STATS 762 Assignment 2

Francis Tang, UPI ftan638

Due: 11 April 2019

1. The data for this question represent a random sample of 79 patients that underwent a particular type of liver surgery. The response is the survival time time of the patient. Prior to the surgery, data were obtained on four variables that were thought to be possible predictors of survival time:

(a) Create a data frame in R. Do an initial assessment of the data and summarize your findings. You are asked to investigate how the four possible predictors are related to survival time. Your ultimate goal is to identify a model that can be used to predict survival times for future patients.

We first import the dataset:

```
liver.df <- read.csv("~/Desktop/STATS 762/liver.txt", sep="")
```

```
str(liver.df)
```

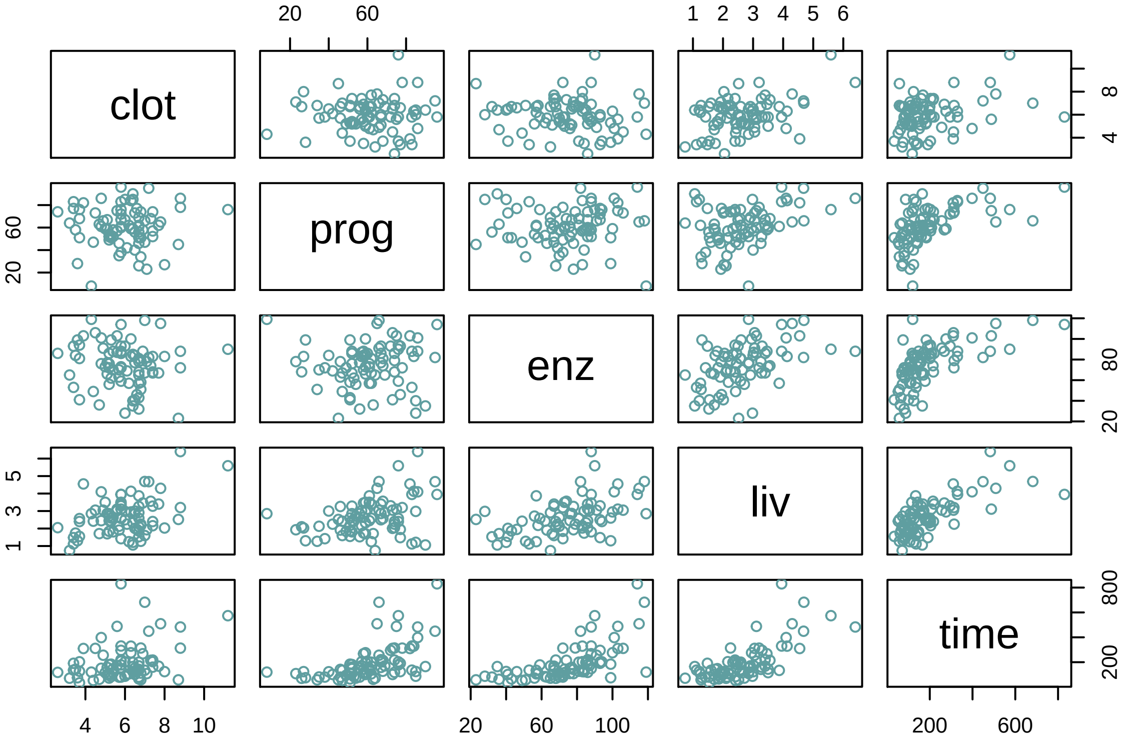
```
## 'data.frame':    79 obs. of  5 variables:
##  $ clot: num  6.7 5.1 7.4 6.5 7.8 5.8 5.7 3.7 6 3.7 ...
##  $ prog: int  62 59 57 73 65 38 46 68 67 76 ...
##  $ enz : int  81 66 83 41 115 72 63 81 93 94 ...
##  $ liv : num  2.59 1.7 2.16 2.01 4.3 1.42 1.91 2.57 2.5 2.4 ...
##  $ time: int  200 101 204 101 509 80 80 127 202 203 ...
```

Let's start with paired plot, which helps us to identify relationships between regressors.

```
# get a summary of each column
summary(liver.df)
```

```
##       clot             prog            enz              liv
##  Min.   : 2.600   Min.   : 8.00   Min.   : 23.0   Min.   :0.740
##  1st Qu.: 5.150   1st Qu.:51.50   1st Qu.: 64.0   1st Qu.:1.940
##  Median : 5.800   Median :61.00   Median : 77.0   Median :2.570
##  Mean   : 5.933   Mean   :61.08   Mean   : 74.8   Mean   :2.676
##  3rd Qu.: 6.700   3rd Qu.:74.00   3rd Qu.: 88.0   3rd Qu.:3.230
##  Max.   :11.200   Max.   :96.00   Max.   :119.0   Max.   :6.400
##       time
##  Min.   : 34.0
##  1st Qu.:103.5
##  Median :148.0
##  Mean   :194.1
##  3rd Qu.:216.0
##  Max.   :830.0
```

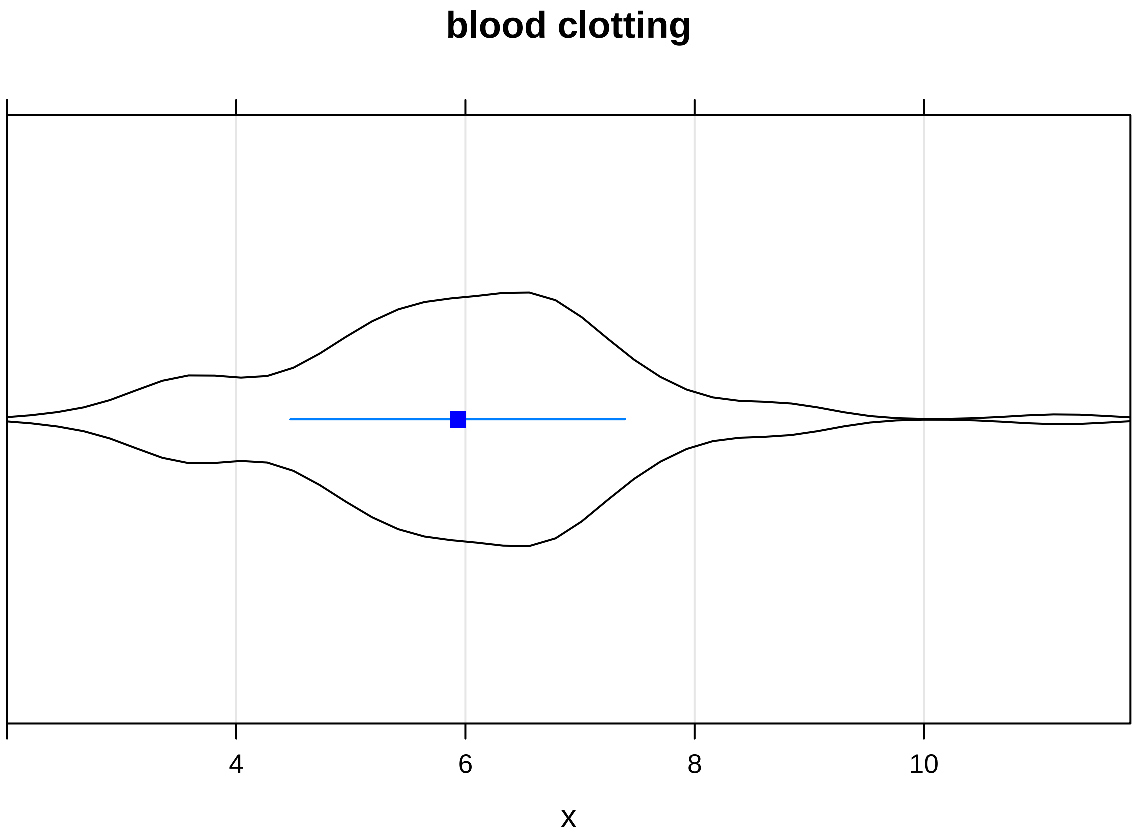
```
# get a paired plot
pairs(liver.df, col = "cadetblue")
```



Violin plots blow help us know the distribution of each variable.

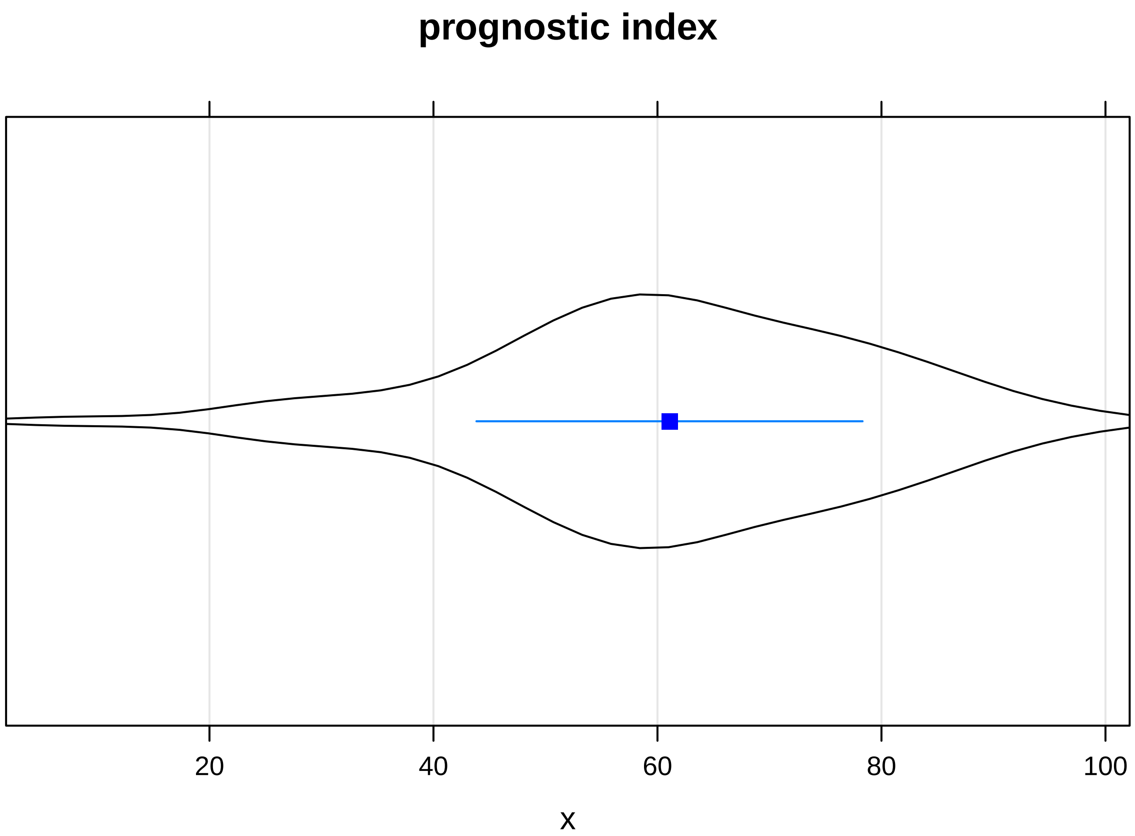
```
library(lattice)
library(violinmplot)
par(mfrow=c(2,3))
violinmplot(liver.df$clot, main="blood clotting")
```

```
## Warning in bwplot.numeric(x = x, data = data, panel = panel.violinm, ...):
## explicit 'data' specification ignored
```



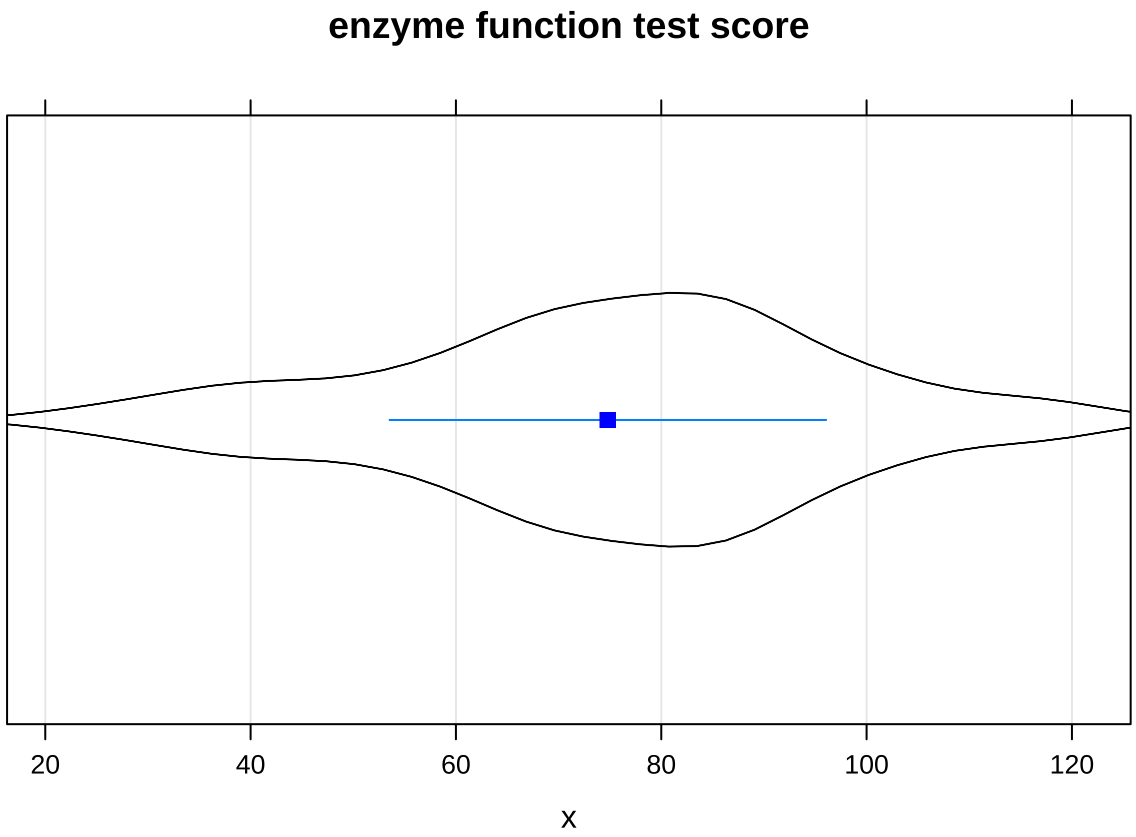
```
#title(main="blood clotting")
violinmplot(liver.df$prog,main="prognostic index")
```

```
## Warning in bwplot.numeric(x = x, data = data, panel = panel.violinm, ...):
## explicit 'data' specification ignored
```



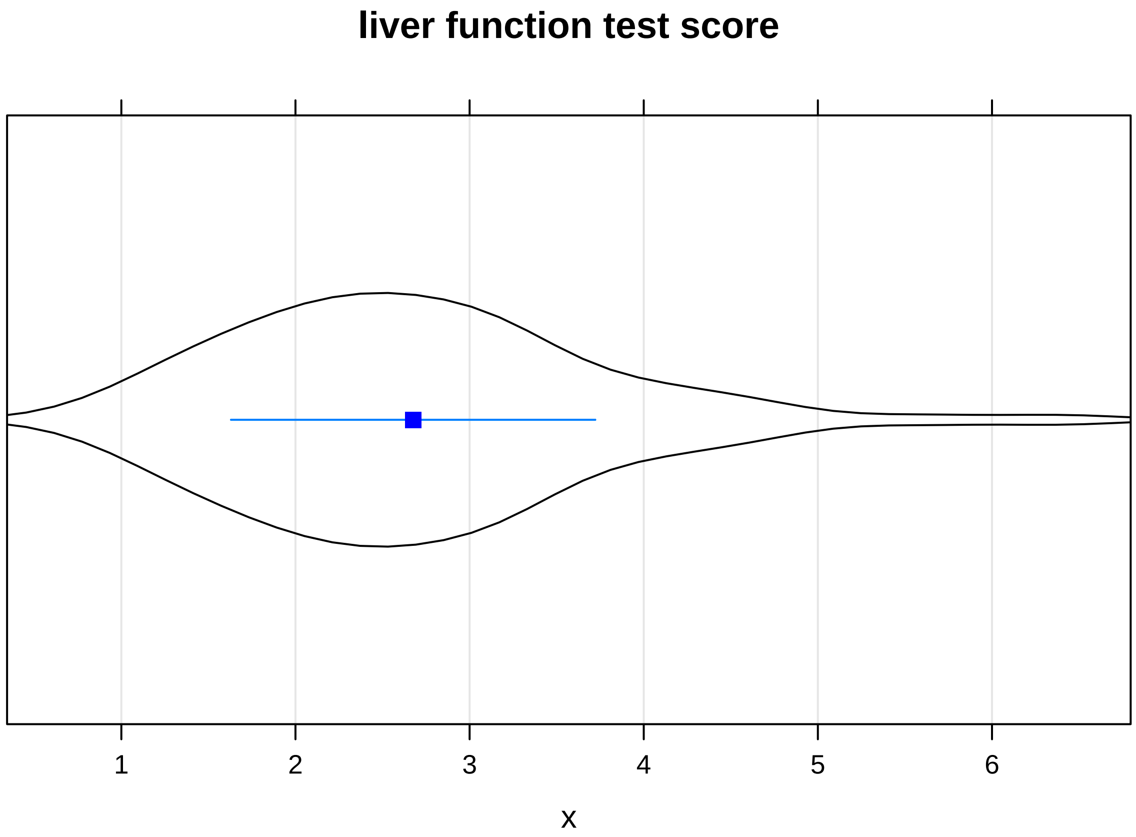
```
#title(main="prognostic index")
violinmplot(liver.df$enz,main="enzyme function test score")
```

```
## Warning in bwplot.numeric(x = x, data = data, panel = panel.violinm, ...):
## explicit 'data' specification ignored
```



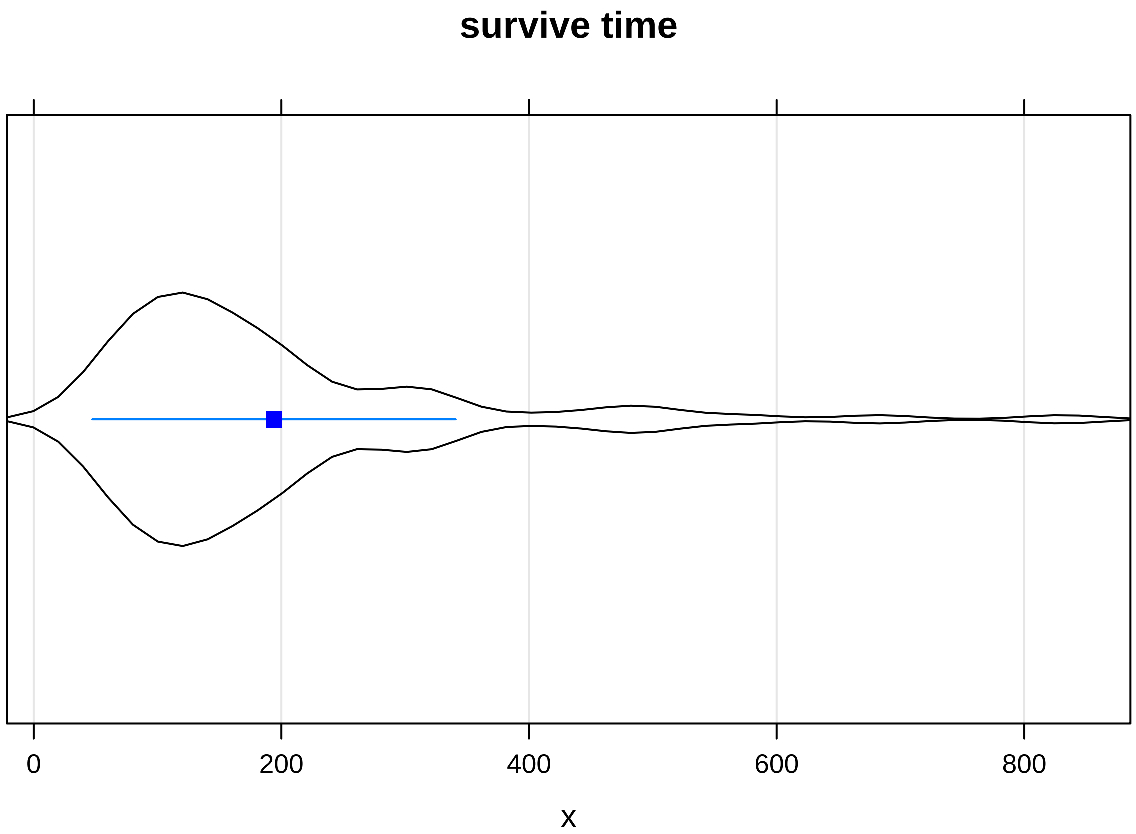
```
#title(main="enzyme function test score")
violinmplot(liver.df$liv,main="liver function test score")
```

```
## Warning in bwplot.numeric(x = x, data = data, panel = panel.violinm, ...):
## explicit 'data' specification ignored
```



```
#title(main="liver function test score")
violinmplot(liver.df$time,main="survive time")
```

```
## Warning in bwplot.numeric(x = x, data = data, panel = panel.violinm, ...):
## explicit 'data' specification ignored
```



```
#title(main="survive time")
```

Let's have a look with the Variance Inflation Factors, VIF indicates extensive multicollinearity:

```
round(diag(solve(cor(liver.df[,1:4]))),2)
```

```
## clot prog  enz  liv
## 1.54 1.29 1.69 2.39
```

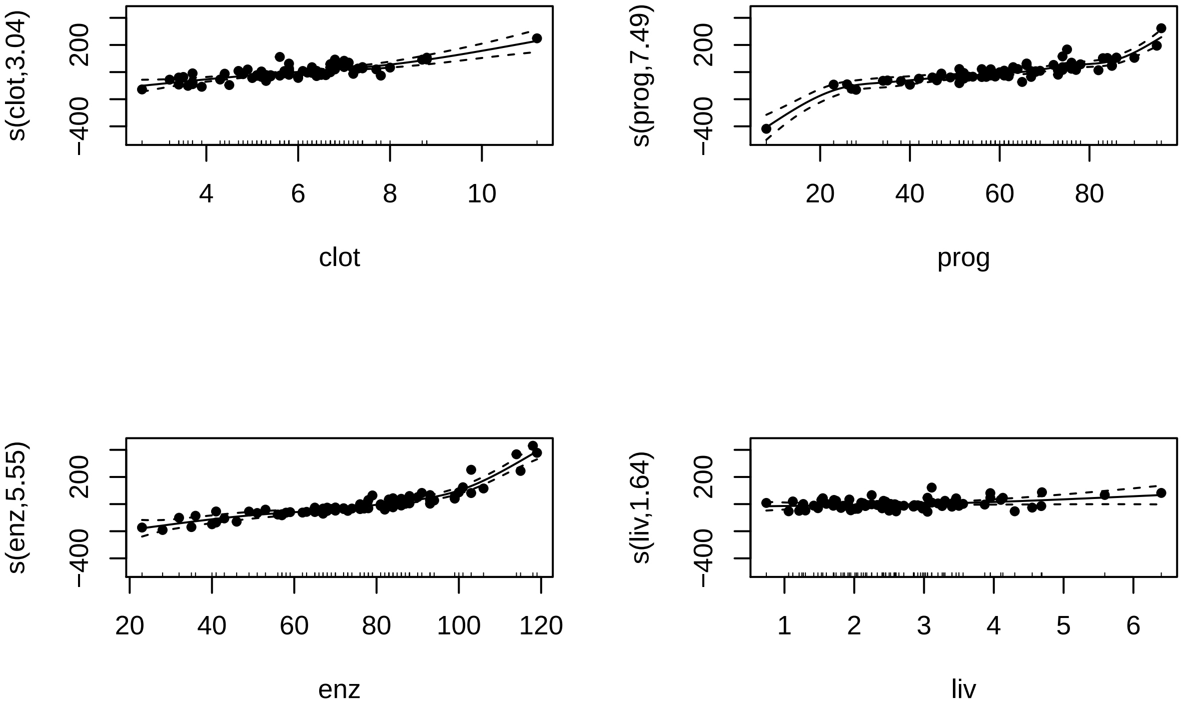
The gam plots below shows that prog and enz may need to be lognised to be fit in a model.

```
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-26. For overview type 'help("mgcv-package")'.
```

```
liver.gam <- gam(time~s(clot)+s(prog)+s(enz)+s(liv),data = liver.df)
plot(liver.gam, residuals = T, pages = 1, pch = 20)
```



(b) To start, fit the basic model that uses time as the response and the remaining variables as regressors. Do a full set of diagnostics on this model. Give a brief assessment of this model based on these diagnostics.

We first build a full model and an ANOVA table based on the model:

```
fit1.lm <- lm(time~., data = liver.df)
summary(fit1.lm)
```

```
##
## Call:
## lm(formula = time ~ ., data = liver.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -93.06 -42.50 -12.10  17.67 312.75
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -612.9135    54.4239 -11.262  < 2e-16 ***
## clot          33.9809     6.0015   5.662 2.69e-07 ***
## prog           4.1786     0.4660   8.967 1.92e-13 ***
## enz            4.1954     0.4315   9.722 7.28e-15 ***
## liv           13.5831    10.4374   1.301    0.197
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 62.35 on 74 degrees of freedom
## Multiple R-squared:  0.8285, Adjusted R-squared:  0.8192
## F-statistic: 89.35 on 4 and 74 DF,  p-value: < 2.2e-16
```

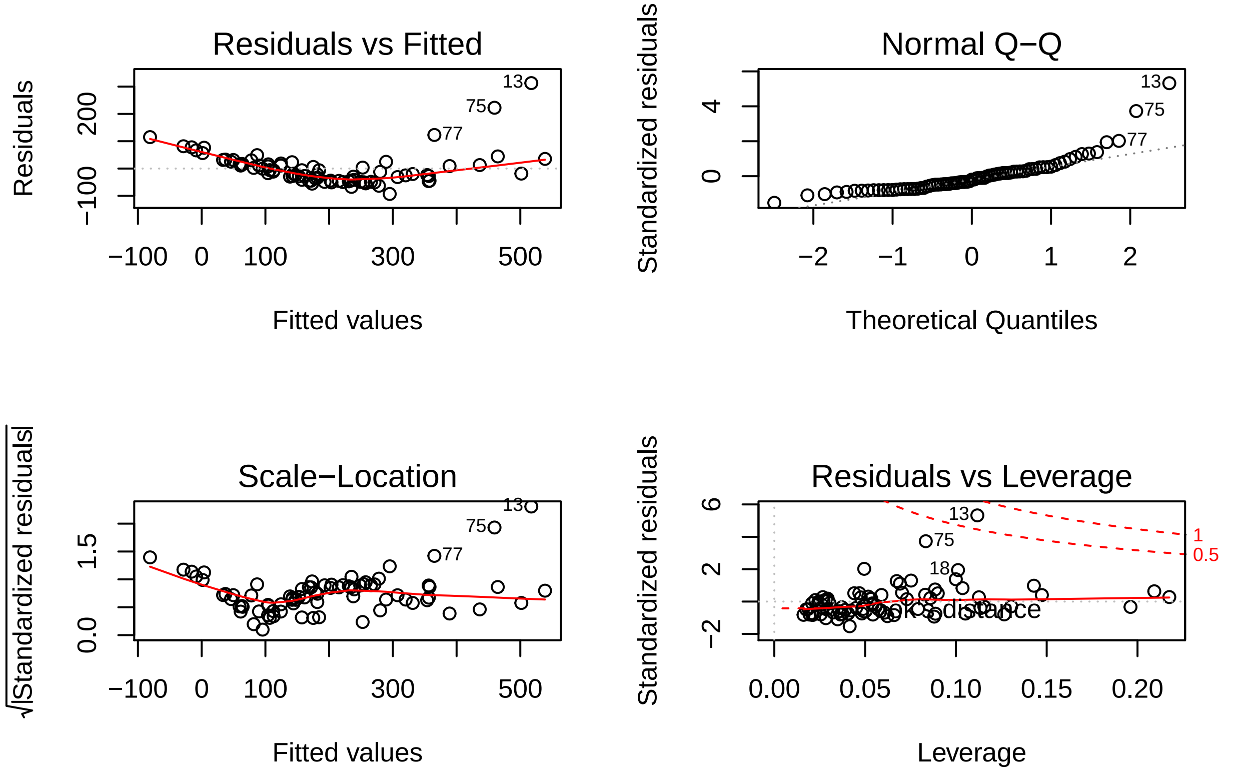
```
anova(fit1.lm)
```

```
## Analysis of Variance Table
```

```
##
## Response: time
##           Df Sum Sq Mean Sq  F value    Pr(>F)
## clot       1 164806  164806  42.3930 7.906e-09 ***
## prog       1 499799  499799 128.5630 < 2.2e-16 ***
## enz        1 718162  718162 184.7325 < 2.2e-16 ***
## liv        1   6584    6584   1.6936    0.1972
## Residuals 74 287681    3888
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

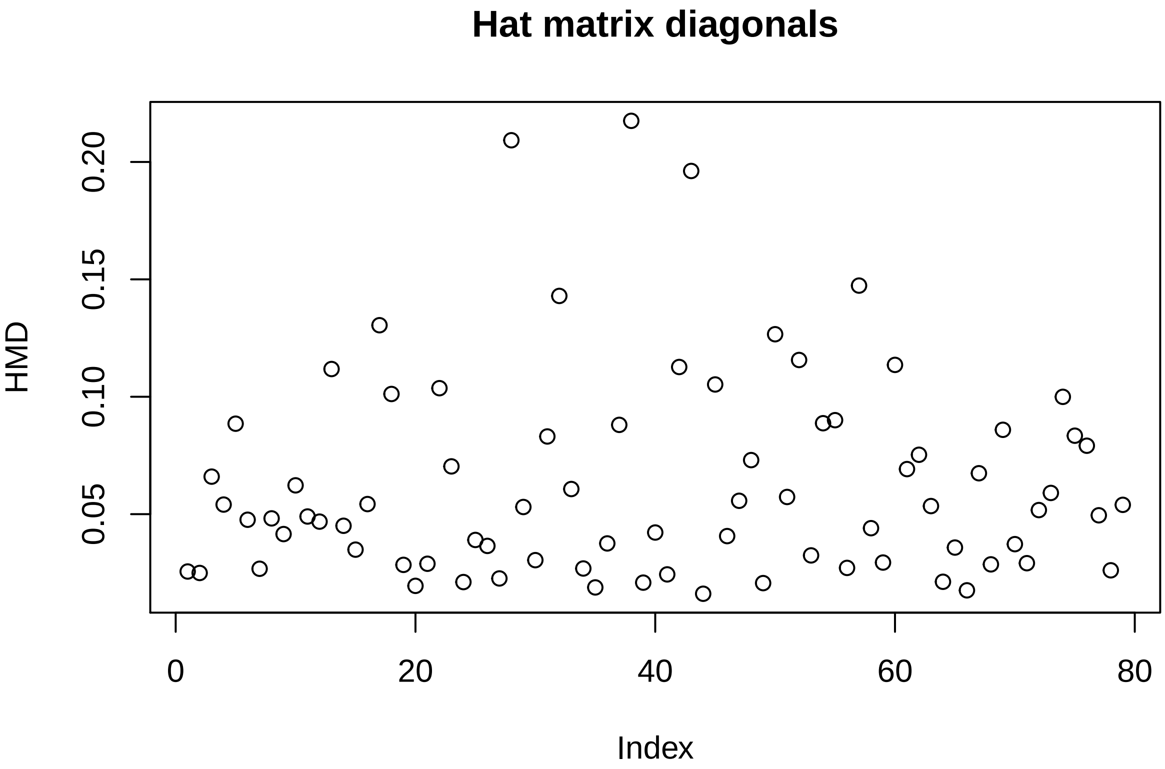
It seems like clot, prog and enz are very significant for surviving time.

```r
par(mfrow=c(2,2))
plot(fit1.lm)
```



As this data is ungrouped plots involving residuals are of little value. Plots that we should look at includethe HMD's, Cook's Distance and Deviance Changes.The plot for the hat matrix diagonals indicates a few unusually large values.

```r
par(mfrow=c(1,1))
HMD<-hatvalues(fit1.lm)
plot(HMD,main="Hat matrix diagonals")
```

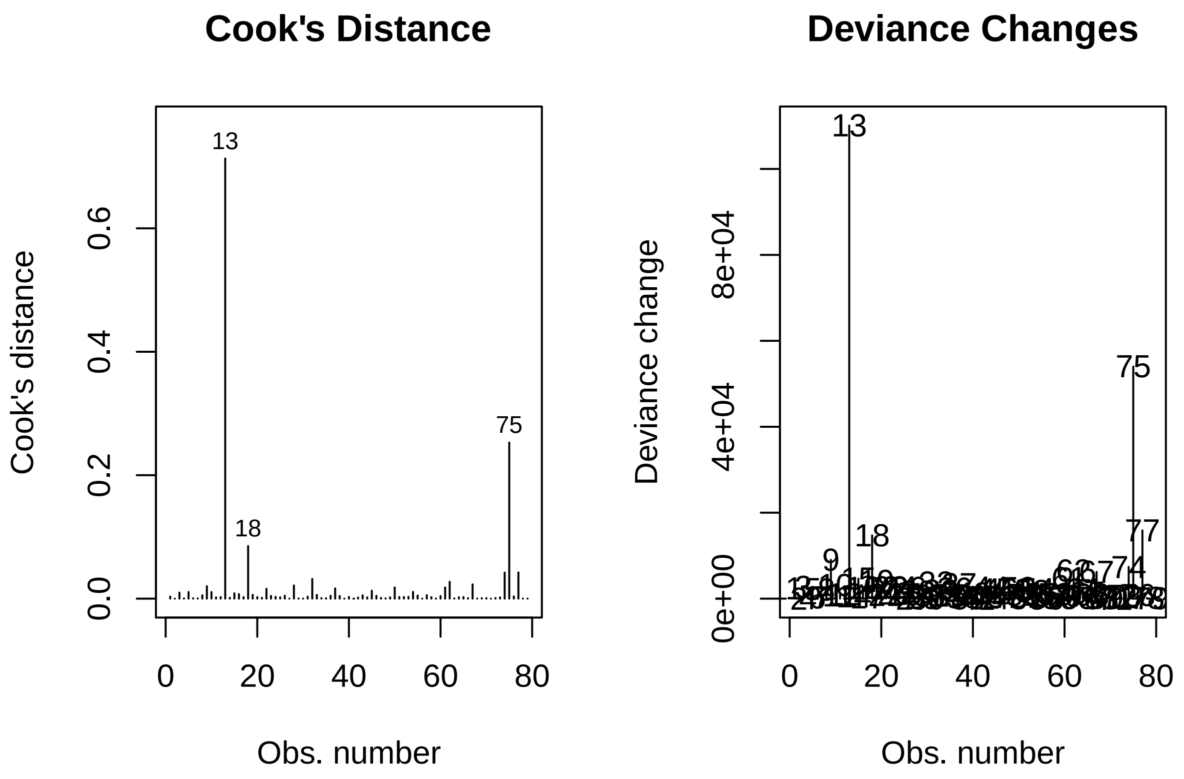


```
liver.df[which(HMD>.1),]
```

```
##    clot prog enz  liv time
## 13  5.8   96 114 3.95  830
## 17  6.0   85  28 2.98   87
## 18  3.7   51  41 1.55   34
## 22  3.4   83  53 1.12  136
## 28 11.2   76  90 5.59  574
## 32  8.7   45  23 2.52   58
## 38  4.3    8 119 2.85  120
## 42  3.6   28  99 1.30   75
## 43  8.8   86  88 6.40  483
## 45  3.4   77  93 1.48  191
## 50  3.9   82 103 4.55  310
## 52  6.4   85  40 1.21  125
## 57  6.4   90  35 1.06  165
## 60  8.0   27  83 2.03  124
```

The observations with values greater than 0.1 correspond to individuals that have unusually large values forone or more of the variables that measure the amount of time spent on different activities.Plots for Cook's Distance and Deviance changes are:

```
dev.r<-residuals(fit1.lm,type="deviance")
pear.r<-residuals(fit1.lm,type="pearson")
Dev.change<-dev.r^2 + pear.r^2*HMD/(1-HMD)
par(mfrow=c(1,2))
plot(fit1.lm,which=4,main="Cook's Distance",caption=" ")
plot(Dev.change,ylab="Deviance change", xlab="Obs. number",type="h", main="Deviance Changes")
text(Dev.change)
bigdev=4
abline(h=bigdev, lty=2)
```



There are 3 points that stand out as having large values of Cook's Distance (13, 18 and 75) but they arenot close to the usual cut off of 0.5. There are quite very few points that have values of Deviance Changes of above 8e+04 but given the number of these it doesn't make sense to consider them as an usual. Point 13 stands out as being the largest on both plots so we might consider the impact of deleting it.

```
liver.df[13,]
```

```
##    clot prog enz  liv time
## 13  5.8   96 114 3.95  830
```

```
fit2.lm<-lm(time~.,data=liver.df[-13,])
round(rbind(coefficients(fit1.lm),coefficients(fit2.lm)),3)
```

```
##      (Intercept)   clot  prog   enz    liv
## [1,]    -612.913 33.981 4.179 4.195 13.583
## [2,]    -544.258 32.209 3.569 3.699 17.978
```

It is very curious that after deleted 13, the coefficients decreased for clot, prog and enz. But liv becomes more associated with response.

To check for multicolinearity we need to get the VIF's:

```
Xmat<-model.matrix(fit1.lm)[,-1]
round(diag(solve(cor(Xmat))),2)
```

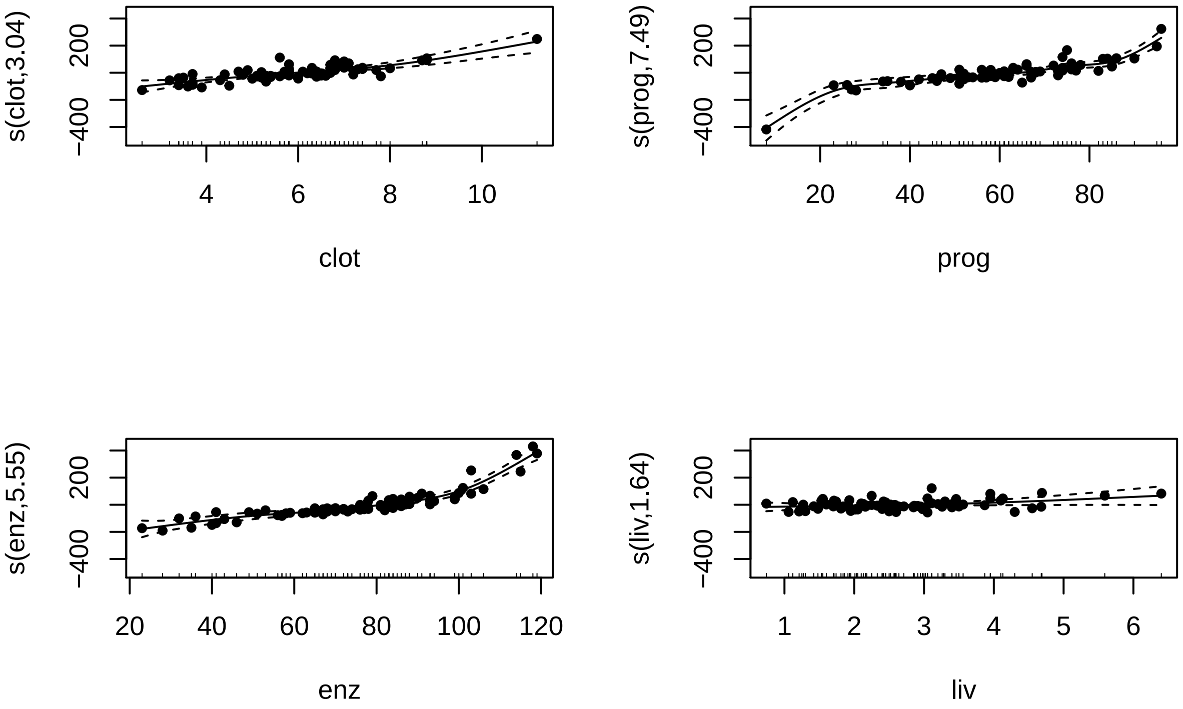
```
## clot prog  enz  liv
## 1.54 1.29 1.69 2.39
```

These values indicate that there are no strong relationships between the regressors but only liv became more significant.

(c) Now try to find an improved model for this data using the model building cycle described in the lectures. Briefly, outline the steps in your search and present key pieces of evidence. Present the full set of diagnostics for your chosen model and briefly discuss these.

Before perform any kind of transformation, we need to look at the gam plots again to determine which transformation we are supposed to perform in order to achieve better goals for modelling.

```r
liver.gam <- gam(time~s(clot)+s(prog)+s(enz)+s(liv),data = liver.df)
plot(liver.gam, residuals = T, pages = 1, pch = 20)
```



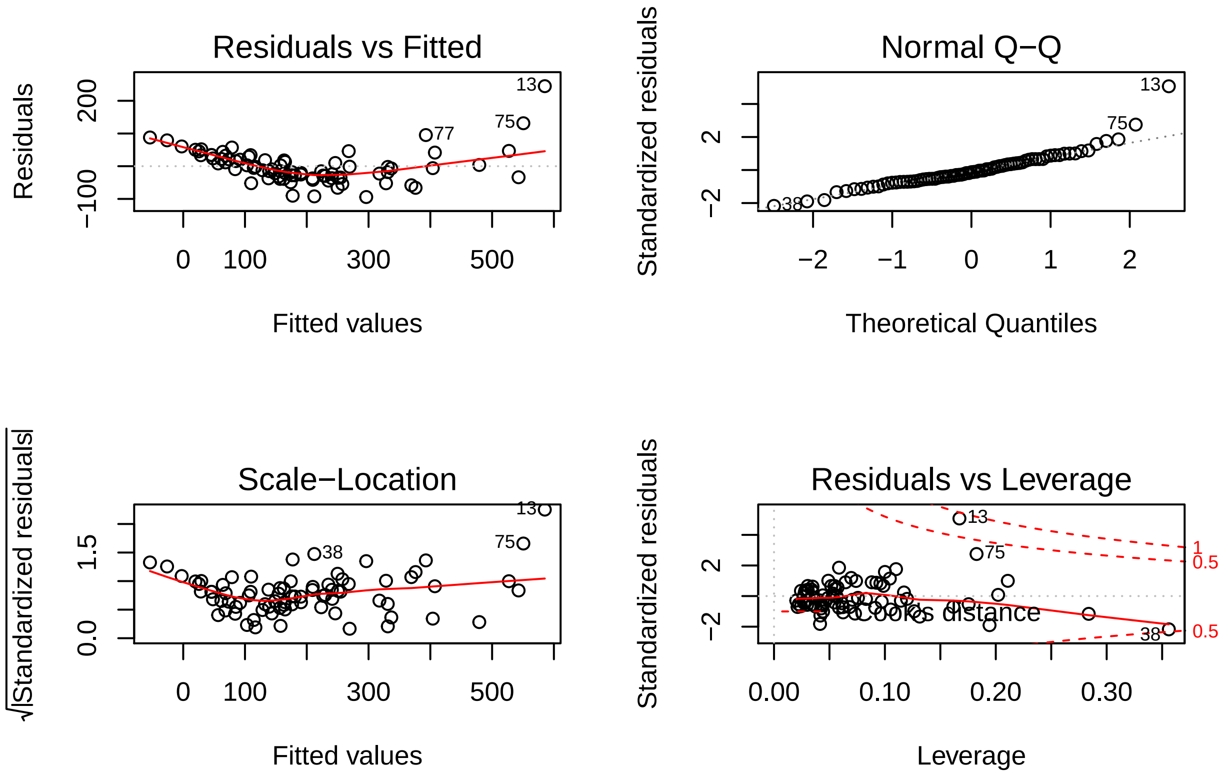
The gam plot above shows that clot and liv seem like linear but prog and enz may need to be added a quadratic or logged term? Let's try to add a quadratic term for enz:

```r
liverA.lm <- lm(time~clot+prog+enz+I(enz^2)+liv,data=liver.df)
summary(liverA.lm)
```

```
##
## Call:
## lm(formula = time ~ clot + prog + enz + I(enz^2) + liv, data = liver.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -94.242 -32.679  -6.835  26.321 244.704
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -346.91556   67.05277  -5.174 1.94e-06 ***
## clot          35.09114    5.09335   6.890 1.67e-09 ***
## prog           4.04014    0.39596  10.203 1.07e-15 ***
## enz           -3.71187    1.49156  -2.489   0.0151 *
## I(enz^2)       0.05568    0.01018   5.468 6.05e-07 ***
## liv           10.20900    8.87251   1.151   0.2536
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.87 on 73 degrees of freedom
## Multiple R-squared:  0.8783, Adjusted R-squared:   0.87
## F-statistic: 105.4 on 5 and 73 DF,  p-value: < 2.2e-16
```

It is surprisingly interesting to see that I(enz^2) becomes significant!

```
par(mfrow=c(2,2))
plot(liverA.lm)
```



Besides the potential outliers 13 and 75, this model looks like a good fit to the data, but there are still some curves in the plots. Let's try to log(time) this time just like fitting a poissom model:

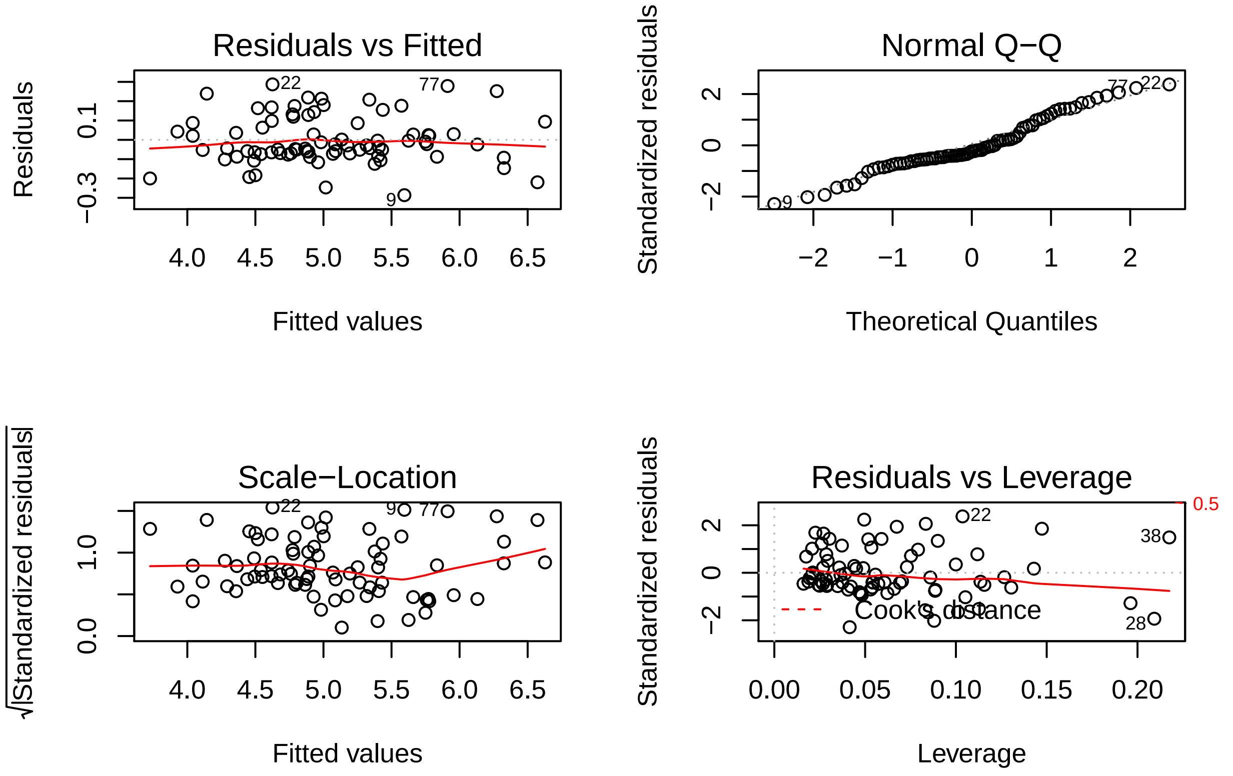
```
liverB.lm <- lm(log(time)~clot+prog+enz+liv,data=liver.df)
summary(liverB.lm)
```

```
##
## Call:
## lm(formula = log(time) ~ clot + prog + enz + liv, data = liver.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28628 -0.07140 -0.02893  0.08687  0.28691
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.1232212  0.1114601  10.077 1.58e-15 ***
## clot         0.1695496  0.0122910  13.795  < 2e-16 ***
## prog         0.0213022  0.0009543  22.322  < 2e-16 ***
## enz          0.0220506  0.0008838  24.951  < 2e-16 ***
## liv         -0.0095714  0.0213759  -0.448    0.656
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.1277 on 74 degrees of freedom
## Multiple R-squared:  0.962,  Adjusted R-squared:   0.96
## F-statistic: 468.7 on 4 and 74 DF,  p-value: < 2.2e-16
```

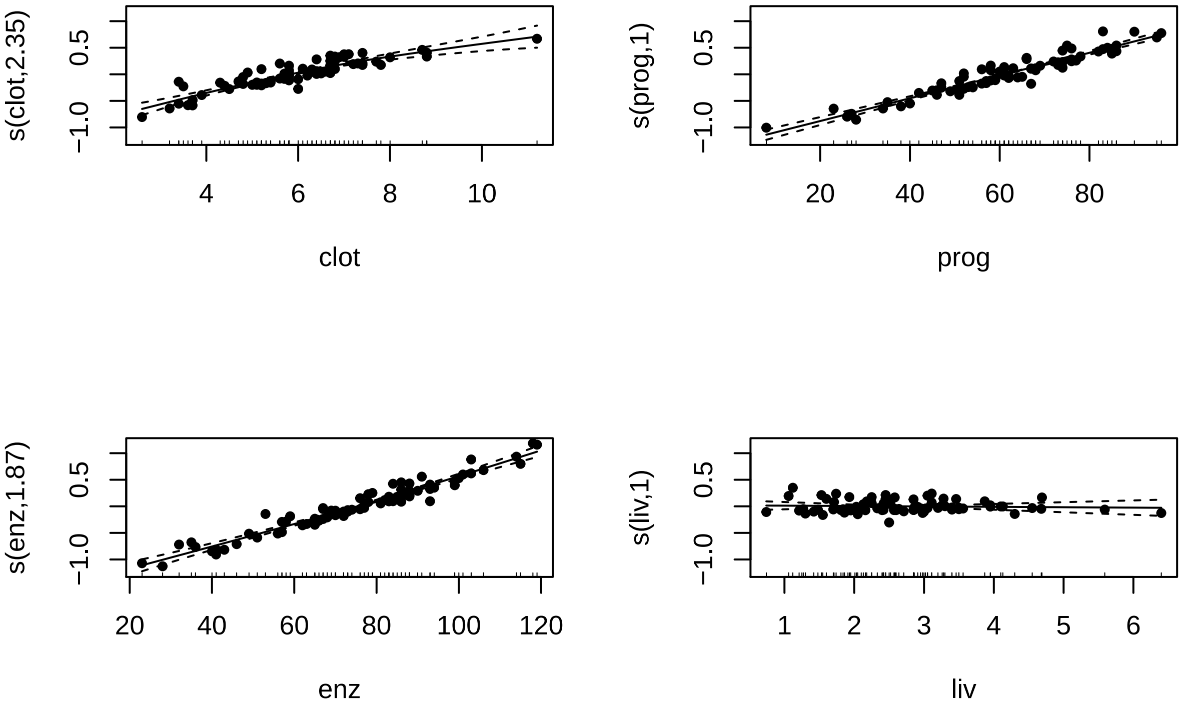
It looks like clot, prog and enz are all very significant. Let's see the plots below:

```
par(mfrow=c(2,2))
plot(liverB.lm)
```



Besides some potential outliars like 22 and 77, the curve looks very flat and Q-Q plots fits very well to a linear regression. This model may be the best fit so far.

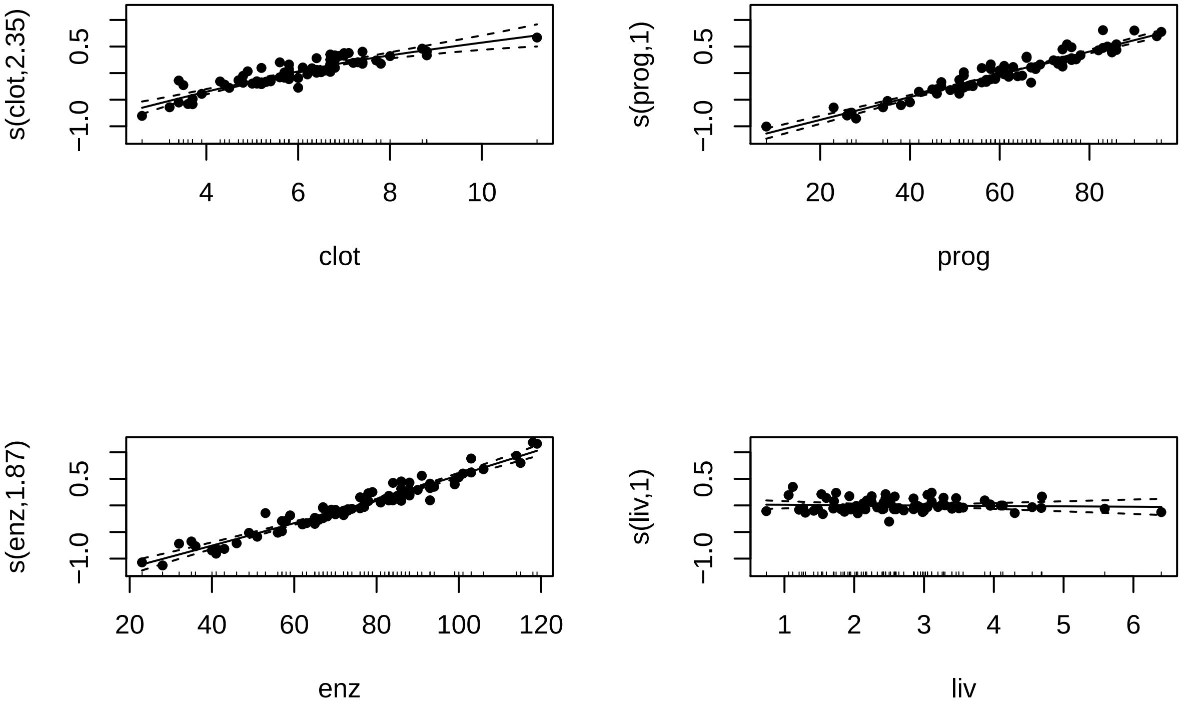
```
liverB.gam <- gam(log(time)~s(clot)+s(prog)+s(enz)+s(liv),data = liver.df)
plot(liverB.gam, residuals = T, pages = 1, pch = 20)
```



The gam plots shows that this model is a very good fit to our data.

Now, let's look at the gam plot for log(time) to check if there's any regressor we can optimise further.

```
library(mgcv)
liver1.gam <- gam(log(time)~s(clot)+s(prog)+s(enz)+s(liv),data = liver.df)
plot(liver1.gam, residuals = T, pages = 1, pch = 20)
```



The gam plots above show that prog, enz and liv have become almost linear, but clot may need to add a quadratic?

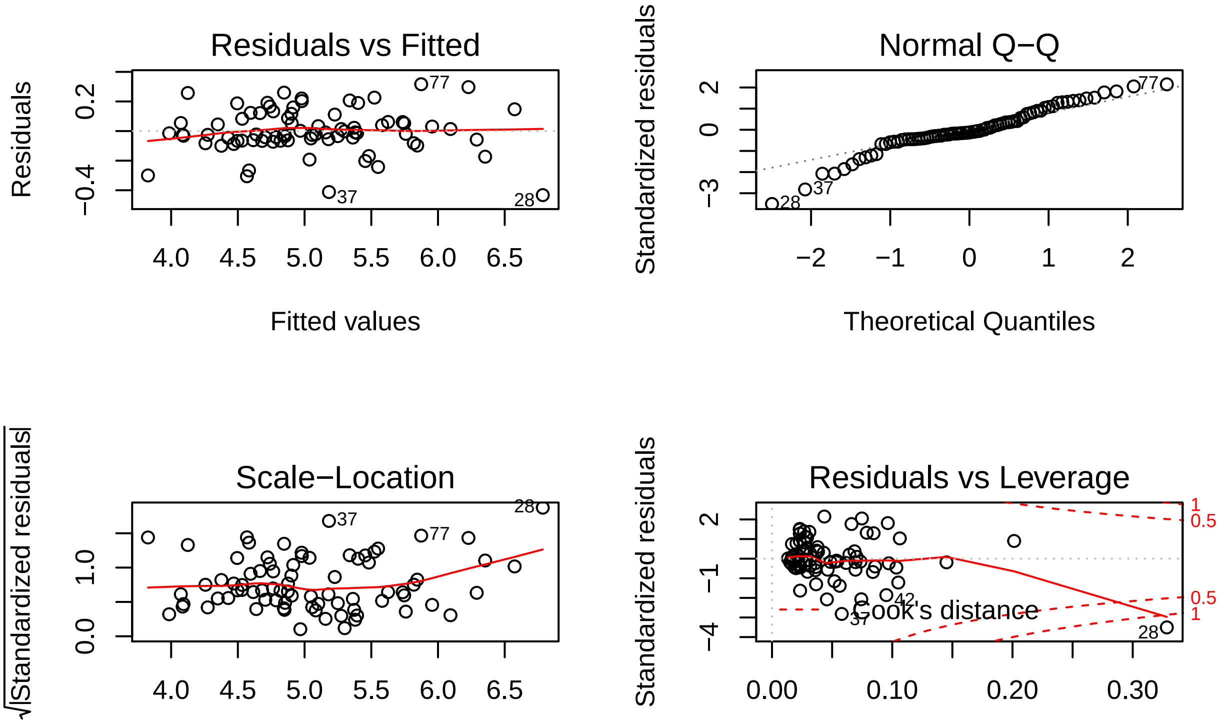
```
liverC.lm <- lm(log(time)~I(clot^2)+prog+enz,data=liver.df)
summary(liverC.lm)
```

```
##
## Call:
## lm(formula = log(time) ~ I(clot^2) + prog + enz, data = liver.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43270 -0.06392 -0.01887  0.08482  0.31637
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.7180449  0.0935106   18.37   <2e-16 ***
## I(clot^2)   0.0124301  0.0009300   13.37   <2e-16 ***
## prog        0.0207616  0.0009906   20.96   <2e-16 ***
## enz         0.0214463  0.0008062   26.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1506 on 75 degrees of freedom
## Multiple R-squared:  0.9465, Adjusted R-squared:  0.9444
## F-statistic: 442.2 on 3 and 75 DF,  p-value: < 2.2e-16
```

```
anova(liverC.lm)
```

```
## Analysis of Variance Table
##
## Response: log(time)
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## I(clot^2)  1  3.1828  3.1828  140.40 < 2.2e-16 ***
## prog       1 10.8508 10.8508  478.64 < 2.2e-16 ***
## enz        1 16.0432 16.0432  707.69 < 2.2e-16 ***
## Residuals 75  1.7002  0.0227
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow=c(2,2))
plot(liverC.lm)
```



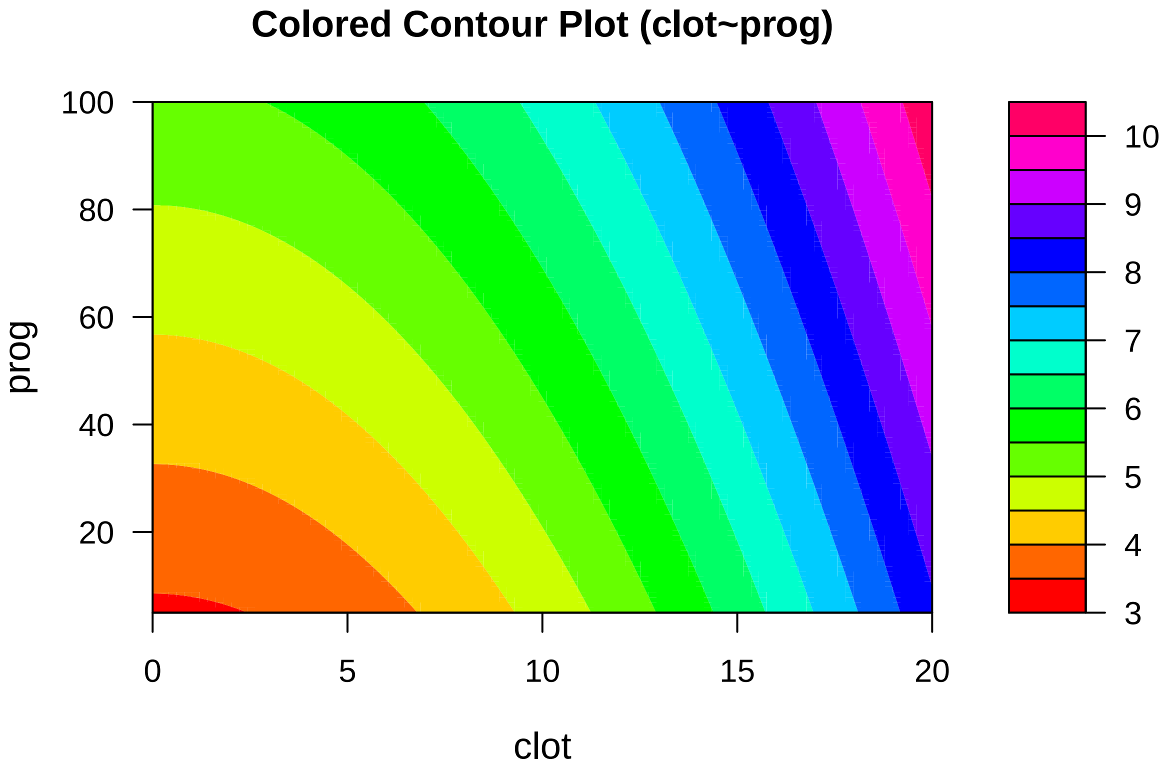
It looks like this model is the best! The curves become flat and Q-Q is very close to linear. In this case, we will say liverC.lm with log(time) and clotˆ2 should be a good model to fit.

(d) Use your chosen model to discuss the relationship between survival time and each of the explanatory variables. Some well-chosen plots may be helpful.

We are going to draw coloured contour plots to show the relationships between survival time and clot, prog and enz. Because we can only display two variables at the same time in a plot, we are going to control one variable by set it into the mean value and perform three times.

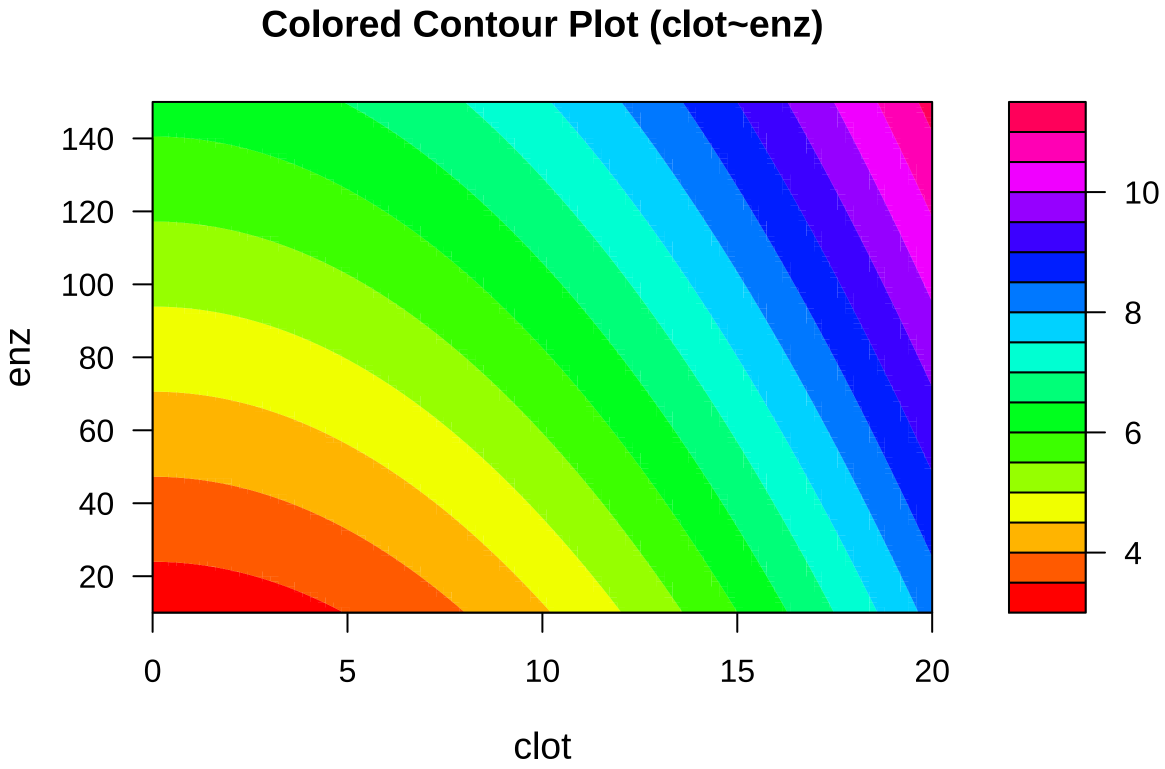
The first one is how clot and prog influence the time. The relationship may be not linear before clot and prog reaches a certain big point.

```
clot.seq <- seq(0,20,length=100)
prog.seq <- seq(5,100,length=100)
time.pred <- outer(X=clot.seq,Y=prog.seq,
                   FUN=function(a,b){predict(liverC.lm,newdata=data.frame
                                             (clot=a,prog=b,enz=mean(liver.df$enz)),type="response")})
filled.contour(clot.seq,prog.seq,time.pred,color.palette = rainbow,
               xlab="clot",ylab="prog",main="Colored Contour Plot (clot~prog)",cex.lab=1.2)
```



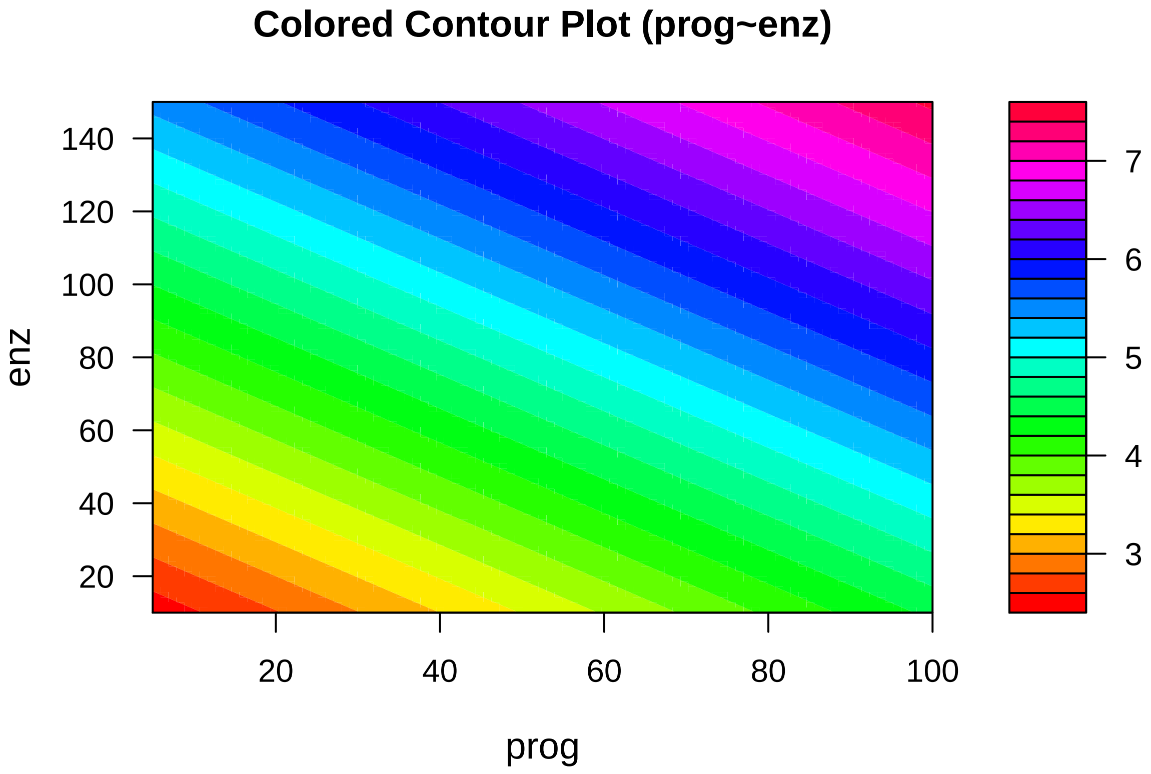
Very similar situation to clot and prog in this plot.

```
clot.seq <- seq(0,20,length=100)
enz.seq <- seq(10,150,length=100)
time.pred <- outer(X=clot.seq,Y=enz.seq,
                   FUN=function(a,b){predict(liverC.lm,newdata=data.frame
                                             (clot=a,enz=b,prog=mean(liver.df$prog)),type="response")})
filled.contour(clot.seq,enz.seq,time.pred,color.palette = rainbow,
               xlab="clot",ylab="enz",main="Colored Contour Plot (clot~enz)",cex.lab=1.2)
```



It is very interesting to see that prog and enz have a linear relationship with survival time in the plot below.

```
prog.seq <- seq(5,100,length=100)
enz.seq <- seq(10,150,length=100)
time.pred <- outer(X=prog.seq,Y=enz.seq,FUN=function(a,b){predict(liverC.lm,newdata=data.frame(prog=a,e
filled.contour(prog.seq,enz.seq,time.pred,color.palette = rainbow,xlab="prog",ylab="enz",main="Colored
```



2. The data for this question comes from a study that investigated the effect of insulin on laboratory mice. The response was whether or not the mice had convulsions when given insulin. We are interested in modelling how the proportion of mice with convulsions differs for a new preparation method compared to the standard method.

(a) Create a data frame in R that contains the information in the table in a form that is suitable for fitting a logistic regression model for grouped data.

The dataframe is shown below with method a factor, 0 for standard and 1 for new.

```
method <- c(0,0,0,0,0,0,0,0,0,1,1,1,1,1)
dose <- c(3.4,5.2,7.0,8.5,10.5,13.0,18.0,21.0,28.0,6.5,10.0,14.0,21.5,29.0)
conv <- c(0,5,11,14,18,21,23,30,27,2,10,18,21,27)
total <- c(33,32,38,37,40,37,31,37,30,40,30,40,35,37)
insulin.df <- data.frame(method, dose, conv, total)
insulin.df$method <- as.factor(insulin.df$method)
str(insulin.df)
```

```
## 'data.frame':    14 obs. of  4 variables:
##  $ method: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
##  $ dose  : num  3.4 5.2 7 8.5 10.5 13 18 21 28 6.5 ...
##  $ conv  : num  0 5 11 14 18 21 23 30 27 2 ...
##  $ total : num  33 32 38 37 40 37 31 37 30 40 ...
```

```
print(insulin.df)
```

```
##    method dose conv total
## 1       0  3.4    0    33
## 2       0  5.2    5    32
## 3       0  7.0   11    38
## 4       0  8.5   14    37
## 5       0 10.5   18    40
## 6       0 13.0   21    37
## 7       0 18.0   23    31
## 8       0 21.0   30    37
## 9       0 28.0   27    30
## 10      1  6.5    2    40
## 11      1 10.0   10    30
## 12      1 14.0   18    40
## 13      1 21.5   21    35
## 14      1 29.0   27    37
```

(b) First fit the model that just uses dose and preparation method as regressors. Assess the suitability of this model.

Start fitting with the simple logistic regression:

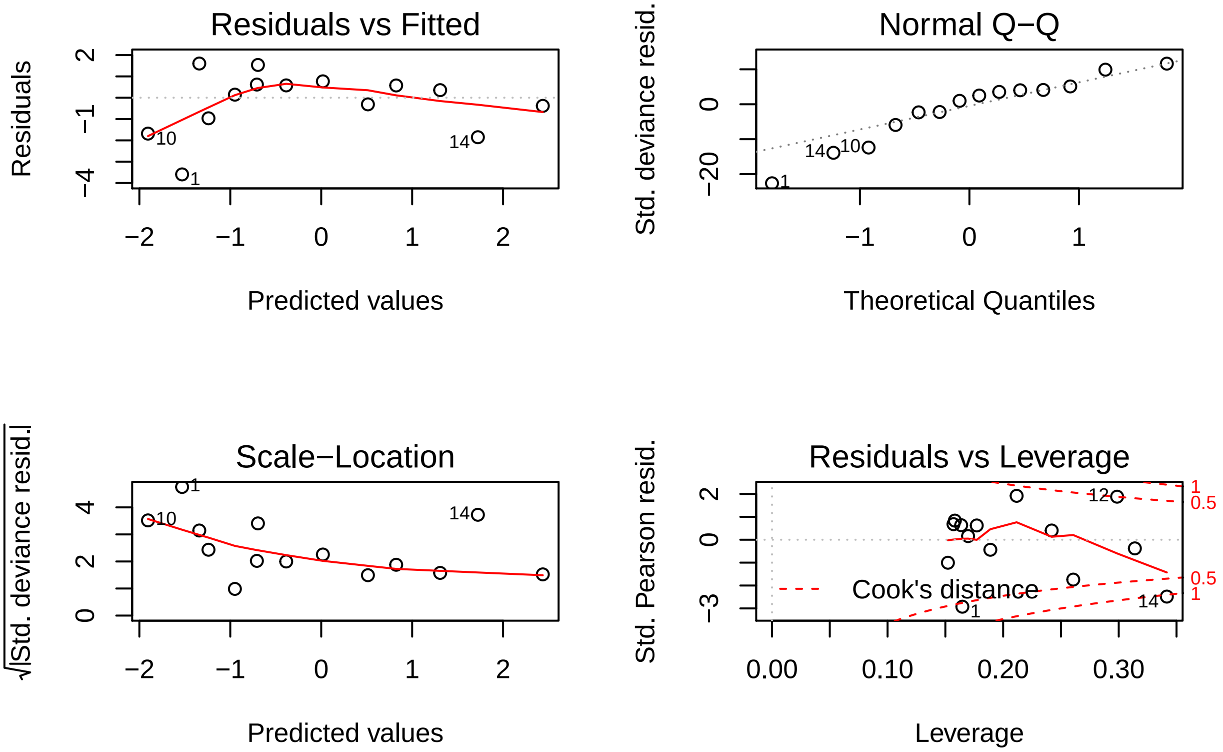
```
insulin.glm <- glm(cbind(conv, total-conv)~method+dose, family = binomial, data = insulin.df)
summary(insulin.glm)
```

```
##
## Call:
## glm(formula = cbind(conv, total - conv) ~ method + dose, family = binomial,
##     data = insulin.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5967  -0.8187   0.2498   0.6074   1.6034
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.07827    0.22688  -9.160  < 2e-16 ***
## method1     -0.87525    0.23393  -3.742 0.000183 ***
## dose         0.16126    0.01601  10.069  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 166.834  on 13  degrees of freedom
## Residual deviance:  27.098  on 11  degrees of freedom
## AIC: 80.951
##
## Number of Fisher Scoring iterations: 4
```

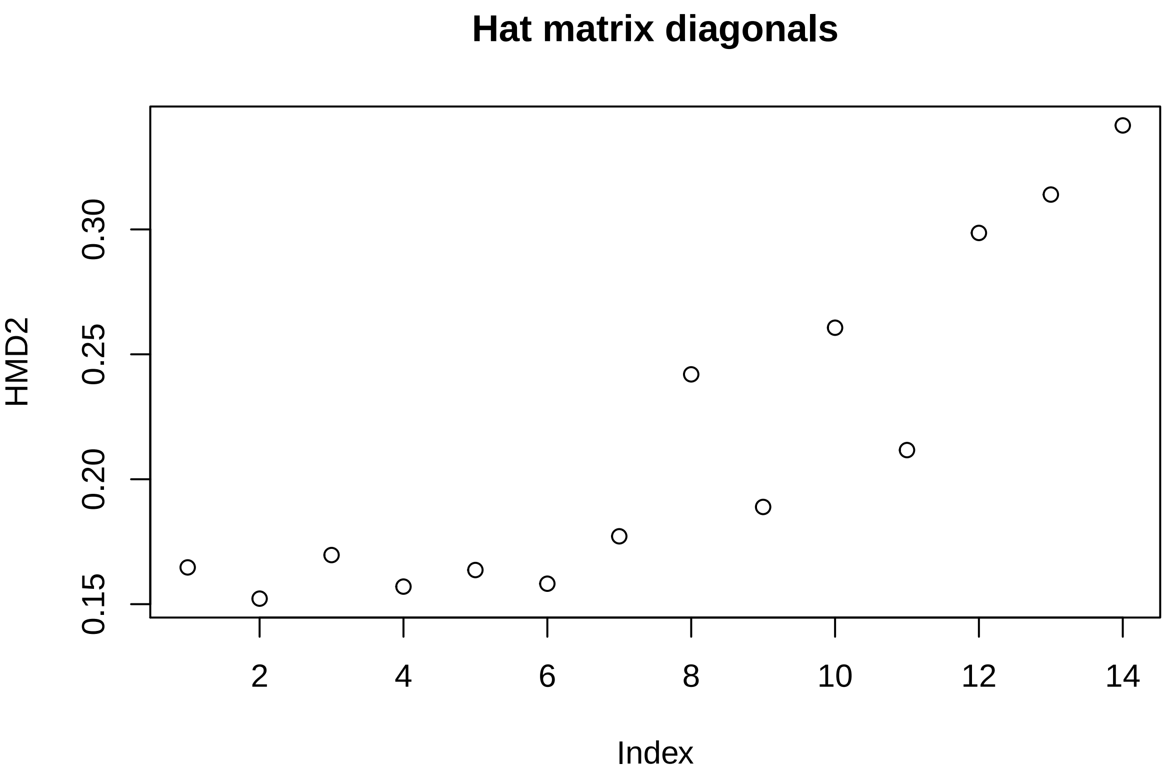
```
anova(insulin.glm)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: cbind(conv, total - conv)
##
## Terms added sequentially (first to last)
##
##
##        Df Deviance Resid. Df Resid. Dev
## NULL                      13    166.834
## method  1      0.92          12     165.914
## dose    1    138.81          11      27.098
```

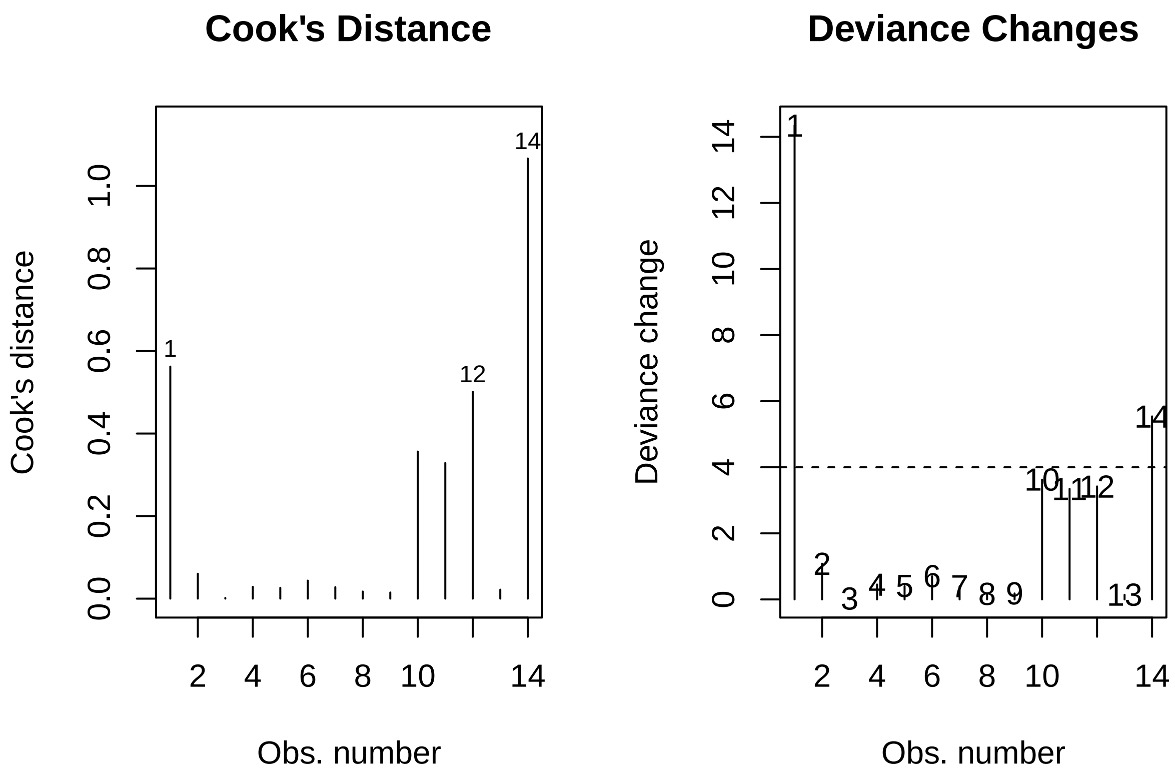
```
par(mfrow=c(2,2))
plot(insulin.glm)
```



```
par(mfrow=c(1,1))
HMD2<-hatvalues(insulin.glm)
plot(HMD2,main="Hat matrix diagonals")
```



```
dev1.r<-residuals(insulin.glm,type="deviance")
pear1.r<-residuals(insulin.glm,type="pearson")
Dev1.change<-dev1.r^2 + pear1.r^2*HMD2/(1-HMD2)
par(mfrow=c(1,2))
plot(insulin.glm,which=4,main="Cook's Distance",caption=" ")
plot(Dev1.change,ylab="Deviance change", xlab="Obs. number",type="h", main="Deviance Changes")
text(Dev1.change)
bigdev=4
abline(h=bigdev, lty=2)
```

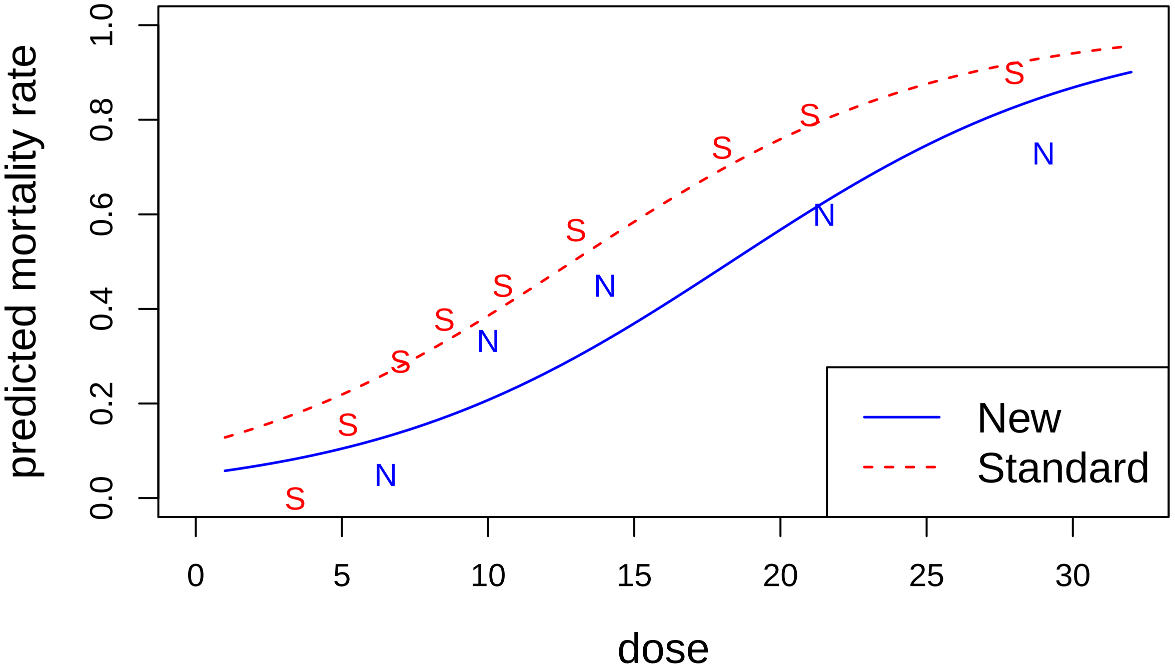


```
1-pchisq(27.098, 11)
```

```
## [1] 0.004441136
```

The result does not look too bad! Let's plot a graph to see how this model actually fit the predictions:

```
ds<-seq(1,32,length=200)
newF.df<-data.frame(method=1,dose=ds)
newM.df<-data.frame(method=0,dose=ds)
newF.df$method <- as.factor(newF.df$method)
newM.df$method <- as.factor(newM.df$method)
estsF<- predict(insulin.glm,newF.df,type="response")
estsM<- predict(insulin.glm,newM.df,type="response")
plot(c(0,32),c(0,1),xlab="dose",
     ylab="predicted mortality rate",type="n",cex.lab=1.3)
lines(ds,estsF,lty=1,col="blue",lwd=1.3)
lines(ds,estsM,lty=2,col="red",lwd=1.3)
points(insulin.df$dose,insulin.df$conv/total,
       pch=c(rep("S",9),rep("N",5)), col=c(rep("red",9),rep("blue",5)))
legend("bottomright",legend=c("New","Standard"),
       lty=1:2,col=c("blue","red"),lwd=1.3,cex=1.3)
```



The fitting was good before does reaches 15, after 15 it's a little bit biased.

(c) Explore the possibility of improving the model. Summarise your model building approach and include key pieces of evidence. Provide a full set of diagnostics for your final model.

Let's try log(dose) first:

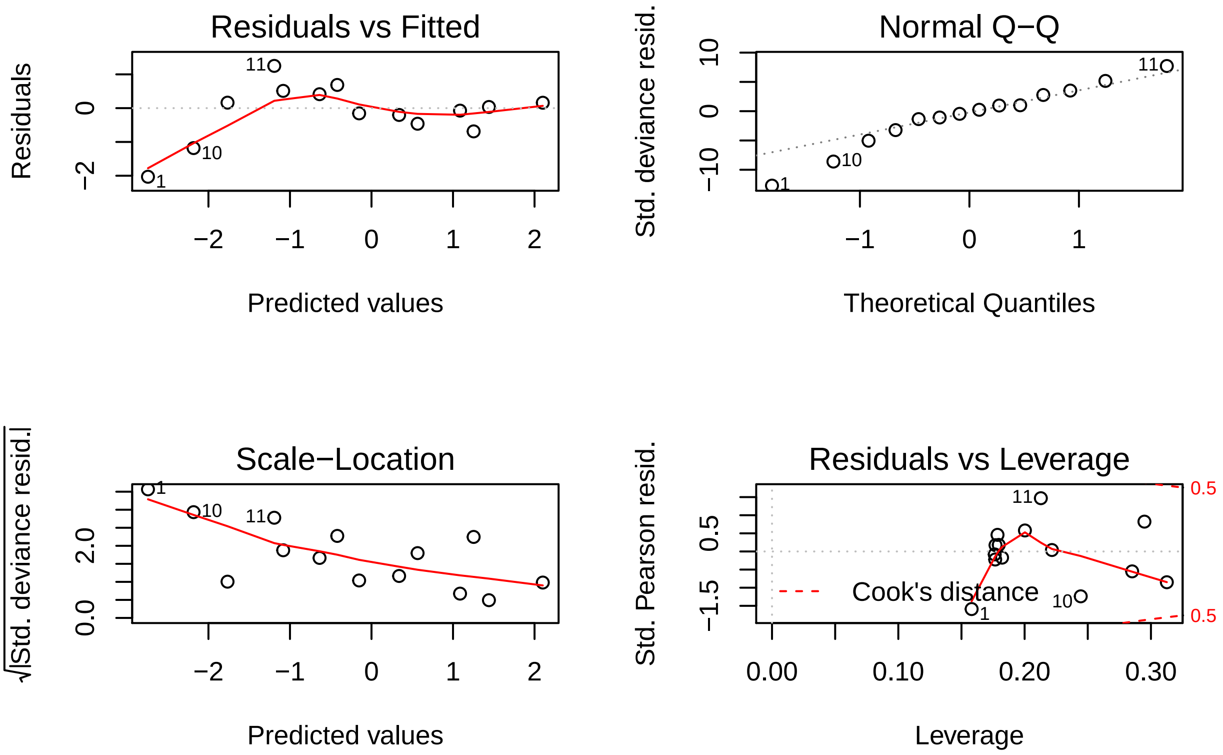
```r
loginsulin.glm <- glm(cbind(conv, total-conv)~method+log(dose), family = binomial, data = insulin.df)
summary(loginsulin.glm)
```

```
##
## Call:
## glm(formula = cbind(conv, total - conv) ~ method + log(dose),
##     family = binomial, data = insulin.df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.03026  -0.39653  -0.01947   0.34974   1.24987
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.5531     0.5427  -10.23  < 2e-16 ***
## method1      -0.9290     0.2334   -3.98 6.89e-05 ***
## log(dose)     2.2972     0.2196   10.46  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 166.8335  on 13  degrees of freedom
## Residual deviance:   8.7912  on 11  degrees of freedom
## AIC: 62.644
##
## Number of Fisher Scoring iterations: 4
```

```r
anova(loginsulin.glm)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: cbind(conv, total - conv)
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                         13    166.834
## method     1     0.92        12    165.914
## log(dose)  1   157.12        11      8.791
```

```r
par(mfrow=c(2,2))
plot(loginsulin.glm)
```



```r
1-pchisq(8.7912, 11)
```

```
## [1] 0.6411586
```

Lack of fit test shows that this model is actually better than the naive model in (b).

Let's try to square dose now see if it improves the result:

```r
insulin2.glm <- glm(cbind(conv, total-conv)~method+I(dose^2), family = binomial, data = insulin.df)
summary(insulin2.glm)
```

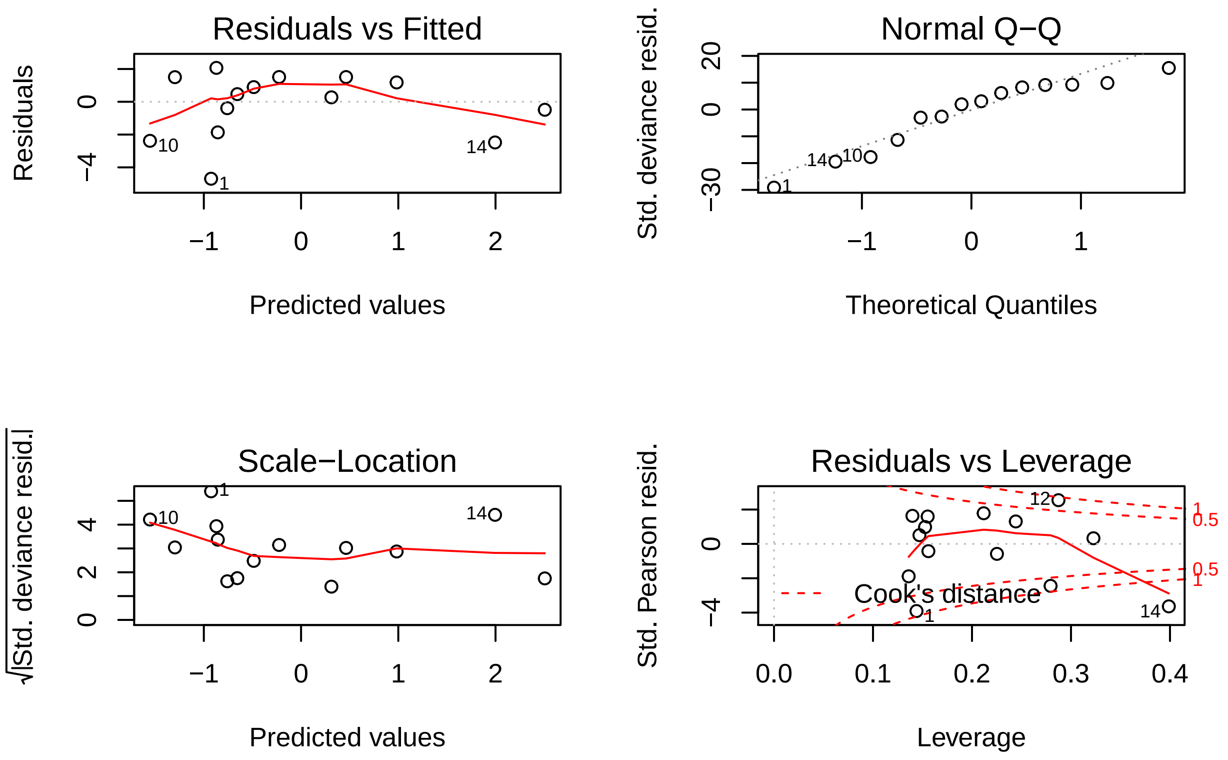
```
##
## Call:
```

```
## glm(formula = cbind(conv, total - conv) ~ method + I(dose^2),
##     family = binomial, data = insulin.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.6945  -1.5208   0.3689   1.4202   2.0672
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.976648   0.149627  -6.527  6.7e-11 ***
## method1     -0.765063   0.226698  -3.375 0.000739 ***
## I(dose^2)    0.004443   0.000495   8.976  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 166.83  on 13  degrees of freedom
## Residual deviance:  51.30  on 11  degrees of freedom
## AIC: 105.15
##
## Number of Fisher Scoring iterations: 4
```

```
anova(insulin2.glm)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: cbind(conv, total - conv)
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                         13     166.83
## method     1     0.92        12     165.91
## I(dose^2)  1   114.61        11      51.30
```

```
par(mfrow=c(2,2))
plot(insulin2.glm)
```

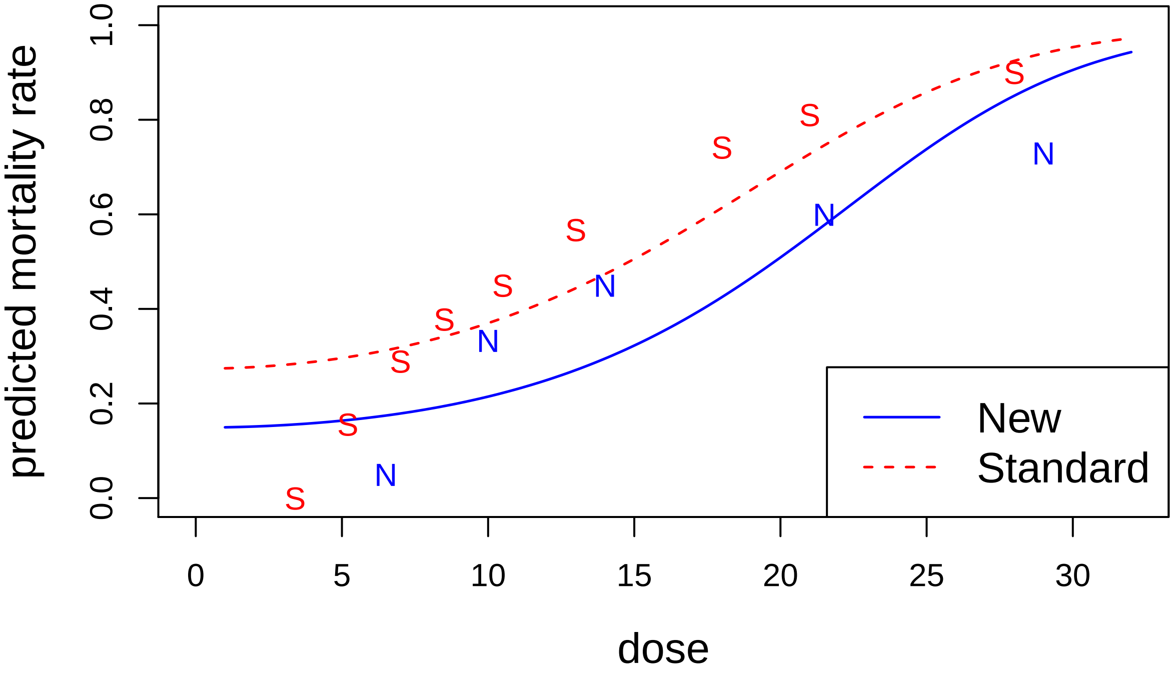


```
1-pchisq(51.30, 11)
```

```
## [1] 3.649498e-07
```

Not at all, dose^2 does not help in this case. How it fits the prediction? The plot below shows that the model cannot fit well to the data.

```
ds<-seq(1,32,length=200)
newN.df<-data.frame(method=1,dose=ds)
newS.df<-data.frame(method=0,dose=ds)
newN.df$method <- as.factor(newN.df$method)
newS.df$method <- as.factor(newS.df$method)
estsN<- predict(insulin2.glm,newN.df,type="response")
estsS<- predict(insulin2.glm,newS.df,type="response")
plot(c(0,32),c(0,1),xlab="dose",
     ylab="predicted mortality rate",type="n",cex.lab=1.3)
lines(ds,estsN,lty=1,col="blue",lwd=1.3)
lines(ds,estsS,lty=2,col="red",lwd=1.3)
points(insulin.df$dose,insulin.df$conv/total,
       pch=c(rep("S",9),rep("N",5)), col=c(rep("red",9),rep("blue",5)))
legend("bottomright",legend=c("New","Standard"),
       lty=1:2,col=c("blue","red"),lwd=1.3,cex=1.3)
```



But, how about the interactions? Let's check if there's any?

```
loginsulinI.glm <- glm(cbind(conv, total-conv)~method*log(dose), family = binomial, data = insulin.df)
summary(loginsulinI.glm)
```

```
##
## Call:
## glm(formula = cbind(conv, total - conv) ~ method * log(dose),
##     family = binomial, data = insulin.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9190  -0.2881  -0.1297   0.4416   1.0651
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -5.7907     0.6839  -8.467   <2e-16 ***
## method1             -0.2170     1.2077  -0.180    0.857
## log(dose)            2.3964     0.2799   8.561   <2e-16 ***
## method1:log(dose)  -0.2723     0.4544  -0.599    0.549
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 166.8335  on 13  degrees of freedom
## Residual deviance:   8.4351  on 10  degrees of freedom
## AIC: 64.287
##
## Number of Fisher Scoring iterations: 4
```

```
anova(loginsulinI.glm, test = "Chisq")
```

```
## Analysis of Deviance Table
```

```
##
## Model: binomial, link: logit
##
## Response: cbind(conv, total - conv)
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                               13    166.834
## method           1    0.920        12    165.914   0.3375
## log(dose)        1  157.122        11      8.791   <2e-16 ***
## method:log(dose) 1    0.356        10      8.435   0.5507
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

No evidence here that the interaction is needed in the model.

```
1-pchisq(8.4351, 10)
```

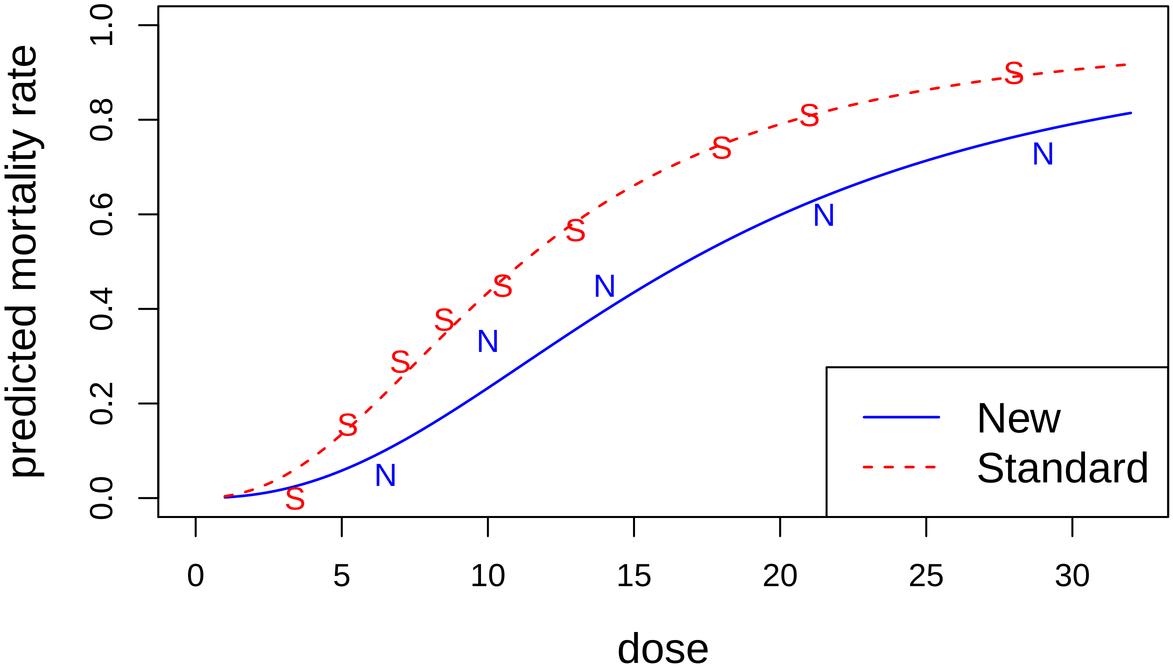
```
## [1] 0.5864164
```

```
str(insulin.df)
```

```
## 'data.frame':    14 obs. of  4 variables:
##  $ method: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
##  $ dose  : num  3.4 5.2 7 8.5 10.5 13 18 21 28 6.5 ...
##  $ conv  : num  0 5 11 14 18 21 23 30 27 2 ...
##  $ total : num  33 32 38 37 40 37 31 37 30 40 ...
```

So now we have a winner! The log(dose) is the best model so far for this case. Let's visualise this model to see how it actually fits:

```
ds<-seq(1,32,length=200)
newN.df<-data.frame(method=1,dose=ds)
newS.df<-data.frame(method=0,dose=ds)
newN.df$method <- as.factor(newN.df$method)
newS.df$method <- as.factor(newS.df$method)
estsN<- predict(loginsulin.glm,newN.df,type="response")
estsS<- predict(loginsulin.glm,newS.df,type="response")
plot(c(0,32),c(0,1),xlab="dose",
     ylab="predicted mortality rate",type="n",cex.lab=1.3)
lines(ds,estsN,lty=1,col="blue",lwd=1.3)
lines(ds,estsS,lty=2,col="red",lwd=1.3)
points(insulin.df$dose,insulin.df$conv/total,
       pch=c(rep("S",9),rep("N",5)), col=c(rep("red",9),rep("blue",5)))
legend("bottomright",legend=c("New","Standard"),
       lty=1:2,col=c("blue","red"),lwd=1.3,cex=1.3)
```



(d) Use your chosen model to compare the probability of convulsions for the new preparation method to that for the standard method. A suitable plot may be helpful in this regard.

Like the plot we drew above, it actually fits not too bad. before dose reaches 15 the fitting was perfect. It is curious to see that the models starts to cross-fit with the other variable after dose reaches 15. Generally, it is still a good model.