1 Robust Regression

Robust regression is an alternative to ordinary least squares regression (OLS , the type of regression we have discussed thus far) when data is contaminated with outliers or influential observations and it can also be used for the purpose of detecting influential observations.

1.1 The Cheese data set

```
FitAll = lm(Taste ~ Acetic + H2S + Lactic)
plot(FitAll)
```

1.2 The stackloss data set

Brownlee's Stack Loss Plant Data contains operational data of a plant for the oxidation of ammonia to nitric acid.

The variables are:

- Air.Flow Flow of cooling air
- Water.Temp Cooling Water Inlet Temperature
- Acid.Conc. Concentration of acid [per 1000, minus 500]
- stack.loss Stack loss

```
fit.sl = lm(stack.loss ~ ., data = stackloss)

attach(stackloss)
plot(Acid.Conc. , stack.loss, pch = 18, col="red")

plot(fit.sl)
```

Robust Regression

- In robust statistics, robust regression is a form of regression analysis designed to circumvent some limitations of traditional parametric and non-parametric methods. Regression analysis seeks to find the relationship between one or more independent variables and a dependent variable.
- Certain widely used methods of regression, such as ordinary least squares, have favourable properties if their underlying assumptions are true, but can give misleading results if those assumptions are not true; thus ordinary least squares is said to be not robust to violations of its assumptions. Robust regression methods are designed to be not overly affected by violations of assumptions by the underlying data-generating process.

Usage of Robust Regression

- 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 they are 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 and including all the data points and treating all 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.
- Robust regression does not address issues of heterogeneity of variance.

Popularity of Robust Regression

Despite their superior performance over least squares estimation in many situations, robust methods for regression are still not widely used. Several reasons may help explain their unpopularity (Hampel et al. 1986, 2005). One possible reason is that there are several competing methods and the field got off to many false starts. Also, computation of robust estimates is much more computationally intensive than least squares estimation; in recent years however, this objection has become less relevant as computing power has increased greatly. Another reason may be that some popular statistical software packages failed to implement the methods (Stromberg, 2004). The belief of many statisticians that classical methods are robust may be another reason.

1.3 Fitting a robust model (`rlm`

The rlm command in the MASS package command implements several versions of robust regression.

```
summary(rlm(stack.loss ~ ., data = stackloss))
```

```
> summary(rlm(stack.loss ~ ., stackloss))

Call: rlm(formula = stack.loss ~ ., data = stackloss)
Residuals:
     Min       1Q   Median       3Q      Max
-8.91753 -1.73127  0.06187  1.54306  6.50163

Coefficients:
            Value    Std. Error t value
```

```
(Intercept) -41.0265   9.8073    -4.1832
Air.Flow      0.8294   0.1112     7.4597
Water.Temp    0.9261   0.3034     3.0524
Acid.Conc.   -0.1278   0.1289    -0.9922

Residual standard error: 2.441 on 17 degrees of freedom
```

```
rlm(stack.loss ~ ., stackloss, psi = psi.hampel, init = "lts")
```

```
> rlm(stack.loss ~ ., stackloss, psi = psi.hampel, init = "lts")
Call:
rlm(formula = stack.loss ~ ., data = stackloss, psi = psi.hampel,
    init = "lts")
Converged in 10 iterations

Coefficients:
(Intercept)    Air.Flow  Water.Temp  Acid.Conc.
-40.4748037   0.7410863   1.2250703  -0.1455242

Degrees of freedom: 21 total; 17 residual
Scale estimate: 3.09
```

1.4 Using Other *Psi* Operators

- huber
- bisquare
- hampel

Fitting is done by ***iterated re-weighted least squares (IWLS).***

Psi functions are supplied for the Huber, Hampel and Tukey bisquare proposals as `psi.huber`, `psi.hampel` and `psi.bisquare`. Huber's corresponds to a convex optimization problem and gives a unique solution (up to collinearity). The other two will have multiple local minima, and a good starting point is desirable.

```
rlm(stack.loss ~ ., stackloss, psi = psi.bisquare)
```

```
Call:
rlm(formula = stack.loss ~ ., data = stackloss, psi = psi.bisquare)
Converged in 11 iterations

Coefficients:
(Intercept)    Air.Flow  Water.Temp  Acid.Conc.
-42.2852537   0.9275471   0.6507322  -0.1123310

Degrees of freedom: 21 total; 17 residual
Scale estimate: 2.28
```

1.5 Implementation of Robust 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 they are from a different population than most of our data. So we have no proper 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 and including all the data points and treating all them equally in OLS regression. The idea of robust regression is to weigh the observations differently based on how well behaved these observations are.
- 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 (i.e. a two step process , first fitting a linear model, then a robust model to correct for the influence of outliers).
- Robust regression is done by iterated re-weighted least squares (IRLS). The rlm command in the MASS package command implements several versions of robust regression.
- There are several weighting functions that can be used for IRLS. We are going to first use the Huber weights in this example. We will then look at the final weights created by the IRLS process. This can be very useful.
- Also we will look at an alternative weighting approach to Hubers weighting called **Bisquare weighting**.

1.5.1 Huber Weighting

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

$$w(e) = 1 \text{ for} |e| \leq k \tag{1}$$

$$w(e) = \frac{k}{|e|} \text{ for} |e| > k \tag{2}$$

The value k for the Huber method is called a ***tuning constant***; smaller values of k produce more resistance to outliers, but at the expense of lower efficiency when the errors are normally distributed.

The tuning constant is generally picked to give reasonably high efficiency in the normal case; in particular,$k = 1.345\sigma$ for the Hubers method, where σ is the standard deviation of the errors) produce 95-percent efficiency when the errors are normal, and still offer protection against outliers.

```
library(MASS)
FitAll.rr = rlm(Taste ~ Acetic + H2S + Lactic)
```

```
> summary(FitAll.rr)

Call: rlm(formula = Taste ~ Acetic + H2S + Lactic)
Residuals:
    Min      1Q  Median      3Q     Max
-16.163  -5.612  -1.153   5.487  27.106

Coefficients:
            Value    Std. Error t value
(Intercept) -20.7529  20.1001    -1.0325
Acetic       -1.5331   4.5422    -0.3375
H2S           4.0515   1.2715     3.1864
Lactic       20.1459   8.7885     2.2923
```

```
Residual standard error: 8.471 on 26 degrees of freedom
```

Regression Equation:

$$\hat{Taste} = -20.75 - 1.53Acetic + 4.05H2S + 20.14Lactic$$

From before, we noticed that observations 15 , 12 and 8 were influential in the determination of the coefficients. The following table indicates the weight given to each observation when using robust regression.

We can see that roughly, as the absolute residual goes down, the weight goes up. In other words, cases with a large residuals tend to be down-weighted.

```
hweights <- data.frame(Taste = Taste,
          resid = FitAll.rr$resid,
          weight = FitAll.rr$w)
hweights2 <- hweights[order(FitAll.rr$w), ]
```

```
> hweights2[1:15, ]
   Taste      resid    weight
15  54.9  27.105636 0.4203556
12  57.2  17.518919 0.6504044
8   21.9 -16.162753 0.7049043
3   39.0  14.318512 0.7957592
18   6.4 -13.609277 0.8371707
28   0.7 -11.410452 0.9985018
1   12.3   9.990965 1.0000000
2   20.9  -1.692664 1.0000000
4   47.9  10.648009 1.0000000
```

```
5     5.6  -1.866642 1.0000000
6    25.9   2.632602 1.0000000
7    37.3   5.744433 1.0000000
9    18.1   4.775657 1.0000000
10   21.0   1.048052 1.0000000
11   34.9   5.723592 1.0000000
```

1.5.2 Implementation with Bisquare Weighting

Implementing with bisquare weighting simply requires the specification of the additional argument, as per the code below, highlighted in green)

```
FitAll.rr.2 = rlm(Taste ~ Acetic + H2S + Lactic,
 psi = psi.bisquare)
```

```
> summary(FitAll.rr.2)

Call: rlm(formula = Taste ~ Acetic + H2S + Lactic,
 psi = psi.bisquare)
Residuals:
     Min       1Q   Median       3Q      Max
-15.7034  -5.1552  -0.9793   5.6933  27.7661

Coefficients:
            Value    Std. Error t value
(Intercept) -17.7730  20.7031    -0.8585
Acetic       -2.2650   4.6784    -0.4841
H2S           4.0569   1.3096     3.0977
Lactic       20.6885   9.0522     2.2855
```

```
Residual standard error: 7.878 on 26 degrees of freedom
```

Weights using Bisquare estimator.

```
> hweights2[1:15, ]
   Taste      resid    weight
15  54.9  27.766087 0.1884633
12  57.2  18.182810 0.5735669
8   21.9 -15.703388 0.6707319
3   39.0  14.384429 0.7193235
18   6.4 -13.462286 0.7516310
28   0.7 -11.190438 0.8246092
19  18.0 -11.112316 0.8269297
4   47.9  10.860173 0.8343637
1   12.3   9.852297 0.8625704
20  38.9  -8.952091 0.8858015
14  25.9   8.588121 0.8946576
30   5.5  -8.019522 0.9078077
7   37.3   6.329446 0.9420556
11  34.9   5.999726 0.9478611
13   0.7  -5.470990 0.9565447
```

1.5.3 Conclusion

- We can see that the weight given to some observations is dramatically lower using the bisquare weighting function than the Huber weighting function and the coefficient 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 the model parameters are being highly influenced by outliers.
- Different functions have advantages and drawbacks. Huber weights can have difficulties with severe outliers, and bisquare weights can have difficulties converging or may yield multiple solutions.