

My frame of reference

- Type the following:

```
attach(D)
```

```
name
```

```
assistants <- c(0, 0, 1, 1, 3, 2, 4)
```

```
detach()
```

Levels and factors

- Type this:

```
D$gender <- c("M", "F", "M", "M", "F", "F", "F")
```

```
D$gender <- factor(D$gender)
```

```
levels(D$gender)
```

tapply

- Type this:

```
tapply(D$net.worth, D$gender, mean)
```

Creating matrices

- Type this:

```
v1 <- c(1,2,3,4,5,6)
```

```
matrix(v1, nrow=2)
```

```
matrix(v1, ncol=2)
```

```
M <- matrix(v1, nrow=2, byrow=TRUE)
```

Extracting from matrices

- Type this:

```
M[1, 1]
```

```
M[1:2, 1:3]
```

```
M[1, c(1, 3)]
```

```
M[1, ]
```

```
M[, 1]
```

Getting the size of a matrix

- Type this:

```
dim(M)
```

Things we covered in part III

- Data types: *data frames*
- Creating data frames
- Adding and removing data from data frames
- Indexing data frames
- Levels and factors
- *tapply*

Part IV: Files

Saving and loading sessions

- Don't type anything for once!

Saving specific things

- To save something in R format:

```
save(D, file="D-table.R")
```

```
rm(D)
```

```
load(file="D-table.R")
```

Loading tables in text files

- Type this:

```
help(read.table)
```

```
D2 <-  
read.table("http://www.trii.org/courses/  
rcourse/D-table.txt")
```

To load tables...

- Plain-text files only
- First row should contain column names
- Items should be separated by
 - whitespace (*read.table*),
 - commas (*read.csv*), or
 - tab characters (*read.delim*)

Writing out tables to text files

- Type this:

```
help(write.table)
```

```
write.table(D2,  
            "~/Desktop/delete-me-now.txt")
```

Getting data to/from Excel

- In Excel, save in CSV format
- Use `read.csv` to load data into R
- Use `write.csv` to save data from R
- Load CSV file into Excel

ACTIVITY #5

- Open a web browser and go to <http://www.trii.org/courses/rcourse/>
- Download the *malpractice.xls* file
- Convert into CSV file using Excel
- Load it into R using *read.csv*

Maybe it is Excel...

- Type this:

```
edit(T)
```

where T is the name of the table you loaded previously

Writing scripts

- Scripts are files that contain a series of R commands
- Must be saved as plain text, usually with a “.r” extension
- Load them into R using the *source* command:

```
source("my-example-script.r")
```

ACTIVITY #6

- Write a script to create a 3x3 matrix with the numbers 1 through 9 in it.
- Make sure the script puts the matrix in a variable named *I.love.this.matrix*.
- Load your script and make sure it works.

Things we covered in part IV

- Loading and saving sessions
- Loading and saving specific things
- Loading tables in text files
- Writing tables to text files
- Getting data to/from Excel
- Editing tables using the built-in editor
- Writing and using scripts

Part V: Plotting

A sinister plot

- Type this:

```
attach(D)
```

```
plot(age)
```

```
plot(age, net.worth)
```

```
plot(age, net.worth, log="xy")
```

What about lines?

- Type this:

```
plot(age, net.worth, type="l")
```

What about color?

- Type this:

```
Plot(age, net.worth, type="l",  
      col="blue")
```

ACTIVITY #7

- Consult R's help system and find out how to add titles and label axes.
- Using the malpractice data, create a scatter plot of age vs. premium with appropriate titles and labels.

Plot on plot

- The following functions add new items to an existing plot:
 - *lines* adds new lines
 - *points* adds new points
 - *abline* adds a single straight line to the plot
 - *text* adds text at a given position
 - *legend* adds a legend to the plot

Activity #8

- Calculate sin and cos of 100 points between 0 and 2π .
- Plot the sin curve as a line in red.
- Plot the cos curve as a line in blue.
- Add an appropriate title.
- Put text on the plot to label the curves. Make sure the text is near the right curve.

Customizing your plot

- Type this:

```
plot(age, net.worth, type="l", bty="l",  
      fg="red", lty=3, col="green")
```

```
help(par)
```

Other plot types

- A histogram:

```
hist(rnorm(1000))
```

- A pairwise plot:

```
pairs(D)
```

Multiple plots

```
mflow(2, 2)
par(mfg=c(1, 1))
plot(c(1, 2, 3), c(3, 4, 2))
par(mfg=c(2, 1))
plot(c(1, 2, 3), c(10, 40, 25))
par(mfg=c(1, 2))
barplot(c(10, 40, 25))
```

Exporting graphics

```
pdf("hi mom.pdf")  
plot(c(1, 2, 3, 4, 5))  
dev.off()  
quartz()
```

A couple more things to mention

- `ls()`
- `rm()`
- `help()`
- `apropos()`

Things we covered in part IV

- Making a basic plot
- Adding details to the plot
- Histograms, pairwise plots
- Multiple plots
- Exporting graphics
- Extras: `ls()`, `rm()`, `help()`, `apropos()`