Block 4: Regression

Huizhi Lin (881125518130) & Jan Orsel (080695690010)

November 20, 2017

In this project we tried to analyse how gene expression levels in specific metabolic pathways are related to each other. We did this with the help of linear regression models where we regress gene expression levels with each other. Unlike the previous project we now used the breast cancer data as shown in the block of code below. For this project the difference in normal and tumour tissue was not taken into account.

```
breast.data <- read.table("get_normal_vs_tumor2_RAW_Breast.out", sep=' ', hea
der=TRUE)
```

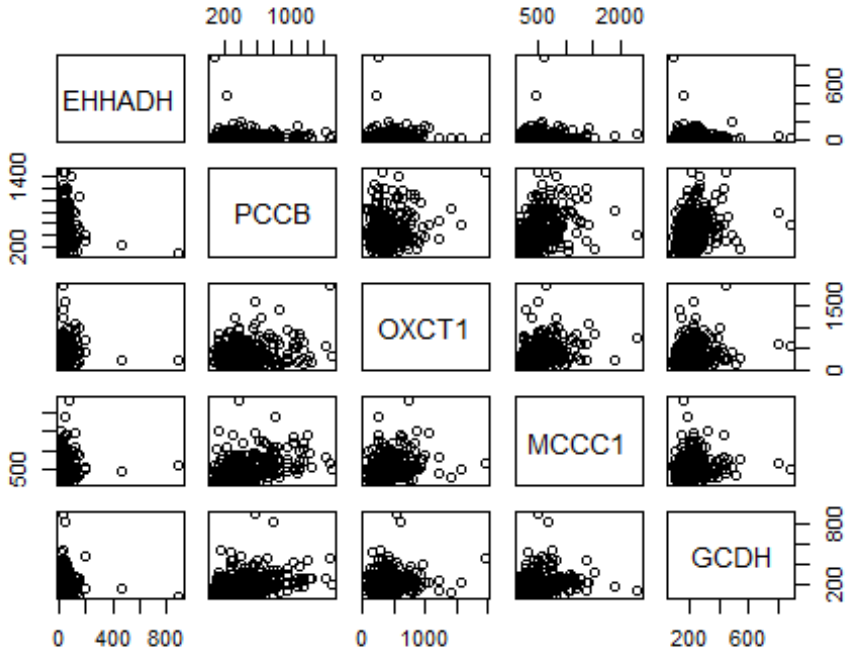
Task A

For the first assignment a new dataframe was created that only contained the genes 'EHHADH, PCCB, OXCT1, MCCC1 and GCDH'. This deleted the header names so we had to add them manually again.

```
small.data <- as.data.frame(cbind(breast.data$EHHADH,breast.data$PCCB,breast.
data$OXCT1,breast.data$MCCC1,breast.data$GCDH))
colnames(small.data) <- c('EHHADH','PCCB','OXCT1','MCCC1','GCDH')
```

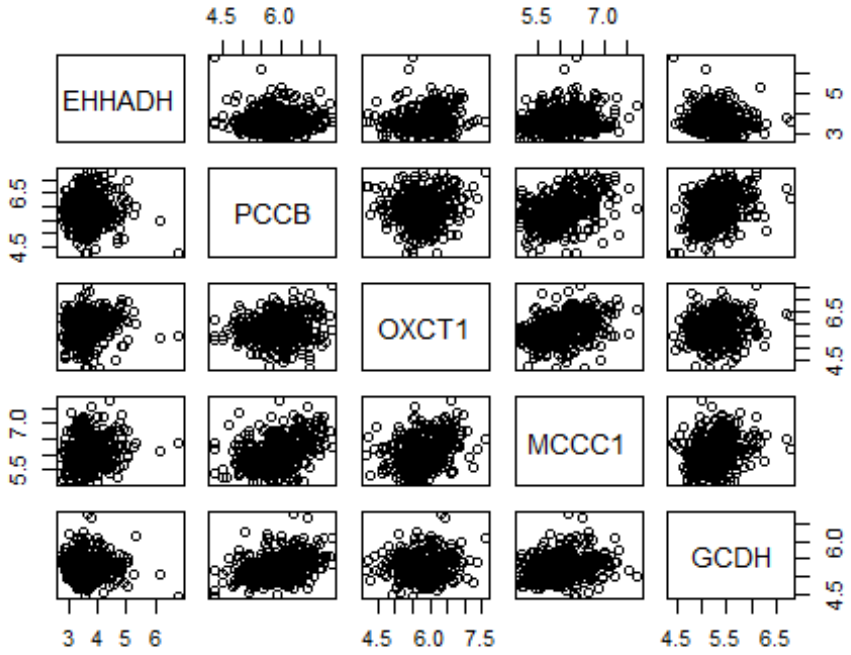
The data of these was visualised with the pairs function to give a first impression on possible correlation. The pairs function created a mirrored matrix scatter plot.

```
pairs(small.data)
```



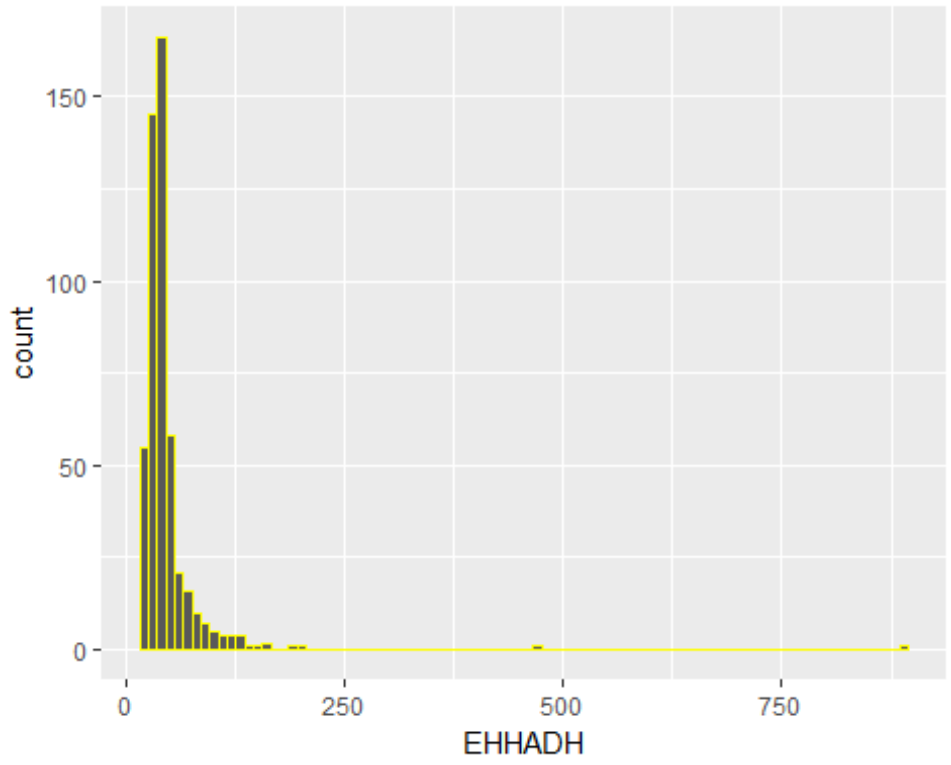
At first impression the plot does not show significant correlations between these five genes. Then we log-transform the data and visualised it.

```
log.data <- log(small.data)
pairs(log.data)
```

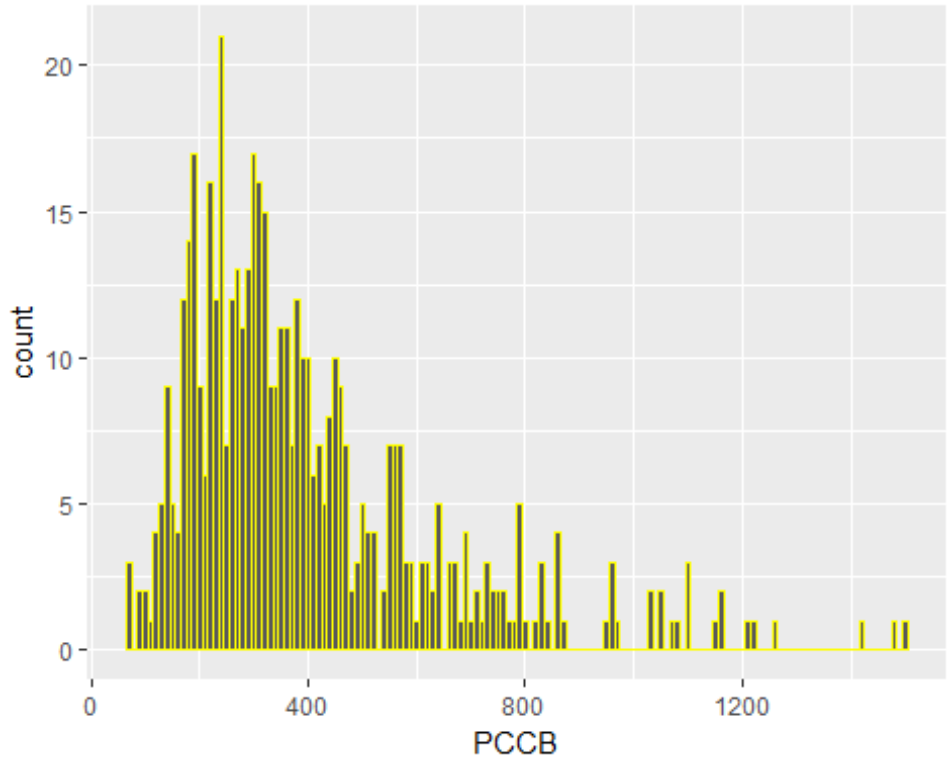


We created histograms using ggplot to visualise if the data was normally distributed. This was done for all the 5 genes of the krebs cycle.

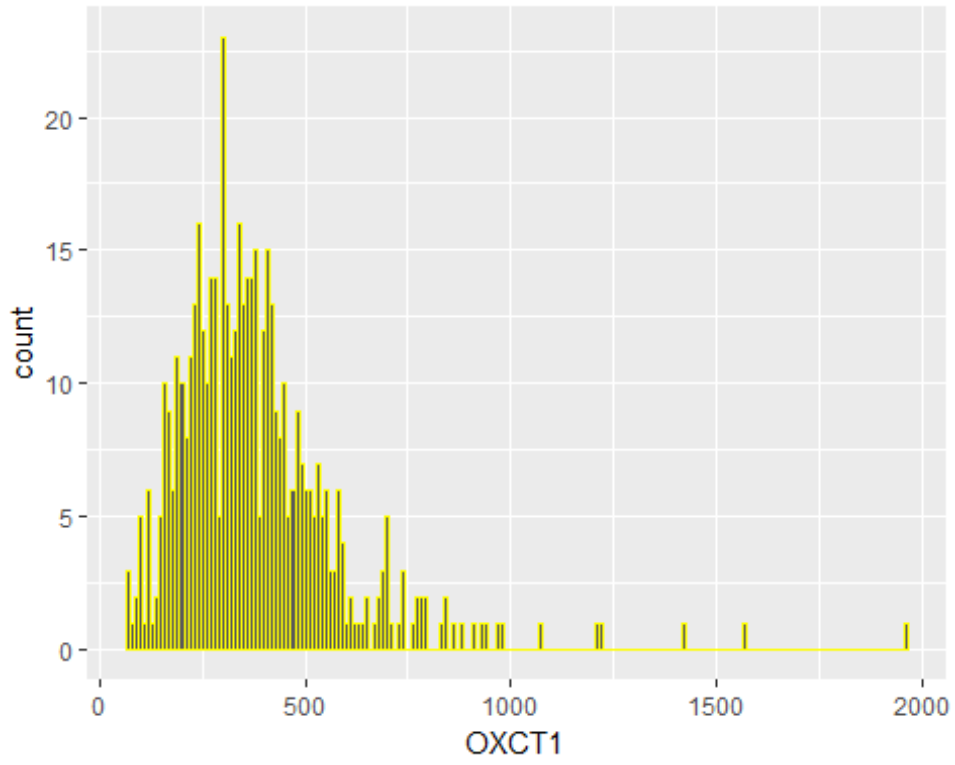
```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.4.2

pl.EHHADH <- ggplot(data=small.data,aes(x=small.data$EHHADH)) + geom_histogra
m(binwidth=10,color='yellow') + xlab("EHHADH")
pl.PCCB <- ggplot(data=small.data,aes(x=small.data$PCCB)) + geom_histogram(bi
nwidth=10,color='yellow') + xlab("PCCB")
pl.OXCT1 <- ggplot(data=small.data,aes(x=small.data$OXCT1)) + geom_histogram(
binwidth=10,color='yellow') + xlab("OXCT1")
pl.MCCC1 <- ggplot(data=small.data,aes(x=small.data$MCCC1)) + geom_histogram(
binwidth=10,color='yellow') + xlab("MCCC1")
pl.GCDH <- ggplot(data=small.data,aes(x=small.data$GCDH)) + geom_histogram(bi
nwidth=10,color='yellow') + xlab("GCDH")
print(pl.EHHADH)
```



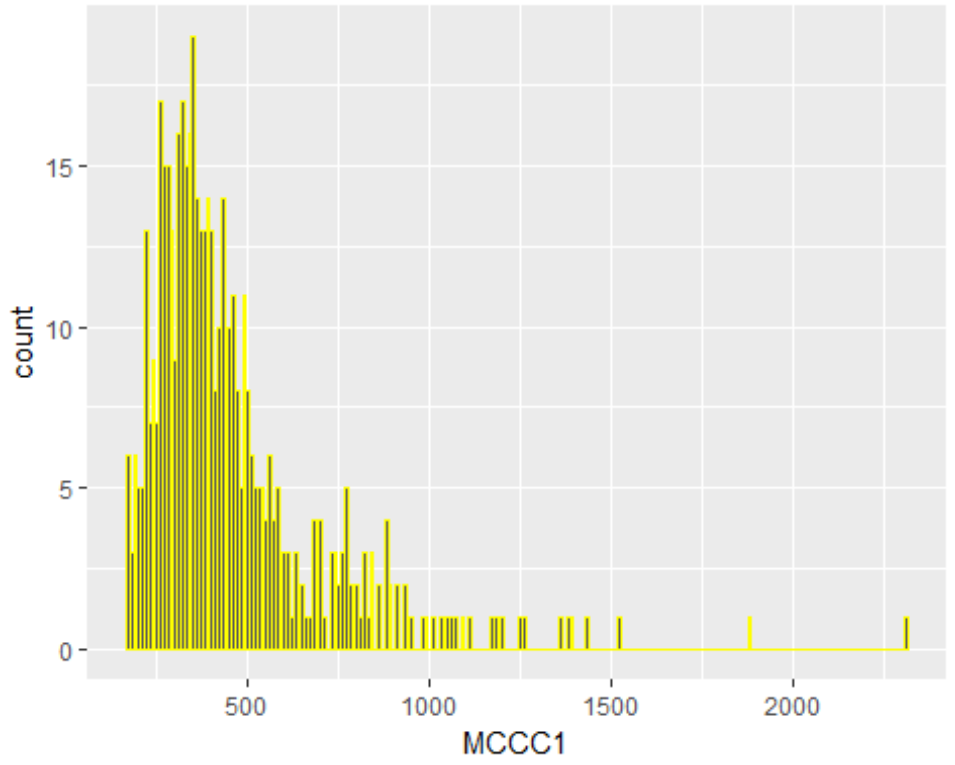

```
print(pl.PCCB)
```



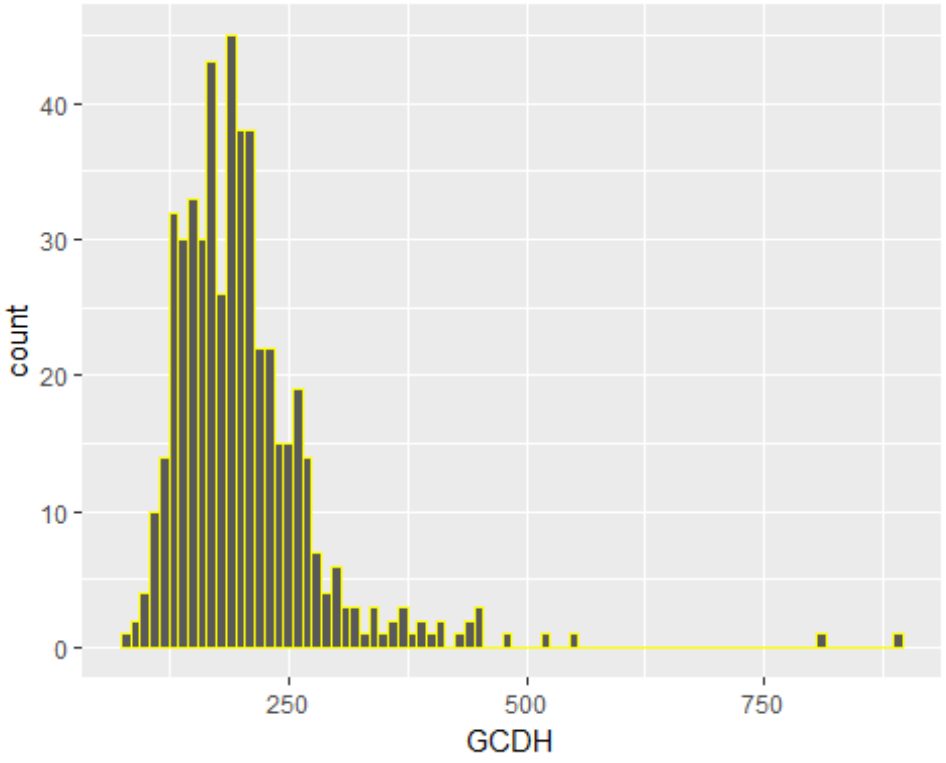

```
print(pl.OXCT1)
```




```
print(pl.MCCC1)
```

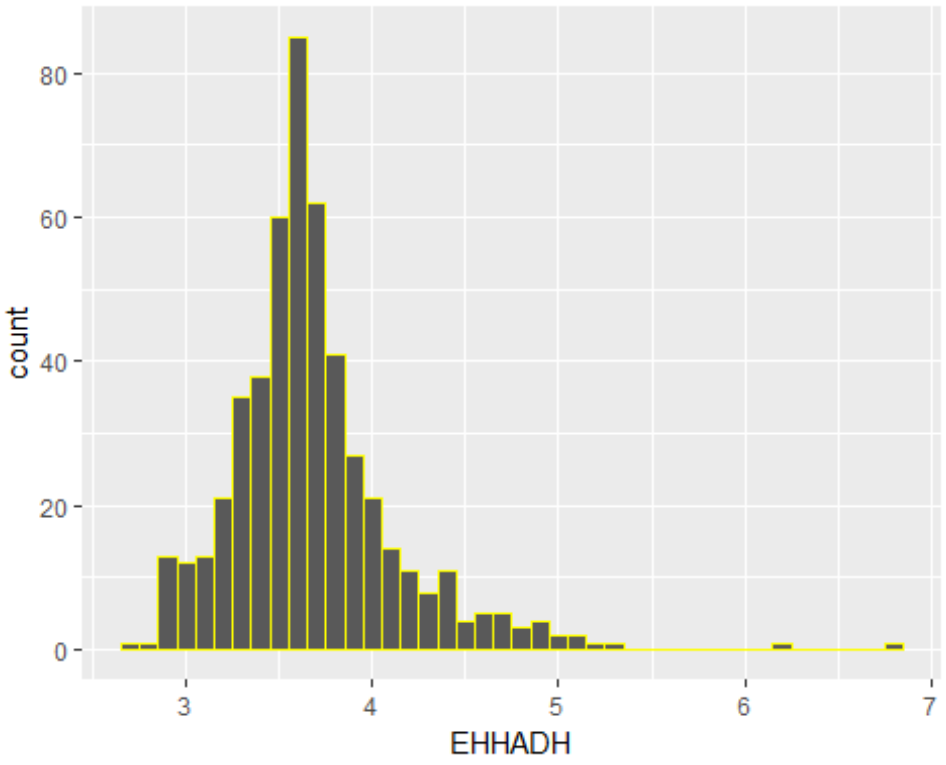



```
print(pl.GCDH)
```

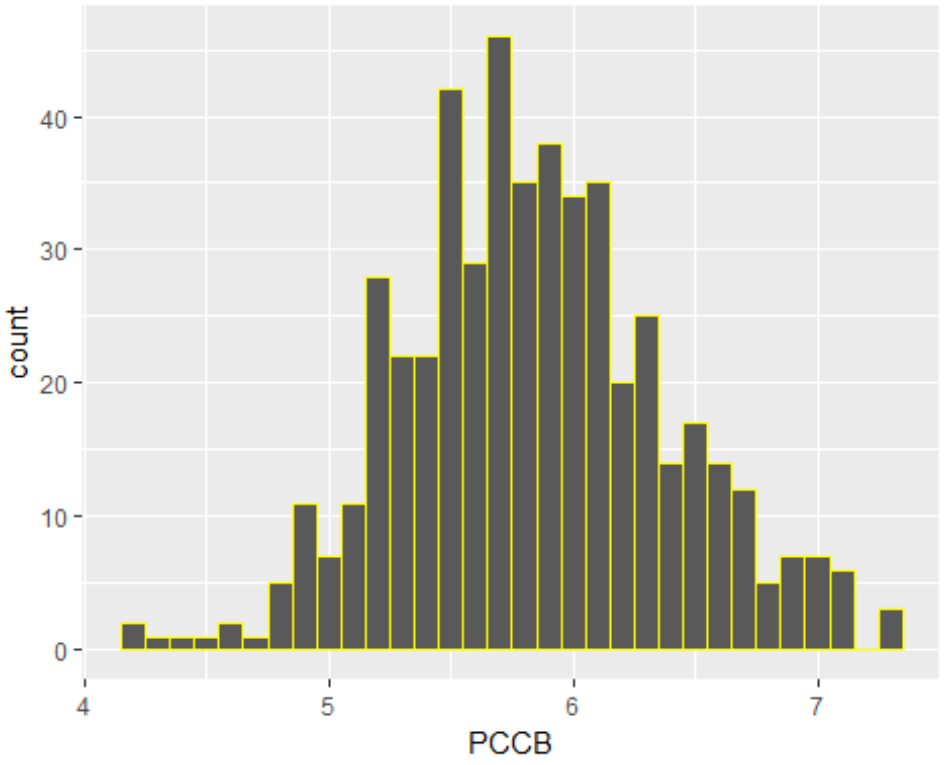


Based on these histograms we conclude that our variables are not normally distributed. Therefore we visualised them after the log-transformation to see if they had become normally distributed. The code for creating histograms on the log-transformed data is shown below.

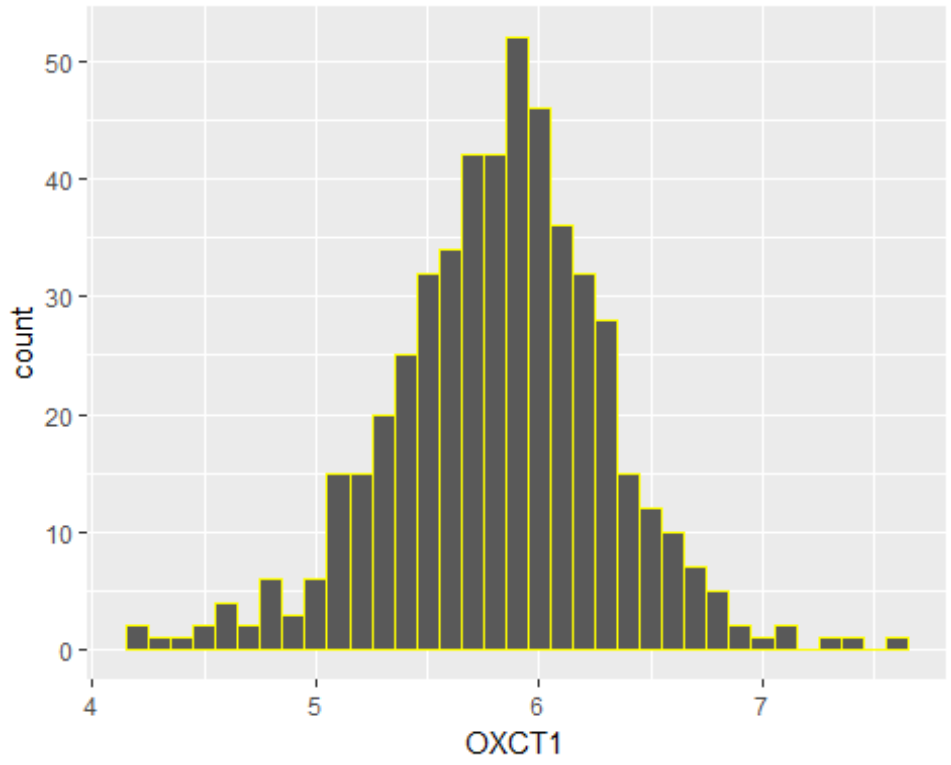
```
library(ggplot2)
pllg.EHHADH <- ggplot(data=log.data,aes(x=log.data$EHHADH)) + geom_histogram(
binwidth=0.1,color='yellow') + xlab("EHHADH")
pllg.PCCB <- ggplot(data=log.data,aes(x=log.data$PCCB)) + geom_histogram(binw
idth=0.1,color='yellow') + xlab("PCCB")
pllg.OXCT1 <- ggplot(data=log.data,aes(x=log.data$OXCT1)) + geom_histogram(bi
nwidth=0.1,color='yellow') + xlab("OXCT1")
pllg.MCCC1 <- ggplot(data=log.data,aes(x=log.data$MCCC1)) + geom_histogram(bi
nwidth=0.1,color='yellow') + xlab("MCCC1")
pllg.GCDH <- ggplot(data=log.data,aes(x=log.data$GCDH)) + geom_histogram(binw
idth=0.1,color='yellow') + xlab("GCDH")
par(mfrow=c(1,5))
print(pllg.EHHADH)
```



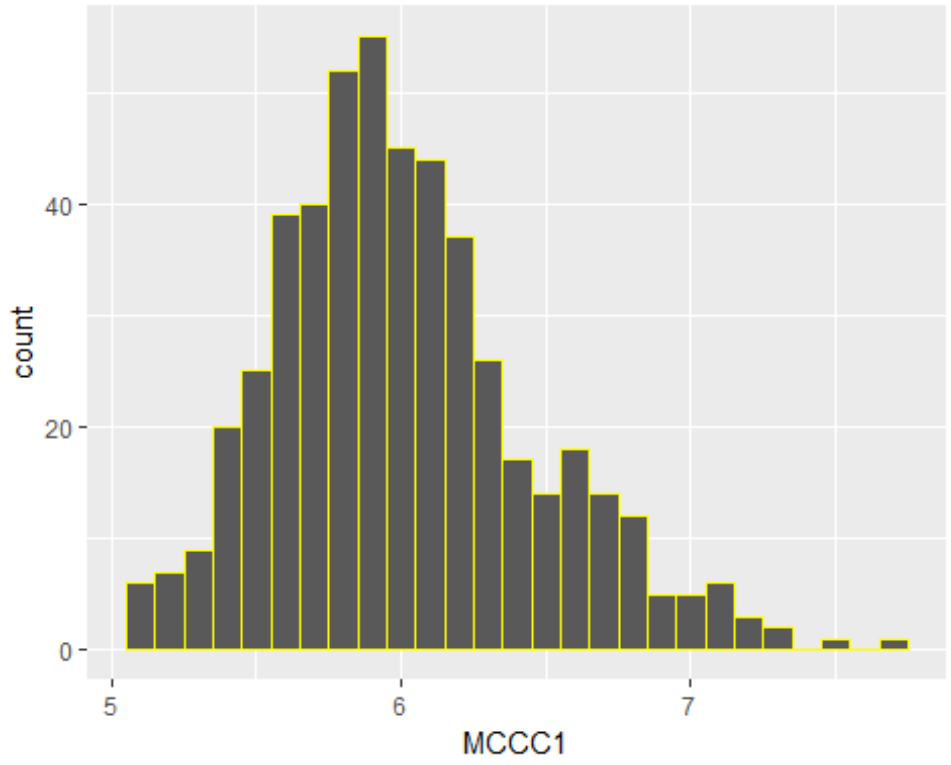

```
print(pllg.PCCB)
```



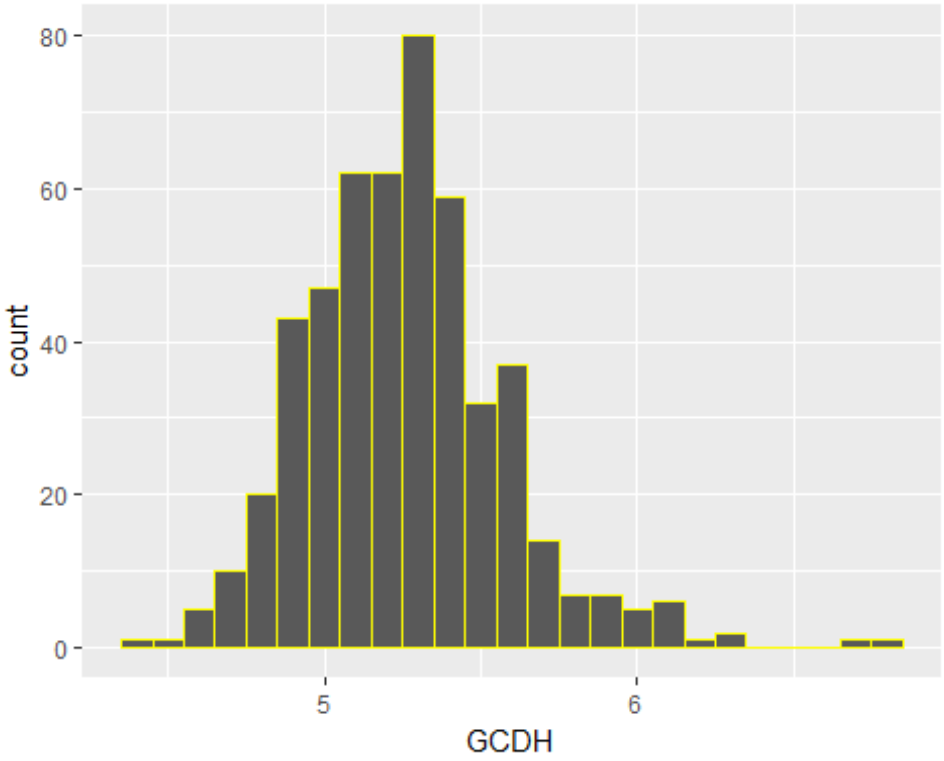

```
print(pllg.OXCT1)
```




```
print(pllg.MCCC1)
```




```
print(pllg.GCDH)
```



Though far from perfect after the log-transformation the variables seem more or less normally distributed.

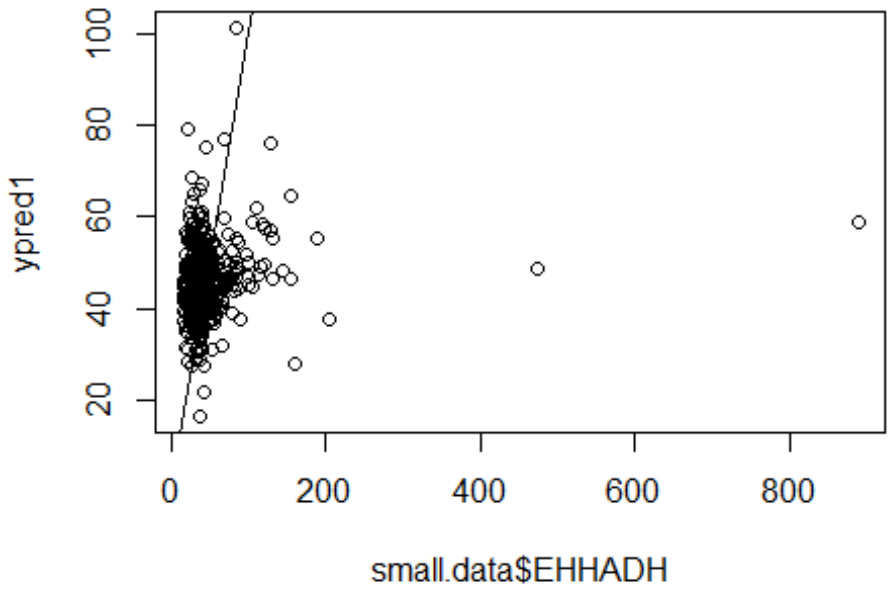
First we created multiple linear regression models that predict genes from the Krebs cycle using the other genes from this cycle with the non-transformed data. For each model a scatter plot was created. On the x-axis of these scatterplots are the observed values and the predicted values are projected on the y-axix. A diagonal line is drawn to visualise what a perfect prediction would look like.

Model 1: EHHADH as dependent variable, other genes as independent variable

```
model1 <- lm(EHHADH ~ PCCB + OXCT1 + MCCC1 + GCDH, data=small.data)
summary(model1)

##
## Call:
## lm(formula = EHHADH ~ PCCB + OXCT1 + MCCC1 + GCDH, data = small.data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -56.79 -15.21  -7.31   1.67 830.11
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 44.869229   7.133768   6.290 6.97e-10 ***
## PCCB        -0.017466   0.010437  -1.673  0.09486 .
## OXCT1        0.012827   0.011144   1.151  0.25027
```

```
## MCCC1         0.026045   0.009769   2.666  0.00792 **
## GCDH         -0.043130   0.028998  -1.487  0.13756
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48.09 on 498 degrees of freedom
## Multiple R-squared:  0.0249, Adjusted R-squared:  0.01706
## F-statistic: 3.179 on 4 and 498 DF,  p-value: 0.01352
```

```
ypred1 <- predict(model1)
plot(small.data$EHHADH,ypred1)
abline(a=0,b=1)
```

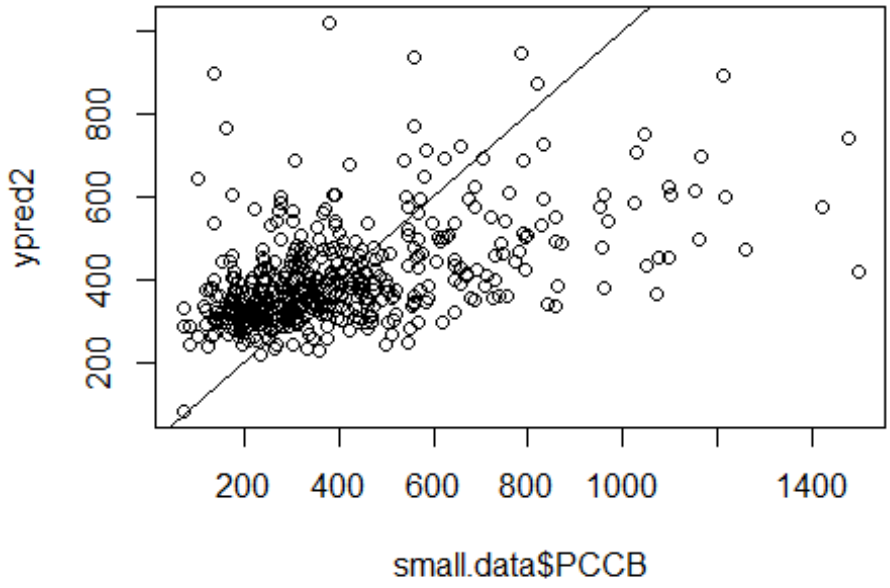


Model 2: PCCB as dependent variable, other genes as independent variable

```
model2 <- lm(PCCB ~ EHHADH + OXCT1 + MCCC1 + GCDH, data=small.data)
summary(model2)
```

```
##
## Call:
## lm(formula = PCCB ~ EHHADH + OXCT1 + MCCC1 + GCDH, data = small.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -758.53 -125.27  -36.58   85.26 1080.39
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 81.16739   31.52345   2.575   0.0103 *
## EHHADH      -0.32016    0.19132  -1.673   0.0949 .
## OXCT1        0.05580    0.04771   1.169   0.2428
## MCCC1        0.35083    0.03908   8.978  < 2e-16 ***
## GCDH         0.74515    0.11986   6.217 1.08e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 205.9 on 498 degrees of freedom
## Multiple R-squared:  0.2505, Adjusted R-squared:  0.2445
## F-statistic: 41.62 on 4 and 498 DF,  p-value: < 2.2e-16

ypred2 <- predict(model2)
plot(small.data$PCCB,ypred2)
abline(a=0,b=1)
```

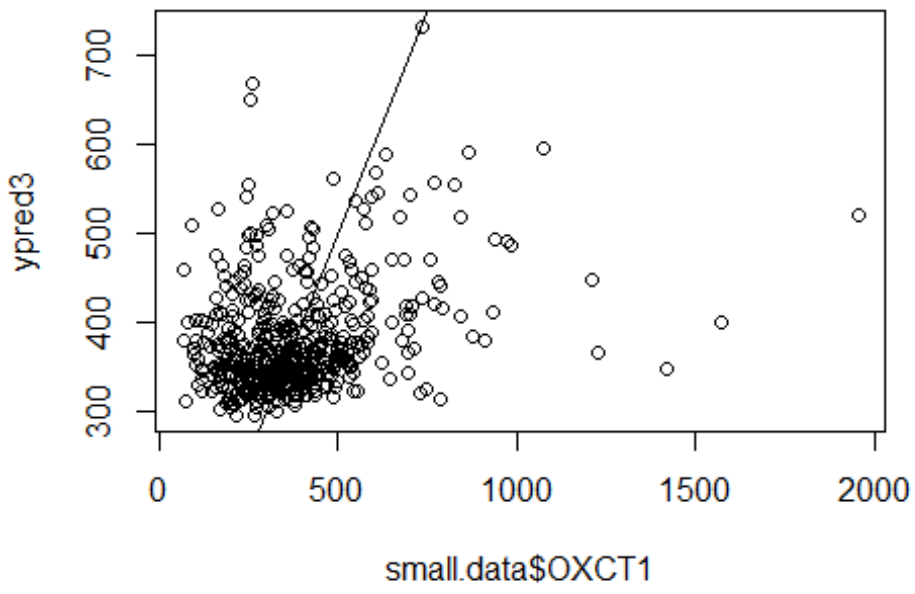


Model 3: OXCT1 as dependent variable, other genes as independent variable

```
model3 <- lm(OXCT1 ~ PCCB + EHHADH + MCCC1 + GCDH, data=small.data)
summary(model3)

##
## Call:
## lm(formula = OXCT1 ~ PCCB + EHHADH + MCCC1 + GCDH, data = small.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -421.64 -111.87  -13.82   75.31 1434.53
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 221.21796   28.06468   7.882 2.04e-14 ***
## PCCB          0.04909    0.04197   1.169   0.2428
## EHHADH        0.20685    0.17971   1.151   0.2503
## MCCC1         0.19189    0.03856   4.976 8.95e-07 ***
## GCDH          0.20990    0.11633   1.804   0.0718 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 193.1 on 498 degrees of freedom
## Multiple R-squared:  0.09262,    Adjusted R-squared:  0.08533
## F-statistic: 12.71 on 4 and 498 DF,  p-value: 7.435e-10
```

```
ypred3 <- predict(model3)
plot(small.data$OXCT1,ypred3)
abline(a=0,b=1)
```



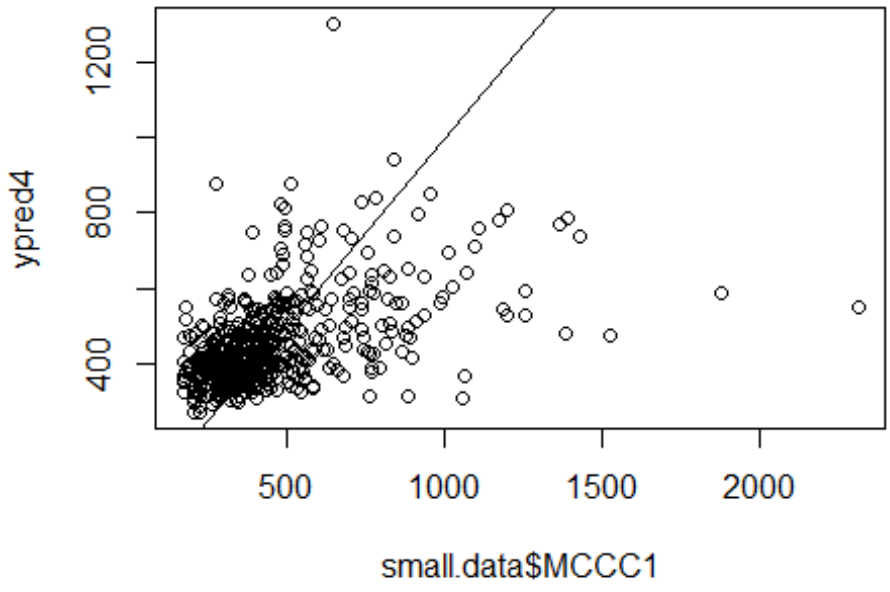
Model 4: MCCC1 as dependent variable, other genes as independent variable

```
model4 <- lm(MCCC1 ~ PCCB + EHHADH + OXCT1 + GCDH, data=small.data)
summary(model4)
```

```
##
## Call:
## lm(formula = MCCC1 ~ PCCB + EHHADH + OXCT1 + GCDH, data = small.data)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -658.07 -120.58  -46.87   57.76 1766.60
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 146.45863   33.11469   4.423 1.20e-05 ***
## PCCB          0.39706    0.04423   8.978  < 2e-16 ***
## EHHADH        0.54034    0.20266   2.666  0.00792 **
## OXCT1         0.24686    0.04961   4.976 8.95e-07 ***
## GCDH          0.15064    0.13220   1.140  0.25504
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 219 on 498 degrees of freedom
## Multiple R-squared:  0.2361, Adjusted R-squared:   0.23
## F-statistic: 38.49 on 4 and 498 DF,  p-value: < 2.2e-16
```

```
ypred4 <- predict(model4)
plot(small.data$MCCC1,ypred4)
abline(a=0,b=1)
```

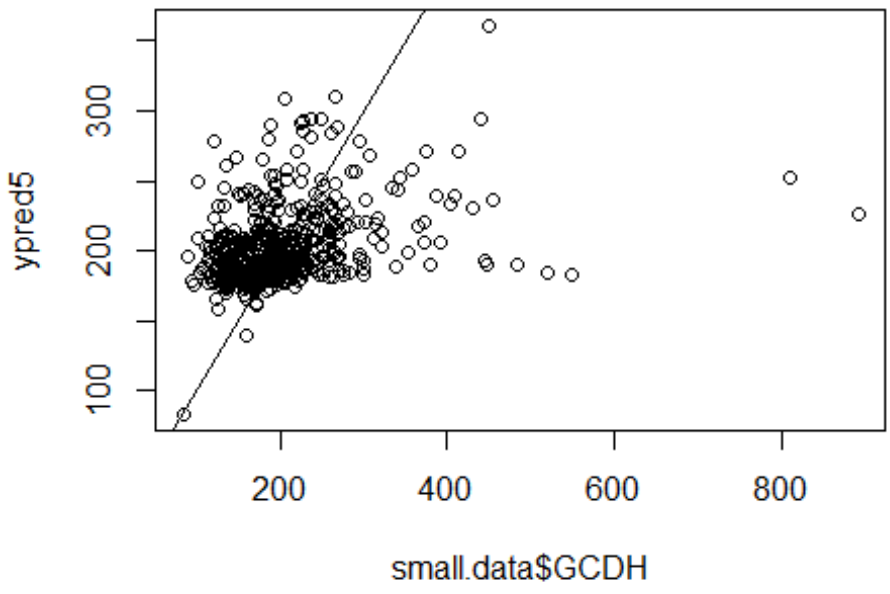


Model 5: GCDH as dependent variable, other genes as independent variable

```
model5 <- lm(GCDH ~ PCCB + EHHADH + OXCT1 + MCCC1, data=small.data)
summary(model5)
```

```
##
## Call:
## lm(formula = GCDH ~ PCCB + EHHADH + OXCT1 + MCCC1, data = small.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -155.95  -43.13   -9.85   24.60  666.00
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 149.96552    9.24340  16.224  < 2e-16 ***
## PCCB          0.09664    0.01555   6.217 1.08e-09 ***
## EHHADH       -0.10254    0.06894  -1.487   0.1376
## OXCT1         0.03094    0.01715   1.804   0.0718 .
## MCCC1         0.01726    0.01515   1.140   0.2550
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 74.15 on 498 degrees of freedom
## Multiple R-squared:  0.1222, Adjusted R-squared:  0.1151
## F-statistic: 17.33 on 4 and 498 DF,  p-value: 2.549e-13
```

```
ypred5 <- predict(model5)
plot(small.data$GCDH,ypred5)
abline(a=0,b=1)
```

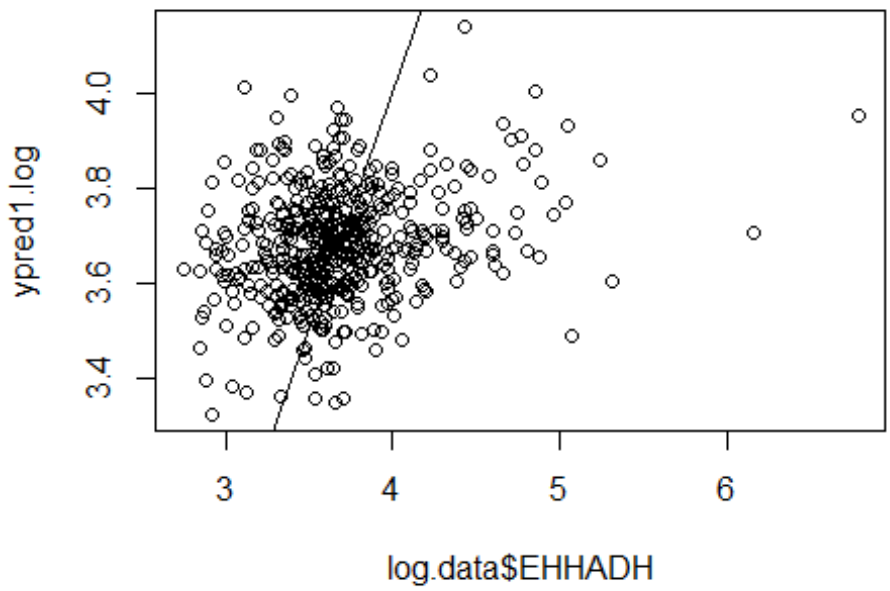


The same multiple linear regression models were created using the log-transformed data. For these models a scatterplot is plotted too, the x and y axis contain the same variables as in the plots shown above. **Model 1 log:** EHHADH as dependent variable, other genes as independent variable

```
model1.log <- lm(EHHADH ~ PCCB + OXCT1 + MCCC1 + GCDH, data=log.data)
summary(model1.log)
```

```
##
## Call:
## lm(formula = EHHADH ~ PCCB + OXCT1 + MCCC1 + GCDH, data = log.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.89620 -0.23564 -0.05122  0.16398  2.83661
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.10264    0.40955   7.576 1.75e-13 ***
## PCCB        -0.06205    0.04141  -1.498 0.134650
## OXCT1        0.12632    0.04137   3.054 0.002381 **
## MCCC1        0.18694    0.04970   3.762 0.000189 ***
## GCDH        -0.17477    0.06692  -2.611 0.009290 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4378 on 498 degrees of freedom
## Multiple R-squared:  0.06659,    Adjusted R-squared:  0.0591
## F-statistic: 8.882 on 4 and 498 DF,  p-value: 6.203e-07
```

```
ypred1.log <- predict(model1.log)
plot(log.data$EHHADH,ypred1.log)
abline(a=0,b=1)
```

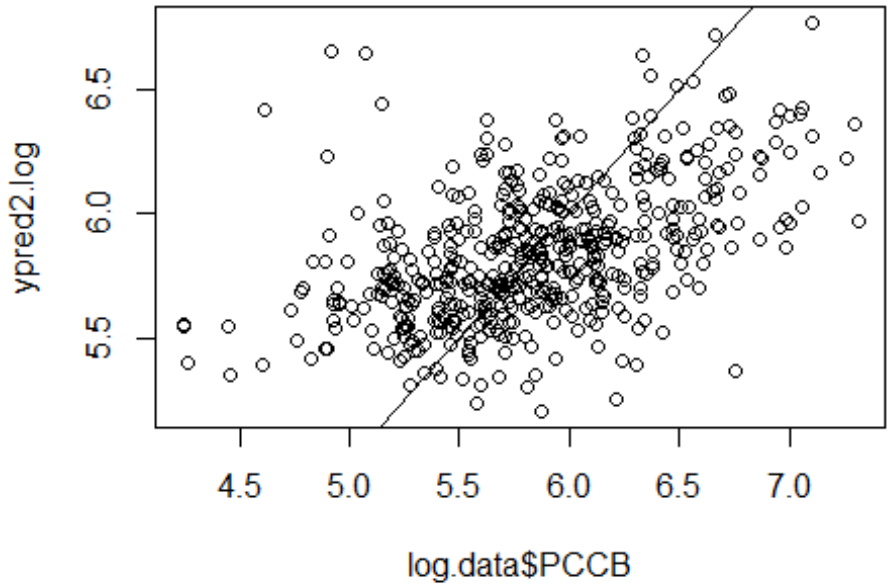


Model 2 log: PCCB as dependent variable, other genes as independent variable

```
model2.log <- lm(PCCB ~ EHHADH + OXCT1 + MCCC1 + GCDH, data=log.data)
summary(model2.log)
```

```
##
## Call:
## lm(formula = PCCB ~ EHHADH + OXCT1 + MCCC1 + GCDH, data = log.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.80948 -0.29931 -0.00212  0.29740  1.38850
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.24460    0.46362   2.685   0.0075 **
## EHHADH      -0.07233    0.04827  -1.498   0.1347
## OXCT1       -0.05259    0.04502  -1.168   0.2433
## MCCC1        0.43690    0.05077   8.605  < 2e-16 ***
## GCDH         0.48300    0.06946   6.954 1.12e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4726 on 498 degrees of freedom
## Multiple R-squared:  0.2498, Adjusted R-squared:  0.2437
## F-statistic: 41.45 on 4 and 498 DF,  p-value: < 2.2e-16
```

```
ypred2.log <- predict(model2.log)
plot(log.data$PCCB,ypred2.log)
abline(a=0,b=1)
```



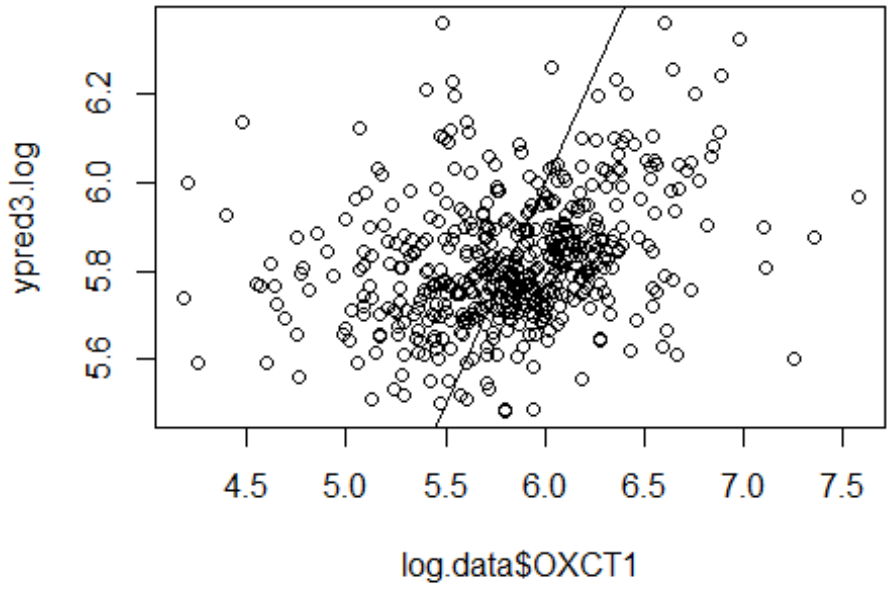
Model 3 log:

OXCT1 as dependent variable, other genes as independent variable

```
model3.log <- lm(OXCT1 ~ PCCB + EHHADH + MCCC1 + GCDH, data=log.data)
summary(model3.log)
```

```
##
## Call:
## lm(formula = OXCT1 ~ PCCB + EHHADH + MCCC1 + GCDH, data = log.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.79642 -0.25027  0.05026  0.28261  1.65574
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.35018    0.43925   7.627 1.23e-13 ***
## PCCB        -0.05197    0.04448  -1.168  0.24326
## EHHADH       0.14551    0.04765   3.054  0.00238 **
## MCCC1        0.26752    0.05275   5.072 5.58e-07 ***
## GCDH         0.12021    0.07212   1.667  0.09616 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4698 on 498 degrees of freedom
```

```
## Multiple R-squared:  0.09047,    Adjusted R-squared:  0.08316
## F-statistic: 12.38 on 4 and 498 DF,  p-value: 1.31e-09

ypred3.log <- predict(model3.log)
plot(log.data$OXCT1,ypred3.log)
abline(a=0,b=1)
```

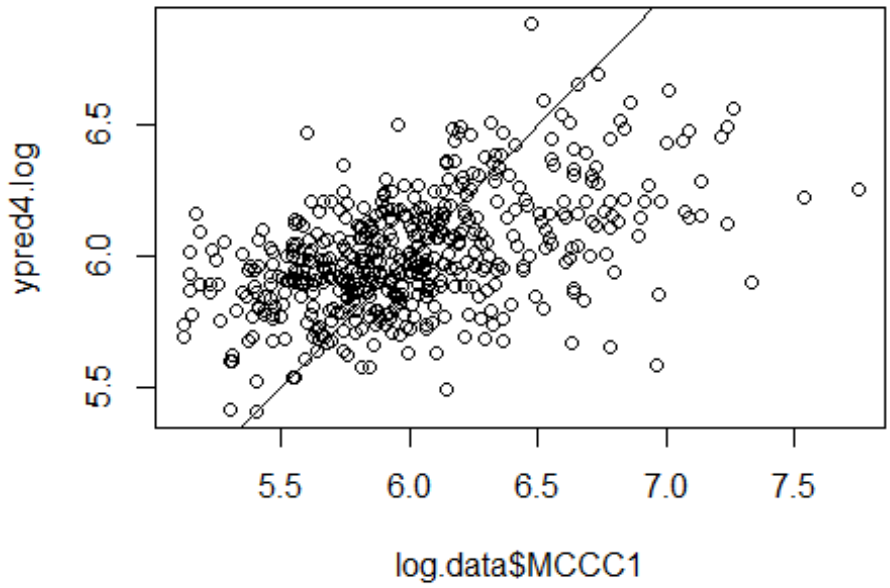


Model 4 log:

MCCC1 as dependent variable, other genes as independent variable

```
model4.log <- lm(MCCC1 ~ PCCB + EHHADH + OXCT1 + GCDH, data=log.data)
summary(model4.log)

##
## Call:
## lm(formula = MCCC1 ~ PCCB + EHHADH + OXCT1 + GCDH, data = log.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.99083 -0.26967 -0.03869  0.20706  1.49293
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.88288    0.37517   5.019 7.25e-07 ***
## PCCB         0.29629    0.03443   8.605  < 2e-16 ***
## EHHADH       0.14778    0.03929   3.762 0.000189 ***
## OXCT1        0.18358    0.03620   5.072 5.58e-07 ***
## GCDH         0.14877    0.05954   2.499 0.012783 *
## ---
```

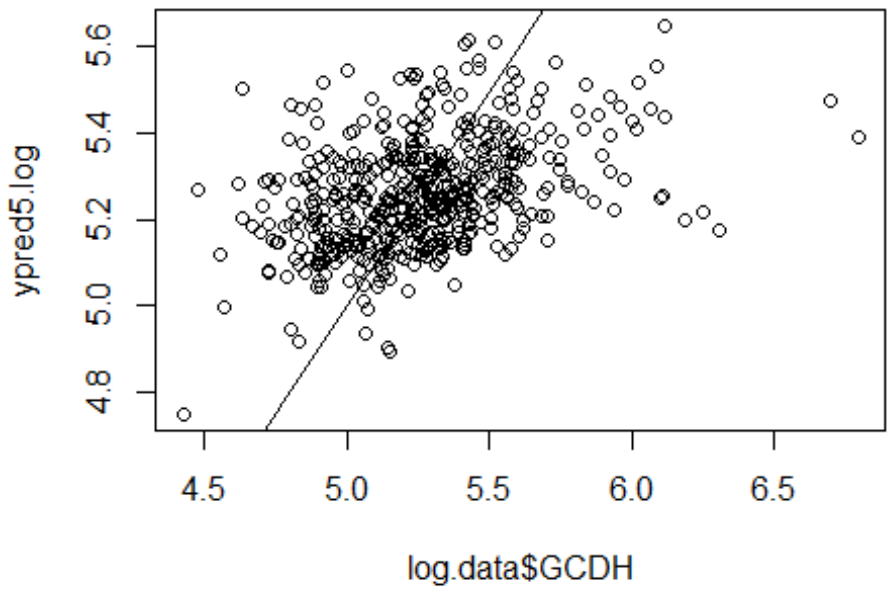
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3892 on 498 degrees of freedom
## Multiple R-squared:  0.249,  Adjusted R-squared:  0.2429
## F-statistic: 41.27 on 4 and 498 DF,  p-value: < 2.2e-16

ypred4.log <- predict(model4.log)
plot(log.data$MCCC1,ypred4.log)
abline(a=0,b=1)
```



Model 5 log: GCDH as dependent variable, other genes as independent variable

```
model5.log <- lm(GCDH ~ PCCB + EHHADH + MCCC1 + OXCT1, data=log.data)
summary(model5.log)

##
## Call:
## lm(formula = GCDH ~ PCCB + EHHADH + MCCC1 + OXCT1, data = log.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.87205 -0.18703 -0.00191  0.16392  1.40369
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.70595    0.23484  15.780  < 2e-16 ***
## PCCB         0.18325    0.02635   6.954 1.12e-11 ***
## EHHADH      -0.07729    0.02960  -2.611  0.00929 **
```

```
## MCCC1         0.08323    0.03331   2.499  0.01278 *
## OXCT1         0.04615    0.02769   1.667  0.09616 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2911 on 498 degrees of freedom
## Multiple R-squared:  0.1629, Adjusted R-squared:  0.1561
## F-statistic: 24.22 on 4 and 498 DF,  p-value: < 2.2e-16

ypred5.log <- predict(model5.log)
plot(log.data$GCDH,ypred5.log)
abline(a=0,b=1)
```



The following table shows the R^2 values for the models created on both the non transformed and log transformed data.

Predicted gene	R^2	Sig.var+	R^2 (log)	Sig.variables+(log)
EHHADH	0.025	MCCC1	0.067	OXCT1,MCCC1,GCDH
PCCB	0.251	MCCC1,GCDH	0.250	CCC1,GCDH
OXCT1	0.092	MCCC1	0.090	EHHADH,MCCC1
MCCC1	0.236	PCCB,EHHADH,OXCT1	0.249	PCCB,EHHADH,OXCT1,GCDH,OXCT1
GCDH	0.122	PCCB	0.162	PCCB,EHHADH,MCCC1

+ Significant variables with $p<0.05$

When looking at the summaries of the models it becomes clear that the coefficients are larger numbers for the non-transformed data but often these variables are not significant. The coefficients for the log-transformed data are closer to zero but often are far more often significant. For model 4 all 4 genes generate significant B-coefficients. We concluded that the predictive properties of these models is rather poor. This is shown by the low R^2 values and the poorly predicted scatterplots.

Task B

For this task, we used the *breast.data* data frame. First we inspect the data.

```
dim(breast.data)

## [1]  503 2562

colnames(breast.data)

tail(colnames(breast.data))

## [1] "SLC23A2" "SLC23A1" "SLC12A6" "KCNE2"   "NAT1"    "tissue"
```

Then we realised the last column of this data is tissue, which is not a gene expression value. Therefore, we created a new data frame without this column, and transform it into a matrix.

```
bdata <- as.matrix(breast.data[,1:2561])
```

Afterwards we loaded in the **glmnet** package, we used lasso regression to create models that predict each gene of the Krebs cycle using all the genes as possible independent variables. The lasso method would then calculate the optimal genes to use to create a model. We then tried to use leave-one-out cross validation however we realised it would take too much time to finish creating just one model. Therefore, we wanted to use the 10-fold cross validation. However, we have 503 samples in our data, and 503 is a prime number so this could not be divided by ten. In the end we had to delete the last 3 samples out of our dataset to be capable of using ten fold cross validation.

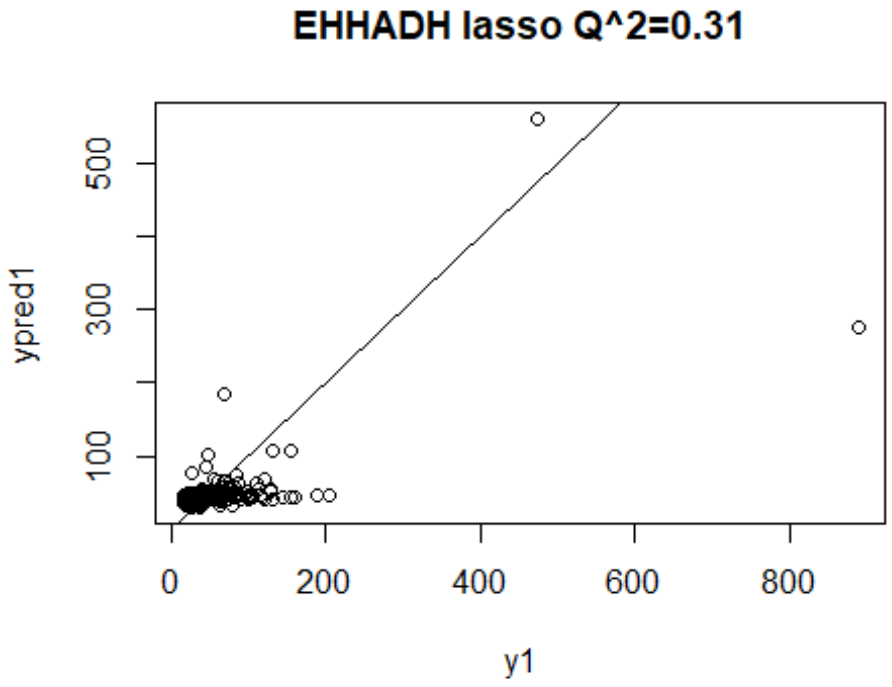
Inside the block of code for each model it is shown which genes the lasso model used to create the model. These genes are shown after the colnames(x.)[which(coef(mdl.) != 0)].

```
# create a new matrix with only 500 samples
bdata.less <- bdata[1:500,]
# create a 100-porin grid
grid <- 10^seq(10,-10,length=200)
# load the glmnet library
library(glmnet)

## Warning: package 'glmnet' was built under R version 3.4.2

## Loading required package: Matrix

## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 3.4.2

## Loaded glmnet 2.0-13
```

Lasso Model 1:

```
x1 <- bdata.less[,-which(colnames(bdata.less) =='EHHADH')]
y1 <- bdata.less[,which(colnames(bdata.less) =='EHHADH')]
# split data into 10-fold
fold1 <- split(sample(1:nrow(bdata.less)), 1:10)
ypred1 <- rep(NA, nrow(bdata.less))
for (i in fold1){
  mdl1 <- cv.glmnet(x=x1[-i,],y=y1[-i], alpha=1, lambda=grid)
  for (pos in i){
  ypred1[pos] <- predict(mdl1, newx=t(x1[pos,]))
  }
}
# calculate the Q-square
Q2.1 <- sum((ypred1-mean(y1))^2)/sum((y1-mean(y1))^2)
colnames(x1)[which(coef(mdl1) != 0)]

## [1] "NAALAD2" "GSTA3"   "PSAT1"   "ASS1"    "PPOX"    "SLC14A1" "PKD2L1"

# visulazition
plot(y1,ypred1,main=sprintf('%s lasso Q^2=%.2f','EHHADH',Q2.1))
abline(a=0,b=1)
```

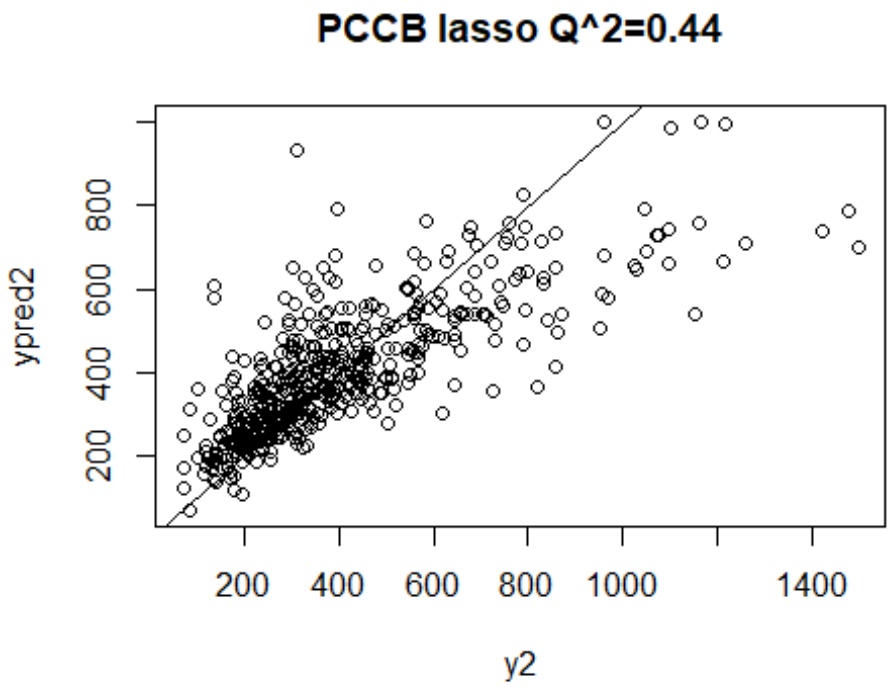


Lasso Model 2:

```r
x2 <- bdata.less[,-which(colnames(bdata.less) =='PCCB')]
y2 <- bdata.less[,which(colnames(bdata.less) =='PCCB')]
# split data into 10-fold
fold2 <- split(sample(1:nrow(bdata.less)), 1:10)
ypred2 <- rep(NA, nrow(bdata.less))
for (i in fold2){
  mdl2 <- cv.glmnet(x=x2[-i,],y=y2[-i], alpha=1, lambda=grid)
  for (pos in i){
  ypred2[pos] <- predict(mdl2, newx=t(x2[pos,]))
  }
}
# calculate the Q-square
Q2.2 <- sum((ypred2-mean(y2))^2)/sum((y2-mean(y2))^2)
colnames(x2)[which(coef(mdl2) != 0)]
```

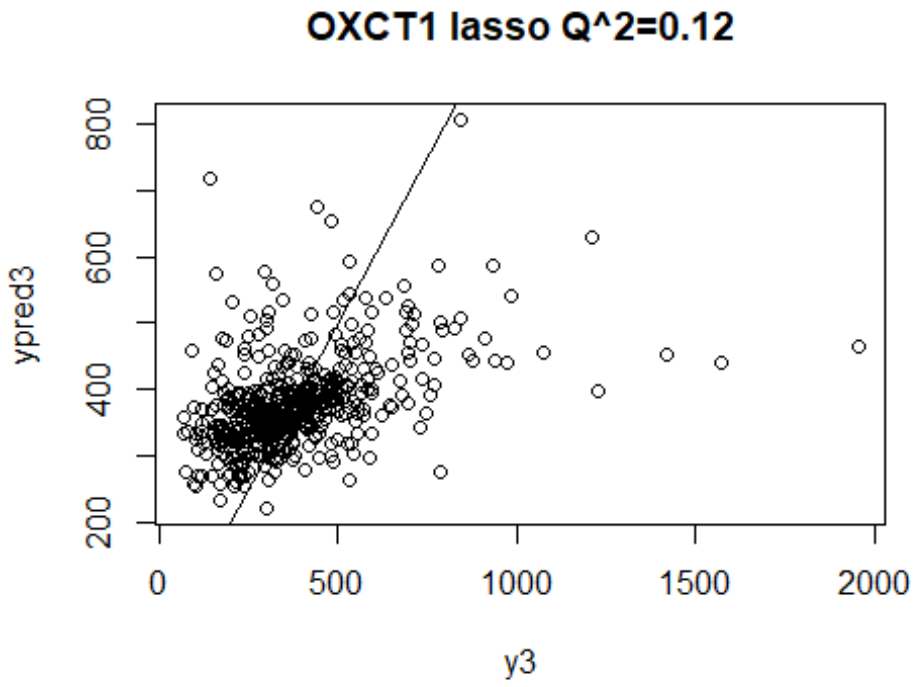
```
##  [1] "NAALAD2"   "STARD3"    "COX10"     "SARDH"     "TGDS"      "ACOT9"
##  [7] "PHGDH"     "B3GAT1"    "GPLD1"     "SLC9A9"    "GRIK1"     "GUCY1B2"
## [13] "DSE"       "SLC39A3"   "HK1"       "GSTK1"     "NDUFAB1"   "NDUFS2"
## [19] "AMDHD2"    "PLCE1"     "SLC25A38"  "AGPAT5"    "BCKDHB"    "RPE65"
## [25] "RRM1"      "GALNT11"   "SOD2"      "STAR"      "UPP1"      "UROD"
## [31] "ACAD11"    "LIPG"      "PTGES"     "ENTPD6"
```

```r
# visulazition
plot(y2,ypred2,main=sprintf('%s lasso Q^2=%.2f','PCCB',Q2.2))
abline(a=0,b=1)
```



Lasso Model 3:

```
x3 <- bdata.less[,-which(colnames(bdata.less) =='OXCT1')]
y3 <- bdata.less[,which(colnames(bdata.less) =='OXCT1')]
# split data into 10-fold
fold3 <- split(sample(1:nrow(bdata.less)), 1:10)
ypred3 <- rep(NA, nrow(bdata.less))
for (i in fold3){
  mdl3 <- cv.glmnet(x=x3[-i,],y=y3[-i], alpha=1, lambda=grid)
  for (pos in i){
  ypred3[pos] <- predict(mdl3, newx=t(x3[pos,]))
  }
}
# calculate the Q-square
Q2.3 <- sum((ypred3-mean(y3))^2)/sum((y3-mean(y3))^2)
colnames(x3)[which(coef(mdl3) != 0)]

##  [1] "NAALAD2" "ABCF2"   "KCNC1"   "LALBA"   "MDH2"    "NUDT1"   "NDUFAF1"
##  [8] "PGM3"    "CACNA1A" "NUDT22"  "CBS"     "SUCLA2"

# visulazition
plot(y3,ypred3,main=sprintf('%s lasso Q^2=%.2f','OXCT1',Q2.3))
abline(a=0,b=1)
```



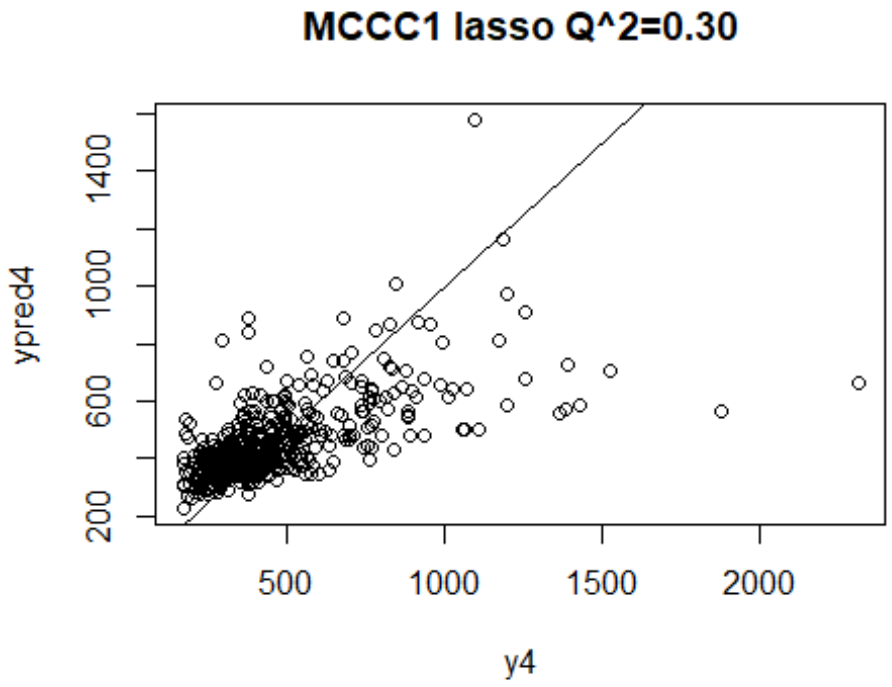
Lasso Model 4:

```
x4 <- bdata.less[,-which(colnames(bdata.less) =='MCCC1')]
y4 <- bdata.less[,which(colnames(bdata.less) =='MCCC1')]
# split data into 10-fold
```

```r
fold4 <- split(sample(1:nrow(bdata.less)), 1:10)
ypred4 <- rep(NA, nrow(bdata.less))
for (i in fold4){
  mdl4 <- cv.glmnet(x=x4[-i,],y=y4[-i], alpha=1, lambda=grid)
  for (pos in i){
  ypred4[pos] <- predict(mdl4, newx=t(x4[pos,]))
  }
}
# calculate the Q-square
Q2.4 <- sum((ypred4-mean(y4))^2)/sum((y4-mean(y4))^2)
colnames(x4)[which(coef(mdl4) != 0)]
```

```
##  [1] "NAALAD2"  "TREH"     "SLC2A13"  "PTGR1"    "P2RX2"    "ALDOB"
##  [7] "GANAB"    "GLS2"     "HYAL1"    "IDS"      "LCAT"     "LDHC"
## [13] "MDH2"     "NDUFB6"   "NDUFA13"  "ACSL5"    "GALNT7"   "THNSL2"
## [19] "AGPAT5"   "SVOP"     "RRM1"     "FN3K"     "SLC16A1"  "KCTD14"
## [25] "UGT2B4"   "SLC25A11" "NUDT22"   "PIGO"     "CDS2"
```

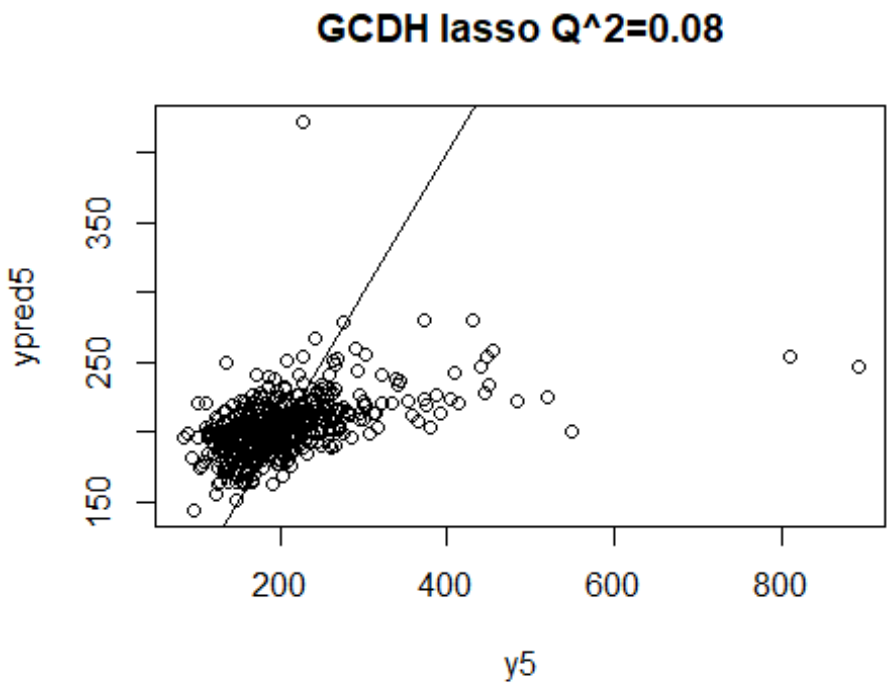
```r
# visulazition
plot(y4,ypred4,main=sprintf('%s lasso Q^2=%.2f','MCCC1',Q2.4))
abline(a=0,b=1)
```



Lasso Model 5:

```r
x5 <- bdata.less[,-which(colnames(bdata.less) =='GCDH')]
y5 <- bdata.less[,which(colnames(bdata.less) =='GCDH')]
# split data into 10-fold
```

```r
fold5 <- split(sample(1:nrow(bdata.less)), 1:10)
ypred5 <- rep(NA, nrow(bdata.less))
for (i in fold5){
  mdl5 <- cv.glmnet(x=x5[-i,],y=y5[-i], alpha=1, lambda=grid)
  for (pos in i){
  ypred5[pos] <- predict(mdl5, newx=t(x5[pos,]))
  }
}
# calculate the Q-square
Q2.5 <- sum((ypred5-mean(y5))^2)/sum((y5-mean(y5))^2)
colnames(x5)[which(coef(mdl5) != 0)]

## [1] "NAALAD2" "SMPDL3A" "DLAT"    "FOLH1"   "IDH3G"   "AQP4"    "NDUFA8"
## [8] "CRYL1"   "TF"

# visulazition
plot(y5,ypred5,main=sprintf('%s lasso Q^2=%.2f','GCDH',Q2.5))
abline(a=0,b=1)
```



Using ten fold cross validation a Q^2 value was calculated for all models. These Q^2 values are shown in the table below. The models for PCCB and MCCC1 are a lot better, the Q^2 is reasonably high and the plots show correlation between the line and the predicted values. Of these two the model for PCCB is clearly the best model.

Predicted gene	Q^2 value for model
EHHADH	0.06
PCCB	0.49
OXCT1	0.11
MCCC1	0.31
GCDH	0.09