

ASTA

The ASTA team

Contents

1	Contingency tables	2
1.1	A contingency table	2
2	Independence	3
2.1	Independence	3
2.2	The Chi-squared test for independence	3
2.3	Calculation of expected table	4
2.4	Chi-squared (χ^2) test statistic	4
2.5	χ^2 -test template.	5
2.6	The function <code>chisq.test</code>	6
3	The χ^2-distribution	6
3.1	The χ^2 -distribution	6
4	Agresti - Summary	7
4.1	Summary	7
5	Standardized residuals	8
5.1	Residual analysis	8
5.2	Residual analysis in R	8
6	Models for table data in R	8
6.1	Example	8
6.2	Model specification	9
6.3	Model specification in R	9
6.4	Expected values and standardized residuals	11
7	Log-linear models	12
7.1	Model specification	12
7.2	Example	12
7.3	Model specification	12
7.4	Example	13
7.5	Model form	13
7.6	Model comparison	14
7.7	Deviance	14

8	Higher order interaction	14
8.1	Higher order interaction	14
8.2	Bigger example	15
8.3	Test of simple model against the saturated model	17
8.4	Model reduction	17
8.5	Final model and parameter estimates	18
8.6	Predicted values	18
9	Graphical representation	20
9.1	Graphical representation	20
9.2	Interpretation of graphical model	20
9.3	Interpretation of graphical model	21
9.4	Final model of the example	22

1 Contingency tables

1.1 A contingency table

- We return to the dataset `popularKids`, where we study **association** between 2 **factors**: `Goals` and `Urban.Rural`.
- Based on a sample we make a cross tabulation of the factors and we get a so-called **contingency table** (`krydstabel`).

```
popKids <- read.delim("https://asta.math.aau.dk/datasets?file=PopularKids.txt")
library(mosaic)
tab <- tally(~Urban.Rural + Goals, data = popKids, margins = TRUE)
tab
```

##		Goals			
##	Urban.Rural	Grades	Popular	Sports	Total
##	Rural	57	50	42	149
##	Suburban	87	42	22	151
##	Urban	103	49	26	178
##	Total	247	141	90	478

1.1.1 A conditional distribution

- Another representation of data is the percent-wise distribution of `Goals` for each level of `Urban.Rural`, i.e. the sum in each row of the table is 100 (up to rounding):

```
tab <- tally(~Urban.Rural + Goals, data = popKids)
addmargins(round(100 * prop.table(tab, 1)),margin = 1:2)
```

```
##           Goals
## Urban.Rural Grades Popular Sports Sum
##   Rural      38      34      28 100
##   Suburban   58      28      15 101
##   Urban     58      28      15 101
##   Sum      154      90      58 302
```

- Here we will talk about the **conditional distribution** of `Goals` given `Urban.Rural`.
- An important question could be:
 - Are the goals of the kids different when they come from urban, suburban or rural areas? I.e. are the rows in the table significantly different?
- There is (almost) no difference between urban and suburban, but it looks like rural is different.

2 Independence

2.1 Independence

- Recall, that two factors are **independent**, when there is no difference between the population's distributions of one factor given the levels of the other factor.
- Otherwise the factors are said to be **dependent**.
- If we e.g. have the following conditional **population distributions** of `Goals` given `Urban.Rural`:

```
##           Goals
## Urban.Rural Grades Popular Sports
##   Rural      500      300      200
##   Suburban   500      300      200
##   Urban     500      300      200
```

- Then the factors `Goals` and `Urban.Rural` are independent.
- We take a sample and “measure” the factors F_1 and F_2 . E.g. `Goals` and `Urban.Rural` for a random child.
- The hypothesis of interest today is:

$$H_0 : F_1 \text{ and } F_2 \text{ are independent, } H_a : F_1 \text{ and } F_2 \text{ are dependent.}$$

2.2 The Chi-squared test for independence

- Our best guess of the distribution of `Goals` is the relative frequencies in the sample:

```
n <- margin.table(tab)
pctGoals <- round(100 * margin.table(tab, 2)/n, 1)
pctGoals
```

```
## Goals
## Grades Popular Sports
##   51.7   29.5   18.8
```

- If we assume independence, then this is also a guess of the conditional distributions of `Goals` given `Urban.Rural`.

- The corresponding expected counts in the sample are then:

```
##           Goals
## Urban.Rural Grades      Popular      Sports      Sum
##   Rural    77.0 (51.7%)  44.0 (29.5%)  28.1 (18.8%) 149.0 (100%)
##   Suburban 78.0 (51.7%)  44.5 (29.5%)  28.4 (18.8%) 151.0 (100%)
##   Urban    92.0 (51.7%)  52.5 (29.5%)  33.5 (18.8%) 178.0 (100%)
##   Sum     247.0 (51.7%) 141.0 (29.5%)  90.0 (18.8%) 478.0 (100%)
```

2.3 Calculation of expected table

pctexptab

```
##           Goals
## Urban.Rural Grades      Popular      Sports      Sum
##   Rural    77.0 (51.7%)  44.0 (29.5%)  28.1 (18.8%) 149.0 (100%)
##   Suburban 78.0 (51.7%)  44.5 (29.5%)  28.4 (18.8%) 151.0 (100%)
##   Urban    92.0 (51.7%)  52.5 (29.5%)  33.5 (18.8%) 178.0 (100%)
##   Sum     247.0 (51.7%) 141.0 (29.5%)  90.0 (18.8%) 478.0 (100%)
```

- We note that
 - The relative frequency for a given column is `columnTotal` divided by `tableTotal`. For example **Grades**, which is $\frac{247}{478} = 51.7\%$.
 - The expected value in a given cell in the table is then the cell's relative column frequency multiplied by the cell's `rowTotal`. For example **Rural** and **Grades**: $149 \times 51.7\% = 77.0$.
- This can be summarized to:
 - The expected value in a cell is the product of the cell's `rowTotal` and `columnTotal` divided by `tableTotal`.

2.4 Chi-squared (χ^2) test statistic

- We have an **observed table**:

tab

```
##           Goals
## Urban.Rural Grades Popular Sports
##   Rural      57      50      42
##   Suburban   87      42      22
##   Urban     103      49      26
```

- And an **expected table**, if H_0 is true:

```
##           Goals
## Urban.Rural Grades Popular Sports Sum
##   Rural    77.0  44.0  28.1 149.0
##   Suburban 78.0  44.5  28.4 151.0
##   Urban    92.0  52.5  33.5 178.0
##   Sum     247.0 141.0  90.0 478.0
```

- If these tables are “far from each other”, then we reject H_0 . We want to measure the distance via the Chi-squared test statistic:

- $X^2 = \sum \frac{(f_o - f_e)^2}{f_e}$: Sum over all cells in the table
- f_o is the frequency in a cell in the observed table
- f_e is the corresponding frequency in the expected table.

- We have:

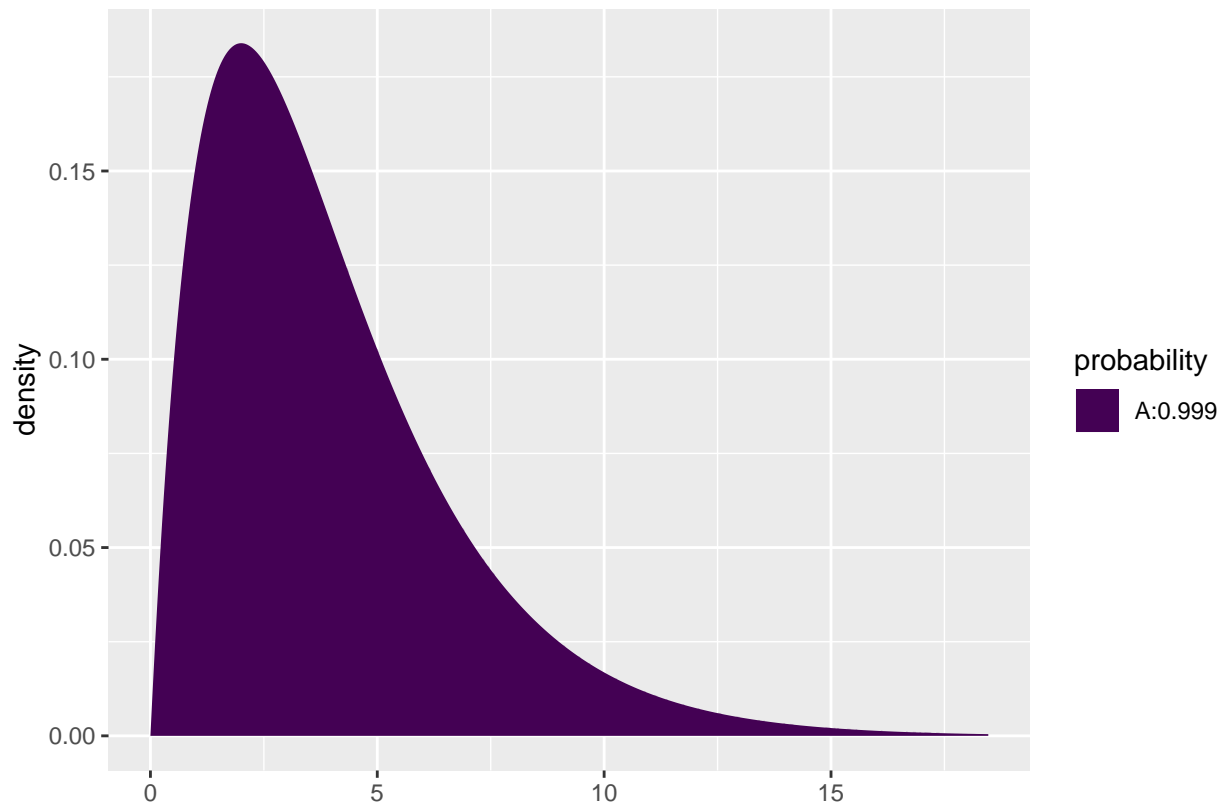
$$X_{obs}^2 = \frac{(57 - 77)^2}{77} + \dots + \frac{(26 - 33.5)^2}{33.5} = 18.8$$

- Is this a large distance??

2.5 χ^2 -test template.

- We want to test the hypothesis H_0 of independence in a table with r rows and c columns:
 - We take a sample and calculate X_{obs}^2 - the observed value of the test statistic.
 - p-value: Assume H_0 is true. What is then the chance of obtaining a larger X^2 than X_{obs}^2 , if we repeat the experiment?
- This can be approximated by the χ^2 -**distribution** with $df = (r - 1)(c - 1)$ degrees of freedom.
- For `Goals` and `Urban.Rural` we have $r = c = 3$, i.e. $df = 4$ and $X_{obs}^2 = 18.8$, so the p-value is:

```
1 - pdist("chisq", 18.8, df = 4)
```



```
## [1] 0.0008603303
```

- There is clearly a significant association between `Goals` and `Urban.Rural`.

2.6 The function `chisq.test`.

- All of the above calculations can be obtained by the function `chisq.test`.

```
tab <- tally(~ Urban.Rural + Goals, data = popKids)
testStat <- chisq.test(tab, correct = FALSE)
testStat
```

```
##
## Pearson's Chi-squared test
##
## data:  tab
## X-squared = 18.828, df = 4, p-value = 0.0008497
```

```
testStat$expected
```

```
##           Goals
## Urban.Rural Grades Popular Sports
## Rural      76.99372 43.95188 28.05439
## Suburban   78.02720 44.54184 28.43096
## Urban      91.97908 52.50628 33.51464
```

-
- The frequency data can also be put directly into a matrix.

```
data <- c(57, 87, 103, 50, 42, 49, 42, 22, 26)
tab <- matrix(data, nrow = 3, ncol = 3)
row.names(tab) <- c("Rural", "Suburban", "Urban")
colnames(tab) <- c("Grades", "Popular", "Sports")
tab
```

```
##           Grades Popular Sports
## Rural      57      50      42
## Suburban   87      42      22
## Urban     103      49      26
```

```
chisq.test(tab)
```

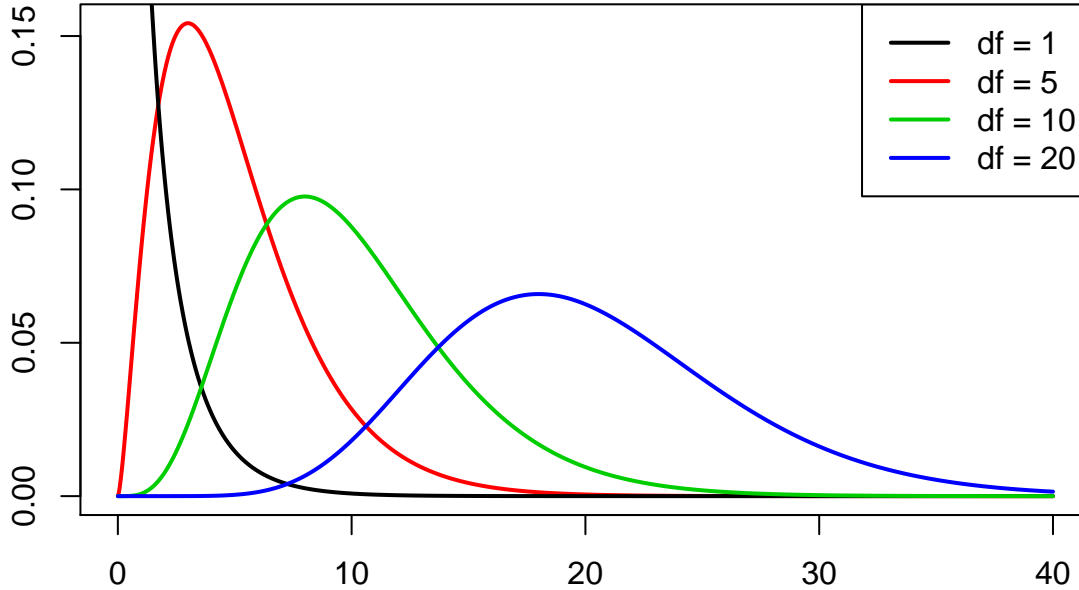
```
##
## Pearson's Chi-squared test
##
## data:  tab
## X-squared = 18.828, df = 4, p-value = 0.0008497
```

3 The χ^2 -distribution

3.1 The χ^2 -distribution

- The χ^2 -distribution with df degrees of freedom:

- Is never negative. And $X^2 = 0$ only happens if $f_e = f_o$.
- Has mean $\mu = df$
- Has standard deviation $\sigma = \sqrt{2df}$
- Is skewed to the right, but approaches a normal distribution when df grows.



4 Agresti - Summary

4.1 Summary

- For the the Chi-squared statistic, X^2 , to be appropriate we require that the expected values have to be $f_e \geq 5$.
- Now we can summarize the ingredients in the Chi-squared test for independence.

TABLE 8.5: The Five Parts of the Chi-Squared Test of Independence

-
1. Assumptions: Two categorical variables, random sampling, $f_e \geq 5$ in all cells
 2. Hypotheses: H_0 : Statistical independence of variables
 H_a : Statistical dependence of variables
 3. Test statistic: $\chi^2 = \sum \frac{(f_o - f_e)^2}{f_e}$, where $f_e = \frac{(\text{Row total})(\text{Column total})}{\text{Total sample size}}$
 4. P -value: P = right-tail probability above observed χ^2 value,
for chi-squared distribution with $df = (r - 1)(c - 1)$
 5. Conclusion: Report P -value
If decision needed, reject H_0 at α -level if $P \leq \alpha$
-

5 Standardized residuals

5.1 Residual analysis

- If we reject the hypothesis of independence it can be of interest to identify the significant deviations.
- In a given cell in the table, $f_o - f_e$ is the deviation between data and the expected values under the null hypothesis.
- We assume that $f_e \geq 5$.
- If H_0 is true, then the standard error of $f_o - f_e$ is given by

$$se = \sqrt{f_e(1 - \text{rowProportion})(1 - \text{columnProportion})}$$

- The corresponding z -score

$$z = \frac{f_o - f_e}{se}$$

should in 95% of the cells be between ± 2 . Values above 3 or below -3 should not appear.

- In popKids table cell **Rural** and **Grade** we got $f_e = 77.0$ and $f_o = 57$. Here $\text{columnProportion} = 51.7\%$ and $\text{rowProportion} = 149/478 = 31.2\%$.
- We can then calculate

$$z = \frac{57 - 77}{\sqrt{77(1 - 0.517)(1 - 0.312)}} = -3.95$$

- Compared to the null hypothesis there are way too few rural kids who find grades important.
- In summary: The standardized residuals allow for cell-by-cell (f_e vs f_o) comparison.

5.2 Residual analysis in R

- In R we can extract the standardized residuals from the output of `chisq.test`:

```
tab <- tally(~ Urban.Rural + Goals, data = popKids)
testStat <- chisq.test(tab, correct = FALSE)
testStat$stdres
```

```
##           Goals
## Urban.Rural  Grades  Popular  Sports
##   Rural    -3.9508449  1.3096235  3.5225004
##   Suburban  1.7666608 -0.5484075 -1.6185210
##   Urban     2.0865780 -0.7274327 -1.8186224
```

6 Models for table data in R

6.1 Example

- We will study the dataset `HairEyeColor`.

```
HairEyeColor <- read.delim("https://asta.math.aau.dk/datasets?file=HairEyeColor.txt")
head(HairEyeColor)
```



```
##   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
```

- Data is organized such that the variable `Freq` gives the frequency of each combination of the factors `Hair`, `Eye` and `Sex`.
- For example: 32 observations are men with black hair and brown eyes.
- We are interested in the association between eye color and hair color ignoring the sex
- We aggregate data, so we have a table with frequencies for each combination of `Hair` and `Eye`.

```
HairEye <- aggregate(Freq ~ Eye + Hair, FUN = sum, data = HairEyeColor)
HairEye
```

```
##      Eye Hair Freq
## 1  Blue Black  20
## 2 Brown Black  68
## 3 Green Black   5
## 4 Hazel Black  15
## 5  Blue Blond  94
## 6 Brown Blond   7
## 7 Green Blond  16
## 8 Hazel Blond  10
## 9  Blue Brown  84
## 10 Brown Brown 119
## 11 Green Brown  29
## 12 Hazel Brown  54
## 13 Blue   Red  17
## 14 Brown  Red  26
## 15 Green  Red  14
## 16 Hazel  Red  14
```

6.2 Model specification

- We can write down a model for (the logarithm of) the expected frequencies by using dummy variables z_{e1}, z_{e2}, z_{e3} and z_{h1}, z_{h2}, z_{h3}
- To denote the different levels of `Eye` and `Hair` (the reference level has all dummy variables equal to 0):

$$\log(f_e) = \alpha + \beta_{e1}z_{e1} + \beta_{e2}z_{e2} + \beta_{e3}z_{e3} + \beta_{h1}z_{h1} + \beta_{h2}z_{h2} + \beta_{h3}z_{h3}.$$

- Note that we haven't included an interaction term, which in this case implies, that we assume independence between `Eye` and `Hair` in the model.
- Since our response variable now is a count it is no longer a linear model (lm) as we have been used to (linear regression).
- Instead it is a so-called generalized linear model and the relevant R command is `glm`.

6.3 Model specification in R

```
model <- glm(Freq ~ Hair + Eye, family = poisson, data = HairEye)
```

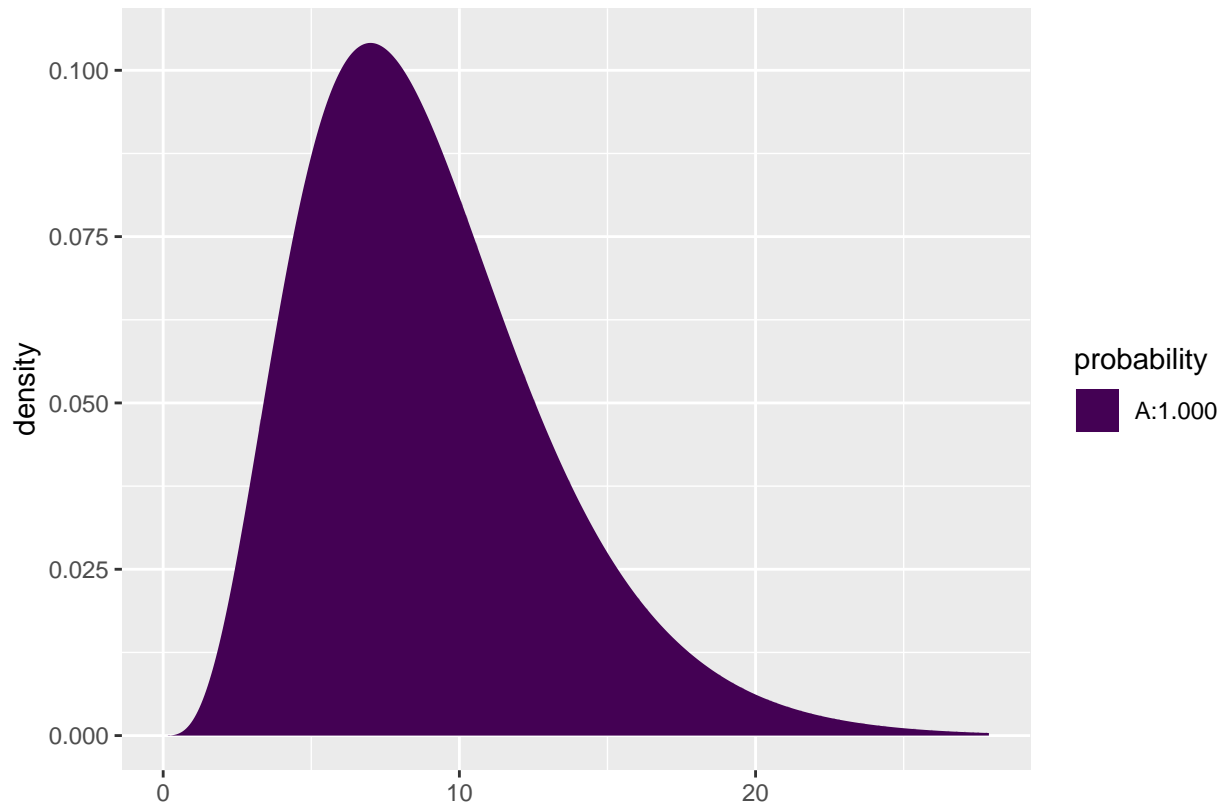
- The argument `family = poisson` ensures that R knows that data should be interpreted as discrete counts and not a continuous variable.

```
summary(model)
```

```
##
## Call:
## glm(formula = Freq ~ Hair + Eye, family = poisson, data = HairEye)
##
## Deviance Residuals:
##   Min       1Q   Median       3Q      Max
## -7.326  -2.065  -0.212   1.235   6.172
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.66926    0.11055  33.191 < 2e-16 ***
## HairBlond    0.16206    0.13089   1.238  0.21569
## HairBrown    0.97386    0.11294   8.623 < 2e-16 ***
## HairRed     -0.41945    0.15279  -2.745  0.00604 **
## EyeBrown     0.02299    0.09590   0.240  0.81054
## EyeGreen    -1.21175    0.14239  -8.510 < 2e-16 ***
## EyeHazel    -0.83804    0.12411  -6.752 1.46e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##   Null deviance: 453.31  on 15  degrees of freedom
## Residual deviance: 146.44  on 9  degrees of freedom
## AIC: 241.04
##
## Number of Fisher Scoring iterations: 5
```

- A value of $X^2 = 146.44$ with $df = 9$ shows that there is very clear significance and we reject the null hypothesis of independence between hair and eye color.

```
1 - pchisq("chisq", 146.44, df = 9)
```



```
## [1] 0
```

6.4 Expected values and standardized residuals

- We also want to look at expected values and standardized (studentized) residuals.
- The null hypothesis predicts $e^{3.67+0.02} = 40.1$ with brown eyes and black hair, but we have observed 68.
- This is significantly too many, since the standardized residual is 5.86.
- The null hypothesis predicts 47.2 with brown eyes and blond hair, but we have seen 7. This is significantly too few, since the standardized residual is -9.42.

```
HairEye$fitted <- fitted(model)
HairEye$resid <- rstudent(model)
HairEye
```

```
##      Eye Hair Freq fitted resid
## 1  Blue Black   20  39.22 -4.492
## 2  Brown Black   68  40.14  5.856
## 3  Green Black    5  11.68 -2.508
## 4  Hazel Black   15  16.97 -0.583
## 5   Blue Blond   94  46.12  9.368
## 6  Brown Blond    7  47.20 -9.423
## 7  Green Blond   16  13.73  0.719
## 8  Hazel Blond   10  19.95 -2.936
## 9   Blue Brown   84 103.87 -3.437
## 10 Brown Brown  119 106.28  2.151
## 11 Green Brown   29  30.92 -0.511
```

```
## 12 Hazel Brown 54 44.93 2.023
## 13 Blue Red 17 25.79 -2.399
## 14 Brown Red 26 26.39 -0.101
## 15 Green Red 14 7.68 2.368
## 16 Hazel Red 14 11.15 0.961
```

7 Log-linear models

7.1 Model specification

- We shall consider the data set `living`, which is a Danish survey on standard of living. The variables are
 - B (Housing, Bolig): bad/acceptable/good
 - H (Health, Helbred): bad/good
 - I (Isolated, Isoleret): yes/no
 - A (Anxiety, Angst): yes/no
- N: The number of respondents for each combination of the 4 factors above
- We want to order the factors such that the reference is “negative”.

```
living <- read.delim("https://asta.math.aau.dk/datasets?file=living.txt", stringsAsFactors = TRUE)
```

```
living$B <- relevel(living$B, "Bad")
living$I <- relevel(living$I, "Yes")
living$A <- relevel(living$A, "Yes")
```

7.2 Example

- At first we study interaction between housing and health. So we aggregate data and only look at the association between B and H without controlling for I and A:

```
BH <- aggregate(N ~ B + H, FUN = sum, data = living)
BH
```

```
##           B     H     N
## 1         Bad  Bad  211
## 2 Acceptable  Bad  327
## 3         Good  Bad 1734
## 4         Bad  Good  145
## 5 Acceptable  Good  211
## 6         Good  Good 1855
```

7.3 Model specification

- Like last time we can write down a model for (the logarithm of) the expected frequencies f_e by using dummy variables.
- We let z_{b1} , z_{b2} and z_{h1} denote the different levels of B and H (the reference level has all dummy variables equal to 0):

$$\log(f_e) = \alpha + \beta_{b1}z_{b1} + \beta_{b2}z_{b2} + \beta_{h1}z_{h1} + \beta_{b1h1}z_{b1}z_{h1} + \beta_{b2h1}z_{b2}z_{h1}.$$

- Note that this time we have included an interaction term, which in this case implies, that we do not assume independence between B and H in the model.
- This model contains all possible terms and there are as many parameters(6) as there are cells(6) in the table. This is called the **saturated** model.

7.4 Example

- We fit the model using `glm`:

```
model <- glm(N ~ B * H, family = poisson, data = BH)
```

- The parameter estimates (of the contrasts, i.e. differences to the reference level (B: Bad, H: Bad) are

```
summary(model)
```

```
##
## Call:
## glm(formula = N ~ B * H, family = poisson, data = BH)
##
## Deviance Residuals:
## [1] 0 0 0 0 0 0
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      5.3519     0.0688   77.74 < 2e-16 ***
## BAcceptable       0.4381     0.0883    4.96 7.0e-07 ***
## BGood             2.1063     0.0729   28.89 < 2e-16 ***
## HGood            -0.3751     0.1079   -3.48 0.00051 ***
## BAcceptable:HGood -0.0630     0.1394   -0.45 0.65144
## BGood:HGood       0.4426     0.1129    3.92 8.9e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##    Null deviance:  4.2103e+03  on 5  degrees of freedom
## Residual deviance: -2.9532e-14  on 0  degrees of freedom
## AIC: 59.48
##
## Number of Fisher Scoring iterations: 2
```

- The combination Good:Good increases the response, i.e. an over representation of people with good housing conditions and good health.

7.5 Model form

- Log-linear models quickly become cumbersome to write down.
- For log-linear models the structure of the model is often more interesting than the value of the parameters.

- Therefore we typically just use the model form

$$B + H + B : H$$

where $B + H$ are the main effects and $B : H$ means that we have interaction between B and H .

- We shall stick to the **hierarchical principle**, which means that if $B : H$ is included then we must include $B + H$. For that reason we use the shorthand notation

$$B * H = B + H + B : H$$

7.6 Model comparison

- If we suggest a simpler model than the saturated model it will always provide a poorer fit to the given data.
- We need to find a model which is as simple (few parameters) as possible but which at the same time fits the data well.
- Last time we used the χ^2 statistic to judge whether the model $B + H$ (independence between B and H) was good enough compared to the saturated model $B * H$. This is only possible for models with two variables.
- More generally we shall consider the **Deviance of a model**, which is - approximately - given by

$$\sum \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$$

7.7 Deviance

- We look at the output from the function `drop1`:

```
drop1(model, test = "Chisq")
```

```
## Single term deletions
##
## Model:
## N ~ B * H
##           Df Deviance   AIC   LRT Pr(>Chi)
## <none>         0.0 59.5
## B:H           2    40.8 96.3 40.8  1.4e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The deviance for the saturated model (`<none>`) is zero as observed=expected.
- Deviance is increased by 40.8, when we remove the interaction $B:H$.
- Removing $B:H$ corresponds to the null hypothesis $H_0 : \beta_{b_1h_1} = \beta_{b_2h_1} = 0$.
- The deviance is compared with a $\chi^2(Df = 2)$ distribution to determine whether it is significant.
- Since the p-value is 1.4×10^{-9} the interaction is significant and cannot be left out.

8 Higher order interaction

8.1 Higher order interaction

- We shall consider models with interaction between more than two variables.

- E.g. there may be a combined effect of adding water, fertilizer and light to a plant at the same time, i.e. the effect of adding water and fertilizer depends on whether the light is on.
- We call this a **3-way interaction**.
- We shall still respect the hierarchical principle:
 - If we include a 3-way interaction, then we must include all main effects and 2-way interactions of the 3 variables.
- Again we use short hand notation

$$B * H * A = B + H + A + B : H + B : A + H : A + B : H : A$$

- Similar considerations hold for 4-way interactions like $B * H * A * I$, etc.

8.2 Bigger example

- We fit the saturated model to the full dataset and look at the estimated parameters (only the last half of the values are printed to save space):

```
satmodel <- glm(N ~ B * H * A * I, family = poisson, data = living)
summary(satmodel)
```

```
##
## Call:
## glm(formula = N ~ B * H * A * I, family = poisson, data = living)
##
## Deviance Residuals:
## [1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [24]  0
##
## Coefficients:
##
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.08e+00   3.54e-01   5.88 4.1e-09 ***
## BAcceptable    5.60e-01   4.43e-01   1.26 0.20671
## BGood          1.42e+00   3.94e-01   3.60 0.00032 ***
## HGood         -2.44e+01   4.22e+04   0.00 0.99954
## ANo           4.86e-01   4.49e-01   1.08 0.27994
## INo           1.75e+00   3.83e-01   4.57 5.0e-06 ***
## BAcceptable:HGood  2.28e+01   4.22e+04   0.00 0.99957
## BGood:HGood      2.27e+01   4.22e+04   0.00 0.99957
## BAcceptable:ANo   5.35e-02   5.61e-01   0.10 0.92408
## BGood:ANo        -3.08e-02   5.01e-01  -0.06 0.95107
## HGood:ANo        2.34e+01   4.22e+04   0.00 0.99956
## BAcceptable:INo  6.19e-03   4.80e-01   0.01 0.98971
## BGood:INo        5.47e-01   4.24e-01   1.29 0.19716
## HGood:INo        2.40e+01   4.22e+04   0.00 0.99955
## ANo:INo          6.56e-01   4.80e-01   1.37 0.17214
## BAcceptable:HGood:ANo -2.35e+01   4.22e+04   0.00 0.99956
## BGood:HGood:ANo   -2.25e+01   4.22e+04   0.00 0.99957
## BAcceptable:HGood:INo -2.30e+01   4.22e+04   0.00 0.99957
## BGood:HGood:INo   -2.27e+01   4.22e+04   0.00 0.99957
## BAcceptable:ANo:INo -2.52e-01   6.01e-01  -0.42 0.67537
## BGood:ANo:INo     2.83e-01   5.33e-01   0.53 0.59571
## HGood:ANo:INo    -2.34e+01   4.22e+04   0.00 0.99956
```

```
## BAcceptable:HGood:ANo:INo  2.36e+01  4.22e+04  0.00  0.99955
## BGood:HGood:ANo:INo      2.30e+01  4.22e+04  0.00  0.99957
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 1.0965e+04 on 23 degrees of freedom
## Residual deviance: 4.1199e-10 on 0 degrees of freedom
## AIC: 179.1
##
## Number of Fisher Scoring iterations: 20
```

- We see that the four-way interaction and all three-way interactions look like they could be left out.
- However, the p-values are related to removing one and leaving everything else in the model so we have to remove terms of the model one at a time and check against the new model.
- This can be done by successive use of `drop1` and `update` as explained in the following.

```
drop1(satmodel, test = "Chi")
```

```
## Single term deletions
##
## Model:
## N ~ B * H * A * I
##           Df Deviance AIC  LRT Pr(>Chi)
## <none>           0.00 179
## B:H:A:I  2       3.51 179 3.51    0.17
```

- The output of `drop1` reveals that the four-way interaction is insignificant and we remove it and save the updated model like this:

```
reducedmodel <- update(satmodel, .~-B:H:A:I)
```

- We use `drop1` again:

```
drop1(reducedmodel, test = "Chi")
```

```
## Single term deletions
##
## Model:
## N ~ B + H + A + I + B:H + B:A + H:A + B:I + H:I + A:I + B:H:A +
##       B:H:I + B:A:I + H:A:I
##           Df Deviance AIC  LRT Pr(>Chi)
## <none>           3.51 179
## B:H:A  2       7.14 178 3.63    0.16
## B:H:I  2       3.53 175 0.02    0.99
## B:A:I  2       5.53 177 2.02    0.36
## H:A:I  1       4.59 178 1.07    0.30
```


- We see B:H:I could be removed and use `update` again:

```
reducedmodel <- update(reducedmodel, .~-B:H:I)
```

- We can continue this way as long as we like.
- If instead we have a specific simple model in mind, we can define this model and test it against the saturated model with `anova` as shown in the following.

8.3 Test of simple model against the saturated model

- We fit the simpler model to the full dataset:

```
simplemodel <- glm(N ~ B*H + B*A + B*I + H*A + H*I + I*A, family = poisson, data = living)
```

- We compare the two models using `anova`:

```
anova(simplemodel, satmodel, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: N ~ B * H + B * A + B * I + H * A + H * I + I * A
## Model 2: N ~ B * H * A * I
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         9      10.7
## 2         0         0.0  9    10.7    0.3
```

- The deviance between the two models is 10.697 with $df = 9$, which has a p-value of 29.7%. So we prefer the simpler model without four- and 3-way interactions.

8.4 Model reduction

- Let us check whether we can make further model reductions, where we again use the function `drop1`.

```
drop1(simplemodel, test = "Chisq")
```

```
## Single term deletions
##
## Model:
## N ~ B * H + B * A + B * I + H * A + H * I + I * A
##           Df Deviance AIC  LRT Pr(>Chi)
## <none>           10.7 172
## B:H           2    38.2 195 27.5  1.1e-06 ***
## B:A           2    37.9 195 27.2  1.2e-06 ***
## B:I           2    35.2 192 24.5  4.7e-06 ***
## H:A           1    41.9 201 31.3  2.3e-08 ***
## H:I           1    56.0 215 45.4  1.6e-11 ***
## A:I           1    26.4 186 15.8  7.2e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Conclusion: All of the pairwise interaction terms are significant.

8.5 Final model and parameter estimates

- In conclusion our final model is `simplemodel`.

```
summary(simplemodel)
```

```
##
## Call:
## glm(formula = N ~ B * H + B * A + B * I + H * A + H * I + I *
##       A, family = poisson, data = living)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.7295  -0.4842  -0.0025   0.3166   1.2319
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      2.1704    0.2296   9.45 < 2e-16 ***
## BAcceptable      0.6532    0.2632   2.48  0.01308 *
## BGood            1.1378    0.2336   4.87  1.1e-06 ***
## HGood           -1.7678    0.2103  -8.41 < 2e-16 ***
## ANo              0.3507    0.1925   1.82  0.06849 .
## INo              1.7480    0.2312   7.56  4.0e-14 ***
## BAcceptable:HGood -0.0412    0.1412  -0.29  0.77037
## BGood:HGood       0.3777    0.1144   3.30  0.00096 ***
## BAcceptable:ANo  -0.1276    0.1583  -0.81  0.42035
## BGood:ANo         0.3941    0.1326   2.97  0.00297 **
## BAcceptable:INo  -0.1402    0.2587  -0.54  0.58788
## BGood:INo         0.7204    0.2280   3.16  0.00158 **
## HGood:ANo         0.4408    0.0794   5.55  2.8e-08 ***
## HGood:INo         1.1235    0.1798   6.25  4.2e-10 ***
## ANo:INo           0.6688    0.1628   4.11  4.0e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 10964.662  on 23  degrees of freedom
## Residual deviance:   10.697  on  9  degrees of freedom
## AIC: 171.8
##
## Number of Fisher Scoring iterations: 4
```

- We see that all significant interactions are positive - as expected(why?).
- (Intercept) is log of the expected number, when all factors are “bad”. So the expected number is $\exp(2.17) = 8.76$, whereas the observed number is 8.

8.6 Predicted values

- What is the expected number of people without anxiety (A), with acceptable housing (B), good health (H) that are isolated (I)?

```
summary(simplemodel)
```

```
##
## Call:
## glm(formula = N ~ B * H + B * A + B * I + H * A + H * I + I *
##      A, family = poisson, data = living)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.7295  -0.4842  -0.0025   0.3166   1.2319
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      2.1704     0.2296   9.45 < 2e-16 ***
## BAcceptable      0.6532     0.2632   2.48 0.01308 *
## BGood            1.1378     0.2336   4.87 1.1e-06 ***
## HGood           -1.7678     0.2103  -8.41 < 2e-16 ***
## ANo              0.3507     0.1925   1.82 0.06849 .
## INo              1.7480     0.2312   7.56 4.0e-14 ***
## BAcceptable:HGood -0.0412     0.1412  -0.29 0.77037
## BGood:HGood      0.3777     0.1144   3.30 0.00096 ***
## BAcceptable:ANo  -0.1276     0.1583  -0.81 0.42035
## BGood:ANo        0.3941     0.1326   2.97 0.00297 **
## BAcceptable:INo -0.1402     0.2587  -0.54 0.58788
## BGood:INo        0.7204     0.2280   3.16 0.00158 **
## HGood:ANo        0.4408     0.0794   5.55 2.8e-08 ***
## HGood:INo        1.1235     0.1798   6.25 4.2e-10 ***
## ANo:INo          0.6688     0.1628   4.11 4.0e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 10964.662  on 23  degrees of freedom
## Residual deviance:   10.697  on  9  degrees of freedom
## AIC: 171.8
##
## Number of Fisher Scoring iterations: 4
```

- Logarithm of the expected:

$$2.17042 + 0.65323 - 1.76785 + 0.35067 - 0.04121 - 0.12759 + 0.44076 = 1.67843$$

- so the expected number is $\exp(1.67843) = 5.36$, whereas the observed number is 5.

-
- We can of course make **R** calculate all the table values expected by the model and add them to the data:

```
living$expected <- fitted(simplemodel)
living[1:12,]
```

##	B	H	I	A	N	expected
## 1	Bad	Good	Yes	No	5	3.30
## 2	Bad	Good	No	No	107	113.78
## 3	Bad	Good	Yes	Yes	0	1.50
## 4	Bad	Good	No	Yes	33	26.42
## 5	Bad	Bad	Yes	No	13	12.44
## 6	Bad	Bad	No	No	144	139.47
## 7	Bad	Bad	Yes	Yes	8	8.76
## 8	Bad	Bad	No	Yes	46	50.32
## 9	Acceptable	Good	Yes	No	5	5.36
## 10	Acceptable	Good	No	No	155	160.54
## 11	Acceptable	Good	Yes	Yes	3	2.76
## 12	Acceptable	Good	No	Yes	48	42.35

9 Graphical representation

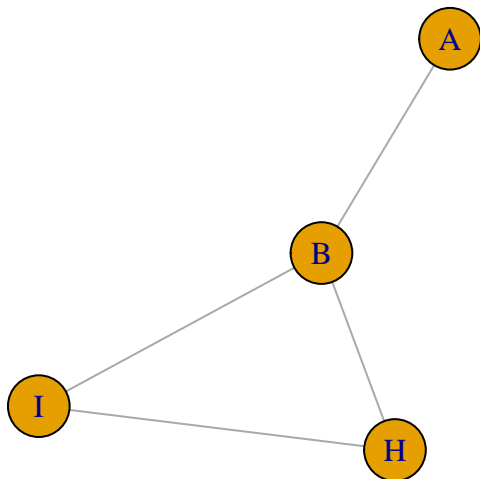
9.1 Graphical representation

- We make a graphical representation by
 - drawing a circle for each variable.
 - connecting variables which enter the same model term.

- Example: Assume the model is

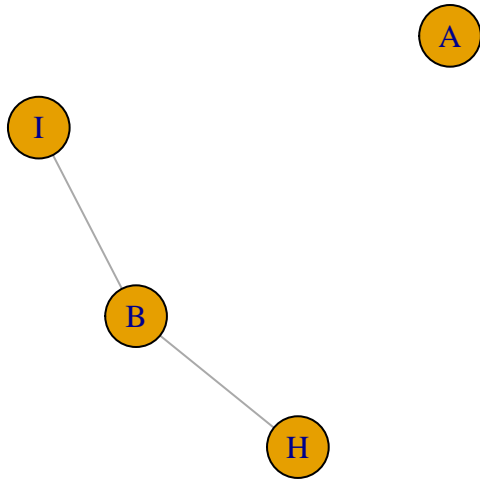
$$A * B + B * H * I$$

- Then the graphical representation is

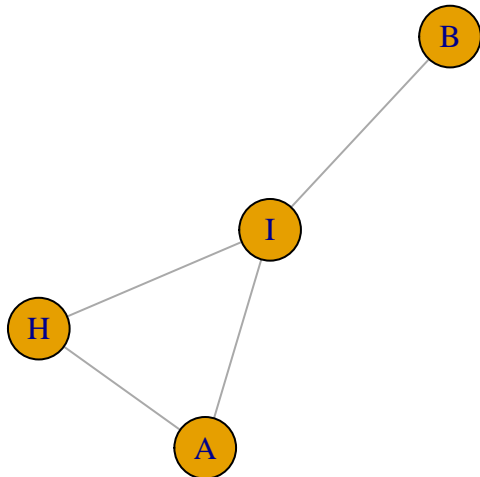


9.2 Interpretation of graphical model

- **Independence:** If A enters in the model formula, but A doesn't enter in any other terms (e.g. $A * B$, $A * H$, etc.), then A is independent of the other variables.
- E.g. $A + B * H + B * I$

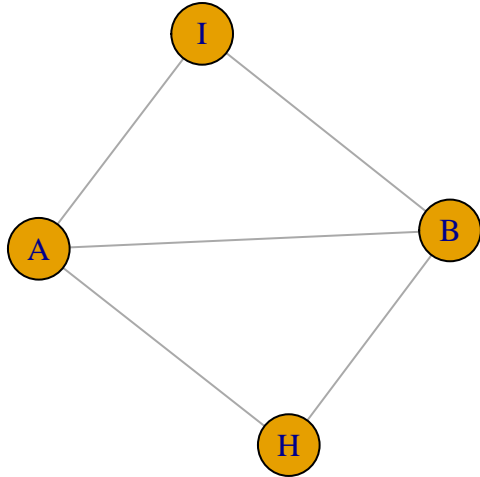


- **Explained association.** If B and H are "connected" via other terms, but don't enter the same term, then the association is explained by other variables. I.e. the model cannot include e.g. $B * H$, $B * H * A$ or $A * B * H * I$.
- E.g. $B * I + A * H * I$

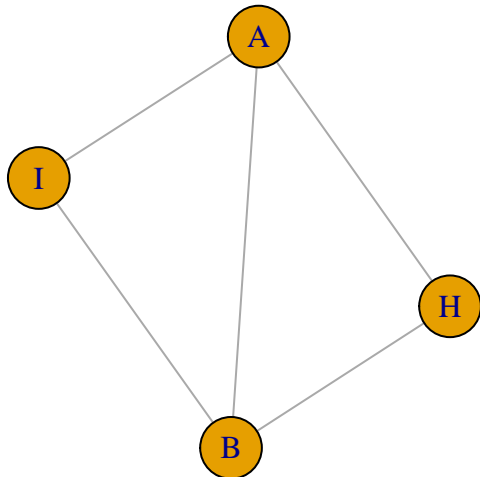


9.3 Interpretation of graphical model

- **Homogeneous association:** If $A * H$ enters the model, but $A * H$ doesn't enter more complicated terms, then the association between A and H is homogeneous.
- I.e. the model cannot contain $A * H * I$, $A * B * H$ or $A * B * H * I$.
- E.g. $A * H + A * I * B + B * H$



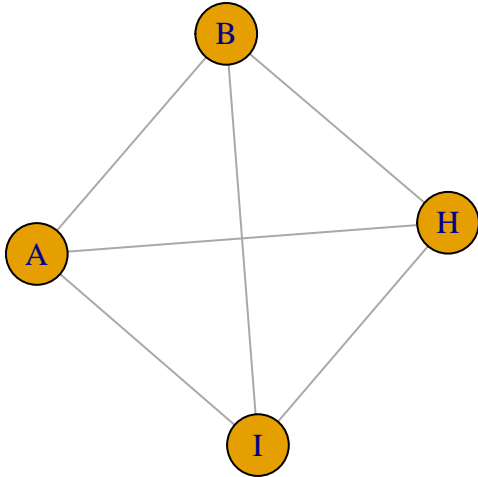
- **Heterogeneous association:** If $A * H$ enter the model as a part of a more complicated term, then the association between A and H is heterogeneous.
- I.e. the model *must* contain $A * H * I$, $A * B * H$ or $A * B * H * I$.
- E.g. $A * B * H + A * I * B$



9.4 Final model of the example

- In the example the final model was:

$$B * I + H * I + I * A + B * H + B * A + H * A$$



- We can directly see from the graph, that:
 - we don't have any independent variables since all variables are connected
 - we do not have an explained association.
- From the formula we have homogeneous association between all pairs of variables.