R Session 5

Introduction to Plot Techniques

William A. Cooper

Research Aviation Facility, Earth Observing Laboratory
National Center for Atmospheric Research

Presentation on 11/12/2014

Outline

1. Packages for Plotting
2. Base graphics
   - Standard routines
   - Extension: plotWAC ( )
3. ggplot2

SOME PACKAGES PROVIDING PLOTS:

Base graphics (in package graphics, always available)

- Easy to construct basic plots; e.g.,
  - histographs, bar charts, box-and-whisker, violin, ...
  - scatterplots, caterpillar plots, density plots
  - time series, line charts, ...
- Often used for exploratory analysis

ggplot2 (“grammar of graphics”):

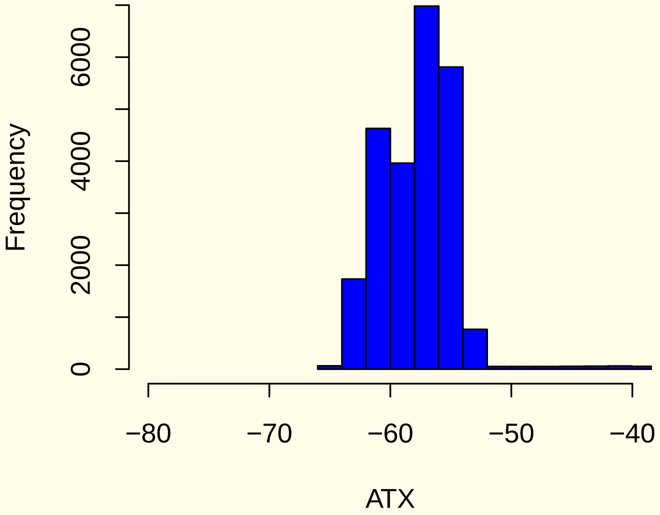
- Often used for final-presentation plots
- Great flexibility and a structured approach
- Can duplicate base-graphics plot functions

lattice graphics

implementation of “trellis” graphics – an alternate structured approach to generating plots, esp. for exploratory analysis

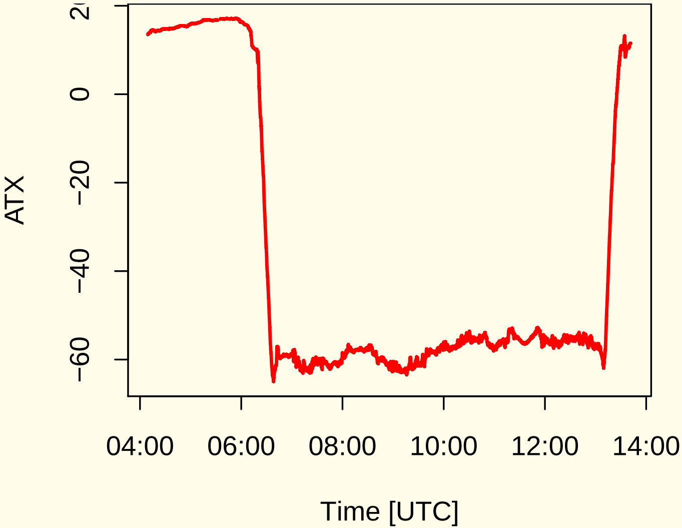
BASE GRAPHICS (console ?hist, or 'hist' in RStudio-help)

```
hist(Data$ATX[Data$TASX > 130], breaks = 40, xlab = "ATX",
    xlim = c(-80, -40), main = NULL, col = "blue")
```



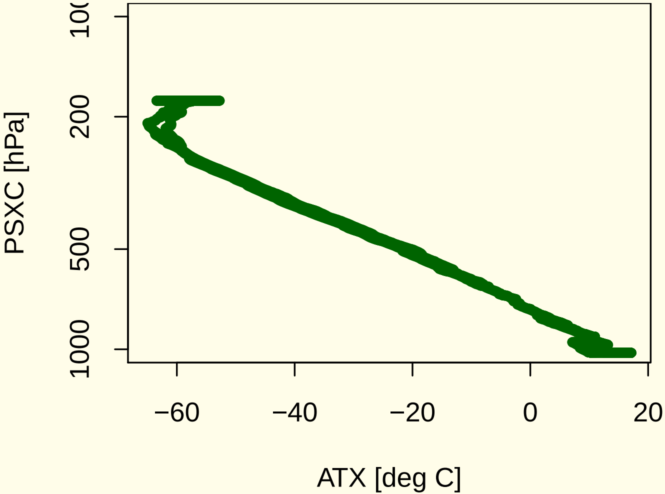
BASE GRAPHICS: line plot'

```
plot(Data$Time, Data$ATX, ylab = "ATX", col = "red",
     type = "l", xlab = "Time [UTC]", lwd = 2)
```



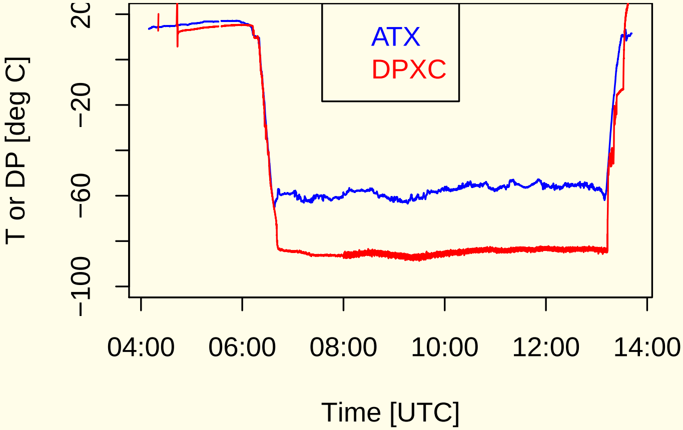
BASE GRAPHICS: scatterplot

```
plot(Data$ATX, Data$PSXC, type = "p", pch = 20, log = "y",
    xlab = "ATX [deg C]", ylab = "PSXC [hPa]", ylim = c(1000,
        100), col = "darkgreen")
```



BASE GRAPHICS: multiple lines'

```
plot(Data$Time, Data$ATX, ylab = "T or DP [deg C]", type = "l",
    col = "blue", ylim = c(-100, 20), xlab = "Time [UTC]")
lines(Data$Time, Data$DPXC, col = "red")
legend("top", legend = c("ATX", "DPXC"), text.col = c("blue",
    "red"))
```

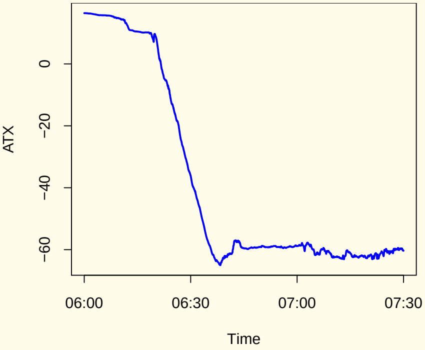


BASE GRAPHICS: adding structure with a data.frame'

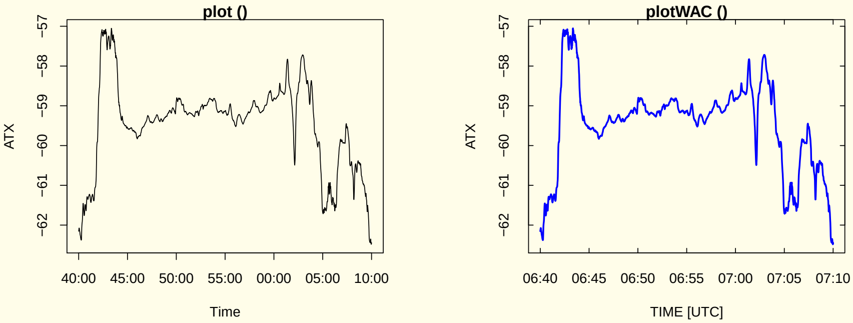
Consider using a data.frame to hold data for a plot:

- subset appropriately
- apply plot operations to this object
- optionally save for archiving
- example:

```
r <- setRange(Data$Time, 60000,
    73000)
Plot1Data <- Data[r, c("Time",
    "ATX")]
plot(Plot1Data, type = "l",
    col = "blue", lwd = 2)
```



BASE GRAPHICS: see 'plotWAC'



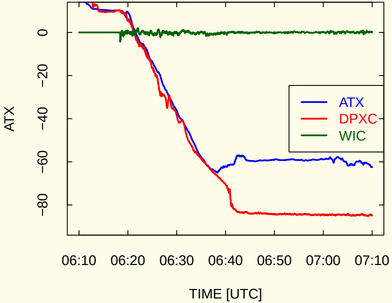
Differences:

1. Some differences in time labels
2. Default line thickness lwd=2
3. Ticks inward and duplicated on opposite axis

BASE GRAPHICS: another 'plotWAC' feature:

```r
r <- setRange(Data$Time, 61000,
    71000)
Plot1Data <- Data[r, c("Time",
    "ATX", "DPXC", "WIC")]
```

```r
plotWAC(Plot1Data, legend = "right",
    ylim = c(-90, 10))
```



REASONS TO CONSIDER ggplot:

Based on a structure called the 'Grammar of Graphics':

- independent components assembled to final plot
- layers: encourages structured composition
- particularly useful for constructing original plots with, e.g., a layer representing the result of a fit.

REASONS TO CONSIDER ggplot:

Based on a structure called the 'Grammar of Graphics':

- independent components assembled to final plot
- layers: encourages structured composition
- particularly useful for constructing original plots with, e.g., a layer representing the result of a fit.

Themes

- Can construct a theme representing the particular tailoring of the plot you favor.
- Just add the theme to the plot definition, optionally with further modifications for an individual use.

REASONS TO CONSIDER ggplot:

Based on a structure called the 'Grammar of Graphics':

- independent components assembled to final plot
- layers: encourages structured composition
- particularly useful for constructing original plots with, e.g., a layer representing the result of a fit.

Themes

- Can construct a theme representing the particular tailoring of the plot you favor.
- Just add the theme to the plot definition, optionally with further modifications for an individual use.

Supports constructing some very nice plots, although with what seems extra work at first.

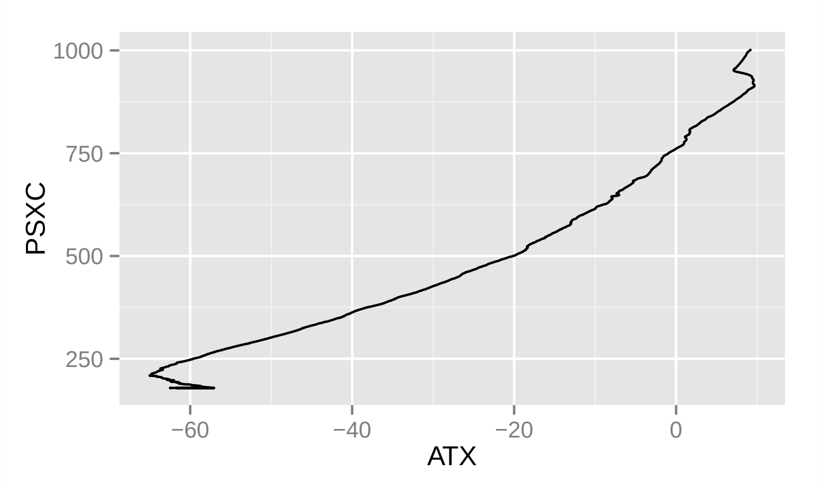
COMPONENTS OF A ggplot

Items that can be added, usually via g <- g + item

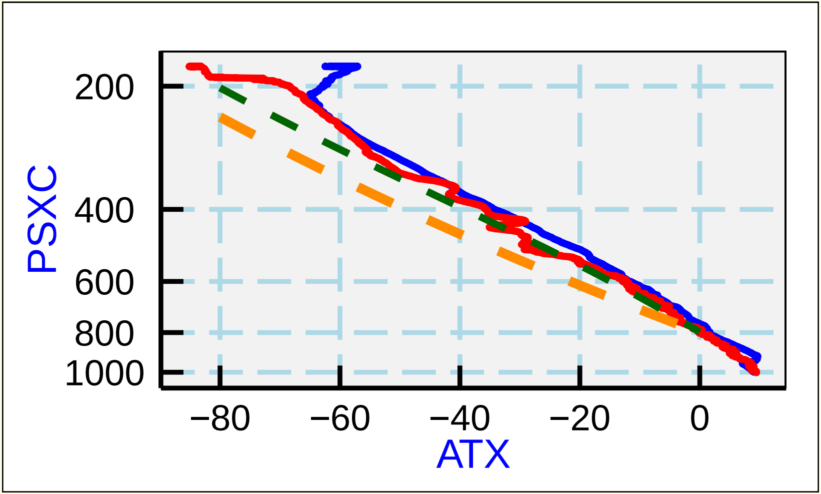
1. An initial definition for the basis of the plot, usually resembling 'g <- ggplot (data=Data, aes(x=Time, y=ATX)', containing:
   (a) The data, often best specified as a data.frame
   (b) "aesthetic mappings" – e.g., which variables are mapped to the abscissa and to the ordinate
2. 'geom's – data representations visible on the plot like lines, points, etc.
3. 'stat's – fits or creating sub-groups for further analysis as in a violin-plot
4. 'scale's – axes, colors, line-widths, symbol-types, ..., anything that helps retrieve an original datum from information on the plot.
5. 'coord's: the mapping from the data values to the plot. linear or log, e.g.; the mapping itself, vs 'scale's like axes that represent the coords with items appearing on the graph.

CONSTRUCTING A SIMPLE ggplot:

```
D <- Data[setRange(Data$Time, 61900, 71000), c("ATX", "PSXC")]
ggplot(D, aes(ATX, PSXC)) + geom_path()
```



CONSTRUCTING A TAILORED ggplot:



HOW THIS FITS IN THE "SESSIONS"

The Plan:

1. Introduction to R and esp. to RStudio
2. The data.frame and other variables
3. Basic math operations; vector operations
4. Packages, including 'Ranadu'
5. Constructing plots
6. Fit procedures; showing fits in plots and tables
7. Reproducible Analyses using R and knitr
8. Specific examples of application for RAF tasks

Next: Fitting

- linear fits: simple regression, parameterized fits, Deming fit
- non-linear fits: numerical methods
- maximum likelihood via R