HW10

library(mice)

##   
## Attaching package: 'mice'

## The following object is masked from 'package:stats':  
##   
## filter

## The following objects are masked from 'package:base':  
##   
## cbind, rbind

rm(list = ls())  
data <- read.table("breast-cancer-wisconsin.data.txt", stringsAsFactors = FALSE, header = FALSE, sep = ",", na.strings="?")

Ref:<http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29>

Attribute Information:

1. Sample code number: id number
2. Clump Thickness: 1 - 10
3. Uniformity of Cell Size: 1 - 10
4. Uniformity of Cell Shape: 1 - 10
5. Marginal Adhesion: 1 - 10
6. Single Epithelial Cell Size: 1 - 10
7. Bare Nuclei: 1 - 10
8. Bland Chromatin: 1 - 10
9. Normal Nucleoli: 1 - 10
10. Mitoses: 1 - 10
11. Class: (2 for benign, 4 for malignant)

look at V7 you can see its the only column that has NA’s with a quantity of 16.

summary(data)

## V1 V2 V3 V4   
## Min. : 61634 Min. : 1.000 Min. : 1.000 Min. : 1.000   
## 1st Qu.: 870688 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 1.000   
## Median : 1171710 Median : 4.000 Median : 1.000 Median : 1.000   
## Mean : 1071704 Mean : 4.418 Mean : 3.134 Mean : 3.207   
## 3rd Qu.: 1238298 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 5.000   
## Max. :13454352 Max. :10.000 Max. :10.000 Max. :10.000   
##   
## V5 V6 V7 V8   
## Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
## 1st Qu.: 1.000 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 2.000   
## Median : 1.000 Median : 2.000 Median : 1.000 Median : 3.000   
## Mean : 2.807 Mean : 3.216 Mean : 3.545 Mean : 3.438   
## 3rd Qu.: 4.000 3rd Qu.: 4.000 3rd Qu.: 6.000 3rd Qu.: 5.000   
## Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
## NA's :16   
## V9 V10 V11   
## Min. : 1.000 Min. : 1.000 Min. :2.00   
## 1st Qu.: 1.000 1st Qu.: 1.000 1st Qu.:2.00   
## Median : 1.000 Median : 1.000 Median :2.00   
## Mean : 2.867 Mean : 1.589 Mean :2.69   
## 3rd Qu.: 4.000 3rd Qu.: 1.000 3rd Qu.:4.00   
## Max. :10.000 Max. :10.000 Max. :4.00   
##

missing\_value <-which(data$V7 == '?')  
missing\_value

## integer(0)

length(missing\_value / nrow(data))

## [1] 0

what percentage is missing?

16/699 \* 100

## [1] 2.288984

1. Use the mean/mode imputation method to impute values for the missing data.

this replaces with mean

imput\_mean <- data  
imput\_mean$V7[is.na(imput\_mean$V7)] <- mean(imput\_mean$V7, na.rm=TRUE)  
imput\_mean$V7

## [1] 1.000000 10.000000 2.000000 4.000000 1.000000 10.000000 10.000000  
## [8] 1.000000 1.000000 1.000000 1.000000 1.000000 3.000000 3.000000  
## [15] 9.000000 1.000000 1.000000 1.000000 10.000000 1.000000 10.000000  
## [22] 7.000000 1.000000 3.544656 1.000000 7.000000 1.000000 1.000000  
## [29] 1.000000 1.000000 1.000000 1.000000 5.000000 1.000000 1.000000  
## [36] 1.000000 1.000000 1.000000 10.000000 7.000000 3.544656 3.000000  
## [43] 10.000000 1.000000 1.000000 1.000000 9.000000 1.000000 1.000000  
## [50] 8.000000 3.000000 4.000000 5.000000 8.000000 8.000000 5.000000  
## [57] 6.000000 1.000000 10.000000 2.000000 3.000000 2.000000 8.000000  
## [64] 2.000000 1.000000 2.000000 1.000000 10.000000 9.000000 1.000000  
## [71] 1.000000 2.000000 1.000000 10.000000 4.000000 2.000000 1.000000  
## [78] 1.000000 3.000000 1.000000 1.000000 1.000000 1.000000 2.000000  
## [85] 9.000000 4.000000 8.000000 10.000000 1.000000 1.000000 1.000000  
## [92] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [99] 6.000000 10.000000 5.000000 5.000000 1.000000 3.000000 1.000000  
## [106] 3.000000 10.000000 10.000000 1.000000 9.000000 2.000000 9.000000  
## [113] 10.000000 8.000000 3.000000 5.000000 2.000000 10.000000 3.000000  
## [120] 2.000000 1.000000 2.000000 10.000000 10.000000 7.000000 1.000000  
## [127] 10.000000 1.000000 10.000000 1.000000 1.000000 1.000000 10.000000  
## [134] 1.000000 1.000000 2.000000 1.000000 1.000000 1.000000 3.544656  
## [141] 1.000000 1.000000 5.000000 5.000000 1.000000 3.544656 8.000000  
## [148] 2.000000 1.000000 10.000000 1.000000 10.000000 5.000000 3.000000  
## [155] 1.000000 10.000000 1.000000 1.000000 3.544656 10.000000 10.000000  
## [162] 1.000000 1.000000 3.000000 3.544656 2.000000 10.000000 1.000000  
## [169] 1.000000 1.000000 1.000000 1.000000 1.000000 10.000000 10.000000  
## [176] 10.000000 1.000000 1.000000 1.000000 10.000000 1.000000 1.000000  
## [183] 1.000000 10.000000 10.000000 1.000000 8.000000 10.000000 8.000000  
## [190] 1.000000 8.000000 10.000000 1.000000 1.000000 1.000000 1.000000  
## [197] 7.000000 1.000000 1.000000 1.000000 10.000000 10.000000 1.000000  
## [204] 1.000000 1.000000 10.000000 5.000000 1.000000 1.000000 1.000000  
## [211] 10.000000 8.000000 1.000000 10.000000 10.000000 5.000000 1.000000  
## [218] 1.000000 4.000000 1.000000 1.000000 10.000000 5.000000 8.000000  
## [225] 10.000000 1.000000 10.000000 5.000000 1.000000 10.000000 7.000000  
## [232] 8.000000 1.000000 10.000000 1.000000 3.544656 10.000000 2.000000  
## [239] 9.000000 10.000000 2.000000 1.000000 1.000000 5.000000 1.000000  
## [246] 2.000000 10.000000 9.000000 1.000000 3.544656 1.000000 10.000000  
## [253] 10.000000 10.000000 8.000000 10.000000 1.000000 1.000000 1.000000  
## [260] 8.000000 10.000000 10.000000 10.000000 10.000000 3.000000 1.000000  
## [267] 10.000000 10.000000 4.000000 1.000000 10.000000 1.000000 10.000000  
## [274] 4.000000 1.000000 3.544656 1.000000 1.000000 1.000000 7.000000  
## [281] 1.000000 1.000000 10.000000 10.000000 10.000000 10.000000 10.000000  
## [288] 1.000000 5.000000 10.000000 1.000000 1.000000 3.544656 10.000000  
## [295] 3.544656 10.000000 5.000000 3.544656 1.000000 10.000000 4.000000  
## [302] 1.000000 10.000000 1.000000 10.000000 10.000000 1.000000 1.000000  
## [309] 3.000000 5.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [316] 3.544656 10.000000 8.000000 1.000000 5.000000 10.000000 3.544656  
## [323] 1.000000 10.000000 1.000000 1.000000 10.000000 1.000000 4.000000  
## [330] 10.000000 8.000000 1.000000 1.000000 10.000000 10.000000 1.000000  
## [337] 10.000000 1.000000 1.000000 10.000000 10.000000 1.000000 1.000000  
## [344] 1.000000 10.000000 1.000000 1.000000 1.000000 1.000000 8.000000  
## [351] 1.000000 1.000000 3.000000 10.000000 1.000000 1.000000 3.000000  
## [358] 10.000000 4.000000 7.000000 10.000000 10.000000 3.000000 3.000000  
## [365] 1.000000 1.000000 10.000000 10.000000 1.000000 1.000000 1.000000  
## [372] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [379] 1.000000 1.000000 1.000000 10.000000 1.000000 1.000000 1.000000  
## [386] 1.000000 10.000000 1.000000 1.000000 2.000000 1.000000 10.000000  
## [393] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [400] 1.000000 9.000000 1.000000 1.000000 4.000000 1.000000 1.000000  
## [407] 1.000000 1.000000 2.000000 1.000000 1.000000 3.544656 4.000000  
## [414] 1.000000 10.000000 3.000000 10.000000 1.000000 2.000000 1.000000  
## [421] 3.000000 10.000000 1.000000 1.000000 1.000000 10.000000 1.000000  
## [428] 2.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [435] 8.000000 10.000000 1.000000 1.000000 1.000000 1.000000 10.000000  
## [442] 4.000000 3.000000 2.000000 1.000000 1.000000 1.000000 1.000000  
## [449] 1.000000 10.000000 1.000000 1.000000 1.000000 10.000000 1.000000  
## [456] 6.000000 10.000000 3.000000 1.000000 1.000000 1.000000 5.000000  
## [463] 1.000000 1.000000 1.000000 4.000000 10.000000 10.000000 1.000000  
## [470] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [477] 1.000000 1.000000 1.000000 10.000000 1.000000 1.000000 5.000000  
## [484] 10.000000 1.000000 3.000000 1.000000 10.000000 3.000000 4.000000  
## [491] 1.000000 10.000000 1.000000 10.000000 5.000000 1.000000 1.000000  
## [498] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [505] 1.000000 1.000000 5.000000 4.000000 1.000000 1.000000 1.000000  
## [512] 1.000000 1.000000 1.000000 10.000000 10.000000 1.000000 1.000000  
## [519] 1.000000 10.000000 1.000000 1.000000 5.000000 10.000000 1.000000  
## [526] 1.000000 1.000000 1.000000 1.000000 1.000000 10.000000 1.000000  
## [533] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [540] 1.000000 2.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [547] 10.000000 1.000000 1.000000 5.000000 1.000000 1.000000 1.000000  
## [554] 5.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [561] 1.000000 1.000000 1.000000 1.000000 1.000000 10.000000 1.000000  
## [568] 3.000000 10.000000 5.000000 10.000000 10.000000 1.000000 1.000000  
## [575] 2.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [582] 10.000000 10.000000 1.000000 1.000000 1.000000 10.000000 1.000000  
## [589] 3.000000 1.000000 1.000000 10.000000 10.000000 1.000000 10.000000  
## [596] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [603] 1.000000 1.000000 10.000000 8.000000 1.000000 1.000000 10.000000  
## [610] 1.000000 10.000000 2.000000 10.000000 1.000000 1.000000 1.000000  
## [617] 1.000000 3.544656 1.000000 1.000000 1.000000 2.000000 1.000000  
## [624] 1.000000 1.000000 4.000000 6.000000 5.000000 1.000000 1.000000  
## [631] 1.000000 1.000000 1.000000 3.000000 1.000000 1.000000 1.000000  
## [638] 2.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [645] 1.000000 1.000000 1.000000 1.000000 2.000000 1.000000 4.000000  
## [652] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [659] 10.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [666] 1.000000 1.000000 1.000000 1.000000 5.000000 8.000000 1.000000  
## [673] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000  
## [680] 1.000000 10.000000 10.000000 1.000000 1.000000 1.000000 1.000000  
## [687] 1.000000 1.000000 1.000000 1.000000 1.000000 5.000000 1.000000  
## [694] 1.000000 2.000000 1.000000 3.000000 4.000000 5.000000

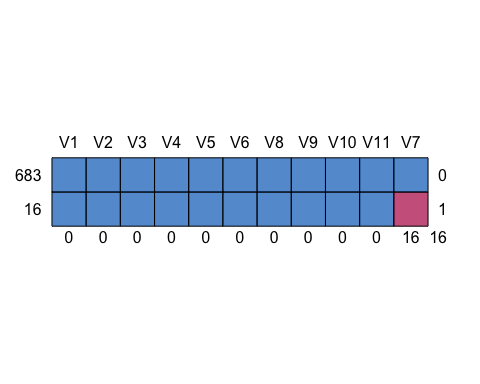
this replaces with mode

imput\_mode <- data  
imput\_mode$V7[is.na(imput\_mode$V7)] <- mode(imput\_mode$V7)  
imput\_mode$V7

## [1] "1" "10" "2" "4" "1" "10" "10"   
## [8] "1" "1" "1" "1" "1" "3" "3"   
## [15] "9" "1" "1" "1" "10" "1" "10"   
## [22] "7" "1" "numeric" "1" "7" "1" "1"   
## [29] "1" "1" "1" "1" "5" "1" "1"   
## [36] "1" "1" "1" "10" "7" "numeric" "3"   
## [43] "10" "1" "1" "1" "9" "1" "1"   
## [50] "8" "3" "4" "5" "8" "8" "5"   
## [57] "6" "1" "10" "2" "3" "2" "8"   
## [64] "2" "1" "2" "1" "10" "9" "1"   
## [71] "1" "2" "1" "10" "4" "2" "1"   
## [78] "1" "3" "1" "1" "1" "1" "2"   
## [85] "9" "4" "8" "10" "1" "1" "1"   
## [92] "1" "1" "1" "1" "1" "1" "1"   
## [99] "6" "10" "5" "5" "1" "3" "1"   
## [106] "3" "10" "10" "1" "9" "2" "9"   
## [113] "10" "8" "3" "5" "2" "10" "3"   
## [120] "2" "1" "2" "10" "10" "7" "1"   
## [127] "10" "1" "10" "1" "1" "1" "10"   
## [134] "1" "1" "2" "1" "1" "1" "numeric"  
## [141] "1" "1" "5" "5" "1" "numeric" "8"   
## [148] "2" "1" "10" "1" "10" "5" "3"   
## [155] "1" "10" "1" "1" "numeric" "10" "10"   
## [162] "1" "1" "3" "numeric" "2" "10" "1"   
## [169] "1" "1" "1" "1" "1" "10" "10"   
## [176] "10" "1" "1" "1" "10" "1" "1"   
## [183] "1" "10" "10" "1" "8" "10" "8"   
## [190] "1" "8" "10" "1" "1" "1" "1"   
## [197] "7" "1" "1" "1" "10" "10" "1"   
## [204] "1" "1" "10" "5" "1" "1" "1"   
## [211] "10" "8" "1" "10" "10" "5" "1"   
## [218] "1" "4" "1" "1" "10" "5" "8"   
## [225] "10" "1" "10" "5" "1" "10" "7"   
## [232] "8" "1" "10" "1" "numeric" "10" "2"   
## [239] "9" "10" "2" "1" "1" "5" "1"   
## [246] "2" "10" "9" "1" "numeric" "1" "10"   
## [253] "10" "10" "8" "10" "1" "1" "1"   
## [260] "8" "10" "10" "10" "10" "3" "1"   
## [267] "10" "10" "4" "1" "10" "1" "10"   
## [274] "4" "1" "numeric" "1" "1" "1" "7"   
## [281] "1" "1" "10" "10" "10" "10" "10"   
## [288] "1" "5" "10" "1" "1" "numeric" "10"   
## [295] "numeric" "10" "5" "numeric" "1" "10" "4"   
## [302] "1" "10" "1" "10" "10" "1" "1"   
## [309] "3" "5" "1" "1" "1" "1" "1"   
## [316] "numeric" "10" "8" "1" "5" "10" "numeric"  
## [323] "1" "10" "1" "1" "10" "1" "4"   
## [330] "10" "8" "1" "1" "10" "10" "1"   
## [337] "10" "1" "1" "10" "10" "1" "1"   
## [344] "1" "10" "1" "1" "1" "1" "8"   
## [351] "1" "1" "3" "10" "1" "1" "3"   
## [358] "10" "4" "7" "10" "10" "3" "3"   
## [365] "1" "1" "10" "10" "1" "1" "1"   
## [372] "1" "1" "1" "1" "1" "1" "1"   
## [379] "1" "1" "1" "10" "1" "1" "1"   
## [386] "1" "10" "1" "1" "2" "1" "10"   
## [393] "1" "1" "1" "1" "1" "1" "1"   
## [400] "1" "9" "1" "1" "4" "1" "1"   
## [407] "1" "1" "2" "1" "1" "numeric" "4"   
## [414] "1" "10" "3" "10" "1" "2" "1"   
## [421] "3" "10" "1" "1" "1" "10" "1"   
## [428] "2" "1" "1" "1" "1" "1" "1"   
## [435] "8" "10" "1" "1" "1" "1" "10"   
## [442] "4" "3" "2" "1" "1" "1" "1"   
## [449] "1" "10" "1" "1" "1" "10" "1"   
## [456] "6" "10" "3" "1" "1" "1" "5"   
## [463] "1" "1" "1" "4" "10" "10" "1"   
## [470] "1" "1" "1" "1" "1" "1" "1"   
## [477] "1" "1" "1" "10" "1" "1" "5"   
## [484] "10" "1" "3" "1" "10" "3" "4"   
## [491] "1" "10" "1" "10" "5" "1" "1"   
## [498] "1" "1" "1" "1" "1" "1" "1"   
## [505] "1" "1" "5" "4" "1" "1" "1"   
## [512] "1" "1" "1" "10" "10" "1" "1"   
## [519] "1" "10" "1" "1" "5" "10" "1"   
## [526] "1" "1" "1" "1" "1" "10" "1"   
## [533] "1" "1" "1" "1" "1" "1" "1"   
## [540] "1" "2" "1" "1" "1" "1" "1"   
## [547] "10" "1" "1" "5" "1" "1" "1"   
## [554] "5" "1" "1" "1" "1" "1" "1"   
## [561] "1" "1" "1" "1" "1" "10" "1"   
## [568] "3" "10" "5" "10" "10" "1" "1"   
## [575] "2" "1" "1" "1" "1" "1" "1"   
## [582] "10" "10" "1" "1" "1" "10" "1"   
## [589] "3" "1" "1" "10" "10" "1" "10"   
## [596] "1" "1" "1" "1" "1" "1" "1"   
## [603] "1" "1" "10" "8" "1" "1" "10"   
## [610] "1" "10" "2" "10" "1" "1" "1"   
## [617] "1" "numeric" "1" "1" "1" "2" "1"   
## [624] "1" "1" "4" "6" "5" "1" "1"   
## [631] "1" "1" "1" "3" "1" "1" "1"   
## [638] "2" "1" "1" "1" "1" "1" "1"   
## [645] "1" "1" "1" "1" "2" "1" "4"   
## [652] "1" "1" "1" "1" "1" "1" "1"   
## [659] "10" "1" "1" "1" "1" "1" "1"   
## [666] "1" "1" "1" "1" "5" "8" "1"   
## [673] "1" "1" "1" "1" "1" "1" "1"   
## [680] "1" "10" "10" "1" "1" "1" "1"   
## [687] "1" "1" "1" "1" "1" "5" "1"   
## [694] "1" "2" "1" "3" "4" "5"

1. Use regression to impute values for the missing data.

md.pattern(data)



## V1 V2 V3 V4 V5 V6 V8 V9 V10 V11 V7   
## 683 1 1 1 1 1 1 1 1 1 1 1 0  
## 16 1 1 1 1 1 1 1 1 1 1 0 1  
## 0 0 0 0 0 0 0 0 0 0 16 16

set.seed(1)  
mice\_imp <- mice(data,method = "norm", m = 1)

##   
## iter imp variable  
## 1 1 V7  
## 2 1 V7  
## 3 1 V7  
## 4 1 V7  
## 5 1 V7

mice\_imp

## Class: mids  
## Number of multiple imputations: 1   
## Imputation methods:  
## V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11   
## "" "" "" "" "" "" "norm" "" "" "" ""   
## PredictorMatrix:  
## V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11  
## V1 0 1 1 1 1 1 1 1 1 1 1  
## V2 1 0 1 1 1 1 1 1 1 1 1  
## V3 1 1 0 1 1 1 1 1 1 1 1  
## V4 1 1 1 0 1 1 1 1 1 1 1  
## V5 1 1 1 1 0 1 1 1 1 1 1  
## V6 1 1 1 1 1 0 1 1 1 1 1

1. Use regression with perturbation to impute values for the missing data.

set.seed(1)  
mice\_imp\_pert <- mice(data,method = "norm.nob", m = 1)

##   
## iter imp variable  
## 1 1 V7  
## 2 1 V7  
## 3 1 V7  
## 4 1 V7  
## 5 1 V7

mice\_imp\_pert$data$V7

## [1] 1 10 2 4 1 10 10 1 1 1 1 1 3 3 9 1 1 1 10 1 10 7 1 NA 1  
## [26] 7 1 1 1 1 1 1 5 1 1 1 1 1 10 7 NA 3 10 1 1 1 9 1 1 8  
## [51] 3 4 5 8 8 5 6 1 10 2 3 2 8 2 1 2 1 10 9 1 1 2 1 10 4  
## [76] 2 1 1 3 1 1 1 1 2 9 4 8 10 1 1 1 1 1 1 1 1 1 1 6 10  
## [101] 5 5 1 3 1 3 10 10 1 9 2 9 10 8 3 5 2 10 3 2 1 2 10 10 7  
## [126] 1 10 1 10 1 1 1 10 1 1 2 1 1 1 NA 1 1 5 5 1 NA 8 2 1 10  
## [151] 1 10 5 3 1 10 1 1 NA 10 10 1 1 3 NA 2 10 1 1 1 1 1 1 10 10  
## [176] 10 1 1 1 10 1 1 1 10 10 1 8 10 8 1 8 10 1 1 1 1 7 1 1 1  
## [201] 10 10 1 1 1 10 5 1 1 1 10 8 1 10 10 5 1 1 4 1 1 10 5 8 10  
## [226] 1 10 5 1 10 7 8 1 10 1 NA 10 2 9 10 2 1 1 5 1 2 10 9 1 NA  
## [251] 1 10 10 10 8 10 1 1 1 8 10 10 10 10 3 1 10 10 4 1 10 1 10 4 1  
## [276] NA 1 1 1 7 1 1 10 10 10 10 10 1 5 10 1 1 NA 10 NA 10 5 NA 1 10  
## [301] 4 1 10 1 10 10 1 1 3 5 1 1 1 1 1 NA 10 8 1 5 10 NA 1 10 1  
## [326] 1 10 1 4 10 8 1 1 10 10 1 10 1 1 10 10 1 1 1 10 1 1 1 1 8  
## [351] 1 1 3 10 1 1 3 10 4 7 10 10 3 3 1 1 10 10 1 1 1 1 1 1 1  
## [376] 1 1 1 1 1 1 10 1 1 1 1 10 1 1 2 1 10 1 1 1 1 1 1 1 1  
## [401] 9 1 1 4 1 1 1 1 2 1 1 NA 4 1 10 3 10 1 2 1 3 10 1 1 1  
## [426] 10 1 2 1 1 1 1 1 1 8 10 1 1 1 1 10 4 3 2 1 1 1 1 1 10  
## [451] 1 1 1 10 1 6 10 3 1 1 1 5 1 1 1 4 10 10 1 1 1 1 1 1 1  
## [476] 1 1 1 1 10 1 1 5 10 1 3 1 10 3 4 1 10 1 10 5 1 1 1 1 1  
## [501] 1 1 1 1 1 1 5 4 1 1 1 1 1 1 10 10 1 1 1 10 1 1 5 10 1  
## [526] 1 1 1 1 1 10 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 10 1 1 5  
## [551] 1 1 1 5 1 1 1 1 1 1 1 1 1 1 1 10 1 3 10 5 10 10 1 1 2  
## [576] 1 1 1 1 1 1 10 10 1 1 1 10 1 3 1 1 10 10 1 10 1 1 1 1 1  
## [601] 1 1 1 1 10 8 1 1 10 1 10 2 10 1 1 1 1 NA 1 1 1 2 1 1 1  
## [626] 4 6 5 1 1 1 1 1 3 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1  
## [651] 4 1 1 1 1 1 1 1 10 1 1 1 1 1 1 1 1 1 1 5 8 1 1 1 1  
## [676] 1 1 1 1 1 10 10 1 1 1 1 1 1 1 1 1 5 1 1 2 1 3 4 5