1. 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.

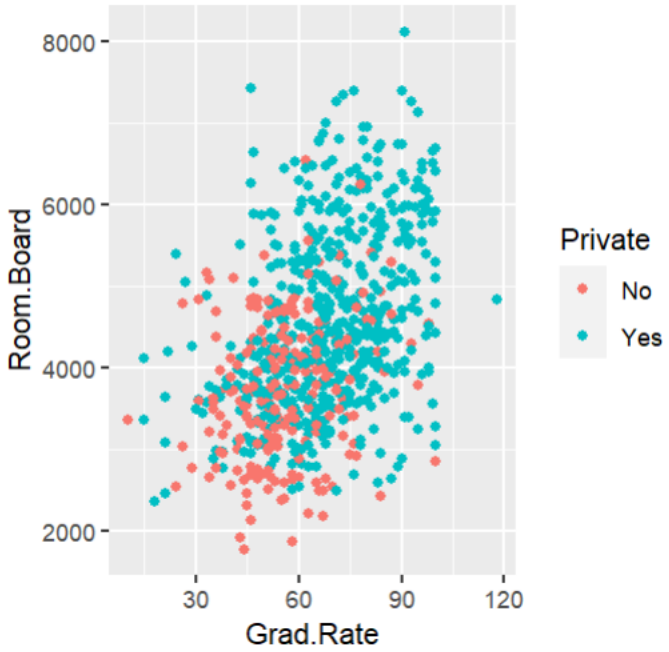
```
install.packages('ISLR')
install.packages('tidyverse')
install.packages('caTools')
install.packages('rpart')
install.packages('rpart.plot')

library(ISLR)
library(tidyverse)
library(caTools)
library(rpart)
library(rpart.plot)

head(College)
df <- College
```

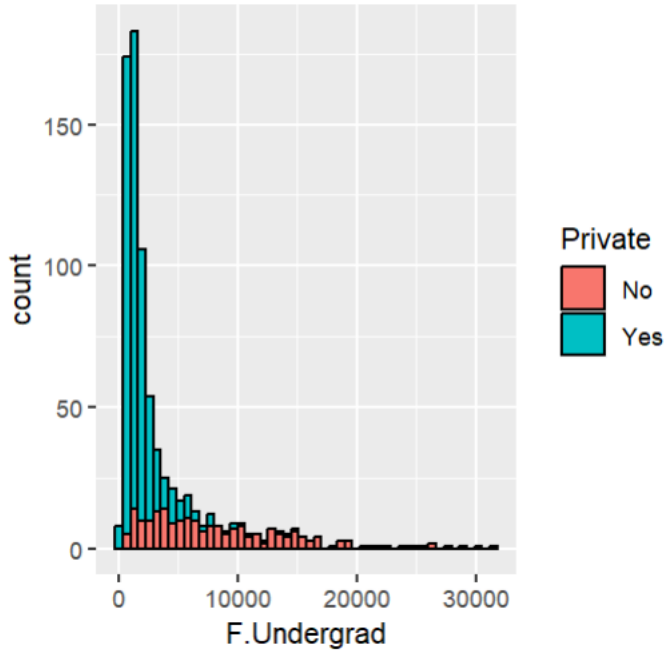
2. Create a scatterplot of Grad.Rate versus Room.Board, colored by the Private column.

```
df %>% ggplot(aes(x=Grad.Rate,y=Room.Board,color=Private))+geom_point()
```



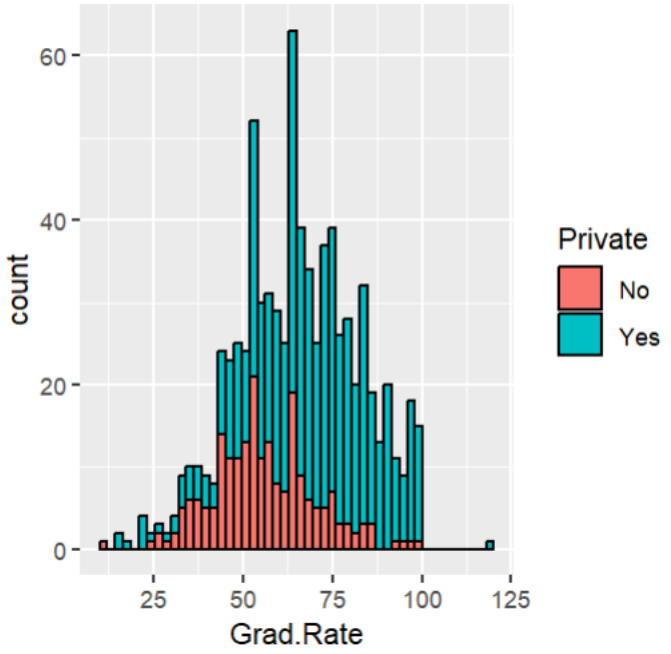
3. Create a histogram of full time undergrad students, color by Private.

```
df %>%
 ggplot(aes(x=F.Undergrad,fill=Private))+geom_histogram(bins =
    50,color="black",position=position_stack(reverse=TRUE))
```



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

```
df %>%
 ggplot(aes(x=Grad.Rate,fill=Private))+geom_histogram(bins =
    50,color="black",position=position_stack(reverse=TRUE))
```



5. What college had a Graduation Rate of above 100% ?

```
df["Cazenovia College","Grad.Rate"] <- 100
```

6. Change that college's grad rate to 100%.

```
row.names(df[df$Grad.Rate>100,])
```

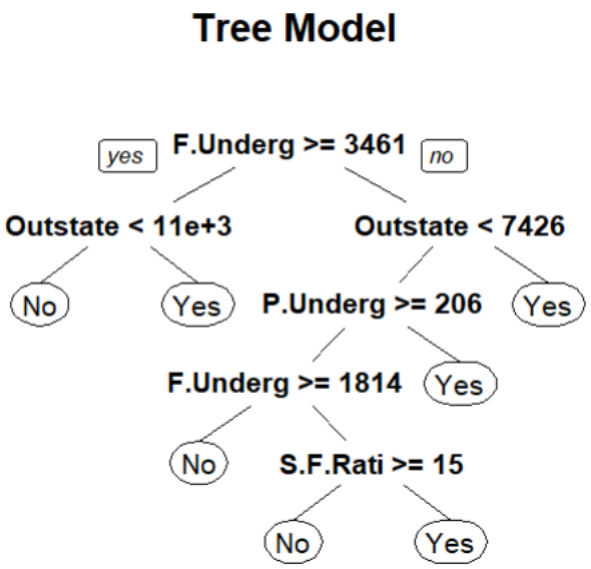
7. Train Test Split:

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

```
seed <- 103
set.seed(seed)
sample <- sample.split(df$Private, SplitRatio = 0.70)
train = subset(df, sample == TRUE)
test = subset(df, sample == FALSE)
```

8. 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.

```
tree_model <- rpart(Private ~ . , method='class', data= train)
prp(tree_model,main="Tree Model")
```



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

```
label <- "Private"
predicted_values <- predict(tree_model, test[,-which(colnames(test)==label)])
```

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

```
head(predicted_values)
```

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

```
test$pred_tree <- predict(tree_model, test[,-which(colnames(test)==label)])[,"Yes"]
classification_point <- 0.5
test$predclass_tree <- ifelse(test$pred_tree>classification_point,'Yes','No')
```

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

```
table(test$Private,test$predclass_tree)
```

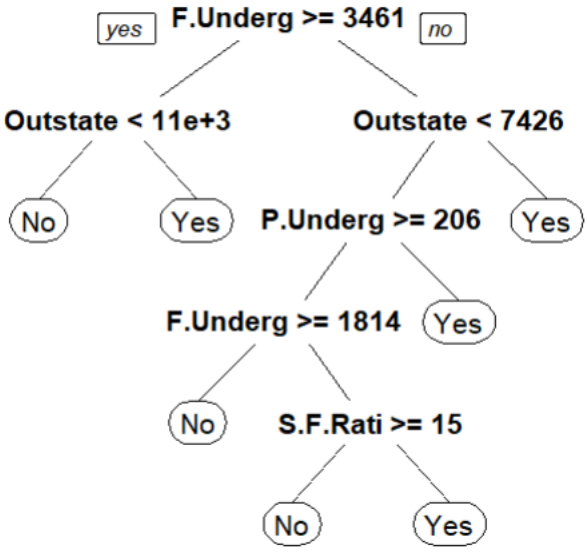
13. Use the rpart.plot library and the prp() function to plot out your tree model.

```
prp(tree_model)
```

14. Random Forest:
    Call the randomForest package library.
    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.

```
install.packages('randomForest')
library(randomForest)

set.seed(seed)
rf_base_model <- randomForest(Private ~ ., data=train, importance= TRUE)
```



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

```
rf_base_model$confusion
```

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

```
rf_base_model$importance
```

17. Predictions:
    Now use your random forest model to predict on your test set!

```
test$pred_class_RF <- predict(rf_base_model,newdata = test[,-
      which(colnames(test)==label)])
table(test$Private,test$pred_class_RF)
```