Biostatistics (MATH11230)

Vanda Inácio

In this document I reproduce the results presented in the slides and illustrate some further calculations. We start wuith the pancreatic cancer example of slide 2 (and onwards). The data are available in the file `coffeedata_2.xls`.

```
require(readxl)
data_coffee_2 <- read_excel("coffeedata_2.xls")
data_coffee_2
```

```
## # A tibble: 8 x 4
##   cases controls coffee   sex
##   <dbl>    <dbl>  <dbl> <dbl>
## 1    60       82      3     0
## 2    53       74      2     0
## 3    94      119      1     0
## 4     9       32      0     0
## 5    28       48      3     1
## 6    53       80      2     1
## 7    59      152      1     1
## 8    11       56      0     1
```

Note that in the coffee column, 0 denotes no coffee consumption, 1 denotes $1-2$ cups of coffee per day, 2 denotes $3-4$ cups of coffee per day and 3 denotes 5 or more cups coffee/day. In turn, in the column sex, 1 stands for a female subject and 0 for a male. I will start by coding these variables as factors and relabelling them so that they a have a more intuitive meaning (at least, to me!).

```
data_coffee_2$sex <- factor(data_coffee_2$sex, levels = c(0, 1),
                            labels = c("Male", "Female"))
data_coffee_2$coffee <- factor(data_coffee_2$coffee, levels = c(0, 1, 2, 3),
                               labels = c("0", "1-2", "3-4", "5+"))
data_coffee_2
```

```
## # A tibble: 8 x 4
##   cases controls coffee sex
##   <dbl>    <dbl> <fct>  <fct>
## 1    60       82 5+     Male
## 2    53       74 3-4    Male
## 3    94      119 1-2    Male
## 4     9       32 0      Male
## 5    28       48 5+     Female
## 6    53       80 3-4    Female
## 7    59      152 1-2    Female
## 8    11       56 0      Female
```

Note that the data are grouped (or in *binomial* format), i.e., for each coffee consumption and gender levels combination, it is listed the number of cases and the number of controls (or, more generally, the number of successes and failures). This a popular way of presenting the data when all exposure variables are discrete.

We then need to pass the number of cases and controls for each exposure variables combination to the `glm` function.

```
res_binom <- glm(cbind(cases, controls) ~ coffee + sex, family = "binomial",
                 data = data_coffee_2)
summary(res_binom)
```

```
##
## Call:
## glm(formula = cbind(cases, controls) ~ coffee + sex, family = "binomial",
##     data = data_coffee_2)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.2434     0.2597  -4.788 1.68e-06 ***
## coffee1-2     0.8668     0.2687   3.226 0.001256 **
## coffee3-4     1.0726     0.2791   3.843 0.000122 ***
## coffee5+      0.9900     0.2862   3.459 0.000543 ***
## sexFemale    -0.4035     0.1347  -2.996 0.002733 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 33.469  on 7  degrees of freedom
## Residual deviance:  4.268  on 3  degrees of freedom
## AIC: 54.219
##
## Number of Fisher Scoring iterations: 3
```

```
exp(res_binom$coefficient)[2:5]
```

```
## coffee1-2 coffee3-4  coffee5+ sexFemale
## 2.3793677 2.9228382 2.6911641 0.6679681
```

```
exp(confint.default(res_binom, level = 0.95))[2:5,]
```

```
##               2.5 %    97.5 %
## coffee1-2 1.4051677 4.0289788
## coffee3-4 1.6913090 5.0511070
## coffee5+  1.5356496 4.7161569
## sexFemale 0.5130039 0.8697426
```

You may see as well in the literature, the following use of the `glm` function with grouped/binomial data: instead of using the pairs of cases and controls, one passes to the function the proportion of cases and in this case the argument `weights` need to be specified as well (corresponding to the total of observations per category).

```
res_binom_alt <- glm(cases/(cases + controls) ~ coffee + sex, family = "binomial",
                     weights = cases + controls, data = data_coffee_2)
summary(res_binom_alt)
```

```
##
## Call:
## glm(formula = cases/(cases + controls) ~ coffee + sex, family = "binomial",
##     data = data_coffee_2, weights = cases + controls)
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.2434     0.2597  -4.788 1.68e-06 ***
## coffee1-2     0.8668     0.2687   3.226 0.001256 **
## coffee3-4     1.0726     0.2791   3.843 0.000122 ***
## coffee5+      0.9900     0.2862   3.459 0.000543 ***
## sexFemale    -0.4035     0.1347  -2.996 0.002733 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 33.469  on 7  degrees of freedom
## Residual deviance:  4.268  on 3  degrees of freedom
## AIC: 54.219
##
## Number of Fisher Scoring iterations: 3
```

All output is obviously the same. Alternatively, we can rearrange the data in an ungrouped (or bernoulli) form and each individual is listed separately (i.e., observations that form say, an exposure class, with the same gender and coffee consumption, are not grouped). The data is stored in this format in the file `coffeedata_1.xls`.

```
data_coffee_1 <- read_excel("coffeedata_1.xls")
data_coffee_1$sex <- factor(data_coffee_1$sex, levels = c(0, 1),
                            labels = c("Male", "Female"))
data_coffee_1$coffee <- factor(data_coffee_1$coffee, levels = c(0, 1, 2, 3),
                               labels = c("0", "1-2", "3-4", "5+"))
head(data_coffee_1)
```

```
## # A tibble: 6 x 3
##   cancer_status coffee sex
##           <dbl> <fct>  <fct>
## 1             0 5+     Male
## 2             0 5+     Male
## 3             0 5+     Male
## 4             0 5+     Male
## 5             0 5+     Male
## 6             0 5+     Male
```

We can now use the function `glm` just passing the response variable, `cancer_status` in this case, which is either a 1 (for a case) or a 0 (for a control).

```
res_bern <- glm(cancer_status ~ coffee + sex,
               family = "binomial", data = data_coffee_1)
summary(res_bern)
```

```
##
## Call:
## glm(formula = cancer_status ~ coffee + sex, family = "binomial",
##     data = data_coffee_1)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.2434     0.2597  -4.788 1.68e-06 ***
## coffee1-2     0.8668     0.2687   3.226 0.001256 **
```

```
## coffee3-4     1.0726     0.2791   3.843 0.000122 ***
## coffee5+      0.9900     0.2862   3.459 0.000543 ***
## sexFemale    -0.4035     0.1347  -2.996 0.002733 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1323.8  on 1009  degrees of freedom
## Residual deviance: 1294.6  on 1005  degrees of freedom
## AIC: 1304.6
##
## Number of Fisher Scoring iterations: 4
```

We see that all results but the deviances and AIC are the same. Still, the difference between the null and the residual deviances are the same under the two models. With respect

```r
AIC(res_binom)
```

```
## [1] 54.21942
```

```r
AIC(res_bern)
```

```
## [1] 1304.567
```

The AIC as we will see later in the lecture slides is given by $-2 \log \text{likelihood}(\widehat{\boldsymbol{\beta}}) + (k+1)$, where $k+1$ is the total number of parameters in the model (k regression coefficients and the intercept). The difference between the AIC comes from the difference between the bernoulli and binomial likelihoods. In particular, the AIC for the binomial case is is just equal to the AIC of the bernoulli model plus the following term: $-2\sum_{k=1}^{8} \log \binom{n_k}{y_k}$, where n_k is the number of subjects (cases + controls) in category (as formed by the gender and coffee consumption levels combination) k, whereas y_k is the number of cases (at category k).

```r
AIC(res_bern) -2*(log(choose(60 + 82, 60)) + log(choose(53 + 74, 53)) +
                  log(choose(94 + 119, 94)) + log(choose(9 + 32, 9)) +
                  log(choose(28 + 48, 28)) + log(choose(53 + 80, 53)) +
                  log(choose(59 + 152, 59)) + log(choose(11 + 56, 11)))
```

```
## [1] 54.21942
```

Let us now illustrate the likelihood ratio method using the CHD example we have analysed before. Just for the sake of illustration, we will be discretizing the weight variable in five categories.

```r
data_wchs <- read_excel("wcgsdata.xls")
names(data_wchs)
```

```
##  [1] "Id"      "Age0"    "Height0" "Weight0" "Sbp0"    "Dbp0"    "Chol0"
##  [8] "Behpat0" "Ncigs0"  "Dibpat0" "Chd69"   "Typechd" "Time169" "Arcus0"
```

```r
head(data_wchs)
```

```
## # A tibble: 6 x 14
##      Id  Age0 Height0 Weight0  Sbp0  Dbp0 Chol0 Behpat0 Ncigs0 Dibpat0 Chd69
##   <dbl> <dbl>   <dbl>   <dbl> <dbl> <dbl> <chr>   <dbl>  <dbl>   <dbl> <dbl>
## 1  2001    49      73     150   110    76 225         2     25       1     0
## 2  2002    42      70     160   154    84 177         2     20       1     0
## 3  2003    42      69     160   110    78 181         3      0       0     0
## 4  2004    41      68     152   124    78 132         4     20       0     0
## 5  2005    59      70     150   144    86 255         3     20       0     1
## 6  2006    44      72     204   150    90 182         4      0       0     0
```

```
## # i 3 more variables: Typechd <dbl>, Time169 <dbl>, Arcus0 <chr>
```

```r
require(readxl)
data_wchs <- read_excel("wcgsdata.xls")
names(data_wchs)
```

```
##  [1] "Id"      "Age0"    "Height0" "Weight0" "Sbp0"    "Dbp0"    "Chol0"
##  [8] "Behpat0" "Ncigs0"  "Dibpat0" "Chd69"   "Typechd" "Time169" "Arcus0"
```

```r
head(data_wchs)
```

```
## # A tibble: 6 x 14
##      Id  Age0 Height0 Weight0  Sbp0  Dbp0 Chol0 Behpat0 Ncigs0 Dibpat0 Chd69
##   <dbl> <dbl>   <dbl>   <dbl> <dbl> <dbl> <chr>   <dbl>  <dbl>   <dbl> <dbl>
## 1  2001    49      73     150   110    76 225         2     25       1     0
## 2  2002    42      70     160   154    84 177         2     20       1     0
## 3  2003    42      69     160   110    78 181         3      0       0     0
## 4  2004    41      68     152   124    78 132         4     20       0     0
## 5  2005    59      70     150   144    86 255         3     20       0     1
## 6  2006    44      72     204   150    90 182         4      0       0     0
## # i 3 more variables: Typechd <dbl>, Time169 <dbl>, Arcus0 <chr>
```

```r
n <- nrow(data_wchs)
data_wchs$weight_cat <- numeric(n)
for(i in 1:n){
data_wchs$weight_cat[i] <- ifelse(data_wchs$Weight0[i] <= 150, 1,
       ifelse(data_wchs$Weight0[i] > 150 & data_wchs$Weight0[i] <= 160, 2,
              ifelse(data_wchs$Weight0[i] > 160 & data_wchs$Weight0[i] <= 170, 3,
                     ifelse(data_wchs$Weight0[i] > 170 & data_wchs$Weight0[i] <= 180, 4, 5))))
}

data_wchs$weight_cat <- factor(data_wchs$weight_cat, levels = c(1, 2, 3, 4, 5),
                               labels = c("<150", "150-160", "160-170",
                                          "170-180", ">180"))

res_weight_cat <- glm(Chd69 ~ weight_cat, family = "binomial",
                      data = data_wchs)
summary(res_weight_cat)
```

```
##
## Call:
## glm(formula = Chd69 ~ weight_cat, family = "binomial", data = data_wchs)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -2.85862    0.18177 -15.727  < 2e-16 ***
## weight_cat150-160  0.06805    0.25938   0.262 0.793041
## weight_cat160-170  0.38377    0.23393   1.641 0.100899
## weight_cat170-180  0.83167    0.22403   3.712 0.000205 ***
## weight_cat>180     0.61003    0.21729   2.807 0.004993 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1781.2  on 3153  degrees of freedom
```

```
## Residual deviance: 1759.8  on 3149  degrees of freedom
## AIC: 1769.8
##
## Number of Fisher Scoring iterations: 5
```

We now test the hypothesis $H_0: \beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$. From the slides, we know that all we need is the deviance from the model only containing β_0 and the deviance from the model containing all five parameters. The deviance of the model only containing the intercept is always given in the output of the `glm` function in the null deviance. The residual deviance is the deviance of the model we have fitted (in this case, the model containing the five parameters).

```
dif_deviance <- res_weight_cat$null.deviance - res_weight_cat$deviance
dif_deviance
```

```
## [1] 21.39813
```

The values of the χ^2_1 distribution can then be used to determine the probability of observing a value as large or larger than this difference of deviances, assuming the null hypothesis H_0 to be true. This probability is known as the p-value, associated with the null hypothesis H_0, generated by the observed data. As mentioned, the p-value is the right hand tail area of the χ^2_1 distribution, greater than the observed value of the test statistic.

```
pchisq(dif_deviance, df = 4, lower = FALSE)
```

```
## [1] 0.0002640013
```

```
1 - pchisq(dif_deviance, df = 4, lower = TRUE)
```

```
## [1] 0.0002640013
```

At any significance level commonly used (e.g., 0.01, 0.05, 0.1) we reject the null hypothesis, i.e., we reject that all four coefficients are zero.

The Wald test statistic, z_{β_j} as in the slides, is available in the `z value` column of the output. The corresponding p-value is given in the next (to the right) column. We do not need but we know how to obtain those p-values. For instance, for β_1

```
pchisq(0.262^2, df = 1, lower = FALSE)
```

```
## [1] 0.7933214
```

The AIC is provided as part of the output of the `glm` function. There are also the functions `AIC` and `BIC`.

```
#Below 5 is the number of parameters.
BIC(res_weight_cat)
```

```
## [1] 1800.128
```

```
res_weight_cat$deviance + 5*log(dim(data_wchs)[1])
```

```
## [1] 1800.128
```

```
AIC(res_weight_cat)
```

```
## [1] 1769.846
```

```
res_weight_cat$deviance + 5*2
```

```
## [1] 1769.846
```