STAT 441: Lecture 24
Multivariate analysis with qualitative variables
Mosaic plots

Venables and Ripley, 11.4

Qualitative - discrete variables

Can arise by themselves, or be created by `cut()` from quantitative. If we have data recorded by items, we can use `table()` to create the cross-tabulated form. Otherwise, data with qualitative variables often come already in a tabular form.

Example: Hair and eye color and sex in 592 statistics students

```
> HairEyeColor
, , Sex = Male
       Eye
Hair    Brown Blue Hazel Green
  Black    32   11    10     3
  Brown    53   50    25    15
  Red      10   10     7     7
  Blond     3   30     5     8
, , Sex = Female
       Eye
Hair    Brown Blue Hazel Green
  Black    36    9     5     2
  Brown    66   34    29    14
  Red      16    7     7     7
  Blond     4   64     5     8
> class(HairEyeColor)
[1] "table"
```

As data frame

The inverse of `table()` is

```
> hair=as.data.frame(HairEyeColor)
> hair
    Hair   Eye    Sex Freq
1  Black Brown   Male   32
2  Brown Brown   Male   53
3    Red Brown   Male   10
4  Blond Brown   Male    3
5  Black  Blue   Male   11
6  Brown  Blue   Male   50
7    Red  Blue   Male   10
8  Blond  Blue   Male   30
9  Black Hazel   Male   10
10 Brown Hazel   Male   25
11   Red Hazel   Male    7
12 Blond Hazel   Male    5
13 Black Green   Male    3
14 Brown Green   Male   15
15   Red Green   Male    7
16 Blond Green   Male    8
17 Black Brown Female   36
18 Brown Brown Female   66
19   Red Brown Female   16
20 Blond Brown Female    4
21 Black  Blue Female    9
22 Brown  Blue Female   34
23   Red  Blue Female    7
24 Blond  Blue Female   64
25 Black Hazel Female    5
26 Brown Hazel Female   29
27   Red Hazel Female    7
28 Blond Hazel Female    5
29 Black Green Female    2
30 Brown Green Female   14
31   Red Green Female    7
32 Blond Green Female    8
```

Conversion back?

Not `table()`, but `xtabs()`.

```
> xtabs(Freq~Hair+Eye+Sex,data=hair)
, , Sex = Male
       Eye
Hair    Brown Blue Hazel Green
  Black    32   11    10     3
  Brown    53   50    25    15
  Red      10   10     7     7
  Blond     3   30     5     8
, , Sex = Female
       Eye
Hair    Brown Blue Hazel Green
  Black    36    9     5     2
  Brown    66   34    29    14
  Red      16    7     7     7
  Blond     4   64     5     8
```

There may be a need to declare the right class: `class(.)="table"`.

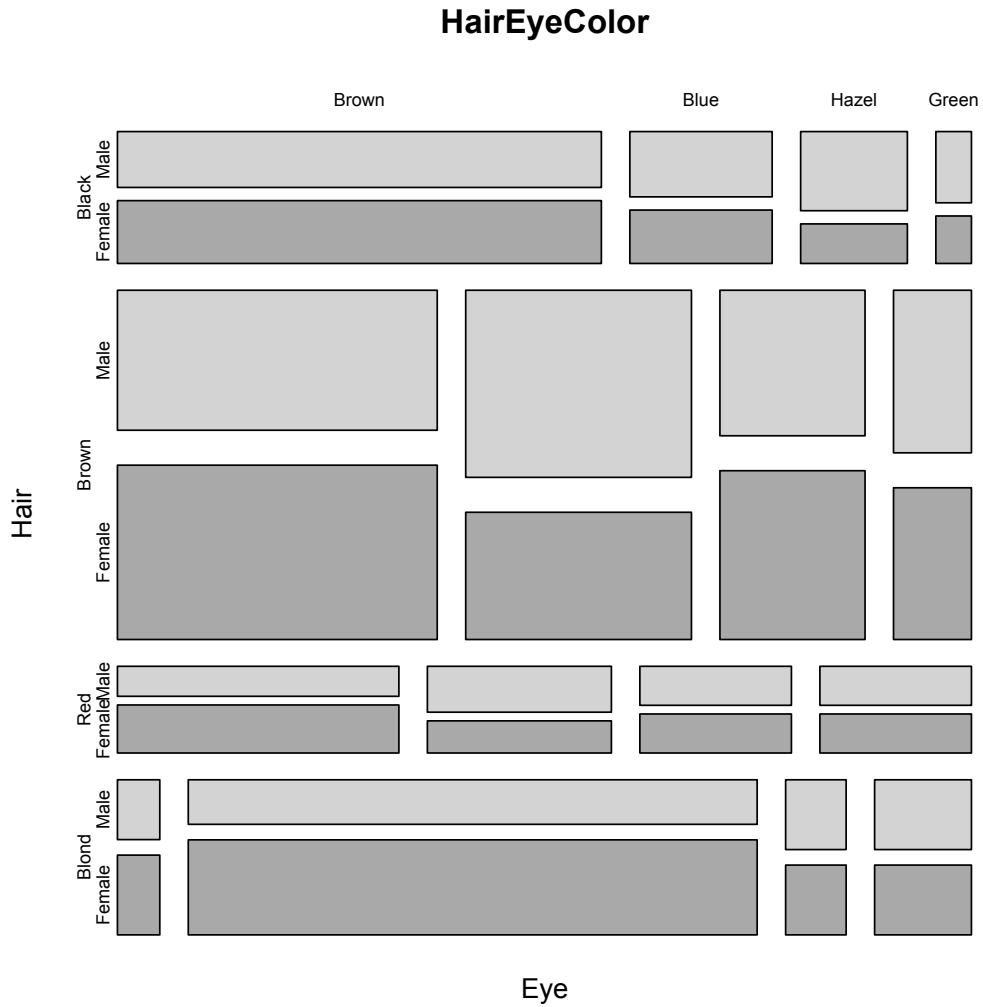
Plotting

As pie charts, we leave also the "3D histograms" to programs like Excel. A real statistician uses something else: for instance

```
> mosaicplot(HairEyeColor, color=c("lightgray","darkgray"))
```

We can control it

```
> mosaicplot(HairEyeColor,
+ color=c("lightgray","darkgray"), dir=c("h","v","h"))
```



How about eye and hair color aggregated?

That is, among "statisticiens et statisticiennes"...

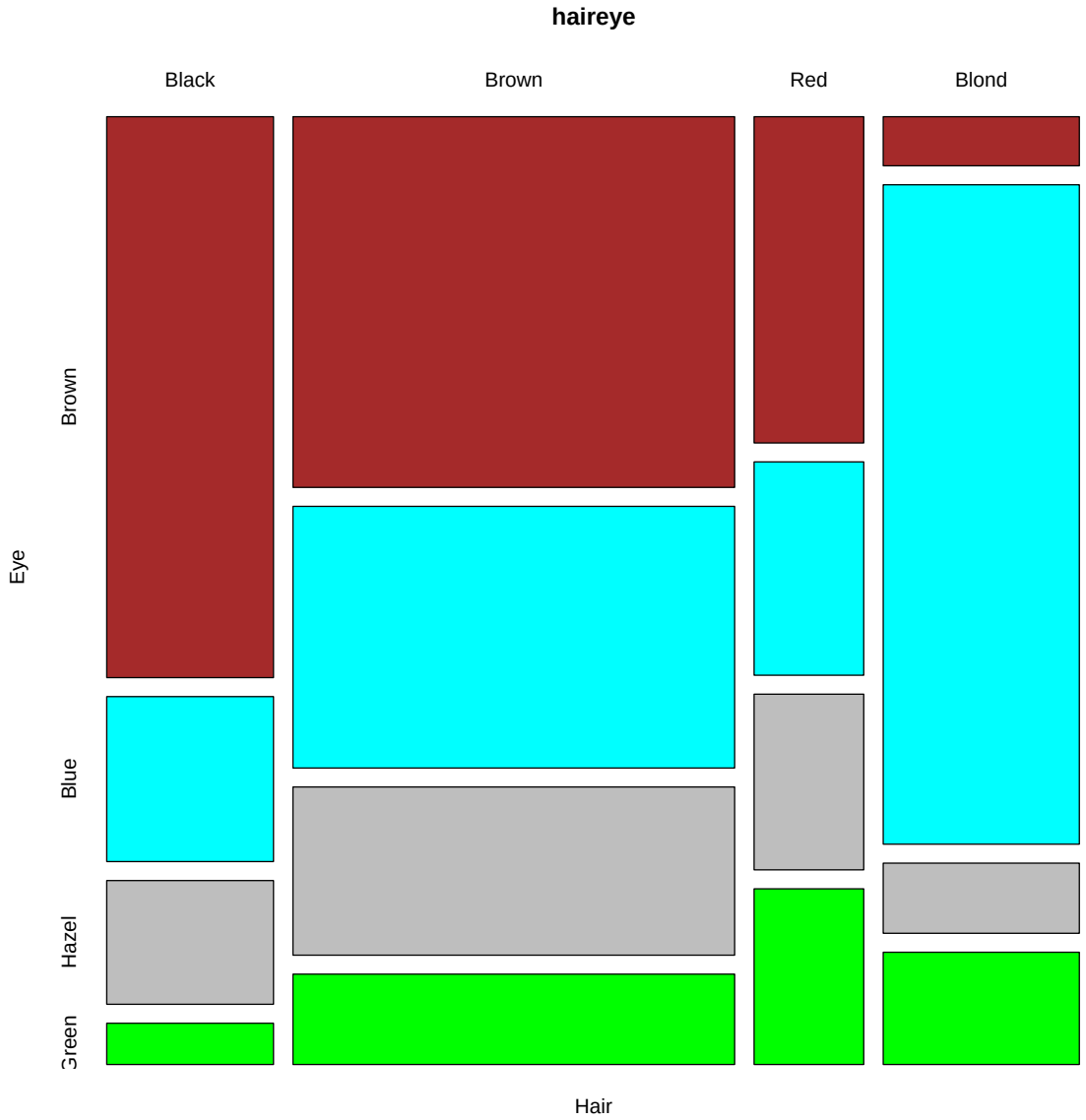
```
> haireye=apply(HairEyeColor,1:2,sum)
> haireye
       Eye
Hair    Brown Blue Hazel Green
  Black    68   20    15     5
  Brown   119   84    54    29
  Red      26   17    14    14
  Blond     7   94    10    16
> mosaicplot(haireye, color=c("brown","blue","grey","green"))
```

And give it nice colors...

Try also:

```
> mosaicplot(haireye,
+ color=c("brown","blue","grey","green"),cex=1)
```

In color



What else we can do?

Log-linear models

...as a special case of `glm()`. For frequency data, the distribution is Poisson; the link is the logarithm function. That is, we construct a linear model where the response are logarithms of *expected* frequences.

We see why we needed to convert the table to the data frame:

```
> hairglm=glm(Freq~Hair*Eye*Sex,family=poisson, data=hair)
> anova(hairglm,test="Chisq")
          Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                         31     475.12
Hair       3  165.592        28     309.53  < 2e-16 ***
Eye        3  141.272        25     168.25  < 2e-16 ***
Sex        1    1.954        24     166.30  0.16218
Hair:Eye   9  146.444        15      19.86  < 2e-16 ***
Hair:Sex   3    8.093        12      11.76  0.04413 *
Eye:Sex    3    5.002         9       6.76  0.17162
Hair:Eye:Sex  9    6.761         0       0.00  0.66196
> hairind=glm(Freq~Hair+Eye+Sex,family=poisson, data=hair)
> hairpr=predict(hairind,type="response")
```

In the context of qualitative data, significant interactions mean dependence between the corresponding variables. Here it is between Hair and Eye Colors, and also between Eye Color and Sex.

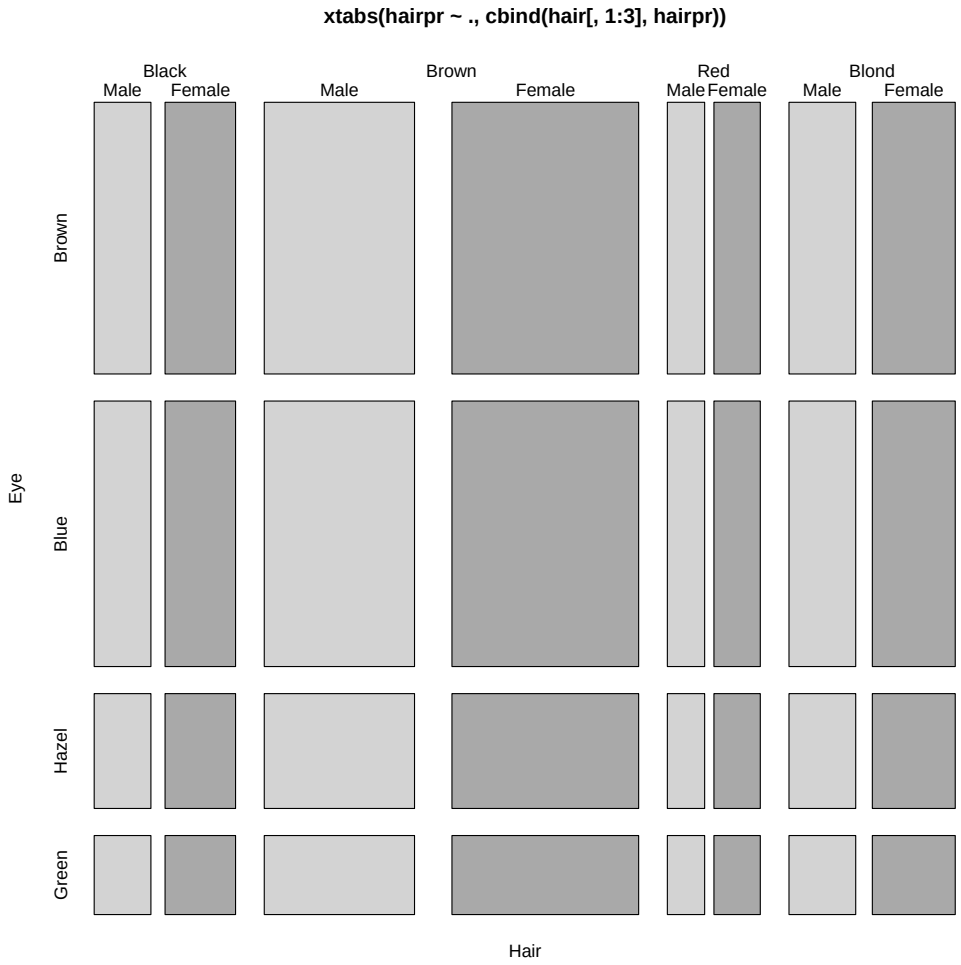
Predicted frequences under independence

Let us take predicted values under the additive model—which in the context of qualitative data means independence:all three variables, Sex, Color of Hair, and Color of Eyes, are independent.

```
> xtabs(hairpr~.,cbind(hair[,1:3],hairpr))
, , Sex = Male

       Eye
Hair        Brown      Blue     Hazel     Green
  Black 18.915038 18.485151  7.995903  5.502557
  Brown 50.089824 48.951419 21.174335 14.571585
  Red   12.434886 12.152275  5.256566  3.617421
  Blond 22.242684 21.737168  9.402589  6.470599

, , Sex = Female

       Eye
Hair        Brown      Blue     Hazel     Green
  Black 21.220097 20.737822  8.970314  6.173119
  Brown 56.193960 54.916825 23.754719 16.347334
  Red   13.950249 13.633198  5.897151  4.058254
  Blond 24.953262 24.386142 10.548424  7.259131
```

A picture?

```
> mosaicplot(xtabs(hairpr~.,cbind(hair[,1:3],hairpr)),
+ color=c("lightgray","darkgray"))
```



Not that interesting...

We would rather like to see residuals

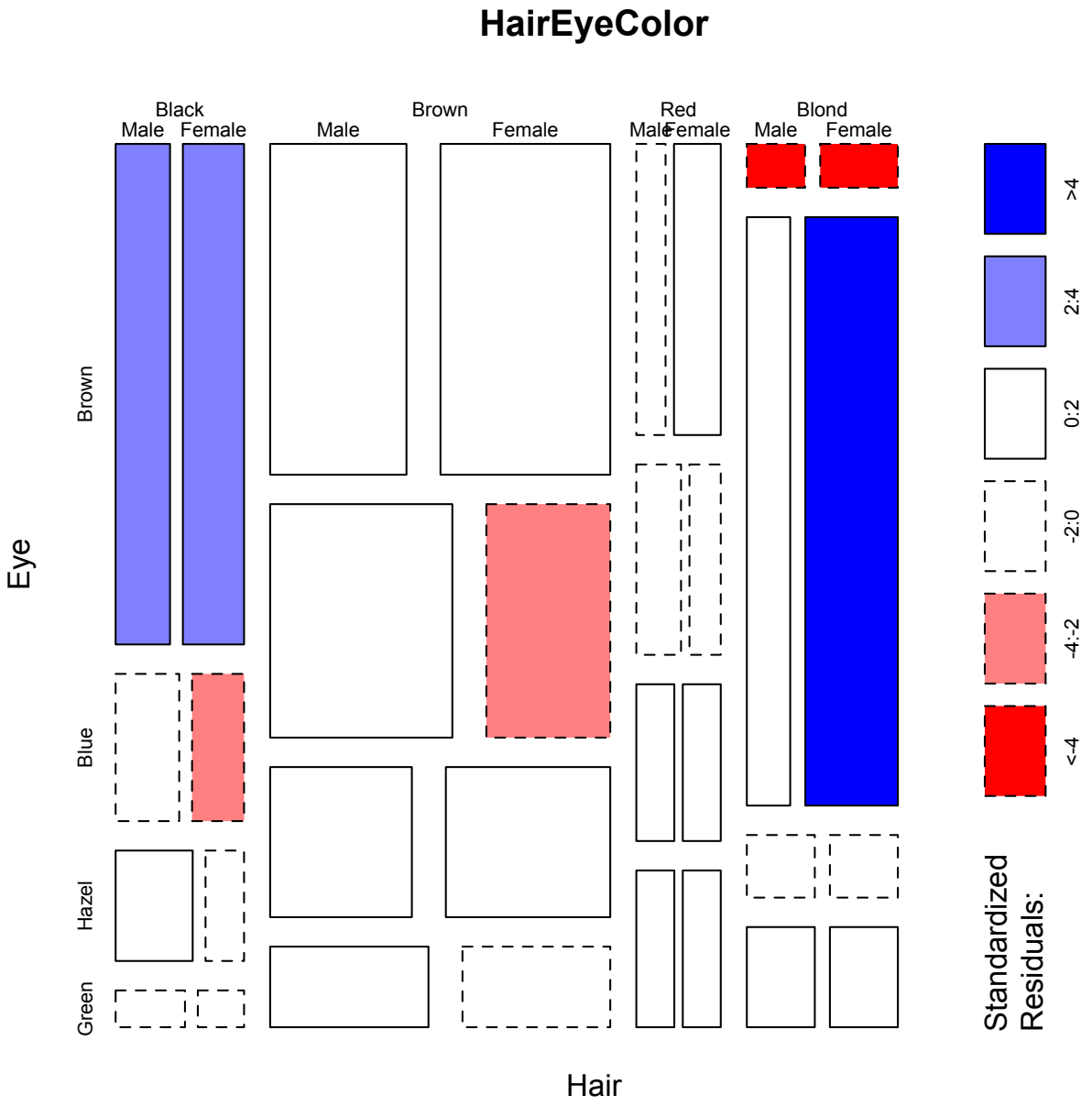
```
> hairr=residuals(hairind,type="response")
> xtabs(hairr~.,cbind(hair[,1:3],hairr))
, , Sex = Male
       Eye
Hair          Brown        Blue       Hazel       Green
  Black  13.0849617  -7.4851511   2.0040974  -2.5025566
  Brown   2.9101762   1.0485813   3.8256654   0.4284149
  Red    -2.4348863  -2.1522753   1.7434344   3.3825785
  Blond -19.2426840   8.2628316  -4.4025891   1.5294010
, , Sex = Female
       Eye
Hair          Brown        Blue       Hazel       Green
  Black  14.7799032 -11.7378219  -3.9703136  -4.1731191
  Brown   9.8060400 -20.9168246   5.2452805  -2.3473338
  Red     2.0497512  -6.6331977   1.1028494   2.9417458
  Blond -20.9532620  39.6138576  -5.5484244   0.7408692
> mosaicplot(HairEyeColor, shade=T, margin=list(1,2,3))
> hairlm=loglin(HairEyeColor, list(1, 2, 3))
2 iterations: deviation 5.684342e-14
> pchisq(hairlm$pearson, hairlm$df, lower.tail = FALSE)
[1] 5.320872e-23
```

Extended mosaic plot - with shading

The *extended mosaic plot* shows also the residuals from the log-linear fit, by shading; the fit is done not by `glm()`, but using function `loglin()`, which is handy when we want to see the result of the χ^2 test for independence. We can check that the predictions are the same:

```
> haipar=loglin(HairEyeColor,margin=list(1,2,3),para=T)$para
2 iterations: deviation 5.684342e-14
> as.vector(exp(outer(outer(outer(haipar[[1]],haipar[[2]],"+"),
+ haipar[[3]],"+"),haipar[[4]],"+")))
 [1] 18.915038 50.089824 12.434886 22.242684 18.485151 48.951419 12.152275
 [8] 21.737168  7.995903 21.174335  5.256566  9.402589  5.502557 14.571585
[15]  3.617421  6.470599 21.220097 56.193960 13.950249 24.953262 20.737822
[22] 54.916825 13.633198 24.386142  8.970314 23.754719  5.897151 10.548424
[29]  6.173119 16.347334  4.058254  7.259131
```

The plot


HairEyeColor
Black
Male
Female
Brown
Male
Female
Red
Male
Female
Blond
Male
Female
Eye
Brown
Blue
Hazel
Green
Hair
Standardized
Residuals:
>4
2:4
0:2
-2:0
-4:-2
<-4

What did we achieve?

We cans see, among other things, that there are more brown-haired, black-eyed individuals and more blond-haired, blue-eyed individuals (especially women) among statistics students than the independence model would suggest. On the other hand, blue-eyed, black- and brown-haired females are underrepresented. Also blondes with brown eyes are rare.

A technical detail: the shading in the mosaic plot works not with "raw" (`type="response"`) residuals, but "Pearson" residuals (`type="pearson"`)

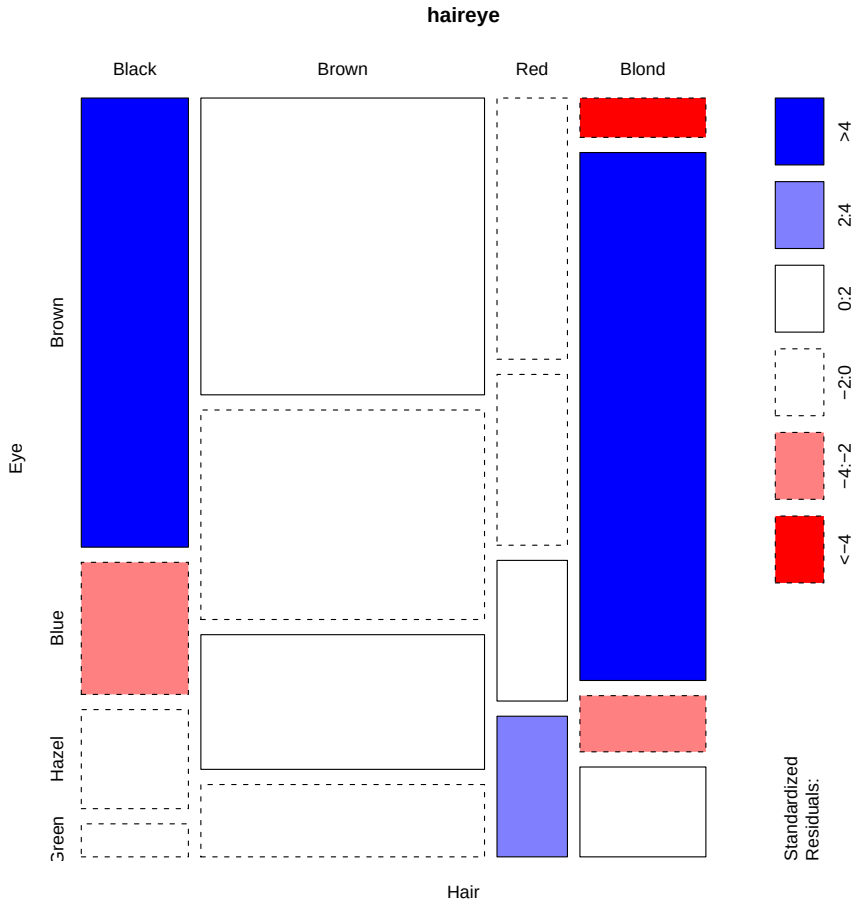
$$\frac{\mathrm{O} - \mathrm{P}}{\sqrt{\mathrm{P}}}$$

O being observed and P predicted frequency

Check it out

```
> hairpea=residuals(hairind,type="pearson")
> as.numeric(hairpea)
 [1]  3.0086304  0.4111919 -0.6904906 -4.0801117 -1.7409611  0.1498716 -0.6174034  1.7722599
 [9]  0.7087370  0.8313848  0.7604218 -1.4357685 -1.0668458  0.1122306  1.7784773  0.6012418
[17]  3.2084695  1.3081238  0.5487950 -4.1945752 -2.5775431 -2.8225591 -1.7964870  8.0218693
[25] -1.3256260  1.0762019  0.4541456 -1.7083462 -1.6796100 -0.5805657  1.4602779  0.2749788
> as.numeric((hair$Freq-hairpr)/sqrt(hairpr))
 [1]  3.0086304  0.4111919 -0.6904906 -4.0801117 -1.7409611  0.1498716 -0.6174034  1.7722599
 [9]  0.7087370  0.8313848  0.7604218 -1.4357685 -1.0668458  0.1122306  1.7784773  0.6012418
[17]  3.2084695  1.3081238  0.5487950 -4.1945752 -2.5775431 -2.8225591 -1.7964870  8.0218693
[25] -1.3256260  1.0762019  0.4541456 -1.7083462 -1.6796100 -0.5805657  1.4602779  0.2749788
> hairq=as.numeric(cut(hairpea,c(-Inf,-4,-2,0,2,4,Inf)))
> xtabs(hairq~.,cbind(hair[,1:3],hairq))
, , Sex = Male
       Eye
Hair    Brown Blue Hazel Green
  Black     5    3     4     3
  Brown     4    4     4     4
  Red       3    3     4     4
  Blond     1    4     3     4
, , Sex = Female
       Eye
Hair    Brown Blue Hazel Green
  Black     5    2     3     3
  Brown     4    2     4     3
  Red       4    3     4     4
  Blond     1    6     3     4
```

Aggregated data give similar result




```
> haind=loglin(haireye, list(1, 2))
2 iterations: deviation 0
> pchisq(haind$pearson, haind$df, lower.tail = FALSE)
[1] 2.325287e-25
```

For qualitative data, it is quite easy to get the hypothesis of independence rejected...

How to find out more?

Log-linear modeling is a way to go; for instance, we can look at the model that sex is independent conditionally on hair and eye colors. What does that mean? Color of eyes and hair are dependent, but there is no dependence between these two and the sex: the chance of encountering a blue-eye blonde man is given by the proportion of blue-eyed blondes (man or women) and the proportion of men about among statistics students

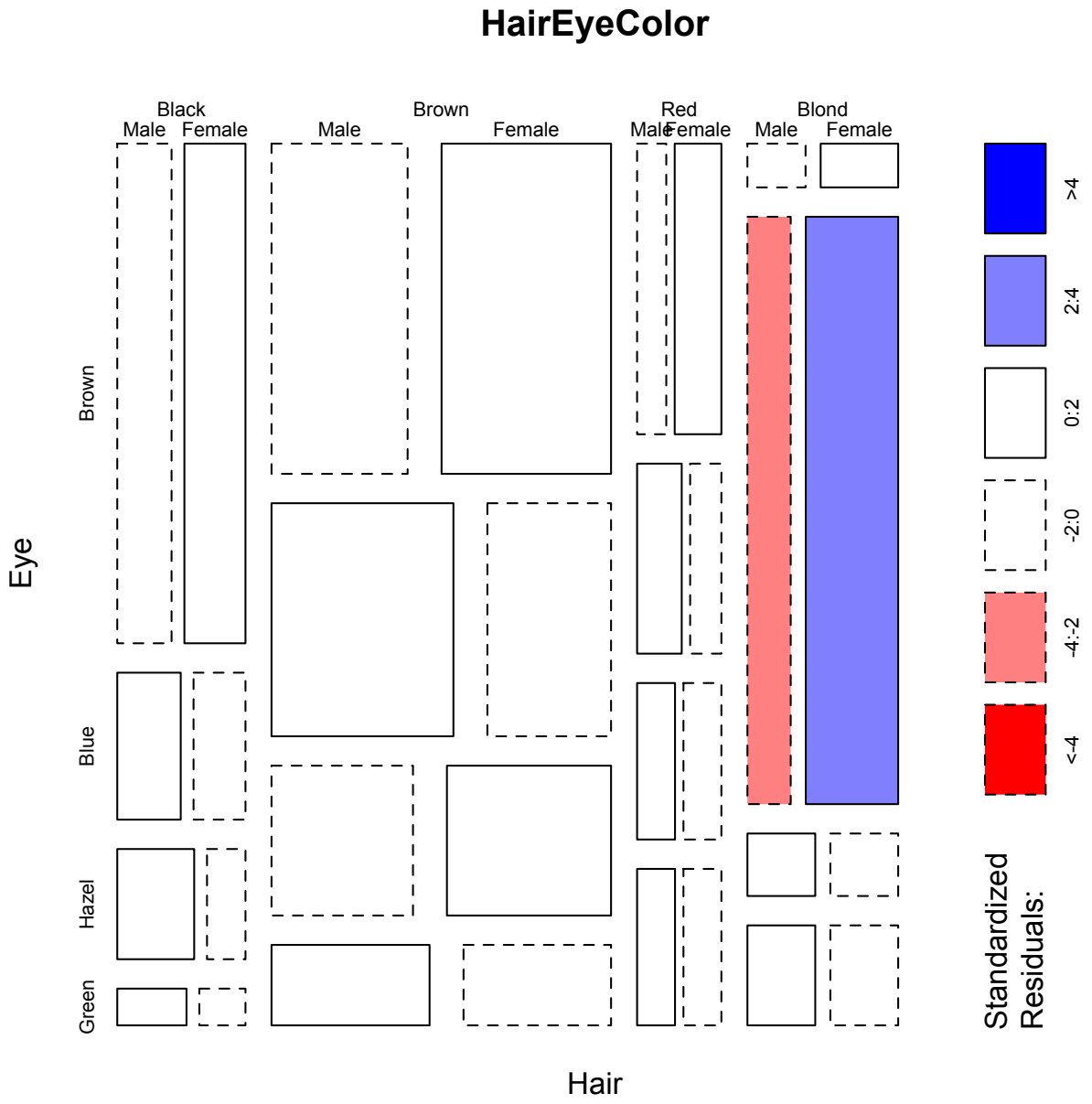
```
> haicind = loglin(HairEyeColor,margin=list(c(1,2),3))
2 iterations: deviation 5.684342e-14
> pchisq(haicind$pearson, haicind$df, lower.tail = FALSE)
[1] 0.1891745
> mosaicplot(HairEyeColor,margin=list(c(1,2),3),shade=T)
```

The plot shows that this fits much better; the most of the discrepancy (pretty much only important from the fitted model) is due to blondand blue-eyed men and women.

Lack of dependence $\equiv$ lack of (nonzero) interaction

Another way (other than log-linear modeling) to investigate the roots of dependence: correspondence analysis.

Sex vs. hair-eye independence model



What is the corresponding glm model?

```
> haircind=glm(formula = Freq ~ Hair*Eye + Sex, family = poisson, data = hair)
> as.numeric(predict(haircind,type="response"))
 [1] 32.047297 56.082770 12.253378  3.298986  9.425676 39.587838  8.011824 44.300676  7.069257
[10] 25.449324  6.597973  4.712838  2.356419 13.667230  6.597973  7.540541 35.952703 62.917230
...
> haicpar=loglin(HairEyeColor,margin=list(c(1,2),3),par=T)$param
> as.vector(exp(outer(outer(outer(haicpar[[1]],haicpar[[2]],"+"),
+ haicpar[[3]],"+"),haicpar[[4]],"+")+rep(as.vector(haicpar[[5]]),2)))
 [1] 32.047297 56.082770 12.253378  3.298986  9.425676 39.587838  8.011824 44.300676  7.069257
[10] 25.449324  6.597973  4.712838  2.356419 13.667230  6.597973  7.540541 35.952703 62.917230
...
> haircr=residuals(haircind,type="pearson")
> haircq=as.numeric(cut(haircr,c(-Inf,-4,-2,0,2,4,Inf)))
> xtabs(haircq~.,cbind(hair[,1:3],haircq))
, , Sex = Male
       Eye
Hair    Brown Blue Hazel Green
  Black     3    4     4     4
  Brown     3    4     3     4
  Red       3    4     4     4
  Blond     3    2     4     4
, , Sex = Female
       Eye
Hair    Brown Blue Hazel Green
  Black     4    3     3     3
  Brown     4    3     4     3
  Red       4    3     3     3
  Blond     4    5     3     3
```