Utilizing  in IR

Eric S. Atchison

Mississippi Institutions of Higher Learning

MBUG 2015

September 14, 2015

Agenda

- What is R?
- Where to get R?
- Getting Started
- Loading Data
- Describing Data
- Plotting Data
- T-Tests
- ANOVA
- Regression Modeling
- Test for Proportion Differ.
- Merging data files
- Resources

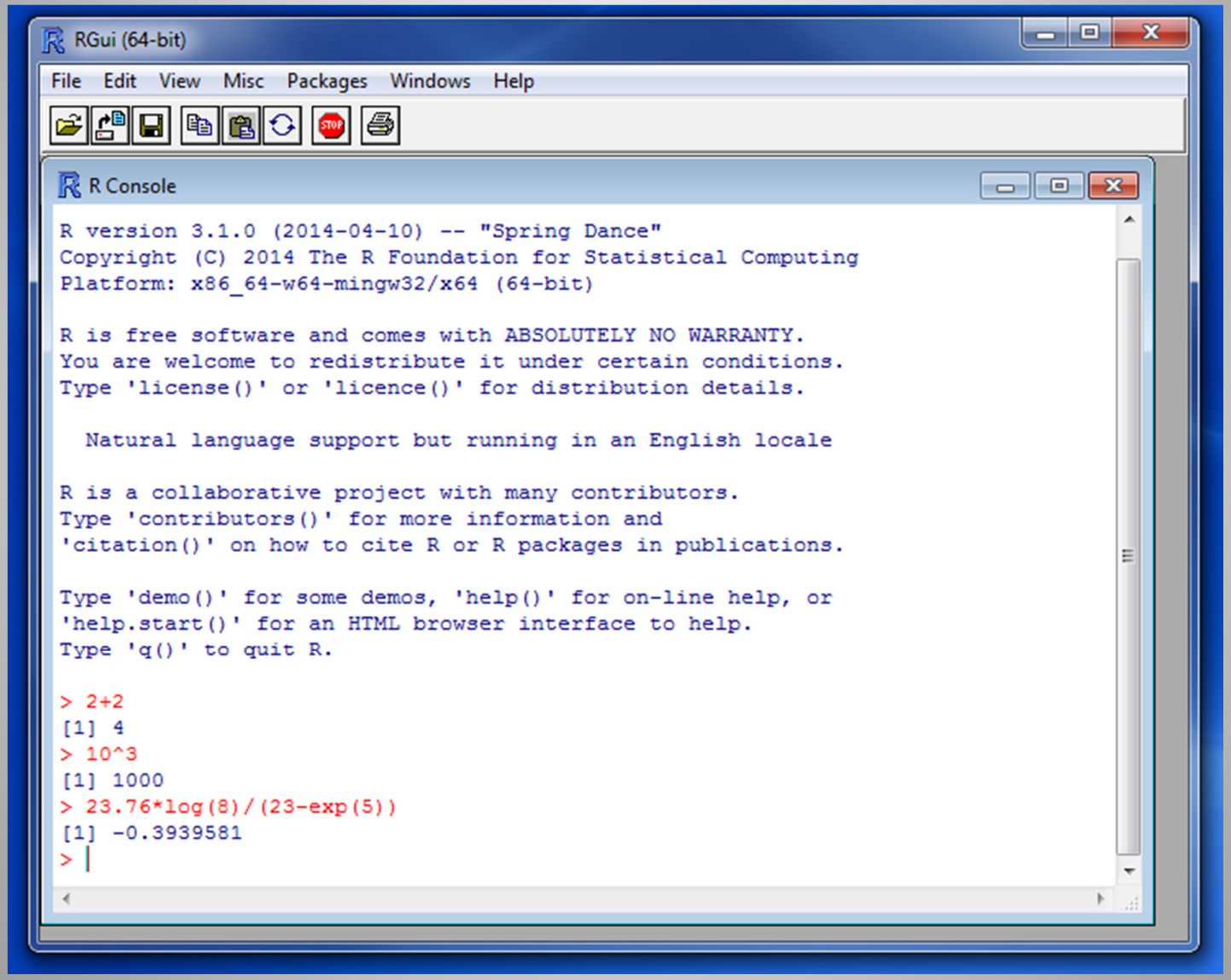
What is R?

- R is a FREE language and environment for statistical computing and graphics
- Available for multiple platforms (i.e. Windows, Mac, Linux)
- More than 2,000 packages available to broaden your capabilities

Where to get R?

- www.r-project.org
- Also consider downloading:
  - R Studio (http://www.rstudio.com)
  - Includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.
  - R Commander: (http://www.rcommander.com)
  - Enables access a selection of commonly-used R commands using a simple interface that should be familiar to most computer users.

Getting Started: Calculator



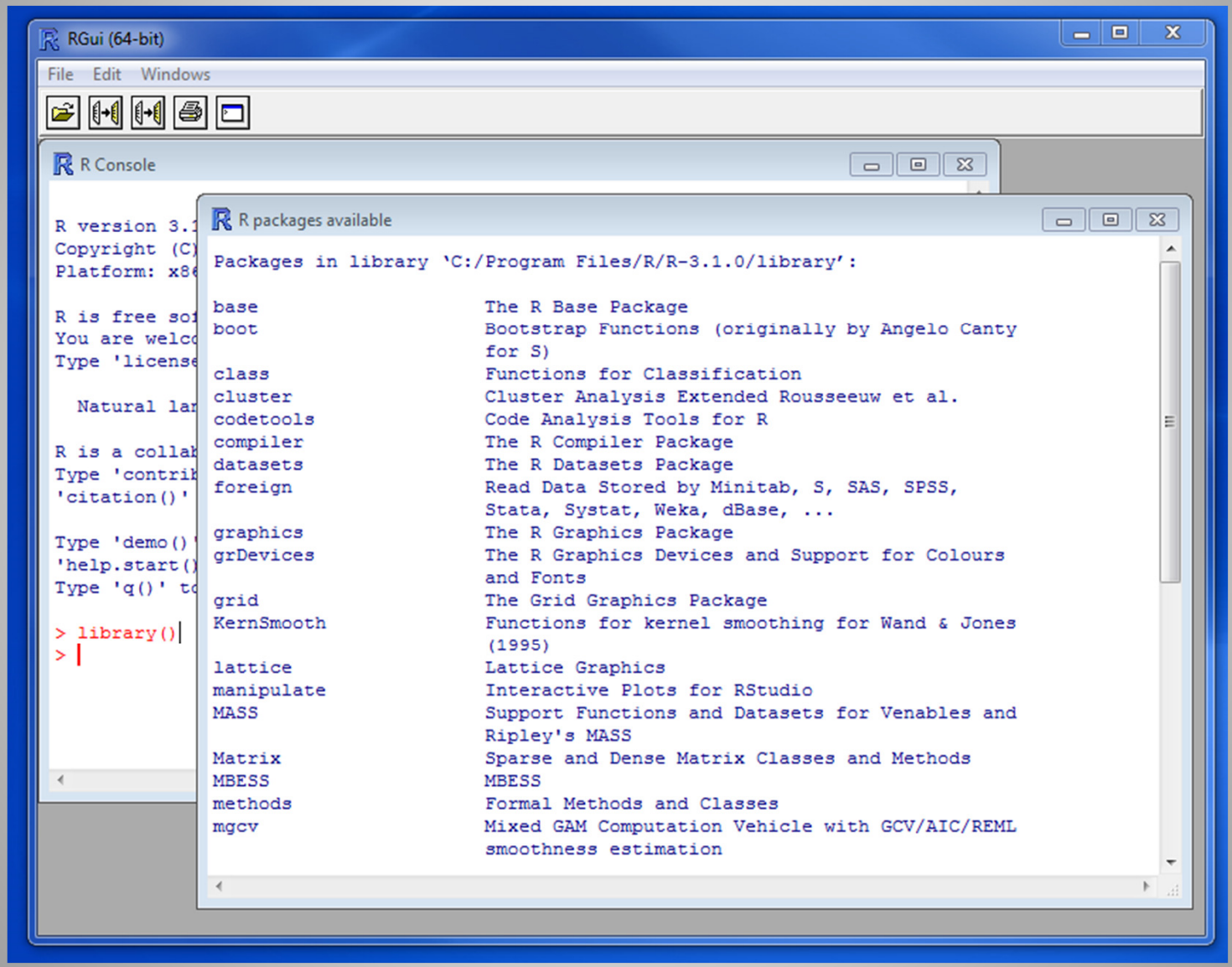
Getting Started: Useful Packages for IR

Package	Description
foreign	Contains functions to read SPSS files
gdata	Contains functions to read Excel spreadsheets
ggplot2	Package for creating nice looking graphics http://had.co.nz/ggplot2
psych	Package contains lots of useful functions for descriptive statistics
rcmdr	R Commander is a graphical interface for R
RMySQL	Package for interfacing with MySQL databases
RODBC	Package contains functions to read and write data from ODBC databases (e.g. Oracle, MS SQLServer)
RSQLite	Package for the creation and editing of SQLite databases embedded within R
stats	Package contains functions for statistical calculations and random number generation

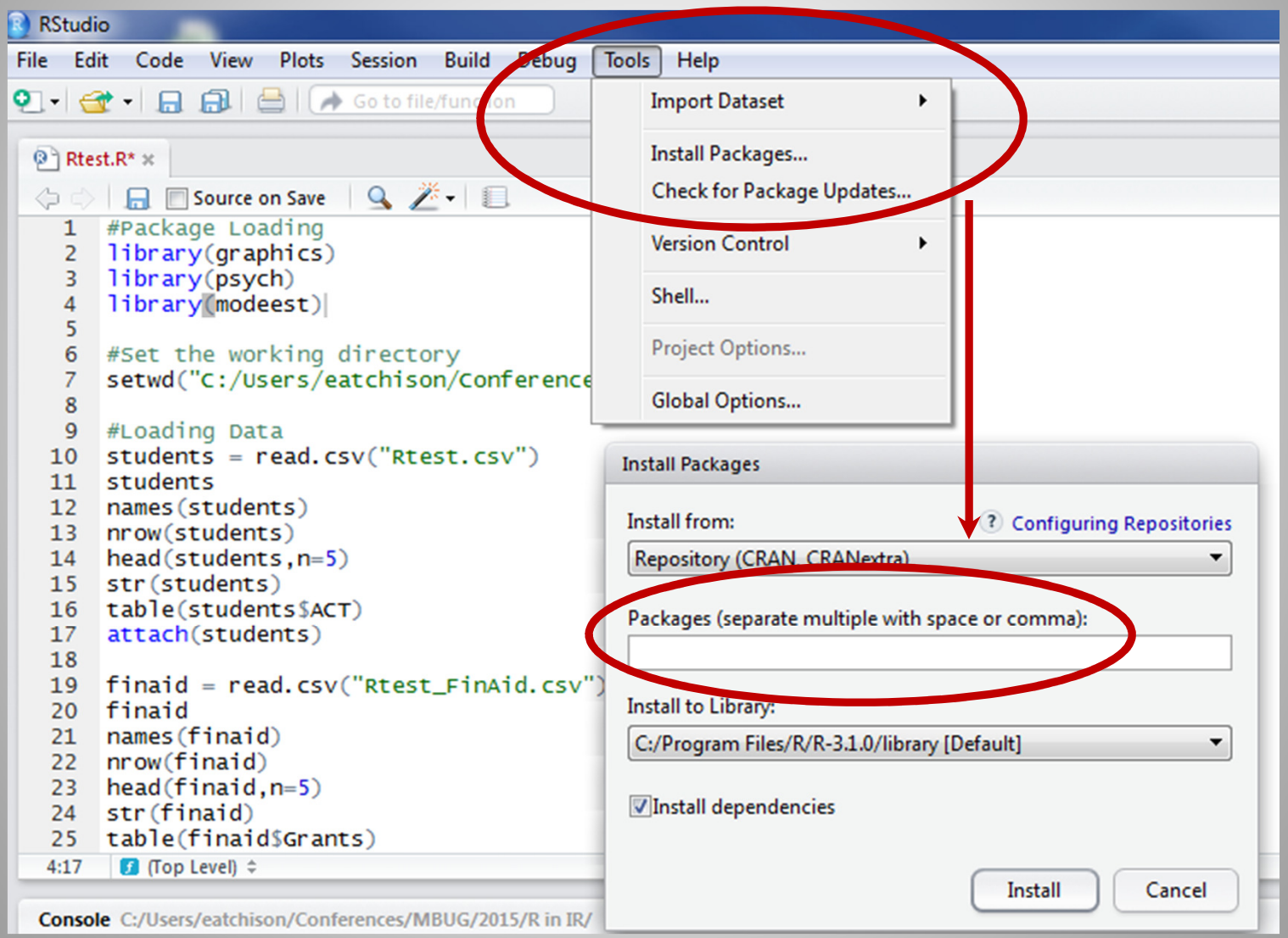
Getting Started: Basic Commands

- search() : Returns all packages that are currently attached to the system
- library(package_name) : Loads the requested package
- ls(package_name) : Returns a list of functions in a particular package
- ?function_name : loads the help file associated with a function

Getting Started: Loading Packages

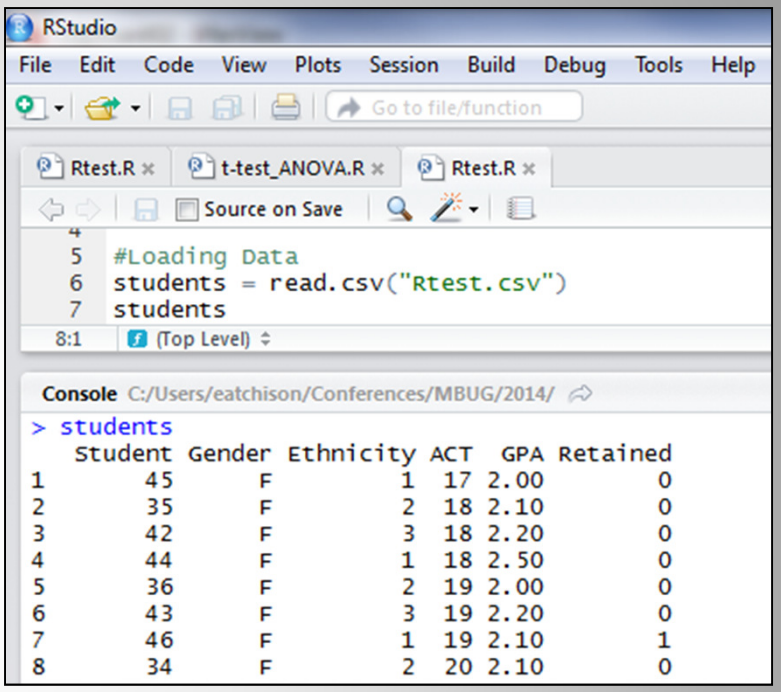


Getting Started: Loading Packages



Loading Data: CSV Files

```
> students = read.csv("Rtest.csv")
> students
```




```
> names(students)
> nrow(students)
```

```
> names(students)
[1] "Student"   "Gender"    "Ethnicity" "ACT"       "GPA"       "Retained"
> nrow(students)
[1] 50
```

```
> str(students)
```

```
> str(students)
'data.frame': 50 obs. of  4 variables:
 $ Student: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Gender : Factor w/ 2 levels "F","M": 2 2
 $ ACT    : int  16 16 17 18 19 19 21 23 24
 $ GPA    : num  1.8 1.9 2 2.2 2.5 2.8 3.2
```

Loading Data: Sampling

> sample.int(50,size=10)

> sample.int(30,size=10)

> sample.int(20,size=10)

```
> #Sampling from Data File
> sample.int(50,size=10)
 [1] 33 16 46 36 21 19 14 38 49 22
> sample.int(30,size=10)
 [1]  7 11 13 21 14  1 22 25 24 23
> sample.int(20,size=10)
 [1] 19  4  9 15  3 20 14  2 10  8
```

Describing Data: Frequency Tables

One-Way Frequency Table:

> table(students$gender)

> table(students$ACT)

```
> table(students$Gender)

 F  M
25 25
> table(students$ACT)

16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 32
 3  3  7  7  6  2  2  2  3  4  1  2  4  2  1  1
```

Two-way Frequency Table:

> table1 = table(students$gender,students$ACT)

```
> table(students$Gender,students$ACT)

   16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 32
 F  0  1  3  3  4  0  1  1  1  3  1  1  3  1  1  1
 M  3  2  4  4  2  2  1  1  2  1  0  1  1  1  0  0
```

Describing Data: Proportions

Cell percentages: prop.table(table1)

Row percentages: prop.table(table1, 1)

Column percentages: prop.table(table1, 2)

```
> prop.table(table1)

     16   17   18   19   20   21   22   23   24   25   26   27   28   30   31   32
F 0.00 0.02 0.06 0.06 0.08 0.00 0.02 0.02 0.02 0.06 0.02 0.02 0.06 0.02 0.02 0.02
M 0.06 0.04 0.08 0.08 0.04 0.04 0.02 0.02 0.04 0.02 0.00 0.02 0.02 0.02 0.00 0.00
> prop.table(table1,1)

     16   17   18   19   20   21   22   23   24   25   26   27   28   30   31   32
F 0.00 0.04 0.12 0.12 0.16 0.00 0.04 0.04 0.04 0.12 0.04 0.04 0.12 0.04 0.04 0.04
M 0.12 0.08 0.16 0.16 0.08 0.08 0.04 0.04 0.08 0.04 0.00 0.04 0.04 0.04 0.00 0.00
> prop.table(table1,2)

          16        17        18        19        20        21        22        23
F 0.0000000 0.3333333 0.4285714 0.4285714 0.6666667 0.0000000 0.5000000 0.5000000
M 1.0000000 0.6666667 0.5714286 0.5714286 0.3333333 1.0000000 0.5000000 0.5000000
```

Describing Data: Mean & SD

> mean(students$ACT, na.rm=TRUE)

> sd(students$ACT, na.rm=TRUE)

```
> mean(students$ACT, na.rm=TRUE)
[1] 21.94
> sd(students$ACT, na.rm=TRUE)
[1] 4.409683
```

> summary(students$ACT)

```
> summary(students$ACT)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  16.00   18.25   20.00   21.94   25.00   32.00
```

Describing Data: Descriptives

- Using the "psych" package provides the *describe* and *describeBy* functions

> describe(students$ACT)

```
> describe(students$ACT)
  vars  n  mean   sd median trimmed  mad min max range skew kurtosis   se
1    1 50 21.94 4.41     20    21.6 4.45  16  32    16 0.59    -0.85 0.62
```

> describeBy(students$ACT, students$Gender)

```
> describeBy(students$ACT, students$Gender)
group: F
  vars  n  mean   sd median trimmed  mad min max range skew kurtosis   se
1    1 25 23.28 4.57     23   23.05 5.93  17  32    15 0.32     -1.3 0.91
------------------------------------------------------------------------
group: M
  vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
1    1 25 20.6 3.88     19   20.24 2.97  16  30    14 0.82    -0.35 0.78
```

Describing Data: Mode

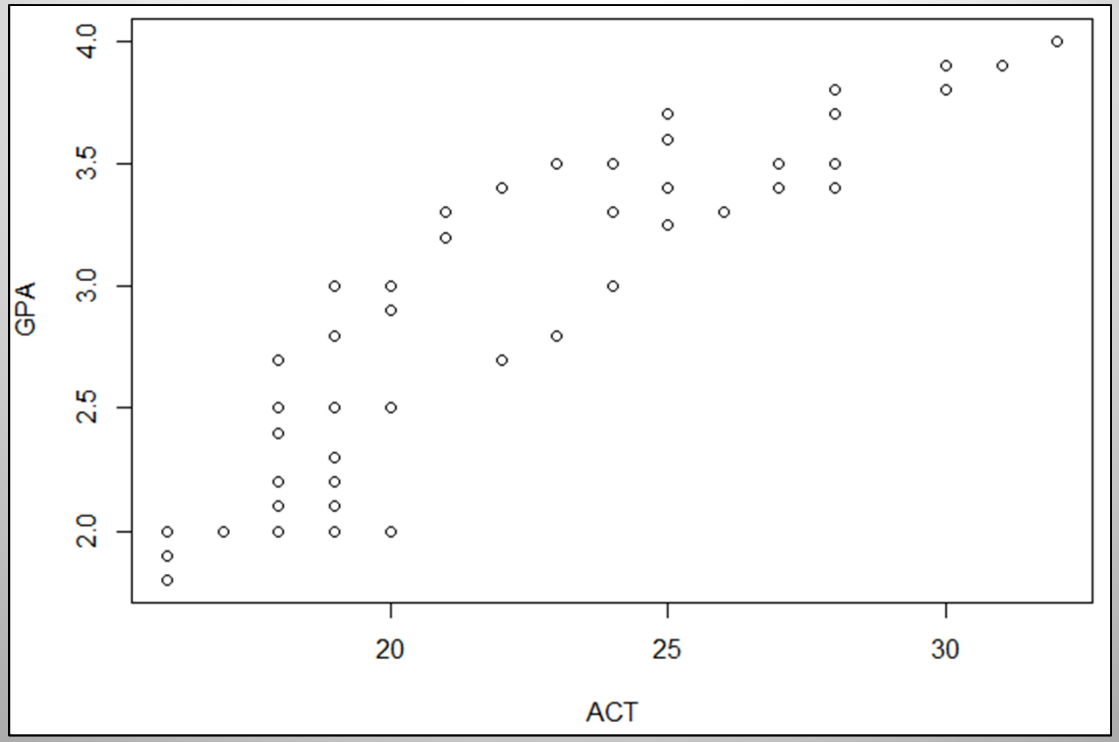
- mode() function returns information about data type instead of the statistical mode;
- Add-on package modeest that adds a mfv() function (most frequent value) to find the statistical mode

- mfv(ACT)
- mfv(GPA)

```
> mfv(ACT)
[1] 19
> mfv(GPA)
[1] 2
```

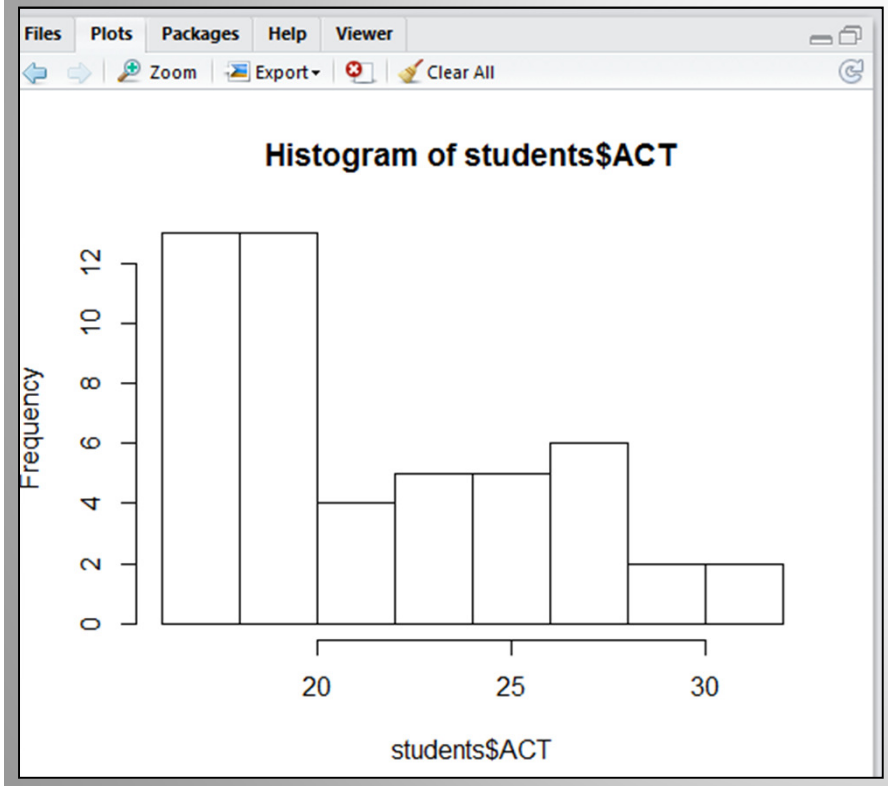
Describing Data: Correlations

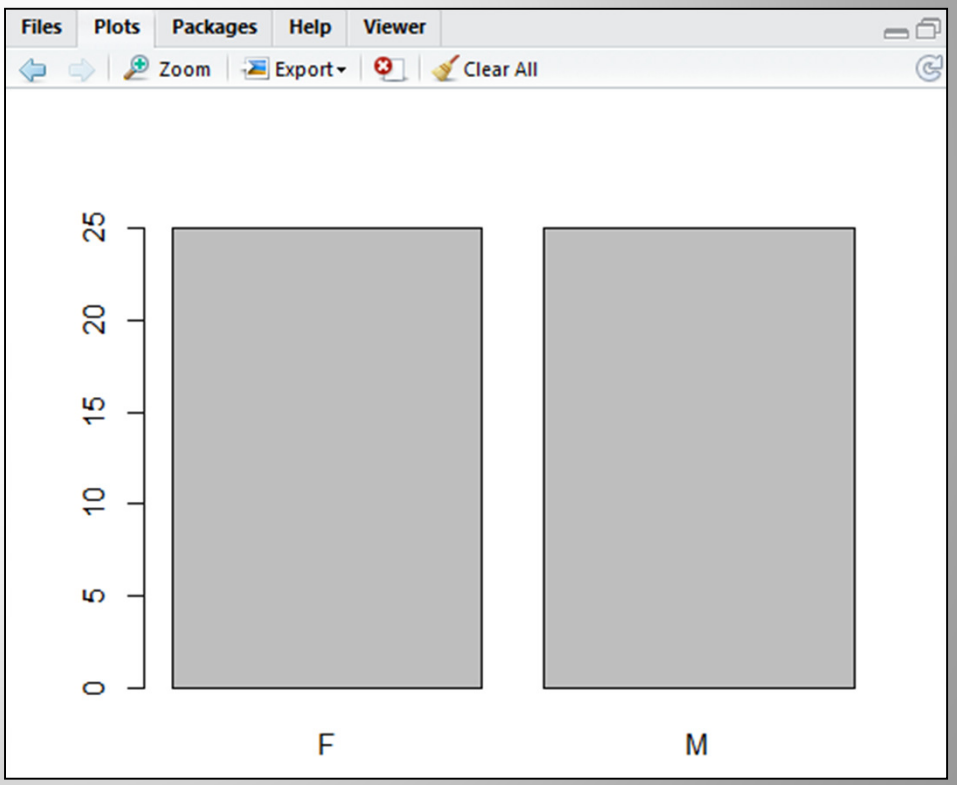
- cor(students$ACT, students$GPA, use="complete")
  - 0.8985408
- plot(students$ACT,students$GPA, xlab="ACT", ylab="GPA")



Plotting Data: Bar Charts

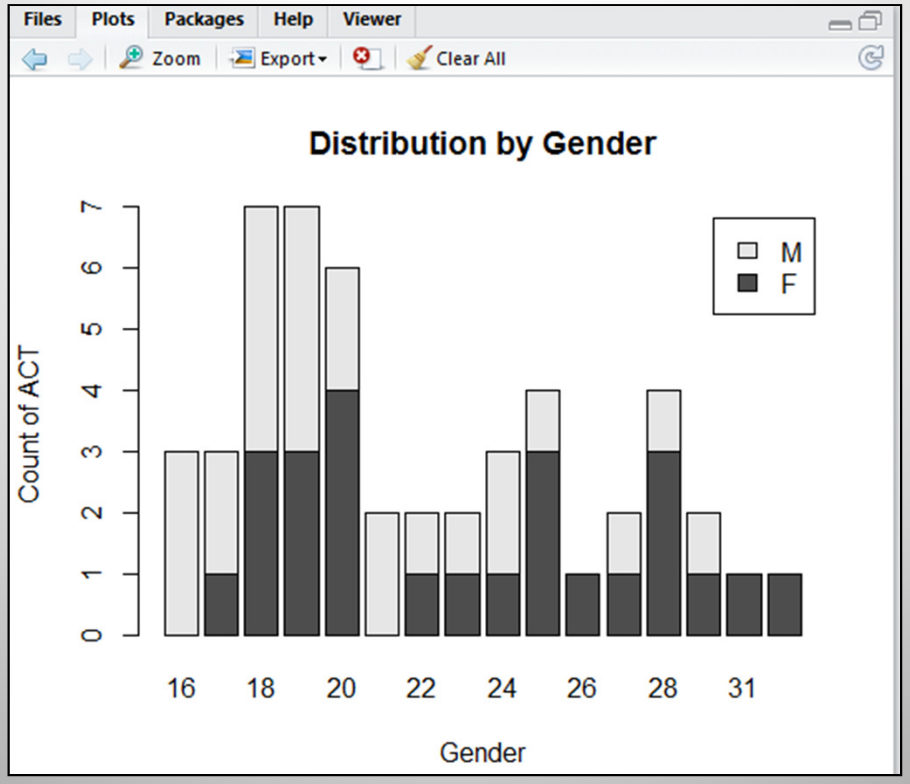
```
> hist(students$ACT)
> barplot(table(students$Gender))
```





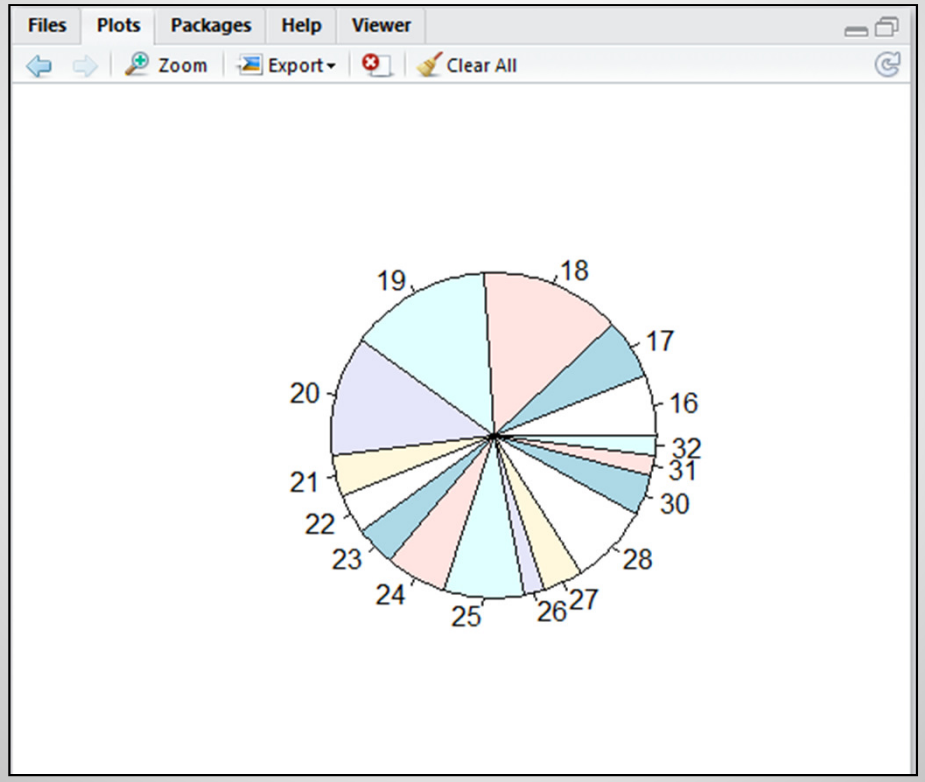
Plotting Data: Stacked Bar Charts

```
> barplot(table1, main='Distribution by Gender',
        xlab='Gender', ylab='Count of ACT',
        legend=rownames(table1))
```



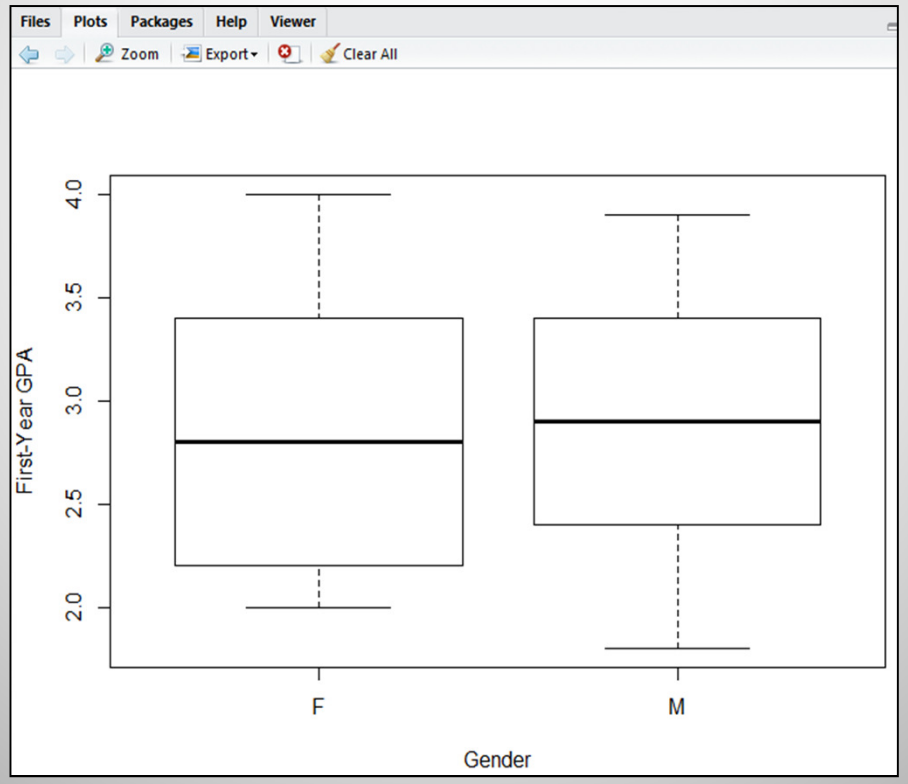
Plotting Data: Pie Charts

```
> pie(table(students$ACT))
```



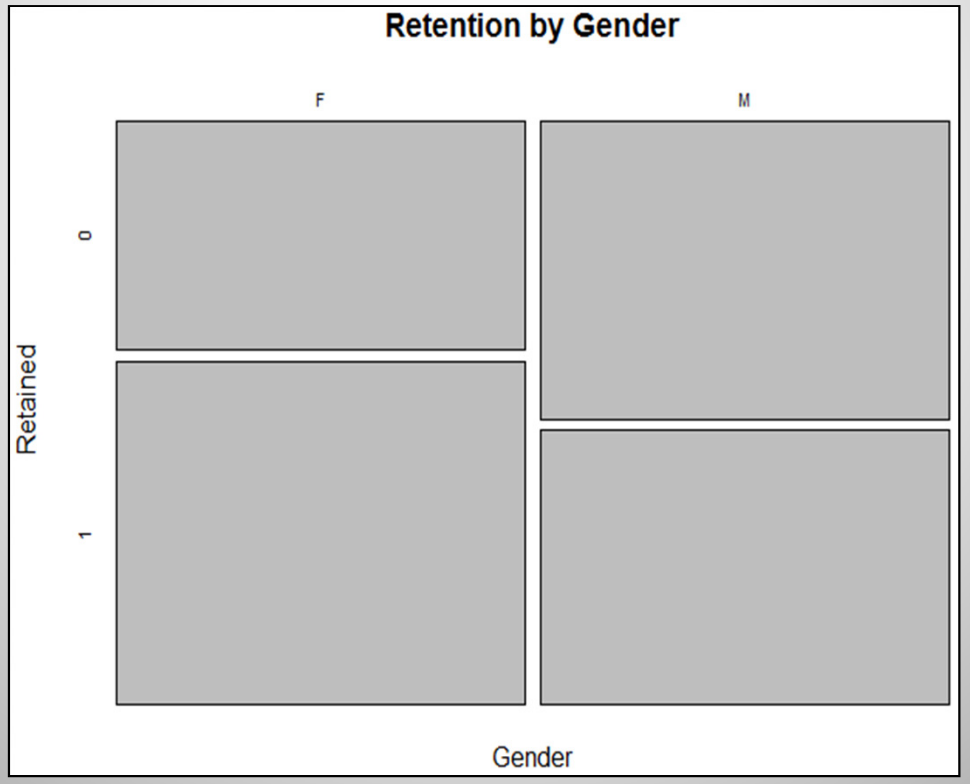
Plotting Data: Box Plots

```
> boxplot(GPA~Gender,data=students, xlab="Gender",
        ylab="First-Year GPA")
```



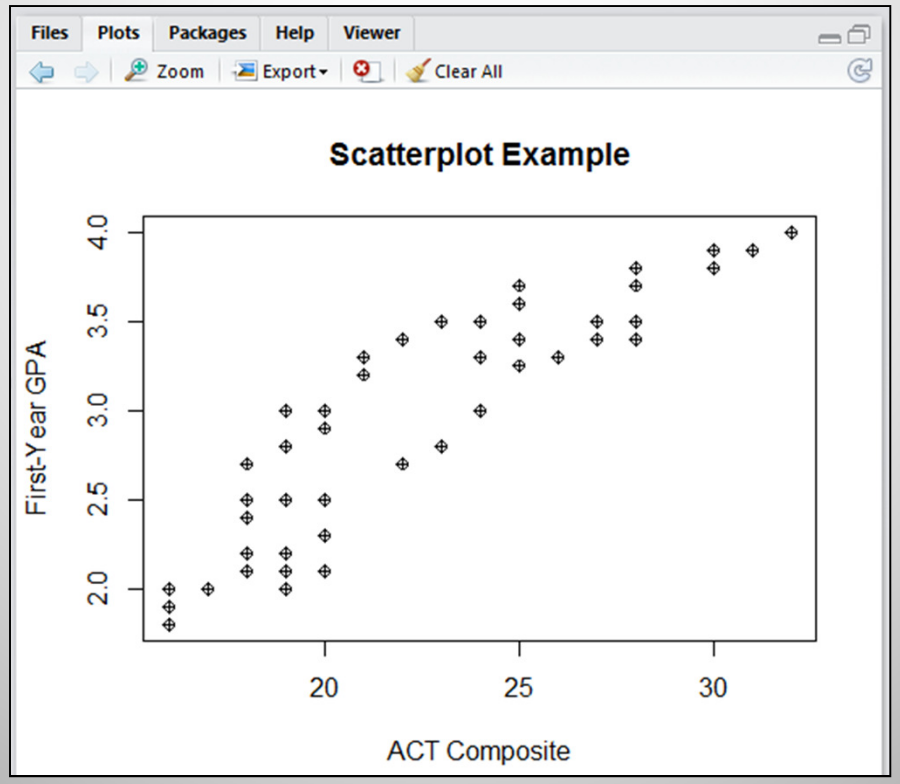
Plotting Data: Mosaic Plots

```
> mosaicplot(~Gender + Retained, data=students,
      main="Retention by Gender")
```



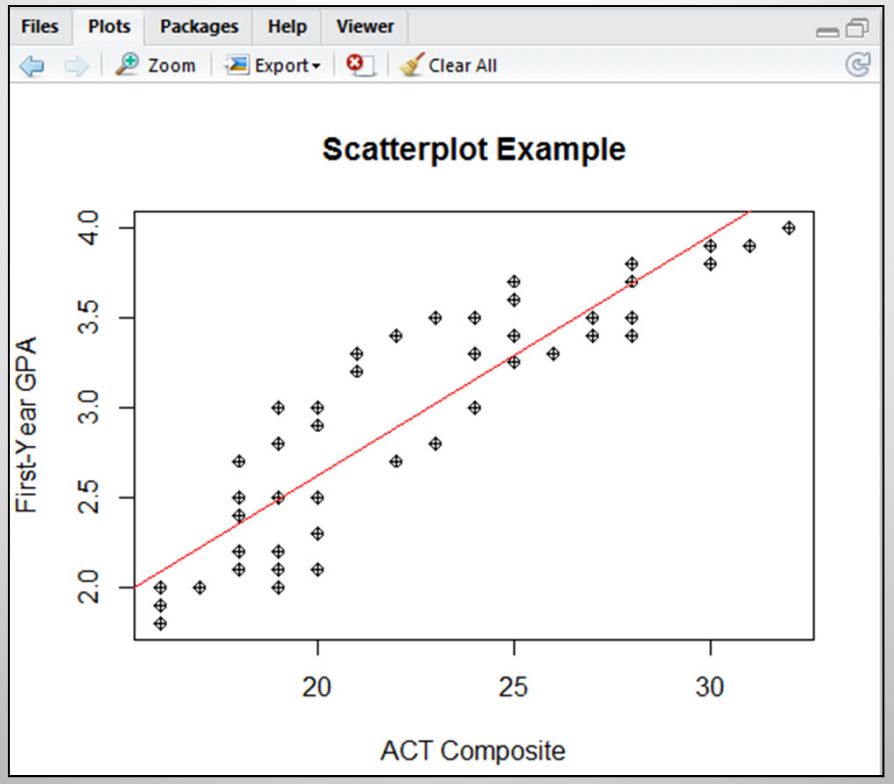
Plotting Data: Scatter Plots

```
> plot(ACT, GPA, main="Scatterplot Example", xlab="ACT Composite", ylab="First-Year GPA", pch=10)
```



Plotting Data: Adding Fit Lines

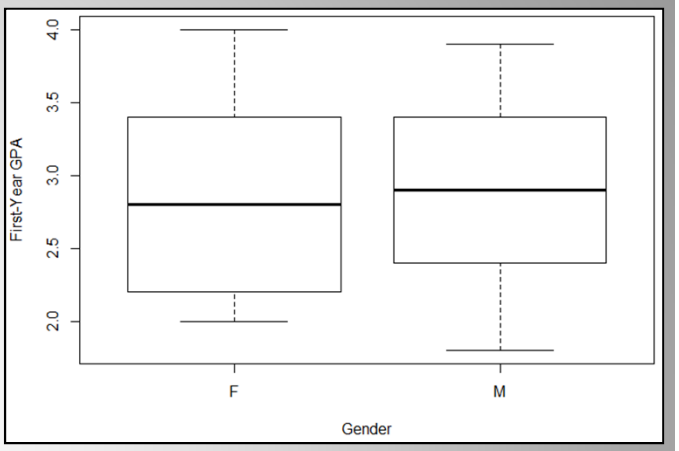
```
> plot(ACT, GPA, main="Scatterplot Example", xlab="ACT
      Composite", ylab="First-Year GPA", pch=10)
> abline(lm(ACT~GPA), col="red")
```



T-Tests: Independent & Paired Samples

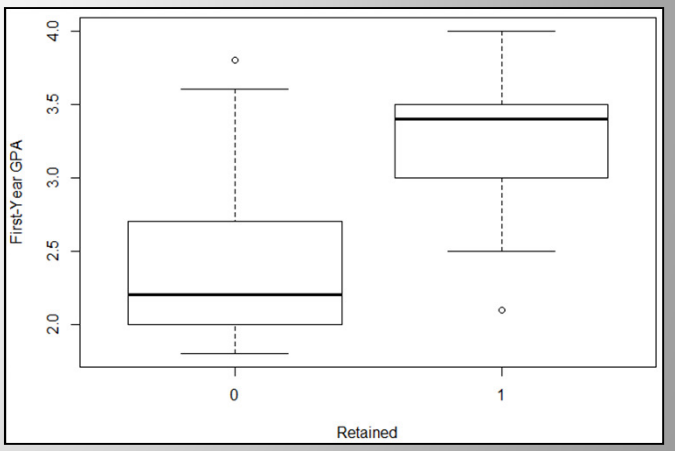
> t.test(GPA~Gender)

```
> t.test(GPA~Gender)

 Welch Two Sample t-test

data:  GPA by Gender
t = 0.1393, df = 47.885, p-value = 0.8898
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.3492603  0.4012603
sample estimates:
mean in group 1 mean in group 2
          2.890           2.864
```



> t.test(GPA~Retained)

```
> t.test(GPA~Retained)

 Welch Two Sample t-test

data:  GPA by Retained
t = -6.0801, df = 45.191, p-value = 2.338e-07
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.1437569 -0.5746006
sample estimates:
mean in group 0 mean in group 1
       2.413043        3.272222
```



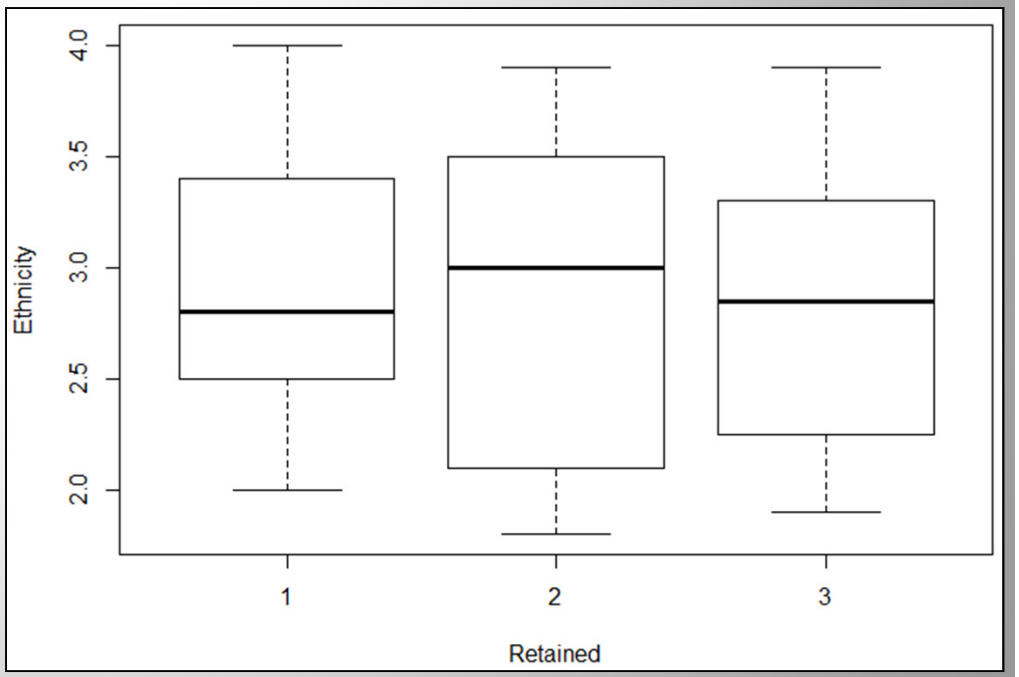
Paired Samples T-Test:

> t.test(pretest,posttest,paired=TRUE)

ANOVA

```
> Ex_anova <- aov(GPA~Ethnicity, students)
> anova(Ex_anova)
> boxplot(GPA~Ethnicity,data=students, xlab="Retained",
        ylab="Ethnicity")
```

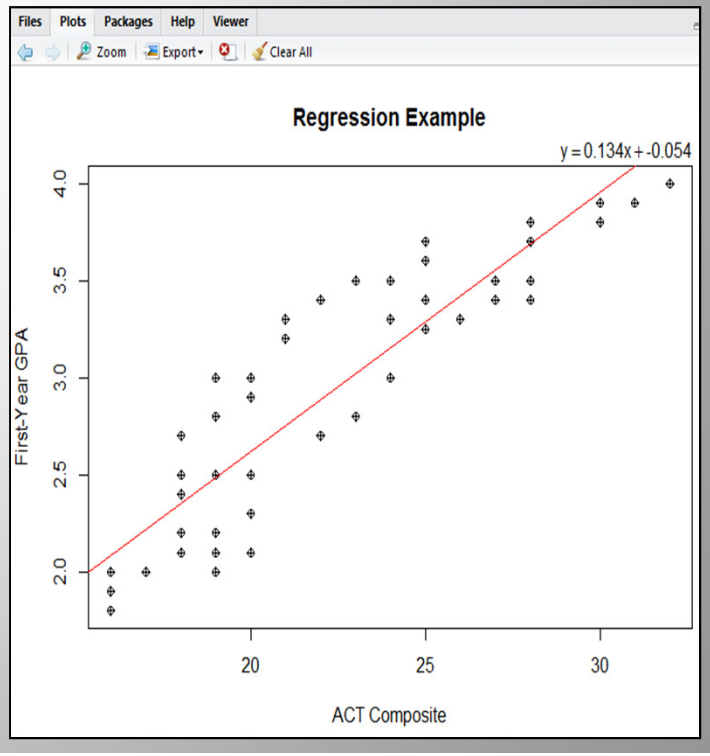
```
Analysis of Variance Table

Response: GPA
           Df  Sum Sq Mean Sq F value Pr(>F)
Ethnicity   1  0.0572 0.05716  0.1316 0.7184
Residuals  48 20.8489 0.43435
>
```



Regression Modeling: Linear

```
> mlr <- lm(GPA~ACT, students)
> summary(mlr)
> lm_coef<-round(coef(mlr),3)
> mtext(bquote(y==.(lm_coef[2])*x + .(lm_coef[1])),
        adj=1,pad    j=0)
```

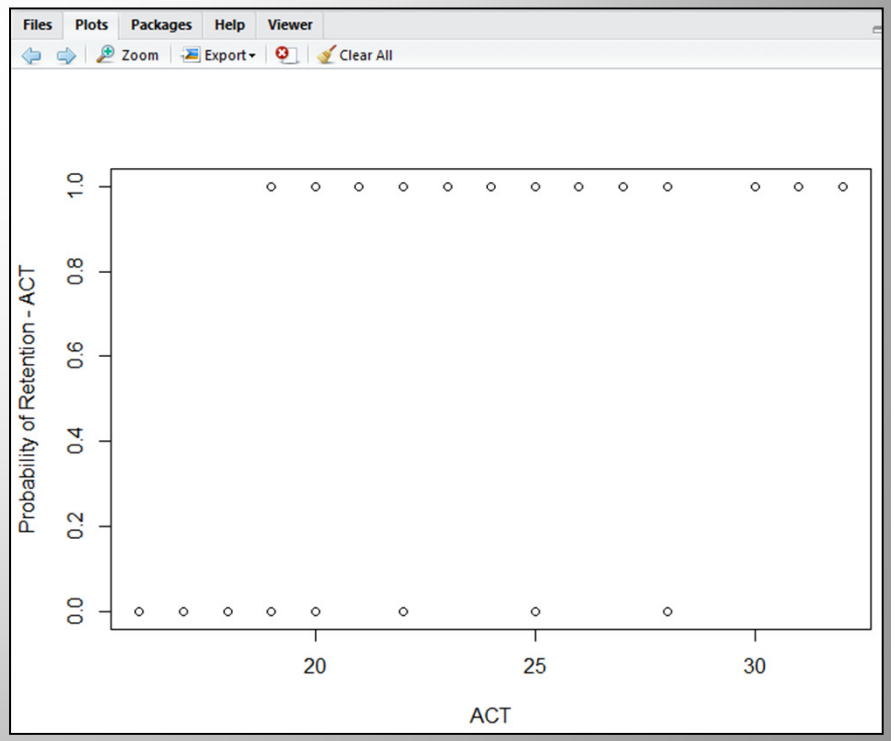
```
> mlr <- lm(GPA~ACT, students)
> summary(mlr)

Call:
lm(formula = GPA ~ ACT, data = students)

Residuals:
    Min      1Q  Median      3Q     Max
-0.5179 -0.2096 -0.1007  0.2489  0.5486

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.053821   0.206663   -0.26    0.796
ACT          0.133583   0.009238   14.46   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2852 on 48 degrees of freedom
Multiple R-squared:  0.8133, Adjusted R-squared:  0.8094
F-statistic: 209.1 on 1 and 48 DF,  p-value: < 2.2e-16
```



Regression Modeling: Logistic

```
> logreg <- glm(cbind(Retained)~ +ACT, family=binomial)
> summary(logreg)
> plot(ACT,Retained,xlab="ACT",ylab="Probability of
      Retention - ACT")
```

```
> summary(logreg)

Call:
glm(formula = cbind(Retained) ~ +ACT, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6486  -0.6719   0.1362   0.6241   1.5703

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -10.1044     2.8311  -3.569 0.000358 ***
ACT           0.4851     0.1374   3.530 0.000415 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 68.994  on 49  degrees of freedom
Residual deviance: 43.862  on 48  degrees of freedom
AIC: 47.862

Number of Fisher Scoring iterations: 5
```



Test for Proportion Differences

	Fall 2009	Fall 2013
First-Time Freshmen	367	385
2nd Year Retained	213	259
% Retained	58.0%	67.3%

```
> prop.test(x=c(213,259), n=c(367,385))

2-sample test for equality of prop. with continuity correction
data: c(213, 259) out of c(367, 385)
X-squared = 6.4667, df = 1, p-value = 0.01099
```

Merging Data Files

```
> stud_finaid <- merge(students, finaid, by="ID")
```

Default setting of the merge() function drops all unmatched cases. If you want to keep all cases in the new data set, include the option all=TRUE

```
> stud_finaid <- merge(students, finaid, by="ID",all=TRUE)
```

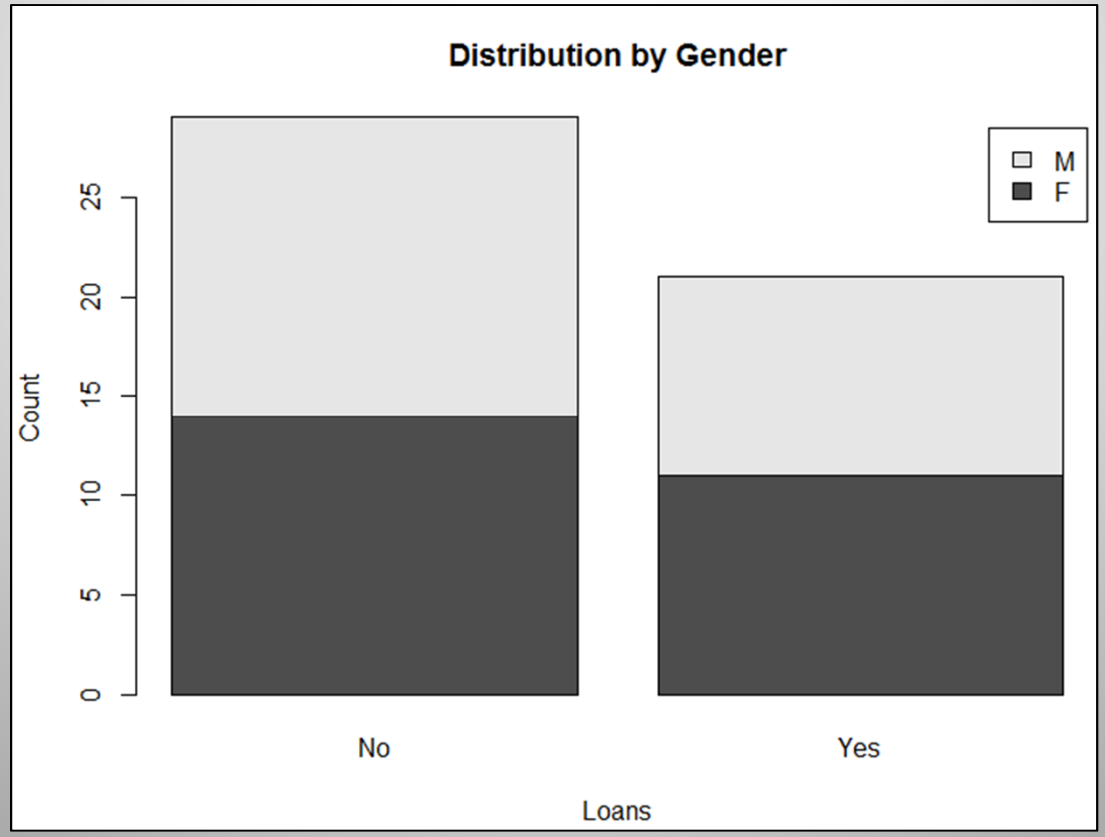
To keep unmatched cases only from students, use the all.x option.

To keep unmatched cases only from finaid, use the all.y option

```
> stud_finaid<- merge(students, finaid, by="ID", all.x=TRUE)
```

Merging Data Files

```
> table2 = table(stud_finaid$Gender,stud_finaid$Loans)
> barplot(table2, main='Distribution by Gender',
        xlab='Loans', ylab='Count',legend=rownames(table2))
```



Resources

- http://cran.r-project.org/doc/manuals/R-intro.html
- http://www.statmethods.net/interface/help.html
- http://www.r-tutor.com/
- http://www.r-bloggers.com/
- http://jason.bryer.org/
- http://stackoverflow.com/questions/tagged/r
- http://oit.utk.edu/scc/RforSAS&SPSSusers.pdf
- http://cran.r-project.org/doc/contrib/Short-refcard.pdf
- http://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf

Contact Information

Eric Atchison

601-432-6288

eatchison@mississippi.edu