M 372 Project 2 Solution

Cody Carroll

12/22/2023

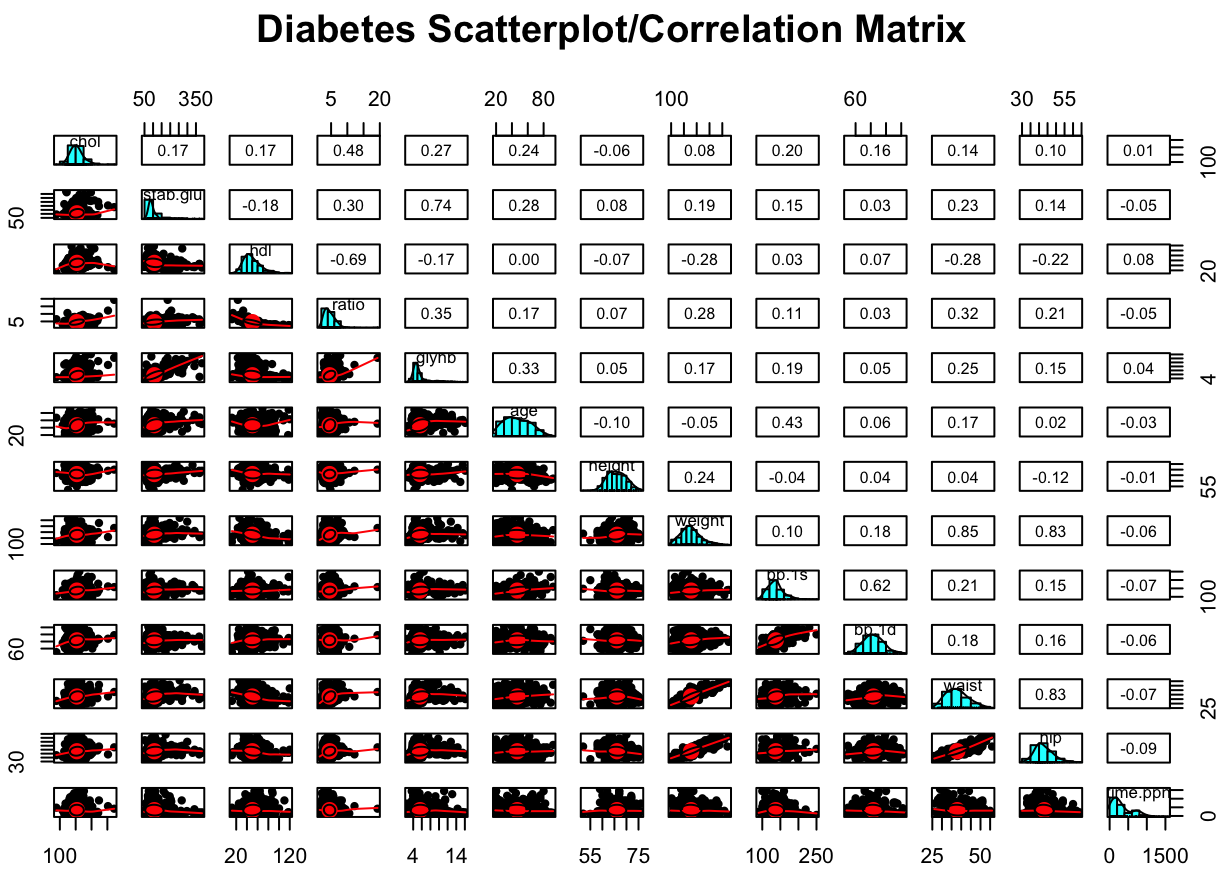
```
library(ggplot2)
library(ggpubr)
#setwd("~/Desktop/repos/math372/Projects")
setwd("~/Desktop/MATH 372 (Grading)/Project 2/")
d=read.table("diabetes.txt",header=T)
```

1.

The quantitative variables are `chol`, `stab.glu`, `hdl`, `ratio`, `glyhb`, `age`, `height`, `weight`, `bp.1d`, `waist`, `hip`, and `time.ppn`. The qualitative variables are `location`, `gender`, and `frame`.

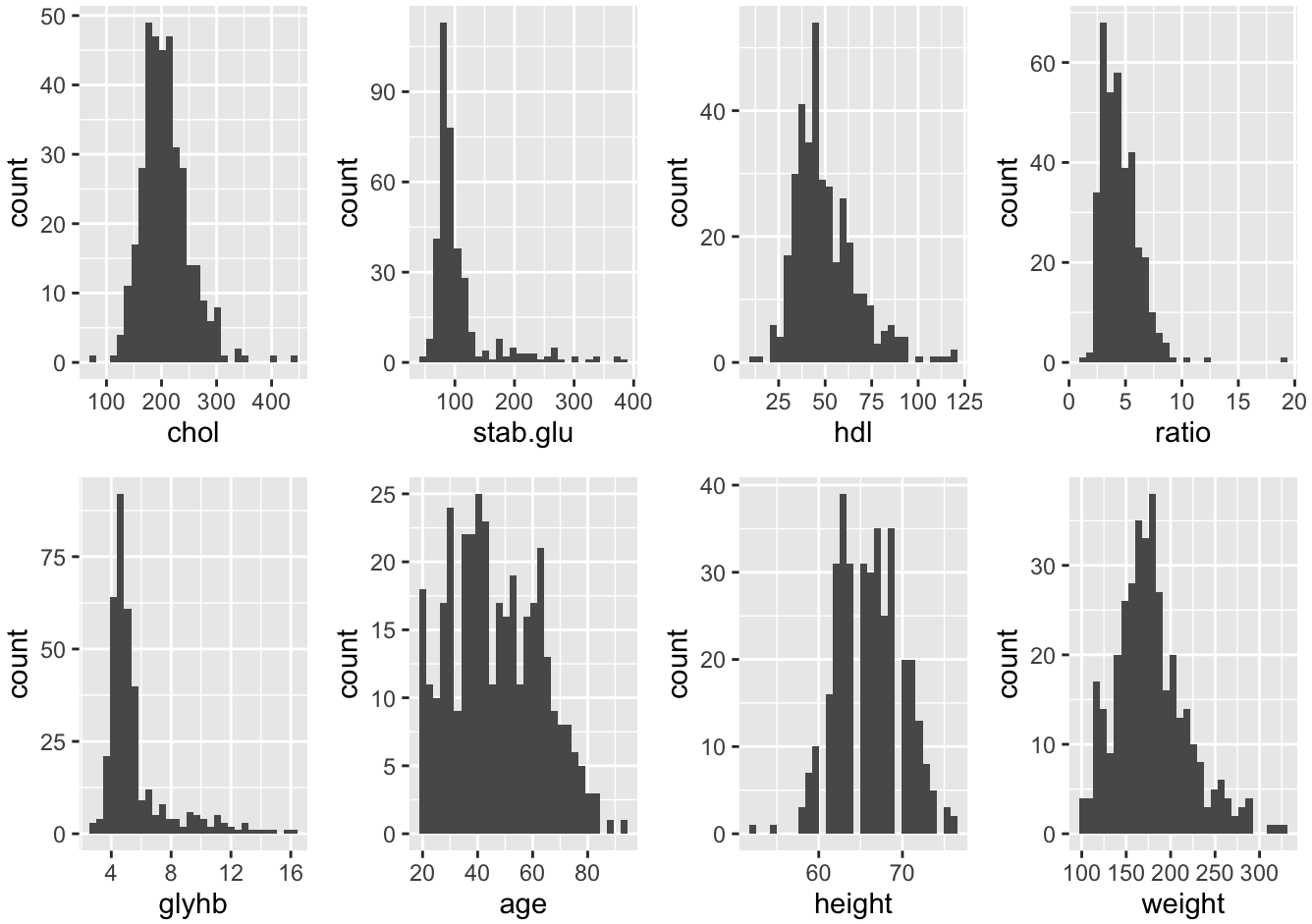
Histograms of the quantitative variables and the scatterplot matrix and correlation matrix are summarized below:

```
library(MASS)
library(psych)
pairs.panels(quant, main="Diabetes Scatterplot/Correlation Matrix", pch=20)
```

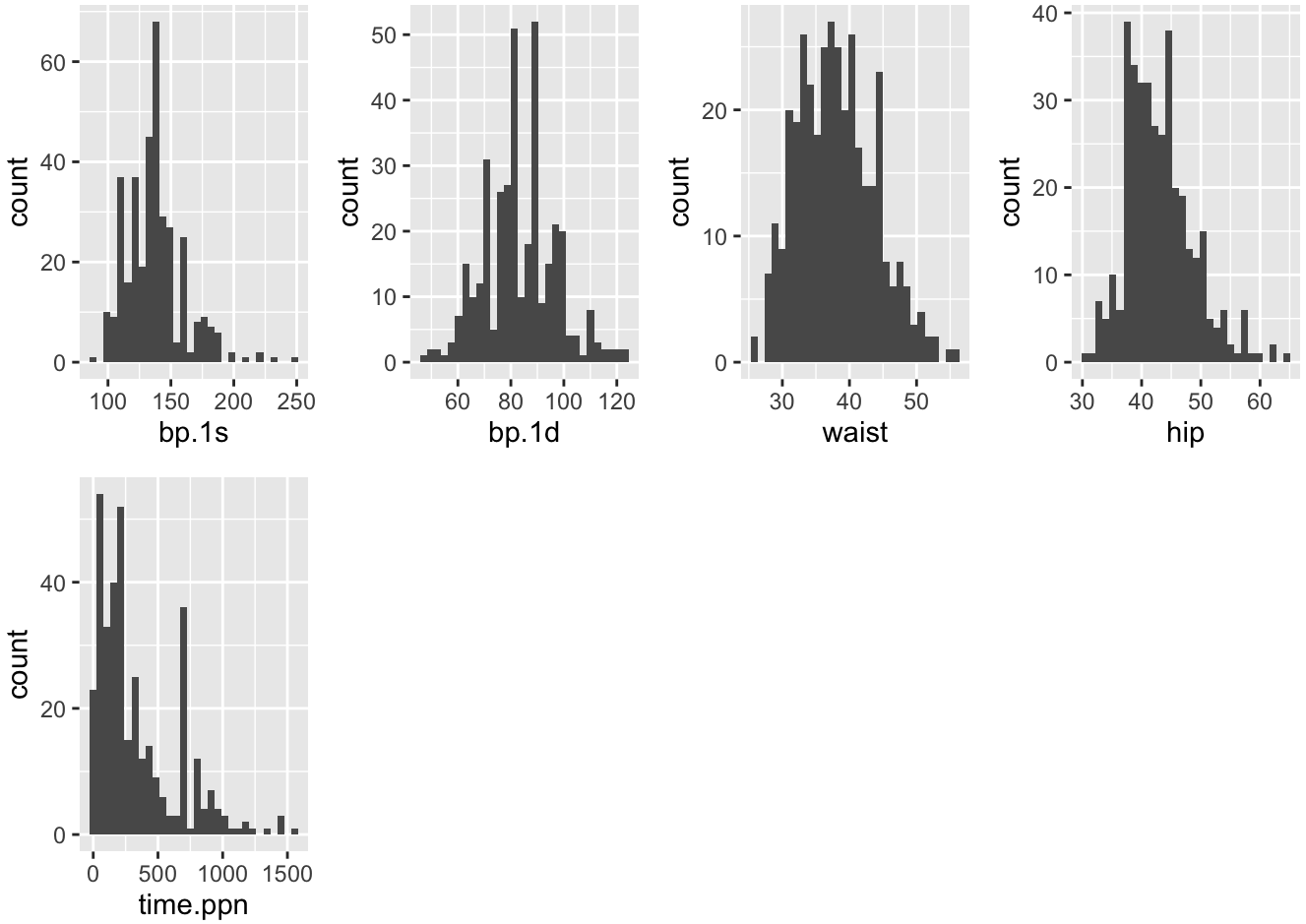


Taking a closer look at the histograms, we can examine their marginal distributions.

```
## $`1`
```

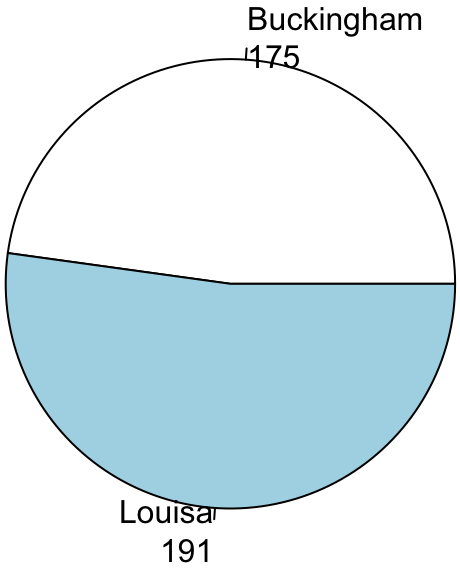


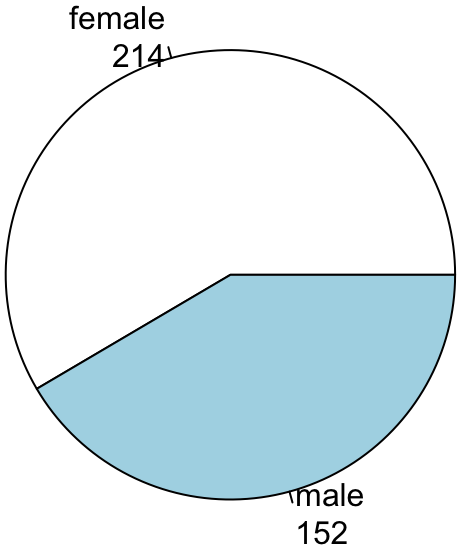

```
##
## $`2`
```

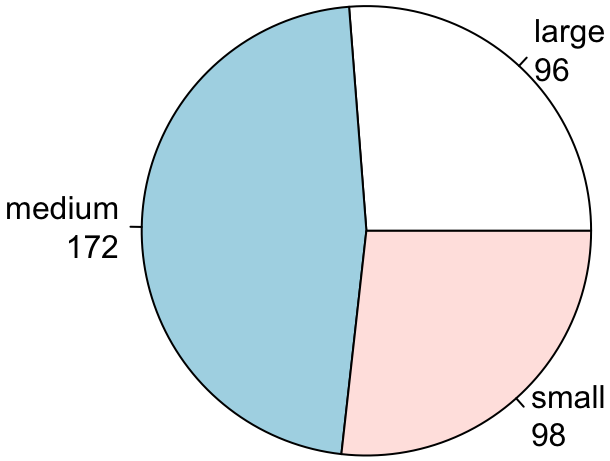



```
##
## attr(,"class")
## [1] "list"      "ggarrange"
```

The distributions of `cholesterol`, `age`, `height`, and `bp.1d` are all roughly symmetric. `stab.glu`, `hdl`, `ratio`, `glyhb`, and `weight` all have right skewed distributions. `bp.1s`, `waist`, and `hip` have slight right skewed distributions. The distribution of `time.ppn` is strongly right skewed.


Buckingham
175
Louisa
191

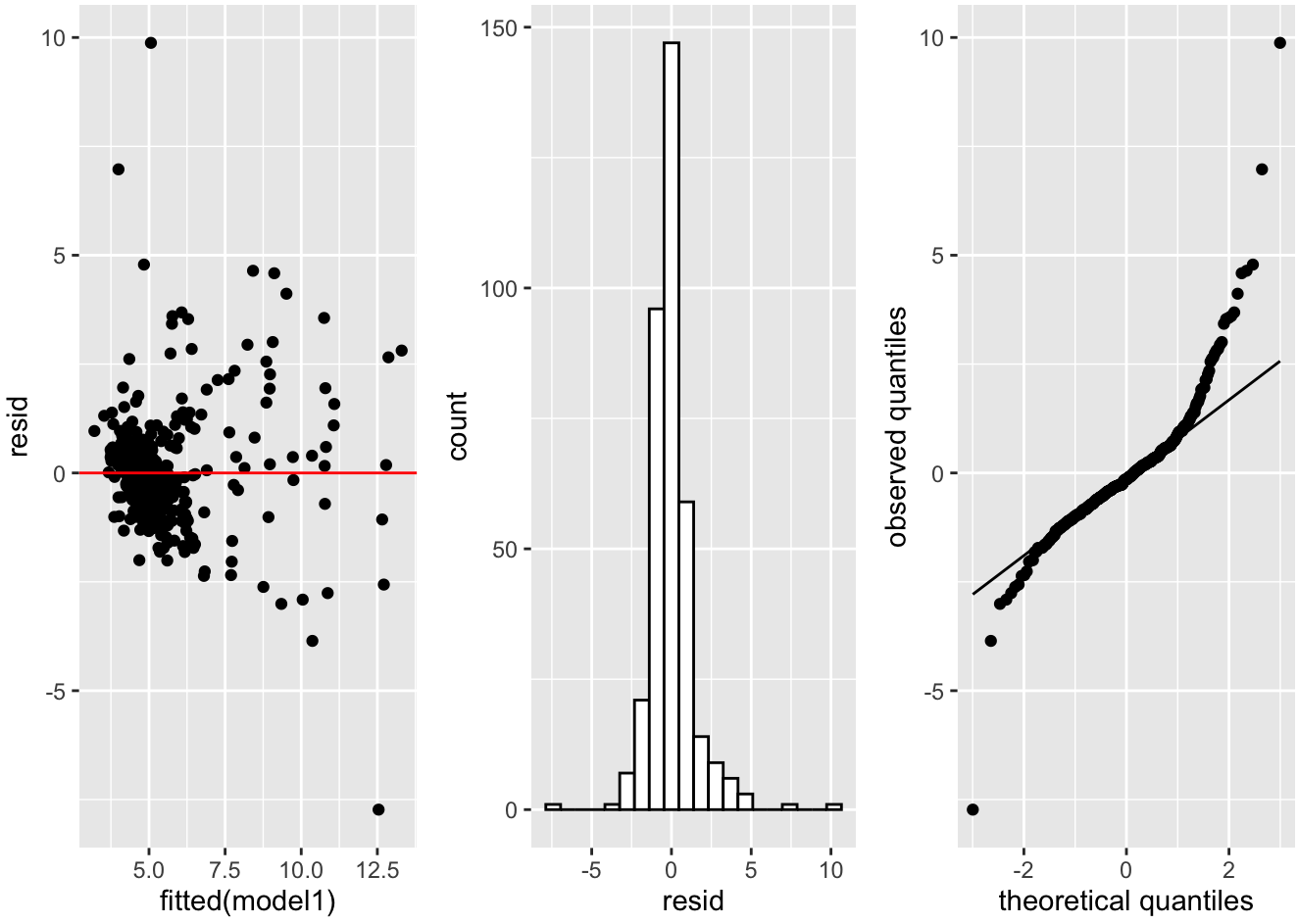

female
214
male
152



All three qualitative variables have an adequate proportion of all subgroups. We're not worried about severe imbalance.

2.

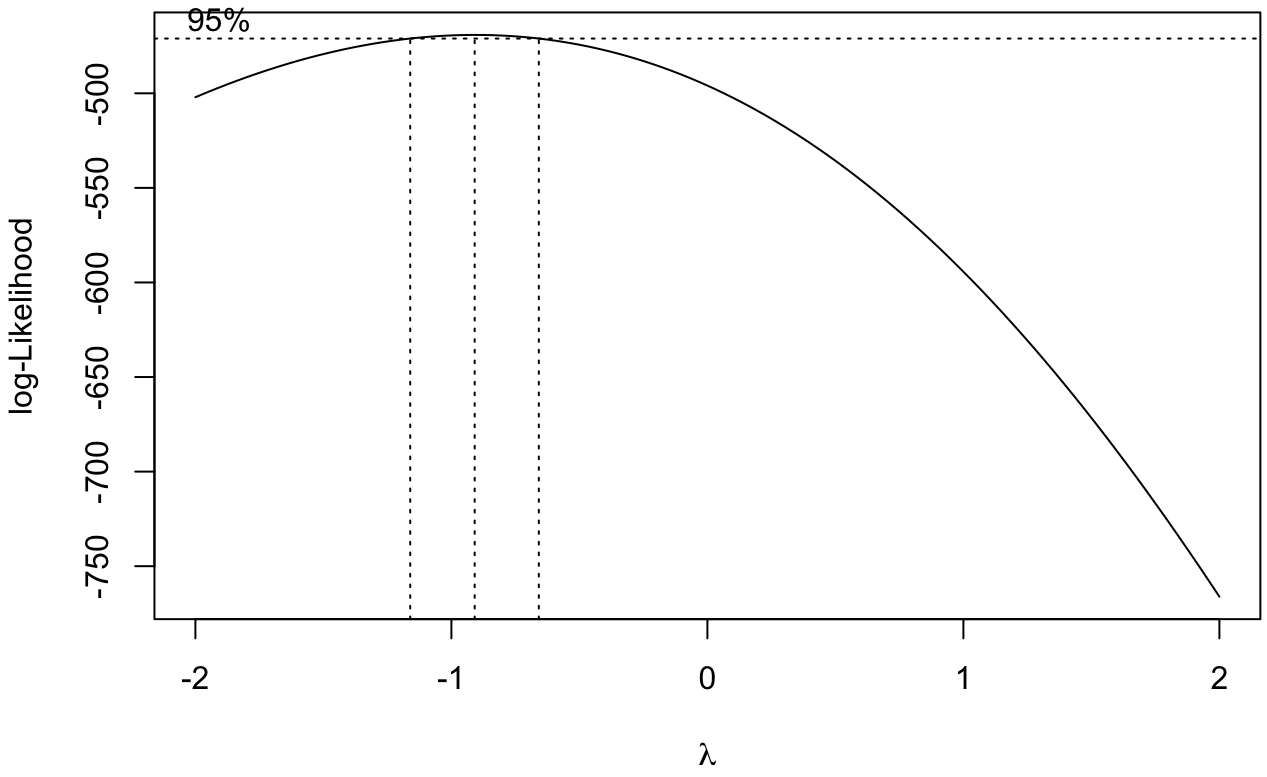
```
model1=lm(glyhb~.,data=d)

d_resid = d
d_resid$resid = model1$residuals

gg1 = ggplot(d_resid) +
  geom_point(aes(x = fitted(model1), y = resid)) +
  geom_hline(aes(yintercept = 0), color = "red")

gg2 = ggplot(d_resid, aes(x = resid)) +
  geom_histogram(bins = 20, color = "black", fill = "white")

gg3 = ggplot(d_resid, aes(sample = resid))
gg3 = gg3 +
  stat_qq() +
  stat_qq_line() +
  xlab("theoretical quantiles") +
  ylab("observed quantiles")


ggarrange(gg1, gg2, gg3, nrow = 1, ncol = 3)
```



The diagnostic plots show evidence of heteroskedasticity and a non-normal Q-Q plot (heavy tails). We should consider transforming the response, `glyhb`.

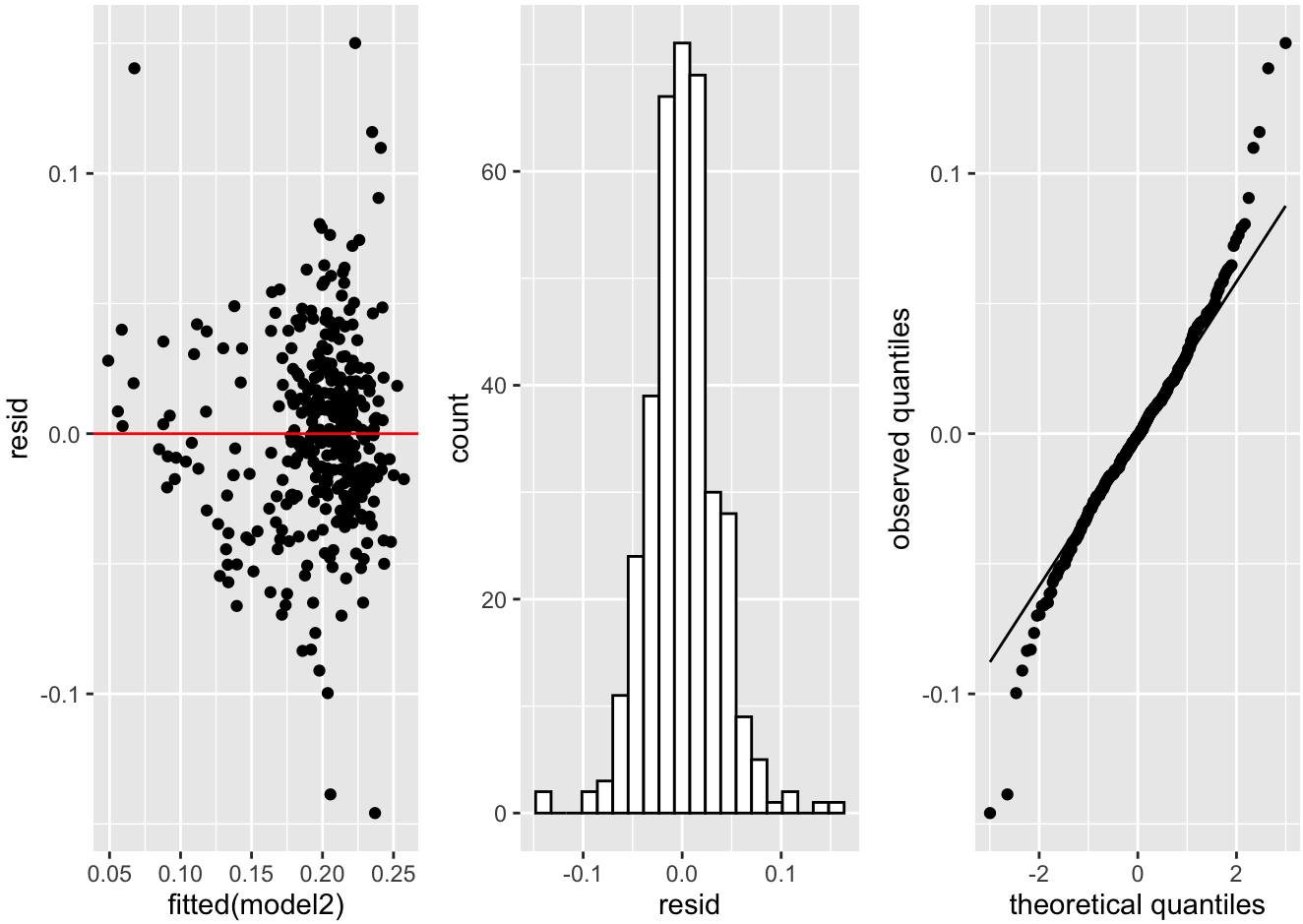
3.

```
boxcox(glyhb~., data=d)
```



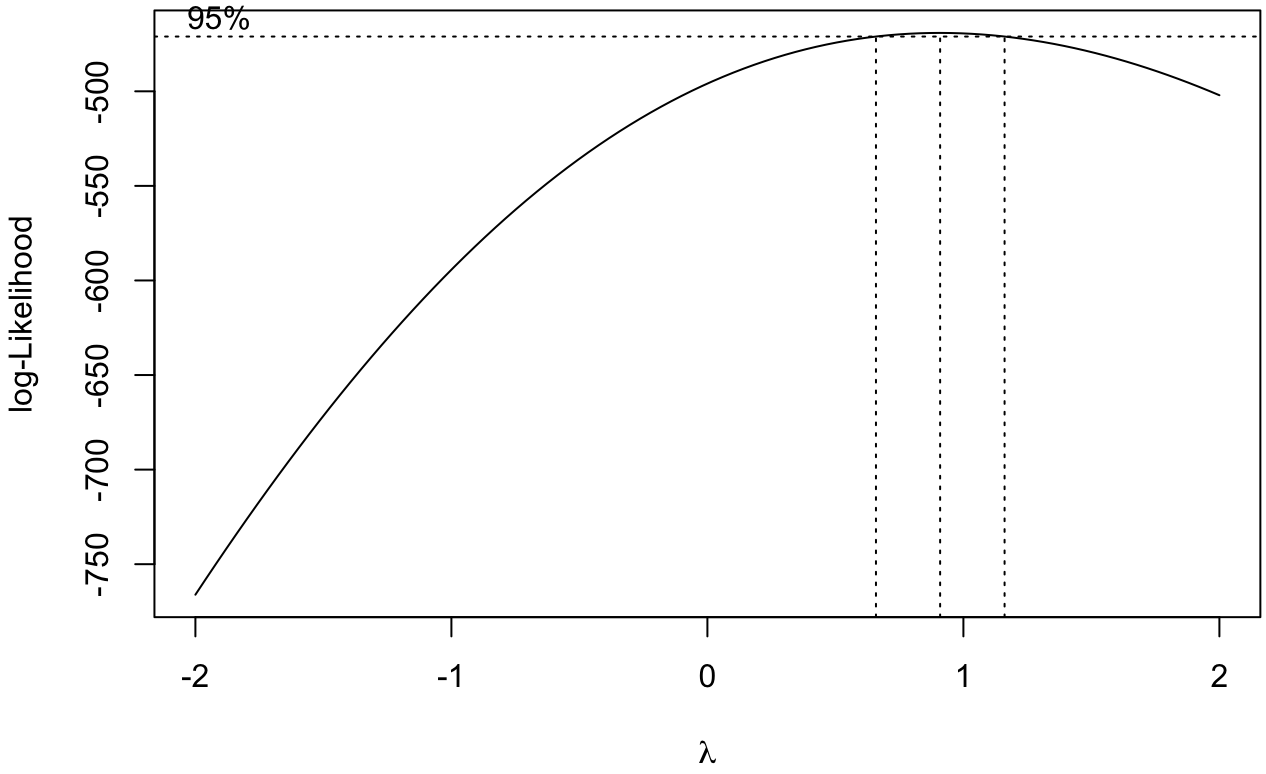
We use the Box-Cox method and see the log-likelihood is maximized around $\lambda = -1$. We take the inverse transformation and set `inv_glyhb` = 1/ `glyhb`.

```
d$inv_glyhb = 1/d$glyhb
d_trans=d[,!names(d)=="glyhb"] #lose the untransformed variable
model2=lm(inv_glyhb~., data=d_trans)
```



The new diagnostic plots show less heteroskedasticity but we are still concerned with the distribution of residuals having heavy tails.

```
boxcox(inv_glyhb~., data=d_trans)
```



The new log-Likelihood plot is maximized around $\lambda = 1$, i.e. no further transformations are necessary.

4.

We split the dataset into a training set and a validation set. We will build our model on the training set and check its validity with the validation set.

```
set.seed(372) ## set seed for random number generator
##so everyone gets the same split of the data.
N=nrow(d_trans) ## number of cases in d (366)
index=sample(1:N, size=round(.7*N, 0), replace=FALSE) ## randomly sample 256 cases to fo
rm the training data.
data.t=d_trans[index,] ## get the training data set.
data.test=d_trans[-index,] ## the remaining 183 cases form the validation set.
```

5.

We fit the model with all first-order effects.

```
model3=lm(inv_glyhb~.,data=data.t)
length(model3$coefficients)
```

```
## [1] 17
```

```
mse_full = summary(model3)$sigma^2
```

The model with all first-order effects has 17 predictors: one intercept, 14 for all the predictors except one qualitative variable frame, which has 3 levels and therefore and 2 dummy variables for frame. The MSE is 0.0015122.

6.

We consider best subsets selection.

```
library(leaps)
best=regsubsets(inv_glyhb~., data=data.t, nbest=1, nvmax=16)
n=nrow(data.t)
summary_stuff = summary(best)
names_of_data = c("Y",colnames(summary_stuff$which)[-1])
K = nrow(summary_stuff$which)
nicer = lapply(1:K,function(i){
  model = paste(names_of_data[summary_stuff$which[i,]],
                collapse = ",")
  p = sum(summary_stuff$which[i,])
  R2a = summary_stuff$adjr2[i]
  BIC = summary_stuff$bic[i]
  AIC = summary_stuff$bic[i] - (log(n)* p) + 2*p
  CP = summary_stuff$cp[i]
  SSE = summary_stuff$rss[i]
  Rsq = summary_stuff$rsq[i]
  results = data.frame(model,p,SSE, Rsq, R2a, CP,AIC, BIC)
  return(results)
})
nicer = Reduce(rbind,nicer)

#add null model
fit0=lm(inv_glyhb~1,data=data.t) # fit the model with only intercept
sse0=sum(fit0$residuals^2)
null_p=1
c1=sse0/mse_full-(n-2*null_p)
aic1=n*log(sse0)+2*null_p
bic1=n*log(sse0)+log(n)*null_p
null_df = data.frame(model = "Y~1", p = 1,SSE = sse0, Rsq = 0, R2a = 0, CP = c1, AIC = a
ic1, BIC =  bic1)
results = rbind(null_df, nicer) # combine the results with other models
```

The summary of the criteria and the best model of each size is below:

model <chr>
Y~1
Y,stab.glu
Y,stab.glu,age

model <chr>
Y,stab.glu,age,waist
Y,stab.glu,ratio,age,waist
Y,stab.glu,ratio,age,waist,time.ppn
Y,chol,stab.glu,hdl,age,waist,time.ppn
Y,chol,stab.glu,hdl,age,height,waist,time.ppn
Y,chol,stab.glu,hdl,age,gendermale,height,waist,time.ppn
Y,chol,stab.glu,hdl,age,gendermale,height,weight,waist,time.ppn

1-10 of 17 rows | 1-1 of 8 columns Previous **1** 2 Next

The best model, according to each criteria, is:
For the best model according to C_p, its C_p value is 3.43. If we see that $C_p << p$, it may be an indication that the MSE was overestimated by the full model. We suspect that the full model may be overspecified.

7.

We define models 3.1, 3.2, and 3.3 based on specific criteria:

```
#model 3.1, 3.2, 3.3
model3.1=lm(inv_glyhb~stab.glu+ ratio+ age+ waist+ time.ppn, data=data.t, x=T)#AIC selec
ted
model3.2=lm(inv_glyhb~stab.glu+age+waist, data=data.t, x=T)#BIC selected
model3.3=lm(inv_glyhb~chol+stab.glu+hdl+age+gender+height+waist+time.ppn, data=data.t, x
=T)#AdjR^2 selected
```

8.

We now fit the model with all 2-way interaction effects.

```
model4=lm(inv_glyhb~.^2, data=data.t, x=TRUE)
length(model4$coefficients)
```

```
## [1] 136
```

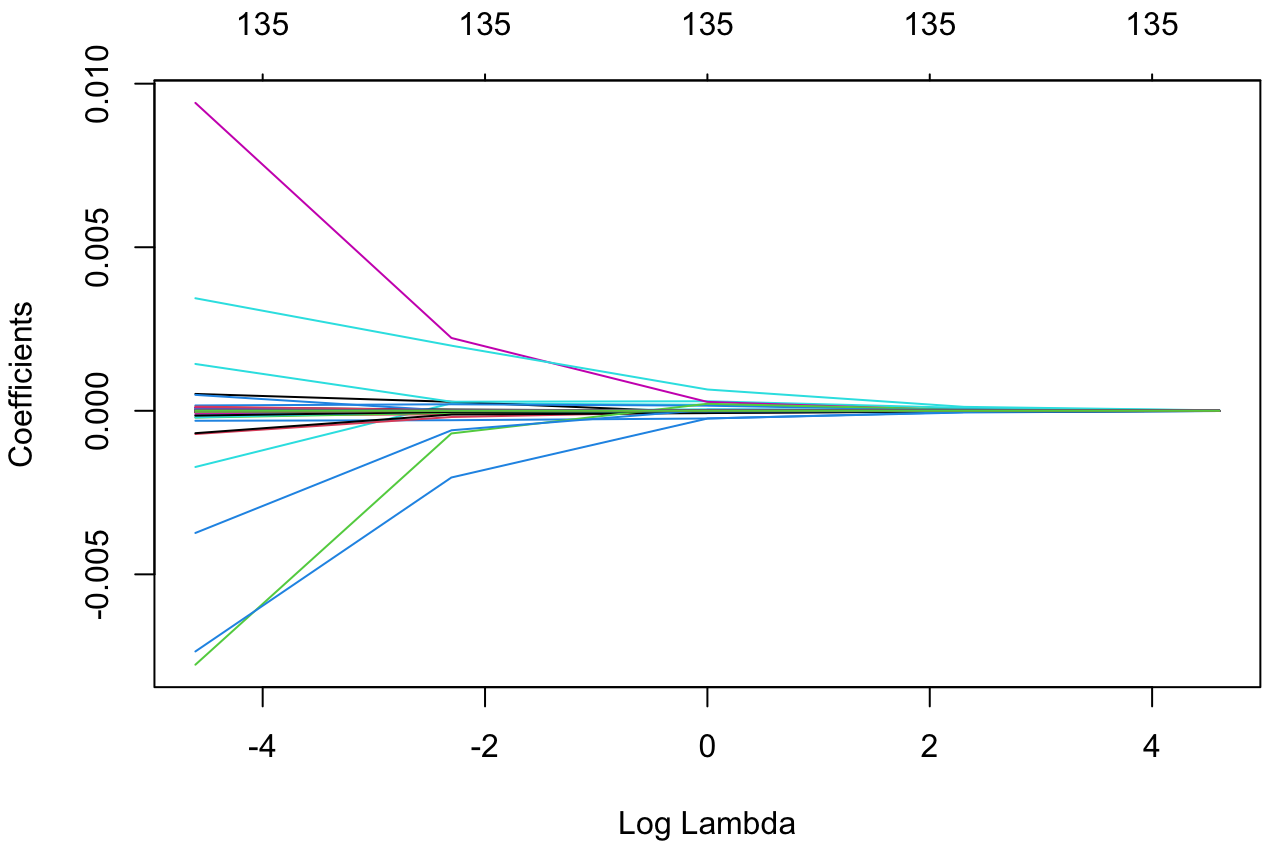
```
#nrow(data.t)
mse_full_int = summary(model4)$sigma^2
```

We have 136 predictors and only 183 data points. Overfitting and multicollinearity is a definite concern. The MSE is 0.0012813 which is even smaller than the MSE from the full first-order model, which we were already concerned is biased downward.

9.

Ridge Regression:

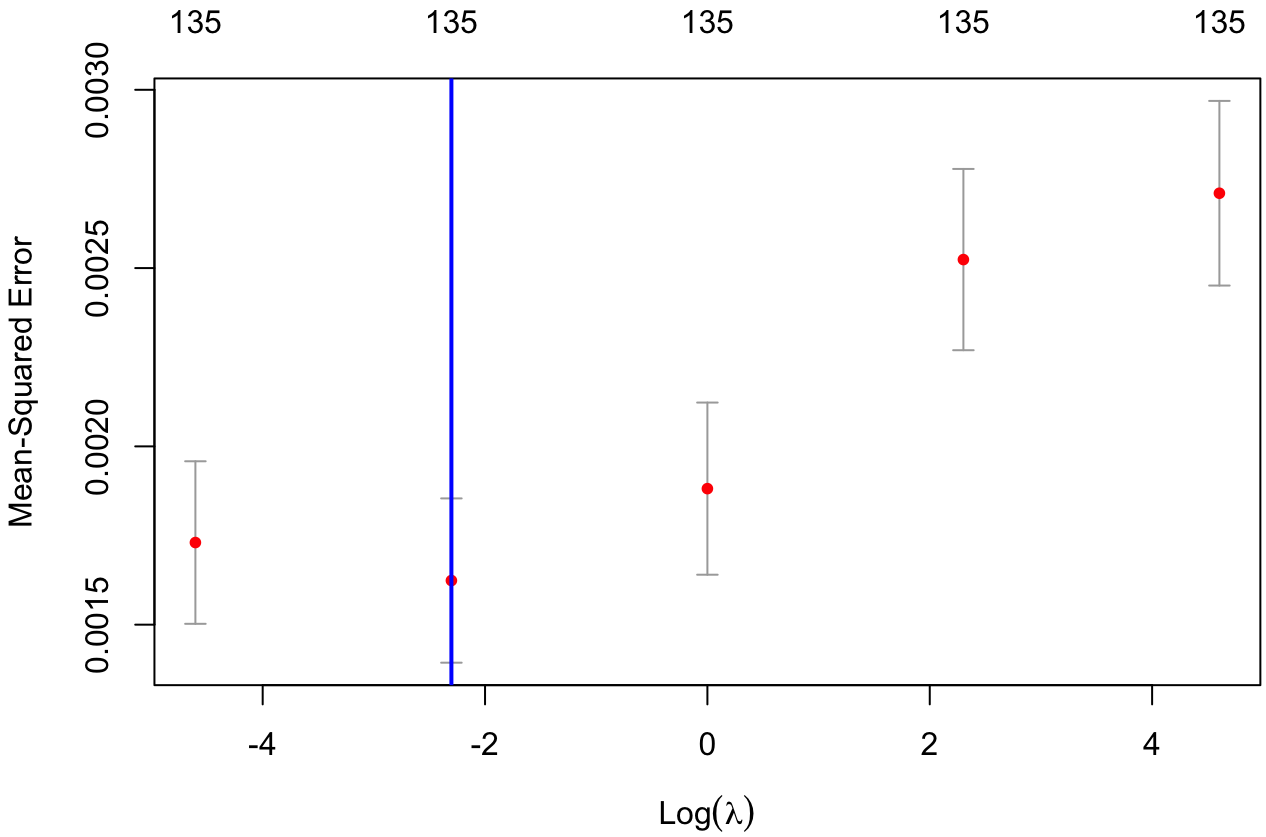
```
library(glmnet)
x.train = model.matrix(inv_glyhb ~.^2, data = data.t)[, -1] #drop intercept
y.train = data.t$inv_glyhb

x.test = model.matrix(inv_glyhb ~.^2, data = data.test)[, -1] #drop intercept
y.test = data.test$inv_glyhb


grid.lambda = 10^seq(-2, 2,  length = 5)
ridge.model = glmnet(x.train,
                     y.train,
                     alpha = 0,
                     lambda = grid.lambda)
plot(ridge.model, xvar = "lambda")
```



```
#cv for ridge
cv.ridge.out = cv.glmnet(x.train,
                         y.train,
                         alpha = 0,
                         lambda = grid.lambda)
plot(cv.ridge.out)


#Find the best lambda value
best.lambda.ridge = cv.ridge.out$lambda.min
best.lambda.ridge
```

```
## [1] 0.1
```

```
plot(cv.ridge.out)
abline(v = log(best.lambda.ridge), col = "blue", lwd = 2)
```



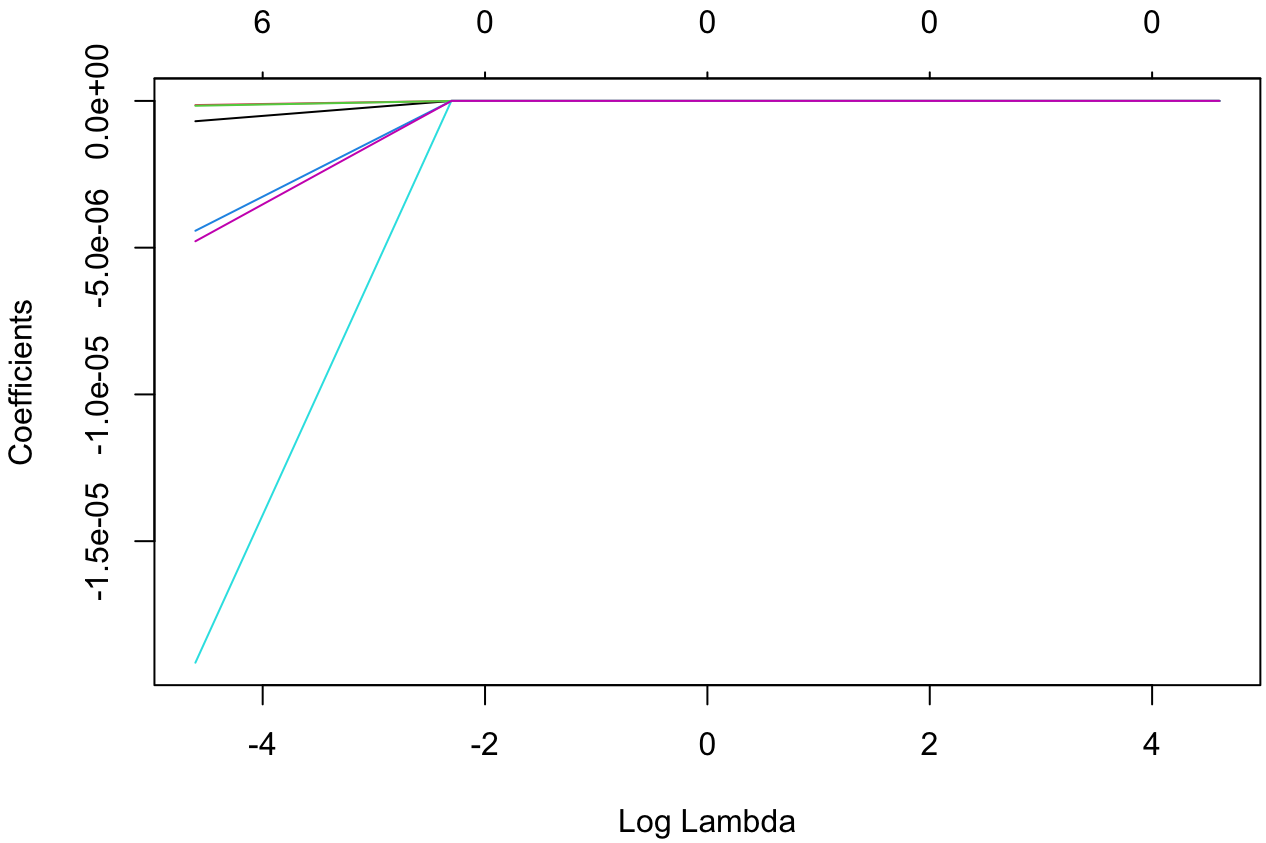
```
model4.1 = glmnet(x.train,
                  y.train,
                  alpha = 0,
                  lambda = best.lambda.ridge)

ridge.predict = predict(model4.1, s = best.lambda.ridge, newx = x.test)

mean((ridge.predict - y.test)^2)
```

```
## [1] 0.0009455666
```

10.

Lasso:

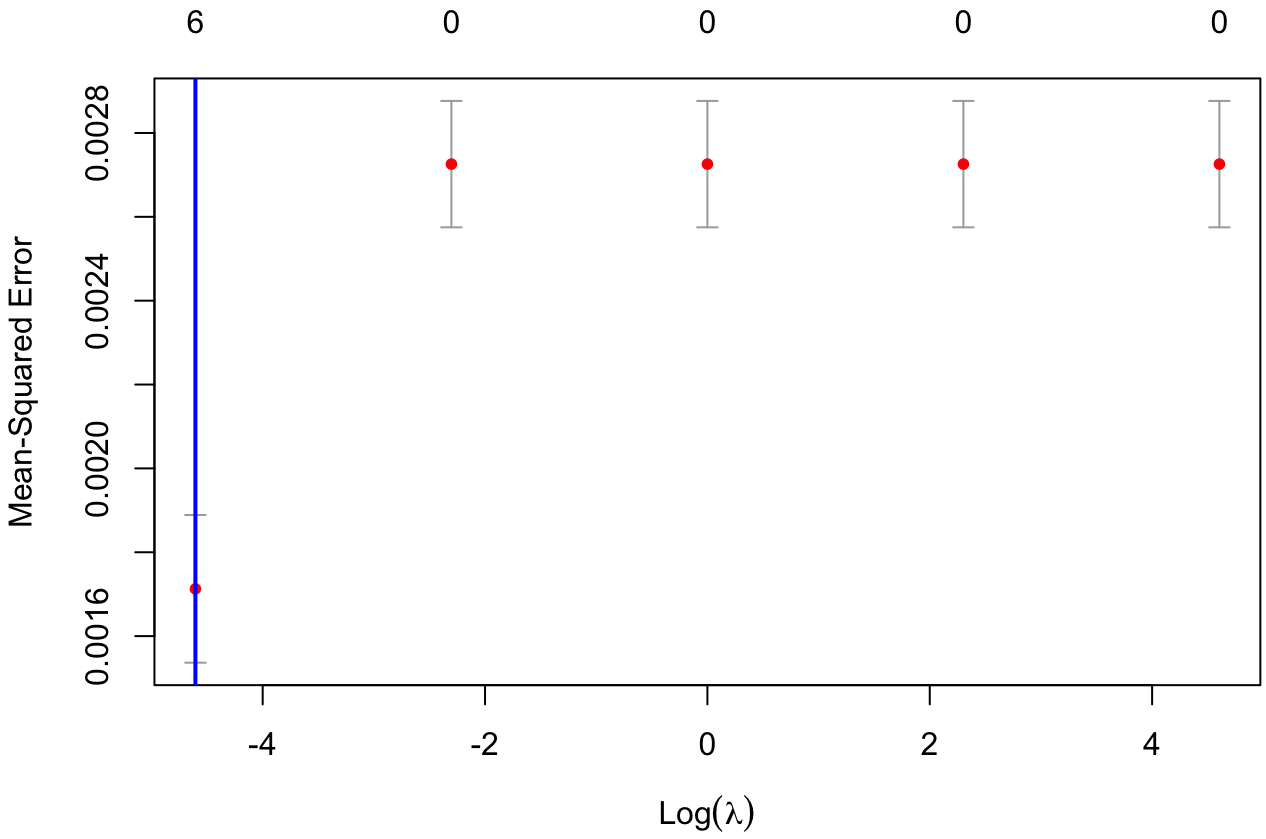
```
grid.lambda = 10^seq(2, -2,  length = 5)
lasso.model = glmnet(x.train,
                     y.train,
                     alpha = 1,
                     lambda = grid.lambda)
plot(lasso.model, xvar = "lambda")
```



```
#cv for lasso
cv.lasso.out = cv.glmnet(x.train, y.train, alpha = 1, lambda = grid.lambda)
plot(cv.lasso.out)

#Find the best lambda value
best.lambda.lasso = cv.lasso.out$lambda.min
best.lambda.lasso
```

```
## [1] 0.01
```

```
plot(cv.lasso.out)
abline(v = log(best.lambda.lasso), col = "blue", lwd = 2)
```



```
int.lasso.model = glmnet(x.train,
                         y.train,
                         alpha = 1,
                         lambda = best.lambda.lasso)

#need to add first order effects for any interaction terms!
model4.2 = lm(inv_glyhb ~ chol + stab.glu + age +
                        ratio + bp.1s + waist + chol:stab.glu +
                        stab.glu:age + stab.glu:bp.1s + stab.glu:waist +
                        ratio:age + age:waist,
                        data = data.t)
summary(model4.2)
```

```
##
## Call:
## lm(formula = inv_glyhb ~ chol + stab.glu + age + ratio + bp.1s +
##     waist + chol:stab.glu + stab.glu:age + stab.glu:bp.1s + stab.glu:waist +
##     ratio:age + age:waist, data = data.t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.15807 -0.02075  0.00111  0.02045  0.14614
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.460e-01  6.202e-02   5.579 6.43e-08 ***
## chol            1.399e-04  1.360e-04   1.028   0.3050
## stab.glu       -8.725e-04  5.015e-04  -1.740   0.0831 .
## age            -6.913e-04  1.134e-03  -0.609   0.5429
## ratio          -5.365e-04  5.685e-03  -0.094   0.9249
## bp.1s          -1.480e-04  3.698e-04  -0.400   0.6894
## waist          -6.779e-04  1.483e-03  -0.457   0.6480
## chol:stab.glu  -1.996e-06  9.916e-07  -2.013   0.0452 *
## stab.glu:age    8.569e-06  3.850e-06   2.226   0.0270 *
## stab.glu:bp.1s  1.352e-06  3.583e-06   0.377   0.7062
## stab.glu:waist  3.283e-06  9.721e-06   0.338   0.7359
## age:ratio      -3.103e-05  1.088e-04  -0.285   0.7758
## age:waist      -1.734e-05  2.744e-05  -0.632   0.5280
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03848 on 243 degrees of freedom
## Multiple R-squared:  0.4813, Adjusted R-squared:  0.4557
## F-statistic: 18.79 on 12 and 243 DF,  p-value: < 2.2e-16
```

```
final.lasso.pred = predict(model4.2,
                     newdata = as.data.frame(x.test))
mspe.lasso = mean((final.lasso.pred - y.test)^2)
mspe.lasso
```

```
## [1] 0.001020283
```

11.

Example differences:

- Shrinkage terms are different, which has effects on the coefficient values for each predictor
- Ridge always results in a full model
- Ridge penalty will never result in a coefficient of 0 for any predictor
- LASSO can be used for model selection (will tend to be smaller in size)
- LASSO penalty can result in coefficients of 0 –> means these predictors should be excluded from the model

12.

We move onto model validation now. For internal validation, we calculate PRESS_p for each model.

```
#PRESS
press.mod3.1=sum(model3.1$residuals^2/(1-influence(model3.1)$hat)^2)
press.mod3.1
```

```
## [1] 0.3944713
```

```
sse.mod3.1=sum(model3.1$residuals^2)
sse.mod3.1
```

```
## [1] 0.371942
```

```
press.mod3.2=sum(model3.2$residuals^2/(1-influence(model3.2)$hat)^2)
press.mod3.2
```

```
## [1] 0.4022217
```

```
sse.mod3.2=sum(model3.2$residuals^2)
sse.mod3.2
```

```
## [1] 0.3850162
```

```
press.mod3.3=sum(model3.3$residuals^2/(1-influence(model3.3)$hat)^2)
press.mod3.3
```

```
## [1] 0.395689
```

```
sse.mod3.3=sum(model3.3$residuals^2)
sse.mod3.3
```

```
## [1] 0.365279
```

```
#PRESS for ridge using a homemade LOOCV function:
loocv_sse = function(idxout){
  #hold one out
  x.train.loocv = x.train[-idxout,]
  y.train.loocv = y.train[-idxout]
  x.test.loocv = x.train[idxout,]
  y.test.loocv = y.train[idxout]

  #fit hold out model
  model_minus_idx = glmnet(x.train.loocv,
                           y.train.loocv,
                           alpha = 0,
                           lambda = best.lambda.ridge)

  #predict hold out fitted val
  fittedval.idx = predict(model_minus_idx,
                          lambda = best.lambda.ridge,
                          newx = x.test.loocv)

  #calculate squared error
  sse = (y.test.loocv - fittedval.idx)^2
  return(sse)
}

press.ridge = lapply(1:n, FUN = loocv_sse) %>% unlist %>% sum

#calculate SSE for ridge
ridge.fitted = predict(model4.1,
                       s = best.lambda.ridge,
                       newx = x.train)

ridge.residuals = y.train - ridge.fitted
sse.mod4.1 = sum(ridge.residuals^2)

#
#compare
press.ridge
```

```
## [1] 0.4092355
```

```
sse.mod4.1
```

```
## [1] 0.3627011
```

```
press.mod4.2=sum(model4.2$residuals^2/(1-influence(model4.2)$hat)^2)
press.mod4.2
```

```
## [1] 0.4049749
```

```
sse.mod4.2=sum(model4.2$residuals^2)
sse.mod4.2
```

```
## [1] 0.3598317
```

For most models, PRESS and SSE are somewhat close, though PRESS is always larger, as expected. Larger differences between PRESS and SSE make us concerned about overfitting, e.g. in the ridge regression case. Model 3.1 had the lowest PRESS, though not by much.

13.

We now try external validation on the validation set.

```
##mean squared prediction error
newdata=data.test[,-16]#exclude inv_glybh response

pred.mod3.1=predict.lm(model3.1, newdata)
mspe.mod3.1=mean((pred.mod3.1-data.test[,16])^2)
mspe.mod3.1
```

```
## [1] 0.0009214996
```

```
press.mod3.1/nrow(data.t)
```

```
## [1] 0.001540903
```

```
pred.mod3.2=predict.lm(model3.2, newdata)
mspe.mod3.2=mean((pred.mod3.2-data.test[,16])^2)
mspe.mod3.2
```

```
## [1] 0.0009775372
```

```
press.mod3.2/nrow(data.t)
```

```
## [1] 0.001571179
```

```
pred.mod3.3=predict.lm(model3.3, newdata)
mspe.mod3.3=mean((pred.mod3.3-data.test[,16])^2)
mspe.mod3.3
```

```
## [1] 0.0009874315
```

```
press.mod3.3/nrow(data.t)
```

```
## [1] 0.00154566
```

```
#ridge mspe
#Calculate the MSPE of the model on the test set
ridge.pred = predict(model4.1,
                     s = best.lambda.ridge,
                     newx = x.test)
mspe.ridge4.1 = mean((ridge.pred - y.test)^2)
mspe.ridge4.1
```

```
## [1] 0.0009455666
```

```
press.ridge/nrow(data.t)
```

```
## [1] 0.001598576
```

```
#lasso mspe
lasso.pred = predict(model4.2,
                     newdata = as.data.frame(x.test))
mspe.lasso4.2 = mean((lasso.pred - y.test)^2)
mspe.lasso4.2
```

```
## [1] 0.001020283
```

```
press.mod4.2/nrow(data.t)
```

```
## [1] 0.001581933
```

The MSPE's are close to the PRESS/n values. Interestingly, the external metric MSPE's are smaller than the internal PRESS/n numbers. This means that the models are generalizing to a test set better than the LOOCV/PRESS statistic would suggest.

Model 3.1 has the smallest MSPE. This model was also the winner in internal validation. It's a small enough model to be interpretable with only 5 predictors, and doesn't involve any interaction terms which can be tricky to explain.

14.

For these reasons we elect to use model 3.1 as our final model. We run it on the full dataset and display the summary and anova outputs:

```
model5=lm(inv_glyhb~stab.glu+ratio+age+waist+time.ppn, data=d_trans) #final model: the AIC-selected one
summary(model5)
```

```
##
## Call:
## lm(formula = inv_glyhb ~ stab.glu + ratio + age + waist + time.ppn,
##     data = d_trans)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.153480 -0.020857 -0.001696  0.020034  0.149250
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.446e-01  1.332e-02  25.862  < 2e-16 ***
## stab.glu    -4.943e-04  3.819e-05 -12.945  < 2e-16 ***
## ratio       -3.731e-03  1.175e-03  -3.176  0.00162 **
## age         -6.576e-04  1.223e-04  -5.379 1.35e-07 ***
## waist       -1.116e-03  3.500e-04  -3.187  0.00156 **
## time.ppn    -1.359e-05  6.125e-06  -2.218  0.02716 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03624 on 360 degrees of freedom
## Multiple R-squared:  0.5073, Adjusted R-squared:  0.5004
## F-statistic: 74.13 on 5 and 360 DF,  p-value: < 2.2e-16
```

```
anova(model5)
```

	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
stab.glu	1	0.397525176	0.397525176	302.756905	1.210279e-49
ratio	1	0.027939264	0.027939264	21.278665	5.528202e-06
age	1	0.042213987	0.042213987	32.150357	2.931364e-08
waist	1	0.012518398	0.012518398	9.534067	2.173440e-03
time.ppn	1	0.006460695	0.006460695	4.920493	2.716341e-02
Residuals	360	0.472686373	0.001313018	*NA*	*NA*

6 rows