POLS/CSSS 503:
Advanced Quantitative Political Methodology

Outliers and Robust Regression Techniques

Christopher Adolph

Department of Political Science
and
Center for Statistics and the Social Sciences
University of Washington, Seattle

Outliers

Suppose we find no overall pattern in the mean or variance of our residuals

But a handful of residuals look odd, as if they came from another distribution.

Outliers

Suppose we find no overall pattern in the mean or variance of our residuals

But a handful of residuals look odd, as if they came from another distribution.

For example, suppose $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\sigma^2 = 2$:

$$\mathbf{Y}_1 \quad = \quad \mathbf{X}_1\boldsymbol{\beta} + \varepsilon_1, \qquad \varepsilon_1 = -0.247$$

Outliers

Suppose we find no overall pattern in the mean or variance of our residuals

But a handful of residuals look odd, as if they came from another distribution.

For example, suppose $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\sigma^2 = 2$:

$$\begin{aligned} \mathbf{Y}_1 &= \mathbf{X}_1\boldsymbol{\beta} + \varepsilon_1, \qquad \varepsilon_1 = -0.247 \\ \mathbf{Y}_2 &= \mathbf{X}_2\boldsymbol{\beta} + \varepsilon_2, \qquad \varepsilon_2 = -0.829 \end{aligned}$$

Outliers

Suppose we find no overall pattern in the mean or variance of our residuals

But a handful of residuals look odd, as if they came from another distribution.

For example, suppose $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\sigma^2 = 2$:

$$
\begin{aligned}
\mathbf{Y}_1 &= \mathbf{X}_1\boldsymbol{\beta} + \varepsilon_1, \qquad \varepsilon_1 = -0.247 \\
\mathbf{Y}_2 &= \mathbf{X}_2\boldsymbol{\beta} + \varepsilon_2, \qquad \varepsilon_2 = -0.829 \\
\mathbf{Y}_3 &= \mathbf{X}_3\boldsymbol{\beta} + \varepsilon_3, \qquad \varepsilon_3 = 0.820 \\
&\ldots \\
\mathbf{Y}_{41} &= \mathbf{X}_{41}\boldsymbol{\beta} + \varepsilon_{41}, \qquad \varepsilon_{41} = 0.644
\end{aligned}
$$

Outliers

Suppose we find no overall pattern in the mean or variance of our residuals

But a handful of residuals look odd, as if they came from another distribution.

For example, suppose $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\sigma^2 = 2$:

$$
\begin{aligned}
\mathbf{Y}_1 &= \mathbf{X}_1\boldsymbol{\beta} + \varepsilon_1, & \varepsilon_1 &= -0.247 \\
\mathbf{Y}_2 &= \mathbf{X}_2\boldsymbol{\beta} + \varepsilon_2, & \varepsilon_2 &= -0.829 \\
\mathbf{Y}_3 &= \mathbf{X}_3\boldsymbol{\beta} + \varepsilon_3, & \varepsilon_3 &= 0.820 \\
&\ldots & & \\
\mathbf{Y}_{41} &= \mathbf{X}_{41}\boldsymbol{\beta} + \varepsilon_{41}, & \varepsilon_{41} &= 0.644 \\
\mathbf{Y}_{42} &= \mathbf{X}_{42}\boldsymbol{\beta} + \varepsilon_{42}, & \varepsilon_{42} &= 10000000
\end{aligned}
$$

Imagine these were data on the charitable contributions of randomly selected Seattle residents, and observation 42 (by chance) was Bill Gates.

Outliers

Observations 1 through 41 should cause no difficulties.

If we regressed Y on X for just these observations, we would get $\hat{\boldsymbol{\beta}}_{LS} \approx \boldsymbol{\beta}$.

Outliers

Observations 1 through 41 should cause no difficulties.

If we regressed Y on X for just these observations, we would get $\hat{\boldsymbol{\beta}}_{LS} \approx \boldsymbol{\beta}$.

But when we include observation 42, all hell breaks loose.

Outliers

Observations 1 through 41 should cause no difficulties.

If we regressed Y on X for just these observations, we would get $\hat{\boldsymbol{\beta}}_{LS} \approx \boldsymbol{\beta}$.

But when we include observation 42, all hell breaks loose.

A less extreme, graphical presentation helps fix ideas

I'll draw 10 observations from the model

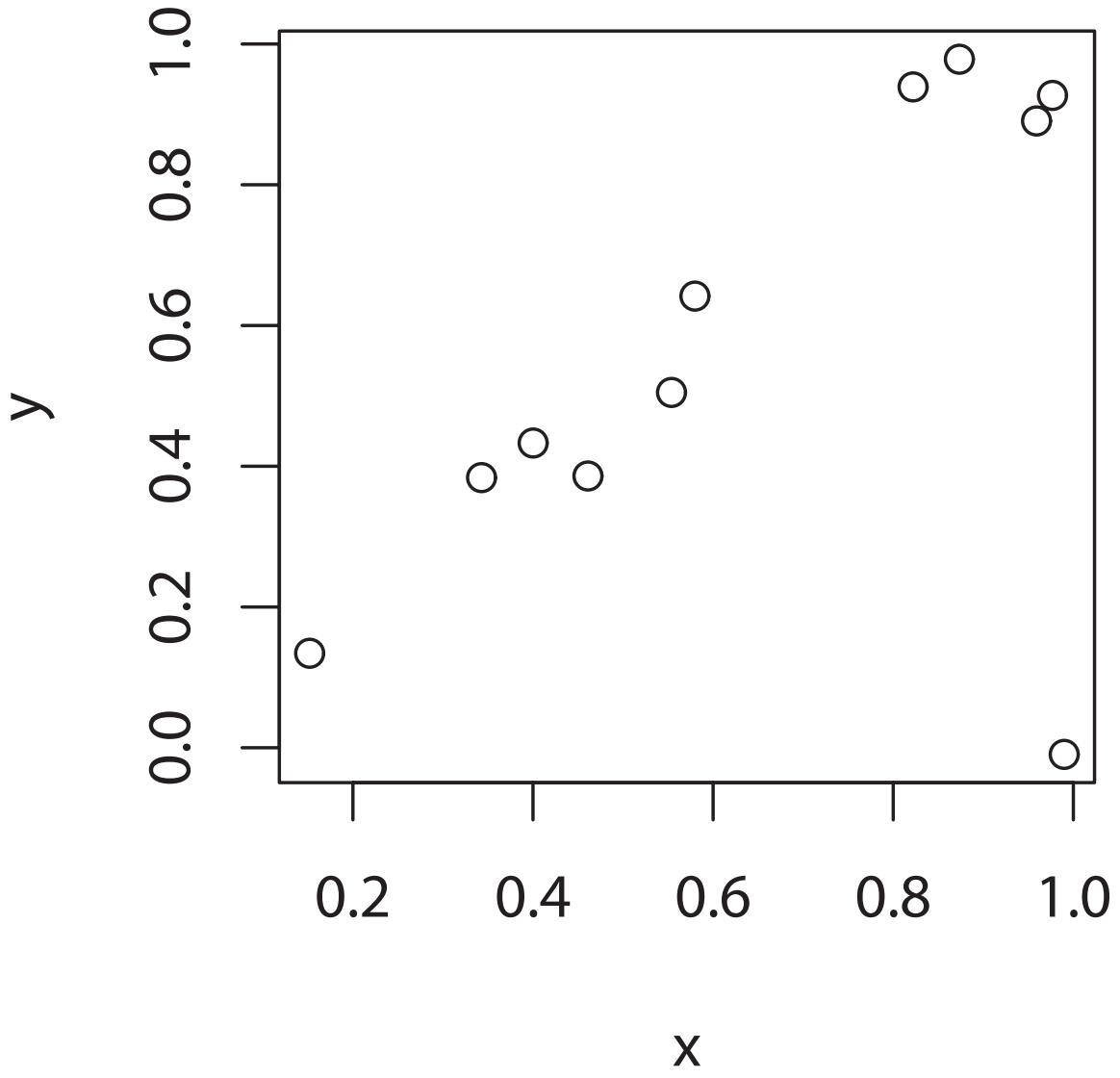
$$\begin{aligned} y_i &= 1 \times x_i + \varepsilon_i \\ \varepsilon &\sim \mathcal{N}(0, \frac{1}{15}) \end{aligned}$$

And add an observation: $x_{11} = 0.99$, $y_{11} = 0$

Notice this last observation is *sui generis*

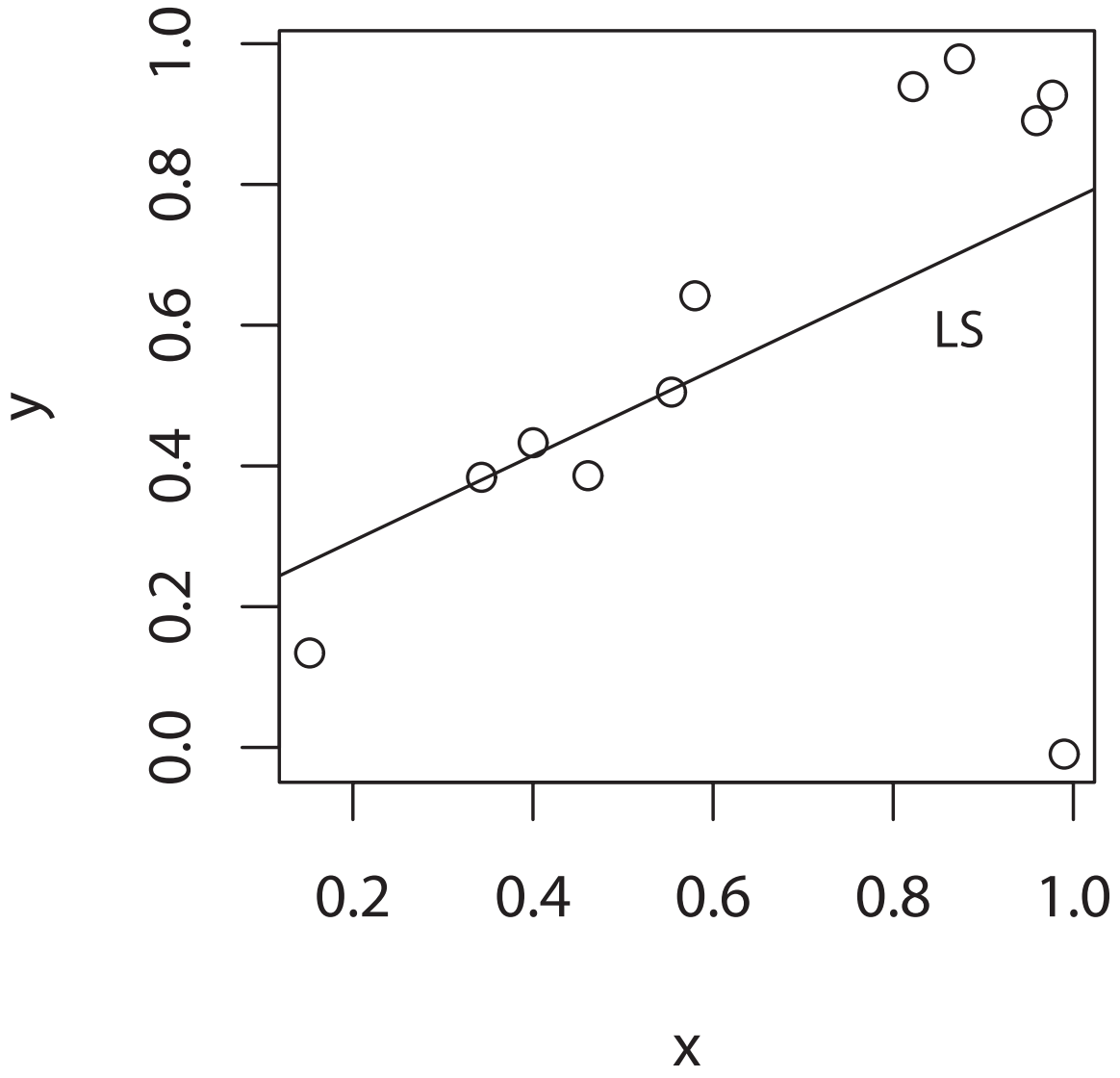
Not a product of the model that produced the other 10 cases

Artificial data with an outlier



What will least squares make of these data?

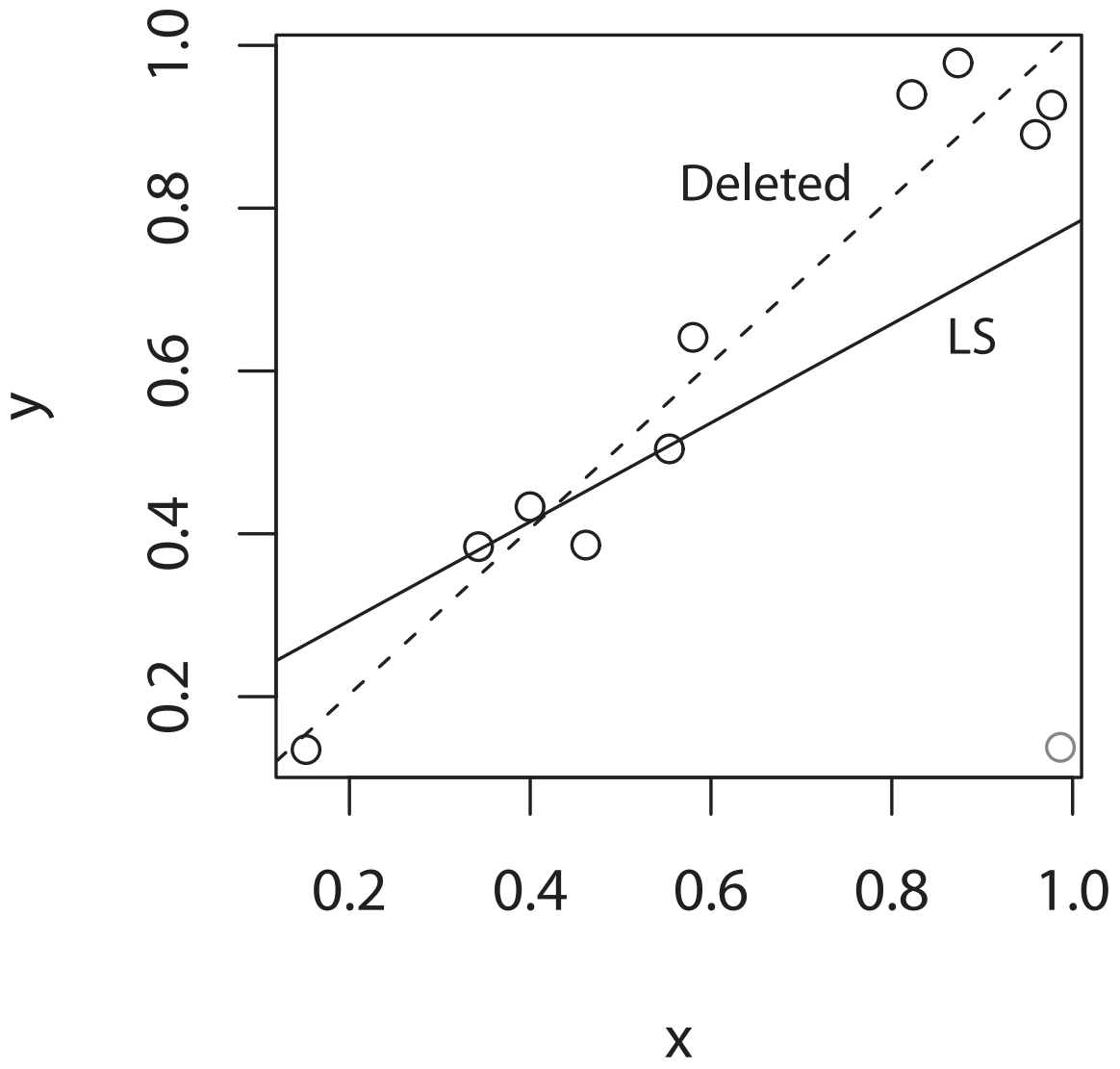
LS fit with an outlier



The outlier pulls the LS line towards itself. $\hat{\beta}_{LS} = 0.61$ $(\text{se} = 0.33)$.

That's far from the β that generated the first 10 observations.

Deleting the outlier



Deleting the outlier allows LS to fit the good data: $\hat{\beta}_{LS} = 1.02$, $(\text{se} = 0.09)$.

Outliers

Outliers are observations that come from another distribution, e.g.,

- From a different "universe" of data

Outliers

Outliers are observations that come from another distribution, e.g.,

- From a different "universe" of data
- Coder error (typo)

Outliers

Outliers are observations that come from another distribution, e.g.,

- From a different "universe" of data
- Coder error (typo)
- We omitted a variable, Z conditional on which ε_{42} comes from $\mathrm{N}(\mathbf{x}_{42}\beta, \sigma^2)$
  For the Bill Gates outlier, this could be because we omitted $\log(\mathrm{income})$

Outliers

Outliers are observations that come from another distribution, e.g.,

- From a different "universe" of data
- Coder error (typo)
- We omitted a variable, Z conditional on which ε_{42} comes from $\mathrm{N}(\mathbf{x}_{42}\beta, \sigma^2)$
  For the Bill Gates outlier, this could be because we omitted $\log(\mathrm{income})$
- ε_{42} was just a really really unusual event (a "black swan")

Outliers

Outliers are observations that come from another distribution, e.g.,

- From a different "universe" of data
- Coder error (typo)
- We omitted a variable, Z conditional on which ε_{42} comes from $\mathrm{N}(\mathbf{x}_{42}\beta, \sigma^2)$
  For the Bill Gates outlier, this could be because we omitted $\log(\mathrm{income})$
- ε_{42} was just a really really unusual event (a "black swan")

The best approach depends on what caused the data to be an outlier

But we can't always tell the cause

Outliers

Outliers are observations that come from another distribution, e.g.,

- From a different "universe" of data
- Coder error (typo)
- We omitted a variable, Z conditional on which ε_{42} comes from $\mathrm{N}(\mathbf{x}_{42}\beta, \sigma^2)$
  For the Bill Gates outlier, this could be because we omitted $\log(\mathrm{income})$
- ε_{42} was just a really really unusual event (a "black swan")

The best approach depends on what caused the data to be an outlier

But we can't always tell the cause

And in the multivariate case, hard to tell whether an obs is an outlier at all

The Multivariate Case

To find outliers in multivariate regressions, we need special tools

Tool 1: A measure of **leverage**

How much weight does an observation carry in LS?

A function of the distance of the obsevation from the mean in the space of the X's.

The Multivariate Case

To find outliers in multivariate regressions, we need special tools

Tool 1: A measure of **leverage**

How much weight does an observation carry in LS?

A function of the distance of the obsevation from the mean in the space of the X's.

Tool 2: A measure of **discrepancy**

How "outlying" is each residual? Mainly a function of the size of the residual

The Multivariate Case

To find outliers in multivariate regressions, we need special tools

Tool 1: A measure of **leverage**

How much weight does an observation carry in LS?

A function of the distance of the obsevation from the mean in the space of the X's.

Tool 2: A measure of **discrepancy**

How "outlying" is each residual? Mainly a function of the size of the residual

Putting these tools together tells us the *influence* of an observation

$$\text{Influence} = \text{Leverage} \times \text{Discrepancy}$$

Influence:
How much does the observation affect (or distort) the regression surface (line)?

Leverage

In the 2D case, the more extreme the X_i, the more leverage of Y_i on $\hat{\beta}_{LS}$

Leverage

In the 2D case, the more extreme the X_i, the more leverage of Y_i on $\hat{\beta}_{LS}$

This generalizes to the multidimensional case:

The greater the distance of $\mathbf{X}_i$ from $\bar{\mathbf{X}}_i$, the greater the leverage

Leverage

In the 2D case, the more extreme the X_i, the more leverage of Y_i on $\hat{\beta}_{LS}$

This generalizes to the multidimensional case:

The greater the distance of $\mathbf{X}_i$ from $\bar{\mathbf{X}}_i$, the greater the leverage

A simple way to measure this distance is the *hat matrix*, which is derived as:

Leverage

In the 2D case, the more extreme the X_i, the more leverage of Y_i on $\hat{\beta}_{LS}$

This generalizes to the multidimensional case:

The greater the distance of $\mathbf{X}_i$ from $\bar{\mathbf{X}}_i$, the greater the leverage

A simple way to measure this distance is the *hat matrix*, which is derived as:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$$

Leverage

In the 2D case, the more extreme the X_i, the more leverage of Y_i on $\hat{\beta}_{LS}$

This generalizes to the multidimensional case:

The greater the distance of $\mathbf{X}_i$ from $\bar{\mathbf{X}}_i$, the greater the leverage

A simple way to measure this distance is the *hat matrix*, which is derived as:

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{X}\hat{\boldsymbol{\beta}} \\ \hat{\mathbf{y}} &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \end{aligned}$$

Leverage

In the 2D case, the more extreme the X_i, the more leverage of Y_i on $\hat{\beta}_{LS}$

This generalizes to the multidimensional case:

The greater the distance of $\mathbf{X}_i$ from $\bar{\mathbf{X}}_i$, the greater the leverage

A simple way to measure this distance is the *hat matrix*, which is derived as:

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{X}\hat{\boldsymbol{\beta}} \\ \hat{\mathbf{y}} &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \\ \hat{\mathbf{y}} &= \mathbf{H}\mathbf{y} \end{aligned}$$

Leverage

In the 2D case, the more extreme the X_i, the more leverage of Y_i on $\hat{\beta}_{LS}$

This generalizes to the multidimensional case:

The greater the distance of $\mathbf{X}_i$ from $\bar{\mathbf{X}}_i$, the greater the leverage

A simple way to measure this distance is the *hat matrix*, which is derived as:

$$\begin{aligned}
\hat{\mathbf{y}} &= \mathbf{X}\hat{\boldsymbol{\beta}} \\
\hat{\mathbf{y}} &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \\
\hat{\mathbf{y}} &= \mathbf{H}\mathbf{y} \\
\mathbf{H} &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'
\end{aligned}$$

so called the hat matrix because it transforms $\mathbf{y}$ to $\hat{\mathbf{y}}$

Leverage

In the 2D case, the more extreme the X_i, the more leverage of Y_i on $\hat{\beta}_{LS}$

This generalizes to the multidimensional case:

The greater the distance of $\mathbf{X}_i$ from $\bar{\mathbf{X}}_i$, the greater the leverage

A simple way to measure this distance is the *hat matrix*, which is derived as:

$$\begin{aligned}
\hat{\mathbf{y}} &= \mathbf{X}\hat{\boldsymbol{\beta}} \\
\hat{\mathbf{y}} &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \\
\hat{\mathbf{y}} &= \mathbf{H}\mathbf{y} \\
\mathbf{H} &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'
\end{aligned}$$

so called the hat matrix because it transforms $\mathbf{y}$ to $\hat{\mathbf{y}}$

The diagonal elements of the hat matrix (the h_i's) are proportional to the distance between $\mathbf{X}_i$ from $\bar{\mathbf{X}}_i$

Hence h_i is a simple measure of the leverage of Y_i

An aside on distances in multiple dimensions

When we say that obs i lies far away from the average observation on multiple dimensions x_1, x_2, what do we mean?

Aside on distances in multiple dimensions

Our usual measure of how far apart two things are is *Euclidian distance*

Suppose we want the distance between a covariate x_j and its mean, $\bar{x}_j$

In one dimension:

$$d_{\text{Euclid}} = x_1 - \bar{x}_1$$

Aside on distances in multiple dimensions

Our usual measure of how far apart two things are is *Euclidian distance*

Suppose we want the distance between a covariate x_j and its mean, $\bar{x}_j$

In one dimension:

$$d_{\text{Euclid}} = x_1 - \bar{x}_1$$

For many dimensions:

$$d_{\text{Euclid}} = \sqrt{\sum_j (x_j - \bar{x}_j)^2}$$

Aside on distances in multiple dimensions

Our usual measure of how far apart two things are is *Euclidian distance*

Suppose we want the distance between a covariate x_j and its mean, $\bar{x}_j$

In one dimension:

$$d_{\text{Euclid}} = x_1 - \bar{x}_1$$

For many dimensions:

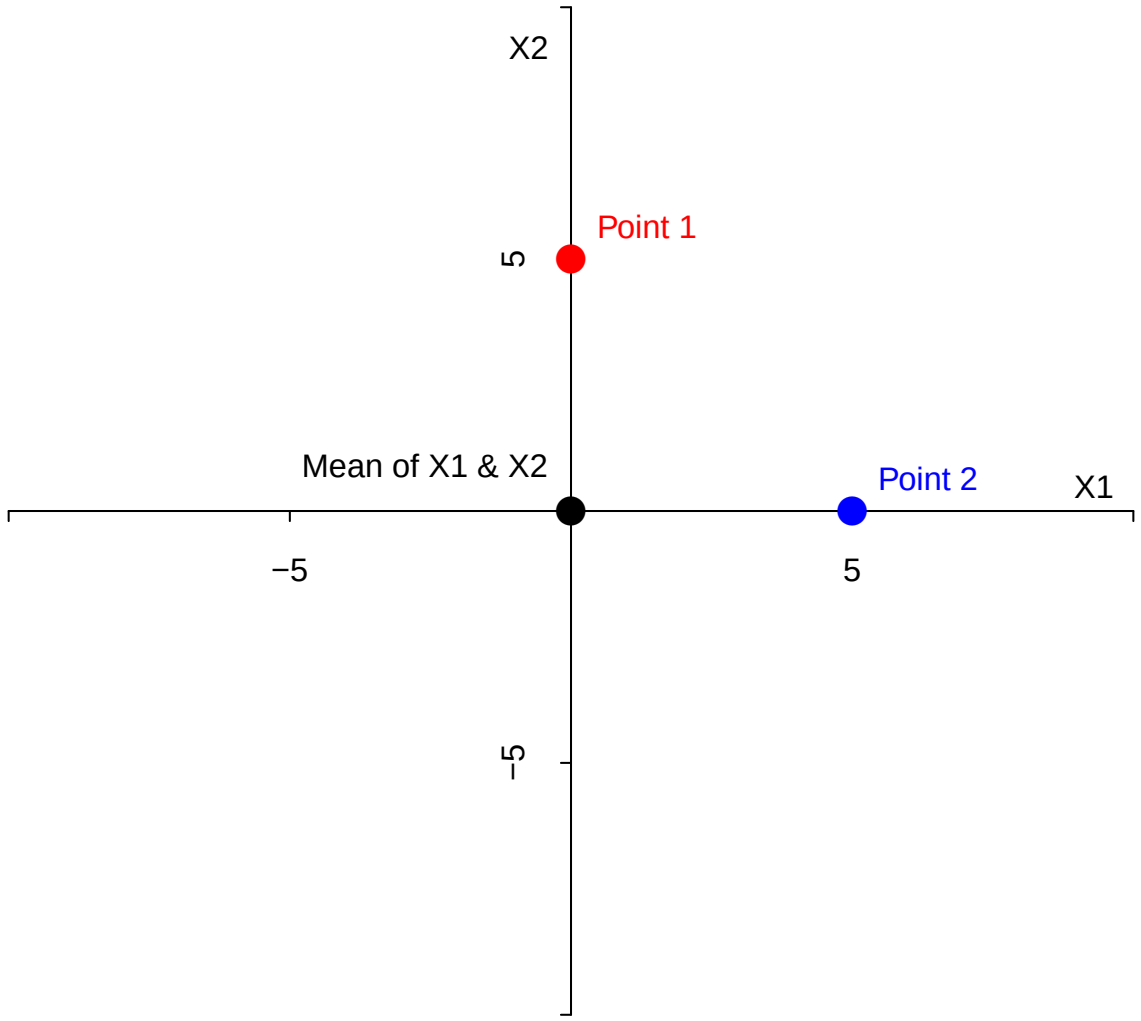
$$d_{\text{Euclid}} = \sqrt{\sum_j (x_j - \bar{x}_j)^2}$$

This is the Pythagorean Theorem, which in matrix form is:

$$D_{\text{Euclid}} = \sqrt{(\mathbf{X} - \bar{\mathbf{x}})'(\mathbf{X} - \bar{\mathbf{x}})}$$

NB: This is also known as the *norm* of X, and is written as $||X||$

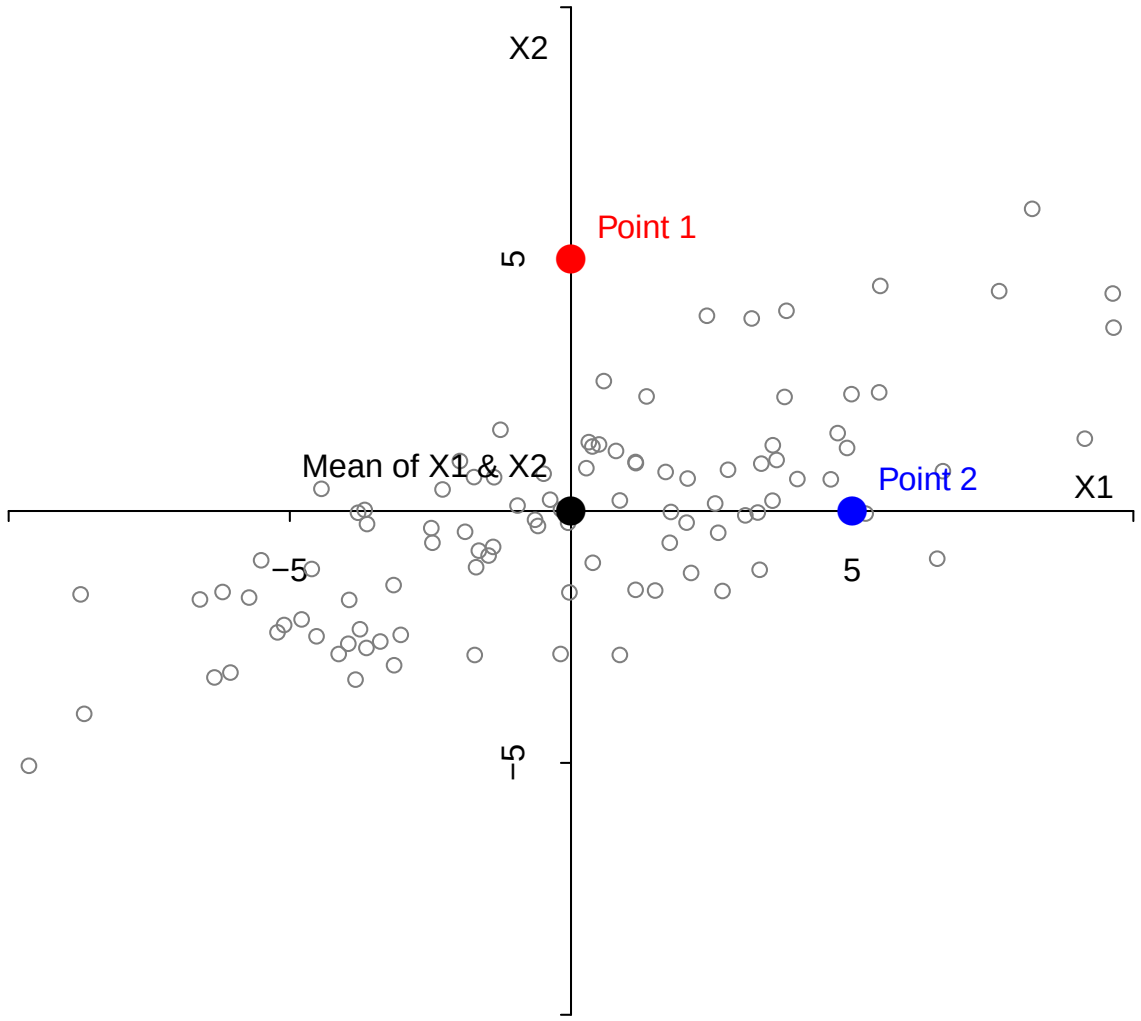
Aside on discrepancy in multiple dimensions



If we use Euclidean distance as our measure of discrepancy, we hold both of the above points to be equally discrepant

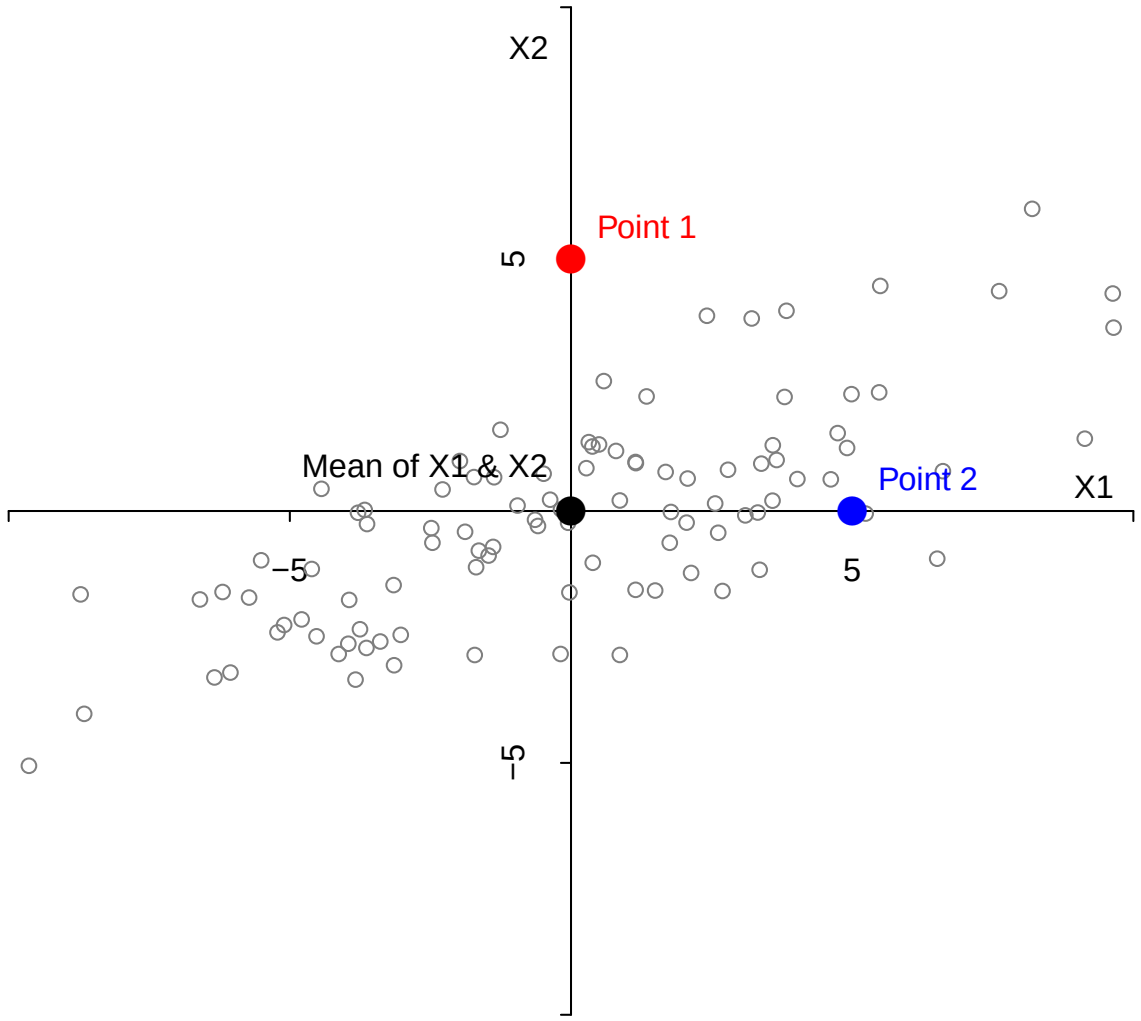
Is this safe? Sensible?

Aside on discrepancy in multiple dimensions



What if the data look like this? Are these points equally "far out"?

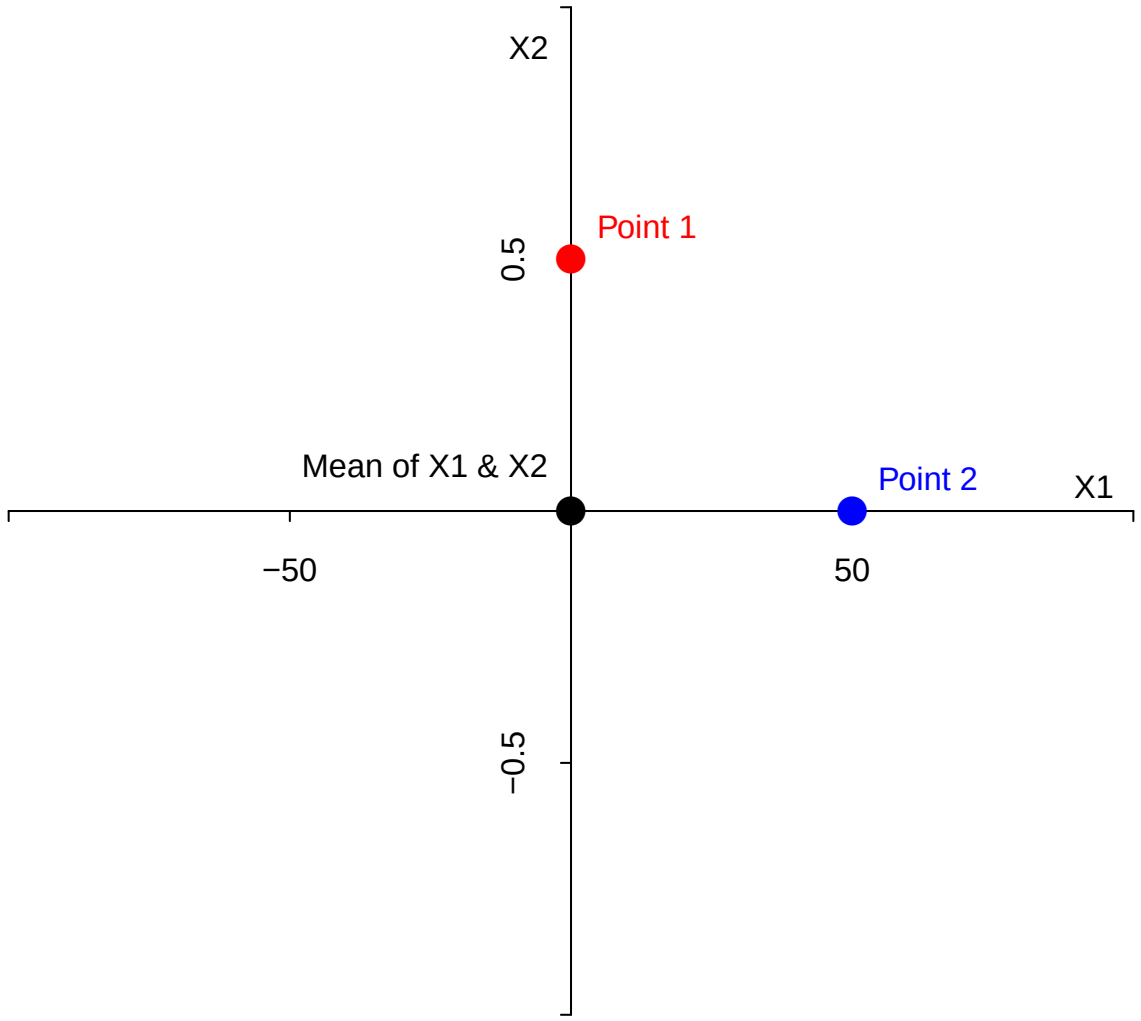
Aside on discrepancy in multiple dimensions



No. Distance from a distribution depends on the variance of x_1 in all cases,

as well as the covariance of x_1 and x_2 (etc.) in the multidimensional case

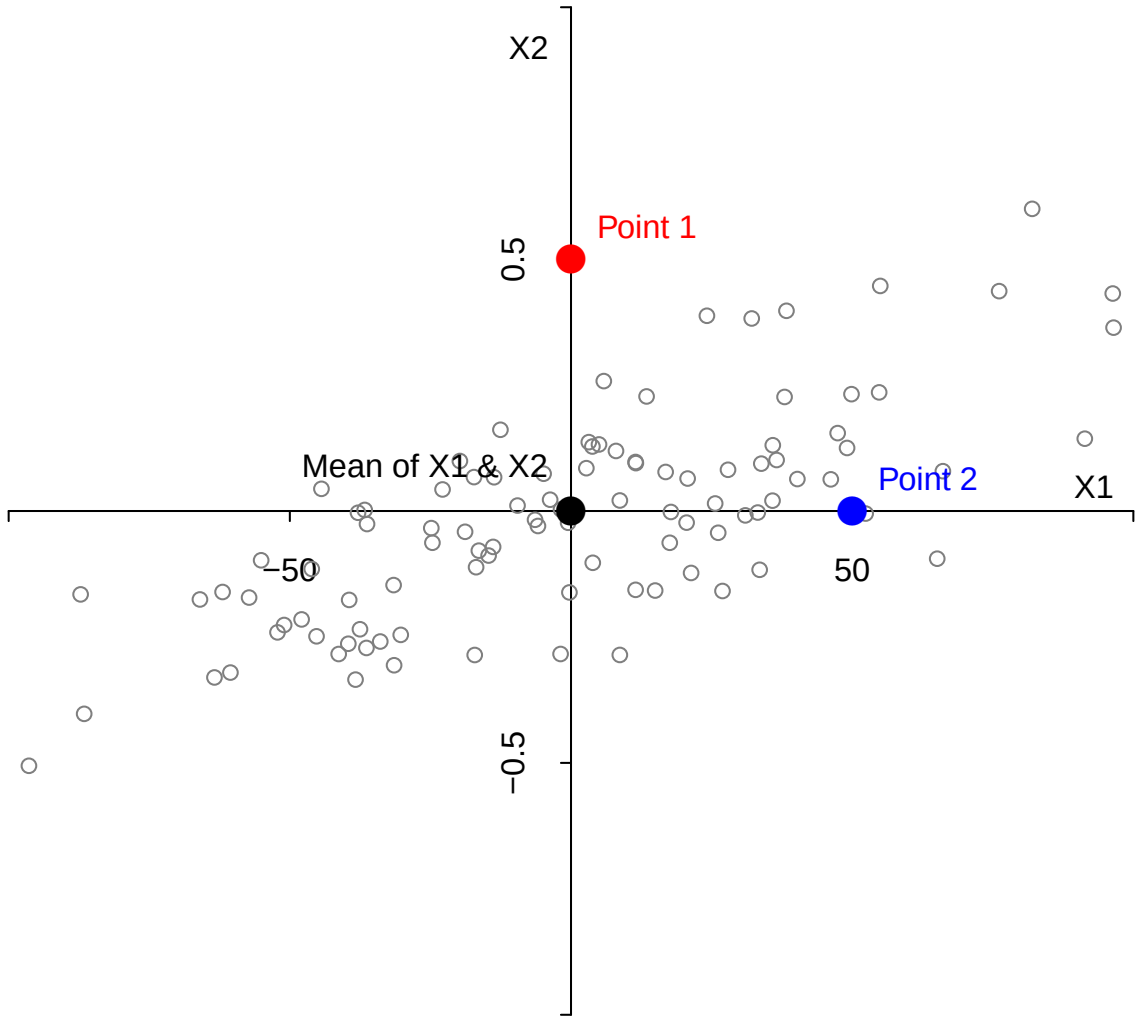
Aside on discrepancy in multiple dimensions



Now suppose the scales were adjusted, Point 2 looks much further out.

Euclid says this make Point 2 farther from the mean than Point 1.
Is he right?

Aside on discrepancy in multiple dimensions



Not unless we're talking about map coordinates.

For social variables, the scales may differ radically.
Suppose X1 is $k of income and X2 is % of family members with a college degree.

Aside on discrepancy in multiple dimensions

When we say that obs i lies far away from the average observation on multiple dimensions x_1, x_2, what do we mean?

We *don't* mean Euclidian distance

Two reasons:

1. If one x_j has a numerically wider scale (e.g., dollars instead of a 0-1 scale), it will dominate the above calculation

2. Outlyingness isn't just about the distance from the mean, but takes into account variance and covariance.

To solve these problems,
we need a scale-invariant measure of multidimensional distance.

Mahalanobis distance

We need a scale-invariant measure of multidimensional distance.

Mahalanobis distance is a good option:

In one dimension, this is just a generalization of Euclid that standardizes for variance:

$$d_{\text{Mahalanobis}} = \frac{x_1 - \bar{x}_1}{\text{sd}(x_1)}$$

Mahalanobis distance

We need a scale-invariant measure of multidimensional distance.

Mahalanobis distance is a good option:

In one dimension, this is just a generalization of Euclid that standardizes for variance:

$$d_{\text{Mahalanobis}} = \frac{x_1 - \bar{x}_1}{\text{sd}(x_1)}$$

In multiple dimensions, we again need to average squared distances, but now we normalize those distances by the variances and covariance first by "dividing" them out:

$$D_{\text{Mahalanobis}} = \sqrt{(\mathbf{X} - \bar{\mathbf{x}})' \text{Var}(\mathbf{X})^{-1} (\mathbf{X} - \bar{\mathbf{x}})}$$

Mahalanobis distance

$$D_{\text{Mahalanobis}} = \sqrt{(\mathbf{X} - \bar{\mathbf{x}})' \text{Var}(\mathbf{X})^{-1} (\mathbf{X} - \bar{\mathbf{x}})}$$

Mahalanobis distance

$$D_{\text{Mahalanobis}} = \sqrt{(\mathbf{X} - \bar{\mathbf{x}})' \text{Var}(\mathbf{X})^{-1} (\mathbf{X} - \bar{\mathbf{x}})}$$

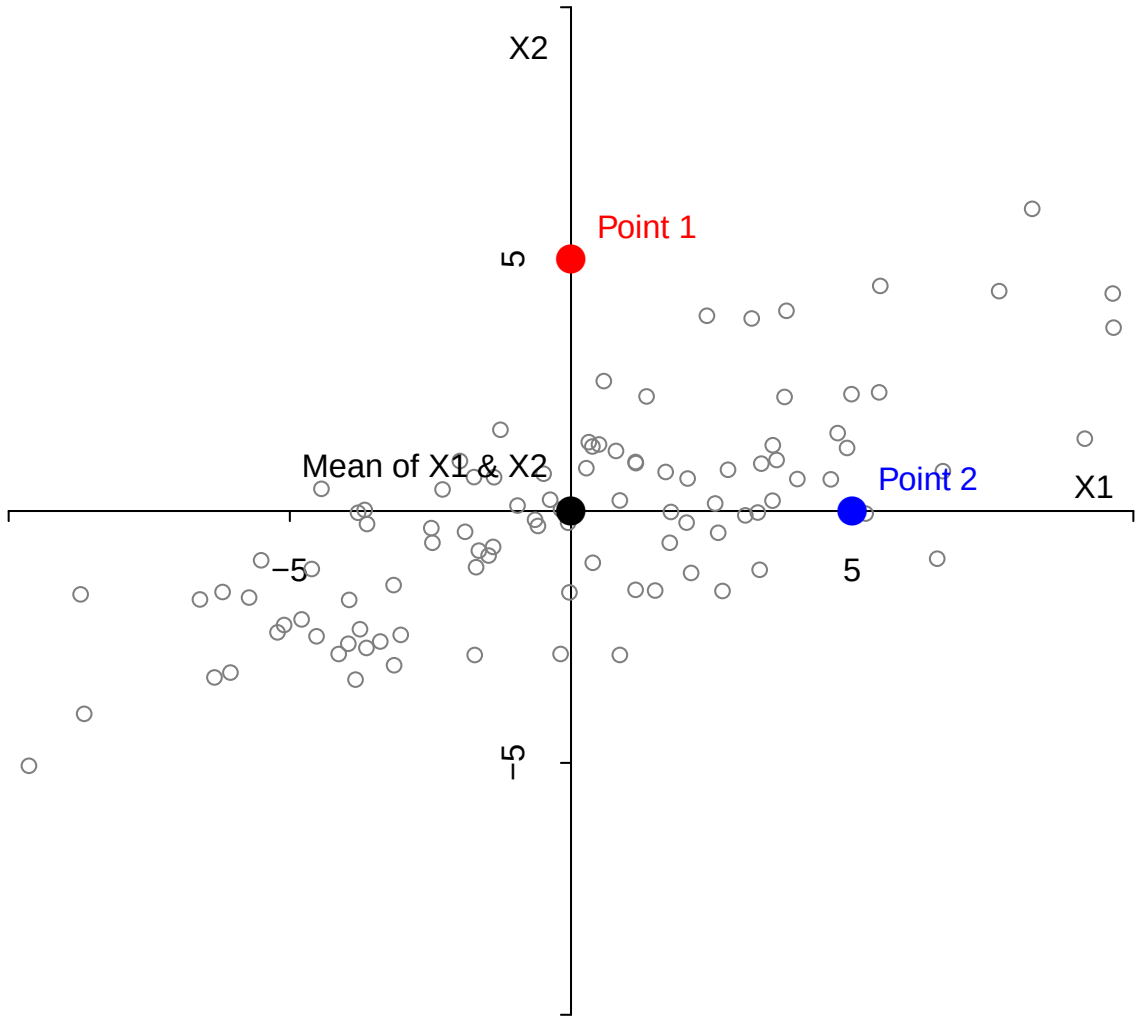
Mahalanobis distance is closely related to hat matrix leverage h_i:

$$h_i = \frac{d^2_{\text{Mahalanobis}}}{n-1} + \frac{1}{n}$$

So h_i is measuring just how far out a given observation is in the Mahalanobis space created by the X's

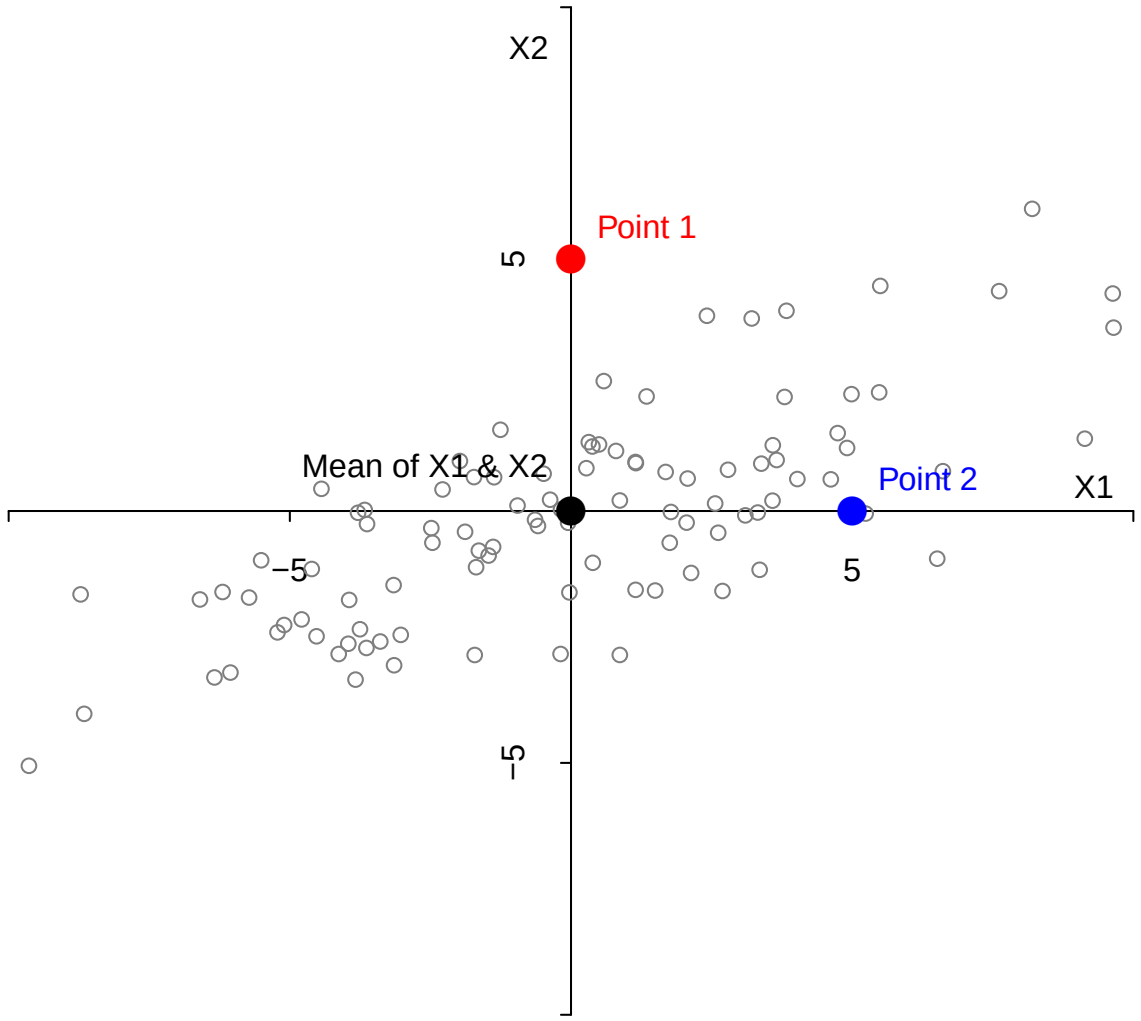
Note that we **do not** take the dependent variable Y into consideration when calculating h_i

Mahalanobis in practice



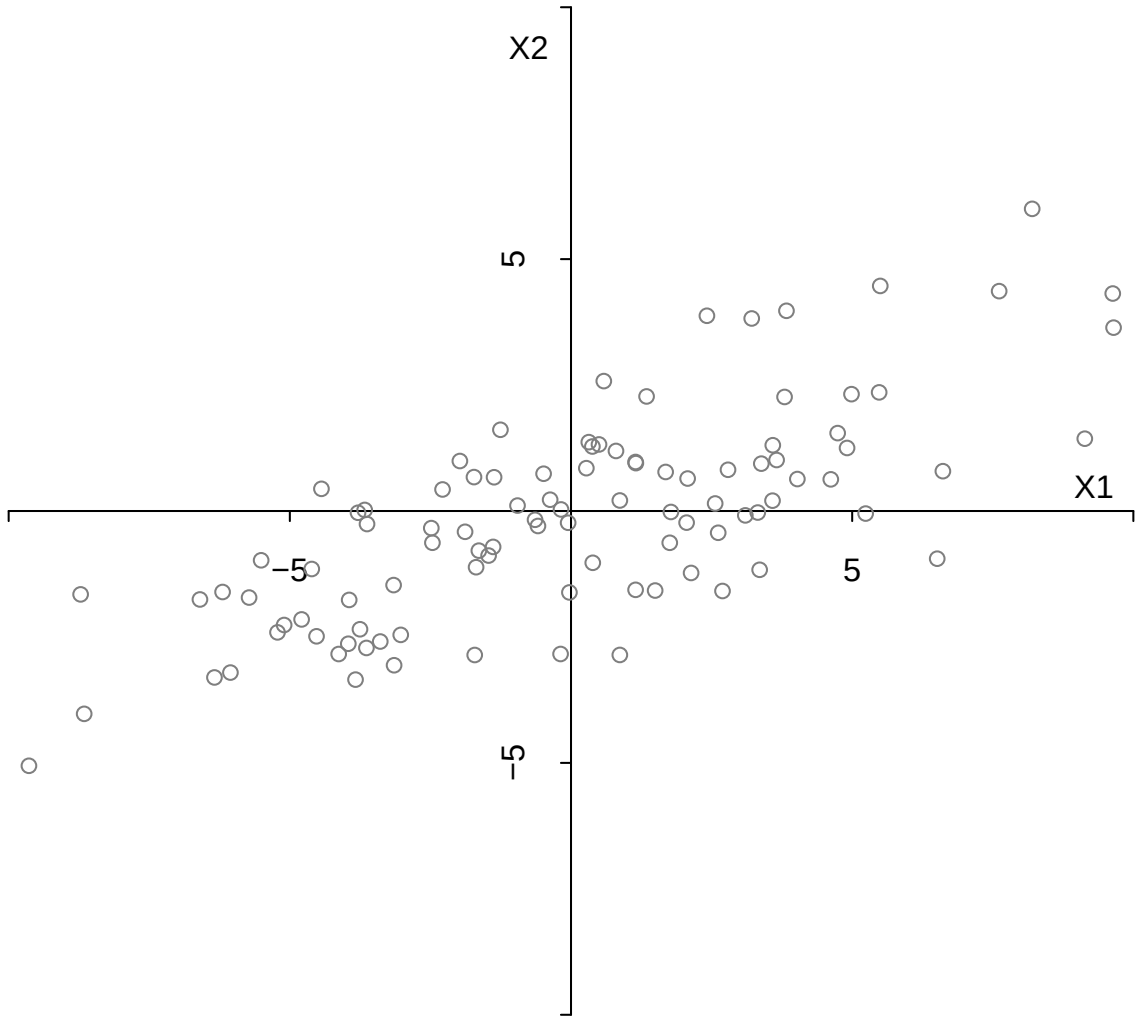
X1 has mean $= 0$, sd $= 2$.
X2 has mean 0, sd $= 5$.
Their correlation is 0.6.

Mahalanobis in practice



	Euclidian distance	Mahalanobis distance
Point 1	5.00	9.77
Point 2	5.00	1.57

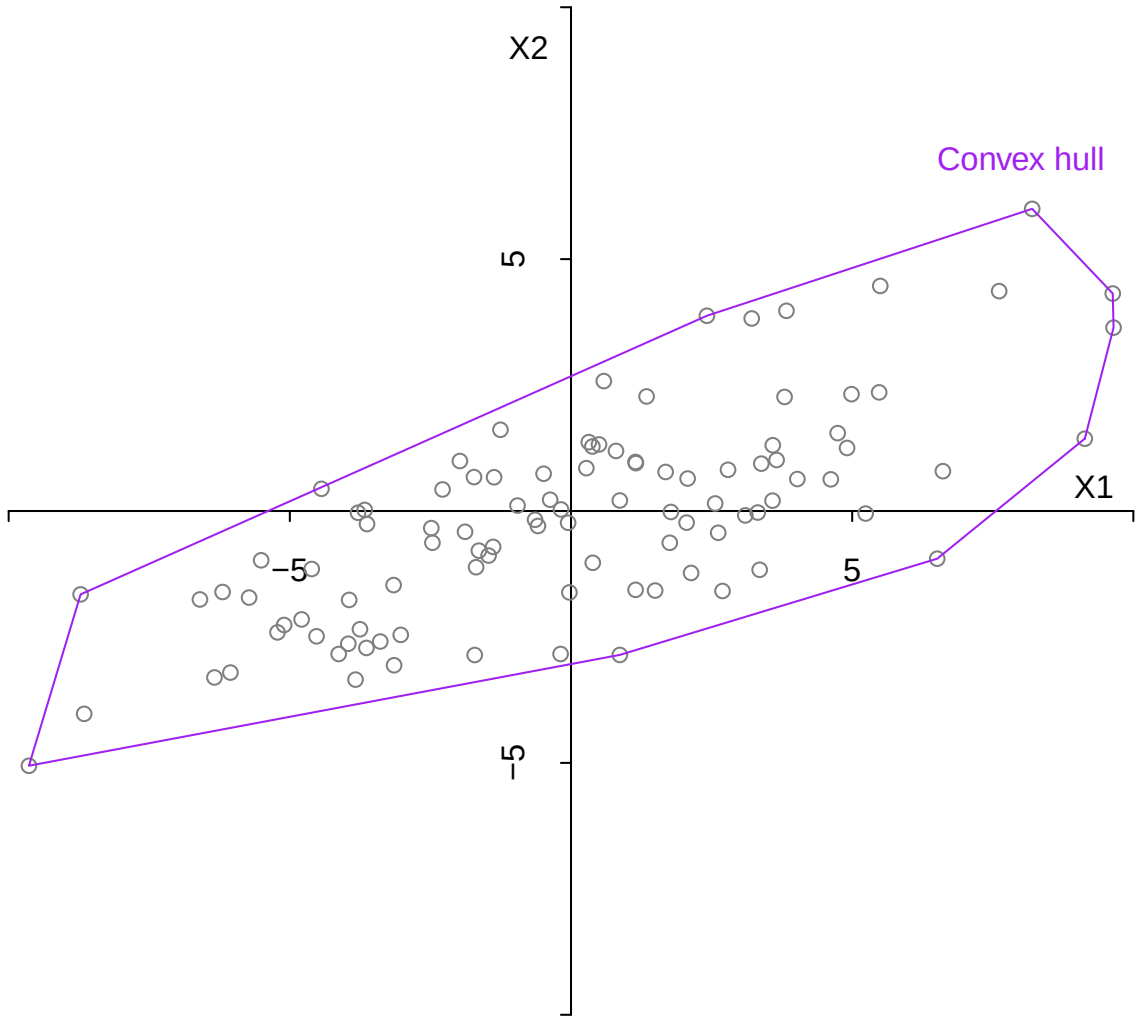
Aside from the aside: Convex hulls



A related issue we've discussed before: extrapolation versus interpolation

What does it mean for a hypothetical set of X's to lie "outside" the above data?

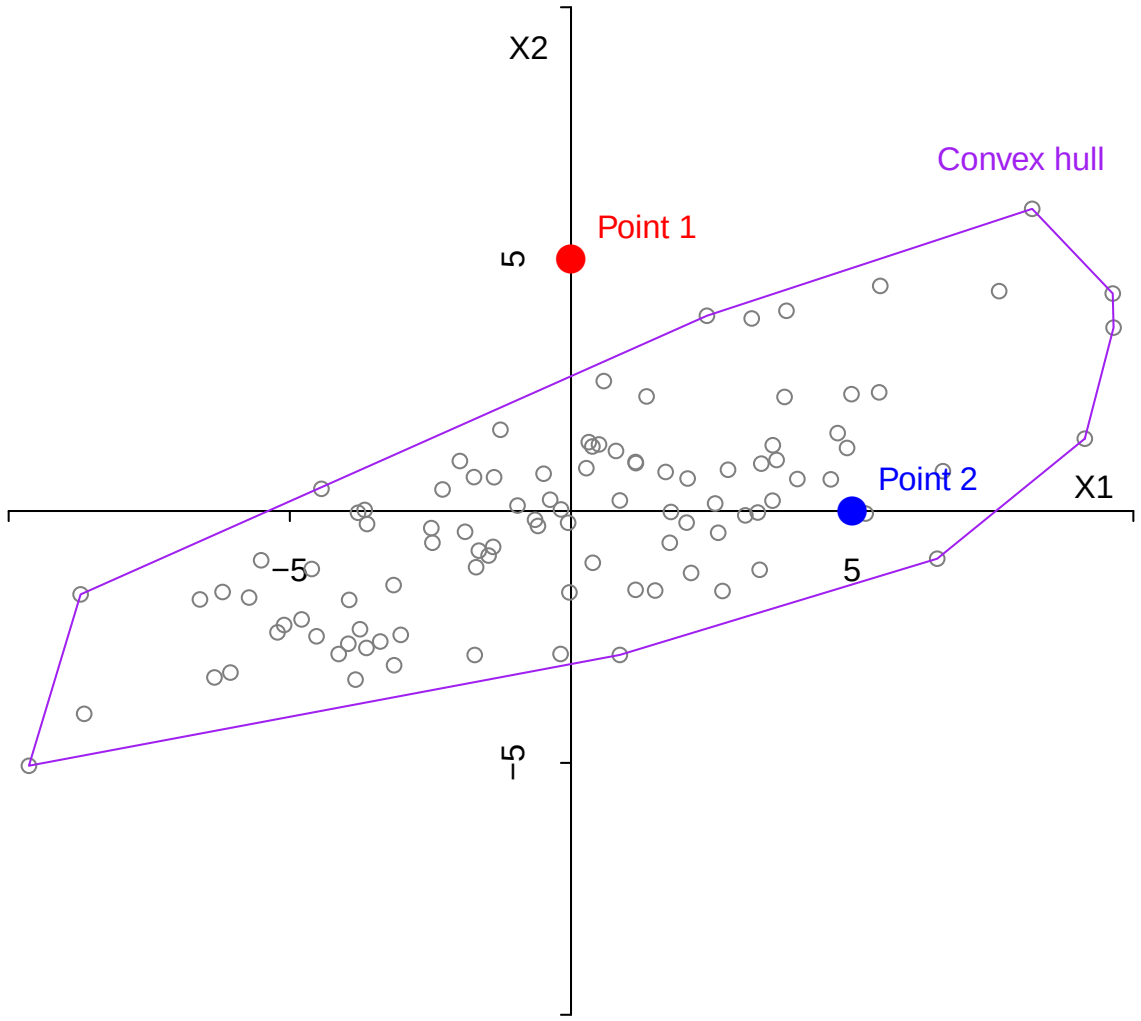
Aside from the aside: Convex hulls



Extrapolation asks a question about a set of X values outside the *convex hull*

A convex hull is an elastic band wrapped around the cloud of X's such that all vertices of the hull are convex

Aside from the aside: Convex hulls



Not coincidentally, the more Mahalanobis distant case is outside the convex hull, while the closer point in is inside

Back to Leverage

The reason we're talking about all this is to understand leverage, which we are measuring with the hat matrix h.

Further notes on the hat matrix:

- The h_i's sum to the number of parameters

Back to Leverage

The reason we're talking about all this is to understand leverage, which we are measuring with the hat matrix h.

Further notes on the hat matrix:

- The h_i's sum to the number of parameters
- Often we standardize the h_i's (divide by their mean) to ease interpretation

Back to Leverage

The reason we're talking about all this is to understand leverage,
which we are measuring with the hat matrix h.

Further notes on the hat matrix:

- The h_i's sum to the number of parameters
- Often we standardize the h_i's (divide by their mean) to ease interpretation
- There are no bright lines between low and high h_i's
  (but above 2 or 3 is often said to be high)

To get the hat matrix in `R`, use `hatvalues(res)`,
where `res` is the result from `lm()`

```
hatvalues(res)/mean(hatvalues(res))
```

gives standardized hat scores: how many times the average leverage each obs has

Discrepancy

To measure discrepancy, we need residuals that reveal the "outlyingness" of each obs

Here we face a problem:

Discrepancy

To measure discrepancy, we need residuals that reveal the "outlyingness" of each obs

Here we face a problem:

Any observation with high influence reduces its own residual!

Discrepancy

To measure discrepancy, we need residuals that reveal the "outlyingness" of each obs

Here we face a problem:

Any observation with high influence reduces its own residual!

This can "mask" outliers. How can we uncover them?

Discrepancy

To measure discrepancy, we need residuals that reveal the "outlyingness" of each obs

Here we face a problem:

Any observation with high influence reduces its own residual!

This can "mask" outliers. How can we uncover them?

First, we need to correct for the effect of leverage.

We can standardize the residuals for leverage using the hat matrix:

$$\hat{\varepsilon}_i^{\text{stand}} = \frac{\hat{\varepsilon}_i}{\sqrt{\sum_i \hat{\varepsilon}_i^2/(n-k-1)}\sqrt{1-h_i}}$$

(Note the residuals are now in standard deviation units; a residual as big as 2 will be seen only 5 percent of the time)

Discrepancy

But this leaves a second problem:

Not just leverage, but also the discrepancy itself reduce the residual

$\rightarrow$ the larger the "true" residual, the greater the downward bias in the observed residual

How can we overcome this?

Discrepancy

But this leaves a second problem:

Not just leverage, but also the discrepancy itself reduce the residual

$\rightarrow$ the larger the "true" residual, the greater the downward bias in the observed residual

How can we overcome this?

Studentized residuals:

$$\hat{\varepsilon}_i^{\text{stud}} = \frac{\hat{\varepsilon}_i}{\sqrt{\sum_{\sim i} \hat{\varepsilon}_{\sim i}^2/(n-k-1)}\sqrt{1-h_i}}$$

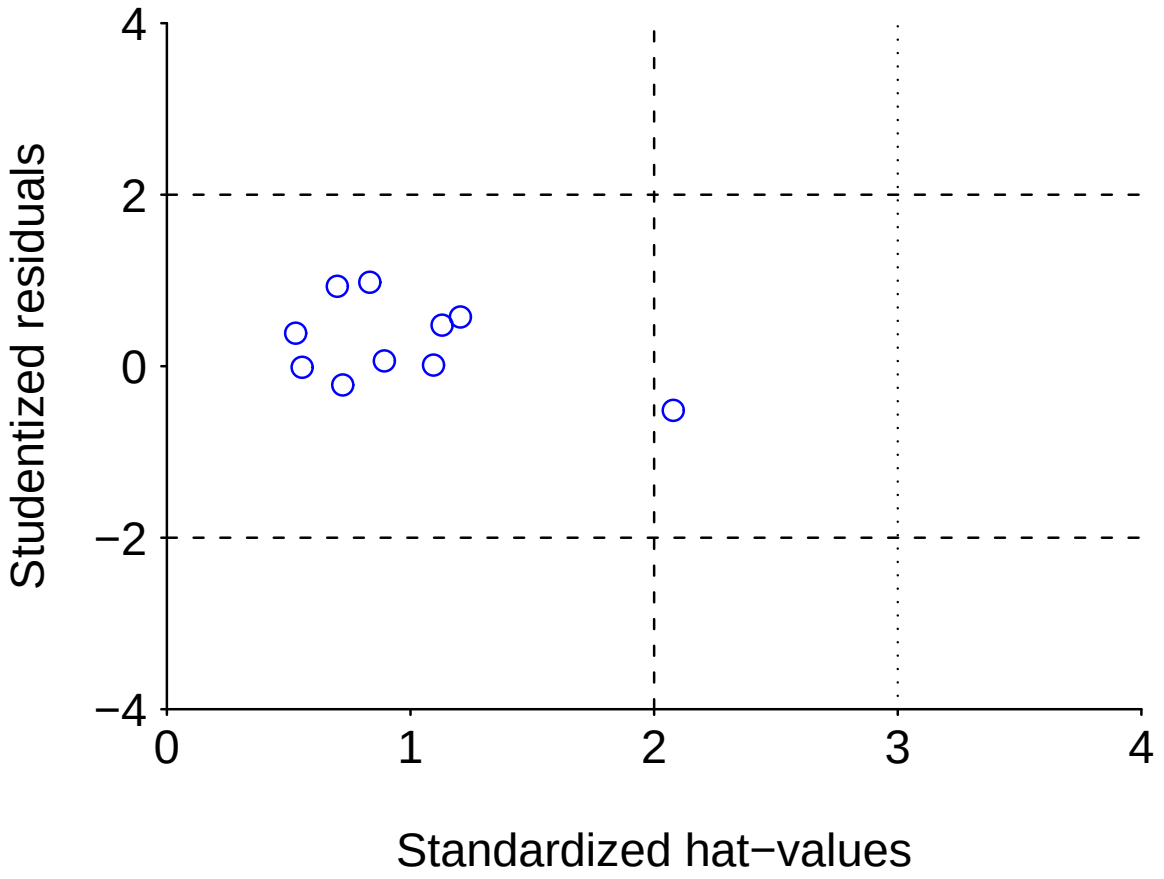
(I.e., calculate the variance of residuals from a regression without observation i)

Equivalent to fitting a (standardized) residual to $\hat{\varepsilon}_i$ from a regression omitting Y_i

Easy to get from `R`

Use `rstudent(res)` where `res` is the result from `lm()`

Influence Plots



We can combine leverage and discrepancy in a simple graphical diagnostic

Helps us decide which obs to investigate (and possibly delete)

Another alternative is respecification (outliers may indicate omitted variables)

Influence Plots

Let's learn how to make these plots, and apply them to the life expectancy example

A common (minor) problem: `lm()` does listwise deletion for us

But then the rows of our original data no longer match the data used in the regression

To plot the data *used in the regression*, we should do the listwise deletion first:

```
library(simcf) # download from chrisadolph.com
data <- read.csv(file="theData.csv", header=TRUE)
model <- lifeexp~I(log(gdpcap85))+I(log(school))+civlib5+wartime
lwdData <- extractdata(formula=model, data, extra=country,
                       na.rm = TRUE)
row.names(lwdData) <- lwdData$country
```

The above sets up a dataframe, `data`, and a set of variables, all of which have only the observations used in the regression

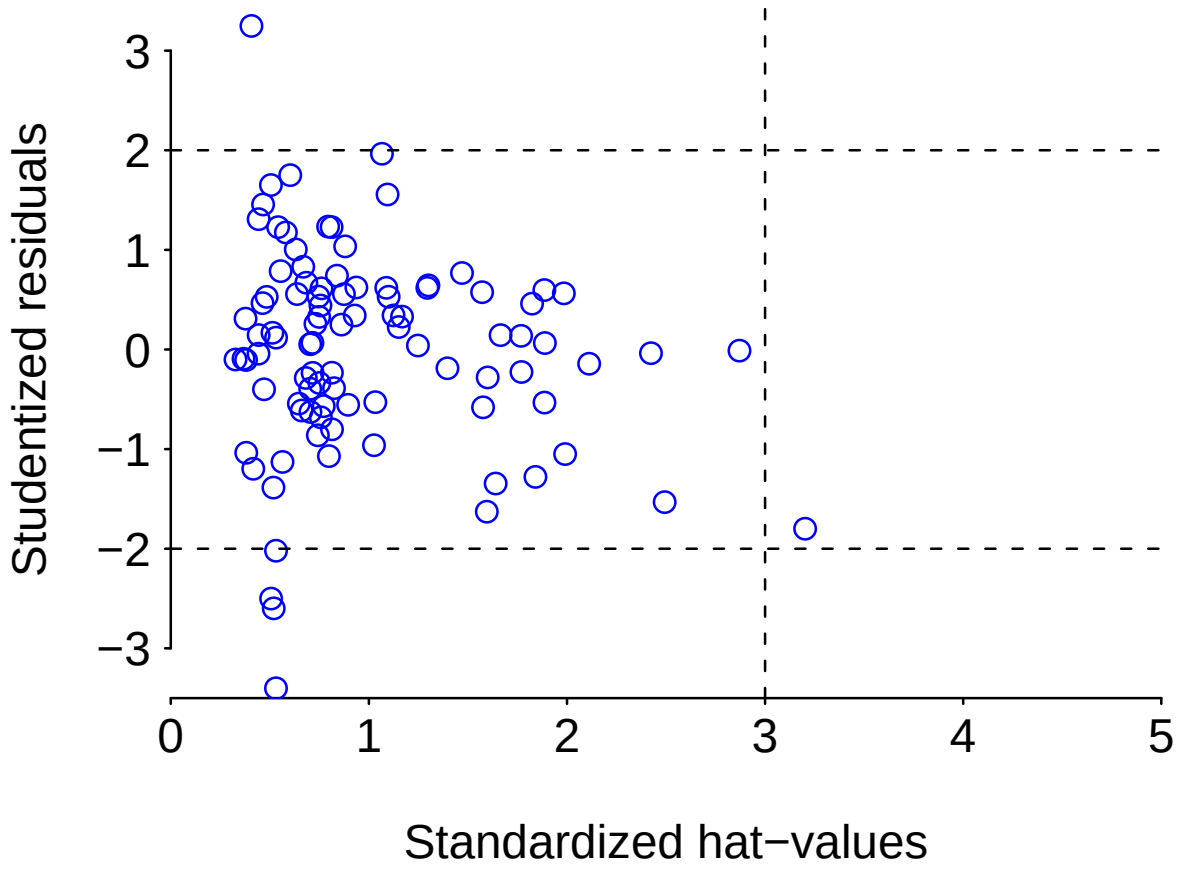
The row names of the dataframe are also our observation names

Influence Plots

To make an influence plot, we use the following code:

```
res <- lm(lifeexp~I(log(gdpcap85))+I(log(school))+civlib5+wartime,
          na.action="na.omit")   # Run the regression

hatscore <- hatvalues(res)/mean(hatvalues(res))
                                 # get the hat scores

rstu <- rstudent(res)            # get the studentized residuals

plot(hatscore,rstu)              # make the plot

identify(hatscore,rstu,row.names(data))
                                 # Use the mouse to label points
```

Applied to the life expectancy data



Would help to know which cases these are

Then we could look for explanations or leave them out

Beware unnecessary dichotomies. . .

Deleting outliers altogether is one solution to the outlier problem

But involves two uncomfortable dichotomies:

Outliers versus non-outliers

Beware unnecessary dichotomies. . .

Deleting outliers altogether is one solution to the outlier problem

But involves two uncomfortable dichotomies:

Outliers versus non-outliers

Delete versus include

Beware unnecessary dichotomies. . .

Deleting outliers altogether is one solution to the outlier problem

But involves two uncomfortable dichotomies:

Outliers versus non-outliers

Delete versus include

What about a continuous version?

Outliers	partial outliers	non-outliers
Delete	Partially include	include

How might we accomplish this?

Robust and resistant regression

Usually, we're unsure which observations are "junk", and which are "good"

Two strategies:

Robust regression: *Reduce*, but do not eliminate, the influence of outliers, at a moderate efficiency cost. Also known as M-estimation.

Resistant regression: *Eliminate* some fraction of the outlying data from the estimation altogether, at a heavy efficiency cost, because much good data will be discarded.

How outliers *should* influence our estimates

Suppose you thought the % of blacks voting Democratic was 0.85, based on many years of data.

Now suppose I analyzed the voting patterns in 2006 and found black %DVS was. . .

How outliers *should* influence our estimates

Suppose you thought the % of blacks voting Democratic was 0.85, based on many years of data.

Now suppose I analyzed the voting patterns in 2006 and found black %DVS was. . .

0.80

How outliers *should* influence our estimates

Suppose you thought the % of blacks voting Democratic was 0.85, based on many years of data.

Now suppose I analyzed the voting patterns in 2006 and found black %DVS was. . .

0.80 "Probably just random fluctuation"

How outliers *should* influence our estimates

Suppose you thought the % of blacks voting Democratic was 0.85, based on many years of data.

Now suppose I analyzed the voting patterns in 2006 and found black %DVS was. . .

0.80 "Probably just random fluctuation"

0.70

How outliers *should* influence our estimates

Suppose you thought the % of blacks voting Democratic was 0.85, based on many years of data.

Now suppose I analyzed the voting patterns in 2006 and found black %DVS was. . .

0.80 "Probably just random fluctuation"

0.70 "Maybe racial politics is changing"

How outliers *should* influence our estimates

Suppose you thought the % of blacks voting Democratic was 0.85, based on many years of data.

Now suppose I analyzed the voting patterns in 2006 and found black %DVS was. . .

0.80 "Probably just random fluctuation"

0.70 "Maybe racial politics is changing"

0.60

How outliers *should* influence our estimates

Suppose you thought the % of blacks voting Democratic was 0.85, based on many years of data.

Now suppose I analyzed the voting patterns in 2006 and found black %DVS was. . .

0.80 "Probably just random fluctuation"

0.70 "Maybe racial politics is changing"

0.60 "Drastic change in political landscape"

How outliers *should* influence our estimates

Suppose you thought the % of blacks voting Democratic was 0.85, based on many years of data.

Now suppose I analyzed the voting patterns in 2006 and found black %DVS was. . .

0.80 "Probably just random fluctuation"

0.70 "Maybe racial politics is changing"

0.60 "Drastic change in political landscape"

0.40

How outliers *should* influence our estimates

Suppose you thought the % of blacks voting Democratic was 0.85, based on many years of data.

Now suppose I analyzed the voting patterns in 2006 and found black %DVS was. . .

0.80 "Probably just random fluctuation"

0.70 "Maybe racial politics is changing"

0.60 "Drastic change in political landscape"

0.40 "You probably made a coding error somewhere"

How outliers *should* influence our estimates

Suppose you thought the % of blacks voting Democratic was 0.85, based on many years of data.

Now suppose I analyzed the voting patterns in 2006 and found black %DVS was. . .

0.80 "Probably just random fluctuation"

0.70 "Maybe racial politics is changing"

0.60 "Drastic change in political landscape"

0.40 "You probably made a coding error somewhere"

Our mental software is treating outliers in *redescending* fashion:

Small deviations have little weight,

Moderate deviations have large weight,

But huge deviations are treated as mistake, and given little weight

What LS *acutally* does with outliers

LS minimizes squared residuals: $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$ chosen to knock down big $\hat{\boldsymbol{\varepsilon}}_{\boldsymbol{i}}$

$|\varepsilon_i| \approx 0$ obs i has little effect on $\hat{\beta}$

$|\varepsilon_i| > 0$ obs i has moderate effect on $\hat{\beta}$

$|\varepsilon_i| >> 0$ obs i has *huge* effect on $\hat{\beta}$

$\rightarrow$ If all of your data are "clean", and from the same DGP, this is exactly what you want

Gauss-Markov theorem still applies: LS is BLUE or even MVU

Big residuals tell us a lot about the data generating process of interest

Help us estimate the right $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$

What LS *acutally* does with outliers

LS minimizes squared residuals: $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$ chosen to knock down big $\hat{\boldsymbol{\varepsilon}}_{\boldsymbol{i}}$

$|\varepsilon_i| \approx 0$ obs i has little effect on $\hat{\beta}$

$|\varepsilon_i| > 0$ obs i has moderate effect on $\hat{\beta}$

$|\varepsilon_i| >> 0$ obs i has *huge* effect on $\hat{\beta}$

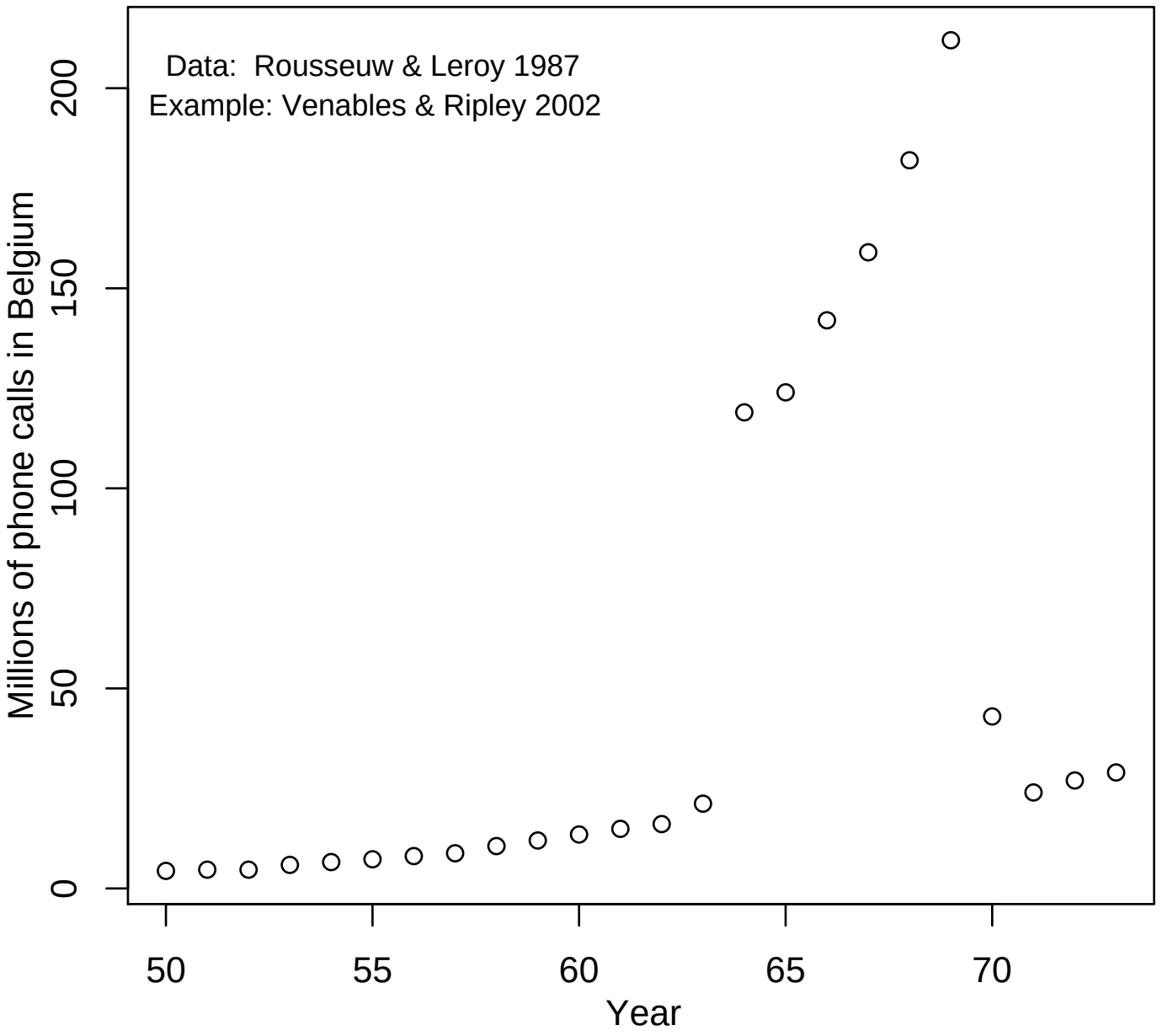
$\rightarrow$ If there is a big outlier or coding error in your data, and you run LS that outlier may have more effect on $\hat{\beta}$ than any other observation!

GM still applies, but LS is estimating parameters for a mixed population

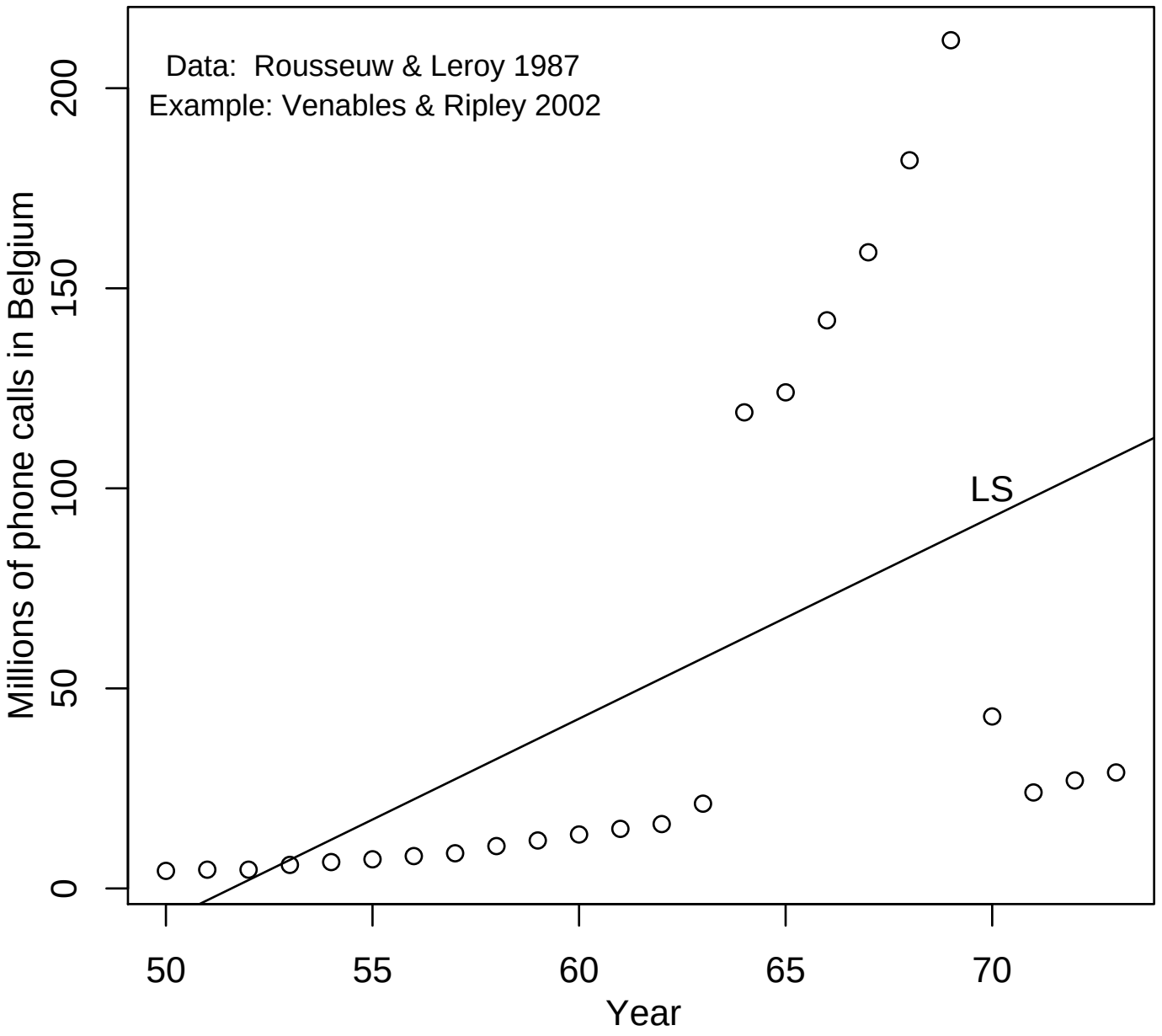
You just want $\hat{\beta}$ for the clean part of the dataset

From that perspective, you could have massive "bias"

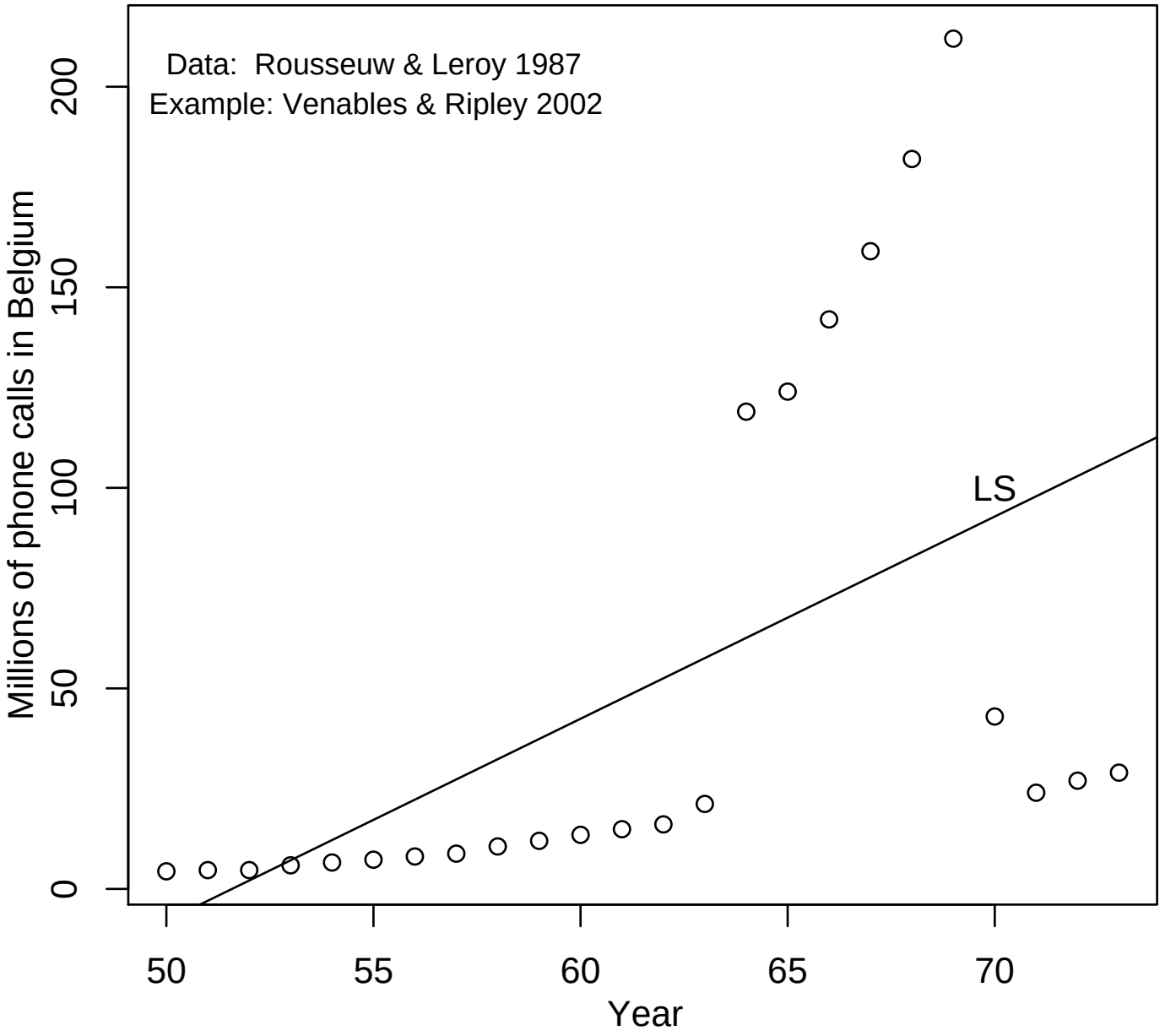
Belgian phone calls example



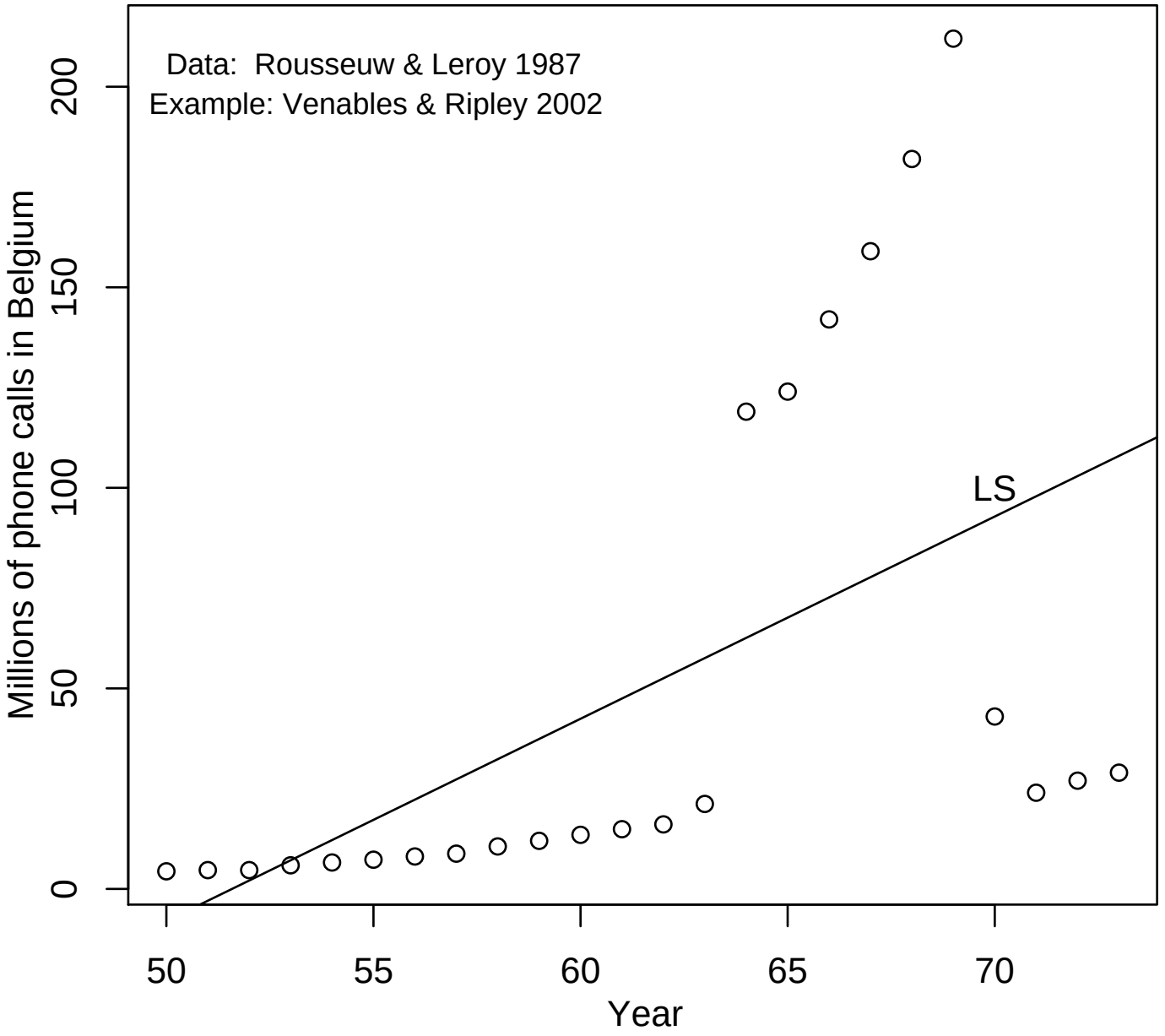
Belgian phone calls example



Outliers: 1964–96, total minutes of calls recorded



Outliers: 1964–96, total minutes of calls recorded



M-estimators

If we think we have outliers, we'd like to give less weight to them

Clearly, LS, which chooses $\hat{\boldsymbol{\beta}}$ to minimize $\sum_{i=1}^{n} \hat{\varepsilon}_i^2$, is the problem

So let's try a generalization called an M-estimator:

Choose $\hat{\boldsymbol{\beta}}$ to minimize a function of the errors, $\sum_{i=1}^{n} \rho(\hat{\varepsilon}_i)$

By clever choice of $\rho(\cdot)$, we reduce influence of outliers, at some cost in efficiency

M-estimators

M-estimators solve the general estimation problem:

$$\min \sum_{i=1}^{n} \rho(\hat{\varepsilon}_i) = \min \sum_{i=1}^{n} \rho(\mathbf{y}_i - \mathbf{x}_i \hat{\boldsymbol{\beta}})$$

M-estimators

M-estimators solve the general estimation problem:

$$\min \sum_{i=1}^{n} \rho(\hat{\varepsilon}_i) = \min \sum_{i=1}^{n} \rho(\mathbf{y}_i - \mathbf{x}_i\hat{\boldsymbol{\beta}})$$

Set the partial derivatives (wrt the elements of $\hat{\boldsymbol{\beta}}$) equal to 0:

$$\sum_{i=1}^{n} \psi(\mathbf{y}_i - \mathbf{x}_i\hat{\boldsymbol{\beta}})\mathbf{x}_i = \mathbf{0}$$

where $\psi(\cdot)$ is the derivative of $\rho(\cdot)$

M-estimators

M-estimators solve the general estimation problem:

$$\min \sum_{i=1}^{n} \rho(\hat{\varepsilon}_i) = \min \sum_{i=1}^{n} \rho(\mathbf{y}_i - \mathbf{x}_i\hat{\boldsymbol{\beta}})$$

Set the partial derivatives (wrt the elements of $\hat{\boldsymbol{\beta}}$) equal to 0:

$$\sum_{i=1}^{n} \psi(\mathbf{y}_i - \mathbf{x}_i\hat{\boldsymbol{\beta}})\mathbf{x}_i = \mathbf{0}$$

where $\psi(\cdot)$ is the derivative of $\rho(\cdot)$

Note that $\mathbf{y}_i - \mathbf{x}_i\hat{\boldsymbol{\beta}}$ are the residuals $\hat{\boldsymbol{\varepsilon}}_i$,

$$\sum_{i=1}^{n} \psi(\hat{\varepsilon}_i)\mathbf{x}_i = \mathbf{0}$$

M-estimators

$$\sum_{i=1}^{n} \psi(\hat{\varepsilon}_i)\mathbf{x}_i = \mathbf{0}$$

Now define the weight function, $\omega(\cdot)$ as

$$\omega(\hat{\varepsilon}_i) = \frac{\psi(\hat{\varepsilon}_i)}{\hat{\varepsilon}_i}$$

M-estimators

$$\sum_{i=1}^{n} \psi(\hat{\varepsilon}_i)\mathbf{x}_i = \mathbf{0}$$

Now define the weight function, $\omega(\cdot)$ as

$$\omega(\hat{\varepsilon}_i) = \frac{\psi(\hat{\varepsilon}_i)}{\hat{\varepsilon}_i}$$

Further, define weights, w_i as

$$w_i = \omega(\hat{\varepsilon}_i)$$

M-estimators

$$\sum_{i=1}^{n} \psi(\hat{\varepsilon}_i)\mathbf{x}_i = \mathbf{0}$$

Now define the weight function, $\omega(\cdot)$ as

$$\omega(\hat{\varepsilon}_i) = \frac{\psi(\hat{\varepsilon}_i)}{\hat{\varepsilon}_i}$$

Further, define weights, w_i as

$$w_i = \omega(\hat{\varepsilon}_i)$$

Using this function, rewrite the estimating equations as:

$$\sum_{i=1}^{n} w_i(\hat{\varepsilon}_i)\mathbf{x}_i = \mathbf{0}$$

M-estimators

$$\sum_{i=1}^{n} \psi(\hat{\varepsilon}_i)\mathbf{x}_i = \mathbf{0}$$

Now define the weight function, $\omega(\cdot)$ as

$$\omega(\hat{\varepsilon}_i) = \frac{\psi(\hat{\varepsilon}_i)}{\hat{\varepsilon}_i}$$

Further, define weights, w_i as

$$w_i = \omega(\hat{\varepsilon}_i)$$

Using this function, rewrite the estimating equations as:

$$\sum_{i=1}^{n} \omega_i(\hat{\varepsilon}_i)\mathbf{x}_i = \mathbf{0}$$

$\hat{\boldsymbol{\beta}}$'s which solve this are the weighted least squares estimates that minimize $\sum_{i=1}^{n} w_i^2 \hat{\varepsilon}_i^2$

M-estimation: Choice of influence function

The M-estimation framework is quite general

We can choose any function, ρ, of the residuals to minimize

That function could just be the sum of squared errors,
so LS is a special case

Let's look at some different choices of ρ, and see what implications they have for the influence of outliers on our estimates

Influence functions: Least squares example

Least squares minimizes the objective function ρ_{LS}:

$$\rho_{\mathrm{LS}} = \hat{\varepsilon}^2$$

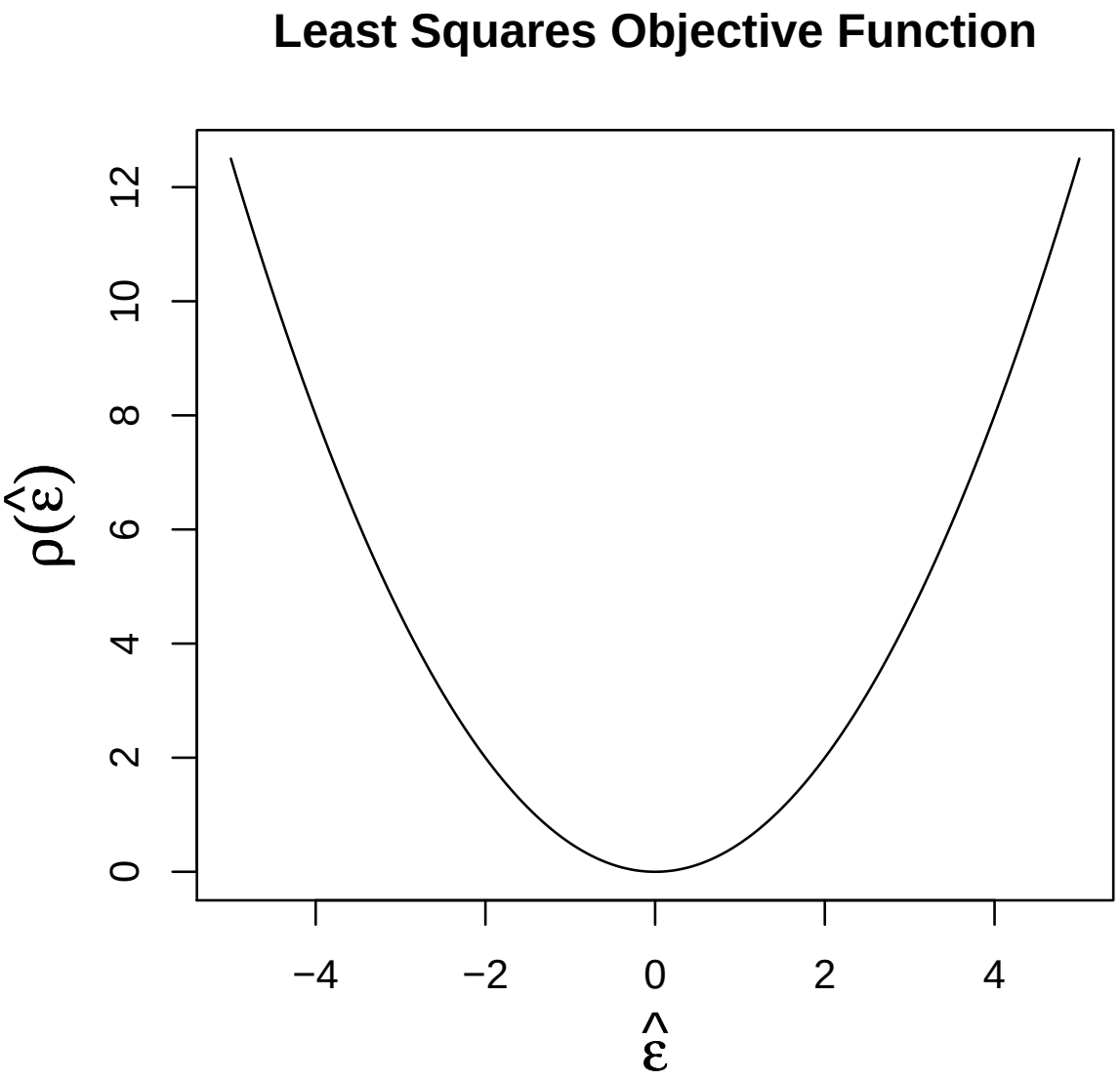
Without loss of generality,
let's say that least squares actually minimizes the $\frac{1}{2}$ times the sum of squared errors, so. . .

Influence functions: Least squares example

Least squares minimizes the objective function ρ_{LS}:

$$\rho_{\mathrm{LS}} = \frac{1}{2}\hat{\boldsymbol{\varepsilon}}^2$$

Influence functions: Least squares example



Influence functions: Least squares example

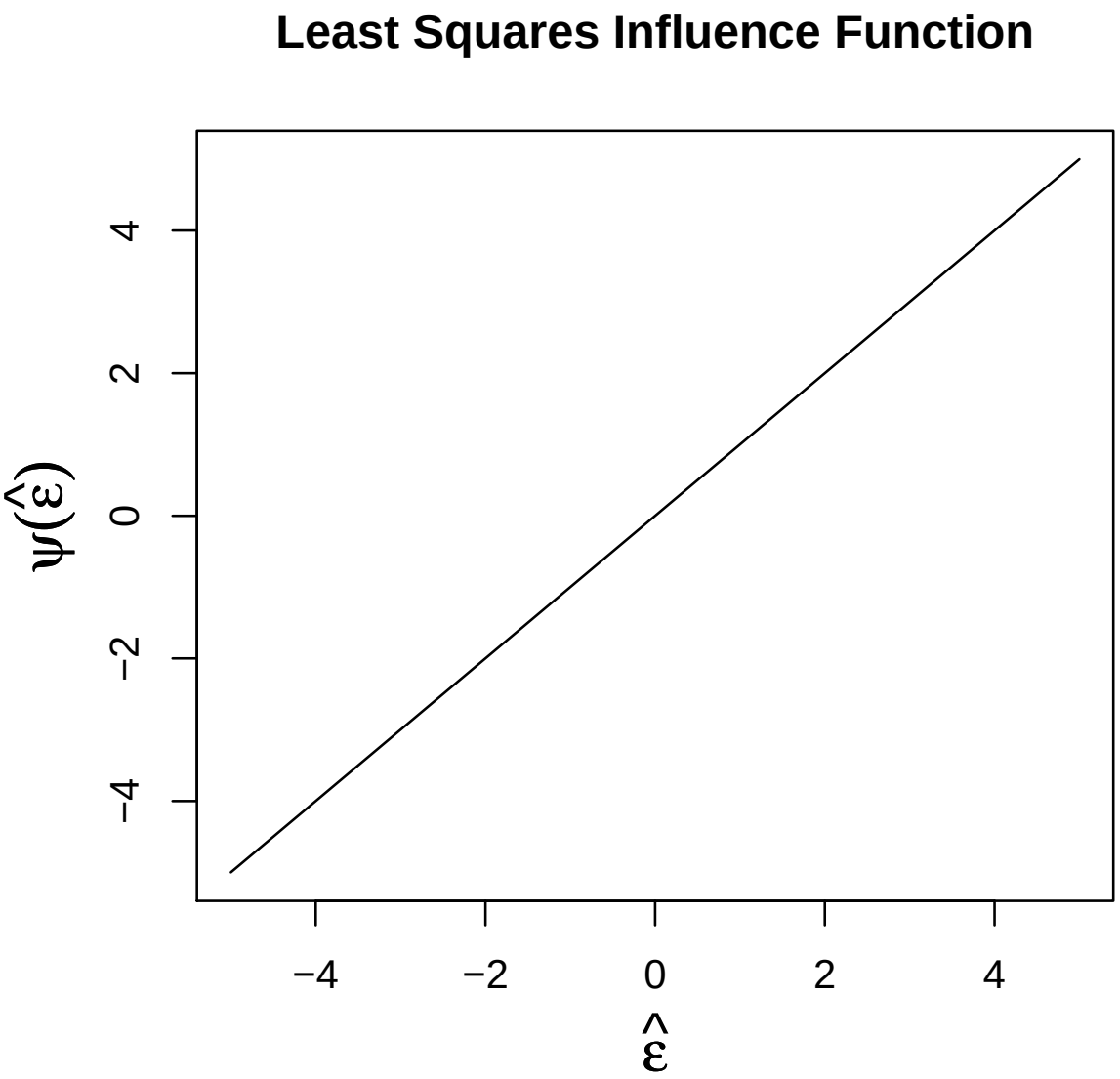
Least squares minimizes the objective function ρ_{LS}:

$$\rho_{\mathrm{LS}} = \frac{1}{2}\hat{\boldsymbol{\varepsilon}}^2$$

We call the derivative of $\rho(\hat{\varepsilon})$ the influence function, $\psi(\hat{\varepsilon})$:

$$\psi_{\mathrm{LS}} = \frac{\partial \rho_{\mathrm{LS}}}{\partial \hat{\varepsilon}} = \boldsymbol{\varepsilon}$$

Influence functions: Least squares example



In LS, influence of observation grows with size of the residual

Influence functions: Least squares example

Least squares minimizes the objective function ρ_{LS}:

$$\rho_{\mathrm{LS}} = \frac{1}{2}\hat{\boldsymbol{\varepsilon}}^2$$

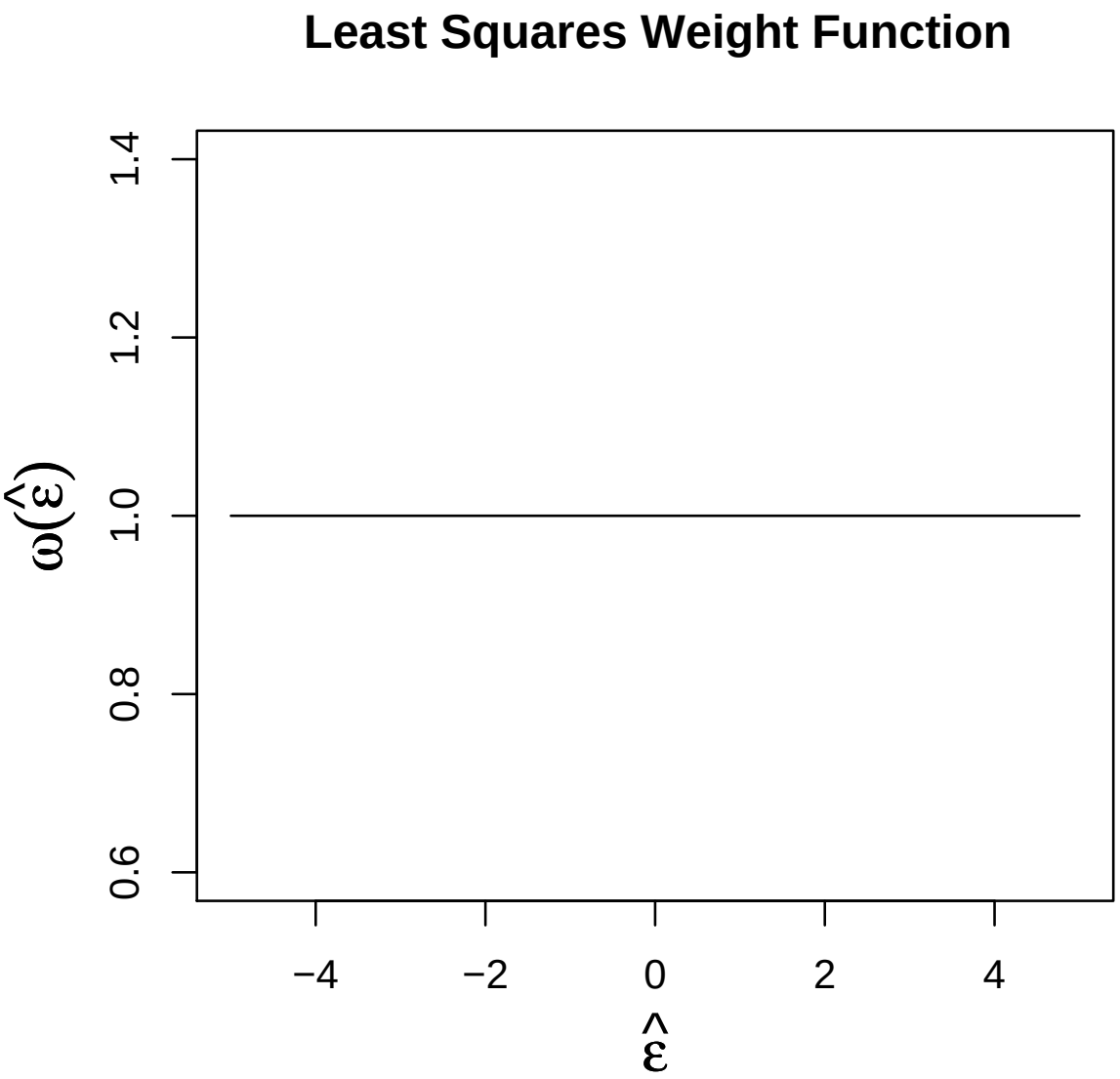
We call the derivative of $\rho(\hat{\varepsilon})$ the influence function, $\psi(\hat{\boldsymbol{\varepsilon}})$:

$$\psi_{\mathrm{LS}} = \frac{\partial \rho_{\mathrm{LS}}}{\partial \hat{\varepsilon}} = \boldsymbol{\varepsilon}$$

Finally, the weight for each observation implied by $\psi(\hat{\boldsymbol{\varepsilon}})$ is:

$$\omega_{\mathrm{LS}} = \frac{\psi_{\mathrm{LS}}(\hat{\varepsilon})}{\hat{\varepsilon}} = 1$$

Influence functions: Least squares example



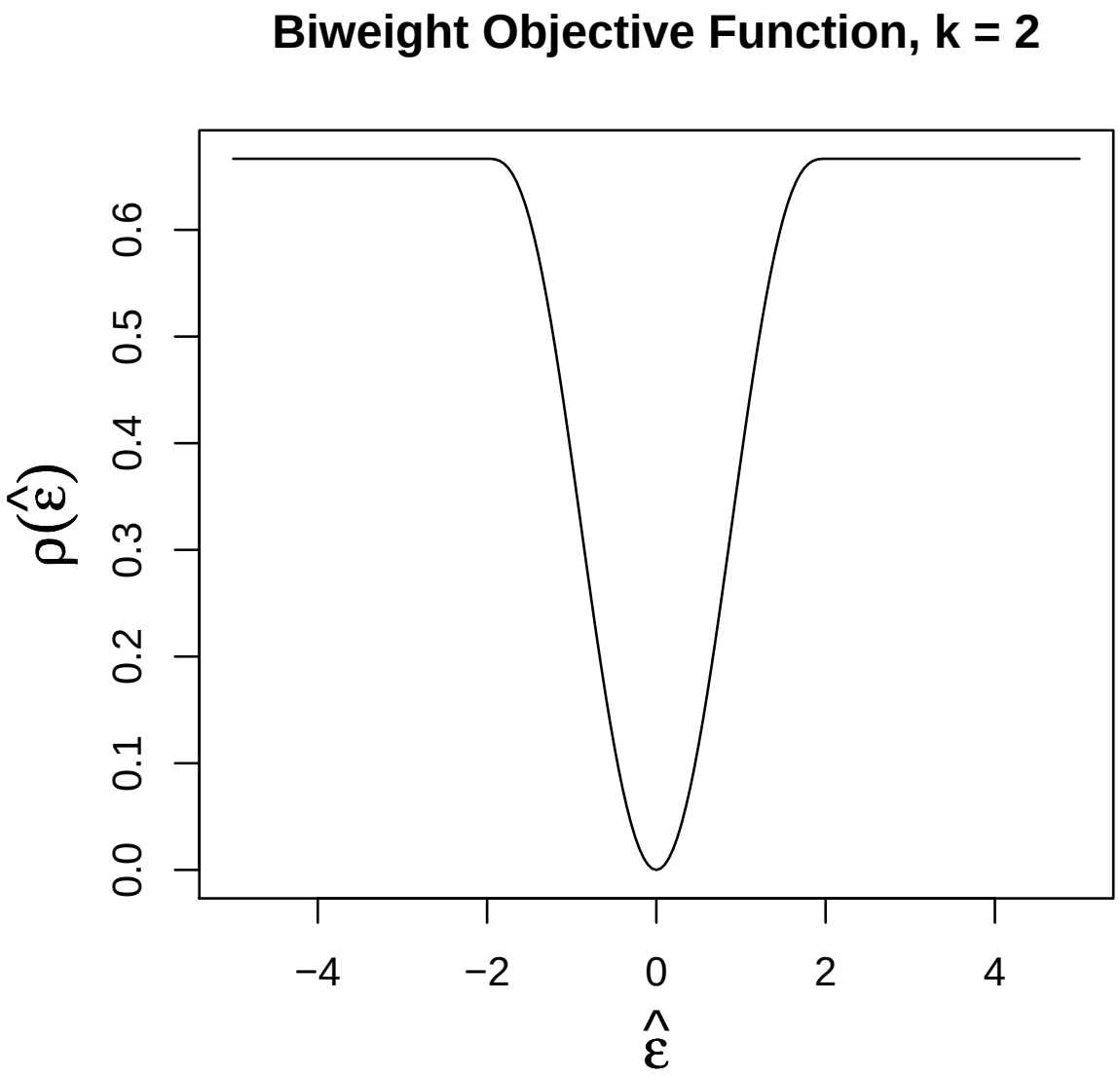
LS gives equal weight to each observation, even if residual is large

Influence functions: Biweight example

Suppose we choose to minimize this objective function of our residuals:

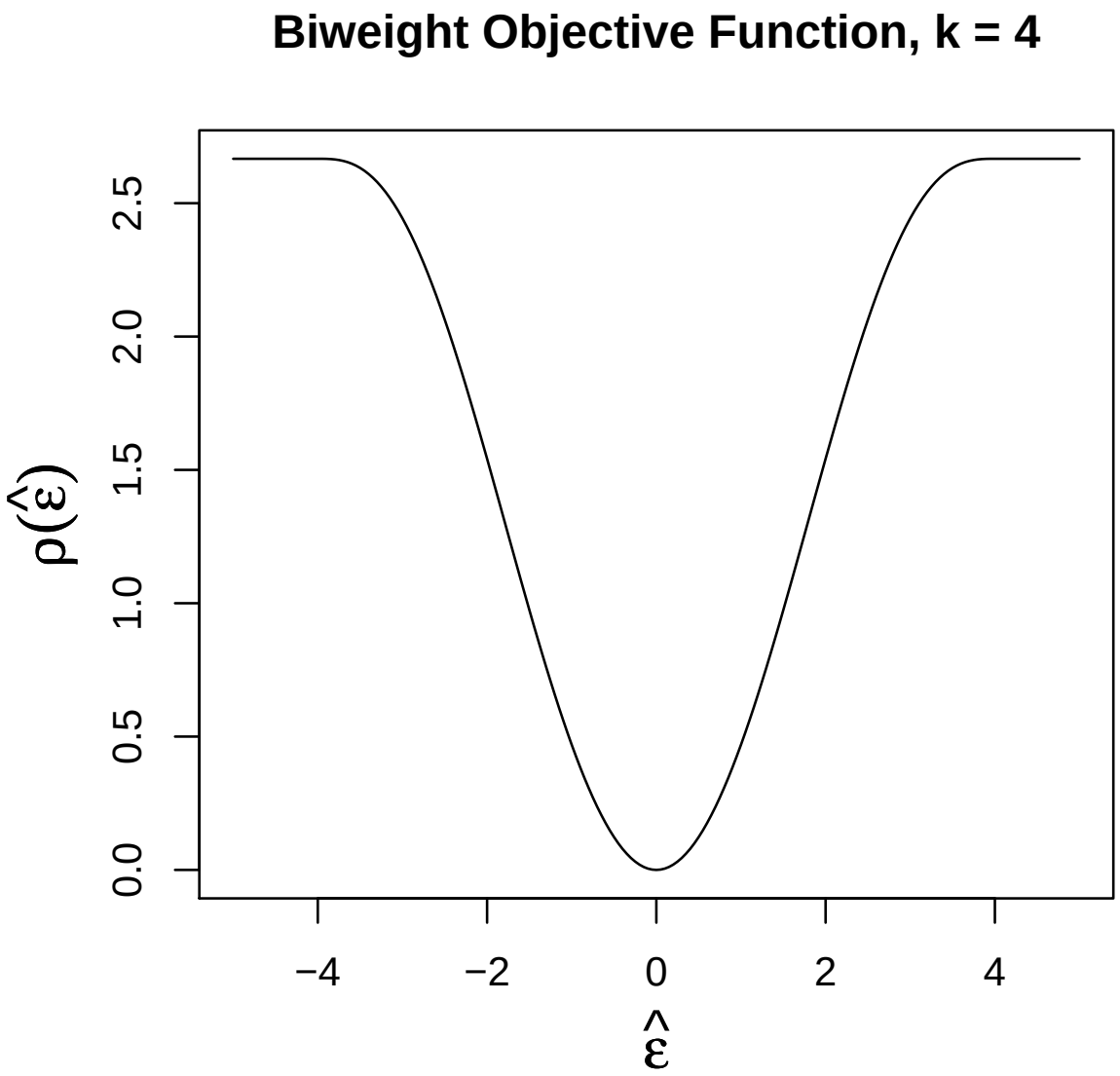
$$\rho_{\text{Biweight}}(\hat{\varepsilon}) = \begin{cases} \frac{k^2}{6}\left\{1 - \left[1 - \left(\frac{\hat{\varepsilon}}{k}\right)^2\right]^3\right\} & \text{for } |\hat{\varepsilon}| \leq k \\ \frac{k^2}{6} & \text{for } |\hat{\varepsilon}| > k \end{cases}$$

Influence functions: Biweight example



A biweight objective function with tuning constant $k = 2$

Influence functions: Biweight example



A biweight objective function with tuning constant $k = 4$

Influence functions: Biweight example

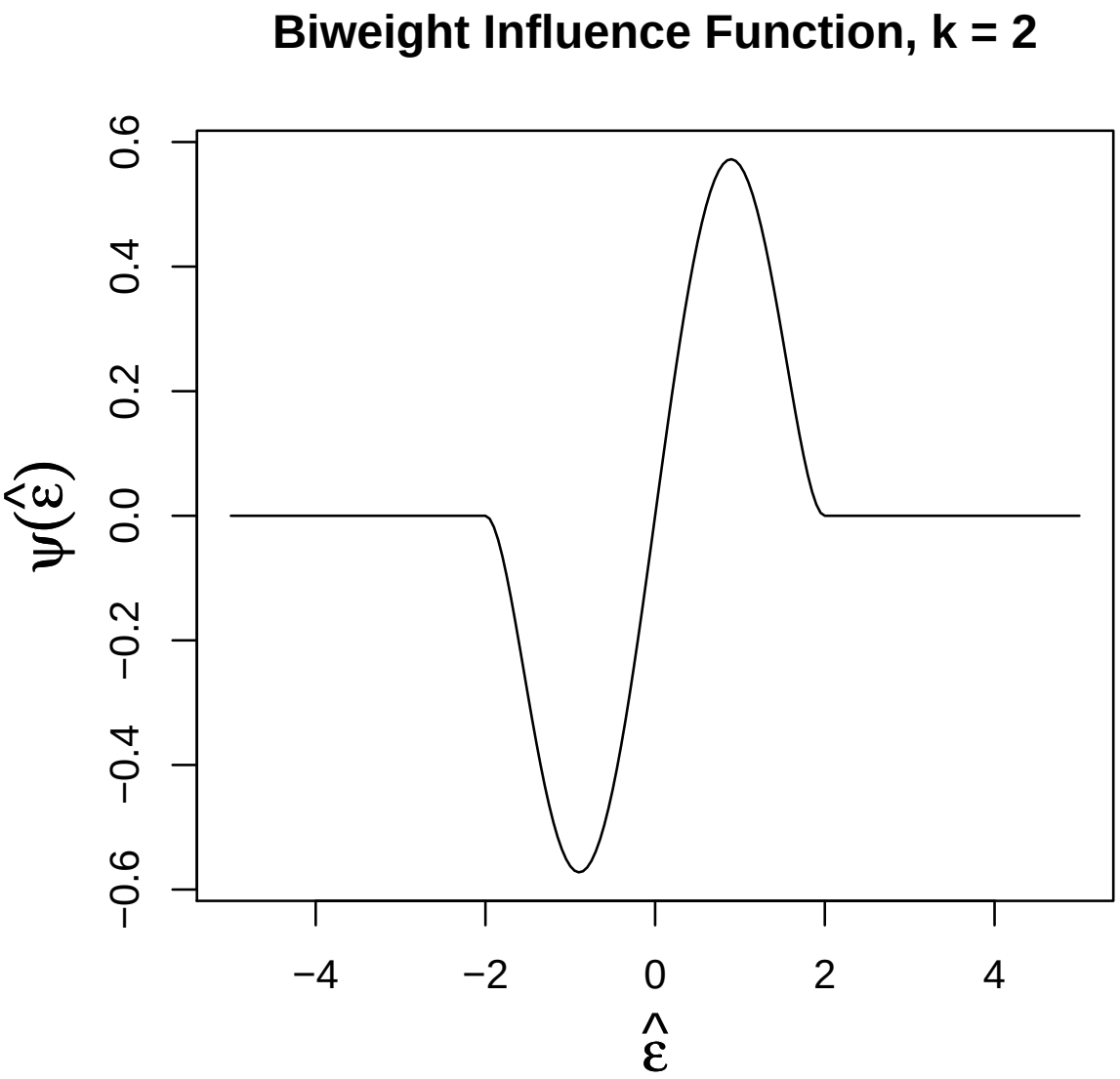
Suppose we choose to minimize this objective function of our residuals:

$$\rho_{\text{Biweight}}(\hat{\varepsilon}) = \begin{cases} \frac{k^2}{6}\left\{1 - \left[1 - \left(\frac{\hat{\varepsilon}}{k}\right)^2\right]^3\right\} & \text{for } |\hat{\varepsilon}| \leq k \\ \frac{k^2}{6} & \text{for } |\hat{\varepsilon}| > k \end{cases}$$

This implies a new, more complex influence function:

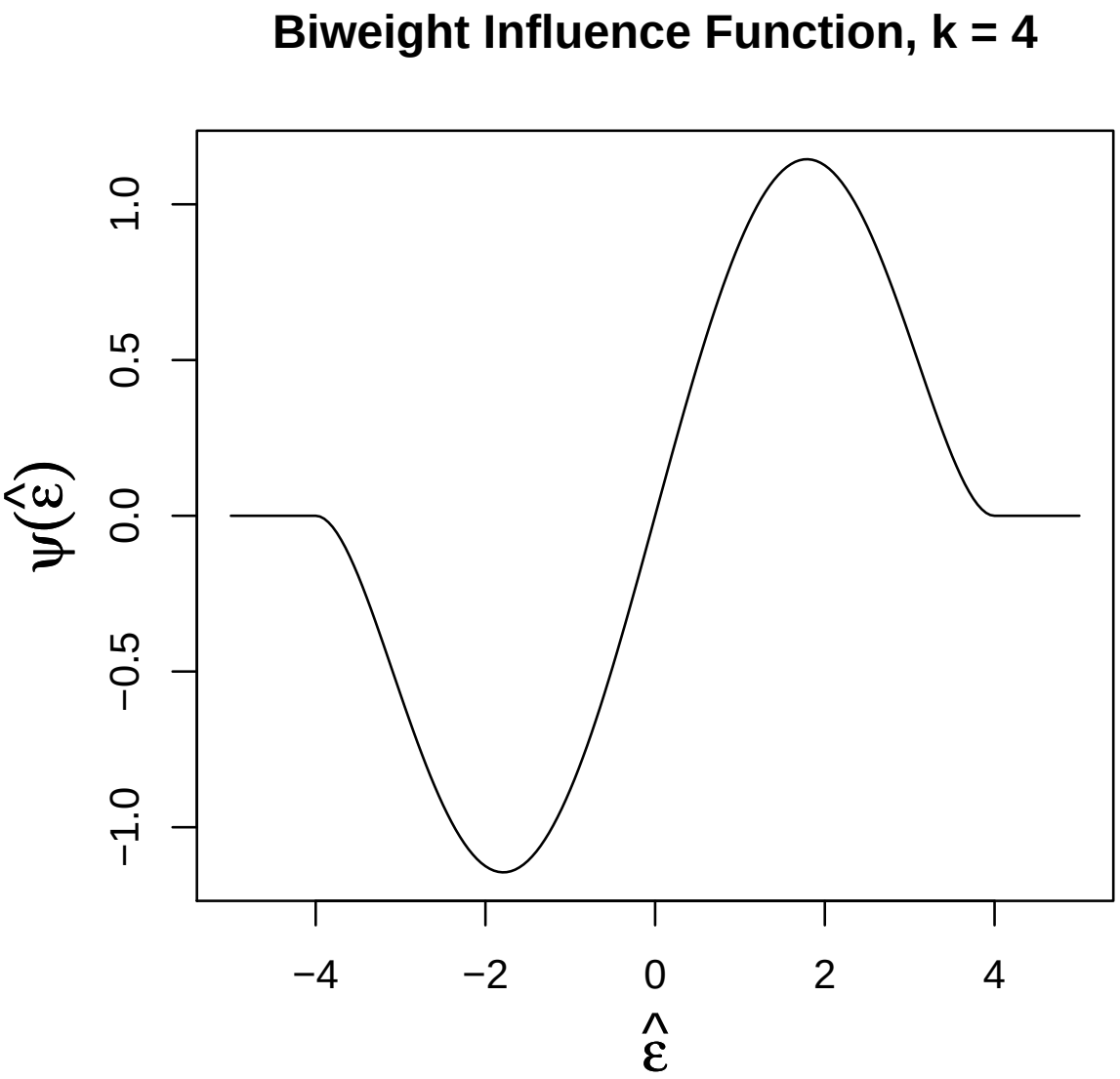
$$\psi_{\text{Biweight}}(\hat{\varepsilon}) = \frac{\partial \rho_{\text{Biweight}}}{\partial \hat{\varepsilon}} = \begin{cases} \hat{\varepsilon}\left[1 - \left(\frac{\hat{\varepsilon}}{k}\right)^2\right]^2 & \text{for } |\hat{\varepsilon}| \leq k \\ 0 & \text{for } |\hat{\varepsilon}| > k \end{cases}$$

Influence functions: Biweight example



Note that the biweight influence function is *redescending*

Influence functions: Biweight example



Extreme outliers get no weight—but if lots of junk data, good data will be outliers!

Influence functions: Biweight example

Suppose we choose to minimize this objective function of our residuals:

$$\rho_{\text{Biweight}}(\hat{\varepsilon}) = \begin{cases} \frac{k^2}{6}\left\{1 - \left[1 - \left(\frac{\hat{\varepsilon}}{k}\right)^2\right]^3\right\} & \text{for } |\hat{\varepsilon}| \leq k \\ \frac{k^2}{6} & \text{for } |\hat{\varepsilon}| > k \end{cases}$$

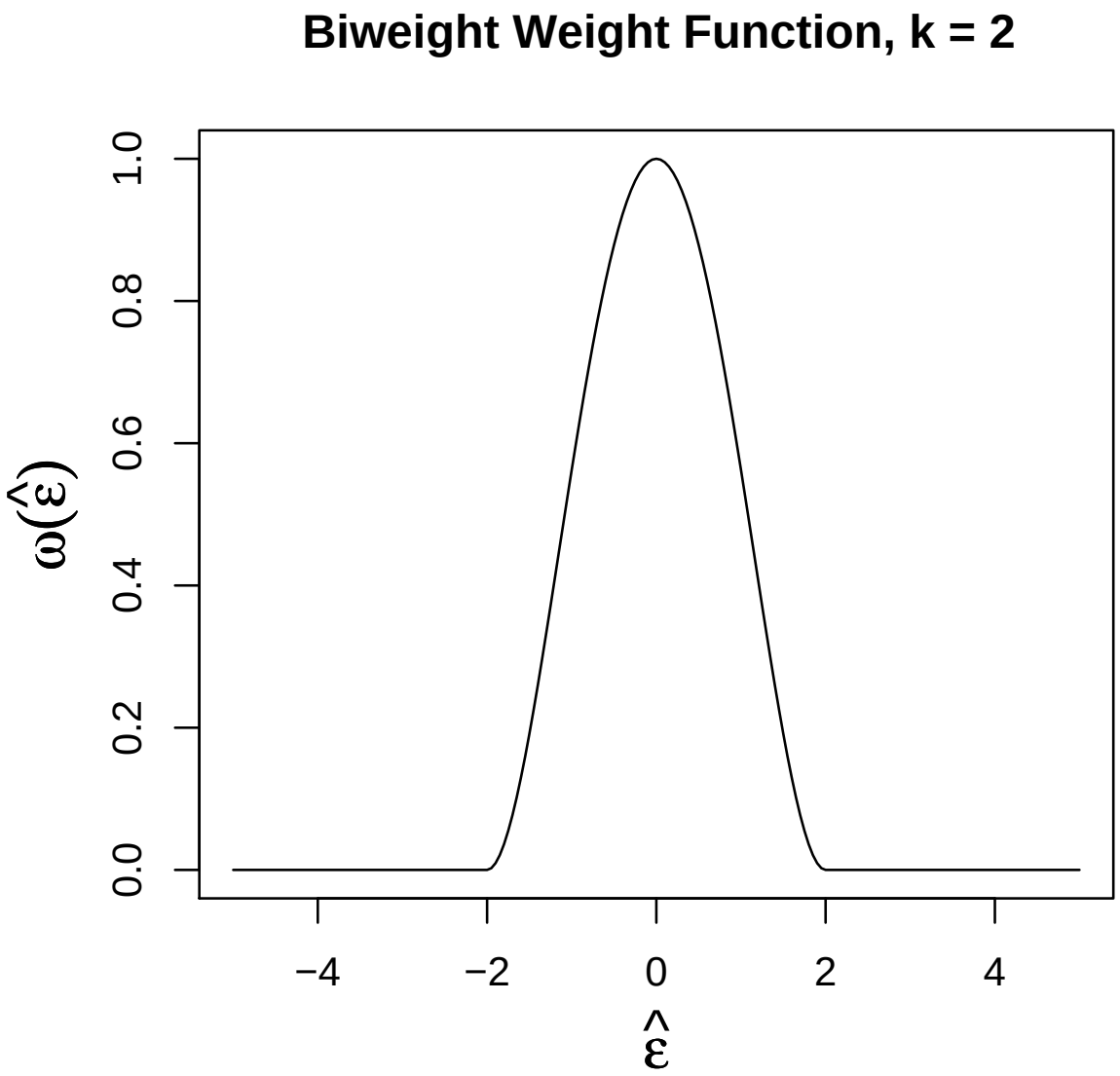
This implies a new, more complex influence function:

$$\psi_{\text{Biweight}}(\hat{\varepsilon}) = \frac{\partial \rho_{\text{Biweight}}}{\partial \hat{\varepsilon}} = \begin{cases} \hat{\varepsilon}\left[1 - \left(\frac{\hat{\varepsilon}}{k}\right)^2\right]^2 & \text{for } |\hat{\varepsilon}| \leq k \\ 0 & \text{for } |\hat{\varepsilon}| > k \end{cases}$$

Which implies a (no longer constant) weight function:

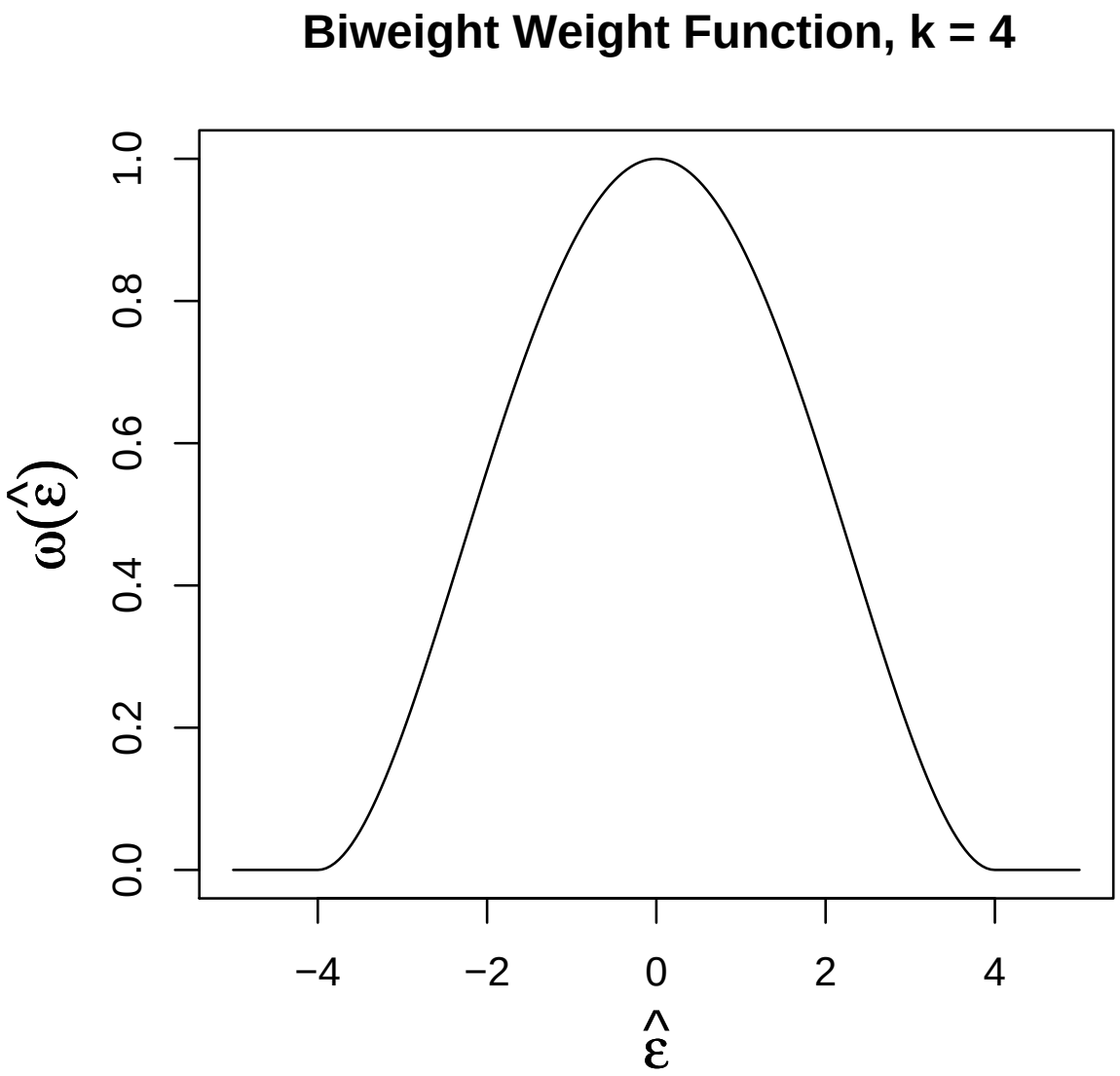
$$\omega_{\text{Biweight}}(\hat{\varepsilon}) = \frac{\psi_{\text{Biweight}}(\hat{\varepsilon})}{\hat{\varepsilon}} = \begin{cases} \left[1 - \left(\frac{\hat{\varepsilon}}{k}\right)^2\right]^2 & \text{for } |\hat{\varepsilon}| \leq k \\ 0 & \text{for } |\hat{\varepsilon}| > k \end{cases}$$

Influence functions: Biweight example



Note the extreme outliers get weight of zero

Influence functions: Biweight example



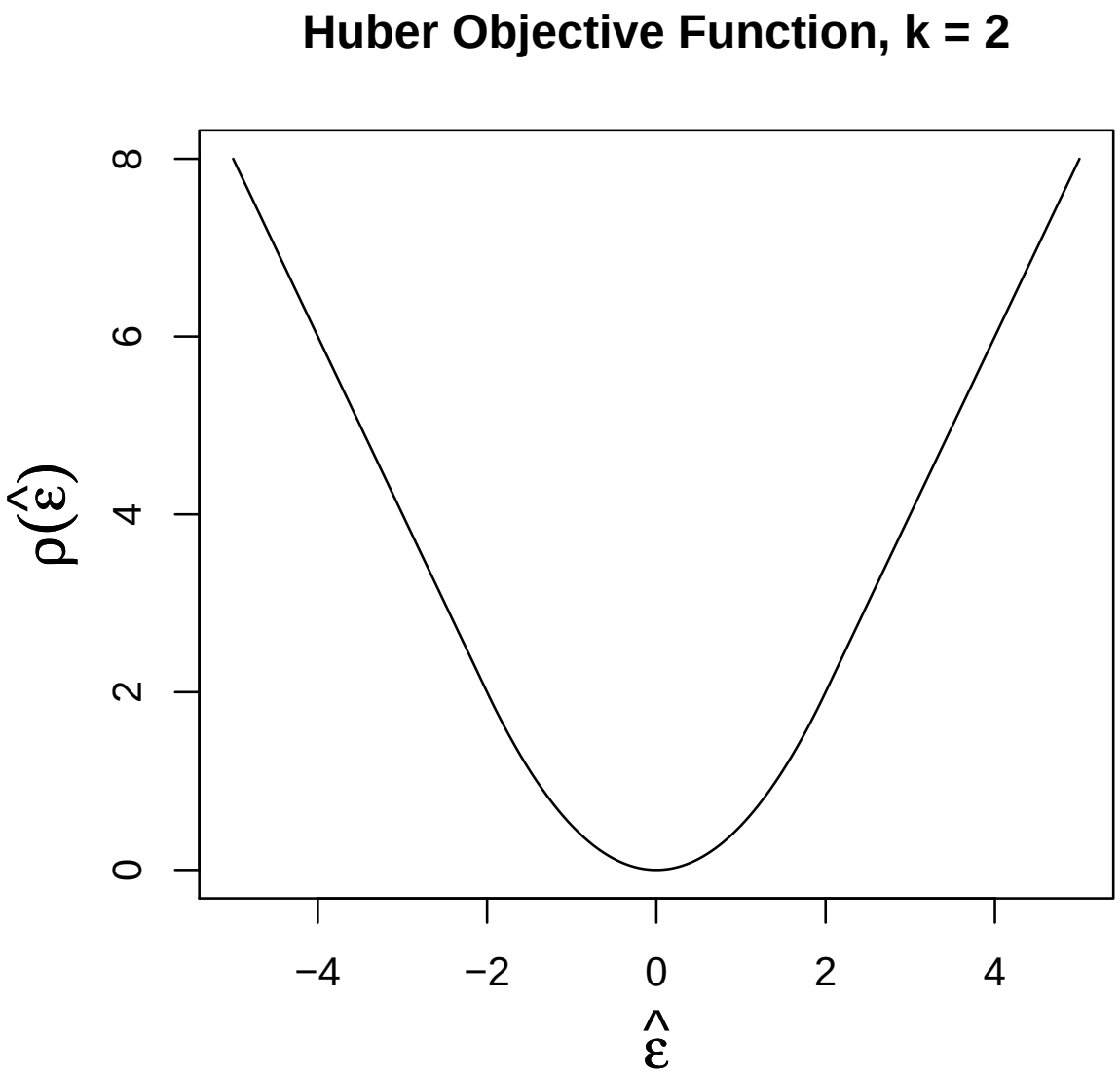
But these could be exactly the data we want to keep if too much contamination!

Influence functions: Huber example

Now suppose we choose to minimize this function of our residuals:

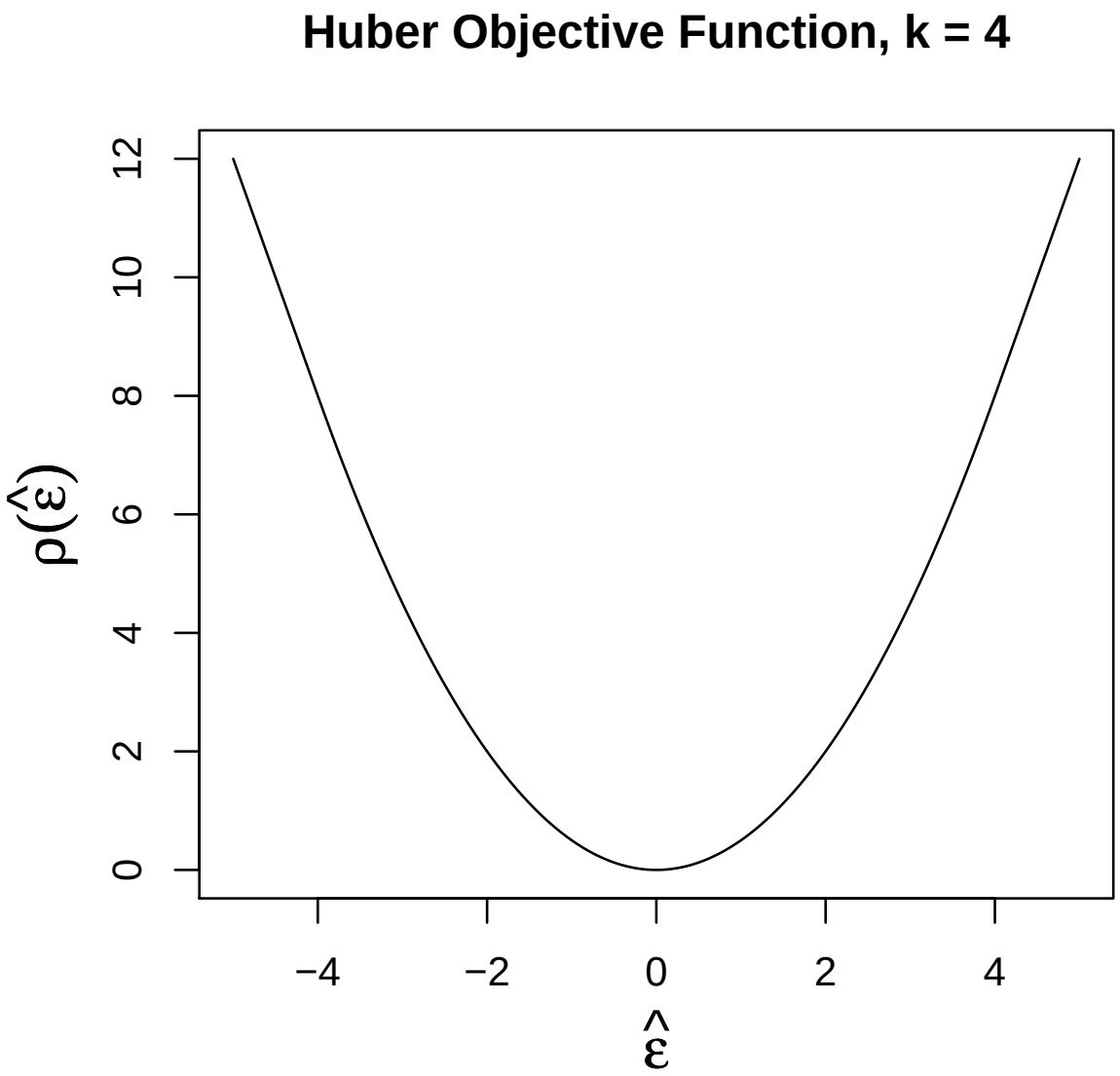
$$\rho_{\text{Huber}}(\hat{\varepsilon}) = \begin{cases} \frac{1}{2}\hat{\varepsilon}^2 & \text{for } |\hat{\varepsilon}| \leq k \\ k|\hat{\varepsilon}| - \frac{1}{2}k^2 & \text{for } |\hat{\varepsilon}| > k \end{cases}$$

Influence functions: Huber example



A Huber objective function with tuning constant $k = 2$

Influence functions: Huber example



A Huber objective function with tuning constant $k = 4$

Influence functions: Huber example

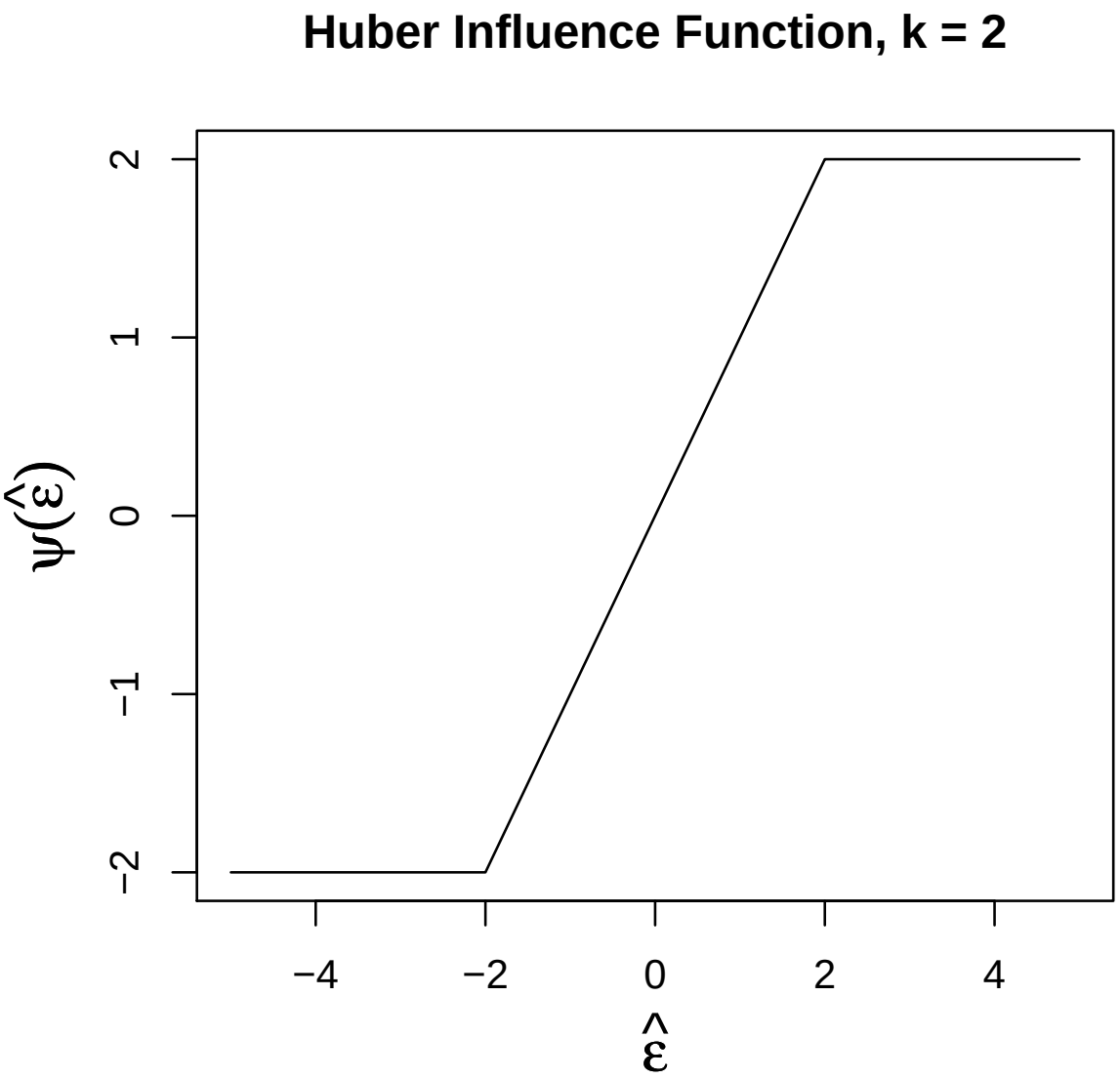
Now suppose we choose to minimize this function of our residuals:

$$\rho_{\text{Huber}}(\hat{\varepsilon}) = \begin{cases} \frac{1}{2}\hat{\varepsilon}^2 & \text{for } |\hat{\varepsilon}| \leq k \\ k|\hat{\varepsilon}| - \frac{1}{2}k^2 & \text{for } |\hat{\varepsilon}| > k \end{cases}$$

This implies a new influence function:

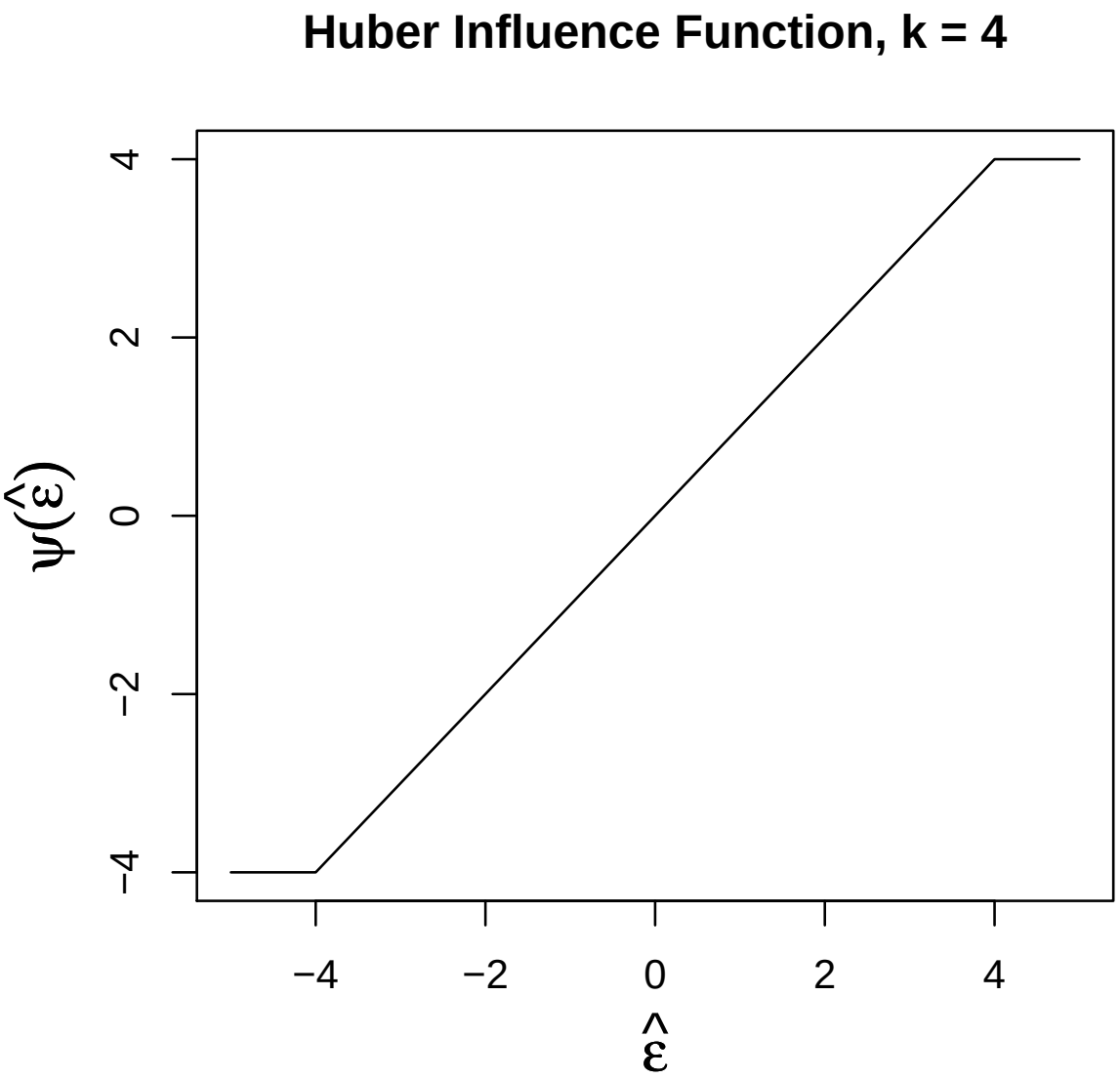
$$\psi_{\text{Huber}}(\hat{\varepsilon}) = \frac{\partial \rho_{\text{Huber}}}{\partial \hat{\varepsilon}} = \begin{cases} k & \text{for } \hat{\varepsilon} > k \\ \hat{\varepsilon} & \text{for } |\hat{\varepsilon}| \leq k \\ -k & \text{for } \hat{\varepsilon} < -k \end{cases}$$

Influence functions: Huber example



Huber influence function is *not* redescending. Easier to estimate

Influence functions: Huber example



Even extreme outliers allowed to influence. Huber less "robust" than Biweight

Influence functions: Huber example

Now suppose we choose to minimize this function of our residuals:

$$\rho_{\text{Huber}}(\hat{\varepsilon}) = \begin{cases} \frac{1}{2}\hat{\varepsilon}^2 & \text{for } |\hat{\varepsilon}| \leq k \\ k|\hat{\varepsilon}| - \frac{1}{2}k^2 & \text{for } |\hat{\varepsilon}| > k \end{cases}$$

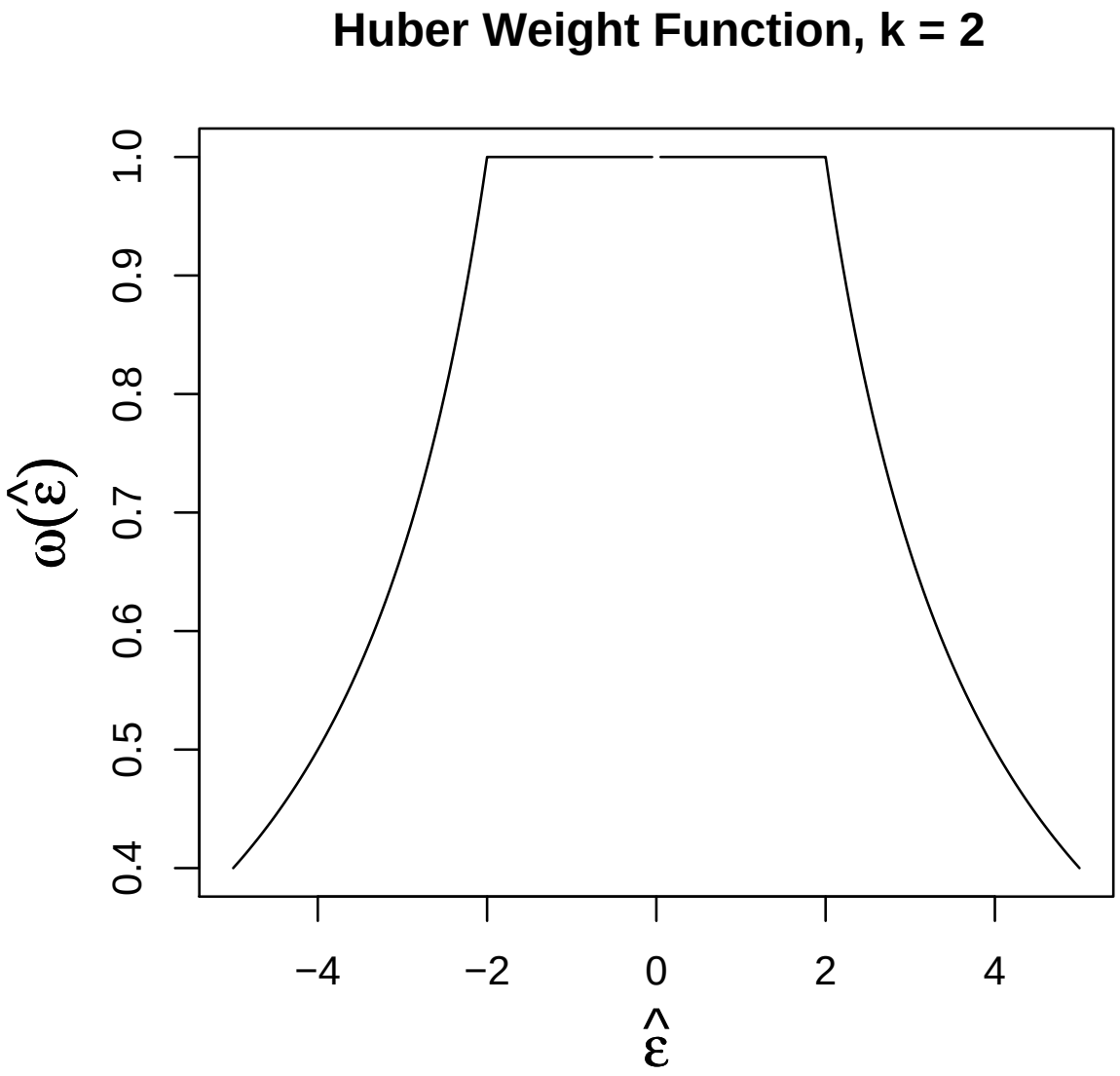
This implies a new influence function:

$$\psi_{\text{Huber}}(\hat{\varepsilon}) = \frac{\partial \rho_{\text{Huber}}}{\partial \hat{\varepsilon}} = \begin{cases} k & \text{for } \hat{\varepsilon} > k \\ \hat{\varepsilon} & \text{for } |\hat{\varepsilon}| \leq k \\ -k & \text{for } \hat{\varepsilon} < -k \end{cases}$$

. . . and weight function:

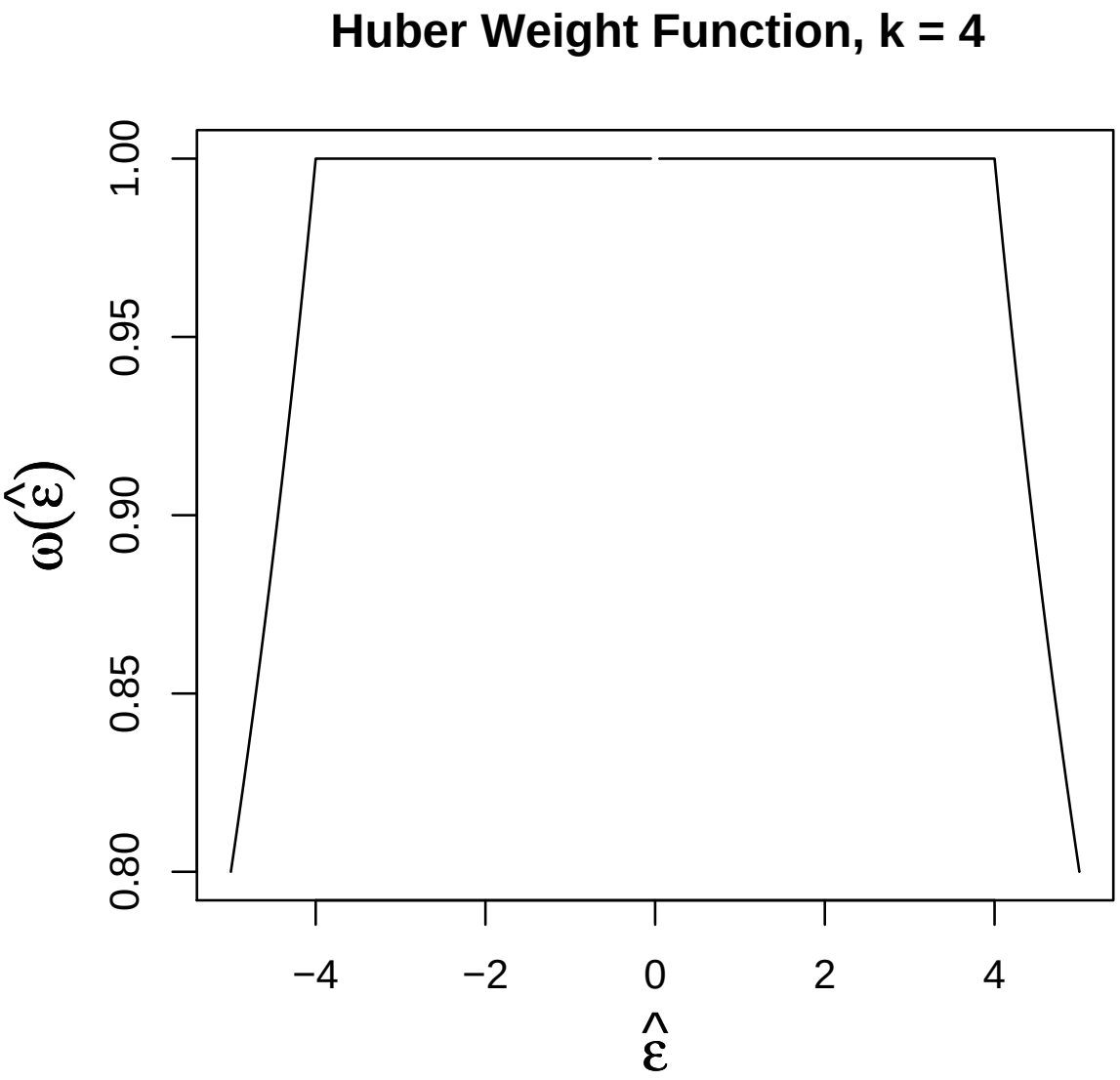
$$\omega_{\text{Huber}}(\hat{\varepsilon}) = \frac{\psi_{\text{Huber}}(\hat{\varepsilon})}{\hat{\varepsilon}} = \begin{cases} 1 & \text{for } |\hat{\varepsilon}| \leq k \\ k/|\hat{\varepsilon}| & \text{for } |\hat{\varepsilon}| > k \end{cases}$$

Influence functions: Huber example



Huber is like LS for non-outliers, and minimum absolute value for outliers

Influence functions: Huber example



M-estimation

How do we choose the tuning constant, k?

Lower k adds more resistance, but gives up more efficiency

We can choose k to give a certain efficiency relative to LS
under the assumption that the errors really are all Normal, $\varepsilon \sim \mathrm{N}(0, \sigma^2)$

First we need a robust estimate of the standard error of the regression, assuming Normality.

The median absolute deviation divided by 0.6745 serves this purpose:

$$\mathrm{median}|\hat{\varepsilon}_i|/0.6745$$

which we will call the scale, S. Then we will set $k = cS$

For Huber, to get 95% efficiency when $\varepsilon \sim \mathrm{N}(0, \sigma^2)$, set: $k = 1.345 \times S$

For Biweight, to get 95% efficiency when $\varepsilon \sim \mathrm{N}(0, \sigma^2)$, set: $k = 4.685 \times S$

Wait! How can we actually implement this method?

We have a problem!

- $\omega(\hat{\varepsilon})$ is a function of $\hat{\varepsilon}$
- We need $\hat{\beta}$ to calculate it
- But to get $\hat{\beta}$, we need $\omega(\hat{\varepsilon})$, and thus $\hat{\varepsilon}$ itself!

Wait! How can we actually implement this method?

We have a problem!

- $\omega(\hat{\boldsymbol{\varepsilon}})$ is a function of $\hat{\boldsymbol{\varepsilon}}$
- We need $\hat{\boldsymbol{\beta}}$ to calculate it
- But to get $\hat{\boldsymbol{\beta}}$, we need $\omega(\hat{\boldsymbol{\varepsilon}})$, and thus $\hat{\boldsymbol{\varepsilon}}$ itself!

Solution: *iterative estimation*

- Start with a guess of $\hat{\boldsymbol{\beta}}$.
- Calculate $\hat{\boldsymbol{\varepsilon}}$ for this guess
- Then calculate a new $\hat{\boldsymbol{\beta}}$
- And so on until we get stable estimates of $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\varepsilon}}$

Robust regresssion by Iteratively Re-Weighted Least Squares (IRWLS)

1. Select $\hat{\boldsymbol{\beta}}^0$. We might use $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$ for this, but there are other choices.

Robust regresssion by Iteratively Re-Weighted Least Squares (IRWLS)

1. Select $\hat{\boldsymbol{\beta}}^0$. We might use $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$ for this, but there are other choices.

2. Calculate $\hat{\boldsymbol{\varepsilon}}^0 = \mathbf{y}_i - \mathbf{x}_i \hat{\boldsymbol{\beta}}^0$.

Robust regresssion by Iteratively Re-Weighted Least Squares (IRWLS)

1. Select $\hat{\boldsymbol{\beta}}^0$. We might use $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$ for this, but there are other choices.

2. Calculate $\hat{\boldsymbol{\varepsilon}}^0 = \mathbf{y}_i - \mathbf{x}_i \hat{\boldsymbol{\beta}}^0$.

3. Repeat the following steps for $\ell = 1, \ldots$ until $\hat{\boldsymbol{\beta}}^\ell \approx \hat{\boldsymbol{\beta}}^{\ell-1}$:

Robust regresssion by Iteratively Re-Weighted Least Squares (IRWLS)

1. Select $\hat{\boldsymbol{\beta}}^0$. We might use $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$ for this, but there are other choices.

2. Calculate $\hat{\boldsymbol{\varepsilon}}^0 = \mathbf{y}_i - \mathbf{x}_i \hat{\boldsymbol{\beta}}^0$.

3. Repeat the following steps for $\ell = 1, \ldots$ until $\hat{\boldsymbol{\beta}}^\ell \approx \hat{\boldsymbol{\beta}}^{\ell-1}$:

   (a) Let $w_i^{\ell-1} = \omega\left(\hat{\boldsymbol{\varepsilon}}_i^{\ell-1}\right)$, where $\omega(\cdot)$ depends on the chosen influence function.

Robust regresssion by Iteratively Re-Weighted Least Squares (IRWLS)

1. Select $\hat{\boldsymbol{\beta}}^0$. We might use $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$ for this, but there are other choices.

2. Calculate $\hat{\boldsymbol{\varepsilon}}^0 = \mathbf{y}_i - \mathbf{x}_i\hat{\boldsymbol{\beta}}^0$.

3. Repeat the following steps for $\ell = 1, \ldots$ until $\hat{\boldsymbol{\beta}}^\ell \approx \hat{\boldsymbol{\beta}}^{\ell-1}$:

   (a) Let $w_i^{\ell-1} = \omega\left(\hat{\boldsymbol{\varepsilon}}_i^{\ell-1}\right)$, where $\omega(\cdot)$ depends on the chosen influence function.

   (b) Construct weight matrix $\mathbf{W}$ with $w_i^{\ell-1}$'s on diagonal.

Robust regresssion by Iteratively Re-Weighted Least Squares (IRWLS)

1. Select $\hat{\boldsymbol{\beta}}^0$. We might use $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$ for this, but there are other choices.

2. Calculate $\hat{\boldsymbol{\varepsilon}}^0 = \mathbf{y}_i - \mathbf{x}_i\hat{\boldsymbol{\beta}}^0$.

3. Repeat the following steps for $\ell = 1, \ldots$ until $\hat{\boldsymbol{\beta}}^\ell \approx \hat{\boldsymbol{\beta}}^{\ell-1}$:

   (a) Let $w_i^{\ell-1} = \omega\left(\hat{\boldsymbol{\varepsilon}}_i^{\ell-1}\right)$, where $\omega(\cdot)$ depends on the chosen influence function.

   (b) Construct weight matrix $\mathbf{W}$ with $w_i^{\ell-1}$'s on diagonal.

   (c) Solve $\hat{\boldsymbol{\beta}}^\ell = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y}$.

Robust regresssion by Iteratively Re-Weighted Least Squares (IRWLS)

1. Select $\hat{\boldsymbol{\beta}}^0$. We might use $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$ for this, but there are other choices.

2. Calculate $\hat{\boldsymbol{\varepsilon}}^0 = \mathbf{y}_i - \mathbf{x}_i\hat{\boldsymbol{\beta}}^0$.

3. Repeat the following steps for $\ell = 1, \ldots$ until $\hat{\boldsymbol{\beta}}^\ell \approx \hat{\boldsymbol{\beta}}^{\ell-1}$:

   (a) Let $w_i^{\ell-1} = \omega\left(\hat{\boldsymbol{\varepsilon}}_i^{\ell-1}\right)$, where $\omega(\cdot)$ depends on the chosen influence function.

   (b) Construct weight matrix $\mathbf{W}$ with $w_i^{\ell-1}$'s on diagonal.

   (c) Solve $\hat{\boldsymbol{\beta}}^\ell = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y}$.

   (d) Calculate $\hat{\boldsymbol{\varepsilon}}^\ell = \mathbf{y}_i - \mathbf{x}_i\hat{\boldsymbol{\beta}}^\ell$.

Robust regresssion by Iteratively Re-Weighted Least Squares (IRWLS)

1. Select $\hat{\boldsymbol{\beta}}^0$. We might use $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$ for this, but there are other choices.

2. Calculate $\hat{\boldsymbol{\varepsilon}}^0 = \mathbf{y}_i - \mathbf{x}_i\hat{\boldsymbol{\beta}}^0$.

3. Repeat the following steps for $\ell = 1, \ldots$ until $\hat{\boldsymbol{\beta}}^\ell \approx \hat{\boldsymbol{\beta}}^{\ell-1}$:

   (a) Let $w_i^{\ell-1} = \omega\left(\hat{\boldsymbol{\varepsilon}}_i^{\ell-1}\right)$, where $\omega(\cdot)$ depends on the chosen influence function.

   (b) Construct weight matrix $\mathbf{W}$ with $w_i^{\ell-1}$'s on diagonal.

   (c) Solve $\hat{\boldsymbol{\beta}}^\ell = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y}$.

   (d) Calculate $\hat{\boldsymbol{\varepsilon}}^\ell = \mathbf{y}_i - \mathbf{x}_i\hat{\boldsymbol{\beta}}^\ell$.

4. Report $\hat{\boldsymbol{\beta}}^\ell$ from the final iteration as the robust M-estimates $\hat{\boldsymbol{\beta}}_{\mathrm{M}}$.

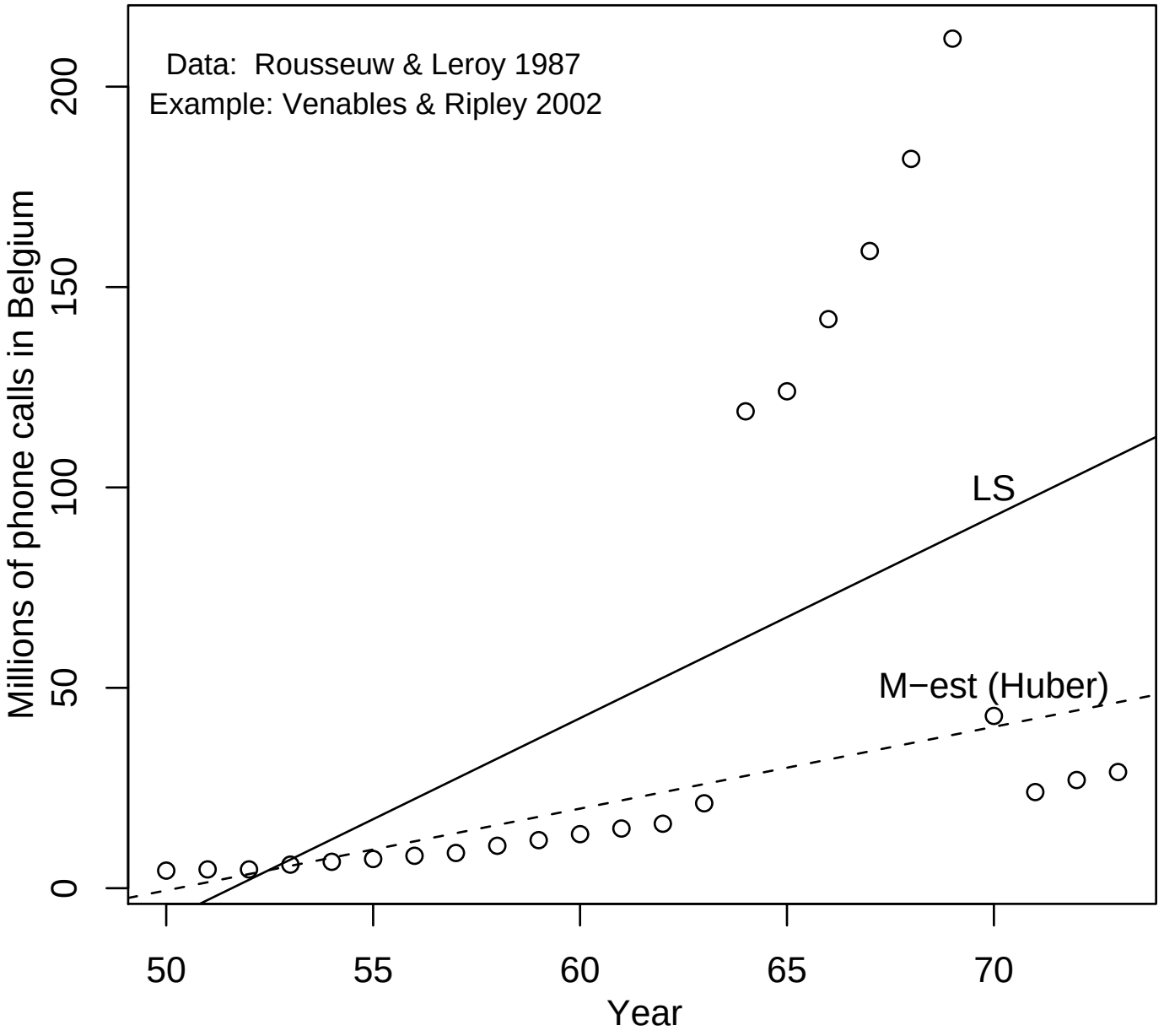
Doing this in R

```
library(MASS)
rlm(y~x,
    method="M",
    init="ls"              # Starting values:  a vector,
                           # or ls for least squares,
                           # or lts for Least Trimmed Squares
    psi=psi.huber,         # Options include psi.bisquare, psi.hampel
    k2 = 1.345             # Tuning constant for psi.huber
    maxit = 20,            # Maximum iterations of WLS to do
    acc = 1e-4             # Required precision of estimates
    )
```

Special problems for iterative estimation techniques:

- Does the choice of starting values ($\hat{\beta}^0$) influence the result?
- That is, does the iteration always lead to the same result?
- How do we know when we can stop?

Belgian phone calls example



Limitations of robust regression

- Less efficient than LS
  If there are no outliers, you have discarded valuable information

Limitations of robust regression

- Less efficient than LS
  If there are no outliers, you have discarded valuable information

- Not robust to outliers on X. Just Y.

Limitations of robust regression

- Less efficient than LS
  If there are no outliers, you have discarded valuable information

- Not robust to outliers on X. Just Y.

- Not even fully "robust" for Y, because every obs is still there

Limitations of robust regression

- Less efficient than LS
  If there are no outliers, you have discarded valuable information

- Not robust to outliers on X. Just Y.

- Not even fully "robust" for Y, because every obs is still there

- What if the outlier(s) have high leverage? Then they won't even have large $\hat{\varepsilon}$ to begin with, so robust regression won't really be enough

Breakdown points

The robust regression methods discussed so far downweight outliers compared to LS

But they are still vulnerable to extreme/numerous outliers. Imagine two cases:

- Case 1: A large fraction (e.g., one-third) of the data are high leverage outliers (i.e., come from a different distribution)

Breakdown points

The robust regression methods discussed so far downweight outliers compared to LS

But they are still vulnerable to extreme/numerous outliers. Imagine two cases:

- Case 1: A large fraction (e.g., one-third) of the data are high leverage outliers (i.e., come from a different distribution)

- Case 2: A single high-leverage outlier is massive, such that it dominates even a robust regression
  (e.g., your sample of 1000 American's incomes drew Bill Gates)

Breakdown points

The robust regression methods discussed so far downweight outliers compared to LS

But they are still vulnerable to extreme/numerous outliers. Imagine two cases:

- Case 1: A large fraction (e.g., one-third) of the data are high leverage outliers (i.e., come from a different distribution)
- Case 2: A single high-leverage outlier is massive, such that it dominates even a robust regression
  (e.g., your sample of 1000 American's incomes drew Bill Gates)

In either case, there is no logical limit to the influence of the outliers, even in robust regression

Breakdown points

The robust regression methods discussed so far downweight outliers compared to LS

But they are still vulnerable to extreme/numerous outliers. Imagine two cases:

- Case 1: A large fraction (e.g., one-third) of the data are high leverage outliers (i.e., come from a different distribution)
- Case 2: A single high-leverage outlier is massive, such that it dominates even a robust regression
  (e.g., your sample of 1000 American's incomes drew Bill Gates)

In either case, there is no logical limit to the influence of the outliers, even in robust regression

Define the **breakdown point** as the fraction of observations which can shift the estimator arbitrarily far from the truth for the non-outlying subset

Breakdown points

The robust regression methods discussed so far downweight outliers compared to LS

But they are still vulnerable to extreme/numerous outliers. Imagine two cases:

- Case 1: A large fraction (e.g., one-third) of the data are high leverage outliers (i.e., come from a different distribution)
- Case 2: A single high-leverage outlier is massive, such that it dominates even a robust regression
  (e.g., your sample of 1000 American's incomes drew Bill Gates)

In either case, there is no logical limit to the influence of the outliers, even in robust regression

Define the **breakdown point** as the fraction of observations which can shift the estimator arbitrarily far from the truth for the non-outlying subset

The breakdown point of both LS and robust regression is $1/N$, the lowest possible

Breakdown points

The robust regression methods discussed so far downweight outliers compared to LS

But they are still vulnerable to extreme/numerous outliers. Imagine two cases:

- Case 1: A large fraction (e.g., one-third) of the data are high leverage outliers (i.e., come from a different distribution)
- Case 2: A single high-leverage outlier is massive, such that it dominates even a robust regression
  (e.g., your sample of 1000 American's incomes drew Bill Gates)

In either case, there is no logical limit to the influence of the outliers, even in robust regression

Define the **breakdown point** as the fraction of observations which can shift the estimator arbitrarily far from the truth for the non-outlying subset

The breakdown point of both LS and robust regression is $1/N$, the lowest possible

Not coincidentally, this happens to be the breakdown point for the mean of a distribution

Resistant regression

What would be more resistant than LS and robust regression?

Resistant regression

What would be more resistant than LS and robust regression?

Let's turn to univariate statistics.

Resistant regression

What would be more resistant than LS and robust regression?

Let's turn to univariate statistics.

The mean is a non-resistant measure of the center of a distribution

Resistant regression

What would be more resistant than LS and robust regression?

Let's turn to univariate statistics.

The mean is a non-resistant measure of the center of a distribution

A very resistant measure of the center is the median.

Fully 50 % - 1 of the observations could shift and not change the median

Hence the breakdown point of the median is 0.5, the highest possible.

Resistant regression

Regression models with high breakdown points are *resistant*

A price: Hard to estimate, very inefficient

Least Median Squares

Least Quantile Squares

Least Trimmed Squares

Resistant regression: Least Median Squares

The median is a very robust measure, so let's build a regression model around it

Choose $\hat{\boldsymbol{\beta}}_{\mathrm{LMS}}$ such that we minimize the median squared residual,

Resistant regression: Least Median Squares

The median is a very robust measure, so let's build a regression model around it

Choose $\hat{\boldsymbol{\beta}}_{\mathrm{LMS}}$ such that we minimize the median squared residual,

$$\min \operatorname{median}(\hat{\varepsilon}_i^2)$$

Note that we don't know which residual is the middle-most one until we run a regression

Then we iterate through many possible $\hat{\boldsymbol{\beta}}$'s until we find a set that makes this median residual as small as possible

Properties of Least Median Squares

- High breakdown point (approaching 50%)

Properties of Least Median Squares

- High breakdown point (approaching 50%)
- Resistant to outliers on Y and X

Properties of Least Median Squares

- High breakdown point (approaching 50%)
- Resistant to outliers on Y and X
- Very inefficient

Properties of Least Median Squares

- High breakdown point (approaching 50%)
- Resistant to outliers on Y and X
- Very inefficient
- Computationally difficult to estimate

Properties of Least Median Squares

- High breakdown point (approaching 50%)
- Resistant to outliers on Y and X
- Very inefficient
- Computationally difficult to estimate
- Difficult to calculate standard errors

Resistant regression: Other methods

Least Quantile Squares:

Choose $\hat{\boldsymbol{\beta}}_{\mathrm{LQS}}$ to minimize the qth quantile of the residuals,

$$\min \text{ quantile}(\hat{\varepsilon}_i^2, q)$$

Resistant regression: Other methods

Least Quantile Squares:

Choose $\hat{\boldsymbol{\beta}}_{\mathrm{LQS}}$ to minimize the qth quantile of the residuals,

$$\min \text{ quantile}(\hat{\varepsilon}_i^2, q)$$

Least Trimmed Squares:

Choose $\hat{\boldsymbol{\beta}}_{\mathrm{LTS}}$ to minimize the q smallest residuals

$$\min \sum_{i=1}^{q} \hat{\varepsilon}_i^2 \text{ such that } \hat{\varepsilon}_i^2 < \hat{\varepsilon}_j^2 \quad \forall i \neq j$$

LTS is as resistant as LMS but more efficient. Other methods available too (S-estimation).

All resistant estimators have very low efficiency compared to LS (<30% or so)

Resistant regression: In R

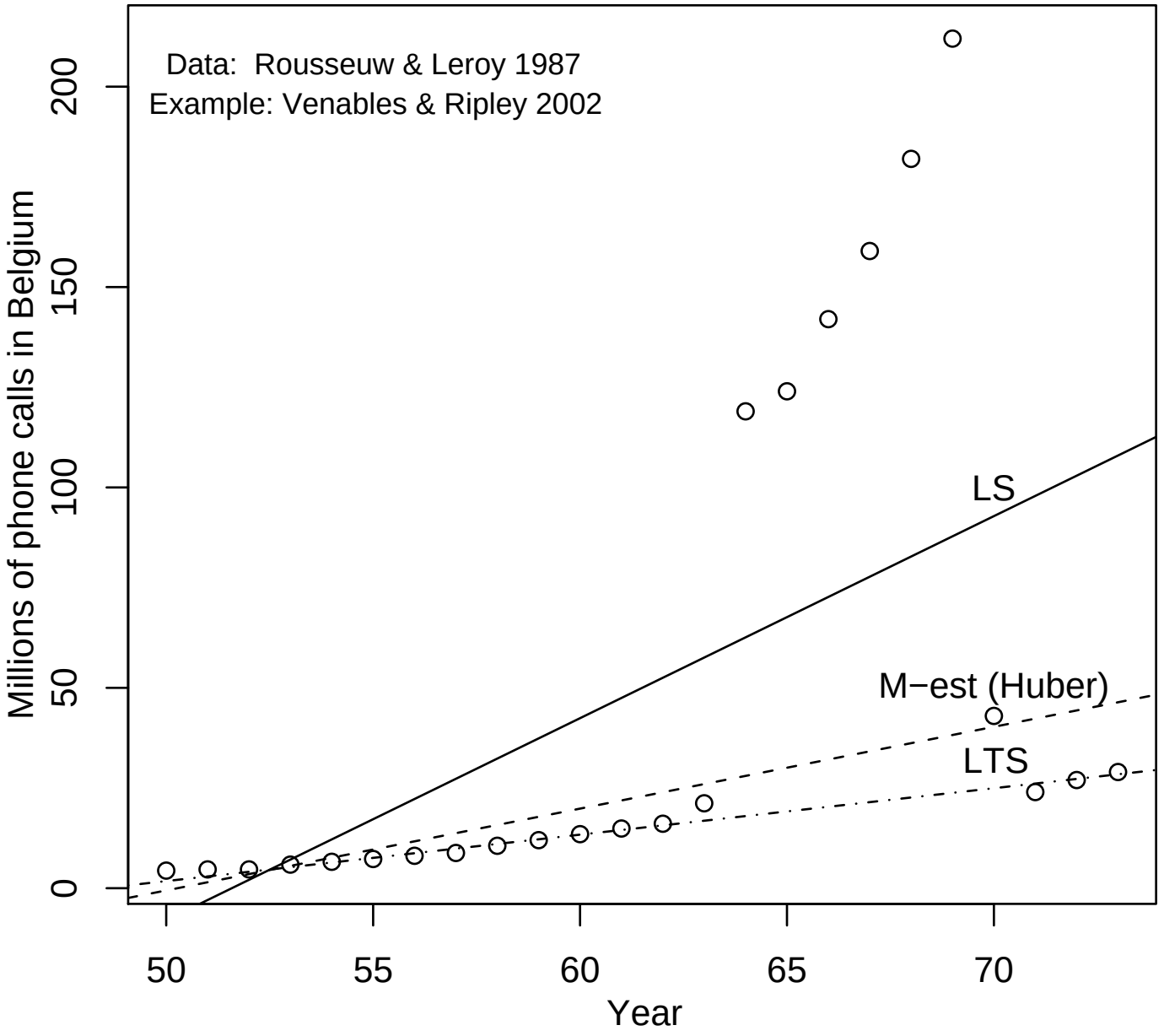
Many of the above methods are available through `lqs`

```
# Least median squares
library(MASS)
lqs(y~x,
    method = "lms"
    )

# Least quantile squares
library(MASS)
lqs(y~x,
    quantile,
    method = "lqs",
    )

# Least trimmed squares
library(MASS)
lqs(y~x,
    method = "lts"
    )
```

Resistant regression: In R

```
# S-estimator
library(MASS)
lqs(y~x,
    method = "S"
    )
```

Belgian phone calls example



Resistant regression and beyond

We can also combine resistant and robust regression to get some benefits of each

Resistant regression and beyond

We can also combine resistant and robust regression to get some benefits of each

1. Estimate an (inefficient) resistant regression

Resistant regression and beyond

We can also combine resistant and robust regression to get some benefits of each

1. Estimate an (inefficient) resistant regression

2. Use the result as a starting point for robust regression

Resistant regression and beyond

We can also combine resistant and robust regression to get some benefits of each

1. Estimate an (inefficient) resistant regression

2. Use the result as a starting point for robust regression

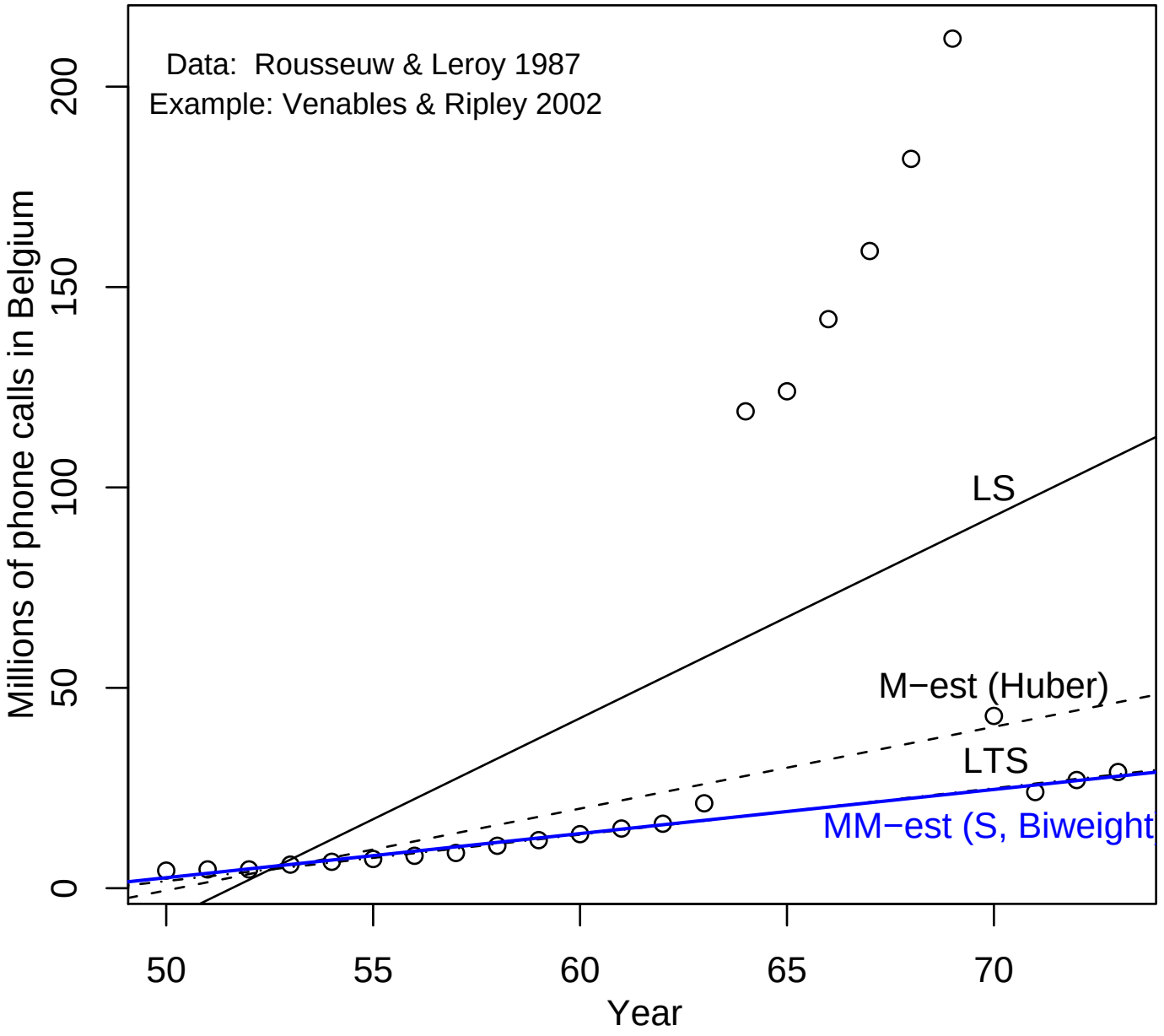
3. Resulting MM-estimator has high breakdown from step 1,
   and high efficiency from step 2

In R:

```
library(MASS)
rlm(y~x,
    method = "MM")
```

performs MM-estimation using the Biweight influence function initialized by a resistant S-estimator

Belgian phone calls example



Example: Redistribution in Rich Democracies

Let's look at an example from Iversen and Soskice (2003).

(Warning: I will ignore most of their data analysis and theory for didactic purposes)

Example: Redistribution in Rich Democracies

Let's look at an example from Iversen and Soskice (2003).

(Warning: I will ignore most of their data analysis and theory for didactic purposes)

IS are interested in the relationship between party systems and redistributive effort

Example: Redistribution in Rich Democracies

Let's look at an example from Iversen and Soskice (2003).

(Warning: I will ignore most of their data analysis and theory for didactic purposes)

IS are interested in the relationship between party systems and redistributive effort

They capture the first using the effective number of parties

Example: Redistribution in Rich Democracies

Let's look at an example from Iversen and Soskice (2003).

(Warning: I will ignore most of their data analysis and theory for didactic purposes)

IS are interested in the relationship between party systems and redistributive effort

They capture the first using the effective number of parties

& the second using the % of people lifted from poverty by taxes and transfers

Example: Redistribution in Rich Democracies

Let's look at an example from Iversen and Soskice (2003).

(Warning: I will ignore most of their data analysis and theory for didactic purposes)

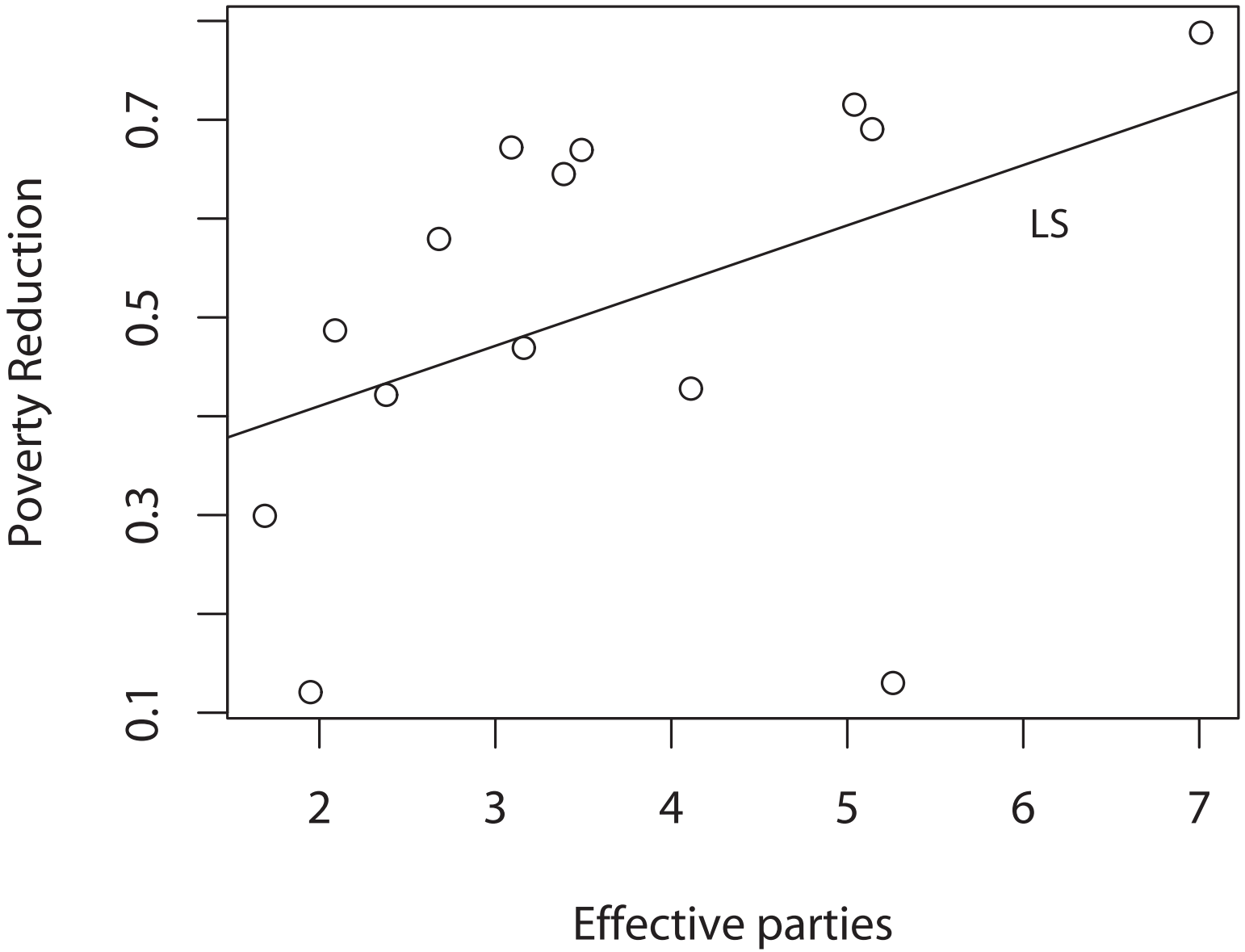
IS are interested in the relationship between party systems and redistributive effort

They capture the first using the effective number of parties

& the second using the % of people lifted from poverty by taxes and transfers

Let's look at a scatterplot of the data

Example: Redistribution in Rich Democracies



Example: Redistribution in Rich Democracies

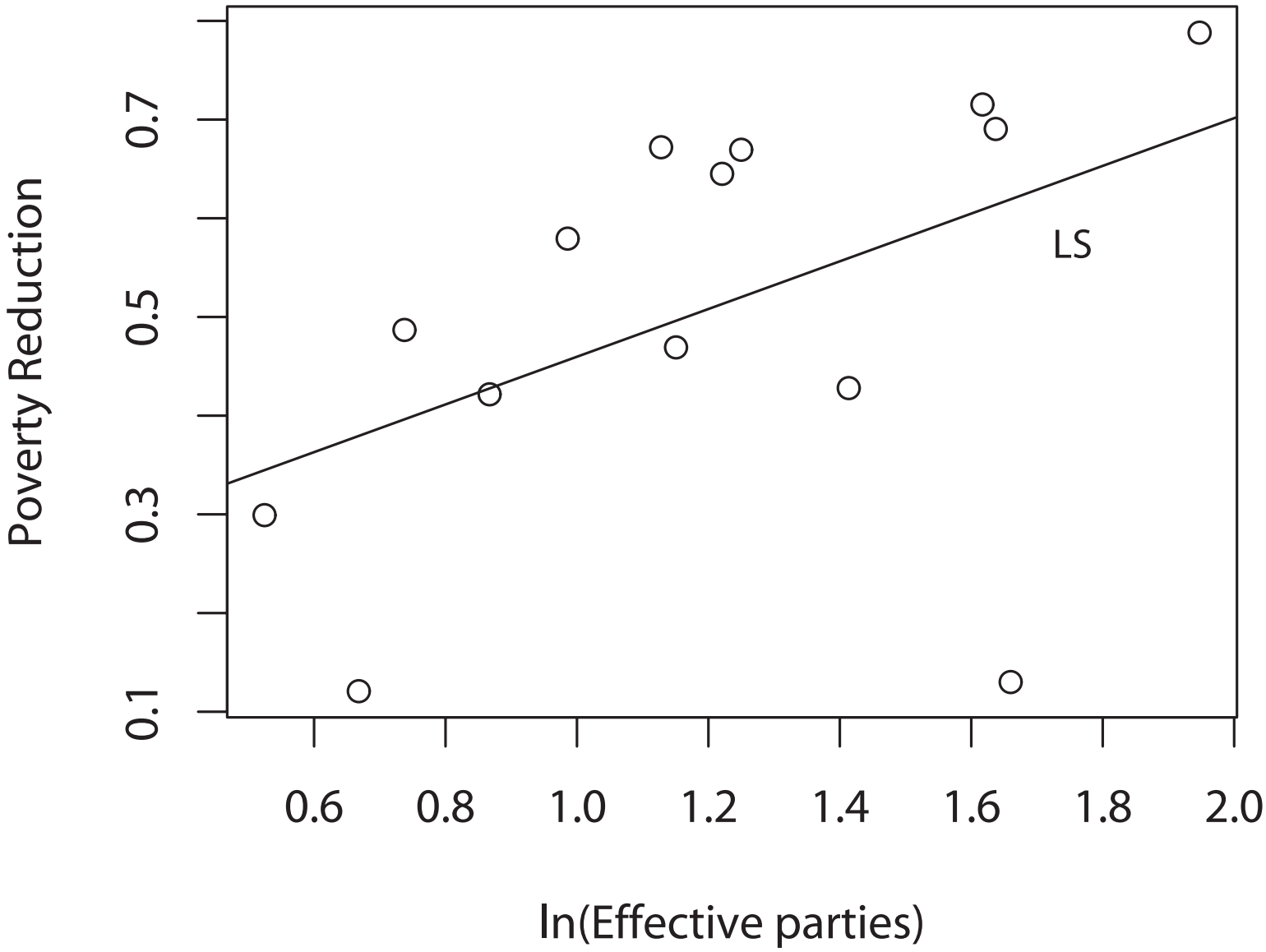
Now it's your turn. We're going to use linear regression to analyze these data

What do we need to do to get it right?

Example: Redistribution in Rich Democracies

First: Because it is a count, and the plot suggests it,
Let's log the independent variable

Example: Redistribution in Rich Democracies



What next?

Example: Redistribution in Rich Democracies

Second: Because it is a proportion, we will logit transform PovRed
that is, the new DV is $\ln(\text{povred}/(1-\text{povred})$

Example: Redistribution in Rich Democracies

Second: Because it is a proportion, we will logit transform PovRed
that is, the new DV is $\ln(\text{povred}/(1-\text{povred})$

(This second step adds considerable interpretative difficulty,
so in practice, we might skip it, if the relation looks essentially linear)

Example: Redistribution in Rich Democracies

Second: Because it is a proportion, we will logit transform PovRed
that is, the new DV is $\ln(\text{povred}/(1-\text{povred})$

(This second step adds considerable interpretative difficulty,
so in practice, we might skip it, if the relation looks essentially linear)

Technical details: I plot the fit by reversing the logit transformation
on a set of fitted values

Example: Redistribution in Rich Democracies

Second: Because it is a proportion, we will logit transform PovRed
that is, the new DV is $\ln(\text{povred}/(1-\text{povred})$

(This second step adds considerable interpretative difficulty,
so in practice, we might skip it, if the relation looks essentially linear)

Technical details: I plot the fit by reversing the logit transformation
on a set of fitted values

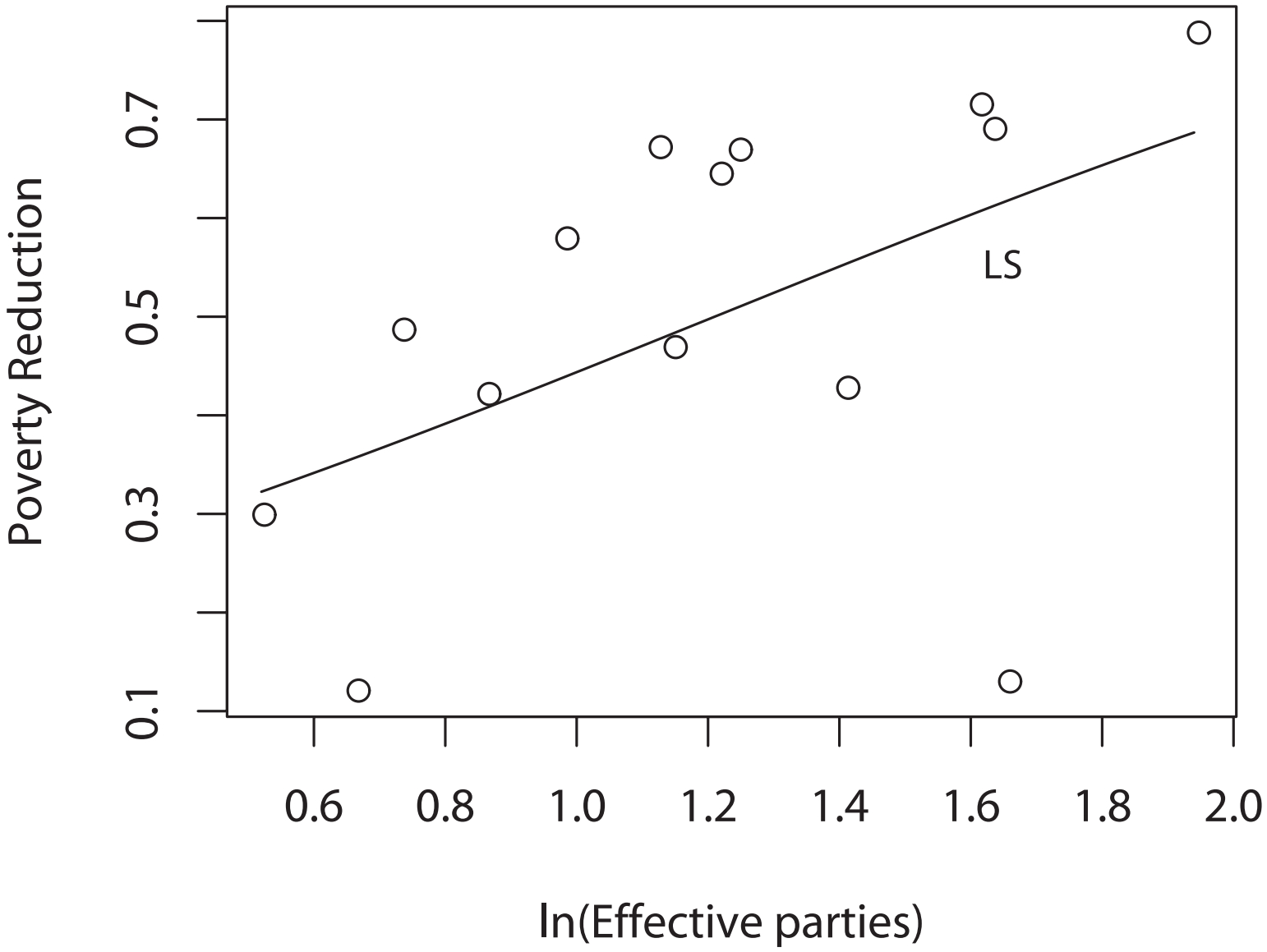
That is, I calculate

$$\hat{y}_k^* = \frac{1}{1+\exp(-\hat{y}_k)}$$

for $k = \{0.50, 0.51, \ldots 1.99, 2.00\}$

This puts a little S-curvature in the fitted line.

Example: Redistribution in Rich Democracies



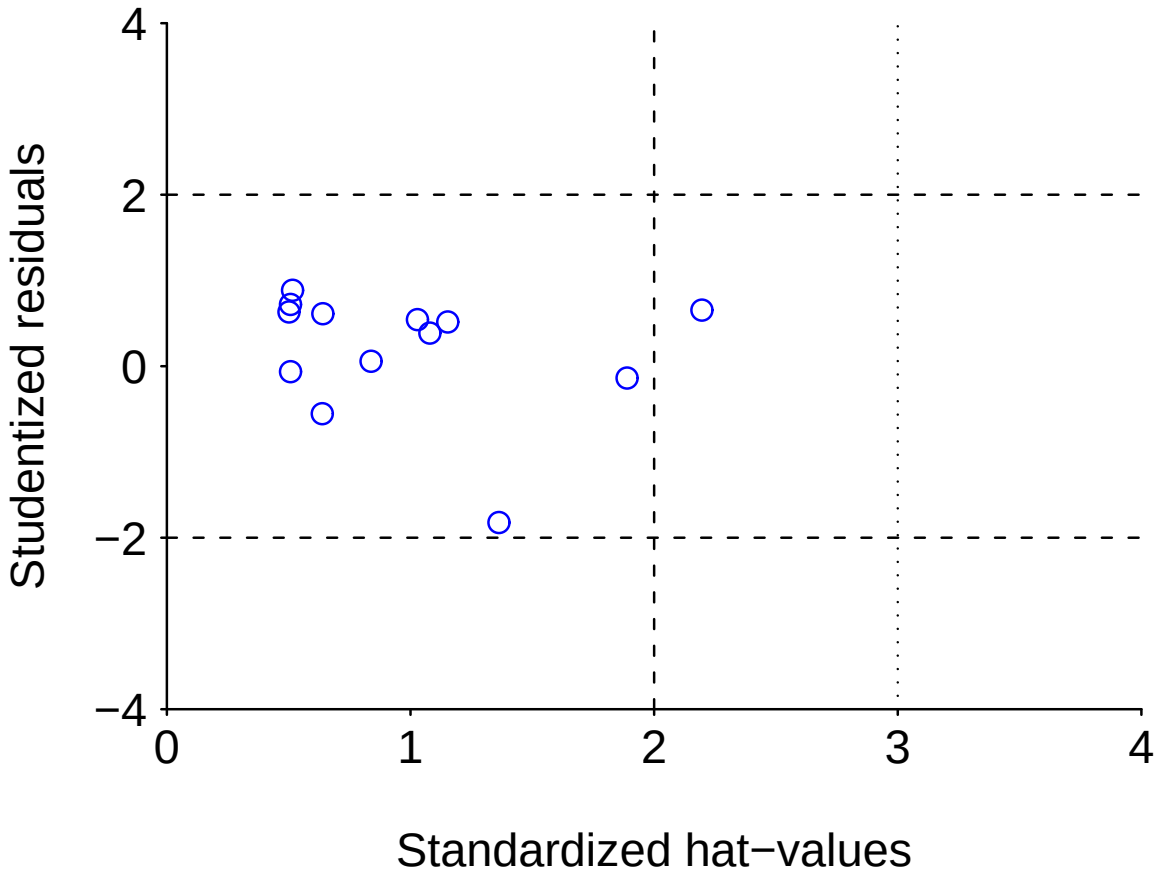
Logit transformation only slightly different (more in a moment). What next?

Example: Redistribution in Rich Democracies

Third: Outliers. There are two. One is particularly gruesome.

Let's look at the influence plot

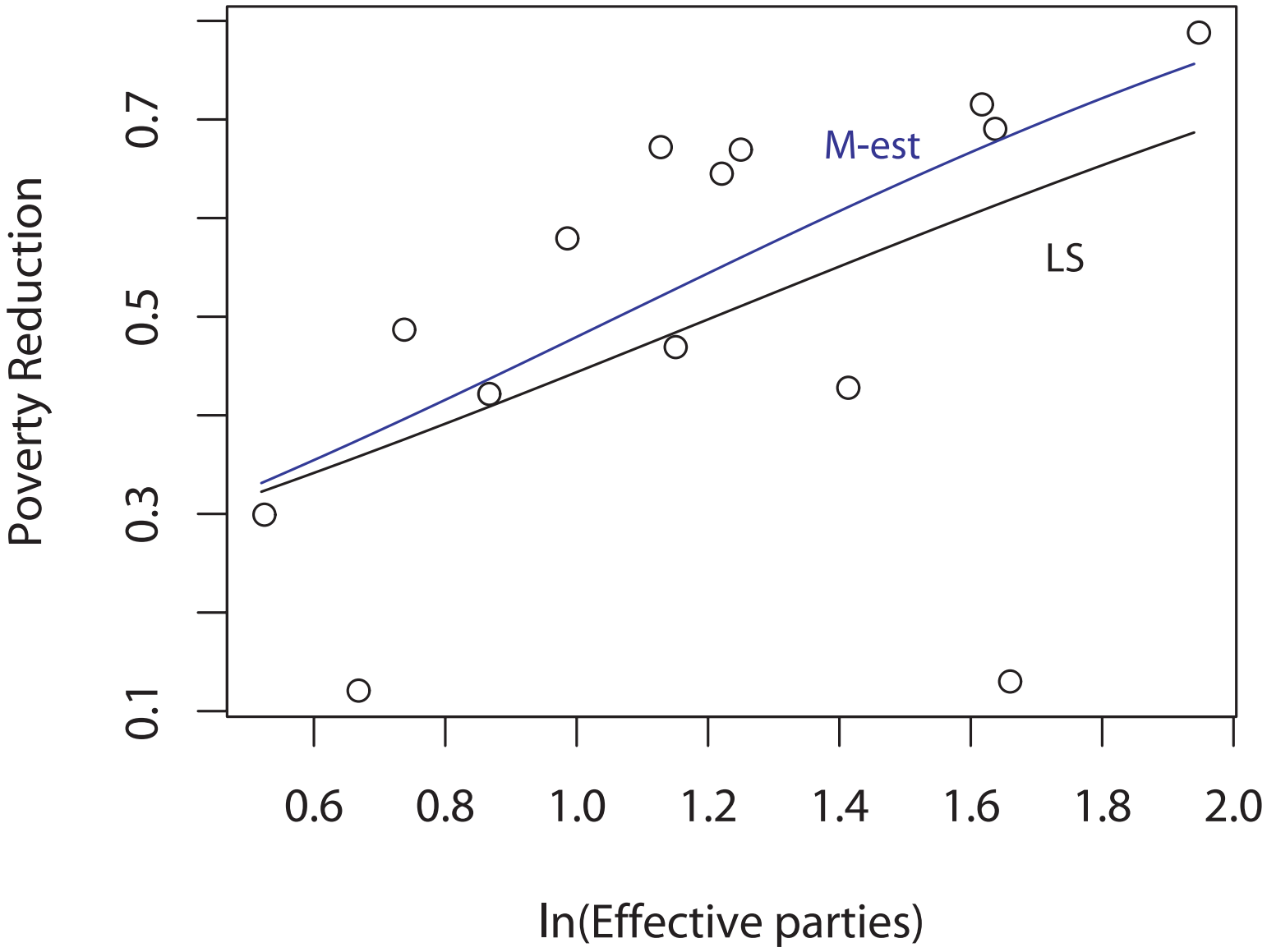
Example: Redistribution in Rich Democracies



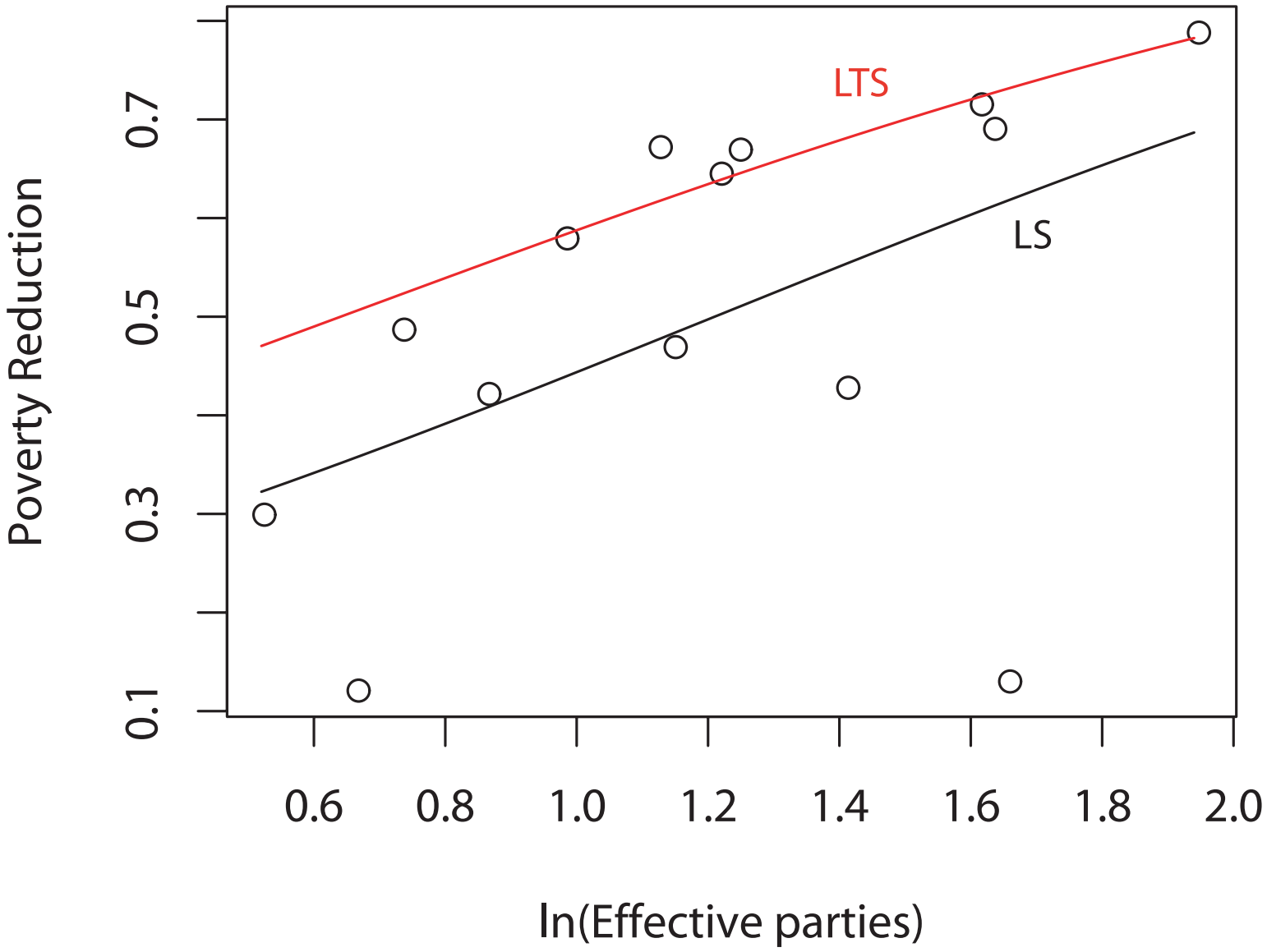
Two or three notable outliers

Now, let's try our battery of robust and resistant estimators

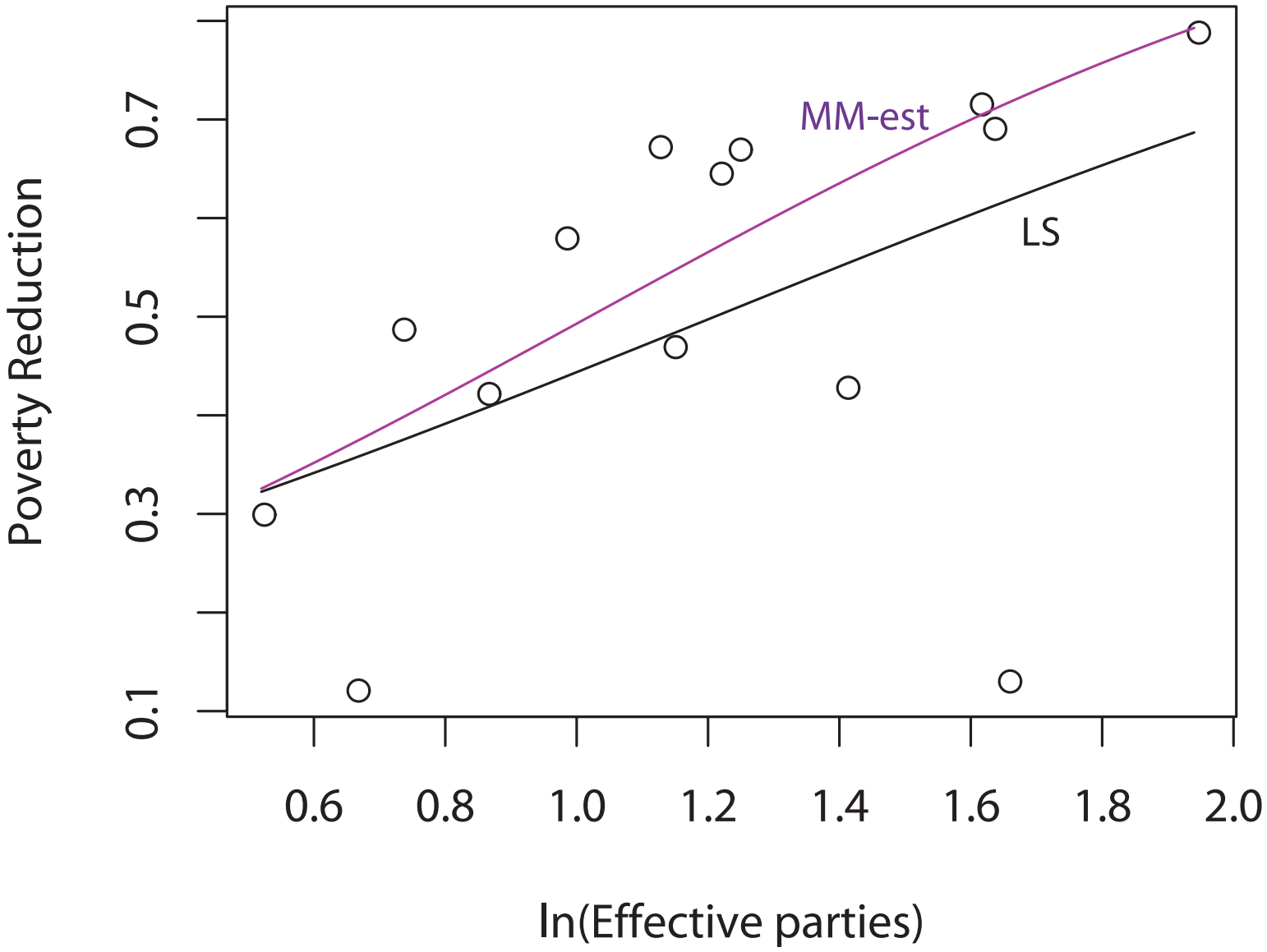
Example: Redistribution in Rich Democracies



Example: Redistribution in Rich Democracies



Example: Redistribution in Rich Democracies



Example: Redistribution in Rich Democracies

Which fit looks best?

Example: Redistribution in Rich Democracies

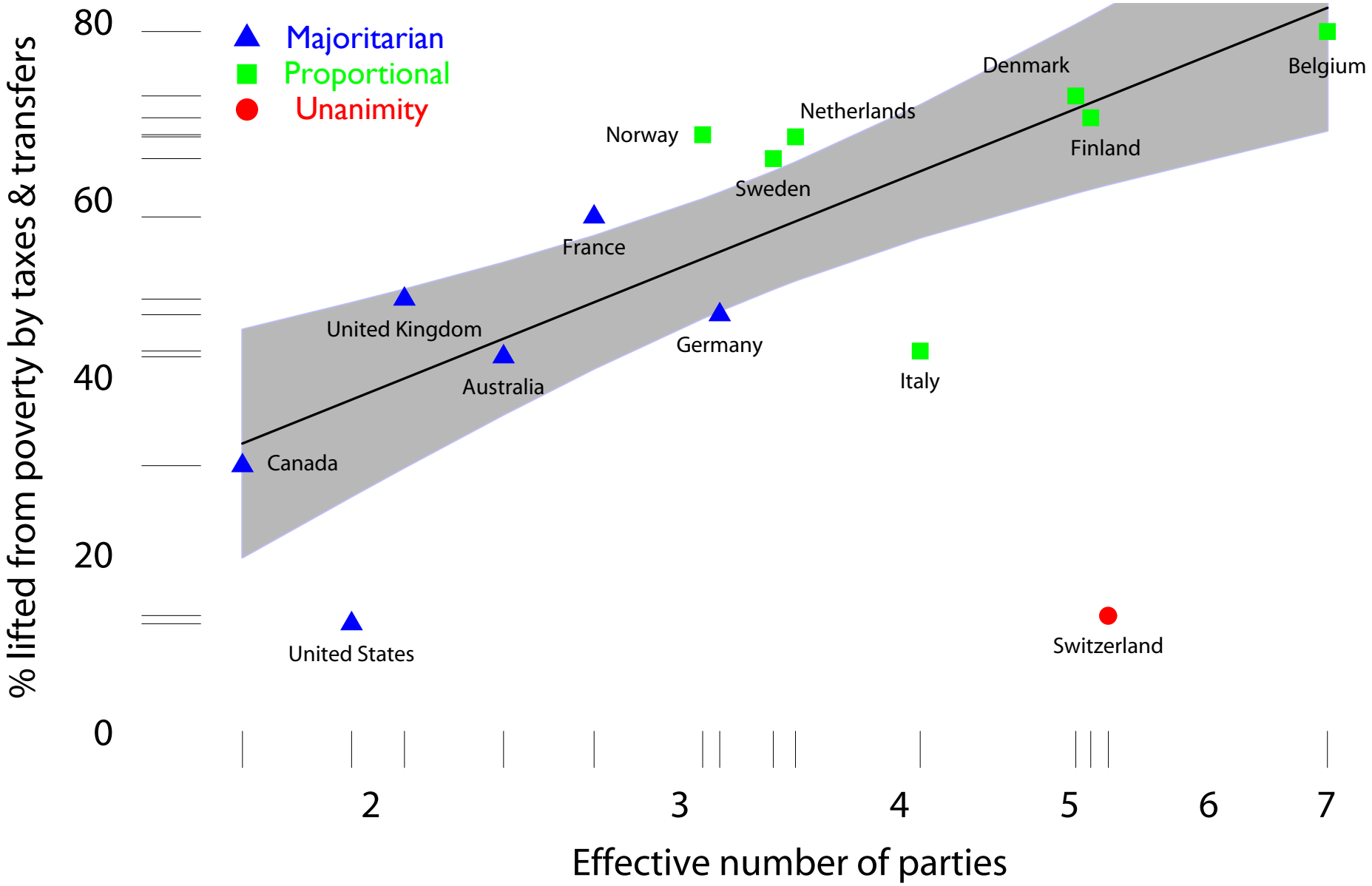
Which fit looks best?

Here's the tabular summary (note the DV is logit(povred), so take care with coefs)

	LS	M est.	LTS	MM est.
Ln(Effective Parties)	1.076	1.292	0.986	1.458
	(0.616)	(0.391)		(0.364)
Intercept	−1.302	−1.374	−0.631	−1.486
	(0.781)	(0.495)		(0.461)

Let's take one last look, this time with more information . . .

Learning from outliers: the data revealed



Source: Torben Iversen and David Soskice, 2002, "Why do some democracies redistribute more than others?", manuscript, Harvard University. Redrawn.

Learning from outliers

Outliers are a nuisance, not substance

Learning from outliers

Outliers are a nuisance, not substance

We'd like to explain all the data, rather than throw out observations as "different"

Learning from outliers

Outliers are a nuisance, not substance

We'd like to explain all the data, rather than throw out observations as "different"

Best case: use substantive knowledge to convert an outlier to an explanation

Learning from outliers

Outliers are a nuisance, not substance

We'd like to explain all the data, rather than throw out observations as "different"

Best case: use substantive knowledge to convert an outlier to an explanation

If we can't reach that case, we may rely on robust/resistant methods more heavily

Learning from outliers

Outliers are a nuisance, not substance

We'd like to explain all the data, rather than throw out observations as "different"

Best case: use substantive knowledge to convert an outlier to an explanation

If we can't reach that case, we may rely on robust/resistant methods more heavily

Regardless, check whether your results are sensitive to outliers

Learning from outliers

Outliers are a nuisance, not substance

We'd like to explain all the data, rather than throw out observations as "different"

Best case: use substantive knowledge to convert an outlier to an explanation

If we can't reach that case, we may rely on robust/resistant methods more heavily

Regardless, check whether your results are sensitive to outliers

Easy to do in R—influence plots, `rlm`, and `lqs` take seconds

Learning from outliers

Outliers are a nuisance, not substance

We'd like to explain all the data, rather than throw out observations as "different"

Best case: use substantive knowledge to convert an outlier to an explanation

If we can't reach that case, we may rely on robust/resistant methods more heavily

Regardless, check whether your results are sensitive to outliers

Easy to do in R—influence plots, `rlm`, and `lqs` take seconds

One more lesson for today: a mystery. . .

The case of the missing standard errors

LTS doesn't return any standard errors, just a point estimate

The case of the missing standard errors

LTS doesn't return any standard errors, just a point estimate

I argued earlier that results without estimates of uncertainty were dangerous

What can we do?

The case of the missing standard errors

LTS doesn't return any standard errors, just a point estimate

I argued earlier that results without estimates of uncertainty were dangerous

What can we do?

Let's think about what standard errors are:

Expected difference between the $\hat{\beta}$ and the average $\bar{\hat{\beta}}$ from repeated samples

The case of the missing standard errors

LTS doesn't return any standard errors, just a point estimate

I argued earlier that results without estimates of uncertainty were dangerous

What can we do?

Let's think about what standard errors are:

Expected difference between the $\hat{\beta}$ and the average $\bar{\hat{\beta}}$ from repeated samples

If we could draw more samples from the population, we could:

1. Run a separate regression on each sample

2. Take the standard deviation of the coefficients across samples

Is there any way we could *simulate* this using just our extant sample?

Re-sampling

There's a nifty, very general trick we can use, based on an analogy:

Samples are to the Population

Re-sampling

There's a nifty, very general trick we can use, based on an analogy:

Samples are to the Population

as

Subsamples are to the Sample

Re-sampling

There's a nifty, very general trick we can use, based on an analogy:

Samples are to the Population

as

Subsamples are to the Sample

We can't draw any more samples from the population.

But we can **re**-sample from our own sample.

The bootstrap

Re-sampling means

1. drawing N observations with replacement from a dataset of size N

The bootstrap

Re-sampling means

1. drawing N observations with replacement from a dataset of size N

2. running a regression on each re-sampled dataset

The bootstrap

Re-sampling means

1. drawing N observations with replacement from a dataset of size N

2. running a regression on each re-sampled dataset

3. repeating to build up a distribution of results

The bootstrap

Re-sampling means

1. drawing N observations with replacement from a dataset of size N
2. running a regression on each re-sampled dataset
3. repeating to build up a distribution of results

It turns out this distribution of re-sampled statistics approximates the dist of sampled statistics

Its standard deviation estimates the standard error

So we can simulate SEs even if we don't know how to calculate them analytically

The bootstrap

Bootstrapping can be time consuming, especially if the underlying analysis is too

And you may need to replicate many times to get stable se's (how can you tell?)

The bootstrap

Bootstrapping can be time consuming, especially if the underlying analysis is too

And you may need to replicate many times to get stable se's (how can you tell?)

I ran 250,000 bootstrap replications of the least trimmed squares regression for the Poverty Reduction data to get 1 sig digit

The bootstrap

Bootstrapping can be time consuming, especially if the underlying analysis is too

And you may need to replicate many times to get stable se's (how can you tell?)

I ran 250,000 bootstrap replications of the least trimmed squares regression for the Poverty Reduction data to get 1 sig digit

The code I used:

```
# A function that runs the underlying regression
leasttrimmed <- function(d,i) {
  lts.result <- lqs(d[,1]~d[,2:ncol(d)], method = "lts",
                                    nsamp="exact",subset=i)
  lts.result$coefficients
}

# Putting the data in a single matrix
yx <- cbind(y,x)

# Running the bootstrap 250,000 times
lts.boot <- boot(yx,leasttrimmed,R=250000,stype="i")
```

LTS redux

It turns out the LTS results are very poorly estimated

LTS redux

It turns out the LTS results are very poorly estimated

Recall that LTS is very inefficient (throws away a lot of potentially good data)

LTS redux

It turns out the LTS results are very poorly estimated

Recall that LTS is very inefficient (throws away a lot of potentially good data)

Hence they aren't much use here

But in larger datasets with lots of potential outliers, LTS is worth checking

	LS	M est.	LTS	MM est.
Ln(Effective Parties)	1.076	1.292	0.986	1.458
	(0.616)	(0.391)	(2.5)	(0.364)
Intercept	−1.302	−1.374	−0.631	−1.486
	(0.781)	(0.495)	(3.5)	(0.461)