Robust regression

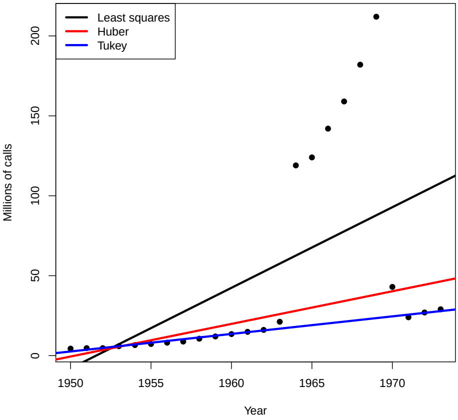
Patrick Breheny

December 1

Belgian phone calls

- We begin our discussion of robust regression with a simple motivating example dealing with the number of phone calls made per year in Belgium
- The data set `phones.txt` contains two columns:
  - `year`
  - `calls`: Number of calls made (in millions)
- As it turns out, there is a flaw in the data – for a period of time from 1964-1969, the total *length* of calls was recorded instead of the number

Belgian phone calls: Linear vs. robust regression



Robust loss

- Robust regression methods achieve their robustness by modifying the loss function
- The linear regression loss function, $l(\mathbf{r}) = \sum_i r_i^2$, increases sharply with the size of the residual
- One alternative is to use the absolute value as a loss function instead of squaring the residual: $l(\mathbf{r}) = \sum_i |r_i|$
- This achieves robustness, but is hard to work with in practice because the absolute value function is not differentiable

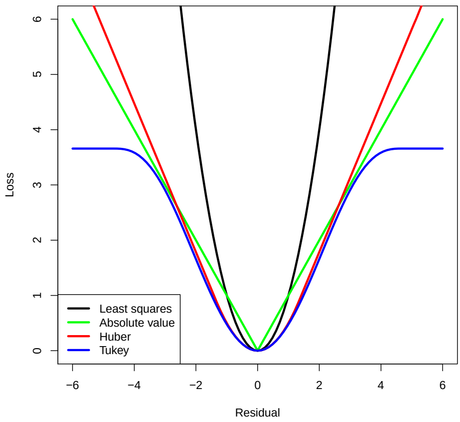
Huber's loss function

- An elegant compromise between these two loss functions was proposed by Peter Huber in 1964 $l(\mathbf{r}) = \sum_i \rho(r_i)$, where

$$\rho(r_i) = \begin{cases} r_i^2 & \text{if } |r_i| \leq c \\ c(2|r_i| - c) & \text{if } |r_i| > c \end{cases}$$

- Huber argued that $c = 1.345$ is a good choice, and showed that asymptotically, it is 95% as efficient as least squares if the true distribution is normal (and much more efficient in many other cases)

Loss functions



Tukey's biweight

- The last loss function, proposed by Tukey and known as *Tukey's biweight* or *Tukey's bisquare*, is given by:

$$\rho'(r_i) = \begin{cases} r_i \left\{1 - \left(\frac{r_i}{c}\right)^2\right\}^2 & \text{if } |r_i| \leq c \\ 0 & \text{if } |r_i| > c \end{cases}$$

- The value $c = 4.685$ is usually used for this loss function, and again, it provides an asymptotic efficiency 95% that of linear regression for the normal distribution

M-estimators

- Huber's and Tukey's estimators fall under the general category of what are called *M-estimators*, because they are obtained by (M)inimizing a loss function, or equivalently, solving

$$\sum_i \psi(r_i)\mathbf{x}_i = \mathbf{0},$$

  where $\psi = \rho'$
- Note that the function ψ defines the M-estimator; this function shows up constantly in the theory of M-estimators
- Note also that "M-estimators" are a rather broad class; for example, all MLEs are M-estimators
- In particular, note that linear regression is an M-estimator with $\psi(r_i) = r_i$

The IRLS algorithm for robust regression

- There are closed form solutions and fast algorithms for solving the least squares problem as well as the weighted least squares problem:

$$\sum_i w_i r_i \mathbf{x}_i = \mathbf{0},$$

- Thus, a convenient way to solve for M-estimators is to use an iteratively reweighted least squares (IRLS) algorithm, in which we calculate $w_i = \psi(r_i)/r_i$, solve the weighted least squares problem, re-calculate the weights, re-solve, and so on until convergence
- It should be noted that Tukey's biweight allows for multiple local minima, and this algorithm may not converge to the global solution

Estimating the scale parameter

- The preceding derivations are slightly oversimplified, in that the arguments for setting $c = 1.345$ or 4.685 are based on the assumption that y has known variance 1
- In reality, of course, this is not true, and we must apply the loss functions to the scaled residuals – *i.e.*, replace every $\rho(r_i)$ with $\rho(r_i/s)$, and every $\psi(r_i)$ with $\psi(r_i/s)$, where s is an estimated scale parameter
- While a number of other estimators have been proposed, the simplest is based on the median absolute deviation of the residuals:

$$\text{MAD} = \text{median}\{|r_i|\},$$

where $\hat{s} = \text{MAD}/0.6745$, based on the idea that, for the standard normal, $\text{E}(\text{MAD}) = 0.6745$

Inference

- Similar to GLMs, robust regression can be shown to be asymptotically normal:

$$\sqrt{n}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{\text{d}} N(0, \mathbf{V}),$$

where $\mathbf{V}$ is the asymptotic variance-covariance matrix

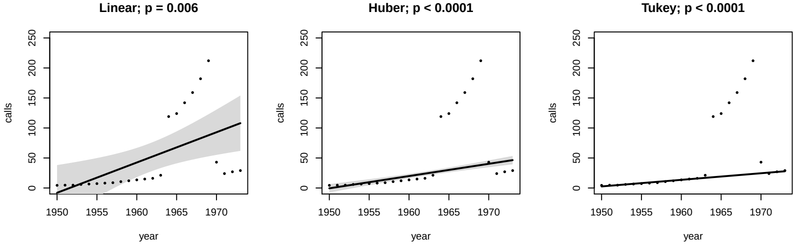
- Various estimators have been proposed to estimate $\mathbf{V}$ based on various approximations:

$$\hat{\mathbf{V}} = \hat{\sigma}^2(\mathbf{X}^T\mathbf{X})^{-1}$$
$$\hat{\mathbf{V}} = \hat{\sigma}^2(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}$$
$$\hat{\mathbf{V}} = \hat{\sigma}^2(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{X})(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1},$$

where the quantity $\hat{\sigma}^2$ is not necessarily the same in the three expression; all of which converge to the true variance $\mathbf{V}$ but have various advantages and disadvantages

Inference (cont'd)

With asymptotic normality and standard errors, we can calculate Wald-style hypothesis tests and confidence intervals in the usual way:



SAS

- In SAS, `PROC ROBUSTREG` can be used to perform robust regression; its syntax is straightforward:

```
PROC ROBUSTREG DATA=phones;
  MODEL Calls = Year;
RUN;
```

- By default, SAS uses Tukey's biweight; to specify Huber's approach, submit:

```
PROC ROBUSTREG DATA=phones METHOD=M(WF=Huber);
  MODEL Calls = Year;
RUN;
```

R

- In R, the MASS library provides the function rlm, a robust companion to lm:

```
fit <- rlm(calls~year,phones)
```

- Somewhat peculiarly, the maximum number of iterations has a default of 20 (the SAS default is 1,000); thus, you may need to increase this number using

```
fit <- rlm(calls~year,phones,maxit=50)
```

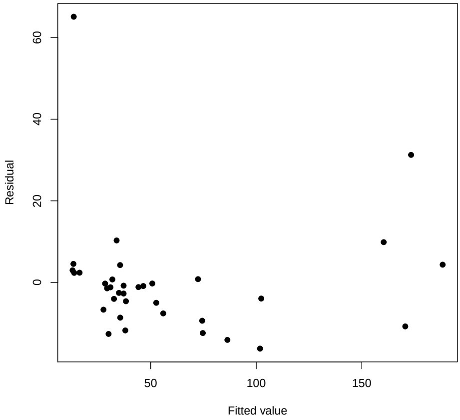
- In R, the default is Huber's approach; to obtain Tukey's biweight, use the psi option:

```
fit <- rlm(calls~year,phones,psi=psi.bisquare)
```

Scottish hill races

- Another classic data set in the outlier/robust regression literature contains information on hill racing (apparently a somewhat popular sport in Scotland)
- The data set `hills.txt` contains information on the winning times in 1984 for 35 Scottish hill races, as well as two factors which presumably influence the duration of the race:
  - `dist`: The distance of the race (in miles)
  - `climb`: The elevation change (in feet)

Residuals from OLS fit



Hill races: LS vs OLS

- Comparing the results of three estimation techniques:

	OLS		Huber		Tukey	
	β	SE	β	SE	β	SE
Dist. (1 mile)	6.22	0.60	6.55	0.25	6.64	0.21
Climb (100 ft)	1.10	0.21	0.83	0.08	0.65	0.07

- Note that there are two large outliers in this data set: as they are downweighted, there is a modest change in the estimates (the distance estimate goes up, while the climb estimate goes down), and a large drop (2-3 fold) in the standard error