R code and output of examples in text

Contents

1 Poisson regression

Number of children: log link

```
> birth <- read.table("Birth.csv",sep=",",header=T)
> birth.log <- glm( formula = children ~ age, family = poisson(link = log),data=birth)
> summary(birth.log)

Call:
glm(formula = children ~ age, family = poisson(link = log), data = birth)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0753  -0.9960  -0.7510   0.5358   2.8532

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.08955    0.71361  -5.731 1.00e-08 ***
age          0.11295    0.02121   5.326 1.00e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 194.42  on 140  degrees of freedom
Residual deviance: 165.01  on 139  degrees of freedom
AIC: 289.98

Number of Fisher Scoring iterations: 5

> anova(birth.log)
Analysis of Deviance Table

Model: poisson, link: log

Response: children

Terms added sequentially (first to last)


      Df Deviance Resid. Df Resid. Dev
NULL                    140    194.420
age    1   29.408       139    165.012
```

Number of children: identity link

R produces the following error message. Notice also the error message in the SAS output. Clearly there is a problem with this model.

```
> birth.id <- glm( formula = children ~ age, family = poisson(link = identity),data=birth)
Error: no valid set of coefficients has been found: please supply starting values
>
```

Diabetes deaths, categorical age

In order the read the data into R, `diabetes.xls` must be saved as `diabetes.csv`. `Gender` and `age` are both character variables in the data file, so R will treat them as categorical. The way that the model is specified is
`deaths` $\sim$ `gender + age`
The default base level in R is the lowest level, which is female gender and age <25. In order to reproduce the SAS output, we control the base level using the `C` function. In the case of `age`, for example, we want "45-54" to be the base level. This is the fourth level of `age`, so the term is specified in the model as `C(age,base=4)`.

```
> Diabetes <- read.table("diabetes.csv",sep=",",header=T)
> attach(Diabetes)
>
> ### categorical age
> Model1 <- glm(deaths ~ C(gender,base=2) + C(age,base=4), family = poisson(link = log), offset = l_popn)
> summary(Model1)

Call:
glm(formula = deaths ~ C(gender, base = 2) + C(age, base = 4),
    family = poisson(link = log), offset = l_popn)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.39640  -0.74227   0.01637   0.75061   1.06267

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)          -9.89155    0.16842 -58.732  < 2e-16 ***
C(gender, base = 2)1 -0.52331    0.06528  -8.017 1.09e-15 ***
C(age, base = 4)1    -2.89386    0.47726  -6.063 1.33e-09 ***
C(age, base = 4)2    -3.67022    1.01374  -3.620 0.000294 ***
C(age, base = 4)3    -0.99648    0.30732  -3.243 0.001185 **
C(age, base = 4)5     1.23566    0.19689   6.276 3.48e-10 ***
C(age, base = 4)6     2.33434    0.18155  12.858  < 2e-16 ***
C(age, base = 4)7     3.41836    0.17475  19.562  < 2e-16 ***
C(age, base = 4)8     4.30545    0.17827  24.151  < 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: 3306.383  on 15  degrees of freedom
Residual deviance:   10.889  on  7  degrees of freedom
AIC: 104.49

Number of Fisher Scoring iterations: 5
```

Diabetes deaths, cubic age

Polynomials are specified in R using the `poly` function.

```
> Model2 <- glm(deaths ~ C(gender,base=2) + poly(agemidpt,3), family = poisson(link = log), offset = l_popn)
> summary(Model2)

Call:
glm(formula = deaths ~ C(gender, base = 2) + poly(agemidpt, 3),
    family = poisson(link = log), offset = l_popn)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.29551  -0.75029  -0.03547   0.71023   1.29745

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)          -9.29873    0.10037 -92.645  < 2e-16 ***
C(gender, base = 2)1 -0.52327    0.06528  -8.016 1.09e-15 ***
poly(agemidpt, 3)1   10.06337    0.47696  21.099  < 2e-16 ***
poly(agemidpt, 3)2   -0.05436    0.37208  -0.146    0.884
poly(agemidpt, 3)3   -0.35669    0.21790  -1.637    0.102
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3306.383  on 15  degrees of freedom
Residual deviance:   15.334  on 11  degrees of freedom
AIC: 100.93

Number of Fisher Scoring iterations: 5
```

This gives different coefficients for the **agemidpt** polynomial to SAS. The SAS solution is reproduced as

```
> minage <- min(agemidpt)
> maxage <- max(agemidpt)
> agestd <- (agemidpt-0.5*(minage+maxage))/(0.5*(maxage-minage))
>
> Model3 <- glm(deaths ~ C(gender,base=2) + agestd + I(agestd^2) + I(agestd^3),
+ family = poisson(link = log), offset = l_popn)
> summary(Model3)

Call:
glm(formula = deaths ~ C(gender, base = 2) + agestd + I(agestd^2) +
    I(agestd^3), family = poisson(link = log), offset = l_popn)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.29551  -0.75029  -0.03547   0.71023   1.29745

Coefficients:
                     Estimate Std. Error  z value Pr(>|z|)
(Intercept)          -9.28316    0.08759 -105.978  < 2e-16 ***
C(gender, base = 2)1 -0.52327    0.06528   -8.016 1.09e-15 ***
agestd                4.17805    0.19271   21.681  < 2e-16 ***
I(agestd^2)          -0.03633    0.24866   -0.146    0.884
I(agestd^3)          -0.44370    0.27105   -1.637    0.102
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3306.383  on 15  degrees of freedom
Residual deviance:   15.334  on 11  degrees of freedom
AIC: 100.93

Number of Fisher Scoring iterations: 5
```

Third party claims

```
> TP <- read.table("Third party claims.csv",sep=",",header=T)
> attach(TP)
>
> model1 <- glm(claims ~ log(accidents), family=poisson, offset=log(population))
> summary(model1)

Call:
glm(formula = claims ~ log(accidents), family = poisson, offset = log(population))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-38.9573   -3.5507    0.1157    3.8422   45.9646

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -7.093809   0.026992 -262.81   <2e-16 ***
log(accidents)  0.259103   0.003376   76.75   <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: 22393  on 175  degrees of freedom
Residual deviance: 15837  on 174  degrees of freedom
AIC: 17066

Number of Fisher Scoring iterations: 4
```

2 Negative binomial regression

Negative binomial regression is in the MASS library, which must be installed and loaded. The function is `glm.nb`.

Third party claims

```
> library(MASS)
> model2 <- glm.nb(claims ~ log(accidents) + offset(log(population)))
> summary(model2)

Call:
glm.nb(formula = claims ~ log(accidents) + offset(log(population)),
    init.theta = 5.83093745788135, link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5448  -0.8172  -0.1964   0.4260   3.7295

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    -6.95443    0.15837  -43.91   <2e-16 ***
log(accidents)  0.25389    0.02472   10.27   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(5.8309) family taken to be 1)

    Null deviance: 298.16  on 175  degrees of freedom
Residual deviance: 192.33  on 174  degrees of freedom
AIC: 2041.3

Number of Fisher Scoring iterations: 1

Correlation of Coefficients:
               (Intercept)
log(accidents) -0.98


              Theta:  5.831
          Std. Err.:  0.671

 2 x log-likelihood:  -2035.255
```

The dispersion parameter is `Theta`=5.831. In SAS the dispersion parameter is given as 0.1715, which is 1/5.831.

Swedish mortality, categorical age and year

```
> mortality <- read.table("mortality.csv",header=T,sep=",")
> mortality <- mortality[,-c(3,5,7,9,11)]
> mortality <- na.omit(mortality)
> attach(mortality)
> library(MASS)
>
> model1 <- glm.nb(Male_death ~ factor(Age) + factor(Year) + offset(L_male_exp))
There were 50 or more warnings (use warnings() to see the first 50)
> summary(model1,corr=F)

Call:
glm.nb(formula = Male_death ~ factor(Age) + factor(Year) + offset(L_male_exp),
    init.theta = 113.809484987441, link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-7.5505  -0.6960  -0.0667   0.4994   6.7282

   [parameter estimates table omitted]
```

```
(Dispersion parameter for Negative Binomial(113.8095) family taken to be 1)

    Null deviance: 1711511  on 5867  degrees of freedom
Residual deviance:    7709  on 5704  degrees of freedom
AIC: 54027

Number of Fisher Scoring iterations: 1


              Theta:  113.81
          Std. Err.:  3.89

 2 x log-likelihood:  -53697.08
```

3 Quasi–likelihood regression

```
> model3 <- glm(claims ~ log(accidents), family=quasi(link="log",variance="mu"),
+ offset=log(population))
> summary(model3)

Call:
glm(formula = claims ~ log(accidents), family = quasi(link = "log",
    variance = "mu"), offset = log(population))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-38.9573   -3.5507    0.1157    3.8422   45.9646

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    -7.09381    0.27223 -26.058  < 2e-16 ***
log(accidents)  0.25910    0.03405   7.609 1.66e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasi family taken to be 101.7172)

    Null deviance: 22393  on 175  degrees of freedom
Residual deviance: 15837  on 174  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 4
```

4 Logistic regression

Vehicle insurance: quadratic vehicle value

```
> car <- read.table("car.csv",sep=",",header=T)
>
> model1 <- glm(clm ~ veh_value + I(veh_value^2), family=binomial, data=na.omit(car))
> summary(model1)

Call:
glm(formula = clm ~ veh_value + I(veh_value^2),
    family = binomial, data = na.omit(car))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.4109  -0.3870  -0.3722  -0.3573   3.1237

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -2.892566   0.044048 -65.668  < 2e-16 ***
veh_value       0.219591   0.035766   6.140 8.27e-10 ***
I(veh_value^2) -0.026039   0.005914  -4.403 1.07e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

    Null deviance: 33767  on 67855  degrees of freedom
Residual deviance: 33713  on 67853  degrees of freedom
AIC: 33719

Number of Fisher Scoring iterations: 6
```

Vehicle insurance: banded vehicle value

```
### create banded variable
> valuecat <- cut(car$veh_value, c(-1,2.5,5.0,7.5,10.0,12.5,100))
> table(valuecat)
valuecat
  (-1,2.5]    (2.5,5]    (5,7.5]   (7.5,10]  (10,12.5] (12.5,100]
     54971      11439       1265        104         44         33
>
> car <- cbind(car,valuecat)
>
> model2 <- glm(clm ~ factor(valuecat), family=binomial, data=na.omit(car))
> summary(model2)

Call:
glm(formula = clm ~ factor(valuecat), family = binomial,
    data = na.omit(car))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.4023  -0.3700  -0.3700  -0.3700   2.6444

Coefficients:
                           Estimate Std. Error  z value Pr(>|z|)
(Intercept)                -2.64749    0.01716 -154.272  < 2e-16 ***
factor(valuecat)(2.5,5]     0.17370    0.03891    4.464 8.04e-06 ***
factor(valuecat)(5,7.5]     0.10196    0.10962    0.930    0.352
factor(valuecat)(7.5,10]   -0.57139    0.51002   -1.120    0.263
factor(valuecat)(10,12.5]  -0.39703    0.72387   -0.548    0.583
factor(valuecat)(12.5,100] -0.81824    1.01432   -0.807    0.420
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 33767  on 67855  degrees of freedom
Residual deviance: 33744  on 67850  degrees of freedom
AIC: 33756

Number of Fisher Scoring iterations: 5
```

Vehicle insurance: full model, adjusted for exposure

```
> source("logit-exposure-adjusted.r")
> attach(car)
> model3 <- glm(clm ~ C(factor(agecat),base=3)+ C(factor(area),base=3) +
+ C(factor(veh_body),base=10) + factor(valuecat), family=binomial(logitexp(exposure)))
> summary(model3)

Call:
glm(formula = clm ~ C(factor(agecat), base = 3) +
    C(factor(area), base = 3) + C(factor(veh_body), base = 10) +
    factor(valuecat), family = binomial(logitexp(exposure)))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9970  -0.4480  -0.3390  -0.2149   3.9902

Coefficients:
```

```
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -1.74963    0.04875 -35.891  < 2e-16 ***
C(factor(agecat), base = 3)1     0.28764    0.06264   4.592 4.39e-06 ***
C(factor(agecat), base = 3)2     0.06435    0.05011   1.284 0.199075
C(factor(agecat), base = 3)4    -0.03600    0.04772  -0.754 0.450708
C(factor(agecat), base = 3)5    -0.26500    0.05567  -4.760 1.93e-06 ***
C(factor(agecat), base = 3)6    -0.25500    0.06694  -3.809 0.000139 ***
C(factor(area), base = 3)1      -0.03580    0.04519  -0.792 0.428240
C(factor(area), base = 3)2       0.05338    0.04699   1.136 0.255964
C(factor(area), base = 3)4      -0.13815    0.05846  -2.363 0.018125 *
C(factor(area), base = 3)5      -0.06636    0.06501  -1.021 0.307327
C(factor(area), base = 3)6       0.02086    0.07633   0.273 0.784617
C(factor(veh_body), base = 10)1  1.13627    0.44921   2.530 0.011422 *
C(factor(veh_body), base = 10)2 -0.37088    0.64132  -0.578 0.563056
C(factor(veh_body), base = 10)3  0.43332    0.14843   2.919 0.003507 **
C(factor(veh_body), base = 10)4 -0.01240    0.04314  -0.288 0.773709
C(factor(veh_body), base = 10)5  0.09897    0.10493   0.943 0.345548
C(factor(veh_body), base = 10)6  0.59606    0.32771   1.819 0.068928 .
C(factor(veh_body), base = 10)7 -0.11119    0.17178  -0.647 0.517448
C(factor(veh_body), base = 10)8  0.01941    0.14484   0.134 0.893375
C(factor(veh_body), base = 10)9  0.06962    0.80135   0.087 0.930773
C(factor(veh_body), base = 10)11 -0.01913   0.04995  -0.383 0.701781
C(factor(veh_body), base = 10)12 -0.09668   0.10823  -0.893 0.371722
C(factor(veh_body), base = 10)13 -0.24555   0.07599  -3.232 0.001231 **
factor(valuecat)(2.5,5]          0.21017    0.04936   4.258 2.06e-05 ***
factor(valuecat)(5,7.5]          0.13652    0.12366   1.104 0.269612
factor(valuecat)(7.5,10]        -0.60664    0.53884  -1.126 0.260239
factor(valuecat)(10,12.5]       -0.29001    0.77292  -0.375 0.707503
factor(valuecat)(12.5,100]      -0.79721    1.07082  -0.744 0.456582
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 33767  on 67855  degrees of freedom
Residual deviance: 32494  on 67828  degrees of freedom
AIC: 32550

Number of Fisher Scoring iterations: 4
```

Vehicle insurance: logistic regression on grouped data

```
> ### grouped data
> car.group <- read.table("car_grouped.csv",sep=",",header=T)
>
> ### the response is a two-column matrix
> ### the first column is the number of successes (claims)
> ### the second column is the number of failures (number-claims)
>
> model4 <- glm(cbind(claims,number-claims) ~ C(factor(agecat),base=6)+ C(factor(area),base=6) +
+ C(factor(veh_body),base=13) + factor(valuecat),
+ family=binomial, data=car.group)
> summary(model4)

Call:
glm(formula = cbind(claims, number - claims) ~ C(factor(agecat),
    base = 6) + C(factor(area), base = 6) + C(factor(veh_body),
    base = 13) + factor(valuecat), family = binomial, data = car.group)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5699  -0.7053  -0.3750   0.3799   3.8452

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -2.588035   0.045106 -57.377  < 2e-16 ***
C(factor(agecat), base = 6)1     0.229595   0.056844   4.039 5.37e-05 ***
C(factor(agecat), base = 6)2     0.026098   0.046220   0.565 0.572305
C(factor(agecat), base = 6)3    -0.031849   0.044208  -0.720 0.471259
C(factor(agecat), base = 6)4    -0.221561   0.052145  -4.249 2.15e-05 ***
C(factor(agecat), base = 6)5    -0.232433   0.062866  -3.697 0.000218 ***
```

```
C(factor(area), base = 6)1      -0.037123   0.041887  -0.886 0.375480
C(factor(area), base = 6)2       0.059338   0.043393   1.367 0.171484
C(factor(area), base = 6)3      -0.127991   0.054437  -2.351 0.018715 *
C(factor(area), base = 6)4      -0.052929   0.060233  -0.879 0.379545
C(factor(area), base = 6)5       0.067663   0.070275   0.963 0.335632
C(factor(veh_body), base = 13)1   1.077394   0.372472   2.893 0.003821 **
C(factor(veh_body), base = 13)2  -0.490457   0.604609  -0.811 0.417252
C(factor(veh_body), base = 13)3   0.252473   0.130502   1.935 0.053036 .
C(factor(veh_body), base = 13)4  -0.014328   0.040026  -0.358 0.720371
C(factor(veh_body), base = 13)5   0.158445   0.096656   1.639 0.101158
C(factor(veh_body), base = 13)6   0.557646   0.285901   1.950 0.051118 .
C(factor(veh_body), base = 13)7  -0.165132   0.159956  -1.032 0.301902
C(factor(veh_body), base = 13)8   0.178233   0.135608   1.314 0.188739
C(factor(veh_body), base = 13)9  -0.049655   0.737682  -0.067 0.946334
C(factor(veh_body), base = 13)10 -0.008798   0.046188  -0.190 0.848937
C(factor(veh_body), base = 13)11 -0.058342   0.100757  -0.579 0.562565
C(factor(veh_body), base = 13)12 -0.250009   0.071095  -3.517 0.000437 ***
factor(valuecat)2                0.173212   0.045314   3.822 0.000132 ***
factor(valuecat)3                0.084221   0.113544   0.742 0.458238
factor(valuecat)4               -0.551497   0.515900  -1.069 0.285069
factor(valuecat)5               -0.343446   0.732547  -0.469 0.639185
factor(valuecat)6               -0.778498   1.021499  -0.762 0.445992
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1010.82  on 928  degrees of freedom
Residual deviance:  868.38  on 901  degrees of freedom
AIC: 2414.3

Number of Fisher Scoring iterations: 5
```

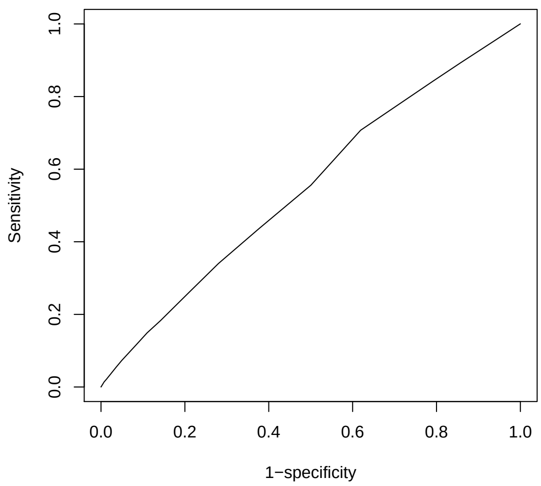
ROC curves and AUC

The AUC is easily computed using the `somers2` function in the `Hmisc` package, which needs to be downloaded from the CRAN website. A function `ROC` for computing and plotting the ROC curve, is given on the book website in file ROC-function.r.

```
> car <- read.table("car.csv",sep=",",header=T)
> valuecat <- cut(car$veh_value, c(-1,2.5,5.0,7.5,10.0,12.5,100))
> car <- cbind(car,valuecat)
> attach(car)

        The following object(s) are masked _by_ .GlobalEnv :

         valuecat

>
> library(Hmisc)  ### need this for somers2 function to compute AUC

Attaching package: 'Hmisc'


        The following object(s) are masked from package:base :

         format.pval

        The following object(s) are masked from package:base :

         round.POSIXt

        The following object(s) are masked from package:base :

         trunc.POSIXt

Warning message:
package 'Hmisc' was built under R version 2.6.0
> source("ROC-function.r")  ### from book website; for plotting ROC curve
```

```
>
> model5 <- glm(clm ~ C(factor(agecat),base=3)+ C(factor(area),base=3) +
+ C(factor(veh_body),base=10) + factor(valuecat), family=binomial)
>
> ## compute fitted values from logistic regression and store in fittedvalues
> fittedvalues <- predict(model5, type = 'response', newdata = car)
> somers2(fittedvalues,clm)
           C          Dxy            n      Missing
5.484406e-01 9.688118e-02 6.785600e+04 0.000000e+00
> ROC(fittedvalues,clm)
```

The AUC is given as the element "`C`" of the `somers2` result, which is 0.5484406.



5 Ordinal regression

Proportional odds model

A few functions for this model are available. We prefer `vglm` in the `VGAM` package. The `VGAM` manual is worth consulting before attempting to implement the next three models.

```
> injury <- read.table("injury.csv",sep=",",header=T)
> attach(injury)
> library(VGAM)
Loading required package: splines
Loading required package: stats4

Attaching package: 'VGAM'

  [warnings omitted]

>
> ## change base levels to those in the text
> ## (not necessary, this is just to demonstrate that the solution is the same
> ##  as the SAS solution)
> road.x <- C(factor(roaduserclass),base=4)
> age.x <- C(factor(agecat),base=7)
> sex.x <- C(sex,base=2)
>
> model1 <- vglm(degree ~ road.x + age.x + sex.x + age.x*sex.x, cumulative(parallel=TRUE),
+ weights=number)
> summary(model1)

Call:
vglm(formula = degree ~ road.x + age.x + sex.x + age.x * sex.x,
    family = cumulative(parallel = TRUE), weights = number)
```

```
Pearson Residuals:
                   Min     1Q   Median     3Q     Max
logit(P[Y<=1]) -67.695 -4.915 -0.50658 5.0683 52.8488
logit(P[Y<=2]) -99.385 -3.218  0.52742 1.4541  6.4659

Coefficients:
                  Value Std. Error   t value
(Intercept):1  0.470450   0.021164  22.22836
(Intercept):2  5.049181   0.045126 111.88996
road.x1       -0.150587   0.026949  -5.58786
road.x2       -0.296705   0.036494  -8.13032
road.x3       -2.448987   0.056805 -43.11235
age.x1         0.178933   0.032280   5.54318
age.x2         0.112220   0.032093   3.49676
age.x3         0.058066   0.036395   1.59543
age.x4        -0.054979   0.029950  -1.83569
age.x5        -0.069905   0.033072  -2.11372
age.x6        -0.150468   0.034439  -4.36917
sex.x1        -0.171892   0.033421  -5.14329
age.x1:sex.x1 -0.129016   0.053129  -2.42837
age.x2:sex.x1 -0.117903   0.052178  -2.25963
age.x3:sex.x1 -0.041927   0.060111  -0.69749
age.x4:sex.x1 -0.028363   0.048663  -0.58285
age.x5:sex.x1 -0.018351   0.055568  -0.33025
age.x6:sex.x1  0.148296   0.059477   2.49331

Number of linear predictors:  2

Names of linear predictors: logit(P[Y<=1]), logit(P[Y<=2])

Dispersion Parameter for cumulative family:   1

Residual Deviance: 107703.4 on 400 degrees of freedom

Log-likelihood: -53851.68 on 400 degrees of freedom

Number of Iterations: 7
```

Partial proportional odds model

We use `vglm` for this model. The partial proportional odds are specified via the `parallel` parameter.

```
> model2 <- vglm(degree ~ road.x + age.x + sex.x + age.x*sex.x,
+ cumulative(parallel=TRUE~age.x*sex.x-1),
+ weights=number)
> summary(model2)

Call:
vglm(formula = degree ~ road.x + age.x + sex.x + age.x * sex.x,
    family = cumulative(parallel = TRUE ~ age.x * sex.x - 1),
    weights = number)

Pearson Residuals:
                    Min      1Q   Median     3Q     Max
logit(P[Y<=1])  -67.644 -4.0605 -0.55117 5.3121 52.8750
logit(P[Y<=2]) -101.307 -5.1598  0.67481 2.0158  5.0855

Coefficients:
                  Value Std. Error   t value
(Intercept):1  0.469735   0.021256  22.09936
(Intercept):2  5.087744   0.052529  96.85659
road.x1:1     -0.139587   0.027039  -5.16246
road.x1:2     -0.783784   0.117927  -6.64633
road.x2:1     -0.255168   0.036679  -6.95672
road.x2:2     -1.587843   0.112076 -14.16754
road.x3:1     -2.865563   0.077568 -36.94261
road.x3:2     -1.545257   0.138845 -11.12935
age.x1         0.180558   0.032440   5.56599
```

```
age.x2         0.113798   0.032287   3.52460
age.x3         0.058887   0.036625   1.60784
age.x4        -0.055497   0.030094  -1.84412
age.x5        -0.070165   0.033195  -2.11374
age.x6        -0.150202   0.034513  -4.35211
sex.x1        -0.172007   0.033500  -5.13451
age.x1:sex.x1 -0.130359   0.053249  -2.44810
age.x2:sex.x1 -0.119478   0.052324  -2.28341
age.x3:sex.x1 -0.042829   0.060286  -0.71043
age.x4:sex.x1 -0.027702   0.048775  -0.56796
age.x5:sex.x1 -0.018043   0.055661  -0.32417
age.x6:sex.x1  0.148231   0.059539   2.48966

Number of linear predictors:  2

Names of linear predictors: logit(P[Y<=1]), logit(P[Y<=2])

Dispersion Parameter for cumulative family:   1

Residual Deviance: 107447.5 on 397 degrees of freedom

Log-likelihood: -53723.73 on 397 degrees of freedom

Number of Iterations: 7
```

6 Nominal regression

As the private health insurance data are not publicly available, nominal regression is illustrated here on the degree of injury data. The `vglm` function in the `VGAM` package is used.

```
> injury <- read.table("injury.csv",sep=",",header=T)
> attach(injury)
> library(VGAM)
Loading required package: splines
Loading required package: stats4

Attaching package: 'VGAM'

   [warnings omitted]

>
> ## change base levels to those in the text
> road.x <- C(factor(roaduserclass),base=4)
> age.x <- C(factor(agecat),base=7)
> sex.x <- C(sex,base=2)
> ## nominal regression model
> model3 <- vglm(degree ~ road.x + age.x + sex.x + age.x*sex.x,
+ multinomial, weights=number)
> summary(model3)

Call:
vglm(formula = degree ~ road.x + age.x + sex.x + age.x * sex.x,
    family = multinomial, weights = number)

Pearson Residuals:
                       Min      1Q  Median     3Q    Max
log(mu[,1]/mu[,3]) -61.896 -11.492 -2.4095 4.2989 40.455
log(mu[,2]/mu[,3]) -59.786 -11.386 -2.9678 3.7426 51.449

Coefficients:
                   Value Std. Error   t value
(Intercept):1    4.646716    0.11289  41.16166
(Intercept):2    4.164019    0.11308  36.82403
road.x1:1       -0.738118    0.12217  -6.04174
road.x1:2       -0.612441    0.12271  -4.99102
road.x2:1       -1.588331    0.12158 -13.06392
road.x2:2       -1.383920    0.12230 -11.31545
road.x3:1       -3.436460    0.15939 -21.56065
road.x3:2       -0.583524    0.14331  -4.07168
```

```
age.x1:1        -0.180901    0.16340  -1.10710
age.x1:2        -0.376460    0.16381  -2.29819
age.x2:1        -0.025761    0.16474  -0.15638
age.x2:2        -0.146057    0.16500  -0.88521
age.x3:1        -0.109392    0.17826  -0.61368
age.x3:2        -0.176618    0.17850  -0.98943
age.x4:1        -0.084816    0.14850  -0.57114
age.x4:2        -0.030285    0.14869  -0.20368
age.x5:1        -0.276066    0.15563  -1.77389
age.x5:2        -0.214588    0.15590  -1.37643
age.x6:1        -0.777363    0.15317  -5.07510
age.x6:2        -0.657339    0.15358  -4.28014
sex.x1:1         0.092248    0.20859   0.44224
sex.x1:2         0.270256    0.20887   1.29387
age.x1:sex.x1:1  0.108107    0.32552   0.33210
age.x1:sex.x1:2  0.250028    0.32612   0.76667
age.x2:sex.x1:1  0.116827    0.33374   0.35006
age.x2:sex.x1:2  0.244845    0.33417   0.73269
age.x3:sex.x1:1  0.628953    0.43165   1.45707
age.x3:sex.x1:2  0.688673    0.43208   1.59386
age.x4:sex.x1:1 -0.091268    0.29346  -0.31101
age.x4:sex.x1:2 -0.063575    0.29380  -0.21639
age.x5:sex.x1:1 -0.245403    0.30780  -0.79727
age.x5:sex.x1:2 -0.230736    0.30829  -0.74845
age.x6:sex.x1:1  0.356051    0.32879   1.08290
age.x6:sex.x1:2  0.226424    0.32948   0.68721

Number of linear predictors:  2

Names of linear predictors: log(mu[,1]/mu[,3]), log(mu[,2]/mu[,3])

Dispersion Parameter for multinomial family:   1

Residual Deviance: 107390.7 on 384 degrees of freedom

Log-likelihood: -53695.36 on 384 degrees of freedom

Number of Iterations: 7
```

7 Gamma regression

Vehicle insurance

```
> car <- read.table("car.csv",sep=",",header=T)
>
> #### banded vehicle value
> valuecat <- cut(car$veh_value, c(-1,2.5,5.0,7.5,10.0,12.5,100))
>
> #### create variables with same base levels as in the text
> age.x <- C(factor(car$agecat),base=3) ## agecat=3 base level
> area.x <- C(factor(car$area),base=3)  ## area C is 3rd level
> gender.x <- C(factor(car$gender),base=2)  ## gender M is 2nd level
> veh_body.x <- C(factor(car$veh_body),base=10) ## SEDAN is 10th level
>
>
> car <- cbind(car,valuecat, age.x,area.x,gender.x,veh_body.x)
>
> model1 <- glm(claimcst0 ~ age.x + gender.x + age.x*gender.x + area.x + veh_body.x,
+ family=Gamma(link="log"),data=subset(car,clm==1))
> summary(model1)

Call:
glm(formula = claimcst0 ~ age.x + gender.x + age.x * gender.x +
    area.x + veh_body.x, family = Gamma(link = "log"), data = subset(car,
    clm == 1))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.01135  -1.35447  -0.80756   0.07785   6.61857
```

```
Coefficients:
  [output omitted]

(Dispersion parameter for Gamma family taken to be 2.844378)

    Null deviance: 7379.9  on 4623  degrees of freedom
Residual deviance: 7172.5  on 4595  degrees of freedom
AIC: 79321

Number of Fisher Scoring iterations: 7
```

Personal injury insurance, no adjustment for quickly settled claims

```
> persinj <-  read.table("persinj.csv",sep=",",header=T)
>
> model3 <- glm(total ~ op_time + factor(legrep) + op_time*factor(legrep),
+ family=Gamma(link="log"), data=persinj)
> summary(model3)

Call:
glm(formula = total ~ op_time + factor(legrep) + op_time * factor(legrep),
    family = Gamma(link = "log"), data = persinj)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6253  -0.9860  -0.4332   0.1345   9.9012

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              8.2118447  0.0329095 249.528  < 2e-16 ***
op_time                  0.0383149  0.0006311  60.707  < 2e-16 ***
factor(legrep)1          0.4667863  0.0424613  10.993  < 2e-16 ***
op_time:factor(legrep)1 -0.0049978  0.0008002  -6.246 4.29e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Gamma family taken to be 2.432031)

    Null deviance: 44010  on 22035  degrees of freedom
Residual deviance: 25412  on 22032  degrees of freedom
AIC: 490944

Number of Fisher Scoring iterations: 6
```

Runoff triangle

```
> runoff <- read.table("runoff triangle.csv",sep=",",header=T)
> runoff$Y[runoff$Y<0] <- 1  ### replace negative value by 1
>
> model4 <- glm(Y ~ factor(devyear) + factor(accyear), family=Gamma(link="log"), data=runoff)
> summary(model4)

Call:
glm(formula = Y ~ factor(devyear) + factor(accyear), family = Gamma(link = "log"),
    data = runoff)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.3067  -0.4424   0.0000   0.2562   0.9835

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        7.740643   0.321184  24.100  < 2e-16 ***
factor(devyear)2   0.752974   0.313497   2.402  0.02160 *
factor(devyear)3   0.757988   0.327863   2.312  0.02662 *
factor(devyear)4   0.324628   0.343546   0.945  0.35099
factor(devyear)5   0.160408   0.362197   0.443  0.66051
factor(devyear)6  -0.122756   0.385846  -0.318  0.75221
```

```
factor(devyear)7  -1.075185   0.417942  -2.573  0.01436 *
factor(devyear)8  -1.252244   0.465613  -2.689  0.01078 *
factor(devyear)9  -1.872183   0.547466  -3.420  0.00157 **
factor(devyear)10 -2.593149   0.738525  -3.511  0.00122 **
factor(accyear)2  -0.199962   0.313497  -0.638  0.52761
factor(accyear)3   0.089378   0.327863   0.273  0.78671
factor(accyear)4   0.317248   0.343546   0.923  0.36192
factor(accyear)5   0.152780   0.362197   0.422  0.67567
factor(accyear)6  -0.172764   0.385846  -0.448  0.65701
factor(accyear)7  -0.359414   0.417942  -0.860  0.39550
factor(accyear)8  -0.003548   0.465613  -0.008  0.99396
factor(accyear)9  -0.091333   0.547466  -0.167  0.86844
factor(accyear)10 -0.108726   0.738525  -0.147  0.88378
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Gamma family taken to be 0.4422604)

    Null deviance: 57.280  on 54  degrees of freedom
Residual deviance: 31.720  on 36  degrees of freedom
AIC: 991.83

Number of Fisher Scoring iterations: 11
```

8 Inverse Gaussian regression

The data frame `car` used here is the one created for the vehicle insurance, Gamma regression model.

```
> model2 <- glm(claimcst0 ~ age.x + gender.x + area.x,
+ family=inverse.gaussian(link="log"),data=subset(car,clm==1))
> summary(model2)

Call:
glm(formula = claimcst0 ~ age.x + gender.x + area.x, family = inverse.gaussian(link = "log"),
    data = subset(car, clm == 1))

Deviance Residuals:
      Min         1Q     Median         3Q        Max
-0.066235  -0.043358  -0.021932   0.001744   0.121605

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.68300    0.07224 106.352  < 2e-16 ***
age.x1       0.25110    0.09950   2.524  0.01164 *
age.x2       0.09266    0.07664   1.209  0.22676
age.x4      -0.00533    0.07125  -0.075  0.94037
age.x5      -0.12129    0.08140  -1.490  0.13626
age.x6      -0.06755    0.09890  -0.683  0.49461
gender.x1   -0.15283    0.05119  -2.986  0.00285 **
area.x1     -0.07289    0.06806  -1.071  0.28425
area.x2     -0.10265    0.06976  -1.471  0.14124
area.x4     -0.09781    0.08632  -1.133  0.25725
area.x5      0.06951    0.10169   0.684  0.49431
area.x6      0.28250    0.12885   2.192  0.02840 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for inverse.gaussian family taken to be 0.001464282)

    Null deviance: 6.4422  on 4623  degrees of freedom
Residual deviance: 6.3765  on 4612  degrees of freedom
AIC: 77162

Number of Fisher Scoring iterations: 11
```

9 Logistic regression GLMM

The software in this area is developing very rapidly. We use here `glmmPQL` in the `MASS` package.

```
> claimslong <- read.table("claimslong.txt",header=T,sep=",")
> ## create binary variable for claim/no claim
> claimslong <- cbind(claimslong,clm=1*(claimslong$numclaims>0))
>
> #### create variables with same base levels as in the text, for comparability
> age.x <- C(factor(claimslong$agecat),base=6)
> value.x <- C(factor(claimslong$valuecat),base=6)
> period.x <- C(factor(claimslong$period),base=3)
> claimslong <- cbind(claimslong,age.x,value.x,period.x)
>
>
> library(MASS)
> model1 <- glmmPQL(clm ~ age.x + value.x + period.x,
+ random=~1|policyID, family=binomial, data=claimslong)
Loading required package: nlme
iteration 1
iteration 2
iteration 3
iteration 4
iteration 5
iteration 6
iteration 7
iteration 8
> summary(model1)
Linear mixed-effects model fit by maximum likelihood
 Data: claimslong
  AIC BIC logLik
   NA  NA     NA

Random effects:
 Formula: ~1 | policyID
        (Intercept)  Residual
StdDev:    2.124486 0.5923166

Variance function:
 Structure: fixed weights
 Formula: ~invwt
Fixed effects: clm ~ age.x + value.x + period.x
                 Value Std.Error    DF   t-value p-value
(Intercept) -2.4995759 0.0307967 79998 -81.16380  0.0000
age.x1       0.2427175 0.0535794 39989   4.53006  0.0000
age.x2       0.0075297 0.0421641 39989   0.17858  0.8583
age.x3      -0.0471549 0.0401732 39989  -1.17379  0.2405
age.x4      -0.2369429 0.0455938 39989  -5.19682  0.0000
age.x5      -0.1966081 0.0534377 39989  -3.67920  0.0002
value.x1     0.2090895 0.0362665 39989   5.76536  0.0000
value.x2     0.0748306 0.1029381 39989   0.72695  0.4673
value.x3    -0.7577450 0.3999978 39989  -1.89437  0.0582
value.x4    -0.4847632 0.6189636 39989  -0.78319  0.4335
value.x5    -1.2043126 0.6933970 39989  -1.73683  0.0824
period.x1   -0.3376763 0.0154086 79998 -21.91482  0.0000
period.x2   -0.1921770 0.0151812 79998 -12.65885  0.0000
 Correlation:
   [correlation matrix omitted]

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-2.2540306 -0.2959930 -0.2688550 -0.2481972  3.0226618

Number of Observations: 120000
Number of Groups: 40000
```

Parameter estimates are similar to those produced by SAS. They are not identical because `proc nlmixed` and `glmmPQL` use different methods for finding the maximum likelihood solution.

10 Logistic regression GEE

As for GLMMs, software for these models is evolving constantly. We use `geeglm` in the `geepack` package, which gives identical parameter estimates to `proc genmod`.

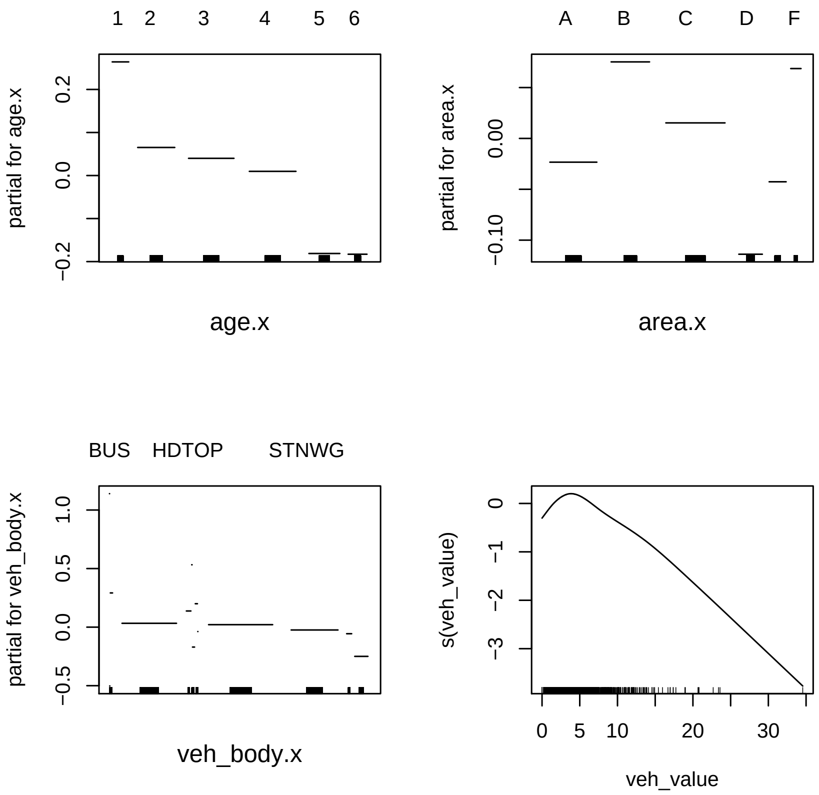
```
> model2 <- geeglm(clm ~ age.x + value.x + period.x,
+ id=policyID, corstr="exchangeable", family=binomial, data=claimslong)
> summary(model2)

Call:
geeglm(formula = clm ~ age.x + value.x + period.x, family = binomial,
    data = claimslong, id = policyID, corstr = "exchangeable")

 Coefficients:
                Estimate    Std.err         Wald        p(>W)
(Intercept) -1.683726369 0.02465746 4.662794e+03 0.000000e+00
age.x1       0.188904924 0.04081014 2.142646e+01 3.676616e-06
age.x2       0.004911148 0.03253987 2.277899e-02 8.800332e-01
age.x3      -0.036162990 0.03114110 1.348531e+00 2.455351e-01
age.x4      -0.195199994 0.03568454 2.992261e+01 4.496400e-08
age.x5      -0.149713839 0.04222537 1.257121e+01 3.917353e-04
value.x1     0.161274753 0.02775222 3.377048e+01 6.201289e-09
value.x2     0.059411809 0.07924582 5.620732e-01 4.534261e-01
value.x3    -0.645585908 0.31285283 4.258218e+00 3.906087e-02
value.x4    -0.236793478 0.58107245 1.660653e-01 6.836326e-01
value.x5    -0.968796136 0.61588836 2.474348e+00 1.157174e-01
period.x1   -0.205116372 0.01663843 1.519763e+02 0.000000e+00
period.x2   -0.116052407 0.01611967 5.183178e+01 6.046275e-13

Estimated Scale Parameters:
            Estimate    Std.err
(Intercept) 1.000044 0.01466313

Correlation: Structure = exchangeable  Link = identity

Estimated Correlation Parameters:
       Estimate     Std.err
alpha 0.3316776 0.007854693
Number of clusters:   40000   Maximum cluster size: 3
```

11 Logistic regression GAM

GAMs can be fitted using either the special–purpose `gam` package, or the more general `gamlss` package. We illustrate the use of both.

```
> ######## vehicle insurance data
> car <- read.table("car.csv",sep=",",header=T)
>
> #### banded vehicle value
> valuecat <- cut(car$veh_value, c(-1,2.5,5.0,7.5,10.0,12.5,100))
>
> #### create variables with same base levels as in the text
> age.x <- C(factor(car$agecat),base=3) ## agecat=3 base level
> area.x <- C(factor(car$area),base=3)  ## area C is 3rd level
> gender.x <- C(factor(car$gender),base=2)  ## gender M is 2nd level
> veh_body.x <- C(factor(car$veh_body),base=10) ## SEDAN is 10th level
>
> car <- cbind(car,valuecat, age.x,area.x,gender.x,veh_body.x)
>
> ### use gam in gam package:
> library(gam)
Loading required package: splines
> model1 <- gam(clm ~ age.x + area.x + veh_body.x + s(veh_value),
+ family=binomial, data=car)
> summary(model1)

Call: gam(formula = clm ~ age.x + area.x + veh_body.x + s(veh_value),
    family = binomial, data = car)
```

```
Deviance Residuals:
    Min      1Q  Median      3Q     Max
-0.7957 -0.3954 -0.3695 -0.3434  2.6809

(Dispersion Parameter for binomial family taken to be 1)

    Null Deviance: 33766.8 on 67855 degrees of freedom
Residual Deviance: 33588.83 on 67829 degrees of freedom
AIC: 33642.83

Number of Local Scoring Iterations: 7

DF for Terms and Chi-squares for Nonparametric Effects

             Df Npar Df Npar Chisq    P(Chi)
(Intercept)   1
age.x         5
area.x        5
veh_body.x   12
s(veh_value)  1       3     29.909 1.443e-06
> par(mfrow=c(2,2))
> plot(model1)
```



The highly nonlinear effect of vehicle value, with a peak around 4 ($40 000), is seen clearly.

The `gamlss` implementation gives parameter estimates for the parametric explanatory variables, which are similar to those given by `proc gam`.

```
> ### use gamlss:
> library(gamlss)
Loading required package: splines
 **********   GAMLSS Version 1.6-0 **********
For more on GAMLSS look at http://www.londonmet.ac.uk/gamlss/
Type gamlssNews() to see new features/changes/bug fixes.
```

```
> model2 <- gamlss(clm ~ age.x + area.x + veh_body.x + cs(veh_value),
+ family=BI, data=car)
GAMLSS-RS iteration 1: Global Deviance = 33588.83
GAMLSS-RS iteration 2: Global Deviance = 33588.83
> summary(model2)
*******************************************************************
Family:  c("BI", "Binomial")

Call:  gamlss(formula = clm ~ age.x + area.x + veh_body.x + cs(veh_value),
    family = BI, data = car)

Fitting method: RS()

-------------------------------------------------------------------
Mu link function:  logit
Mu Coefficients:
               Estimate  Std. Error  t value   Pr(>|t|)
(Intercept)    -2.67836     0.04949  -54.1184  0.000e+00
age.x1          0.22410     0.05702    3.9299  8.506e-05
age.x2          0.02523     0.04645    0.5433  5.869e-01
age.x4         -0.03027     0.04435   -0.6827  4.948e-01
age.x5         -0.22114     0.05228   -4.2298  2.342e-05
age.x6         -0.22280     0.06286   -3.5444  3.937e-04
area.x1        -0.03861     0.04194   -0.9206  3.573e-01
area.x2         0.05996     0.04347    1.3794  1.678e-01
area.x4        -0.12911     0.05467   -2.3616  1.820e-02
area.x5        -0.05792     0.06060   -0.9557  3.392e-01
area.x6         0.05342     0.07111    0.7512  4.525e-01
veh_body.x1     1.11844     0.37123    3.0128  2.590e-03
veh_body.x2    -0.52367     0.46575   -1.1244  2.609e-01
veh_body.x3     0.27095     0.12760    2.1234  3.372e-02
veh_body.x4     0.01172     0.04006    0.2926  7.698e-01
veh_body.x5     0.11676     0.09810    1.1902  2.340e-01
veh_body.x6     0.51143     0.28743    1.7794  7.519e-02
veh_body.x7    -0.19124     0.16196   -1.1807  2.377e-01
veh_body.x8     0.17880     0.13599    1.3148  1.886e-01
veh_body.x9    -0.05886     0.71090   -0.0828  9.340e-01
veh_body.x11   -0.04548     0.04487   -1.0136  3.108e-01
veh_body.x12   -0.07733     0.10140   -0.7626  4.457e-01
veh_body.x13   -0.27078     0.07146   -3.7891  1.513e-04
cs(veh_value)   0.06975     0.01330    5.2460  1.559e-07

-------------------------------------------------------------------
No. of observations in the fit:  67856
Degrees of Freedom for the fit:  27.00079
      Residual Deg. of Freedom:  67829
                      at cycle:  2

Global Deviance:     33588.83
            AIC:     33642.83
            SBC:     33889.22
*******************************************************************
Warning message:
addive terms exists in the mu formula results maybe are not appropriate in: vcov.gamlss(object, "all")
```