Big 5 Notebook

```r
library('mda')
```

```
## Loading required package: class
## Loaded mda 0.4-10
```

```r
library('MASS')
library('klaR')
library('nnet')
library('kernlab')
library('caret')
```

```
## Loading required package: lattice
## Loading required package: ggplot2
##
## Attaching package: 'ggplot2'
## The following object is masked from 'package:kernlab':
##
##     alpha
```

```r
library('e1071')
library("tidyverse")
```

```
## -- Attaching packages ---------------------------------- tidyverse 1.2.1 --
## v tibble  1.4.2     v purrr   0.2.4
## v tidyr   0.8.0     v dplyr   0.7.4
## v readr   1.1.1     v stringr 1.3.0
## v tibble  1.4.2     v forcats 0.3.0
## -- Conflicts ------------------------------------- tidyverse_conflicts() --
## x ggplot2::alpha() masks kernlab::alpha()
## x purrr::cross()   masks kernlab::cross()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x purrr::lift()    masks caret::lift()
## x dplyr::select()  masks MASS::select()
```

```r
library("keras")
```

```r
data <- read.csv("./data/data.csv") %>%
  filter(gender %in% c(1,2))
data_gu <- read.csv("./data/data.csv") %>%
  filter(gender %in% c(3))
str(data)
```

```
## 'data.frame':    19595 obs. of  57 variables:
##  $ race   : int  3 1 3 13 1 3 11 13 5 4 ...
##  $ age    : int  21 17 53 46 14 19 25 31 20 23 ...
##  $ engnat : int  1 2 1 1 2 2 2 1 1 2 ...
##  $ gender : int  1 2 1 2 2 2 2 2 2 1 ...
##  $ hand   : int  1 5 1 1 1 1 1 1 1 1 ...
##  $ source : int  2 6 1 1 1 1 2 2 5 2 ...
```

```
##  $ country: Factor w/ 160 levels "","(nu","A1",..: 31 31 151 151 122 129 151 151 151 74 ...
##  $ E1     : int  2 4 4 2 5 2 3 1 5 4 ...
##  $ E2     : int  2 1 2 2 1 5 1 5 1 3 ...
##  $ E3     : int  4 4 5 3 1 2 3 2 5 5 ...
##  $ E4     : int  3 1 2 3 4 4 3 4 1 3 ...
##  $ E5     : int  4 4 5 3 5 3 3 1 5 5 ...
##  $ E6     : int  2 1 1 3 1 4 1 3 1 1 ...
##  $ E7     : int  2 3 4 1 1 3 3 2 5 4 ...
##  $ E8     : int  2 1 3 5 5 4 1 4 4 3 ...
##  $ E9     : int  4 4 5 1 5 4 3 1 4 4 ...
##  $ E10    : int  2 3 1 5 1 5 5 5 1 3 ...
##  $ N1     : int  1 4 1 2 5 5 3 1 2 1 ...
##  $ N2     : int  4 2 5 3 1 4 3 5 4 4 ...
##  $ N3     : int  2 5 2 4 5 4 3 4 2 4 ...
##  $ N4     : int  2 2 5 2 5 2 4 5 4 4 ...
##  $ N5     : int  1 2 1 3 5 4 3 1 2 1 ...
##  $ N6     : int  1 5 1 4 5 5 3 4 2 1 ...
##  $ N7     : int  1 5 1 3 5 5 3 4 3 1 ...
##  $ N8     : int  2 4 1 2 5 5 3 1 2 1 ...
##  $ N9     : int  2 2 1 2 5 4 3 5 2 1 ...
##  $ N10    : int  3 1 1 4 5 5 4 2 2 1 ...
##  $ A1     : int  2 5 1 1 5 2 5 2 5 2 ...
##  $ A2     : int  3 2 5 3 1 5 5 2 5 5 ...
##  $ A3     : int  1 5 1 3 5 4 3 3 1 1 ...
##  $ A4     : int  2 1 5 4 5 4 5 4 5 4 ...
##  $ A5     : int  2 5 2 4 1 3 1 3 1 3 ...
##  $ A6     : int  3 1 3 4 5 5 5 4 5 3 ...
##  $ A7     : int  2 5 1 2 1 3 1 3 1 1 ...
##  $ A8     : int  3 5 5 3 5 4 5 5 5 3 ...
##  $ A9     : int  2 5 4 4 5 4 5 5 4 4 ...
##  $ A10    : int  3 5 5 3 5 3 5 3 5 5 ...
##  $ C1     : int  3 5 4 4 4 3 3 2 2 4 ...
##  $ C2     : int  2 5 1 1 1 3 1 5 4 2 ...
##  $ C3     : int  4 3 5 3 5 4 5 4 3 5 ...
##  $ C4     : int  1 1 1 2 1 5 3 3 3 1 ...
##  $ C5     : int  4 1 5 3 5 1 3 3 3 4 ...
##  $ C6     : int  1 4 1 1 1 4 1 4 3 1 ...
##  $ C7     : int  4 3 4 5 5 5 1 5 3 4 ...
##  $ C8     : int  2 2 1 1 1 4 3 3 3 1 ...
##  $ C9     : int  4 5 4 4 5 2 3 5 3 3 ...
##  $ C10    : int  4 2 5 4 5 3 3 3 3 5 ...
##  $ O1     : int  4 2 4 3 4 4 3 4 3 3 ...
##  $ O2     : int  1 5 1 3 5 3 1 2 1 1 ...
##  $ O3     : int  4 2 3 3 5 5 1 1 5 5 ...
##  $ O4     : int  1 5 1 3 1 2 1 3 1 1 ...
##  $ O5     : int  3 1 5 2 5 4 3 3 4 4 ...
##  $ O6     : int  2 5 1 3 1 2 1 5 1 1 ...
##  $ O7     : int  5 2 4 3 5 5 3 5 4 5 ...
##  $ O8     : int  4 5 2 1 5 2 1 4 3 3 ...
##  $ O9     : int  4 5 5 3 5 5 5 5 3 2 ...
##  $ O10    : int  4 NA 5 2 5 5 3 3 4 5 ...
```

```r
#summary(data)
```

Only select rows with predictable genders. Non-specified could also be considered if there were more samples, but at the time of writing this only **102** observations are present in the data set. In the original question set several questions were posed such that they measured the low-end of the trait rather than the high end. These values have been reversed.

Trait Definitions:
O: Low Openness -> High Openness
C: Low Conscientiousness -> High Conscientiousness
E: Intraversion -> Extraversion
A: Disagreeableness -> Agreeableness
N: Low Neuroticism -> High Neuroticism

Define R function to correct for question

```
qCorrect <- function(n){
  switch(as.character(n),
         '1' = 5,
         '2' = 4,
         '3' = 3,
         '4' = 2,
         '5' = 1, 0)
}
qCorrect <- Vectorize(qCorrect) # Vectorize function to allow interoperation with dplyr::mutate
```

```
data_qc <- data %>%
              mutate_at(
                .vars = c("E2", "E4", "E6", "E8", "E10", "N2",
                          "N4", "A1", "A3", "A5", "A7", "C2",
                          "C4", "C6", "C8", "O2", "O4", "O6"), funs(qCorrect)) %>%
              mutate(sex = ifelse(gender == 1, 0, 1))
#str(data_qc)
```

Calculate score based on 10 relevant questions to metric

```
data_rs <- data_qc %>%
            mutate(o_sum = rowSums(.[8:17]))  %>%
            mutate(c_sum = rowSums(.[18:27])) %>%
            mutate(e_sum = rowSums(.[28:37])) %>%
            mutate(a_sum = rowSums(.[38:47])) %>%
            mutate(n_sum = rowSums(.[48:57]))
#str(data_rs)
```

Calculate percentile scoring

```
data_ps <- data_rs %>%
            # mutate_at(.vars = c("o_sum","c_sum","e_sum","a_sum","n_sum"), funs(percent_rank))
            # Can replicate the below functionality with the above line, but this overwrites the column
            mutate(o_ps = percent_rank(o_sum)) %>%
            mutate(c_ps = percent_rank(c_sum)) %>%
            mutate(e_ps = percent_rank(e_sum)) %>%
            mutate(a_ps = percent_rank(a_sum)) %>%
            mutate(n_ps = percent_rank(n_sum))

data_shaped <- data_ps[,c(58:68)] # sums, percent scores, and sex id
big5 <- data_ps[,c(58,64:68)] # percent scores and sex id
# str(data_shaped)
```

```r
# Formulate proper data model
set.seed(5)
n <- nrow(big5)
big5_a <- big5[sample(n),] # Shuffle dataset
#big5_a <- big5 # Unshuffed dataset
str(big5_a)
```

```
## 'data.frame':    19595 obs. of  6 variables:
##  $ sex : num  1 1 0 1 1 0 0 1 0 1 ...
##  $ o_ps: num  0.140094 0.920996 0.361335 0.512759 0.000051 ...
##  $ c_ps: num  0.641 0.943 0.821 0.434 0.981 ...
##  $ e_ps: num  0.00566 0.25809 0.06512 0.83352 0.06512 ...
##  $ a_ps: num  0.2557 0.9491 0.7836 0.0889 0.0279 ...
##  $ n_ps: num  0.951 0.558 0.785 0.124 0.884 ...
```

```r
big5 <- big5_a
big5$x <- big5[,2:6] # Select only relevant columns
big5$y <- big5$sex # Convert y to categorical
```

Mixture Discriminant Analysis

```r
mda_fit <- mda(sex~., data = train_a)
summary(mda_fit)
```

```
##                   Length Class   Mode
## percent.explained  5     -none-  numeric
## values             5     -none-  numeric
## means             30     -none-  numeric
## theta.mod         25     -none-  numeric
## dimension          1     -none-  numeric
## sub.prior          2     -none-  list
## fit                4     polyreg list
## call               3     -none-  call
## weights            2     -none-  list
## prior              2     table   numeric
## assign.theta       2     -none-  list
## deviance           1     -none-  numeric
## confusion          4     table   numeric
## terms              3     terms   call
```

```r
mda_predictions <-predict(mda_fit,test_a[,2:6])
pred_table <-table(mda_predictions,test_a$sex)
confusionMatrix(as.factor(mda_predictions),as.factor(test_a$sex))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  560  375
##          1  916 2069
##
##                Accuracy : 0.6707
##                  95% CI : (0.6557, 0.6854)
##     No Information Rate : 0.6235
##     P-Value [Acc > NIR] : 4.21e-10
##
```

```
##                  Kappa : 0.2437
##  Mcnemar's Test P-Value : < 2.2e-16
##
##            Sensitivity : 0.3794
##            Specificity : 0.8466
##         Pos Pred Value : 0.5989
##         Neg Pred Value : 0.6931
##             Prevalence : 0.3765
##         Detection Rate : 0.1429
##   Detection Prevalence : 0.2385
##      Balanced Accuracy : 0.6130
##
##       'Positive' Class : 0
##
```

```
mda_predictions
```

```
##    [1] 0 1 0 1 0 0 1 0 1 0 1 1 1 0 0 1 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 0 1 1
##   [35] 0 0 1 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
##   [69] 0 1 1 1 0 1 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1
##  [103] 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 0 0 1 0 1 1 1 1
##  [137] 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 0 0 1 0 1 1 1 1 0 1 0 1 1 0
##  [171] 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 0 0 1 1
##  [205] 0 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 0 1 0 1 1 1 0 1 0 1 0 1 1 1
##  [239] 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0
##  [273] 0 1 1 1 1 1 0 0 1 1 1 0 1 0 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1
##  [307] 0 1 1 0 1 1 1 0 1 1 1 0 1 0 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 0 0 1 1
##  [341] 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 0 0 1 1 1 0 1 1 1 1 1
##  [375] 1 1 0 1 1 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1
##  [409] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 0 0
##  [443] 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [477] 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
##  [511] 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 0 1 0
##  [545] 1 1 0 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 0
##  [579] 1 0 0 0 0 0 1 1 1 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1
##  [613] 1 1 1 0 1 1 1 1 1 0 0 1 0 1 1 1 0 1 0 1 1 0 0 0 1 1 0 1 1 0 1 1 0 1
##  [647] 1 1 0 1 1 1 1 1 0 1 0 1 0 1 0 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1 1 1 1
##  [681] 0 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 1 1 1
##  [715] 1 0 0 1 0 1 0 1 1 1 1 0 1 0 0 1 1 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1
##  [749] 0 1 1 1 1 1 0 0 1 1 0 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1
##  [783] 1 1 1 1 0 1 1 0 0 1 0 1 1 0 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1
##  [817] 1 1 1 1 0 1 1 0 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1
##  [851] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0
##  [885] 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
##  [919] 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 0 0 0 1
##  [953] 1 0 1 0 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 1 0 0 1 1 0 1
##  [987] 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 0 0 0 1 1 1 0 0 0 1 0 1
## [1021] 1 1 1 1 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 0 1 1 0 0
## [1055] 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 0 0 0 1 1 0 1 1 1 0 1 0 1 1 1 1 1 1 1
## [1089] 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
## [1123] 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 1 1 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 1
## [1157] 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0
## [1191] 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0 1 0 0 1 1 1 1 1 1 1 1 0 1 1 0
## [1225] 1 1 0 1 1 0 1 0 1 1 1 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1
## [1259] 0 0 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1
```

```
## [1293] 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 1 1 0 0 1
## [1327] 1 1 1 1 1 1 1 1 0 0 1 1 0 1 1 1 0 1 1 1 1 0 1 1 0 0 0 1 1 1 0 1 0 1
## [1361] 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
## [1395] 0 1 1 0 0 0 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 0 0 0 1 1 1 1 0 1 1
## [1429] 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 0 0 0 1 1 1 1 1 1 1 0 1 1 0 1
## [1463] 0 0 1 0 1 0 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 0 0 1 1 1 1 1 1 1 0 1 1
## [1497] 1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 0
## [1531] 0 1 1 0 0 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1
## [1565] 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 0 1 1 1 1 1 0 1 1 1 0 1 1
## [1599] 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0
## [1633] 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 0 1 1 1
## [1667] 1 1 0 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1
## [1701] 1 1 0 1 1 0 1 0 1 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0
## [1735] 1 1 1 0 1 1 0 1 1 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1
## [1769] 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 1 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1
## [1803] 1 1 1 1 1 1 1 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0 1 0 1 1
## [1837] 0 1 1 1 0 0 1 0 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 1 0 0 0 1 1 1 0
## [1871] 1 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 0 1 1 0 1 1 1 1
## [1905] 1 1 1 1 0 1 0 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
## [1939] 1 0 0 1 1 1 1 1 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1
## [1973] 1 1 1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
## [2007] 1 1 1 0 1 1 1 0 1 1 1 1 1 0 0 1 1 1 0 1 1 1 0 1 0 1 1 1 0 1 0 1 1 1
## [2041] 1 0 0 0 0 1 1 1 1 1 1 0 1 1 0 1 0 1 0 0 0 1 1 1 1 1 0 1 1 0 1 1 1 1
## [2075] 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 1 1 1 0 1 1
## [2109] 0 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1
## [2143] 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1 1
## [2177] 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 1 1 0 1 1 1 0 1 1 1 0 1
## [2211] 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 1 0 1 1 1 0 0 1 0 1 0 1
## [2245] 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1
## [2279] 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1
## [2313] 1 1 1 1 1 0 0 1 1 0 1 1 1 0 0 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
## [2347] 0 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 0 0 1 1 1 1 1 1 1 1
## [2381] 1 1 1 0 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1 0 1 0
## [2415] 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 0 1 0 1 1 0 1 1 0 1
## [2449] 1 1 0 1 1 0 1 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0
## [2483] 1 0 1 0 1 1 1 1 0 1 1 1 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1
## [2517] 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 0 1 1 1 1 1 1
## [2551] 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 1 1 1 0 1 1 1 0 0 0 1 1 0 1 1 1 0 0
## [2585] 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
## [2619] 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 1 0 1 1 1 1 1 1
## [2653] 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 1 1 0 1 0 1 0 1
## [2687] 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1
## [2721] 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1
## [2755] 1 1 1 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 1 0 1 1 1 1 1 1 0 1
## [2789] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 0 1 0 1 1 1 1 1 0
## [2823] 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 0 0 1 1 1 0 1 0 1 0 1 1 1 0
## [2857] 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0
## [2891] 0 0 1 1 1 0 0 1 1 0 0 0 1 0 1 1 0 1 0 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1
## [2925] 1 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 0 0 1 1 1 0 1 1 1 1 0 1
## [2959] 1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1
## [2993] 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 0 1 1 1 1 0 1 1 1 1
## [3027] 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 0 0 0 1 1 1 0 1 0 1 1
## [3061] 1 0 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3095] 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
```

```
## [3129] 0 1 0 1 0 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0
## [3163] 1 1 1 1 1 0 1 1 1 1 0 0 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 0 1 1 1
## [3197] 1 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 0 0
## [3231] 1 1 1 0 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 0
## [3265] 1 0 1 1 1 0 0 0 0 0 1 0 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1
## [3299] 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1
## [3333] 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 1 0 1 1 1 1 1 1
## [3367] 1 0 1 1 1 0 0 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 0 0 1 1 0 1 1 1 0
## [3401] 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 0 1 1 1 1 1 0
## [3435] 1 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 1 1
## [3469] 1 1 1 1 1 0 1 1 0 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 0 0 1 1
## [3503] 1 1 1 1 1 0 1 0 0 1 1 1 1 1 0 1 1 1 0 1 1 0 0 1 1 1 1 0 1 0 1 0 0 0
## [3537] 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0 1 0 0 0 1 1 0 1 1 1 0 1 1 1 0 0 0 1
## [3571] 1 1 1 1 0 1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 0
## [3605] 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 0 0 1 1 1 1 0 0 0 1 1 1 1 1
## [3639] 1 1 1 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0
## [3673] 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1
## [3707] 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 0 1 1 0 1 1
## [3741] 1 0 0 1 1 0 1 0 0 1 1 0 1 1 1 1 1 0 1 1 0 0 1 0 0 1 0 1 1 1 1 1 1 1
## [3775] 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0
## [3809] 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 1 0 1 0 1 1 1 1
## [3843] 1 1 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3877] 1 1 0 0 1 1 0 0 1 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 0 1 0 1 1 1 1 1 1
## [3911] 1 0 0 1 1 1 1 1 1 0
## Levels: 0 1
```

```
# 0.6642 accuracy
```

Quadratic Discriminant Analysis

```
qda_fit <- qda(sex~., data = train_a)
summary(qda_fit)
```

```
##           Length Class  Mode
## prior      2     -none- numeric
## counts     2     -none- numeric
## means     10     -none- numeric
## scaling   50     -none- numeric
## ldet       2     -none- numeric
## lev        2     -none- character
## N          1     -none- numeric
## call       3     -none- call
## terms      3     terms  call
## xlevels    0     -none- list
## na.action  1     omit   numeric
```

```
qda_predictions <-predict(qda_fit,test_a[,2:6])$class
confusionMatrix(as.factor(qda_predictions),as.factor(test_a$sex))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  565  365
##          1  911 2079
##
```

```
##                Accuracy : 0.6745
##                  95% CI : (0.6596, 0.6892)
##     No Information Rate : 0.6235
##     P-Value [Acc > NIR] : 1.554e-11
##
##                   Kappa : 0.2519
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3828
##             Specificity : 0.8507
##          Pos Pred Value : 0.6075
##          Neg Pred Value : 0.6953
##              Prevalence : 0.3765
##          Detection Rate : 0.1441
##    Detection Prevalence : 0.2372
##       Balanced Accuracy : 0.6167
##
##        'Positive' Class : 0
##
```

```
qda_predictions
```

```
##    [1] 0 1 1 1 1 0 1 0 1 0 1 1 1 0 0 1 0 1 1 1 0 0 1 1 0 1 1 1 1 0 1 0 1 1
##   [35] 0 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
##   [69] 0 1 1 1 0 1 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 1 1 1
##  [103] 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 0 1 1 1 1
##  [137] 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 0 1 1 0
##  [171] 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 0 1 1
##  [205] 0 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 0 1 0 1 1 1 0 1 0 1 0 1 1 1
##  [239] 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0
##  [273] 0 1 1 1 1 1 0 1 1 1 1 0 1 0 1 1 1 1 0 0 1 0 1 1 1 0 1 1 1 1 1 1 0 1
##  [307] 0 1 1 1 0 1 1 0 1 1 1 0 1 0 0 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 0 0 1 1
##  [341] 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 1 1 0 1 1 1 1 1
##  [375] 1 1 0 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0 1 1
##  [409] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 1 1
##  [443] 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1
##  [477] 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
##  [511] 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 0 1 0
##  [545] 1 1 0 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 0
##  [579] 1 0 0 0 0 0 1 1 1 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1
##  [613] 1 1 1 0 1 0 1 1 1 0 0 1 1 1 1 1 0 1 0 1 1 0 0 0 1 1 0 1 1 0 1 1 0 1
##  [647] 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1 1 1 1
##  [681] 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 0 0 0 0 1 1 1
##  [715] 1 0 0 1 0 1 0 1 1 1 1 1 1 0 0 1 1 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1
##  [749] 0 1 1 1 1 1 0 0 1 1 0 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1
##  [783] 1 1 1 1 1 1 1 0 0 1 0 1 1 0 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1
##  [817] 1 1 1 1 0 1 1 0 1 1 1 1 0 1 0 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1 1
##  [851] 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 1 0
##  [885] 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
##  [919] 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 0 0 0 1
##  [953] 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 1 0 0 1 1 0 1
##  [987] 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 0 0 0 1 1 1 0 0 0 1 0 1
## [1021] 1 1 1 1 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 0 1 1 0 0
## [1055] 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0 1 1 0 1 1 0 0 1 0 1 1 1 1 1 1 1
## [1089] 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1
```

```
## [1123] 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 0 0 1 1 1 0 0 0 1 1 0 1 1
## [1157] 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0
## [1191] 1 1 1 1 1 1 0 1 1 1 0 1 1 0 1 1 1 1 0 1 1 0 1 1 1 1 0 1 1 1 0 1 1 0
## [1225] 1 1 0 1 1 0 1 0 1 1 1 0 0 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 1 0 1 1
## [1259] 0 0 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 1 0 1 0 1 1 1 0 1 1 0 1 1 1 1
## [1293] 1 0 1 0 1 1 1 0 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 1 0 0 1 1
## [1327] 1 1 1 1 1 1 0 1 0 0 1 1 0 1 1 1 1 1 1 1 1 0 1 1 0 0 0 1 1 1 0 1 0 1
## [1361] 0 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
## [1395] 0 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 0 0 1 1 1 1 0 1 1
## [1429] 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 0 0 0 1 1 1 1 1 1 0 1 1 0 1
## [1463] 0 0 1 0 1 0 1 1 1 1 1 1 1 1 0 0 1 0 0 1 1 1 0 0 1 1 1 1 1 1 1 0 1 1
## [1497] 1 0 1 1 1 1 1 1 1 0 1 1 1 1 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0
## [1531] 0 1 1 0 0 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1
## [1565] 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 0 1 1 1 1 1 0 1 1 1 1 1 1
## [1599] 1 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1
## [1633] 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 0 1 1 1
## [1667] 1 1 0 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1
## [1701] 1 1 0 1 1 1 1 0 1 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
## [1735] 1 1 1 0 1 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1
## [1769] 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1
## [1803] 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0 1 0 1 1
## [1837] 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 0 0 0 1 1 1 1 0
## [1871] 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 0 1 1 0 1 1 1 1
## [1905] 1 1 1 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1
## [1939] 1 0 0 1 1 1 1 1 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1 1 0
## [1973] 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
## [2007] 1 1 1 0 1 1 1 0 1 1 1 1 1 0 0 1 1 1 0 1 1 1 0 1 0 1 1 0 0 1 0 1 1 1
## [2041] 1 0 0 1 0 1 1 1 1 1 1 0 1 1 0 1 0 1 0 0 0 1 1 1 1 1 0 1 1 0 1 1 1 1
## [2075] 1 1 0 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 0 1 1 1 1 0 1 1 1 1
## [2109] 0 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 0 0 1
## [2143] 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 1 1
## [2177] 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1 1 0 1 1 1 0 1
## [2211] 1 1 0 1 1 1 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 0 1 1 1 0 0 1 0 1 0 1
## [2245] 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1
## [2279] 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1
## [2313] 1 1 1 1 1 0 0 1 1 0 1 1 1 0 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
## [2347] 0 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 0 0 1 1 1 1 1 1 1 1
## [2381] 1 1 1 0 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 1 0 1 0
## [2415] 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 0 0 1 0 1 1 0 1 1 0 1
## [2449] 1 1 1 1 1 0 1 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0
## [2483] 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 1
## [2517] 1 1 1 1 1 0 1 1 1 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 1 1 1
## [2551] 1 1 0 1 0 1 1 0 1 0 0 0 1 0 1 1 1 1 1 0 1 1 1 0 0 0 1 1 0 1 1 1 0 0
## [2585] 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 0 1
## [2619] 1 1 0 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 0 1 1 1 0 1 1 1 1 1 1
## [2653] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 1 1 0 1
## [2687] 1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1
## [2721] 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1
## [2755] 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 0 1 1 0 1 1 0 1 1 1 1 1 0 1
## [2789] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 0 1 1 0 1 0 1 1 1 1 0
## [2823] 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 0 1 1 1 0 1 0 1 0 1 1 1 0
## [2857] 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0
## [2891] 0 0 1 1 1 0 0 1 1 0 0 0 1 0 1 1 0 1 0 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1
## [2925] 1 0 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 0 1 1 1 0 1 1 0 1 0 1
```

```
## [2959] 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1
## [2993] 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 0 1 1 1 1 0 1 1 1 1
## [3027] 1 1 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1 1 0 1 0 1 1 1 0 1 0 1 1
## [3061] 1 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3095] 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3129] 0 1 0 1 0 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0
## [3163] 0 1 1 1 1 0 1 1 1 1 0 0 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 0 1 0 0 1 1 1
## [3197] 1 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 0 0
## [3231] 1 1 1 0 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 0
## [3265] 1 0 1 1 1 0 0 0 0 0 1 0 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1
## [3299] 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
## [3333] 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0 0 1 1 1 1 1 1 1 1
## [3367] 1 0 1 1 1 0 0 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 0 0 0 1 0 1 1 1 0
## [3401] 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 0 1 1 1 1 1 0
## [3435] 1 1 0 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 1 1
## [3469] 1 1 1 1 0 0 1 1 0 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 0 0 1 1
## [3503] 1 1 1 1 1 0 1 0 0 1 1 1 1 1 0 1 1 1 0 1 1 0 0 1 1 1 1 0 1 0 1 0 0 0
## [3537] 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0 1 0 0 0 1 1 0 1 1 1 0 1 1 1 0 0 0 1
## [3571] 1 1 1 1 0 1 0 1 1 1 0 1 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 0
## [3605] 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 0 0 1 1 1 1 0 1 0 1 1 1 1 1
## [3639] 1 1 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 0
## [3673] 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1
## [3707] 1 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1 0 0 1
## [3741] 1 0 0 1 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 0 0 1 0 1 1 1 0 1 1 1
## [3775] 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 0
## [3809] 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 0 1 0 1 0 1 1 1 1
## [3843] 1 1 1 1 1 0 0 1 0 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3877] 1 0 0 0 1 1 0 0 1 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 0 1 0 1 1 1 1 1 1
## [3911] 1 0 0 1 1 1 0 1 1 0
## Levels: 0 1
```

```r
# 0.6675 accuracy
```

Regularized Discriminant Analysis

```r
rda_fit <- rda(sex~., data = train_a, gamma = 0.05, lambda = 0.01)
summary(qda_fit)
```

```
##            Length Class  Mode
## prior       2     -none- numeric
## counts      2     -none- numeric
## means      10     -none- numeric
## scaling    50     -none- numeric
## ldet        2     -none- numeric
## lev         2     -none- character
## N           1     -none- numeric
## call        3     -none- call
## terms       3     terms  call
## xlevels     0     -none- list
## na.action   1     omit   numeric
```

```r
rda_predictions <-predict(rda_fit,test_a[,2:6])$class
confusionMatrix(as.factor(rda_predictions),as.factor(test_a$sex))
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction    0    1
##          0  563  364
##          1  913 2080
##
##                Accuracy : 0.6742
##                  95% CI : (0.6593, 0.6889)
##     No Information Rate : 0.6235
##     P-Value [Acc > NIR] : 1.951e-11
##
##                   Kappa : 0.251
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3814
##             Specificity : 0.8511
##          Pos Pred Value : 0.6073
##          Neg Pred Value : 0.6950
##              Prevalence : 0.3765
##          Detection Rate : 0.1436
##    Detection Prevalence : 0.2365
##       Balanced Accuracy : 0.6163
##
##        'Positive' Class : 0
##
```

```
# 0.667 accuracy
```

Neural Net

```
nnet_fit <- nnet(as.factor(sex)~., data=train_a, size=4, decay=0.0001, maxit=500)
```

```
## # weights:  29
## initial  value 9491.125087
## iter  10 value 7902.890879
## iter  20 value 7844.445236
## iter  30 value 7831.740414
## iter  40 value 7830.889556
## iter  50 value 7830.094918
## iter  60 value 7817.912036
## iter  70 value 7814.959182
## iter  80 value 7813.313298
## iter  90 value 7812.836348
## iter 100 value 7811.791308
## iter 110 value 7811.611890
## iter 120 value 7811.348237
## iter 130 value 7811.162742
## iter 140 value 7810.994719
## iter 150 value 7810.850920
## final  value 7810.751542
## converged
```

```
summary(nnet_fit)
```

```
## a 5-4-1 network with 29 weights
## options were - entropy fitting  decay=1e-04
##  b->h1 i1->h1 i2->h1 i3->h1 i4->h1 i5->h1
```

```
##   2.97  -3.06  -2.44  -0.67   1.37   1.86
##  b->h2 i1->h2 i2->h2 i3->h2 i4->h2 i5->h2
##  17.92   6.14   0.24 -21.48  -6.75   5.72
##  b->h3 i1->h3 i2->h3 i3->h3 i4->h3 i5->h3
##  -7.80   4.54  59.29   6.82  -3.72  -2.37
##  b->h4 i1->h4 i2->h4 i3->h4 i4->h4 i5->h4
##  -1.60  -0.79   2.00   3.49   1.42  -1.11
##   b->o  h1->o  h2->o  h3->o  h4->o
##   1.11  -1.67  -1.28   0.60   2.01
```

```
str(nnet_fit)
```

```
## List of 20
##  $ n            : num [1:3] 5 4 1
##  $ nunits       : int 11
##  $ nconn        : num [1:12] 0 0 0 0 0 0 0 6 12 18 ...
##  $ conn         : num [1:29] 0 1 2 3 4 5 0 1 2 3 ...
##  $ nsunits      : int 11
##  $ decay        : num 1e-04
##  $ entropy      : logi TRUE
##  $ softmax      : logi FALSE
##  $ censored     : logi FALSE
##  $ value        : num 7811
##  $ wts          : num [1:29] 2.971 -3.065 -2.444 -0.675 1.374 ...
##  $ convergence  : int 0
##  $ fitted.values: num [1:12735, 1] 0.756 0.506 0.786 0.439 0.794 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:12735] "13427" "17965" "5572" "2051" ...
##   .. ..$ : NULL
##  $ residuals    : num [1:12735, 1] 0.244 -0.506 0.214 0.561 -0.794 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:12735] "13427" "17965" "5572" "2051" ...
##   .. ..$ : NULL
##  $ lev          : chr [1:2] "0" "1"
##  $ call         : language nnet.formula(formula = as.factor(sex) ~ ., data = train_a, size = 4,
##  $ terms        :Classes 'terms', 'formula'  language as.factor(sex) ~ o_ps + c_ps + e_ps + a_ps + n_
##   .. ..- attr(*, "variables")= language list(as.factor(sex), o_ps, c_ps, e_ps, a_ps, n_ps)
##   .. ..- attr(*, "factors")= int [1:6, 1:5] 0 1 0 0 0 0 0 0 1 0 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:6] "as.factor(sex)" "o_ps" "c_ps" "e_ps" ...
##   .. .. .. ..$ : chr [1:5] "o_ps" "c_ps" "e_ps" "a_ps" ...
##   .. ..- attr(*, "term.labels")= chr [1:5] "o_ps" "c_ps" "e_ps" "a_ps" ...
##   .. ..- attr(*, "order")= int [1:5] 1 1 1 1 1
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(as.factor(sex), o_ps, c_ps, e_ps, a_ps, n_ps)
##   .. ..- attr(*, "dataClasses")= Named chr [1:6] "factor" "numeric" "numeric" "numeric" ...
##   .. .. ..- attr(*, "names")= chr [1:6] "as.factor(sex)" "o_ps" "c_ps" "e_ps" ...
##  $ coefnames    : chr [1:5] "o_ps" "c_ps" "e_ps" "a_ps" ...
##  $ na.action    : 'omit' Named int 5404
##   ..- attr(*, "names")= chr "2"
##  $ xlevels      : Named list()
##  - attr(*, "class")= chr [1:2] "nnet.formula" "nnet"
```

```r
nnet_predictions <-predict(nnet_fit,test_a[,2:6], type='class')
confusionMatrix(as.factor(nnet_predictions),as.factor(test_a$sex))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  582  375
##          1  894 2069
##
##                Accuracy : 0.6763
##                  95% CI : (0.6614, 0.6909)
##     No Information Rate : 0.6235
##     P-Value [Acc > NIR] : 3.052e-12
##
##                   Kappa : 0.2589
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3943
##             Specificity : 0.8466
##          Pos Pred Value : 0.6082
##          Neg Pred Value : 0.6983
##              Prevalence : 0.3765
##          Detection Rate : 0.1485
##    Detection Prevalence : 0.2441
##       Balanced Accuracy : 0.6204
##
##        'Positive' Class : 0
##
```

```r
# 0.6693 accuracy
```

Flexible Discriminant Analysis

```r
fda_fit <- fda(as.factor(sex)~., data=train_a)
summary(fda_fit)
```

```
##                   Length Class   Mode
## percent.explained 1      -none-  numeric
## values            1      -none-  numeric
## means             2      -none-  numeric
## theta.mod         1      -none-  numeric
## dimension         1      -none-  numeric
## prior             2      table   numeric
## fit               4      polyreg list
## call              3      -none-  call
## terms             3      terms   call
## confusion         4      table   numeric
```

```r
str(fda_fit)
```

```
## List of 10
##  $ percent.explained: Named num 100
##   ..- attr(*, "names")= chr "v1"
##  $ values           : Named num 0.101
##   ..- attr(*, "names")= chr "v1"
```

```
##  $ means            : num [1:2, 1] -0.416 0.271
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "0" "1"
##   .. ..$ : chr "v1"
##   ..- attr(*, "scaled:scale")= num 2.98
##  $ theta.mod        : num [1, 1] 1
##  $ dimension        : num 1
##  $ prior            : 'table' num [1:2(1d)] 0.394 0.606
##   ..- attr(*, "dimnames")=List of 1
##   .. ..$ fg: chr [1:2] "0" "1"
##  $ fit              :List of 4
##   ..$ coefficients: num [1:6, 1] -0.673 0.147 0.712 0.834 0.151 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:6] "Intercept" "o_ps" "c_ps" "e_ps" ...
##   .. .. ..$ : NULL
##   ..$ degree      : num 1
##   ..$ monomial    : logi FALSE
##   ..$ df          : int 6
##   ..- attr(*, "class")= chr "polyreg"
##  $ call             : language fda(formula = as.factor(sex) ~ ., data = train_a)
##  $ terms            :Classes 'terms', 'formula'  language as.factor(sex) ~ o_ps + c_ps + e_ps + a_ps
##   .. ..- attr(*, "variables")= language list(as.factor(sex), o_ps, c_ps, e_ps, a_ps, n_ps)
##   .. ..- attr(*, "factors")= int [1:6, 1:5] 0 1 0 0 0 0 0 0 1 0 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:6] "as.factor(sex)" "o_ps" "c_ps" "e_ps" ...
##   .. .. .. ..$ : chr [1:5] "o_ps" "c_ps" "e_ps" "a_ps" ...
##   .. ..- attr(*, "term.labels")= chr [1:5] "o_ps" "c_ps" "e_ps" "a_ps" ...
##   .. ..- attr(*, "order")= int [1:5] 1 1 1 1 1
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(as.factor(sex), o_ps, c_ps, e_ps, a_ps, n_ps)
##   .. ..- attr(*, "dataClasses")= Named chr [1:6] "factor" "numeric" "numeric" "numeric" ...
##   .. .. ..- attr(*, "names")= chr [1:6] "as.factor(sex)" "o_ps" "c_ps" "e_ps" ...
##  $ confusion        : 'table' int [1:2, 1:2] 1918 3102 1229 6486
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ predicted: chr [1:2] "0" "1"
##   .. ..$ true     : chr [1:2] "0" "1"
##   ..- attr(*, "error")= num 0.34
##  - attr(*, "class")= chr "fda"
```

```r
fda_predictions <-predict(fda_fit,test_a[,2:6])
confusionMatrix(as.factor(fda_predictions),as.factor(test_a$sex))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  579  386
##          1  897 2058
##
##                Accuracy : 0.6727
##                  95% CI : (0.6578, 0.6874)
##     No Information Rate : 0.6235
##     P-Value [Acc > NIR] : 7.479e-11
```

```
##
##                   Kappa : 0.2516
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3923
##             Specificity : 0.8421
##          Pos Pred Value : 0.6000
##          Neg Pred Value : 0.6964
##              Prevalence : 0.3765
##          Detection Rate : 0.1477
##    Detection Prevalence : 0.2462
##       Balanced Accuracy : 0.6172
##
##        'Positive' Class : 0
##
```

```
# 0.666 accuracy
```

Support Vector Machine

```
svm_fit <- ksvm(as.factor(sex)~., data=train_a)
svm_predictions <- predict(svm_fit, test_a[,2:6], type='response')
table(svm_predictions, test_a$sex)
```

```
##
## svm_predictions    0    1
##               0  470  279
##               1 1006 2165
```

```
confusionMatrix(as.factor(svm_predictions),as.factor(test_a$sex))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  470  279
##          1 1006 2165
##
##                Accuracy : 0.6722
##                  95% CI : (0.6572, 0.6869)
##     No Information Rate : 0.6235
##     P-Value [Acc > NIR] : 1.16e-10
##
##                   Kappa : 0.2263
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3184
##             Specificity : 0.8858
##          Pos Pred Value : 0.6275
##          Neg Pred Value : 0.6827
##              Prevalence : 0.3765
##          Detection Rate : 0.1199
##    Detection Prevalence : 0.1911
##       Balanced Accuracy : 0.6021
##
##        'Positive' Class : 0
```

```
##
```

```
# 0.6721 accuracy
```

k-Nearest Neighbours

```
knn_fit <- knn3(as.factor(sex)~., data=train_a, k = 10)
knn_predictions <- predict(knn_fit, test_a[,2:6], type='class')
table(knn_predictions, test_a$sex)
```

```
##
## knn_predictions    0    1
##               0  602  580
##               1  874 1864
```

```
confusionMatrix(as.factor(knn_predictions),as.factor(test_a$sex))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  602  580
##          1  874 1864
##
##                Accuracy : 0.6291
##                  95% CI : (0.6137, 0.6442)
##     No Information Rate : 0.6235
##     P-Value [Acc > NIR] : 0.2395
##
##                   Kappa : 0.1775
##  Mcnemar's Test P-Value : 1.542e-14
##
##             Sensitivity : 0.4079
##             Specificity : 0.7627
##          Pos Pred Value : 0.5093
##          Neg Pred Value : 0.6808
##              Prevalence : 0.3765
##          Detection Rate : 0.1536
##    Detection Prevalence : 0.3015
##       Balanced Accuracy : 0.5853
##
##        'Positive' Class : 0
##
```

```
# 0.6295 accuracy
```

Naive Bayes

Neural Network Model using 'keras' package

```
# Model initialization
# epochs <- 15
# batch_size <- 15
# initializer_random_normal() # Initial weighting initialization
#
# model <- keras_model_sequential()
# model %>%
#   layer_dense(units = 5,
```

```
#               input_shape = c(5),
#               name = "Input_Layer") %>%
#   layer_activation(activation = 'sigmoid') %>%3
#   layer_dense(units = 7,
#               name = "Dense_2",
#               kernel_regularizer = ) %>%
#   layer_activation(activation = 'relu') %>%
#   layer_dense(units = 7) %>%
#   layer_activation(activation = 'sigmoid')
#
# model %>% compile(
#   loss = loss_binary_crossentropy,
#   optimizer = optimizer_rmsprop(lr = 0.01), # Modified learning rate, being checked in logarithmic st
#   metrics = metric_binary_accuracy
# )
#
# # validation_model <- model
# # validation_model %>% fit(
# #   x = train_x_subset, y = train_y_subset,
# #   epochs = epochs,
# #   batch_size = batch_size,
# #   verbose = 2
# # )
# #summary(validation_model)
#
# model %>% fit(
#   x = train_x, y = train_y,
#   epochs = epochs,
#   batch_size = batch_size,
#   verbose = 2,
#   validation_split = 0.1
# )
```

```
# weightHistory <- R6::R6Class("weightHistory",
#   inherit = KerasCallback,
#   public = list(
#     weights = NULL,
#     on_batch_end = function(batch, logs = list()) {
#       for(i in 1:3){
#         var(self$weights[[i]],1)
#       }
#     }
# ))
```

```
# validation_model %>% evaluate(test_x, test_y, verbose = 0)
# model %>% evaluate(test_x, test_y, verbose = 0)
```

References: https://engineering.semantics3.com/debugging-neural-networks-a-checklist-ca52e11151ec https://keras.rstudio.com/articles/functional_api.html#multi-input-and-multi-output-models https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c https://machinelearningmastery.com/non-linear-classification-in-r/