---

title: “Machine_2”

Question 1

Which of the following commands will create training and test sets with about 50% of the observations assigned to each?

```
library(AppliedPredictiveModeling)
library(caret)
data(AlzheimerDisease)

adData <- data.frame(diagnosis,predictors)
trainIndex <- createDataPartition(adData$diagnosis,p=0.5,list = FALSE)

training <- adData[ trainIndex,]
testing <- adData[ -trainIndex,]

print(rbind(dim(training),dim(testing)))
```

```
##      [,1] [,2]
## [1,]  167  131
## [2,]  166  131
```

Each dataset has the same dimensions

Question 2

Make a plot of the outcome (CompressiveStrength) versus the index of the samples. Color by each of the variables in the data set (you may find the cut2() function in the Hmisc package useful for turning continuous covariates into factors). What do you notice in these plots?

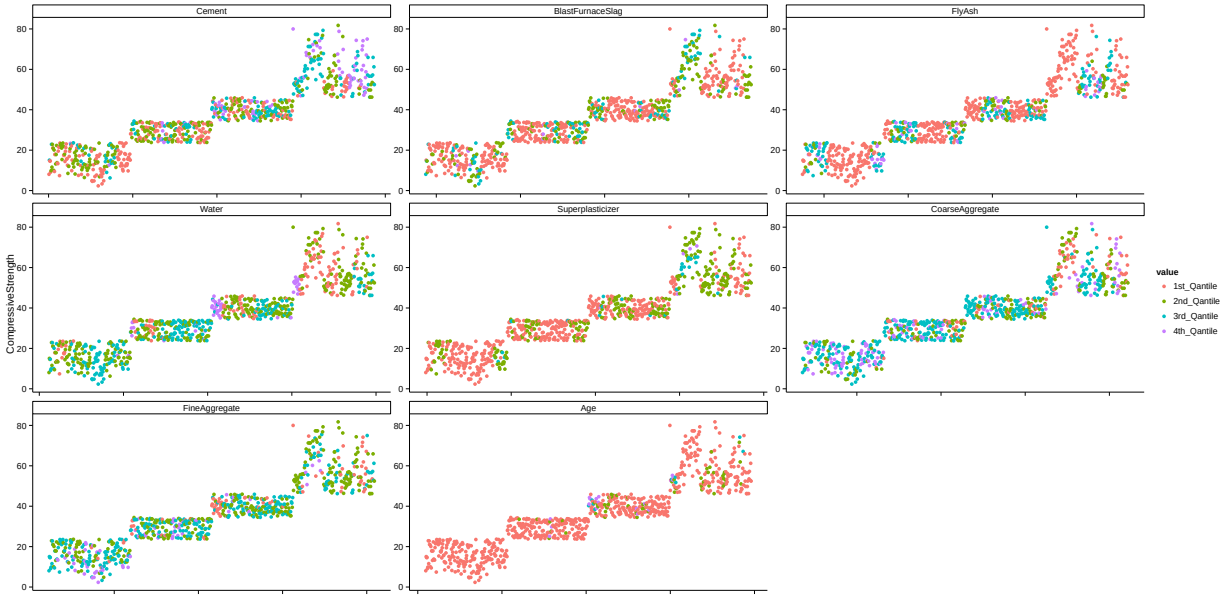
```
library(AppliedPredictiveModeling)
data(concrete)
library(caret)
set.seed(975)
inTrain = createDataPartition(mixtures$CompressiveStrength, p = 3/4)[[1]]
training = mixtures[ inTrain,]
testing = mixtures[-inTrain,]

# Add qunatiles as Factor variables
train_Factors <- as.data.frame(sapply(training,function(x) {
      b <- cut(x,breaks = 4)
      levels(b)<- c("1st_Qantile","2nd_Qantile","3rd_Qantile","4th_Qantile")
      b
      }))

TrainingNew <- cbind(CompressiveStrength=training[,9],train_Factors[,-9])
```

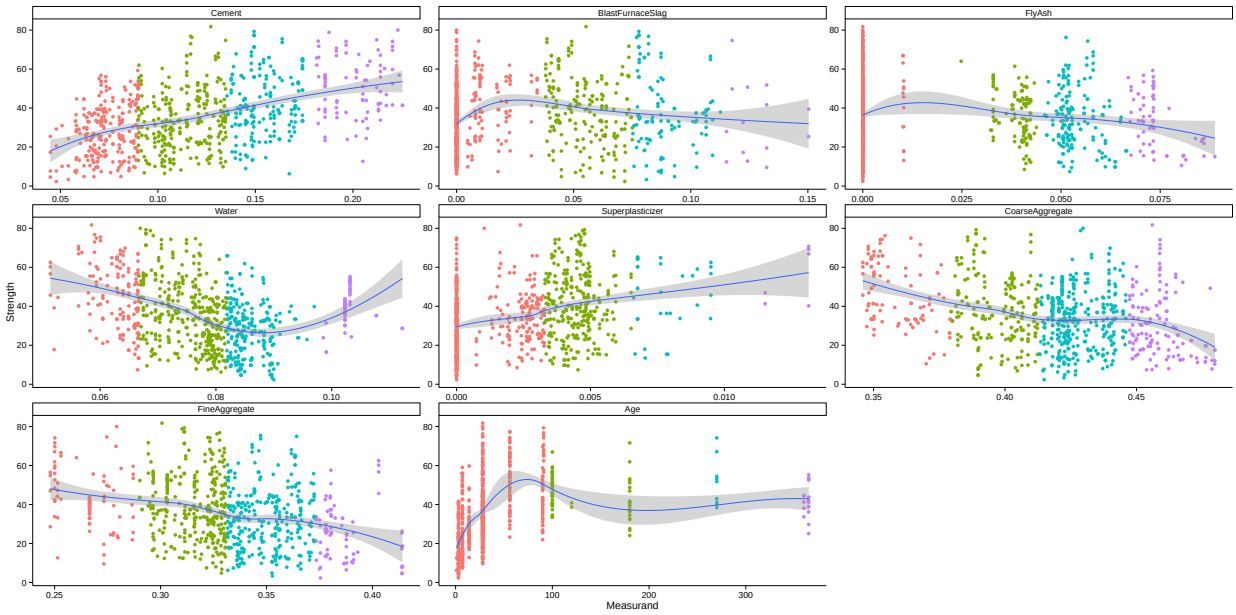
```
library(reshape2)
tidyFactor <- melt(TrainingNew,id.vars = "CompressiveStrength")

#Plot of data as collected
ggplot(tidyFactor,aes(y=CompressiveStrength,x=seq_along(CompressiveStrength)))+
      geom_point(aes(colour=value),alpha=1)+
      xlab("")+
      theme_classic()+
      facet_wrap(~variable,scales = "free")+
      theme(axis.text.x = element_blank())
```




```
#Create a data frame for ggplot
tidy <- melt(training,id.vars = "CompressiveStrength")
tidy <- cbind(tidyFactor,tidy$value)
colnames(tidy)<-c("Strength","Component","Quantile","Measurand")
tidy <-tidy[order(tidy$Strength),]

ggplot(tidy,aes(y=Strength,x=Measurand))+
      geom_point(aes(colour=Quantile))+
      facet_wrap(~Component,scales = "free")+
      geom_smooth(method="loess")+
      theme_classic()+
      theme(legend.position="none")
```



Step like pattern in data, probably a missing variable such as time collected as the change in strength is clearly evident by the relationship with the variables in the data. There is correlation when the variable are sorted by size but this correlation is not evident in the original plot.

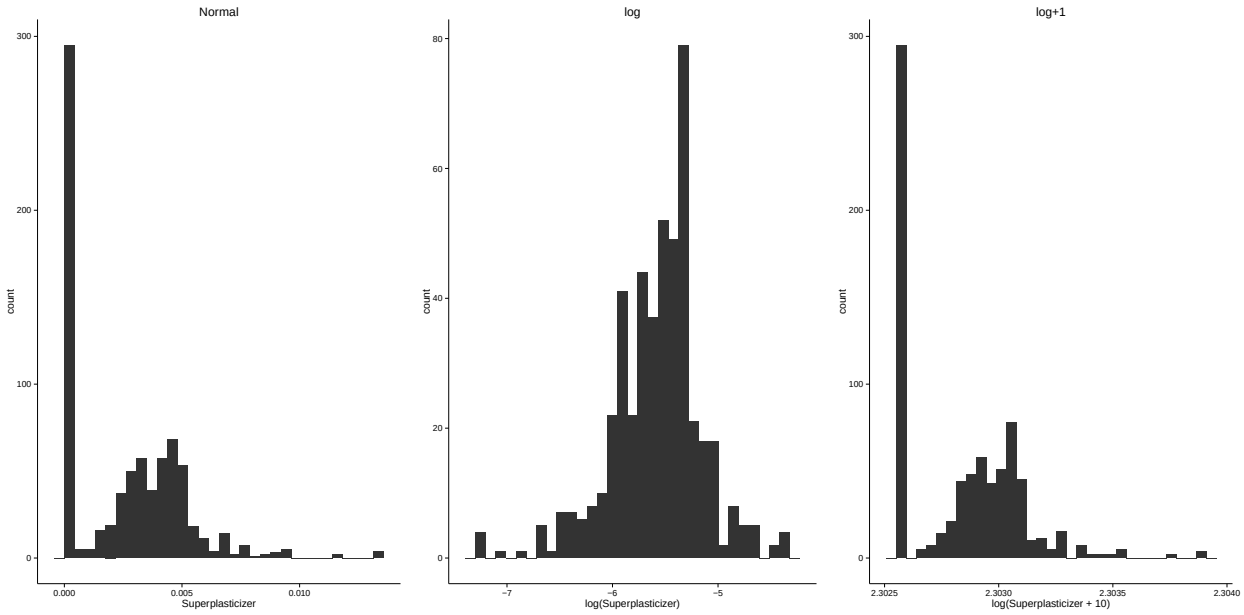
Question 3

Make a histogram and confirm the SuperPlasticizer variable is skewed. Normally you might use the log transform to try to make the data more symmetric. Why would that be a poor choice for this variable?

```
library(AppliedPredictiveModeling)
data(concrete)
library(caret)
set.seed(975)
inTrain = createDataPartition(mixtures$CompressiveStrength, p = 3/4)[[1]]
training = mixtures[ inTrain,]
testing = mixtures[-inTrain,]

p1 <- ggplot(training,aes(Superplasticizer))+geom_histogram(stat = "bin")+ggtitle("Normal")+theme_classi
p2 <- ggplot(training,aes(log(Superplasticizer)))+geom_histogram(stat = "bin")+ggtitle("log")+theme_clas
# adding ten so as to get log of zero
p3 <- ggplot(training,aes(log(Superplasticizer+10)))+geom_histogram(stat = "bin")+ggtitle("log+1")+theme

library(gridExtra)
grid.arrange(p1,p2,p3,ncol=3)
```




```
inf <- table(is.infinite(log(training$Superplasticizer)))
names(inf) <-c("NonZero","Zero/infs")
print(inf)
```

```
##   NonZero Zero/infs
##       479       295
```

the log tranform did not remove the skewness.

Question 4

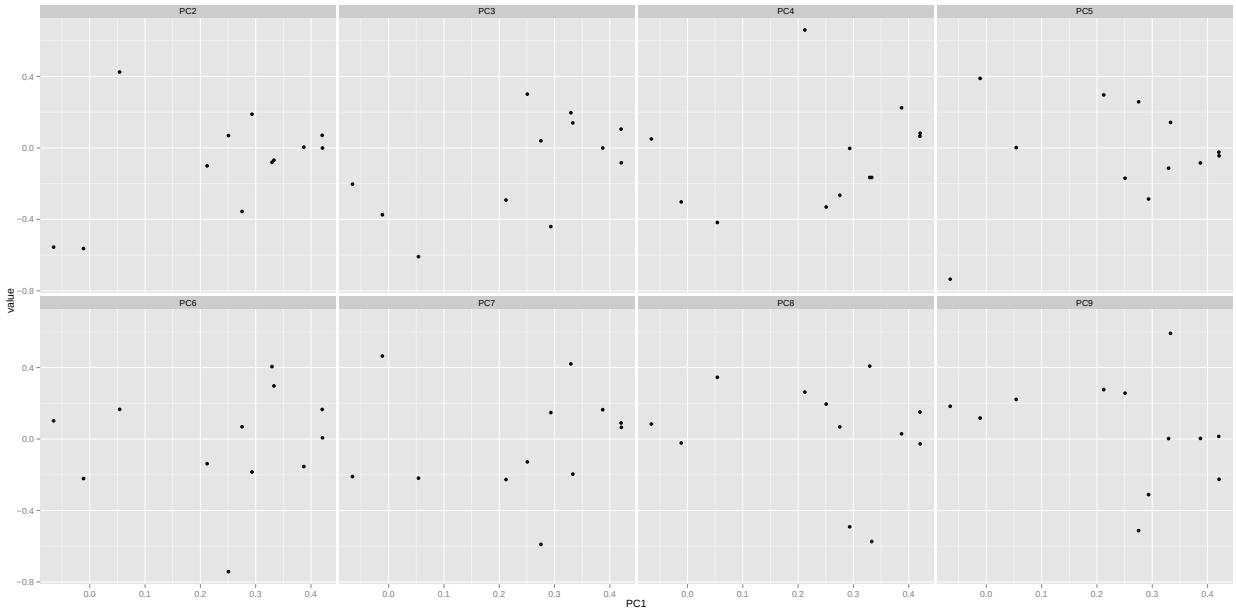
Find all the predictor variables in the training set that begin with IL. Perform principal components on these variables with the preProcess() function from the caret package. Calculate the number of principal components needed to capture 90% of the variance. How many are there?

```
library(caret)
library(AppliedPredictiveModeling)
set.seed(3433)
data(AlzheimerDisease)
adData <- data.frame(diagnosis,predictors)
inTrain <- createDataPartition(adData$diagnosis, p = 3/4)[[1]]
training <- adData[ inTrain,]
testing <- adData[-inTrain,]

library(dplyr)
training1l <- select(training,starts_with("IL"))
trainingPCA <- preProcess(training1l,method = "pca",thresh=.90 )
print(trainingPCA$numComp)
```

```
## [1] 9
```

```
library(reshape2)
plots <- melt(as.data.frame((trainingPCA$rotation)),id = "PC1")
ggplot(plots,aes(y=value,x=PC1))+geom_point()+facet_wrap(~variable,ncol = 4)
```



There are principle components that capture 90% of the variane.

Question 5

Create a training data set consisting of only the predictors with variable names beginning with IL and the diagnosis. Build two predictive models, one using the predictors as they are and one using PCA with principal components explaining 80% of the variance in the predictors. Use method="glm" in the train function. What is the accuracy of each method in the test set? Which is more accurate?

```
library(caret)
library(AppliedPredictiveModeling)
library(dplyr)
set.seed(3433)
data(AlzheimerDisease)
adData <- data.frame(diagnosis,predictors)
inTrain <- createDataPartition(adData$diagnosis, p = 3/4)[[1]]
training <- adData[ inTrain,]
testing <- adData[-inTrain,]

library(dplyr)
training1l <- select(training,diagnosis,starts_with("IL"))

fitglm <- train(diagnosis~.,data = training1l,method = "glm" )
predglm <- predict(fitglm,testing)
resultGLM <- confusionMatrix(predglm,testing$diagnosis)

preProsTrain <- preProcess(training1l[,-1],method = "pca",thresh=.80 )
```

```
trainingPC <- predict(preProsTrain,training1l[,-1])
fitPCA <- train(training$diagnosis~.,data = trainingPC,method = "glm" )
predPCA <- predict(preProsTrain,select(testing,diagnosis,starts_with("IL"))[,-1])
resultPCA <- confusionMatrix(testing$diagnosis,predict(fitPCA,predPCA))

cbind(NonPCA=resultGLM$overall[1],PCA=resultPCA$overall[1])
```

```
##               NonPCA       PCA
## Accuracy 0.6463415 0.7195122
```

The PCA method provides more accuracy

Scatted Matrix plot

```
library(GGally)
ggpairs(training1l,lower = list(continuous = "smooth"),color="diagnosis")
```