Class 10:
More simple linear regression

Alison Presmanes Hill

Swiss Fertility and Socioeconomic Indicators (1888)

Details (paraphrasing Mosteller and Tukey):

- Switzerland, in 1888, was entering a period known as the *demographic transition*; i.e., its fertility was beginning to fall from the high level typical of underdeveloped countries.
- The data collected are for 47 French-speaking "provinces" at about 1888.
- Here, all variables are scaled to *[0, 100]*.

6 swiss variables

- Fertility: (common standardized measure)
- Agriculture: % of males involved in agriculture as an occupation
- Examination: % draftees received highest mark on army exam
- Education: % draftees educated beyond primary school
- Catholic: % Catholic (as opposed to Protestant)
- InfantMortality: Percent of live births who live less than one year

HLO swiss

```
> library(datasets)
> data(swiss)
> names(swiss)
[1] "Fertility"      "Agriculture"      "Examination"      "Education"        "Catholic"
[6] "Infant.Mortality"
> head(swiss)
             Fertility Agriculture Examination Education Catholic Infant.Mortality
Courtelary        80.2        17.0          15        12     9.96             22.2
Delemont          83.1        45.1           6         9    84.84             22.2
Franches-Mnt      92.5        39.7           5         5    93.40             20.2
Moutier           85.8        36.5          12         7    33.77             20.3
Neuveville        76.9        43.5          17        15     5.16             20.6
Porrentruy        76.1        35.3           9         7    90.57             26.6
> ?swiss
```

HLO swiss (continued)

```
> glimpse(swiss)
Observations: 47
Variables:
$ Fertility         (dbl) 80.2, 83.1, 92.5, 85.8, 76.9, 76.1, 83.8, 92.4, 82.4, 82.9, 87.1, 64...
$ Agriculture       (dbl) 17.0, 45.1, 39.7, 36.5, 43.5, 35.3, 70.2, 67.8, 53.3, 45.2, 64.5, 62...
$ Examination       (int) 15, 6, 5, 12, 17, 9, 16, 14, 12, 16, 14, 21, 14, 19, 22, 18, 17, 26,...
$ Education         (int) 12, 9, 5, 7, 15, 7, 7, 8, 7, 13, 6, 12, 7, 12, 5, 2, 8, 28, 20, 9, 1...
$ Catholic          (dbl) 9.96, 84.84, 93.40, 33.77, 5.16, 90.57, 92.85, 97.16, 97.67, 91.38, ...
$ Infant.Mortality (dbl) 22.2, 22.2, 20.2, 20.3, 20.6, 26.6, 23.6, 24.9, 21.0, 24.4, 24.5, 16...
```

tolower()

```
> colnames(swiss) <- swiss %>% # dplyr
+   colnames() %>% # colnames is built into base R
+   tolower() # tolower is also built into base R
> colnames(swiss)
[1] "fertility"        "agriculture"      "examination"      "education"        "catholic"
[6] "infant.mortality"
```

R Style Guide

- http://adv-r.had.co.nz/Style.html

- *"Good coding style is like using correct punctuation. You can manage without it, but it sure makes things easier to read. As with styles of punctuation, there are many possible variations. The following guide describes the style that I use (in this book and elsewhere). It is based on Google's R style guide, with a few tweaks. You don't have to use my style, but you really should use a consistent style."*- Hadley Wickham

Naming things

- `SomeLikeCamelCase`
- `some.do.the.dot.thing`
- `i_like_snake_case`

- Do you need to name that thing?

```
> x <- 4*3
> x
[1] 12
> 4*3
[1] 12
```

- Unless you need to use x later on, no you don't need to create an object

Spacing

- Place spaces around all infix operators (`=`, `+`, `-`, `<-`, etc.). The same rule applies when using `=` in function calls. Always put a space after a comma, and never before (just like in regular English).

```
# Good
average <- mean(feet / 12 + inches, na.rm = TRUE)

# Bad
average<-mean(feet/12+inches,na.rm=TRUE)
```

Indentation

- When indenting your code, use two spaces. Never use tabs or mix tabs and spaces.
- The only exception is if a function definition runs over multiple lines. In that case, indent the second line to where the definition starts.

```
# Good
ggplot(mtcars, aes(x = cyl, y = mpg)) +
      geom_point() +
      geom_line()
# Bad
ggplot(mtcars, aes(x = cyl, y = mpg)) +
geom_point() + geom_line()
```

Organization

- Comment your code. Each line of a comment should begin with the comment symbol and a single space: **#**.
- Comments should explain the why, not the what.
- Use commented lines of **-** and **=** to break up your file into easily readable chunks.

```
# Load data ----------------------------

# Plot data ----------------------------
```

First chunk: your midterm .Rmd file

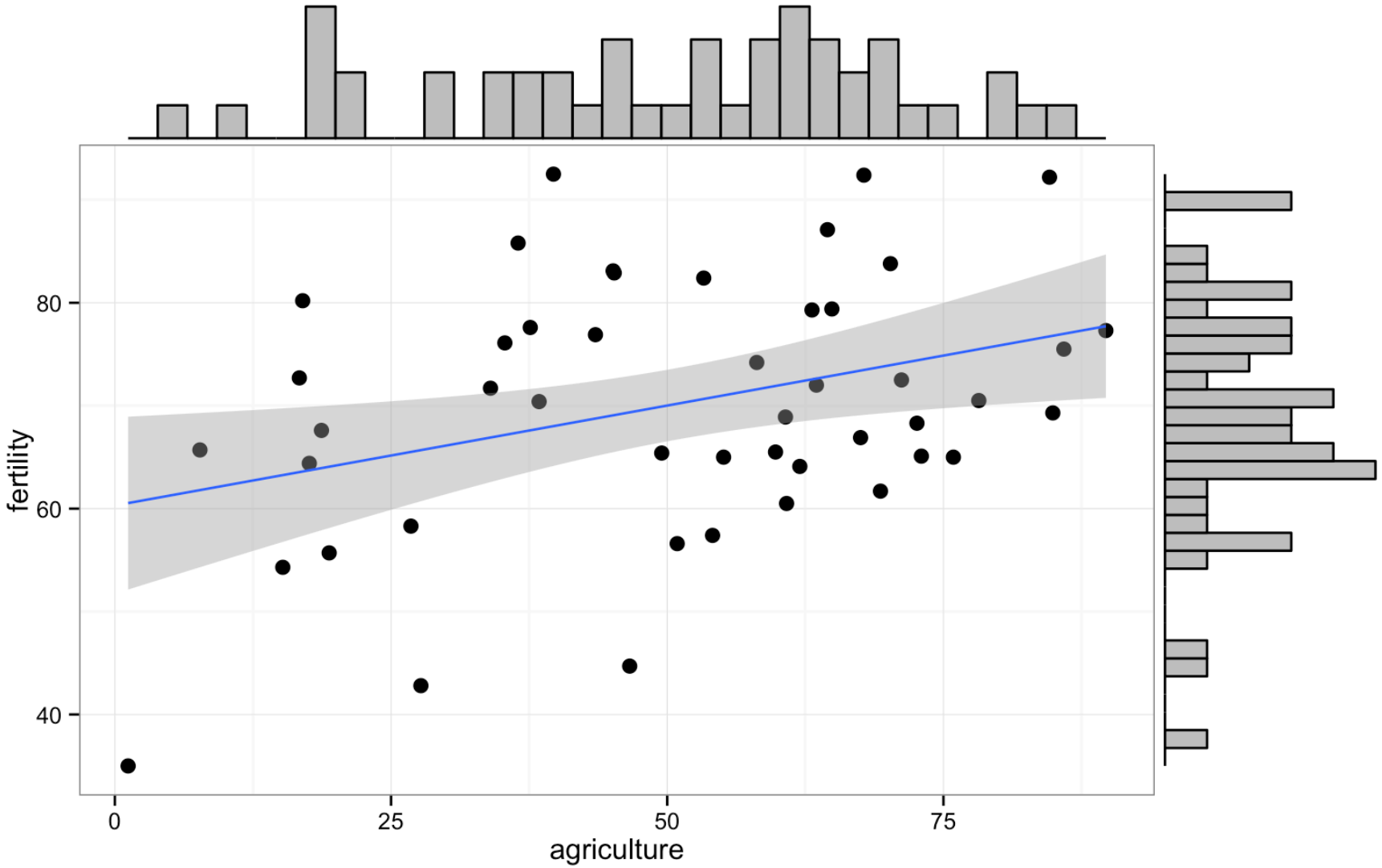
```
```{r setup, cache = FALSE}
library(knitr)
knitr::opts_chunk$set(error = TRUE, comment = NA, warnings = FALSE, messages =
FALSE, tidy = TRUE, fig.path="Figs/", echo = TRUE, include = TRUE, digits = 3)
```
```

Shhh! for loading packages (do this 1x!)

```
```{r load_packages}
suppressWarnings(suppressMessages(library(plyr)))
suppressWarnings(suppressMessages(library(dplyr)))
suppressWarnings(suppressMessages(library(ggplot2)))
suppressWarnings(suppressMessages(library(boot)))
suppressWarnings(suppressMessages(library(psych)))
suppressWarnings(suppressMessages(library(broom)))
suppressWarnings(suppressMessages(library(coefplot)))
```
```

Think before you create subsets of your data …

- Do computations or make figures *in situ* – don't create little copies and excerpts of your data. This will leave a cleaner workspace and cleaner code.
- Before you ever create a subset of your data, ask yourself: do I need this to compute or graph something (the same thing) for each level of something (as in, do the same thing repetitively)?
- If YES, use proper data aggregation techniques in dplyr or facetting in ggplot2 – don't subset the data.
- Or, more realistic, only subset the data as a temporary measure while you develop your elegant code for computing on or visualizing these data subsets.
- *"Copies and excerpts of your data clutter your workspace, invite mistakes, and sow general confusion. Avoid whenever possible."*– Jenny Bryan

SIMPLE LINEAR REGRESSION

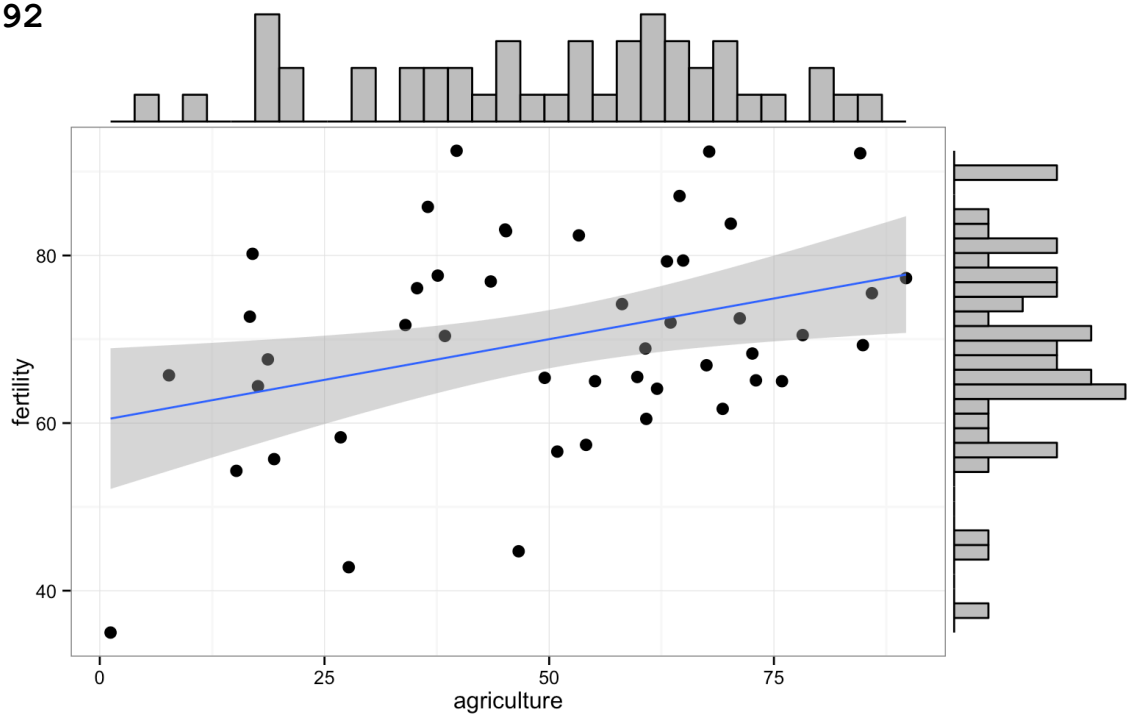
Did provinces that were more agricultural have higher fertility rates?


fertility
80
60
40
0
25
50
75
agriculture

```
> with(swiss, cor.test(fertility, agriculture))

        Pearson's product-moment correlation

data:  fertility and agriculture
t = 2.5316, df = 45, p-value = 0.01492
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.07334947 0.58130587
sample estimates:
      cor
0.3530792
```



```
> m_agr <- lm(fertility ~ agriculture, data = swiss) # simple linear regression
> summary(m_agr)
```

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon, i = 1, \ldots, n$$

```
Call:
lm(formula = fertility ~ agriculture, data = swiss)

Residuals:
     Min       1Q   Median       3Q
-25.5374  -7.8685  -0.6362   9.0464  2

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 60.30438    4.25126  14.185   <2e-16 ***
agriculture  0.19420    0.07671   2.532   0.0149 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.82 on 45 degrees of freedom
Multiple R-squared:  0.1247,      Adjusted R-squared:  0.1052
F-statistic: 6.409 on 1 and 45 DF,  p-value: 0.01492
```

```
> swiss %>% summarise(r2 = cor(agriculture, fertility)^2)
         r2
1 0.1246649
```

```
> tidy(m_agr) # broom
         term   estimate  std.error statistic      p.value
1 (Intercept) 60.3043752 4.25125562 14.185074 3.216304e-18
2 agriculture  0.1942017 0.07671176  2.531577 1.491720e-02
```

```
> model.matrix(m_agr) # peek under the hood
             (Intercept) agriculture
Courtelary             1        17.0
Delemont               1        45.1
Franches-Mnt           1        39.7
Moutier                1        36.5
Neuveville             1        43.5
Porrentruy             1        35.3
Broye                  1        70.2
Glane                  1        67.8
Gruyere                1        53.3
Sarine                 1        45.2
Veveyse                1        64.5
Aigle                  1        62.0
Aubonne                1        67.5
Avenches               1        60.7
Cossonay               1        69.3
Echallens              1        72.6
Grandson               1        34.0
Lausanne               1        19.4
La Vallee              1        15.2
Lavaux                 1        73.0
Morges                 1        59.8
Moudon                 1        55.1
Nyone                  1        50.9
Orbe                   1        54.1
Oron                   1        71.2
Payerne                1        58.1
…
```

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_n \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

```
> m_agr <- lm(fertility ~ agriculture, data = swiss) # simple linear regression
> summary(m_agr)

Call:
lm(formula = fertility ~ agriculture, data = swiss)

Residuals:
     Min
-25.5374  -7.

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 60.30438    4.25126  14.185   <2e-16 ***
agriculture  0.19420    0.07671   2.532   0.0149 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.82 on 45 degrees of freedom
Multiple R-squared:  0.1247,       Adjusted R-squared:  0.1052
F-statistic: 6.409 on 1 and 45 DF,  p-value: 0.01492
```

```
> X <- with(swiss, cbind(1, agriculture))
> solve( t(X) %*% X ) %*% t(X) %*% swiss$fertility
                    [,1]
              60.3043752
agriculture    0.1942017
```

```
> tidy(m_agr) # broom
         term   estimate  std.error statistic      p.value
1 (Intercept) 60.3043752 4.25125562 14.185074 3.216304e-18
2 agriculture  0.1942017 0.07671176  2.531577 1.491720e-02
```

Matrices

- Linear regression is essentially the solution to a matrix inversion problem

$$2\mathbf{X}^\top(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = 0$$

$$\mathbf{X}^\top\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^\top\mathbf{Y}$$

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$$

```
> X <- with(swiss, cbind(1, agriculture))
> solve( t(X) %*% X ) %*% t(X) %*% swiss$fertility
                    [,1]
            60.3043752
agriculture  0.1942017
```

The simplest linear model in the world

```
> m_mean <- lm(fertility ~ 1, data = swiss) # intercept ONLY
```

```
> model.matrix(m_mean)
             (Intercept)
Courtelary             1
Delemont               1
Franches-Mnt           1
Moutier                1
Neuveville             1
Porrentruy             1
Broye                  1
Glane                  1
Gruyere                1
Sarine                 1
Veveyse                1
Aigle                  1
Aubonne                1
Avenches               1
Cossonay               1
Echallens              1
Grandson               1
Lausanne               1
La Vallee              1
Lavaux                 1
Morges                 1
Moudon                 1
Nyone                  1
Orbe                   1
Oron                   1
Payerne                1
...
```

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} 1 & \\ 1 & \\ \vdots & \\ 1 & \end{pmatrix} \begin{pmatrix} \beta_0 \\ \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

The simplest linear model in the world

```
> m_mean <- lm(fertility ~ 1, data = swiss) # intercept ONLY
> summary(m_mean)

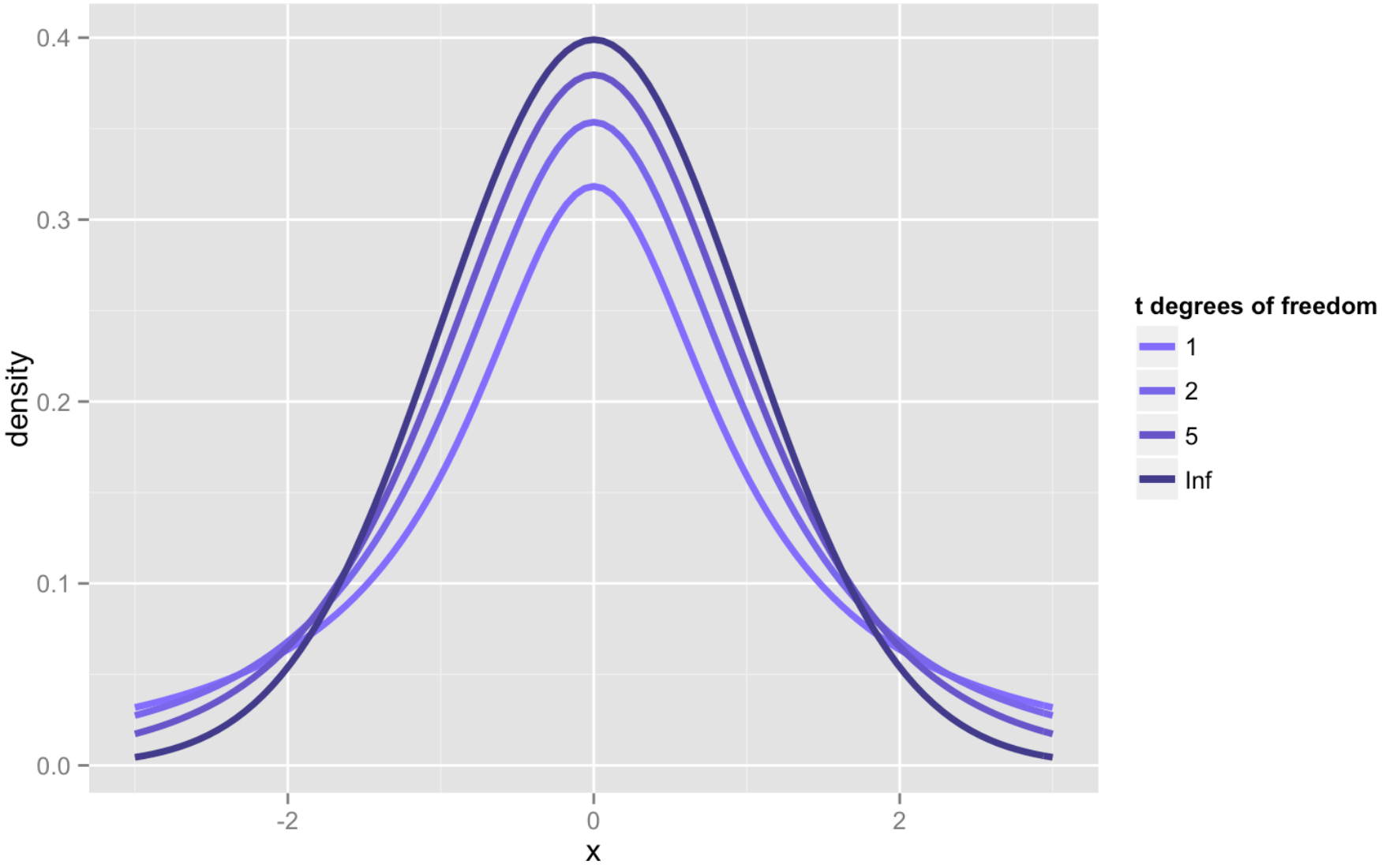
Call:
lm(formula = fertility ~ 1, data = swiss)

Residuals:
    Min      1Q  Median      3Q     Max
-35.143  -5.443   0.257   8.307  22.357

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   70.143      1.822   38.49   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

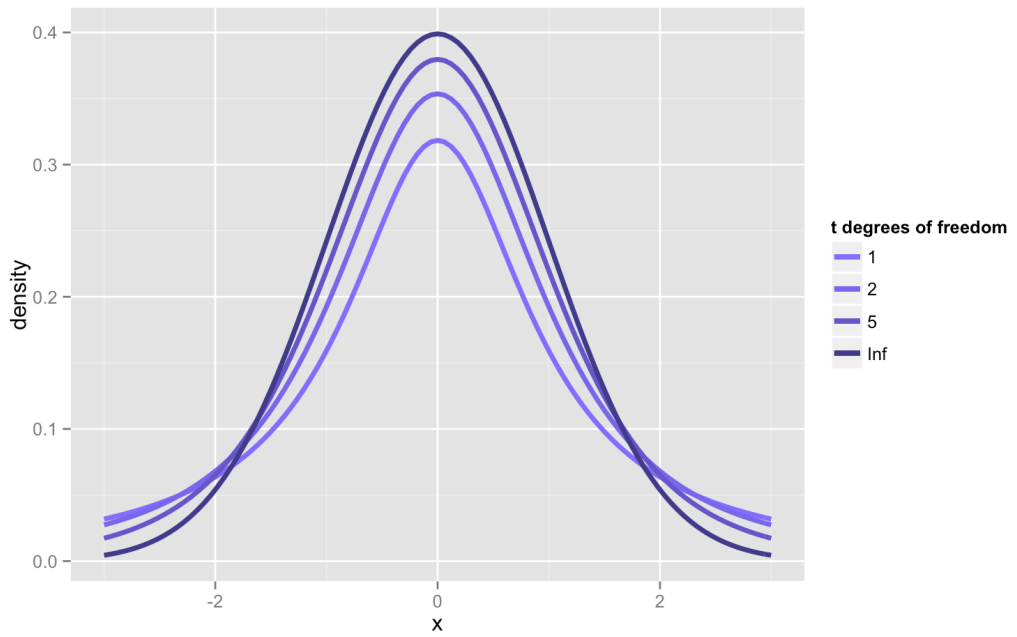
Residual standard error: 12.49 on 46 degrees of freedom
```

Recall student's *t*-distribution



Student's *t*-distribution

- For statistics that are the ratio of an estimate to its standard deviation (the "standard error")



The simplest linear model in the world

```
> augment(m_mean)[1:5]
      .rownames fertility  .fitted  .se.fit      .resid
1     Courtelary      80.2 70.14255 1.822101  10.0574468
2       Delemont      83.1 70.14255 1.822101  12.9574468
3   Franches-Mnt      92.5 70.14255 1.822101  22.3574468
4        Moutier      85.8 70.14255 1.822101  15.6574468
5     Neuveville      76.9 70.14255 1.822101   6.7574468
6     Porrentruy      76.1 70.14255 1.822101   5.9574468
7          Broye      83.8 70.14255 1.822101  13.6574468
8          Glane      92.4 70.14255 1.822101  22.2574468
9        Gruyere      82.4 70.14255 1.822101  12.2574468
10        Sarine      82.9 70.14255 1.822101  12.7574468
11       Veveyse      87.1 70.14255 1.822101  16.9574468
12         Aigle      64.1 70.14255 1.822101  -6.0425532
13       Aubonne      66.9 70.14255 1.822101  -3.2425532
14      Avenches      68.9 70.14255 1.822101  -1.2425532
15      Cossonay      61.7 70.14255 1.822101  -8.4425532
```

Nested models

```
> m_mean <- lm(fertility ~ 1, data = swiss) # intercept ONLY
> m_agr <- lm(fertility ~ agriculture, data = swiss)
> anova(m_mean, m_agr)
Analysis of Variance Table

Model 1: fertility ~ 1
Model 2: fertility ~ agriculture
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     46 7178.0
2     45 6283.1  1    894.84 6.4089 0.01492 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> glance(m_agr)
  r.squared adj.r.squared    sigma statistic   p.value df    logLik
1 0.1246649      0.105213 11.81629  6.408884 0.0149172  2 -181.7337
```

Sequences of nested models

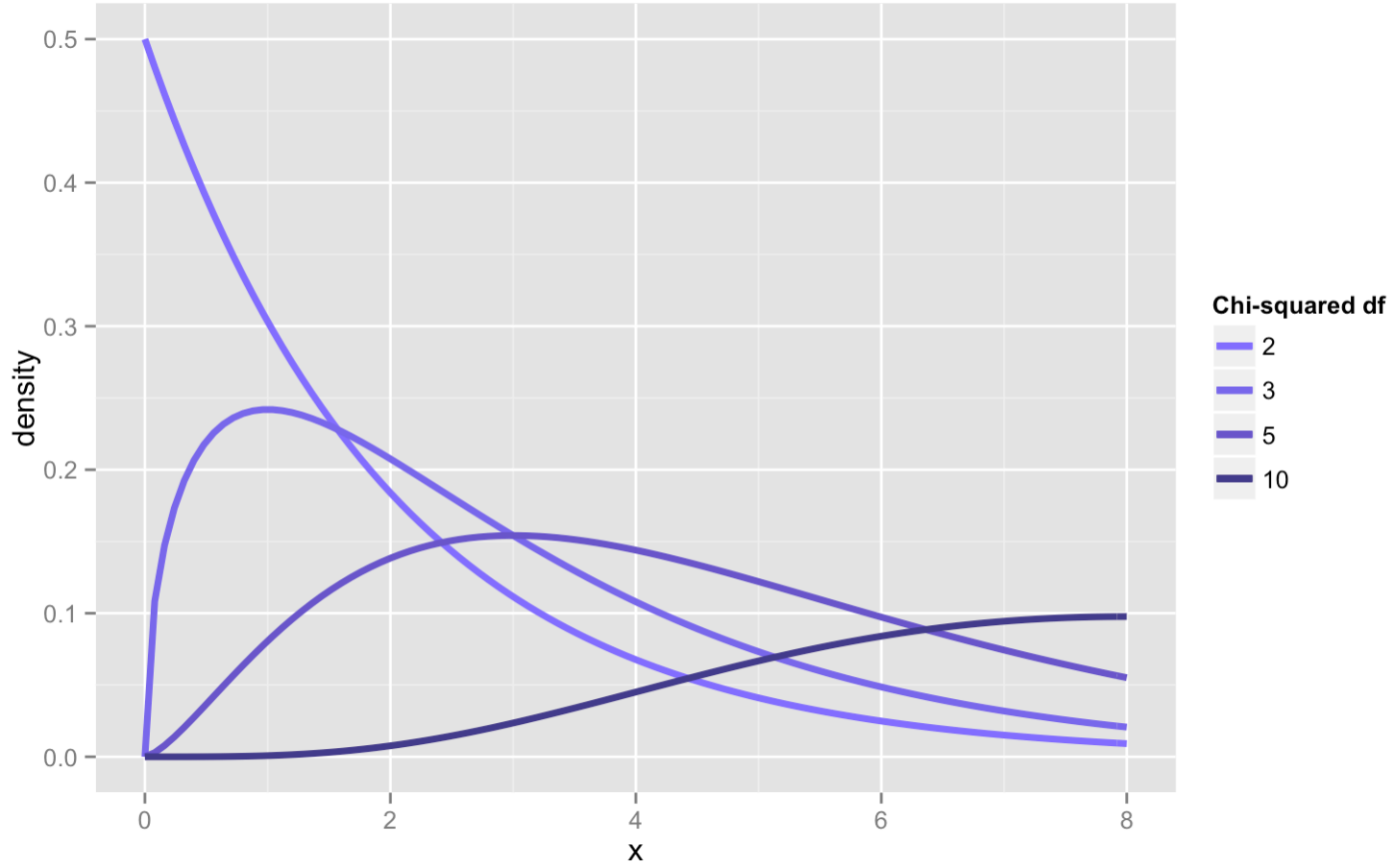
- The more complex of two nested models will always fit at least as well as the less complex model.

Sequences of nested models

- The "1" is implicitly included in your `lm()` formula

New distribution: χ^2

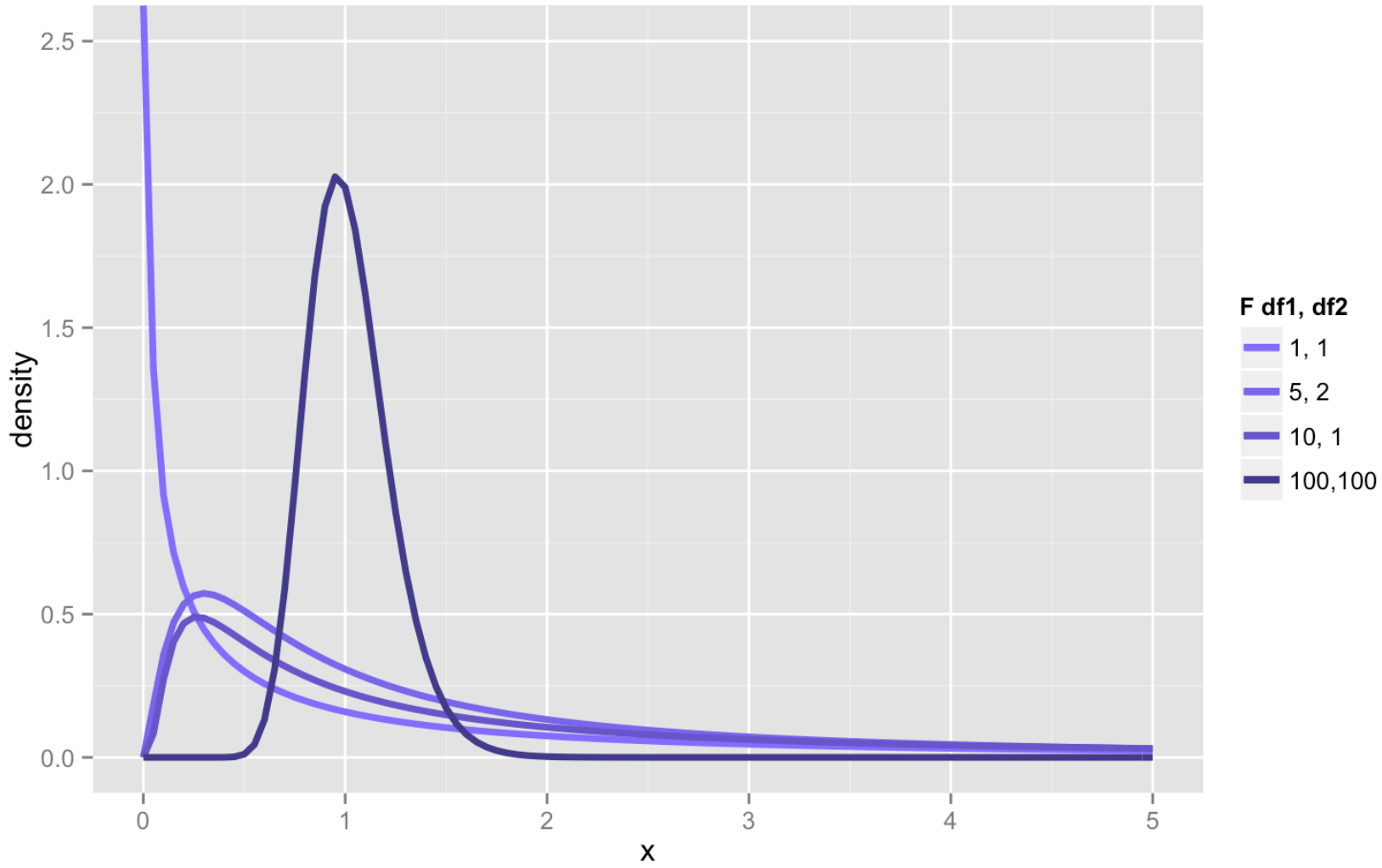
- Distribution of a **sum of the squares** of *k* independent standard normal random variables (*k* degrees of freedom)



New distribution: F

- Distribution of a **ratio of two χ^2 distributed variables** (divided by their degrees of freedom)
- Asymmetric, minimum = 0, no maximum
- Two degrees of freedom:
  - one for numerator
  - one for denominator

New distribution: F



Nested models

```
> m_mean <- lm(fertility ~ 1, data = swiss) # model 1
> m_agr <- lm(fertility ~ agriculture, data = swiss) # model 2
> anova(m_mean, m_agr) # compare model 1 vs. model 2
Analysis of Variance Table

Model 1: fertility ~ 1
Model 2: fertility ~ agriculture
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     46 7178.0
2     45 6283.1  1    894.84 6.4089 0.01492 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> glance(m1)
  r.squared adj.r.squared    sigma statistic   p.value df    logLik
1 0.1246649      0.105213 11.81629  6.408884 0.0149172  2 -181.7337
```

$$F_{1,45} = \frac{\frac{894.84}{2-1}}{\frac{6283.1}{45}} = 6.4089$$

$$F_{df_{m1}-df_{m2},df_{m2}} = \frac{\frac{RSS_{m1}-RSS_{m2}}{p_{m2}-p_{m1}}}{\frac{RSS_{m2}}{df_{m2}}}$$

```
> m_agr <- lm(fertility ~ agriculture, data = swiss) # simple linear regression
> summary(m_agr)

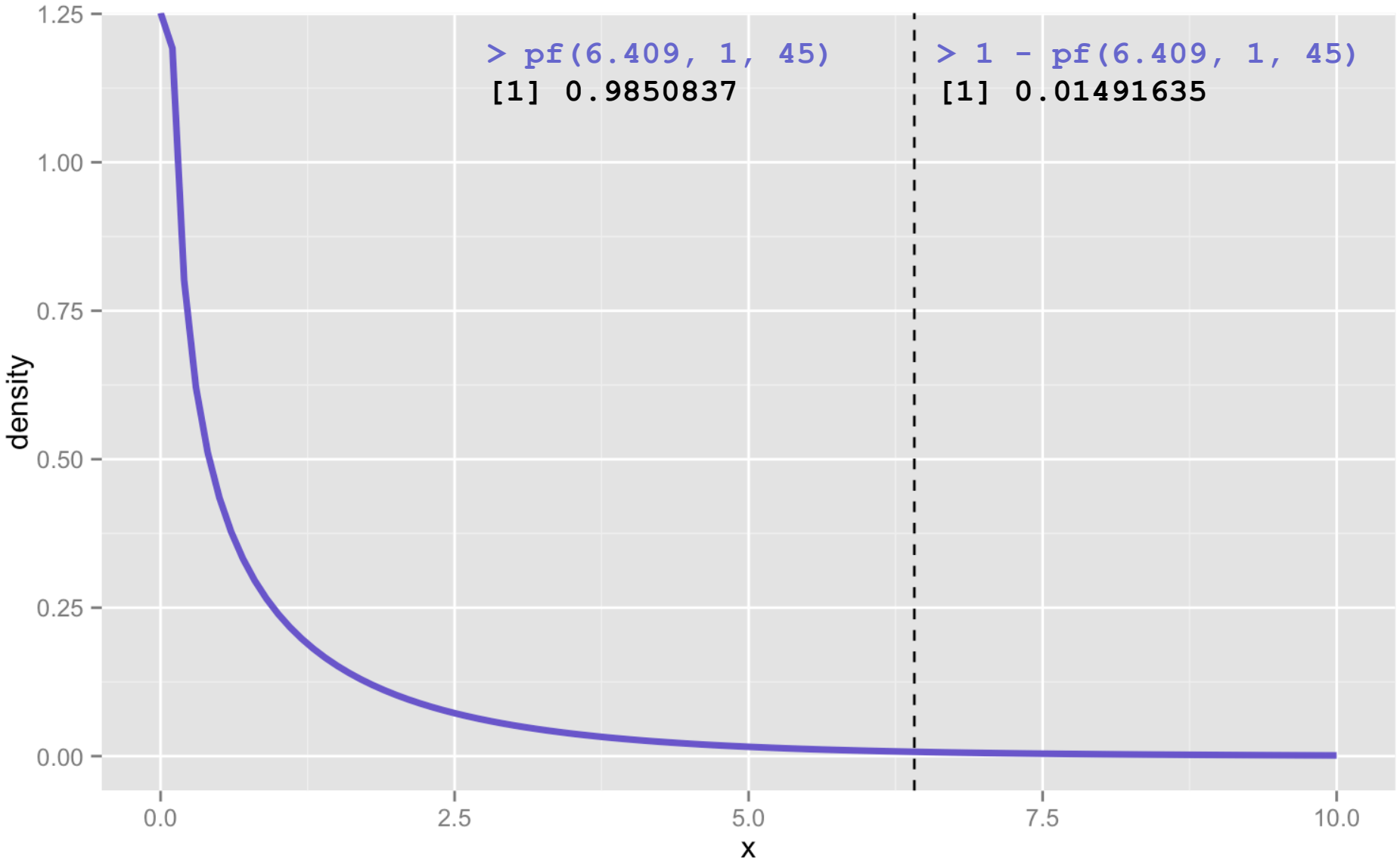
Call:
lm(formula = fertility ~ agriculture, data = swiss)

Residuals:
     Min       1Q   Median       3Q      Max
-25.5374  -7.8685  -0.6362   9.0464  24.4858

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 60.30438    4.25126  14.185   <2e-16 ***
agriculture  0.19420    0.07671   2.532   0.0149 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.82 on 45 degrees of freedom
Multiple R-squared:  0.1247,      Adjusted R-squared:  0.1052
F-statistic: 6.409 on 1 and 45 DF,  p-value: 0.01492
```

$$F_{1,45} = \frac{\frac{894.84}{2-1}}{\frac{6283.1}{45}} = 6.4089$$


> pf(6.409, 1, 45)
[1] 0.9850837
> 1 - pf(6.409, 1, 45)
[1] 0.01491635
1.25
1.00
0.75
0.50
0.25
0.00
density
0.0
2.5
5.0
7.5
10.0
x

Sequences of nested models

- `lm()` takes ANY model you specify and compares it to the simplest linear model in the whole world- the one with just an intercept- the one with just the mean of Y.

Sequences of nested models

- But you can use **`anova()`** to compare any two nested models!

Sequences of nested models

- But you can use **`anova()`** to compare any two nested models!

Why should I care?

- `lm()` in R compares your stated model (with however many predictors you choose, can be > 1!) against the simplest linear model in the whole world- the mean model (the mean model includes just an intercept term and zero predictors).

- You can directly compare any two nested linear models using the `anova()` command (more on this later).

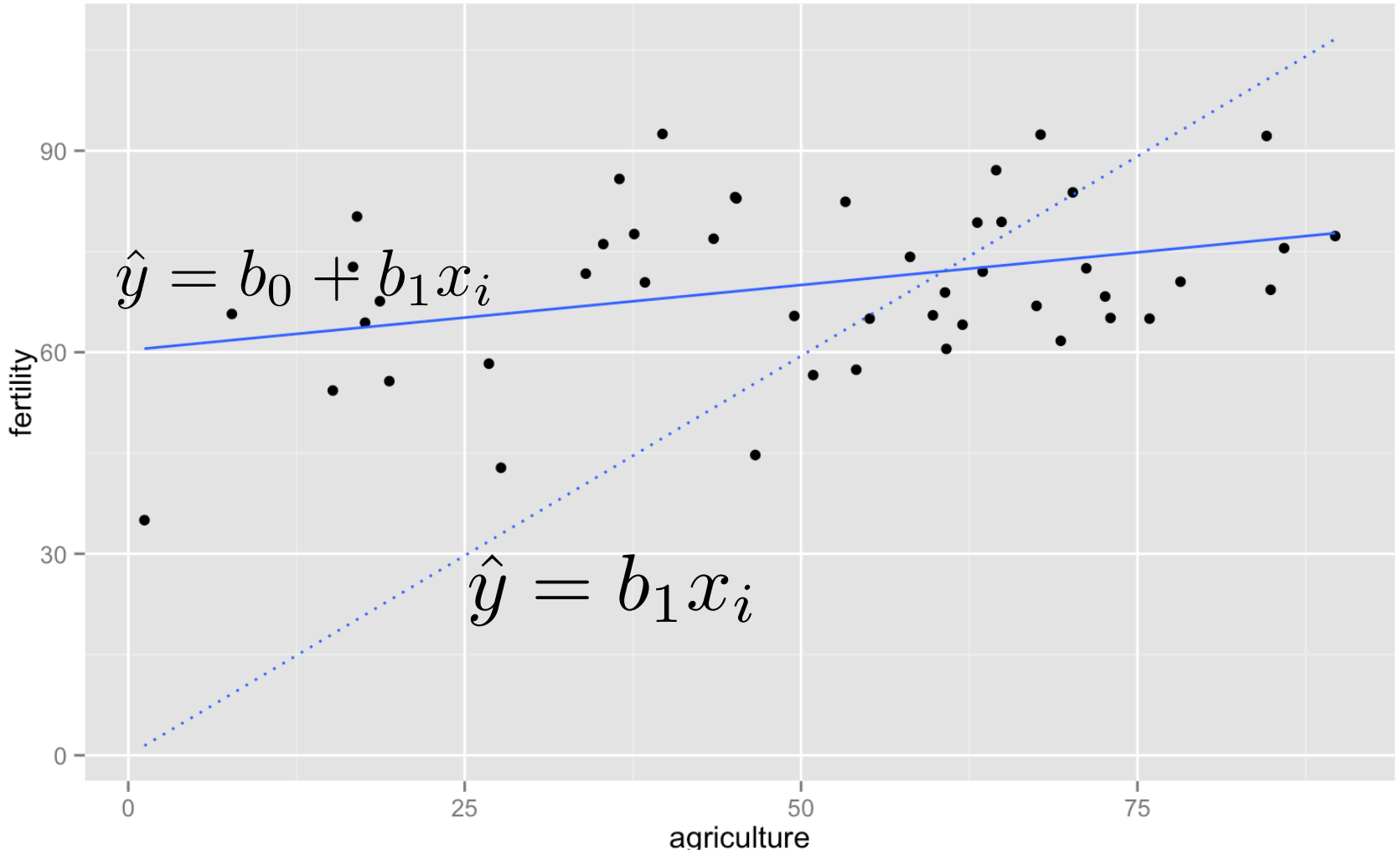
```
> m_nob0 <- lm(fertility ~ agriculture - 1, data = swiss) # no intercept
> model.matrix(m_nob0)
             agriculture
Courtelary          17.0
Delemont            45.1
Franches-Mnt        39.7
Moutier             36.5
Neuveville          43.5
Porrentruy          35.3
Broye               70.2
Glane               67.8
Gruyere             53.3
Sarine              45.2
Veveyse             64.5
Aigle               62.0
Aubonne             67.5
Avenches            60.7
Cossonay            69.3
Echallens           72.6
Grandson            34.0
Lausanne            19.4
La Vallee           15.2
Lavaux              73.0
Morges              59.8
...
```

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} \blacksquare & x_1 \\ \blacksquare & x_2 \\ \blacksquare & \vdots \\ \blacksquare & x_n \end{pmatrix} \begin{pmatrix} \blacksquare \\ \beta_1 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

```
> glance(m_nob0) # model without intercept including agriculture
  r.squared adj.r.squared    sigma statistic      p.value df    logLik      AIC      BIC deviance
1 0.8558076      0.852673 27.33762  273.0182 5.703938e-21  1 -221.6731 447.3462 451.0465  34377.9
> glance(m_agr) # model including intercept + agriculture
  r.squared adj.r.squared    sigma statistic   p.value df    logLik      AIC      BIC deviance
1 0.1246649      0.105213 11.81629  6.408884 0.0149172  2 -181.7337 369.4675 375.0179 6283.116
```

Why do I need an intercept?



summary.lm

```
z <- object
...
f <- z$fitted.values
...
if (is.null(w)) {
        mss <- if (attr(z$terms, "intercept"))
                sum((f - mean(f))^2)
        else sum(f^2)
        rss <- sum(r^2)
...
ans$r.squared <- mss/(mss + rss)
```

R legend:

`f` is our fitted values for y

`mss` is our model sums of squares

`rss` is our residual sums of squares

`mss + rss` = total sums of squares

R^2 is…

summary.lm

```
z <- object
…
f <- z$fitted.values
…
if (is.null(w)) {
        mss <- if (attr(z$terms, "intercept"))
            sum((f - mean(f))^2)
        else sum(f^2)
        rss <- sum(r^2)…
ans$r.squared <- mss/(mss + rss)
```

The mean of the fitted values is the same as…

$$\frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{i=1}^{n} \hat{y}$$

```
> slr_vars %>%
+   summarise(mean_y = mean(fertility),
+             mean_yhat = mean(.fitted))
    mean_y mean_yhat
1 70.14255  70.14255
```

Squared multiple correlation, R^2

Total sums of squares	Model sums of squares	Residual sums of squares
$\sum(y_i - \overline{y})^2$	$\sum(\hat{y}_i - \overline{y})^2$	$\sum(y_i - \hat{y}_i)^2$

total variation = "explained" variation + residual variation

$$R^2 = \frac{\text{explained variation}}{\text{total variation}}$$

$$= \frac{\sum(\hat{y}_i - \overline{y})^2}{\sum(y_i - \overline{y})^2}$$

$$= 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \overline{y})^2}$$

894.8

7177.9

= .1247

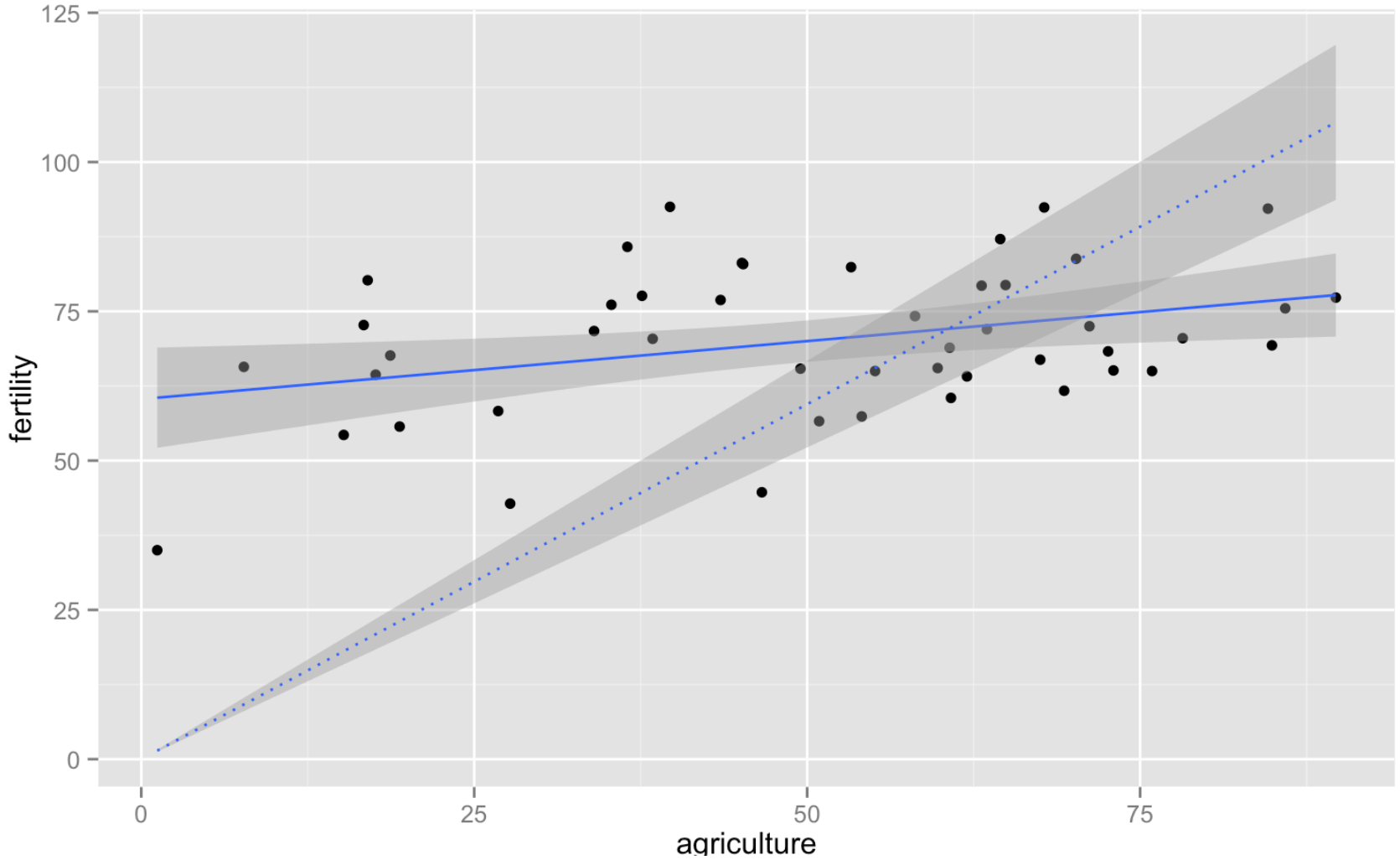
Only true when there is an intercept!

Why the different formulas?

- In the R help documents, they outline several reasons for the different formulas:
  - Otherwise, *R*2 could be negative (because the model with zero intercept can fit worse than the constant-mean model it is implicitly compared to).
  - If you set the slope to zero in the model with a line through the origin you get fitted values $\hat{y}=0$
  - The model with constant, non-zero mean is not nested in the model with a line through the origin.

- Bottom-line: unless you have a very very very strong reason for doing so, include the intercept term in your model. And friends don't let friends get excited about "too good to be true" results when the intercept term was left out.

```
> glance(m_nob0) # model without intercept including agriculture
  r.squared adj.r.squared    sigma statistic      p.value df    logLik      AIC      BIC deviance
1 0.8558076      0.852673 27.33762  273.0182 5.703938e-21  1 -221.6731 447.3462 451.0465  34377.9
> glance(m_agr) # model including intercept + agriculture
  r.squared adj.r.squared    sigma statistic   p.value df    logLik      AIC      BIC deviance
1 0.1246649      0.105213 11.81629  6.408884 0.0149172  2 -181.7337 369.4675 375.0179 6283.116
```



Confidence interval for R^2 : : `library(MBESS)`

We can also calculate a 95% confidence interval for R^2, providing more information about the preciseness of our estimates. The formula is based on Steiger and Fouladi (1992, Behavior Research Methods, Instruments & Computers, 4, 581-582):

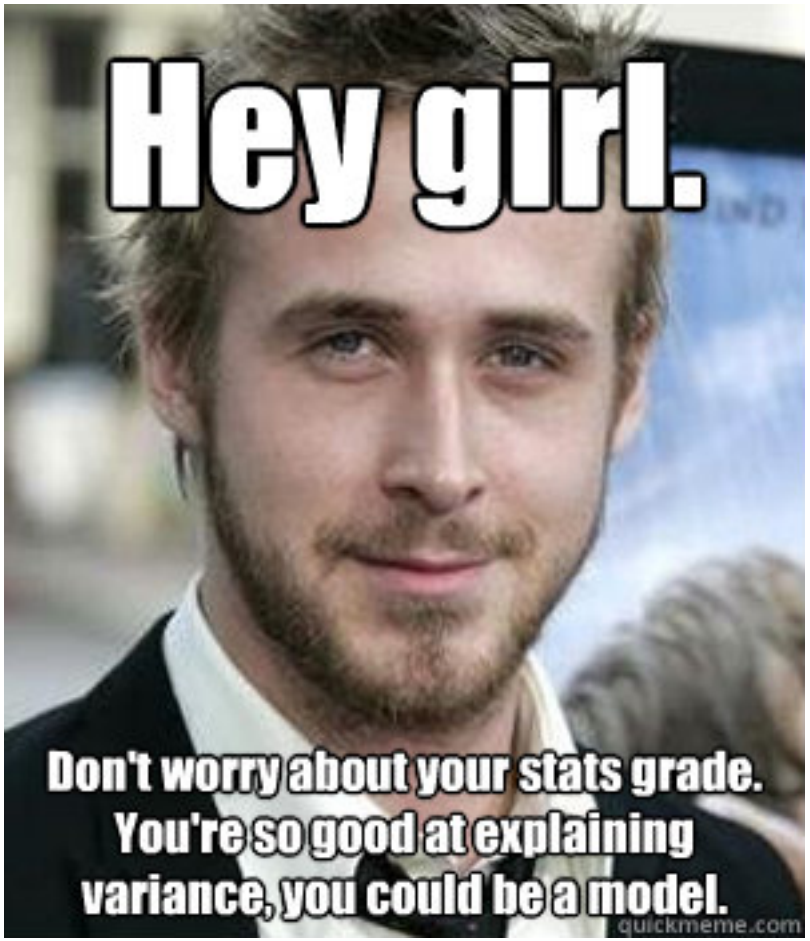
```
> ci.R2(glance(m1)$r.squared, df.1 = 2, df.2 = 45, conf.level = .95)
$Lower.Conf.Limit.R2
[1] 0

$Prob.Less.Lower
[1] 0.025

$Upper.Conf.Limit.R2
[1] 0.3119511

$Prob.Greater.Upper
[1] 0.025

> # ci.R2(0.1246649, df.1 = 2, df.2 = 45,
conf.level = .95)
```


Hey girl.
Don't worry about your stats grade. You're so good at explaining variance, you could be a model.
quickmeme.com

Prediction

- Confidence intervals often plotted with regression lines. There are actually two distinctly different kinds of predictions we can make:
  - **(Conditional) mean response of Y at X = x*:**

$$\overline{y}_{x*} = b_0 + b_1 x*$$

  - **New observation of Y at X = x*:**

$$\hat{y}_{x*} = b_0 + b_1 x * + e_{x*}$$

Standard errors of…

Predicted means

$$SE_{\overline{y}|x*} = s_e\sqrt{\frac{1}{n} + \frac{(x-\overline{x})^2}{\sum(x_i-\overline{x})^2}}$$

Predicted values

$$\begin{aligned} SE^2_{\hat{y}|x*} &= s^2_{Y|x} + SE^2_{\overline{y}|x*} \\ &= s^2_e + SE^2_{\overline{y}|x*} \end{aligned}$$

```
## Create a run of 100 new x's
newx <- data.frame(agriculture = seq(from = min(swiss$agriculture),
                                     to = max(swiss$agriculture),
                                     length.out = 100))
```

```
pred_mean <- data.frame(newx,
predict(m_agr, newx, interval =
"confidence"))
```

```
> head(pred_mean)
  agriculture      fit      lwr      upr
1    1.200000 60.53742 52.14410 68.93074
2    2.093939 60.71102 52.44326 68.97879
3    2.987879 60.88463 52.74201 69.02725
4    3.881818 61.05823 53.04033 69.07613
5    4.775758 61.23184 53.33821 69.12546
6    5.669697 61.40544 53.63562 69.17526
```

```
pred_obs <- data.frame(newx,
predict(m_agr, newx, interval =
"prediction"))
```

```
> head(pred_obs)
  agriculture      fit      lwr      upr
1    1.200000 60.53742 35.30150 85.77333
2    2.093939 60.71102 35.51659 85.90546
3    2.987879 60.88463 35.73098 86.03827
4    3.881818 61.05823 35.94468 86.17178
5    4.775758 61.23184 36.15769 86.30599
6    5.669697 61.40544 36.36999 86.44089
```

95% CI for predicting a conditional mean

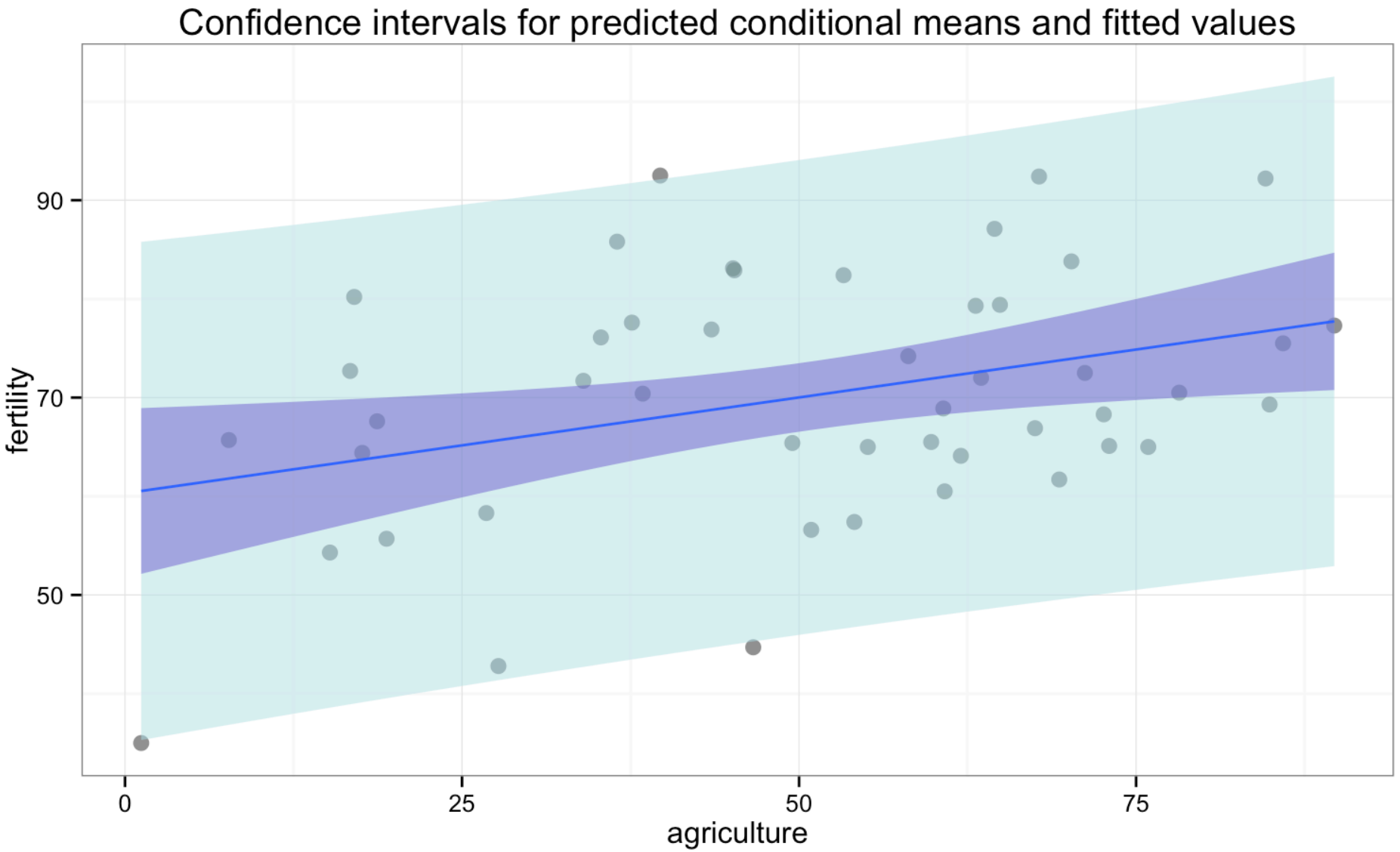
```
> pred_mean %>% filter(agriculture > 50 &
agriculture < 51)
  agriculture      fit      lwr      upr
1    50.36667 70.08567 66.6139 73.55744
```

I am 95% confident that a new region with agriculture percentage ≈ 50.4% will have a population conditional mean fertility between 66.6 and 73.6!

95% CI for predicting a new observation

```
> pred_obs %>% filter(agriculture > 50 &
agriculture < 51)
  agriculture      fit      lwr     upr
1    50.36667 70.08567 46.03454 94.1368
```

I am 95% confident that a new region with agriculture percentage ≈ 50.4% will have fertility between 46.0 and 94.1!


Confidence intervals for predicted conditional means and fitted values
fertility
90
70
50
0
25
50
75
agriculture

Scheffe corrected confidence interval for conditional means

```
scheffe.rescaled.ci <- function(model, conf.level, new){
  ## Get df and number of predictors from model object
  df <- model$df.residual
  p <- model$rank
  alpha <- 1 - conf.level
  ## NOTE Scheffe value uses 1-tailed F critical value
  scheffe.crit <- sqrt(p * qf(1 - alpha, p, df))
  ci <- predict(model, new, interval = "confidence", level =
conf.level)
  ## Create multiplier to expand the width of the ci
  multiplier <- scheffe.crit/qt(1 - alpha/2, df)
  ## Recompute the ci
  ci[,2] <- ci[,1] -(ci[,1]-ci[,2])*multiplier
  ci[,3] <- ci[,1] + (ci[,3]-ci[,1])*multiplier
  return(ci)
}
```

```
##Create a run of 100 new x's
newx <- data.frame(agriculture = seq(from = min(swiss$agriculture),
                                     to = max(swiss$agriculture),
                                     length.out = 100))
```

```
pred_mean <- data.frame(newx,
predict(m1, newx, interval =
"confidence"))
```

```
> head(pred_mean)
  agriculture      fit      lwr      upr
1    1.200000 60.53742 52.14410 68.93074
2    2.093939 60.71102 52.44326 68.97879
3    2.987879 60.88463 52.74201 69.02725
4    3.881818 61.05823 53.04033 69.07613
5    4.775758 61.23184 53.33821 69.12546
6    5.669697 61.40544 53.63562 69.17526
```

```
pred_mean_scheffe <- data.frame(newx,
scheffe.rescaled.ci(m1, 0.95, newx))
```

```
> head(pred_mean_scheffe)
  agriculture      fit      lwr      upr
1    1.200000 60.53742 49.98785 71.08699
2    2.093939 60.71102 50.31926 71.10278
3    2.987879 60.88463 50.65016 71.11909
4    3.881818 61.05823 50.98053 71.13593
5    4.775758 61.23184 51.31033 71.15334
6    5.669697 61.40544 51.63955 71.17133
```

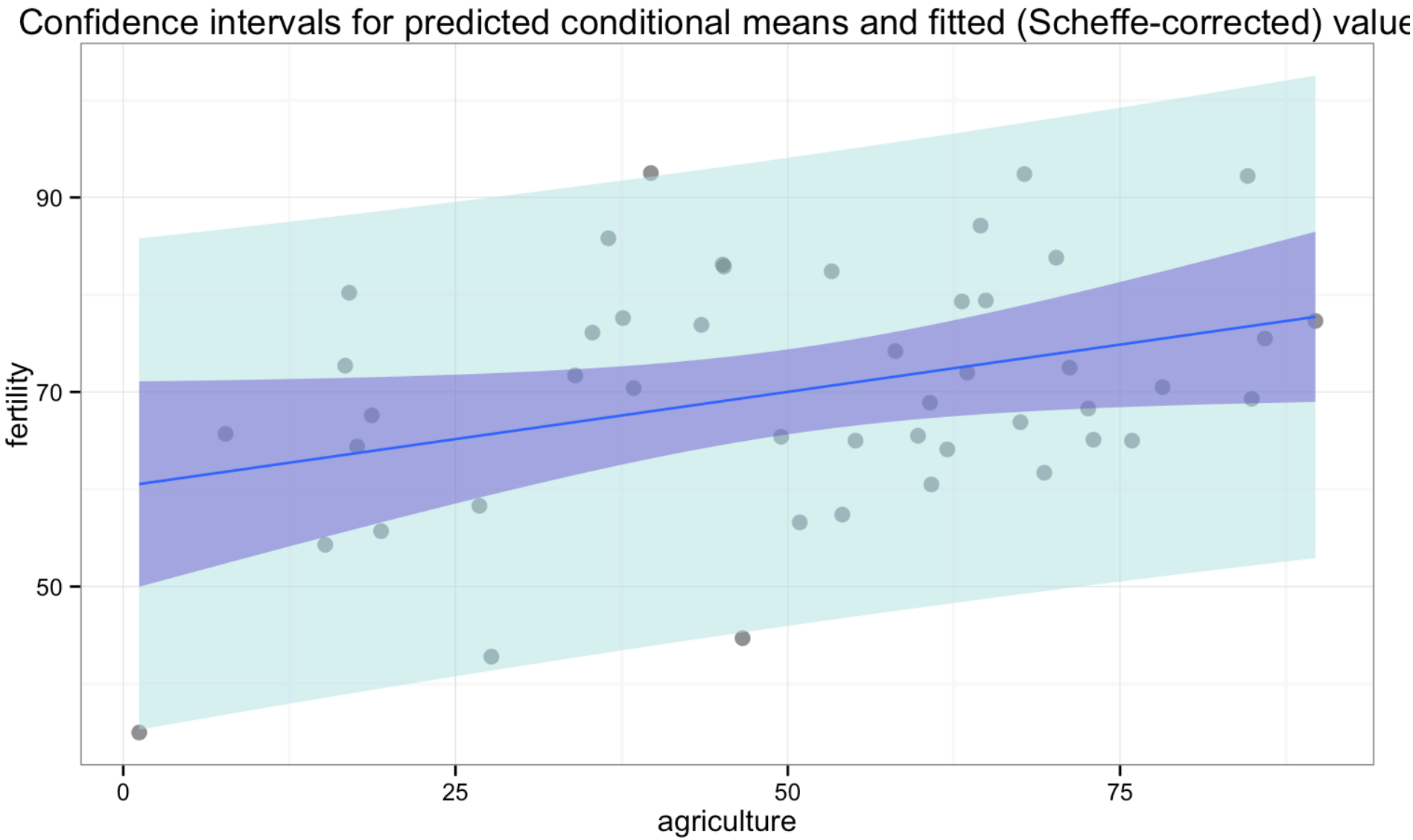
95% CI for predicting a conditional mean (Scheffe corrected)

```
> pred_mean_scheffe %>% filter(agriculture > 50 & agriculture < 51)
  agriculture      fit    lwr      upr
1    50.36667 70.08567 65.722 74.44934
```

~~I am 95% confident that a new region with agriculture percentage ≈ 50.4% will have a population conditional mean fertility between 66.6 and 73.6!~~

I am 95% confident that a new region with agriculture percentage ≈ 50.4% will have a population conditional mean fertility between 65.7 and 74.4!*

**when I have estimated this confidence interval for all possible estimated conditional means*


Confidence intervals for predicted conditional means and fitted (Scheffe-corrected) value
fertility
90
70
50
0
25
50
75
agriculture

Prediction: summing up

- This illustrates how *the uncertainty of predicted y-values as an estimate of Y is much greater than the uncertainty of predicted y-values as an estimate of the conditional mean of Y*.
- To predict a new conditional mean, the SE must only account for uncertainty in our estimate of the conditional mean
- To predict a new observation, the SE must also account variability in the conditional distribution