Stats216v: Statistical Learning

Stanford University
Summer 2017
Gyu-Ho Lee (gyuhox@gmail.com (mailto:gyuhox@gmail.com))

Problem Set 1

Q1. Explain whether it is a supervised learning problem or an unsupervised learning problem. In the case of supervised learning, specify whether it is regression or classification and provide the the number of observations and predictors.

(a) A study in the Netherlands compares the perfomance of students from refugee families and students from Dutch families. In this study for each of 500 students the researchers have grades point average of the students, their country of origin, family's wealth and parent's education.

Gyu-Ho's Answer: Unsupervised, because there is no response variable to predict. It rather seeks to understand the relationships between the observations (e.g. family background, grades).

(b) A group of Stanford students want to make a mobile application that suggests an entertaining actvity from 3 different options to their users based on their current location, age, gender, level of education and the time of the day. Also they have access to the 3000 reviews that people who have done those activities in the past 2 months.

Gyu-Ho's Answer: Supervised because:

- For each observation(student), there are predictors of current location, age, etc..
- Then, there is an associated response variable, *entertaining actvity suggestion*, to predict.

This is classification because:

- Response variable, *entertaining actvity suggestion*, is not numerical, but qualitative.
- We can assign each test observation to the most likely class, *most entertaining activity class*, based on its predictor values, using K-Nearest neighbors classifier.

Number of observations: 3000 (number of reviews). Number of predictors: 5 (current location, age, gender, education, time of the day)

(c) Hospitals have supplies of human blood to use during emergency medical procedures. Some parts of the human blood can last at most five days before it has expired. To prevent blood shortages and wasted blood, a group of statisticians want to know how many bags of blood a hospital will need for each day. For this pupose they have monitored three hospitals for a time period of 2 years. Each day, they record the amount of blood used, the number of patients in the trauma ward, the number of scheduled surgeries, day of the week, and wheter or not it was a holiday.

Gyu-Ho's Answer: Supervised because:

- For each observation(day), there are predictors of amount of blood used, number of patients, etc..
- Then, there is an associated response variable, *number of blood bags needed per day*, to predict.

This is regression because response variable, *number of blood bags needed per day*, is numerical, quantitative.

Number of observations: 730 (number of days). Number of predictors: 5 (amount of blood used, number of patients, number of scheduled surgeries, day of the week, and wheter or not it was a holiday).

(d) A legal firm wishes to gain a better understanding of the types cases that the firm has handled in the last 10 years. A tech-savy lawyer has created a database of with the text of the legal briefs associated with each of the 6000 cases that the firm handled.

Gyu-Ho's Answer: Unsupervised, because there is no response variable to predict. It rather seeks to understand the relationships between the observations (e.g. legal database, case results).

(e) Oil excavation is a very expensive process and the oil resources are not distributed uniformly in an area, so it is important to find the best spots for for oil extraction. To do this engineers consider a very coarse grid (each edge length is of order of miles) and dig a well in the vertices of the grid and take a sample of the sand of the grid points. In one example, 28 different measurements are taken from each sand sample. An engineer has sand samples for 35 locations where they know the results of the digging (how much oil was present at that location). Additionally, the engineer has sand samples for 80 prospective well locations, and wishes to find the most promising spot to dig a future well.

Gyu-Ho's Answer: Supervised because:

- For each observation(location), there are predictors of 28 different measurements from each sand sample.
- Then, there is an associated response variable, *prospective well location* or *how much oil exists*, to predict.

This is regression because response variable, *how much oil exists*, is numerical, quantitative. It seeks to find the most promising well location by estimating the amount of oil that exists.

Number of observations: 80 (total number of prospective locations). Number of predictors: 28 (number of sand sample measurements from each location).

(f) During the world war II when the US was doing nuclear experiments in the Nevada deserts people of the neighborhood area could feel the ground shakings as a result of the explosions and had been thinking that they were ordinary earthquakes. Today, the US wants to see which of the earthquakes in the year 2017 in South Korea are real earthquakes and which are caused by nuclear testing in North Korea. They have historical data from 100 earthquakes in South Korea before North Korea began nuclear testing, as well as measurements from the 18 nuclear tests in the Neveda desert. Each data point is a time series of earthquake intensity sampled 10 times per second for 60 seconds.

Gyu-Ho's Answer: Supervised because:

- For each observation(earthquake), there are predictors of time series of earthquake intensities.
- Then, there is an associated response variable, *nuclear testing or ordinary earthquake*, to predict.

This is classification because response variable, *whether or not earthquake is from a nuclear test*, is not numerical, but qualitative.

Number of observations: 118 (number of earthquakes). Number of predictors: 600 (number of time series data points).

Q2 (a). How would you explain the difference between supervised and unsupervised learning? Can you give an example for each case that describes the differences?

Gyu-Ho's Answer: **Supervised learning** has outcome, *or dependent*, **variable** to guide statistical learning process. It seeks to predict **response variable** from **input variables**. **Unsupervised learning** does not measure or predict the outcome: **only observe features**. It seeks to understand how data are organized or clustered. Linear regression method is supervised learning, that can fit a model based on training observations and predict the response of previously unseen test data. For instance, linear regression can predict someone's wage, based on education level and seniority. In contrast, unsupervised learning wants probability distribution of entire dataset, or groups data into clusters of similar features. For instance, in marketing data of hundreds of features (age, gender, income, etc.), unsupervised learning can group those individuals to their shared demographic features.

Q2 (b). How would you explain the difference between regression and classification? Can you give an example where a regression methods could be used for classification? An example where a classification method could not be used for regression.

Gyu-Ho's Answer: Regression predicts numerical values, *such as stock price*, while classification predicts qualitative values, *such as posivie or negative*. Regression method can be used to predict qualitative values of levels. For instance, each observation is a student's previous test scores. And expect to estimate a response variable, *letter grade in the final exame (A, B, C, D, F)*. Regression first predicts the *test score out of 100 scale*, which can be converted to *letter grades* with certain thresholds. Regression assigns a numberical value to each response variable, whereas logistic regression, *or classification*, assigns probability, *or discrete values*. A classification method can not be used when regression requires explit ordering in output classes. For instance, classification method for predicting *letter grades in final exams* cannot be used to regress *numerical test score values*.

Q3. You are a data science consultant! In each of the following cases decide whether you would suggest a flexible regression model or an inflexible one. Provide your reasons for your client as clearly as possible.

(a) In the study of breast cancer, scientist are trying to find the associated genes. The total number of genes in the study is 50000 and the number of patients is 120.

Gyu-Ho's Answer: **Inflexible regression** because the number of predictors p is extremely large, *5000*, and the sample size n is relatively small, *120*. Flexible model would decrease *bias*, but might overfit the data due to small sample size. Plus, the study tries to find the associated genes, rather than predicting on unseen test data. Thus, it is more about **inference** to understand relationship between genes and cancer.

(b) The ministry of education in a certain country wants to identify student who need extra help. They wish to design a system which estimate student performance in the final 8th grad math exam based on their math, science and history grades in the 7th grade. To do this they want to run a regression on the data consisting of the information of all the students who have graduated from the 8th level in the last 10 years.

Gyu-Ho's Answer: **Flexible regression**, because the number of predictors p is relatively small, *3*, and the sample size n is extremely large, *all students who have graduated from the 8th level in the last 10 years*. Inflexible model would decrease *variance* being more interpretable but *highly biased* thus less accurate. Moreover, the ministry of education is not interested in understanding the relationships between variables. Thus, there is no need to choose trade-offs between flexibility and interpretability.

(c) Kelly is a very hardworking chemistry student and she has run an experiment to find a mathematical expression that explains the speed of corrosion of iron according to the humidity and temperature of the envirnoment and the percentage of different elements in the alloy. Unfortunately the lab that she is working in was established in 1967 and the equipment has not been changed seen then. She is skeptical about the quality of her measurements of the speed of corrosion.

Gyu-Ho's Answer: **Inflexible regression**, because flexible regression would be fitting irreducible error terms too closely. When $\sigma^2 = Var(\epsilon)$ is high due to poor quality of old lab facilities, flexible method would overfit to those errors.

(d) Kelly's advisor won the chemistry nobel prize and used the prize money to outfit the lab with the most modern equipment. Kelly has run her experiments again with the new equipments and now she can trust her numbers. But her advisor believes that she should not expect that the real relationship be linear.

Gyu-Ho's Answer: **Flexible regression** to find the nonlinear relationships between variables. For instance, linear regression, which is a **inflexible method**, assumes the linear relationship **with high bias**. If the true relationship is not linear, the model won't be accurate.

Q4. This exercise relates to the College dataset, which can be found in the file College.csv. It contains a number of variables for 777 different universities and colleges in the US.

(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.

In [23]:
```
college = read.csv("chapter-02-hw-College.csv", header=TRUE,
na.strings="?")
# data = na.omit(college)
names(college)
dim(college)
```

'X' 'Private' 'Apps' 'Accept' 'Enroll' 'Top10perc' 'Top25perc' 'F.Undergrad' 'P.Undergrad' 'Outstate' 'Room.Board' 'Books' 'Personal' 'PhD' 'Terminal' 'S.F.Ratio' 'perc.alumni' 'Expend' 'Grad.Rate'

777 19

(b) To acces documentation within R, you can use the ? operator. As an example, try ? read.csv. Based on this documentation, if our file did not have a header row how should we modify the call to read.csv in part a?

```
# if the file did not have a header
read.csv("chapter-02-hw-College.csv", header=FALSE, na.strings="?")
```

(c) Look at the data using the View() function. You should notice that the first column is just the name of each university. We don't really want R to treat this as data. However, it may be handy to have these names for later. Try the following commands:

```
rownames(college)=college[,1]
fix(college)
```

You should see that there is now a `row.names` column with the name of each university recorded. This means that R has given each row a name corresponding to the appropriate university. R will not try to perform calculations on the row names. However, we still need to eliminate the first column in the data where the names are stored. Try

```
college=college[,-1]
View(college)
```

Now you should see that the first data column is `Private`. Note that another column labeled `row.names` now appears before the `Private` column. However, this is not a data column but rather the name that R is giving to each row.

In [24]:
```
# to list all row numbers(names) 'rownames(college)'
# '1' '2' '3' ...

# to list all rows in first column 'college[,1]'
# Abilene Christian University Adelphi University ...

# to overwrite row numbers with first column
rownames(college)=college[,1]
# rownames(college)
# 'Abilene Christian University' 'Adelphi University' ...
```

In [25]:
```
# 'college' still has two redundant columns
# all columns but first column
# college[,-1]
college[1:1,]

# to remove first column 'X' and just keep row names
college=college[,-1]
college[1:1,]
```

	X	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	I
Abilene Christian University	Abilene Christian University	Yes	1660	1232	721	23	52	2885	5

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergra
Abilene Christian University	Yes	1660	1232	721	23	52	2885	537

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

In [26]:
```
summary(college)
```

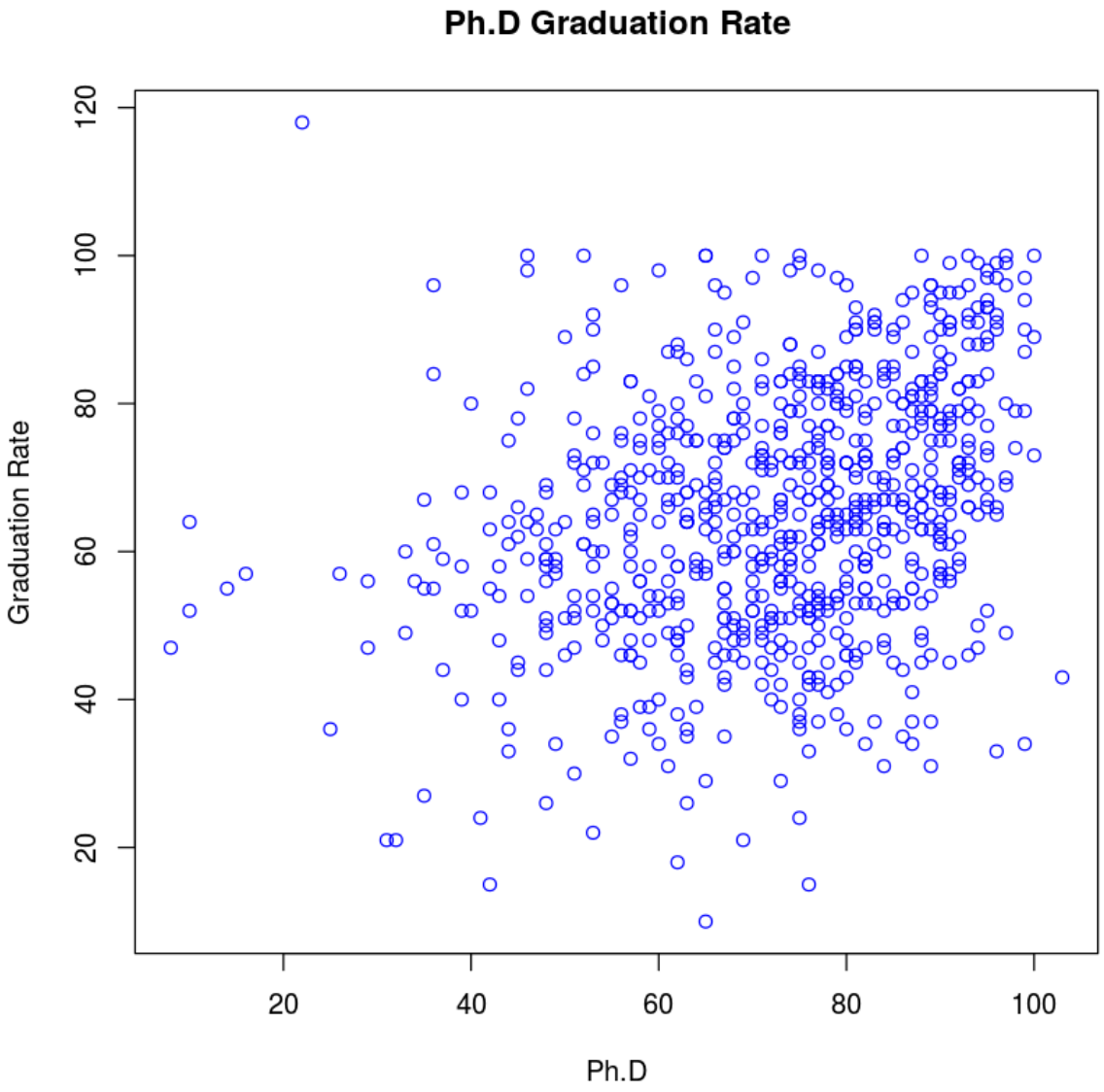
```
 Private        Apps           Accept          Enroll       Top10perc
 No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
 Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
           Median : 1558   Median : 1110   Median : 434   Median :23.00
           Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
           3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
           Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00

  Top25perc      F.Undergrad     P.Undergrad         Outstate
Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
  Room.Board       Books           Personal         PhD
Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
Median :4200   Median : 500.0   Median :1200   Median : 75.00
Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
   Terminal       S.F.Ratio      perc.alumni        Expend
Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
Median : 82.0   Median :13.60   Median :21.00   Median : 8377
Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
  Grad.Rate
Min.   : 10.00
1st Qu.: 53.00
Median : 65.00
Mean   : 65.46
3rd Qu.: 78.00
Max.   :118.00
```

(e) Use the `plot()` function to produce a scatterplot of the column `PhD` versus the column `Grad.Rate`.

In [27]:
```
names(college)
plot(college$PhD, college$Grad.Rate, xlab="Ph.D", ylab="Graduation
Rate", main="Ph.D Graduation Rate", col="blue")
```

'Private' 'Apps' 'Accept' 'Enroll' 'Top10perc' 'Top25perc' 'F.Undergrad' 'P.Undergrad' 'Outstate' 'Room.Board' 'Books' 'Personal' 'PhD' 'Terminal' 'S.F.Ratio' 'perc.alumni' 'Expend' 'Grad.Rate'



(f) Use the `which()` and `length()` function to see how many of these colleges are private.

In [28]:
```
length(which(college$Private == "Yes"))
```

565

(g) Create a new qualitative variable, called `Elite`, by *binning* the `Top10perc` variable. We are going to divide universities into two groups based on whether or not the proportion of students coming from the top 10 of their high school classes exceeds 50.

```
Elite = rep("No",nrow(college))
Elite[college$Top10perc >50] = "Yes"
Elite = as.factor(Elite)
college = data.frame(college ,Elite)
```

Use the `summary()` function to see how many elite universities there are. Now use the `plot()` function to produce side-by-side boxplots of `Outstate` versus `Elite`.

In [29]:
```
# to create a new column vector
Elite = rep("No", nrow(college))
Elite[3:5] # 'No' 'No' 'No'

# to get all rows where top  1010  of their high school classes exceeds
 50
# 'college$Top10perc >50'
Elite[college$Top10perc >50] = "Yes"
Elite[3:5] # 'No' 'Yes' 'No'

# to convert quantitative to qualitative variables
Elite = as.factor(Elite)

# to add a new column
college = data.frame(college, Elite)
college[1:1,]

# to see how many elite universities there are
length(which(college$Elite == "Yes")) # 78
summary(college$Elite) # Yes: 78

plot(college$Elite,
     college$Outstate,
     xlab="Elite University",
     ylab="Outstate Tuition",
     main="Outstate Tution (Elite Universities Yes/No)",
     col="blue")
```

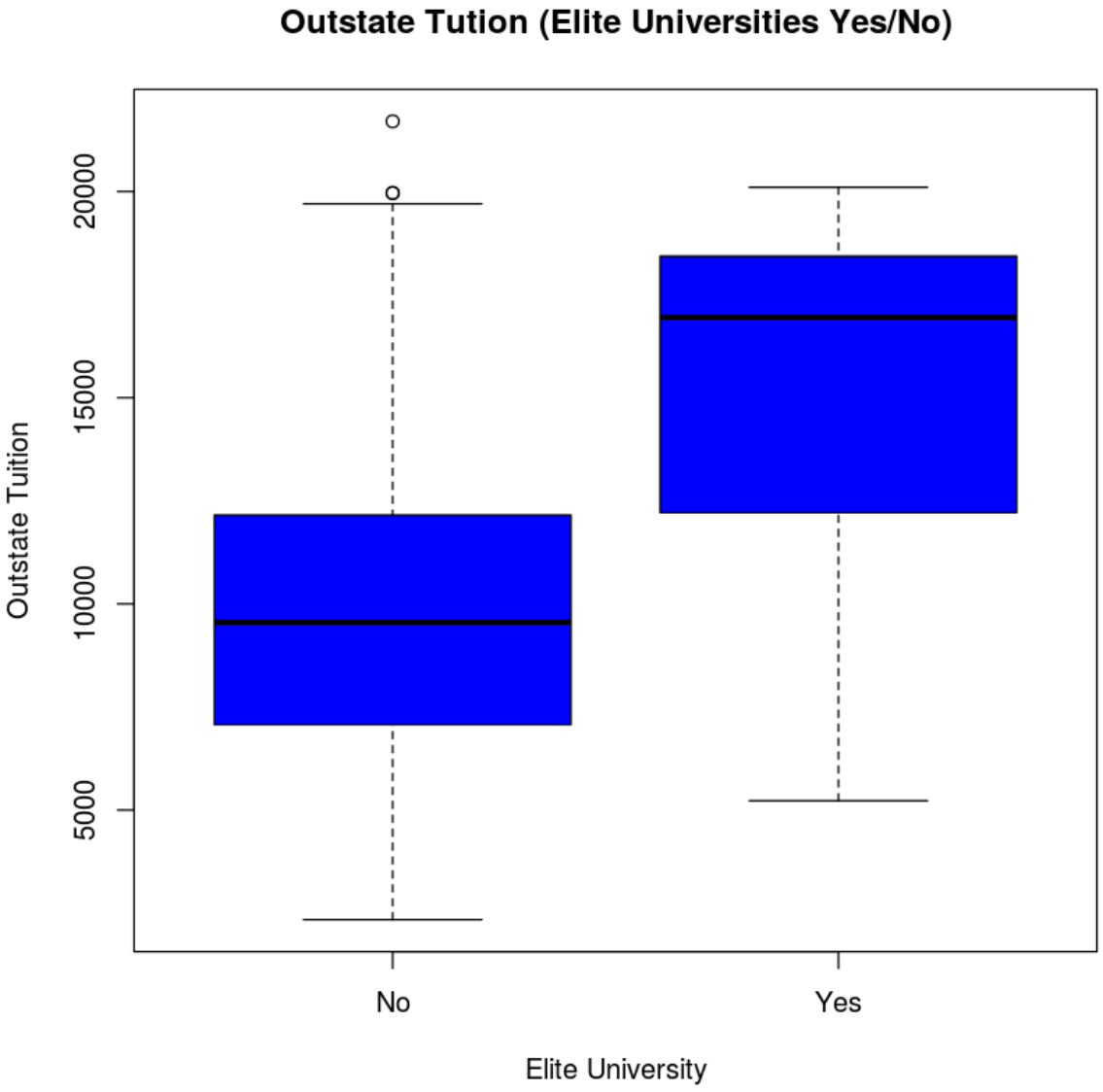
'No' 'No' 'No'

'No' 'Yes' 'No'

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergra
Abilene Christian University	Yes	1660	1232	721	23	52	2885	537

78

No	699
Yes	78



(h) Use the `hist()` function to produce 4 histograms with differing numbers of bins for 4 different quantitative variables. Use the command $par(mfrow = c(2, 2))$ to divide the print window into four regions so that four plots can be made simultaneously.

Note: modifying the arguments to the $par(mfrow = c())$ command can be used divide the screen in other ways, which will be useful in future assignments.

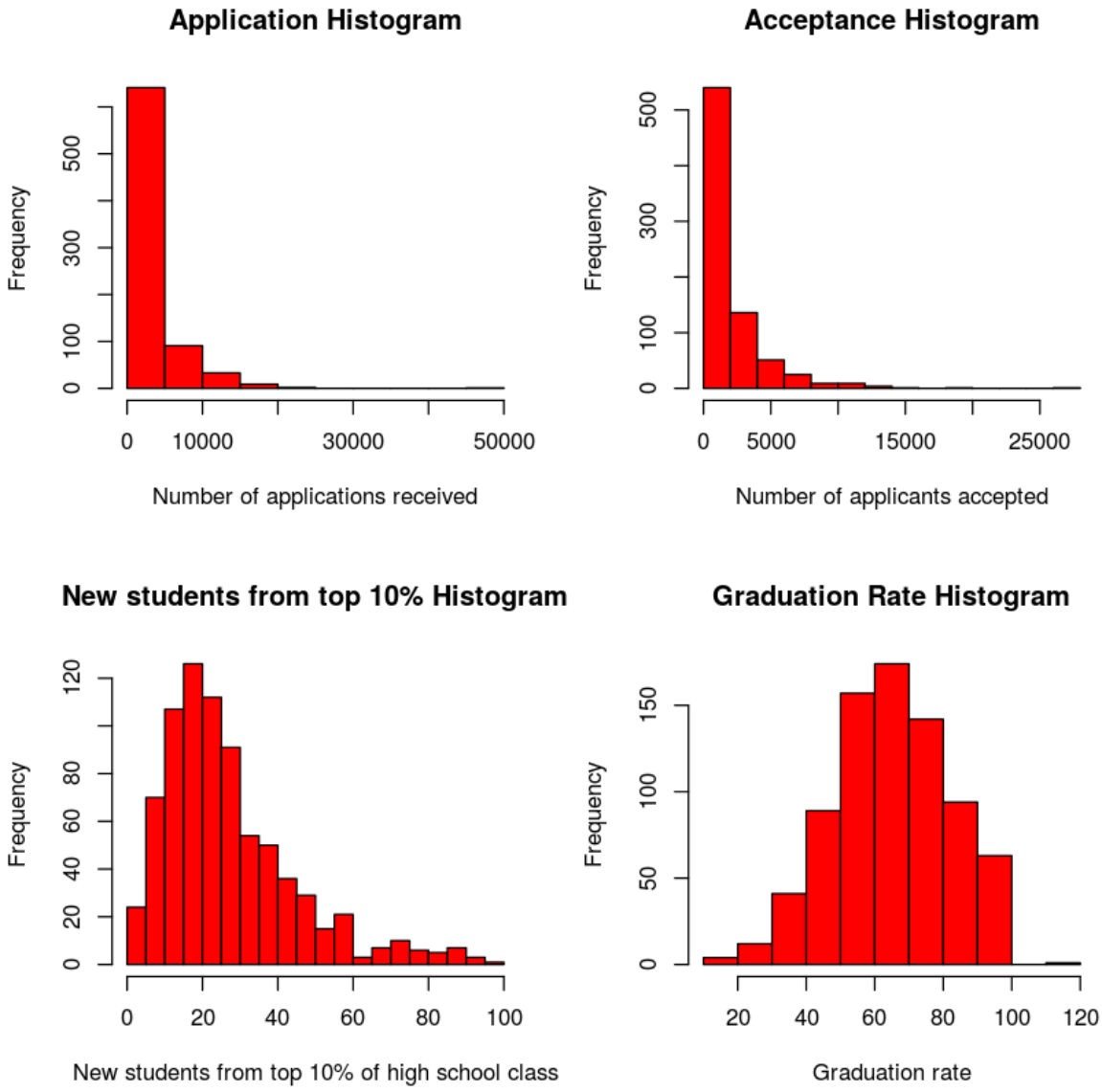
In [30]:
```
# to divide the screen
op <- par(mfrow = c(2, 2))

hist(college$Apps,
     col=2,
     breaks=15,
     xlab="Number of applications received",
     ylab="Frequency",
     main="Application Histogram")

hist(college$Accept,
     col=2,
     breaks=15,
     xlab="Number of applicants accepted",
     ylab="Frequency",
     main="Acceptance Histogram")

hist(college$Top10perc,
     col=2,
     breaks=15,
     xlab="New students from top 10% of high school class",
     ylab="Frequency",
     main="New students from top 10% Histogram")

hist(college$Grad.Rate,
     col=2,
     breaks=15,
     xlab="Graduation rate",
     ylab="Frequency",
     main="Graduation Rate Histogram")
```



Q5.

For this problem, we will consider the ALS dataset. `ALS` (amyotrophic lateral sclerosis), or Lou Gehrig's disease, is a fatal neurodegenerative disease with no known cure and few known causes. In July of 2012, Prize4Life launched a challenge to most accurately predict the rate of progression of `ALS` in patients.1 The `ALS` Prediction `Prize4Life` Challenge featured a subset of the `PRO-ACT` database, the largest compilation of `ALS` clinical trial data ever assembled. The `ALS` dataset is drawn from this challenge, and is worth getting acquainted with: it will be used in the four problem sets in this course. Our goal will be to predict the rate of progression of ALS in patients.

The `ALS` dataset is composed of four objects: the recorded values in `train.y` and `test.y` represent the rate of change of that patient's `ALS` Functional Rating Score (a 40-point measure of a person's ability to carry out everyday tasks like walking, speaking, swallowing, dressing, etc.) over the final 9 months of the trial. The values in `train.X` and `test.X` contain the predictors we will be using — there are 323 of them.

(a) First, load the `als.RData` file using the load command. Make sure that you have the directory set to the correct location for the data.

In [31]:
```
loaded = load("chapter-02-hw-als.RData")
head(loaded) # 'train.X' 'train.y' 'test.X' 'test.y'

rData = get(loaded)
head(rData)

ls() # 'Elite' 'college' 'isfar' 'loaded' 'op' 'rData' 'test.X' 'test.y'
 'train.X' 'train.y'
head(test.X)

# names(rData)
# ?get
```

'train.X' 'train.y' 'test.X' 'test.y'

Onset.Delta	Symptom.Speech	Symptom.WEAKNESS	Site.of.Onset.Onset..Bulbar	Site.o
-341	1	1	1	0
-1768	0	1	0	1
-334	1	0	1	0
-268	0	1	0	1
-440	0	1	0	1
-773	0	1	0	1

'Elite' 'college' 'lm.model' 'lm.test.pred' 'loaded' 'op' 'rData' 'test.X' 'test.y' 'train.X' 'train.y'

Onset.Delta	Symptom.Speech	Symptom.WEAKNESS	Site.of.Onset.Onset..Bulbar	Site.o
-1181	1	0	1	0
-1324	0	1	0	1
-1061	0	0	0	1
-1736	0	1	0	1
-354	1	0	1	0
-500	1	1	1	0

(b) Each entry in `train.y` or `test.y` corresponds to an `ALS` patient in a 12-month clinical trial. What are the lengths of each object? Use the `length()` function.

In [32]:
```
length(train.y) # 1197
length(test.y) # 625
```

1197

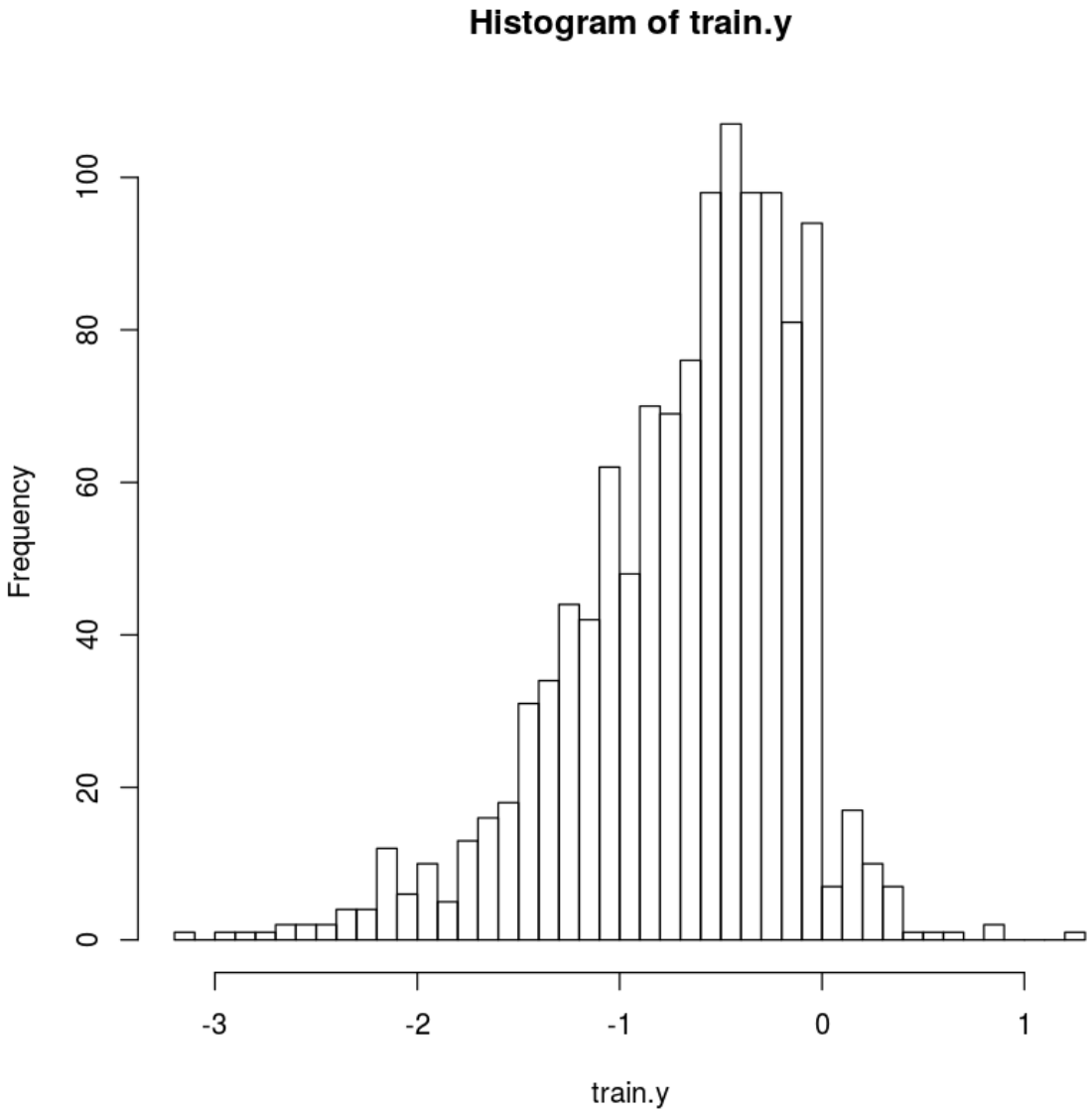
625

(c) Look at the information in `summary(train.y)`, and produce a histogram of `train.y`, and use the parameter `breaks=40` to ensure you have enough bins. What are some striking features of the distribution of the target values in `train.y`?

Gyu-Ho's Answer: The distribution of target values in train.y is heavily skewed towards median, -0.5786.

In [33]:
```
summary(train.y)
hist(train.y, breaks=40)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-3.1030 -1.0150 -0.5786 -0.6802 -0.2568  1.2080 
```

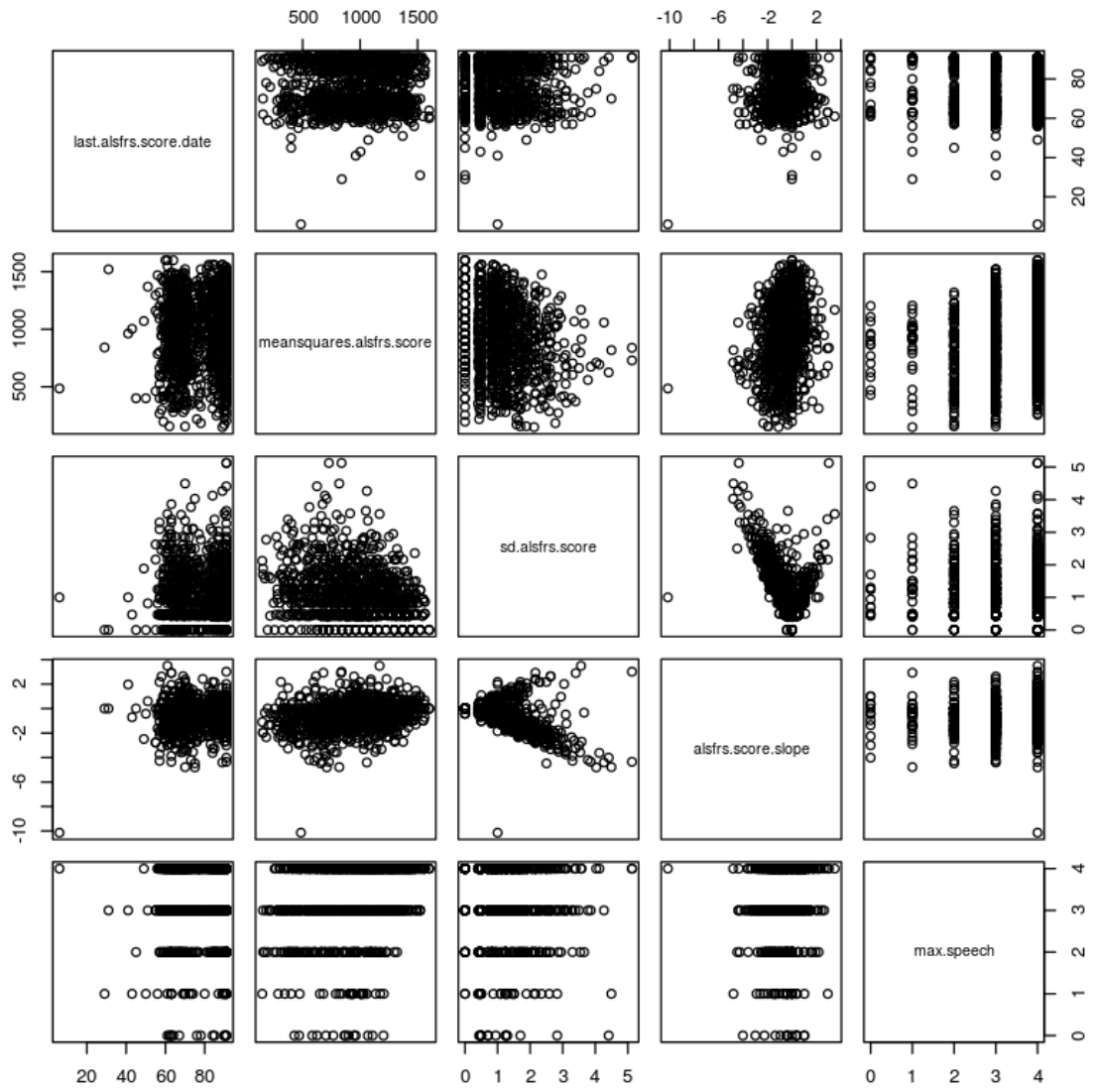


(d) Investigate the names of the first few columns in `train.X` and look at some pairwise plots by trying

```
colnames(train.X)[1:20]
pairs(train.X[,21:25])
```

In [34]:
```
colnames(train.X)[1:20]
pairs(train.X[,21:25])
```

'Onset.Delta' 'Symptom.Speech' 'Symptom.WEAKNESS'
'Site.of.Onset.Onset..Bulbar' 'Site.of.Onset.Onset..Limb' 'Race...Caucasian'
'Age' 'Sex.Female' 'Sex.Male' 'Mother' 'Family' 'Study.Arm.PLACEBO'
'Study.Arm.ACTIVE' 'max.alsfrs.score' 'min.alsfrs.score' 'last.alsfrs.score'
'mean.alsfrs.score' 'num.alsfrs.score.visits' 'sum.alsfrs.score'
'first.alsfrs.score.date'



(e) Fit a linear regression to your model using only the train.X data via the `lm()` function, and call it `lm.model`. It is ok if you get a warning. Since there are too many predictors, look at the first few coefficients fitted and the R^2 statistic using the following commands:

```
coef(summary(lm.model))[1:20, 1:4]
summary(lm.model)$r.squared
```

In [37]:
```
lm.model = lm(train.y ~ ., data=train.X)
coef(summary(lm.model))[1:20, 1:4]
summary(lm.model)$r.squared
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	5.2523261623	1.005248e+01	0.522490777	6.014605e-01
Onset.Delta	-0.0004375411	4.516149e-05	-9.688368476	3.700894e-21
Symptom.Speech	-0.0225458435	9.120369e-02	-0.247203188	8.048088e-01
Symptom.WEAKNESS	-0.1030724618	8.140383e-02	-1.266186858	2.057820e-01
Site.of.Onset.Onset..Bulbar	-0.3672390249	2.492628e-01	-1.473300789	1.410284e-01
Site.of.Onset.Onset..Limb	-0.2722652619	2.516364e-01	-1.081979013	2.795589e-01
Race...Caucasian	-0.1472308983	9.490391e-02	-1.551368083	1.211739e-01
Age	-0.0005163468	1.814937e-03	-0.284498398	7.760955e-01
Sex.Female	-0.0605416162	9.573716e-02	-0.632373219	5.273077e-01
Sex.Male	0.0195376288	8.868793e-02	0.220296371	8.256916e-01
Mother	-0.0462694066	7.581925e-02	-0.610259353	5.418479e-01
Family	0.0067961369	5.629768e-02	0.120717895	9.039421e-01
Study.Arm.PLACEBO	-3.1549354316	1.918113e+00	-1.644811989	1.003665e-01
Study.Arm.ACTIVE	-3.0519454033	1.917047e+00	-1.592003115	1.117439e-01
max.alsfrs.score	0.0622267218	7.830124e-02	0.794709319	4.269974e-01
min.alsfrs.score	-0.1666353650	8.465840e-02	-1.968326306	4.934477e-02
last.alsfrs.score	0.4090457993	2.074172e-01	1.972092254	4.891257e-02
mean.alsfrs.score	-0.1208903623	3.538618e-01	-0.341631529	7.327100e-01
num.alsfrs.score.visits	-0.0058420080	7.139877e-01	-0.008182225	9.934735e-01
sum.alsfrs.score	-0.0778763045	7.899171e-02	-0.985879479	3.244639e-01

0.463117270326138

(f) Find the test root mean squared error (`RMSE`) of your model. The `predict()` function will probably be useful.

In [38]:
```
lm.test.pred = predict(lm.model, test.X)
sqrt(mean((lm.test.pred - test.y)^2))
```

```
Warning message in predict.lm(lm.model, test.X):
"prediction from a rank-deficient fit may be misleading"
```

0.752712245034207

(g) As we shall later see, the error rate produced by using a simple linear regression on this data is much too high. What could account for this? Try to relate your answer to the bias-variance trade-off.

Gyu-Ho's Answer: High error rate in linear regressions indicates possible overfitting, where the model follows the noise too closely. As bias-variance trade-off explains, flexible methods often show low bias with more predictors. However, adding irrelevant predictors could cause high bias and variance in flexible methods, thus poor test error rates.