MIS2502: Data Analytics

*Introduction to Advanced Analytics and R*
The Information Architecture of an Organization

Transactional Database
- Stores real-time transactional data

Analytical Data Store
- Stores historical transactional and summary data

Data entry
Data extraction
Data analysis

Now we’re here...
The difference between OLAP and data mining

OLAP can tell you what is happening, or what *has* happened.

Data mining can tell you *why* it is happening, and help predict what *will* happen.

The (dimensional) data warehouse feed both...

...like a pivot table

...like what we’ll do with R
Data Mining and Predictive Analytics is

Extraction of implicit, previously unknown, and potentially useful information from data

Exploration and analysis of large data sets to discover meaningful patterns
What data mining is not...

- **Sales analysis**
  - How do sales compare in two different stores in the same state?

- **Profitability analysis**
  - Which product lines are the highest revenue producers this year?

- **Sales force analysis**
  - Did salesperson X meet this quarter’s target?

If these aren’t data mining examples, then what are they?
Example: Smarter Customer Retention

• Consider a marketing manager for a brokerage company

• Problem: High churn (customers leave)
  – Customers get an average reward of $160 to open an account
  – 40% of customers leave after the 6 month introductory period
  – Giving incentives to everyone who might leave is expensive
  – Getting a customer back after they leave is expensive
Answer: Not all customers have the same value

One month before the end of the introductory period, *predict which customers will leave*

Offer those customers something based on their *future value*

*Ignore* the ones that are not predicted to churn
Three Analytics Tasks We Will Be Doing in this Class

- Decision Tree Induction
- Clustering
- Association Rule Mining
**Decision Trees**

Used to classify data according to a pre-defined outcome

Based on characteristics of that data

**Uses**

- Predict whether a customer should receive a loan
- Flag a credit card charge as legitimate
- Determine whether an investment will pay off
Clustering

Used to determine distinct groups of data

Based on data across multiple dimensions

Uses

Customer segmentation

Identifying patient care groups

Performance of business sectors

Association Mining

Find out which events predict the occurrence of other events

Often used to see which products are bought together

Uses

What products are bought together?

Amazon’s recommendation engine

Telephone calling patterns
Introduction to R and RStudio
• Software development platform and language
• Open source
• Many, many, many statistical add-on “packages” that perform data analysis

(The base/engine)

• Integrated Development Environment for R
• Nicer interface that makes R easier to use
• Requires R to run

(The pretty face)
RStudio Interface

- **Console** (just like a command line)
- **Environment** for info of your data
- **History** for Previous commands

It may have an additional window for R script(s) and data view if you have any of them open.

- **Files**
- **Plots**
- **Packages**
- **Help**
- **Viewer**
The Basics: Calculations

- R will do math for you:

```r
> 12+23
[1] 35
> sqrt(100)
[1] 10
> 15/2
[1] 7.5
> pi
[1] 3.141593
> 2^4
[1] 16
> log(10)
[1] 2.302585
> abs(-4)
[1] 4
> exp(2)
[1] 7.389056
> #THis is a comment line
```

Type commands into the `console` and it will give you an answer
The Basics: Variables

- Variables are named containers for data.
- The assignment operator in R is: \(<-\) or \(=\).
- Variable names can start with a letter or digits.
  - Just not a number by itself.
  - Examples: `result`, `x1`, `2b` (not 2).
- R is case sensitive (i.e. `Resul t` is a different variable than `r esul t`).

$$>	ext{x}=5 \quad \text{y}= -10 \quad \text{z}=8$$
$$>	ext{name}<"David"$$
$$>	ext{x+y-z} \quad \text{[1]} \quad 7$$

\(\text{rm(x)}\) removes the variable from memory.

\(<-\) and \(=\) do the same thing.

\(x, y, \text{and } z\) are variables that can be manipulated.
## Basic Data Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Range</th>
<th>Assign a Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Numeric</strong></td>
<td>Numbers</td>
<td>x &lt;- 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>y &lt;- -2.5</td>
</tr>
<tr>
<td><strong>Character</strong></td>
<td>Text strings</td>
<td>name &lt;- &quot;Mark&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>col or &lt;- &quot;red&quot;</td>
</tr>
<tr>
<td><strong>Logical (Boolean)</strong></td>
<td>TRUE or FALSE</td>
<td>female &lt;- TRUE</td>
</tr>
</tbody>
</table>
Vectors of values

• A vector is a sequence of data elements of the same basic type.

```r
> scores <- c(65, 75, 80, 88, 82, 99, 100, 100, 50)
> scores
[1] 65 75 80 88 82 99 100 100 50
> student.num <- 1:9
> student.num
[1] 1 2 3 4 5 6 7 8 9
> ones <- rep(1, 4)
> Ones
[1] 1 1 1 1
> sort(scores)
[1] 50 65 75 80 82 88 99 100 100
> scores
[1] 65 75 80 88 82 99 100 100 50
> names <- c("Nikita", "Dexter", "Sherlock")
> names
[1] "Nikita" "Dexter" "Sherlock"
```

c(), rep(), and sort() are functions

Functions accept parameters (arguments) and return a value

Note that sort() puts the scores in order but doesn’t change the original collection
Indexing Vectors

• We use brackets \([\ ]\) to pick specific elements in the vector.

• In R, the index of the first element is 1

```r
> scores
[ 1]  65  75  80  88  82  99 100 100  50
> scores[1]
[1]  65
> scores[2:3]
[1]  75  80
> scores[c(1,4)]
[1]  65  88
```
Simple statistics with R

• You can get descriptive statistics from a vector

```r
> scores
[1] 65 75 80 88 82 99 100 100 50
> length(scores)
[1] 9
> min(scores)
[1] 50
> max(scores)
[1] 100
> mean(scores)
[1] 82.11111
> median(scores)
[1] 82
> sd(scores)
[1] 17.09857
> var(scores)
[1] 292.3611
> summary(scores)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
 50.000 75.000  82.000  82.111  99.000 100.000
```
Packages

• Packages (add-on) are collections of R functions and code in a well-defined format.

• To install a package:
  `install.packages("pysch")`

• To load the package into the current session to be used:
  `library(psych)`
  or
  `require(psych)`
Creating and opening a .R file

• The R script is where you keep a record of your work in R and RStudio.

• To create a .R file
  – Click “File|New File|R Script” in the menu

• To save the .R file
  – click “File|Save”

• To open an existing .R file
  – click “File|Open File” to browse for the .R file
Working directory

• The working directory is where Rstudio will look first for scripts and files

• Keeping everything in a self contained directory helps organize code and analyses

• Check you current working directory with

  \texttt{getwd()}

\texttt{\textcolor{blue}{get wd( )}}
To change working directory

Use the **Session | Set Working Directory** Menu

- If you already have an .R file open, you can select “Set Working Directory>To Source File Location”.

![Session Menu](image.png)
Reading data from a file

• Usually you won’t type in data manually, you’ll get it from a file
• Example: 2009 Baseball Statistics
  http://www2.stetson.edu/~jrasp/data.htm

reads data from a CSV file and creates a data frame called teamData that store the data table.

A data frame is a type of variable used for storing data tables.

reference a column in the data frame using datasetname$ columnname
Looking for differences across groups: The setup

- We want to know if National League (NL) teams scored more runs than American League (AL) Teams
  - And if that difference is statistically significant
- To do this, we need a package that will do this analysis
  - In this case, it’s the “psych” package
Looking for differences across groups: The analysis

Descriptive statistics, broken up by group (League)

```r
> describeBy(teamData$Runs, teamData$League)
group: AL
  vars  n  mean   sd median trimmed   mad  min  max  range  skew  kurtosis  se
  1   14 781.21 75.68    778  781.83  56.34  640  915   275  0.04   -0.84  20.23

group: NL
  vars  n  mean   sd median trimmed   mad  min  max  range  skew  kurtosis  se
  1   16 717.56 61.30    715  716.07  85.25  636  820  184  0.15   -1.46  15.33

> t.test(teamData$Runs ~ teamData$League);

      Welch Two Sample t-test

data:  teamData$Runs by teamData$League
t = 2.5082, df = 25.055, p-value = 0.01897
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  11.39211 115.91147
sample estimates:
mean in group AL mean in group NL 781.2143      717.5625
```

Results of t-test for differences in Runs by League)
Histogram

```r
hist(teamData$BattingAvg, 
  xlab="Batting Average", 
  main="Histogram Batting Average")
```

- hist() first parameter – data values
- xlab parameter – label for x axis
- main parameter - sets title for chart
Plotting data

`plot(teamData$BattingAvg, teamData$WinningPct,
     xlab="Batting Average",
     ylab="Winning Percentage",
     main="Do Teams With Better Batting Averages Win More?")`

`plot()`

- first parameter – x data values
- second parameter – y data values
- xlab parameter – label for x axis
- ylab parameter – label for y axis
- main parameter - sets title for chart
Drawing a regression (trend) line

```r
plot(teamData$BattingAvg, teamData$WinningPct, 
     xlab="Batting Average", 
     ylab="Winning Percentage", 
     main="Do Teams With Better Batting Averages Win More?")

reg1 <- lm(teamData$WinningPct ~ teamData$BattingAvg)
abline(reg1)
```

Calculates the regression line (`lm()`)
And plots the line (`abline()`)

---

Do Teams with Better Batting Averages Win More?

```
Batting Average
```

```
Winning Percentage
```
But is the correlation statistically significant?

So we can say:
“Teams with a better overall batting average tend to have a better winning percentage.”

```
> install.packages("Hmisc")
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/Hmisc_3.17-0.zip'
Content type 'application/zip' length 1627957 bytes (1.6 MB)
downloaded 1.6 MB
package 'Hmisc' successfully unpacked and MD5 sums checked
The downloaded binary packages are in
C:\Users\David\AppData\Local\Temp\Rtmpa4B2Xi\downloaded_packages
> library("Hmisc")
Loading required package: grid
Loading required package: lattice
Loading required package: survival
Loading required package: Formula
Loading required package: ggplot2
Attaching package: ‘Hmisc’
The following objects are masked from ‘package:base’:
    format.pval, round.POSIXt, trunc.POSIXt, units
> rcorr(teamData$BattingAvg,teamData$WinningPct)
   x   y
x 1.00 0.46
y 0.46 1.00
n= 30

> p
   x    y
x 0.0097
y 0.0097
> |
```

“medium” strength correlation

strongly statistically significant
Running this analysis as a script

```
# Adapted from David Schuff, Professor of MIS, Temple University

# Install and load required packages as needed
if (!require("psych")) { install.packages("psych") }
if (!require("Hmisc")) { install.packages("Hmisc") }

# Import data from a .csv file
teamData <- read.csv("2009BaseballTeamStats.csv")

# Descriptive statistics
summary(teamData$BattingAvg)
summary(teamData$WinningPct)

# Descriptive statistics by league
describeBy(teamData$Runs, teamData$League)

# t-test for differences in average runs by League
t.test(teamData$Runs-teamData$League)

# Create the histogram
hist(teamData$BattingAvg,
     xlab="Batting Average",
     main="Histogram: Batting Average")
```

Commands can be entered one at a time, but usually they are all put into a single file that can be saved and run over and over again.
Getting help

help.start()          general help

help(mean)            help about function mean()

?mean                 same. Help about function mean()

example(mean)         show an example of function mean()

help.search("regression") get help on a specific topic such as regression.

If you’d like to know more about R, check this out: Quick-R (http://www.statmethods.net/index.html)