`find`: find the indices of all non-zero elements of an array

`arrayfun`: apply a function to all elements of an array

See the diary of this lecture (L24) for explanations and examples of how to use these functions

Useful functions: find and arrayfun

`find`: find the indices of all non-zero elements of an array

`arrayfun`: apply a function to all elements of an array

See the diary of this lecture (L24) for explanations and examples of how to use these functions

You can code anything without using these functions (*e.g.*, by using `for` loops instead), but using them might make your code shorter and/or more readable

Note: you do not need to use these functions for lab 09

E7 programming project: first glance

You will have to write a function that **directs a spaceship in a two-dimensional space** (*i.e.* x- and y-coordinates)

E7 programming project: first glance

You will have to write a function that **directs a spaceship in a two-dimensional space** (*i.e.* x- and y-coordinates)

- Your spaceship **collects "scrap"** for points
- Your spaceship **avoids (moving) asteroids**
- There may be **fixed obstacles** to avoid, or **nebulae that slow your spaceship down**
- The grading will be non-competitive
  (*i.e.* your functions will **not** compete against each other)

E7 programming project: first glance

You will have to write a function that **directs a spaceship in a two-dimensional space** (*i.e.* x- and y-coordinates)

- Your spaceship **collects "scrap"** for points
- Your spaceship **avoids (moving) asteroids**
- There may be **fixed obstacles** to avoid, or **nebulae that slow your spaceship down**
- The grading will be non-competitive
  (*i.e.* your functions will **not** compete against each other)

We will provide you with a visualizer, that shows how your ship is doing

E7 programming project: first glance

You will have to write a function that **directs a spaceship in a two-dimensional space** (*i.e.* x- and y-coordinates)

- Your spaceship **collects "scrap"** for points
- Your spaceship **avoids (moving) asteroids**
- There may be **fixed obstacles** to avoid, or **nebulae that slow your spaceship down**
- The grading will be non-competitive
  (*i.e.* your functions will **not** compete against each other)

We will provide you with a visualizer, that shows how your ship is doing

---

The grading will be based on various criteria, such as:

- Collecting scrap efficiently
- Avoiding (moving), asteroids, fixed obstacles, and/or nebulae

E7 programming project: forming teams

Form a team of **2 to 3 students** (no more, no less) **with other students from your lab section**

E7 programming project: forming teams

Form a team of **2 to 3 students** (no more, no less) **with other students from your lab section**

Why from the same lab section?

- **To encourage you to work on your project during lab section** (*e.g.*, get advice from your GSIs)
- **To keep attendance balanced between lab sections** (Please do not attend other lab sections)

E7 programming project: forming teams

Form a team of **2 to 3 students** (no more, no less) **with other students from your lab section**

Why from the same lab section?

- **To encourage you to work on your project during lab section** (*e.g.*, get advice from your GSIs)
- **To keep attendance balanced between lab sections** (Please do not attend other lab sections)

By next Wednesday (March 22^{nd}): **email/bCourses message your primary GSI with the name(s) of your teammate(s)**

If you cannot find teammates, let your GSI know, they will help you form a team