College_random_forest

Nabil Momin

2024-06-10

```
library(corrgram)
library(corrplot)

## corrplot 0.92 loaded

library(caTools)
library(Amelia)

## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.2, built: 2024-04-10)
## ## Copyright (C) 2005-2024 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##

library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(rpart)
library(rpart.plot)
library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(ISLR)
```

```
#### College data is inside ISLR
```

```
str(College)
```

```
## 'data.frame':    777 obs. of  18 variables:
##  $ Private    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Apps       : num  1660 2186 1428 417 193 ...
##  $ Accept     : num  1232 1924 1097 349 146 ...
##  $ Enroll     : num  721 512 336 137 55 158 103 489 227 172 ...
##  $ Top10perc  : num  23 16 22 60 16 38 17 37 30 21 ...
##  $ Top25perc  : num  52 29 50 89 44 62 45 68 63 44 ...
##  $ F.Undergrad: num  2885 2683 1036 510 249 ...
##  $ P.Undergrad: num  537 1227 99 63 869 ...
##  $ Outstate   : num  7440 12280 11250 12960 7560 ...
##  $ Room.Board : num  3300 6450 3750 5450 4120 ...
##  $ Books      : num  450 750 400 450 800 500 500 450 300 660 ...
##  $ Personal   : num  2200 1500 1165 875 1500 ...
##  $ PhD        : num  70 29 53 92 76 67 90 89 79 40 ...
##  $ Terminal   : num  78 30 66 97 72 73 93 100 84 41 ...
##  $ S.F.Ratio  : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
##  $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
##  $ Expend     : num  7041 10527 8735 19016 10922 ...
##  $ Grad.Rate  : num  60 56 54 59 15 55 63 73 80 52 ...
```

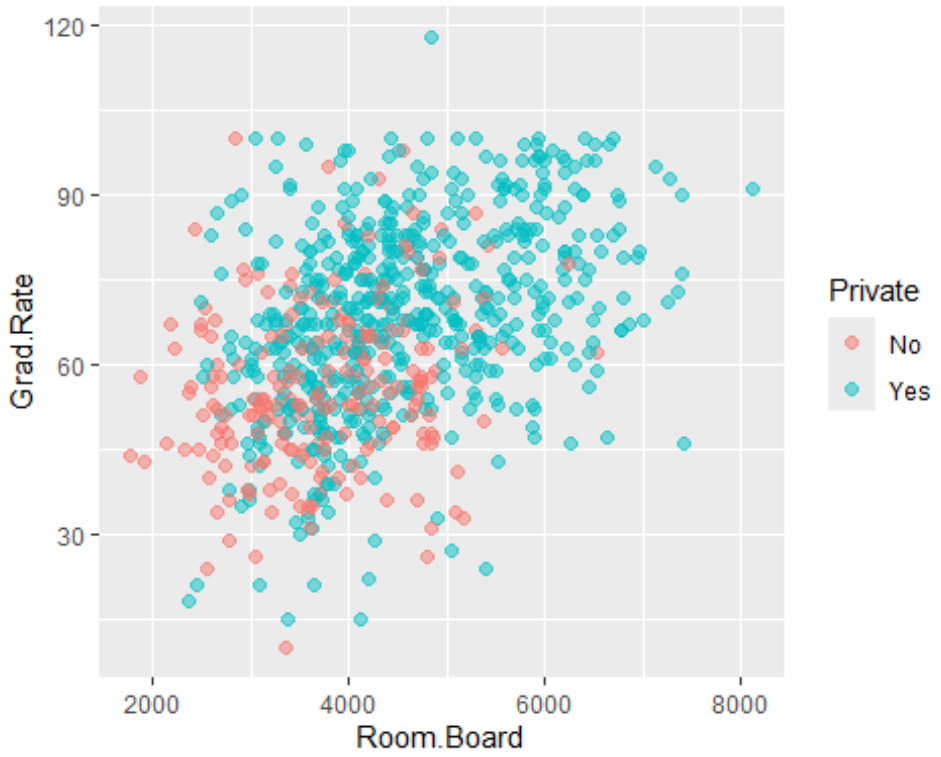
```
View(College)
```

```
#### changing the name for the simplicity
```

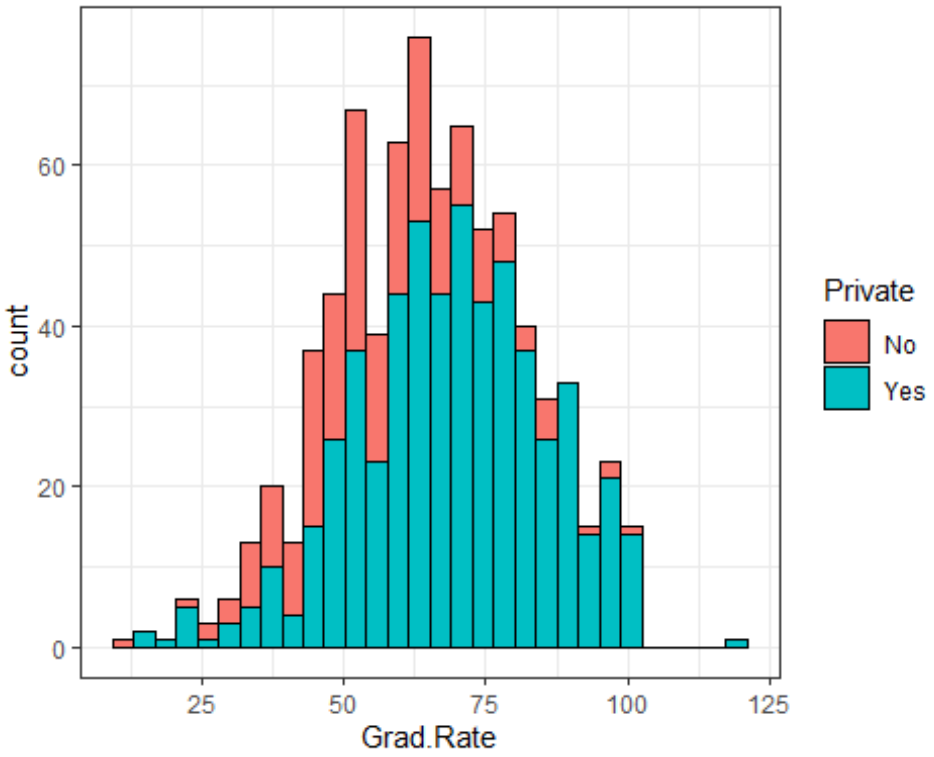
```
df <- College
```

```
#### EDA time
```

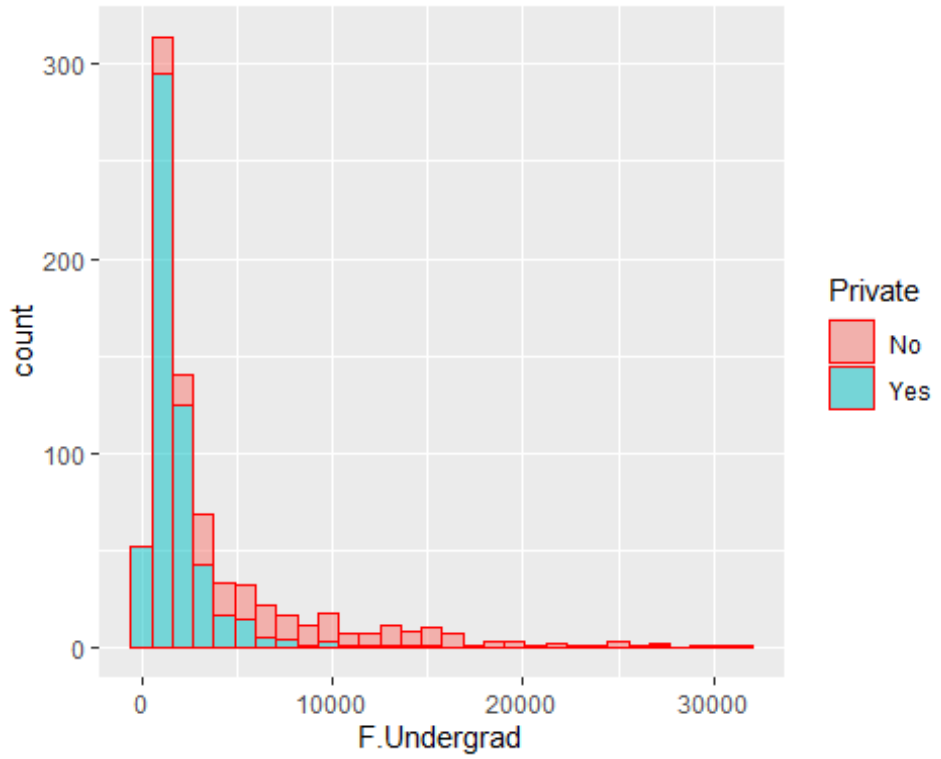
```
ggplot(df,aes(Room.Board, Grad.Rate)) +
geom_point(position=position_jitter(w=1,
h=0),aes(color=Private),alpha=0.5,size=2)
```




```
ggplot(df,aes(Grad.Rate)) + geom_histogram(aes(fill=Private),color='black') +
theme_bw()

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(df,aes(F.Undergrad)) +
geom_histogram((aes(fill=Private)),color='red',alpha=0.5)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

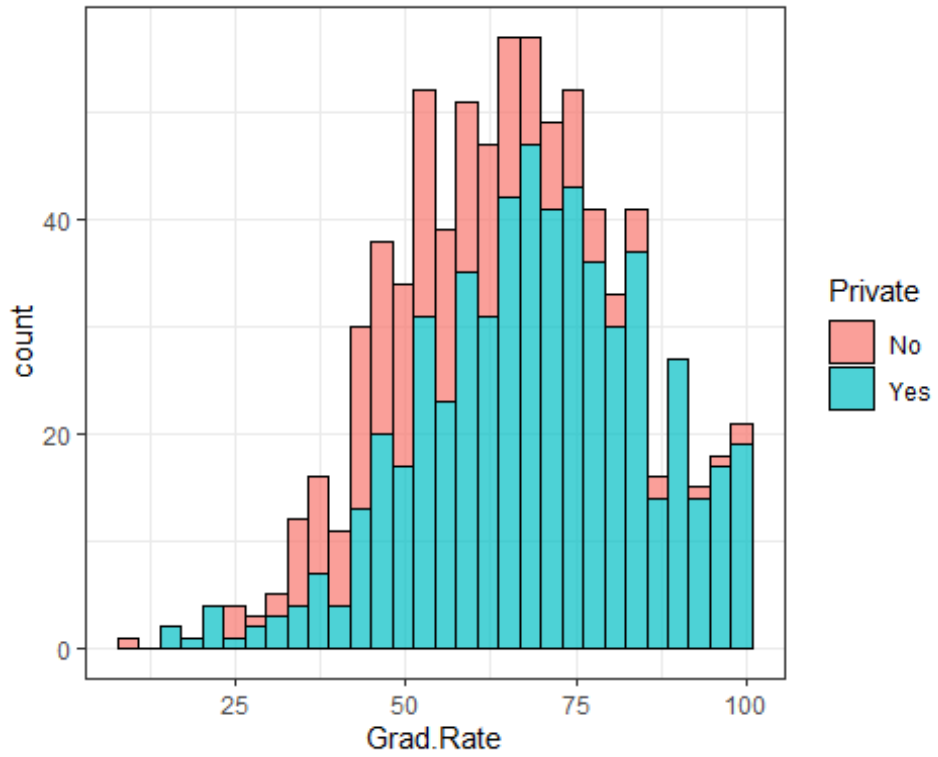



```
#### in the ggplot we see there something off because the graduation rate is
going above 100
#### lets find out which one is it

subset(df,Grad.Rate > 100)

##                    Private Apps Accept Enroll Top10perc Top25perc
F.Undergrad
## Cazenovia College     Yes 3847   3433    527         9        35
1010
##                    P.Undergrad Outstate Room.Board Books Personal PhD
Terminal
## Cazenovia College          12     9384       4840   600      500  22
47
##                    S.F.Ratio perc.alumni Expend Grad.Rate
## Cazenovia College      14.3          20   7697       118

#### Getting rid of 118 graduation rate and making it to 100

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

#### Run subset again just to make sure

subset(df,Grad.Rate > 100)

##  [1] Private     Apps        Accept      Enroll      Top10perc   Top25perc
##  [7] F.Undergrad P.Undergrad Outstate    Room.Board  Books       Personal
```

```
## [13] PhD         Terminal    S.F.Ratio   perc.alumni Expend      Grad.Rate
## <0 rows> (or 0-length row.names)
```

```
ggplot(df,aes(Grad.Rate)) +
geom_histogram(aes(fill=Private),color='black',alpha=0.7) + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#### I think our data is all clean to undergo model transformation
#### train and test data

sample <- sample.split(df$Private,SplitRatio = 0.7)

train <- subset(df,sample == TRUE)

test <- subset(df,sample == FALSE)

#### Making the model now for random forest

rf.model <- randomForest(Private ~ .,train)

print(rf.model$confusion)
```

```
##      No Yes class.error
## No  129  19  0.12837838
## Yes  14 382  0.03535354
```

```r
#### looking pretty good but do know this just the model and not the
predicted with test data so lets do that now

rf.predict <- predict(rf.model,test)

table(rf.predict,test$Private)

##
## rf.predict  No Yes
##        No   52   5
##        Yes  12 164

#### calculating accuracy

predict.acc <- (58+161) / (58+6+8+161)

print(predict.acc)

## [1] 0.9399142
```