R Session 5

Introduction to Plot Techniques

William A. Cooper

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

December 14, 2017

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"):

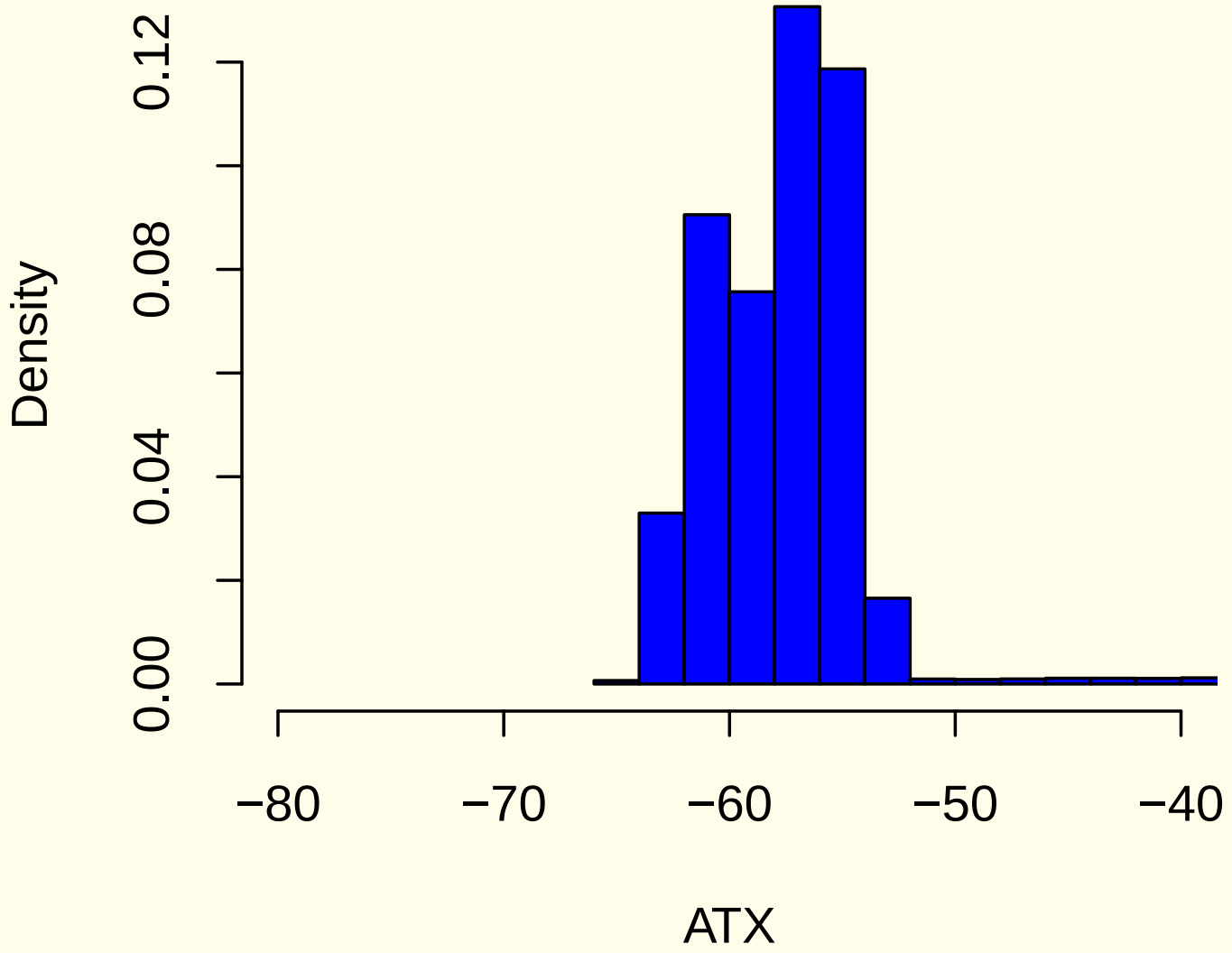
- Often used for final-presentation plots
- Great flexibility and a structured approach
- Can duplicate base-graphics plot functions

lattice graphics

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

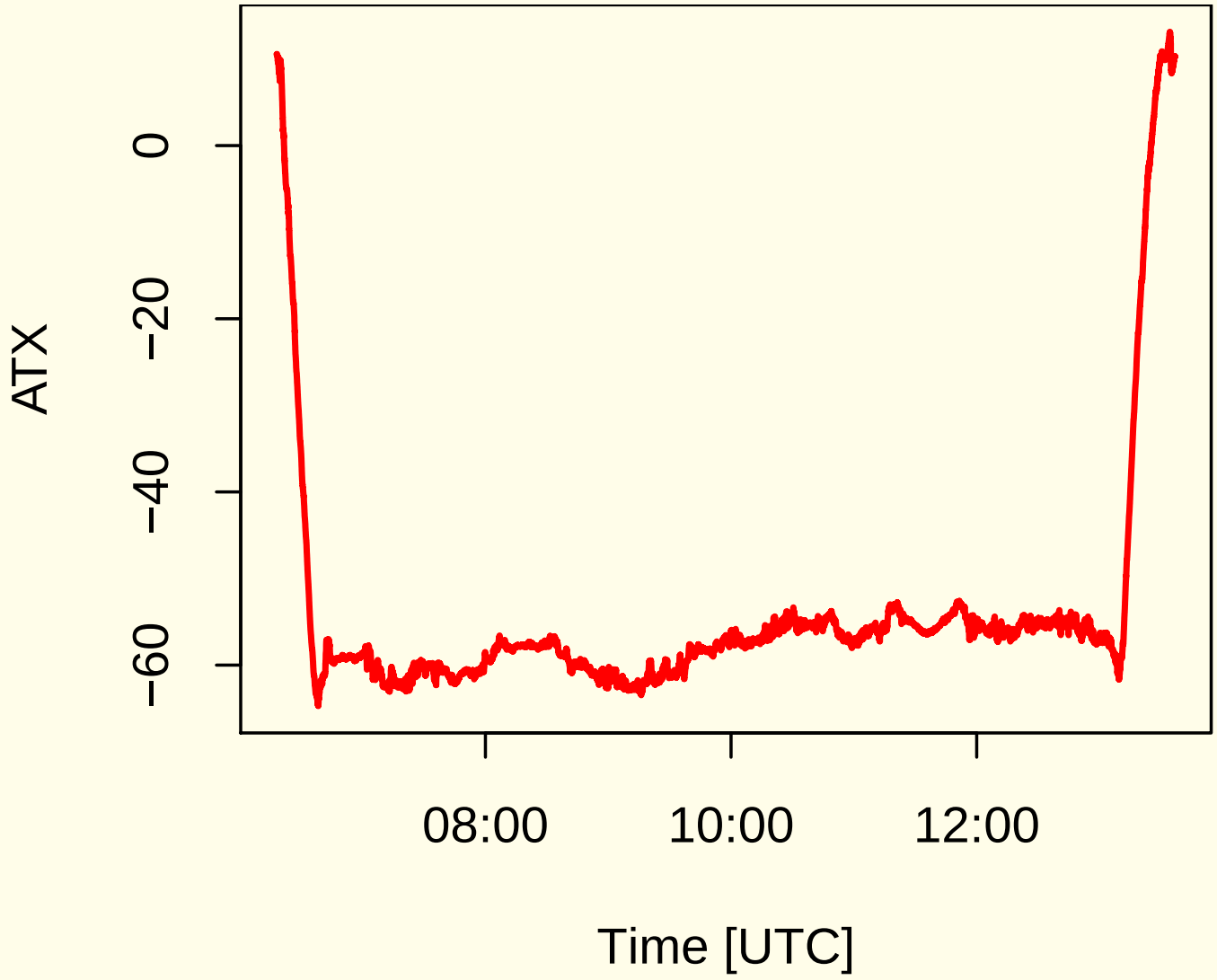
BASE GRAPHICS (console ?hist, or 'hist' in RStudio-help)

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



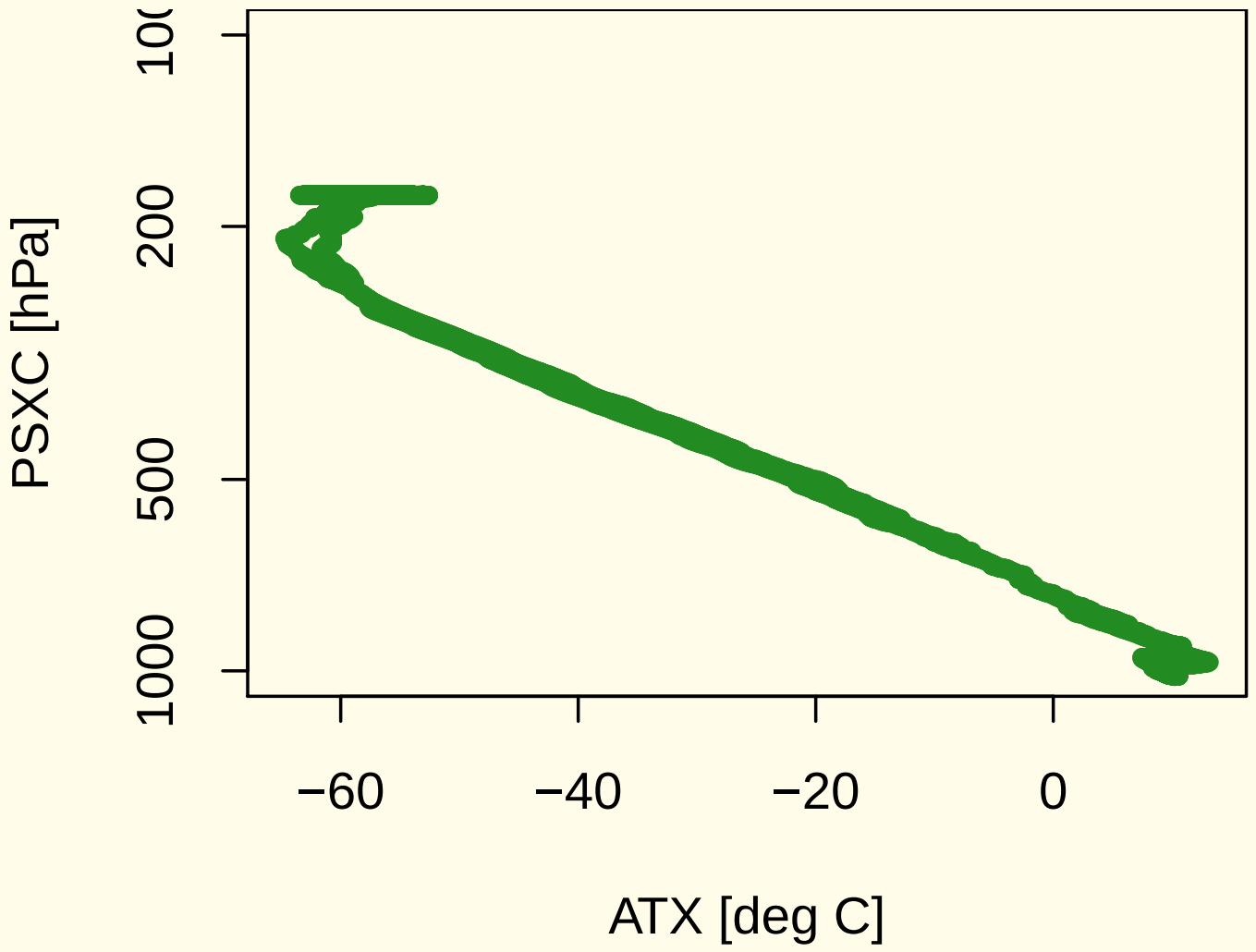
BASE GRAPHICS: line plot'

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



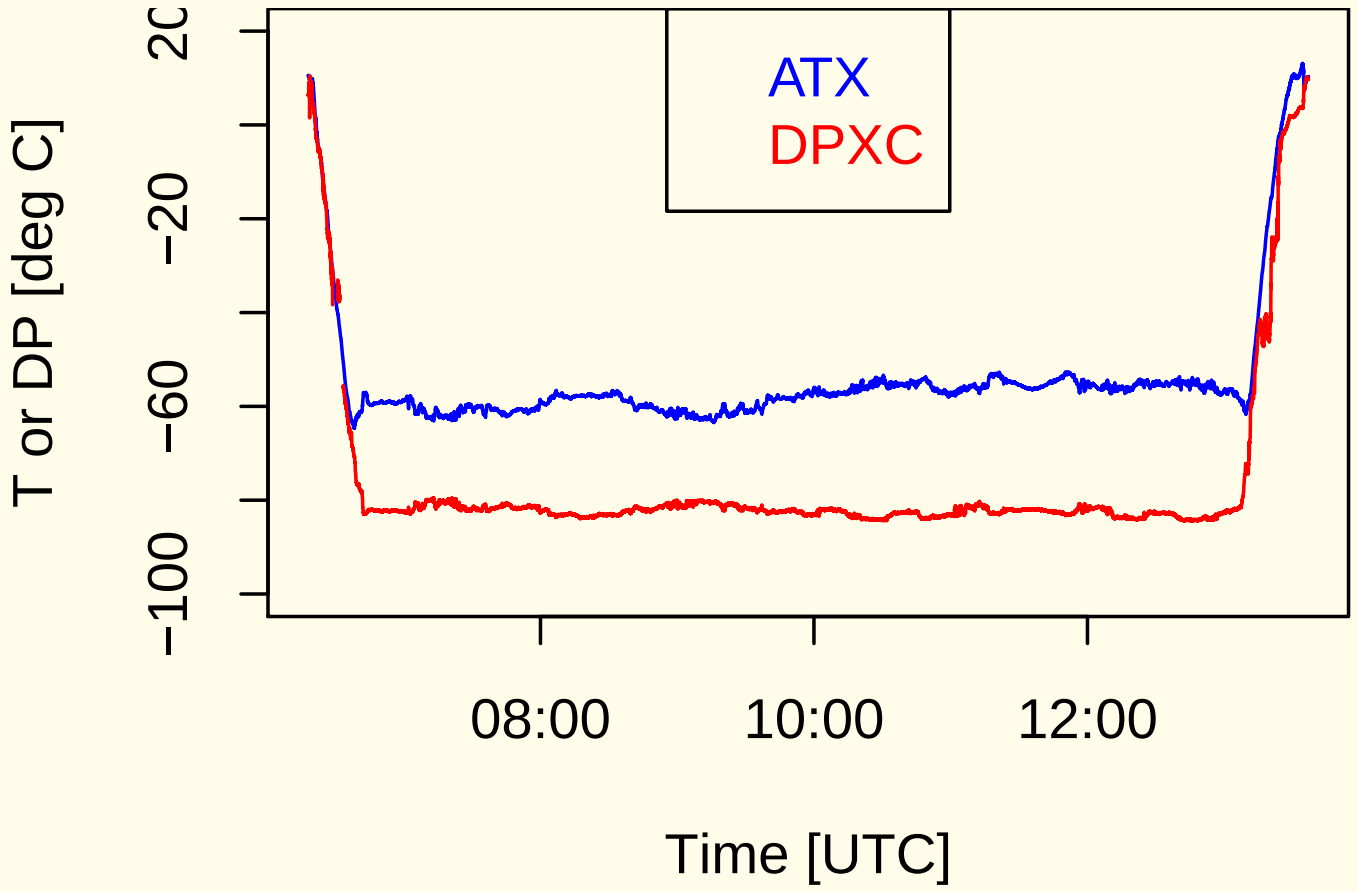
BASE GRAPHICS: scatterplot

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



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"))
```

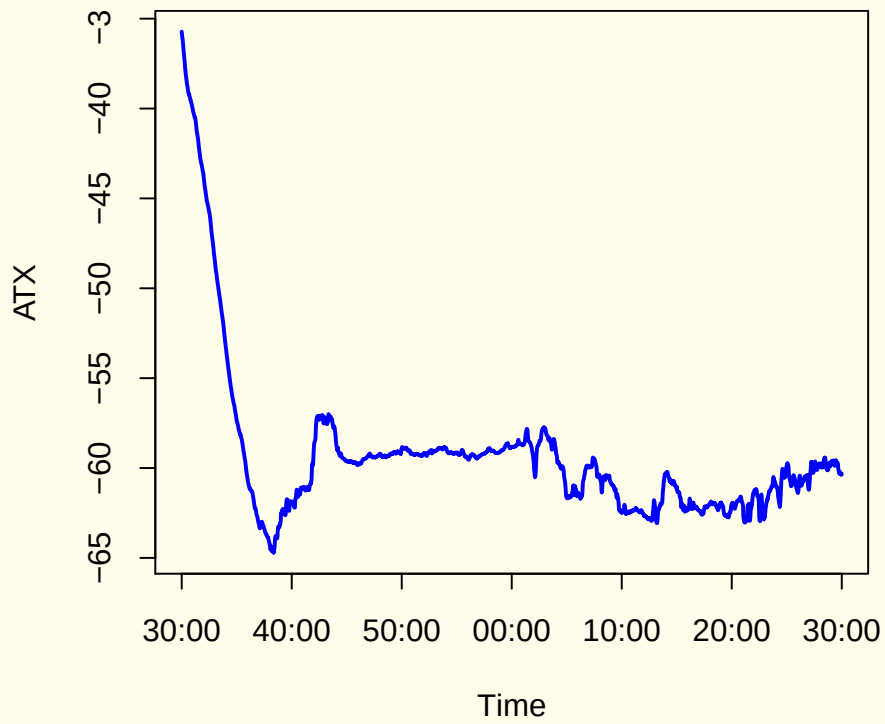


BASE GRAPHICS: adding structure with a data.frame'

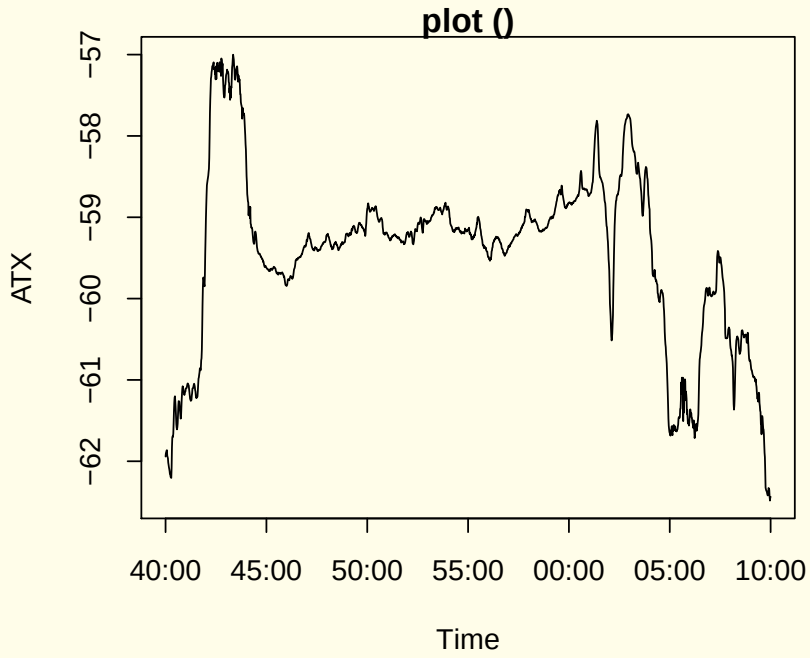
Consider using a data.frame to hold data for a plot:

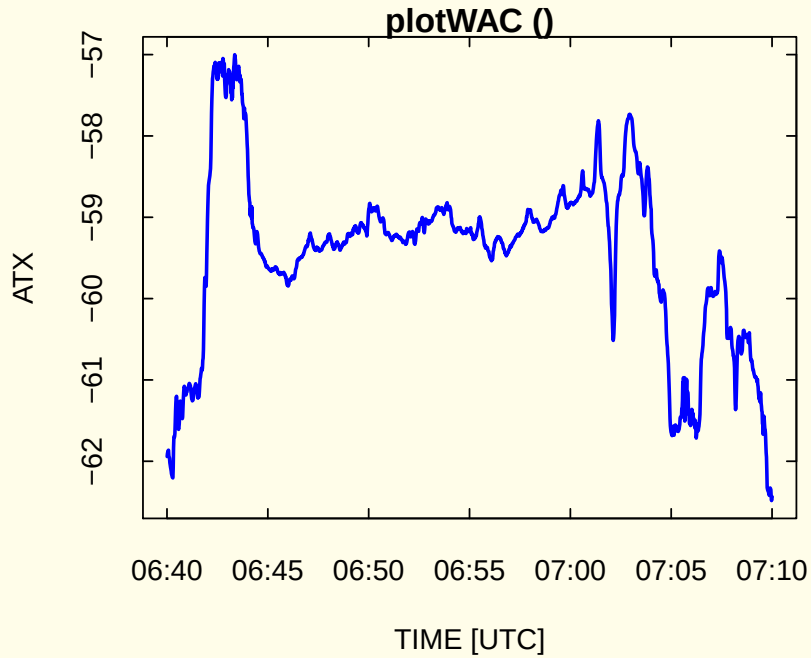
- subset appropriately
- apply plot operations to this object
- optionally save for archiving
- example:

```
r <- setRange(Data$Time, 63000,
    73000)
Plot1Data <- Data[r, c("Time",
    "ATX")]
plot(Plot1Data, type = "l",
    col = "blue", lwd = 2)
```



Ranadu GRAPHICS: see 'plotWAC'





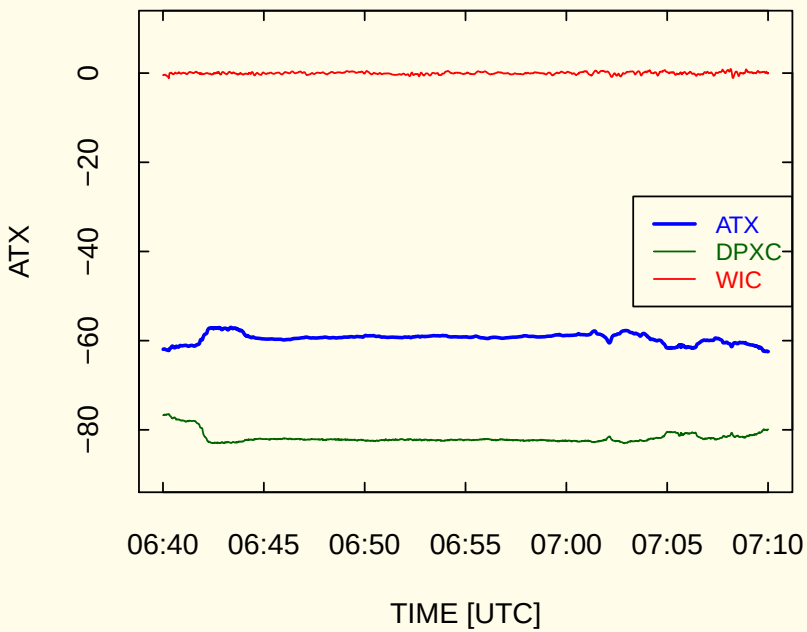
Differences:

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

Ranadu GRAPHICS: another 'plotWAC' feature:

```
r <- setRange(Data$Time, 64000,
    71000)
Plot1Data <- Data[r, c("Time",
    "ATX", "DPXC", "WIC")]
```

```
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.
- faceted plots (discussed below) are particularly useful for showing multiple variables.

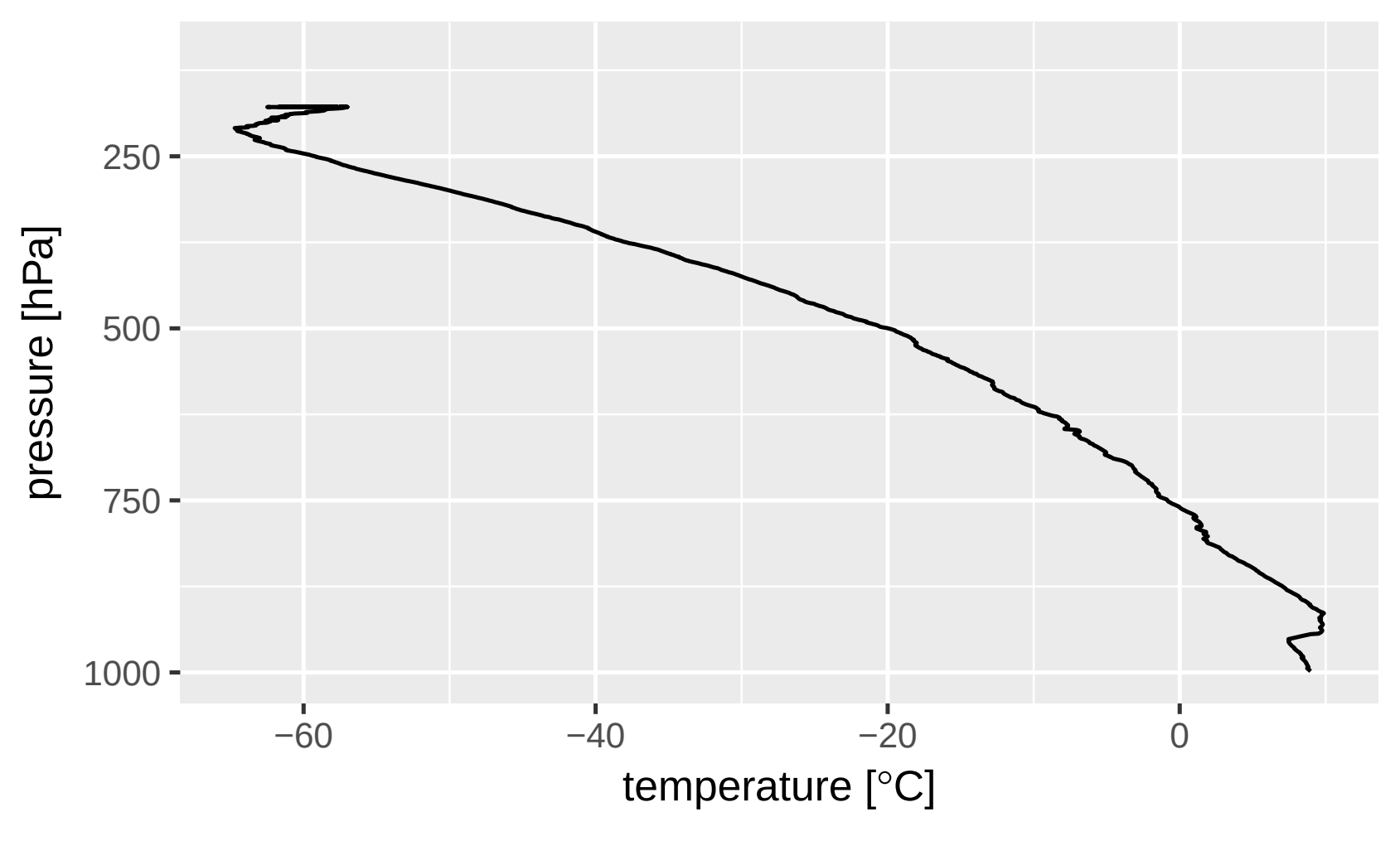
COMPONENTS of a ggplot

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, 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.

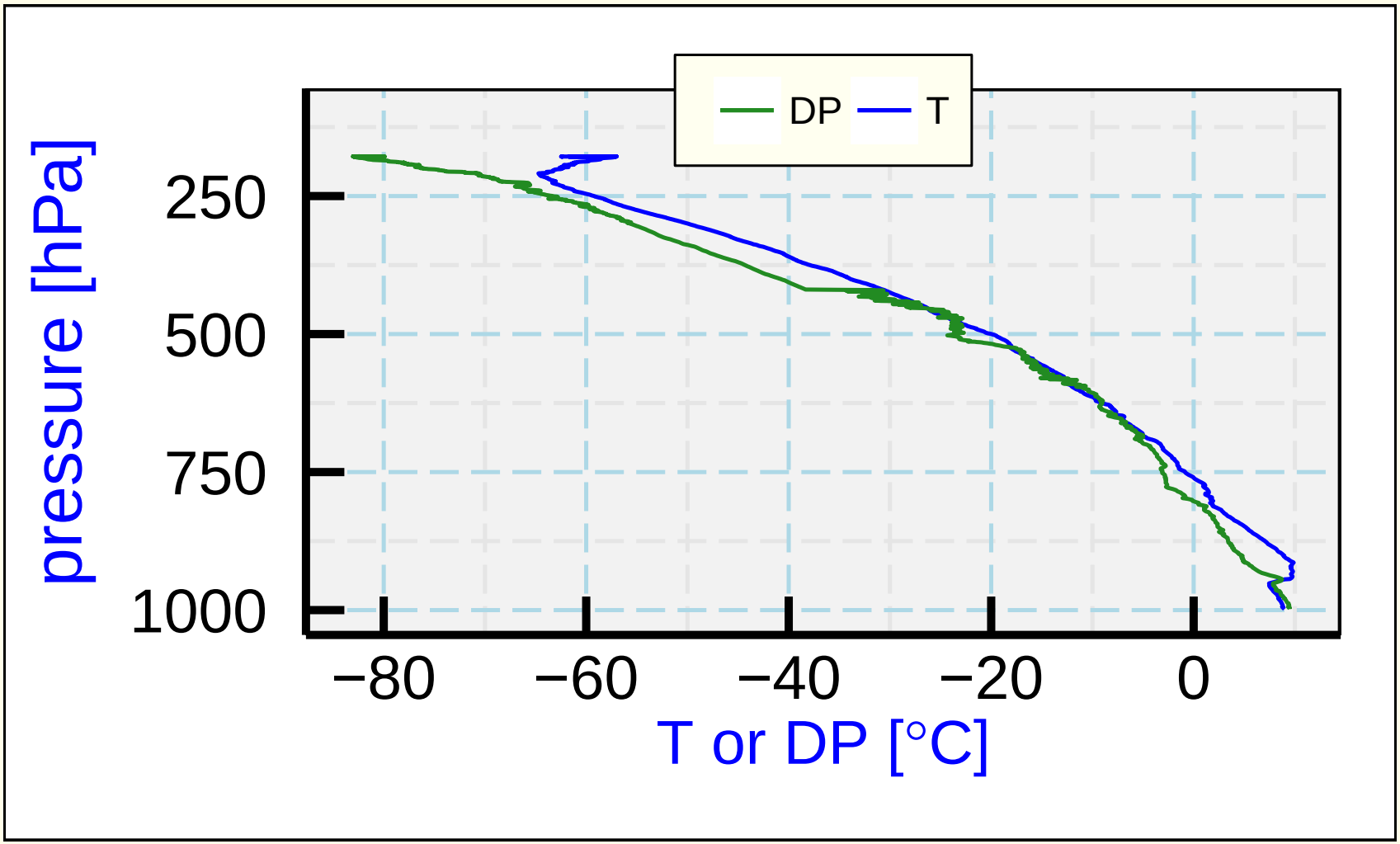
CONSTRUCTING A SIMPLE ggplot:

```
D <- Data[setRange(Data, 61900, 71000), c("Time", "ATX", "PSXC", "DPXC", "WIC")]
ggplot(D, aes(ATX, PSXC)) + geom_path() + ylim(1000, 100) + ylab("pressure [hPa]") +
    xlab(expression(paste("temperature [", degree, "C]")))
```



Multiple Lines With Legend (ggplot):

```
## Using same data frame as for previous plot.
## Note how cLines and scale_colour_manual() select the colors used, over-riding ggplot defaults
cLines <- c('blue', 'forestgreen'); names(cLines) <- c('T', 'DP')
ggplot (D, aes(ATX, PSXC)) + geom_path (aes(colour='T')) + ylab('pressure [hPa]') +
  geom_path (aes(x=DPXC, colour='DP')) +  scale_colour_manual (name='', values=cLines) +
  ylim(1000, 100) + xlab(expression(paste('T or DP [',degree, 'C]'))) + theme_WAC()
```



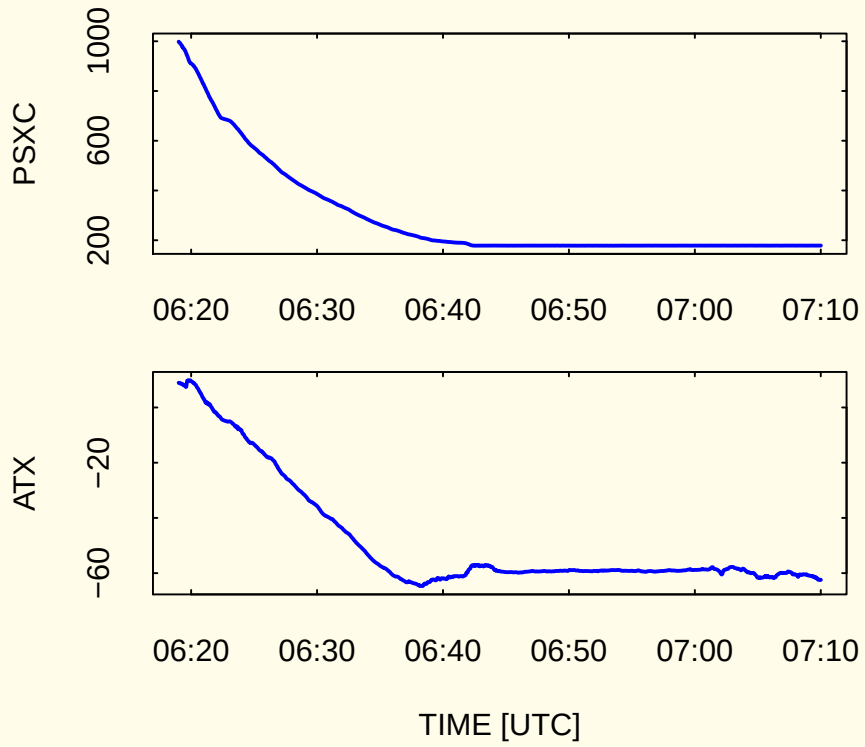
Examples of Some Other Types of Plots
Study These as Guides

Plot Type:

1. Stacked plots sharing the same time axis. (But see also the faceted plots below.)

R code and result:

```
layout(matrix(1:2), widths = 1, heights = c(9,
    12))  ## 1 col, 2 rows
op <- par(mar = c(2, 4, 1, 2) + 0.1, oma = c(1.1,
    0, 0, 0))  ## margins hide top plot abscissa title
with(D, plotWAC(Time, PSXC, ylab = "PSXC"))
op <- par(mar = c(5, 4, 1, 2) + 0.1)
with(D, plotWAC(Time, ATX, ylab = "ATX"))
```



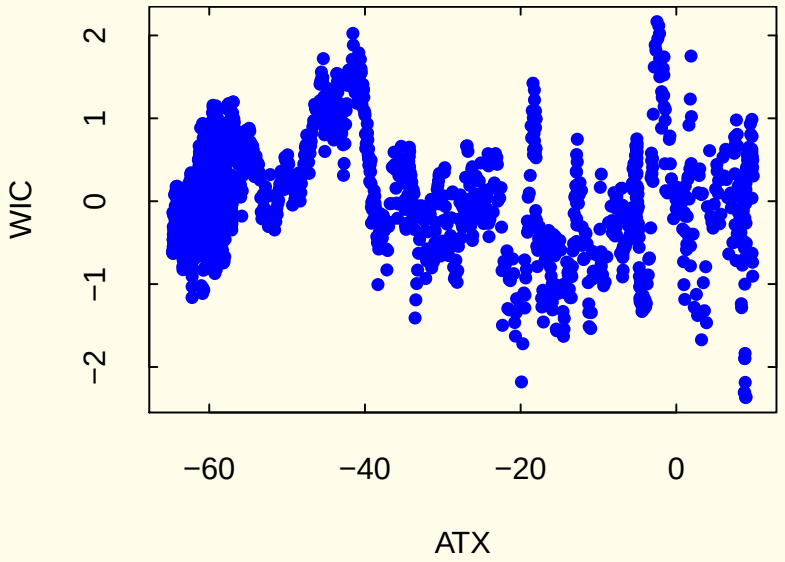
Examples of Some Other Types of Plots
Study These as Guides

Plot Type:

1. Stacked plots sharing the same time axis. (But see also the faceted plots below.)
2. Standard scatterplot.

R code and result:

```
## Note: plotWAC assumes a time series. If the
## first plot variable is not 'Time', must
## specify 'xlab='ATX'' or the function will
## fail trying to interpret 'ATX' as a time.
## 'type='p'' specifies plotted points for
## each measurement pair. Can also use
## 'plot()' similarly.
with(D, plotWAC(data.frame(ATX, WIC), type = "p",
    xlab = "ATX", ylab = "WIC"))
```



Examples of Some Other Types of Plots

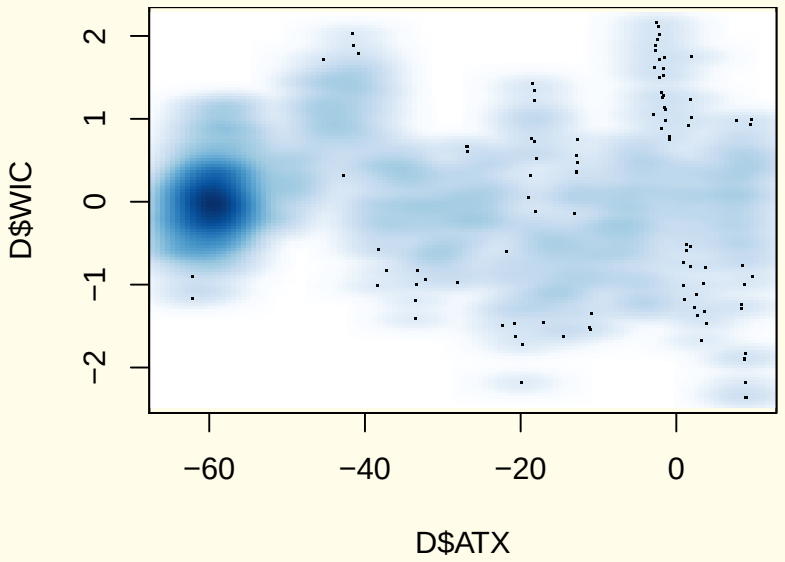
Study These as Guides

Plot Type:

1. Stacked plots sharing the same time axis. (But see also the faceted plots below.)
2. Standard scatterplot.
3. Density plot

R code and result:

```
## Scatterplots with large numbers of points
## become hard to interpret. One solution is
## to use the function 'smoothScatter' to get
## a representation of the density of points.
## Here are the same measurements used in the
## previous plot:
smoothScatter(D$ATX, D$WIC)
```



Examples of Some Other Types of Plots

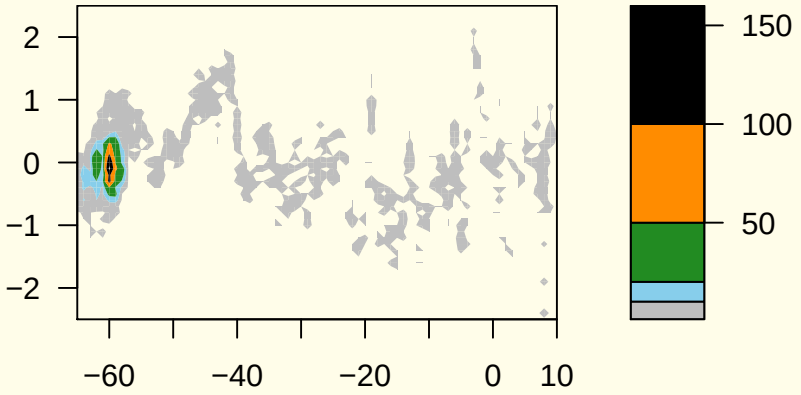
Study These as Guides

Plot Type:

1. Stacked plots sharing the same time axis. (But see also the faceted plots below.)
2. Standard scatterplot.
3. Density plot
4. Contour plot

R code and result:

```
## Another technique: bin and contour
## Same data as previous plots
colors <- c('gray', 'skyblue', 'forestgreen',
  'darkorange', 'black')
ix <- iy <- rep(0, nrow(D))
xlim <- seq(-65,10,by=1); ylim <- seq(-2.5,2.5,by=0.1)
for (i in 1:nrow(D)) {
  ix[i] <- which(xlim > D$ATX[i])[1]-1
  iy[i] <- which(ylim > D$WIC[i])[1]-1
}
A <- rep(0, length(xlim)*length(ylim))
dim(A) <- c(length(xlim), length(ylim))
for (i in 1:nrow(D)) {
     A[ix[i],iy[i]] <- A[ix[i],iy[i]]+1
}
filled.contour(xlim, ylim, A,
 levels=c(1,10,20,50,100,160), col=colors)
```



Examples of Some Other Types of Plots

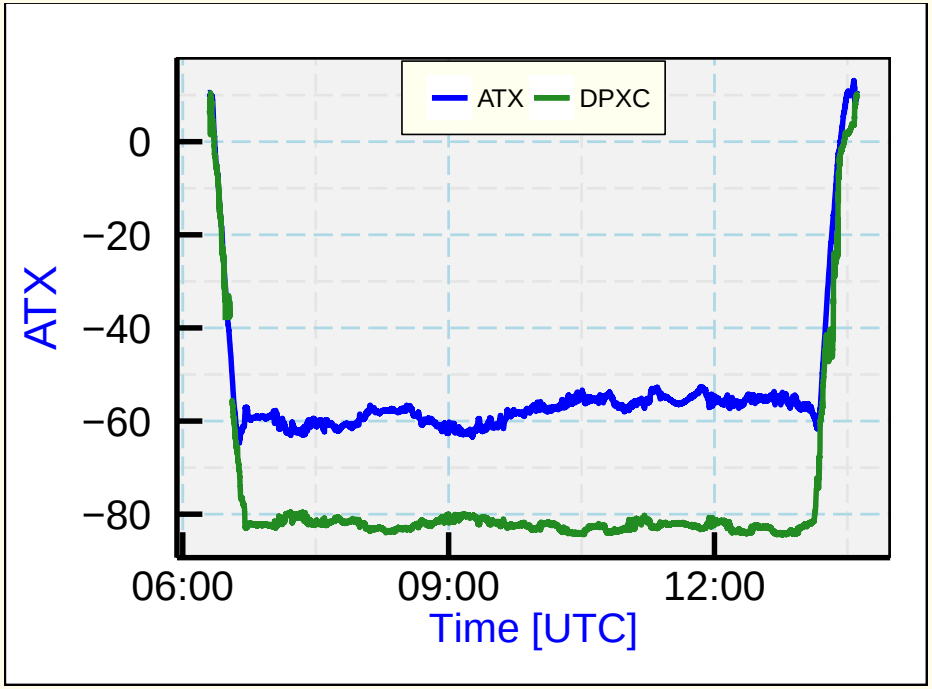
Study These as Guides

Plot Type:

1. Stacked plots sharing the same time axis. (But see also the faceted plots below.)
2. Standard scatterplot.
3. Density plot
4. Contour plot
5. ggplotWAC() standard plot

R code and result:

```
## ggplotWAC() provides single-command access
## to some ggplot2 commands often useful for
## time-series displays. (See also
## ?ggplot2::qplot as an alternative.)
## ggplotWAC() requires a data.frame as first
## argument with Time the first variable and
## plots histories of all other variables.
with(Data, ggplotWAC(data.frame(Time, ATX, DPXC)))
```



Examples of Some Other Types of Plots

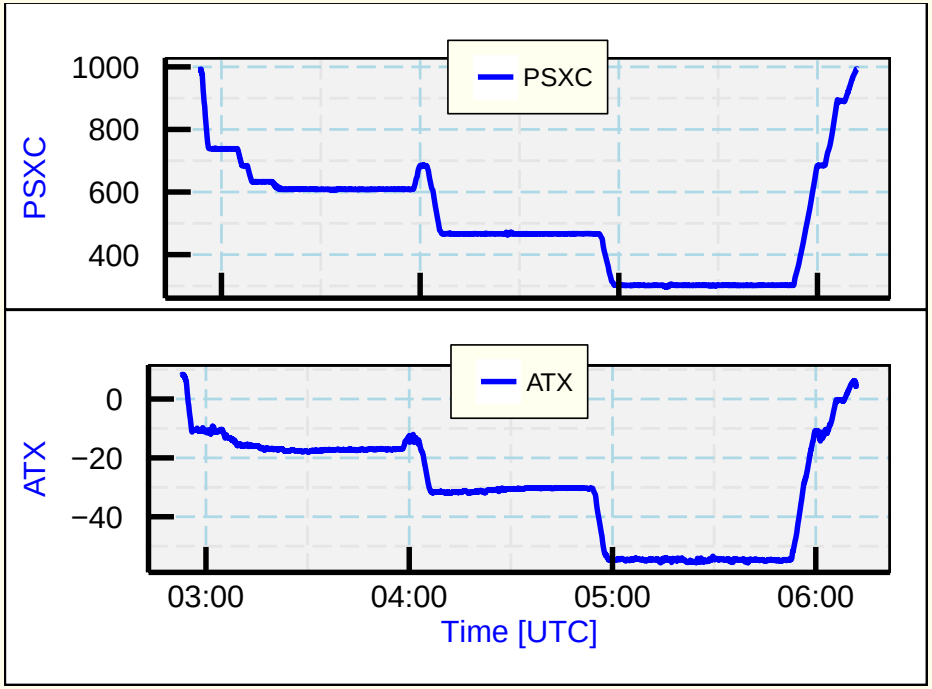
Study These as Guides

Plot Type:

1. Stacked plots sharing the same time axis. (But see also the faceted plots below.)
2. Standard scatterplot.
3. Density plot
4. Contour plot
5. ggplotWAC() standard plot
6. Multiple plots with ggplotWAC()

R code and result:

```
## first of two ways to generate multiple plots with
## ggplotWAC():
grid.newpage()
ggplotWAC(Data[, c("Time", "ATX")], pos = c(1, 2))
ggplotWAC(Data[, c("Time", "PSXC")], pos = c(2, 2))
## note the mis-alignment, fixed by faceting (next), or
## you can fine-tune margins via elements of 'theme()'
```



Examples of Some Other Types of Plots

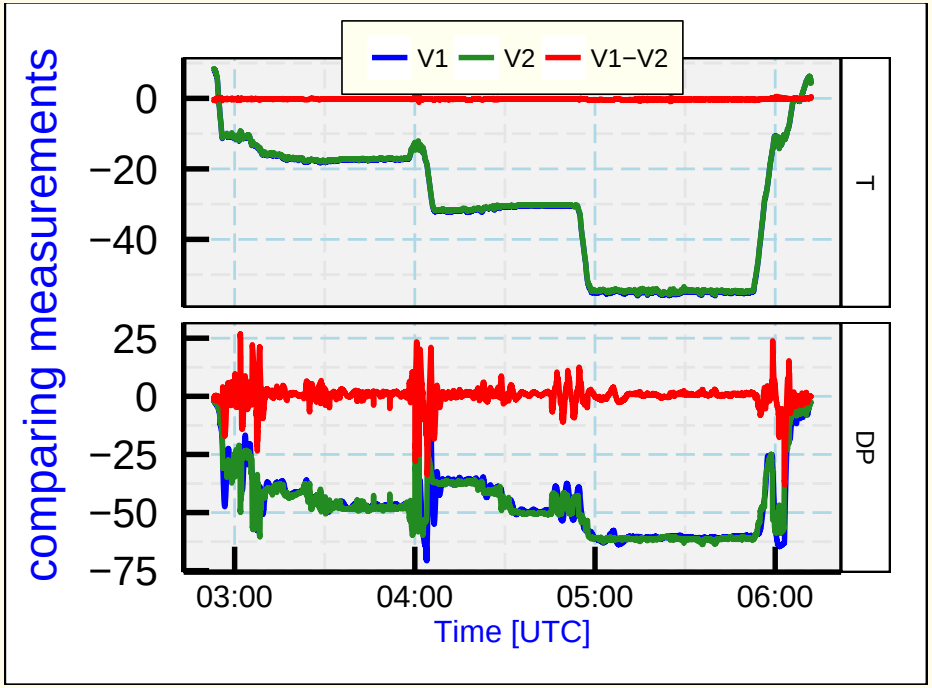
Study These as Guides

Plot Type:

1. Stacked plots sharing the same time axis. (But see also the faceted plots below.)
2. Standard scatterplot.
3. Density plot
4. Contour plot
5. ggplotWAC() standard plot
6. Multiple plots with ggplotWAC()
7. Faceted plots with ggplotWAC()

R code and result:

```
## for faceted plots using ggplotWAC, the
## data.frame order is important; see
## ?ggplotWAC().
with(Data, ggplotWAC(data.frame(Time, ATHR1, ATRL,
    ATHR1 - ATRL, DP_DPL, DP_VXL, DP_DPL - DP_VXL),
    ylab = "comparing measurements", panels = 2,
    labelL = c("V1", "V2", "V1-V2"), labelP = c("T",
        "DP"), legend.position = c(0.5, 1)))
```



Examples of Some Other Types of Plots

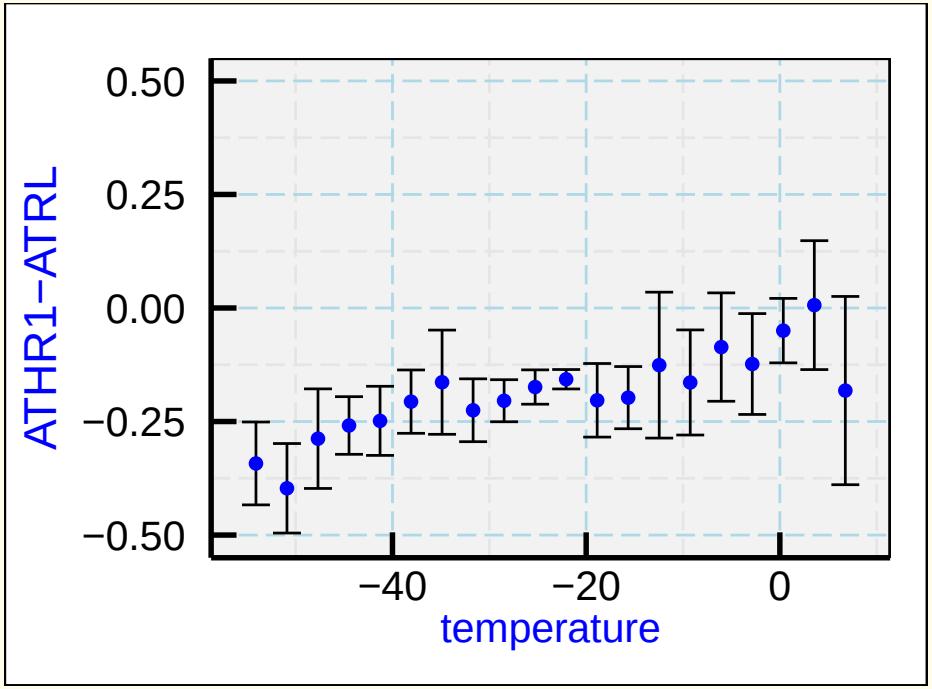
Study These as Guides

Plot Type:

1. Stacked plots sharing the same time axis. (But see also the faceted plots below.)
2. Standard scatterplot.
3. Density plot
4. Contour plot
5. ggplotWAC() standard plot
6. Multiple plots with ggplotWAC()
7. Faceted plots with ggplotWAC()
8. Error bars

R code and result:

```
## binStats partitions into bins and
## calculates the mean and standard deviation
## in each bin. See ?Ranadu::binStats
B <- binStats(data.frame(DT = Data$ATHR1 - Data$ATRL,
    ATX = Data$ATX))
ggplot(data = B) + geom_errorbar(aes(x = xc,
    ymin = ybar - sigma, ymax = ybar + sigma)) +
    geom_point(aes(x = xc, y = ybar), colour = "blue",
        size = 2) + ylab("ATHR1-ATRL") + xlab("temperatur
    ylim(-0.5, 0.5) + theme_WAC()
```



Proceed to the next tab (fitting)