

FinalProject2

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#Name: Aleksei Furlong  
#ID:2297345  
#Assignment: Final Project
```

##! Question 1

```
College2 <- na.omit(College)  
sum(is.na(College))
```

```
## [1] 4
```

##! Question 2

```
sum(College2$Apps)
```

```
## [1] 2327914
```

```
sum(College2$Enroll)
```

```
## [1] 604666
```

##! Question 3

```
College2$costWT = College2$Room.Board+ College2$Books+ College2$Personal  
mean(College2$costWT)
```

```
## [1] 6246.082
```

##! Question 4

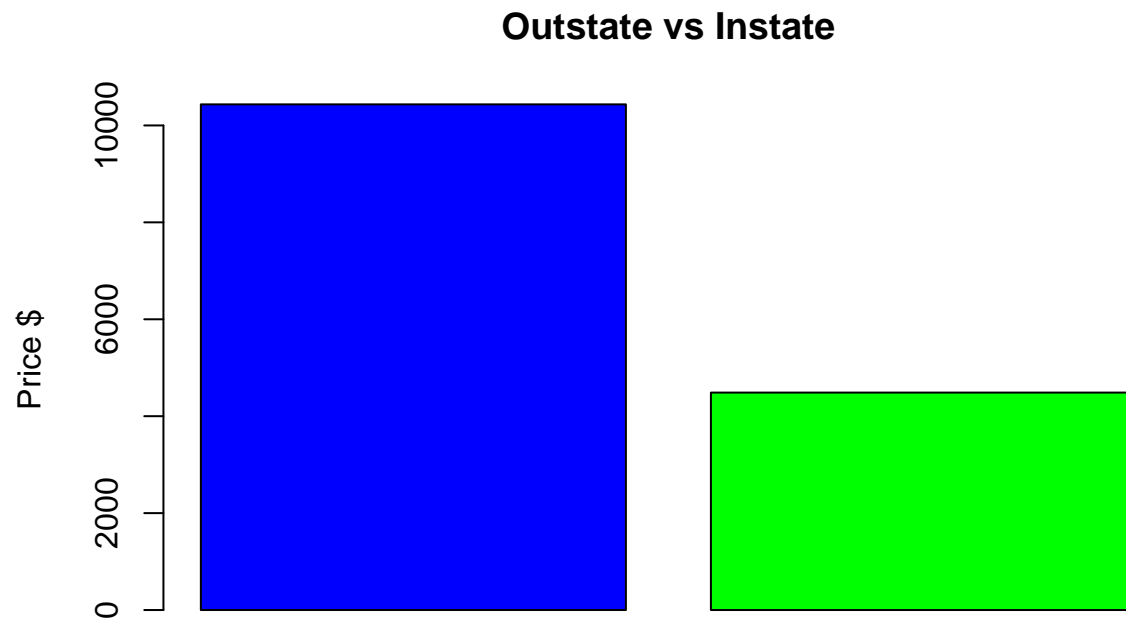
```
mean(College2$Outstate)
```

```
## [1] 10434.45
```

```
College2$Instate= College2$Outstate * 0.43
mean(College2$Instate)
```

```
## [1] 4486.814
```

```
barplot(c(mean(College2$Outstate),mean(College2$Instate)), col = c("blue", "green"), main = "Outstate v
```



Outstate(Blue) vs Instate(Green)

##! Question 5

```
CollegePrivate = College2 %>% select_all() %>% filter(Private == "Yes")
CollegePublic = College2 %>% select_all() %>% filter(Private == "No")
mean(CollegePrivate$Outstate)
```

```
## [1] 11794.1
```

```
mean(CollegePrivate$Instate)
```

```
## [1] 5071.465
```

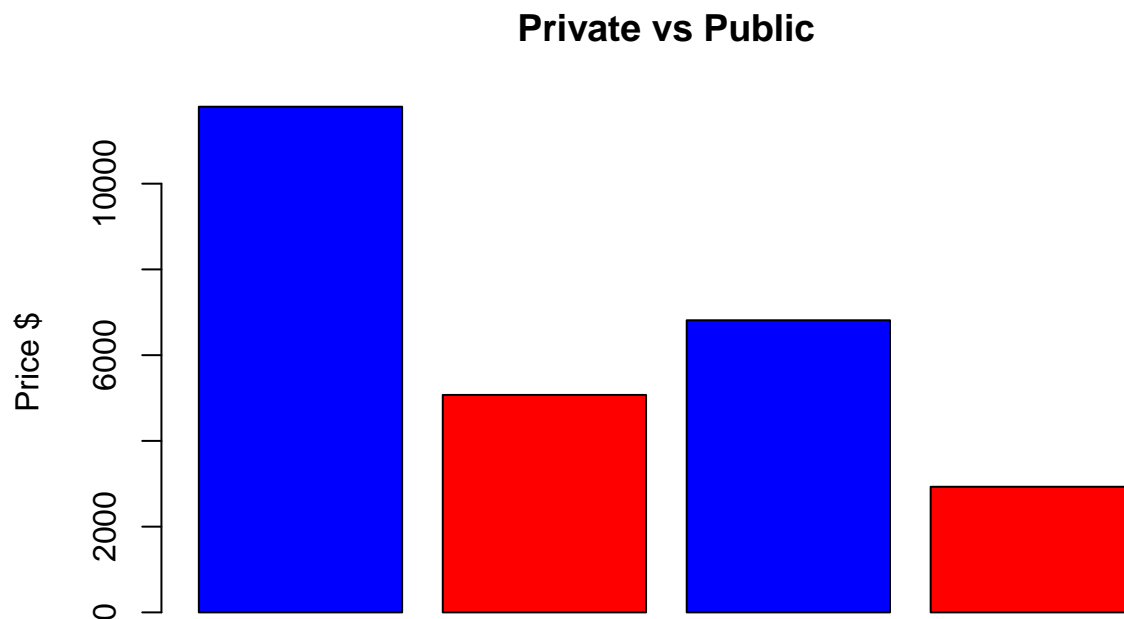
```
mean(CollegePublic$Outstate)
```

```
## [1] 6813
```

```
mean(CollegePublic$Instate)
```

```
## [1] 2929.59
```

```
barplot(c(mean(CollegePrivate$Outstate),mean(CollegePrivate$Instate),mean(CollegePublic$Outstate),mean(CollegePublic$Instate)))
```



Outstate(Blue) Instate(Red)
Private (left) vs Public (right)

##! Question 6

```
CollegeMatrix <- College2[,3:20]
CollegeMatrix <-cor(CollegeMatrix)
col <- colorRampPalette(c("white","black"))(20)
corrplot(CollegeMatrix,method = "color",order = "hclust", col = col, bg = "white", outline = "white", t = FALSE)
```

```
## Warning in text.default(pos.xlabel[, 1], pos.xlabel[, 2], newcolnames, srt = 
## t1.srt, : "t1.col" is not a graphical parameter
```

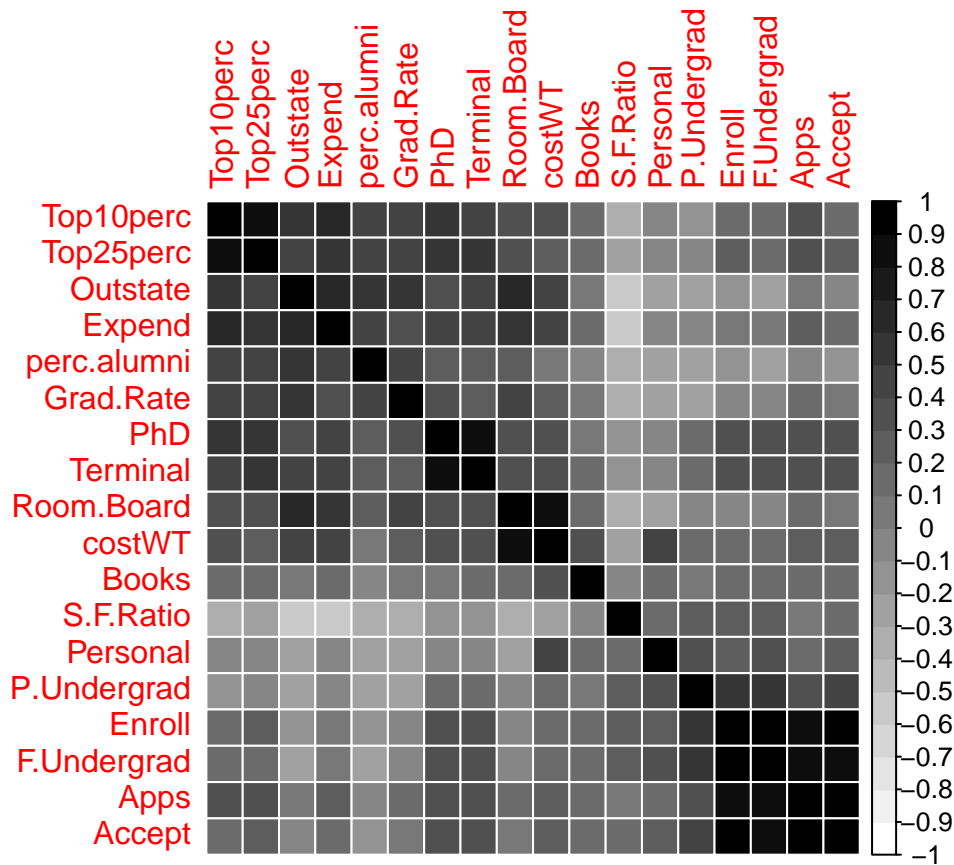
```
## Warning in text.default(pos.xlabel[, 1], pos.xlabel[, 2], newcolnames, srt = 
## t1.srt, : "t1.srt" is not a graphical parameter
```

```
## Warning in text.default(pos.ylabel[, 1], pos.ylabel[, 2], newrownames, col = 
## t1.col, : "t1.col" is not a graphical parameter
```

```
## Warning in text.default(pos.ylabel[, 1], pos.ylabel[, 2], newrownames, col = 
## t1.col, : "t1.srt" is not a graphical parameter
```

```
## Warning in title(title, ...): "t1.col" is not a graphical parameter
```

```
## Warning in title(title, ...): "t1.srt" is not a graphical parameter
```



```
##! Question 7
```

```
College2$TotalEnrollment = College2$F.Undergrad + College2$P.Undergrad
College2$EnrollmentPercentage = College2$F.Undergrad/College2$TotalEnrollment
sum(College2$EnrollmentPercentage >= 0.9)
```

```
## [1] 294
```

```
##! Question 8
```

```
College2$AcceptanceRate = College2$Accept/College2$Apps
CollegeGreater = College2 %>% select_all() %>% filter(College2$EnrollmentPercentage >= 0.9)
CollegeLess = College2 %>% select_all() %>% filter(College2$EnrollmentPercentage <= 0.9)
mean(CollegeGreater$AcceptanceRate)
```

```
## [1] 0.7145089
```

```
mean(CollegeLess$AcceptanceRate)
```

```
## [1] 0.7663636
```

```
# acceptance rate of colleges with less than 90% full time enrollment is LOWER
```

```
##! Question 9
```

```
College2$College = grepl("College",College2$Name)
College2Collegename = College2 %>% select_all() %>% filter(College2$College == "TRUE")
College2Universityname = College2 %>% select_all() %>% filter(College2$College == "FALSE")
mean(College2Collegename$AcceptanceRate)-mean(College2Universityname$AcceptanceRate)
```

```
## [1] 0.02193126
```

```
# difference is 2.19%
```

```
##! Question 10
```

```
ggplot(data = College2,aes(x=College2$perc.alumni,y=College2$Grad.Rate))+geom_smooth()+geom_point()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

