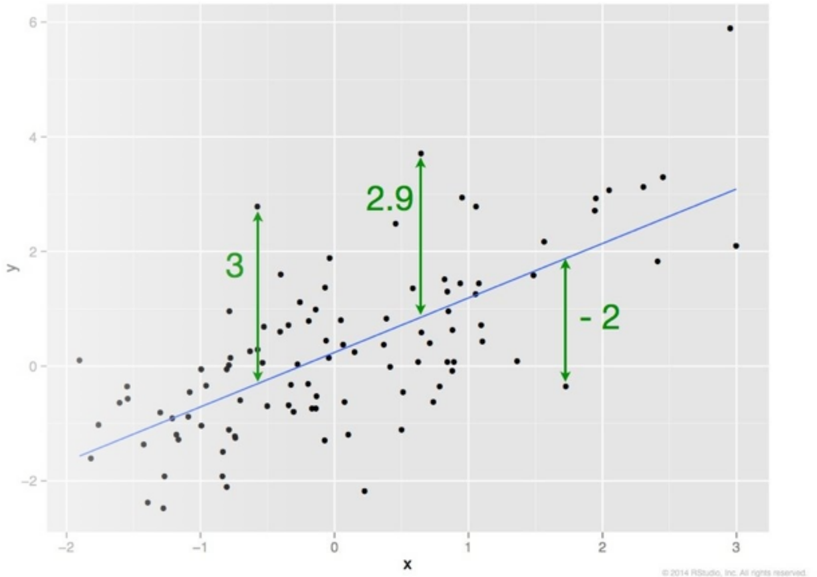
All Training materials are provided "as is" and without warranty and RStudio disclaims any and all express and implied warranties including without limitation the implied warranties of title, fitness for a particular purpose, merchantability and noninfringement.

Studio

Modeling Basics

Estimate the relationships within your data



Garrett Grolemund

Master Instructor, RStudio

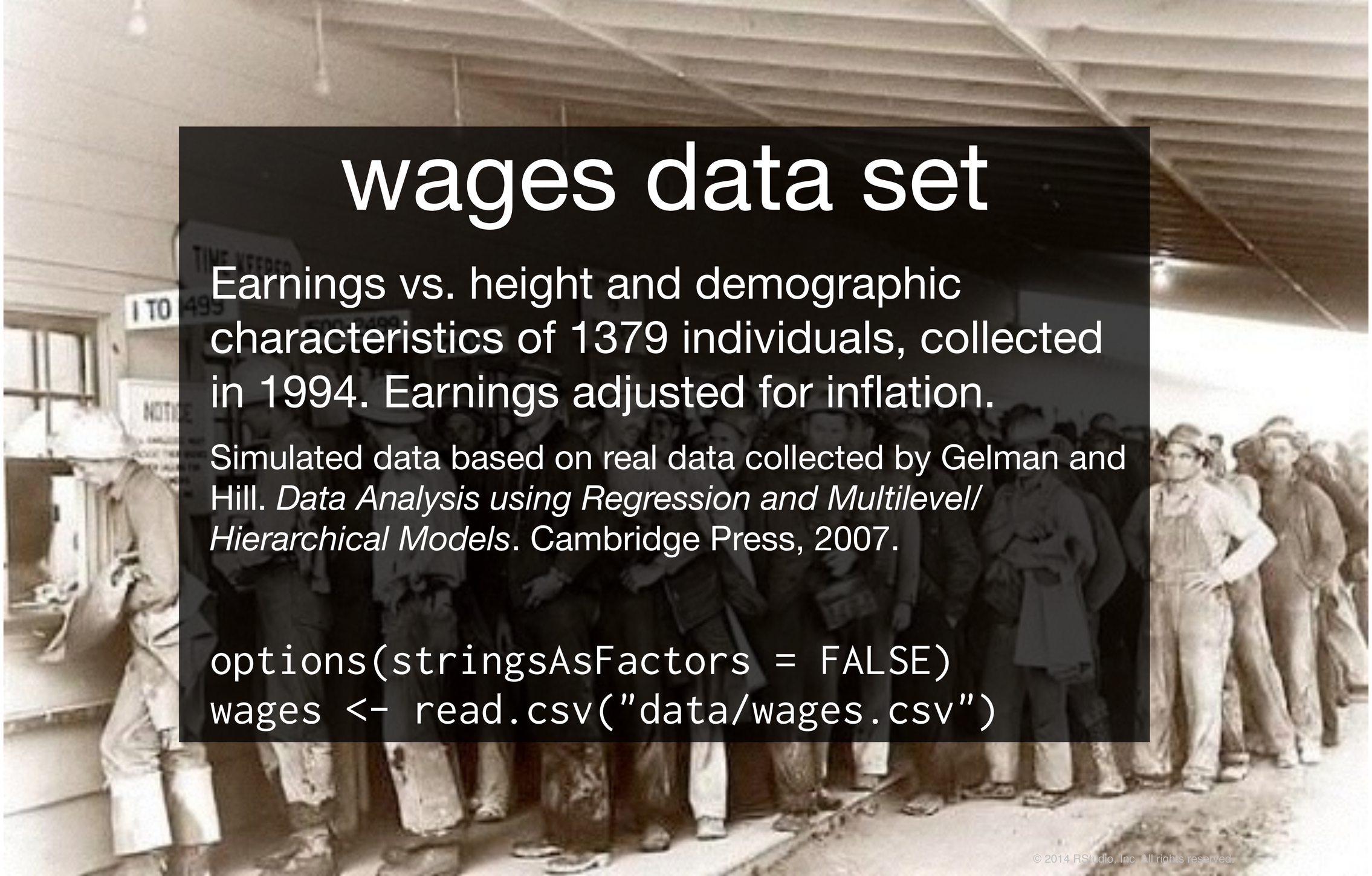
August 2014

1. Modeling
2. Linear regression
3. Multivariate regression
4. Interaction terms

Warm up

Do you think that taller people make more money?

Do you think that hotter places have more crime?



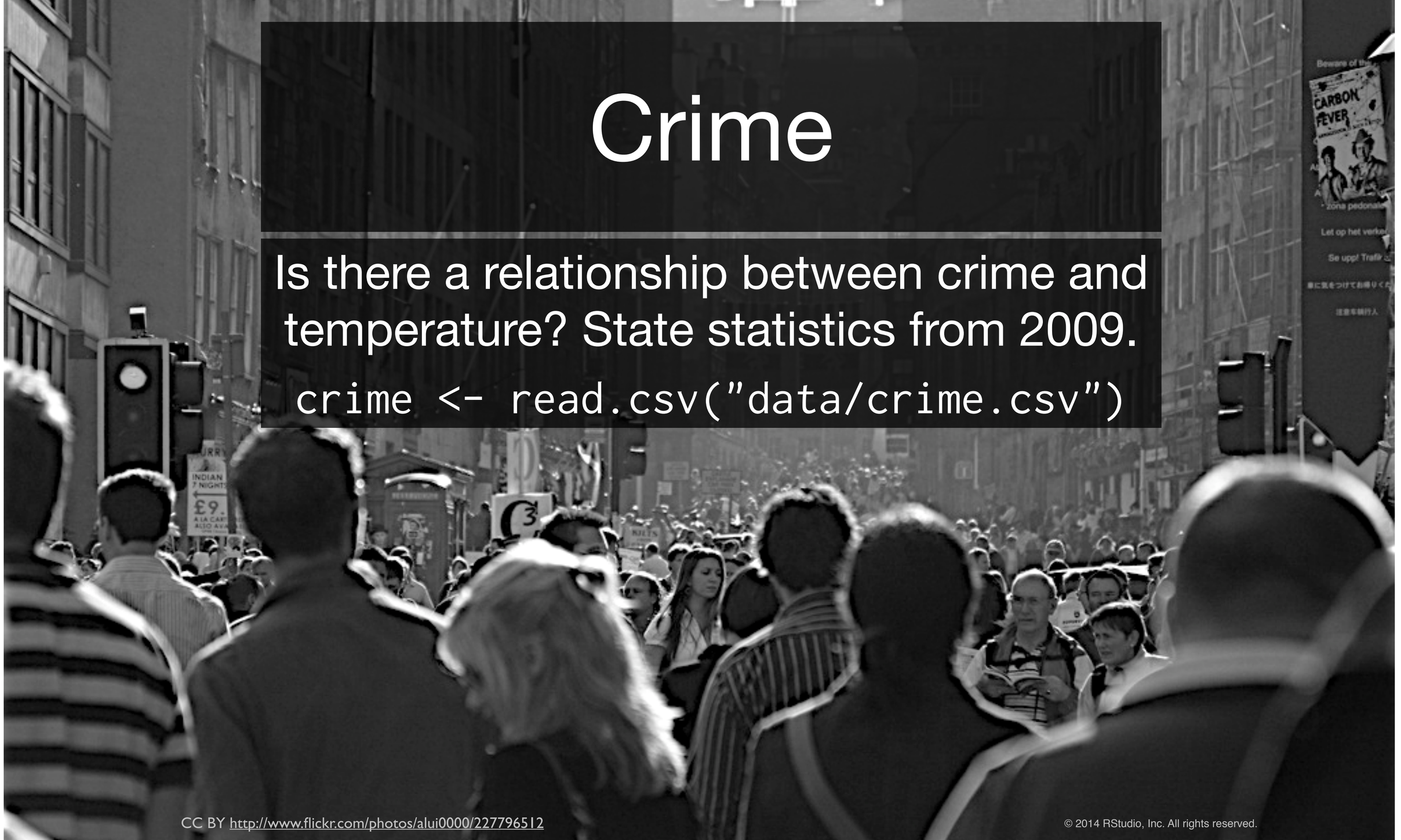
wages data set

Earnings vs. height and demographic characteristics of 1379 individuals, collected in 1994. Earnings adjusted for inflation.

Simulated data based on real data collected by Gelman and Hill. *Data Analysis using Regression and Multilevel/ Hierarchical Models*. Cambridge Press, 2007.

```
options(stringsAsFactors = FALSE)
wages <- read.csv("data/wages.csv")
```

```
library(dplyr)
tbl_df(wages)
## Source: local data frame [1,379 x 6]

##        earn height    sex     race ed age
## 1  79571.30  73.89   male    white 16  49
## 2  96396.99  66.23 female    white 16  62
## 3  48710.67  63.77 female    white 16  33
## 4  80478.10  63.22 female    other 16  95
## 5  82089.35  63.08 female    white 17  43
## 6  15313.35  64.53 female    white 15  30
## 7  47104.17  61.54 female    white 12  53
## 8  50960.05  73.29   male    white 17  50
## 9   3212.65  72.24   male hispanic 15  25
## 10 42996.64  72.40   male    white 12  30
## ..      ...    ...    ...      ... .. ...
```



Crime

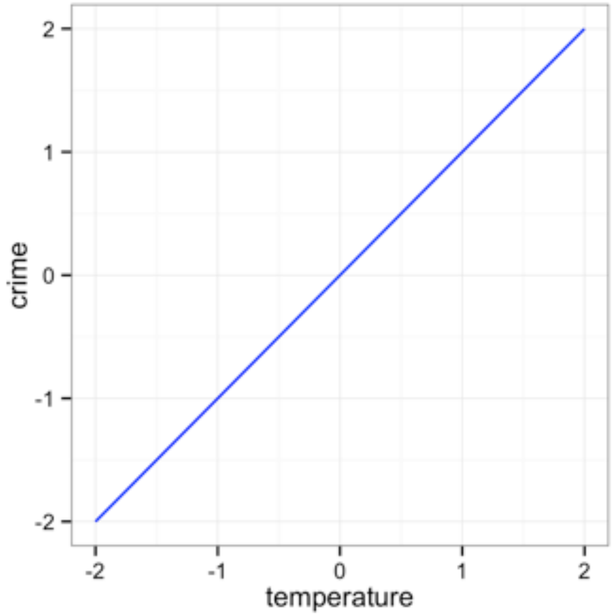
Is there a relationship between crime and temperature? State statistics from 2009.

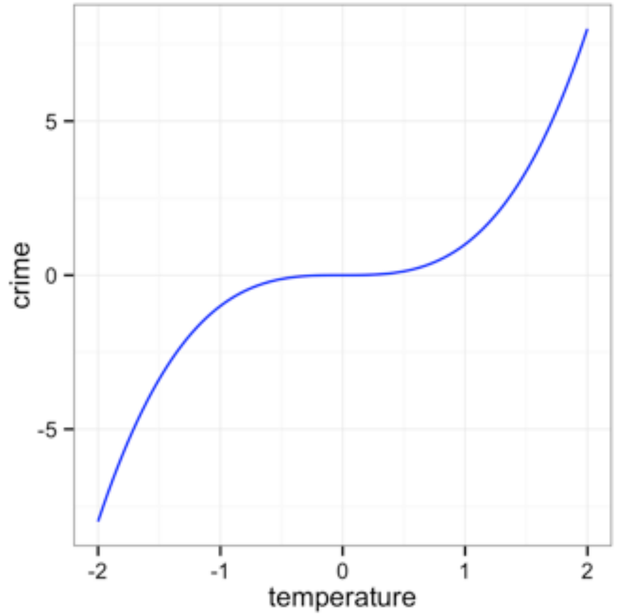
```
crime <- read.csv("data/crime.csv")
```

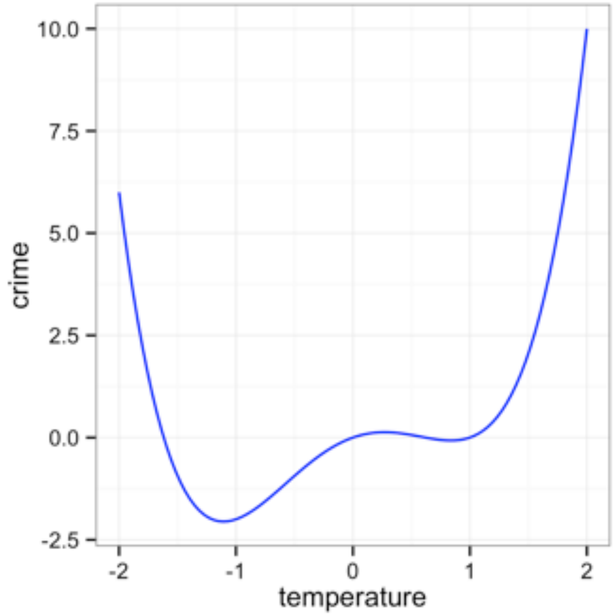
```
tbl_df(crime)
## Source: local data frame [48 x 5]

##           state abbr low murder tc2009
## 1       Alabama   AL -27    7.1 4337.5
## 2        Alaska   AK -80    3.2 3567.1
## 3       Arizona   AZ -40    5.5 3725.2
## 4      Arkansas   AR -29    6.3 4415.4
## 5    California   CA -45    5.4 3201.6
## 6      Colorado   CO -61    3.2 3024.5
## 7   Connecticut   CT -32    3.0 2646.3
## 8      Delaware   DE -17    4.6 3996.8
## 9       Florida   FL  -2    5.5 4453.7
## 10      Georgia   GA -17    6.0 4180.6
```

Modeling

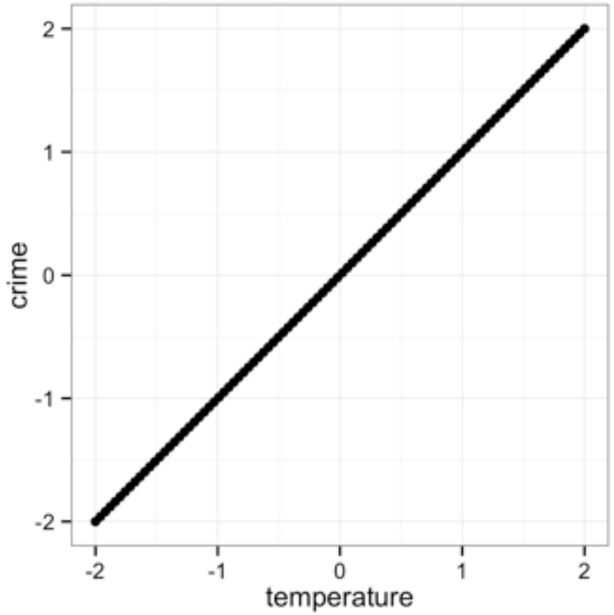
A function is a mathematical description of a relationship.

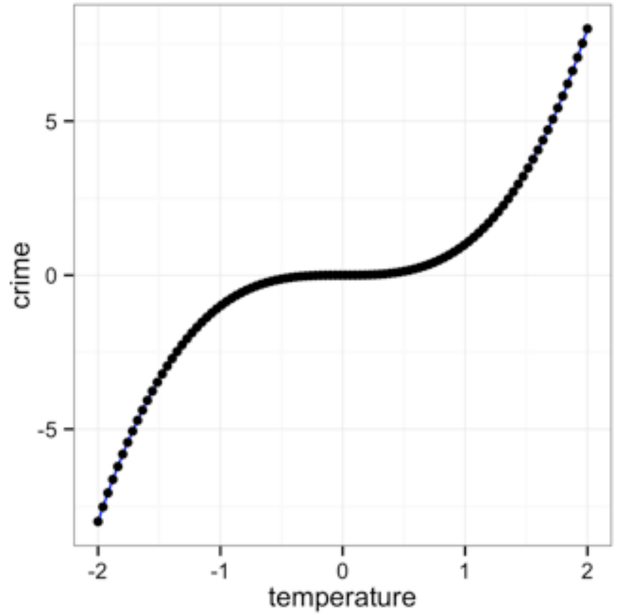


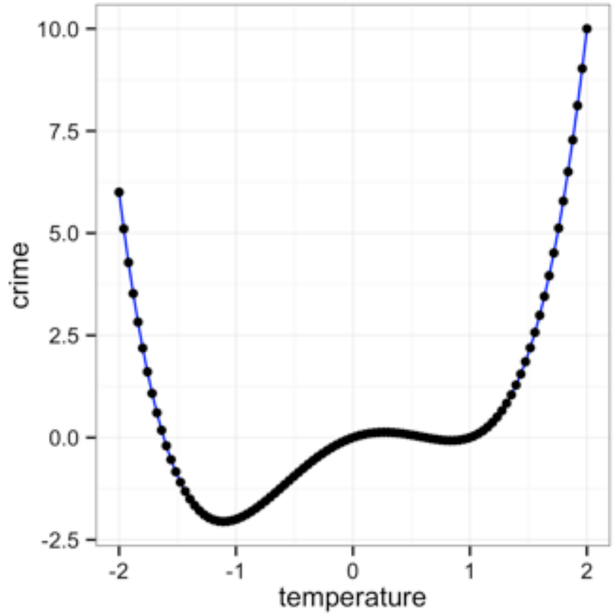




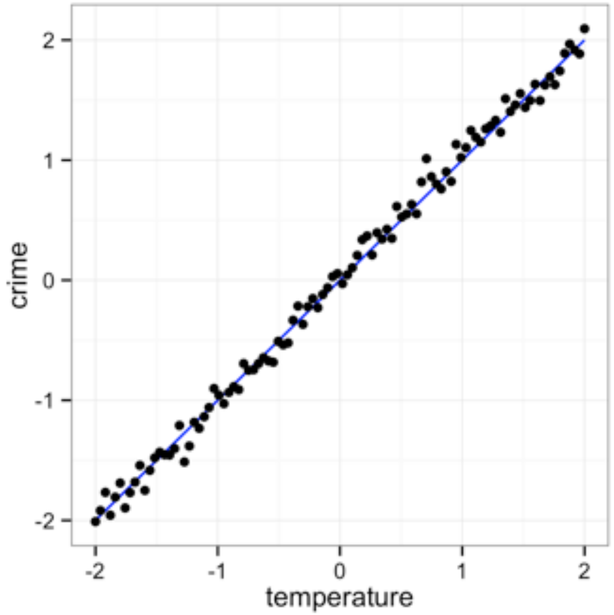
If one variable completely determines another, every (x, y) data point will fall on the function line.

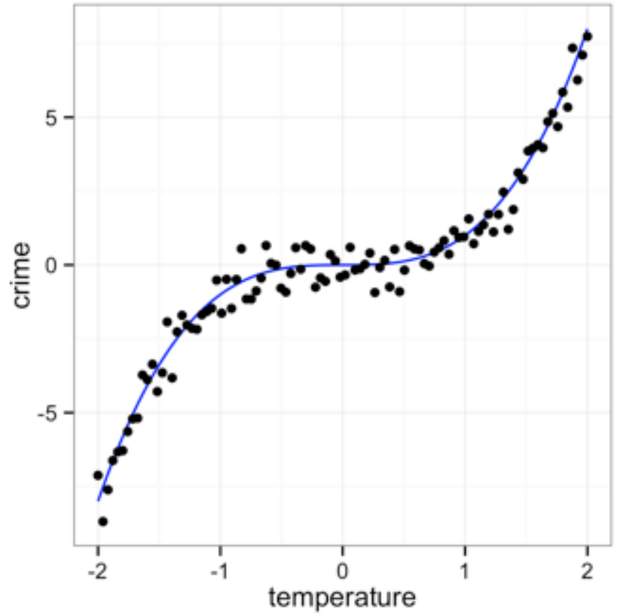


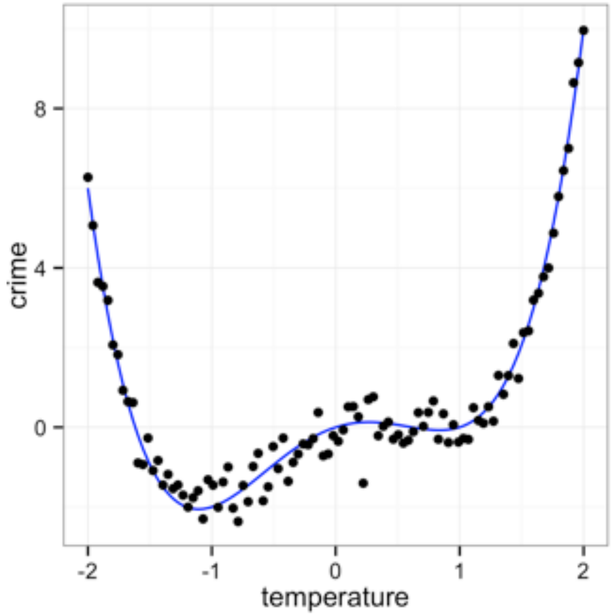




If the relationship is also affected by other variables, data points may not fall directly on the function line.

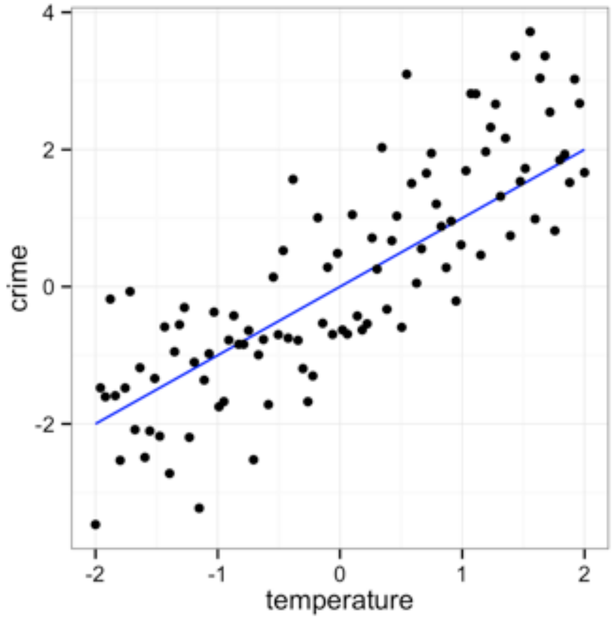


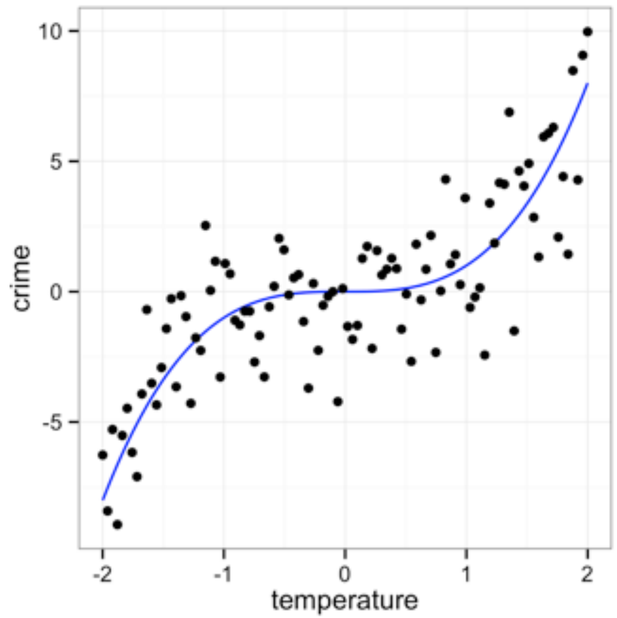


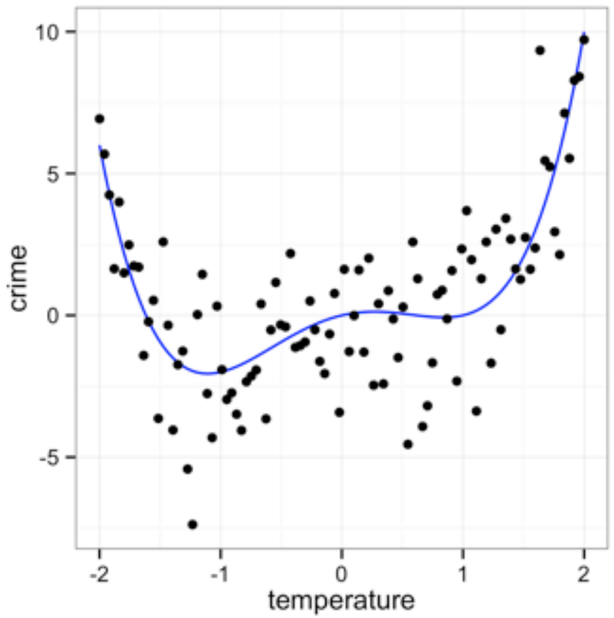


The greater the effect of other variables, the weaker the relationship. This is normally the situation with real data.

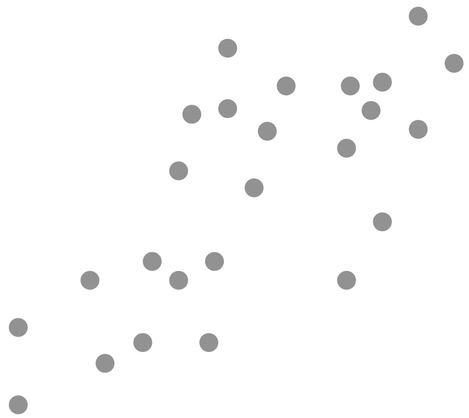
Modeling attempts to correctly identify relationships in noisy data.







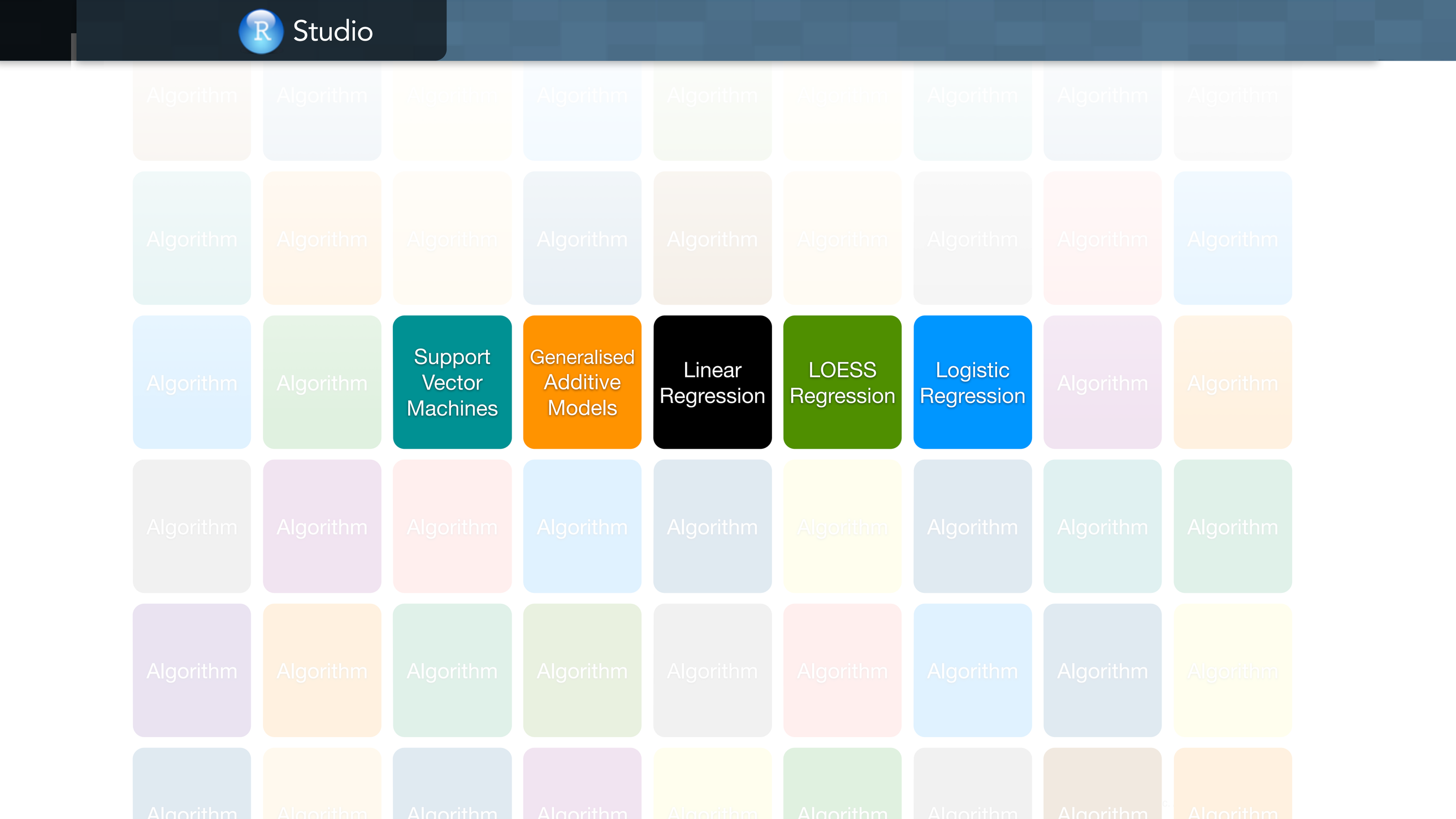
What is a model?



Algorithm

Data

Model


Studio
Algorithm
Support Vector Machines
Generalised Additive Models
Linear Regression
LOESS Regression
Logistic Regression

Linear Regression

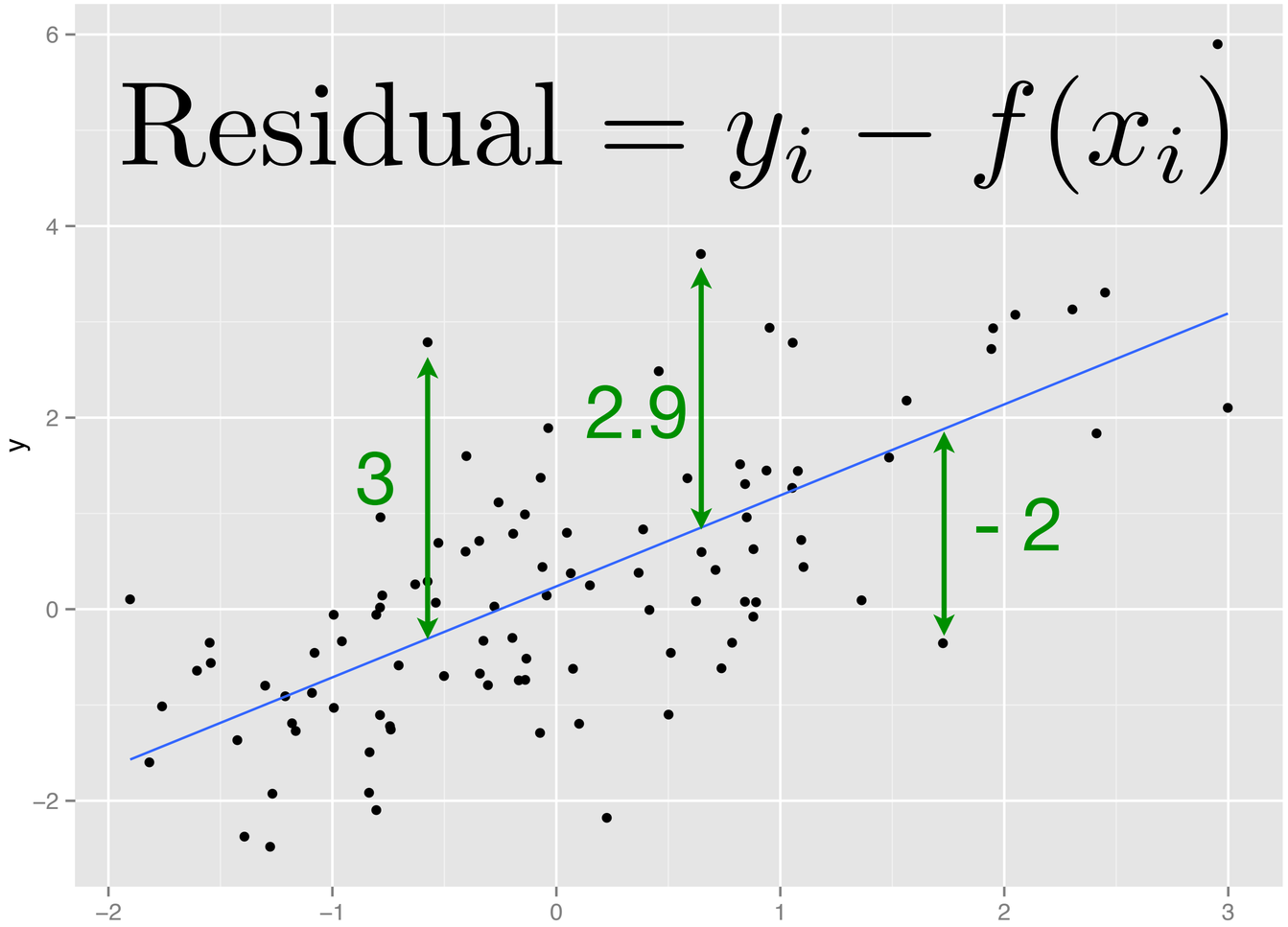
Linear models

The linear regression algorithm constrains $f(\hat{x})$ to have the form

$$f(\hat{x}) = \alpha + \beta x + \epsilon$$

e.g., $f(\hat{x})$ will be a straight line in x.

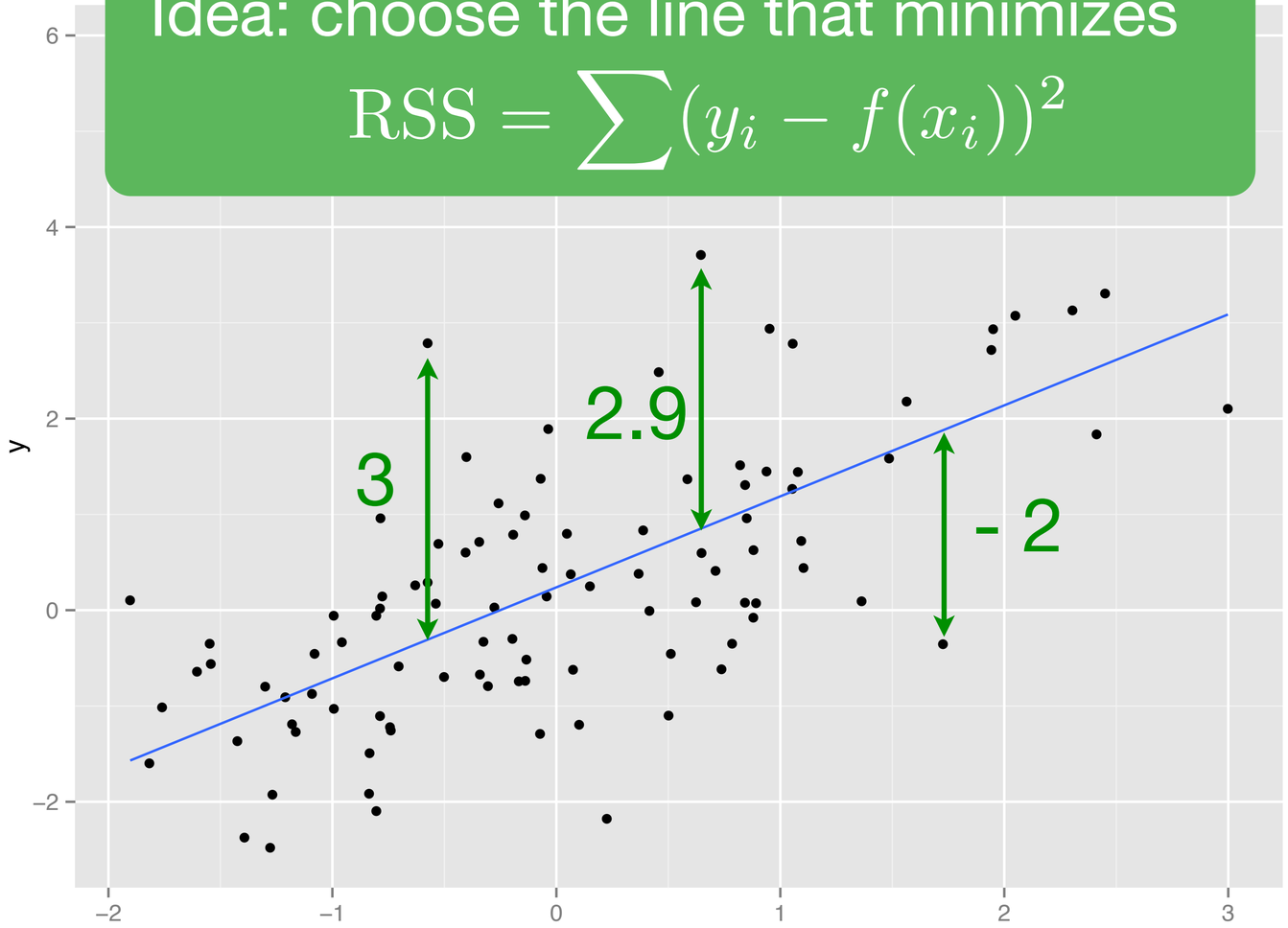
How to fit the best line?



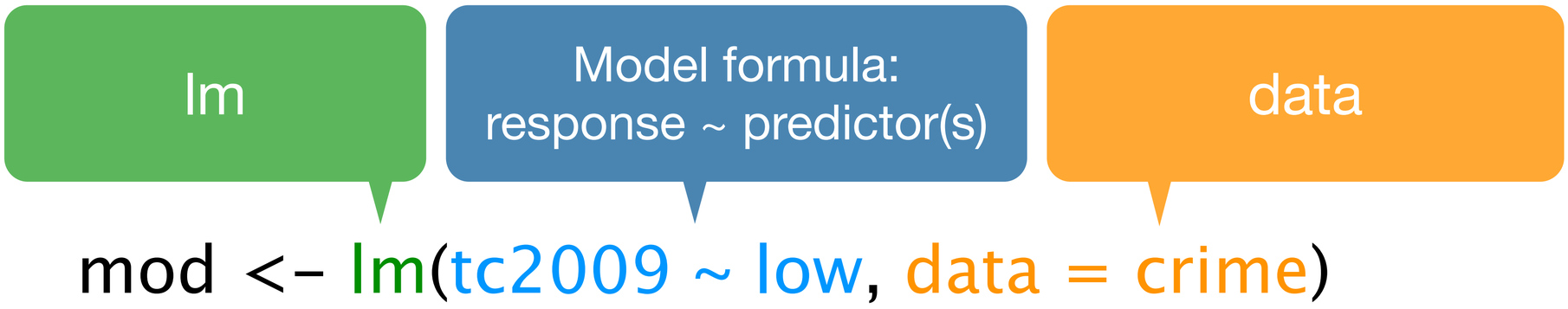
How to fit the best line?

Idea: choose the line that minimizes

$$\text{RSS} = \sum (y_i - f(x_i))^2$$



linear model syntax



formulas

R formulas are expressions built with ~

```
tc2009 ~ low
# tc2009 ~ low

class(tc2009 ~ low)
# [1] "formula"
```

Formulas only need to include the response and predictor variables

$$y = \alpha + \beta x + \epsilon$$

```
y ~ x
```

```
response ~ explanatory

dependent ~ independent

outcome ~ predictors
```

Modeling functions

```
mod <-    lm (tc2009 ~ low, data = crime)
```

nlm

All share same syntax

Your turn

Fit a linear model to the `crime` data set. Predict `tc2009` with `low`. Can you tell what function describes the best fit line?

$$y = \ ? \ + \ ? \ x + \epsilon$$

```
mod <- lm(tc2009 ~ low, data = crime)
mod
```

```
## Call:
## lm(formula = tc2009 ~ low, data = crime)

## Coefficients:
## (Intercept)          low
##     4256.86        21.65
```

Always save your model object. There's info in it

```
names(mod)
##  [1] "coefficients"  "residuals"     "effects"
##  [4] "rank"          "fitted.values" "assign"
##  [7] "qr"            "df.residual"   "xlevels"
## [10] "call"          "terms"         "model"
```

Extracting info

A common pattern for R models:
store and explore

1. Create model object
2. Run function on model object

```
summary(mod)
predict(mod) # predictions at original x values
resid(mod) # residuals
```

```
summary(mod)
# Call:
# lm(formula = tc2009 ~ low, data = crime)
#
# Residuals:
#      Min       1Q   Median       3Q      Max
# -1134.36  -647.13    98.03   533.62  1344.30
#
# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  4256.86     233.44  18.236  < 2e-16 ***
# low            21.65       5.33   4.061 0.000188 ***
# ---
# Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
#
# Residual standard error: 649.9 on 46 degrees of
freedom
# Multiple R-squared:  0.2639, Adjusted R-squared:
0.2479
# F-statistic: 16.49 on 1 and 46 DF,  p-value: 0.000188
```

```
predict(mod)
#        1        2        3        4        5        6        7
# 3672.386 2525.082 3390.972 3629.091 3282.735 2936.379 3564.150
#        8        9       10       11       12       13       14
# 3888.858 4213.567 3888.858 4516.629 2958.027 3477.561 3239.441
#       15       16       17       18       19       20       21
# 3390.972 3455.913 3910.506 3217.794 3390.972 3499.208 3152.852
#       22       23       24       25       26       27       28
# 3845.564 3390.972 2741.554 3239.441 3174.499 3239.441 3520.855
#       29       30       31       32       33       34       35
# 3174.499 3131.205 3520.855 2958.027 3412.619 3672.386 3087.910
#       36       37       38       39       40       41       42
# 3347.677 3715.680 3845.564 3001.321 3564.150 3758.975 2763.201
#       43       44       45       46       47       48
# 3174.499 3607.444 3217.794 3455.913 3066.263 2828.143
```

```
resid(mod)
#           1           2           3           4           5
#  665.114163 1042.018405  334.228411  786.308663  -81.135339
#           6           7           8           9          10
#   88.120658 -917.849587  107.941665  240.132917  291.741665
#          11          12          13          14          15
# -573.828580 -702.426592  194.439412 -508.540840  270.428411
#          16          17          18          19          20
# -631.913338  538.694415 -692.793590  397.428411 -704.407838
#          21          22          23          24          25
#  207.848160 -202.963835  531.928411   86.345907  -55.540840
#          26          27          28          29          30
#  590.500910 -786.540840 -1134.355088 1344.300910 -818.204590
#          31          32          33          34          35
#  622.844912 -725.826592  282.481162  475.814163  160.589909
#          36          37          38          39          40
# -739.577089 -844.780336  716.636165 -975.121091  868.450413
#          41          42          43          44          45
#  749.625164  761.798657 -597.299090 -921.344087  866.106410
#          46          47          48
# -418.513338 -193.962841    5.056907
```

Interpreting models

Linear models are very easy to interpret

$$y = \alpha + \beta x + \epsilon$$

α is the expected value of y when x is 0.

β is the expected increase in y associated with a one unit increase in x

coef

```
coef(mod)
coefficients(mod)
# (Intercept)          low
#  4256.86158    21.64725
```

α β

coef

```
coef(mod)
coefficients(mod)
# (Intercept)          low
#  4256.86158    21.64725
```

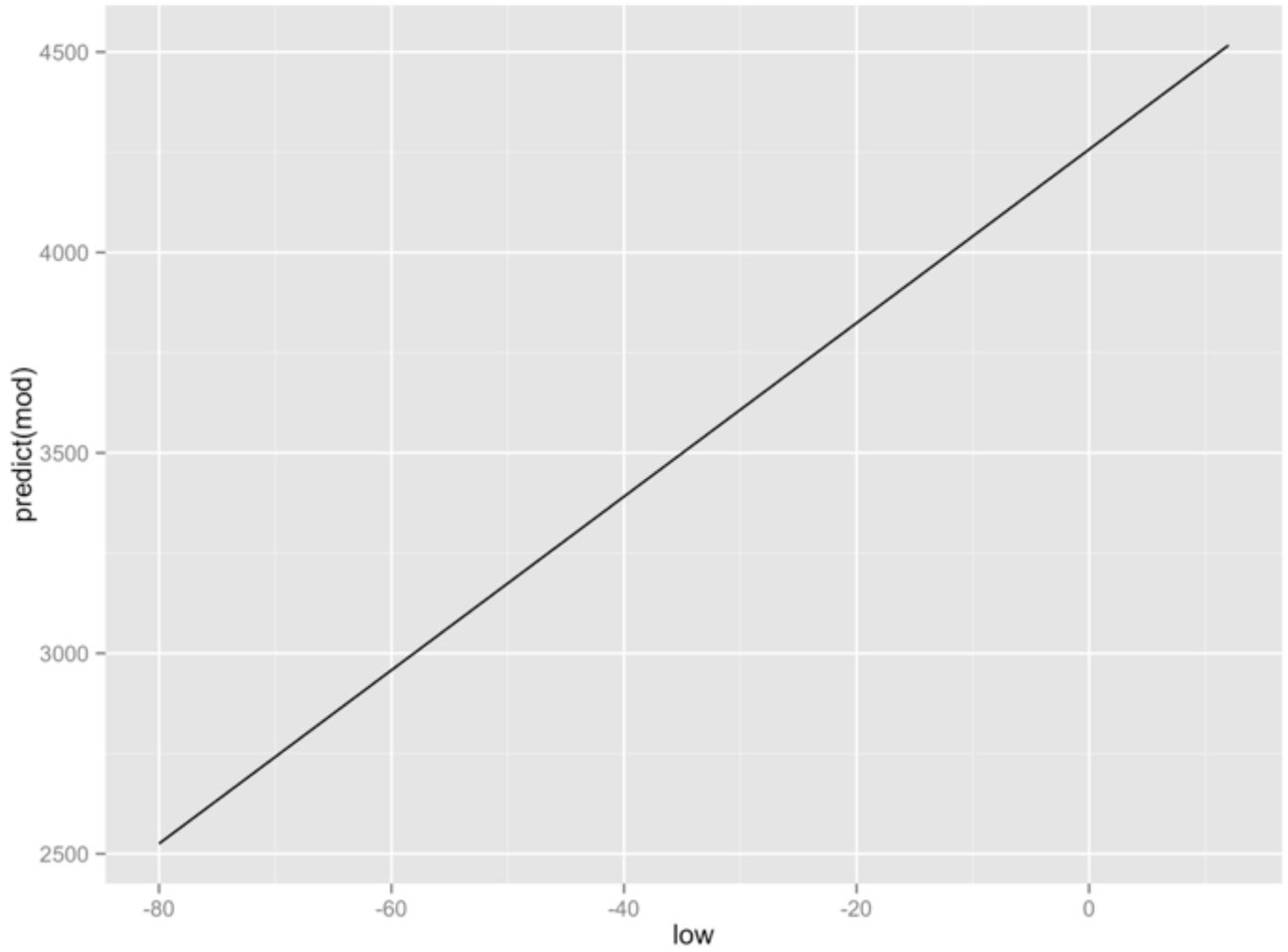
The best estimate of
tc2009 for a state with low = -10 is
4256.86 + **21.6** * (-10) = 4040.86

coef

```
coef(mod)
coefficients(mod)
# (Intercept)         low
#  4256.86158    21.64725
```

The best estimate of tc2009 for a state with low = -10 is **4256.86** + **21.6** * (-10) = 4040.86

A one unit change in low is associated with a **21.6** change in tc2009



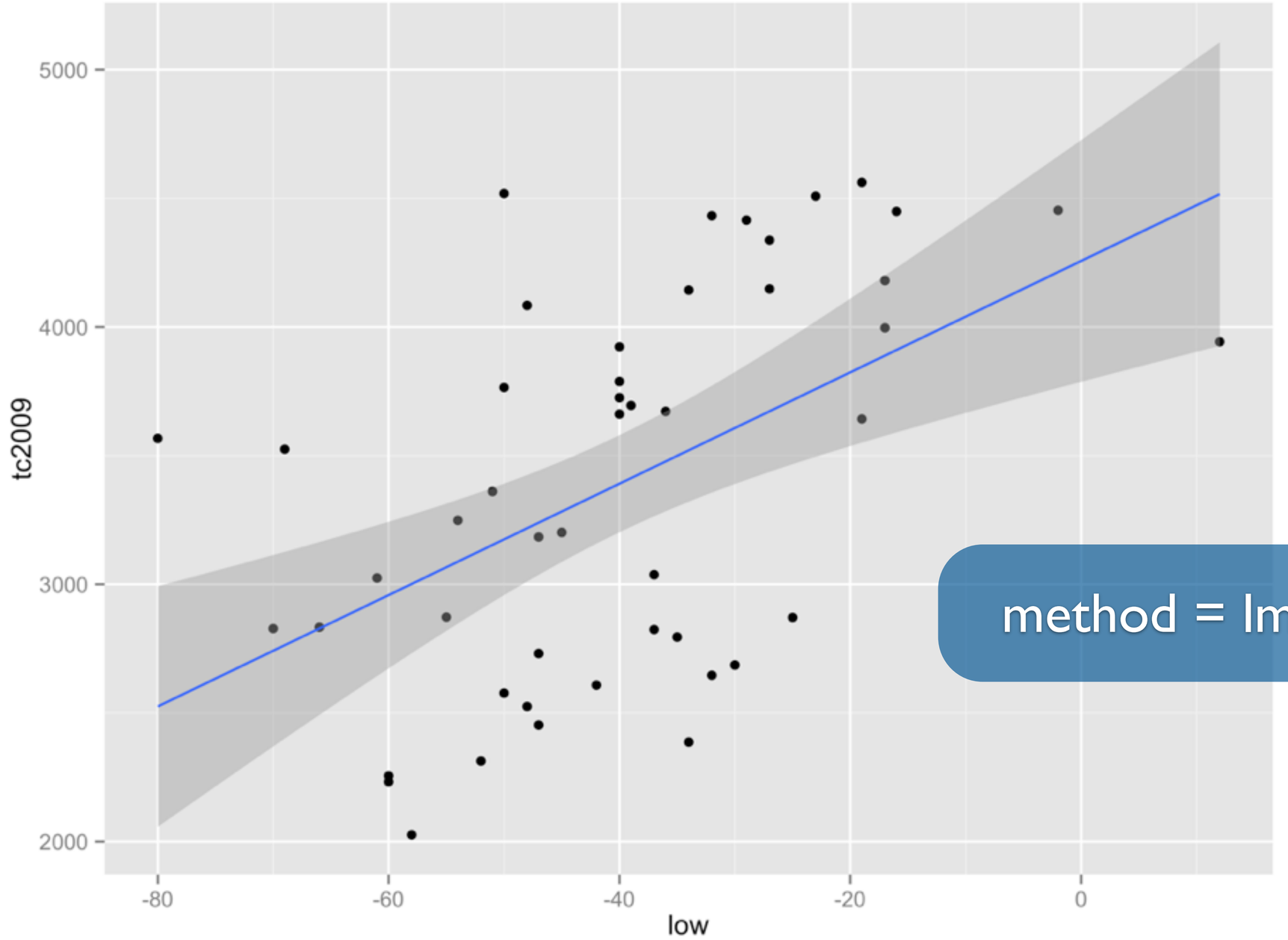

```
qplot(low, predict(mod), data = crime, geom = "line")
```

geom_smooth

Set the `method` argument to the name of the R function that creates the model you want.

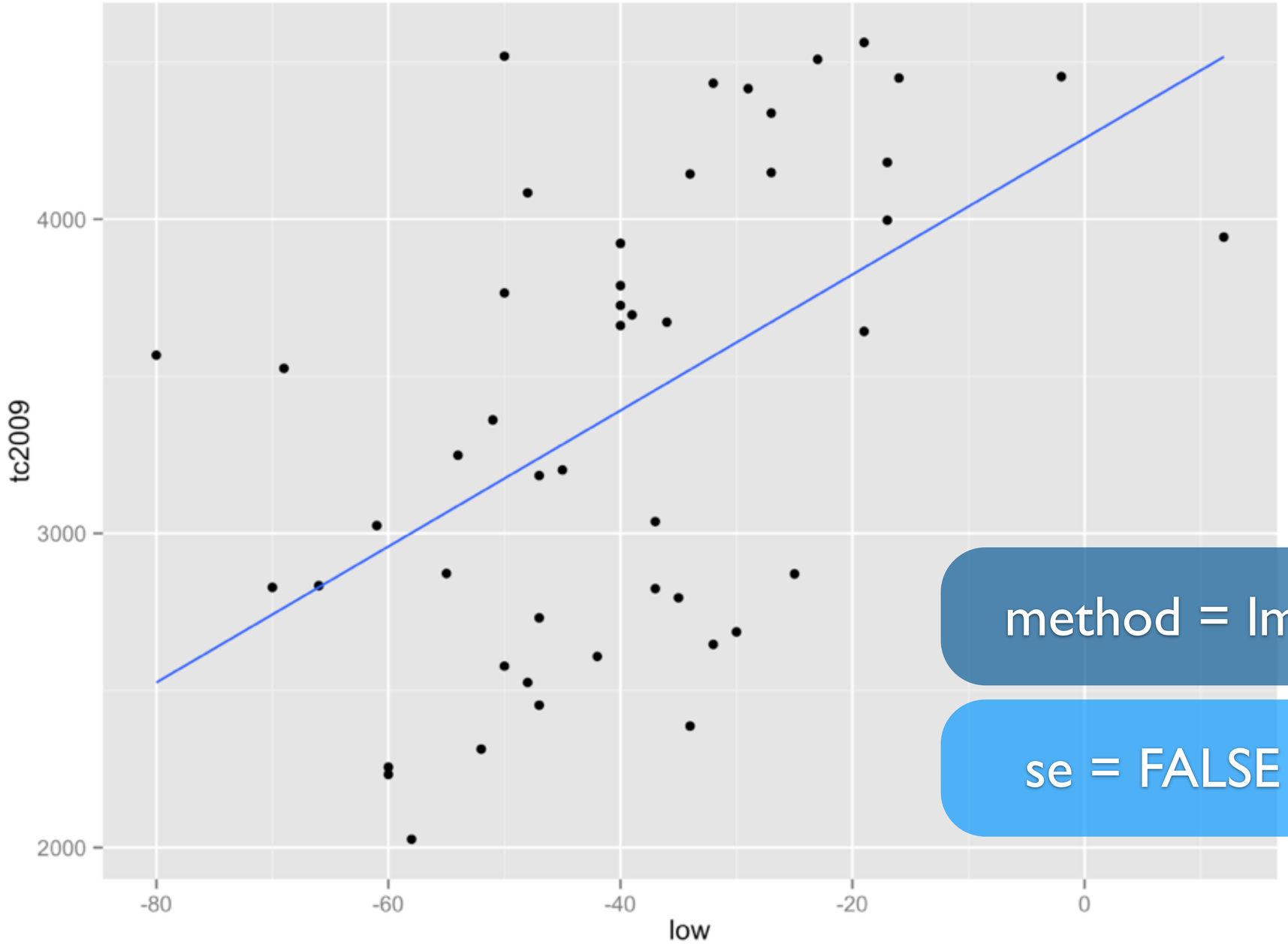
*`geom_smooth` *will assume you wish to regress y ~ x.*

```
qplot(low, tc2009, data = crime) +
  geom_smooth(method = lm)
```



method = lm

```
qplot(low, tc2009, data = crime) +
  geom_smooth(method = lm)
```



method = lm

se = FALSE

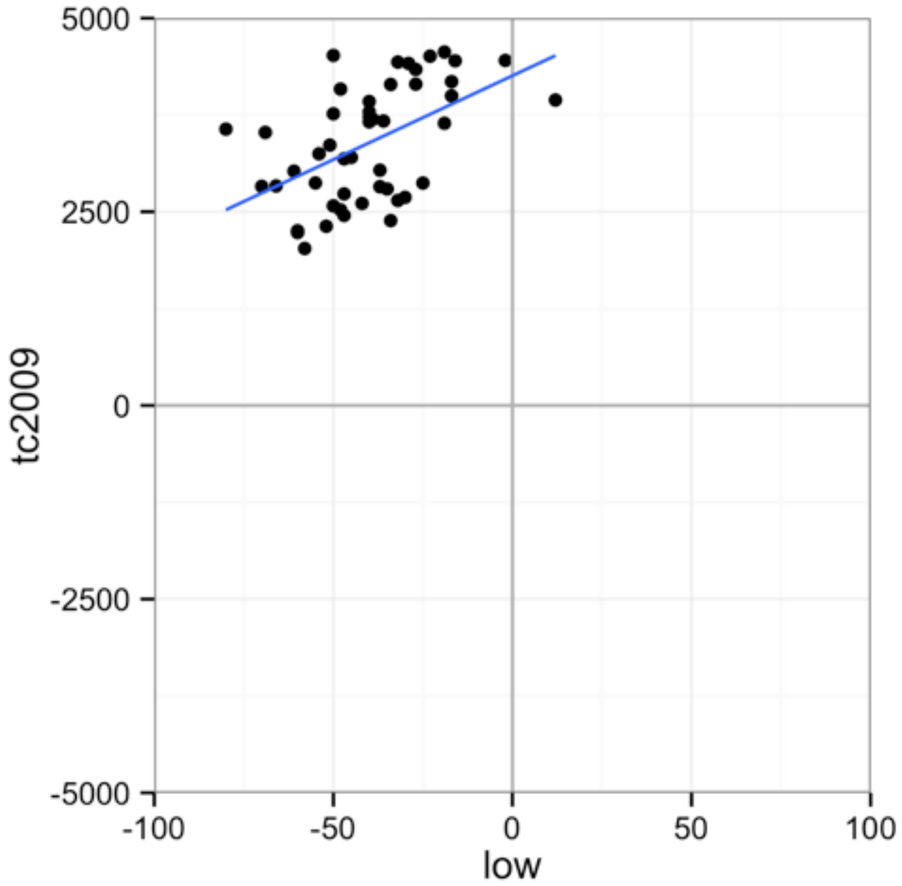
```
qplot(low, tc2009, data = crime) +
  geom_smooth(se = FALSE, method = lm)
```

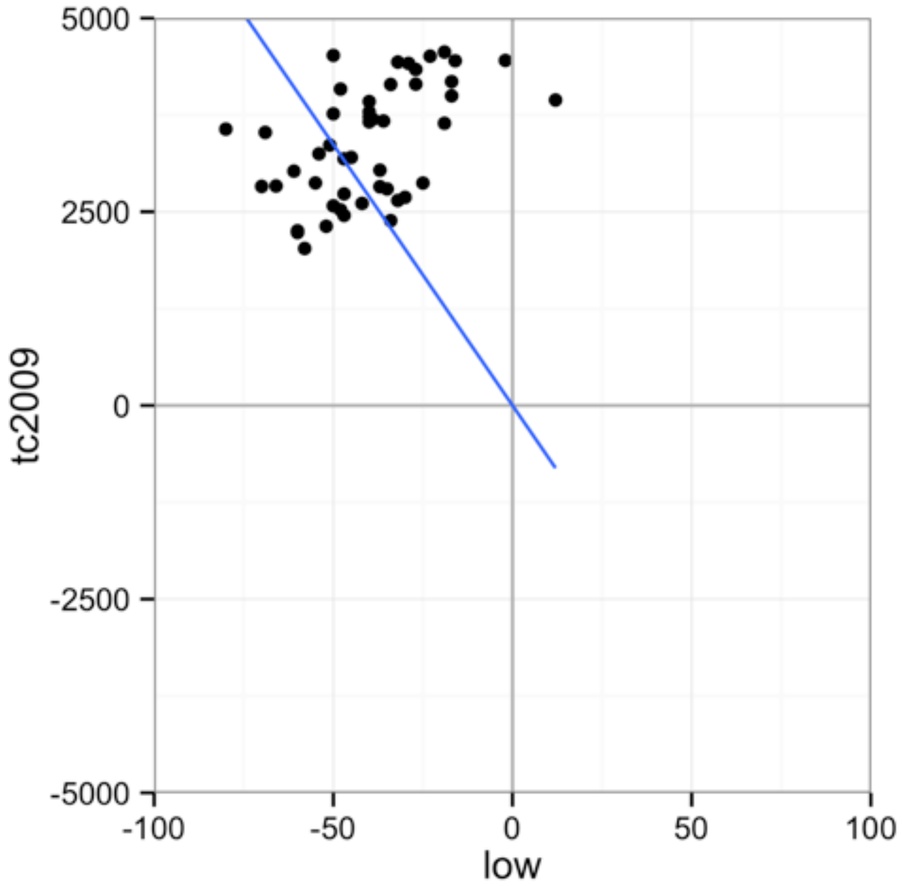
Aside: intercept terms

R includes an intercept term in each model by default

$$y = \alpha + \beta x + \epsilon$$

```
y ~ x
```





With α

Without α

Every linear model has a y intercept. Including α lets this term vary. Not including α forces the intercept to (0, 0).

An intercept term

You can explicitly ask for an intercept by including the number one, 1, as a formula term. You can remove the intercept by including a zero or negative 1.

```
# equivalent - includes intercept
lm(tc2009 ~ 1 + low, data = crime)
lm(tc2009 ~ low, data = crime)

# equivalent - removes intercept
lm(tc2009 ~ low - 1, data = crime)
lm(tc2009 ~ 0 + low, data = crime)
```

Your turn

Fit a linear model to the `wages` data set that predicts `earn` with `height`.

How do you interpret the relationship between height and earnings?

```
wages <- read.csv("data/wages.csv")
```

```
hmod <- lm(earn ~ height, data = wages)
```

$$y = \alpha + \beta x + \epsilon$$

```
hmod <- lm(earn ~ height, data = wages)
```

$$earn = \alpha + \beta \times height + \epsilon$$

```
hmod <- lm(earn ~ height, data = wages)
coef(hmod)
## (Intercept)      height
## -126523.359    2387.196
```

$$earn = \alpha + \beta \times height + \epsilon$$

```
hmod <- lm(earn ~ height, data = wages)
coef(hmod)
## (Intercept)      height
## -126523.359    2387.196
```

$$earn = -126523.36 + 2387.20 \times height + \epsilon$$

```
hmod <- lm(earn ~ height, data = wages)
coef(hmod)
## (Intercept)      height
## -126523.359    2387.196
```

$$earn = -126523.36 + 2387.20 \times height + \epsilon$$

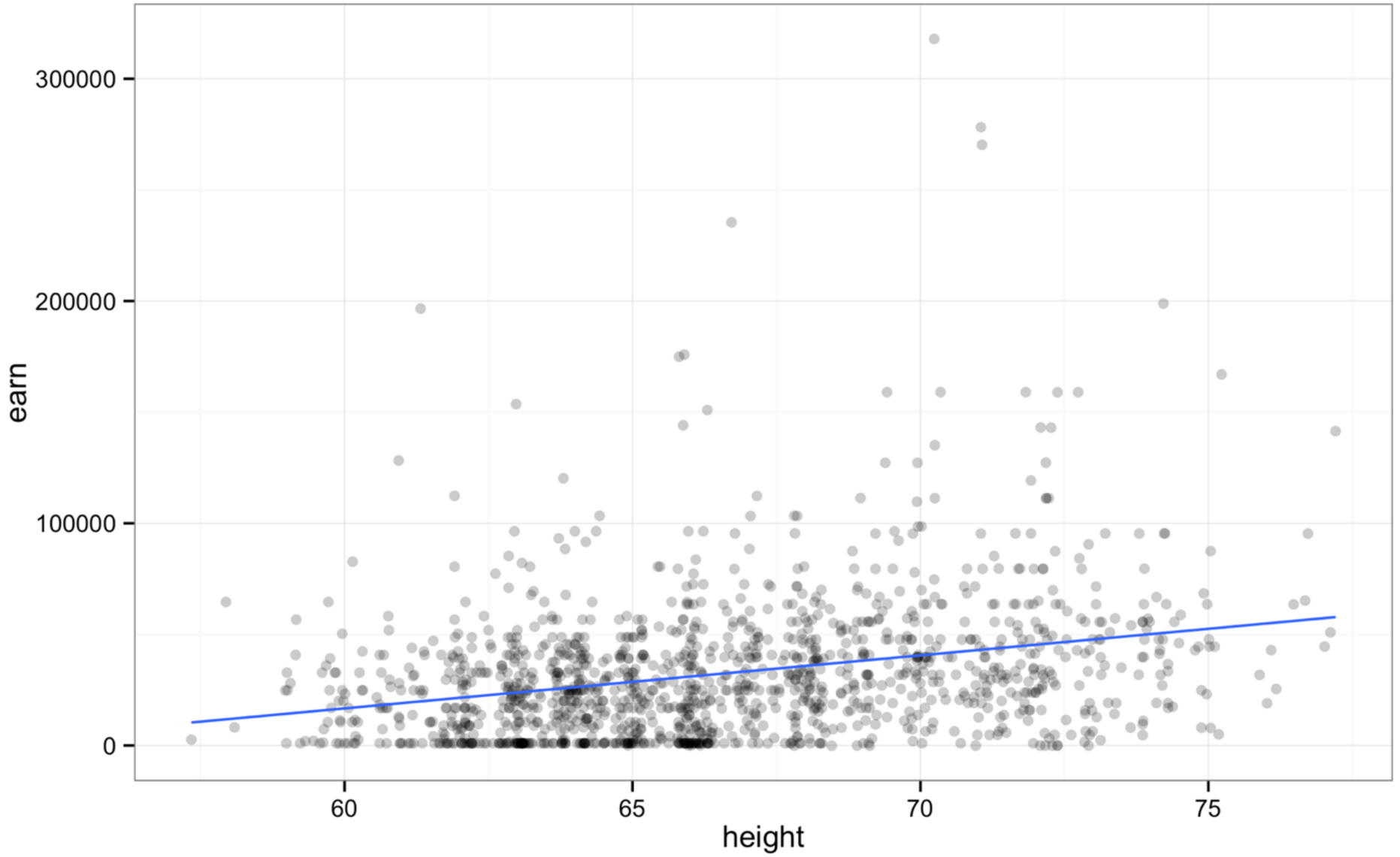
The best estimate of earn for someone 68 inches tall is

$$earn = -126523.36 + 2387.20 \times \quad 68 \quad + \epsilon$$

$$earn = 35806.24$$

Each 1 inch increase in height is associated with a $2387.20 increase in earnings.

$$earn = -126523.36 + \underbrace{2387.20}_{\beta} \times height + \epsilon$$




```
qplot(height, earn, data = wages, alpha = I(1/4)) +
  geom_smooth(se = FALSE, method = lm) + theme_bw()
```

Summary

Fitting models in R is a three step process

1. Describe the relationships in the data with a formula

```
earn ~ heights
```

2. Use a model function to fit the model

```
hmod <- lm(earn ~ heights, data = wages)
```

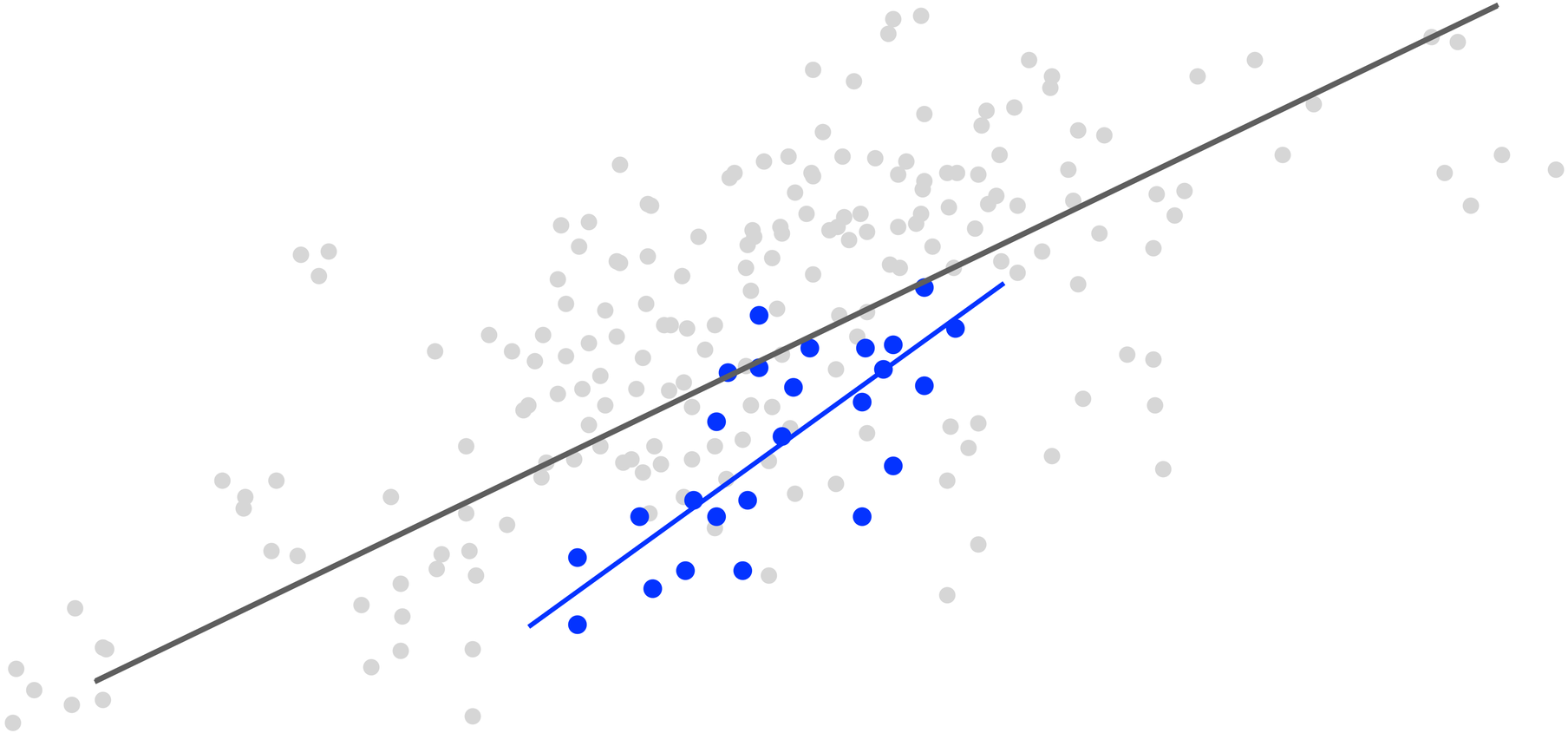
3. Examine the output in various ways

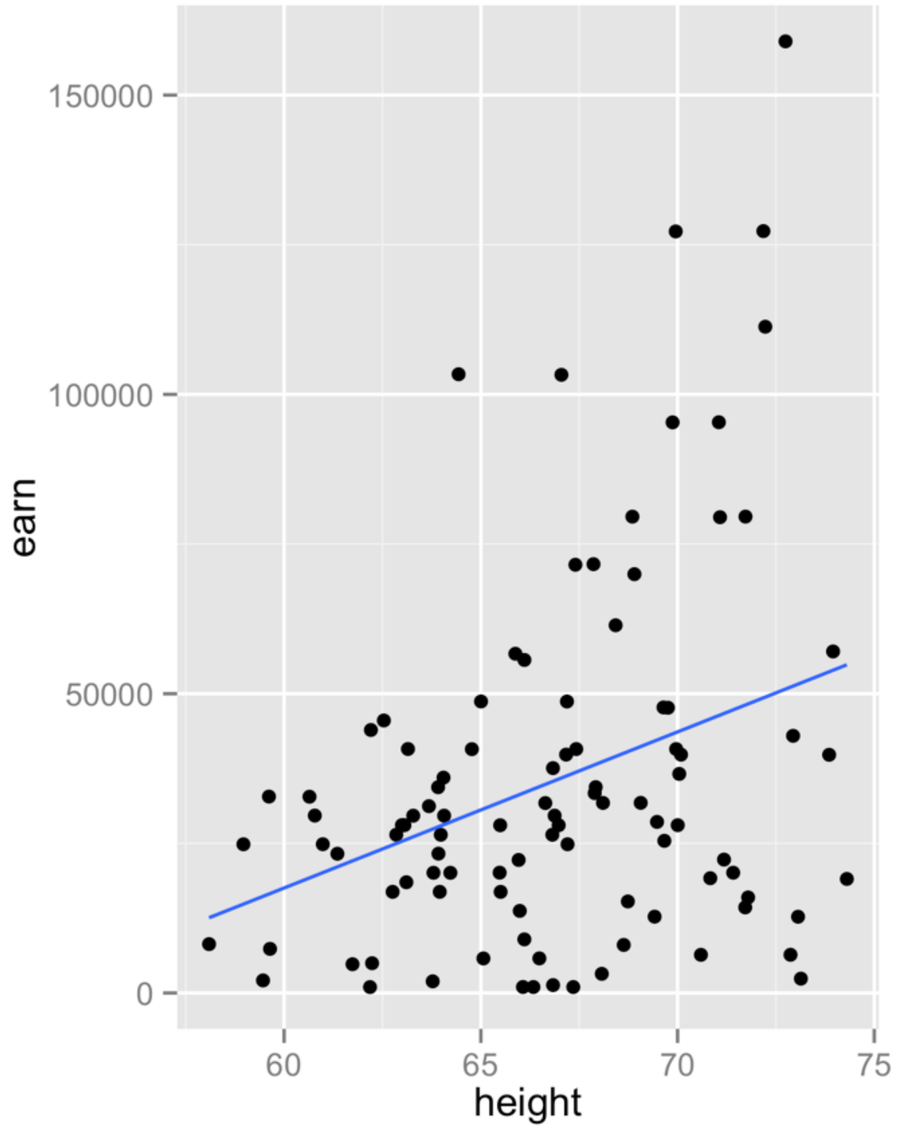
```
summary(hmod)
plot(hmod)
```

Model Inference

Deduce real relationships

(modeling)





w1 sample

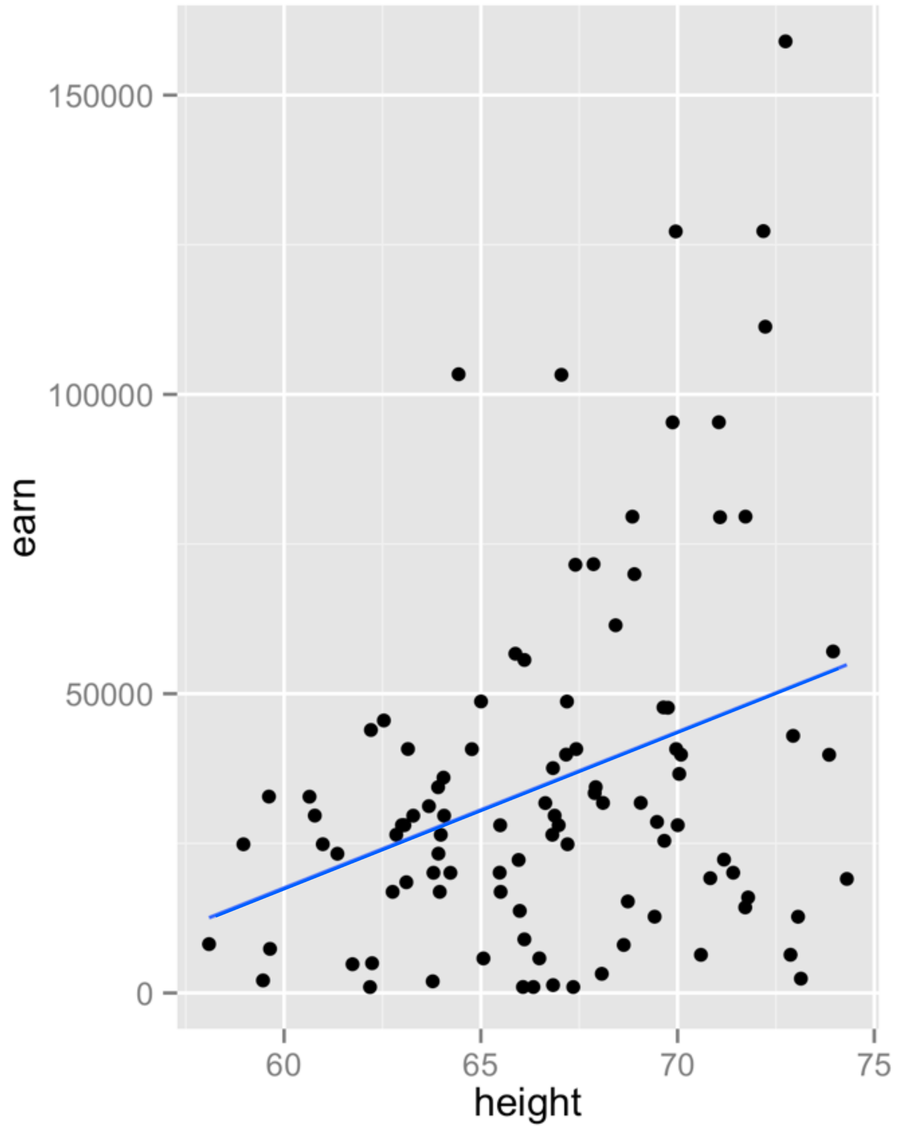
w1 is a sample of 100 points randomly selected from wages

```
w1 <- read.csv("data/w1.csv")
```

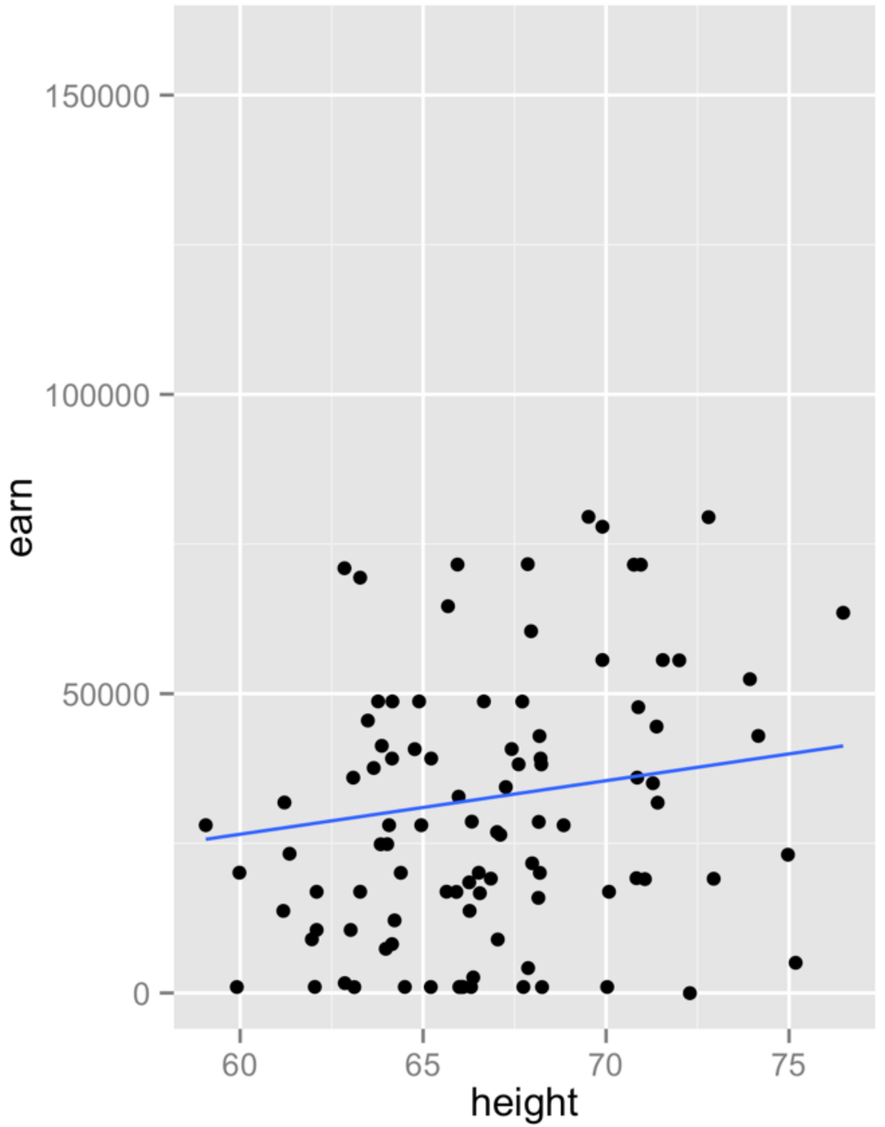
```
qplot(height, earn, data = w1)
+
geom_smooth(method = lm, se =
F)
```

Warm up

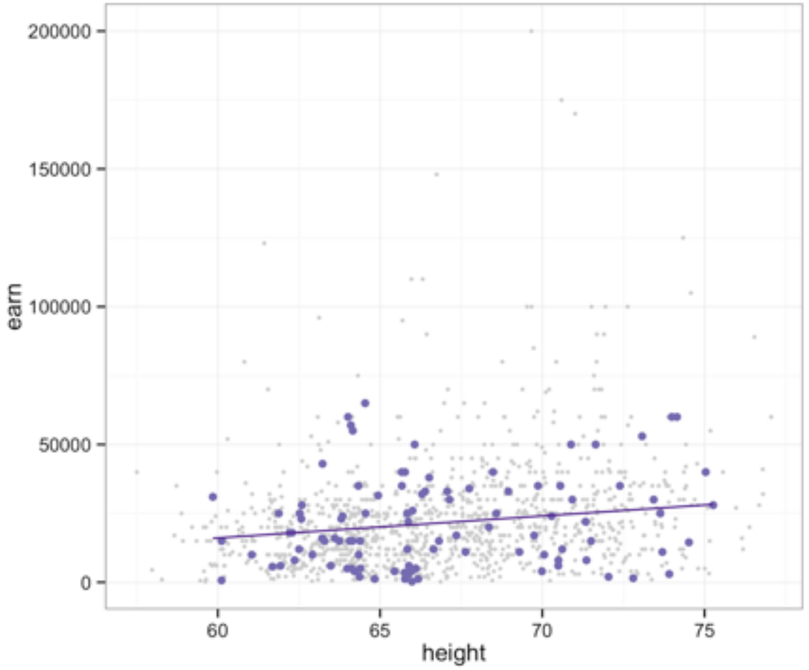
Would we get a slightly different model if we took a *second* random sample and ran a linear regression on it?

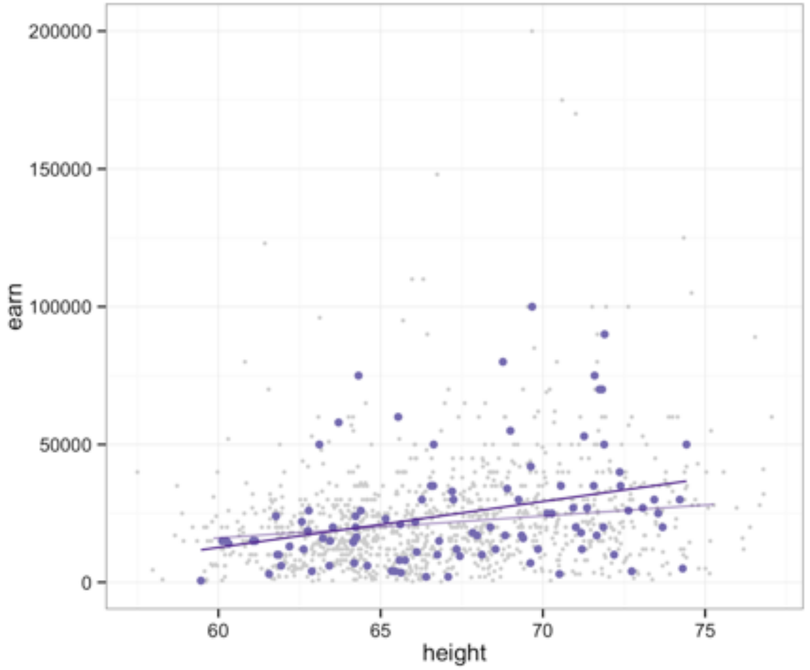


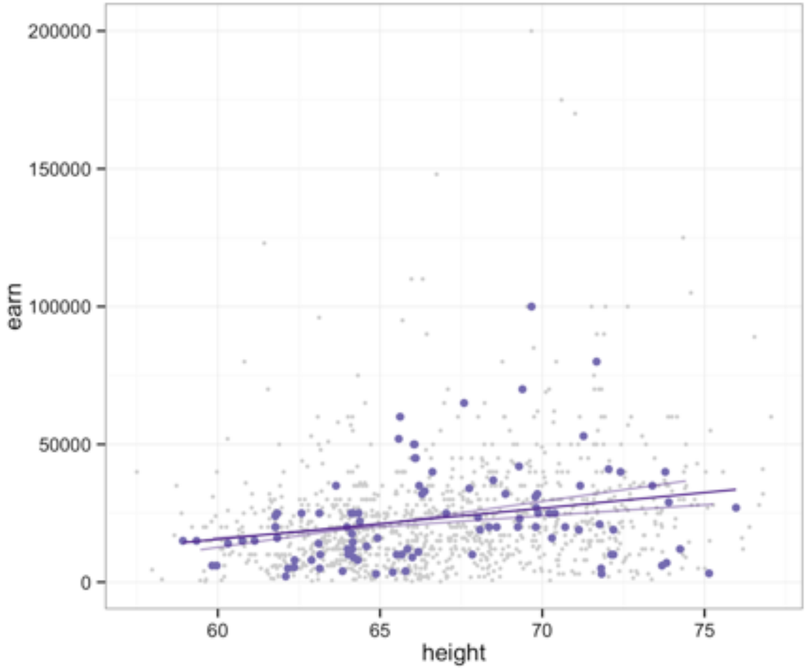
w1 sample

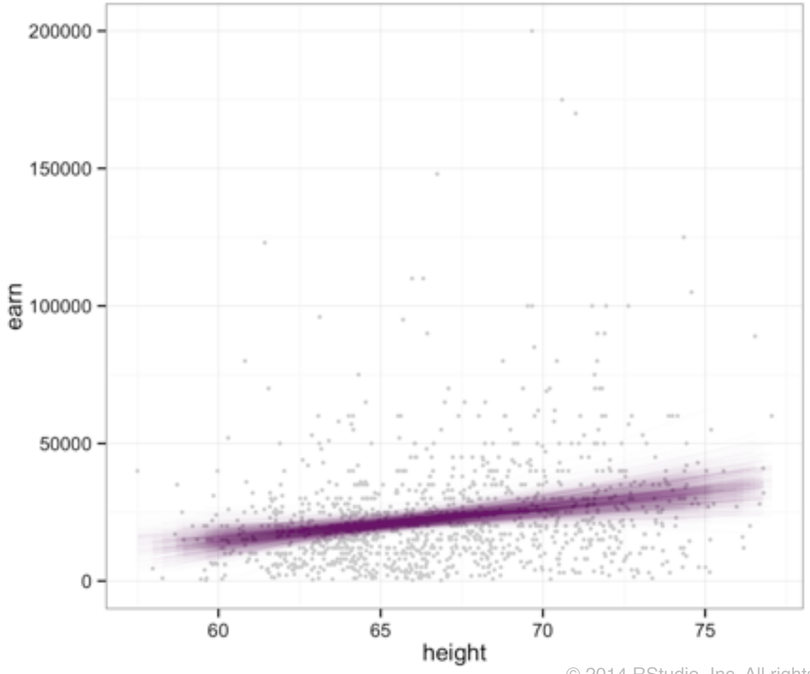


w2 sample









Uncertainty

Would $\beta = \beta$ if we took a new sample? Probably not.

Would $\beta = 0$ if we took a new sample? Maybe.

What interval captures all of the typical results?

Inference

You have two toolkits for reasoning about uncertain model parameters

- parametric statistics
- nonparametric statistics (bootstrapping)

Parametric statistics

Parametric reasoning

If a model meets a set of assumptions, it is easy to calculate the probability of observing a given β when the true β is 0

```
summary(hmod)
# Call:
# lm(formula = earn ~ height, data = wages)

# Residuals:
#    Min     1Q Median     3Q    Max
# -47903 -19744  -5184  11642 276796

# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  -126523      14076  -8.989   <2e-16 ***
# height          2387        211  11.312   <2e-16 ***
# ---
# Signif. codes:
# 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# Residual standard error: 29910 on 1377 degrees of freedom
# Multiple R-squared:  0.08503, Adjusted R-squared:  0.08437
# F-statistic:   128 on 1 and 1377 DF,  p-value: < 2.2e-16
```

```
summary(hmod)
# Call:
# lm(formula = earn ~ height, data = wages)

# Residuals:
#    Min     1Q Median     3Q    Max
# -47903 -19744  -5184  11642 276796

# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  -126523      14076  -8.989   <2e-16 ***
# height         2387        211  11.312   <2e-16 ***
# ---
# Signif. codes:
# 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# Residual standard error: 29910 on 1377 degrees of freedom
# Multiple R-squared:  0.08503, Adjusted R-squared:  0.08437
# F-statistic:   128 on 1 and 1377 DF,  p-value: < 2.2e-16
```

```
summary(hmod)
# Call:
# lm(formula = earn ~ height, data = wages)

# Residuals:
#    Min     1Q Median     3Q    Max
# -47903 -19744  -5184  11642 276796

# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  -126523      14076  -8.989   <2e-16 ***
# height         2387        211  11.312   <2e-16 ***
# ---
# Signif. codes:
# 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# Residual standard error: 29910 on 1377 degrees of freedom
# Multiple R-squared:  0.08503, Adjusted R-squared:  0.08437
# F-statistic:   128 on 1 and 1377 DF,  p-value: < 2.2e-16
```

```
summary(hmod)
# Call:
# lm(formula = earn ~ height, data = wages)

# Residuals:
#    Min     1Q Median     3Q
# -47903 -19744  -5184  11642 2

# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  -126523      14076  -8.989   <2e-16 ***
# height          2387        211  11.312   <2e-16 ***
# ---
# Signif. codes:
# 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# Residual standard error: 29910 on 1377 degrees of freedom
# Multiple R-squared:  0.08503, Adjusted R-squared:  0.08437
# F-statistic:   128 on 1 and 1377 DF,  p-value: < 2.2e-16
```

probability of β_{height} >= 2387 if the true β_{height} = 0 *

α level

If a p-value is very low (< 0.05), it suggests that either

1. you have an unusual sample
2. the true β does not equal 0
3. your model assumptions are wrong

α level

If a p-value is very low (< 0.05), it suggests that either

1. you have an unusual sample
2. the true β does not equal 0
3. your model assumptions are wrong

The cut-off for low p-values is the α level

```
summary(hmod)
# Call:
# lm(formula = earn ~ height, data = wages)

# Residuals:
#    Min     1Q Median     3Q    Max
# -47903 -19744  -5184  11642 276796

# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  -126523      14076  -8.989   <2e-16 ***
# height          2387        211  11.312   <2e-16 ***
# ---
# Signif. codes:
# 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# Residual standard error: 29910 on 1377 degrees of freedom
# Multiple R-squared:  0.08503, Adjusted R-squared:  0.08437
# F-statistic:   128 on 1 and 1377 DF,  p-value: < 2.2e-16
```

confidence intervals

Knowing probabilities also lets us calculate confidence intervals for β

```
confint(hmod, level = 0.95)
#                    2.5 %     97.5 %
# (Intercept) -154135.798 -98910.920
# height         1973.228   2801.163
```

We are 95% confident that the true coefficients are in these intervals

Your turn

Examine your crime model of `tc2009` and `low` in the `crime` data set.

Is the low statistically significant?

How do you explain this result?

```
mod <- lm(tc2009 ~ low, data = crime)
```

```
summary(mod)

# Call:
# lm(formula = tc2009 ~ low, data = crime)

# Residuals:
#      Min       1Q   Median       3Q      Max
# -1134.36  -647.13    98.03   533.62  1344.30

# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  4256.86     233.44  18.236  < 2e-16 ***
# low            21.65       5.33   4.061 0.000188 ***
# ---
# Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# Residual standard error: 649.9 on 46 degrees of freedom
# Multiple R-squared:  0.2639, Adjusted R-squared:  0.2479
# F-statistic: 16.49 on 1 and 46 DF,  p-value: 0.000188
```

Unusual sample

If a p-value is very low (< 0.05), it suggests that either

1. you have an unusual sample
2. the true β does not equal 0
3. your model assumptions are wrong

Incorrect assumptions

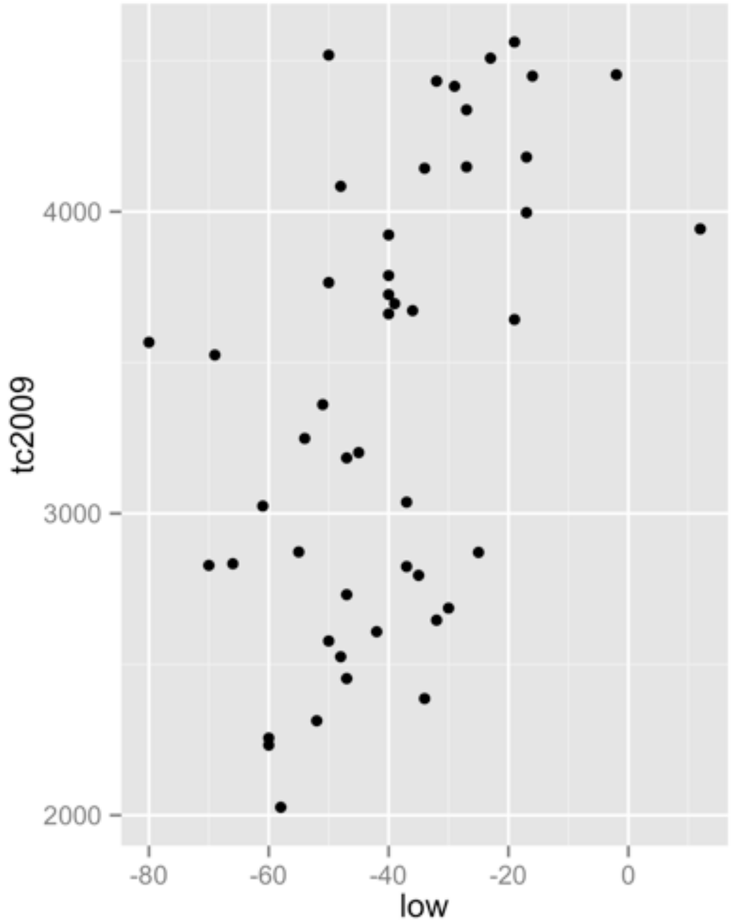
If a p-value is very low (< 0.05), it suggests that either

1. you have an unusual sample
2. the true β does not equal 0
3. your model assumptions are wrong

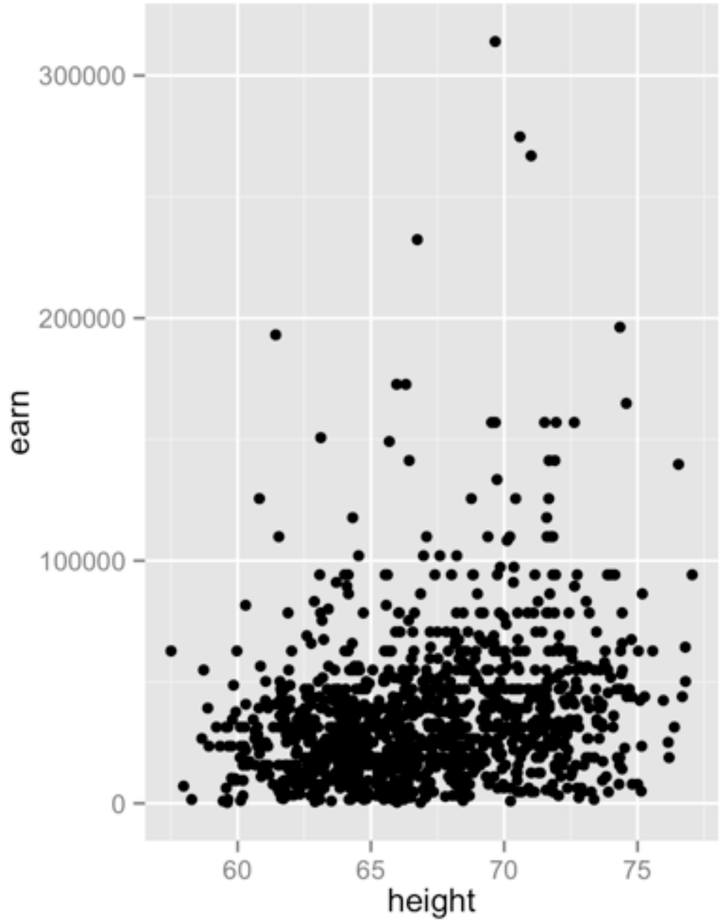
What do we assume and how can we check those assumptions?

a. y is linear in x (each x)

Plot y vs. x



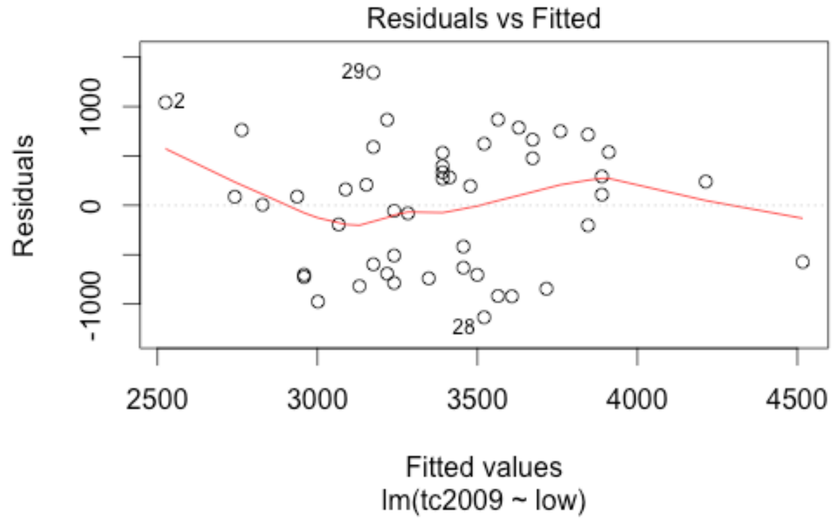

```
qplot(low, tc2009, data = crime)
```

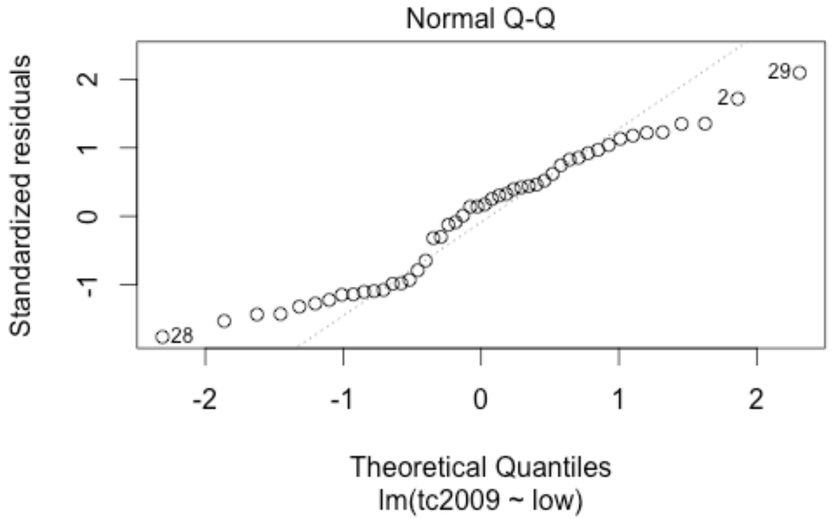


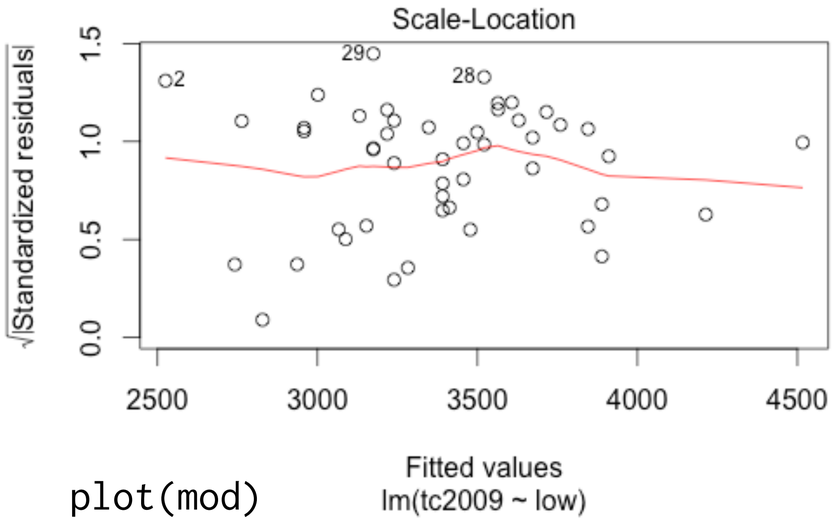

```
qplot(height, earn, data = wages)
```

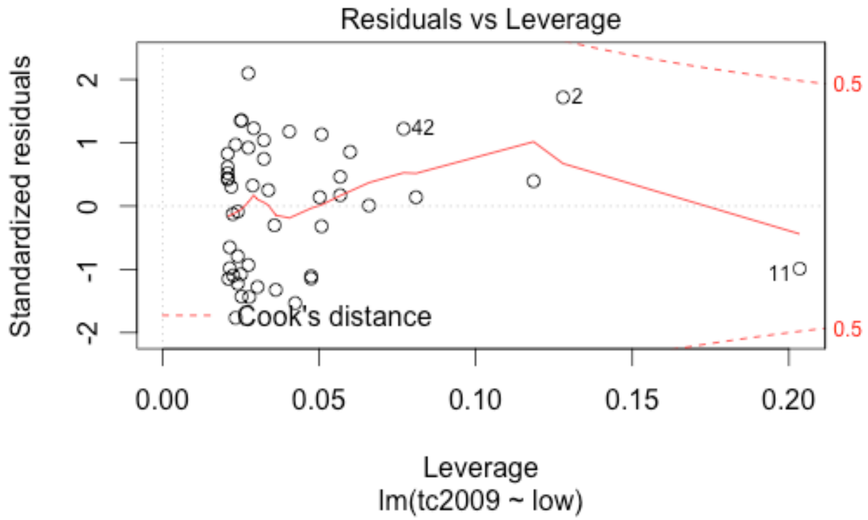
b. ε uncorrelated with y
c. ε ~ Normal
d. No high-leverage ε

plot(mod)









plot(mod)

Comparing multiple groups (categorical variables)

Your turn

Fit a linear model to the `wages` data set.

This time regress `earn` on `race`. How do you interpret the results?

```
rmod <- lm(earn ~ race, data = wages)

coef(rmod)
# (Intercept) racehispanic   raceother   racewhite
#   28372.09      -2886.79     3905.32     4993.33
```

race levels
black
hispanic
other
white

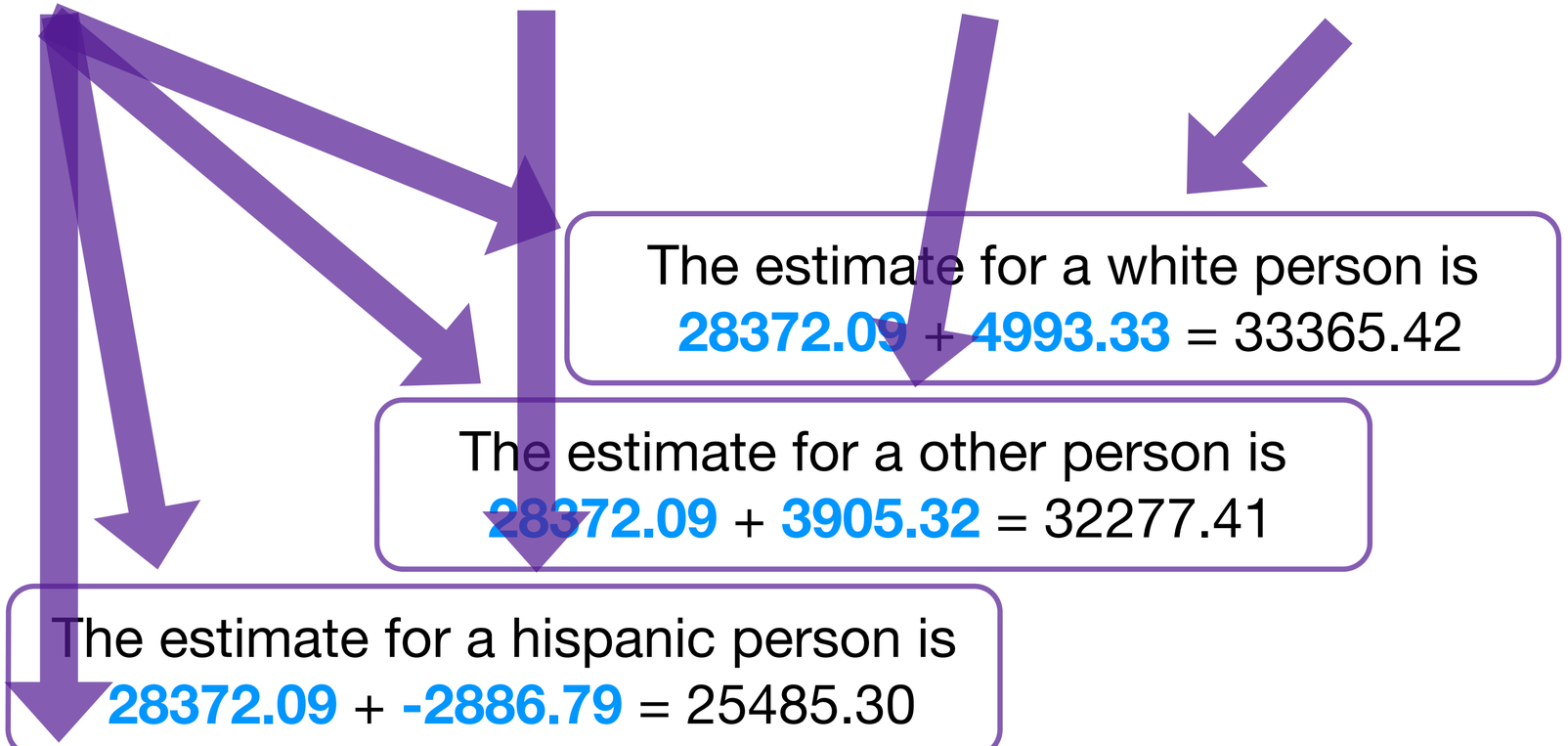
What is missing?

Categorical variables

One value of the variable is chosen as a baseline. Each remaining value gets its own coefficient.

Interpret coefficients as the effect of moving from the baseline value to the new value.

```
coef(rmod)
#  (Intercept) racehispanic   raceother   racewhite
#     28372.09     -2886.79     3905.32     4993.33
```



The estimate for a white person is
28372.09 + **4993.33** = 33365.42

The estimate for a other person is
28372.09 + **3905.32** = 32277.41

The estimate for a hispanic person is
28372.09 + **-2886.79** = 25485.30

The estimate for a black person is
28372.09 = 28372.09

factor

Hispanic may make a better baseline (since it has the lowest estimate).

You can change the order of your levels with `factor`.

```
wages$race <- factor(wages$race,
  levels = c("hispanic", "white", "black", "other"))
```

The new order

```
rmod2 <- lm(earn ~ race, data = wages)
coef(rmod2)
```

```
coef(rmod2)
# (Intercept)   racewhite   raceblack   raceother
#   25485.303    7880.121    2886.791    6792.111
```

The coefficients change, but the actual estimates are the same.

ANOVA

Linear regression with a categorical variable, is the equivalent of ANOVA (Analysis of Variance). To see the output as an ANOVA table, use `anova`

```
anova(rmod2)
# Analysis of Variance Table

# Response: earn
#             Df     Sum Sq    Mean Sq F value  Pr(>F)
# race         3 6.7924e+09 2264121503  2.3241 0.07328 .
# Residuals 1375 1.3395e+12  974196170
# ---
# Signif. codes:
# 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Multi-group tests

function	test	statistic
lm	ANOVA	mean
aov	ANOVA	mean
anova	ANOVA	mean
oneway.test	ANOVA with unequal variances	mean
pairwise.t.test	t tests between multiple groups	mean
kruskal.test	Kruskal Wallis Rank Sum	sum
friedman.test	Friedman Rank Sum	sum
fligner.test	Fligner-Killeen	variance
bartlett.test	Bartlett test	variance

Your turn

Model `earn` on `sex` with a linear model.

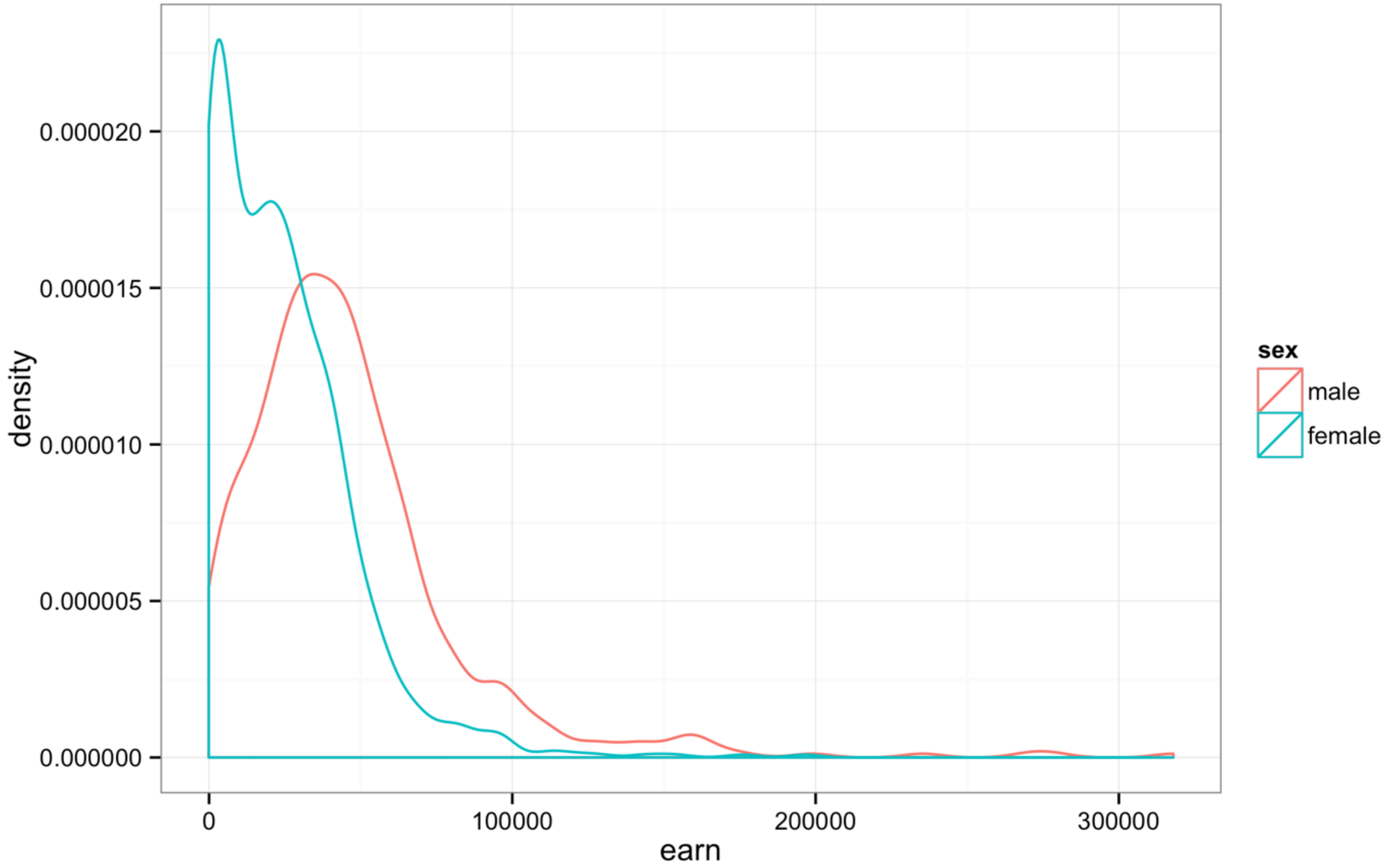
Does the output suggest there is a difference between male and female salaries (at a statistically significant level)?

Bonus: make male salaries the baseline.

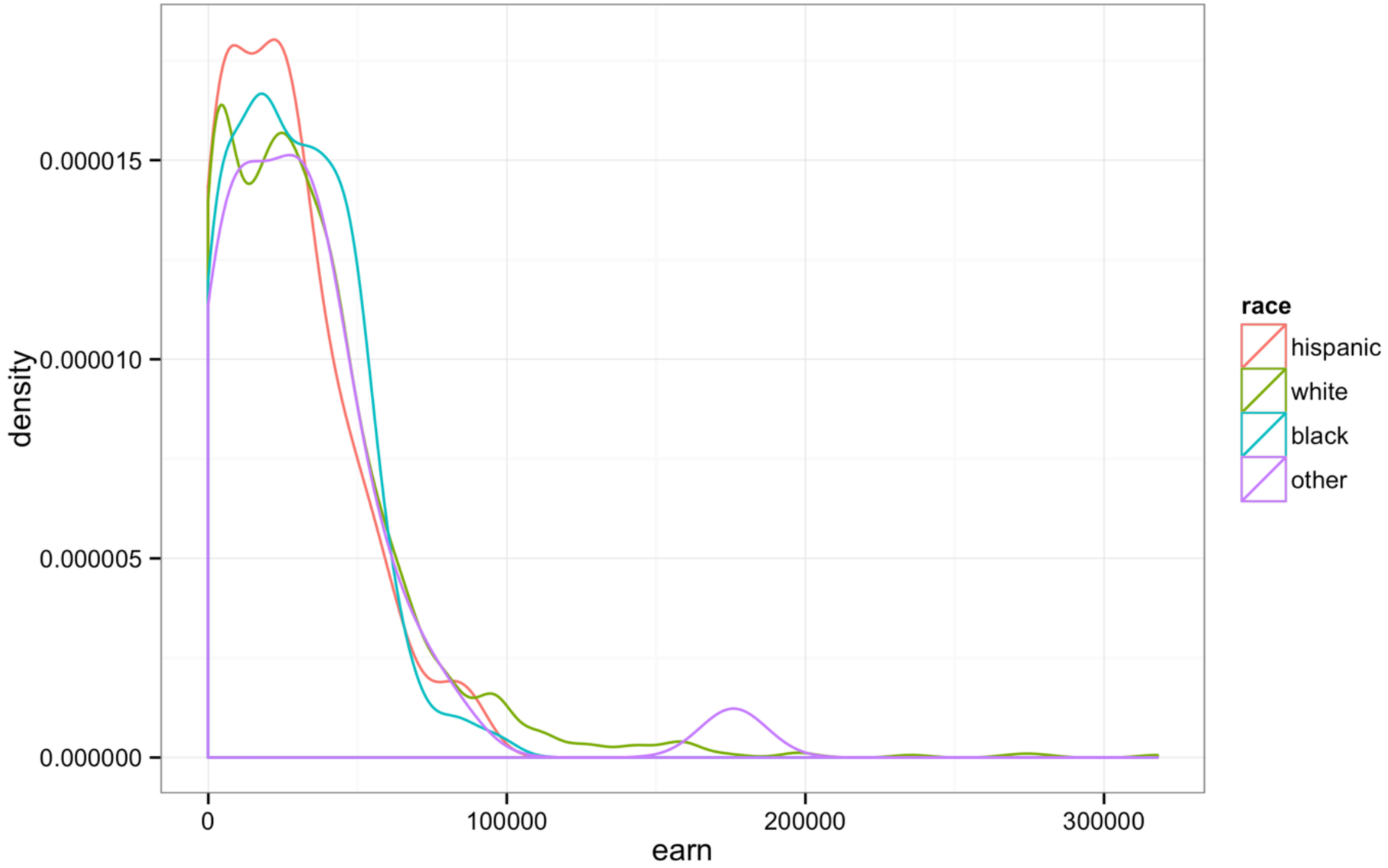
```
smod <- lm(earn ~ sex, data = wages)

coef(smod)
# (Intercept)     sexmale
#    24245.65    21747.48
```

```
wages$sex <- factor(wages$sex,
  levels = c("male", "female"))
smod <- lm(earn ~ sex, data = wages)

coef(smod)
# (Intercept)   sexfemale
#    45993.13   -21747.48
```

```
anova(smod)
Analysis of Variance Table

Response: earn
            Df     Sum Sq    Mean Sq F value    Pr(>F)
sex          1 1.5320e+11 1.5320e+11  176.81 < 2.2e-16 ***
Residuals 1377 1.1931e+12 8.6646e+08
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```




```
qplot(earn, data = wages, geom = "density",
  color = sex) + theme_bw()
```




```
qplot(earn, data = wages, geom = "density",
  color = race) + theme_bw()
```

Estimating multivariate functions

Adding variables

Use height to predict earn

```
m1 <- lm(earn ~ height, data = wages)
```

Adding variables

Use height **and sex** to predict earn

```
m2 <- lm(earn ~ height + sex, data = wages)
```

Adding variables

Use height **and sex** to predict earn

```
m2 <- lm(earn ~ height + sex, data = wages)
```

```
coef(m1)
# (Intercept)      height
# -126523.359    2387.196

coef(m2)
# (Intercept)      height   sexfemale
#  -15605.703     879.424  -16874.158
```

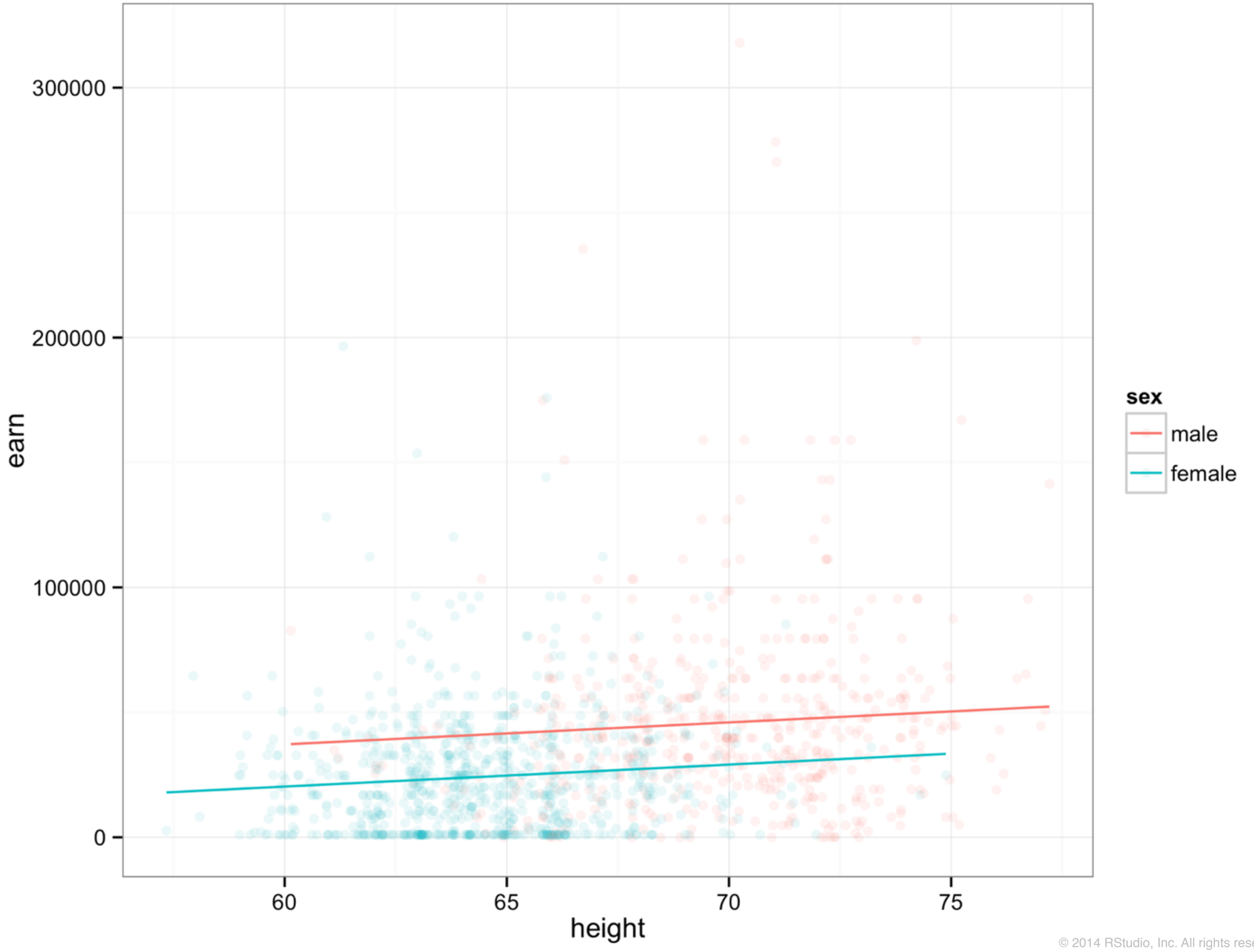
```
coef(m1)
# (Intercept)      height
# -126523.359    2387.196

coef(m2)
# (Intercept)      height   sexfemale
#  -15605.703     879.424  -16874.158
```

Effect of
being female
...when height is held constant

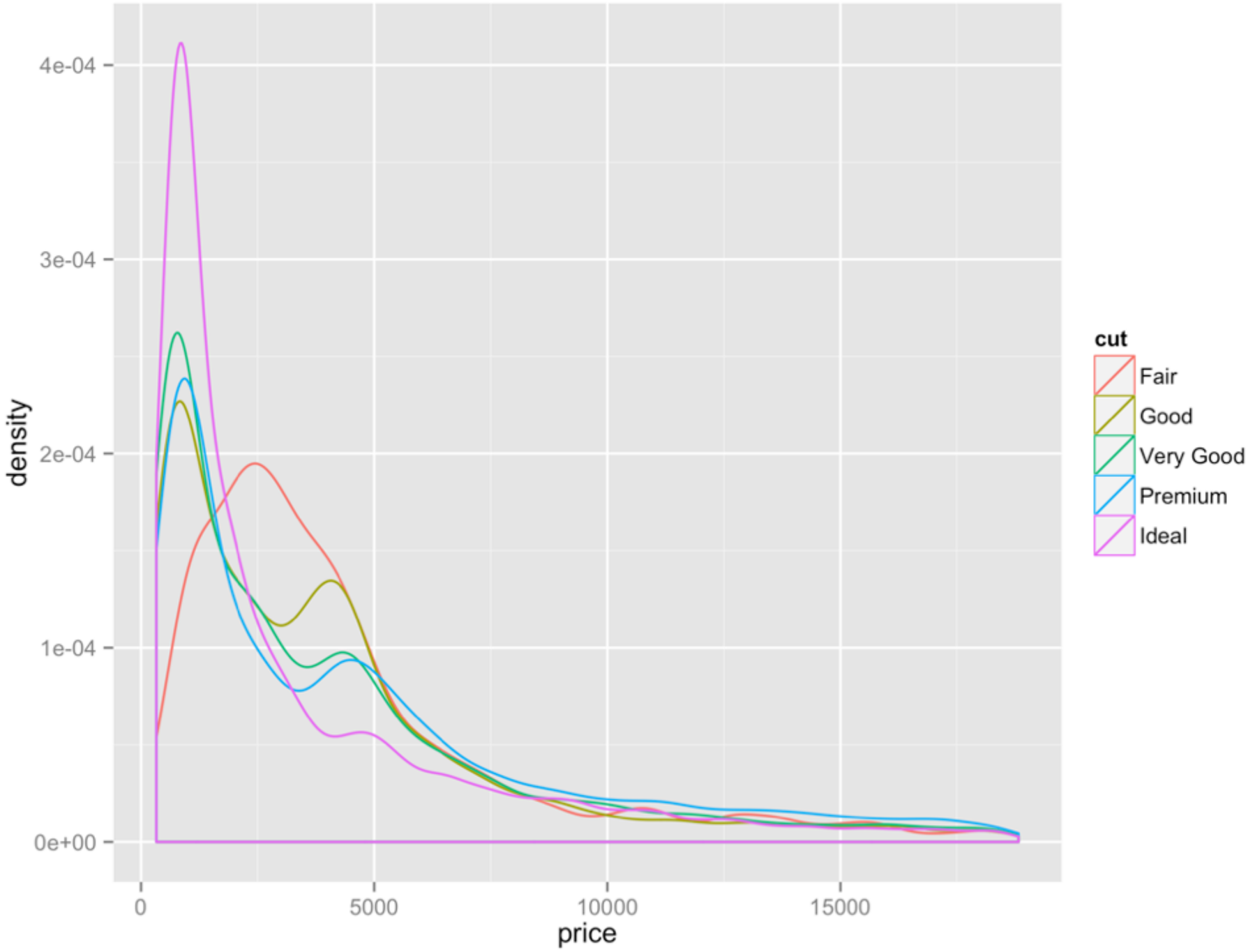
```
coef(m1)
# (Intercept)      height
# -126523.359    2387.196

coef(m2)
# (Intercept)      height   sexfemale
#  -15605.703     879.424  -16874.158
```

Effect of change in height **...when sex is held constant**

Effect of being female **...when height is held constant**

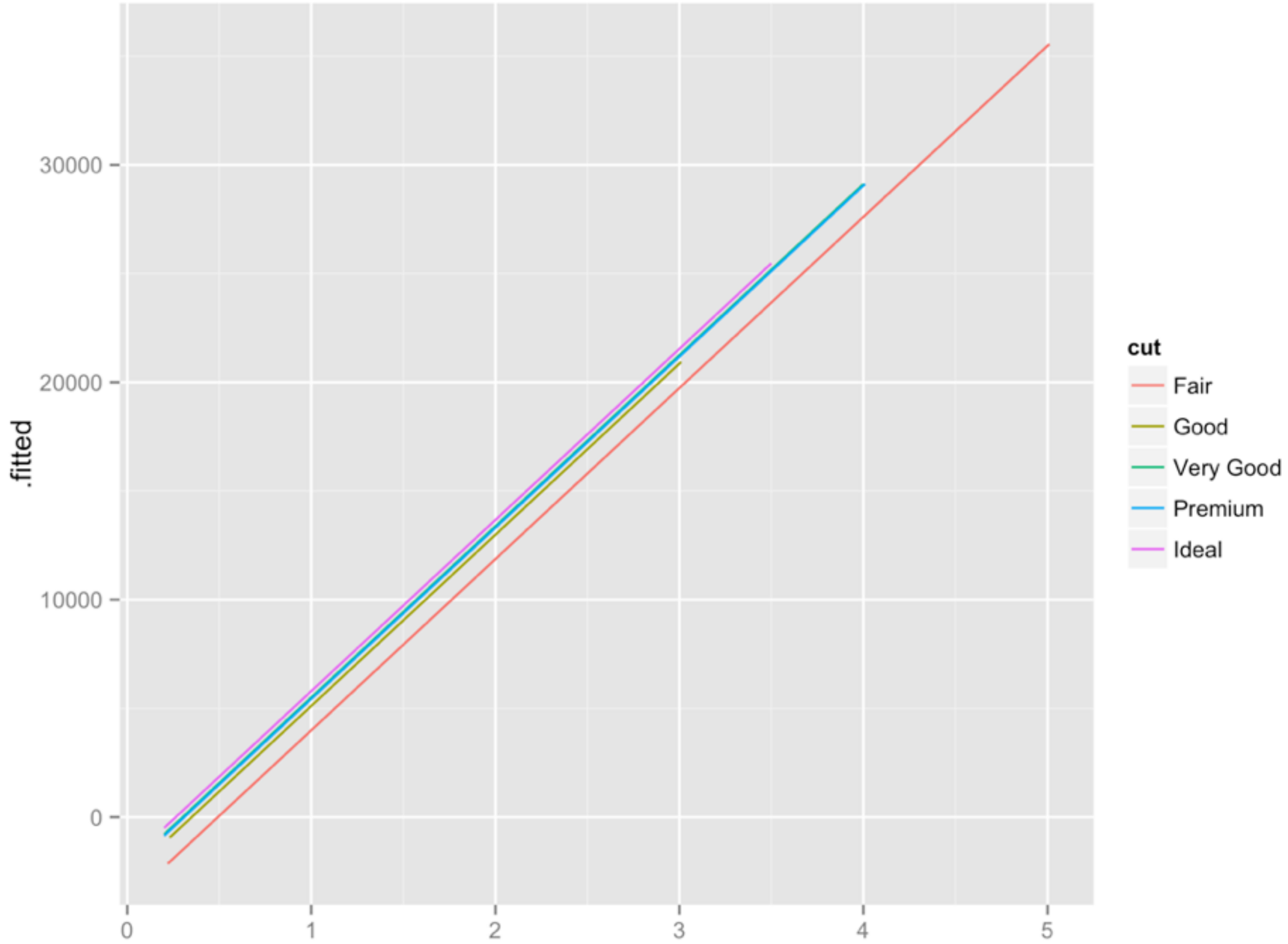


```
coef(m1)
# (Intercept)      height
# -126523.359    2387.196

coef(m2)
# (Intercept)      height   sexfemale
#  -15605.703     879.424  -16874.158
```



```
d1 <- lm(price ~ cut, data = diamonds)
coef(d1)
# (Intercept)      cutGood     cutIdeal   cutPremium
#   4358.7578    -429.8933    -901.2158     225.4999
# cutVery Good
#    -376.9979
```

```
d2 <- lm(price ~ cut + carat, data = diamonds)
coef(d2)
# (Intercept)      cutGood     cutIdeal   cutPremium
#   -3875.470     1120.332     1800.924     1439.077
# cutVery Good        carat
#    1510.135     7871.082
```




```
qplot(carat, predict(d2), data = diamonds, color = cut,
  geom = "line")
```

Simpson's paradox

A relationship between two variables may change when you consider a third, related variable.

Simpson's paradox

A relationship between two variables may change when you consider a third, related variable.

Solution

Model every source of variation within a system simultaneously.

Your turn

Create a new model that predicts `earn` with `height`, `sex`, `race`, `ed`, and `age`.

Does the relationship between `height` and `earn` change? `sex` and `earn`?

```
m3 <- lm(earn ~ height + sex + race + ed + age,
  data = wages)

coef(m3)
# (Intercept)      height   sexfemale   racewhite
# -74354.1693    632.7391 -17552.5441   4592.7865

#   raceblack   raceother          ed         age
#   2086.0663   1708.3196   4382.0853    287.4744
```

.

. is shorthand for "everything else." So these create the same model

```
lm(earn ~ height + sex + race + ed + age, data = wages)
lm(earn ~ ., data = wages)
```

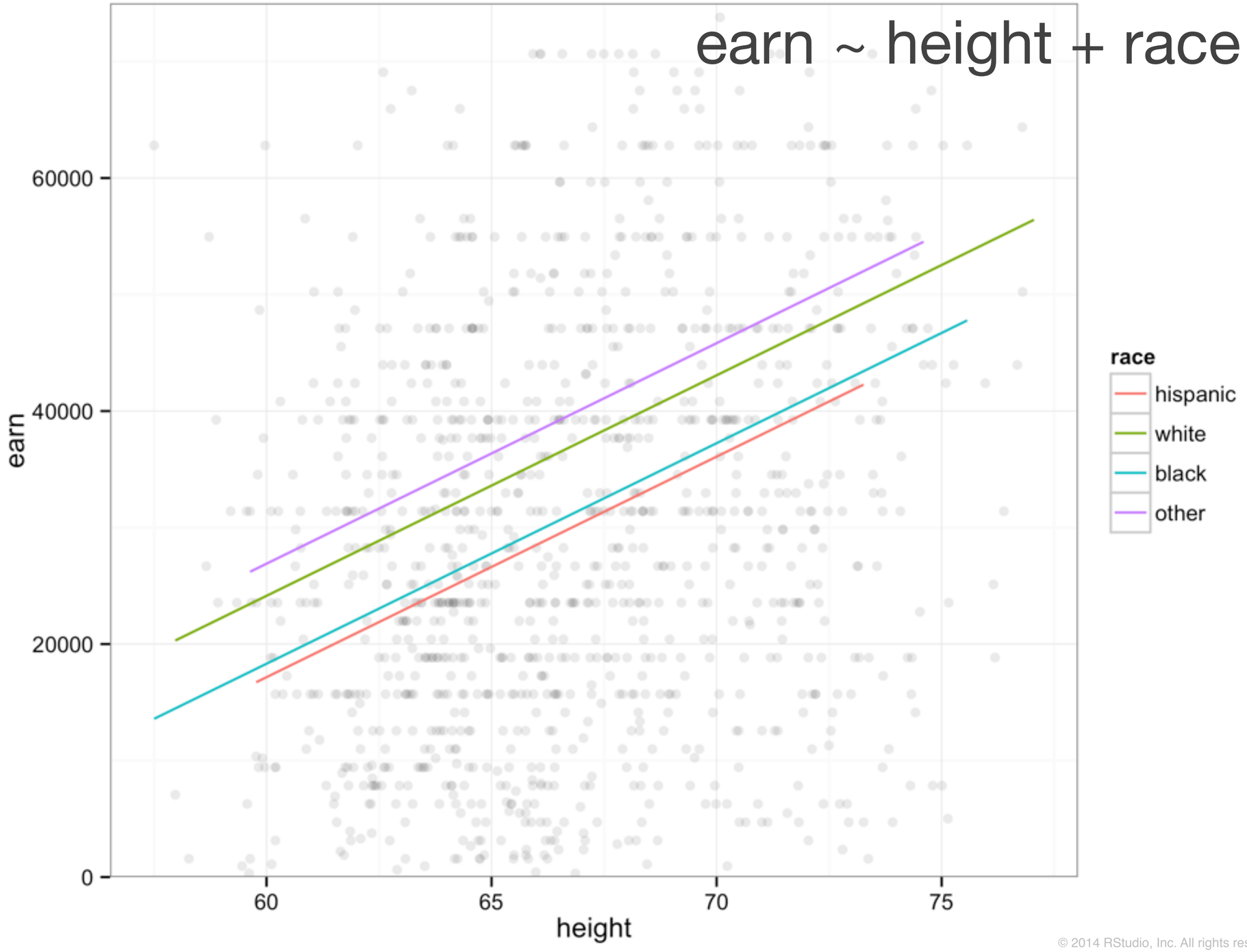
.

You may also modify . by removing individual variables.

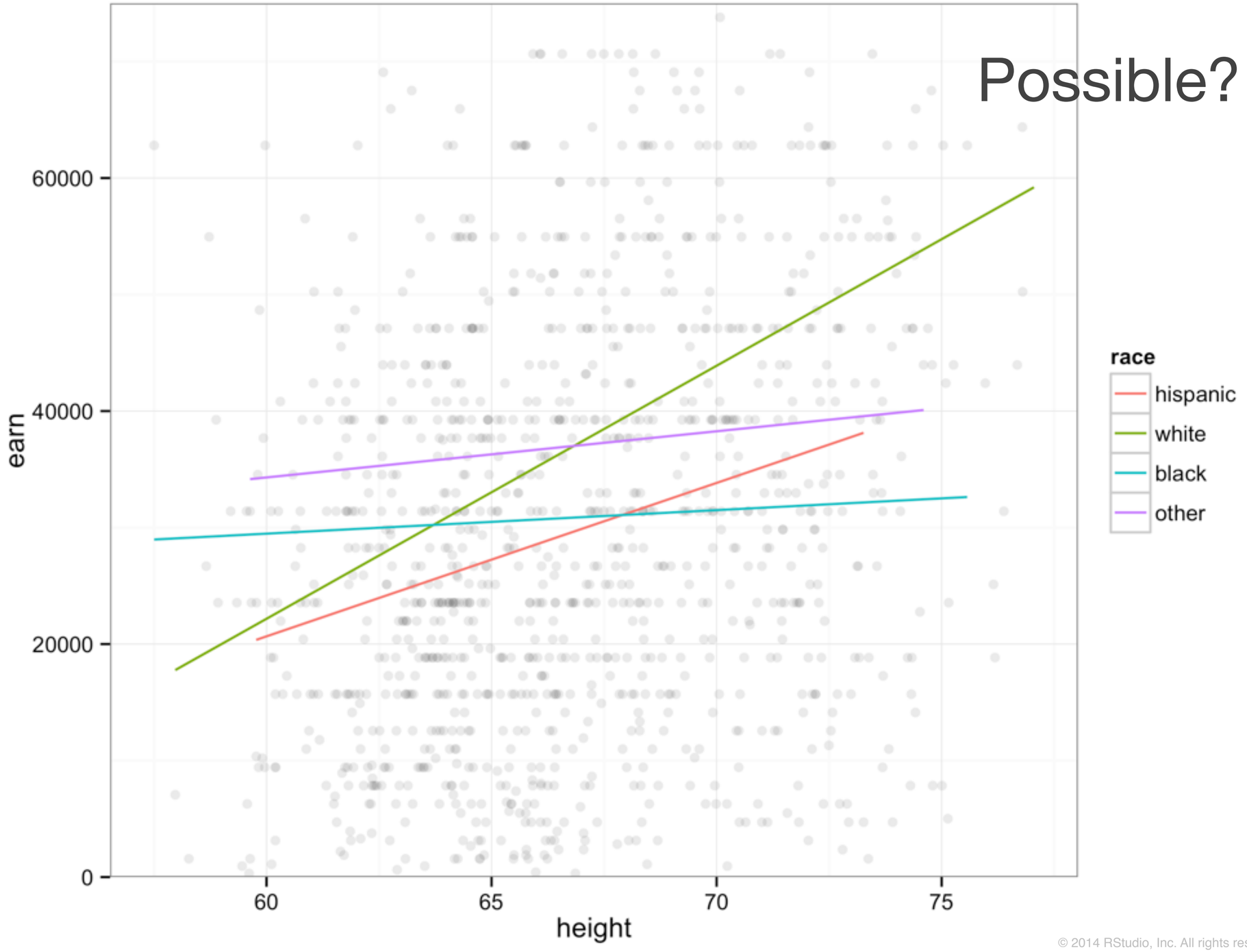
```
lm(earn ~ height + sex + race + ed, data = wages)
lm(earn ~ . - age, data = wages)
```

Interaction terms

earn ~ height + race



Possible?



Interaction effects

Use height and sex **and the interaction effect of height and sex** to predict earn

```
m4 <- lm(earn ~ height + sex + height:sex,
    data = wages)
```

```
coef(m4)
# (Intercept)      height   sexfemale  height:sexfemale
#   -42677.40     1265.92    30510.43           -701.41
```

```
coef(m4)
# (Intercept)    height  sexfemale  height:sexfemale
#   -42677.40   1265.92   30510.43           -701.41
```

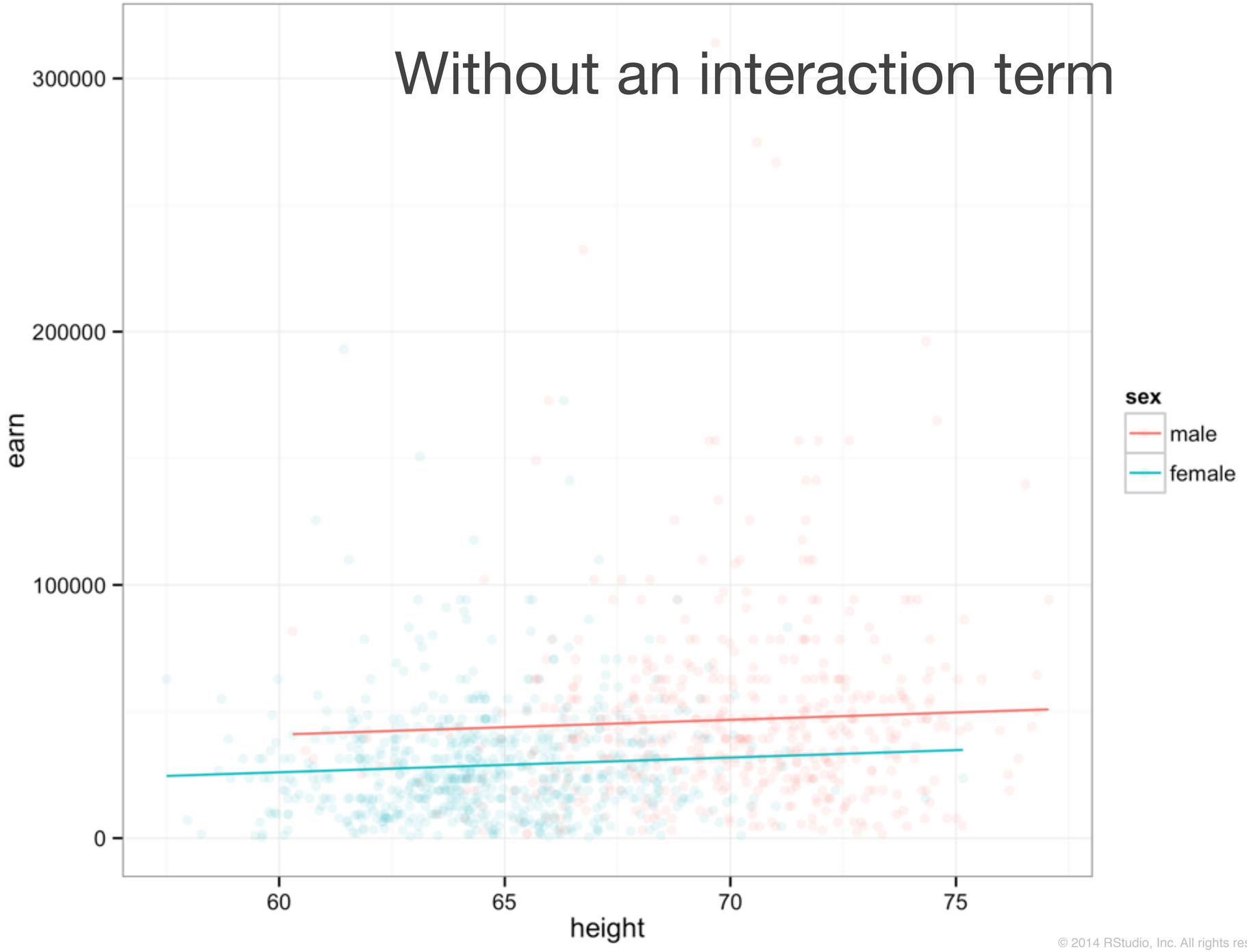
For men, a 1" increase in height is associated with a gain in earnings of **1265.92**

```
coef(m4)
# (Intercept)    height  sexfemale  height:sexfemale
#   -42677.40   1265.92   30510.43           -701.41
```

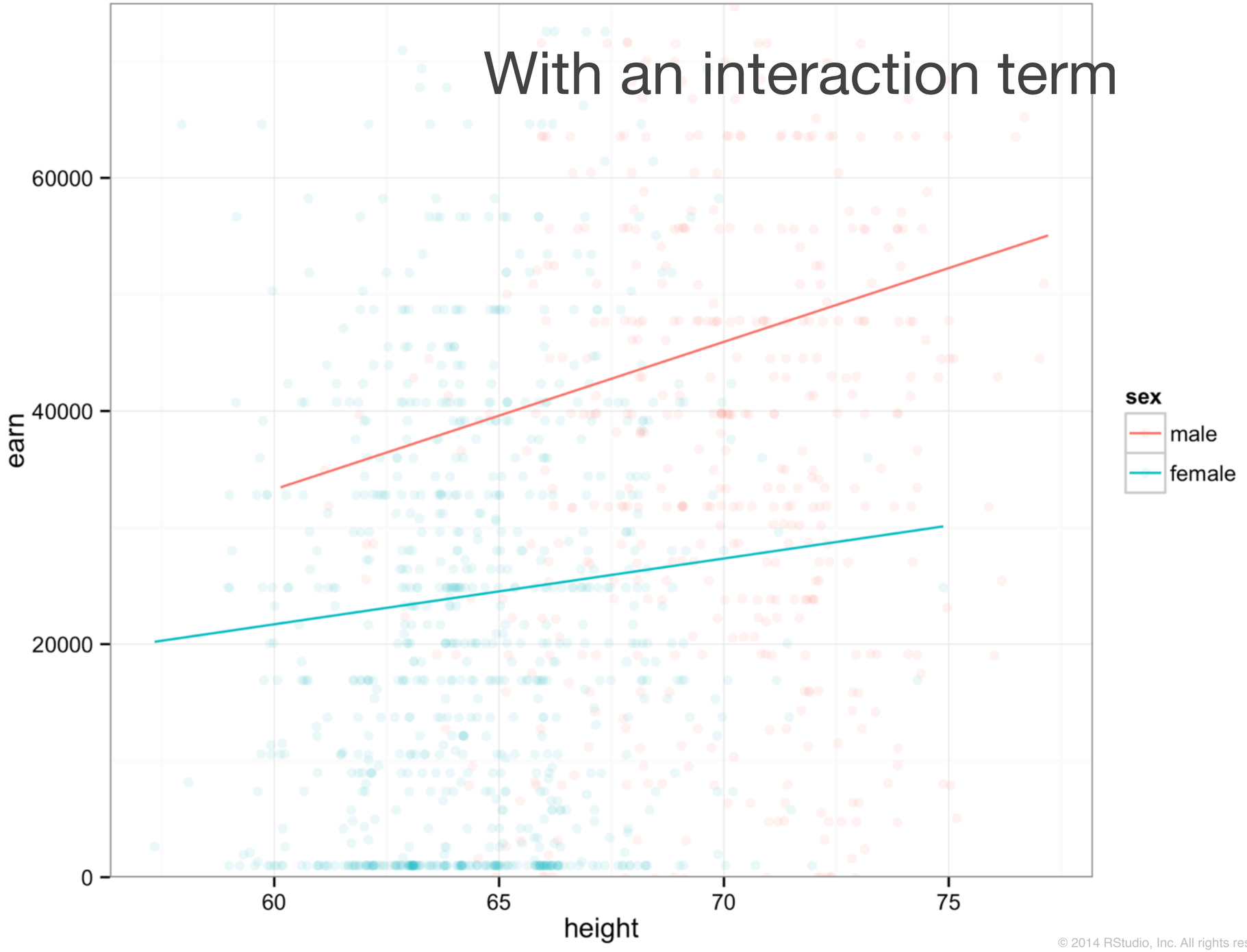
For men, a 1" increase in height is associated with a gain in earnings of **1265.92**

For women, a 1" increase in height is associated with a gain in earnings of **1265.92** + (**-701.41**) = 564.51

Without an interaction term

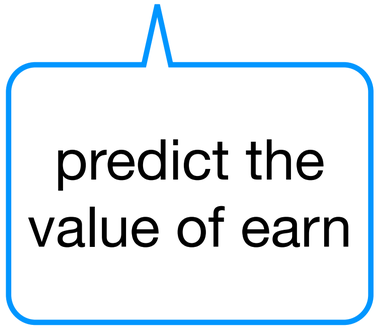


With an interaction term



Syntax

```
lm(earn ~ height, data = wages)
lm(earn ~ height + sex, data = wages)
lm(earn ~ height + sex + height:sex, data = wages)
```



Syntax

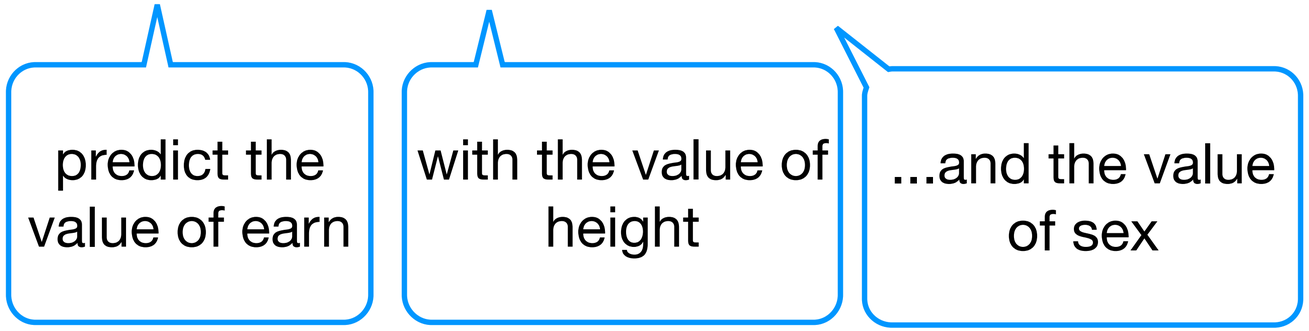
```
lm(earn ~ height, data = wages)
lm(earn ~ height + sex, data = wages)
lm(earn ~ height + sex + height:sex, data = wages)
```

predict the value of earn

with the value of height

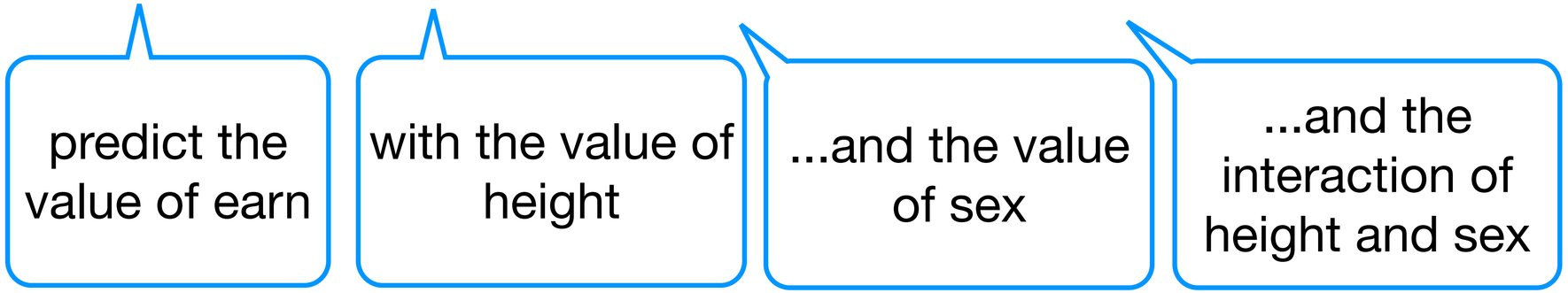
Syntax

```
lm(earn ~ height, data = wages)
lm(earn ~ height + sex, data = wages)
lm(earn ~ height + sex + height:sex, data = wages)
```



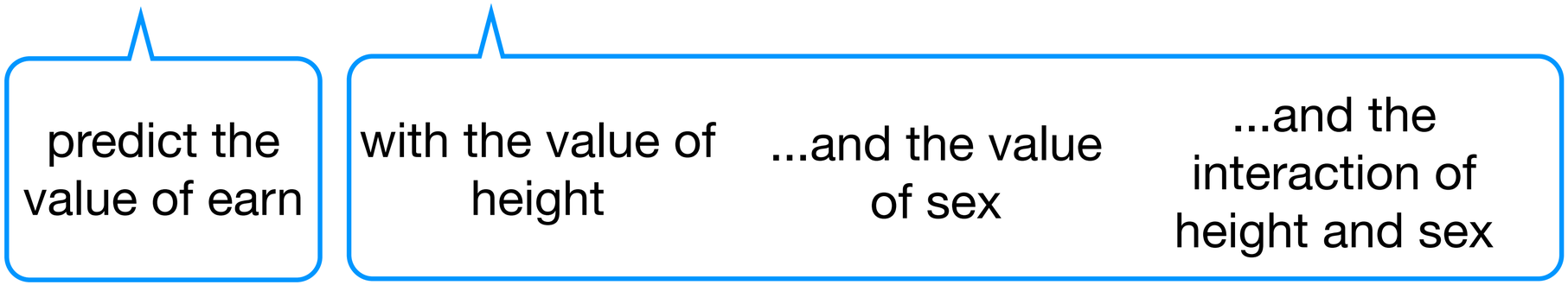
Syntax

```
lm(earn ~ height, data = wages)
lm(earn ~ height + sex, data = wages)
lm(earn ~ height + sex + height:sex, data = wages)
```



Syntax

```
lm(earn ~ height, data = wages)
lm(earn ~ height + sex, data = wages)
lm(earn ~ height + sex + height:sex, data = wages)
lm(earn ~ height * sex, data = wages)
```



Syntax

lm(earn ~ height, data = wages)

lm(earn ~ height + sex, data = wages)

lm(earn ~ **height + sex + height:sex**, data = wages)

lm(earn ~ **height * sex**, data = wages)

shortcuts

* is shorthand for both first order terms and the interaction effect

^2 is shorthand for all first order terms and all second order interactions

(i.e., these create the same model)

```
lm(earn ~ height + sex + height:sex, data = w1)
lm(earn ~ height * sex, data = w1)
lm(earn ~ (height + sex)^2, data = w1)
```

three way interactions

add a three way interaction by combining three variables with colons, height:sex:race

^3 is shorthand for all first order terms and all second order interactions and all third order interactions

```
lm(earn ~ (height + sex + race)^3,
   data = w1)
```

operator	adds to model
a + b	a and b
a:b	the interaction of a and b
a*b	a, b, and the interaction of a and b
(a + b + c + ...)^n	a, b, c, ... and all n order or less interactions between a, b, c, ...

Inference for multivariate regression

Multicollinearity

If the x_i are correlated, it can bias the p-values.

`cor` calculates the correlation of two numeric vectors. A score close to 1 or -1 implies the vectors are correlated.

```
cor(wages$height, wages$ed)
# 0.114
cor(wages$height, wages$age)
# -0.134
cor(wages$height, as.numeric(wages$sex))
# -0.704
```

Multiple testing bias

You can expect a p-value < 0.05 once every 20 p-values due to random chance.

Probability of one pvalue < 0.05 = 0.05

Probability of at least one of 2 pvalues < 0.05 = 0.098

.

.

.

Probability of at least one of 20 pvalues < 0.05 = 0.64

Multiple testing bias

If your model includes multiple β's you can

1. require a lower α for significance (α / p)
2. First test that the model as a whole is statistically significant

```
summary(m4)
# Call:
# lm(formula = earn ~ height + sex + height:sex, data = wages)

# Residuals:
#    Min     1Q Median     3Q    Max
# -49699 -20090  -5034  11553 271709

# Coefficients:
#                  Estimate Std. Error t value Pr(>|t|)
# (Intercept)      -42677.4    30488.0  -1.400  0.16180
# height             1265.9      434.9   2.911  0.00366 **
# sexfemale         30510.4    3964
# height:sexfemale   -701.4      58
# ---
# Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
1

# Residual standard error: 29340 on 1375 degrees of freedom
# Multiple R-squared:  0.1205, Adjusted R-squared:  0.1186
# F-statistic: 62.82 on 3 and 1375 DF,  p-value: < 2.2e-16
```

The model predicts y better than random chance

anova

This is an anova test, which you can also use to compare models

```
anova(m1, m4)
# Analysis of Variance Table
#
# Model 1: earn ~ height
# Model 2: earn ~ height + sex + height:sex
#   Res.Df           RSS Df   Sum of Sq      F    Pr(>F)
# 1   1377 1231834230435
# 2   1375 1184037470387  2 47796760048 27.753 1.528e-12 ***
# ---
# Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1
```

The improvement of m4 over m1 is unlikely to be due to random chance

anova

...as well as the variables within a single model

A model that includes heights + sex + heights:sex does not do significantly better than one that includes heights + sex

```
anova(m4)
Analysis of Variance Table

Response: earn
             Df   Sum Sq  Mean Sq F value  Pr(>F)
height        1 2.64e+10 2.64e+10   76.29 < 2e-16 ***
sex           1 1.34e+10 1.34e+10   38.61 7.2e-10 ***
height:sex    1 4.97e+08 4.97e+08    1.43    0.23
Residuals  1188 4.11e+11 3.46e+08
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Recap

Building models

Single x	`mod <- lm(y ~ x, data = df)`
Multiple x	`mod <- lm(y ~ x1 + x2, data = df)`
Interactions	`mod <- lm(y ~ x1 + x2 + x1:x2, data = df)`

Interpreting models

Coefficients	`coef(mod)`
Residuals	`resid(mod)`
Confidence intervals	`confint(mod)`
P-Values	`summary(mod)`
Compare models	`anova(mod1, mod2)`