> glass <- read.csv(file.choose())

> View(glass)

> summary(glass)

RI Na Mg Al Si

Min. :1.511 Min. :10.73 Min. :0.000 Min. :0.290 Min. :69.81

1st Qu.:1.517 1st Qu.:12.91 1st Qu.:2.115 1st Qu.:1.190 1st Qu.:72.28

Median :1.518 Median :13.30 Median :3.480 Median :1.360 Median :72.79

Mean :1.518 Mean :13.41 Mean :2.685 Mean :1.445 Mean :72.65

3rd Qu.:1.519 3rd Qu.:13.82 3rd Qu.:3.600 3rd Qu.:1.630 3rd Qu.:73.09

Max. :1.534 Max. :17.38 Max. :4.490 Max. :3.500 Max. :75.41

K Ca Ba Fe Type

Min. :0.0000 Min. : 5.430 Min. :0.000 Min. :0.00000 Min. :1.00

1st Qu.:0.1225 1st Qu.: 8.240 1st Qu.:0.000 1st Qu.:0.00000 1st Qu.:1.00

Median :0.5550 Median : 8.600 Median :0.000 Median :0.00000 Median :2.00

Mean :0.4971 Mean : 8.957 Mean :0.175 Mean :0.05701 Mean :2.78

3rd Qu.:0.6100 3rd Qu.: 9.172 3rd Qu.:0.000 3rd Qu.:0.10000 3rd Qu.:3.00

Max. :6.2100 Max. :16.190 Max. :3.150 Max. :0.51000 Max. :7.00

> describe(glass)

vars n mean sd median trimmed mad min max range skew kurtosis se

RI 1 214 1.52 0.00 1.52 1.52 0.00 1.51 1.53 0.02 1.60 4.72 0.00

Na 2 214 13.41 0.82 13.30 13.38 0.64 10.73 17.38 6.65 0.45 2.90 0.06

Mg 3 214 2.68 1.44 3.48 2.87 0.30 0.00 4.49 4.49 -1.14 -0.45 0.10

Al 4 214 1.44 0.50 1.36 1.41 0.31 0.29 3.50 3.21 0.89 1.94 0.03

Si 5 214 72.65 0.77 72.79 72.71 0.57 69.81 75.41 5.60 -0.72 2.82 0.05

K 6 214 0.50 0.65 0.56 0.43 0.17 0.00 6.21 6.21 6.46 52.87 0.04

Ca 7 214 8.96 1.42 8.60 8.74 0.66 5.43 16.19 10.76 2.02 6.41 0.10

Ba 8 214 0.18 0.50 0.00 0.03 0.00 0.00 3.15 3.15 3.37 12.08 0.03

Fe 9 214 0.06 0.10 0.00 0.04 0.00 0.00 0.51 0.51 1.73 2.52 0.01

Type 10 214 2.78 2.10 2.00 2.48 1.48 1.00 7.00 6.00 1.10 -0.33 0.14

> str(glass)

'data.frame': 214 obs. of 10 variables:

$ RI : num 1.52 1.52 1.52 1.52 1.52 ...

$ Na : num 13.6 13.9 13.5 13.2 13.3 ...

$ Mg : num 4.49 3.6 3.55 3.69 3.62 3.61 3.6 3.61 3.58 3.6 ...

$ Al : num 1.1 1.36 1.54 1.29 1.24 1.62 1.14 1.05 1.37 1.36 ...

$ Si : num 71.8 72.7 73 72.6 73.1 ...

$ K : num 0.06 0.48 0.39 0.57 0.55 0.64 0.58 0.57 0.56 0.57 ...

$ Ca : num 8.75 7.83 7.78 8.22 8.07 8.07 8.17 8.24 8.3 8.4 ...

$ Ba : num 0 0 0 0 0 0 0 0 0 0 ...

$ Fe : num 0 0 0 0 0 0.26 0 0 0 0.11 ...

$ Type: int 1 1 1 1 1 1 1 1 1 1 ...

> glass$Type <- factor(glass$Type)

> str(glass)

'data.frame': 214 obs. of 10 variables:

$ RI : num 1.52 1.52 1.52 1.52 1.52 ...

$ Na : num 13.6 13.9 13.5 13.2 13.3 ...

$ Mg : num 4.49 3.6 3.55 3.69 3.62 3.61 3.6 3.61 3.58 3.6 ...

$ Al : num 1.1 1.36 1.54 1.29 1.24 1.62 1.14 1.05 1.37 1.36 ...

$ Si : num 71.8 72.7 73 72.6 73.1 ...

$ K : num 0.06 0.48 0.39 0.57 0.55 0.64 0.58 0.57 0.56 0.57 ...

$ Ca : num 8.75 7.83 7.78 8.22 8.07 8.07 8.17 8.24 8.3 8.4 ...

$ Ba : num 0 0 0 0 0 0 0 0 0 0 ...

$ Fe : num 0 0 0 0 0 0.26 0 0 0 0.11 ...

$ Type: Factor w/ 6 levels "1","2","3","5",..: 1 1 1 1 1 1 1 1 1 1 ...

> attach(glass)

> ggpairs(glass)

plot: [10,1] [===============================================>-----] 91% est: 1s `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

plot: [10,2] [================================================>----] 92% est: 1s `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

plot: [10,3] [================================================>----] 93% est: 1s `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

plot: [10,4] [=================================================>---] 94% est: 0s `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

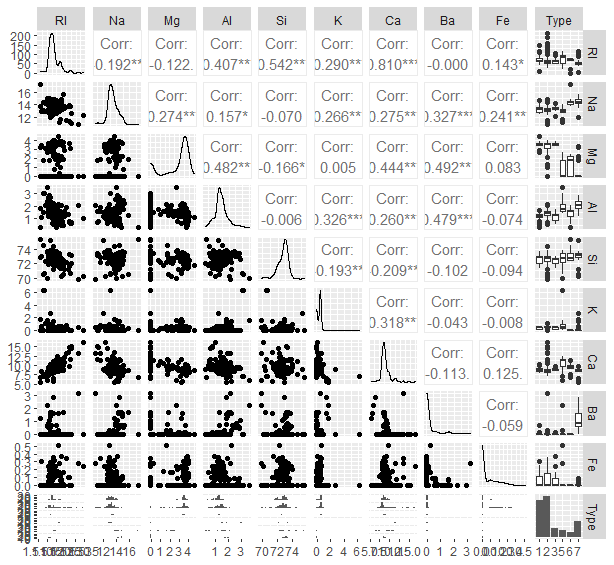
plot: [10,5] [=================================================>---] 95% est: 0s `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

plot: [10,6] [==================================================>--] 96% est: 0s `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

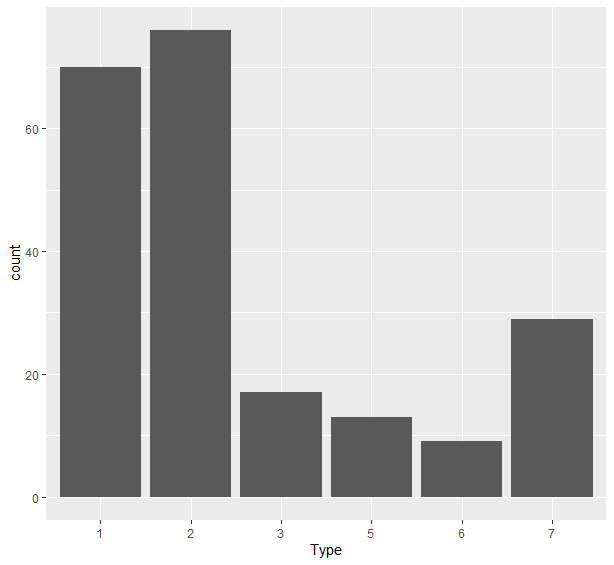
plot: [10,7] [==================================================>--] 97% est: 0s `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

plot: [10,8] [===================================================>-] 98% est: 0s `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

plot: [10,9] [===================================================>-] 99% est: 0s `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



> ggplot(glass)+geom\_bar(mapping = aes(x=Type))



> table(Type)

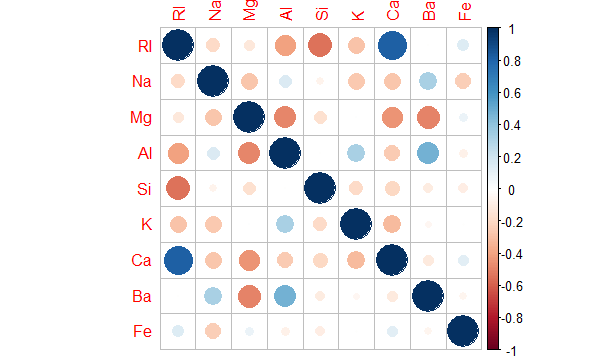
Type

1 2 3 5 6 7

70 76 17 13 9 29

> cor <- cor(glass[,-10])

> corrplot(cor,method = "circle")



> norm <- function(x){

+ return((x-min(x))/(max(x)-min(x)))

+ }

> glass1 <- as.data.frame(lapply(glass[,-10],norm))

> View(glass1)

> glass2 <- data.frame(glass1,Type)

> #splitting of data to train and test

> set.seed(100)

> splt <- createDataPartition(Type,p=0.8,list = F)

> trn <- glass2[splt,]

> tst <- glass2[-splt,]

> #bagging method to choose k value

> trn\_acc <- c()

> tst\_acc <- c()

> for(i in seq(1,50,2)){

+ set.seed(100)

+ trn\_model <- knn(trn,trn,cl=trn$Type,k=i)

+ tst\_model <- knn(trn,tst,cl=trn$Type,k=i)

+ trn\_acc <- c(trn\_acc,mean(trn\_model==trn$Type))

+ tst\_acc <- c(tst\_acc,mean(tst\_model==tst$Type))

+ }

> trn\_acc

[1] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000

[9] 1.0000000 0.9827586 0.9712644 0.9540230 0.9540230 0.9482759 0.8735632 0.8333333

[17] 0.8218391 0.8218391 0.8218391 0.8160920 0.8160920 0.8103448 0.8103448 0.8103448

[25] 0.8103448

> tst\_acc

[1] 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 0.975 0.975

[14] 0.975 0.900 0.850 0.850 0.850 0.850 0.850 0.850 0.850 0.850 0.850 0.850

> acc <- data.frame(list(train=trn\_acc,test=tst\_acc,k=seq(1,50,2)))

> acc

train test k

1 1.0000000 1.000 1

2 1.0000000 1.000 3

3 1.0000000 1.000 5

4 1.0000000 1.000 7

5 1.0000000 1.000 9

6 1.0000000 1.000 11

7 1.0000000 1.000 13

8 1.0000000 1.000 15

9 1.0000000 1.000 17

10 0.9827586 1.000 19

11 0.9712644 1.000 21

12 0.9540230 0.975 23

13 0.9540230 0.975 25

14 0.9482759 0.975 27

15 0.8735632 0.900 29

16 0.8333333 0.850 31

17 0.8218391 0.850 33

18 0.8218391 0.850 35

19 0.8218391 0.850 37

20 0.8160920 0.850 39

21 0.8160920 0.850 41

22 0.8103448 0.850 43

23 0.8103448 0.850 45

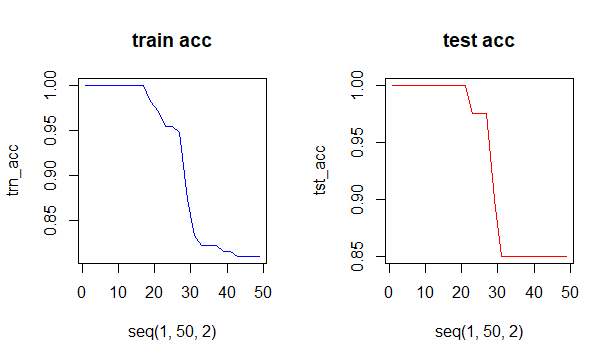
24 0.8103448 0.850 47

25 0.8103448 0.850 49

> par(mfrow=c(1,2))

> plot(seq(1,50,2),trn\_acc,type = 'l',main = "train acc",col='blue')

> plot(seq(1,50,2),tst\_acc,type = "l",main = "test acc",col='red')

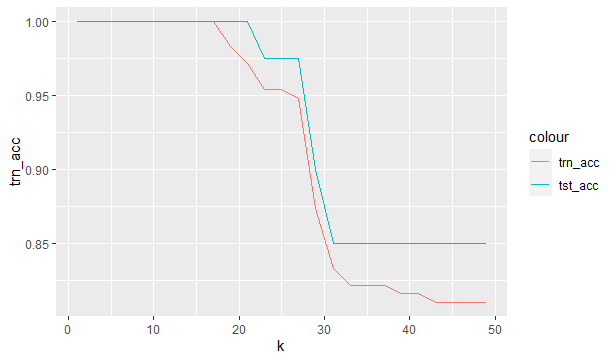


> #visualisation of bagging method

> ggplot(acc,aes(x=k))+

+ geom\_line(aes(y=trn\_acc,colour="trn\_acc"))+

+ geom\_line(aes(y=tst\_acc,colour="tst\_acc"))



> model <- knn(trn,tst,k=21,cl=trn$Type)

> summary(model)

1 2 3 5 6 7

14 15 3 2 1 5

> pred <- mean(model==tst$Type)

> pred

[1] 1

> confusionMatrix(model,tst$Type)

Confusion Matrix and Statistics

Reference

Prediction 1 2 3 5 6 7

1 14 0 0 0 0 0

2 0 15 0 0 0 0

3 0 0 3 0 0 0

5 0 0 0 2 0 0

6 0 0 0 0 1 0

7 0 0 0 0 0 5

Overall Statistics

Accuracy : 1

95% CI : (0.9119, 1)

No Information Rate : 0.375

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 1

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: 1 Class: 2 Class: 3 Class: 5 Class: 6 Class: 7

Sensitivity 1.00 1.000 1.000 1.00 1.000 1.000

Specificity 1.00 1.000 1.000 1.00 1.000 1.000

Pos Pred Value 1.00 1.000 1.000 1.00 1.000 1.000

Neg Pred Value 1.00 1.000 1.000 1.00 1.000 1.000

Prevalence 0.35 0.375 0.075 0.05 0.025 0.125

Detection Rate 0.35 0.375 0.075 0.05 0.025 0.125

Detection Prevalence 0.35 0.375 0.075 0.05 0.025 0.125

Balanced Accuracy 1.00 1.000 1.000 1.00 1.000 1.000