Basics of modeling

Today's topics

- Fields of Machine Learning
- Procedures of analyzing data
- Bias-Variance trade off
- Parametric model vs Nonparametric model
- Data type

Fields of Machine Learning

- Supervised Learning : Y~X
- Unsupervised Learning : X
- Reinforcement Learning : R

Procedure of analyzing data

- Specifying a goal of analysis
- Collecting data with respect to the goal
- while(Fail to satisfy the goal)
  Do EDA(visualization, simple modeling, clustering…)
  Do Modeling
- Conclusion

Procedure of analyzing data

- Specifying a goal of analysis
- Collecting data with respect to the goal
- while(Fail to satisfy the goal)
  Do EDA(visualization, simple modeling, clustering…)
  Do Modeling
- Conclusion

Bias and Variance trade off

Bias-Variance trade off

- This is the basic concept of modeling.

- You MUST understand OVERFITTING and UNDERFITTING during this class.

- as well as BIAS and VARIANCE.

Bias-Variance trade off

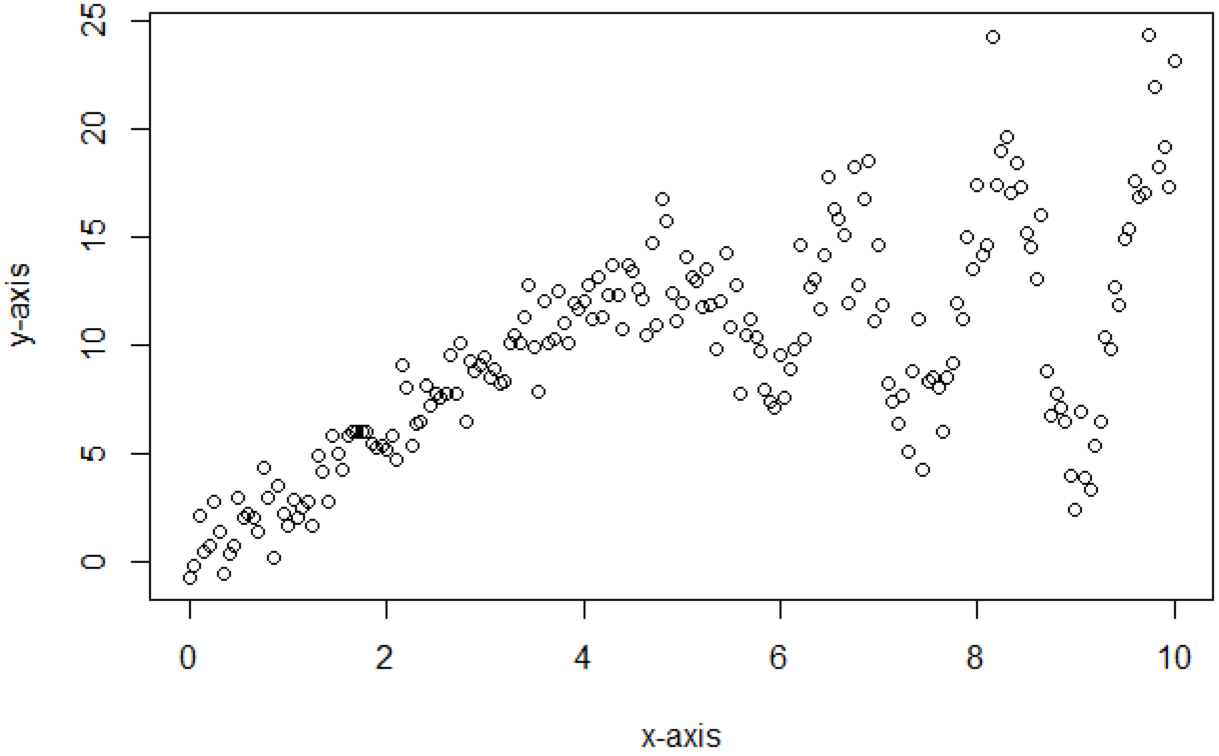
- sohnnn.tistory.com

https://sohnnn.tistory.com/entry/Bias-and-Variance-tradeoff-1?category=783791

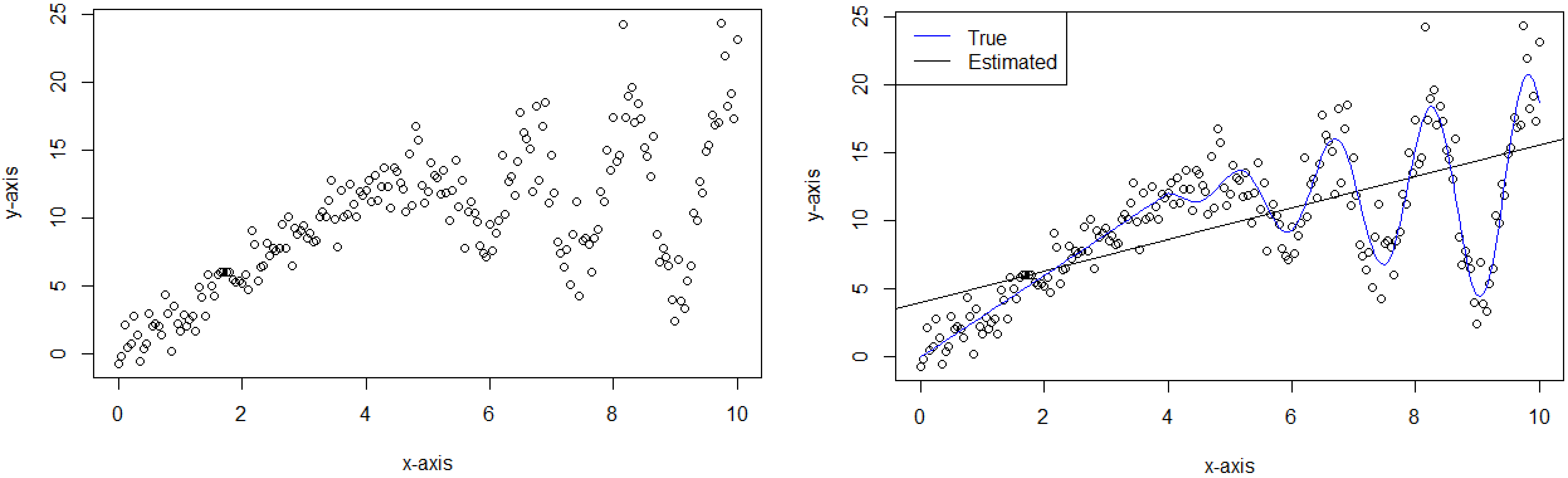
https://sohnnn.tistory.com/entry/Bias-and-Variance-tradeoff2?category=783791

Overfitting and Underfitting

- Considering following example

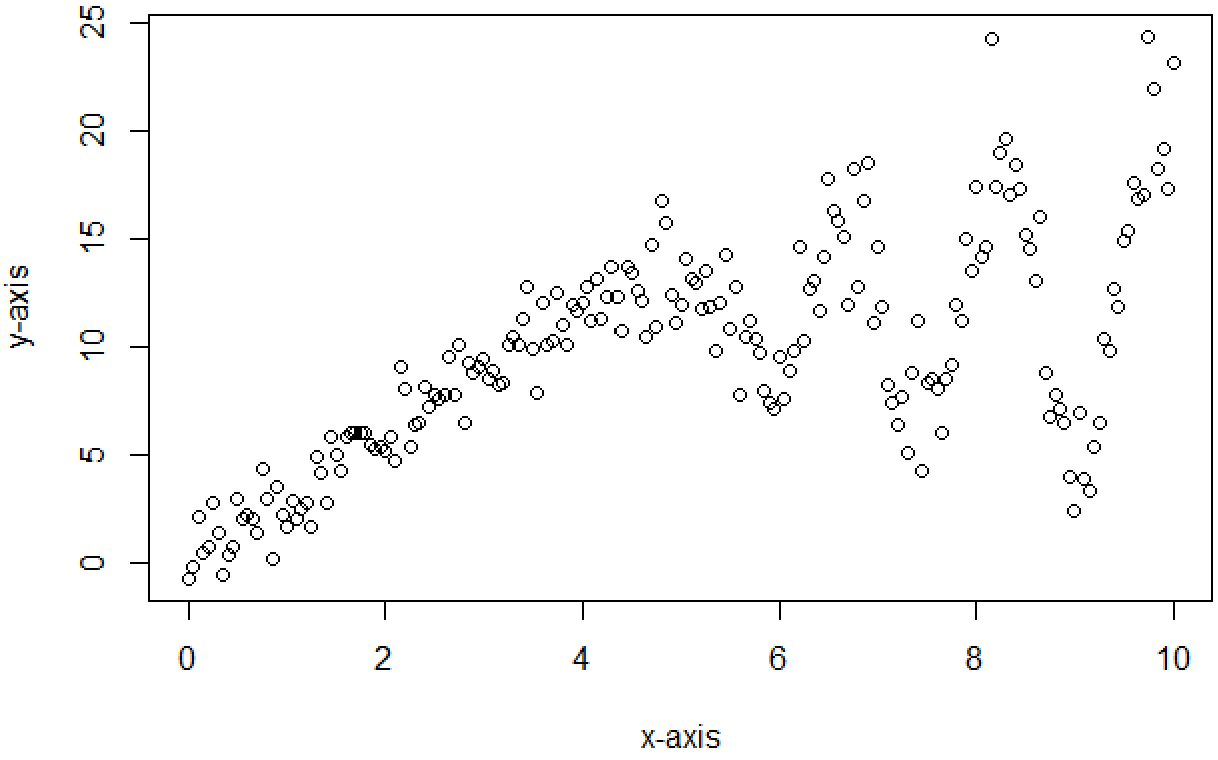


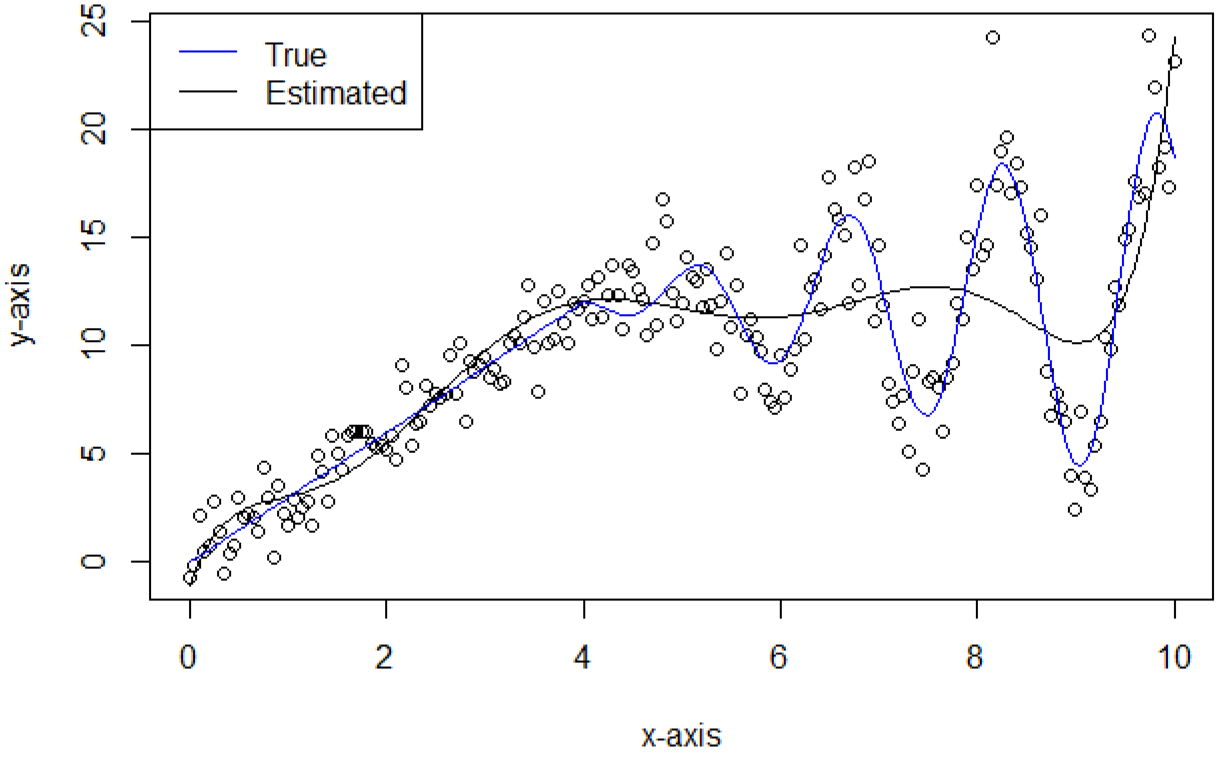
$$y = \beta_0 + \beta_1 x + \varepsilon$$



Too Biased!

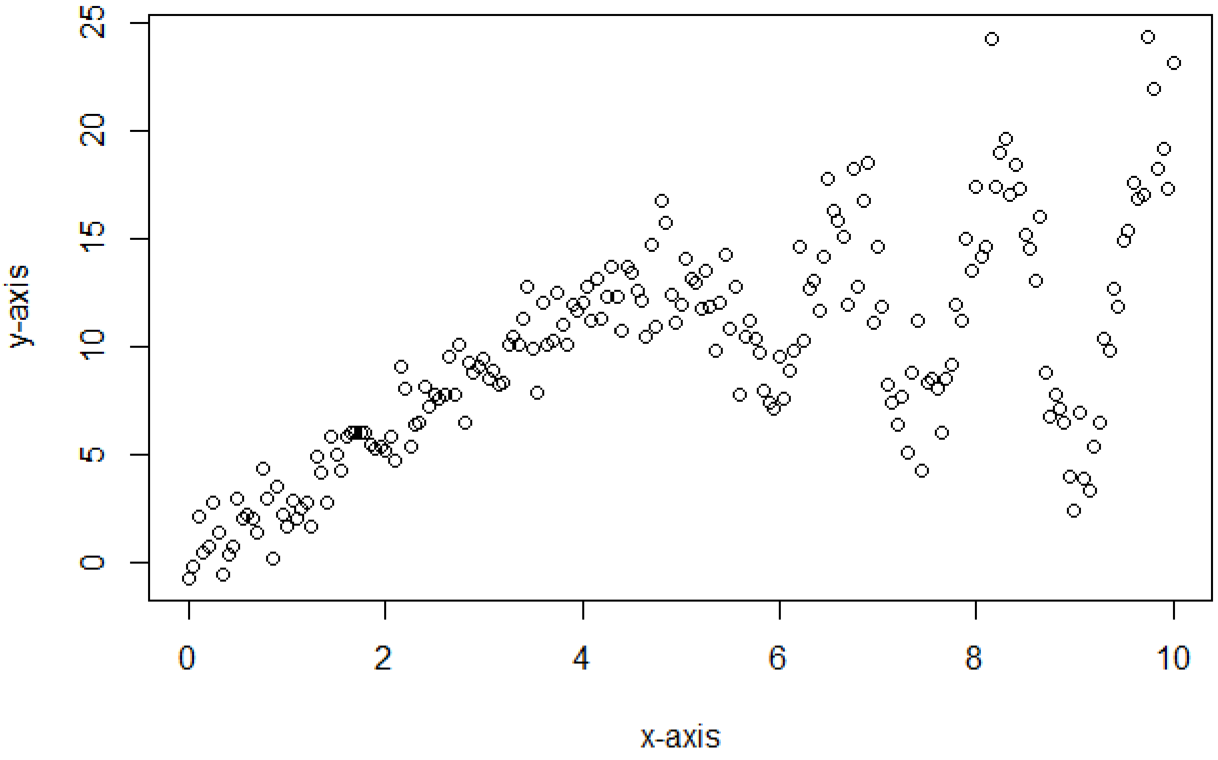
$$y = \beta_0 + \beta_1 x + \cdots + \beta_7 x^7 + \varepsilon$$

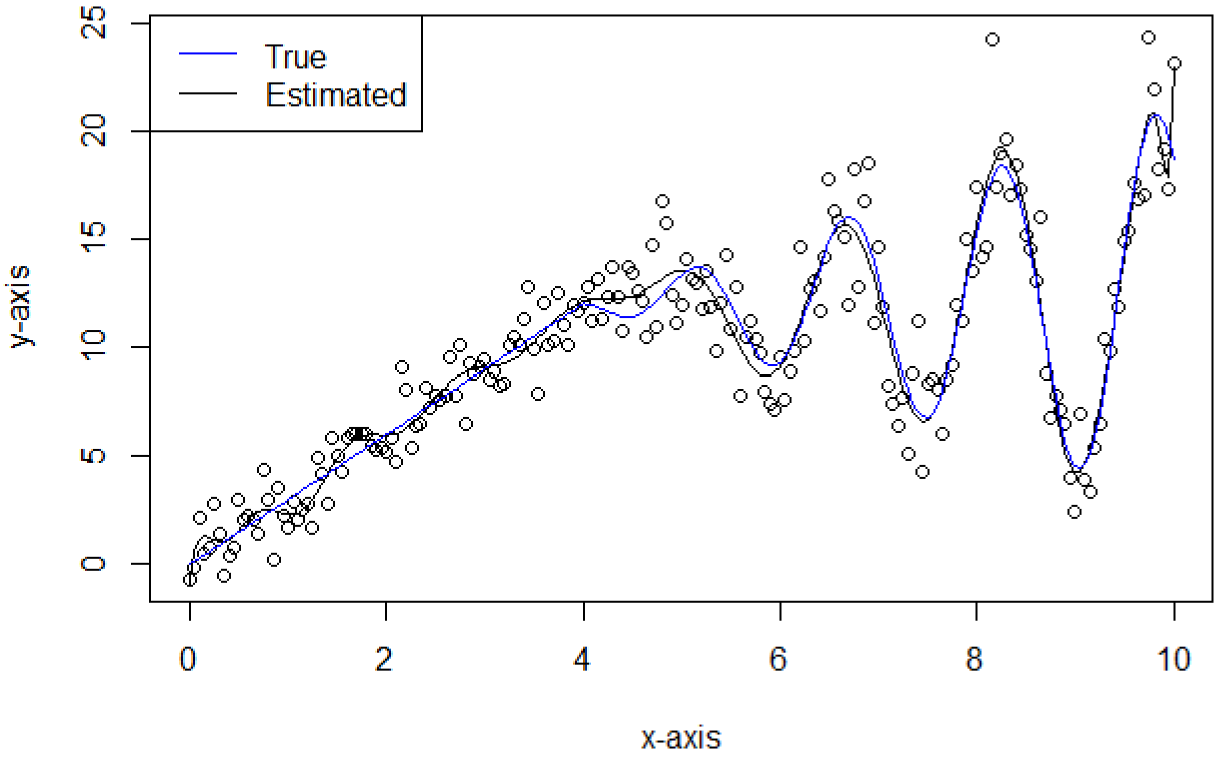




Still Biased…

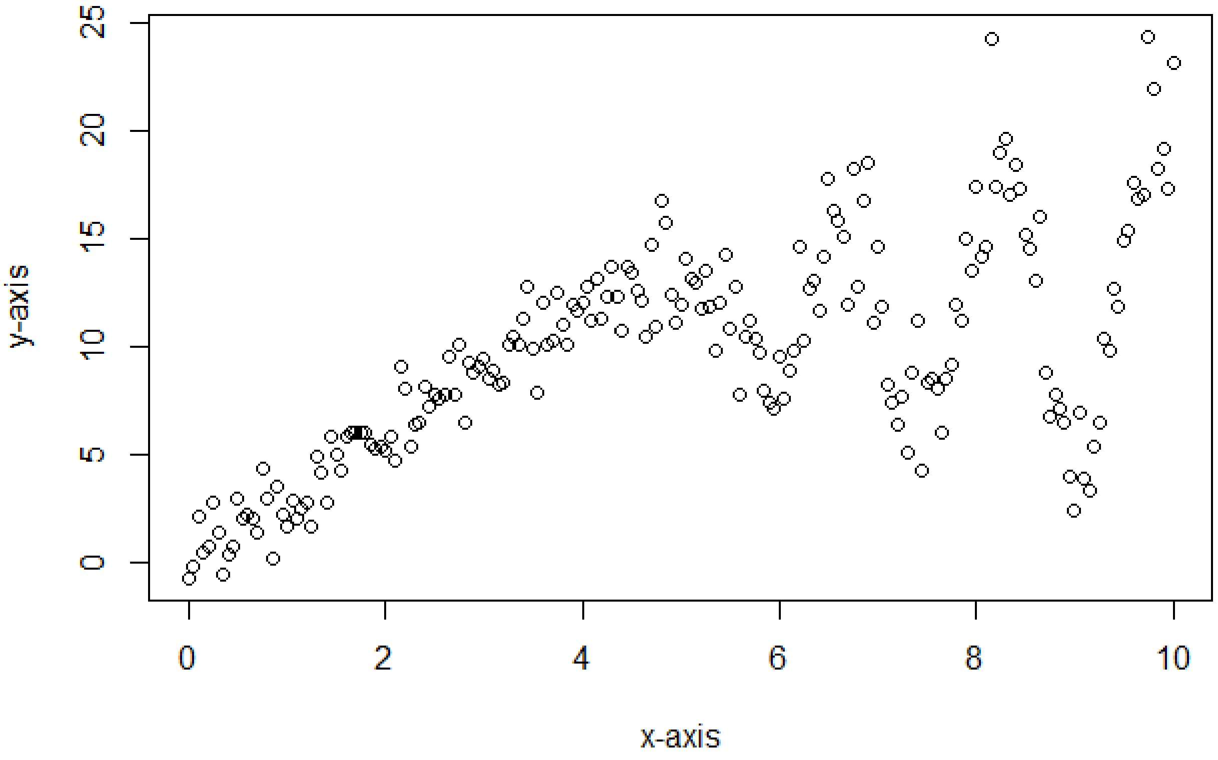
$$y = \beta_0 + \beta_1 x + \cdots + \beta_{25} x^{25} + \varepsilon$$

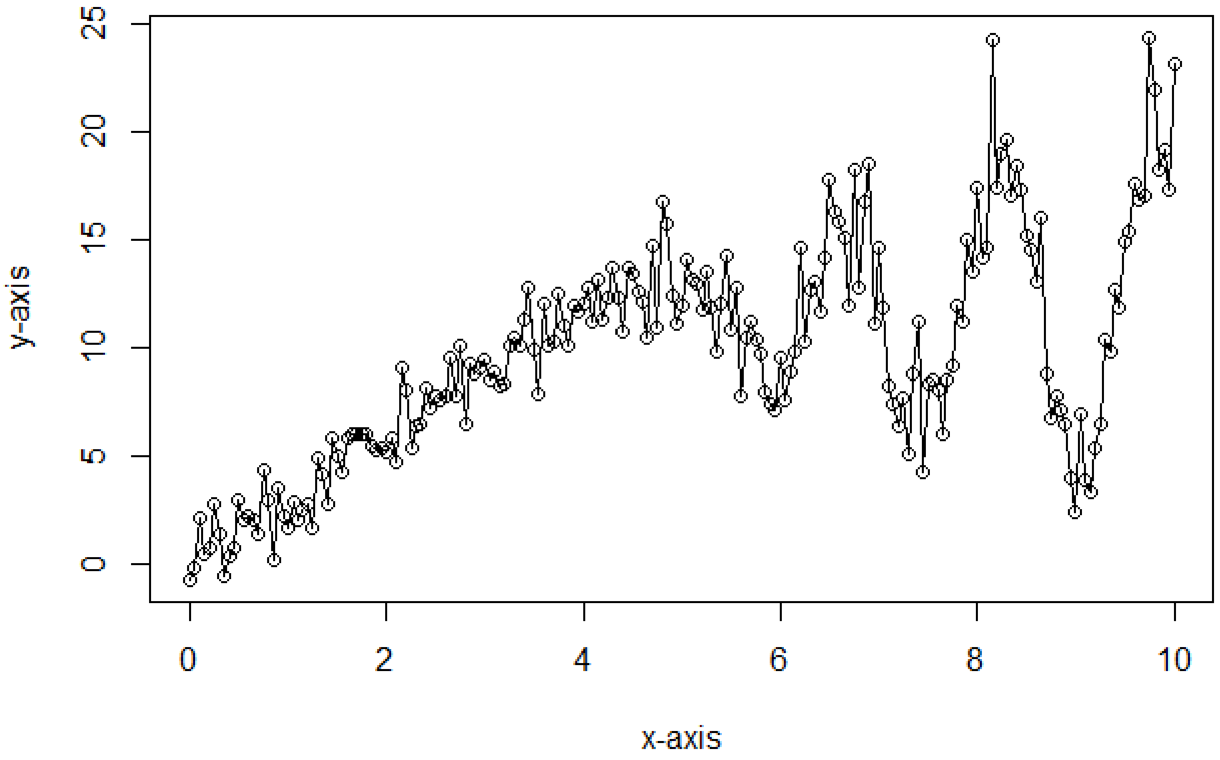




A bit Overfitted…

$$y = f_1(x) + \cdots + f_{200}(x) \qquad f_i(x) = y_i I(x = x_i)$$





Fully Overfitted!!

A (non)parametric model

A Parametric Model

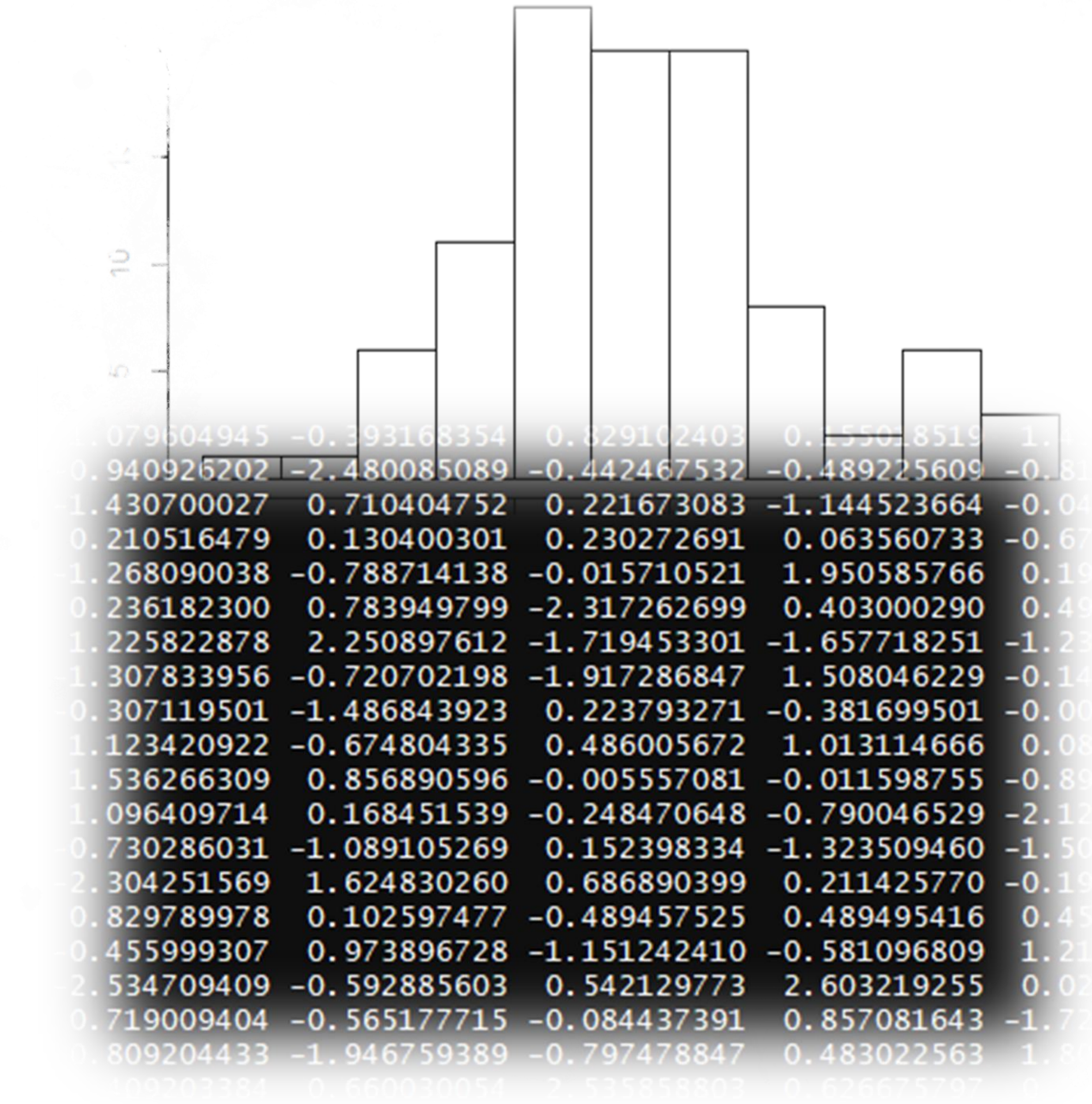
- Imagine data is given to you!

Index	Y	X
1	-0.68	0.00
2	0.95	0.05
3	0.82	0.10
…	…	…
199	18.77	9.90
200	17.68	9.95
201	18.53	10.00

“Your goal is to predict response Y with pre-specified finite parameters.”

What is meaning of ‘parametric’?

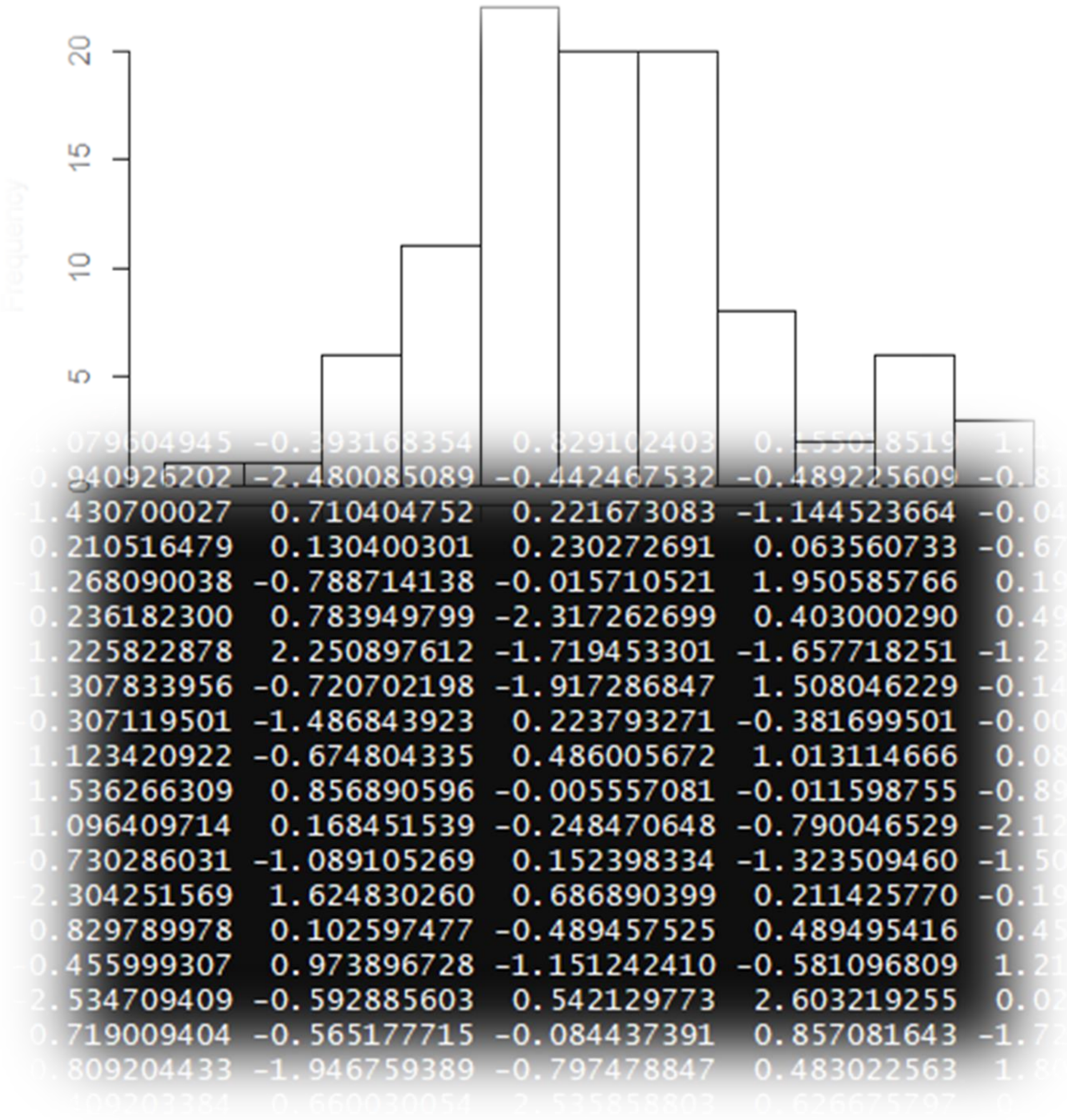
- Imagine normal distribution.



What is meaning of 'parametric'?

- Imagine normal distribution.

$$\mu \qquad \sigma^2$$



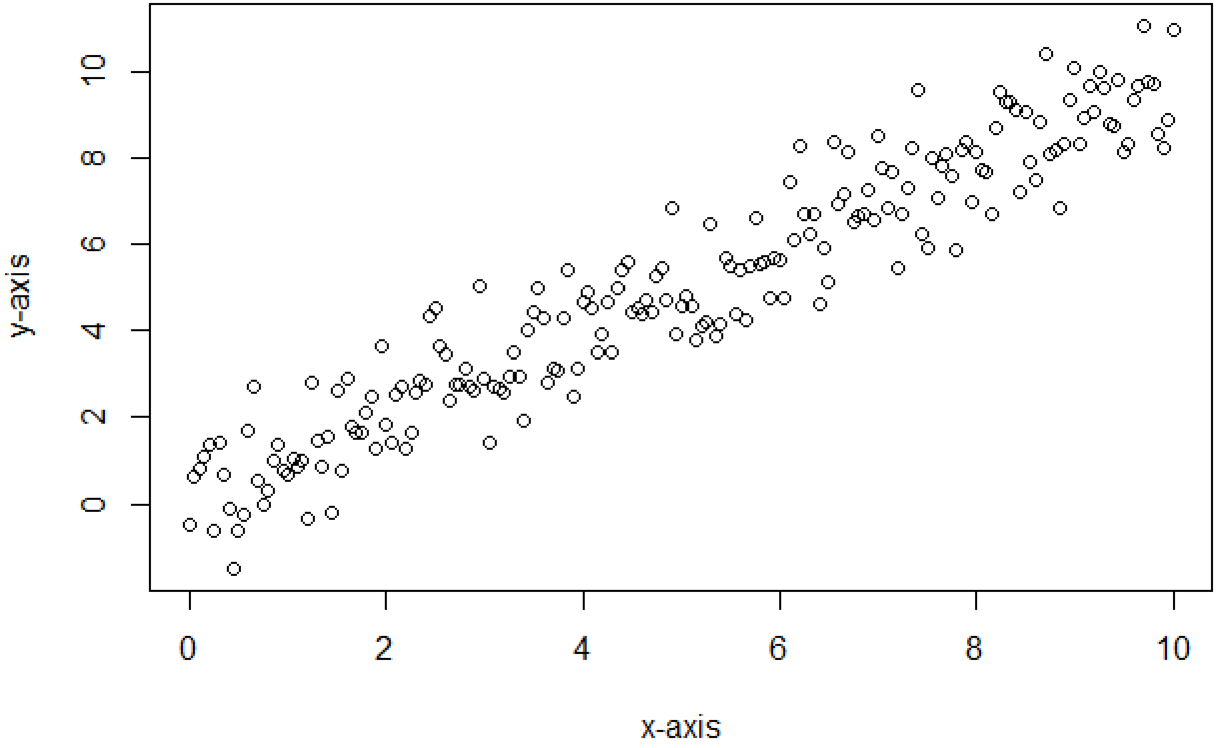
- Without loss of information

- Sufficient Statistics

What is meaning of ‘parametric’?

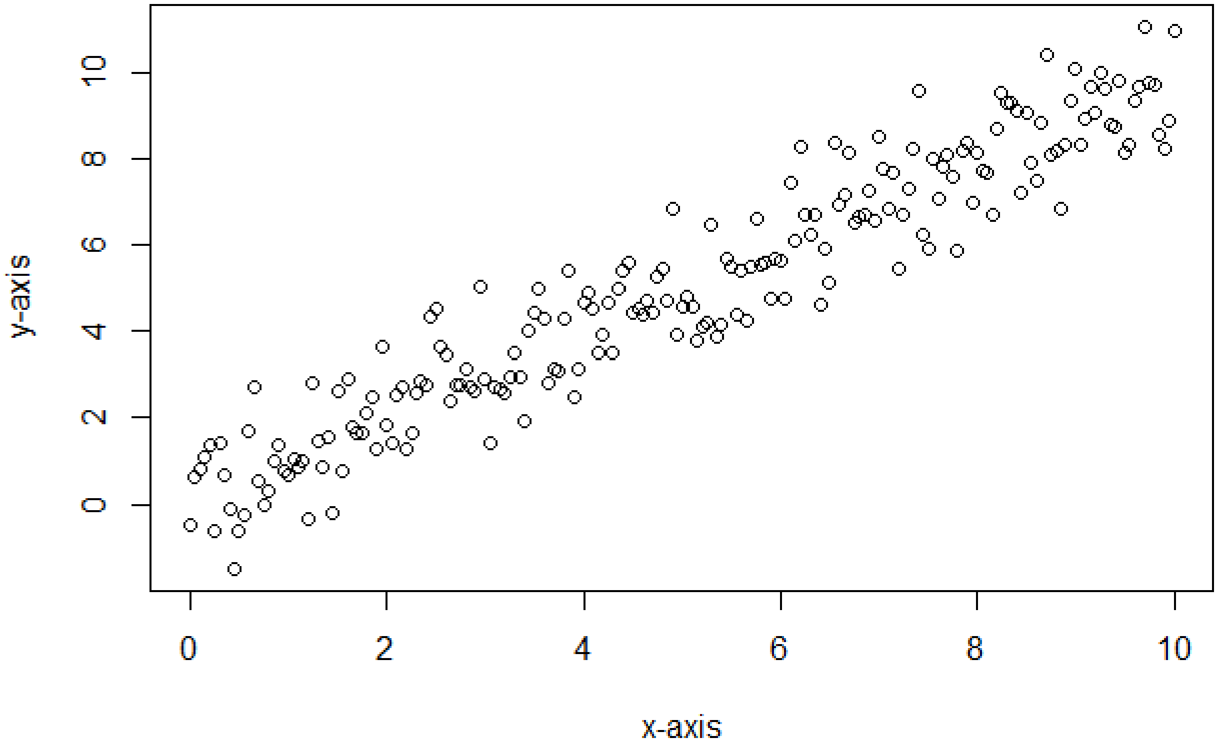
- Imagine simple linear model.

	x	sapply.x..function.x..f.x..
1	0.00	0.3564862
2	0.05	-1.7162925
3	0.10	-0.2283640
4	0.15	-1.0949264
5	0.20	0.9651375
6	0.25	0.5103146
196	9.75	9.727495
197	9.80	10.201826
198	9.85	10.863627
199	9.90	10.937694
200	9.95	9.810087
201	10.00	9.668022



What is meaning of ‘parametric’?

- Imagine simple linear model.

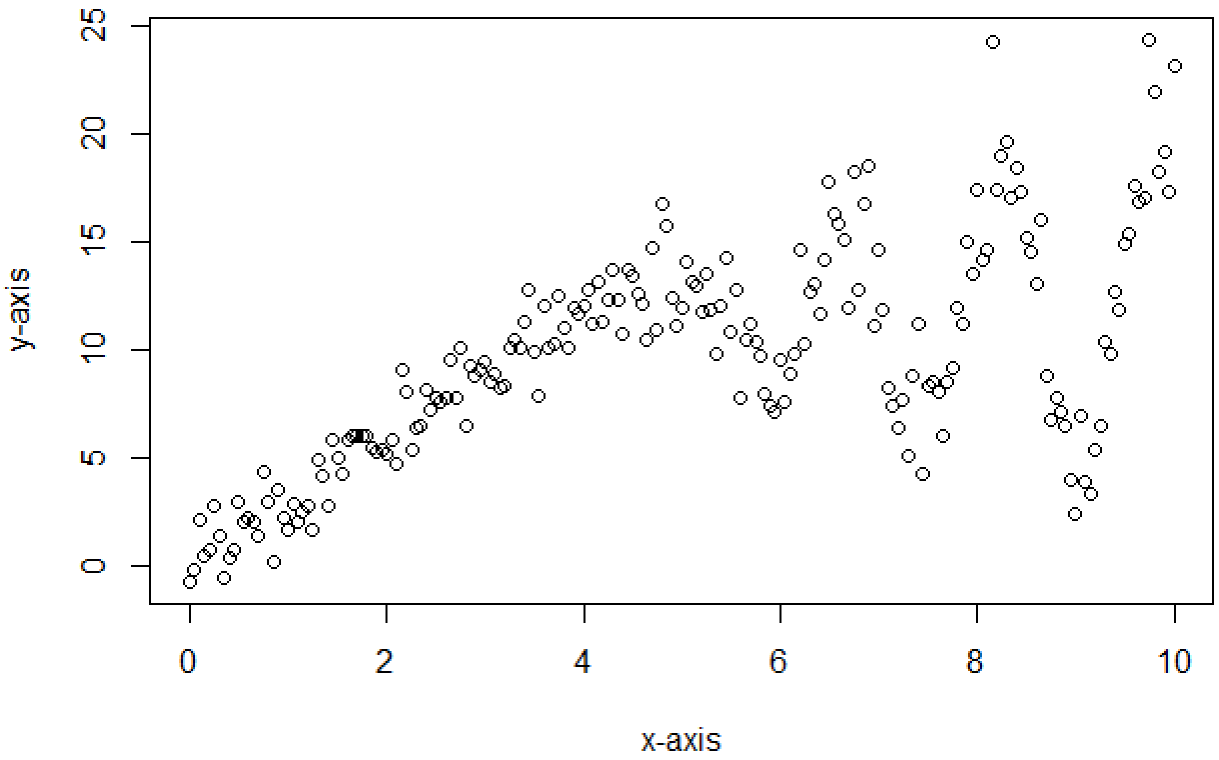


$$y = \beta_0 + \beta_1 x + \varepsilon$$

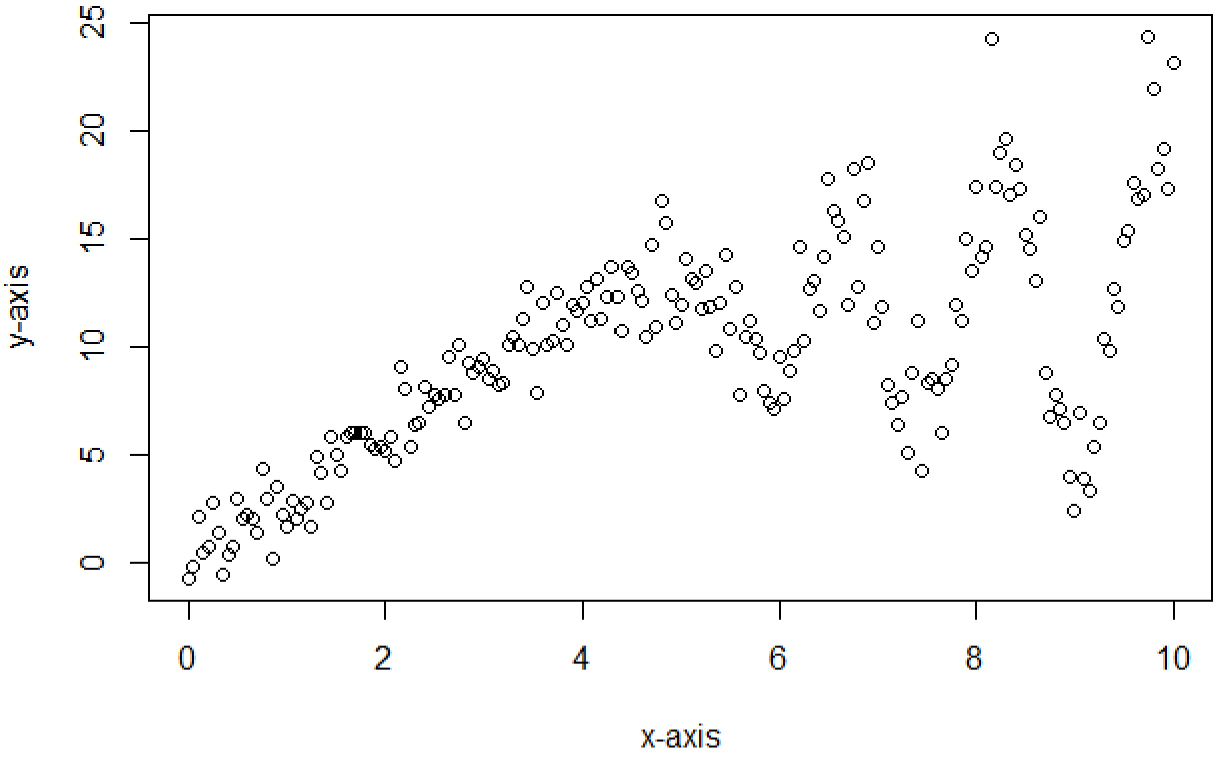
$$\beta_0 \quad \beta_1$$

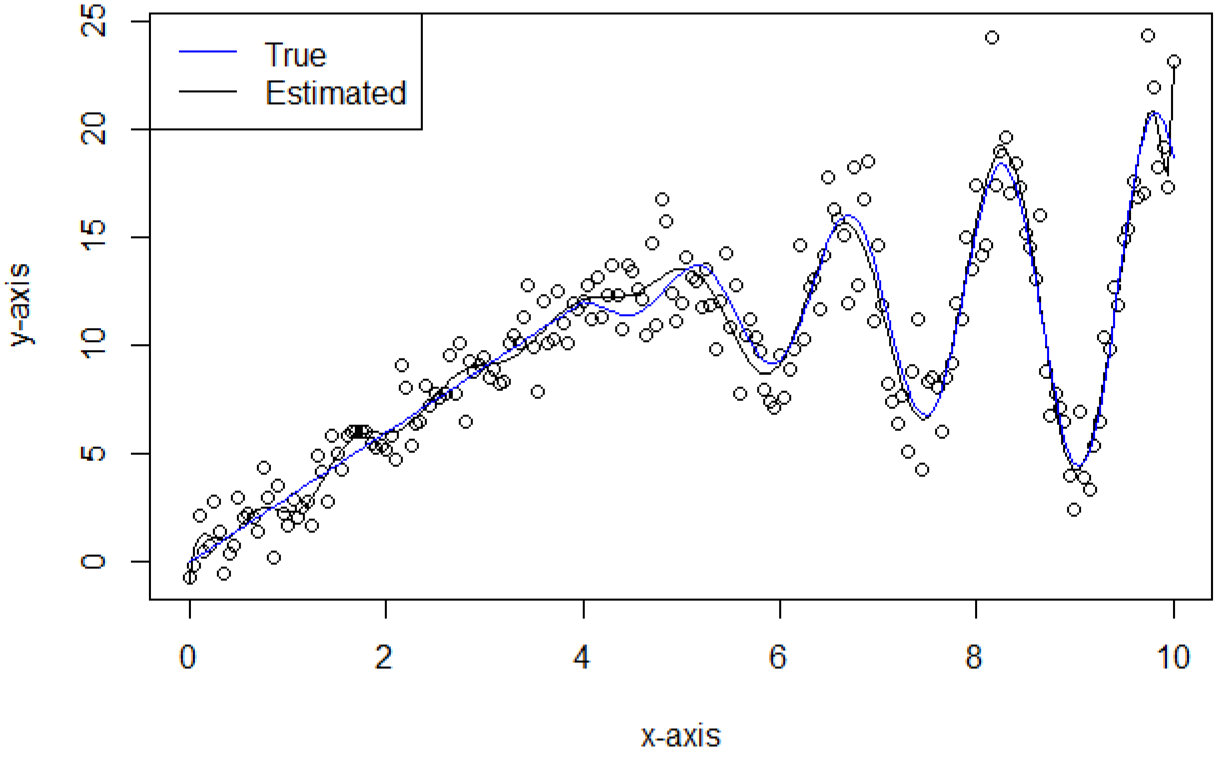
Capturing trend with parameters

- Considering previous example



$$y = \beta_0 + \beta_1 x + \cdots + \beta_{25} x^{25} + \varepsilon$$





The model seems to be plausible, but

A Parametric Model

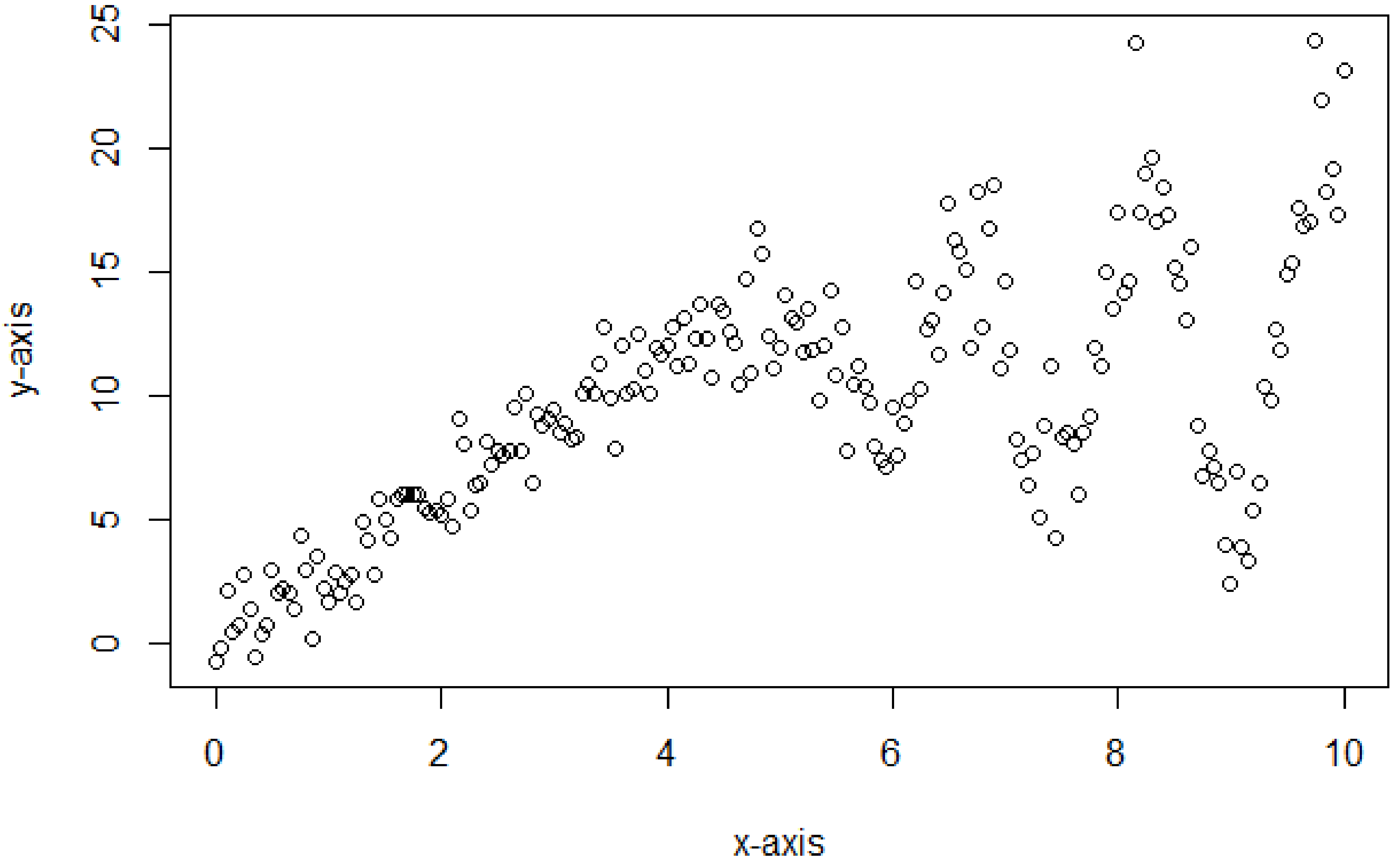
- can give us puzzled interpretations when a number of parameters are used although it has great prediction performance.

- can have the large variance of prediction by means of increasing the number of parameters.

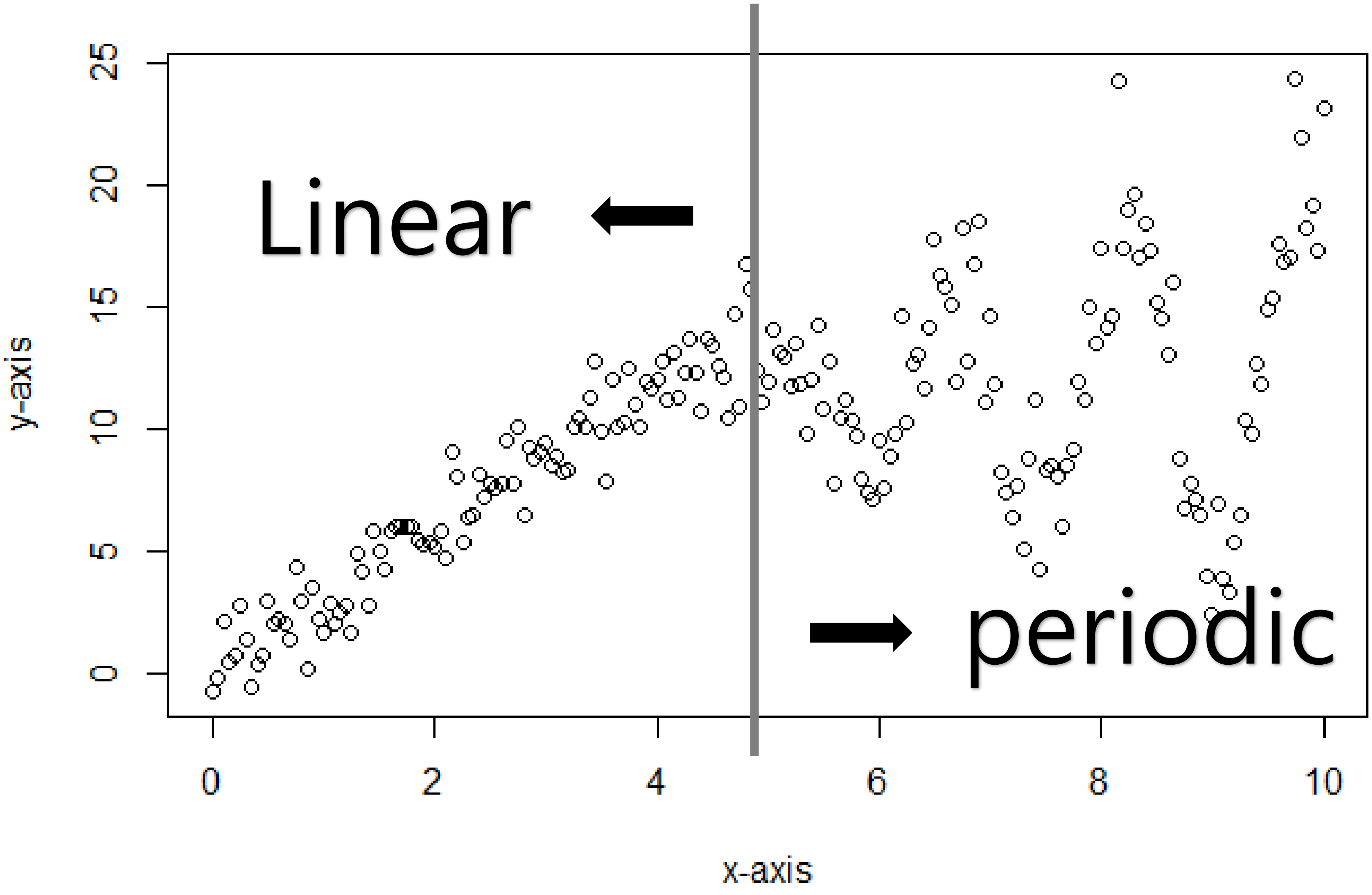
A Parametric Model

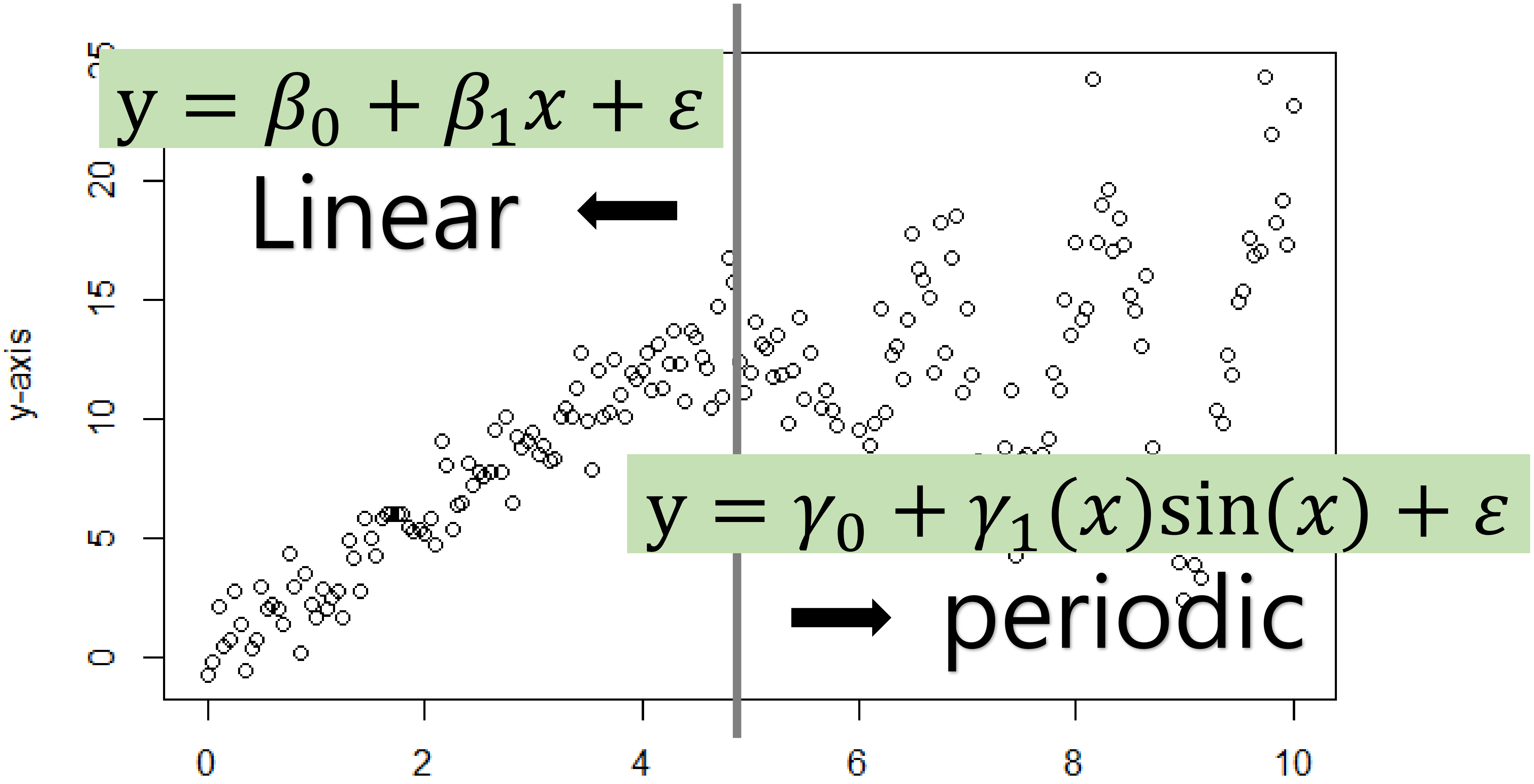
[Remedy]

- Find a parsimonious model.
- It can be partially achieved by finding proper features reducing the bias and variance.
- or by using more sophisticated parametric models.



0
5
10
15
20
25
y-axis
0
2
4
6
8
10
x-axis


Linear
periodic
y-axis
x-axis
0
5
10
15
20
25
0
2
4
6
8
10



Only four parameters are needed!

Raw data

Index	Y	X
1	-0.68	0.00
2	0.95	0.05
3	0.82	0.10
…	…	…
199	18.77	9.90
200	17.68	9.95
201	18.53	10.00

Feature engineering

Index	Y	X	I(X>4)	1.5(X-4)sin(X)
1	-0.68	0.00	0	0.000
2	0.95	0.05	0	-1.17
3	0.82	0.10	0	-2.27
…	…	…	…	…
199	18.77	9.90	1	8.37
200	17.68	9.95	1	7.70
201	18.53	10.00	1	6.70

A Parametric model

- So, basically, our aim is to estimate pre-specified and finite model parameters to capture a signal avoiding overfitting and underfitting.
- via a parsimonious model as much as possible.
  > Occam’s razor.
- This can be substantially and normally achieved through **newly introduced features,** such as nonlinearity and interaction between variables. “Heart-of-DataScience”

A Parametric model

- If you succeed in finding those features and fitting to a proper model, not only to obtain great prediction accuracy but also to make inference about parameters can be achieved.

- However, this is so time-consuming that I think data science is a kind of 3D jobs. Think about when there are many parameters more than a few thousand!

A Nonparametric Model

- When the data is too complex to be captured with finite number of parameters.

- When to find desired features is almost impossible.
  > ex) 1.5(X-4)sin(X).

A Nonparametric Model

- Unlike the parametric model, a nonparametric model assumes that there are infinite number of parameters, which implies that

- the model complexity grows with respect to the size of data,

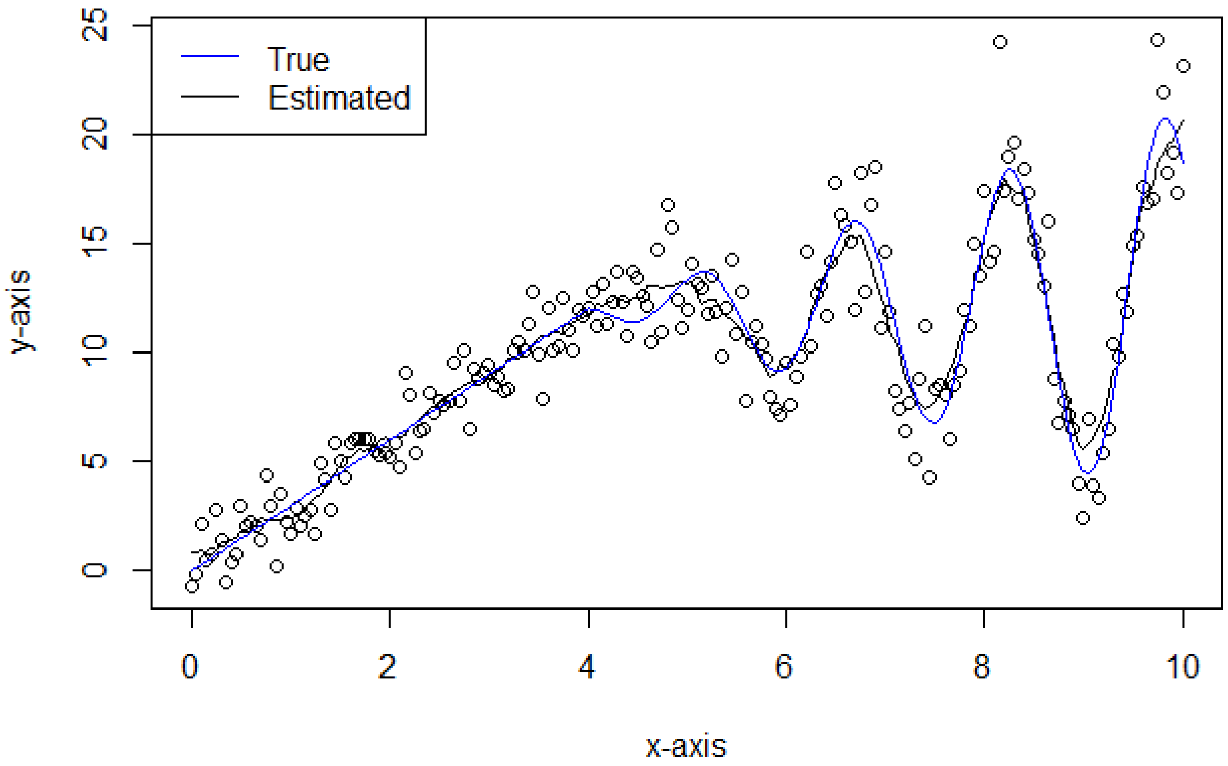
- and a structure of the model generating data is not fixed a priori.

- but, the structure will be determined from the data instead.

Kernel smoothing

$$K_h(x_{new}, x_i) = e^{-\left(\frac{x_{new}-x_i}{\sqrt{2}h}\right)^2}$$

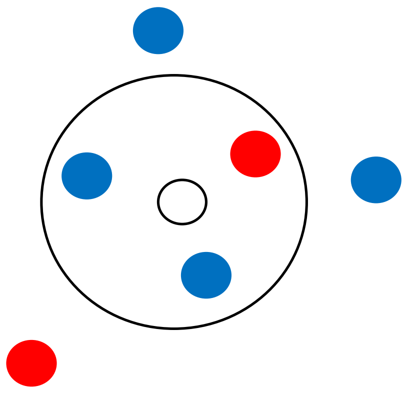
$$\hat{y}_{new} = \sum_{i=1}^{N} \frac{K_h(x_{new}, x_i)}{\sum_{i=1}^{N} K_h(x_{new}, x_i)} y_i$$



- Kernel function represents the distance between two points.

- Imagine when h goes up or down.

K nearest neighborhood



- The empty circle will be filled with either blue or red color along with K.
- In this case, K=3.
- Euclidian distance, Mahalanobis distance can be chosen.

A Nonparametric Model

- Highly flexible model that better performances can be achieved.
- Requires too much computation along with the size of data.
- Generally, more difficult to distinguish the effects of the variables than the parametric model.
- Need to tune hyperparameter in some models, such as KNN.

Tuning parameters

Parameters

- Parameters : to be estimated (almost uniquely) with data, such as coefficients of regression, split points in tree, etc.

- Hyperparameters : can not be estimated with data. We have to specify proper values before constructing a model. It is also called as tuning parameters.

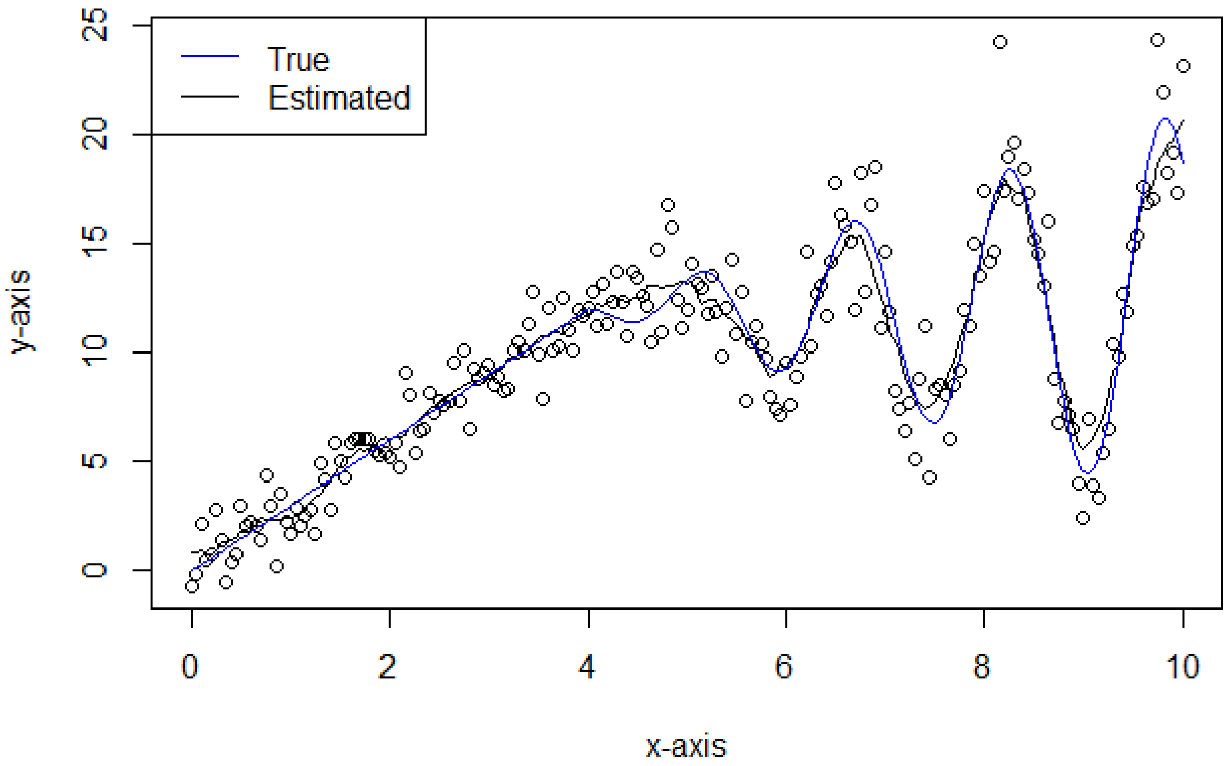
Tuning parameters

- Consider the previous kernel smoothing example.
- There is one tuning parameter, the bandwidth h.
- Simply, we can find the best tuning parameter, **using cross validation**.

Kernel smoothing

$$K_h(x_{new}, x_i) = e^{-\left(\frac{x_{new}-x_i}{\sqrt{2}h}\right)^2}$$

$$\hat{y}_{new} = \sum_{i=1}^{N} \frac{K_h(x_{new}, x_i)}{\sum_{i=1}^{N} K_h(x_{new}, x_i)} y_i$$



- Kernel function represents the distance between two points.

- Imagine when h goes up or down.

K-fold cross-validation(K-CV)

- In cross-validation, we trust. <Kaggler>.
- Measuring performance

Train data

1	2	3	4	5

5-fold →

Iter 1 : train with : 2, 3, 4, 5 // validate with 1

Iter 2 : train with : 1, 3, 4, 5 // validate with 2

....

Iter 5 : train with : 1, 2, 3, 4 // validate with 5

→ Average

Leave One Out Cross Validation(LOOCV)

- When the size of data is big, it is computationally beneficial to make use of K-fold cross-validation.

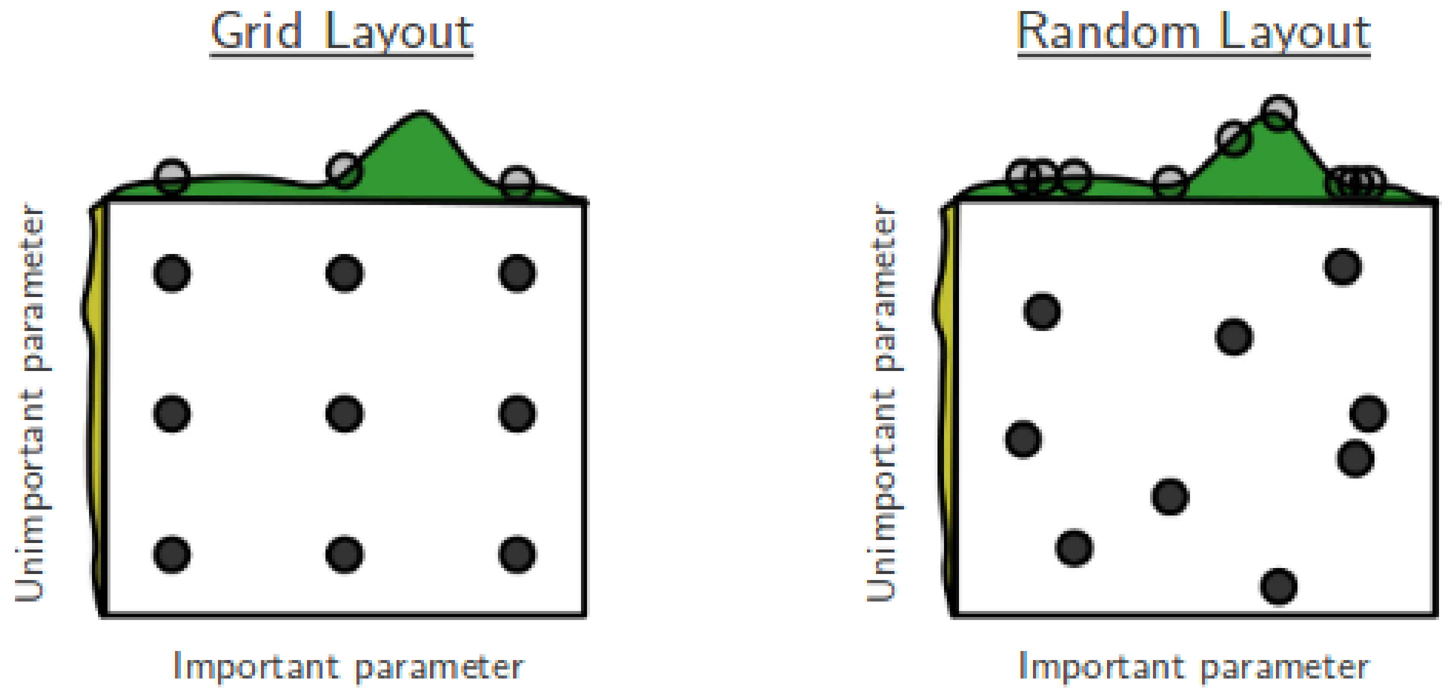
- But, if you have small data, or the model requires only trivial computations, LOOCV might give you a plausible value.

- If you specify K as N, the number of observations, then N-fold, LOOCV, will be made.

Grid search vs Random search

- There are many tuning parameters. For examples, learning rate of neural network, sub-sampling ratio and the depth of tree of the tree based boosting, and bandwidth of kernel smoothing need to be adjusted.
- Unfortunately, performances of models heavily depend on the tuning parameters; therefore, we have to find the optimal parameter among a list of the parameters.
- K-fold cross-validation is a major gadget for comparing the parameters in the list.

Grid search vs Random search



Data type

Given mixed data

- Many models have been developed for continuous predictors and responses.

- PCA, FA, K-means are only available for continuous data.

- If nominal type categorical predictors are in data, you should transform the categorical predictors into plausibly continuous data, or ordinal data.

- Or, you can search through google!(highly recommended)

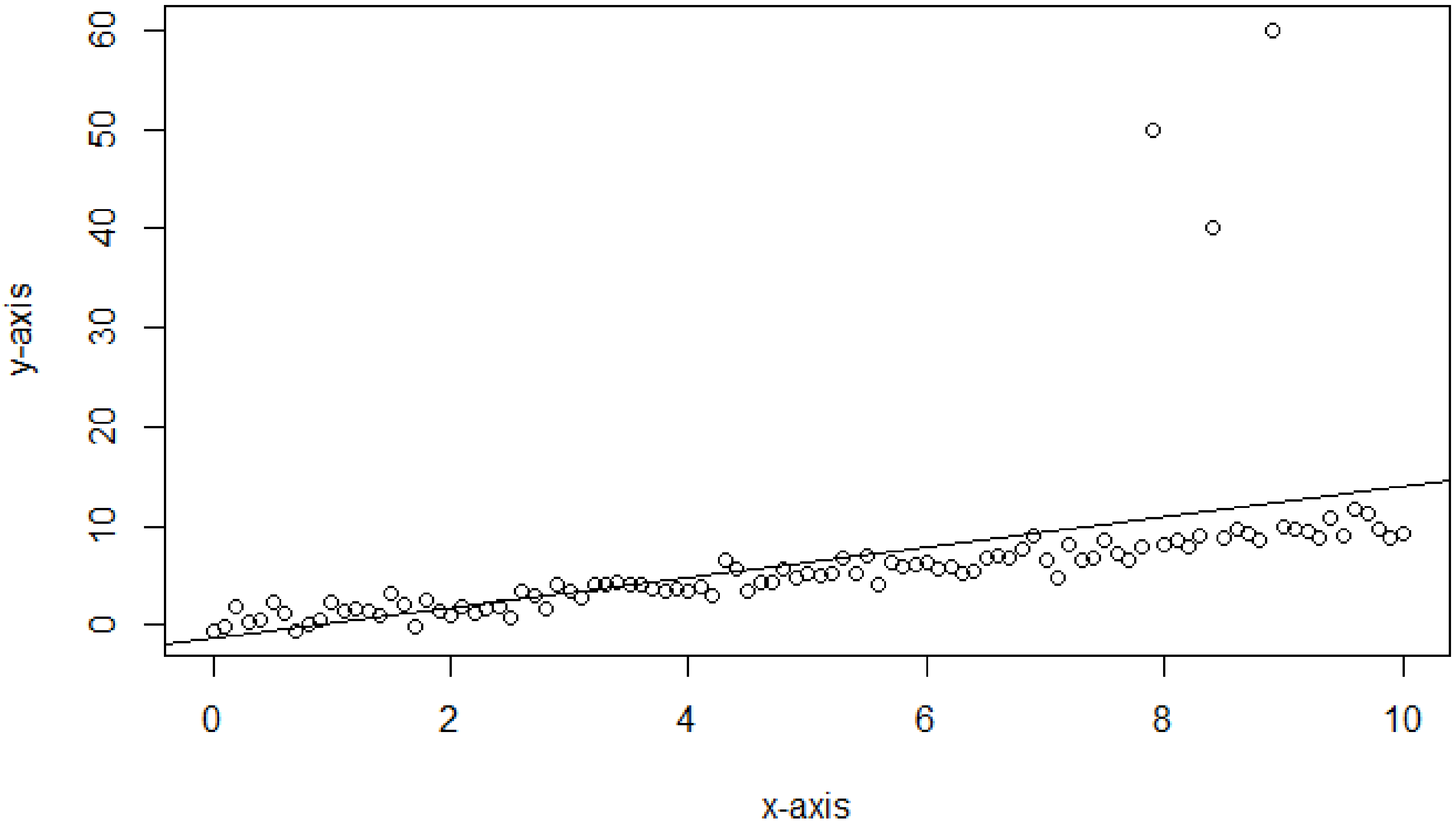
Outliers

- Models based on ordinary least square can not be free from outliers.

- K-means, PCA, FA can be examples.

- In those cases, you have to find more robust models than the OLS based model., such as Robust K-means, Robust PCA.

Outliers



Sparse data

- ‘Sparsity’ denotes that too many zeros are included in data.
- You should try to decrease undesired effects caused by sparse data.
- Most of the models to handle sparsity data are developed already like Sparse logistic regression, sparse PCA.

Sparse data

Index	Y	X1	X2	X3
1	-0.68	0	0	1
2	0.95	0.05	0	0
3	0.82	0	0	0
…	…	…	…	…
199	18.77	0	0	20
200	17.68	9.95	0	0
201	18.53	10.00	1	0

Too many zeros

Imbalanced data

Zero-inflated data

- When too many zeros exist in data, it is better to use models which are developed to deal with zero-inflated structure.

- For example, zero-inflated poisson regression, zero-inflated factor analysis, zero-inflated logistic regression exist.

Imbalanced class

- Let’s consider a binary classification problem.

- While doing EDA, we see that 95% of subjects belongs to ‘Innocent’, and only 5% to ‘Fraud’.

- In this case, we call this ‘imbalanced class problem’.

- If a model you consider ignores this imbalanced class, the model might become to have poor performance.

Imbalanced class

[Remedy]

- In perspective of data : up sampling, down sampling, SMOTE, …

- In perspective of model : imposing more weight on minority class. Adjusting a threshold.

- In perspective of measure : Using proper metric to measure performance, such as f1-score.

Imbalanced class

A / P	Innocent	Faud
Innocent	800	100
Fraud	50	50

- Accuracy : 85%

- Precision : 33%
- Recall : 50%

- F1 score : 40%
  > harmonic mean of precision and recall

Summary

- First of all, by doing EDA, you have to understand data thoroughly.

- Finding a proper models to catch a structure what data shows. Almost all of models you want can be found in Google.

Summary

- Most of the problems you will encounter are either causal inference or prediction.
- You must look for the best model to avoid overfitting and underfitting with appropriate feature engineering.
- If you are able to estimate pre-specified parameters well, the parametric model would outperform the nonparametric model in view of prediction.

Questions…

(Q) How does a Deep Neural Network model reach great prediction accuracy?

(Q) Why overfitting is a principal concern in DNN based models

(Q) Isn’t it unnecessary to consider new features when we decide to use a nonparametric model?

(Q) What is the difference between multiple linear regression and regression tree in terms of interaction?

Questions…

(Q) How to decide the number of fold in K-fold cross-validation when data is too big or has highly imbalanced label?

(Q) How to measure performances when the computation is so burdensome for fitting a model actually? For instance, imagine that fitting a model with a set of parameters require the day.