Tree Methods Project - Solutions

We'll make use of the Tree based methods to classify schools as Private or Public based off their features

Let's start by getting the data which is included in the ISLR library, the College data frame.

A data frame with 777 observations on the following 18 variables.

- Private A factor with levels No and Yes indicating private or public university
- Apps Number of applications received
- Accept Number of applications accepted
- Enroll Number of new students enrolled
- Top10perc Pct. new students from top 10% of H.S. class
- Top25perc Pct. new students from top 25% of H.S. class
- F.Undergrad Number of fulltime undergraduates
- P.Undergrad Number of parttime undergraduates
- Outstate Out-of-state tuition
- Room.Board Room and board costs
- Books Estimated book costs
- Personal Estimated personal spending
- PhD Pct. of faculty with Ph.D.'s
- Terminal Pct. of faculty with terminal degree
- S.F.Ratio Student/faculty ratio
- perc.alumni Pct. alumni who donate
- Expend Instructional expenditure per student
- Grad.Rate Graduation rate

Get the Data

Call the ISLR library and check the head of College (a built-in data frame with ISLR, use data() to check this.) Then reassign College to a dataframe called df

In [1]:
```
library(ISLR)
```

In [2]:
```
head(College)
```

Out[2]:

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergra
Abilene Christian University	Yes	1660	1232	721	23	52	2885	537
Adelphi University	Yes	2186	1924	512	16	29	2683	1227
Adrian College	Yes	1428	1097	336	22	50	1036	99
Agnes Scott College	Yes	417	349	137	60	89	510	63
Alaska Pacific University	Yes	193	146	55	16	44	249	869
Albertson College	Yes	587	479	158	38	62	678	41

In [3]:
```
df<-College
```

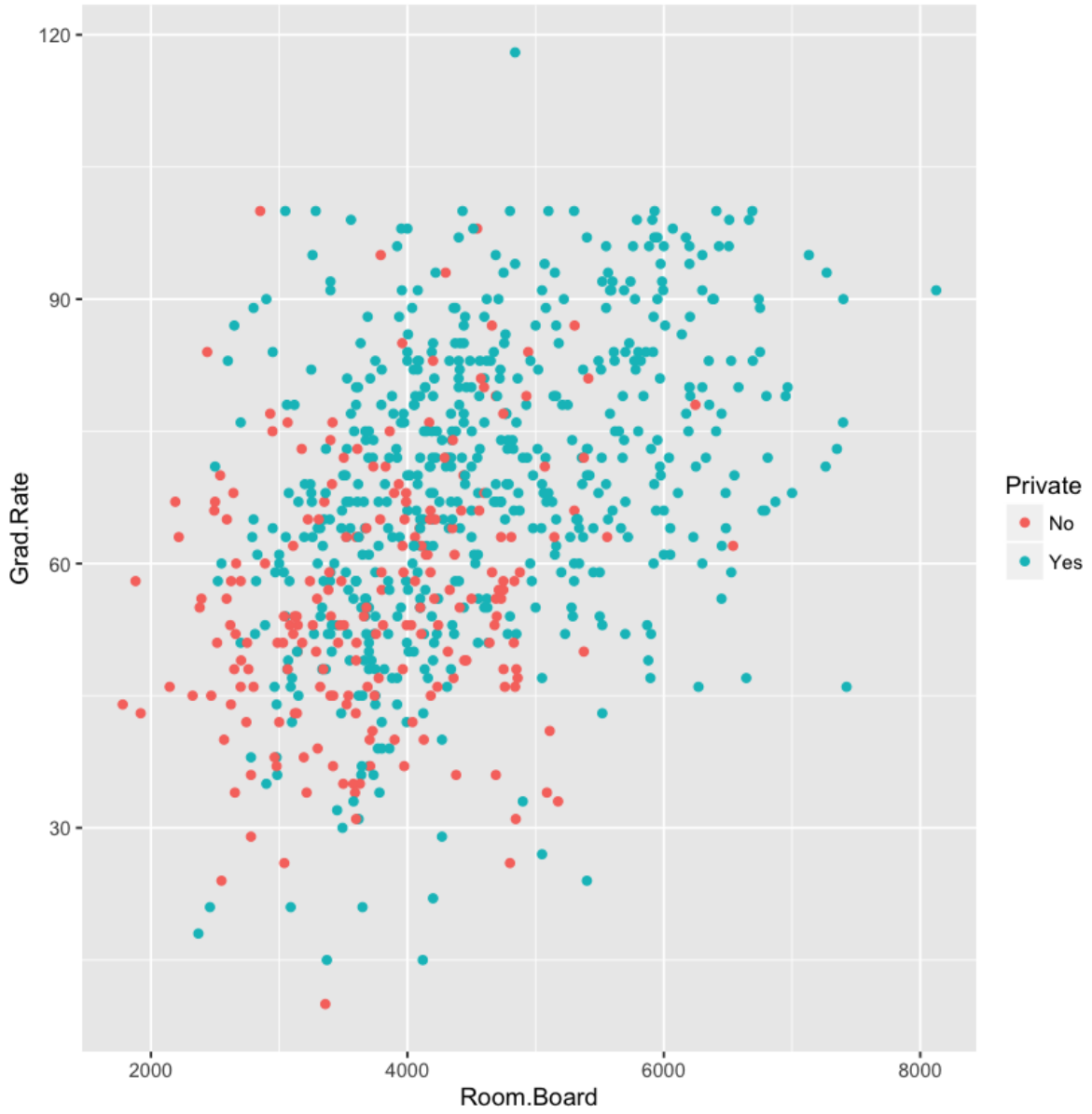
EDA

Let's explore the data!

Create a scatterplot of Grad.Rate versus Room.Board, colored by the Private column.

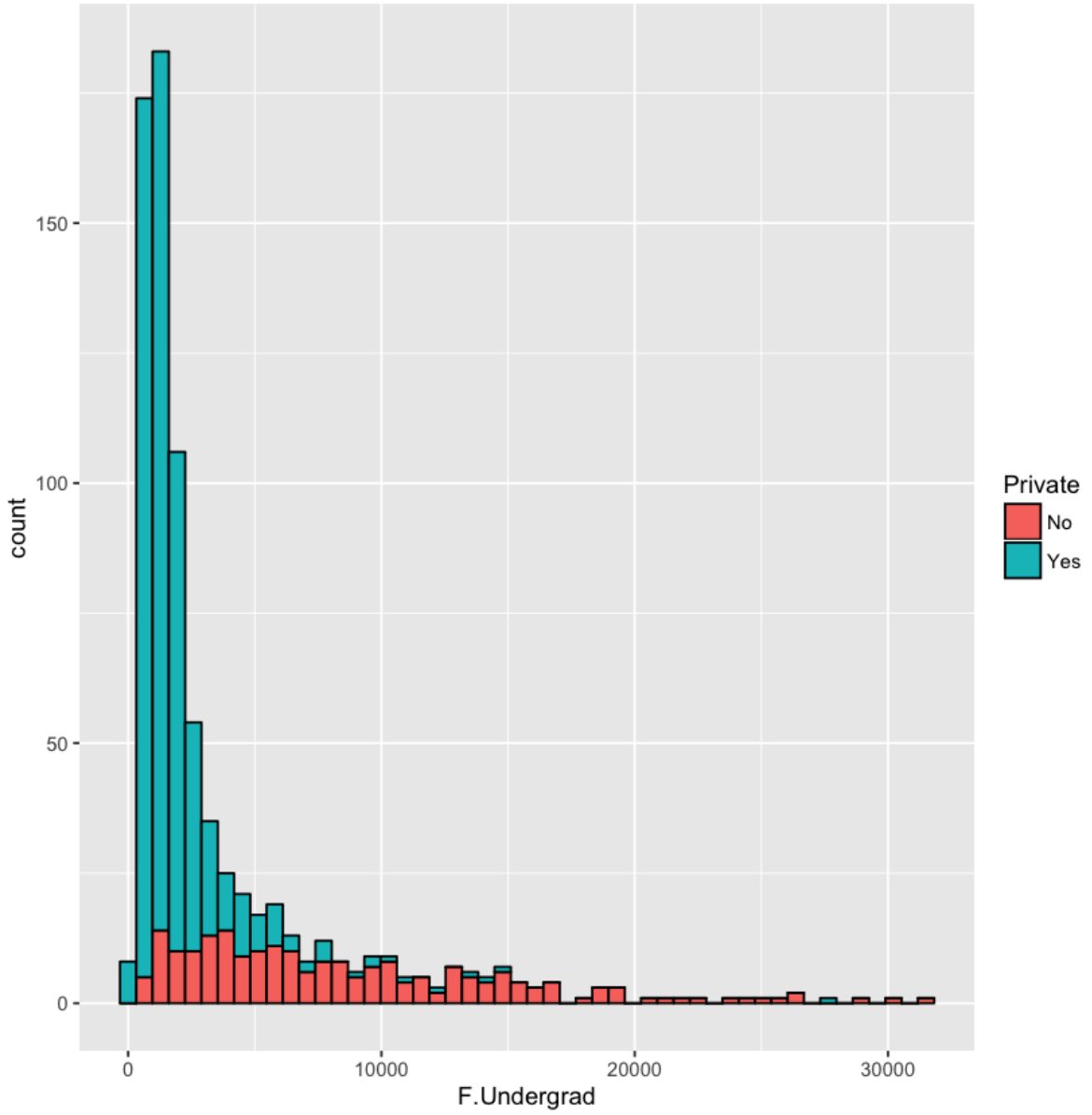
In [4]:
```
library(ggplot2)
```

In [5]:
```
ggplot(df,aes(Room.Board,Grad.Rate)) + geom_point(aes(color=Private))
```



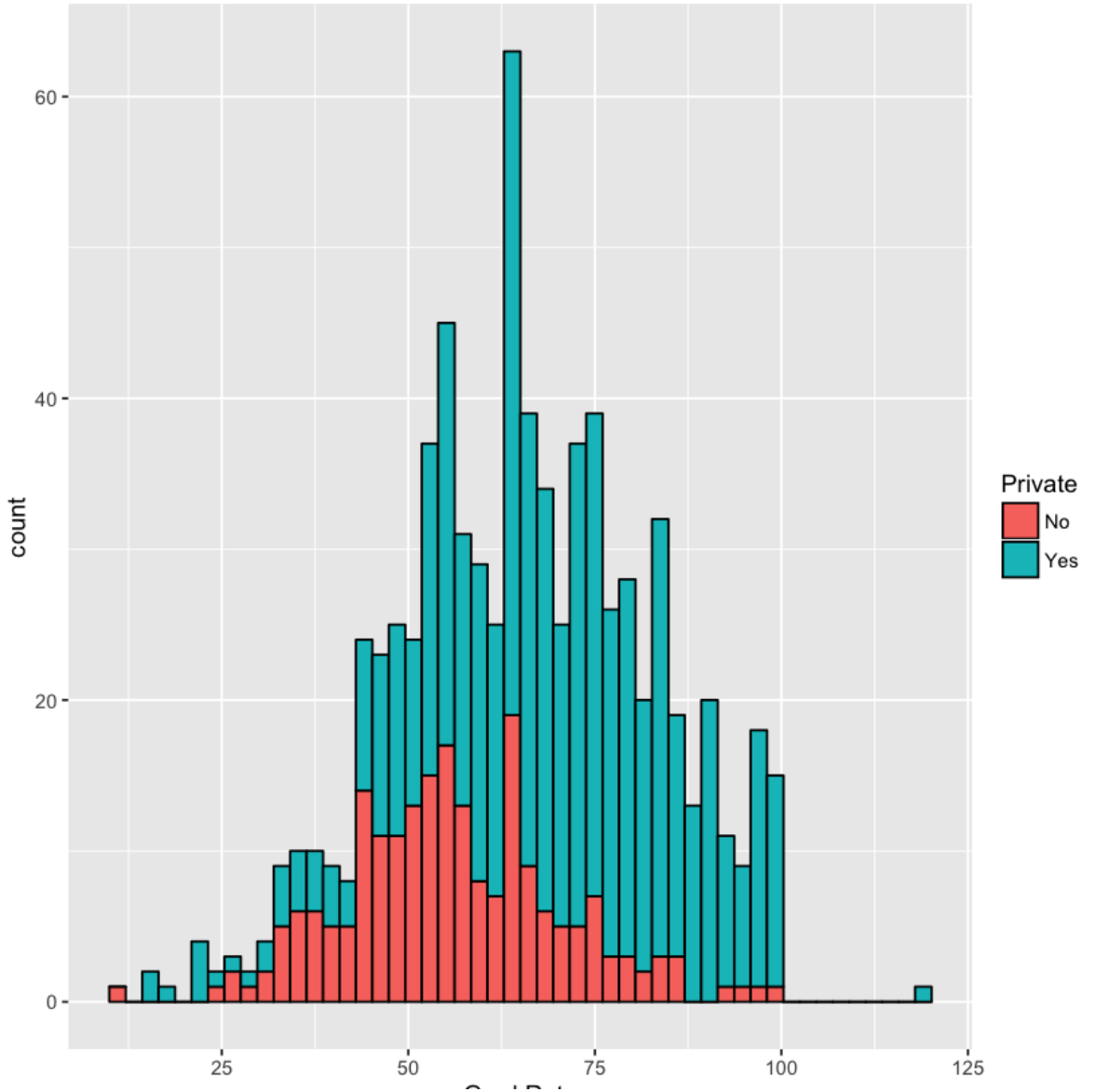
Create a histogram of full time undergrad students, color by Private.

In [6]:
```
ggplot(df,aes(F.Undergrad)) + geom_histogram(aes(fill=Private),color='b
lack',bins=50)
```



Create a histogram of Grad.Rate colored by Private. You should see something odd here.

In [7]:
```
ggplot(df,aes(Grad.Rate)) + geom_histogram(aes(fill=Private),color='bla
ck',bins=50)
```



Grad.Rate

What college had a Graduation Rate of above 100% ?

In [8]:
```
subset(df,Grad.Rate > 100)
```

Out[8]:

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergra
Cazenovia College	Yes	3847	3433	527	9	35	1010	12

Change that college's grad rate to 100%

In [9]:
```
df['Cazenovia College','Grad.Rate'] <- 100
```

Train Test Split

Split your data into training and testing sets 70/30. Use the caTools library to do this.

In [10]:
```
library(caTools)

set.seed(101)

sample = sample.split(df$Private, SplitRatio = .70)
train = subset(df, sample == TRUE)
test = subset(df, sample == FALSE)
```

Decision Tree

Use the rpart library to build a decision tree to predict whether or not a school is Private. Remember to only build your tree off the training data.

In [11]:
```
library(rpart)
```

In [12]:
```
tree <- rpart(Private ~.,method='class',data = train)
```

Use predict() to predict the Private label on the test data.

In [26]:
```
tree.preds <- predict(tree,test)
```

Check the Head of the predicted values. You should notice that you actually have two columns with the probabilities.

In [27]:
```
head(tree.preds)
```

Out[27]:

	No	Yes
Adrian College	0.003311258	0.996688742
Alfred University	0.003311258	0.996688742
Allegheny College	0.003311258	0.996688742
Allentown Coll. of St. Francis de Sales	0.003311258	0.996688742
Alma College	0.003311258	0.996688742
Amherst College	0.003311258	0.996688742

Turn these two columns into one column to match the original Yes/No Label for a Private column.

In [32]:
```
tree.preds <- as.data.frame(tree.preds)
# Lots of ways to do this
joiner <- function(x){
    if (x>=0.5){
        return('Yes')
    }else{
```

```
        return("No")
    }
}
```

In [33]:
```
tree.preds$Private <- sapply(tree.preds$Yes,joiner)
```

In [34]:
```
head(tree.preds)
```

Out[34]:

	No	Yes	Private
Adrian College	0.003311258	0.9966887	Yes
Alfred University	0.003311258	0.9966887	Yes
Allegheny College	0.003311258	0.9966887	Yes
Allentown Coll. of St. Francis de Sales	0.003311258	0.9966887	Yes
Alma College	0.003311258	0.9966887	Yes
Amherst College	0.003311258	0.9966887	Yes

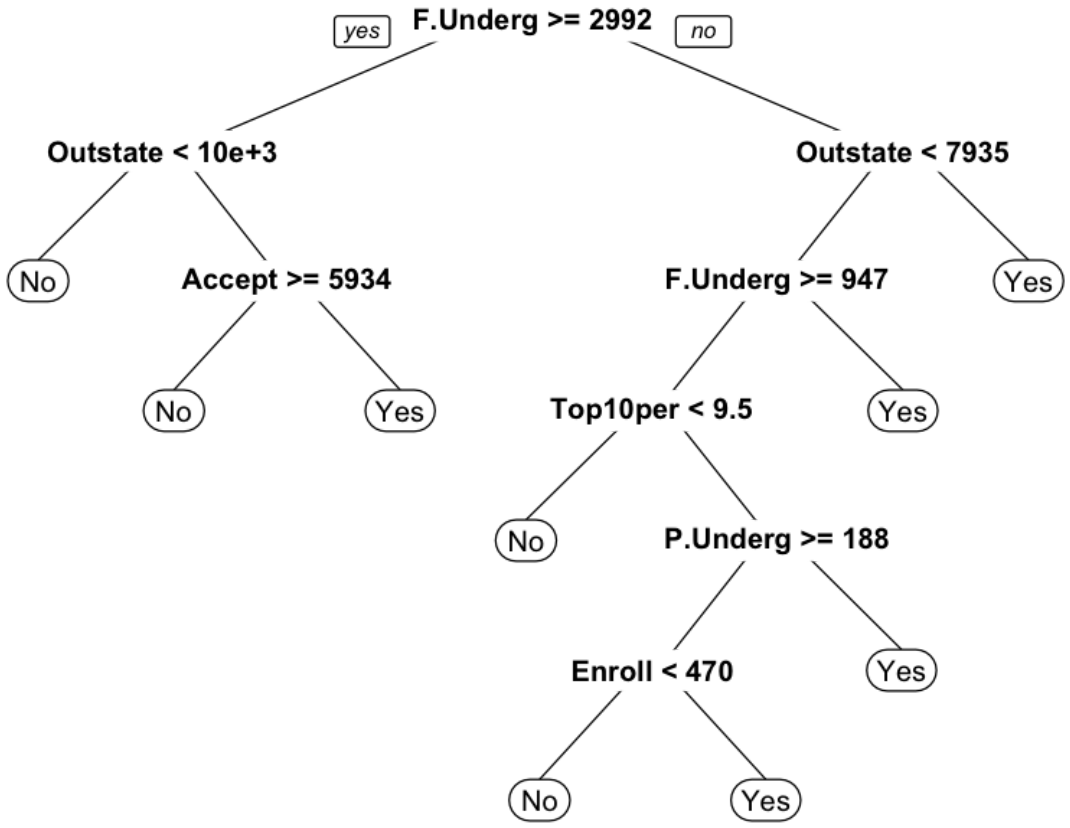
Now use table() to create a confusion matrix of your tree model.

In [37]:
```
table(tree.preds$Private,test$Private)
```

Out[37]:
```
     No Yes
No   57   9
Yes   7 160
```

Use the rpart.plot library and the prp() function to plot out your tree model.

In [38]:
```
library(rpart.plot)
prp(tree)
```



Random Forest

Now let's build out a random forest model!

Call the randomForest package library

In [39]:
```
library(randomForest)
```

Now use randomForest() to build out a model to predict Private class. Add importance=TRUE as a parameter in the model. (Use help(randomForest) to find out what this does.

In [42]:
```
rf.model <- randomForest(Private ~ . , data = train,importance = TRUE)
```

What was your model's confusion matrix on its own training set? Use model$confusion.

In [47]:
```
rf.model$confusion
```

Out[47]:

	No	Yes	class.error
No	128.0000000	20.0000000	0.1351351
Yes	11.00000000	385.00000000	0.02777778

Grab the feature importance with model$importance. Refer to the reading for more info on what Gini[1] means.[2]

In [50]:
```
rf.model$importance
```

Out[50]:

	No	Yes	MeanDecreaseAccuracy	MeanDecreaseGini
Apps	0.02728403	0.01548610	0.01855833	11.12145164
Accept	0.02758290	0.01376933	0.01753925	11.90902939
Enroll	0.03543096	0.02850524	0.03019823	21.78309626
Top10perc	0.012074927	0.004876602	0.006829200	5.878231682

	No	Yes	MeanDecreaseAccuracy	MeanDecreaseGini
Top25perc	0.005939966	0.004574398	0.004956138	4.636132072
F.Undergrad	0.14066600	0.06924742	0.08876258	37.37014532
P.Undergrad	0.048479660	0.005760397	0.017361364	15.537621126
Outstate	0.14299823	0.06438651	0.08555537	41.35682991
Room.Board	0.01640514	0.01315531	0.01407650	12.21199808
Books	0.0018960791	-0.0002561684	0.0003830364	2.2014363026
Personal	0.004212604	0.001688864	0.002336789	3.651182123
PhD	0.010413383	0.005385062	0.006756072	4.651126830
Terminal	0.003424653	0.004307162	0.004127311	3.996871039
S.F.Ratio	0.034579932	0.008941163	0.015922849	16.901749643
perc.alumni	0.023282086	0.002815539	0.008356919	4.969591602
Expend	0.02378523	0.01147495	0.01485984	10.16450065
Grad.Rate	0.01425191	0.00554695	0.00787915	6.75926369

Predictions

Now use your random forest model to predict on your test set!

In [51]:
```
p <- predict(rf.model,test)
```

In [52]:
```
table(p,test$Private)
```

Out[52]:
```
p       No Yes
  No    57   5
  Yes    7 164
```

It should have performed better than just a single tree, how much better depends on whether you are emasuring recall, precision, or accuracy as the most important measure of the model.

** Do make use of other Data Sets from UCL Maachine Learning Repository!