Module 3: Linear Regression

Daesoo Lee,

Department of Mathematical Sciences, NTNU

NTNU

Norwegian University of
Science and Technology

23/01/2023

What will you learn?

- Simple linear regression
- Multiple linear regression

Linear Regression again!?

- Also, we’ll learn about cool ML models from Module 7/8!

Warm-up

Warm-up

Do-it-yourself “by hand”

- Go to the following webpage: https://gallery.shinyapps.io/simple_regression/
- We have:

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

Try to estimate $\hat{\beta}_0$ (intercept) and $\hat{\beta}_1$ (slope).

Warm-up

Do-it-yourself “by hand”

- Sum of Squares of Residuals (RSS)

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \hat{f}(x_i)\right)^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

The best estimates can be found at the minimal RSS!

- The distribution of the residuals form the *normal* distribution!

Linear Regression

Linear Regression

- Very simple approach for *supervised learning*.
- Parametric
- Is linear regression too simple?

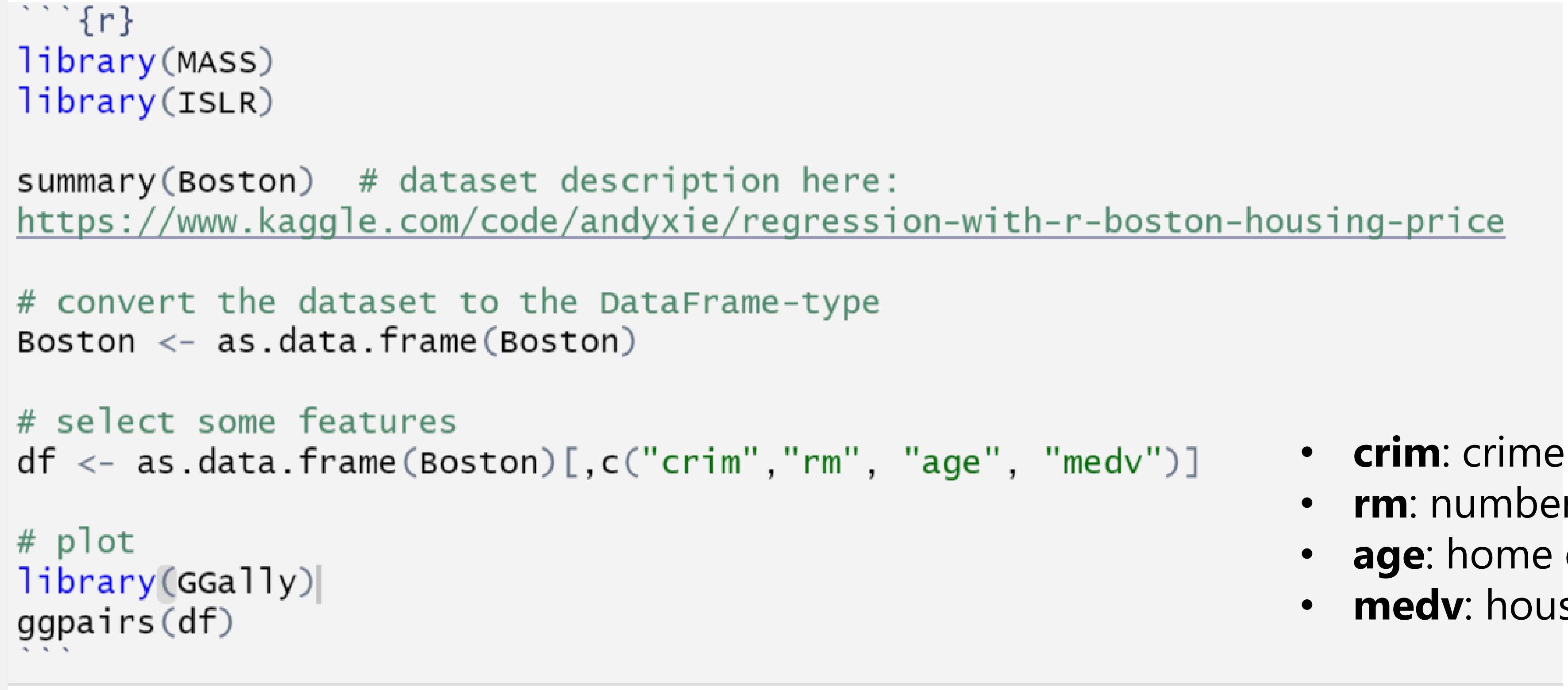
  → Could be, but very useful.

  → Many learning methods can be seen as generalization of the linear model.

Linear Regression

Example: Boston Housing Price Dataset

- {crime rate in the region, a number of rooms, home owner's age, housing price}



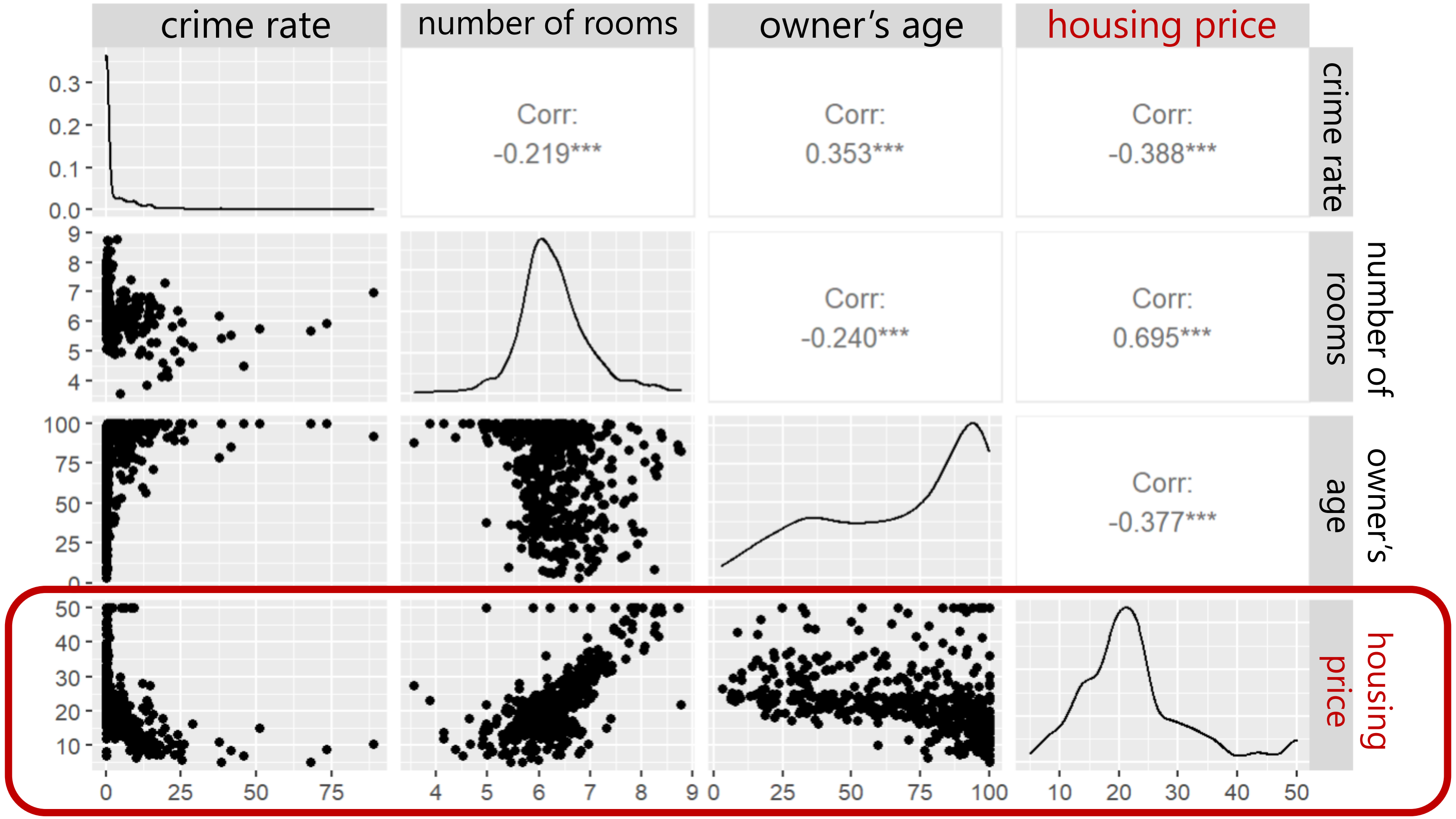
~~~
```{r}
library(MASS)
library(ISLR)

summary(Boston)  # dataset description here:
https://www.kaggle.com/code/andyxie/regression-with-r-boston-housing-price

# convert the dataset to the DataFrame-type
Boston <- as.data.frame(Boston)

# select some features
df <- as.data.frame(Boston)[,c("crim","rm", "age", "medv")]

# plot
library(GGally)
ggpairs(df)
```
~~~

- **crim**: crime rate
- **rm**: number of rooms
- **age**: home owner's age
- **medv**: housing price

https://www.kaggle.com/code/andyxie/regression-with-r-boston-housing-price

Linear Regression

Example: Boston Housing Price Dataset



https://www.kaggle.com/code/andyxie/regression-with-r-boston-housing-price

Linear Regression

Interesting Questions

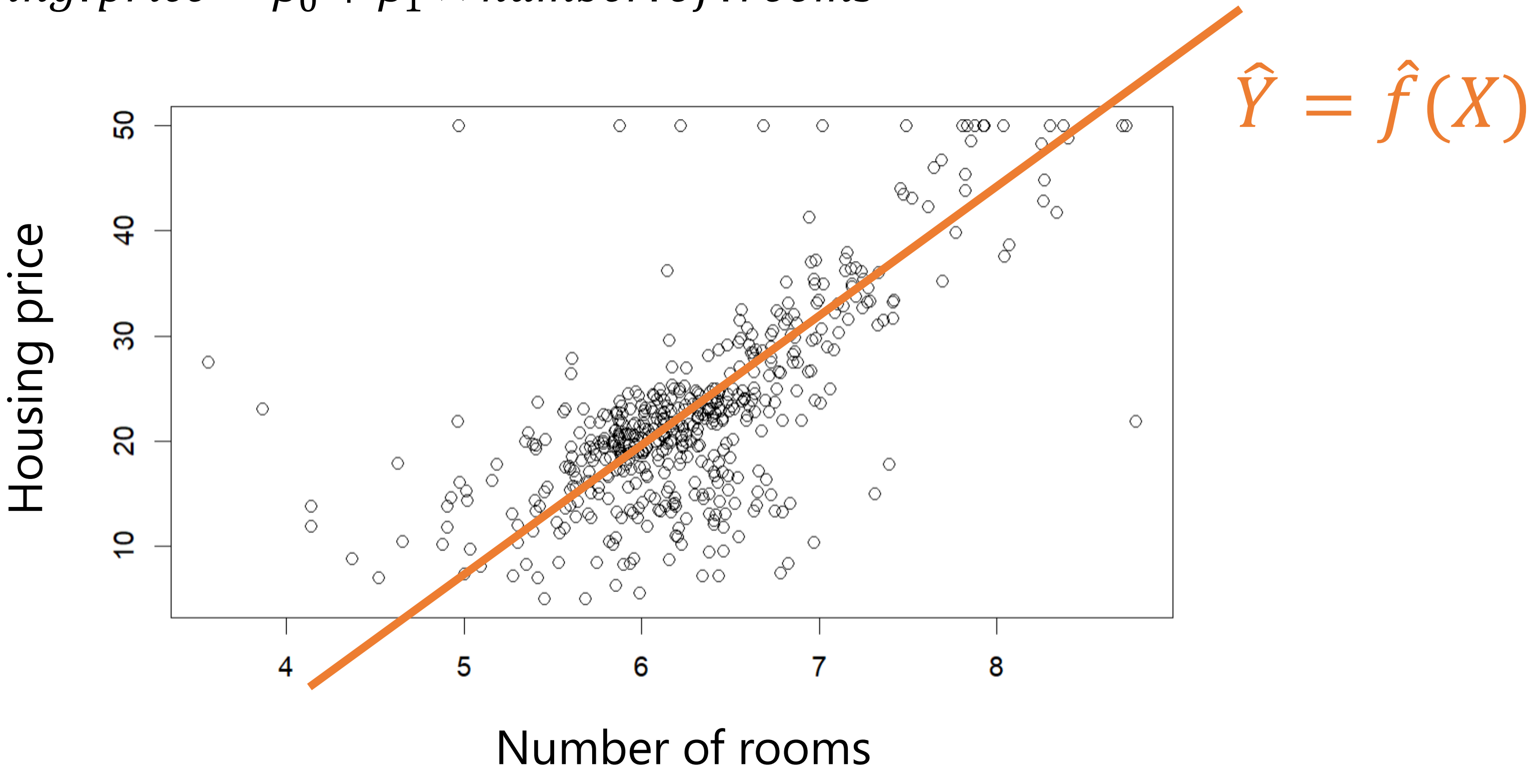
- How good is “a number of room” as an input feature for predicting the housing price?
- How strong is the relationship?
- Is the relationship linear?
- How well can we predict the housing price?

Simple Linear Regression

Linear Regression

Let’s think about the simplest form

- $Y = \beta_0 + \beta_1 X_1 + \epsilon$
- $housing.price = \beta_0 + \beta_1 \times number.of.rooms$



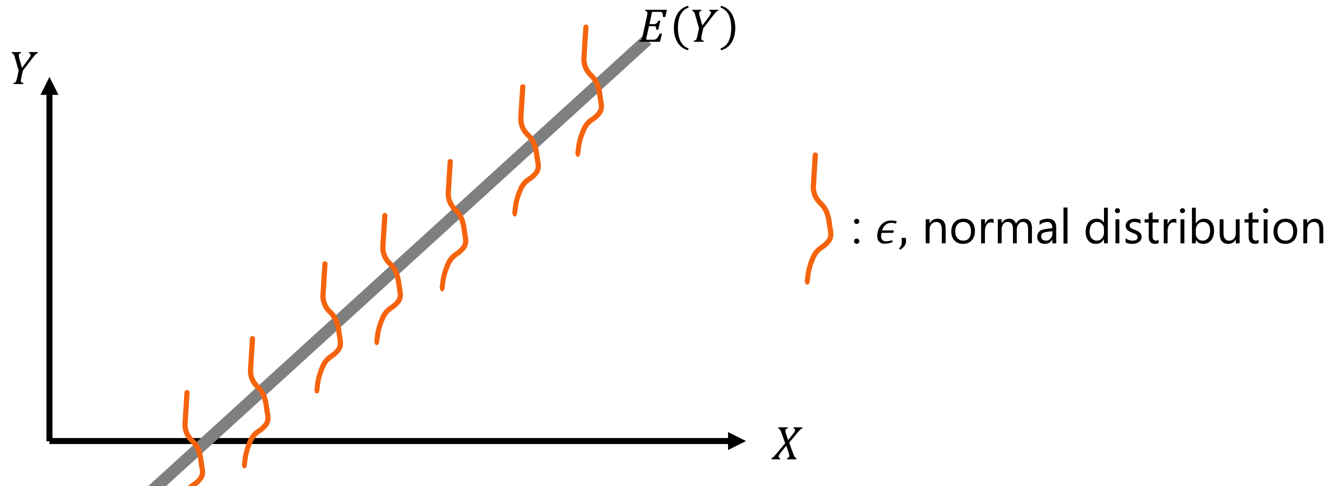
Linear Regression

Modeling Assumptions

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

- $\epsilon \sim N(0, \sigma^2)$
- In this formulation, Y is a random variable, $Y \sim N(\beta_0 + \beta_1 X_1, \sigma^2)$

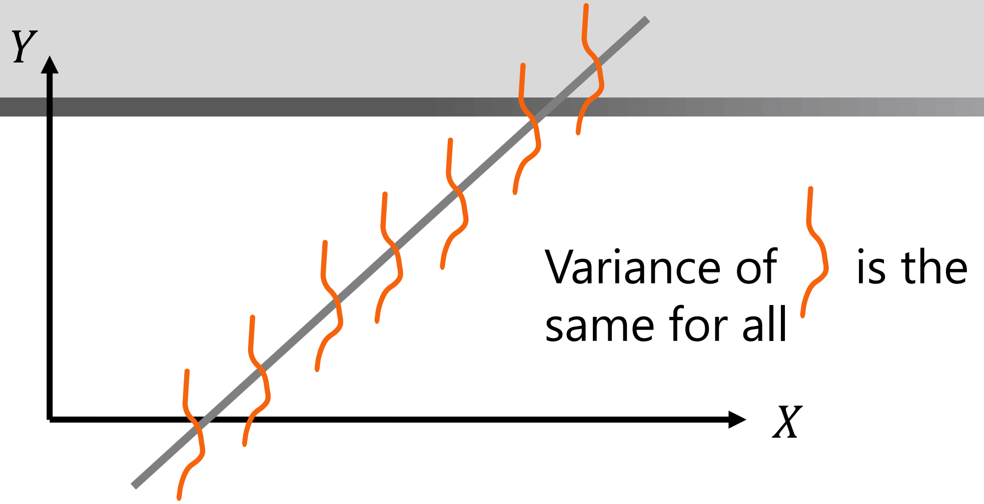
$$Y = \underbrace{\text{expected value}}_{\mathrm{E}(Y) = \beta_0 + \beta_1 X_1} + \underbrace{\text{error}}_{\varepsilon}$$



Linear Regression

Modeling Assumptions

- $\mathrm{E}[\epsilon] = 0$
- $\mathrm{Var}[\epsilon] = \sigma^2$
- All ϵ_i are normal distributed; $\epsilon_i \in \epsilon$
- ϵ is independent of any variable.
  i.e., ϵ is independent of X_1 and Y.
- $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ are independent of each other.



Variance of ∿ is the same for all

$$Y = \overbrace{\beta_0 + \beta_1 X_1}^{\text{The straight line}} + \overbrace{\epsilon}^{\text{normal distribution } \epsilon \sim N(0,\sigma^2)}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \beta_0 + \beta_1 \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Linear Regression

Parameter estimation ("model fitting")

$$Y = \underbrace{\text{expected value}}_{\mathsf{E}(Y) = \beta_0 + \beta_1 X_1} + \underbrace{\text{error}}_{\varepsilon}$$

- Then, our best bet is
  to estimate the expected value.
- It'd be impossible to accurately predict "error".

- We can estimate $\hat{\beta}$ that minimizes RSS (Residual Sum of Squares)

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \hat{f}(x_i)\right)^2$$
$$= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

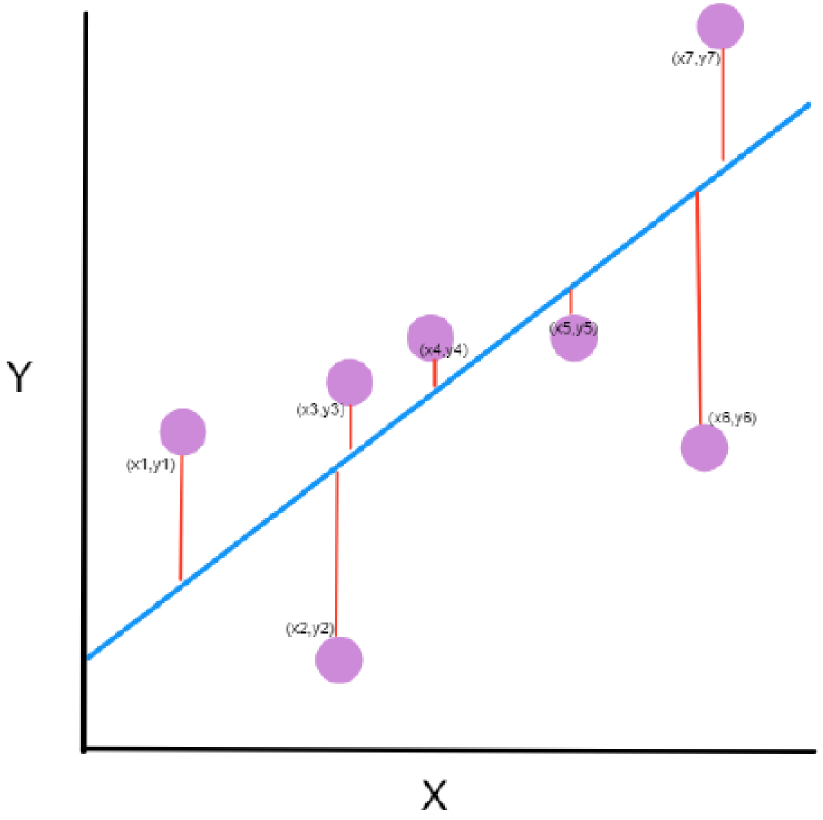
$$Y = \overbrace{\beta_0 + \beta_1 X_1}^{\textbf{The straight line}} + \overbrace{\epsilon}^{\substack{\textbf{normal}\\ \textbf{distribution}\\ \epsilon\ \sim N(0,\sigma^2)}}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \beta_0 + \beta_1 \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Linear Regression

$\sum_{i=1}^{n}(y_i - \hat{y})^2$ **RSS is quite similar to MSE.**

MSE (Mean Squared Error) Illustration



$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2$$

Linear Regression

Residual Sum of Squares (RSS)

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \hat{f}(x_i)\right)^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- We estimate $\hat{f}$ that minimizes RSS.
- Then, we get “best” estimates of the learnable parameters $\hat{\beta}_0, \hat{\beta}_1$. (NB! $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$)
- Residual is defined as $y_i - \hat{y}$

Linear Regression

Least Squares Estimators:

- Analytical solution for

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

The estimates for simple linear regression are given as

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\mathrm{Cov}(\boldsymbol{x}, \boldsymbol{y})}{\mathrm{Var}(\boldsymbol{x})}$$

where $\bar{x}$ and $\bar{y}$ are the sample means.

  - You've learned it in your previous statistics course. So, we're skipping the proof for this. You can find the proof in Chapter 11 of the book by Walepole et al. (2012), see here.
  - We'll look at the generic matrix form in a later slide here.

- Numerical Solution:
  Gradient Descent (we'll learn about it during the module for neural networks)

Linear Regression

Do-it-yourself “by hand”

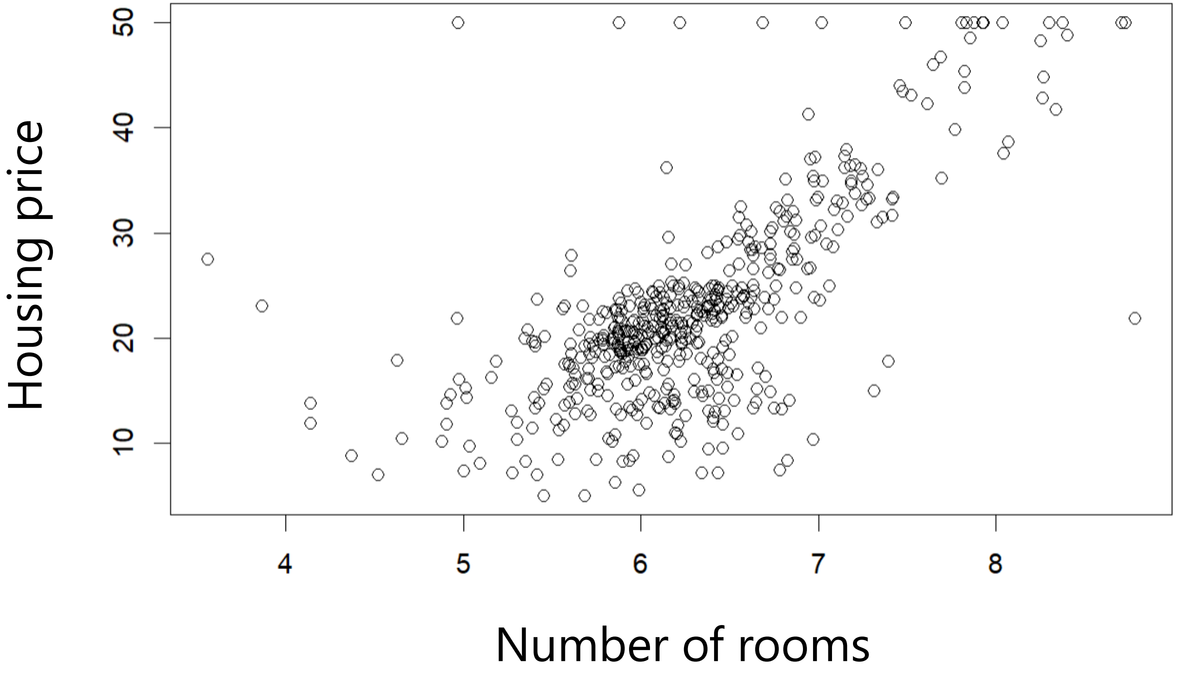
- Go to the following webpage: https://gallery.shinyapps.io/simple_regression/
- Try to “estimate” the correct parameters.

Linear Regression

Example (continued): Boston Housing Price Dataset

- Let’s look at the simplest form

$$housing.price = \beta_0 + \beta_1 \times number.of.rooms$$



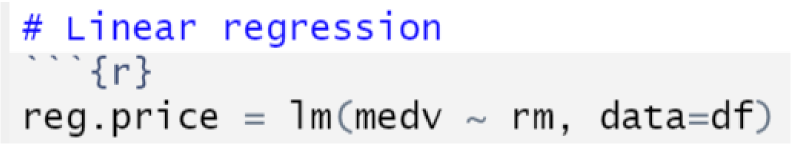
- and let’s estimate $\hat{f}$ in R!

Linear Regression

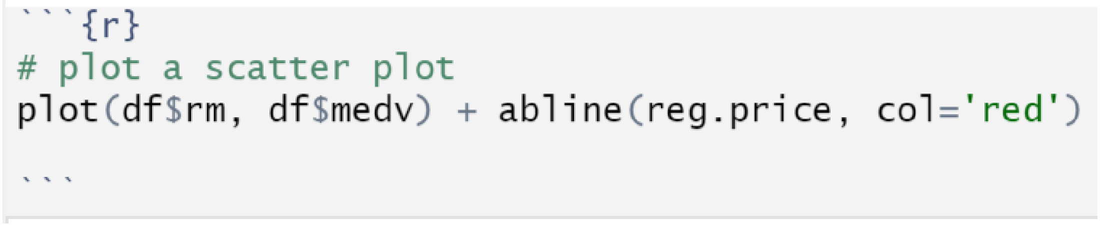
Example (continued): Boston Housing Price Dataset

- Let's look at the simplest form

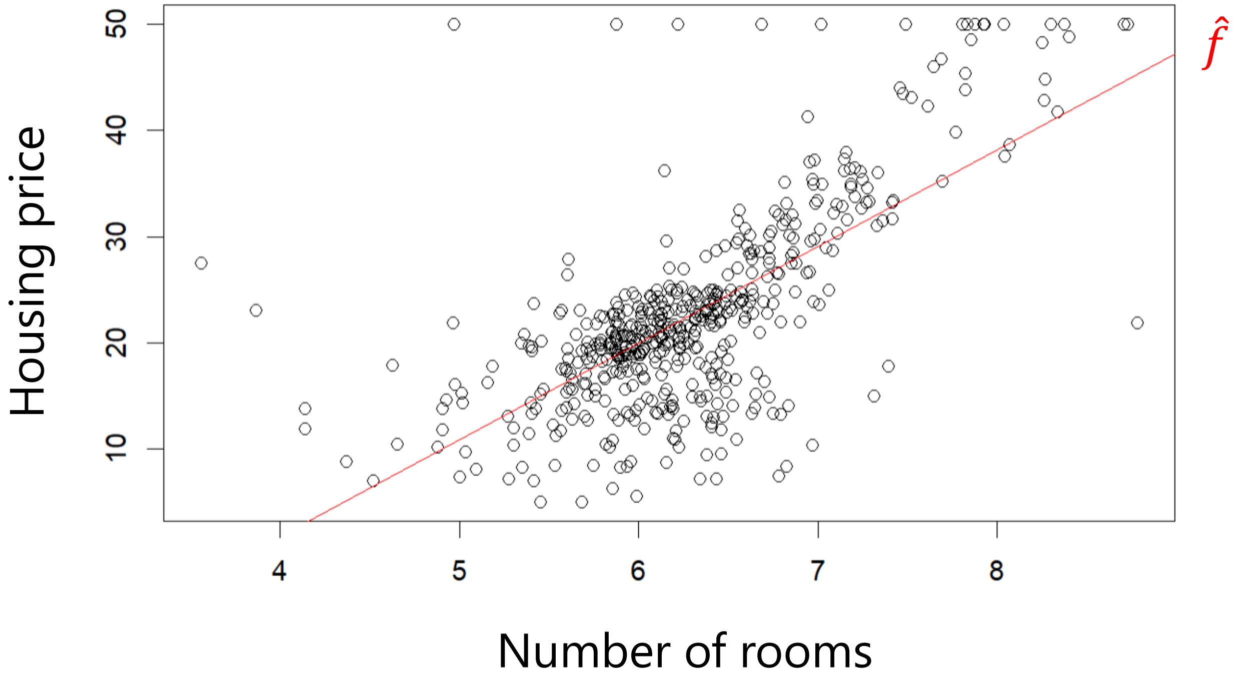
$$\widehat{housing.price} = \hat{\beta}_0 + \hat{\beta}_1 \times \widehat{number.of}.rooms$$



~~~
# Linear regression
```{r}
reg.price = lm(medv ~ rm, data=df)
~~~

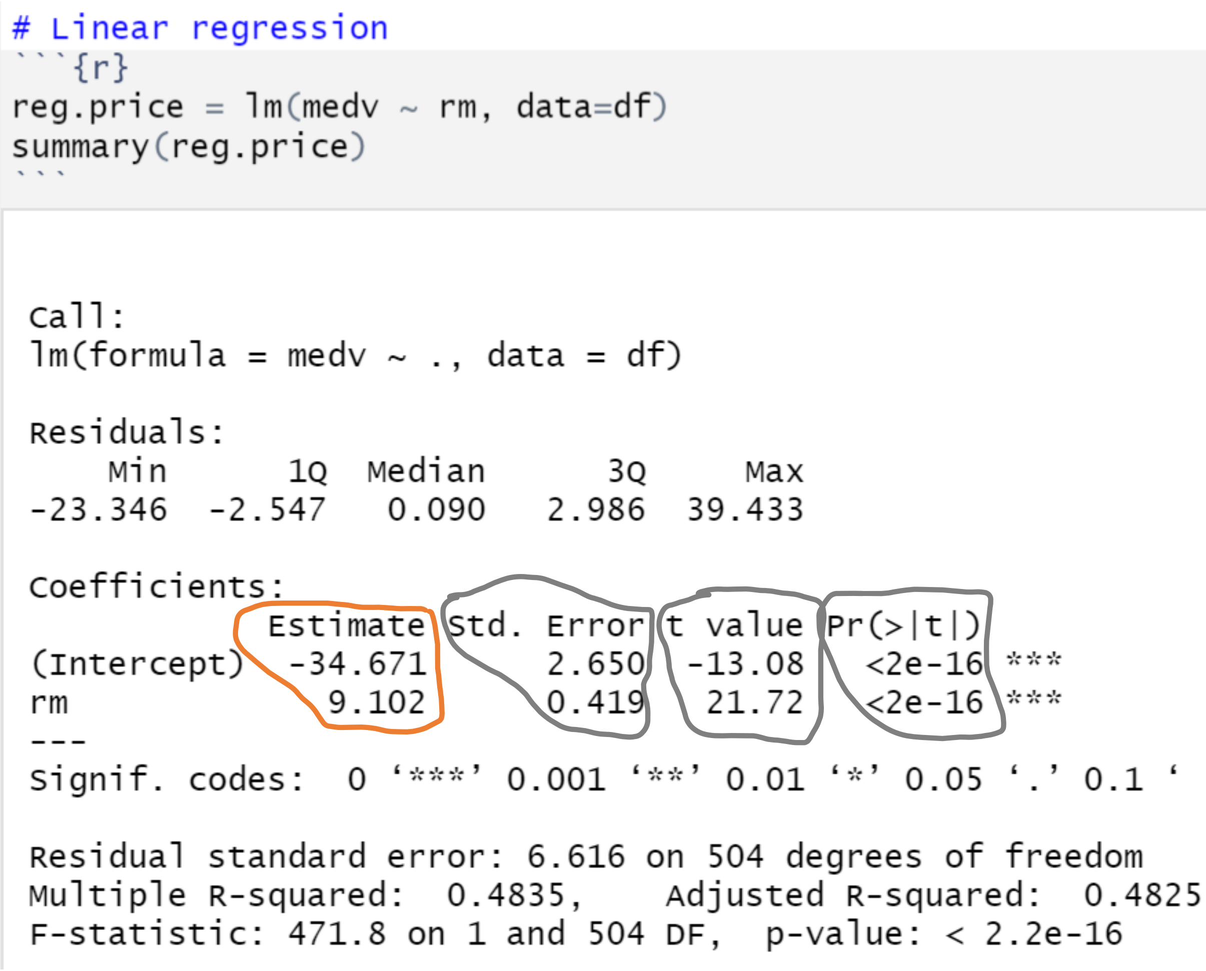


~~~
```{r}
# plot a scatter plot
plot(df$rm, df$medv) + abline(reg.price, col='red')

```
~~~



Linear Regression

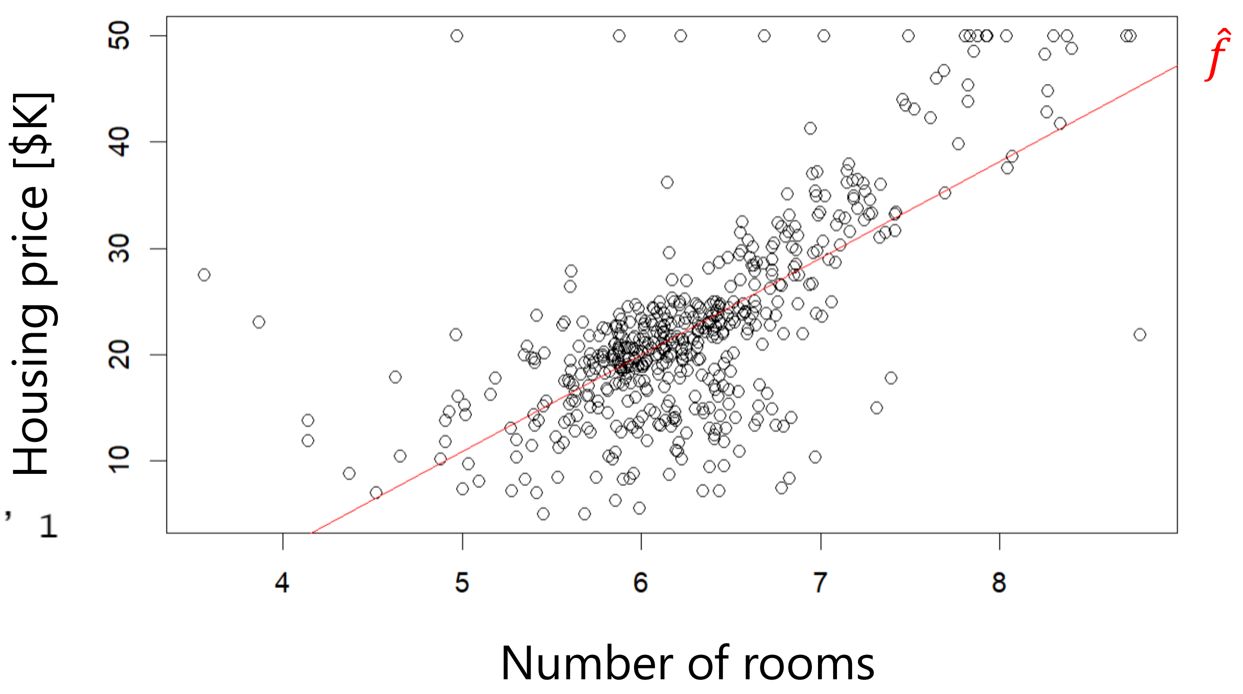
Example (continued): Boston Housing Price Dataset



~~~
# Linear regression
```{r}
reg.price = lm(medv ~ rm, data=df)
summary(reg.price)
```


Call:
lm(formula = medv ~ ., data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-23.346  -2.547   0.090   2.986  39.433 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -34.671      2.650  -13.08   <2e-16 ***
rm             9.102      0.419   21.72   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.616 on 504 degrees of freedom
Multiple R-squared:  0.4835,    Adjusted R-squared:  0.4825 
F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
~~~

$$\widehat{housing.price} = \hat{\beta}_0 + \hat{\beta}_1 \times \widehat{number.of}.rooms$$

$$\widehat{housing.price} = -34.7 + 9.1 \times number.of.rooms$$



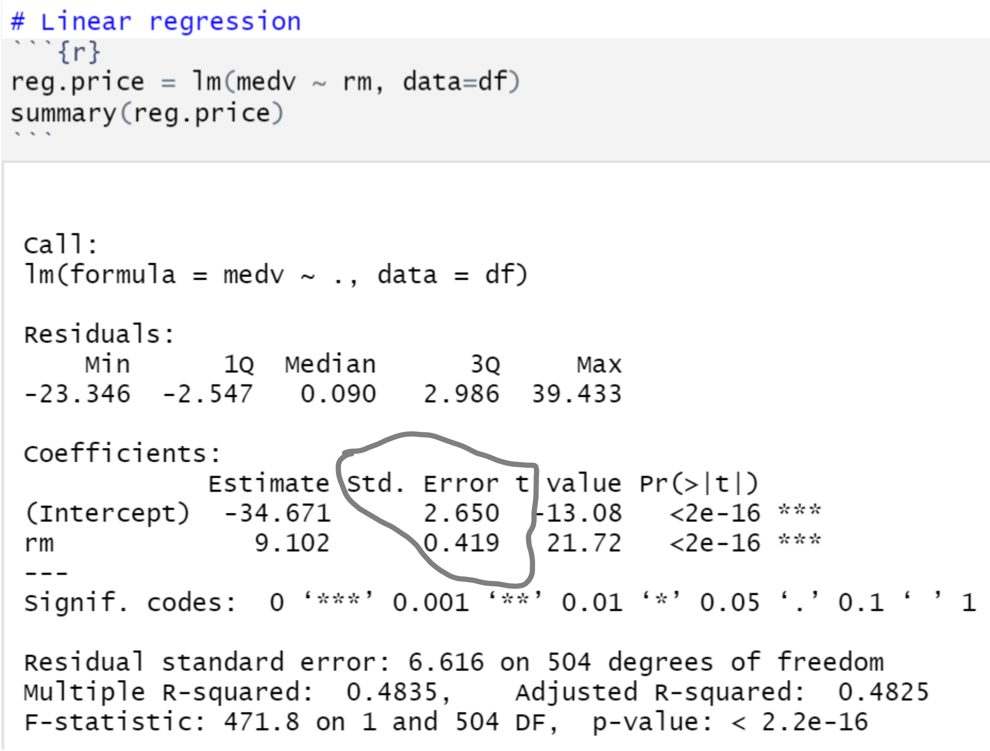
Linear Regression

Uncertainty in the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$

- Because we "estimate" $\hat{\beta}$, it contains some uncertainty.
- *e.g.*, $\widehat{\mathrm{BMI}} = \hat{\beta}_0 + \hat{\beta}_1 \times Weight$

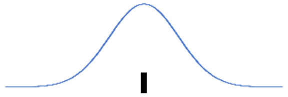
$\widehat{\mathrm{BMI}} = \underbrace{\hat{\beta}_0 + \hat{\beta}_1} \times Weight$ $\widehat{\mathrm{BMI}} = \underbrace{\hat{\beta}_0 + \hat{\beta}_1} \times Weight$

Should be similar but probably not the same!
Therefore, $\hat{\beta}$ carries some uncertainty!



~~~
# Linear regression
```{r}
reg.price = lm(medv ~ rm, data=df)
summary(reg.price)
```
~~~

```
Call:
lm(formula = medv ~ ., data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-23.346  -2.547   0.090   2.986  39.433 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -34.671      2.650  -13.08   <2e-16 ***
rm             9.102      0.419   21.72   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.616 on 504 degrees of freedom
Multiple R-squared:  0.4835,	Adjusted R-squared:  0.4825 
F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
```



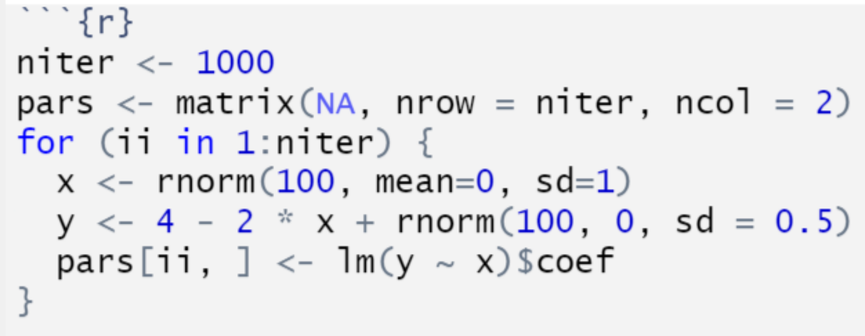
$\hat{\beta}_{rm}$

9.1

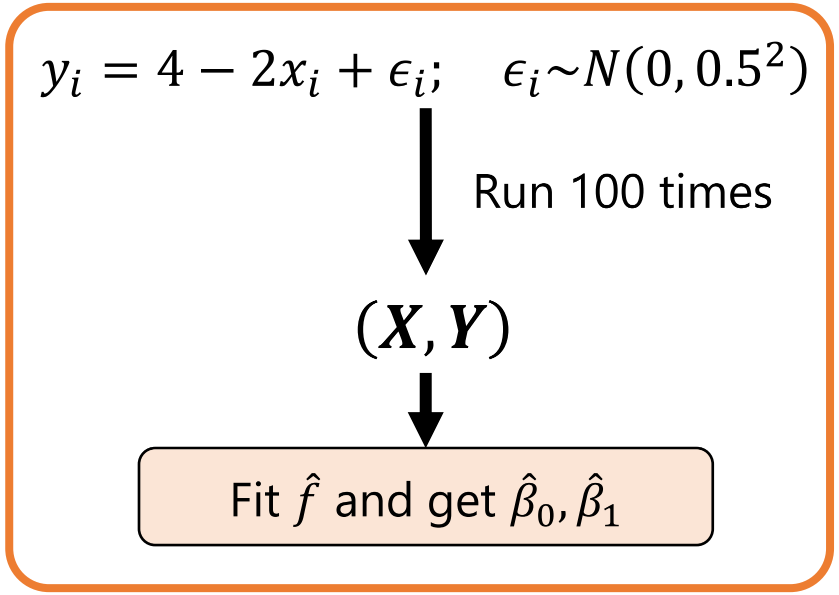
Linear Regression

Uncertainty in the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$

- Let's look at some simulated example.



~~~
```{r}
niter <- 1000
pars <- matrix(NA, nrow = niter, ncol = 2)
for (ii in 1:niter) {
  x <- rnorm(100, mean=0, sd=1)
  y <- 4 - 2 * x + rnorm(100, 0, sd = 0.5)
  pars[ii, ] <- lm(y ~ x)$coef
}
~~~



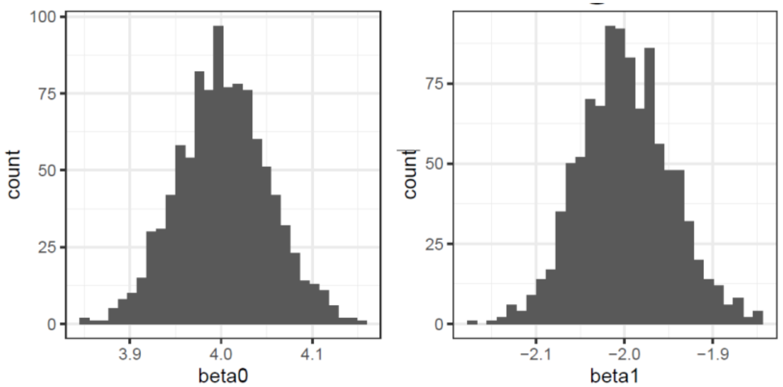
$y_i = 4 - 2x_i + \epsilon_i; \quad \epsilon_i \sim N(0, 0.5^2)$

Run 100 times

$(\boldsymbol{X}, \boldsymbol{Y})$

Fit $\hat{f}$ and get $\hat{\beta}_0, \hat{\beta}_1$

Run this loop 1,000 times → 1,000 pairs of $\hat{\beta}_0, \hat{\beta}_1$



$$y_i = \beta_0 - \beta_1 x_i + \epsilon_i$$

$$\hat{y}_i = \hat{\beta}_0 - \hat{\beta}_1 x_i$$

$$\hat{\beta}_0 \sim N\left(\beta_0, \sigma^2_{\beta_0}\right)$$

$$\hat{\beta}_1 \sim N\left(\beta_1, \sigma^2_{\beta_1}\right)$$

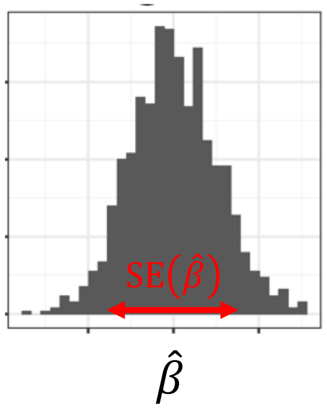
Linear Regression

Uncertainty in the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$

- The standard errors of $\hat{\beta}_0$ and $\hat{\beta}_1$ are given as follows:

$$\mathrm{SE}\left(\hat{\beta}_0\right)^2 = \hat{\sigma}\left[\frac{1}{n} + \frac{\bar{x}}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right]$$

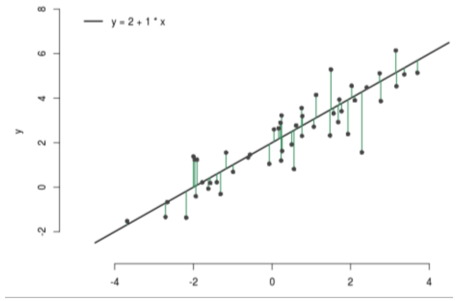
$$\mathrm{SE}\left(\hat{\beta}_1\right)^2 = \frac{\hat{\sigma}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

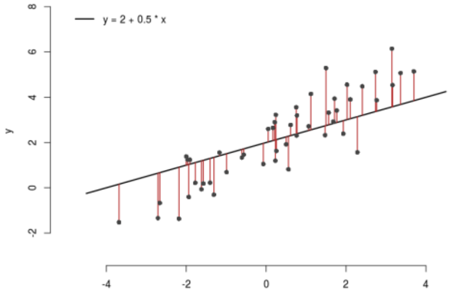


$$\hat{\sigma} = \mathrm{RSE} = \sqrt{\frac{1}{n-2}\mathrm{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

RSS: Residual Sum of Squares
RSE: Residual Standard of Error

- $\mathrm{Cov}\left(\hat{\beta}_0, \hat{\beta}_1\right)$ is, in general, different from zero.



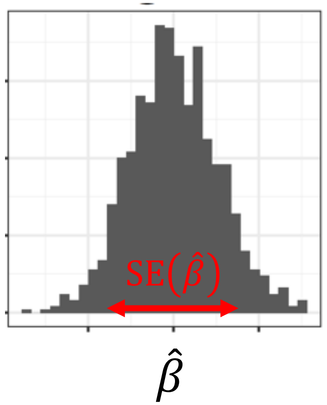


Linear Regression

Design issue with "Data Collection"

- We typically want to minimize uncertainty

$$\mathrm{SE}(\hat{\beta}_1)^2 = \frac{\hat{\sigma}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$



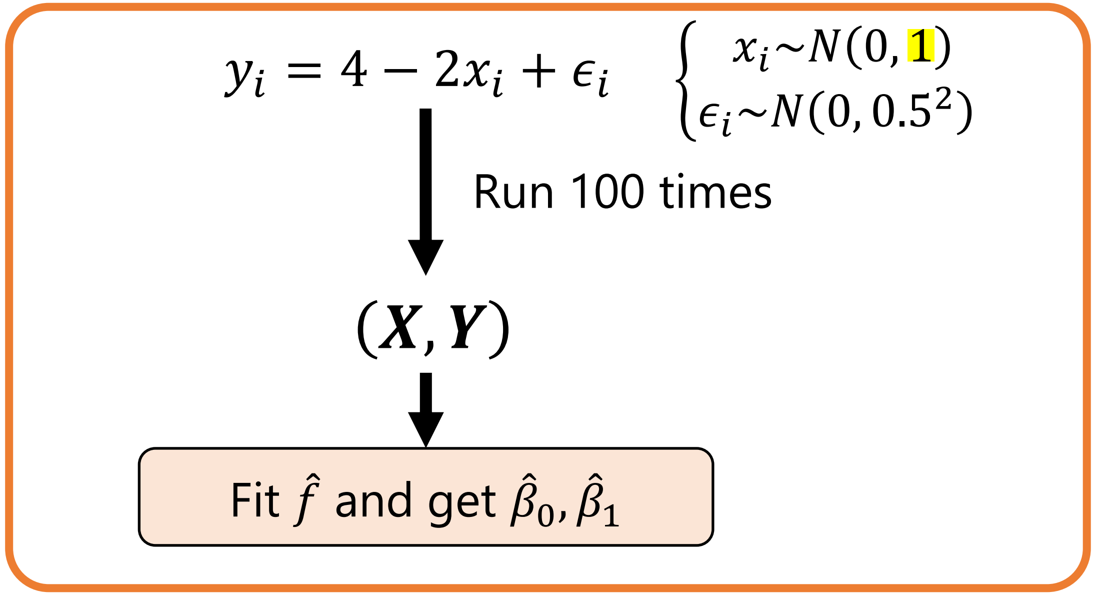
- We can reduce it through "*appropriate data collection*".
  We should increase $\sum(x - \bar{x})^2$

- That means, we should sample *diversly*.

Linear Regression

$$\mathrm{SE}(\hat{\beta}_1)^2 = \frac{\hat{\sigma}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Appropriate data collection to reduce the standard error



$$y_i = 4 - 2x_i + \epsilon_i \quad \begin{cases} x_i \sim N(0, 1) \\ \epsilon_i \sim N(0, 0.5^2) \end{cases}$$

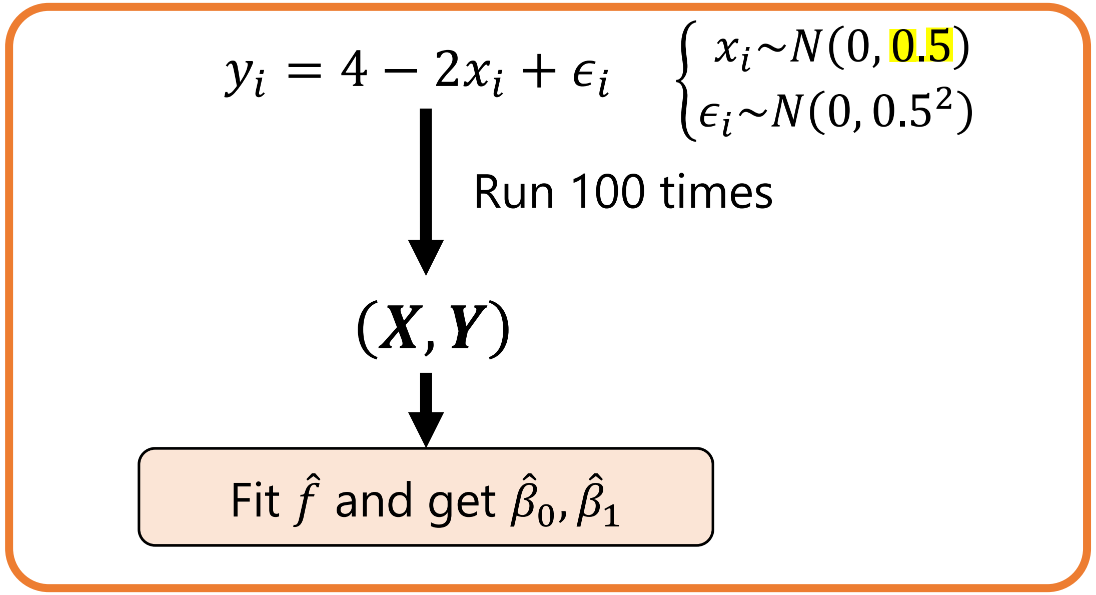
Run 100 times

$(\boldsymbol{X}, \boldsymbol{Y})$

Fit $\hat{f}$ and get $\hat{\beta}_0, \hat{\beta}_1$

Run this loop 1,000 times → 1,000 pairs of $\hat{\beta}_0, \hat{\beta}_1$

$$\mathrm{SE}(\hat{\beta}_1) = 0.05$$



$$y_i = 4 - 2x_i + \epsilon_i \quad \begin{cases} x_i \sim N(0, 0.5) \\ \epsilon_i \sim N(0, 0.5^2) \end{cases}$$

Run 100 times

$(\boldsymbol{X}, \boldsymbol{Y})$

Fit $\hat{f}$ and get $\hat{\beta}_0, \hat{\beta}_1$

Run this loop 1,000 times → 1,000 pairs of $\hat{\beta}_0, \hat{\beta}_1$

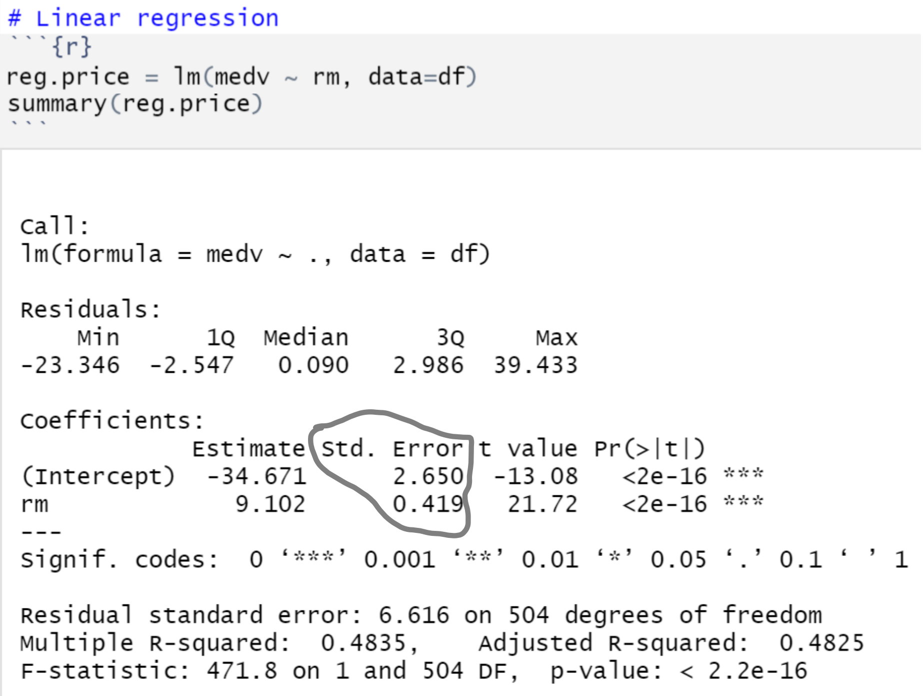
$$\mathrm{SE}(\hat{\beta}_1) = 0.1$$

Linear Regression

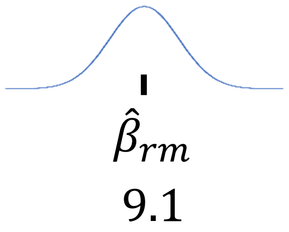
Residual Standard Error (RSE)

$$\hat{\sigma} = \text{RSE} = \sqrt{\frac{1}{n-2}\text{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- $\text{SE}\left(\hat{\beta}_0\right)^2 = \hat{\sigma}\left[\frac{1}{n} + \frac{\bar{x}}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right]$
- $\text{SE}\left(\hat{\beta}_1\right)^2 = \frac{\hat{\sigma}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$



~~~
# Linear regression
```{r}
reg.price = lm(medv ~ rm, data=df)
summary(reg.price)
```


Call:
lm(formula = medv ~ ., data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-23.346  -2.547   0.090   2.986  39.433 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -34.671      2.650  -13.08   <2e-16 ***
rm             9.102      0.419   21.72   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.616 on 504 degrees of freedom
Multiple R-squared:  0.4835,	Adjusted R-squared:  0.4825 
F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
~~~



Testing and Confidence Intervals

Testing and Confidence Intervals

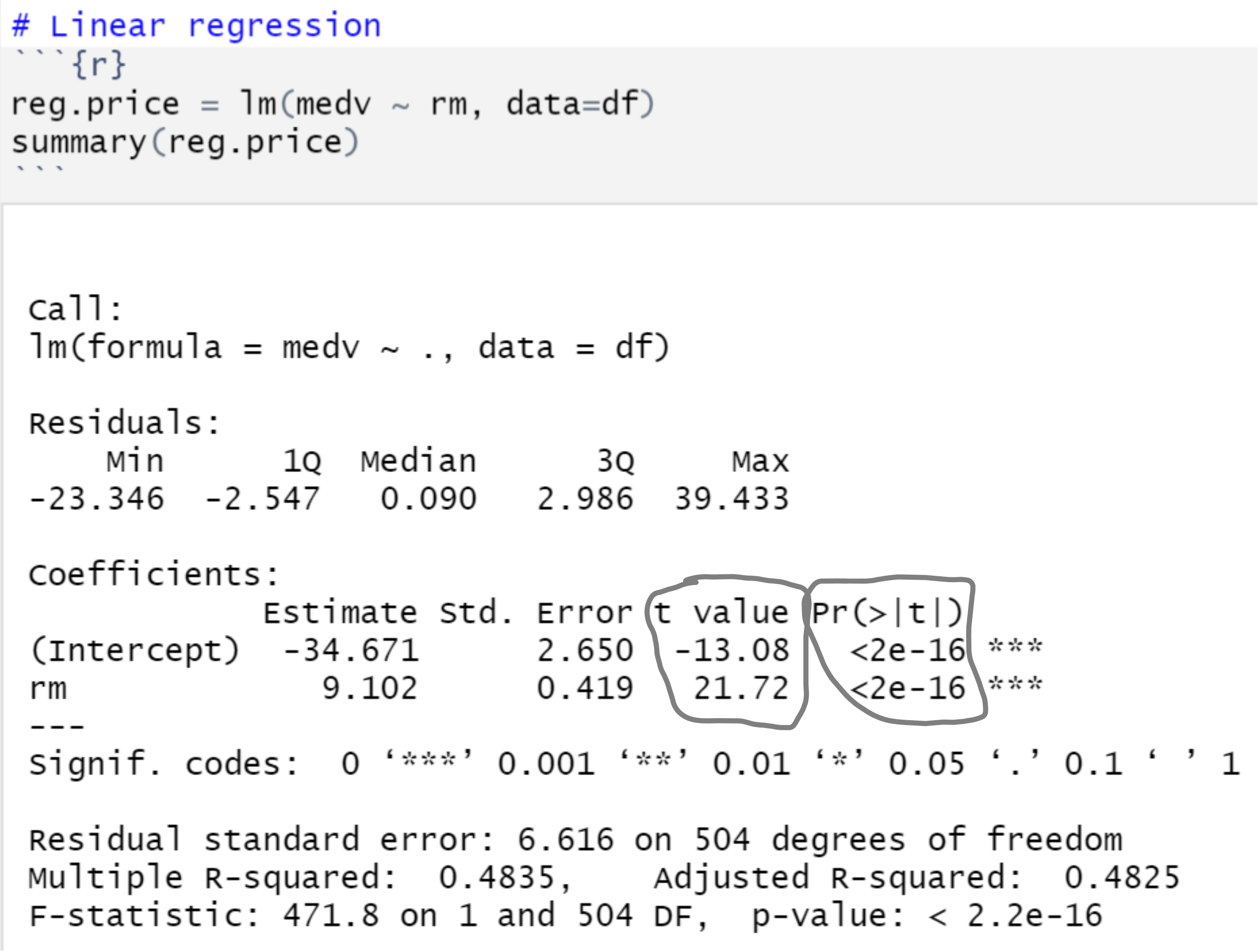
- After the regression parameters and their uncertainties have been estimated, there are typically two fundamental questions:
  - Is X_i an informative feature or not? (*i.e.*, whether $\hat{\beta}_i$ might be 0 or not)

    → *Statistical test*
  - Which values of $\hat{\beta}$ are compatible with the data?

    → *Confidence interval*

Testing and Confidence Intervals

Let’s talk about the statistical test, first



~~~
# Linear regression
```{r}
reg.price = lm(medv ~ rm, data=df)
summary(reg.price)
```
~~~

```
Call:
lm(formula = medv ~ ., data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-23.346  -2.547   0.090   2.986  39.433 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -34.671      2.650  -13.08   <2e-16 ***
rm             9.102      0.419   21.72   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.616 on 504 degrees of freedom
Multiple R-squared:  0.4835,    Adjusted R-squared:  0.4825 
F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
```

Testing and Confidence Intervals

Testing the effect of a covariate

- **Null hypothesis**

$$H_0: \quad \beta_1 = 0$$

In our case, $H_0 =$ “There is no relationship between X_i and Y”

- **Alternative hypothesis**

$$H_A: \quad \beta_1 \neq 0$$

In our case, $H_1 =$ “There is some relationship between X_i and Y”

Learned in TMA4240 (Statistics)

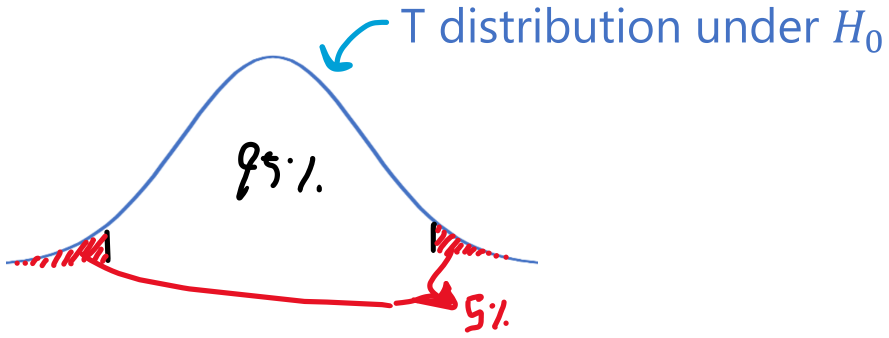
Testing and Confidence Intervals

- To carry out a statistical test, we need a *test statistic*.

  This is some type of summary statistic that follows a known distribution under H_0.

  For our purpose, we use the ***T-statistic***.

$$T = \frac{\hat{\beta}_1}{\mathrm{SE}(\hat{\beta}_1)}$$



- *Note:* If you want to test against another value c than $\beta_1 = 0$, the formula is

$$T = \frac{\hat{\beta}_1 - c}{\mathrm{SE}(\hat{\beta}_1)}$$

Testing and Confidence Intervals

- **Null hypothesis**

$$H_0: \quad \beta_1 = 0$$

In our case, $H_0 =$ "There is no relationship between X_i and Y"

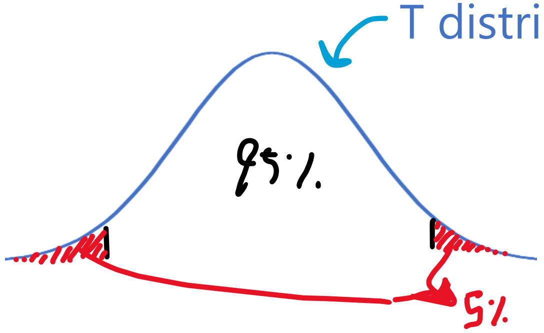
- **Alternative hypothesis**

$$H_A: \quad \beta_1 \neq 0$$

In our case, $H_1 =$ "There is some relationship between X_i and Y"

- Our question is

"Is X_i an informative feature or not? (*i.e.*, whether $\hat{\beta}_i$ might be 0 or not)?"



T distribution under H_0

$$T = \frac{\hat{\beta}_1}{\mathrm{SE}(\hat{\beta}_1)}$$

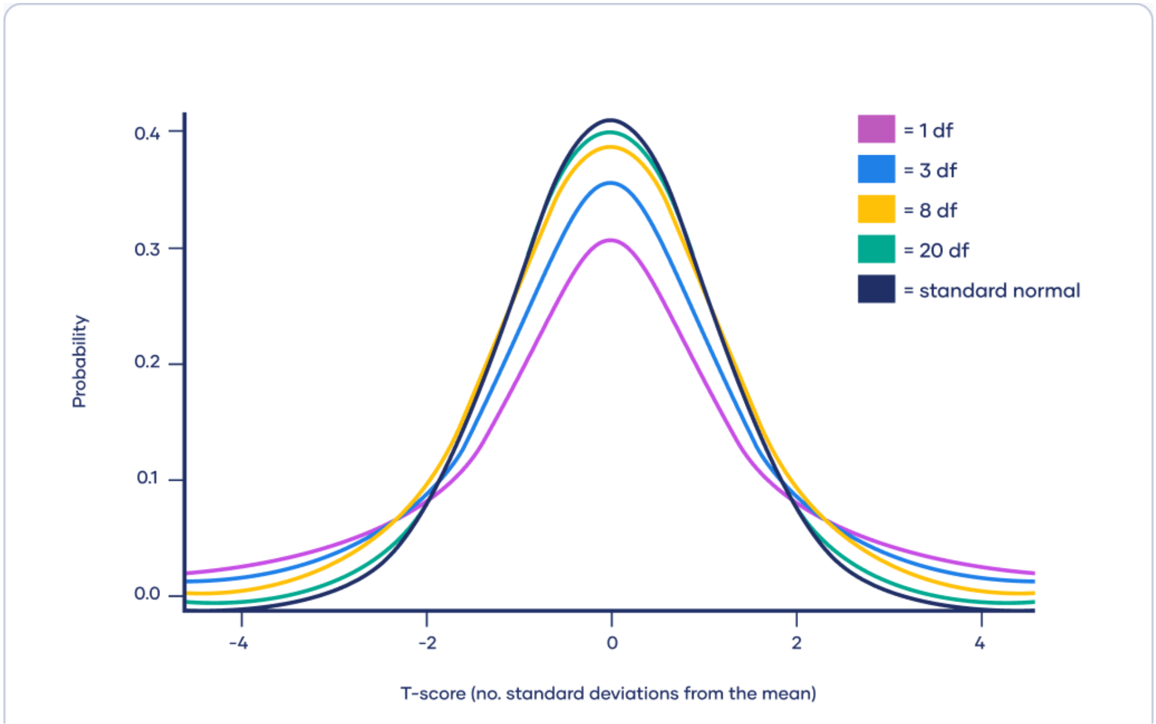
(recap) the t-distribution with $n - p$ degrees of freedom.
(n: num. of data samples; p: num. of learnable parameters)

- If T falls into the 95% region, then we *cannot* reject H_0
→ X_1 is not informative.
- If T falls into the 5% region (red), then we *can* reject H_0
→ X_1 is informative.

Learned in TMA4240 (Statistics)

Testing and Confidence Intervals

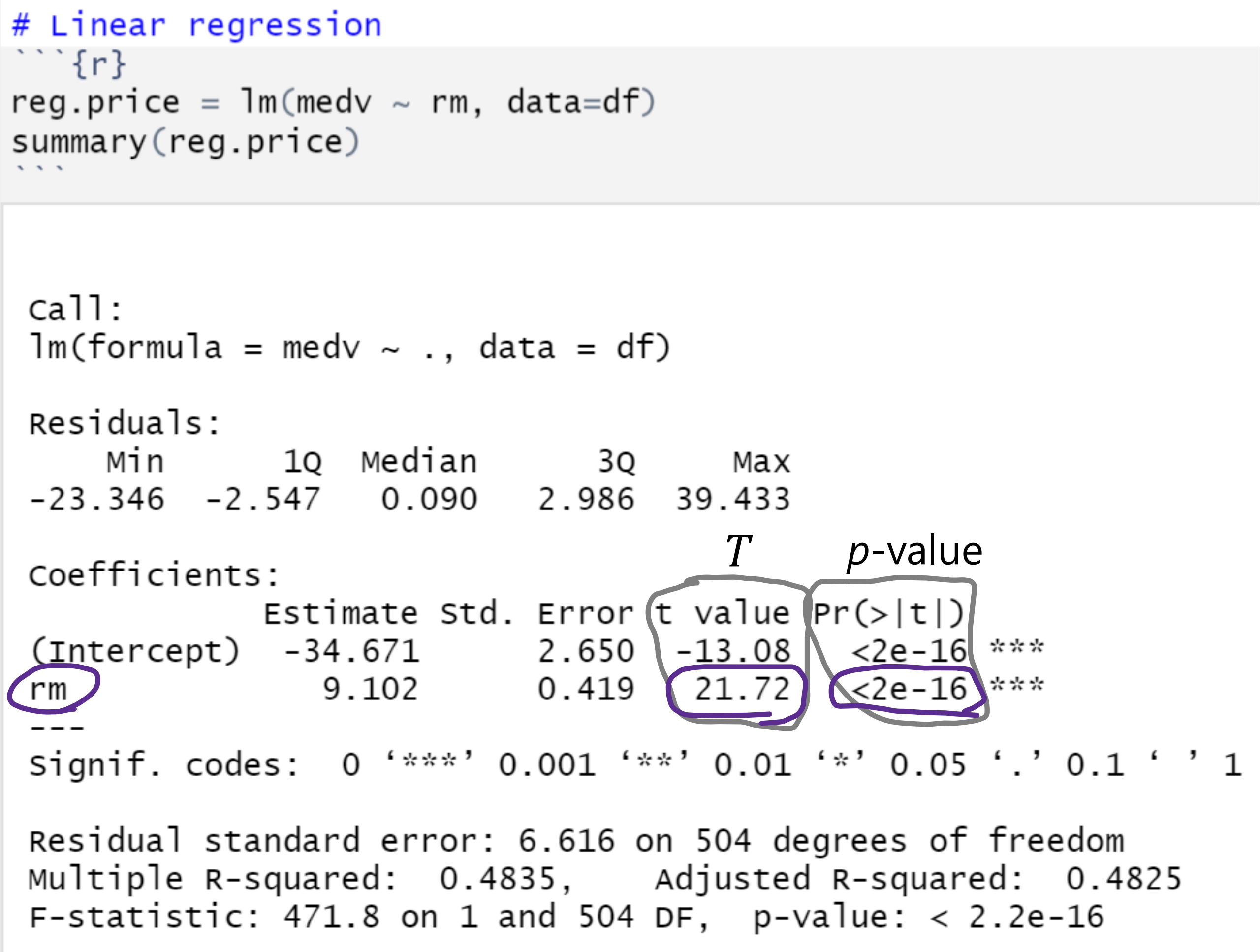
Recap: The *t*-distribution



- The *t*-distribution has heavier tails than the normal distribution
- For degrees-of-freedom (df) ≥ 30, the *t* and normal distributions are quite similar.

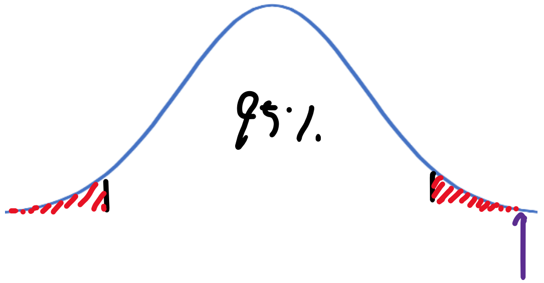
Testing and Confidence Intervals

Hypothesis tests for the example



~~~
# Linear regression
```{r}
reg.price = lm(medv ~ rm, data=df)
summary(reg.price)
```
~~~

```
Call:
lm(formula = medv ~ ., data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-23.346  -2.547   0.090   2.986  39.433 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -34.671      2.650  -13.08   <2e-16 ***
rm             9.102      0.419   21.72   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.616 on 504 degrees of freedom
Multiple R-squared:  0.4835,	Adjusted R-squared:  0.4825 
F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
```



$$T = \frac{\hat{\beta}_{rm}}{\mathrm{SE}(\hat{\beta}_{rm})}$$

Testing and Confidence Intervals

Cautionary notes regarding *p*-values

- The (mis)use of *p*-values is heavily under critique in the scientific community!
- Simple yes/not decisions do often stand on very wiggly scientific ground.
  - We often reject H_0 when *p*-value is smaller than 0.05.
  - But what if you got *p*-value of 0.051 ?
  - It’s not the best idea to do hard-cut-off.
    Better to be “*soft*”.

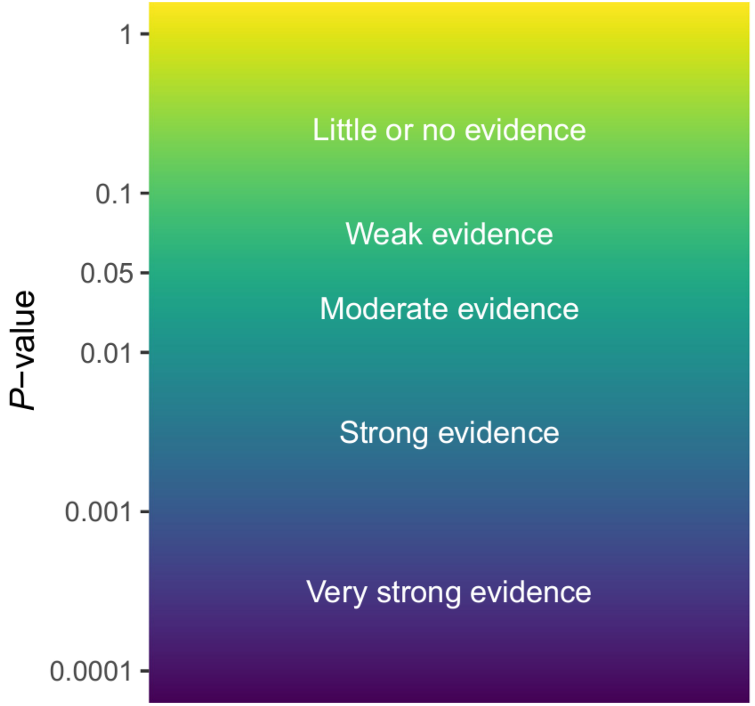


Figure 1. Suggested ranges to approximately translate the *P*-value into the language of evidence. The ranges are based on Bland (1986) [27], but the boundaries should not be understood as hard thresholds.

Muff, Stefanie, et al. "Rewriting results sections in the language of evidence." Trends in ecology & evolution (2021).

Testing and Confidence Intervals

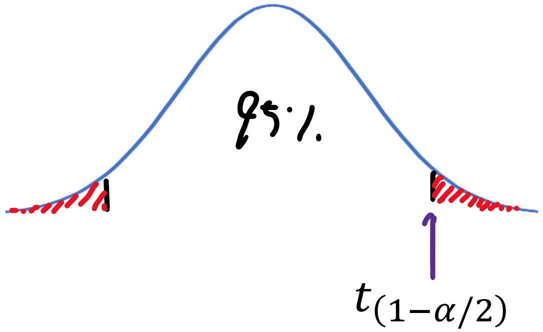
Confidence Intervals

- Confidence intervals (CIs) are the range of $\hat{\beta}$ that are *compatible with the data*.
- The t-distribution can be used to create confidence intervals for $\hat{\beta}$.

$$\hat{\beta} \pm t_{(1-\alpha/2),\, n-p} \cdot \mathrm{SE}(\hat{\beta})$$

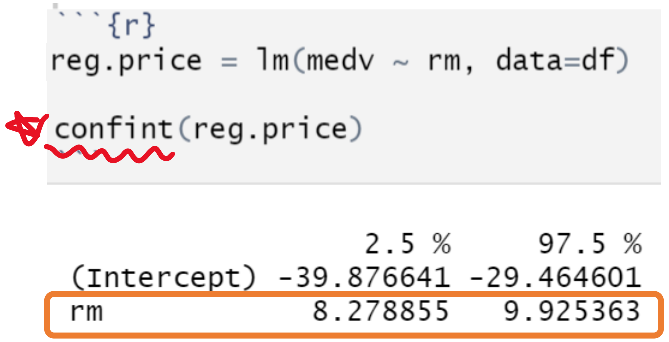
α can be 0.95 (*i.e.*, 95% confidence interval)

$n - p$: degree-of-freedom



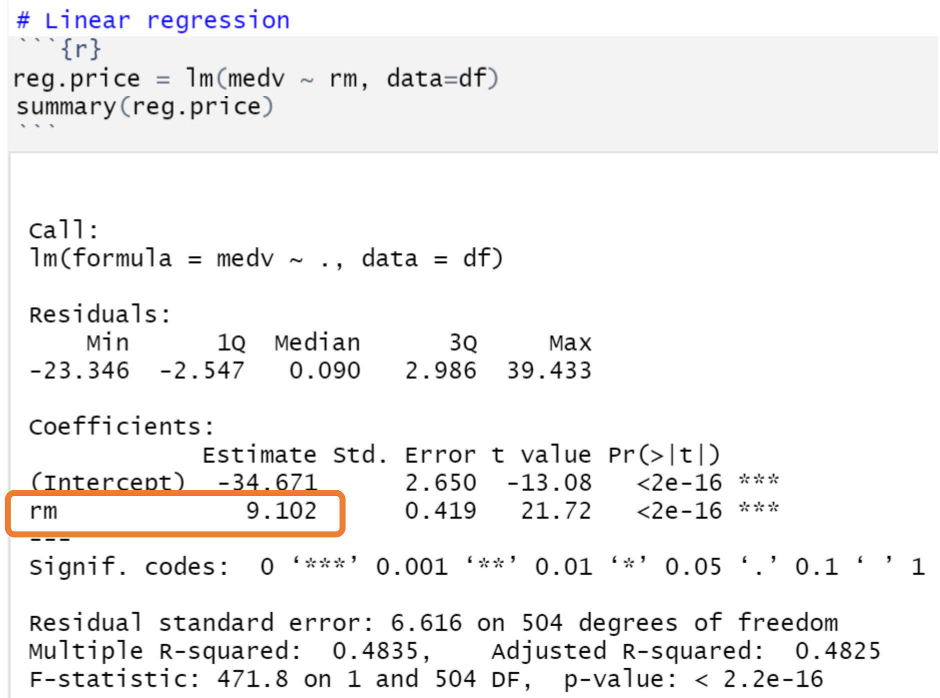
Testing and Confidence Intervals

Confidence Interval in R



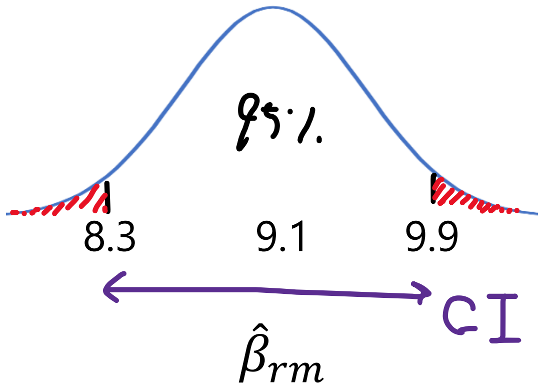
~~~
```{r}
reg.price = lm(medv ~ rm, data=df)

confint(reg.price)
~~~

```
                  2.5 %     97.5 %
 (Intercept) -39.876641 -29.464601
 rm            8.278855   9.925363
```



~~~
# Linear regression
```{r}
reg.price = lm(medv ~ rm, data=df)
summary(reg.price)
```
~~~

```
Call:
lm(formula = medv ~ ., data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-23.346  -2.547   0.090   2.986  39.433 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -34.671      2.650  -13.08   <2e-16 ***
rm             9.102      0.419   21.72   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.616 on 504 degrees of freedom
Multiple R-squared:  0.4835,	Adjusted R-squared:  0.4825 
F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
```



Model Accuracy

Model Accuracy

Measured by

- **Residual Standard Error (RSE)**

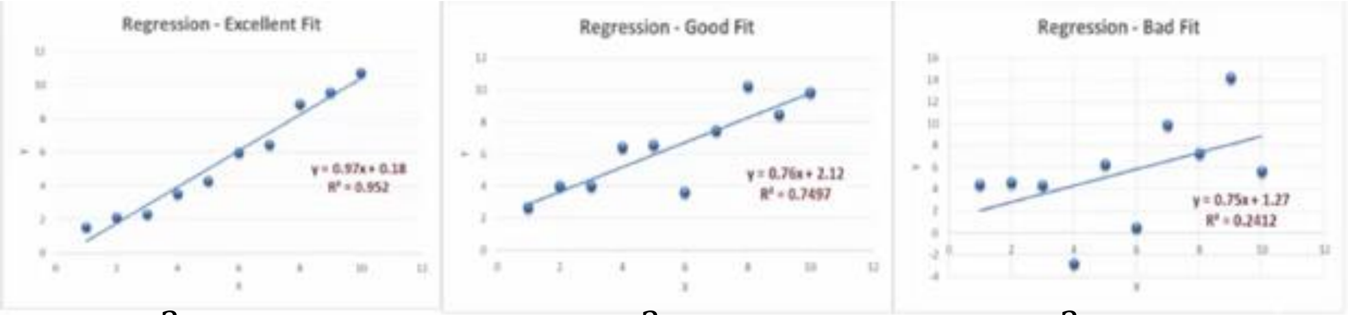
  It provides an absolute measure of *lack of fit*.

$$\mathrm{RSE} = \sqrt{\frac{1}{n-2}\mathrm{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- $\boldsymbol{R^2}$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$$

  R^2 typically ranges between 0 and 1.



$R^2 = 0.95$ $R^2 = 0.75$ $R^2 = 0.24$

Multiple Linear Regression

Multiple Linear Regression

Model

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

where X_j is the j-th predictor and β_j is the respective regression coefficient.

$$\begin{array}{|c|}\hline y_1 \\ \hline y_2 \\ \hline \vdots \\ \hline y_n \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|}\hline 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ \hline 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline 1 & x_{n1} & x_{n2} & \cdots & x_{np} \\ \hline \end{array} \cdot \begin{array}{|c|}\hline \beta_0 \\ \hline \beta_1 \\ \hline \vdots \\ \hline \beta_p \\ \hline \end{array} + \begin{array}{|c|}\hline \epsilon_1 \\ \hline \epsilon_2 \\ \hline \vdots \\ \hline \epsilon_n \\ \hline \end{array}$$

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$$

Multiple Linear Regression

Notation

- $\boldsymbol{Y}$: $(n \times 1)$ output
- $\boldsymbol{X}$: $(n \times (p + 1))$ input
- $\boldsymbol{\beta}$: $(\ (p + 1) \times 1)$ regression parameters
- $\boldsymbol{\epsilon}$: $(\ n \times 1)$ random errors

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$$

Multiple Linear Regression

Classical linear model

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$$

- Assumptions:
  1. $\mathrm{E}[\epsilon] = 0$
  2. $\mathrm{Cov}(\epsilon) = \mathrm{E}[\epsilon\epsilon^T] = \sigma^2\mathbf{I}$

     (*i.e.*, all ϵ_i are independent of each other and ϵ is homogenous)
  3. $\boldsymbol{X}$ has full rank, $\mathrm{rank}(\boldsymbol{X}) = p + 1$ (we assume $n \gg (p+1)$)

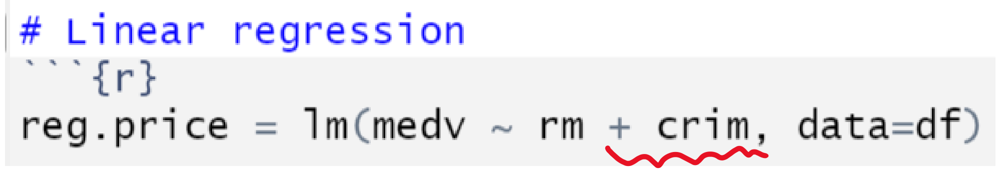
     (*i.e.*, all X_j are independent of each other)
  4. The classical *normal* linear regression model is obtained if additionally $\epsilon \sim N_n(\mathbf{0}, \sigma^2\mathbf{I})$ holds. N_n denotes the *n*-dimensional multivariate normal distribution.

Multiple Linear Regression

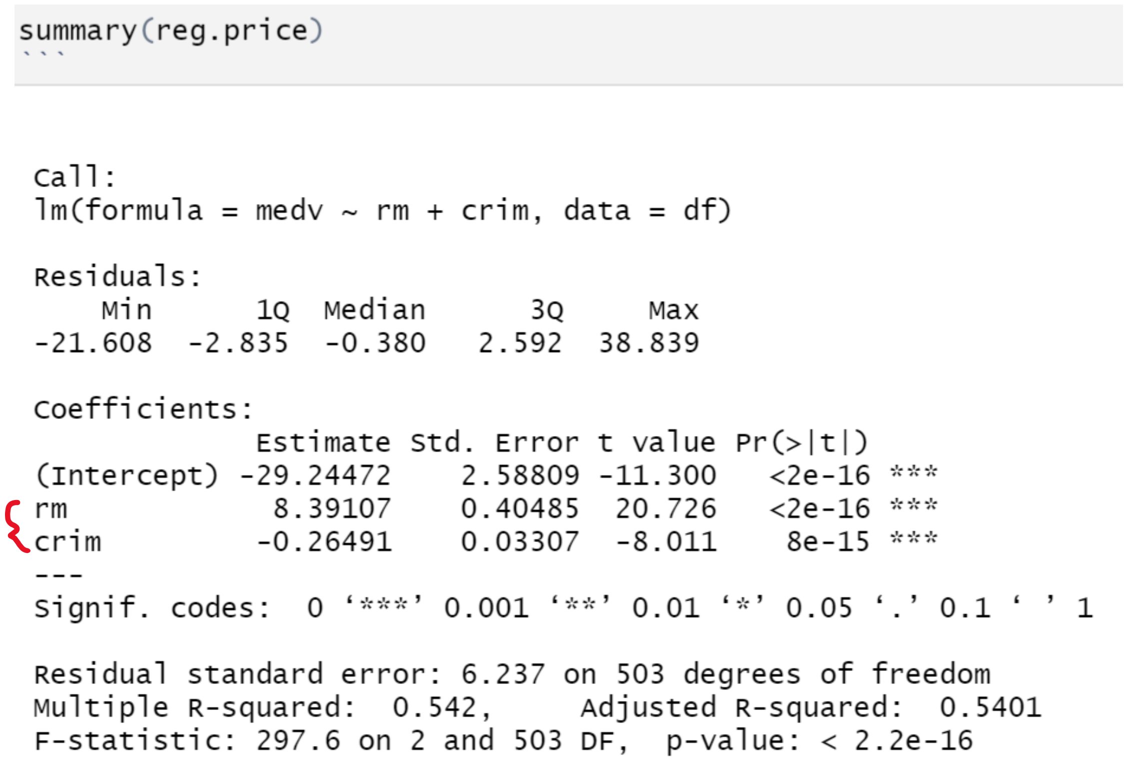
The Boston-housing example for two predictors (input features)

- We're looking at the regression model

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



~~~
# Linear regression
```{r}
reg.price = lm(medv ~ rm + crim, data=df)
~~~



```
summary(reg.price)
```
```

```
Call:
lm(formula = medv ~ rm + crim, data = df)

Residuals:
    Min   1Q Median   3Q   Max
-21.608 -2.835 -0.380   2.592 38.839

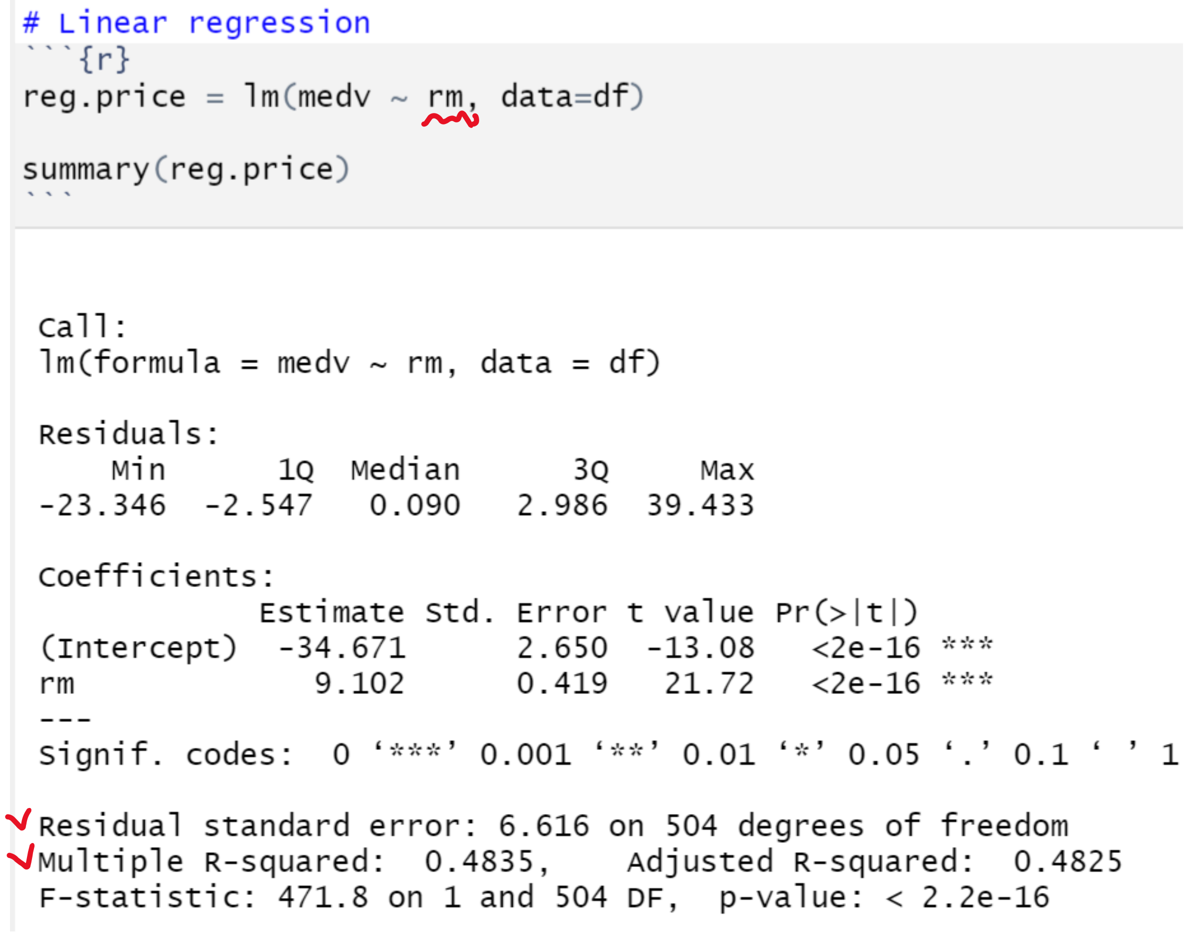
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -29.24472   2.58809 -11.300   <2e-16 ***
rm   8.39107   0.40485 20.726   <2e-16 ***
crim   -0.26491   0.03307 -8.011   8e-15 ***

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

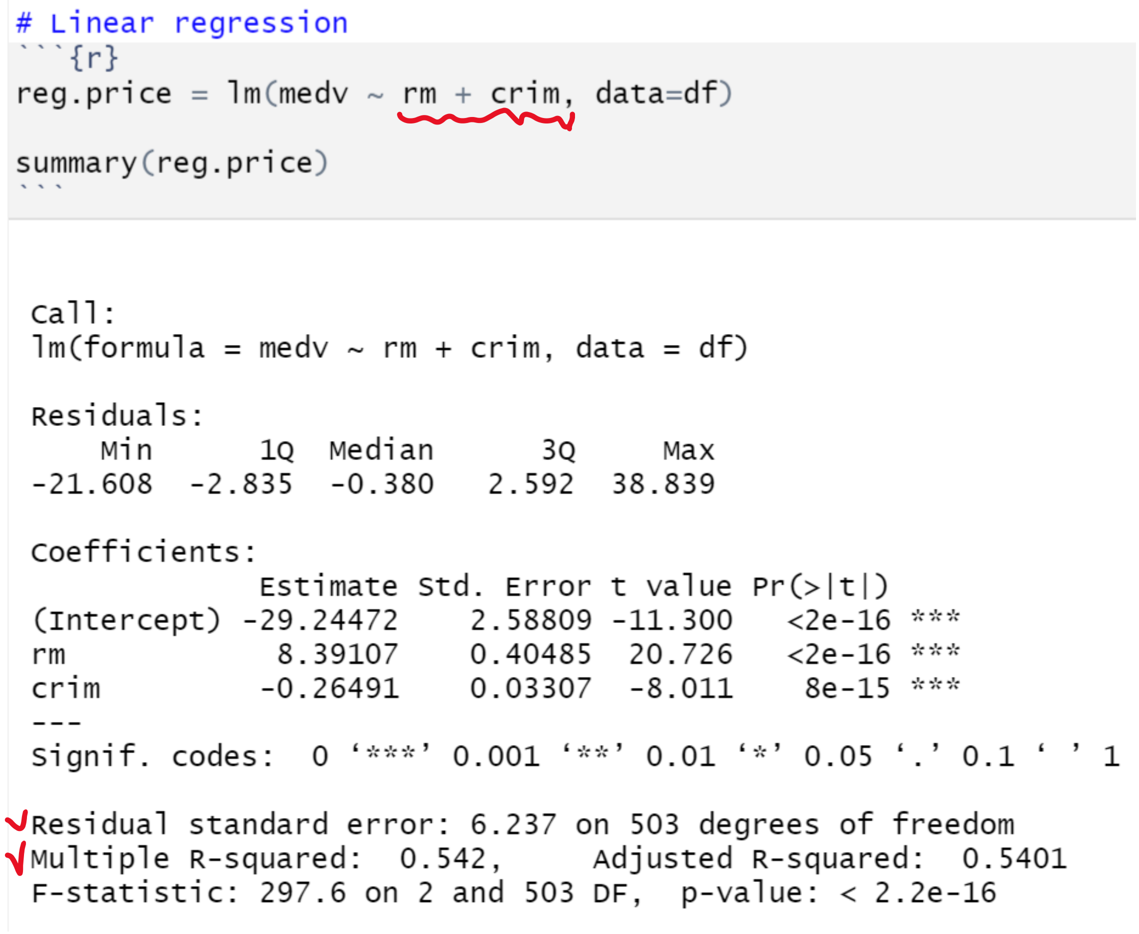
Residual standard error: 6.237 on 503 degrees of freedom
Multiple R-squared: 0.542,	Adjusted R-squared: 0.5401
F-statistic: 297.6 on 2 and 503 DF, p-value: < 2.2e-16
```
```

Multiple Linear Regression

The Boston-housing example for two predictors (input features)



~~~
# Linear regression
```{r}
reg.price = lm(medv ~ rm, data=df)

summary(reg.price)
```


Call:
lm(formula = medv ~ rm, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-23.346  -2.547   0.090   2.986  39.433 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -34.671      2.650  -13.08   <2e-16 ***
rm             9.102      0.419   21.72   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

✓Residual standard error: 6.616 on 504 degrees of freedom
✓Multiple R-squared:  0.4835,	Adjusted R-squared:  0.4825 
F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
~~~



~~~
# Linear regression
```{r}
reg.price = lm(medv ~ rm + crim, data=df)

summary(reg.price)
```


Call:
lm(formula = medv ~ rm + crim, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-21.608  -2.835  -0.380   2.592  38.839 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -29.24472    2.58809 -11.300   <2e-16 ***
rm            8.39107    0.40485  20.726   <2e-16 ***
crim         -0.26491    0.03307  -8.011    8e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

✓Residual standard error: 6.237 on 503 degrees of freedom
✓Multiple R-squared:  0.542,	Adjusted R-squared:  0.5401 
F-statistic: 297.6 on 2 and 503 DF,  p-value: < 2.2e-16
~~~

Multiple Linear Regression

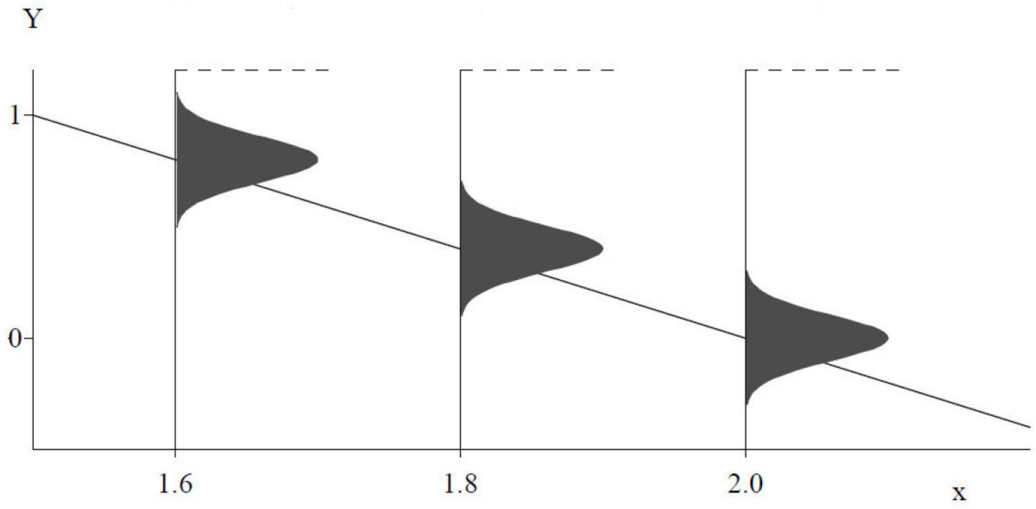
Distribution of the response vector

- For

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$$

where $\epsilon \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I})$

$$\boldsymbol{Y} \sim N_n(\boldsymbol{X\beta}, \sigma^2 \mathbf{I})$$



Multiple Linear Regression

Parameter estimation for $\boldsymbol{\beta}$

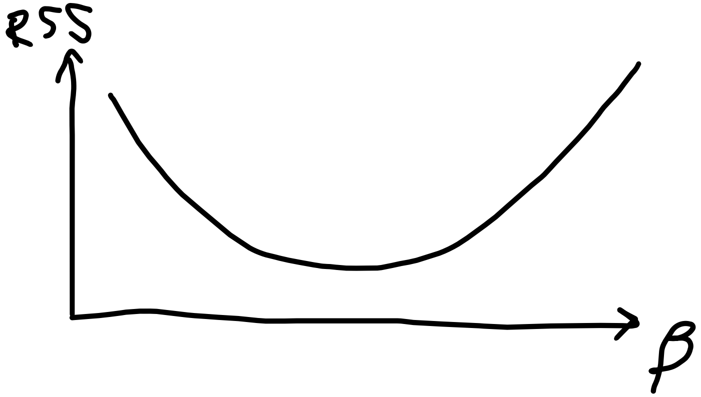
- In multiple linear regression, $\boldsymbol{\beta}$ is estimated with *maximum likelihood* and *least squares*.

- Aim: we want to estimate $\boldsymbol{\beta}$ by minimizing

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

then, $\boldsymbol{\beta}$ can be found by solving

$$\frac{\partial \text{RSS}}{\partial \boldsymbol{\beta}} = \mathbf{0}$$



Multiple Linear Regression

Parameter estimation for $\boldsymbol{\beta}$

- $\mathrm{RSS} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}\left(y_i - \boldsymbol{x}_i^T\widehat{\boldsymbol{\beta}}\right)^2 = \left(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\right)^T\left(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\right)$

- $\frac{\partial \mathrm{RSS}}{\partial \boldsymbol{\beta}} = \left(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\right)^T\left(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\right) = \boldsymbol{0}$

- $\mathrm{RSS} = \left(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\right)^T\left(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\right)$

$$= \left(\boldsymbol{Y}^T - \widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\right)\left(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\right)$$

$$= \boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{Y} + \widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}}$$

$\widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{Y} \rightarrow scalar$

$(1 \times p)\cdot(p \times n)\cdot(n \times 1) \rightarrow (1 \times 1)$

Therefore, $\widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{Y} = \left(\widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{Y}\right)^T$

$$= \boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \left(\widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{Y}\right)^T + \widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}}$$

$$= \boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{Y}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}}$$

$$= \boldsymbol{Y}^T\boldsymbol{Y} - 2\boldsymbol{Y}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}}$$

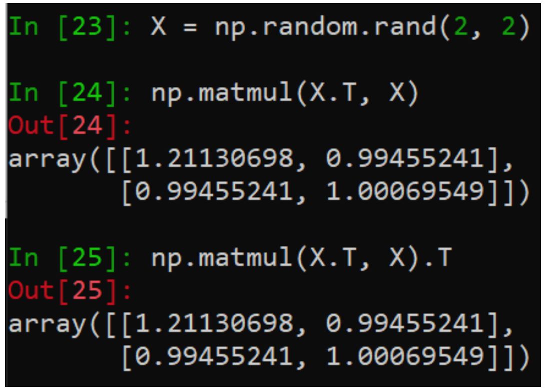
- $\frac{\partial \boldsymbol{A}^T\widehat{\boldsymbol{\beta}}}{\partial \widehat{\boldsymbol{\beta}}} = \boldsymbol{A}$
- $\frac{\partial \widehat{\boldsymbol{\beta}}^T\boldsymbol{A}\widehat{\boldsymbol{\beta}}}{\partial \widehat{\boldsymbol{\beta}}} = (\boldsymbol{A} + \boldsymbol{A}^T)\widehat{\boldsymbol{\beta}}$

$(\boldsymbol{X}^T\boldsymbol{X})^T = \boldsymbol{X}^T\boldsymbol{X}$

- $\frac{\partial \mathrm{RSS}}{\partial \widehat{\boldsymbol{\beta}}} = -2(\boldsymbol{Y}^T\boldsymbol{X})^T + (\boldsymbol{X}^T\boldsymbol{X} + (\boldsymbol{X}^T\boldsymbol{X})^T)\widehat{\boldsymbol{\beta}} = \boldsymbol{0}$

$$= -2(\boldsymbol{Y}^T\boldsymbol{X})^T + (\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{X}^T\boldsymbol{X})\widehat{\boldsymbol{\beta}} = \boldsymbol{0}$$

$$= -2(\boldsymbol{Y}^T\boldsymbol{X})^T + 2(\boldsymbol{X}^T\boldsymbol{X})\widehat{\boldsymbol{\beta}} = \boldsymbol{0}$$



```
In [23]: X = np.random.rand(2, 2)

In [24]: np.matmul(X.T, X)
Out[24]:
array([[1.21130698, 0.99455241],
       [0.99455241, 1.00069549]])

In [25]: np.matmul(X.T, X).T
Out[25]:
array([[1.21130698, 0.99455241],
       [0.99455241, 1.00069549]])
```

$\boldsymbol{X}^T\boldsymbol{X}$

$(\boldsymbol{X}^T\boldsymbol{X})^T$

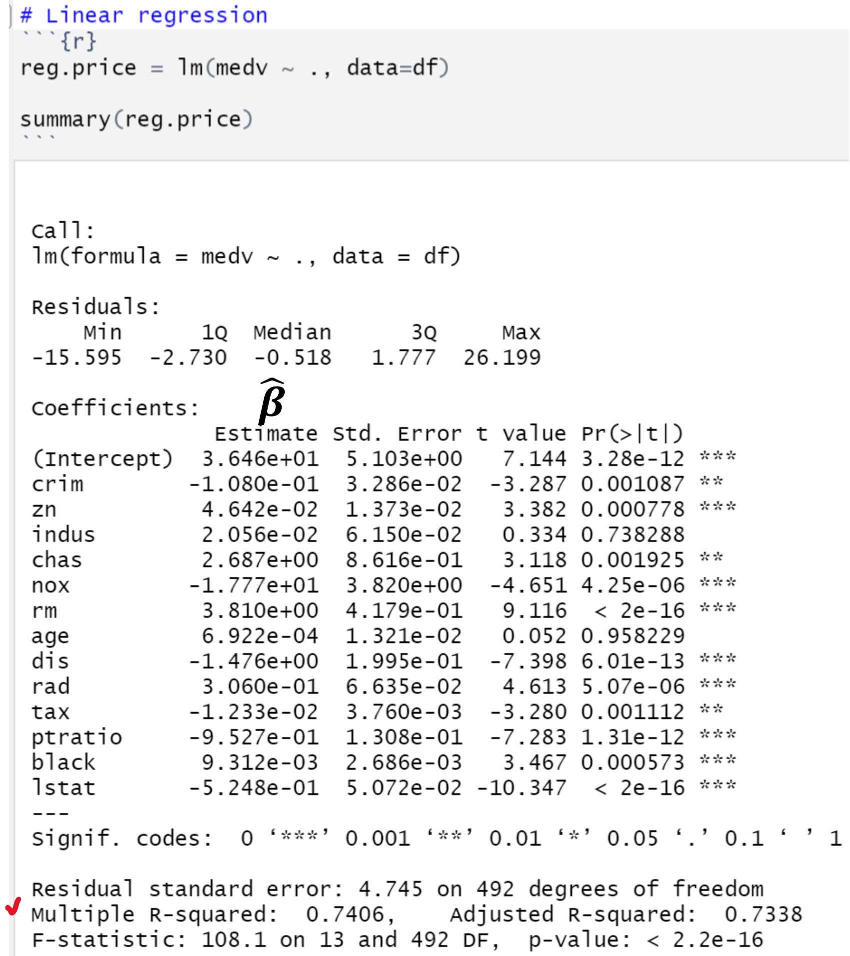
$$(\boldsymbol{Y}^T\boldsymbol{X})^T = (\boldsymbol{X}^T\boldsymbol{X})\widehat{\boldsymbol{\beta}}$$

$$(\boldsymbol{X}^T\boldsymbol{X})^{-1}(\boldsymbol{Y}^T\boldsymbol{X})^T = \widehat{\boldsymbol{\beta}}$$

Multiple Linear Regression

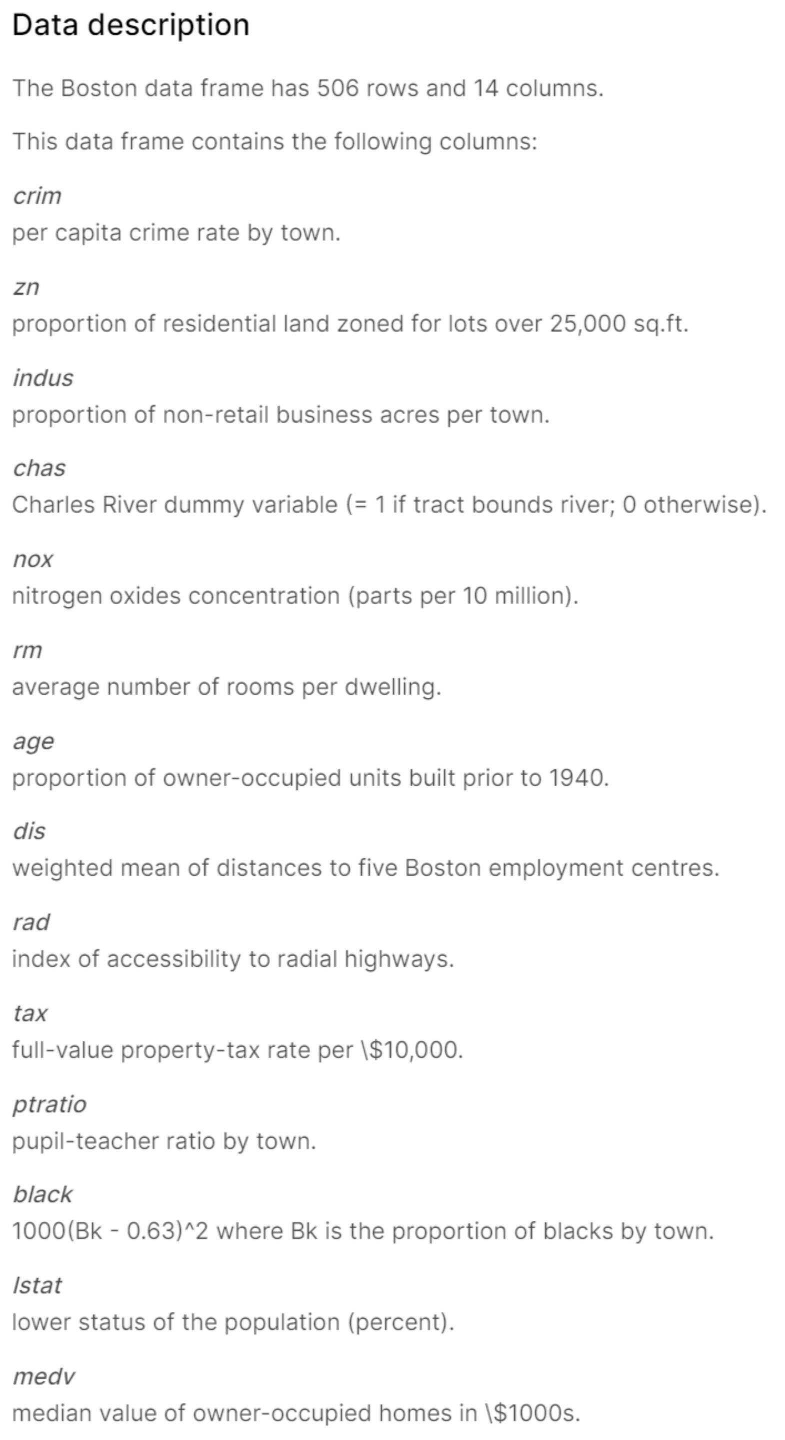
Example continued

- Housing price prediction using all the input features
- Simple to implement,
  yet important to really know
  what you're using!



~~~
# Linear regression
```{r}
reg.price = lm(medv ~ ., data=df)

summary(reg.price)
```
~~~

```
Call:
lm(formula = medv ~ ., data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.595  -2.730  -0.518   1.777  26.199 

Coefficients:     β̂
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
crim        -1.080e-01  3.286e-02  -3.287 0.001087 ** 
zn           4.642e-02  1.373e-02   3.382 0.000778 ***
indus        2.056e-02  6.150e-02   0.334 0.738288    
chas         2.687e+00  8.616e-01   3.118 0.001925 ** 
nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
age          6.922e-04  1.321e-02   0.052 0.958229    
dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
tax         -1.233e-02  3.760e-03  -3.280 0.001112 ** 
ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
black        9.312e-03  2.686e-03   3.467 0.000573 ***
lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.745 on 492 degrees of freedom
✓ Multiple R-squared:  0.7406,	Adjusted R-squared:  0.7338 
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```



Data description

The Boston data frame has 506 rows and 14 columns.

This data frame contains the following columns:

crim
per capita crime rate by town.

zn
proportion of residential land zoned for lots over 25,000 sq.ft.

indus
proportion of non-retail business acres per town.

chas
Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).

nox
nitrogen oxides concentration (parts per 10 million).

rm
average number of rooms per dwelling.

age
proportion of owner-occupied units built prior to 1940.

dis
weighted mean of distances to five Boston employment centres.

rad
index of accessibility to radial highways.

tax
full-value property-tax rate per \$10,000.

ptratio
pupil-teacher ratio by town.

black
1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town.

lstat
lower status of the population (percent).

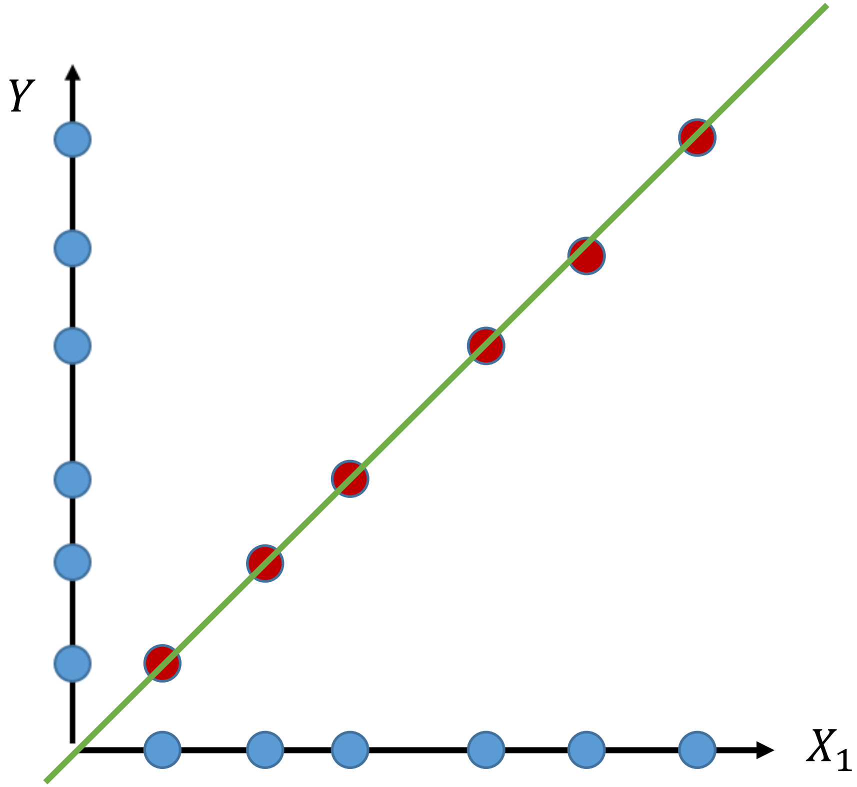
medv
median value of owner-occupied homes in \$1000s.

The End!

Linear Regression

- One quantitative response Y
- One covariate X_1
- Linear relationship between X_1 and Y

Then,



$$Y = \beta_0 + \beta_1 X_1$$

Linear Regression

Least Squares Estimators:

- Analytical solution:
  - $\boldsymbol{Y} = \beta_0 + \boldsymbol{X\beta} + \epsilon$
    - $\boldsymbol{X} \in \mathbb{R}^{n \times p}$
    - $\boldsymbol{\beta} \in \mathbb{R}^{p \times 1}$
    - $\boldsymbol{Y} \in \mathbb{R}^{n \times 1}$
  - You want to find $\widehat{\boldsymbol{Y}} = \hat{\beta}_0 + \widehat{\boldsymbol{X}}\widehat{\boldsymbol{\beta}}$
    - $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T Y$
    - $\hat{\beta}_0 = \overline{\boldsymbol{Y}} - \widehat{\boldsymbol{\beta}}\overline{\boldsymbol{X}}$
  - You've learned it in your previous statistics course. So, we're skipping the proof. You can find the proof here: https://statproofbook.github.io/P/mlr-ols
- Numerical Solution: Gradient Descent (we'll learn about it during the module for neural networks)