GLM Examples

Nancy

01/12/2021

I'm sure there are nicer ways to set things up, but I'm using a default version, except I added `scipen`, to force the output not to use exponential notation, which I hate.

Example 1: binomial data (Exercise 2.1 in ELM)

Faraway asks first to fit a model with all interactions, so here it is:

```
data(esoph)
head(esoph)
```

```
##   agegp     alcgp    tobgp ncases ncontrols
## 1 25-34 0-39g/day 0-9g/day      0        40
## 2 25-34 0-39g/day    10-19      0        10
## 3 25-34 0-39g/day    20-29      0         6
## 4 25-34 0-39g/day      30+      0         5
## 5 25-34     40-79 0-9g/day      0        27
## 6 25-34     40-79    10-19      0         7
```

```
help(esoph)
xtabs(ncases ~ agegp+tobgp, data = esoph)
```

```
##        tobgp
## agegp   0-9g/day 10-19 20-29 30+
##   25-34        0     1     0   0
##   35-44        2     4     3   0
##   45-54       14    13     8  11
##   55-64       25    23    12  16
##   65-74       31    12    10   2
##   75+          6     5     0   2
```

There could be more elegant ways to tabulate the data, but this is at least feasible to look at. In the help file for `xtabs` this dataset is used as an example. And I learned about "flat tables".

```
ftable(xtabs(cbind(ncases,ncontrols) ~ agegp, data = esoph))
```

```
##        ncases ncontrols
## agegp
## 25-34       1       115
## 35-44       9       190
## 45-54      46       167
## 55-64      76       166
## 65-74      55       106
## 75+        13        31
```

```
ftable(xtabs(cbind(ncases,ncontrols) ~ agegp + tobgp, data = esoph))
```

```
##                  ncases ncontrols
## agegp tobgp
## 25-34 0-9g/day       0        70
##       10-19          1        18
##       20-29          0        11
##       30+            0        16
## 35-44 0-9g/day       2       107
##       10-19          4        42
##       20-29          3        24
##       30+            0        17
## 45-54 0-9g/day      14        90
##       10-19         13        44
##       20-29          8        25
##       30+           11         8
## 55-64 0-9g/day      25        92
##       10-19         23        42
##       20-29         12        26
##       30+           16         6
## 65-74 0-9g/day      31        68
##       10-19         12        26
##       20-29         10        10
##       30+            2         2
## 75+   0-9g/day       6        20
##       10-19          5         6
##       20-29          0         3
##       30+            2         2
```

And now I'll try the full model as requested in (a):

```
farmod1 <- glm(cbind(ncases, ncontrols) ~ agegp * tobgp * alcgp, data = esoph, family = binomial)

summary(farmod1)
```

```
##
## Call:
## glm(formula = cbind(ncases, ncontrols) ~ agegp * tobgp * alcgp,
##     family = binomial, data = esoph)
##
## 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  0  0
## [26]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [51]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [76]  0  0  0  0  0  0  0  0  0  0  0  0  0
##
## Coefficients: (8 not defined because of singularities)
##                             Estimate  Std. Error z value Pr(>|z|)
## (Intercept)                   -1.519  485996.041       0        1
## agegp.L                       25.884 1745715.047       0        1
## agegp.Q                       -3.797 1606898.316       0        1
## agegp.C                       11.522 1107949.215       0        1
## agegp^4                       -1.572  599507.714       0        1
## agegp^5                        3.534  219676.773       0        1
## tobgp.L                        4.047 1091161.295       0        1
## tobgp.Q                       -5.561  582191.995       0        1
## tobgp.C                       -6.498  298391.996       0        1
```

```
## alcgp.L                     13.814 1029893.558        0         1
## alcgp.Q                      9.645  592894.707        0         1
## alcgp.C                      1.618  477407.582        0         1
## agegp.L:tobgp.L            -24.385 4041769.557        0         1
## agegp.Q:tobgp.L            -18.917 3671901.132        0         1
## agegp.C:tobgp.L             10.641 2487996.170        0         1
## agegp^4:tobgp.L            -29.392 1459466.051        0         1
## agegp^5:tobgp.L             16.769  475978.274        0         1
## agegp.L:tobgp.Q            -42.610 2393370.339        0         1
## agegp.Q:tobgp.Q            -46.011 2101171.639        0         1
## agegp.C:tobgp.Q             -4.343 1291147.760        0         1
## agegp^4:tobgp.Q            -20.588  978492.867        0         1
## agegp^5:tobgp.Q              0.827  227116.570        0         1
## agegp.L:tobgp.C            -45.628 1277441.809        0         1
## agegp.Q:tobgp.C            -28.380 1114751.643        0         1
## agegp.C:tobgp.C            -16.125  590115.895        0         1
## agegp^4:tobgp.C            -14.288  551276.141        0         1
## agegp^5:tobgp.C              3.473   66882.268        0         1
## agegp.L:alcgp.L            -32.923 3887459.739        0         1
## agegp.Q:alcgp.L            -25.028 3477018.544        0         1
## agegp.C:alcgp.L             11.417 2266520.509        0         1
## agegp^4:alcgp.L            -29.027 1417029.399        0         1
## agegp^5:alcgp.L             -4.936  383723.512        0         1
## agegp.L:alcgp.Q             -5.833 2512995.062        0         1
## agegp.Q:alcgp.Q             11.650 2193169.175        0         1
## agegp.C:alcgp.Q             23.249 1254385.934        0         1
## agegp^4:alcgp.Q            -20.477 1023107.256        0         1
## agegp^5:alcgp.Q             12.100  157752.433        0         1
## agegp.L:alcgp.C            -23.608 1891714.189        0         1
## agegp.Q:alcgp.C            -12.838 1677219.507        0         1
## agegp.C:alcgp.C             10.969  970914.062        0         1
## agegp^4:alcgp.C            -14.506  691933.133        0         1
## agegp^5:alcgp.C             -0.454  114515.907        0         1
## tobgp.L:alcgp.L              6.261 2372772.376        0         1
## tobgp.Q:alcgp.L              6.464 1319917.800        0         1
## tobgp.C:alcgp.L              1.870  500008.795        0         1
## tobgp.L:alcgp.Q              2.951 1388011.629        0         1
## tobgp.Q:alcgp.Q            -17.100  856540.432        0         1
## tobgp.C:alcgp.Q            -22.635  448738.830        0         1
## tobgp.L:alcgp.C             -0.586 1059741.693        0         1
## tobgp.Q:alcgp.C             -4.238  446937.960        0         1
## tobgp.C:alcgp.C             -1.521   34427.071        0         1
## agegp.L:tobgp.L:alcgp.L    -96.328 9327513.619        0         1
## agegp.Q:tobgp.L:alcgp.L    -91.349 8302966.145        0         1
## agegp.C:tobgp.L:alcgp.L      6.067 5178793.060        0         1
## agegp^4:tobgp.L:alcgp.L   -105.060 3493738.395        0         1
## agegp^5:tobgp.L:alcgp.L     15.149  935007.159        0         1
## agegp.L:tobgp.Q:alcgp.L    -69.427 5603666.977        0         1
## agegp.Q:tobgp.Q:alcgp.L    -69.948 4878040.820        0         1
## agegp.C:tobgp.Q:alcgp.L     28.512 2784010.158        0         1
## agegp^4:tobgp.Q:alcgp.L    -82.615 2290266.609        0         1
## agegp^5:tobgp.Q:alcgp.L     21.473  485431.933        0         1
## agegp.L:tobgp.C:alcgp.L    -36.864 2300238.915        0         1
## agegp.Q:tobgp.C:alcgp.L     -6.768 1981041.808        0         1
```

```
## agegp.C:tobgp.C:alcgp.L     -10.759  959111.154       0        1
## agegp^4:tobgp.C:alcgp.L      -4.878 1020835.008       0        1
## agegp^5:tobgp.C:alcgp.L          NA          NA      NA       NA
## agegp.L:tobgp.L:alcgp.Q     -50.753 6165106.567       0        1
## agegp.Q:tobgp.L:alcgp.Q     -27.816 5339092.459       0        1
## agegp.C:tobgp.L:alcgp.Q      29.406 2884824.359       0        1
## agegp^4:tobgp.L:alcgp.Q     -62.372 2601860.374       0        1
## agegp^5:tobgp.L:alcgp.Q      24.700  327614.973       0        1
## agegp.L:tobgp.Q:alcgp.Q    -126.927 4062058.998       0        1
## agegp.Q:tobgp.Q:alcgp.Q    -111.404 3423516.250       0        1
## agegp.C:tobgp.Q:alcgp.Q      -6.680 1713401.926       0        1
## agegp^4:tobgp.Q:alcgp.Q     -67.367 1813421.259       0        1
## agegp^5:tobgp.Q:alcgp.Q          NA          NA      NA       NA
## agegp.L:tobgp.C:alcgp.Q    -123.787 1936368.121       0        1
## agegp.Q:tobgp.C:alcgp.Q     -80.072 1597153.868       0        1
## agegp.C:tobgp.C:alcgp.Q     -50.706  931915.984       0        1
## agegp^4:tobgp.C:alcgp.Q     -40.309  808568.033       0        1
## agegp^5:tobgp.C:alcgp.Q          NA          NA      NA       NA
## agegp.L:tobgp.L:alcgp.C     -30.516 4286442.833       0        1
## agegp.Q:tobgp.L:alcgp.C     -12.267 3746888.257       0        1
## agegp.C:tobgp.L:alcgp.C      24.658 2148864.032       0        1
## agegp^4:tobgp.L:alcgp.C     -23.112 1566605.642       0        1
## agegp^5:tobgp.L:alcgp.C       4.383  248195.100       0        1
## agegp.L:tobgp.Q:alcgp.C     -20.031 2093997.219       0        1
## agegp.Q:tobgp.Q:alcgp.C     -11.500 1706481.573       0        1
## agegp.C:tobgp.Q:alcgp.C      12.846  844262.716       0        1
## agegp^4:tobgp.Q:alcgp.C      -9.404  834733.653       0        1
## agegp^5:tobgp.Q:alcgp.C          NA          NA      NA       NA
## agegp.L:tobgp.C:alcgp.C     -18.199  288037.539       0        1
## agegp.Q:tobgp.C:alcgp.C          NA          NA      NA       NA
## agegp.C:tobgp.C:alcgp.C          NA          NA      NA       NA
## agegp^4:tobgp.C:alcgp.C          NA          NA      NA       NA
## agegp^5:tobgp.C:alcgp.C          NA          NA      NA       NA
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 367.95345785593372  on 87  degrees of freedom
## Residual deviance:   0.00000000038717  on  0  degrees of freedom
## AIC: 291.1
##
## Number of Fisher Scoring iterations: 25
```

So we probably didn't really want this. We've fit a parameter to every observation, not helpful. What is with all the .L, .C etc?

```
is.ordered(esoph$agegp)
```

```
## [1] TRUE
```

Ordered factors use polynomial contrasts to fit the various levels. This is a kind of compromise between treating the values as continuous and treating them as completely unordered. In the `help` file for this dataset you'll see models with covariates `unclass(tobgp)` and `unclass(alcgp)`.

Now Faraway suggests using backward elimination to simplify the model "as far as is reasonable".

```r
farmod2 <- step(farmod1, direction = "backward")
```

```
## Start:  AIC=291.1
## cbind(ncases, ncontrols) ~ agegp * tobgp * alcgp

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                      Df Deviance AIC
## - agegp:tobgp:alcgp 37     30.8 248
## <none>                      0.0 291

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=247.9
## cbind(ncases, ncontrols) ~ agegp + tobgp + alcgp + agegp:tobgp +
##     agegp:alcgp + tobgp:alcgp

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                Df Deviance AIC
## - tobgp:alcgp  9     37.5 237
## - agegp:tobgp 15     50.3 237
## - agegp:alcgp 15     56.8 244
## <none>               30.8 248

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=236.6
## cbind(ncases, ncontrols) ~ agegp + tobgp + alcgp + agegp:tobgp +
##     agegp:alcgp
##
##                Df Deviance AIC
## - agegp:tobgp 15     56.3 225
## - agegp:alcgp 15     62.8 232
## <none>               37.5 237
##
## Step:  AIC=225.3
## cbind(ncases, ncontrols) ~ agegp + tobgp + alcgp + agegp:alcgp
##
##                Df Deviance AIC
## - agegp:alcgp 15     82.3 221
## <none>               56.3 225
## - tobgp        3     80.3 243
##
## Step:  AIC=221.4
## cbind(ncases, ncontrols) ~ agegp + tobgp + alcgp
##
##         Df Deviance AIC
## <none>         82.3 221
## - tobgp  3    105.9 239
## - agegp  5    208.8 338
## - alcgp  3    210.3 343
```

```r
summary(farmod2)
```

```
##
```

```
## Call:
## glm(formula = cbind(ncases, ncontrols) ~ agegp + tobgp + alcgp,
##     family = binomial, data = esoph)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.951  -0.738  -0.244   0.613   2.413
##
## Coefficients:
##             Estimate Std. Error z value     Pr(>|z|)
## (Intercept)  -1.1904     0.2074   -5.74 0.0000000094 ***
## agegp.L       3.9966     0.6939    5.76 0.0000000084 ***
## agegp.Q      -1.6574     0.6212   -2.67       0.0076 **
## agegp.C       0.1109     0.4681    0.24       0.8127
## agegp^4       0.0789     0.3246    0.24       0.8079
## agegp^5      -0.2622     0.2134   -1.23       0.2192
## tobgp.L       1.1175     0.2401    4.65 0.0000032639 ***
## tobgp.Q       0.3452     0.2241    1.54       0.1236
## tobgp.C       0.3169     0.2109    1.50       0.1329
## alcgp.L       2.5390     0.2638    9.62      < 2e-16 ***
## alcgp.Q       0.0938     0.2242    0.42       0.6758
## alcgp.C       0.4393     0.1835    2.39       0.0166 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 367.953  on 87  degrees of freedom
## Residual deviance:  82.337  on 76  degrees of freedom
## AIC: 221.4
##
## Number of Fisher Scoring iterations: 6
```

This eliminated all the interactions, but still used orthogonal polynomials for the variables. And up to 5th degree for age! Doesn't quite make sense. We could either treat the variables as continuous, or as unordered factors.

```
farmod3 <- glm(cbind(ncases, ncontrols) ~ unclass(agegp) + unclass(alcgp) + unclass(tobgp), family = bi
summary(farmod3)
```

```
##
## Call:
## glm(formula = cbind(ncases, ncontrols) ~ unclass(agegp) + unclass(alcgp) +
##     unclass(tobgp), family = binomial, data = esoph)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.648  -0.925  -0.434   0.674   2.457
##
## Coefficients:
##                Estimate Std. Error z value  Pr(>|z|)
## (Intercept)     -7.1640     0.5093  -14.07   < 2e-16 ***
## unclass(agegp)   0.7438     0.0818    9.09   < 2e-16 ***
## unclass(alcgp)   1.1026     0.1032   10.69   < 2e-16 ***
## unclass(tobgp)   0.4309     0.0939    4.59 0.0000045 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 367.95  on 87  degrees of freedom
## Residual deviance: 108.78  on 84  degrees of freedom
## AIC: 231.8
##
## Number of Fisher Scoring iterations: 4
```

This seems a bit too simple, treating each covariate as continuous and just fitting a linear model. Myabe we could try linear in age, but with tobacco and alcohol as factors. Below are several versions, they can be compared using `anova`, or their residual deviances can be compared.

```
esoph2 <- mutate(esoph,age = factor(agegp,ordered=FALSE), tob = factor(tobgp, ordered = F), alc = facto
# new dataset has unordered factors
farmod4 <- glm(cbind(ncases, ncontrols) ~ unclass(agegp) + tob*alc, data = esoph2, family = binomial)
farmod5 <- glm(cbind(ncases, ncontrols) ~ unclass(agegp) + tob + alc, data = esoph2, family = binomial)
farmod6 <- glm(cbind(ncases, ncontrols) ~ agegp + tob + alc, data = esoph2, family = binomial)
```

Example 2 Poisson

This example in Chapter 3 of ELM is used to introduce Poisson regression. Faraway first considers ordinary linear regression, and notes that the residuals don't seem to have constant variance. Back in the day we'd take logs or square root of the response and then fit linear regression again, but a more modern approach is to fit a Poisson glm.

The residual deviance in a Poisson model is a test of fit of the model, as it is with a Binomial. The second fit below allows an inflation factor on the variance, while still using the simplicity of the Poisson for fitting.

```
data(gala)
head(gala)
```

```
##              Species Endemics  Area Elevation Nearest Scruz Adjacent
## Baltra            58       23 25.09       346     0.6   0.6     1.84
## Bartolome         31       21  1.24       109     0.6  26.3   572.33
## Caldwell           3        3  0.21       114     2.8  58.7     0.78
## Champion          25        9  0.10        46     1.9  47.4     0.18
## Coamano            2        1  0.05        77     1.9   1.9   903.82
## Daphne.Major      18       11  0.34       119     8.0   8.0     1.84
```

```
help(gala)

gal.glm <- glm(Species ~ . - Endemics,
             family = poisson,
             data = gala)

summary(gal.glm)
```

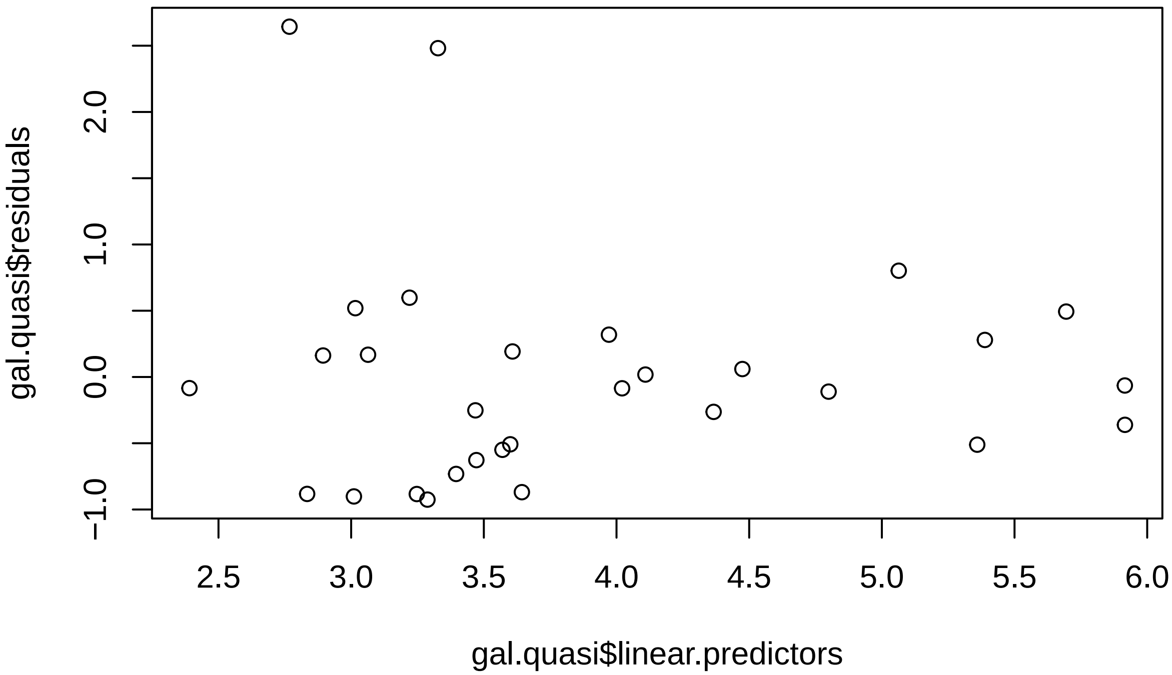
```
##
## Call:
## glm(formula = Species ~ . - Endemics, family = poisson, data = gala)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -8.275  -4.497  -0.944   1.917  10.185
```

```
##
## Coefficients:
##               Estimate Std. Error z value  Pr(>|z|)
## (Intercept)  3.1548079  0.0517495   60.96   < 2e-16 ***
## Area        -0.0005799  0.0000263  -22.07   < 2e-16 ***
## Elevation    0.0035406  0.0000874   40.51   < 2e-16 ***
## Nearest      0.0088256  0.0018213    4.85 0.0000013 ***
## Scruz       -0.0057094  0.0006256   -9.13   < 2e-16 ***
## Adjacent    -0.0006630  0.0000293  -22.61   < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 3510.73  on 29  degrees of freedom
## Residual deviance:  716.85  on 24  degrees of freedom
## AIC: 889.7
##
## Number of Fisher Scoring iterations: 5
```

```r
gal.quasi <- glm(Species ~ . - Endemics,
                 family = quasipoisson,
                 data = gala)
summary(gal.quasi)
```

```
##
## Call:
## glm(formula = Species ~ . - Endemics, family = quasipoisson,
##     data = gala)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -8.275  -4.497  -0.944   1.917  10.185
##
## Coefficients:
##              Estimate Std. Error t value     Pr(>|t|)
## (Intercept)  3.154808   0.291590   10.82 0.0000000001 ***
## Area        -0.000580   0.000148   -3.92      0.00065 ***
## Elevation    0.003541   0.000493    7.19 0.0000001983 ***
## Nearest      0.008826   0.010262    0.86      0.39829
## Scruz       -0.005709   0.003525   -1.62      0.11838
## Adjacent    -0.000663   0.000165   -4.01      0.00051 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 31.75)
##
##     Null deviance: 3510.73  on 29  degrees of freedom
## Residual deviance:  716.85  on 24  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```
plot(gal.quasi$linear.predictors, gal.quasi$residuals)
```



The canonical link for the Poisson is the log link. Identity links are used in rate models, see ELM §3.2.

In SM (p. 511) the suggestion is made that one route to an overdispersed Poisson would be to model Y as Poisson with mean $\mu\epsilon$, where ϵ is a latent random variable with expected value 1. If a gamma distribution with shape ν and mean 1 is postulated, then the marginal distribution of Y is negative binomial. The negative binomial model can be fit by maximum likelihood using `MASS::glm.nb`; see ELM 3.3.

Example 3 Gamma

I will use the `chimp` data set analysed in SM, Example 10.16.

Table 10.5 Times in minutes taken by four chimpanzees to learn ten words (Brown and Hollander, 1977, p. 257).

	Word									
Chimpanzee	1	2	3	4	5	6	7	8	9	10
1	178	60	177	36	225	345	40	2	287	14
2	78	14	80	15	10	115	10	12	129	80
3	99	18	20	25	15	54	25	10	476	55
4	297	20	195	18	24	420	40	15	372	190

```
data(chimps)
head(chimps)
```

```
##   chimp word   y
## 1     1    1 178
## 2     1    2  60
## 3     1    3 177
## 4     1    4  36
## 5     1    5 225
## 6     1    6 345
```

The response is time to learn each of 10 words, in minutes, and there are four chimpanzees. See p.485 of SM.

Davison's first analysis is a simple `lm(y ~ chimp + word, data = chimps)`, and then he applies Tukey's 1 df test for non-additivity (see Example 8.24) to confirm that the additive model is not a good fit.

```
model1 <- glm(y ~ chimp + word, family=Gamma(link = "log"), data = chimps)
options(digits=6)
anova(model1)
```

```
## Analysis of Deviance Table
##
## Model: Gamma, link: log
##
## Response: y
##
## Terms added sequentially (first to last)
##
##
##       Df Deviance Resid. Df Resid. Dev
## NULL                     39      60.38
## chimp  3     6.95        36      53.43
## word   9    38.46        27      14.97
```

This reproduces the first half of Table 10.6. The other side is the same thing with the factors introduced in the opposite order.

```
model2 <- glm(y ~ word + chimp, family = Gamma(link = "log"), data = chimps)
summary(model2)
```

```
##
## Call:
## glm(formula = y ~ word + chimp, family = Gamma(link = "log"),
##     data = chimps)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6766  -0.4510  -0.0488   0.3074   1.2847
##
## Coefficients:
##             Estimate Std. Error t value          Pr(>|t|)
## (Intercept)    5.458      0.375   14.54 0.000000000000027 ***
## word2         -1.771      0.466   -3.80           0.00074 ***
## word3         -0.365      0.466   -0.78           0.44007
## word4         -1.821      0.466   -3.91           0.00056 ***
## word5         -1.102      0.466   -2.37           0.02541 *
## word6          0.279      0.466    0.60           0.55461
## word7         -1.725      0.466   -3.70           0.00096 ***
## word8         -2.555      0.466   -5.49 0.000008278115332 ***
## word9          0.811      0.466    1.74           0.09301 .
## word10        -0.514      0.466   -1.10           0.27972
## chimp2        -0.973      0.295   -3.31           0.00269 **
## chimp3        -0.808      0.295   -2.74           0.01068 *
## chimp4        -0.113      0.295   -0.38           0.70479
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for Gamma family taken to be 0.433666)
##
##     Null deviance: 60.378  on 39  degrees of freedom
## Residual deviance: 14.972  on 27  degrees of freedom
## AIC: 418.4
##
## Number of Fisher Scoring iterations: 12
```

```
anova(model2)
```

```
## Analysis of Deviance Table
##
## Model: Gamma, link: log
##
## Response: y
##
## Terms added sequentially (first to last)
##
##
##       Df Deviance Resid. Df Resid. Dev
## NULL                     39      60.38
## word   9    39.19        30      21.19
## chimp  3     6.22        27      14.97
```

It makes a bit of difference, but not substantial. To test for a `chimp` effect we need an estimate of the dispersion parameter, which is obtained from `summary` as 0.434 (Davison reports .432).

The `chimp` test statistic is $(6.22/3)/0.434 = 4.77$ which gives a p-value of 0.008 when referred to an $F_{3,27}$ distribution. The `word` test statistic is $(39.19/9)/0.434$ about 10^{-6}.

```
test1 <- (6.22/3)/0.434
pf(test1, 3, 27, lower.tail = F)
```

```
## [1] 0.00849023
```

```
test2 <- (38.46/9)/0.434
pf(test2, 9, 27, lower.tail = F)
```

```
## [1] 0.00000166957
```

Davison suggests two other approaches to this data: an ordinary normal theory model fit to $\log(y)$, and a gamma model with the canonical link $1/\mu$. I'll let you explore those on your own.