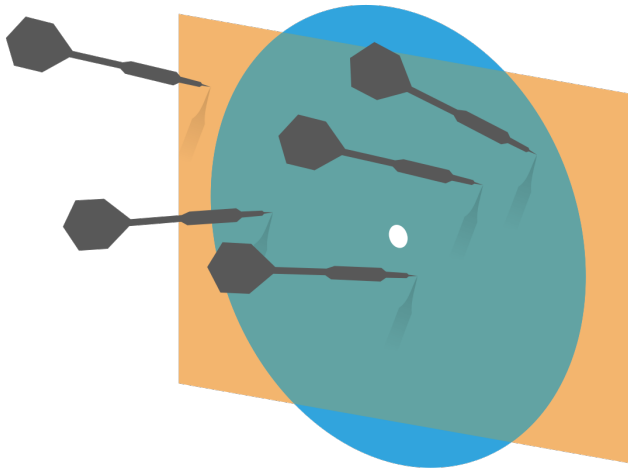


Probability and Statistics for Computer Science



“All models are wrong, but some models are useful” --- George Box

Credit: wikipedia

Last time

- ✱ Linear regression
 - ✱ **The problem**
 - ✱ The least square solution
 - ✱ The training and prediction
 - ✱ The R-squared for the evaluation of the fit.

Linear model

- ✱ We begin by modeling y as a linear function of $\mathbf{x}^{(j)}$ plus randomness

$$y = \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \dots + \mathbf{x}^{(d)}\beta_d + \xi$$

Where ξ is a zero-mean random variable that represents model error

- ✱ In vector notation:

$$y = \mathbf{x}^T \boldsymbol{\beta} + \xi$$

Where $\boldsymbol{\beta}$ is the d -dimensional vector of coefficients that we train

| $\mathbf{x}^{(1)}$ | $\mathbf{x}^{(2)}$ | y |
|--------------------|--------------------|-----|
| 1 | 3 | 0 |
| 2 | 3 | 2 |
| 3 | 6 | 5 |

Each data item gives an equation

✿ The model: $y = \mathbf{x}^T \boldsymbol{\beta} + \xi = \mathbf{x}^{(1)} \beta_1 + \mathbf{x}^{(2)} \beta_2 + \xi$

Training data

| $\mathbf{x}^{(1)}$ | $\mathbf{x}^{(2)}$ | y |
|--------------------|--------------------|-----|
| 1 | 3 | 0 |
| 2 | 3 | 2 |
| 3 | 6 | 5 |

Which together form a matrix equation

✿ The model $y = \mathbf{x}^T \boldsymbol{\beta} + \xi = \mathbf{x}^{(1)} \beta_1 + \mathbf{x}^{(2)} \beta_2 + \xi$

Training data

| $\mathbf{x}^{(1)}$ | $\mathbf{x}^{(2)}$ | y |
|--------------------|--------------------|-----|
| 1 | 3 | 0 |
| 2 | 3 | 2 |
| 3 | 6 | 5 |

$$\begin{bmatrix} 0 \\ 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 2 & 3 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix}$$

Which together form a matrix equation

✿ The model $y = \mathbf{x}^T \boldsymbol{\beta} + \xi = \mathbf{x}^{(1)} \beta_1 + \mathbf{x}^{(2)} \beta_2 + \xi$

Training data

| $\mathbf{x}^{(1)}$ | $\mathbf{x}^{(2)}$ | y |
|--------------------|--------------------|-----|
| 1 | 3 | 0 |
| 2 | 3 | 2 |
| 3 | 6 | 5 |

$$\begin{bmatrix} 0 \\ 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 2 & 3 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix}$$

$$\mathbf{y} = \mathbf{X} \cdot \boldsymbol{\beta} + \mathbf{e}$$

Training the model is to choose β

- ✱ Given a training dataset $\{(\mathbf{x}, y)\}$, we want to fit a model $y = \mathbf{x}^T \beta + \xi$

- ✱ Define $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$ and $X = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix}$ and $\mathbf{e} = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_N \end{bmatrix}$

- ✱ To train the model, we need to choose β that makes \mathbf{e} small in the matrix equation $\mathbf{y} = X \cdot \beta + \mathbf{e}$

Training using least squares

- ✱ In the least squares method, we aim to **minimize** $\|\mathbf{e}\|^2$

$$\|\mathbf{e}\|^2 = \|\mathbf{y} - X\boldsymbol{\beta}\|^2 = (\mathbf{y} - X\boldsymbol{\beta})^T (\mathbf{y} - X\boldsymbol{\beta})$$

- ✱ Differentiating with respect to $\boldsymbol{\beta}$ and setting to zero

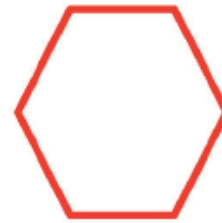
$$X^T X \boldsymbol{\beta} - X^T \mathbf{y} = 0$$

- ✱ If $X^T X$ is invertible, the least squares estimate of the coefficient is:

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \mathbf{y}$$

Convex set and convex function

- ✱ If a set is convex, any line connecting two points in the set is completely included in the set



(a)



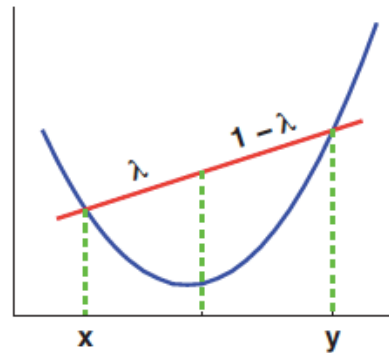
(b)

Figure 7.4 (a) Illustration of a convex set. (b) Illustration of a nonconvex set.

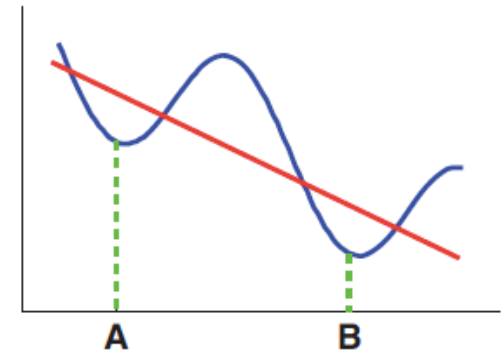
- ✱ A convex function: the area above the curve is convex

$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y)$$

- ✱ The least square function is **convex**



(a)



(b)

Training using least squares example

✿ Model: $y = \mathbf{x}^T \boldsymbol{\beta} + \xi = \mathbf{x}^{(1)} \beta_1 + \mathbf{x}^{(2)} \beta_2 + \xi$

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \mathbf{y} = \begin{bmatrix} 2 \\ -\frac{1}{3} \end{bmatrix}$$

Training data

| $\mathbf{x}^{(1)}$ | $\mathbf{x}^{(2)}$ | y |
|--------------------|--------------------|-----|
| 1 | 3 | 0 |
| 2 | 3 | 2 |
| 3 | 6 | 5 |

$$\hat{\beta}_1 = 2$$
$$\hat{\beta}_2 = -\frac{1}{3}$$

Prediction

- ✱ If we train the model coefficients $\hat{\beta}$, we can predict y_0^p from \mathbf{x}_0

$$y_0^p = \mathbf{x}_0^T \hat{\beta}$$

- ✱ In the model $y = \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \xi$ with $\hat{\beta} = \begin{bmatrix} 2 \\ -\frac{1}{3} \end{bmatrix}$

- ✱ The prediction for $\mathbf{x}_0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ is y_0^p

- ✱ The prediction for $\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ is y_0^p

A linear model with constant offset

- ✱ The problem with the model $y = \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \xi$ is it always predicts $y_0^p = 0$ if the input vector $\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

- ✱ Let's add a constant offset β_0 to the model

$$y = \beta_0 + \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \xi$$

Training and prediction with constant offset

✱ The model $y = \beta_0 + \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \xi = \mathbf{x}^T\boldsymbol{\beta} + \xi$

✱ Training data:

$$\begin{bmatrix} 1 & x^{(1)} & x^{(2)} \end{bmatrix}$$

| 1 | $\mathbf{x}^{(1)}$ | $\mathbf{x}^{(2)}$ | y |
|----------|--------------------------------------|--------------------------------------|-----------------------|
| 1 | 1 | 3 | 0 |
| 1 | 2 | 3 | 2 |
| 1 | 3 | 6 | 5 |

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \mathbf{y} = \begin{bmatrix} -3 \\ 2 \\ \frac{1}{3} \end{bmatrix}$$

✱ For $\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$$y_0^p = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} -3 \\ 2 \\ \frac{1}{3} \end{bmatrix} = -3$$

Variance of the linear regression model

- ✱ The least squares estimate satisfies this property

$$\text{var}(\{y_i\}) = \text{var}(\{\mathbf{x}_i^T \hat{\boldsymbol{\beta}}\}) + \text{var}(\{\xi_i\})$$

- ✱ The random error is uncorrelated to the least square solution of linear combination of explanatory variables.

Evaluating models using R-squared

- ✱ The least squares estimate satisfies this property

$$\text{var}(\{y_i\}) = \text{var}(\{\mathbf{x}_i^T \hat{\boldsymbol{\beta}}\}) + \text{var}(\{\xi_i\})$$

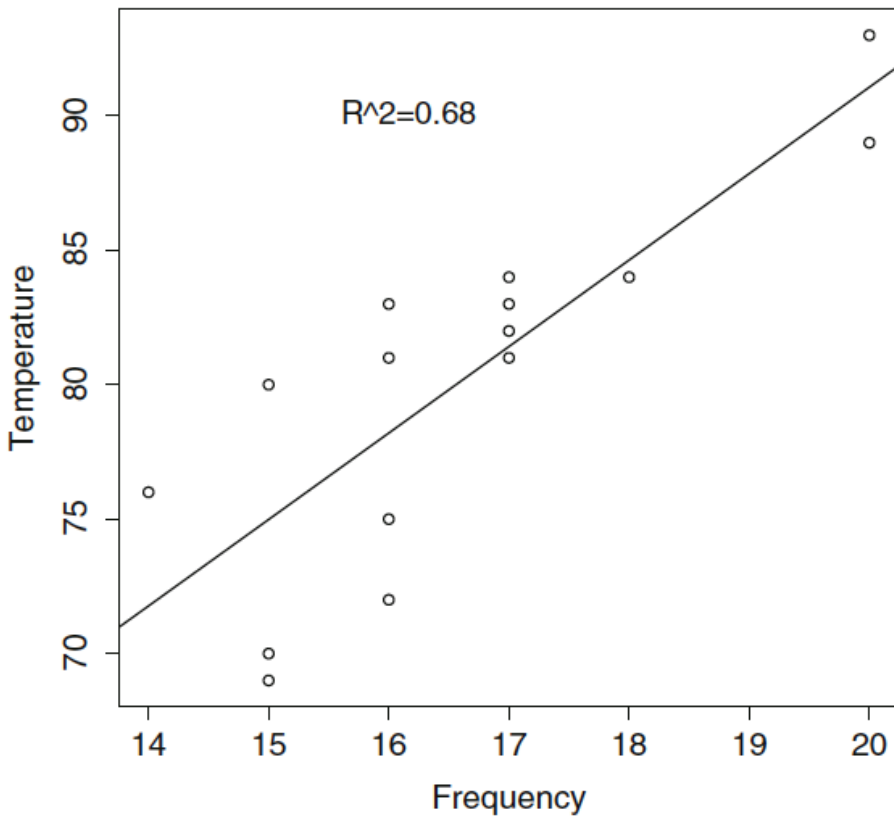
- ✱ This property gives us an evaluation metric called R-squared

$$R^2 = \frac{\text{var}(\{\mathbf{x}_i^T \hat{\boldsymbol{\beta}}\})}{\text{var}(\{y_i\})}$$

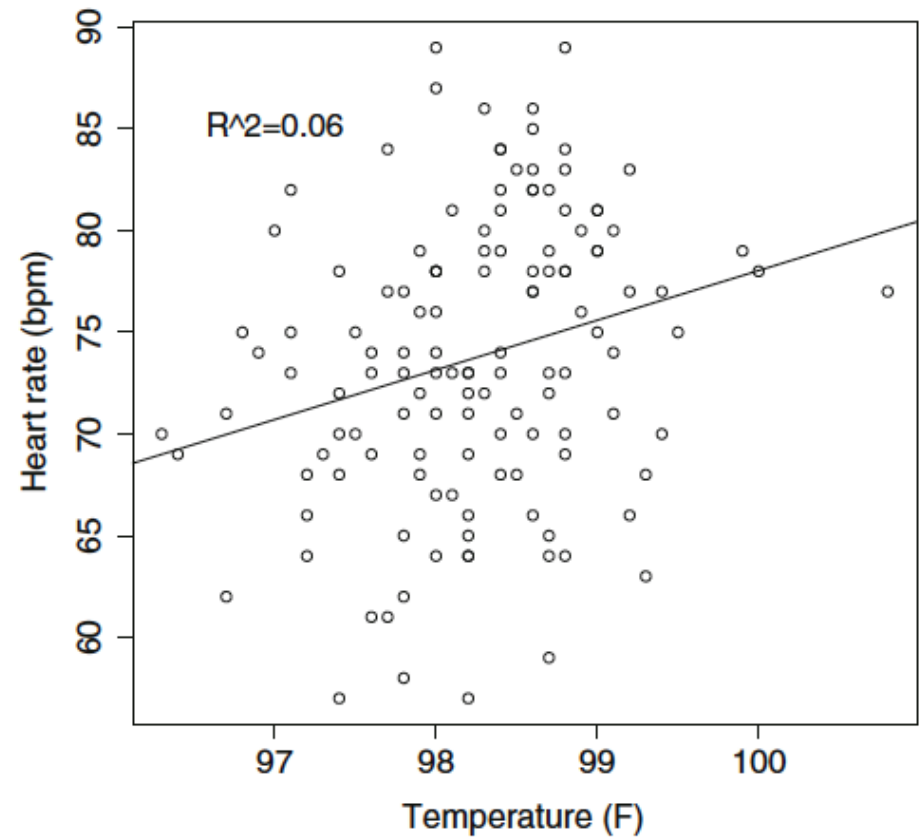
- ✱ We have $0 \leq R^2 \leq 1$ with a larger value meaning a better fit.

R-squared examples

Chirp frequency vs temperature in crickets



Heart rate vs temperature in humans



Linear regression model for the Chicago census data

Call:

```
lm(formula = HardshipIndex ~ ., data = dat)
```

Residuals:

| Min | 1Q | Median | 3Q | Max |
|----------|---------|--------|--------|--------|
| -15.7157 | -1.9230 | 0.1301 | 1.9810 | 8.6719 |

Coefficients:

| | Estimate | Std. Error | t value | Pr(> t) | |
|--|----------|------------|---------|----------|-----|
| (Intercept) | 105.1394 | 37.3622 | 2.814 | 0.006346 | ** |
| PERCENT_OF_HOUSING_CROWDED | 0.7189 | 0.2753 | 2.612 | 0.011014 | * |
| PERCENT_HOUSEHOLDS_BELOW_POVERTY | 0.6665 | 0.0781 | 8.534 | 1.90e-12 | *** |
| PERCENT_AGED_16p_UNEMPLOYED | 0.8023 | 0.1350 | 5.941 | 9.93e-08 | *** |
| PERCENT_AGED_25p_WITHOUT_HIGH_SCHOOL_DIPLOMA | 0.7751 | 0.1063 | 7.293 | 3.64e-10 | *** |
| PERCENT_AGED_UNDER_18_OR_OVER_64 | 0.4807 | 0.1202 | 3.998 | 0.000156 | *** |
| PER_CAPITA_INCOME | -11.8819 | 3.1888 | -3.726 | 0.000391 | *** |

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.9 on 70 degrees of freedom

Multiple R-squared: 0.983, Adjusted R-squared: 0.9815

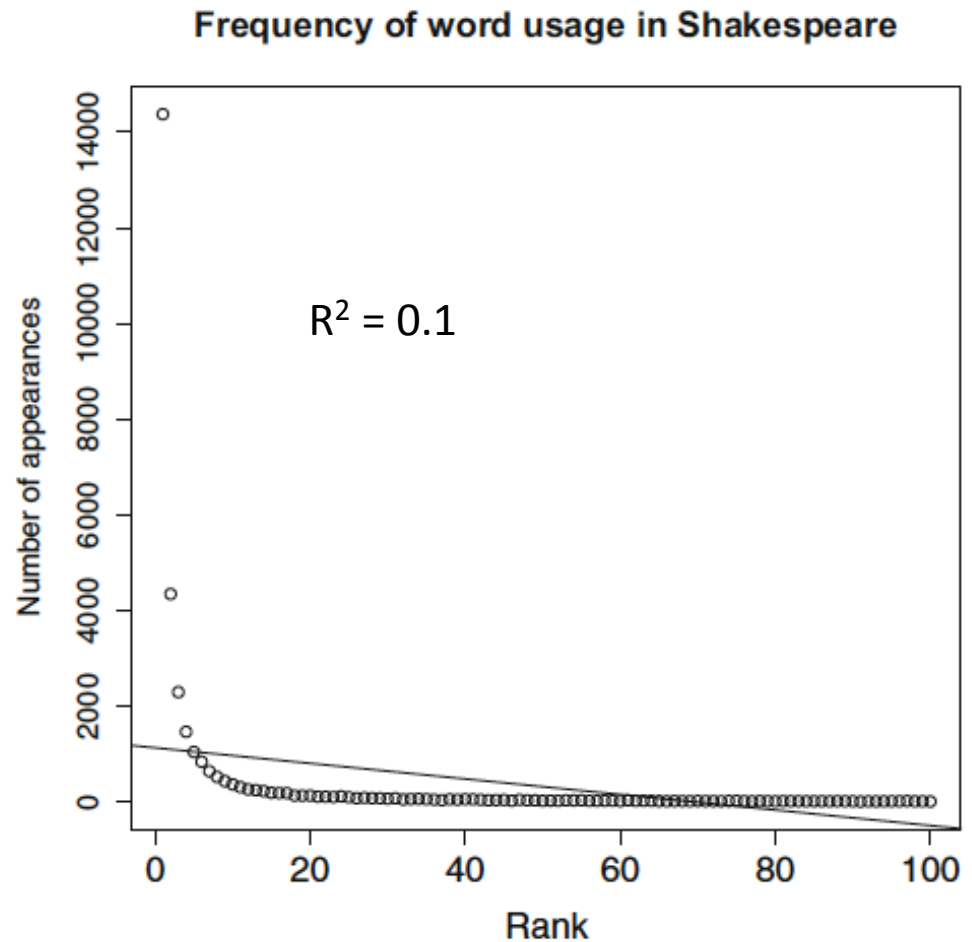
F-statistic: 673.9 on 6 and 70 DF, p-value: < 2.2e-16

Contents

- ✱ Linear regression (cont.)
 - ✱ Modeling non-linear relationship with linear regression
 - ✱ Outliers and over-fitting issues
 - ✱ Regularized linear regression/Ridge regression
- ✱ Nearest neighbor regression

What if the relationship between variables is non-linear?

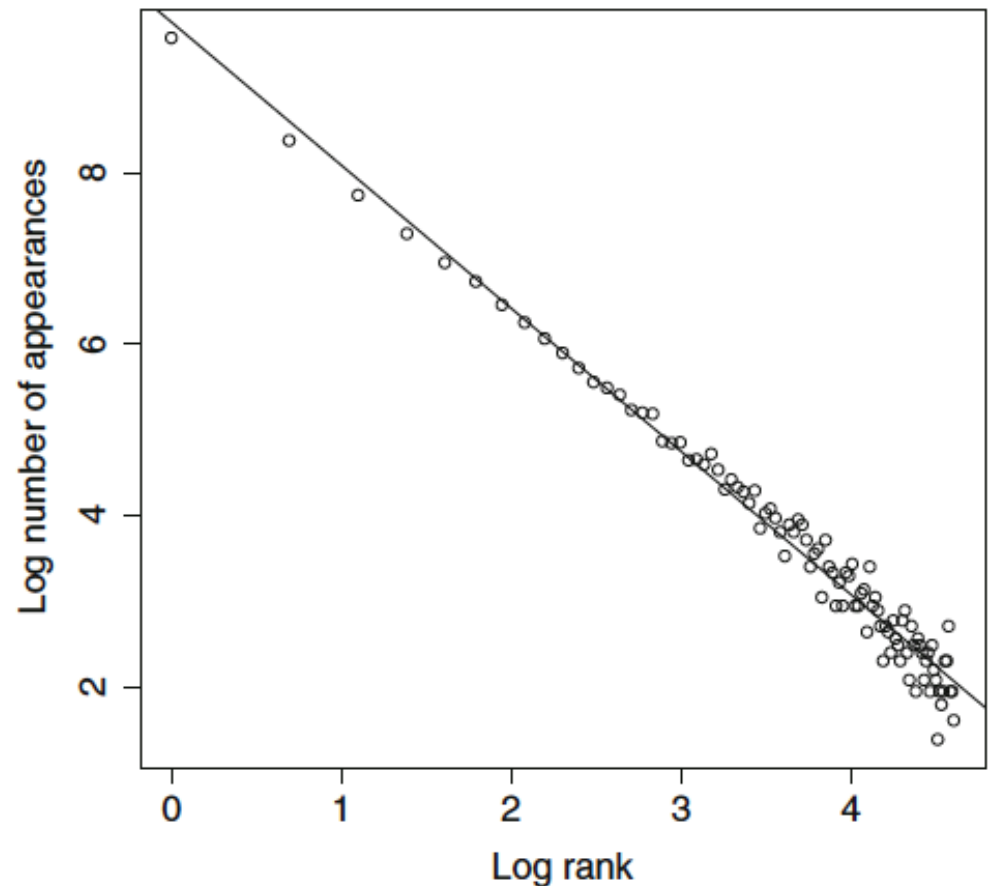
- ✱ A linear model will not produce a good fit if the dependent variable is **not** linear combination of the explanatory variables



Transforming variables could allow linear model to model non-linear relationship

- ✱ In the word-frequency example, log-transforming both variables would allow a linear model to fit the data well.

Frequency of word usage in Shakespeare, log-log



More example: Data of fish in a Finland lake

- ✧ Perch (a kind of fish) in a lake in Finland, 56 data observations
- ✧ Variables include: Weight, Length, Height, Width
- ✧ In order to illustrate the point, let's model **Weight** as the dependent variable and the **Length** as the explanatory variable.



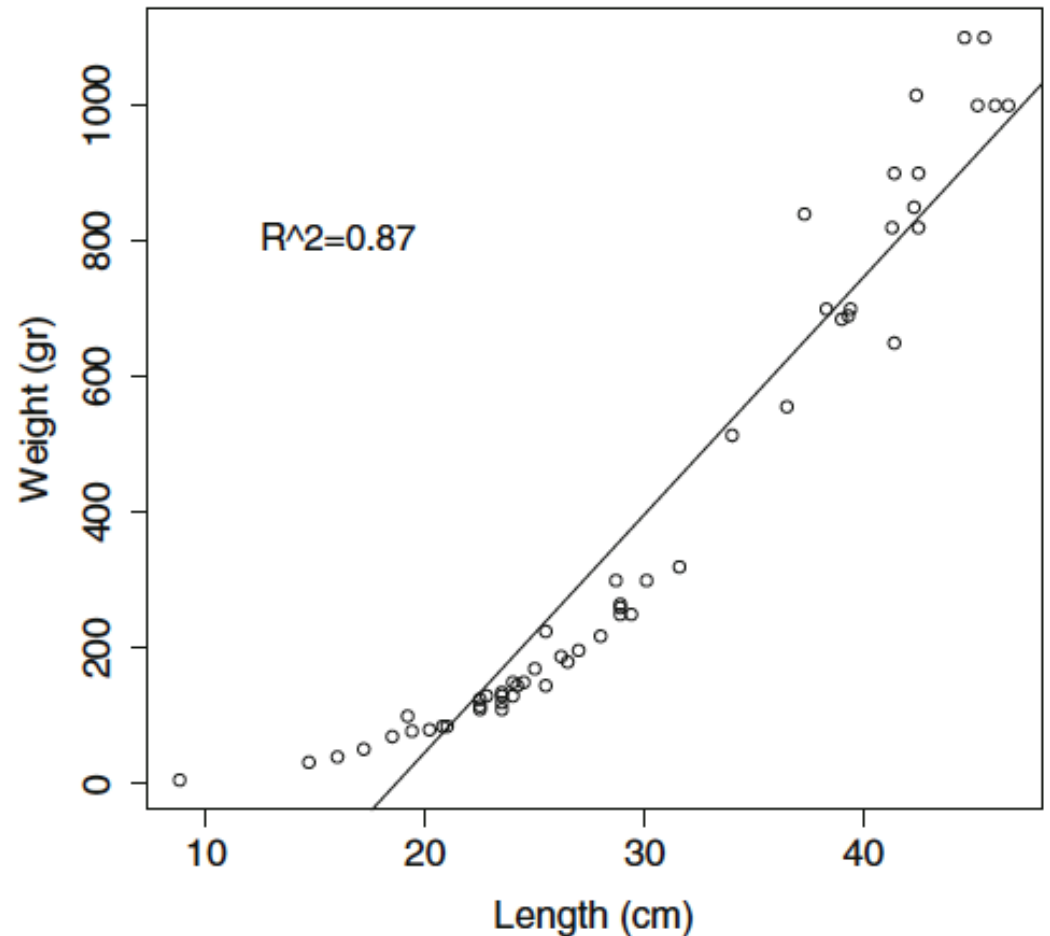
Yellow Perch

Is the linear model fine for this data?

A. YES

B. NO

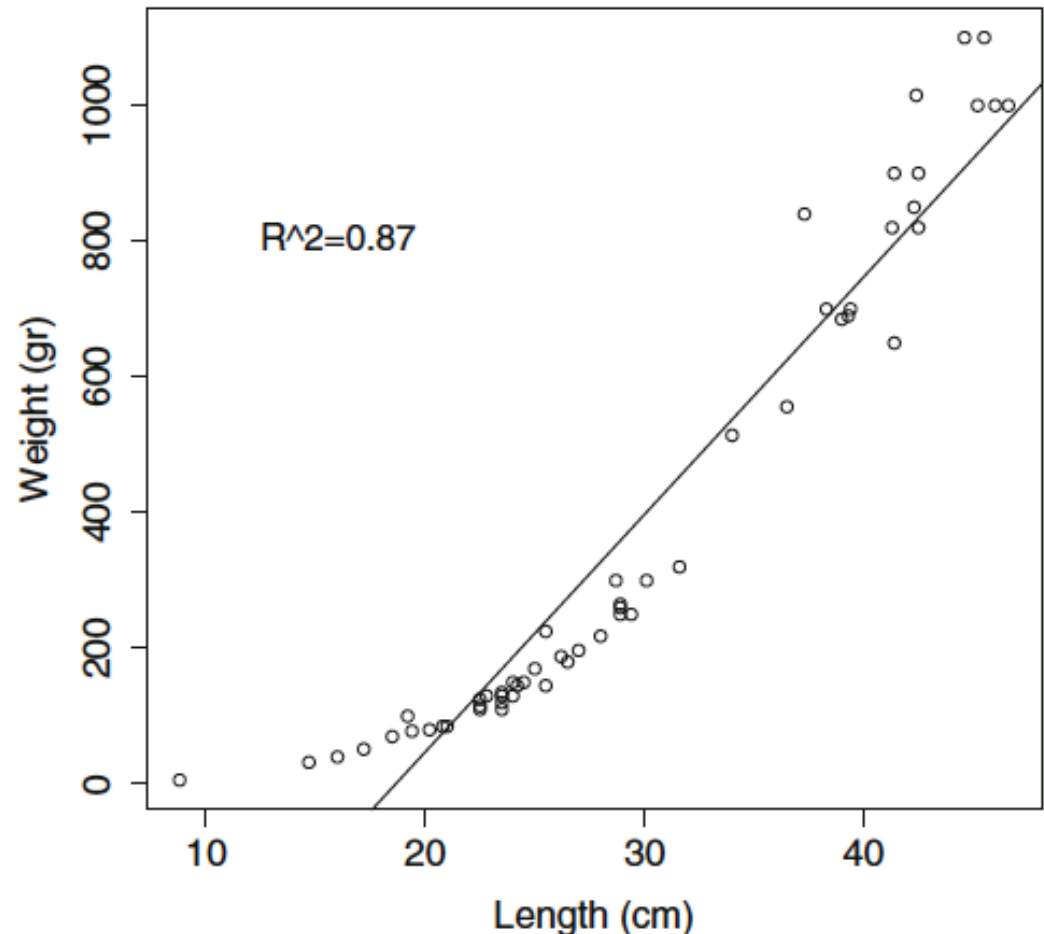
Weight vs length in perch from Lake Laengelmavesi



Is the linear model fine for this data?

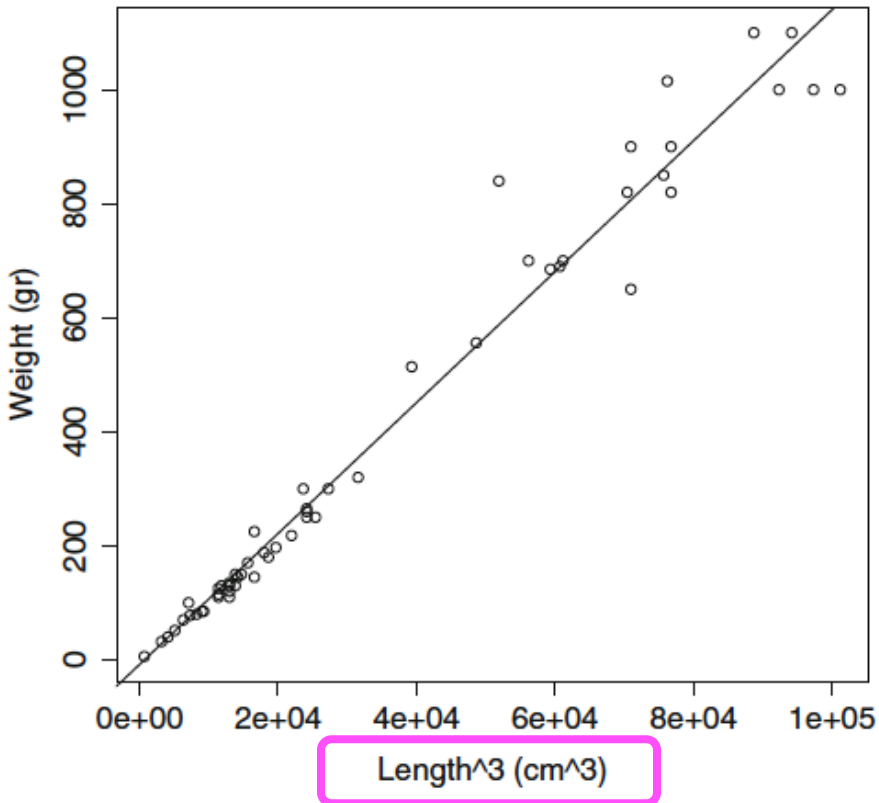
- ✱ R-squared is 0.87 may suggest the model is OK
- ✱ But the trend of the data suggests non-linear relationship
- ✱ Intuition tells us length is not linear to weight given fish is 3-dimensional
- ✱ We can do better!

Weight vs length in perch from Lake Laengelmavesi

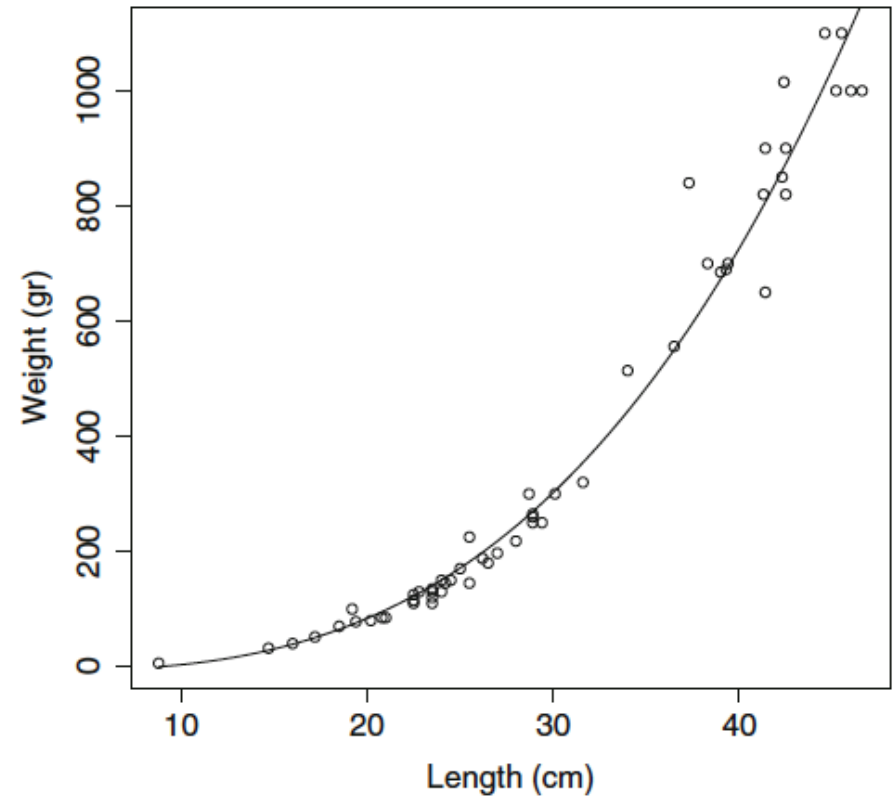


Transforming the explanatory variables

Weight vs length³ in perch from Lake Laengelmavesi



Weight predicted from length³ in perch from Lake Laengelmavesi

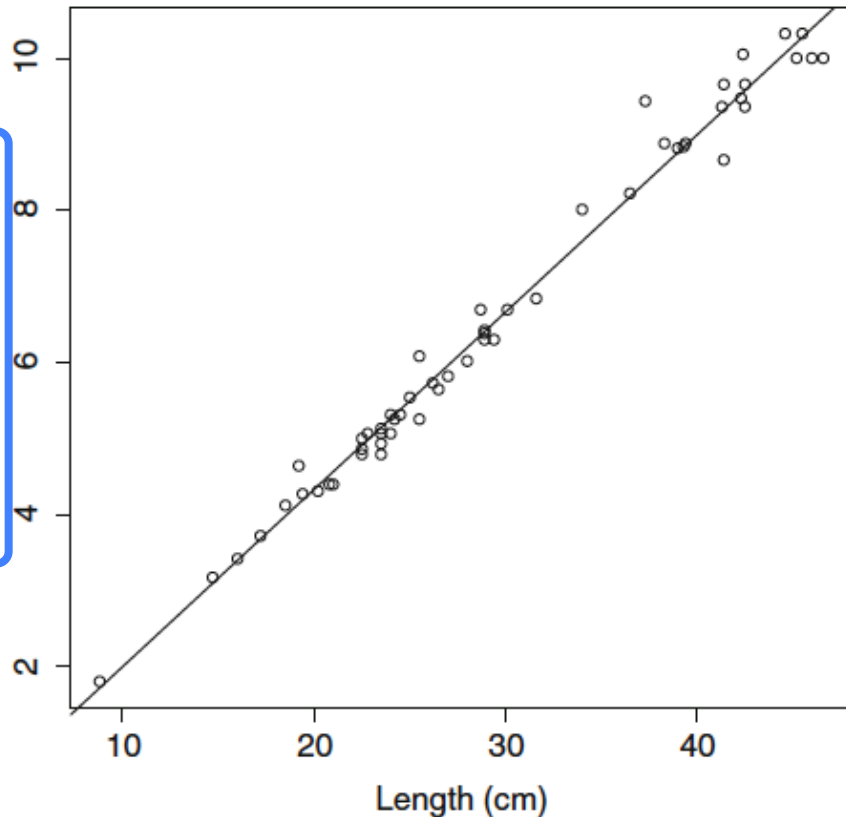


Q. What are the matrix X and y ?

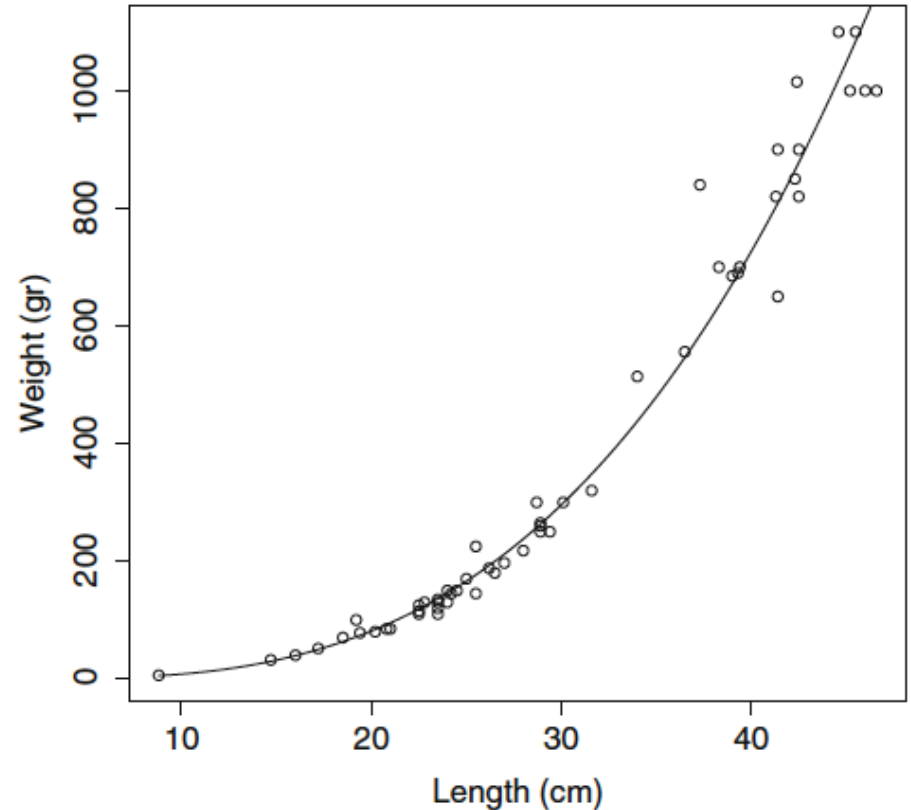
| 1 | Length ³ | Weight |
|---|---------------------|--------|
| | | |

Transforming the dependent variables

Weight^(1/3) vs length in perch from Lake Laengelmavesi



Weight^(1/3) predicted from length in perch from Lake Laengelmavesi



What is the model now?



What are the matrix X and y ?

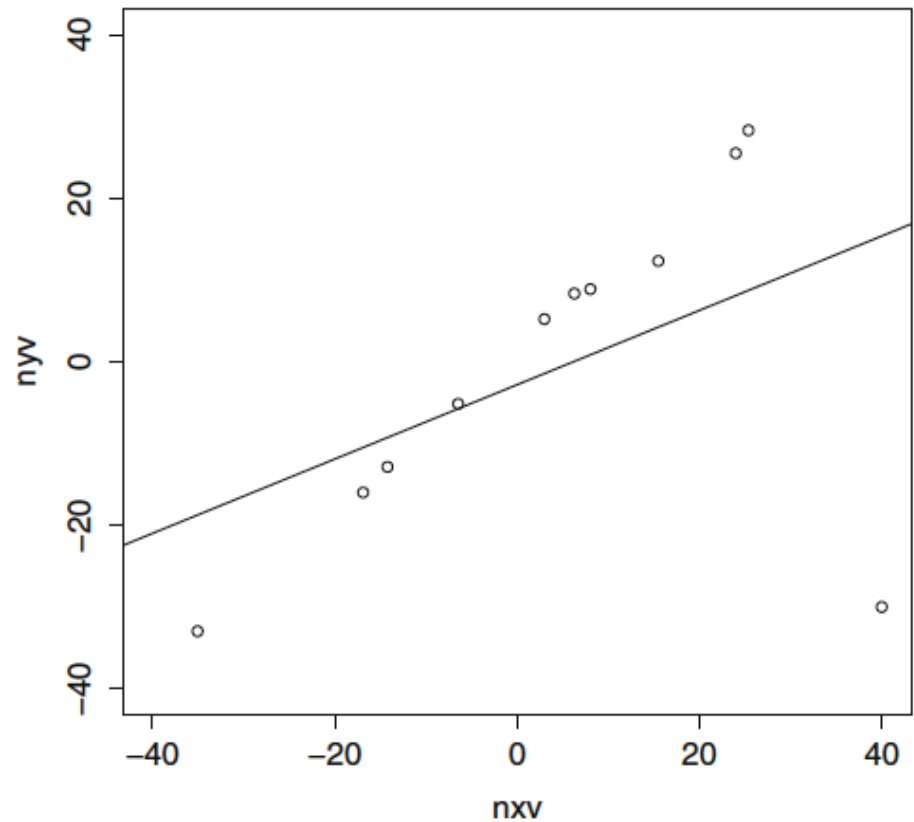
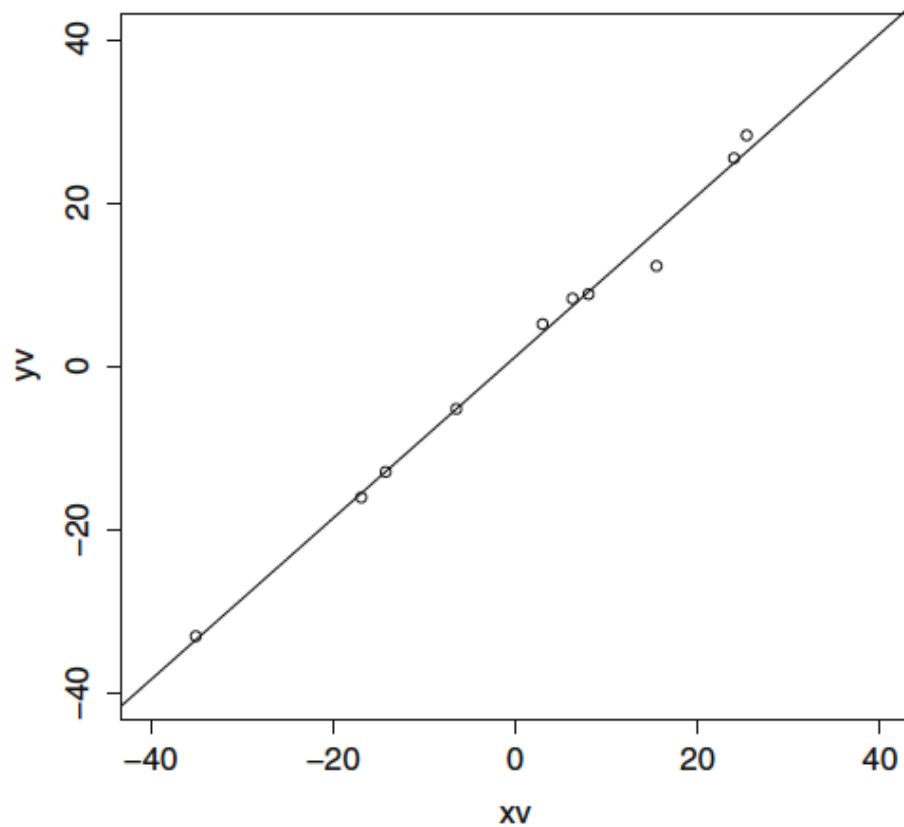
| | | |
|---|--------|---------------|
| 1 | Length | $\sqrt[3]{w}$ |
|---|--------|---------------|

Contents

- ✱ Linear regression (cont.)
 - ✱ Modeling non-linear relationship with linear regression
 - ✱ **Outliers and over-fitting issues**
 - ✱ Regularization of linear regression/Ridge regression
- ✱ Nearest neighbor regression

Effect of outliers on linear regression

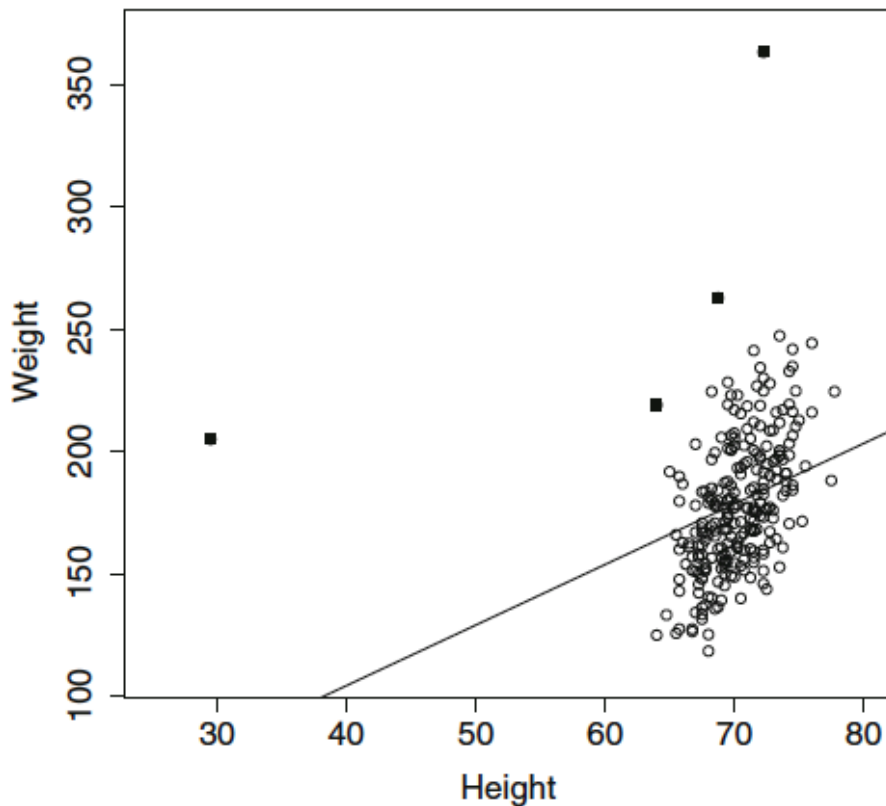
✱ Linear regression is sensitive to outliers



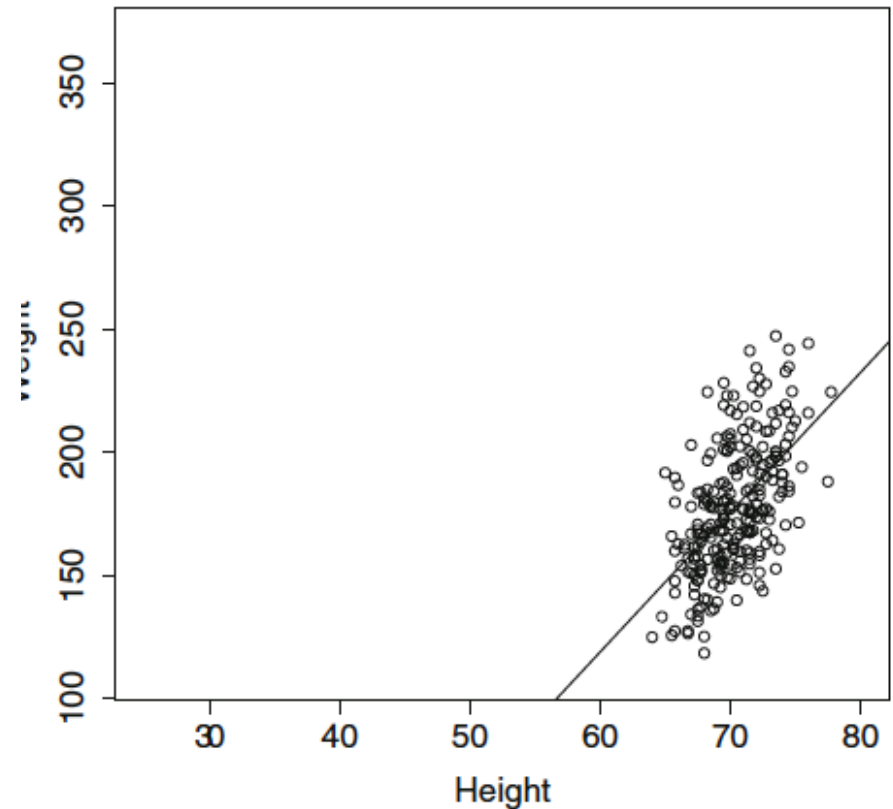
Effect of outliers: body fat example

- ✱ Linear regression is sensitive to outliers

Weight against height, all points

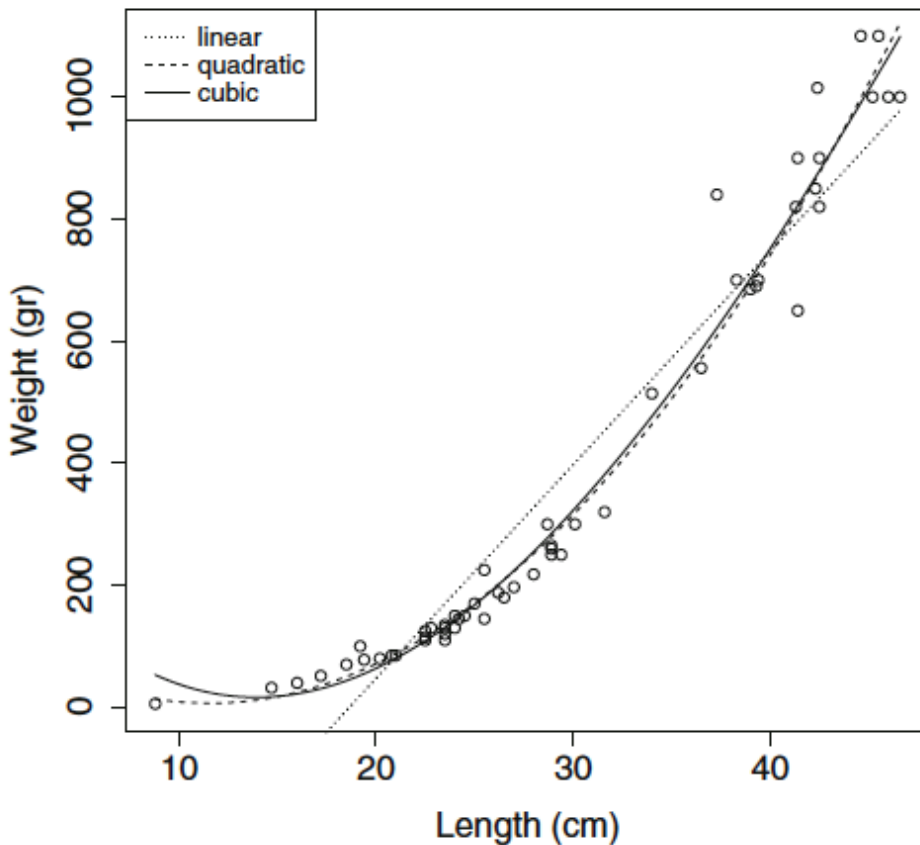


Weight against height, 4 outliers removed

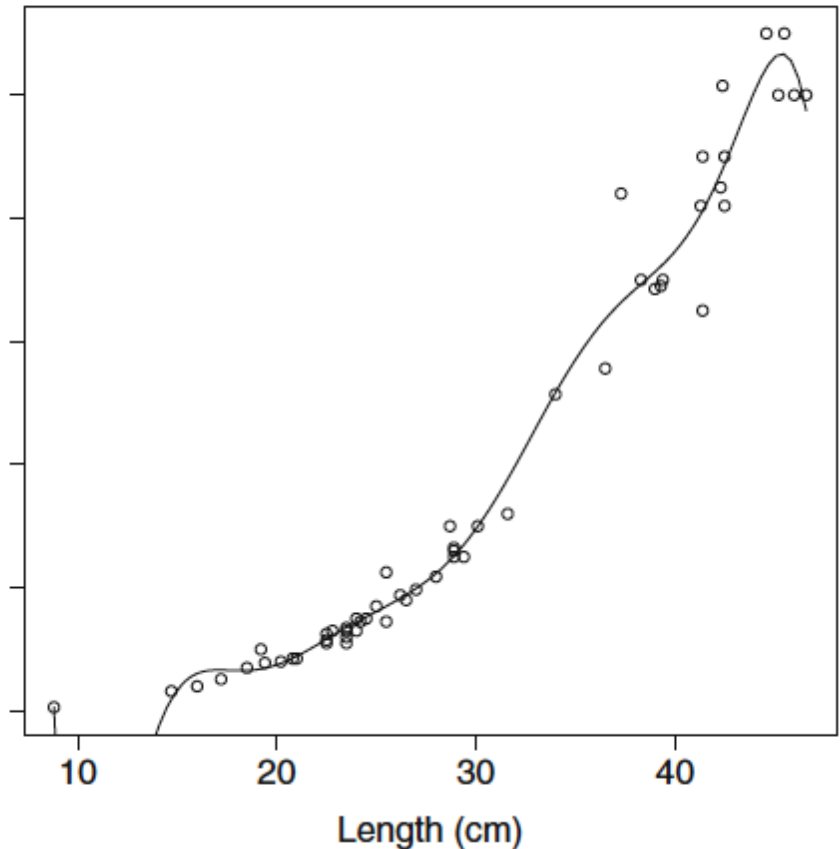


Over-fitting issue: example of using too many power transformations

Weight vs length in perch from Lake Laengelmavesi, three models.



Weight vs length in perch from Lake Laengelmavesi, all powers up to 10.



Avoiding over-fitting

* **Method 1: validation**

- * Use a validation set to choose the transformed explanatory variables
- * The difficulty is the number of combination is exponential in the number of variables.

* **Method 2: regularization**

- * Impose a penalty on complexity of the model during the training
- * Encourage smaller model coefficients
- * We can use validation to select regularization parameter λ

Regularized linear regression

- ✱ In ordinary least squares, the cost function is $\|\mathbf{e}\|^2$:

$$\|\mathbf{e}\|^2 = \|\mathbf{y} - X\boldsymbol{\beta}\|^2 = (\mathbf{y} - X\boldsymbol{\beta})^T (\mathbf{y} - X\boldsymbol{\beta})$$

- ✱ In regularized least squares, we add a penalty weighted parameter λ ($\lambda > 0$):

$$\|\mathbf{y} - X\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|^2 = (\mathbf{y} - X\boldsymbol{\beta})^T (\mathbf{y} - X\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^T \boldsymbol{\beta}$$

Training using regularized least squares

- ✱ Differentiating the cost function and setting it to zero, one gets:

$$(X^T X + \lambda I)\boldsymbol{\beta} - X^T \mathbf{y} = 0$$

- ✱ $(X^T X + \lambda I)$ is always invertible, so the regularized least squares estimation of the coefficients is:

$$\hat{\boldsymbol{\beta}} = (X^T X + \lambda I)^{-1} X^T \mathbf{y}$$

Why is the regularized version always invertible?

Prove: $(X^T X + \lambda I)$ is invertible ($\lambda > 0$, λ is not the eigenvalue).

Energy based definition of **semi-positive definite**:

Given a matrix A and any nonzero vector f , we have

$$f^T A f \geq 0$$

and **positive definite** means

$$f^T A f > 0$$

If A is positive definite, then all eigenvalues of A are positive, then it's invertible

Why is the regularized version always invertible?

Prove: $(X^T X + \lambda I)$ is invertible ($\lambda > 0$, λ is not the eigenvalue).

Energy based definition of **semi-positive definite**:

Given a matrix A and any nonzero vector f , we have

$$f^T A f \geq 0$$

and **positive definite** means

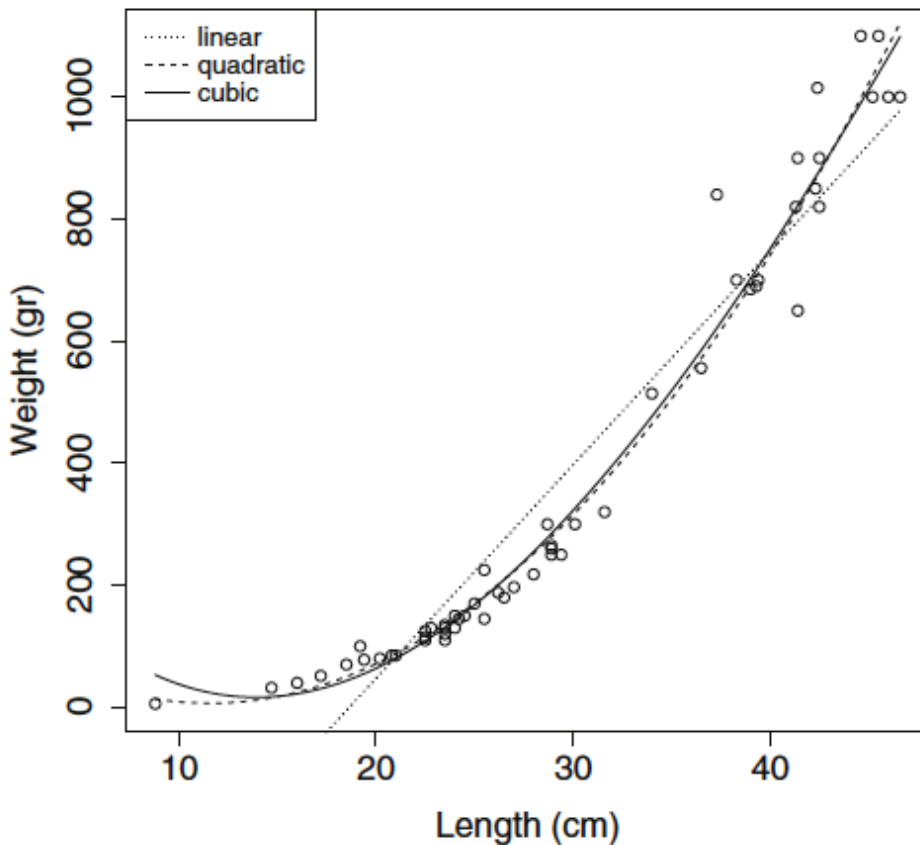
$$f^T A f > 0$$

If A is positive definite, then all eigenvalues of A are positive, then it's invertible

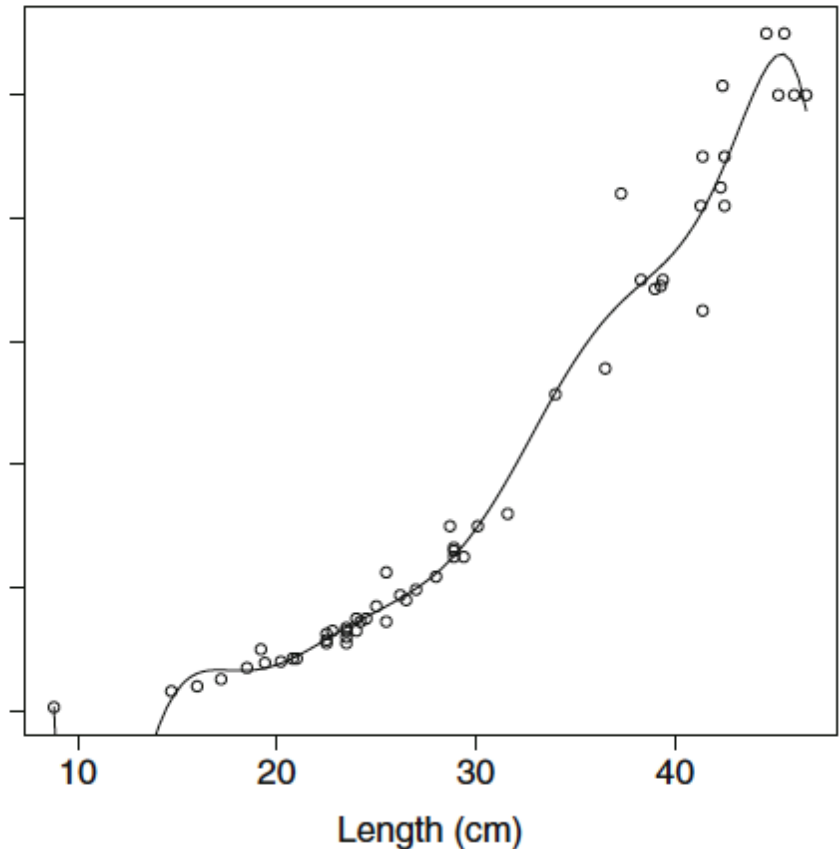
*for any nonzero vector f
consider $f^T (X^T X + \lambda I) f$
suppose $A = X^T X + \lambda I$
 $f^T A f = f^T X^T X f + \lambda f^T f$
 $= f^T X^T X f + \lambda \|f\|^2$
given $X^T X$ is semi positive definite
 $f^T X^T X f \geq 0$
given $\lambda > 0$
we know $\lambda \|f\|^2 > 0$
 $\Rightarrow f^T A f > 0$*

Over-fitting issue: example from using too many power transformations

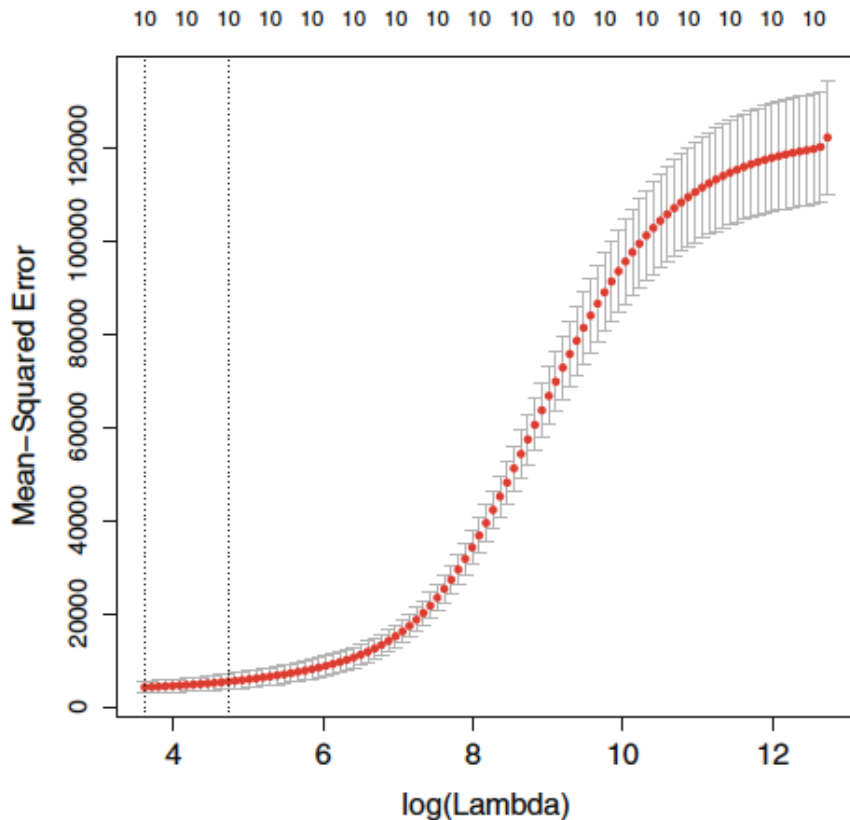
Weight vs length in perch from Lake Laengelmavesi, three models.



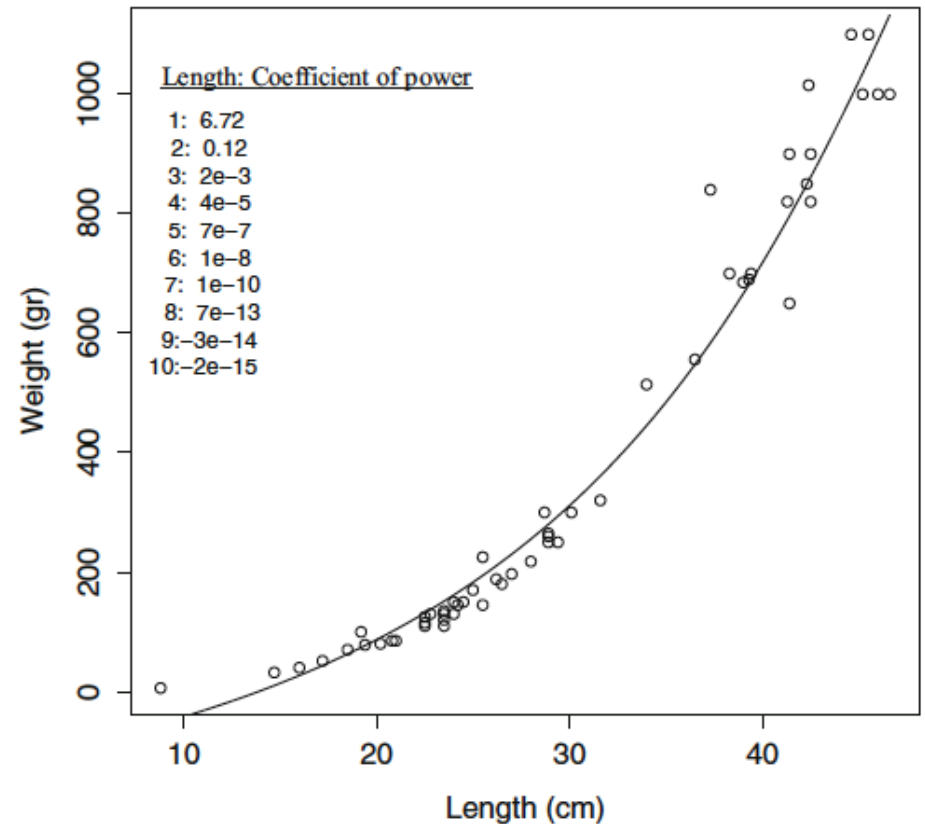
Weight vs length in perch from Lake Laengelmavesi, all powers up to 10.



Choosing lambda using cross-validation



Weight vs length in perch from Lake Laengelmavesi, all powers up to 10, regularized



Some remarks on the regularized regression

- * The penalty is not placed on the constant offset
- * The regularized regression is not linear scale invariant any more

Q. Can we use the R-squared to evaluate the regularized model correctly?

A. YES

B. NO

Contents

- ✱ Linear regression (cont.)
 - ✱ Modeling non-linear relationship with linear regression
 - ✱ Outliers and over-fitting issues
 - ✱ Regularized linear regression/Ridge regression
- ✱ **Nearest neighbor regression**

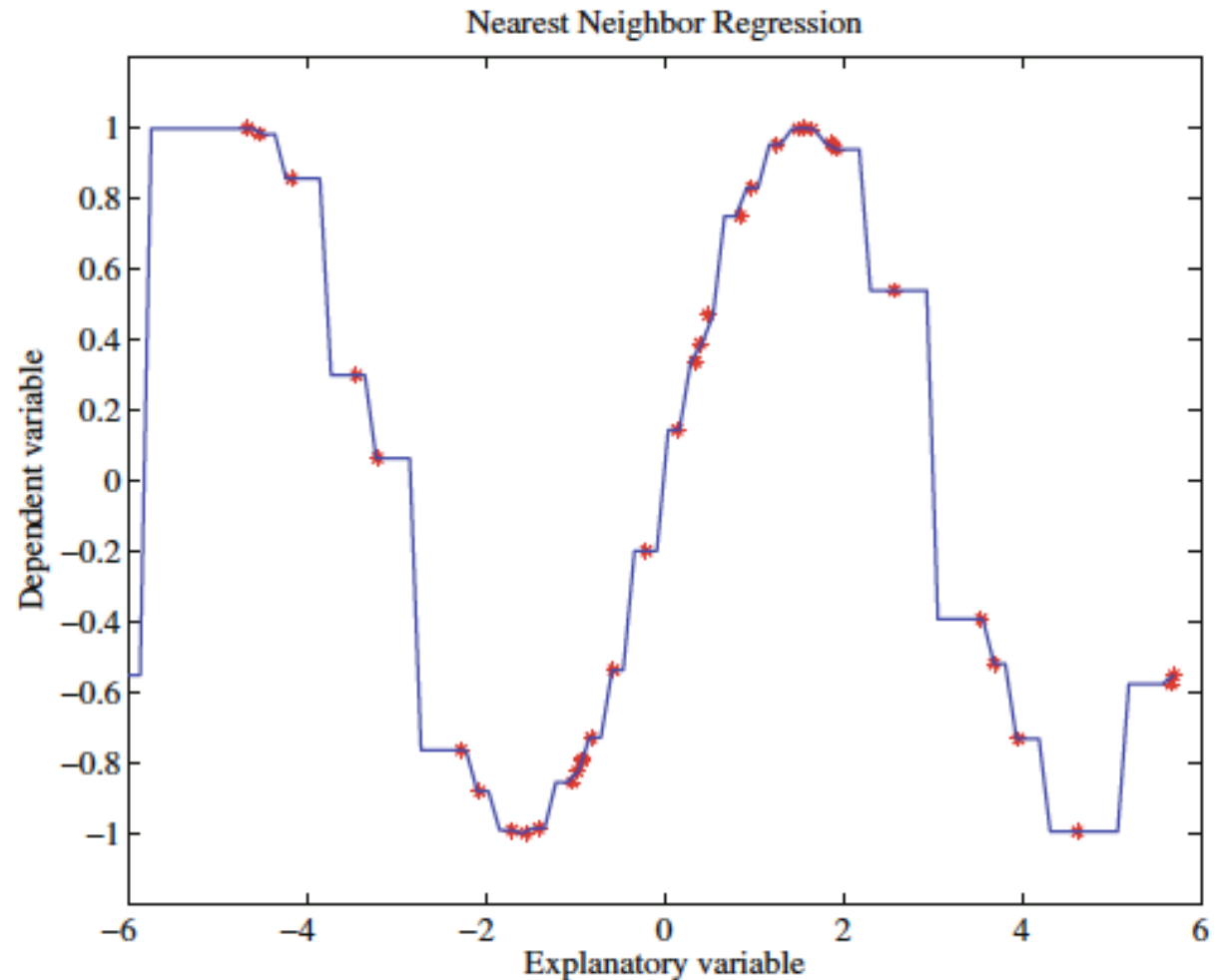
Nearest neighbor regression

- ✱ In addition to linear regression and generalized linear regression models, there are methods such as **Nearest neighbor regression** that do not need much training for the model parameters.
- ✱ When there is plenty of data, nearest neighbors regression can be used effectively

K nearest neighbor regression with $k=1$

The idea is very similar to k-nearest neighbor classifier, but the regression model predicts numbers

$K=1$ gives piecewise constant predictions



K nearest neighbor regression with weights

The goal is to predict y_0^p from \mathbf{x}_0 using a training set $\{(\mathbf{x}, y)\}$

- ✱ Let $\{(\mathbf{x}_j, y_j)\}$ be the set of k items in the training data set that are closest to \mathbf{x}_0 .
- ✱ Prediction is following:

$$y_0^p = \frac{\sum_j \mathbf{w}_j y_j}{\sum_j \mathbf{w}_j}$$

Where \mathbf{w}_j are weights that drop off as \mathbf{x}_j gets further away from \mathbf{x}_0 .

Choose different weights functions for KNN regression

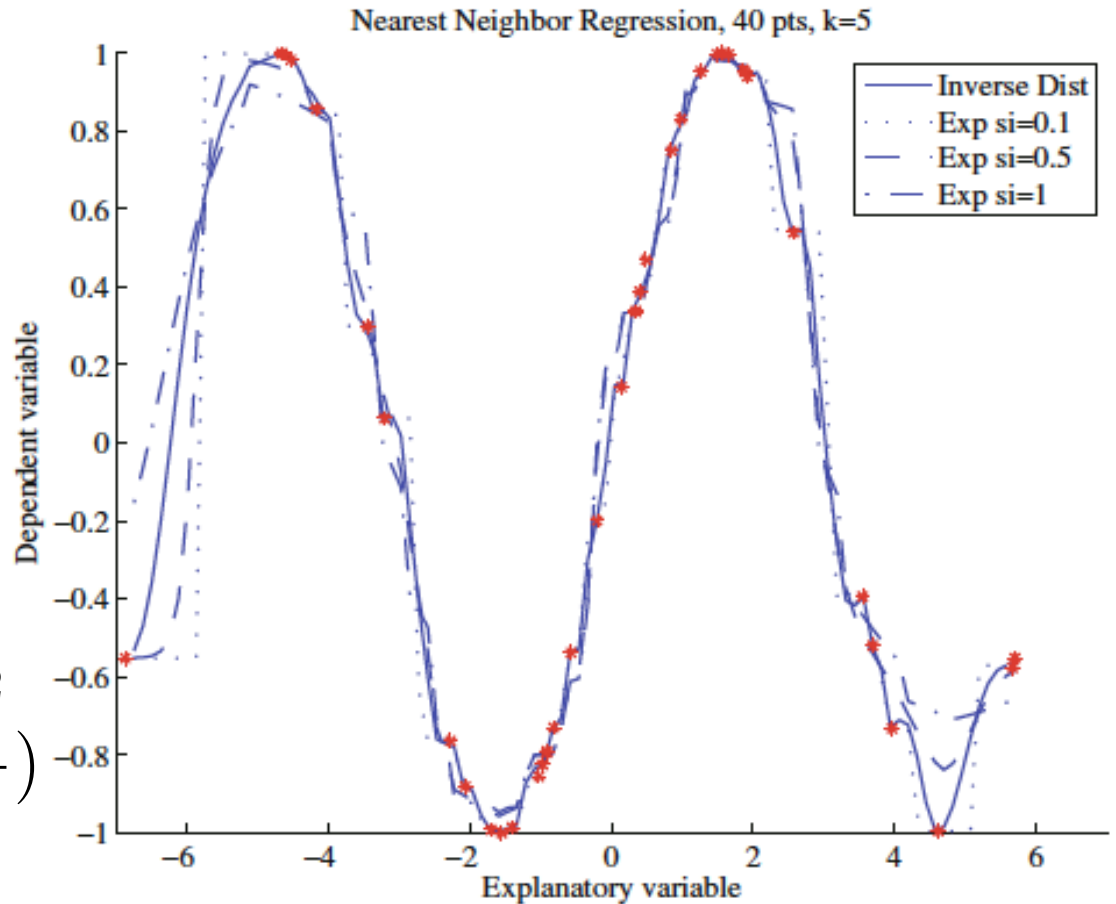
$$y_0^p = \frac{\sum_j w_j y_j}{\sum_j w_j}$$

✱ Inverse distance

$$w_j = \frac{1}{\|\mathbf{x}_0 - \mathbf{x}_j\|}$$

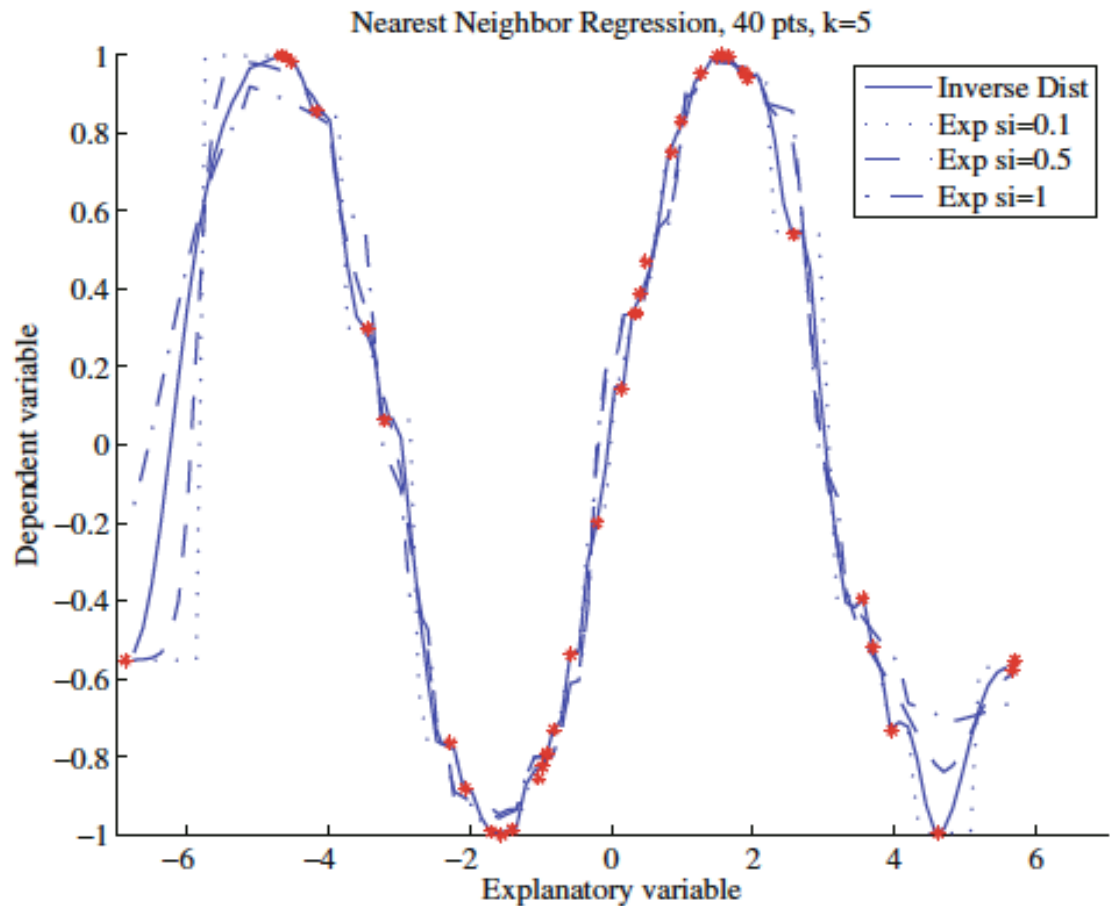
✱ Exponential function

$$w_j = \exp\left(-\frac{\|\mathbf{x}_0 - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$



Evaluation of KNN models

- ✱ Which methods do you use to choose K and weight functions?
- A. Cross validation
 - B. Evaluation of MSE
 - C. Both A and B



The Pros and Cons of K nearest neighbor regression

✱ Pros:

- ✱ The method is very intuitive and simple
- ✱ You can predict more than numbers as long as you can define a similarity measure.

✱ Cons

- ✱ The method doesn't work well for very high dimensional data
- ✱ The model depends on the scale of the data

Assignments

- ✱ Finish Chapter 13 of the textbook
- ✱ Next time: Curse of Dimension, clustering

Additional References

- ✱ Robert V. Hogg, Elliot A. Tanis and Dale L. Zimmerman. “Probability and Statistical Inference”
- ✱ Kelvin Murphy, “Machine learning, A Probabilistic perspective”

See you next time

*See
You!*

