Anomaly Detection via Depth Quantile Functions

A vignette for using the (adative) 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 a subset of the data , $n = 52$, each consisting of 4 variables. The first 50 all come from the same species (Versicolor, these are observations 51-100 in the original data set). Two additonal observations from a different species (Virginica species, observations 101 and 102 in the original data set) are also included. These observations, 51 and 52 in the current context, constitute anomalies.

```
data(iris)
iris.sub <- iris[51:102,1:4]
```

An appropriate value of *g.scale* (a scaling factor on the base distribution G_{ij}) 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 *adaptive=FALSE* when calling *dqf.outlier*. A uniform base distribution can also be used via the argument *G="unif"*.

```
fit.dqf <- dqf.outlier(iris.sub, g.scale=6)
```

We use the functionality in *dqf.explore* to highlight these known anomalies (51 and 52) via the *show* argument.

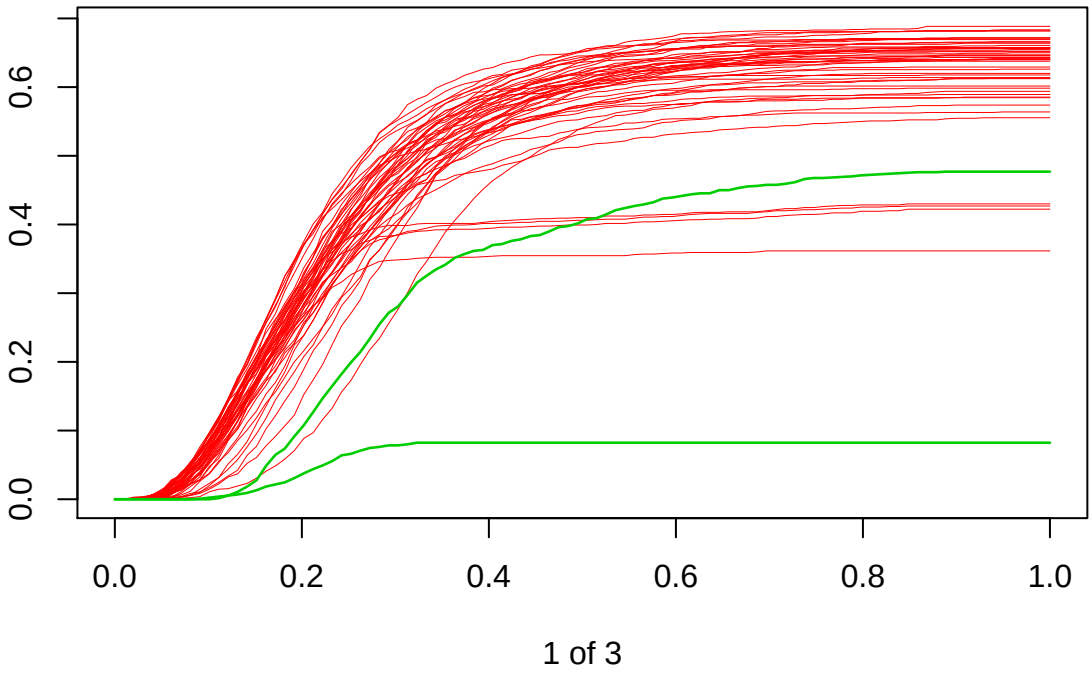
Rather than specifying the indicies of outlying observations as we do here, the code presents a sequence of three *interactive* graphs, where the user can click on any interesting functions.

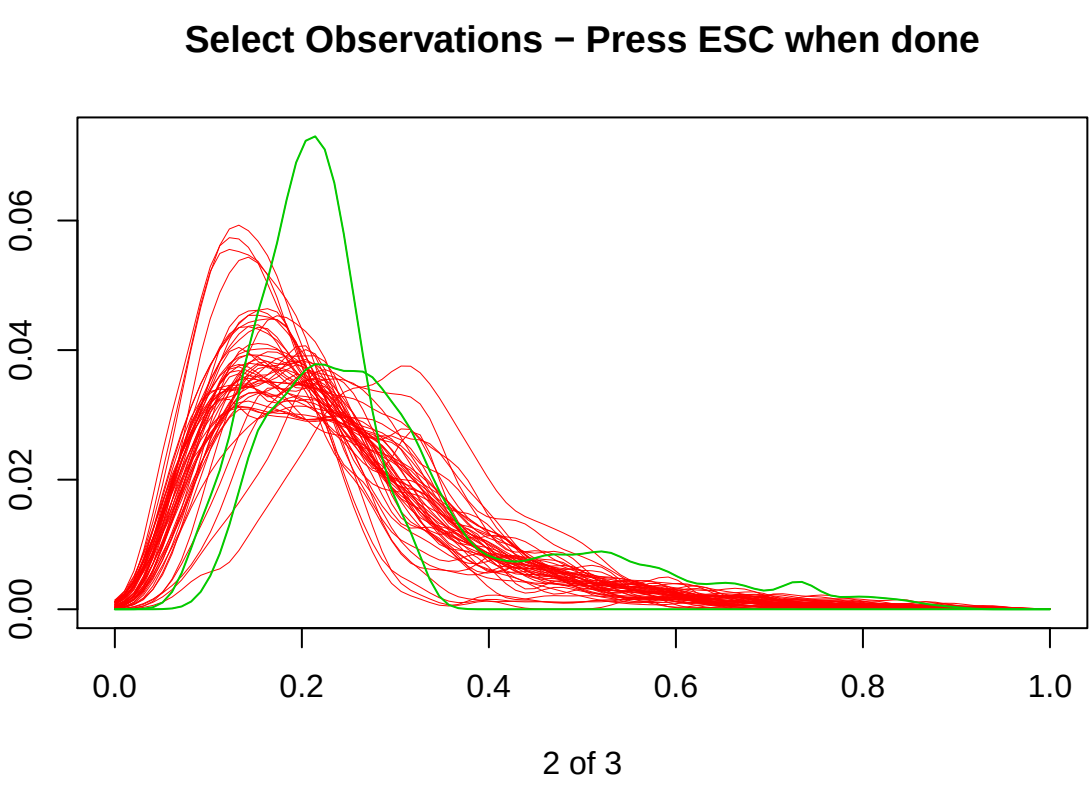
The three graphs are the averaged DQFs $\bar{q}_i(\delta)$, their derivatives after normalization $\frac{d}{d\delta}\frac{\bar{q}_i(\delta)}{\bar{q}_i(1)}$ and the normalized (averaged) DQF itself $\tilde{q}_i(\delta) = \frac{\bar{q}_i(\delta)}{\bar{q}_i(1)}$.

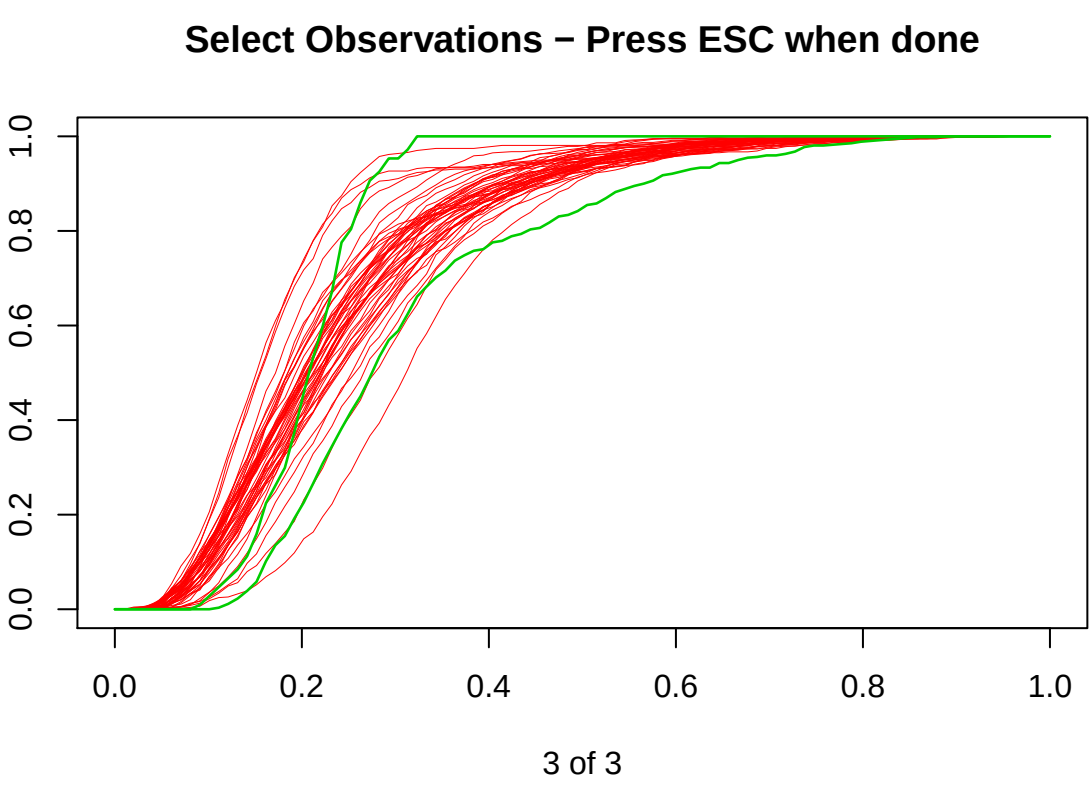
The functions corresponding to those observations will be highlighted in subsequent graphs, with their indices given in the final graph. The interactive graphs are shown at the second of the three angles (default of 30, 45 and 60 degrees). For high dimensional data, larger angles may be reqiured to obtain meaningful visualizations. These can be set using the *angle* argument to *dqf.outlier*.

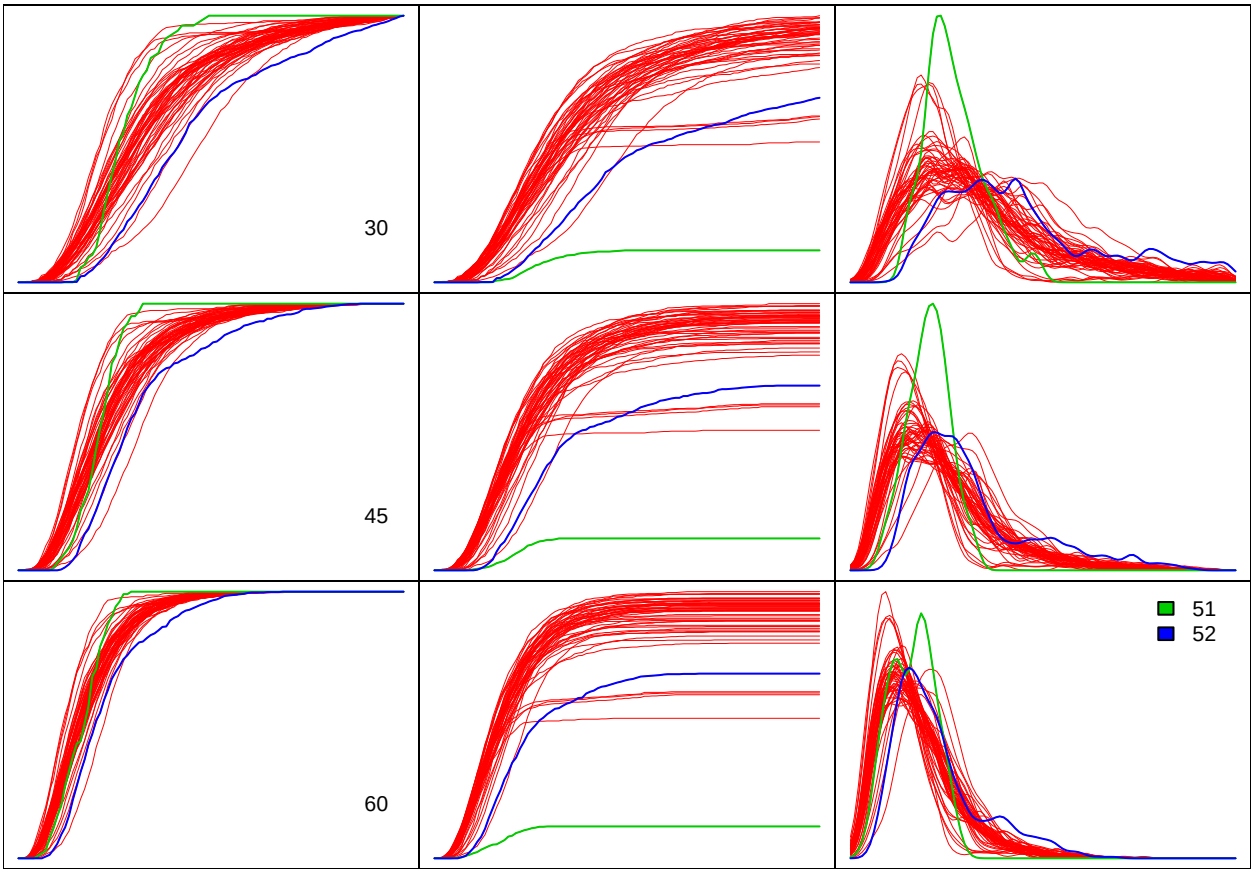
After you are finished selecting interesting functions, press the *ESC* key as directed to procede to the next graph.

```
dqf.explore(fit.dqf, show=51:52)  #visualize the DQFs highlighting the known anomalies
```


Select Observations − Press ESC when done
0.0
0.2
0.4
0.6
0.0
0.2
0.4
0.6
0.8
1.0
1 of 3


Select Observations − Press ESC when done
0.00
0.02
0.04
0.06
0.0
0.2
0.4
0.6
0.8
1.0
2 of 3

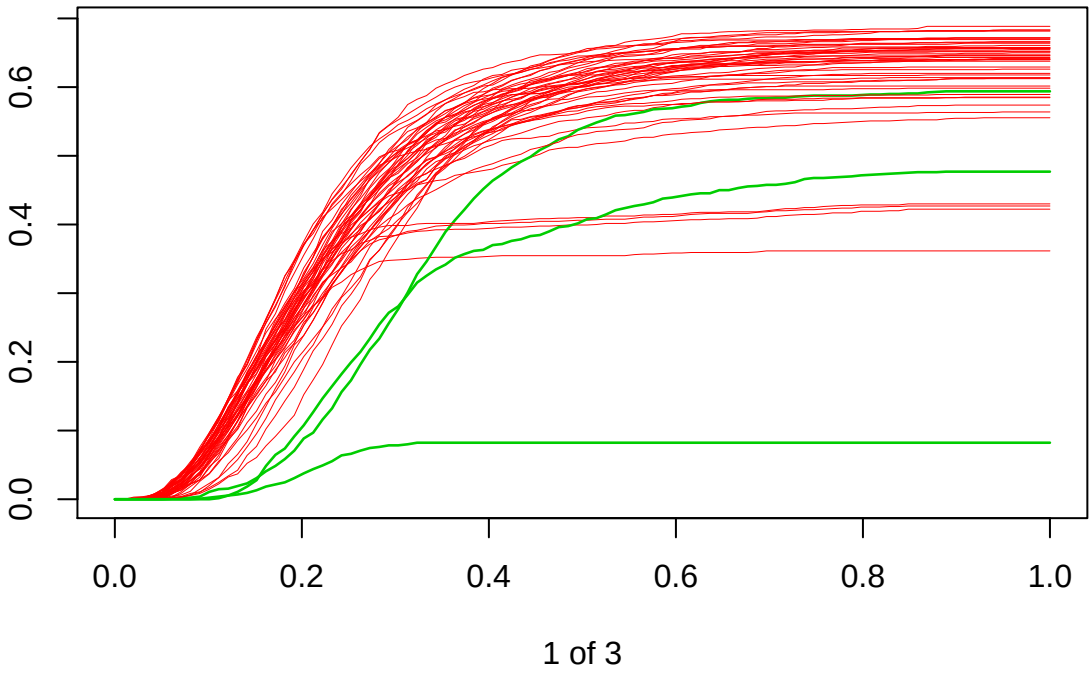

Select Observations − Press ESC when done
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
3 of 3

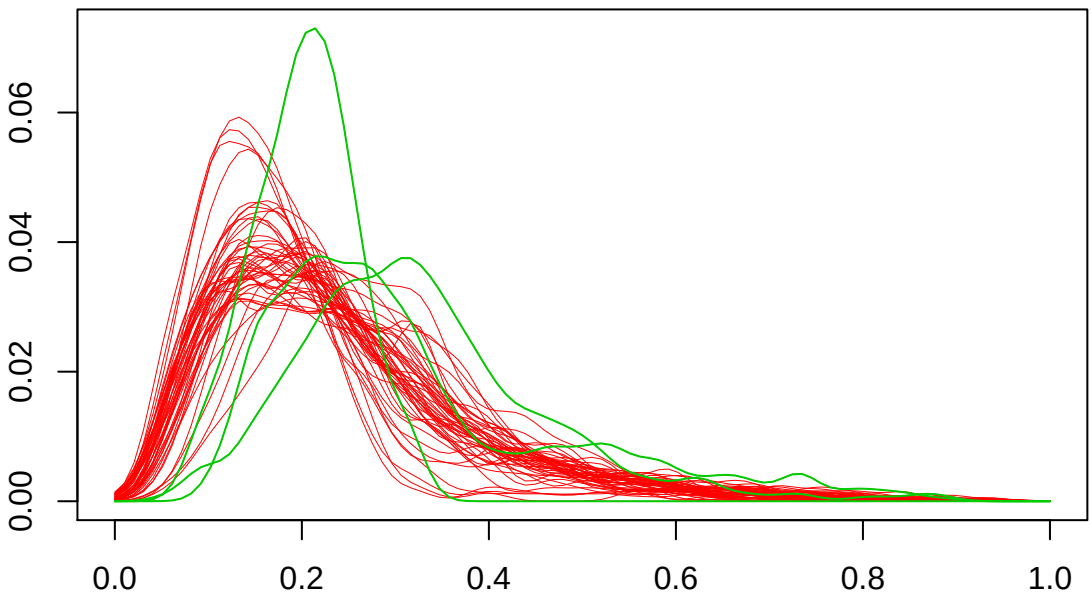


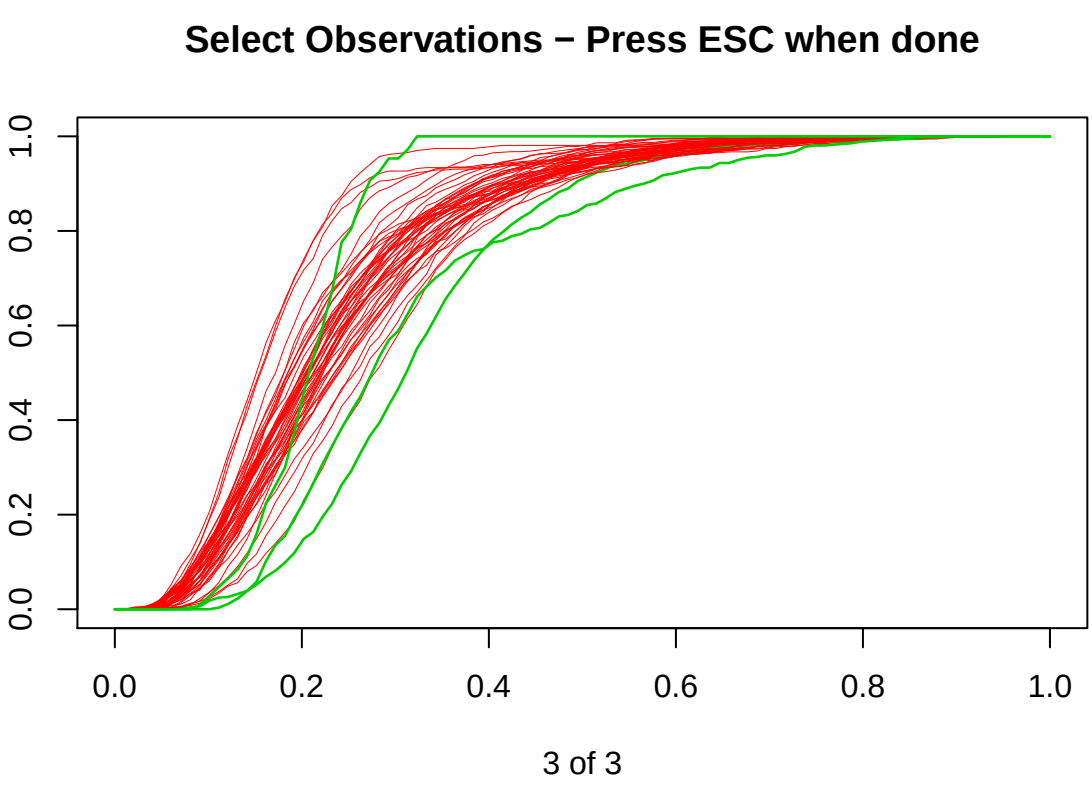

```
## [1] 51 52
```

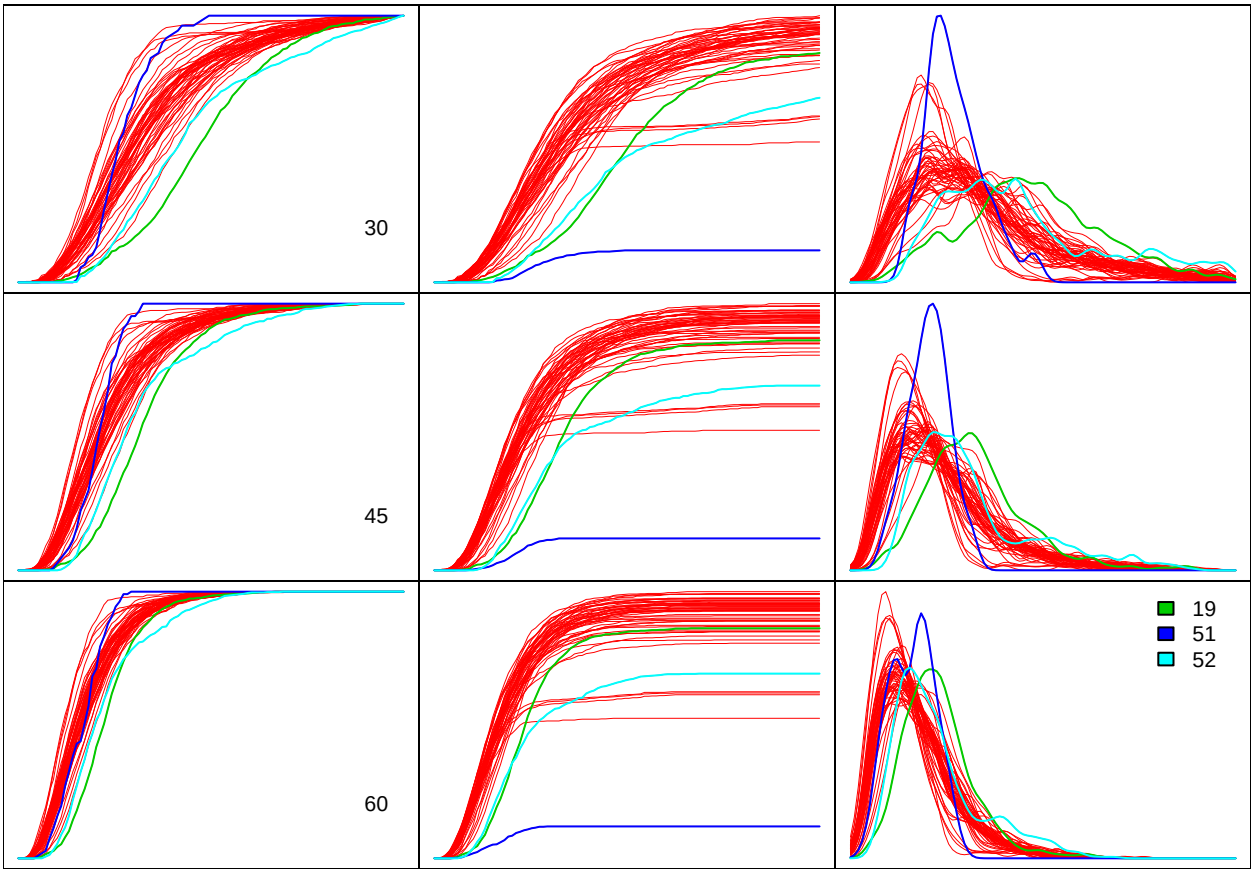
We might note that there is at least one additional interesting observation detected by the method. This turns out to observation 19 in our subsetted data (69 in the iris data set).

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


Select Observations − Press ESC when done
0.0
0.2
0.4
0.6
0.0
0.2
0.4
0.6
0.8
1.0
1 of 3


Select Observations − Press ESC when done
0.00
0.02
0.04
0.06
0.0
0.2
0.4
0.6
0.8
1.0
2 of 3

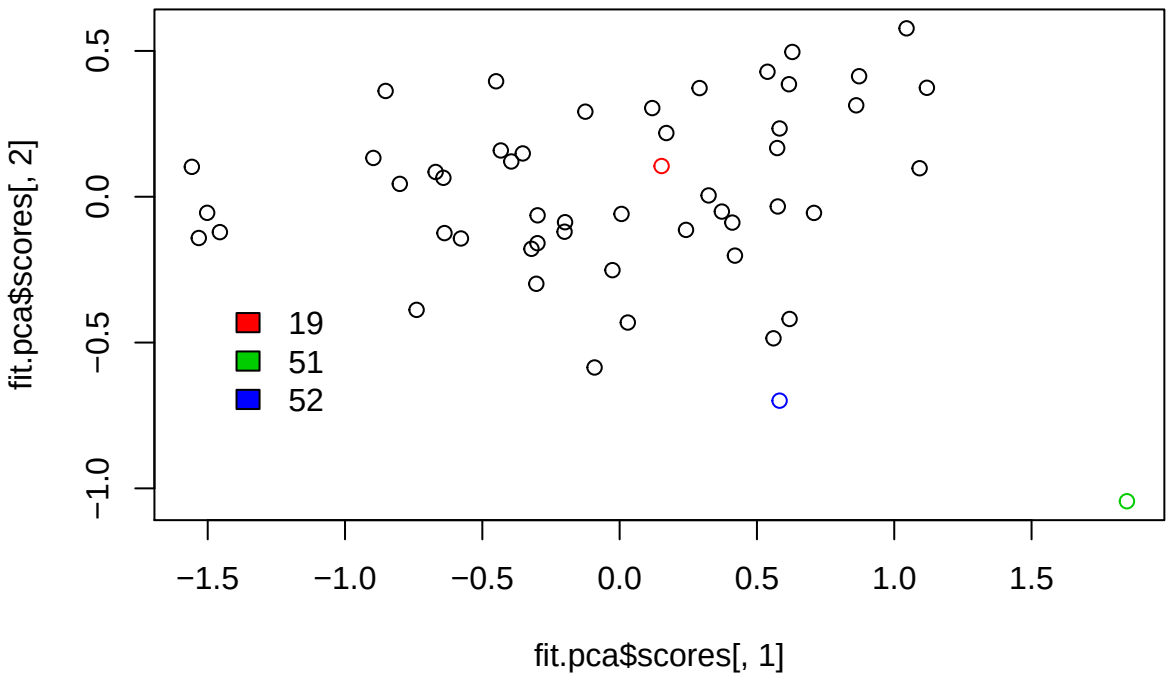

Select Observations − Press ESC when done
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
3 of 3




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

Let's explore these observations via principal component analysis (PCA).

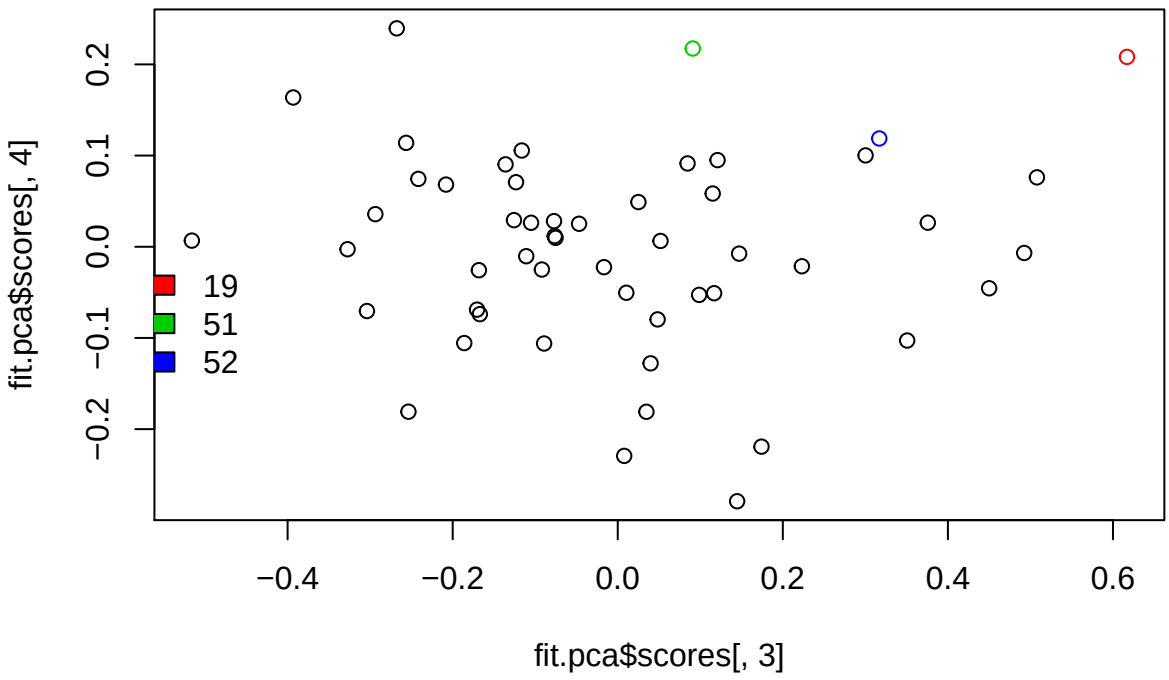
```
colors <- rep(1,52)
colors[c(19,51,52)] <- 2:4
fit.pca <- princomp(iris.sub)
plot(fit.pca$scores[,1], fit.pca$scores[,2], col=colors)
legend(-1.5, -.3, c(19,51,52), 2:4, bty="n")
```



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")
```



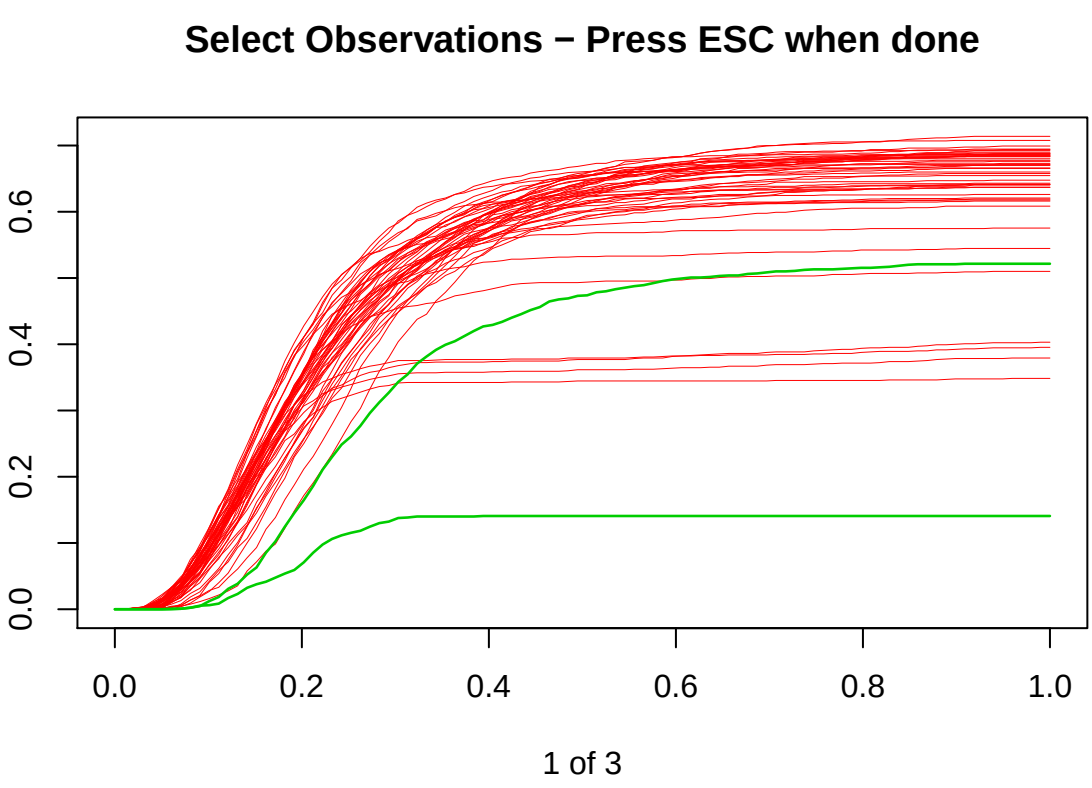
We note that observation 52 does not stand out with respect to the PCA plots, while for the DQF plots, it is very low for small δ values, a telltale sign of being an anomaly. This is especially apparent by considering the plot of the derivatives $\frac{d}{d\delta}\tilde{q}_i(\delta)$.

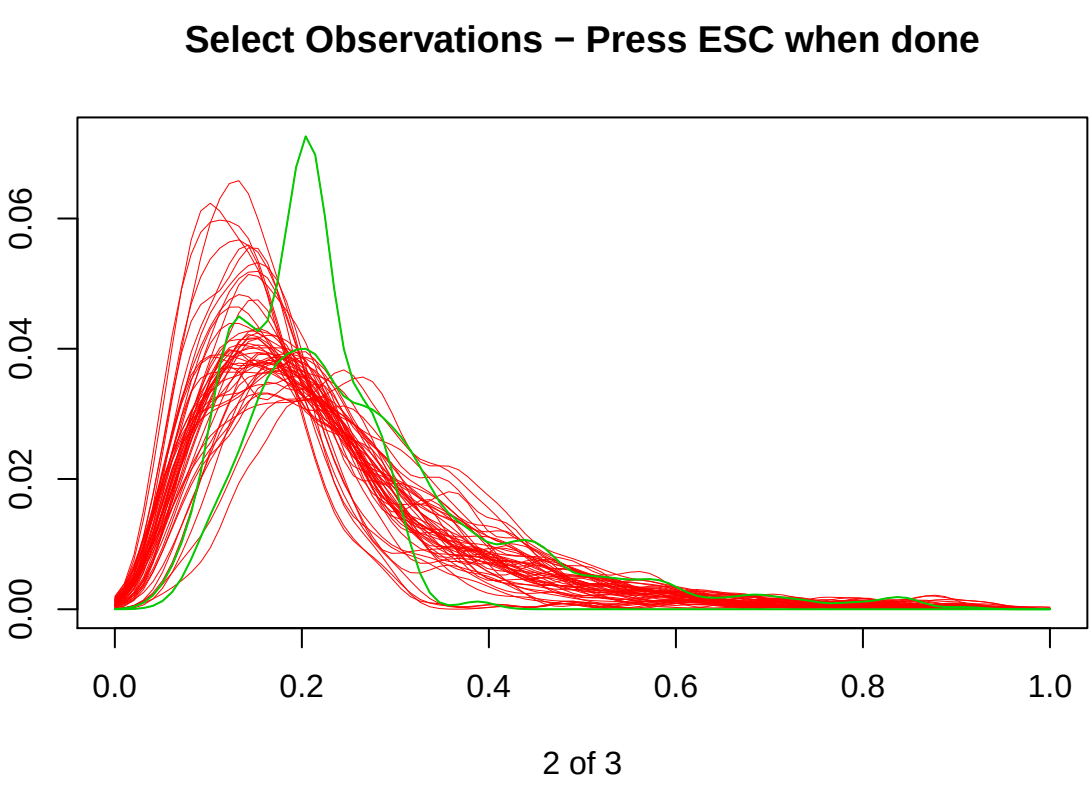
dqf.outlier from the Gram matrix

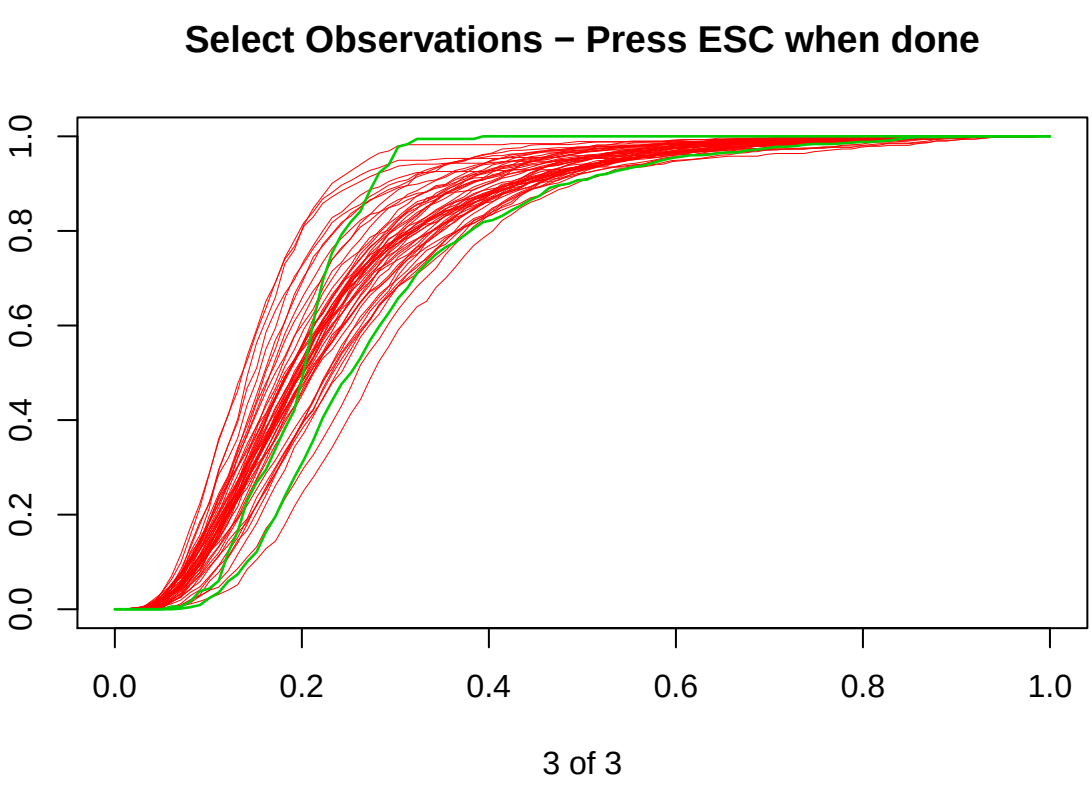
For non-Euclidean data, the DQF methodology can still create a meaningful visualization of the data, assuming that a notion of inner product on exists, typically via a kernel function. *dqf.outlier* alternatively accepts a Gram-matrix.

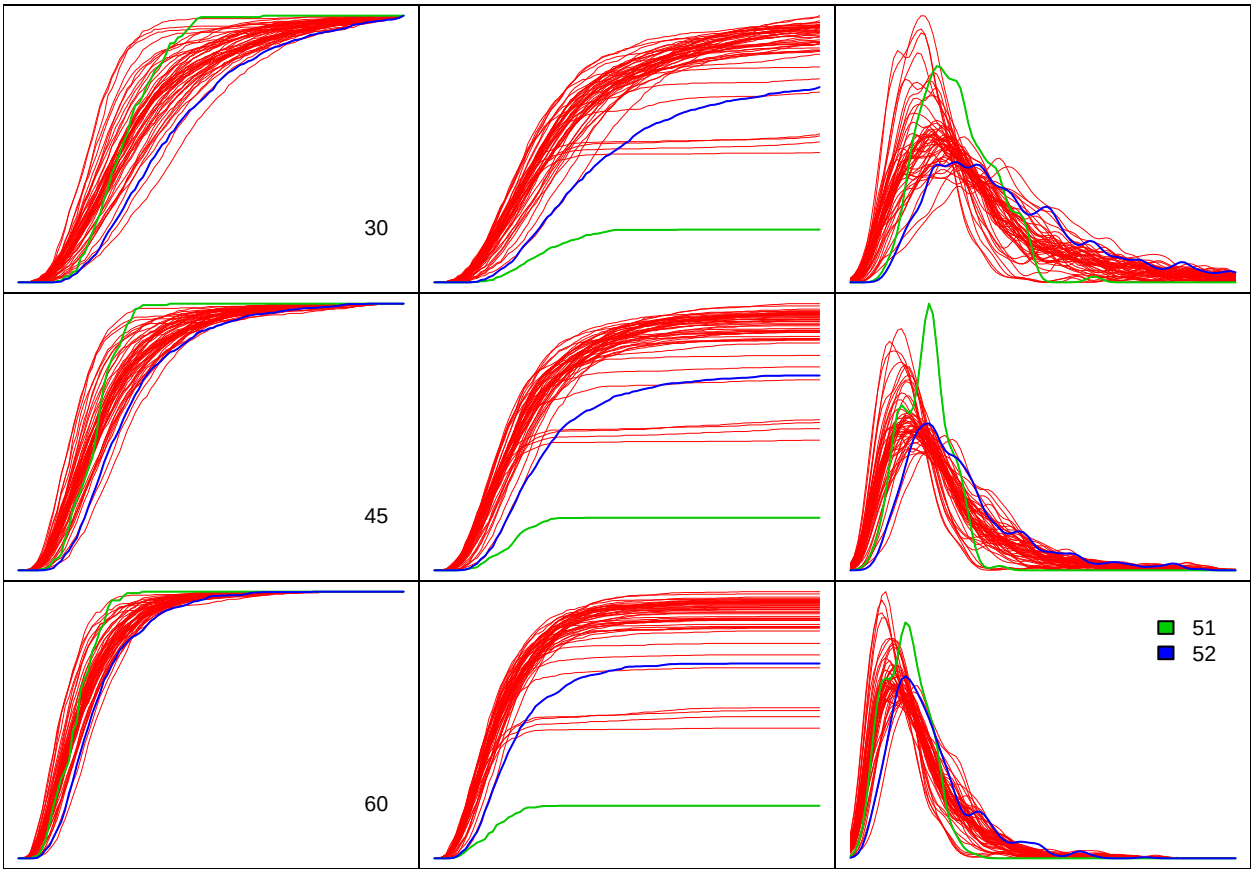
Here, we reproduce the above analysis

```
gram <- matrix(0, nrow=52, ncol=52)
for (i in 1:52) {
  for (j in i:52) {
    gram[i,j] <- sum(iris.sub[i,] * iris.sub[j,]) -> gram[j,i]
  }
}
fit.dqf.gram <- dqf.outlier(gram.mat = gram, g.scale=6)
dqf.explore(fit.dqf.gram, c(51,52))
```


Select Observations − Press ESC when done
0.0
0.2
0.4
0.6
0.0
0.2
0.4
0.6
0.8
1.0
1 of 3


Select Observations − Press ESC when done
0.00
0.02
0.04
0.06
0.0
0.2
0.4
0.6
0.8
1.0
2 of 3


Select Observations − Press ESC when done
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
3 of 3




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