Exercise 3: Multivariable analysis in R part 1: Logistic regression

At the end of this exercise you should be able to:

a. Know how to use logistic regression in R
b. Know how to properly remove factors for which most likely adjustment is not required

In Exercise 1, you learned some basic principles about the R language. As you go along, it will prove worthwhile to familiarize yourself more and more with the workings of the R / S language, and we provided you with links to some particularly useful texts that help in becoming more proficient.

In Exercise 2, you learned how to import a dataset, to assign the special values for missing data that R recognizes as such, to create data subsets for analysis. Next you learned to write your own basic function.

For none of the things you learned in Exercises 1 and 2 will we have a need for R. EpiData delivers all that, and in a very simple and intuitive way. We will take recourse to R only if we cannot solve a problem analytically with EpiData Analysis. One such application is the logistic regression analysis which is the subject of this exercise.

Before we get started with the actual work, open a new script page and save it as "`e_ex03.r`". We will use the dataset `e_ex02_02.dat` as our starting point, that is, the set with 501 cases with known fluoroquinolone drug susceptibility test result. The simplest way to make a dataset available in R is with the command `read.table`:

```
e_ex03 <- read.table("e_ex02_02.dat")
attach(e_ex03)
```

We assign its content to an object `e_ex03` which we attach, so that it is available in the path.

The indication for a logistic regression

A major challenge in the analysis of epidemiologic data is to avoid falsely inferring causality between some factor and an outcome. As a simplified example we might find in a given population that cases of lung cancer occur far more frequently among males than among females. One would not conclude that being male is a risk factor for lung cancer before first examining whether smoking might also be a virtually male addiction in that population. For categorical variables we might use the Mantel-Haenszel stratification approach to adjust for such confounding. The method has two major limitations. First, the more variables we need to adjust for, the larger the uncertainty of our estimates and the more likely some strata are left without counts and making a summary measure becomes impossible. Second, the method is limited to categorical variables but often we do not wish to categorize a naturally continuous variable (like age) just to accommodate the method. From various possibilities, one favored method is logistic regression analysis that overcomes these two major limitations of stratified

analysis. If carefully done, factors that independently predict a given outcome can be isolated and thus get the investigator closer to inference of causality.

Logistic regression using R

Logistic regression is part of `glm` which is used to fit **g**eneralized **l**inear **m**odels. `GLM` is part of the R base package. The basic formulation of the model is simple:

```
output <- glm(formula = outcome ~ factor(var01) + factor (var02) + var03,
  data=datasetname, family=binomial)
```

where `output` is the object to which the model results are assigned to, and `glm` is the actual function. In parenthesis, one opens with the formula statement and the name of the outcome variable following the `=` sign, followed by a tilde `~` and then all the variables in the model. Specification with `factor` is required for categorical variables, followed by the variable name of the variable in parenthesis as with any function (`factor` being the function here). In the example `var01` and `var02` are categorical variables, while `var03` is treated as a continuous variable. All the variables entering the equation are connected by `+` signs. A comma designates the end of the variable list and is followed by `data=` and the name of the dataset. Finally, `family=binomial` defines that the logit member of the `glm` family is to be used.

Why don't we just enter here for starters an example from our dataset? Please add the following line (all written onto a single line) at the end of our `e_ex03.r`:

```
mylogit <- glm(formula = outcome02 ~ factor(fq02) + factor(sex) + age, data=
  e_ex03, family=binomial)
```

Into the next line type:

```
summary(mylogit)
```

and you get:

```
Call:
glm(formula = outcome02 ~ factor(fq02) + factor(sex) + age, family = binomial,
    data = e_ex03)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3770   0.3750   0.5025   0.6043   1.3678

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)             2.54862    0.37757   6.750 1.48e-11 ***
factor(fq02)2-Resistant -1.10402    0.32517  -3.395 0.000686 ***
factor(sex)Male          0.90295    0.29063   3.107 0.001891 **
age                     -0.03619    0.01032  -3.506 0.000455 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 429.92  on 500  degrees of freedom
Residual deviance: 404.93  on 497  degrees of freedom
AIC: 412.93

Number of Fisher Scoring iterations: 5
```

Let's examine the output a bit, notably the part that begins with the `Coefficients`. We have `factor(fq02)2-Resistant` and `factor(sex)Male`. The referent for a variable is the lowest value. The values for `fq02` are `1-Susceptible` and `2-Resistant`. We

therefore see `factor(fq02)2-Resistant` in the line. `age` is a continuous variable, and as such it does not have a referent and is shown as it is.

For the numeric output we have the `Estimate` and `Std. Error`, and in the last column a probability `Pr(>|z|)` for the `Estimate`. The latter shows that all three variables are highly significant predictors, but otherwise it is difficult to gauge what these numbers mean. Only by making this more explicit it becomes more meaningful. The estimate is a logartithm, thus we do some exponentiation and column binding (`cbind`):

```
lreg.or <-exp(cbind(OR = coef(mylogit), confint(mylogit)))
round(lreg.or, digits=4)
```

and then get more meaningfully:

```
                            OR  2.5 %  97.5 %
(Intercept)            12.7894 6.2062 27.3682
factor(fq02)2-Resistant 0.3315 0.1768  0.6367
factor(sex)Male         2.4669 1.3943  4.3737
age                     0.9645 0.9450  0.9841
```

The two functions `coef` and `confint` take the estimate and calculate the confidence interval from the standard error respectively. `cbind` "glues" (Dalgaard) the vectors together. Exponentiating the logarithmic terms gives the odds ratios (labeled here as `OR`) with the 95% confidence interval that is more amenable to interpretation. Thus, we need both the `summary` and the conversion of the estimates. The former is required to determine how to proceed in the stepwise exclusion of variables from the model. Intuitively, we correctly use `age` as a continuous variable, but it is not certain whether this is actually also analytically correct. We should at least check first whether the influence of age has continuity as logistic regression will force a fixed slope.

How should we categorize age? We could follow the standard classification of WHO, but perhaps a bit less arbitrary is to have the data themselves dictating the categories. In our dataset we have the variables `agequart` and `agemed`. These are indeed data-driven quartiles of age and the variable age split into a binary variable defined by the median respectively. These variables were made based on the entire set of 515 records. As we are using a dataset that has 14 records removed, the values are now conceptually if not actually incorrect. We will thus first remove the two obsolete variables and then create a new variable for quartiles of age based on the actual 501 records. We used:

```
names(e_ex03)
```

before. This gives us the names of the variables and their position in the dataset:

```
 [1] "age"        "fq04"       "sex"        "totobstime" "agequart"   "agemed"     "outcome07"
 [8] "outcome02"  "pza02"      "kmy02"      "pth02"      "cxr02"      "fq02"
```

We can see / count that `agequart` and `agemed` take positions 5 and 6 respectively in the sequence of variables. In other words, we need only the variables in positions 1 through 4 and 7 through 13, which we write as:

```
e_ex03b <- e_ex03[c(1:4, 7:13)]
detach(e_ex03)
attach(e_ex03b)
names(e_ex03b)
```

where the last line `names(e_ex03b)` is only to check that we got what we intended to get and we did:

```
[1] "age"        "fq04"       "sex"        "totobstime" "outcome07"  "outcome02"  "pza02"
[8] "kmy02"      "pth02"      "cxr02"      "fq02"
```

Thus here the principle how to drop variables in R. Next we want to create a new variable from the existing variable `age`. As the values for the new variable are driven by the data, we need to know first more about the distribution of `age`. R has a powerful way of showing quantiles (look up the Help file by typing `??quantiles`). To obtain quartiles, we type:

```
quantile(age,  probs = c(25, 50, 75)/100)
```

and we get:

```
25% 50% 75%
 23  31  42
```

To create a new variable `agecat` from the existing variable age, we ensure (no problem if it is duplicated from above) that the correct file is in the path and then make the assignments followed by again detaching the file:

```
attach(e_ex03b)
e_ex03b$agecat[age >= 00 & age < 23] <- "Q1"
e_ex03b$agecat[age >= 23 & age < 31] <- "Q2"
e_ex03b$agecat[age >= 31 & age < 42] <- "Q3"
e_ex03b$agecat[age >= 42]            <- "Q4"
detach(e_ex03b)
```

We also note that our outcome variable "`outcome02`" is alphabetically listed as "`Failure`" first, then "`Success`". Intuitively, we might be more interested in risk factors for "Failure" rather than in risk factors for "Success". We thus make a rearrangement to get the order sequentially such that the odds calculated is Failure / Success, taking into account that R wishes the outcome to be `0` and `1` (zero and one) and write in full:

```
attach(e_ex03b)
e_ex03b$agecat[age >= 00 & age < 23] <- "Q1"
e_ex03b$agecat[age >= 23 & age < 31] <- "Q2"
e_ex03b$agecat[age >= 31 & age < 42] <- "Q3"
e_ex03b$agecat[age >= 42]            <- "Q4"
e_ex03b$out02[outcome02 == "Success"] <- 0
e_ex03b$out02[outcome02 == "Failure"] <- 1
detach(e_ex03b)
```

Then we give both new variables new names, write the data to disk, attach the new file, check the names to verify (not necessary, but it is always a good idea in the beginning to verify), and then write the output:

```
e_ex03c <- data.frame(e_ex03b)
write.table(e_ex03c, file="e_ex03c", row.names=TRUE)
attach(e_ex03c)
names(e_ex03c)
table(agecat, out02)
```

and get:

```
      out02
agecat   0   1
    Q1 103  10
    Q2 107  23
    Q3 109  20
    Q4 105  24
```

This ensures that for the age quartiles `Q1` is the referent and for treatment outcome `1-Success` is the referent.

Then we repeat the regression model with the categorized `agecat`:

```
# Logistic regression with AGE as categorical variable
mylogit2 <- glm(formula = out02 ~ factor(fq02) + factor(sex) + factor(agecat),
    data= e_ex03c, family=binomial)
summary(mylogit2)
lreg.or <-exp(cbind(OR = coef(mylogit2), confint(mylogit2)))
round(lreg.or, digits=4)
```

and get:

```
                            OR  2.5 %  97.5 %
(Intercept)             0.0841 0.0322  0.1807
factor(fq02)2-Resistant 0.7392 0.1711  2.2200
factor(sex)Male         0.4388 0.2357  0.8244
factor(agecat)Q2        3.1949 1.2311  9.3846
factor(agecat)Q3        3.4681 1.3332 10.2368
factor(agecat)Q4        5.6083 2.1556 16.7452
```

The referent is the youngest quartile set to unity. Relative to the referent, the risk of failure increases in the third and fourth quartile to similar levels and is then highest in the last quartile. This observation seems to support the treating of age as a continuous variable, or at least not plainly contradicting it.

This, so far has been "sampling" only. What we really wanted to evaluate is the influence of several factors that might modify the treatment outcome of multidrug-resistant tuberculosis. We will be starting with a full model inputting all variables, using instead of the binary `fq02` the more detailed `fq04`, slightly modified. The variable `fq04` had originally 4 levels, but in the source dataset of 501 records we are using in this exercise, the 14 records with a missing result have been dropped. We will first re-categorize the remaining 3 levels so as to ensure that the referent is `Susceptible`. As shown earlier, we do this by adding a sequential number to the desired sequence to get it alphabetically correct:

```
attach(e_ex03c)
e_ex03c$fq03[fq04 == "Susceptible"]           <- "1-Susceptible"
e_ex03c$fq03[fq04 == "Low-level resistance"]  <- "2-Low-level resistance"
e_ex03c$fq03[fq04 == "High-level resistance"] <- "3-High-level resistance"
detach(e_ex03c)
```

Not that it is essential but for getting things tidy, save the new as a data frame and read it in:

```
e_ex03d <- data.frame(e_ex03c)
write.table(e_ex03d, file="e_ex03d", row.names=TRUE)
e_ex03e <- read.table("e_ex03d")
```

and then we are ready for the full model:

```
mylogit3 <- glm(formula = out02 ~ factor(fq03) + factor(sex) + age +
    factor(pza02) + factor(kmy02) + factor(pth02) + factor(cxr02),
    data=e_ex03e, family=binomial)
summary(mylogit3)
```

and we get:

```
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -2.54930    0.39982  -6.376 1.82e-10 ***
factor(fq03)2-Low-level resistance  -0.16501    0.63817  -0.259 0.795972
factor(fq03)3-High-level resistance  2.21715    0.46176   4.802 1.57e-06 ***
factor(sex)Male                 -0.85679    0.30327  -2.825 0.004726 **
age                              0.03765    0.01054   3.571 0.000356 ***
factor(pza02)PZA resistant      -0.37179    0.37015  -1.004 0.315169
factor(kmy02)KM resistant        0.18024    1.49296   0.121 0.903908
factor(pth02)PTH resistant      -0.27951    0.37987  -0.736 0.461845
factor(cxr02)Not known bilateral  0.06088    0.32561   0.187 0.851678
```

Note also the "AIC" at the end to which will pay particular attention in the following:

```
AIC: 407.95
```

AIC stands for **A**kaike's **I**nformation **C**riterion and is a weighted criterion of goodness of fit. The smaller the value, the better the fit. As we are going with stepwise elimination as one of the common procedures (Crawley M J. The R book. Second edition, Wiley, Chichester, UK, 2013, 1051 pp), we are checking how the criterion changes. Gauging from the probabilities above (last column), kanamycin resistance is the first factor to be removed, thus the model is reduced to:

```
mylogit3 <- glm(formula = out02 ~ factor(fq03) + factor(sex) + age +
    factor(pza02)   +   factor(pth02)   +   factor(cxr02),   data=e_ex03e,
    family=binomial)
summary(mylogit3)
```

Improving the AIC to 405.96 and putting removal of the chest radiography result as the next. This removal improves the AIC to 403.99 and suggest removal of prothionamide resistance next. This removal improves the AIC to 402.54 and suggests removal of pyrazinamide resistance next. This removal improves the AIC to 401.55. The model is now:

```
mylogit3 <- glm(formula = out02 ~ factor(fq03) + factor(sex) + age,
    data=e_ex03e, family=binomial)
summary(mylogit3)
```

and gives:

```
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -2.65259    0.38786  -6.839 7.97e-12 ***
factor(fq03)2-Low-level resistance  -0.28525    0.62974  -0.453 0.650577
factor(fq03)3-High-level resistance  2.07719    0.41969   4.949 7.45e-07 ***
factor(sex)Male                 -0.80689    0.29781  -2.709 0.006741 **
age                              0.03733    0.01047   3.564 0.000366 ***
```

While low-level resistance to the fluoroquinolone is not a predictor, high-level resistance is a strong predictor, and both `sex` and `age` are predictors. Following the rules of being strict, none of the remaining factors should thus be removed.

If you watched it through the process of elimination, you found:

Modeling step	Resulting AIC value
Full model	407.95
Remove `kmy02`	405.96
Remove `cxr02`	403.99
Remove `pth02`	402.54
Remove `pza02`	401.55

What we haven't done yet, but are very much obliged to do is to test whether there is heterogeneity in the data and we therefore need one or more interaction terms. *Woolf's test for interaction* (also known as *Woolf's test for the homogeneity of odds ratios*) provides a formal test for Interaction. According to Mark Myatt (see text recommended at the beginning of Part E), R does not provide Woolf's test specifically, but the grammar of the function can be found in the help section on mantelhaen.test which we have used earlier. We use here the slightly modified script by Myatt (giving the same result), but is slightly more explicit. We type (perhaps best in the console, so that it does not interfere with our script `e_ex03.r`:

```
woolf.test <- function(x) {}
fix(woolf.test)
```

Then in the editor box:

```
function(x)
    {
    x <- x + 0.5
    k <- dim(x)[3]
    or <- apply(x, 3, function(x)
    {(x[1, 1] / x[1, 2]) / (x[2, 1] / x[2, 2])})
    w <- apply(x, 3, function(x) {1 / sum(1 / x)})
    chi.sq <- sum(w * (log(or) - weighted.mean(log(or), w))^2)
    p <- pchisq(chi.sq, df = k - 1, lower.tail = FALSE)
    cat("\nWoolf's X2 :", chi.sq, "\np-value :", p, "\n")
}
```

Finally we save:

```
save(woolf.test, file = "woolf.test.r")
```

Back to our `e_ex03.r` script (not necessary, but no harm either for this time, as it is loaded, but for the next run after we close, and also that we note where the credit goes to):

```
# Test for interaction, using the script for
# Woolf's test provided by Mark Myatt
load("c:/epidata_course/woolf.test.r")
```

The function requires one parameter, denoted here as `x`. If we are interested in the interaction between outcome, ofloxacin resistance, and sex, we could make one object:

```
tabofl <- table(fq03, out02, sex)
```

and then:

```
woolf.test(tabofl)
```

and we get:

```
Woolf's X2 : 0.2271536
p-value : 0.6336425
```

Thus, not to worry about heterogeneity. For the sake of exercise, and also to look what happens, we will nevertheless put in an interaction term:

```
mylogit3 <- glm(formula = out02 ~ factor(fq03) + factor(sex) + age +
   factor(sex):factor(fq03), data=e_ex03e, family=binomial)
summary(mylogit3)
```

and we get a poorer AIC of 405.45 and insignificant interaction terms:

```
                                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                                     -2.666178   0.391316  -6.813 9.53e-12 ***
factor(fq03)2-Low-level resistance               0.008907   1.132799   0.008 0.993727
factor(fq03)3-High-level resistance              2.146511   0.626505   3.426 0.000612 ***
factor(sex)Male                                 -0.772537   0.326321  -2.367 0.017913 *
age                                              0.037092   0.010518   3.526 0.000421 ***
factor(fq03)2-Low-level resistance:factor(sex)Male -0.413234   1.365405  -0.303 0.762160
factor(fq03)3-High-level resistance:factor(sex)Male -0.123552   0.837512  -0.148 0.882719
```

If we find a significant interaction and perhaps even the AIC improves, then the interaction term must be retained. As you note above, it is simple to write the interaction term which is writing the two variables connected by a colon as in `factor(var1):factor(var2)`.

We are therefore pretty much assured that this is our final model:

```
# Final model:
mylogit3 <- glm(formula = out02 ~ factor(fq03) + factor(sex) + age,
  data=e_ex03e, family=binomial)
summary(mylogit3)
lreg.or <-exp(cbind(OR = coef(mylogit3), confint(mylogit3)))
round(lreg.or, digits=4)
```

With the following odds ratios and 95% confidence intervals:

```
                                    OR  2.5 %  97.5 %
(Intercept)                     0.0705 0.0322  0.1478
factor(fq03)2-Low-level resistance  0.7518 0.1744  2.2454
factor(fq03)3-High-level resistance 7.9820 3.5104 18.4214
factor(sex)Male                 0.4462 0.2485  0.8021
age                             1.0380 1.0170  1.0598
```

Task:

- ***Examine whether there are interactions between age and sex and fluoroquinolone resistant and age, so that all possibilities have been checked before we accept the model.***