Linear Models

Statistics with R

Basel R Bootcamp

April 2019

Linear Models

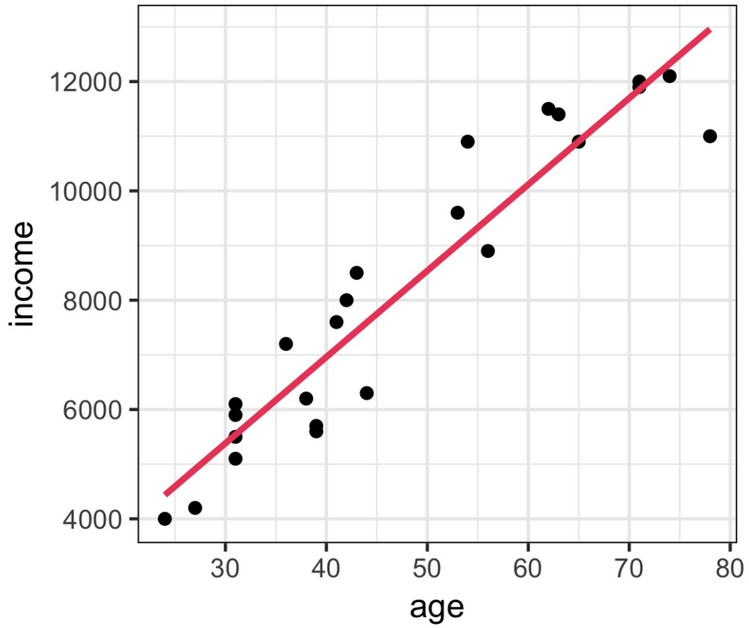
Linear models are, by far, **the most important models in all of statistics**.

Many statistical tests you may know of are special types of linear models.

Why are linear models so great?

- They are **easy to interpret**.
- They can **approximate non-linear** data well.
- They are **easy to calculate** and implement (just addition and multiplication!).
- They **just work**!

$$\Large income = 885 + age \times 149.3$$



What is a linear model?

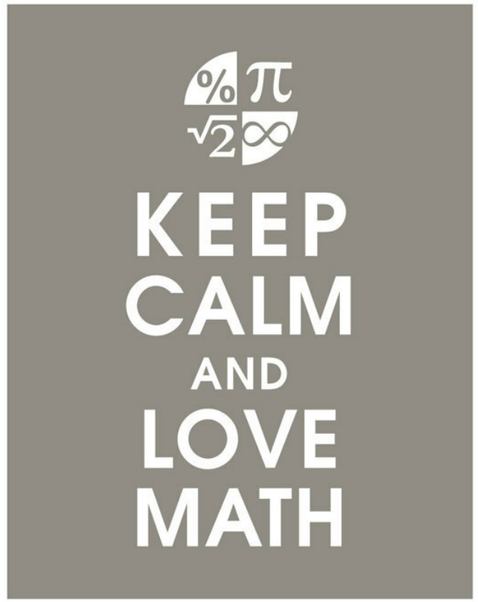
A linear model is **just addition and multiplication** and can be written in the following forms:

Version 1

$$\huge y = \beta_{0} + \beta_{1}x_{1} + \beta_{2} x_{2} + ... + \beta_{n}x_{n} + \epsilon$$

Version 2

$$\huge y = \beta_{0} + \sum_{i=1}^{n}\beta_{i}x_{i}+ \epsilon$$



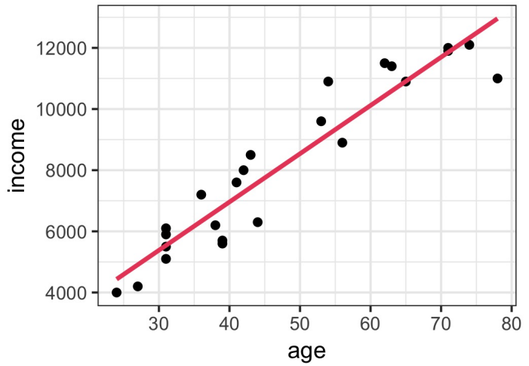
from media.tumblr.com

Simple Linear Regression

Definition: Simple linear regression is a linear model with one predictor \(x\), and where the error term \(\epsilon\) is Normally distributed.

$$\huge y=\beta_{0} + \beta_{1} x + \epsilon$$

Parameter	Description	In words
β_0	Intercept	When x = 0, what is the predicted value for y?
β_1	Coefficient for x	For every increase of 1 in x, how does y change?



Formula

$$\Large income = 885 + 149.3 \times age + \epsilon$$

Coefficients

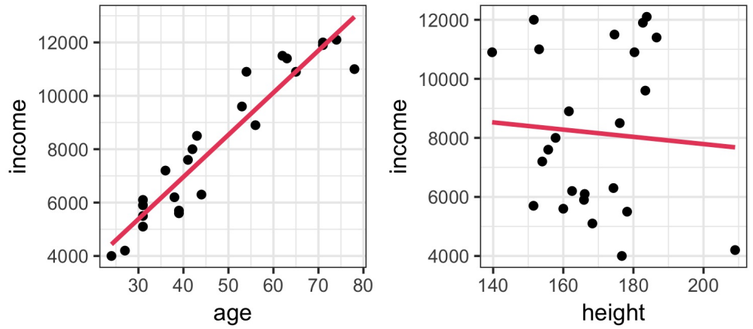
$$\Large \beta_{0} = 885, \beta_{age} = 149.3$$

Multiple Linear Regression

Definition: Multiple linear regression is a linear model with many predictors \(x_{1}, x_{2}, ... x_{n}\), and where the error term \(\epsilon\) is Normally distributed.

$$\LARGE y=\beta_{0} + \beta_{1}x_{1} + \beta_{2} x_{2}+...+ \beta_{n} x_{n} + \epsilon$$

Parameter	Description	In words
β_0	Intercept	When all x values are 0, what is the predicted value for y?
β_1, β_2, ...	Coefficient for x_1, x_2, ...	For every increase of 1 in coefficient for x_1, x_2, ... how does y change?



Formula

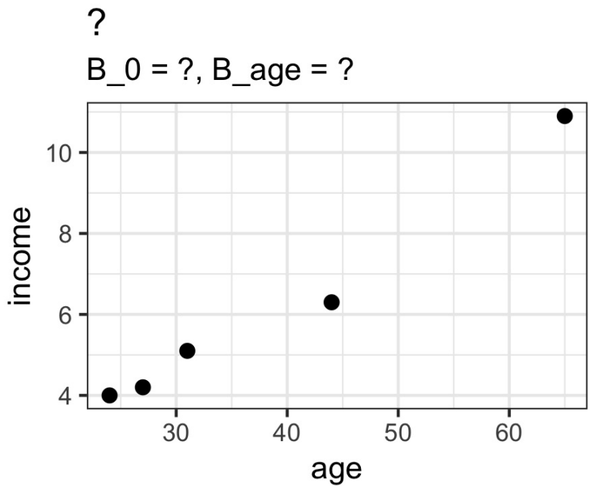
$$\large income = 1628 + 147 \times age - 4.1 \times height + \epsilon$$

Coefficients

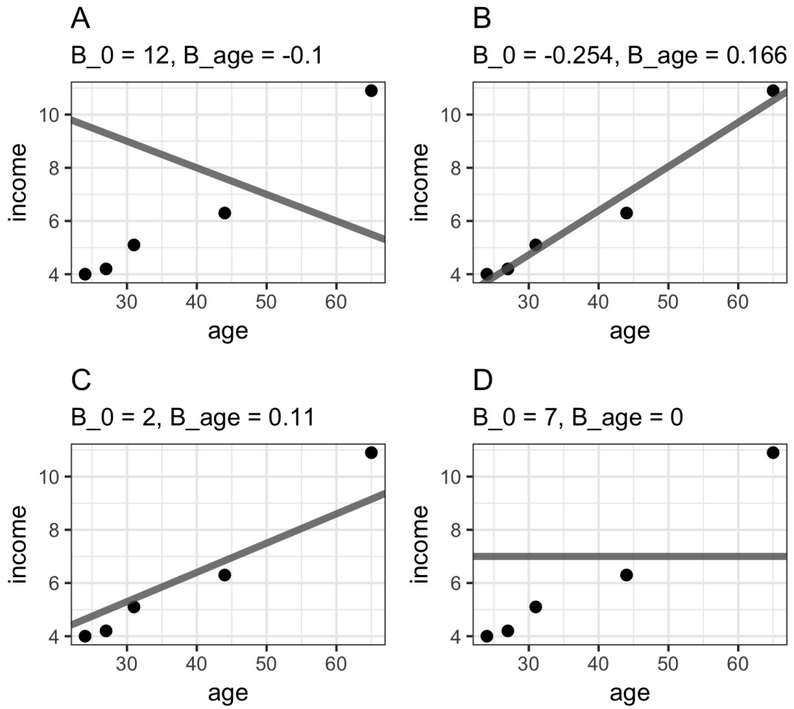
$$\large \beta_{0} = 1628, \beta_{age} = 147, \beta_{weight}=-4.1$$

Estimating coefficients

How do we estimate the 'right' coefficients in a linear model? **Which of the four would be a good fit?**

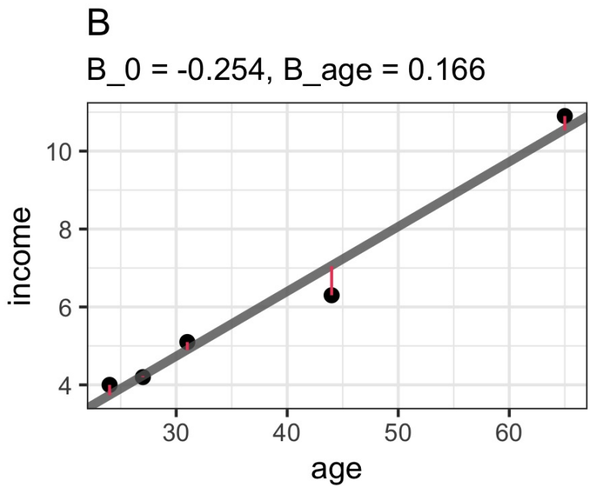


$$\Large income = \beta_{0} + \beta_{1} \times age + \epsilon$$

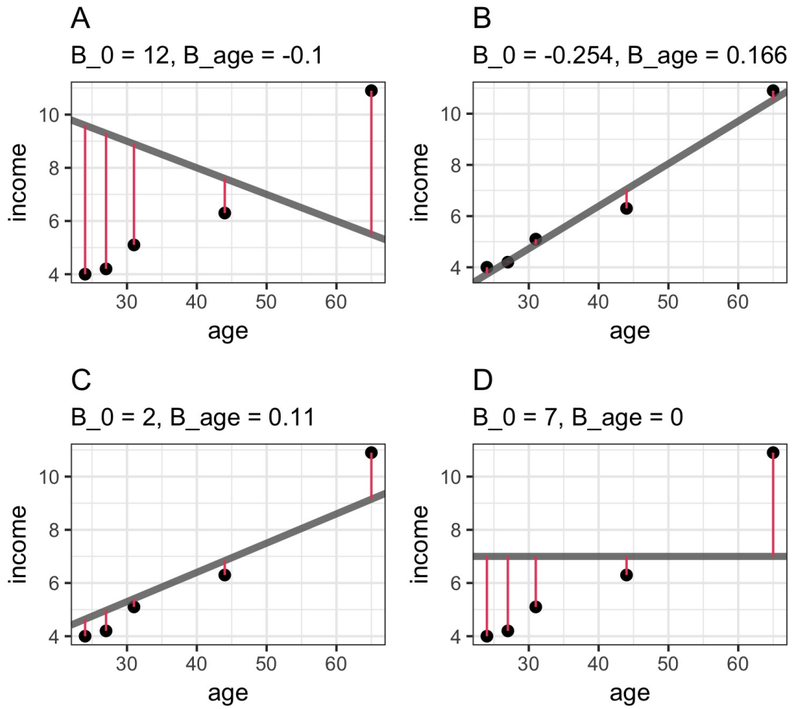


Estimating coefficients

How do we estimate the 'right' coefficients in a linear model? **Which of the four would be a good fit?**



$$\Large income = -0.254 + 0.167 \times age + \epsilon$$



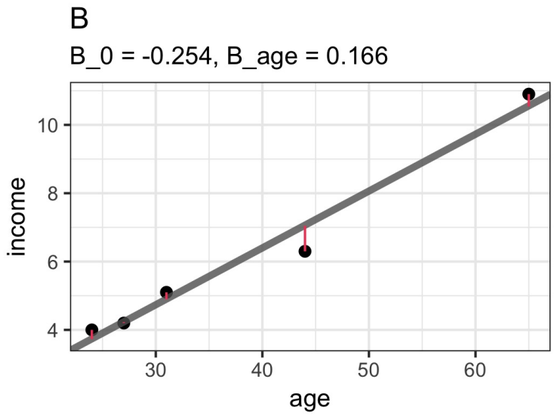
Estimating coefficients

How do we estimate the 'right' coefficients in a linear model?

Find the values that minimise **Mean Squared Error** (MSE).

MSE: The average squared distance between the prediction and the data.

$$\Large MSE = \frac{1}{N}\sum_{i=1}^{n}(y_{i}-Prediction_{i})^2$$



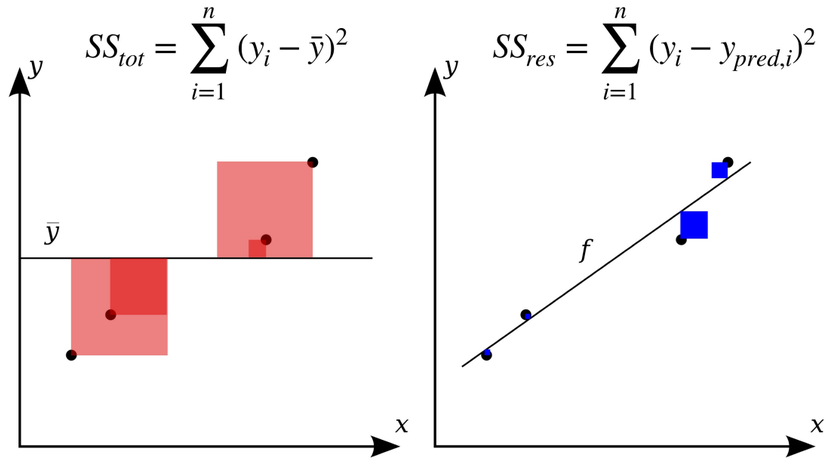
id	age	income	Prediction	SE
1	24	4.0	3.730	0.0729
2	27	4.2	4.228	0.0008
3	31	5.1	4.892	0.0433
4	44	6.3	7.050	0.5625
5	65	10.9	10.536	0.1325
			MSE	**0.16**

R^2 (R-Squared)

R-Squared \(R^{2}\) is the most common method of calculating the **overall performance** of a model.

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

R^2	Interpretation
0	Model explains no variance in y.
.5	Model explains half the variance in y.
1	Model explains half the variance in y.



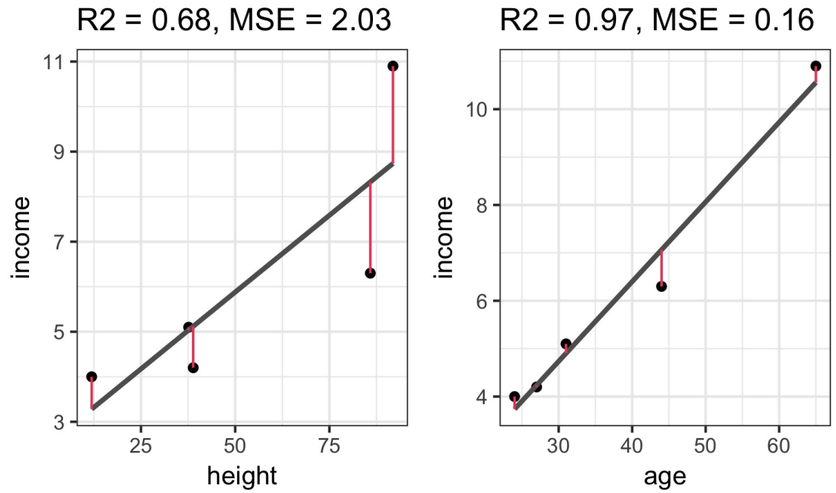
from Wikipedia

R^2 (R-Squared)

R-Squared \(R^{2}\) is the most common method of calculating the **overall performance** of a model.

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

R^2	Interpretation
0	Model explains no variance in y.
.5	Model explains half the variance in y.
1	Model explains half the variance in y.



How to fit and and dissect linear models in R

with `glm()`

Key Functions

Fitting

Function	Description
`glm(formula, data)`	Fit a linear model to data and calculate best coefficients

Evaluation

Function	Description
`coef(mod)`	Get coefficients from a model
`fitted(mod)`	Get fitted results.
`resid(mod)`	Get residuals (errors)

```
# Create a model income_glm

# Y = income
# X1 = age, X2 = children

income_glm <- glm(formula = income ~ age + children,
                  data = baselers)
```

id	income	age	children
1	6300	44	2
2	10900	65	2

Key Functions

Fitting

Function	Description
`glm(formula, data)`	Fit a linear model to data and calculate best coefficients

Evaluation

Function	Description
`coef(mod)`	Get coefficients from a model
`fitted(mod)`	Get fitted results.
`resid(mod)`	Get residuals (errors)

```
# Print income_glm

income_glm
```

```
##
## Call:  glm(formula = income ~ age + children, data = baselers)
##
## Coefficients:
## (Intercept)          age     children
##      871.10       149.25         7.78
##
## Degrees of Freedom: 8509 Total (i.e. Null);  8507 Residual
##   (1490 observations deleted due to missingness)
## Null Deviance:        6.33e+10
## Residual Deviance: 1.29e+10     AIC: 145000
```

Key Functions

Fitting

Function	Description
`glm(formula, data)`	Fit a linear model to data and calculate best coefficients

Evaluation

Function	Description
`coef(mod)`	Get coefficients from a model
`fitted(mod)`	Get fitted results.
`resid(mod)`	Get residuals (errors)

```
# Show summary info
summary(income_glm)
```

```
##
## Call:
## glm(formula = income ~ age + children, data = baselers)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -4250    -835       5     820    4779
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  871.101     44.959    19.4   <2e-16 ***
## age          149.249      0.818   182.5   <2e-16 ***
## children       7.777     12.861     0.6     0.55
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1513138)
```


Key Functions

Fitting

Function	Description
`glm(formula, data)`	Fit a linear model to data and calculate best coefficients

Evaluation

Function	Description
`coef(mod)`	Get coefficients from a model
`fitted(mod)`	Get fitted results.
`resid(mod)`	Get residuals (errors)

```
# Get fitted values (only first 5)
fitted(income_glm)

# Get residuals (only first 10)
resid(income_glm)
```

```
##     1     2     3     4     5
##  7454 10588  5513  4916  4461

##       1       2       3       4       5
## -1153.6   312.2  -413.4  -716.4  -460.9
```

id	income	age	children	fitted	resid
1	6300	44	2	7454	-1153.6
2	10900	65	2	10588	312.2

Key Functions

Fitting

Function	Description
`glm(formula, data)`	Fit a linear model to data and calculate best coefficients

Evaluation

Function	Description
`rsq(mod)`	Get R^2 value from model.
`tidy(mod)`	Get tidy results from a model.

```
library(rsq)
library(broom)

# Show R-squared from model
rsq(income_glm)
```

```
## [1] 0.7967
```

```
# Show 'tidy' results from my model
tidy(income_glm)
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   871.      45.0      19.4   7.11e-82
## 2 age           149.       0.818   183.    0.
## 3 children        7.78    12.9       0.605 5.45e- 1
```

Practical