Data Exploration of Universities

Xiaojia Zhang

12/01/2018

1. Data Exploration

a. Import the "Universities-Exam1.csv" file to R Studio and write R code to produce summary statistics of all numeric variables. For each numeric variable, include the following: Variable name, Minimum, Maximum, Median, Mean, Standard Deviation, Number of Missing Values

```
uni <- read.csv("Universities-Exam1.csv")
#take first column as rowname.
rownames(uni) <- make.names(uni[,1], unique = T)
uni <- uni[,-1]
#as we can see, the type of "Admits" is factor.
str(uni)
```

```
## 'data.frame':    1302 obs. of  7 variables:
##  $ State              : Factor w/ 51 levels "AK","AL","AR",..: 44 6 35 23 11 2 2 1 11 14 ...
##  $ Type               : Factor w/ 2 levels "Private","Public": 1 2 1 1 1 2 2 1 2 1 ...
##  $ Applicants         : int  1660 1508 2186 1428 417 2817 4639 193 1461 587 ...
##  $ Admits             : Factor w/ 1067 levels "","         '",..: 88 104 306 47 581 305 553 177 47 7
##  $ Enrolled           : int  721 569 512 336 137 984 1278 55 580 158 ...
##  $ StudentFacultyRatio: num  18.1 27.9 12.2 12.9 7.7 14.3 18.7 11.9 17 9.4 ...
##  $ GraduationRate     : int  60 60 56 54 59 40 15 15 62 55 ...
```

```
#convert factor to character first, and then convert it to numeric.
uni$Admits <- as.numeric(as.character(uni$Admits))
```

```
## Warning: NAs introduced by coercion
```

```
#now the type of "Admits" is numeric.
str(uni)
```

```
## 'data.frame':    1302 obs. of  7 variables:
##  $ State              : Factor w/ 51 levels "AK","AL","AR",..: 44 6 35 23 11 2 2 1 11 14 ...
##  $ Type               : Factor w/ 2 levels "Private","Public": 1 2 1 1 1 2 2 1 2 1 ...
##  $ Applicants         : int  1660 1508 2186 1428 417 2817 4639 193 1461 587 ...
##  $ Admits             : num  1232 1259 1924 1097 349 ...
##  $ Enrolled           : int  721 569 512 336 137 984 1278 55 580 158 ...
##  $ StudentFacultyRatio: num  18.1 27.9 12.2 12.9 7.7 14.3 18.7 11.9 17 9.4 ...
##  $ GraduationRate     : int  60 60 56 54 59 40 15 15 62 55 ...
```

```
#show all the variable names.
colnames(uni)
```

```
## [1] "State"              "Type"               "Applicants"
## [4] "Admits"             "Enrolled"           "StudentFacultyRatio"
## [7] "GraduationRate"
```

```r
#summary statistics of all numeric variables.
uni_num <- uni[,3:7]
uni_stats <- data.frame(
  min = sapply(uni_num, min, na.rm = T),
  max = sapply(uni_num, max, na.rm = T),
  med = sapply(uni_num, median, na.rm = T),
  mean = sapply(uni_num, mean, na.rm = T),
  sd = sapply(uni_num, sd, na.rm = T),
  miss.val=sapply(uni_num, function(x)
    sum(length(which(is.na(x)))))
)
uni_stats
```

```
##                         min     max    med       mean          sd miss.val
## Applicants             35.0 48094.0 1470.0 2752.09752 3541.974712       10
## Admits                 35.0 26330.0 1095.0 1870.68319 2250.866400       11
## Enrolled             -288.0  7425.0  447.0  778.43639  884.969414        5
## StudentFacultyRatio    2.3    91.8   14.3   14.85877    5.186399        2
## GraduationRate         8.0  1000.0   60.0   61.13787   32.975231       98
```

```r
#we can see the min of Enroll is -288 which doesn't make sense, the values of Enroll
#should always be >= 0, so here I change all the negative values to NA.
uni_num$Enrolled[uni_num$Enrolled < 0] = NA
#and also the the max of GraduationRate is 1000 which doesn't make sense, the values of
#GraduationRate should be [0,100], so here I change values, which are higher than 100, to NA.
uni_num$GraduationRate[uni_num$GraduationRate > 100] = NA
#check statistics values.
uni_stats <- data.frame(
  min = sapply(uni_num, min, na.rm = T),
  max = sapply(uni_num, max, na.rm = T),
  med = sapply(uni_num, median, na.rm = T),
  mean = sapply(uni_num, mean, na.rm = T),
  sd = sapply(uni_num, sd, na.rm = T),
  miss.val=sapply(uni_num, function(x)
    sum(length(which(is.na(x)))))
)
uni_stats
```

```
##                        min     max    med       mean          sd miss.val
## Applicants            35.0 48094.0 1470.0 2752.09752 3541.974712       10
## Admits                35.0 26330.0 1095.0 1870.68319 2250.866400       11
## Enrolled              18.0  7425.0  448.0  779.25926  884.814521        6
## StudentFacultyRatio    2.3    91.8   14.3   14.85877    5.186399        2
## GraduationRate         8.0   100.0   60.0   60.35744   18.823692       99
```

```r
#calculate the number of missing value
miss <- length(which(is.na(uni_num)))
miss
```

```
## [1] 128
```

```r
#now we can see the min of Enroll is 18, the max of GraduationRate is 100, the data
#is clean now. Also the number of NAs increases.
```

b. Create a new column called "AdmitRate" and add it to the data set as the last column. Admit rate should be calculated by dividing "Admits" by "Applicants" to get the percent of

students admitted. Paste a screen shot of the first 10 rows of the dataset with the new column below.

```
Admits.df <- data.frame(uni_num$Admits)
colnames(Admits.df) <- c("Admits")
Applicants.df <- data.frame(uni_num$Applicants)
colnames(Applicants.df) <- c("Applicants")

AdmitRate <- round((Admits.df/Applicants.df)*100, 2)
colnames(AdmitRate) <- c('AdmitRate')
#add "AdmitRate" column to original data frame
uni_rate <- cbind(uni_num, AdmitRate)
head(uni_rate,10)
```

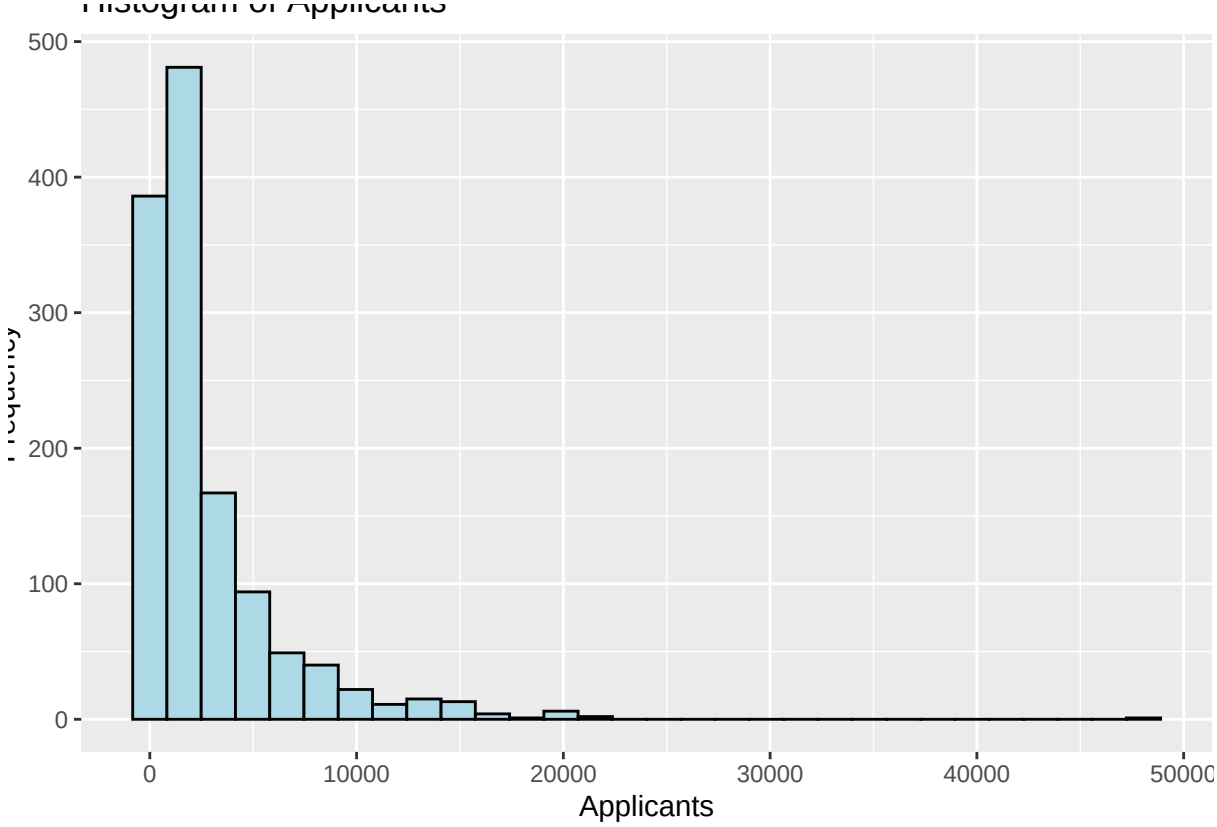
```
##                              Applicants Admits Enrolled StudentFacultyRatio
## Abilene.Christian.University       1660   1232      721                18.1
## Adams.State.College                1508   1259      569                27.9
## Adelphi.University                 2186   1924      512                12.2
## Adrian.College                     1428   1097      336                12.9
## Agnes.Scott.College                 417    349      137                 7.7
## Alabama.Agri....Mech..Univ.        2817   1920      984                14.3
## Alabama.State.University           4639   3272     1278                18.7
## Alaska.Pacific.University           193    146       55                11.9
## Albany.State.College               1461   1097      580                17.0
## Albertson.College                   587    479      158                 9.4
##                              GraduationRate AdmitRate
## Abilene.Christian.University             60     74.22
## Adams.State.College                      60     83.49
## Adelphi.University                       56     88.01
## Adrian.College                           54     76.82
## Agnes.Scott.College                      59     83.69
## Alabama.Agri....Mech..Univ.              40     68.16
## Alabama.State.University                 15     70.53
## Alaska.Pacific.University                15     75.65
## Albany.State.College                     62     75.09
## Albertson.College                        55     81.60
```

c. Create a histogram of the Applicants variable.

```
library(ggplot2)
ggplot(data = uni_rate, aes(Applicants)) +
  geom_histogram(col = 'black', fill = 'light blue') +
  labs (title = 'Histogram of Applicants',
        x = 'Applicants', y = 'Frequency')
```

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

```
## Warning: Removed 10 rows containing non-finite values (stat_bin).
```

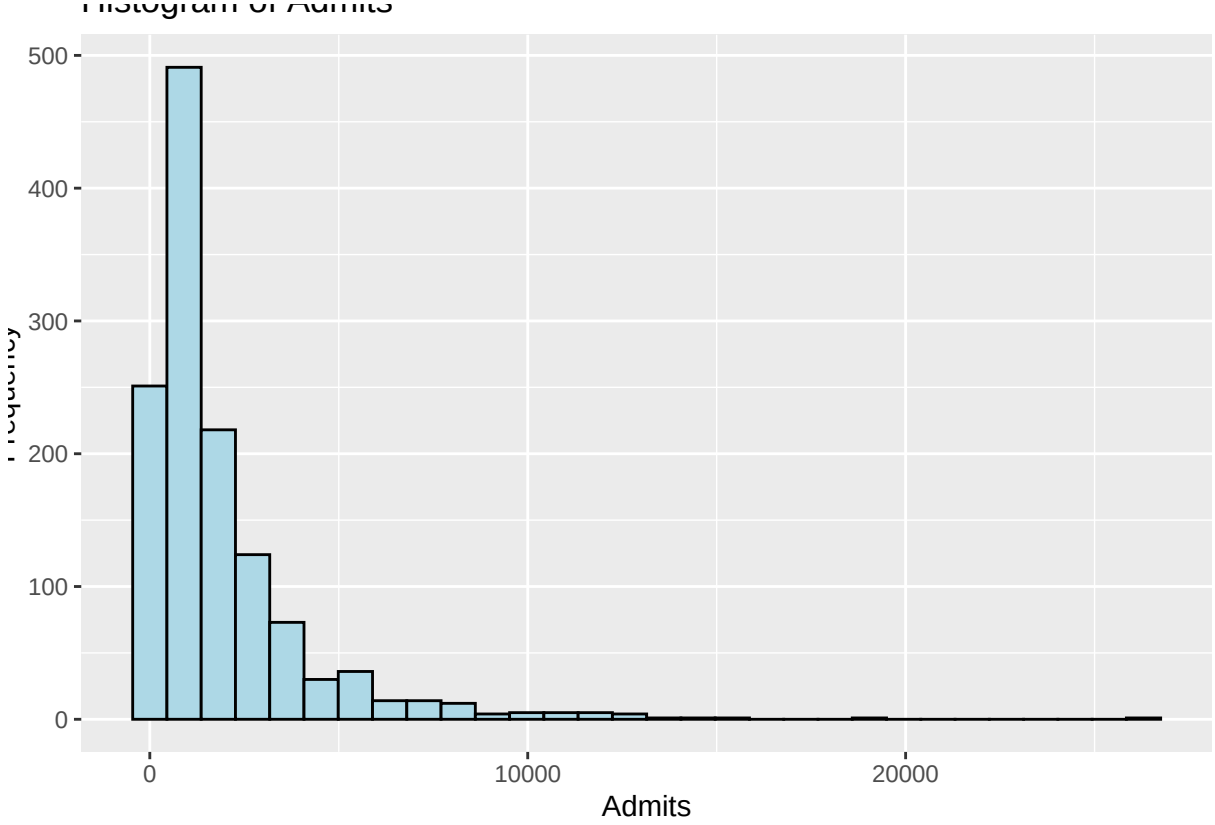


d. Create a histogram of Admits.

```
ggplot(data = uni_rate, aes(Admits)) +
  geom_histogram(col = 'black', fill = 'light blue') +
  labs (title = 'Histogram of Admits',
        x = 'Admits', y = 'Frequency')
```

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

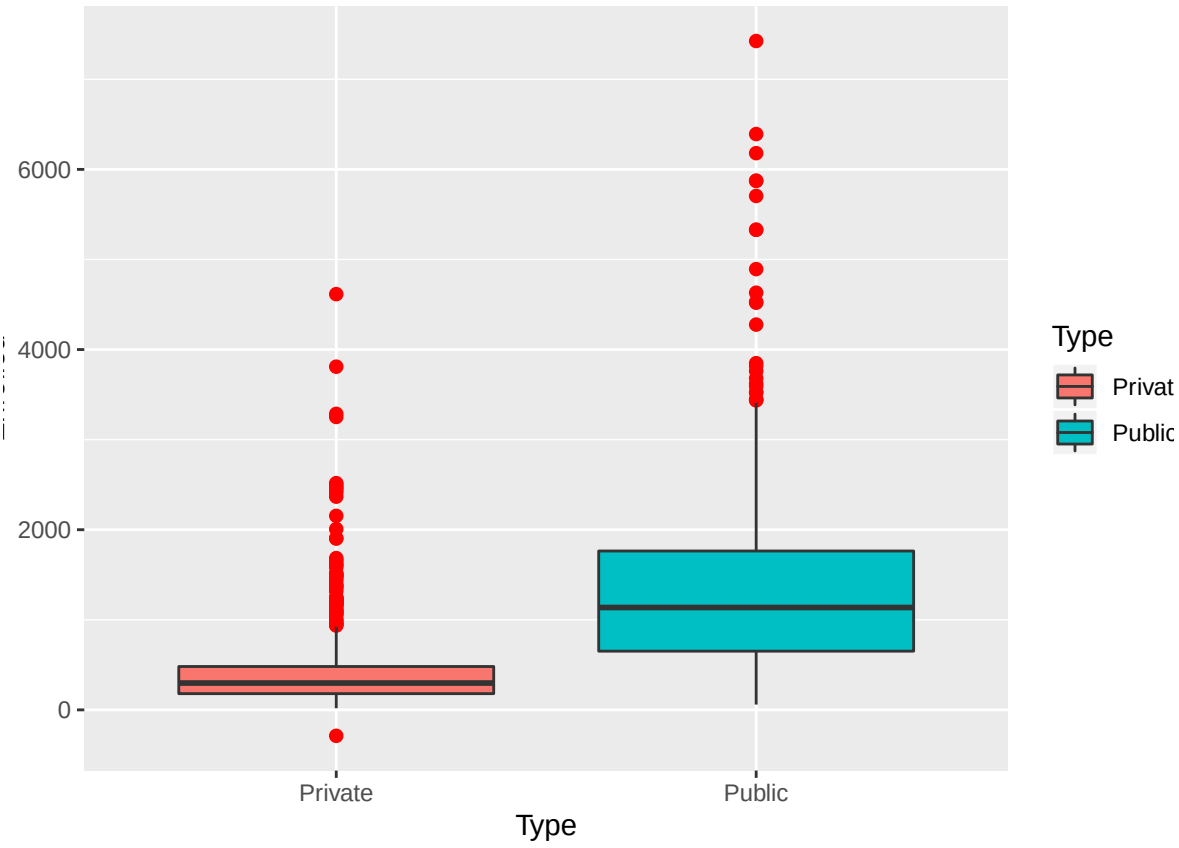
```
## Warning: Removed 11 rows containing non-finite values (stat_bin).
```



e. Create a side by side Box Plot for Enrolled using Type as the by variable.

```
ggplot(uni, aes(Type, Enrolled)) +
  geom_boxplot(aes(fill = Type), outlier.color = 'red',
               outlier.shape = 20, outlier.size = 3)
```

```
## Warning: Removed 5 rows containing non-finite values (stat_boxplot).
```



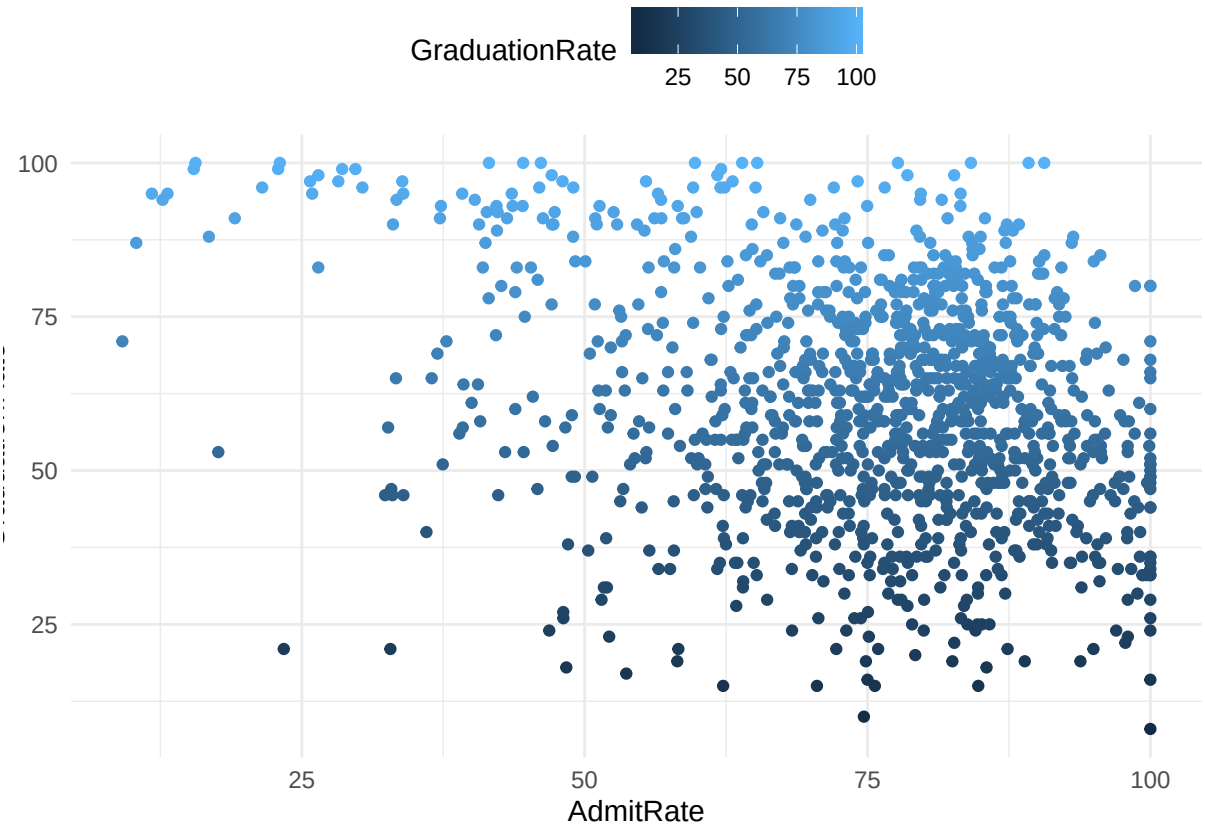

```
  theme(legend.position = 'right')
```

```
## List of 1
##  $ legend.position: chr "right"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

f. Create a scatter plot of AdmitRate and GraduationRate

```
ggplot(uni_rate, aes(AdmitRate, GraduationRate)) +
             geom_point(aes(color = GraduationRate)) + theme_minimal() +
             theme(legend.position = "top")
```

```
## Warning: Removed 106 rows containing missing values (geom_point).
```



g. Plotting correlation matrix

```
library('PerformanceAnalytics')
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

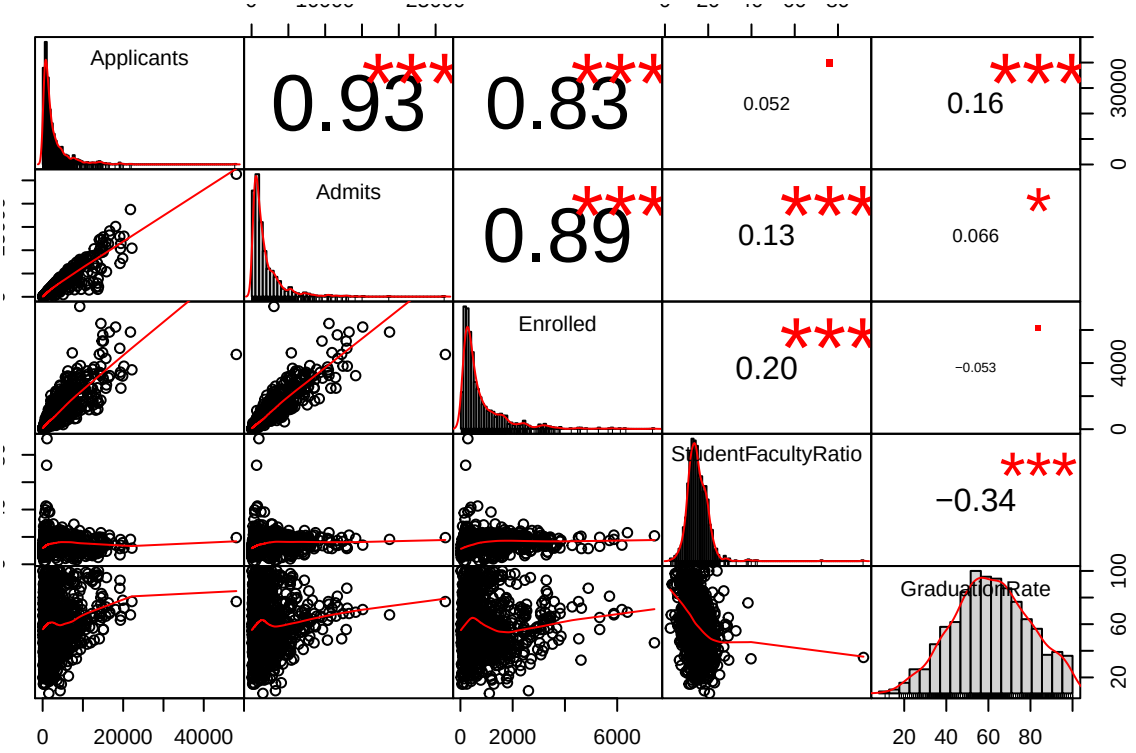
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
##
##     legend
```

```
chart.Correlation(uni_num, histogram = T, pch = 19)
```



h. Find three errors in the data provided.

- The data type of "Admits" in the original data set is Factor, if we directly convert the Factor to Numeric, the values in "Admits" are changed. These will make the whole following process wrong. So I have to convert "Admits" to Character first, and then convert it to Numeric. "This is because factors are stored internally as integers with a table to give the factor level labels" (by James, Stack Overflow)
- In the original data set, some values of "Enrolled" are negative number which is impossible. The method I use is to replace all the negative numbers to NA.
- In the original data set, some values of "GraduationRate" are larger than 100 which don't make sense. The method I use is to replace the abnormal number(>100) to NA.