Data Quality: Lecture 7

Philipp Drebes

16.04.2023

Task 1

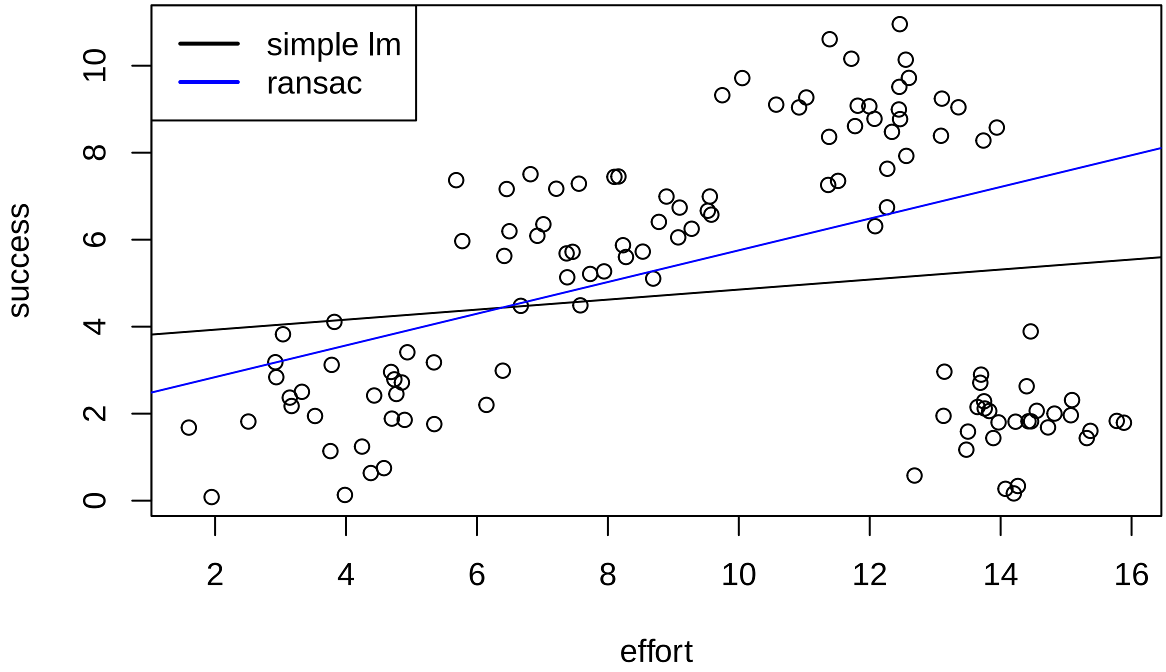
The data set *learning* is given. Run the RANSAC algorithm to determine, whether there are outliers.

```
ransac <- function(data, n, k, t, d) {
  iterations <- 0
  bestfit <- NULL
  besterr <- 1e5
  while (iterations < k) {
    maybeinliers <- sample(nrow(data), n)
    maybemodel <- lm(y ~ x, data = data, subset = maybeinliers)
    alsoinliers <- NULL
    for (point in setdiff(1:nrow(data), maybeinliers)) {
      if (abs(maybemodel$coefficients[2]*data[point, 1] - data[point, 2] +
             maybemodel$coefficients[1])/(sqrt(maybemodel$coefficients[2] + 1)) < t)
        alsoinliers <- c(alsoinliers, point)
    }
    if (length(alsoinliers) > d) {
      bettermodel <- lm(y ~ x, data = data, subset = c(maybeinliers, alsoinliers))
      thiserr <- summary(bettermodel)$sigma
      if (thiserr < besterr) {
        bestfit <- bettermodel
        besterr <- thiserr
      }
    }
    iterations <- iterations + 1
  }
  bestfit
}
```

```
library(readxl)

learning <- read_xlsx('data/learning.xlsx')
head(learning)
```

```
## # A tibble: 6 x 2
##   effort success
##    <dbl>   <dbl>
## 1   2.93   2.84
## 2   3.82   4.11
## 3   4.69   2.96
## 4   4.38   0.635
## 5   4.58   0.748
## 6   6.39   2.99
```

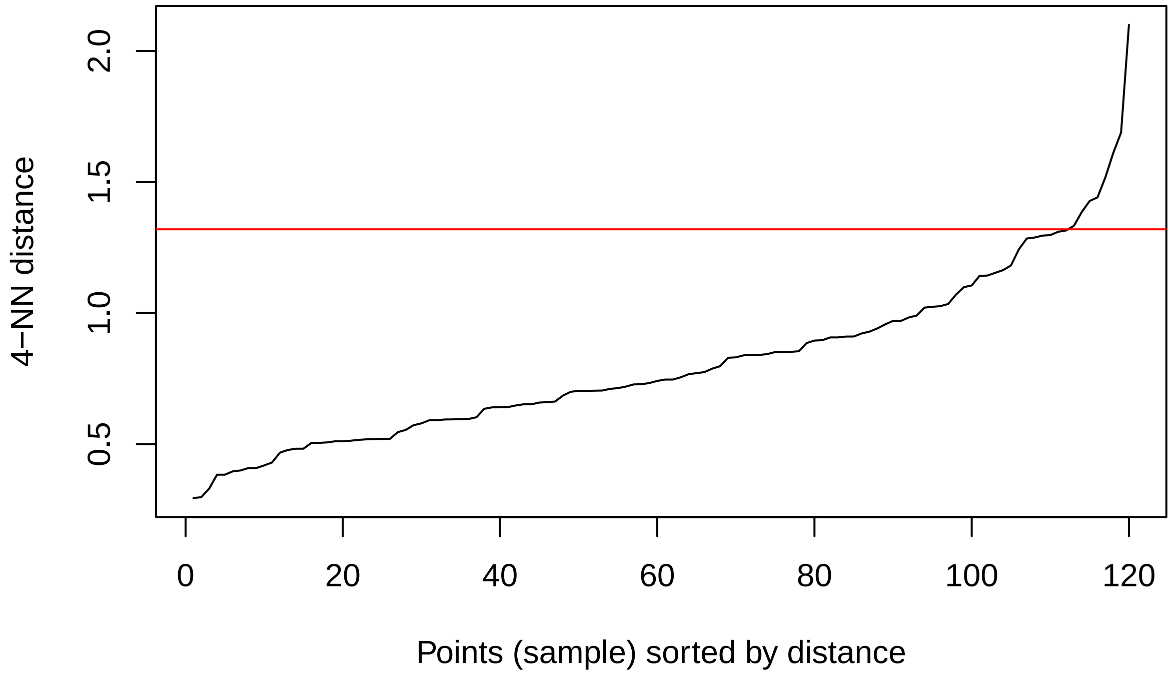
```
plot(success ~ effort, data = learning)
abline(lm(success ~ effort, data = learning))

learning$x <- learning$effort
learning$y <- learning$success
learning <- subset(learning, select = c(x, y))

set.seed(998899)
abline(ransac(learning, n = 10, k = 10, t = 0.5, d = 10), col = "blue")
legend(x = 'topleft', legend = c('simple lm', 'ransac'), col = c('black', 'blue'), lwd = 2)
```



Task 2

The data set *learning* is given.
Run the DBSCAN algorithm to determine, whether there are outliers.

```
library(dbscan)

kNNdistplot(learning,  k = 4)
abline(h = 1.32, col = 'red', lwd = 1)
```

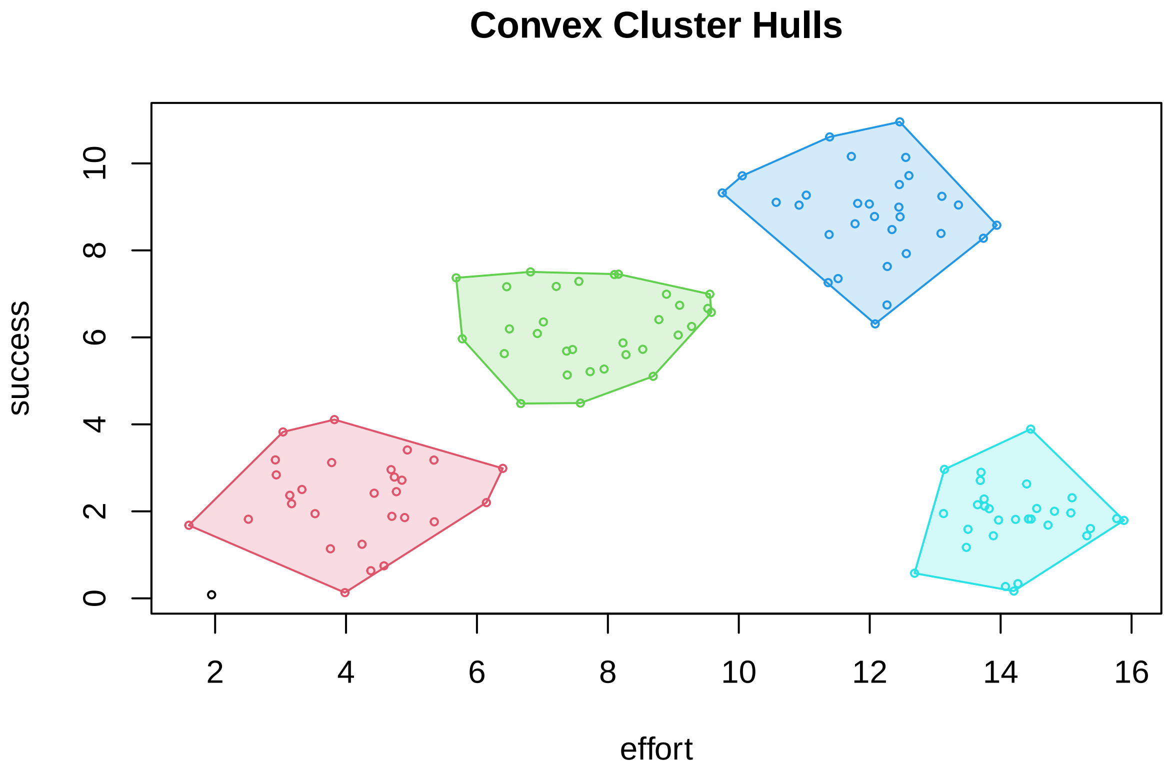


We will use 1.32 as the size of the epsilon neighborhood, as the kNN-dist-plot curve has a steeper angle at that value.

```
dbscanResult <- dbscan(learning, eps= 1.32, minPts=4)
dbscanResult
```

```
## DBSCAN clustering for 120 objects.
## Parameters: eps = 1.32, minPts = 4
## Using euclidean distances and borderpoints = TRUE
## The clustering contains 4 cluster(s) and 1 noise points.
##
##  0  1  2  3  4
##  1 28 31 30 30
##
## Available fields: cluster, eps, minPts, dist, borderPoints
```

```
hullplot(learning, dbscanResult, xlab = 'effort', ylab = 'success')
```



We see one outlier in the lower left corner.