Trellis Graphics

Statistics 135

Autumn 2005

`.First` and `.Last` Functions

It is possible to configure **S** to load certain packages, override defaults, etc when started. One approach for this is the `.First` function. The idea behind this is similar to the use of `.cshrc` or `.bashrc`, and `.login` files in Unix/Linux.

Any commands that you wish to run everytime you start **S** should go into the `.First` function. An example is

```
.First <- function() {
  library(lattice)  # load trellis graphics library
  library(MASS)     # load MASS package
  options(width=75) # set output width at 75 characters
  options(contrasts=c("contr.treatment", "contr.poly"))
      # set contraint method for contrasts of factors
      # actually the default in R, not in S-Plus
}
```

When **S** starts, this function is run.

Similarly, there is a `.Last` function, similar in concept to Unix/Linux's `.logout` file. Any commands you want run when quitting should be put in this function.

Trellis Graphics

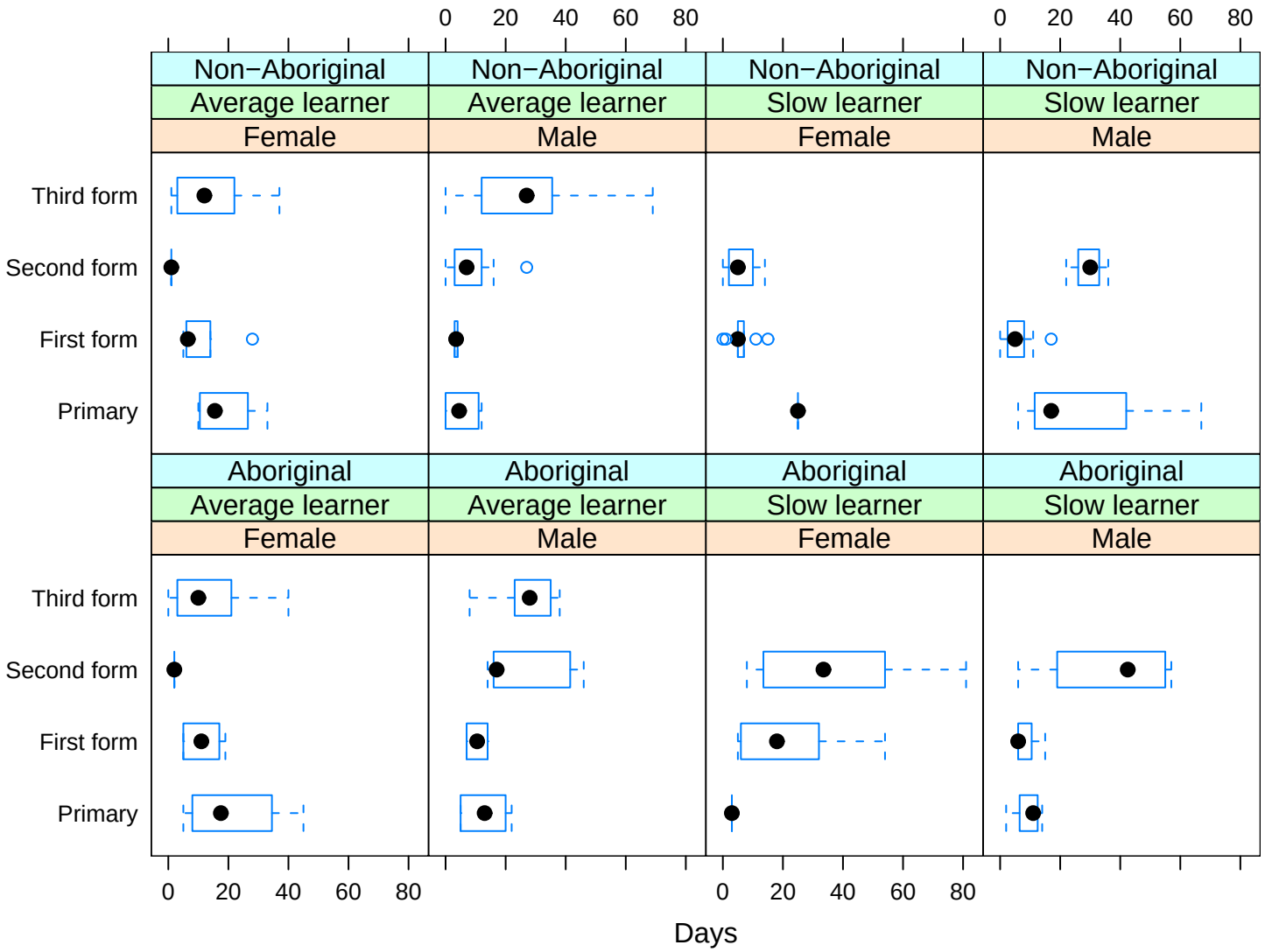
Example: Absenteeism from School in Rural New South Wales

Description:

The 'quine' data frame has 146 rows and 5 columns. Children from Walgett, New South Wales, Australia, were classified by Culture, Age, Sex and Learner status and the number of days absent from school in a particular school year was recorded.

This data set is available in the `MASS` package under the name `quine`. To access the package, give the command `library(MASS)`.

Want to look at what variables are associated with differences in the number of days absent. Lets start with some box plots.


Non−Aboriginal
Average learner
Female
Non−Aboriginal
Average learner
Male
Non−Aboriginal
Slow learner
Female
Non−Aboriginal
Slow learner
Male
Aboriginal
Average learner
Female
Aboriginal
Average learner
Male
Aboriginal
Slow learner
Female
Aboriginal
Slow learner
Male
Third form
Second form
First form
Primary
0
20
40
60
80
Days

The command for this plot are

```
trellis.device("postscript", file="../quinebox.eps", width=9,
  height=6 ,horiz=F, col=T)
bwplot(Age ~ Days | Sex*Lrn*Eth, data=Quine, layout=c(4,2))
dev.off()
```

Comments:

1. The trellis function for boxplots, `bwplot`, gives a newer style of boxplots, where a dot is used for the median, instead of a line. I can't see anything in the `bwplot` options that would allow for the older, more standard style of boxplots.

2. Something similar to this plot could be done with more standard plotting commands, but it would be much more work and probably wouldn't look as good.

Trellis plots are a useful approach for graphical exploratory data analysis (EDA)

Allows for the examination of complicated, multiple variable relationships.

Plot types - univariate:

- `barchart`: guess :)
- `bwplot`: boxplots
- `densityplot`: kernel density estimates
- `dotplot`:
- `histogram`: guess again
- `qqmath`: quantile plots against known distributions

- `stripplot`: display numeric data against a numeric variable
- `piechart`: (Not available in **R**, only in **S-Plus**)

Plot types - bivariate:

- `qq`: q-q plots for comparing two distributions
- `xyplot`: scatterplot

Plot types - trivariate:

- `contourplot`: contourplot of a surface on a regular grid
- `levelplot`: pseudo-colour plot of a surface on a rectangular grid
- `wireframe`: perspective plot of a surface evaluated on a regular grid

- `cloud`: 3-D scatterplot

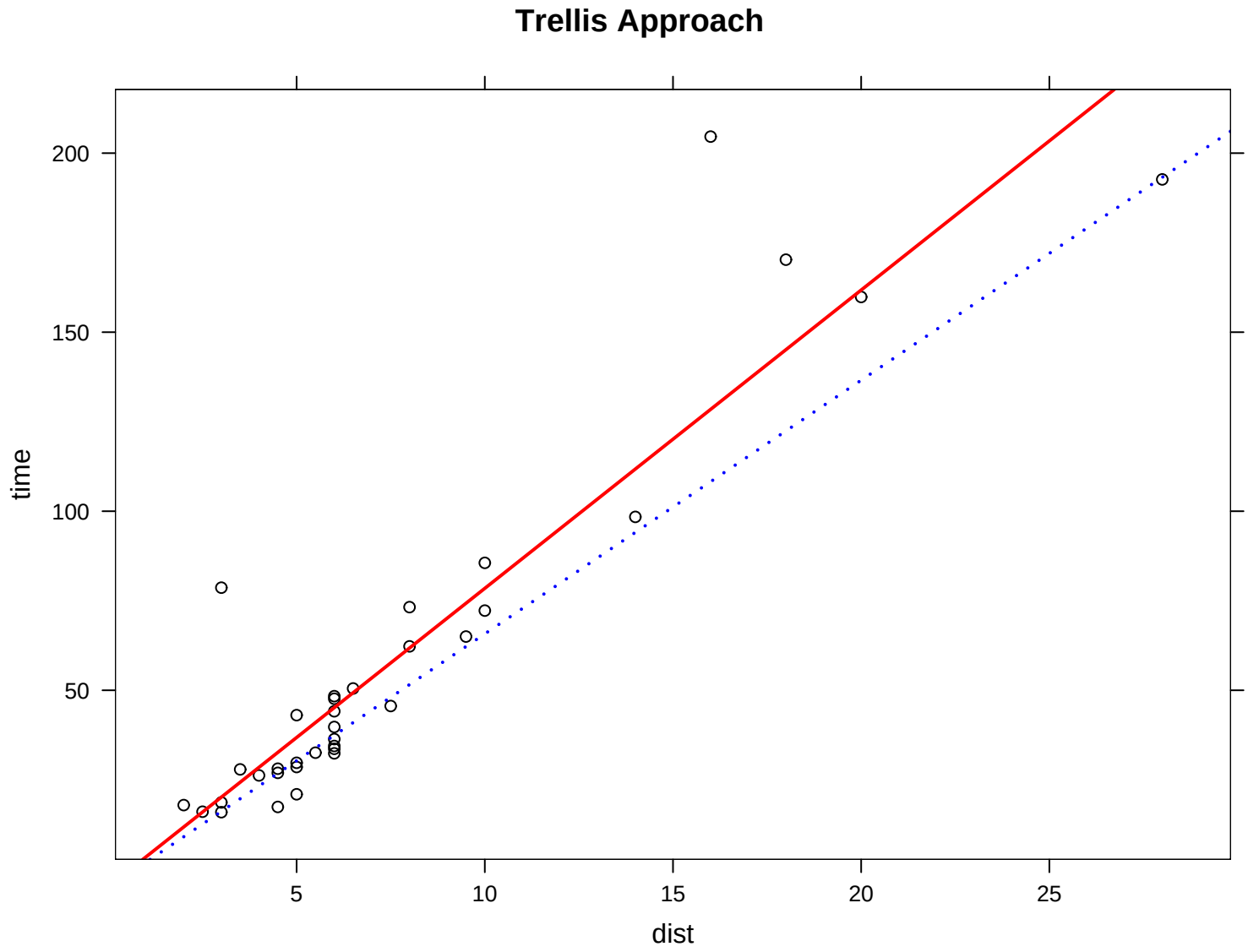
Plot types - Hypervariate (at least two)

- `splom`: scatterplot matrix
- `parallel`: parallel coordinates plots

In **S-Plus**, the trellis graphics routines are contained in the library `trellis`. This library should be automatically loaded when starting **S-Plus**. However, it is doesn't, give the command `library(trellis)`. In **R**, you need to load it with the command `library(lattice)`

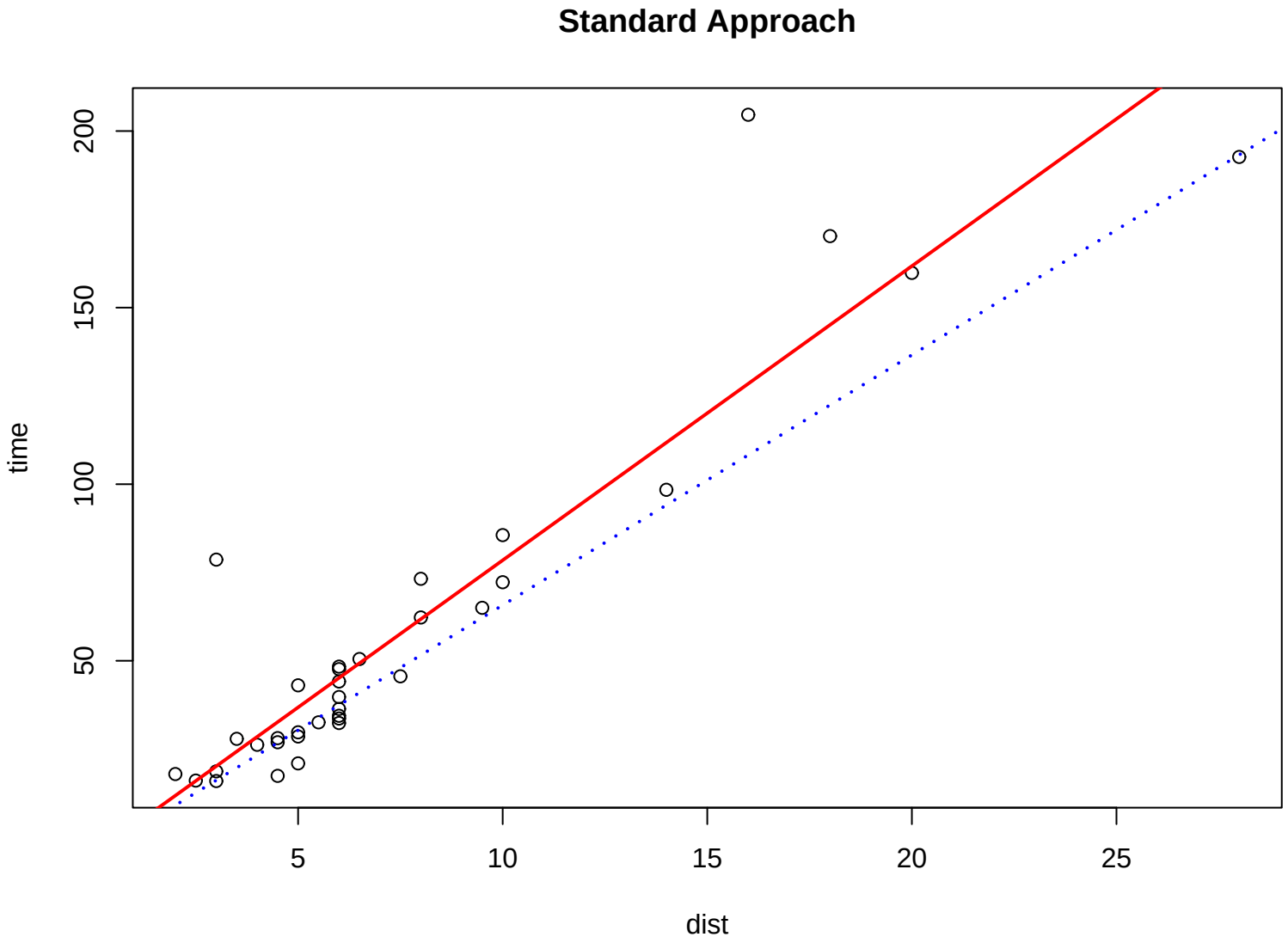
The approach to creating trellis graphics is a bit different that we seen so far. Instead of starting with the basic plot and adding elements to it, a single command is given for the plot.

For example, lets look at the data set `hills` from `MASS` by a scatterplot with some different fits added. This data set looks at the record times in 1984 for 35 Scottish hill races.


Trellis Approach
time
dist
200
150
100
50
5
10
15
20
25

```
trellis.device("postscript", file="../hills_trellis.eps",
  width=8, height=6 ,horiz=F, col=T)
xyplot(time ~ dist, data=hills, main="Trellis Approach",
  panel=function(x, y, ...) {
    panel.xyplot(x, y, ..., col=1)
    panel.lmline(x, y, type="l", col=2, lwd=2)
    panel.abline(lqs(y~x), lty=3, col=4, lwd=2)
  }
)
dev.off()
```

Now lets look at the same plot created by the standard commands discussed earlier.


Standard Approach
time
dist
50
100
150
200
5
10
15
20
25

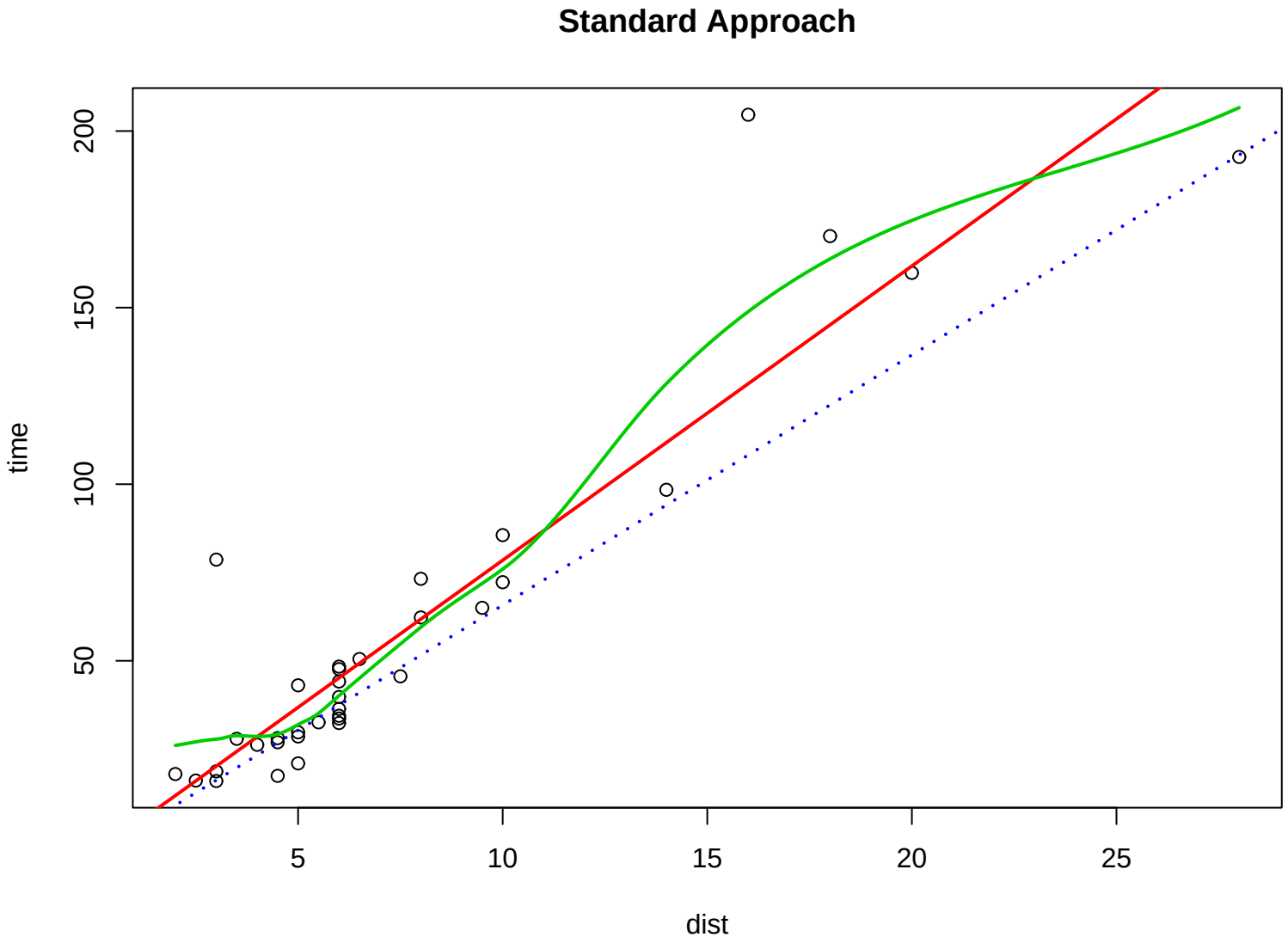
```
postscript("../hills_stand.eps", width=8, height=6, horiz=F)
par(mar=c(4,4,4,1) + 0.1, pty="m")
plot(time ~ dist, data=hills, main="Standard Approach")
abline(lm(time ~ dist, hills), col=2, lwd=2)
abline(lqs(time ~ dist, hills), col=4, lty=3, lwd=2)
dev.off()
```

They both look similar. Now suppose that we want to add a LOWESS (Locally Weighted Scatterplot Smooth) fit to this plot. For the standard approach, it can be done easily by running the additional commands

```
hills.lo <- loess(time ~ dist, hills, span=2/3, degree = 1)
hills.pred <-predict(hills.lo, data.frame(dist = seq(0, 30, 0.1)))
lines(seq(0, 30, 0.1), hills.pred, col=3, lwd=2)
```

assuming the figure is still in the current graphics device.

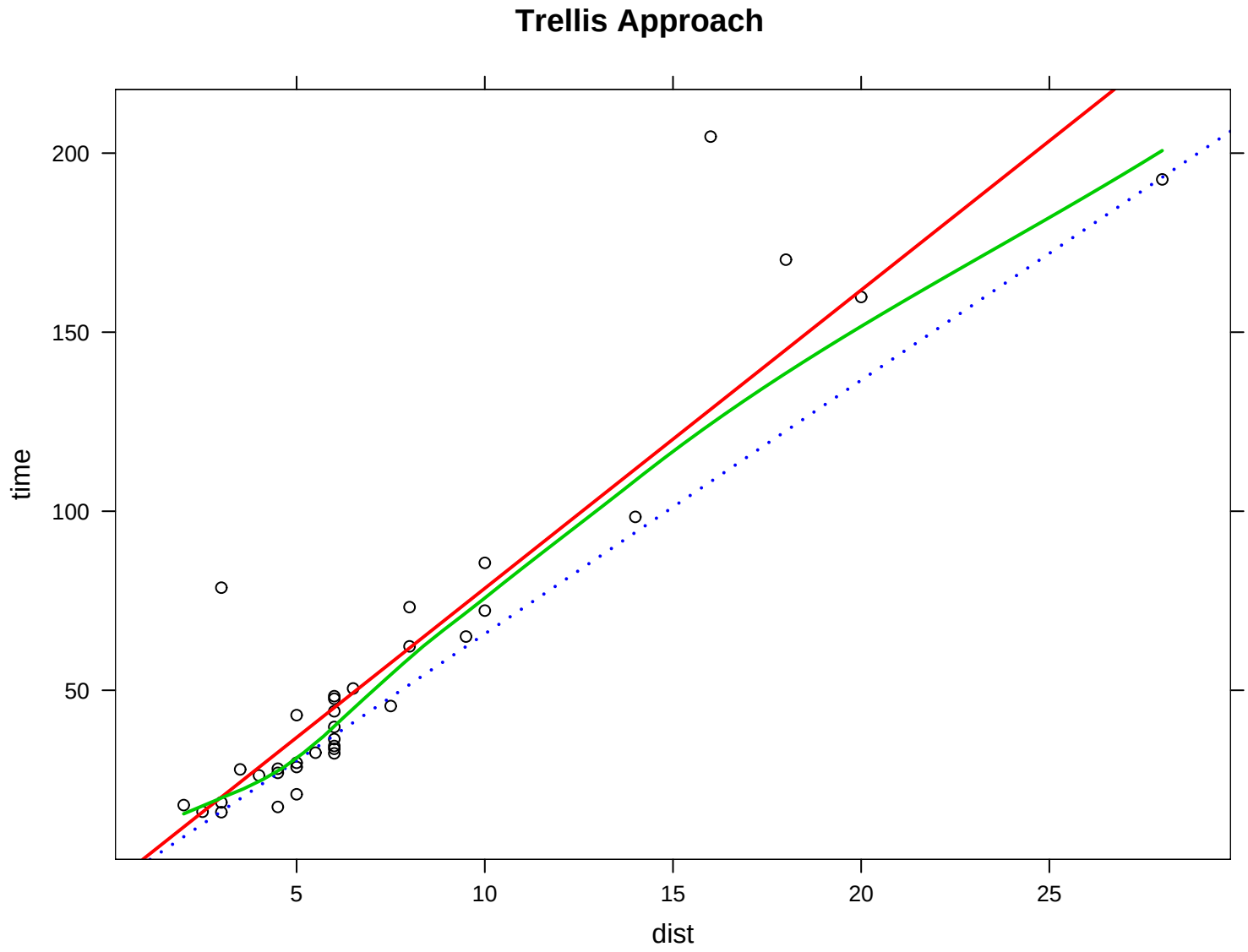
The resultant figure is


Standard Approach
time
dist
50
100
150
200
5
10
15
20
25

However this cannot be done in the trellis approach. The figure would need to be recreated from the beginning with the command

```
trellis.device("postscript", file="../hills_trellis.eps",
  width=8, height=6 ,horiz=F, col=T)
xyplot(time ~ dist, data=hills, main="Trellis Approach",
  panel=function(x, y, ...) {
    panel.xyplot(x, y, ..., col=1)
    panel.lmline(x, y, type="l", col=2, lwd=2)
    panel.abline(lqs(y~x), lty=3, col=4, lwd=2)
    panel.loess(x, y, ..., col=3, lwd=2)
  }
)
dev.off()
```

Note that the implementation of loess is a bit different in the two cases, which is why the figures aren't quite the same.


Trellis Approach
time
dist
200
150
100
50
5
10
15
20
25

Aside: `lqs` and `loess`

In the previous figures, three regression procedures were used

- Least squares: `lm`

- Resistant regression: `lqs`

  Fits a straight lines to the data with a procedure that is resistant to the effects of outliers. It does so working with ordered residuals.

  Suppose there are n data points and p regressors, including any intercept.

  The first three methods minimize some function of the sorted squared residuals. For methods '"lqs"' and '"lms"' is the 'quantile' squared residual, and for '"lts"' it is the sum of the 'quantile' smallest squared residuals. '"lqs"' and '"lms"' differ in the defaults for 'quantile', which are 'floor((n+p+1)/2)' and 'floor((n+1)/2)' respectively. For '"lts"' the default is 'floor(n/2) + floor((p+1)/2)'.

I believe the default for the method is "lts". The example used the default whatever it is. Given that p is small here, it should make much difference.

One way to think of this estimator is that acts like a trimmed estimator. It "removes" the most extreme values from the data set before calculating the statistics.

Think about the trimmed mean, which is implemented in **S** by `mean(x, trim=p)`, where `trim` is between 0 and 0.5. So when **S** calculates this, it removes the smallest $\lfloor np \rfloor$ and the largest $\lfloor np \rfloor$ observations from the data set and calculates the sample average of the remaining values.

What `lqs` is more complicated, but it has the same feel by removing observation with the most extreme residuals.

- LOWESS: `loess`

  This is an example of a scatterplot smoother. Like the other 2 cases, it will not return a straight line for the fit, but a smooth curve.

  The idea behind the method is if you wish to get a fit at a point x_0, only use near by points, and to weight the closest points more. A common weighting scheme is

$$w(x; x_0) = \left(1 - \left|\frac{x - x_0}{\text{maxdist}}\right|^3\right)^3$$

  where maxdist depends on the density of points in the data set and the span, the fraction of points to be included in the fit.

  The standard approaches use weighted least squares for the fits (either linear or quadratic), though other fitting methods can be used.