Lecture 4 BIO206

Logistic regression: categorical variables

Logistic regression

- Logistic regression requires the transition from the basic (least-square-based) *general linear model* to the intermediate/advanced *generalised linear model*

- The generalised linear model extends linear techniques to variables that are not normally distributed

- For example, we may want to use regression techniques to predict *binary* responses:
  - we may want to predict probability that someone is dead or alive, voted Brexit or Remain etc. as a function of other variables (age, smoking etc.)

- In other words, we want a regression of the form:

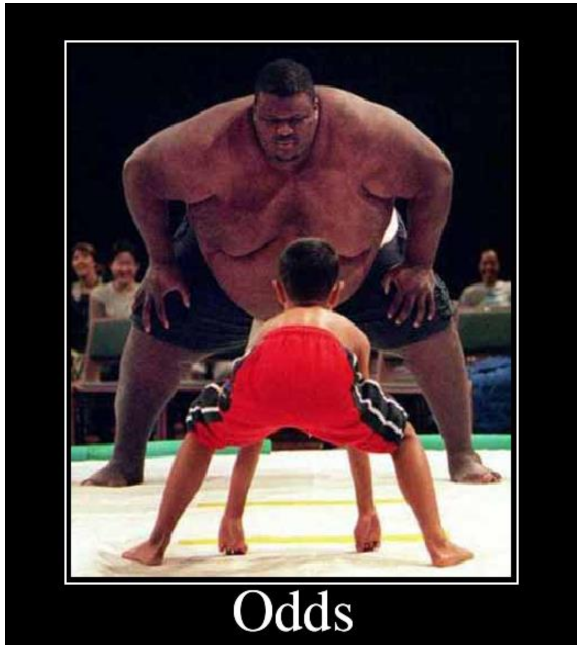
$$\text{probability of binary outcome} = a + b_1X_1 + b_2X_2 \ldots + b_nX_n = a + \Sigma b_iX_i$$

with
a = intercept
b_i = regression coefficients
X_i = independent variables (continuous or categorical)

Odds and log(odds)

- To understand logistic regressions, first we need to understand the concepts of *odds* and *odds ratios*
- Important: odds are not the same as the *probability* of the event!
- Gamblers know all about *odds of an event*:

$$\text{odds of event} = \frac{probability\ of\ event\ occurring}{probability\ of\ event\ not\ occurring}$$

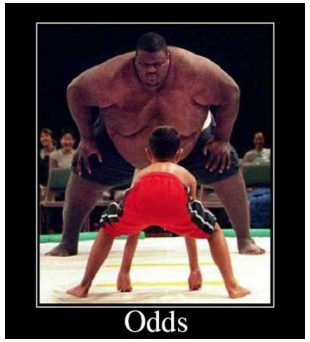


Odds and log(odds)

- Example: what is the *probability* of your birthday falling on a weekday this year?
  - probability of weekday=5/7=0.71 = p

Odds of a weekday = $\dfrac{\text{probability of weekday}}{\text{probability of weekend day}}$

  - odds of weekday = (5/7) / (2/7) = 5/2 = 2.5 = p/(1-p)
  - ln(odds of weekday) = log(2.5) = 0.91 = log(p/(1-p))



- And the probability of non-event, i.e. weekend day?
  - probability of weekend day = 2/7=0.29 =1-p
  - odds of weekend day = 2/5 =0.4 =(1-p)/p
  - ln(odds of weekend day)= –0.91 = ln((1-p)/p)

Exercises

Calculate:

- Tossing a fair coin:
    - Probability of heads?
    - Odds of heads?
    - Odds of tails?
    - Ln(odds of heads)

- Now throwing a die:
    - Probability of 1?
    - Odds of 1?
    - Odds of *not 1*?
    - Ln(odds of 1)?

Odds ratio

- Now imagine you have to choose between betting on coins (bet on ‘heads’) or dice (bet on ‘1’); which is best?
  - odds of heads = 1/1 = 1
  - odds of a 1 = 1/5 =0.2

- So it is easier to win a coin toss; how much easier?
- We can calculate the odds ratio of success in coins vs. dice

- Odds ratio = $\frac{odds\ of\ heads}{odds\ of\ a\ 1} = \frac{1}{0.2} = 5$

- This means you are 5 times more likely to win if you are tossing a coin than throwing a die

Notes

So far we concluded that:

- probability p is always between 0 and 1
- odds and odds ratio: from 0 to $+\infty$
- ln(odds) and ln(odds ratio): $-\infty$ to $+\infty$

Odd and probabilities

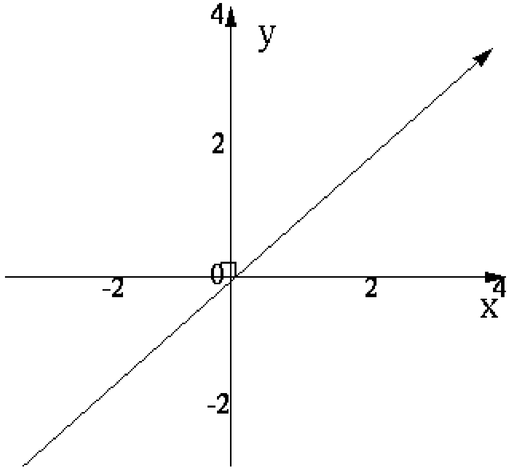
- If odds = p/(1-p), then:

- p = odds(1-p)
- p = odds – odds.p
- p + odds.p = odds
- p(1 + odds) = odds
- p = odds/(1 + odds)
- p = $\frac{1}{1+\frac{1}{odds}}$

Logistic function

- Back to logistic regression: we want to use a regression model to calculate probability of binary events (dead/alive, head/tail etc.) from a set of predictors:

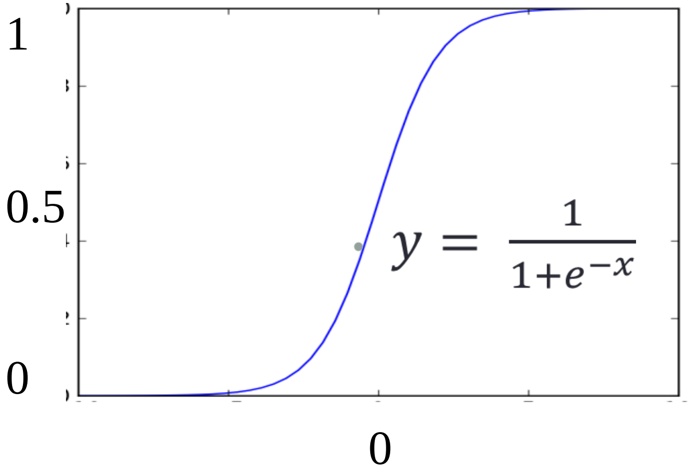
  $y = a + b_1X_1 + b_2X_2 \ldots + b_nX_n = a + \Sigma b_iX_i$



- Problem:
  - linear regression predicts y between $-\infty$ and $+\infty$
  - but probability is always between 0 and 1

- Solution:
  - we want our probabilities to be estimated by a model such as the logistic function
  - Why? Because whatever x, it will always return a value between 0 and 1

$$y = \frac{1}{1+e^{-x}}$$



Link function: Logit

- We need a link between the linear regression $a+\Sigma b_i X_i$ and logistic function $y = \frac{1}{1+e^{-f}}$:

$a+\Sigma b_i X_i \rightarrow$ link f $\rightarrow$ *prob* $p = \frac{1}{1+e^{-f}}$

- Therefore, we need to find the link function *f* that satisfies the condition:

$$p = \frac{1}{1+e^{-f}} = p = \frac{1}{1+\frac{1}{e^f}}$$

But $$p = \frac{1}{1+\frac{1}{odds}}$$

- Therefore e^f = odds; or *f* = log(odds)

- The link function we need is called **logit p** and is:

  f =logit p = log(odds of event) = $\log(\frac{p}{1-p})$

another derivation:

- If we want $p = \frac{1}{1+e^{-f}}$, then:
- $p = \frac{e^f}{e^f+1}$
- $p(e^f+1) = e^f$
- $pe^f + p = e^f$
- $p = e^f - pe^f$
- $p = e^f(1-p)$
- $e^f = \frac{p}{1-p}$
- $\log(e^f) = \log(\frac{p}{1-p})$
- $\boldsymbol{f = \log(\frac{p}{1-p})}$
- note: logit is always natural log (i.e. log on base e=2.71)

Logistic regression

- Logit function provides the link between predictors X_i and an event with probability p

- The *logistic regression model* is thus

$$a + \Sigma b_i X_i = \text{link function } f = \text{logit } p = \log\left(\frac{p}{1-p}\right) = \log(\text{odds of event})$$

- and probability p of event:

$$p = \frac{1}{1+e^{-logit}} = \frac{1}{1+e^{-(a+\Sigma bX)}} = \frac{1}{1+e^{-\log(odds)}} = \frac{1}{1+odds^{-1}} =$$

Fitting logistic regression

- The parameters a and b_i are estimated by MML (method of maximum likelihood), not by least squares
  - (we can't expand on MML in this course)

- For this reason, statistical significance or goodness of fit are based not on minimising variance, but on measures of 'deviance' between observed and predicted values
  - i.e. a comparison between right and wrong predictions of individual cases
  - remember: in logistic regressions, y is binary (yes/no)

- But as in linear regression, estimated parameters (coefficients, intercept) have a P-value that determines their significance
  - significance test based on a z-distribution similar to t and normal distributions
  - interpreted just like t-tests or F-tests. i.e. parameter is significant if $P<0.05$; 95% confidence intervals are provided etc.

Logistic regression: categorical variable

Example: let's say we want to test the effect of smoking (x, binary, yes or no) on hypertension (y, also binary, yes or no)

- Y=0: no hypertension; Y=1: hypertension
- X=0: non-smoker (baseline group); X=1: smoker (exposure group)

- Important: logistic regression model is:

logit p = log(odds of outcome happening) = a + bX

In baseline group, X=0; Therefore

- **Intercept a = log(odds of outcome happening when X=0)**

=Baseline or reference level

If I exponentiate a or log(odds), I get the odds

- e^a = (p/1 – p) = the odds of hypertension for non-smokers
- $p = \frac{1}{1+e^{-a}} = \frac{odds\ of\ non\!-\!smokers}{1+odds\ of\ non\!-\!smokers}$ = probability of hypertension for non-smokers

- Those are the ***baseline values***, i.e. the odds and probabilities for groups without exposure (when all X_i=0, i.e. even if nobody smoked)

Logistic regression: categorical variable

- Now the odds for smokers:

  - logit = $\ln(\frac{p}{1-p})$ = a + bX = a + b.1 = a + b

a + b = log(odds of hypertension for smokers)

$e^{a+b} = e^a e^b$ = the odds of hypertension for smokers

$p = \frac{1}{1+e^{-(a+b)}} = \frac{odds\ of\ smokers}{1+odds\ of\ smokers}$ = probability of hypertension for smokers

Those are the results for the ***exposure group*** (smokers)

Important: b=log(odds ratio)

If $\quad odds(non\text{-}smokers) = e^a$

$odds(smokers) = e^{a+b} = e^a e^b$

then $\quad odds(smokers)/odds(non\text{-}smokers)= e^a e^{b/} e^a = e^b$

$log(odds(smokers)/odds(smokers)) = log(e^b)=b$

- **The coefficient *b* in the logistic regression is the log(odds of hypertension in exposure group *relative to baseline*)**
  - In logistic regression, we test for significance of coefficient *b* (as in linear regression, where regression test is the slope test)
    - for a significant effect of variable, we need *b* different from 0 (i.e. P value < 0.05)
  - If b=0
    - odds ratio for exposure vs. baseline = $e^b = e^0 = 1$
    - = the odds are the same for exposure and baseline, i.e. the variable has no effect on output probability

Odds ratio

- Let’s add some hypothetical numbers to the example:
  - odds of hypertension for smokers =0.3 = 30%
  - odds of hypertension for non-smokers =0.1 = 10%
- This means that the odds of hypertension in smokers are three times higher in smokers
  - ***odds ratio*** = odds smokers/odds non smokers = 3
- The ***odds ratio of the two groups (exposure/baseline)*** is a very useful representation of the effect of a factor on the occurrence of event
- Logistic regression always reports odds of event in exposure group relative to baseline
  - more precisely, as ***log(odds ratio of event in exposure vs. baseline)***
  - So in the example above, it would give us *log(3)* as the result

Example 1: hypertension, smoking, obesity

- File *hypertension* presents data on people with or without hypertension as a function of two factors: smoking and obesity

- Cases coded as ‘yes’ or ‘no’
  - ‘no’ comes first alphabetically and is read as baseline
  - alternatively: ‘no’=0, ‘yes’=1 (don’t use 1 or 2!!!)

  - In this example, data are presented as a table
    - (we’ll see a different way of presenting data with each case as a line)

>hypertension

	smoking	obesity	total	hyper	nonhyper
1	no	no	247	40	207
2	yes	no	102	15	87
3	no	yes	59	16	43
4	yes	yes	25	8	17

Example 1: hypertension, smoking, obesity

- When data are presented as table
  - matrix has to be created from file
  - we have to create a matrix with two columns: number of positives or event occurrences (hypertension) and negatives (no hypertension)
  - this has been done already (file *hypnonhyp*)
    - i.e. the dependent variable will be the matrix *hypnonhyp*

	hyper	nonhyper
1	40	207
2	15	87
3	16	43
4	8	17

Running model

```
> model.hyper <- glm(hypnonhyp ~ smoking+obesity, binomial)
> summary(model.hyper)
Call:
glm(formula = hypnonhyp ~ smoking + obesity, family = binomial)
Deviance Residuals:
      1          2          3          4
 0.1593  -0.2520  -0.2653   0.4018
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.67143    0.16731  -9.990  < 2e-16 ***
smokingyes  -0.01654    0.27617  -0.060  0.95224
obesityyes   0.76005    0.28270   2.689  0.00718 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 7.15022  on 3  degrees of freedom
Residual deviance: 0.32067  on 1  degrees of freedom
AIC: 23.935
Number of Fisher Scoring iterations: 3
```

- Logistic regression is an example of generalised linear model
  - function *glm*
- Logistic model written like a multiple regression with *two* predictors:
  - *hypnonhyp ~ smoking+ obesity*
  - (ps. interactions later)
- Argument *binomial* sets logistic regression
  - Never forget to add binomial! Otherwise it fits a Gaussian rather than the logistic function!!!

Residuals

```
> model.hyper <- glm(hypnonhyp ~ smoking+obesity, binomial)
> summary(model.hyper)
Call:
glm(formula = hypnonhyp ~ smoking + obesity, family = binomial)
Deviance Residuals:
      1        2        3        4
 0.1593  -0.2520  -0.2653   0.4018
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.67143    0.16731  -9.990  < 2e-16 ***
smokingyes  -0.01654    0.27617  -0.060  0.95224
obesityyes   0.76005    0.28270   2.689  0.00718 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 7.15022  on 3  degrees of freedom
Residual deviance: 0.32067  on 1  degrees of freedom
AIC: 23.935
Number of Fisher Scoring iterations: 3
```

- Residuals are given as deviance (not variance)
  - difference between observed and predicted logit values in each group (no/no, no/yes, yes/no, yes/yes)
  - residuals in logit scale (neither probability nor cell count)

Intercept

```
> model.hyper <- glm(hypnonhyp ~ smoking+obesity, binomial)
> summary(model.hyper)
Call:
glm(formula = hypnonhyp ~ smoking + obesity, family = binomial)
Deviance Residuals:
      1         2         3         4
 0.1593  -0.2520  -0.2653   0.4018
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.67143   0.16731  -9.990  < 2e-16 ***
smokingyes  -0.01654   0.27617  -0.060  0.95224
obesityyes   0.76005   0.28270   2.689  0.00718 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 7.15022  on 3  degrees of freedom
Residual deviance: 0.32067  on 1  degrees of freedom
AIC: 23.935
Number of Fisher Scoring iterations: 3
```

- Intercept a = -1.67
- a=ln(odds of hypertension, baseline group)
  - =non-smokers, non-obese
  - e^a =the odds of hypertension if you're non-smoker, non-obese
  - =0.188=18.8%
- z-test: intercept is significantly different from 0
  - odds of hypertension (e^a)= not 1
  - probability of hypertension different from 0.5 in the sample

Effect of smoking

```
> model.hyper <- glm(hypnonhyp ~ smoking+obesity, binomial)
> summary(model.hyper)
Call:
glm(formula = hypnonhyp ~ smoking + obesity, family = binomial)
Deviance Residuals:
      1        2        3        4
 0.1593  -0.2520  -0.2653   0.4018
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -1.67143    0.16731  -9.990  < 2e-16 ***
smokingyes    -0.01654    0.27617  -0.060  0.95224
obesityyes     0.76005    0.28270   2.689  0.00718 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 7.15022  on 3  degrees of freedom
Residual deviance: 0.32067  on 1  degrees of freedom
AIC: 23.935
Number of Fisher Scoring iterations: 3
```

- Regression coefficient for smoking:
  - smokers (X=1) are shown as *smokingyes*,
    - i.e. variable name plus group ('yes')
  - b=log(odds ratio)=-0.0165
  - =log odds of hypertension for smokers relative to non-smokers
- But P(z) = 0.95!
  - b is not significantly different from 0
  - odds ratio not different from e^0 =1
- So smokers are not more likely to have hypertension than non-smokers ***in this sample***

Effect of obesity

```
> model.hyper <- glm(hypnonhyp ~ smoking+obesity, binomial)
> summary(model.hyper)
Call:
glm(formula = hypnonhyp ~ smoking + obesity, family = binomial)
Deviance Residuals:
      1        2        3        4
 0.1593  -0.2520  -0.2653   0.4018
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.67143    0.16731  -9.990  < 2e-16 ***
smokingyes   -0.01654    0.27617  -0.060  0.95224
obesityyes    0.76005    0.28270   2.689  0.00718 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 7.15022  on 3  degrees of freedom
Residual deviance: 0.32067  on 1  degrees of freedom
AIC: 23.935
Number of Fisher Scoring iterations: 3
```

- Regression coefficient for obesity: b=0.76
  - =log odds of hypertension for obese relative to non-obese
- P(z) = 0.00718
  - b is significantly different from 0
  - b = ln(odds of hypertension in obese relative to baseline) > 0
  - odds ratio= $e^{0.76}$ =2.14
    - odds ratio >1; obese at higher risk!
- So obesity more than doubles odds of hypertension ***in this sample***

Goodness of fit

```
> model.hyper <- glm(hypnonhyp ~ smoking+obesity, binomial)
> summary(model.hyper)
Call:
glm(formula = hypnonhyp ~ smoking + obesity, family = binomial)
Deviance Residuals:
      1        2        3        4
 0.1593  -0.2520  -0.2653   0.4018
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.67143    0.16731  -9.990  < 2e-16 ***
smokingyes   -0.01654    0.27617  -0.060  0.95224
obesityyes    0.76005    0.28270   2.689  0.00718 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 7.15022  on 3  degrees of freedom
Residual deviance: 0.32067  on 1  degrees of freedom
AIC: 23.935
Number of Fisher Scoring iterations: 3
```

- MML does not use variance to measure goodness of fit
  - it includes no 'dispersion parameter', which has to be taken as 1
- In MML, deviance replaces variance
  - null deviance = deviance when model includes only intercept (i.e. before predictors *smoking* and *obesity*)
  - Residual deviance is unexplained deviance after predictors
  - So difference between null and residual is the contribution of predictors to model

Goodness of fit

```
> model.hyper <- glm(hypnonhyp ~ smoking+obesity, binomial)
> summary(model.hyper)
Call:
glm(formula = hypnonhyp ~ smoking + obesity, family = binomial)
Deviance Residuals:
      1        2        3        4
 0.1593  -0.2520  -0.2653   0.4018
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.67143    0.16731  -9.990  < 2e-16 ***
smokingyes  -0.01654    0.27617  -0.060  0.95224
obesityyes   0.76005    0.28270   2.689  0.00718 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 7.15022  on 3  degrees of freedom
Residual deviance: 0.32067  on 1  degrees of freedom
AIC: 23.935
Number of Fisher Scoring iterations: 3
```

- Because there is no variance, goodness of fit is not measured by R^2
  - we use AIC (Akaike Information Criterion) instead
- Remember: adding additional predictors to regression may increase goodness of fit even when predictor is not significant
- AIC measures goodness of fit while punishing models for use of additional predictors
  - *the better and more parsimonious the model, the lower the AIC*
- Models with lowest AIC are selected

Guide to calculations:

- Look at a = log(baseline odds)
- exp(a) = baseline odds of event
- Probability in baseline: baseline odds/(baseline odds+1)

Then

- Look at b = log(odds ratio); if b is significant:
- exp(b) = odds ratio
- exp(a+b) = exp(a)*exp(b) = odds(baseline)*odds ratio = exposure odds
- Probability in exposure group = exposure odds/(exposure odds + 1)

Exercises

- Since *smoking* is not significant, you must optimise the model by excluding *smoking*, and run model only with variable *obesity* (manually, or with *step* function)

1. Is a significant? What does that mean?
2. Is b significant? What does that mean?

- Calculate:

3. Baseline odds of hypertension
4. Odds ratio of hypertension (obese vs. non-obese)
5. Odds of hypertension in obese
6. Probability of hypertension in non-obese
7. Probability of hypertension in obese