Analysis of Matched Case-Control Studies

Kamarul Imran Musa

21 February 2017

Contents

Matched Case-Control Studies Require Special Logistic Regression Approach

To analyze case-control data with pairwise matching we use conditional logistic regression. Cases are individually matched to 1 (or up to 4) control subjects based on matching criteria. These matching criteria represent the confounders to be controlled. Each case and matched control(s) are analyzed as separate strata. For example, if there are 150 cases, then there are 150 strata that need to be fitted. This is accomplished with the Cox proportional hazards regression functions available in the Survival package.

Motivation

With matched pairs data the form of the logistic model involves the probability, ϕ, that in matched pair number i, for a given value of the explanatory variable the member of the pair is a case. Specifically the model is

$$logit(\phi_i) = \alpha_i + \beta x$$

The odds that a subject with x = 1 is a case equals $exp(\beta)$ times the odds that a subject with x = 0 is a case.

Tutorial 1

Prepare workspace and data

Set working directory

```r
setwd("E:/Epi_Stat_Matters/LectureNotes2015/Clogit-DrPH-Epid-2015-16")
```

Let's get an overview of the data.

```r
library(HSAUR2)
```

```
## Loading required package: tools
```

```r
head(backpain)
```

```
##   ID  status driver suburban
## 1  1    case    yes      yes
## 2  1 control    yes       no
## 3  2    case    yes      yes
## 4  2 control    yes      yes
## 5  3    case    yes       no
## 6  3 control    yes      yes
```

Describe data

```r
library(psych)
describe(backpain)
```

```
##           vars   n   mean    sd median trimmed   mad min max range  skew
## ID*          1 434 109.00 62.71  109.0  109.00 80.06   1 217   216  0.00
## status*      2 434   1.50  0.50    1.5    1.50  0.74   1   2     1  0.00
## driver*      3 434   1.80  0.40    2.0    1.88  0.00   1   2     1 -1.51
## suburban*    4 434   1.54  0.50    2.0    1.55  0.00   1   2     1 -0.16
##           kurtosis   se
## ID*          -1.21 3.01
## status*      -2.00 0.02
## driver*       0.28 0.02
## suburban*    -1.98 0.02
```

Perform survival::clogit

```
library(survival)
backpain_glm <- clogit(I(status == 'case') ~ driver + suburban + strata(ID), data = backpain)
summary(backpain_glm)
```

```
## Call:
## coxph(formula = Surv(rep(1, 434L), I(status == "case")) ~ driver +
##     suburban + strata(ID), data = backpain, method = "exact")
##
##   n= 434, number of events= 217
##
##               coef exp(coef) se(coef)     z Pr(>|z|)
## driveryes   0.6579    1.9307   0.2940 2.238   0.0252 *
## suburbanyes 0.2555    1.2911   0.2258 1.131   0.2580
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##             exp(coef) exp(-coef) lower .95 upper .95
## driveryes       1.931     0.5180    1.0851     3.435
## suburbanyes     1.291     0.7746    0.8293     2.010
##
## Rsquare= 0.022   (max possible= 0.5 )
## Likelihood ratio test= 9.55  on 2 df,   p=0.008457
## Wald test            = 8.85  on 2 df,   p=0.01195
## Score (logrank) test = 9.31  on 2 df,   p=0.0095
```

Interpretation

The estimate of the odds ratio of a herniated disc occurring in a driver relative to a nondriver is 1.93 with a 95% confidence interval of (1.09, 3.44). Conditional on residence we can say that the risk of a herniated disc occurring in a driver is about twice that of a nondriver. There is no evidence that where a person lives affects the risk of lower back pain.

Tutorial 2

Prepare workspace and data

Set working directory

```
setwd("E:/Epi_Stat_Matters/LectureNotes2015/Clogit-DrPH-Epid-2015-16")
```

Read data

```r
# source for data
# use read.table('http://www.medepi.net/data/mi.txt',sep="")
data1<-read.csv('dataclogit.csv',header = TRUE)
```

Overview of data

View the first 6 observations

```r
head(data1)
```

```
##   match person mi smk sbp ecg
## 1     1      1  1   0 160   1
## 2     1      2  0   0 140   0
## 3     1      3  0   0 120   0
## 4     2      4  1   0 160   1
## 5     2      5  0   0 140   0
## 6     2      6  0   0 120   0
```

Convert and Labels Data

Convert the variables mi, smk and ecg to categorical variables

```r
data1$mi2 <- factor(data1$mi, levels = c(1,0), labels = c("Case","Control"))
data1$smk2 <- factor(data1$smk, levels = c(0,1), labels=c("Not current","Current"))
data1$ecg2 <- factor(data1$ecg, levels = c(0,1), labels=c("Normal","Abnormal"))
str(data1,15)
```

```
## 'data.frame':    117 obs. of  9 variables:
##  $ match : int  1 1 1 2 2 2 3 3 3 4 ...
##  $ person: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ mi    : int  1 0 0 1 0 0 1 0 0 1 ...
##  $ smk   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sbp   : int  160 140 120 160 140 120 160 140 120 160 ...
##  $ ecg   : int  1 0 0 1 0 0 0 0 0 0 ...
##  $ mi2   : Factor w/ 2 levels "Case","Control": 1 2 2 1 2 2 1 2 2 1 ...
##  $ smk2  : Factor w/ 2 levels "Not current",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ ecg2  : Factor w/ 2 levels "Normal","Abnormal": 2 1 1 2 1 1 1 1 1 1 ...
```

Quick Look at Data

View the first 15 observations

```r
head(data1,15)
```

```
##    match person mi smk sbp ecg     mi2        smk2     ecg2
## 1      1      1  1   0 160   1    Case Not current Abnormal
## 2      1      2  0   0 140   0 Control Not current   Normal
## 3      1      3  0   0 120   0 Control Not current   Normal
## 4      2      4  1   0 160   1    Case Not current Abnormal
## 5      2      5  0   0 140   0 Control Not current   Normal
## 6      2      6  0   0 120   0 Control Not current   Normal
```

```
## 7       3       7  1    0 160   0     Case Not current   Normal
## 8       3       8  0    0 140   0  Control Not current   Normal
## 9       3       9  0    0 120   0  Control Not current   Normal
## 10      4      10  1    0 160   0     Case Not current   Normal
## 11      4      11  0    0 140   0  Control Not current   Normal
## 12      4      12  0    0 120   0  Control Not current   Normal
## 13      5      13  1    0 160   0     Case Not current   Normal
## 14      5      14  0    0 140   0  Control Not current   Normal
## 15      5      15  0    0 120   0  Control Not current   Normal
```

```r
# load survival package to run clogit
```

Perfom data exploration

```r
library(psych)
describe(data1)
```

```
##         vars   n   mean    sd median trimmed   mad min max range  skew
## match      1 117  20.00 11.30     20   20.00 14.83   1  39    38  0.00
## person     2 117  59.00 33.92     59   59.00 43.00   1 117   116  0.00
## mi         3 117   0.33  0.47      0    0.29  0.00   0   1     1  0.70
## smk        4 117   0.28  0.45      0    0.23  0.00   0   1     1  0.96
## sbp        5 117 136.41 16.11    140  135.58 29.65 120 160    40  0.33
## ecg        6 117   0.21  0.41      0    0.14  0.00   0   1     1  1.44
## mi2*       7 117   1.67  0.47      2    1.71  0.00   1   2     1 -0.70
## smk2*      8 117   1.28  0.45      1    1.23  0.00   1   2     1  0.96
## ecg2*      9 117   1.21  0.41      1    1.14  0.00   1   2     1  1.44
##         kurtosis   se
## match      -1.23 1.04
## person     -1.23 3.14
## mi         -1.53 0.04
## smk        -1.09 0.04
## sbp        -1.40 1.49
## ecg         0.08 0.04
## mi2*       -1.53 0.04
## smk2*      -1.09 0.04
## ecg2*       0.08 0.04
```

Now, by groups

```r
describeBy(data1, group = 'mi2')
```

```
## $Case
##         vars  n   mean    sd median trimmed   mad min max range  skew
## match      1 39  20.00 11.40     20   20.00 14.83   1  39    38  0.00
## person     2 39  58.00 34.21     58   58.00 44.48   1 115   114  0.00
## mi         3 39   1.00  0.00      1    1.00  0.00   1   1     0   NaN
## smk        4 39   0.38  0.49      0    0.36  0.00   0   1     1  0.46
## sbp        5 39 145.13 18.76    160  146.06  0.00 120 160    40 -0.51
## ecg        6 39   0.33  0.48      0    0.30  0.00   0   1     1  0.68
## mi2*       7 39   1.00  0.00      1    1.00  0.00   1   1     0   NaN
## smk2*      8 39   1.38  0.49      1    1.36  0.00   1   2     1  0.46
## ecg2*      9 39   1.33  0.48      1    1.30  0.00   1   2     1  0.68
##         kurtosis   se
## match      -1.29 1.83
## person     -1.29 5.48
## mi           NaN 0.00
```

```
## smk       -1.84 0.08
## sbp       -1.69 3.00
## ecg       -1.58 0.08
## mi2*        NaN 0.00
## smk2*     -1.84 0.08
## ecg2*     -1.58 0.08
##
## $Control
##        vars  n   mean    sd median trimmed   mad min max range skew
## match     1 78  20.00 11.33   20.0   20.00 14.83   1  39    38 0.00
## person    2 78  59.50 33.99   59.5   59.50 43.74   2 117   115 0.00
## mi        3 78   0.00  0.00    0.0    0.00  0.00   0   0     0  NaN
## smk       4 78   0.23  0.42    0.0    0.17  0.00   0   1     1 1.25
## sbp       5 78 132.05 12.62  140.0  130.62 29.65 120 160    40 0.53
## ecg       6 78   0.14  0.35    0.0    0.06  0.00   0   1     1 2.02
## mi2*      7 78   2.00  0.00    2.0    2.00  0.00   2   2     0  NaN
## smk2*     8 78   1.23  0.42    1.0    1.17  0.00   1   2     1 1.25
## ecg2*     9 78   1.14  0.35    1.0    1.06  0.00   1   2     1 2.02
##        kurtosis   se
## match     -1.25 1.28
## person    -1.25 3.85
## mi          NaN 0.00
## smk       -0.43 0.05
## sbp       -0.69 1.43
## ecg        2.12 0.04
## mi2*        NaN 0.00
## smk2*     -0.43 0.05
## ecg2*      2.12 0.04
##
## attr(,"call")
## by.data.frame(data = x, INDICES = group, FUN = describe, type = type)
```

Run clogit Function

This requires **survival** package

```
library("survival")
res.clog <- clogit(I(mi2=='Case') ~ smk2 + strata(match), data = data1)
summary(res.clog)
```

```
## Call:
## coxph(formula = Surv(rep(1, 117L), I(mi2 == "Case")) ~ smk2 +
##     strata(match), data = data1, method = "exact")
##
##   n= 117, number of events= 39
##
##               coef exp(coef) se(coef)     z Pr(>|z|)
## smk2Current 0.8434    2.3242   0.4661 1.809   0.0704 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##             exp(coef) exp(-coef) lower .95 upper .95
## smk2Current     2.324     0.4303    0.9322     5.794
```

```
##
## Rsquare= 0.028   (max possible= 0.519 )
## Likelihood ratio test= 3.37  on 1 df,   p=0.06655
## Wald test            = 3.27  on 1 df,   p=0.07038
## Score (logrank) test = 3.43  on 1 df,   p=0.06408
```

Reference

Can read here http://www.medepi.net/docs/ph251d2013fall_REGRESSION-CHAP.pdf

Tutorial 3

Dataset 2

Let us play with another dataset. This tutorial comes from:

https://denishaine.wordpress.com/2013/03/22/veterinary-epidemiologic-research-glm-part-4-exact-and-conditional-logistic-reg

Salmonella outbreak dataset in stata format

Read stata file

load foreign library to read stata file

```r
library(foreign)
# read data
data2 <- read.dta('sal_outbrk.dta', convert.factors = T)
# see variable names
names(data2)
```

```
##  [1] "match_grp"   "date"        "age"         "gender"      "casecontrol"
##  [6] "eatbeef"     "eatpork"     "eatveal"     "eatlamb"     "eatpoul"
## [11] "eatcold"     "eatveg"      "eatfruit"    "eateggs"     "slt_a"
## [16] "dlr_a"       "dlr_b"
```

Quickly examine data

```r
head(data2)
```

```
##   match_grp       date      age gender casecontrol eatbeef eatpork eatveal
## 1         1 1996-09-27 52.28748   Male        case     yes     yes     yes
## 2         1 1996-09-29 52.29295   Male     control     yes      no      no
## 3         1 1996-09-28 52.29021   Male     control     yes     yes      no
## 4         2 1996-10-01 41.01300   Male        case     yes    <NA>    <NA>
## 5         2 1996-10-12 41.03765   Male     control     yes     yes      no
## 6         2 1996-09-29 41.01027   Male     control     yes     yes      no
##   eatlamb eatpoul eatcold eatveg eatfruit eateggs slt_a dlr_a dlr_b
## 1      no     yes     yes     no      yes     yes   yes    no   yes
## 2      no      no     yes    yes      yes      no    no    no    no
## 3      no     yes     yes    yes      yes     yes    no    no    no
## 4    <NA>    <NA>     yes    yes     <NA>    <NA>    no  <NA>  <NA>
```

```
## 5      no       no      yes     yes       no      yes    yes    yes     no
## 6      no       no      yes     yes      yes      yes    yes     no    yes
```

Run the clogit analysis

Load survival package to run analysis. clogit is a function under survival package (survival::clogit)

```
library(survival)
mod7 <- clogit(I(casecontrol=='case') ~ slt_a + strata(match_grp), data = data2)
summary(mod7)
```

```
## Call:
## coxph(formula = Surv(rep(1, 112L), I(casecontrol == "case")) ~
##     slt_a + strata(match_grp), data = data2, method = "exact")
##
##   n= 112, number of events= 39
##
##            coef exp(coef) se(coef)     z Pr(>|z|)
## slt_ayes 1.4852    4.4159   0.5181 2.867  0.00415 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          exp(coef) exp(-coef) lower .95 upper .95
## slt_ayes     4.416     0.2265       1.6     12.19
##
## Rsquare= 0.085   (max possible= 0.518 )
## Likelihood ratio test= 10  on 1 df,   p=0.001568
## Wald test            = 8.22  on 1 df,   p=0.004148
## Score (logrank) test = 9.48  on 1 df,   p=0.002075
```

Tutorial 4

Dataset 3

This comes from hosmer book

Reference

This data and tutorial come from example : http://www.ats.ucla.edu/stat/stata/examples/alr2/alr2stata7.htm

Check data

```
readLines('lowbwt11.dat', n=5)
```

```
## [1] "           1      0      14     135        1        0       0      0      0"
## [2] "           1      1      14     101        3        1       1      0      0"
## [3] "           2      0      15      98        2        0       0      0      0"
## [4] "           2      1      15     115        3        0       0      0      1"
## [5] "           3      0      16      95        3        0       0      0      0"
```

Import data

We will read a .dat data.

```
data3<-read.table('lowbwt11.dat')
```

Quickly view data

Overview of data

```
head(data3,10)
```

```
##    V1 V2 V3  V4 V5 V6 V7 V8 V9
## 1   1  0 14 135  1  0  0  0  0
## 2   1  1 14 101  3  1  1  0  0
## 3   2  0 15  98  2  0  0  0  0
## 4   2  1 15 115  3  0  0  0  1
## 5   3  0 16  95  3  0  0  0  0
## 6   3  1 16 130  3  0  0  0  0
## 7   4  0 17 103  3  0  0  0  0
## 8   4  1 17 130  3  1  1  0  1
## 9   5  0 17 122  1  1  0  0  0
## 10  5  1 17 110  1  1  0  0  0
```

Names the columns

We give names to columns

```
colnames(data3)<-c('pair','low','age','lwt','race','smoke','ptd','ht','ui')
```

Declare variables as factors (categorical variables)

Using lapply is fast

```
data3[,c(2,5:9)]<-lapply(data3[,c(2,5:9)],  as.factor)
head(data3)
```

```
##   pair low age lwt race smoke ptd ht ui
## 1    1   0  14 135    1     0   0  0  0
## 2    1   1  14 101    3     1   1  0  0
## 3    2   0  15  98    2     0   0  0  0
## 4    2   1  15 115    3     0   0  0  1
## 5    3   0  16  95    3     0   0  0  0
## 6    3   1  16 130    3     0   0  0  0
```

Specify the levels of the categorical variables

```
levels(data3$low) <- c('bwt>2500g','bwt=<2500g')
levels(data3$race) <- c('white','black','other')
levels(data3$smoke) <- c('no','yes')
levels(data3$ptd) <- c('none','yes')
levels(data3$ht) <- c('no','yes')
```

```r
levels(data3$ui) <- c('no','yes')
str(data3)
```

```
## 'data.frame':    112 obs. of  9 variables:
##  $ pair : int  1 1 2 2 3 3 4 4 5 5 ...
##  $ low  : Factor w/ 2 levels "bwt>2500g","bwt=<2500g": 1 2 1 2 1 2 1 2 1 2 ...
##  $ age  : int  14 14 15 15 16 16 17 17 17 17 ...
##  $ lwt  : int  135 101 98 115 95 130 103 130 122 110 ...
##  $ race : Factor w/ 3 levels "white","black",..: 1 3 2 3 3 3 3 3 1 1 ...
##  $ smoke: Factor w/ 2 levels "no","yes": 1 2 1 1 1 1 1 2 2 2 ...
##  $ ptd  : Factor w/ 2 levels "none","yes": 1 2 1 1 1 1 1 2 1 1 ...
##  $ ht   : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ui   : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 2 1 1 ...
```

Perform clogit function

covariate = lwt

```r
c.data3 <- clogit(I(low=='bwt=<2500g') ~ lwt + strata(pair), data = data3)
summary(c.data3)
```

```
## Call:
## coxph(formula = Surv(rep(1, 112L), I(low == "bwt=<2500g")) ~
##     lwt + strata(pair), data = data3, method = "exact")
##
##   n= 112, number of events= 56
##
##          coef exp(coef)  se(coef)      z Pr(>|z|)
## lwt -0.009375  0.990669  0.006165 -1.521    0.128
##
##     exp(coef) exp(-coef) lower .95 upper .95
## lwt    0.9907      1.009    0.9788     1.003
##
## Rsquare= 0.022   (max possible= 0.5 )
## Likelihood ratio test= 2.51  on 1 df,   p=0.1131
## Wald test            = 2.31  on 1 df,   p=0.1284
## Score (logrank) test = 2.44  on 1 df,   p=0.1182
```

```r
c.data3sm <- clogit(I(low=='bwt=<2500g') ~ smoke + strata(pair), data = data3)
summary(c.data3sm)
```

```
## Call:
## coxph(formula = Surv(rep(1, 112L), I(low == "bwt=<2500g")) ~
##     smoke + strata(pair), data = data3, method = "exact")
##
##   n= 112, number of events= 56
##
##            coef exp(coef) se(coef)    z Pr(>|z|)
## smokeyes 1.0116    2.7500   0.4129 2.45   0.0143 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          exp(coef) exp(-coef) lower .95 upper .95
## smokeyes      2.75     0.3636     1.224     6.177
```

```
##
## Rsquare= 0.059   (max possible= 0.5 )
## Likelihood ratio test= 6.79  on 1 df,   p=0.009147
## Wald test            = 6  on 1 df,   p=0.01428
## Score (logrank) test = 6.53  on 1 df,   p=0.01059
```

Other issues to consider in clogit

Test Functional Form for Numerical Variable

unable to do with mfp with surv

Can refer here http://www.ats.ucla.edu/stat/stata/examples/alr2/alr2stata7.htm

cut function to break numerical variables

```r
data3$cat.lwt <- cut(data3$lwt, breaks = c(min(data3$lwt)-1, 106.5, 120.0, 136.5, max(data3$lwt)))
table(data3$cat.lwt)
```

```
##
##  (79,106] (106,120] (120,136] (136,241]
##        28        31        25        28
```

Run clogit again

```r
c.data3des <- clogit(I(low=='bwt=<2500g') ~ cat.lwt + smoke + ptd + ht + ui + strata(pair), data = data3
summary(c.data3des)
```

```
## Call:
## coxph(formula = Surv(rep(1, 112L), I(low == "bwt=<2500g")) ~
##     cat.lwt + smoke + ptd + ht + ui + strata(pair), data = data3,
##     method = "exact")
##
##   n= 112, number of events= 56
##
##                      coef exp(coef) se(coef)      z Pr(>|z|)
## cat.lwt(106,120] -0.3991    0.6710   0.6635 -0.601   0.5475
## cat.lwt(120,136] -0.4430    0.6421   0.6718 -0.659   0.5096
## cat.lwt(136,241] -0.8887    0.4112   0.6255 -1.421   0.1553
## smokeyes          1.3527    3.8680   0.5568  2.429   0.0151 *
## ptdyes            1.7398    5.6964   0.7462  2.332   0.0197 *
## htyes             1.8926    6.6363   0.9647  1.962   0.0498 *
## uiyes             1.3162    3.7293   0.6886  1.911   0.0559 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                  exp(coef) exp(-coef) lower .95 upper .95
## cat.lwt(106,120]    0.6710     1.4904    0.1828     2.463
## cat.lwt(120,136]    0.6421     1.5574    0.1721     2.396
## cat.lwt(136,241]    0.4112     2.4320    0.1207     1.401
## smokeyes            3.8680     0.2585    1.2988    11.520
## ptdyes              5.6964     0.1756    1.3195    24.591
## htyes               6.6363     0.1507    1.0018    43.960
```

```
## uiyes               3.7293     0.2681    0.9672    14.379
##
## Rsquare= 0.19   (max possible= 0.5 )
## Likelihood ratio test= 23.55  on 7 df,   p=0.001365
## Wald test            = 12.29  on 7 df,   p=0.09145
## Score (logrank) test = 18.74  on 7 df,   p=0.009055
```

Tidy your R output

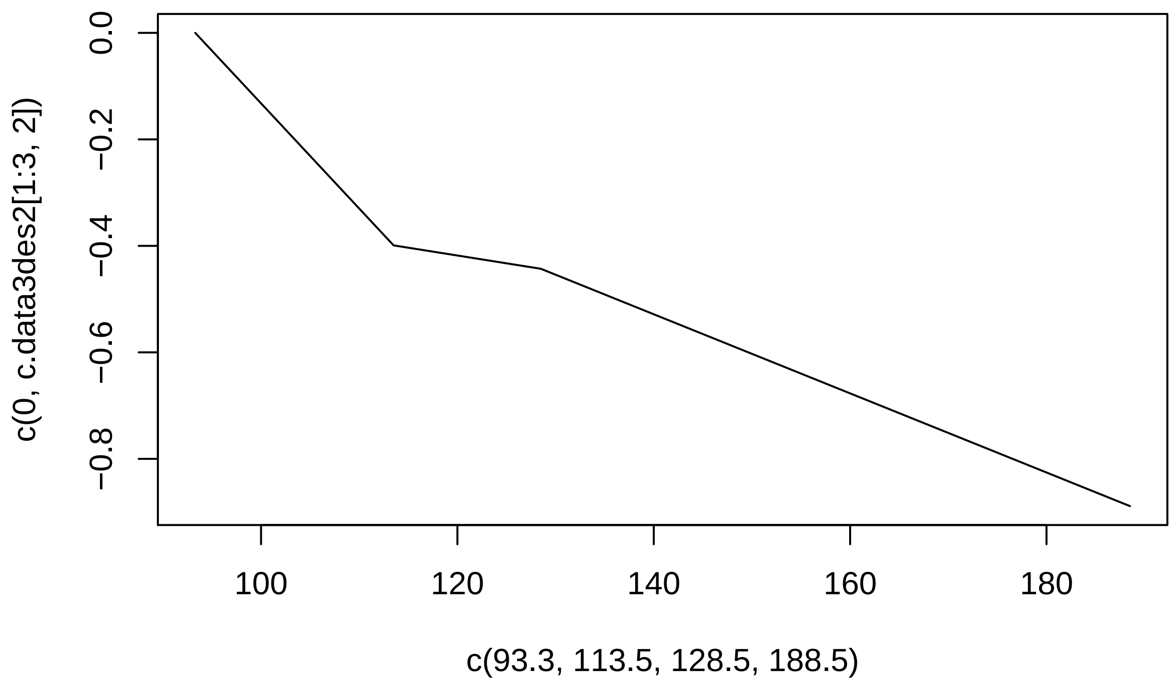
Nice outputs

```r
library(broom)
c.data3des2 <- tidy(c.data3des)
c.data3des2
```

```
##                term   estimate std.error  statistic    p.value     conf.low
## 1 cat.lwt(106,120] -0.3990522 0.6634509 -0.6014796 0.54752057 -1.699392150
## 2 cat.lwt(120,136] -0.4430378 0.6718024 -0.6594764 0.50958990 -1.759746225
## 3 cat.lwt(136,241] -0.8887328 0.6254701 -1.4209037 0.15534475 -2.114631600
## 4         smokeyes  1.3527363 0.5568023  2.4294734 0.01512077  0.261423912
## 5           ptdyes  1.7398286 0.7462135  2.3315426 0.01972477  0.277276973
## 6            htyes  1.8925552 0.9646784  1.9618509 0.04977985  0.001820261
## 7            uiyes  1.3162091 0.6885803  1.9114822 0.05594264 -0.033383589
##   conf.high
## 1 0.9012877
## 2 0.8736706
## 3 0.3371661
## 4 2.4440487
## 5 3.2023802
## 6 3.7832900
## 7 2.6658017
```

Now, we plot the mid-points to see the pattern of 'linearity in logits'

```r
# plot (midpoint vs beta)
plot(c(93.3, 113.5, 128.5, 188.5),c(0, c.data3des2[1:3,2]), type = 'l')
```



Prediction

```
data3final <- clogit(I(low=='bwt=<2500g') ~ lwt + smoke + ptd + ht + ui + strata(pair), data = data3)
summary(data3final)
```

```
## Call:
## coxph(formula = Surv(rep(1, 112L), I(low == "bwt=<2500g")) ~
##     lwt + smoke + ptd + ht + ui + strata(pair), data = data3,
##     method = "exact")
##
##   n= 112, number of events= 56
##
##                coef exp(coef)  se(coef)      z Pr(>|z|)
## lwt       -0.015083  0.985030  0.008147 -1.852  0.06409 .
## smokeyes   1.479564  4.391033  0.562019  2.633  0.00847 **
## ptdyes     1.670594  5.315326  0.746806  2.237  0.02529 *
## htyes      2.329361 10.271381  1.002549  2.323  0.02016 *
## uiyes      1.344895  3.837782  0.693843  1.938  0.05258 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          exp(coef) exp(-coef) lower .95 upper .95
## lwt          0.985    1.01520    0.9694     1.001
## smokeyes     4.391    0.22774    1.4594    13.212
## ptdyes       5.315    0.18814    1.2298    22.973
```

```
## htyes       10.271    0.09736    1.4397    73.283
## uiyes        3.838    0.26057    0.9851    14.951
##
## Rsquare= 0.201   (max possible= 0.5 )
## Likelihood ratio test= 25.16  on 5 df,   p=0.0001298
## Wald test            = 12.59  on 5 df,   p=0.0275
## Score (logrank) test = 19.78  on 5 df,   p=0.001372
```

```
nice.op <- tidy(data3final)
write.csv(nice.op,'tableclogit.csv')
```

calculate the probability of a positive outcome conditional on one positive outcome within group in stata we use

1. [predict probposOC, pc1] for probability and
2. [predict LinPred, xb] for linear predictor (log odds)

Predict (not as good as stata): * type = 'expected' gives the predicted probability - calculates the probability of a positive outcome conditional on one positive outcome within group (strata)

```
# predicted probability
data3finalfitted <- predict(data3final, type = 'expected')
cbind(data3[1:10, c(1:3, 4,6:9)], data3finalfitted[1:10])
```

```
##    pair        low age lwt smoke  ptd ht  ui data3finalfitted[1:10]
## 1     1  bwt>2500g  14 135    no none no  no             0.02501381
## 2     1 bwt=<2500g  14 101   yes  yes no  no             0.97498619
## 3     2  bwt>2500g  15  98    no none no  no             0.25190531
## 4     2 bwt=<2500g  15 115    no none no yes             0.74809469
## 5     3  bwt>2500g  16  95    no none no  no             0.62899786
## 6     3 bwt=<2500g  16 130    no none no  no             0.37100214
## 7     4  bwt>2500g  17 103    no none no  no             0.01649929
## 8     4 bwt=<2500g  17 130   yes  yes no yes             0.98350071
## 9     5  bwt>2500g  17 122   yes none no  no             0.45487285
## 10    5 bwt=<2500g  17 110   yes none no  no             0.54512715
```

- in a conditional logistic the "expected number of events" is just exp(eta)/(1 + exp(eta)) where eta is the linear predictor. In stata this is known as the probability of a positive outcome, assuming that the fixed effect is zero. See http://grokbase.com/t/r/r-help/146gcqqxse/r-prediction-based-on-conditional-logistic-regression-clogit. Also see below

```
odds_low <- predict(data3final, type = "risk")
(odds_low/(odds_low+1))[1:10]
```

```
##         1         2         3         4         5         6         7
## 0.1380600 0.8619400 0.3672022 0.6327978 0.5656095 0.4343905 0.1146702
##         8         9        10
## 0.8853298 0.4773903 0.5226097
```

Assignments

1. Find a suitable matched data
2. Run conditional logistic analysis
3. Run a model with and without an interaction term
4. Run diagnostic test
5. Create a publishable table

References

1. https://cran.r-project.org/web/packages/HSAUR2/vignettes/Ch_logistic_regression_glm.pdf
2. http://grokbase.com/t/r/r-help/146gcqqxse/r-prediction-based-on-conditional-logistic-regression-clogit
3. http://stackoverflow.com/questions/35329585/how-to-get-fitted-values-from-clogit-model

Notes

See page 300, Chapter 7, Regression Models for categorical Dependent variables using Stata

If we estimate the predict probability (option pc1: conditional probability for single outcome within group) then we interpret like this.

For example, the predicted probability for

```
id3 <- c(1,1,1)
prob3 <- c(0.064, 0.107, 0.925)
outcome3 <- c(0,0,1)
data3 <- cbind(id3, outcome3, prob3)
data3
```

```
##      id3 outcome3 prob3
## [1,]   1        0 0.064
## [2,]   1        0 0.107
## [3,]   1        1 0.925
```

It means that this group, the predicted probability to be the case (outcome = 1) for first observation is 6.4%, the second observation was 10.7% and the third observation was 92.5%.