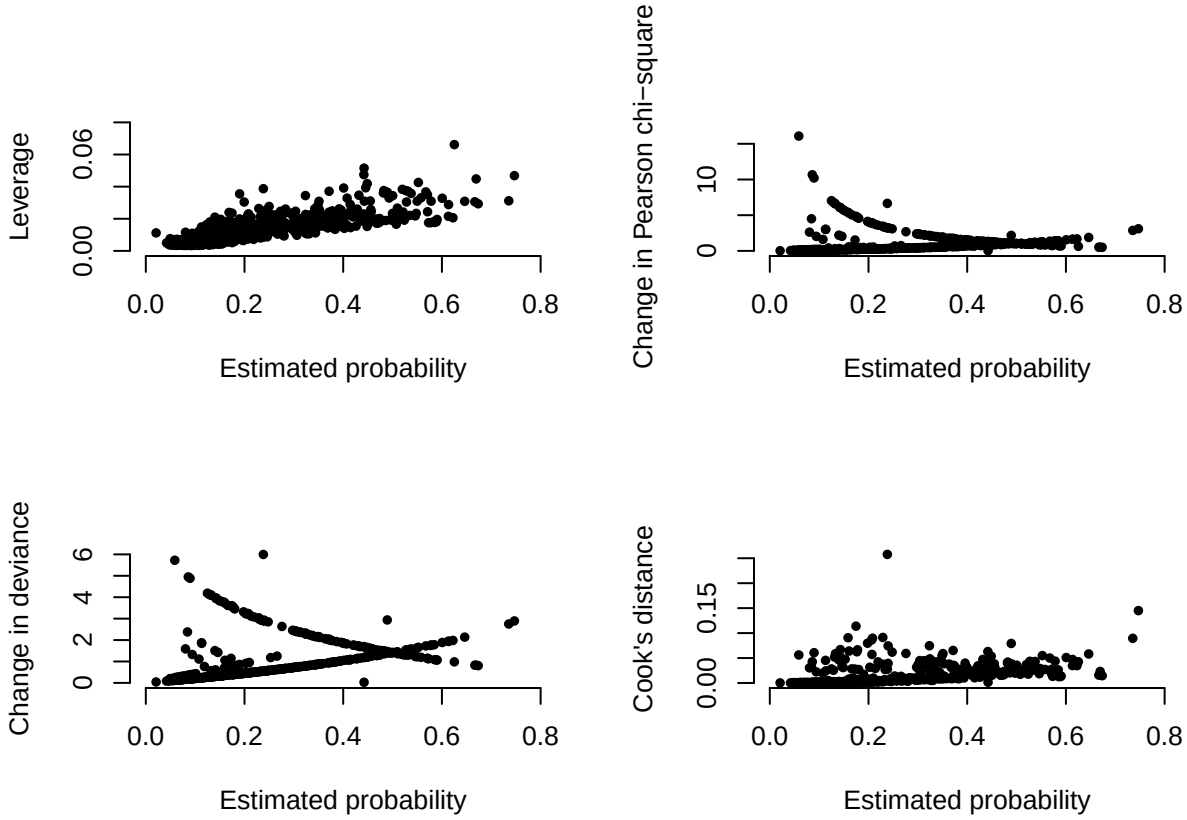
Inconsistencies in the reproduction of figures 5.12 and 5.15 of Hosmer et al. 2013

Johann Popp

4 November 2017

Hosmer et al[1] have suggested several graphics to identify and investigate extreme and influencing covariate patterns in logistic regression models. These plots where reproduced using the computer software R[2], taking data from package aplore3[3] and using package epiR[4] for the calculation of statistics based on covariate patterns rather then single data rows.



This works fine for the graphs showings change of Pearson chi-square and change of deviance but the plots of leverage vs. fitted values and Cook's distance vs. fitted values differ substantially from figures 5.12 and 5.15 of the book.

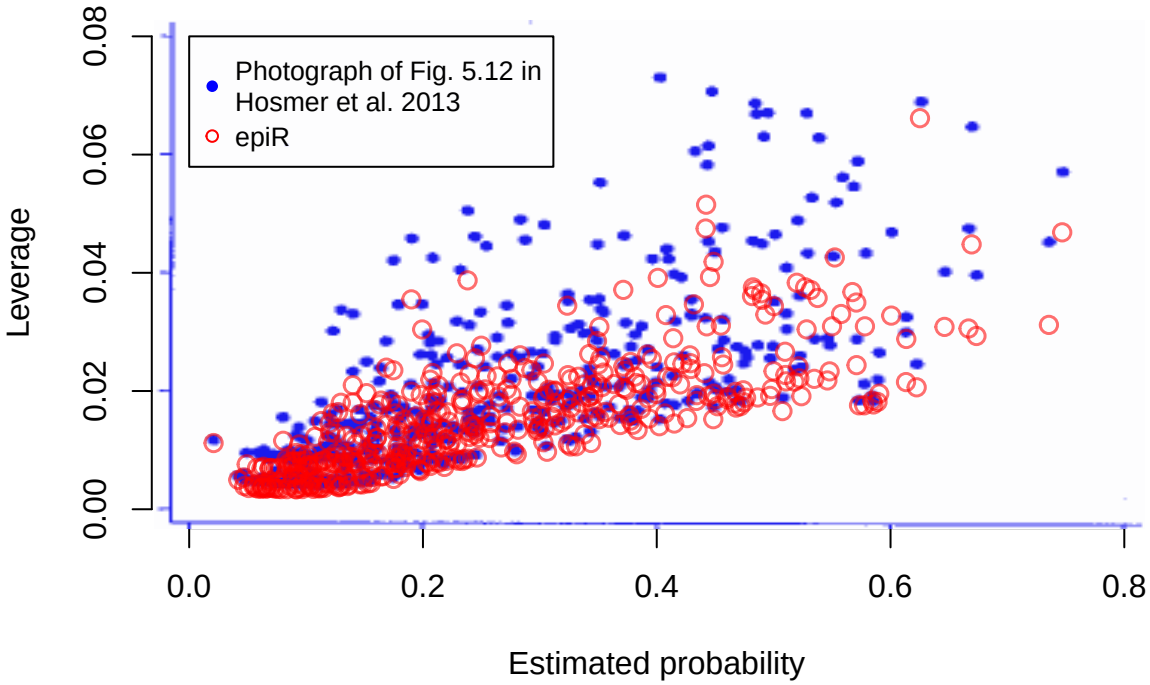
[1]Hosmer, David W., Stanley Lemeshow, und Rodney X. Sturdivant. Applied logistic regression. 3rd Ed. Wiley series in probability and statistics. Hoboken, NJ: Wiley, 2013, p 186 ff.

[2]R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

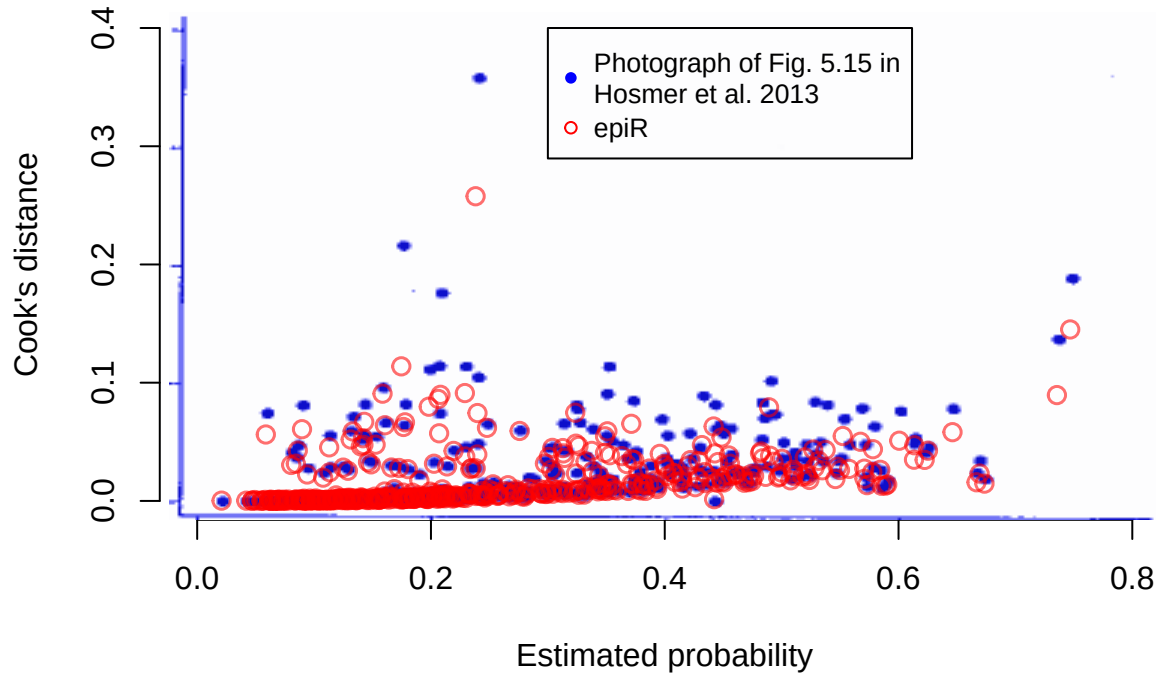
[3]Luca Braglia (2016). aplore3: Datasets from Hosmer, Lemeshow and Sturdivant, "Applied Logistic Regression" (3rd Ed., 2013). R package version 0.9. https://CRAN.R-project.org/package=aplore3

[4]Mark Stevenson with contributions from Telmo Nunes, Cord Heuer, Jonathon Marshall, Javier Sanchez, Ron Thornton, Jeno Reiczigel, Jim Robison-Cox, Paola Sebastiani, Peter Solymos, Kazuki Yoshida, Geoff Jones, Sarah Pirikahu, Simon Firestone and Ryan Kyle. (2017). epiR: Tools for the Analysis of Epidemiological Data. R package version 0.9-87. https://CRAN.R-project.org/package=epiR

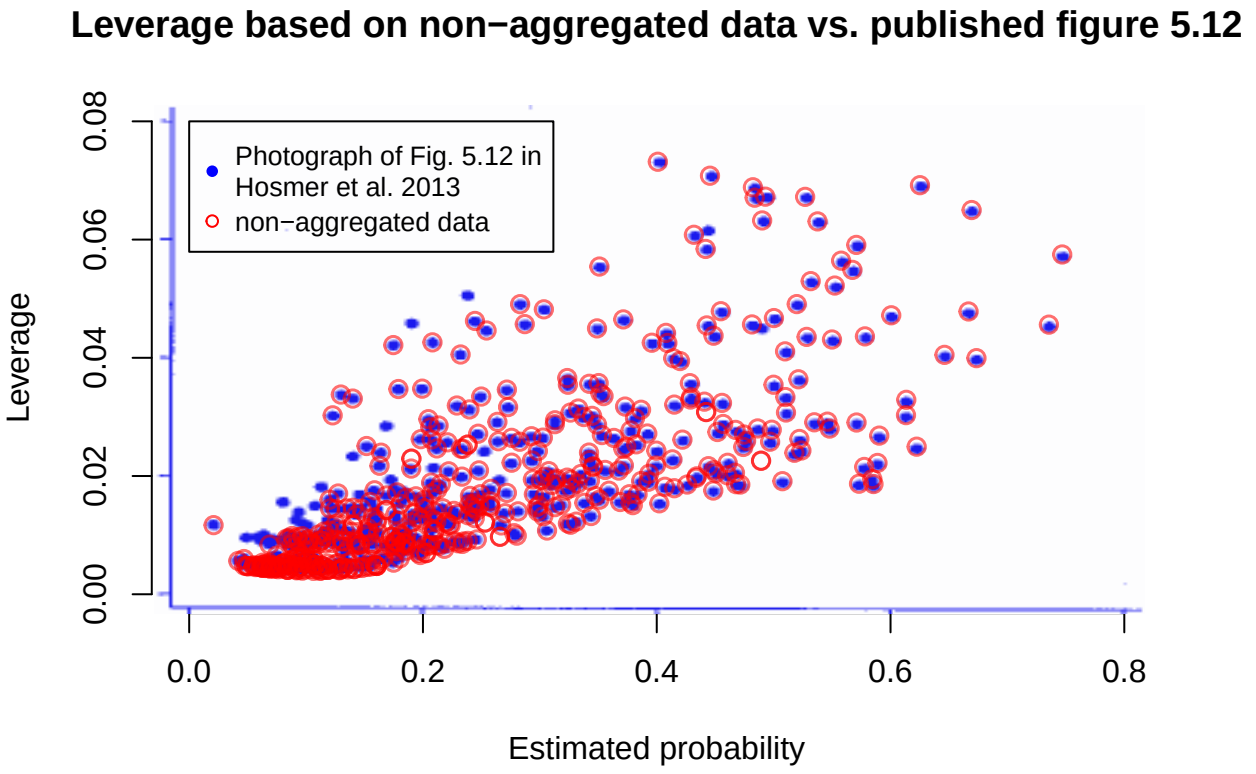
Leverage based on package epiR vs. published figure 5.12



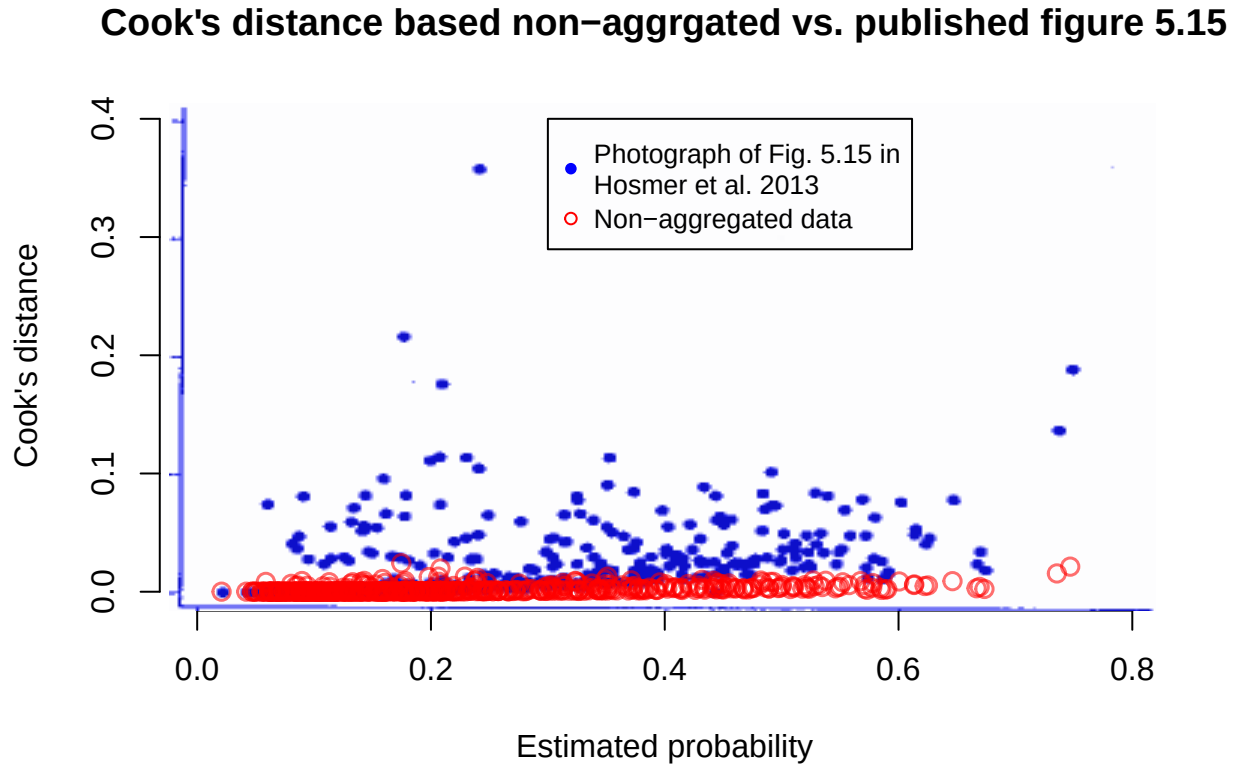
Cook's distance based on package epiR vs. published figure 5.15



With leverages based on non-aggregated n-statistics instead of covariate patterns (m-statistics) the plot can be roughly reproduced, but there are still some points that do not match.



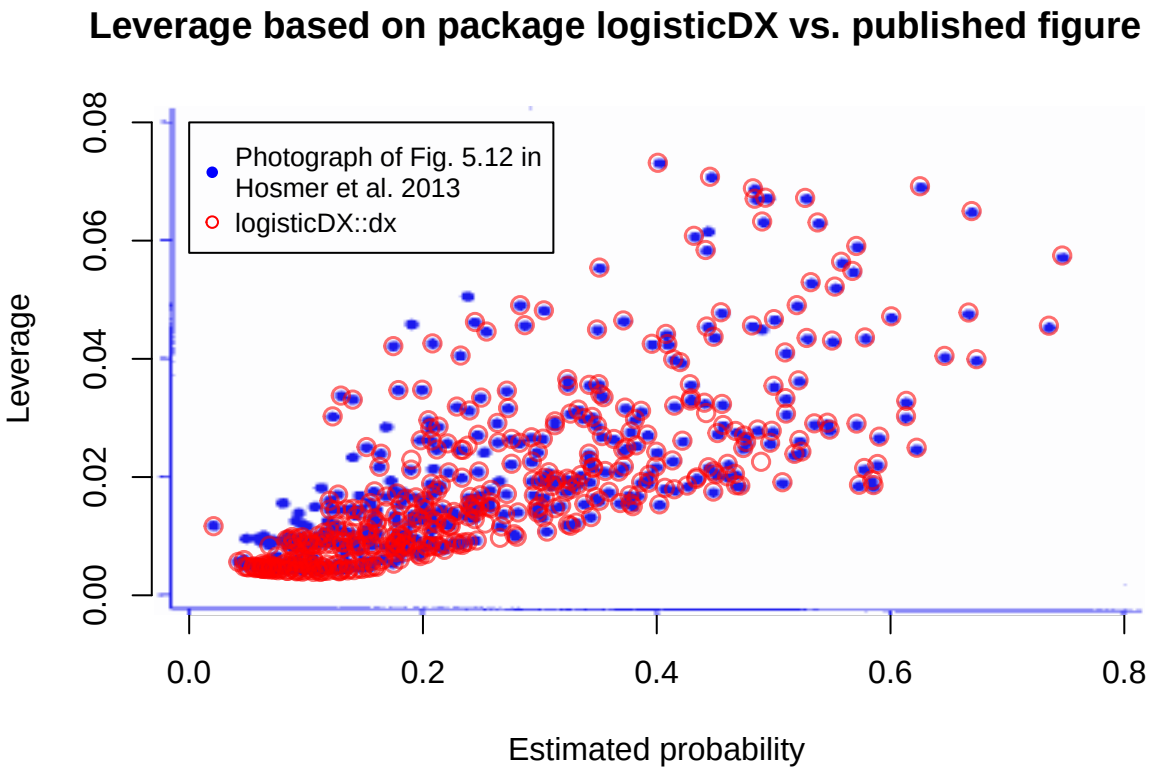
But this remedy does not work for Cook's distance. The values based on n-statistics are completely different from those printed in the book and those calculated by epiR in magnitude and pattern.

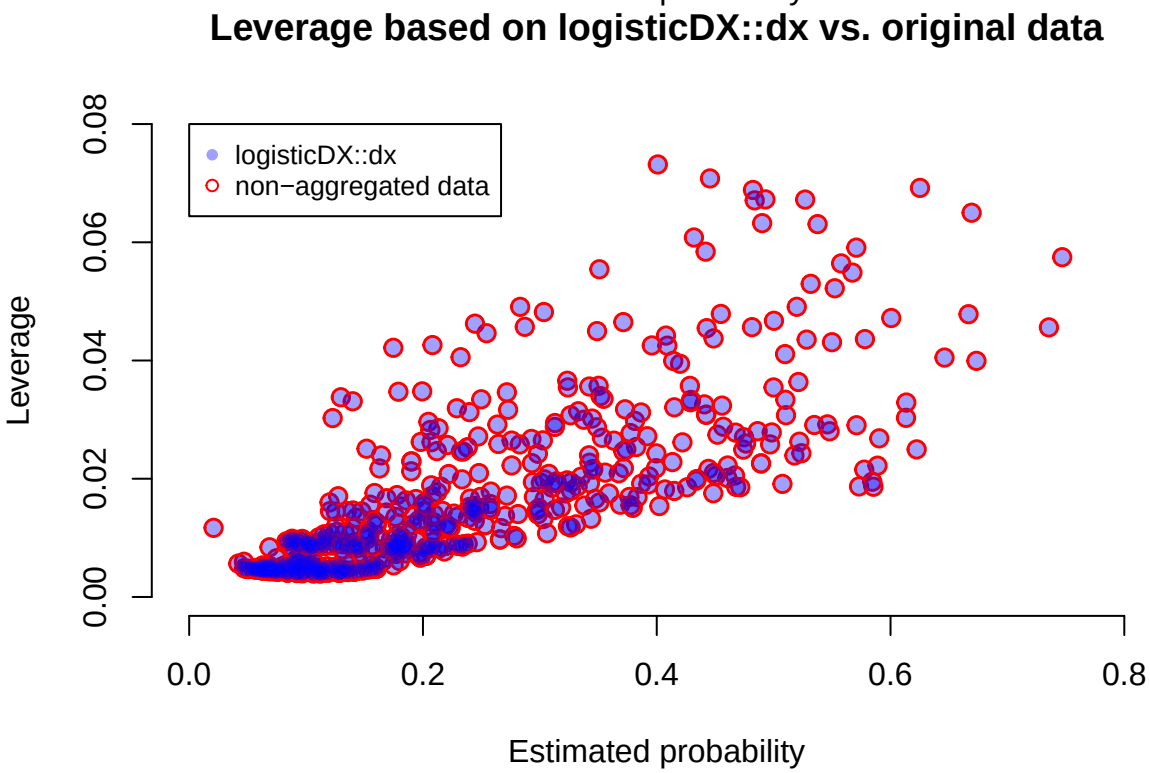


I tried out the newer package logisticDx[5] to calculate the diagnostic statistics. This packackage specializes on

[5]Chris Dardis (2015). LogisticDx: Diagnostic Tests for Models with a Binomial Response. R package version 0.2. https:

diagnostic tests for regression models with binomial response and is explicitely based on the book of Hosmer et al. albeit in its second edition from 2000.



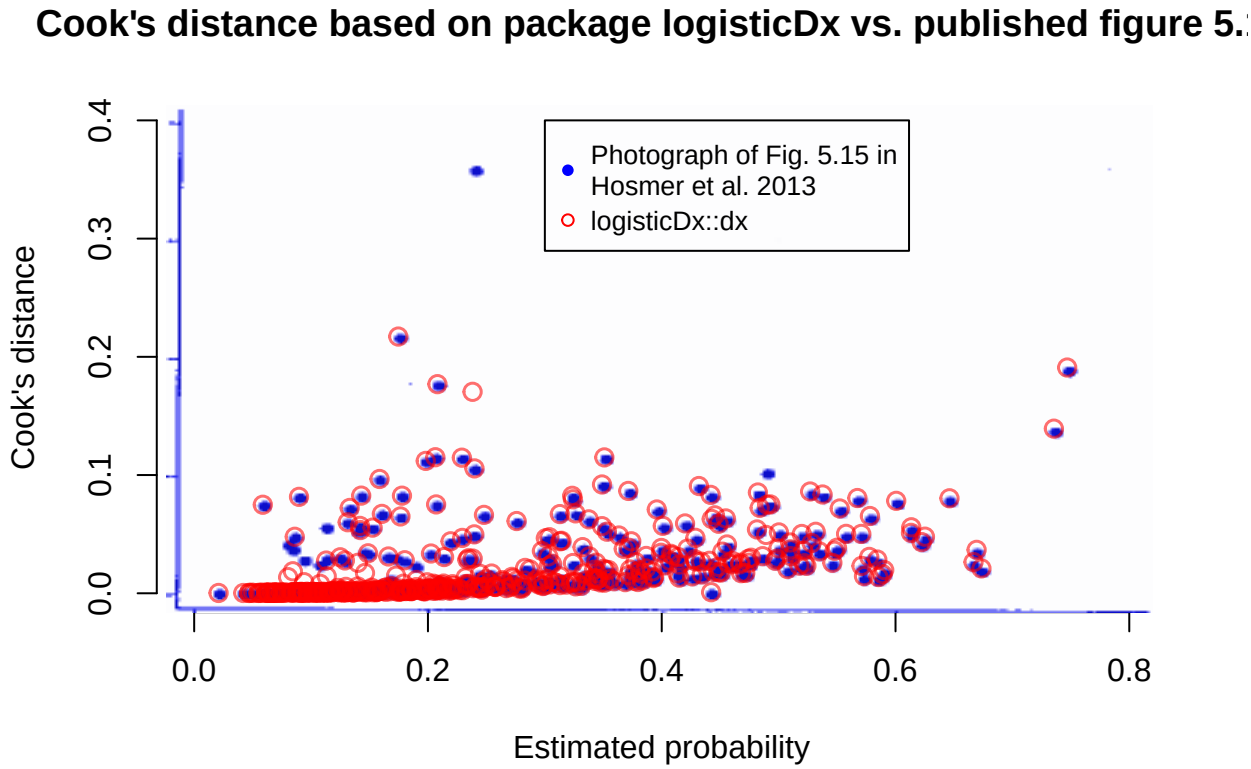


You can see, that the leverages of logisticDX::dx are exactly the same as those calculated from non-aggregated data (n-statistics). We have seen bevore that theese are matching figure 5.12 quite well but not with all the

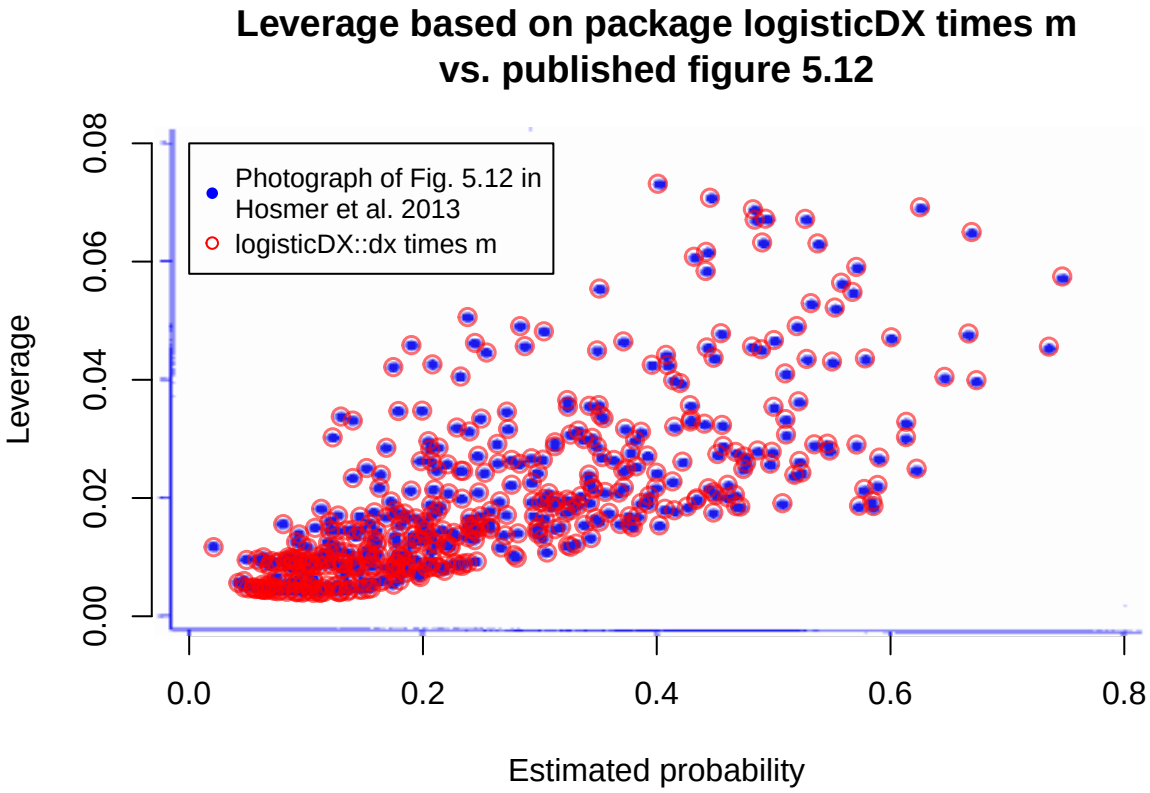
//CRAN.R-project.org/package=LogisticDx

points.

When they are calculated with logisticDX::dx, also the values of Cook's distance are fitting well with figure 5.15 despite of some single cases. I guess these are the same as those that do not fit in leverarge.



Let's see what happens if we multiply the leverage times m, the number of cases in the covariate pattern:



This fits perfectly. But the Stata manual[6] gives an other formula:

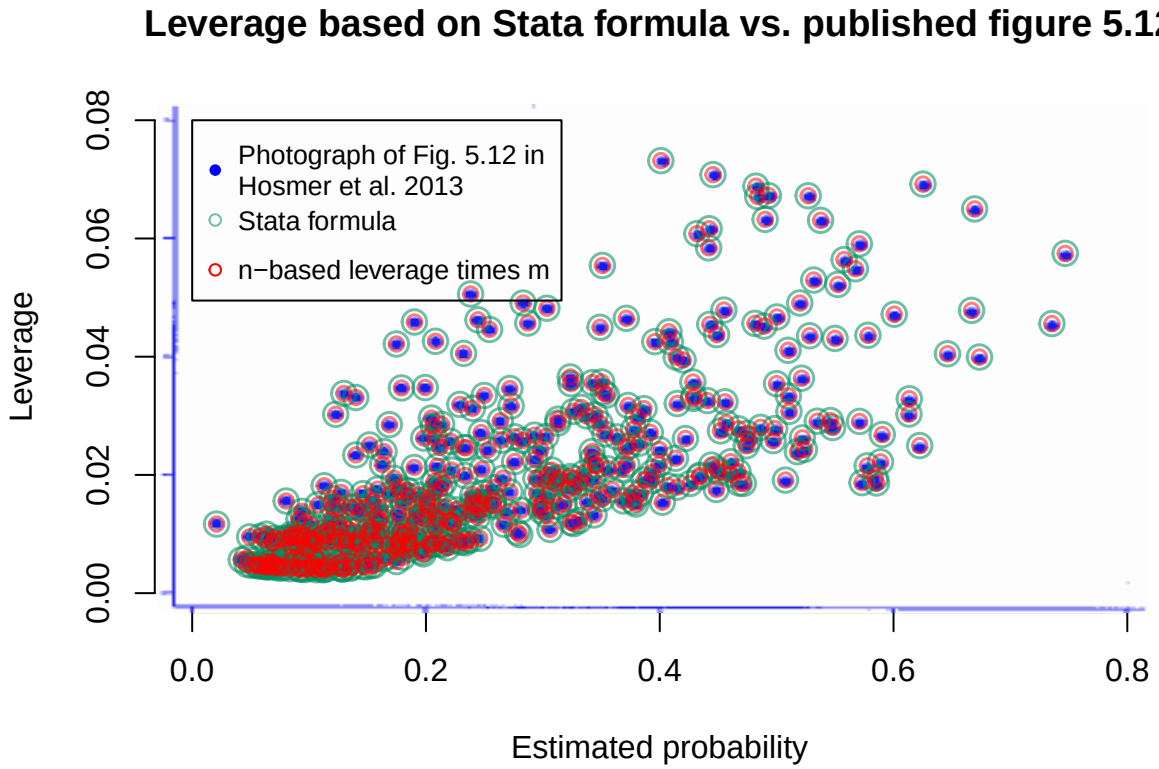
The diagonal $h_j = (XVX')_{jj}$ times $M_j p_j(1-p_j)$

where X is the design matrix for the j covariate patterns and V is the covariance matrix, M is the number of cases in covariate pattern j, p is the estimated outcome probability for covariate pattern j.

```
# Leverage following Stata formula

# Extract the design matrix
X <- model.matrix(model)
# Extract data used by the model
datN <- model$model
# aggregate data to covariate patterns
cp <- unique(datN[,-1])
# Extract an indicator of covariate patterns by pasting the values of all the x-values
cp$cpid <- apply(cp, 1, function(x) paste(x, collapse = ""))
# Get the estimated y-probability for each covariate pattern
cp$fit <- model$fitted.values[rownames(cp)]
# Get n-based hat values for each covariate pattern
cp$H <- hatvalues(model)[rownames(cp)]

# create a covariate pattern indicator for the non-aggregated data (Same method as for covairate patter
cpid <- apply(datN[,-1], 1, function(x) paste(x, collapse = ""))
# calculate group size for each covariate pattern
m <- tapply(datN[,1], cpid, length)[rank(cp$cpid)]
# (tapply sorts the entries by cpid; [rank(cp$cpid)] sorts back to the original order)
```

[6] "Logistic Postestimation - Postestimation Tools for Logistic." In Base Reference Manual, Release 15. College Station, Texas: Stata Press, 2017. https://www.stata.com/bookstore/base-reference-manual/.

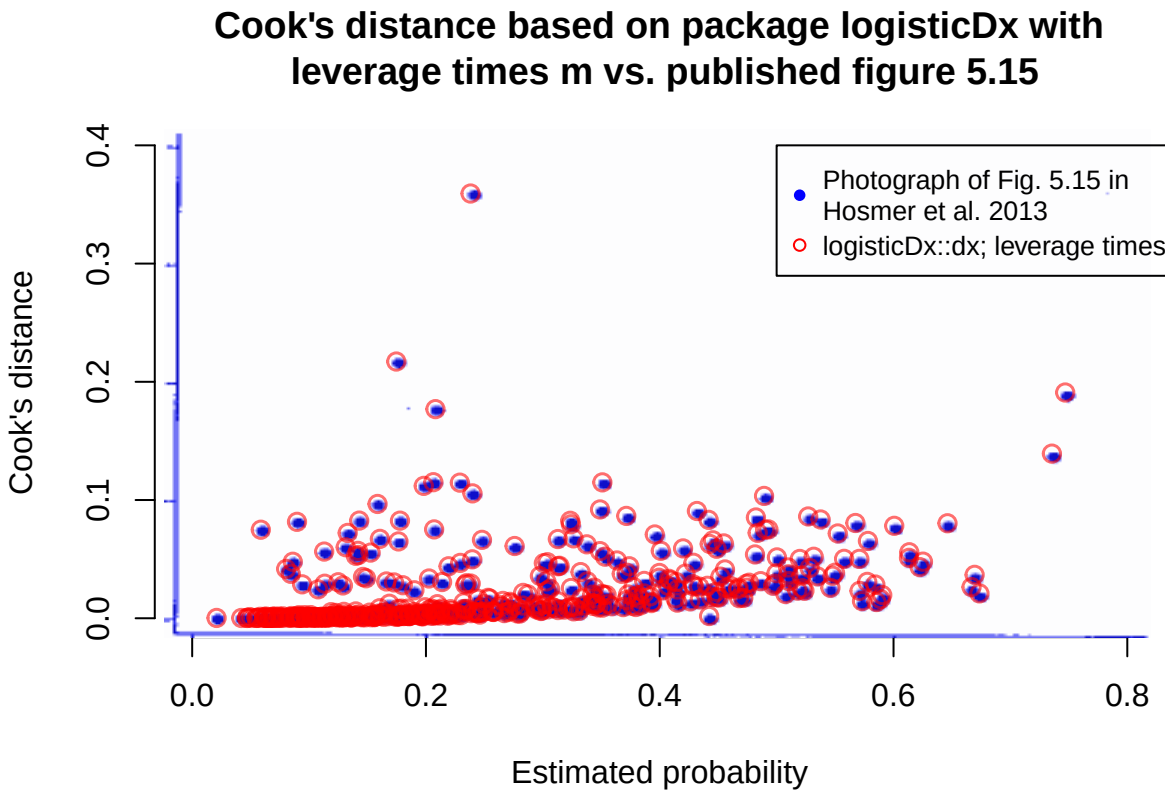
```r
# Aggregate the design matrix to covariate patterns
Xu <- unique(X)
# Extract the variance-covariance matrix of the model
V <- vcov(model)

### This is the formula describet in the Stata reference manual
# Calculate the raw hat values
rawH <- diag(Xu %*% V %*% t(Xu))
# Adjust for covariate pattern
H <- rawH * m * cp$fit * (1-cp$fit)
```



You can see that the result is exactly the same the n-based leverages times the group size for each covariate pattern. I would be delighted if somebody who knows more about matrix algebra could proove this instead of only showing it by example as I did.

I also calculated Cook's distance based on these leverages:

```r
nLev <- ldx$h*ldx$n                    # Leverare times n ("ldx is the dx-obect i have created from the
nsPr <- ldx$Pr / sqrt(1 - nLev)        # New standardized Pearson residuals
nCook <-(nsPr^2 * nLev) / (1 - nLev)   # New Cook's distance
```



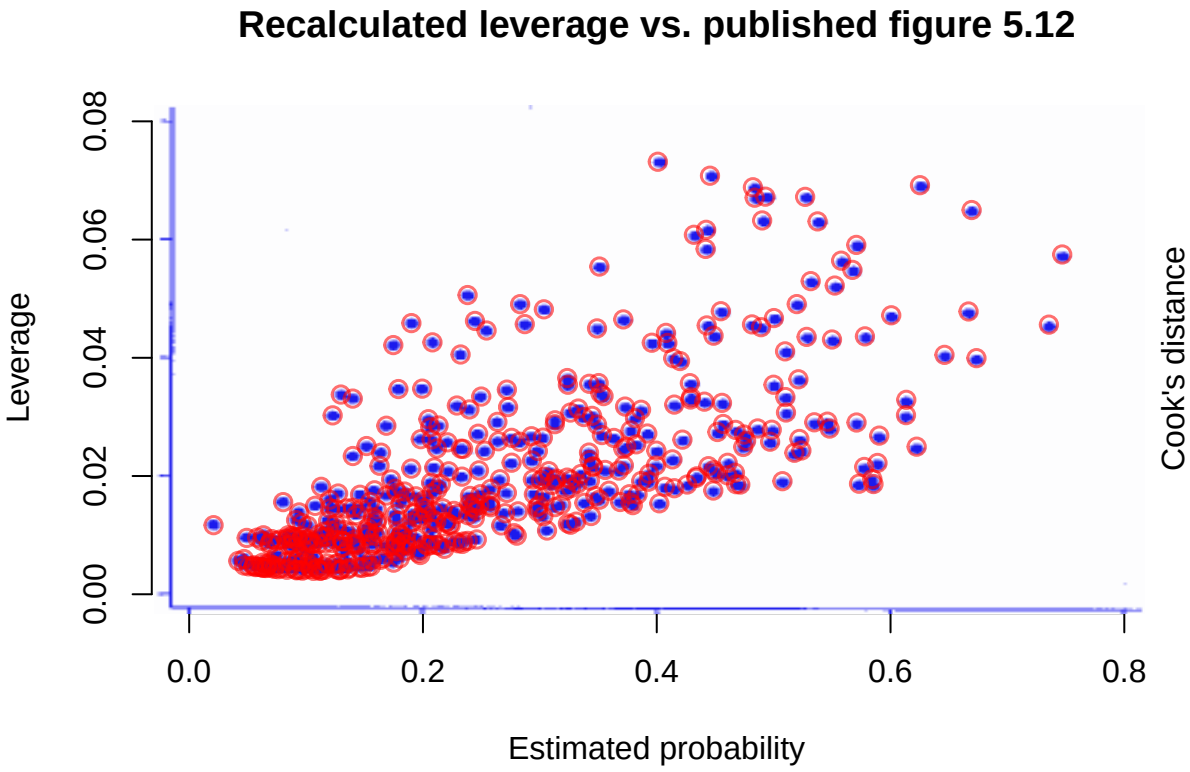
This looks pretty good as well.

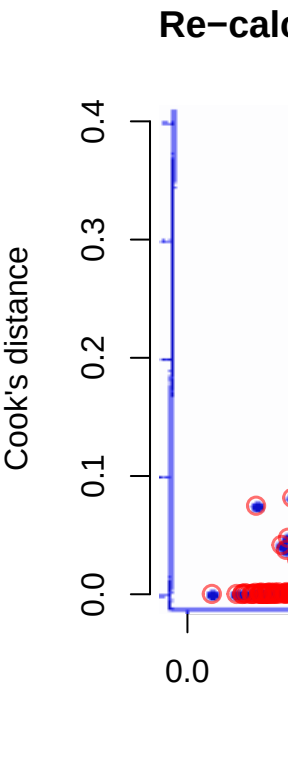
I decided to recalculate all the statistics my self, based on the formulas given in Hosmer et al 2013.

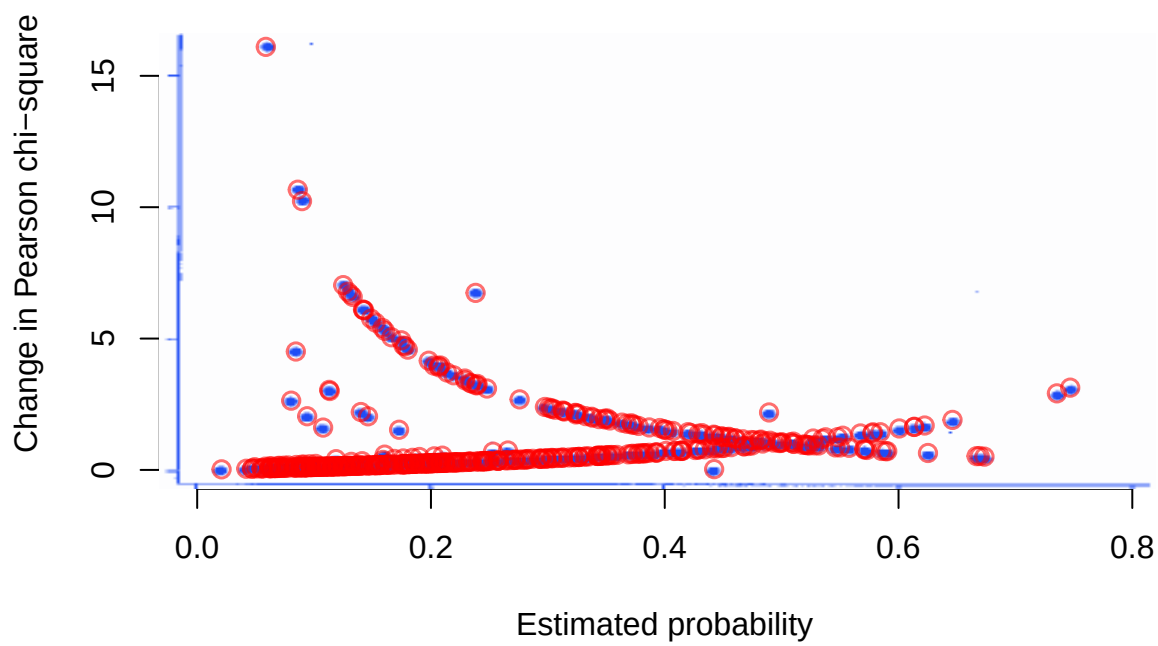
```
logRegDiagn <- function(model){

  ################
  # Basic data extraction and calculation

  # Extract non-aggregated data
  datN <- model$model
    # convert to covariate pattern
  cp <- unique(datN[,-1])

  # Indicator for covariate patterns
  cpIdN<- apply(datN[,-1], 1, function(x) paste(x, collapse = ""))
  cpIdM <- apply(cp, 1, function(x) paste(x, collapse = ""))

  # Size of covariate patterns
  m <- tapply(datN[,1], cpIdN, length)[rank(cpIdM)]
  # Number of cases per covariate pattern
  Y <- tapply(model$y, cpIdN, sum)[rank(cpIdM)]
  # Estimated probability
  Pi <- fitted.values(model)[rownames(cp)]

  ####################
  # Basic residuals

  # Pearson residual (formula 5.1)
  rPears <- (Y - Pi * m) / sqrt(m * Pi * (1 - Pi))
```

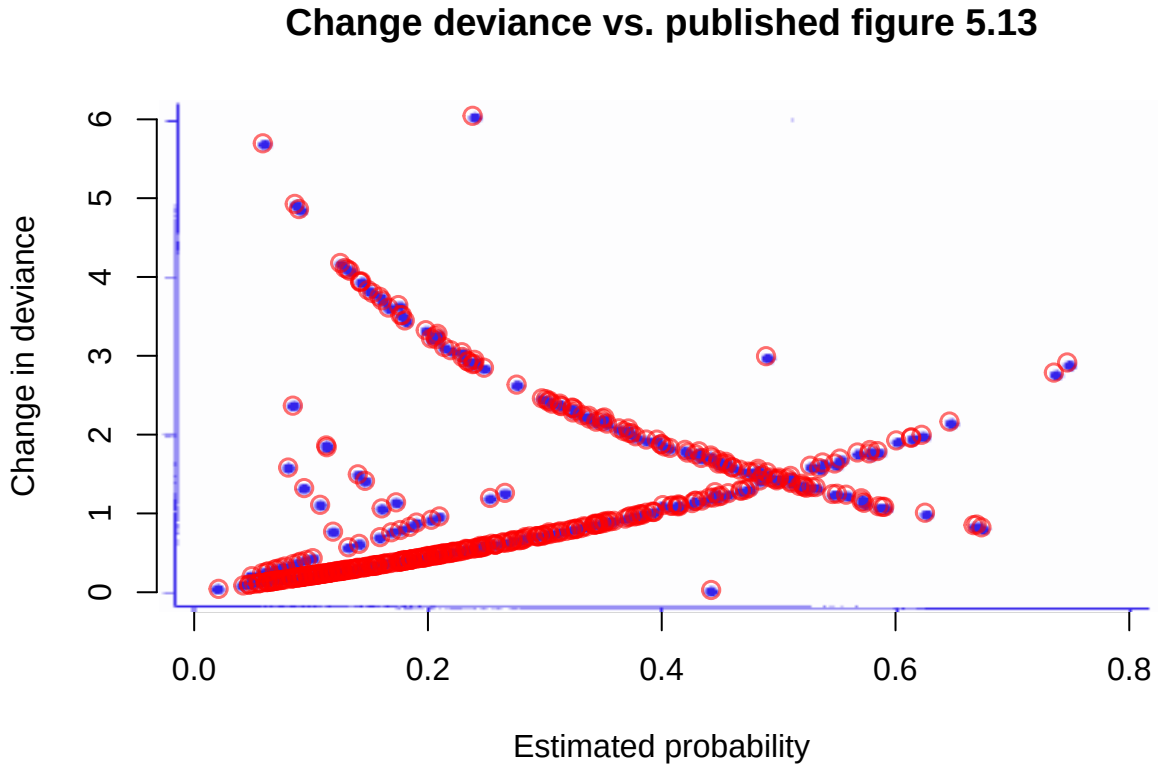
```
  # Deviance residual (formula 5.3)
  rDev <- ifelse(Y - m * Pi > 0, 1, -1) *
    (2 * (Y * log(Y / (m * Pi)) + (m - Y) * log((m - Y) / (m * (1 - Pi)))))^(1/2)
  rDev[Y == 0] <- -sqrt(2 * m[Y == 0] * abs(log(1 - Pi[Y == 0])))
  rDev[Y == m] <- sqrt(2 * m[Y == m] * abs(log(Pi[Y == m])))


  ############
  # Diagnostic statistics to plot

  # Leverage
  V <- diag(m * Pi * (1 - Pi))
  X <- unique(model.matrix(model))

  H <- V^(1/2) %*% X %*% solve(t(X) %*% V %*% X) %*% t(X) %*% (V^(1/2)) # formula 5.21
  h <- diag(H)

  # Cook's distance (formula 5.24)
  deltaBeta <- rPears^2 * h / (1 - h)^2

  # Change in Pearson chi-square (formula 5.25)
  deltaChi <- rPears^2 / (1 - h)

  # Change in deviance (formula 5.26)
  deltaDeviance <- rDev^2 / (1 - h)


  #################
  # collect output data
  out <- data.frame(m, Y, Pi, rPears, rDev, h, deltaBeta, deltaChi, deltaDeviance, cpIdM, cp)
  rownames(out) <- rownames(cp)
  out

}
```

The resulting plots finely fit the figures of the book. I think that the marginal deviations that can be seen are due to my less than perfect scans of the printed graphs.


Recalculated leverage vs. published figure 5.12
Leverage
0.00
0.02
0.04
0.06
0.08
0.0
0.2
0.4
0.6
0.8
Estimated probability


Re−calc
Cook's distance
0.0
0.1
0.2
0.3
0.4
0.0


Change in Pearson chi−square vs. published figure 5.13
Change in Pearson chi−square
0
5
10
15
0.0
0.2
0.4
0.6
0.8
Estimated probability



Summary

There are two R packages epiR and logisticDx that provide functions to calculate diagnostics of outliers and influential covariate patterns for logistic regression models. Non of them exactly reproduces the plots recommended by Hosmer et al. that where produced with Stata. Plots based on epiR differ substatially from the book, those based on logisticDx only marginally. Stata - the program used to produce the plots in the book - uses a different formula but it seems that you can multiply the single case leverages (that are routinely calculated by R) with the group size of the covariate pattern to get the same result. In the end I calculated the statistcs my self in R, using the formulas from the book. The resulting plots fit perfectly with the graphs from the book.