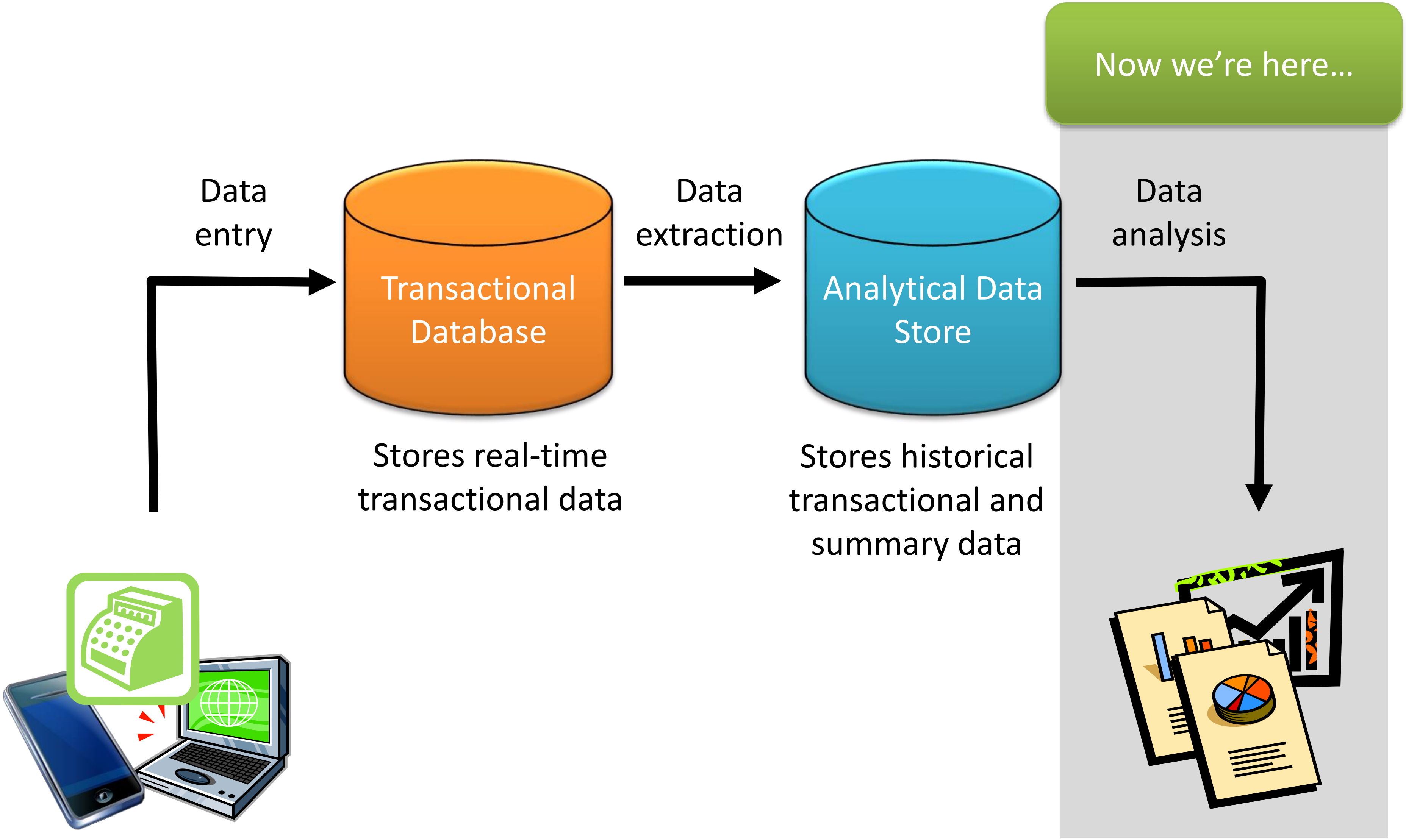
MIS2502:
Data Analytics
Introduction to Advanced Analytics and R

Alvin Zuyin Zheng
zheng@temple.edu
http://community.mis.temple.edu/zuyinzheng/

The Information Architecture of an Organization



Now we’re here…

Data entry

Transactional Database

Stores real-time transactional data

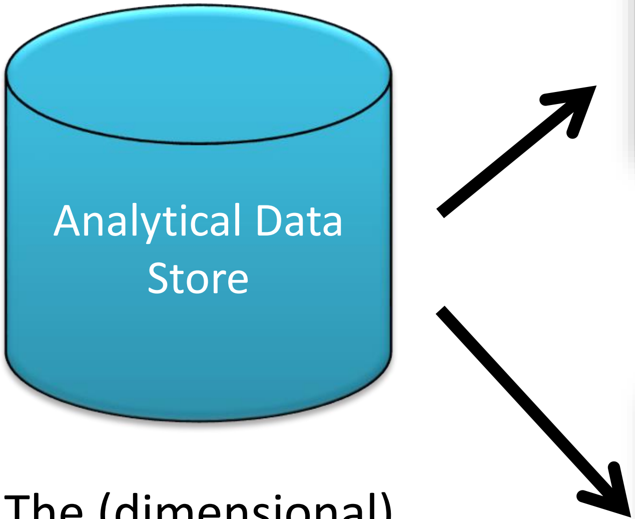
Data extraction

Analytical Data Store

Stores historical transactional and summary data

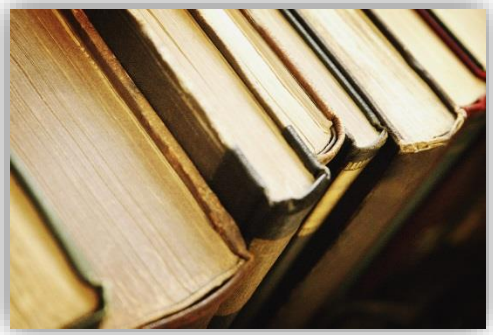
Data analysis

The difference between OLAP and data mining



Analytical Data Store

The (dimensional) data warehouse feed both…



…like a pivot table

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

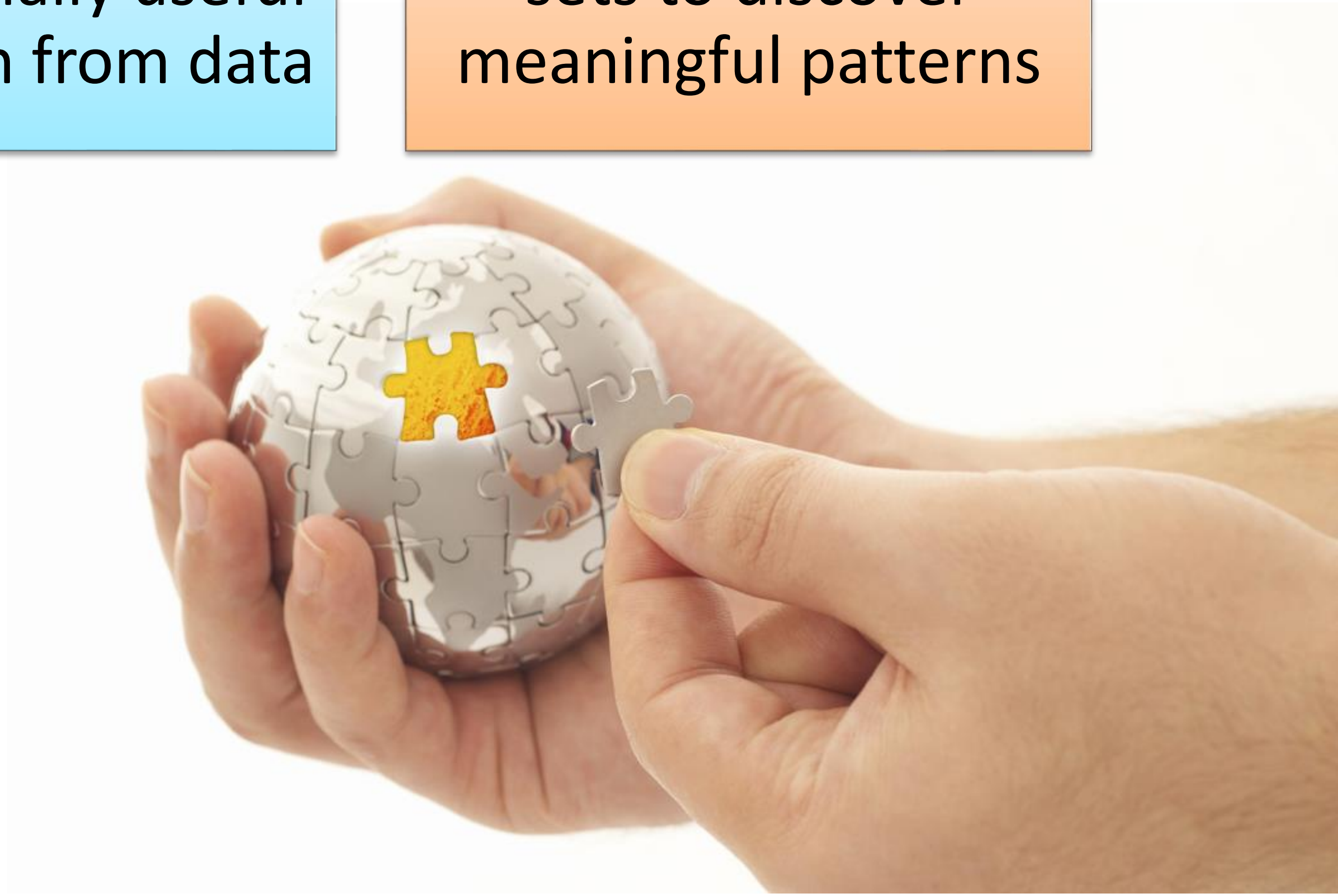
…like what we'll do with R

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

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

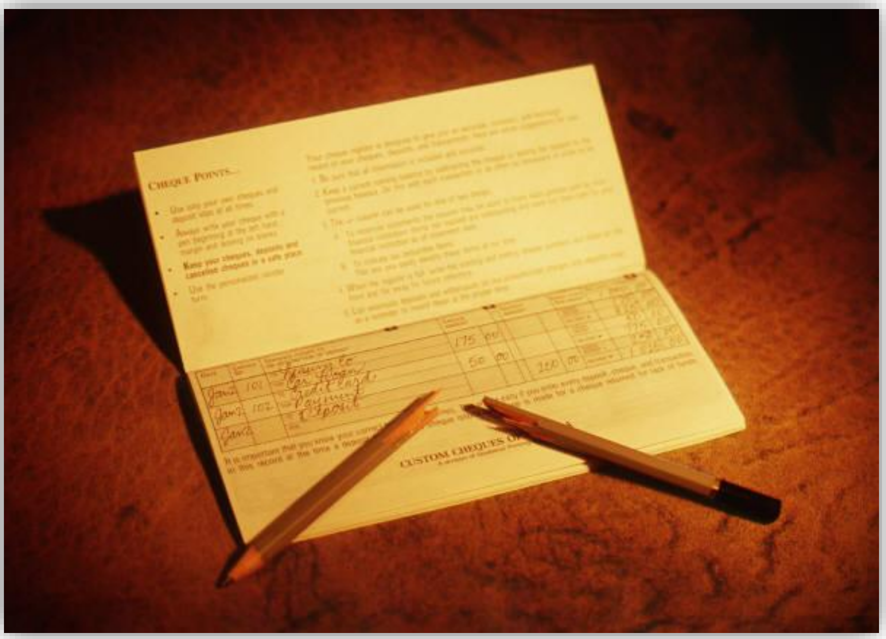
- Which product lines are the highest revenue producers this year?

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

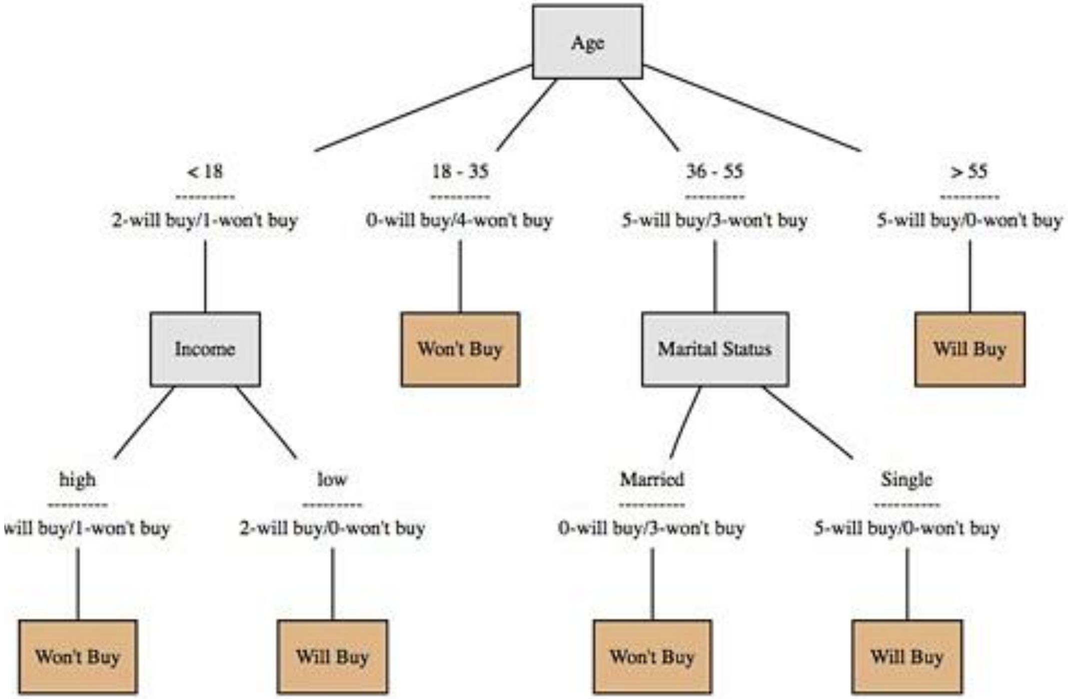
Three Analytics Tasks We Will Be Doing in this Class

- Decision Trees
- 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

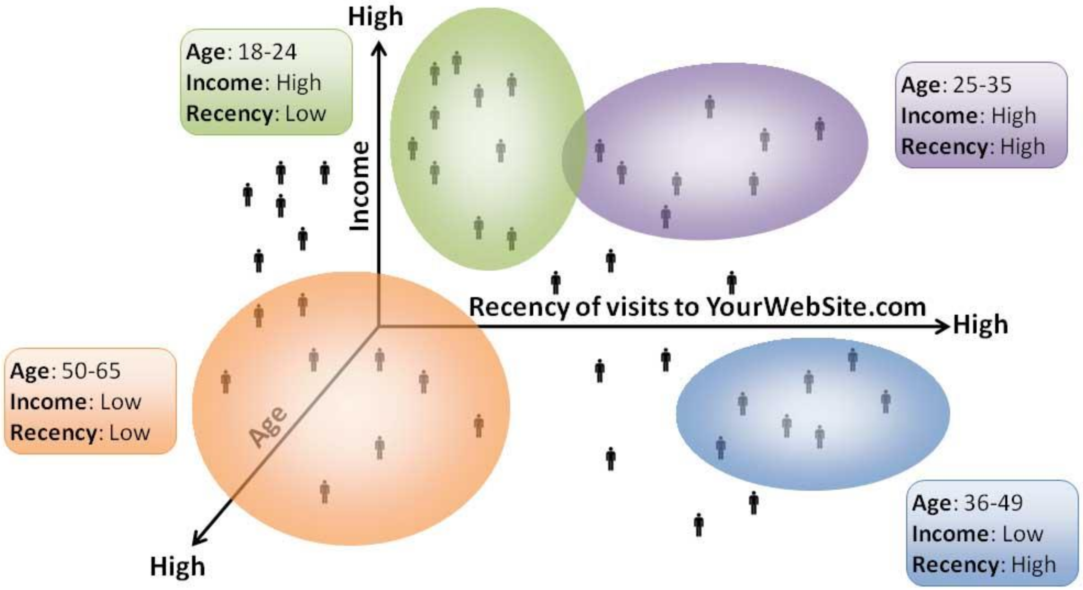
Flag a credit card charge as legitimate

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

http://www.datadrivesmedia.com/two-ways-performance-increases-targeting-precision-and-response-rates/

Association Rule 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

R

- Software development platform and language
- Open source, free
- Many, many, many statistical add-on “packages” that perform data analysis

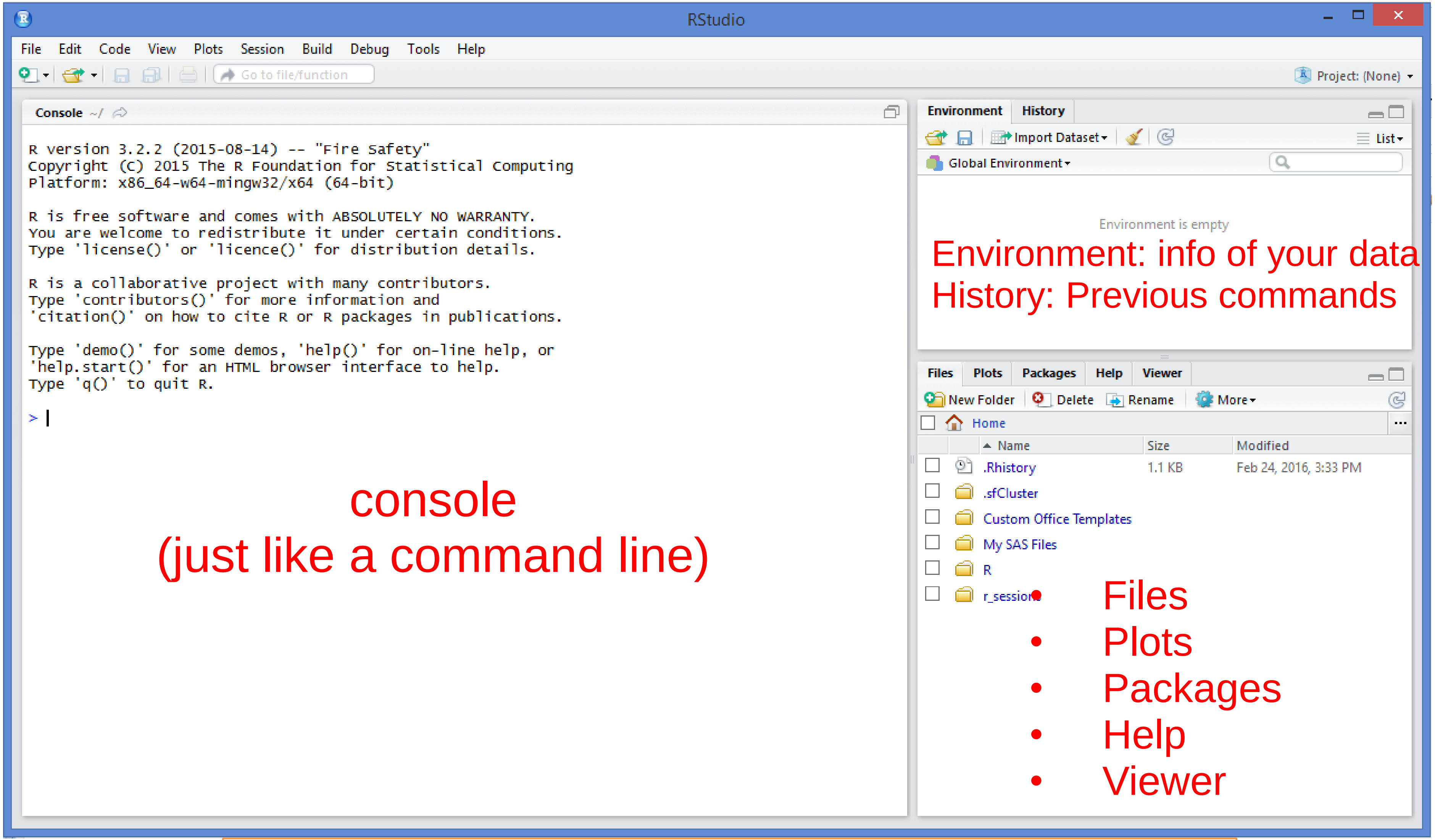
(The base/engine)

RStudio®

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

(The pretty face)

RStudio Interface



```
R version 3.2.2 (2015-08-14) -- "Fire Safety"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

Name	Size	Modified
.Rhistory	1.1 KB	Feb 24, 2016, 3:33 PM
.sfCluster		
Custom Office Templates		
My SAS Files		
R		
r_session		

Environment: info of your data
History: Previous commands

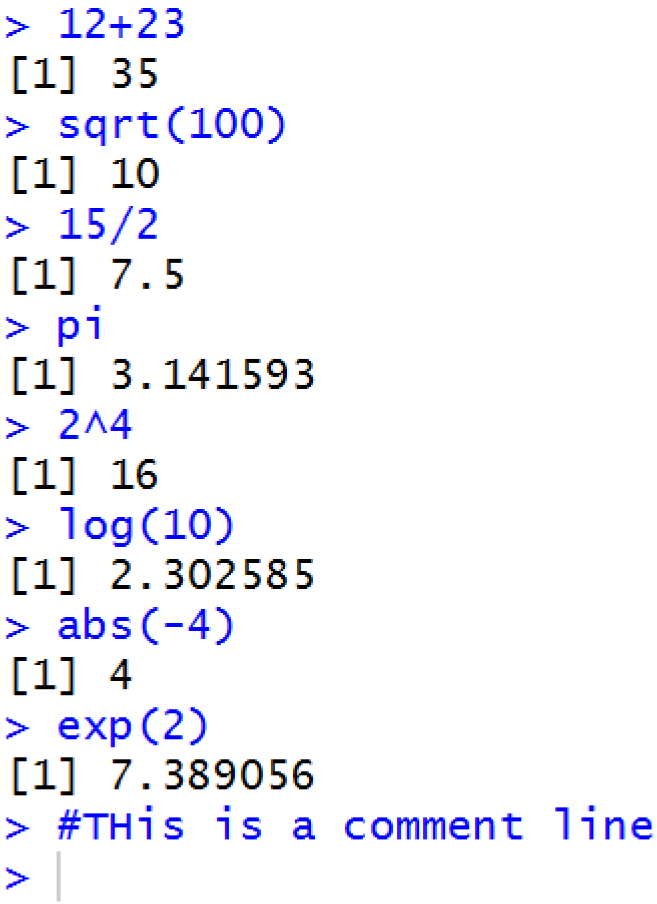
console
(just like a command line)

- Files
- Plots
- Packages
- Help
- Viewer

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

The Basics: Calculations

- R will do math for you:



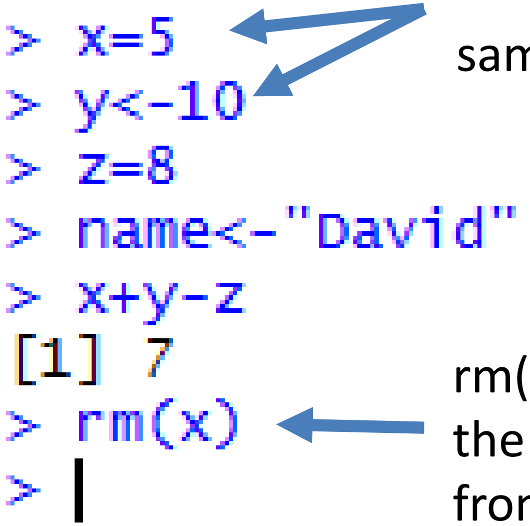
```
> 12+23
[1] 35
> sqrt(100)
[1] 10
> 15/2
[1] 7.5
> pi
[1] 3.141593
> 2^4
[1] 16
> log(10)
[1] 2.302585
> abs(-4)
[1] 4
> exp(2)
[1] 7.389056
> #THis is a comment line
> 
```

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. `Result` is a different variable than `result`)



```
> x=5
> y<-10
> z=8
> name<-"David"
> x+y-z
[1] 7
> rm(x)
> 
```

`<-` and `=` do the same thing

rm() removes the variable from memory

x, y, and z are **variables** that can be manipulated

Basic Data Types

Type	Range	Assign a Value

Numeric	Numbers	`x<-1`
		`y<--2.5`

Character	Text strings	`name<-"Mark"`
		`color<-"red"`

Logical (Boolean)	TRUE or FALSE	`female<-TRUE`

Vectors of values

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

```
> scores<-c(65,75,80,88,82,99,100,100,50)
> scores
[1] 65 75 80 88 82 99 100 100 50
> studentnum<-1:9
> studentnum
[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

```
> 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

```
> 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.
 50.00 75.00   82.00 82.11 99.00 100.00
```

Packages

- Packages (add-ons) 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/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

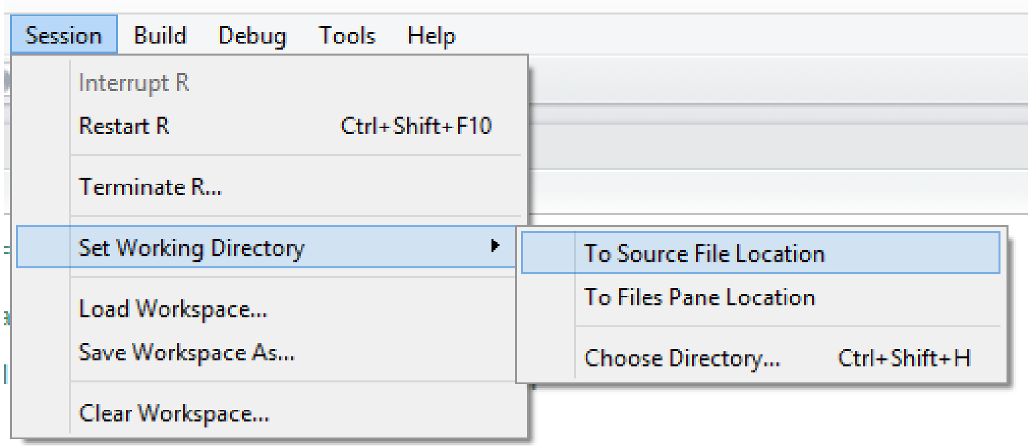
- Keeping everything in a self contained directory helps organize code and analyses

- Check you current working directory with
  `getwd()`

To change working directory

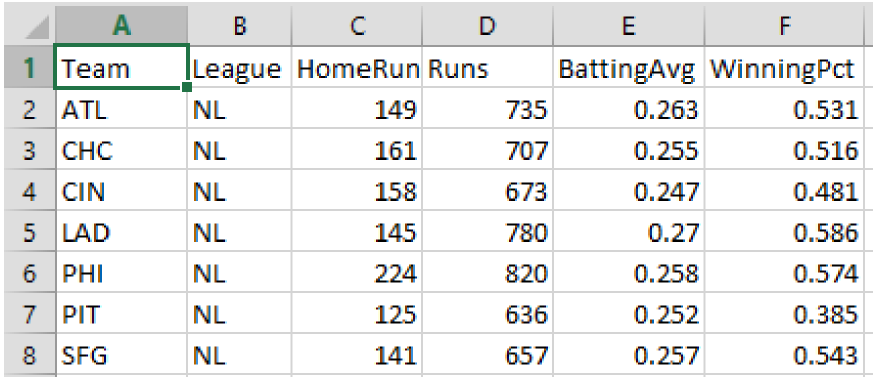
Use the **Session | Set Working Directory** Menu

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



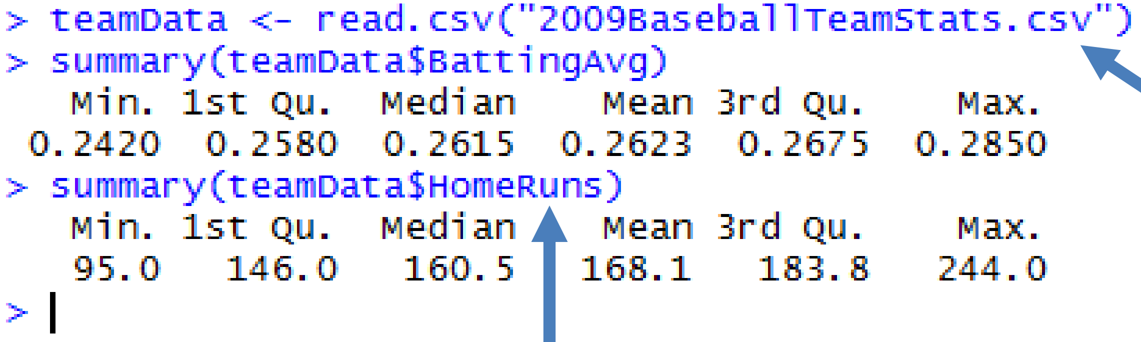
Reading data from a file

- 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)



	A	B	C	D	E	F
1	Team	League	HomeRun	Runs	BattingAvg	WinningPct
2	ATL	NL	149	735	0.263	0.531
3	CHC	NL	161	707	0.255	0.516
4	CIN	NL	158	673	0.247	0.481
5	LAD	NL	145	780	0.27	0.586
6	PHI	NL	224	820	0.258	0.574
7	PIT	NL	125	636	0.252	0.385
8	SFG	NL	141	657	0.257	0.543

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



```
> teamData <- read.csv("2009BaseballTeamStats.csv")
> summary(teamData$BattingAvg)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2420  0.2580  0.2615  0.2623  0.2675  0.2850
> summary(teamData$HomeRuns)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   95.0   146.0   160.5   168.1   183.8   244.0
> |
```

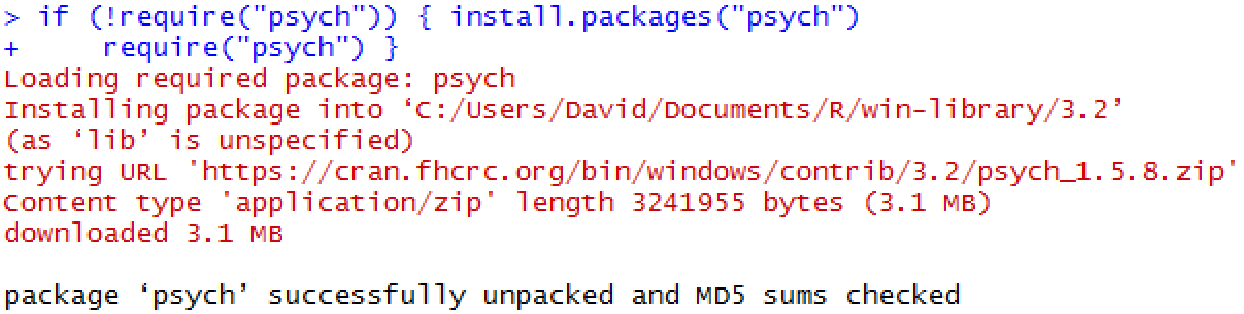
reads data from a CSV file and creates a **data frame** called `teamData` that store the data table.

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

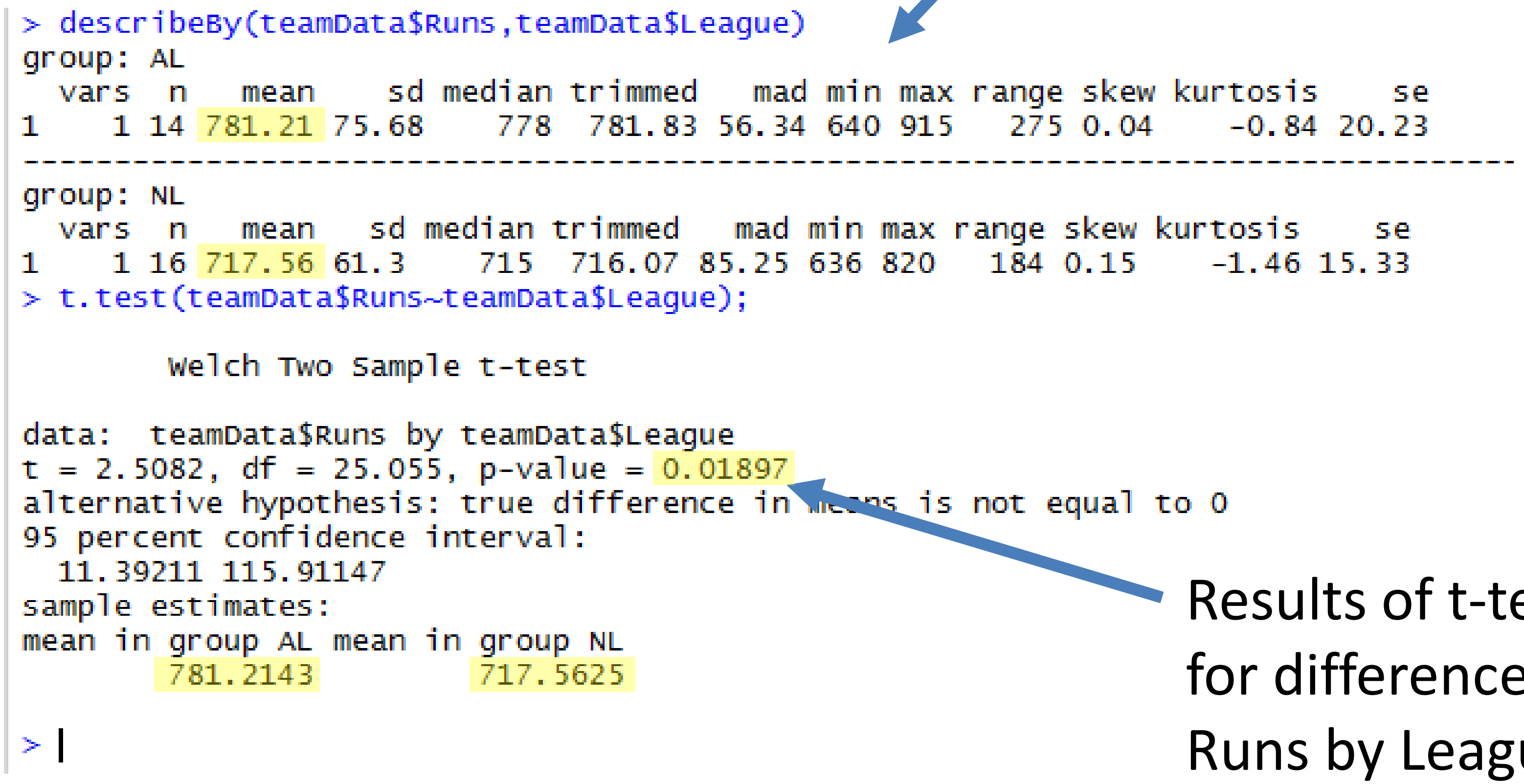
Downloads and installs the package (once per R installation)



```
> if (!require("psych")) { install.packages("psych")
+       require("psych") }
Loading required package: psych
Installing package into ‘C:/Users/David/Documents/R/win-library/3.2’
(as ‘lib’ is unspecified)
trying URL 'https://cran.fhcrc.org/bin/windows/contrib/3.2/psych_1.5.8.zip'
Content type 'application/zip' length 3241955 bytes (3.1 MB)
downloaded 3.1 MB

package ‘psych’ successfully unpacked and MD5 sums checked
```

Looking for differences across groups: The analysis (t-test)

Descriptive statistics, broken up by group (League)



```
> describeBy(teamData$Runs,teamData$League)
group: AL
  vars  n   mean    sd median trimmed   mad min max range skew kurtosis    se
1    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    1 16 717.56 61.3    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

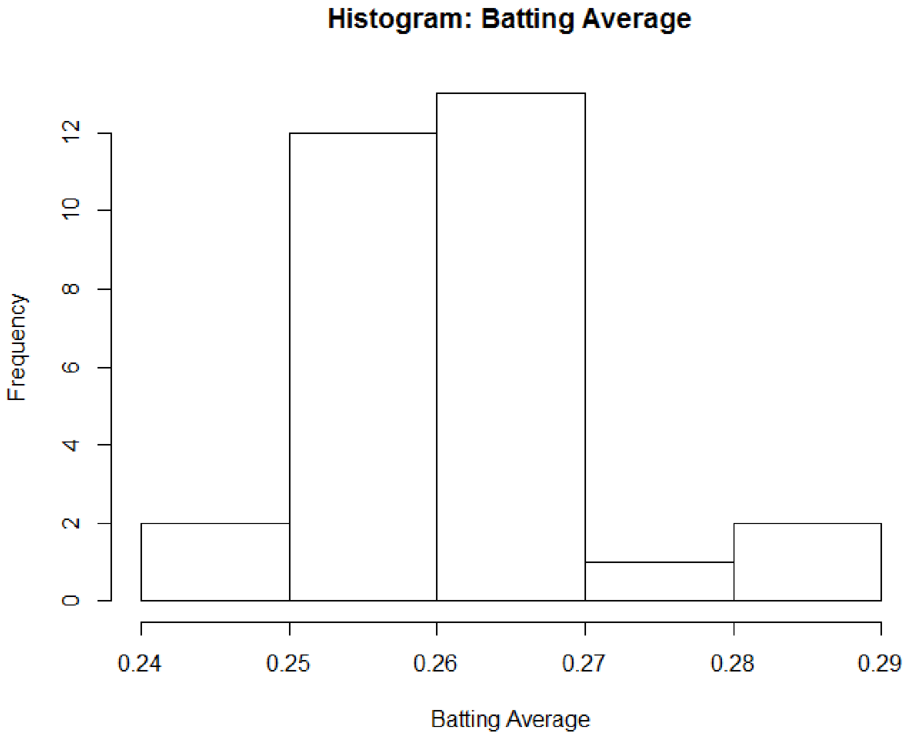
```
hist(teamData$BattingAvg,
        xlab="Batting Average",
        main="Histogram: Batting Average")
```

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()

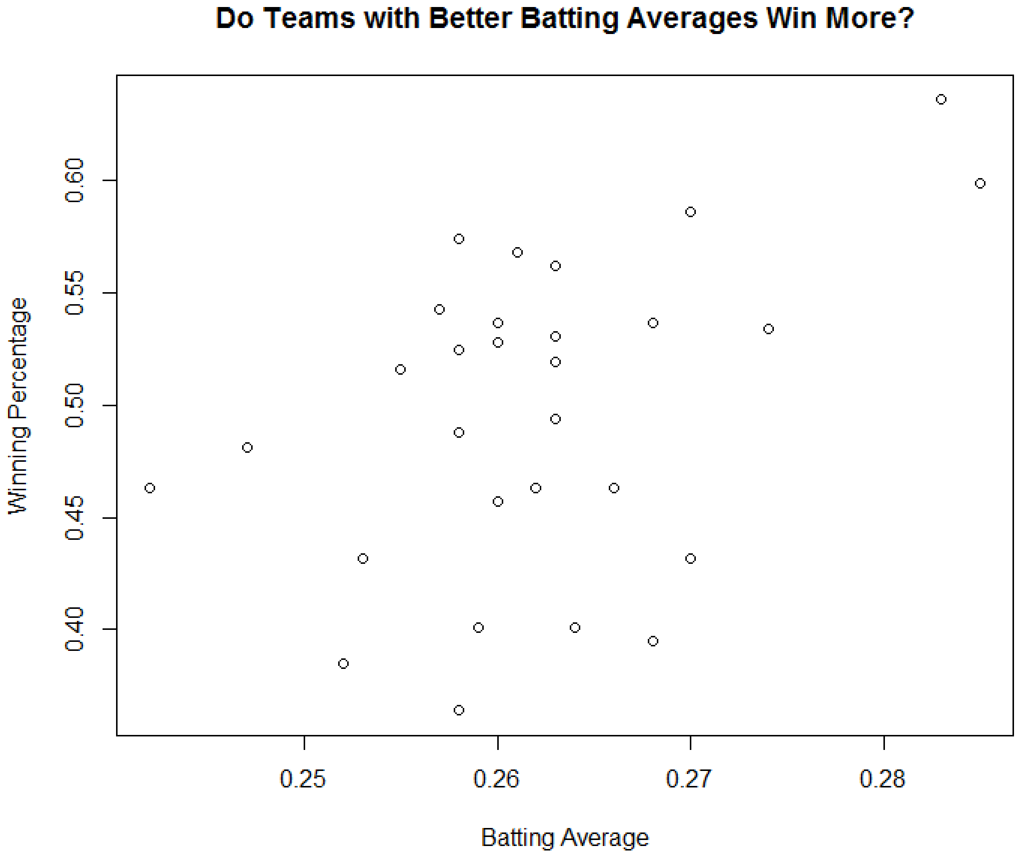
first parameter – x data values

second parameter – y data values

xlab parameter – label for x axis

ylab parameter – label for y axis

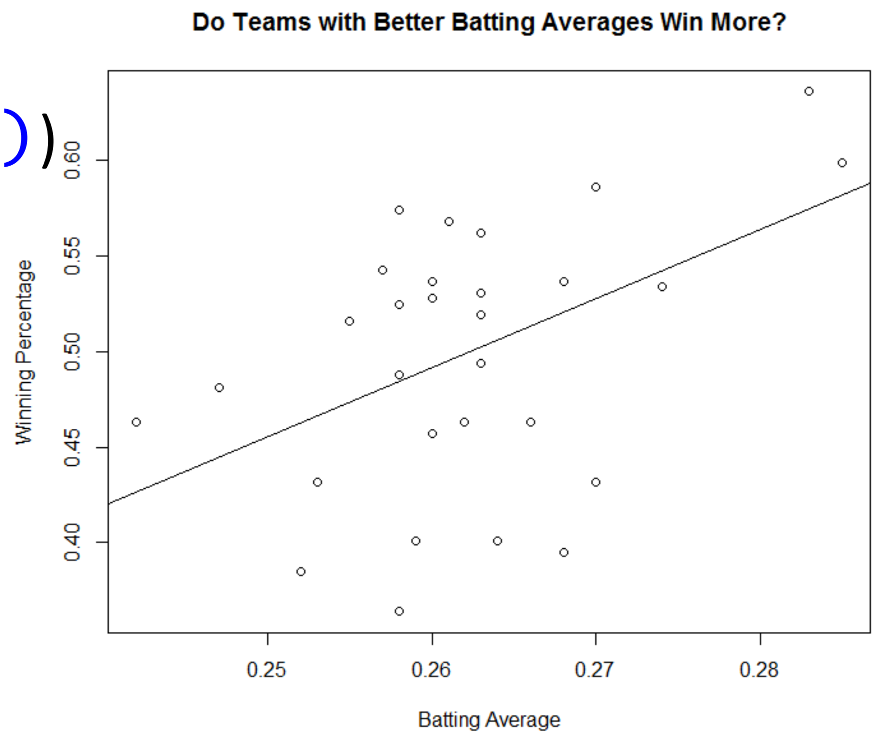
main parameter - sets title for chart



Drawing a regression (trend) line

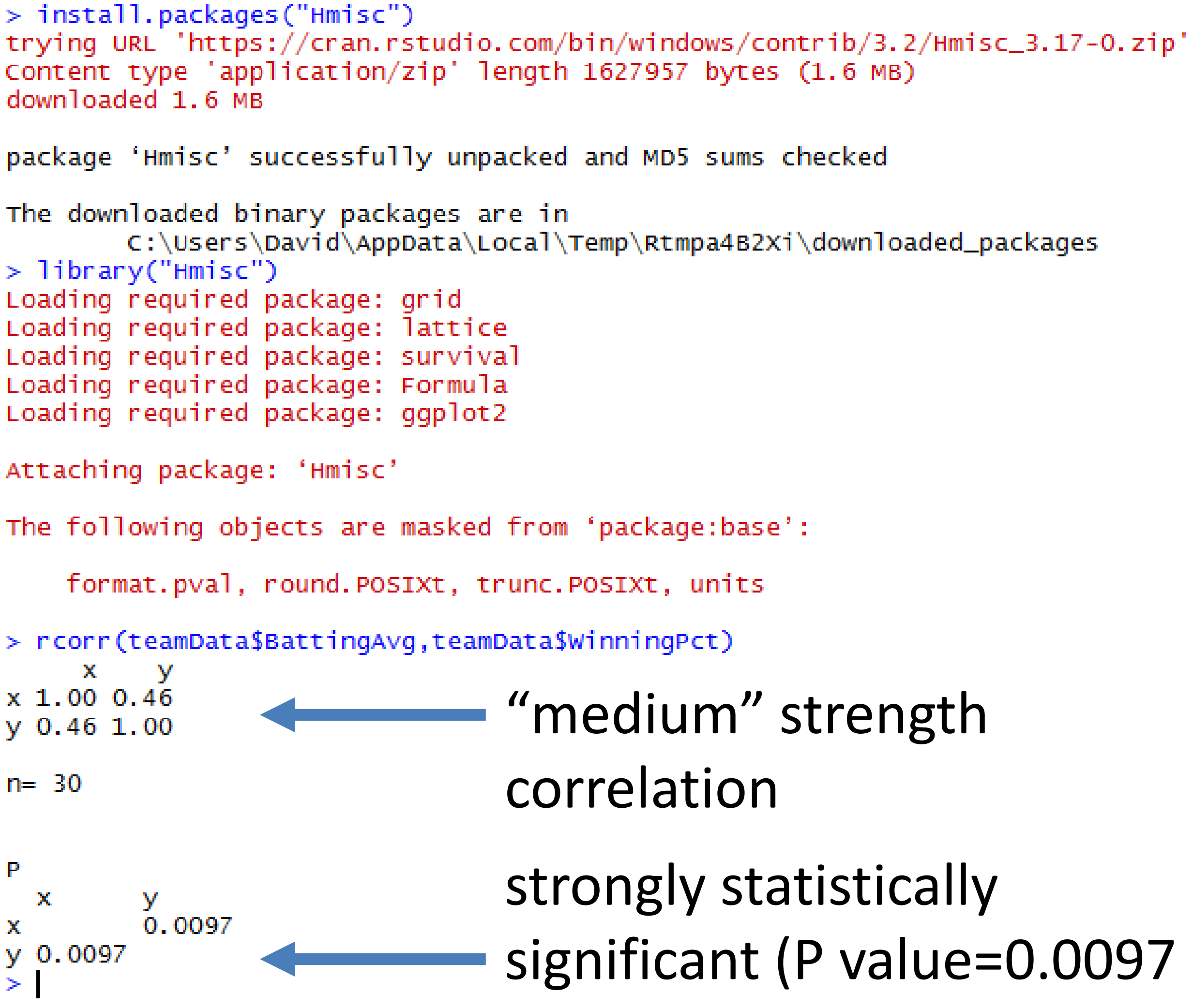
```
plot(teamData$BattingAvg,teamData$WinningPct,
       xlab="Batting Average",
       ylab="Winning Percentage",
       main="Do Teams With Better Batting Averages Win More?")
reg1 <- lm(teamData$WinningPct~teamData$BattingAvg)
abline(reg1)
```

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



But is the correlation statistically significant?

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



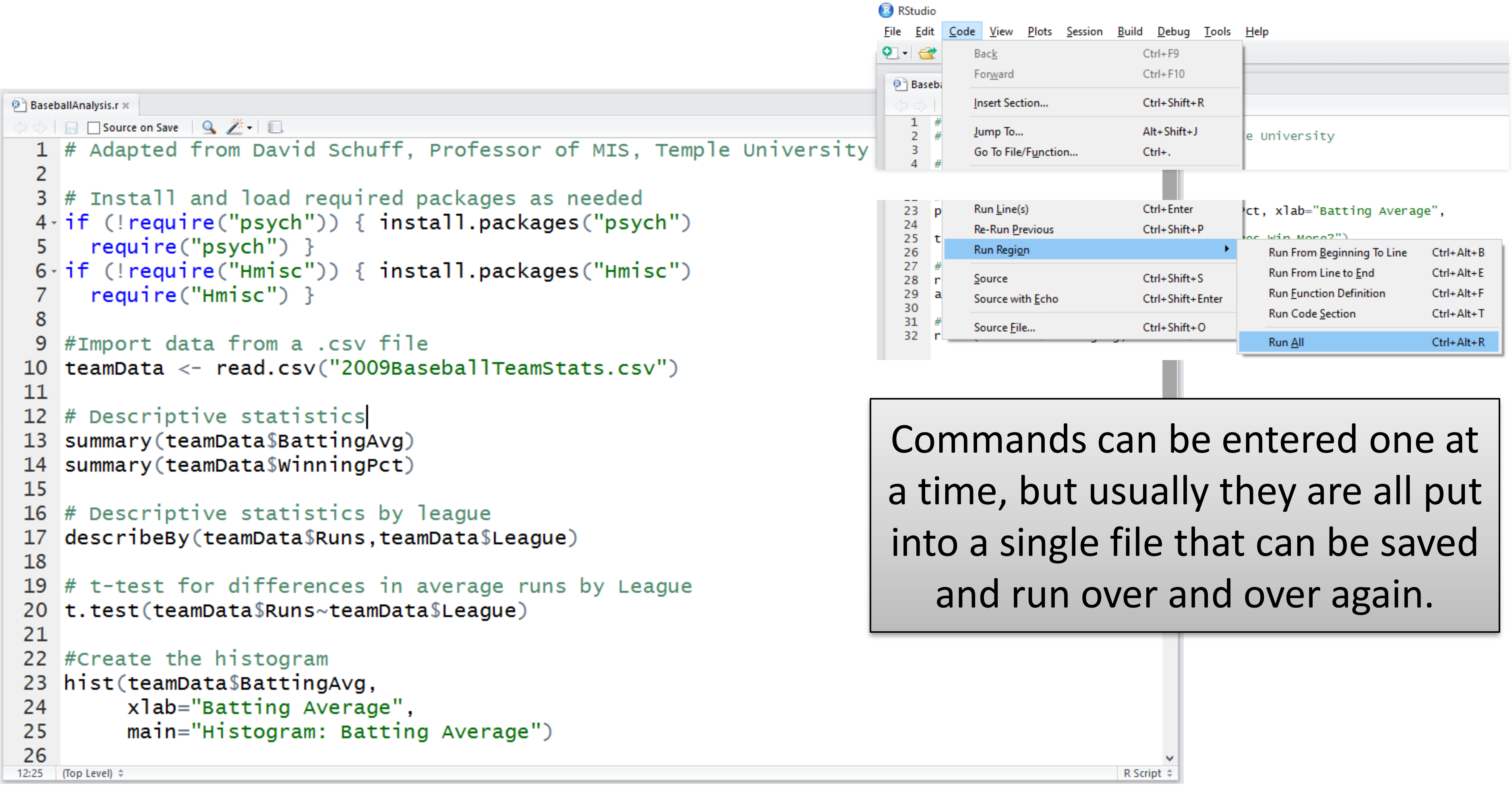
```
> install.packages("Hmisc")
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/Hmisc_3.17-0.zip'
Content type 'application/zip' length 1627957 bytes (1.6 MB)
downloaded 1.6 MB

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 (P value=0.0097 <0.05)

Running this analysis as a script

Use the **Code | Run Region | Run All** Menu



RStudio
File Edit Code View Plots Session Build Debug Tools Help
Back Ctrl+F9
Forward Ctrl+F10
Insert Section... Ctrl+Shift+R
Jump To... Alt+Shift+J
Go To File/Function... Ctrl+.
Run Line(s) Ctrl+Enter
Re-Run Previous Ctrl+Shift+P
Run Region
Source Ctrl+Shift+S
Source with Echo Ctrl+Shift+Enter
Source File... Ctrl+Shift+O
Run From Beginning To Line Ctrl+Alt+B
Run From Line to End Ctrl+Alt+E
Run Function Definition Ctrl+Alt+F
Run Code Section Ctrl+Alt+T
Run All Ctrl+Alt+R
BaseballAnalysis.r
Source on Save
12:25
(Top Level)
R Script

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

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