HW8

Rosemary Kinuthia

4/17/2018

K-nearest neighbor

Let's try a variation on the NHANES data set again.

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────── tidyverse 1.2.1 ──
```

```
## ✔ ggplot2 2.2.1     ✔ purrr   0.2.4
## ✔ tibble  1.4.2     ✔ dplyr   0.7.4
## ✔ tidyr   0.8.0     ✔ stringr 1.3.0
## ✔ readr   1.1.1     ✔ forcats 0.3.0
```

```
## ── Conflicts ────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(class)
library(rpart)
library(NHANES)
library(RColorBrewer)
library(plot3D)
library(parallel)
library(randomForestSRC)
```

```
##
##  randomForestSRC 2.5.1
##
##  Type rfsrc.news() to see new features, changes, and bug fixes.
##
```

```
##
## Attaching package: 'randomForestSRC'
```

```
## The following object is masked from 'package:purrr':
##
##     partial
```

```
library(ggRandomForests)
```

```
##
## Attaching package: 'ggRandomForests'
```

```
## The following object is masked from 'package:randomForestSRC':
##
##     partial.rfsrc
```

```
library(mosaic)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggformula
```

```
##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

```
## Loading required package: mosaicData
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following object is masked from 'package:purrr':
##
##     cross
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median,
##     prop.test, quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

Create the NHANES dataset again, just like we did in class, only using sleep trouble (variable name = SleepTrouble) as the dependent variable, instead of Diabetes.

```
# Create the NHANES dataset again
people <- NHANES %>% dplyr::select(Age, Gender, SleepTrouble, BMI, HHIncome, PhysActive)
#%>% na.omit()

glimpse(people)
```

```
## Observations: 10,000
## Variables: 6
## $ Age          <int> 34, 34, 34, 4, 49, 9, 8, 45, 45, 45, 66, 58, 54, ...
## $ Gender       <fct> male, male, male, male, female, male, male, femal...
## $ SleepTrouble <fct> Yes, Yes, Yes, NA, Yes, NA, NA, No, No, No, No, N...
## $ BMI          <dbl> 32.22, 32.22, 32.22, 15.30, 30.57, 16.82, 20.64, ...
## $ HHIncome     <fct> 25000-34999, 25000-34999, 25000-34999, 20000-2499...
## $ PhysActive   <fct> No, No, No, NA, No, NA, NA, Yes, Yes, Yes, Yes, Y...
```

Problem 1

What is the marginal distribution of sleep trouble?

```
tally(~ SleepTrouble, data = people, format = "percent")
```

```
## SleepTrouble
##    No   Yes  <NA>
## 57.99 19.73 22.28
```

Recall from our prior work, the packages work better if the dataset is a dataframe, and the variables are numeric.

```
class(people)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
# Convert back to dataframe
people <- as.data.frame(people)
glimpse(people)
```

```
## Observations: 10,000
## Variables: 6
## $ Age          <int> 34, 34, 34, 4, 49, 9, 8, 45, 45, 45, 66, 58, 54, ...
## $ Gender       <fct> male, male, male, male, female, male, male, femal...
## $ SleepTrouble <fct> Yes, Yes, Yes, NA, Yes, NA, NA, No, No, No, No, N...
## $ BMI          <dbl> 32.22, 32.22, 32.22, 15.30, 30.57, 16.82, 20.64, ...
## $ HHIncome     <fct> 25000-34999, 25000-34999, 25000-34999, 20000-2499...
## $ PhysActive   <fct> No, No, No, NA, No, NA, NA, Yes, Yes, Yes, Yes, Y...
```

```
# Convert factors to numeric - the packages just seem to work better that way
people$Gender <- as.numeric(people$Gender)
people$SleepTrouble <- as.numeric(people$SleepTrouble)
people$HHIncome <- as.numeric(people$HHIncome)
people$PhysActive <- as.numeric(people$PhysActive)

people <- na.omit(people)

glimpse(people)
```

```
## Observations: 7,037
## Variables: 6
## $ Age          <int> 34, 34, 34, 49, 45, 45, 45, 66, 58, 54, 58, 50, 3...
## $ Gender       <dbl> 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 1, 2, 2, 2, 1, 1, 2...
## $ SleepTrouble <dbl> 2, 2, 2, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 2...
## $ BMI          <dbl> 32.22, 32.22, 32.22, 30.57, 27.24, 27.24, 27.24, ...
## $ HHIncome     <dbl> 6, 6, 6, 7, 11, 11, 11, 6, 12, 10, 11, 4, 6, 4, 1...
## $ PhysActive   <dbl> 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2...
```

Apply the k-nearest neighbor procedure to predict SleepTrouble from the other covariates, as we did for Diabetes. Use k = 1, 3, 5, and 20.

Problem 2

```
# Apply knn procedure to predict SleepTrouble
# Let's try different values of k to see how that affects performance

knn.1 <- knn(train = people, test = people, cl = as.numeric(people$SleepTrouble), k = 1)
knn.3 <- knn(train = people, test = people, cl = people$SleepTrouble, k = 3)
knn.5 <- knn(train = people, test = people, cl = people$SleepTrouble, k = 5)
knn.20 <- knn(train = people, test = people, cl = people$SleepTrouble, k = 20)

#knn.1
```

Now let's see how well these classifiers work overall

Problem 3

```
# Calculate the percent predicted correctly
# How well do these classifiers (k = 1, 3, 5, 20) work?

100*sum(people$SleepTrouble == knn.1)/length(knn.1)
```

```
## [1] 100
```

```
100*sum(people$SleepTrouble == knn.3)/length(knn.3)
```

```
## [1] 92.25522
```

```
100*sum(people$SleepTrouble == knn.5)/length(knn.5)
```

```
## [1] 88.58889
```

```
100*sum(people$SleepTrouble == knn.20)/length(knn.20)
```

```
## [1] 78.8262
```

Problem 4

What about success overall?

```
table(knn.1, people$SleepTrouble)
```

```
##
## knn.1    1    2
##     1 5239    0
##     2    0 1798
```

```
table(knn.3, people$SleepTrouble)
```

```
##
## knn.3    1    2
##     1 5070  376
##     2  169 1422
```

```
table(knn.5, people$SleepTrouble)
```

```
##
## knn.5    1    2
##     1 5023  587
##     2  216 1211
```

```
table(knn.20, people$SleepTrouble)
```

```
##
## knn.20    1    2
##      1 5104 1355
##      2  135  443
```

The github repository for this assignment can be accessed via this link (https://github.com/RosemaryKinuthia/2018week11.git

(https://github.com/RosemaryKinuthia/2018week11.git))
[https://github.com/RosemaryKinuthia/2018week11.git
(https://github.com/RosemaryKinuthia/2018week11.git)]