

Anomaly Detection via Depth Quantile Functions

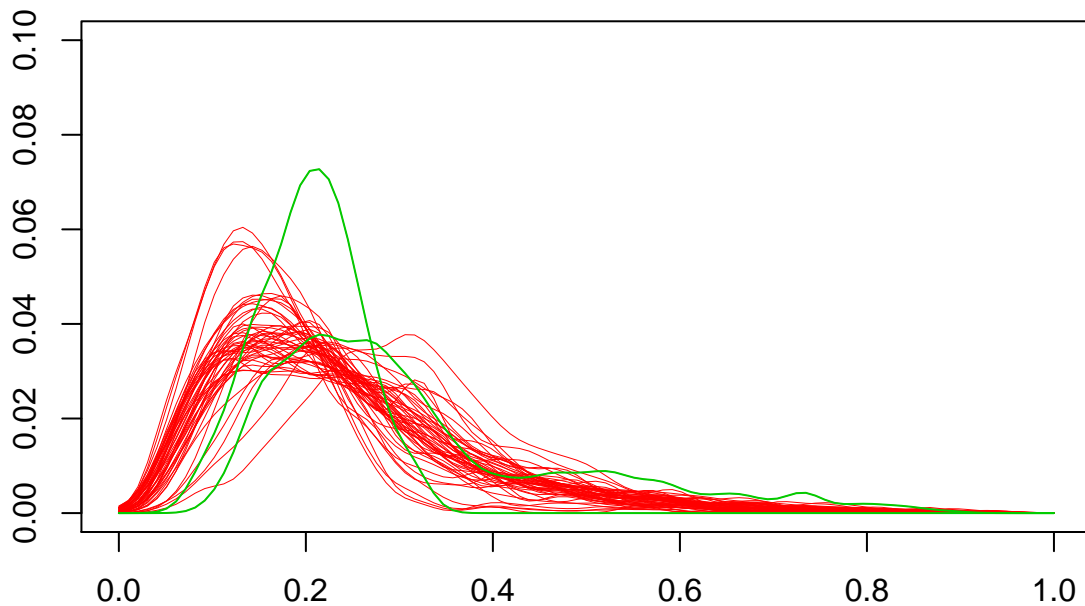
A vignette for using the (adaptive) depth quantile function approach for graphical anomaly detection, as described in “Antimodes and Graphical Anomaly Exploration via Depth Quantile Functions” by Gabriel Chandler and Wolfgang Polonik (2022).

First, we import the relevant functions (hidden in output), available at <https://github.com/GabeChandler/AnomalyDetection/blob/main/RCode>. All code runs in base-R.

We illustrate this on the Iris data set, available in base-R. We consider the observations 51-102, each consisting of 4 variables. The first 50 all come from the same species (Versicolor), so observations 51 and 52 constitute outliers (Virginica species).

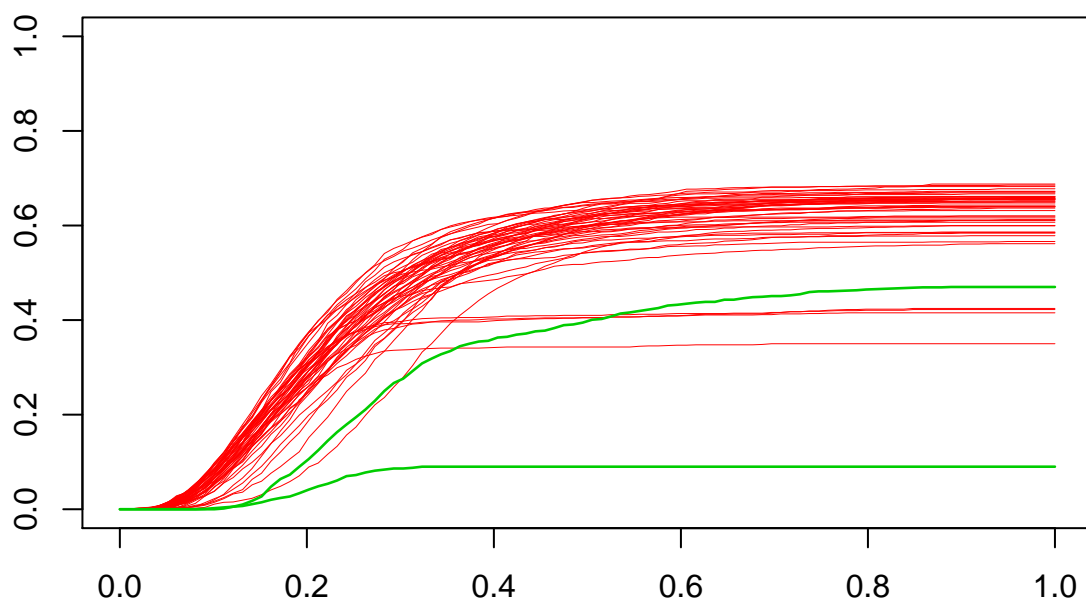
```
data(iris)
iris.sub <- iris[51:102,1:4]
fit.dqf <- dqf.outlier(iris.sub, range=6)
dqf.explore(fit.dqf, 51:52) #visualize the DQFs highlighting the known anomalies
```

Select Observations – Press ESC when done



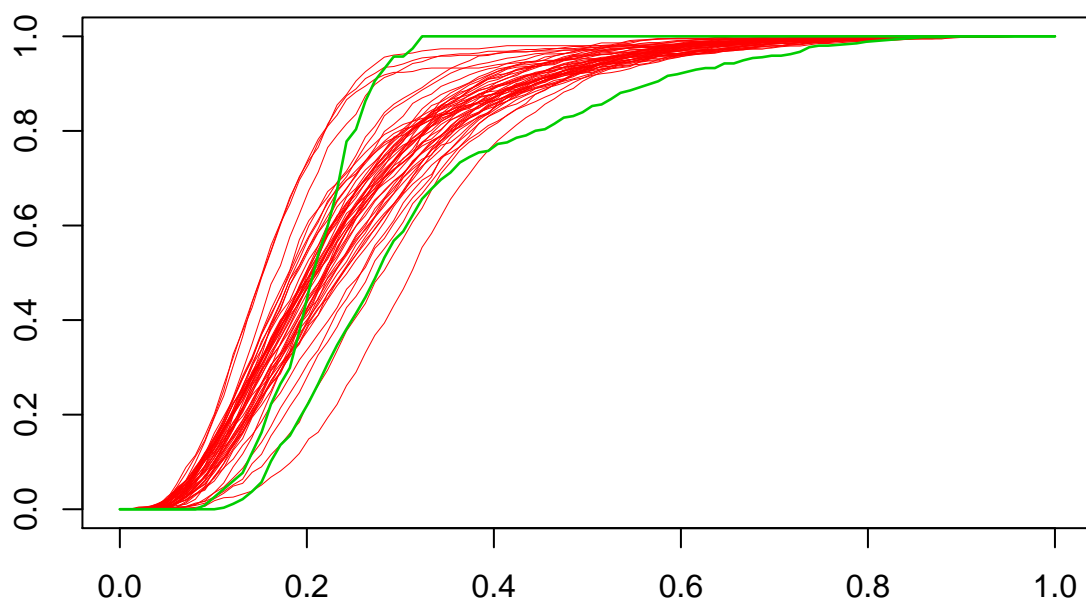
1 of 3

Select Observations – Press ESC when done

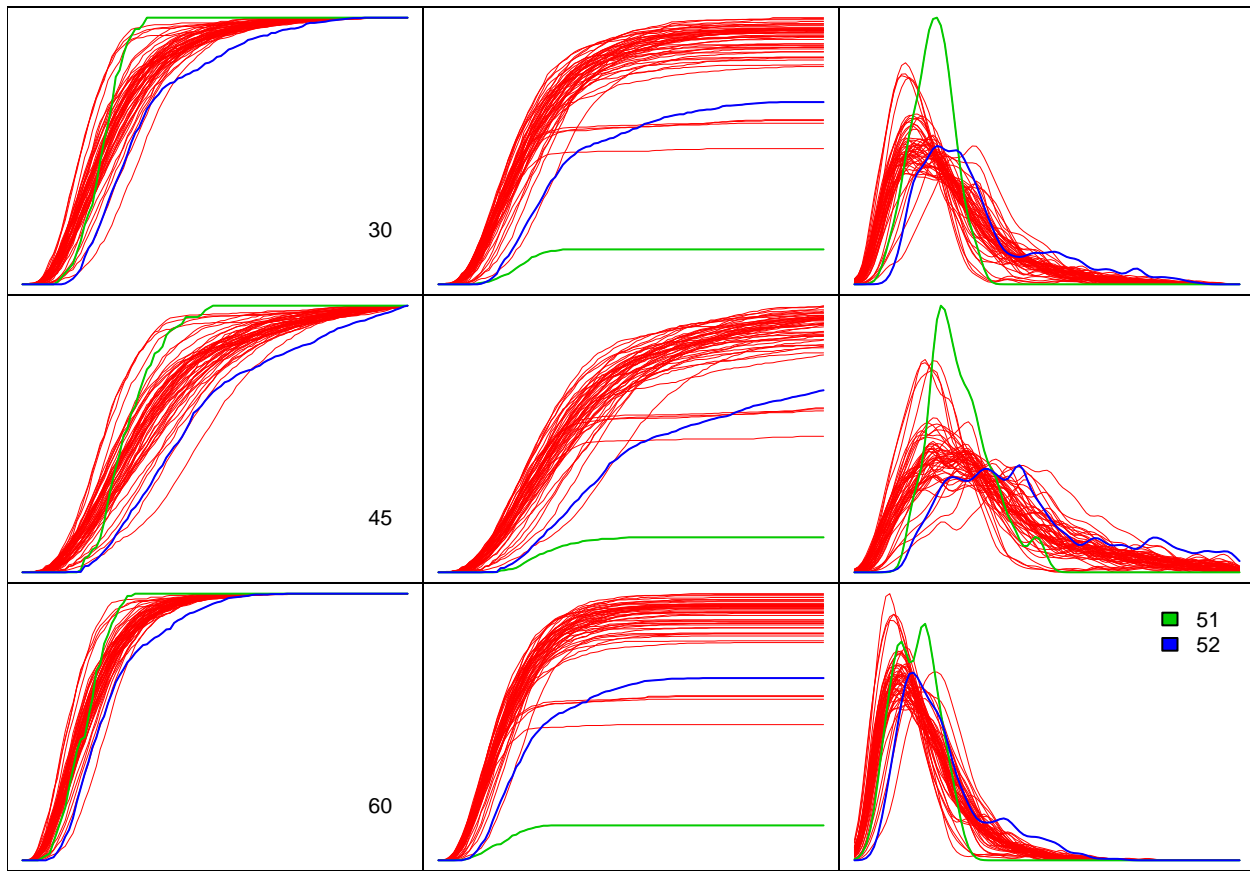


2 of 3

Select Observations – Press ESC when done



3 of 3



```
## [1] 51 52
```

An appropriate value of *range* used in *dqf.outlier* depends on both the dimension of the data as well as the geometry of the point cloud. This is usually done via guess-and-check. For large sample sizes, it is advisable to determine this value using a smaller subset of the data for computational simplicity.

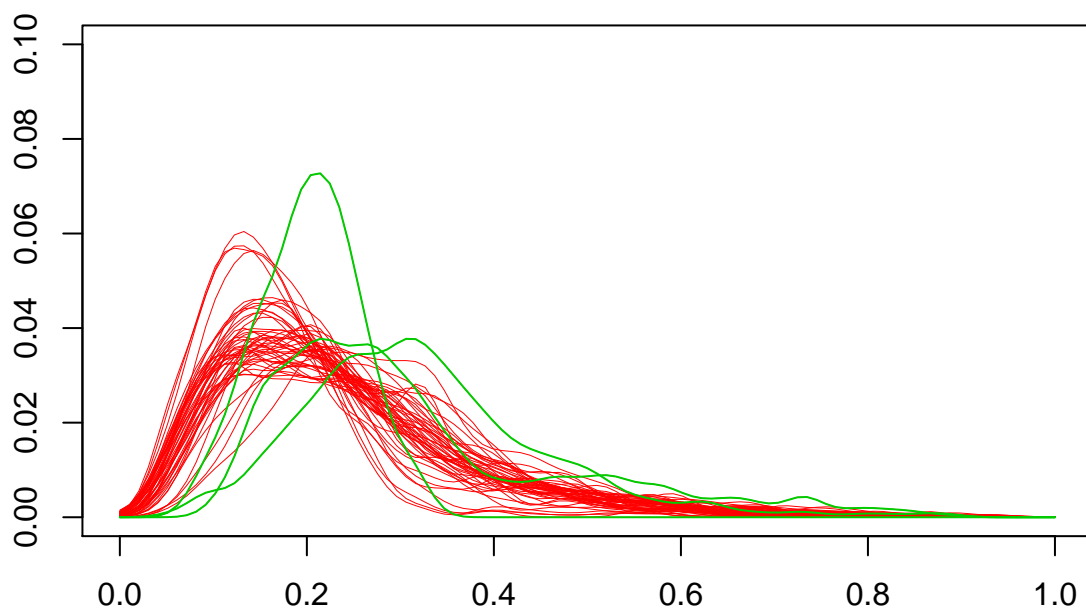
The default is to use a normal base distribution and adaptive DQF (windsorized variance of projections). For the non-adaptive DQF, add the argument *k.w=0* when calling *dqf.outlier*.

Rather than specify the indices of outlying observations, the code presents a sequence of three interactive graphs, where the user can click on any interesting functions. The functions corresponding to those observations will be highlighted in subsequent graphs, with their indices given in the final graph. The interactive graphs are shown at the second of the three angles. After you are finished selecting interesting functions, press the *ESC* key as directed to proceed to the next graph.

We might note that there is at least one additional interesting observation detected by the method. This turns out to observation 69 in the iris data set.

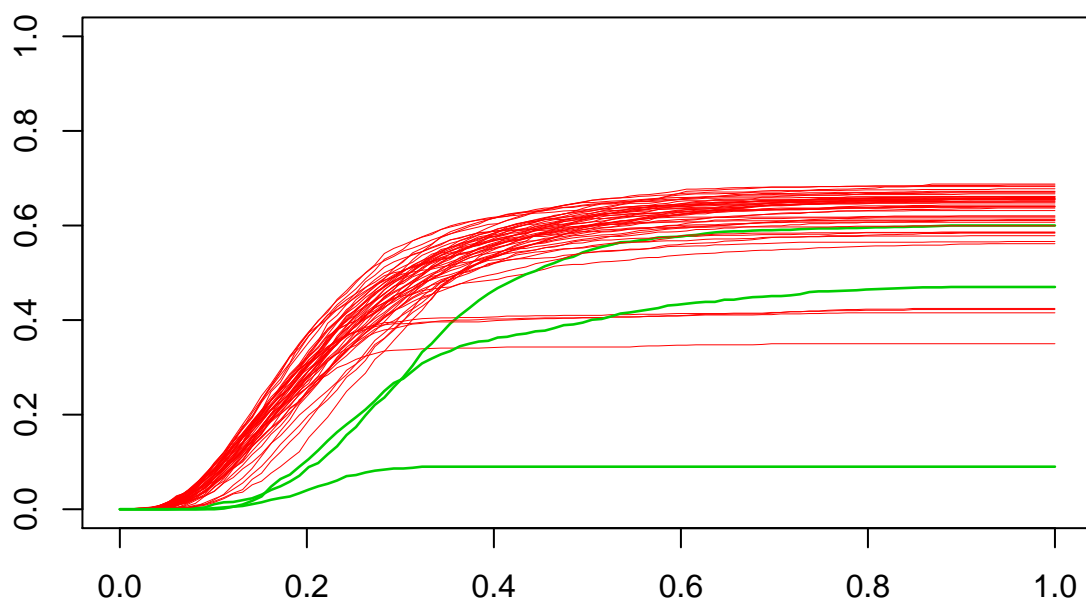
```
dqf.explore(fit.dqf, c(19,51:52))
```

Select Observations – Press ESC when done



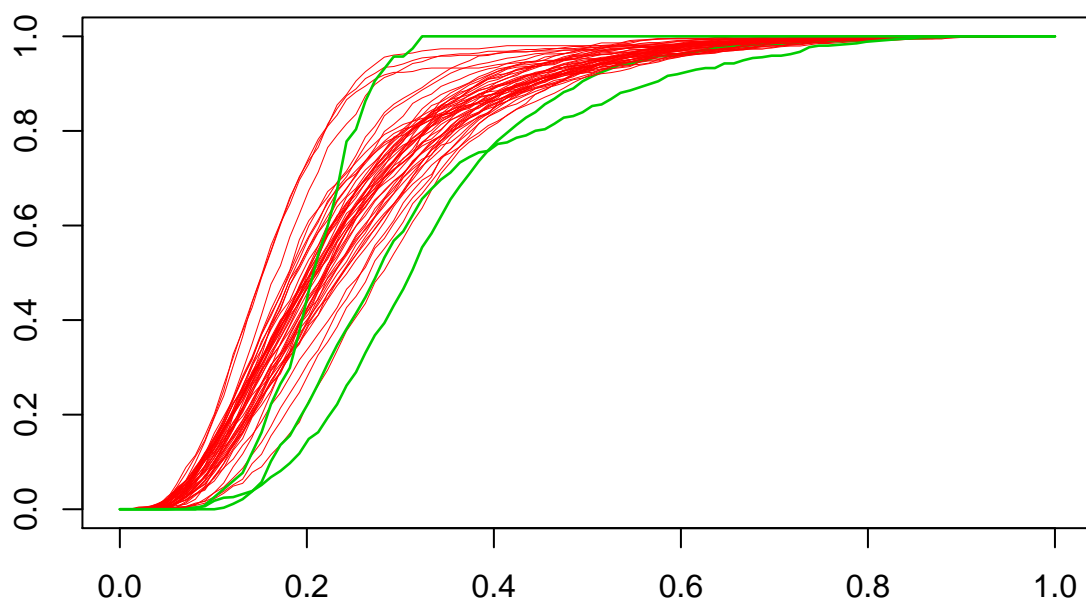
1 of 3

Select Observations – Press ESC when done

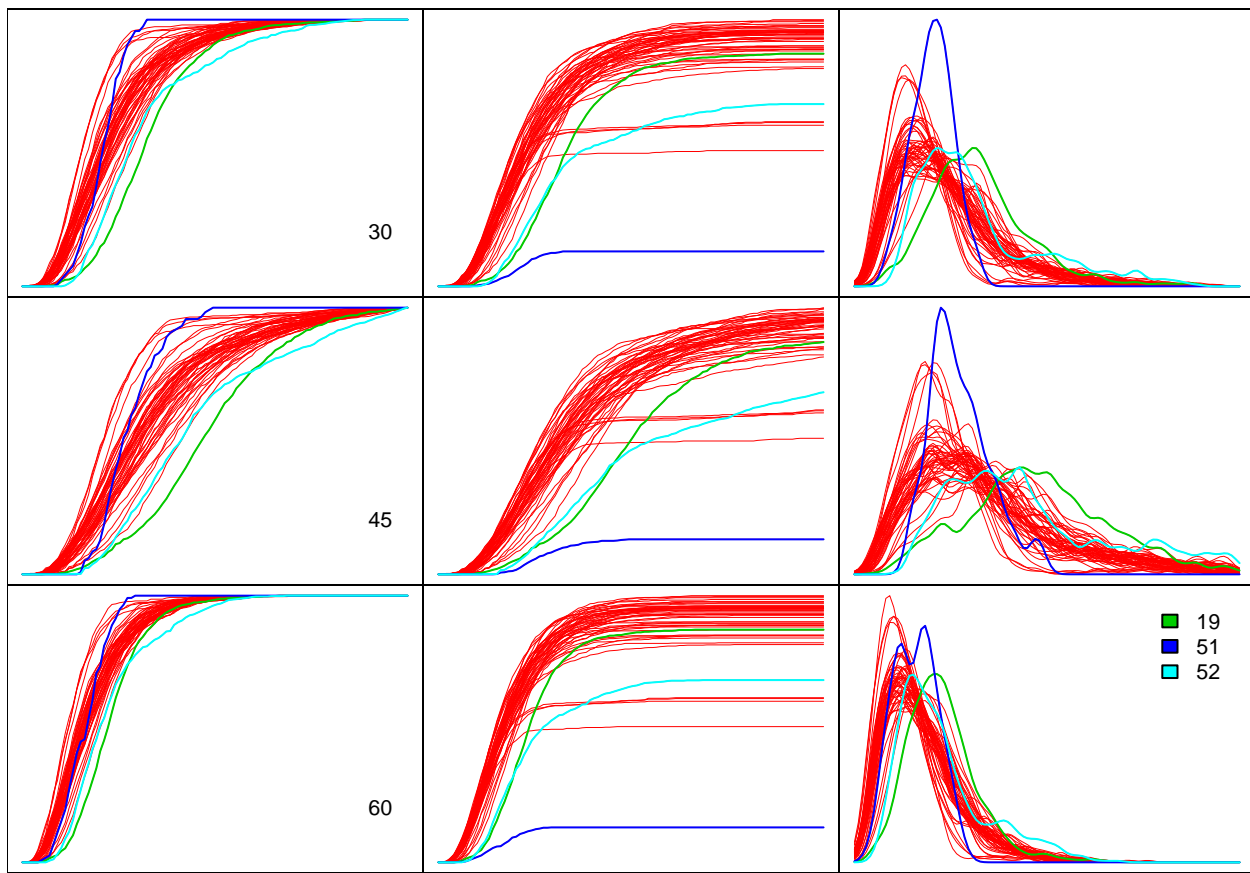


2 of 3

Select Observations – Press ESC when done



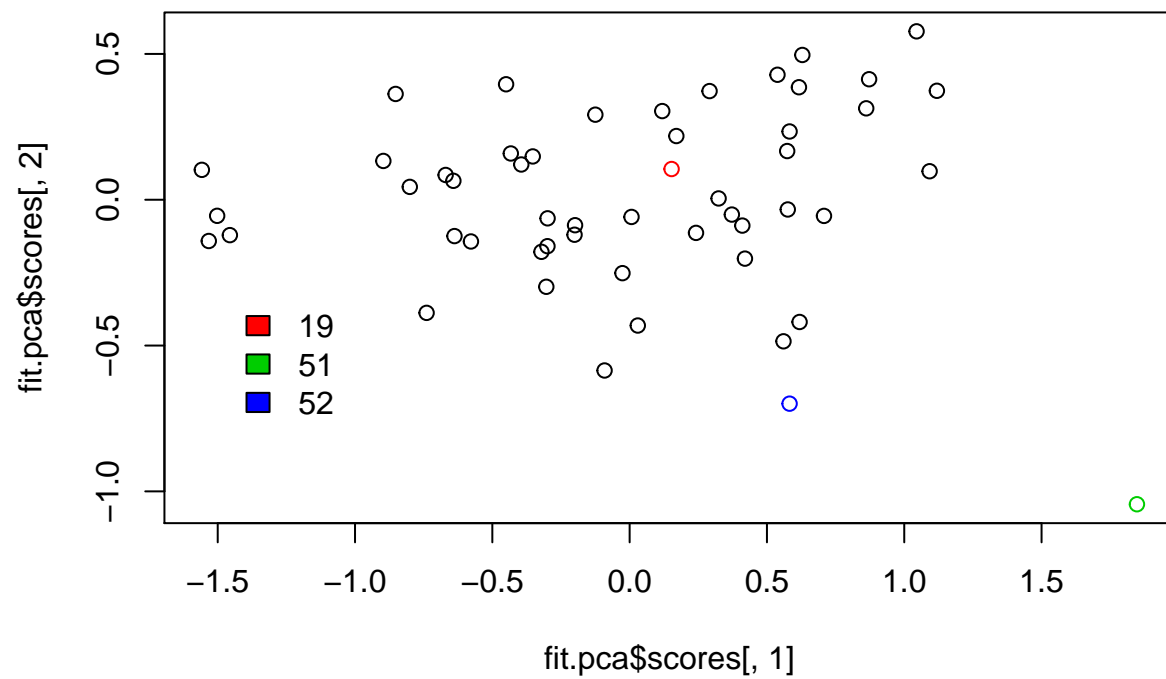
3 of 3



```
## [1] 19 51 52
```

Let's explore it out via principal component analysis (PCA).

```
colors <- rep(1,52)
colors[c(19,51,52)] <- 2:4
fit.pca <- princomp(iris.sub)
plot(fit.pca$scores[,1], fit.pca$scores[,2], col=colors)
legend(-1.5, -.3, c(19,51,52), 2:4, bty="n")
```

Looking at the first two principal components, the observation seems very standard.

Let's look at the last two components.

```
colors <- rep(1,52)
colors[c(19,51,52)] <- 2:4
fit.pca <- princomp(iris.sub)
plot(fit.pca$scores[,3], fit.pca$scores[,4], col=colors)
legend(-.6, 0, c(19,51,52), 2:4, bty="n")
```

