Generalised Linear Mixed Models practical

O Rodriguez de Rivera Ortega, PhD
SE@K (Statistical Ecology @ Kent), University of Kent

10/11/2020

```r
KW <- read.csv("~/Google Drive/Course/data/pollen.csv")
names(KW)
```

```
## [1] "Time"      "Hive"      "Treatment" "Dandelion" "X"
```

```r
str(KW)
```

```
## 'data.frame':    60 obs. of  5 variables:
##  $ Time     : int  1 2 3 4 1 2 3 4 1 2 ...
##  $ Hive     : int  1 1 1 1 2 2 2 2 3 3 ...
##  $ Treatment: chr  "Syrup" "Syrup" "Syrup" "Syrup" ...
##  $ Dandelion: int  236 167 227 225 306 308 241 304 36 10 ...
##  $ X        : logi  NA NA NA NA NA NA ...
```

```r
#Load packages and library files
library(lattice)  #Needed for multi-panel graphs
library(lme4)
```

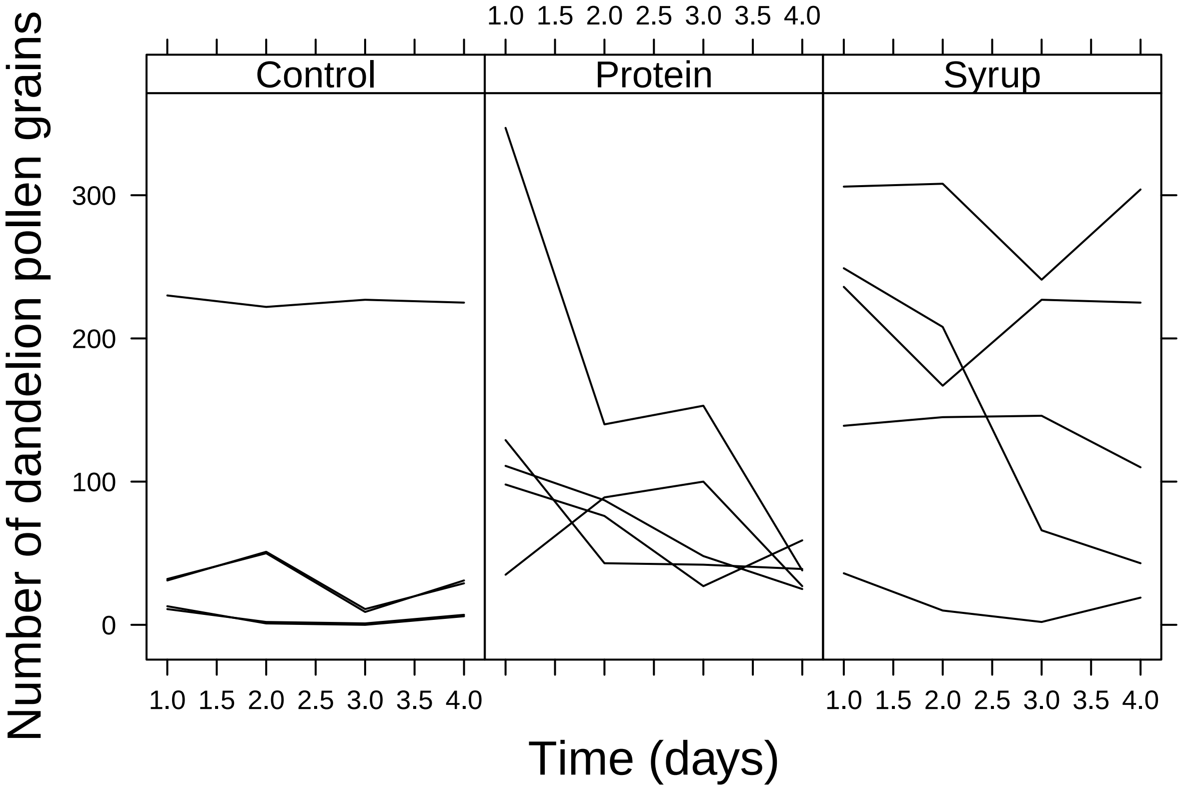
```
## Loading required package: Matrix
```

#House keeping

```r
KW$fHive <- factor(KW$Hive)
```

#Data exploration

```r
xyplot(Dandelion ~ Time | Treatment,
       xlab = list("Time (days)", cex = 1.5),
       ylab = list("Number of dandelion pollen grains", cex = 1.5),
       data = KW, layout = c(3,1),
       groups = Hive,
       type = "l", col = 1,
       strip = strip.custom(bg = 'white',
                            par.strip.text = list(cex = 1.2)),
       scales = list(alternating = T,
                     x = list(relation = "same"),
                     y = list(relation = "same"))
)
```



Building the model

$$D_{ij} \sim Poisson(\mu_{ij})$$

$$\log(\mu_{ij}) = Time_{ij} + Treatment_{ij} + Treatment_{ij} \times Time_{ij} + a_i$$

$$a_i \sim N(0, \sigma^2_{Hive})$$

The model uses 6 regression parameters (1 intercept, 1 slope for Time, 2 slopes for Treatment and 2 slopes for their interaction) and one variance term for the variance of the random intercept `Hive`

```
M1 <- glmer(Dandelion ~ Time * Treatment + (1|fHive),
            data = KW, family = poisson)

print(summary(M1), digits = 2, signif.stars=FALSE)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: Dandelion ~ Time * Treatment + (1 | fHive)
##    Data: KW
##
##      AIC      BIC   logLik deviance df.resid
##   1035.6   1050.3   -510.8   1021.6       53
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
##  -6.85  -1.91  -0.12   1.98   7.88
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  fHive  (Intercept) 1        1
## Number of obs: 60, groups:  fHive, 15
##
## Fixed effects:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)               3.232      0.463     7.0    3e-12
## Time                     -0.045      0.026    -1.8    0.080
## TreatmentProtein          2.043      0.650     3.1    0.002
## TreatmentSyrup            1.804      0.650     2.8    0.005
## Time:TreatmentProtein    -0.360      0.035   -10.4   <2e-16
## Time:TreatmentSyrup      -0.074      0.030    -2.4    0.015
##
## Correlation of Fixed Effects:
##             (Intr) Time   TrtmnP TrtmnS Tm:TrP
## Time        -0.137
## TretmntPrtn -0.712  0.097
## TretmntSyrp -0.712  0.098  0.507
## Tm:TrtmntPr  0.102 -0.747 -0.120 -0.073
## Tm:TrtmntSy  0.117 -0.852 -0.083 -0.113  0.636
```

We get two `p-values` for the interaction, which mekes it difficult to assess whether the interaction is significant.

```r
drop1(M1, test = "Chi")
```

```
## Single term deletions
##
## Model:
## Dandelion ~ Time * Treatment + (1 | fHive)
##                npar    AIC    LRT   Pr(Chi)
## <none>               1035.6
## Time:Treatment    2 1171.9 140.25 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
M1A <- glmer(Dandelion ~ Time + Treatment + (1|fHive),
            data = KW, family = poisson)

logLik(M1) - logLik(M1A)
```

```
## 'log Lik.' 70.12406 (df=7)
```

The model M1 is the full model, and in MA1, we have dropped the interaction term. The difference between the two likelihood values is 70.12. Twice the difference (140.25 in the drop1 table) follows a Chi-square distribution with 2 degrees of freedom. The 2 is because the interaction contents two parameters. Hence the results of the Poisson GLMM indicate that the interaction between `Time` and `Treatement` is significant. However, the standard errors and the `p-values` are based on the assumption that the Poisson GLMM in the appropriate model. **We need to check overdispersion**

#Check for overdispersion

```r
E1 <- resid(M1, type = "pearson")
N  <- nrow(KW)
p  <- length(fixef(M1)) + 1
Overdispersion <- sum(E1^2) / (N - p)
Overdispersion
```

```
## [1] 10.29562
```

Results indicate overdispersion, we need to determine its source.

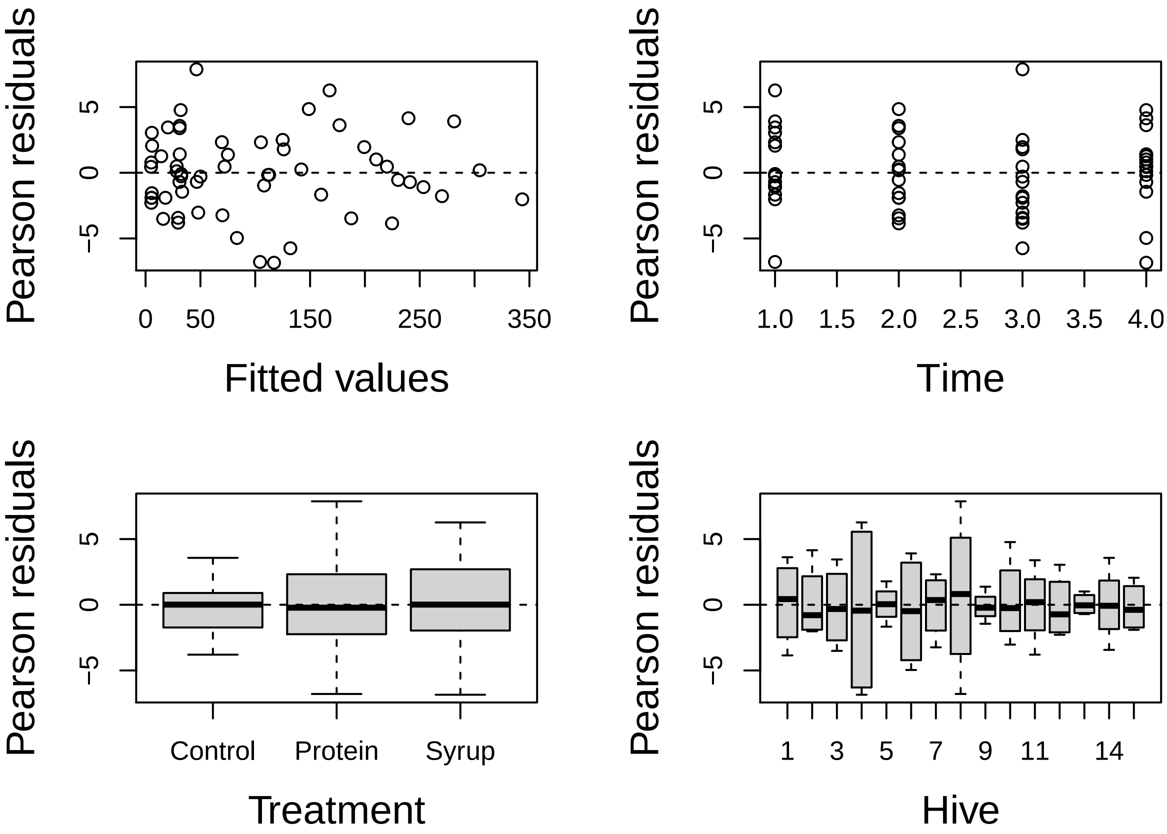
Residuals

```
F1 <- fitted(M1, type ="response")
par(mfrow = c(2,2), mar = c(5,5,2,2))
plot(x = F1,
     y = E1,
     xlab = "Fitted values",
     ylab = "Pearson residuals",
     cex.lab = 1.5)
abline(h = 0, lty = 2)

plot(x = KW$Time, y = E1,
     xlab = "Time",
     ylab = "Pearson residuals",
     cex.lab = 1.5)
abline(h = 0, lty = 2)

boxplot(E1 ~ Treatment, data = KW,
        xlab = "Treatment",
        ylab = "Pearson residuals",
        cex.lab = 1.5)
abline(h = 0, lty = 2)

boxplot(E1 ~ fHive, data = KW,
        xlab = "Hive",
        ylab = "Pearson residuals",
        cex.lab = 1.5)
abline(h = 0, lty = 2)
```



Based on the range of the data, it appears that a negative binomial GLMM is required. ## Negative Binomial

Building the model

$$D_{ij} \sim NB(\mu_{ij}, k)$$
$$E(D_{ij}) = \mu_{ij}$$

$$var(D_{ij}) = \mu_{ij} + \frac{\mu_{ij}^2}{k}$$

$$\log(\mu_{ij}) = Time_{ij} + Treatment_{ij} + Treatment_{ij} \times Time_{ij} + a_i$$
$$a_i \sim N(0, \sigma^2_{Hive})$$

```
# install.packages("R2admb")
# install.packages("glmmADMB",
#     repos=c("http://glmmadmb.r-forge.r-project.org/repos",
#             getOption("repos")),
#     type="source")
```

```
library("glmmADMB")
```

```
## Loading required package: MASS

##
## Attaching package: 'glmmADMB'
## The following object is masked from 'package:MASS':
##
##     stepAIC

## The following object is masked from 'package:stats':
##
##     step
```

```
M2 <- glmmadmb(Dandelion ~ Time * Treatment,
             random =~ 1|fHive,
             family = "nbinom", data=KW)
summary(M2)
```

```
##
## Call:
## glmmadmb(formula = Dandelion ~ Time * Treatment, data = KW, family = "nbinom",
##     random = ~1 | fHive)
##
## AIC: 630.2
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)               3.4943     0.5279    6.62  3.6e-11 ***
## Time                     -0.1441     0.1046   -1.38    0.168
## TreatmentProtein          1.7491     0.7433    2.35    0.019 *
## TreatmentSyrup            1.7074     0.7402    2.31    0.021 *
## Time:TreatmentProtein    -0.2301     0.1464   -1.57    0.116
## Time:TreatmentSyrup      -0.0356     0.1437   -0.25    0.804
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of observations: total=60, fHive=15
## Random effect variance(s):

## Warning in .local(x, sigma, ...): 'sigma' and 'rdig' arguments are present for
## compatibility only: ignored

## Group=fHive
##             Variance StdDev
## (Intercept)   0.9845 0.9922
##
## Negative binomial dispersion parameter: 4.3219 (std. err.: 1.1156)
##
## Log-likelihood: -307.105
```

```
E2 <- resid(M2, type = "pearson")
p <- 6 + 1 + 1 #Number of betas + k + sigma
Overdispersion2 <-sum(E2^2) / (N - p)
Overdispersion2
```

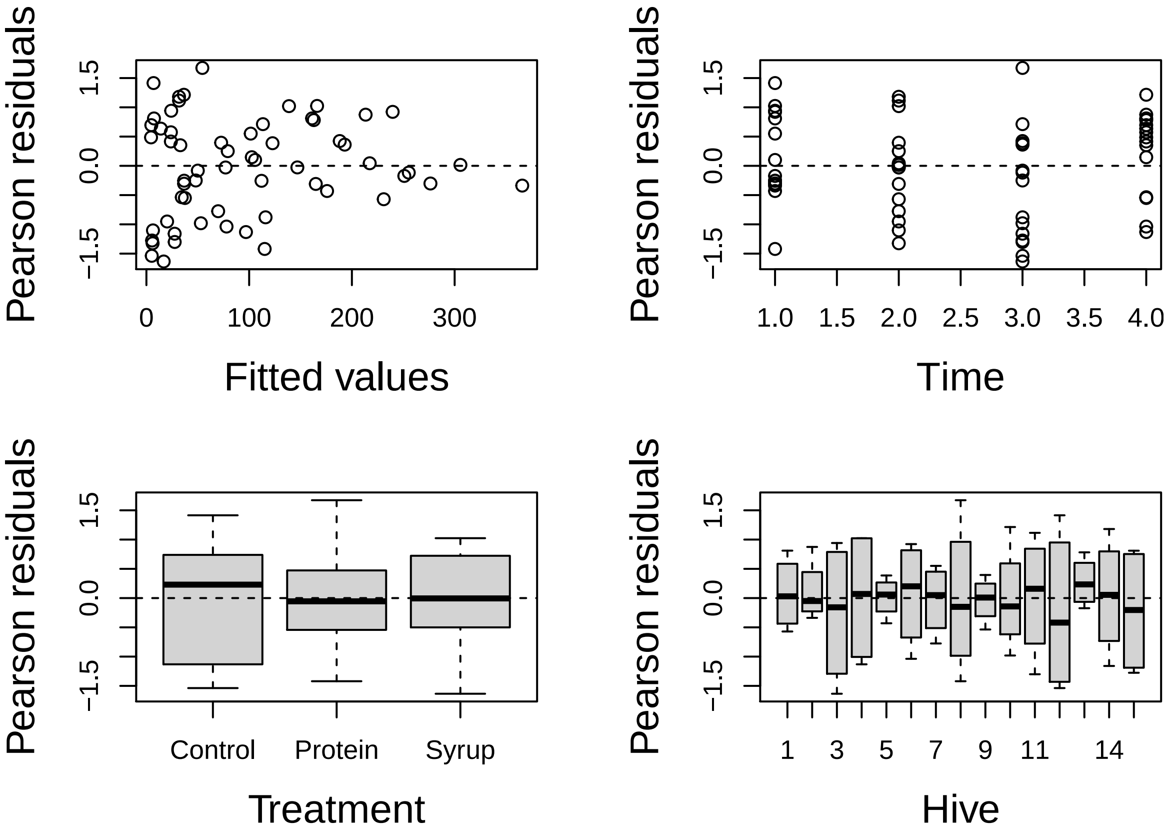
```
## [1] 0.7832768
```

```
F2 <- fitted(M2, type ="response")
par(mfrow = c(2,2), mar = c(5,5,2,2))
plot(x = F2,
     y = E2,
     xlab = "Fitted values",
     ylab = "Pearson residuals",
     cex.lab = 1.5)
abline(h = 0, lty = 2)

plot(x = KW$Time, y = E2,
     xlab = "Time",
     ylab = "Pearson residuals",
     cex.lab = 1.5)
abline(h = 0, lty = 2)

boxplot(E2 ~ Treatment, data = KW,
        xlab = "Treatment",
        ylab = "Pearson residuals",
        cex.lab = 1.5)
abline(h = 0, lty = 2)

boxplot(E2 ~ fHive, data = KW,
        xlab = "Hive",
        ylab = "Pearson residuals",
        cex.lab = 1.5)
abline(h = 0, lty = 2)
```



Binomial GLMM

```
ZooData <- read.csv("~/Google Drive/Course/data/ZooData.csv")
names(ZooData)
```

```
##  [1] "Number"     "Scans"      "Proportion" "Size"       "Visual"
##  [6] "Raised"     "Visitors"   "Feeding"    "Oc"         "Other"
## [11] "Enrichment" "Group"      "Sex"        "Enclosure"  "Vehicle"
## [16] "Diet"       "Age"        "Zoo"        "Eps"
```

```
str(ZooData)
```

```
## 'data.frame':    88 obs. of  19 variables:
##  $ Number    : int  41 47 28 26 13 25 15 78 77 64 ...
##  $ Scans     : int  300 150 300 150 148 152 301 299 300 300 ...
##  $ Proportion: num  0.1367 0.3133 0.0933 0.1733 0.0878 ...
##  $ Size      : num  650 2405 1781 390 390 ...
##  $ Visual    : int  2 2 4 5 2 3 4 4 4 1 ...
##  $ Raised    : int  2 1 2 1 1 1 2 2 2 1 ...
##  $ Visitors  : int  6418 13607 13607 0 0 0 11713 11713 11713 11713 ...
##  $ Feeding   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Oc        : int  1 2 2 2 2 2 2 2 2 2 ...
##  $ Other     : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ Enrichment: int  2 2 2 1 1 1 2 2 2 2 ...
##  $ Group     : int  2 1 2 1 1 1 2 2 2 2 ...
##  $ Sex       : int  1 1 1 2 2 2 2 2 1 1 ...
##  $ Enclosure : int  0 0 0 0 0 0 20 12 16 20 ...
##  $ Vehicle   : int  59 263 263 0 0 0 215 215 215 215 ...
##  $ Diet      : int  3 1 1 1 1 1 3 3 3 3 ...
##  $ Age       : int  2 2 3 3 3 2 8 2 2 8 ...
##  $ Zoo       : int  1 2 2 2 2 2 3 3 3 3 ...
##  $ Eps       : int  1 2 3 4 5 6 7 8 9 10 ...
```

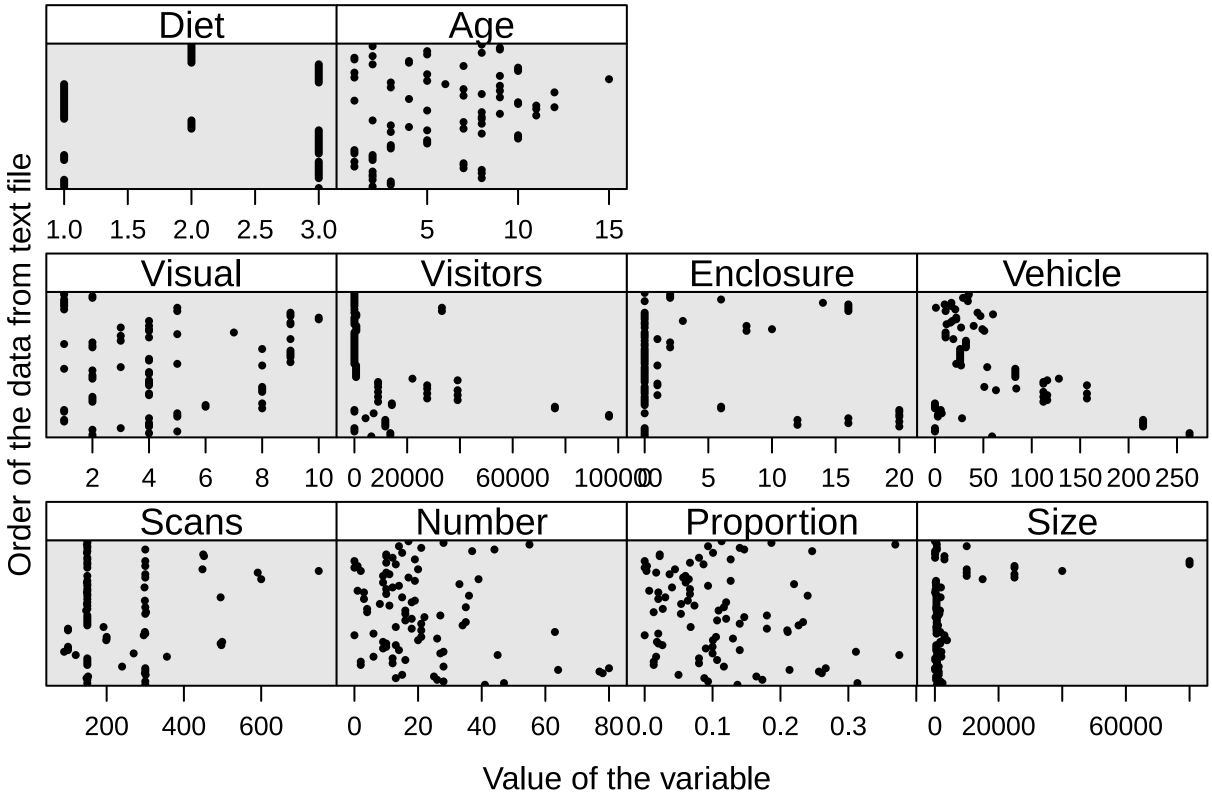
```
#House keeping
ZooData$fRaised     <- factor(ZooData$Raised)
ZooData$fFeeding    <- factor(ZooData$Feeding)
ZooData$fOc         <- factor(ZooData$Oc)
ZooData$fOther      <- factor(ZooData$Other)
ZooData$fEnrichment <- factor(ZooData$Enrichment)
ZooData$fGroup      <- factor(ZooData$Group)
ZooData$fSex        <- factor(ZooData$Sex)
ZooData$fZoo        <- factor(ZooData$Zoo)
```

```
#Data exploration
#Outliers
MyVar <- c("Scans", "Number", "Proportion", "Size", "Visual", "Visitors", "Enclosure", "Vehicle", "Diet
Mydotplot <- function(DataSelected){

P <- dotplot(as.matrix(as.matrix(DataSelected)),
               groups=FALSE,
               strip = strip.custom(bg = 'white',
                                    par.strip.text = list(cex = 1.2)),
               scales = list(x = list(relation = "free", draw = TRUE),
                             y = list(relation = "free", draw = FALSE)),
               col=1, cex  = 0.5, pch = 16,
               xlab = list(label = "Value of the variable", cex = 1),
               ylab = list(label = "Order of the data from text file", cex = 1))

  print(P)
}
```

```
Mydotplot(ZooData[,MyVar])
```



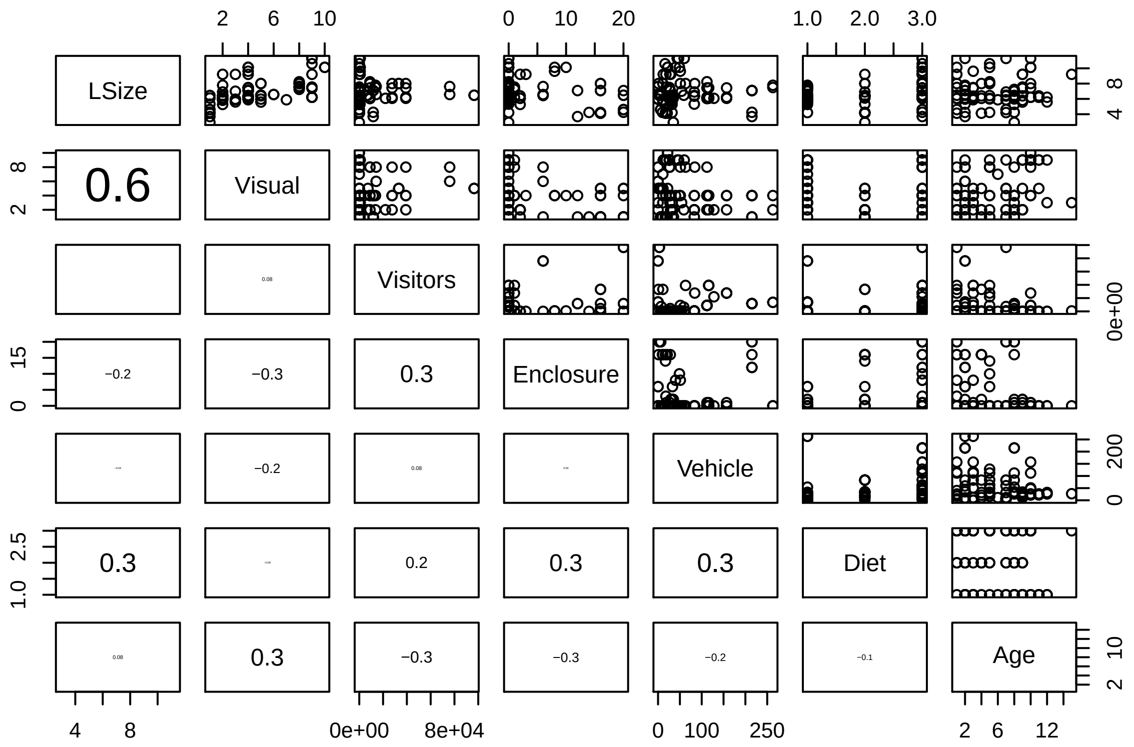
We decide to `log-transform` the covariate Size, as there are several larga values

```
ZooData$LSize <- log(ZooData$Size)
```

Collinearity

We check collinearity. The sample size is relatively low (88 observations), and there are 15 covariates. A statistics rule of thumb when employing regression models is to have approximately 15-25 times as many observations as there are covariates. The easy solution is to drop covariates.

```
MyVar <- c("LSize", "Visual", "Visitors", "Enclosure", "Vehicle", "Diet", "Age")
panel.cor <- function(x, y, digits=1, prefix="", cex.cor = 6)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r1=cor(x,y,use="pairwise.complete.obs")
  r <- abs(cor(x, y,use="pairwise.complete.obs"))
  txt <- format(c(r1, 0.123456789), digits=digits)[1]
  txt <- paste(prefix, txt, sep="")
  if(missing(cex.cor)) { cex <- 0.9/strwidth(txt) } else {
     cex = cex.cor}
  text(0.5, 0.5, txt, cex = cex * r)
}
pairs(ZooData[,MyVar], lower.panel = panel.cor)
```



```
corvif(ZooData[ ,c("LSize",  "Visual", "Visitors",
                   "Age", "Enclosure", "Vehicle",
                   "Diet" , "fRaised",
                   "fFeeding", "fOc",   "fOther",
                   "fEnrichment","fGroup",   "fSex")])
```

```
##
##
## Variance inflation factors
##
##                 GVIF
## LSize       3.390017
## Visual      3.406517
## Visitors    1.934461
## Age         1.606160
## Enclosure   1.865990
## Vehicle     1.934853
## Diet        4.194461
## fRaised     2.294482
## fFeeding    2.220489
## fOc         2.768179
## fOther      1.881560
## fEnrichment 1.818273
## fGroup      1.624768
## fSex        1.315063
```

```
corvif(ZooData[ ,c("LSize",  "Visual", "Visitors",
                   "Age", "Enclosure", "Vehicle",
                   "fRaised",
                   "fFeeding", "fOc",   "fOther",
                   "fEnrichment","fGroup",   "fSex")])
```

```
##
##
## Variance inflation factors
##
##                 GVIF
## LSize       2.446572
## Visual      3.152347
## Visitors    1.795288
## Age         1.509785
## Enclosure   1.538524
## Vehicle     1.864943
## fRaised     2.134036
## fFeeding    2.189233
## fOc         1.453954
## fOther      1.881288
## fEnrichment 1.754485
## fGroup      1.621652
## fSex        1.278277
```

```
corvif(ZooData[ ,c("LSize",  "Visitors",
                   "Age", "Enclosure", "Vehicle",
                   "fRaised",
                   "fFeeding", "fOc",  "fOther",
                   "fEnrichment","fGroup",   "fSex")])
```

```
##
##
## Variance inflation factors
##
##                 GVIF
## LSize       2.040831
## Visitors    1.588905
## Age         1.347137
## Enclosure   1.485076
## Vehicle     1.801960
## fRaised     2.113856
## fFeeding    1.653351
## fOc         1.452516
## fOther      1.877585
## fEnrichment 1.752556
## fGroup      1.446987
## fSex        1.200737
```

Remove diet and visual

```
#Number of observations per zoo
table(ZooData$Zoo)
```

```
##
##  1  2  3  4  5  6  7  8  9
##  1  5  6  5  4 15 28 12 12
```

```
ZD <- ZooData[ZooData$Zoo != 1, ]  #Remove the first zoo
ZD$fZoo <- factor(ZD$Zoo)
dim(ZD)
```

```
## [1] 87 28
```

```
#Standardise all continuous covariates
MyNorm <- function(x) { (x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE)}
ZD$cLSize        <- MyNorm(ZD$LSize)
ZD$cVisitors     <- MyNorm(ZD$Visitors)
ZD$cAge          <- MyNorm(ZD$Age)
ZD$cEnclosure    <- MyNorm(ZD$Enclosure)
ZD$cVehicle      <- MyNorm(ZD$Vehicle)
```

```
ZD$Neg <- ZD$Scans - ZD$Number
M1 <- glmer(cbind(Number, Neg) ~ cLSize + cVisitors+ fFeeding+
            fOc + fOther + fEnrichment + fGroup + fSex +
            cEnclosure + cVehicle+ cAge + (1 | fZoo),
            family = binomial, data = ZD)
summary(M1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(Number, Neg) ~ cLSize + cVisitors + fFeeding + fOc + fOther +
##     fEnrichment + fGroup + fSex + cEnclosure + cVehicle + cAge +
##     (1 | fZoo)
##    Data: ZD
##
##      AIC      BIC   logLik deviance df.resid
##   1111.7   1143.8   -542.8   1085.7       74
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.1159 -1.8641 -0.4953  1.7841 12.0372
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  fZoo   (Intercept) 0.376    0.6132
## Number of obs: 87, groups:  fZoo, 8
##
## Fixed effects:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.43184    0.29901  -8.133 4.19e-16 ***
## cLSize         -0.04292    0.04734  -0.907 0.364591
## cVisitors      -0.24103    0.04894  -4.925 8.44e-07 ***
## fFeeding2      -0.47075    0.09412  -5.002 5.69e-07 ***
## fOc2            1.14416    0.21215   5.393 6.92e-08 ***
## fOther2        -0.03163    0.12007  -0.263 0.792192
## fEnrichment2    0.20279    0.08732   2.322 0.020208 *
## fGroup2        -0.99059    0.07066 -14.019  < 2e-16 ***
## fSex2          -0.16012    0.05601  -2.859 0.004251 **
## cEnclosure     -0.10373    0.05185  -2.001 0.045412 *
## cVehicle        0.12979    0.03938   3.296 0.000981 ***
## cAge           -0.14062    0.03299  -4.262 2.02e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
```

```
##             (Intr) cLSize cVstrs fFdng2 fOc2   fOthr2 fEnrc2 fGrop2 fSex2
## cLSize       0.249
## cVisitors    0.073 -0.160
## fFeeding2   -0.010  0.068  0.175
## fOc2        -0.612 -0.206 -0.120 -0.042
## fOther2     -0.369 -0.424 -0.128 -0.305  0.276
## fEnrichmnt2 -0.108 -0.189  0.045  0.399 -0.018 -0.015
## fGroup2     -0.137 -0.178  0.036  0.023  0.072 -0.105  0.034
## fSex2       -0.110  0.072 -0.080 -0.084  0.044 -0.065 -0.091  0.169
## cEnclosure  -0.189  0.054 -0.335  0.013  0.228  0.180 -0.074 -0.108  0.065
## cVehicle     0.010  0.073 -0.137 -0.039  0.006  0.004 -0.405 -0.032  0.161
## cAge         0.033 -0.070  0.052 -0.162 -0.050 -0.008 -0.049  0.093  0.254
##             cEncls cVehcl
## cLSize
## cVisitors
## fFeeding2
## fOc2
## fOther2
## fEnrichmnt2
## fGroup2
## fSex2
## cEnclosure
## cVehicle     0.166
## cAge         0.112  0.170
```

```
E1 <- residuals(M1)
p1 <- length(fixef(M1)) + 1
Overdisp1 <- sum(E1^2) / (nrow(ZD) - p1)
Overdisp1
```

```
## [1] 9.076217
```

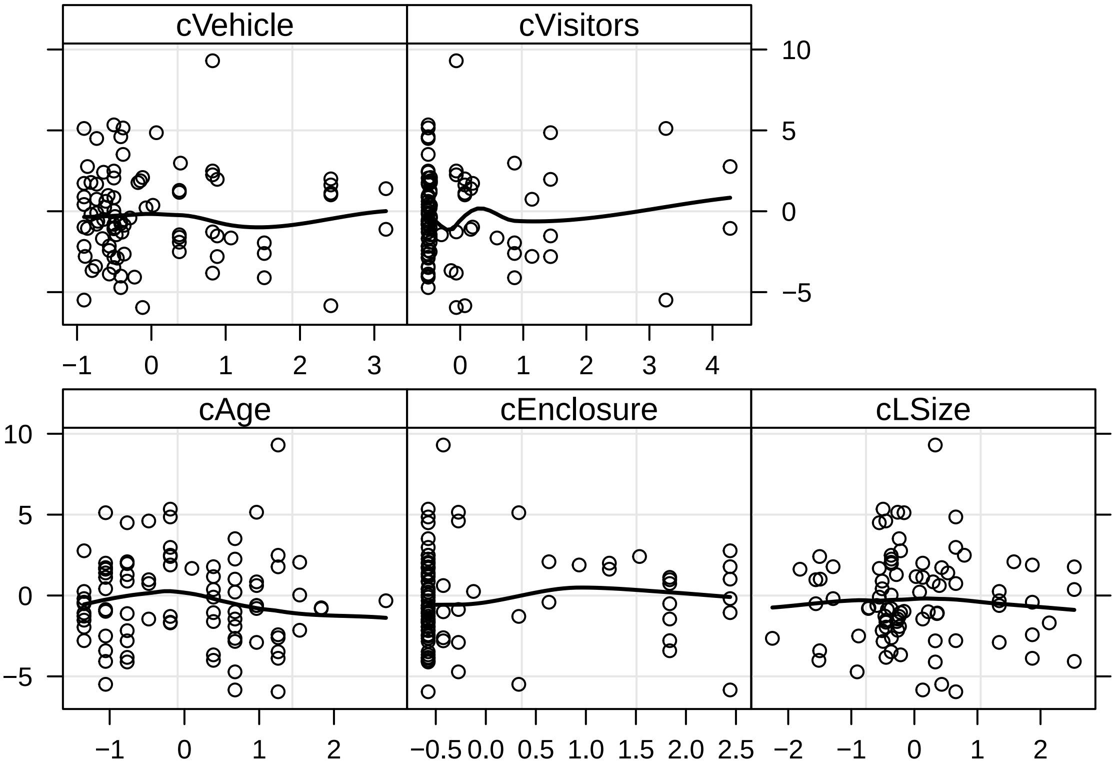
We have overdispersion, we need to check reasons of thid overdispersion

```
ZD$E1 <- E1
vars <- c("cLSize", "cVisitors",  "cEnclosure", "cVehicle", "cAge")
Myxyplot <- function(Z, MyV, NameY1, MyXlab = "", MyYlab="") {
  AllX  <- as.vector(as.matrix(Z[,MyV]))
  AllY  <- rep(Z[,NameY1] , length(MyV))
  AllID <- rep(MyV, each = nrow(Z))


  library(mgcv)
  library(lattice)

  P <- xyplot(AllY ~ AllX|factor(AllID), col = 1,
              xlab = list(MyXlab, cex = 1.5),
              #ylab = list("Response variable", cex = 1.5),
              #ylab = list("Pearson residuals", cex = 1.5),
              ylab = list(MyYlab, cex = 1.5),
              #layout = c(2,2),   #Modify
              strip = function(bg='white', ...)
                strip.default(bg='white', ...),
              scales = list(alternating = TRUE,
                            x = list(relation = "free"),
                            y = list(relation = "same")),
              panel=function(x, y){
                panel.grid(h=-1, v= 2)
                panel.points(x, y, col = 1)
                panel.loess(x, y, span = 0.8,col = 1, lwd = 2)
                })

  print(P)
}
Myxyplot(ZD, vars,"E1")
```



There are not linear patterns so non-linear residual patterns are not the reason for the overdispersion.

Zero-inflation

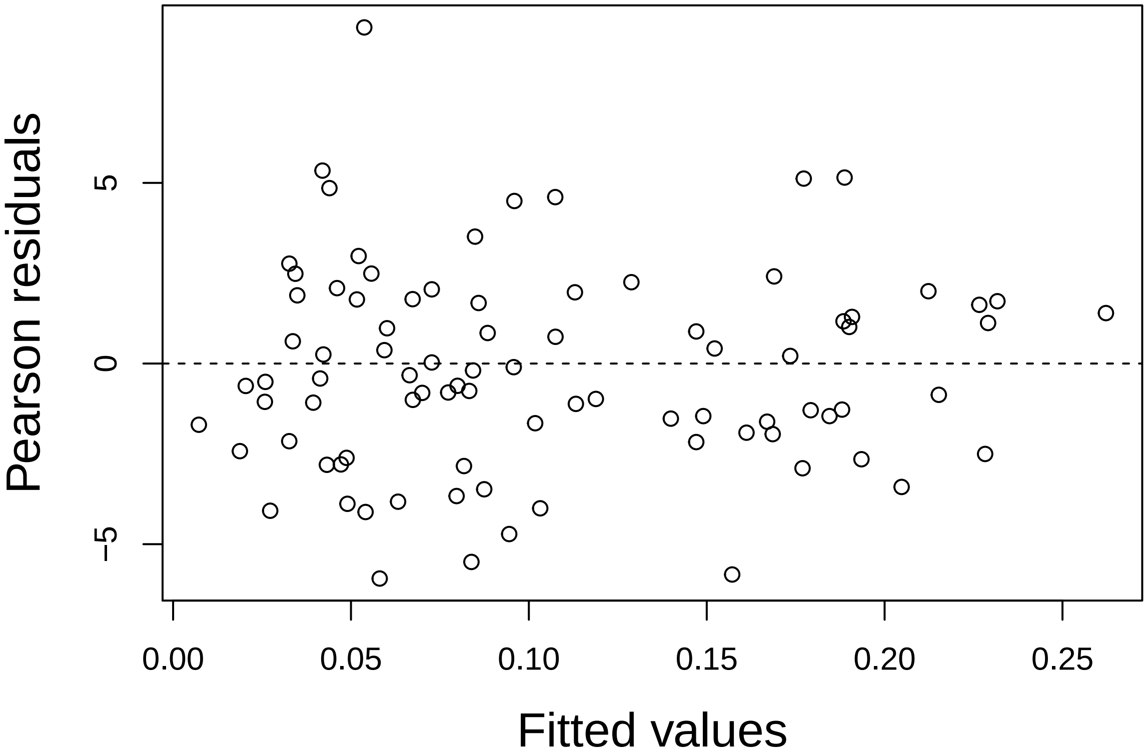
```
table(ZD$Number)
```

```
##
##  0  1  2  3  4  6  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 25 26 27 28 33
##  3  2  3  2  2  2  1  4  8  2  4  4  3  3  4  2  3  3  2  4  1  1  2  2  4  1
## 34 35 36 37 39 44 45 47 55 63 64 77 78 80
##  1  2  1  1  1  1  1  1  1  1  1  1  1  1
```

Zero-inflation is no the reason as 0 value was observed only 3 times.

Pearson residuals

```
F1 <- fitted(M1, type = "response")
par(mar = c(5, 5, 2, 2))
plot(F1,E1,
     xlab = "Fitted values",
     ylab = "Pearson residuals",
     cex.lab = 1.5)
abline(h = 0, lty = 2)
```



Pearson residuals does not indicate problems, although some values are fairly large

Since we cannot pinpoint one of the most common sources of overdispersion (outliers, non-linear patterns, zero inflation), we may consider another distribution or conclude that we have missing covariates