

Framingham - Logistic Regression

Data Analysis - EX 3

Babak Barghi, Han Jia

November 30, 2020

Contents

1	Introduction	3
2	Data description	3
3	Framework of analysis:	4
3.1	Task 1: Split the data set in two sets (training and testing) taking into account which is the outcome of the data set.	4
3.1.1	Split the data set	6
3.2	Task 2: Analyse the correlation between independent variables and choose the set of variables for a prediction model.	8
3.3	Taks 4: Visualise the probabilities produced by the model.	13
3.4	Task 5: Choose a threshold using the ROC.	13
3.5	Task 6: Evaluate its specificity and sensitivity and the AUC.	14
3.6	Task 7: Use the model and assess its predictions.	15
4	Additional Task	16
4.1	Change the proportion of observations between the training and testing set and repeat the task. Compare the results on the testing set.	16
5	Refrences	16

The analysis in this report is carried out in *R* 4.0.2[1] and the libraries below has been used.

```
library(tidyverse)
library(kableExtra)
library(mice)
library(caTools)
library(corrplot)
library(RColorBrewer)
library(ROCR)
```

1 Introduction

The Framingham Heart Study is one of the most important epidemiological studies ever conducted, and the underlying analytics that led to our current understanding of cardiovascular disease. In the late 1940s, the US government set out to better understand cardiovascular disease. The plan was to track a large cohort of initially healthy patients over their lifetimes. The study included 5,209 patients, aged 30 to 59. Patients were given a questionnaire and an examination every two years. During this examination, their physical characteristics were recorded, their behavioural characteristics, as well as test results. Exams and questions expanded over time, but the key in the study was that the trajectory of the health of the patients was followed during their entire lifespan. In the following report the analysis and predictions are carried by Logistic regression which is the appropriate regression analysis to conduct when the dependent variable is binary. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

2 Data description

From the provided information it is stated that there are 4240 patients and 16 variables in *Framingham* dataset. In this report, the factors are categorized as follow for better understanding of analysis;

First group is related to the demographic risk factors:

- **male**
- **age**
- **education**

The behavioral risk factors:

- **currentSmoker**
- **cigsPerDay**

The medical history risk factors:

- **BPMeds**
- **prevalentStroke**
- **prevalentHyp**
- **diabetes**

Also physical examination risk factors:

- The total cholesterol level (totChol)
- systolic blood pressure (sysBP)
- diastolic blood pressure (diaBP)
- Body Mass Index (BMI)
- heartRate
- glucose level

The last variable is the outcome or dependent variable, whether or not the patient developed coronary heart disease **CHD** in the next 10 years.

Before getting into the analysis and specific tasks the data set would be imported.

```
framinghamdata <- read_csv("framingham.csv")
```

The Data frame is read by using *read.csv* function.

3 Framework of analysis:

1. Split the data set in two sets (training and testing) taking into account which is the outcome of the data set.
2. Analyse the correlation between independent variables and choose the set of variables for a prediction model.
3. Build the logistic regression model.
4. Visualise the probabilities produced by the model.
5. Choose a threshold using the ROC.
6. Evaluate its specificity and sensitivity and the AUC.
7. Use the model and assess its predictions.

Additional Tasks:

Change the proportion of observations between the training and testing set and repeat the task. Compare the results on the testing set.

3.1 Task 1: Split the data set in two sets (training and testing) taking into account which is the outcome of the data set.

First a general glance at the dataset would be beneficiary using the *summary* function.

```
summary(framinghamdata)
```

```
##      male      age      education      currentSmoker
## Min.   :0.0000  Min.   :32.00  Min.    :1.000  Min.    :0.0000
## 1st Qu.:0.0000  1st Qu.:42.00  1st Qu.:1.000  1st Qu.:0.0000
## Median :0.0000  Median :49.00  Median :2.000  Median :0.0000
## Mean   :0.4292  Mean   :49.58  Mean    :1.979  Mean    :0.4941
## 3rd Qu.:1.0000  3rd Qu.:56.00  3rd Qu.:3.000  3rd Qu.:1.0000
## Max.   :1.0000  Max.    :70.00  Max.    :4.000  Max.    :1.0000
##
##      NA's      :105
##      cigsPerDay      BPMeds      prevalentStroke      prevalentHyp
## Min.    : 0.000  Min.    :0.000000  Min.    :0.000000  Min.    :0.0000
## 1st Qu.: 0.000  1st Qu.:0.000000  1st Qu.:0.000000  1st Qu.:0.0000
```

```
## Median : 0.000 Median :0.00000 Median :0.000000 Median :0.0000
## Mean : 9.006 Mean :0.02962 Mean :0.005896 Mean :0.3106
## 3rd Qu.:20.000 3rd Qu.:0.00000 3rd Qu.:0.000000 3rd Qu.:1.0000
## Max. :70.000 Max. :1.00000 Max. :1.000000 Max. :1.0000
## NA's :29 NA's :53
## diabetes totChol sysBP diaBP
## Min. :0.00000 Min. :107.0 Min. : 83.5 Min. : 48.0
## 1st Qu.:0.00000 1st Qu.:206.0 1st Qu.:117.0 1st Qu.: 75.0
## Median :0.00000 Median :234.0 Median :128.0 Median : 82.0
## Mean :0.02571 Mean :236.7 Mean :132.4 Mean : 82.9
## 3rd Qu.:0.00000 3rd Qu.:263.0 3rd Qu.:144.0 3rd Qu.: 90.0
## Max. :1.00000 Max. :696.0 Max. :295.0 Max. :142.5
## NA's :50
## BMI heartRate glucose TenYearCHD
## Min. :15.54 Min. : 44.00 Min. : 40.00 Min. :0.0000
## 1st Qu.:23.07 1st Qu.: 68.00 1st Qu.: 71.00 1st Qu.:0.0000
## Median :25.40 Median : 75.00 Median : 78.00 Median :0.0000
## Mean :25.80 Mean : 75.88 Mean : 81.96 Mean :0.1519
## 3rd Qu.:28.04 3rd Qu.: 83.00 3rd Qu.: 87.00 3rd Qu.:0.0000
## Max. :56.80 Max. :143.00 Max. :394.00 Max. :1.0000
## NA's :19 NA's :1 NA's :388
```

There are many NA values in the data fram, so we would have a closer look.

```
sum(is.na(framinghamdata))
```

#number of missing values

```
## [1] 645
```

```
colnames(framinghamdata)[colSums(is.na(framinghamdata)) > 0] #which columns have missing values
```

```
## [1] "education" "cigsPerDay" "BPMeds" "totChol" "BMI"
## [6] "heartRate" "glucose"
```

Since, there are 645 NA values in the dataframe for 7 variables, these NA numbers can be imputed using the *mice* function.

```
data <- framinghamdata
data <- mice(data, seed = 42)
```

```
##
## iter imp variable
## 1 1 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 1 2 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 1 3 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 1 4 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 1 5 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 2 1 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 2 2 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 2 3 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 2 4 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 2 5 education cigsPerDay BPMeds totChol BMI heartRate glucose
```

```
## 3 1 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 3 2 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 3 3 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 3 4 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 3 5 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 4 1 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 4 2 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 4 3 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 4 4 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 4 5 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 5 1 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 5 2 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 5 3 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 5 4 education cigsPerDay BPMeds totChol BMI heartRate glucose
## 5 5 education cigsPerDay BPMeds totChol BMI heartRate glucose
```

```
data <- mice::complete(data)
```

After imputing the baseline accuracy will be built for the *TenYearCHD* binary variable.

```
table(data$TenYearCHD)
```

```
##
##      0      1
## 3596    644
```

```
baseline_accuracy <- (table(data$TenYearCHD)[1]/(table(data$TenYearCHD)[1] +
                                                    table(data$TenYearCHD)[2]))*100
```

```
baseline_accuracy <- baseline_accuracy %>% round(digits = 2)
print(baseline_accuracy)
```

```
##      0
## 84.81
```

Considering the variable *CHD*, it can be notice that 3596 out of the 4240 patients did not develop coronary heart disease, or 0, and 644 patients who developed coronary heart disease, labeled with 1.

Hence the baseline model accuracy is 84.81 percent, where baseline model accuracy is defined as the most frequent outcome of the dependent variable *CHD*. The model in the report, that will be built, has the goal of improve this baseline accuracy.

3.1.1 Split the data set

It is necessary to randomly split our data set into a training set and testing set. Testing set is essential to validate results. For splitting the data *caTools* package is used also in order to get the same split the *set.seed* function is used, which initializes the random number generator. The *caTools* package contains *sample.split* command to split the data, setting split ratio of 0.75, meaning that 75% of the data will determine the training set, which it is use to build the model, while the other 25% of the data will constitute the testing set.

```
set.seed(42)
split <- sample.split(data$TenYearCHD, SplitRatio = 0.75)
```

The function assign a *TRUE* or *FALSE* value for each observation, where *TRUE* means that the observation will be put in the **training** set while *FALSE* means that the observation will be part of the **testing** set.

In order to create our the two sets the *subset* function is used. Training set will be called *Training* and testing set *Test*.

```
Training <- subset(data, split == TRUE)
Test <- subset(data, split == FALSE)

nrow(Training)
```

```
## [1] 3180
```

```
nrow(Test)
```

```
## [1] 1060
```

In the train data set there are 3180 rows while in the test data set there are 1060 rows.

Separation result is summarized below as number of observation in the training and test set per each CHD value and proportion of 0 values in each set.

```
P_Train <- c(table(Training$TenYearCHD)[1], table(Training$TenYearCHD)[2])
P_Test <- c(table(Test$TenYearCHD)[1], table(Test$TenYearCHD)[2])

CHD_0_Proportion <- c(P_Train[1]/(P_Train[1]+P_Train[2]),
                      P_Test[1]/(P_Test[1]+P_Test[2]))

S_result <- data.frame(P_Train, P_Test) %>% as.data.frame()

S_result[3,] <- CHD_0_Proportion

S_result %>%
  kable(align= c('c'), digits = 2) %>%
  kable_styling(full_width = F, position = "center") %>%
  row_spec(3, bold = T)
```

	P_Train	P_Test
0	2697.00	899.00
1	483.00	161.00
3	0.85	0.85

Since the proportion of zero values for the two sets is equal, it can be claimed that a good separation has been performed.

3.2 Task 2: Analyse the correlation between independent variables and choose the set of variables for a prediction model.

In order to choose the variables that will be used for creating the model, correlation among variables is checked in the training set. The correlation has to be checked only for the numerical and binary variables, hence, a new dataset by eliminating the categorical variable *education* is created.

```
train_cor <- Training[,-3]
corrplot(abs(cor(Training)>0.7) ,tl.col = "black",
         type="upper", col = brewer.pal(n = 8, name = "Reds"))
```

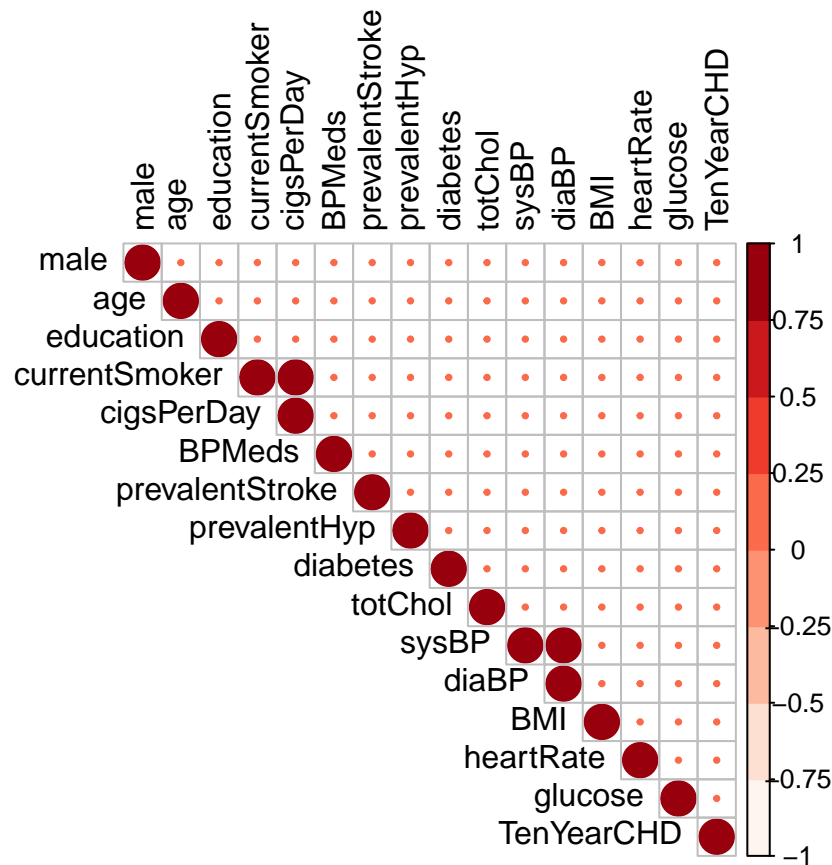


Figure 1: Correlation among variables

Figure 1 highlights that the pairs of variable correlated are:

- *diaBP* and *sysBP*.
- *cigsPerDay* and *currentSmoker*.

Since there are two pairs of variables which are correlated, four different models will be built by changing only the pairs of correlated variables with considering all the other independent variables.

The First model is built by *sysBP*., *cigsPerDay* and the others independent variables.


```

M1_CHD <- glm(TenYearCHD ~ male + age + education + cigsPerDay +
              BPMeds + prevalentStroke + prevalentHyp + diabetes +
              totChol + sysBP + BMI + heartRate + glucose,
              data=Training, family=binomial)

summary(M1_CHD)

##
## Call:
## glm(formula = TenYearCHD ~ male + age + education + cigsPerDay +
##      BPMeds + prevalentStroke + prevalentHyp + diabetes + totChol +
##      sysBP + BMI + heartRate + glucose, family = binomial, data = Training)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9162  -0.5974  -0.4302  -0.2833   2.6319
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -8.504174   0.736835 -11.541 < 2e-16 ***
## male           0.497613   0.115681   4.302 1.70e-05 ***
## age            0.061980   0.007062   8.776 < 2e-16 ***
## education     -0.005865   0.052233  -0.112 0.91059
## cigsPerDay     0.022438   0.004530   4.953 7.31e-07 ***
## BPMeds         0.299493   0.248642   1.205 0.22839
## prevalentStroke 0.777549   0.511777   1.519 0.12868
## prevalentHyp   0.269004   0.144741   1.859 0.06310 .
## diabetes       0.228800   0.339496   0.674 0.50035
## totChol        0.002608   0.001213   2.151 0.03148 *
## sysBP          0.012877   0.003218   4.002 6.28e-05 ***
## BMI            0.007285   0.013344   0.546 0.58513
## heartRate     -0.001469   0.004443  -0.331 0.74088
## glucose        0.006500   0.002449   2.655 0.00794 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 2709.2  on 3179  degrees of freedom
## Residual deviance: 2398.1  on 3166  degrees of freedom
## AIC: 2426.1
##
## Number of Fisher Scoring iterations: 5

```

The second model is built considering *sysBP.*, *currentSmoker.* and the others eleven independent variables.

```

M2_CHD <- glm(TenYearCHD ~ male + age + education + currentSmoker +
              BPMeds + prevalentStroke + prevalentHyp + diabetes +
              totChol + sysBP + BMI + heartRate + glucose,
              data=Training, family=binomial)

summary(M2_CHD)

```

```
##
## Call:
## glm(formula = TenYearCHD ~ male + age + education + currentSmoker +
##      BPMeds + prevalentStroke + prevalentHyp + diabetes + totChol +
##      sysBP + BMI + heartRate + glucose, family = binomial, data = Training)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9315  -0.5924  -0.4351  -0.2920   2.5999
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -8.537308   0.741580 -11.512  < 2e-16 ***
## male           0.594223   0.111357   5.336  9.49e-08 ***
## age            0.059707   0.007007   8.520  < 2e-16 ***
## education     -0.007395   0.052129  -0.142  0.887187
## currentSmoker  0.389983   0.113360   3.440  0.000581 ***
## BPMeds         0.306036   0.248311   1.232  0.217773
## prevalentStroke 0.764286   0.510593   1.497  0.134430
## prevalentHyp   0.271145   0.144429   1.877  0.060468 .
## diabetes       0.212007   0.339393   0.625  0.532192
## totChol        0.002673   0.001210   2.210  0.027122 *
## sysBP          0.012885   0.003211   4.012  6.01e-05 ***
## BMI            0.008059   0.013384   0.602  0.547088
## heartRate      -0.000160   0.004406  -0.036  0.971040
## glucose        0.006442   0.002442   2.638  0.008331 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 2709.2  on 3179  degrees of freedom
## Residual deviance: 2410.2  on 3166  degrees of freedom
## AIC: 2438.2
##
## Number of Fisher Scoring iterations: 5
```

The third model is built considering *diaBP*, *cigsPerDay* and the others eleven independent variables.

```
M3_CHD <- glm(TenYearCHD ~ male + age + education + cigsPerDay +
              BPMeds + prevalentStroke + prevalentHyp + diabetes +
              totChol + diaBP + BMI + heartRate + glucose,
              data=Training, family=binomial)

summary(M3_CHD)
```

```
##
## Call:
## glm(formula = TenYearCHD ~ male + age + education + cigsPerDay +
##      BPMeds + prevalentStroke + prevalentHyp + diabetes + totChol +
##      diaBP + BMI + heartRate + glucose, family = binomial, data = Training)
##
## Deviance Residuals:
```

```
##      Min      1Q   Median      3Q      Max
## -1.7861 -0.6041 -0.4281 -0.2839  2.5998
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -8.4502417  0.7652974 -11.042 < 2e-16 ***
## male           0.4454029  0.1151727   3.867  0.00011 ***
## age            0.0684774  0.0069392   9.868 < 2e-16 ***
## education     -0.0191648  0.0520637  -0.368  0.71280
## cigsPerDay     0.0228705  0.0045335   5.045 4.54e-07 ***
## BPMeds         0.3874054  0.2437343   1.589  0.11196
## prevalentStroke 0.7487508  0.5094807   1.470  0.14166
## prevalentHyp   0.4106393  0.1363456   3.012  0.00260 **
## diabetes       0.2279562  0.3387314   0.673  0.50097
## totChol        0.0026934  0.0012095   2.227  0.02595 *
## diaBP          0.0152969  0.0054290   2.818  0.00484 **
## BMI            0.0051356  0.0135993   0.378  0.70570
## heartRate     -0.0009533  0.0044496  -0.214  0.83036
## glucose        0.0071724  0.0024334   2.948  0.00320 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 2709.2  on 3179  degrees of freedom
## Residual deviance: 2406.2  on 3166  degrees of freedom
## AIC: 2434.2
##
## Number of Fisher Scoring iterations: 5
```

The last model is built considering *diaBP*, *currentSmoker*. and the others eleven independent variables.

```
M4_CHD <- glm(TenYearCHD ~ male + age + education + currentSmoker +
              BPMeds + prevalentStroke + prevalentHyp + diabetes +
              totChol + diaBP + BMI + heartRate + glucose,
              data=Training, family=binomial)

summary(M4_CHD)

##
## Call:
## glm(formula = TenYearCHD ~ male + age + education + currentSmoker +
##      BPMeds + prevalentStroke + prevalentHyp + diabetes + totChol +
##      diaBP + BMI + heartRate + glucose, family = binomial, data = Training)
##
## Deviance Residuals:
##      Min       1Q   Median      3Q      Max
## -1.8008 -0.6002 -0.4347 -0.2922  2.5442
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -8.4776129  0.7707585 -10.999 < 2e-16 ***
## male           0.5432392  0.1108204   4.902 9.49e-07 ***
```

```
## age            0.0662430  0.0068840   9.623 < 2e-16 ***
## education      -0.0205380  0.0519518  -0.395 0.692601
## currentSmoker   0.3996895  0.1132052   3.531 0.000415 ***
## BPMeds          0.3964521  0.2432602   1.630 0.103155
## prevalentStroke 0.7348772  0.5084014   1.445 0.148327
## prevalentHyp    0.4149860  0.1361507   3.048 0.002304 **
## diabetes        0.2125739  0.3384464   0.628 0.529947
## totChol         0.0027553  0.0012060   2.285 0.022331 *
## diaBP           0.0150674  0.0054119   2.784 0.005367 **
## BMI             0.0061823  0.0136320   0.454 0.650181
## heartRate       0.0004294  0.0044070   0.097 0.922375
## glucose         0.0070886  0.0024237   2.925 0.003448 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 2709.2 on 3179 degrees of freedom
## Residual deviance: 2418.6 on 3166 degrees of freedom
## AIC: 2446.6
##
## Number of Fisher Scoring iterations: 5
```

Considering the results obtained, it can be notice that in all the models six variables (*male*, *age*, *sysBP*, *prevalentHyp*, *totChol*, *glucose*) are significant while depending on the two pairs of correlated variables mentioned before, there are one or two significant variable.

To understand which is the best model to use, a comparison of the AIC method of the models is presented below.

```
AIC <- data.frame(M1_AIC= M1_CHD$aic, M2_AIC= M2_CHD$aic,
                  M3_AIC= M3_CHD$aic, M4_AIC= M4_CHD$aic)

AIC %>% kable(align= c('c'), digits = 3) %>%
  kable_styling(full_width = F, position = "center")
```

M1_AIC	M2_AIC	M3_AIC	M4_AIC
2426.095	2438.219	2434.182	2446.578

From results it is possible to claim that **Model 1** is the best model since it has the lowest value of AIC.

Then, the final model is obtained by considering only the significant variables of *Model 1*.

```
CHD_fm <- glm(TenYearCHD ~ male + age + cigsPerDay +
              sysBP + glucose, data=Training, family=binomial)

summary(CHD_fm)

##
## Call:
## glm(formula = TenYearCHD ~ male + age + cigsPerDay + sysBP +
##      glucose, family = binomial, data = Training)
##
```

```
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0575  -0.5925  -0.4356  -0.2953   2.6279
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.632628   0.454645 -18.988  < 2e-16 ***
## male         0.472970   0.112090   4.220 2.45e-05 ***
## age          0.064770   0.006875   9.421  < 2e-16 ***
## cigsPerDay   0.021871   0.004452   4.912 9.01e-07 ***
## sysBP        0.018386   0.002376   7.737 1.02e-14 ***
## glucose      0.007478   0.001874   3.990 6.61e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 2709.2  on 3179  degrees of freedom
## Residual deviance: 2412.9  on 3174  degrees of freedom
## AIC: 2424.9
##
## Number of Fisher Scoring iterations: 5
```

Therefore, the final logistic regression model obtained, with a $AIC = 2429.1$

3.3 Taks 4: Visualise the probabilities produced by the model.

In order to make predictions on the final model the *predict* function is used. Also to obtain the probabilities the type is *response*.

```
PTrain <- predict(CHD_fm, type="response")
summary(PTrain)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01794 0.06790 0.11819 0.15189 0.20152 0.93035
```

The *tapply* function allows us to find the mean of predictions with respect to each outcome of the *CHD*.

```
tapply(PTrain, Training$TenYearCHD, mean)
```

```
##           0           1
## 0.1363095 0.2388681
```

The table shows that for all of the true cases in which patient developed coronary heart disease, it is predicted an average probability of 0.24. On the other hand, for all of the true cases of $CHD = 0$, it is predicted an average probability of about 0.14.

3.4 Task 5: Choose a threshold using the ROC.

In order to convert probabilities to predictions, initially, a threshold value needs to be set. Indeed, if the probability of develop coronary heart disease is greater than this threshold value, it is predicted to suffer of CHD, while if this probability is less than the threshold value, then it is predicted the opposite.

The Receiver Operator Characteristic curve, or ROC curve, is used to define which value of the threshold is the best option. The sensitivity, or true positive rate of the model, is shown on the y-axis while the false positive rate is given on the x-axis. The line shows how these two outcome measures vary with different threshold values.

```
ROCpred <- prediction(PTrain, Training$TenYearCHD)
ROCperf <- performance(ROCpred, "tpr", "fpr")
plot(ROCperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))
```

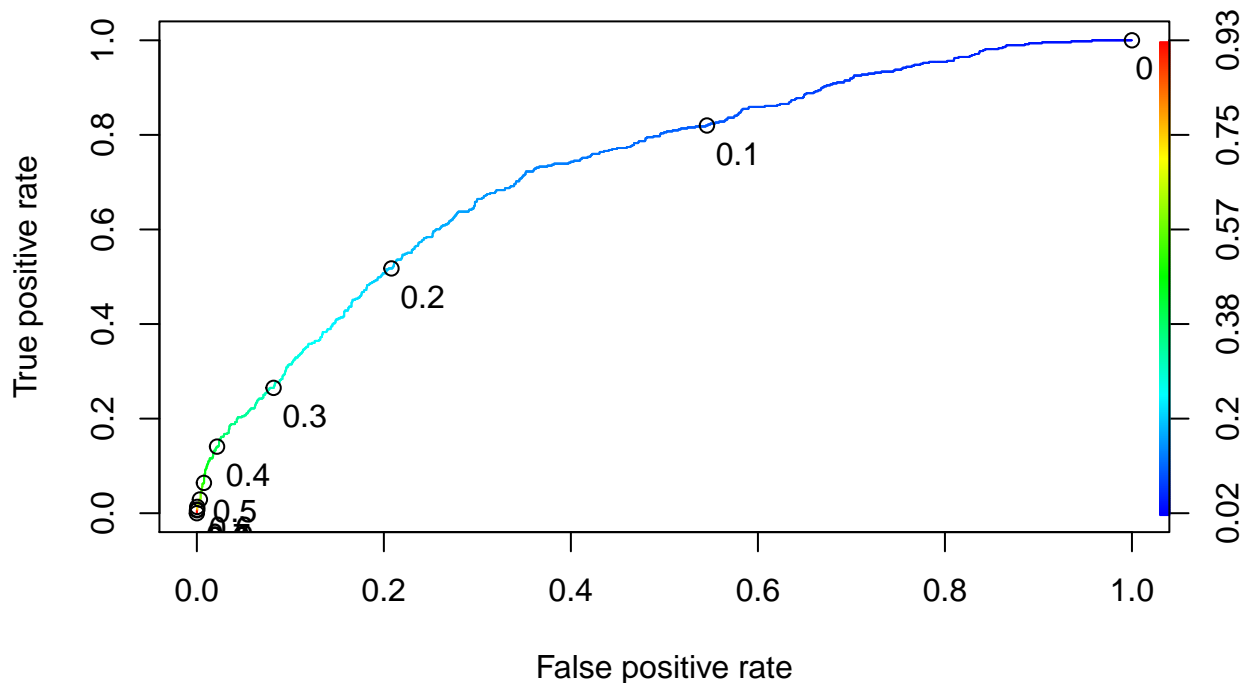


Figure 2: ROC curve

The threshold is selected considering the trade-off between false positive rate and true positive rate.

Since the model is predicting heart disease too many type II errors is not advisable. Indeed, a False Negative (ignoring the probability of disease when there actually is one) is more dangerous than a False Positive in this particular case.

Thus in order to increase the sensitivity, the threshold is set as 0.3.

3.5 Task 6: Evaluate its specificity and sensitivity and the AUC.

By using the threshold value set, the following confusion matrix is used to identify the sensitivity and specificity of model.

To evaluate the *specificity* and the *sensitivity*, the following formulas are used:

$$Sensitivity = \frac{TN}{TN + FP}$$

$$Specificity = \frac{TP}{TP + FN}$$

Now we calculate the level of *specificity* and *sensitivity* of the model obtained with the threshold equal to 0.4.

```
SS <- data.frame("Sensitivity" = comat[2,2]/(comat[2,1]+comat[2,2]),
                 "Specificity" = comat[1,1]/(comat[1,2]+comat[1,1]))

SS %>%
  kable(align= c('c'), digits = 2 )%>%
  kable_styling(full_width = F, position = "center")
```

Sensitivity	Specificity
0.27	0.92

Then the AUC coefficient is evaluated, knowing that the closer the AUC to 1, the better. The area under the ROC curve quantifies model classification accuracy; the higher the area, the greater the disparity between true and false positives, and the stronger the model in classifying members of the *Training* dataset.

```
AUC <- as.numeric(performance(ROCRpred, "auc")@y.values)
```

In this case the AUC is equal 0.7303468, which corresponds to the rate of successful classification by the logistic model used.

3.6 Task 7: Use the model and assess its predictions.

In this part, the **prediction** made previously is used on **Test** data set in order to check the model effectiveness of predicting patients' health status in a 10 years timeline.

To achieve, first it is required to evaluate the **accuracy** and compare it to **baseline** value.

```
predictions <- predict(CHD_fm, newdata=Test, type="response")

predTable <- table(Test$TenYearCHD, predictions > 0.4)
predTable
```

```
##
##      FALSE TRUE
##  0    872   27
##  1    137   24
```

```
accuracy <- table(predictions, Test[, "TenYearCHD"])

data.frame("Accuracy" = (predTable[2,2]+predTable[1,1])/
              (predTable[2,1]+predTable[2,2]+predTable[1,1]+predTable[1,2]),
           "Baseline" = (predTable[1,1]+predTable[1,2])/
              (predTable[2,1]+predTable[2,2]+predTable[1,1]+predTable[1,2]))
```

```
## Accuracy Baseline
## 1 0.845283 0.8481132
```

The actual accuracy is almost equal to the baseline, therefore, the model could be considered as good.

4 Additional Task

4.1 Change the proportion of observations between the training and testing set and repeat the task. Compare the results on the testing set.

For the additional task, we can change the ratio of split to 0.6 and then determine the training and test data set. After that the same steps would be taken to analyse, predict and find the accuracy of the model. It is obvious that by having a lower split case both parameter of accuracy and baseline values will drop because of the lower number of data for calculation. Finally, it can be concluded that using more number of data increase the AIC value which is expected to be minimum to claim one model is good, but also increase the true prediction power of the model.

5 References

- [1] R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- [2] Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, <https://doi.org/10.21105/joss.01686>
- [3] Hao Zhu (2020). kableExtra: Construct Complex Table with ‘kable’ and Pipe Syntax. R package version 1.2.1. <https://CRAN.R-project.org/package=kableExtra>
- [4] Erich Neuwirth (2014). RColorBrewer: ColorBrewer Palettes. R package version 1.1-2. <https://CRAN.R-project.org/package=RColorBrewer>
- [5] Taiyun Wei and Viliam Simko (2017). R package “corrplot”: Visualization of a Correlation Matrix (Version 0.84). Available from <https://github.com/taiyun/corrplot>
- [6] Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K.(2019). CaTools Basics and Extensions. R package version 2.1.0.
- [7] Sing T, Sander O, Beerenwinkel N, Lengauer T (2005). “ROCR: visualizing classifier performance in R.” *Bioinformatics*, 21(20), 7881. <URL: <http://rocr.bioinf.mpi-sb.mpg.de>>.

```
split
```

```
## [1] FALSE FALSE TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [13] TRUE TRUE FALSE TRUE TRUE FALSE TRUE FALSE FALSE TRUE TRUE TRUE
## [25] FALSE TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE FALSE
## [37] TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE FALSE FALSE TRUE
## [49] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [61] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [73] TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE FALSE FALSE TRUE TRUE TRUE
## [85] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [97] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [109] FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE FALSE TRUE TRUE
## [121] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE
```



```

## [133] TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
## [145] TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE
## [157] TRUE TRUE FALSE FALSE TRUE TRUE FALSE TRUE TRUE FALSE FALSE TRUE
## [169] FALSE FALSE TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE FALSE TRUE
## [181] FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
## [193] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [205] TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [217] TRUE TRUE TRUE FALSE TRUE FALSE FALSE FALSE FALSE TRUE TRUE FALSE
## [229] TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE FALSE FALSE
## [241] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE
## [253] TRUE FALSE TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE
## [265] TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [277] TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
## [289] TRUE TRUE TRUE FALSE FALSE TRUE FALSE FALSE TRUE TRUE TRUE TRUE
## [301] TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE
## [313] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE
## [325] TRUE TRUE TRUE FALSE FALSE TRUE FALSE FALSE TRUE FALSE TRUE TRUE
## [337] FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [349] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [361] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [373] TRUE TRUE TRUE FALSE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE
## [385] TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE
## [397] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE
## [409] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
## [421] FALSE TRUE TRUE TRUE FALSE TRUE TRUE FALSE FALSE TRUE TRUE TRUE
## [433] TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE
## [445] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE
## [457] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE FALSE
## [469] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE
## [481] TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE FALSE TRUE
## [493] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE
## [505] TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [517] TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE
## [529] TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE FALSE
## [541] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [553] TRUE TRUE FALSE FALSE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE
## [565] TRUE TRUE TRUE FALSE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE
## [577] TRUE FALSE TRUE TRUE FALSE FALSE TRUE TRUE FALSE TRUE FALSE TRUE
## [589] FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE FALSE TRUE TRUE TRUE
## [601] TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [613] FALSE TRUE FALSE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE FALSE
## [625] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE
## [637] TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE FALSE TRUE TRUE
## [649] TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE
## [661] TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE
## [673] TRUE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [685] TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [697] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE FALSE
## [709] TRUE TRUE FALSE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
## [721] TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [733] TRUE TRUE TRUE FALSE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [745] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE
## [757] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE
## [769] TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE

```

```

## [781] FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [793] TRUE TRUE TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE TRUE
## [805] FALSE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [817] TRUE TRUE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [829] TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE
## [841] FALSE FALSE TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [853] FALSE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [865] TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
## [877] FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [889] TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE
## [901] TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [913] FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [925] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [937] FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [949] TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE
## [961] TRUE FALSE FALSE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [973] TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE
## [985] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
## [997] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
## [1009] FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE
## [1021] TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE TRUE TRUE
## [1033] TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE TRUE TRUE
## [1045] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE FALSE FALSE TRUE TRUE
## [1057] TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE
## [1069] TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [1081] TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1093] TRUE TRUE TRUE TRUE FALSE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE
## [1105] TRUE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [1117] TRUE TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [1129] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1141] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [1153] TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
## [1165] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1177] FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
## [1189] TRUE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1201] FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [1213] TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE
## [1225] TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
## [1237] TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1249] TRUE TRUE FALSE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1261] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1273] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
## [1285] TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
## [1297] TRUE FALSE TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1309] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE FALSE TRUE TRUE TRUE
## [1321] FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE FALSE TRUE TRUE TRUE TRUE
## [1333] FALSE TRUE TRUE FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1345] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1357] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1369] TRUE FALSE TRUE TRUE FALSE FALSE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
## [1381] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1393] TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE
## [1405] FALSE TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [1417] TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE

```

##	[1429]	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE
##	[1441]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[1453]	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[1465]	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE
##	[1477]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[1489]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[1501]	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[1513]	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[1525]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[1537]	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE
##	[1549]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[1561]	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE
##	[1573]	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[1585]	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[1597]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[1609]	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[1621]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE
##	[1633]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[1645]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[1657]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[1669]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
##	[1681]	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE
##	[1693]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[1705]	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
##	[1717]	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[1729]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[1741]	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
##	[1753]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[1765]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[1777]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
##	[1789]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[1801]	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE
##	[1813]	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[1825]	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[1837]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[1849]	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[1861]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[1873]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[1885]	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
##	[1897]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
##	[1909]	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[1921]	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
##	[1933]	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
##	[1945]	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[1957]	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE

##	[2077]	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE
##	[2089]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[2101]	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
##	[2113]	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE
##	[2125]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[2137]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
##	[2149]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE
##	[2161]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
##	[2173]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE
##	[2185]	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
##	[2197]	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[2209]	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[2221]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
##	[2233]	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
##	[2245]	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE
##	[2257]	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[2269]	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
##	[2281]	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[2293]	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[2305]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[2317]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE
##	[2329]	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2341]	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE
##	[2353]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE
##	[2365]	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2377]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
##	[2389]	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[2401]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[2413]	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[2425]	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2437]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE
##	[2449]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2461]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[2473]	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2485]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
##	[2497]	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
##	[2509]	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2521]	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[2533]	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2545]	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[2557]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2569]	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[2581]	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
##	[2593]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2605]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
##	[2617]	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[2629]	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
##	[2641]	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2653]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[2665]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
##	[2677]	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[2689]	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2701]	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE
##	[2713]	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE

##	[2725]	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
##	[2737]	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE
##	[2749]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[2761]	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[2773]	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE
##	[2785]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[2797]	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2809]	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
##	[2821]	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
##	[2833]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE
##	[2845]	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[2857]	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE
##	[2869]	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
##	[2881]	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2893]	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2905]	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
##	[2917]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
##	[2929]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[2941]	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
##	[2953]	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[2965]	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[2977]	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[2989]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
##	[3001]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE
##	[3013]	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE
##	[3025]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
##	[3037]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE
##	[3049]	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[3061]	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[3073]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE
##	[3085]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE
##	[3097]	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[3109]	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[3121]	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE
##	[3133]	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[3145]	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[3157]	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE
##	[3169]	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[3181]	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[3193]	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[3205]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE
##	[3217]	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[3229]	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[3241]	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[3253]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE

##	[3373]	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
##	[3385]	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
##	[3397]	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[3409]	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
##	[3421]	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[3433]	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[3445]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[3457]	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE
##	[3469]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
##	[3481]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE
##	[3493]	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE
##	[3505]	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE
##	[3517]	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
##	[3529]	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE
##	[3541]	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE
##	[3553]	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
##	[3565]	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[3577]	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
##	[3589]	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[3601]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[3613]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[3625]	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[3637]	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE
##	[3649]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[3661]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
##	[3673]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
##	[3685]	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[3697]	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE
##	[3709]	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[3721]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[3733]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[3745]	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[3757]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE
##	[3769]	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[3781]	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE
##	[3793]	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[3805]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
##	[3817]	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
##	[3829]	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[3841]	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[3853]	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
##	[3865]	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE
##	[3877]	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
##	[3889]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
##	[3901]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE

```

## [4021] TRUE TRUE TRUE FALSE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
## [4033] TRUE TRUE FALSE FALSE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
## [4045] TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE
## [4057] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE FALSE
## [4069] TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE
## [4081] TRUE TRUE FALSE TRUE FALSE TRUE FALSE FALSE TRUE FALSE FALSE TRUE
## [4093] FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE
## [4105] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
## [4117] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [4129] FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE
## [4141] TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [4153] FALSE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE FALSE TRUE
## [4165] TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [4177] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [4189] FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE TRUE FALSE TRUE
## [4201] TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [4213] TRUE TRUE FALSE FALSE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
## [4225] TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [4237] TRUE TRUE TRUE TRUE

```