Colleen Moore & Mike Amato
University of Wisconsin - Madison

This handout covers:

- Making a simple bar graph, using the handout3 data set
- Adding standard error lines to the bars
- Other options for snazzing up your graph

Data: 2x3 design

We'll use a sample data set from an experiment that investigated the effects of prenatal alcohol exposure on the brains of monkeys.

The experiment looked at brain activation using a 2(group: alcohol, no-alcohol) x 3(brain area: medial pfc, orbitalfrontal cortex, medial frontal gyrus) design. Means and standard errors are shown at right.

mean activation level			
	med_pfc	orbfr_ctx	medfr_gy
alcohol-exposed	1.0805	0.8692	1.3511
no alcohol	1.0918	.9039	1.3667

standard errors			
	med_pfc	orbfr_ctx	medfr_gy
alcohol-exposed	0.0131	0.0078	0.015
no alcohol	0.0138	0.0094	0.0115

1. Create matrices of your means and standard errors

First we'll create a separate vector (i.e. list) of the brain area means for both levels of the alcohol factor. Here we will just type them in, but with a little imagination you could figure out how to calculate them directly from your data. See the "one-way ANOVA" handout for more.

```
> alc.means=c(1.0805, .8692,1.3511)      # c( x, y, z ...) creates a vector of values x, y, and z
> alc.means                              # alc.means is just a variable name
[1] 1.0805  0.8692  1.3511
> noalc.means = c(1.0918, .9039, 1.3667)
> noalc.means
[1] 1.0918  0.9039  1.3667
```


See the "one-way ANOVA" handout for more on entering your data, and using the very handy *tapply( )* function for calculating group means.

Next we'll put the two vectors together and form a matrix that will look like the table at the top of this page (which is itself a matrix). Just like that table, our new matrix will have the levels of "brain area" along one dimension, and the levels of "alcohol exposure" along another dimension.

```
> data1.means = rbind(alc.means, noalc.means)
> data1.means
                            [,1]       [,2]
alc.means     1.0805  0.8692  1.3511
noalc.means 1.0918  0.9039  1.3667
```

Instead of using *rbind* to put them together as rows, we could just as easily have used *cbind* to put them together as columns.

Now we need a similar matrix of the standard errors. We could do it the same way. But instead we'll use a shortcut- instead of making two vectors and then binding them together, we'll just make a matrix in a single step.

```
> data1.se = matrix( c(0.0131, .0138, .0078, .0094, .015, .0115),   2,  3)
> data1.se                              # makes a vector from the values given
     [,1]     [,2]    [,3]             # from that vector, create a matrix with 2 rows and 3 columns
[1,] 0.0131  0.0078  0.0150
[2,] 0.0138  0.0094  0.0115
```

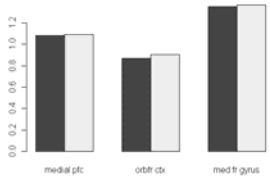
Let's give names to the rows and columns in that matrix we just made, and to the columns in data1.means (data1.means already has names for its rows).

```
> rownames(data1.se)=c("Alcohol-exposed", "No Alcohol")
> colnames(data1.se)=c("medial pfc", "orbfr ctx", "med fr gyrus")
> data1.se
                  medial pfc   orbfr ctx    med fr gyrus
Alcohol-exposed      0.0131       0.0078       0.0150
No Alcohol           0.0138       0.0094       0.0115
> colnames(data1.means)=c("medial pfc", "orbfr ctx", "med fr gyrus")
```

2. Drawing the Graph

Ok, now we're ready to make a barplot! Go ahead and enter the following command:

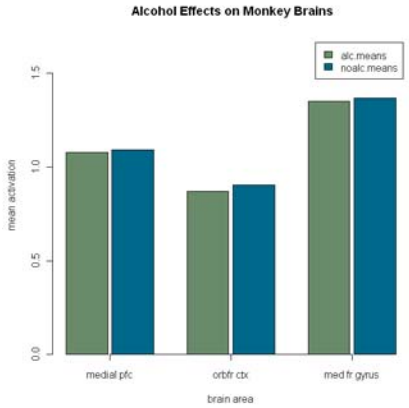
```
> alcbars = barplot(data1.means, beside=T)
```



R should show you a bar graph like this one.

As you can see, the graph is pretty bare. Fortunately there are many **optional** parameters we can set when calling *barplot( )*. We get a much nicer looking graph with the following (all one long command):

```
>alcbars = barplot(data1.means, beside=T, ylim=c(0,1.7), space=c(.1,.8), main=" Alcohol Effects on Monkey
  Brains", xlab="brain area", ylab="mean activation", legend =T, axis.lty=1,
  col=c("darkseagreen4","deepskyblue4"))
```



beside = T	groups the bars next to each other
ylim = c(0,1.7)	specifies the y-axis min and max
space = c(.1,.8)	space between bars; as proportion of bar width *1st # is space between bars of same group* *2nd# is space between bars of different groups*
main	main title for the graph
xlab	label for the x-axis
ylab	label for the y-axis
legend = T	display a legend
axis.lty=1	draws a line for the x-axis *line type* *letter "L" after the dot* *number "one" after equal sign*
col=	colors for the bars

3. Colors

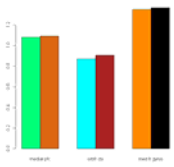
R has 657 colors to choose from, ranging from dull *saddlebrown* to provocative *hotpink4*. You can read a list of all the colors R can display by typing

```
> colors( )
```

or see them on this very helpful chart: http://research.stowers-institute.org/efg/R/Color/Chart/ColorChart.pdf

You're also able to specify each bar's color individually.

```
> monkeycolors = c("springgreen", "chocolate", "cyan", "brown",   "darkorange", "gray6")
> alcbars = barplot(data1.means, beside=T, col=monkeycolors)
```

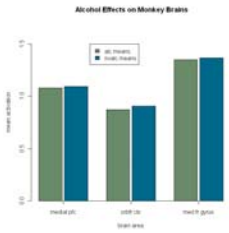


Finding a reason why you would want to do this up to you.

4. Legend

If you're unhappy with where R places the legend, you can set "legend=F" when you draw the graph and assert manual control over it:

```
> legend(x=3,y=1.5, legend=rownames(data1.means), fill=c("darkseagreen4","deepskyblue4"))
```



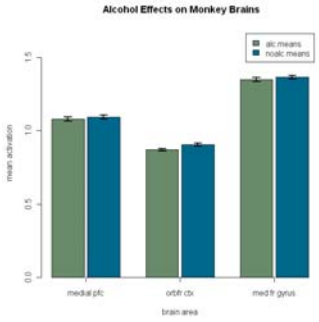
x = 3	x position for left side of legend *for categoric axis, units are tickmarks*
y = 1.5	y position for top of legend
legend =	text for the legend *could also use c("text1", "text2", etc.)*
fill =	colors to go next to the text *make sure these are correct!!!*

You can read more about formatting legends here:
http://stat.ethz.ch/R-manual/R-patched/library/graphics/html/legend.html

5. Error Bars

Of course no bar graph is complete without error bars. Using R's built in function *errbar( )* is a little tricky; it's actually easier to define our own function *superpose.eb( )* to do it.

```
> superpose.eb = function (x, y, ebl, ebu = ebl, length = 0.08, ...) arrows(x, y + ebu, x, y - ebl, angle = 90, code
   = 3, length = length, ...)          # don't worry about understanding exactly what this command is doing
> graph.abscissa = alcbars             # this creates a new matrix of the x-values of the bars in graph alcbars"
> superpose.eb(x=graph.abscissa, y=data1.means, ebl=data1.se, col="black", lwd=2)
```



x =	x coordinates for the error bars
y =	y coordinates for the **center** of the error bars
ebl =	length of the error bars
col =	color for the bars
lwd =	thickness of the error bar lines *line width*

The superpose.eb function and other useful R things can be found on the website of Raoul Grasman:
http://users.fmg.uva.nl/rgrasman/rpages/2005/09/error-bars-in-plots.html

6. Drawing a box around your graph

Are you someone who likes things neatly separated? If so, you might consider drawing a box around your graph.

```
> par( mar = c(1,1,1,1) )          # set margins for bottom, left, top, and right to be 1 unit in from each
                                   # side of the window
> box( )                           # draw a box
```