### STA 206 HW8

**1、**

1. False . It is Ra, not Rp
2. False .
3. True .
4. True
5. False . It depends on the number of n.
6. False . It may find the best model, but it is uncertainly.

**2、**

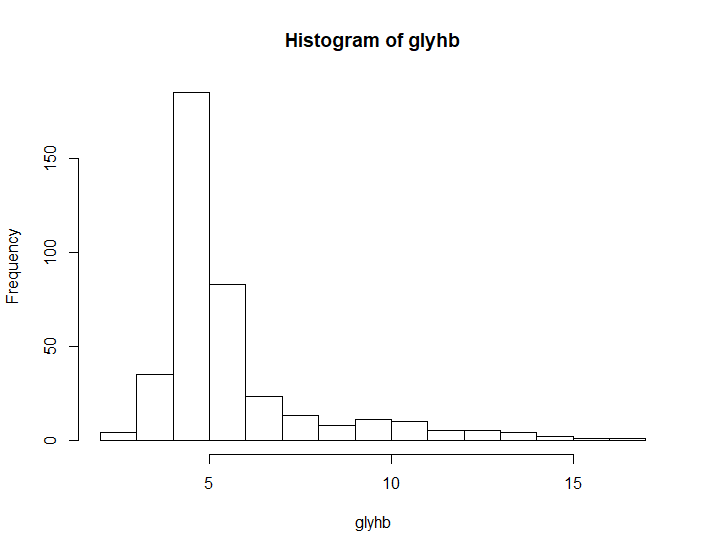
1. dataOri = read.table("d:\\diabetes.txt",header = T,as.is = T)

dataOri$frame[which(dataOri$frame=="")]="NA"

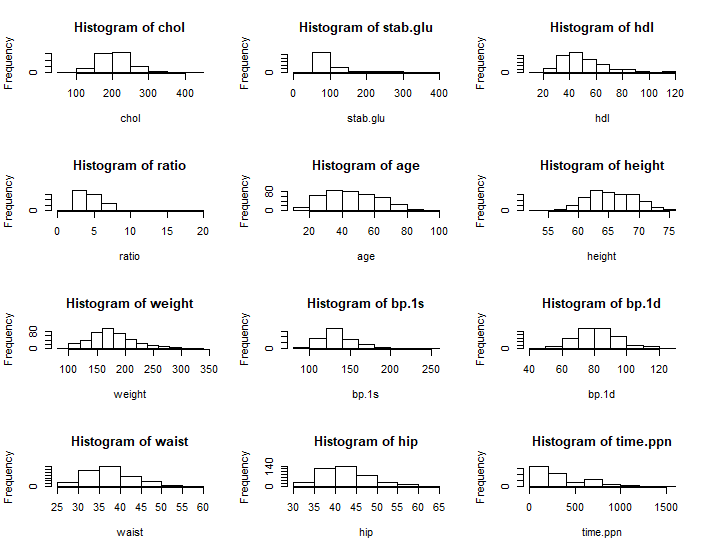
1. drops=c("id","bp.2s", "bp.2d")

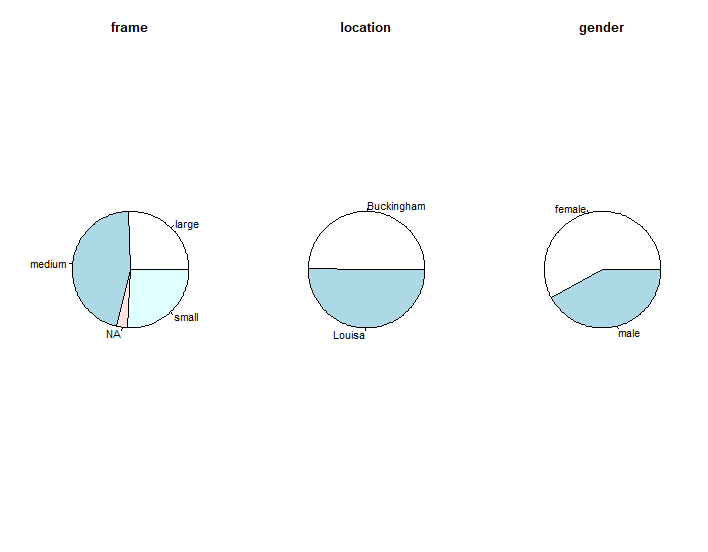
data=dataOri[,!(names(dataOri)%in%drops)]

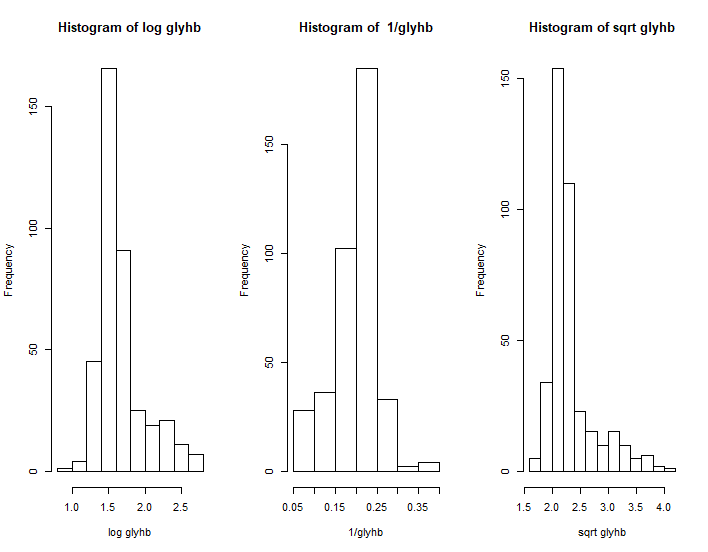
1. The frame, gender and location are qualitative variables. The others are quantitative variables.



It is a right heavy tail distribution.





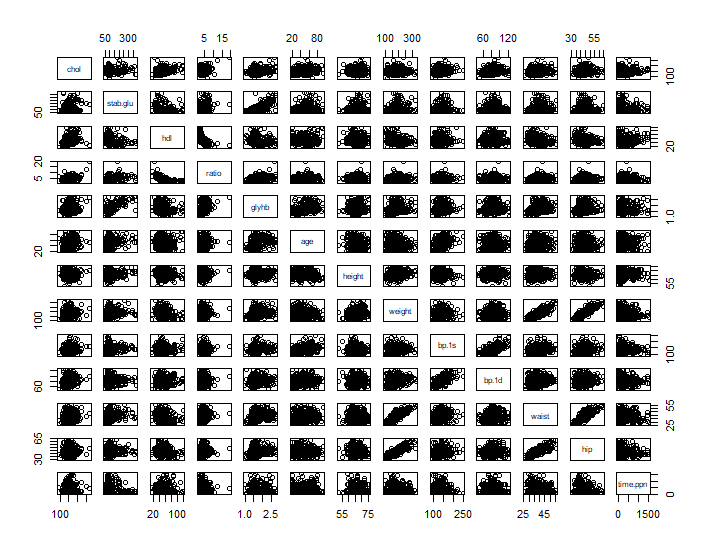


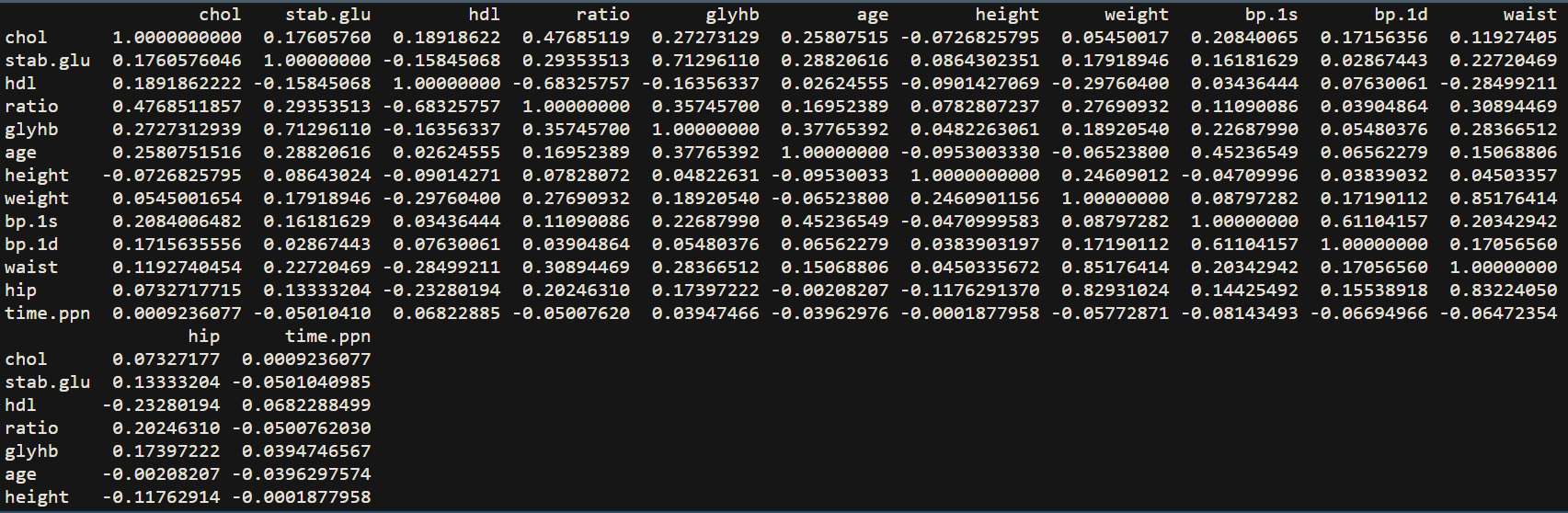
I think the 1/glyhb transformation is the best.

1. data$glyhb = lg.glyhb

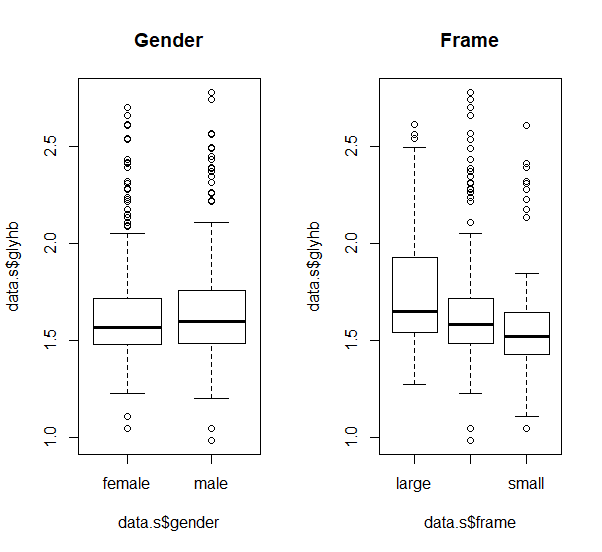
Replace the column glyhb in data by glyhb\*

1. Drop all the cases having missing value





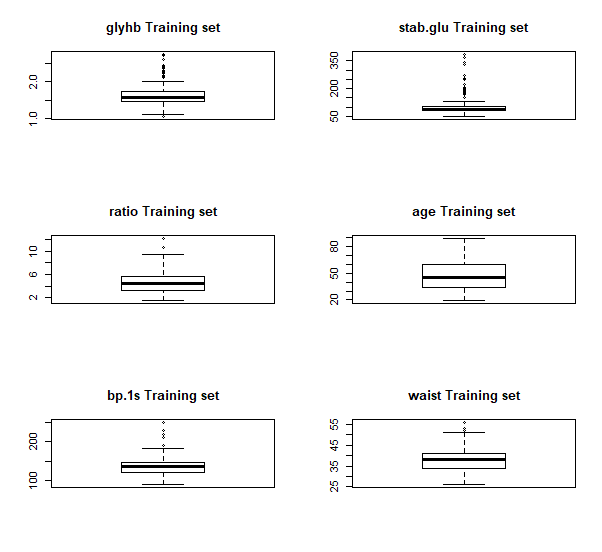
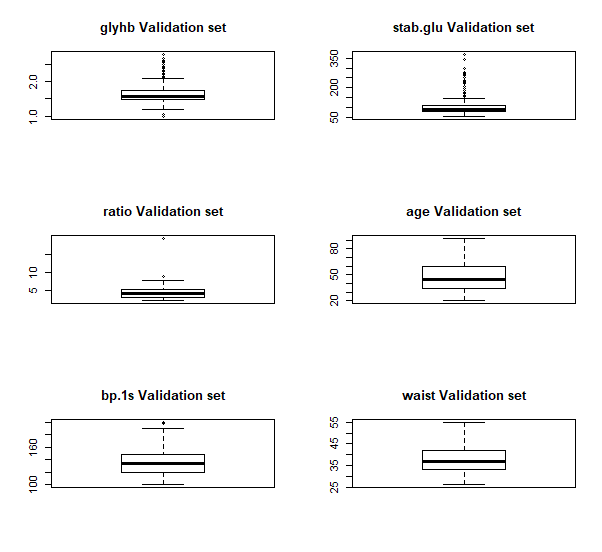
I do not observe the nonlinearity.



The value of glyhb in male is a little bit higher than the glyhb in female.

The value of glyhb will be smaller if the frame is small.

1. Randomly split data into two equal sets. The training data set and the validation data set.

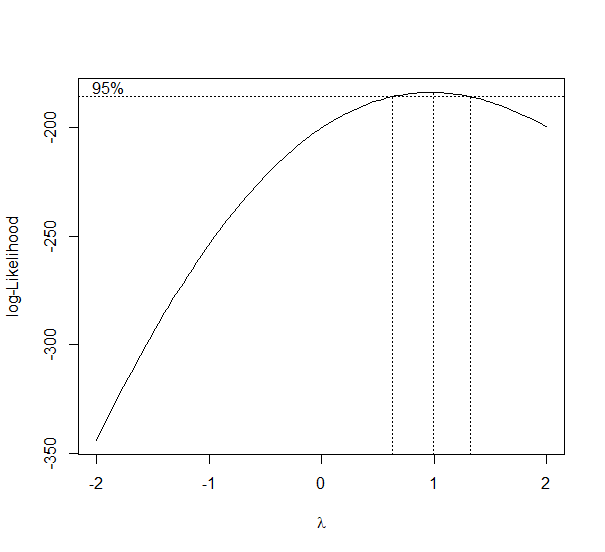


Its have the similar distribution between training set and validation set.

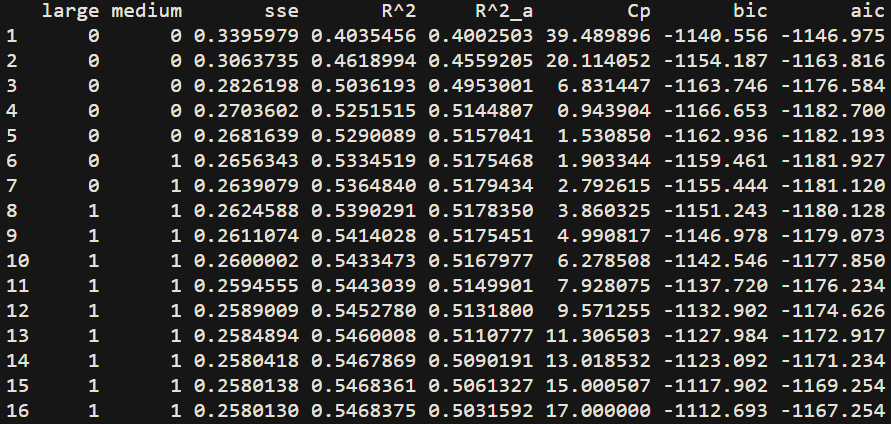
**3、**

1. There are 17 regression coefficients in this model. One indicator variable for gender, one indicator variable for location and two indicator variables for frame.

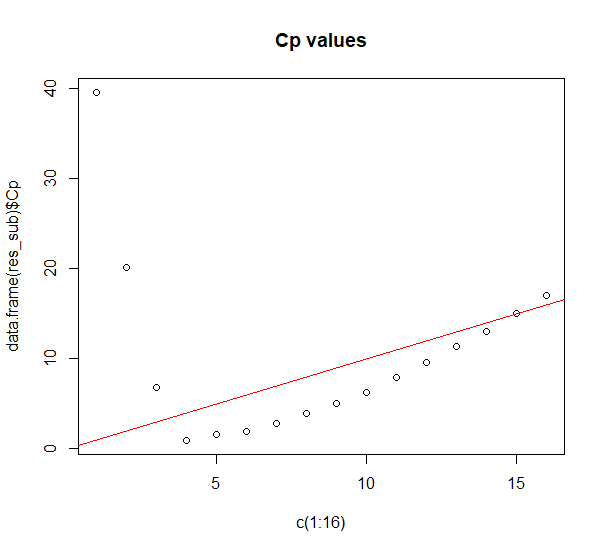
MSE = 0.001554



No, we do not need to transformation.



The best model chooses by using **R^2\_a** is the 7th model, the Ra^2 is 0.5179 .



The best model chooses by **Cp** is the 12th model.

The best model chooses by **bic** is the 4th model.

The best model chooses by **aic** is the 4th model.

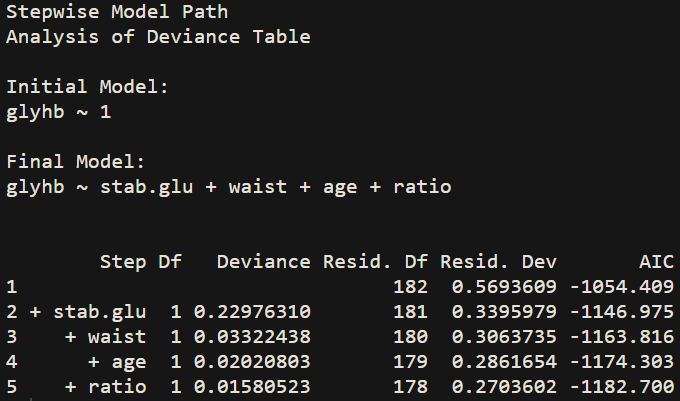
The value of Cp is big with small variables but smaller when the variable increases and becomes big when there are too many variables.

When there is a few variables, the bias of the model is huge, so the value of Cp is big.

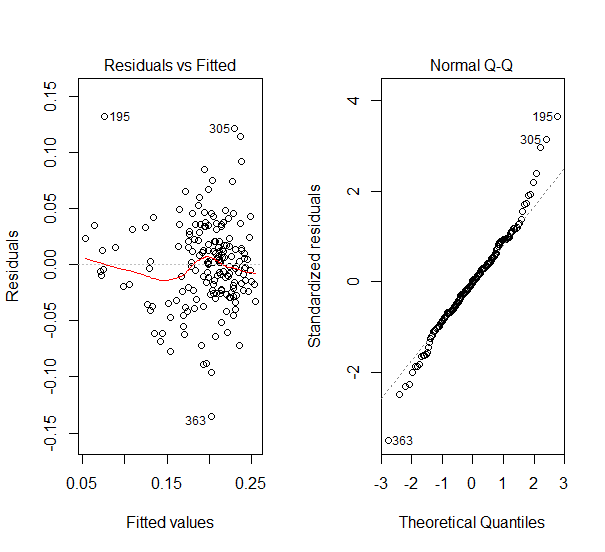
With the variable increasing, the bias of model became small, the variance of model increasing slower than the bias, so, the Cp became small.

At last, the bias of model decreases to around 0 but the variance became larger, so, the value of Cp became large again.

1. The model which be selected is : glyhb ~ stab.glu + ratio + age + waist



It is the best model according to AICp criterion.



The variance of residuals seems not the same.

It seems linearity.

But it seems not obey the normal distribution.

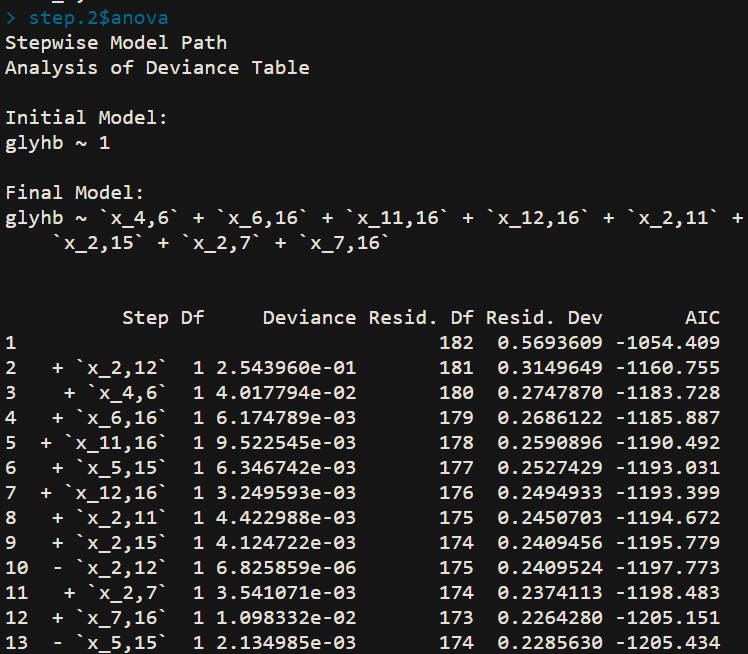
It seems not adequate.

**4、**

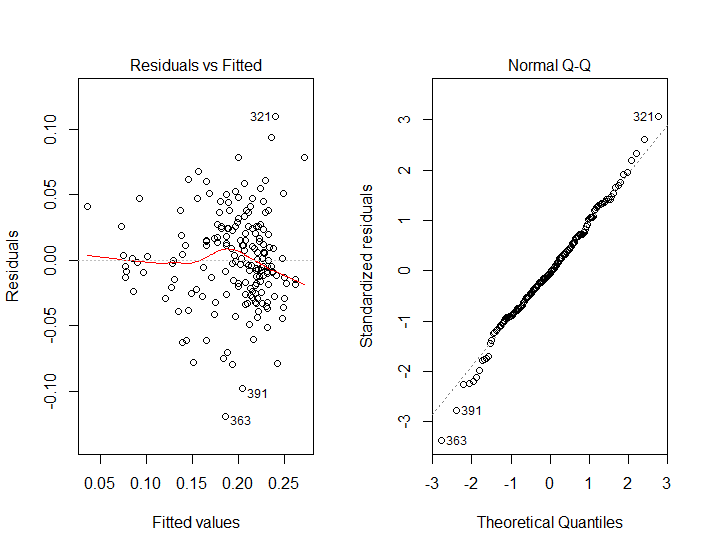
1. There are 120 + 16 +16 + 1= 153 variables in this model.

MSE = 0.0009803161

I concern the the model is too over-fit with this data. Because there are too many variables in this model.



It has smaller AICp

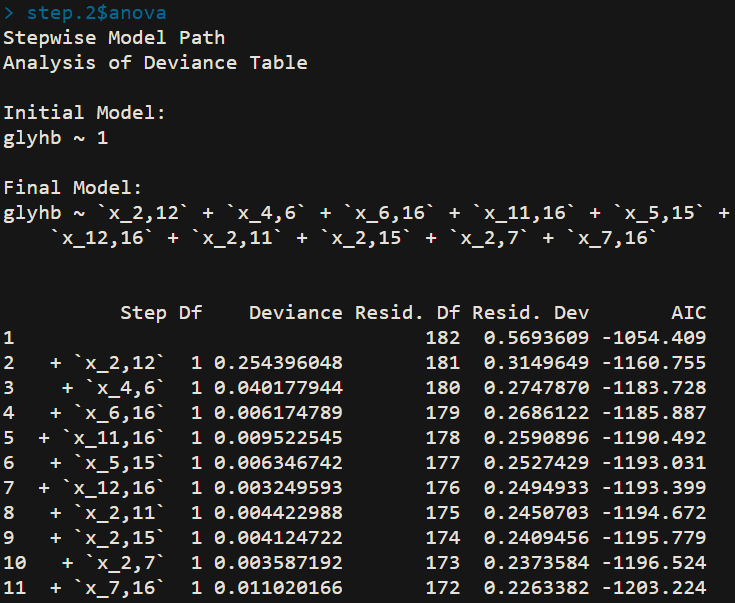


It has no non-linearity and the variance is not the same.

The model obey the normal-distribution.

I think it is a good adequate.

(4)



It is not as good as the model which is selected by using forward setpwise produce.

#### R\_code

dataOri = read.table("d:\\diabetes.txt",header = T,as.is = T)

dataOri$frame[which(dataOri$frame=="")] = NA

drops=c("id","bp.2s", "bp.2d")

data=dataOri[,!(names(dataOri)%in%drops)]

hist(data$glyhb,main = paste("Histogram of glyhb"),xlab = "glyhb")

par(mfrow = c(4,3))

hist(data$chol,main = paste("Histogram of chol"),xlab = "chol")

hist(data$stab.glu,main = paste("Histogram of stab.glu"),xlab = "stab.glu")

hist(data$hdl,main = paste("Histogram of hdl"),xlab = "hdl")

hist(data$ratio,main = paste("Histogram of ratio"),xlab = "ratio")

hist(data$age,main = paste("Histogram of age"),xlab = "age")

hist(data$height,main = paste("Histogram of height"),xlab = "height")

hist(data$weight,main = paste("Histogram of weight"),xlab = "weight")

hist(data$bp.1s,main = paste("Histogram of bp.1s"),xlab = "bp.1s")

hist(data$bp.1d,main = paste("Histogram of bp.1d"),xlab = "bp.1d")

hist(data$waist,main = paste("Histogram of waist"),xlab = "waist")

hist(data$hip,main = paste("Histogram of hip"),xlab = "hip")

hist(data$time.ppn,main = paste("Histogram of time.ppn"),xlab = "time.ppn")

par(mfrow = c(1,3))

pie(table(data$frame),main = "frame")

pie(table(data$location),main = "location")

pie(table(data$gender),main = "gender")

lg.glyhb = log(data$glyhb)

one.glyhb = 1 / data$glyhb

square.glyhb = sqrt(data$glyhb)

par(mfrow = c(1,3))

hist(lg.glyhb,main = paste("Histogram of log glyhb"),xlab = "log glyhb")

hist(one.glyhb,main = paste("Histogram of 1/glyhb"),xlab = "1/glyhb")

hist(square.glyhb,main = paste("Histogram of sqrt glyhb"),xlab = "sqrt glyhb")

data$glyhb = one.glyhb

### drop NA value

index.na=apply(is.na(data), 1, any)

data.s=data[index.na==FALSE,]

### examine the value

any(is.na(data.s))

data.q = data.s[,-6]

data.q = data.q[,-7]

data.q = data.q[,-9]

pairs(data.q)

cor(data.q)

par(mfrow = c(1,2))

boxplot(data.s$glyhb~data.s$gender,main="Gender")

boxplot(data.s$glyhb~data.s$frame,main="Frame")

set.seed(10) ## set seed for random number generator

##so everyone gets the same split of the data.

n.s=nrow(data.s)

index.s=sample(1: n.s, size=366/2, replace=FALSE)

data.c=data.s[index.s,]

data.v=data.s[-index.s,]

par(mfrow = c(3,2))

boxplot(data.c$glyhb,main = "glyhb Training set")

boxplot(data.c$stab.glu,main = "stab.glu Training set")

boxplot(data.c$ratio,main = "ratio Training set")

boxplot(data.c$age,main = "age Training set")

boxplot(data.c$bp.1s,main = "bp.1s Training set")

boxplot(data.c$waist,main = "waist Training set")

par(mfrow = c(3,2))

boxplot(data.v$glyhb,main = "glyhb Validation set")

boxplot(data.v$stab.glu,main = "stab.glu Validation set")

boxplot(data.v$ratio,main = "ratio Validation set")

boxplot(data.v$age,main = "age Validation set")

boxplot(data.v$bp.1s,main = "bp.1s Validation set")

boxplot(data.v$waist,main = "waist Validation set")

indicatorLocationB = rep(0,366)

indicatorLocationB[which(data.s$location=="Buckingham")] = 1

data.s$location = indicatorLocationB

indicatorGenderM = rep(0,366)

indicatorGenderM[which(data.s$gender=="male")] = 1

data.s$gender = indicatorGenderM

indicatorFL = rep(0,366)

indicatorFL[which(data.s$frame== "large")] = 1

indicatorFM = rep(0,366)

indicatorFM[which(data.s$frame== "medium")] = 1

data.s$large = indicatorFL

data.s$medium = indicatorFM

data.s = data.s[,-11]

n.s=nrow(data.s)

index.s=sample(1: n.s, size=366/2, replace=FALSE)

data.c=data.s[index.s,]

data.v=data.s[-index.s,]

model1 = lm(glyhb~.,data=data.c)

summary(model1)

library(leaps)

a = regsubsets(glyhb~.,data = data.c,method = "exhaustive",nvmax = 16,nbest = 1)

sum\_sub = summary(a)

p.m = as.integer(rownames(sum\_sub$which))+1

ssto = sum((data.c$glyhb - mean(data.c$glyhb))^2)

sse = (1 - sum\_sub$rsq)\*ssto

n = 183

aic = n \* log(sse/n) + 2\*p.m

bic = n \* log(sse/n) + log(n) \* p.m

res\_sub = cbind(sum\_sub$which,sse,sum\_sub$rsq,

sum\_sub$adjr2,sum\_sub$cp,bic,aic)

colnames(res\_sub) = c(colnames(sum\_sub$which),"sse", "R^2", "R^2\_a", "Cp","bic", "aic")

library(MASS)

fit.0 = lm(glyhb~1,data = data.c)

step.0 = stepAIC(fit.0,scope = list(upper=~chol+stab.glu+hdl+ratio+location+

age+gender+height+weight+bp.1s+bp.1d+waist+hip

+time.ppn+large+medium, lower=~1), direction="both", k=2)

step.0$anova

summary(step.0)

par(mfrow = c(1,2))

plot(step.0,which = 1)

plot(step.0,which = 2)

data.variables = data.c[,-5]

data.x = data.c

vNumber = dim(data.variables)[2]

for (i in c(1:(vNumber))){

for (j in c((i):vNumber)){

interactionT = data.variables[i] \* data.variables[j]

s = as.character(k)

colnames(interactionT) = paste("x\_",paste(i,j,sep=","),sep="")

data.x = cbind(data.x,interactionT)

}

}

model2 = lm(glyhb~.,data = data.x)

summary(model2)

library(MASS)

fit.2 = lm(glyhb~1,data = data.x)

step.2 = stepAIC(fit.2,scope = list(upper =~ stab.glu + location + gender + height + weight + bp.1s + waist + time.ppn + large + medium + `x\_1,2` + `x\_1,3` + `x\_1,4` + `x\_1,6` + `x\_1,7` + `x\_1,8` + `x\_1,9` + `x\_1,11` + `x\_1,12` + `x\_1,13` + `x\_1,14` + `x\_1,15` + `x\_2,2` + `x\_2,3` + `x\_2,4` + `x\_2,5` + `x\_2,7` + `x\_2,8` + `x\_2,9` + `x\_2,10` + `x\_2,11` + `x\_2,12` + `x\_2,13` + `x\_2,15` + `x\_2,16` + `x\_3,3` + `x\_3,5` + `x\_3,6` + `x\_3,7` + `x\_3,8` + `x\_3,9` + `x\_3,11` + `x\_3,12` + `x\_3,13` + `x\_3,14` + `x\_3,15` + `x\_4,4` + `x\_4,5` + `x\_4,6` + `x\_4,7` + `x\_4,8` + `x\_4,9` + `x\_4,11` + `x\_4,12` + `x\_4,13` + `x\_4,15` + `x\_4,16` + `x\_5,6` + `x\_5,8` + `x\_5,9` + `x\_5,15` + `x\_6,6` + `x\_6,8` + `x\_6,10` + `x\_6,11` + `x\_6,12` + `x\_6,13` + `x\_6,16` + `x\_7,9` + `x\_7,11` + `x\_7,12` + `x\_7,13` + `x\_7,14` + `x\_7,15` + `x\_7,16` + `x\_8,8` + `x\_8,9` + `x\_8,10` + `x\_8,11` + `x\_8,12` + `x\_8,13` + `x\_8,15` + `x\_8,16` + `x\_9,10` + `x\_9,11` + `x\_9,13` + `x\_9,14` + `x\_9,15` + `x\_9,16` + `x\_10,10` + `x\_10,11` + `x\_10,12` + `x\_10,13` + `x\_10,14` + `x\_11,12` + `x\_11,13` + `x\_11,14` + `x\_11,16` + `x\_12,12` + `x\_12,14` + `x\_12,15` + `x\_12,16` + `x\_13,13` + `x\_13,14` + `x\_13,15` + `x\_14,14` + `x\_14,15` + `x\_14,16` , lower = ~1),direction = "both",steps = 5000,k=2)

### this can get the formular

step.2$anova

par(mfrow = c(1,2))

plot(step.2,which = 1)

plot(step.2,which = 2)

step.2 = stepAIC(fit.2,scope = list(upper =~ stab.glu + location + gender + height + weight + bp.1s + waist + time.ppn + large + medium + `x\_1,2` + `x\_1,3` + `x\_1,4` + `x\_1,6` + `x\_1,7` + `x\_1,8` + `x\_1,9` + `x\_1,11` + `x\_1,12` + `x\_1,13` + `x\_1,14` + `x\_1,15` + `x\_2,2` + `x\_2,3` + `x\_2,4` + `x\_2,5` + `x\_2,7` + `x\_2,8` + `x\_2,9` + `x\_2,10` + `x\_2,11` + `x\_2,12` + `x\_2,13` + `x\_2,15` + `x\_2,16` + `x\_3,3` + `x\_3,5` + `x\_3,6` + `x\_3,7` + `x\_3,8` + `x\_3,9` + `x\_3,11` + `x\_3,12` + `x\_3,13` + `x\_3,14` + `x\_3,15` + `x\_4,4` + `x\_4,5` + `x\_4,6` + `x\_4,7` + `x\_4,8` + `x\_4,9` + `x\_4,11` + `x\_4,12` + `x\_4,13` + `x\_4,15` + `x\_4,16` + `x\_5,6` + `x\_5,8` + `x\_5,9` + `x\_5,15` + `x\_6,6` + `x\_6,8` + `x\_6,10` + `x\_6,11` + `x\_6,12` + `x\_6,13` + `x\_6,16` + `x\_7,9` + `x\_7,11` + `x\_7,12` + `x\_7,13` + `x\_7,14` + `x\_7,15` + `x\_7,16` + `x\_8,8` + `x\_8,9` + `x\_8,10` + `x\_8,11` + `x\_8,12` + `x\_8,13` + `x\_8,15` + `x\_8,16` + `x\_9,10` + `x\_9,11` + `x\_9,13` + `x\_9,14` + `x\_9,15` + `x\_9,16` + `x\_10,10` + `x\_10,11` + `x\_10,12` + `x\_10,13` + `x\_10,14` + `x\_11,12` + `x\_11,13` + `x\_11,14` + `x\_11,16` + `x\_12,12` + `x\_12,14` + `x\_12,15` + `x\_12,16` + `x\_13,13` + `x\_13,14` + `x\_13,15` + `x\_14,14` + `x\_14,15` + `x\_14,16` , lower = ~1),direction = "forward",steps = 5000,k=2)