Introduction to R and RStudio

Part 5: Introduction to Graphics in R

Rob Cribbie
Department of Psychology
York University

http://www.psych.yorku.ca/cribbie/r_course_trent.html
Graphics are an extremely important part of data analysis
  ◦ However, difficulties producing appropriate or required graphics means that many researchers do not take the time to “visualize” their data
R has excellent graphical capabilities
  ◦ For those who want more on graphics I recommend:
    • Paul Murrel’s book “R Graphics”
    • Hadley Wickham’s book “ggplot2
    • Googling graphics in R
A researcher is interested in evaluating two therapies for perfectionism; specifically investigating whether they will be effective in reducing levels of perfectionism.

- Levels of perfectionism are recorded at baseline, 1 month (mid intervention) and 2 months (post intervention) for each experimental group (CBT, General Stress) and a control group.

The researcher also records depression at baseline, as well as the sex of the subject.
The following commands were used to generate the data set:

```r
sex <- rep(c("m", "f", "m", "f"), c(17, 30, 18, 25))
group <- rep(c("cbt", "stress", "control"), c(30, 30, 30))
perf1 <- rnorm(90, mean = 40, sd = 15)
dep1 <- perf1 + rnorm(90, sd = 30)
perf2 <- perf1 - 7 + rnorm(90, sd = 10)
perf3 <- perf2 - 3 + rnorm(90, sd = 10)
perf2[group == "control"] <- perf2[group == "control"] + 6
perf3[group == "control"] <- perf3[group == "control"] + 3
newdat <- data.frame(sex, group, dep1, perf1, perf2, perf3)
```
Simple Histogram

- We can produce a simple histogram with the command `hist()`
- For example, we might want to look at the frequencies for scores on depression
  - `hist(newdat$dep1)`

![Histogram of dep1](image)
Customizing a Histogram

```r
> hist(newdat$perf1, main="Baseline Perfectionism", xlab="Baseline Perfectionism", col="blue", xlim=c(0,120), ylim=c(0,40))
> rect(70,0,100,30, border="red", lwd=4)
> text(85,35, "At Risk Levels", col="red")
```
Multiple Plots

> par(mfrow=c(2,2))
> hist(newdat$dep1)
> hist(newdat$perf1)
> hist(newdat$perf2)
> hist(newdat$perf3)

Two rows and two columns for the graphical space
Simple Boxplot

> boxplot(newdat$perf2)
Fancier Boxplot

> boxplot(perf2~group*sex, col="purple", xlab="Group", ylab="perf2", data=newdat)
Obtaining information about outliers with the ‘boxplot’ function

- Using the values ‘out’ and ‘group’ we can print the cases that are extreme
  - > boxplot(perf1~sex, data=newdat)$out
    - [1] 81.441446 2.163886 2.993359
  - > boxplot(perf1~sex, data=newdat)$group
    - [1] 1 2 2

We are again utilizing information (values) available via the functions
Obtaining information about outliers with the ‘boxplot’ function

- The ‘out’ and ‘group’ options can also be used with more complex boxplots

```r
> boxplot(perf2~group*sex,col="purple", xlab="Group",ylab="perf2",data=newdat)$out
  [1] -9.862943 2.022603
> boxplot(perf2~group*sex,col="purple", xlab="Group",ylab="perf2",data=newdat)$group
  [1] 2 5
```

- The “2” and “5” represent cases in the combination of group and sex (e.g., ‘male, stress’, ‘female, stress’)
Using the ‘plot’ tab in the extras panel, you can easily save your plot in many different formats, or copy it to the clipboard.
Creating a ‘pdf’ of a Plot with Text

```r
> pdf("boxplot.pdf")
> boxplot(perf2~group*sex, col="purple", xlab="Group", ylab="perf2", data=newdat)
> dev.off()
```

- Creating a new pdf called boxplotb
  ```r
  > pdf("boxplotb.pdf")
  > par(mfrow=c(2,2))
  > hist(newdat$perf1, col="blue")
  > hist(newdat$perf2, col="red")
  > hist(newdat$perf3, col="pink")
  > hist(newdat$dep1, col="green")
  > dev.off()
  ```
boxplot.pdf
plot()

- plot() is the main graphing function in R
- Automatically produces simple plots for vectors, data frames, relationships, etc.
- It includes several options for customization
- For example, what if we plot() our entire data set:
  - plot(newdata)
Plotting a Subset of the Variables in the Dataset

```
plot(newdat[-c(1,2,3)], main="Relationships among Perfectionism Variables")
```

Remove the first three variables from the dataset
Simple Bivariate Plots

plot(newdat$perf1, newdat$perf2, main="Relationship between Baseline and Time 2 Perfectionism")
The main plotting characters are represented by the command “pch” and range from 1:25.
Simple Bivariate Plot with Customized Plotting Character

```
plot(perf1,perf2, main="Relationship between Baseline and Time 2 Perfectionism",pch=8, col="green", data=newdat)
```
Adding “rug” plots

- `rug(newdat$perf1, side=1)`
- `rug(newdat$perf2, side=2)`
Adding a Regression Line to a Plot

- `mod1 <- lm(perf3 ~ perf1)`
- `plot(perf1, perf3)`
- `abline(mod1)`
Diagnostic Plots for a Model

- `par(mfrow=c(2,2))`
- `plot(mod1)`
Scatterplot from the ‘car’ Package

scatterplot(perf3~perf1*sex, main="Scatterplot with Separate Regression Lines for Groups")
Added Variable Plots (i.e., partial regression coefficient plots)

- `par(mfrow=c(2,1))`
- `plot(perf1,perf3)`
- `mod2 <- lm(perf3 ~ perf1 + dep1)`
- `av.plots(mod2)`
Lattice Graphics

- “lattice” is an add-on R package
- lattice provides a high-level system for statistical graphics that is independent of traditional R graphics
- It is modeled on the Trellis suite in S-PLUS, and implements most of its features
lattice uses the ‘grid’ package (Murrell, 2005) as the underlying implementation engine, and thus inherits many of its features by default

The nature of the graphics depends on the type of display, but common options are:

- primary variables: represent the primary display
- conditioning variables: divides data into subgroups, each of which are presented in a different panel
- grouping variables: subgroups are contrasted within panels by superimposing the corresponding displays
Lattice Graphics Examples

`histogram(~ perf1 | group)`
densityplot(~ perf1 | group, groups=sex, layout=c(1,3), data=newdat)
longdata1 <- reshape(newdat, varying = list(4:6), direction = "long", v.names = c("perfect"))
xyplot(perfect ~ time, groups = id, type = "l", data = subset(longdata1, id < 50))
Lattice Graphics Examples

xyplot(perf3~perf1, groups = sex, type = c("r", "p", "smooth"))
Lattice Graphics Examples

xyplot(perf3~perf1|group,type=c("r","p","smooth"),layout=c(1,3))
Lattice Graphics Examples

```r
xyplot(perf3~dep1|group+sex,type=c("r","p","smooth"),layout=c(2,3))
```
Lattice Graphics Examples

\[
\text{cloud(} \text{perf3} \sim \text{perf1} * \text{dep1} \mid \text{sex}, \text{main="3-dimensional Scatterplot"}, \text{data=newdat})
\]
gg stands for the ‘Grammar of Graphics’

- This also implies that there is a grammar to the composition of statistical plots
- By controlling that grammar you can control the nature of your plots
  - In fact, control a wide variety of plots with a fair amount of precision and options
- For example, you can control the type of plot, color of elements, shape of elements, position of elements, axis characteristics, labels, titles, grouping factors, etc.
Getting Started with ggplot2

There is a function within the ggplot2 package called ‘qplot’ that does quick plots of various types of graphs
  ◦ However, we will learn ggplot2 by using the more customizable ‘ggplot’ function

The ggplot function takes two primary arguments:
  ◦ data
    • The data frame containing the data
  ◦ aes
    • The aesthetics (i.e., variables, plot options, etc.)
Example:
- `> plot1 <- ggplot(newdat, aes(dep1))`
- At this point you have not added any layers so you can’t generate a plot
- However:
  - `> plot1 + geom_histogram()`
Geometric Objects (geoms)

- Geoms are the shapes that are added to the plot layer(s)
  - For example:
    - geom_histogram()
    - geom_boxplot()
    - geom_bar()
    - geom_point()
    - geom_errorbar()

- Since each geom contains () at the end, each geom can also accept new aesthetic statements (e.g., fill, colour)
Adding and Improving Layers

- `plot1 <- ggplot(newdat, aes(dep1))`
- `plot1 + geom_bar(colour='black', fill='red') + ylab("Frequency") + xlab("Depression") + labs(title="Histogram of Depression")`
What About Multiple Variables

- `ggplot(newdat,aes (dep1,perf1)) + geom_point + labs(x="Depression", y="Perfectionism", title="Relationship b/w Perfectionism and Depression")`
Adding options to a scatterplot

- ggplot(newdat,aes(dep1,perf1,colour=group)) + geom_point() + stat_smooth(method=lm)) + labs(x="Depression", y = "Perfectionism", title ="Relationship b/w Perfectionism and Depression")
Boxplots

- `ggplot(newdat,aes(group,perf1))+geom_boxplot(fill='blue')`
  - But what about the small x-axis tick labels
Boxplots

- `ggplot(newdat,aes(group,perf1)) + geom_boxplot(fill='blue') + theme(axis.text.x=element_text(size=20))`

This is just one of numerous ways of controlling the display of the plot.
Adding Error Bars to a Plot of Group Means

- `ggplot (newdat,aes(group,dep1)) + stat_summary (fun.y=mean, geom='bar', fill="red") + stat_summary (fun.data =mean_cl_boot, geom="errorbar") + ylab ("Mean Depression") + xlab ("Treatment Group") + coord_cartesian (ylim=c(30,50))`
Adding Error Bars to a Plot of Group Means
What if we have more than one grouping variable?

- `ggplot(newd,aes(group,dep1,fill=sex)) + stat_summary(fun.y=mean, geom='bar',position= "dodge") + stat_summary(fun.data= mean_cl_boot, geom="errorbar", position= position_dodge()) + ylab("Mean Depression") + xlab("Treatment Group") + coord_cartesian(ylim=c(20,60))`
What if we have more than one grouping variable?