STAT 2450 Assignment #2

James MacPhee

Banner: B00768516

======================================================================

This exercise relates to the College data set, which can be found in the file at <http://www-bcf.usc.edu/~gareth/ISL/College.csv> It contains a collection of variables for a number of universities and colleges in the US. The variables are described in question 8, chapter 2, of *An Introduction to Statistical Learning*.

1. Use the *read.csv()* function to read the data into R. Call the data frame *college*. Use the *dim()* function to find the dimensions of the data set.

college=read.csv('http://www-bcf.usc.edu/~gareth/ISL/College.csv')  
dim(college)

## [1] 777 19

attach(college)

How many records are there in the data set? There are 777 records in the data set

1. Look at the data using the *fix()* or *edit()* function. You should notice that the first column is just the name of each university. We don’t really want R to treat this as data. However, it may be handy have these names for later. Try the following commands, which will copy the college names into row labels which are not part of the data set, but rather as labels, and then delete the associated column.

rownames(college)= college[,1]  
college= college[,2:19]  
college[1:2,]

## Private Apps Accept Enroll Top10perc  
## Abilene Christian University Yes 1660 1232 721 23  
## Adelphi University Yes 2186 1924 512 16  
## Top25perc F.Undergrad P.Undergrad Outstate  
## Abilene Christian University 52 2885 537 7440  
## Adelphi University 29 2683 1227 12280  
## Room.Board Books Personal PhD Terminal  
## Abilene Christian University 3300 450 2200 70 78  
## Adelphi University 6450 750 1500 29 30  
## S.F.Ratio perc.alumni Expend Grad.Rate  
## Abilene Christian University 18.1 12 7041 60  
## Adelphi University 12.2 16 10527 56

You should see that the first data column is Private. Note that another column of labels now appears before the Private column. This is not a data column but contains the labels that R is giving to each row. If you look at the data with *fix()* or *edit()*, youu will see that these labels are called “row.names”."

1. Use the *summary()* function to produce a numerical summary of the variables in the data set.

summary(college)

## Private Apps Accept Enroll Top10perc   
## No :212 Min. : 81 Min. : 72 Min. : 35 Min. : 1.00   
## Yes:565 1st Qu.: 776 1st Qu.: 604 1st Qu.: 242 1st Qu.:15.00   
## Median : 1558 Median : 1110 Median : 434 Median :23.00   
## Mean : 3002 Mean : 2019 Mean : 780 Mean :27.56   
## 3rd Qu.: 3624 3rd Qu.: 2424 3rd Qu.: 902 3rd Qu.:35.00   
## Max. :48094 Max. :26330 Max. :6392 Max. :96.00   
## Top25perc F.Undergrad P.Undergrad Outstate   
## Min. : 9.0 Min. : 139 Min. : 1.0 Min. : 2340   
## 1st Qu.: 41.0 1st Qu.: 992 1st Qu.: 95.0 1st Qu.: 7320   
## Median : 54.0 Median : 1707 Median : 353.0 Median : 9990   
## Mean : 55.8 Mean : 3700 Mean : 855.3 Mean :10441   
## 3rd Qu.: 69.0 3rd Qu.: 4005 3rd Qu.: 967.0 3rd Qu.:12925   
## Max. :100.0 Max. :31643 Max. :21836.0 Max. :21700   
## Room.Board Books Personal PhD   
## Min. :1780 Min. : 96.0 Min. : 250 Min. : 8.00   
## 1st Qu.:3597 1st Qu.: 470.0 1st Qu.: 850 1st Qu.: 62.00   
## Median :4200 Median : 500.0 Median :1200 Median : 75.00   
## Mean :4358 Mean : 549.4 Mean :1341 Mean : 72.66   
## 3rd Qu.:5050 3rd Qu.: 600.0 3rd Qu.:1700 3rd Qu.: 85.00   
## Max. :8124 Max. :2340.0 Max. :6800 Max. :103.00   
## Terminal S.F.Ratio perc.alumni Expend   
## Min. : 24.0 Min. : 2.50 Min. : 0.00 Min. : 3186   
## 1st Qu.: 71.0 1st Qu.:11.50 1st Qu.:13.00 1st Qu.: 6751   
## Median : 82.0 Median :13.60 Median :21.00 Median : 8377   
## Mean : 79.7 Mean :14.09 Mean :22.74 Mean : 9660   
## 3rd Qu.: 92.0 3rd Qu.:16.50 3rd Qu.:31.00 3rd Qu.:10830   
## Max. :100.0 Max. :39.80 Max. :64.00 Max. :56233   
## Grad.Rate   
## Min. : 10.00   
## 1st Qu.: 53.00   
## Median : 65.00   
## Mean : 65.46   
## 3rd Qu.: 78.00   
## Max. :118.00

1. You will note that all variables are quantitative, except for “Private”, which is a factor variable taking the values “Yes” and “No”. It may be more convenient to work with numeric variables.
2. Use the “ifelse” command to define a new variable Private2 which takes the value 1 if Private is “Yes”, and otherwise takes the value 0, as follows:

“Private2=ifelse(Private==”Yes“, value 1 here, value 2 here)”

Private2=ifelse(Private=="Yes",1,0)

4(b ) Make a two way frequency table of Private vs Private2, as

“table(Private,Private2)”.

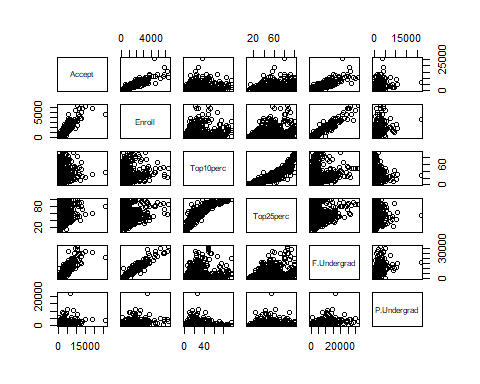
table(Private,Private2)

## Private2  
## Private 0 1  
## No 212 0  
## Yes 0 565

This will tell you whether the “ifelse” command worked. Do Private and Private 2 carry the same information? Yes, except the labels are different, no=0 and yes=1.

1. Use the pairs() function to produce a scatterplot matrix of the columns 3 through 8 of the data. Recall that you can reference columns 3-8 of A using A[,3:8].

pairs(college[,3:8])



1. Create a new qualitative variable, called Elite, by binning the Top10perc variable. We are going to divide universities into two groups based on whether or not the proportion of students coming from the top 10 % of their high school classes exceeds 50%.

Elite=rep(0,nrow(college))  
Elite[college$Top10perc >50]=1

1. Use the *ifelse()* command to calculate a variable Elite2, which should be indentical to Elite.

Elite2=ifelse(Elite==1,1,0)

1. Use a for loop, and the  
   *if(condition){statement 1} else {statement 2}* construct, to define a variable Elite3, which should be indentical to Elite.

N=dim(college)[1]  
Elite3=rep(NA,N)  
for (i in 1:N){  
 if(college$Top10perc[i] >50){   
 Elite3[i]=1  
 }   
 else{   
 Elite3[i]=0  
 }  
}

1. Use the table command to create two way frequency tables of Elite vs Elite2 and Elite vs Elite3. Then print the tables.

table(Elite,Elite2)

## Elite2  
## Elite 0 1  
## 0 699 0  
## 1 0 78

table(Elite,Elite3)

## Elite3  
## Elite 0 1  
## 0 699 0  
## 1 0 78

Do the three variables carry the same information? Yes all three variables carry the same information. You can tell this because there is zeroes in the spots where the labels of each variable don’t match up.

1. There are subtle difference in the way that character and factor variables are treated. Use an *ifelse()* statement to define a character variable Elitec which takes the value “Yes” if the school is an elite school, and otherwise “No”. Then convert the character variable to a factor variable, using the command *Elitef=as.factor(Elitec)*. Use the command “table(Elitec,Elitef)” to see if the character and factor variables carry the same information. Use the *summary()* function on each of Elitec and Elitef to see if they are treated the same. Similarly, use *boxplot(Outstate~Elitef)* and/or *boxplot(Outstate~Elitec)* to see whether *boxplot()* accommodates character and/or factor variables.

Elitec=ifelse(Elite==1,1,0)  
Elitef=as.factor(Elitec)  
table(Elitec,Elitef)

## Elitef  
## Elitec 0 1  
## 0 699 0  
## 1 0 78

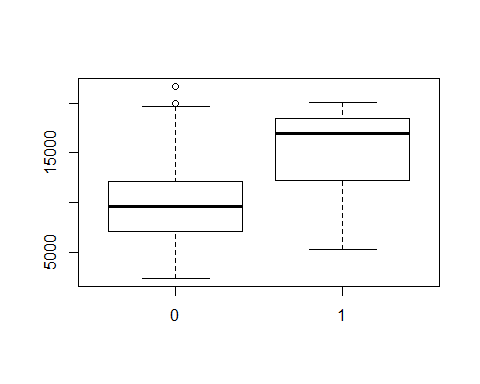
summary(Elitec)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.1004 0.0000 1.0000

summary(Elitef)

## 0 1   
## 699 78

boxplot(Outstate~Elitef)  
boxplot(Outstate~Elitec)



Do Elitec and Elitef carry the same information? Do the *summary()* and *boxplot()* functions work with the character form? They do carry the same information, but store it in a different data type. when calling summary() on character variable it simply prints out the values. But when summary() is called on a factor variable it prints out measures of central tendency and variability.

1. To further explore the *if(){} else{}*" construct, try the following:

“if(college$Top10perc>50){Elite4=1} else {Elite4=0}”

if(college$Top10perc>50){Elite4=1} else {Elite4=0}

## Warning in if (college$Top10perc > 50) {: the condition has length > 1 and  
## only the first element will be used

Elite4

## [1] 0

What went wrong? Explore the help info on ifelse and if. Better yet, make up some of your own examples to see how they work. What went wrong is that ‘if’ statements aren’t vectorized, meaning the condition variable (Elite4) is a vector but the if statement only works with assigning one value so simply uses the first one. ‘ifelse’ statements are vectorized so in this scenario it would be best to either use an ‘ifelse’ or but the ‘if’ inside a loop.

How do the *ifelse()* and *if(){ } else {}* constructs differ? See previous answer. They differ in the main way that ifelse() is vectorized and the other isn’t.

1. A student wishes to find colleges for which

((the student faculty ratio S.F.Ratio is less than 14) OR (the proportion of teachers with PhD’s , PhD, is greater than 85)) AND (the graduation rate, Grad.Rate, is greater than 70%).

By using a suitable combination of AND (&), OR (|), < and > operators, create a new data frame college2 which contains all such colleges. How many colleges in the data set satisfy the specified conditions? See below

college2 = college[(S.F.Ratio<14|PhD>85)&Grad.Rate>70,]  
#I changed the next line because I didn't think you would want me to hand in 10 pages of universities names and stats.  
college2[1:2,]

## Private Apps Accept Enroll Top10perc Top25perc  
## Albion College Yes 1899 1720 489 37 68  
## Albright College Yes 1038 839 227 30 63  
## F.Undergrad P.Undergrad Outstate Room.Board Books  
## Albion College 1594 32 13868 4826 450  
## Albright College 973 306 15595 4400 300  
## Personal PhD Terminal S.F.Ratio perc.alumni Expend  
## Albion College 850 89 100 13.7 37 11487  
## Albright College 500 79 84 11.3 23 11644  
## Grad.Rate  
## Albion College 73  
## Albright College 80

How many colleges meet the specified conditions? 230 colleges out of 777 meet the conditions.