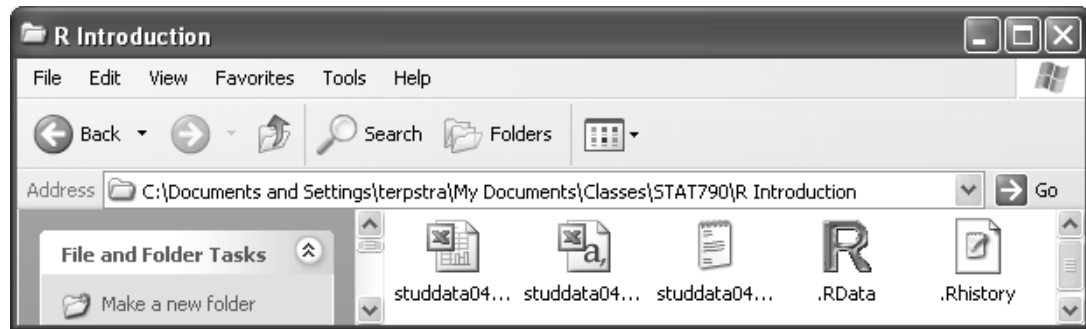


# **A Brief Introduction to R**

# Project Directories

- An Example Project Directory



- The .RData File

- This image file contains all of the R objects for a particular workspace.
- File → Load [Save] Workspace ...

- The .Rhistory File

- This text file contains all of the command lines for a particular R session.
- File → Load [Save] History ...

# R's Command Line

- Help Commands

```
> help(help)
```

```
> help(t.test)
```

```
> help.search("t.test")
```

- R as a Calculator

```
> 1+3
```

```
[1] 4
```

```
> x<-4
```

```
> x
```

```
[1] 4
```

```
> sqrt(25)
```

```
[1] 5
```

```
> x<-1:3
```

```
> x
```

```
[1] 1 2 3
```

```
> x+2
```

```
[1] 3 4 5
```

## ● Combining Commands

```
> e<-rnorm(10);y<-5+e;print(y)
[1] 6.047218 6.309090 4.458829 5.933691 5.350011
[6] 5.809859 4.925154 2.386602 2.774224 5.571341
```

```
> x<-1;{x<-2;print(x)}
[1] 2
```

# R Vectors

- **Constructing Numeric Vectors**

```
> 1:3  
[1] 1 2 3
```

```
> c(1:3,5:7)  
[1] 1 2 3 5 6 7
```

```
> x<-scan()  
1: 1  
2: 10  
3: 100  
4:  
Read 3 items  
> x  
[1] 1 10 100
```

```
> seq(1,10,2)  
[1] 1 3 5 7 9
```

- **Constructing Character Vectors**

```
> c("A","B","C")  
[1] "A" "B" "C"
```

```
> paste("A",1:5,sep="")  
[1] "A1" "A2" "A3" "A4" "A5"
```

## ● Constructing Logical Vectors

```
> x<-1:5  
> x>3  
[1] FALSE FALSE FALSE  TRUE  TRUE
```

## ● Vector Operations

```
> x<-c(1,10,100)  
> x[c(1,3)]  
[1]  1 100
```

```
> x+2  
[1]  3 12 102
```

```
> sqrt(x)  
[1] 1.000000  3.162278 10.000000
```

# R Matrices

## ● Constructing Matrices

```
> cbind(1:2,3:4)
```

```
      [,1] [,2]
[1,]    1    3
[2,]    2    4
```

```
> rbind(1:2,3:4)
```

```
      [,1] [,2]
[1,]    1    2
[2,]    3    4
```

```
> x<-matrix(1:4,byrow=TRUE,ncol=2)
```

```
> x
```

```
      [,1] [,2]
[1,]    1    2
[2,]    3    4
```

## ● Accessing Matrix Elements

```
> x[,1]
```

```
[1] 1 3
```

```
> x[1,]
```

```
[1] 1 2
```

```
> x[1,2]
```

```
[1] 2
```

## ● Matrix Multiplication

```
> x%*%cbind(0:1)
```

```
      [,1]
```

```
[1,]    3
```

```
[2,]    4
```

# R Lists

## ● Constructing Lists

```
> x<-1:3
> y<-c("A","B")
> z<-y=="A"
> lst<-list(x=x,y=y,z=z)
> lst
$x
[1] 1 2 3

$y
[1] "A" "B"

$z
[1] TRUE FALSE
```

## ● Accessing List Components

```
> lst[[1]]
[1] 1 2 3

> lst$x
[1] 1 2 3

> lst$z
[1] TRUE FALSE
```

# R Data Frames

- Data Object with Continuous and Discrete Random Variables
- Constructing a Data Frame in R

```
> y=c(rnorm(5,mean=0),rnorm(5,mean=3))
> x=rep(c("One","Two"),c(5,5))
> z=data.frame(y=y,x=as.factor(x))
> z
```

	y	x
1	-0.2047465	One
2	-0.4098199	One
3	0.3604416	One
4	-0.8666295	One
5	0.7997580	One
6	3.3613841	Two
7	3.4539022	Two
8	4.5531913	Two
9	1.4619965	Two
10	2.5606199	Two

```
> summary(z)
```

	y	x
Min.	:-0.86663	One:5
1st Qu.:	:-0.06345	Two:5
Median :	1.13088	
Mean :	1.50701	
3rd Qu.:	3.16119	
Max.	: 4.55319	

## ● Copying and Pasting Data into a Data Frame

```
> y=read.table(stdin(),header=TRUE)
```

```
0: Cross      Self
1: 23.500    17.375
2: 12.000    20.375
3: 21.000    20.000
4: 22.000    20.000
5: 19.125    18.375
6: 21.550    18.625
7: 22.125    18.625
8: 20.375    15.250
9: 18.250    16.500
10: 21.625    18.000
11: 23.250    16.250
12: 21.000    18.000
13: 22.125    12.750
14: 23.000    15.500
15: 12.000    18.000
16:
```

```
> y[1:2,]
```

```
  Cross      Self
1  23.5  17.375
2  12.0  20.375
```

## ● .xls to .csv to an R Data Frame

```
> examp1=read.csv("examp1.csv")
```

```
> dim(examp1);examp1[1:2,]
```

```
[1] 9 5
```

```
  Gender Age Hand Weight  GPA
1      M  22   R    180 3.76
2      F  21   R    154 2.12
```

# Descriptive Statistics

- Read Sample Data In from a .csv File

```
> examp1=read.csv("examp1.csv")
> examp1
  Gender Age Hand Weight  GPA
1      M  22   R   180 3.76
2      F  21   R   154 2.12
3      F  19   R   104 1.90
4      M  18   R   210 3.45
5      F  25   R   160 3.56
6      M  23   R   210 3.00
7      F  23   L   114 2.56
8      M  21   R   165 2.10
9      M  26   L   178 3.90
```

- Sample Mean, Median, and Standard Deviation

```
> mean(examp1$Age)
[1] 22

> median(examp1[,5])
[1] 3

> sqrt(var(examp1$GPA))
[1] 0.7781995
```

- The Five Number Summary

```
> fivenum(examp1[,2])
[1] 18 21 22 23 26
```

- **A One-Way Frequency Table**

```
> table(examp1$Gender)
```

```
F M
```

```
4 5
```

- **A Two-Way Contingency Table (with Relative Frequencies)**

```
> table(examp1[,c(1,3)])/dim(examp1)[1]
```

```
Hand
```

```
Gender          L          R
```

```
F 0.1111111 0.3333333
```

```
M 0.1111111 0.4444444
```

# Basic Inference

- One Sample T-Test

```
> t.test(examp1$Weight,mu=150)
```

```
One Sample t-test
```

```
data:  examp1$Weight
t = 1.1288, df = 8, p-value = 0.2917
alternative hypothesis:
true mean is not equal to 150
95 percent confidence interval:
 135.5146 192.2632
sample estimates:
mean of x
 163.8889
```

- Two Sample T-Test

```
> t.test(Weight~Gender,data=examp1,var.equal=TRUE)
```

```
Two Sample t-test
```

```
data:  Weight by Gender
t = -3.4538, df = 7, p-value = 0.01064
alternative hypothesis:
true difference in means is not equal to 0
95 percent confidence interval:
 -93.66594 -17.53406
sample estimates:
```

mean in group F	mean in group M
133.0	188.6

## ● Simple Linear Regression

```
> temp=lm(Weight~Age,data=examp1)
> summary(temp)
```

Call:

```
lm(formula = Weight ~ Age, data = examp1)
```

Residuals:

Min	1Q	Median	3Q	Max
-57.389	-9.056	1.944	16.111	49.444

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	145.5556	118.6683	1.227	0.260
Age	0.8333	5.3609	0.155	0.881

Residual standard error: 39.39

on 7 degrees of freedom

Multiple R-Squared: 0.00344,

Adjusted R-squared: -0.1389

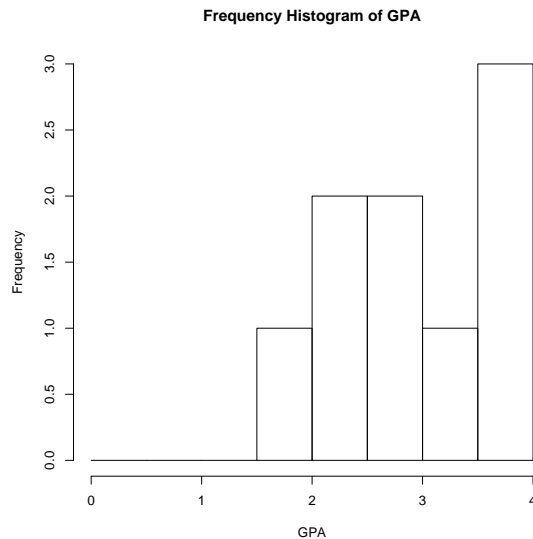
F-statistic: 0.02416 on 1 and 7 DF,

p-value: 0.8809

# R Plot Functions

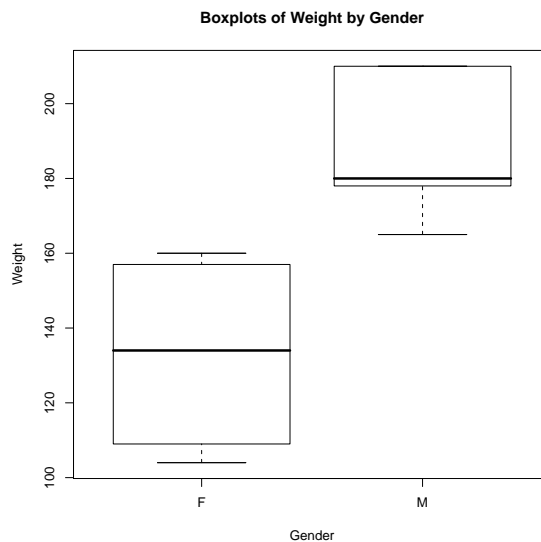
## ● Simple Histogram

```
> hist(examp1$GPA,breaks=seq(0,4,.5),xlab="GPA",  
+ main="Frequency Histogram of GPA")
```



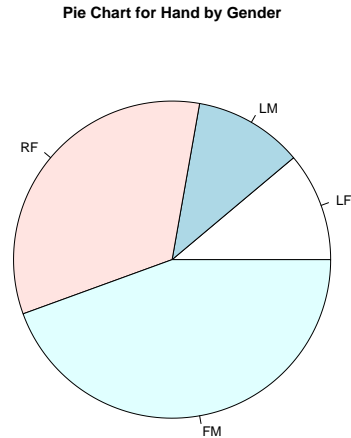
## ● Comparison Boxplots

```
> boxplot(Weight~Gender,data=examp1,xlab="Gender",  
+ ylab="Weight",main="Boxplots of Weight by  
+ Gender")
```



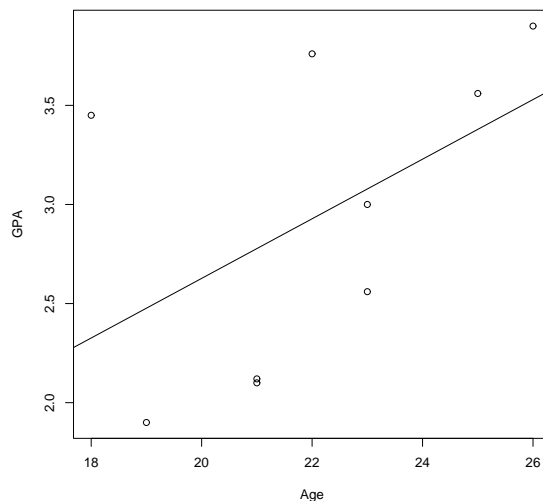
## ● Simple Pie Chart

```
> pie(table(examp1[,c(1,3)]),c("LF","LM","RF",  
+ "FM"), main="Pie Chart for Hand by Gender")
```



## ● Scatter Plot with Regression Fit

```
> plot(examp1$Age,examp1$GPA,xlab="Age",ylab="GPA")  
> abline(lm(GPA~Age,data=examp1))
```



# Creating Design Matrices from Data Frames

- Multiple Linear Regression

```
> model.matrix(GPA~Age+Weight,data=examp1)
  (Intercept) Age Weight
1             1  22   180
2             1  21   154
3             1  19   104
4             1  18   210
5             1  25   160
6             1  23   210
7             1  23   114
8             1  21   165
9             1  26   178
attr(,"assign")
[1] 0 1 2
```

- One-Way ANOVA model (2 groups)

```
> model.matrix(GPA~Gender,data=examp1,
+ contrasts.arg=list(Gender="contr.sum"))
  (Intercept) Gender1
1             1      -1
2             1       1
3             1       1
4             1      -1
5             1       1
6             1      -1
7             1       1
8             1      -1
```

```

9           1       -1
attr(,"assign")
[1] 0 1
attr(,"contrasts")
attr(,"contrasts")$Gender
[1] "contr.sum"

```

## ● 2-Way ANOVA with Interaction

```

> model.matrix(GPA~Gender*Hand,data=examp1,
+ contrasts.arg=list(Gender="contr.SAS",
+ Hand="contr.SAS"))

```

	(Intercept)	GenderF	HandL	GenderF:HandL
1	1	0	0	0
2	1	1	0	0
3	1	1	0	0
4	1	0	0	0
5	1	1	0	0
6	1	0	0	0
7	1	1	1	1
8	1	0	0	0
9	1	0	1	0

```

attr(,"assign")
[1] 0 1 2 3
attr(,"contrasts")
attr(,"contrasts")$Gender
[1] "contr.SAS"
attr(,"contrasts")$Hand
[1] "contr.SAS"

```

## ● Analysis of Covariance with Interaction

```
> model.matrix(GPA~Age+Gender+Age:Gender,  
+ data=examp1,  
+ contrasts.arg=list(Gender="contr.sum"))  
  (Intercept) Age Gender1 Age:Gender1  
1           1  22      -1         -22  
2           1  21       1          21  
3           1  19       1          19  
4           1  18      -1         -18  
5           1  25       1          25  
6           1  23      -1         -23  
7           1  23       1          23  
8           1  21      -1         -21  
9           1  26      -1         -26  
attr(,"assign")  
[1] 0 1 2 3  
attr(,"contrasts")  
attr(,"contrasts")$Gender  
[1] "contr.sum"
```

## (User Defined) R Functions

- Simple Text File Containing a User Defined R Function

```
#
# Function that sets up pairwise comparisons
#
# x: vector or matrix
#
pairup=function(x,type="less") {
  x=as.matrix(x)
  n=dim(x)[1]
  i=rep(1:n,rep(n,n))
  j=rep(1:n,n)
  c1=apply(x,2,function(y){rep(y,rep(length(y),
    length(y)))})
  c2=apply(x,2,function(y){rep(y,
    length(y))})
  ans=cbind(c1,c2)
  ans=switch(type, less=ans[(i<j), ],
    leq=ans[i<=j, ], neq=ans)
  ans
}
```

- “Sourcing” R Commands and Functions

```
> source("pairup.r")
> #pairup can now be used
> ls(pattern="p")
[1] "examp1" "pairup" "temp"
```

## ● Examples of the pairup function

```
> pairup(1:3)
      [,1] [,2]
[1,]    1    2
[2,]    1    3
[3,]    2    3
```

```
> pairup(1:3,"leq")
      [,1] [,2]
[1,]    1    1
[2,]    1    2
[3,]    1    3
[4,]    2    2
[5,]    2    3
[6,]    3    3
```

```
> x=rbind(1:2,3:4,5:6)
> pairup(x)
      [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    1    2    5    6
[3,]    3    4    5    6
```

```
> x=rbind(1:3,4:6,7:9)
> #Upper triangle of x matrix
> x[pairup(1:3,"leq")]
[1] 1 2 3 5 6 9
```