DATA501 Assignement 2 - Corvin Idler - ID 300598312

2024-08-13

Introduction

This is a PDF generated from a RMarkdown file for Assignment 2 of the DATA 501 class 2024 from Victoria University Wellington https://www.wgtn.ac.nz/courses/data/501/2024/offering?crn=33170

The repository underpinning this file and assignment can be found at https://github.com/econdatatech/distancemeasures/

Install instructions

I created an R package hosted at the above URL that will be loaded with the following lines of code

```
knitr::opts_chunk$set(echo = TRUE,  warning = FALSE, message = FALSE)
suppressWarnings({
library(devtools)
install_github("econdatatech/distancemeasures",force=TRUE)
})
```

```
## Loading required package: usethis
```

```
## Downloading GitHub repo econdatatech/distancemeasures@HEAD
```

```
## -- R CMD build -----------------------------------------------------------------
##          checking for file 'C:\Users\corvini\AppData\Local\Temp\RtmpYNRCpW\remotes946060a52987\econda
##       -  preparing 'distancemeasures': (596ms)
##    checking DESCRIPTION meta-information ...  v  checking DESCRIPTION meta-information
##       -  checking for LF line-endings in source and make files and shell scripts
##   -  checking for empty or unneeded directories
##    Omitted 'LazyData' from DESCRIPTION
##       -  building 'distancemeasures_0.1.0.tar.gz'
##
##
```

```
## Installing package into 'C:/Users/corvini/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

Purpose and execution example

The user will provide the program with a data set as well as a model (which is an object of class lm) Three measures of influence will be calculated and plotted”

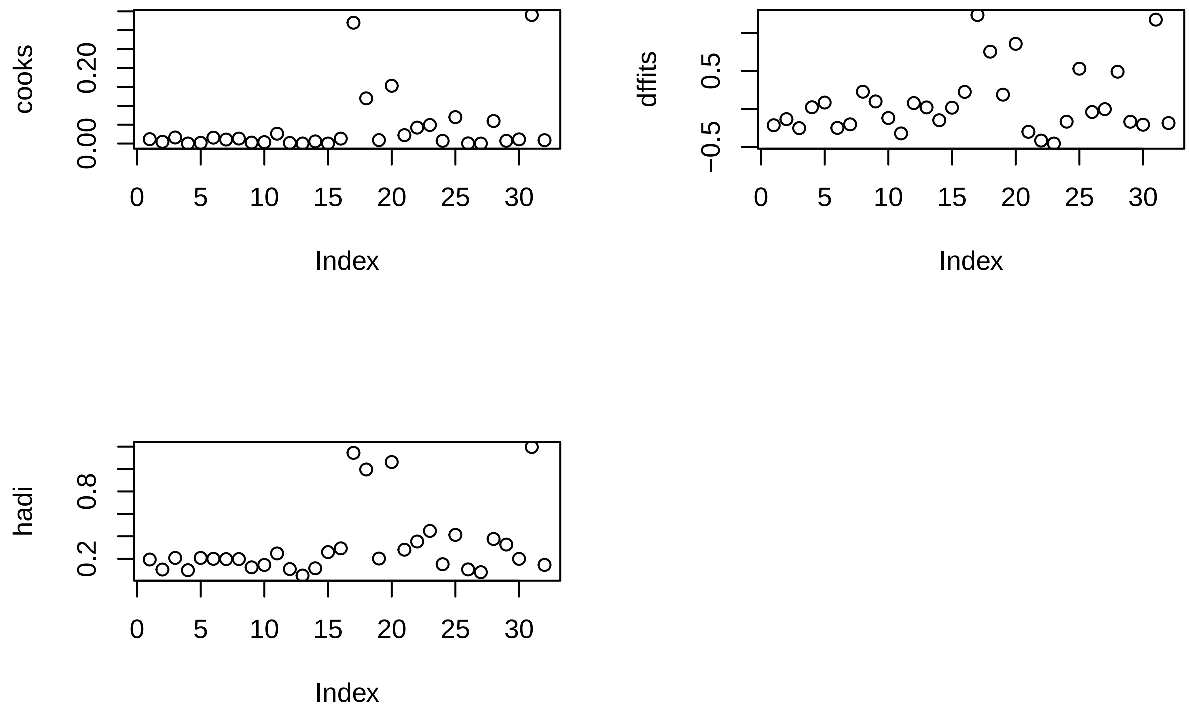
- Cooks Distance Measure (Cook, 1977)
- DFFITS (Welsch and Kuh, 1977; Belsley, 1980)
- Hadis Influence Measure (Hadi, 1992)

For sake of consistency and due to the discussion in Hadi (1992) (page 14) I decided to not include any cutoff values for the various influence measures.

To test the package we can use the data from the famous car package and fit a linear model and then plot some distance/influence measures

```r
knitr::opts_chunk$set(echo = TRUE)
library(car)
library(distancemeasures)
data(mtcars)
# Fit a linear regression model
# Predicting 'mpg' (miles per gallon) based on
# 'disp' (displacement), 'hp' (horsepower), and 'wt' (weight)
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
distances(mtcars,model)
```

```
## $cooks
##           Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4 Drive 
##        1.152035e-02        4.621112e-03        1.598334e-02        1.283888e-04 
##   Hornet Sportabout             Valiant          Duster 360           Merc 240D 
##        1.839055e-03        1.560119e-02        1.053270e-02        1.313511e-02 
##            Merc 230            Merc 280           Merc 280C          Merc 450SE 
##        2.525382e-03        3.671067e-03        2.606104e-02        1.551454e-03 
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood Lincoln Continental 
##        1.049983e-04        5.648180e-03        7.218880e-05        1.298764e-02 
##   Chrysler Imperial            Fiat 128         Honda Civic      Toyota Corolla 
##        3.199707e-01        1.196019e-01        9.092102e-03        1.529771e-01 
##       Toyota Corona    Dodge Challenger         AMC Javelin          Camaro Z28 
##        2.215865e-02        4.218196e-02        4.909944e-02        7.181085e-03 
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2        Lotus Europa 
##        6.980693e-02        4.163138e-04        1.732523e-06        5.959750e-02 
##      Ford Pantera L        Ferrari Dino       Maserati Bora          Volvo 142E 
##        7.279943e-03        1.100867e-02        3.402911e-01        8.796726e-03 
## 
## $dffits
##           Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4 Drive 
##        -0.214635969        -0.134436580        -0.252613308         0.022255503 
##   Hornet Sportabout             Valiant          Duster 360           Merc 240D 
##         0.084277714        -0.249193012        -0.202524979         0.226564439 
##            Merc 230            Merc 280           Merc 280C          Merc 450SE 
##         0.098857553        -0.119239383        -0.321760166         0.077444340 
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood Lincoln Continental 
##         0.020127530        -0.148727038         0.016686929         0.224608876 
##   Chrysler Imperial            Fiat 128         Honda Civic      Toyota Corolla 
##         1.235429008         0.753455967         0.187983207         0.856585474 
##       Toyota Corona    Dodge Challenger         AMC Javelin          Camaro Z28 
##        -0.300312659        -0.415994791        -0.454728883        -0.167175639 
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2        Lotus Europa 
##         0.529876307        -0.040083844        -0.002585074         0.490175087 
##      Ford Pantera L        Ferrari Dino       Maserati Bora          Volvo 142E 
##        -0.167849319        -0.207088112         1.174684221        -0.185577850 
## 
## $hadi
##           Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4 Drive 
##          0.19318654          0.10375645          0.20740884          0.09775893 
##   Hornet Sportabout             Valiant          Duster 360           Merc 240D 
##          0.20724815          0.19946745          0.19652103          0.19692137 
##            Merc 230            Merc 280           Merc 280C          Merc 450SE
##           0.12314263          0.14455206          0.24742810          0.10784806
##           Merc 450SL         Merc 450SLC  Cadillac Fleetwood Lincoln Continental
##           0.05027013          0.11471138          0.25914561          0.29265274
##    Chrysler Imperial            Fiat 128         Honda Civic      Toyota Corolla
##           1.14426624          0.99609055          0.20171955          1.06239341
##        Toyota Corona    Dodge Challenger         AMC Javelin          Camaro Z28
##           0.28064340          0.35373215          0.44820010          0.15099700
##     Pontiac Firebird           Fiat X1-9       Porsche 914-2        Lotus Europa
##           0.41281074          0.10477548          0.08003177          0.37633639
##       Ford Pantera L        Ferrari Dino       Maserati Bora          Volvo 142E
##           0.32631790          0.19910073          1.19634000          0.14496664
```



Implementation details

The user facing function distances() makes use of the following helper functions (one for each distance/influence measure):

Cooks distance

Below implementation is based on Cook (1977) (page 16ff.)

```
# based on https://doi.org/10.1080/00401706.1977.10489493
# and https://github.com/SurajGupta/r-source/blob/master/src/library/stats/R/lm.influence.R
cooks_distance_lm <- function(model) {
  # as per page 15 of Cook 1977 (above equation 1)
  resid <- stats::weighted.residuals(model) # to allow for weighted regression
  # residual degrees of freedom (number of observ.
  # minus number of regression coefficients)
  df <- stats::df.residual(model)
  sd <- sqrt(stats::deviance(model) / df)
  # diagonals of the 'hat' matrix.
  hat <- stats::lm.influence(model, do.coef = FALSE)$hat

  p <- model$rank
  # equation 7 in Cook 1977 page 16
  D <- ((resid / (sd * sqrt((1 - hat))))^2 * hat / (p * (1 - hat)))
  D[is.infinite(D)] <- NaN
  return(D)
}
```

dffits

This implementation is based on Belsley, Kuh, and Welsch (1980) and Chatterjee and Hadi (2015)

```
# based on https://avys.omu.edu.tr/storage/app/public/rezzanu/141865/[David_A._Belsley,_Edwin_Kuh,_Roy_
# and https://github.com/SurajGupta/r-source/blob/master/src/library/stats/R/lm.influence.R
dffits_lm <- function(model) {
  hat <- stats::lm.influence(model, do.coef = FALSE)$hat
  sigma <- stats::lm.influence(model, do.coef = FALSE)$sigma
  res <- stats::weighted.residuals(model)
  # based on equation 2.11 on page 15 of Belsley 1980
  # re-written to avoid one sqrt(1-hat)
  dffits <- res * sqrt(hat) / (sigma * (1 - hat))
  dffits[is.infinite(dffits)] <- NaN
  return(dffits)
}
```

Hadi

Ali Haid criticises in his 1992 paper that all prexisting influence measure only assessed the influence on a specific regression result, he on the other hand proposes "a measure of overall potential influence" Hadi (1992). The implementation below is based on the formula on page 113 in Chatterjee and Hadi (2015). The formula tries to measure "outlyingness" and X-space (first term) as well as response variable space (second term) Chatterjee and Hadi (2015).

```
#based on https://ideas.repec.org/a/eee/csdana/v14y1992i1p1-27.html
#and https://sadbhavnapublications.org/research-enrichment-material/2-Statistical-Books/Regression-Anal
hadi_lm <- function(model) {
  h <- stats::hatvalues(model)
  # based on sentence under equation 3.7 "normalized residuals".
  di <- stats::residuals(model) / sqrt(sum(stats::residuals(model)^2))
  p <- length(stats::coef(model)) - 1
  result <-(h / (1 - h) + (p + 1) / (1 - h) * di^2 / (1 - di^2))
  result[is.infinite(result)] <- NaN
  return(result)
}
```

Bibliograhy

Belsley, David A, Edwin Kuh, and Roy E. Welsch. 1980. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity /*. Wiley Series in Probability and Mathematical Statistics. New York: Wiley-Interscience.

Chatterjee, Samprit, and Ali S Hadi. 2015. *Regression Analysis by Example*. 5th ed. New York: John Wiley,.

Cook, R. Dennis. 1977. "Detection of Influential Observation in Linear Regression." *Technometrics* 19 (1): 15–18.

Hadi, Ali S. 1992. "A new measure of overall potential influence in linear regression." *Computational Statistics & Data Analysis* 14 (1): 1–27.