Regression Modelling for Biostatistics 1

Schlub T, Heritier S, Teixeira-Pinto A

2024-02-19

Table of contents

Preface

The following chapters include notes, videos, R and Stata code, required readings, and exercises for the BCA unit RM1 (Regression Modelling for Biostatistics 1).

These pages were generated with Quarto https://quarto.org/. On the left menu you have the topics that correspond roughly to the weekly modules. On the right side you should see the subtopics in the current topic.

Make sure that you have access to **Regression Methods in Biostatistics** book by *Vittinghoff et al.* You should be able to obtain a digital copy of the book from the library of your University.

I will be releasing more chapters throughout the semester.

Please let me know of any typos or bugs so I can update the notes.

1 Simple Linear Regression

Learning objectives

By the end of this week you should be able to:

1. Describe the different motivations for regression modelling
2. Formulate a simple linear regression model
3. Understand the least squares method of parameter estimation and its equivalence to maximum likelihood
4. Interpret statistical output for a simple linear regression model
5. Calculate and interpret confidence intervals and prediction intervals for simple linear regression

Learning activities

This week's learning activities include:

Learning Activity	Learning objectives
Preparation for week 2	
Video 1	1, 2, 3
Readings	1, 2, 3, 4
Video 2	4, 5
Independent exercises	2, 4, 5
Live tutorial/discussion	2, 4, 5

Preperation for week 2

In week 2 you will be required to collaboratively complete some exercises. To do this, in week 1 you will be allocated into groups of 3-4 and you are encouraged to meet with your group in week 2 by zoom at a mutually beneficial time. Each group has their own discussion board, which you can use to help organise a meet up time. Interacting, discussing, and working through problems with your peers is an important skill for any biostatistician. This is also nice activity to get to know your peers in this online course.

Introduction to regression

This lecture introduces you to the purpose of regression models to answer three types of research questions: prediction; isolating the effect of a single predictor; and understanding multiple predictors. You will also learn what a simple linear regression looks like and learn about the method used to estimate it's parameters.

Book Chapter 1. Introduction to section 1.3.3 (pages 1-4).

This reading supplements lecture 1 with a similar motivation for the need for regression models (which they refer to as multipredictor regression models) to answer three types of research questions: prediction; isolating the effect of a single predictor; and understanding multiple predictors. Nothing new is introduced in this reading, so its purpose is to allow you to become familiar with the writing style of the textbook that we follow in this course.

Book Chapter 3. Section 3.3 to 3.3.3 (pages 35-38).

This reading introduces the simple linear regression model and describes how to interpret each parameter of the model. This will be further explored in lecture 2. It also describes the error term between individual observations and the mean behaviour of the population – which is important as the assumptions of linear regression are all about the error term. Stata and R code corresponding to the output in this reading can be found below

Stata code and output

```
use hersdata, clear
set seed 90896
sample 10
reg SBP age
## (2,487 observations deleted)
##
```

```
##
##       Source |       SS           df       MS      Number of obs   =       276
## -------------+----------------------------------   F(1, 274)       =     11.70
##        Model |  4595.93381         1  4595.93381   Prob > F        =    0.0007
##     Residual |  107671.294       274  392.960929   R-squared       =    0.0409
## -------------+----------------------------------   Adj R-squared   =    0.0374
##        Total |  112267.228       275  408.244466   Root MSE        =    19.823
##
## ------------------------------------------------------------------------------
##          SBP | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##          age |   .6411057   .1874638     3.42   0.001     .2720533    1.010158
##        _cons |   93.87961   12.43407     7.55   0.000     69.40115    118.3581
## ------------------------------------------------------------------------------
```

R code and output

```
hers_subset <- read.csv("hers_subset.csv")
lm.hers <- lm(SBP ~ age, data = hers_subset)
summary(lm.hers)
##
## Call:
## lm(formula = SBP ~ age, data = hers_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.193 -14.346  -1.578  13.391  57.961
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  93.8796    12.4341    7.55 6.48e-13 ***
## age           0.6411     0.1875    3.42 0.000722 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.82 on 274 degrees of freedom
## Multiple R-squared:  0.04094,    Adjusted R-squared:  0.03744
## F-statistic:  11.7 on 1 and 274 DF,  p-value: 0.0007219
confint(lm.hers)
##                  2.5 %     97.5 %
## (Intercept) 69.4011476 118.358063
## age          0.2720533   1.010158
```

> **Note**
>
> This table does not use the complete HERS dataset, rather it takes a random sample of 10% of the data. In Stata this is achieved by using `set seed 90896` and `sample 10`. Here the `set seed 90896` ensures that the random sample is reproducible. i.e. we draw the same random sample each time. As random sampling is hard to replicate across statistical programs, to get the same output in R we needed to take the random sample in Stata and then import this subset of the data into R. A copy of this subset is provided in the data resources titled `hers_subset.csv`

Notation

Before continuing further with the theory of linear regression it is helpful to see some of the variations in notation around regression formula. In general greek letters are used for true population values, whereas the latin (or modern) alphabet is used to denote estimated values from a sample. The hat symbol ($\widehat{\ }$) can also be used to indicated estimated or fitted values. Subscripts on the Y's and x's indicate the observation number. Some examples of the different ways regression notation is used in this course is shown below. Don't worry if some of these terms are not familiar to you yet, they will be introduced to you in due course.

Term	True population	Estimated from data/sample
Regression line	$\mathrm{E}(Y) = \beta_0 + \beta_1 x Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$	$\bar{Y_x} = \hat{\beta_0} + \hat{\beta_1} x$ $\bar{Y_x} = b_0 + b_1 x$ $\hat{Y_i} = b_0 + b_1 x_i + e_i$
Expected values / means	$\mathrm{E}(Y)$ $\mathrm{E}(x)$	$\bar{Y}$ $\bar{x}$
Parameters, regression coefficients	β	$\hat{\beta}$, b
Error terms	ε - called "error"	e or $\hat{\varepsilon}$ called "residual" or "residual error"
Variance of error	σ^2, $\mathrm{Var}(\varepsilon)$	Mean square error, MSE, $\hat{\sigma}^2$, $\hat{\mathrm{Var}}(\varepsilon)$, s^2

Properties of ordinary least squares

There are many ways to fit a straight line to data in a scatterplot. Linear regression uses the principle of *ordinary least squares*, which finds the values of the parameters (β_0 and β_1) of the

regression line that minimise the sum of the squared *vertical* deviations of each point from the fitted line. That is, the line that minimises:

$$\sum (Y_i - \bar{Y}_i)^2 = \sum (Y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$$

This principle is illustrated in the diagram below, where a line is shown passing near the value of Y for six values of x. Each choice of values for $\hat{\beta}_0$ and $\hat{\beta}_1$ would define a different line resulting in different values for the vertical deviations. There is however one pair of parameter values that produces the least possible value of the sum of the squared deviations called the least squares estimate.

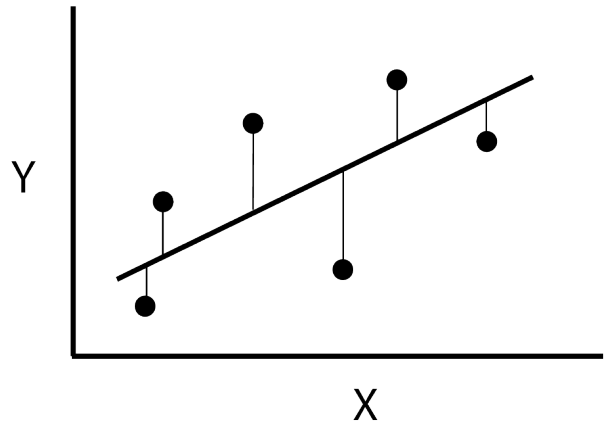


Figure 1.1: Ordinary least squares minimises the square of the vertical distance between data and the line

In the scatterplot below, you can see how the line adjusts to points in the graph. Try *dragging* some of the points, or creating new points by clicking in an empty area of the plot, and see how the equation changes. In particular, notice how moving up and down a point at the extremes of the x scale, affects the fitted line much more than doing the same to a point in the mid-range of the x scale. We will see later that this is the reason for caution when we have outliers in the data.

Linear regression uses least squares to estimate parameters because when regression assumptions are met (more on this later) the estimator is BLUE: *Best Linear Unbiased Estimator*. Unbiased means that over many repetitions of sampling and fitting a model, the estimated parameter values average out to equal the true "population" parameter value (i.e. $\mathrm{E}[\hat{\beta}_1] = \beta_1$). Unbiasedness does not mean that all parameter estimates are close to the true value—in fact it says nothing about the sample-to-sample variability in the parameter estimates, since that is a precision issue. The "linear" means that the class of estimators are those that can be written as linear combinations of the observations Y. More specifically, any linear unbiased estimator of the slope parameter β_1 can be written as $\sum_{i=1}^{n} a_i Y_i$ where the values of $(a_1, ...a_n)$ must be such that $\mathrm{E}(\sum_{i=1}^{n} a_i Y_i) = \beta_1$. This was important when computational power was

limited as linear estimators can be easily computed. The "best" component of BLUE says that least squares estimators are best in the sense of having the smallest variance of all linear unbiased estimators. That is, they have the best precision or they are the most efficient.

The mathematical theorem and proof that the least squares estimator is the best linear unbiased estimator (BLUE) is called the *Gauss-Markov theorem*. The least squares estimator also identical to the maximum likelihood estimator when the regression assumptions are met.

Chapter 3. Section 3.3.5 to 3.3.9 (pages 39-42).

The reading describes the basic properties of regression coefficients including: their standard error; hypothesis testing; confidence intervals; and their involvement in the calculation of R^2.

Regression in Stata and R

The lecture below (which you can watch for either Stata or R (or both if you are keen)) shows how to carry out and interpret the results of a simple linear regression in statistical software. It then shows how to calculate and interpret confidence and prediction intervals.

Stata Lecture

Download video

R Lecture

Download video

Exercises

The following exercise will allow you to test yourself against what you have learned so far. The solutions will be released at the end of the week.

Using the dataset hers_subset.csv dataset, use simple linear regression in R or Stata to measure the association between diastolic blood pressure (DBP - the outcome) and body mass index (BMI - the exposure).

a) Summarise the important findings by interpreting the relevant parameter values, associated P-values and confidence intervals, and R^2 value. Three to four sentences is usually enough here.

b) From your regression output, calculate by how much the mean DBP changes for a 5kgm^{-2} increase in BMI? Can you verify this by modifying your data and re-running your regression?

c) Manually calculate the β_1 standard error, the t-value, p-value and R^2

d) Based on your regression, make a prediction for the mean diastolic blood pressure of people with a BMI of 28kgm^{-2}.

e) Calculate and interpret a confidence interval for this prediction.

f) Calculate and interpret a prediction interval for this prediction.

Live tutorial and discussion

The final learning activity for this week is the live tutorial and discussion. This tutorial is an opportunity for you to to interact with your teachers, ask questions about the course, and learn about biostatistics in practice. You are expected to attend these tutorials when possible for you to do so. For those that cannot attend, the tutorial will be recorded and made available on Canvas. We hope to see you there!

Summary

This week's key concepts are:

1.
   - Regression models have three main purposes: prediction; isolating the effect of a single exposure; and understanding multiple predictors. The purpose of a regression model will influence the procedures you follow to build a regression model, and this will be explored more in week 8.
   - Simple linear regression measures the association between a continuous outcome and a single exposure. This exposure can be continuous, or binary (in which case simple linear regression is equivalent to a two-sample students t-test).
   - The relevant output to interpret and report from a simple linear regression includes:
     - The p-value for the exposure's regression coefficient (slope)
     - The effect size of the exposure regression coefficient and 95% confidence interval
     - The amount of variation in the outcome explained by the exposure (R^2)
   - The confidence interval for a prediction represents the uncertainty associated with an estimated predicted mean. Conversely, the prediction interval represents the uncertainty associated with the spread of observations around the predicted mean.

2 Checking Assumptions

Learning objectives

By the end of this week you should be able to:

1. Use residuals to test simple linear regression assumptions
2. Recognise when transformations of the response and/or covariate may be needed and be familiar with common transformations
3. Derive the interpretation of regression coefficients after transformation

Learning activities

This week's learning activities include:

Learning Activity	Learning objectives
Video 1	1
Readings	1, 2, 3
Collaborative exercise	1, 2

Collaborative exercise reminder

> **!** Important
>
> In week 1 you were allocated into small groups to discuss the exercise at the end of this week. You should have already organised a time with your group to meet up by zoom and either discuss your solutions to this exercise, or help and guide each other to complete the task.

Assumptions of linear regression

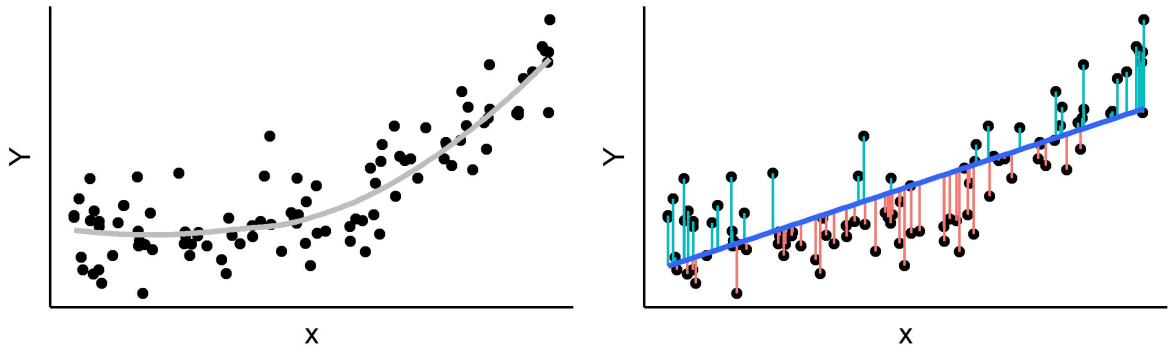
In this section we introduce some simple methods for assessing the underlying assumptions of the simple regression model. For regression analysis we should always perform a reality check that the model we are fitting is at least approximately appropriate for the problem at hand. The assumptions of linear regression fitted by ordinary least squares are:

1. Linearity of the relationship
2. Homoscedasticity (constant variance)
3. Normality of the residuals
4. Independence of observations.

As shown below, assumptions (1) - (3) can be defined in terms of the random part of the regression model - the residuals. The residuals are the observed minus the predicted Y's, i.e. $e_i = Y_i - \hat{Y}_i = Y_i - (b_0 + b_1 x_i)$. Assumption (4) however is usually an assessment of the study design rather than assessed with a diagnostic analysis of the data.

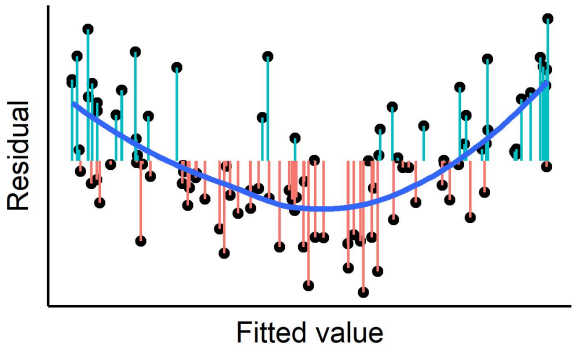
Linearity

This assumption requires a linear (straight) relationship between the outcome and predictor. The plot below shows the effect of a non-linear relationship on the residuals.



In the first figure, the relationship is curved upwards and so when fitting a straight line through the data, the outcome value (Y) is frequently above the regression line for small and large values of x, and below the regression line in the middle. Although this is reasonably obvious in the scatter plot above, it is helpful to plot the actual residual values instead of the outcome as per the plot below.

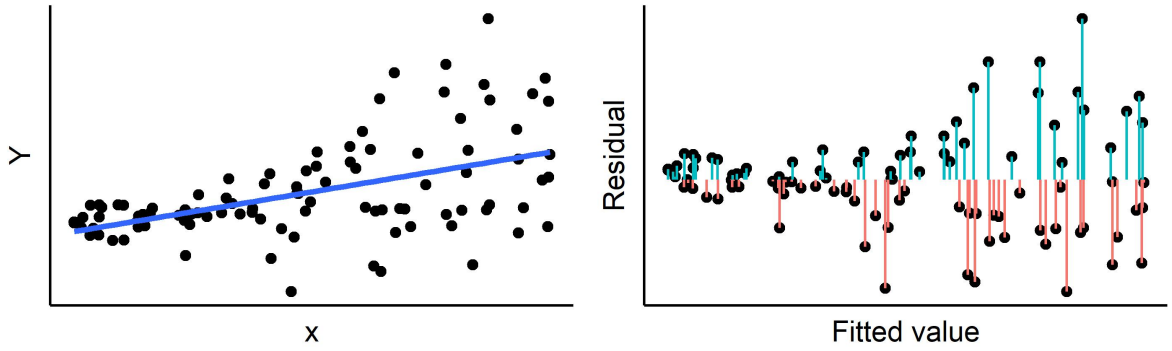
Note in this figure we plot the residuals against the *predicted* values of the regression line instead of the x-values themselves. When there is just one predictor variable, as the regression line is linear this is equivalent to plotting against x. The advantage to plotting against the



fitted value is that this extends naturally into multivariable models. From this *residual vs. fitted* plot, we can also see that the linearity assumption could be rewritten in terms of the residuals. E.g. the residuals have a mean of zero for all values of x.

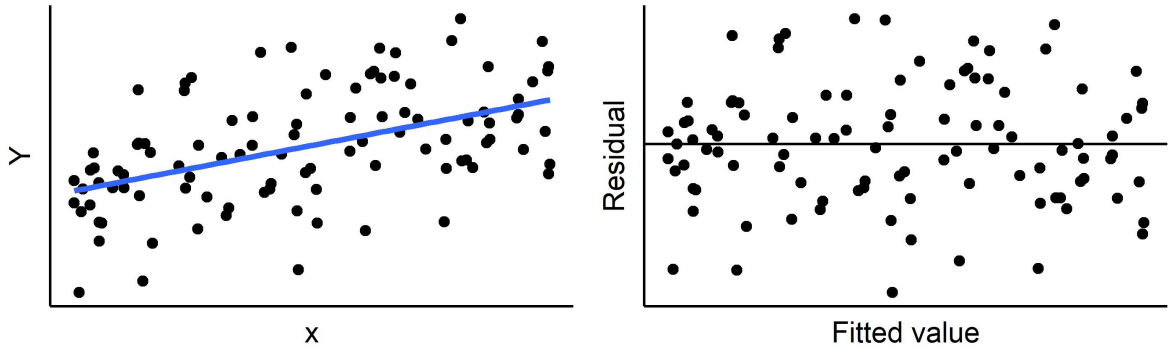
Homoscedasticity (constant variance)

This assumption requires that the variance around the regression line is constant for all values of x. That is, the residuals have *constant variance*. The "residual vs fitted" plot we looked at above is also useful for assessing violations of the homoscedasticity assumption, as illustrated in the scatter and residual vs fitted plots below.



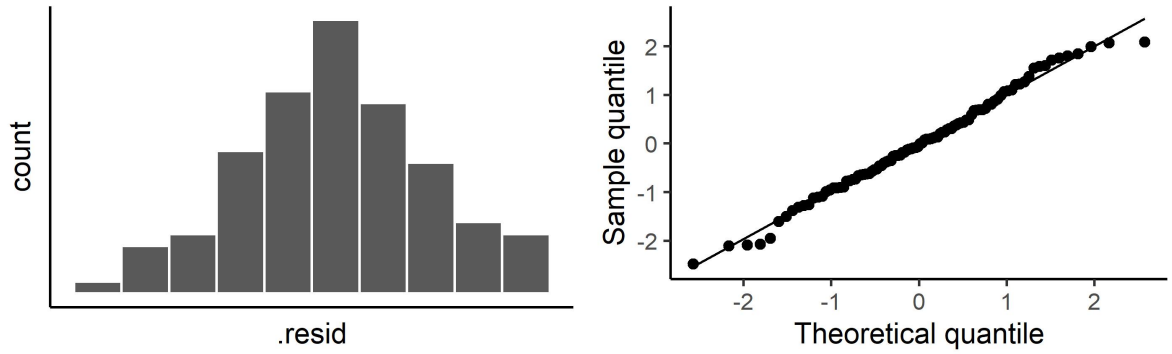
Here, we can see that as x increases, so does the variance of the residuals. This leads to a fanning out pattern in the residual vs fitted plot.

To contrast these plots, the below figures demonstrate what these plots look like when both the linearity and homoscedasticity assumptions are met.



Normalitiy

This assumption requires that the residuals be normally distributed. Note that this is an assumption about the distribution of the residuals, not about the distribution of Y or x. We can plot the residuals on a histogram or normal quantile plot to assess the normality of residuals assumption – which has clearly been met in the figures below.



A well-known formal method for assessing non-normality is the Shapiro-Wilk test. It is based approximately on the correlation between the observed and expected residuals. The null hypothesis is that the residuals are normally distributed, so small p-values indicate evidence against a normal distribution. However, for small samples this test has low power for realistic non-normal alternatives, so it is difficult to use a non-statistically-significant result as evidence to support an assumption of normality in a small sample. Conversely, for large samples, small but unimportant deviations from normality may be declared statistically. Therefore we do not recommend this (or other) formal statistical tests of normality. Rather a visual assessment of a histogram or normal quantile plot is recommended.

Worked example

Stata Lecture

Download video

R Lecture

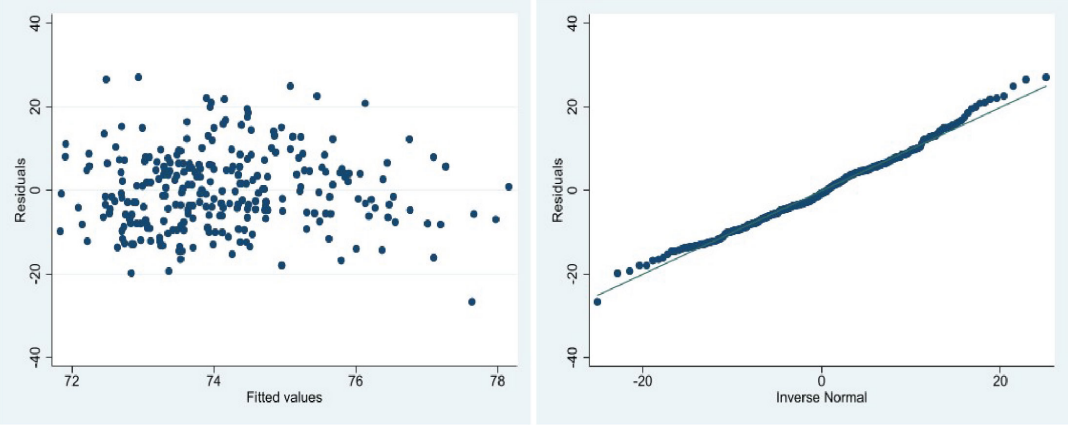
Download video

Let's check the assumptions of the regression the week 1 exercise where we use hers_subset.csv data to measure the association between diastolic blood pressure (DBP - the outcome) and body mass index (BMI - the exposure). The first plot shows the residual versus fitted plot. This shows an even scatter around zero with no fanning present for the majority of the plot. Towards the end of the plot, the residuals do seem to be a bit more negative than positive - but the sparsity of the data here makes this difficult to confirm. I would conclude here is no evidence against linearity or homoscedasticity and we can accept these assumptions as true. I would also like to see more data at higher fitted values levels to better confirm this. In the second plot, we see that the residuals are normally distributed as they follow a reasonably straight line on the normal quantile plot. Therefore we conclude the assumption of normality of residuals is true. Without further context of the study design that collected this data, we cannot draw any conclusions about the indepdence of observations - so we will assume this has been met. Therefore all the assumptions have been met and we conclude that a linear regression is appropriate for this data.

Stata code and output

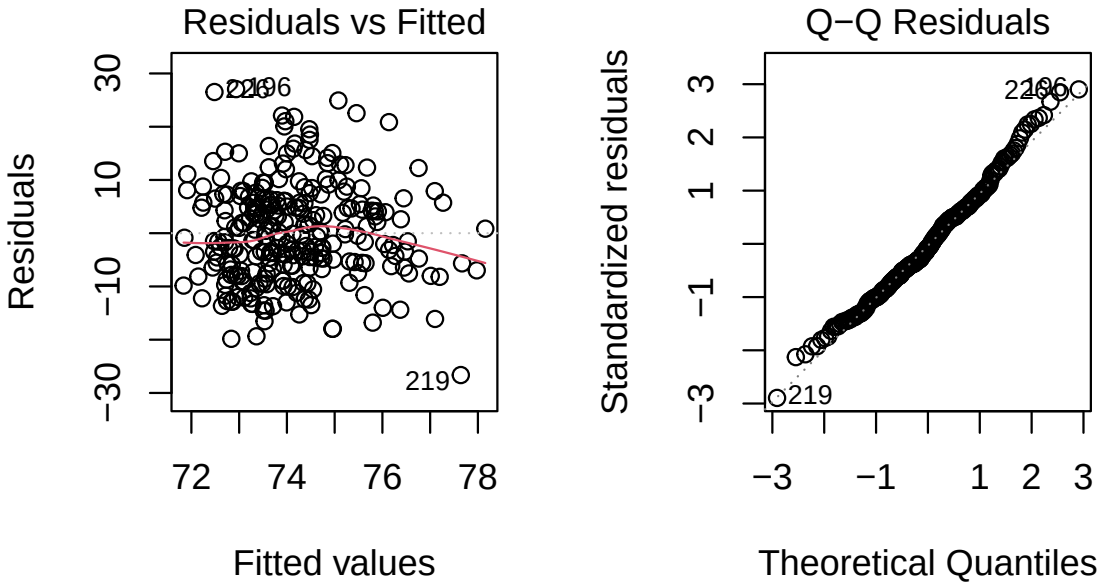
```
use hers_subset
reg DBP BMI
rvfplot /* the residual versus fitted plot */
predict res_std, residuals /* calculate the residuals*/
qnorm res_std /* The normal quantile plot of the residuals*/
##       Source |       SS           df       MS      Number of obs   =       276
## -------------+----------------------------------   F(1, 274)       =      4.84
##        Model |  423.883938         1  423.883938   Prob > F        =    0.0286
##     Residual |  23988.8842       274  87.5506722   R-squared       =    0.0174
## -------------+----------------------------------   Adj R-squared   =    0.0138
##        Total |  24412.7681       275  88.7737022   Root MSE        =    9.3569
##
## ------------------------------------------------------------------------------
##          DBP | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##          BMI |   .2221827   .1009756     2.20   0.029     .0233961    .4209693
##        _cons |   67.82592   2.923282    23.20   0.000     62.07097    73.58087
```

```
## ------------------------------------------------------------------------
```



R code and output

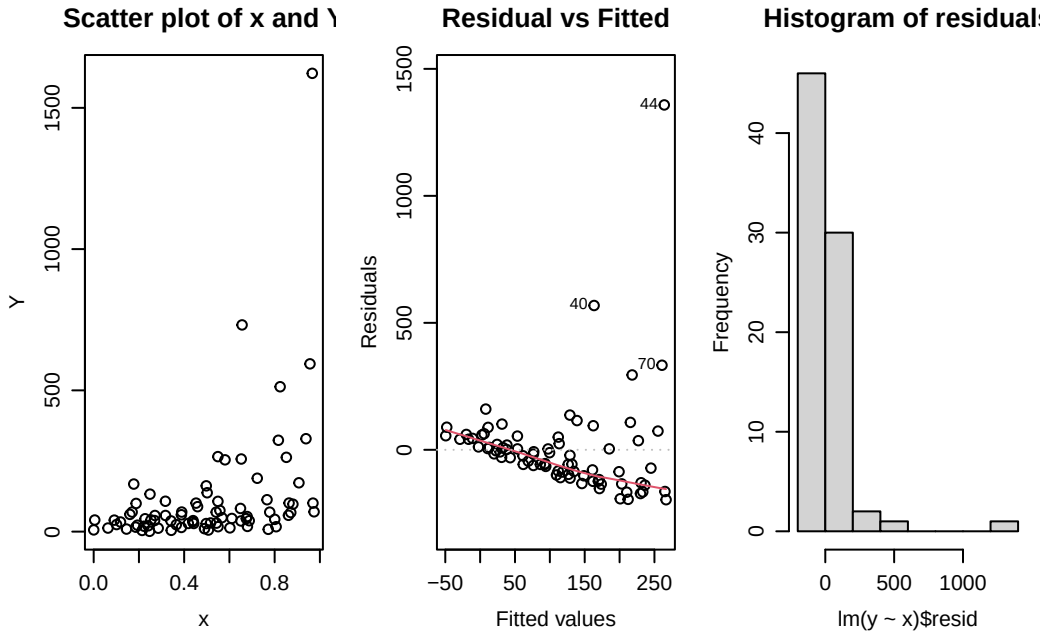
```
par(mfrow=c(1,2))
hers_subset <- read.csv("https://www.dropbox.com/s/t0ml83xesaaazd0/hers_subset.csv?dl=1")
lm.hers <- lm(DBP ~ BMI, data = hers_subset)
plot(lm.hers,1) # The residual versus fitted plot
plot(lm.hers,2) # The normal quantile plot
```



Note that in R, you can also generate these plots manually by calling the "residuals" or "fitted" property in the "lm.hers" object. e.g. `plot(lm.hers$fitted, lm.hers$residuals)`. These properties can be helpful if you wish to use the residuals in other ways.

Transformations

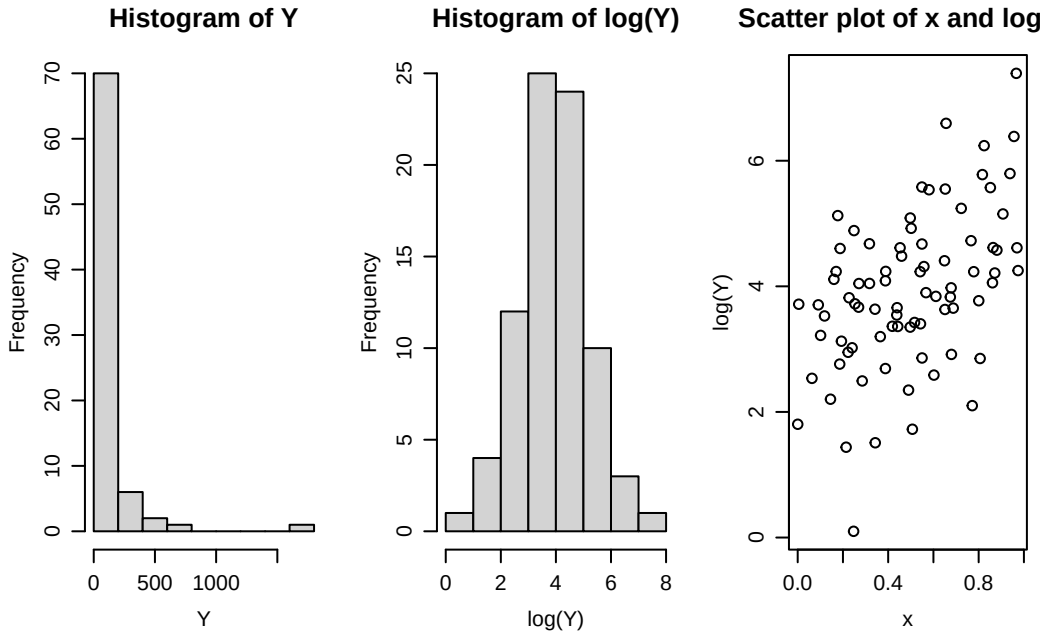
Transformations are one of many possible remedies to deal with an assumption violation. They are particularly helpful when either the outcome, a covariate, or both are strongly right skewed - as is the case for many measurements in clinical and health research (e.g. length of hospital stay, survival times, and serum measurements). Although this skew isn't of itself a problem (remember, the assumption is that the residuals are normally distributed, not the data itself), skewed data often results in violations to normality of residuals, homoscedasticity, and/or linearity. For example, consider the following plots.



We can immediately see a number of problems that would deter us from fitting a linear regression of Y on x. First, there is no clear linear relationship between Y and x. Secondly, the variability of the Y-values increases with x. Thirdly, the residuals of a linear regression are clearly non-normal, and heavily right-skewed. Indeed a histogram of Y shows that the outcome is heavily right skewed.

Log transformations of the outcome

The scatter plot above is an example where the skew in the outcome (Y) variable has caused problems in the linear regression. A common transformation to reduce skew is the *Log* transformation. Indeed log-transforming Y eliminates this skew, and produces a scatter plot where linear regression would be appropriate, but this time with $\log(Y)$ as the outcome variable.



The log transformation is a very natural tool for many biological measures because it converts multiplicative relationships (which characterise processes such as cell division and successive dilution of assays, for instance) into additive relationships - which by the central limit theorem are more likely to be normally distributed. For example on a logarithmic scale the values 0.1, 1 and 10 equals $log(0.1) = -2.30$, $log(1) = 0$ and $\log(10) = 2.30$ which are all equally spaced (with a difference of 2.3) as the original numbers (0.1, 1 and 10) all differ by a multiplicative factor of 10. Note that in this course "Log" is used to designate the natural logarithm. But logarithms to any base will work equally well as this corresponds simple to rescalling the logged values.

In the example above the variable Y is said to have a *log-normal* distribution as it is normally distributed after taking a log transformation. i.e. $\log(Y) \sim N(\mu, \sigma^2)$. For log-normally distributed variables, the geometric mean as defined by $GM(Y) = e^{\mathrm{E}[log(Y)]} = e^{\mu}$ is a more suitable summary statistic than the (arithmetic) mean we normally refer to. The geometric mean is also helpful for interpreting a linear regression when the outcome has been log-transformed as we see below.

The regression model is $\log Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$, where $\varepsilon_i \sim N(0, \sigma^2)$ and therefore $\mathrm{E}[\log Y_i] = \beta_0 + \beta_i x_i$. Exponentiating both sides of the above, and recalling that $\mathrm{GM}(Y) = e^{\mathrm{E}[\log Y]}$, we can obtain an interpretation of the parameter β_1 in terms of the geometric mean of Y.

Introducing the notation $\mathrm{GM}(Y|x)$ to denote the geometric mean of Y at a given value of the

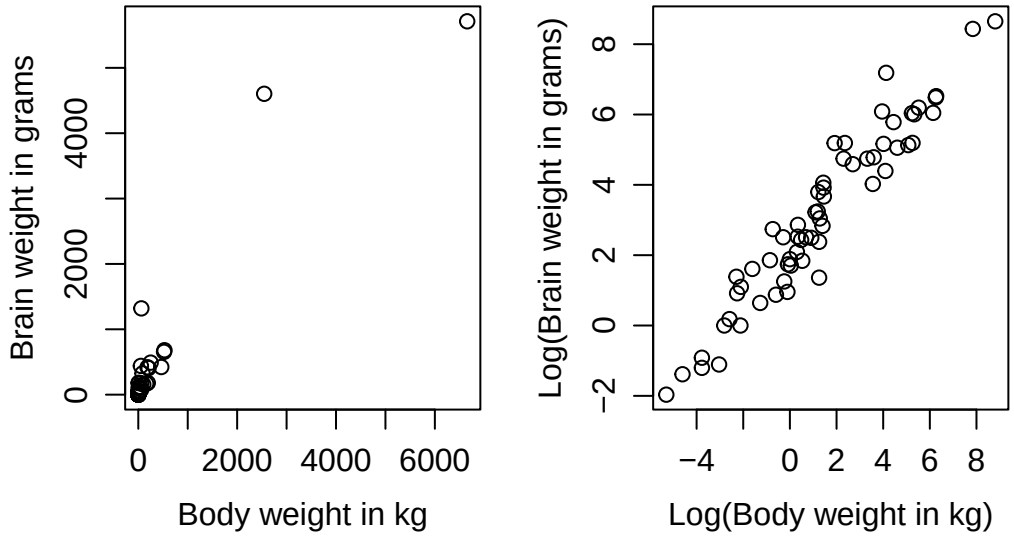
covariate x, we now examine the effect on $\mathrm{GM}(Y|x)$ for a change of 1 unit in x:

$$\frac{\mathrm{GM}(Y \mid x+1)}{\mathrm{GM}(Y \mid x)} = \frac{e^{\mathrm{E}[\log Y|x+1]}}{e^{\mathrm{E}[\log Y|x]}} = \frac{e^{\beta_0+\beta_1(x+1)}}{e^{\beta_0+\beta_1 x}} = \frac{e^{\beta_0}e^{\beta_1 x}e^{\beta_1}}{e^{\beta_0}e^{\beta_1 x}} = e^{\beta_1}.$$

Therefore increasing x by 1 is associated with a *proportionate* (multiplicative) increase of e^{β_1} in the geometric mean of Y.

Log transformations of the outcome and covariates

Consider the following scatterplot of brain weight (in grams) versus body weight (in kg) for 62 different mammal species (Allison and Cicchetti, *Science*, 1976).



As almost all the data are bunched up in the bottom left hand corner. However, a logarithmic transformation of both brain and body weight reins in the extreme values on both axes and produce a scatterplot worth pursuing with linear regression.

The regression model that would be fitted is $\log Y_i = \beta_0 + \beta_1 \log x_i + \varepsilon_i$. The parameter β_1 reflects the effect on $\mathrm{E}[\log Y]$ of additive changes in $\log(x)$, which correspond to multiplicative changes in x. Therefore let us examine the proportionate change in $\mathrm{GM}(Y)$ for a *doubling* of covariate x.

$$\frac{\mathrm{GM}(Y \mid 2x)}{\mathrm{GM}(Y \mid x)} = \frac{e^{\mathrm{E}[\log Y|2x]}}{e^{\mathrm{E}[\log Y|x]}} = \frac{e^{\beta_0+\beta_1 \log(2x)}}{e^{\beta_0+\beta_1 \log(x)}} = \frac{e^{\beta_0}(2x)^{\beta_1}}{e^{\beta_0}x^{\beta_1}} = 2^{\beta_1}.$$

The estimated slope for this regression is 0.752, and so $2^{0.752} = 1.68$. Therefore, we estimate the geometric mean brain weight to increase 1.68-fold or by 68% for each doubling in weight of mammals. The 95% confidence interval for 2^{β_1} is obtained from the 95% confidence interval for β_1 (say (L,U)) by raising 2 to the power of the endpoints, namely $(2^L, 2^U)$, producing (1.62, 1.75) for this example.

This example is a rather extreme case with respect to the distribution of the covariate and it is probably fair to say that transformations of x are not usually as important to consider as transformations of the outcome. Since the linear model takes the values of x as given, so there is no modelling of their distribution, there is no formal need to obtain a "nice" symmetric distribution of the independent variable. However, when we discuss the concept of leverage in week 3 we will see that highly skewed x distributions are undesirable because they may lead to just one or two observations having an unduly large influence on the estimated regression model.

Other methods for handling non-constant variance

Weighted least squares

Weighted least squares is an alternative linear regression fitting algorithm to ordinary least squares which is suitable when the variance is non-constant. Here, instead of minimising the sum of square error

$$\sum (Y_i - \bar{Y}_i)^2$$

we instead minimise the weighted sum of square error

$$\sum w_i (Y_i - \bar{Y}_i)^2$$

where $w_i = \frac{1}{\text{var}(Y_i)}$. That is, the error for each observation is scaled by it's predicted variance. When the predicted variance is unknown iterative procedures exist to progressively improve homoscedasticity in weighted linear regression.

Collaborative exercise

assumptions.csv consists of simulated data for you to practice checking the assumptions of simple linear regression. It contains 4 continuous outcome variables `y1`, `y2`, `y3` and `y4`. It has two explanatory variables `x1` and `x2`. Your task is to investigate the assumptions of any or all of the possible different simple linear regressions between the outcome variables and the explanatory variables. There are 8 possible regressions to check in total (4 choices for outcome variable, and 2 choices for explanatory variable. $4 \times 2 = 8$). You can either investigate all 8, or however many you have time for - but the more practice the better!

Discuss your analysis of the assumptions with your group in a self organised zoom meeting, or on your groups dedicated discussion board. If you need help with setting up a zoom meeting, please contact the unit coordinator for assistance.

Summary

This week's key concepts are:

1. Linear regression has four key assumptions that should be investigated:
    - Linearity - investigated with a residual versus fitted plot. An even scatter around zero indicates this assumption has been met
    - Homoscedasticity (constant variance) - investigated with a residual versus fitted plot. An even scatter around zero with no "fanning" indicates this assumption has been met
    - Normality of residuals - investigated with a normal quantile plot, or a histogram, of the residuals. For large sample sizes, moderate deviations to this assumption are ok.
    - Independence of observations. This is usually investigated by reviewing the study design. However sometimes there can be warning flags present in the data (e.g. a participant identifier column with duplicate entries).
2. Violations of any of the assumptions above will result in regression results that are biased, have incorrect interpretations, or have inflated type 1 errors. The greater the violation to the assumption, the greater the consequence
3. Log transformations are a useful tool that can often improve assumption violations. However, the interpretation of regression coefficients will change with a log transformation.

3 Binary Covariates, Outliers, and Influential Observations

Learning objectives

By the end of this week you should be able to:

1. Formulate, carry out and interpret a simple linear regression with a binary independent variable
2. Check for outliers and influential observations using measures and diagnostic plots
3. Understand the key principals of methods to deal with outliers and influential observations

Learning activities

This week's learning activities include:

Learning Activity	Learning objectives
Video 1	1
Notes&Readings	2, 3

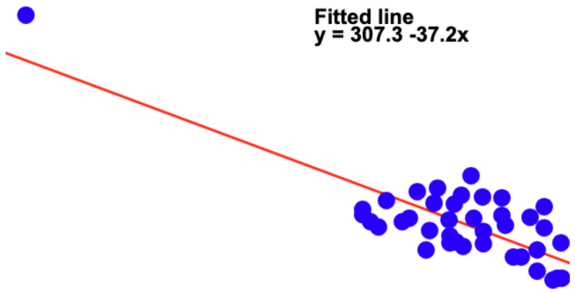
Binary covariates

So far we have limited our analysis to continuous independent variables (e.g. age). This week we are going to explore the situation where the independent variable (the x variable) is binary.

Outliers and influential observations

An outlier is a point with a large residual. Sometimes an outlier can have a large impact on the estimates of the regression parameter. If you move some of the points in the scatter so they become outliers (far from other points), you can see this will affect the estimated regression line. However, not all outliers are the same. Try moving up and down one of the points at the beginning or the end of the X scale. The impact in the regression line is much stronger than if you do the same with a point in the mid range of X.

Conversely, a data point may not have a large residual but still have an important influence in the estimated regression line. Below, you can see that the data point in the left does not appear to have a large residual but it strongly affects the regression line.



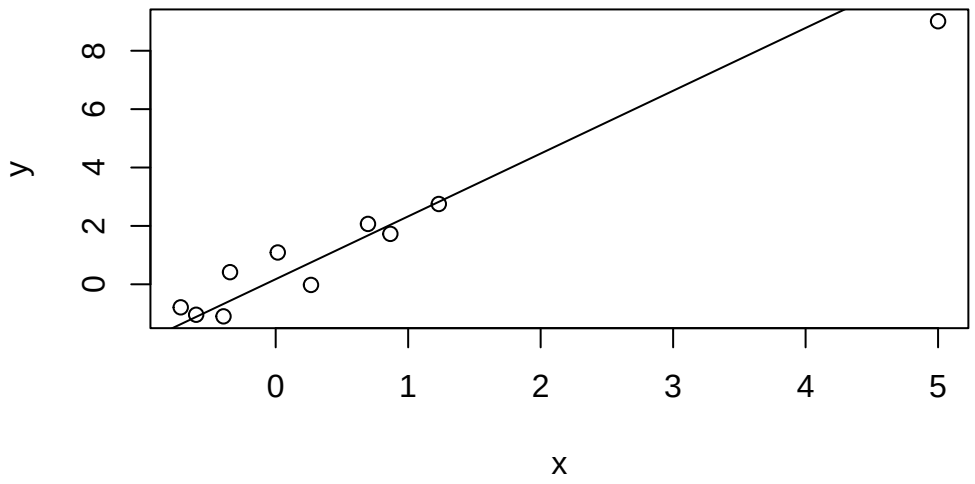
There are several statistics to measure and explore outliers and influential points. We will discuss some here:

Leverage: measure of how far away each value of the independent variable is from the others. Data points with high-leverage points, if any, are outliers with respect to the independent variables. The leverage is given by $\frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum (x_i - \bar{x})^2}$ and varies between 0 and 1.

Consider the simple example below with 10 simulated data points. The 10th value for X was chosen to be far from the others.

R code

```
set.seed(1011)
x<-rnorm(9)                #random 9 values
x[10]<-5                   #value far from the others
y<-rnorm(10,0.5+2*x,1)    #generate y

#plot the data
lmodel<-lm(y~x)            #fit the model
plot(x,y)                  #plot the data
abline(line(x,y))         # add the regression line
```



Stata code

```
clear
set seed 1011
set obs 10

generate x = rnormal()
replace x = 5 in 10
generate y = rnormal(0.5+2*x,1)

graph twoway (lfit y x) (scatter y x)
regress y x
## Number of observations (_N) was 0, now 10.
##
##
## (1 real change made)
##
##
##
##
##       Source |       SS           df       MS      Number of obs   =        10
## -------------+----------------------------------   F(1, 8)         =     52.69
```

```
##        Model |  86.4227998         1  86.4227998   Prob > F        =    0.0001
##     Residual |  13.1226538         8  1.64033173   R-squared       =    0.8682
## -------------+----------------------------------   Adj R-squared   =    0.8517
##        Total |  99.5454536         9   11.060606   Root MSE        =    1.2808
##
## ------------------------------------------------------------------------------
##            y | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##            x |    1.98254   .2731326     7.26   0.000     1.352695    2.612385
##        _cons |    .516953    .477044     1.08   0.310    -.5831126    1.617018
## ------------------------------------------------------------------------------
```

If we compute the leverage of each point, it is not surprising that x=5 has a high leverage.

R code

```
influence(lmodel)$hat     #leverage
##         1         2         3         4         5         6         7         8
## 0.1027407 0.1570337 0.1686229 0.1003533 0.1156210 0.1044403 0.1135604 0.1391403
##         9        10
## 0.1353478 0.8631394
1/10 + (x-mean(x))^2/(var(x)*9)  #leverage manually computed
##  [1] 0.1027407 0.1570337 0.1686229 0.1003533 0.1156210 0.1044403 0.1135604
##  [8] 0.1391403 0.1353478 0.8631394
```

Stata code

```
clear
set seed 1011
*generate the data
  set obs 10
  generate x = rnormal()
  replace x = 5 in 10
  generate y = rnormal(0.5+2*x,1)

  regress y x

*leverage
  predict lev, leverage
*leverage computed manually
  gen lev_manual =1/10 + (x- .9228745)^2/( 1.563043^2 *9)
```

```
  list
## Number of observations (_N) was 0, now 10.
##
##
## (1 real change made)
##
##
##
##       Source |       SS           df       MS      Number of obs   =        10
## -------------+----------------------------------   F(1, 8)         =     52.69
##        Model |  86.4227998         1  86.4227998   Prob > F        =    0.0001
##     Residual |  13.1226538         8  1.64033173   R-squared       =    0.8682
## -------------+----------------------------------   Adj R-squared   =    0.8517
##        Total |  99.5454536         9   11.060606   Root MSE        =    1.2808
##
## ------------------------------------------------------------------------------
##            y | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##            x |    1.98254   .2731326     7.26   0.000     1.352695    2.612385
##        _cons |    .516953    .477044     1.08   0.310    -.5831126    1.617018
## ------------------------------------------------------------------------------
##
##
##
##
##      +-------------------------------------------+
##      |         x           y        lev   lev_ma~l |
##      |-------------------------------------------|
##   1. |  .8897727    2.534658   .1000498   .1000498 |
##   2. |  .0675149    1.056028   .1332746   .1332746 |
##   3. |  .1199856    1.148383   .1293175   .1293175 |
##   4. |  .1817483    1.488953   .1249804   .1249804 |
##   5. |  1.257954    3.040046   .1051064   .1051064 |
##      |-------------------------------------------|
##   6. | -.2278177   -1.908616    .160219   .1602191 |
##   7. | -.1390651   -1.624345   .1512879   .1512879 |
##   8. |  .3885022    2.991632   .1129868   .1129868 |
##   9. |   1.69015    5.084781   .1267743   .1267743 |
##  10. |         5    9.654363   .8560032   .8560035 |
##      +-------------------------------------------+
```

DFBETA: We can also compute how the coefficients change if each observation is removed

from the data. This will produce a vector for β_0 and β_1 corresponding to n regressions fitted by deleting each observation at a time. The difference betweeen the full data estimates and the estimates by removing each data point is called *DFBETA*. In the example of the small simulated dataset set above, the dfbeta can also be obtained from the `influence()` function in r.

R code

```
influence(lmodel)$coefficients     #DFBETA
```

```
   (Intercept)            x
1  -0.015131790 -0.001678535
2  -0.053246101  0.019647712
3   0.017216660 -0.006821716
4   0.053369207  0.002038393
5   0.022861311  0.006672433
6  -0.101915091  0.012493753
7   0.090627673 -0.018400391
8  -0.109982670  0.034951829
9   0.093779788 -0.028593745
10  0.003248704 -0.126864091
```

```
#dfbeta(lmodel)                    #does the same thing

#computing the DFBETA manually for the 10th observation
coef(lm(y~x)) - coef(lm(y[-10]~x[-10]))
```

```
 (Intercept)            x
 0.003248704 -0.126864091
```

Stata code

```
*I am not aware of a function in Stata to compute
*the unstandardised version of DFBETA
*See below the code for the standardised DFBETA
```

```
Error: <text>:1:1: unexpected '*'
1: *
    ^
```

Note that the *DFBETA* above are computed in the original scale of the data. Thus, the magnitude of the difference is dependent on this scale. An alternative is to standardise the *DFBETA* using the standard errors. This will give as deviance from the original estimate in standard errors. A common cut-off for a very strong influence in the results is the value 2.

```
 #computes the standardised dfbeta.
#note that there is also a
#dfbeta() function that computes the non-stantandardised dfbeta
dfbetas(lmodel)
```

```
   (Intercept)           x
1  -0.07239559 -0.01366864
2  -0.26072268  0.16374799
3   0.08223075 -0.05545644
4   0.26788376  0.01741473
5   0.11021369  0.05475095
6  -0.57846582  0.12069940
7   0.48505258 -0.16762085
8  -0.60589936  0.32773222
9   0.49569526 -0.25724655
10  0.01568953 -1.04282523
```

Stata code

```
clear
set seed 1011
*generate the data
  set obs 10
  generate x = rnormal()
  replace x = 5 in 10
  generate y = rnormal(0.5+2*x,1)

  regress y x

  dfbeta
  list

## Number of observations (_N) was 0, now 10.
##
##
## (1 real change made)
##
```

```
##
##
##       Source |       SS           df       MS      Number of obs   =        10
## -------------+----------------------------------   F(1, 8)         =     52.69
##        Model |  86.4227998         1  86.4227998   Prob > F        =    0.0001
##     Residual |  13.1226538         8  1.64033173   R-squared       =    0.8682
## -------------+----------------------------------   Adj R-squared   =    0.8517
##        Total |  99.5454536         9   11.060606   Root MSE        =    1.2808
##
## ------------------------------------------------------------------------------
##            y | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##            x |    1.98254   .2731326     7.26   0.000     1.352695    2.612385
##        _cons |    .516953    .477044     1.08   0.310    -.5831126    1.617018
## ------------------------------------------------------------------------------
##
##
## Generating DFBETA variable ...
##
##     _dfbeta_1: DFBETA x
##
##
##      +-----------------------------------+
##      |         x           y   _dfbeta_1 |
##      |-----------------------------------|
##   1. |  .8897727    2.534658   -.0014574 |
##   2. |  .0675149    1.056028   -.0627431 |
##   3. |  .1199856    1.148383   -.0569127 |
##   4. |  .1817483    1.488953   -.0820419 |
##   5. |  1.257954    3.040046    .0017001 |
##      |-----------------------------------|
##   6. | -.2278177   -1.908616    .5239667 |
##   7. | -.1390651   -1.624345    .4384975 |
##   8. |  .3885022    2.991632   -.1846272 |
##   9. |   1.69015    5.084781    .1784969 |
##  10. |         5    9.654363   -4.140455 |
##      +-----------------------------------+
```

In the example above, the 10th observation seems to haves reasonable impact in the estimates.

Cook's distance - This is another measure of influence that combines the leverage of the data point and its residual. It summarizes how much all the values in the regression model

change when each observation is removed. It is computed as

$D_j = \frac{\sum(\hat{Y}_i - \hat{Y}_{(-j)})^2}{2\sigma^2}$

A general rule of thumb is that a point with a Cook's Distance above $4/n$ is considered to be an outlier.

R code

```
#the column cook.d is the Cook's distance
#note that this function also computes some of the measures discussed above
influence.measures(lmodel)
```

```
Influence measures of
     lm(formula = y ~ x) :

    dfb.1_   dfb.x   dffit cov.r  cook.d   hat inf
1  -0.0724 -0.0137 -0.0837 1.431 0.00397 0.103
2  -0.2607  0.1637 -0.2717 1.388 0.03993 0.157
3   0.0822 -0.0555  0.0869 1.554 0.00430 0.169
4   0.2679  0.0174  0.2935 1.178 0.04433 0.100
5   0.1102  0.0548  0.1490 1.408 0.01238 0.116
6  -0.5785  0.1207 -0.5854 0.724 0.13792 0.104
7   0.4851 -0.1676  0.4851 0.925 0.10651 0.114
8  -0.6059  0.3277 -0.6179 0.848 0.16313 0.139
9   0.4957 -0.2572  0.5034 0.996 0.11760 0.135
10  0.0157 -1.0428 -1.1090 9.033 0.68380 0.863   *
```

Stata code

```
clear
set seed 1011
*generate the data
  set obs 10
  generate x = rnormal()
  replace x = 5 in 10
  generate y = rnormal(0.5+2*x,1)

  regress y x

  predict cook_d, cook
  list
```

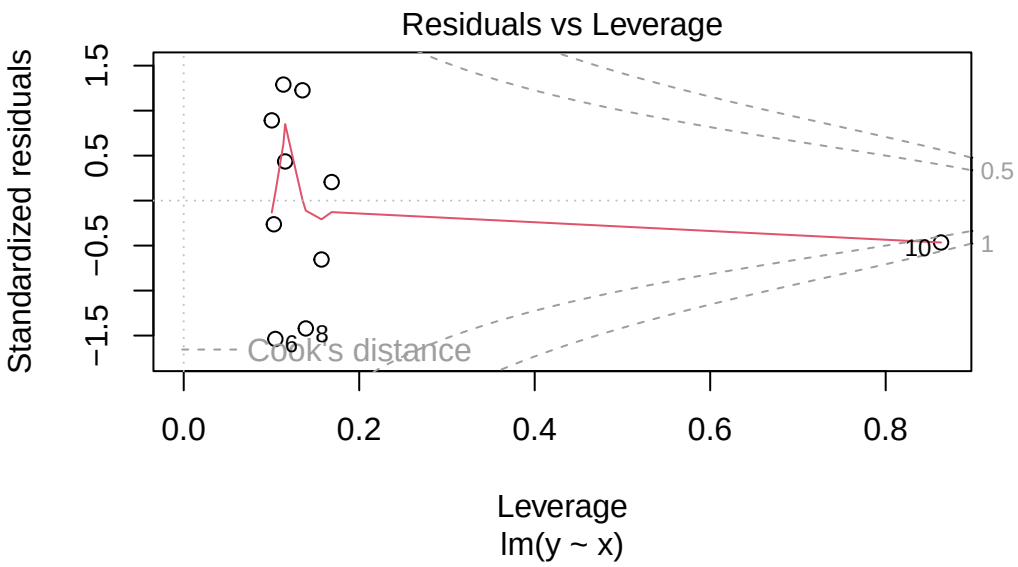
```
## Number of observations (_N) was 0, now 10.
##
##
## (1 real change made)
##
##
##
##       Source |       SS           df       MS      Number of obs   =        10
## -------------+----------------------------------   F(1, 8)         =     52.69
##        Model |  86.4227998         1  86.4227998   Prob > F        =    0.0001
##     Residual |  13.1226538         8  1.64033173   R-squared       =    0.8682
## -------------+----------------------------------   Adj R-squared   =    0.8517
##        Total |  99.5454536         9   11.060606   Root MSE        =    1.2808
##
## ------------------------------------------------------------------------------
##            y | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##            x |    1.98254   .2731326     7.26   0.000     1.352695    2.612385
##        _cons |    .516953    .477044     1.08   0.310    -.5831126    1.617018
## ------------------------------------------------------------------------------
##
##
##
##      +---------------------------------+
##      |         x           y     cook_d |
##      |---------------------------------|
##   1. |  .8897727    2.534658   .0024235 |
##   2. |  .0675149    1.056028     .00888 |
##   3. |  .1199856    1.148383   .0080535 |
##   4. |  .1817483    1.488953   .0186161 |
##   5. |  1.257954    3.040046    .000034 |
##      |---------------------------------|
##   6. | -.2278177   -1.908616   .2698209 |
##   7. | -.1390651   -1.624345   .2228213 |
##   8. |  .3885022    2.991632   .1271682 |
##   9. |   1.69015    5.084781   .0750629 |
##  10. |         5    9.654363   7.563721 |
##      +---------------------------------+
```

With the example using the simulated sample with 10 observations, the rule of thumb would be 4/10. Again, the 10th observation is above this threshold and would requires some consideration.

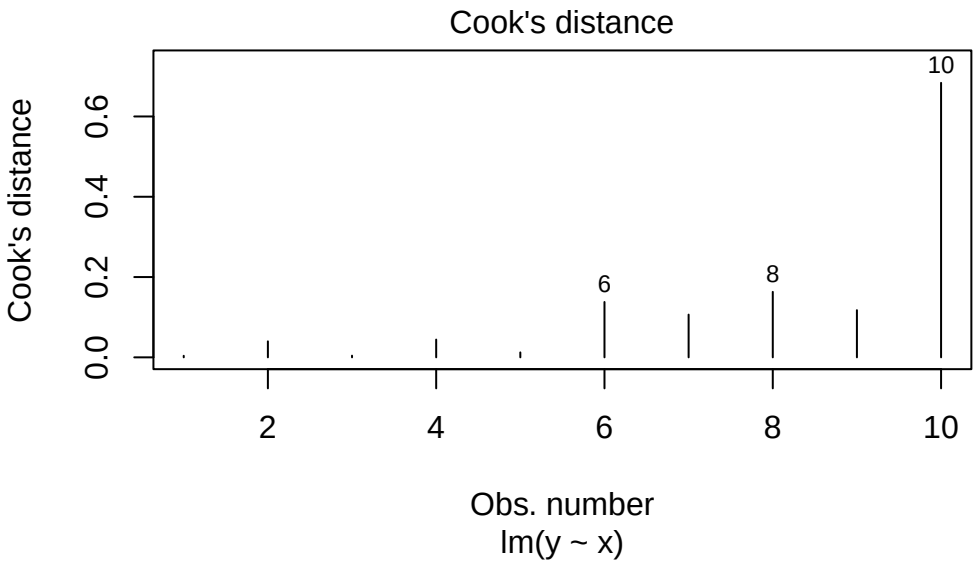
Plots: The above mesaures are commonly represented in a graphical way. There are many variations of these plots. Below are some examples of these plots but many other plots are available in different packages.

R code

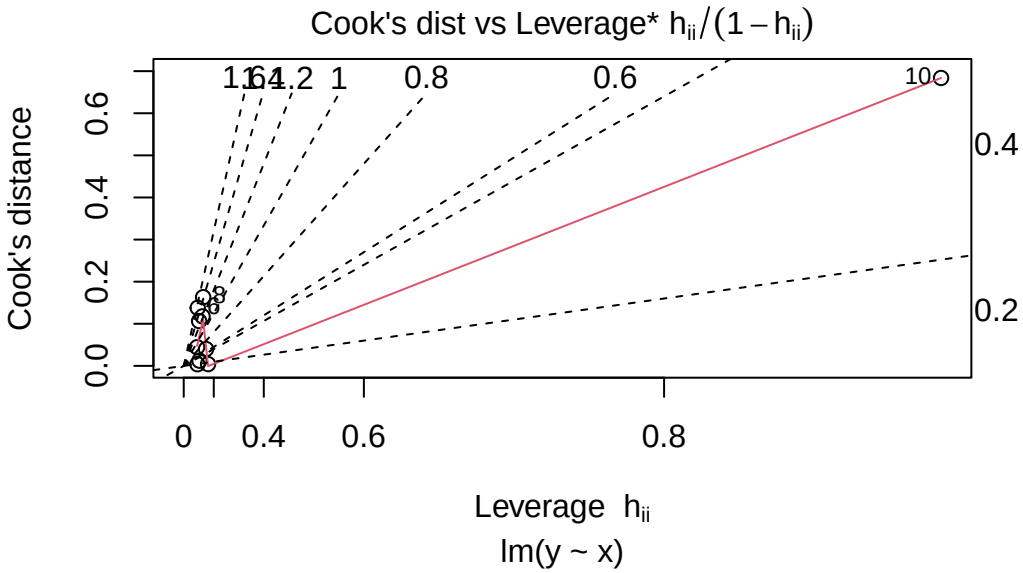
```
#leverage vs residuals
#A data point with high leverage and high residual may be problematic
plot(lmodel,5)
```




```
#Cook's distance
plot(lmodel,4)
```

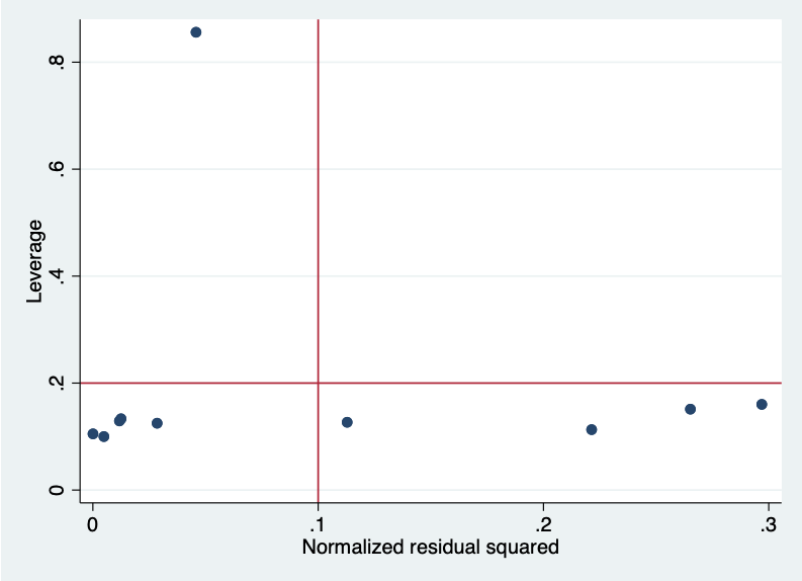



```
#Leverage vs Cook's distance
plot(lmodel,6)
```



Stata code

```
clear
set seed 1011
*generate the data
  set obs 10
  generate x = rnormal()
  replace x = 5 in 10
  generate y = rnormal(0.5+2*x,1)

  regress y x

  lvr2plot

## Number of observations (_N) was 0, now 10.
##
##
## (1 real change made)
##
##
##
##       Source |       SS           df       MS      Number of obs   =        10
## -------------+----------------------------------   F(1, 8)         =     52.69
##        Model |  86.4227998         1  86.4227998   Prob > F        =    0.0001
##     Residual |  13.1226538         8  1.64033173   R-squared       =    0.8682
## -------------+----------------------------------   Adj R-squared   =    0.8517
##        Total |  99.5454536         9   11.060606   Root MSE        =    1.2808
##
## ------------------------------------------------------------------------------
##            y | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##            x |    1.98254   .2731326     7.26   0.000     1.352695    2.612385
##        _cons |    .516953    .477044     1.08   0.310    -.5831126    1.617018
## ------------------------------------------------------------------------------
```



###Book Chapter 4. Outlying, High Leverage, and Influential Poitns 4.7.4 (pages 124-128).

This reading supplements the notes above with emphasis in the DFBETA plots. Note that this subchapter appears in the book after the extension of simple linear regression to the use of multiple independent variables (covariates) in the regression model, which we did not yet cover. However, there are only a few references to the multiple linear regression case.

Exercises:

The dataset lowbwt.csv was part of a study aiming to identify risk factors associated with giving birth to a low birth weight baby (weighing less than 2500 grams).

1 - Fit a linear model for the variable *bwt* (birth weight) using the covariate *age* (mother's age), evaluate the assumptions and interpret the results.

2 - Evaluate potential outliers and influential observations. How would the results change if you excluded this/these observation(s)?

Summary

This weeks key concepts are:

1. The key concepts around binary variables will be added here after you have had a chance to finish your independent investigation.

2. Outliers are observations with a very large absolute residual value. That is, we normally refer to outliers as observations with extreme values in the outcome variable Y. Outliers in the covariate x are observations with high *leverage*. The precise formula for leverage is less important than understanding how high leverage observations can impact your regression.

3. A residual versus leverage plot is a very useful diagnostic to see which observations may be highly influential

    - Observations that are outliers (in the outcome) and that have low leverage, may influence the intercept of your regression model
    - Observations that are not outliers, but have high leverage might artificially inflate the precision of your regression model
    - Observations that are outliers AND have high leverage may influence the intercept and slope of your regression model

4. When potentially highly influential observations are detected, a sensitivity analysis where the results are compared with and without those observations is a useful tool for measuring influence.

4 Multiple Linear Regression - Application

Learning objectives

By the end of this week you should be able to:

1. Understand and explain the effects of uncontrolled confounding, and the concept of its control by holding extraneous factors constant
2. Formulate a multiple linear regression model and interpret it's parameters
3. Formulate and test hypothesis based on linear combinations of regression parameters
4. Use residuals to test multiple linear regression assumptions

Learning activities

This week's learning activities include:

Learning Activity	Learning objectives
Lecture 1	1
Lecture 2	2
Reading	1, 2
Lecture 3	3
Independent exercises	4

Introduction to confounding

Until this week we have focussed on regression between an outcome and a single covariate and called this *simple linear regression*. This week we introduce the concept of *multiple linear regression* where the outcome is regressed on more than one covariate. The motivating reason for multiple linear regression we will present here is for the adjustment of confounding by other factors. However as we will discover in subsequent weeks multiple covariates is a powerful

tool that allows regression analysis to be much more adaptable, and have greater predictive power.

This week begins with a brief recap on confounding in the lecture below.

Download video

Introduction to multiple linear regression

In this video we introduce the multiple linear regression model, where multiple covariates are included. We also look at an example of this implemented in Stata and R and interpret the multiple linear regression output

Stata instructions

Download video

R instructions

Download video

Chapter 4. Linear regression to 4.2.1.1 (pages 69-73).

This reading supplements the above two lectures by providing some examples of confounding, and how this adjusted for in multiple linear regression

Chapter 4. 4.2.2 to 4.2.3 (pages 73-75).

This reading reinforces the content from lecture 2 on important output generated from multiple linear regressions including: the variance of regression coefficients, confidence intervals, and measures of goodness of fit with R squared.

Chapter 4. 4.3 to 4.3.2 (pages 76-81).

This reading supplements the lecture video by describing how categorical variables are included in multiple linear regression - particularrly when those categorical variables have more than 2 categories.

Linear combinations of regression coefficients

Particularly with categorical variables of more than one category, we frequently wish to make inferences on linear combinations of regression coefficients. For example, with categorical variables, the regression coefficients represent the mean difference between one of the groups and the reference category. In this section we learn how to make different comparisons - any comparison that is a linear combination of the regression coefficients. Let us return to the regression for the recorded video earlier, using the `hers_subset` data from the Heart and Estrogen/progestin study (HERS). In the video we looked at a regression on systolic blood pressure (sbp) against age, BMI, alcohol consumption (drinkany), and physical activity (physact). With the following regression results:

Stata code

```
use hersdata, clear
set seed 90896
sample 10

reg SBP age BMI i.drink i.physact
## (2,487 observations deleted)
##
##
##       Source |       SS           df       MS      Number of obs   =       276
## -------------+----------------------------------   F(7, 268)       =      2.72
##        Model |  7440.99084         7  1062.99869   Prob > F        =    0.0097
##     Residual |  104826.237       268  391.142677   R-squared       =    0.0663
## -------------+----------------------------------   Adj R-squared   =    0.0419
##        Total |  112267.228       275  408.244466   Root MSE        =    19.777
##
## ------------------------------------------------------------------------------
##                  SBP | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## ---------------------+--------------------------------------------------------------
##                  age |   .6807476   .1903908     3.58   0.000     .3058957      1.0556
##                  BMI |   .3459596   .2241932     1.54   0.124    -.0954443    .7873636
##                      |
##              drinkany |
##                  yes |  -3.717148   2.453773    -1.51   0.131    -8.548272    1.113975
##                      |
##               physact |
## somewhat less active |   1.517198    5.29391     0.29   0.775    -8.905744    11.94014
##      about as active |   3.092056   4.861608     0.64   0.525    -6.479747    12.66386
## somewhat more active |   3.010075   4.977285     0.60   0.546    -6.789479    12.80963
```

```
##     much more active  |  -2.391424   5.577128    -0.43   0.668    -13.37198    8.589133
##                       |
##                 _cons |   81.07379    15.1853     5.34   0.000     51.17613    110.9714
## ------------------------------------------------------------------------------------------
```

R code

```
hers_subset <- read.csv("hers_subset.csv")
lm.multiple <- lm(SBP ~ age + BMI + drinkany + physact, data = hers_subset)
summary(lm.multiple)
##
## Call:
## lm(formula = SBP ~ age + BMI + drinkany + physact, data = hers_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.190 -14.050  -2.257  13.858  61.159
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   84.16584   14.86302   5.663 3.83e-08 ***
## age                            0.68075    0.19039   3.576 0.000415 ***
## BMI                            0.34596    0.22419   1.543 0.123979
## drinkanyyes                   -3.71715    2.45377  -1.515 0.130984
## physactmuch less active       -3.09206    4.86161  -0.636 0.525309
## physactmuch more active       -5.48348    3.95433  -1.387 0.166685
## physactsomewhat less active -1.57486    3.73468  -0.422 0.673593
## physactsomewhat more active -0.08198    3.00415  -0.027 0.978249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.78 on 268 degrees of freedom
## Multiple R-squared:  0.06628,    Adjusted R-squared:  0.04189
## F-statistic: 2.718 on 7 and 268 DF,  p-value: 0.009724
confint(lm.multiple)
##                                   2.5 %      97.5 %
## (Intercept)                  54.9027060 113.4289776
## age                           0.3058957   1.0555995
## BMI                          -0.0954443   0.7873636
## drinkanyyes                  -8.5482722   1.1139755
## physactmuch less active     -12.6638586   6.4797472
```

```
## physactmuch more active     -13.2689900   2.3020311
## physactsomewhat less active  -8.9278976   5.7781819
## physactsomewhat more active  -5.9967253   5.8327632
```

Now suppose we wish to make some specific pairwise comparisons that are not captured by the comparison between the reference category. For example, perhaps we wish to compare the difference between the means of the "much less" category, and the "much more" category. Let's think about what this comparison would be from the regression coefficients of this regression equation using the following acroynms:

- MLA - Much less active
- MMA - Much more active
- SLA - Somewhat less active
- SMA - Somewhat more active

$$SBP = \beta_0 + \beta_1 \text{age} + \beta_2 \text{BMI} + \beta_3 \text{drinkany} + \beta_4 \text{MLA} + \beta_5 \text{MMA} + \beta_6 \text{SLA} + \beta_7 \text{SMA}$$

Given that

$$\beta_4 = \text{mean of MLA} - \text{mean of reference category}$$

and

$$\beta_5 = \text{mean of MMA} - \text{mean of reference category}$$

then it follows that

$$\beta_4 - \beta_5 = \text{mean of MLA} - \text{mean of MMA}$$

Therefore a calculation of $\beta_4 - \beta_5$ will give us the desired mean difference between the much less active group and the much more active group - after adjusting for age, bmi and alcohol consumption. We can of course do this manually from the regression output, however we save a lot of time if we do this in Stata or R, as those packages will also automatically calulate P-values and confidence intervals for the associated comparison.

In Stata, we do this with the "lincom" command, and specify the levels of the physical activity category with a numeral followed by a ".". i.e. for the comparison above the much less active group is level 2 of the physical activity variable, and the much more active group is level 3. So the Stata code and output would be

```
use hersdata, clear
set seed 90896
sample 10

reg SBP age BMI i.drink i.physact
lincom 2.physact - 3.physact

## (2,487 observations deleted)
##
##
##       Source |       SS           df       MS      Number of obs   =       276
## -------------+----------------------------------   F(7, 268)       =      2.72
##        Model |  7440.99084         7  1062.99869   Prob > F        =    0.0097
##     Residual |  104826.237       268  391.142677   R-squared       =    0.0663
## -------------+----------------------------------   Adj R-squared   =    0.0419
##        Total |  112267.228       275  408.244466   Root MSE        =    19.777
##
## ------------------------------------------------------------------------------
##                  SBP | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## ---------------------+--------------------------------------------------------------
##                  age |   .6807476   .1903908     3.58   0.000     .3058957      1.0556
##                  BMI |   .3459596   .2241932     1.54   0.124    -.0954443    .7873636
##                      |
##              drinkany |
##                  yes |  -3.717148   2.453773    -1.51   0.131    -8.548272    1.113975
##                      |
##               physact |
## somewhat less active  |   1.517198    5.29391     0.29   0.775    -8.905744    11.94014
##      about as active  |   3.092056   4.861608     0.64   0.525    -6.479747    12.66386
## somewhat more active  |   3.010075   4.977285     0.60   0.546    -6.789479    12.80963
##     much more active  |  -2.391424   5.577128    -0.43   0.668    -13.37198    8.589133
##                       |
##                 _cons |   81.07379    15.1853     5.34   0.000     51.17613    110.9714
## ------------------------------------------------------------------------------
##
##
##  ( 1)  2.physact - 3.physact = 0
##
## ------------------------------------------------------------------------------
##          SBP | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## -------------+----------------------------------------------------------------
```

```
##          (1) |  -1.574858   3.734678    -0.42   0.674    -8.927898    5.778182
## ------------------------------------------------------------------------------
```

In R, we do this calculation by first specifying a matrix which designates the comparison we would like to make. The matrix here must have the same number of columns as the number of regression coefficients in our regression equation - in this example 8 (β_0 to β_7). We would like to make a subtraction between the β_4 and β_5 corresponding to the fifth and sixth regression coefficients. So our matrix comparison is defined as `comparison <- matrix(c(0,0,0,0,1,-1,0,0), nrow=1)`. We then use the `glht` command from the `multcomp` library to calculate this linear combination, and use the `summary` and `confint` commands to output the P-value and confidence intervals.

In both Stata and R, we observe that the much more group has a lower SBP mean of 2.39mmHG compared with the much less active group (95% CI 8.59mmHG greater to 13.37mmHG lower) - corresponding to no evidence for a difference (P = 0.67).

Model checking for multiple linear regression

In week 2 we investigated how residuals can be used to check assumptions 1-3 of linear regression. These tests are already fit for purpose for multiple linear regression as described below

1. Linearity. A residual versus fitted plot is still an excellent tool for assessing linearity in multiple linear regression. If there are concerns with this plot, the assumption can be further investigated with a residual versus predictor plot for each covariate in the regression. Remember, this is only useful for continuous covariates and does not need checking for categorical covariates
2. Homoscedasticity (constant variance). A residual versus fitted plot is still an excellent tool for assessing homoscedasticity in multiple linear regression. If there are concerns with this plot, the assumption can be further investigated with a residual versus predictor plot for each covariate in the regression. For continuous covariates, this is best shown with a scatter plot. For categorical variables, a boxplot is best for comparing the variance across categories.
3. Normality. A normal quantile plot of the residuals, or a histogram of the residuals is still an excellent tool for assessing the normality of the residuals

Independent exercise

Continuing on with the `hers_subset` example. Write down the regression equation for a regression with an outcome of body mass index, and with age and physical activity (physact)

as covariates. Interpret each parameter in this equation.

Carry out this regression and report on the key findings.

Finally, express the following comparisons in terms of the regression coefficients of your equation above, and calculate these using Stata or R

- The mean difference between much more active, and much less active
- The mean difference between much more active, and somewhat more active
- [Challenge question] The mean difference between the more active groups (somewhat and much more active combined), and the less active groups (somewhat and less active combined).

Summary

This weeks key concepts are:

1. Multiple linear regression is the natural extension to simple linear regression where more than one covariate is included as an independent variable. The formula for a multiple linear regression is
$$Y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \beta_3 x_{3,i} + ... \epsilon_i$$
- β_0 still represents the *intercept*, the estimated mean outcome when all covariates (x's) equal zero.
- β_i still represents the mean change in the outcome for a one unit increase in x_i. For categorical variables, this represents the mean change from group x_i to the reference category

2. The statistical evidence (P-value) for a group of regression coefficients can be calculated with an F test or likelihood ratio test. This is important for categorical variables of more than 2 categories as it provides a single P-value for that categorical variable that does not depend on the reference category
3. Linear combinations of regression coefficients can be calculated with the *lincom* command in Stata and the *glht* command from the *multcomp* package in R.
4. Checking model assumptions can be done in largely the same was for multiple linear regresssion as for simple linear regression. Residual versus predictor plots are an additional plot you can use to investigate deviations to linearity or homoscedasticity.

5 Multiple linear regression theory

Learning objectives

This week materials provide the theoretical basis for multiple linear regression that you have been using in the previous 4 weeks. It is somehow more technical but it is nevertheless important that you understand where these results come from.

By the end of this week you should be able to:

1. Be familiar with the basic facts of matrix algebra and the way in which they are used in setting up and analysing regression models
2. Understand the algebraic formulation of the LS estimator and its properties
3. Discover the principal forms of statistical inference applied to the multiple regression model, and in particular how these relate to partitioning of the total sum of squares
4. Learn how 95% confidence intervals and 95% prediction intervals are derived
5. Discover how this is linked to likelihood-based inference

Learning activities

This week's learning activities include:

Learning Activity	Learning objectives
Lecture 1	1, 2, 3
Lecture 2	3, 4, 5
Practice/Exercises	1, 2, 4
Discussion	All

Lecture 1 in R

Download video here

Lecture 1 in Stata

Download video here

Matrix algebra for simple linear regression

Notational convention

All vectors are assumed to be column vectors. So for example a vector of length n with elements $a_1, ..., a_n$ is defined as the *column vector*

$$a = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix},$$

so its *transpose* is the *row vector* a' (or a^T) with $a' = (a_1, \dots, a_n)$.

Generally, we will use capital letters for matrices, as commonly done (in many textbooks and other writings, is also common the have the names of vectors and matrices in boldface but we *do not 8follow this convention in these notes for simplicity and let the context decide).

Consider the simple linear regression model:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

We can write this in terms of vectors/matrices:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

And, more compactly, as $Y = X\beta + \varepsilon$.

Notice that the matrix X consists of two columns (i.e. has dimension $n \times 2$) - the column vector of 1's and the column vector of covariate values $(x_1, x_2, ..., x_n)$ corresponding to observations $i, \dots, n$

Exercise 1

To illustrate some of the important matrix operations we ask you to carry through "by hand" a regression analysis using just 10 pairs of (x, y) values.

Below the exercise is interactive so you are able to write the R code (sorry, only works for R, but see below the Stata code) and execute it.

See below for the Stata code for Exercise 1

Stata code and output

```
## here we suppose that you have entered the following data
## by hand using the Stata editor. The data has two columns xx and yy
## where: xx^T=(10, 20, 25, 18, 43, 13, 50)
## yy^T=(100, 130, 125, 98, 149, 89, 149)
use test_data.dta
## generate a column of ones (called cons)
gen cons =1
## Create a matrix consisting of the column of 1's and xx and store this in a matrix calle
mkmat cons xx, matrix(X)
## Create a matrix with one column containing the yy's  and call it Y
mkmat yy, matrix(Y)
## Create the matrix X'X (with the name XTX)
matrix XTX = X'*X
## Create the inverse of XTX and call it XTX
matrix invXTX = inv(XTX)
## compute the LSE anc call it b
matrix b=invXTX*X'*Y
## list b
matrix list b
## Extract the diagonal of the squared matrix (here invXTX) and list it
matrix D=vecdiag(invXTX)
matrix list D
## more information on matrix expressions in Stata can be found here:
## https://www.stata.com/manuals/u14.pdf
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
##
## Unknown #command
##
```

```
## Unknown #command
##
## Unknown #command
##
## Unknown #command
## X' not found
## r(111);
##
## r(111);
```

Least squares estimates for multiple linear regression

The formulation of the least squares (LS) principle in multiple regression model and the derivation of the LS estimation will now be briefly described. Suppose we have p independent variables, the LS solution requires finding the values of the regression parameters $\beta_0, \beta_1, \ldots, \beta_p$, that minimise the sum of squares:

$$S = \sum_{i=1}^{n} [y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{pi})]^2.$$

Using the matrix formulation of the model just as we did with simple linear regression but having this time p covariates, $Y = X\beta + \varepsilon$ and we can write this sum as:

$$S = (Y - X\beta)'(Y - X\beta) = Y'Y - 2Y'X\beta + \beta' X'X\beta.$$

This is actually a scalar quantity (i.e. a single number), calculated from vectors and matrices, so to solve for the values of $\beta_0, \beta_1, \ldots, \beta_p$ that minimise S, we need to find the zero of its derivative with respect to the β coefficients.

Before we proceed,we need to understand how to differentiate a function of a vector quantity. Let $g(\beta) = g(\beta_1, \ldots, \beta_p)$ be a function of $\beta = (\beta_1, \ldots, \beta_p)'$ that returns a scalar (single number) answer. An example of such a function $g(.)$ is a linear combination of $(\beta_1, \ldots, \beta_p)$, say $a'\beta = a_1\beta_1 + \cdots + a_p\beta_p$. Define

$$\frac{\partial g(\beta)}{\partial \beta} = \begin{bmatrix} \frac{\partial g(\beta)}{\partial \beta_1} \\ \vdots \\ \frac{\partial g(\beta)}{\partial \beta_p} \end{bmatrix},$$

where we have used the ∂ notation to indicate partial derivatives, i.e. the derivatives of $g(\beta)$ with respect to each component β_j of $(\beta_1, \ldots, \beta_p)$, holding all other components fixed. Then it is easy to see that

$$\frac{\partial (a'\beta)}{\partial \beta} = \begin{bmatrix} \frac{\partial a'\beta}{\partial \beta_1} \\ \vdots \\ \frac{\partial a'\beta}{\partial \beta_p} \end{bmatrix} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = a,$$

and it is also true (although not quite as simple to show) that

$$\frac{\partial(\beta' A\beta)}{\partial\beta} = (A + A')\beta$$

and in the important special case when A is symmetric

$$\frac{\partial(\beta' A\beta)}{\partial\beta} = 2A\beta$$

We may now apply these results where $g(\beta)$ is the sum S above, to produce the matrix formula for the LS estimates. Differentiating with respect to β we get:

$$\frac{\partial S}{\partial\beta} = 0 - 2X'Y + 2X'X\beta = -2X'Y + 2X'X\beta$$

Solving $\frac{\partial S}{\partial\beta} = 0$ yields $X'X\beta = X'Y$, and so the solution is

$$\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \vdots \\ \hat{\beta}_p \end{bmatrix} = (X'X)^{-1}X'Y$$

. Estimates can be computed without matrix calculation but the general formula given above applies to all cases including $p = 1$.

Exercise 2: Adjusted regression of glucose on exercise in non-diabetes patients, Table 4.2 in Vittinghof et al. (2012)

1) Reproduce the adjusted analysis of glucose carried out in p. 72. Make sure that you exclude diabetes patients
2) Use matrix operations in Stata or R to create the X, Y, $X'X$ and $X'Y$ matrices and use these to obtain the LS estimates. [Caution: there are missing values in some of these covariates so delete first all observations with missing values before any matrix manipulation]
3) Optional: Use an explicit matrix calculation in Stata/R to obtain the variance-covariance matrix for b in the regression of glucose on the previous covariates. Calculate the standard errors and confirm your results by comparing with the regression output.

To help you with this exercise, you may want check the code given in the lectures (R or Stata depending on your favourite software). Also, for Stata users, some key commands will be reminded at the beginning of the solutions. You may have to increase the memory before creating matrices by typing: *set matsize 2500*. It turns out that the memory for matrices is pretty limited by default. We expect you to try on your own before looking at the solutions.

It is worth noting that the normal equations $X'X\beta = X'Y$ are *not* solved by using methods that involve the direct calculation of the inverse matrix $(X'X)^{-1}$. Computer programs use numerical algorithms that are both quicker and more numerically stable than working out the full inverse and then multiplying it with $X'Y$ to obtain $\hat{\beta}$.

Predicted values and residuals

It is now simple to write the vector of predicted values (at the observed covariate values) using the matrix notation:

$$\widehat{Y} = \begin{bmatrix} \widehat{Y}_1 \\ \widehat{Y}_2 \\ \vdots \\ \widehat{Y}_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & ... & x_{p1} \\ 1 & x_{12} & ... & x_{p2} \\ \vdots & \vdots & ... & \vdots \\ 1 & x_{1n} & ... & x_{pn} \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_p \end{bmatrix} = X\hat{\beta} = X(X'X)^{-1}X'Y$$

Letting $H = X(X'X)^{-1}X'$, we then have $\widehat{Y} = HY$.

The matrix H is often called the "hat" matrix, as it "puts the hat on Y", that is, it transforms Y into $\widehat{Y}$. In the previous weeks, we introduced the diagonal elements of H in the form of h_{ii} and termed the values *leverages*. This explains technically how leverage is computed and, again, some simpler formulas can be derived for the simple regresssion case. In general, this H matrix as an important role since many additional regression results can be expressed simply in terms of the matrix H.

The matrix H has some important properties. We can see easily that:\ (i) H is symmetric \ (ii) H^2=H [and so H is *idempotent* in mathematical terms]

An important property of idempotent matrices is that their trace (sum of diagonal elements) is equal to the rank of the matrix. It then follows that since H is idempotent its rank is equal to the sum of its diagonal elements (i.e., $\sum_{i=1}^{n} h_{ii}$), which is the number of columns in X (or equivalently the number of parameters in the regression model - assuming X is of full rank). So, for example, for simple linear regression the rank of H is 2.

Using the matrix H, we can express residuals in the simple form $e = Y - \widehat{Y} = (I - H)Y$ and immediately deduct that their expectation is 0. Note that the sum of residuals is also zero for all models with an intercept. Deriving their variance-covariance is slightly more complicated but a bit of algebra and the properties of the H matrix yield $var(e) = (I - H)\sigma^2$. This is used to compute standardised residuals in all statistical packages. Note that the variance is different from $\sigma^2 I$ which is variance of the (true) error vector ε.

Geometric interpretation

It is possible to interpret LS estimation as a projection onto the linear space spanned by the regressors. Watch this short video to understand why:

Watch video here

Standard inference in multiple linear regression

The first level of inference for a multiple regression model does not require specific distributional assumptions about the random errors, which we can now represent as the vector ε. By this we refer to the fact that the expected value (and therefore unbiasedness) of regression coefficients and the variances of estimates - and covariances between estimates - all follow from the assumption that $\varepsilon \sim (0, \sigma^2 I_n)$, i.e. that the elements of ε are independently distributed with common variance σ^2. The additional standard assumption that the errors follow a *normal distribution* is important in providing the formal basis for the calculation of confidence intervals and tests based on the t distribution.

A simple calculation using matrix algebra for random vector to shows that $\hat{\beta}$ is an unbiased estimate of β. The The variance-covariance matrix of $\hat{\beta}$, assuming the errors $\varepsilon \sim (0, \sigma^2 I_n)$, is:

$$var(\hat{\beta}) = var[(X'X)^{-1}X'Y] = (X'X)^{-1}X'var(Y)X(X'X)^{-1} = \sigma^2(X'X)^{-1}$$

(using the general result that $var(CY) = C \times var(Y) \times C'$ for any matrix C). For simple linear regression this is a 2×2 matrix, and the (2,2) element is $var(\hat{\beta}_1) = \sigma^2 / \sum_{i=1}^{n}(X_i - \bar{X})^2$.

To use the formula for the variance of $\hat{\beta}$ we need to replace σ^2 in the formula with an estimated value, and the natural (unbiased) estimate is the Mean Square for Error from the analysis of variance table, MSE (more on this below). For multiple regression, the *estimated* variance-covariance matrix of $\hat{\beta}$ is thus $\widehat{var}(\hat{\beta}) = MSE \times (X'X)^{-1}$. The diagonal elements of this matrix are generally the ones of interest, since they provide the squared standard error for each coefficient estimate. However, the covariance terms can also be important, as these reflect the extent to which inferences about each coefficient are independent of each other. In fact you have already seen an important application of this idea in Week 4 where you had to test whether the difference of two coefficients was equal to 0. Specifically, you were asked to compute the mean SBP difference between the much less active group and the much more active group (called $\beta_4 - \beta_5$) after adjusting for age, BMI and alcohol consumption. A subset of *hers* data was used for this analysis. To obtain the corresponding SE we need to compute first the following variance:

$$\widehat{var}(\hat{\beta}_4 - \hat{\beta}_5) = \widehat{var}(\hat{\beta}_4) + \widehat{var}(\hat{\beta}_5) - 2\widehat{cov}(\hat{\beta}_4, \hat{\beta}_5)$$

SE is the squared root conveniently provided to us using the "lincom" command in Stata and "glht" in R. We can check the results by asking the package to output the variance-covariance

matrix for the vector $\hat{\beta}$ after fitting the model, extract the terms we need, and finally derive $\widehat{var}(\hat{\beta}_4 + \hat{\beta}_5)$ using the formula given above. The corresponding SE is simply the square root. Then, we can proceed with the t-test (or z-test for large samples) as commonly done.

We don't provide here the detail of this calculation, only the logic that illustrates the importance of the whole variance-covariance matrix.

Lecture 2 in R Download video here

Lecture 2 in Stata Download video here

The analysis of variance for multiple linear regression (SST decomp)

The output of a fitted model in linear regression is typically displayed as an ANOVA table. The fundamental idea is that the total Sum of Squares (denoted SST is decomposed into two components, the Regression Sum of Squares (SSR) and the Error (or Residual) Sum of Squares (SSE): $SST = SSR + SSE$.

The Total Sum of Squares measures the total variation of the Y values around the sample mean $\bar{Y}$, and the ANOVA decomposition displays the two components and what fraction of the total variation can be "explained" by the regression model. The fraction SSR/SST is the R^2 ('R-squared"), sometimes called the *coefficient of determination*. R^2 (and its squared root R) are essentially descriptive quantities that provide a measure of the strength of association between X representing several covariates considered jointly and the outcome Y. In simple regression, R^2 is equal to the square of the correlation coefficent between the outcome and lone covariate.

An important point to note is that all three sums of squares are quadratic forms, meaning that they can be expressed as YAY for some symmetric matrix A. This is important in deriving the sampling properties of the sums of squares and related standard errors and test statistics, which we now review without giving full details or derivations. The fundamental fact about quadratic forms is that under a normal error model and with appropriate scaling they have chi-squared distributions.

The ANOVA table for a (multiple) regression model $E(y_i) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi}$ is as follows (in Stata):

Source of variation	SS	df	MS
Regression	SSR	p	$MSR = SSR/p$
Error	SSE	n-(p+1)	$MSE = SSE/[n-(p+1)]$
Total	SST	n-1	

The final column, of Mean Squares, is obtained by dividing each SS by its degrees of freedom (df). Note that the df for SSR is now p, representing the number of independent covariates fitted in the regression model (not counting the constant), while the df for SSE is reduced accordingly, to $n-(p+1)$. The Mean Square for Error (MSE) is especially important because dividing by df gives a quantity that has expected value σ^2, making it a natural estimate for σ^2. Furthermore, $SSE/\sigma^2 \sim \chi^2$ with $n-(p+1)$ degrees of freedom. This is the reason that for the normal error regression model we can use the standard inferences for each estimated regression coefficient, based on $SE(\hat{\beta}_j)^2 = j^{th}$ diagonal element of the estimated variance-covariance matrix $\widehat{var}(\hat{\beta}) = MSE \times (X'X)^{-1}$. In particular, confidence intervals and tests are constructed in the familiar way using this estimated standard error and the t distribution with $n-(p+1)$ degrees of freedom.

Some additional output is also provided, e.g. the overall test of the (global) null hypothesis that $\beta_1 = \beta_2 = ... = \beta_p = 0$. We are effectively testing whether the model under investigation is better than a model with only the intercept. This is carried out by forming the F-ratio: $F^* = MSR/MSE$. Under the null hypothesis, MSR is proportional to a chi-squared random variable and has expected value σ^2, so F^* has an F distribution with degrees of freedom $p, n-(p+1)$.

Stata code and output

```
use hersdata, clear
regress glucose exercise age drinkany BMI if diabetes == 0
##       Source |       SS           df       MS      Number of obs   =     2,028
## -------------+----------------------------------   F(4, 2023)      =     39.22
##        Model |  13828.8486         4  3457.21214   Prob > F        =    0.0000
##     Residual |  178319.973     2,023  88.1463042   R-squared       =    0.0720
## -------------+----------------------------------   Adj R-squared   =    0.0701
##        Total |  192148.822     2,027  94.7946828   Root MSE        =    9.3886
##
## ------------------------------------------------------------------------------
##      glucose | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##     exercise |   -.950441     .42873    -2.22   0.027    -1.791239   -.1096426
##          age |   .0635495   .0313911     2.02   0.043     .0019872    .1251118
##     drinkany |   .6802641   .4219569     1.61   0.107    -.1472514     1.50778
##          BMI |    .489242   .0415528    11.77   0.000     .4077512    .5707328
##        _cons |   78.96239   2.592844    30.45   0.000     73.87747    84.04732
## ------------------------------------------------------------------------------
```

In the example of the previous section that we reproduce here, we see that $F^* = 30.22$, which is an extremely high value for an F distribution with degrees of freedom $(4, 2023)$, leading to $P < .001$, and the unsurprising conclusion that the data are highly inconsistent with the null hypothesis.

Note that this anova table is not provided in R where a simpler output is displayed. It is possible to obtain something close to this using the *anova* command *after* a fit of the same linear model provided by *ols* from the *rms* library (developed by F Harrell).

R code and output using *ols*

```
library(rms)
# library(haven)
# hers<-read_dta("hersdata.dta")
hers <- read.csv("hers.csv")
hers.nondiab<-hers[hers$diabetes ==0,]
fit <- ols(glucose ~ exercise + age + drinkany + BMI, data = hers.nondiab)
anova(fit)
```

Prediction in multiple regression (95% CI + 95% prediction interval)

The idea of using a fitted model to create predictions of either the expected (mean) value of the outcome variable or the value to be expected for a new individual response at a given covariate value was explored in week 1. It naturally carries over to the multiple regression case. The matrix notation makes it easy to justify how this is done. Assume that we are interested in getting a predicted value of y^* the expected outcome when $x = x^*$ that may or may not be a vector of covariates from the sample. Here x^* is now a $(p+1) \times 1$ vector containing 1 and the values of the p covariates). A prediction for $y*$ is $\hat{y}* = x^{*\prime}\hat{\beta}$ i.e. we just plug-in the LS estimate in the linear combination for a patient with that profile. For inference concerning this quantity, the relevant standard error is given by:

$$SE(\hat{y}^*) = \hat{\sigma} \times \sqrt{x^{*\prime}(X'X)^{-1}x^*},$$

where $\hat{\sigma}$ is the root-MSE. For inference concerning the predicted value for a new individual y at $x = x^*$ the relevant standard error is given by:

$$SE(\hat{y}^* + \epsilon) = \hat{\sigma} \times \sqrt{1 + x^{*\prime}(X'X)^{-1}x^*}.$$

The notation used on the left-hand side here is a reminder that the uncertainty involved in making a prediction for a new individual involves not only the uncertainty in the estimated parameters, but also the contribution due to the random error term ϵ.

The corresponding 95% CI or 95% prediction interval follows by using the usual formula (e.g. $\hat{y} \pm 1.96SE(\hat{y}^*)$ for the 95% CI in large samples). When the sample is not so large the .975 quantile from the t-distribution with $n-(p+1)$ degrees of freedom should be used instead of 1.96.

Computational note: In Stata and R, as in most packages, the regression command comes equipped with a facility for generating predicted values with appropriate standard errors. The command *predict* command works just the same for multiple regression as for simple regression.

Exercise 3: 95% CI for glucose in non-diabetes patients - Optional

We will use the same model as in Exercise 5.2 (and Table 4.2 p. 72).

1) Using your favourite software compute the 95% CI for the mean glucose of a patient aged 65, who does not drink nor exercise and has BMI=29.

2) Can you reproduce this result using matrix manipulations and the formula given above?

You are on your own for this exercise. By now you should be more familiar with matrix manipulation and be able to reproduce in 2) the 95% CI for the mean glucose obtained using your favourite software.

Likelihood-based inference with the normal error model

The OL estimator is the same as the maximum likehood estimator (MLE) under the assuption of normality $N(0, \sigma^2)$ for the error term. To see this, we can just write the log-likelihood of the data under normal linear model, yielding:

$$LL(\beta) = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi}))^2$$

The log-likelihood $LL(\beta)$ is proportional to the negative of $S = S(\beta)$ used earlier up to a constant that only depends on σ. Therefore, minimising $S(\beta)$ is equivalent to maximising $LL(\beta)$, the multiplicative constant $1/(2\sigma^2)$ playing no role in this problem since it does not depend on the regression parameter. You can also derive separately the MLE of σ^2. StraightforWard calculation leads to:

$$\widehat{\sigma}^2_{ML} = \frac{(Y - X\widehat{\beta})'(Y - X\widehat{\beta})}{n} = \frac{SSE}{n} = \frac{(n - (p+1))}{n} MSE.$$

This shows that the MLE of σ^2 is slightly biased in small samples, the bias becoming negligible for large n's. All statistical packages use the unbiased estimate to deal with all possible situations.

Finally, we note without giving further detail that the standard F-tests of multiple linear regression are also *likelihood ratio tests*. The F distribution provides an exact sampling distribution for these test statistics. For large sample sizes (as the estimate of σ^2 becomes better,

i.e. the denominator of MSE can be regarded as effectively fixed) this approaches the chi-squared distribution that applies for large n to all likelihood ratio tests.

It is important to establish this connection given that the ML theory will be used in generalised linear models that extend linear regression. This includes logistic regression for binary data that will be studied in weeks 9-12 of this unit.

Summary

The following are the key takeaway messages from this week:

1. LS estimates and their variance can be derived from linear algebra
2. The properties of the LS estimator have been justified theoretically
3. 95% confidence intervals and 95% prediction intervals can also be expressed using matrix formulation.
4. The LS estimate is the maximum likelihood estimator under the assumption of a Gaussian error term.

6 Interaction and Collinearity

Learning objectives

By the end of this week you should be able to:

1. Understand and explain the concept of interaction (effect modification)
2. Carry out linear regression analysis that accounts for interaction and interpret the key findings
3. Understand the concept of collinearity and how it affects linear regression
4. Implement model building strategies for dealing with collinearity

Learning activities

This week's learning activities include:

Learning Activity	Learning objectives
Reading	1, 2
Collaborative exercise	1, 2
Independent investigation	3, 4
Independent exercises	3, 4
Live tutorial/discussion	3, 4

Independent investigation - Collinearity

This week you will be practicing learning independently an unfamiliar biostatistics topic. This is aimed to provide to you the skills of independent learning a biostatistician needs to carry out an unfamiliar analysis that was not covered in the Master of Biostatistics. These are skills that will also be needed in your workplace placement project (Master students only).

The topic of this weeks independent learning exercise is collinearity. Collinearity is an issue that can be faced in any regression model with more than one covariate. This week you will

independently investigating collinearity and methods to deal with this statistical phenomena. To do this perform the steps below:

1. Use appropriate resources to learn about collinearity in linear regression. Specifically, by the end of this investigation you should be able to:
    - Describe what collinearity is in linear regression
    - Carry out an investigation to detect collinearity
    - Be familiar with methods of dealing with collinarity
2. Investigate the reliability of your source. Some helpful questions to ask yourself are:
    - How does this information compare to other sources?
    - What authority does the author have to provide this information?
    - When was this resource developed and is the information likely to have changed?
    - What is the purpose of this resource?
3. Create a resource that describes to your peers your findings. This resource could be a short video, or just a discussion board post. The resource should include the following
    - A description of what is collinearity
    - Instructions on how to detect collinearity in linear regression analysis
    - A description of at least one technique of dealing with collinearity
    - Your proposed solution to the exercise below (optional)

For the third task, each learning resource should be relatively short (taking 2-4 minutes to watch or read) and be posted in the week 5 independent task assignment (even though it is not an assignment). All learning resources that are posted will receive feedback from the unit coordinator. Your resource will also be available to other students who may also wish to provide some peer feedback and or encouragement.

Independent Exercise

In managing children with asthmas and other respiratory diseases, it is important to monitor various measures of lung function, one of the most import of which is the volume of air that can be forced out fot he lungs in one second, known as the Forced Expiratory Volume (1 second) or FEV1 for short (measured in litres). The "lungfun.csv" dataset contains the following variables:

- idnum: patient identification number

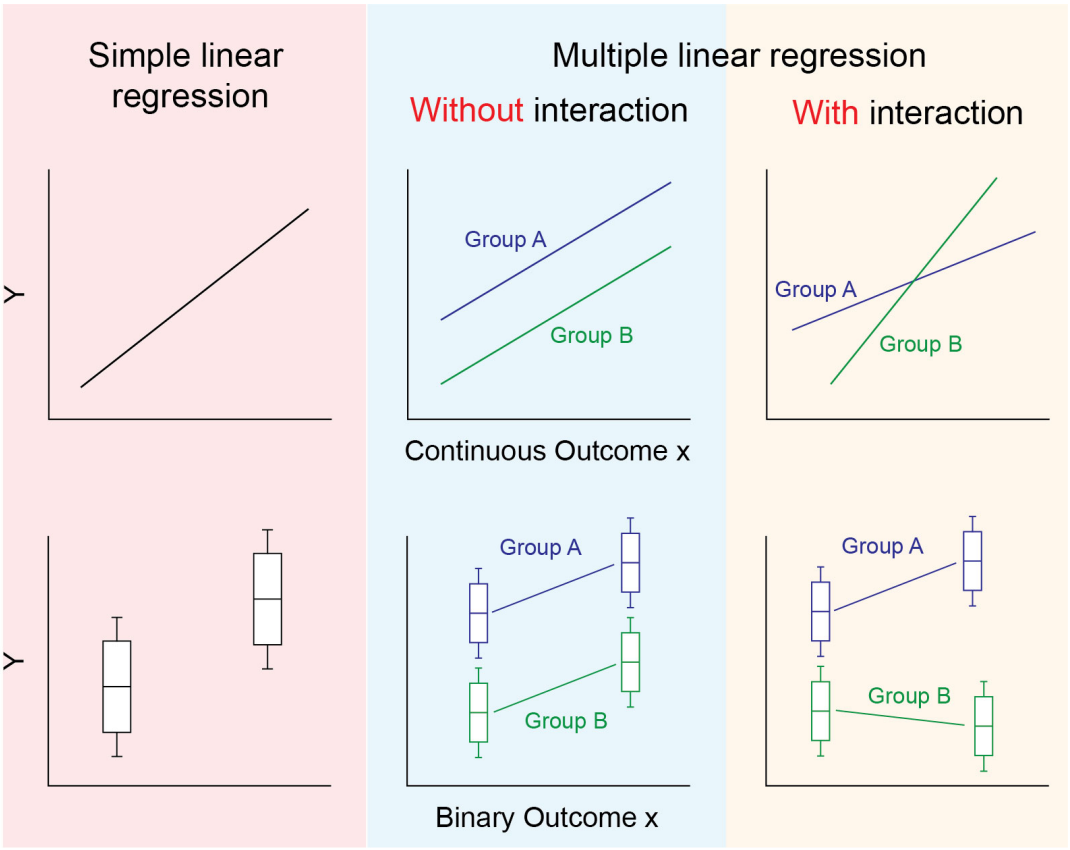
- age: age in years
- wt: weight in kg
- height: in cm
- armsp: arm span in cm (the distance between the fingertips of left and right hands when hands outstretched)
- ulna: ulna length in cm (the ulna is the main bone of the arm below the elbow)
- farm: forearm length in cm (distance from elbow to the fingertip)
- fev1: Forced Expiratory Volume in 1 second (L)

Carry out a regression with fev1 as the outcome, and all covariates included (excluding idnum). Describe any collinearity you observe. How would you recommend dealing with this collinearity?

Interaction (effect modification)

In previous weeks we assumed that the true underlying effect of an exposure on an outcome was constant over all individuals. This is not always the case and when the effect of the exposure is different for different groups this is called "interaction". Another term frequently used instead of interaction is "effect modification". You may wish to review your epidemiology notes for more information and examples of interaction.

The panel of images below is a helpful visual representation of the differences in regression models with and without interaction. The first column shows simple linear regression with a single relationship between x and Y (a continuous x is shown in the top 3 panels, and a binary x is shown in the bottom three. The second column shows multiple linear regression as we have used it so far. Here a binary "group" variable is included to adjust for the mean difference in Y across groups A and B. As the two lines are parallel along x the mean difference between groups is constant for all values of x. In the third panel, we see that the relationship between x and Y is different for group A and group B. That is, there is interaction between x and the group variable. The lines are not parallel so the mean difference between group A and B depends on the value of x.



A regression model for interaction

We introduce the mathematical form of the regression model for interaction by considering a regression model with two covariates x_1 and x_2 which are regressed on our outcome variable Y. The equation for this model without interaction is shown below:

$$\mathrm{E}(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

The term we add to this model to account for, and test for interaction is the *product* of x_1 and x_2 as follows:

$$\mathrm{E}(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$$

To see why this works, consider the following factorisations of this regression equation

$$\mathrm{E}(Y) = \beta_0 + \beta_1 x_1 + (\beta_2 + \beta_3 x_1) x_2$$

Here, that the effect of x_2 on Y equals $\beta_2 + \beta_3 x_1$. That is the effect of x_2 is dependent on the value of x_1 - the definition of interaction. Also, you could instead factor out x_1 instead of x_2 to obtain the following

$$\mathrm{E}(Y) = \beta_0 + (\beta_1 + \beta_3 x_2)x_1 + \beta_2 x_2$$

where we see that the effect of x_1 equals $\beta_1 + \beta_3 x_2$. That is, the effect of x_1 is dependent on the value of x_2.

This factorisation is also useful when considering the interpretation of each of the regression coefficients β_0, β_1, β_2 and β_3. These are:

- β_0: the mean of Y when $x_1 = x_2 = 0$
- β_1: the effect of x_1 when $x_2 = 0$. i.e. The mean of Y will increase by β_1 for every one unit increase in x_1 when $x_2 = 0$.
- β_2: the effect of x_2 when $x_1 = 0$. i.e. The mean of Y will increase by β_2 for every one unit increase in x_2 when $x_1 = 0$.
- β_3: How the effect of x_1 changes for every one unit increase in x_2. Or alternatively, how the effect of x_2 changes for every one unit increase in x_1.

Importantly, the statistical test for trend here is the test with the null hypothesis $\beta_3 = 0$. Therefore the P-value for β_3 should be interpreted as the evidence for interaction between x_1 and x_2.

Interaction in statistical software

There are two approaches for implementing an interaction regression model in statistical software. The first is to manually create a new column of data which is the product of the two columns you wish to test for interaction for. You can then include this new variable as a covariate in the regression. The second method is to allow Stata or R to do this automatically, by specifying the interaction in the regression formula. The second method is always recommended as interaction involving categorical variables of more than two categories requires more than just creating a product of two covariates, but instead creating a product of all dummy variables involved in the categorical variable.

Stata

In Stata the hash symbol `#` is used to specify interaction. E.g. `reg Y x1 x2 x1#x2` would be a regression model with outcome Y, covariates x_1 and x_2 and interaction between x_1 and x_2. Alternatively you can use the double hash `##` as a shorthand to specify the inclusion of both covariates and interaction between them. i.e. `reg Y x1 x2 x1#x2` is equivalent to `reg Y x1##x2`

R

In R the colon symbol `:` is used to specify interaction. E.g. `lm(Y ~ x1 + x2 + x1:x2, data=...)` would be a regression model with outcome Y, covariates x_1 and x_2 and interaction between x_1 and x_2. Alternatively you can use the star `*` as a shorthand to specify the inclusion of both covariates and interaction between them. i.e. `lm(Y ~ x1 + x2 + x1*x2, data=...)` is equivalent to `lm(Y ~ x1:x2, data=...)`.

Book Chapter 4. Section 4.6.5 (pages 107-108).

This small reading provides brief descriptions of several important factors to consider when using a regression model with interaction.

Example

We will show two examples from the textbook that demonstrate interaction. These are examples 4.6.1 (interaction between two binary variables) and 4.6.2 (interaction between a binary variable and a continuous variable). You can also of course have interaction between two continuous variables.

Example 4.6.1

We return to the HERS dataset with low-density lioprotein (`LDL`) as the outcome and statin use (`statin`) and hormone therapy (`HT`) as the two binary covariates. Note that the results below will differ to those in the text book as the textbook uses `LDL1` as the outcome instead of `LDL`. The regression model with interaction is:

Stata code and output

```
use hersdata, clear
reg LDL i.HT i.statins  i.HT#i.statins
##       Source |       SS           df       MS      Number of obs   =     2,752
## -------------+----------------------------------   F(3, 2748)      =     44.06
##        Model |  180425.141         3  60141.7137   Prob > F        =    0.0000
##     Residual |  3750983.32     2,748  1364.98665   R-squared       =    0.0459
## -------------+----------------------------------   Adj R-squared   =    0.0449
##        Total |  3931408.46     2,751  1429.08341   Root MSE        =    36.946
##
## ------------------------------------------------------------------------------
##                  LDL | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## ---------------------+--------------------------------------------------------
```

```
##                     HT |
##     hormone therapy    |   .6656023   1.764553     0.38   0.706    -2.794382    4.125587
##                        |
##               statins  |
##                  yes   |   -15.8714   2.055967    -7.72   0.000     -19.9028   -11.84001
##                        |
##           HT#statins   |
## hormone therapy#yes    |  -1.918487   2.931023    -0.65   0.513    -7.665717    3.828743
##                        |
##                 _cons  |   150.7981   1.257647   119.90   0.000     148.3321    153.2642
## ------------------------------------------------------------------------------
```

R code and output

```
hers <- read.csv("https://www.dropbox.com/s/7f5lnv19drg6655/hersdata.csv?dl=1")
## Warning in file(file, "rt"): cannot open URL
## 'https://www.dropbox.com/s/dl/7f5lnv19drg6655/hersdata.csv': HTTP status was
## '400 Bad Request'
## Error in file(file, "rt"): cannot open the connection to 'https://www.dropbox.com/s/7f5
lm.hers <- lm(LDL ~ HT + statins + HT:statins, data = hers)
## Error in eval(mf, parent.frame()): object 'hers' not found
summary(lm.hers)
## Error in eval(expr, envir, enclos): object 'lm.hers' not found
confint(lm.hers)
## Error in eval(expr, envir, enclos): object 'lm.hers' not found
```

You may notice here that the R and Stata output have some differences as they choose different reference categories for the HT variable (Stata chose the placebo group as the first observation in the dataset is from the placebo group, and R chose the hormone therapy group as "hormone therapy" is before "placebo" when arranging in alphabetical order). In both instances here however the primary result is the same: "There is no evidence of interacation between HT and statin use (P = 0.513)". As there is no evidence for interaction, you would then proceed with a non-interaction regression model.

Example 4.6.2

Using the same HERS data as example 4.6.1 we now investigate possible interaction between statin use (binary variable) and body mass index (BMI - a continuous variable). In this example, we also adjust for several other covariates including `age`, `nonwhite`, `smoking` and `drinkany`. We also use a centered version of the BMI variable. Unlike in example 4.6.1 our results should match the textbook as we both use the `LDL` variable.

Stata code and output

```
use hersdata, clear
gen BMIc = BMI - 28.57925
regress LDL i.statins##c.BMIc age nonwhite smoking drinkany
## (5 missing values generated)
##
##
##       Source |       SS           df       MS      Number of obs   =     2,745
## -------------+----------------------------------   F(7, 2737)      =     22.85
##        Model |  216681.484         7  30954.4978   Prob > F        =    0.0000
##     Residual |     3707501     2,737  1354.58568   R-squared       =    0.0552
## -------------+----------------------------------   Adj R-squared   =    0.0528
##        Total |  3924182.49     2,744  1430.09566   Root MSE        =    36.805
##
## ------------------------------------------------------------------------------
##            LDL | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## ---------------+--------------------------------------------------------------
##        statins |
##           yes  |  -16.25301   1.468788   -11.07   0.000    -19.13305   -13.37296
##           BMIc |   .5821275    .160095     3.64   0.000     .2682082    .8960468
##                |
## statins#c.BMIc |
##           yes  |   -.701947   .2693752    -2.61   0.009    -1.230146   -.1737478
##                |
##           age  |  -.1728526   .1105696    -1.56   0.118    -.3896608    .0439556
##       nonwhite |   4.072767   2.275126     1.79   0.074    -.3883704    8.533903
##        smoking |   3.109819    2.16704     1.44   0.151    -1.139381    7.359019
##       drinkany |  -2.075282   1.466581    -1.42   0.157    -4.950999    .8004355
##          _cons |   162.4052   7.583312    21.42   0.000     147.5356    177.2748
## ------------------------------------------------------------------------------
```

R code and output

```
hers <- read.csv("https://www.dropbox.com/s/7f5lnv19drg6655/hersdata.csv?dl=1")
## Warning in file(file, "rt"): cannot open URL
## 'https://www.dropbox.com/s/dl/7f5lnv19drg6655/hersdata.csv': HTTP status was
## '400 Bad Request'
## Error in file(file, "rt"): cannot open the connection to 'https://www.dropbox.com/s/7f5
hers$BMIc <- hers$BMI - 28.57925
## Error in eval(expr, envir, enclos): object 'hers' not found
lm.hers <- lm(LDL ~ statins*BMIc + age + nonwhite + smoking + drinkany, data = hers)
```

```
## Error in eval(mf, parent.frame()): object 'hers' not found
summary(lm.hers)
## Error in eval(expr, envir, enclos): object 'lm.hers' not found
confint(lm.hers)
## Error in eval(expr, envir, enclos): object 'lm.hers' not found
```

We can see from the output above that there is strong evidence for interaction between body mass index and statin use, after adjusting for age, nonwhite, smoking, and drinkany covariates (P = 0.009). Given the strong evidence for interaction, we would decide to keep interaction in this model. This makes the interpretation of the base `statins` and `BMIc` variables different as follows:

- `statins`. When BMIc = 0, the those taking statins have a LDL on average 16.25mg/dL lower than those not taking statins. Here we can see the value of centering BMI, as BMIc = 0 corresponds to the effect of statin use for the mean BMI of this sample. Otherwise it would correspond to the effect of statins for a BMI of 0 (which is not appropriate to calculate)
- `BMIc`. For those not taking statins (i.e. statins = 0), for every 1kg/m^2 increase in BMI, the mean LDL increases by 0.58mg/dL.

To calculate the mean effect of BMI on those taking statins, we would need to take linear combinations of the above regression coefficients. In this case the mean effect would be 0.58 - 0.70 = -0.12. To get a more accurate calculation of this, along with the corresponding confidence interval use the `lincom BMIc + 1.statins#c.BMIc` statement in stata and the `summary(glht(lm.hers, "BMIc + statinsyes:BMIc = 0"))` command in R (you can swap "summary" with "confint" in this statement to obtain the confidence intervals).

Summary

This weeks key concepts are:

1. Collinearity occurs when two or more covariates in a regression model are associated with each other, and do not have sufficient independent associations with the outcome
2. Collinearity increases standard errors of regression coefficients
3. You should check for collinearity when carrying out linear regression. If detected, the effects of collinearity can be determined by removing some of the collinear covariates.
4. Interaction between two covariates occurs when the effect size for variable 1, depends on the level of covariate 2. This is also called effect modification.

5. Interaction between two variables in regression can be tested by including an additional covariate in your regression model that is the multiplication of your two covariates. If one or more of these covariates is categorical (with more than 2 categories), this will be the addition of several interaction terms between all dummy variables.
6. The interpretation of regression models with interaction terms is more complex as effect sizes are not constant for interacting covariates.

7 Violations of assumptions

Learning objectives

By the end of this week you should:

1. Be able to run polynomial regression
2. Know about restricted cubic splines and how they can model nonlinearities
3. Have a general understanding of the use of flexible techniques to evaluate the shape of a regression function
4. Be familar with the bootstrap and its use in linear regression
5. Understand how heteroscedasticity can be handled in practice

Learning activities

This week's learning activities include:

Learning Activity	Learning objectives
Lecture 1	1, 2, 3
Reading	4
Lecture 2	4, 5
Investigations	2, 4

Lecture 1 in R

Download video here

Lecture 1 in Stata

Download video here

We have seen how to use residuals to identify non-linearities, this week we will first revisit the issue and introduce new modelling techniques to deal with nonlinear associations.

Polynomial regression

To motivate, we consider data on 892 females under 50 years collected in three villages in West Africa. Investigators are interested in exploring the relationship between age and triceps skinfold thickness, a crude measure of body fat. The dataset is called *triceps* and include *age*, *thick* and *logthick*, for respectively age in years, triceps skinfold thickness in mm and its logarithm. The figure below displays the the relationship between *age* and *logthick* and it is immediately clear that this relationship in nonlinear on such a wide age span. A nonparametric smoother (LOWESS) can be applied to the data and returns the blue fitted line confirming the visual impression.

```
Warning in file(file, "rt"): cannot open URL
'https://www.dropbox.com/s/dl/5yk6u248adycvq1/triceps.csv': HTTP status was
'400 Bad Request'
```

```
Error in file(file, "rt"): cannot open the connection to 'https://www.dropbox.com/s/5yk6u248a
```

```
Error in eval(expr, envir, enclos): object 'triceps' not found
```

```
Error in eval(m$data, eframe): object 'triceps' not found
```

```
Error in eval(x[[2L]], environment(x)): object 'triceps' not found
```

A simple regression model of the form $logthick_i = \beta_0 + \beta_1 age_i + \epsilon_i$ for $i = 1, \ldots, n = 892$ would hardly give an approximation of the observed pattern, so the question arises: how can we modify it to make it more realistic? The first strategy is to replace the linear function of age by a cubic polynomial: $f(age) = \beta_0 + \beta_1 age + \beta_2 age^2 + \beta_2 age^3$. The term "cubic'' is used because we are fitting a polynomial of degree 3 due to the largest exponent being 3. A quadratic curve could also be fitted but would not be appropriate given the shape of the plot. Such a model is naturally called polynomial regression, irrespective of the degree of the polynomial function $f(x)$. Although we can define by hand the quadratic and cubic terms it's preferable to use the built-in tool *poly(x,3)* in R to define the polynomial (here x refers to the variable of interest, e.g. $x = age$). This leads to a simpler univariate smooth model of the form: $y_i = f(x_i) + \epsilon_i$ where y is the generic term for the response ($y = logthick$ in our example).

The following fit is now obtained and there is clearly some improvement over a linear fit but we may be left with the visual impression that fitting a cubic polynomial is not quite enough to capture the structure of the data, especially around year 10 where the fit does not seem so good.

```
Error in eval(expr, envir, enclos): object 'triceps' not found
```

```
Error in eval(m$data, eframe): object 'triceps' not found
```

```
Error in eval(mf, parent.frame()): object 'triceps' not found
```

```
Error in eval(mf, parent.frame()): object 'triceps' not found
```

```
Error in eval(expr, envir, enclos): object 'triceps' not found
```

Although all polynomial coefficients are significant - results not shown -, we should *not* use significance to decide to include a particular term or not. Another parametrisation based on orthogonal polynomials is often preferable as it is more stable, especially for high-degree polynomials. Using this parametrisation which is the default parametrisation in R yields a non-significant quadratic term. It is nevertheless needed to give the same fit. The F-test (or a LRT test) can help decide whether the cubic polynomial in age is necessary and is, for instance, better than a quadratic polynomial or a simple linear fit. We leave this as an exercise.

The R code for this section and a slightly simpler version for Stata are given below:

R Code

```
Warning in file(file, "rt"): cannot open URL
'https://www.dropbox.com/s/dl/5yk6u248adycvq1/triceps.csv': HTTP status was
'400 Bad Request'
```

```
Error in file(file, "rt"): cannot open the connection to 'https://www.dropbox.com/s/5yk6u248a
```

```
Error in eval(expr, envir, enclos): object 'triceps' not found
```

```
Error in eval(m$data, eframe): object 'triceps' not found
```

```
Error in eval(x[[2L]], environment(x)): object 'triceps' not found
```

```
Error in eval(expr, envir, enclos): object 'triceps' not found
```

```
Error in eval(m$data, eframe): object 'triceps' not found
```

```
Error in eval(mf, parent.frame()): object 'triceps' not found
```

```
Error in eval(mf, parent.frame()): object 'triceps' not found
```

```
Error in eval(expr, envir, enclos): object 'triceps' not found
```

Stata code and output

```
import delimited "https://www.dropbox.com/s/5yk6u248adycvq1/triceps.csv?dl=1"
gen age2 = age^2
gen age3 = age^3
regress logthick age age2 age3
quiet predict pred
twoway  scatter logthick age || line pred age, sort
test age2 age3
## compare cubic polynomial fit to linear fit
## server refused to send file
## could not open url
## r(603);
##
## r(603);
```

Restricted cubic splines

The disadvantage of polynomials is that they tend to fit the data globally and are not very good locally, for instance around year 10 for *logthick*. One way to get around this issue is to partition the range of *age* (or the covariate x in general) into smaller intervals, utilising a small number of points called *knots*. We can then fit (cubic) polynomials locally but by doing this we will get discontinuities. A simple remedy is to impose constraints on these polynomials so that the global result is a *smooth* function. We typically do this by forcing the successive derivatives of the function to be continuous up to some order (e.g. 2). The resulting function is called a spline. Different splines exist and a common type consists of portions of cubic polynomials in the middle joined together at the knots that become linear in the edges (i.e before the first knot and after the last one). Such splines are called Restricted Cubic Splines (RCS) with RCS(4) denoting a RCS with 4 knots. Restrictions are imposed to the different parts to yield a smooth function. It is not necessary to know their complex algebraic expression but we can see how the look on Figure 7.1 below:

```
Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
logical.return = TRUE, : there is no package called 'rms'
```

```
Error in rcspline.eval(age.grid, knots = kk, nk = 5, inclx = TRUE): could not find function "
```

```
Error in eval(expr, envir, enclos): object 'tt' not found
```

```
Error in eval(expr, envir, enclos): object 'tt' not found
```

```
Error in eval(expr, envir, enclos): object 'tt' not found
```

```
Error in eval(expr, envir, enclos): object 'tt' not found
```

```
Error: object 'triceps' not found
```

```
Error in ols(logthick ~ rcs(age, kk), data = triceps): could not find function "ols"
```

```
Error in eval(expr, envir, enclos): object 'fit.rcs' not found
```

```
Error in eval(expr, envir, enclos): object 'pred.rcs' not found
```

** Figure 7.1:** Spline basis functions and fitted line

There are 4 terms because we choose 5 knots (i.e, d.f. = number of knots minus 1) noted $S_1(age) = age$ in blue, $S_2(age)$ in red, $S_3(age)$ in magenta, $S_4(age)$ in green on the left panel. Each individual spline function (called spline basis) should not be interpreted individually. What matters is their "combined effect", i.e the function $f(age) = b_0 + b_1 S_1(age) + b_2 S_2(age) + b_3 S_3(age) + b_4 S_4(age)$ which is very smooth and seems to capture well the features of the data. Here the coefficients $b_0, b_1, \ldots, b_4$ where calculated by fitting $f(a)$ to the data. Such a fit is possible using standard commands since the function *linear in the coefficients* despite its *nonlinearity in age*.

Syntax and outputs

This is how you can fit such a model in R. This requires a slightly different syntax from what you have been using so far. R makes use of the *rms* library and the command *ols* instead of *lm*. The spline function itself is specified by adding *rcs* within the formula yielding:

```
require(rms)
```

```
Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
logical.return = TRUE, : there is no package called 'rms'
```

```
ddist <- datadist(triceps)
```

```
Error in datadist(triceps): could not find function "datadist"
```

```
options(datadist='ddist')
fit.rcs <- ols(logthick ~ rcs(age,4),data=triceps)
```

```
Error in ols(logthick ~ rcs(age, 4), data = triceps): could not find function "ols"
```

The command *rcs(age,4)* specifies a RCS in age with 4 knots placed at their default location. Note that the first two lines after calling the *rms* library are not absolutely needed to get the fit but there will be used to plot the splines, so it's recommended to add them anyway. The output is given below:

```
fit.rcs
## Error in eval(expr, envir, enclos): object 'fit.rcs' not found
```

You may notice that the syntax is slightly different from the standard way to display an output after typing the *lm* command since *summary()* is not used. The display is rather unsual and indicates the coefficients for the various spline bases with the notation age' and age'' corresponding to two additional splines terms $S_2(a)$ and $S_3(a)$ that are never interpreted separarely.

The Stata syntax still uses the usual *regress* command but it's preceeded by a command *mkspline* explaining what splines need to be fitted. Below the syntax and output in Stata.

Stata code and output

```
import delimited "https://www.dropbox.com/s/5yk6u248adycvq1/triceps.csv?dl=1"
mkspline age_spl = age, cubic nknots(4)
regress logthick  age_spl*
## server refused to send file
## could not open url
## r(603);
##
## r(603);
```

The command *mkspline age_spl = age, cubic nknots(4)* asks Stata to create a RCS in age with 4 knots, Stata will generate the correspponding terms called *age_spl1*, *age_spl2* and *age_spl3*. Note that *age_spl1* is always *age* so that the syntax is equivalent to *regress logthick*

age age_spl1 age_spl2. The generic command with * makes sure that *all* terms are included. Some other options are available that you can explore using the help.

What remained to be discussed is: 1) how do we choose the knots?; 2) how many knots do we choose?; and possibly 3) how do we know that the splines are needed?

Choosing the knots and their number

Typically a small number of knots is recommended, say between 3 and 5. We chose 5 knots earlier because we have a lot of data but 4 is ofen appropriate. Their location is critical and up to the investigator. A default choice have been fortunately implemented in Stata and R at specific quantiles of the variable of interest. The two figures below gives you the data and 2 different fits, one with 4 knots located using the default (Figure 7.2) and at age 10, 20, 35, 45 (Figure 7.3). We added the vertical dotted lines to indicate their location.

```
Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
logical.return = TRUE, : there is no package called 'rms'

Error in eval(expr, envir, enclos): object 'triceps' not found

Error in datadist(triceps): could not find function "datadist"

Error in ols(logthick ~ rcs(age, 4), data = triceps): could not find function "ols"

Error in eval(expr, envir, enclos): object 'fit.rcs4a' not found

Error in eval(m$data, eframe): object 'triceps' not found

Error in eval(expr, envir, enclos): object 'triceps' not found

Error in eval(expr, envir, enclos): object 'triceps' not found
```

** Figure 7.2:** A RCS spline fit with 4 knots (default location)

```
Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
logical.return = TRUE, : there is no package called 'rms'

Error in datadist(triceps): could not find function "datadist"
```

```
Error in ols(logthick ~ rcs(age, c(10, 20, 35, 45)), data = triceps): could not find functio

Error in eval(expr, envir, enclos): object 'fit.rcs4b' not found

Error in eval(m$data, eframe): object 'triceps' not found

Error in eval(expr, envir, enclos): object 'triceps' not found

Error in eval(expr, envir, enclos): object 'triceps' not found

Error in eval(expr, envir, enclos): object 'triceps' not found

Error in eval(expr, envir, enclos): object 'triceps' not found

Error in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): plot.new has not been cal
```

** Figure 7.3:** A RCS spline fit with 4 knots at age 10, 20, 35 and 45.

We clearly see that the default choice (Figure 7.2) gives a better fit visually than the manual choice with knots at 10, 20, 35, 45 (Figure 7.3), the first two knots being too high. By default, the knots in Figure 7.2 are chosen at the $.05^{th}$, $.35^{th}$, $.65^{th}$ and $.95^{th}$ quantiles of *age*, e.g the first knot is at age 1.2775 (!), the second at 8.1955. Note in Figure 7.3 that the function is indeed linear for extreme age values (blue dotted line). This is also the case for Figure 7.2 but it's less apparent on the plot.

Do we need the splines? Which fit should we choose?

Because the first spline basis is always the variable of interest (here *age*) we can use a F-test to test whether the added terms are needed. This is typically carried out in R using the *anova* command following the fit obtained previously. The F-test concludes that the additional terms are indeed needed with either the default choice of knots or the one chosen by the investigator ($p< 0.0001$).

The R and Stata code to fit splines with explanation is below.

R Code

```r
# Figure 7.2
par(mfrow=c(1,1))
# sort the data by increasing age
triceps<-triceps[order(triceps$age),]
```

```
## Error in eval(expr, envir, enclos): object 'triceps' not found
# top panel: default
ddist <- datadist(triceps)
## Error in datadist(triceps): could not find function "datadist"
options(datadist='ddist')
fit.rcs4a <- ols(logthick ~ rcs(age,4),data=triceps)
## Error in ols(logthick ~ rcs(age, 4), data = triceps): could not find function "ols"
pred4a<-predict(fit.rcs4a)
## Error in eval(expr, envir, enclos): object 'fit.rcs4a' not found
plot(logthick~age, xlab="Age",ylab="Skinfold thickness (log)", data=triceps)
## Error in eval(m$data, eframe): object 'triceps' not found
lines(triceps$age,pred4a,  col="red", lwd=2,lty=2)
## Error in eval(expr, envir, enclos): object 'triceps' not found
abline(v=quantile(triceps$age,c(0.05,0.35,0.65,0.95)),lty=2,col="red")
## Error in eval(expr, envir, enclos): object 'triceps' not found

# Testing whether the additional splines terms are necessary
anova(fit.rcs4a)
## Error in eval(expr, envir, enclos): object 'fit.rcs4a' not found
# look at age nonlinear with 2 df. Highly significant. Splines necessary

# Figure 7.3
fit.rcs4b <- ols(logthick ~ rcs(age,c(10,20,35,45)),data=triceps)
## Error in ols(logthick ~ rcs(age, c(10, 20, 35, 45)), data = triceps): could not find fu
pred4b<-predict(fit.rcs4b)
## Error in eval(expr, envir, enclos): object 'fit.rcs4b' not found
plot(logthick~age, xlab="Age",ylab="Skinfold thickness (log)",data=triceps)
## Error in eval(m$data, eframe): object 'triceps' not found
lines(triceps$age,pred4b,  col="blue", lwd=2,lty=2)
## Error in eval(expr, envir, enclos): object 'triceps' not found
abline(v=c(10,20,35,45),lty=2,col="blue")
## Error in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): plot.new has not bee
mini<-max(which(triceps$age <10))
## Error in eval(expr, envir, enclos): object 'triceps' not found
maxi<-min(which(triceps$age >=45))-1
## Error in eval(expr, envir, enclos): object 'triceps' not found
lines(triceps$age[mini:maxi],pred4b[mini:maxi],  col="red", lwd=2,lty=1)
## Error in eval(expr, envir, enclos): object 'triceps' not found
# poor fit for lower age
# Figure without fancy colours, do not run the last 3 lines
```

```
# Testing whether the additional splines terms are necessary
anova(fit.rcs4b)
## Error in eval(expr, envir, enclos): object 'fit.rcs4b' not found
# look at age nonlinear with 2 df. Highly significant. Splines necessary
```

Indications on how to produce a simpler figure in R were also provided. We don't suggest you produce complex figures like this one. Even the vertical lines displaying the knots location are often omitted.

Stata code and output

```
## Figure 7.2, default knots, and fit
import delimited "https://www.dropbox.com/s/5yk6u248adycvq1/triceps.csv?dl=1"
mkspline age_spl = age, cubic nknots(4) displayknots
regress logthick  age_spl*
predict pred
twoway scatter logthick age, xline(1.243  8.1865  17.469 42.72) || line pred age, sort cls

# Testing whether the additional splines terms are necessary
test age_spl2 age_spl3

## Figure 7.3, knots at 10, 20, 35 and 45, and fit

drop pred age_spl*
mkspline age_spl = age, cubic knots(10 20 35 45)
matrix list r(knots)
regress logthick  age_spl*
predict pred
twoway scatter logthick age, xline(10 20 35 45) || line pred age, sort clstyle(solid)

# Testing whether the additional splines terms are necessary
test age_spl2 age_spl3
## Unknown #command
## server refused to send file
## could not open url
## r(603);
##
## r(603);
```

Now we have produced 3 different fits of the same data (ignoring the RCS(5) fit given to show what a spline function is): a) cubic polynomial; b) RCs(4), default knots; c) RCS(4), knots at age 10, 20 35, 45. Visually, it's clearly that b) gives the better fit, so this is the one we'll choose. There are many other fits you could get by playing with the number of knots and

their location, so you could add them to the list. There is a way to compare such models that are not nested, we defer the explanation to week 8 but hopefully the approach agrees with the visual impression.

Interpretation

There are several ways you can interpret the results after fitting a spline model to the data and concluding that splines are indeed necessary. Let's for instance consider the RCS(4) model used for the triceps data. The simplest (and possibly most common) way is to display a plot like Figure 7.2 or only the spline function with its 95% CI and interpret it *qualitatively*. Here log-thickness decreases with age until the age of 8-9 and starts to increase again to reach a maximum slighlty above 2.5 around 30-35. From then on, it does not change much with a very slow reduction as women get older (up to the maximum age in the data, i.e. 52).

The second way is to try to quantify the relationship. One might wonder what the average change in log-thickness per each additional year of age is? The problem is more complicated than usual as we have fitted a function of age, $f(age) = b_0 + b_1 age + b_2 S_2(age) + b_3 S_3(age)$. Any increment of age will also affect the spline functions $S_2(age)$ and $S_3(age)$ so we can not interpet each coefficient separately. However, the same principle applies and, at least theoretially, we can compute the marginal change in log-thickness associated with a i year increment. To make things simpler, let's choose a particular age, say 10. What we need to do is compute the (expected) change in log-thickness for a change in age from 10 to 11. This amounts to computing the difference: $d = f(11) - f(10) = b_1 + b_2[S_2(11) - S_2(10)] + b_3[S_3(11) - S_3(10)]$. This is a *linear* combination of the model coefficients b_1, b_2, b_3 so in principle we should be able to estimate this but the difficulty is to get the "weights" $w_2 = S_2(11) - S_2(10)$ and $w_3 = S_3(11) - S_3(10)$ that involve the spline basis functions. The weight of age in $f(age)$ is always one here by construction.

OPTIONAL: Regarding this calculation, Stata users have the edge since the software provides an approximation through the derivative since $f(11) - f(10) \simeq f'(10)$. In general, $f(a+1) - f(age)) \simeq f'(a)$ thanks to a Taylor expansion. The R users don't have this luxury and have to work out for themselves what this difference is. For that reason, what follows is considered optional but Stata users should not have any problem to run the few lines of code that are needed. Note that the splines have to be generated using the command *mkspline2* command instead of *mkspline*. This command and the one that computes the derivative are available once a user-defined code has been installed (type: *ssc install postrcspline*).

Stata code

```
import delimited "https://www.dropbox.com/s/5yk6u248adycvq1/triceps.csv?dl=1"
mkspline2 age_spl = age, cubic nknots(4)
regress logthick  age_spl*
mfxrcspline, gen(delta lower upper)
```

```
sort age

## ------------------------------------------------
## this additional command allows you to browse (only)
## the new variables created. Deactivated here.
## You can also browse the data manually
## --------------------------------

## br age delta lower upper
## server refused to send file
## could not open url
## r(603);
##
## r(603);
```

The code plots the derivative as a function of age and produces the marginal difference per additional year of age for each datapoint in the triceps dataset. The last two lines allow us to sort the data and list all the differenes (variable *delta* and *lower* and *upper* for the 95% CI). If we look at what was produced, we can see that for a women aged 10, the average change in log-thickness for a 1-year change in age is: 0.026, 95% CI=0.022 ; 0.0299). We choose 10 since it is an age that we have in the dataset to make things simpler. This can be obtained by looking at the results (type*br age delta lower upper* to see all differences). We typically talk about the rate of change in log-thickness associated with age. Had we chosen age=27, we would have obtained: 0.017, 95%CI=(0.0144 ; 0.0194). Of course, when age is beyond the last knot (i.e. 42.7 years) or prior to the first knot, the rate of change become constant since by definition the RCS is linear in the tails.

R-users have to work a lot harder to get something similar. The code below explains how it's done and give you some explanation. Results may be sligthly different since we are not approximating the difference here. An *optional* video is available so maybe R-users can watch this video first.

Lecture 1b in R [OPTIONAL - help with an interpretation that is not directly provided in R]

Download video here

R code

```
triceps<-read.csv("https://www.dropbox.com/s/5yk6u248adycvq1/triceps.csv?dl=1")
## Warning in file(file, "rt"): cannot open URL
## 'https://www.dropbox.com/s/dl/5yk6u248adycvq1/triceps.csv': HTTP status was
## '400 Bad Request'
```

```
## Error in file(file, "rt"): cannot open the connection to 'https://www.dropbox.com/s/5yk
triceps<-data.frame(triceps)
## Error in eval(expr, envir, enclos): object 'triceps' not found
triceps <- triceps[order(triceps$age),]
## Error in eval(expr, envir, enclos): object 'triceps' not found

# fit RCS(4) to the data
require(rms)
## Loading required package: rms
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'rms'
ddist <- datadist(triceps)
## Error in datadist(triceps): could not find function "datadist"
options(datadist='ddist')
fit.rcs4 <- ols(logthick ~ rcs(age,4),data=triceps)
## Error in ols(logthick ~ rcs(age, 4), data = triceps): could not find function "ols"
                                                        # Comment
                                                        # -------
age.val<-c(10)                                          # Choose a particular age, here 1
kk=attributes(rcs(triceps$age,4))$parms                 # get the knots
## Error in rcs(triceps$age, 4): could not find function "rcs"
tt<-rcspline.eval(age.val,knots=kk,nk=4,inclx=TRUE)     # evaluate the spline functions a
## Error in rcspline.eval(age.val, knots = kk, nk = 4, inclx = TRUE): could not find funct
tt1<-rcspline.eval(age.val+1,knots=kk,nk=4,inclx=TRUE)  # evaluate the spline functions a
## Error in rcspline.eval(age.val + 1, knots = kk, nk = 4, inclx = TRUE): could not find f
diff=tt1-tt                                             # compute the differences
## Error in eval(expr, envir, enclos): object 'tt1' not found
c(diff)
## [[1]]
## function (x, ...)
## UseMethod("diff")
## <bytecode: 0x112279230>
## <environment: namespace:base>

# these are the "weights"  you need to use in lincom or glht
# for coefficients beta1, beta2, beta3 (all splines terms)
# what I called w_1=1, w_2 and w_3. Note that w_1=1 because the
# first basis function is age, with the two others used to
# model non-linearities. In their absence we would go
# back to the usual interpretation and therefore w1=1
# is expected.
```

```r
require(multcomp)
## Loading required package: multcomp
## Loading required package: mvtnorm
## Loading required package: survival
## Loading required package: TH.data
## Loading required package: MASS
##
## Attaching package: 'TH.data'
## The following object is masked from 'package:MASS':
##
##     geyser
test <- rbind("test marginal effect at that age" = c(0,diff))
              # the leading 0 is for the intercept coeff (unused)
lincom<-glht(fit.rcs4, linfct=test)
## Error in eval(expr, envir, enclos): object 'fit.rcs4' not found
summary(lincom)
## Error in eval(expr, envir, enclos): object 'lincom' not found
# you can then get the 95% CI "by hand"
lower=0.032036-1.96*0.002092
upper=0.032036+1.96*0.002092
c(lower,upper)
## [1] 0.02793568 0.03613632

#  95% CI=( 0.02793568 ; 0.03613632)
# slightly different results in R (no approximation)

# you can repeat the process by choosing another age value (e.g. 27) but make sure your
# update the manual calculation of the 95% CI
```

Investigation

The objective of this activity is to become familiar with RCS using the *hers* data

1) Start by reading p. 113 - 114 of Vittinghof et al. (2012) where the authors model *HDL* as RCS of *BMI* adjusted for a few covariates, namely *age*, *nonwhite*, *smoking* and *drinkany* and fit a simple model with these covariates. We suspect the association with BMI may not be linear (see Table 4.20, p. 111 for more)

2) Fit a RCS in BMI with 4 knots while adjusting for the other covariates. Does it change the conclusion regarding the linearity of BMI? Test whether the additional spline terms in BMI are necessary for this data? Note that if you want to reproduce the results of

Table 4.21 p. 114, you need to use *age10* (instead of *age* or *agec*) and have 5 knots (not requested).

3) Plot the fitted line with its 95% band. To get a nice plot you need to fix the *other* covariates than BMI. One way to do this is to set these covariates at the mean or median value.

R users: The latter is provided by the R command *plot(Predict(fit, BMI))* where *fit* is the RCS fit obtained earlier. Stick to the plots obtained after fitting a RCS using *rms* and *ols* (and avoid using *lm*).

Stata users: You may have to be cautious here as there are other covariates than BMI. Again, follow the recommendation of Vittinghof et al. (2012) and use the *postrcspline* package. Recreate the splines using the command *mkspline2*. The command *adjustrcspline* used in combination with *at()* where you specify the other covariate values will give you a nice plot. The choice is up to you but we could use: the mean (or the median) age in the sample for age and 0 for nonwhite, smoking and drinkany.

4) Change the location of the 4 knots and refit the model. Try also a model with a different number of knots. Does it change your conclusion? Provide a qualitative interpretation of the BMI effect in the model you decide to keep.

5) Do we need a RCS model for age?

Fractional polynomials and other methods

RCS are not the only approach we could use to model nonlinearities. Other types of splines exist including smoothing splines that impose a penalty for a lack of smoothness, they are more nonparametric by nature. We would like to say a few words about fractional polynomials that were introduced by Royston and Atlman (1994) and further developed since then. For the sake of simplicity, we describe the procedure using a model with a single continuous covariate (called x). One way to generalise the linear combination $\beta_0 + \beta_1 x$ is to replace it by a special type of polynomial that might include logarithms, noninteger powers, and repeated powers. One can show that, With a suitable range of powers, such polynomials provide a considerable range of functional forms in x that are useful to model real data. They are called "Fractional Polynomials'' (FP) to distinguish from standard polynomials. The default set of powers from which FP powers are selected is typically $\{-2, -1, -0.5, 0, 0.5, 1, 2, 3\}$, with 0 signifying log. There is also the convention that, every time a power repeats in an FP function of x, it is multiplied by another $\log(x)$. To describe the FP we typically list the series of powers that will be used. For example, a FP in x with powers $(-1, 0, 0.5, 1)$ and coefficients β has the following form: $\beta_1 x^{-1} + \beta_2 \log(x) + \beta_3 x^{1/2} + \beta_4 x$; also, a FP with powers $(0, 1, 2, 2)$ will correspond to $\beta_1 \log(x) + \beta_2 x + \beta_3 x^2 + \beta_4 x^2 \log(x)$. Now the key issue will be to choose the best powers. Some automated procedures are available but we strongly suggest that you keep things simple

and use plots (possibly some obtained using splines) to guide your choice. We do not present here a full account of this approach available in Stata through the commands *fp* and *mfp* and in R via the library *mfp*; it's more a note mentioning that they exist and are part of a suite of techniques that could be used in model building. More will be said in week 11 in the context of logistic regression where we have more tools to decide on how to compare different models. Finally, splines and FP are not necessarily opposed. There will be situations like the triceps data where splines are more appropriate due the local flexibiliy they provide, others where FP can be valuably used to capture more general trends that can be extrapolated, and situations where they complement each other.

Lecture 2 in R

Download video here

Lecture 2 in Stata

Download video here

Other issues

Bootstrapping

IN PSI you learned about the bootstrap, a resampling technique widely applicable for inference purposes. This method is very important in regression since it does not rely on normality. Therefore it can be used in situations where the normality assumption is questionable and we have doubt about the validity of standard SEs and 95%CIs. In addition, by resampling the data (i.e. what is called case resampling), we can also obtain valid SEs and 95%CI for complex statistics for which nothing has been implemented in standard statistical packages.

[@vittinghoff2012] Chapter 3. Section 3.6 (pages 62-63).

Read this short introduction to the different bootstrap CIs that are routinely available in Stata and R, namely the *normal approximation*, the *percentile* method and the *Bias-Corrected and accelerated (BCa)* approach. To go beyond this succinct summary and get a bit of practice in Stata or R with this approach, we will conduct this investigation.

Bootstrap investigation

Click below to see a heavily-documented code illustrating how the resampling works on the *hers* data and how we can obtain bootstrap-based 95% CIs.

R Code

```r
# Part A)
require(haven)
## Loading required package: haven
hers<-read_dta("https://www.dropbox.com/s/ndtd4o20qogq7fv/hersdata.dta?dl=1")
## Error in open.connection(con, "rb"): HTTP error 400.
hers<-data.frame(hers)
## Error in eval(expr, envir, enclos): object 'hers' not found
hers.nondiab<-hers[hers$diabetes==0,]
## Error in eval(expr, envir, enclos): object 'hers' not found

# 1) standard analysis on reduced data + normal probability plot

hers1<-cbind(hers.nondiab$HDL,hers.nondiab$age,hers.nondiab$BMI,hers.nondiab$drinkany)
## Error in eval(expr, envir, enclos): object 'hers.nondiab' not found
colnames(hers1)<-c("HDL","age","BMI","drinkany")
## Error: object 'hers1' not found
hers1<-data.frame(hers1)
## Error in eval(expr, envir, enclos): object 'hers1' not found
hers2<-na.omit(hers1) # 2032 --> 2021
## Error in eval(expr, envir, enclos): object 'hers1' not found


# fit the linear model (HDL on age BMI and drinkany() and draw a normality plot of the res

# 2) Part A) - bootstrap "by hand"

# we assumed that the reduced dataset is called hers2
# 2021 observation, 4 columns if you keep only what you need
# (i.e. HDL, age, BM, drinkany)

set.seed(1001)
R=1000
n=dim(hers2)[1]
## Error in eval(expr, envir, enclos): object 'hers2' not found
all.replicates<-NULL
for(r in 1:R){
  # generate bootstrap sample by resampling the data
  hers2.r=hers2[sample(nrow(hers2), n,replace = TRUE), ]
  # fiited model (based on the bootsrap sample)
  out.r<-lm(HDL~age+BMI+drinkany,data=hers2.r)
  # store all coefficients in all replicates
```

```
  all.replicates=rbind(all.replicates,out.r$coeff)
}
## Error in eval(expr, envir, enclos): object 'hers2' not found

# all replicates is a matrix Rx4 (since we have R replicates
# and 4 coefficients in the model)

head(all.replicates)
dim(all.replicates)

# draw an histogram of the replicates + normal probability ploy
# for each coefficient
# histogram of the replicates + normal probabilty plot


# 3) percentiles 95% CI

# get the 2.5% and 95.% percentile for each column
# either directly (or using apply)

# directly


# 4) Part B) - bootstrap using the library boot
# -------------------------------------------

# We will now use the library boot to do the same job



library(boot)
##
## Attaching package: 'boot'
## The following object is masked from 'package:survival':
##
##     aml

# function collecting the coefficients. In general this function
# derive the statistic we want to bootstrap.

coeff<- function(data, indices){
```

```
  data <- data[indices,] # select obs. in bootstrap sample
  mod <- lm(HDL~age+BMI+drinkany, data=data) # modify formula here
  coefficients(mod) # return coefficient vector
}

# NB: R doc says or parametric bootstrap (i.e. the one we are using)
#  "the first argument to statistic must be the data"
#  " The second will be a vector of indices, frequencies or weights
#   which define the bootstrap sample".

# LS-based 95% CI

out<-lm(HDL~age+BMI+drinkany,data=hers2)
## Error in eval(mf, parent.frame()): object 'hers2' not found
confint(out)
## Error in eval(expr, envir, enclos): object 'out' not found

# bootsset.seed(1001)
B = boot(data=hers2,statistic=coeff,R=3000)
## Error in eval(expr, envir, enclos): object 'hers2' not found
    # increase the nb of replicates for BCA
    # (R=1000 is too small, numerical issues)

# 3 types for the 3rd coeff = BMI's
boot.ci(B,index=3,type="norm")   # normal
## Error in eval(expr, envir, enclos): object 'B' not found
boot.ci(B,index=3,type="perc")   # percentile
## Error in eval(expr, envir, enclos): object 'B' not found
boot.ci(B,index=3,type="bca")    # BCa CI (3 types_)
## Error in eval(expr, envir, enclos): object 'B' not found

# default coverage =95%

# all types in one command
boot.ci(B,index=3,type=c("norm","perc", "bca"))
## Error in eval(expr, envir, enclos): object 'B' not found


# plots
# -----
```

```r
plot(B, index=1) # intercept
## Error in eval(expr, envir, enclos): object 'B' not found
plot(B, index=2) # x1=age
## Error in eval(expr, envir, enclos): object 'B' not found
plot(B, index=3) # x2=BMI
## Error in eval(expr, envir, enclos): object 'B' not found
plot(B, index=4) # x3=drinkany
## Error in eval(expr, envir, enclos): object 'B' not found

# nicer plots
all<-B$t
## Error in eval(expr, envir, enclos): object 'B' not found
par(mfrow=c(1,2))
hist(all[,1],main="Histogram",xlab="Intercept",prob=TRUE)
## Error in all[, 1]: object of type 'builtin' is not subsettable
qqnorm(all[,1])
## Error in all[, 1]: object of type 'builtin' is not subsettable
hist(all[,2],main="Histogram",xlab="Age coeff",prob=TRUE)
## Error in all[, 2]: object of type 'builtin' is not subsettable
qqnorm(all[,2])
## Error in all[, 2]: object of type 'builtin' is not subsettable
# etc


# Alternative coding (part B): slower
# -----------------------------------

coeff<- function(data, indices, formula){
  data <- data[indices,] # select obs. in bootstrap sample
  mod <- lm(formula=formula, data=data)
  coefficients(mod) # return coefficient vector
}

# percentile 95% CI, 3rd coefficient (BMI)
set.seed(1001)
B = boot(data=hers2,statistic=coeff,R=3000, formula=HDL~age+BMI+drinkany)
## Error in eval(expr, envir, enclos): object 'hers2' not found
boot.ci(B,index=3,type="perc")
## Error in eval(expr, envir, enclos): object 'B' not found

# percentile 95% CI, 4tg coefficient (drinkany)
```

```
set.seed(1001)
B = boot(data=hers2,statistic=coeff,R=3000, formula=HDL~age+BMI+drinkany)
## Error in eval(expr, envir, enclos): object 'hers2' not found
boot.ci(B,index=4,type="perc")
## Error in eval(expr, envir, enclos): object 'B' not found
```

Stata code

```
## read data
use hersdata.dta
drop if diabetes ==1   // 731 obs deleted
drop if mi(HDL) | mi(BMI) | mi(age) | mi(drinkany) // 11 obs deleted
keep HDL BMI age drinkany

## Part A) = bootstrap "by hand"
## ============================

## Writing our own bootstrap program requires four steps.
##
## 1) In the first step we obtain initial estimates and store the results in a matrix,
## say observe. In addition, we must also note the number of observations used in the anal
## This information will be used when we summarize the bootstrap results.
##
## 2) Second, we write a program which we will call myboot that samples the
## data with replacement and returns the statistic of interest. In this step,
## we start by preserving the data with the preserve command, then take a bootstrap
## sample with bsample. bsample samples the data in memory with replacement,
## which is the essential element of the bootstrap. From the bootstrap sample
## we run our regression model and output the statistic of interest with the return
## scalar command. Note that when we define the program, program define myboot,
## we specify the rclass option; without that option, we would not be able to output
## the bootstrapped statistic. myboot concludes with the restore command,
## which returns the data to the original state (prior to the bootstrapped sample).
##
## 3) In the third step, we use the simulate prefix command along with myboot,
## which collects the statistic from the bootstrapped sample.
## We specify the seed and number of replications at this step, which coincide
## with those from the example above.
##
## 4) Finally, we use the bstat command to summarize the results.
## We include the initial estimates, stored in the matrix observe, and the
```

```
## sample size with the stat( ) and n() options, respectively.



## Step 1 - define model and store the coefficients via the observe command
regress HDL BMI age drinkany

matrix observe= (_b[_cons], _b[BMI], _b[age], _b[drinkany])
matrix list observe


## Step 2 - program to be repeated
capture program drop myboot2
program define myboot2, rclass
 preserve
  bsample
    regress HDL BMI age drinkany
    ## fit model, store coeff
      return scalar b0 = _b[_cons]
    return scalar b1 = _b[BMI]
    return scalar b2 = _b[age]
    return scalar b3 = _b[drinkany]
 restore
end

## Step 3 - simulation = resampling the data using the program myboot2, R=1000 replicates
simulate  b0=r(b0) b1=r(b1) b2=r(b2) b3=r(b3), reps(1000) seed(12345): myboot2

## Step 4 - compute 95% CIs
bstat, stat(observe) n(2021)
                    ## n = nb of observations --> CAUTION HERE
estat bootstrap, all

## NB: you can reduce the output of estat bootstrap by specifying
## the option (e.g. percentile) instead of all

estat bootstrap, percentile


## NB: you can change the number of replicates i.e. the argument of reps()
##     we need at least 1000 replicates for 95% CU
```

```
##     The seed use here is only there to replicate the simulations
##     if you don't specify a seed, a random seed will be chosen and different results
##     will be obtained each time (very similar though). The difference is due to the
##     Monte Carlo variability.



##  select the code above and run

## NB: browse the active dataset, the dimension and the columns. NO LONGER hers

desc
list if _n<20


#  percentile CI for each coefficient & histogram
#  ----------------------------------------------

## write a one line command for the histogram and another line for the percentile CI (per


##  4) boostrap  using the libary boot - use Part B below


# Part B) use a Stata command - SIMPLER
# =====================================


clear
## read the dataset again

use hersdata.dta
drop if diabetes ==1
drop if mi(HDL) | mi(BMI) | mi(age) | mi(drinkany)
keep HDL BMI age drinkany

bootstrap, reps(1000)  seed(12345): regress HDL BMI age drinkany
estat bootstrap, all
## all 3 types

bootstrap, reps(1000)  seed(12345): regress HDL BMI age drinkany
```

```
estat bootstrap, percentile
## percentile

bootstrap, reps(1000)  seed(12345): regress HDL BMI age drinkany
estat bootstrap, normal
## normal

## to get the BCa option alone, type this
bootstrap, bca reps(1000)  seed(12345): regress HDL BMI age drinkany
estat bootstrap

## again you can use more than 1000 replicates and change the seed

## Unknown #command
##
## (731 observations deleted)
##
## (11 observations deleted)
##
##
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
```

```
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
##
##       Source |       SS           df       MS      Number of obs   =     2,021
## -------------+----------------------------------   F(3, 2017)      =     46.91
##        Model |  24006.0684         3  8002.02279   Prob > F        =    0.0000
##     Residual |  344071.218     2,017  170.585631   R-squared       =    0.0652
## -------------+----------------------------------   Adj R-squared   =    0.0638
##        Total |  368077.286     2,020  182.216478   Root MSE        =    13.061
##
## ------------------------------------------------------------------------------
##          HDL | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##          BMI |  -.4036859   .0571063    -7.07   0.000    -.5156793   -.2916924
##          age |   .2086808   .0437416     4.77   0.000     .1228974    .2944643
##     drinkany |   4.502504   .5880671     7.66   0.000     3.349222    5.655787
##        _cons |   46.68225   3.571831    13.07   0.000     39.67739    53.68712
## ------------------------------------------------------------------------------
##
##
##
## observe[1,4]
##             c1          c2          c3          c4
## r1   46.682253  -.40368587   .20868084   4.5025044
##
## Unknown #command
##
## Unknown #command
##
## Unknown #command
##
##       Command: myboot2
##            b0: r(b0)
##            b1: r(b1)
##            b2: r(b2)
##            b3: r(b3)
##
```

```
## Simulations (1,000)
## ----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
## ..................................................    50
## ..................................................   100
## ..................................................   150
## ..................................................   200
## ..................................................   250
## ..................................................   300
## ..................................................   350
## ..................................................   400
## ..................................................   450
## ..................................................   500
## ..................................................   550
## ..................................................   600
## ..................................................   650
## ..................................................   700
## ..................................................   750
## ..................................................   800
## ..................................................   850
## ..................................................   900
## ..................................................   950
## .................................................. 1,000
##
## Unknown #command
##
## Bootstrap results                                     Number of obs = 2,021
##                                                       Replications  = 1,000
##
## ------------------------------------------------------------------------------
##              |   Observed   Bootstrap                         Normal-based
##              | coefficient  std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##           b0 |   46.68225   3.455292    13.51   0.000     39.91001     53.4545
##           b1 |  -.4036859   .0548886    -7.35   0.000    -.5112656   -.2961062
##           b2 |   .2086808   .0430616     4.85   0.000     .1242817      .29308
##           b3 |   4.502504   .5957828     7.56   0.000     3.334792    5.670217
## ------------------------------------------------------------------------------
##
## Unknown #command
##
## Bootstrap results                               Number of obs     =      2,021
```

```
##                                                 Replications      =       1000
##
## ------------------------------------------------------------------------------
##              |    Observed                Bootstrap
##              | coefficient       Bias    std. err.  [95% conf. interval]
## -------------+----------------------------------------------------------------
##           b0 |   46.682253   .1759177   3.4552918    39.91001    53.4545   (N)
##              |                                       40.42489   54.07221   (P)
##              |                                       39.84275   53.29212  (BC)
##           b1 |  -.40368587  -.0038854    .0548886   -.5112656  -.2961062   (N)
##              |                                      -.5188768  -.2992493   (P)
##              |                                       -.507547  -.2924751  (BC)
##           b2 |   .20868084  -.0009771   .04306157    .1242817     .29308   (N)
##              |                                        .117856   .2910675   (P)
##              |                                       .1173203   .2900742  (BC)
##           b3 |   4.5025044   .0024019   .59578279    3.334792   5.670217   (N)
##              |                                       3.368253   5.692966   (P)
##              |                                       3.405528   5.748656  (BC)
## ------------------------------------------------------------------------------
## Key:  N: Normal
##       P: Percentile
##      BC: Bias-corrected
##
## Unknown #command
## Unknown #command
##
## Bootstrap results                               Number of obs     =      2,021
##                                                 Replications      =       1000
##
## ------------------------------------------------------------------------------
##              |    Observed                Bootstrap
##              | coefficient       Bias    std. err.  [95% conf. interval]
## -------------+----------------------------------------------------------------
##           b0 |   46.682253   .1759177   3.4552918    40.42489   54.07221   (P)
##           b1 |  -.40368587  -.0038854    .0548886   -.5188768  -.2992493   (P)
##           b2 |   .20868084  -.0009771   .04306157     .117856   .2910675   (P)
##           b3 |   4.5025044   .0024019   .59578279    3.368253   5.692966   (P)
## ------------------------------------------------------------------------------
## Key: P: Percentile
##
## Unknown #command
## Unknown #command
```

```
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
##
## Contains data
##  Observations:         1,000                  simulate: myboot2
##     Variables:             4                  19 Feb 2024 14:45
## ------------------------------------------------------------------------------
## Variable      Storage   Display    Value
##     name         type    format    label      Variable label
## ------------------------------------------------------------------------------
## b0              float   %9.0g                 r(b0)
## b1              float   %9.0g                 r(b1)
## b2              float   %9.0g                 r(b2)
## b3              float   %9.0g                 r(b3)
## ------------------------------------------------------------------------------
## Sorted by:
##
##
##       +-----------------------------------------------+
##       |       b0          b1         b2         b3 |
##       |-----------------------------------------------|
##    1. |  55.0277    -.507547   .1273913   4.579051 |
##    2. | 43.05033   -.3214125   .2198346    4.56848 |
##    3. | 41.79856   -.3580904   .2595156   4.108671 |
##    4. | 47.37144   -.3629469   .1825078   4.238028 |
##    5. | 45.83468   -.3676654   .2150579   3.279417 |
##       |-----------------------------------------------|
##    6. | 43.68774   -.4242001   .2591675   4.742793 |
##    7. | 46.68452   -.4392134   .2143669   5.071074 |
##    8. | 46.22334   -.3971224   .2033755   5.373112 |
##    9. | 45.68821   -.4942878   .2594136   4.191213 |
##   10. |  45.2424   -.3911622   .2145017   5.476406 |
##       |-----------------------------------------------|
##   11. |  48.1397   -.3311509   .1490672   5.302119 |
##   12. | 50.02382    -.433796   .1687294   5.122212 |
##   13. | 41.93714   -.3431965   .2403703   5.655903 |
##   14. | 43.53362   -.3849073   .2465488   4.113903 |
```

```
##   15. |   45.835    -.441162   .2361516   4.045438 |
##       |-----------------------------------------------|
##   16. | 46.65306   -.4823789   .2538138   3.141874 |
##   17. | 44.25373   -.3605336   .2135229   5.084305 |
##   18. | 49.38696   -.4354429   .1772449   5.243911 |
##   19. | 44.58269   -.4229109   .2491518   4.564682 |
##       +-----------------------------------------------+
##
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
## Unknown #command
##
## Unknown #command
##
## (731 observations deleted)
##
## (11 observations deleted)
##
##
## (running regress on estimation sample)
##
## Bootstrap replications (1,000)
## ----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
## ..................................................    50
## ..................................................   100
## ..................................................   150
## ..................................................   200
## ..................................................   250
## ..................................................   300
## ..................................................   350
## ..................................................   400
## ..................................................   450
## ..................................................   500
## ..................................................   550
## ..................................................   600
## ..................................................   650
## ..................................................   700
## ..................................................   750
```

```
## ..................................................   800
## ..................................................   850
## ..................................................   900
## ..................................................   950
## .................................................. 1,000
##
## Linear regression                                      Number of obs =   2,021
##                                                        Replications  =   1,000
##                                                        Wald chi2(3)  =  122.77
##                                                        Prob > chi2   =  0.0000
##                                                        R-squared     =  0.0652
##                                                        Adj R-squared =  0.0638
##                                                        Root MSE      = 13.0608
##
## ------------------------------------------------------------------------------
##              |   Observed   Bootstrap                         Normal-based
##          HDL | coefficient  std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##          BMI |  -.4036859   .0548886    -7.35   0.000    -.5112656   -.2961062
##          age |   .2086808   .0430616     4.85   0.000     .1242817      .29308
##     drinkany |   4.502504   .5957828     7.56   0.000     3.334792    5.670217
##        _cons |   46.68225   3.455292    13.51   0.000     39.91001     53.4545
## ------------------------------------------------------------------------------
##
##
## Linear regression                                 Number of obs     =      2,021
##                                                   Replications      =       1000
##
## ------------------------------------------------------------------------------
##              |    Observed               Bootstrap
##          HDL | coefficient       Bias    std. err.  [95% conf. interval]
## -------------+----------------------------------------------------------------
##          BMI |  -.40368587  -.0038854    .0548886   -.5112656  -.2961062   (N)
##              |                                      -.5188768  -.2992493   (P)
##              |                                       -.507547  -.2924751  (BC)
##          age |   .20868084  -.0009771   .04306157    .1242817     .29308   (N)
##              |                                        .117856   .2910675   (P)
##              |                                       .1173203   .2900742  (BC)
##     drinkany |   4.5025044   .0024019   .59578279    3.334792   5.670217   (N)
##              |                                       3.368253   5.692966   (P)
##              |                                       3.405528   5.748656  (BC)
```

```
##        _cons |   46.682253   .1759177   3.4552918    39.91001    53.4545   (N)
##              |                                       40.42489   54.07221   (P)
##              |                                       39.84275   53.29212  (BC)
## ------------------------------------------------------------------------------
## Key:  N: Normal
##       P: Percentile
##      BC: Bias-corrected
##
## Unknown #command
## (running regress on estimation sample)
##
## Bootstrap replications (1,000)
## ----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
## .................................................    50
## .................................................   100
## .................................................   150
## .................................................   200
## .................................................   250
## .................................................   300
## .................................................   350
## .................................................   400
## .................................................   450
## .................................................   500
## .................................................   550
## .................................................   600
## .................................................   650
## .................................................   700
## .................................................   750
## .................................................   800
## .................................................   850
## .................................................   900
## .................................................   950
## ................................................. 1,000
##
## Linear regression                               Number of obs =   2,021
##                                                 Replications  =   1,000
##                                                 Wald chi2(3)  =  122.77
##                                                 Prob > chi2   =  0.0000
##                                                 R-squared     =  0.0652
##                                                 Adj R-squared =  0.0638
##                                                 Root MSE      = 13.0608
```

```
##
## ------------------------------------------------------------------------------
##              |   Observed   Bootstrap                         Normal-based
##          HDL | coefficient  std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##          BMI |  -.4036859   .0548886    -7.35   0.000    -.5112656   -.2961062
##          age |   .2086808   .0430616     4.85   0.000     .1242817      .29308
##     drinkany |   4.502504   .5957828     7.56   0.000     3.334792    5.670217
##        _cons |   46.68225   3.455292    13.51   0.000     39.91001     53.4545
## ------------------------------------------------------------------------------
##
##
## Linear regression                               Number of obs     =      2,021
##                                                 Replications      =       1000
##
## ------------------------------------------------------------------------------
##              |    Observed               Bootstrap
##          HDL | coefficient       Bias    std. err.  [95% conf. interval]
## -------------+----------------------------------------------------------------
##          BMI |  -.40368587  -.0038854    .0548886   -.5188768  -.2992493   (P)
##          age |   .20868084  -.0009771   .04306157     .117856   .2910675   (P)
##     drinkany |   4.5025044   .0024019   .59578279    3.368253   5.692966   (P)
##        _cons |   46.682253   .1759177   3.4552918    40.42489   54.07221   (P)
## ------------------------------------------------------------------------------
## Key: P: Percentile
##
## Unknown #command
## (running regress on estimation sample)
##
## Bootstrap replications (1,000)
## ----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
## ..................................................    50
## ..................................................   100
## ..................................................   150
## ..................................................   200
## ..................................................   250
## ..................................................   300
## ..................................................   350
## ..................................................   400
## ..................................................   450
## ..................................................   500
## ..................................................   550
```

```
## ..................................................   600
## ..................................................   650
## ..................................................   700
## ..................................................   750
## ..................................................   800
## ..................................................   850
## ..................................................   900
## ..................................................   950
## .................................................. 1,000
##
## Linear regression                                      Number of obs =   2,021
##                                                        Replications  =   1,000
##                                                        Wald chi2(3)  =  122.77
##                                                        Prob > chi2   =  0.0000
##                                                        R-squared     =  0.0652
##                                                        Adj R-squared =  0.0638
##                                                        Root MSE      = 13.0608
##
## ------------------------------------------------------------------------------
##              |   Observed   Bootstrap                         Normal-based
##          HDL | coefficient  std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##          BMI |  -.4036859   .0548886    -7.35   0.000    -.5112656   -.2961062
##          age |   .2086808   .0430616     4.85   0.000     .1242817      .29308
##     drinkany |   4.502504   .5957828     7.56   0.000     3.334792    5.670217
##        _cons |   46.68225   3.455292    13.51   0.000     39.91001     53.4545
## ------------------------------------------------------------------------------
##
##
## Linear regression                                 Number of obs     =      2,021
##                                                   Replications      =       1000
##
## ------------------------------------------------------------------------------
##              |    Observed               Bootstrap
##          HDL | coefficient       Bias    std. err.  [95% conf. interval]
## -------------+----------------------------------------------------------------
##          BMI |  -.40368587  -.0038854    .0548886   -.5112656  -.2961062   (N)
##          age |   .20868084  -.0009771   .04306157    .1242817     .29308   (N)
##     drinkany |   4.5025044   .0024019   .59578279    3.334792   5.670217   (N)
##        _cons |   46.682253   .1759177   3.4552918    39.91001    53.4545   (N)
## ------------------------------------------------------------------------------
```

```
## Key: N: Normal
##
## Unknown #command
## Unknown #command
## (running regress on estimation sample)
##
## Jackknife replications (2,021)
## ----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
## ..................................................    50
## ..................................................   100
## ..................................................   150
## ..................................................   200
## ..................................................   250
## ..................................................   300
## ..................................................   350
## ..................................................   400
## ..................................................   450
## ..................................................   500
## ..................................................   550
## ..................................................   600
## ..................................................   650
## ..................................................   700
## ..................................................   750
## ..................................................   800
## ..................................................   850
## ..................................................   900
## ..................................................   950
## .................................................. 1,000
## .................................................. 1,050
## .................................................. 1,100
## .................................................. 1,150
## .................................................. 1,200
## .................................................. 1,250
## .................................................. 1,300
## .................................................. 1,350
## .................................................. 1,400
## .................................................. 1,450
## .................................................. 1,500
## .................................................. 1,550
## .................................................. 1,600
## .................................................. 1,650
```

```
## ................................................. 1,700
## ................................................. 1,750
## ................................................. 1,800
## ................................................. 1,850
## ................................................. 1,900
## ................................................. 1,950
## ................................................. 2,000
## ..................
##
## Bootstrap replications (1,000)
## ----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
## ..................................................    50
## ..................................................   100
## ..................................................   150
## ..................................................   200
## ..................................................   250
## ..................................................   300
## ..................................................   350
## ..................................................   400
## ..................................................   450
## ..................................................   500
## ..................................................   550
## ..................................................   600
## ..................................................   650
## ..................................................   700
## ..................................................   750
## ..................................................   800
## ..................................................   850
## ..................................................   900
## ..................................................   950
## ................................................. 1,000
##
## Linear regression                                 Number of obs =   2,021
##                                                   Replications  =   1,000
##                                                   Wald chi2(3)  =  122.77
##                                                   Prob > chi2   =  0.0000
##                                                   R-squared     =  0.0652
##                                                   Adj R-squared =  0.0638
##                                                   Root MSE      = 13.0608
##
## ------------------------------------------------------------------------------
```

```
##              |   Observed   Bootstrap                         Normal-based
##          HDL | coefficient  std. err.      z    P>|z|     [95% conf. interval]
## -------------+------------------------------------------------------------------
##          BMI |  -.4036859   .0548886    -7.35   0.000    -.5112656   -.2961062
##          age |   .2086808   .0430616     4.85   0.000     .1242817      .29308
##     drinkany |   4.502504   .5957828     7.56   0.000     3.334792    5.670217
##        _cons |   46.68225   3.455292    13.51   0.000     39.91001     53.4545
## ------------------------------------------------------------------------------
##
##
## Linear regression                               Number of obs     =      2,021
##                                                 Replications      =       1000
##
## ------------------------------------------------------------------------------
##              |    Observed               Bootstrap
##          HDL | coefficient       Bias    std. err.  [95% conf. interval]
## -------------+----------------------------------------------------------------
##          BMI |  -.40368587  -.0038854    .0548886    -.507547  -.2924751  (BC)
##          age |   .20868084  -.0009771   .04306157    .1173203   .2900742  (BC)
##     drinkany |   4.5025044   .0024019   .59578279    3.405528   5.748656  (BC)
##        _cons |   46.682253   .1759177   3.4552918    39.84275   53.29212  (BC)
## ------------------------------------------------------------------------------
## Key: BC: Bias-corrected
##
## Unknown #command
```

To understand better how the bootstrap works in linear regression, we ask to carry out the following tasks:

1) First, run a multiple linear regression of *HDL* on *BMI*, *age* and *drinkany* after removing diabetic patients. For simplicity we consider only 3 covariates but this can be extended to a more complex model. Create a *reduced dataset* with only these four variables and *no missing data* before any further treatment (since bootstrapping cannot accommodate missing data). You should get *n=2021* observations. Compute the residuals and observe that there is a bit of curvature on the normal QQ-plot.

2) Given the large sample size, normality is not that critical but we are going to check that inference is indeed valid using the bootstrap. Read Part A) of the code above explaining how we can resample the data and calculate R replicates (here $R = 1000$) of the coefficients. Run the corresponding code and draw a histogram of the bootstrap samples (or replicates) for each of the coefficients.

3) Can you provide the percentile 95% CI *directly* (using a one-line command in R/Stata) for each of the 3 coefficients? Compare with what Stata/R gives you using the code provided

(NB: R users will have to use the package *boot* to get results - see next question)

4) Fortunately, both Stata and R have a built-in command that avoids having to do the resampling "by hand''. Use Part B) of the code to provide the 3 types of bootstrap 95% cI. Do you see any meaningful difference with the standard LS analysis?

Heteroscedasticity

As discussed over the previous weeks, a critical assumption for the normal theory or even the bootstrap methods to be valid is the assumption of constant variance of the error term. What can we do if this assumption is not met? A standard way to deal with this issue (known as heteroscedasticity) is to transform the endpoint but such a variance-stabilising transformation does not always exists. Another way is to use weighted LS that provide valid inference in situations where the variance of the error term is $var(\varepsilon) = \sigma^2 W^{-1}$, the only issue with this approach is to find a proper symmetric matrix W often chosen diagonal. Its choice can be driven by plots, for instance plot of residuals vs covariate but it's not always straighforward. Ather convenient means of dealing with nonconstant residual variance is to use the so-called "robust'' variance matrix due to White. The corresponding SEs allows reliable inference when the constant-variance assumption is violated.

This little technial note explains how this can be done: robust SEs be calculated in Stata using the option *vce(robust)* of *regress* with a more accurate version *vce(hc3)* being preferable in small samples. R users can install the libraries *sandwich* and *lmtest* and then use *vcovHC*. The same options and a few other alternatives can be obtained through a command like this: *coeftest(m, vcov. = vcovHC(m, type = 'HC3'))* where *type = "HC3"* specifies the chosen option, here the same as Stata's *vce(hc3)*. The command to match Stata's output for the 95% CI is: *coefci(out, vcov. = vcovHC(out, type = 'HC3'))*. Of course, the object *out* contains the result of the initial *lm* fit. We let you explore these options, for instance, you can reproduce the fits given p. 123 of Vittinghof et al. (2012) using the *hers* dataset and the appropriate sandwich matrix. A textbook example is given in the second lecture so you can also explore that example.

Summary

This weeks key concepts are:

The following are the key takeaway messages from this week:

1. Polynomial regression and RCS are powerful tools to capture nonlinearities
2. RCS and fractional polynomials are part of a suite of flexible techniques that can be used to model complex relationships

3. The bootstrap provides an alternative way to draw inference in situations where the normality assumption is questionable. It can also be used to get SEs for complex statistics.

4. Sandwich formulas exist to cope with hererescedasticity issues (e.g. non constant variance) when no variance-stabilising transformation exists.

8 Regression model building and variable selection

Learning objectives

By the end of this week you should be able to:

1. Understand how model diagnostics are used to compare regression models
2. Build regression models suitable for prediction
3. Build regression models suitable for isolating the effect of a single predictor
4. Build regression models suitable for understanding multiple predictors

Learning activities

This week's learning activities include:

Learning Activity	Learning objectives
Reading	1
Independent exercise	1
Lecture 1	2
Lecture 2	3
Lecture 2	4

Model building

The previous weeks provide the tools to carry out a multiple linear regression in Stata or R and interpret the results. However, the process of choosing exactly which regression model best answers your research question is not always clear. This process involves choosing: which covariates to include, what functional form they should take, and what (if any) transformations may be necessary. Before we illustrate these steps in three different contexts, it is helpful to first introduce several measures of regression model performance to help inform our decisions.

F-tests and Likelihood ratio tests for nested models

The first method of comparing regression models is one we have already been using all course: P-values that test the specific inclusion or exclusion of a variable or groups of variables (e.g. a group of dummy variables associated with a categorical covariate) from the model. This is achieved in linear regression with an F-test which produces an equivalent p-value to the t-test P-value shown in the common regression output for continuous and binary variables. For linear regression, this is also equivalent to a likelihood ratio test P-value. Although this method of comparison is the most intuitive, it is limited in that it can only compare *nested* models - models that differ by the inclusion of one or more variables. So it is not useful for comparing models that differ in other ways. e.g. comparing models with different methods of adjusting for non-linearity (categorisation, cubic-splines, or log-transformation of the covariate). For these comparisons, different model comparison measures need to be employed.

R^2

We are already familiar with *coefficient of determination* R^2 from weeks 1's reading. Recall that this is the proportion of the total variability of the outcome that can be explained by the covariates in the model. Or alternatively 1 minus the proportion of variability remaining unexplained.

$$R^2 = \frac{\text{Model sum of squares}}{\text{Total sum of squares}} = 1 - \frac{\text{Residual sum of squares}}{\text{Total sum of squares}}$$

R^2 therefore provides a natural and intuitive measure of regression performance with higher R^2 values indicating better model performance (as more variability of the outcome can be explained). However, issues arise when R^2 is used to compare models as it will always favour more complex models regardless of whether that increased complexity is justified. This will lead to over paramaterised, or overfitted models. Therefore R^2 is only a useful *comparitive* measures for models of equal complexity e.g. models with the same number of parameters.

Adjusted R^2

The adjusted R^2 attempts to compensate for the overfitting issues associated with the *unadjusted* R^2 by penalising the R^2 calculation by the number of parameters of the model. There are several ways of doing this, and the exact method isn't too important, so the common formula below is shown purely so you can compare the calculation to the regular R^2.

$$\text{adjusted } R^2 = 1 - \frac{\text{Residual sum of squares}/(n-p)}{\text{Total sum of squares}/(n-1)}$$

where n is the number of observations and p is the number of parameters of the model (equal to the number of regression coefficients). So the adjusted R^2 will only increase if the residual error in the more complex model reduces enough to compensate for the penalty of the extra parameter (increasing p).

AIC

The adjusted R^2 is just one of many ways to penalise unnecesary complexity to avoid over fitting. One popular method of adjustment would be the Akaike Information Criterion, commonly known as the AIC. Here, instead of quantifying the model fit through least squares, the *maximum likelihood* is used to compare models - again with a penalty proportional to the number of paramaters. Using a likelihood based approach has the advantage that it can be applied in models not fit through ordinary least squares (such as logistic regression taught in the second half of this course). AIC is calculated as

$$\text{AIC} = 2p - 2\log(\mathcal{L})$$

where p is the number of paramaters and $\mathcal{L}$ is the maximum likelihood value of the model fit. As we are subtracting the maximum likelihood, lower AIC values indicate better models (i.e. higher likelihoods are better). The AIC can be either negative or positive, and so it is important to remember that a "lower" AIC could mean either a smaller positive AIC value, or a "more negative" negative AIC value.

BIC

The Bayesian Information Criterion (BIC) is very similar to the AIC, however instead of penalising by $2p$, it penalises by $2p\log(n)$.

$$\text{BIC} = 2p\log(n) - 2\log(\mathcal{L})$$

This change in penalty between BIC and AIC is important for two reasons. Firstly, the BIC penalty is a stricter penalty than AIC. Secondly, the BIC penalties become progressively stricter as the sample size increases. Both of these generally lead to BIC favouring more simple models than AIC.

Independent exercise

Use the tools above to investigate the ideal number of knots for the week 7 investigation between HDL and BMI.

Lecture 1 - Prediction (more done in week 10)

In this video, we will look at how the tools above can be used to help build regression models suitable for prediction, where the goal is to minimise the predictive error.

Download video here

Lecture 2 - Isolating the effect of a single predictor

In this video, we will look at how the tools above can be used to help build regression models suitable for measuring the effect of an exposure on an outcome, where the goal is to measure this effect without bias due to confounding.

Download video here

Lecture 3 - Understanding multiple predictors

In this video, we will look at how the tools above can be used in exploratory research, where the goal is to identify which covariates are associated with an outcome. It is common in this type of research for potential confounders or predictors of interest to be less well established.

Download video here

Summary

This weeks key concepts are:

1. There are several measures available to help statistically compare regression models - including P-values from t-tests and f-tests, R^2, adjusted R^2, AIC and BIC.
2. How these tools will be applied will be different depending on the context of your research question
3. This course focuses on prediction models, models to understand the effect of a single exposure, and understanding multiple predictors
4. All types of models should also consider contextualised field specific issues and norms.
5. You should not use automatic covariate selection algorithms for use in this course. Rather build models where you are comfortable justifying the inclusion or exclusion of each covariate.

9 Logistic Regression

Learning objectives

By the end of this week you should be able to:

1. Understand the motivation for logistic regression modelling
2. Realise how logistic regression extends linear regression for binary outcomes
3. Understand the link between odd-ratios derived from a frequency table and logistic regression
4. Interpret statistical output for a multiple logistic regression model with or without confounding
5. Understand the likelihood ratio test and how to use it in this setting
6. Interpret a logistic regression model with grouped data/categorical covariates

Learning activities

This week's learning activities include:

Learning Activity	Learning objectives
Lecture 1	1, 2, 3
Readings	1, 2, 3, 4, 5
Lecture 2	4, 5, 6
Pratice/Investigation	3, 4, 6
Discussions	3, 4, 5, 6

Introduction to logistic regression

In simple linear regression, the expectation of a continous variable y is modelled as a linear function of a covariate x i.e.

$$E(y) = \beta_0 + \beta_1 x$$

It's therefore natural to wonder whether a similar idea could not be used for a binary endpoint y taking only 0 or 1 values. Such outcomes are frequent in biostatistics because investigators are typically interested in events affecting a patient's life like complete response at 3 months, resolution of a targeted tumour at 6 months or 28 day mortality. Using directly the linear model would lead to wrong conclusions because the response is clearly not normally distributed and linearity on this case is very unlikely. Here $E(y|x) = P(y = 1|x) = p(x)$ because y can only be 0 or 1 with 1 representing the occurence of the event and 0 its absence. What we need is a proper way to model this probability p and inevitably the question of an appropriate scale arises. It turns out that a convenient way to model p is to use the logistic function g where $g(p) = \log(p/(1-p))$. Then the natural counterpart to simple linear regression is the simple logistic regression model expressed as:

$$\log \frac{p(x)}{(1-p(x))} = \beta_0 + \beta_1 x$$

yielding that linearity arises on the logistic scale. Read p. 139 - 143 of the book for more motivation on this choice of the scale. Assuming that the model is correct for n independent observations from a sample we can write the log-likehood of this sample and derive the maximum likehood estimates (MLE) of the parameters β_0 and β_1). As before, we will note them $\widehat{\beta}_0$ and $\widehat{\beta}_1$ (or b_0 and b_1). The MLE in logistic regression and its corresponding standard error are routinely provided in all statistical packages.

Lecture 1 in R

Download video here

Lecture 1 in Stata

Download video here

Interpretation of regression coefficients

To illustrate the use of logistic regression, we consider data from the Western Collaborative Group Study (WCGS), a large epidemiological study designed to investigate the relationship between coronary heart disease (CHD) and various potential predictors including behavioural patterns. The dataset is called *wcgs.csv* and the outcome is *chd69* (0/1) with 1 indicating the occurence of a coronary event over the course of the study. An association of interest to the original investigators was the relationship between CHD risk and the presence/absence of

corneal arcus senilis (*arcus*) among participants upon entry into the study. Arcus senilis is is a whitish annular deposit around the iris that occurs in a small percentage of older adults and is a legitimate predictor since it is thought to be related to serum cholesterol level. An exploratory analysis indicates that patients arcus senilis are more likely to develop CHD compared with the others (11% vs 7%) and a standard Chi2 test returns a significant result (p=0.000). A simple logistic regression analysis of the same data is given below.

R code and output

```
wcgs <- read.csv("wcgs.csv")
wcgs<-data.frame(wcgs)
model0<-glm(chd69 ~ arcus,  family=binomial, data=wcgs)
summary(model0)
##
## Call:
## glm(formula = chd69 ~ arcus, family = binomial, data = wcgs)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.5991     0.0838 -31.016  < 2e-16 ***
## arcus         0.4918     0.1342   3.664 0.000248 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1771.2  on 3151  degrees of freedom
## Residual deviance: 1758.2  on 3150  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: 1762.2
##
## Number of Fisher Scoring iterations: 5
exp(model0$coefficients)[2]
##   arcus
## 1.63528
exp(confint(model0))[2,]
##    2.5 %   97.5 %
## 1.254294 2.124062
```

Stata code and output

```
use wcgs.dta
** fitted model with coefficients
logistic chd69 arcus, coef
** fitted model with OR (default in Stata)
logistic chd69 arcus
** arcus is coded 0/1 so i.arcus is not needed
## Logistic regression                                      Number of obs =  3,152
##                                                          LR chi2(1)    =  12.98
##                                                          Prob > chi2   = 0.0003
## Log likelihood = -879.10783                              Pseudo R2     = 0.0073
##
## ------------------------------------------------------------------------------
##        chd69 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##        arcus |   .4918144   .1342299     3.66   0.000     .2287286    .7549002
##        _cons |  -2.599052   .0837965   -31.02   0.000     -2.76329   -2.434814
## ------------------------------------------------------------------------------
##
##
## Logistic regression                                      Number of obs =  3,152
##                                                          LR chi2(1)    =  12.98
##                                                          Prob > chi2   = 0.0003
## Log likelihood = -879.10783                              Pseudo R2     = 0.0073
##
## ------------------------------------------------------------------------------
##        chd69 | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##        arcus |   1.635281   .2195036     3.66   0.000     1.257001    2.127399
##        _cons |    .074344   .0062298   -31.02   0.000     .0630839    .0876141
## ------------------------------------------------------------------------------
## Note: _cons estimates baseline odds.
```

The coefficient label (Intercept) or _cont in Stata is the intercept ($\hat{\beta}_0$) in the model and the coefficient for arcus is $\hat{\beta}_1$ representing the effect of this condition on CHD in the analysis. To interpret the coefficient, we go back to the general formula above and notice that linearity is specified on the log-odd scale. Applying this to our example reveals that the log-odds of CHD is $\hat{\beta}_0 = -2.599$ in patients *without* arcus and $\hat{\beta}_0 + \hat{\beta}_1 = -2.107$ in patients *with* arcus. The difference in CHD risk due to arcus is then $\hat{\beta}_1 = .492$ on the *log-odds scale*, i.e. the corresponding odds-ratio (OR) is exp(.492) = 1.64. By the same token we can compute the 95% CI for this odds-ratio leading to (1.25 ; 2.12) and the effect is statistically significant ($p < 0.000$).

Exercise

You can certainly remember that we can derive an OR from a 2x2 table. Using the same WCGS data, carry out the following analysis:

a) Reproduce the exploratory analysis with the χ^2-test

b) Compute the OR and check that is exactly the same result as the one obtained via simple logistic regression

c) A large sample formula for the standard error of the log-OR estimate in a 2x2 table is given by: $SE(log(\hat{OR})) = \sqrt{1/a + 1/b + 1/c + 1/d)}$ where a, b, c and d are the frequencies in the 2x2 table. Compute the 95% CI for the estimate you have just computed. How does it compare with the 95% obtained from logistic regression. Hint: start by computing a 95% CI for the log-OR.

9.0.0.0.1 * [@vittinghoff2012] Chapter 5. Logistic regression (Section 5.1.1) but read also all pages 139 - 146 for the introduction).

This reading explains how the same logic can be used to interpret the coefficient of a continuous predictor in a logistic regression model. The example is age that is thought to be associated with CHD as well.

Multiple logistic regression

Like in linear regression the simple logistic regression model can be extendend to include multiple predictors $x_1, \dots, x_p$. We can follow the same logic and model $E(Y|x_1, \dots, x_p) = P(Y = 1|x_1, \dots, x_p) = p(x_1, \dots, x_p)$ on the logistic scale yielding:

$$\log \frac{p(x_1, \dots, x_p)}{(1 - p(x_1, \dots, x_p))} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

. This can be seen as an extension of multiple linear regression using similar arguments to the simple case. the interpretation of the coefficients in terms of log log-OR also holds with the difference that the analysis is now adjusted for the other covariates.

9.0.0.0.1 * [@vittinghoff2012] Chapter 5. Logistic regression (Section 5.2) p. 150-54).

This reading shows how potential predictors like age, cholesterol, sytolic blood pressure (SBP), body max index (BMI) and smoking can be included in a logistic regression model for CHD. It also illustrates the interpretation of the coefficients and how rescaling of the covariates affects the coefficients but leads to a similar interpretation.

Lecture 2 in R

Download video here

Lecture 2 in Stata

Download video here

Likelihood ratio tests

The likelihood ratio tests (LRT) has been introduced in PSI, it has an important role to play in (logistic) regression analysis since it allows us to compare two nested models. The word *nested* means that a model is bigger than the other in the sense that it contains all the covariates of the smaller model.

They are particular useful when investigating the contribution of more than one covariate or predictors with multiple levels. Say we want to assess the impact of self-reported behaviour called *behpat* representing different behaviours (type (A or B)) subdivided into two further levels leading to 4 categories A_1, A_2, B_3 and B_4 (coded 1-4 respectively). The null hypothesis corresponding to this question is H_0: $\beta_6 = \beta_7 = \beta_8 = 0$ since the coefficients corresponding to all *behpat* except the reference A_1 are $beta_6$, β_7, and β_8. The LRT can be used to test thus (multiple) null hypothesis H_0 by computing the difference in twice the log-likelihood (logLL) between the two models. We often used the term *full model* for the larger model (all previous covariates and *behpat*) and *reduced model* for the smaller one (only the previous covariates). The term *reduced* stems from the restrictions to the full model by imposing that all the coefficients for the *behpat* categories except the reference are zero. We know that, under the absence of effect of *behpat*, the LRT statistic follow a Chi2 distribution with 3 degrees of freedom (i.e. the number of parameters tested in the model, that is the number of categories minus 1).

$$LRT = 2 \times logLL(full) - 2 \times logLL(reduced) = 2 \times (-784.81 - (-807.19) = 24.76$$

The corresponding p-value is derived from a Chi2 distribution with 3 degrees of freedom yielding $p = 1.7e - 05 < 0.0001$. A neat effect of the self-reported behavious is observed overall.

R code and output

```
myvars <- c("id","chd69", "age", "bmi", "chol", "sbp", "smoke", "dibpat", "behpat_type")
wcgs1 <- wcgs[myvars]
wcgs1=wcgs1[wcgs1$chol <645,]
wcgs1cc=na.omit(wcgs1)
# remove missing values - complete case (cc) analysis
```

```r
wcgs1cc<-na.omit(wcgs1)
# rescale variables
wcgs1cc$age_10<-wcgs1cc$age/10
wcgs1cc$bmi_10<-wcgs1cc$bmi/10
wcgs1cc$chol_50<-wcgs1cc$chol/50
wcgs1cc$sbp_50<-wcgs1cc$sbp/50
# define factor variable
wcgs1cc$behpat<-factor(wcgs1cc$behpat_type)
reduced<-glm(chd69 ~ age_10+chol_50+bmi_10+sbp_50+smoke, family=binomial, data=wcgs1cc)
summary(reduced)
##
## Call:
## glm(formula = chd69 ~ age_10 + chol_50 + bmi_10 + sbp_50 + smoke,
##     family = binomial, data = wcgs1cc)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -12.31099    0.97726 -12.598  < 2e-16 ***
## age_10        0.64448    0.11907   5.412 6.22e-08 ***
## chol_50       0.53706    0.07586   7.079 1.45e-12 ***
## bmi_10        0.57436    0.26355   2.179   0.0293 *
## sbp_50        0.96469    0.20454   4.716 2.40e-06 ***
## smoke         0.63448    0.14018   4.526 6.01e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1774.2  on 3140  degrees of freedom
## Residual deviance: 1614.4  on 3135  degrees of freedom
## AIC: 1626.4
##
## Number of Fisher Scoring iterations: 6
full<-glm(chd69 ~ age_10+chol_50+bmi_10+sbp_50+smoke+factor(behpat), family=binomial, data
summary(full)
##
## Call:
## glm(formula = chd69 ~ age_10 + chol_50 + bmi_10 + sbp_50 + smoke +
##     factor(behpat), family = binomial, data = wcgs1cc)
##
## Coefficients:
```

```
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -11.64889    1.01421 -11.486  < 2e-16 ***
## age_10             0.60636    0.11990   5.057 4.25e-07 ***
## chol_50            0.53304    0.07637   6.980 2.96e-12 ***
## bmi_10             0.55355    0.26563   2.084  0.03717 *
## sbp_50             0.90158    0.20647   4.367 1.26e-05 ***
## smoke              0.60468    0.14110   4.285 1.82e-05 ***
## factor(behpat)A2   0.06603    0.22123   0.298  0.76535
## factor(behpat)B3  -0.66522    0.24226  -2.746  0.00603 **
## factor(behpat)B4  -0.55849    0.31921  -1.750  0.08019 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1774.2  on 3140  degrees of freedom
## Residual deviance: 1589.6  on 3132  degrees of freedom
## AIC: 1607.6
##
## Number of Fisher Scoring iterations: 6
LRT=2*(logLik(full)-logLik(reduced))
# LRT and p-value
LRT
## 'log Lik.' 24.76497 (df=9)
pval=1-pchisq(LRT,3)
pval
## 'log Lik.' 1.728999e-05 (df=9)
# another way using anova (possibly quicker)
out<-anova(reduced,full)
out
## Analysis of Deviance Table
##
## Model 1: chd69 ~ age_10 + chol_50 + bmi_10 + sbp_50 + smoke
## Model 2: chd69 ~ age_10 + chol_50 + bmi_10 + sbp_50 + smoke + factor(behpat)
##   Resid. Df Resid. Dev Df Deviance
## 1      3135     1614.4
## 2      3132     1589.6  3   24.765
1-pchisq(as.vector(out$Deviance)[2],3)
## [1] 1.728999e-05
```

Stata code and output

```
use wcgs.dta
gen age_10=age/10
gen chol_50=chol/50
gen bmi_10=bmi/10
gen sbp_50=sbp/50
** fit reduced and full model without outlier (chol=645)
logistic chd69 age_10 chol_50 bmi_10 sbp_50 i.smoke if chol<645
estimates store mod1
logistic chd69 age_10 chol_50 sbp_50 bmi_10 i.smoke i.behpat if chol<645
lrtest mod1
** the command estimates is needed to store the 1st fit and computes the LRT


**
** NB: if you use the csv dataset you have to use behpat_type and recode
**     i. does not work with string, Then use the same commands
** gen behpat=1 if behpat_type=="A1"
** replace behpat=2 if behpat_type=="A2"
** replace behpat=3 if behpat_type=="B3"
** replace behpat=4 if behpat_type=="B4"
** drop behpat_type
**

## (12 missing values generated)
##
##
##
##
## Logistic regression                                     Number of obs =  3,141
##                                                         LR chi2(5)    = 159.80
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -807.19249                             Pseudo R2     = 0.0901
##
## ------------------------------------------------------------------------------
##        chd69 | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##       age_10 |   1.904989   .2268333     5.41   0.000     1.508471    2.405735
##      chol_50 |   1.710974   .1297977     7.08   0.000     1.474584    1.985259
##       bmi_10 |   1.775995   .4680613     2.18   0.029     1.059518    2.976973
##       sbp_50 |   2.623972   .5367142     4.72   0.000     1.757326    3.918016
##      1.smoke |   1.886037   .2643914     4.53   0.000     1.432933    2.482417
```

```
##        _cons |   4.50e-06   4.40e-06   -12.60   0.000     6.63e-07    .0000306
## ------------------------------------------------------------------------------
## Note: _cons estimates baseline odds.
##
##
##
## Logistic regression                                     Number of obs =  3,141
##                                                         LR chi2(8)    = 184.57
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -794.81                                Pseudo R2     = 0.1040
##
## ------------------------------------------------------------------------------
##        chd69 | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##       age_10 |    1.83375   .2198681     5.06   0.000     1.449707    2.319529
##      chol_50 |   1.704097   .1301391     6.98   0.000     1.467201    1.979243
##       sbp_50 |   2.463504   .5086518     4.37   0.000     1.643621    3.692369
##       bmi_10 |   1.739415   .4620341     2.08   0.037     1.033479    2.927551
##      1.smoke |   1.830672   .2583097     4.29   0.000      1.38837    2.413882
##              |
##       behpat |
##           2  |   1.068257   .2363271     0.30   0.765     .6924157    1.648103
##           3  |   .5141593   .1245593    -2.75   0.006     .3198064    .8266243
##           4  |    .572071   .1826117    -1.75   0.080     .3060107    1.069457
##              |
##        _cons |   8.73e-06   8.85e-06   -11.49   0.000     1.20e-06    .0000637
## ------------------------------------------------------------------------------
## Note: _cons estimates baseline odds.
##
##
## Likelihood-ratio test
## Assumption: mod1 nested within .
##
##  LR chi2(3) =  24.76
## Prob > chi2 = 0.0000
```

9.0.0.0.1 * [@vittinghoff2012] Chapter 5. Logistic regression (Section 5.2.1) p. 154-56).

This reading gives you the two fitted models, the calculation of the LRT in Stata and some explanation of why this test is important.

We woud like to end this paragraph by stressing out that it is important to have no missing

values in the dat, at least in the covariates involved in the computation of the LRT. Many statistical packages will always provide log-likelihood values irrespective of the presence of missing data. The calculation of the LRT will be wrong since you are basically computing the log-likelihood from *different* samples. They are different because the packages delete the missing data before fitting the model. As a result, you may end up having a different number of observations in the full and reduced models, which makes the whole calculation meanningless.

Investigation - group variables

In this activity, you are asked to investigate the impact of a risk factor with multiple levels that are possibly ordered. Age has been divided in 5 age group categories given by *agec*. Draw a 2×5 table investigating the association between *chd69* and *agec* and compute the corresponding ORs. Can you get similar results using logistic regression, how? Can you test the global effect of *agec* on *chd69*. How would you go about it? [Hint: it may help to look at p. 146-148] This analysis is unadjusted, what do you suggest we do next?

Confounding

Confounding can occur in logistic regression for the same reasons as in linear regression. The same criteria can be applied here, i.e. a variable c is a confounder of a relationship $x \rightarrow y$ where y is a binary endpoint if 1) c associated with y; 2) c associated with c; c is not on the causal pathway $x \rightarrow y$. Several predictors can also act as potential confounders in practice. To briefly illustrate the concept in this setting, consider the association between CHD and a behaviour pattern considered previously, the only difference being that for simplicity, we only consider type A or B. The resulting binary covariate is *dibpat* coded 0/1 with 0 for B or 1 for A. A simple logistic regression analysis returns a "crude" OR=2.36, 95%CI=(1.80 ; 3.12) after deletion of an outlier (a case with a cholesterol reading of 645 mg/dl = 16.68 mmol/l at the beginning of the study). The word crude (or raw) OR is used to indicate that the analyis is unadjusted. Potential confounders include age, BMI, cholesterol, SBP and smoking but, to facilate the reading and comparison with the textbook, we use rescaled versions of these predictors. For example, the authors were interested to produce OR for ten-year increase in age, so they use *age_10=age/10* with the subscript indicating the scaling factor. It is easy to see that all the predictors are independently associated with *chd69* - criterion 1). They are also linked to *dibpat* - criterion 2) - with the exception of BMI and we let you check this as an exercise. Criterion 3) is always harder to assess and come mainly from the knowledge of the field; in this case, it is unlikely to be met for either of these factors. All predictors but BMI can confound the association of interest so it is legitimate to wonder what impact they have collectively. An adjusted analysis is then conducted yielding:

R code and output

```
#   chd69, smoke and dibpat are assumed to be coded 0/1
# wcgs <- read.csv("wcgs.csv")
myvars <- c("id","chd69", "age", "bmi", "chol", "sbp", "smoke", "dibpat")
wcgs1 <- wcgs[myvars]
wcgs1=wcgs1[wcgs1$chol <645,]
wcgs1cc=na.omit(wcgs1)
# remove missing values - complete case (cc) analysis
wcgs1cc<-na.omit(wcgs1)
# rescale variables
wcgs1cc$age_10<-wcgs1cc$age/10
wcgs1cc$bmi_10<-wcgs1cc$bmi/10
wcgs1cc$chol_50<-wcgs1cc$chol/50
wcgs1cc$sbp_50<-wcgs1cc$sbp/50
# adjusted
out1<-glm(chd69 ~ age_10+chol_50+bmi_10+sbp_50+smoke + dibpat, family=binomial, data=wcgs1
summary(out1)
##
## Call:
## glm(formula = chd69 ~ age_10 + chol_50 + bmi_10 + sbp_50 + smoke +
##     dibpat, family = binomial, data = wcgs1cc)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -12.27086    0.98211 -12.494  < 2e-16 ***
## age_10        0.60445    0.11969   5.050 4.41e-07 ***
## chol_50       0.53204    0.07634   6.970 3.18e-12 ***
## bmi_10        0.54948    0.26531   2.071   0.0384 *
## sbp_50        0.90338    0.20602   4.385 1.16e-05 ***
## smoke         0.60386    0.14109   4.280 1.87e-05 ***
## dibpat        0.69657    0.14437   4.825 1.40e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1774.2  on 3140  degrees of freedom
## Residual deviance: 1589.9  on 3134  degrees of freedom
## AIC: 1603.9
##
## Number of Fisher Scoring iterations: 6
# ORs and 95% CIs
```

```r
## adjusted
exp(out1$coefficients)[2:7]
##   age_10  chol_50   bmi_10   sbp_50    smoke   dibpat
## 1.830251 1.702406 1.732348 2.467919 1.829162 2.006855
exp(confint(out1))[2:7,]
##            2.5 %   97.5 %
## age_10  1.447496 2.314909
## chol_50 1.465976 1.977767
## bmi_10  1.027046 2.906856
## sbp_50  1.640586 3.682099
## smoke   1.390013 2.418030
## dibpat  1.517449 2.674349
# unadjusted
out1<-glm(chd69 ~  dibpat, family=binomial, data=wcgs1cc)
exp(out1$coefficients)[2]
##   dibpat
## 2.356834
exp(confint(out1))[2,]
##    2.5 %   97.5 %
## 1.796912 3.117231
```

Stata code and output

```
use wcgs.dta
gen age_10=age/10
gen chol_50=chol/50
gen bmi_10=bmi/10
gen sbp_50=sbp/50
** delete missing and outliers permanently
drop if missing(chd69) | missing(bmi) | missing(age) | missing(sbp) | missing(smoke) | mis
drop if chol ==645
**
** turns out to be equivalent to: drop if chol >=645
** This is because the missing are in cholesterol and
** missing values are eliminated by the condition
**
** adjusted ORs
logistic chd69 bmi age sbp smoke chol dibpat
**  smoke and dibpat are coded 0/1 so it is equivalent to using i.smoke and i.dibpat
** unadjusted ORs
logistic chd69 dibpat
```

```
## (12 missing values generated)
##
##
##
## (12 observations deleted)
##
## (1 observation deleted)
##
##
## Logistic regression                                     Number of obs =  3,141
##                                                         LR chi2(6)    = 184.34
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -794.92603                             Pseudo R2     = 0.1039
##
## ------------------------------------------------------------------------------
##        chd69 | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##          bmi |   1.056485   .0280297     2.07   0.038     1.002952    1.112876
##          age |    1.06231   .0127148     5.05   0.000     1.037679    1.087525
##          sbp |   1.018232   .0041955     4.38   0.000     1.010042    1.026488
##        smoke |   1.829162   .2580698     4.28   0.000     1.387265    2.411822
##         chol |   1.010698   .0015431     6.97   0.000     1.007678    1.013727
##       dibpat |   2.006855   .2897341     4.82   0.000     1.512259    2.663212
##        _cons |   4.69e-06   4.60e-06   -12.49   0.000     6.84e-07    .0000321
## ------------------------------------------------------------------------------
## Note: _cons estimates baseline odds.
##
##
## Logistic regression                                     Number of obs =  3,141
##                                                         LR chi2(1)    =  40.14
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -867.02525                             Pseudo R2     = 0.0226
##
## ------------------------------------------------------------------------------
##        chd69 | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##       dibpat |   2.356834   .3307581     6.11   0.000     1.790076    3.103035
##        _cons |   .0534145    .006168   -25.37   0.000     .0425958    .0669809
## ------------------------------------------------------------------------------
## Note: _cons estimates baseline odds.
```

The adjusted OR for *dibpat* (type A) is smaller OR=2.01 95%CI =(1.52 ; 2.67) than the un-

adjusted OR=2.36 95%CI=(1.80 ; 3.12), so some degree of confounding indeed occurs. This example illustrates what researchers typically do in practice, i.e. they compare the unadjusted analysis with the adjusted analysis with all potential confounders added to the model, irrespective of whether they are indeed confounders or not, and sometimes without looking at significance. This is somehow a looser interpretation of the criteria listed above but can also be understood in a global context where, for instance, a factor like BMI may not appear to confound the association between *chd69* and *dibpat* in this dataset but might have been in other similar epidemiological studies.

You can also read Section 5.2.2. p. 156-58 of the book for a slightly more elaborate discussion of this example - optional reading.

Summary

The following are the key takeaway messages from this week:

1. Logistic regression is the natural way to extend the notion of odds-ratio by modelling the log-odds of an event as linear combination of covariates.
2. Binary logistic regression models the expected response (i.e. the probability of an event occurence) on the log-odds scale and inference relies on maximum likelihood theory; it is therefore an extension of linear regression for binary outcomes.
3. Likelihood ratio tests are very useful in this context. They can allow us to test a null hypothesis involving several regression parameters.
4. General concepts like adjustment and interaction are similar to the ones described for linear regression.

10 Logistic Regression

Learning objectives

By the end of this week you should be able to:

1. Interpret statistical output for a multiple logistic regression model with interactions
2. Understand how predicted values and residuals can be extended to this model
3. Learn about prediction and residuals in this setting
4. Discover how to identify outliers and influential observations
5. Understand how to use these tools to assess the model fit

Learning activities

This week's learning activities include:

Learning Activity	Learning objectives
Lecture 1	1, 2, 3
Reading	1,
Lecture 2	2, 3, 4, 5
Practice/Investigation	1, 3
Discussion	all

Lecture 1 in R

Download video here

Lecture 1 in Stata

Download video here

10.1 Interactions {.unnumbered}}

The interaction between two predictors or effect modification is quite similar to what we saw in linear regression. We will revisit this concept since its interpretation is typically done in terms of OR (whereas the intearction term is typically added on the log-odds scale). This deserves explanation. The general form of an interaction model with two covariates x_1 and x_2 is: $logit(p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 \times x_2$ where $logit(p)$ is shortened version of $\log(p/(1-p))$ and, for simplicity, we have simplified the notation since $p = p(y = 1|x_1, x_2, x_1 \times x_2)$. Also x_1 and x_2 can be binary or continuous with the convention that a binary indicator is always coded 0/1.

10.1.1 Interaction between two binary predictors {.unnumbered}}

We again consider the WcGS study and consider the potential interaction between *arcus* and a binary indicator for patients aged over 50 called *bage_50*. The specification of the logistic model in Stata or R follows the general syntax used with the linear model. You can either define "by hand" the interaction term and add it to the model with the the two covariates *arcus* and *bage_50* or let the software do the job for you. We will use the second approach here but it critical to be sure about the coding (0/1) or tell Stata or R to create the right indicators for you. The Stata syntax will then be: *logistic chd69 i.arcus##i.bage_50, coef* and R's: *glm(chd69 ~ factor(arcus)*factor(bage_50), family=binomial, data=wcgs).* The command *factor()* is not necessary when both covariates are coded 0/1. Note that in the Stata command we used the option *coef* to avoid reporting the ORs. The following results are obtained:

R code and output

```
wcgs <- read.csv("wcgs.csv")
wcgs<-data.frame(wcgs)
wcgs$bage_50<-as.numeric(wcgs$age>=50)
out1<-glm(chd69 ~ arcus*bage_50, family=binomial, data=wcgs)
summary(out1)
##
## Call:
## glm(formula = chd69 ~ arcus * bage_50, family = binomial, data = wcgs)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.8829     0.1089 -26.467  < 2e-16 ***
## arcus           0.6480     0.1789   3.623 0.000292 ***
## bage_50         0.8933     0.1721   5.190 2.11e-07 ***
## arcus:bage_50  -0.5921     0.2722  -2.175 0.029640 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1771.2  on 3151  degrees of freedom
## Residual deviance: 1730.9  on 3148  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: 1738.9
##
## Number of Fisher Scoring iterations: 5
```

Stata code and output

```
use wcgs.dta
gen bage50=(age>=50)
logistic chd69 arcus##bage50, coef
## Logistic regression                                     Number of obs =  3,152
##                                                         LR chi2(3)    =  40.33
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -865.43251                             Pseudo R2     = 0.0228
##
## ------------------------------------------------------------------------------
##        chd69 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##      1.arcus |   .6479628   .1788637     3.62   0.000     .2973964    .9985293
##     1.bage50 |   .8932677   .1721239     5.19   0.000     .5559111    1.230624
##              |
## arcus#bage50 |
##          1 1 |  -.5920552   .2722269    -2.17   0.030     -1.12561   -.0585002
##              |
##        _cons |  -2.882853   .1089261   -26.47   0.000    -3.096344   -2.669362
## ------------------------------------------------------------------------------
```

The interaction term is significant ($p = 0.03$) which means that we cannot igore the age effect when considering the association of arcus with CHD. Let's use the notation introduced above, $x_1 = arcus$ and $x_2 = bage\_50$ and consider a young patient (less than 50) without arcus. The log-odds of CHD occurence is: $\hat{\beta}_0 = -2.88$ up to rounding. Compare with someone in the same age category with arcus, the log-odds of CHD occurence is: $\hat{\beta}_0 + \hat{\beta}_1 = -2.23$, therefore the OR for arcus in this age group is: $\exp(\hat{\beta}_0 + \hat{\beta}_1)/\exp(\hat{\beta}_0) = \exp(\hat{\beta}_1) = 1.91$, 95%CI=(1.34 ; 2.70). By the same token, the log-odds of CHD occurence for a patient aged 50+ without arcus is: $\hat{\beta}_0 + \hat{\beta}_2 = -2.88 + .89 = -1.99$ and changes to $\hat{\beta}_0 + \hat{\beta}_1 + \hat{\beta}_2 + \hat{\beta}_3 = -1.93$ for patients

with arcus, therefore the OR for arcus in patients aged 50+ is $\exp(\hat{\beta}_1 + \hat{\beta}_3) = 1.06$ by taking the ratio of the corresponding exponentiated terms. To get a 95% CI for this OR we need to use *lincom* or the corresponding R command yielding OR=1.06, 95%CI=(0.71 ; 1.58).

R code and output

```
library(multcomp)
## Loading required package: mvtnorm
## Loading required package: survival
## Loading required package: TH.data
## Loading required package: MASS
##
## Attaching package: 'TH.data'
## The following object is masked from 'package:MASS':
##
##     geyser
lincom <- glht(out1,linfct=c("arcus+arcus:bage_50 =0"))
out2<-summary(lincom)$test
OR<-exp(out2$coefficients)
lower<-exp(out2$coefficients -1.96*out2$sigma)
upper<-exp(out2$coefficients +1.96*out2$sigma)
# estimate + 95% CI for the OR
lincom
##
##   General Linear Hypotheses
##
## Linear Hypotheses:
##                             Estimate
## arcus + arcus:bage_50 == 0  0.05591
cbind(OR,lower,upper)
##                            OR     lower    upper
## arcus + arcus:bage_50 1.0575 0.7072835 1.581129
```

Stata code and output

```
use wcgs.dta
gen bage50=(age>=50)
** OR for arcus in patients aged less than 50, direct from output
logistic chd69 arcus##bage50
**  OR for arcus in patients aged >= 50, use lincom
lincom 1.arcus + 1.arcus#1.bage50, or
## Logistic regression                                    Number of obs =  3,152
```

```
##                                                 LR chi2(3)    =  40.33
##                                                 Prob > chi2   = 0.0000
## Log likelihood = -865.43251                     Pseudo R2     = 0.0228
##
## ------------------------------------------------------------------------------
##        chd69 | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##      1.arcus |   1.911643   .3419235     3.62   0.000     1.346349    2.714287
##     1.bage50 |     2.4431   .4205159     5.19   0.000     1.743529    3.423366
##              |
## arcus#bage50 |
##          1 1 |   .5531892    .150593    -2.17   0.030     .3244544     .943178
##              |
##        _cons |   .0559748   .0060971   -26.47   0.000     .0452142    .0692964
## ------------------------------------------------------------------------------
## Note: _cons estimates baseline odds.
##
##
##  ( 1)  [chd69]1.arcus + [chd69]1.arcus#1.bage50 = 0
##
## ------------------------------------------------------------------------------
##        chd69 | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##          (1) |     1.0575   .2170202     0.27   0.785     .7072887    1.581117
## ------------------------------------------------------------------------------
```

In other words, the OR for arcus is $\exp(\hat{\beta}_1) = 1.91$, 95% CI=(1.34 ; 2.70) in patients aged less than 50 and $\exp(\hat{\beta}_1 + \hat{\beta}_3) = 1.06$, 95% CI= (0.71 ; 1.58) in patients aged 50+. We clearly see here the effect modification at play, the OR in patients aged less than 50 is *multiplied* by $\exp(\hat{\beta}3)$ to provide the OR in patients aged 50+. The additive interaction term on the log-odds scale translates into a multiplicative factor for the OR (due to the well known property of the exponential function).

10.1.2 Interaction between a binary indicator and a continuous predictor {.unnumbered}}

Interactions between a continuous variable and a binary predictor can be handled in a similar way. The is the purpose of next activity.

Investigation:

1) start by reading the compulsory reading:

10.1.2.0.1 * [@vittinghoff2012] Chapter 5. Logistic regression (Section 5.2.4. p 163-165)

This reading explains how to introduce a possible interaction the between *age* seen this time as a continuous variable and *arcus* (coded 0/1).

2) try to reproduce the output

3) can you give the association between *chd69* and *age* in patients without arcus? Provide the OR, its 95% CI and give an interpretation of the association.

4) give the association between *chd69* and *age* in patients with arcus. Provide the OR, its 95% C and give an interpretation of the association. Does it make sense to add an interaction in this model?

5) Can we interpret the coefficient of *arcus* alone? How can we get a more meaningful coefficient for *arcus* (either on the log-odds scale or as an OR)?

10.2 Prediction {.unnumbered}}

Just as we did for the linear model we can create predictions for all patients in the dataset and beyond. What does it mean in this context? Let rewrite the formula defining the logistic model for a given sample of $i = 1, \dots, n$ observations. Again we simplify the notation and note p_i the probability of observing an event (e.g. cHD in our example) for patient i given all their characteristics $x_{1i}, \dots, x_{pi}$. It is convenient to create the vector of covariates for this individual x_i = (1,x_{1i},..., x_{pi})^T$, the leading one being added for the intercept. The logistic model then stipulates that:

$$log(p_i/(1-p_i)) = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi} = x_i^T \beta,$$

where $\beta = (\beta_0, \beta_1, \dots beta_p)^T$ is the vector or parameter. It's very similar to the formula in linear for the mean response in the linear model (up to the logistic transformation called link). It's possible to extract p_i from this equation by using the inverse transformation yielding $p_i = \exp(x_i^T\beta)/(1 + \exp(x_i^T\beta))$. This expression represents the probability of the patient experiencing the event and is between 0 and 1 [why?]. Now, it's easy to get the predicted probability or prediction noted $\hat{p}_i$ for patient i by plugging in the MLE $\hat{\beta}$ for β in the formula, i.e.:

$$\hat{p}_i = \frac{\exp(x_i^T\hat{\beta})}{1 + \exp(x_i^T\hat{\beta})}$$

By the same token we can compute the probability of a new patient experiencing an event by using a different set of covariates x_{new}. All statistical packages provide predicted values for all patients using a command predict or equivalent. AS an example we can compute the predicted

probabilities of CHD occurence for all patients in the dataset. Also, the same command(s) can be used for a new patient with *age=40*, *bmi=25*, *chol=400*, *sbp=130*, *smoke=0*, *dibpat=0*. To avoid issue with the rescaling/centring we will refit the model with the original variables. The Stata and R code can be found here and give the same results, i.e. $\hat{p} = 0.13$, 95%CI=(0.079 ; 0.216).

R code and output

```
myvars <- c("id","chd69", "age", "bmi", "chol", "sbp", "smoke", "dibpat")
wcgs1 <- wcgs[myvars]
wcgs1=wcgs1[wcgs1$chol <645,]
wcgs1cc=na.omit(wcgs1) # 3141 x 11
model1<-glm(chd69 ~ age + chol + sbp + bmi + smoke + dibpat, family=binomial, data=wcgs1cc
summary(model1)
##
## Call:
## glm(formula = chd69 ~ age + chol + sbp + bmi + smoke + dibpat,
##     family = binomial, data = wcgs1cc)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -12.270864   0.982111 -12.494  < 2e-16 ***
## age           0.060445   0.011969   5.050 4.41e-07 ***
## chol          0.010641   0.001527   6.970 3.18e-12 ***
## sbp           0.018068   0.004120   4.385 1.16e-05 ***
## bmi           0.054948   0.026531   2.071   0.0384 *
## smoke         0.603858   0.141086   4.280 1.87e-05 ***
## dibpat        0.696569   0.144372   4.825 1.40e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1774.2  on 3140  degrees of freedom
## Residual deviance: 1589.9  on 3134  degrees of freedom
## AIC: 1603.9
##
## Number of Fisher Scoring iterations: 6
#wcgs1cc$pred.prob <- fitted(model1)
pred<-predict(model1,type = "response",se.fit = TRUE)
pred<-predict(model1,type = "response")
```

```r
#  prediction + 95% CI for a patient age = 40, bmi=25, chol =400, sbp=130, smoke=0, dibpat
new <- data.frame(age = 40, bmi=25, chol =400, sbp=130, smoke=0, dibpat=0)
out <- predict(model1, new, type="link",se.fit=TRUE)
mean<-out$fit
SE<-out$se.fit
# 95% CI for the linear predictor (link option)
CI=c(mean-1.96*SE,mean+1.96*SE)
# 95% CI for the CHD probability by transformation Via the reciprocal of logit = expit
f.expit<-function(u){exp(u)/(1+exp(u))}
f.expit(c(mean,CI))
##          1          1          1
## 0.13305183 0.07875639 0.21600251
```

Stata code and output

```
use wcgs.dta
drop if missing(chd69) | missing(bmi) | missing(age) | missing(sbp) | missing(smoke) | mis
drop if chol ==645

** proba CHD as a function of age, bmi, chol, sbp, smoke, dibpat
** only for patients in the dataset

logistic chd69 age chol sbp bmi smoke dibpat, coef
predict proba, pr

** prediction for a new patient: age = 40, bmi=25, chol =400, sbp=130, smoke=0, dibpat=0
adjust age = 40 bmi=25 chol =400 sbp=130 smoke=0 dibpat=0, ci pr

** by hand transforming the linear predictor and its 95% CI

adjust age = 40 bmi=25 chol =400 sbp=130 smoke=0 dibpat=0, ci
disp exp( -1.87424)/(1+exp(-1.87424))
disp exp(-2.45935)/(1+exp(-2.45935))
disp exp(-1.28913)/(1+exp(-1.28913))
## (12 observations deleted)
##
## (1 observation deleted)
##
##
## Logistic regression                                     Number of obs =  3,141
##                                                         LR chi2(6)    = 184.34
##                                                         Prob > chi2   = 0.0000
```

```
## Log likelihood = -794.92603                         Pseudo R2     = 0.1039
##
## ------------------------------------------------------------------------------
##        chd69 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##          age |   .0604453    .011969     5.05   0.000     .0369866    .0839041
##         chol |   .0106408   .0015267     6.97   0.000     .0076485    .0136332
##          sbp |   .0180675   .0041204     4.38   0.000     .0099917    .0261433
##          bmi |   .0549478   .0265311     2.07   0.038     .0029478    .1069478
##        smoke |   .6038582   .1410863     4.28   0.000     .3273341    .8803823
##       dibpat |   .6965686   .1443722     4.82   0.000     .4136043     .979533
##        _cons |  -12.27086   .9821107   -12.49   0.000    -14.19577   -10.34596
## ------------------------------------------------------------------------------
##
##
##
## ------------------------------------------------------------------------------------------
##      Dependent variable: chd69     Equation: chd69     Command: logistic
## Covariates set to value: age = 40, bmi = 25, chol = 400, sbp = 130, smoke = 0, dibpat =
## ------------------------------------------------------------------------------------------
##
## ------------------------------------------------
##       All |         pr          lb          ub
## ----------+-------------------------------------
##           |    .133052    [.078757    .216001]
## ------------------------------------------------
##      Key:  pr         =  Probability
##            [lb , ub]  =  [95% Confidence Interval]
##
##
## ------------------------------------------------------------------------------------------
##      Dependent variable: chd69     Equation: chd69     Command: logistic
## Covariates set to value: age = 40, bmi = 25, chol = 400, sbp = 130, smoke = 0, dibpat =
## ------------------------------------------------------------------------------------------
##
## ------------------------------------------------
##       All |         xb          lb          ub
## ----------+-------------------------------------
##           |   -1.87424   [-2.45935   -1.28913]
## ------------------------------------------------
##      Key:  xb         =  Linear Prediction
```

```
##            [lb , ub]  =  [95% Confidence Interval]
##
## .13305188
##
## .07875748
##
## .2160001
```

Some nice plots of predicted probabilities versus a (continuous) covariate can be obtained using the command margins available both in Stata and R - see lecture 1. Note that the predicted probabilities you may get from a logistic regression model used to analyse case control studies are not reliable. Only ORs can be estimated with such a retrospective design. We defer for now the notion of prediction accuracy, i.e. how well a logistic regression model predicts. This will be discussed in week 11.

10.2.1 Investigation:

1) calculate the predicted probability of CHD occurence for a patient with the following characteristics: *age=50*, *BMI=27*, *chol=200*, *sbp=150*, *smoke=1*, *dibpat=0*. Give the 95% CI.

2) Represent the probability of an event as a function of age for a particular patient profile, e.g. use *BMI=27*, *chol=200*, *sbp=150*, *smoke=1*, *dibpat=0* and let *age* free to vary. Hint: look at the Stata/R code provided in the lecture to produce this plot using the command/library *margins* and related plots.

3) Contrast with a plot of the CHD probability vs age for *smoke=0*, the other characteristics remaining the same. Draw the 2 plots side-by-side.

10.3 Residuals and other diagnostic tools {.unnumbered}}

Raw residuals were calculated as the difference between observed and fitted values with several standardised versions available. A natural way to extend this idea in logistic regression is to compute the Pearson residuals defined as a standardised difference beween the binary endpoint and the prediction, i.e.

$$r_{P,i} = \frac{y_i - \hat{p}_i}{\sqrt{\hat{p}_i(1-\hat{p}_i)}}$$

In other words, the Pearson residual has the following structure "(observed - expected)/SD(observed)" for the corresponding observation. The standardisation performed here corresponds to dividing by the standard deviation of the response y_i both other variants have also been suggested. They cannot be easily represented due to the discrete nature of

the outcome. Another form of residuals exists for this model, they are called the deviance residuals ($r_{D,i}$). The formula is omitted for simplicity but the deviance residuals measure the disagreement between the maxima of the observed and the fitted log-likelihood functions. Since logistic regression uses the maximal likelihood principle, the goal in logistic regression is to minimize the sum of the deviance residuals, so we do something similar to what we do in linear regression. Both types of residuals are readily available in standard packages. One can wonder whether the Pearson or the deviance residuals are normally distributed and the answer is "no". A normal probability plot of either type is usually of little interest since the plot will typically show a kick (a broken line) due to the discreteness of the outcome, even when we simulate data from a logistic regression model. Still other plots can be very helpful such as the plot of the residuals versus prediction or an index plot. Before we examine this in detail on an example, the notion of leverage also exist in logistic regression and is correspond to the diagonal element h_{ii} of the "hat matrix" that has a slightly more complicated definition than in linear model but a similar interpretation. It is not always easy to see on scatter points whether an observation is a leverage point since the fitted curve is no longer a hyperplane geometrically. An approximation to the Cook's distance is and the various dfbetas are also available in this model. They allow us to examine whether some observations are unduly influential on the fit.

Lecture 2 in R

[Download video here]https://www.dropbox.com/s/i8vaa1cfi326kjv/RM1_week10_lecture2_R.mp4?dl=1)

Lecture 2 in Stata

Download video here

10.4 Model checking - diagnostics {.unnumbered}}

Cheking the model assumptions is just as important in logistic regression as it is in linear regression. First, we work under the assumption that the observations are independent. If some form of clustering is anticipated, a more complex modelling is required. This is essentially a design issue and in most cases we know from the start whether this assumption is met or not. Since we do not have an error term *per se'*, no checks of distributional assumptions analogous to normally distributed residuals and constant variance are required. This comes from the fact that the probability distribution for binary outcomes has a simple form that does not include a separate variance parameter. However, we still need to check linearity and whether outliers or influential observations are present in the data making inference invalid. We will defer the examination of linearity until next week and focus on outliers and influential observations in this section.

As an example, we use the *final* model proposed by Vittinghof et al. 2012) for *chd69* of the WGCS study. The authors don't delve into their model building strategy so we will not speculate for now and simply reproduce their analysis with some additional checks. The

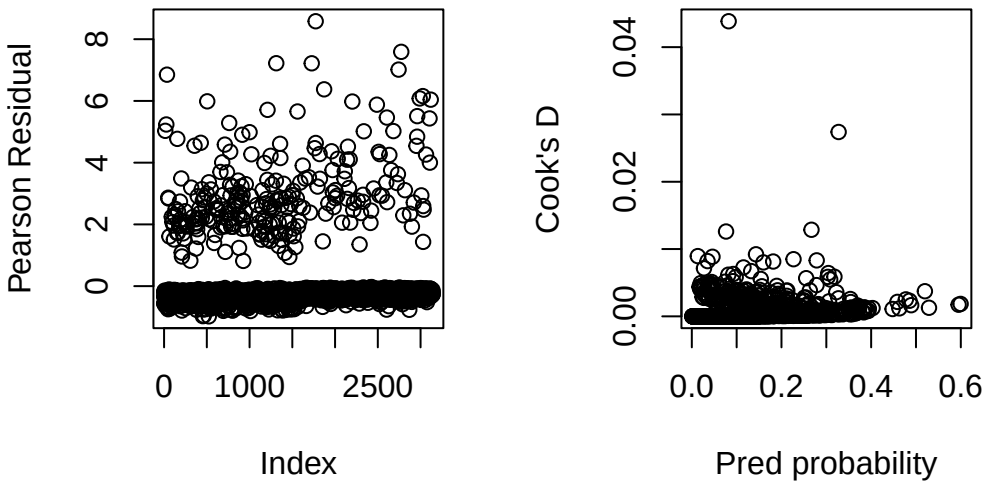
selected covariates are *age_10*, *chol_50*, *sbp_50*, *bmi_10*, *smoke*, *dibpat*, and two interactions terms involving 2 continuous previous predictors named *bmichol* and *bmisbp*. Their results can be reproduced using the code below. Note that they have scaled and CENTRED variables otherwise we get different results. The sample means for age, cholesterol, SBP and BMI were used to centre the variables.

There are different types of useful residual plots we can produce: one is an index plot displaying the Pearson (or deviance) residual vs the ordered observation number in the dataset. This plot allows us to identify large residuals. Another possibly the plot of the Cook's distance vs the predicted probability. These two plots can be found side-by-side in the figure below, see also p. 175-176 in Vittinghof et al. (2012). You can also examine other diagnostic tools like leverage and draw other plots e.g. Pearson (or deviance) residuals vs probabilities

R code and output

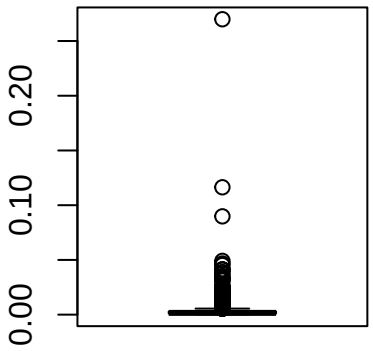
```
# rescale and centre variables
wcgs$age_10<-(wcgs$age-mean(wcgs$age))/10
wcgs$bmi_10<-(wcgs$bmi-mean(wcgs$bmi))/10
wcgs$sbp_50<-(wcgs$sbp-mean(wcgs$sbp))/50
wcgs$chol_50<-(wcgs$chol-mean(wcgs$chol,na.rm=T))/50
myvars <- c("id","chd69", "age", "bmi", "chol", "sbp", "smoke", "dibpat", "age_10", "bmi_1
wcgs2 <- wcgs[myvars]
wcgs2<-wcgs2[wcgs2$chol<645,]
wcgs2cc<-na.omit(wcgs2)
dim(wcgs2cc) # remove missing data --> complete case (cc)
## [1] 3141   12

wcgs2cc$bmichol<-wcgs2cc$bmi_10*wcgs2cc$chol_50
wcgs2cc$bmisbp<-wcgs2cc$bmi_10*wcgs2cc$sbp_50

model3<-glm(chd69 ~ age_10 + chol_50 + sbp_50 +  bmi_10 + smoke + dibpat + bmichol + bmisb
summary(model3)
##
## Call:
## glm(formula = chd69 ~ age_10 + chol_50 + sbp_50 + bmi_10 + smoke +
##     dibpat + bmichol + bmisbp, family = binomial, data = wcgs2cc)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.41606    0.15047 -22.702  < 2e-16 ***
## age_10       0.59497    0.12011   4.954 7.29e-07 ***
## chol_50      0.57571    0.07779   7.401 1.35e-13 ***
## sbp_50       1.01965    0.20660   4.935 8.00e-07 ***
```

```
## bmi_10       1.04884    0.29982   3.498 0.000468 ***
## smoke        0.60619    0.14105   4.298 1.73e-05 ***
## dibpat       0.72343    0.14490   4.993 5.96e-07 ***
## bmichol     -0.88969    0.27465  -3.239 0.001198 **
## bmisbp      -1.50346    0.63182  -2.380 0.017332 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1774.2  on 3140  degrees of freedom
## Residual deviance: 1576.0  on 3132  degrees of freedom
## AIC: 1594
##
## Number of Fisher Scoring iterations: 6

# compute residuals and diagnostic tool (leverage, Cook's distance)

wcgs2cc$devres <- residuals(model3)          # deviance residuals
wcgs2cc$res <- residuals(model3, "pearson")  # Pearson residuals
wcgs2cc$pred.prob <- fitted(model3)          # predicted prob
wcgs2cc$lev  <- hatvalues(model3)            # leverage
wcgs2cc$cd  <- cooks.distance(model3)        # Cook's distance
#i <- order(-wcgs1cc$lev)  # sort by decreasing leverage
par(mfrow=c(1,2))
plot(wcgs2cc$res,ylab="Pearson Residual")
plot(wcgs2cc$pred.prob, wcgs2cc$cd, xlab="Pred probability", ylab="Cook's D")
```




```
# high CD
wcgs2cc<-wcgs2cc[order(-wcgs2cc$cd),]
wcgs2cc[1:5,]
##          id chd69 age      bmi chol sbp smoke dibpat      age_10     bmi_10
## 2734 10078     1  43 38.94737  188 166     0      0 -0.32786937  1.4428994
## 712  12453     0  47 11.19061  188 196     1      1  0.07213063 -1.3327763
## 599  12281     1  48 29.85652  308 186     1      1  0.17213063  0.5338145
## 2119 10196     1  52 19.22490  298 102     1      0  0.57213063 -0.5293477
## 2432  3175     1  46 20.52444  337 110     1      0 -0.02786937 -0.3993938
##          chol_50     sbp_50    bmichol     bmisbp     devres        res
## 2734 -0.7674475  0.7473431 -1.1073495  1.0783409  2.2357619  3.3427774
## 712  -0.7674475  1.3473431  1.0228358 -1.7957069 -0.8907384 -0.6977971
## 599   1.6325525  1.1473431  0.8714802  0.6124684  1.6254118  1.6574372
## 2119  1.4325525 -0.5326569 -0.7583184  0.2819607  2.2679381  3.4769227
## 2432  2.2125525 -0.3726569 -0.8836797  0.1488369  1.9716135  2.4462330
##       pred.prob         lev          cd
## 2734 0.08214118 0.033022199 0.043847556
## 712  0.32746925 0.269896352 0.027393268
## 599  0.26687318 0.038937547 0.012867584
## 2119 0.07640008 0.009212782 0.012606008
```

```
## 2432 0.14318327 0.013496547 0.009221012
# cases 10078 and 12453

# high leverage
boxplot(wcgs2cc$lev)
wcgs2cc<-wcgs2cc[order(-wcgs2cc$lev),]
wcgs2cc[1:5,]
##         id chd69 age      bmi chol sbp smoke dibpat     age_10     bmi_10
## 712  12453     0  47 11.19061  188 196     1      1 0.07213063 -1.3327763
## 2800 10214     0  51 37.65281  153 170     1      0 0.47213063  1.3134431
## 332  13390     0  47 34.69812  129 130     0      1 0.07213063  1.0179742
## 772  19088     0  53 31.92690  133 180     0      1 0.67213063  0.7408528
## 1235 22076     0  48 37.24805  181 154     0      1 0.17213063  1.2729680
##         chol_50     sbp_50   bmichol      bmisbp     devres        res
## 712  -0.7674475 1.34734306  1.022836 -1.79570685 -0.8907384 -0.6977971
## 2800 -1.4674475 0.82734306 -1.927409  1.08666801 -0.7674821 -0.5852130
## 332  -1.9474475 0.02734306 -1.982451  0.02783453 -0.8074791 -0.6208309
## 772  -1.8674475 1.02734306 -1.383504  0.76111000 -0.6472336 -0.4827077
## 1235 -0.9074475 0.50734306 -1.155152  0.64583147 -0.7245481 -0.5478658
##      pred.prob        lev          cd
## 712  0.3274693 0.26989635 0.027393268
## 2800 0.2551067 0.11636894 0.005671273
## 332  0.2782029 0.08980852 0.004642545
## 772  0.1889744 0.04911439 0.001406300
## 1235 0.2308621 0.04670133 0.001713867
# case 12453 stands out again
```



Stata code and output - figures omitted

```
use wcgs.dta
** sample mean will all obs is used to centre the variables
gen age10=(age-46.27869)/10
gen bmi10=(bmi-24.51837)/10
gen chol50=(chol-226.3724)/50
gen sbp50=(sbp-128.6328)/50
** interaction terms
gen bmichol=bmi10*chol50
gen bmisbp=bmi10*sbp50
 ** missing (all in chol) and outlier removed
drop if chol >= 645

logistic chd69 age10 chol50 sbp50 bmi10 smoke dibpat  bmichol bmisbp, coef
predict hat, hat

** dbeta similar to Cooks distance (due to Pregibon) - different from R
predict cookd, dbeta
predict resP, res
predict dv, dev
predict proba, pr
```

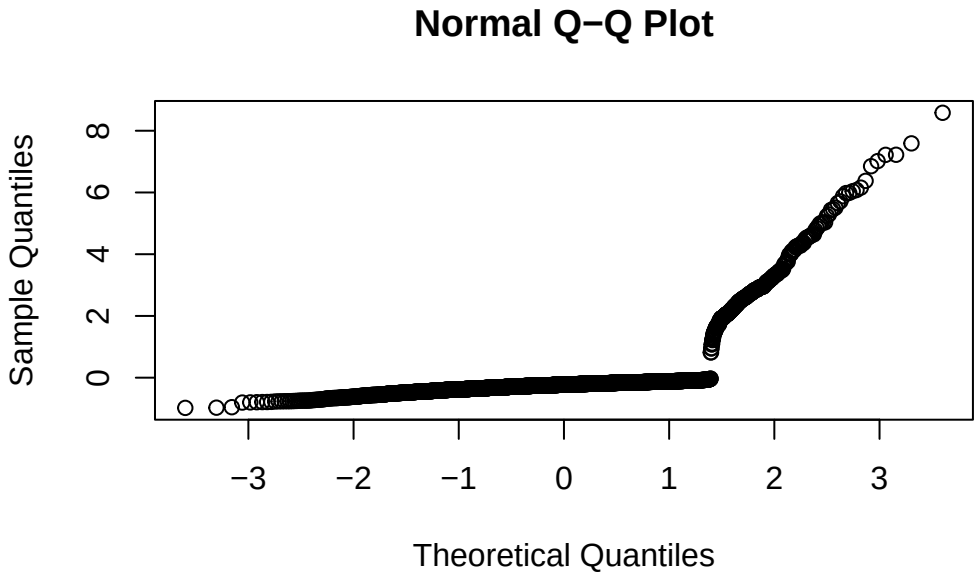
```
** various plots

gen index=_n
scatter resP ind, yline(0) name(temp1)
scatter cookd proba,  yline(0) name(temp2)
graph combine temp1 temp2

** scatter dv proba,  yline(0) name(temp3)
** scatter resP proba,  yline(0) name(temp4)
** graph combine temp3 temp4


gsort -cookd
list id cookd  chd69 age chol chol sbp bmi smoke in 1/5
** case 10078 has dbeta twice as big as the next one


graph box hat
gsort -hat
list id hat  chd69 age chol chol sbp bmi smoke in 1/5
## (12 missing values generated)
##
##
## (12 missing values generated)
##
##
## (13 observations deleted)
##
##
## Logistic regression                                     Number of obs =  3,141
##                                                         LR chi2(8)    = 198.15
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -788.01957                             Pseudo R2     = 0.1117
##
## ------------------------------------------------------------------------------
##        chd69 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##        age10 |   .5949712   .1201093     4.95   0.000     .3595612    .8303811
##       chol50 |   .5757133   .0777901     7.40   0.000     .4232475    .7281792
##        sbp50 |   1.019648   .2066016     4.94   0.000     .6147161     1.42458
##        bmi10 |    1.04884   .2998181     3.50   0.000     .4612068    1.636472
```

```
##        smoke |   .6061928   .1410534     4.30   0.000     .3297331    .8826525
##       dibpat |   .7234267   .1448997     4.99   0.000     .4394284    1.007425
##      bmichol |  -.8896929   .2746472    -3.24   0.001    -1.427992   -.3513942
##       bmisbp |  -1.503455   .6318153    -2.38   0.017     -2.74179   -.2651194
##        _cons |  -3.416062    .150472   -22.70   0.000    -3.710982   -3.121143
## ------------------------------------------------------------------------------
##
##
##
##
##
##
##
##
##
##
##
##
##
##      +-------------------------------------------------------------------------+
##      |    id       cookd   chd69   age   chol   chol   sbp        bmi   smoke |
##      |-------------------------------------------------------------------------|
##   1. | 10078   .3946279       1    43    188    188   166   38.94737       0 |
##   2. | 12453   .2465399       0    47    188    188   196   11.19061       1 |
##   3. | 12281   .1158082       1    48    308    308   186   29.85652       1 |
##   4. | 10196   .1134538       1    52    298    298   102    19.2249       1 |
##   5. |  3175   .0829891       1    46    337    337   110   20.52444       1 |
##      +-------------------------------------------------------------------------+
##
##
##
##
##
##      +-------------------------------------------------------------------------+
##      |    id        hat   chd69   age   chol   chol   sbp        bmi   smoke |
##      |-------------------------------------------------------------------------|
##   1. | 12453   .2698967       0    47    188    188   196   11.19061       1 |
##   2. | 10214   .1163691       0    51    153    153   170   37.65281       1 |
##   3. | 13390   .0898088       0    47    129    129   130   34.69812       0 |
##   4. | 19088   .0491144       0    53    133    133   180    31.9269       0 |
##   5. | 22076   .0467014       0    48    181    181   154   37.24805       0 |
##      +-------------------------------------------------------------------------+
```

We clearly see the grouping of residuals by CHD status on the left panel but there is no

indication that an observation has a residual that is much larger than the others. Note that some of these residuals are well above 2 and that's perfectly fine. The standardisation carried out in the standard Pearson residuals does not mean that their variance is 1. A similar plot can be produced for the deviance residuals (omitted) and will lead to a similar interpretation. The plot of the Cook's distance vs the predicted probabilities identifies two observations with a slightly bigger Cook's distance (CD). We can identify who they are by sorting the data by decreasing CD and printing the top 2-5 observations. The one with the largest CD is patient 10078 with CHD who does not smoke, is very obese (BMI=38.9) and a cholesterol below average (188 mg/dL = 4.86 mmol/L). The next one is case 12453 who did not have CHD, smokes and has a very high SBP (196). Although these two observations stand out, they are not overly influential on the fit. A case-deletion and refit would lead us to a similar conclusion. Note that you may get different values in Stata but similar looking plots (up tp scaling). This has to do with the way the Cook's distance is calculated. A lot of other plots can be produced like in linear regression: you can for instance compute the dfbetas (one per parameter) and plot them to identify influential observations or use leverage values again for other plots. There is no real equivalent of residuals-squared versus leverage. Plot of Pearson or deviance residuals versus a particular covariate are not particularly useful to identify remaining structures than the model may not have captured (like a quadratic trend in age). This will be examined using splines - see week 11 material.

We said earlier that we should not expect the Pearson or deviance residuals to be normally distributed. What happens if we draw a normal probability plot anyway? The figure below displays such a plot and we can clearly see the separation between two groups based on the outcome status and plot is a broken line.



R-users may be able to produce a more meaningful plot using the library *gamlss* . This library allows you to fit a large variety of models that will be explored in RM2. You can force *gamlss* to fit a logistic regression model and get the exact same fit as the standard *glm* command. The code can be found here. The advantage is that a standard plot of the output gives you a much nicer plot of residuals. These resdiduals are called randomised quantile residuals due to Dunn and Smyth (2012). They essentially use randomisation to achieve continuous residuals when the response variable is discrete. Irrespective of the technicality (details to be given in RM2), R produces very nice plots for these residuals.

R code and output

```
par(mfrow=c(1,1))
require(gamlss)
## Loading required package: gamlss
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'gamlss'
out4<-gamlss(chd69 ~ age_10 + chol_50 + sbp_50 +  bmi_10 + smoke + dibpat + bmichol + bmis
## Error in gamlss(chd69 ~ age_10 + chol_50 + sbp_50 + bmi_10 + smoke + dibpat + : could n
plot(out4)
## Error in eval(expr, envir, enclos): object 'out4' not found
```

A nice straight QQ plot of this randomised quantile residuals is produced meaning that there are no particular issue in the residuals to be concerned about. R-users can try to run the code given above and first verify that *gamlss* and *glm* gave the same fit. The plot command issued right after the fit returns though a completely different figure due to a different implementation in *gamlss*. There is no equivalent to this function *gamlss* in Stata that we know of.

The following are the key takeaway messages from this week:

1. The concept of interaction is similar to the one used in linear regression when expressed on the logit scale. Effect modification is mutiplicative on the odds-ratio scale
2. Predicted probabilities of an event adn 95% CI can be calculated for any patient's profile. Transforming the linear predictor and its 95% CI to the probability scale is the way to go.
3. Residuals can also be extended (e.g. the Pearson and deviance residuals) but they are not normally distributed.
4. Other diagnostic tools (e.g. leverage, Cook's distance) exist and have similar interpretarion. They cam help identify outliers and influential observations.

11 Logistic Regression

Learning objectives

By the end of this week you should have

1. Deepened your understanding of diagnostic tools
2. Discovered how to assess linearity in logistic regression models
3. Learned about splines and other flexible methods in this setting
4. Gained familiarity with ROC curves
5. Understood pros and the cons of goodness of fit techniques in logistic regression

Learning activities

This week's learning activities include:

Learning Activity	Learning objectives
Lecture 1	1, 2 3,
Reading	2, 3
Lecture 2	4, 5
Investigation	3
Discussions/tutorial	2, 3, 4

Lecture 1 in R

Download video here

Lecture 1 in Stata

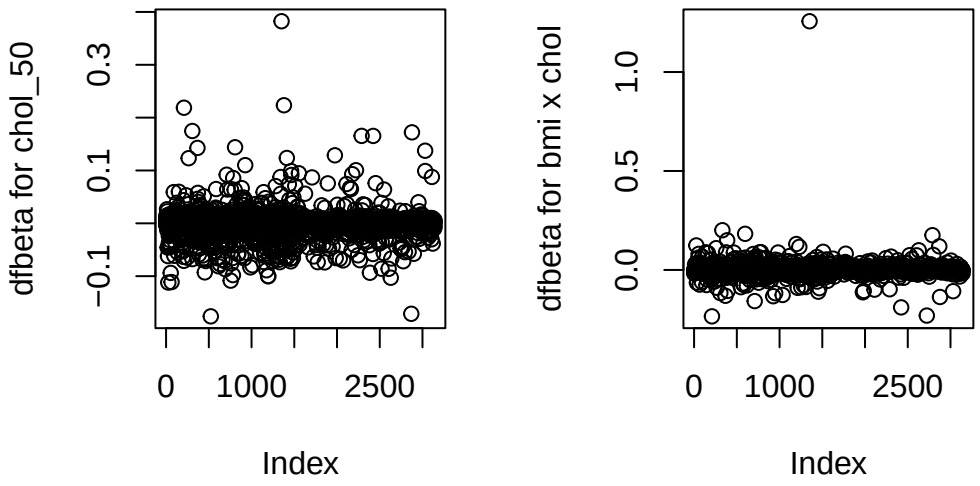
Download video here

Other diagnostic tools

Residuals, leverage, Cook's distance and all related plots are not the only diagnostic tools you can use in logistic regression. Similarly to linear regression you can compute *dfbetas* for each coefficient. They have the same interpretation, i.e. they measure the standardized difference between a regression coefficient before and after the removal of each observation in turn. As an example, we go back to the final for *chd69* we used earlier and compute dfbetas for *chol_50* and *bmichol* but this time we keep case 12237 who has a very high cholesterol reading (645 mg/dL=16.68 mmol/L). The figure below shows that this observation has the highest dfbeta on the two plots. It's particularly extreme for *bmichol*, meaning that this leverage point was particularly influential on this parameter.

R code and output

```
wcgs <- read.csv("wcgs.csv")
wcgs<-data.frame(wcgs)
# centre variables  (use wcgs again)
wcgs$age_10<-(wcgs$age-mean(wcgs$age))/10
wcgs$bmi_10<-(wcgs$bmi-mean(wcgs$bmi))/10
wcgs$sbp_50<-(wcgs$sbp-mean(wcgs$sbp))/50
wcgs$chol_50<-(wcgs$chol-mean(wcgs$chol,na.rm=T))/50
myvars <- c("id","chd69", "age", "bmi", "chol", "sbp", "smoke", "dibpat",
            "age_10", "bmi_10", "chol_50", "sbp_50")
wcgs3 <- wcgs[myvars]
wcgs3cc<-na.omit(wcgs3)  # here case with chol=645 is  kept, missing deleted
# 3142x 12
wcgs3cc$bmichol<-wcgs3cc$bmi_10*wcgs3cc$chol_50
wcgs3cc$bmisbp<-wcgs3cc$bmi_10*wcgs3cc$sbp_50
out5<-glm(chd69 ~ age_10 + chol_50 + sbp_50 +  bmi_10 + smoke + dibpat + bmichol + bmisbp,
#summary(out5)
dfbetas<-dfbetas(out5) # 3142  x 9
par(mfrow=c(1,2))
plot(dfbetas[,3], ylab="dfbeta for chol_50")
plot(dfbetas[,8], ylab="dfbeta for bmi x chol")
```



Stata code and output

```
clear
use wcgs.dta
gen age10=(age-46.27869)/10
gen bmi10=(bmi-24.51837)/10
gen chol50=(chol-226.3724)/50
gen sbp50=(sbp-128.6328)/50
gen bmichol=bmi10*chol50
gen bmisbp=bmi10*sbp50
** remove missing
drop if missing(chd69) | missing(bmi) | missing(age) | missing(sbp) | missing(smoke) | mis
** n=3142 observations
logistic chd69 age10 chol50 sbp50 bmi10 smoke dibpat  bmichol bmisbp, coef
ldfbeta
desc
** Stata has created dfbeta for each coefficient
gen index=_n
scatter DFchol50 index, name(temp1) mlab(id)
scatter DFbmicho index, name(temp2) mlab(id)
graph combine temp1 temp2
** the figure is not displayed (Mardown compatibility with Stata)
```

```
** but you will see it when running the code
## (12 missing values generated)
##
##
## (12 missing values generated)
##
##
## (12 observations deleted)
##
##
## Logistic regression                                     Number of obs =  3,142
##                                                         LR chi2(8)    = 200.57
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -789.31393                             Pseudo R2     = 0.1127
##
## ------------------------------------------------------------------------------
##        chd69 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##        age10 |   .5914968   .1200155     4.93   0.000     .3562708    .8267229
##       chol50 |   .5948924   .0766472     7.76   0.000     .4446667    .7451181
##        sbp50 |   1.009179   .2065092     4.89   0.000     .6044289     1.41393
##        bmi10 |   1.010223   .3009453     3.36   0.001     .4203815    1.600065
##        smoke |   .5947879   .1406769     4.23   0.000     .3190663    .8705096
##       dibpat |   .7221468   .1448671     4.98   0.000     .4382124    1.006081
##      bmichol |  -.6920673   .2439215    -2.84   0.005    -1.170145     -.21399
##       bmisbp |  -1.395698   .6280649    -2.22   0.026    -2.626683   -.1647136
##        _cons |  -3.411468   .1501942   -22.71   0.000    -3.705844   -3.117093
## ------------------------------------------------------------------------------
##
## command ldfbeta is unrecognized
## r(199);
##
## r(199);
```

It's worth noting that the Cook's distance and leverage would have identified this case as well (Cook's D=.107, leverage=0.297 in R), so Vittinghof et al. (2012) were right to remove this observation from their analysis. Note that *dfbetas* are obtained via the command *ldfbeta* in Stata, this is slightly different from the code used for the linear model. You may also find other diagnostic tools in your favourite software, we let you explore this further. We have presented here the ones that are more commonly used and readily available in all statistical packages. We will end this paragraph by saying that multicollinearity can occur in logistic regression for the same reason as in the linear model. VIF and various other indicators can be obtained by

installing *collin* in Stata (i.e. type *net install collin*) and running it. A generalised version of the concept called (GVIF) is available in R via the library *car*. The interpretation is the same as in linear regression so we will not illustrated its use in this context.

Checking linearity

Categorising

One of the logistic model assumption is linearity on the log-odds scale and this should be checked. Unlike linear regression, we cannot really use the Pearson or deviance residuals and examine whether some structure remains there and scatter plots are not meaningful for a binary endpoint. There are several ways you can check this assumption and, perhaps, the most common one consists of creating categories for a particular predictor of interest. This is the logic we followed in week 9 where *agec* representing decades of age was used instead of *age* both in a $2 \times k$ table and a logistic model. If linearity is met, you expect to find a regular increase or decrease in the coefficient with each additional decade. There are obvious disadvantages with this approach: first, the choice of the categories is not obvious; second, categorising may result in small numbers in some categories for which it's difficult to conclude anything; third, it may have an impact on the interpretation and in some cases can even mask existing structures.

Using polynomials

An important outcome from a hospital resources point of view is whether or not a newborn baby is admitted to neonatal intensive care. The dataset *birth* records information on 141 babies including the response variable, i,e admittance to neonatal intensive care (*admin_nc* coded on 0/1 with 0=`no'' and 1=`yes''), weight in kg (*weight*) and a bunch of other predictors. A boxplot of *weight* by *admin_nc* (omitted) shows that babies with *admit.nc*=1 tend to have lower birthweights. A simple logistic shows a very significant effect of *weight* with $\widehat{\beta}_1 = -1.64$, $p = 7.7e-5$ meaning that the probability of admission to neonatal intensive care *decreases* with increasing birthweight.

R code and output

```
birth <- read.csv("birth.csv")
birth<-data.frame(birth)
model1 <- glm(admit.nc ~ weight, family=binomial, data=birth)
summary(model1)
##
## Call:
```

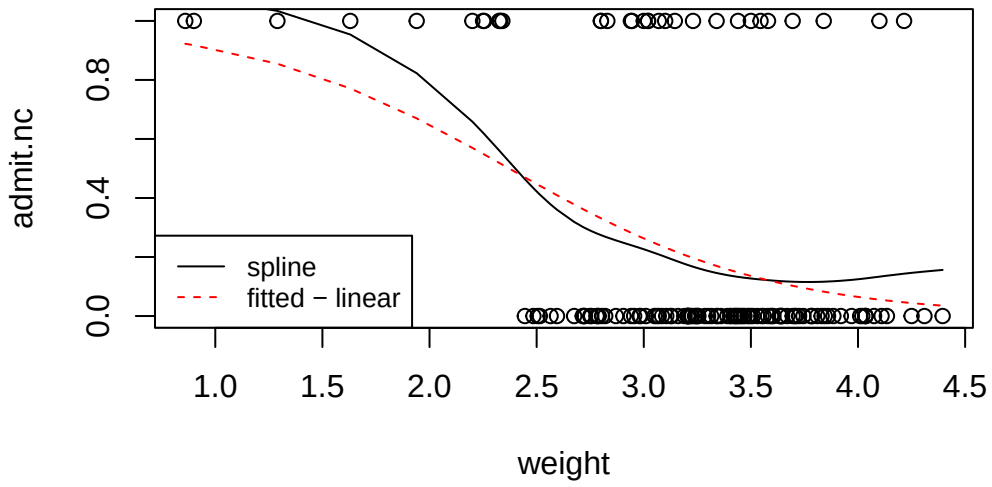
```
## glm(formula = admit.nc ~ weight, family = binomial, data = birth)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.8878     1.2865   3.022  0.00251 **
## weight       -1.6394     0.4147  -3.953 7.72e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 151.03  on 140  degrees of freedom
## Residual deviance: 129.88  on 139  degrees of freedom
## AIC: 133.88
##
## Number of Fisher Scoring iterations: 4
```

Stata code and output

```
use birth.dta
logistic admitnc weight, coef
** OR for a 0.5 kg increase in weight
disp exp(-1.64*0.5)
lincom 0.5*weight, or
## Logistic regression                                     Number of obs =    141
##                                                         LR chi2(1)    =  21.15
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -64.938259                             Pseudo R2     = 0.1401
##
## ------------------------------------------------------------------------------
##      admitnc | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##       weight |  -1.639373   .4147598    -3.95   0.000    -2.452287   -.8264585
##        _cons |   3.887806   1.286577     3.02   0.003     1.366161    6.409451
## ------------------------------------------------------------------------------
##
## .44043165
##
##
##  ( 1)  .5*[admitnc]weight = 0
##
## ------------------------------------------------------------------------------
```

```
##      admitnc | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
## -------------+------------------------------------------------------------------
##          (1) |   .4405698   .0913653    -3.95   0.000     .2934219    .6615106
## ------------------------------------------------------------------------------
```

We can interpret the output in terms of OR but we need to choose a meaningful weight increase of, say, 500g=0.5kg, yielding $OR = \exp(-1.64 * 0.5) = 0.44$. So an increase in birthweight of 500g decreases the odds of admittance to neonatal intensive care by 66% (assuming that the model is correct). Fitted and smoothed (spline) curves are plotted below.



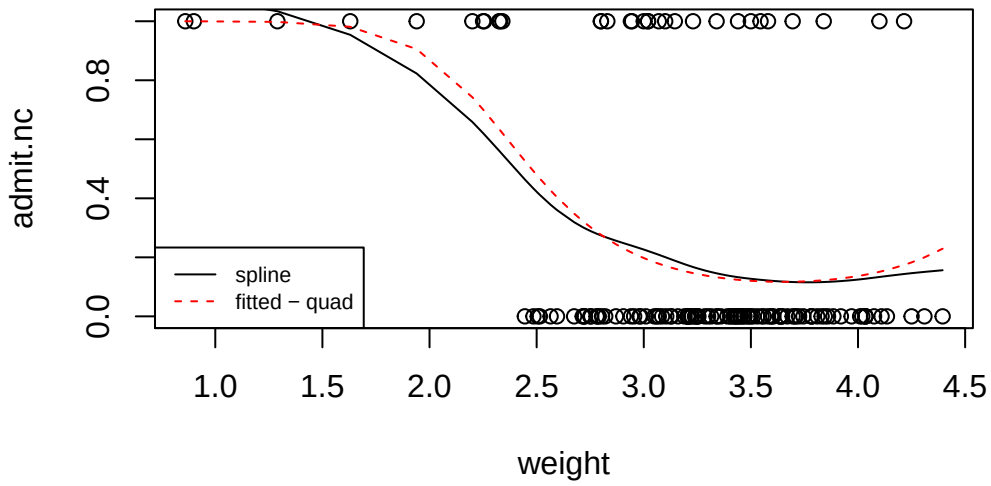
R code

```
plot(admit.nc~weight,data=birth)
lines(smooth.spline(birth$admit.nc~birth$weight,df=5))
# NB: smooth.spline (fits a smoothing spline to the supplied data)
lines(fitted(model1)[order(birth$weight)]~sort(birth$weight),lty=2,col="red")
legend("bottomleft",legend=c("spline","fitted - linear"),lty=1:2, cex=0.8,
       col=c("black","red"))
```

Stata code and output (with a slightly different smoother)

```
clear
use birth.dta
sort weight
logistic admitnc weight
predict proba, pr
graph twoway (lpoly admitnc weight, degree(1) kernel(epan2) bwidth(0.8))(line proba weight
** the figure is not displayed (Mardown compatibility with Stata)
** but you will see it when running the code
## Logistic regression                                     Number of obs =    141
##                                                         LR chi2(1)    =  21.15
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -64.938259                             Pseudo R2     = 0.1401
##
## ------------------------------------------------------------------------------
##      admitnc | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##       weight |   .1941017   .0805056    -3.95   0.000     .0860964    .4375963
##        _cons |   48.80371   62.78974     3.02   0.003     3.920273    607.5603
## ------------------------------------------------------------------------------
## Note: _cons estimates baseline odds.
```

It's clear that the fitted curve doesn't quite get the curvature that we see in the smoothing spline; this means that the effect of birthweight on the probability of admittance to intensive care is not well captured by this simple model. To get better explanation, we try a quadratic polynomial of birthweight, i.e. fit the model $\log(p/(1-p)) = \beta_0 + \beta_1 weight + \beta_2 weight^2$) to the data.



R code

```
model2 <- glm(admit.nc ~ weight + I(weight^2), family=binomial,data=birth)
plot(admit.nc~weight,data=birth)
lines(smooth.spline(birth$admit.nc~birth$weight,df=5))
# NB: smooth.spline (fits a smoothing spline to the supplied data)
lines(fitted(model2)[order(birth$weight)]~sort(birth$weight),lty=2,col="red")
legend("bottomleft",legend=c("spline","fitted - quad"),lty=1:2, cex=0.7,
       col=c("black","red"))
```

Stata code and output (with a slighly different smoother)

```
clear
use birth.dta
sort weight
gen weight2=weight^2
logistic admitnc weight weight2
predict proba, pr
graph twoway (lpoly admitnc weight, degree(1) kernel(epan2) bwidth(0.8))(line proba weight
** the figure is not displayed (Mardown compatibility with Stata)
** but you will see it when running the code
## Logistic regression                                    Number of obs =    141
##                                                        LR chi2(2)    =  26.59
```

```
##                                                       Prob > chi2   = 0.0000
## Log likelihood = -62.221059                           Pseudo R2     = 0.1760
##
## ------------------------------------------------------------------------------
##      admitnc | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##       weight |    .000023   .0001014    -2.42   0.016     4.02e-09    .1312583
##      weight2 |   4.317368   2.948764     2.14   0.032     1.132005    16.46607
##        _cons |   3.90e+07   2.74e+08     2.49   0.013     41.86106    3.64e+13
## ------------------------------------------------------------------------------
## Note: _cons estimates baseline odds.
```

A much better fit is observed. We can try to fit a cubic polynomial and repeat the procedure. Note that when fitting high order polynomials we may be better off using orthogonal polynomials which results terms that are independent by construction. This can be done using the command *poly*() in R but does not affect the fitted curve and global interpretation. Irrespective of the basis you choose for the polynomials, you should always keep all of the lower order terms in the model, *even if they are not significant*. To decide which model (linear, quadratic or cubic) is better we can use the AIC. We find AIC=133.9, 130.4 and 131.4 respectively, so the quadratic model is better. A similar conclusion would have been reached using BIC (but they don't always agree). We don't pursue further the modelling of *admin_nc*; of course, other predictors could be added that could potentially confound the association with *weight* but they are extremely unlikely to make the quadratic trend identified earlier disappear.

R code to fit the 3 models and output

```
model1 <- glm(admit.nc ~ weight, family=binomial,data=birth)
AIC(model1)
## [1] 133.8765
model2 <- glm(admit.nc ~ weight + I(weight^2), family=binomial,data=birth)
AIC(model2)
## [1] 130.4421
model3 <- glm(admit.nc ~ weight + I(weight^2) + I(weight^3), family=binomial,data=birth)
AIC(model3)
## [1] 131.4079
BIC(model1,model2,model3)
##        df      BIC
## model1  2 139.7740
## model2  3 139.2884
## model3  4 143.2030
```

Stata code to fit the 3 models and output

```
use birth.dta
** various models and AIC
logistic admitnc weight, coef
estat ic
gen weight2=weight^2
logistic admitnc weight weight2, coef
estat ic
gen weight3=weight^3
logistic admitnc weight weight2 weight3, coef
estat ic
## Logistic regression                                     Number of obs =    141
##                                                         LR chi2(1)    =  21.15
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -64.938259                             Pseudo R2     = 0.1401
##
## ------------------------------------------------------------------------------
##      admitnc | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##       weight |  -1.639373   .4147598    -3.95   0.000    -2.452287   -.8264585
##        _cons |   3.887806   1.286577     3.02   0.003     1.366161    6.409451
## ------------------------------------------------------------------------------
##
##
## Akaike's information criterion and Bayesian information criterion
##
## -----------------------------------------------------------------------------
##        Model |          N   ll(null)  ll(model)      df        AIC        BIC
## -------------+---------------------------------------------------------------
##            . |        141  -75.51468  -64.93826       2   133.8765    139.774
## -----------------------------------------------------------------------------
## Note: BIC uses N = number of observations. See [R] BIC note.
##
##
##
## Logistic regression                                     Number of obs =    141
##                                                         LR chi2(2)    =  26.59
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -62.221059                             Pseudo R2     = 0.1760
##
## ------------------------------------------------------------------------------
##      admitnc | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
```

```
##       weight |  -10.68154   4.413832    -2.42   0.016    -19.33249   -2.030588
##      weight2 |   1.462646   .6830003     2.14   0.032        .12399    2.801302
##        _cons |   17.48003   7.013229     2.49   0.013     3.734356    31.22571
## ------------------------------------------------------------------------------
##
##
## Akaike's information criterion and Bayesian information criterion
##
## -----------------------------------------------------------------------------
##        Model |          N   ll(null)  ll(model)      df        AIC        BIC
## -------------+---------------------------------------------------------------
##            . |        141  -75.51468  -62.22106       3   130.4421   139.2884
## -----------------------------------------------------------------------------
## Note: BIC uses N = number of observations. See [R] BIC note.
##
##
##
## Logistic regression                                     Number of obs =    141
##                                                         LR chi2(3)    =  27.62
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -61.703968                             Pseudo R2     = 0.1829
##
## ------------------------------------------------------------------------------
##      admitnc | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##       weight |   -49.2661    41.5909    -1.18   0.236    -130.7828    32.25057
##      weight2 |   13.67673   13.04288     1.05   0.294    -11.88685    39.24031
##      weight3 |  -1.263134   1.343428    -0.94   0.347    -3.896205    1.369936
##        _cons |   57.25872   43.52855     1.32   0.188    -28.05567    142.5731
## ------------------------------------------------------------------------------
## Note: 0 failures and 2 successes completely determined.
##
##
## Akaike's information criterion and Bayesian information criterion
##
## -----------------------------------------------------------------------------
##        Model |          N   ll(null)  ll(model)      df        AIC        BIC
## -------------+---------------------------------------------------------------
##            . |        141  -75.51468  -61.70397       4   131.4079    143.203
## -----------------------------------------------------------------------------
## Note: BIC uses N = number of observations. See [R] BIC note.
```

Splines or more flexible models

The truth is that we were lucky that a quadratic polynomial fit could capture so well the effect of weight on the probability of admission to neonatal intensive care. More often than not, we need more flexible models. Building on what we did in week 7, we could fit restricted cubic splines or fractional polynomials. To illustrate their use here, let's consider the *medcare* data where a public insurance program called medcare collected information on 4406 individuals, aged 66 and over. The objective is to determine what factors can impact poor heath represented here by *healthpoor* (0=average health, 1=poor). Potential predictors are available including *age*, *ofp* the number of physician office visits, *male* (0=female, 1=male), *married* (0=no, 1=yes), years of education (*school*). Age is obviously an important predictor, so for sake of simplicity, we consider a model with a single continuous covariate (*age*) through the standard specification $\log(p/(1-p)) = \beta_0 + \beta_1 age$) where p is the probability that *healthpoor*=1 given age. One way to generalise this function is to add terms for restricted cubic splines (RCS), Say we are interested in a model with with 4 knots, we need to add two terms $S_2(age)$ and $S_3(age)$ yielding:

$$\log(\frac{p}{1-p}) = \beta_0 + \beta_1 age + \beta_2 S_2(age) + \beta_3 S_3(age),$$

with the usual convention that $S_1(age) = age$. As discussed earlier for the linear case, the exact algebraic formulation of S_1 and S_2 is not important as long as we understand that we fit a smooth curve to the data on the logit scale. The curve is based on cubic polynomials in the middle and linear terms before the first knot and after the last knot, where we have less information. The way to create RCS in R and Stata is the same as before. We need to use *logistic* instead of *regress* in Stata to fit the corresponding logistic regression model; also use the command *lrm* instead of *ols* from the *rms* library in R. The R (or Stata) output has the typical format with the age coefficient being diplayed first. The added terms are listed as age' and age'' etc in R whereas Stata lets you choose the names.

R code and output

```
medcare<- read.csv("medcare.csv")
medcare<-data.frame(medcare)
require(rms)
## Loading required package: rms
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'rms'
medcare$age<-medcare$age*10
# assume that age is in years in these data (unlike in the R library)
ddist <- datadist(medcare)
## Error in datadist(medcare): could not find function "datadist"
options(datadist='ddist')
```

```
model1 <- lrm(healthpoor  ~  rcs(age,4), data=medcare)
## Error in lrm(healthpoor ~ rcs(age, 4), data = medcare): could not find function "lrm"
model1
##
## Call:  glm(formula = admit.nc ~ weight, family = binomial, data = birth)
##
## Coefficients:
## (Intercept)       weight
##       3.888       -1.639
##
## Degrees of Freedom: 140 Total (i.e. Null);  139 Residual
## Null Deviance:       151
## Residual Deviance: 129.9     AIC: 133.9
plot(Predict(model1, age))
## Error in Predict(model1, age): could not find function "Predict"
anova(model1)
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: admit.nc
##
## Terms added sequentially (first to last)
##
##
##        Df Deviance Resid. Df Resid. Dev
## NULL                     140     151.03
## weight  1   21.153       139     129.88
# AICs
model0 <- lrm(healthpoor  ~  age, data=medcare)
## Error in lrm(healthpoor ~ age, data = medcare): could not find function "lrm"
AIC(model0)
## Error in eval(expr, envir, enclos): object 'model0' not found
AIC(model1)
## [1] 133.8765
```

Stata code and output

```
use medcare.dta
replace age=age*10
** age in years
```

```
mkspline agespl = age, cubic nknots(4) displayknots
logistic healthpoor agespl*, coef
test agespl2 agespl3
estat ic
logistic healthpoor age, coef
estat ic
** plot
drop agespl1 agespl2 agespl3
mkspline2 agespl = age, cubic nknots(4)
logistic healthpoor agespl*, coef
adjustrcspline if age <= 95, custominvlink("xb()")  ytitle("log-odds")
** the option is necessary to get the plot on the log-odds scale
** defalut = proba in Stata.
** the figure is not displayed (Mardown compatibility with Stata)
** but you will see it when running the code
## (4,406 real changes made)
##
##
##              |     knot1      knot2      knot3      knot4
## -------------+--------------------------------------------
##          age |        66         70         76         86
##
##
## Logistic regression                                    Number of obs =  4,406
##                                                        LR chi2(3)    =  44.71
##                                                        Prob > chi2   = 0.0000
## Log likelihood = -1644.0085                            Pseudo R2     = 0.0134
##
## ------------------------------------------------------------------------------
##   healthpoor | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##      agespl1 |   -.036536   .0508291    -0.72   0.472    -.1361593    .0630872
##      agespl2 |   .4449931   .2730339     1.63   0.103    -.0901435    .9801297
##      agespl3 |   -.833504   .5151554    -1.62   0.106     -1.84319    .1761819
##        _cons |   .2433287   3.465927     0.07   0.944    -6.549763     7.03642
## ------------------------------------------------------------------------------
##
##
##  ( 1)  [healthpoor]agespl2 = 0
##  ( 2)  [healthpoor]agespl3 = 0
##
```

```
##            chi2(  2) =    2.67
##          Prob > chi2 =    0.2636
##
##
## Akaike's information criterion and Bayesian information criterion
##
## -----------------------------------------------------------------------------
##        Model |          N   ll(null)  ll(model)      df        AIC        BIC
## -------------+---------------------------------------------------------------
##            . |      4,406  -1666.363  -1644.009       4   3296.017    3321.58
## -----------------------------------------------------------------------------
## Note: BIC uses N = number of observations. See [R] BIC note.
##
##
## Logistic regression                                     Number of obs =  4,406
##                                                         LR chi2(1)    =  42.05
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -1645.3389                             Pseudo R2     = 0.0126
##
## ------------------------------------------------------------------------------
##   healthpoor | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##          age |   .0441932   .0066896     6.61   0.000     .0310818    .0573046
##        _cons |  -5.240621   .5064129   -10.35   0.000    -6.233172    -4.24807
## ------------------------------------------------------------------------------
##
##
## Akaike's information criterion and Bayesian information criterion
##
## -----------------------------------------------------------------------------
##        Model |          N   ll(null)  ll(model)      df        AIC        BIC
## -------------+---------------------------------------------------------------
##            . |      4,406  -1666.363  -1645.339       2   3294.678   3307.459
## -----------------------------------------------------------------------------
## Note: BIC uses N = number of observations. See [R] BIC note.
##
##
## command mkspline2 is unrecognized
## r(199);
##
## r(199);
```

To interpret the RCS, we don't look as the coefficients per see but plot the curve and assess whether these additional terms are needed. The plot above displays a rather linear effect of age on the log-odds scale (up to a small part of the curve where there does not seem to be real effect for age 66 to 70). Is it enough to justify the more complex model? Since the difference with simple logistic regression is due to S_1 and S_2 that have been added to the model, we can easily test whether they are actually needed using the *anova* command in R or test the relevant parameters (i.e. H_0$: $\beta_2 = \beta_3 = 0$) in Stata. The corresponding p-value is $p = 0.26$ providing little evidence that the RCS are necessary. In other words linearity in age is plausible on this data. Another way to look at this is to use the AIC (or BIC) and compare the two values (with/without splines). This will become particularly helpful when more complex models with several splines are fitted and are not necessarily nested. The AIC is 3294.7 for the simple logistic regression model and 3296.0 for the the RCS model. This favours the simpler model because a smaller AIC is preferable. This is in agreement with both the LRT analysis and the visual impression we get from the plot.

Note that we still have to choose the number of knots (often between 3 and 5) and their location, By default we let the software place them but there may be case where you have a better idea, in which case you can modify the command and list where the knots are. This is similar to what we do in linear regression (see week 6 material). We have illustrated the concept of RCS here but you could add fractional polynomials of age like $\sqrt{age}$ or age^2 instead of S_1 and S_2 and proceed the same way. The plots in the RCS approach may also help find the type of fractional polynomials of potential interest.

Investigation: medcare data

In this activity, we will conduct a more thorough analysis of the *medcare* date and consider whether linearity is satisfied for the other continuous covariates.

1) Start by fitting the initial model including *male*, *married*, *age*, *school* and *ofp* (model0) and get the corresponding AIC. Do not try to remove the nonsignficant predictors for now.

2) The age effect is assumed to be linear, now we need to assess whether linearity holds for *school* and *ofp*. Fit a model with RCS(4) for both of these covariates (model1). Plot the two splines separately (the command(s) we used for the linear model works here just the same). Are they necessary? Comment on the AIC of this model (compared with model0's). What would you recommend at this stage?

Note: here you have to plot two splines (*ols* and *school*). Stata users have to repeat the same procedure (spline definition and refit) to plot the 2nd spline. Otherwise the 2nd spline is not displayed. Also you need to specify the values for the other variables in the model, e.g. use the sample medians. R-users don't need to worry because the package does it for you.

3) There still seems to be some downward curvature in the relationship between *school* and *healthpoor* on the log-odds scale. We decide to investigate this further by replacing the RCS in *school* by a polynomial. Discuss the appropriateness of fitting a quadratic polynomial in *school* (model 2); use preferably the *poly()* command in R. NB: you can keep the RCS(4) for *ofp* as in 2) if need be.

4) What is the better fitted model based on the AIC (or BIC)?

5) Is there a way to lower the AIC (or BIC) further? What do suggest we do?

6) Write your conclusions

Note that in terms of interpretation of the splines, we have the same options as in linear regression. For simplicity, the splines will only be interpreted qualitatively. Present plots and stay away from a more complex quantitative explanation. This is what most researchers would do in pratice. You may have a choice of the scale to present the results. We will stick to the logit scale (where linearity is expressed) to keep it as simple as possible.

We have illustrated the use of RCS (and polynomials) on this example but they are part of suite of techniques you can actually use to make your model more flexible and assess linearity. Other types of splines exist and are particular well developed in R. A common strategy is to use smoothing splines that avoid having to choose knots (they are available via the R library *mgcv*). They are mentioned here for the sake of completeness. An interesting illustration on how they work can be found in the following *optional* reading.

11.0.0.0.1 * [@Wand2012] Using Semiparametric Regression and Parametric Regression Models: How to Use Data to Get the (Right) Answer?.

This reading illustrates how a semiparametric model can be fitted to the data and identify structures that could otherwise have been missed using ad-hoc categorisation. The situation described here is a more complex U-shape that can arise in real data. Note as well that ROC (Receiver Operating Curve) and AUC (Area Under the Curve) are used here to assess the model performance of the model. This part can only be understood after studying the next section, so you may want to study it before returning to the reading.

Lecture 2 in R

Download video here

Lecture 2 in Stata

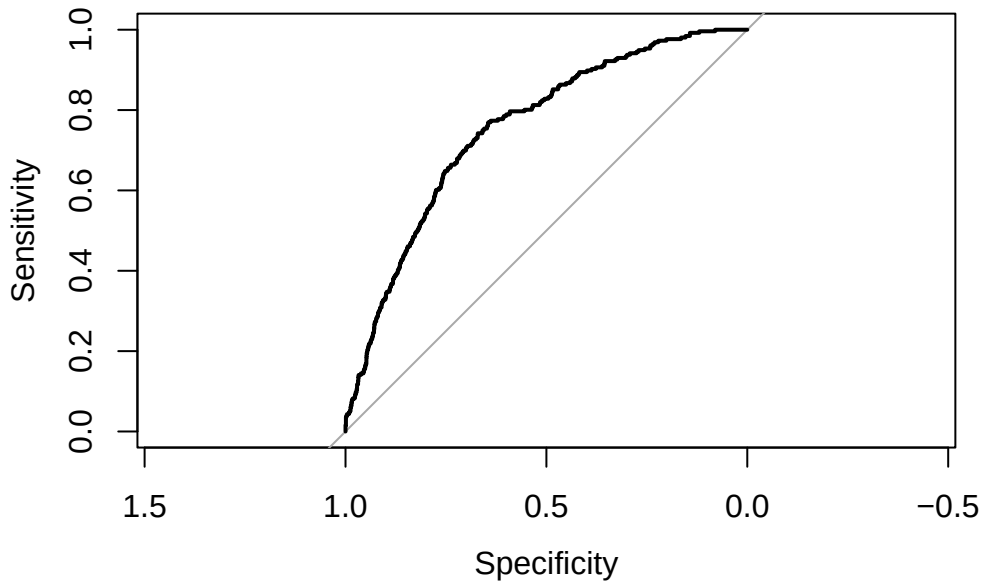
Download video here

11.1 ROC curve

In some applications, we may be interested in using a logistic regression model as a tool to classify outcomes of observed individuals based on values of measured predictors. After model fitting, we can compute the fitted probabilities $\hat{p}_i$ and by using a cutoff decide whether they are predictive of an event or not and compare with the real observations. By doing so, we will determine the sensitivity (i.e. the proportion of events that are well classified), the specificity (i.e. the proportion of no events that are well classified) and their respective complements, the false negative and false positive. The sensitivity and specificity will depend critically on the choice of cutoff probability. In practice, we let this cutoff vary on the (0-1) interval, compute the resulting sensitivity and specificity and draw a plot. Traditionally, it is (1-specificity) that is plotted on the horizontal axis, and sensitivity on the vertical axis. This is called a ROC curve (ROC stands for Receiver Operating Characteristic, which originated in quality control where this technique was first developed). The figure below displays the ROC curve for the final model for *chd69* without the outlier (chol=645 mg/dL)

R code and output

```
library(pROC)
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
wcgs4=wcgs3cc[wcgs3cc$chol < 645,]
dim(wcgs4)
## [1] 3141   14
out6<-glm(chd69 ~ age_10 + chol_50 + sbp_50 +  bmi_10 + smoke + dibpat + bmichol + bmisbp,
wcgs4$pred.prob<-fitted(out6)
wcgs4$pred.prob<-fitted(out6)
roc(wcgs4$chd69,wcgs4$pred.prob, plot=TRUE,ci=TRUE)
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```




```
##
## Call:
## roc.default(response = wcgs4$chd69, predictor = wcgs4$pred.prob,     ci = TRUE, plot = TRU
##
## Data: wcgs4$pred.prob in 2885 controls (wcgs4$chd69 0) < 256 cases (wcgs4$chd69 1).
## Area under the curve: 0.7535
## 95% CI: 0.7243-0.7827 (DeLong)
```

Stata code and ouput

```
use "wcgs.dta"
gen age10=(age-46.27869)/10
gen bmi10=(bmi-24.51837)/10
gen chol50=(chol-226.3724)/50
gen sbp50=(sbp-128.6328)/50
gen bmichol=bmi10*chol50
gen bmisbp=bmi10*sbp50
** remove missing cholesterol values and outlier (chol=645)
drop if chol>=645
logistic chd69 age10 chol50 sbp50 bmi10 smoke dibpat  bmichol bmisbp, coef
lroc
** with 95%CI
```

```
predict fitted, pr
roctab chd69 fitted
## (12 missing values generated)
##
##
## (12 missing values generated)
##
##
## (13 observations deleted)
##
##
## Logistic regression                                     Number of obs =  3,141
##                                                         LR chi2(8)    = 198.15
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -788.01957                             Pseudo R2     = 0.1117
##
## ------------------------------------------------------------------------------
##        chd69 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##        age10 |   .5949712   .1201093     4.95   0.000     .3595612    .8303811
##       chol50 |   .5757133   .0777901     7.40   0.000     .4232475    .7281792
##        sbp50 |   1.019648   .2066016     4.94   0.000     .6147161     1.42458
##        bmi10 |    1.04884   .2998181     3.50   0.000     .4612068    1.636472
##        smoke |   .6061928   .1410534     4.30   0.000     .3297331    .8826525
##       dibpat |   .7234267   .1448997     4.99   0.000     .4394284    1.007425
##      bmichol |  -.8896929   .2746472    -3.24   0.001    -1.427992   -.3513942
##       bmisbp |  -1.503455   .6318153    -2.38   0.017     -2.74179   -.2651194
##        _cons |  -3.416062    .150472   -22.70   0.000    -3.710982   -3.121143
## ------------------------------------------------------------------------------
##
##
## Logistic model for chd69
##
## Number of observations =     3141
## Area under ROC curve   =   0.7535
##
##
##
##                       ROC                     Asymptotic normal
##            Obs       area     Std. err.      [95% conf. interval]
##      -----------------------------------------------------------
```

```
##          3,141     0.7535       0.0149        0.72432     0.78267
```

In general, the more the ROC curve is stretched towards the top left corner, the more accurate the model is. The area under the ROC curve called the AUC (or C-statistic) provides an overall measure of classification accuracy, with the value of one representing perfect accuracy. The AUC returned here is 0.754, 95% CI=(0.72 ; 0.78) which could be seen as fair. In general, the following categories are considered: 0.5 - 0.6 = no discrimination, 0.6 - 0.7 = poor; 0.7 - 0.8 = fair; 0.8 - 0.8 = good; and 0.9 - 1 = excellent.

The AUC is not the only tool that can be used to assess the performance of a model but it is by far the most common. Alternatives include the Brior score or Somer's D statistic. An important point must be stressed here: because the data have been used twice (to build the model and to assess its discriminative ability), the AUC or any other statistic are *optimistic*. They tend to return a slightly better estimate than would have been obtained with new data. This is called *optimism bias* in the literature. We will see in Week 12 that this (usually small) bias can be corrected using bootstrapping or cross-validation.

11.2 Goodness of fit

Two other checks can be found in the literature to assess the adequacy of a model. The first is a *specification* test available in Stata through the command *linktest* direcly after a model fit. This test involves fitting a second model using the right hand since (the linear predictor in the logistic model) called *_hat* and its square called *_hatsq*. You expect the first Wald test for *_hat* to be significant if your model can explain the data. If the second test corresponding *_hatsq* is significant further modelling is required. Running this command after fitting the same model examined above indicated no evidence that the quadratic term *_hatsq* is needed (p=0.146), therefore we don't have evidence of model inadequacy. The test has obvious limitations but is provided as a standard tool in Stata (a R counterpart requires a few lines of code). Another tool is to perform a goodness of fit (GoF) test due to Hosmer and Lemershow. The test works by forming groups of ordered predicted probabilities and comparing observed vs expected frequencies. It's not clear how many groups should be used although 10 is often used in practice. The test is directly available in Stata via the command *lfit, group(10) table*. You need to work a bit harder in R to get it or use the *ResourcesSelection* library.

R code and output

```
library(ResourceSelection)
## Error in library(ResourceSelection): there is no package called 'ResourceSelection'
hl<-hoslem.test(x = wcgs4$chd69, y = fitted(out6), g = 10)
## Error in hoslem.test(x = wcgs4$chd69, y = fitted(out6), g = 10): could not find functio
```

```r
hl
## Error in eval(expr, envir, enclos): object 'hl' not found
# syntax: x = outcome, y=predicted values, g = number of groups
# observed vs expected can also be listed
cbind(hl$observed,hl$expected)
## Error in eval(expr, envir, enclos): object 'hl' not found
hl1<-hoslem.test(x = wcgs4$chd69, y = fitted(out6), g = 20)
## Error in hoslem.test(x = wcgs4$chd69, y = fitted(out6), g = 20): could not find functio
hl1
## Error in eval(expr, envir, enclos): object 'hl1' not found
# link test
wcgs4$linpred<-predict(out6,type="link")
wcgs4$linpred2=wcgs4$linpred^2
out.link<-glm(chd69 ~ linpred + linpred2, family=binomial, data=wcgs4)
summary(out.link)
##
## Call:
## glm(formula = chd69 ~ linpred + linpred2, family = binomial,
##     data = wcgs4)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.39838    0.32305  -1.233    0.218
## linpred      0.56468    0.30606   1.845    0.065 .
## linpred2    -0.10024    0.06889  -1.455    0.146
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1774.2  on 3140  degrees of freedom
## Residual deviance: 1573.8  on 3138  degrees of freedom
## AIC: 1579.8
##
## Number of Fisher Scoring iterations: 7
```

Stata code and output

```
use wcgs.dta
gen age10=(age-46.27869)/10
gen bmi10=(bmi-24.51837)/10
```

```
gen chol50=(chol-226.3724)/50
gen sbp50=(sbp-128.6328)/50
gen bmichol=bmi10*chol50
gen bmisbp=bmi10*sbp50
** remove missing cholesterol values and outlier (chol=645)
drop if chol>=645
** n=3141 observations
logistic chd69 age10 chol50 sbp50 bmi10 smoke dibpat  bmichol bmisbp, coef
** GoF test and link test
linktest, nolog
lfit, group(10) table
lfit, group(20)
## (12 missing values generated)
##
##
## (12 missing values generated)
##
##
## (13 observations deleted)
##
##
## Logistic regression                                     Number of obs =  3,141
##                                                         LR chi2(8)    = 198.15
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -788.01957                             Pseudo R2     = 0.1117
##
## ------------------------------------------------------------------------------
##        chd69 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##        age10 |   .5949712   .1201093     4.95   0.000     .3595612    .8303811
##       chol50 |   .5757133   .0777901     7.40   0.000     .4232475    .7281792
##        sbp50 |   1.019648   .2066016     4.94   0.000     .6147161     1.42458
##        bmi10 |    1.04884   .2998181     3.50   0.000     .4612068    1.636472
##        smoke |   .6061928   .1410534     4.30   0.000     .3297331    .8826525
##       dibpat |   .7234267   .1448997     4.99   0.000     .4394284    1.007425
##      bmichol |  -.8896929   .2746472    -3.24   0.001    -1.427992   -.3513942
##       bmisbp |  -1.503455   .6318153    -2.38   0.017     -2.74179   -.2651194
##        _cons |  -3.416062    .150472   -22.70   0.000    -3.710982   -3.121143
## ------------------------------------------------------------------------------
##
##
```

```
## Logistic regression                                      Number of obs =  3,141
##                                                          LR chi2(2)    = 200.40
##                                                          Prob > chi2   = 0.0000
## Log likelihood = -786.89257                              Pseudo R2     = 0.1130
##
## ------------------------------------------------------------------------------
##        chd69 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##         _hat |   .5646789   .3060568     1.85   0.065    -.0351813    1.164539
##       _hatsq |  -.1002356   .0688903    -1.46   0.146    -.2352581     .034787
##        _cons |  -.3983751   .3230501    -1.23   0.218    -1.031542    .2347915
## ------------------------------------------------------------------------------
##
## note: obs collapsed on 10 quantiles of estimated probabilities.
##
## Goodness-of-fit test after logistic model
## Variable: chd69
##
##   Table collapsed on quantiles of estimated probabilities
##   +--------------------------------------------------------+
##   | Group |   Prob | Obs_1 | Exp_1 | Obs_0 | Exp_0 | Total |
##   |-------+--------+-------+-------+-------+-------+-------|
##   |     1 | 0.0160 |     1 |   3.3 |   314 | 311.7 |   315 |
##   |     2 | 0.0251 |     6 |   6.5 |   308 | 307.5 |   314 |
##   |     3 | 0.0344 |    11 |   9.3 |   303 | 304.7 |   314 |
##   |     4 | 0.0450 |    12 |  12.5 |   302 | 301.5 |   314 |
##   |     5 | 0.0575 |    18 |  16.0 |   296 | 298.0 |   314 |
##   |-------+--------+-------+-------+-------+-------+-------|
##   |     6 | 0.0728 |    10 |  20.4 |   304 | 293.6 |   314 |
##   |     7 | 0.0963 |    28 |  26.5 |   286 | 287.5 |   314 |
##   |     8 | 0.1268 |    44 |  34.7 |   270 | 279.3 |   314 |
##   |     9 | 0.1791 |    50 |  46.7 |   264 | 267.3 |   314 |
##   |    10 | 0.5996 |    76 |  80.3 |   238 | 233.7 |   314 |
##   +--------------------------------------------------------+
##
##  Number of observations =  3,141
##        Number of groups =     10
## Hosmer-Lemeshow chi2(8) =  11.36
##             Prob > chi2 = 0.1824
##
## note: obs collapsed on 20 quantiles of estimated probabilities.
##
```

```
## Goodness-of-fit test after logistic model
## Variable: chd69
##
##   Number of observations =  3,141
##         Number of groups =     20
## Hosmer-Lemeshow chi2(18) =  19.67
##              Prob > chi2 = 0.3517
```

There is no evidence of violation of the goodness of fit assumptions using this procedure ($p > 0.18$ with either 10 or 20 groups). This GoF test is presented here for the sake of completeness. It's not always a reliable test and it should be used with caution. In particular, because power of GoF tests increases with sample size, practically irrelevant discrepancies between observed and expected number of events are increasingly likely to cause the rejection of the hypothesis of a good fit in large samples. Calibration plots displaying observed vs predicted probabilities (see week 12() can be used to get a visual impression of how well the model fit. They constitute a reasonable alternative but can also be affected by optimism bias as well.

Summary

The following are the key takeaway messages from this week:

1. Most diagnostic tools developed for the linear model can be extended to logistic regression
2. Restricted cubic splines or fractional polynomials provide a way to assess linearity.
3. AIC can be used to compare non-nested models and decide which model to keep.
4. ROC curves are a very convenient way to assess the discriminative ability of a model but we need to be aware of optimism bias
5. Goodness of fit tests are available for binary logistic regression but should only be used with caution.

12 model building for binary outcomes

Learning objectives

1. Revisit the fundamental difference between two aspects of modelling: prediction vs explanation
2. Be familiar with the concept of training and validation datasets
3. Discover the different ways to validate a prediction model, particularly in logistic regression
4. Learn about optimism bias and how it can be removed
5. Discover how calibration plots are built and why they are important
6. Revisit variety of ways in which multiple logistic regression models are constructed when the emphasis is on interpretation

Learning activities

This week's learning activities include:

Learning Activity	Learning objectives
Lecture 1	1, 2, 3, 4
Readings	2, 3, 4, 5
Lecture 2	5, 6
Investigation	3, 4
Practice	5

We have learnt tools to build a linear or logistic regression model. In this last week of RM1, we take a higher perspective and consider model building strategies *in general*. Perhaps, in preamble, it is worth reminding that: *Successful modeling of a complex data set is part science, part statistical methods, and part experience and common sense.* The quote is due to Hosmer and Lemershow (2013) in their book on applied logistic regression but applies to any model. This issue of model building has attracted attention and a lot of controversy over the years

and no doubt the discussion will keep going in the foreseeable future. Still, we think it is important for you to know key ideas before you decide on your own strategy and formulate your personal view on this important topic. In this week materials we revisit and expand on what was discussed in the context of linear regression. The first key concept - and probably most statisticians would agree on this - is to make the distinction between prediction and explanation/interpretation. This has clear implications in terms of modelling and that is why this concept is so important.

Result explaining the fundamental difference between explanatory and predictive modelling

In prediction, we are typically interested in predicting an outcome y with some function of x, say, $f(x)$ where x is the vector of covariates. To clarify what we mean by *predicting*, let us say that we would like $f(x)$ to be close to y in some sense. A common way to define close is to consider the squared (expected) error loss of estimating y using $f(x)$, i.e. $E[(y-f(x))^2]$. It then makes sense to minimise this quantity mathematically yielding $f = E(y|x)$, the conditional expectation of y given x (which is called a regression function). Now we have data (x_i, y_i), $i = 1, \dots, n$ to play with and our goal becomes to find $\hat{f}$ that is a good estimate of the regression function f. For instance, $\hat{f}$ can be the fitted model at a particular vector of covariates x. The model is choosen to fit the data well. How good is that estimate $\hat{f}$? A good $\hat{f}$ should have a low expected prediction error:

$$EPE(y, \hat{f}(x)) = E\Big[(y - \hat{f}(x))^2]$$

This expectation is over x, y and also the sampled data used to build $\hat{f}$. Next, we can rewrite EPE as follows:

‘

$$EPE = \text{var}(y) + \text{bias}^2(\hat{f}(x)) + \text{var}(\hat{f}(x))$$

In other words, EPE can be expressed as the sum of 3 terms

$$EPE = \text{irreducible error} + \text{bias}^2 + \text{variance}$$

where the first term is the irreducible error that results even is the model is correctly specified and accurarely estimated; the second term linked to bias is the result of misspecifying the statistical model f; the third term called (estimation) variance is the result of using a sample to estimate f. The above decomposition reveals a source of the difference between explanatory and predictive modeling: In explanatory modeling the focus is on minimising bias to obtain the most accurate representation. In contrast, predictive modeling seeks to minimise the

combination of bias and estimation variance, occasionally sacrificing theoretical accuracy for improved empirical precision. This is essentially a bias-variance tradeoff. A *wrong* model can sometimes predict better than the correct one! We have used here the EPE to quantify the closeness of $\hat{f}$ but we could use other measures of prediction error. In binary logistic regression, we may use tools adapted to the nature of the data but the fundamental distinction between prediction and explanation remains.

12.1 How to build a prediction model

This reading lists seven steps that need to be undertaken to build a (risk) prediction model that is often the objective of the investigation in medical statistics. We will not redicuss all the steps but focus on steps 5 and 6 that are essential and have not been fully discussed so far.

12.1.0.1 [@Steyerberg2014] Steyerberg and Vergouwe (2014). Towards better clinical prediction models: seven steps for development and an ABCD for validation {#reading_wk12_Steyerberg .unnumbered}

This lecture discusses the issue of optimism bias when evaluating the AUC, the notion of training and validation dataset, crossvalidation and how to produce an AUC free of optimism bias.

Lecture 1 in R

Download video here

Lecture 1 in Stata

Download video here

Optimism, training and validation datasets

First we need a good measure of prediction error for a binary endpoint. The ROC curve or more exactly the AUC introduced in week 11 could be used since the closer to 1 the AUC is, the better the model discriminates and predict adequately the 0's and the 1's. Another common measure is the Brier score that directly measures the prediction error. Since the AUC is very popular we will focus on the AUC but everything we say next will be true for The Brier score or other prediction error measures. The first point to consider is how to we evaluate the AUC? A naive way to proceed is to build a model using possibly steps 1-4 of the reading and then evaluate the AUC using the *same* data. The problem with this approach is the data is

used twice (once to build the model and a second time to evaluate the AUC). This results in an overestimation of the AUC (on average), the problem is known as *optimism bias* as briefly stated in week 11.

Bias can be severe when large models are used and tend to overfit the idiosyncrasies of the data and, as a result, will poorly predict new, independent observations. This issue of overfitting is well known in machine learning particularly when using neural networks that are prone to overfitting. The issue is the same with complex statistical models. A way to overcome this avoid the optimism bias is to randomly split the dataset in two: the first dataset is for training and is therefore called the training (or development) dataset; the 2nd data set is for validation therefore its name: validation (or test) dataset. This is known as the *split sample validation* approach. It aims at addressing the stability of the selection of predictors and the quality of predictions using a random sample for model development and the remaining patients for validation. The logic is rather simple: 1) generate a random variable that splits the data in two (i.e. the equivalent of a coin toss to decide to which dataset each patient) will belong - call this indicator *val* ; 2) fit the model on the development dataset (*val=0*); 3) evaluate its performance (e.g. AUC) on the validation dataset (*val=1*).

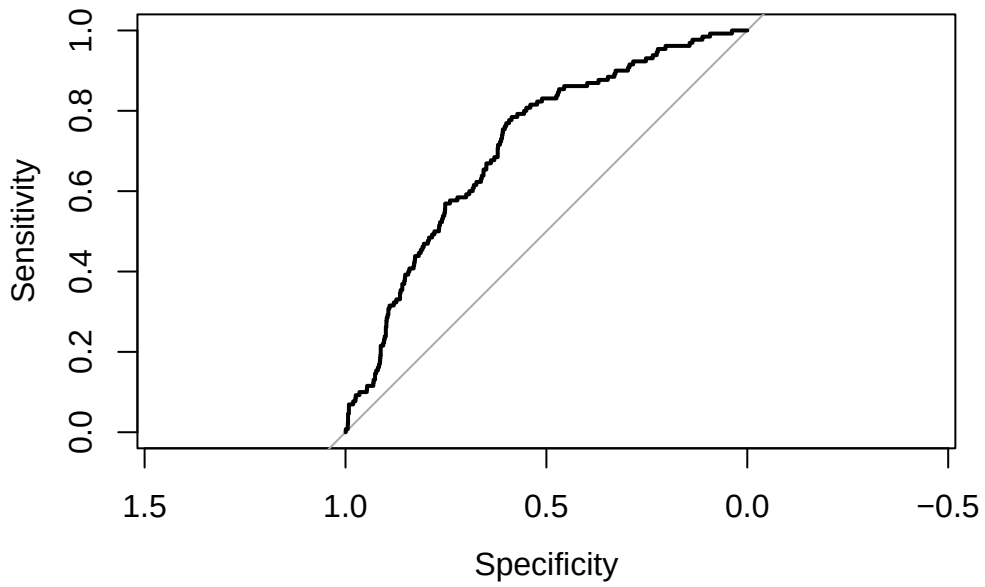
To illustrate, we again use the WGCS data and reproduce the analysis presented in Vittinghof et al. (2012) p. 400 but delete the outlier in cholesterol (n=3141 observations). The endpoint is once again *chd69* and the covariates retained for this exercise *age*, *chol*, *sbp*, *bmi* and *smoke*. The AUC is 0.713, 95% CI=(0.67 ; 0.76) in R up to rounding, a slightly lower value and wider CI than what is observed on all patients (n=3141), AUC=0.732, 95% CI=(0.70 ; 0.76)and in the development dataset. A slightly different value may be obtained in Stata (AUC=0.72, 95%=(0.68 ; 0.76) due to the seed choice and a possibly different random number generator.

R code and output

```
wcgs <- read.csv("wcgs.csv")
wcgs<-data.frame(wcgs)
wcgs1=cbind(wcgs$age,wcgs$chol,wcgs$sbp,wcgs$bmi,wcgs$smoke,wcgs$chd69)
colnames(wcgs1)=c("age", "chol", "sbp", "bmi", "smoke","chd69")
wcgs1=data.frame(wcgs1)
wcgs1=na.omit(wcgs1)
wcgs1<-wcgs1[wcgs1$chol < 645,]
# remove outlier in cholesterol
n=dim(wcgs1)[1]
set.seed(1001)
wcgs1$val=rbinom(n,1,0.5) # val =0 for development and val=1 for validation
table(wcgs1$val)
##
##    0    1
## 1596 1545
```

```r
# building model on development dataset
wcgs1.dev=wcgs1[wcgs1$val==0,]
fit.dev<-glm(chd69 ~ age+chol+sbp+bmi+smoke, family=binomial, data=wcgs1.dev)
summary(fit.dev)
##
## Call:
## glm(formula = chd69 ~ age + chol + sbp + bmi + smoke, family = binomial,
##     data = wcgs1.dev)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -13.016252   1.397805  -9.312  < 2e-16 ***
## age           0.063685   0.017046   3.736 0.000187 ***
## chol          0.013148   0.002172   6.054 1.42e-09 ***
## sbp           0.019793   0.005747   3.444 0.000573 ***
## bmi           0.060258   0.036968   1.630 0.103102
## smoke         0.602919   0.203779   2.959 0.003090 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 881.60  on 1595  degrees of freedom
## Residual deviance: 786.35  on 1590  degrees of freedom
## AIC: 798.35
##
## Number of Fisher Scoring iterations: 6
# evaluation on the validation dataset
wcgs1.val=wcgs1[wcgs1$val==1,]
wcgs1.val$pred<- predict(fit.dev, wcgs1.val,type="response")
# predicted probabilities for the validation dataset using the previous fit
require(pROC)
## Loading required package: pROC
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
# ROC + AUC on the validation dataset (suffix .val)
out<- roc(wcgs1.val$chd69, wcgs1.val$pred,plot=TRUE,ci=TRUE)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```




```
out
##
## Call:
## roc.default(response = wcgs1.val$chd69, predictor = wcgs1.val$pred,     ci = TRUE, plot
##
## Data: wcgs1.val$pred in 1415 controls (wcgs1.val$chd69 0) < 130 cases (wcgs1.val$chd69
## Area under the curve: 0.7127
## 95% CI: 0.6694-0.756 (DeLong)
```

Stata code and output

```
use wcgs.dta
drop if chol>=645
** missing chol and chol=645 deleted
set seed 1001
gen val = runiform()<.5
** Derive a prediction model for y-chd69 using the development dataset
logistic chd69 age chol sbp bmi smoke if val==0
```

```
**Generate a new variable containing the predicted probabilities for all observations
predict fitted, pr
** AUC on the development data (training), n=1578; to compare to.
roctab chd69 fitted if  val==0
** AUC on the validation data  - n= 1563; the one we need!
roctab chd69 fitted if  val==1
roctab chd69 fitted if  val==1,  graph title("Validation")
## (13 observations deleted)
##
##
##
##
## Logistic regression                                     Number of obs =  1,578
##                                                         LR chi2(5)    =  86.80
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -395.91089                             Pseudo R2     = 0.0988
##
## ------------------------------------------------------------------------------
##        chd69 | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##          age |   1.045713   .0178468     2.62   0.009     1.011313    1.081284
##         chol |   1.013229   .0021723     6.13   0.000      1.00898    1.017496
##          sbp |   1.020727   .0062837     3.33   0.001     1.008485    1.033117
##          bmi |   1.046858   .0403416     1.19   0.235     .9707017    1.128988
##        smoke |   2.034663   .4094141     3.53   0.000     1.371558    3.018359
##        _cons |   6.91e-06   9.78e-06    -8.39   0.000     4.30e-07    .0001109
## ------------------------------------------------------------------------------
## Note: _cons estimates baseline odds.
##
##
##
##                       ROC                    Asymptotic normal
##            Obs       area     Std. err.      [95% conf. interval]
##      ------------------------------------------------------------
##          1,578     0.7385       0.0224        0.69452     0.78241
##
##
##                       ROC                    Asymptotic normal
##            Obs       area     Std. err.      [95% conf. interval]
##      ------------------------------------------------------------
##          1,563     0.7195       0.0219        0.67657     0.76242
```

We let you change the seed or the proportion of patients in each dataset and check that you will get slightly different results. Although commonly used, the split sample validation approach is a suboptimal form of internal validation.

Different forms of validation

A more efficient way of splitting the data is to use *h-fold cross validation* where the whole dataset is used for both steps. We illustrate the procedure using $h = 10$, often considered as the default but other values can be used (typically between 10 and 20).

(1) Randomly divide the data into $h = 10$ mutually exclusive subsets of approximately the same size.\

(2) In turn, set aside each subset of the data (approximately 10%) and use the remaining other subsets (approximately 90%) to fit the model.

(3) Use the parameter estimates to get the summary statistics needed to predict the AUC (or any other measure of prediction error), for the subset of the observations that were set aside. In practice, we typically compute the predicted probabilities for the 10% of the data that we did not use to estimate the model.

(4) Repeat this for all 10 subsets and compute a summary estimate of the AUC (or any other measure of prediction error)

In addition, this 4 step procedure can be repeated k times, and the average AUC computed but for simplicity we will consider the case $k = 1$.

Going back to the WCGS data and model used previously, the naive estimate for the AUC (based on 3141 observations) is 0.732. 95%CI=(0.702 ; 0.763). A 10-fold cross-validated AUC computed through steps 1)-4) is 0.727, 95% CI=(0.697 ; 0.757) in R. Note that the 95% cI is shorter than the one obtained with the split sample approach since all the data has been used.

R code and output

```
require(cvAUC)
## Loading required package: cvAUC
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'cvAUC'
## function required to run CV
cv_eval <- function(data, V=10){
  f.cvFolds <- function(Y, V){ #Create CV folds (stratify by outcome)
    Y0 <- split(sample(which(Y==0)), rep(1:V, length=length(which(Y==0))))
    Y1 <- split(sample(which(Y==1)), rep(1:V, length=length(which(Y==1))))
```

```
    folds <- vector("list", length=V)
    for (v in seq(V)) {folds[[v]] <- c(Y0[[v]], Y1[[v]])}
    return(folds)
  }
  f.doFit <- function(v, folds, data){ #Train/test glm for each fold
    fit <- glm(Y~., data=data[-folds[[v]],], family=binomial)
    pred <- predict(fit, newdata=data[folds[[v]],], type="response")
    return(pred)
  }
  folds <- f.cvFolds(Y=data$Y, V=V) #Create folds
  predictions <- unlist(sapply(seq(V), f.doFit, folds=folds, data=data)) #CV train/predict
  predictions[unlist(folds)] <- predictions #Re-order pred values
  ci.pooled.cvAUC
  # Get CV AUC and confidence interval
  out <- ci.cvAUC(predictions=predictions, labels=data$Y, folds=folds, confidence=0.95)
  return(out)
}

# this fonction requires to have a dataset with only the covariates
# and the endpoint recoded Y as columns, NOTE that the outcome (last
# column) is now named Y)

wcgs1=wcgs1[,1:6]
colnames(wcgs1)=c("age", "chol", "sbp", "bmi", "smoke","Y")
fit<-glm(Y ~ age+chol+sbp+bmi+smoke, family=binomial, data=wcgs1)
set.seed(1001)
out <- cv_eval(data=wcgs1, V=10)
## Error in cv_eval(data = wcgs1, V = 10): object 'ci.pooled.cvAUC' not found
out
##
## Call:
## roc.default(response = wcgs1.val$chd69, predictor = wcgs1.val$pred,     ci = TRUE, plot
##
## Data: wcgs1.val$pred in 1415 controls (wcgs1.val$chd69 0) < 130 cases (wcgs1.val$chd69
## Area under the curve: 0.7127
## 95% CI: 0.6694-0.756 (DeLong)
```

As expected the naive AUC estimate is slightly bigger than its cross-validated counterpart. This illustrates a small optimism bias discussed above. The fact that the difference is small suggests that the model used to predict *chd69* is not badly overfitted. Once again, the result depends on the seed and may differ (slighytly) across packages. In this instance, we get a cross-validated AUC=0.727, 95% CI=(0.696 ; 0.758) in Stata with the same seed. The code

below shows how to do this.

Stata code and output

```
clear
use wcgs.dta
drop if chol>=645
** missing chol and chol=645 deleted
set seed 1001
xtile group = uniform(), nq(10)
quietly gen cvfitted = .
forvalues i = 1/10 {

    ** Step 2: estimate model omitting each subset
    quietly xi: logistic chd69 age chol sbp bmi smoke if group~=`i'
    quietly predict cvfittedi, pr

    ** Step 3: save cross-validated statistic for each omitted subset
    quietly replace cvfitted = cvfittedi if group==`i'
    quietly drop cvfittedi
    }

** Step 4: calculate cross-validated area under ROC curve
roctab chd69 cvfitted
## (13 observations deleted)
##
##
##
##
##
##
##                        ROC                    Asymptotic normal
##            Obs       area     Std. err.      [95% conf. interval]
##      -------------------------------------------------------------
##          3,141     0.7270       0.0157        0.69619     0.75779
```

General crossvalidation is not the only technique that can be used to correct for optimism bias, the bootstrap is also a popular method. We do not describe in detail how it can be used in this context but interested readers can look here (optional)

link

So far we have only talked about internal validation, either using the split sample approach or cross-validation. This word "internal'' is used because we only use the dataset under

investigation. Another form of validation called *external validation* exists and is generally considered superior. In that case, data from a different source are used to construct the ROC curve (and compute the AUC). This form of validation addresses transportability rather than reproducibility and constitutes a stronger test for prediction models.

Investigation: \

1) The *infarct* data provides information on 200 female patients who experienced MI. The response is *mi* (0=no,1=yes), *age* in years, *smoke* (0=no,1=yes) and *oral* for oral contraceptive use (0=no,1=yes). You can check that all covariates are important predictors. We will assume that the age effect is rather linear - you can also check that it seems a reasonable assumption for these data. Give the the naive AUC, its 95% CI and draw the ROC curve. Does the model seem to predict well *mi*?
2) We have used the same data to build and evaluate the model, so the AUC may be affected by optimism bias. Use the split sample validation approach to compute the AUC. Do you think this is a good approach to use here?
3) Build a 10-fold cross-validated AUC and compare with the naive estimate. What do you conclude?

Of course, the concept of crossvalidation illustrated here in logistic regression is general and can be used in linear regression as well using an appropriate measure of prediction error.

Calibration

This lecture discusses how calibration plots can be obtained and why they are important in risk prediction modelling.

Lecture 2 in R

Download video here

Lecture 2 in Stata

Download video here

Evaluation of a model performance does not stop with the computation of an external or cross-validated AUC and the evaluation of the model discriminative ability. A quality prediction model should also provide a good agreement between observed endpoints and predictions over the whole range of predicted values. In the WCGS example, it may well be that the model correctly predicts the outcome for patients with low risk of CHD but is not so accurate for high risk patients. It is important as poorly calibrated risk estimates can lead to false expectations with patients and healthcare professionals. The term *calibration* is used to describe the agreement between observed endpoints and predictions. A calibration plot, displaying observed vs predicted probabilities with 95% CI, is often used to get a visual impression on

how close they are to the 45 degrees line, seen as a proof of good calibration. Calibration plots can also be affected by overfitting and optimism bias so, again, plots based on external data are preferable. If no external data is available the *split sample approach* can be used. It may be possible to use cross-validation but it is more complicated since you would have to save predicted probabilities evaluared on the 10% of the data left aside, repeat the procedure, for all other 9 different datasets and draw a single calibration plot at the end. To illustrate the concept, we go back to the WCGS example investigated earlier. The following plot is obtained when data are grouped in 10 categories of risk and split as before in training/validation datasets. We could choose more categories but 10 is often the default choice. A larger number of categories is possible for large dataset, we need to keep in mind that the more categories we choose, the wider the confidence intervals.

‘

```
require(rms)
```

```
Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
logical.return = TRUE, : there is no package called 'rms'
```

```
# val.prob(wcgs1.val$pred, wcgs1.val$chd69, m=150, cex=.5) # without CIs

# calibration function to get the 95% CI
val.prob.ci<-
  function(p, y, logit, group, weights = rep(1, length(y)), normwt = F, pl = T,
           smooth = T, logistic.cal = F, xlab = "Predicted probability", ylab =
             "Observed frequency", xlim = c(-0.02, 1),ylim = c(-0.15,1), m, g, cuts, emax.
           legendloc =  c(0.55 , 0.27), statloc = c(0,.85),dostats=c(12,13,2,15,3),roundst
           riskdist = "predicted", cex = 0.75, mkh = 0.02, connect.group =
             F, connect.smooth = T, g.group = 4, evaluate = 100, nmin = 0, d0lab="0", d1la
           dist.label=0.04, line.bins=-.05, dist.label2=.03, cutoff, cex.lab=1, las=1, len
    if(missing(p))
      p <- 1/(1 + exp( - logit))
    else logit <- log(p/(1 - p))
    if(length(p) != length(y))
      stop("lengths of p or logit and y do not agree")
    names(p) <- names(y) <- names(logit) <- NULL
    if(!missing(group)) {
      if(length(group) == 1 && is.logical(group) && group)
        group <- rep("", length(y))
      if(!is.factor(group))
        group <- if(is.logical(group) || is.character(group))
```

```
      as.factor(group) else cut2(group, g =
                                 g.group)
  names(group) <- NULL
  nma <- !(is.na(p + y + weights) | is.na(group))
  ng <- length(levels(group))
}
else {
  nma <- !is.na(p + y + weights)
  ng <- 0
}
logit <- logit[nma]
y <- y[nma]
p <- p[nma]
if(ng > 0) {
  group <- group[nma]
  weights <- weights[nma]
  return(val.probg(p, y, group, evaluate, weights, normwt, nmin)
  )
}
if(length(unique(p)) == 1) {
  #22Sep94
  P <- mean(y)
  Intc <- log(P/(1 - P))
  n <- length(y)
  D <- -1/n
  L01 <- -2 * sum(y * logit - log(1 + exp(logit)), na.rm = T)
  L.cal <- -2 * sum(y * Intc - log(1 + exp(Intc)), na.rm = T)
  U.chisq <- L01 - L.cal
  U.p <- 1 - pchisq(U.chisq, 1)
  U <- (U.chisq - 1)/n
  Q <- D - U

  stats <- c(0, 0.5, 0, D, 0, 1, U, U.chisq, U.p, Q, mean((y - p[
    1])^2), Intc, 0, rep(abs(p[1] - P), 2))
  names(stats) <- c("Dxy", "C (ROC)", "R2", "D", "D:Chi-sq",
                    "D:p", "U", "U:Chi-sq", "U:p", "Q", "Brier",
                    "Intercept", "Slope", "Emax", "Eavg")
  return(stats)
}
i <- !is.infinite(logit)
nm <- sum(!i)
```

```
if(nm > 0)
  warning(paste(nm,
                "observations deleted from logistic calibration due to probs. of 0 or
  ))
f <- lrm.fit(logit[i], y[i])
f2<-    lrm.fit(offset=logit[i], y=y[i])
stats <- f$stats
n <- stats["Obs"]
predprob <- seq(emax.lim[1], emax.lim[2], by = 0.0005)
lt <- f$coef[1] + f$coef[2] * log(predprob/(1 - predprob))
calp <- 1/(1 + exp( - lt))
emax <- max(abs(predprob - calp))
if (pl) {
  plot(0.5, 0.5, xlim = xlim, ylim = ylim, type = "n", xlab = xlab,
       ylab = ylab, las=las)
  abline(0, 1, lty = 2)
  lt <- 2
  leg <- "Ideal"
  marks <- -1
  if (logistic.cal) {
    lt <- c(lt, 1)
    leg <- c(leg, "Logistic calibration")
    marks <- c(marks, -1)
  }
  if (smooth) {
    Sm <- lowess(p, y, iter = 0)
    if (connect.smooth) {
      lines(Sm, lty = 3)
      lt <- c(lt, 3)
      marks <- c(marks, -1)
    }
    else {
      points(Sm)
      lt <- c(lt, 0)
      marks <- c(marks, 1)
    }
    leg <- c(leg, "Nonparametric")
    cal.smooth <- approx(Sm, xout = p)$y
    eavg <- mean(abs(p - cal.smooth))
  }
  if(!missing(m) | !missing(g) | !missing(cuts)) {
```

```
    if(!missing(m))
      q <- cut2(p, m = m, levels.mean = T, digits = 7)
    else if(!missing(g))
      q <- cut2(p, g = g, levels.mean = T, digits = 7)
    else if(!missing(cuts))
      q <- cut2(p, cuts = cuts, levels.mean = T, digits = 7)
    means <- as.single(levels(q))
    prop <- tapply(y, q, function(x)mean(x, na.rm = T))
    points(means, prop, pch = 2, cex=cex)
    #18.11.02: CI triangles
    ng  <-tapply(y, q, length)
    og  <-tapply(y, q, sum)
    ob  <-og/ng
    se.ob   <-sqrt(ob*(1-ob)/ng)
    g       <- length(as.single(levels(q)))

    for (i in 1:g) lines(c(means[i], means[i]), c(prop[i],min(1,prop[i]+1.96*se.ob[i]))
    for (i in 1:g) lines(c(means[i], means[i]), c(prop[i],max(0,prop[i]-1.96*se.ob[i]))

    if(connect.group) {
      lines(means, prop)
      lt <- c(lt, 1)
    }
    else lt <- c(lt, 0)
    leg <- c(leg, "Grouped patients")
    marks <- c(marks, 2)
  }
}
lr <- stats["Model L.R."]
p.lr <- stats["P"]
D <- (lr - 1)/n
L01 <- -2 * sum(y * logit - logb(1 + exp(logit)), na.rm = TRUE)
U.chisq <- L01 - f$deviance[2]
p.U <- 1 - pchisq(U.chisq, 2)
U <- (U.chisq - 2)/n
Q <- D - U
Dxy <- stats["Dxy"]
C <- stats["C"]
R2 <- stats["R2"]
B <- sum((p - y)^2)/n
# ES 15dec08 add Brier scaled
```

```
Bmax  <- mean(y) * (1-mean(y))^2 + (1-mean(y)) * mean(y)^2
Bscaled <- 1 - B/Bmax
stats <- c(Dxy, C, R2, D, lr, p.lr, U, U.chisq, p.U, Q, B,
           f2$coef[1], f$coef[2], emax, Bscaled)
names(stats) <- c("Dxy", "C (ROC)", "R2", "D", "D:Chi-sq",
                  "D:p", "U", "U:Chi-sq", "U:p", "Q", "Brier", "Intercept",
                  "Slope", "Emax", "Brier scaled")
if(smooth)
  stats <- c(stats, c(Eavg = eavg))

# Cut off definition
if(!missing(cutoff)) {
  arrows(x0=cutoff,y0=.1,x1=cutoff,y1=-0.025,length=.15)
}
if(pl) {
  logit <- seq(-7, 7, length = 200)
  prob <- 1/(1 + exp( - logit))
  pred.prob <- f$coef[1] + f$coef[2] * logit
  pred.prob <- 1/(1 + exp( - pred.prob))
  if(logistic.cal) lines(prob, pred.prob, lty = 1)
  # pc <- rep(" ", length(lt))
  # pc[lt==0] <- "."
  lp <- legendloc
  if (!is.logical(lp)) {
    if (!is.list(lp))
      lp <- list(x = lp[1], y = lp[2])
    legend(lp, leg, lty = lt, pch = marks, cex = cex, bty = "n")
  }
  if(!is.logical(statloc)) {
    dostats <- dostats
    leg <- format(names(stats)[dostats])    #constant length
    leg <- paste(leg, ":", format(stats[dostats], digits=roundstats), sep =
                 "")
    if(!is.list(statloc))
      statloc <- list(x = statloc[1], y = statloc[2])
    text(statloc, paste(format(names(stats[dostats])),
                        collapse = "\n"), adj = 0, cex = cex)
    text(statloc$x + (xlim[2]-xlim[1])/3 , statloc$y, paste(
      format(round(stats[dostats], digits=roundstats)), collapse =
        "\n"), adj = 1, cex = cex)
  }
```

```
      if(is.character(riskdist)) {
        if(riskdist == "calibrated") {
          x <- f$coef[1] + f$coef[2] * log(p/(1 - p))
          x <- 1/(1 + exp( - x))
          x[p == 0] <- 0
          x[p == 1] <- 1
        }
        else x <- p
        bins <- seq(0, min(1,max(xlim)), length = 101)
        x <- x[x >= 0 & x <= 1]
        #08.04.01,yvon: distribution of predicted prob according to outcome
        f0  <-table(cut(x[y==0],bins))
        f1  <-table(cut(x[y==1],bins))
        j0  <-f0 > 0
        j1  <-f1 > 0
        bins0 <-(bins[-101])[j0]
        bins1 <-(bins[-101])[j1]
        f0  <-f0[j0]
        f1  <-f1[j1]
        maxf <-max(f0,f1)
        f0  <-(0.1*f0)/maxf
        f1  <-(0.1*f1)/maxf
        segments(bins1,line.bins,bins1,length.seg*f1+line.bins)
        segments(bins0,line.bins,bins0,length.seg*-f0+line.bins)
        lines(c(min(bins0,bins1)-0.01,max(bins0,bins1)+0.01),c(line.bins,line.bins))
        text(max(bins0,bins1)+dist.label,line.bins+dist.label2,d1lab,cex=cex.d01)
        text(max(bins0,bins1)+dist.label,line.bins-dist.label2,d0lab,cex=cex.d01)
      }
    }
    stats
  }

# calibration plot - with 95% CIs - using the validation dataset
val.prob.ci(wcgs1.val$pred,wcgs1.val$chd69, pl=T,smooth=T,logistic.cal=F,
            g=10)
```

```
Error in lrm.fit(logit[i], y[i]): could not find function "lrm.fit"
```

Stata code (figures will be displayed when running the code)

```
use wcgs.dta
drop if chol > 645
set seed 1001
gen val = runiform()<.5
logistic chd69 age chol sbp bmi smoke if val==0
predict proba, pr

** Use pmcalplot to produce a calibration plot of the apparent performance of
** the model in the Training dataset

pmcalplot proba chd69 if val==0, ci

** Now produce the calibration plot for the models performance in the
** validation dataset = the one we need

pmcalplot proba chd69 if val==1, ci
** highest category poorly fitted

** Now produce the calibration plot for the models performance in the
** validation cohort using risk thresholds of <5%, 5-15%, 15-20% and >20%.
** This is useful if the model has been proposed for decision making at these
** thresholds.  Use the keep option to return the groupings to your dataset

pmcalplot proba chd69 if val==1, cut(.05 0.10 .15 .20) ci keep

## (12 observations deleted)
##
##
##
##
## Logistic regression                                     Number of obs =  1,579
##                                                         LR chi2(5)    = 100.20
##                                                         Prob > chi2   = 0.0000
## Log likelihood = -413.32347                             Pseudo R2     = 0.1081
##
## ------------------------------------------------------------------------------
##        chd69 | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##          age |   1.065177   .0175377     3.83   0.000     1.031353    1.100111
##         chol |   1.012124   .0020439     5.97   0.000     1.008126    1.016138
##          sbp |   1.019312   .0064403     3.03   0.002     1.006767    1.032014
```

```
##          bmi |   1.047462   .0397636     1.22   0.222     .9723557     1.12837
##        smoke |   2.443136   .4891536     4.46   0.000     1.650152     3.61719
##        _cons |   4.26e-06   5.96e-06    -8.83   0.000     2.74e-07    .0000663
## ------------------------------------------------------------------------------
## Note: _cons estimates baseline odds.
##
##
## command pmcalplot is unrecognized
## r(199);
##
## r(199);
```

The plot is reasonably close to the 45-degree line up to the highest risk category where the CHD risk tends to be overestimated but the 95% CI may just touch the 45-degree line. This plot is an internal calibration plot since only one dataset was used. We would proceed in a similar way for external calibration, were new data available, the difference being that we would use the whole WGSC dataset to fit the model and the new data to build the calibration plot. Constructing calibration plots is part of the validation process in general.

Now the next logical question is: what do we do if the calibration plot is poor? This can occur, particularly when external calibration is conducted. This can be due to the baseline risk being higher in the new sample or the two populations. This is a difficult problem to solve for which there is no unique fix. Options include: 1) recalibrate the intercept or *recalibration in the large*; 2) recalibrate the intercept and the slopes; 3) change the model to get a better plot. More details are given in the Steyerberg and Vergouwe paper given as a reading that we encourage you to read carefully.\

Optional reading: Those of you who want to know more can read the paper by Van Calster et al. (2020), Calibration: the Achilles heel of predictive analytics, BMC Medicine.\

The title of this reading says it all; the paper talks about issues related to this complex problem. We mentioned this a a supplement but the examples and plots they give are particularly illustrative. The Appendix gives the fitted models but, unfortunately, they don't give the R code.

Practice

1) We are still using the WCGS data. In week 10, we came up with a more complex model for *chd69* after deletion of the outlier (chol=645) - see Table 5.18 in Vittinghof et al. (2012). Use this model to build a calibration plot based on the split sample approach. Provide an interpretation. We use the split sample approachthe on the data at hand because we don't have external data (which would typically be the way to go for calibration).

2) Has the calibration plot improved compared with the simpler model? Keep the same seed here for a fair comparison.

3) Repeat the procedure with 20 groups using more patients in the development/training dataset (2/3 is also used as indicated in Vittinghof et al. (2012) p. 399).

Recap on building an model focusing on explanation/interpretation

We can distinguish the two cases when it comes to building a model focusing on interpretation: 1) evaluating a predictor of primary interest; 2) identifying important predictors. Generally speaking, recommendations developed in week 8 apply but, as discussed in the preamble, there is not absolute consensus on the best way to build a model. A few points are worth adding. You may also want to read the beginning of Chapter 10 (pp. 395-402) of Vittinghof et al. (2012). Regarding 1), when the study is a randomised clinical trial and therefore the focus is on assessing the effect of treatment, it's uncommon to adjust for baseline because the binary outcome is not often measured at baseline. The issue of what we should adjust for in a randomised clinical trial is controversial and our view is that you should adjust for a small number of predetermined factors listed in a statistical analysis plan. The adjusted OR is technically not the same once have adjusted for covariates. It's more a conditional than a marginal OR but in most cases the difference is tiny. In case of a non-randomised comparison between two groups, adjustment or more sophisticated methods (e.g. propensity score) are preferable. Regarding 2), model building is complex and we do not generally recommend automated model selection techniques including the backward selection procedure that has the favour of Vittinghof et al. (2012). We nevertheless agree that it's preferable to forward selection. In a long commentary paper published in 2020 on the *State of the art in selection of variables and functional forms in multivariable analysis*, the STRATOS group concluded:

Our overview revealed that there is not yet enough evidence on which to base recommendations for the selection of variables and functional forms in multivariable analysis. Such evidence may come from comparisons between alternative methods.

They also listed seven important topics for the direction of further research. Given that this group includes eminent statisticians (whose names should be familiar to you by now since many readings in this course were written by the same researchers), we will not go further and let you make your own decisions. Elements of this commentary could be useful to you but it might be better to read such an article after having completed RM1 and RM2.

Summary

The following are the key takeaway messages from this week:

1. There is a fundamental difference between models built to predict and models used to explain
2. Evaluation of the performance of a (risk) prediction model requires several steps including the assessment of its discriminative ability and calibration.
3. Optimism bias exists but can be removed using appropriate techniques (e.g. crossvalidation or bootstrap).
4. External validation is superior to internal validation but is not always possible.
5. Calibration plots assess whether the prediction model predicts well over the whole range of predicted values.
6. The general ideas developed for linear models built with interpretation as the main focus extend to binary logistic regression