Quantitative Methods in Linguistics – Lecture 11

Adrian Brasoveanu*

April 20, 2014

Contents

This set of lecture notes is primarily based on Wright and London (2009), and Chris Manning's course materials available here `http://www-nlp.stanford.edu/manning/courses/ling289/`.

1 Another logistic regression example: manipulating reasonable doubt

```
> webreg <- "http://www.sagepub.co.uk//wrightandlondon//"
> juryrd <- read.table(paste(webreg, "juryrd.dat", sep = ""), header = T)
> head(juryrd)

  SUBNO GUILTY FORM BELIEF
1     1      1    1   0.75
2     2      1    1   0.70
3     3      1    0   0.90
4     4      0    0   0.20
5     5      0    0   0.75
6     6      0    1   0.45

> str(juryrd)

'data.frame': 172 obs. of  4 variables:
 $ SUBNO : int  1 2 3 4 5 6 7 8 9 10 ...
 $ GUILTY: int  1 1 1 0 0 0 0 1 0 1 ...
 $ FORM  : int  1 1 0 0 0 1 1 1 1 1 ...
 $ BELIEF: num  0.75 0.7 0.9 0.2 0.75 0.45 0.6 0.6 0.8 0.6 ...
```

---

[*] These notes have been generated with the 'knitr' package (Xie 2013) and are based on many sources, including but not limited to: Abelson (1995), Miles and Shevlin (2001), Faraway (2004), De Veaux et al. (2005), Braun and Murdoch (2007), Gelman and Hill (2007), Baayen (2008), Johnson (2008), Wright and London (2009), Gries (2009), Kruschke (2011), Diez et al. (2013), Gries (2013).

```
> attach(juryrd)
```

Participants read a brief crime summary and were asked to render a verdict.

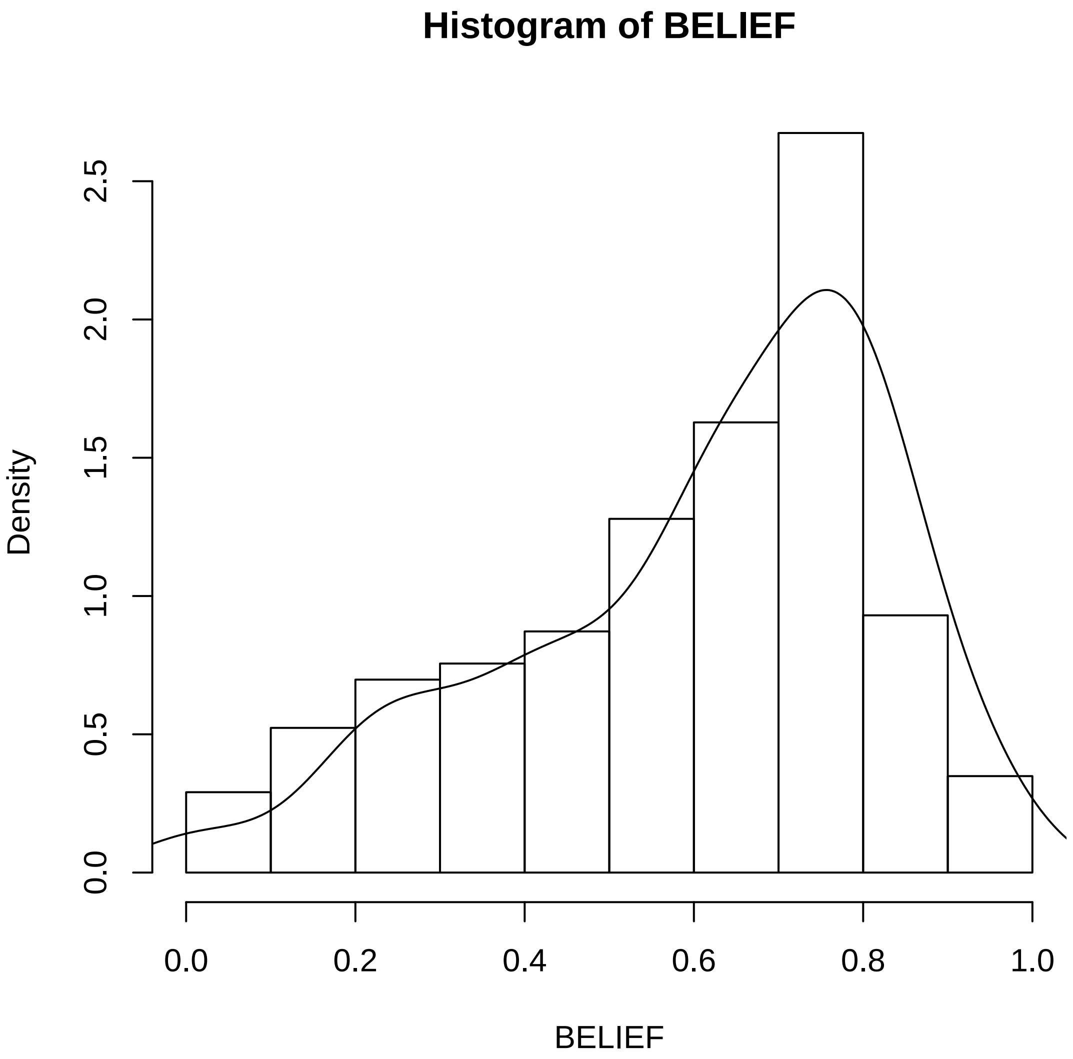
- GUILTY (binary): they were told to tick ’guilty’ (1) if they believed the defendant’s guilt was ’beyond a reasonable doubt
- BELIEF (proportion): they also rated their belief in guilt on a 0.00 to 1.00 probability scale
- FORM (binary): participants were either in a control condition where they received no further instructions or in an experimental condition where they received a more elaborate instruction form

The form was expected to lower belief in guilt and to lower the reasonable doubt threshold.

172 participants made both a belief-in-guilt judgment and a binary verdict (guilty vs. not guilty).

The variable BELIEF is important because we want to see if it is affected by the condition and how it relates to verdict.

```
> hist(BELIEF, freq = FALSE)
> lines(density(BELIEF))
```



BELIEF is negatively skewed.

```
> library("e1071")
> skewness(BELIEF)

[1] -0.7809
```

A rough estimate of the error of skewness is $\sqrt{\frac{6}{n}}$, so the 95% confidence interval is:

```
> lb <- format(skewness(BELIEF) - sqrt(6/length(BELIEF)) * 1.96, digit = 3)
> ub <- format(skewness(BELIEF) + sqrt(6/length(BELIEF)) * 1.96, digit = 3)
> cat(paste("Lower Bound = ", lb, "\t", "Upper Bound = ", ub, "\n",
+     sep = ""))

Lower Bound = -1.15 Upper Bound = -0.415
```

Alternatively, we can calculate bootstrap estimates of this, which are fairly similar to the one above (the BCa intervals, i.e., the adjusted bootstrap intervals, are usually preferred):

```
> library("boot")
> beliefboot <- boot(BELIEF, function(x, i) skewness(x[i]), R = 1000)
> boot.ci(beliefboot)

Warning:  bootstrap variances needed for studentized intervals

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = beliefboot)

Intervals :
Level      Normal              Basic
95%   (-1.0595, -0.5155 )   (-1.0535, -0.4986 )

Level     Percentile            BCa
95%   (-1.0631, -0.5083 )   (-1.1086, -0.5356 )
Calculations and Intervals on Original Scale
```

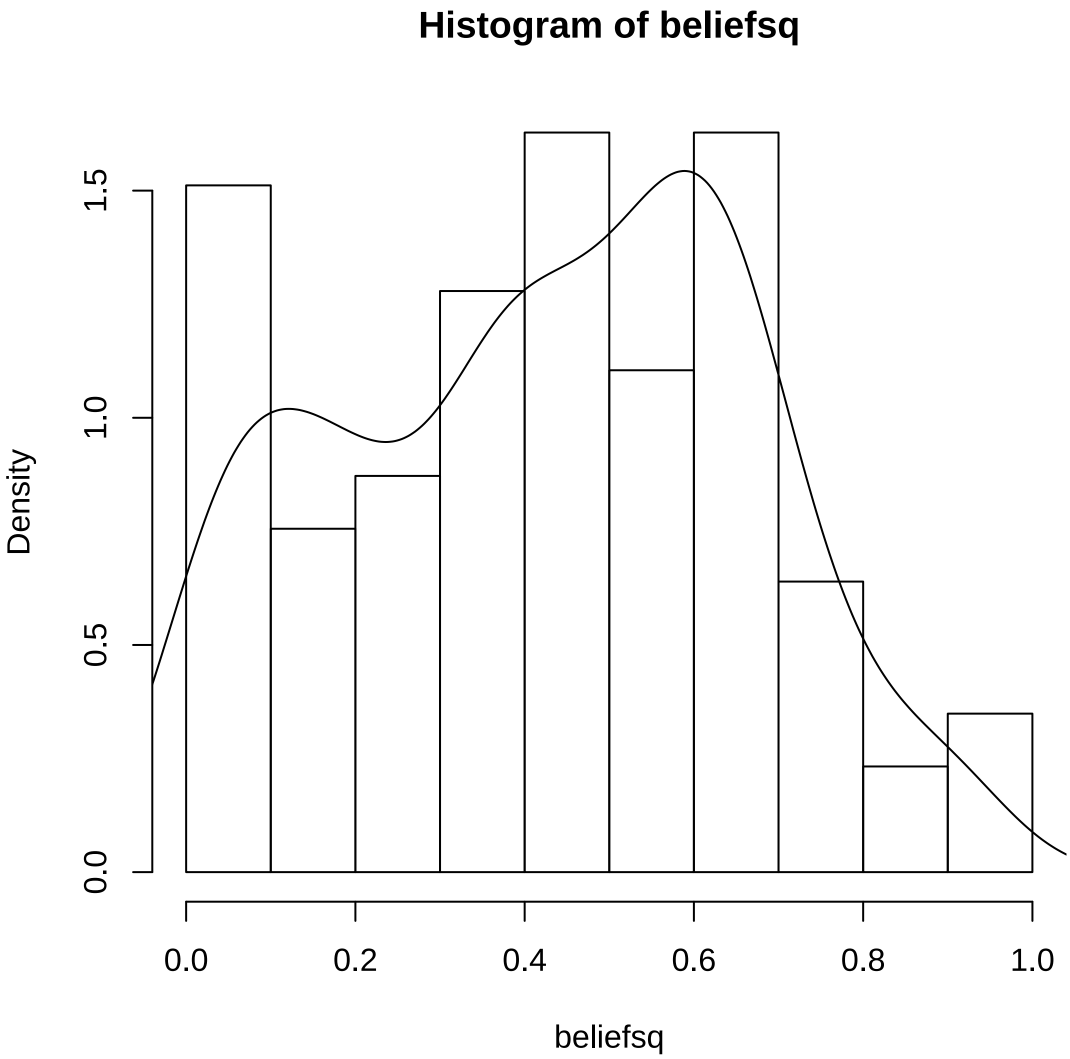
The 95% BCa CI does not overlap with 0, so we can be confident that the variable is skewed.

When data are negatively skewed, we can try squaring the variable.

```
> beliefsq <- BELIEF^2
```

This works fairly well:

```
> hist(beliefsq, freq = FALSE)
> lines(density(beliefsq))
```



The BCa 95% CI overlaps with zero:

```
> bboot2 <- boot(beliefsq, function(x, i) {
+     skewness(x[i])
+ }, R = 1000)
> boot.ci(bboot2)

Warning:  bootstrap variances needed for studentized intervals

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = bboot2)

Intervals :
Level      Normal              Basic
95%   (-0.3088,  0.1485 )   (-0.3155,  0.1575 )
```

```
Level     Percentile            BCa
95%   (-0.3229,  0.1501 )   (-0.3176,  0.1540 )
Calculations and Intervals on Original Scale
```

We run the t-test in regression format, which assumes equal variances (the function `t.test` does not make this assumption):

```
> reg1 <- lm(beliefsq ~ FORM)
> summary(reg1)


Call:
lm(formula = beliefsq ~ FORM)

Residuals:
    Min      1Q  Median      3Q     Max
-0.4618 -0.2118  0.0282  0.1782  0.5157

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.4618     0.0257   17.95   <2e-16 ***
FORM         -0.0750     0.0364   -2.06    0.041 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.239 on 170 degrees of freedom
Multiple R-squared:  0.0244,Adjusted R-squared:  0.0187
F-statistic: 4.25 on 1 and 170 DF,  p-value: 0.0408
```

We reject the H_0: the instruction does affect people's beliefs in guilt.
Compare with a t-test that does not assume equal variances:

```
> t.test(beliefsq ~ FORM)


Welch Two Sample t-test

data:  beliefsq by FORM
t = 2.062, df = 168.2, p-value = 0.04078
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.003183 0.146813
sample estimates:
mean in group 0 mean in group 1
         0.4618          0.3868
```

And a t.test that assumes equal variances:

```
> t.test(beliefsq ~ FORM, var.equal = T)


Two Sample t-test

data:  beliefsq by FORM
```

```
t = 2.062, df = 170, p-value = 0.04076
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.003188 0.146807
sample estimates:
mean in group 0 mean in group 1
         0.4618          0.3868
```

They are not that different because the variances / sd.s of the two groups are very similar:

```
> wbeliefsq <- split(beliefsq, FORM)
> sd(wbeliefsq[[1]])

[1] 0.2504

> sd(wbeliefsq[[2]])

[1] 0.226
```

Now we run the logistic regressions. We begin with the model that has the belief-squared variable as the only predictor for the verdict:

```
> reg2 <- glm(GUILTY ~ beliefsq, binomial)
> summary(reg2)


Call:
glm(formula = GUILTY ~ beliefsq, family = binomial)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.594  -0.663  -0.227   0.724   1.648

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -3.973      0.626   -6.34  2.2e-10 ***
beliefsq       8.092      1.228    6.59  4.4e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 235.08  on 171  degrees of freedom
Residual deviance: 153.66  on 170  degrees of freedom
AIC: 157.7

Number of Fisher Scoring iterations: 5
```

We then add the experimental condition:

```
> reg3 <- glm(GUILTY ~ beliefsq + FORM, binomial)
> summary(reg3)


Call:
```

```
glm(formula = GUILTY ~ beliefsq + FORM, family = binomial)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.527  -0.627  -0.183   0.612   2.120

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -5.644      0.882   -6.40  1.6e-10 ***
beliefsq       9.744      1.475    6.61  3.9e-11 ***
FORM           1.744      0.484    3.61  0.00031 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 235.08  on 171  degrees of freedom
Residual deviance: 138.31  on 169  degrees of freedom
AIC: 144.3

Number of Fisher Scoring iterations: 5
```

ANOVA compares the two models:

```
> anova(reg2, reg3)

Analysis of Deviance Table

Model 1: GUILTY ~ beliefsq
Model 2: GUILTY ~ beliefsq + FORM
  Resid. Df Resid. Dev Df Deviance
1       170        154
2       169        138  1     15.3
```

The p-value for the χ^2 statistic is:

```
> 1 - pchisq(15.351, 1)

[1] 8.927e-05
```

Or in one go:

```
> anova(reg2, reg3, test = "Chi")

Analysis of Deviance Table

Model 1: GUILTY ~ beliefsq
Model 2: GUILTY ~ beliefsq + FORM
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       170        154
2       169        138  1     15.3  8.9e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The values for `reg3`:

```
> round(reg3$coef, 2)

(Intercept)    beliefsq        FORM
      -5.64        9.74        1.74
```

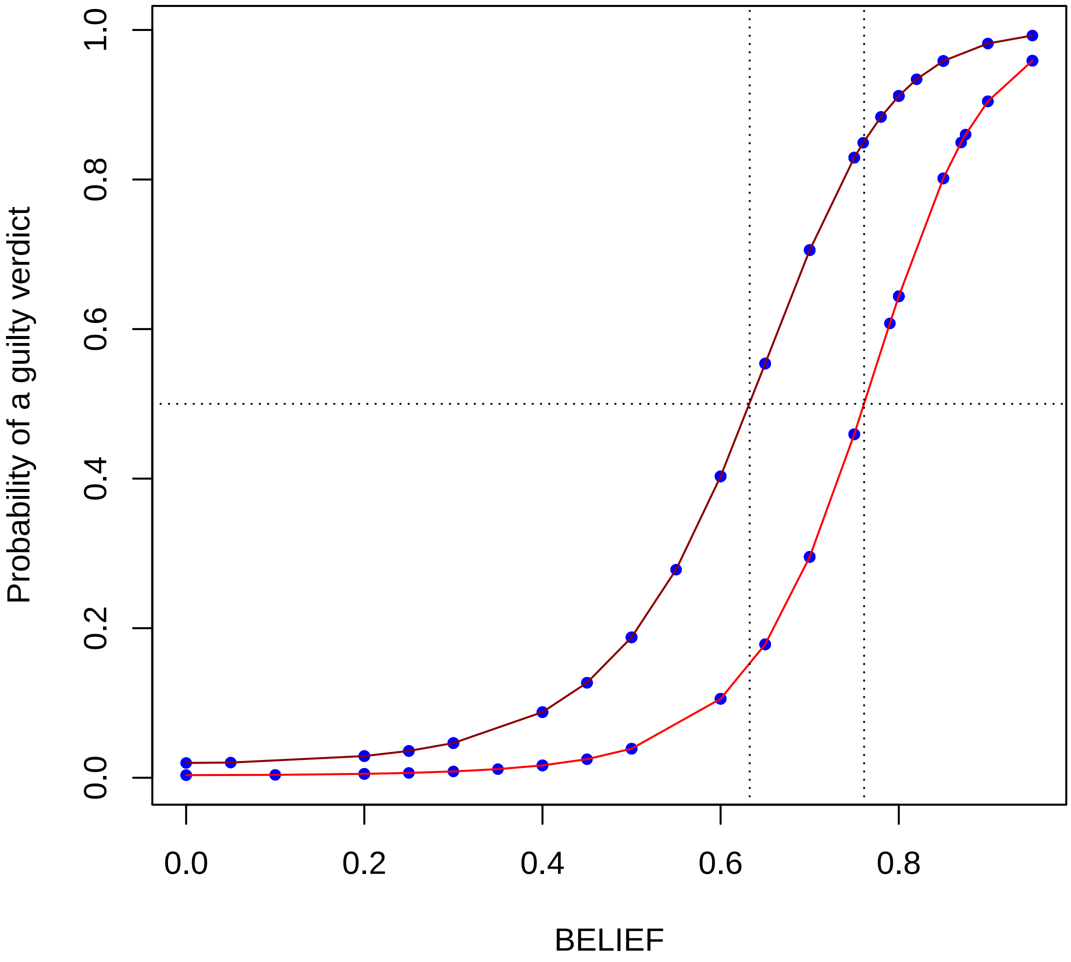
We note that there is a large effect for BELIEFS (as expected): if you believe the person is guilty, you are more likely to deliver a guilty verdict.

There is also an effect for FORM: those in the experimental group have a higher probability of making a guilty verdict once BELIEFSQ has been controlled for.

This can be more clearly seen if we plot the model. We use the original BELIEF variable is used rather than the transformed one because this will be in an easier scale to understand:

- there are 2 groups and we draw separate lines for them, so we split the BELIEF and the prediction variables
- the `lines` function requires the variables to be ordered
- we do this with the `order` function, which records the order of the belief variable for each group
- the `lines` function plots the curves in this order
- we also draw a dashed horizontal line at 50% probability of giving a guilty verdict and vertical lines where this line intersects the curves for the two groups

```
> plot(BELIEF, predict(reg3, type = "response"), ylab = "Probability of a guilty verdict",
+     col = "blue", pch = 20)
> beliefs <- split(BELIEF, FORM)
> predicts <- split(predict(reg3, type = "response"), FORM)
> o1 <- order(beliefs[[1]])
> o2 <- order(beliefs[[2]])
> lines(beliefs[[1]][o1], predicts[[1]][o1], col = "red")
> lines(beliefs[[2]][o2], predicts[[2]][o2], col = "darkred")
> abline(h = 0.5, lty = 3)
> ld1 <- sqrt(-(reg3$coef[1]/reg3$coef[2]))
> ld2 <- sqrt(-((reg3$coef[1] + reg3$coef[3])/reg3$coef[2]))
> cat("Control 50% = ", format(ld1, dig = 2), "\t", "Instruction 50% = ",
+     format(ld2, dig = 2), "\n", sep = "")

Control 50% = 0.76 Instruction 50% = 0.63

> abline(v = ld1, lty = 3)
> abline(v = ld2, lty = 3)
```



For completeness, we check the interaction between BELIEFSQ and FORM:

```
> anova(reg3, glm(GUILTY ~ FORM * beliefsq, binomial), test = "Chi")

Analysis of Deviance Table

Model 1: GUILTY ~ beliefsq + FORM
Model 2: GUILTY ~ FORM * beliefsq
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       169        138
2       168        133  1     5.13    0.024 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The interaction is significant:

```
> summary(glm(GUILTY ~ FORM * beliefsq, binomial))


Call:
glm(formula = GUILTY ~ FORM * beliefsq, family = binomial)

Deviance Residuals: 
   Min      1Q  Median      3Q     Max  
-2.601  -0.608  -0.156   0.619   1.886  

Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept)     -4.113      0.944   -4.36  1.3e-05 ***
FORM            -1.554      1.578   -0.98    0.325    
beliefsq         6.997      1.641    4.26  2.0e-05 ***
FORM:beliefsq    7.089      3.403    2.08    0.037 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 235.08  on 171  degrees of freedom
Residual deviance: 133.18  on 168  degrees of freedom
AIC: 141.2

Number of Fisher Scoring iterations: 6
```

The slope is steeper for the instruction group:

- one interpretation: the variance in the interpretation of reasonable doubt is less for this group
- another interpretation: the effect exists for values of BELIEF above about 0.4

```
> detach(juryrd)
```

2 Simulating datasets for logistic regression

The nature of the random error in logistic regression models can be further clarified by simulating a dataset for logistic regression.

We start by generating the predictor values:

```
> x <- rnorm(1000)
```

We set the coefficient of the predictor:

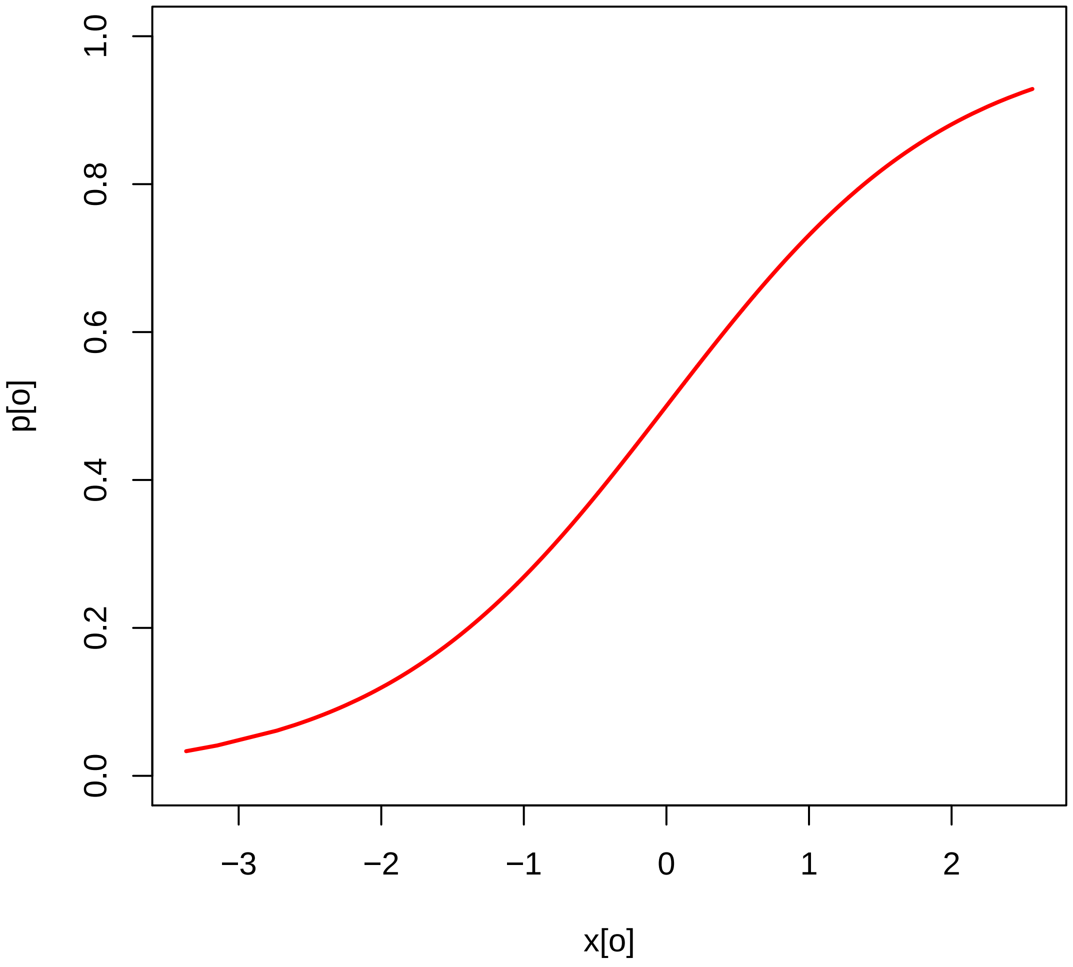
```
> beta.x <- 1
```

We obtain the probabilities:

```
> p <- 1/(1 + exp(-beta.x * x))
```

We plot the predictor and the probabilities:

```
> o <- order(x)
> plot(x[o], p[o], ylim = c(0, 1), type = "l", col = "red", lwd = 2)
```

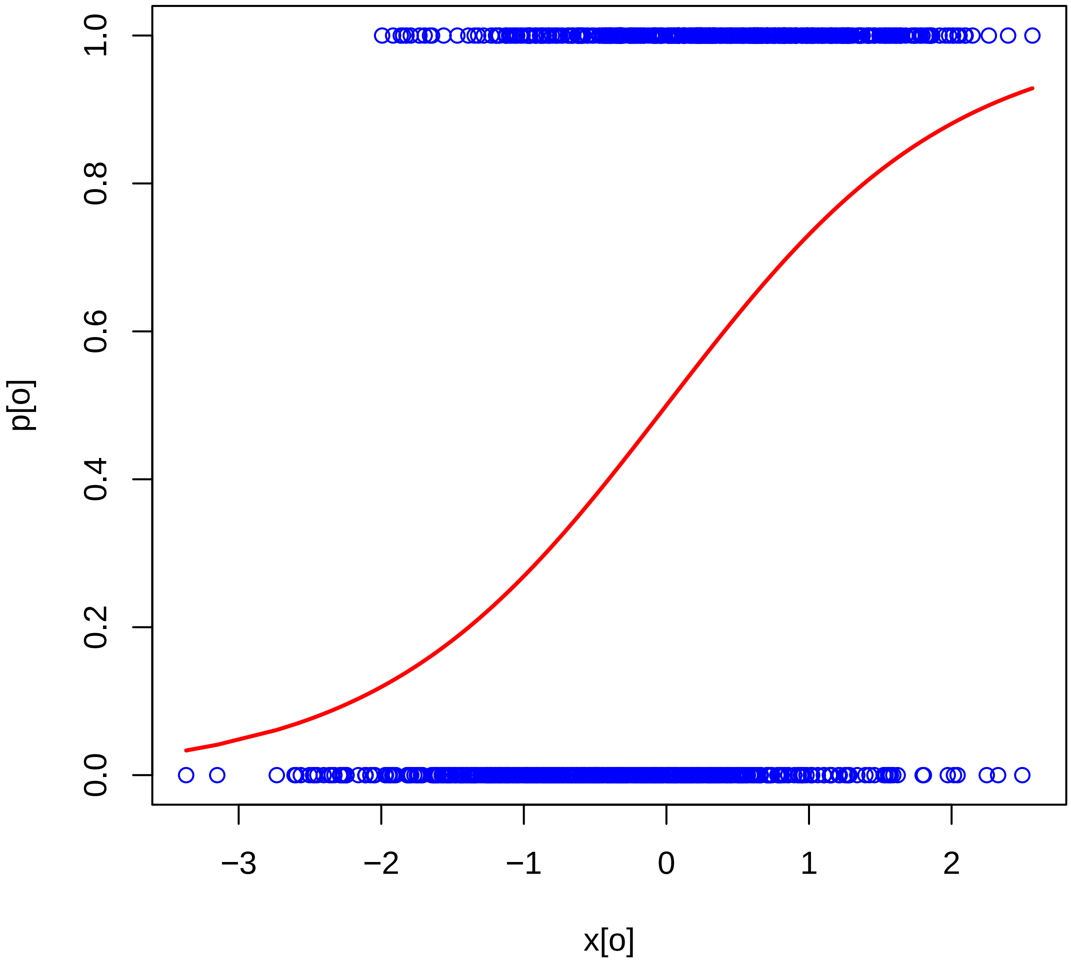


We generate the response values based on the vector of probabilities:

```
> y <- rbinom(1000, 1, prob = p)
```

This is a plot of the dataset superimposed on the above plot of predictor values and corresponding probabilities:

```
> plot(x[o], p[o], ylim = c(0, 1), type = "l", col = "red", lwd = 2)
> points(jitter(x[o]), y[o], col = "blue")
```



This is the logistic regression model:

```
> m1 <- glm(y ~ x, family = binomial)
> summary(m1)


Call:
glm(formula = y ~ x, family = binomial)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.172  -0.979  -0.546   1.016   2.066

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.1130     0.0695   -1.63      0.1
x             0.9503     0.0800   11.89   <2e-16 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1380.8  on 999  degrees of freedom
Residual deviance: 1202.5  on 998  degrees of freedom
AIC: 1207

Number of Fisher Scoring iterations: 3
```

The model without an intercept:

```
> m2 <- glm(y ~ -1 + x, family = binomial)
> summary(m2)


Call:
glm(formula = y ~ -1 + x, family = binomial)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.223  -1.023  -0.573   0.971   2.021

Coefficients:
  Estimate Std. Error z value Pr(>|z|)
x   0.9545     0.0798      12   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1386.3  on 1000  degrees of freedom
Residual deviance: 1205.1  on  999  degrees of freedom
AIC: 1207

Number of Fisher Scoring iterations: 4
```

We test for the significance of the intercept:

```
> anova(m1, m2, test = "Chi")

Analysis of Deviance Table

Model 1: y ~ x
Model 2: y ~ -1 + x
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       998       1203
2       999       1205 -1    -2.65      0.1
```

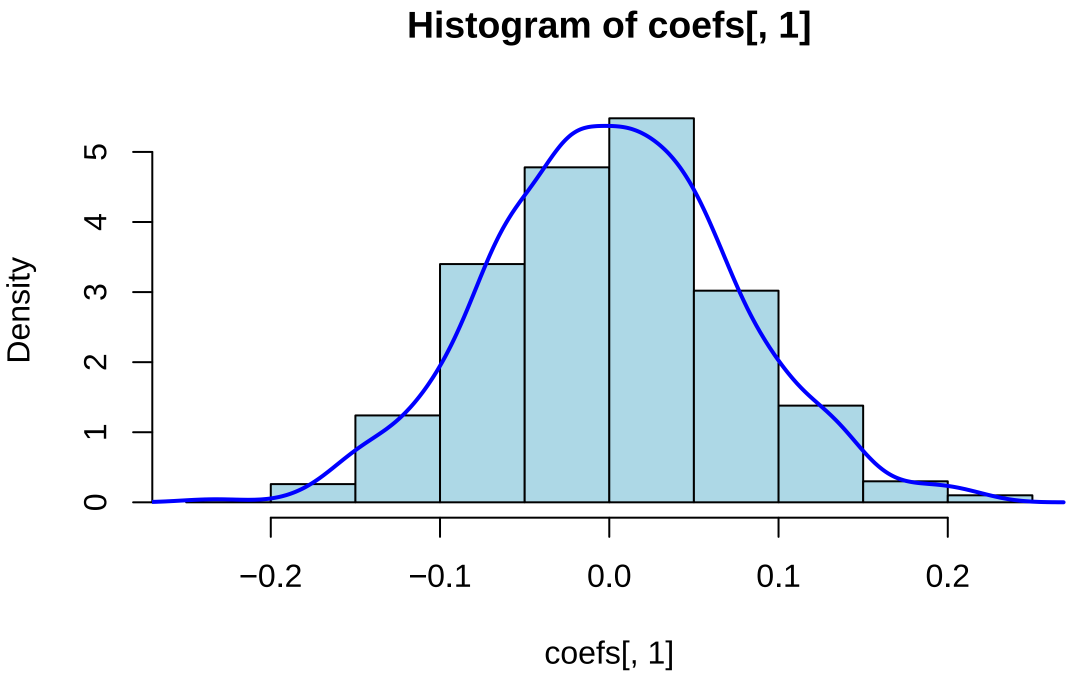
And here are the logistic regression coefficients for 1000 simulations of the response variable y (assuming the same x and p as above):

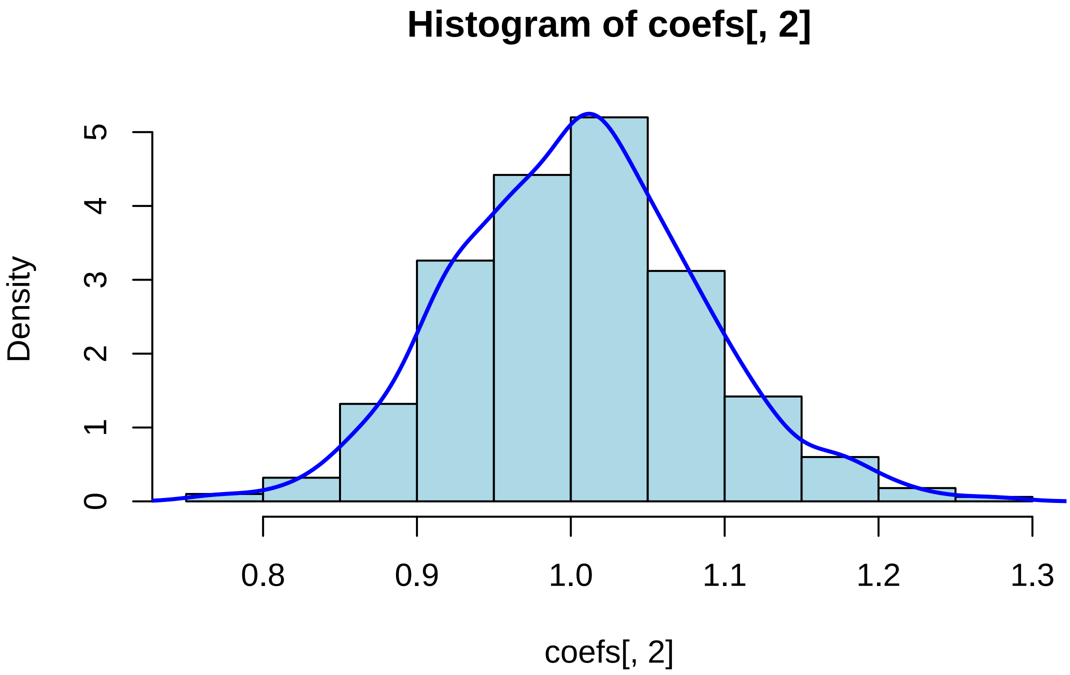
```
> no.of.sims <- 1000
> coefs <- matrix(0, no.of.sims, 2)
> str(coefs)
```

```
 num [1:1000, 1:2] 0 0 0 0 0 0 0 0 0 0 ...

> head(coefs)

     [,1] [,2]
[1,]    0    0
[2,]    0    0
[3,]    0    0
[4,]    0    0
[5,]    0    0
[6,]    0    0

> for (i in 1:no.of.sims) {
+     coefs[i, ] <- coef(glm(rbinom(1000, 1, prob = p) ~ x, family = binomial))
+ }
```

The histograms of the coefficients are centered at the true values:

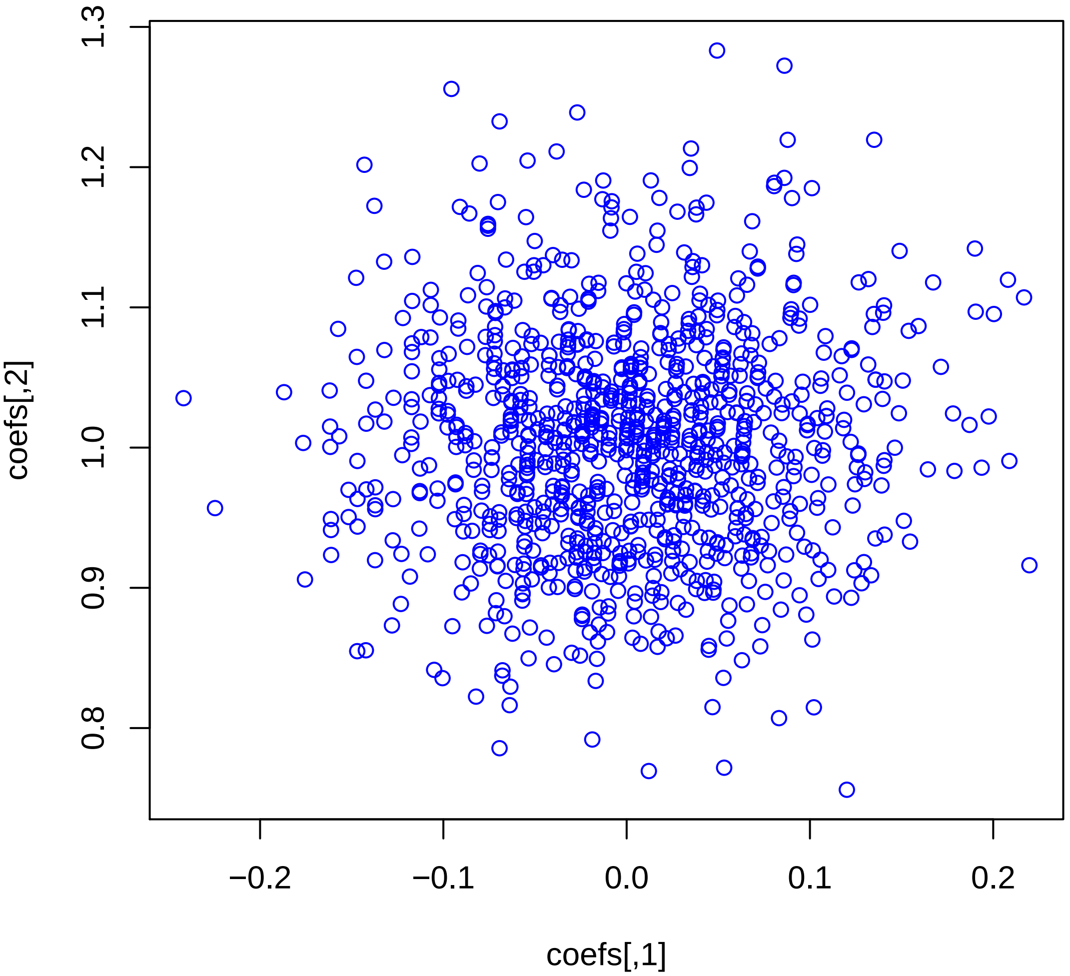
```
> par(mfrow = c(2, 1))
> hist(coefs[, 1], col = "lightblue", freq = FALSE)
> lines(density(coefs[, 1]), col = "blue", lwd = 2)
> hist(coefs[, 2], col = "lightblue", freq = FALSE)
> lines(density(coefs[, 2]), col = "blue", lwd = 2)
```





```
> par(mfrow = c(1, 1))
```

```
> plot(coefs, col = "blue")
```



3 A linguistic example: /*s*/-deletion in Panamanian Spanish

This section is partly based on materials available at http://www-nlp.stanford.edu/manning/courses/ling289/. The data is from Henrietta Cedergren's 1973 study of /*s*/-deletion in Panamanian Spanish (Panama City, Panama).

```
> cedegren <- read.table("http://www-nlp.stanford.edu/manning/courses/ling289/cedegren.txt",
+     header = T)
> head(cedegren)

  sDel sNoDel cat follows class
1   38    120   m       C     1
2    0      6   v       C     1
3   18     90   d       C     1
```

```
4    7     44   a       C     1
5   34     40   n       C     1
6   30     42   m       V     1

> str(cedegren)

'data.frame': 52 obs. of  5 variables:
 $ sDel   : int  38 0 18 7 34 30 0 2 5 20 ...
 $ sNoDel : int  120 6 90 44 40 42 5 23 8 1 ...
 $ cat    : Factor w/ 5 levels "a","d","m","n",..: 3 5 2 1 4 3 5 2 1 4 ...
 $ follows: Factor w/ 3 levels "C","P","V": 1 1 1 1 1 3 3 3 3 3 ...
 $ class  : int  1 1 1 1 1 1 1 1 1 1 ...
```

Speakers in Panama City, like in many dialects of Spanish, variably delete the *s* at the end of words. Cedegren undertook a study to find out if there was a change in progress: if final /*s*/ was systematically dropping out of Panamanian Spanish.

/*s*/-deletion counts:

```
> cedegren$sDel

 [1]  38   0  18   7  34  30   0   2   5  20   7   0   4 377  10 160  66
[18] 232 233   8  24  15 141  98  12 138 358   9 120  45 183 215   3  30
[35]  15  80 145  12 105 582   4 226  87 284 252   5  24  10 135 221  24
[52] 238
```

Corresponding non-deletion counts:

```
> cedegren$sNoDel

 [1] 120   6  90  44  40  42   5  23   8   1  12   8  15 349  21 218  60
[18] 129 120  24  32   9   7  27   3  34 345   6 173  81 143 123   2  27
[35]   6  56  48   9  46 547   9 197  76 106 112   4  29   5  60  59   2
[52]  37
```

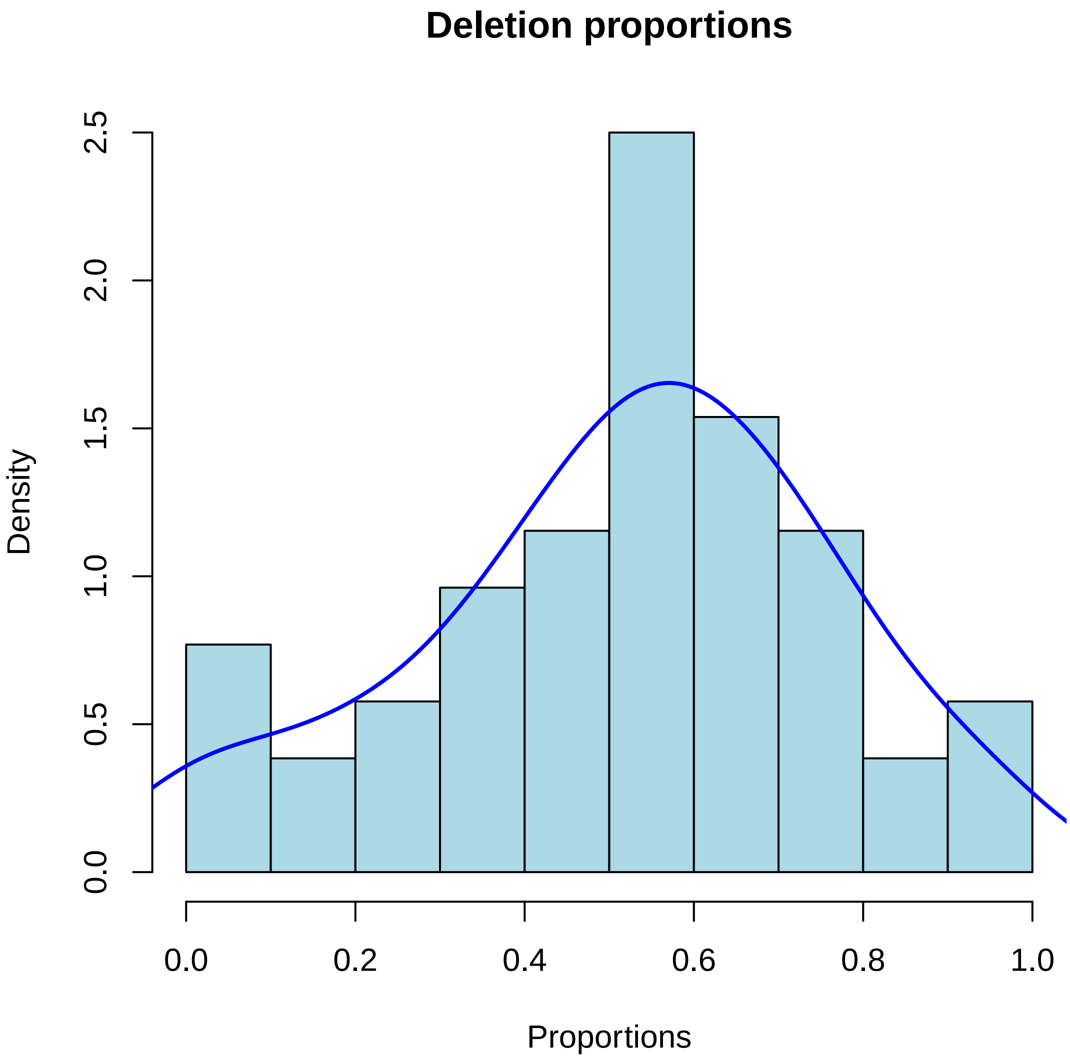
Both together:

```
> cbind(cedegren$sDel, cedegren$sNoDel)

      [,1] [,2]
 [1,]   38  120
 [2,]    0    6
 [3,]   18   90
 [4,]    7   44
 [5,]   34   40
 [6,]   30   42
 [7,]    0    5
 [8,]    2   23
 [9,]    5    8
[10,]   20    1
[11,]    7   12
[12,]    0    8
[13,]    4   15
[14,]  377  349
[15,]   10   21
[16,]  160  218
```

```
[17,]   66   60
[18,]  232  129
[19,]  233  120
[20,]    8   24
[21,]   24   32
[22,]   15    9
[23,]  141    7
[24,]   98   27
[25,]   12    3
[26,]  138   34
[27,]  358  345
[28,]    9    6
[29,]  120  173
[30,]   45   81
[31,]  183  143
[32,]  215  123
[33,]    3    2
[34,]   30   27
[35,]   15    6
[36,]   80   56
[37,]  145   48
[38,]   12    9
[39,]  105   46
[40,]  582  547
[41,]    4    9
[42,]  226  197
[43,]   87   76
[44,]  284  106
[45,]  252  112
[46,]    5    4
[47,]   24   29
[48,]   10    5
[49,]  135   60
[50,]  221   59
[51,]   24    2
[52,]  238   37
```

Deletion proportions:

```
> del.props <- cedegren$sDel/(cedegren$sDel + cedegren$sNoDel)
> round(del.props, 2)

 [1] 0.24 0.00 0.17 0.14 0.46 0.42 0.00 0.08 0.38 0.95 0.37 0.00 0.21 0.52
[15] 0.32 0.42 0.52 0.64 0.66 0.25 0.43 0.62 0.95 0.78 0.80 0.80 0.51 0.60
[29] 0.41 0.36 0.56 0.64 0.60 0.53 0.71 0.59 0.75 0.57 0.70 0.52 0.31 0.53
[43] 0.53 0.73 0.69 0.56 0.45 0.67 0.69 0.79 0.92 0.87

> summary(del.props)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.381   0.534   0.518   0.692   0.953

> hist(del.props, freq = FALSE, main = "Deletion proportions", xlab = "Proportions",
+     col = "lightblue")
> lines(density(del.props), col = "blue", lwd = 2)
```



The data: interviews she performed across the city in several different social classes to see how the variation was structured in the community.

- social class: 1=highest, 2=second highest, 3=second lowest, 4=lowest

```
> (cedegren$class <- as.factor(cedegren$class))

 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3
[36] 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4
Levels: 1 2 3 4
```

She also investigated 2 linguistic constraints on deletion.

- a phonetic constraint: whether the following segment was consonant, vowel or pause

```
> cedegren$follows

 [1] C C C C C V V V V V P P P C C C C C V V V V V P P P C C C C C V V V V
[36] V P P P C C C C C V V V V V P P P
Levels: C P V
```

- whether the grammatical category of the word that the /*s*/ is part of is significant
  - monomorpheme, where the *s* is part of the free morpheme (e.g., *menos*)
  - verb, where the *s* is the second singular inflection (e.g., *tu tienes*)
  - determiner, where *s* is plural marked on a determiner (e.g., *los*, *las*)
  - adjective, where *s* is a nominal plural agreeing with the noun (e.g., *buenos*)
  - noun, where *s* marks a plural noun (e.g., *amigos*)

```
> cedegren$cat

 [1] m v d a n m v d a n m a n m v d a n m v d a n m a n m v d a n m v d a
[36] n m a n m v d a n m v d a n m a n
Levels: a d m n v
```

Goal: identify the conditions under which /*s*/ is deleted most often.

```
> attach(cedegren)

The following objects are masked from ced.long (position 3):

    cat, class, follows, sDel
The following objects are masked from cedegren (position 4):

    cat, class, follows, sDel, sNoDel
The following objects are masked from ced.long (position 6):

    cat, class, follows, sDel
The following objects are masked from cedegren (position 7):

    cat, class, follows, sDel, sNoDel
The following objects are masked from ced.long (position 8):

    cat, class, follows, sDel
The following objects are masked from cedegren (position 9):

    cat, class, follows, sDel, sNoDel

> contrasts(cat)

  d m n v
a 0 0 0 0
d 1 0 0 0
m 0 1 0 0
n 0 0 1 0
v 0 0 0 1

> contrasts(follows)
```

```
  P V
C 0 0
P 1 0
V 0 1

> contrasts(class)

  2 3 4
1 0 0 0
2 1 0 0
3 0 1 0
4 0 0 1

> ced.del <- cbind(sDel, sNoDel)
> m1 <- glm(ced.del ~ cat + follows + class, binomial)
> print(summary(m1), dig = 1)


Call:
glm(formula = ced.del ~ cat + follows + class, family = binomial)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
    -3      -1       0       1       6

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    -1.32       0.12     -11   <2e-16 ***
catd           -0.17       0.10      -2     0.09 .
catm            0.18       0.09       2     0.05 *
catn            0.67       0.10       7    5e-12 ***
catv           -0.77       0.22      -4    4e-04 ***
followsP        0.95       0.07      13   <2e-16 ***
followsV        0.53       0.06       9   <2e-16 ***
class2          1.27       0.10      12   <2e-16 ***
class3          1.05       0.10      10   <2e-16 ***
class4          1.37       0.10      14   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 958.66  on 51  degrees of freedom
Residual deviance: 198.63  on 42  degrees of freedom
AIC: 446.1

Number of Fisher Scoring iterations: 4
```

The high residual deviance shows that the model is not likely to have generated the data – this deviance is significantly greater for m1 compared to the saturated model (with 52 parameters):

```
> 1 - pchisq(198.63, 42)

[1] 0
```

But it certainly fits the data better than the null model in which a fixed mean probability of deletion (i.e., the intercept) is used for all cells / responses:

```
> 1 - pchisq(958.66 - 198.63, 9)

[1] 0
```

Equivalently:

```
> anova(glm(ced.del ~ 1, binomial), m1, test = "Chi")

Analysis of Deviance Table

Model 1: ced.del ~ 1
Model 2: ced.del ~ cat + follows + class
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        51        959
2        42        199  9      760   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can calculate the likelihood, loglikelihood and deviances for m1 and the null / intercept-only model by hand.

We start with predicted probabilities of SDELETION for each cell, and turn them into log probabilities:

```
> (probsdel <- fitted(m1))

     1      2      3      4      5      6      7      8      9     10
0.2424 0.1105 0.1843 0.2111 0.3426 0.3531 0.1748 0.2782 0.3134 0.4707
    11     12     13     14     15     16     17     18     19     20
0.4534 0.4096 0.5747 0.5326 0.3067 0.4459 0.4880 0.6500 0.6603 0.4301
    21     22     23     24     25     26     27     28     29     30
0.5786 0.6192 0.7601 0.7471 0.7119 0.8280 0.4771 0.2616 0.3919 0.4329
    31     32     33     34     35     36     37     38     39     40
0.5978 0.6088 0.3767 0.5236 0.5656 0.7172 0.7028 0.6643 0.7940 0.5584
    41     42     43     44     45     46     47     48     49     50
0.3293 0.4717 0.5140 0.6732 0.6832 0.4557 0.6037 0.6434 0.7785 0.7662
    51     52
0.7327 0.8423

> (logprobdel <- log(probsdel))

      1       2       3       4       5       6       7       8       9
-1.4173 -2.2029 -1.6913 -1.5554 -1.0711 -1.0411 -1.7439 -1.2795 -1.1602
     10      11      12      13      14      15      16      17      18
-0.7536 -0.7911 -0.8926 -0.5539 -0.6299 -1.1817 -0.8076 -0.7173 -0.4308
     19      20      21      22      23      24      25      26      27
-0.4150 -0.8436 -0.5472 -0.4793 -0.2744 -0.2915 -0.3398 -0.1888 -0.7400
     28      29      30      31      32      33      34      35      36
-1.3410 -0.9369 -0.8374 -0.5144 -0.4962 -0.9764 -0.6470 -0.5699 -0.3324
     37      38      39      40      41      42      43      44      45
-0.3526 -0.4091 -0.2307 -0.5827 -1.1109 -0.7514 -0.6655 -0.3957 -0.3809
     46      47      48      49      50      51      52
-0.7858 -0.5047 -0.4410 -0.2504 -0.2663 -0.3110 -0.1717

> (logprobnodel <- log(1 - probsdel))
```

```
      1       2       3       4       5       6       7       8       9
-0.2776 -0.1171 -0.2037 -0.2371 -0.4195 -0.4355 -0.1922 -0.3260 -0.3760
     10      11      12      13      14      15      16      17      18
-0.6361 -0.6039 -0.5269 -0.8549 -0.7607 -0.3664 -0.5905 -0.6695 -1.0497
     19      20      21      22      23      24      25      26      27
-1.0798 -0.5624 -0.8641 -0.9655 -1.4273 -1.3748 -1.2445 -1.7602 -0.6484
     28      29      30      31      32      33      34      35      36
-0.3032 -0.4973 -0.5671 -0.9109 -0.9386 -0.4727 -0.7415 -0.8338 -1.2630
     37      38      39      40      41      42      43      44      45
-1.2135 -1.0914 -1.5797 -0.8173 -0.3994 -0.6381 -0.7215 -1.1184 -1.1496
     46      47      48      49      50      51      52
-0.6083 -0.9255 -1.0311 -1.5072 -1.4534 -1.3195 -1.8469
```

We then work out the total data loglikelihood in log space by summing the vectors of log expected counts.

```
> (loglike <- sum(sDel * logprobdel) + sum(sNoDel * logprobnodel))

[1] -5650
```

The actual likelihood is too small to be accurately represented:

```
> exp(loglike)

[1] 0
```

This is one of the important reasons for using log-likelihoods: avoid large-impact rounding errors when doing numerical computations with very small (or very large) numbers. Another reason is that they simplify the symbolic calculations that we, as probabilistic-model designers, have to do.

We can now also work out the loglikelihood of the null model, which just uses the overall rate of SDELETION in the data:

```
> sum(sDel)/(sum(sDel) + sum(sNoDel))

[1] 0.5755

> fitted(glm(ced.del ~ 1, binomial))

     1      2      3      4      5      6      7      8      9     10
0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755
    11     12     13     14     15     16     17     18     19     20
0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755
    21     22     23     24     25     26     27     28     29     30
0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755
    31     32     33     34     35     36     37     38     39     40
0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755
    41     42     43     44     45     46     47     48     49     50
0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755 0.5755
    51     52
0.5755 0.5755

> (sDelProb <- sum(sDel)/(sum(sDel) + sum(sNoDel)))

[1] 0.5755

> (loglikenull <- sum(sDel) * log(sDelProb) + sum(sNoDel) * log(1 -
+     sDelProb))

[1] -6030
```

This likelihood is also too small to be represented:

```
> exp(loglikenull)

[1] 0
```

G^2, i.e., the deviance, is just $-2\times$ the difference in log likelihoods:

```
> (Gsq <- -2 * (loglikenull - loglike))

[1] 760

> anova(glm(ced.del ~ 1, binomial), m1, test = "Chi")$Deviance[2]

[1] 760
```

We cannot drop any predictor, as shown by both `anova` and `drop1`:

```
> anova(m1, test = "Chi")

Analysis of Deviance Table

Model: binomial, link: logit

Response: ced.del

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                       51        959
cat      4      315        47        644   <2e-16 ***
follows  2      229        45        415   <2e-16 ***
class    3      216        42        199   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> drop1(m1, test = "Chi")

Single term deletions

Model:
ced.del ~ cat + follows + class
        Df Deviance AIC LRT Pr(>Chi)
<none>          199 446
cat      4      369 608 170   <2e-16 ***
follows  2      425 668 226   <2e-16 ***
class    3      415 656 216   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA test tries adding the factors only in the order given in the model formula (left to right). If certain comparisons are close, we need to reorder the predictors in the model formula ordering or we use drop1.

3.1 Exploring smaller models

Classes 2 and 4 seem to have somewhat similar effects, so we might try modelling class as a 3-way distinction:

```
> m2 <- glm(ced.del ~ cat + follows + I(class == 1) + I(class == 3),
+     binomial)
> summary(m2)


Call:
glm(formula = ced.del ~ cat + follows + I(class == 1) + I(class ==
    3), family = binomial)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-3.253   -1.385    0.007    1.067    6.168

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        0.00984    0.08605    0.11   0.9090
catd              -0.17072    0.10030   -1.70   0.0888 .
catm               0.18187    0.08949    2.03   0.0421 *
catn               0.66614    0.09650    6.90  5.1e-12 ***
catv              -0.78945    0.21823   -3.62   0.0003 ***
followsP           0.95749    0.07394   12.95  < 2e-16 ***
followsV           0.52965    0.05654    9.37  < 2e-16 ***
I(class == 1)TRUE -1.32823    0.09846  -13.49  < 2e-16 ***
I(class == 3)TRUE -0.28090    0.05068   -5.54  3.0e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 958.66  on 51  degrees of freedom
Residual deviance: 202.08  on 43  degrees of freedom
AIC: 447.5

Number of Fisher Scoring iterations: 4
```

Moreover, the deviance for this smaller model is not significantly greater than for model m1, so (as far as statistical significance goes) we might want to keep this model and abandon m1:

```
> anova(m2, m1, test = "Chi")

Analysis of Deviance Table

Model 1: ced.del ~ cat + follows + I(class == 1) + I(class == 3)
Model 2: ced.del ~ cat + follows + class
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        43        202
2        42        199  1     3.45    0.063 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

But class is a scale (an ordinal variable) – and collapsing together classes 2 and 4 does not make much sense, so we put m2 aside.

We could try modelling class as a 2-way distinction, but the resulting increase in deviance is significant, so we do not prefer m2.1 to m1:

```
> m2.1 <- glm(ced.del ~ cat + follows + I(class == 1), binomial)
> summary(m2.1)


Call:
glm(formula = ced.del ~ cat + follows + I(class == 1), family = binomial)

Deviance Residuals:
   Min       1Q  Median       3Q      Max
-4.211  -1.317  -0.083   1.022   6.572

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        -0.0757     0.0845   -0.90  0.37046
catd               -0.1614     0.1001   -1.61  0.10700
catm                0.1876     0.0893    2.10  0.03579 *
catn                0.6710     0.0963    6.96  3.3e-12 ***
catv               -0.7508     0.2175   -3.45  0.00056 ***
followsP            0.9509     0.0738   12.88  < 2e-16 ***
followsV            0.5195     0.0564    9.21  < 2e-16 ***
I(class == 1)TRUE  -1.2452     0.0972  -12.81  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 958.66  on 51  degrees of freedom
Residual deviance: 232.72  on 44  degrees of freedom
AIC: 476.2

Number of Fisher Scoring iterations: 4

> anova(m2.1, m1, test = "Chi")

Analysis of Deviance Table

Model 1: ced.del ~ cat + follows + I(class == 1)
Model 2: ced.del ~ cat + follows + class
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        44        233
2        42        199  2     34.1    4e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We could also collapse together some of the word classes:

- A and D are a natural class of noun modifiers
- we could also group them with M

So we group A, D and M together:

```
> m3 <- glm(ced.del ~ I(cat == "n") + I(cat == "v") + follows + class,
+     binomial)
> summary(m3)


Call:
glm(formula = ced.del ~ I(cat == "n") + I(cat == "v") + follows +
    class, family = binomial)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-3.355  -1.399  -0.009   1.020   6.331

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1.2699     0.0965  -13.17  < 2e-16 ***
I(cat == "n")TRUE   0.5750     0.0545   10.55  < 2e-16 ***
I(cat == "v")TRUE  -0.8559     0.2033   -4.21  2.6e-05 ***
followsP            1.0133     0.0732   13.85  < 2e-16 ***
followsV            0.5771     0.0560   10.31  < 2e-16 ***
class2              1.2865     0.1029   12.50  < 2e-16 ***
class3              1.0733     0.1032   10.40  < 2e-16 ***
class4              1.4029     0.1012   13.87  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 958.66  on 51  degrees of freedom
Residual deviance: 229.11  on 44  degrees of freedom
AIC: 472.6

Number of Fisher Scoring iterations: 4
```

The deviance compared with m1 increases significantly, so we do not want to accept m3 and abandon m1:

```
> 1 - pchisq(229.11 - 198.63, 2)

[1] 2.406e-07

> anova(m3, m1, test = "Chi")

Analysis of Deviance Table

Model 1: ced.del ~ I(cat == "n") + I(cat == "v") + follows + class
Model 2: ced.del ~ cat + follows + class
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        44        229
2        42        199  2     30.5  2.4e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We group only A and D together:

```
> m4 <- glm(ced.del ~ I(cat == "n") + I(cat == "v") + I(cat == "m") +
+     follows + class, binomial)
> summary(m4)


Call:
glm(formula = ced.del ~ I(cat == "n") + I(cat == "v") + I(cat ==
    "m") + follows + class, family = binomial)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-3.268  -1.222   0.096   1.053   6.413

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1.4331     0.1018  -14.08  < 2e-16 ***
I(cat == "n")TRUE   0.7833     0.0674   11.62  < 2e-16 ***
I(cat == "v")TRUE  -0.6488     0.2068   -3.14   0.0017 **
I(cat == "m")TRUE   0.2960     0.0563    5.25  1.5e-07 ***
followsP            0.9619     0.0738   13.03  < 2e-16 ***
followsV            0.5345     0.0566    9.45  < 2e-16 ***
class2              1.2656     0.1031   12.27  < 2e-16 ***
class3              1.0451     0.1035   10.10  < 2e-16 ***
class4              1.3701     0.1015   13.50  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 958.66  on 51  degrees of freedom
Residual deviance: 201.47  on 43  degrees of freedom
AIC: 446.9

Number of Fisher Scoring iterations: 4
```

Model m4 has fewer parameters, but does not significantly increase the deviance – so we want to keep m4 and abandon m1:

```
> anova(m4, m1, test = "Chi")

Analysis of Deviance Table

Model 1: ced.del ~ I(cat == "n") + I(cat == "v") + I(cat == "m") + follows +
    class
Model 2: ced.del ~ cat + follows + class
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        43        202
2        42        199  1     2.85    0.091 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A number of cells for class 1 are poorly predicted by both m1 and m4:

```
> subset(data.frame(cedegren, fitted.counts = round(fitted(m1) * (sDel +
+     sNoDel), 2)), class == 1 & cat == "n")[, -2]

   sDel cat follows class fitted.counts
5    34   n       C     1         25.36
10   20   n       V     1          9.88
13    4   n       P     1         10.92

> subset(data.frame(cedegren, fitted.counts = round(fitted(m4) * (sDel +
+     sNoDel), 2)), class == 1 & cat == "n")[, -2]

   sDel cat follows class fitted.counts
5    34   n       C     1         25.38
10   20   n       V     1          9.90
13    4   n       P     1         10.97
```

We might worry that the models do okay on average only because very little class 1 data was collected. We can add an interaction term between just this category and class explicitly. The resulting model is significantly better:

```
> m5 <- glm(ced.del ~ cat + follows + class + I(cat == "n" & class ==
+     1), binomial)
> summary(m5)


Call:
glm(formula = ced.del ~ cat + follows + class + I(cat == "n" &
    class == 1), family = binomial)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-4.175  -1.083   0.120   0.925   6.512

Coefficients:
                               Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -1.4573     0.1351  -10.78  < 2e-16 ***
catd                            -0.1732     0.1006   -1.72   0.0852 .
catm                             0.1715     0.0898    1.91   0.0562 .
catn                             0.6243     0.0979    6.38  1.8e-10 ***
catv                            -0.7749     0.2189   -3.54   0.0004 ***
followsP                         0.9561     0.0740   12.92  < 2e-16 ***
followsV                         0.5388     0.0567    9.50  < 2e-16 ***
class2                           1.4222     0.1201   11.84  < 2e-16 ***
class3                           1.1999     0.1204    9.96  < 2e-16 ***
class4                           1.5256     0.1186   12.86  < 2e-16 ***
I(cat == "n" & class == 1)TRUE   0.6147     0.2279    2.70   0.0070 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 958.66  on 51  degrees of freedom
Residual deviance: 191.34  on 41  degrees of freedom
AIC: 440.8
```

```
Number of Fisher Scoring iterations: 4

> anova(m1, m5, test = "Chi")

Analysis of Deviance Table

Model 1: ced.del ~ cat + follows + class
Model 2: ced.del ~ cat + follows + class + I(cat == "n" & class == 1)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        42        199
2        41        191  1     7.29   0.0069 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(m4, m5, test = "Chi")

Analysis of Deviance Table

Model 1: ced.del ~ I(cat == "n") + I(cat == "v") + I(cat == "m") + follows +
    class
Model 2: ced.del ~ cat + follows + class + I(cat == "n" & class == 1)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        43        202
2        41        191  2     10.1   0.0063 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can add interaction terms between all categories and classes:

```
> m6 <- glm(ced.del ~ cat + follows + class + cat:class, binomial)
> print(summary(m6), dig = 1)


Call:
glm(formula = ced.del ~ cat + follows + class + cat:class, family = binomial)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
  -4.2    -0.7     0.0     0.9     5.7

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2e+00      3e-01    -5.8    7e-09 ***
catd           2e-03      4e-01     0.0    0.995
catm           8e-01      3e-01     2.2    0.029 *
catn           2e+00      4e-01     4.4    1e-05 ***
catv          -2e+01      1e+03     0.0    0.989
followsP       1e+00      7e-02    12.9   <2e-16 ***
followsV       5e-01      6e-02     9.5   <2e-16 ***
class2         2e+00      4e-01     5.5    4e-08 ***
class3         1e+00      4e-01     3.9    1e-04 ***
class4         2e+00      4e-01     5.9    3e-09 ***
catd:class2   -5e-01      4e-01    -1.1    0.278
catm:class2   -8e-01      4e-01    -1.9    0.052 .
catn:class2   -9e-01      4e-01    -2.3    0.022 *
```

```
catv:class2    1e+01      1e+03     0.0    0.990
catd:class3    1e-01      4e-01     0.2    0.825
catm:class3   -2e-01      4e-01    -0.6    0.540
catn:class3   -1e+00      4e-01    -2.6    0.009 **
catv:class3    2e+01      1e+03     0.0    0.989
catd:class4   -2e-01      4e-01    -0.4    0.664
catm:class4   -9e-01      4e-01    -2.2    0.025 *
catn:class4   -1e+00      4e-01    -2.6    0.009 **
catv:class4    2e+01      1e+03     0.0    0.990
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 958.66  on 51  degrees of freedom
Residual deviance: 135.68  on 30  degrees of freedom
AIC: 407.1

Number of Fisher Scoring iterations: 15
```

We can make the saturated model (i.e., a model with as many parameters as observations and with a deviance that is practically 0) adding all three-way interactions:

```
> m7 <- glm(ced.del ~ cat * follows * class, binomial)
> print(summary(m7), dig = 1)


Call:
glm(formula = ced.del ~ cat * follows * class, family = binomial)

Deviance Residuals:
 [1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[24]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[47]  0  0  0  0  0  0

Coefficients: (8 not defined because of singularities)
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)            -2e+00      4e-01    -4.5    6e-06 ***
catd                    2e-01      5e-01     0.5    0.635
catm                    7e-01      4e-01     1.5    0.124
catn                    2e+00      5e-01     3.6    4e-04 ***
catv                   -2e+01      9e+04     0.0    1.000
followsP               -2e+01      9e+04     0.0    1.000
followsV                1e+00      7e-01     2.0    0.051 .
class2                  2e+00      4e-01     4.4    1e-05 ***
class3                  1e+00      4e-01     2.8    0.005 **
class4                  2e+00      4e-01     4.5    6e-06 ***
catd:followsP              NA         NA      NA       NA
catm:followsP           2e+01      9e+04     0.0    1.000
catn:followsP           2e+01      9e+04     0.0    1.000
catv:followsP              NA         NA      NA       NA
catd:followsV          -2e+00      1e+00    -2.1    0.036 *
catm:followsV          -6e-01      8e-01    -0.7    0.467
catn:followsV           2e+00      1e+00     1.4    0.156
```

```
catv:followsV          -1e+00      1e+05     0.0    1.000
catd:class2            -6e-01      5e-01    -1.2    0.227
catm:class2            -7e-01      5e-01    -1.4    0.147
catn:class2            -1e+00      5e-01    -2.3    0.021 *
catv:class2             2e+01      9e+04     0.0    1.000
catd:class3            -7e-03      5e-01     0.0    0.990
catm:class3            -6e-02      5e-01    -0.1    0.897
catn:class3            -8e-01      5e-01    -1.6    0.103
catv:class3             2e+01      9e+04     0.0    1.000
catd:class4            -2e-01      5e-01    -0.4    0.661
catm:class4            -8e-01      5e-01    -1.6    0.111
catn:class4            -8e-01      5e-01    -1.6    0.104
catv:class4             2e+01      9e+04     0.0    1.000
followsP:class2         3e+01      9e+04     0.0    1.000
followsV:class2        -1e+00      8e-01    -1.1    0.255
followsP:class3         2e+01      9e+04     0.0    1.000
followsV:class3         1e-01      9e-01     0.2    0.876
followsP:class4         3e+01      9e+04     0.0    1.000
followsV:class4        -8e-01      9e-01    -0.9    0.369
catd:followsP:class2       NA         NA      NA       NA
catm:followsP:class2   -2e+01      9e+04     0.0    1.000
catn:followsP:class2   -2e+01      9e+04     0.0    1.000
catv:followsP:class2       NA         NA      NA       NA
catd:followsV:class2    2e+00      1e+00     1.5    0.126
catm:followsV:class2    7e-01      9e-01     0.8    0.420
catn:followsV:class2    2e-01      1e+00     0.2    0.881
catv:followsV:class2    5e-01      1e+05     0.0    1.000
catd:followsP:class3       NA         NA      NA       NA
catm:followsP:class3   -2e+01      9e+04     0.0    1.000
catn:followsP:class3   -2e+01      9e+04     0.0    1.000
catv:followsP:class3       NA         NA      NA       NA
catd:followsV:class3    1e+00      1e+00     1.0    0.332
catm:followsV:class3   -4e-01      9e-01    -0.5    0.646
catn:followsV:class3   -3e+00      1e+00    -2.3    0.021 *
catv:followsV:class3   -3e-01      1e+05     0.0    1.000
catd:followsP:class4       NA         NA      NA       NA
catm:followsP:class4   -3e+01      9e+04     0.0    1.000
catn:followsP:class4   -2e+01      9e+04     0.0    1.000
catv:followsP:class4       NA         NA      NA       NA
catd:followsV:class4    1e+00      1e+00     1.1    0.284
catm:followsV:class4    7e-01      1e+00     0.8    0.438
catn:followsV:class4   -3e+00      1e+00    -1.8    0.071 .
catv:followsV:class4    2e+00      1e+05     0.0    1.000
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 9.5866e+02  on 51  degrees of freedom
Residual deviance: 2.5389e-10  on  0  degrees of freedom
AIC: 331.5

Number of Fisher Scoring iterations: 23
```

Models m6 and m7 are not linguistically very interesting: a lot of parameters are not significant and no clear generalizations are emerging.

Looking at the interaction terms in m6 a bit closer, it seems that CAT N and, more marginally, CAT M have a special behavior:

```
> print(summary(m6), dig = 1)


Call:
glm(formula = ced.del ~ cat + follows + class + cat:class, family = binomial)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
  -4.2    -0.7     0.0     0.9     5.7

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2e+00      3e-01    -5.8    7e-09 ***
catd           2e-03      4e-01     0.0    0.995
catm           8e-01      3e-01     2.2    0.029 *
catn           2e+00      4e-01     4.4    1e-05 ***
catv          -2e+01      1e+03     0.0    0.989
followsP       1e+00      7e-02    12.9   <2e-16 ***
followsV       5e-01      6e-02     9.5   <2e-16 ***
class2         2e+00      4e-01     5.5    4e-08 ***
class3         1e+00      4e-01     3.9    1e-04 ***
class4         2e+00      4e-01     5.9    3e-09 ***
catd:class2   -5e-01      4e-01    -1.1    0.278
catm:class2   -8e-01      4e-01    -1.9    0.052 .
catn:class2   -9e-01      4e-01    -2.3    0.022 *
catv:class2    1e+01      1e+03     0.0    0.990
catd:class3    1e-01      4e-01     0.2    0.825
catm:class3   -2e-01      4e-01    -0.6    0.540
catn:class3   -1e+00      4e-01    -2.6    0.009 **
catv:class3    2e+01      1e+03     0.0    0.989
catd:class4   -2e-01      4e-01    -0.4    0.664
catm:class4   -9e-01      4e-01    -2.2    0.025 *
catn:class4   -1e+00      4e-01    -2.6    0.009 **
catv:class4    2e+01      1e+03     0.0    0.990
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 958.66  on 51  degrees of freedom
Residual deviance: 135.68  on 30  degrees of freedom
AIC: 407.1

Number of Fisher Scoring iterations: 15
```

This is related to what we noticed in model m4 above (the best model without interaction terms, i.e., the best among m1 through m4) – namely that the CAT distinctions can be conflated by lumping `cat=="a"` and `cat=="d"` together:

```
> print(summary(m4), dig = 1)


Call:
glm(formula = ced.del ~ I(cat == "n") + I(cat == "v") + I(cat ==
    "m") + follows + class, family = binomial)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
  -3.3     -1.2      0.1      1.1      6.4

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)          -1.43       0.10     -14   <2e-16 ***
I(cat == "n")TRUE     0.78       0.07      12   <2e-16 ***
I(cat == "v")TRUE    -0.65       0.21      -3    0.002 **
I(cat == "m")TRUE     0.30       0.06       5    1e-07 ***
followsP              0.96       0.07      13   <2e-16 ***
followsV              0.53       0.06       9   <2e-16 ***
class2                1.27       0.10      12   <2e-16 ***
class3                1.05       0.10      10   <2e-16 ***
class4                1.37       0.10      14   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 958.66  on 51  degrees of freedom
Residual deviance: 201.47  on 43  degrees of freedom
AIC: 446.9

Number of Fisher Scoring iterations: 4
```

We conflate the CAT distinctions along the same lines and add interaction terms between `cat=="n"` and classes 1 and 3 (given that classes 2 and 4 behave in similar ways):

```
> m8 <- glm(ced.del ~ I(cat == "n") + I(cat == "m") + follows + class +
+     I(cat == "n" & class == 3) + I(cat == "n" & class == 1), binomial)
> print(summary(m8), dig = 1)


Call:
glm(formula = ced.del ~ I(cat == "n") + I(cat == "m") + follows +
    class + I(cat == "n" & class == 3) + I(cat == "n" & class ==
    1), family = binomial)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
  -4.2     -1.3      0.1      1.1      6.0

Coefficients:
                               Estimate Std. Error z value Pr(>|z|)
(Intercept)                       -1.60       0.12     -14   <2e-16 ***
I(cat == "n")TRUE                  0.97       0.08      12   <2e-16 ***
```

```
I(cat == "m")TRUE                    0.33       0.06       6    4e-09 ***
followsP                             0.96       0.07      13   <2e-16 ***
followsV                             0.52       0.06       9   <2e-16 ***
class2                               1.36       0.12      11   <2e-16 ***
class3                               1.29       0.12      11   <2e-16 ***
class4                               1.48       0.12      12   <2e-16 ***
I(cat == "n" & class == 3)TRUE      -0.58       0.12      -5    8e-07 ***
I(cat == "n" & class == 1)TRUE       0.42       0.23       2     0.07 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 958.66  on 51  degrees of freedom
Residual deviance: 180.56  on 42  degrees of freedom
AIC: 428

Number of Fisher Scoring iterations: 4
```

This model has the same number of parameters as the initial model we built, but improves the loglikelihood by deviance/2, i.e., it makes the observed data more than 8000 times more likely.

```
> anova(m1, m8)

Analysis of Deviance Table

Model 1: ced.del ~ cat + follows + class
Model 2: ced.del ~ I(cat == "n") + I(cat == "m") + follows + class + I(cat ==
    "n" & class == 3) + I(cat == "n" & class == 1)
  Resid. Df Resid. Dev Df Deviance
1        42        199
2        42        181  0     18.1

> exp(anova(m1, m8)$Deviance[2]/2)

[1] 8393

> logLik(m1)

'log Lik.' -213 (df=10)

> logLik(m8)

'log Lik.' -204 (df=10)

> exp(logLik(m8) - logLik(m1))

'log Lik.' 8393 (df=10)

> exp(logLik(m8))/exp(logLik(m1))

'log Lik.' 8393 (df=10)
```

Quick summary:

```
> cedegren <- read.table("http://www-nlp.stanford.edu/manning/courses/ling289/cedegren.txt",
+     header = T)
> cedegren$class <- as.factor(cedegren$class)
> attach(cedegren)

The following objects are masked from cedegren (position 3):

    cat, class, follows, sDel, sNoDel
The following objects are masked from ced.long (position 4):

    cat, class, follows, sDel
The following objects are masked from cedegren (position 5):

    cat, class, follows, sDel, sNoDel
The following objects are masked from ced.long (position 7):

    cat, class, follows, sDel
The following objects are masked from cedegren (position 8):

    cat, class, follows, sDel, sNoDel
The following objects are masked from ced.long (position 9):

    cat, class, follows, sDel
The following objects are masked from cedegren (position 10):

    cat, class, follows, sDel, sNoDel

> ced.del <- cbind(sDel, sNoDel)
> m1 <- glm(ced.del ~ cat + follows + class, binomial)
> m2 <- glm(ced.del ~ cat + follows + I(class == 1) + I(class == 3),
+     binomial)
> m3 <- glm(ced.del ~ I(cat == "n") + I(cat == "v") + follows + class,
+     binomial)
> m4 <- glm(ced.del ~ I(cat == "n") + I(cat == "v") + I(cat == "m") +
+     follows + class, binomial)
> m5 <- glm(ced.del ~ cat + follows + class + I(cat == "n" & class ==
+     1), binomial)
> m6 <- glm(ced.del ~ cat + follows + class + cat:class, binomial)
> m7 <- glm(ced.del ~ cat * follows * class, binomial)
> m8 <- glm(ced.del ~ I(cat == "n") + I(cat == "m") + follows + class +
+     I(cat == "n" & class == 3) + I(cat == "n" & class == 1), binomial)
```

If we look back at the raw stats for nouns of class 1, we notice that we almost always have /*s*/-deletion before a vowel and almost never before a pause:

```
> subset(cedegren, class == 1 & cat == "n")

   sDel sNoDel cat follows class
5    34     40   n       C     1
10   20      1   n       V     1
13    4     15   n       P     1
```

To capture this, we add an interaction between CLASS 1, CAT N and the levels of FOLLOWS:

- we add two more parameters, but the residual deviance goes down a lot

- the deviance is close to the m6 interaction model (full CAT×CLASS interaction) – but in a more interesting way; and pretty much all parameters are significant

```
> m9 <- glm(ced.del ~ I(cat == "n") + I(cat == "m") + follows + class +
+     I(cat == "n" & class == 3) + I(cat == "n" & class == 1) + I(cat ==
+     "n" & class == 1):follows, binomial)
> print(summary(m9), dig = 1)


Call:
glm(formula = ced.del ~ I(cat == "n") + I(cat == "m") + follows +
    class + I(cat == "n" & class == 3) + I(cat == "n" & class ==
    1) + I(cat == "n" & class == 1):follows, family = binomial)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
  -3.6    -1.3     0.0     0.9     6.1

Coefficients:
                                        Estimate Std. Error z value
(Intercept)                                -1.60       0.12     -14
I(cat == "n")TRUE                           0.96       0.08      12
I(cat == "m")TRUE                           0.32       0.06       6
followsP                                    1.00       0.08      13
followsV                                    0.50       0.06       9
class2                                      1.36       0.12      11
class3                                      1.29       0.12      11
class4                                      1.48       0.12      12
I(cat == "n" & class == 3)TRUE             -0.58       0.12      -5
I(cat == "n" & class == 1)TRUE              0.47       0.27       2
followsP:I(cat == "n" & class == 1)TRUE    -2.16       0.61      -4
followsV:I(cat == "n" & class == 1)TRUE     2.66       1.05       2
                                        Pr(>|z|)
(Intercept)                               <2e-16 ***
I(cat == "n")TRUE                         <2e-16 ***
I(cat == "m")TRUE                          4e-09 ***
followsP                                  <2e-16 ***
followsV                                  <2e-16 ***
class2                                    <2e-16 ***
class3                                    <2e-16 ***
class4                                    <2e-16 ***
I(cat == "n" & class == 3)TRUE             9e-07 ***
I(cat == "n" & class == 1)TRUE              0.08 .
followsP:I(cat == "n" & class == 1)TRUE    4e-04 ***
followsV:I(cat == "n" & class == 1)TRUE     0.01 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 958.66  on 51  degrees of freedom
Residual deviance: 146.72  on 40  degrees of freedom
AIC: 398.2
```

```
Number of Fisher Scoring iterations: 4
```

Quick comparison:

```
> anova(m9, m6, test = "Chi")

Analysis of Deviance Table

Model 1: ced.del ~ I(cat == "n") + I(cat == "m") + follows + class + I(cat ==
    "n" & class == 3) + I(cat == "n" & class == 1) + I(cat ==
    "n" & class == 1):follows
Model 2: ced.del ~ cat + follows + class + cat:class
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        40        147
2        30        136 10       11     0.35

> anova(m8, m9, test = "Chi")

Analysis of Deviance Table

Model 1: ced.del ~ I(cat == "n") + I(cat == "m") + follows + class + I(cat ==
    "n" & class == 3) + I(cat == "n" & class == 1)
Model 2: ced.del ~ I(cat == "n") + I(cat == "m") + follows + class + I(cat ==
    "n" & class == 3) + I(cat == "n" & class == 1) + I(cat ==
    "n" & class == 1):follows
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        42        181
2        40        147  2     33.8  4.5e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We make sure that we need all the 3-way interaction terms by comparing m9 with a model that just has an interaction term for `cat=="n"`, `class==1` and `follows=="V"`.

```
> m10 <- glm(ced.del ~ I(cat == "n") + I(cat == "m") + follows + class +
+     I(cat == "n" & class == 3) + I(cat == "n" & class == 1) + I(cat ==
+     "n" & class == 1 & follows == "V"), binomial)
> print(summary(m10), dig = 1)


Call:
glm(formula = ced.del ~ I(cat == "n") + I(cat == "m") + follows +
    class + I(cat == "n" & class == 3) + I(cat == "n" & class ==
    1) + I(cat == "n" & class == 1 & follows == "V"), family = binomial)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
    -4      -1       0       1       6

Coefficients:
                                              Estimate Std. Error
(Intercept)                                      -1.59       0.12
I(cat == "n")TRUE                                 0.97       0.08
I(cat == "m")TRUE                                 0.33       0.06
```

```
followsP                                            0.96       0.07
followsV                                            0.49       0.06
class2                                              1.36       0.12
class3                                              1.29       0.12
class4                                              1.48       0.12
I(cat == "n" & class == 3)TRUE                     -0.58       0.12
I(cat == "n" & class == 1)TRUE                      0.05       0.25
I(cat == "n" & class == 1 & follows == "V")TRUE     3.07       1.05
                                                z value Pr(>|z|)
(Intercept)                                       -13.6   <2e-16 ***
I(cat == "n")TRUE                                  12.3   <2e-16 ***
I(cat == "m")TRUE                                   5.9    3e-09 ***
followsP                                           12.9   <2e-16 ***
followsV                                            8.7   <2e-16 ***
class2                                             11.3   <2e-16 ***
class3                                             10.6   <2e-16 ***
class4                                             12.4   <2e-16 ***
I(cat == "n" & class == 3)TRUE                     -4.9    8e-07 ***
I(cat == "n" & class == 1)TRUE                      0.2    0.830
I(cat == "n" & class == 1 & follows == "V")TRUE     2.9    0.003 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 958.66  on 51  degrees of freedom
Residual deviance: 161.77  on 41  degrees of freedom
AIC: 411.2

Number of Fisher Scoring iterations: 4

> anova(m10, m9, test = "Chi")

Analysis of Deviance Table

Model 1: ced.del ~ I(cat == "n") + I(cat == "m") + follows + class + I(cat ==
    "n" & class == 3) + I(cat == "n" & class == 1) + I(cat ==
    "n" & class == 1 & follows == "V")
Model 2: ced.del ~ I(cat == "n") + I(cat == "m") + follows + class + I(cat ==
    "n" & class == 3) + I(cat == "n" & class == 1) + I(cat ==
    "n" & class == 1):follows
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        41        162
2        40        147  1     15.1    1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

m9 is significantly better, as we might expect given that the three level of FOLLOWS interact very differently with `class==1 & cat=="n"` :

```
> subset(cedegren, class == 1 & cat == "n")

   sDel sNoDel cat follows class
5    34     40    n       C     1
```

```
10   20      1   n       V      1
13    4     15   n       P      1

> detach(cedegren)
```

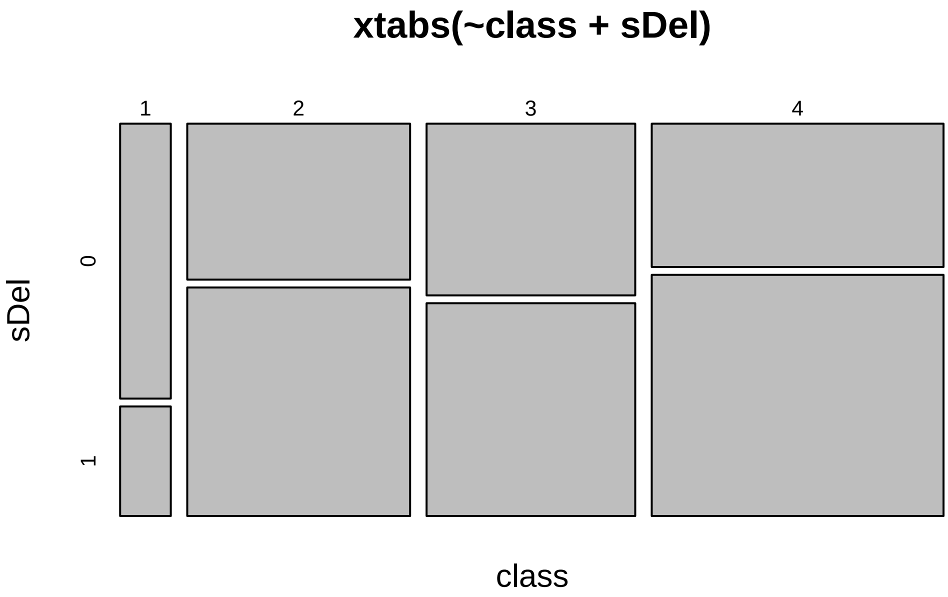
4 Long format data & logistic regression

We used binary outcome data in a summary format (counts of sDel and sNoDel for each combination of levels of factors). An alternative is long format, where each observation is a line; collected data often starts out in this form.

```
> ced.long <- read.table("http://www-nlp.stanford.edu/manning/courses/ling289/cedegren-long.txt",
+     header = T)
> str(ced.long)

'data.frame': 8846 obs. of  4 variables:
 $ sDel   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ cat    : Factor w/ 5 levels "a","d","m","n",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ follows: Factor w/ 3 levels "C","P","V": 1 1 1 1 1 1 1 1 1 1 ...
 $ class  : int  1 1 1 1 1 1 1 1 1 1 ...

> summary(ced.long)

      sDel         cat      follows      class     
 Min.   :0.000   a: 609   C:5600   Min.   :1.00  
 1st Qu.:0.000   d:1393   P:1304   1st Qu.:2.00  
 Median :1.000   m:4460   V:1942   Median :3.00  
 Mean   :0.576   n:2268            Mean   :2.96  
 3rd Qu.:1.000   v: 116            3rd Qu.:4.00  
 Max.   :1.000                     Max.   :4.00  

> ced.long[sample(nrow(ced.long), 15), ]

     sDel cat follows class
5144    0   n       V     3
3337    1   m       C     3
1997    1   n       C     2
4737    1   m       V     3
4066    0   d       C     3
5051    1   n       V     3
7508    1   n       C     4
3805    0   m       C     3
3100    0   n       P     2
7862    1   m       V     4
7156    1   a       C     4
7572    0   n       C     4
6086    1   m       C     4
2006    1   n       C     2
7961    0   m       V     4
```

We can then examine the data by constructing cross-tabulations. This can be done with any number of variables, but they get harder to read with more than two.

```
> attach(ced.long)

The following objects are masked from cedegren (position 3):

    cat, class, follows, sDel
The following objects are masked from ced.long (position 4):

    cat, class, follows, sDel
The following objects are masked from cedegren (position 5):

    cat, class, follows, sDel
The following objects are masked from ced.long (position 7):

    cat, class, follows, sDel
The following objects are masked from cedegren (position 8):

    cat, class, follows, sDel
The following objects are masked from ced.long (position 9):

    cat, class, follows, sDel
The following objects are masked from cedegren (position 10):

    cat, class, follows, sDel

> xtabs(~sDel + class)

    class
sDel    1    2    3    4
   0  414 1033 1065 1243
   1  165 1514 1320 2092

> mosaicplot(xtabs(~class + sDel))
```



Do we need to build logistic regression models, or could we get everything we need from just looking at crosstabs?

For example, we can already see here that /*s*/-deletion is strongly disfavored by class 1 speakers, but moderately preferred by other classes – as the class-wise proportions below show:

```
> round(prop.table(xtabs(~sDel + class), 2), 2)

    class
sDel    1    2    3    4
   0 0.72 0.41 0.45 0.37
   1 0.28 0.59 0.55 0.63
```

Sometimes looking at crosstabs works, but the fundamental observation is this:

- the predictive effect of variables can be spurious, hidden or reversed when we look at the marginal totals in crosstabs (recall Simpson's paradox)

Reason: they do not take into account correlations between predictors.

Logistic regression works out the best way to attribute "causal" effect to predictors (as long as the predictors are actually coded in the data).

We can load the long-form data into the basic glm function:

```
> summary(glm(sDel ~ cat + follows + class, binomial, ced.long))


Call:
glm(formula = sDel ~ cat + follows + class, family = binomial,
    data = ced.long)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
```

```
-1.929  -1.193   0.727   1.071   1.631

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.8106     0.1064   -7.62  2.6e-14 ***
catd         -0.1347     0.0992   -1.36   0.1746
catm          0.2172     0.0885    2.46   0.0141 *
catn          0.7127     0.0954    7.47  8.0e-14 ***
catv         -0.6368     0.2161   -2.95   0.0032 **
followsP      0.9416     0.0735   12.81  < 2e-16 ***
followsV      0.5265     0.0560    9.40  < 2e-16 ***
class         0.2119     0.0233    9.10  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 12061  on 8845  degrees of freedom
Residual deviance: 11434  on 8838  degrees of freedom
AIC: 11450

Number of Fisher Scoring iterations: 4
```

We get the same estimates and SEs as before – and the same difference between residual deviance and null deviance:

```
> summary(glm(ced.del ~ cat + follows + class, binomial, cedegren))


Call:
glm(formula = ced.del ~ cat + follows + class, family = binomial,
    data = cedegren)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
 -3.24   -1.34    0.05    1.01    6.40

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.3183     0.1222  -10.79  < 2e-16 ***
catd         -0.1693     0.1003   -1.69  0.09146 .
catm          0.1786     0.0895    1.99  0.04605 *
catn          0.6667     0.0965    6.91  4.9e-12 ***
catv         -0.7675     0.2184   -3.51  0.00044 ***
followsP      0.9525     0.0740   12.87  < 2e-16 ***
followsV      0.5341     0.0566    9.44  < 2e-16 ***
class2        1.2704     0.1032   12.31  < 2e-16 ***
class3        1.0480     0.1035   10.12  < 2e-16 ***
class4        1.3742     0.1016   13.53  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 958.66  on 51  degrees of freedom
Residual deviance: 198.63  on 42  degrees of freedom
AIC: 446.1

Number of Fisher Scoring iterations: 4
```

But the deviances (and the corresponding dof.s) are much larger.

References

Abelson, R.P. (1995). *Statistics as Principled Argument*. L. Erlbaum Associates.

Baayen, R. Harald (2008). *Analyzing Linguistic Data: A Practical Introduction to Statistics Using R*. Cambridge University Press.

Braun, J. and D.J. Murdoch (2007). *A First Course in Statistical Programming with R*. Cambridge University Press.

De Veaux, R.D. et al. (2005). *Stats: Data and Models*. Pearson Education, Limited.

Diez, D. et al. (2013). *OpenIntro Statistics: Second Edition*. CreateSpace Independent Publishing Platform. URL: `http://www.openintro.org/stat/textbook.php`.

Faraway, J.J. (2004). *Linear Models With R*. Chapman & Hall Texts in Statistical Science Series. Chapman & Hall/CRC.

Gelman, A. and J. Hill (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Analytical Methods for Social Research. Cambridge University Press.

Gries, S.T. (2009). *Quantitative Corpus Linguistics with R: A Practical Introduction*. Taylor & Francis.

— (2013). *Statistics for Linguistics with R: A Practical Introduction, 2nd Edition*. Mouton De Gruyter.

Johnson, K. (2008). *Quantitative methods in linguistics*. Blackwell Pub.

Kruschke, John K. (2011). *Doing Bayesian Data Analysis: A Tutorial with R and BUGS*. Academic Press/Elsevier.

Miles, J. and M. Shevlin (2001). *Applying Regression and Correlation: A Guide for Students and Researchers*. SAGE Publications.

Wright, D.B. and K. London (2009). *Modern regression techniques using R: A practical guide for students and researchers*. SAGE.

Xie, Yihui (2013). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC.