Extra Topics 1: Data and Graphing

1. R markdown options
2. “Base” graphics options
3. ggplot2 package
4. gplots package

Examples:

1. Importing Excel files
2. Organizing data in spreadsheets
3. R markdown “Ugly” and “Pretty” examples
4. Graphing Examples

1. R markdown options

- Add sections of plain text. Anything that is not within a code chunk appears as plain text.
- To start a new paragraph (of plain text), end a line with two spaces.
- Include headers (#, ## etc).
- Insert inline equations ($....$). These can include greek letters and other symbols.
- Use multiple code chunks. There is no charge!!!
- Use options on the next slide to control what code and output is shown in the final document.
- Use inline R code (`r ….`).

Chunk Options:

- include = FALSE prevents code and results from appearing in the finished file. R Markdown still runs the code in the chunk, and the results can be used by other chunks.
- echo = FALSE prevents code, but not the results from appearing in the finished file.
- message = FALSE prevents messages that are generated by code from appearing in the finished file. Can be handy when loading packages (ex: tidyverse!)
- warning = FALSE prevents warnings that are generated by code from appearing in the finished.
- fig.length = X, fig.height = X to control figure sizes.

- For “pretty” tables, use kable() and the kableExtra package.

A few more ideas (but these are not shown in example):

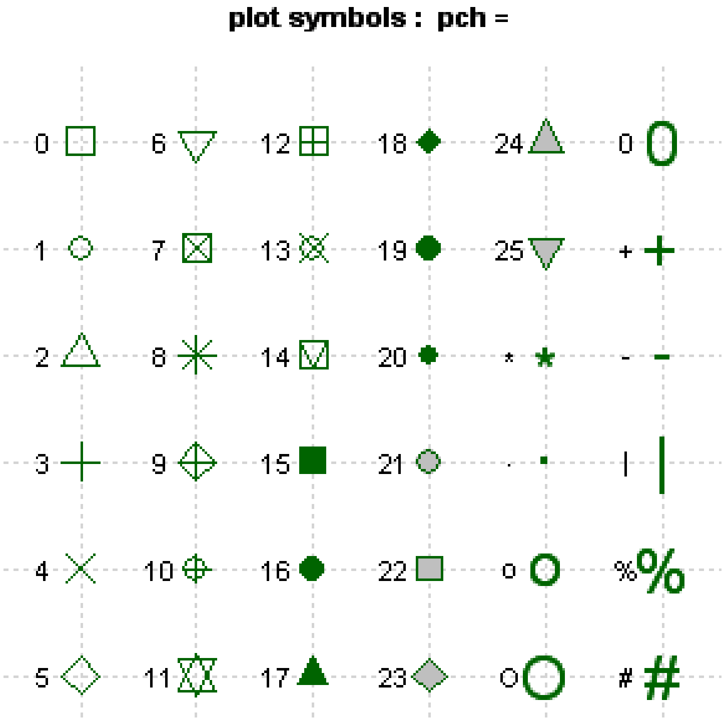
- Use bold (*…*) and italics (*…*).
- Include bullets and ordered lists.
- For pdf documents, insert page breaks (\newpage).

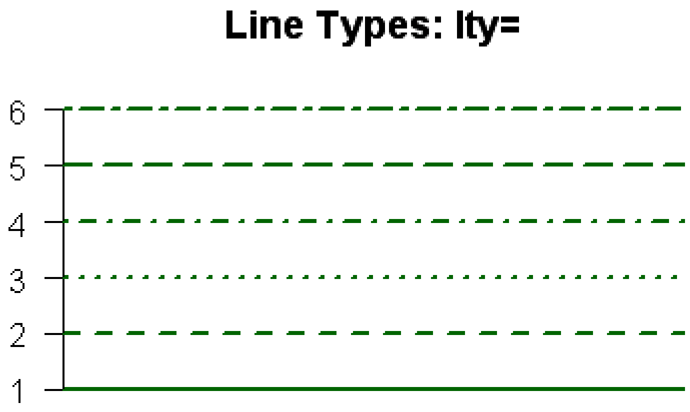
2. “Base” graphics options

There are several graphical parameters that can be adjusted.

This first list gives options that are used within the plot() function.

- **type** controls what type of plot should be drawn: “p” for points, “l” for lines, “b” for both, “n” for none, etc.
- **main** gives the overall title for the graph.
- **xlab** and **ylab** give the titles for the x and y labels.
- **xlim** and **ylim** give the limits for the x and y axes.
- **cex** adjusts the text size. Cex is a number indicating the amount by which plotting text and symbols should be scaled relative to the default. 1=default, 1.5 is 50% larger, 0.5 is 50%.





Images from http://www.statmethods.net/advgraphs/parameters.html

- **pch** controls the plot symbols, see above for options.
- **lwd** controls the line width.
- **lty** controls the line type, see above for options.
- **col** controls the colors for lines and symbols. Can be specified by name or number. Too many to list – google R colors!

This second list gives options that are used outside the plot() function.

- **par()** controls various graphic parameters.
  - par(mfrow=c(2,2)) puts 4 plots in a single "window".
  - par(new=TRUE) overlays a new graph on top of an existing graph.
- **text()**, **points()**, **lines()**, **abline()** are additional ways to add information to an existing graph.
- **title()**
- **legend()**
- **axis()** create custom axes.

- Important note about plot() syntax. There are two equivalent ways to create a scatterplot in R with the plot() function:

```
plot(y ~ x, data = InData)
plot(InData$x, InData$y)
```

- Note that the first approach (formula syntax) matches the syntax of the lm() function and also allows the data = option.

3. ggplot2 package

- **`ggplot2`** by Wickham and Chang et al. is a powerful package for customized graphics.
- Important: ggplot2 functions can be chained with "+" signs to generate the final plot. <u>Builds plots in layers</u>.
- ggplot2 is named after the book “Grammar of Graphics” by Leland Wilkinson.
- See http://www.cookbook-r.com/Graphs/ for many examples.
- Note: Unlike base R graphs, the ggplot2 graphs are not effected by many of the options set in the par( ) function.

ggplot2: Basic Syntax

- The `qplot()` function can be used to create most common graph types.

```
qplot(x, y, data = InData)
```

- For a basic scatterplot this is equivalent to:

```
ggplot(InData,aes(x= ,y= )) +
   geom_point()
```

- Can save the graphical results to an R object.
- Creates legends automatically.

ggplot2: Adding Layers

- `geom_smooth()` adds a smooth fitted curve (LOESS is the default). Use `geom_smooth(method = “lm”)` to add regression line (or more generally a linear model).
- `labs()` and `ggtitle()` modify labels, legends and title.
- `facet_grid(row.variable ~ column.variable)` creates tables of graphs by splitting the data into subsets. Very handy!!
- General Strategy: Create a ggplot object and add layers and adjust labels, legends, faceting etc... in an incremental process

4. gplots package

- I consider this a “one hit wonder” for making heatmaps.
- **`heatmap.2()`** for enhanced heatmaps. A heat map is a graphical representation of data where the individual values contained in a matrix are represented as colors.
- **NOTE:** For a quick (but pretty) Venn Diagram try Venny by Oliveros:

  http://bioinfogp.cnb.csic.es/tools/venny/

  No coding required, just copy/paste lists into the online tool.