Multivariable regression examples

Regression Models

Brian Caffo, Jeff Leek and Roger Peng

Data set for discussion

```
require(datasets); data(swiss); ?swiss
```

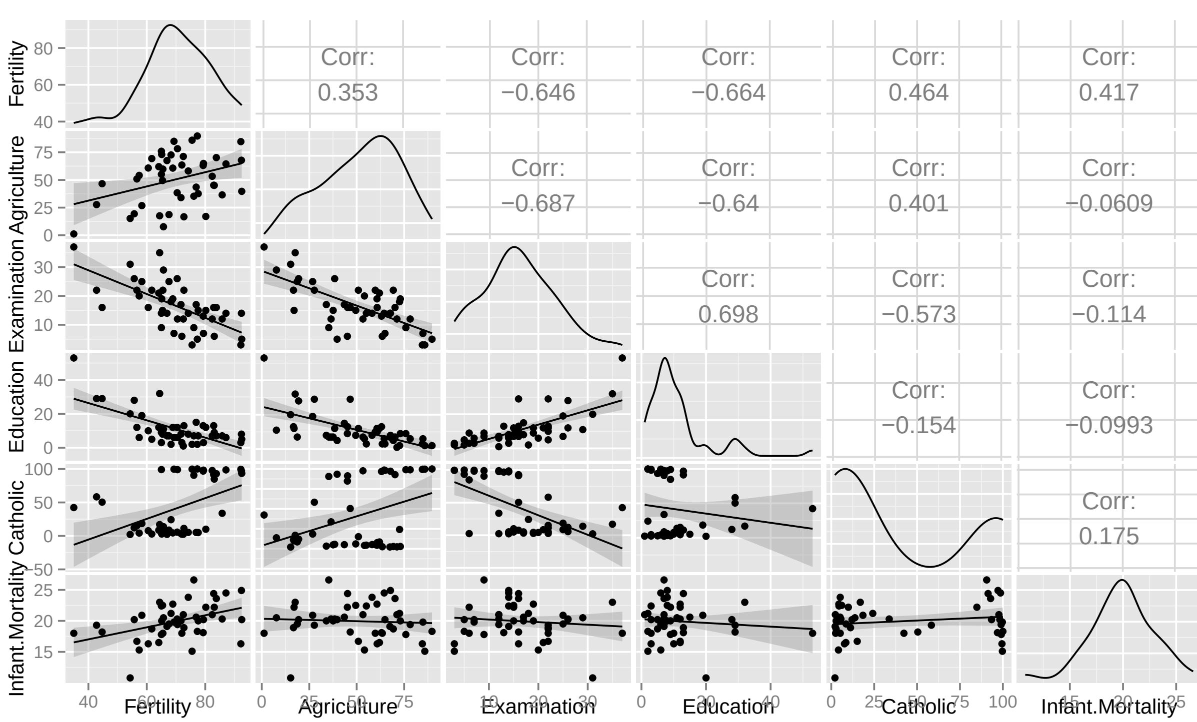
Standardized fertility measure and socio-economic indicators for each of 47 French-speaking provinces of Switzerland at about 1888.

A data frame with 47 observations on 6 variables, each of which is in percent, i.e., in [0, 100].

- [,1] Fertility a common standardized fertility measure
- [,2] Agriculture % of males involved in agriculture as occupation
- [,3] Examination % draftees receiving highest mark on army examination
- [,4] Education % education beyond primary school for draftees
- [,5] Catholic % catholic (as opposed to protestant)
- [,6] Infant.Mortality live births who live less than 1 year

All variables but Fertility give proportions of the population.

```
## Loading required package: GGally
## Loading required package: ggplot2
```



Calling lm

```
summary(lm(Fertility ~ .   , data = swiss))
```

```
##                    Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept)      66.9151817 10.70603759  6.250229 1.906051e-07
## Agriculture      -0.1721140  0.07030392 -2.448142 1.872715e-02
## Examination      -0.2580082  0.25387820 -1.016268 3.154617e-01
## Education        -0.8709401  0.18302860 -4.758492 2.430605e-05
## Catholic          0.1041153  0.03525785  2.952969 5.190079e-03
## Infant.Mortality  1.0770481  0.38171965  2.821568 7.335715e-03
```

Example interpretation

- Agriculture is expressed in percentages (0 - 100)
- Estimate is -0.1721.
- Our models estimates an expected 0.17 decrease in standardized fertility for every 1% increase in percentage of males involved in agriculture in holding the remaining variables constant.
- The t-test for $H_0 : \beta_{Agri} = 0$ versus $H_a : \beta_{Agri} \neq 0$ is significant.
- Interestingly, the unadjusted estimate is

```r
summary(lm(Fertility ~ Agriculture, data = swiss))$coefficients
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 60.3043752 4.25125562 14.185074 3.216304e-18
## Agriculture  0.1942017 0.07671176  2.531577 1.491720e-02
```

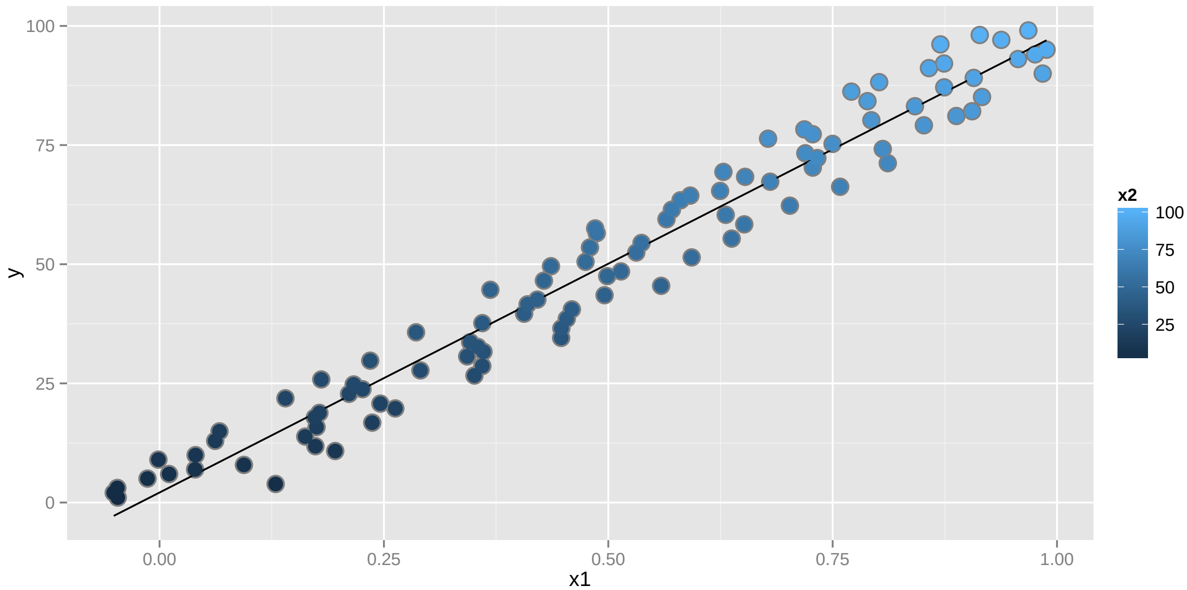
How can adjustment reverse the sign of an effect? Let's try a simulation.

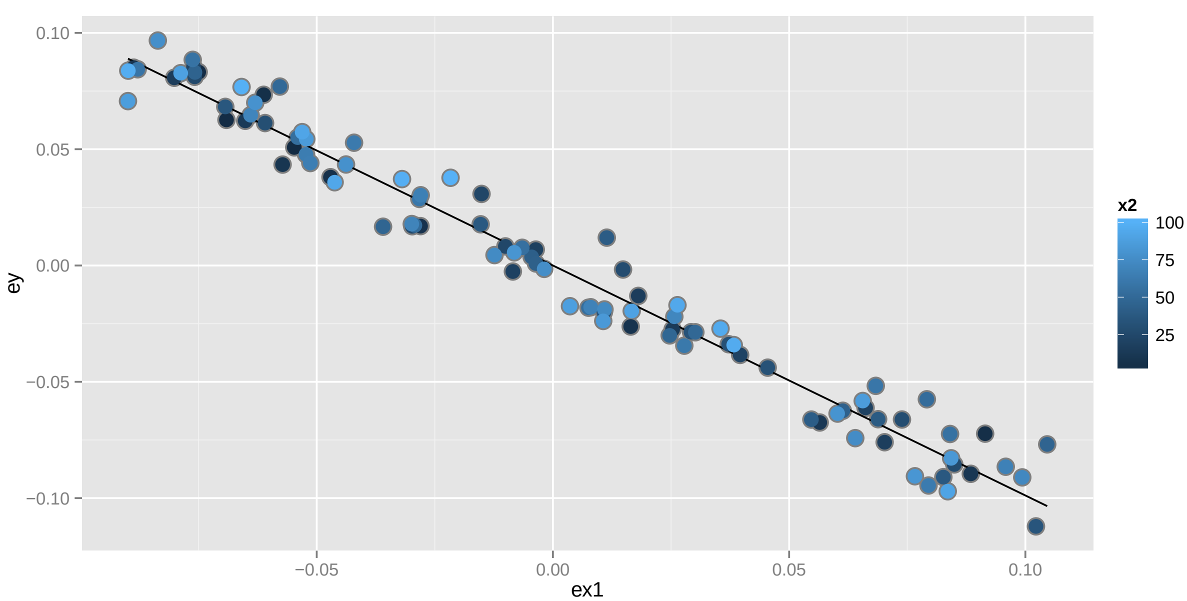
```r
set.seed(3793)
n <- 100; x2 <- 1 : n; x1 <- .01 * x2 + runif(n, -.1, .1); y = -x1 + x2 + rnorm(n, sd = .01)
summary(lm(y ~ x1))$coef
```

```
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)  2.100344   1.128227  1.861634 6.565164e-02
## x1          96.012028   1.951963 49.187423 6.871677e-71
```

```r
summary(lm(y ~ x1 + x2))$coef
```

```
##                 Estimate   Std. Error      t value      Pr(>|t|)
## (Intercept) -0.001781856 0.0019811152   -0.8994205  3.706561e-01
## x1          -0.988683346 0.0172412013  -57.3442261  1.219580e-76
## x2           0.999938050 0.0001743071 5736.6470230 4.742028e-270
```


100
75
50
25
0
y
0.00
0.25
0.50
0.75
1.00
x1
x2
100
75
50
25


0.10
0.05
0.00
−0.05
−0.10
ey
−0.05
0.00
0.05
0.10
ex1
x2
100
75
50
25

What if we include an unnecessary variable?

z adds no new linear information, since it's a linear combination of variables already included. R just drops terms that are linear combinations of other terms.

```r
z <- swiss$Agriculture + swiss$Education
lm(Fertility ~ . + z, data = swiss)
```

```
##
## Call:
## lm(formula = Fertility ~ . + z, data = swiss)
##
## Coefficients:
##      (Intercept)       Agriculture       Examination         Education
##          66.9152           -0.1721           -0.2580           -0.8709
##         Catholic  Infant.Mortality                 z
##           0.1041            1.0770                NA
```

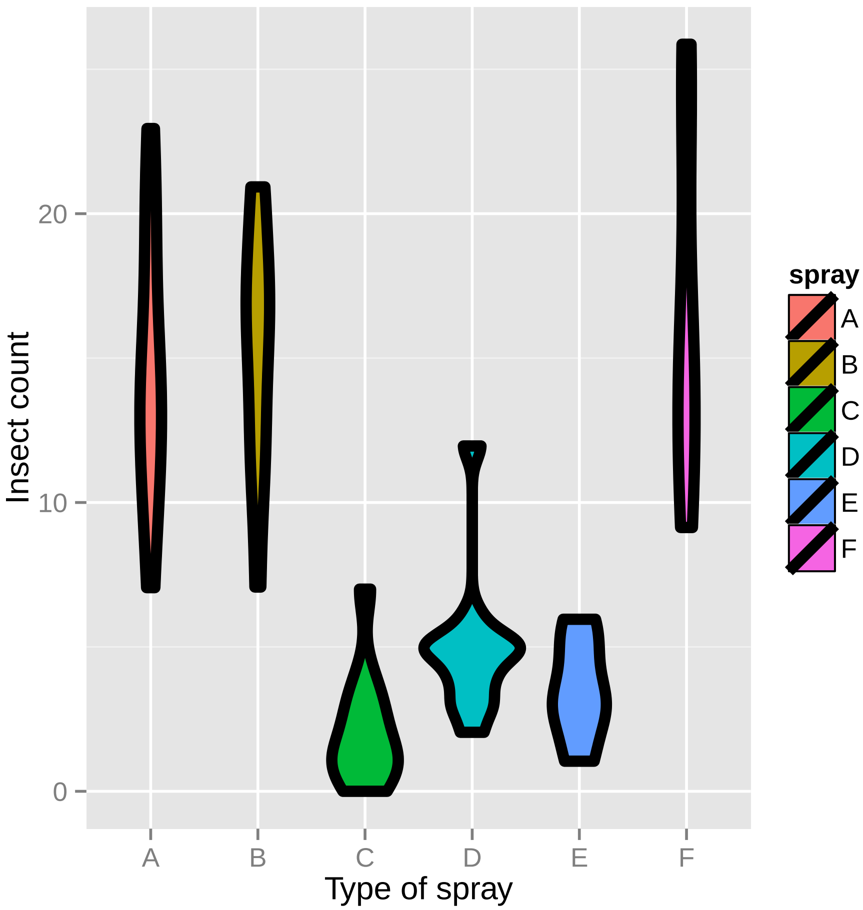
Dummy variables are smart

- Consider the linear model
$$Y_i = \beta_0 + X_{i1}\beta_1 + \epsilon_i$$
where each X_{i1} is binary so that it is a 1 if measurement i is in a group and 0 otherwise. (Treated versus not in a clinical trial, for example.)
- Then for people in the group $E[Y_i] = \beta_0 + \beta_1$
- And for people not in the group $E[Y_i] = \beta_0$
- The LS fits work out to be $\hat\beta_0 + \hat\beta_1$ is the mean for those in the group and $\hat\beta_0$ is the mean for those not in the group.
- β_1 is interpretted as the increase or decrease in the mean comparing those in the group to those not.
- Note including a binary variable that is 1 for those not in the group would be redundant. It would create three parameters to describe two means.

More than 2 levels

- Consider a multilevel factor level. For didactic reasons, let's say a three level factor (example, US political party affiliation: Republican, Democrat, Independent)
- $Y_i = \beta_0 + X_{i1}\beta_1 + X_{i2}\beta_2 + \epsilon_i$.
- X_{i1} is 1 for Republicans and 0 otherwise.
- X_{i2} is 1 for Democrats and 0 otherwise.
- If i is Republican $E[Y_i] = \beta_0 + \beta_1$
- If i is Democrat $E[Y_i] = \beta_0 + \beta_2$.
- If i is Independent $E[Y_i] = \beta_0$.
- β_1 compares Republicans to Independents.
- β_2 compares Democrats to Independents.
- $\beta_1 - \beta_2$ compares Republicans to Democrats.
- (Choice of reference category changes the interpretation.)

Insect Sprays



Linear model fit, group A is the reference

```r
summary(lm(count ~ spray, data = InsectSprays))$coef
```

```
##               Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)  14.5000000   1.132156 12.8074279 1.470512e-19
## sprayB        0.8333333   1.601110  0.5204724 6.044761e-01
## sprayC      -12.4166667   1.601110 -7.7550382 7.266893e-11
## sprayD       -9.5833333   1.601110 -5.9854322 9.816910e-08
## sprayE      -11.0000000   1.601110 -6.8702352 2.753922e-09
## sprayF        2.1666667   1.601110  1.3532281 1.805998e-01
```

Hard coding the dummy variables

```
summary(lm(count ~
             I(1 * (spray == 'B')) + I(1 * (spray == 'C')) +
             I(1 * (spray == 'D')) + I(1 * (spray == 'E')) +
             I(1 * (spray == 'F'))
           , data = InsectSprays))$coef
```

```
##                         Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)           14.5000000   1.132156 12.8074279 1.470512e-19
## I(1 * (spray == "B"))   0.8333333   1.601110  0.5204724 6.044761e-01
## I(1 * (spray == "C")) -12.4166667   1.601110 -7.7550382 7.266893e-11
## I(1 * (spray == "D"))  -9.5833333   1.601110 -5.9854322 9.816910e-08
## I(1 * (spray == "E")) -11.0000000   1.601110 -6.8702352 2.753922e-09
## I(1 * (spray == "F"))   2.1666667   1.601110  1.3532281 1.805998e-01
```

What if we include all 6?

```
summary(lm(count ~
   I(1 * (spray == 'B')) + I(1 * (spray == 'C')) +
   I(1 * (spray == 'D')) + I(1 * (spray == 'E')) +
   I(1 * (spray == 'F')) + I(1 * (spray == 'A')), data = InsectSprays))$coef
```

```
##                         Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)           14.5000000   1.132156 12.8074279 1.470512e-19
## I(1 * (spray == "B"))   0.8333333   1.601110  0.5204724 6.044761e-01
## I(1 * (spray == "C")) -12.4166667   1.601110 -7.7550382 7.266893e-11
## I(1 * (spray == "D"))  -9.5833333   1.601110 -5.9854322 9.816910e-08
## I(1 * (spray == "E")) -11.0000000   1.601110 -6.8702352 2.753922e-09
## I(1 * (spray == "F"))   2.1666667   1.601110  1.3532281 1.805998e-01
```

What if we omit the intercept?

```
summary(lm(count ~ spray - 1, data = InsectSprays))$coef
```

```
##         Estimate Std. Error   t value     Pr(>|t|)
## sprayA 14.500000   1.132156 12.807428 1.470512e-19
## sprayB 15.333333   1.132156 13.543487 1.001994e-20
## sprayC  2.083333   1.132156  1.840148 7.024334e-02
## sprayD  4.916667   1.132156  4.342749 4.953047e-05
## sprayE  3.500000   1.132156  3.091448 2.916794e-03
## sprayF 16.666667   1.132156 14.721181 1.573471e-22
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:GGally':
##
##     nasa
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
summarise(group_by(InsectSprays, spray), mn = mean(count))
```

```
## Source: local data frame [6 x 2]
##
##    spray        mn
##   (fctr)     (dbl)
## 1      A 14.500000
## 2      B 15.333333
## 3      C  2.083333
## 4      D  4.916667
## 5      E  3.500000
## 6      F 16.666667
```

Summary

- If we treat Spray as a factor, R includes an intercept and omits the alphabetically first level of the factor.
- All t-tests are for comparisons of Sprays versus Spray A.
- Emprirical mean for A is the intercept.
- Other group means are the itc plus their coefficient.
- If we omit an intercept, then it includes terms for all levels of the factor.
- Group means are the coefficients.
- Tests are tests of whether the groups are different than zero. (Are the expected counts zero for that spray.)
- If we want comparisons between, Spray B and C, say we could refit the model with C (or B) as the reference level.

Other thoughts on this data

- Counts are bounded from below by 0, violates the assumption of normality of the errors.
- Also there are counts near zero, so both the actual assumption and the intent of the assumption are violated.
- Variance does not appear to be constant.
- Perhaps taking logs of the counts would help.
- There are 0 counts, so maybe log(Count + 1)
- Also, we'll cover Poisson GLMs for fitting count data.

Recall the `swiss` data set

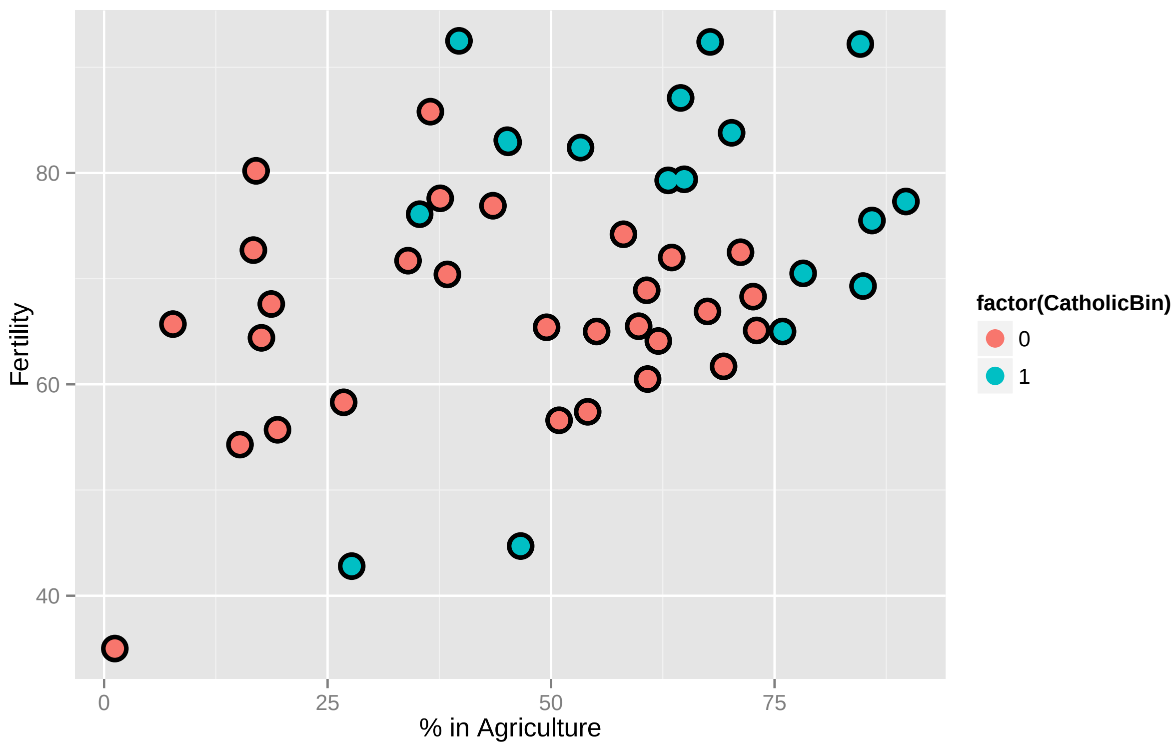
```r
library(datasets); data(swiss)
head(swiss)
```

```
##              Fertility Agriculture Examination Education Catholic
## Courtelary        80.2        17.0          15        12     9.96
## Delemont          83.1        45.1           6         9    84.84
## Franches-Mnt      92.5        39.7           5         5    93.40
## Moutier           85.8        36.5          12         7    33.77
## Neuveville        76.9        43.5          17        15     5.16
## Porrentruy        76.1        35.3           9         7    90.57
##              Infant.Mortality
## Courtelary               22.2
## Delemont                 22.2
## Franches-Mnt             20.2
## Moutier                  20.3
## Neuveville               20.6
## Porrentruy               26.6
```

Create a binary variable

```r
library(dplyr);
swiss = mutate(swiss, CatholicBin = 1 * (Catholic > 50))
```

Plot the data

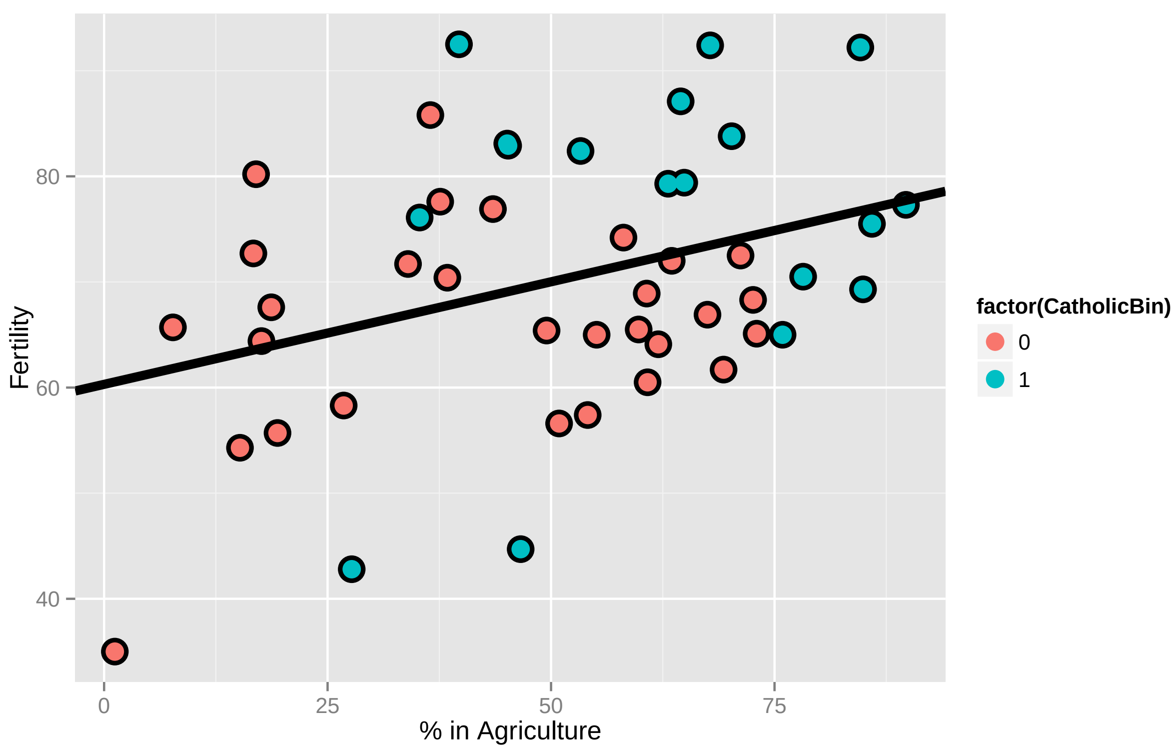


No effect of religion

```
summary(lm(Fertility ~ Agriculture, data = swiss))$coef
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 60.3043752 4.25125562 14.185074 3.216304e-18
## Agriculture  0.1942017 0.07671176  2.531577 1.491720e-02
```

The associated fitted line

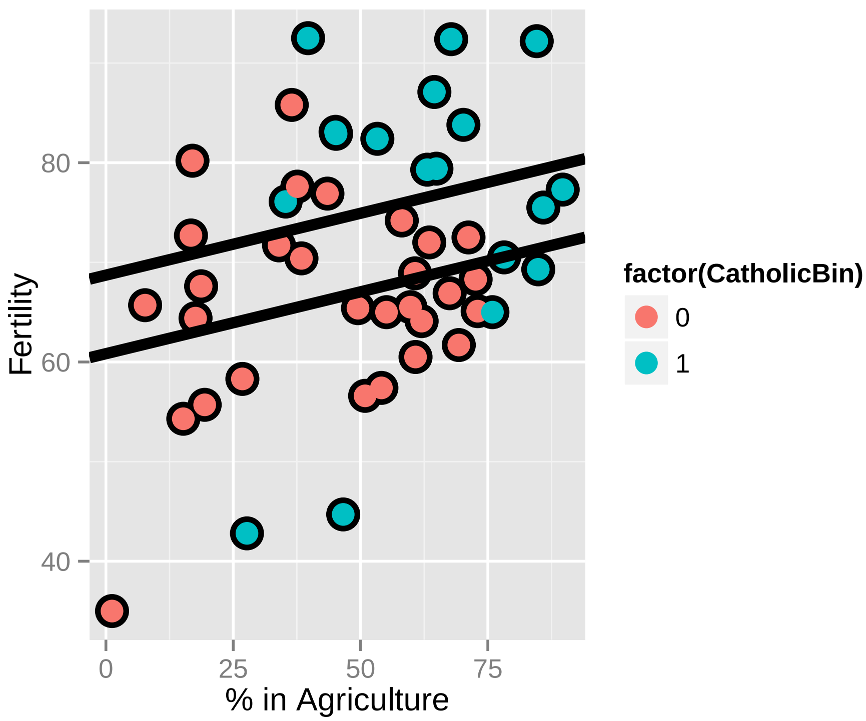


Parallel lines

```r
summary(lm(Fertility ~ Agriculture + factor(CatholicBin), data = swiss))$coef
```

```
##                       Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)         60.8322366  4.1058630 14.815944 1.032493e-18
## Agriculture          0.1241776  0.0810977  1.531210 1.328763e-01
## factor(CatholicBin)1  7.8843292  3.7483622  2.103406 4.118221e-02
```

Fitted lines



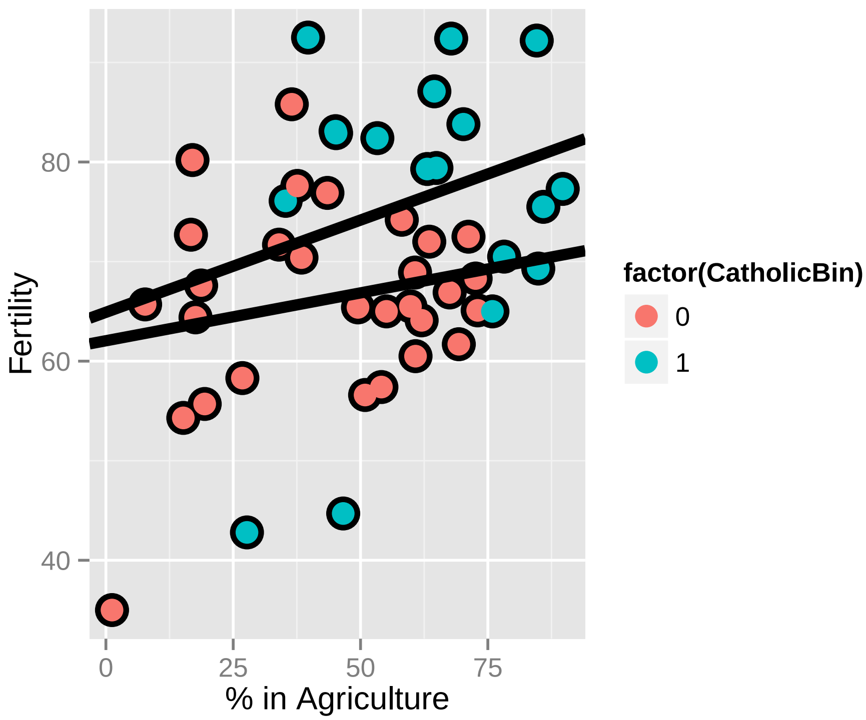
Lines with different slopes and intercepts

```r
summary(lm(Fertility ~ Agriculture * factor(CatholicBin), data = swiss))$coef
```

```
##                                    Estimate  Std. Error    t value
## (Intercept)                     62.04993019  4.78915566 12.9563402
## Agriculture                      0.09611572  0.09881204  0.9727127
## factor(CatholicBin)1             2.85770359 10.62644275  0.2689238
```

```
## Agriculture:factor(CatholicBin)1  0.08913512  0.17610660  0.5061430
##                                      Pr(>|t|)
## (Intercept)                        1.919379e-16
## Agriculture                        3.361364e-01
## factor(CatholicBin)1               7.892745e-01
## Agriculture:factor(CatholicBin)1 6.153416e-01
```

Fitted lines



Just to show you it can be done

```r
summary(lm(Fertility ~ Agriculture + Agriculture : factor(CatholicBin), data = swiss))$coef
```

```
##                                     Estimate Std. Error    t value
## (Intercept)                      62.63037278 4.22989475 14.8066031
## Agriculture                       0.08539357 0.08945287  0.9546209
## Agriculture:factor(CatholicBin)1  0.13339603 0.06198753  2.1519817
##                                      Pr(>|t|)
## (Intercept)                        1.056741e-18
## Agriculture                        3.449849e-01
## Agriculture:factor(CatholicBin)1 3.692561e-02
```