R Notebook

import libraries

```
library(class)
library(chemometrics)
```

```
## Loading required package: rpart
```

```
library(boot)
library(tree)
#library(tidyverse)
library(vcd)
```

```
## Loading required package: grid
```

import data and filter

```
data = read.csv("healthcare-dataset-stroke-data.csv")
stroke = data.frame(data)
stroke$gender[stroke$gender=='Male'] = 1
stroke$gender[stroke$gender=='Female'] = 0
stroke$ever_married[stroke$ever_married=='Yes'] = 1
stroke$ever_married[stroke$ever_married=='No'] = 0
stroke$Residence_type[stroke$Residence_type=='Urban'] = 1
stroke$Residence_type[stroke$Residence_type=='Rural'] = 0
stroke = stroke[-which(stroke$gender=='Other'),]
stroke = stroke[-which(stroke$bmi=='N/A'),]
stroke$bmi = as.numeric(stroke$bmi)
```

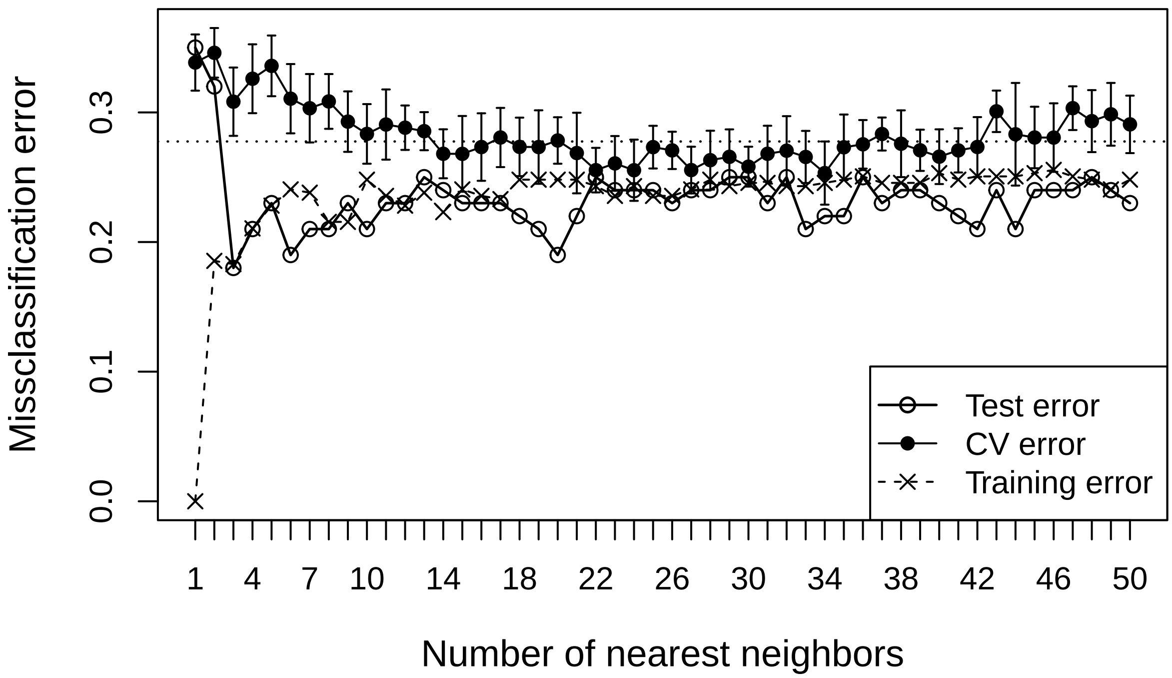
KNN model to predict stroke

Requires: data, libraries TODO: randomly choose nonstroke people and loop over several times choosing different people each time

```
stroke.knn = stroke[-c(500:nrow(stroke)),]
# convert variables to numeric
stroke.knn$gender = as.numeric(stroke.knn$gender)
stroke.knn$hypertension = as.numeric(stroke.knn$hypertension)
stroke.knn$heart_disease = as.numeric(stroke.knn$heart_disease)
```

```
stroke.knn$ever_married = as.numeric(stroke.knn$ever_married)
stroke.knn$Residence_type = as.numeric(stroke.knn$Residence_type)
# create training and testing datasets
train.size = round(nrow(stroke.knn)*0.8)
train.ind = sample(1:nrow(stroke.knn), train.size)
training = stroke.knn[train.ind,]
testing = stroke.knn[-train.ind,]

knnEval(scale(stroke.knn[,c(2,3,4,5,6,9,10)]), as.factor(stroke.knn$stroke), train.ind, kfold=10, knnvec
```




```
## $trainerr
##  [1] 0.0000000 0.1854637 0.1829574 0.2105263 0.2280702 0.2406015 0.2380952
##  [8] 0.2155388 0.2155388 0.2481203 0.2355890 0.2280702 0.2380952 0.2230576
## [15] 0.2406015 0.2355890 0.2330827 0.2481203 0.2481203 0.2481203 0.2481203
## [22] 0.2431078 0.2355890 0.2431078 0.2355890 0.2355890 0.2406015 0.2481203
## [29] 0.2431078 0.2456140 0.2456140 0.2431078 0.2431078 0.2456140 0.2481203
## [36] 0.2506266 0.2456140 0.2456140 0.2456140 0.2531328 0.2481203 0.2506266
## [43] 0.2506266 0.2506266 0.2531328 0.2556391 0.2506266 0.2481203 0.2406015
## [50] 0.2481203
##
## $testerr
##  [1] 0.35 0.32 0.18 0.21 0.23 0.19 0.21 0.21 0.23 0.21 0.23 0.23 0.25 0.24 0.23
## [16] 0.23 0.23 0.22 0.21 0.19 0.22 0.25 0.24 0.24 0.24 0.23 0.24 0.24 0.25 0.25
## [31] 0.23 0.25 0.21 0.22 0.22 0.25 0.23 0.24 0.24 0.23 0.22 0.21 0.24 0.21 0.24
## [46] 0.24 0.24 0.25 0.24 0.23
```

```
##
## $cvMean
##  [1] 0.3384615 0.3459615 0.3082692 0.3259615 0.3358974 0.3105769 0.3032051
##  [8] 0.3084615 0.2928846 0.2833974 0.2906410 0.2882051 0.2855128 0.2680769
## [15] 0.2680128 0.2732692 0.2806410 0.2733974 0.2732692 0.2783333 0.2686538
## [22] 0.2554487 0.2606410 0.2553846 0.2732051 0.2707051 0.2555128 0.2631410
## [29] 0.2657051 0.2581410 0.2680769 0.2704487 0.2656410 0.2532051 0.2730769
## [36] 0.2753846 0.2833333 0.2758333 0.2707692 0.2658333 0.2707051 0.2733333
## [43] 0.3008333 0.2831410 0.2806410 0.2805769 0.3032692 0.2932692 0.2985256
## [50] 0.2907692
##
## $cvSe
##  [1] 0.02169661 0.01912245 0.02635239 0.02657165 0.02341455 0.02672892
##  [7] 0.02639853 0.02113286 0.02332849 0.02299674 0.02708508 0.01713702
## [13] 0.01468252 0.01888623 0.02922971 0.02604245 0.02282843 0.02255266
## [19] 0.02834095 0.01792474 0.03107229 0.01717562 0.02112365 0.02361864
## [25] 0.01646292 0.01438180 0.01801529 0.02271544 0.02119638 0.01536711
## [31] 0.02162870 0.02664907 0.02009551 0.02436403 0.02528600 0.01872366
## [37] 0.01272938 0.02574357 0.01590415 0.02109831 0.01703436 0.02303379
## [43] 0.01597886 0.03957171 0.02372349 0.02643553 0.01685800 0.02391679
## [49] 0.02421197 0.02216281
##
## $cverr
##            [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
##  [1,] 0.4250000 0.3000000 0.4500000 0.4250000 0.3000000 0.2750000 0.4500000
##  [2,] 0.3000000 0.4750000 0.3500000 0.3000000 0.4000000 0.4250000 0.2500000
##  [3,] 0.2000000 0.2750000 0.1250000 0.3250000 0.4000000 0.3750000 0.3250000
##  [4,] 0.4000000 0.3750000 0.3250000 0.2500000 0.3500000 0.3000000 0.3500000
##  [5,] 0.3500000 0.3500000 0.3000000 0.3750000 0.4500000 0.4000000 0.2000000
##  [6,] 0.3000000 0.3250000 0.3750000 0.2750000 0.3250000 0.2750000 0.1750000
##  [7,] 0.3750000 0.3750000 0.2750000 0.3000000 0.3250000 0.2500000 0.3500000
##  [8,] 0.2750000 0.3250000 0.2750000 0.1750000 0.2500000 0.4000000 0.2750000
##  [9,] 0.3750000 0.2750000 0.3000000 0.4500000 0.2000000 0.1750000 0.3750000
## [10,] 0.3846154 0.3846154 0.3076923 0.3846154 0.3589744 0.2307692 0.2820513
##            [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
##  [1,] 0.3250000 0.3000000 0.3500000 0.3500000 0.2750000 0.2750000 0.2250000
##  [2,] 0.3750000 0.3250000 0.2500000 0.2750000 0.2000000 0.3000000 0.3250000
##  [3,] 0.3500000 0.2250000 0.2250000 0.3750000 0.3250000 0.3250000 0.2250000
##  [4,] 0.2250000 0.2750000 0.2500000 0.2250000 0.2000000 0.3500000 0.2500000
##  [5,] 0.3000000 0.4000000 0.2250000 0.2000000 0.3000000 0.3500000 0.3000000
##  [6,] 0.2750000 0.3500000 0.4250000 0.2000000 0.3250000 0.2500000 0.3500000
##  [7,] 0.3250000 0.3750000 0.2500000 0.2250000 0.3750000 0.2750000 0.1750000
##  [8,] 0.3500000 0.2500000 0.2000000 0.4500000 0.3000000 0.2500000 0.3500000
##  [9,] 0.1750000 0.2750000 0.3000000 0.3500000 0.3000000 0.2750000 0.2500000
## [10,] 0.3846154 0.1538462 0.3589744 0.2564103 0.2820513 0.2051282 0.2307692
##           [,15]     [,16]     [,17]     [,18]     [,19]     [,20]     [,21]
##  [1,] 0.3500000 0.2500000 0.3250000 0.3250000 0.2500000 0.2750000 0.2000000
##  [2,] 0.2500000 0.2750000 0.3000000 0.3750000 0.1500000 0.3000000 0.3250000
##  [3,] 0.1500000 0.2000000 0.2000000 0.3000000 0.3250000 0.3000000 0.1250000
##  [4,] 0.3250000 0.2250000 0.1750000 0.2000000 0.2250000 0.2500000 0.3000000
##  [5,] 0.3250000 0.4250000 0.2250000 0.2500000 0.1500000 0.3000000 0.3250000
##  [6,] 0.4250000 0.3000000 0.3250000 0.2500000 0.2500000 0.2250000 0.2750000
##  [7,] 0.1250000 0.3500000 0.2750000 0.1500000 0.3250000 0.3750000 0.2250000
##  [8,] 0.2750000 0.1250000 0.4250000 0.2250000 0.3000000 0.2500000 0.3000000
```

```
##  [9,] 0.2500000 0.2750000 0.3000000 0.3000000 0.4500000 0.1750000 0.1500000
## [10,] 0.2051282 0.3076923 0.2564103 0.3589744 0.3076923 0.3333333 0.4615385
##            [,22]     [,23]     [,24]     [,25]     [,26]     [,27]     [,28]
##  [1,] 0.2750000 0.2250000 0.1750000 0.3500000 0.3000000 0.3000000 0.1750000
##  [2,] 0.2250000 0.2250000 0.3250000 0.3000000 0.2500000 0.3250000 0.2250000
##  [3,] 0.2250000 0.3750000 0.3250000 0.2000000 0.2000000 0.2750000 0.2250000
##  [4,] 0.2750000 0.2250000 0.3250000 0.3500000 0.2250000 0.3250000 0.2500000
##  [5,] 0.2750000 0.1750000 0.2500000 0.2750000 0.2250000 0.2750000 0.4250000
##  [6,] 0.2750000 0.2500000 0.3250000 0.2500000 0.2750000 0.1750000 0.3250000
##  [7,] 0.2000000 0.2000000 0.3000000 0.2500000 0.3000000 0.2750000 0.3000000
##  [8,] 0.3750000 0.3250000 0.1500000 0.2000000 0.3500000 0.1750000 0.2000000
##  [9,] 0.2500000 0.3500000 0.2250000 0.2750000 0.3000000 0.2250000 0.2500000
## [10,] 0.1794872 0.2564103 0.1538462 0.2820513 0.2820513 0.2051282 0.2564103
##            [,29]     [,30]     [,31]     [,32]     [,33]     [,34]     [,35]
##  [1,] 0.1500000 0.3250000 0.1750000 0.3750000 0.1750000 0.1750000 0.3500000
##  [2,] 0.3000000 0.2250000 0.2500000 0.2500000 0.1750000 0.2000000 0.3750000
##  [3,] 0.1750000 0.3250000 0.3250000 0.3000000 0.3750000 0.2000000 0.1500000
##  [4,] 0.2250000 0.2250000 0.2500000 0.4250000 0.2500000 0.4500000 0.2250000
##  [5,] 0.2750000 0.2000000 0.3000000 0.3000000 0.2500000 0.2500000 0.3250000
##  [6,] 0.3500000 0.2250000 0.1750000 0.2750000 0.2750000 0.2250000 0.3000000
##  [7,] 0.3000000 0.3250000 0.4000000 0.1750000 0.2750000 0.2250000 0.3000000
##  [8,] 0.3500000 0.2250000 0.3000000 0.1750000 0.3500000 0.2500000 0.3250000
##  [9,] 0.2500000 0.2500000 0.2750000 0.2500000 0.2750000 0.2750000 0.1500000
## [10,] 0.2820513 0.2564103 0.2307692 0.1794872 0.2564103 0.2820513 0.2307692
##            [,36]     [,37]     [,38]     [,39]     [,40]     [,41]     [,42]
##  [1,] 0.3250000 0.2500000 0.2500000 0.2500000 0.1500000 0.2000000 0.2250000
##  [2,] 0.2750000 0.3000000 0.2000000 0.2500000 0.3250000 0.3000000 0.2250000
##  [3,] 0.2750000 0.3000000 0.2500000 0.4000000 0.1500000 0.2750000 0.3000000
##  [4,] 0.3500000 0.2500000 0.2250000 0.2500000 0.2500000 0.3000000 0.1750000
##  [5,] 0.2000000 0.2750000 0.4000000 0.2500000 0.3000000 0.3000000 0.2500000
##  [6,] 0.3000000 0.3500000 0.1750000 0.2500000 0.2750000 0.3500000 0.3750000
##  [7,] 0.3250000 0.2500000 0.3500000 0.2250000 0.3250000 0.2250000 0.3500000
##  [8,] 0.2750000 0.3000000 0.3750000 0.2500000 0.2750000 0.3000000 0.3250000
##  [9,] 0.2750000 0.2250000 0.2000000 0.2750000 0.2750000 0.1750000 0.1750000
## [10,] 0.1538462 0.3333333 0.3333333 0.3076923 0.3333333 0.2820513 0.3333333
##            [,43]     [,44]     [,45]     [,46]     [,47]     [,48]     [,49]
##  [1,] 0.3250000 0.5000000 0.2500000 0.2000000 0.2500000 0.4250000 0.4250000
##  [2,] 0.2750000 0.4750000 0.3500000 0.4000000 0.3250000 0.1750000 0.2750000
##  [3,] 0.3750000 0.2500000 0.4250000 0.3000000 0.3250000 0.2500000 0.3000000
##  [4,] 0.2500000 0.3000000 0.1500000 0.2000000 0.3750000 0.3500000 0.2250000
##  [5,] 0.3750000 0.3000000 0.2750000 0.3000000 0.3750000 0.3000000 0.1750000
##  [6,] 0.2750000 0.1500000 0.3250000 0.2250000 0.3000000 0.3500000 0.3000000
##  [7,] 0.2250000 0.3000000 0.3000000 0.4250000 0.3000000 0.2000000 0.2500000
##  [8,] 0.2750000 0.1250000 0.2250000 0.3250000 0.2000000 0.3250000 0.3000000
##  [9,] 0.3000000 0.1750000 0.2500000 0.2000000 0.2750000 0.2500000 0.3250000
## [10,] 0.3333333 0.2564103 0.2564103 0.2307692 0.3076923 0.3076923 0.4102564
##            [,50]
##  [1,] 0.4000000
##  [2,] 0.2500000
##  [3,] 0.2000000
##  [4,] 0.3500000
##  [5,] 0.3500000
##  [6,] 0.2750000
##  [7,] 0.2750000
```

```
##  [8,] 0.1750000
##  [9,] 0.3250000
## [10,] 0.3076923
##
## $knnvec
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
```

```
# predict using KNN
predictions = knn(scale(training[,c(2,3,4,5,6,9,10)]), scale(testing[,c(2,3,4,5,6,9,10)]), training$stro
#predictions
#testing$stroke

# compute accuracy
accuracy = sum(predictions==testing$stroke)/length(predictions)
cat("Accuracy:\n")
```

```
## Accuracy:
```

```
accuracy
```

```
## [1] 0.81
```

```
#which(predictions==1)
#length(which(predictions==1))
cat("Predictions:\n")
```

```
## Predictions:
```

```
predictions
```

```
##   [1] 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 1 1 1 1 0
##  [38] 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
## Levels: 0 1
```

```
#which(testing$stroke==1)
#length(which(testing$stroke==1))
cat("Actual values:\n")
```

```
## Actual values:
```

```
testing$stroke
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

linear model to predict age

Requires: data, libraries

```
stroke.lm = stroke
inds = sample(1:nrow(stroke.lm), round(nrow(stroke.lm)/2))
train = stroke.lm[inds,]
test = stroke.lm[-inds,]
super.model = lm(age~gender+hypertension+heart_disease+ever_married+work_type+Residence_type+avg_glucos
basic.model = lm(age~hypertension+heart_disease+ever_married+work_type+avg_glucose_level+smoking_status+
summary(super.model)
```

```
##
## Call:
## lm(formula = age ~ gender + hypertension + heart_disease + ever_married +
##     work_type + Residence_type + avg_glucose_level + smoking_status +
##     stroke + bmi, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.179  -8.987  -1.193   7.558  52.354
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                6.77769    1.50802   4.494 7.30e-06 ***
## gender1                    0.04328    0.54567   0.079 0.936781
## hypertension               7.19572    0.96076   7.490 9.60e-14 ***
## heart_disease             14.31184    1.28469  11.140  < 2e-16 ***
## ever_married1             17.92184    0.68791  26.053  < 2e-16 ***
## work_typeGovt_job         28.04724    1.30933  21.421  < 2e-16 ***
## work_typeNever_worked     10.75240    4.06684   2.644 0.008248 **
## work_typePrivate          23.77117    1.07855  22.040  < 2e-16 ***
## work_typeSelf-employed    34.73576    1.26454  27.469  < 2e-16 ***
## Residence_type1            0.24752    0.53184   0.465 0.641686
## avg_glucose_level          0.04201    0.00628   6.690 2.76e-11 ***
## smoking_statusnever smoked -3.87843    0.78629  -4.933 8.66e-07 ***
## smoking_statussmokes      -5.36596    0.94647  -5.669 1.60e-08 ***
## smoking_statusUnknown     -3.47203    0.89242  -3.891 0.000103 ***
## stroke                    11.69058    1.37304   8.514  < 2e-16 ***
## bmi                       -0.05110    0.03857  -1.325 0.185350
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.14 on 2438 degrees of freedom
## Multiple R-squared:  0.6603, Adjusted R-squared:  0.6582
## F-statistic: 315.9 on 15 and 2438 DF,  p-value: < 2.2e-16
```

```
summary(basic.model)
```

```
##
## Call:
## lm(formula = age ~ hypertension + heart_disease + ever_married +
```

```
##     work_type + avg_glucose_level + smoking_status + stroke,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.135  -9.095  -1.114   7.509  52.563
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 6.009304   1.270830   4.729 2.39e-06 ***
## hypertension                7.043014   0.953206   7.389 2.02e-13 ***
## heart_disease              14.413647   1.280337  11.258  < 2e-16 ***
## ever_married1              17.797128   0.681510  26.114  < 2e-16 ***
## work_typeGovt_job          27.624843   1.266859  21.806  < 2e-16 ***
## work_typeNever_worked      10.432735   4.055357   2.573 0.010153 *
## work_typePrivate           23.350706   1.028712  22.699  < 2e-16 ***
## work_typeSelf-employed     34.365231   1.229035  27.961  < 2e-16 ***
## avg_glucose_level           0.040869   0.006221   6.570 6.13e-11 ***
## smoking_statusnever smoked -3.886066   0.784669  -4.952 7.83e-07 ***
## smoking_statussmokes       -5.389728   0.946051  -5.697 1.37e-08 ***
## smoking_statusUnknown      -3.470656   0.891310  -3.894 0.000101 ***
## stroke                     11.756576   1.371927   8.569  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.14 on 2441 degrees of freedom
## Multiple R-squared:   0.66,  Adjusted R-squared:  0.6583
## F-statistic: 394.9 on 12 and 2441 DF,  p-value: < 2.2e-16
```

```r
# How good are these models on the training data
cat("supermodel error on training data: \n")
```

```
## supermodel error on training data:
```

```r
mean((predict(super.model)-train$age)^2)
```

```
## [1] 171.6135
```

```r
cat("basicmodel error on training data: \n")
```

```
## basicmodel error on training data:
```

```r
mean((predict(basic.model)-train$age)^2)
```

```
## [1] 171.7516
```

```r
# How good are these models on the testing data
cat("supermodel error on testing data: \n")
```

```
## supermodel error on testing data:
```

```r
mean((predict(super.model, newdata=test)-test$age)^2)
```

```
## [1] 172.5648
```

```r
cat("basicmodel error on testing data: \n")
```

```
## basicmodel error on testing data:
```

```r
mean((predict(basic.model, newdata=test)-test$age)^2)
```

```
## [1] 172.7688
```

```r
# K-fold cross validation for the age linear model
basic.model = glm(age~hypertension+heart_disease+ever_married+work_type+avg_glucose_level+smoking_statu
model.cv = cv.glm(data=stroke.lm, basic.model, K=5)
cat("K-fold cross validation linear model error:\n")
```

```
## K-fold cross validation linear model error:
```

```r
model.cv$delta[2] # tells us error
```

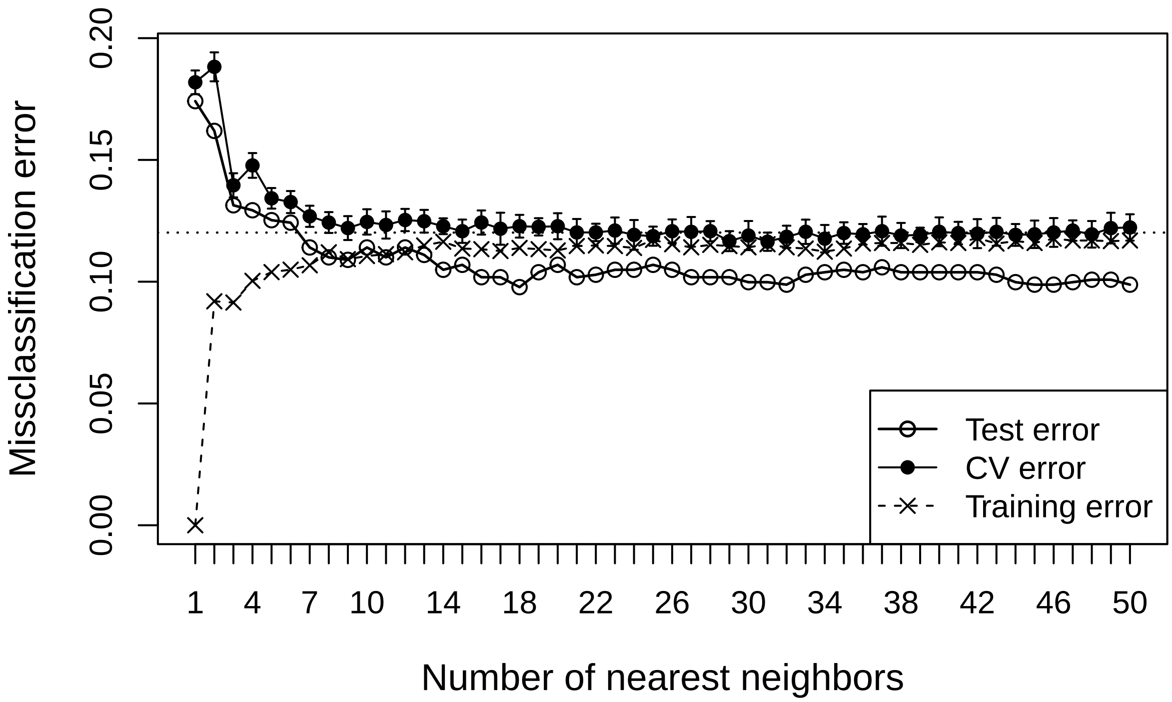
```
## [1] 172.6028
```

KNN model to predict if ever_married

Requires: data, libraries

```r
stroke.knn = stroke
# convert variables to numeric
stroke.knn$gender = as.numeric(stroke.knn$gender)
stroke.knn$hypertension = as.numeric(stroke.knn$hypertension)
stroke.knn$heart_disease = as.numeric(stroke.knn$heart_disease)
stroke.knn$Residence_type = as.numeric(stroke.knn$Residence_type)
# create training and testing datasets
train.size = round(nrow(stroke.knn)*0.8)
train.ind = sample(1:nrow(stroke.knn), train.size)
training = stroke.knn[train.ind,]
testing = stroke.knn[-train.ind,]

knnEval(stroke.knn[,c(2,3,4,5,9,10)], as.factor(stroke.knn$ever_married), train.ind, kfold=10, knnvec=se
```




```
## $trainerr
##  [1] 0.00000000 0.09195110 0.09144167 0.10035660 0.10392257 0.10494142
##  [7] 0.10672440 0.11207336 0.10927152 0.11054508 0.11130922 0.11207336
## [13] 0.11487519 0.11640346 0.11360163 0.11334692 0.11258278 0.11385634
## [19] 0.11334692 0.11283749 0.11462048 0.11487519 0.11462048 0.11385634
## [25] 0.11716760 0.11538462 0.11411105 0.11512990 0.11462048 0.11411105
## [31] 0.11538462 0.11411105 0.11360163 0.11232807 0.11360163 0.11563933
## [37] 0.11589404 0.11589404 0.11512990 0.11614875 0.11589404 0.11793174
## [43] 0.11563933 0.11665818 0.11589404 0.11767702 0.11691289 0.11691289
## [49] 0.11665818 0.11691289
##
## $testerr
##  [1] 0.17413442 0.16191446 0.13136456 0.12932790 0.12525458 0.12423625
##  [7] 0.11405295 0.10997963 0.10896130 0.11405295 0.10997963 0.11405295
## [13] 0.11099796 0.10488798 0.10692464 0.10183299 0.10183299 0.09775967
## [19] 0.10386965 0.10692464 0.10183299 0.10285132 0.10488798 0.10488798
## [25] 0.10692464 0.10488798 0.10183299 0.10183299 0.10183299 0.09979633
## [31] 0.09979633 0.09877800 0.10285132 0.10386965 0.10488798 0.10386965
## [37] 0.10590631 0.10386965 0.10386965 0.10386965 0.10386965 0.10386965
## [43] 0.10285132 0.09979633 0.09877800 0.09877800 0.09979633 0.10081466
## [49] 0.10081466 0.09877800
##
## $cvMean
##  [1] 0.1818702 0.1882231 0.1395778 0.1477326 0.1342330 0.1327017 0.1268506
##  [8] 0.1242899 0.1220102 0.1245437 0.1232824 0.1253226 0.1248144 0.1227781
## [15] 0.1207243 0.1242944 0.1217401 0.1227722 0.1225223 0.1227696 0.1202212
```

```
## [22] 0.1202323 0.1209969 0.1192190 0.1186887 0.1207275 0.1204711 0.1207353
## [29] 0.1166491 0.1189438 0.1164109 0.1184362 0.1204887 0.1176689 0.1199687
## [36] 0.1194656 0.1207334 0.1189619 0.1192028 0.1204880 0.1199674 0.1197285
## [43] 0.1204783 0.1192015 0.1194598 0.1202141 0.1209904 0.1194579 0.1220004
## [50] 0.1222731
##
## $cvSe
##  [1] 0.004864322 0.005943257 0.004947930 0.005079436 0.004224539 0.004520198
##  [7] 0.004338331 0.004278875 0.004906240 0.005200589 0.005584700 0.004563347
## [13] 0.004666390 0.003245513 0.004781781 0.004960563 0.006568822 0.004666473
## [19] 0.003550299 0.005348052 0.005522980 0.003568970 0.005379523 0.006102818
## [25] 0.003927247 0.004846369 0.006097380 0.004135846 0.004110703 0.005976378
## [31] 0.003749517 0.004548617 0.004992377 0.005623786 0.004416926 0.004216290
## [37] 0.005987887 0.005165969 0.002971251 0.005912910 0.004605687 0.005948859
## [43] 0.005722555 0.004477491 0.005638746 0.005904094 0.004187700 0.005449852
## [49] 0.006296600 0.005412753
##
## $cverr
##            [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
##  [1,] 0.1857506 0.1832061 0.1577608 0.1526718 0.1552163 0.1577608 0.1170483
##  [2,] 0.1653944 0.1832061 0.1526718 0.1221374 0.1195929 0.1272265 0.1221374
##  [3,] 0.1832061 0.2086514 0.1221374 0.1526718 0.1501272 0.1119593 0.1297710
##  [4,] 0.1832061 0.2111959 0.1526718 0.1603053 0.1221374 0.1424936 0.1145038
##  [5,] 0.1781170 0.2010178 0.1221374 0.1374046 0.1170483 0.1348601 0.1119593
##  [6,] 0.1730280 0.1781170 0.1475827 0.1628499 0.1424936 0.1348601 0.1501272
##  [7,] 0.1632653 0.1989796 0.1403061 0.1632653 0.1454082 0.1505102 0.1505102
##  [8,] 0.1760204 0.1760204 0.1556122 0.1581633 0.1326531 0.1198980 0.1173469
##  [9,] 0.2168367 0.1938776 0.1198980 0.1198980 0.1275510 0.1224490 0.1275510
## [10,] 0.1938776 0.1479592 0.1250000 0.1479592 0.1301020 0.1250000 0.1275510
##             [,8]      [,9]      [,10]      [,11]      [,12]     [,13]     [,14]
##  [1,] 0.11704835 0.1246819 0.12213740 0.11450382 0.14249364 0.1297710 0.1195929
##  [2,] 0.12977099 0.1170483 0.13740458 0.09160305 0.12977099 0.1043257 0.1043257
##  [3,] 0.12213740 0.1475827 0.12977099 0.12468193 0.13740458 0.1145038 0.1170483
##  [4,] 0.12977099 0.1246819 0.13231552 0.14503817 0.11704835 0.1348601 0.1119593
##  [5,] 0.13740458 0.1017812 0.15012723 0.14503817 0.11195929 0.1374046 0.1323155
##  [6,] 0.14758270 0.1043257 0.11704835 0.11195929 0.09669211 0.1068702 0.1246819
##  [7,] 0.12500000 0.1198980 0.09693878 0.10714286 0.14030612 0.1505102 0.1403061
##  [8,] 0.11734694 0.1147959 0.10204082 0.11989796 0.11734694 0.1147959 0.1224490
##  [9,] 0.09693878 0.1479592 0.13775510 0.13265306 0.12755102 0.1224490 0.1275510
## [10,] 0.11989796 0.1173469 0.11989796 0.14030612 0.13265306 0.1326531 0.1275510
##           [,15]     [,16]      [,17]     [,18]      [,19]     [,20]      [,21]
##  [1,] 0.1094148 0.1195929 0.15267176 0.1323155 0.11959288 0.1526718 0.11450382
##  [2,] 0.1246819 0.1628499 0.14249364 0.1170483 0.11704835 0.1017812 0.10687023
##  [3,] 0.1374046 0.1272265 0.13740458 0.1170483 0.09923664 0.1145038 0.13231552
##  [4,] 0.1399491 0.1195929 0.13231552 0.1017812 0.13994911 0.1195929 0.12468193
##  [5,] 0.1297710 0.1145038 0.12468193 0.1424936 0.11959288 0.1145038 0.13994911
##  [6,] 0.1195929 0.1221374 0.08905852 0.1221374 0.11704835 0.1399491 0.11450382
##  [7,] 0.1352041 0.1198980 0.12500000 0.1275510 0.12755102 0.0994898 0.09693878
##  [8,] 0.0994898 0.1173469 0.11479592 0.1198980 0.12755102 0.1377551 0.09438776
##  [9,] 0.0994898 0.1352041 0.09693878 0.1454082 0.12244898 0.1198980 0.14285714
## [10,] 0.1122449 0.1045918 0.10204082 0.1020408 0.13520408 0.1275510 0.13520408
##           [,22]      [,23]      [,24]      [,25]     [,26]      [,27]     [,28]
##  [1,] 0.1221374 0.12977099 0.12213740 0.11704835 0.1170483 0.10687023 0.1170483
##  [2,] 0.1068702 0.11704835 0.09669211 0.13231552 0.1221374 0.12977099 0.1170483
```

```
##  [3,] 0.1246819 0.08905852 0.12468193 0.13231552 0.1603053 0.14249364 0.1195929
##  [4,] 0.1195929 0.13740458 0.09669211 0.11704835 0.1170483 0.13740458 0.1043257
##  [5,] 0.1119593 0.09923664 0.09160305 0.11195929 0.1170483 0.10687023 0.1221374
##  [6,] 0.1043257 0.11959288 0.12977099 0.12977099 0.1145038 0.12977099 0.1374046
##  [7,] 0.1173469 0.12500000 0.14540816 0.11989796 0.1224490 0.12244898 0.1198980
##  [8,] 0.1428571 0.14795918 0.14540816 0.10969388 0.1071429 0.13010204 0.1071429
##  [9,] 0.1301020 0.12244898 0.12500000 0.09183673 0.1250000 0.12244898 0.1147959
## [10,] 0.1224490 0.12244898 0.11479592 0.12500000 0.1045918 0.07653061 0.1479592
##            [,29]      [,30]     [,31]      [,32]     [,33]      [,34]     [,35]
##  [1,] 0.13486005 0.12977099 0.1119593 0.11195929 0.1094148 0.09923664 0.1170483
##  [2,] 0.13231552 0.13740458 0.1170483 0.11704835 0.1043257 0.11450382 0.1170483
##  [3,] 0.11195929 0.12722646 0.1043257 0.10941476 0.1043257 0.12722646 0.1475827
##  [4,] 0.11195929 0.09160305 0.1195929 0.12722646 0.1348601 0.13994911 0.1221374
##  [5,] 0.11450382 0.10941476 0.1145038 0.13740458 0.1017812 0.14249364 0.1094148
##  [6,] 0.12977099 0.14503817 0.1017812 0.12722646 0.1297710 0.11450382 0.1094148
##  [7,] 0.09183673 0.12755102 0.1096939 0.11224490 0.1454082 0.08418367 0.1122449
##  [8,] 0.10969388 0.12755102 0.1275510 0.08673469 0.1352041 0.10969388 0.1428571
##  [9,] 0.11989796 0.10204082 0.1428571 0.13010204 0.1275510 0.12755102 0.1096939
## [10,] 0.10969388 0.09183673 0.1147959 0.12500000 0.1122449 0.11734694 0.1122449
##           [,36]      [,37]      [,38]     [,39]      [,40]      [,41]
##  [1,] 0.1094148 0.10178117 0.11195929 0.1246819 0.12468193 0.11959288
##  [2,] 0.1068702 0.11704835 0.10432570 0.1374046 0.13231552 0.09923664
##  [3,] 0.1323155 0.13231552 0.09669211 0.1145038 0.09414758 0.13486005
##  [4,] 0.1043257 0.08396947 0.13994911 0.1094148 0.09669211 0.11959288
##  [5,] 0.1017812 0.13994911 0.11195929 0.1297710 0.14249364 0.11704835
##  [6,] 0.1399491 0.15012723 0.10432570 0.1094148 0.09669211 0.13740458
##  [7,] 0.1173469 0.11479592 0.11224490 0.1122449 0.12755102 0.09948980
##  [8,] 0.1301020 0.12500000 0.13265306 0.1122449 0.13775510 0.14030612
##  [9,] 0.1275510 0.11479592 0.13775510 0.1198980 0.11479592 0.12244898
## [10,] 0.1250000 0.12755102 0.13775510 0.1224490 0.13775510 0.10969388
##            [,42]      [,43]      [,44]      [,45]      [,46]     [,47]
##  [1,] 0.11704835 0.11959288 0.10687023 0.11704835 0.12213740 0.1043257
##  [2,] 0.13486005 0.11195929 0.12213740 0.13231552 0.16284987 0.1145038
##  [3,] 0.12468193 0.10432570 0.11704835 0.09923664 0.12213740 0.1221374
##  [4,] 0.10432570 0.13994911 0.11450382 0.10941476 0.12722646 0.1424936
##  [5,] 0.08905852 0.10941476 0.15012723 0.14503817 0.12468193 0.1246819
##  [6,] 0.09414758 0.13994911 0.11959288 0.11450382 0.10178117 0.1094148
##  [7,] 0.14030612 0.15051020 0.12244898 0.10714286 0.12244898 0.1326531
##  [8,] 0.14030612 0.11989796 0.09438776 0.09693878 0.11224490 0.1326531
##  [9,] 0.11734694 0.11734694 0.11989796 0.12500000 0.09183673 0.1250000
## [10,] 0.13520408 0.09183673 0.12500000 0.14795918 0.11479592 0.1020408
##            [,48]      [,49]      [,50]
##  [1,] 0.10432570 0.13486005 0.11450382
##  [2,] 0.12213740 0.12977099 0.10687023
##  [3,] 0.11959288 0.13994911 0.12213740
##  [4,] 0.10687023 0.13486005 0.11195929
##  [5,] 0.11704835 0.08142494 0.09669211
##  [6,] 0.15521628 0.13740458 0.13740458
##  [7,] 0.14030612 0.11734694 0.12755102
##  [8,] 0.11224490 0.12755102 0.11989796
##  [9,] 0.11989796 0.12500000 0.15816327
## [10,] 0.09693878 0.09183673 0.12755102
##
## $knnvec
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
```

```r
# predict using KNN
predictions = knn(scale(training[,c(2,3,4,5,8,9,10)]), scale(testing[,c(2,3,4,5,8,9,10)]), training$eve
cat("Predictions:\n")
```

```
## Predictions:
```

```r
predictions
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 0 1 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1
##  [75] 0 0 0 1 1 1 1 1 1 0 1 1 0 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 0 1 1 1 1 0 1 0 0
## [112] 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 0 0 1 1 1 1 1 0 1 1 0 1 1 1 1
## [149] 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 0 0 1 1 1 1 0 1 1 1 1 1 1
## [186] 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 1 1 0 1 0 1 0 1 1 0 1 1 0 1 1 1 1 1
## [223] 1 1 1 1 0 1 1 1 1 1 1 0 1 1 0 1 1 0 1 0 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 1 0
## [260] 1 0 1 1 1 0 0 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1
## [297] 0 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 1 0 1 0
## [334] 0 1 0 1 0 1 1 1 1 1 0 1 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 1 0 1 1 1 1 0 1 1
## [371] 1 1 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 1 1 1 1
## [408] 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 0 1 1 1 0 0 1 0 1 1 0 0 1 0 1 1 1 1 1 1 1
## [445] 1 0 0 1 1 1 0 0 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1 0 0 1 0 1 1 1 1 1 0 1 1
## [482] 0 0 1 1 0 0 1 1 1 0 1 1 1 0 0 0 0 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0
## [519] 1 0 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1 1 0 1 1 1 1
## [556] 1 1 1 1 0 1 0 1 1 1 1 1 1 0 0 0 0 1 0 1 0 1 1 1 1 1 0 0 1 1 1 0 1 1 0 0 0
## [593] 1 1 1 0 1 1 1 0 1 1 1 1 0 0 1 0 1 1 1 1 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 0
## [630] 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 0 1 1 0 1 1 1 0 0 1
## [667] 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 1 1 1 0 1 1 1 1 1 0 1 1 1 0 0
## [704] 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 0 1 1 1 0 1 1 0 1 0 0 1 0 1 0 0 1 0 1 1 1
## [741] 1 1 0 0 1 0 1 1 1 1 0 1 0 0 1 1 0 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0
## [778] 0 1 1 1 0 0 1 1 0 1 0 0 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 0 1 1 0 1 1 1 1 0
## [815] 1 1 1 0 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 1 1 1 1 1 1 0 1 1 0 0 1 1 0 1 1
## [852] 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
## [889] 1 0 1 1 1 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1 0 0 0 0 1 1 1 0 1 0 0 0 1 1 0 0
## [926] 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0 1 1 0 1 1 0 1 0 1 1 1 0 0 1 1 1
## [963] 1 1 0 1 0 1 1 0 1 1 1 0 1 0 1 1 1 1 1 1
## Levels: 0 1
```

```r
cat("Actual values:\n")
```

```
## Actual values:
```

```r
testing$ever_married
```

```
##   [1] "1" "1" "1" "0" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1"
##  [19] "1" "1" "1" "0" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1"
##  [37] "1" "0" "1" "1" "0" "1" "1" "1" "0" "1" "1" "0" "1" "1" "1" "1" "0" "1"
##  [55] "1" "1" "1" "1" "1" "0" "1" "1" "1" "1" "1" "1" "1" "1" "1" "0" "1" "0"
##  [73] "0" "1" "0" "0" "0" "1" "1" "1" "1" "1" "1" "0" "1" "1" "0" "1" "1" "0"
```

```
##  [91] "1" "1" "0" "1" "0" "1" "1" "1" "1" "0" "0" "1" "1" "1" "1" "1" "1" "0"
## [109] "1" "0" "1" "1" "1" "1" "0" "0" "1" "1" "0" "1" "1" "1" "0" "0" "1" "1"
## [127] "1" "0" "1" "0" "1" "0" "1" "0" "0" "1" "1" "1" "0" "1" "0" "0" "1" "0"
## [145] "1" "1" "1" "0" "1" "1" "0" "1" "1" "1" "1" "1" "0" "1" "1" "1" "0" "1"
## [163] "1" "1" "1" "1" "0" "1" "1" "1" "0" "1" "0" "0" "1" "1" "1" "1" "1" "1"
## [181] "1" "1" "1" "1" "1" "1" "1" "0" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1"
## [199] "1" "0" "0" "1" "0" "0" "1" "1" "0" "1" "0" "1" "0" "1" "1" "0" "1" "1"
## [217] "1" "1" "1" "1" "0" "1" "1" "1" "1" "1" "0" "1" "1" "1" "1" "1" "1" "0"
## [235] "1" "1" "0" "1" "1" "0" "1" "1" "1" "1" "0" "1" "0" "0" "1" "1" "1" "0"
## [253] "1" "1" "0" "1" "1" "0" "0" "0" "0" "0" "1" "1" "0" "0" "1" "1" "1" "1"
## [271] "0" "1" "1" "1" "0" "0" "1" "0" "1" "1" "1" "1" "1" "1" "1" "0" "1" "1"
## [289] "1" "0" "1" "1" "0" "1" "1" "1" "0" "0" "1" "1" "1" "1" "1" "0" "1" "1"
## [307] "0" "1" "1" "1" "1" "1" "1" "0" "1" "1" "1" "0" "0" "0" "0" "0" "1" "1"
## [325] "1" "1" "1" "0" "0" "1" "0" "0" "0" "0" "1" "0" "0" "0" "1" "1" "1" "1"
## [343] "1" "0" "1" "0" "0" "0" "0" "1" "1" "1" "1" "0" "1" "1" "1" "1" "0" "0"
## [361] "0" "1" "0" "1" "1" "1" "1" "0" "0" "1" "1" "1" "1" "1" "1" "1" "1" "1"
## [379] "1" "0" "1" "1" "0" "0" "0" "0" "1" "0" "1" "1" "1" "1" "1" "1" "1" "0"
## [397] "1" "1" "0" "0" "0" "0" "0" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1"
## [415] "0" "0" "1" "0" "1" "1" "0" "0" "0" "0" "1" "1" "0" "0" "0" "1" "0" "1"
## [433] "1" "0" "0" "0" "0" "1" "1" "1" "1" "1" "1" "1" "1" "0" "0" "1" "1" "1"
## [451] "0" "0" "0" "1" "1" "0" "0" "1" "1" "1" "0" "1" "1" "1" "1" "1" "1" "0"
## [469] "1" "0" "0" "1" "0" "0" "1" "1" "1" "1" "0" "1" "1" "0" "0" "0" "1" "0"
## [487] "0" "1" "1" "1" "0" "1" "1" "1" "0" "0" "0" "0" "1" "0" "1" "0" "0" "1"
## [505] "1" "1" "1" "1" "1" "0" "1" "1" "1" "1" "1" "1" "1" "0" "1" "0" "1" "1"
## [523] "1" "1" "1" "0" "0" "1" "0" "1" "1" "1" "0" "1" "1" "1" "1" "1" "1" "1"
## [541] "0" "0" "1" "1" "0" "1" "0" "1" "1" "1" "0" "1" "1" "1" "1" "1" "1" "1"
## [559] "1" "0" "1" "0" "0" "0" "1" "1" "1" "1" "0" "0" "1" "0" "1" "0" "1" "0"
## [577] "1" "1" "1" "1" "1" "0" "0" "1" "1" "1" "0" "1" "0" "0" "0" "0" "1" "1"
## [595] "1" "0" "1" "1" "1" "0" "1" "1" "1" "1" "1" "0" "1" "0" "1" "1" "1" "1"
## [613] "1" "0" "0" "0" "1" "0" "1" "1" "1" "1" "0" "1" "1" "1" "1" "0" "1" "1"
## [631] "1" "1" "1" "0" "1" "0" "1" "1" "0" "1" "1" "0" "1" "1" "0" "1" "0" "0"
## [649] "1" "1" "1" "1" "1" "1" "0" "1" "1" "1" "1" "0" "1" "1" "1" "0" "0" "1"
## [667] "0" "1" "0" "1" "1" "1" "0" "1" "1" "1" "1" "1" "1" "1" "1" "0" "1" "1"
## [685] "1" "0" "1" "0" "1" "1" "1" "0" "1" "1" "1" "1" "1" "0" "1" "0" "1" "0"
## [703] "0" "0" "1" "1" "1" "1" "1" "1" "0" "1" "1" "1" "0" "0" "0" "1" "0" "0"
## [721] "1" "1" "1" "0" "1" "1" "0" "0" "0" "0" "1" "0" "1" "0" "0" "1" "0" "1"
## [739] "1" "0" "1" "1" "0" "0" "0" "0" "1" "1" "1" "1" "0" "1" "0" "0" "1" "1"
## [757] "0" "0" "0" "1" "1" "0" "1" "1" "1" "1" "1" "1" "1" "0" "1" "1" "1" "1"
## [775] "1" "1" "1" "0" "1" "1" "1" "0" "0" "1" "1" "0" "1" "1" "1" "1" "1" "1"
## [793] "1" "1" "1" "1" "1" "0" "1" "1" "0" "1" "1" "0" "1" "1" "1" "1" "0" "1"
## [811] "1" "1" "1" "0" "0" "1" "1" "0" "0" "0" "0" "1" "1" "0" "0" "1" "1" "0"
## [829] "1" "1" "1" "0" "0" "0" "1" "1" "1" "1" "1" "1" "1" "0" "1" "1" "0" "0"
## [847] "0" "1" "0" "1" "1" "1" "1" "1" "0" "0" "1" "0" "1" "1" "1" "1" "1" "1"
## [865] "1" "1" "0" "0" "1" "1" "1" "1" "0" "0" "1" "1" "1" "1" "1" "0" "1" "1"
## [883] "1" "1" "1" "1" "0" "1" "1" "0" "1" "1" "1" "1" "1" "0" "0" "1" "1" "1"
## [901] "1" "1" "0" "0" "0" "1" "1" "0" "1" "0" "0" "0" "0" "1" "1" "1" "0" "1"
## [919] "0" "0" "0" "1" "1" "0" "0" "0" "1" "0" "1" "0" "0" "1" "1" "1" "1" "1"
## [937] "0" "0" "1" "1" "0" "0" "1" "0" "0" "0" "1" "1" "0" "1" "1" "0" "1" "0"
## [955] "1" "1" "1" "0" "0" "1" "1" "1" "1" "1" "0" "1" "1" "0" "1" "0" "1" "0"
## [973] "1" "0" "1" "0" "1" "1" "1" "1" "1" "1"
```

```r
# compute accuracy
accuracy = sum(predictions==testing$ever_married)/length(predictions)
cat("Accuracy:\n")
```

```
## Accuracy:
```

```
accuracy
```

```
## [1] 0.9022403
```

logistic model to predict stroke

Requires: data, libraries

```
stroke.glm = stroke
stroke.glm$smoking_status[which(stroke.glm$smoking_status=='smokes' | stroke.glm$smoking_status=='forme
stroke.glm$smoking_status[which(stroke.glm$smoking_status!=1)] = 0

inds = sample(1:nrow(stroke.glm), round(nrow(stroke.glm)/2))
train = stroke.glm[inds,]
test = stroke.glm[-inds,]
super.model = glm(stroke~gender+hypertension+heart_disease+ever_married+work_type+Residence_type+avg_glu
basic.model = glm(stroke~hypertension+avg_glucose_level+smoking_status+age,data=train, family=binomial(l
summary(super.model)
```

```
##
## Call:
## glm(formula = stroke ~ gender + hypertension + heart_disease +
##     ever_married + work_type + Residence_type + avg_glucose_level +
##     smoking_status + age + bmi, family = binomial(logit), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1642  -0.2907  -0.1478  -0.0851   3.3388
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -6.828533   1.083353  -6.303 2.92e-10 ***
## gender1               -0.054863   0.216288  -0.254  0.79976
## hypertension           0.665830   0.242889   2.741  0.00612 **
## heart_disease          0.358755   0.283452   1.266  0.20563
## ever_married1         -0.136971   0.358853  -0.382  0.70269
## work_typeGovt_job     -1.405947   1.203628  -1.168  0.24277
## work_typeNever_worked  -10.577984 456.969352  -0.023  0.98153
## work_typePrivate      -1.142183   1.179387  -0.968  0.33282
## work_typeSelf-employed -1.849073   1.218169  -1.518  0.12904
## Residence_type1        0.283919   0.212352   1.337  0.18122
## avg_glucose_level      0.004764   0.001809   2.633  0.00846 **
## smoking_status1        0.278324   0.214595   1.297  0.19464
## age                    0.073650   0.008977   8.205 2.32e-16 ***
## bmi                   -0.001571   0.016951  -0.093  0.92616
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 885.82  on 2453  degrees of freedom
## Residual deviance: 691.99  on 2440  degrees of freedom
## AIC: 719.99
##
## Number of Fisher Scoring iterations: 14
```

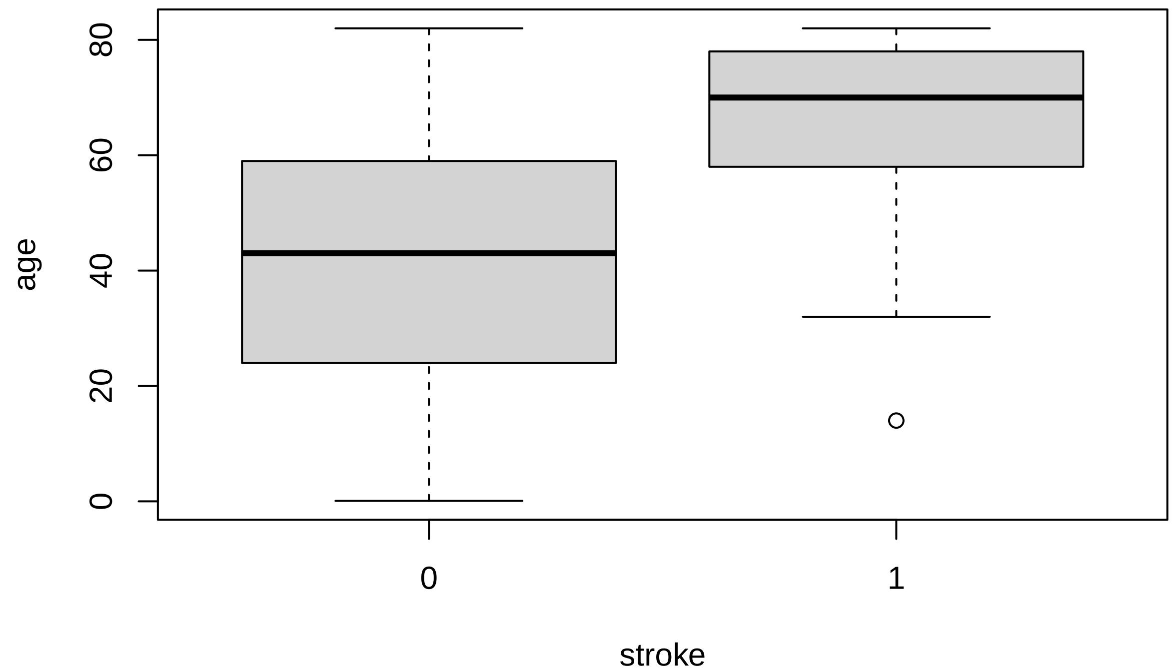
```
summary(basic.model)
```

```
##
## Call:
## glm(formula = stroke ~ hypertension + avg_glucose_level + smoking_status +
##     age, family = binomial(logit), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0849  -0.3093  -0.1608  -0.0778   3.6236
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.795056   0.556277 -14.013  < 2e-16 ***
## hypertension       0.655595   0.236944   2.767  0.00566 **
## avg_glucose_level  0.004882   0.001737   2.810  0.00495 **
## smoking_status1    0.302172   0.208558   1.449  0.14738
## age                0.067741   0.007799   8.686  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 885.82  on 2453  degrees of freedom
## Residual deviance: 704.80  on 2449  degrees of freedom
## AIC: 714.8
##
## Number of Fisher Scoring iterations: 7
```

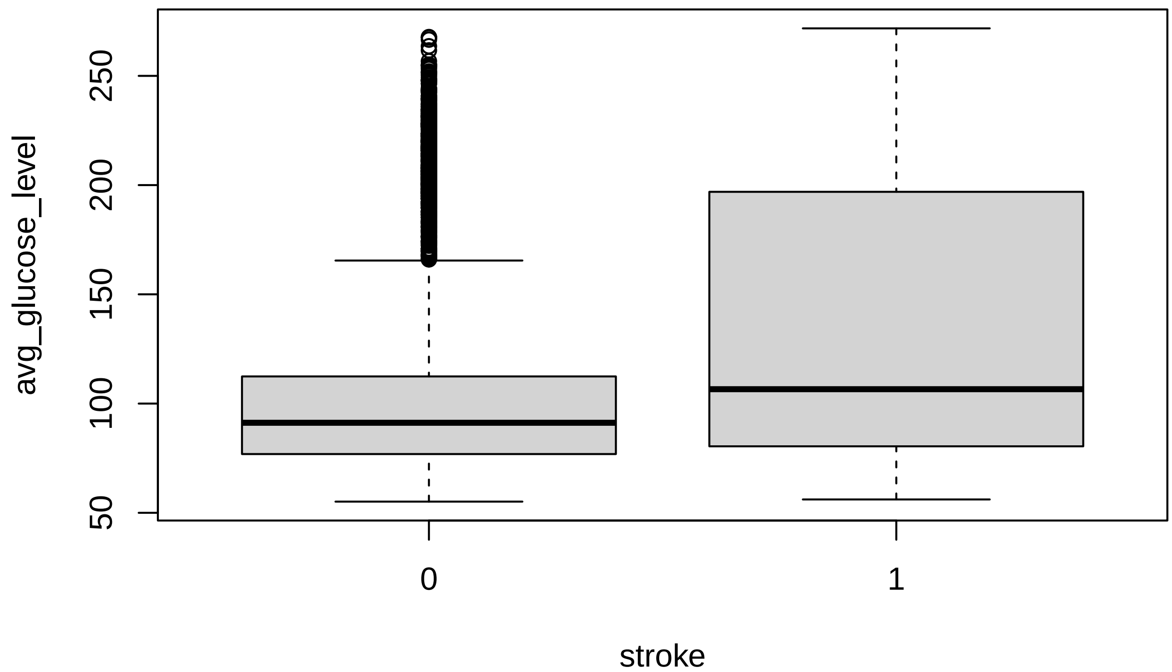
Analyzing results from logistic regression

Requires: logistic model, data, libraries

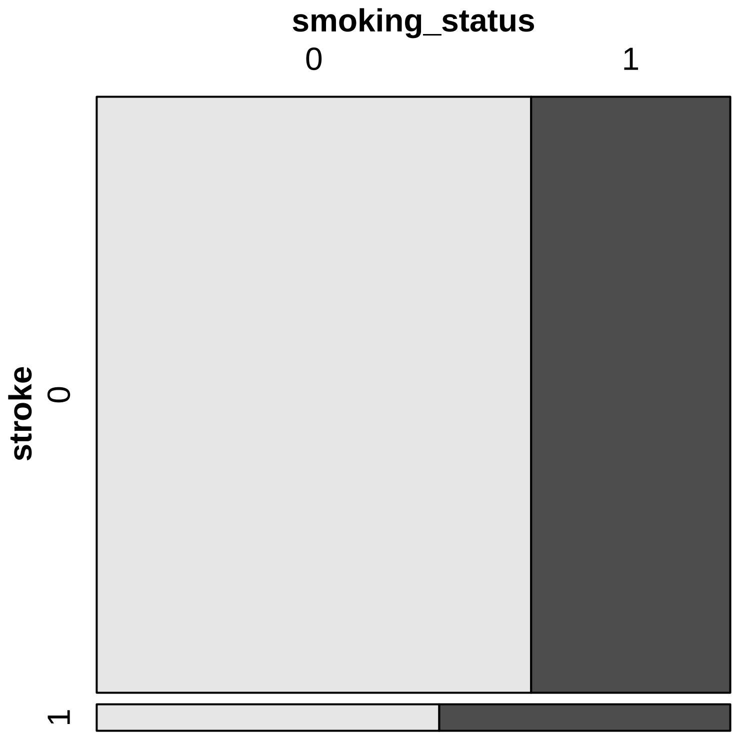
```
# plotting
boxplot(age~stroke, data=stroke.glm)
```



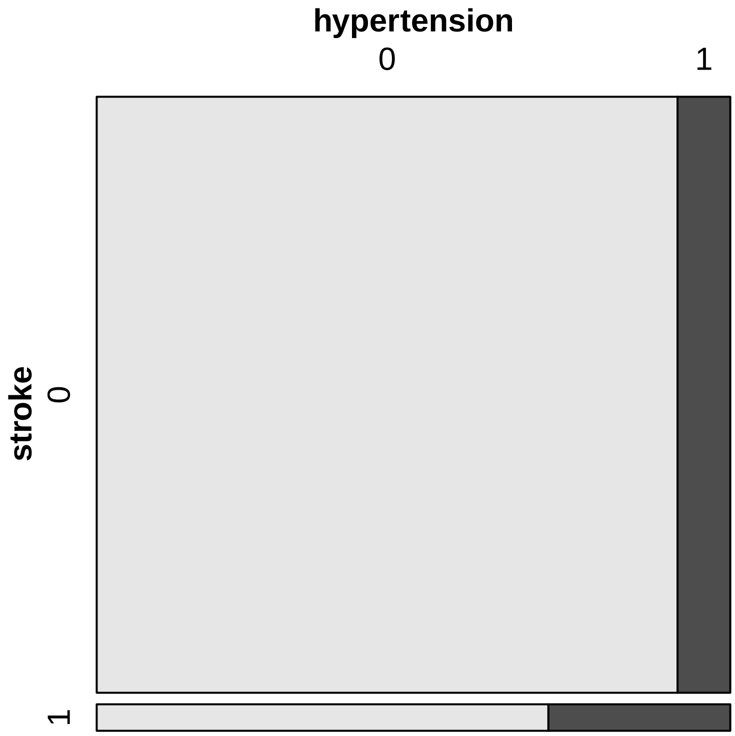

```
boxplot(avg_glucose_level~stroke, data=stroke.glm)
```




```
#stroke.glm$smoking_status = as.factor(stroke.glm$smoking_status)
mosaic(smoking_status~stroke, data=stroke.glm)
```




```
mosaic(hypertension~stroke, data=stroke.glm)
```




```
# hypothesis tests for differences
t.test(age~stroke, data=stroke.glm)
```

```
##
##  Welch Two Sample t-test
##
## data:  age by stroke
## t = -28.286, df = 271.68, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -27.75517 -24.14305
## sample estimates:
## mean in group 0 mean in group 1
##        41.76381        67.71292
```

```
t.test(avg_glucose_level~stroke, data=stroke.glm)
```

```
##
##  Welch Two Sample t-test
##
## data:  avg_glucose_level by stroke
## t = -7.0034, df = 216.86, p-value = 3.093e-11
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -39.18102 -21.97102
```

```
## sample estimates:
## mean in group 0 mean in group 1
##        103.9954        134.5714
```

```
#stroke.glm$stroke[stroke.glm$stroke==1] = 'Y'
#stroke.glm$stroke[stroke.glm$stroke==0] = 'N'
smoke.table = table(stroke.glm$stroke, stroke.glm$smoking_status)
prop.test(smoke.table) # does not
```

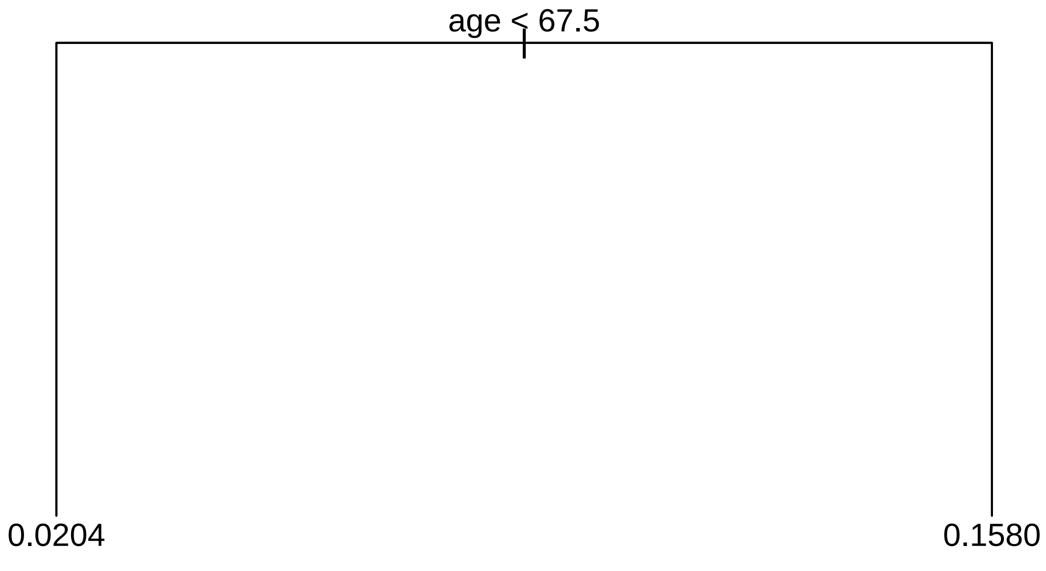
```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  smoke.table
## X-squared = 18.66, df = 1, p-value = 1.562e-05
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.07365546 0.21636043
## sample estimates:
##    prop 1    prop 2
## 0.6856778 0.5406699
```

```
ht.table = table(stroke.glm$hypertension, stroke.glm$stroke)
prop.test(ht.table) # works
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  ht.table
## X-squared = 97.239, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.06660165 0.13261262
## sample estimates:
##    prop 1    prop 2
## 0.9665694 0.8669623
```

ignore for now

```
tr.age = tree(stroke~age, data=stroke.glm)
stroke.glm$stroke = as.factor(stroke.glm$stroke)
plot(tr.age)
text(tr.age)
```


age < 67.5
0.0204
0.1580