Practical 1: Linear and logistic regression

Verena Zuber, Jelena Besevic, Alpha Forna, and Saredo Said

24/1/2019

Part 1: Analysing type 2 diabetes progression using linear regression

Type 2 diabetes is a long-term metabolic disorder that is characterized by high blood sugar, insulin resistance, and relative lack of insulin. The number of people diagnosed with diabetes in the UK has more than doubled in the last twenty years. According to diabetes.org.uk/ figures show that there are now almost 3.7 million people living with a diagnosis of the condition in the UK, an increase of 1.9 million since 1998.

In this practical we consider an observational study that measures progression in type 2 diabetes as outcome and several clinical parameters like age. sex, and bmi, but also common risk factors like map: blood pressure, tc: total cholesterol, ldl: low-density lipoprotein, hdl: high-density lipoprotein, tch: total cholesterol, ltg: triglycerides, and glu: glucose. The dataset includes n=442 cases. It is available in the lars package, which is easy to install using for example the install.packages("lars") command.

As a first step we load the data and assign x as a data.frame of the predictors and y as the quantitative outcome of type 2 diabetes progression.

```
library(lars)
data(diabetes)
x = as.data.frame.matrix(diabetes$x)
y = diabetes$y
```

From the literature we know that the following 6 predictors are important for type 2 diabetes progression: sex, age, bmi, glu, map and ltg. We consider these 6 predictors as model 1.

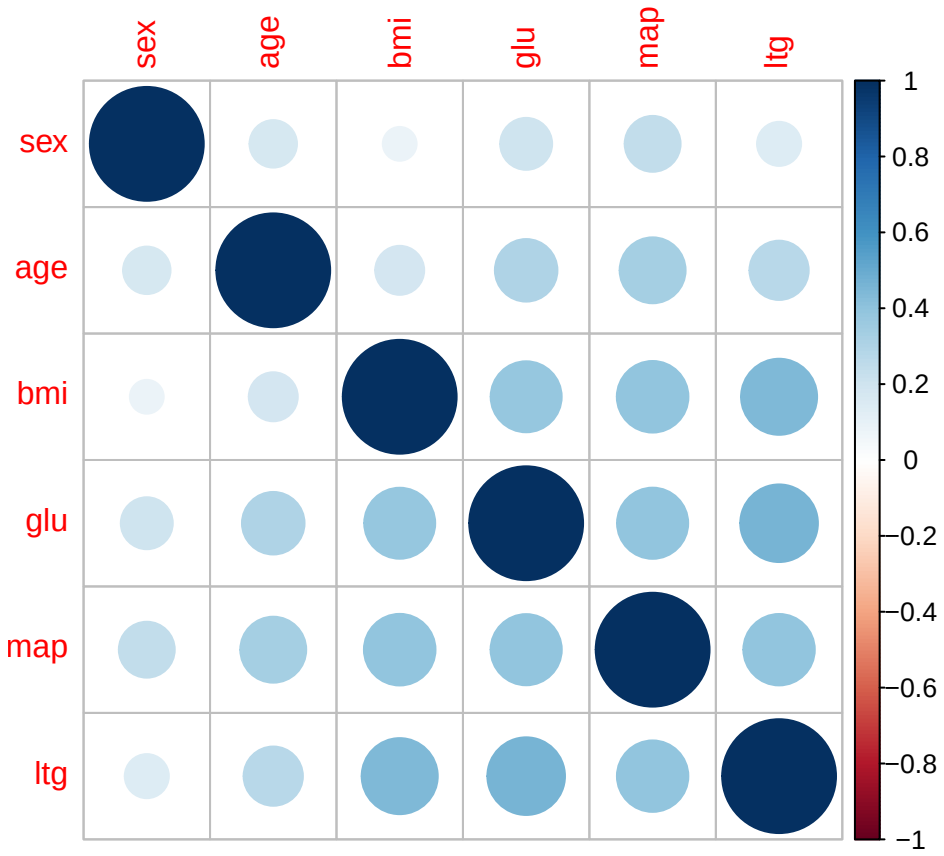
Question 1.1 Look at the correlation structure between those 6 predictors and discuss the implications. Use the function corrplot() in the corrplot package to visulalise the correlation structure.

Reply: The command cor() computes the correlation matrix between the indicated variables.

```
library(corrplot)
x_cor=cbind(x$sex, x$age, x$bmi, x$glu, x$map, x$ltg)
colnames(x_cor) = c("sex","age","bmi","glu","map","ltg")
cor(x_cor)
```

```
##           sex       age       bmi       glu       map       ltg
## sex 1.0000000 0.1737371 0.0881614 0.2081332 0.2410132 0.1499176
## age 0.1737371 1.0000000 0.1850847 0.3017310 0.3354267 0.2707768
## bmi 0.0881614 0.1850847 1.0000000 0.3886800 0.3954153 0.4461586
## glu 0.2081332 0.3017310 0.3886800 1.0000000 0.3904294 0.4646705
## map 0.2410132 0.3354267 0.3954153 0.3904294 1.0000000 0.3934781
## ltg 0.1499176 0.2707768 0.4461586 0.4646705 0.3934781 1.0000000
```

```
corrplot(cor(x_cor))
```

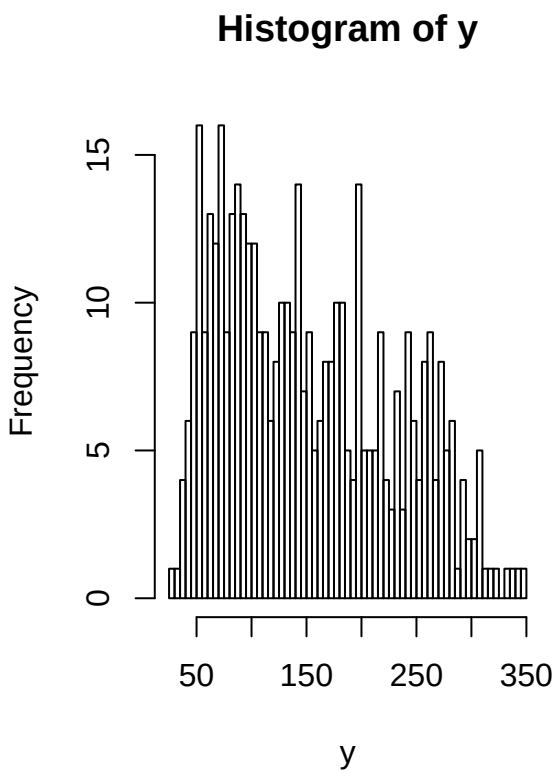


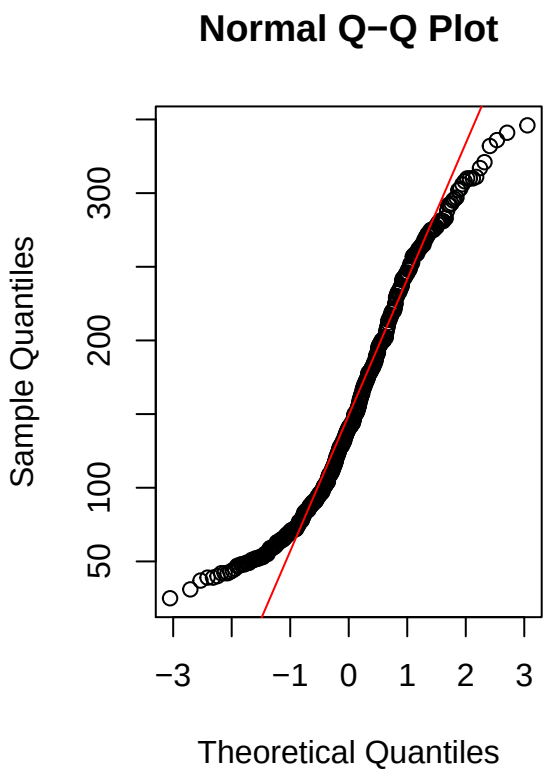
The correlation between the predictors is low to moderate. The strongest correlation is observed between ltg, map and bmi.

Question 1.2 Does the outcome disease progression follow a Normal-distributed? Look at general summary statistics of y, plot a histogram and a q-q plot against the Normal-distribution.

Reply: The outcome is certainly not perfectly Normal-distributed, but for now this assumption is fine.

```
par(mfrow=c(1,2))
hist(y,breaks=50)
qqnorm(y)
qqline(y,col="red")
```






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

Question 1.3 Fit a linear model including the predictors of model 1 (sex, age, bmi, glu, map and ltg) using the lm() function and discuss the summary of the model.

Reply: bmi has the strongest impact on disease progression with a very small p-value, followed by ltg and map. There is a minimal sex difference. An increase of unit in ltg increases the disease progression score by 540.784. Overall there is a good model fit with an adjusted R-squared of 0.4814.

```
lm1 = lm(y~sex+age+bmi+glu+map+ltg, data=x)
summary(lm1)
```

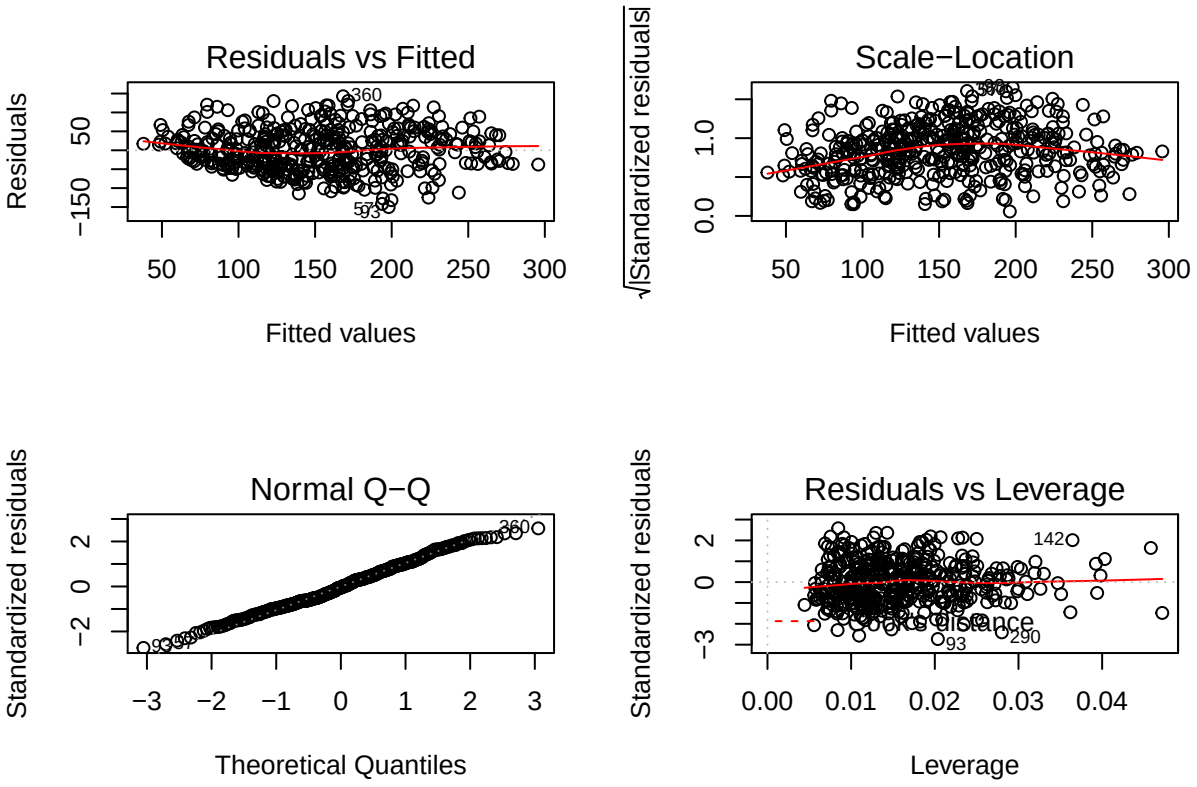
```
##
## Call:
## lm(formula = y ~ sex + age + bmi + glu + map + ltg, data = x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -150.043  -39.021   -2.356   38.871  142.753
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  152.133      2.641  57.610  < 2e-16 ***
## sex         -139.901     57.931  -2.415   0.0161 *
## age          -45.201     60.581  -0.746   0.4560
## bmi          586.705     65.453   8.964  < 2e-16 ***
## glu           68.628     66.989   1.024   0.3062
## map          292.048     66.417   4.397 1.38e-05 ***
## ltg          540.784     67.806   7.975 1.35e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 55.52 on 435 degrees of freedom
## Multiple R-squared:  0.4884, Adjusted R-squared:  0.4814
## F-statistic: 69.22 on 6 and 435 DF,  p-value: < 2.2e-16
```

Question 1.4 Perform model diagnostics and outlier detection of model 1. Do you think this is a good model fit? Justify your answers.

Reply: All of the diagnostic plots look fine.

```
par(mfrow=c(2,2))
plot(lm1,which=1)
plot(lm1,which=3)
plot(lm1,which=2)
plot(lm1,which=5)
```




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

Question 1.5 For model 1, compute the OLS regression coefficient estimate using matrix multiplication $\hat{beta}_{OLS} = (x^t x)^{-1} x^t y$. Use the solve() function to invert a matrix. R distinguishes between scalar multiplication ($\star$) and matrix multiplication (%$\star$%). Make sure to use the matrix multiplication (%$\star$%) for this task and ensure that your matrices have the correct dimensions. Add an intercept by including a row of ones like for example this.

```
x1 =  cbind(rep(1,nrow(diabetes$x)), x$sex, x$age, x$bmi, x$glu, x$map, x$ltg)
```

Reply:

```
beta_ols = solve(t(x1) %*% x1) %*% t(x1) %*% y
```

Question 1.6 Compute the regression coefficient estimate using the sample covariance based estimate, defined as $\hat{beta}_{COV} = cov(x)^{-1} cov(xy)$. Use the solve() function to invert the covariance matrix cov(x) of dimension

6 x 6 and compute this estimate without the intercept using

```
x11 =  cbind(x$sex, x$age, x$bmi, x$glu, x$map, x$ltg)
```

As an additional hint, please use the solve() function to invert the covariance matrix cov(x) of dimension 6 x 6.

Reply:

```
beta_cov = solve(cov(x11)) %*% cov(x11,y)
```

Question 1.7 Compare the 3 regression coefficient estimates from questions 1.4-1.6.

Reply: All three estimators give the same answer for the regression coefficients.

```
compare_beta = matrix(NaN, ncol = 3, nrow = 7)
colnames(compare_beta)=c("lm","ols","cov")
compare_beta[,1] = lm1$coefficients
compare_beta[,2] = beta_ols
compare_beta[2:7,3] = beta_cov
compare_beta
```

```
##              lm        ols        cov
## [1,]  152.13348  152.13348        NaN
## [2,] -139.90054 -139.90054 -139.90054
## [3,]  -45.20079  -45.20079  -45.20079
## [4,]  586.70493  586.70493  586.70493
## [5,]   68.62836   68.62836   68.62836
## [6,]  292.04767  292.04767  292.04767
## [7,]  540.78417  540.78417  540.78417
```

Question 1.8 Fit model 2 that only includes glucose and compare how it differs from the multivariable model 1.

Reply: The effect of glucose substantially attenuates after conditioning on other covariates. In an univariable model the beta is 619, while in the multivariable model the effect is 69 and not significant. This might suggest that the disease progression does not depend on glu but on other correlated risk factors. Also the adjusted R-squared (0.1444) is much smaller than in the multivariable model.

```
lm2=lm(y~glu, data = x)
summary(lm2)
```

```
##
## Call:
## lm(formula = y ~ glu, data = x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -153.069  -57.716   -5.466   54.656  186.839
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  152.133      3.392  44.851   <2e-16 ***
## glu          619.223     71.312   8.683   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 71.31 on 440 degrees of freedom
## Multiple R-squared:  0.1463, Adjusted R-squared:  0.1444
```

```
## F-statistic:  75.4 on 1 and 440 DF,  p-value: < 2.2e-16
```

Part 2: Predict type 2 diabetes progression using linear regression

Assume we only observed the first 300 cases and need to use the first 300 cases as training data.

```
x_train = data.frame(x[1:300,])
y_train = y[1:300]
```

Now we can consider the remaining 112 cases as new data points for whom we want to predict disease progression.

```
x_new = data.frame(x[301:442,])
y_new = y[301:442]
```

Question 2.1 Use the linear model 1 to predict the disease progression for the 112 cases with predictor information stored in x_new.

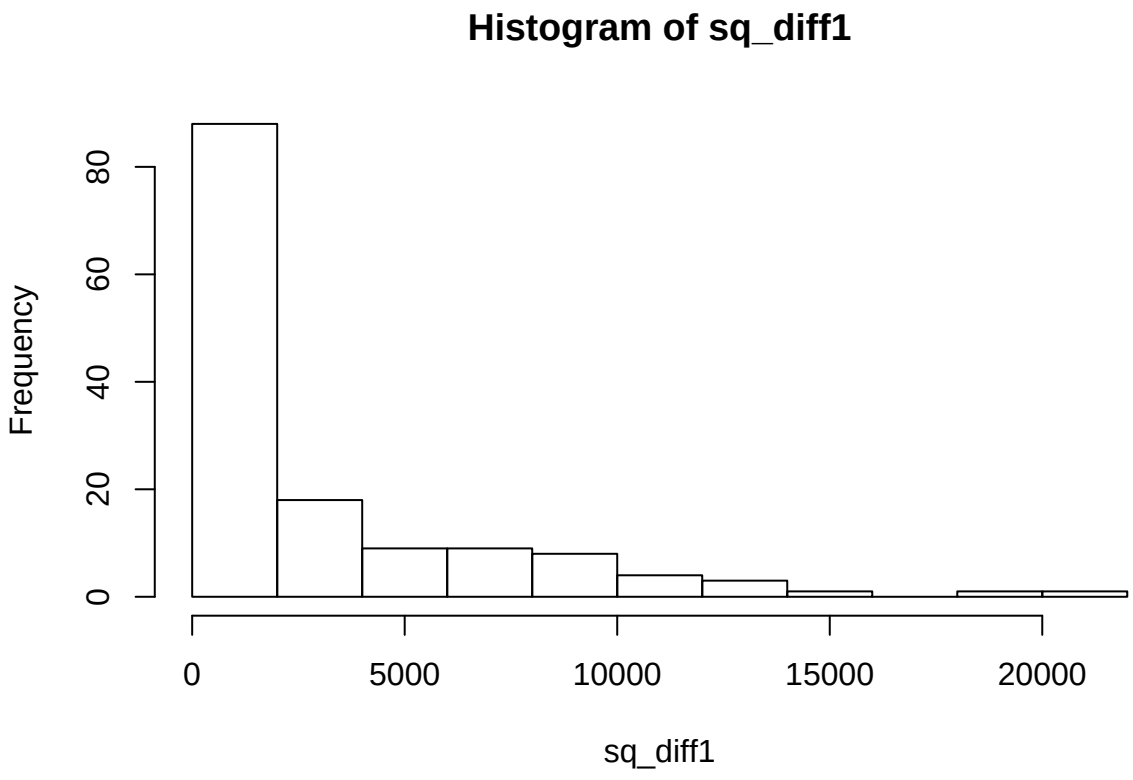
Reply:

```
lm1_train = lm(y_train~sex+age+bmi+glu+map+ltg, data=x_train)
y_predict1 = predict.lm(lm1_train,x_new)
```

Question 2.2 Evaluate the error of your prediction based on linear model 1 by computing the squared difference between the predicted progression and the actual observed progression saved in y_new. Plot a histogram of the squared difference and compute the mean and median.

Reply:

```
sq_diff1 = (y_predict1 - y_new)^2
hist(sq_diff1)
```



```
summary(sq_diff1)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##     0.001   280.290  1510.249  2979.775  4032.200 20455.815
```

Question 2.3 Repeat the steps 2.1 and 2.2 using the univariable linear model 2 including only glucose. Contrast the prediction error of linear model 2 with the prediction error of linear model 1.

Reply: The prediction using model 2 (glu) has a much larger mean squared error (MSE) than model 1 (sex, age, bmi, glu, map and ltg).

```
lm2_train = lm(y_train~glu, data=x_train)
y_predict2 = predict.lm(lm2_train,x_new)
sq_diff2 = (y_predict2 - y_new)^2
hist(sq_diff2)
```


```
summary(sq_diff2)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##     2.055   923.010  3074.797  4975.382  7295.496 22350.041
```

Question 2.4 Is it good practise to evaluate the prediction performance on a single training data? How appropriate is the split to take the first 300 cases?

Reply: No, ideally one would repeat the process (split the data, learn the prediction rule and evaluate) many times to eliminate the randomness of the data split. When just doing this process once we only might capture a prediction rule that is specific for the first 300 cases, but does not generalise to other data-sets. The main aim of building a prediction rule is that it generalises to new data.

Part 3: Distinguishing between severe and light cases of type 2 diabetes using logistic regression

Doctors are particularly concerned with type 2 diabetes cases that have a bad disease progression, in particular cases that have a disease progression score larger than 200. Binarise your outcome like this:

```
y_binary = as.numeric(y>200)
```

Question 3.1 Fit a generalised linear model using the glm() function that can distinguish between bad disease progression and normal progression. Use the 6 predictors as considered in model 1. Look at the summary of the glm output and interpret the findings.

Reply: Similar like in the linear model we find the strongest effects for bmi and ltg, and also for map. The beta coefficients represent log odds, so the odd-ratio of is equal to exp(13). The effect of glu is not significant.

```
glm1 = glm(y_binary~sex+age+bmi+glu+map+ltg, data = x, family=binomial)
summary(glm1)
```

```
##
## Call:
## glm(formula = y_binary ~ sex + age + bmi + glu + map + ltg, family = binomial,
##     data = x)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9181  -0.5767  -0.2620   0.3627   3.0045
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.6504     0.1753  -9.416  < 2e-16 ***
## sex          -2.3334     3.0187  -0.773    0.440
## age          -1.7950     3.3780  -0.531    0.595
## bmi          21.0236     3.6247   5.800 6.63e-09 ***
## glu           1.0369     3.5349   0.293    0.769
## map          13.2277     3.3866   3.906 9.39e-05 ***
## ltg          22.0036     3.7190   5.917 3.29e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 518.87  on 441  degrees of freedom
## Residual deviance: 323.04  on 435  degrees of freedom
## AIC: 337.04
##
## Number of Fisher Scoring iterations: 6
```

Question 3.2 Consider now model 2 including only glucose. Fit a glm and see if glucose can distinguish between bad disease progression and normal progression.

Reply: When not considering the other covariates the effect of glu becomes highly significant also for the binary trait. But again the model fit (look here at AIC (Lecture 2a)) is not as good as for model 1. The AIC of model 1 is much smaller than the AIC of model 2, thus model 1 provides the better overall fit.

```
glm2 = glm(y_binary~glu, data = x, family=binomial)
summary(glm2)
```

```
##
## Call:
## glm(formula = y_binary ~ glu, family = binomial, data = x)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5650  -0.7989  -0.6073   0.9609   2.2670
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.098      0.118  -9.303  < 2e-16 ***
## glu           16.587      2.604   6.370 1.89e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 518.87  on 441  degrees of freedom
## Residual deviance: 471.24  on 440  degrees of freedom
## AIC: 475.24
##
## Number of Fisher Scoring iterations: 4
```

Question 3.3 Look again at the training data (x_train and ybin_train) based on the first 300 cases, where

```
ybin_train = y_binary[1:300]
```

Build a prediction rule based on model 1 using the training data (x_train and ybin_train) using the glm function. In a second step predict which of the new samples (using x_new as predictor matrix) are at high risk for having a bad diagnosis. How many of the 112 new observations have a probability larger than 0.5 to have bad progression?

PS Use the inverse logit function ($logit^{-1}(eta) = exp(eta)/(exp(eta)+1)$) to transform the linear predictor ($eta = x\beta$) back to a probability which ranges between 0 and 1.

Reply: First we learn the prediction rule and we predict the linear predictor eta for the new observations. Note that eta is on the linear and continuous scale.

```
glm_predict = glm(ybin_train~age+sex+bmi+map+ltg, data = x_train, family=binomial)
eta = predict.glm(glm_predict,x_new)
summary(eta)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -6.6064 -3.4261 -1.5884 -1.5545  0.1552  4.0714
```

In order to transform the linear predictor to probabilities we use the inverse logit link. There are 39 cases that are predicted to have a bad disease progression.

```
p_predict = (exp(eta)/(exp(eta)+1))
summary(p_predict)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00135 0.03149 0.16970 0.30604 0.53872 0.98323
```

```
sum(p_predict>0.5)
```

```
## [1] 39
```

Part 4 (Optional): Which risk factors are important for type 2 diabetes progression?

Look again at the complete dataset including all n=442 cases including all 10 predictors. How would you perform variable selection to decide which variables are important for disease progression in type 2 diabetes?

Reply: First we can fit a linear regression including all predictors.

```
lm_all=lm(y~as.matrix(x))
summary(lm_all)
```

```
##
## Call:
## lm(formula = y ~ as.matrix(x))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -155.829  -38.534   -0.227   37.806  151.355
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      152.133      2.576  59.061  < 2e-16 ***
## as.matrix(x)age  -10.012     59.749  -0.168 0.867000
## as.matrix(x)sex -239.819     61.222  -3.917 0.000104 ***
## as.matrix(x)bmi  519.840     66.534   7.813 4.30e-14 ***
## as.matrix(x)map  324.390     65.422   4.958 1.02e-06 ***
## as.matrix(x)tc  -792.184    416.684  -1.901 0.057947 .
## as.matrix(x)ldl  476.746    339.035   1.406 0.160389
## as.matrix(x)hdl  101.045    212.533   0.475 0.634721
## as.matrix(x)tch  177.064    161.476   1.097 0.273456
## as.matrix(x)ltg  751.279    171.902   4.370 1.56e-05 ***
## as.matrix(x)glu   67.625     65.984   1.025 0.305998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.15 on 431 degrees of freedom
## Multiple R-squared:  0.5177, Adjusted R-squared:  0.5066
## F-statistic: 46.27 on 10 and 431 DF,  p-value: < 2.2e-16
```

The full model includes many covariates that do not have a significant effect, eg. age or hdl. Still for the interpretation we would prefer to have our model adjusted for age. Some clincal covariates might be better left in a model so that they do not act as a confounder. With respect to variable selection, backward or forward selection are often used, but they are not recommended as these approaches depend on the order how variables are included. In case you want to test a very specific hypothesis, for example if glu should be included or not it is possible to use an anova test for linear models.

```
lm_noglu=lm(y~as.matrix(x[,-10]))
anova(lm_noglu,lm_all)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ as.matrix(x[, -10])
## Model 2: y ~ as.matrix(x)
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1    432 1267064
## 2    431 1263983  1    3080.4 1.0504  0.306
```

The above test says that there is no improvement in the model fit when including glu. This type of comparison of nested models is only valid to test very specific hypothesis, and not for variable selection in general. In lectures 2a and 2b we will learn more about variable selection and in lecture 3b we cover regularised regression approaches like lasso and elastic net that can perform variable selection.