Chapter 1

Statistical methods part 2

1.1 The formula object

The formula object is useful and important in estimation of linear models in R. It is described in the help as "a generic function formula and its specific methods provide a way of extracting formulae which have been included in other objects". It will be easier to understand how it is used from examples e.g. in the linear models below.

```
f1<-y~x+z # f1 is a formula

class(f1)

[1] "formula"

typeof(f1)

[1] "language"
```

1.2 Linear regression

Linear regression is a model where the observed value (dependent variable) is explained by one or more variables called independent variables (however, many names are used; linear predictors, explanatory variables, background variables or covariates). When only one predictor variable is used we talk about simle regression and with two or more predictors we call it multiple regression (sometimes multivariable regression or (incorrectly) multivariate regression)

1.2.1 Simple linear regression

The model can be written $y_i = a + bx_i + \epsilon_i$ where y_i is the random observation and x_i is the linear predictor. In the model x_i is not regarded as random. The variable ϵ_i a random error assumed to be independent between observations and have a normal distribution $N(0, \sigma)$. The parameters a and b are constants which are estimated. The interpretation is that b (slope) is the expected change of y per one unit increase of x while a (intercept) is the expected y-value given $x = 0$. The intercept is most often not of interest or relevant e.g. if y_i is the number of sold Volvo cars in Umeå a certain year x_i during 1990-2010. The intercept is then an estimate of the number of Volvo cars sold in Umeå year 0. However, the intercept is important if we are interested to use the model for prediction of the expected value on

the dependent variable. The estimation is shown below. In the example we estimate how cholesterol can be predicted by BMI in the norsjo86 data set.
We will use the norsjo86 data set which has been earlier described. We first remove some missing data.

```
library(haven)
norsjo86 <- read_sav("../data/norsjo86.sav")

summary(norsjo86) # There are some missing, especially in health

     agegrp          health            sex            height          weight
 Min.   :30.00   Min.   :0.0000   Min.   :1.000   Min.   :145.0   Min.   : 47.00
 1st Qu.:40.00   1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:165.0   1st Qu.: 64.00
 Median :40.00   Median :0.0000   Median :2.000   Median :171.0   Median : 74.00
 Mean   :45.12   Mean   :0.4046   Mean   :1.508   Mean   :170.9   Mean   : 73.89
 3rd Qu.:60.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:178.0   3rd Qu.: 81.00
 Max.   :60.00   Max.   :1.0000   Max.   :2.000   Max.   :191.0   Max.   :116.00
                 NA's   :40                       NA's   :4       NA's   :4
      sbp             dbp            kolester          smoker            bmi
 Min.   : 90.0   Min.   : 48.00   Min.   : 3.300   Min.   :0.0000   Min.   :16.71
 1st Qu.:110.0   1st Qu.: 72.00   1st Qu.: 5.400   1st Qu.:0.0000   1st Qu.:22.58
 Median :120.0   Median : 78.00   Median : 6.300   Median :0.0000   Median :24.52
 Mean   :124.5   Mean   : 79.17   Mean   : 6.536   Mean   :0.2077   Mean   :25.27
 3rd Qu.:135.0   3rd Qu.: 86.00   3rd Qu.: 7.400   3rd Qu.:0.0000   3rd Qu.:27.28
 Max.   :220.0   Max.   :120.00   Max.   :11.600   Max.   :1.0000   Max.   :45.03
 NA's   :3       NA's   :3        NA's   :3                         NA's   :4

nrow(norsjo86)

[1] 260

# remove the variable health and remove missing in the remaining variables
norsjo<-na.omit(norsjo86[,-2])

# rename kolester to chorelstorol
names(norsjo)[names(norsjo) == "kolester"] <- "cholesterol"

nrow(norsjo) # 8 observations removed

[1] 252

res<-lm(norsjo$cholesterol~norsjo$bmi) # this is one method but somewhat prolix
res<-lm(cholesterol~bmi,data=norsjo)   # this is a more convenient way

res                                 # printing the results don't give a comprehensive output

Call:
lm(formula = cholesterol ~ bmi, data = norsjo)

Coefficients:
(Intercept)          bmi
    5.27744      0.04915

class(res)

[1] "lm"

summary(res)                        # using summary gives a reasonable amount of information

Call:
lm(formula = cholesterol ~ bmi, data = norsjo)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-3.1892 -1.0369 -0.1388  0.8443  4.5600

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.27744    0.59932   8.806   <2e-16 ***
bmi          0.04915    0.02346   2.095   0.0372 *
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 1.431 on 250 degrees of freedom
Multiple R-squared:  0.01725,Adjusted R-squared:  0.01332
F-statistic: 4.387 on 1 and 250 DF,  p-value: 0.03722
```

From these results we can see the estimates of a and b , the latter called bmi although it is the coefficient for BMI. The BMI coefficient is estimated at 0.049 and the p-value is 0.037 for the test of the null hypothesis ($H_0 : b = 0$). This means that we reject the null hypothesis if we use the significance level 0.05. The conclusion is that cholesterol is affected by BMI. The estimate can be interpreted as follows: the change in cholesterol is 0.049 per unit increase in BMI.

Another information which can be of value is the multiple R-squared (coefficient of determination). It is a measure of the proportion of the total variation (of the independent variable) that is explained by the regression model.

The result object include a lot of information (check `str(res)`). Some of the information can be used for a plot.

```
#par(mfrow=c(1,2))
plot(cholesterol~bmi,data=res$model,cex=0.7) # same as plot(cholesterol~bmi,data=norsjo)
abline(res$coefficients,col=2)

abline(coef(res),col=2) # alternative
```

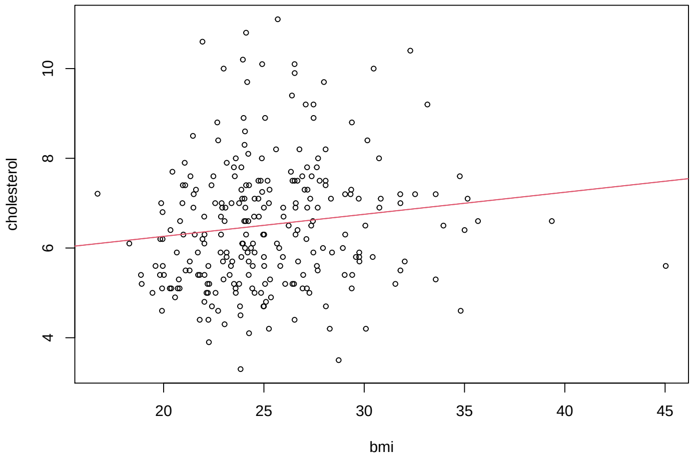
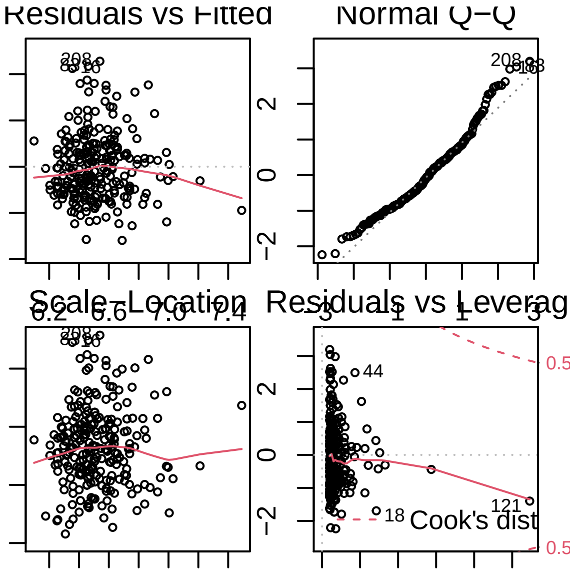


Figure 1.1: Cholesterol vs BMI and fitted model

We can even plot the result object as it is du to the generic property of plot. It shows diagnostic information about the model fit.

```
par(mfrow=c(2,2))
par(mar = c(1,1,1,1))
plot(res,cex=0.6) # plot is generic, here it gives four plots
```



```
par(mfrow=c(1,1))
```

Residuals are $y_i - \hat{y}_i$ where $\hat{y}_i$ is the estimated y_i given x_i. Thus it is the deviation of the observed value from the predicted value. Residuals are used for analysing the goodness of fit of a model. In simple linear regression we have the possibility to plot the regression line and the data and study the fit. However, in multiple regression where the possibilities of graphic illustration of y vs the independent variables are limited residuals are really useful.

1.2.2 Multiple linear regression

Multiple linear regression is similar as simple regression but include two or more predictors. One difference is that in simple regression you can plot y vs x. It is not possible with a multiple model. Assume we want to extend the model above including age. Let us first check the age variable by making a table on its values. Observe how the formula object is used.

```
table(norsjo$agegrp)

30 40 50 60
62 66 61 63

res<-lm(cholesterol~bmi+agegrp,data=norsjo)
summary(res)

Call:
lm(formula = cholesterol ~ bmi + agegrp, data = norsjo)

Residuals:
    Min      1Q  Median      3Q     Max
-3.2112 -0.9551 -0.0558  0.6956  4.0167

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.592065   0.586747   6.122 3.57e-09 ***
```

```
bmi         0.015922   0.021682   0.734    0.463
agegrp      0.056147   0.007472   7.514 1.03e-12 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 1.294 on 249 degrees of freedom
Multiple R-squared:  0.1989,Adjusted R-squared:  0.1925
F-statistic: 30.91 on 2 and 249 DF,  p-value: 1.019e-12
```

In the results we can see the estimates of intercept and the parameters for BMI and agegrp. The agegrp coefficient is estimated at 0.056 and the p-value is 0. Thus, the null hypothesis saying that the parameter is zero can be rejected which means that we can conclude that cholesterol is affected by age. An interpretation of the estimate is that for each year of increase in age the mean cholesterol level will increase 0.056.

You can also see that the estimate for BMI has changed to 0.016. Also its p-value has changed to 0.463 and is no longer significant.

An interpretation of the change of estimation and p-value can be that age is explaining most of the relationship between BMI and cholesterol we saw in the simle regression model.

1.2.3 Confidence intervals

Confidence interval is an estimated interval that with a certain probability cover the true parameter, e.g. a 95% confidence interval cover the parameter of bmi with 95% probability. We can use the function **ci.lin** in the library **Epi**. It may be convenient to round off the result. You can change the confidence level (95% is default) with the argument `alpha`.

```
library(Epi)
round(ci.lin(res),3)  # with 95% confidence interval

            Estimate StdErr     z     P   2.5% 97.5%
(Intercept)    3.592  0.587 6.122 0.000  2.442 4.742
bmi            0.016  0.022 0.734 0.463 -0.027 0.058
agegrp         0.056  0.007 7.514 0.000  0.042 0.071

round(ci.lin(res,alpha=0.1),3)  # with 90% confidence interval

            Estimate StdErr     z     P   5.0% 95.0%
(Intercept)    3.592  0.587 6.122 0.000  2.627 4.557
bmi            0.016  0.022 0.734 0.463 -0.020 0.052
agegrp         0.056  0.007 7.514 0.000  0.044 0.068
```

The 95% confidence interval for agegr is 0.042, 0.071

1.2.4 Including factors in the regression model

The variable agegroup was estimated as a continuous variable. However, only individuals aged 30, 40, 50 and 60 years are included so we can also think of agegrp as a categorical varaible. If we convert agegrp to a factor in the model we get a different result.

```
res<-lm(cholesterol~bmi+as.factor(agegrp),data=norsjo)
summary(res)
```

```
Call:
lm(formula = cholesterol ~ bmi + as.factor(agegrp), data = norsjo)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.1476 -0.9234 -0.0442  0.7456  3.8675 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)          5.35783    0.54969   9.747  < 2e-16 ***
bmi                  0.01353    0.02188   0.619   0.5368    
as.factor(agegrp)40  0.43900    0.23005   1.908   0.0575 .  
as.factor(agegrp)50  1.24829    0.23957   5.211 3.97e-07 ***
as.factor(agegrp)60  1.60695    0.23535   6.828 6.64e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.296 on 247 degrees of freedom
Multiple R-squared:  0.2031,Adjusted R-squared:  0.1902 
F-statistic: 15.74 on 4 and 247 DF,  p-value: 1.731e-11
```

The estimates `as.factor(agegrp)40, as.factor(agegrp)50, as.factor(agegrp)60` are actually the differences from the reference group which is 30 years. We can see that an estimate for age 30 is not printed out. Since there are 4 age groups but the estimates are differences from age 30 there will be only 3 estimates. For example age 40 has 0.439 higher cholesterol than age 30.

When we use categories (factors) in the model they are converted to dummy variables which are used in the model instead of the original factor. A dummy variable has either the value 1 or 0. If the factor have two levels one dummy variable will be created and used. In our example with 4 levels there will be 3 dummy variables. The estimates in the results are the coefficients for the dummies. The **contrasts** funtion give information on the dummy variables used with the dummy variables in the columns below. For example for age group 30 all three dummies are = 0 but for age group 50 the dummy varable for 50 is = 1 and the other two dummies = 0.

However, the converting of factors to dummy variables or other kinds of variables (contrasts) can be made in different ways. The default we will use is calculated by **contr.treatment** based on the levels, see below. Let us look at agegrp as a factor. It can be a good idea to include the new variable in the data frame. We see that it has four levels; 30, 40, 50 and 60.

```
fage<-as.factor(norsjo$agegrp)
norsjo<-cbind(norsjo,fage)  # add the factor to the data frame

levels(norsjo$fage)

[1] "30" "40" "50" "60"

contrasts(norsjo$fage)

   40 50 60
30  0  0  0
40  1  0  0
50  0  1  0
60  0  0  1

contr.treatment(1:4)

  2 3 4
1 0 0 0
2 1 0 0
```

```
3 0 1 0
4 0 0 1

contr.treatment(levels(norsjo$fage))

   40 50 60
30  0  0  0
40  1  0  0
50  0  1  0
60  0  0  1
```

An example of other contrasts is:

```
contr.sum(levels(norsjo$fage))

   [,1] [,2] [,3]
30    1    0    0
40    0    1    0
50    0    0    1
60   -1   -1   -1

options()$contrasts   # using contr.treatment is the default

        unordered           ordered
"contr.treatment"      "contr.poly"
```

If we use the new variable fage in the regression we see that the result is unchanged but the names of the age categories have changed.

```
res.n2<-lm(cholesterol~bmi+fage,data=norsjo)
summary(res.n2)

Call:
lm(formula = cholesterol ~ bmi + fage, data = norsjo)

Residuals:
    Min      1Q  Median      3Q     Max
-3.1476 -0.9234 -0.0442  0.7456  3.8675

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.35783    0.54969   9.747  < 2e-16 ***
bmi          0.01353    0.02188   0.619   0.5368
fage40       0.43900    0.23005   1.908   0.0575 .
fage50       1.24829    0.23957   5.211 3.97e-07 ***
fage60       1.60695    0.23535   6.828 6.64e-11 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 1.296 on 247 degrees of freedom
Multiple R-squared:  0.2031,Adjusted R-squared:  0.1902
F-statistic: 15.74 on 4 and 247 DF,  p-value: 1.731e-11
```

Changing the reference level of a factor

What if we don’t want the default reference category (30 years). We can change the reference level by the **relevel** function. Assume we want to change it to age 50.

```
norsjo$newfage<-relevel(norsjo$fage,"50")

levels(norsjo$newfage)

[1] "50" "30" "40" "60"

res<-lm(cholesterol~bmi+newfage,data=norsjo)
summary(res)

Call:
lm(formula = cholesterol ~ bmi + newfage, data = norsjo)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.1476 -0.9234 -0.0442  0.7456  3.8675 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  6.60612    0.60050  11.001  < 2e-16 ***
bmi          0.01353    0.02188   0.619 0.536762    
newfage30   -1.24829    0.23957  -5.211 3.97e-07 ***
newfage40   -0.80929    0.23251  -3.481 0.000591 ***
newfage60    0.35866    0.23310   1.539 0.125170    
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 1.296 on 247 degrees of freedom
Multiple R-squared:  0.2031,Adjusted R-squared:  0.1902 
F-statistic: 15.74 on 4 and 247 DF,  p-value: 1.731e-11
```

Now the newfage30 estimate the difference compared to newfage50. Compare with the earlier result of fage50 with age 30 as reference. It is only the sign that have changed.

1.3 Analysis of variance - ANOVA

Let us start looking at the linear regression above including agegrp only. We can there get an estimate and test of three of the age groups compared to the reference group. This is similar as above but without BMI.

```
lm.age<-lm(cholesterol~fage,data=norsjo)
summary(lm.age)

Call:
lm(formula = cholesterol ~ fage, data = norsjo)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.1143 -0.9409 -0.0727  0.7665  3.8369 

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.6823     0.1644  34.568  < 2e-16 ***
fage40        0.4512     0.2289   1.971   0.0498 *
fage50        1.2809     0.2334   5.487 1.01e-07 ***
fage60        1.6320     0.2315   7.049 1.78e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.294 on 248 degrees of freedom
Multiple R-squared:  0.2019,Adjusted R-squared:  0.1922
F-statistic: 20.91 on 3 and 248 DF,  p-value: 4.127e-12
```

We can see that all age groups differ significantly compared to the reference age 30 years.

Linear regression and ANOVA may be thought of as different methods but as you see they are very related.

1.3.1 One-way ANOVA

The p-values for example fage40 at 0 tell us that the cholesterol level for age group 50 is significantly different from the reference age group 30 years. If we want to test age group as one factor meaning the mean level of cholesterol is equal for all the four age groups (the null hypothesis) the most straightforward test is ANOVA (one-way analysis of variance) by using the function **aov**. Here the contrasts does not matter.

```
aov.age<-aov(cholesterol~fage,data=norsjo)
summary(aov.age)

             Df Sum Sq Mean Sq F value   Pr(>F)
fage          3  105.1   35.03   20.91 4.13e-12 ***
Residuals   248  415.5    1.68
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is calculated from the test statistic F-value 20.91 and is statistically significant.

ANOVA is a test based on the sums of squares (Sum Sq) measuring the variation between and within the groups. The ratio between them is the basis for F so a large F occurs if there is more variation between the groups than expected had the null hypothesis been true. The null hypothesis state that the population mean values are the same in all groups.

Thus, the main differences between ANOVA and regression is that the variable in the ANOVA model is categorical and the null hypothesis of more then two parameters at the same time which is tested with the F test.

However, if you take a look at the last row of the result from the linear regression model lm(cholesterol fage,data=norsjo) above you will find the same F-value and test result there. In fact if you look at the model before that, i.e. lm(cholesterol bmi+newfage,data=norsjo) which also include the continuous variable BMI you will find a F test in the result as well.

1.3.2 Likelihood ratio (LR) test

We can generalize the F test in the ANOVA above. It is actually a comparison between two nested models. As an example we compare the model including age group only compared to a model including both sbp and bmi. It may be somewhat confusing that the name of the LR test is anova.

```
lm.m1<-lm(cholesterol~fage,data=norsjo)
lm.m2<-lm(cholesterol~fage+sbp+bmi,data=norsjo)

anova(lm.m1,lm.m2)

Analysis of Variance Table

Model 1: cholesterol ~ fage
Model 2: cholesterol ~ fage + sbp + bmi
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1    248 415.47
2    246 394.28  2    21.183 6.6081 0.001602 **
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

OBS remember that we first removed all missing data from the data. This comparison can only be made if the numer of observations are the same in the two models. We can see that the result of the **anova** (Likelihood ratio test) is significant saying that we can reject the hypothesis that the parameters of BMI and sbp both =0. Remember that earlier we showed that BMI was not significant so probably most of the result is due to sbp. Let us check.

```
round(summary(lm.m2)$coef,4)

            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.8115     0.6892  5.5305   0.0000
fage40        0.3944     0.2251  1.7523   0.0810
fage50        1.0251     0.2422  4.2325   0.0000
fage60        1.3267     0.2429  5.4626   0.0000
sbp           0.0171     0.0048  3.5798   0.0004
bmi          -0.0038     0.0219 -0.1749   0.8613
```

We can see that sbp is statistically significant but not BMI as suspected. The result of summary was here restricted using **summary()$coef**.

What if we use the likelihood ratio test for the ANOVA example and compare with a model with intercept only. Do you recognise the F test from the ANOVA?

```
lm.m0<-lm(cholesterol~1,data=norsjo)
anova(lm.m0,lm.m1)

Analysis of Variance Table

Model 1: cholesterol ~ 1
Model 2: cholesterol ~ fage
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    251 520.56
2    248 415.47  3    105.09 20.911 4.127e-12 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

1.3.3 Two-way anova

Now we study both the factors agegrp and sbp split into thre categories.

```
sbpcat<-cut(norsjo$sbp,c(0,110,130,Inf))
is.factor(sbpcat)

[1] TRUE

levels(sbpcat)

[1] "(0,110]"   "(110,130]" "(130,Inf]"

norsjo<-cbind(norsjo,sbpcat)  # alternative to add a variable to a data frame
head(norsjo)

  agegrp sex height weight sbp dbp cholesterol smoker      bmi fage newfage    sbpcat
1     60   2    157     61 110  70         6.7      0 24.74745   60      60   (0,110]
2     60   2    157     97 150 100         6.6      0 39.35251   60      60 (130,Inf]
3     60   1    170     74 136  96         8.2      0 25.60554   60      60 (130,Inf]
4     60   2    163     66 156  76         7.5      0 24.84098   60      60 (130,Inf]
5     60   2    166     66 110  70        10.2      0 23.95123   60      60   (0,110]
6     60   2    168     61 130  78         7.3      0 21.61281   60      60 (110,130]

aov.agesbp<-aov(cholesterol~fage+sbpcat,data=norsjo)
summary(aov.agesbp)

             Df Sum Sq Mean Sq F value   Pr(>F)    
fage          3  105.1   35.03  21.666 1.74e-12 ***
sbpcat        2   17.7    8.85   5.474  0.00472 ** 
Residuals   246  397.8    1.62                     
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

1.3.4 Interaction

The effects we have estimated so far is usually called main effects. When we analyse a two-way anova we may also want to include the interaction between the two factors in the model. Interaction can be described as if the effect of one factor is dependent on the level of another factor. For example if the dependent outcome variable is blood pressure and the two factors are smoking (yes/no) and sex. Without interaction we can estimate the general effect of smoking on blood pressure. However, with interaction between smoking and sex the effect of smoking is different for men and women.

Interaction can be estimated in all linear models. It can also be estimated for two continuous variables e.g. cholesterol and bmi or one factor and one continuous varaible, eg. bmi and smoking.

In R formula we use the "*" which adds the interaction parameters in the model. Also ":" can be used but it has a somewhat different meaning.

```
aov.agesbp.int<-aov(cholesterol~fage*sbpcat,data=norsjo)
                    # main effects plus remaining interaction terms
summary(aov.agesbp.int)

             Df Sum Sq Mean Sq F value   Pr(>F)    
fage          3  105.1   35.03  21.278 2.95e-12 ***
sbpcat        2   17.7    8.85   5.376   0.0052 ** 
fage:sbpcat   6    2.6    0.44   0.266   0.9525    
Residuals   240  395.1    1.65                     
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
aov.agesbp.int<-aov(cholesterol~fage:sbpcat,data=norsjo)
                    # only interaction terms
summary(aov.agesbp.int)

             Df Sum Sq Mean Sq F value   Pr(>F)
fage:sbpcat  11  125.4  11.402   6.925 3.86e-10 ***
Residuals   240  395.1   1.646
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

1.4 Generalized linear models

We will not go through the theory behind generalised linear models but show some examples how to estimate and interpret the results. Within this family we can find some common distributions which can be estimated in a similar way as for linerar regression with only minor modifications. The function used for this is **glm**. Normal distribution also belong to this family. Actually it don't seem necessary because of the functions used so far but let us start to compare with the results from those. Let us also compare the class of the resulting objects.

```
glm.m2<-glm(cholesterol~fage+sbp+bmi,data=norsjo)
summary(glm.m2)

Call:
glm(formula = cholesterol ~ fage + sbp + bmi, data = norsjo)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0829  -0.8835  -0.0724   0.7454   4.0417

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.811453   0.689174   5.530 8.15e-08 ***
fage40       0.394414   0.225086   1.752 0.080972 .
fage50       1.025092   0.242196   4.232 3.27e-05 ***
fage60       1.326732   0.242874   5.463 1.15e-07 ***
sbp          0.017069   0.004768   3.580 0.000414 ***
bmi         -0.003833   0.021919  -0.175 0.861326
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 1.602775)

    Null deviance: 520.56  on 251  degrees of freedom
Residual deviance: 394.28  on 246  degrees of freedom
AIC: 841.95

Number of Fisher Scoring iterations: 2

summary(lm.m2) # Same result with glm and lm

Call:
lm(formula = cholesterol ~ fage + sbp + bmi, data = norsjo)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-3.0829 -0.8835 -0.0724  0.7454  4.0417

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.811453   0.689174   5.530 8.15e-08 ***
fage40       0.394414   0.225086   1.752 0.080972 .
fage50       1.025092   0.242196   4.232 3.27e-05 ***
fage60       1.326732   0.242874   5.463 1.15e-07 ***
sbp          0.017069   0.004768   3.580 0.000414 ***
bmi         -0.003833   0.021919  -0.175 0.861326
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 1.266 on 246 degrees of freedom
Multiple R-squared:  0.2426,Adjusted R-squared:  0.2272
F-statistic: 15.76 on 5 and 246 DF,  p-value: 1.861e-13
```

As we can see the results are similar for lm and glm and have a similar structure but there are some differences. If you check the help for **glm** you can see that there are some more arguments than for **lm**.

One of the arguments for glm is family (= gaussian by default). Gaussian is the same as normal distribution which is the reason why we got the same result as above - it is actually the same model. However, chosing another family we will get something completely different. You may check the alternatives for family in the help.

The link function is decribing how the expected observations relates to the linear predictors, e.g. linear or log-linear models with linear and log links, respectively. We don’t need to bother about the link and ather specifications except family, beacuse the default arguments for a given family is most often what we want.

If we check the class we can see that glm belong to the same class as lm but also to another namely glm.

```
class(glm.m2)

[1] "glm" "lm"

class(lm.m2)

[1] "lm"
```

1.4.1 Logistic regression

In logistic regression we study the probability of an event. The response variable is binary i.e. it can only have two outcomes "event" or "no event". It can be coded 1 or 0 where 1 is regarded as the event. We have to change the family object in the **glm** function to binomial (or similarly binomial()). The independent variable can be either numeric 0/1 (with 1 as the event) or TRUE/FALSE. It can also be a factor with two levels where the "event" is the second level (i.e. not the reference level).

Below we analyse if BMI can explain the probability of having high diastolic blood pressure.

```
dbpcat<-cut(norsjo$dbp,c(0,80,Inf))
levels(dbpcat)

[1] "(0,80]"   "(80,Inf]"
norsjo<-cbind(norsjo,dbpcat)

bin.m1<-glm(dbpcat~bmi,data=norsjo,family=binomial)
summary(bin.m1)

Call:
glm(formula = dbpcat ~ bmi, family = binomial, data = norsjo)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0385  -0.9238  -0.6849   1.1251   2.0695

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.18343    1.09608  -5.641 1.69e-08 ***
bmi          0.22784    0.04283   5.319 1.04e-07 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 339.36  on 251  degrees of freedom
Residual deviance: 303.38  on 250  degrees of freedom
AIC: 307.38

Number of Fisher Scoring iterations: 4
```

The estimated parameters here are log(odds ratio). The positive estimate of BMI indicate an increasing odds (risk/(1-risk)) for high diastolic blood pressure with increasing BMI and it is statistically significant. However, usually it is more interesting to get an estimate of the odds ratio (se below).

1.4.2 Confidence intervals

We have earlier seen the possibilities to calculate confidence intervals using package **Epi** and the function **ci.lin**

In log-linear models like logistic regression we may are most often more interested in odds ratios than log(odds) ratios, the latter is the estimated coefficients. To get these estimates and confidence intervals we can use the corresponding function **ci.exp**. By deafault the p-value is not included but can be added. The p-value is the same as for the ordinary result on the log scale.

```
library(Epi)
res.ci<-ci.lin(bin.m1)  # log(odds) ratio with 95% confidence interval
round(res.ci,3)

            Estimate StdErr      z P   2.5%  97.5%
(Intercept)   -6.183  1.096 -5.641 0 -8.332 -4.035
bmi            0.228  0.043  5.319 0  0.144  0.312

res.cie<-ci.exp(bin.m1)  # odds ratio with 95% confidence interval for
round(res.cie,3)

            exp(Est.)  2.5% 97.5%
(Intercept)     0.002 0.000 0.018
bmi             1.256 1.155 1.366

round(ci.exp(bin.m1,pval=T,alpha=0.1),3)  # p-value added and 90% CI
```

```
            exp(Est.) 5.0% 95.0% P
(Intercept)     0.002 0.00 0.013 0
bmi             1.256 1.17 1.348 0
```

Interpretation: The odds ratio of high diastolic blood pressure is increased 1.256 times (i.e. 25.6 %) per unit increase in BMI. Compare with log odds ratio which is increased 0.228 per unit increase in BMI.

1.5 Survival analysis

Survival analysis means methods used to analyse data where the response is a waiting time for a certain event. There are two things that are distinctive; the observations are always positive and all events may not have happened at the time of the analysis. These non-complete observations, called censorings, also carry information and have to be included in the analysis to avoid bias.

1.5.1 Survival data

Anderson data set

The data Anderson.dta consists of remission time data for two groups of leukemia patients with 21 patients in each group.

survt:	Remission time in weeks is denoted and the variable
status:	censored=0, relapse (which is the event)=1
sex:	female=0, male=1
logwbc:	a well-known prognostic indicator of survival for leukemia patients
rx:	(Treatment group =0, Placebo group=1)
catlogwbc:	the variable logWBC divided into low=1, medium=2 and high values=3

```
library(haven)
adf <- read_dta("../data/Anderson.dta")
adf<-as.data.frame(adf)
head(adf)

  survt status sex logwbc rx catlogwbc
1    19      0   0   2.05  0         1
2    17      0   0   2.16  0         1
3    13      1   0   2.88  0         2
4    11      0   0   2.60  0         2
5    10      0   0   2.70  0         2
6    10      1   0   2.96  0         2
```

1.5.2 The survival object

The observed data (independent variable) have two dimensions: time to event and status (status = if the time is an event or a censored observation). There is a function **Surv** for defining such survival object.

```
library(survival)
survtime<-Surv(adf$survt,adf$status)
class(survtime)

[1] "Surv"

head(survtime)

[1] 19+ 17+ 13  11+ 10+ 10
```

1.5.3 Survival curves - Kaplan Meier

A survival curve show the probability of the waiting time to exceed the value on the x-axis. It is the same as one minus the cumulative distribution function (for a continuous variable). The **survfit** function can be used either for a survival table or a plot of the Kaplan Meier.

```
adf<-cbind(adf,survtime)
sf<-survfit(survtime~1,data=adf)
sf

Call: survfit(formula = survtime ~ 1, data = adf)

      n  events  median 0.95LCL 0.95UCL
     42      30      12       8      22

summary(sf)  # events but not censorings are shown

Call: survfit(formula = survtime ~ 1, data = adf)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1     42       2    0.952  0.0329       0.8901        1.000
    2     40       2    0.905  0.0453       0.8202        0.998
    3     38       1    0.881  0.0500       0.7883        0.985
    4     37       2    0.833  0.0575       0.7279        0.954
    5     35       2    0.786  0.0633       0.6709        0.920
    6     33       3    0.714  0.0697       0.5899        0.865
    7     29       1    0.690  0.0715       0.5628        0.845
    8     28       4    0.591  0.0764       0.4588        0.762
   10     23       1    0.565  0.0773       0.4325        0.739
   11     21       2    0.512  0.0788       0.3783        0.692
   12     18       2    0.455  0.0796       0.3227        0.641
   13     16       1    0.426  0.0795       0.2958        0.615
   15     15       1    0.398  0.0791       0.2694        0.588
   16     14       1    0.369  0.0784       0.2437        0.560
   17     13       1    0.341  0.0774       0.2186        0.532
   22      9       2    0.265  0.0765       0.1507        0.467
   23      7       2    0.189  0.0710       0.0909        0.395
```

For example at time=17 you can see that there are 13 at risk (just before) and 1 death. At time 22 there are 9 at risk. Thus there must be 3 censorings between time 17 and 22. It can be seen in the plot.

1.5.4 Comparing Survival curves - log rank test

The same function **survfit** is also used to calculate Kaplan-meier estimates for different groups and to thest if they are equal. For a test of the null hypothesis: the survival is the same in the groups, we use

```
par(mfrow=c(1,2))
plot(sf)  # 95% confidence interval by default
plot(sf,conf.int=F,mark.time=T) # remove CI and add the censorings
```

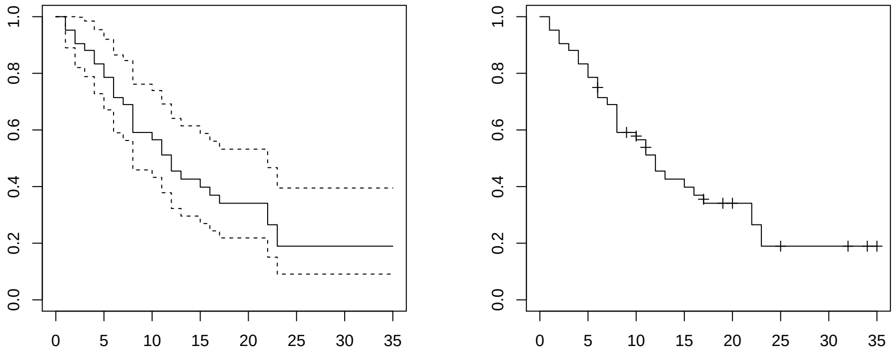


Figure 1.2: Kaplan Meier curves with confidence interval and with censorings, respectively

log-rank test (function **survdiff**). In the example we compare treated (rx=0) vs not treated (rx=1).

```
adf<-cbind(adf,frx=factor(adf$rx,labels=c("treatment","placebo")))
sf2<-survfit(survtime~frx,data=adf)
par(mfrow=c(1,2))

survdiff(survtime~rx,data=adf) # log-rank test

Call:
survdiff(formula = survtime ~ rx, data = adf)

      N Observed Expected (O-E)^2/E (O-E)^2/V
rx=0 21        9     19.3      5.46      16.8
rx=1 21       21     10.7      9.77      16.8

 Chisq= 16.8  on 1 degrees of freedom, p= 4e-05
```

The log-rank test is based on the observed and expected numbers. In this example there is a statistically significant difference between treated and not treated.

Hazard is a rate which can be thought of as e.g. mortality at a certain age or at a certain time of follow-up. The second plot is showing the log-log transformed kurve which can be used to check if the so called hazard functions are proportional. If the curves are parallell this is fulfilled.

1.6 Cox models

A Cox model is a regression method similar as the ones we have seen above where the outcome (independent) variable is a survival time and include censorings. It can be described as $\lambda(t;\mathbf{z}) = \lambda_0(t)exp(z_1\beta_1 + z_2\beta_2 + ...)$ where $\lambda()$ is the hazard function, $\mathbf{z}$ is a vector of predictors andβ_i the parameters. $\lambda_0()$ is called the baseline hazard function which is assumed to be the same for all observations. It is not part of the generalised linear models family so we need another function for the

```
par(mfrow=c(1,2))
plot(sf2,col=c(1,2),lty=c(1,2)) # different color and pattern of the curves added
legend(20,0.95,legend=c(levels(adf$frx)),col=c(1,2),lty=c(1,2),cex=0.8)
plot(sf2,col=c(1,2),lty=c(1,2),fun="cloglog") # plot of log-log(S(t))
```

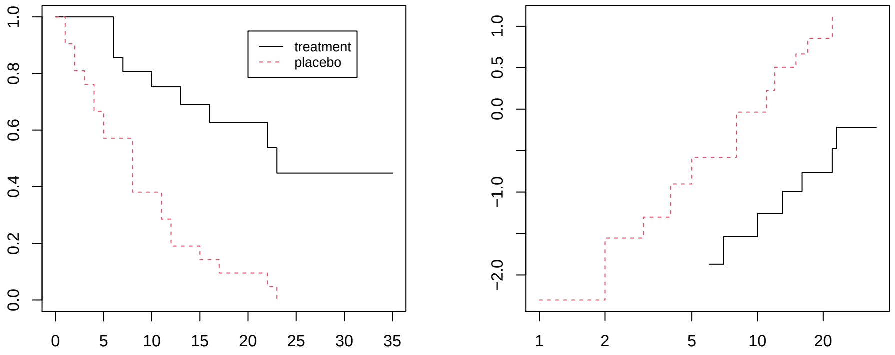


Figure 1.3: Kaplan Meier and log-log transformation of Kaplan Meier

estimation. It is a so called semi-parametric method. The parameters are estimates in a similar way as for other regression models but the survival, i.e. $\lambda_0()$ is not estimated by a parametric method. However, there is an assumption that the hazards should be proportional. This can also be described as the hazard ratios (HR) (and thus the estimated coefficients) are independent of time.

Data set Addicts

We will use the Addicts data set for an example. In a 1991 Australian study by Caplehorn et al., two methadone treatment clinics for heroin addicts were compared to assess patient time remaining under methadone treatment. A patients treatment time was determined as the time, in days, until the person dropped out of the clinic

ID-Patient:	ID
Clinic:	Indicated which methadone treatment clinic the patient attended (coded 1 or 2)
Status:	Indicates if the individual dropped out (coded 1) or was censored (coded 0)
Survt:	The time (in days) until the patient dropped out of the clinic.
Prison:	Indicates whether the patient had a prison record (coded 1) or not (coded 0)
Dose:	A continuous variable for the patients maximum methadone dose (mg/day).
Dosekat:	1 if Dose <65, 2 if Dose 65 or higher

```
addicts <- read_dta("../data/addicts.dta")
addicts

# A tibble: 238 x 6
   Clinic Status Survt Prison  Dose Dosekat
    <dbl>  <dbl> <dbl>  <dbl> <dbl>   <dbl>
 1      1      1   428      0    50       1
 2      1      1   275      1    55       1
 3      1      1   262      0    55       1
 4      1      1   183      0    30       1
 5      1      1   259      1    65       2
 6      1      1   714      0    55       1
 7      1      1   438      1    65       2
 8      1      0   796      1    60       1
 9      1      1   892      0    50       1
10      1      1   393      1    65       2
# ... with 228 more rows

dim(addicts)

[1] 238   6
```

As before we have to start creating a survival object.

```
survtime<-Surv(addicts$Survt,addicts$Status)
fclinic<-as.factor(addicts$Clinic)
addicts<-cbind(addicts,survtime,fclinic)  # add to the data farme

fitc<-coxph(survtime~Dose+fclinic,data=addicts)
summary(fitc)

Call:
coxph(formula = survtime ~ Dose + fclinic, data = addicts)

  n= 195, number of events= 123
   (43 observations deleted due to missingness)

              coef exp(coef)  se(coef)      z Pr(>|z|)
Dose     -0.028668  0.971739  0.007191 -3.987 6.69e-05 ***
fclinic2 -0.939289  0.390906  0.232238 -4.045 5.24e-05 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

         exp(coef) exp(-coef) lower .95 upper .95
Dose        0.9717      1.029    0.9581    0.9855
fclinic2    0.3909      2.558    0.2480    0.6162

Concordance= 0.627  (se = 0.031 )
Likelihood ratio test= 36.83  on 2 df,   p=1e-08
Wald test            = 32.59  on 2 df,   p=8e-08
Score (logrank) test = 34.07  on 2 df,   p=4e-08
```

Interpretation: We can see that we get both coef and exp(coef) in the result but only CI for the latter. The model is log-linear so we most often prefer exp(coef). Both dose and clinic are significant with low p-values. The hazard ratio (HR) is multiplied by 0.972, i.e. -2.83% for one unit increase in dose and the HR is 0.391 for clinic 2 vs clinic 1 (which is the reference).

1.6.1 Model check

We start with looking at the plot of survival and log-log plot for the two clinics. The curves in the second plot is not parallel, i.e. the condition of proportional hazards dont seem to be fulfilled.

Proportional hazards can also be tested looking at time specific estimates using the **cox.zph** function. This function also provides a plot of an estimate of the time-dependent coefficient beta(t). If the proportional hazards assumption holds then the true beta(t) function would be a horizontal line.

```
sfc<-survfit(survtime~fclinic,data=addicts)
par(mfrow=c(1,2))
plot(sfc,col=c(1,2))
plot(sfc,fun="cloglog",col=c(1,2))
```

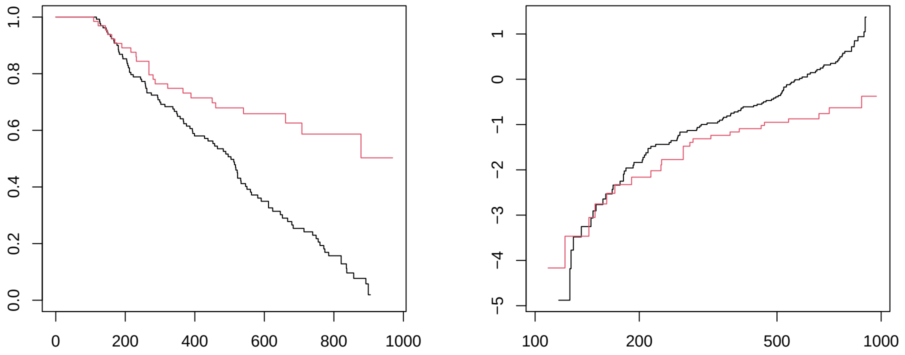


Figure 1.4: Kaplan Meier and log-log transformation of Kaplan Meier for clinic

```
ph<-cox.zph(fitc)
ph

           chisq df      p
Dose     0.00717  1 0.9325
fclinic 7.38630  1 0.0066
GLOBAL   7.42611  2 0.0244
```

We can see that the result is significant for clinic but not for dose.

```
par(mfrow=c(1,2))
plot(ph)
```

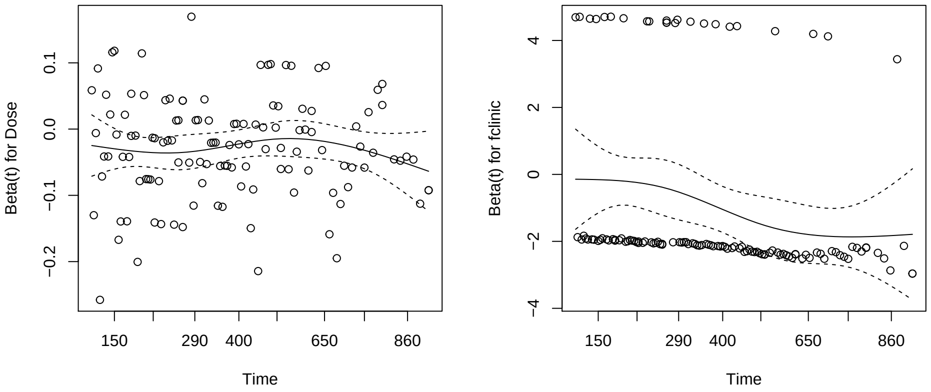


Figure 1.5: Time specific estimates for the estimates (plot of the proportional hazards test)

1.6.2 Stratification

Now since the results show that clinic is statistically significant it seems inappropriate to use this model. However, we can overcome this problem (especially if we are not specifically interested in the hazard ratio of clinic) by stratifying the model on clinic.

```
library(survival)
fitcs<-coxph(survtime~Dose+strata(fclinic),data=addicts)
summary(fitcs)

Call:
coxph(formula = survtime ~ Dose + strata(fclinic), data = addicts)

  n= 195, number of events= 123
   (43 observations deleted due to missingness)

          coef exp(coef)  se(coef)      z Pr(>|z|)
Dose -0.028061  0.972329  0.007264 -3.863 0.000112 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

     exp(coef) exp(-coef) lower .95 upper .95
Dose    0.9723      1.028    0.9586    0.9863

Concordance= 0.602  (se = 0.031 )
Likelihood ratio test= 14.81  on 1 df,   p=1e-04
Wald test            = 14.93  on 1 df,   p=1e-04
Score (logrank) test = 15.13  on 1 df,   p=1e-04
```

If we compare the results of HR for dose the new result 0.972 using stratification on clinic does not differ much from the model without stratification. So actually the stratification was not so necessary even if there was a significant result for violation of the proportionality assumption. This is however not always the case.