Bankruptcy data

```
bank<- read.csv(file.choose())
sum(is.na(bank))
bank$Bankrupt.[bank$Bankrupt.== 1] <- 2
bank$Bankrupt.[bank$Bankrupt.== 0] <- 1
library(dplyr)
set.seed(100)
bankrupy<- sample_frac(bank, 0.10)

# remove category as input data
bankrupy_nocateg <-bankrupy[-45]
bankrupy_ <-bankrupy[45]
```

Read data 取 10%，並將 0/1 改為 1/2

```
library(cluster)
fanny_bank <- fanny(bankrupy_nocateg, 2, metric="SqEuclidean")
# summary(fanny_diamonds)
fanny_bank$clustering
fanny_bank$membership
fanny_bankrupy_membership <- fanny_bank$membership
new_bankrupy <- cbind(fanny_bankrupy_membership,bankrupy_)
```

重新建 fuzzy model ,分兩組

```
bank_split <- split(new_bankrupy,new_bankrupy$Bankrupt.)
bank_1 <- bank_split$"1"
bank_2 <- bank_split$"2"
set.seed(150)
bank_1_sample <- bank_1[sample(nrow(bank_1)), ] # shuffle
bank_2_sample <- bank_2[sample(nrow(bank_2)), ] # shuffle
bank_1_percent <- floor(nrow(bank_1_sample)*0.7)
bank_2_percent <- floor(nrow(bank_2_sample)*0.7)
bank_1_sample_70 <- bank_1_sample[1:bank_1_percent, ]
bank_1_sample_30 <- bank_1_sample[(bank_1_percent+1):nrow(bank_1_sample), ]
bank_2_sample_70 <- bank_2_sample[1:bank_2_percent, ]
bank_2_sample_30 <- bank_2_sample[(bank_2_percent+1):nrow(bank_2_sample), ]
bank_train <- rbind(bank_1_sample_70, bank_2_sample_70)
bank_test_all <- rbind(bank_1_sample_30, bank_2_sample_30)
bank_ans <- bank_test_all[3]###
bank_test <- bank_test_all[-3]
```

區分 train & test data

```
bank_range <-apply(new_bankrupy[ , -3], 2, range)
bank_method <- "FRBCS.W"
bank_control <- list(num.labels = 2,
                     type.mf = "TRIANGLE", type.defuz = "WAM",
                     type.tnorm = "MIN", type.snorm = "MAX",
                     type.implication.func = "ZADEH")
library(frbs)
start.time <- Sys.time()

object.reg <- frbs.learn(bank_train,bank_range,bank_method,bank_control)

end.time <- Sys.time()
time.taken <- end.time - start.time
time.taken
```

建 classification model 的模糊系統

```
> time.taken
Time difference of 1.466311 secs
```

```
bank_predict <- predict(object.reg, bank_test)
bank_compare <- cbind(bank_predict, bank_ans)
colnames(bank_compare) <- c("predict", "real")
confusion <- table(bank_compare$predict, bank_compare$real)
library(caret)
confusionMatrix(confusion, positive = "2")
```

最後 predict data

```
> confusionMatrix(confusion, positive = "2")
Confusion Matrix and Statistics


     1   2
  1 198   7
  2   1   0

               Accuracy : 0.9612
                 95% CI : (0.9249, 0.9831)
    No Information Rate : 0.966
    P-Value [Acc > NIR] : 0.7314

                  Kappa : -0.0086

 Mcnemar's Test P-Value : 0.0771

            Sensitivity : 0.000000
            Specificity : 0.994975
         Pos Pred Value : 0.000000
         Neg Pred Value : 0.965854
             Prevalence : 0.033981
         Detection Rate : 0.000000
   Detection Prevalence : 0.004854
      Balanced Accuracy : 0.497487

       'Positive' Class : 2
```

Accuracy =0.96

Diamonds Data

```
# read data
diamonds_raw <- read.csv(file.choose())
sum(is.na(diamonds_raw))

# random sample 10%
library(dplyr)
set.seed(100)
diamonds <- sample_frac(diamonds_raw, 0.10)

# remove category as input data
diamonds_nocateg <- diamonds[-7]
diamonds_price <- diamonds[7]
```

```
library(cluster)
fanny_diamonds <- fanny(diamonds_nocateg, 2, metric="SqEuclidean")
# summary(fanny_diamonds)
fanny_diamonds$clustering
fanny_diamonds$membership
```

以上為 fuzzy 建模

```
fanny_diamonds_membership <- fanny_diamonds$membership
new_diamond <- cbind(fanny_diamonds_membership, diamonds_price)

##############################################################################
##############################################################################
#  set.seed(seed, kind = NULL, normal.kind = NULL, sample.kind = NULL)
percentage <- floor(nrow(diamonds)*0.75)
set.seed(100)
diamonds_sample <- new_diamond[sample(nrow(diamonds)), ] # shuffle
diamonds_train <- diamonds_sample[1:percentage, ]
diamonds_test_all <- diamonds_sample[(percentage+1):nrow(diamonds), ]
ans <- diamonds_test_all[3]
diamonds_test <- diamonds_test_all[-3]
```

以上將資料分為 train 和 test data

```
## Define interval of data
range.data <-apply(diamonds_train[ , 1:3], 2, range)

## Set the method and its parameters,
method.type <- "ANFIS"
control <- list(num.labels = 2, max.iter = 100, step.size = 0.01,
                type.tnorm = "MIN", type.snorm = "MAX",
                type.implication.func = "ZADEH",
                name = "diamonds")
## Learning step: Generate an FRBS model
library(frbs)
start.time <- Sys.time()

object.reg <- frbs.learn(diamonds_train, range.data, method.type, control)

end.time <- Sys.time()
time.taken <- end.time - start.time
time.taken
```

Anfis 的建模方式(method type & control) 參考 frbs R 套件

```
> time.taken
Time difference of 5.374535 mins
```

```
#################################
diamonds_residuals <- (ans - diamonds_predict)
diamonds_residuals_sqr <- diamonds_residuals$price^2
diamonds_MSE <- mean(diamonds_residuals_sqr)
diamonds_RMSE <- sqrt(mean(diamonds_residuals_sqr))
diamonds_NRMSE <- sqrt(mean(diamonds_residuals$price)^2)/sd(diamonds_predict)
diamonds_MAPE<-mean(abs((diamonds_residuals$price)/ans$price))*100
diamonds_error <- c(diamonds_MSE,diamonds_RMSE, diamonds_NRMSE, diamonds_MAPE)
names(diamonds_error) <- c("MSE","RMSE", "NRMSE", "MAPE")
print(diamonds_error)
```

```
> print(diamonds_error)
         MSE          RMSE         NRMSE          MAPE
3.225111e+07 5.679006e+03 3.622745e+01 7.690408e+01
>
```

在均方誤差的部分非常大，其實代表預測的結果不是太好，可能的原因是 fuzzy 的模型建的不夠好。