

# Linear Regression

IGP Biostatistics Teaching Demo

Katie Evans

6.22.21



[https://github.com/katieevans/IGP\\_biostatistics](https://github.com/katieevans/IGP_biostatistics)

# Course materials: slides and code

The screenshot shows a GitHub repository page for 'katiesevans / IGP\_biostatistics'. The repository has 1 branch and 0 tags. The README.md file contains the following content:

```
Quantitative Biology: Statistics and Data Analysis for Life Scientists

This repository contains all the course materials for IGP-484 (2021). You can find both the lecture slides and the code to generate all plots and statistical analyses in the designated date folder. All analyses are performed in R and require previous installation of tidyverse. If you don't have tidyverse, you can install the package with install.packages("tidyverse") or you can install the specific packages required for this class: install.packages(c("dplyr", "tidyverse", "ggplot2")).

Data analysis with the Tidyverse

Looking for help with basic data wrangling in R? New to the "Tidyverse"? Check out the course materials from my workshop for NUIT for step-by-step help and lots of examples and practice questions.

Questions?

For all questions, contact Katie at kathryn.evans@northwestern.edu
```

The repository has 1 commit by katieevans, updating README.md 5 hours ago. It also contains a folder '20210622\_demo' containing 'first draft slides' and a file 'README.md' updated 5 hours ago.

**About**: No description, website, or topics provided. Readme.

**Releases**: No releases published. Create a new release.

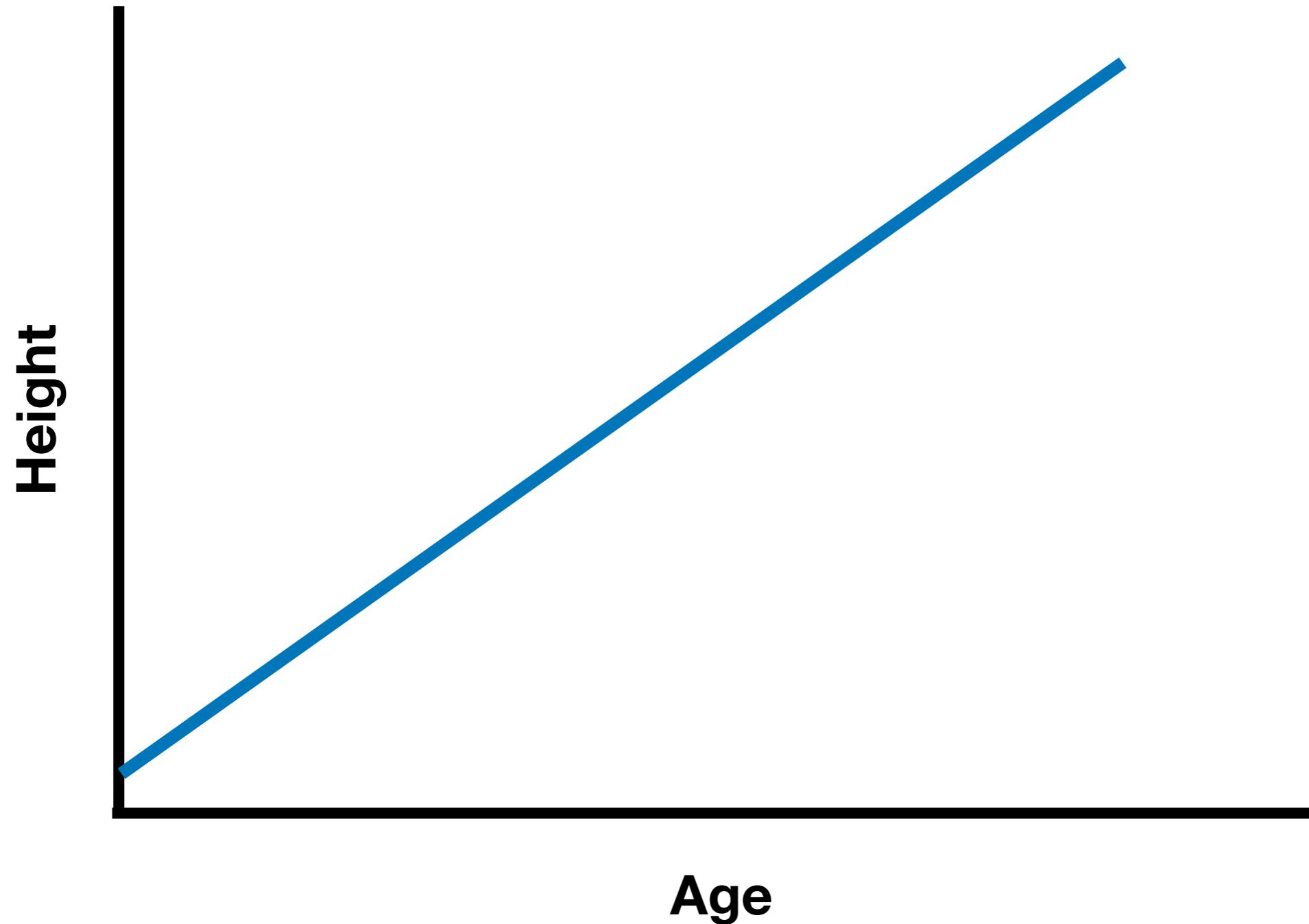
**Packages**: No packages published. Publish your first package.

**Languages**: R 100.0%

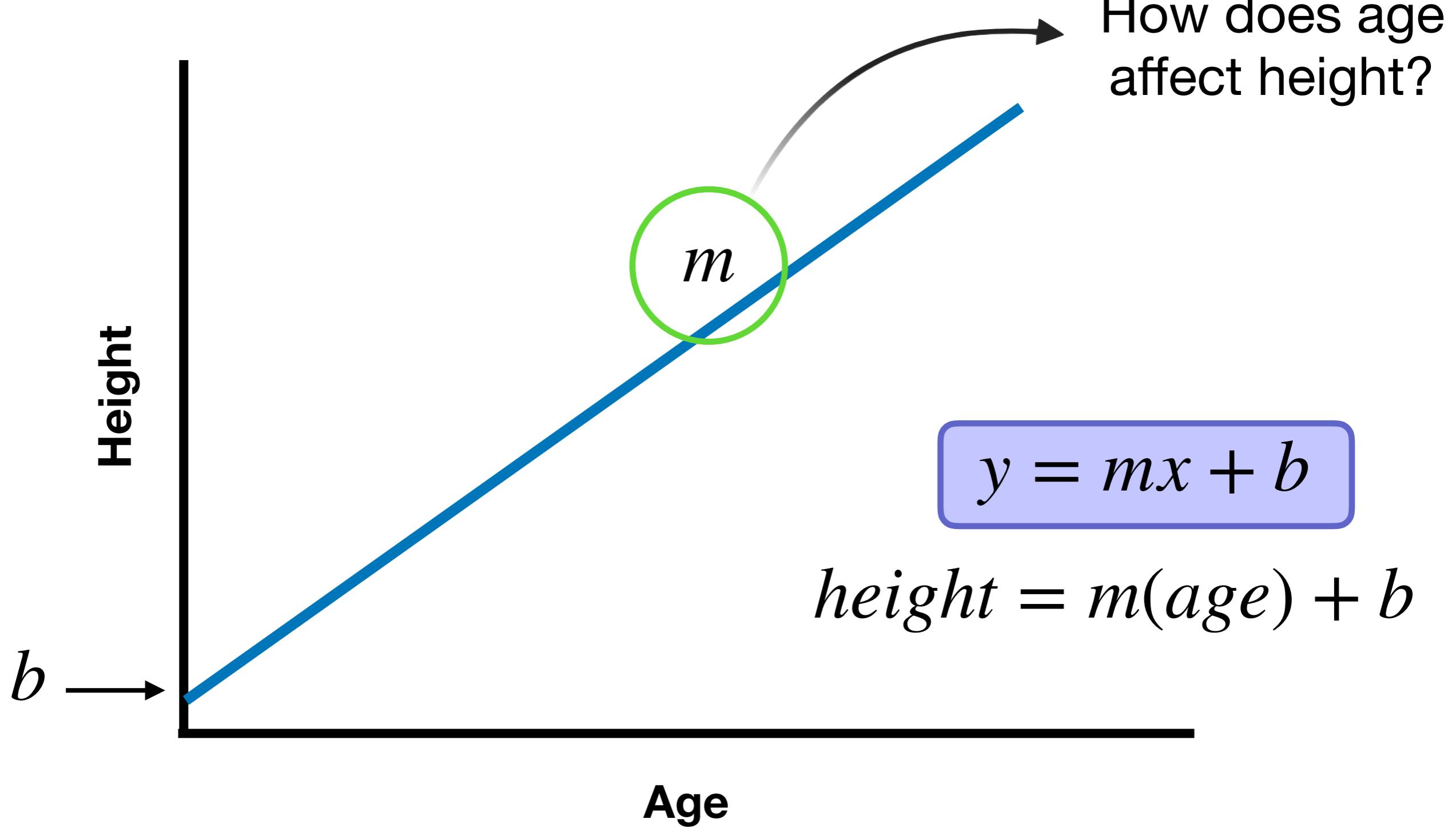


[https://github.com/katiesevans/IGP\\_biostatistics](https://github.com/katiesevans/IGP_biostatistics)

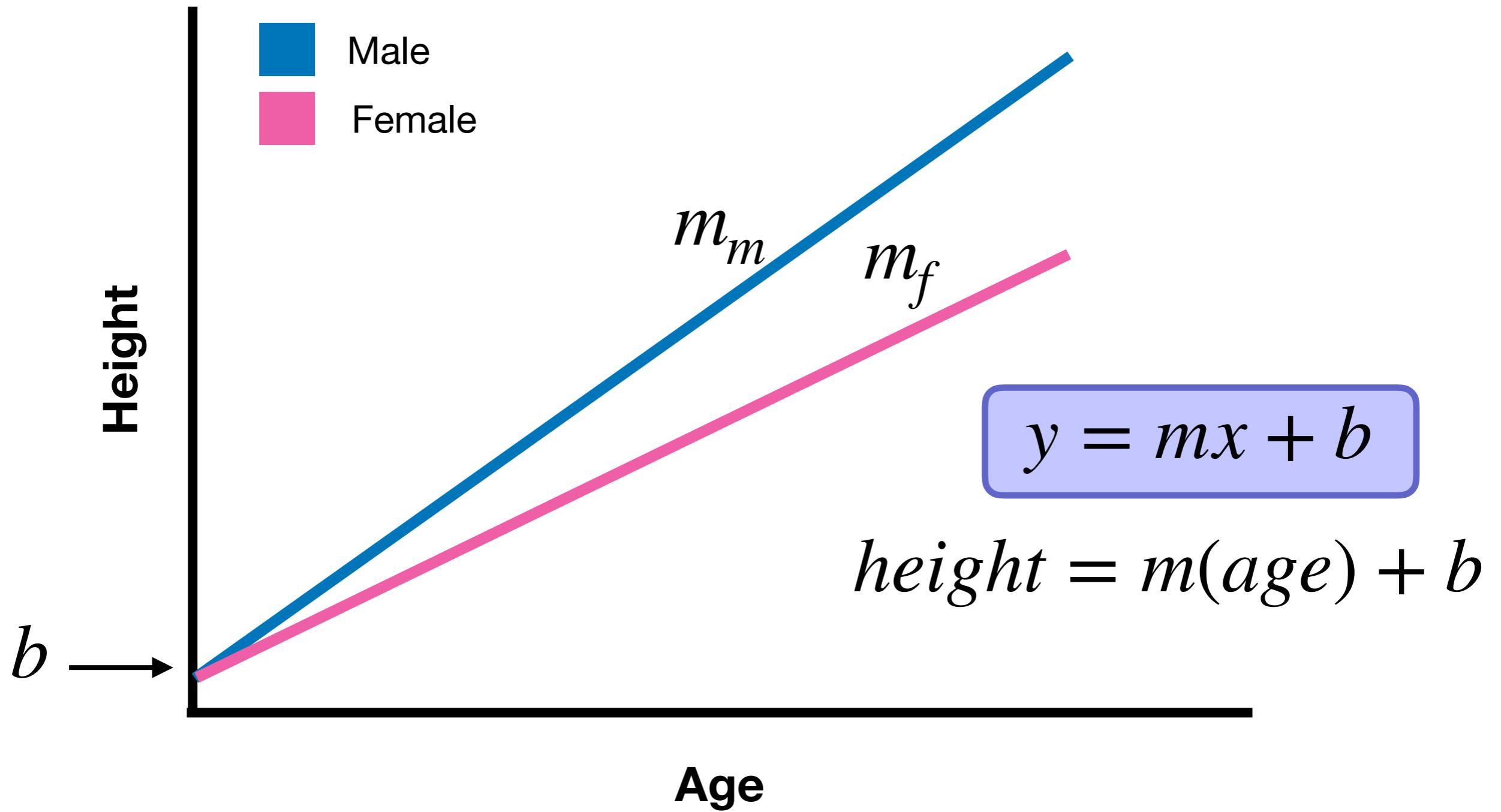
# Linear relationships



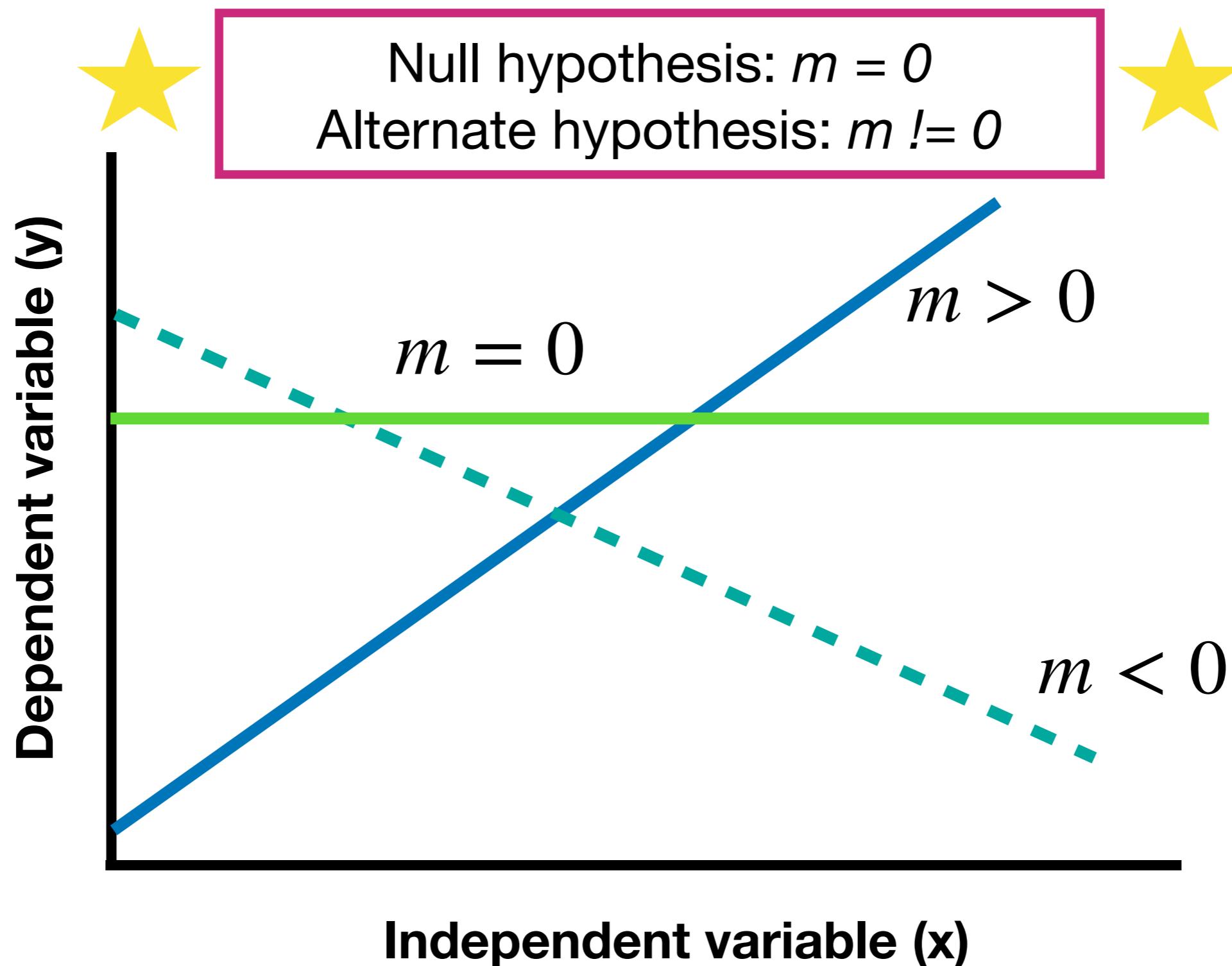
# Linear relationships



# Linear relationships



# Linear regression tests for a relationship between variables



# Linear regression: an example



Independent Variable

$$\text{?} =$$



Dependent Variable

# Linear regression: an example

- Survey 500 people with a range of incomes
- Ask them to rank their happiness on a scale from 1 to 10



Independent Variable

?  
=



Dependent Variable

# Linear regression: an example

```
income_data <- read.csv("income.data.csv")
```

| income   | happiness |
|----------|-----------|
| 3.862647 | 2.3144890 |
| 4.979381 | 3.4334898 |
| 4.923957 | 4.5993734 |
| 3.214372 | 2.7911138 |
| 7.196409 | 5.5963983 |
| 3.729643 | 2.4585559 |
| 4.674517 | 3.1929918 |
| 4.498104 | 1.9071368 |
| 3.121631 | 2.9424499 |
| 4.639914 | 3.7379416 |
| 4.632840 | 3.1754061 |
| 2.773179 | 2.0090465 |
| 7.119479 | 5.9518141 |

```
summary(income_data$income)
```

| Min.  | 1st Qu. | Median | Mean  | 3rd Qu. | Max.  |
|-------|---------|--------|-------|---------|-------|
| 1.506 | 3.006   | 4.424  | 4.467 | 5.992   | 7.482 |

*(data is in \$10k)*

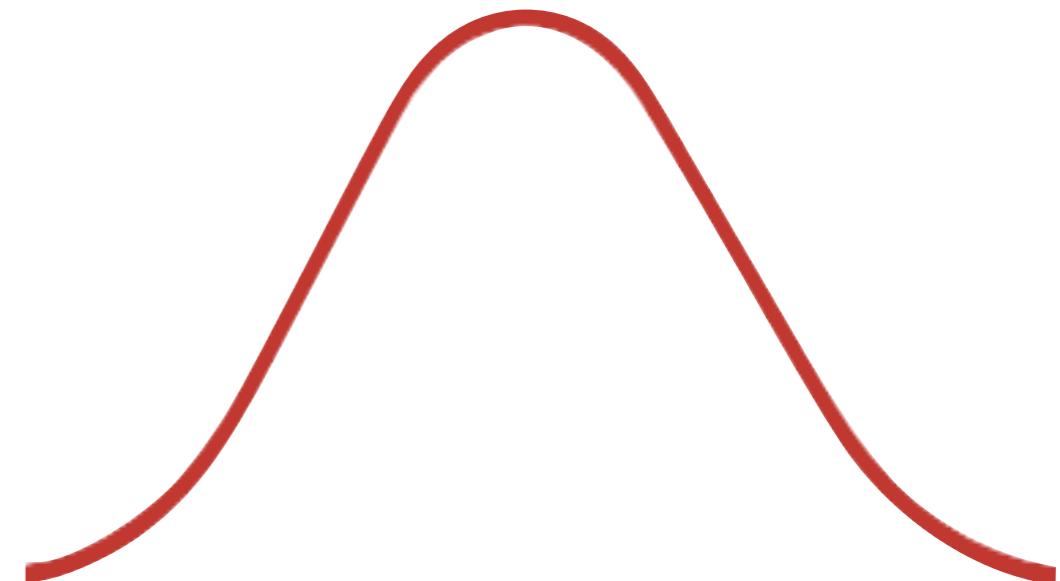
```
summary(income_data$happiness)
```

| Min.  | 1st Qu. | Median | Mean  | 3rd Qu. | Max.  |
|-------|---------|--------|-------|---------|-------|
| 0.266 | 2.266   | 3.473  | 3.393 | 4.503   | 6.863 |

*(no very happy people...)*

# Assumptions for linear regression

- Homogeneity of variance
  - *The error is similar across all samples*
- Independence of observations
  - *No hidden groups or relationships*
- Normality
  - *Data follows normal distribution*



# The experiment setup

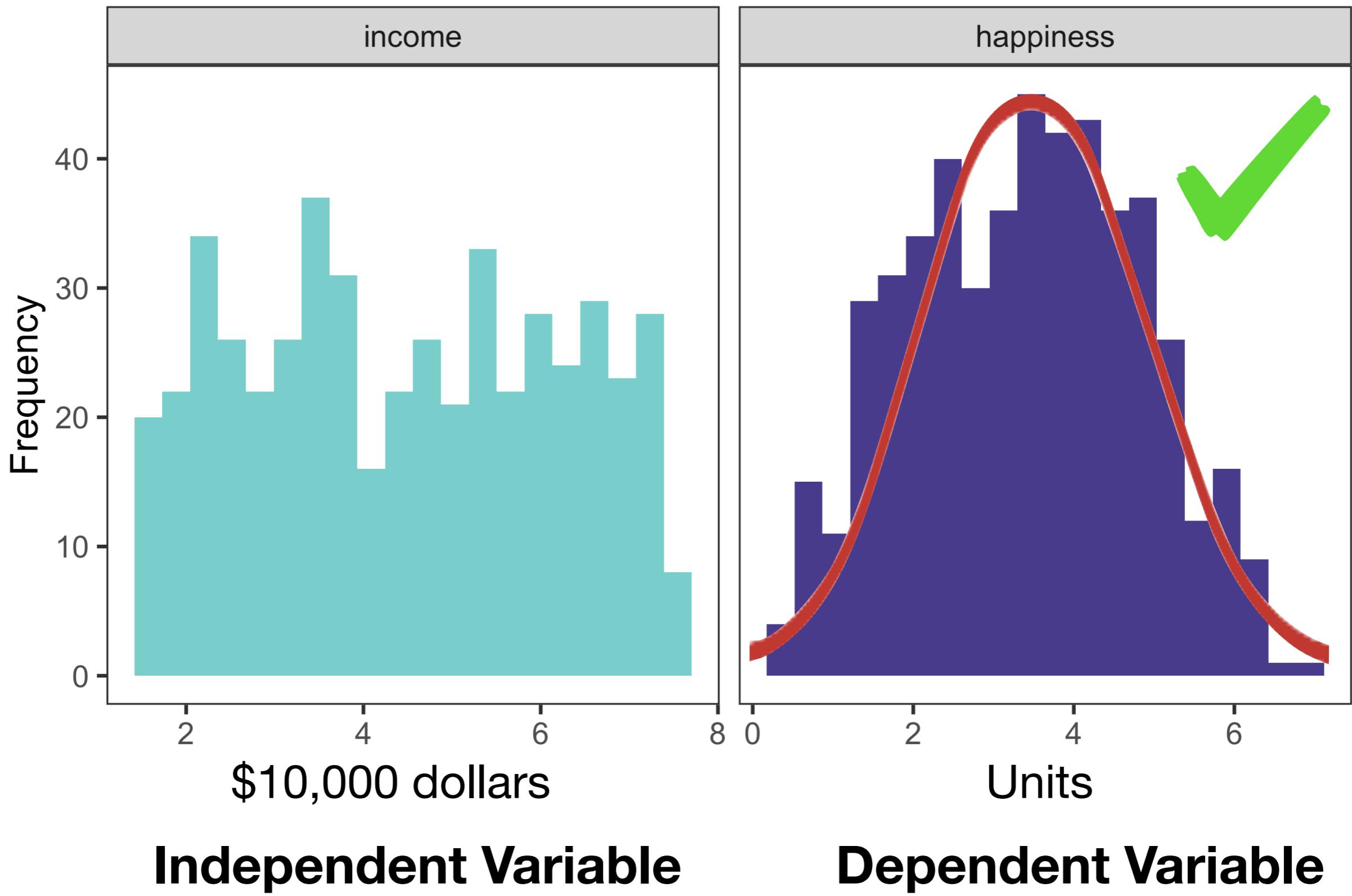
## Base R

```
hist(income_data$income)  
hist(income_data$happiness)
```

## ggplot2

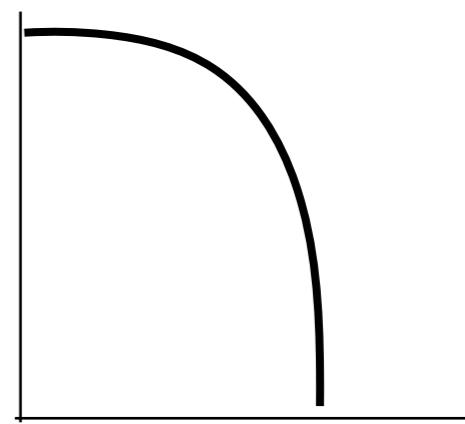
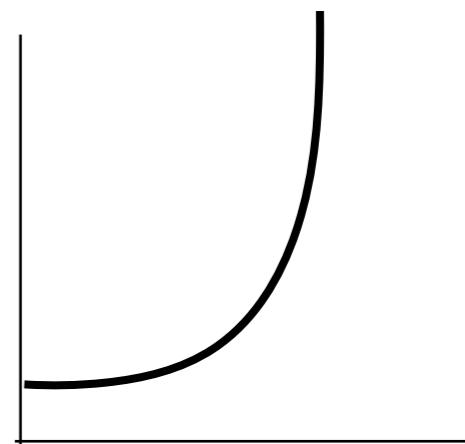
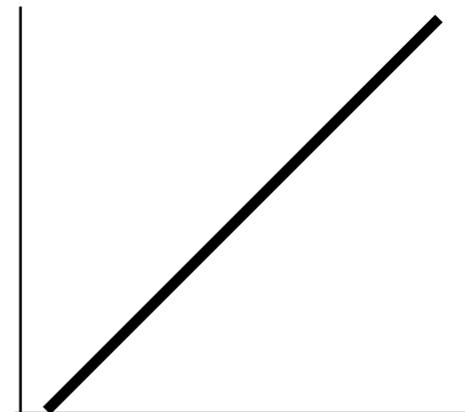
```
income_data %>%  
  tidyverse::gather(var, value) %>%  
  ggplot2::ggplot(.) +  
  ggplot2::aes(x = value) +  
  ggplot2::geom_histogram() +  
  ggplot2::facet_grid(var)
```

# The experiment setup



# Assumptions for linear regression

- Homogeneity of variance
  - *The error is similar across all samples*
- Independence of observations
  - *No hidden groups or relationships*
- Normality
  - *Data follows normal distribution*
- Linear relationship
  - *As opposed to exponential or other*



# Plotting data in R

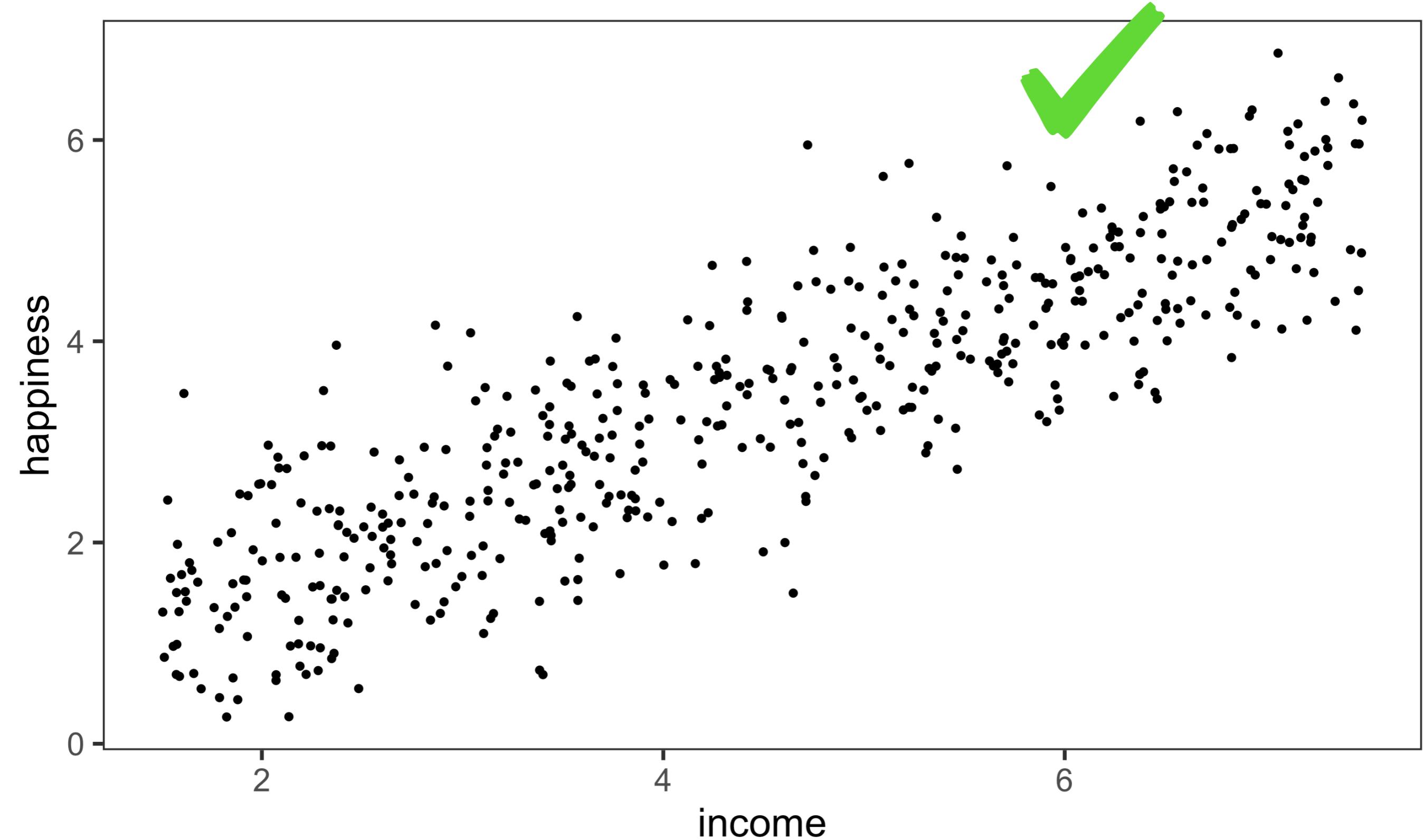
## Base R

```
plot(input_data$income, input_data$happiness)
```

## ggplot2

```
ggplot2::ggplot(income_data) +  
  ggplot2::aes(x = income, y = happiness) +  
  ggplot2::geom_point()
```

# The experiment setup



# The nuts and bolts

$$(y = b + mx + e)$$

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

Predicted value of  
**dependent variable**

(i.e. happiness)

**Intercept**

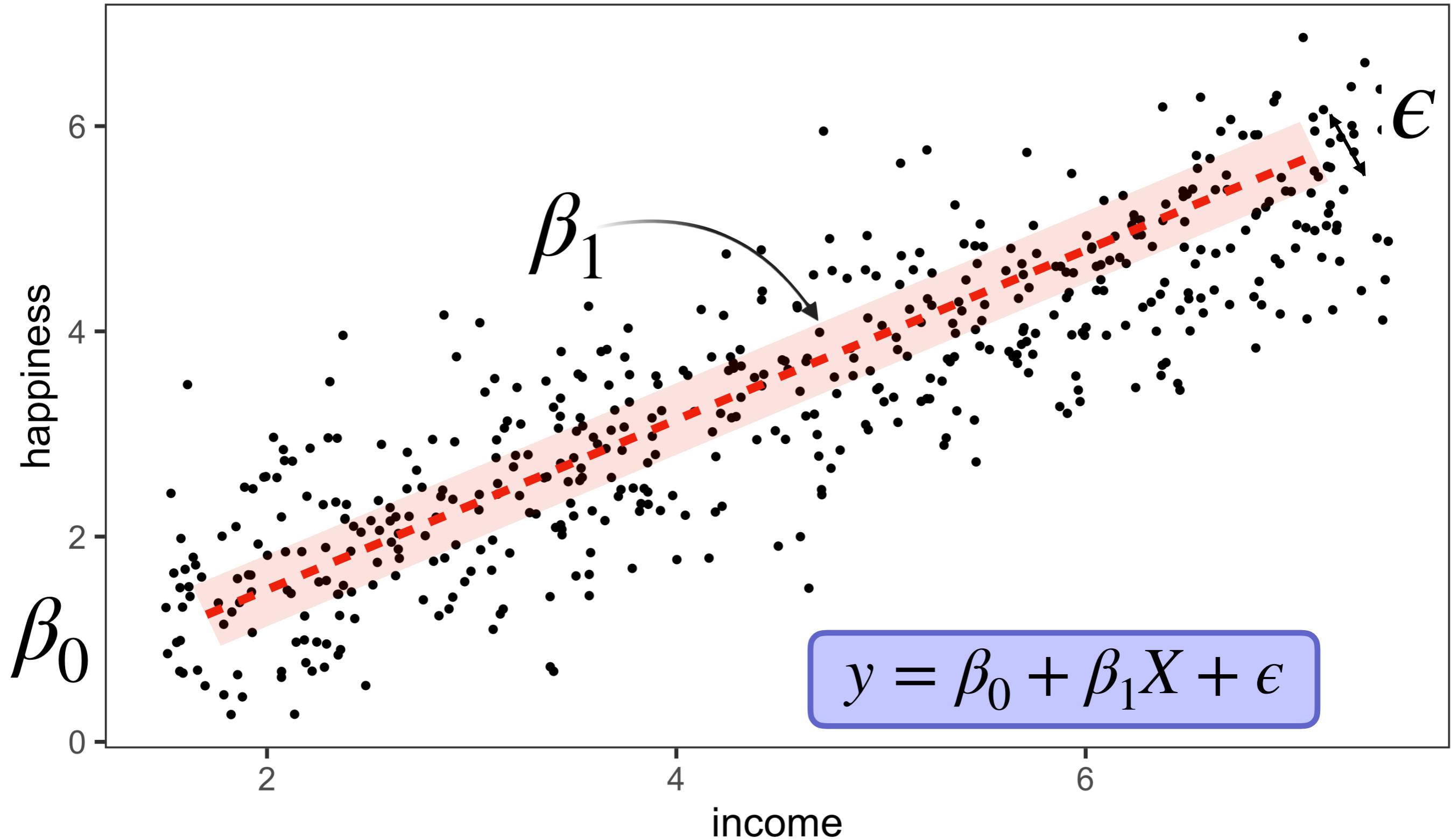
(i.e. predicted happiness  
at \$0 income)

**Regression coefficient**  
(i.e. How much we expect  
y to change with x)

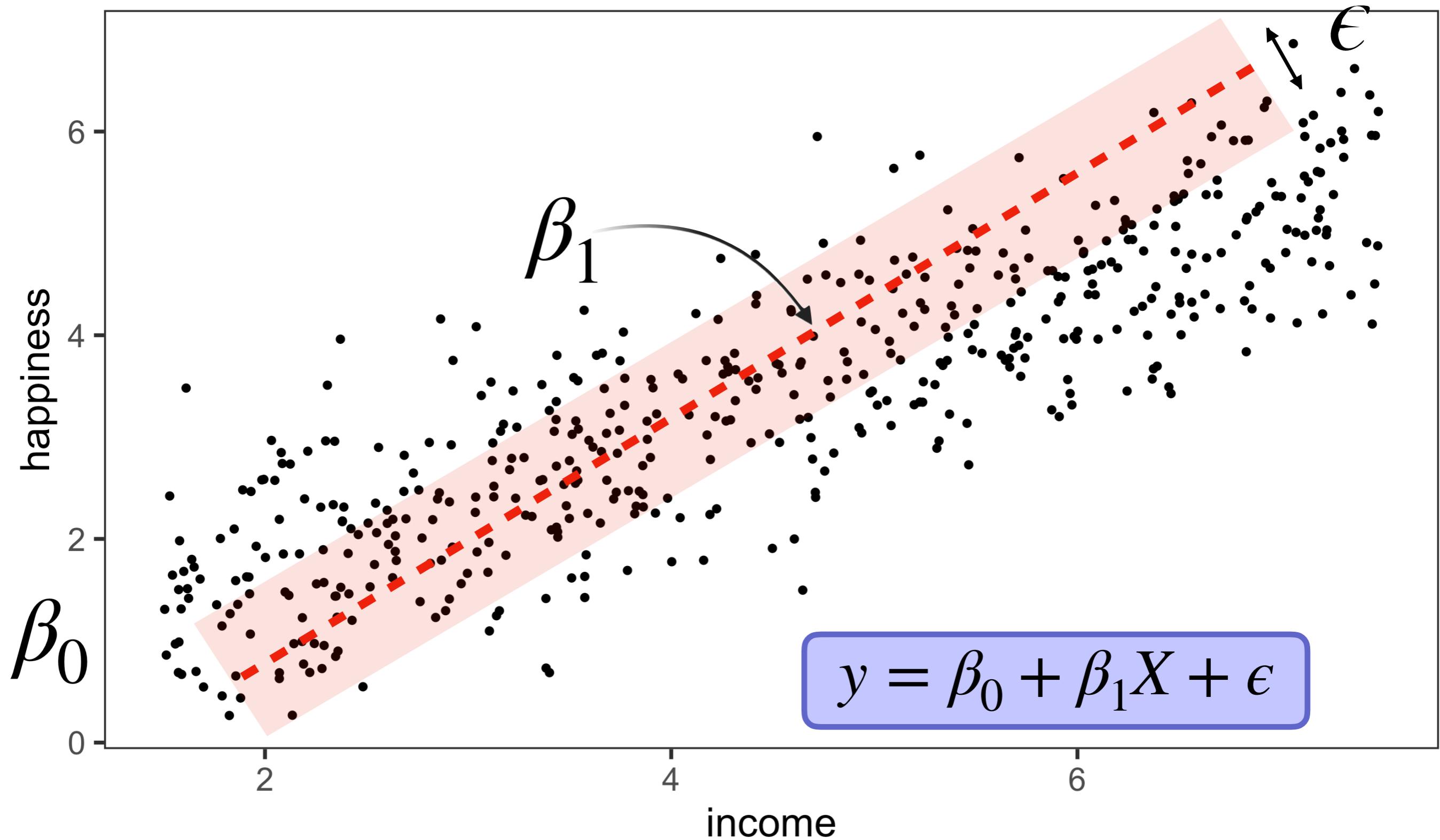
**Error**  
(i.e. variation  
in the estimate)

**Independent variable**  
(i.e. income)

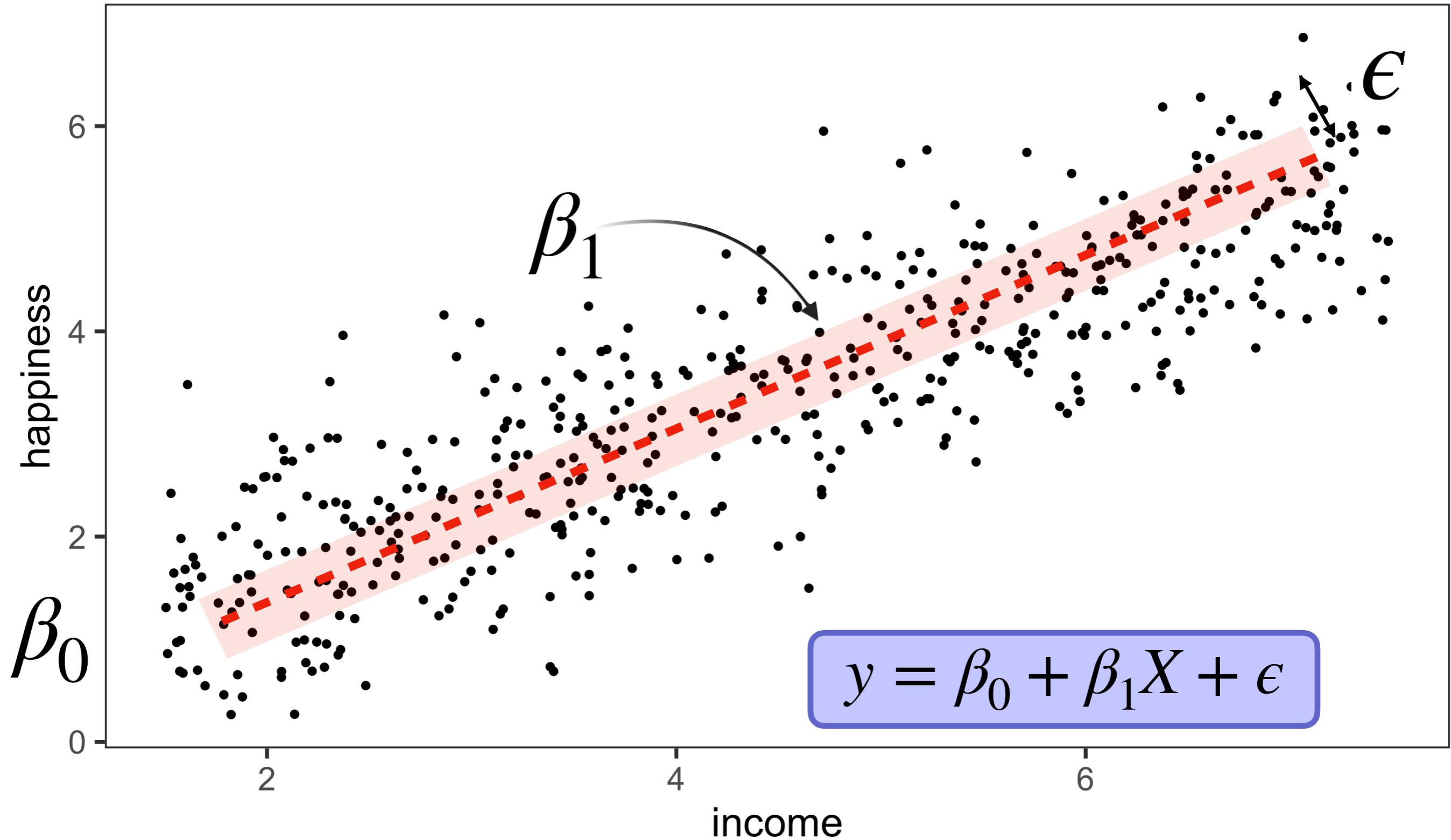
# The regression coefficient



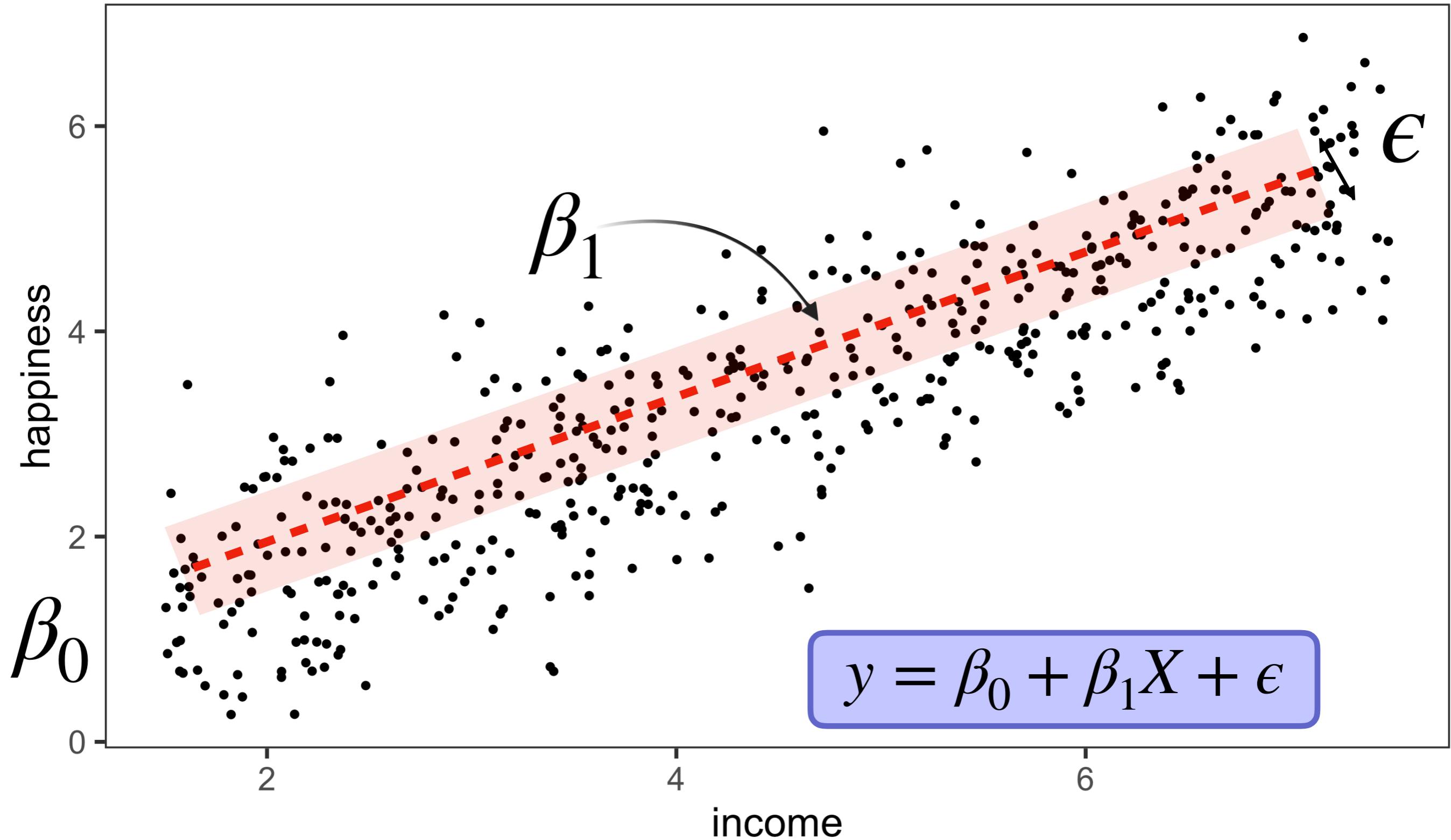
# The regression coefficient



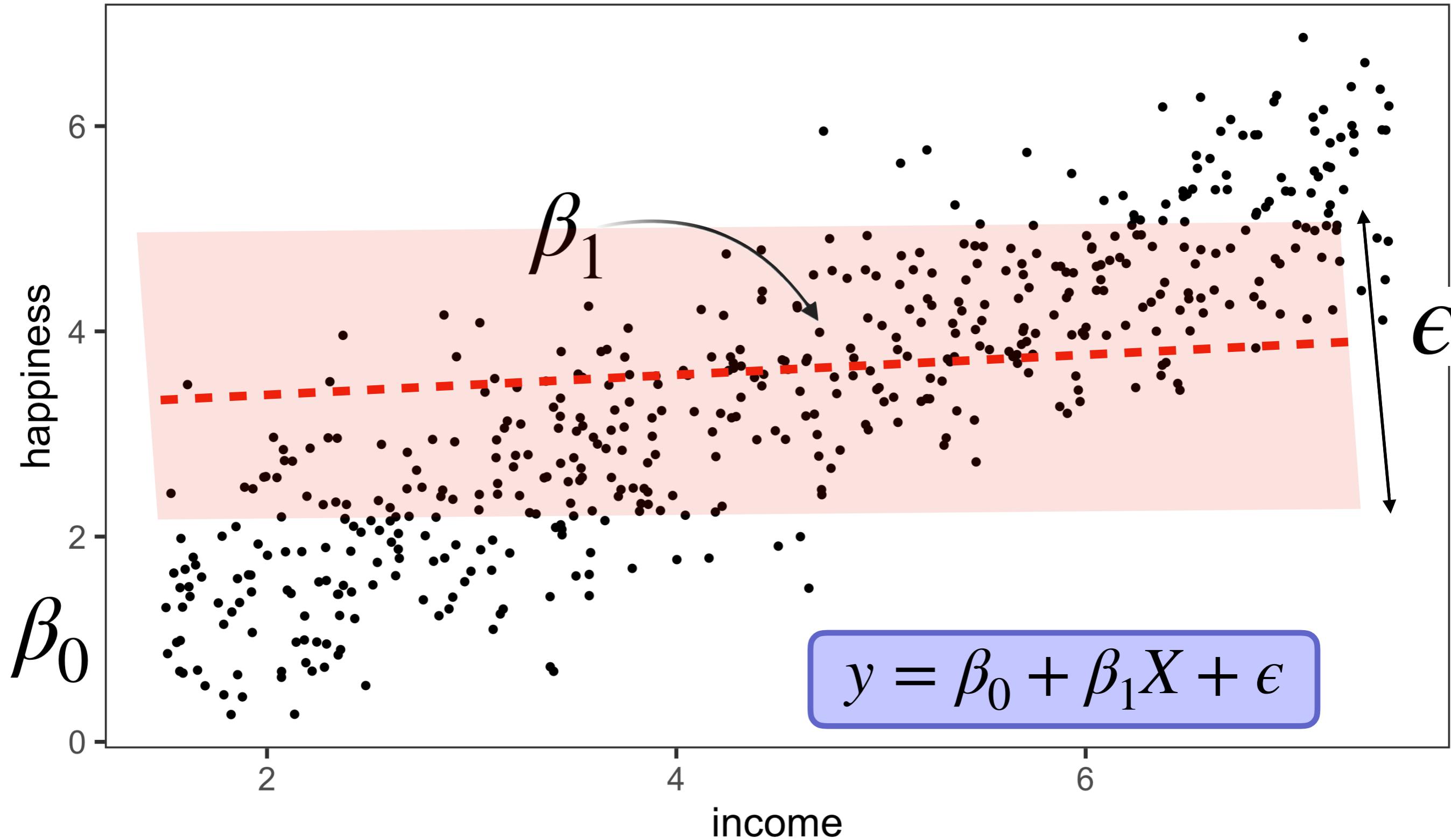
# The regression coefficient



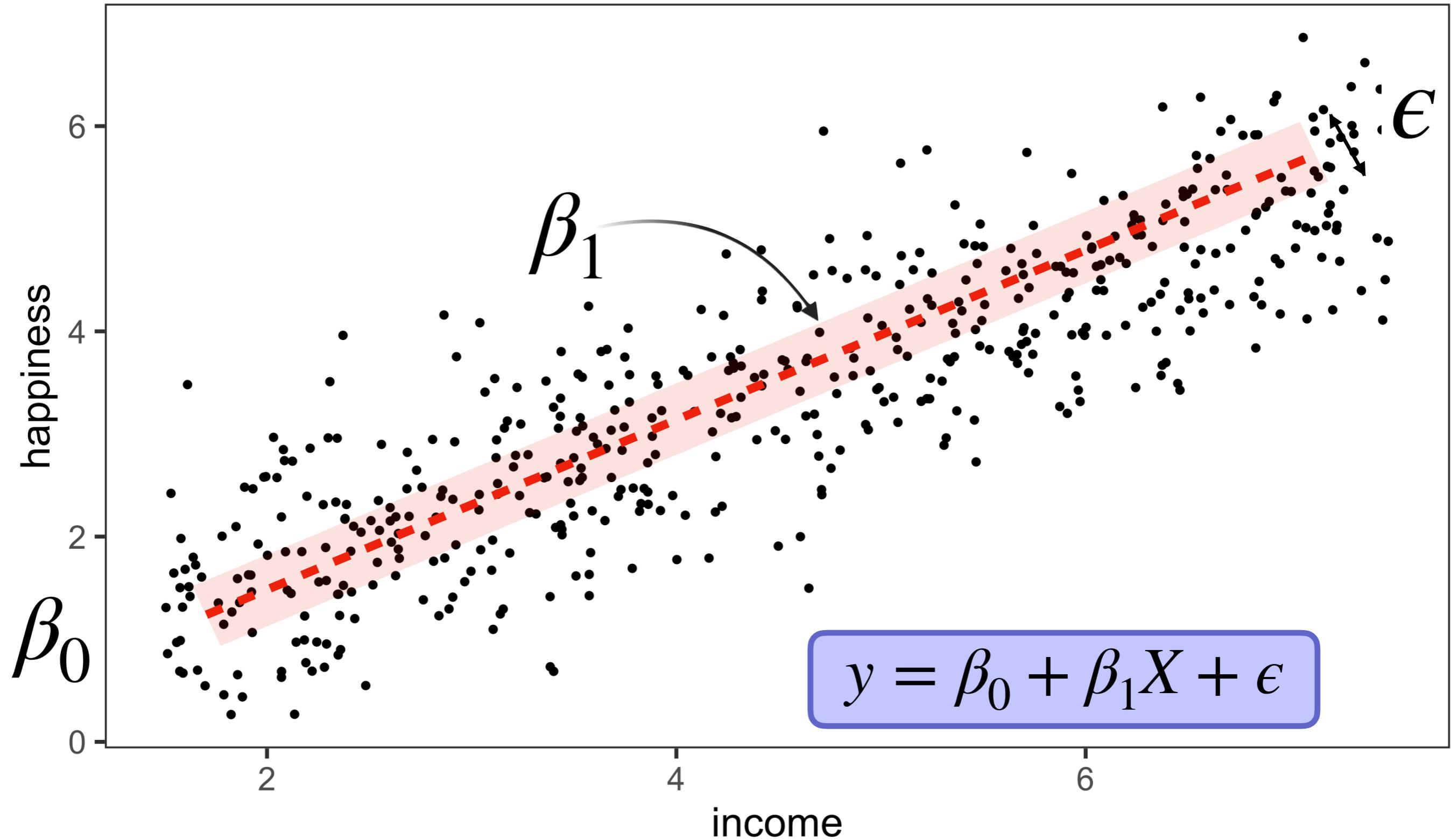
# The regression coefficient



# The regression coefficient



# The regression coefficient



# Linear regression in R

```
lm(happiness ~ income, data = income_data)
```

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

Call:

```
lm(formula = happiness ~ income, data = income_data)
```

Coefficients:

(Intercept)

0.2043

income  
0.7138

*For every 1 unit increase in income,  
there is a 0.7 unit increase in happiness*

# Linear regression in R

```
lm(happiness ~ income, data = income_data)
```

$$happiness = 0.2 + 0.7(income)$$

Call:

```
lm(formula = happiness ~ income, data = income_data)
```

Coefficients:

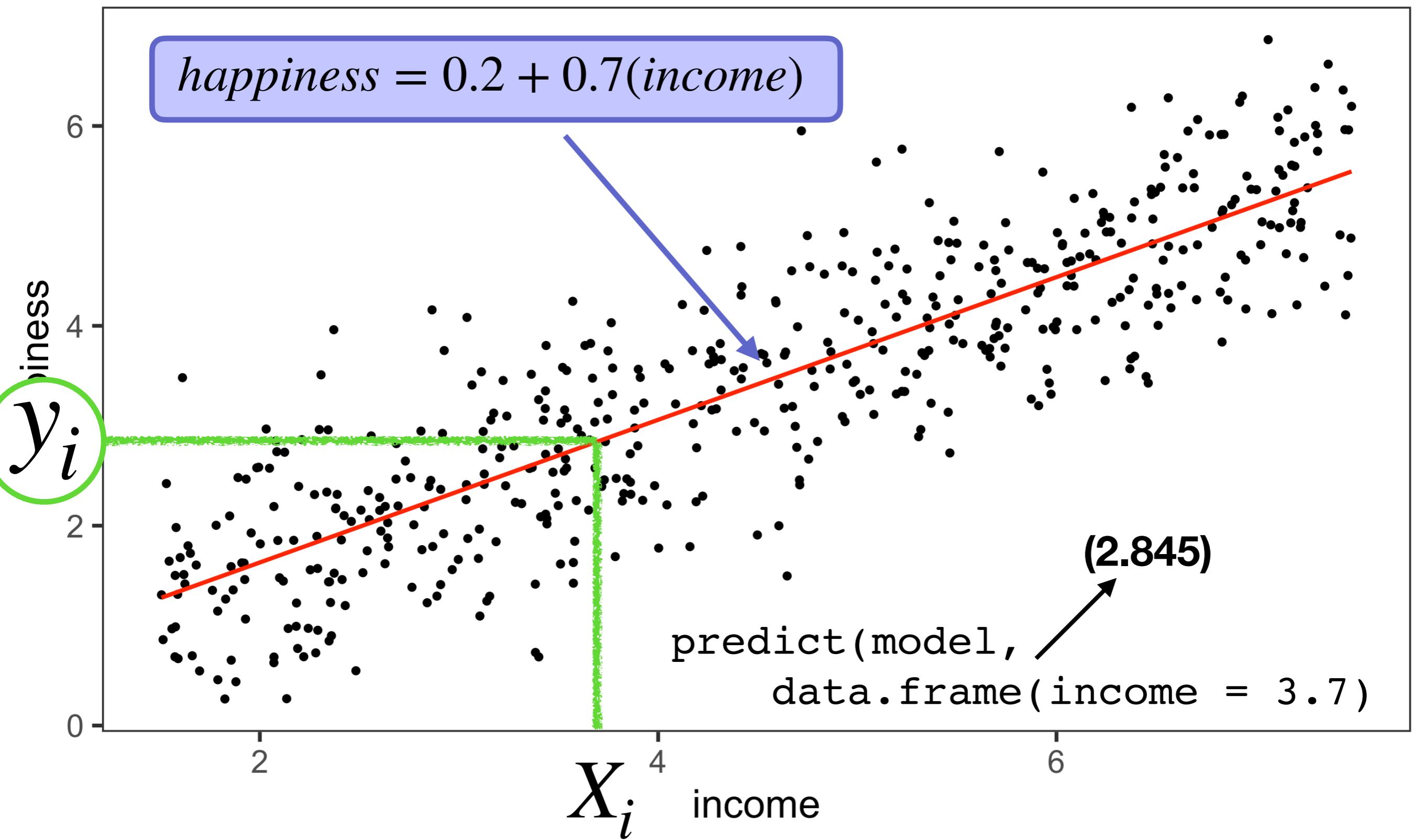
(Intercept)

0.2043

income  
0.7138

*For every 1 unit increase in income,  
there is a 0.7 unit increase in happiness*

# Linear regression in R



# Plotting linear regression in R

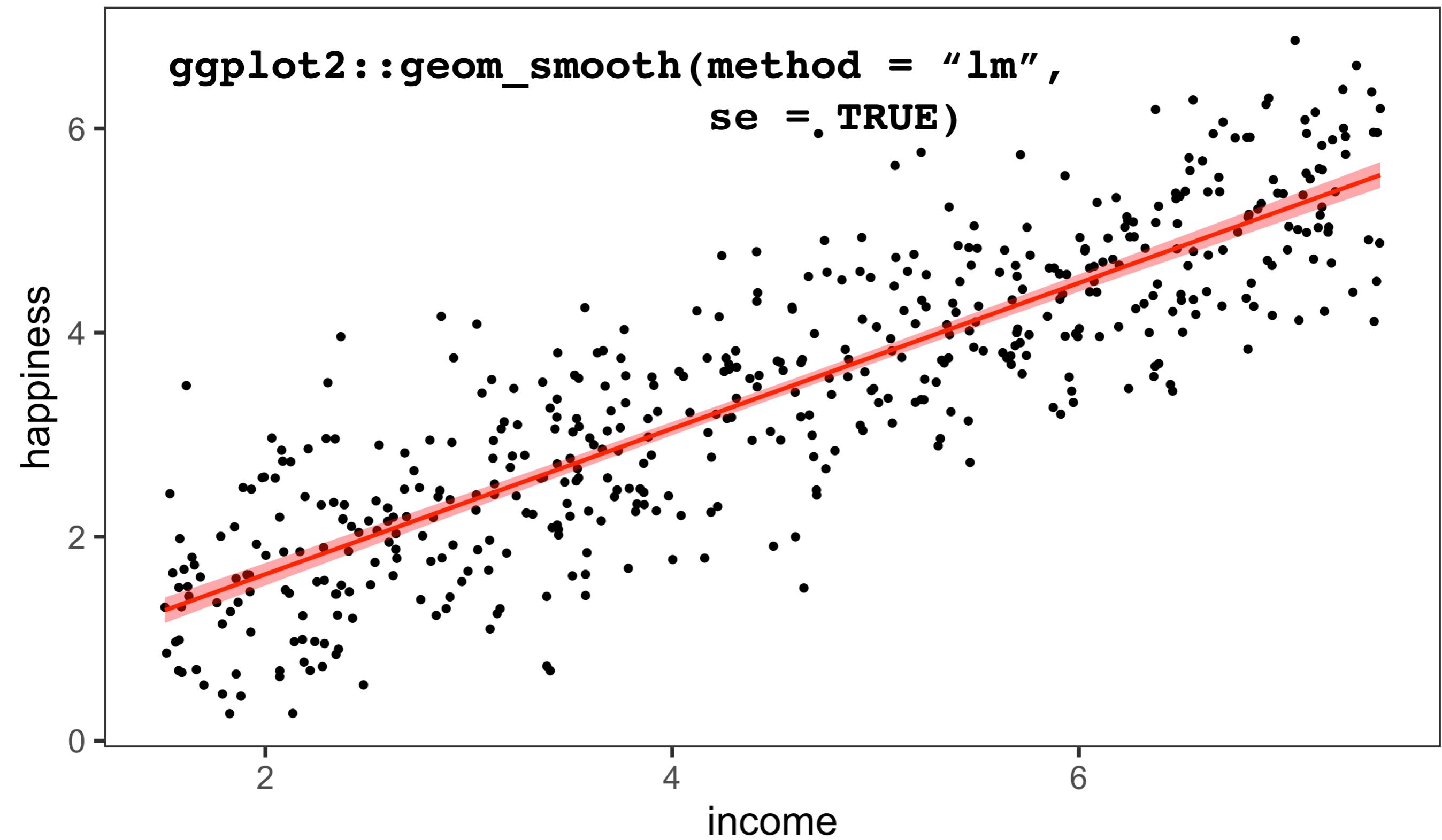
## Base R

```
plot(input_data$income, input_data$happiness)  
  
abline(lm(happiness ~ income, data = input_data))
```

## ggplot2

```
ggplot2::ggplot(income_data) +  
  ggplot2::aes(x = income, y = happiness) +  
  ggplot2::geom_point() +  
  ggplot2::geom_smooth(method = "lm")
```

# Linear regression in R



# Linear regression in R

```
> summary(lm(happiness ~ income, data = income_data))
```

Call:

```
lm(formula = happiness ~ income, data = income_data)
```

Residuals:

| Min      | 1Q       | Median  | 3Q      | Max     |
|----------|----------|---------|---------|---------|
| -2.02479 | -0.48526 | 0.04078 | 0.45898 | 2.37805 |

Coefficients:

**How well the model fits the data**

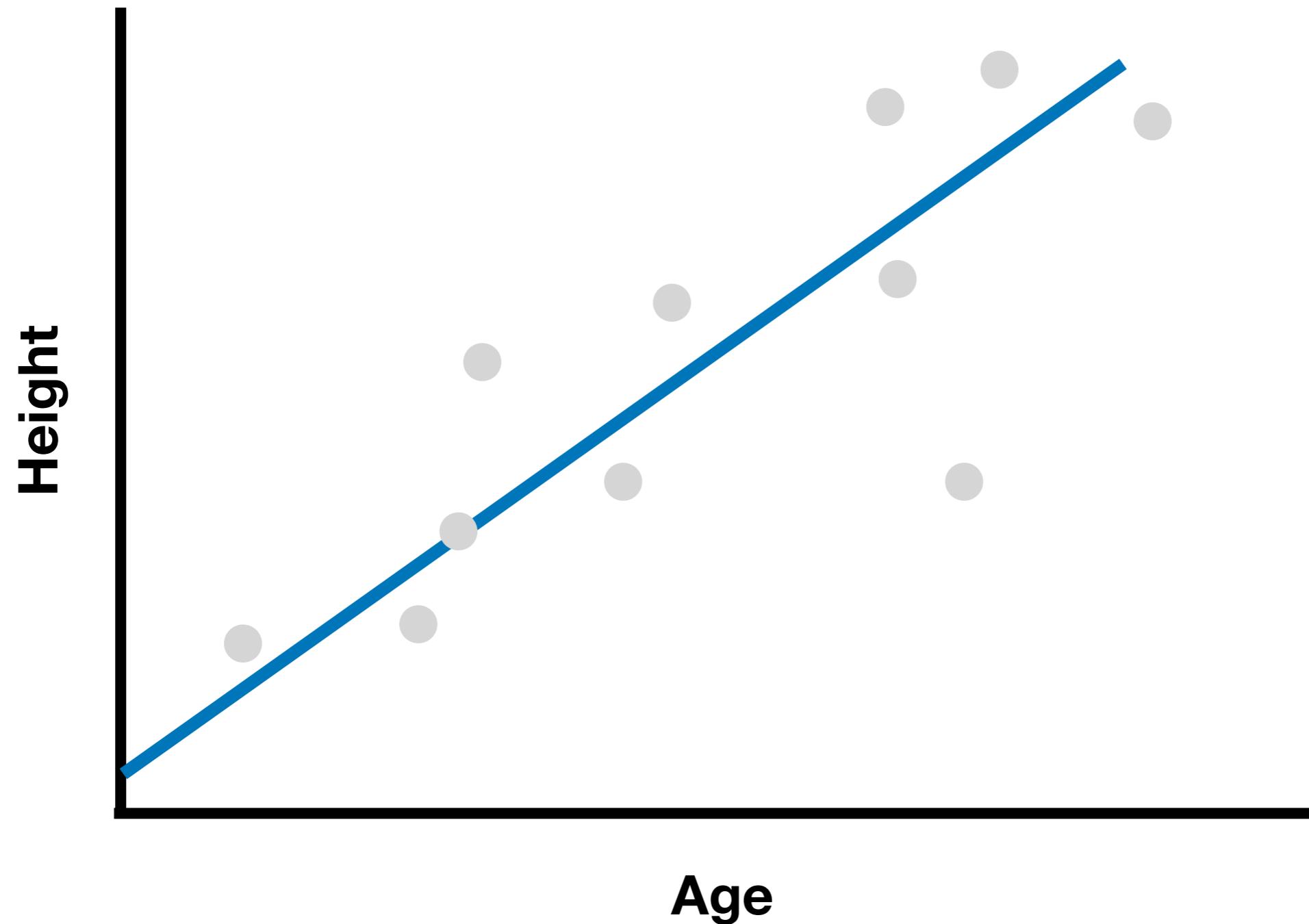
|   | Estimate | Std. Error | t value | Pr(> t )   |
|---|----------|------------|---------|------------|
| (Intercept)   | 0.20427  | 0.08884    | 2.299   | 0.0219 *   |
| income  | 0.71383  | 0.01854    | 38.505  | <2e-16 *** |
| ---   |          |            |         |            |
| Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 |          |            |         |            |

Residual standard error: 0.7181 on 496 degrees of freedom

Multiple R-squared: 0.7493, Adjusted R-squared: 0.7488

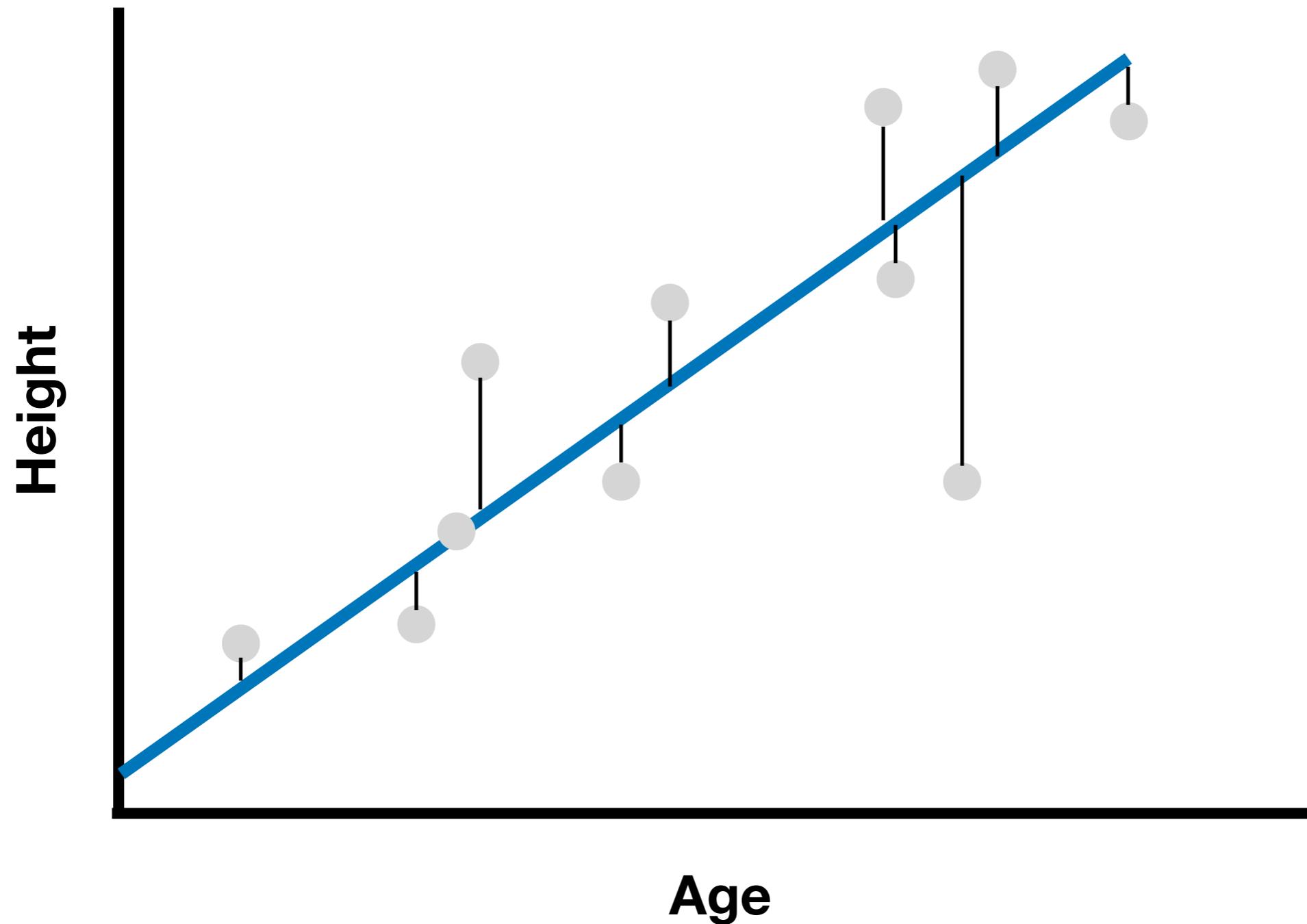
F-statistic: 1483 on 1 and 496 DF, p-value: < 2.2e-16

# Using residual values



# Using residual values

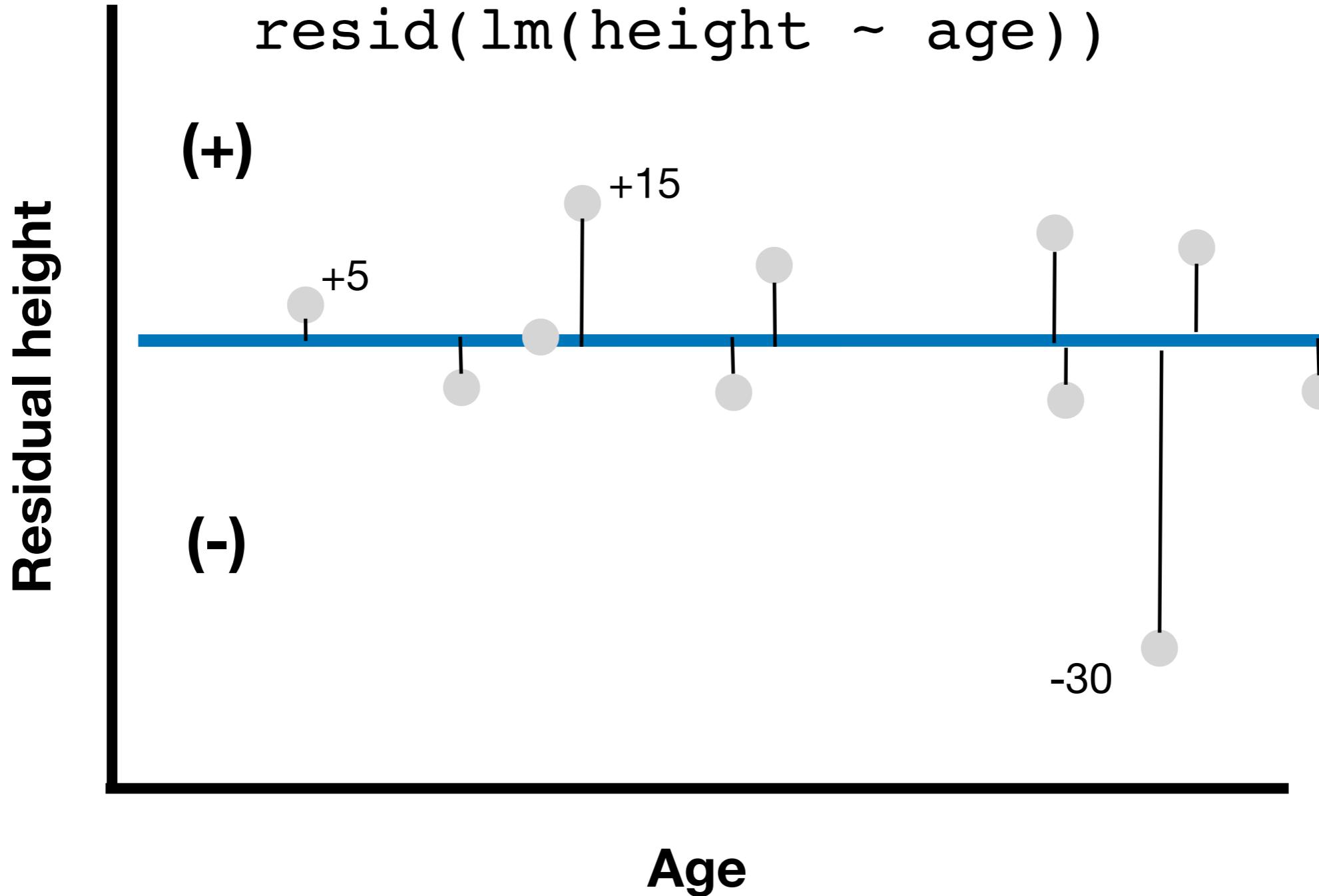
$\text{residual} = \text{observed} - \text{fitted}$



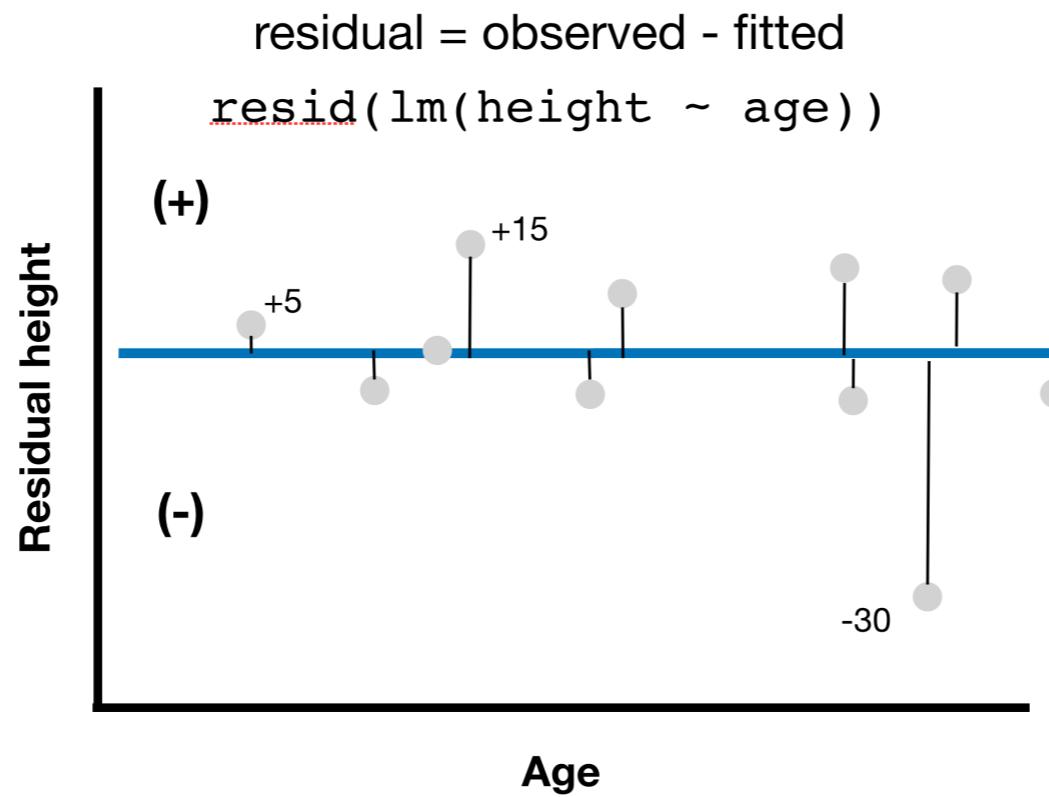
# Using residual values

residual = observed - fitted

`resid(lm(height ~ age))`



# Using residual values



- Mean of zero
- Equally positive as negative
- No remaining shape
- Low residual values

# Linear regression in R

```
> summary(lm(happiness ~ income, data = income_data))
```

Call:

```
lm(formula = happiness ~ income, data = income_data)
```

Residuals:

| Min      | 1Q       | Median  | 3Q      | Max     |
|----------|----------|---------|---------|---------|
| -2.02479 | -0.48526 | 0.04078 | 0.45898 | 2.37805 |

Coefficients:

**How well the model fits the data**

|   | Estimate | Std. Error | t value | Pr(> t )   |
|---|----------|------------|---------|------------|
| (Intercept)   | 0.20427  | 0.08884    | 2.299   | 0.0219 *   |
| income  | 0.71383  | 0.01854    | 38.505  | <2e-16 *** |
| ---   |          |            |         |            |
| Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 |          |            |         |            |

Residual standard error: 0.7181 on 496 degrees of freedom

Multiple R-squared: 0.7493, Adjusted R-squared: 0.7488

F-statistic: 1483 on 1 and 496 DF, p-value: < 2.2e-16

# Using residual values

```
# assign linear model to variable  
model <- lm(happiness ~ income, data = income_data)  
  
# calculate residuals  
residuals <- income_data %>%  
  dplyr::mutate(resid = resid(model))  
  
# plot predicted data  
ggplot2::ggplot(residuals) +  
  ggplot2::aes(x = income, y = resid) +  
  ggplot2::geom_point()  
  
# plot in base R  
plot(residuals$income, residuals$resid)
```

# Using residual values

Great package for  
tidy statistics!

`broom::augment(model)`

residual happiness

2

1

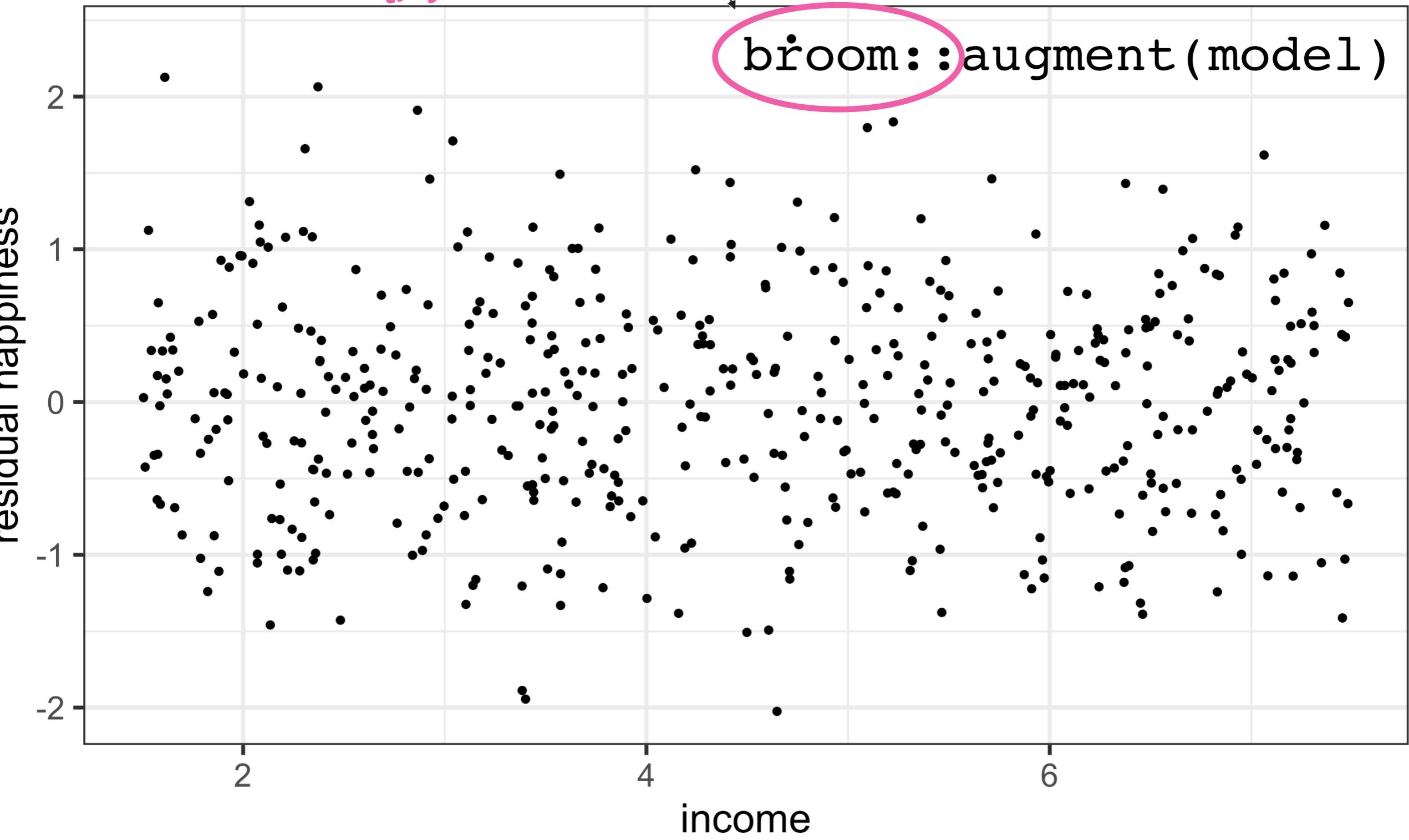
0

-1

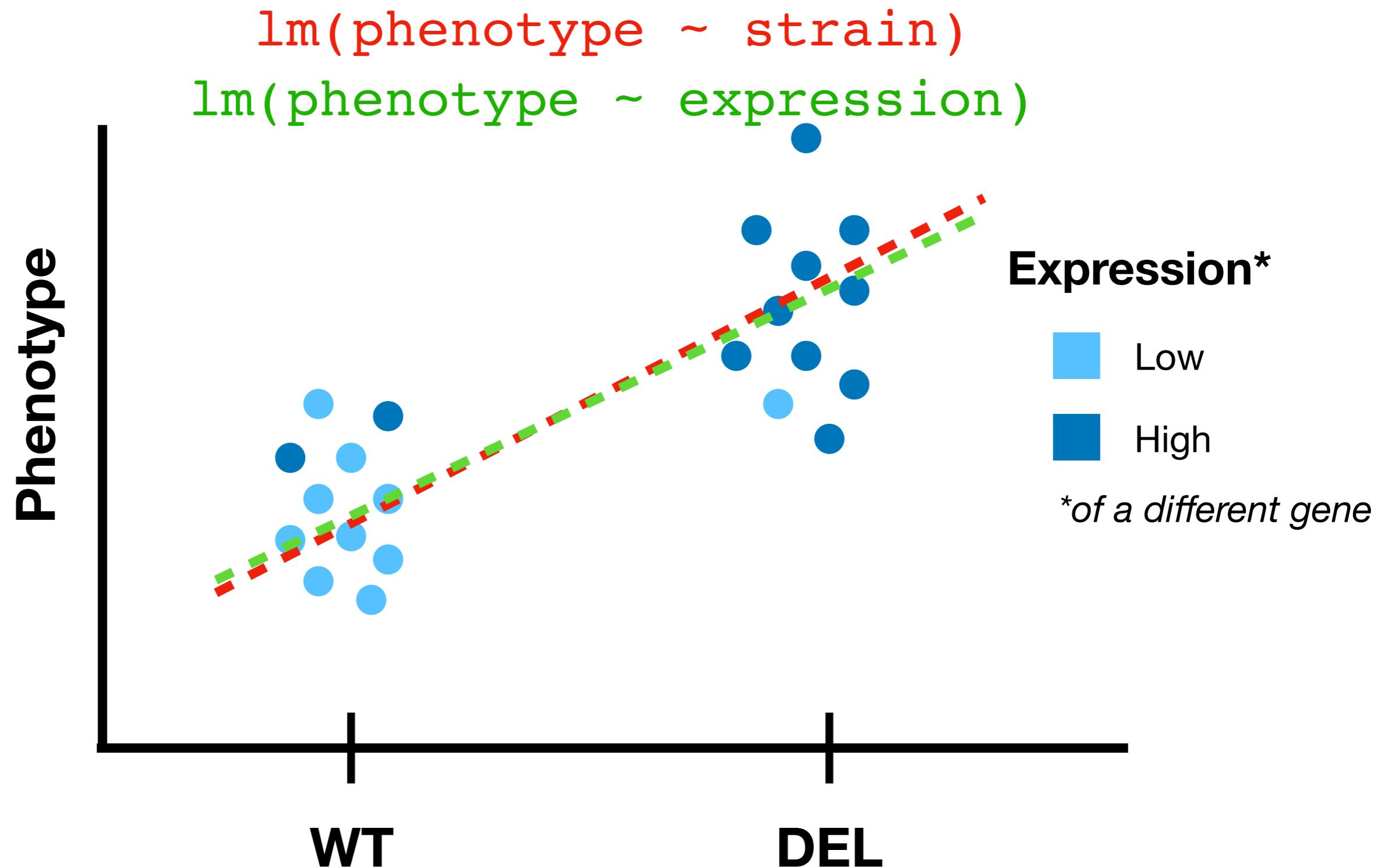
-2

2

income



# Residual values (teaser!)



# Linear regression in R

```
> summary(lm(happiness ~ income, data = income_data))
```

Call:

```
lm(formula = happiness ~ income, data = income_data)
```

Residuals:

| Min      | 1Q       | Median  | 3Q      | Max     |
|----------|----------|---------|---------|---------|
| -2.02479 | -0.48526 | 0.04078 | 0.45898 | 2.37805 |

Slope and intercept

Coefficients:

|             | Estimate | Std. Error | t value | Pr(> t )   |
|-------------|----------|------------|---------|------------|
| (Intercept) | 0.20427  | 0.08884    | 2.299   | 0.0219 *   |
| income      | 0.71383  | 0.01854    | 38.505  | <2e-16 *** |

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7181 on 496 degrees of freedom

Multiple R-squared: 0.7493, Adjusted R-squared: 0.7488

F-statistic: 1483 on 1 and 496 DF, p-value: < 2.2e-16

# Linear regression in R

```
> summary(lm(happiness ~ income, data = income_data))
```

Call:

```
lm(formula = happiness ~ income, data = income_data)
```

Residuals:

| Min      | 1Q       | Median  | 3Q      | Max     | How much variation |
|----------|----------|---------|---------|---------|--------------------|
| -2.02479 | -0.48526 | 0.04078 | 0.45898 | 2.37805 |                    |

Coefficients:

|             | Estimate | Std. Error | t value | Pr(> t )   |
|-------------|----------|------------|---------|------------|
| (Intercept) | 0.20427  | 0.08884    | 2.299   | 0.0219 *   |
| income      | 0.71383  | 0.01854    | 38.505  | <2e-16 *** |

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7181 on 496 degrees of freedom

Multiple R-squared: 0.7493, Adjusted R-squared: 0.7488

F-statistic: 1483 on 1 and 496 DF, p-value: < 2.2e-16

# Linear regression in R

```
> summary(lm(happiness ~ income, data = income_data))
```

Call:

```
lm(formula = happiness ~ income, data = income_data)
```

Residuals:

| Min      | 1Q       | Median  | 3Q      | Max     |
|----------|----------|---------|---------|---------|
| -2.02479 | -0.48526 | 0.04078 | 0.45898 | 2.37805 |

**Test statistic**

Coefficients:

|             | Estimate | Std. Error | t value | Pr(> t )   |
|-------------|----------|------------|---------|------------|
| (Intercept) | 0.20427  | 0.08884    | 2.299   | 0.0219 *   |
| income      | 0.71383  | 0.01854    | 38.505  | <2e-16 *** |

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7181 on 496 degrees of freedom

Multiple R-squared: 0.7493, Adjusted R-squared: 0.7488

F-statistic: 1483 on 1 and 496 DF, p-value: < 2.2e-16

# Linear regression in R

```
> summary(lm(happiness ~ income, data = income_data))
```

Call:

```
lm(formula = happiness ~ income, data = income_data)
```

Residuals:

| Min      | 1Q       | Median  | 3Q      | Max     | p-value |
|----------|----------|---------|---------|---------|---------|
| -2.02479 | -0.48526 | 0.04078 | 0.45898 | 2.37805 |         |

Coefficients:

|             | Estimate | Std. Error | t value | Pr(> t )   |
|-------------|----------|------------|---------|------------|
| (Intercept) | 0.20427  | 0.08884    | 2.299   | 0.0219 *   |
| income      | 0.71383  | 0.01854    | 38.505  | <2e-16 *** |

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7181 on 496 degrees of freedom

Multiple R-squared: 0.7493, Adjusted R-squared: 0.7488

F-statistic: 1483 on 1 and 496 DF, p-value: < 2.2e-16

# Linear regression in R

```
> summary(lm(happiness ~ income, data = income_data))
```

Call:

```
lm(formula = happiness ~ income, data = income_data)
```

Residuals:

| Min      | 1Q       | Median  | 3Q      | Max     |
|----------|----------|---------|---------|---------|
| -2.02479 | -0.48526 | 0.04078 | 0.45898 | 2.37805 |

Is intercept significantly different from zero?

p-value

Coefficients:

|             | Estimate | Std. Error | t value | Pr(> t )   |
|-------------|----------|------------|---------|------------|
| (Intercept) | 0.20427  | 0.08884    | 2.299   | 0.0219 *   |
| income      | 0.71383  | 0.01854    | 38.505  | <2e-16 *** |

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7181 on 496 degrees of freedom

Multiple R-squared: 0.7493, Adjusted R-squared: 0.7488

F-statistic: 1483 on 1 and 496 DF, p-value: < 2.2e-16

# Linear regression in R

```
> summary(lm(happiness ~ income, data = income_data))
```

Call:

```
lm(formula = happiness ~ income, data = income_data)
```

Residuals:

| Min      | 1Q       | Median  | 3Q      | Max     |
|----------|----------|---------|---------|---------|
| -2.02479 | -0.48526 | 0.04078 | 0.45898 | 2.37805 |

Coefficients:

|             | Estimate | Std. Error | t value | Pr(> t )   |
|-------------|----------|------------|---------|------------|
| (Intercept) | 0.20427  | 0.08884    | 2.299   | 0.0219 *   |
| income      | 0.71383  | 0.01854    | 38.505  | <2e-16 *** |
| ---         |          |            |         |            |

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

**More fit values**

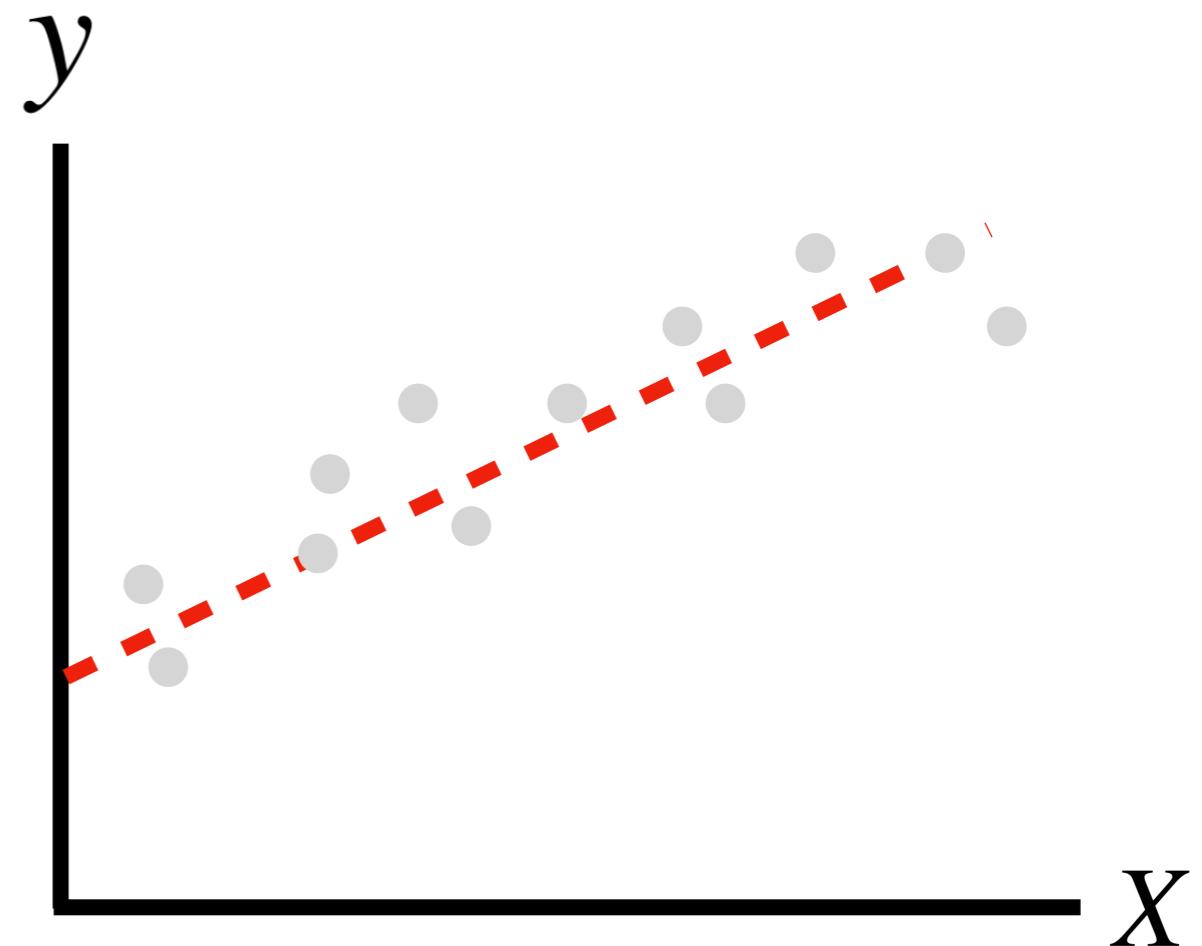
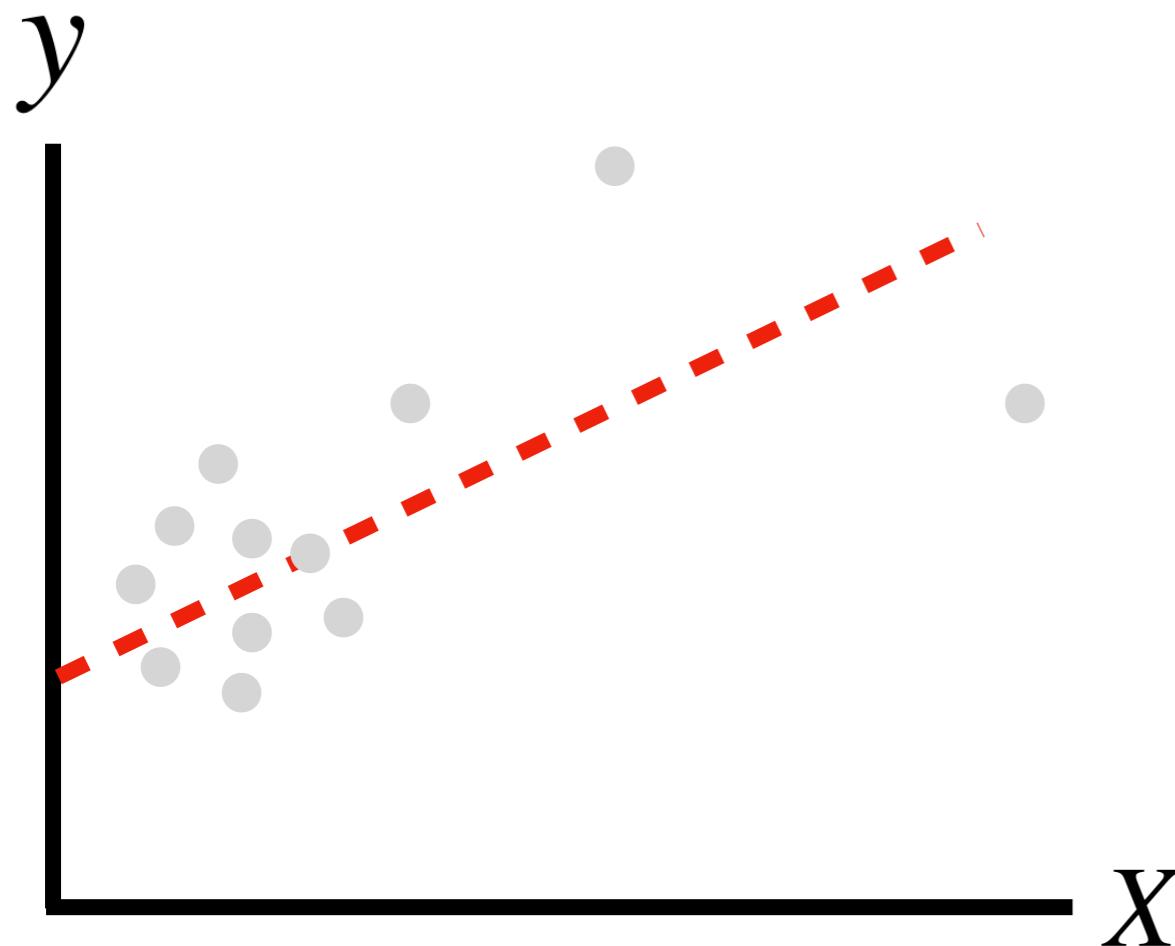
Residual standard error: 0.7181 on 496 degrees of freedom

Multiple R-squared: 0.7493, Adjusted R-squared: 0.7488

F-statistic: 1483 on 1 and 496 DF, p-value: < 2.2e-16

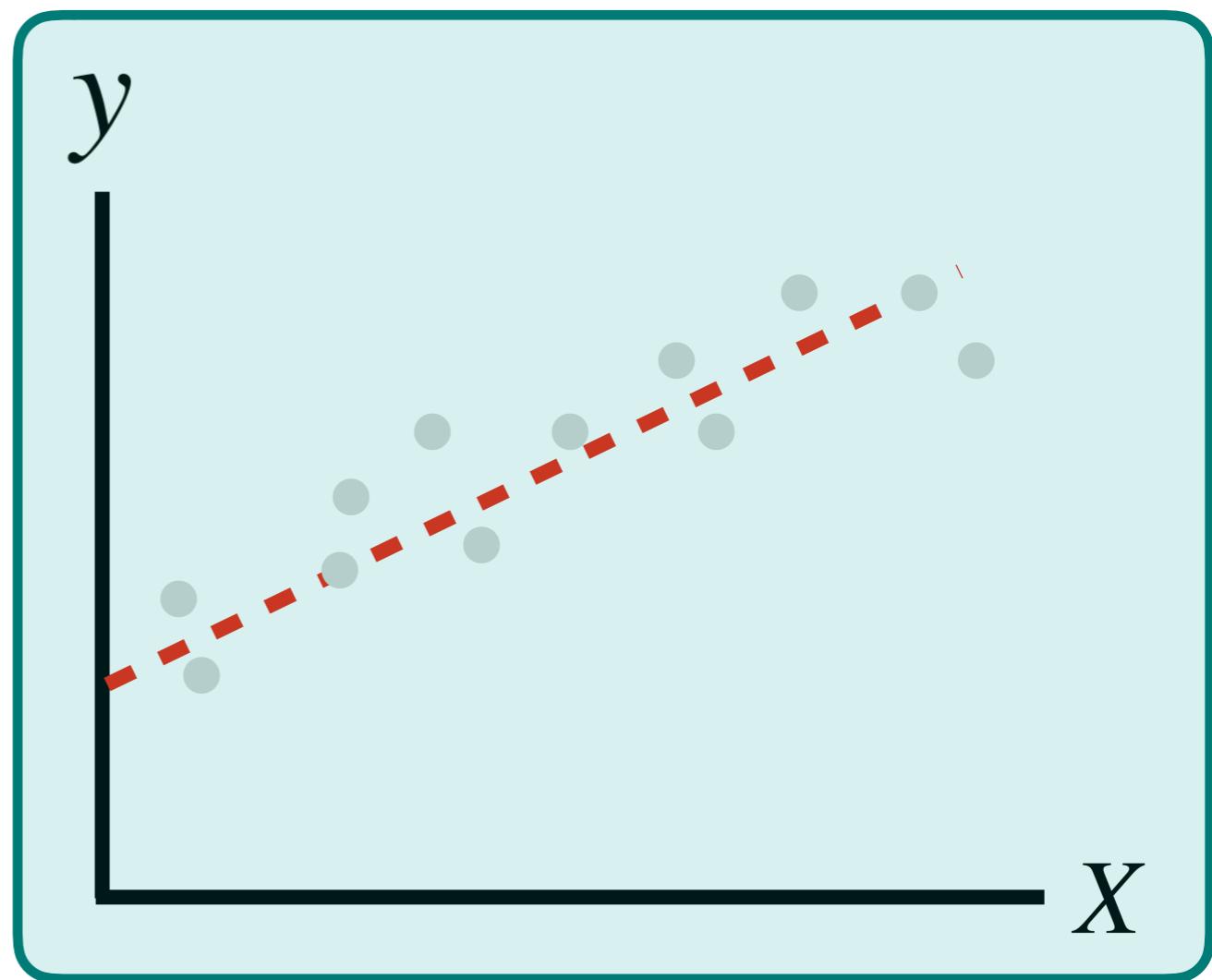
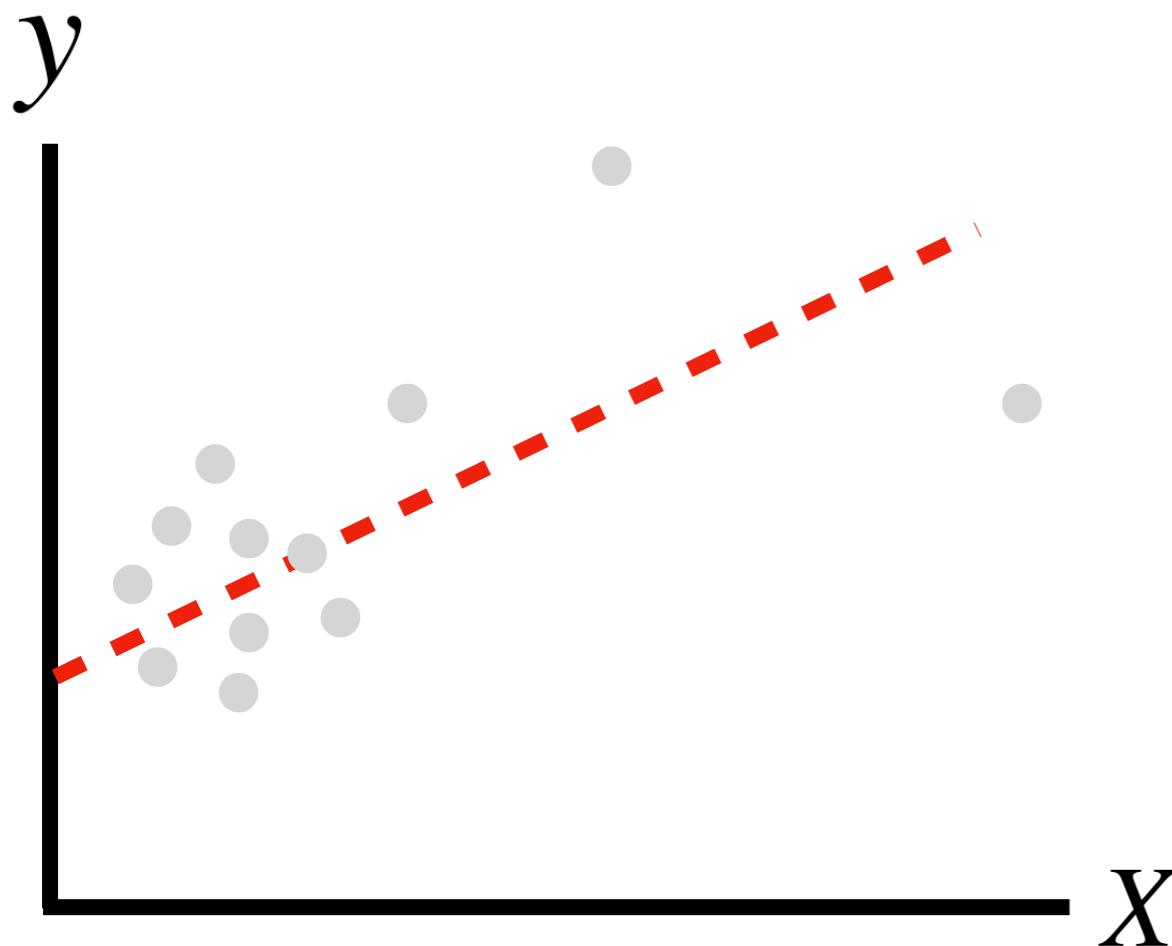
# Linear regression in R

Which model has the higher  $R^2$  value?



# Linear regression in R

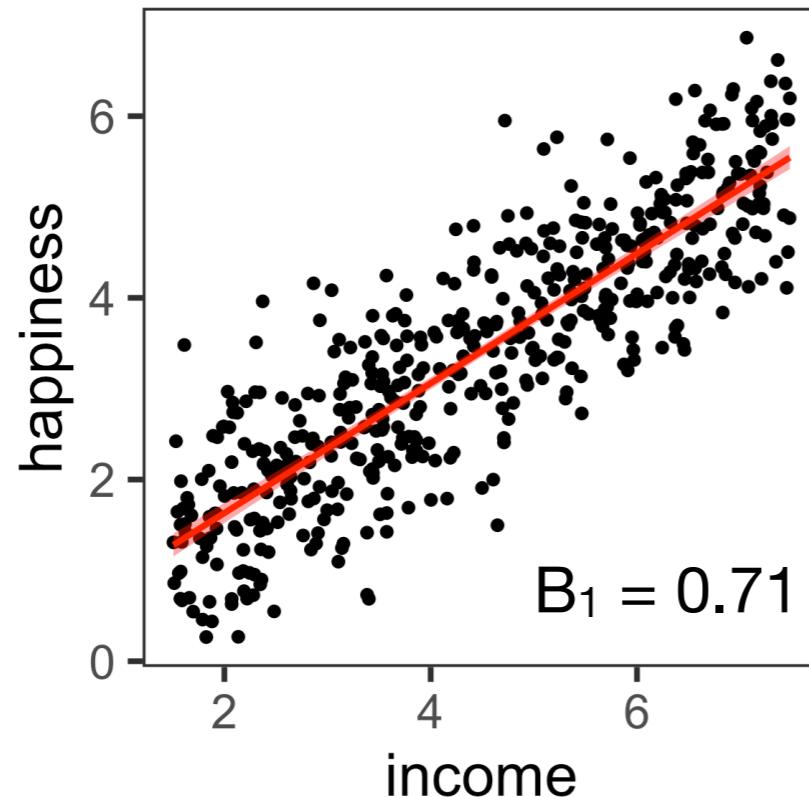
Which model has the higher  $R^2$  value?



# Regression vs. correlation

**Regression:** Describes how one variable (x) affects another variable (y)

**Correlation:** Quantifies the direction and strength of the relationship between two variables (x and y)



`cor(happiness, income)`

0.865

$(0.865^2 = R^2 = 0.75)$

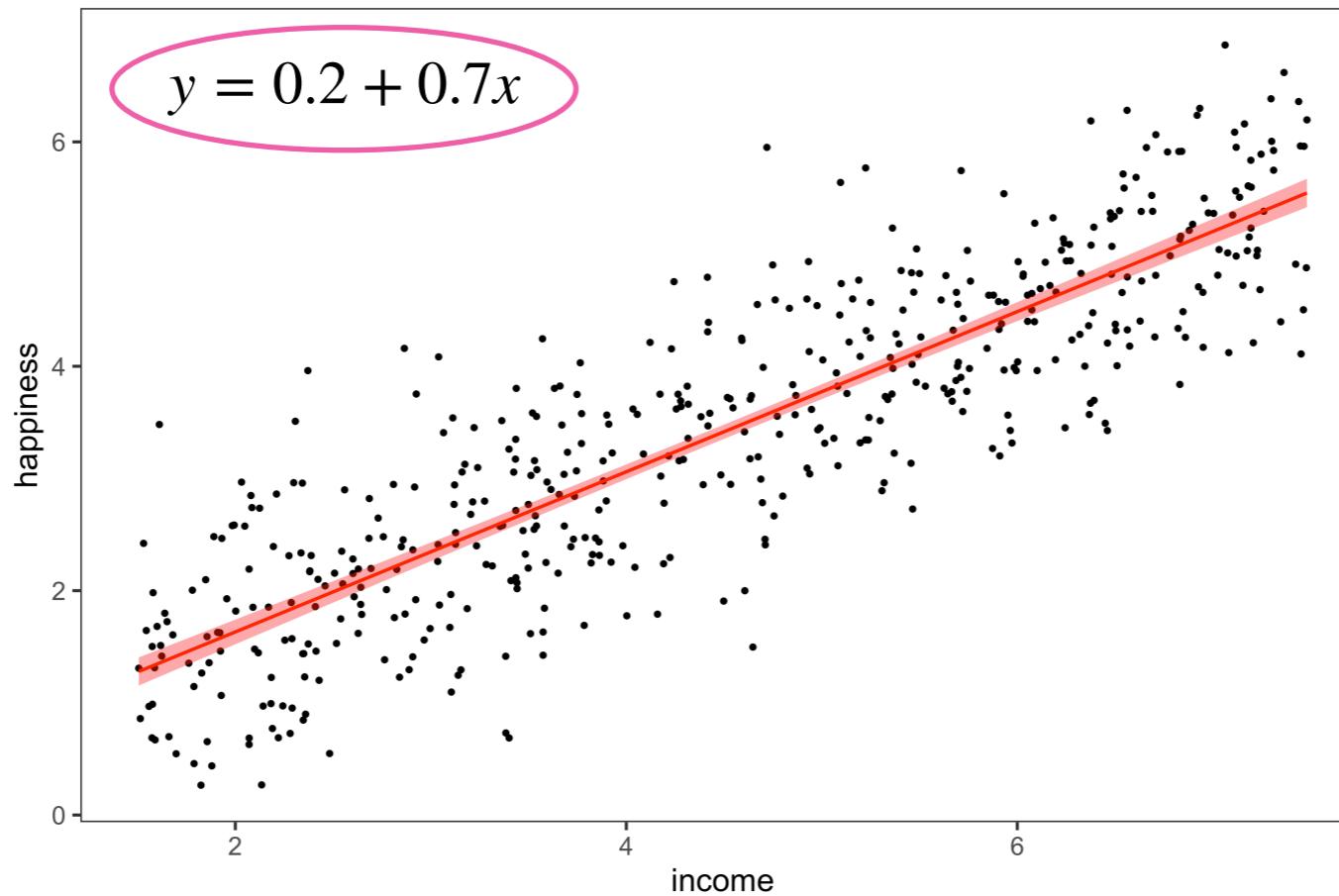
# Regression vs. correlation

**Regression:** Describes how one variable (x) affects another variable (y)

**Correlation:** Quantifies the direction and strength of the relationship between two variables (x and y)

|   | Correlation  | Regression                      |
|---|--|---------------------------------|
| When to use                                 | When summarizing direct relationship between two variables | To predict or explain responses |
| Able to quantify direction of relationship? | Yes  | Yes                             |
| Able to quantify strength of relationship?  | Yes  | Yes                             |
| Able to show cause and effect?              | No   | Yes                             |
| Able to predict and optimize?               | No   | Yes                             |
| X and Y are interchangeable?                | Yes  | No                              |

# Reporting results



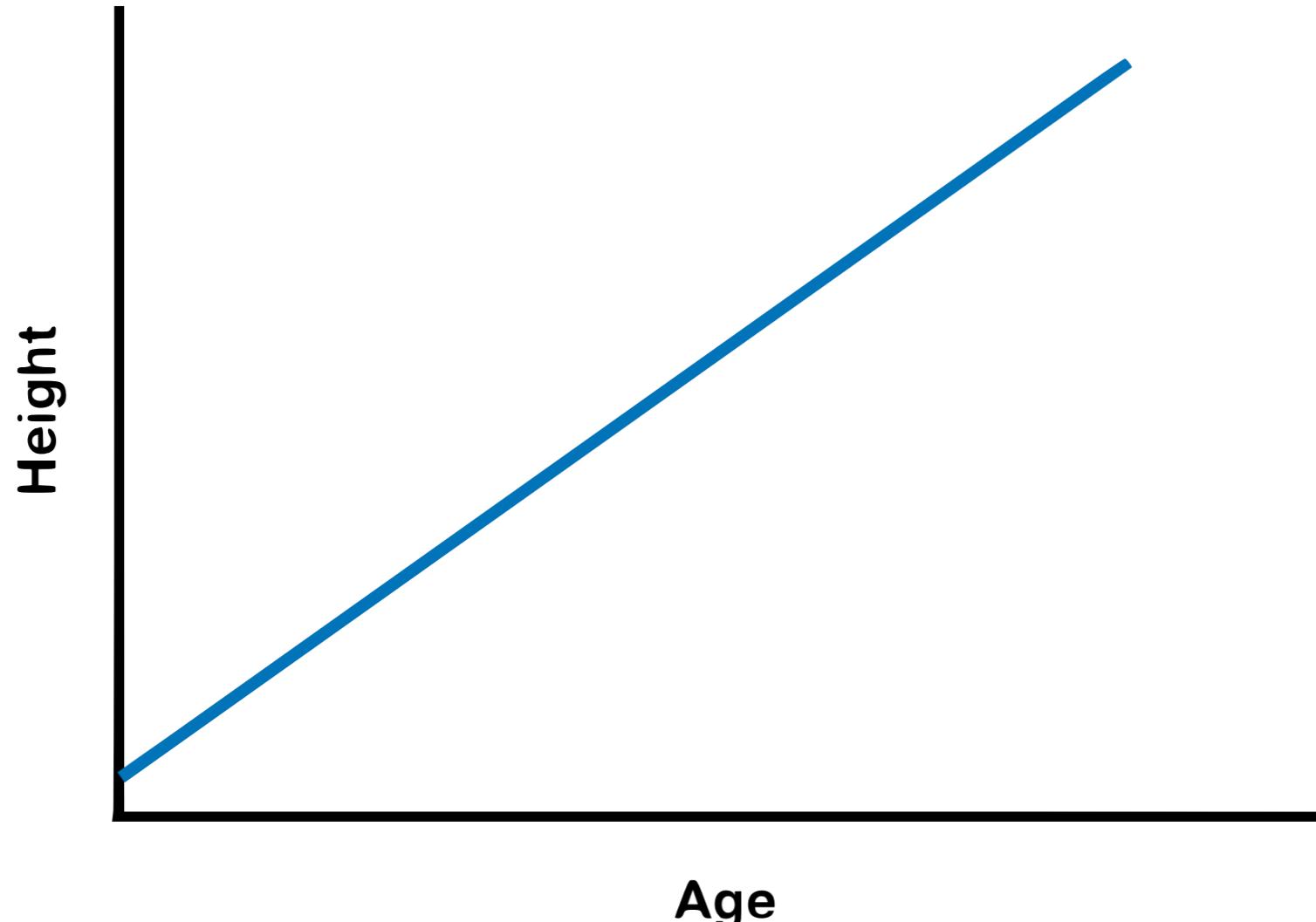
$B_1$

$R^2$

A simple linear regression was performed to test if income significantly predicted happiness. (Describe experiment here). The analysis indicated that the model (equation here) explained 75% of the variation in happiness and that income did significantly predict happiness with a 0.71 unit increase in happiness for every \$10,000 increase in income ( $B_1 = 0.71 \pm 0.018$ ,  $p\text{-value} < 2e-16$ ).

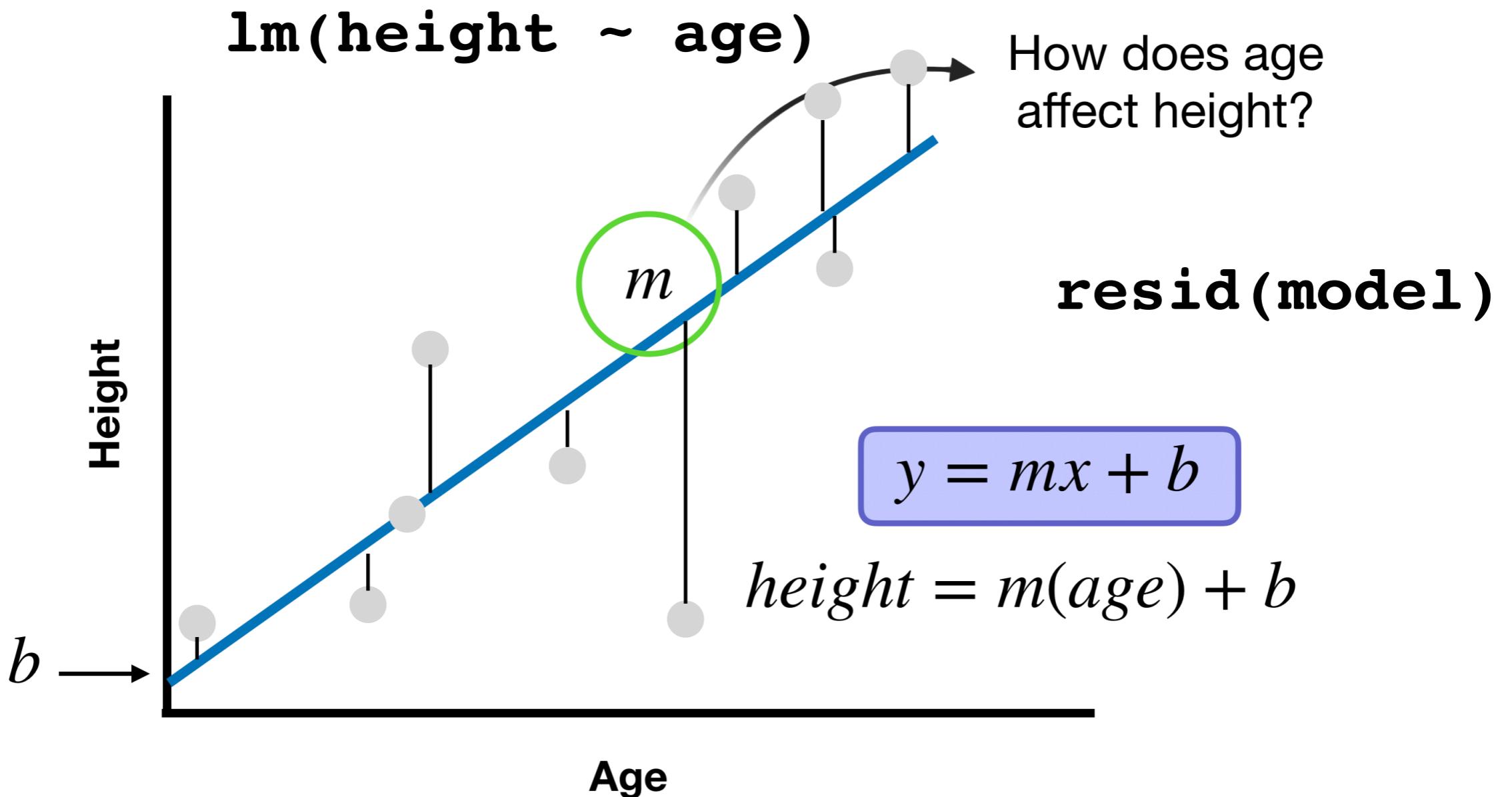
# Summary: linear regression

`lm(height ~ age)`



[https://github.com/katieselevans/IGP\\_biostatistics](https://github.com/katieselevans/IGP_biostatistics)

# Summary: linear regression



[https://github.com/katieselevans/IGP\\_biostatistics](https://github.com/katieselevans/IGP_biostatistics)

# Course materials: slides and code

The screenshot shows a GitHub repository page for 'katiesevans / IGP\_biostatistics'. The repository has 1 unwatched star, 0 forks, and 0 issues. The 'Code' tab is selected, showing a main branch with 6 commits. The latest commit is from 'katiesevans' updating README.md. Other commits include 'first draft slides' and another update to README.md. The repository description states: 'Quantitative Biology: Statistics and Data Analysis for Life Scientists'. It explains that the repository contains course materials for IGP-484 (2021), including lecture slides and R code for tidyverse analyses. It provides instructions for installing the tidyverse package or specific packages like dplyr, tidyr, and ggplot2. A 'Data analysis with the Tidyverse' section links to course materials for help with basic data wrangling. Contact information for Katie Evans is provided at kathryn.evans@northwestern.edu.

katiesevans / IGP\_biostatistics

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

**katiesevans** Update README.md 25c27ef 5 hours ago 6 commits

20210622\_demo first draft slides 5 hours ago

README.md Update README.md 5 hours ago

**README.md**

## Quantitative Biology: Statistics and Data Analysis for Life Scientists

This repository contains all the course materials for IGP-484 (2021). You can find both the lecture slides and the code to generate all plots and statistical analyses in the designated date folder. All analyses are performed in R and require previous installation of `tidyverse`. If you don't have `tidyverse`, you can install the package with `install.packages("tidyverse")` or you can install the specific packages required for this class: `install.packages(c("dplyr", "tidyverse", "ggplot2"))`.

### Data analysis with the Tidyverse

Looking for help with basic data wrangling in R? New to the "Tidyverse"? Check out the [course materials](#) from my workshop for NUIT for step-by-step help and lots of examples and practice questions.

### Questions?

For all questions, contact Katie at [kathryn.evans@northwestern.edu](mailto:kathryn.evans@northwestern.edu)

About

No description, website, or topics provided.

Readme

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

Languages

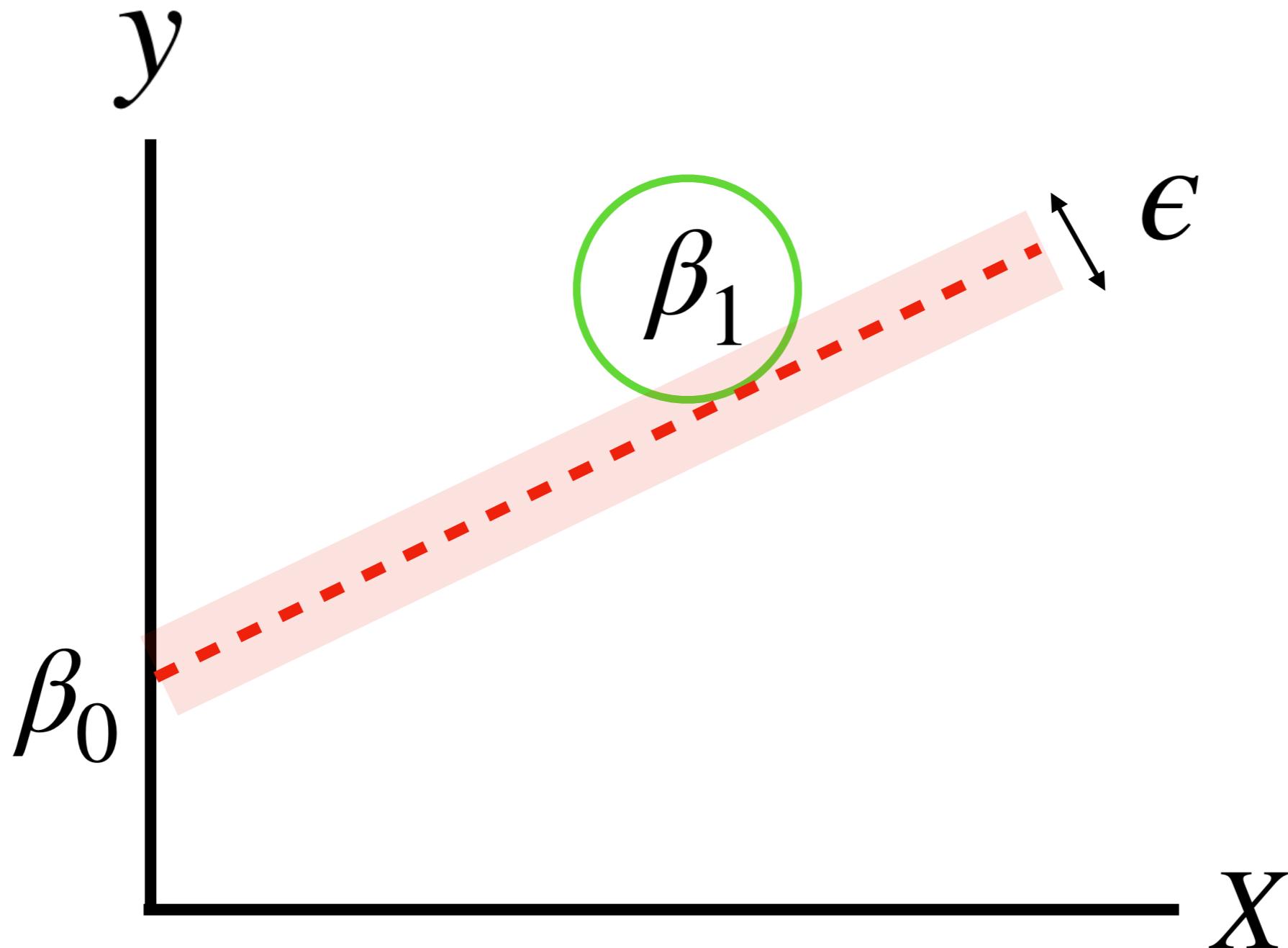
R 100.0%



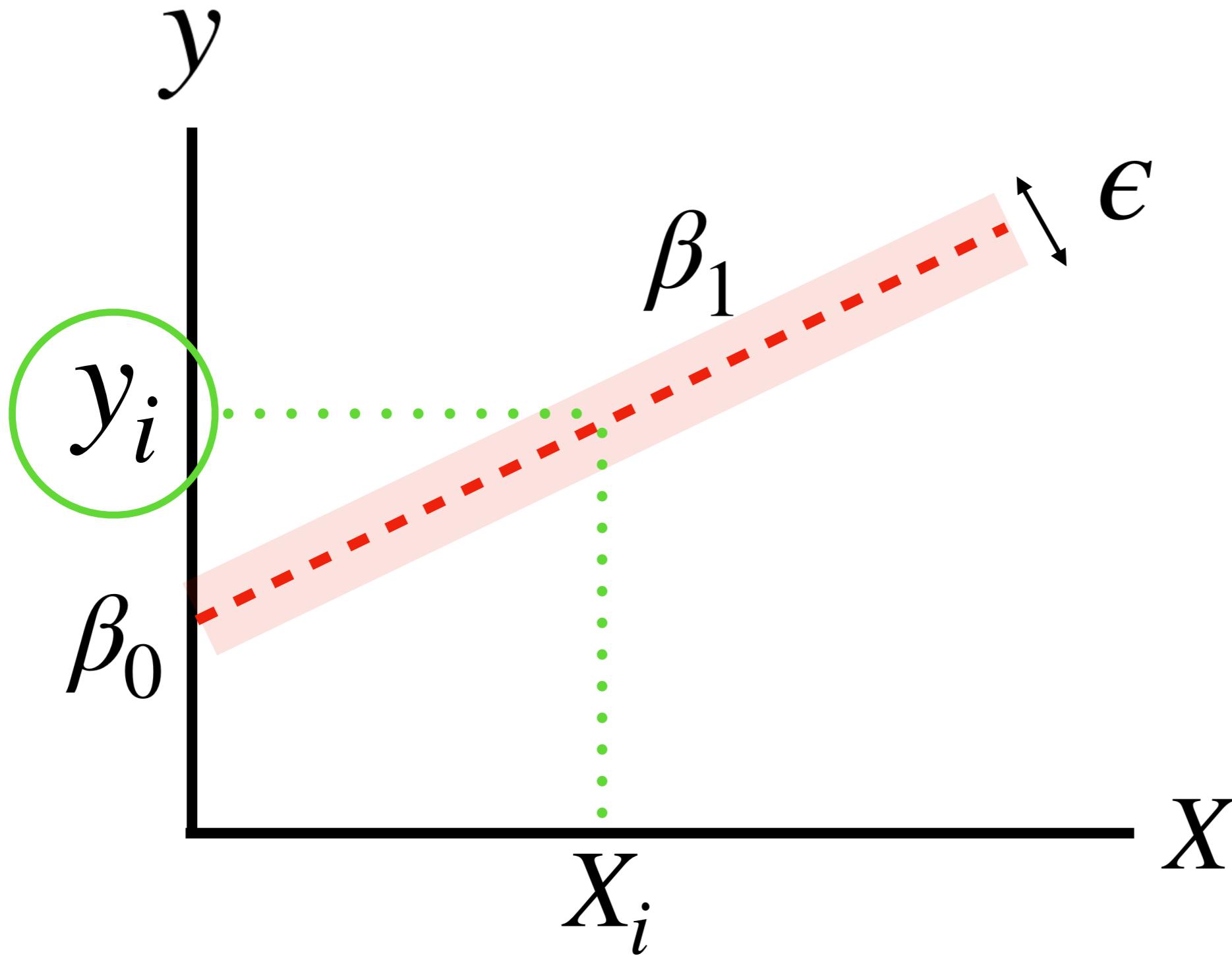
[https://github.com/katiesevans/IGP\\_biostatistics](https://github.com/katiesevans/IGP_biostatistics)

# **Extra Slides**

# Predicting values



# Predicting values



# Predicting values

```
# assign linear model to variable
model <- lm(happiness ~ income, data = income_data)

# generate predicted data for set of values
predicted_data <- data.frame(income = seq(1.5, 7.5, 0.1), %>%
  dplyr::mutate(predicted_happiness = predict(model, newdata = .)))
```

Income from \$15k to \$75k at intervals of \$1,000

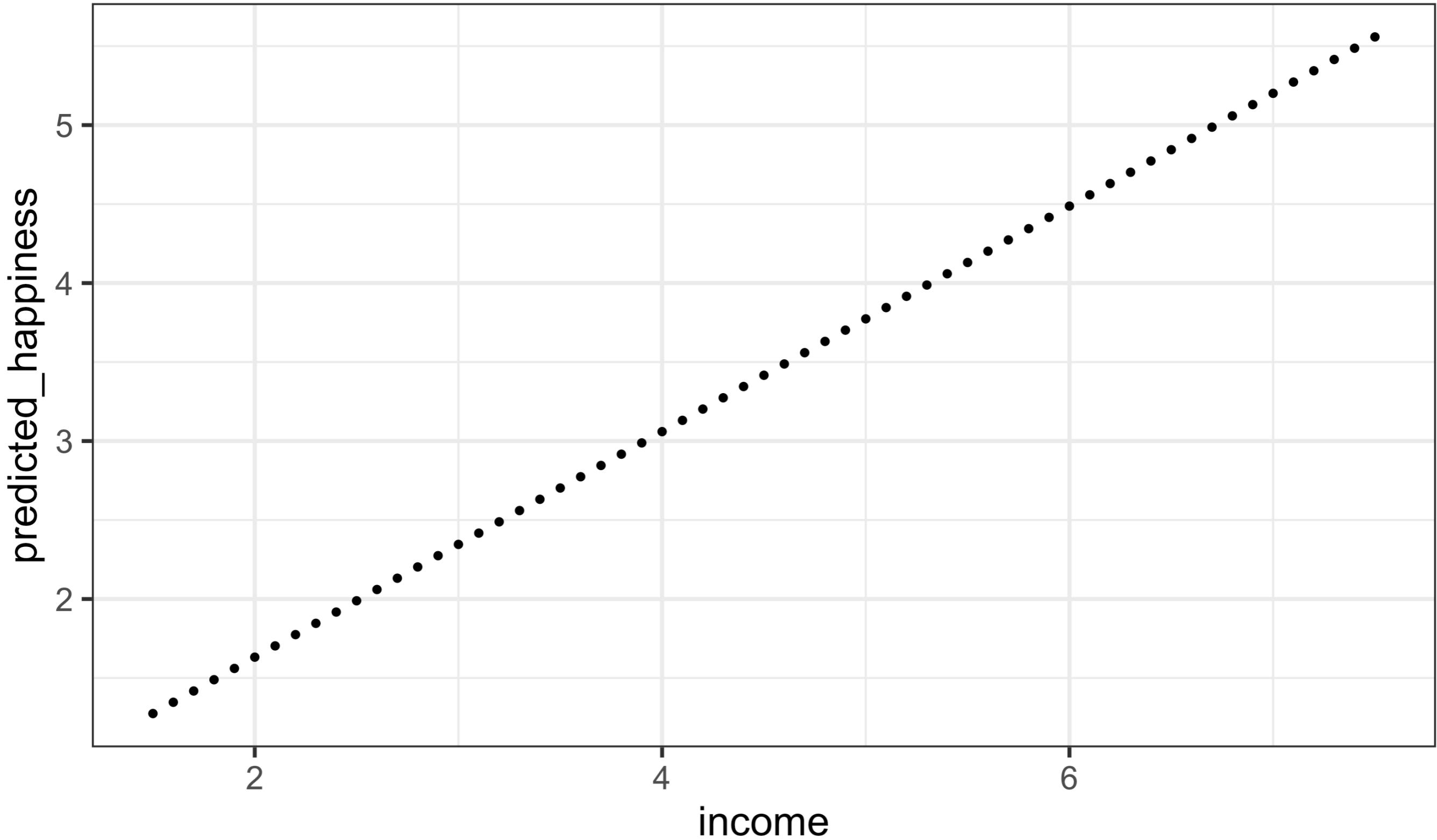
```
# plot predicted data
ggplot2::ggplot(predicted_data) +
  ggplot2::aes(x = income, y = predicted_happiness) +
  ggplot2::geom_point()

# plot in base R
plot(predicted_data$income, predicted_data$predicted_happiness)
```



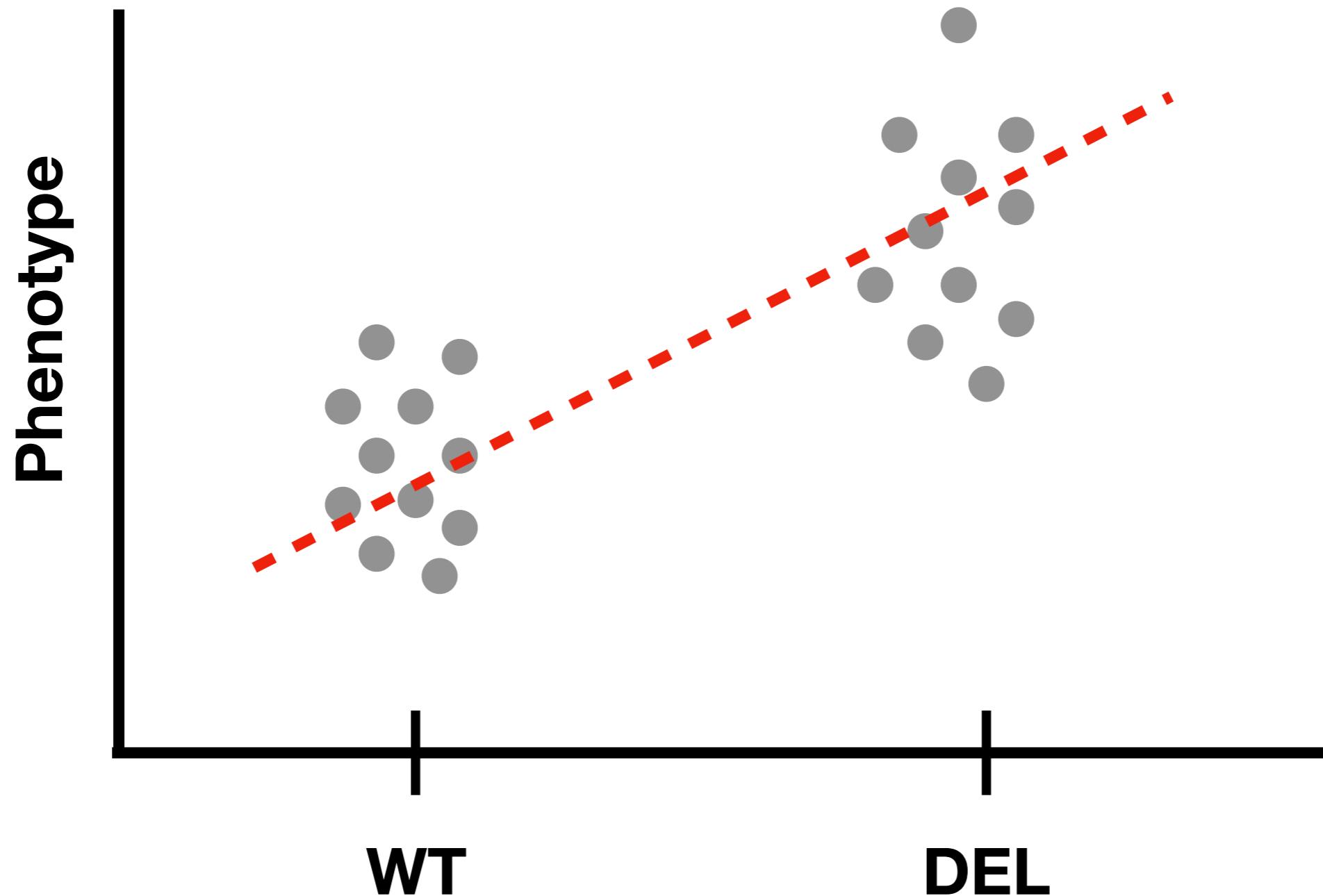
**Important to not predict values outside of the data range!**

# Predicting values



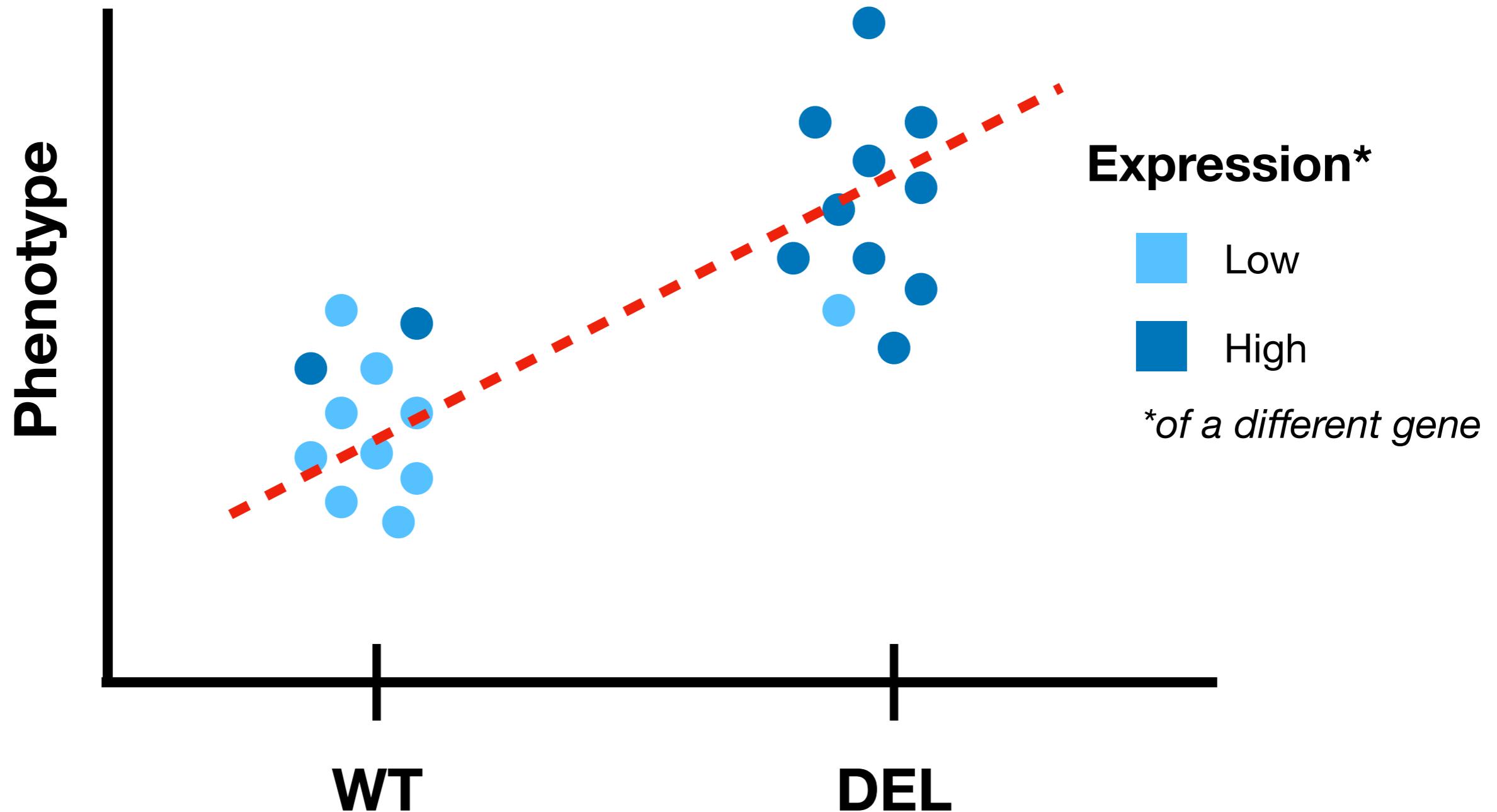
# Using residual values

`lm(phenotype ~ strain)`

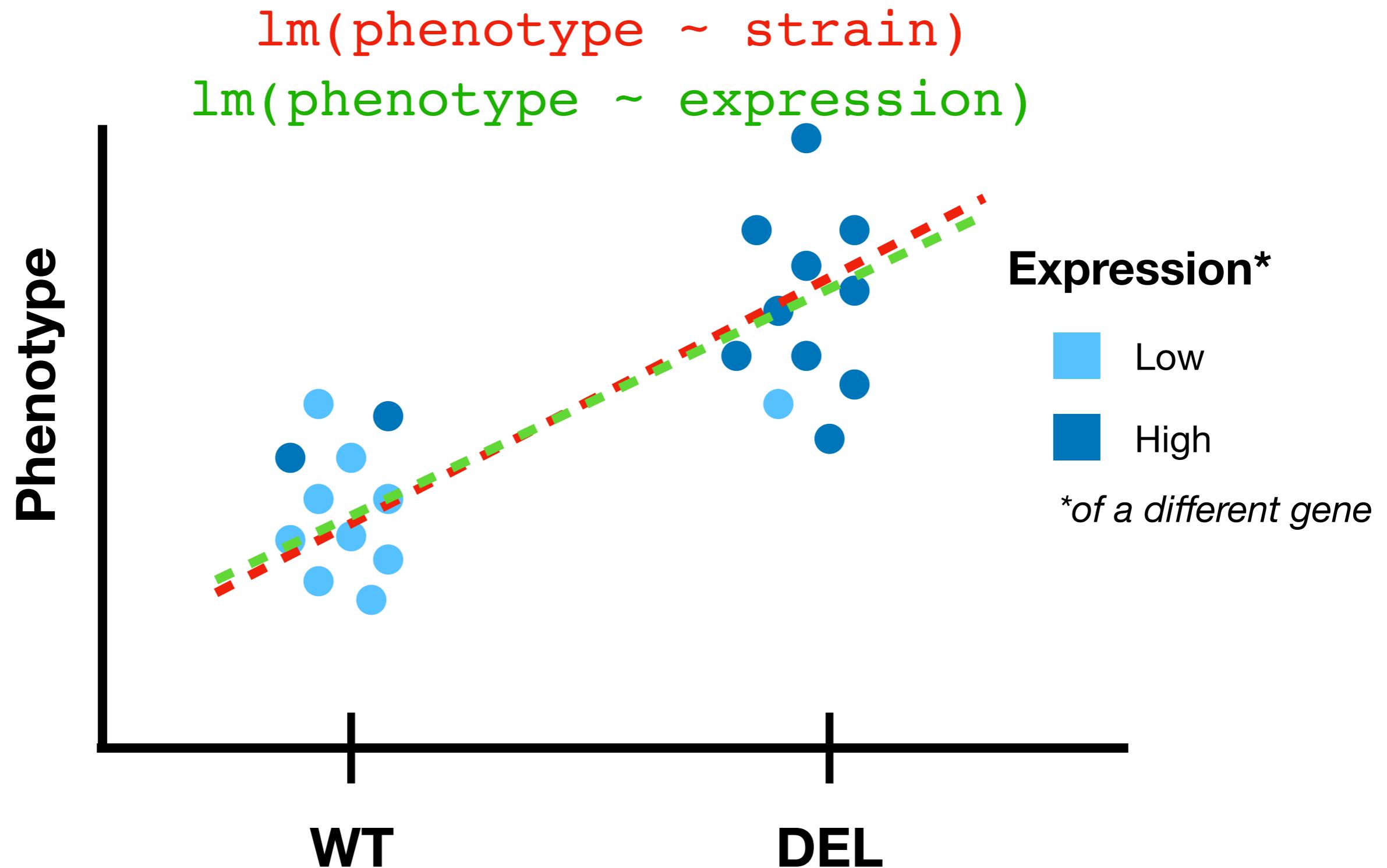


# Using residual values

`lm(phenotype ~ strain)`



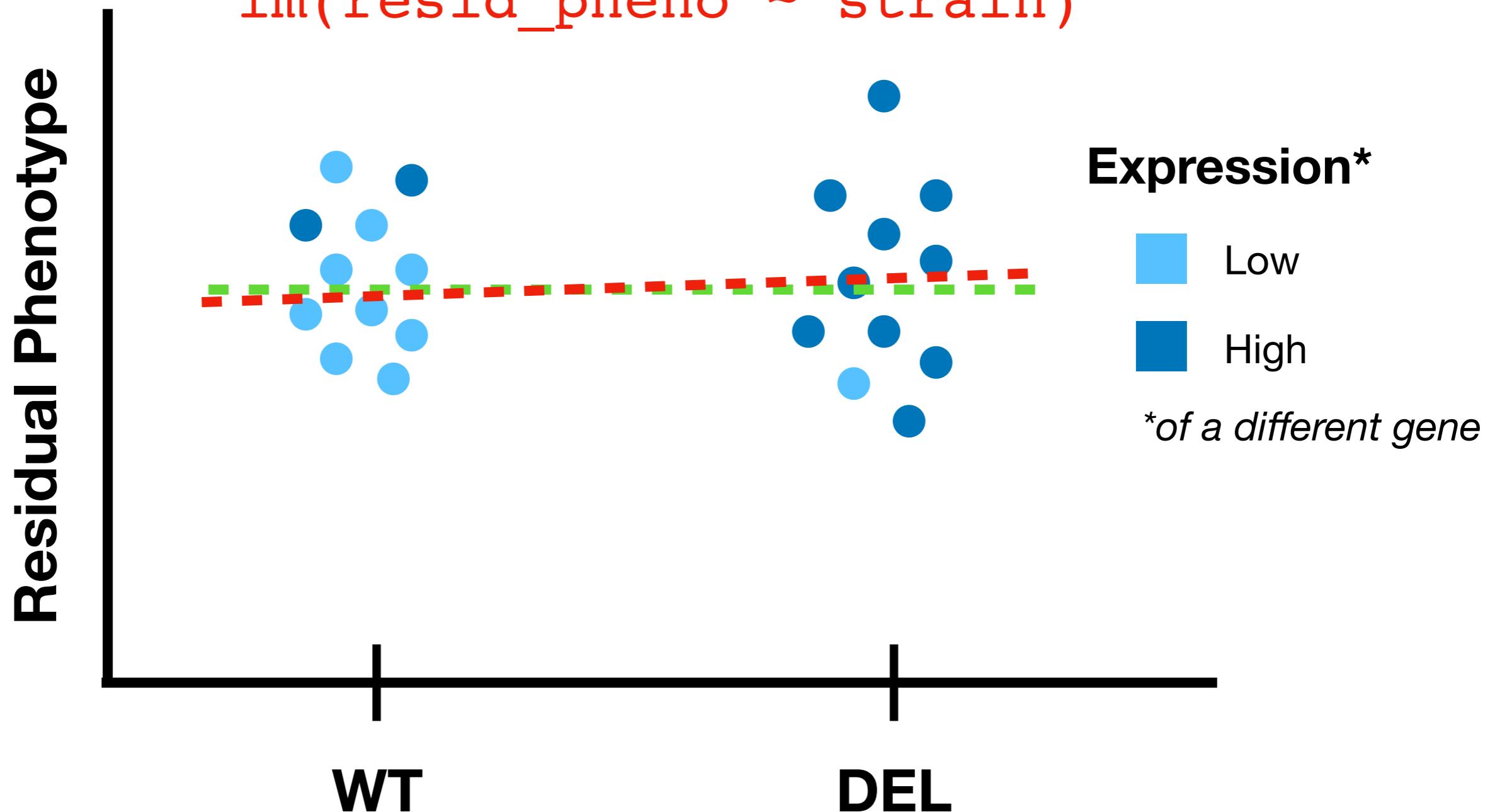
# Using residual values



# Using residual values

```
resid(lm(phenotype ~ expression))
```

```
lm(resid_pheno ~ strain)
```



# Using residual values

Great package for  
tidy statistics!

broom::augment(model)

| happiness | income   | .fitted  | .resid      | .std.resid  | .hat        | .sigma    | .cooksdi     |
|-----------|----------|----------|-------------|-------------|-------------|-----------|--------------|
| 2.3144890 | 3.862647 | 2.961527 | -0.64703769 | -0.90205658 | 0.002251376 | 0.7182356 | 9.180461e-04 |
| 3.4334898 | 4.979381 | 3.758680 | -0.32519010 | -0.45334273 | 0.002183070 | 0.7186765 | 2.248227e-04 |
| 4.5993734 | 4.923957 | 3.719116 | 0.88025693  | 1.22713119  | 0.002147257 | 0.7177335 | 1.620203e-03 |
| 2.7911138 | 3.214372 | 2.498771 | 0.29234235  | 0.40772807  | 0.003053610 | 0.7187050 | 2.545969e-04 |
| 5.5963983 | 7.196409 | 5.341251 | 0.25514736  | 0.35655407  | 0.006973369 | 0.7187334 | 4.463777e-04 |
| 2.4585559 | 3.729643 | 2.866585 | -0.40802919 | -0.56888086 | 0.002370292 | 0.7185909 | 3.844547e-04 |
| 3.1929918 | 4.674517 | 3.541060 | -0.34806836 | -0.48520140 | 0.002036760 | 0.7186549 | 2.402367e-04 |
| 1.9071368 | 4.498104 | 3.415132 | -1.50799483 | -2.10208970 | 0.002008681 | 0.7156164 | 4.446893e-03 |
| 2.9424499 | 3.121631 | 2.432570 | 0.50987997  | 0.71118370  | 0.003214180 | 0.7184589 | 8.154585e-04 |
| 3.7379416 | 4.639914 | 3.516360 | 0.22158191  | 0.30888007  | 0.002027982 | 0.7187563 | 9.693831e-05 |
| 3.1754061 | 4.632840 | 3.511309 | -0.33590329 | -0.46824107 | 0.002026383 | 0.7186666 | 2.225930e-04 |
| 2.0090465 | 2.773179 | 2.183836 | -0.17478978 | -0.24388421 | 0.003919933 | 0.7187824 | 1.170366e-04 |
| 5.9518141 | 7.119479 | 5.286336 | 0.66547825  | 0.92983921  | 0.006697419 | 0.7181987 | 2.914819e-03 |
| 5.9605473 | 7.466653 | 5.534158 | 0.42638937  | 0.59616497  | 0.008005262 | 0.7185679 | 1.434066e-03 |

# Using residual values

Great package for  
tidy statistics!

broom::augment(model)

| happiness | income   | .fitted  | .resid      | .std.resid  | .hat        | .sigma    | .cooksdi     |
|-----------|----------|----------|-------------|-------------|-------------|-----------|--------------|
| 2.3144890 | 3.862647 | 2.961527 | -0.64703769 | -0.90205658 | 0.002251376 | 0.7182356 | 9.180461e-04 |
| 3.4334898 | 4.979381 | 3.758680 | -0.32519010 | -0.45334273 | 0.002183070 | 0.7186765 | 2.248227e-04 |
| 4.5993734 | 4.923957 | 3.719116 | 0.88025693  | 1.22713119  | 0.002147257 | 0.7177335 | 1.620203e-03 |
| 2.7911138 | 3.214372 | 2.498771 | 0.29234235  | 0.40772807  | 0.003053610 | 0.7187050 | 2.545969e-04 |
| 5.5963983 | 7.196409 | 5.341251 | 0.25514736  | 0.35655407  | 0.006973369 | 0.7187334 | 4.463777e-04 |
| 2.4585559 | 3.729643 | 2.866585 | -0.40802919 | -0.56888086 | 0.002370292 | 0.7185909 | 3.844547e-04 |
| 3.1929918 | 4.674517 | 3.541060 | -0.34806836 | -0.48520140 | 0.002036760 | 0.7186549 | 2.402367e-04 |
| 1.9071368 | 4.498104 | 3.415132 | -1.50799483 | -2.10208970 | 0.002008681 | 0.7156164 | 4.446893e-03 |
| 2.9424499 | 3.121631 | 2.432570 | 0.50987997  | 0.71118370  | 0.003214180 | 0.7184589 | 8.154585e-04 |
| 3.7379416 | 4.639914 | 3.516360 | 0.22158191  | 0.30888007  | 0.002027982 | 0.7187563 | 9.693831e-05 |
| 3.1754061 | 4.632840 | 3.511309 | -0.33590329 | -0.46824107 | 0.002026383 | 0.7186666 | 2.225930e-04 |
| 2.0090465 | 2.773179 | 2.183836 | -0.17478978 | -0.24388421 | 0.003919933 | 0.7187824 | 1.170366e-04 |
| 5.9518141 | 7.119479 | 5.286336 | 0.66547825  | 0.92983921  | 0.006697419 | 0.7181987 | 2.914819e-03 |
| 5.9605473 | 7.466653 | 5.534158 | 0.42638937  | 0.59616497  | 0.008005262 | 0.7185679 | 1.434066e-03 |