Seok-Oh Jeong, In Heok Lee, and Jay W. Rojewski
University of Georgia

The *R* Workshop

Applying the Integrated Suite of Software Facilities for Statistical Computing and Graphics

University of Georgia
Department of Workforce Education, Leadership, and Social Foundations
College of Education Research Office

January 23-January 24, 2012

January 23-January 24, 2012

1. Introduction

University of Georgia
Department of Workforce Education, Leadership, and Social Foundations
College of Education Research Office

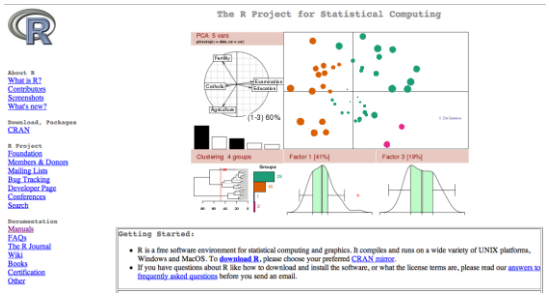
R is…

- A programming language and an environment for data manipulation, (statistical) computing, and graphical display.
- Powerful, but **FREE**!

Introduction

Download and Installation

http://www.r-project.org



Download and Installation

1. Click on "CRAN".
2. Select a mirror site near you.
3. Click on "Download and install R".
4. Click on "base".
5. Download the installation file and run it!

Console

- Run R, then a R-GUI window will appear.
- In R-GUI window, you'll see another window called 'R console'.
- Command prompt:

```
> 3+2
[1] 5
> pi
[1] 3.141693
```


Working Directory

- *Working directory* is the default location for all file input and output.
- Use `getwd()` to report the current working directory, and use `setwd()` to change your working directory.

```
> getwd( )
> setwd("c:/users/mywork")
```

- Or, from the main menu, select

  `"File" → "Change dir…"`

Help

- Need a help for persp()? Just type in the command prompt

  `> ? persp`

  or

  `> help(persp)`
- Need an extended help? Type

  `> ?? log`

  or

  `> help.search("log")`
- Online documentation: Visit R-project website and click on "Manuals".

Package(s)

- All R functions and datasets are stored in packages
- Installation of a package
  `> install.packages("package name")`
- Loading a package
  `> library("package name")`
- Unloading a package
  `> detach("package name")`

R command

- For variable names, we may use alphabets, numbers, period(.), underscore(_), etc.
- For assignment, <– is used. You may use =, but not preferable
- All names should begin with alphabet or period(.)
- Semicolon(;) separates multiple commands.
  `> beta.0 <- 3 ; beta.1 <- 2`
- Comments begin with #
  `> rnorm(100) # to generates 100 random numbers`

R command

- Use arrow keys for recalling former commands.
- Type the name of a variable to print its value onto console.

```
> beta.0
[1] 3
> beta.0 + 1
[1] 4
```


R command

- Case-sensitive
```
> a <- 1
> A <- 2
> a==A
[1] FALSE
```
- The objects are stored in R's database
  `> ls()  # list the objects stored in database`
- Run the script files.
  `> source("sample.R")`

January 23-January 24, 2012

2. Data Manipulation

University of Georgia
Department of Workforce Education, Leadership, and Social Foundations
College of Education Research Office

Data Types

- Vector
```
x <- c(1,2,3,4,5)
> y <- c("Clarke", "Oconee", "Barrow")
> x[3]
```
- Array
```
> z <- array(1:20, dim=c(4,5))
> z[4,5]
```
- Matrix
```
> z <- matrix(1:20, 4, 5)
> A <- matrix(2, 4, 5)
> z[3,2]
```


Data Types

- List
```
> Jeong <- list(first.name="Seok-Oh",
age=40, citizenship="South Korea")
```
- Data frame
```
> X <- c("Jeong", "Lee", "Rojewski")
> Y <- c(40, 35, 53)
> data <- data.frame(last.names=X,
age=Y)
```
- Factor
```
> z <- c("LD","BD", "BD", "LD", "Non",
"Non", "Non", "Non")
> z <- factor(z)
> levels(z)
```


Vectors

- Concatenation
```
> a <- c(2,2,2,2,2,2)
> a <- c(1, 2, 3); b <- c(5, 6)
> x <- c(a, 4, b) # x <- c(1,2,3,4,5,6)
```
- Sequence
```
> x <- seq(from=0, to=1, by=0.1)
> y <- seq(from=0, to=1, length=11)
> z <- 1:10
> rep(1, 10)
```

Vectors

- Arithmetic: Componentwise

```
> x <- 1:3; y <- c(2,2,2)
> x+y
> x-y
> x*y
> x/y
> x^y
> z <- rep(2, 5)
> x+z
> y-3
```


Vectors

- Mathematical functions

```
> x <- 1:10          > log(x)
> exp(x)             > sin(x)   # cos(x), tan(x)
> abs(x)             > sqrt(x)
> sort(x)            > length(x)
> min(x)             > max(x)
> mean(x)            > sum(x)
> prod(x)
```


Vectors

- Logical vectors

```
> x <- 1:10; y <- rep(5, 10)
> z <- x<5          # less than
> sum(z)
> x<=5              # less than or equal to
> x==5              # equal
> x!=5              # not equal
> (x>5)&(y<2)       # and
> (x<5)|(y<2)       # or
```

- Missing values

```
> x <- c(1, 2, 3, NA, 5)
> is.na(x)
```


Vectors

- Index vectors

```
> x <- -10:10
> x[3]
> x[1:3]
> x[c(1,3,5)]
> y <- x[x<0]
> x[x<0] <- -x[x<0]
> x <- c(1, 2, 3, NA, 5)
> x[!is.na(x)]
> x[is.na(x)] <- 4
> fruit <- c(5, 3, 2)
> names(fruit) <- c("apple", "orange", "peach")
> fruit[c("apple", "peach")]
```

Arrays and Matrices

- To generate an array and a matrix

```
> z <- array(1:20, dim=c(4,5))
> A <- matrix(1:20, 4, 5)
> B <- matrix(2, 4, 5)
> z[3,4]
> A[3,4]
> x <- c(1,2,3)
> y <- c(4,5,6)
> cbind(x, y)
> rbind(x, y)
> cbind(B, 1:4)
> C <- cbind(A, B)
```


Arrays and Matrices

- Arithmetic: Componentwise

```
> A <- matrix(1:20, 4, 5)
> B <- matrix(1:20, 4, 5)
> A+B
> A-B
> A*B
> A/B
```

- Arithmetic: Matrix multiplication, inverse

```
> A <- matrix(runif(20), 4, 5)
> B <- A%*%t(A)      # t(): transpose
> solve(B)           # inverse
```


Lists

- A list is an object consisting of a collection of objects called components.

```
> Jeong <- list(first.name="Seok-Oh", age=40,
  married=T, no.children=2, child.ages=c(9, 6))
> Jeong$age
> Jeong[[1]]
> Jeong$child.ages
> Jeong[[5]][1]
```


Factors

- A factor is a vector object used to specify a discrete classification (grouping) of the components of other vectors of the same length.

```
> x <- c(80, 90, 85, 85, 50, 60, 45, 50)
> z <- c("LD","BD", "BD", "LD", "Non", "Non",
"Non", "Non")
> z <- factor(z)
> levels(z)
> x.means <- tapply(x, z, mean)
```

Data frames

- A data frame is a list with restrictions:
  a. The components must be vectors (numeric, character, or logical), factors, numeric matrices, lists, or other data frames.
  b. Numeric vectors, logicals and factors are included as is, and character vectors are coerced to be factors, whose levels are the unique values appearing in the vector.
  c. Vector structures appearing as variables of the data frame must all have the same length, and matrix structures must all have the same row size.

January 23-January 24, 2012

3. Data Import/Export

University of Georgia
Department of Workforce Education, Leadership, and Social Foundations
College of Education Research Office

Read Data

- From the console: `scan()`

```
> x <- scan()
1: 1
2: 2
3: 3
4:
Read 3 items
> x
[1] 1 2 3
```


Read Data

- From a file: `scan()`

```
> x <- scan(file="c:/mydata/data_x.txt")
> y <- matrix(scan("c:/mydata/data_y.txt"),
ncol=3, byrow=T)
```

- From a file: `read.table()`

```
> x <- read.table(file="table.txt", header=T,
sep=" ")
```

- From a file: `read.csv()`

```
> x <- read.csv(file="table.csv", header=T)
```

Read Data

- Accessing built-in datasets: `data()`

```
> library("MatchIt")
> data("lalonde")
```

- Want the list of built-in datasets contained in the currently loaded packages? Just type `data()`.

```
> data()
```


Export Data

- To the console: `print()`

```
> x <- scan()
1: 1
2: 2
3: 3
4:
Read 3 items
> print(x)
[1] 1 2 3
```


Export Data

- To a file: `write()`

```
> x <- seq(from=0, to=1, by=0.1)
> write(x, file="output.txt")
```

- To a file: `write.table()`

```
> x <- matrix(1:20, 4, 5)
> write.table(x, file="table.txt")
```


January 23-January 24, 2012

4. Graphics:
Visualization of Data

University of Georgia
Department of Workforce Education, Leadership, and Social Foundations
College of Education Research Office

Distribution of One-Dimensional Data

- Qualitative data
  - Bar chart: `barplot()`
  - Pie chart: `pie()`
- Quantitative data
  - Stem-and-leaf plot: `stem()`
  - Histogram: `hist()`
  - Boxplot: `boxplot()`


```
## Beer Preference Example

beer <- c(3, 4, 1, 1, 3, 4, 3, 3, 1, 3, 2, 1, 2, 1, 2, 3, 2, 3, 1,
 1, 1, 1, 4, 3, 1)
# (1) Domestic can  (2) Domestic bottle,
# (3) Microbrew     (4) Import

barplot(table(beer))
barplot(table(beer)/length(beer),
  col=c("lightblue", "mistyrose", "lightcyan","cornsilk"),
  names.arg=c("Domestic can", "Domestic bottle", "Microbrew",
 "Import"),
  ylab="Relative frequency", main="Beer Preference Survey")

beer.counts <- table(beer) # store the table result
pie(beer.counts) # first pie -- kind of dull
names(beer.counts) <- c("Domestic\n can","Domestic\n bottle",
 "Microbrew","Import") # give names
pie(beer.counts) # prints out names
```



```
## Stem-and-leaf
scores <- c(2, 3, 16, 23, 14, 12, 4, 13, 2, 0, 0, 0,
   6, 28, 31, 14, 4, 8, 2, 5)
stem(scores)

## histogram
x <- rnorm(1000)  # To generate 1,000 random numbers from N(0,1)
hist(x, xlab="data")
hist(x, probability=T, xlab="data")
z <- seq(from=-3, to=3, by=0.01)
lines(z, dnorm(z), col=2)

## Boxplot
growth <- c(75,72,73,61,67,64,62,63) # the size of flies
sugar <- c("C","C","C","F","F","F","S","S") # diet
fly <- list(growth=growth, sugar=sugar)
boxplot(fly$growth)
jpeg(file="flygrowth.jpg", width=480, height=360)
```


Distribution of Multi-Dimensional Data

- Categorical and Quantitative Data
  - Boxplot: `boxplot()`
- Qualitative and Quantitative Data
  - Scatterplot: `plot()`

```
## Boxplot
boxplot(growth~sugar, xlab="Sugar Type", ylab="Growth",
   main="Growth against sugar types", data=fly)


## Scatterplot
plot(cars$speed, cars$dist)
# the speed of cars and the distances taken to stop
attach(cars)
plot(speed, dist, col="blue", pch="+",
   ylab="Distance taken to stop", xlab="Speed",
   ylim=c(-20, 140))
lm(dist~speed)
abline(-17.579, 3.932, col="red")
title(main="Scatterplot with best fit line", font.main=4)
```

Graphics

```
## Scatterplot matrix
attach(iris)
pairs(iris[,1:4])
pairs(iris[Species=="virginica", 1:4])


## 2D Histogram
library(hexbin)
plot(hexbin(iris[,3], iris[,4]),
  xlab="Petal Length", ylab="Petal Width")
```

Graphics

January 23-January 24, 2012

5. Advanced Programming

University of Georgia
Department of Workforce Education, Leadership, and Social Foundations
College of Education Research Office

Conditional Execution

- A conditional statement by 'if-else'

```
> if (x<3) print("x<3") else print("x>4")
```

- Commands can be grouped by braces

```
> x <- 4
> if ( x < 3 ) {print("x<3"); z <- "M"} else
{print("x>3"); z <- "F"}
```

Advanced Programming

Iteration, loop

- Loop: A repeatedly executed instruction cycle
- for-loop: loop over all elements in a vector

```
x <- 1:10
n <- length(x)
y <- rep(0, n)
for ( i in 1:n ) {
   y[i] <- x[i]^2
}
z <- x^2
print(cbind(y, z))
```


Iteration, loop

- while-loop: for which we don't know in advance how many iterations where will be

```
n <- 0
sum.so.far <- 0
while ( sum.so.far <= 1000 ) {
   n <- n+1
   sum.so.far <- sum.so.far + n
}
print(c(n, sum.so.far))
sum(1:45)
```

✓ **Whenever possible, try to avoid loops !**

Applying a function to every row/column

- To apply a function to every row[or column], use `apply()`.

```
> A <- matrix(1:20, 4, 5)
> apply(A, 1, sum)    # to every row
> apply(A, 2, sum)    # to every column
```

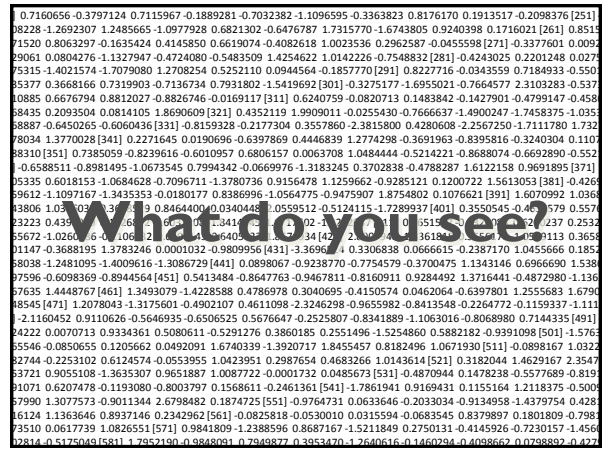

Writing New Functions

```
my.stat <- function(x)
  {
     m <- mean(x); s <- sd(x)
     res <- list(x=x, m=m, s=s)

     par(mfrow=c(1,2))
     boxplot(x, main="Boxplot")
     hist(x, prob=T, col="lightgray", main="Histrogram",
          xlab="data")
     z <- seq(from=min(x), to=max(x), by=0.01)
     lines(z, dnorm(z, mean=3, sd=1), col=2, lwd=3, lty=2)
     return(res)
  }

  X <- rnorm(1000, mean=3, sd=1)
  my.stat(x=X)
```

January 23-January 24, 2012

6. Summarizing Data

University of Georgia
Department of Workforce Education, Leadership, and Social Foundations
College of Education Research Office

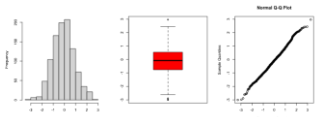


What or How to Summarize Your Data

- It depends on your research question(s)
- Do not make blind ventures with your data
- Avoid overcooking
- Use your intelligence and common sense

Distribution!

- Distributional information would be *be-all* and *end-all*
- Statistics such as mean, standard deviation, and regression coefficients stand for some partial (but important) features of the distribution
- Visualization is always the first step of your analysis



Summary Statistics

- Location (of the distribution):
  mean, median, trimmed mean...
- Scale (of the distribution):
  standard deviation, variance, range, IQR...
- Shape (of the distribution):
  skewness, kurtosis...
- Relative location (in the distribution):
  quantile, percentile, minimum, maximum...

Summarizing Data

```
library(MatchIt)
data(lalonde)
attach(lalonde)

boxplot(re78~treat)

re78.treat <- re78[treat==1]
re78.contr <- re78[treat==0]

mean(re78); sd(re78)
mean(re78.treat); sd(re78.treat)
mean(re78.contr); sd(re78.contr)

summary(lalonde)
```

Summarizing Data

```
summary.stats <- function(y)
{
x <- na.omit(y)        # Omit missing values
m <- mean(x)
s <- sd(x)
z <- (x-m)/s           # Standardization
skew <- mean(z^3)      # Skewness
kurt <- mean(z^4)      # Kurtosis
mini <- min(x)         # Minimum
maxi <- max(x)         # Maximum
q <- quantile(x, probs=c(.25, .50, .75))
res <- list(average=m, stdev=s,
            skewness=skew, kurtosis=kurt,
            q1=q[1], q2=q[2], q3=q[3],
            minimum=mini, maximum=maxi)

return(res)
}

summary.stats(re78.treat)
summary.stats(re78.contr)
```

Summarizing Data

```
table(treat, black)
table(treat, hisp)
table(treat, married)
table(treat, nodegr)
par(mfrow=c(1,2))
boxplot(age~treat, names=c("Control", "Treatment"), ylab="age")
boxplot(educ~treat, names=c("Control", "Treatment"),
ylab="educ")
```

Summarizing Data

January 23-January 24, 2012

7. One- and Two-Sample Tests

University of Georgia
Department of Workforce Education, Leadership, and Social Foundations
College of Education Research Office

One-sample t-test

- Given a sample from a population, want to know if the population mean could be a particular value μ_0?
  1. Compute the sample mean.
  2. Compute the normalized difference (t-statistic) between the sample mean and the hypothesized mean μ_0.
  3. Compare it with the reference distribution (t-distribution).

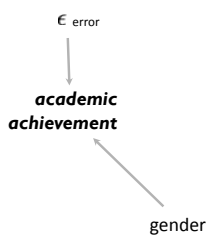
One-sample *t*-test

```
# Daily energy intake in kJ for 11 women
daily.intake <- c(5260, 5470, 5640, 6180, 6390, 6515,
  6805, 7515, 7515, 8230, 8770)
# To investigate whether women's intake deviates
  # from a recommended value of 7725kJ
mean(daily.intake)
sd(daily.intake)
boxplot(daily.intake); abline(h=7725, col=2, lty=2)
t.test(daily.intake, mu=7725, alternative="less")
# alternative = "greater", "two.sided"
```


One-sample Wilcoxon Signed-Rank Test

- Assume only that the data distribution is symmetric around the theoretical mean.
  1. Subtract the theoretical mean and rank the differences.
  2. Ignoring the sign, calculate the sum of positive or negative ranks.

```
wilcox.test(daily.intake, mu=7725)
```



id	gender	GPA
1	M	4.0
2	M	3.8
3	M	3.5
4	M	3.1
5	F	3.3
6	F	3.5
7	F	4.0
8	F	4.2
9	F	3.8
10	F	3.7

Two-sample t-test

- Given one sample each from two population, want to know if the population means could differ each other?
  1. Compute the sample means.
  2. Compute the normalized difference (t-statistic) between the sample.
  3. Compare it with the reference distribution (t-distribution).

Two-sample Tests

```
group <- c(rep("M", 4), rep("F", 6))
y <- c(4.0, 3.8, 3.5, 3.1, 3.3, 3.5, 4.0, 4.2, 3.8, 3.7)
boxplot(y~group)

# Two-sample t-test
t.test(y~group)

# Wilcoxon rank-sum test
wilcox.test(y~group)
```


Paired Tests

```
library(ISwR)
attach(intake)
intake
t.test(pre, post, paired=T)
t.test(pre, post) # WRONG!!!
wilcox.test(pre, post, paired=T)
# Or...
diff <- post-pre
t.test(diff, mu=0)
wilcox.test(diff, mu=0)
```

January 23-January 24, 2012

8. Analysis of Variance (*ANOVA*)

University of Georgia
Department of Workforce Education, Leadership, and Social Foundations
College of Education Research Office

ϵ error

academic achievement

school

id	school	GPA
1	C	4.0
2	B	3.8
3	B	3.5
4	B	3.1
5	A	3.3
6	A	3.5
7	C	4.0
8	C	4.2
9	A	3.8
10	C	3.7

ANOVA

Compare Means

Group A: **treatment level 1**

Group B: **treatment level 2**

Group C: **control**

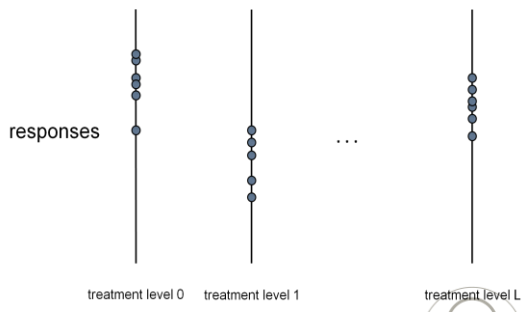
ANOVA

One-way ANOVA
Two-way ANOVA
Multi-way ANOVA

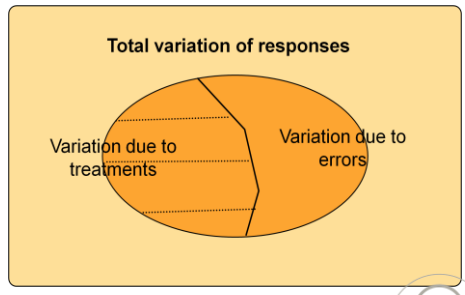
Treatment level

Treatment effect
- main effect
- interaction

ANOVA



ANOVA



ANOVA

```
school <- c(3, 2, 2, 2, 1, 1, 3, 3, 1, 3)
school <- factor(school, labels=c("A", "B", "C"))
y <- c(4.0, 3.8, 3.5, 3.1, 3.3, 3.5, 4.0, 4.2, 3.8, 3.7)
boxplot(y~school)

# ANOVA
res <- lm(y~school)
anova(res)
summary(res)
pairwise.t.test(y, school, p.adj="bonferroni")
res <- aov(y~school)
TukeyHSD(res)

# Kruskal-Wallis test
kruskal.test(y~school)
```

ANOVA

```
# Two-way ANOVA

# Heart rate after administration of enalaprilate
library(ISwR)
attach(heart.rate)
heart.rate
anova(lm(hr~subj+time))

# Nonparametric counterpart
friedman.test(hr~time|subj)
```

ANOVA

January 23-January 24, 2012

9. Regression

University of Georgia
Department of Workforce Education, Leadership, and Social Foundations
College of Education Research Office

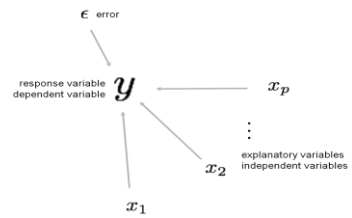
id	SAT	GEN	GPA
1	388	F	4.0
2	354	F	3.8
3	361	F	3.5
4	329	F	3.1
5	331	M	3.3
6	364	M	3.5
7	399	M	4.0
8	421	F	4.2
9	398	M	3.8
10	383	M	3.7

Correlation

- Pearson's correlation
- Kendall's tau
- Spearman's rank correlation

```
SAT <- c(388, 354, 361, 329, 331, 364, 399, 421, 398, 383)
GPA <- c(4.0, 3.8, 3.5, 3.1, 3.3, 3.5, 4.0, 4.2, 3.8, 3.7)
cor(SAT, GPA, method="pearson")
cor(SAT, GPA, method="kendall")
cor(SAT, GPA, method="spearman")
```


Regression Model



$$y = m(x_1, x_2, \cdots, x_p) + \epsilon$$

where $m(x_1, x_2, \cdots, x_p) = E(y|x_1, x_2, \cdots, x_p)$

and $\epsilon \sim \bullet(0, \sigma^2)$.

Linear model

$$m(x_1, x_2, \cdots, x_p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

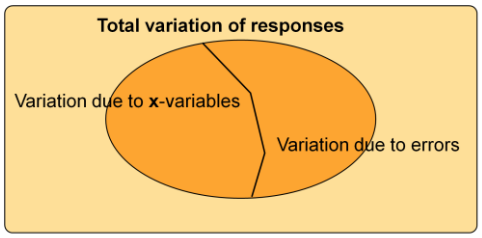
the expected change in y per unit change in x_1 when all the other regressors are held constant

: partial regression coefficient

Nonlinear model

(e.g) $m(x) = A \cdot exp(-\beta x)$

Regression



Null hypothesis.: The regression model is **not** significant

F-test again!

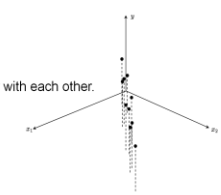
Regression

```
SAT <- c(388, 354, 361, 329, 331, 364, 399, 421, 398, 383)
GEN <- c("F", "F", "F", "F", "M", "M", "M", "F", "M", "M")
GPA <- c(4.0, 3.8, 3.5, 3.1, 3.3, 3.5, 4.0, 4.2, 3.8, 3.7)

par(mfrow=c(1,2))
plot(SAT, GPA)
boxplot(GPA~GEN, xlab="GENDER", ylab="GPA")

res <- lsfit(SAT, GPA)
ls.print(res)

res.no <- lsfit(SAT, GPA, intercept=F)
ls.print(res.no)

t.test(GPA~GEN)

res.lm <- lm(GPA~GEN+SAT)
summary(res.lm)
```

Regression

Multicollinearity

Regressors may have (nearly) linear dependency with each other.



VIF: variance inflation factor

$$\mathrm{VIF}_j > 10$$

Regression

Remedies when multicollinearity is detected:

- ✓ Model re-specification
- ✓ Ridge regression
- ✓ Principal component regression

Variable Selection

- Stepwise regression

```
> data(state)
> statedata <- data.frame(state.x77,
     row.names=state.abb,check.names=T)
> g <- lm(Life.Exp~., data=statedata)
> summary(g)
> step(g)
```


January 23-January 24, 2012

10. Categorical Data Analysis

University of Georgia
Department of Workforce Education, Leadership, and Social Foundations
College of Education Research Office

Any evidence of association between **education level** and **income level**? If any, **how strong** is it?

Education	Income		sum
	Low	High	
High school	25	12	37
College	11	14	25
sum	36	26	62

To test whether

*there is **no** association between factors*

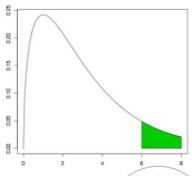
- or -

*there **is** an association between factors*

Observed cell count

$$\chi^2 = \sum \frac{(O-E)^2}{E} \sim \chi^2(\nu)$$

Expected cell count



Categorical Data Analysis

```
M <- as.table(rbind(c(25, 12), c(11,14)))
dimnames(M) <- list(education=c("High School","College"),
                    income=c("Low","High"))
res <- chisq.test(M)
res$expected
res
```

Categorical Data Analysis