

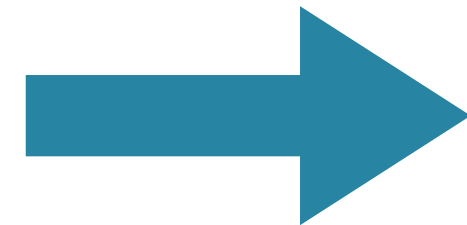


INTRODUCTION TO MACHINE LEARNING

# **Regression: Simple and Linear**

# Regression Principle

PREDICTORS



REGRESSION



RESPONSE

# Example

Shop Data: sales, competition, district size, ...

Data Analyst       Relationship?

- Predictors: competition, advertisement, ...
- Response: sales

Shopkeeper       Predictions!

# Simple Linear Regression

- Simple: one *predictor* to model the *response*
- Linear: *approximately* linear relationship

Linearity Plausible?



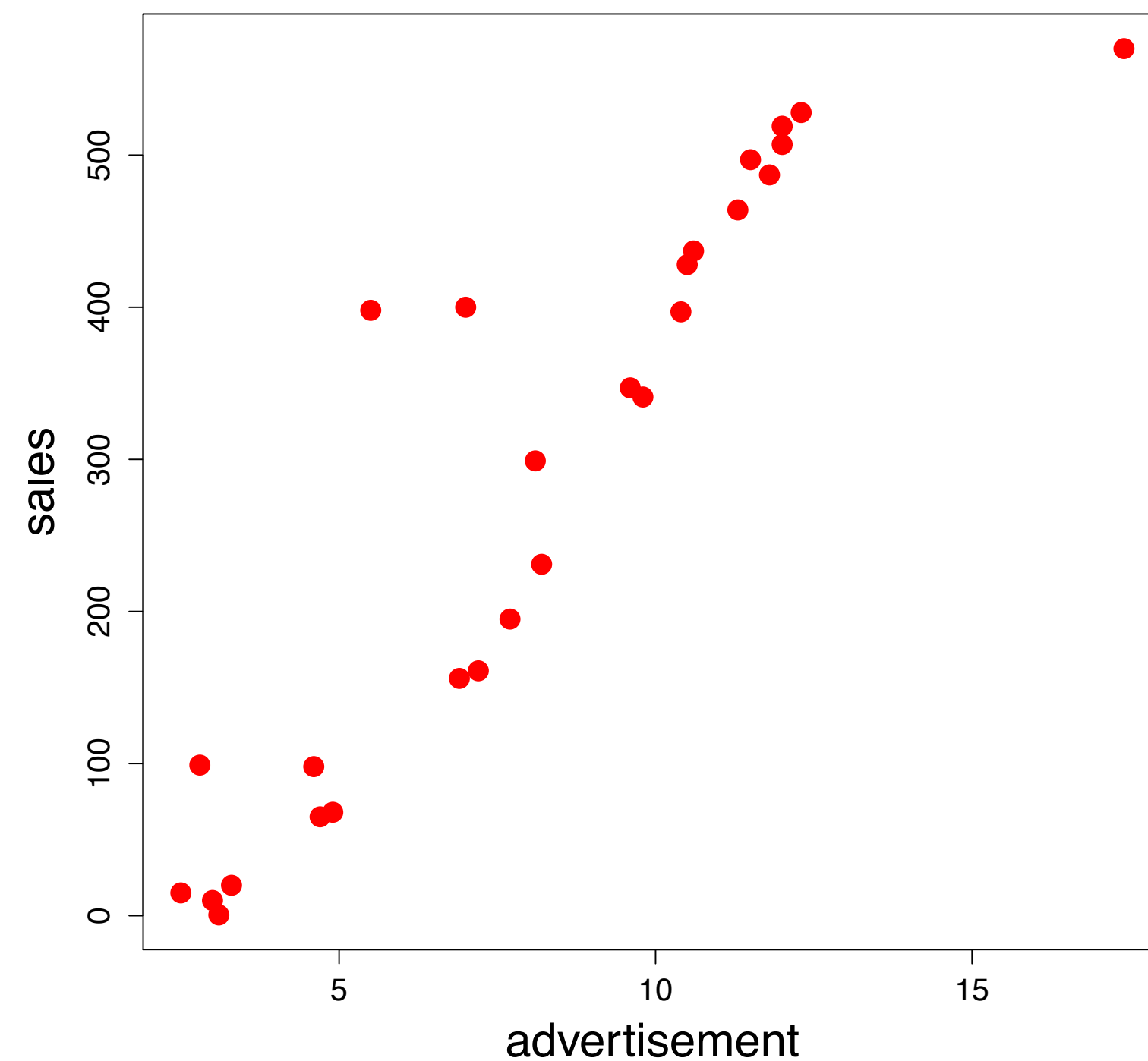
Scatterplot!

# Example

- Relationship: advertisement  $\longrightarrow$  sales
- Expectation: positively correlated

# Example

- **Observation:** upwards linear trend
- **First Step:** simple linear regression



# Model

Fitting a line

$$Y = \beta_0 + X\beta_1 + \epsilon$$

- Predictor:  $X$
- Response:  $Y$
- Statistical Error:  $\epsilon$
- Intercept:  $\beta_0$
- Slope:  $\beta_1$

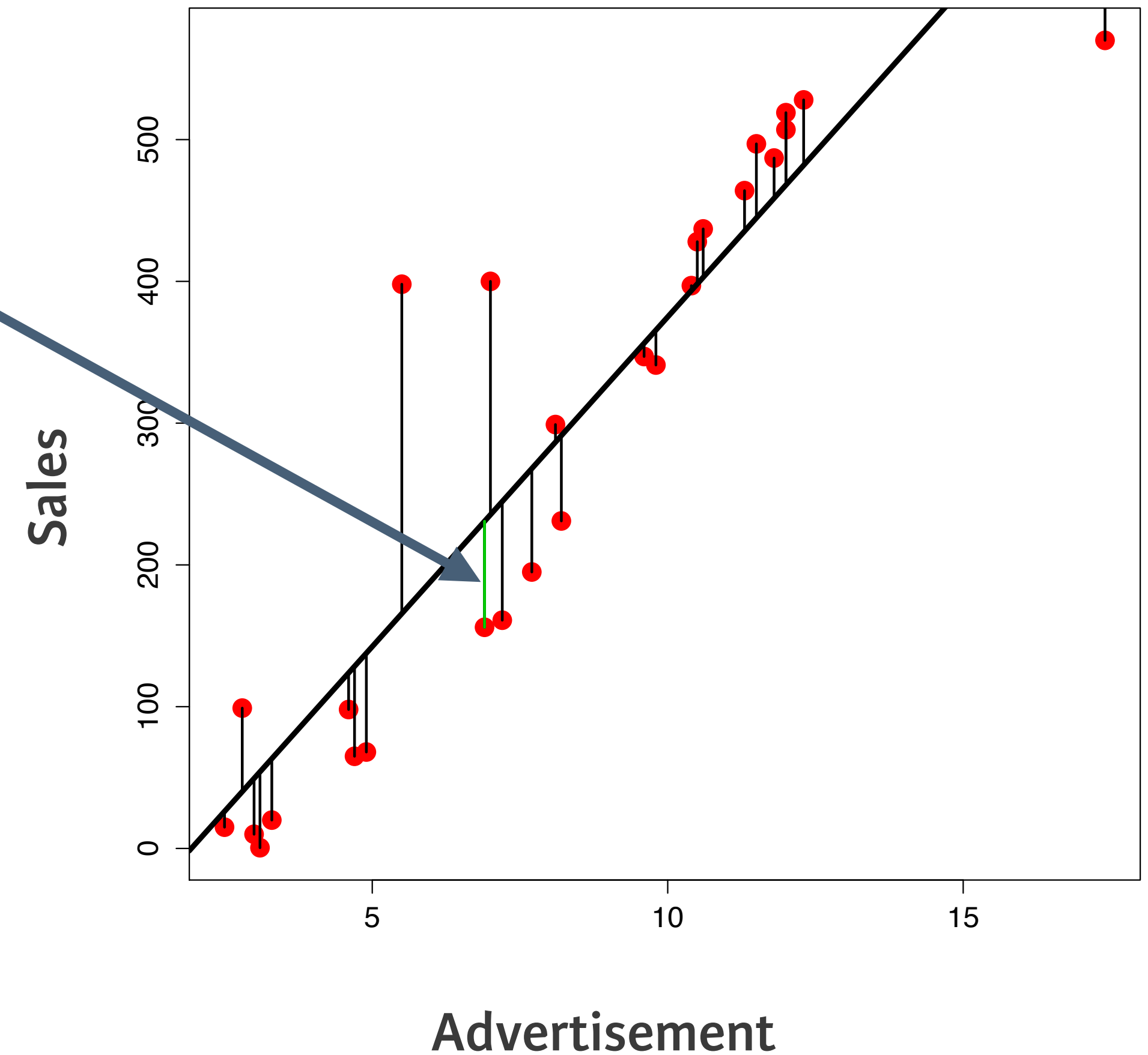
# Estimating Coefficients

$$\text{res}_i = y_i - (\beta_0 + \beta_1 x_i)$$

True Response      Fitted Response      Residuals

$$SS_{\text{res}} = \sum_{i=1}^N \text{res}_i^2$$

#Observations      Minimize!





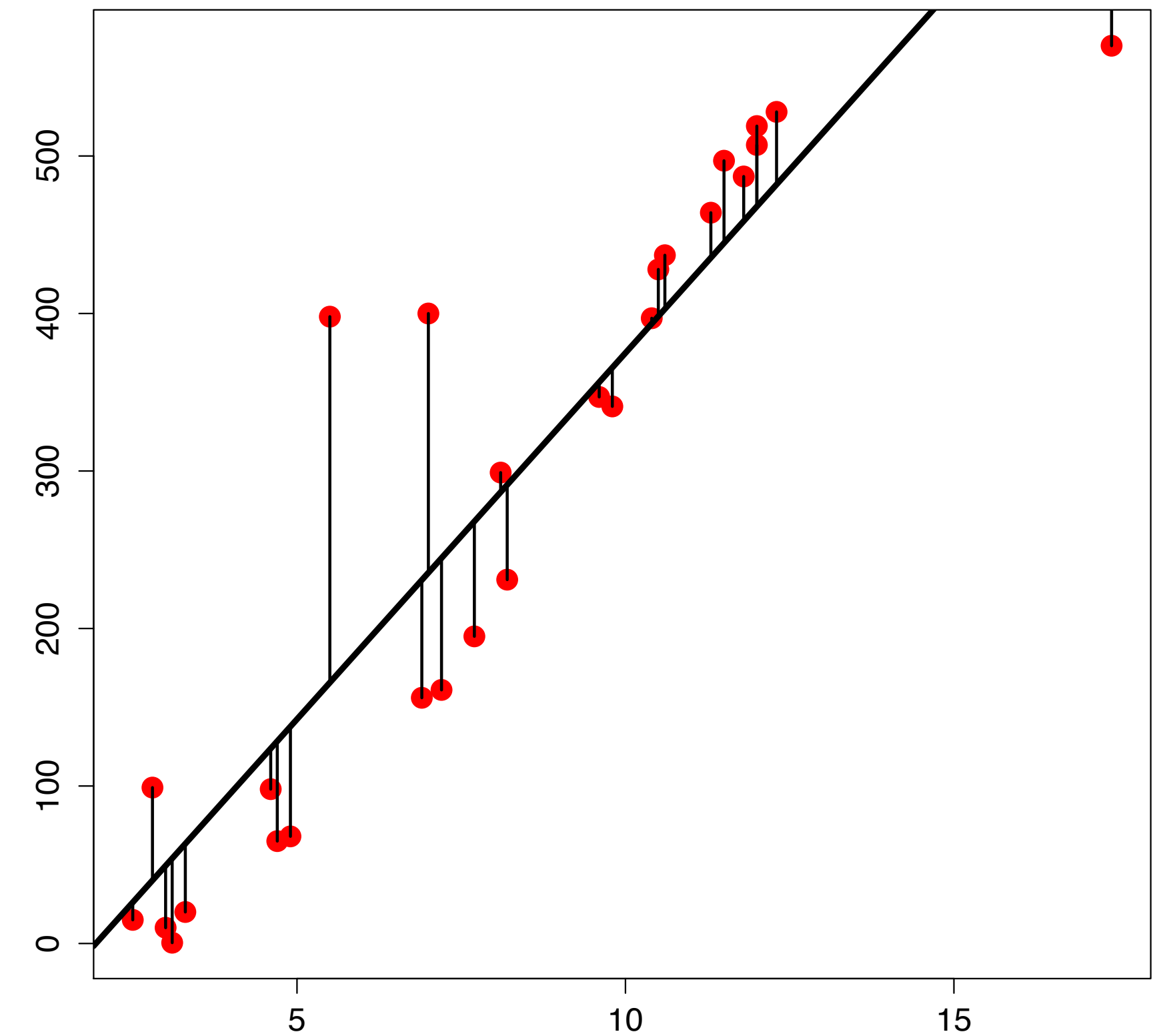
# Estimating Coefficients

**Response**      **Predictor**

```
> my_lm <- lm(sales ~ ads, data = shop_data)
```

```
> my_lm$coefficients
```

**Returns coefficients**



# Prediction with Regression

Predicting new outcomes

$$\hat{y}_{new} := \hat{\beta}_0 + \hat{\beta}_1 x_{new}$$

$\hat{\beta}_0, \hat{\beta}_1$  ← Estimated Coefficients

$x_{new}$  ← New Predictor Instance

$\hat{y}_{new}$  ← Estimated Response

Example: Ads: 11.000\$ → Sales: 380.000\$

Must be data frame

```
> y_new <- predict(my_lm, x_new, interval = "confidence")
```

Provides confidence interval

# Accuracy: RMSE

Measure of accuracy:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}$$

# Observations

True Response

Estimated Response

Example:  $\text{RMSE} = 76.000\$$  → Meaning?

RMSE has unit + scale → difficult to interpret!

# Accuracy: R-squared

$$SS_{\text{res}} = \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

$$SS_{\text{tot}} = \sum_{i=1}^N (y_i - \bar{y})^2$$

Sample mean response

Total SS

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

R-squared

Interpretation: % explained variance,

$R^2$  close to 1 → good fit!

```
> summary(my_lm)$r.squared
```

→ Example: 0.84



## INTRODUCTION TO MACHINE LEARNING

# Let's practice!



INTRODUCTION TO MACHINE LEARNING

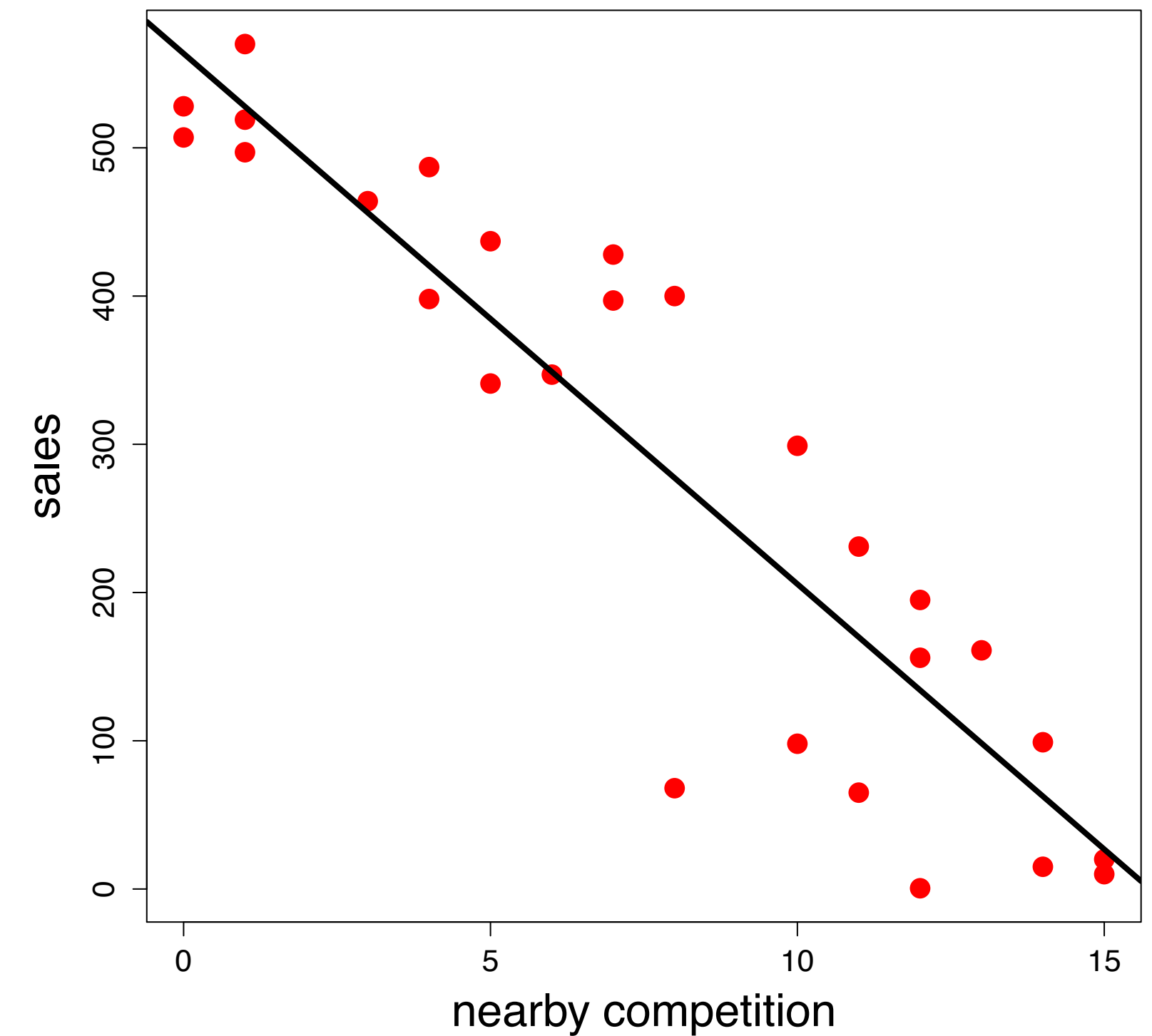
# **Multivariable Linear Regression**

# Example

## Simple Linear Regression:

```
> lm(sales ~ ads, data = shop_data)
> lm(sales ~ comp, data = shop_data)
```

Loss of information!



# Multi-Linear Model

Solution: combine in multi linear model!

- Higher predictive power
- Higher accuracy

$$\text{Sales} = \beta_0 + \beta_1 \times \text{Competition} + \beta_2 \times \text{Advertisement} + \epsilon$$

Individual Effect





# Multi-Linear Regression Model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

- Predictors:  $X_1, X_2$
- Response:  $Y$
- Statistical Error:  $\epsilon$
- Coefficients:  $\beta_0, \beta_1, \beta_2$

# Estimating Coefficients

$$res_i = y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}) \quad \leftarrow \text{Residuals}$$

$\uparrow$  True Response       $\uparrow$  Fitted Response

$$SS_{res} = \sum_{i=1}^n res_i^2 \quad \leftarrow \text{Minimize!}$$

$\nwarrow$  #Observations

# Extending!

More predictors: total inventory, district size, ...

→ Extend methodology to  $p$  predictors:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \epsilon$$

Response

Predictors

```
> my_lm <- lm(sales ~ ads + comp + ..., data = shop_data)
```

# RMSE & Adjusted R-Squared

More predictors  $\rightarrow$   $\left\{ \begin{array}{l} \text{Higher } \textit{complexity} \text{ and } \textit{cost} \\ \text{Lower } \textit{RMSE} \text{ and higher } \textit{R-squared} \end{array} \right.$

Solution: *adjusted R-squared*

- Penalizes more predictors
- Used to compare

```
> summary(my_lm)$adj.r.squared
```

In Example: 0.819  $\rightarrow$  0.906

# Influence of predictors

- p-value: indicator influence of parameter
- p-value low — more likely parameter has significant influence

```
> summary(my_lm)
```

Call:

```
lm(formula = sales ~ ads + comp, data = shop_data)
```

Residuals:

Min	1Q	Median	3Q	Max
-131.920	-23.009	-4.448	33.978	146.486

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	228.740	80.592	2.838	0.009084	**
ads	25.521	5.900	4.325	0.000231	***
comp	-19.234	4.549	-4.228	0.000296	***

P-Values



# Example

- Want 95% confidence — p-value  $\leq 0.05$
- Want 99% confidence — p-value  $\leq 0.01$

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	228.740	80.592	2.838	0.009084	**
ads	25.521	5.900	4.325	0.000231	***
comp	-19.234	4.549	-4.228	0.000296	***

P-Values



**Note:** Do not mix up R-squared with p-values!

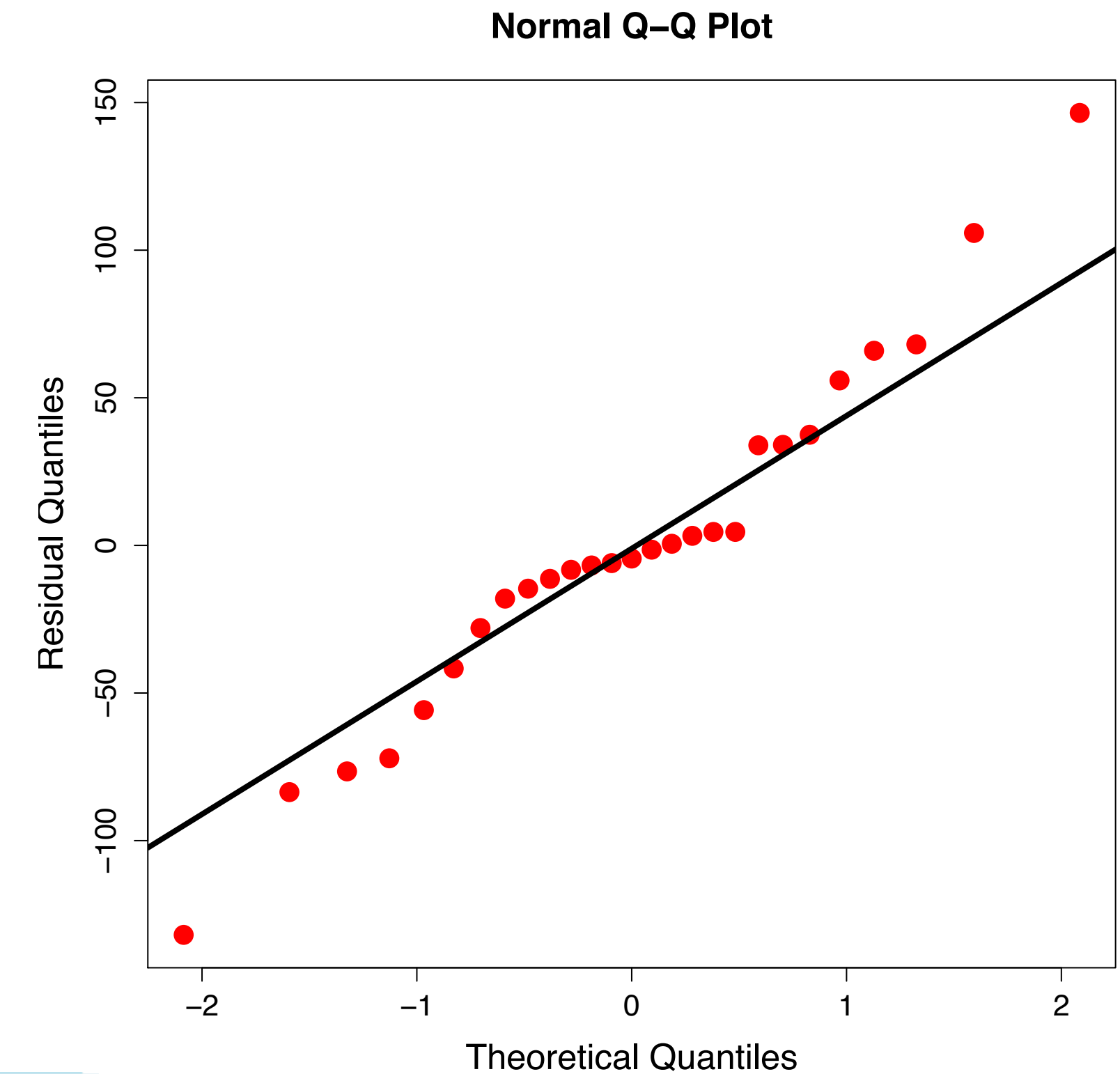
# Assumptions

- Just make a model, make a summary and look at p-values?
- Not that simple!
- We made some implicit assumptions

# Verifying Assumptions

Residuals:

- Independent: No pattern?
- Identical Normal: Approximately a line?



```
> plot(lm_shop$fitted.values, lm_shop$residuals)
```

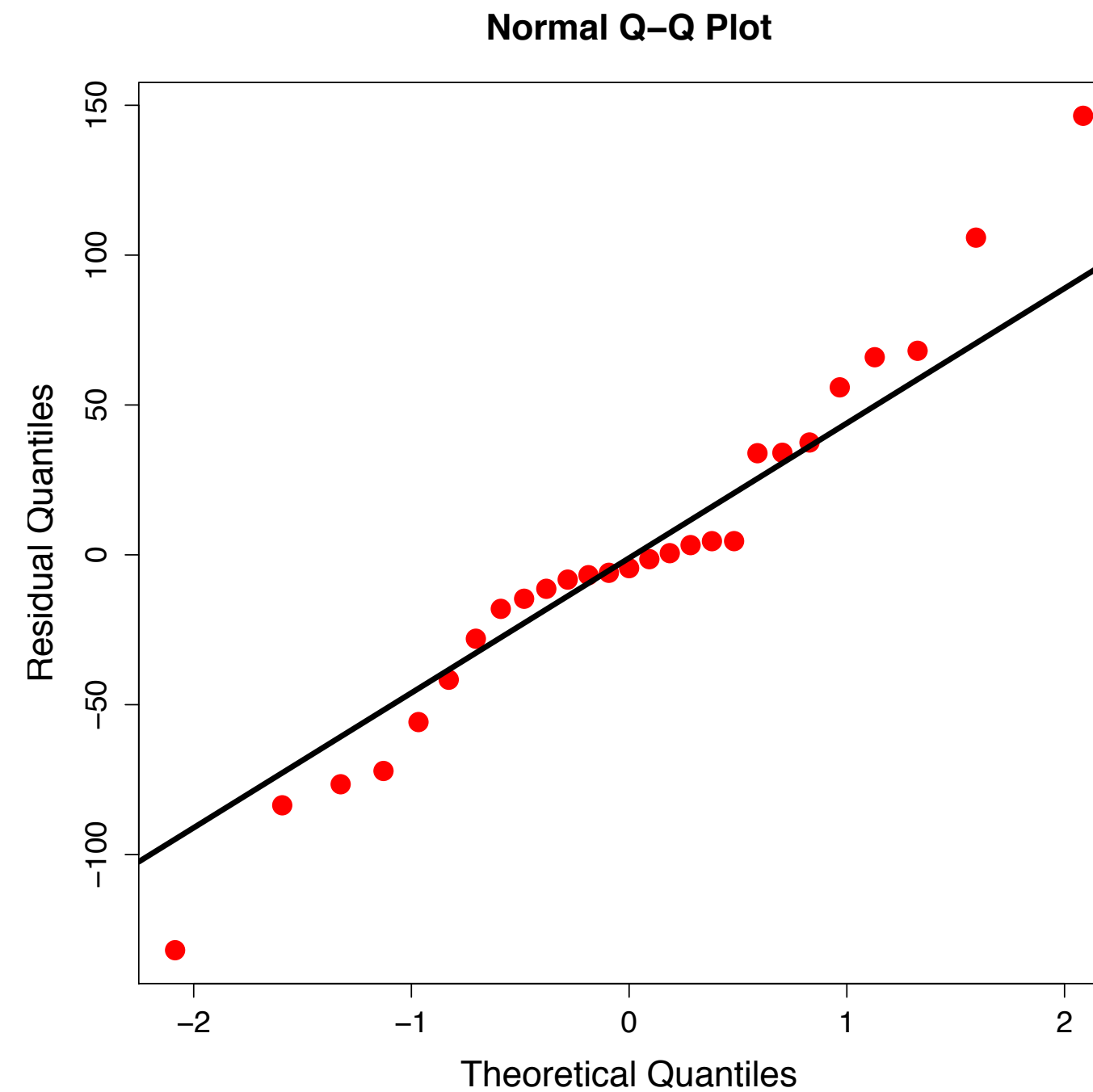
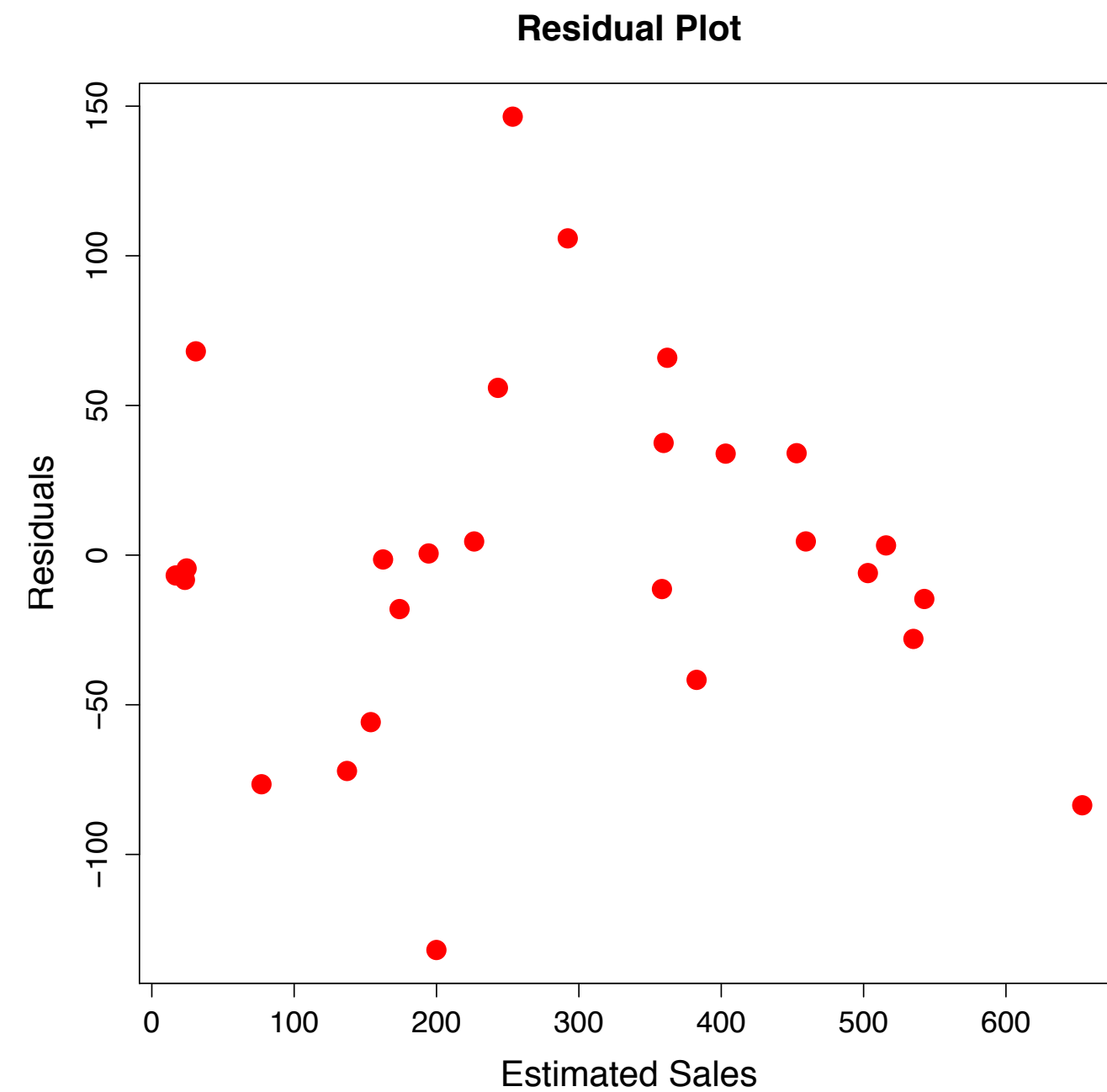
```
> qqnorm(lm_shop$residuals)
```



**Draws normal Q-Q plot**



# Verifying Assumptions



- Important to avoid mistakes!
- Alternative tests exist



## INTRODUCTION TO MACHINE LEARNING

# Let's practice!

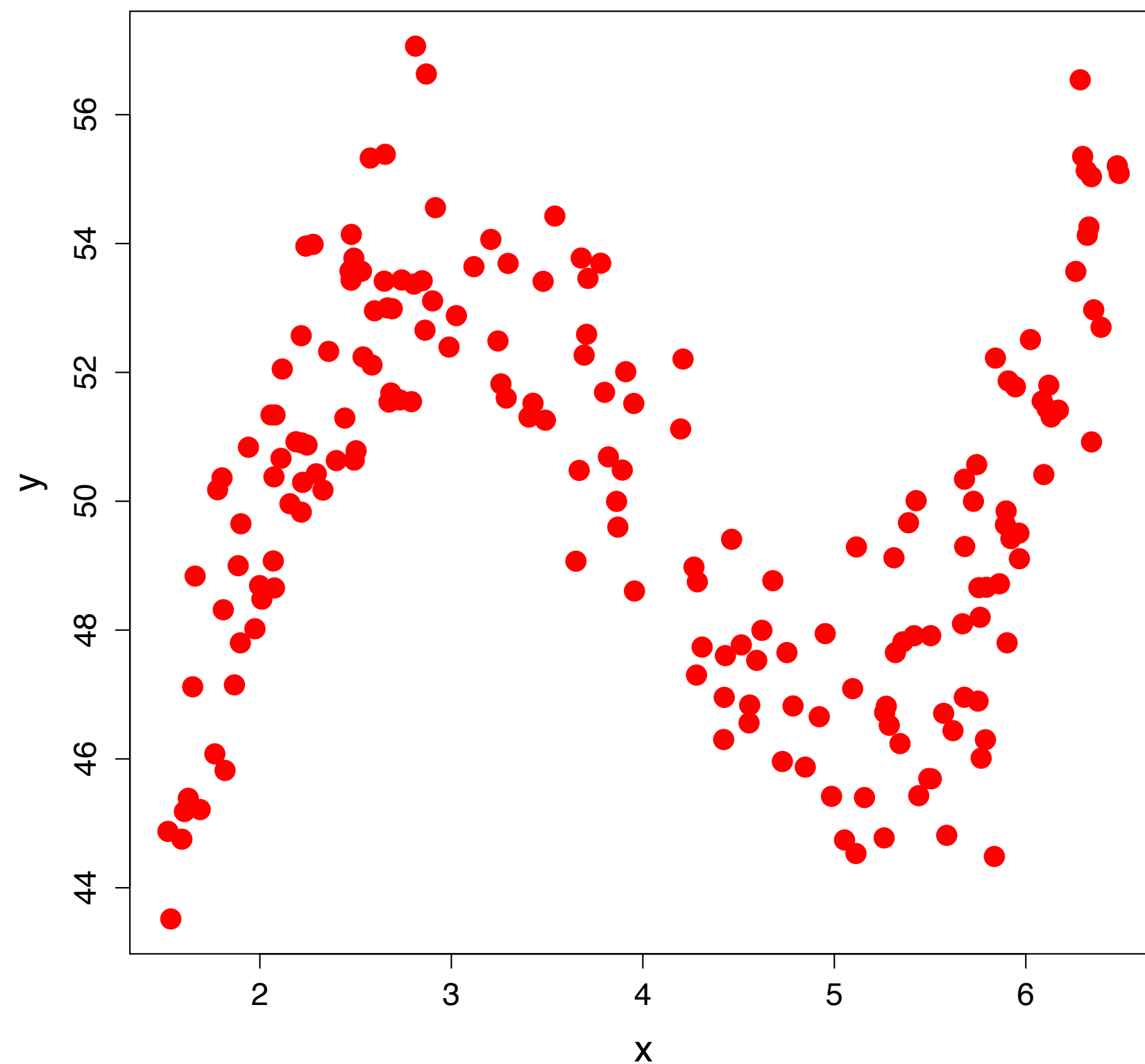


INTRODUCTION TO MACHINE LEARNING

# **k-Nearest Neighbors and Generalization**

# Non-Parametric Regression

Problem: Visible pattern, but not linear



# Non-Parametric Regression

Problem: Visible pattern, but not linear

Solutions:

- |                             |   |          |
|-----------------------------|---|----------|
| • Transformation            | ← | Tedious  |
| • Multi-linear Regression   | ← | Advanced |
| • non-Parametric Regression | ← | Doable   |

# Non-Parametric Regression

Problem: Visible pattern, but not linear

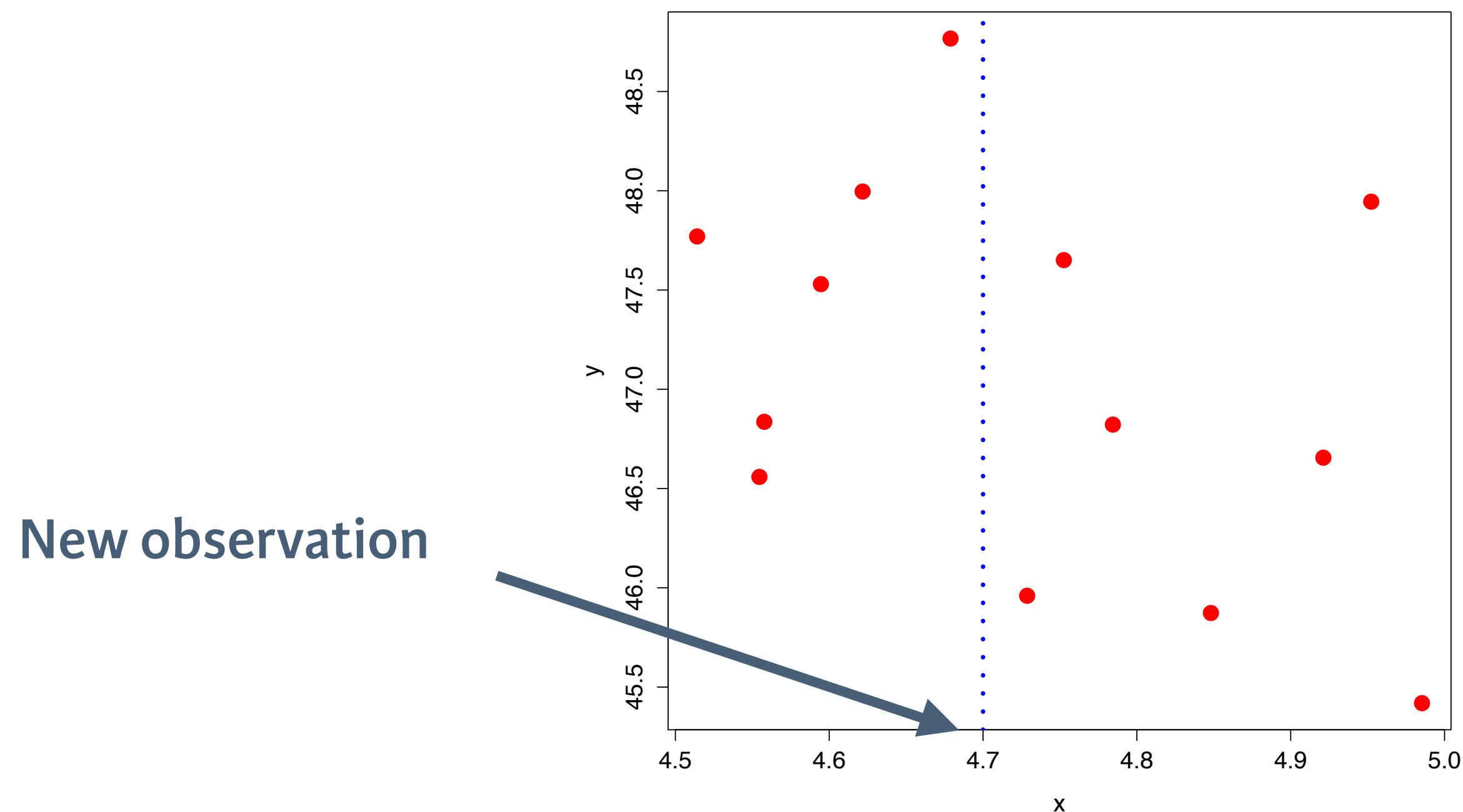
Techniques:

- k-Nearest Neighbors
- Kernel Regression
- Regression Trees
- ...

No parameter estimations required!

# k-NN: Algorithm

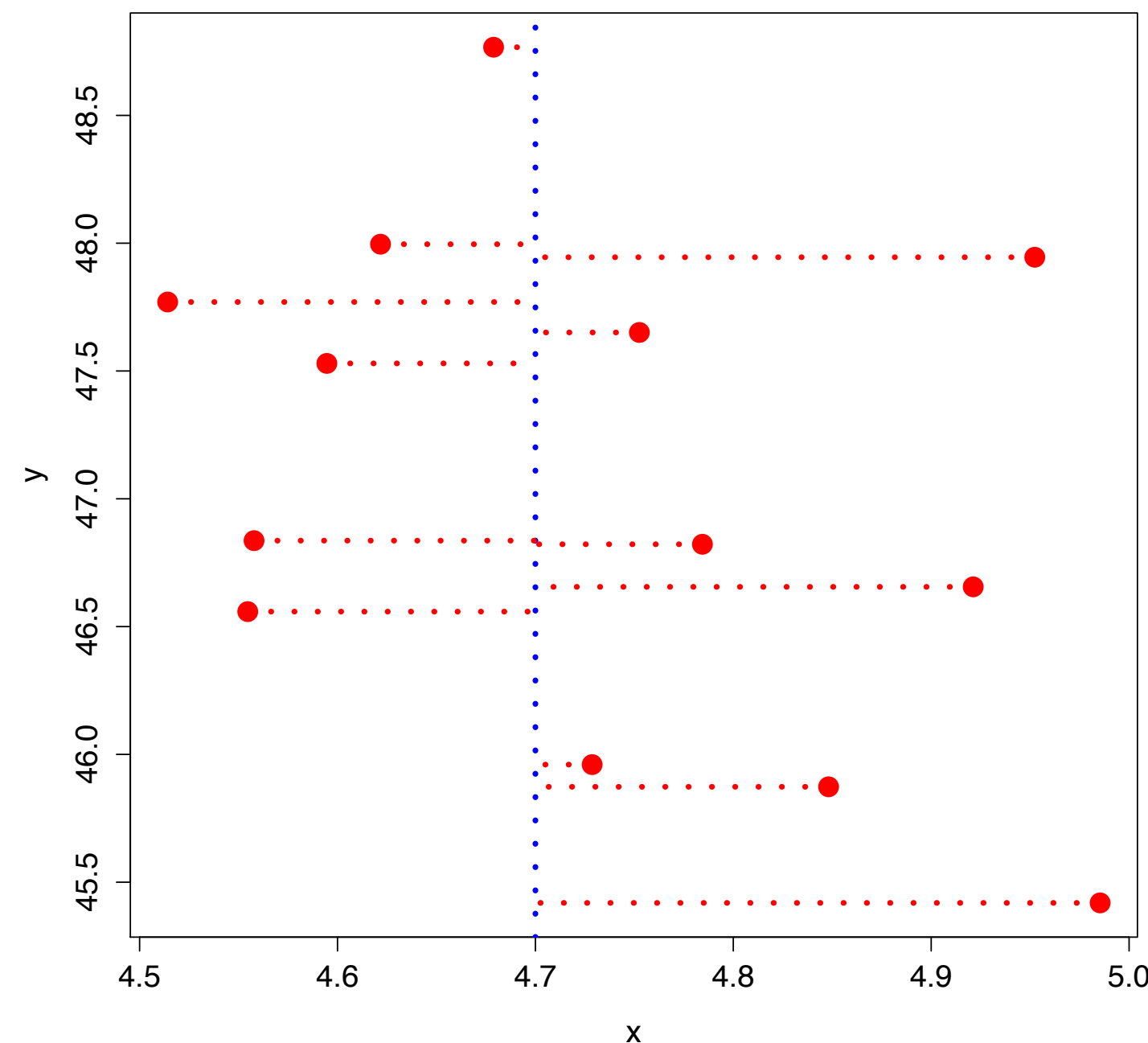
Given a *training set* and a *new observation*:



# k-NN: Algorithm

Given a *training set* and a *new observation*:

1. Calculate the distance in the predictors



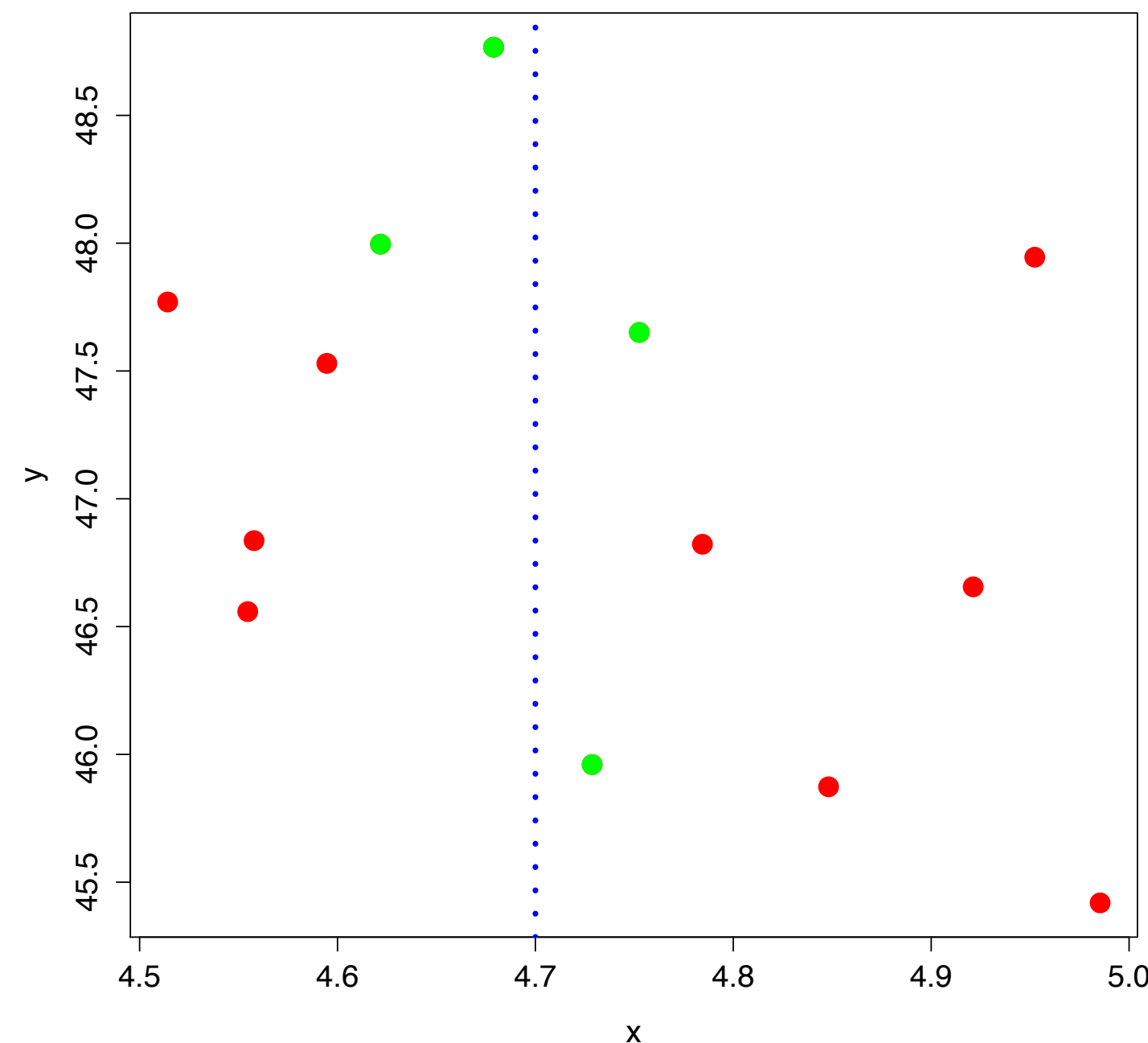


# k-NN: Algorithm

Given a *training set* and a *new observation*:

2. Select the  $k$  nearest

$k = 4$  →

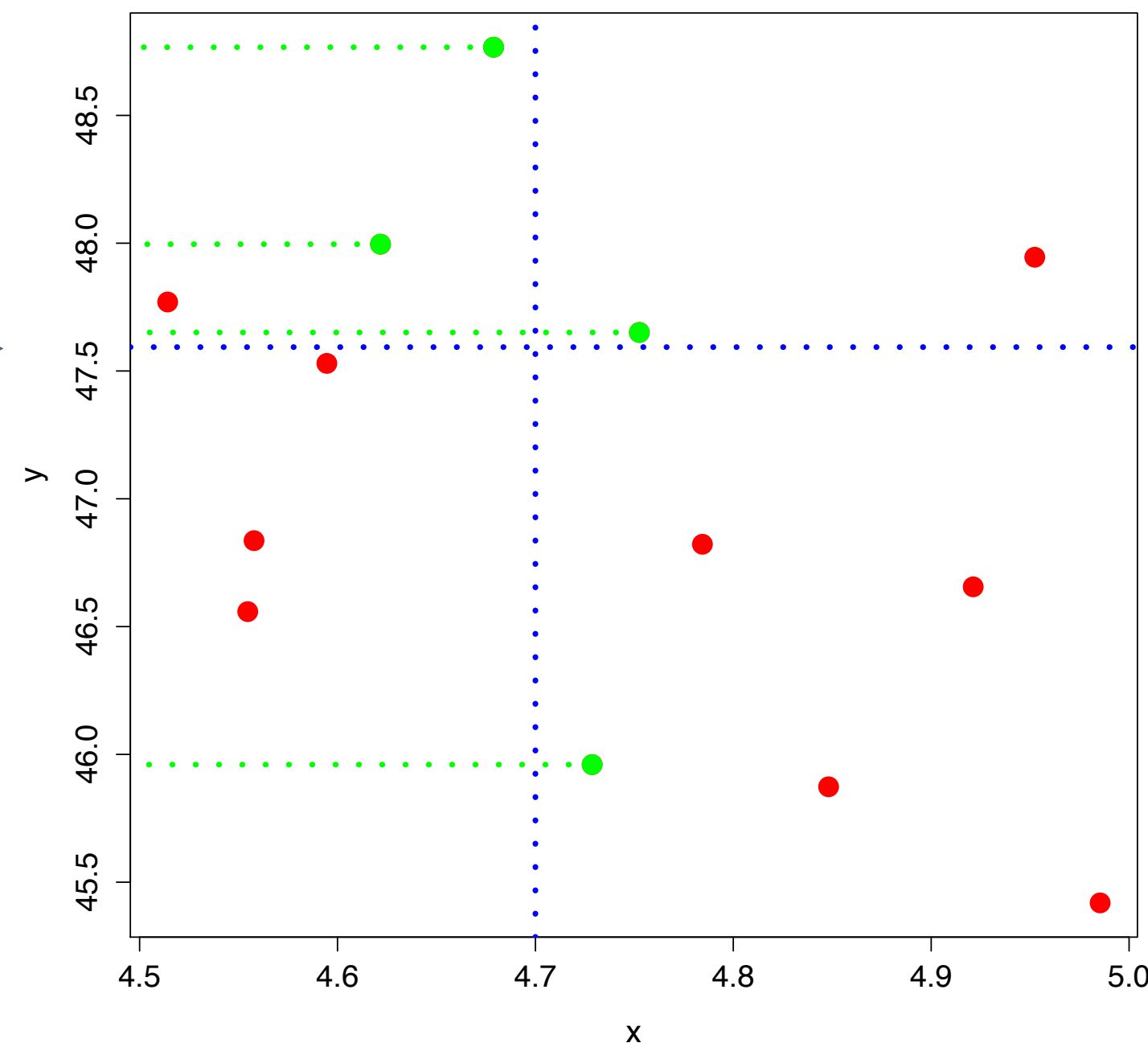


# k-NN: Algorithm

Given a *training set* and a *new observation*:

3. Aggregate the response of the  $k$  nearest

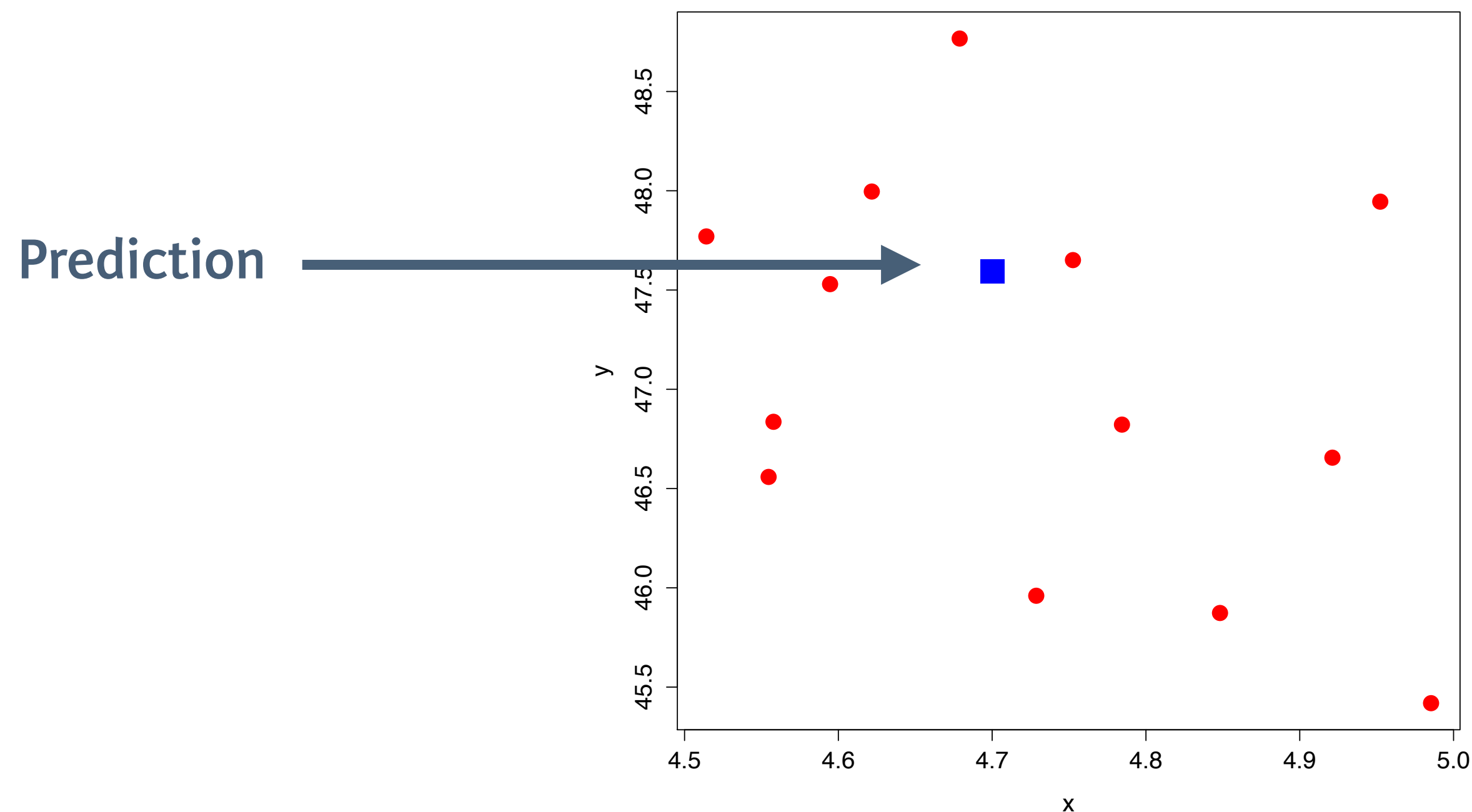
Mean of 4 responses



# k-NN: Algorithm

Given a *training set* and a *new observation*:

4. The outcome is your prediction



# Choosing $k$

- $k = 1$ : Perfect fit on training set but poor predictions
- $k = \text{\#obs in training set}$ : Mean, also poor predictions

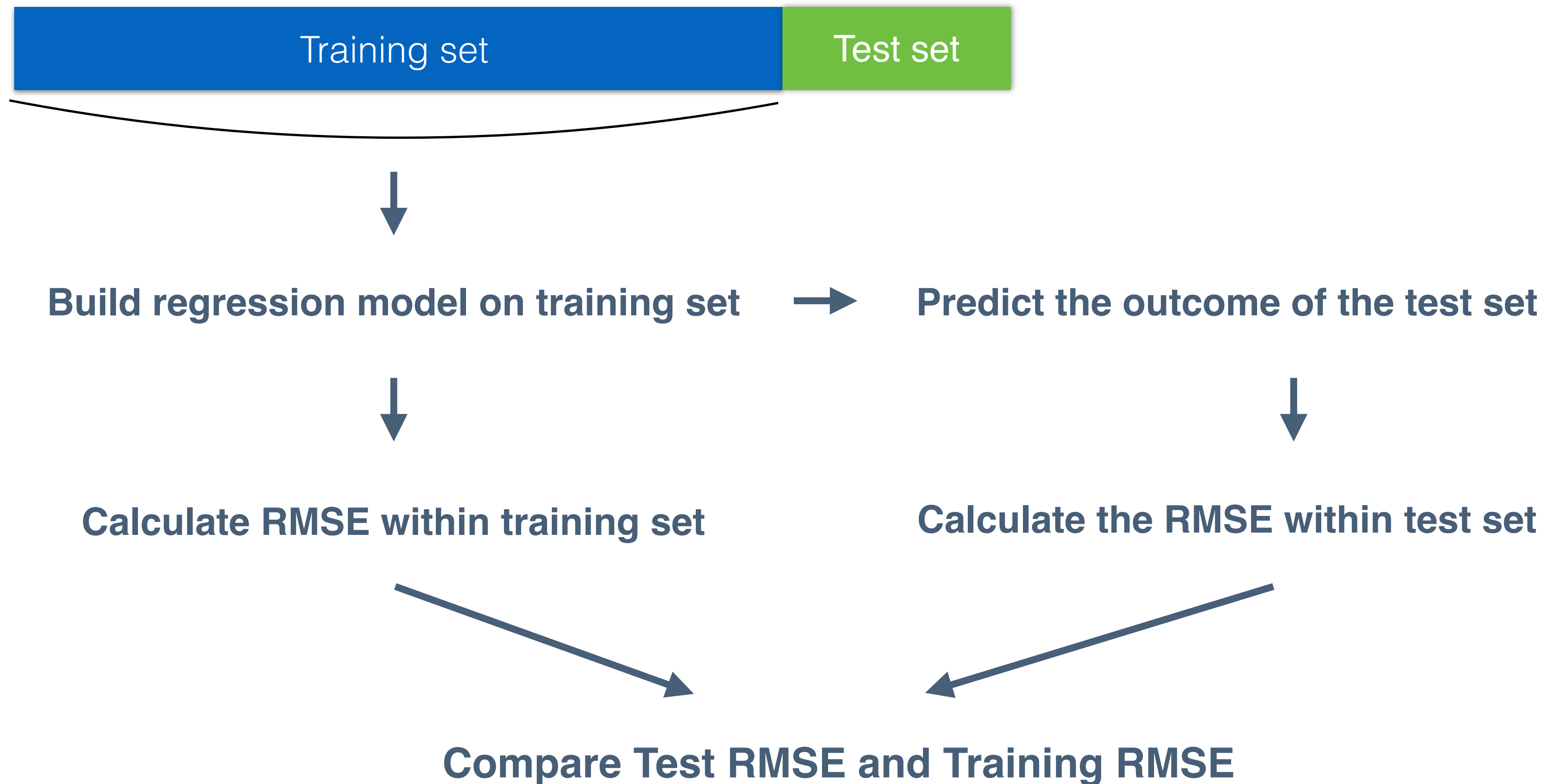
**Bias - Variance** trade off!

Reasonable:  $k = 20\%$  of  $\text{\#obs in training set}$

# Generalization in Regression

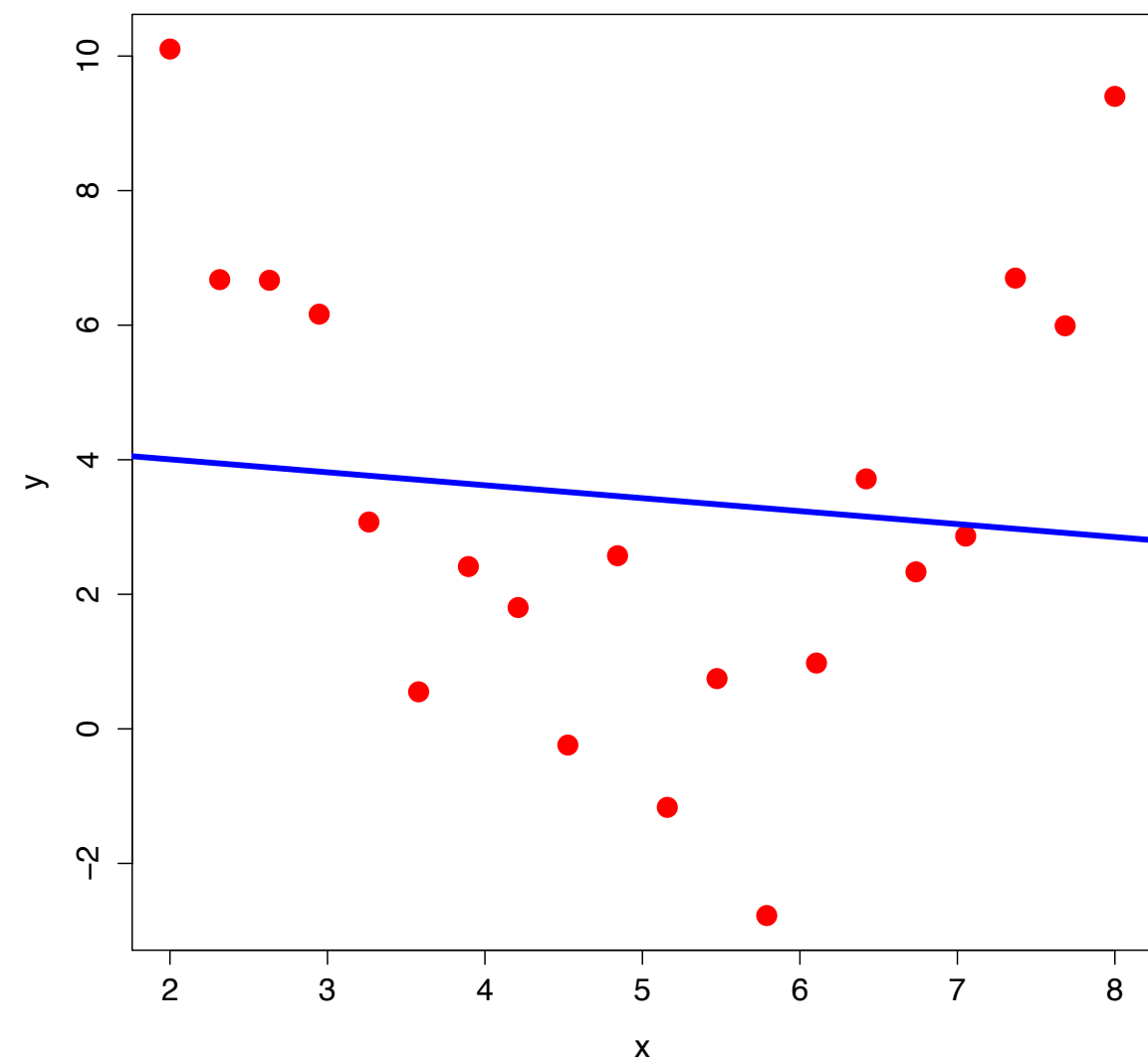
- Built your own regression model
- Worked on training set
- Does it generalize well?!
- Two techniques
  - Hold Out: simply split the dataset ←
  - K-fold cross-validation

# Hold Out Method for Regression



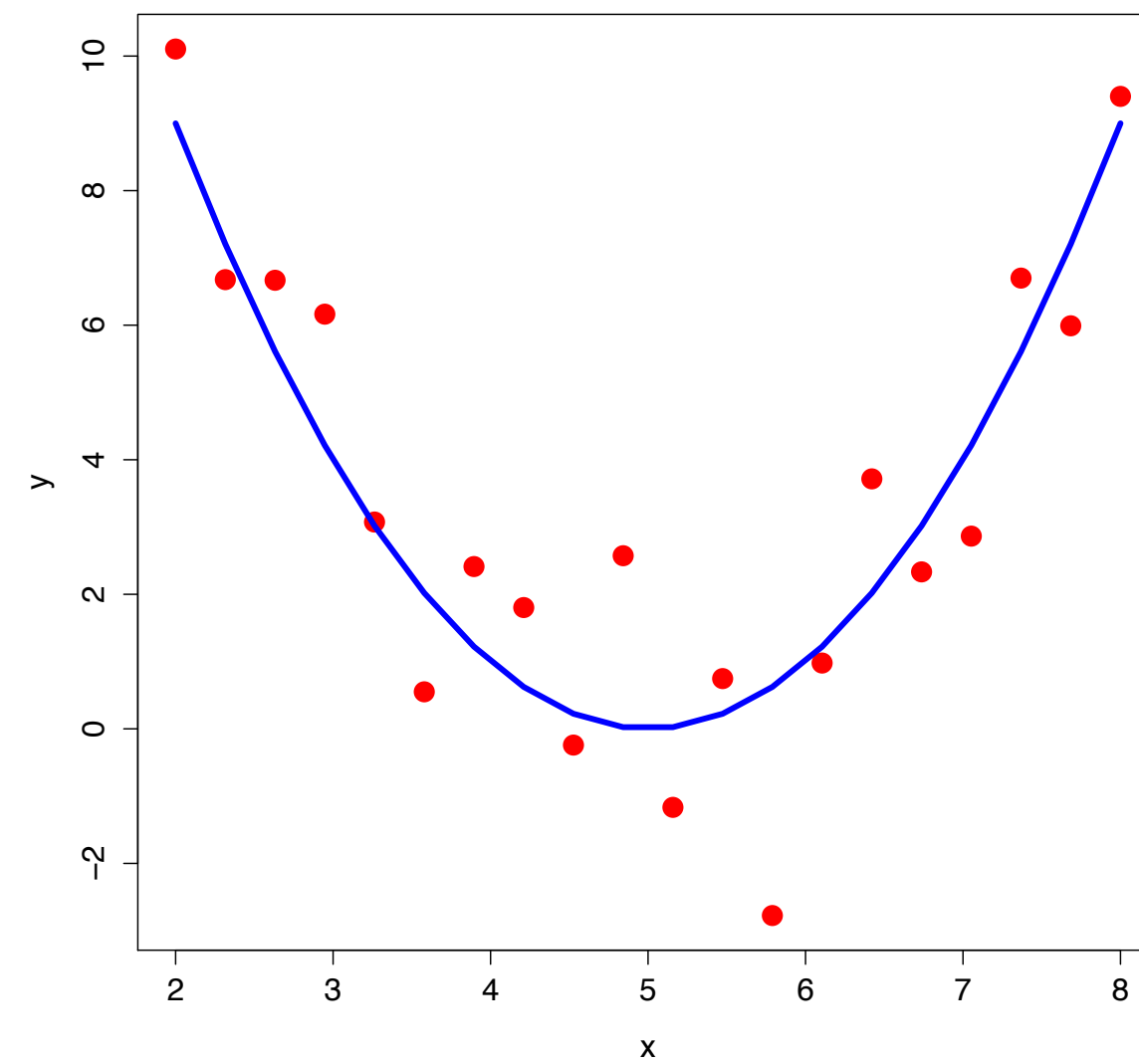
# Under and Overfitting

Underfit

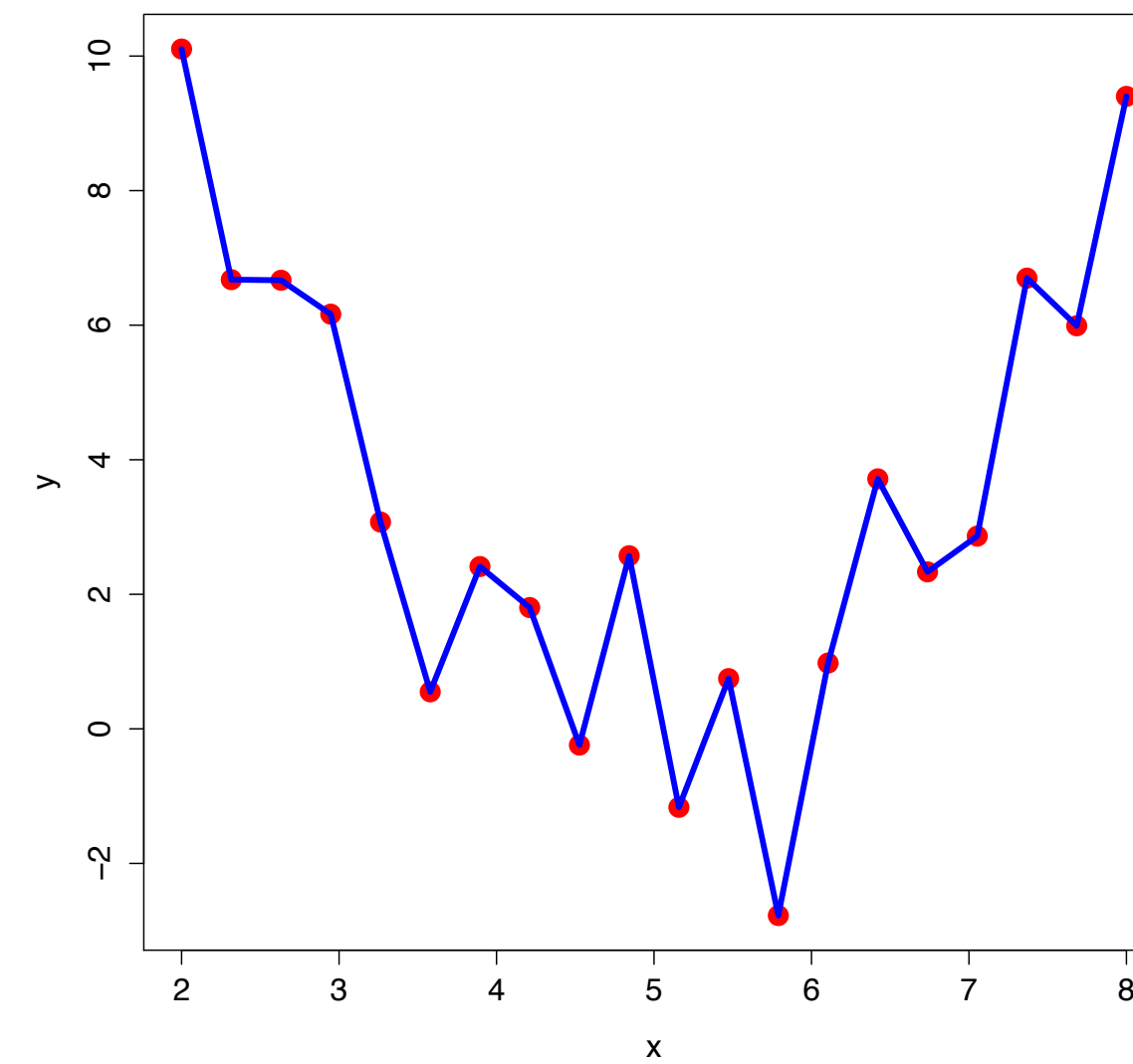


- Fit: ✗
- Generalize: ✓
- Prediction: ✗

Overfit



- Fit: ✓
- Generalize: ✓
- Prediction: ✓



- Fit: ✓
- Generalize: ✗
- Prediction: ✗



## INTRODUCTION TO MACHINE LEARNING

# Let's practice!