Robust Regression (R)

Robust regression is an alternative to least squares regression when data are "polluted" with outliers or influential observations, and it can also be used to detect influential observations.

Please note: The purpose of this page is to show how to use various data analysis commands. It does not cover all aspects of the research process which researchers are expected to do. In particular, it does not cover data cleaning and checking, verification of assumptions, model diagnostics, or potential follow-up analyses.

Introduction

Let's begin our tutorial on robust regression with some terms in linear regression.

Residual: The difference between the predicted value and the actual, observed value.

Outlier: In linear regression, an outlier is an observation with a large residual. In other words, it is an observation whose dependent-variable value is unusual given its value on the predictor variables. An outlier may indicate a sample peculiarity or may indicate a data entry error or other problem.

Leverage: An observation with an extreme value on a predictor variable is a point with high leverage. Leverage is a measure of how far an independent variable deviates from its mean. Thus, high leverage points can have a tremendous amount of effect on the estimate of regression coefficients.

Influence: An observation is said to be influential if removing the observation substantially changes the estimate of the regression coefficients.

Cook's distance (or Cook's D): A measure that combines the information of leverage and residual of the observation.

Robust regression can be used in any situation in which you would use least-squares regression. When fitting a least-squares regression, we might find some outliers or high leverage data points. We have decided that these data points are not data entry errors, neither are they from a different population than most of our data. So we have no compelling reason to exclude them from the analysis. Robust regression might be a good strategy since it is a compromise between excluding these points entirely from the analysis, including all the data points, and treating them equally in OLS regression. The idea of robust regression is to weigh the observations differently based on how well-behaved these observations are. Roughly speaking, it is a form of weighted and reweighted least squares regression.

The **rlm** command in the **MASS** package command implements several versions of robust regression. On this page, we will show M-estimation with Huber and bisquare weighting. These two are very standard. M-estimation defines a weight function such that the estimating equation becomes $\sum_{i=1}^{n} = w_i(y_i - x^T b)x_i^T = 0$. But the weights depend on the residuals and the residuals on the weights. The equation is solved using Iteratively Reweighted Least Squares (IRLS). For example, the coefficient matrix at iteration j is $B_j = [X^T W_{j-1} X]^{-1} X^T W_{j-1} Y$ where the subscripts indicate the matrix at a particular iteration (not rows or columns). The process continues until it converges. In Huber weighting, observations with small residuals get a weight of 1 and the larger the residual, the smaller the weight. This is defined by the weight function

$$w(e) = \begin{cases} 1, & \text{for } |e| \leq k \\ \frac{k}{|e|}, & \text{for } |e| > k \end{cases}$$

With bisquare weighting, all cases with a non-zero residual get down-weighted at least a little.

Description of our data analysis

For our data analysis below, we will use the crime dataset that appears in "Statistical Methods for Social Sciences, Third Edition" by Alan Agresti and Barbara Finlay (Prentice Hall, 1997). The variables are state id (**sid**), state name (**state**), violent crimes per 100,000 people (**crime**), murders per 1,000,000 (**murder**),

the percent of the population living in metropolitan areas (**pctmetro**), the percent of the population that is white (**pctwhite**), percent of the population with a high school education or above (**pcths**), percent of the population living under the poverty line (**poverty**), and percent of the population that are single parents (**single**). It has 51 observations. We are going to use **poverty** and **single** to predict **crime*. **Note** that you have to use the **foreign** package to read the data.

Let's look first at the summary of our data.

```
cdata <- as_tibble(foreign::read.dta(file = "~/Desktop/Tutorials/crime.dta"))
summary(cdata)
```

```
##       sid          state               crime            murder
##  Min.   : 1.0   Length:51          Min.   :  82.0   Min.   : 1.600
##  1st Qu.:13.5   Class :character   1st Qu.: 326.5   1st Qu.: 3.900
##  Median :26.0   Mode  :character   Median : 515.0   Median : 6.800
##  Mean   :26.0                      Mean   : 612.8   Mean   : 8.727
##  3rd Qu.:38.5                      3rd Qu.: 773.0   3rd Qu.:10.350
##  Max.   :51.0                      Max.   :2922.0   Max.   :78.500
##     pctmetro         pctwhite         pcths          poverty
##  Min.   : 24.00   Min.   :31.80   Min.   :64.30   Min.   : 8.00
##  1st Qu.: 49.55   1st Qu.:79.35   1st Qu.:73.50   1st Qu.:10.70
##  Median : 69.80   Median :87.60   Median :76.70   Median :13.10
##  Mean   : 67.39   Mean   :84.12   Mean   :76.22   Mean   :14.26
##  3rd Qu.: 83.95   3rd Qu.:92.60   3rd Qu.:80.10   3rd Qu.:17.40
##  Max.   :100.00   Max.   :98.50   Max.   :86.60   Max.   :26.40
##      single
##  Min.   : 8.40
##  1st Qu.:10.05
##  Median :10.90
##  Mean   :11.33
##  3rd Qu.:12.05
##  Max.   :22.10
```

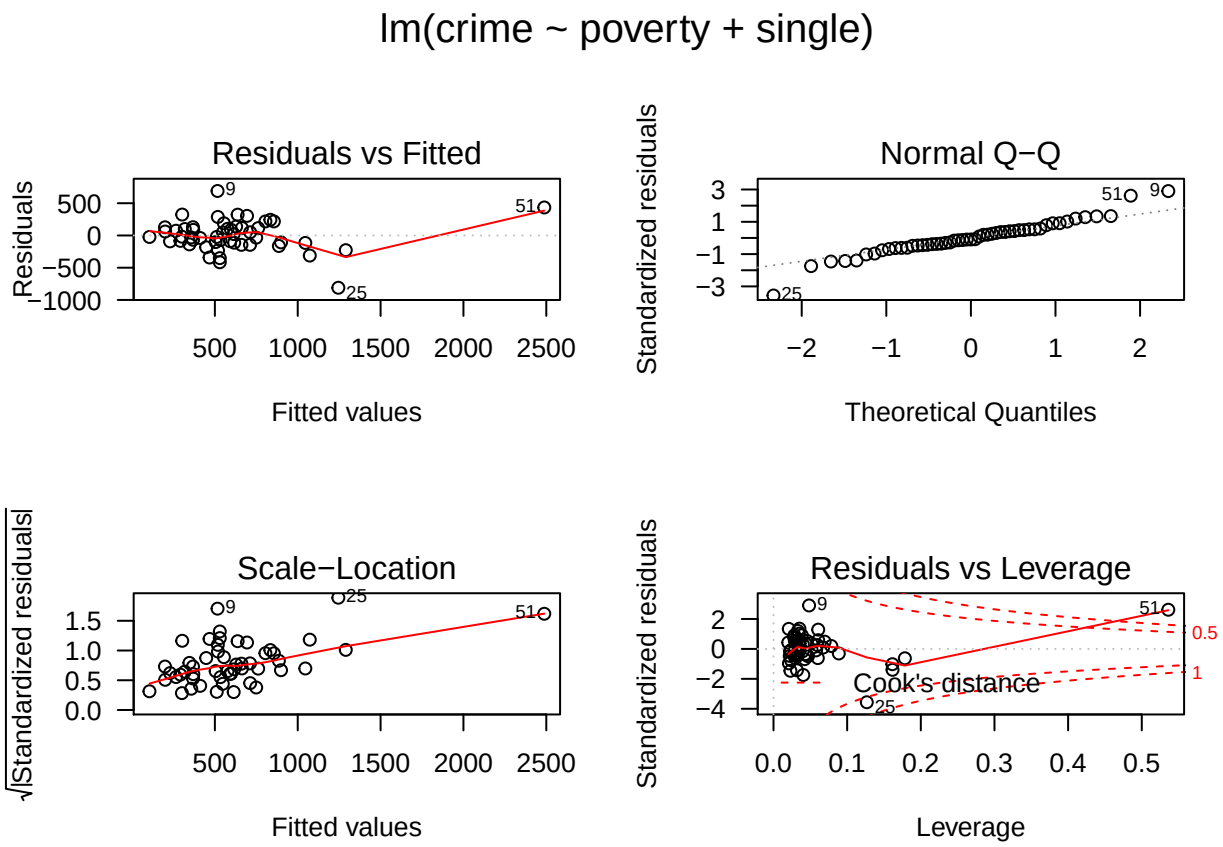
In most cases, we begin by running an OLS regression and doing some diagnostics. We will start by running an OLS regression and looking at diagnostic plots examining residuals, fitted values, Cook's distance, and leverage.

```
summary(ols <- lm(crime ~ poverty + single, data = cdata))
```

```
##
## Call:
## lm(formula = crime ~ poverty + single, data = cdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -811.14 -114.27  -22.44  121.86  689.82
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1368.189    187.205  -7.308 2.48e-09 ***
## poverty         6.787      8.989   0.755    0.454
## single        166.373     19.423   8.566 3.12e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 243.6 on 48 degrees of freedom
```

```
## Multiple R-squared:  0.7072, Adjusted R-squared:  0.695
## F-statistic: 57.96 on 2 and 48 DF,  p-value: 1.578e-13
```

```
opar <- par(mfrow = c(2,2), oma = c(0, 0, 1.1, 0))
plot(ols, las = 1)
```




```
par(opar)
```

We can identify observations 9, 25, and 51 as possibly problematic to our model from these plots. We can look at these observations to see which states they represent.

```
cdata %>% filter(sid == 9 | sid == 25 | sid == 51)
```

```
## # A tibble: 3 x 9
##     sid state crime murder pctmetro pctwhite pcths poverty single
##   <dbl> <chr> <int>  <dbl>    <dbl>    <dbl> <dbl>   <dbl>  <dbl>
## 1     9 fl     1206   8.90     93       83.5  74.4    17.8   10.6
## 2    25 ms      434  13.5      30.7     63.3  64.3    24.7   14.7
## 3    51 dc     2922  78.5     100       31.8  73.1    26.4   22.1
```

DC, Florida, and Mississippi have either high leverage or large residuals. We can display the observations with relatively large values of Cook's D. A conventional cut-off point is $4/n$, where n is the number of observations in the data set. We will use this criterion to select the values to display taking also the standardized residuals of our model into account.

We probably should drop DC, to begin with since it is not even a state. However, we include it in the analysis just to show that it has a large Cook's D and demonstrate how it will be handled by **rlm**. We also look at the residuals, for which we generated a new variable called **abd_stdres**, which is the absolute value of the residuals since the sign of the residual does not matter. We then print the ten observations with the highest absolute residual values.

```
cdata_inf <- cdata %>% mutate(cooks_d = cooks.distance(ols),
                             std_res = stdres(ols),
                             abs_stdres = abs(std_res))
cdata_inf %>% dplyr::select(sid, state, crime, murder, single, cooks_d, std_res,
                     abs_stdres) %>%
  arrange(desc(abs_stdres)) %>%
  head(15)
```

```
## # A tibble: 15 x 8
##      sid state crime murder single cooks_d std_res abs_stdres
##    <dbl> <chr> <int>  <dbl>  <dbl>   <dbl>   <dbl>      <dbl>
##  1    25 ms      434  13.5   14.7  0.614    -3.56       3.56
##  2     9 fl     1206   8.90  10.6  0.143     2.90       2.90
##  3    51 dc     2922  78.5   22.1  2.64      2.62       2.62
##  4    46 vt      114   3.60  11    0.0427   -1.74       1.74
##  5    26 mt      178   3     10.8  0.0168   -1.46       1.46
##  6    21 me      126   1.60  10.6  0.0223   -1.43       1.43
##  7     1 ak      761   9     14.3  0.125    -1.40       1.40
##  8    31 nj      627   5.30   9.60 0.0223    1.35       1.35
##  9    14 il      960  11.4   11.5  0.0127    1.34       1.34
## 10    20 md      998  12.7   12    0.0357    1.29       1.29
## 11    19 ma      805   3.90  10.9  0.0164    1.20       1.20
## 12    18 la     1062  20.3   14.9  0.0670   -1.02       1.02
## 13     5 ca     1078  13.1   12.5  0.0123    1.02       1.02
## 14    50 wy      286   3.40  10.8  0.00667  -0.966      0.966
## 15    40 sc     1023  10.3   12.3  0.0111    0.912      0.912
```

Now let's run our first robust regression. Robust regression is done by iterated re-weighted least squares (IRLS). The command for running robust regression is **rlm** in the **MASS** package. Several weighting functions can be used for IRLS. We are going first to use the Huber weights in this example. We will then look at the final weights created by the IRLS process, which can be very useful.

```
summary(rr_huber <- MASS::rlm(crime ~ poverty + single, data = cdata))
```

```
##
## Call: rlm(formula = crime ~ poverty + single, data = cdata)
## Residuals:
##     Min      1Q  Median      3Q     Max
## -846.09 -125.80  -16.49  119.15  679.94
##
## Coefficients:
##             Value      Std. Error t value
## (Intercept) -1423.0373   167.5899    -8.4912
## poverty         8.8677     8.0467     1.1020
## single        168.9858    17.3878     9.7186
##
## Residual standard error: 181.8 on 48 degrees of freedom
```

```
cdata_weights <- cdata %>% mutate(resid = rr_huber$resid,
                                  weight = rr_huber$w)

cdata_weights %>% dplyr::select(state, resid, weight) %>%
  arrange(weight) %>% head(15)
```

```
## # A tibble: 15 x 3
##    state  resid weight
```

```
##    <chr>  <dbl>  <dbl>
##  1 ms    -846.   0.289
##  2 fl     680.   0.360
##  3 vt    -410.   0.596
##  4 dc     376.   0.649
##  5 mt    -356.   0.686
##  6 me    -337.   0.725
##  7 nj     331.   0.738
##  8 il     319.   0.766
##  9 ak    -313.   0.781
## 10 md     307.   0.796
## 11 ma     291.   0.840
## 12 la    -267.   0.916
## 13 al     105.   1
## 14 ar      30.5  1
## 15 az     -43.3  1
```

We can see that roughly, as the absolute residual goes down, the weight goes up. In other words, cases with large residuals tend to be down-weighted. This output shows us that the observation for Mississippi will be down-weighted the most. Florida will also be substantially down-weighted. All observations not shown above have a weight of 1. In OLS regression, all cases have a weight of 1. Hence, the more cases in the robust regression that have a weight close to one, the closer the results of the OLS and robust regressions.

Next, let's run the same model but using the bisquare weighting function. Then, again, we can look at the weights.

```
summary(rr_bisquare <- MASS::rlm(crime ~ poverty + single, data=cdata, psi = psi.bisquare))
```

```
##
## Call: rlm(formula = crime ~ poverty + single, data = cdata, psi = psi.bisquare)
## Residuals:
##     Min      1Q  Median      3Q     Max
## -905.59 -140.97  -14.98  114.65  668.38
##
## Coefficients:
##             Value      Std. Error t value
## (Intercept) -1535.3338   164.5062    -9.3330
## poverty        11.6903     7.8987     1.4800
## single        175.9303    17.0678    10.3077
##
## Residual standard error: 202.3 on 48 degrees of freedom
```

```
cdata_biweights <- cdata %>% mutate(resid = rr_bisquare$resid,
                                    weight = rr_bisquare$w)

cdata_biweights %>% dplyr::select(state, resid, weight) %>%
  arrange(weight) %>% head(15)
```

```
## # A tibble: 15 x 3
##    state resid  weight
##    <chr> <dbl>   <dbl>
##  1 ms    -906. 0.00765
##  2 fl     668. 0.253
##  3 vt    -403. 0.671
##  4 mt    -361. 0.731
##  5 nj     346. 0.751
```

```
##  6 la    -333. 0.769
##  7 me    -329. 0.774
##  8 ak    -326. 0.778
##  9 il     313. 0.794
## 10 md     309. 0.799
## 11 ma     298. 0.813
## 12 dc     261. 0.854
## 13 wy    -234. 0.882
## 14 ca     201. 0.912
## 15 ga    -187. 0.924
```

We can see that the weight given to Mississippi is dramatically lower using the bisquare weighting function than the Huber weighting function. The parameter estimates from these two different weighting methods differ. When comparing the results of a regular OLS regression and a robust regression, if the results are very different, you will most likely want to use the results from the robust regression. Large differences suggest that outliers are highly influencing the model parameters. Various functions have advantages and drawbacks. Huber weights can have difficulties with severe outliers, and bisquare weights can have issues converging or may yield multiple solutions.

```
results <- data.frame(ols_coeffs = ols$coefficients,
                      rr_huber_coeffs = rr_huber$coefficients,
                      rr_bisquare_coeffs = rr_bisquare$coefficients)
results
```

```
##               ols_coeffs rr_huber_coeffs rr_bisquare_coeffs
## (Intercept) -1368.188661    -1423.037337        -1535.33376
## poverty         6.787359        8.867678           11.69033
## single        166.372670      168.985787          175.93032
```

As we can see, the results of our models are pretty different, especially concerning the coefficients of the (**intercept**). While we usually are not interested in the intercept, the intercept would be useful if we had centered one or both of the predictor variables. On the other hand, we notice that **poverty** is not statistically significant in any analysis, whereas **single** is significant in all analyses.

Bare in Mind

- Robust regression does not address issues of heterogeneity of variance. This problem can be addressed using functions in the **sandwich** package after the **lm** function.
- The examples are shown here have presented the R code for **M** estimation. Other estimation options are also available in **rlm** and other R commands and packages: Least trimmed squares using **ltsReg** in the **robustbase** package and **MM** using **rlm**.

References

- John Fox, "Applied regression analysis, linear models, and related models", Sage publications, Inc, 1997

See also

R documentation for **rlm**