Regression 11

Weighted least squares regression and Robust regression

1 Weighted least squares regression

The first assumption of the least squares regression we have studied is that $\mathrm{Var}(\epsilon_i) = \sigma^2$ for all cases in the data. This assumption is in doubt in many problems, as variances can depend on the response, on one or more of the predictors, or possibly on other factors.

If nonconstant variance is diagnosed, but exact variances are unknown, we could consider two remedies. First, a transformation of the response Y can be used. The second alternative is weighted least squares (WLS) with empirically chosen weights. Weights that are simple functions such as $\sigma_i^2 = \mathrm{Var}(\epsilon_i) = \sigma^2 X_{i1}$ are used. If large samples with replication are available, then within-group variances may be used to provide approximate weights. Generally, however, empirical weights that are functions of the $\hat{Y}_i$ or $\hat{\epsilon}_i$ from ordinary least squares (OLS) cannot be recommended unless nonstandard methods are used to estimate variances.

1.1 Parameter estimation by weighted least squares

Formerly, we assumed

$$\mathbf{Y} \;\sim\; N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}).$$

That is, the regression errors $\epsilon_1, \dots, \epsilon_n$ were assumed to be *iid* $N(0, \sigma^2)$. Now suppose that the errors have unequal variances, which are known up to a proportionality constant,

$$\sigma_i^2 = \mathrm{Var}(\epsilon_i) = v_i \sigma^2, \qquad i = 1, \dots, n,$$

where $v_1, \dots, v_n$ are known. In matrix notation, we denote

$$\mathbf{Y} \;\sim\; N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{V}),$$

where $\mathbf{V} = \mathrm{diag}[v_1, \dots, v_n]$. Hence we have

$$\boldsymbol{\epsilon} = \mathbf{Y} - \mathbf{X}\boldsymbol{\beta} \;\sim\; N(0, \sigma^2\mathbf{V}).$$

Let us denote $\mathbf{W}^{1/2} = \mathrm{diag}\big(1/\sqrt{v_1}, \dots, 1/\sqrt{v_n}\big)$. Notice that $\mathbf{W} = \mathbf{W}^{1/2}\mathbf{W}^{1/2} = \mathbf{V}^{-1}$. Then we have

$$\mathbf{W}^{1/2}\boldsymbol{\epsilon} = \mathbf{W}^{1/2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \;\sim\; N(0, \sigma^2\mathbf{I}).$$

For convenience, let us denote

$$\boldsymbol{\epsilon}^* = \mathbf{W}^{1/2}\boldsymbol{\epsilon}, \qquad \mathbf{Y}^* = \mathbf{W}^{1/2}\mathbf{Y}, \quad \text{and} \quad \mathbf{X}^* = \mathbf{W}^{1/2}\mathbf{X}.$$

Thus, the weighted least squares (WLS) is equivalent to the OLS estimator on $\mathbf{Y}^*$ and $\mathbf{X}^*$:

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= (\mathbf{X}^{*\prime}\mathbf{X}^*)^{-1}\mathbf{X}^{*\prime}\mathbf{Y}^* = (\mathbf{X}'\mathbf{W}^{1/2}\mathbf{W}^{1/2}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}^{1/2}\mathbf{W}^{1/2}\mathbf{Y} \\ &= (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{Y}.\end{aligned}$$

Remark 11.1. The fitted values $\hat{\mathbf{Y}}$ are given by $\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{Y} = \mathbf{H}\mathbf{Y}$, where $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}$. We can easily see that $\mathbf{H}\mathbf{H} = \mathbf{H}$ (idempotent), but $\mathbf{H}' \neq \mathbf{H}$ (asymmetric). Thus, $\mathbf{H}$ is a projection matrix, but it is not an orthogonal projection matrix. △

That is, the WLS estimator is equivalent to minimizing

$$Q_2^* = \|\boldsymbol{\epsilon}^*\|^2 = \sum_{i=1}^{n} w_i \cdot \left\{Y_i - (\beta_0 + \beta_1 X_i + \ldots + \beta_{p-1} X_{p-1})\right\}^2. \tag{11.1}$$

For convenience, we define the row vectors in the data matrix $\mathbf{X}$ by $\mathbf{x}_i{}' = [1 \ X_{i1} \ X_{i2} \ \cdots \ X_{i,p-1}]$ so that we have

$$\underset{n\times p}{\mathbf{X}} = \begin{bmatrix} 1 & X_{11} & X_{12} & \cdots & X_{1,p-1} \\ 1 & X_{21} & X_{22} & \cdots & X_{2,p-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & X_{n1} & X_{n2} & \cdots & X_{n,p-1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1{}' \\ \mathbf{x}_2{}' \\ \vdots \\ \mathbf{x}_n{}' \end{bmatrix}.$$

Then we can rewrite (11.1) as

$$Q_2^* = \sum_{i=1}^{n} w_i \cdot \left\{Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}\right\}^2. \tag{11.2}$$

Note that if $w_i = 1$ for all i, then this is equivalent to the OLS.

Differentiating (11.2) with respect to $\boldsymbol{\beta}$, we have

$$\frac{\partial Q_2^*}{\partial \boldsymbol{\beta}} = -2 \sum_{i=1}^{n} w_i \cdot \left\{Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}\right\} \mathbf{x}_i{}'.$$

Thus, the WLS estimator is also obtained by solving

$$\sum_{i=1}^{n} w_i \cdot \left\{Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}\right\} \mathbf{x}_i{}' = \mathbf{0}, \tag{11.3}$$

where $\mathbf{0} = (0, 0, \ldots, 0)$.

1.2 Where do we get the weights

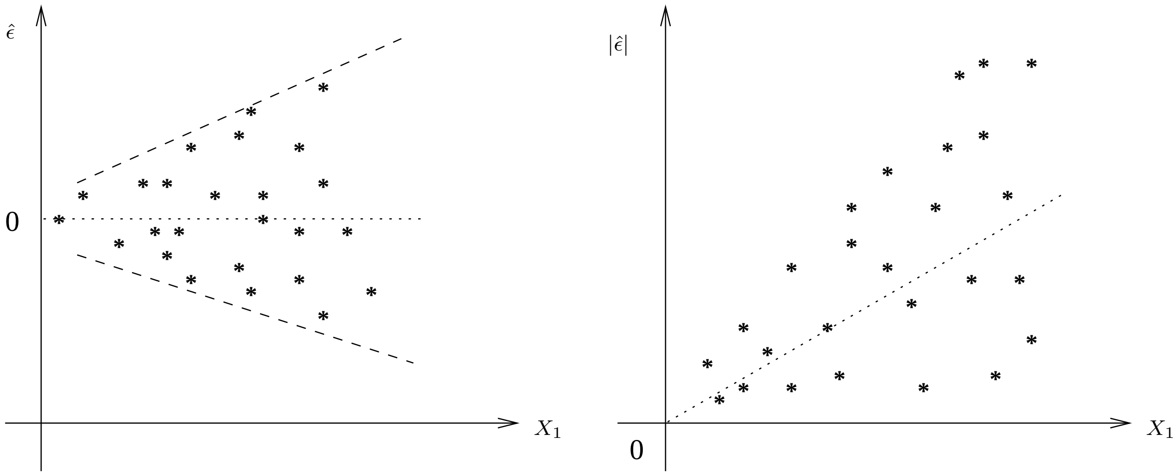
In WLS, we assume that $\sigma_i^2 = \text{Var}(\epsilon_i) = \sigma^2 v_i$, where $v_1, \ldots, v_n$ are known. Where do we get the weights in real data analysis? Note that the weights is inversely proportional to the variance σ_i^2, that is, $w_i \propto 1/\sigma_i^2$, or, $w_i \propto 1/v_i$ in this case.

1. From a prediction variable.

   Suppose that we fit an OLS regression, and a residual plot of $\hat{\epsilon}$ versus a predictor X_1 looks like this.

Residual plot against X_1

Residual plot against X_1



Then we might suppose that

$$\mathrm{Var}(\epsilon) = \sigma^2 X_1^{1/2}$$

$$\mathrm{Var}(\epsilon) = \sigma^2 X_1$$

$$\mathrm{Var}(\epsilon) = \sigma^2 X_1^2$$

$$\vdots$$

$$\mathrm{Var}(\epsilon) = \sigma^2(\hat{\alpha}_0 + \hat{\alpha}_1 X_1)^2,$$

where $\hat{\alpha}_0$ and $\hat{\alpha}_1$ are obtained by regressing $|\hat{\epsilon}|$ on X_1.

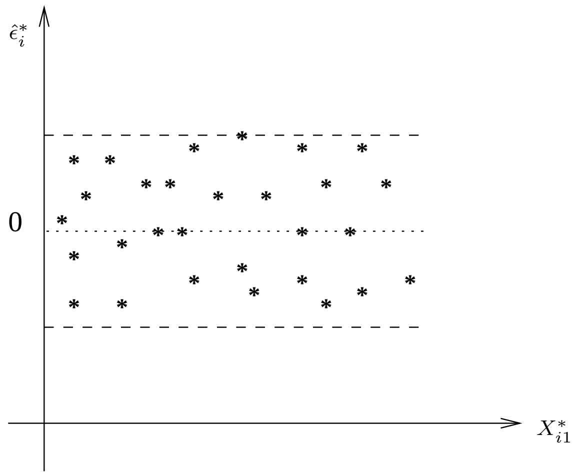
How do we know which power of X_1 to use? Suppose, for example, we try $\mathrm{Var}(\epsilon) = \sigma^2 X_1^{1/2}$, *i.e.*, we use $w_i = 1/X_{i1}^{1/2}$ as the weights. If the residual plot from this regression ($\hat{\epsilon}_i^* = w_i^{1/2}(Y_i - \hat{Y}_i)$ versus $X_{i1}^* = w_i^{1/2} X_{i1}$) looks good, then our variance function $\mathrm{Var}(\epsilon) = \sigma^2 X_1^{1/2}$ is OK. If the plot still fans out, then we need to use a stronger variance function (*e.g.*, $\mathrm{Var}(\epsilon) = \sigma^2 X_1$ or $\mathrm{Var}(\epsilon) = \sigma^2 X_1^2$).

2. From replication

Suppose that only a few distinct patterns of predictors are present. For example,

$$X = \begin{cases} 0 & : \quad \text{female} \\ 1 & : \quad \text{male} \end{cases}$$

Residual plot against X_1^*



If the sample sizes within each group are large enough, we can estimate σ^2 within each group.

3. From varying sample sizes

   Suppose that our responses Y_i are actually averages from sample of varying sizes. For example, let Y_i be the average wage for workers at the ith firm and n_i be the number of workers at the ith firm. Then we expect Y_i to have more random variation when n_i is smaller than when n_i is large. In building a regression model for $Y_1, \ldots, Y_n$, it may be sensible to assume that $\text{Var}(\epsilon_i) \propto 1/n_i$ and thus use the n_i's as weights.

Example 11.1. Textbook Example (Table 11.1 on Page 427).

Minitab

Read Data

```
MTB >READ c1 c2;
SUBC>  file  "S:\LM\CH11TA01.TXT" .
Entering data from file: S:\LM\CH11TA01.TXT
54 rows read.
```

Regression of C2 on C1

```
MTB > regr c2 1 c1;     # c2 = blood pressure    c1 = age
SUBC> resid c3;
SUBC> fits  c4;
SUBC> brief 1.

Regression Analysis: C2 versus C1
The regression equation is
C2 = 56.2 + 0.580 C1
```

```
Predictor      Coef   SE Coef       T      P
Constant     56.157     3.994   14.06  0.000
C1          0.58003   0.09695    5.98  0.000

S = 8.14575   R-Sq = 40.8%   R-Sq(adj) = 39.6%

Analysis of Variance
Source          DF      SS      MS      F      P
Regression       1  2375.0  2375.0  35.79  0.000
Residual Error  52  3450.4    66.4
Total           53  5825.3

Residual Plots for C2
```

Regression of C5 ($|\hat{\epsilon}_i|$) on C1

```
MTB > let c5 = abs(c3)
MTB > regr c5 1 c1;
SUBC> fits c6;
SUBC> brief 1.

Regression Analysis: C5 versus C1

The regression equation is
C5 = - 1.55 + 0.198 C1

Predictor      Coef   SE Coef       T      P
Constant     -1.549     2.187   -0.71  0.482
C1          0.19817   0.05309    3.73  0.000

S = 4.46057   R-Sq = 21.1%   R-Sq(adj) = 19.6%

Analysis of Variance

Source          DF       SS      MS      F      P
Regression       1   277.23  277.23  13.93  0.000
Residual Error  52  1034.63   19.90
Total           53  1311.86

Residual Plots for C5
```

Table 11.1 on Page 427

```
MTB > let c7 = 1/c6      # c7 = weight^(1/2)
MTB > let c8 = c7*c7     # c8 = weight
MTB > print c1 c2 c3 c5 c6 c8

Data Display
Row  C1   C2       C3       C5      C6        C8
  1  27   73   1.1822   1.1822  3.8012  0.069209
  2  21   66  -2.3376   2.3376  2.6121  0.146557
  3  22   63  -5.9176   5.9176  2.8103  0.126617
  4  24   75   4.9223   4.9223  3.2067  0.097251
 ................................................

 51  50   91   5.8415   5.8415  8.3591  0.014311
 52  52  100  13.6815  13.6815  8.7555  0.013045
 53  58   80  -9.7987   9.7987  9.9445  0.010112
 54  57  109  19.7813  19.7813  9.7463  0.010527
```

WLS (using weight option)

```
MTB > regr c2 1 c1;
SUBC> weight c8;
SUBC> resid  c9;
SUBC> fits   c10;
SUBC> brief 1.

Regression Analysis: C2 versus C1

```

```
Weighted analysis using weights in C8

The regression equation is
C2 = 55.6 + 0.596 C1

Predictor      Coef   SE Coef       T      P
Constant     55.566     2.521   22.04  0.000
C1          0.59634   0.07924    7.53  0.000

S = 1.21302   R-Sq = 52.1%   R-Sq(adj) = 51.2%


Analysis of Variance
Source          DF       SS       MS       F      P
Regression       1   83.341   83.341   56.64  0.000
Residual Error  52   76.514    1.471
Total           53  159.854

Residual Plots for C2
```

WLS (using OLS)

```
MTB > let c12 = c7*c1   # c12 = X*
MTB > let c22 = c7*c2   # c22 = Y*
MTB > regr c22 2 c7 c12 ;  # Regress Y* on X*
SUBC> noconstant ;
SUBC> fits c23;
SUBC> brief 1.

Regression Analysis: C22 versus C7, C12

The regression equation is
C22 = 55.6 C7 + 0.596 C12

Predictor       Coef   SE Coef       T      P
Noconstant
C7            55.566     2.521   22.04  0.000
C12          0.59634   0.07924    7.53  0.000
S = 1.21302

Analysis of Variance
Source          DF       SS       MS        F      P
Regression       2  12446.6   6223.3  4229.48  0.000
Residual Error  52     76.5      1.5
Total           54  12523.1
Residual Plots for C22
```

Print c10 and c23. Print c10 and c24

```
MTB > print c10 c23

Data Display
Row       C10      C23
  1   71.6670  18.8539
  2   68.0889  26.0663
  3   68.6853  24.4404
  4   69.8780  21.7915
     ................
 51   85.3829  10.2143
 52   86.5755   9.8882
 53   90.1536   9.0657
 54   89.5572   9.1888

MTB > let c24 = c23/c7
MTB > print c10 c24

Data Display
Row       C10      C24
  1   71.6670  71.6670
  2   68.0889  68.0889
  3   68.6853  68.6853
```

```
  4  69.8780  69.8780
     ................
 51  85.3829  85.3829
 52  86.5755  86.5755
 53  90.1536  90.1536
 54  89.5572  89.5572
```

R

Read Data

```
> mydata =
    read.table("https://raw.githubusercontent.com/AppliedStat/LM/master/CH11TA01.txt")
> x = mydata[,1]  ;   y = mydata[,2]
```

Regression of y (blood pressure) on x (age)

```
> LM = lm ( y ~ x )
> summary(LM)
Call:
lm(formula = y ~ x)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 56.15693    3.99367  14.061  < 2e-16 ***
x            0.58003    0.09695   5.983 2.05e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.146 on 52 degrees of freedom
Multiple R-Squared: 0.4077, Adjusted R-squared: 0.3963
F-statistic: 35.79 on 1 and 52 DF,  p-value: 2.050e-07
```

Regression of c5 ($|\hat{\epsilon}_i|$) on x (age)

```
> c3 = resid(LM)
> c4 = fitted(LM)
>
> c5 = abs (c3)
>
> LM2 = lm ( c5 ~ x )
> summary(LM2)

Call:
lm(formula = c5 ~ x)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.54948    2.18692  -0.709  0.48179
x            0.19817    0.05309   3.733  0.00047 ***
---
Residual standard error: 4.461 on 52 degrees of freedom
Multiple R-squared:  0.2113,    Adjusted R-squared:  0.1962
F-statistic: 13.93 on 1 and 52 DF,  p-value: 0.0004705
```

Table 11.1 on Page 427

```
> c6  = fitted (LM2)
> c7 = 1/c6         # c7 = w^(1/2)
> c8 = c7 * c7      # c8 = w

> cbind(x, y, c3, c5, c6, c8)
    x   y         c3        c5       c6          c8
1  27  73  1.1822391 1.1822391 3.801175 0.069209280
2  21  66 -2.3375761 2.3375761 2.612141 0.146557083
3  22  63 -5.9176069 5.9176069 2.810313 0.126616574
4  24  75  4.9223315 4.9223315 3.206658 0.097251155
```

```
.....................................................

51 50  91   5.8415308  5.8415308  8.359138 0.014311232
52 52 100  13.6814692 13.6814692  8.755482 0.013044872
53 58  80  -9.7987156  9.7987156  9.944516 0.010111898
54 57 109  19.7813152 19.7813152  9.746344 0.010527289
```

WLS using lm() with options

```
>  WLS = lm ( y ~ x , weights = c8)   # weights are given in c8

>  summary (WLS)
Call:
lm(formula = y ~ x, weights = c8)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 55.56577    2.52092  22.042  < 2e-16 ***
x            0.59634    0.07924   7.526 7.19e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.213 on 52 degrees of freedom
Multiple R-Squared: 0.5214, Adjusted R-squared: 0.5122
F-statistic: 56.64 on 1 and 52 DF,  p-value: 7.187e-10

>  c9  = resid (WLS)
>  c10 = fitted (WLS)
```

WLS using OLS, that is, lm() without options

```
> c12 = c7 * x  # c12 = X*
> c22 = c7 * y  # c22 = Y*

> LM2 = lm ( c22 ~ 0 + c7 + c12 )  # Regress Y* on X*
> c23 = fitted(LM2)
> summary(LM2)
Call:
lm(formula = c22 ~ 0 + c7 + c12)

Coefficients:
    Estimate Std. Error t value Pr(>|t|)
c7  55.56577    2.52092  22.042  < 2e-16 ***
c12  0.59634    0.07924   7.526 7.19e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.213 on 52 degrees of freedom
Multiple R-Squared: 0.9939, Adjusted R-squared: 0.9937
F-statistic:  4229 on 2 and 52 DF,  p-value: < 2.2e-16
```

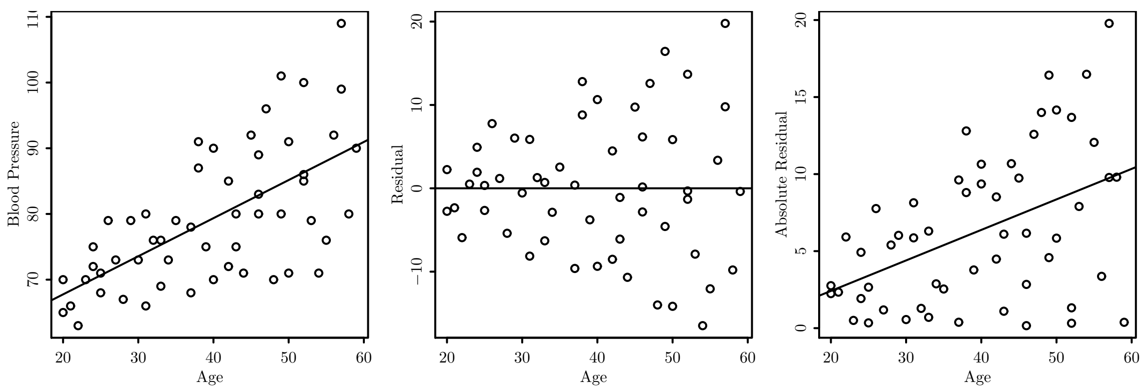
Print c10 and c23. Print c10 and c24

```
>  cbind(c10, c23)
        c10       c23
1  71.66699 18.853906
2  68.08894 26.066336
3  68.68528 24.440438
4  69.87797 21.791526
.....................................................

51 85.38285 10.214313
52 86.57554  9.888151
53 90.15359  9.065658
54 89.55724  9.188804

>  c24 = c23/c7
>  cbind(c10, c24)
        c10      c24
1  71.66699 71.66699
```

```
2  68.08894 68.08894
3  68.68528 68.68528
4  69.87797 69.87797
......................................................

51 85.38285 85.38285
52 86.57554 86.57554
53 90.15359 90.15359
54 89.55724 89.55724
```

Figure 11.1 on Page 428

```
> par (mfrow=c(1,3))
> plot(x,y)       # x=age, y=blood pressure
> abline(LM)

> plot(x,c3)      # c3=residual of blood pressures
> abline( lm(c3~x) )

> plot(x,c5)      # c5=abs(c3)
> abline( lm(c5~x) )
```



‖

2 Robust Regression

We can also write the OLS as

$$Q_2 = \sum_{i=1}^{n} \{Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}\}^2 = \sum_{i=1}^{n} \rho\big(Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}\big), \tag{11.4}$$

where $\rho(t) = t^2$.

Differentiating (11.4) with respect to $\boldsymbol{\beta}$, we have

$$\frac{\partial Q_2}{\partial \boldsymbol{\beta}} = -\sum_{i=1}^{n} \psi\big(Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}\big)\mathbf{x}_i{}'$$

where $\psi(t) = \rho'(t)$. Thus, the OLS estimator is also obtained by solving

$$\sum_{i=1}^{n} \psi\big(Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}\big)\mathbf{x}_i{}' = \mathbf{0}. \tag{11.5}$$

If $\psi(t)$ is Winsorized at c ($\psi_c(t) = t$ for $|t| \leq c$ and $\psi_c(t) = c$ for $|t| > c$), then we can obtain the robustness. We can choose $c = k\sigma$. It is known that $c = 1.345\sigma$ give 95% efficiency at the normal model. (95 is often used for a magic number in statistics). It should be noted that $\psi_c(t) = \psi_k(t/\sigma) \cdot \sigma$ where $c = k\sigma$. This ψ_k is also known as the Huber's ψ function with the tuning constant k. The equation (11.5) can be rewritten as

$$\sum_{i=1}^{n} \psi_c\big(Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}\big)\mathbf{x}_i{}' = \sum_{i=1}^{n} \frac{\psi_k\Big(\frac{Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}}{\sigma}\Big)\sigma}{\frac{Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}}{\sigma}} \Big(\frac{Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}}{\sigma}\Big)\mathbf{x}_i{}' = \mathbf{0}.$$

For convenience, let $u_i = (Y_i - \mathbf{x}_i{}'\boldsymbol{\beta})/\sigma$. Then we have

$$\sum_{i=1}^{n} \frac{\psi_k(u_i)}{u_i}(Y_i - \mathbf{x}_i{}'\boldsymbol{\beta})\mathbf{x}_i{}' = \sum_{i=1}^{n} w_k(u_i) \cdot (Y_i - \mathbf{x}_i{}'\boldsymbol{\beta})\mathbf{x}_i{}' = \mathbf{0},$$

which is in a form of the WLS. The weight function $w(t)$ is also known as the Huber weight function which is given by

$$w_k(u) = \begin{cases} 1 & \text{if} \quad |u| \leq k \\ \frac{k}{|u|} & \text{if} \quad |u| > k \end{cases},$$

where k is usually given by $k = 1.345$.

The problem is how to find the weights $w_k(u_i)$ and solve the equation above. An iterative method (iteratively reweighted least squares, IRLS) can be applied. Let m be the m-th step in the iterative algorithm. Let $\boldsymbol{\beta}^{(m)}$ be the estimate of the parameter vector and $\hat{\sigma}^{(m)}$ be the scale estimate obtained at the m-th step. Denote $u_i^{(m)} = (Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}^{(m)})/\hat{\sigma}^{(m)}$. Then the parameter vector $\boldsymbol{\beta}$ is estimate as follows.

Algorithm 1 IRLS (iteratively reweighted least squares) procedures

1: Select initial estimate $\boldsymbol{\beta}^{(0)}$ and estimate $\hat{\sigma}^{(0)}$.

The $\boldsymbol{\beta}^{(0)}$ is usually obtained using the OLS, and $\hat{\sigma}^{(0)}$ is usually obtained by the MAD of the residuals.

2: At the m-th iteration step, estimate $\boldsymbol{\beta}^{(m)}$ and $\hat{\sigma}^{(m)}$ using the WLS with the previous values ($\boldsymbol{\beta}^{(m-1)}$ and $\hat{\sigma}^{(m-1)}$).

That is, we solve the following for $\boldsymbol{\beta}$ and let the solution denote $\boldsymbol{\beta}^{(m)}$:

$$\sum_{i=1}^{n} w_k(u_i^{(m-1)}) \cdot (Y_i - \mathbf{x}_i{}'\boldsymbol{\beta})\mathbf{x}_i{}' = \mathbf{0}.$$

3: $m \leftarrow m + 1$.

4: And then repeat Steps 2 and 3.

Example 11.2. Textbook Example 1 on Page 441. The education testing service (ETS) study data set are provided. The mathematics proficiency (Y) is regressed on X_2 (home library) using the robust regression. Note that Figure 11.5 on Page 442 has a typo (X_3 in the figure should read X_2).

R

Read Data

```
> url = "https://raw.githubusercontent.com/AppliedStat/LM/master/CH11TA04.txt"
> Data=read.table(url)
> y  = Data[,2]
> X2 = Data[,4]
```

OLS

```
> X2bar = mean(X2); x2 = X2 - X2bar  # de-meaned

> # OLS
> LM0 = lm( y~x2 + I(x2^2) )
> e0 = resid(LM0)
> u0 = e0 / mad(e0)
```

WLS with Huber

```
> weight.huber  <- function(x, k=1.345) { pmin(1, k/abs(x) ) }

> # WLS: 1st iteration
> w1 = weight.huber(u0)
> LM1 = lm( y~x2 + I(x2^2), weights=w1)
> e1 = resid(LM1)

> # WLS: 2nd iteration
> u1= e1 / mad(e1)
> w2 = weight.huber(u1)
> LM2 = lm( y~x2 + I(x2^2), weights=w2)
```

```
> e2 = resid(LM2)

> # WLS: 3rd iteration
> u2= e2 / mad(e2)
> w3 = weight.huber(u2)
> LM3 = lm( y~x2 + I(x2^2), weights=w3)
> e3 = resid(LM3)

> # WLS: 4th iteration
> u3= e3 / mad(e3)
> w4 = weight.huber(u3)
> LM4 = lm( y~x2 + I(x2^2), weights=w4)
> e4 = resid(LM4)

> # WLS: 5th iteration
> u4= e4 / mad(e4)
> w5 = weight.huber(u4)
> LM5 = lm( y~x2 + I(x2^2), weights=w5)
> e5 = resid(LM5)

> # WLS: 6th iteration
> u5= e5 / mad(e5)
> w6 = weight.huber(u5)
> LM6 = lm( y~x2 + I(x2^2), weights=w6)
> e6 = resid(LM6)

> # WLS: 7th iteration
> u6= e6 / mad(e6)
> w7 = weight.huber(u6)
> LM7 = lm( y~x2 + I(x2^2), weights=w7)
> e7 = resid(LM7)
```

Iteratively Huber-Reweighted least squares

```

> # Table 11.5 (Page 444)
> round(cbind(e0,u0, w1,e1, w2, e2, w7,e7),4)
         e0      u0     w1       e1     w2       e2     w7       e7
1   -2.4109 -0.5164 1.0000  -3.7542 1.0000  -4.0354 1.0000  -4.1269
2   10.5724  2.2646 0.5939   8.4297 0.7152   7.4848 0.8601   6.7698
3    3.0454  0.6523 1.0000   1.5411 1.0000   1.1559 1.0000   0.9731
4   10.3104  2.2085 0.6090   7.3822 0.8166   5.4138 1.0000   3.6583
    ..............................
8  -20.6282 -4.4186 0.3044 -22.2929 0.2704 -22.7964 0.2526 -23.0873
    ..............................
11 -14.8358 -3.1779 0.4232 -18.3824 0.3280 -21.4286 0.2402 -24.3167
    ..............................
36 -33.6282 -7.2032 0.1867 -35.2929 0.1708 -35.7964 0.1616 -36.0873
37   2.4659  0.5282 1.0000   1.7722 1.0000   1.7627 1.0000   1.8699
38  -1.7129 -0.3669 1.0000  -2.7325 1.0000  -2.8491 1.0000  -2.8079
39   3.2658  0.6995 1.0000   3.2304 1.0000   3.2624 1.0000   3.3014
40   1.2658  0.2711 1.0000   1.2304 1.0000   1.2624 1.0000   1.3014
```

Using rlm() MASS library, which is slightly different from the above

```

> library(MASS)
> RLM1 = rlm(y~x2 + I(x2^2), method="M", scale.est="MAD", k2=1.345, maxit=1 )
> RLM1
Call:
rlm(formula = y ~ x2 + I(x2^2), scale.est = "MAD", k2 = 1.345,
    maxit = 1, method = "M")
Ran 1 iterations without convergence
Coefficients:
 (Intercept)           x2       I(x2^2)
259.38160409   1.67081807   0.06476101

> RLM7 = rlm(y~x2 + I(x2^2), method="M", scale.est="MAD", k2=1.345, maxit=7 )
> RLM7
Call:
rlm(formula = y ~ x2 + I(x2^2), scale.est = "MAD", k2 = 1.345,
```

```
    maxit = 7, method = "M")
Ran 7 iterations without convergence
Coefficients:
 (Intercept)          x2      I(x2^2)
259.42100205   1.56491518   0.08016681

> RLM  = rlm(y~x2 + I(x2^2))
> RLM
Call:
rlm(formula = y ~ x2 + I(x2^2))
Converged in 10 iterations
Coefficients:
 (Intercept)          x2      I(x2^2)
259.42112605   1.56460704   0.08021299
```

Plot to compare OLS, WLS and rlm()

```
> # Scatter plot and fitted curves
> postscript(file="Figure-11-5.eps", width=5, height=5)
> plot(X2, y)
> legend(63,285, bty="n", lty=c(1,2,2), col=c("blue","black","red"),
+        legend=c("rlm", "WLS", "OLS") )
> # From LM0 (OLS)
> curve( 258.43557+1.83272*(x-X2bar)+0.06491*(x-X2bar)^2,add=TRUE, col="red", lty=2)
>
> # From LM1 (WLS after 1st iteration)
> curve( 259.39021+ 1.67011*(x-X2bar)+0.06463*(x-X2bar)^2,add=TRUE, lty=2)
>
> # From RLM (MASS library) # same as WLS after 10th iteration.;
> curve( 259.421126+ 1.5646*(x-X2bar)+0.08021299*(x-X2bar)^2,add=TRUE, col="blue")
```

∥

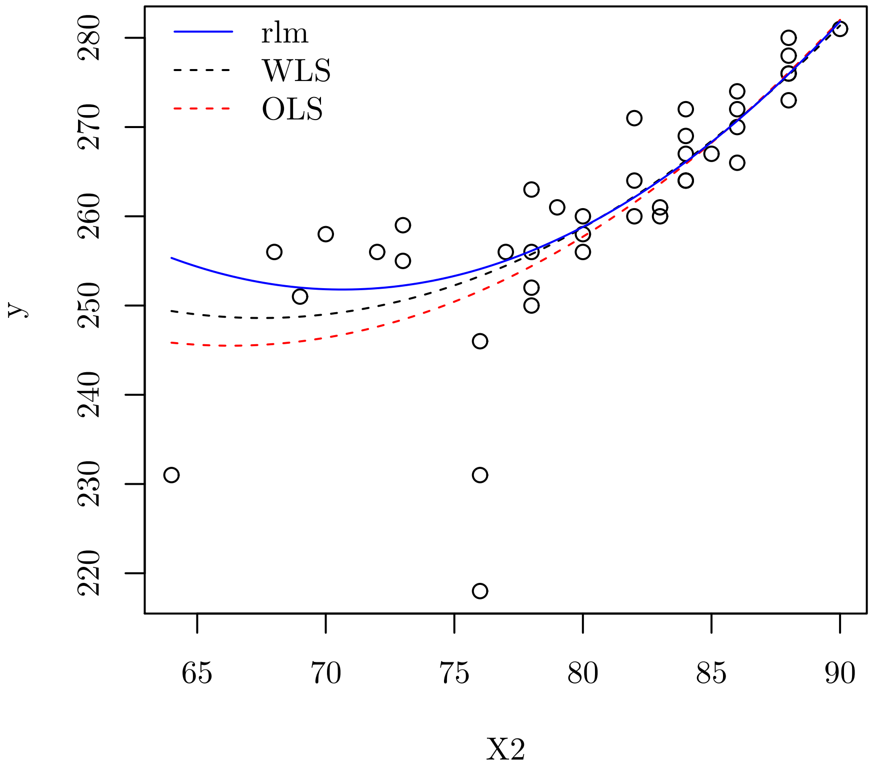
Remark 11.2. There are other ψ functions which give M-estimators (MLE-like estimators).

- Metric trimming.
- Metric Winsorizing (also called Huber).
- Tukey's biweight.
- Hampel's ψ.

△

Remark 11.3.

1. The equation in (11.5) is a regression M-estimation equation. But it is vanilla-plain or naïve so that the estimator based on (11.5) has zero breakdown point with respect to high leverage points, which is much maligned. For more details, see Huber and Ronchetti (2009).
2. To overcome the above problem, Yohai et al. (1991) develop a new regression estimator by combining the resistance of these methods with the high efficiency of M-estimation, which is called the MM-estimator.



△

References

Huber, P. J. and Ronchetti, E. M. (2009). *Robust Statistics*. John Wiley & Sons, New York, 2nd edition.

Yohai, V., Stahel, W. A., and Zamar, R. H. (1991). A procedure for robust estimation and inference in linear regression. In Stahel, W. A. and Weisberg, S. W., editors, *Directions in Robust Statistics and Diagnostics, Part II*. Springer-Verlag, New York.