4. Adding Features to Plots

Ken Rice
Timothy Thornton

University of Washington

Seattle, July 2015
In this session

R has very flexible built-in graphing capabilities to add a wide-range of features to a plot.

- Plotting options
- Adding points, lines, and segments to existing plots
- Creating a legend for a plot
Scatterplot Options

The command \texttt{plot(x,y)} will create a scatterplot when \(x\) and \(y\) are numeric. The default setting will plot points but one can graph lines or both (or neither):

- \texttt{plot(x,y,type=\textquote{p})} is the default option that plots points
- \texttt{plot(x,y,type=\textquote{l})} connects points by lines but does not plot point symbols
- \texttt{plot(x,y,type=\textquote{b})} plots point symbols connected by lines
- \texttt{plot(x,y,type=\textquote{o})} plots point symbols connected by lines, points on top of lines
- \texttt{plot(x,y,type=\textquote{h})} will plot histogram-like (a.k.a. high-density) vertical lines
- \texttt{plot(x,y,type=\textquote{n})} plots axes only, no symbols
Examples: Plotting two variables

Let's consider the `airquality` dataset.

data(airquality)
names(airquality)
airquality$date <- with(airquality, ISOdate(1973, Month, Day))

(ISOdate() takes year/month/day information and returns an object containing the same information, but in a format R recognizes as numeric information.)
Examples: Plotting two variables

```r
plot(Ozone~date, data=airquality)
```
Examples: Plotting two variables

```r
plot(Ozone~date, data=airquality,type="l")
```
Examples: Plotting two variables

```r
plot(Ozone~date, data=airquality,type="h")
```
Adding points to a graph

We can add points to an existing plot with the command `points(x,y)`

The `lines(x,y)` command can be used to add connected points by lines to an existing plot without symbols
Adding points to a graph

For example, create a graph that contains axes only.

```r
plot(Ozone~Solar.R, data=airquality,type="n")
```
Adding points to a graph

Now add the points to the graph:

```r
points(airquality$Solar.R, airquality$Ozone, col="blue", pch=7)
```
Adding lines to plots

Horizontal, vertical, and sloped lines can be added to an existing plot with `abline()`:

- `abline(h=ycoordinate)` adds a horizontal line at the specified y-coordinate
- `abline(v=xcoordinate)` adds a vertical line at the specified x-coordinate
- `abline(intercept,slope)` adds a line with the specified intercept and slope

As well as using `lines()`, line segments can also be added to an existing plot with `segments()`:

- `segments(x0,y0,x1,y1)` adds a line segment from (x0,y0) to (x1,y1)
Adding lines to plots

bad <- ifelse(airquality$Ozone>=90, "orange","forestgreen")
plot(Ozone~date,data=airquality,type="h",col=bad)
abline(h=90,lty=2,col="red")
**Adding text to plots**

Text labels can be added to a plot with the `text()` command:

- `text(x,y,"Here is my text")` adds text centered at the specified \((x,y)\) coordinates

Text colors and size can be specified with the options `col` and `cex`, respectively.
Adding text to plots

```r
bad <- ifelse(airquality$Ozone>=90, "orange", "forestgreen")
plot(Ozone~Solar.R, data=airquality, col=bad)
abline(h=90, lty=2, col="red")
text(85,100,"High Ozone Level",cex=.8,col="blue")
```
Adding a legend to a plot

Including a legend is often essential for explaining symbols, colors, or line types used in a plot. The `legend()` command can be used to add a legend to an existing plot:

- The position of the legend can be specified by \((x,y)\) coordinates or by using preset positions:
  
  - `legend(x,y,c("name1","name2"), pch=c(1,5))` adds a legend to the plot with its top-left corner at coordinate \((x,y)\)
  
  - `legend("topright",c("name1","name2"),pch=c(1,5))` adds a legend in the top right corner of the plot. Can also use ”bottom”, ”bottomleft”, ”left”, ”topleft”, ”top”, ”topright”, ”right” and ”center”.

4.14
Adding a legend to a plot

Options such as symbols (\texttt{pch}), colors (\texttt{col}), and line types (\texttt{lt}) can be specified in the legend command. See \texttt{?legend} for more details.

\begin{verbatim}
lowwinds <- ifelse(airquality$Wind<=8, "red", "blue")
symbols <- ifelse(airquality$Wind<=8, 5,1)
plot(Ozone~Solar.R,data=airquality,col=lowwinds,pch=symbols)
legend("topleft",c("low wind","high wind"),col=c("red","blue"),
pch=c(5,1))
\end{verbatim}
Adding a legend to a plot

![Plot with legend](image)

**Legend:**
- Red diamonds: low wind
- Blue circles: high wind
Smoothing

A straight line may not adequately represent the relationship between two variables.

Smoothing is a way of illustrating the local relationship between two variables over parts of their ranges, which may differ from their global relationship.

Locally weighted scatterplot smoothing (LOWESS) can be performed in R with the `lowess()` function, which calculates a smooth curve that fits the relationship between $y$ and $x$ locally.

The `supsmu()` function can also be used for smoothing.

The output from both smoothing functions have attributes $x$ and $y$ that can be used with the generic plotting function `lines()`.
Smoothing

Consider the built-in dataset `cars`.

data(cars)
plot(dist~speed, data=cars)
with(cars, lines(lowess(speed, dist), col="tomato", lwd=2))
Smoothing

plot(dist~speed, data=cars, log="xy")
with(cars, lines(lowess(speed, dist), col="tomato", lwd=2))
with(cars, lines(supsmu(speed, dist), col="purple", lwd=2))
Smoothing

legend("bottomright", legend=c("lowess","supersmoother"), bty="n", lwd=2, col=c("tomato","purple"))
Multiple plots in a single figure

The `par()` and `layout()` functions can be used for drawing several plots in one figure.

`par()` with the option `mfrow=c(nrows,ncols)` creates a matrix of $nrows \times ncols$ plots that are filled in by row.

Using `par( mfcol=c(nrows,ncols) )` fills in the matrix by columns instead.

`layout(mat)` allows for a more customized panel with multiple plots, where `mat` is a matrix object that specifies the locations of the plots in the figure.
Multiple plots in a single figure

The ToothGrowth dataset, supplied with R, contains data from a study on the effect of vitamin C on tooth growth in 10 guinea pigs.

- There are two treatments/supplement types: orange juice and ascorbic acid
- There are three vitamin C dose levels for each of the two treatments: 0.5, 1, and 2mg
- The response is length of odontoblast;

![Diagram of ToothGrowth dataset](image-url)
Multiple plots in a single figure

Commands for plotting multiple figures with the ToothGrowth dataset, using `par()`;

data(ToothGrowth)  # load data into current R session
par(mfrow=c(2,2))  # Set up a 2x2 layout

#1st Plot - scatterplot of length vs dose;
plot(len~dose, data=ToothGrowth, xlab="Vitamin C dose (mg)",
     ylab="Tooth Length", col="blue", cex.main=.8)

#2nd plot - boxplot of length vs dose;
boxplot(len~dose, data=ToothGrowth, horizontal=TRUE,
        ylab="Vitamin C dose (mg)", xlab="Tooth Length", cex.main=.8)

#3rd plot - boxplot of length vs type of supplement;
boxplot(len~ supp, data=ToothGrowth, horizontal=TRUE,
        ylab="Supplement Type", xlab="Tooth Length", cex.main=.8)

#4th plot - length vs *interaction* (i.e. all combinations) of supp and dose;
boxplot(len~ supp*dose, data=ToothGrowth, horizontal=TRUE, col=c("orange","yellow"),
        ylab="Supplement and Dose", xlab="Tooth Length")

#... and give this one a legend
legend("topleft", c("Ascorbic acid", "Orange juice"), fill=c("yellow","orange"))
Multiple plots in a single figure
Multiple plots in a single figure

Commands for a more customized multiple-plot figure using \texttt{layout()}

\begin{verbatim}
# set up a 2x2 layout, but merge first 2 cells, i.e. the top row
layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE))

# 1st plot - the interactions again, with a legend added
boxplot(len~supp*dose, data=ToothGrowth, col=c("orange","yellow"),
        xlab="Supplement and Dose",ylab="Tooth Length")

legend("bottomright",c("Ascorbic acid", "Orange juice"),
       fill = c("yellow", "orange"))

# 2nd plot (in bottom left position) - scatterplot length vs dose
plot(len~dose, data=ToothGrowth, xlab="Vitamin C dose (mg)",
     ylab="Tooth Length", col="blue", cex.main=.8)

# 3rd plot (in bottom right position) - histogram of tooth length
hist(ToothGrowth$len, xlab="Tooth Length", main="", cex.main=.8)
\end{verbatim}

(This is far too much effort for a quick look at your data – but
useful for making slides, or final copies of your paper)
Multiple plots in a single figure

![Box plots and scatter plots showing the effect of supplement and dose on tooth length.](image-url)
Summary

- R has a variety of plotting options

- `points()` adds points to an existing plot and `lines()` adds connected points by lines to an existing plot without symbols

- `abline()` draws a single straight line on a plot

- `lowess()` and `supsmu()` are scatterplot smoothers

- `legend()` adds a legend to a plot

- `par()` and `layout()` can be used for multi-panel plotting