Chapter 14 Count Data | Chapter 15 Count Data in Tables

Qianqian Shan

June 5, 2017

Chapter 14 Count Data

Reason why linear regression not appropriate: 1. linear regression may lead to negative counts

2. variance of the response variable is likely to increase with the mean
3. error may not be normally distributed
4. zeros are difficult to handle in transformations.

Regression with Poisson errors

Introduce zero-inflated distribution for data with a lot of zeros.

```
options(contrasts = c("contr.treatment", "contr.poly"))

clusters<-read.table("clusters.txt", header = TRUE)
attach(clusters)
head(clusters, 4)
```

```
##   Cancers Distance
## 1       0 11.46952
## 2       0 66.55395
## 3       0 47.46230
## 4       0 48.38129
```

```
table(Cancers) # a lot of zero values
```

```
## Cancers
##  0  1  2  3  4  6
## 48 23 12  7  3  1
```

```
# glm with poisson errors
model1 <- glm(Cancers ~ Distance, family = poisson)
summary(model1)
```

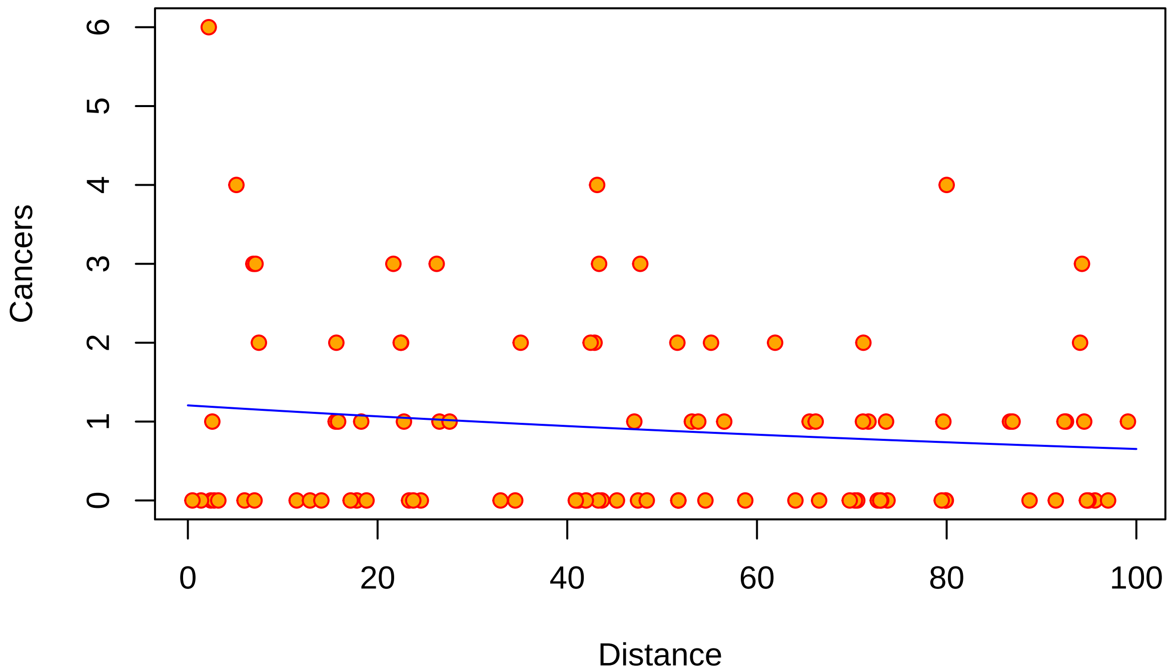
```
##
## Call:
## glm(formula = Cancers ~ Distance, family = poisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5504  -1.3491  -1.1553   0.3877   3.1304
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.186865   0.188728   0.990   0.3221
```

```
## Distance    -0.006138   0.003667  -1.674   0.0941 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 149.48  on 93  degrees of freedom
## Residual deviance: 146.64  on 92  degrees of freedom
## AIC: 262.41
##
## Number of Fisher Scoring iterations: 5
```

```r
# Under poisson errors, the residual deviance is equal to the residual degrees of freedoms,
# there is an obvious sign of overdispersion here

# use quasipoisson instead , check page 563 of "The R book" or Stat520 notes for quasi likeilhood
# quasilikelihood only specifies the mean-variance relationship up to a proportionality constant
model2 <- glm(Cancers ~ Distance, family = quasipoisson)
summary(model2)
```

```
##
## Call:
## glm(formula = Cancers ~ Distance, family = quasipoisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5504  -1.3491  -1.1553   0.3877   3.1304
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.186865   0.235364   0.794    0.429
## Distance    -0.006138   0.004573  -1.342    0.183
##
## (Dispersion parameter for quasipoisson family taken to be 1.555271)
##
##     Null deviance: 149.48  on 93  degrees of freedom
## Residual deviance: 146.64  on 92  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```r
# show the fitted line on plot
xv <- seq(0, 100)
yv <- predict(model2, list(Distance = xv))
plot(Cancers ~ Distance, pch = 21, col = "red", bg = "orange")
lines(xv, exp(yv), col = "blue")
```




```
# no obvious trend
# need to use exp(yv) as y as we used log link

detach(clusters)






# a way to deal with spike at zeros
# this is an example using beta binomial distribution for a certain data set
# Y is a random variable that is almost surely 0 when Z = 0
# and distributed Beta-Binomial(n, alpha, beta) when Z = 1. Z ~ Bernoulli(p).
# More details on HW3 of Stat 601
mydata <- c(rep(0, 400),
            rep(1, 16),
            rep(2, 12),
            rep(3, 12),
            rep(4, 5),
            rep(5, 10),
            rep(6, 3),
            rep(7, 4),
            rep(8, 2)
)

n <- 8

# pmf for the specific data
zibb.pmf <- function(y, par){
    p <- par[1]
    a <- par[2]
    b <- par[3]
```

```r
if (y == 0)
    return((1 - p) + p * beta(a, n + b) / beta(a, b))

return(p * choose(n, y) * beta(y + a, n - y + b) / beta(a, b))
}


# the log likelihood of the above pmf for each y
zibb.loglik <- function(i, data, par) return(log(zibb.pmf(data[i], par)))

# the overall likelihood
full.loglik <- function(par, data) {
L <- length(data)
sum <- sum(sapply(1:L, FUN = zibb.loglik, data = data, par = par))
return(sum)
}

# starting values
startingpar <- c(.5, 1, 1)

# use optim function to find the estimated paramters
results <- optim(par = startingpar, fn = full.loglik, data = mydata,
                 method = "Nelder-Mead", control = list(fnscale = -1))
# By default optim performs minimization, but it will maximize if control$fnscale is negative.
estpars <- results$par
estpars
```

```
## [1] 0.1894463 0.7298926 1.7367949
```

```r
p <- estpars[1]
a <- estpars[2]
b <- estpars[3]

# calculate the predicted frequencies
y <- 0:8

f<- numeric(length(y))
f[1] <- (1- p) + p * beta(a, n + b)/beta(a, b)

for(i in 2:9){
  f[i] <- p * choose(n, i - 1) * beta(i - 1 + a, n - (i - 1) + b)/ beta(a, b)
}

f * sum(table(mydata))
```

```
## [1] 400.005526  15.979068  12.504823  10.134440   8.236388   6.579509
## [7]   5.044414   3.544121   1.971712
```

```r
data <- data.frame(observed = table(mydata), predicted = f * (sum(table(mydata))))
data
```

```
##   observed.mydata observed.Freq  predicted
## 1               0           400 400.005526
## 2               1            16  15.979068
## 3               2            12  12.504823
```

```
## 4                3            12  10.134440
## 5                4             5   8.236388
## 6                5            10   6.579509
## 7                6             3   5.044414
## 8                7             4   3.544121
## 9                8             2   1.971712
```

```r
rm(list = c("y", "n"))
```

Analysis of deviance with count data

```r
# no data file found for this chunk
count <- read.table("cellcounts.txt", header = TRUE)
attach(count)
names(count)

table(cells)

tapply(cells, smoker, mean)

tapply(cells, weight, mean)

tapply(cells, sex, mean)

tapply(cells, age, mean)

model1 <- glm(cells ~ smoker * sex * age * weight, family = poisson)
summary(model1)


model2 <- glm(cells ~ smoker * sex * age * weight, family = quasipoisson)
summary(model2)

model3 <- update(model2, ~. -smoker:sex:age:weight)
summary(model3)


newWt <- weight
levels(newWt)[c(1, 3)] <- "not"
summary(model15)

tapply(cells, list(smoker, weight), mean)

barplot(tapply(cells, list(smoker, weight), mean), col = c("wheat2", "wheat4"),
        beside = TRUE, ylab = "damaged cells", xlab = "body mass")
legend(1.2, 3.4, c("non-smoker", "smoker"), fill = c("wheat2", "wheat4"))

detach(count)
```

Analysis of covariance with count data

```
species1 <- read.table("species.txt", header = TRUE)
attach(species1)
names(species1)
```

```
## [1] "pH"      "Biomass" "Species"
```

```
plot(Biomass, Species, type = "n")

# split divides the data in the vector x into the groups defined by f.
# split(x, f, drop = FALSE, ...)

spp <- split(Species, pH)
spp
```

```
## $high
##  [1] 30 39 44 35 25 29 23 18 19 12 39 35 30 30 33 20 26 36 18  7 39 39 34
## [24] 31 24 25 20 21 12 11
##
## $low
##  [1] 18 19 15 19 12 11 15  9  3  2 18 19 13  9  8 14 13  4  8  2 17 14 15
## [24] 17  9  8 12 14  7  3
##
## $mid
##  [1] 29 30 21 18 13 13  9 24 26 26 20 21 15  8 31 28 18 16 19 20  6 25 23
## [24] 25 22 15 11 17 24 27
```

```
bio <- split(Biomass, pH)
bio
```

```
## $high
##  [1] 0.46929722 1.73087043 2.08977848 3.92578714 4.36679265 5.48197468
##  [7] 6.68468591 7.51165063 8.13220251 9.57212864 0.08665367 1.23697390
## [13] 2.53204324 3.40794153 4.60504596 5.36771709 6.56084215 7.24206214
## [19] 8.50363299 9.39095342 0.76488801 1.17647020 2.32512082 3.22288207
## [25] 4.13612930 5.13717652 6.42193811 7.06552638 8.74592918 9.98177013
##
## $low
##  [1] 0.10084790 0.13859609 0.86351508 1.29291903 2.46916355 2.36655309
##  [7] 2.62921708 3.25228652 4.41727619 4.78081039 0.05017529 0.48283691
## [13] 0.65266714 1.55533656 1.67163820 2.87005390 2.51072052 3.49760385
## [19] 3.67876186 4.83154245 0.28972266 0.07756009 1.42902041 1.12074092
## [25] 1.50795384 2.32596318 2.99570582 3.53819909 4.36454121 4.87050789
##
## $mid
##  [1] 0.1757627 1.3767783 2.5510426 3.0002743 4.9056239 5.3433054 7.7000000
##  [8] 0.5536889 1.9902964 2.9126367 3.2164513 4.9798847 5.6587229 8.1000000
## [15] 0.7395699 1.5269342 2.2321224 3.8852882 4.6265054 5.1209684 8.3000000
## [22] 0.5112786 1.4782327 2.9345580 3.5054889 4.6179091 5.6969638 6.0930169
## [29] 0.7300628 1.1580684
```

```
points(bio[[1]], spp[[1]], pch = 16, col = "red")
points(bio[[2]], spp[[2]], pch = 16, col = "green")
points(bio[[3]], spp[[3]], pch = 16, col = "blue")
legend("topright", legend = c("high", "low", "medium"),
```

```r
      pch = c(16, 16, 16), col = c("red", "green", "blue"),
      title = "pH")


# check the main effects and the interaction effects
model1 <- glm(Species ~ Biomass * pH, family = poisson)
summary(model1)
```

```
##
## Call:
## glm(formula = Species ~ Biomass * pH, family = poisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4978  -0.7485  -0.0402   0.5575   3.2297
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.76812    0.06153  61.240  < 2e-16 ***
## Biomass       -0.10713    0.01249  -8.577  < 2e-16 ***
## pHlow         -0.81557    0.10284  -7.931 2.18e-15 ***
## pHmid         -0.33146    0.09217  -3.596 0.000323 ***
## Biomass:pHlow -0.15503    0.04003  -3.873 0.000108 ***
## Biomass:pHmid -0.03189    0.02308  -1.382 0.166954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 452.346  on 89  degrees of freedom
## Residual deviance:  83.201  on 84  degrees of freedom
## AIC: 514.39
##
## Number of Fisher Scoring iterations: 4
```

```r
# no evidence of overdispersion


# check  if different slopes for different pHs are necessary or not
model2 <- glm(Species ~ Biomass + pH, family = poisson)
summary(model2)
```

```
##
## Call:
## glm(formula = Species ~ Biomass + pH, family = poisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5959  -0.6989  -0.0737   0.6647   3.5604
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.84894    0.05281  72.885  < 2e-16 ***
## Biomass     -0.12756    0.01014 -12.579  < 2e-16 ***
```

```
## pHlow       -1.13639    0.06720 -16.910  < 2e-16 ***
## pHmid       -0.44516    0.05486  -8.114 4.88e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 452.346  on 89  degrees of freedom
## Residual deviance:  99.242  on 86  degrees of freedom
## AIC: 526.43
##
## Number of Fisher Scoring iterations: 4
```

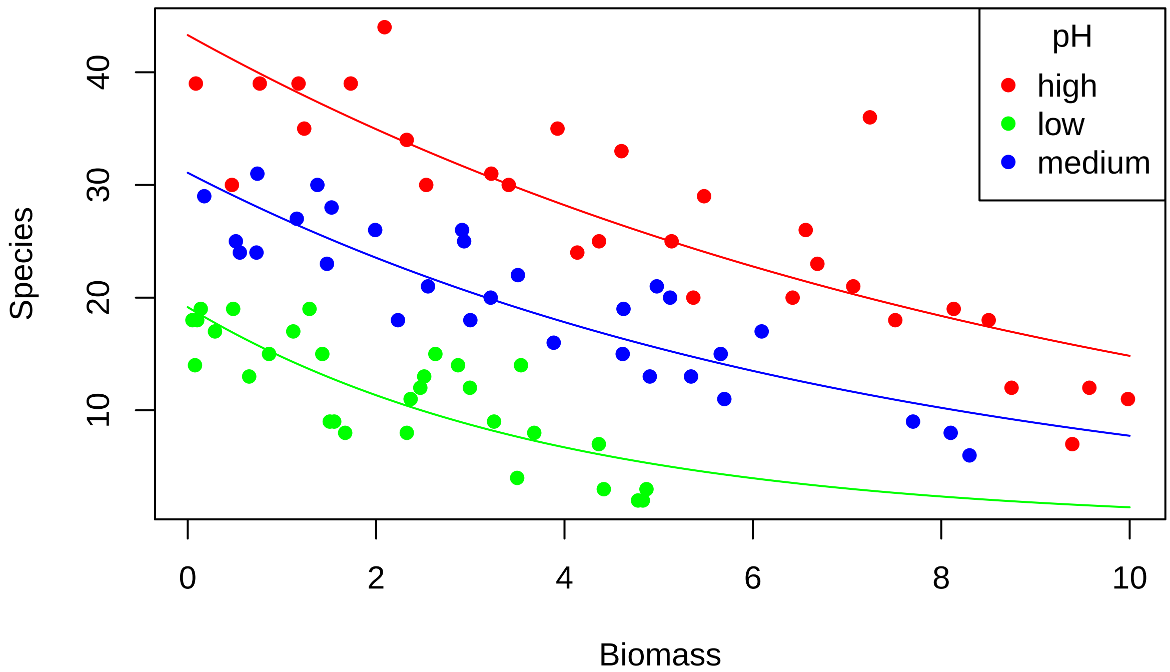
```r
anova(model1, model2, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Species ~ Biomass * pH
## Model 2: Species ~ Biomass + pH
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        84     83.201
## 2        86     99.242 -2   -16.04 0.0003288 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# yes, slopes are significantly different


# draw fitted lines
levels(pH)
```

```
## [1] "high" "low"  "mid"
```

```r
pHs <- factor(rep("high", 101))
xv <- seq(0, 10, 0.1)
yv <- predict(model1, list(Biomass = xv, pH = pHs))
# draw line for high pH level
lines(xv, exp(yv), col = "red")

# low
pHs <- factor(rep("low", 101))
yv <- predict(model1, list(Biomass = xv, pH = pHs))
lines(xv, exp(yv), col = "green")

# mid
pHs <- factor(rep("mid", 101))
yv <- predict(model1, list(Biomass = xv, pH = pHs))
lines(xv, exp(yv), col = "blue")
```




```
detach(species1)
```

Frequency distribution

Negative binoamial distribution is used, one parameter is the mean number of cases, the other parameter is the clumping parameter k(the degree of aggregation in the data, small k values show high aggregation).

With an approximate estimate of the magnitude of k : $\hat{k} = \frac{\bar{x}^2}{s^2 - \bar{x}}$.

```
case.book <- read.table("cases.txt", header = TRUE)
attach(case.book)
names(case.book)
```

```
## [1] "cases"
```

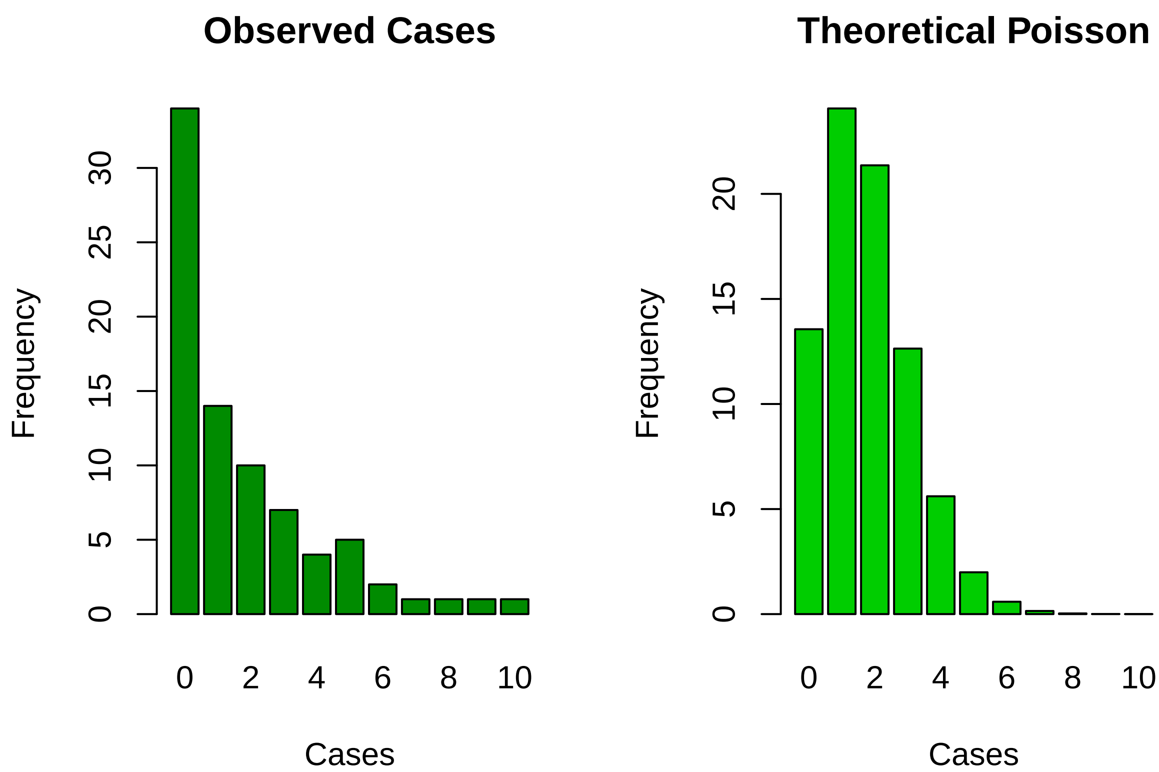
```
frequencies <- table(cases)
frequencies # a lot of zeros
```

```
## cases
##  0  1  2  3  4  5  6  7  8  9 10
## 34 14 10  7  4  5  2  1  1  1  1
```

```
mean(cases)
```

```
## [1] 1.775
```

```
par(mfrow = c(1, 2))

barplot(frequencies, ylab = "Frequency", xlab = "Cases", col = "green4", main = "Observed Cases")
barplot(dpois(0:10, 1.775) * 80, names = as.character(0:10),
        ylab = "Frequency", xlab = "Cases", col = "green3", main = "Theoretical Poisson")
```



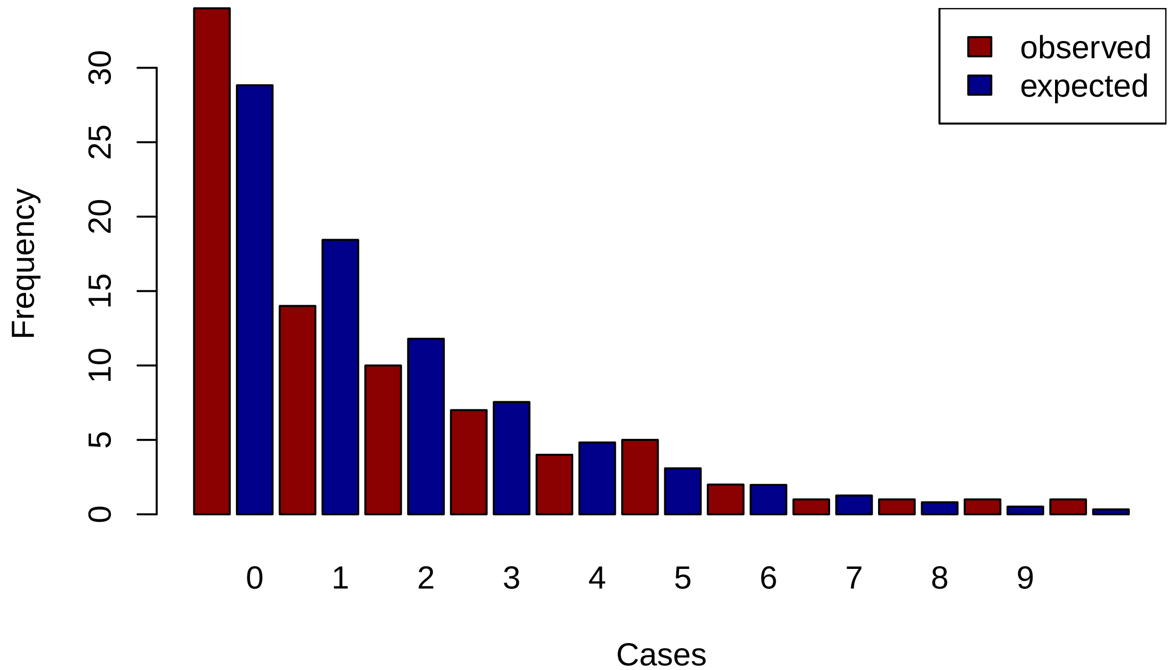

```r
par(mfrow = c(1, 1))
# modes are different , i.e., the observed data are highly aggregated

var(cases)/mean(cases)
```

```
## [1] 2.99483
```

```r
# k value for negative binomial distribution
mean(cases)^2/(var(cases) - mean(cases))
```

```
## [1] 0.8898003
```

```r
expected <- dnbinom(0:10, 1, mu = 1.775) * 80
# 1 is the number of success


# plot observed and expected
both <- numeric(22)
both[1:22 %% 2 != 0] <- frequencies
both[1:22 %% 2 == 0] <- expected

labels <- character(22)
labels[1:22 %% 2 == 0] <- as.character(0:10)

barplot(both, col = rep(c("red4", "blue4"), 11), names = labels, ylab = "Frequency", xlab ="Cases")

legend("topright", legend = c("observed", "expected"), fill = c("red4", "blue4"))
```




```
expected # accumulate the last six frequencies for all values bigger than 4
```

```
##  [1] 28.8288288 18.4400617 11.7949944  7.5445460  4.8257907  3.0867670
##  [7]  1.9744185  1.2629164  0.8078114  0.5167082  0.3305070
```

```
# then do Pearson's chi-square test for lack of fit


# accumulate the last six frequencies
cs <- factor(0:10)
levels(cs)[6:11] <- "5+"
levels(cs)
```

```
## [1] "0"  "1"  "2"  "3"  "4"  "5+"
```

```
(ef <- as.vector(tapply(expected, cs, sum)))
```

```
## [1] 28.828829 18.440062 11.794994  7.544546  4.825791  7.979128
```

```
(of <- as.vector(tapply(frequencies, cs, sum)))
```

```
## [1] 34 14 10  7  4 11
```

```
chi.statistic <- sum((of - ef)^2/ef)

# df is the number of legitimate comparisons(6) minus the number of parameters
# estimated from the data(2) , minus 1
1 - pchisq(chi.statistic, 3)
```

```
## [1] 0.3087556
```

```
detach(case.book)
```

Overdispersion in log-linear models

```r
library(MASS)
data(quine)
attach(quine)
names(quine)
```

```
## [1] "Eth"  "Sex"  "Age"  "Lrn"  "Days"
```

```r
str(quine) # all factors except for response variable
```

```
## 'data.frame':    146 obs. of  5 variables:
##  $ Eth : Factor w/ 2 levels "A","N": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Sex : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Age : Factor w/ 4 levels "F0","F1","F2",..: 1 1 1 1 1 1 1 1 2 2 ...
##  $ Lrn : Factor w/ 2 levels "AL","SL": 2 2 2 1 1 1 1 1 2 2 ...
##  $ Days: int  2 11 14 5 5 13 20 22 6 6 ...
```

```r
# maximal model
model1 <- glm(Days ~ Eth * Sex * Age * Lrn, family = poisson)
summary(model1) # overdispersion
```

```
##
## Call:
## glm(formula = Days ~ Eth * Sex * Age * Lrn, family = poisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.3872  -2.5129  -0.4205   1.7424   6.6783
##
## Coefficients: (4 not defined because of singularities)
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)             3.0564     0.1085  28.178  < 2e-16 ***
## EthN                   -0.1386     0.1590  -0.872 0.383394
## SexM                   -0.4914     0.1648  -2.982 0.002860 **
## AgeF1                  -0.6227     0.1712  -3.638 0.000275 ***
## AgeF2                  -2.3632     0.7154  -3.303 0.000955 ***
## AgeF3                  -0.3784     0.1393  -2.717 0.006592 **
## LrnSL                  -1.9577     0.5875  -3.333 0.000860 ***
## EthN:SexM              -0.7524     0.2682  -2.806 0.005021 **
## EthN:AgeF1              0.1029     0.2408   0.427 0.669209
## EthN:AgeF2             -0.5546     1.2350  -0.449 0.653410
## EthN:AgeF3              0.0633     0.2008   0.315 0.752564
## SexM:AgeF1              0.4092     0.3038   1.347 0.178074
## SexM:AgeF2              3.1098     0.7296   4.262 2.02e-05 ***
## SexM:AgeF3              1.1145     0.2001   5.570 2.55e-08 ***
## EthN:LrnSL              2.2588     0.6314   3.578 0.000347 ***
## SexM:LrnSL              1.5900     0.6305   2.522 0.011673 *
## AgeF1:LrnSL             2.6421     0.6059   4.361 1.30e-05 ***
## AgeF2:LrnSL             4.8585     0.9212   5.274 1.33e-07 ***
## AgeF3:LrnSL                 NA         NA      NA       NA
## EthN:SexM:AgeF1        -0.3105     0.5432  -0.572 0.567587
## EthN:SexM:AgeF2         0.3469     1.2620   0.275 0.783401
## EthN:SexM:AgeF3         0.8329     0.3122   2.668 0.007627 **
## EthN:SexM:LrnSL        -0.1639     0.7024  -0.233 0.815496
```

```
## EthN:AgeF1:LrnSL       -3.5493     0.6715  -5.286 1.25e-07 ***
## EthN:AgeF2:LrnSL       -3.3315     1.3856  -2.404 0.016202 *
## EthN:AgeF3:LrnSL            NA         NA      NA       NA
## SexM:AgeF1:LrnSL       -2.4285     0.7100  -3.420 0.000626 ***
## SexM:AgeF2:LrnSL       -4.1914     0.9555  -4.387 1.15e-05 ***
## SexM:AgeF3:LrnSL            NA         NA      NA       NA
## EthN:SexM:AgeF1:LrnSL   2.1711     0.8924   2.433 0.014985 *
## EthN:SexM:AgeF2:LrnSL   2.1029     1.4330   1.467 0.142254
## EthN:SexM:AgeF3:LrnSL       NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2073.5  on 145  degrees of freedom
## Residual deviance: 1173.9  on 118  degrees of freedom
## AIC: 1818.4
##
## Number of Fisher Scoring iterations: 5
```

```r
# fit quasi poisson model
model2 <- glm(Days ~ Eth * Sex * Age * Lrn, family = quasipoisson)
summary(model2)
```

```
##
## Call:
## glm(formula = Days ~ Eth * Sex * Age * Lrn, family = quasipoisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.3872  -2.5129  -0.4205   1.7424   6.6783
##
## Coefficients: (4 not defined because of singularities)
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.0564     0.3346   9.135 2.22e-15 ***
## EthN                   -0.1386     0.4904  -0.283   0.7780
## SexM                   -0.4914     0.5082  -0.967   0.3356
## AgeF1                  -0.6227     0.5281  -1.179   0.2407
## AgeF2                  -2.3632     2.2066  -1.071   0.2864
## AgeF3                  -0.3784     0.4296  -0.881   0.3802
## LrnSL                  -1.9577     1.8120  -1.080   0.2822
## EthN:SexM              -0.7524     0.8272  -0.910   0.3649
## EthN:AgeF1              0.1029     0.7427   0.139   0.8901
## EthN:AgeF2             -0.5546     3.8094  -0.146   0.8845
## EthN:AgeF3              0.0633     0.6194   0.102   0.9188
## SexM:AgeF1              0.4092     0.9372   0.437   0.6632
## SexM:AgeF2              3.1098     2.2506   1.382   0.1696
## SexM:AgeF3              1.1145     0.6173   1.806   0.0735 .
## EthN:LrnSL              2.2588     1.9474   1.160   0.2484
## SexM:LrnSL              1.5900     1.9448   0.818   0.4152
## AgeF1:LrnSL             2.6421     1.8688   1.414   0.1601
## AgeF2:LrnSL             4.8585     2.8413   1.710   0.0899 .
## AgeF3:LrnSL                 NA         NA      NA       NA
## EthN:SexM:AgeF1        -0.3105     1.6756  -0.185   0.8533
## EthN:SexM:AgeF2         0.3469     3.8928   0.089   0.9291
```

```
## EthN:SexM:AgeF3         0.8329     0.9629   0.865   0.3888
## EthN:SexM:LrnSL        -0.1639     2.1666  -0.076   0.9398
## EthN:AgeF1:LrnSL       -3.5493     2.0712  -1.714   0.0892 .
## EthN:AgeF2:LrnSL       -3.3315     4.2739  -0.779   0.4373
## EthN:AgeF3:LrnSL            NA         NA      NA       NA
## SexM:AgeF1:LrnSL       -2.4285     2.1901  -1.109   0.2697
## SexM:AgeF2:LrnSL       -4.1914     2.9472  -1.422   0.1576
## SexM:AgeF3:LrnSL            NA         NA      NA       NA
## EthN:SexM:AgeF1:LrnSL   2.1711     2.7527   0.789   0.4319
## EthN:SexM:AgeF2:LrnSL   2.1029     4.4203   0.476   0.6351
## EthN:SexM:AgeF3:LrnSL       NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 9.514226)
##
##     Null deviance: 2073.5  on 145  degrees of freedom
## Residual deviance: 1173.9  on 118  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```r
# ftable(table(Eth, Sex, Age, Lrn))

# AIC is not defined for this model and thus step function for mdoel selection is not available
# remove the Age by Lrn interaction from model 2
model4 <- update(model2, ~.-Age:Lrn)
summary(model4)
```

```
##
## Call:
## glm(formula = Days ~ Eth + Sex + Age + Lrn + Eth:Sex + Eth:Age +
##     Sex:Age + Eth:Lrn + Sex:Lrn + Eth:Sex:Age + Eth:Sex:Lrn +
##     Eth:Age:Lrn + Sex:Age:Lrn + Eth:Sex:Age:Lrn, family = quasipoisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.3872  -2.5129  -0.4205   1.7424   6.6783
##
## Coefficients: (4 not defined because of singularities)
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.0564     0.3346   9.135 2.22e-15 ***
## EthN                   -0.1386     0.4904  -0.283   0.7780
## SexM                   -0.4914     0.5082  -0.967   0.3356
## AgeF1                  -0.6227     0.5281  -1.179   0.2407
## AgeF2                  -2.3632     2.2066  -1.071   0.2864
## AgeF3                  -0.3784     0.4296  -0.881   0.3802
## LrnSL                  -1.9577     1.8120  -1.080   0.2822
## EthN:SexM              -0.7524     0.8272  -0.910   0.3649
## EthN:AgeF1              0.1029     0.7427   0.139   0.8901
## EthN:AgeF2             -0.5546     3.8094  -0.146   0.8845
## EthN:AgeF3              0.0633     0.6194   0.102   0.9188
## SexM:AgeF1              0.4092     0.9372   0.437   0.6632
## SexM:AgeF2              3.1098     2.2506   1.382   0.1696
## SexM:AgeF3              1.1145     0.6173   1.806   0.0735 .
```

```
## EthN:LrnSL             2.2588     1.9474   1.160   0.2484
## SexM:LrnSL             1.5900     1.9448   0.818   0.4152
## EthN:SexM:AgeF1       -0.3105     1.6756  -0.185   0.8533
## EthN:SexM:AgeF2        0.3469     3.8928   0.089   0.9291
## EthN:SexM:AgeF3        0.8329     0.9629   0.865   0.3888
## EthN:SexM:LrnSL       -0.1639     2.1666  -0.076   0.9398
## EthA:AgeF1:LrnSL       2.6421     1.8688   1.414   0.1601
## EthN:AgeF1:LrnSL      -0.9072     0.8930  -1.016   0.3117
## EthA:AgeF2:LrnSL       4.8585     2.8413   1.710   0.0899 .
## EthN:AgeF2:LrnSL       1.5270     3.1927   0.478   0.6333
## EthA:AgeF3:LrnSL           NA         NA      NA       NA
## EthN:AgeF3:LrnSL           NA         NA      NA       NA
## SexM:AgeF1:LrnSL      -2.4285     2.1901  -1.109   0.2697
## SexM:AgeF2:LrnSL      -4.1914     2.9472  -1.422   0.1576
## SexM:AgeF3:LrnSL           NA         NA      NA       NA
## EthN:SexM:AgeF1:LrnSL   2.1711     2.7527   0.789   0.4319
## EthN:SexM:AgeF2:LrnSL   2.1029     4.4203   0.476   0.6351
## EthN:SexM:AgeF3:LrnSL       NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 9.514226)
##
##     Null deviance: 2073.5  on 145  degrees of freedom
## Residual deviance: 1173.9  on 118  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```
anova(model2, model4, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: Days ~ Eth * Sex * Age * Lrn
## Model 2: Days ~ Eth + Sex + Age + Lrn + Eth:Sex + Eth:Age + Sex:Age +
##     Eth:Lrn + Sex:Lrn + Eth:Sex:Age + Eth:Sex:Lrn + Eth:Age:Lrn +
##     Sex:Age:Lrn + Eth:Sex:Age:Lrn
##   Resid. Df Resid. Dev Df Deviance F Pr(>F)
## 1       118     1173.9
## 2       118     1173.9  0        0
```

```
anova(model2, model4)
```

```
## Analysis of Deviance Table
##
## Model 1: Days ~ Eth * Sex * Age * Lrn
## Model 2: Days ~ Eth + Sex + Age + Lrn + Eth:Sex + Eth:Age + Sex:Age +
##     Eth:Lrn + Sex:Lrn + Eth:Sex:Age + Eth:Sex:Lrn + Eth:Age:Lrn +
##     Sex:Age:Lrn + Eth:Sex:Age:Lrn
##   Resid. Df Resid. Dev Df Deviance
## 1       118     1173.9
## 2       118     1173.9  0        0
```

```
ftable(tapply(Days, list(Eth, Sex, Lrn), mean))
```

```
##            AL       SL
```

```
##
## A F  14.47368 27.36842
##   M  22.28571 20.20000
## N F  13.14286  7.00000
##   M  13.36364 17.00000
```

Negative binomial errors

Use `glm.nb` function and `MASS` package.

```r
#
model.nb1 <- glm.nb(Days ~ Eth * Sex * Age * Lrn)
summary(model.nb1, cor = FALSE)
```

```
##
## Call:
## glm.nb(formula = Days ~ Eth * Sex * Age * Lrn, init.theta = 1.928360145,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2377  -0.9079  -0.2019   0.5173   1.7043
##
## Coefficients: (4 not defined because of singularities)
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)             3.0564     0.3760   8.128 4.38e-16 ***
## EthN                   -0.1386     0.5334  -0.260 0.795023
## SexM                   -0.4914     0.5104  -0.963 0.335653
## AgeF1                  -0.6227     0.5125  -1.215 0.224334
## AgeF2                  -2.3632     1.0770  -2.194 0.028221 *
## AgeF3                  -0.3784     0.4546  -0.832 0.405215
## LrnSL                  -1.9577     0.9967  -1.964 0.049493 *
## EthN:SexM              -0.7524     0.7220  -1.042 0.297400
## EthN:AgeF1              0.1029     0.7123   0.144 0.885175
## EthN:AgeF2             -0.5546     1.6798  -0.330 0.741297
## EthN:AgeF3              0.0633     0.6396   0.099 0.921159
## SexM:AgeF1              0.4092     0.8299   0.493 0.621973
## SexM:AgeF2              3.1098     1.1655   2.668 0.007624 **
## SexM:AgeF3              1.1145     0.6365   1.751 0.079926 .
## EthN:LrnSL              2.2588     1.3019   1.735 0.082743 .
## SexM:LrnSL              1.5900     1.1499   1.383 0.166750
## AgeF1:LrnSL             2.6421     1.0821   2.442 0.014618 *
## AgeF2:LrnSL             4.8585     1.4423   3.369 0.000755 ***
## AgeF3:LrnSL                 NA         NA      NA       NA
## EthN:SexM:AgeF1        -0.3105     1.2055  -0.258 0.796735
## EthN:SexM:AgeF2         0.3469     1.7965   0.193 0.846875
## EthN:SexM:AgeF3         0.8329     0.8970   0.929 0.353092
## EthN:SexM:LrnSL        -0.1639     1.5250  -0.107 0.914411
## EthN:AgeF1:LrnSL       -3.5493     1.4270  -2.487 0.012876 *
## EthN:AgeF2:LrnSL       -3.3315     2.0919  -1.593 0.111256
## EthN:AgeF3:LrnSL            NA         NA      NA       NA
## SexM:AgeF1:LrnSL       -2.4285     1.4201  -1.710 0.087246 .
## SexM:AgeF2:LrnSL       -4.1914     1.6201  -2.587 0.009679 **
## SexM:AgeF3:LrnSL            NA         NA      NA       NA
```

```
## EthN:SexM:AgeF1:LrnSL   2.1711     1.9192   1.131 0.257963
## EthN:SexM:AgeF2:LrnSL   2.1029     2.3444   0.897 0.369718
## EthN:SexM:AgeF3:LrnSL       NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.9284) family taken to be 1)
##
##     Null deviance: 272.29  on 145  degrees of freedom
## Residual deviance: 167.45  on 118  degrees of freedom
## AIC: 1097.3
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.928
##           Std. Err.:  0.269
##
##  2 x log-likelihood:  -1039.324
```

```
# theta in the model summary is the k parameter

model.nb2 <- stepAIC(model.nb1)
```

```
## Start:  AIC=1095.32
## Days ~ Eth * Sex * Age * Lrn
##
##                   Df    AIC
## - Eth:Sex:Age:Lrn  2 1092.7
## <none>               1095.3
##
## Step:  AIC=1092.73
## Days ~ Eth + Sex + Age + Lrn + Eth:Sex + Eth:Age + Sex:Age +
##     Eth:Lrn + Sex:Lrn + Age:Lrn + Eth:Sex:Age + Eth:Sex:Lrn +
##     Eth:Age:Lrn + Sex:Age:Lrn
##
##               Df    AIC
## - Eth:Sex:Age  3 1089.4
## <none>           1092.7
## - Eth:Sex:Lrn  1 1093.3
## - Eth:Age:Lrn  2 1094.7
## - Sex:Age:Lrn  2 1095.0
##
## Step:  AIC=1089.41
## Days ~ Eth + Sex + Age + Lrn + Eth:Sex + Eth:Age + Sex:Age +
##     Eth:Lrn + Sex:Lrn + Age:Lrn + Eth:Sex:Lrn + Eth:Age:Lrn +
##     Sex:Age:Lrn
##
##               Df    AIC
## <none>           1089.4
## - Sex:Age:Lrn  2 1091.1
## - Eth:Age:Lrn  2 1091.2
## - Eth:Sex:Lrn  1 1092.5
```

```r
summary(model.nb2, cor = F)
```

```
##
## Call:
## glm.nb(formula = Days ~ Eth + Sex + Age + Lrn + Eth:Sex + Eth:Age +
##     Sex:Age + Eth:Lrn + Sex:Lrn + Age:Lrn + Eth:Sex:Lrn + Eth:Age:Lrn +
##     Sex:Age:Lrn, init.theta = 1.865343469, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1387  -0.9777  -0.2000   0.5349   1.7630
##
## Coefficients: (3 not defined because of singularities)
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)        3.1693     0.3411   9.292  < 2e-16 ***
## EthN              -0.3560     0.4210  -0.845 0.397848
## SexM              -0.6920     0.4138  -1.672 0.094459 .
## AgeF1             -0.6405     0.4638  -1.381 0.167329
## AgeF2             -2.4576     0.8675  -2.833 0.004612 **
## AgeF3             -0.5880     0.3973  -1.480 0.138885
## LrnSL             -1.0264     0.7378  -1.391 0.164179
## EthN:SexM         -0.3562     0.3854  -0.924 0.355364
## EthN:AgeF1         0.1500     0.5644   0.266 0.790400
## EthN:AgeF2        -0.3833     0.5640  -0.680 0.496746
## EthN:AgeF3         0.4719     0.4542   1.039 0.298824
## SexM:AgeF1         0.2985     0.6047   0.494 0.621597
## SexM:AgeF2         3.2904     0.8941   3.680 0.000233 ***
## SexM:AgeF3         1.5412     0.4548   3.389 0.000702 ***
## EthN:LrnSL         0.9651     0.7753   1.245 0.213255
## SexM:LrnSL         0.5457     0.8013   0.681 0.495873
## AgeF1:LrnSL        1.6231     0.8222   1.974 0.048373 *
## AgeF2:LrnSL        3.8321     1.1054   3.467 0.000527 ***
## AgeF3:LrnSL            NA         NA      NA       NA
## EthN:SexM:LrnSL    1.3578     0.5914   2.296 0.021684 *
## EthN:AgeF1:LrnSL  -2.1013     0.8728  -2.408 0.016058 *
## EthN:AgeF2:LrnSL  -1.8260     0.8774  -2.081 0.037426 *
## EthN:AgeF3:LrnSL       NA         NA      NA       NA
## SexM:AgeF1:LrnSL  -1.1086     0.9409  -1.178 0.238671
## SexM:AgeF2:LrnSL  -2.8800     1.1550  -2.493 0.012651 *
## SexM:AgeF3:LrnSL       NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.8653) family taken to be 1)
##
##     Null deviance: 265.27  on 145  degrees of freedom
## Residual deviance: 167.44  on 123  degrees of freedom
## AIC: 1091.4
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.865
##           Std. Err.:  0.258
```

```
##
##  2 x log-likelihood:  -1043.409
```

```r
# further simplify the model from above
model.nb3 <- update(model.nb2, ~. - Sex:Age:Lrn)
anova(model.nb3, model.nb2)
```

```
## Likelihood ratio tests of Negative Binomial Models
##
## Response: Days
##
## 1               Eth + Sex + Age + Lrn + Eth:Sex + Eth:Age + Sex:Age + Eth:Lrn + Sex:Lrn + Age:Lrn + E
## 2 Eth + Sex + Age + Lrn + Eth:Sex + Eth:Age + Sex:Age + Eth:Lrn + Sex:Lrn + Age:Lrn + Eth:Sex:Lrn + E
##      theta Resid. df    2 x log-lik.   Test    df LR stat.    Pr(Chi)
## 1 1.789507       125       -1049.111
## 2 1.865343       123       -1043.409 1 vs 2     2 5.701942 0.05778817
```

```r
#
model.nb4 <- update(model.nb3, ~. - Eth:Age:Lrn)
anova(model.nb3, model.nb4)
```

```
## Likelihood ratio tests of Negative Binomial Models
##
## Response: Days
##
## 1               Eth + Sex + Age + Lrn + Eth:Sex + Eth:Age + Sex:Age + Eth:Lrn + Sex:Lrn + Age:Lrn + E
## 2 Eth + Sex + Age + Lrn + Eth:Sex + Eth:Age + Sex:Age + Eth:Lrn + Sex:Lrn + Age:Lrn + Eth:Sex:Lrn + E
##      theta Resid. df    2 x log-lik.   Test    df LR stat.   Pr(Chi)
## 1 1.724987       127       -1053.431
## 2 1.789507       125       -1049.111 1 vs 2     2 4.320086 0.1153202
```

```r
#
model.nb5 <- update(model.nb4, ~. - Age:Lrn)
anova(model.nb4, model.nb5)
```

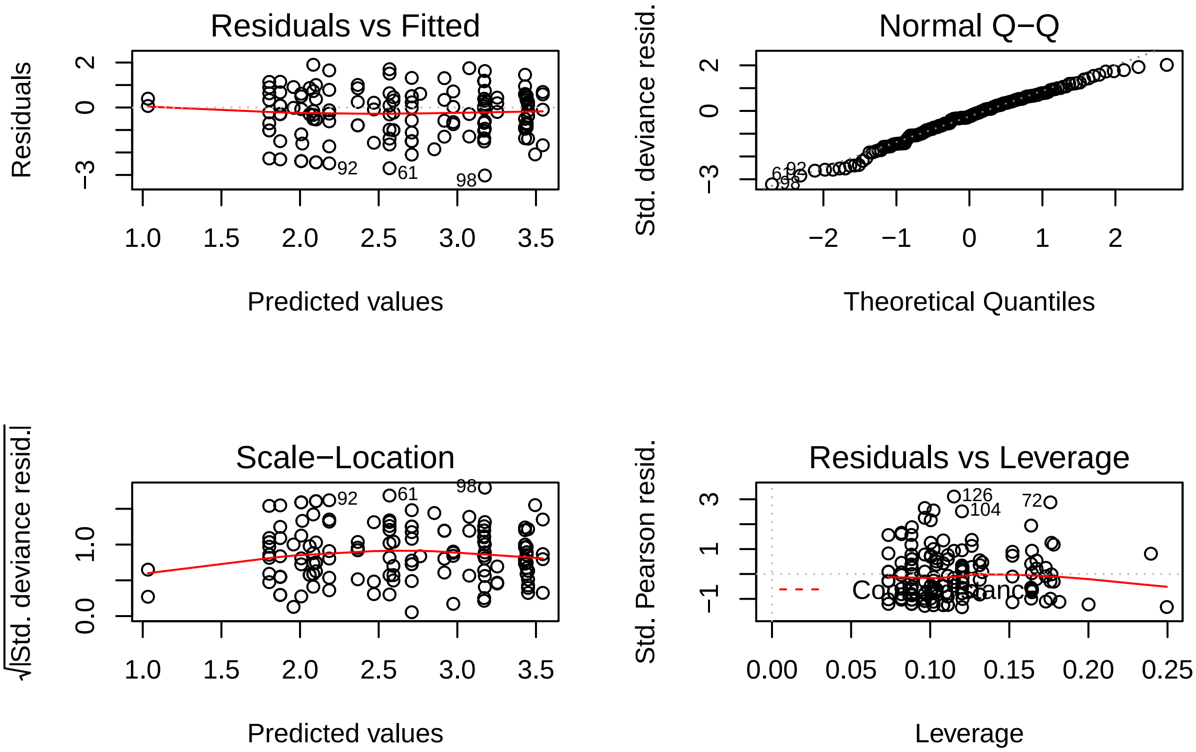
```
## Likelihood ratio tests of Negative Binomial Models
##
## Response: Days
##                                                                                               Model
## 1           Eth + Sex + Age + Lrn + Eth:Sex + Eth:Age + Sex:Age + Eth:Lrn + Sex:Lrn + Eth:Sex:Lrn
## 2 Eth + Sex + Age + Lrn + Eth:Sex + Eth:Age + Sex:Age + Eth:Lrn + Sex:Lrn + Age:Lrn + Eth:Sex:Lrn
##      theta Resid. df    2 x log-lik.   Test    df LR stat.  Pr(Chi)
## 1 1.678620       129       -1057.219
## 2 1.724987       127       -1053.431 1 vs 2     2 3.787823 0.150482
```

```r
summary(model.nb5, cor=F)
```

```
##
## Call:
## glm.nb(formula = Days ~ Eth + Sex + Age + Lrn + Eth:Sex + Eth:Age +
##     Sex:Age + Eth:Lrn + Sex:Lrn + Eth:Sex:Lrn, init.theta = 1.678619829,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0246  -0.9449  -0.2228   0.4847   1.9002
##
```

```
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       2.91755    0.32626   8.942  < 2e-16 ***
## EthN              0.05666    0.39515   0.143  0.88598
## SexM             -0.55047    0.39014  -1.411  0.15825
## AgeF1            -0.32379    0.38373  -0.844  0.39878
## AgeF2            -0.06383    0.42046  -0.152  0.87933
## AgeF3            -0.34854    0.39128  -0.891  0.37305
## LrnSL             0.57697    0.33382   1.728  0.08392 .
## EthN:SexM        -0.41608    0.37491  -1.110  0.26708
## EthN:AgeF1       -0.56613    0.43162  -1.312  0.18965
## EthN:AgeF2       -0.89577    0.42950  -2.086  0.03702 *
## EthN:AgeF3        0.08467    0.44010   0.192  0.84744
## SexM:AgeF1       -0.08459    0.45324  -0.187  0.85195
## SexM:AgeF2        1.13752    0.45192   2.517  0.01183 *
## SexM:AgeF3        1.43124    0.44365   3.226  0.00126 **
## EthN:LrnSL       -0.78724    0.43058  -1.828  0.06750 .
## SexM:LrnSL       -0.47437    0.45908  -1.033  0.30147
## EthN:SexM:LrnSL   1.75289    0.58341   3.005  0.00266 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.6786) family taken to be 1)
##
##     Null deviance: 243.98  on 145  degrees of freedom
## Residual deviance: 168.03  on 129  degrees of freedom
## AIC: 1093.2
##
## Number of Fisher Scoring iterations: 1
##
##
##                 Theta:  1.679
##             Std. Err.:  0.227
##
##  2 x log-likelihood:  -1057.219
```

```r
par(mfrow = c(2, 2))
plot(model.nb5)
```




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

detach(quine)
```

Chapter 15 Count Data in Tables

The general method of analysis for contingency tables involves log-linear modeling, but the simplest contingency tables are often analyzed by Pearson's Chi-square, Fisher's exact test or tests of binomial proportions.

A two-class table of counts

Pearson's chi-square $\chi^2 = \sum \frac{(observed - expected)^2}{expected}$.

```r
# test if the sex ratio is significant from 50:50 or not

observed <- c(29, 18)
chisq.test(observed) # not significant
```

```
##
##  Chi-squared test for given probabilities
##
## data:  observed
## X-squared = 2.5745, df = 1, p-value = 0.1086
```

```r
# performs chi-squared contingency table tests and goodness-of-fit tests.

# or try binomial test alternatively
binom.test(observed)
```

```
##
##  Exact binomial test
##
## data:  observed
## number of successes = 29, number of trials = 47, p-value = 0.1439
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.4637994 0.7549318
## sample estimates:
## probability of success
##              0.6170213
```

Sample size for count data

Test how many samples are needed for detect a significant departure from p = 0.5.

```r
binom.test(1, 8) # n =8 not significant
```

```
##
##  Exact binomial test
##
## data:  1 and 8
## number of successes = 1, number of trials = 8, p-value = 0.07031
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.003159724 0.526509671
## sample estimates:
## probability of success
##                  0.125
```

```r
binom.test(1, 9) # 9 is significant
```

```
##
##  Exact binomial test
##
## data:  1 and 9
## number of successes = 1, number of trials = 9, p-value = 0.03906
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.002809137 0.482496515
## sample estimates:
## probability of success
##              0.1111111
```

A four-class table of counts

```r
# Mendel's famous peas produced 315 yellow round phenotypes
# 101 yellow wrinkled
# 108 green round
# 32 green wrinkled

# test if the data depart significantly from 9:3:3:1
observed <- c(315, 101, 108, 32)
```

```r
(expected <- 556 * c(9, 3, 3, 1)/16)
```

```
## [1] 312.75 104.25 104.25  34.75
```

```r
chisq.test(observed, p = c(9, 3, 3, 1), rescale.p = TRUE)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  observed
## X-squared = 0.47002, df = 3, p-value = 0.9254
```

```r
# rescale is true as the expected values don't sum to 1
# p-value = 0.9254 , not significant

# or calculate it by hand
sum((observed-expected)^2/expected)
```

```
## [1] 0.470024
```

```r
1 - pchisq(0.470024, 3)
```

```
## [1] 0.9254259
```

Two-by-two contingency tables

When there are two explanatory variables and both have just two levels, we have the famous 2 by 2 contingency table.

```r
# convert the vector into a matrix
observed <- matrix(observed, nrow = 2)
observed
```

```
##      [,1] [,2]
## [1,]  315  108
## [2,]  101   32
```

```r
# Fisher's exact test
fisher.test(observed)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  observed
## p-value = 0.819
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.5667874 1.4806148
## sample estimates:
## odds ratio
##  0.9242126
```

```r
# Pearson' chi square test
chisq.test(observed)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
```

```
##
## data:  observed
## X-squared = 0.051332, df = 1, p-value = 0.8208
```

Using log-linear models for simple contingency tables

```r
# 29 males and 18 females
observed <- c(29, 18)

glm(observed ~ 1, family = poisson)
```

```
##
## Call:  glm(formula = observed ~ 1, family = poisson)
##
## Coefficients:
## (Intercept)
##       3.157
##
## Degrees of Freedom: 1 Total (i.e. Null);  1 Residual
## Null Deviance:       2.599
## Residual Deviance: 2.599     AIC: 14.55
```

```r
summary(glm(observed ~ 1, family = poisson))
```

```
##
## Call:
## glm(formula = observed ~ 1, family = poisson)
##
## Deviance Residuals:
##      1       2
##  1.094  -1.184
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.1570     0.1459   21.64   <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: 2.5985  on 1  degrees of freedom
## Residual deviance: 2.5985  on 1  degrees of freedom
## AIC: 14.547
##
## Number of Fisher Scoring iterations: 4
```

```r
# compare the residual deviance with the critical value of a chisq test
1 - pchisq(2.5985, 1)
```

```
## [1] 0.1069649
```

```r
# Mendel's peas : a four level categorical variable
observed <- c(315, 101, 108, 32)

# two explanatory variables
```

```r
shape <- factor(c("round", "round", "wrinkled", "wrinkled"))
colour <- factor(c("yellow", "green", "yellow", "green"))

# maixmal/saturated model
model1 <- glm(observed ~ shape * colour, family = poisson)
# model w/o interaction

model2 <- glm(observed ~ shape + colour, family = poisson)
anova(model1, model2, test = "Chi") # no significant difference
```

```
## Analysis of Deviance Table
##
## Model 1: observed ~ shape * colour
## Model 2: observed ~ shape + colour
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         0    0.00000
## 2         1    0.11715 -1 -0.11715   0.7322
```

```r
summary(model2)
```

```
##
## Call:
## glm(formula = observed ~ shape + colour, family = poisson)
##
## Deviance Residuals:
##        1         2         3         4
## -0.08378   0.14892   0.14396  -0.25928
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    4.60027    0.09013   51.04   <2e-16 ***
## shapewrinkled -1.08904    0.09771  -11.15   <2e-16 ***
## colouryellow   1.15702    0.09941   11.64   <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: 302.38754  on 3  degrees of freedom
## Residual deviance:   0.11715  on 1  degrees of freedom
## AIC: 31.993
##
## Number of Fisher Scoring iterations: 3
```

The danger of contingency tables

Sometimes we may fail to measure a number of factors that have an important influence on the behavior of the system in question.

```r
induced <- read.table("induced.txt", header = TRUE)
attach(induced)
names(induced)
```

```
## [1] "Tree"        "Aphid"       "Caterpillar" "Count"
```

```r
# fit saturated model
model <- glm(Count ~ Tree * Aphid * Caterpillar, family = poisson)

model2 <- update(model, ~ . - Tree:Aphid:Caterpillar)

anova(model, model2, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Count ~ Tree * Aphid * Caterpillar
## Model 2: Count ~ Tree + Aphid + Caterpillar + Tree:Aphid + Tree:Caterpillar +
##     Aphid:Caterpillar
##   Resid. Df Resid. Dev Df    Deviance Pr(>Chi)
## 1         0 0.00000000
## 2         1 0.00079137 -1 -0.00079137   0.9776
```

```r
model3 <- update(model2, ~ . - Aphid:Caterpillar)
anova(model3, model2, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Count ~ Tree + Aphid + Caterpillar + Tree:Aphid + Tree:Caterpillar
## Model 2: Count ~ Tree + Aphid + Caterpillar + Tree:Aphid + Tree:Caterpillar +
##     Aphid:Caterpillar
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         2  0.0040853
## 2         1  0.0007914  1 0.003294   0.9542
```

```r
# fit a model without Tree factor
wrong <- glm(Count ~ Aphid * Caterpillar, family = poisson)
wrong1 <- update (wrong,~. - Aphid:Caterpillar)
anova(wrong, wrong1, test = "Chi") # shows a significant effect of Aphid:Caterpillar,
```

```
## Analysis of Deviance Table
##
## Model 1: Count ~ Aphid * Caterpillar
## Model 2: Count ~ Aphid + Caterpillar
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         4     550.19
## 2         5     556.85 -1  -6.6594 0.009864 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# but not in the previous model

detach(induced)
```

Summary: always fit a saturated model first, containing all the variables of interest and all interactions.

Quasi-Poisson and negative binomial models compared

```r
data <- read.table("bloodcells.txt", header = TRUE)
attach(data)
head(data)
```

```
##   count
## 1     0
## 2     1
## 3     1
## 4     0
## 5     0
## 6     0
```

```r
dim(data)
```

```
## [1] 10000     1
```

```r
gender <- factor(rep(c("female", "male"), c(5000, 5000)))

tapply(count, gender, mean)
```

```
## female   male
## 1.1986 1.2408
```

```r
# fit log-linear model with Poisson errors
model <- glm(count ~ gender, family = poisson)
summary(model) # gender effects not significant
```

```
##
## Call:
## glm(formula = count ~ gender, family = poisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5753  -1.5483  -1.5483   0.6254   7.3023
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.18115    0.01292   14.02   <2e-16 ***
## gendermale   0.03460    0.01811    1.91   0.0561 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 23158  on 9999  degrees of freedom
## Residual deviance: 23154  on 9998  degrees of freedom
## AIC: 36107
##
## Number of Fisher Scoring iterations: 6
```

```r
# fit quasi Poisson errors
model <- glm(count ~ gender, family = quasipoisson)
summary(model) # no significant effects
```

```
##
## Call:
## glm(formula = count ~ gender, family = quasipoisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5753  -1.5483  -1.5483   0.6254   7.3023
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.18115    0.02167   8.360   <2e-16 ***
## gendermale   0.03460    0.03038   1.139    0.255
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 2.813817)
##
##     Null deviance: 23158  on 9999  degrees of freedom
## Residual deviance: 23154  on 9998  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```

```
# negative binoamial error with glm.nb
library(MASS)
model <- glm.nb(count ~ gender)
summary(model) # p value slightly different
```

```
##
## Call:
## glm.nb(formula = count ~ gender, init.theta = 0.6676246007, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1842  -1.1716  -1.1716   0.3503   3.1522
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.18115    0.02160   8.388   <2e-16 ***
## gendermale   0.03460    0.03045   1.136    0.256
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.6676) family taken to be 1)
##
##     Null deviance: 9610.8  on 9999  degrees of freedom
## Residual deviance: 9609.5  on 9998  degrees of freedom
## AIC: 30362
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.6676
##           Std. Err.:  0.0185
##
##  2 x log-likelihood:  -30355.6010
```

```
rm(gender)
detach(data)
```

A contingency table of intermediate complexity

```r
# three dimensional table of count data
numbers <- c(24, 30, 29, 41, 14, 31, 36, 35)
dim(numbers) <- c(2, 2, 2)
numbers
```

```
## , , 1
##
##      [,1] [,2]
## [1,]   24   29
## [2,]   30   41
##
## , , 2
##
##      [,1] [,2]
## [1,]   14   36
## [2,]   31   35
```

```r
dimnames(numbers)[[3]] <- list("male", "female")
dimnames(numbers)[[2]] <- list("arts", "science")
dimnames(numbers)[[1]] <- list("freshman", "sophomore")

numbers
```

```
## , , male
##
##           arts science
## freshman    24      29
## sophomore   30      41
##
## , , female
##
##           arts science
## freshman    14      36
## sophomore   31      35
```

```r
# convert table into a data frame
as.data.frame.table(numbers)
```

```
##        Var1    Var2   Var3 Freq
## 1  freshman    arts   male   24
## 2 sophomore    arts   male   30
## 3  freshman science   male   29
## 4 sophomore science   male   41
## 5  freshman    arts female   14
## 6 sophomore    arts female   31
## 7  freshman science female   36
## 8 sophomore science female   35
```

```r
frame <- as.data.frame.table(numbers)
names(frame) <- c("year", "discipline", "gender", "count")
frame
```

```
##        year discipline gender count
## 1  freshman       arts   male    24
## 2 sophomore       arts   male    30
```

```
## 3  freshman    science   male    29
## 4 sophomore    science   male    41
## 5  freshman       arts female    14
## 6 sophomore       arts female    31
## 7  freshman    science female    36
## 8 sophomore    science female    35
```

```
attach(frame)
model1 <- glm(count ~ year * discipline * gender, family = poisson)

model2 <- update(model1, ~. - year:discipline:gender)

anova(model1, model2, test = "Chi") # no significant difference
```

```
## Analysis of Deviance Table
##
## Model 1: count ~ year * discipline * gender
## Model 2: count ~ year + discipline + gender + year:discipline + year:gender +
##     discipline:gender
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         0     0.0000
## 2         1     3.0823 -1  -3.0823  0.07915 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
detach(frame)
```

Schoener's lizards: A complex contingency table

Test if there are any separation across various factors and whether there are any interactions.

```
lizards <- read.table("lizards.txt", header = TRUE)
attach(lizards)
names(lizards)
```

```
## [1] "n"       "sun"     "height"  "perch"   "time"    "species"
```

```
# n is response variable


# saturated model
model1 <- glm(n ~ sun * height * perch * time * species, family = poisson)

# remove the highest order interaction
model2 <- update(model1, ~.-sun:height:perch:time:species)
```

```
## Warning: glm.fit: fitted rates numerically 0 occurred
```

```
anova(model1, model2, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: n ~ sun * height * perch * time * species
## Model 2: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:species +
```

```
##     sun:perch:species + height:perch:species + sun:time:species +
##     height:time:species + perch:time:species + sun:height:perch:time +
##     sun:height:perch:species + sun:height:time:species + sun:perch:time:species +
##     height:perch:time:species
##   Resid. Df Resid. Dev Df   Deviance Pr(>Chi)
## 1         0 3.3473e-10
## 2         2 2.1808e-10 -2 1.1665e-10
```

```
# deviance is close to zero, no p value produed

# remove a kind of four way interaction
model3 <- update(model2, ~.-sun:height:perch:species)
```

```
## Warning: glm.fit: fitted rates numerically 0 occurred
```

```
anova(model2, model3, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:species +
##     sun:perch:species + height:perch:species + sun:time:species +
##     height:time:species + perch:time:species + sun:height:perch:time +
##     sun:height:perch:species + sun:height:time:species + sun:perch:time:species +
##     height:perch:time:species
## Model 2: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:species +
##     sun:perch:species + height:perch:species + sun:time:species +
##     height:time:species + perch:time:species + sun:height:perch:time +
##     sun:height:time:species + sun:perch:time:species + height:perch:time:species
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         2     0.0000
## 2         3     2.7088 -1  -2.7088   0.0998 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# remove another four-way interaction
model4 <- update(model2, ~.-sun:height:time:species)
anova(model2, model4, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:species +
##     sun:perch:species + height:perch:species + sun:time:species +
##     height:time:species + perch:time:species + sun:height:perch:time +
##     sun:height:perch:species + sun:height:time:species + sun:perch:time:species +
##     height:perch:time:species
## Model 2: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
```

```
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:species +
##     sun:perch:species + height:perch:species + sun:time:species +
##     height:time:species + perch:time:species + sun:height:perch:time +
##     sun:height:perch:species + sun:perch:time:species + height:perch:time:species
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         2    0.00000
## 2         4    0.44164 -2 -0.44164   0.8019
```

```r
model5 <- update(model2, ~.-sun:perch:time:species)
```

```
## Warning: glm.fit: fitted rates numerically 0 occurred
```

```r
anova(model2, model5, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:species +
##     sun:perch:species + height:perch:species + sun:time:species +
##     height:time:species + perch:time:species + sun:height:perch:time +
##     sun:height:perch:species + sun:height:time:species + sun:perch:time:species +
##     height:perch:time:species
## Model 2: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:species +
##     sun:perch:species + height:perch:species + sun:time:species +
##     height:time:species + perch:time:species + sun:height:perch:time +
##     sun:height:perch:species + sun:height:time:species + height:perch:time:species
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         2    0.00000
## 2         4    0.81008 -2 -0.81008    0.667
```

```r
model6 <- update(model2, ~.-height:perch:time:species)
```

```
## Warning: glm.fit: fitted rates numerically 0 occurred
```

```r
anova(model2, model6, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:species +
##     sun:perch:species + height:perch:species + sun:time:species +
##     height:time:species + perch:time:species + sun:height:perch:time +
##     sun:height:perch:species + sun:height:time:species + sun:perch:time:species +
##     height:perch:time:species
## Model 2: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:species +
##     sun:perch:species + height:perch:species + sun:time:species +
```

```
##     height:time:species + perch:time:species + sun:height:perch:time +
##     sun:height:perch:species + sun:height:time:species + sun:perch:time:species
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         2     0.0000
## 2         4     3.2217 -2  -3.2217   0.1997
```

```
model7 <- step(model1, lower = ~sun*height*perch*time) # still two four way interactions left
```

```
## Start:  AIC=259.25
## n ~ sun * height * perch * time * species

## Warning: glm.fit: fitted rates numerically 0 occurred

##                                   Df   Deviance    AIC
## - sun:height:perch:time:species  2 2.1808e-10 255.25
## <none>                              3.3473e-10 259.25

## Warning: glm.fit: fitted rates numerically 0 occurred

##
## Step:  AIC=255.25
## n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:species +
##     sun:perch:species + height:perch:species + sun:time:species +
##     height:time:species + perch:time:species + sun:height:perch:time +
##     sun:height:perch:species + sun:height:time:species + sun:perch:time:species +
##     height:perch:time:species

## Warning: glm.fit: fitted rates numerically 0 occurred

## Warning: glm.fit: fitted rates numerically 0 occurred

## Warning: glm.fit: fitted rates numerically 0 occurred

## Warning: glm.fit: fitted rates numerically 0 occurred

##                               Df Deviance    AIC
## - sun:height:time:species    2   0.4416 251.69
## - sun:perch:time:species     2   0.8101 252.06
## - height:perch:time:species  2   3.2217 254.47
## <none>                           0.0000 255.25
## - sun:height:perch:species   1   2.7088 255.96
## - sun:height:perch:time      2   4.7901 256.04
##
## Step:  AIC=251.69
## n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:species +
##     sun:perch:species + height:perch:species + sun:time:species +
##     height:time:species + perch:time:species + sun:height:perch:time +
##     sun:height:perch:species + sun:perch:time:species + height:perch:time:species
##
##                               Df Deviance    AIC
## - sun:perch:time:species     2   1.0713 248.32
## <none>                           0.4416 251.69
```

```
## - height:perch:time:species  2   4.6476 251.90
## - sun:height:perch:time      2   4.9482 252.20
## - sun:height:perch:species   1   3.1113 252.36
##
## Step:  AIC=248.32
## n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:species +
##     sun:perch:species + height:perch:species + sun:time:species +
##     height:time:species + perch:time:species + sun:height:perch:time +
##     sun:height:perch:species + height:perch:time:species
##
##                             Df Deviance    AIC
## - sun:time:species           2   3.3403 246.59
## <none>                           1.0713 248.32
## - sun:height:perch:time      2   5.1261 248.38
## - sun:height:perch:species   1   3.3016 248.55
## - height:perch:time:species  2   5.7906 249.04
##
## Step:  AIC=246.59
## n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:species +
##     sun:perch:species + height:perch:species + height:time:species +
##     perch:time:species + sun:height:perch:time + sun:height:perch:species +
##     height:perch:time:species
##
##                             Df Deviance    AIC
## <none>                           3.3403 246.59
## - sun:height:perch:time      2   7.5288 246.78
## - sun:height:perch:species   1   5.8273 247.08
## - height:perch:time:species  2   8.5418 247.79
```

```r
# lower argument prevent step from removing any interactions that don NOT
# involve species, as they're essential


# start from the lower model and all three way interactions
model8 <- glm(n ~ sun*height*perch*time + (species + sun + height + perch + time)^3,
              family = poisson)

summary(model8)
```

```
##
## Call:
## glm(formula = n ~ sun * height * perch * time + (species + sun +
##     height + perch + time)^3, family = poisson)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.18357  -0.27937  -0.00012   0.12000   1.15977
##
## Coefficients:
```

```
##                                             Estimate Std. Error z value
## (Intercept)                                 1.06625    0.52811   2.019
## sunSun                                      1.21539    0.58798   2.067
## heightLow                                  -1.63040    1.12713  -1.447
## perchNarrow                                 0.38966    0.63802   0.611
## timeMid.day                                -1.11001    0.92393  -1.201
## timeMorning                                -0.16480    0.71221  -0.231
## speciesopalinus                             0.56212    0.61592   0.913
## sunSun:heightLow                            0.66622    1.19017   0.560
## sunSun:perchNarrow                         -0.59007    0.70997  -0.831
## heightLow:perchNarrow                      -1.25975    1.05466  -1.194
## sunSun:timeMid.day                          1.80232    0.95383   1.890
## sunSun:timeMorning                          0.39383    0.76179   0.517
## heightLow:timeMid.day                      -1.49370    0.95902  -1.558
## heightLow:timeMorning                      -2.01611    0.93456  -2.157
## perchNarrow:timeMid.day                    -0.34518    0.87999  -0.392
## perchNarrow:timeMorning                    -0.42147    0.77703  -0.542
## sunSun:speciesopalinus                      0.05806    0.67203   0.086
## heightLow:speciesopalinus                   2.43838    1.14061   2.138
## perchNarrow:speciesopalinus                -0.70656    0.70012  -1.009
## timeMid.day:speciesopalinus                 1.56643    0.95316   1.643
## timeMorning:speciesopalinus                 1.50873    0.75255   2.005
## sunSun:heightLow:perchNarrow                1.38366    1.09944   1.259
## sunSun:heightLow:timeMid.day                0.97270    0.79734   1.220
## sunSun:heightLow:timeMorning                1.61997    0.73600   2.201
## sunSun:perchNarrow:timeMid.day              1.05204    0.85811   1.226
## sunSun:perchNarrow:timeMorning              0.73858    0.76537   0.965
## heightLow:perchNarrow:timeMid.day         -17.44646 4042.65287  -0.004
## heightLow:perchNarrow:timeMorning           1.58932    1.12644   1.411
## sunSun:heightLow:speciesopalinus           -1.86918    1.19706  -1.561
## sunSun:perchNarrow:speciesopalinus          0.08280    0.69114   0.120
## sunSun:timeMid.day:speciesopalinus         -0.92022    0.97309  -0.946
## sunSun:timeMorning:speciesopalinus         -1.15262    0.78494  -1.468
## heightLow:perchNarrow:speciesopalinus      -0.29341    0.53313  -0.550
## heightLow:timeMid.day:speciesopalinus       0.67511    0.66818   1.010
## heightLow:timeMorning:speciesopalinus       0.79583    0.72632   1.096
## perchNarrow:timeMid.day:speciesopalinus    -0.03659    0.56916  -0.064
## perchNarrow:timeMorning:speciesopalinus    -0.08054    0.60930  -0.132
## sunSun:heightLow:perchNarrow:timeMid.day   16.83723 4042.65291   0.004
## sunSun:heightLow:perchNarrow:timeMorning   -1.99076    1.30454  -1.526
##                                           Pr(>|z|)
## (Intercept)                                 0.0435 *
## sunSun                                      0.0387 *
## heightLow                                   0.1480
## perchNarrow                                 0.5414
## timeMid.day                                 0.2296
## timeMorning                                 0.8170
## speciesopalinus                             0.3614
## sunSun:heightLow                            0.5756
## sunSun:perchNarrow                          0.4059
## heightLow:perchNarrow                       0.2323
## sunSun:timeMid.day                          0.0588 .
## sunSun:timeMorning                          0.6052
## heightLow:timeMid.day                       0.1193
```

```
## heightLow:timeMorning                      0.0310 *
## perchNarrow:timeMid.day                    0.6949
## perchNarrow:timeMorning                    0.5875
## sunSun:speciesopalinus                     0.9312
## heightLow:speciesopalinus                  0.0325 *
## perchNarrow:speciesopalinus                0.3129
## timeMid.day:speciesopalinus                0.1003
## timeMorning:speciesopalinus                0.0450 *
## sunSun:heightLow:perchNarrow               0.2082
## sunSun:heightLow:timeMid.day               0.2225
## sunSun:heightLow:timeMorning               0.0277 *
## sunSun:perchNarrow:timeMid.day             0.2202
## sunSun:perchNarrow:timeMorning             0.3345
## heightLow:perchNarrow:timeMid.day          0.9966
## heightLow:perchNarrow:timeMorning          0.1583
## sunSun:heightLow:speciesopalinus           0.1184
## sunSun:perchNarrow:speciesopalinus         0.9046
## sunSun:timeMid.day:speciesopalinus         0.3443
## sunSun:timeMorning:speciesopalinus         0.1420
## heightLow:perchNarrow:speciesopalinus      0.5821
## heightLow:timeMid.day:speciesopalinus      0.3123
## heightLow:timeMorning:speciesopalinus      0.2732
## perchNarrow:timeMid.day:speciesopalinus    0.9487
## perchNarrow:timeMorning:speciesopalinus    0.8948
## sunSun:heightLow:perchNarrow:timeMid.day   0.9967
## sunSun:heightLow:perchNarrow:timeMorning   0.1270
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 737.555  on 47  degrees of freedom
## Residual deviance:   8.573  on  9  degrees of freedom
## AIC: 249.82
##
## Number of Fisher Scoring iterations: 17
```

```r
# remove the four-way interaction
model9 <- step(model8, lower = ~sun*height*perch*time, trace = FALSE)

model10 <- update(model9, ~. -sun:height:species)
anova(model9, model10, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:species +
##     sun:height:perch:time
## Model 2: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:perch:time
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
```

```
## 1        17     11.984
## 2        18     14.205 -1  -2.2203   0.1362
```

```
# remove two-way interaction
model11 <- update(model10, ~. -sun:species)
model12 <- update(model10, ~. -height:species)
model13 <- update(model10, ~. -perch:species)
model14 <- update(model10, ~. -time:species)
anova(model10, model11, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:perch:time
## Model 2: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + height:species +
##     perch:species + time:species + sun:height:perch + sun:height:time +
##     sun:perch:time + height:perch:time + sun:height:perch:time
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        18     14.205
## 2        19     21.892 -1  -7.6871 0.005562 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(model10, model12, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:perch:time
## Model 2: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     perch:species + time:species + sun:height:perch + sun:height:time +
##     sun:perch:time + height:perch:time + sun:height:perch:time
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        18     14.205
## 2        19     36.271 -1  -22.066 2.634e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(model10, model13, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:perch:time
## Model 2: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + time:species + sun:height:perch + sun:height:time +
##     sun:perch:time + height:perch:time + sun:height:perch:time
```

```
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        18     14.205
## 2        19     27.335 -1   -13.13 0.0002906 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(model10, model14, test = "Chi") # significant
```

```
## Analysis of Deviance Table
##
## Model 1: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:perch:time
## Model 2: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + sun:height:perch + sun:height:time +
##     sun:perch:time + height:perch:time + sun:height:perch:time
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        18     14.205
## 2        20     25.802 -2  -11.597 0.003032 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# a summary table
ftable(tapply(n, list(species, sun, height, perch, time), sum))
```

```
##                             Afternoon Mid.day Morning
##
## grahamii Shade High Broad           4       1       2
##                     Narrow          3       1       3
##                Low  Broad           0       0       0
##                     Narrow          1       0       0
##          Sun   High Broad          10      20      11
##                     Narrow          8      32      15
##                Low  Broad           3       4       5
##                     Narrow          4       5       1
## opalinus Shade High Broad           4       8      20
##                     Narrow          5       4       8
##                Low  Broad          12       8      13
##                     Narrow          1       0       6
##          Sun   High Broad          18      69      34
##                     Narrow          8      60      17
##                Low  Broad          13      55      31
##                     Narrow          4      21      12
```

```
# check if we need to keep all three levels for time of day
tod <- factor(1 + (time == "Afternoon"))
model15 <- update(model10, ~.-species:time+species:tod)
anova(model10, model15, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + time:species + sun:height:perch +
```

```
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:perch:time
## Model 2: n ~ sun + height + perch + time + species + sun:height + sun:perch +
##     height:perch + sun:time + height:time + perch:time + sun:species +
##     height:species + perch:species + species:tod + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time + sun:height:perch:time
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        18     14.205
## 2        19     15.023 -1 -0.81863   0.3656
```

```
# two levels are ok

detach(lizards)
```

Plot methods fro contingency tables

`assocplot` produce a Cohen-Friendly association plot indicating deviations from independence of rows and columns in a 2-dimensional contingency table.

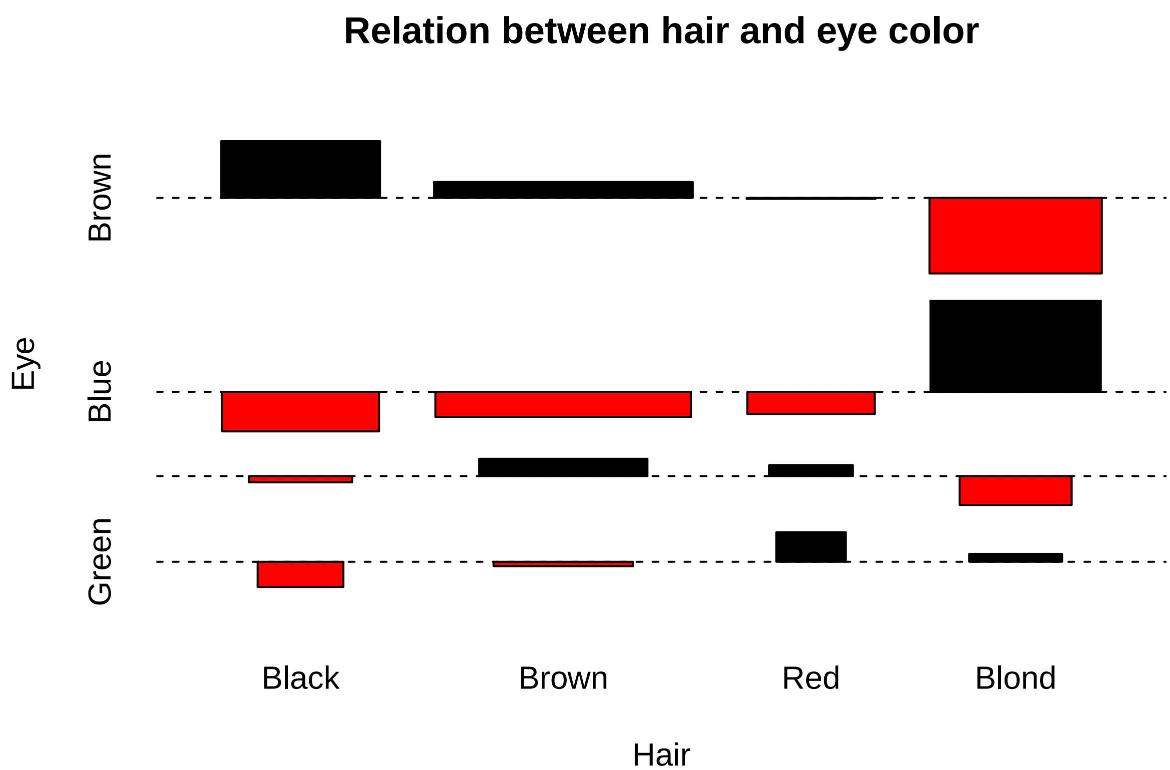
`mosaicplot` plots a mosaic on the current graphics device.

`fourfoldplot` creates a fourfold display of a 2 by 2 by k contingency table on the current graphics device, allowing for the visual inspection of the association between two dichotomous variables in one or several populations (strata)

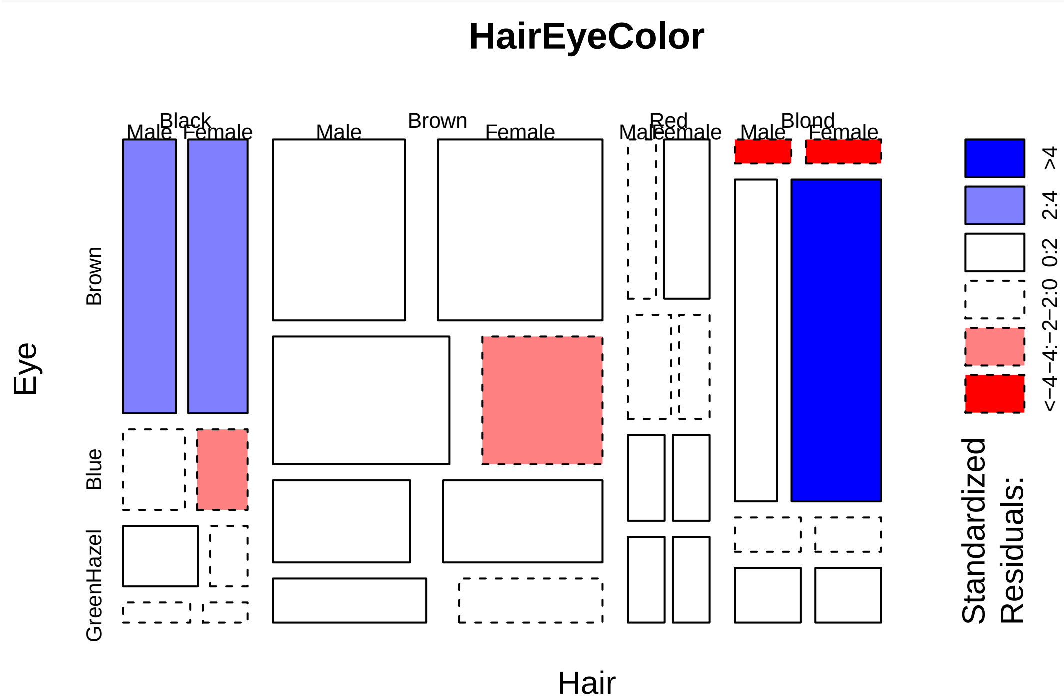
```
data(HairEyeColor)
(x <- margin.table(HairEyeColor, c(1, 2)) )
```

```
##        Eye
## Hair    Brown Blue Hazel Green
##   Black    68   20    15     5
##   Brown   119   84    54    29
##   Red      26   17    14    14
##   Blond     7   94    10    16
```

```
# margin.table computes the sum of table entries for a given index for a contingency table in array for
assocplot(x, main = "Relation between hair and eye color")
```




```r
# 1. the red bars show categories where fewer people were observed than expected
# under the null hypothesis of independence of hair color and eye color.
# 2. the black bard show the excess of people with black hair who have brown eyes etc

# same data plotted as mosaic plot
mosaicplot(HairEyeColor, shade = TRUE)
```



```r
# 1. indicates that there are significantly more blue eyed blond than expected in the case of independe
# 2. negative residuals are drawn in shades of red and with broken lines
# 3. positive residuals are drawn in shades of blue with solid lines

# admission policy of different departments
data(UCBAdmissions)
head(UCBAdmissions)
```

```
## [1] 512 313  89  19 353 207
```

```r
str(UCBAdmissions)
```

```
##  table [1:2, 1:2, 1:6] 512 313 89 19 353 207 17 8 120 205 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ Admit : chr [1:2] "Admitted" "Rejected"
##   ..$ Gender: chr [1:2] "Male" "Female"
##   ..$ Dept  : chr [1:6] "A" "B" "C" "D" ...
```

```r
x <- aperm(UCBAdmissions, c(2, 1, 3)) # transpose the x an y for each table
# Transpose an array by permuting its dimensions and optionally resizing it.
x
```

```
## , , Dept = A
##
##         Admit
## Gender   Admitted Rejected
##   Male        512      313
##   Female       89       19
##
## , , Dept = B
##
##         Admit
## Gender   Admitted Rejected
##   Male        353      207
##   Female       17        8
##
## , , Dept = C
##
##         Admit
## Gender   Admitted Rejected
##   Male        120      205
##   Female      202      391
##
## , , Dept = D
##
##         Admit
## Gender   Admitted Rejected
##   Male        138      279
##   Female      131      244
##
## , , Dept = E
##
##         Admit
## Gender   Admitted Rejected
##   Male         53      138
##   Female       94      299
```

```
##
## , , Dept = F
##
##         Admit
## Gender   Admitted Rejected
##   Male         22      351
##   Female       24      317
```

```r
UCBAdmissions
```

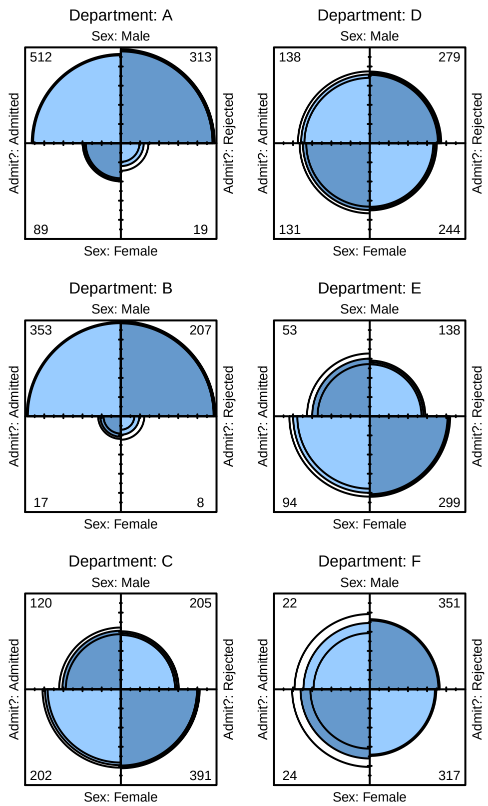
```
## , , Dept = A
##
##           Gender
## Admit      Male Female
##   Admitted  512     89
##   Rejected  313     19
##
## , , Dept = B
##
##           Gender
## Admit      Male Female
##   Admitted  353     17
##   Rejected  207      8
##
## , , Dept = C
##
##           Gender
## Admit      Male Female
##   Admitted  120    202
##   Rejected  205    391
##
## , , Dept = D
##
##           Gender
## Admit      Male Female
##   Admitted  138    131
##   Rejected  279    244
##
## , , Dept = E
##
##           Gender
## Admit      Male Female
##   Admitted   53     94
##   Rejected  138    299
##
## , , Dept = F
##
##           Gender
## Admit      Male Female
##   Admitted   22     24
##   Rejected  351    317
```

```r
names(dimnames(x)) <- c("Sex", "Admit?", "Department")
ftable(x)
```

```
##                 Department   A   B   C   D   E   F
```

```
## Sex    Admit?
## Male   Admitted            512 353 120 138  53  22
##        Rejected            313 207 205 279 138 351
## Female Admitted             89  17 202 131  94  24
##        Rejected             19   8 391 244 299 317
```

```r
fourfoldplot(x, margin = 2)
```




```r
# use gl to generate factor levels
dept <- gl(6, 4)
dept
```

```
##  [1] 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6
## Levels: 1 2 3 4 5 6
```

```r
sex <- gl(2, 1, 24)
sex
```

```
##  [1] 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
## Levels: 1 2
```

```r
admit <- gl(2, 2, 24)
admit
```

```
##  [1] 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2
## Levels: 1 2
```

```r
model1 <- glm(as.vector(x) ~ dept*sex*admit, family = poisson)
model2 <- update(model1, ~. -dept:sex:admit)
anova(model1, model2, test = "Chi")
```

```
## Analysis of Deviance Table
```

```
##
## Model 1: as.vector(x) ~ dept * sex * admit
## Model 2: as.vector(x) ~ dept + sex + admit + dept:sex + dept:admit + sex:admit
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         0      0.000
## 2         5     20.204 -5  -20.204 0.001144 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# interaction significant



# another way to do the same test as above
# convert the three dim contingency table into a dataframe
admissions <- as.data.frame(UCBAdmissions)
admissions
```

```
##       Admit Gender Dept Freq
## 1  Admitted   Male    A  512
## 2  Rejected   Male    A  313
## 3  Admitted Female    A   89
## 4  Rejected Female    A   19
## 5  Admitted   Male    B  353
## 6  Rejected   Male    B  207
## 7  Admitted Female    B   17
## 8  Rejected Female    B    8
## 9  Admitted   Male    C  120
## 10 Rejected   Male    C  205
## 11 Admitted Female    C  202
## 12 Rejected Female    C  391
## 13 Admitted   Male    D  138
## 14 Rejected   Male    D  279
## 15 Admitted Female    D  131
## 16 Rejected Female    D  244
## 17 Admitted   Male    E   53
## 18 Rejected   Male    E  138
## 19 Admitted Female    E   94
## 20 Rejected Female    E  299
## 21 Admitted   Male    F   22
## 22 Rejected   Male    F  351
## 23 Admitted Female    F   24
## 24 Rejected Female    F  317
```

```
xtabs(Freq ~ Gender + Dept, admissions)
```

```
##         Dept
## Gender     A   B   C   D   E   F
##   Male   825 560 325 417 191 373
##   Female 108  25 593 375 393 341
```

```
# xtabs creates a contingency table (optionally a sparse matrix) from cross-classifying factors, usuall


summary(xtabs(Freq ~ ., admissions))
```

```
## Call: xtabs(formula = Freq ~ ., data = admissions)
## Number of cases in table: 4526
## Number of factors: 3
## Test for independence of all factors:
##  Chisq = 2000.3, df = 16, p-value = 0
```

```
str(xtabs(Freq ~ Admit + Dept + Gender, admissions))
```

```
##  xtabs [1:2, 1:6, 1:2] 512 313 353 207 120 205 138 279 53 138 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ Admit : chr [1:2] "Admitted" "Rejected"
##   ..$ Dept  : chr [1:6] "A" "B" "C" "D" ...
##   ..$ Gender: chr [1:2] "Male" "Female"
##  - attr(*, "class")= chr [1:2] "xtabs" "table"
##  - attr(*, "call")= language xtabs(formula = Freq ~ Admit + Dept + Gender, data = admissions)
```

```
xtabs(Freq ~ Admit + Dept + Gender, admissions)[, , 2]
```

```
##           Dept
## Admit        A   B   C   D   E   F
##   Admitted  89  17 202 131  94  24
##   Rejected  19   8 391 244 299 317
```

```
females <- colSums(xtabs(Freq ~ Admit + Dept + Gender, admissions)[, ,2])
females
```

```
##   A   B   C   D   E   F
## 108  25 593 375 393 341
```

```
admitted.females <- xtabs(Freq ~ Admit + Dept + Gender, admissions)[, ,2][1, ]

(female.success <- admitted.females/females)
```

```
##          A          B          C          D          E          F
## 0.82407407 0.68000000 0.34064081 0.34933333 0.23918575 0.07038123
```

```
# the success rate varies a lot
```

Graphics for count data: Spine plots and spinograms

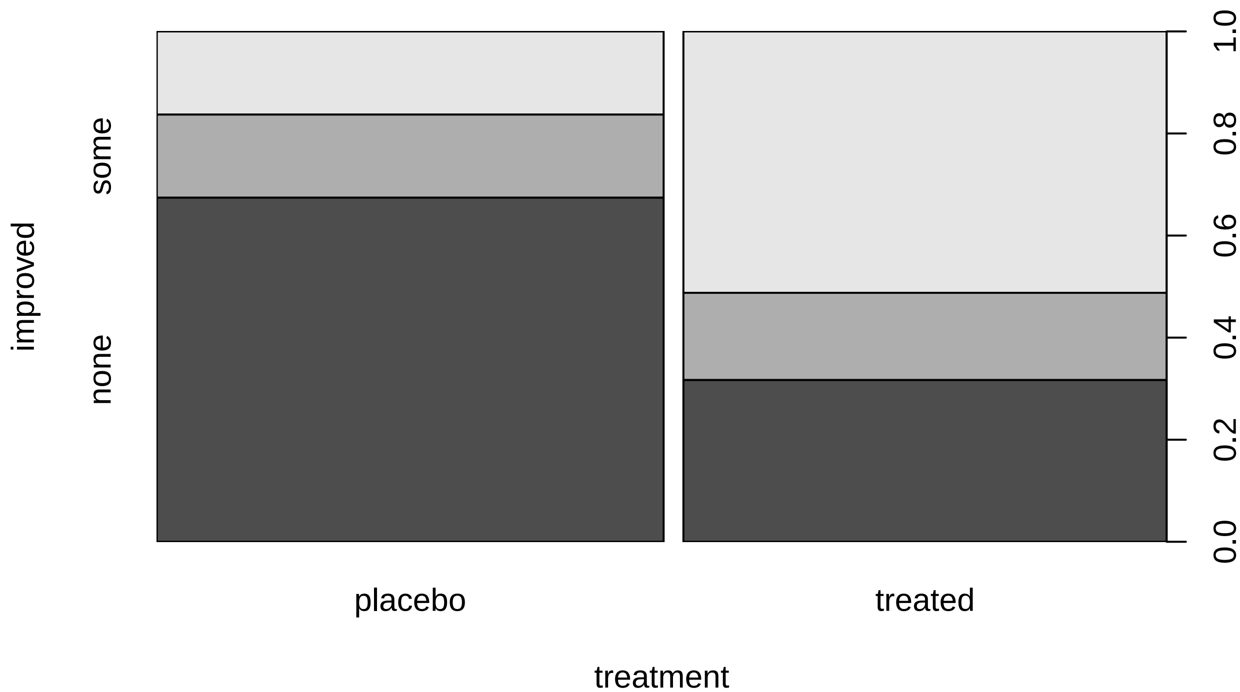
The data for this section cannot be found from the book's websit.

`spineplot` is a special cases of mosaic plots, and can be seen as a generalization of stacked (or highlighted) bar plots.

Analogously, `spinograms` are an extension of histograms.

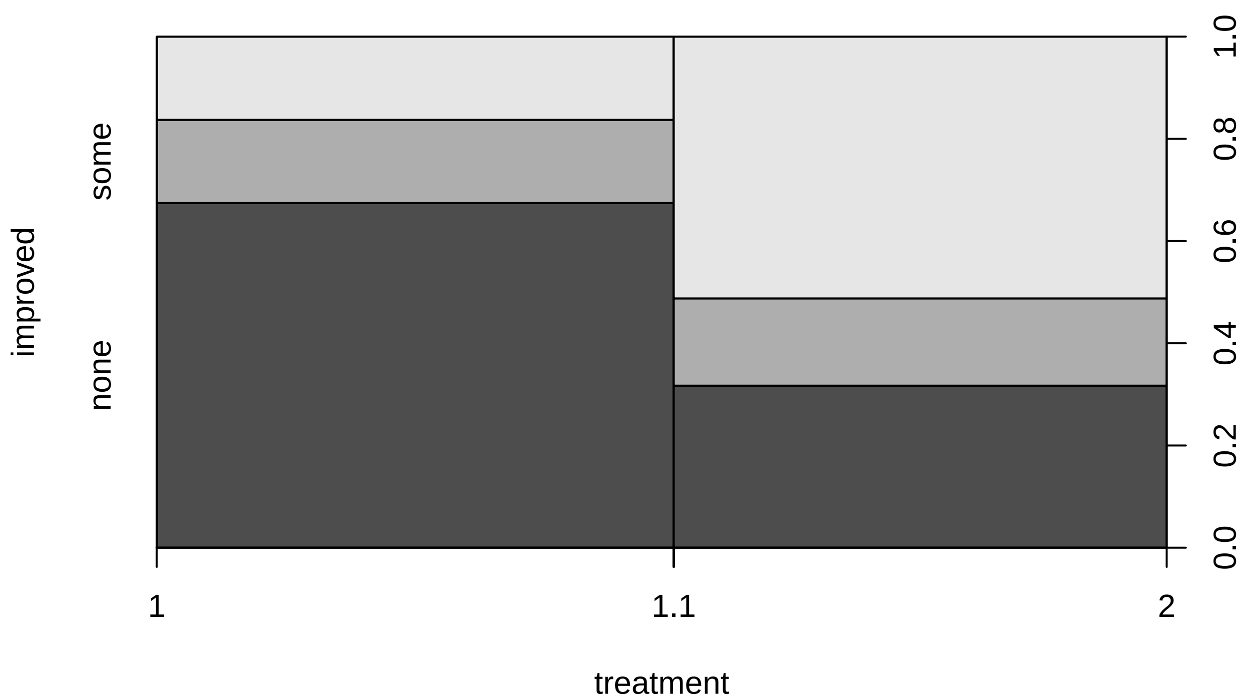
In **spineplot(x, ...)**, x can be either categorical (then a spine plot is created) or numerical (then a spinogram is plotted).

```
# treatment and improvement of patients with rheumatoid arthritis
treatment <- factor(rep(c(1, 2), c(43, 41)), levels = c(1, 2),
                    labels = c("placebo", "treated"))
improved <- factor(rep(c(1, 2, 3, 1, 2, 3), c(29, 7, 7, 13, 7, 21)),
                   levels = c(1, 2, 3),
                   labels = c("none", "some", "marked"))

## (dependence on a categorical variable)
(spineplot(improved ~ treatment))
```




```
##          improved
## treatment none some marked
##   placebo   29    7      7
##   treated   13    7     21
```

```
treatment <- as.numeric(treatment)
(spineplot(improved ~ treatment))
```




```
##              improved
## treatment   none some marked
##   [1,1.1]     29    7      7
```

```
##   (1.1,1.2]    0    0      0
##   (1.2,1.3]    0    0      0
##   (1.3,1.4]    0    0      0
##   (1.4,1.5]    0    0      0
##   (1.5,1.6]    0    0      0
##   (1.6,1.7]    0    0      0
##   (1.7,1.8]    0    0      0
##   (1.8,1.9]    0    0      0
##   (1.9,2]     13    7     21
```