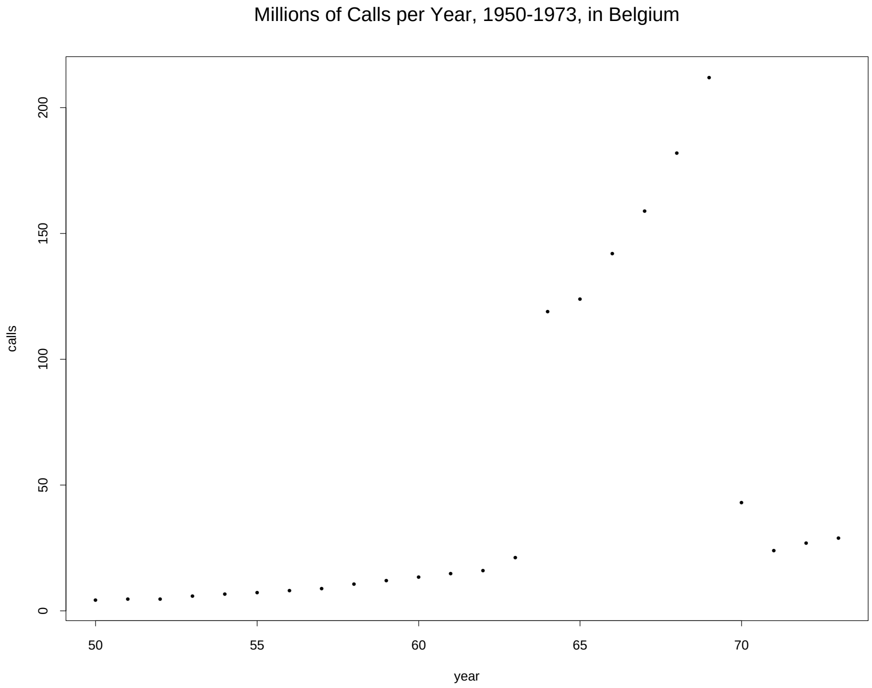
36-401 Class Notes and Exercise
Robust Regression I

Introduction and WLS

The plot below comes from a study of the number of phone calls per year in Belgium, in millions, for the years 1950–1973.



There are some obvious problems with this data (what *are* those six outliers??), but we will use it to illustrate *robust regression.* When

- there are outliers in the data; or
- the distribution of ϵ in $Y = X\beta + \epsilon$ has fatter tails than a Normal distribution

then robust regression provides an alternative to simply eliminating the outliers and the applying ordinary least squares (OLS) regression.

Robust regression is based on a computational technique called *weighted least squares (WLS).* The idea of weighted least squares is that instead of minimizing

$$RSS = \sum_i (y_i - \underline{x_i\beta})^2$$

we try to minimize

$$RSS_{\underline{w}} = \sum_i w_i \cdot (y_i - \underline{x_i\beta})^2$$

where we specify the weights w_i to indicate how much attention should be paid to making the i^{th} residual $(y_i - \underline{x_i\beta})^2$ small, when finding $\underline{\hat{\beta}}$.

It turns out that this is easy to do, using standard least-squares software. If our regression equation is

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_{K-1} x_{i(K-1)} + \epsilon_i$$

and we want to do WLS with weights w_i, then we simply multiply through by the square roots of the weights $\sqrt{w_i}$,

$$\begin{aligned} \sqrt{w_i} y_i &= \beta_0 \sqrt{w_i} + \beta_1 \sqrt{w_i} x_{i1} + \cdots + \beta_{K-1} \sqrt{w_i} x_{i(K-1)} + \sqrt{w_i} \epsilon_i, \text{ or} \\ y_i^* &= \beta_0 x_{i0}^* + \beta_1 x_{i1}^* + \cdots + \beta_{K-1} x_{i(K-1)}^* + \epsilon_i^* \end{aligned}$$

and apply OLS to the resulting equation. In matrix form, this is equivalent to changing from the model $\underline{y} = X\underline{\beta} + \underline{\epsilon}$ to the model

$$\begin{aligned} W^{1/2} \underline{y} &= W^{1/2} X \underline{\beta} + W^{1/2} \underline{\epsilon}, \text{ or} \\ \underline{y}^* &= X^* \underline{\beta} + \underline{\epsilon}^* \end{aligned}$$

where

$$W^{1/2} = \begin{bmatrix} w_1^{1/2} & 0 & \cdots & 0 \\ 0 & w_2^{1/2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n^{1/2} \end{bmatrix}$$

The WLS estimates of the β's are just

$$B = (X^{*\prime} X^*)^{-1} X^{*\prime} \underline{y}^* = (X' W X)^{-1} X' W \underline{y}$$

In Splus it's even easier: we specify the weights w_i using the `weights` argument of the `lm()` command. Instead of doing

```
lm(calls ~ year, data=phones)
```

for the Belgian phones data, if we want to use WLS, we just specify the weights:

```
lm(calls ~ year, data=phones, weights=wi)
```

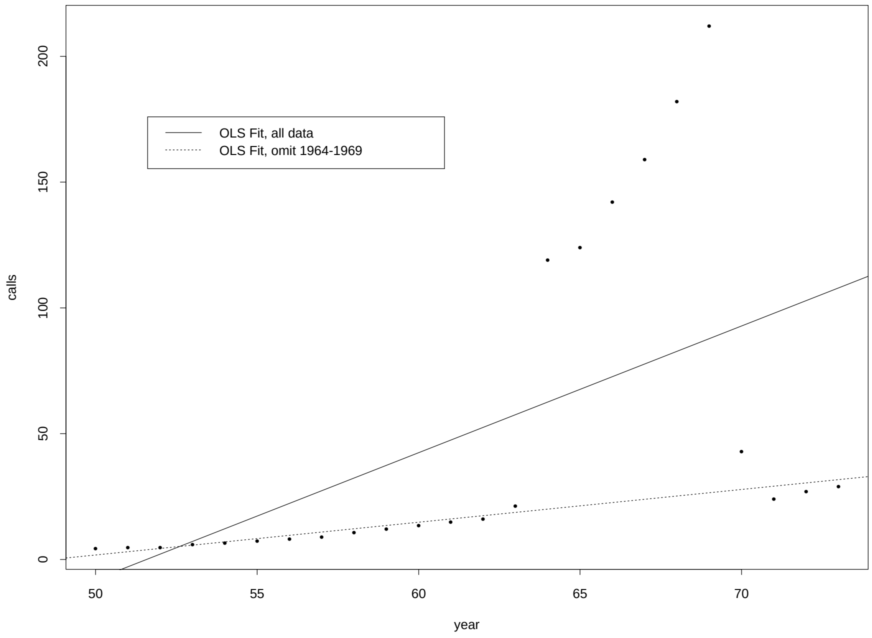
Before we do any robust regression, we will use WLS to illustrate how we might try to down-weight the outliers in the Belgian telephone data.

The following ideas are a little silly, but they illustrate what we are trying to get at, by downweighting "outliers" when fitting the regression.

Slighlty Silly Idea #1

First, we could just set the weights corresponding the outliers to be 0, and set the other weights equal to 1. This has the same effect of just dropping the outliers from the analysis:

```
> attach(phones)
> weights _ c(rep(1,14),rep(0,6),rep(1,4))
> weights
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1
> plot(year,calls)
> abline(lm(calls~year))
> abline(lm(calls~year,weights=weights),lty=2)
> legend(locator(),c("OLS Fit, all data","OLS Fit, omit 1964-1969"),lty=1:2)
```



In this case we have minimized

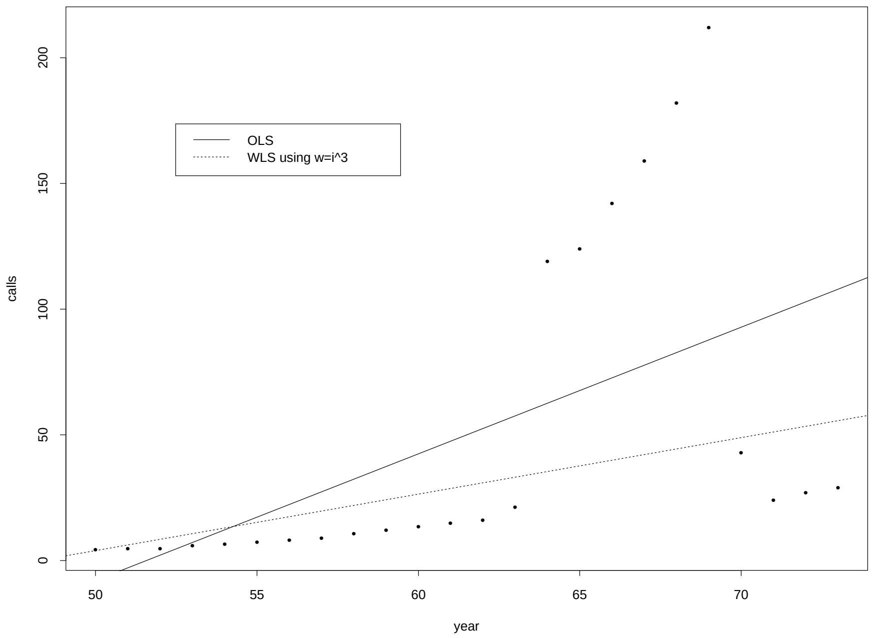
$$RSS_{\underline{w}} = \sum_{i=1}^{24} w_i \cdot (y_i - \underline{x}_i\underline{\beta})^2 = \sum_{i=1}^{14}(y_i - \underline{x}_i\underline{\beta})^2 + \sum_{i=21}^{24}(y_i - \underline{x}_i\underline{\beta})^2,$$

completely ignoring observations 15–20.

Slightly Silly Idea #2

Instead of omitting the outliers completely, we might try to "emphasize them less". It turns out the observations are recorded in order in the data set, so that observation 1 is from 1950, observation 2 is from 1951, etc. So, for example, if we weight observation $\#i$ by something like $1/i^3$, then the later observations, where the outliers are, will receive less weight than the earlier ones, and we should get a "better" regression line.

```
> weights _ 1/(1:24)^3
> plot(year,calls)
> abline(lm(calls~year))
> abline(lm(calls~year,weights=weights),lty=2)
> legend(locator(),c("OLS","WLS using w=i^3"),lty=1:2)
```



In this case we have minimized

$$RSS_{\underline{w}} = \sum_{i=1}^{24} w_i \cdot (y_i - \underline{x}_i\underline{\beta})^2 = \sum_{i=1}^{24} \frac{(y_i - \underline{x}_i\underline{\beta})^2}{i^3}$$

so that the weights applied to observations 15–20 are $1/15^3 = 0.0002962963$, $1/16^3 = 0.0002441406$, etc. However, also the weights applied to the last four observations are very tiny.

Slightly Silly Idea #3

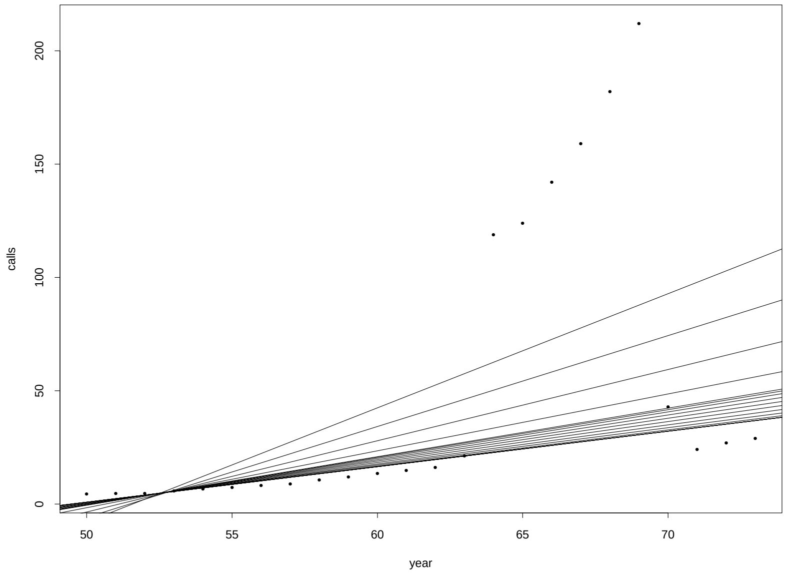
A problem with pre-specifying the weights is that you sort of have to work as hard to specify the weights as you would to just delete the outliers in the first place. A better idea would be to let the data specify the weights.

One idea for this would be to downweight the each observation according to the size of its residual, i.e. $w_i = 1/|y_i - \hat{y}_i|$.

In the plot below we have done this not one time, but 20 times, each time taking the residuals from the last WLS regression and using them to generate the weights for the next regression.

The result is pictured below; the main thing to see from the picture is that this iterative WLS procedure eventually seems to settle on a regression line that seems to be "pulled" by the outliers only a little bit.

```
> plot(year,calls)
> weights _ rep(1,24)
> for (i in 1:20) {
+ abline(fit _ lm(calls~year,weights=weights))
+ weights _ 1/abs(residuals(fit))
+ }
```



In the first iteration, with $w_i \equiv 1$, we are just minimizing

$$RSS_{\underline{w}} = \sum_{i=1}^{24} w_i \cdot (y_i - \underline{x}_i \underline{\beta})^2 = \sum_{i=1}^{24} (y_i - \underline{x}_i \underline{\beta})^2$$

i.e. it is the OLS fit.

In the second iteration, with $w_i = 1/|y_i - \hat{y}_i|$, we are minimizing

$$RSS_{\underline{w}} = \sum_{i=1}^{24} w_i \cdot (y_i - \underline{x}_i\underline{\beta})^2 = \sum_{i=1}^{24} (y_i - \underline{x}_i\underline{\beta})^2 / |y_i - \underline{x}_i\hat{\underline{\beta}}_i^{OLS}| \approx \sum_{i=1}^{24} |y_i - \underline{x}_i\underline{\beta}|$$

This is approximately the "L1 fit" or "least absolute deviations fit", i.e. it is similar to minimizing the sum of absolute residuals.

In each successive iteration, we are using weights proportional to the inverses of the previous iteration's residuals. This is an example of *iteratively re-weighted least squares* (IRLS, or sometimes IWLS). As this continues, the outliers become even more outlying, and so the weights for them get smaller. Finally after 20 iterations, the weights are

```
> weights # after 20 iterations...
        1         2        3        4       5         6         7         8
 0.2802939 0.4340796 1.351778 2.660932 2.04855 0.7395855 0.4725748 0.3355676
        9        10        11        12        13       14         15
 0.3644336 0.3438858 0.336485 0.3188919 0.2857292 27.60578 0.01038721
         16          17          18          19          20         21
 0.01002925 0.008609975 0.007599917 0.006535246 0.005511085 0.09183988
        22        23        24
 0.1033544 0.121368 0.1281499
```

and we can see that the outliers, observations 15–20, have very low weights compared to the other observations. *It is this idea of "downweighting large residuals" that is at the heart of robust regression.*

It turns out that as these iterations continue, the regression line is exactly the L1 fit line, that minimizes the residual sum of absolute deviations

$$RSAD = \sum_{i=1}^{24} |y_i - \underline{x}_i\underline{\beta}|$$

(There is a function in Splus, `l1fit()`, that will do the L1 fit directly, so you wouldn't have to do these iterations yourself in general).

Aside.

What about those outliers in the Belgian phone calls data?

It turns out that in the years 1964–1969, the <u>*number of minutes*</u> was recorded instead of the number of phone calls! In 1963 and 1970, <u>*a combination*</u> of number of minutes and number of phone calls was recorded.

AAAUUUUGGGGHHHHHHH!

End of Aside.

Terminology

- *Robust:* Performs well under small violations of assumptions.
- *Resistant:* Not changed much when there are small changes in the sample data (small changes in many cases or large changes in a few cases). Usually implies robustness.
- OLS is inefficient for violated assumptions, e.g. heavy tailed errors (not robust). Slopes can change large amounts if a single data point is moved (not resistant).

Main Ideas of Robust Regression

- Eliminating outliers (or using a single-case dummy variable) is an all-or-nothing proposition. Robust regression *down-weights* outliers according to one of several possible approaches.
- When all regression assumptions are met, ordinary least squares performs better than robust regression. Robust regression often performs better for heavy-tailed distributions, e.g. in the presence of outliers.

Goals of Robust Regression

- Consistency and reasonable efficiency when the assumed model is true
- Only slightly impaired by small departures from the assumptions
- Not drastically affected by larger departures from the assumptions

How it Works

- OLS minimizes
$$\sum_i (y_i - \underline{x}_i \underline{\beta})^2$$
  Robust estimators minimize
$$\sum_i \rho\left(\frac{y_i - \underline{x}_i \underline{\beta}}{s}\right)$$
  where
  - $\rho(u)$ is chosen to "downweight" large residuals;
  - s is a resistant estimator of the residual SE.

- We work with "scaled residuals"
$$u_i = e_i/s = (y_i - \underline{x}_i\underline{\beta})/s$$
instead of raw residuals, so that the function ρ can be the same for all problems. (This scaling factors out of OLS).

Resistant estimate of s: Median absolute deviation is
$$\text{MAD} = \text{median}|e_i - \text{median}(e_i)|$$
which is a resistant measure of spread. A resistant estimate of the residual standard error is $s = \text{MAD}/0.6745$.

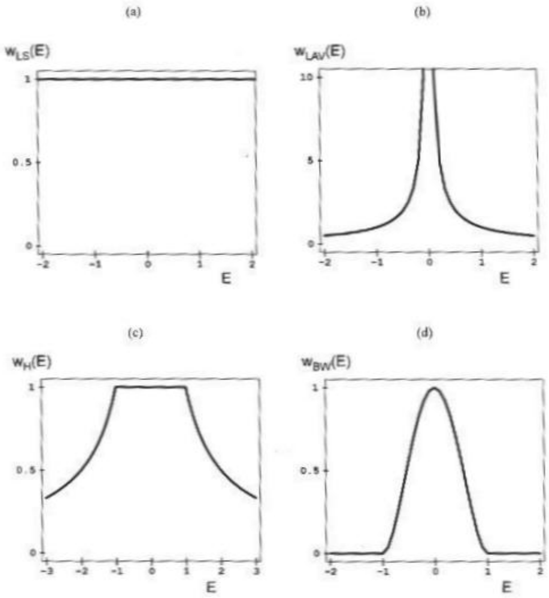
- Robust estimators are examples of *M-estiators*, they find the $\underline{\hat{\beta}}$'s that minimize sums of functions:
$$S = \sum_i \rho(u_i) = \sum_i \rho(y_i - \underline{x}_i\underline{\beta})$$
Differentiating and setting to zero, this is the same as finding the $\underline{\hat{\beta}}$ that solves the set of equations
$$\frac{\partial S}{\partial \beta_k} = \sum_i (\psi(u_i)/u_i)\, u_i\, x_{ik} = 0 \qquad (*)$$
where $\psi(u) = \rho'(u)$. Note that $(*)$ has exactly the same form as the normal equations for WLS
$$\frac{\partial RSS_{\underline{w}}}{\partial \beta_k} = \frac{\partial}{\partial \beta_k} \sum_i w_i (y_i - \underline{x}_i\underline{\beta})^2 = \sum_i 2w_i\, u_i\, x_{ik} = 0$$
with weights
$$w_i = \psi(u_i)/u_i$$
This leads to the following *iteratively re-weighted least squares (IRLS) procedure*:

1. Obtain starting values for the coefficients, e.g. from OLS.
2. Calculate scaled residuals u_i. Use the residuals to calculate weights.
3. Apply WLS using the most recent set of weights to get new coefficient estimates.
4. Go to step b if things are still changing.

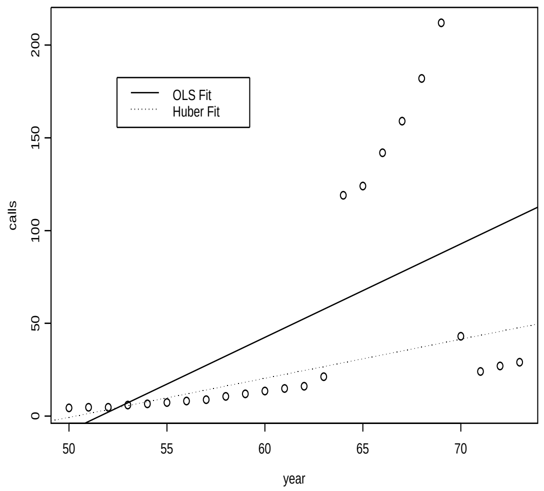
This is exactly how we found the L1 fit line earlier.

- Only thing that remains is to pick the function $\rho(u)$. Here are several choices:

Method	OLS	L1	Huber	Bisquare
$\rho(u)$	u^2	$\lvert u\rvert$	u^2 if $\lvert u\rvert \le c$ [1] $c(2\lvert u\rvert - c)$ if $\lvert u\rvert > c$	$(c^2/3)(1-[1-(u/c)^2]^3)$ if $\lvert u\rvert \le c$ [2] $c^2/3$ if $\lvert u\rvert > c$
$w = \psi(u)/u$	1 [3]	$1/\lvert u\rvert$	??? if $\lvert u\rvert \le c$??? if $\lvert u\rvert > c$	$[1-(u/c)^2]^2$ if $\lvert u\rvert \le c$ 0 if $\lvert u\rvert > c$



The weight functions $w = \psi(u)/u$ for each regression method.



The OLS fit and the robust fit using Huber's function $\rho(u)$.

```
> library(MASS) # need once per session, to use ‘‘rlm’’ below
> plot(year,calls)
> abline(lm(calls~year))
> abline(rlm(calls~years),lty=2)
> legend(locator(),c("OLS fit","Huber fit"),lty=1:2)
```

[1] c is a tuning parameter usually set equal to 1.345 for the Huber function.
[2] c is a tuning parameter usually set equal to 4.685 for the Bisquare function.
[3] Ignoring any multiplicative constants...

Breakout Session

1. For OLS, $\rho(u_i) = u_i^2$. Calculate $\psi(u_i) = \rho'(u_i)$ and $w_i = \psi(u_i)/u_i$.

2. For Huber estimation,

$$\begin{array}{ll} \rho(u_i) = u_i^2 & \text{if} \quad |u_i| \leq c \\ \rho(u_i) = c(2|u_i| - c) & \text{if} \quad |u_i| > c \end{array}$$

(Assume c=1.345.) Calculate $\psi(u_i)$ and w_i. Intuitively, how will the regression coefficients change (from OLS) if we re-run the regression with these weights?

3. Interpret the sketches of the weight function vs. the residual for the above two weight functions (see plots on last page of handout).

4. Consider the data x=1, 2, 3, 4, 5, and y=7.384, 14.265, 11.490, 12.580, 15.442. Plot the data including your best guess for the regression line. Interpret the following where wn, rn and bn are the weights, the scaled residuals, and the coefficients for the n^{th} iteration of a robust regression based on the Huber weight function.

```
w1    r1   w2    r2   w3    r3   w4    r4
 1 -1.84 0.73 -1.67 0.80 -1.16 1.00 -0.97
 1  3.26 0.41  5.66 0.24  7.42 0.18  9.90
 1 -0.70 1.00 -0.46 1.00 -0.16 1.00  0.03
 1 -1.03 1.00 -1.19 1.00 -1.27 1.00 -1.54
 1  0.30 1.00  0.54 1.00  0.59 1.00  0.70
```

```
   (Intercept)    x
b0        7.90 1.44
b1        7.90 1.44
b2        6.97 1.62
b3        6.32 1.75
b4        6.02 1.82
```

Turn-in Sheet

My Name: ____________________________________

My Group: ____________________________________

____

____

1. For OLS, $\rho(u_i) = u_i^2$. Calculate $\psi(u_i)$ and w_i.

2. For Huber estimation,

$$\begin{array}{ll} \rho(u_i) = u_i^2 & \text{if} \quad |u_i| \le c \\ \rho(u_i) = c(2|u_i| - c) & \text{if} \quad |u_i| > c \end{array}$$

(Assume c=1.345.) Calculate $\psi(u_i)$ and w_i. Intuitively, how will the regression coefficients change (from OLS) if we re-run the regression with these weights?

3. Interpret the sketches of the weight function vs. the residual for the above two weight functions (see plots on last page of handout).

4. Consider the data x=1, 2, 3, 4, 5, and y=7.384, 14.265, 11.490, 12.580, 15.442. Plot the data including your best guess for the regression line. Interpret the following where wn, rn and bn are the weights, the scaled residuals, and the coefficients for the n^{th} iteration of a robust regression based on the Huber weight function.

w1	r1	w2	r2	w3	r3	w4	r4
1	-1.84	0.73	-1.67	0.80	-1.16	1.00	-0.97
1	3.26	0.41	5.66	0.24	7.42	0.18	9.90
1	-0.70	1.00	-0.46	1.00	-0.16	1.00	0.03
1	-1.03	1.00	-1.19	1.00	-1.27	1.00	-1.54
1	0.30	1.00	0.54	1.00	0.59	1.00	0.70

	(Intercept)	x
b0	7.90	1.44
b1	7.90	1.44
b2	6.97	1.62
b3	6.32	1.75
b4	6.02	1.82