Stat 432 HW 01

Name: Ahmadreza Eslaminia, netID: ae15

Summer 2024

Include the R code for this HW.

```
knitr::opts_chunk$set(echo = TRUE)
library(ISLR2)
library(GGally)
```

There are some useful R chunk options that you may use (for this entire semester):

- echo - Display code in output document (default = TRUE)
- include - Include chunk in document after running (default = TRUE)
- message - display code messages in document (default = TRUE)
- results (default = 'markup')
  - 'asis' - passthrough results
  - 'hide' - do not display results
  - 'hold' - put all results below all code
- error - Display error messages in doc (TRUE) or stop render when errors occur (FALSE) (default = FALSE)

See R markdown cheat sheet for more information.

Question 1

This question relates to the College data set, which can be found in the file `College.csv`. It contains a number of variables for 777 different universities and colleges in the US.

1-1

a. Use the `read.csv()` function to read the data into R. Call the loaded data `college`. Make sure that you have the directory set to the correct location for the data.

```
college <- read.csv("College.csv")
```

b. What are the variable names?

```
names(college)
```

```
##  [1] "X"           "Private"     "Apps"        "Accept"      "Enroll"
##  [6] "Top10perc"   "Top25perc"   "F.Undergrad" "P.Undergrad" "Outstate"
## [11] "Room.Board"  "Books"       "Personal"    "PhD"         "Terminal"
## [16] "S.F.Ratio"   "perc.alumni" "Expend"      "Grad.Rate"
```

c. Use the `summary()` function to produce a numerical summary of the variables in the data set.

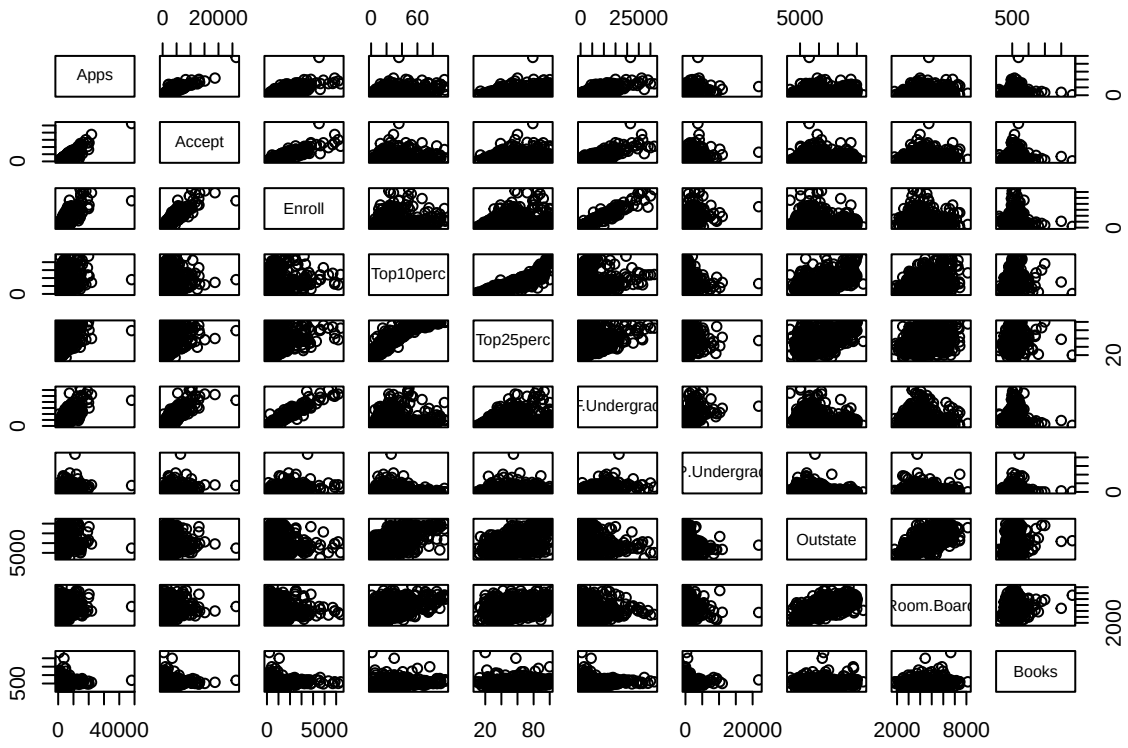
```
summary(college)
```

```
##       X                Private               Apps           Accept
##  Length:777         Length:777         Min.   :   81   Min.   :   72
##  Class :character   Class :character   1st Qu.:  776   1st Qu.:  604
##  Mode  :character   Mode  :character   Median : 1558   Median : 1110
##                                        Mean   : 3002   Mean   : 2019
##                                        3rd Qu.: 3624   3rd Qu.: 2424
##                                        Max.   :48094   Max.   :26330
##      Enroll       Top10perc       Top25perc      F.Undergrad
##  Min.   :  35   Min.   : 1.00   Min.   :  9.0   Min.   :  139
##  1st Qu.: 242   1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992
##  Median : 434   Median :23.00   Median : 54.0   Median : 1707
##  Mean   : 780   Mean   :27.56   Mean   : 55.8   Mean   : 3700
##  3rd Qu.: 902   3rd Qu.:35.00   3rd Qu.: 69.0   3rd Qu.: 4005
##  Max.   :6392   Max.   :96.00   Max.   :100.0   Max.   :31643
##   P.Undergrad         Outstate       Room.Board       Books
##  Min.   :    1.0   Min.   : 2340   Min.   :1780   Min.   :  96.0
##  1st Qu.:   95.0   1st Qu.: 7320   1st Qu.:3597   1st Qu.: 470.0
##  Median :  353.0   Median : 9990   Median :4200   Median : 500.0
##  Mean   :  855.3   Mean   :10441   Mean   :4358   Mean   : 549.4
##  3rd Qu.:  967.0   3rd Qu.:12925   3rd Qu.:5050   3rd Qu.: 600.0
##  Max.   :21836.0   Max.   :21700   Max.   :8124   Max.   :2340.0
##     Personal          PhD            Terminal       S.F.Ratio
##  Min.   : 250   Min.   :  8.00   Min.   : 24.0   Min.   : 2.50
##  1st Qu.: 850   1st Qu.: 62.00   1st Qu.: 71.0   1st Qu.:11.50
##  Median :1200   Median : 75.00   Median : 82.0   Median :13.60
##  Mean   :1341   Mean   : 72.66   Mean   : 79.7   Mean   :14.09
```

```
##  3rd Qu.:1700   3rd Qu.: 85.00   3rd Qu.: 92.0   3rd Qu.:16.50
##  Max.   :6800   Max.   :103.00   Max.   :100.0   Max.   :39.80
##   perc.alumni        Expend        Grad.Rate
##  Min.   : 0.00   Min.   : 3186   Min.   : 10.00
##  1st Qu.:13.00   1st Qu.: 6751   1st Qu.: 53.00
##  Median :21.00   Median : 8377   Median : 65.00
##  Mean   :22.74   Mean   : 9660   Mean   : 65.46
##  3rd Qu.:31.00   3rd Qu.:10830   3rd Qu.: 78.00
##  Max.   :64.00   Max.   :56233   Max.   :118.00
```

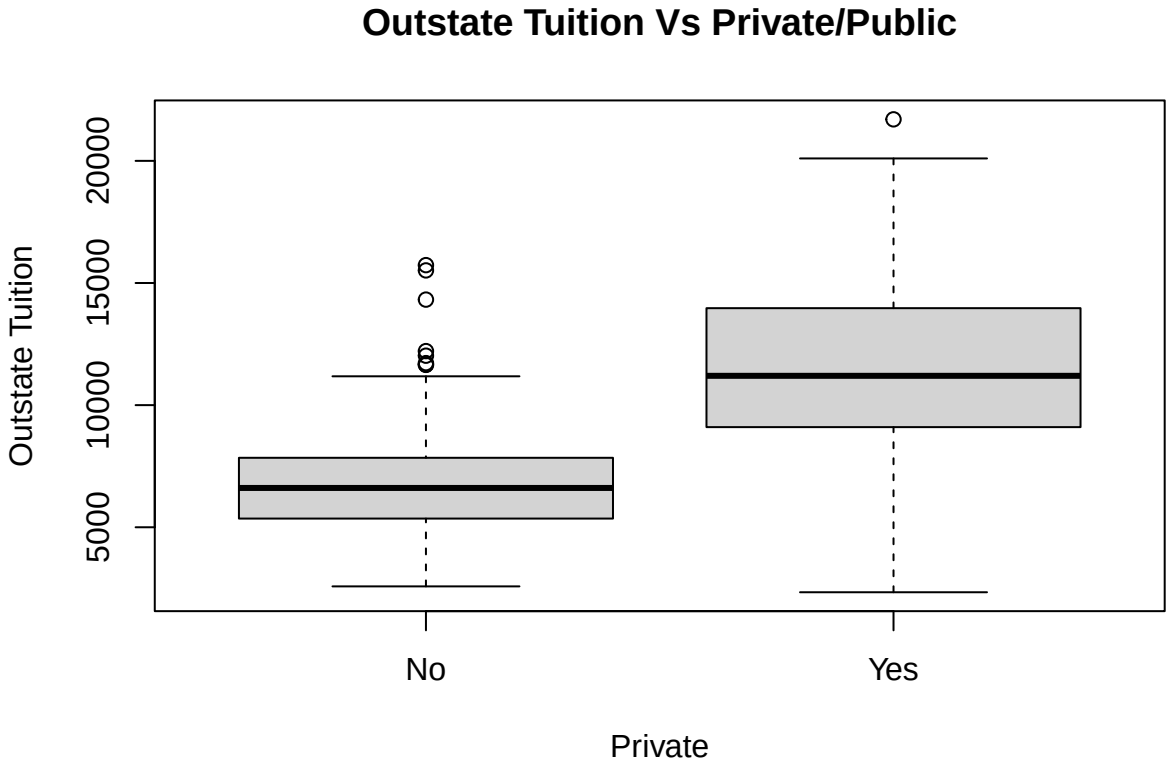
d. Use the `pairs()` function to produce a scatter plot matrix of the first ten columns or variables of the data. (First two columns are not numeric, so start from the third column.)

```
pairs(college[, 3:12])
```



e. Use the `plot()` function to produce side-by-side boxplots of `Outstate` versus `Private`.

```
boxplot(Outstate ~ Private, data = college,
        main = "Outstate Tuition Vs Private/Public",
        xlab = "Private",
        ylab = "Outstate Tuition")
```



###

1-2

a. Is there any missing data? If so, use `na.omit()` to remove rows containing missing observations.

```
# Check for missin valuees

any_missing <- anyNA(college)
print(paste("have missing value : ?", any_missing))
```

```
## [1] "have missing value : ? FALSE"
```

```
# deletes rows with missing data
college_clean <- if (anyNA(college)) na.omit(college) else college
```

b. Split your data into two parts: a testing data that contains 100 observations, and the rest as training data. You may use `sample` function to et the indices of the testing data. For this question, you need to set a random seed while generating this split so that the result can be replicated. Use `4322` as the random seed. Report the mean of `Outstate` of your testing data and training data, respectively.

```
set.seed(4322)

test_indices <- sample(1:nrow(college_clean), 100)

# making test and training
college_test <- college_clean[test_indices, ]
college_train <- college_clean[-test_indices, ]

# make mean of 'Outstate'
```

```r
mean_outstate_test <- mean(college_test$Outstate)
mean_outstate_train <- mean(college_train$Outstate)

print(paste("Mean for the tes data:", mean_outstate_test))
```

```
## [1] "Mean for the tes data: 9955.46"
```

```r
print(paste("Mean for the training data:", mean_outstate_train))
```

```
## [1] "Mean for the training data: 10512.3397341211"
```

c. Use the training data to perform a EDA (Exploratory Data Analysis). Our goal is to predict the `enroll` of the data set.

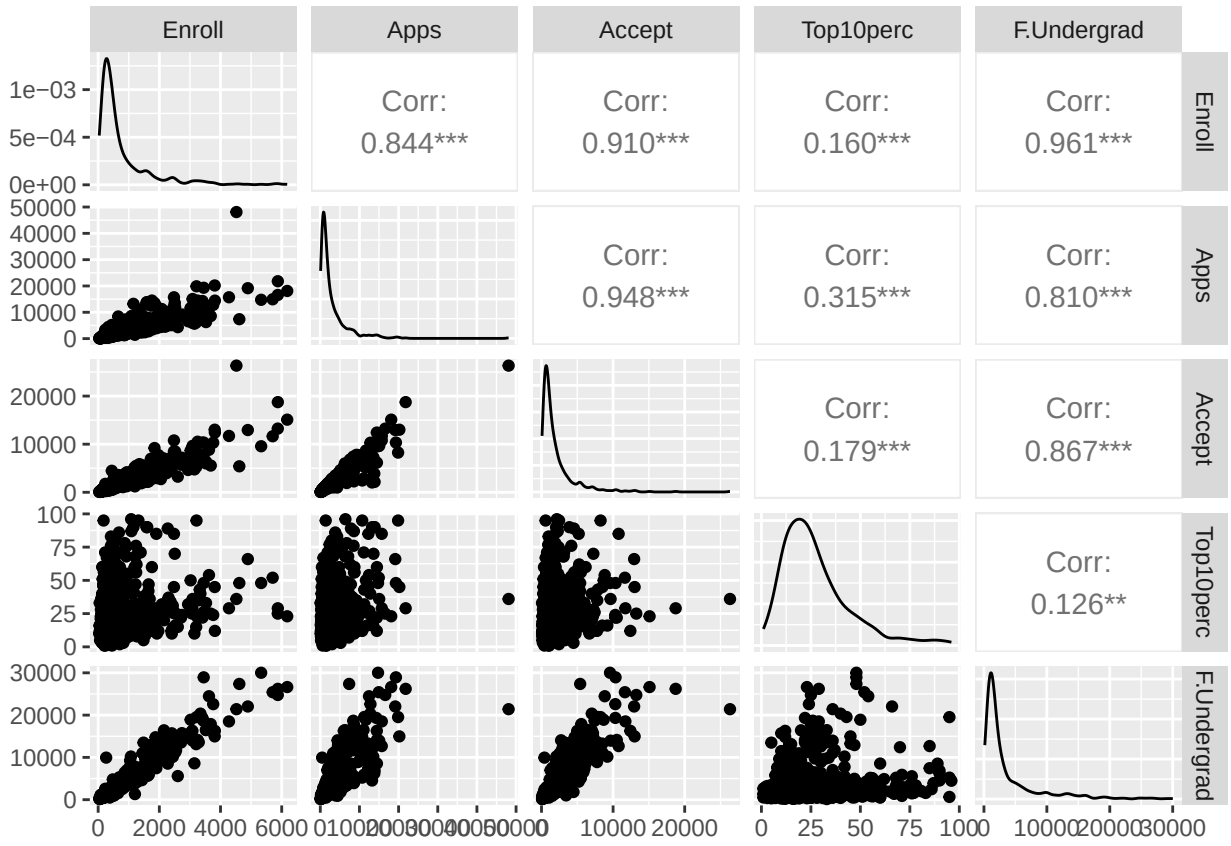
- Use the `head` function to have a look at how our data set looks like
- Use the `GGally::ggpairs` function to make visual plots between `enroll` and other variables. Include 4~5 variables in one plot. Since the goal is to predict `enroll`, include `enroll` variable in all the plots.

Note: use the *training data*!

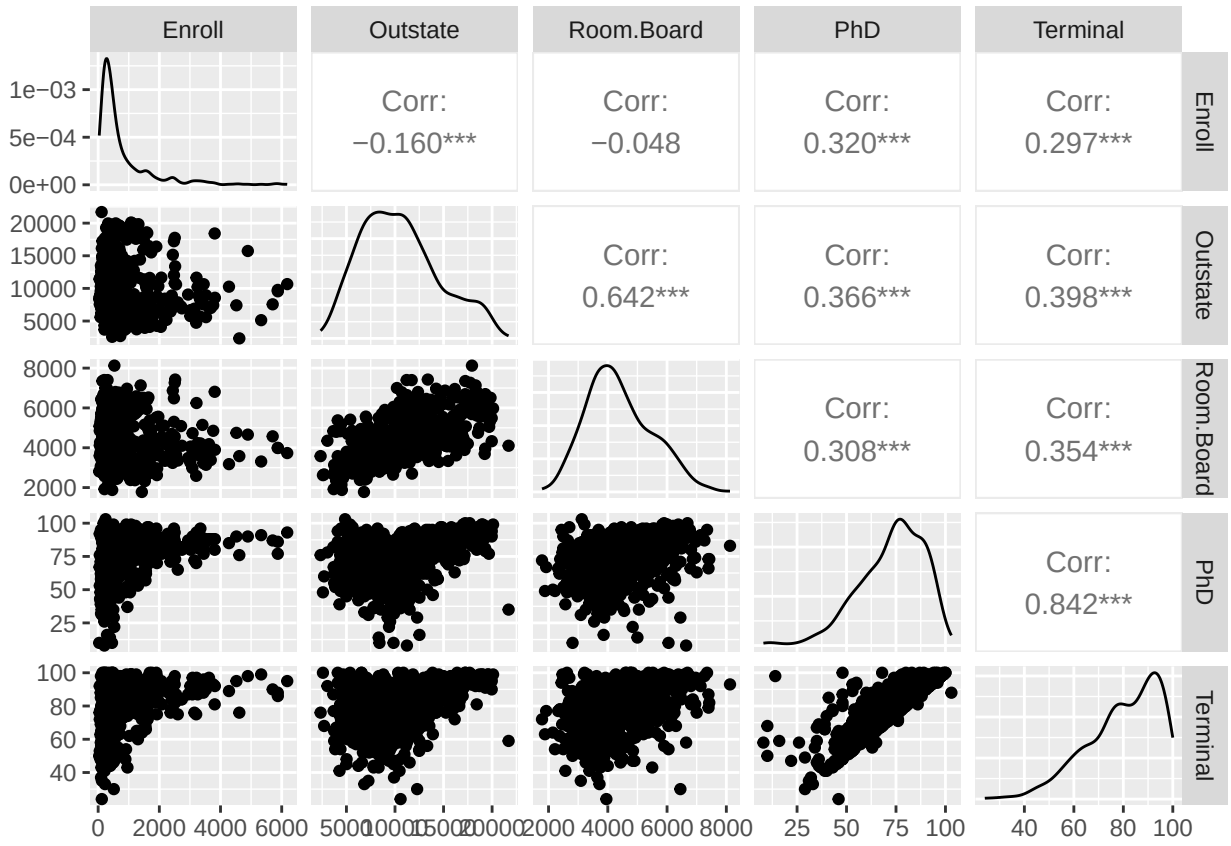
```r
head(college_train)
```

```
##                              X Private Apps Accept Enroll Top10perc Top25perc
## 1 Abilene Christian University     Yes 1660   1232    721        23        52
## 2           Adelphi University     Yes 2186   1924    512        16        29
## 3               Adrian College     Yes 1428   1097    336        22        50
## 4          Agnes Scott College     Yes  417    349    137        60        89
## 5    Alaska Pacific University     Yes  193    146     55        16        44
## 6            Albertson College     Yes  587    479    158        38        62
##   F.Undergrad P.Undergrad Outstate Room.Board Books Personal PhD Terminal
## 1        2885         537     7440       3300   450     2200  70       78
## 2        2683        1227    12280       6450   750     1500  29       30
## 3        1036          99    11250       3750   400     1165  53       66
## 4         510          63    12960       5450   450      875  92       97
## 5         249         869     7560       4120   800     1500  76       72
## 6         678          41    13500       3335   500      675  67       73
##   S.F.Ratio perc.alumni Expend Grad.Rate
## 1      18.1          12   7041        60
## 2      12.2          16  10527        56
## 3      12.9          30   8735        54
## 4       7.7          37  19016        59
## 5      11.9           2  10922        15
## 6       9.4          11   9727        55
```

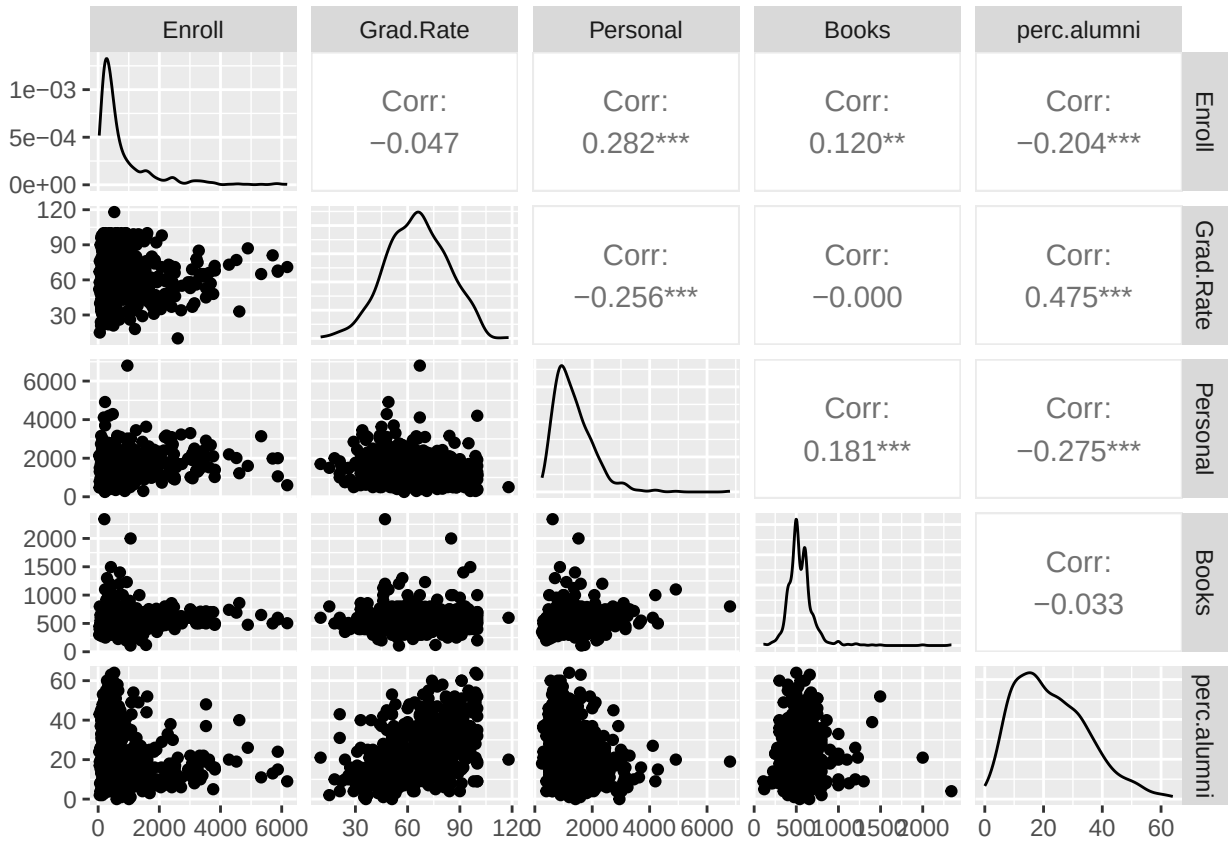
```r
ggpairs(college_train, columns = c("Enroll", "Apps", "Accept", "Top10perc", "F.Undergrad"))
```



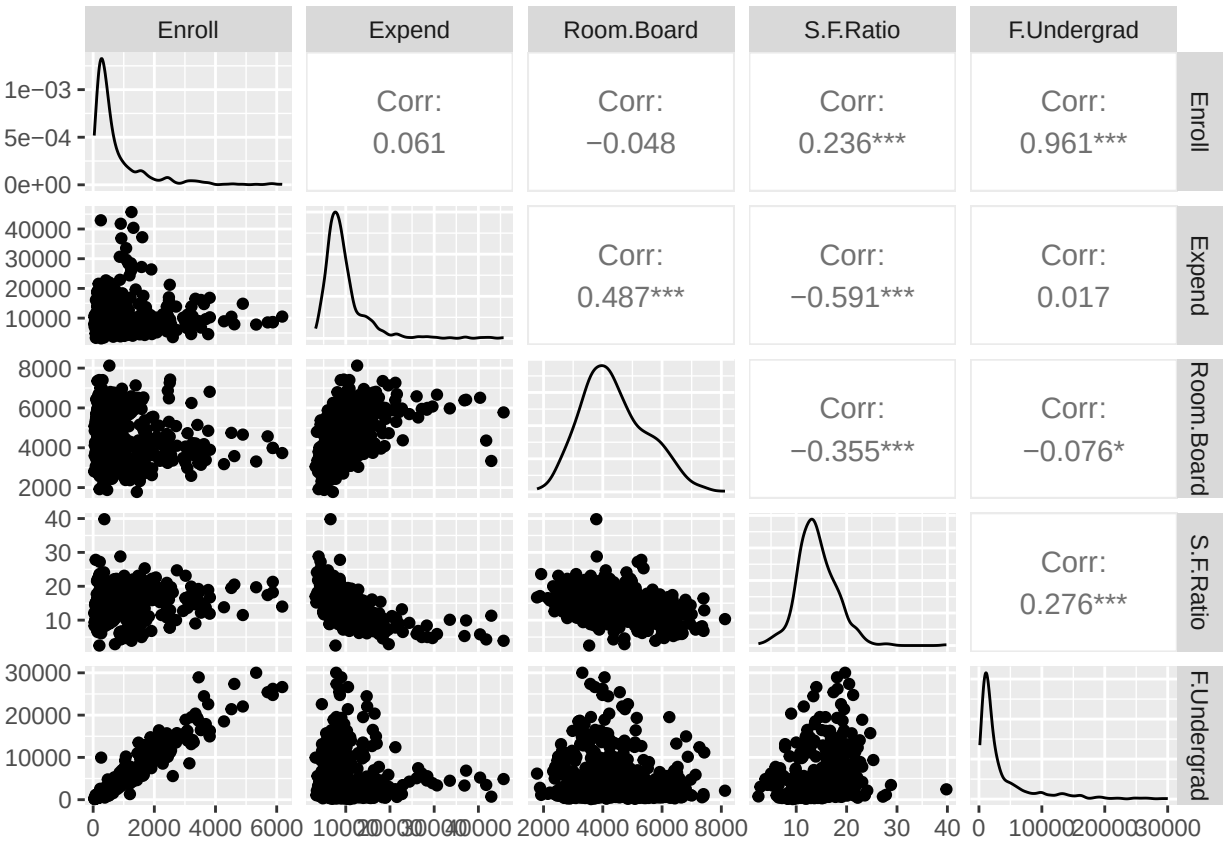

```
ggpairs(college_train, columns = c("Enroll", "Outstate", "Room.Board", "PhD", "Terminal"))
```




```
ggpairs(college_train, columns = c("Enroll", "Grad.Rate", "Personal", "Books", "perc.alumni"))
```




```
ggpairs(college_train, columns = c("Enroll", "Expend", "Room.Board", "S.F.Ratio", "F.Undergrad"))
```



d. Based on your EDA analysis, pick three variables might be most relevant to `enroll` (the variables may vary from student to student). Explain your reason.

Facros; F-Undergrad - Accept - Apps When picking the best variables to predict Enroll, we looking at scatter plots to see which ones show high corroletion patterns with Enroll. For example, the number of students accepted (Accept) has a strong positive correlation number with Enroll, because more acceptances usually means more students enroll. Then there's the number of undergraduates (F.Undergrad), which is related into the overall size of the student. Also the number of applications (Apps), which is another sign of how many students might enroll; more apps generally lead to more enrollments. we can see in the plots that the number of corolation for these three factor is higher than other factors.

Question 2

Load in the `Boston` data set. The Boston data set is part of the `ISLR2` library.

a. How many rows are in this data set? How many columns? What do the rows and columns represent?

```
data("Boston")

num_rows <- nrow(Boston)
num_columns <- ncol(Boston)
print(paste(" rows:", num_rows))
```

```
## [1] " rows: 506"
```

```
print(paste("columns:", num_columns))
```

```
## [1] "columns: 13"
```

```
head(Boston)
```

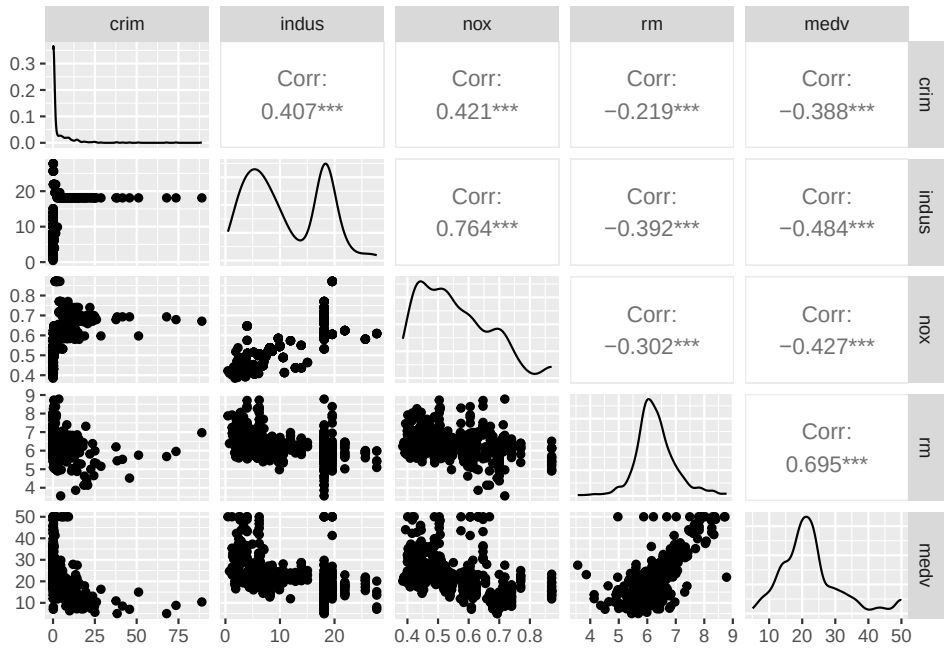
```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3  4.98 24.0
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8  9.14 21.6
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7  2.94 33.4
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7  5.33 36.2
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7  5.21 28.7
```

Each row is a town in the Boston area, and each column represents a feature related to housing in these towns.

b. Make some pairwise scatterplots of the predictors (columns) in this data set. Adjust the `R` chunk option of the plot such that the plot is at the center and occupies 75% of the page width. Describe your findings.

Hint: https://bookdown.org/yihui/rmarkdown-cookbook/figure-size.html

```
ggpairs(Boston[, c("crim", "indus", "nox", "rm", "medv")])
```



It seems between these random features correlation between the nox and indus is highest with number of 0.764 ater that the rm Vs medv is highest one with 0.695. For other pairwise combination it seems they are not so much correlated.

c. Are any of the predictors associated with per capita crime rate? If so, explain the relationship.

```
correlations <- cor(Boston)
cor_crim <- correlations["crim", ]
print(cor_crim)
```

```
##        crim          zn       indus        chas         nox          rm
##  1.00000000 -0.20046922  0.40658341 -0.05589158  0.42097171 -0.21924670
##         age         dis         rad         tax     ptratio       lstat
##  0.35273425 -0.37967009  0.62550515  0.58276431  0.28994558  0.45562148
##        medv
## -0.38830461
```

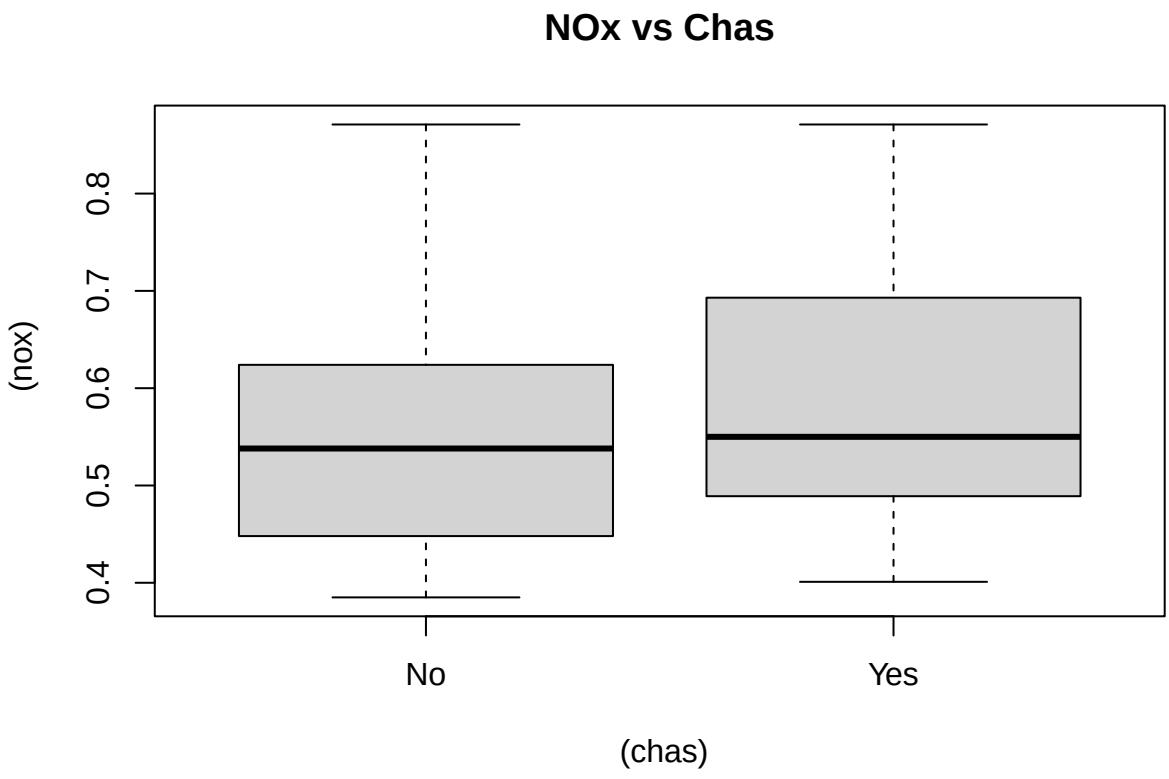
It seems places with areas with good access to highways (rad) 0.62 , high property taxes (tax)0.58, and lots of tend to have higher crime.in next step more industry land (indus)0.40 , lstat 0.45 and more air pollution (nox)0.42,Old neighborhoods (age)0.32 , and places with higher student-teacher ratios (ptratio)0.28 have caused more crime. But, neighborhoods with big residential zones (zn)-0.2, homes with more rooms (rm)-0.21, and those further away from job centers (dis)-0.37 and the high home values (medv)-0.38 usually have lower crime rates. Being close to the Charles River (chas) doesn't really change the crime much.

d. The `chas` variable is stored as a numeric vector, so `R` has treated it as quantitative. Convert `chas` variables into qualitative variables.Make a side-by-side boxplot of `nox` versus `chas`.

```
Boston$chas <- as.factor(Boston$chas)


boxplot(nox ~ chas, data = Boston,
```

```
        main = "NOx vs Chas",
        xlab = " (chas)",
        ylab = " (nox)",
        names = c("No", "Yes"))  # Rename levels for clarity
```



##

Question 3

$X_1, X_2, \ldots, X_n$ are i.i.d. $Uniform(0, 1)$ random variables.

a. Generate a set of $n = 100$ observations from this distribution. Only display the first 10 observations in your `R` output. Use your UIN as the seed.

```
set.seed(655533073)

observations <- runif(100, min = 0, max = 1)

print(observations[1:10])
```

```
##  [1] 0.9464926 0.8406841 0.2749549 0.2735737 0.5105138 0.9569128 0.9393758
##  [8] 0.1570902 0.1931335 0.1340425
```

b. What is the sample mean and sample variance? Use your own code to calculate these quantities. That means, you should not use `mean()`, `sd()` or `var()`.

```
sample_mean <- sum(observations) / 100

sample_variance <- sum((observations - sample_mean)^2) / (100 -1)

print(paste("Sample Mean:", sample_mean))
```

```
## [1] "Sample Mean: 0.470889973382"
```

```
print(paste("Sample Variance:", sample_variance))
```

```
## [1] "Sample Variance: 0.0808987419204797"
```

c. Use default R functions to check if your answers in (b) are correct.

```
print(paste("Sample Mean with built0in func:", mean(observations)))
```

```
## [1] "Sample Mean with built0in func: 0.470889973382"
```

```
print(paste("Sample Variancewith built0in func:", var(observations)))
```

```
## [1] "Sample Variancewith built0in func: 0.0808987419204797"
```

They are the same so both are correct.

Question 4 (GR Only)

Note that $f(x) = E[Y|X = x]$ minimizes $E[(Y - f(X))^2|X = x]$. Then what g minimizes $E[|Y - g(X)||X = x]$? Justify your answer.

We want find the fuction g that minimize the expected absolute deviation:

$$E[|Y - g(X)||X = x]$$

For any x, the function $g(x)$ that minimiz this expected absolute deviation is the conditional median of Y given $X = x$. This is because the median minimizes the sum of absolute differencs from the median values.

Mathmatically, the conditional median $m(x)$ is defined like this:

$$\text{median}(Y|X = x) = m(x) \quad \text{so that} \quad P(Y \leq m(x)|X = x) = 0.5$$

This means $m(x)$ is the value where half of the probability of Y goes below and half goes above, given $X = x$. By balancing values on both side, the median reduces the total sum of absolute deviations from it.

So, the function $g(x)$ that minimize $E[|Y - g(X)||X = x]$ is:

$$g(x) = \text{median}(Y|X = x)$$