

## ▼ Homework 4

```
In [2]: library(tree)
```

## ▼ Problem 1

### ▼ (a)

```
In [142]: tahoe<-read.csv("Tahoe_Healthcare_Data.csv")
```

```
In [143]: head(tahoe)
```

A data.frame: 6 × 7

	age	female	flu_season	ed_admit	severity.score	comorbidity.score	readmit30
	<int>	<int>	<int>	<int>	<int>	<int>	<int>
1	100	1	1	1	38	112	0
2	83	1	0	1	8	109	1
3	74	0	1	0	1	80	0
4	66	1	1	1	25	4	0
5	68	1	1	1	25	32	0
6	80	1	0	1	29	172	0

In [144]: `str(tahoe)`

```
'data.frame':  4382 obs. of  7 variables:
 $ age           : int  100 83 74 66 68 80 71 72 69 65 ...
 $ female        : int   1 1 0 1 1 1 1 0 1 1 ...
 $ flu_season    : int   1 0 1 1 1 0 0 0 0 0 ...
 $ ed_admit      : int   1 1 0 1 1 1 1 0 1 0 ...
 $ severity.score : int   38 8 1 25 25 29 31 47 44 10 ...
 $ comorbidity.score: int  112 109 80 4 32 172 271 221 193 130 ...
 $ readmit30     : int   0 1 0 0 0 0 1 1 0 0 ...
```

In [145]: `tahoe$readmit30 <- factor(tahoe$readmit30)`

In [146]: `str(tahoe)`

```
'data.frame':  4382 obs. of  7 variables:
 $ age           : int  100 83 74 66 68 80 71 72 69 65 ...
 $ female        : int   1 1 0 1 1 1 1 0 1 1 ...
 $ flu_season    : int   1 0 1 1 1 0 0 0 0 0 ...
 $ ed_admit      : int   1 1 0 1 1 1 1 0 1 0 ...
 $ severity.score : int   38 8 1 25 25 29 31 47 44 10 ...
 $ comorbidity.score: int  112 109 80 4 32 172 271 221 193 130 ...
 $ readmit30     : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 2 2 1 1 ...
```

In [147]: `tree_tahoe<-tree(readmit30 ~ . , data = tahoe, split = "deviance", minsize=4, mindev=1e-3)`

In [148]: `tree_tahoe`

node), split, n, deviance, yval, (yprob)

\* denotes terminal node

```

1) root 4382 4702.000 0 ( 0.77225 0.22775 )
 2) comorbidity.score < 121.5 3084 2440.000 0 ( 0.86511 0.13489 )
 4) severity.score < 31.5 2423 1609.000 0 ( 0.89682 0.10318 )
 8) comorbidity.score < 68.5 1337 616.700 0 ( 0.93867 0.06133 )
16) flu_season < 0.5 790 255.100 0 ( 0.96203 0.03797 )
 32) severity.score < 14.5 444 80.120 0 ( 0.98198 0.01802 )
    64) comorbidity.score < 64.5 415 54.130 0 ( 0.98795 0.01205 ) *
    65) comorbidity.score > 64.5 29 19.290 0 ( 0.89655 0.10345 )
    130) severity.score < 7.5 16 0.000 0 ( 1.00000 0.00000 ) *
    131) severity.score > 7.5 13 14.050 0 ( 0.76923 0.23077 )
        262) severity.score < 8.5 2 0.000 1 ( 0.00000 1.00000 ) *
        263) severity.score > 8.5 11 6.702 0 ( 0.90909 0.09091 ) *
 33) severity.score > 14.5 346 163.800 0 ( 0.93642 0.06358 )
    66) age < 67.5 47 0.000 0 ( 1.00000 0.00000 ) *
    67) age > 67.5 299 157.200 0 ( 0.92642 0.07358 ) *
17) flu_season > 0.5 547 343.600 0 ( 0.90494 0.09506 ) *
 9) comorbidity.score > 68.5 1086 935.600 0 ( 0.84530 0.15470 )
18) severity.score < 11.5 500 346.500 0 ( 0.89000 0.11000 )
    36) severity.score < 9.5 434 326.100 0 ( 0.87558 0.12442 ) *
    37) severity.score > 9.5 66 10.360 0 ( 0.98485 0.01515 )
        74) age < 87 63 0.000 0 ( 1.00000 0.00000 ) *
        75) age > 87 3 3.819 0 ( 0.66667 0.33333 ) *
19) severity.score > 11.5 586 574.600 0 ( 0.80717 0.19283 )
    38) flu_season < 0.5 346 296.300 0 ( 0.84682 0.15318 )
        76) comorbidity.score < 118.5 330 290.800 0 ( 0.83939 0.16061 )
        152) comorbidity.score < 100.5 218 167.200 0 ( 0.87156 0.12844 ) *
        153) comorbidity.score > 100.5 112 118.900 0 ( 0.77679 0.22321 ) *
        77) comorbidity.score > 118.5 16 0.000 0 ( 1.00000 0.00000 ) *
    39) flu_season > 0.5 240 269.900 0 ( 0.75000 0.25000 )
        78) female < 0.5 118 107.400 0 ( 0.83051 0.16949 ) *
        79) female > 0.5 122 154.400 0 ( 0.67213 0.32787 )
            158) age < 74.5 44 41.720 0 ( 0.81818 0.18182 ) *
            159) age > 74.5 78 105.600 0 ( 0.58974 0.41026 ) *
 5) severity.score > 31.5 661 745.100 0 ( 0.74887 0.25113 )
    10) comorbidity.score < 47.5 195 153.100 0 ( 0.86667 0.13333 )
    20) severity.score < 85 192 140.700 0 ( 0.88021 0.11979 )

```

```

40) age < 88.5 164 133.000 0 ( 0.85976 0.14024 )
80) comorbidity.score < 45.5 149 128.200 0 ( 0.84564 0.15436 ) *
81) comorbidity.score > 45.5 15 0.000 0 ( 1.00000 0.00000 ) *
41) age > 88.5 28 0.000 0 ( 1.00000 0.00000 ) *
21) severity.score > 85 3 0.000 1 ( 0.00000 1.00000 ) *
11) comorbidity.score > 47.5 466 569.700 0 ( 0.69957 0.30043 )
22) severity.score < 54.5 362 418.800 0 ( 0.73481 0.26519 )
44) age < 75.5 121 111.700 0 ( 0.82645 0.17355 ) *
45) age > 75.5 241 298.900 0 ( 0.68880 0.31120 )
90) comorbidity.score < 80.5 113 124.300 0 ( 0.76106 0.23894 ) *
91) comorbidity.score > 80.5 128 169.400 0 ( 0.62500 0.37500 )
182) comorbidity.score < 115.5 108 146.700 0 ( 0.58333 0.41667 ) *
183) comorbidity.score > 115.5 20 16.910 0 ( 0.85000 0.15000 )
366) severity.score < 39.5 8 10.590 0 ( 0.62500 0.37500 ) *
367) severity.score > 39.5 12 0.000 0 ( 1.00000 0.00000 ) *
23) severity.score > 54.5 104 141.700 0 ( 0.57692 0.42308 )
46) female < 0.5 41 55.640 1 ( 0.41463 0.58537 ) *
47) female > 0.5 63 78.740 0 ( 0.68254 0.31746 )
94) comorbidity.score < 104.5 53 59.050 0 ( 0.75472 0.24528 ) *
95) comorbidity.score > 104.5 10 12.220 1 ( 0.30000 0.70000 )
190) age < 77.5 5 0.000 1 ( 0.00000 1.00000 ) *
191) age > 77.5 5 6.730 0 ( 0.60000 0.40000 ) *
3) comorbidity.score > 121.5 1298 1786.000 0 ( 0.55162 0.44838 )
6) comorbidity.score < 167.5 770 1001.000 0 ( 0.64545 0.35455 )
12) flu_season < 0.5 467 542.600 0 ( 0.73233 0.26767 )
24) severity.score < 43.5 395 419.000 0 ( 0.77722 0.22278 )
48) age < 92.5 386 396.600 0 ( 0.79016 0.20984 ) *
49) age > 92.5 9 9.535 1 ( 0.22222 0.77778 ) *
25) severity.score > 43.5 72 99.760 1 ( 0.48611 0.51389 ) *
13) flu_season > 0.5 303 419.900 0 ( 0.51155 0.48845 )
26) severity.score < 26.5 177 239.200 0 ( 0.59322 0.40678 )
52) comorbidity.score < 166.5 172 233.900 0 ( 0.58140 0.41860 ) *
53) comorbidity.score > 166.5 5 0.000 0 ( 1.00000 0.00000 ) *
27) severity.score > 26.5 126 169.300 1 ( 0.39683 0.60317 )
54) severity.score < 66.5 114 155.800 1 ( 0.42982 0.57018 )
108) comorbidity.score < 152.5 83 115.100 0 ( 0.50602 0.49398 ) *
109) comorbidity.score > 152.5 31 33.120 1 ( 0.22581 0.77419 )
218) comorbidity.score < 159.5 13 0.000 1 ( 0.00000 1.00000 ) *
219) comorbidity.score > 159.5 18 24.060 1 ( 0.38889 0.61111 )
438) severity.score < 34 7 8.376 0 ( 0.71429 0.28571 ) *
439) severity.score > 34 11 10.430 1 ( 0.18182 0.81818 ) *
55) severity.score > 66.5 12 6.884 1 ( 0.08333 0.91667 ) *
7) comorbidity.score > 167.5 528 716.500 1 ( 0.41477 0.58523 )

```

```

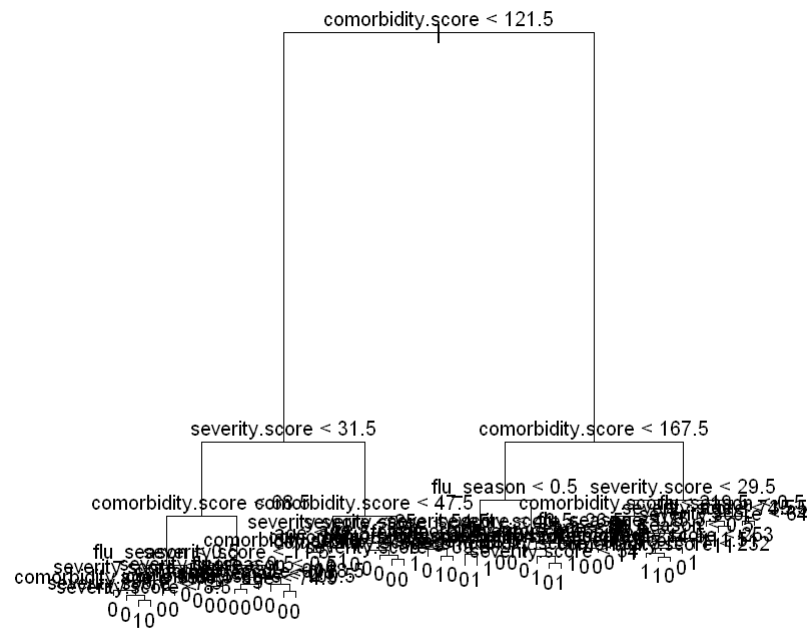
14) severity.score < 29.5 270 374.300 0 ( 0.50370 0.49630 )
28) comorbidity.score < 219.5 206 280.600 0 ( 0.57767 0.42233 )
56) flu_season < 0.5 117 148.900 0 ( 0.66667 0.33333 )
112) age < 86.5 106 139.500 0 ( 0.63208 0.36792 )
224) comorbidity.score < 214.5 101 134.700 0 ( 0.61386 0.38614 ) *
225) comorbidity.score > 214.5 5 0.000 0 ( 1.00000 0.00000 ) *
113) age > 86.5 11 0.000 0 ( 1.00000 0.00000 ) *
57) flu_season > 0.5 89 122.800 1 ( 0.46067 0.53933 ) *
29) comorbidity.score > 219.5 64 74.090 1 ( 0.26562 0.73438 )
58) age < 75.5 27 18.840 1 ( 0.11111 0.88889 ) *
59) age > 75.5 37 49.080 1 ( 0.37838 0.62162 )
118) flu_season < 0.5 19 25.010 0 ( 0.63158 0.36842 )
236) comorbidity.score < 253 14 19.410 0 ( 0.50000 0.50000 )
472) severity.score < 11.5 3 0.000 1 ( 0.00000 1.00000 ) *
473) severity.score > 11.5 11 14.420 0 ( 0.63636 0.36364 )
946) comorbidity.score < 232 7 9.561 1 ( 0.42857 0.57143 ) *
947) comorbidity.score > 232 4 0.000 0 ( 1.00000 0.00000 ) *
237) comorbidity.score > 253 5 0.000 0 ( 1.00000 0.00000 ) *
119) flu_season > 0.5 18 12.560 1 ( 0.11111 0.88889 ) *
15) severity.score > 29.5 258 324.100 1 ( 0.32171 0.67829 )
30) flu_season < 0.5 159 210.800 1 ( 0.37736 0.62264 )
60) severity.score < 35.5 38 39.110 1 ( 0.21053 0.78947 ) *
61) severity.score > 35.5 121 165.300 1 ( 0.42975 0.57025 )
122) severity.score < 64 107 147.900 1 ( 0.46729 0.53271 ) *
123) severity.score > 64 14 11.480 1 ( 0.14286 0.85714 ) *
31) flu_season > 0.5 99 107.300 1 ( 0.23232 0.76768 )
62) age < 74.5 25 13.940 1 ( 0.08000 0.92000 ) *
63) age > 74.5 74 88.280 1 ( 0.28378 0.71622 ) *

```

In [149]: `summary(tree_tahoe)`

```
Classification tree:
tree(formula = readmit30 ~ ., data = tahoe, split = "deviance",
      minsize = 4, mindev = 0.001)
Variables actually used in tree construction:
[1] "comorbidity.score" "severity.score"    "flu_season"
[4] "age"              "female"
Number of terminal nodes: 54
Residual mean deviance: 0.8214 = 3555 / 4328
Misclassification error rate: 0.1839 = 806 / 4382
```

```
In [150]: plot(tree_tahoe)
text(tree_tahoe, pretty=0)
```



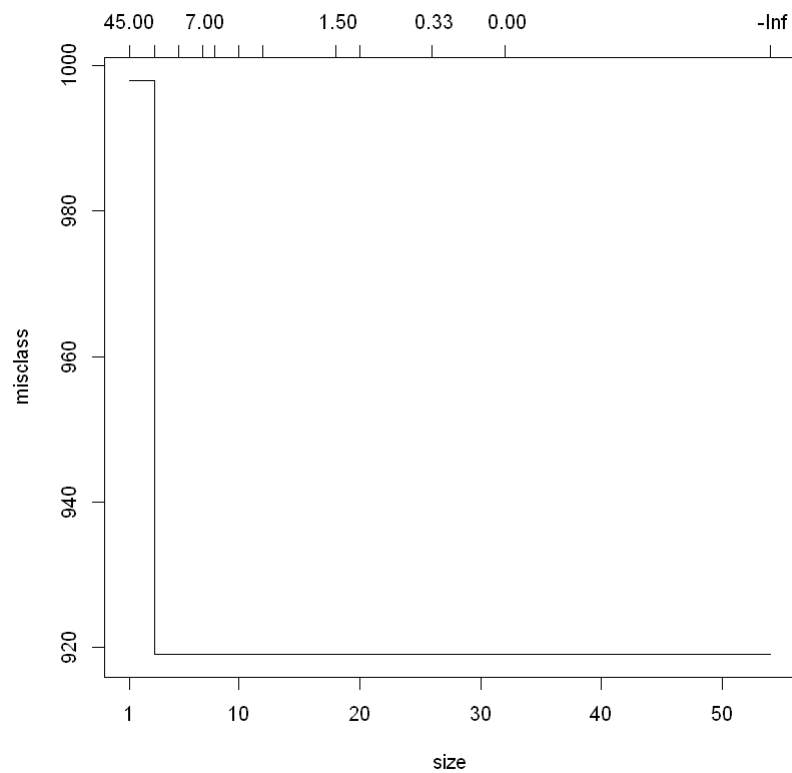
▼ (b)

```
In [163]: cv.tree_tahoe <- cv.tree(tree_tahoe, FUN = prune.misclass, K = 7)
```

```
In [164]: ▶ names(cv.tree_tahoe)
```

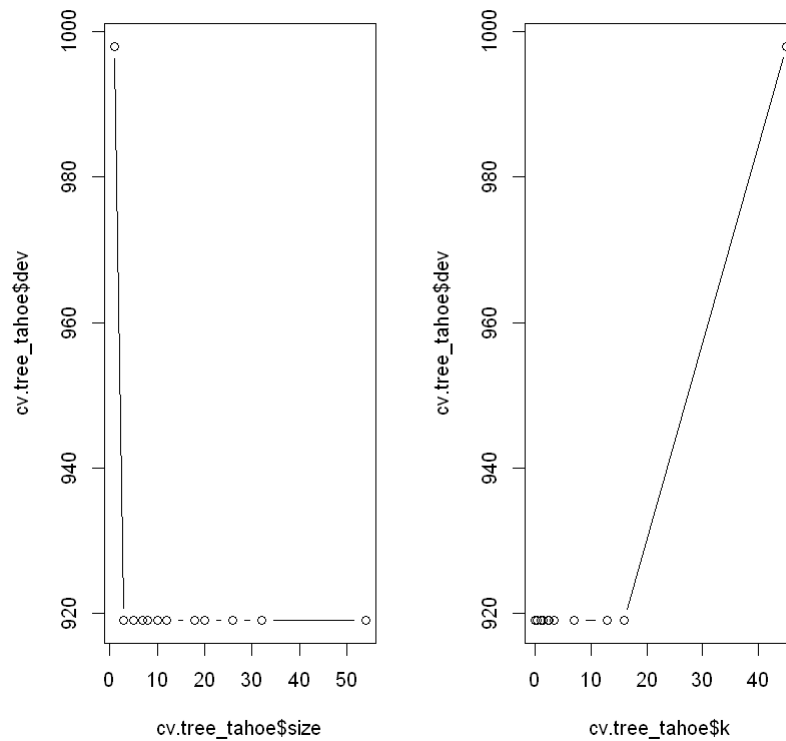
```
'size' 'dev' 'k' 'method'
```

```
In [165]: ▶ plot(cv.tree_tahoe)
```





```
In [166]: par(mfrow = c(1,2))  
#Here deviance is actually the number of misclassification  
plot(cv.tree_tahoe$size,cv.tree_tahoe$dev,type = "b")  
plot(cv.tree_tahoe$k,cv.tree_tahoe$dev,type = "b")
```



In [167]: ▶ `cv.tree_tahoe`

```
$size
[1] 54 32 26 20 18 12 10  8  7  5  3  1

$dev
[1] 919 919 919 919 919 919 919 919 919 919 919 998

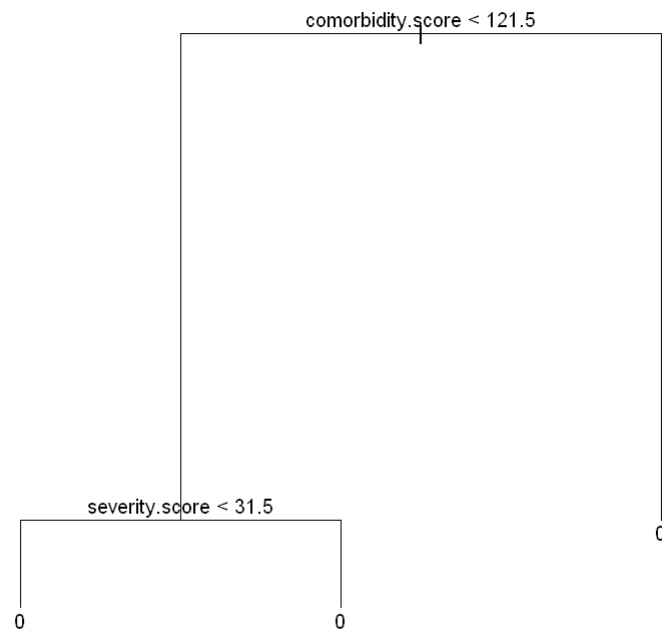
$k
[1]      -Inf  0.0000000  0.3333333  1.0000000  1.5000000  2.3333333
[7]  2.5000000  3.5000000  7.0000000 13.0000000 16.0000000 45.0000000

$method
[1] "misclass"

attr(,"class")
[1] "prune"          "tree.sequence"
```

In [168]: ▶ `prune.cv.tree_tahoe = prune.tree(tree_tahoe, best=3)`

```
In [169]: ▶ # final pruned tree  
plot(prune.cv.tree_tahoe)  
text(prune.cv.tree_tahoe, pretty = 0)
```



▼ (c)

In [170]: ▶ `prune.cv.tree_tahoe`

```
node), split, n, deviance, yval, (yprob)
      * denotes terminal node
```

```
1) root 4382 4702.0 0 ( 0.7723 0.2277 )
  2) comorbidity.score < 121.5 3084 2440.0 0 ( 0.8651 0.1349 )
    4) severity.score < 31.5 2423 1609.0 0 ( 0.8968 0.1032 ) *
    5) severity.score > 31.5 661 745.1 0 ( 0.7489 0.2511 ) *
  3) comorbidity.score > 121.5 1298 1786.0 0 ( 0.5516 0.4484 ) *
```

In [171]: ▶ *# predict(prune.cv.tree\_tahoe,newdata =tahoe)[,2] denotes the probability of readmit30 =1*  
*# so sum (predict(prune.cv.tree\_tahoe,newdata =tahoe)[,2]>0.15 returns the number of people exceeding thresh*  
`sum (predict(prune.cv.tree_tahoe,newdata =tahoe)[,2]>0.15)`

1959

**We could also get this number by choosing leaf node 5) 3) as they pass threshold 0.15, so the total number of people should receive CareTracker is 661 + 1298 = 1959.**

In [172]: ▶ `nrow(tahoe)`

4382


In [173]:  1959/4382

0.44705613874943

So the percentage is  $1959/4382 = 44.7\%$

▼ (d)

We pick demographic parity and unawareness as fairness measures.

In [174]:  *# demographic parity -  $P(C(X,A) = 1 | A = 1) = P(C(X,A) = 1 | A = 0)$*   
*# p1 denotes the probability of receiving CareTracker for females*  
*# p1 denotes the probability of receiving CareTracker for non-females*  
 number.females = sum(tahoe\$female==1)  
 number.non\_females = sum(tahoe\$female==0)  
 number.females.withCareTacker = sum(predict(prune.cv.tree\_tahoe,newdata =tahoe[tahoe\$female==1,])[,2]>0.15)  
 number.non\_females.withCareTacker = sum(predict(prune.cv.tree\_tahoe,newdata =tahoe[tahoe\$female==0,])[,2]>0.  
 p1 = number.females.withCareTacker/number.females  
 p2 = number.non\_females.withCareTacker / number.non\_females  
 print(c(p1,p2))  
 p1-p2

[1] 0.4815175 0.4165950

0.0649224968299481

We still have 6.49% gap between females and non-females

```
In [175]: ► # unawareness -  $P(C(X, A) = 1) = P(C(X) = 1)$ 
# p.XA denotes the probability of receiving CareTracker awaring of gender
# p.XA denotes the probability of receiving CareTracker without awareness of gender
tahoe2 <- tahoe
tahoe2$female <- c(rep(0, nrow(tahoe)))

number.withCareTacker.X_A = sum(predict(prune.cv.tree_tahoe,newdata =tahoe)[,2]>0.15)
number.withCareTacker.X = sum(predict(prune.cv.tree_tahoe,newdata = tahoe2)[,2]>0.15)

p.XA = number.withCareTacker.X_A / nrow(tahoe)
p.X = number.withCareTacker.X / nrow(tahoe)
print(c(p.XA,p.X))
p.XA - p.X
```

```
[1] 0.4470561 0.4470561
```

```
0
```

It seems fair.

▼ (e)

I think demographic parity is the most important fair measure since we do have gap and it is not fair, while unawareness is the least important since we do not even use `female` in the pruned tree.

## ▼ Problem 2

▼ (a)

```
In [109]: ► college <- read.csv("CollegeData.csv")
```

In [110]: `head(college)`

A data.frame: 6 × 10

	INSTNM	SAT_AVG	UGDS	COSTT4_A	TUITIONFEE_OUT	TUITFTE	AVGFACSAL	PFTFAC	C150_4	PFTFTUG1_EF
	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<dbl>
1	California Institute of Technology	1534	977	56382	41538	15679	16120	0.9570	0.9307	0.9725
2	University of Chicago	1504	5697	62425	47514	26409	16589	0.8076	0.9268	0.9834
3	Massachusetts Institute of Technology	1503	4510	57010	43498	28012	15617	0.9862	0.9307	0.9721
4	Harvard University	1501	7278	57950	42292	27867	17861	0.8595	0.9747	0.4143
5	Yale University	1497	5422	59320	44000	14701	16042	0.7281	0.9779	0.9777
6	Princeton University	1495	5234	55430	40170	13049	15711	0.8485	0.9694	1.0000

In [111]: `# drop na`  
`college <- na.omit(college)`

In [112]: `str(college)`

```
'data.frame':  1136 obs. of  10 variables:
 $ INSTNM      : chr  "California Institute of Technology" "University of Chicago" "Massachusetts Institute of Technology" "Harvard University" ...
 $ SAT_AVG     : int   1534 1504 1503 1501 1497 1495 1475 1474 1471 1466 ...
 $ UGDS        : int   977 5697 4510 7278 5422 5234 6794 6851 7970 6980 ...
 $ COSTT4_A    : int   56382 62425 57010 57950 59320 55430 59890 62594 61540 58408 ...
 $ TUITIONFEE_OUT: int   41538 47514 43498 42292 44000 40170 42978 44841 49138 43683 ...
 $ TUITFTE     : int   15679 26409 28012 27867 14701 13049 22796 23619 31924 23849 ...
 $ AVGFACSA    : int   16120 16589 15617 17861 16042 15711 12871 13286 15706 19862 ...
 $ PFTFAC      : num   0.957 0.808 0.986 0.86 0.728 ...
 $ C150_4      : num   0.931 0.927 0.931 0.975 0.978 ...
 $ PFTFTUG1_EF : num   0.973 0.983 0.972 0.414 0.978 ...
 - attr(*, "na.action")= 'omit' Named int [1:47] 22 28 36 145 177 187 236 240 257 259 ...
 ..- attr(*, "names")= chr [1:47] "22" "28" "36" "145" ...
```

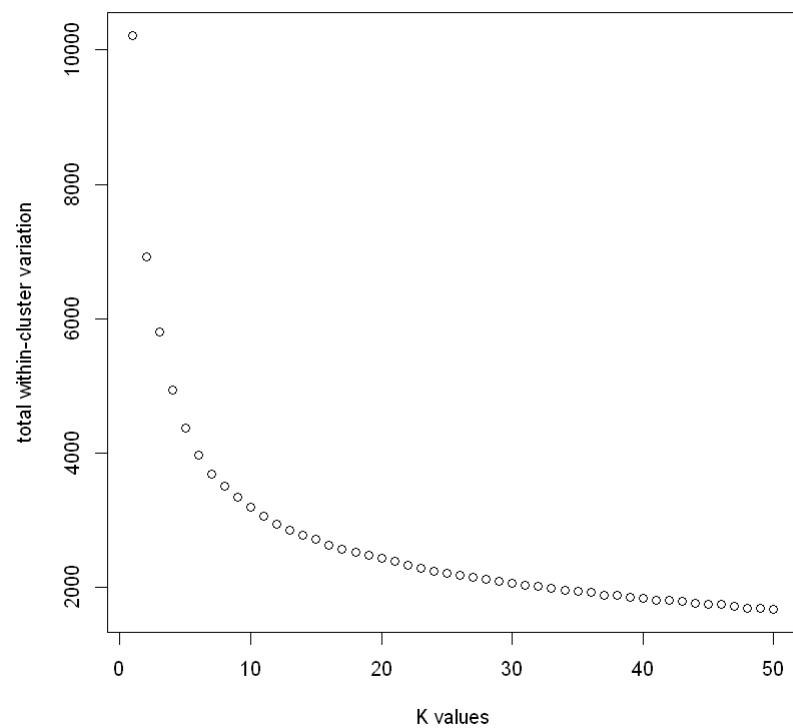
In [113]: `# We do not need INSTNM`  
`college <- subset(college, select = -`INSTNM`)`

In [114]: `college.scale <- scale(college, center = TRUE, scale = TRUE)`

In [117]: `k_range = c(1:50)`  
`total_variance = rep(0,50)`  
`for (k in 1:50) {`  
 `total_variance[k] <- kmeans(college.scale, centers = k, nstart = 10, iter.max = 15)$`tot.withinss``  
`}`



```
In [118]: plot(k_range, total_variance,  
               xlab = "K values", ylab = "total within-cluster variation")
```



**I will choose K to be 8.**



**(b)**

```
In [119]: ► college <- read.csv("CollegeData.csv")
```

```
In [120]: ► college <- na.omit(college)
```

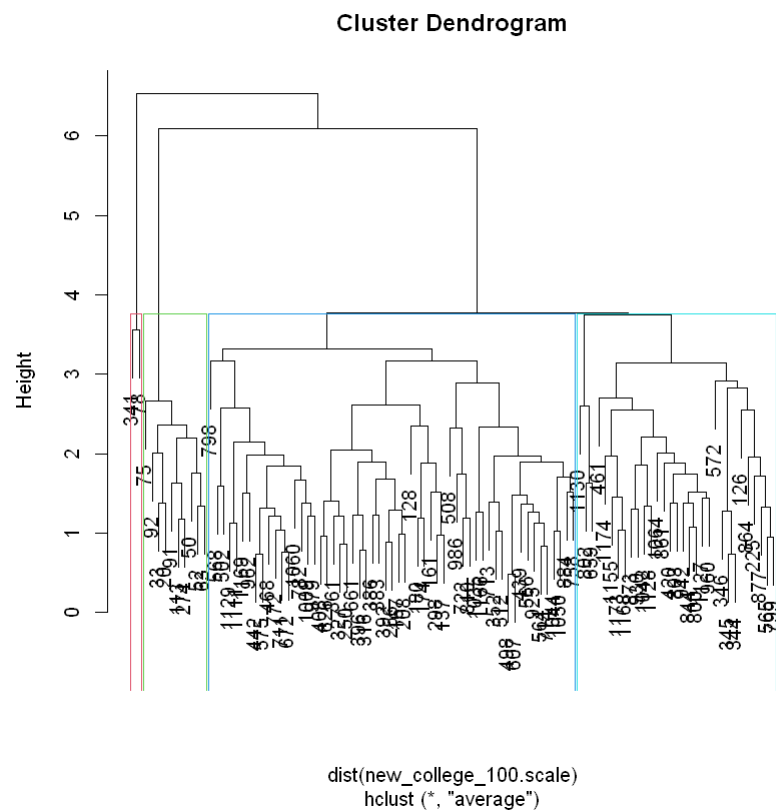
```
In [126]: ► new_college_100 <- head(college[order(college$INSTNM),],100)
```

```
In [127]: ► # We do not need INSTNM  
new_college_100 <- subset(new_college_100, select = -`INSTNM`)
```

```
In [128]: ► new_college_100.scale <- scale(new_college_100, center = TRUE, scale = TRUE)
```

```
In [129]: ► hc.average = hclust(dist(new_college_100.scale), method = "average")
```

```
In [133]: plot(hc.average)
rect.hclust(hc.average, k=4, border = 2:10)
```



```
In [136]: ▶ sub_grp <- cutree(hc.average, k=4)
table(sub_grp)
```

```
sub_grp
 1  2  3  4
57 31 10 2
```

```
In [137]: ▶ centroid = aggregate(new_college_100.scale, list(cluster = sub_grp), mean)
centroid
```

A data.frame: 4 × 10

cluster	SAT_AVG	UGDS	COSTT4_A	TUITIONFEE_OUT	TUITFTE	AVGFACSAL	PFTFAC	C150_4	PFTFTUG1_EF
<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	-0.08306239	-0.4146077	0.2707976	0.1438011	0.1294921	-0.3339513	-0.28611972	0.0569641	-0.005129142
2	-0.52724194	0.3899094	-1.0317358	-0.8218523	-0.8556620	-0.1712831	0.48946200	-0.7735935	-0.407261543
3	1.89324817	0.1741540	1.8830832	1.9493824	2.0525177	2.1897702	0.03087882	1.9178066	1.181724219
4	1.07328730	4.9019538	-1.1412418	-1.1065335	-0.6903518	1.2236488	0.41335689	0.7781886	0.550113358

Above are centroids. Each row is a centroid.

▼ (c)

For each number of clusters, We calculate total within sum of square. Then we plot number of clusters vs total within sum of square. After observing the plot, we use elbow method to determine number of clusters. More specifically, we could choose a ratio such as 5 %, whenever the difference between total within sum of square of consecutive k values divided by the difference of total within sum of square between k=2 and k=1 is less the 5 %, we should stop and use the last k.

