Project 2 - MATH 372

Hadley Dixon

2023-11-15

Data Description

The data "diabetes.txt" contains 16 variables on 366 subjects who were interviewed in a study to understand the prevalence of obesity, diabetes, and other cardiovascular risk factors in central Virginia for African Americans. We will consider building regression models with glyhb as the response variable, as Glycosolated Hemoglobin levels greater than 70 is often taken as a positive diagnosis of diabetes. The goal is to find the "best" model for explaining the factors which are predictive of diabetes diagnosis.

Data Analysis

```
library(ggplot2, quietly = TRUE)
library(dplyr, quietly = TRUE)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse, quietly = TRUE)
```

```
## ── Attaching packages ─────────────────────────────────────
## tidyverse 1.3.2 ──
```

```
## ✔ tibble  3.2.1     ✔ purrr   1.0.2
## ✔ tidyr   1.3.0     ✔ stringr 1.5.1
## ✔ readr   2.1.2     ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(Hmisc, quietly = TRUE)
```

```
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
##
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(GGally, quietly = TRUE)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(MASS, quietly = TRUE)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(leaps, quietly = TRUE)
library(glmnet, quietly = TRUE)
```

```
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-8
```

```
library(MPV, quietly = TRUE)
```

```
## KernSmooth 2.23 loaded
## Copyright M. P. Wand 1997-2009
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
##
##
## Attaching package: 'MPV'
##
## The following object is masked from 'package:MASS':
##
##     cement
```

```
library(olsrr, quietly = TRUE)
```

```
##
## Attaching package: 'olsrr'
##
## The following object is masked from 'package:MPV':
##
##     cement
##
## The following object is masked from 'package:MASS':
##
##     cement
##
## The following object is masked from 'package:datasets':
##
##     rivers
```

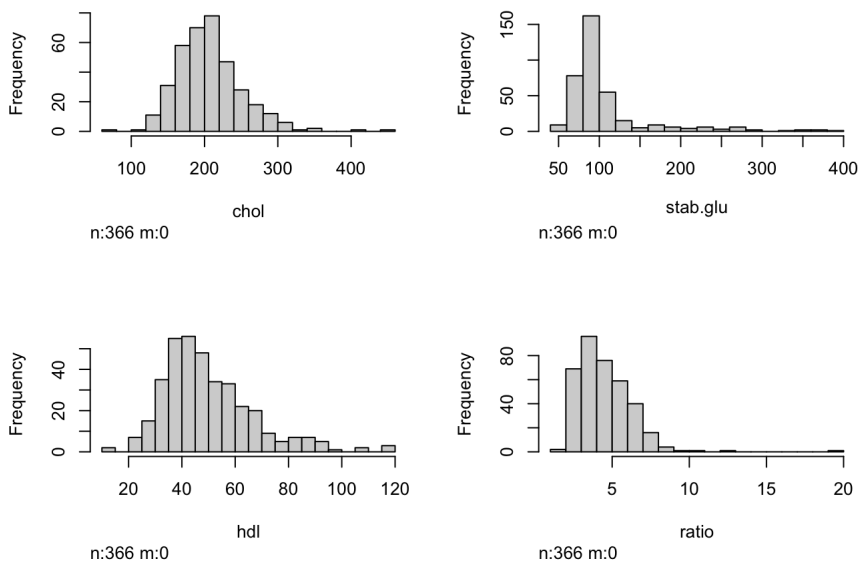
```
diabetes <- read.table(file = "diabetes.txt", header = TRUE)
```

Data exploration and data splitting

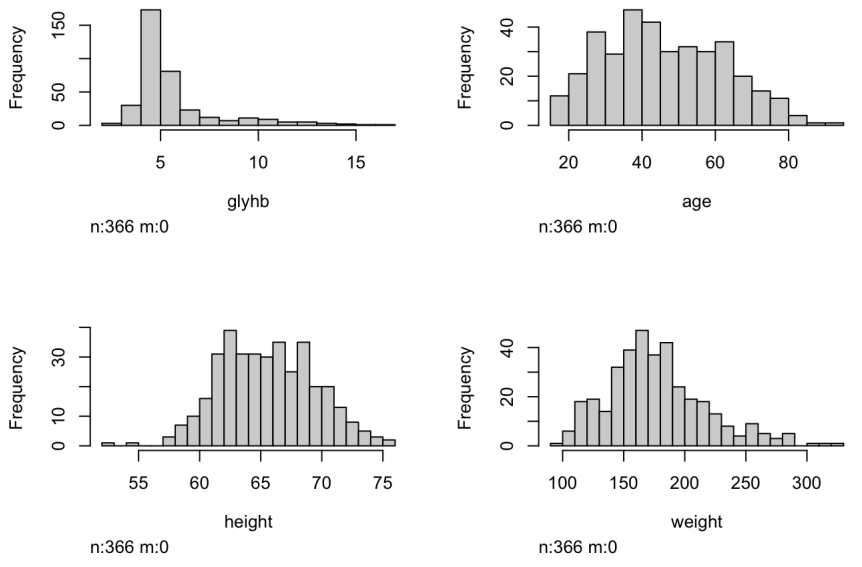
1. Among all the variables, which are quantitative variables? Which are qualitative variables? Draw histogram for each quantitative variable and comment on its distribution. Draw pie chart for each qualitative variable and comment on how its classes are distributed. Draw scatter plot matrix and obtain the pairwise correlation matrix for all quantitative variables in the data. Comment on their relationships.

The quantitative variables are as follows: chol, stab.glu, hdl, ratio, glyhb (response), age, height, weight, bp.1s, bp.1d, waist, hip, and time.ppn. The qualitative variables are as follows: location, gender, and frame.

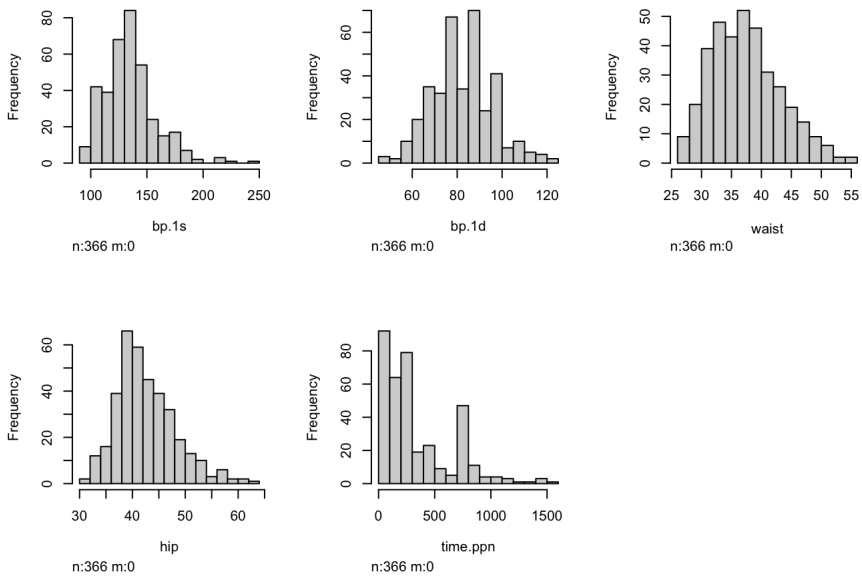
```
# Numeric columns excluding response variable
quant <-diabetes[, sapply(diabetes, is.numeric)]
hist(quant[0:4])
```




```
hist(quant[5:8])
```



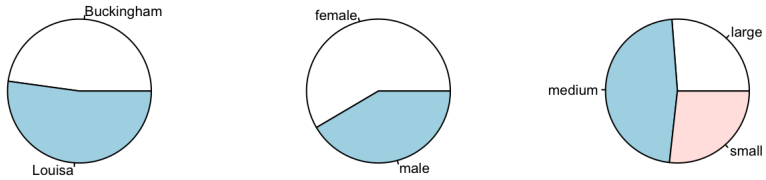

```
hist(quant[9:13])
```



Histogram of chol: Follows a Normal distribution with a slight right skew.
Histogram of stab.glu: Follows a Normal distribution with under dispersion, meaning that the data is heavily concentrated at 0 and has light tails. Also, the data is has a strong right skew.
Histogram of hdl: Follows a Normal distribution with a moderate right skew.
Histogram of ratio: Follows a Normal distribution with a string right skew.
Histogram of glyhb: Follows a Normal distribution with under dispersion, meaning that the data is heavily concentrated at 0 and has light tails. Also, the data is has a strong right skew.
Histogram of age: Follows a Uniform distribution.
Histogram of height: Follows a Normal distribution with over dispersion and heavy tails. Also the data has a slight left skew.
Histogram of weight: Follows a Normal distribution with over dispersion and heavy tails. Also the data has a slight right skew.
Histogram of bp.1s: Follows a Normal distribution with a moderate right skew.
Histogram of bp.1d: Follows a Normal distribution with slight over dispersion.
Histogram of waist: Follows a Normal distribution with over dispersion and heavy tails. Also the data has a slight right skew.
Histogram of hip: Follows a Normal distribution with a slight right skew.
Histogram of time.ppn: Follows a Chi-squared distribution.

```
# Pie chart for each qualitative variable
qual <-diabetes[, sapply(diabetes, negate(is.numeric))]

par(mfrow = c(1, ncol(qual)))

for (col in names(qual)) {
  temp <- data.frame(table(qual[[col]]))
  names(temp) <- c("label", "value")
  pie(temp$value, labels = temp$label)
  title(paste0("Pie Chart for ", col))
}
```

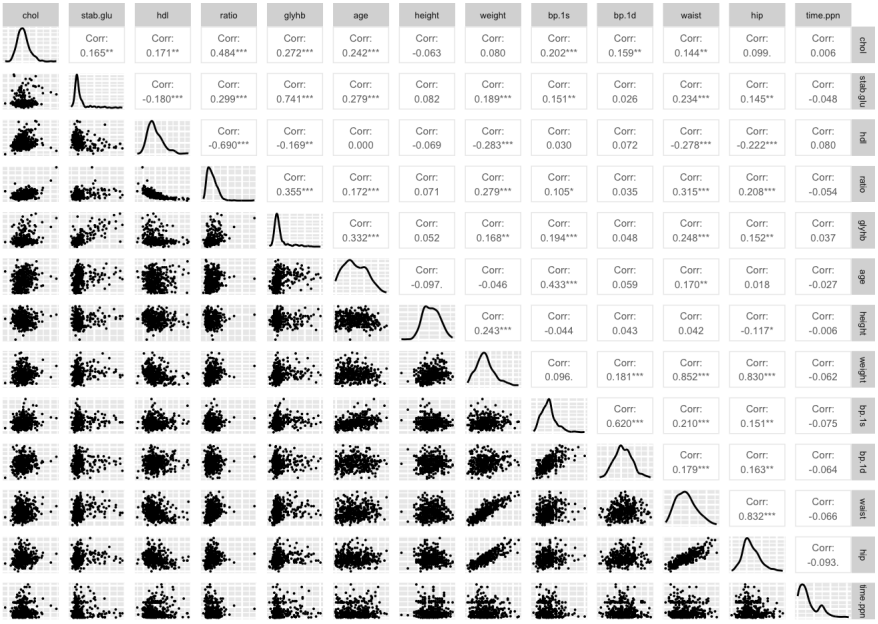
Pie Chart for location **Pie Chart for gender** **Pie Chart for frame**




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

Pie chart of location: There is approximately a 50/05 split between Louise and Buckingham.
Pie chart of gender: There is approximately a 60/40 split between female and male, respectively.
Pie chart of frame: There is approximately a 50/25/25 split between medium, large, and small, respectively.

```
# Scatter/Correlation matrix of variables
ggpairs(quant, progress = FALSE, lower = list(continuous = wrap('points', size = 0.02)), upper = list(continuous
= wrap('cor', size = 2))) + theme(axis.line=element_blank(), axis.text=element_blank(), axis.ticks=element_blank
(), strip.text.x = element_text(size = 5), strip.text.y = element_text(size = 5))
```



Important correlations:
Variables: chol & ratio –> moderate positive correlation Variables: stab.gl & glyhb (response) –> strong positive correlation
Variables: hdl & ratio –> moderate negative correlation
Variables: –> moderate positive correlation
Variables: weight & waist –> strong positive correlation
Variables: bp.1s & bp.1d –> moderate positive correlation
Variables: waist & hip –> strong positive correlation

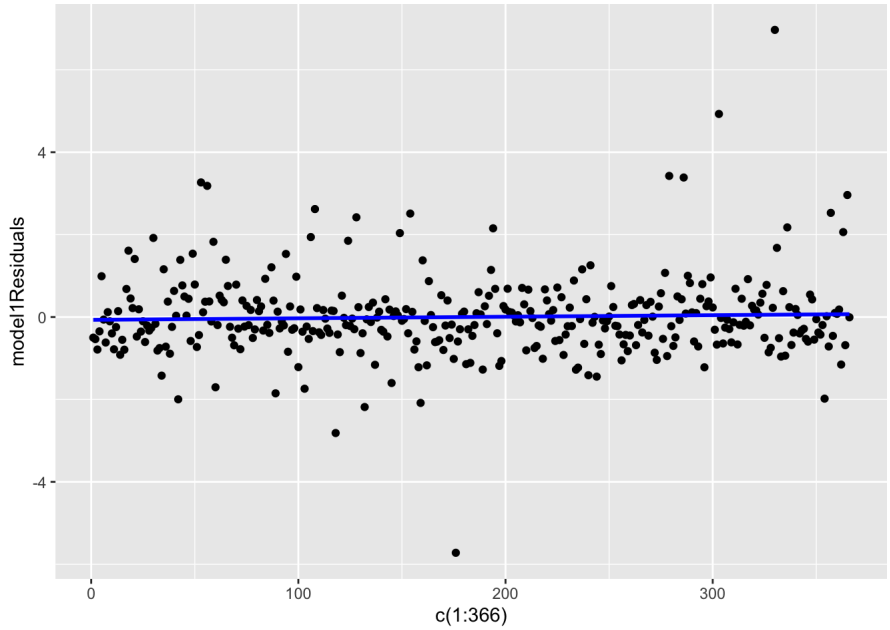
2. Regress glybh on all predictor variables (Model 1). Draw the diagnostic plots of the model and comment.

```
model1 <- lm(diabetes$glyhb ~ diabetes$chol + diabetes$stab.glu + diabetes$hdl + diabetes$ratio + diabetes$locati
on + diabetes$age + diabetes$gender + diabetes$height + diabetes$weight + diabetes$frame + diabetes$bp.1s + diabe
tes$bp.1d + diabetes$waist + diabetes$hip + diabetes$time.ppn, data = diabetes)

# Diagnostic Tests
model1_residuals <- rstandard(model1)
diabetes$model1Residuals = model1_residuals
print("This is the residual plot")
```

```
## [1] "This is the residual plot"
```

```
ggplot(diabetes, aes(x=c(1:366), y=model1Residuals)) + geom_point() + geom_smooth(method=lm , color="blue", se=FA
LSE)
```

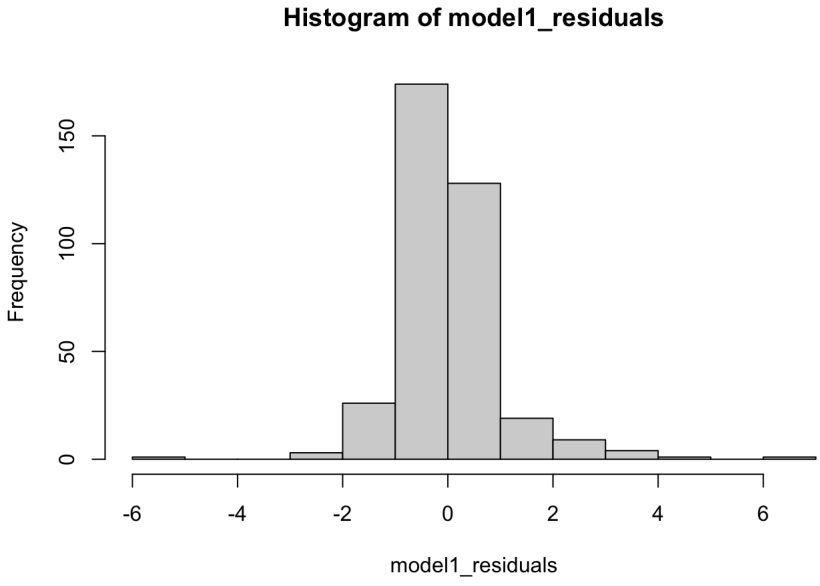
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
print("This is the histogram of the standardized residuals")
```

```
## [1] "This is the histogram of the standardized residuals"
```

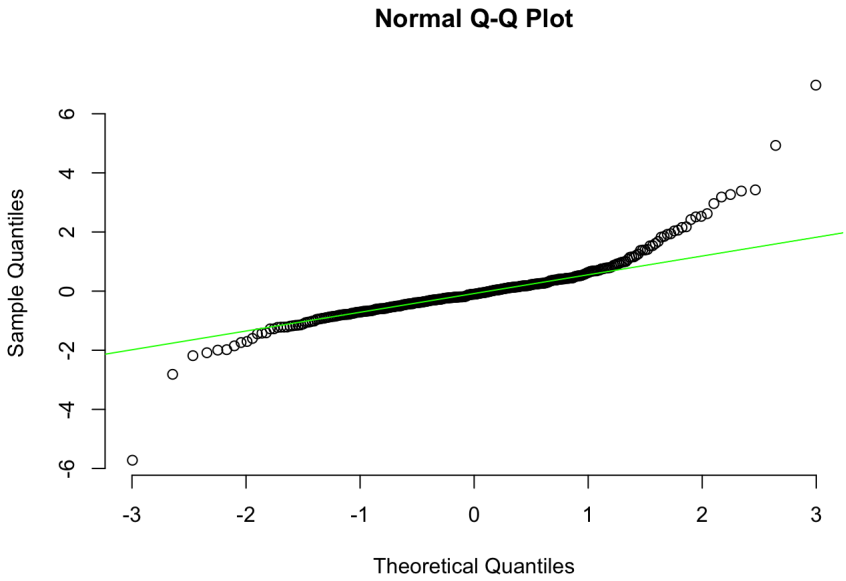
```
hist(model1_residuals)
```



```
print("This is the QQ plot of the standardized residuals")
```

```
## [1] "This is the QQ plot of the standardized residuals"
```

```
qqnorm(diabetes$model1Residuals, pch = 1, frame = FALSE)
qqline(diabetes$model1Residuals , col = "green")
```



From our diagnostic plots, we are able to tell the following: The equal variance assumption does not hold as we see slight heteroscedasticity of the residuals, despite centering around y = 0. The distribution of the residuals is Normally distributed but highly under dispersed. Similarly, we see light tails relative to the standard normal distribution in the QQ plot. I would not be very comfortable performing inference.

3. You want to check whether any transformation on the response variable is needed. You use the function 'boxcox' to help you make the decision. State the transformation you decide to use. In the following, we denote the transformed response variable to be $g\tilde{l}\tilde{y}hb$. Regress $g\tilde{l}\tilde{y}hb$ on all predictor variables (Model 2). Draw the diagnostic plots of this model and comment. Apply boxcox again on Model 2; what do you find?

```
# Transform Y
BC <- boxcox(model1, plotit = FALSE)
lambda <- BC$x[which.max(BC$y)]
Ytrans <- 1/diabetes$glyhb
transformed.data <- data.frame(YT = Ytrans, chol = diabetes$chol, stab.glu= diabetes$stab.glu, hdl= diabetes$hdl,
ratio= diabetes$ratio, location=diabetes$location, age= diabetes$age, gender= diabetes$gender, height=diabetes$he
ight, weight= diabetes$weight, frame= diabetes$frame, bp.1s=diabetes$bp.1s, bp.1d=diabetes$bp.1d, waist=diabetes
$waist, hip=diabetes$hip, time.ppn=diabetes$time.ppn)
```

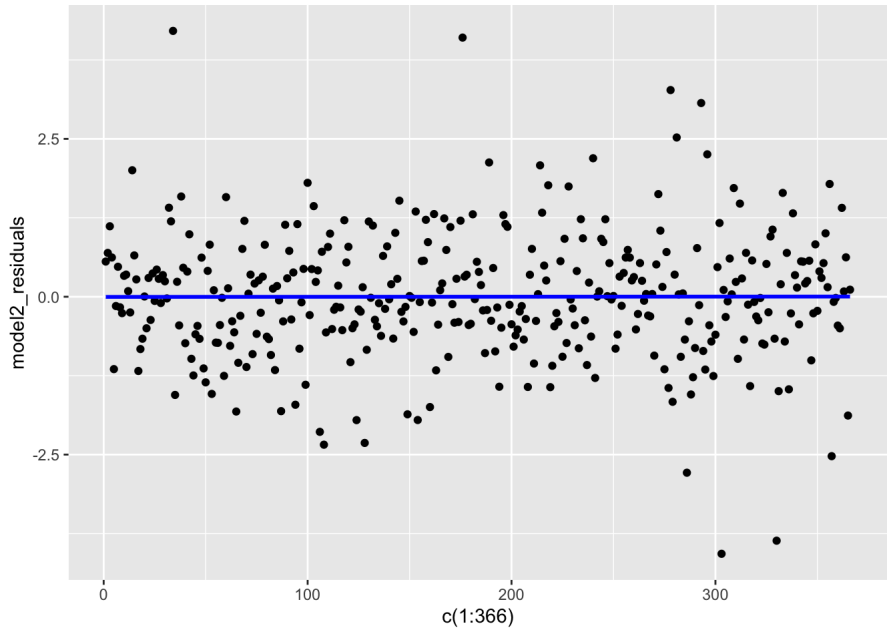
Using boxcox, the lambda value I found was -0.9. In general, boxcox maximizes the log-likelihood function and allows us to transform Y in a way which addresses the variance issues of our original data. However this lambda value makes our transformation hard to interpret value, so instead I performed the following transformation: $Y_T = 1/Y$.

```
# Regress new Y on all predictor variables
model2 <- lm(YT ~ transformed.data$chol + transformed.data$stab.glu + transformed.data$hdl + transformed.data$rat
io + transformed.data$location + transformed.data$age + transformed.data$gender + transformed.data$height + trans
formed.data$weight + transformed.data$frame + transformed.data$bp.1s + transformed.data$bp.1d + transformed.data
$waist + transformed.data$hip + transformed.data$time.ppn, data = transformed.data)

# Diagnostic Tests
model2_residuals <- rstandard(model2)

print("This is the transformed residual plot")
```

```
## [1] "This is the transformed residual plot"
```

```
ggplot(transformed.data, aes(x=c(1:366), y=model2_residuals)) + geom_point() + geom_smooth(method=lm , color="blu
e", se=FALSE)
```

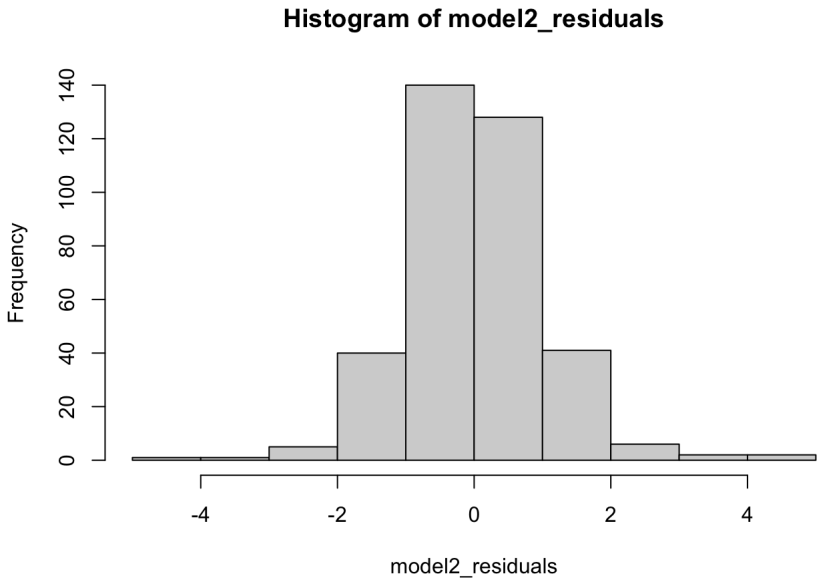
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
print("This is the histogram of the transformed standardized residuals")
```

```
## [1] "This is the histogram of the transformed standardized residuals"
```

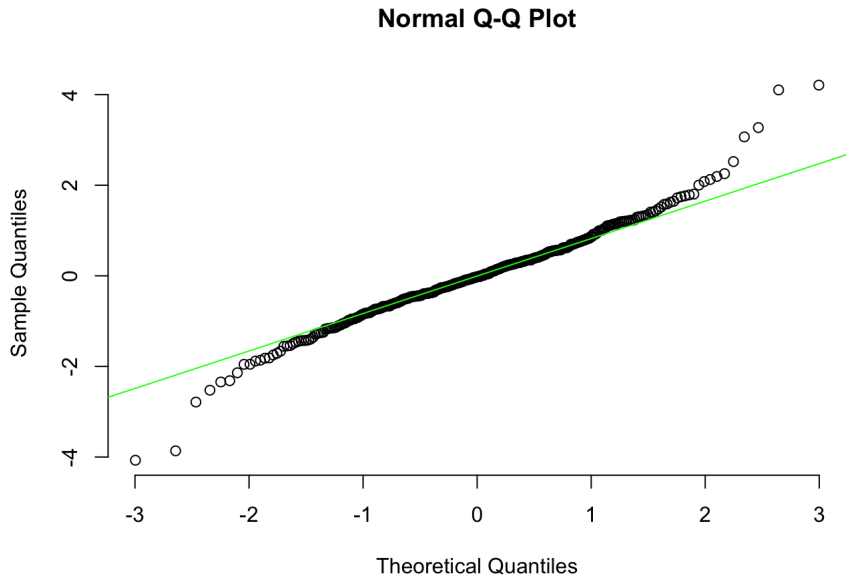
```
hist(model2_residuals)
```




```
print("This is the QQ plot of the transformed standardized residuals")
```

```
## [1] "This is the QQ plot of the transformed standardized residuals"
```

```
qqnorm(model2_residuals, pch = 1, frame = FALSE)
qqline(model2_residuals, col = "green")
```




```
# Transform Y using boxcox, AGAIN
BC2 <- boxcox(model2,plotit = FALSE)
lambda2 <- BC2$x[which.max(BC2$y)]
```

When I ran boxcox again, the lambda value chosen was $\lambda_2 = 0.9$. If I were to use this value and apply it to the transformation $Y_T = (Y^{\lambda_2} - 1)/lambda_2$, it would not have significant impact on our data, and therefore not be a useful transformation in terms of addressing our diagnostic tests.

4. Set the seed to "372" and randomly split data into two parts: a training data set (70%) and a validation data set (30%).

```
set.seed(372)
train <- slice_sample(.data = transformed.data, prop = .7)
validate <- anti_join(transformed.data, train)
```

```
## Joining with `by = join_by(YT, chol, stab.glu, hdl, ratio, location, age,
## gender, height, weight, frame, bp.1s, bp.1d, waist, hip, time.ppn)`
```

Selection of first-order effects

We now consider subsets selection from the pool of all first-order effects of the 15 predictors. $\tilde{glyhb}$ is used as the response variable for the following problems.

5. Fit a model with all first-order effects (Model 3). How many regression coefficients are there in this model? What is the MSE from Model 3?

There are 16 regression coefficients in model 3. The MSE of model 3 is 0.001411762

```
model3 <- lm(YT ~ ., data = train)
MSEmodel3 <- mean(summary(model3)$residuals^2)
summary(model3)
```

```
##
## Call:
## lm(formula = YT ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.14624 -0.02104 -0.00019  0.01995  0.14047
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4.874e-01  7.807e-02   6.243 1.95e-09 ***
## chol           -1.444e-04  1.186e-04  -1.217  0.22485
## stab.glu       -4.755e-04  5.184e-05  -9.171  < 2e-16 ***
## hdl             2.702e-04  4.010e-04   0.674  0.50103
## ratio          -6.638e-04  4.475e-03  -0.148  0.88222
## locationLouisa  2.999e-03  5.326e-03   0.563  0.57396
## age            -6.049e-04  2.017e-04  -2.999  0.00299 **
## gendermale      6.486e-03  8.356e-03   0.776  0.43836
## height         -1.882e-03  1.048e-03  -1.796  0.07372 .
## weight          2.058e-04  1.712e-04   1.202  0.23047
## framemedium    -7.094e-04  6.935e-03  -0.102  0.91861
## framesmall      3.213e-04  8.714e-03   0.037  0.97062
## bp.1s          -1.048e-04  1.530e-04  -0.685  0.49384
## bp.1d           9.821e-05  2.425e-04   0.405  0.68584
## waist          -1.395e-03  9.793e-04  -1.424  0.15567
## hip            -1.011e-03  1.097e-03  -0.921  0.35782
## time.ppn       -1.487e-05  8.331e-06  -1.784  0.07563 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03889 on 239 degrees of freedom
## Multiple R-squared:  0.479,  Adjusted R-squared:  0.4442
## F-statistic: 13.74 on 16 and 239 DF,  p-value: < 2.2e-16
```

6. Consider best subsets selection using the R function regsubsets() from the leaps library with Model 3 as the full model. Return the top 1 best subset of all subset sizes (i.e., number of X variables) up to 16 (because frame has 3 levels). Compute SSE_p , ${R_p}^2$, ${R_{a,p}}^2$, C_p, AIC_p, and BIC_p for each of models, as well as the intercept-only model. Identify the best model according to each criterion. For the best model according to C_p, what do you observe about its C_p value? Do you have a possible explanation for it?

See output for criteria calculations.

```
# Intercept only model
intercept_only <- lm(YT ~ 1, data = train)
SSEp_intercept <- sum(intercept_only$residuals^2) # 0.6937511
R2_intercept <- summary(intercept_only)$r.squared # 0
R2adj_intercept <- summary(intercept_only)$adj.r.squared # 0
Cp_intercept <- ols_mallows_cp(intercept_only, model3) # 204.7754
AIC_intercept <- 256*log(SSEp_intercept) + 2*(1) # -91.60435
BIC_intercept <- 256*log(SSEp_intercept) + (1)*log(256) # -88.05917
```

```
all.models <- regsubsets(YT ~ train$chol + train$stab.glu + train$hdl + train$ratio + train$location + train$age
+ train$gender + train$height + train$weight + train$frame + train$bp.1s + train$bp.1d + train$waist + train$hip
+ train$time.ppn, data = train, nbest=1, nvmax=16)
summary_stuff <- summary(all.models)
names_of_data <- c("YT",colnames(summary_stuff$which)[-1])
n <- nrow(train)
K <- nrow(summary_stuff$which)
nicer <- lapply(1:K,function(i){
  model <- paste(names_of_data[summary_stuff$which[i,]],collapse = ",")
  R2 <- summary_stuff$rsq[i]
  R2adj <- summary_stuff$adjr2[i]
  p <- sum(summary_stuff$which[i,])
  SSE <- summary_stuff$rss[i]
  BIC <- summary_stuff$bic[i]
  AIC <- summary_stuff$bic[i] - (log(n)* p) + 2*p
  CP <- summary_stuff$cp[i]
  results <- data.frame(model,p,CP,AIC, BIC, R2, R2adj, SSE)
  return(results)
})
nicer <- Reduce(rbind,nicer)
nicer
```

```
##
model
## 1
YT,train$stab.glu
## 2
YT,train$stab.glu,train$age
## 3
YT,train$stab.glu,train$age,train$waist
## 4
YT,train$stab.glu,train$ratio,train$age,train$waist
## 5
YT,train$stab.glu,train$ratio,train$age,train$waist,train$time.ppn
## 6
YT,train$chol,train$stab.glu,train$hdl,train$age,train$waist,train$time.ppn
## 7
YT,train$chol,train$stab.glu,train$hdl,train$age,train$height,train$waist,train$time.ppn
## 8
YT,train$chol,train$stab.glu,train$hdl,train$age,train$gendermale,train$height,train$waist,train$time.ppn
## 9                                                                                                    YT,tr
ain$chol,train$stab.glu,train$hdl,train$age,train$gendermale,train$height,train$weight,train$waist,train$time.ppn
## 10                                                                                                  YT,train$chol,t
rain$stab.glu,train$hdl,train$age,train$gendermale,train$height,train$weight,train$waist,train$hip,train$time.ppn
## 11                                                                                YT,train$chol,train$stab.glu,train$h
dl,train$locationLouisa,train$age,train$gendermale,train$height,train$weight,train$waist,train$hip,train$time.ppn
## 12                                                                             YT,train$chol,train$stab.glu,train$hdl,train$loc
ationLouisa,train$age,train$gendermale,train$height,train$weight,train$bp.1s,train$waist,train$hip,train$time.ppn
## 13                                                                    YT,train$chol,train$stab.glu,train$hdl,train$locationLouisa,
train$age,train$gendermale,train$height,train$weight,train$bp.1s,train$bp.1d,train$waist,train$hip,train$time.ppn
## 14                                                YT,train$chol,train$stab.glu,train$hdl,train$locationLouisa,train$age,train$ge
ndermale,train$height,train$weight,train$framemedium,train$bp.1s,train$bp.1d,train$waist,train$hip,train$time.ppn
## 15                                    YT,train$chol,train$stab.glu,train$hdl,train$ratio,train$locationLouisa,train$age,train$ge
ndermale,train$height,train$weight,train$framemedium,train$bp.1s,train$bp.1d,train$waist,train$hip,train$time.ppn
## 16 YT,train$chol,train$stab.glu,train$hdl,train$ratio,train$locationLouisa,train$age,train$gendermale,train$he
ight,train$weight,train$framemedium,train$framesmall,train$bp.1s,train$bp.1d,train$waist,train$hip,train$time.ppn
##     p        CP       AIC        BIC        R2     R2adj       SSE
## 1   2 36.479326 -114.7680 -107.67763 0.3711971 0.3687215 0.4362327
## 2   3 14.830655 -134.6644 -124.02886 0.4227445 0.4181812 0.4004717
## 3   4  6.609973 -142.7400 -128.55926 0.4450227 0.4384158 0.3850162
## 4   5  3.435598 -145.9962 -128.27030 0.4563013 0.4476368 0.3771916
## 5   6  1.964092 -147.5841 -126.31301 0.4638682 0.4531456 0.3719420
## 6   7  2.395340 -147.2221 -122.40582 0.4672877 0.4544512 0.3695698
## 7   8  3.302529 -146.3693 -118.00791 0.4696697 0.4547007 0.3679173
## 8   9  3.557832 -146.2117 -114.30510 0.4734726 0.4564191 0.3652790
## 9  10  4.695419 -145.1273 -109.67553 0.4753524 0.4561580 0.3639748
## 10 11  5.925727 -143.9473 -104.95030 0.4770301 0.4556844 0.3628109
## 11 12  7.530778 -142.3690  -99.82688 0.4778910 0.4543533 0.3622137
## 12 13  9.210649 -140.7114  -94.62407 0.4785888 0.4528401 0.3617296
## 13 14 11.054009 -138.8791  -89.24658 0.4789302 0.4509389 0.3614927
## 14 15 13.023085 -136.9122  -83.73452 0.4789976 0.4487319 0.3614460
## 15 16 15.001360 -134.9355  -78.21261 0.4790450 0.4464853 0.3614131
## 16 17 17.000000 -132.9369  -72.66889 0.4790480 0.4441725 0.3614111
```

The best model according to each criterion is as follows:
SSE_p: YT ~ . (full model)
${R_p}^2$: YT ~ . (full model)
${R_{a,p}}^2$: YT ~ chol + stab.glu + hdl + age + gender{male} + height + waist + time.ppn
C_p: YT ~ chol + stab.glu + hdl + ratio + location{Louisa} + age, + gender{male} + height + weight + frame{medium} + bp.1s + bp.1d + waist + hip, + time.ppn
AIC_p: YT ~ stab.glu + ratio + age + waist + time.ppn
BIC_p: YT ~ stab.glu + age + waist

For the best model according to C_p, we notice that the best model is the full model. This is because in the full model, C_p is always equal to p*(n-1) - 2, which simplifies to p, making p = C_p always true. To account for this, we only look at the first 15 models when assessing C_p.

```
which.min(nicer$SSE)
```

```
## [1] 16
```

```
which.max(nicer$R2)
```

```
## [1] 16
```

```
which.max(nicer$R2adj)
```

```
## [1] 8
```

```
which.min(abs(nicer[0:15,]$CP - nicer[0:15,]$p))
```

```
## [1] 15
```

```
which.min(nicer$AIC)
```

```
## [1] 5
```

```
which.min(nicer$BIC)
```

```
## [1] 3
```

7. Denote the best models according to AIC, BIC, and adjusted R^2 as Models 3.1, 3.2, and 3.3, respectively. It is possible that some of the three models are the same. We will examine these later on.

```
model3.1 <- lm(YT ~ train$stab.glu + train$ratio + train$age + train$waist + train$time.ppn, data = train) # AIC
model3.2 <- lm(YT ~ train$stab.glu + train$age + train$waist, data = train) # BIC
model3.3 <- lm(YT ~ train$chol + train$stab.glu + train$hdl + train$age + train$gender + train$height + train$wai
st + train$time.ppn, data = train) #R2adj
```

Selection of first- and second-order effects

We now consider subset selection from the pool of first- and second- order effects as well as 2-way interactions between the 15 predictors.

8. Fit a model with all first-order and 2-way interaction effects (Model 4). How many regression coefficients are there in this model? What is the MSE from this model? Do you have any concern about the fit of this model? If yes, why?

There are 136 coefficients in the model. The MSE of this model is 0.0006005911 The MSE is very low, but a small MSE does not necessarily guarantee a perfect model. In fact, our model might be over fitted. To test this, we will perform model validation.

```
model4 <- lm(YT ~ .^2, data = train)
length(model4$coefficients)
```

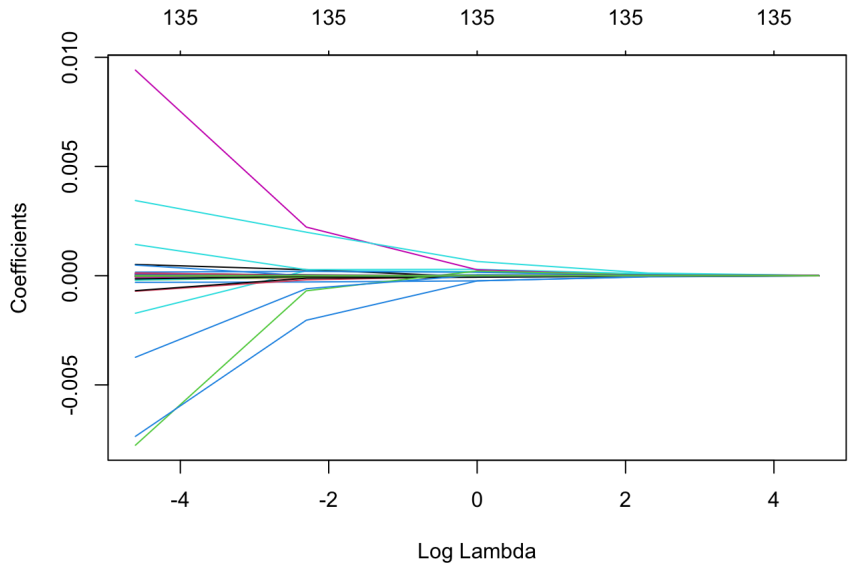
```
## [1] 136
```

```
MSEmodel4 <- mean(summary(model4)$residuals^2)
```

9. Apply ridge regression. Consider the penalty parameters λ = 0.01, 0.1, 1, 10, 100. Use cross validation to select the best value of λ. What is the model being selected with this λ value? Name this model Model 4.1.

After applying ridge regression and using cross validation, I found that $\lambda = 0.1$ to be the best value of λ. The model being selected with this λ value is model 4.1 a full model with 136 predictors with adjusted beta values found in Coef.Ridge.

```
lambda.vec <- c(0.01, 0.1, 1, 10, 100)
x_ridge <- model.matrix(YT ~.^2, data = train)[,-1]
y_ridge <- train$YT
ridge.model <- glmnet(x_ridge, y_ridge, alpha = 0, lambda = lambda.vec)
plot(ridge.model, xvar = "lambda")
```




```
# Cross validation
set.seed(372)
cv.out.ridge <- cv.glmnet(x_ridge, y_ridge, alpha = 0, lambda = lambda.vec)

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

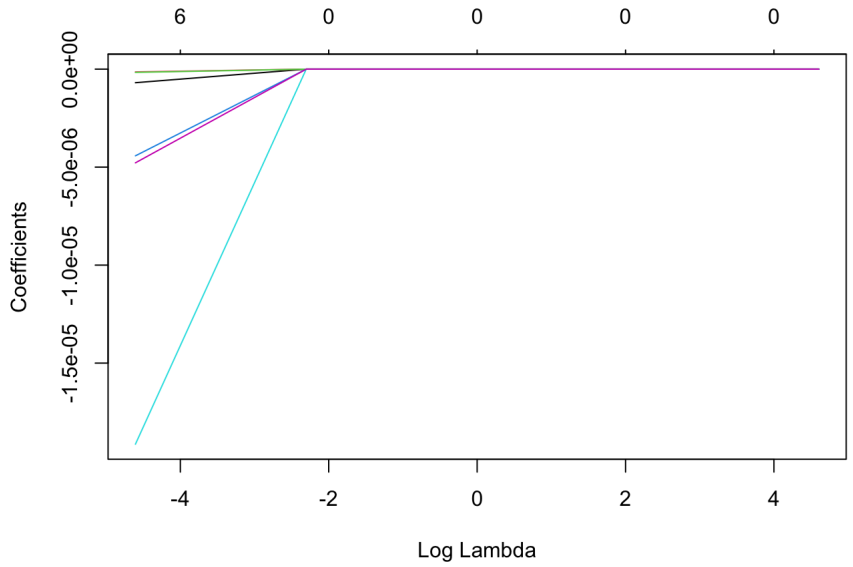
```
## [1] 0.1
```

```
#Fit the final model to the entire data set using the chosen lambda
model4.1 <- glmnet(x_ridge, y_ridge, alpha = 0, lambda = best.lambda.ridge)
Coef.Ridge <- coef(model4.1)[1:136, ] # View beta values in full ridge model: Coef.Ridge[Coef.Ridge != 0]
```

10. Apply LASSO regression. Consider the penalty parameters λ = 0.01, 0.1, 1, 10 100. Use cross validation to select the best value of λ. What is the model being selected with this λ value? Name this model Model 4.2.

After applying LASSO regression and using cross validation, I found that $\lambda = 0.01$ to be the best value of λ. The model being selected with this λ value is model 4.2 such that
$Y\ 3.460e-01 + 1.399e-04(chol) + (-8.725e-04)(stab.glu) + (-6.913e-04)(age) + (-5.365e-04)(ratio) + (-1.480e-04)(bp.1s) + (-6.779e$
$+(-3.103e-05)(age : ratio) + (-1.734e-05)(age : waist)$

```
x_LASSO <- model.matrix(YT ~.^2, data = train)[,-1]
y_LASSO <- train$YT
LASSO.model <- glmnet(x_LASSO, y_LASSO, alpha = 1, lambda = lambda.vec)
plot(LASSO.model, xvar <- "lambda")
```




```
# Cross validation
set.seed(372)
cv.out.LASSO = cv.glmnet(x_LASSO, y_LASSO, alpha = 1, lambda = lambda.vec)

#Find the best lambda value
best.lambda.LASSO <- cv.out.LASSO$lambda.min
best.lambda.LASSO
```

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

```
#Fit the final model to the entire data set using the chosen lambda
temp_LASSO <- glmnet(x_LASSO, y_LASSO, alpha = 1, lambda = best.lambda.LASSO)
Coef.LASSO <- coef(temp_LASSO)[1:136, ]
Coef.LASSO[Coef.LASSO != 0]
```

```
##    (Intercept)  chol:stab.glu   stab.glu:age stab.glu:bp.1s stab.glu:waist
##   2.470986e-01  -6.924052e-07  -1.420930e-07  -1.657606e-07  -4.424060e-06
##      ratio:age      age:waist
##  -1.914001e-05  -4.781876e-06
```

```
model4.2 <- lm(YT ~ 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 = train)
summary(model4.2)
```

```
##
## Call:
## lm(formula = YT ~ 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 = train)
##
## 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
```

11. Discuss the difference in methodologies behind ridge and LASSO regression. Why do they result in such different models?

Ridge regression asymptotically decreases the magnitude of beta coefficients as λ increases, while LASSO regression snaps them to 0. This difference allows us to use LASSO regression in model selection and find a model without redundant/unnecessary predictors, because their coefficients have been snapped to 0.

Model validation

We now consider validation of the models (Model 3.1, Model 3.2, Model 3.3, Models 4.1, Models 4.2) you selected in the previous studies.

12. Internal validation. We use PRESS for this purpose. Calculate PRESS for each of these models. Comment.

PRESS of Model 3.1: 0.3944713
PRESS of Model 3.2: 0.4022217
PRESS of Model 3.3: 0.395689
PRESS of Models 4.1: 0.4092355
PRESS of Models 4.2: 0.4049749

PRESS gives us insight into the predictive power of the model on unseen data. A lower PRESS value indicates better predictive ability, such that the model is likely to perform well on new data, not just the data it was trained on. PRESS also addresses potential over fitting of a model, which was a problem when interpreting MSE earlier, because it is calculated using a 'leave-one-out' cross-validation approach.

From the values above, we can conclude that Model 3.1 has the greatest predictive power and is the best performing model out of the models above, in terms of predicting the left out observation. This also suggests that model ??? has the least prediction error on new or unseen data.

```
# Note: adapted from Cody's OH, my own implementation in a for loop below
# PRESS function for ridge
PRESS_ridge_fucnt <- function(index) {
  xtrain_exempt <- x_ridge[-index,]
  ytrain_exempt <- y_ridge[-index]
  xtest <- x_ridge[index, ]
  ytest <- y_ridge[index]

  exempt_model <- glmnet(xtrain_exempt, ytrain_exempt, alpha = 0, lambda = best.lambda.ridge)

  exempt_predict <- predict(exempt_model, lambda = best.lambda.ridge, newx = xtest)

  error2 = (ytest - exempt_predict)^2

  return(error2)
}
```

```
PRESSmodel3.1 <- PRESS(model3.1) # Model 3.1
PRESSmodel3.2 <- PRESS(model3.2) # Model 3.2
PRESSmodel3.3 <- PRESS(model3.3) # Model 3.3
# Model 4.1
PRESSmodel4.1 <- 0
for (i in 1:nrow(train)) {
  PRESSmodel4.1 = PRESSmodel4.1 + PRESS_ridge_fucnt(i)
}
PRESSmodel4.2 <- PRESS(model4.2) # Model 4.2
```

13. External validation using the validation set. For each of these models (Model 3.1, Model 3.2, Model 3.3, Model 4.1, Model 4.2), calculate the mean squared prediction error (MSPE), i.e., you use the model to predict the 110 observations in the validation set and calculate the averaged squared prediction error. How do these MSPEs compare with the respective PRESS/n (here n is the sample size of the training data, i.e., 256). Which model has the smallest MSPE?

MSPE of Model 3.1: 0.003498903
MSPE of Model 3.1 is greater than its respective PRESS/n where n = 256
MSPE of Model 3.2: 0.003478664
MSPE of Model 3.2 is greater than its respective PRESS/n where n = 256
MSPE of Model 3.3: 0.003516655
MSPE of Model 3.3 is greater than its respective PRESS/n where n = 256

WRONG

MSPE of Model 4.1: 0.0009455666
MSPE of Model 4.1 is ???? than than its respective PRESS/n where n = 256
MSPE of Model 4.2: 0.003565263
MSPE of Model 4.2 is greater than its respective PRESS/n where n = 256

The model with the smallest MSPE is ????.
WRONG

```
y_test <- validate$YT
x_test <- model.matrix(YT ~.^2, data = validate)[,-1]

# Model 3.1
MSPEmodel3.1 <- mean((y_test - predict.lm(model3.1, validate)) ^ 2)
```

```
## Warning: 'newdata' had 110 rows but variables found have 256 rows
```

```
## Warning in y_test - predict.lm(model3.1, validate): longer object length is not
## a multiple of shorter object length
```

```
# Model 3.2
MSPEmodel3.2 <- mean((y_test - predict.lm(model3.2, validate)) ^ 2)
```

```
## Warning: 'newdata' had 110 rows but variables found have 256 rows
```

```
## Warning in y_test - predict.lm(model3.2, validate): longer object length is not
## a multiple of shorter object length
```

```
# Model 3.3
MSPEmodel3.3 <- mean((y_test - predict.lm(model3.3, validate)) ^ 2)
```

```
## Warning: 'newdata' had 110 rows but variables found have 256 rows
```

```
## Warning in y_test - predict.lm(model3.3, validate): longer object length is not
## a multiple of shorter object length
```

```
##### ERRORS #####
# Model 4.1
# mspe_pred_ridge <- predict(model4.1, s = best.lambda.ridge, newx = x_test)
# MSPEmodel4.1 <- mean((mspe_pred_ridge - y_test)^2)
##### ERRORS #####

# Model 4.2
mspe_pred_LASSO <- predict(model4.2, a = best.lambda.LASSO, newx = x_test)
MSPEmodel4.2 <- mean((mspe_pred_LASSO - y_test)^2)
```

```
## Warning in mspe_pred_LASSO - y_test: longer object length is not a multiple of
## shorter object length
```

```
cat("Model 3.1\n")
```

```
## Model 3.1
```

```
cat("Internal: " , PRESSmodel3.1 / 256, "     External: ", MSPEmodel3.1, "\n")
```

```
## Internal:  0.001540903      External:  0.003498903
```

```
cat("Model 3.2\n")
```

```
## Model 3.2
```

```
cat("Internal: " , PRESSmodel3.2 / 256, "     External: ", MSPEmodel3.2, "\n")
```

```
## Internal:  0.001571179      External:  0.003478664
```

```
cat("Model 3.3\n")
```

```
## Model 3.3
```

```
cat("Internal: " , PRESSmodel3.3 / 256, "     External: ", MSPEmodel3.3, "\n")
```

```
## Internal:  0.00154566      External:  0.003516655
```

```
cat("Model 4.1\n")
```

```
## Model 4.1
```

```
cat("Internal: " , PRESSmodel4.1 / 256, "     External: SEE ERROR DESCRIPTION BELOW\n")
```

```
## Internal:  0.001598576      External: SEE ERROR DESCRIPTION BELOW
```

```
cat("Model 4.2\n")
```

```
## Model 4.2
```

```
cat("Internal: " , PRESSmodel4.2 / 256, "     External: ", MSPEmodel4.2, "\n")
```

```
## Internal:  0.001581933      External:  0.003565263
```

ERROR DESCRIPTION: - I faced challenged when calculating external validation (MSPE) on Model 4.1. I have left my commented out code in the model 4.1 section to see what I attempted. I ran into either (1) a variable number error, because my validation set did not contain all 135 predictors that model 4.1 is made with or (2) would generate a very very small number that did not seem to fit into the other MSPE calculations. I chose to exclude this statistic for the sake of argument towards my final model, but would love to know what I did wrong.

14. Based on both internal and external validation, which model you would choose as the final model? Fit the final model using the entire data set (training and validation combined) (Model 5). Write down the fitted regression function and report the R summary(). Give a complete interpretation of your model in terms of the real life context of the problem.

Based on my results from the internal and external validation, the final model selected is model 3.2. This is because it has the smallest MSPE, and therefore has the strongest prediction power on external data. From model 5, I have concluded that stab.glu, age, and waist are the most significant predictors on Glycosolated Hemoglobin levels in African Americans, and therefore are great indicators for diabetes diagnosis.

```
# Entire dataset (transformed Y aka. all rows from validate + all rows from train)
final_validate <- data.frame(validate[,])
final_full <- rbind(final_validate,train[,])

# Fit the final model using final data set
model5 <- lm(YT ~ stab.glu + age + waist, data = final_full)
summary(model5)
```

```
##
## Call:
## lm(formula = YT ~ stab.glu + age + waist, data = final_full)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.151205 -0.022804 -0.001137  0.019969  0.160914
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.361e-01  1.320e-02  25.461  < 2e-16 ***
## stab.glu    -5.186e-04  3.787e-05 -13.693  < 2e-16 ***
## age         -6.825e-04  1.241e-04  -5.501 7.14e-08 ***
## waist       -1.359e-03  3.438e-04  -3.954 9.24e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03686 on 362 degrees of freedom
## Multiple R-squared:  0.4873, Adjusted R-squared:  0.483
## F-statistic: 114.7 on 3 and 362 DF,  p-value: < 2.2e-16
```