

# R Commander HowTo

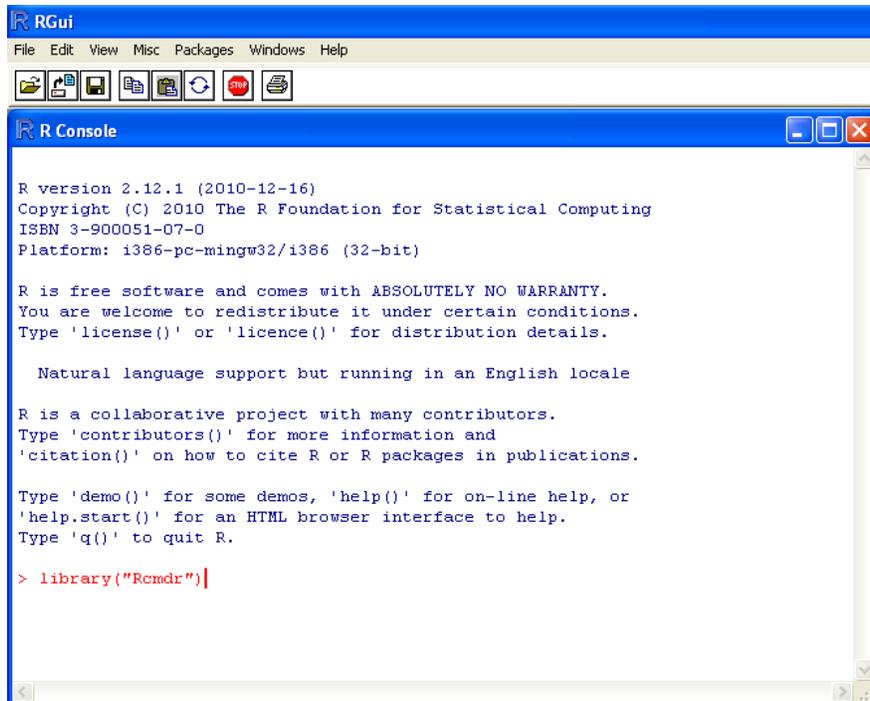
David McCallum

## O. Installation (Windows)

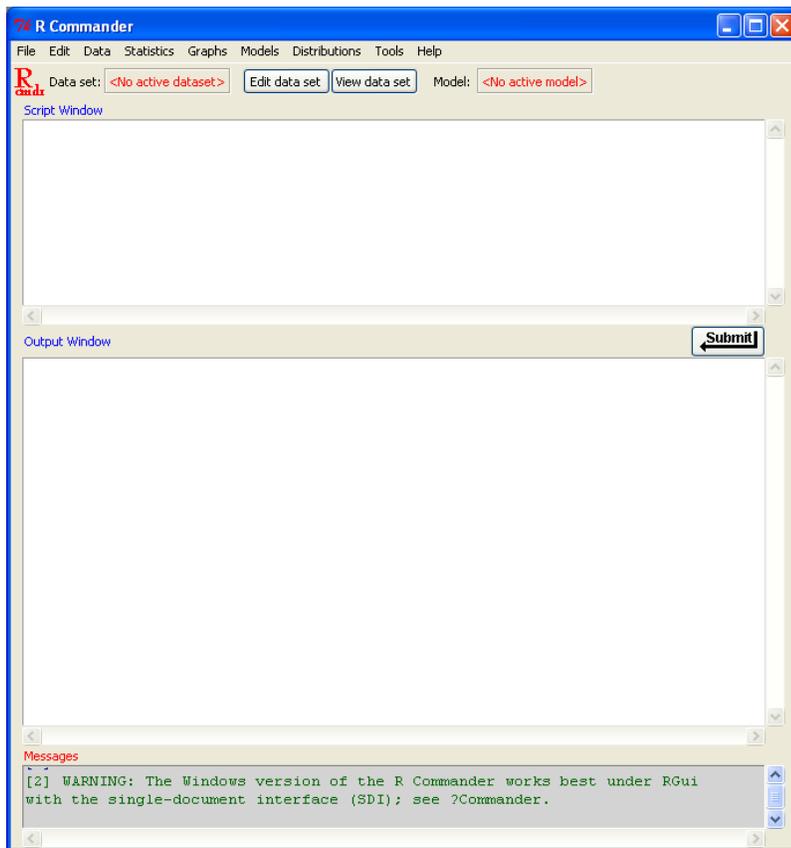
- 1) To install R, go to <http://www.r-project.org> and select a download mirror
- 2) Choose the Windows binary distribution
- 3) Navigate to the 'base' directory
- 4) Download R and run the installation wizard
- 5) To run R, go to Start->Programs->R->R 2.12.1
- 6) To install R Commander, type the following into the R Console:  
`install.packages("Rcmdr", dependencies=TRUE)`

-This will install many files from the R website's 'contrib' directory

- 7) To run R Commander, type the following into the R Console:  
`library("Rcmdr")`



# I. R Commander Interface



The script window shows R commands that were generated by the GUI.

- You can also enter your own commands here,
- To enter a command, place the editing 'I-beam' on the line, and click the Submit button,

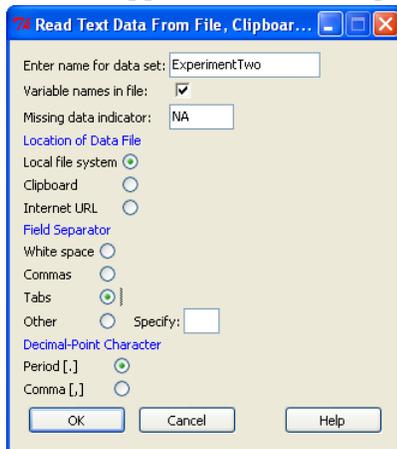
The output window shows command that was executed, and its results.

The messages window displays errors or notes from R Commander.

## II. Import the data into R Commander

### 1) Select Data->Import Data

-supports text files, clipboard data and SPSS files



### 2) Take a look at the data

-Click on the View data set button. (Note that dialogs may appear inside of the R GUI window)

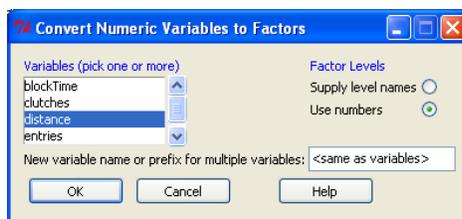
-Make sure the data imported properly

	partNumber	technique	taskType	practice	expTime	blockTime	taskNum	blockNum	trialN
1	1	6	0	1	3045229	50500	0	0	
2	1	6	0	1	3045229	50500	0	0	
3	1	6	0	1	3045229	50500	0	0	
4	1	6	0	1	3045229	50500	0	0	
5	1	6	0	1	3045229	50500	0	0	
6	1	6	0	1	3045229	50500	0	0	
7	1	6	0	1	3045229	50500	0	0	
8	1	6	0	1	3045229	50500	0	0	
9	1	6	0	1	3045229	50500	0	0	
10	1	6	0	1	3045229	50500	0	0	
11	1	6	0	1	3045229	50500	0	0	
12	1	6	0	1	3045229	50500	0	0	

### 3) Convert numeric variables to factors

-Data->Manage variables in the active data set->Convert numeric variables to factors...

-Supply level names if appropriate, eg if 3 equals “My Technique”



### 4) Specify any non-factor, non-numeric data

-Click on the Edit data set button

-Click on the variable name

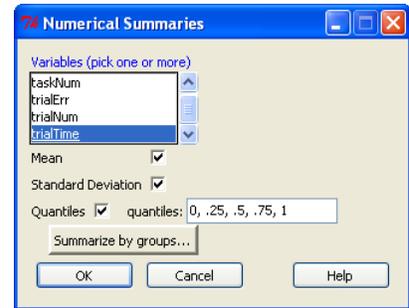
-Select the data type, either Real (default), or character

### 5) Save the data set

-Data->Active data set->Save active data set...

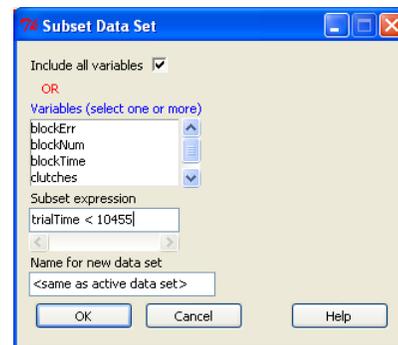
### III. Identify Outliers

- 1) Statistics->Summaries->Numerical Summaries...
- 2) Read the standard deviation and mean from the output window
- 3) Calculate a threshold 2 standard deviations from the mean  
-may need to do this separately for each technique and task
- 4) Remove trials beyond this threshold as in section IV.



### IV. Remove practice, outliers or other rows

- 1) Data->Active Data Set->Subset Active Data Set...
- 2) Remove columns if desired, by selecting the variables to keep
- 3) Modify the Subset expression to remove rows:



==	equal
!=	not equal
&	AND
	OR
is.na(varname)	Include the missing values of a variable
!is.na(varname)	Exclude the missing values of a variable
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

- If you use a name in an expression you should surround it with double quotes e.g. "name".
- Variable names are case-sensitive.

eg:    practice == 0  
       techniqueName != "LensMouse"

## V. Generate Graphs

1) Graphs->Plot of means...

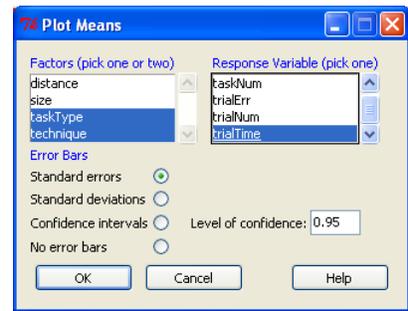
2) If the wrong variable was chosen for an axis:

a) Check the script window. You should see a line like this:

```
plotMeans(OneDTapping$trialTime,  
OneDTapping$technique, OneDTapping$size,  
error.bars="se")
```

b) Rearrange the variables appropriately.

c) Click the submit button while the I-beam is on that line. You may need to remove line breaks.



## VI. Perform an ANOVA

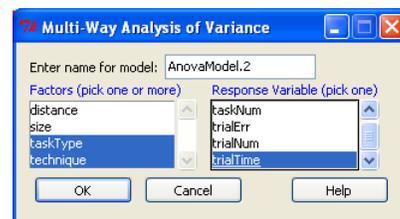
1) Choose one of the following:

Statistics->Means->One-way ANOVA

Statistics->Means->Multi-way ANOVA

2) Check for residuals and normality

Models->Graphs->Basic diagnostic plots



## VII. Post-hoc Analysis

This example uses Tukey's Honest Significance Test (TukeyHSD). TukeyHSD isn't available in R Commander, and the commands must be entered manually into the script window. It tells us which means are statistically significant, and assumes normal data where the observations are independent.

### TukeyHSD Syntax

```
TukeyHSD(x, which, ordered = FALSE, conf.level = 0.95)
```

- **x** A fitted model object, usually an aov fit.
- **which** A character vector listing terms in the fitted model for which the intervals should be calculated. Defaults to all the terms.
- **ordered** A logical value indicating if the levels of the factor should be ordered according to increasing average in the sample before taking differences. If ordered is true then the calculated differences in the means will all be positive. The significant differences will be those for which the lwr end point is positive.
- **conf.level** A numeric value between zero and one giving the family-wise confidence level to use.

## TukeyHSD Example

```
summary(fm1 <- aov( trialTime ~ distance*size*technique, data = TwoDTapping_NoOutliers))  
TukeyHSD(fm1, "technique", ordered = FALSE)  
plot(TukeyHSD(fm1, "technique"))
```

To enter each line, place the I-beam on the line and click the 'submit' button.

Previously, we had performed an ANOVA, and its command can still be found in the script window. We can copy the parameters directly from the ANOVA command, and paste them into the aov command above.

A sample ANOVA command, generated by Statistics->Means->Multi-way ANOVA:

```
> AnovaModel.2 <- (lm(trialTime ~ distance*size*technique,  
+ data=TwoDTapping_NoOutliers))
```