Decision Trees and Random Forests report

▼ **Attribute**

use of tree 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

```
library(ISLR)

df <- College
head(df)
```

EDA

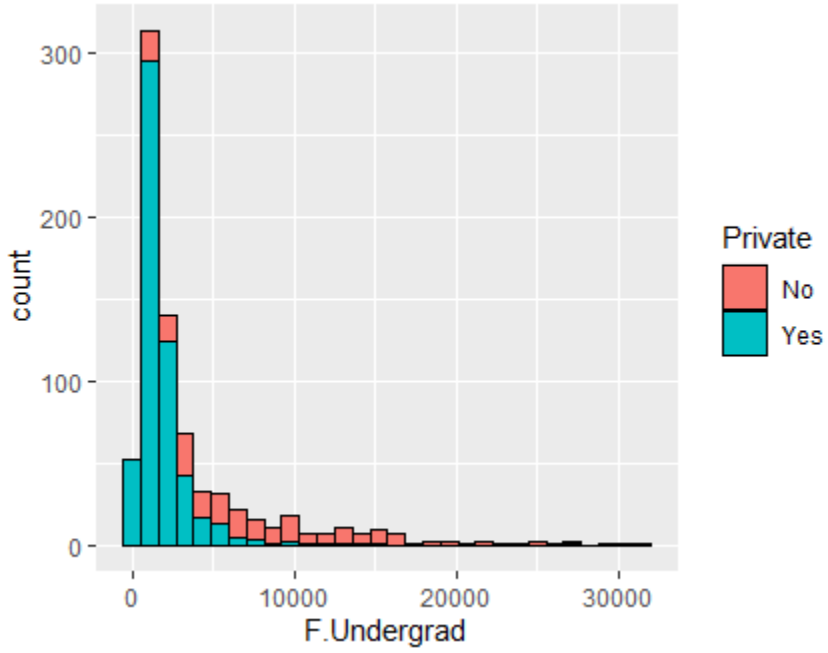
→ Create a scatterplot of Grad.Rate versus Room.Board, colored by the Private column

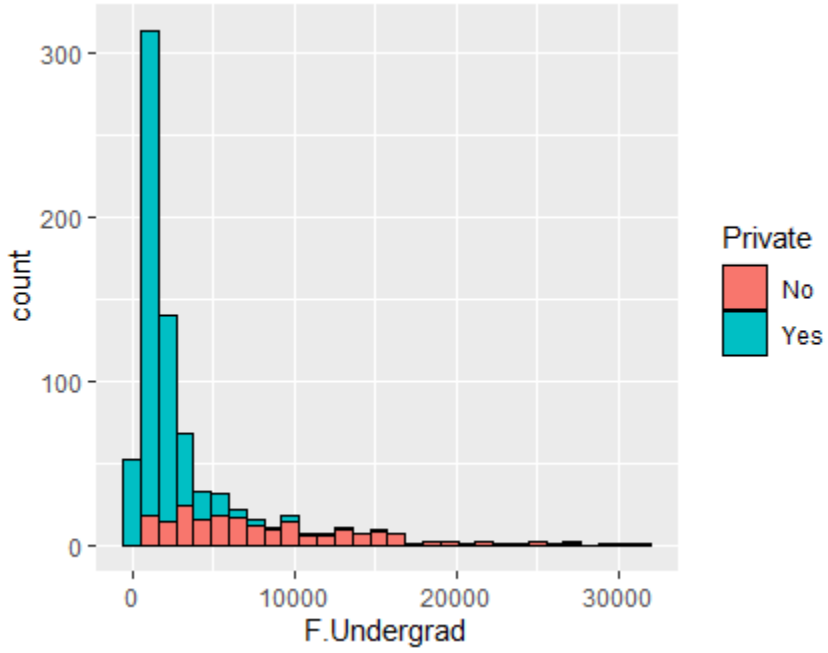
```
pl <- ggplot(df, aes(y=Grad.Rate,x=Room.Board, color = Private))
pl + geom_point(size = 2)
```

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

```
pl2 <- ggplot(df, aes(F.Undergrad, fill=Private), order=(Private))
pl2 + geom_histogram(color = "black")
```

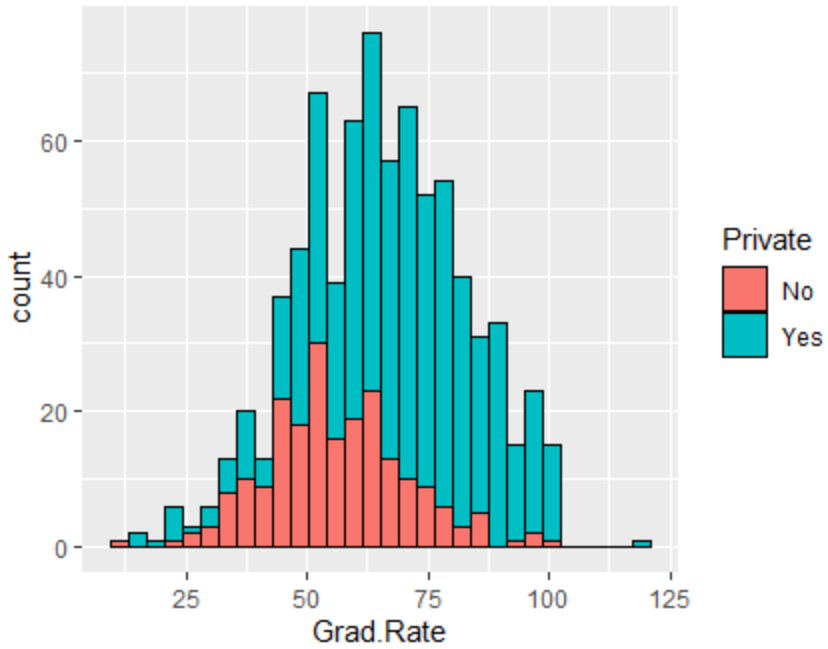
```
# position = position_stack(reverse = TRUE)
pl2 <- ggplot(df, aes(F.Undergrad, fill=Private), order=(Private))
pl2 + geom_histogram(color = "black", position = position_stack(reverse = TRUE))
```





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

```
pl3 <- ggplot(df, aes(Grad.Rate, fill=Private))
pl3 + geom_histogram(color = "black", position = position_stack(reverse = TRUE))
```



→ What college had a Graduation Rate of above 100% ?

```
# Change that college's grad rate to 100%
df[df$Grad.Rate > 100,]
df[df$Grad.Rate > 100, "Grad.Rate"] <- 100
```

Train Test Split

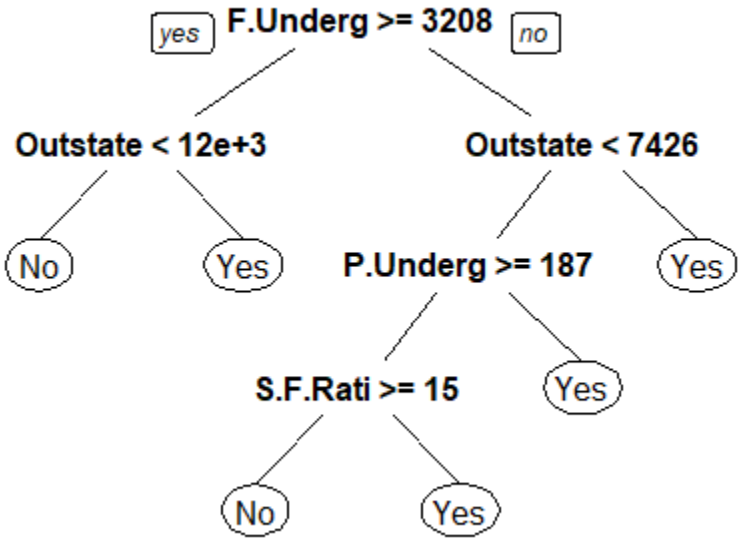
```
library(caTools)

set.seed(24)
split = sample.split(df, SplitRatio = 0.70)
train = subset(df, split == TRUE)
test = subset(df, split == FALSE)
```

Decision Tree model

```
library(rpart)

tree <- rpart(Private ~ . , method='class', data= train)
```

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

```
library(rpart.plot)

p <-predict(tree, newdata = test)
# Turn these two columns into one column to match the original
p$Private <- ifelse(p$Yes >= 0.5 , "Yes", "No")
table(test$Private,p$Private)
prp(tree)

# No Yes
#  No   64   9
#  Yes  15 171
```



Random Forest

```
# Random Forest
model <- randomForest(Private ~ .,data= train, importance=TRUE)
# confusion matrix
model$confusion
```

```
# importance
model$importance
# predict model randomForest
prf <- predict(model, newdata = test)
head(prf)
table(prf,test$Private)

# prf    No Yes
#   No   63   7
#   Yes  10 179
```