Exploratory Data Analysis

Transforms and Robust Regression

David B King, Ph.D.

September 21, 2015

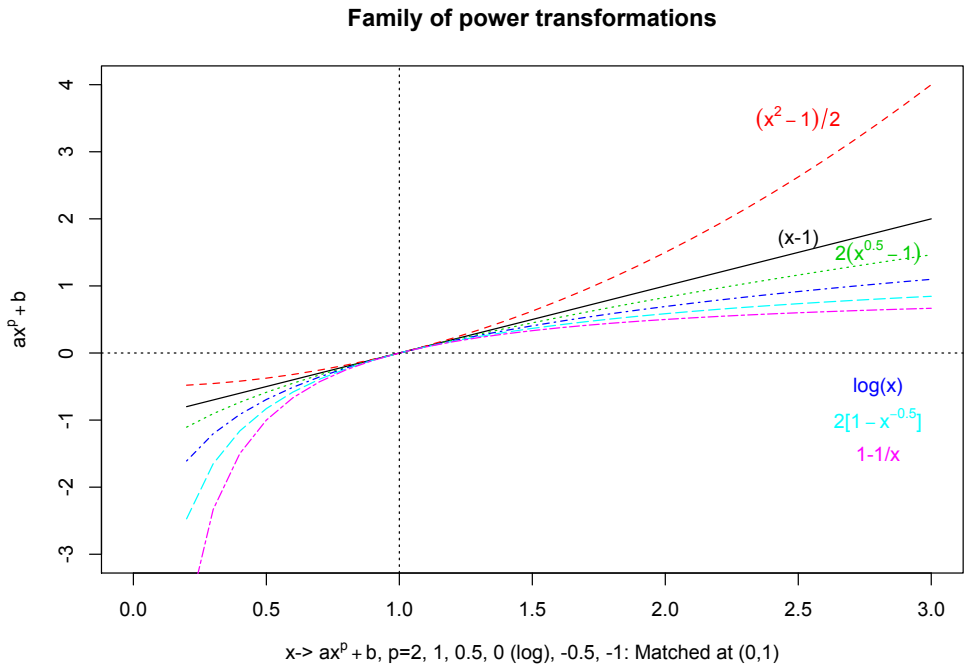
Recap from Last Lecture

In the last lecture we learned about some of the serendipitous effects of transformation.

- To transform so data has uniform spread (homoscedasticity) **across groups** we:
  1. Plot $\log(d_F)$ versus $\log(M)$ from batch to bath
  2. Fit a straight line through data of the form $\log(d_F) = a + b\log(M)$
  3. Estimate b and set power $p = 1 - b$ in transform $T_p(x)$.
- To transform so data has a more symmetric shape **within each group** we:
  1. Plot $\log(d_F)$ versus $\log(M)$ from batch to bath
  2. Fit a straight line through data of the form $\log(d_F) = a + b\log(M)$
  3. Estimate b and set power $p = 1 - b$ in transform $T_p(x)$.

Tukey's Ladder Family of Transforms

$$T_p(x) = \begin{cases} \frac{x^p - 1}{p} & \text{for } p \neq 0 \\ \ln(x) & \text{for } p = 0 \end{cases}$$



$T_p(x)$ all look about the same near middle $(1, 0)$.

Tukey's Family of Transforms

Notice that

$$T_p'(x) = \begin{cases} x^{p-1} & \text{for } p \neq 0 \\ \frac{1}{x} & \text{for } p = 0 \end{cases}$$

so the transforms satisfy $T_p'(1) = 1$ and are "locally linear" $(\propto x)$ in the Taylor series expansion around $x = 1$.
The log is used for $p = 0$ since by L'Hospital's Rule

$$\lim_{p \to 0} \frac{x^p - 1}{p} = \lim_{p \to 0} \frac{e^{p \log(x)} - 1}{p} = \lim_{p \to 0} \frac{\log(x) e^{p \log(x)}}{1} = \log(x)$$

Tukey's Family of Transforms

From page 251, Ch 8 of UREDA text "As p changes in value, the resulting change from one member of the family to another occurs in a smooth and continuous way".
From Calculus, the curvature of a twice differentiable function $f(x)$ is given by

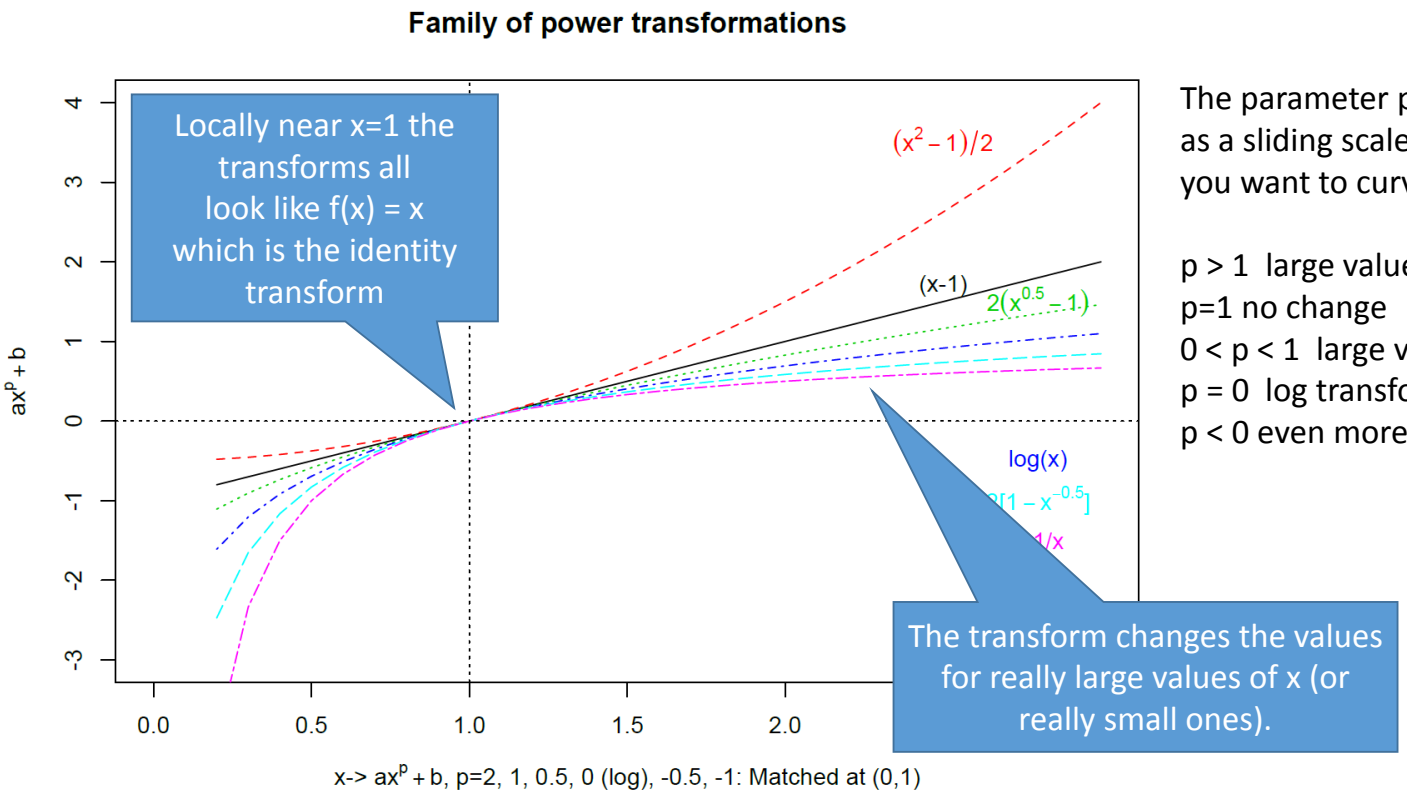
$$K(x) = \frac{|f''(x)|}{\{1 + [f'(x)]^2\}^{3/2}},$$

and so the curvature at x for $T_p(x)$ is given by

$$K(x) = \frac{|(p-1)x^{p-2}|}{\{1 + x^{2p-2}\}^{3/2}}.$$

Thus when $x = 1 \Longrightarrow K(1) = \frac{|1-p|}{2^{3/2}} \propto 1 - p = b$ our slope!!

Logic Behind Family of Transforms



The parameter p can be viewed as a sliding scale of how aggressive you want to curve the data values:

p > 1 large values get larger "push out"
p=1 no change
0 < p < 1 large values get "pulled in"
p = 0 log transform (pretty aggressive)
p < 0 even more aggressive.

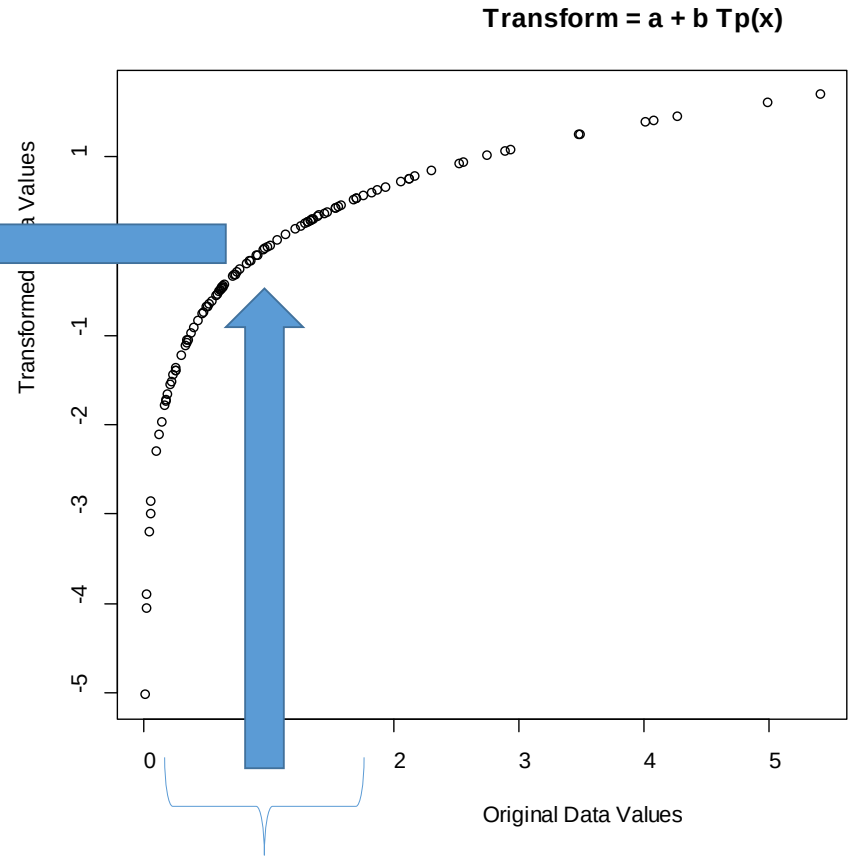
Matching Values Near Median

Let us suppose taht we have decided to transform our data using $T_p(x)$. Tukey then suggests we further transform to $a + bT_p(X)$ so that values are matched near median M.
The benefits of matching values are:

- First of all, in going from $T(x)$ to $a + bT(x)$, there presents little additional difficulty in interpretation.
- Can arrange for the re-expressed data to look like the original data, except the extreme values
- Emphasizes the changes that are due to transformation
- Helps to compare the effect of different transformations

What is Matching?

Matching means values
a + b T(x)
have nearly the
same value locally
near median



Philosophy
As long as we've decided
to apply $T_p(x)$ to our data

Let's find constants a and b
so that the values of Y = T(X)
are nearly unchanged or matched
to the original data

Illustration of Matching

Tukey's example using population of US cities example

Table: Population of large US cities in three scales

City	Population	$\log_{10}$(Population)	$200 \log_{10}$(Population) $- 300$
New York	778	2.89	278.2
Chicago	355	2.55	210
Los Angeles	248	2.39	178.9
Philadelphia	200	2.30	160.2
Detroit	167	2.22	144.5
Baltimore	94	1.97	94.6
Houston	94	1.97	94.6
Cleveland	88	1.94	88.9
Washington	76	1.88	76.2
St Louis	75	1.88	75.0
Milwaukee	74	1.87	73.8
San Francisco	74	1.87	73.8
Boston	70	1.85	69.0
Dallas	68	1.83	66.5
New Orleans	63	1.80	59.9

Notice for the values after the break in the table above, the transformed values are nearly equal to numbers before transform

How to Match Transformations

There are two possible ways for obtaining matched transformation

- Original data x, nonlinear transformation $y = T(x)$, further transformation $z = a + by = a + bT(x)$.
- Objective: choose a and b to match z to x.
- Method 1: choose x_1 and x_2 in original scale, find a and b

$$\begin{cases} z_1 = a + bT(x_1) = x_1 \\ z_2 = a + bT(x_2) = x_2 \end{cases}$$

How to Match Transformations

Method 2: choose $x_0 = M$ (or some central value),
$z_0 = a + bT(x_0) = x_0$

$$\frac{dz}{dx}\Big|_{x_0} = b\frac{dT(x)}{dx}\Big|_{x_0} = 1 \Rightarrow \boxed{b = \frac{1}{T'(x_0)}}$$

$$\Rightarrow \boxed{a = x_0 - \frac{T(x_0)}{T'(x_0)}} \Longrightarrow \boxed{z = x_0 + \frac{T(x) - T(x_0)}{T'(x_0)}}$$

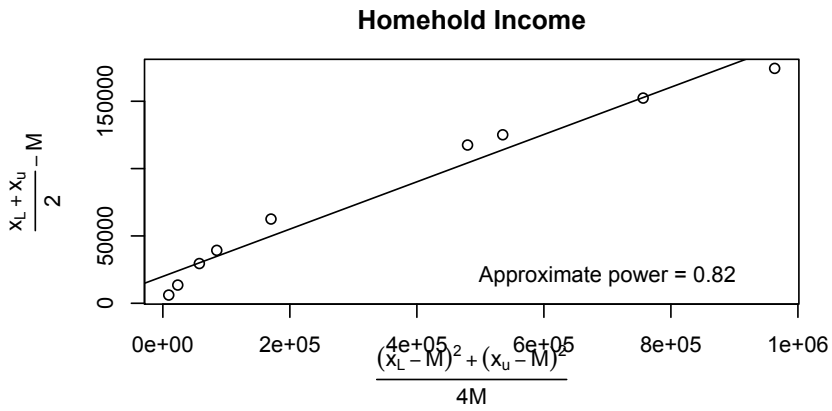
- If $T(x) = x^p$ then $T'(x_0) = px_0^{p-1}$ so the matched transform is $\boxed{z = x_0 + (x^p - x_0^p)/px_0^{p-1}}$
- For $p = 0$ we have $T(x) = \log(x)$ and $T'(x) = 1/x$, so the matched transform is $\boxed{z = x_0(1 + \log(x) - \log(x_0))}$

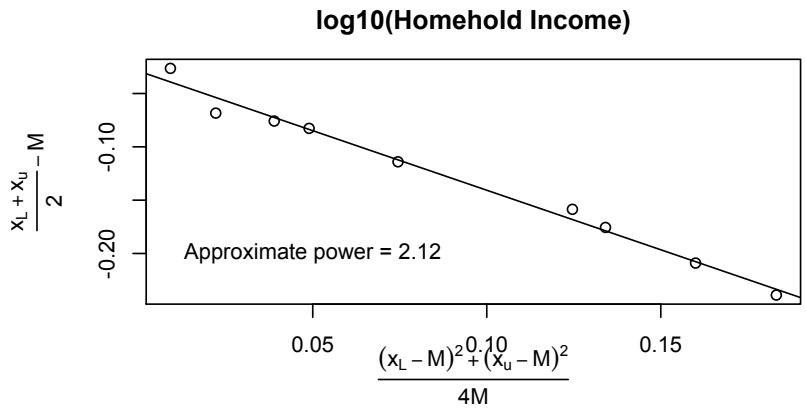
Example: Household Income

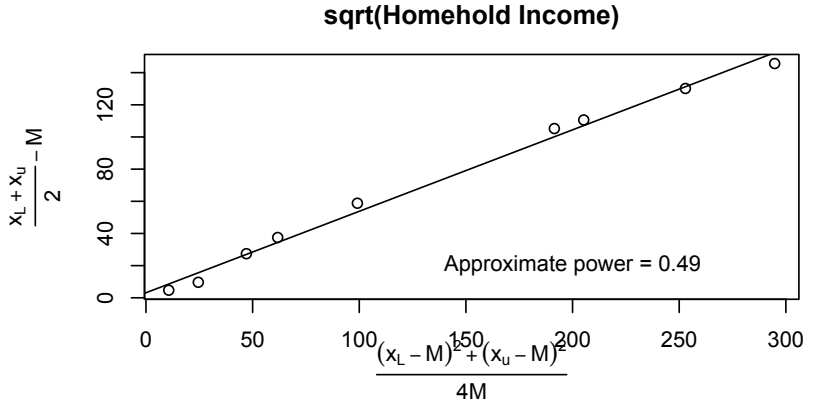
Letter value display of original Household Income data set

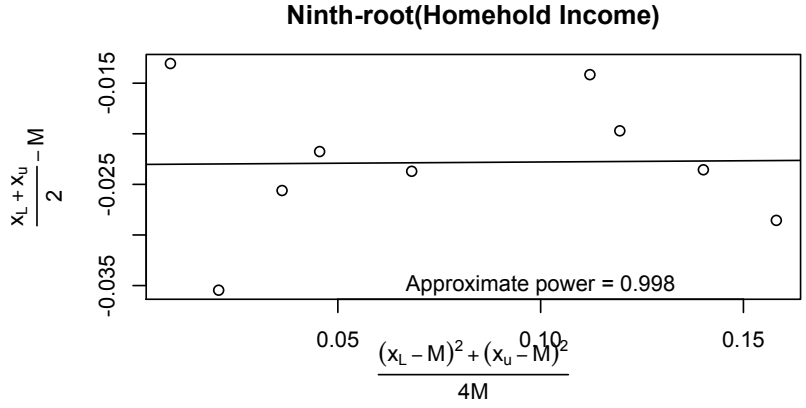
	Depth	Lower	Upper	Mid	Spread	pseudo-s
M	195.0	39200	39200	39200	0	0.00
F	98.0	19000	71600	45300	52600	38992.44
E	49.5	12000	93450	52725	81450	35402.29
D	25.0	8400	129000	68700	120600	39305.91
C	13.0	7000	150000	78500	143000	38384.48
B	7.0	4570	198900	101735	194330	45111.72
A	4.0	2380	311000	156690	308620	63828.85
Z	2.5	2100	326500	164300	324400	60975.90
Y	1.5	1600	381500	191550	379900	65826.07
X	1.0	1200	426000	213600	424800	68576.54

Example: Household Income



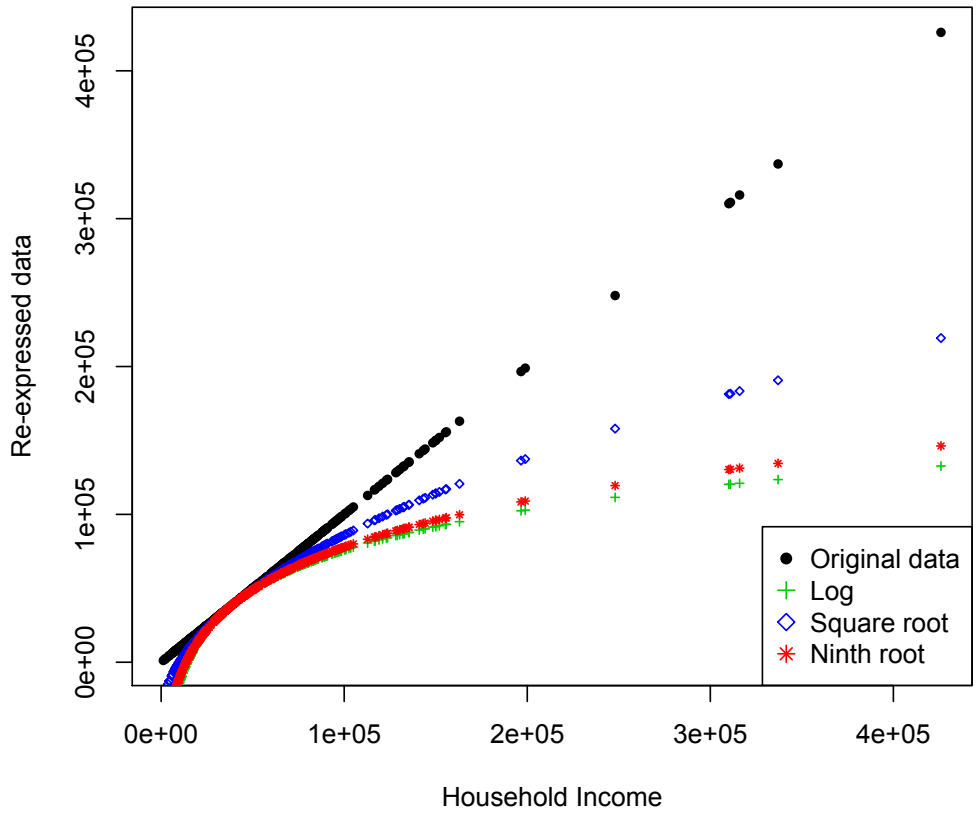






Example: Household Income

Re-expressed household income data versus original data



Iterative Refinement

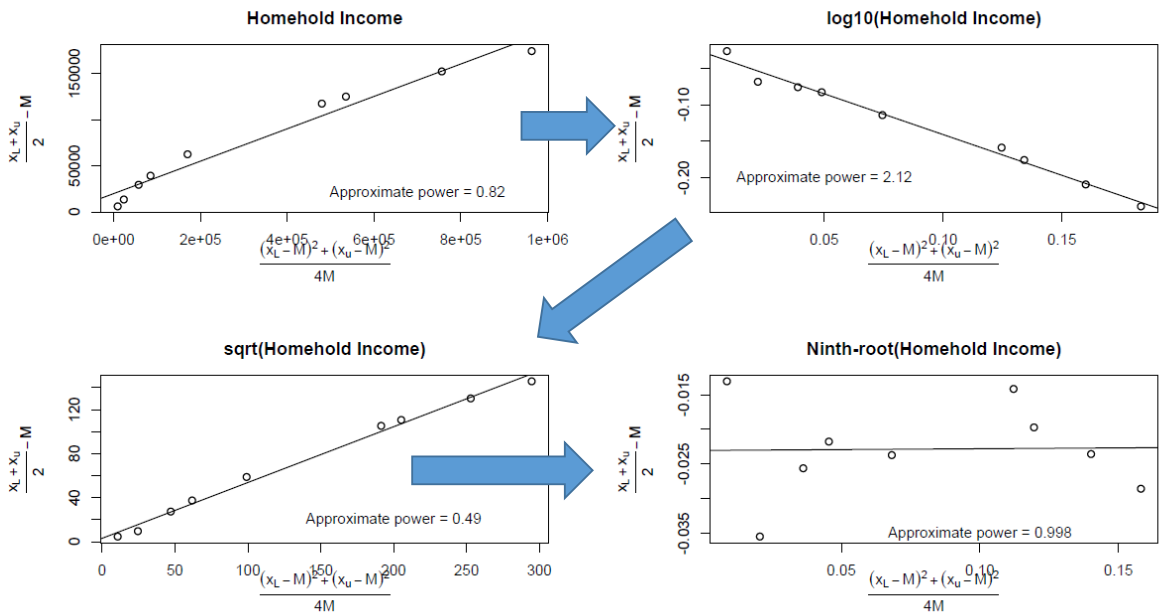
Iterative Refinement

Step 1: Plotted original data and it suggested p=0 so then I took log(data).

Step 2: Plotted log(data) and slope <0 ➔ so then I took sqrt(data).

Step 3: Plotted sqrt(data) and slope > 0 ➔ so I took the ninth root(data)

Step 4: Plotted ninth root(data) and slope = 0 ➔ I can stop iterating (found the sweet spot).



Example: Household Income

Matched: log10(Household Income) mid summaries decreasing

	Depth	Lower	Upper	Mid	Spread	pseudo-s
M	195.0	39200.000	39200.00	39200.00	0.00	0.00
F	98.0	10810.240	62814.50	36812.37	52004.26	38550.81
E	49.5	-7203.198	73254.42	33025.61	80457.62	34970.95
D	25.0	-21184.678	85891.92	32353.62	107076.60	34898.37
C	13.0	-28331.593	91804.11	31736.26	120135.70	32247.18
B	7.0	-45046.140	102864.91	28909.38	147911.05	34336.04
A	4.0	-70620.394	120386.72	24883.16	191007.12	39504.13
Z	2.5	-75571.221	122272.99	23350.88	197844.21	37187.82
Y	1.5	-87451.332	128127.38	20338.02	215578.71	37353.77
X	1.0	-97463.387	132720.71	17628.66	230184.09	37159.20

Original data $F_L = 19000$, $F_U = 71600$ and $M = 39200$

Example: Household Income

Matched: Square-root (Household Income) mid summaries increasing

	Depth	Lower	Upper	Mid	Spread	pseudo-s
M	195.0	39200.000	39200.00	39200.00	0.00	0.00
F	98.0	15382.048	66756.97	41069.51	51374.92	38084.29
E	49.5	4177.413	81849.30	43013.36	77671.89	33760.13
D	25.0	-2907.852	103022.36	50057.25	105930.21	34524.74
C	13.0	-6069.953	114162.32	54046.18	120232.27	32273.10
B	7.0	-12431.063	137399.89	62484.41	149830.95	34781.72
A	4.0	-19882.029	181627.53	80872.75	201509.56	41676.25
Z	2.5	-21059.073	187034.30	82987.61	208093.38	39114.30
Y	1.5	-23487.051	204961.89	90737.42	228448.94	39583.83
X	1.0	-25482.857	219250.77	96883.96	244733.63	39507.97

Original data $F_L = 19000$, $F_U = 71600$ and $M = 39200$

Example: Household Income

Matched: Ninth-root (Household Income) mid summaries constant

	Depth	Lower	Upper	Mid	Spread	pseudo-s
M	195.0	39200.000	39200.00	39200.00	0.00	0.00
F	98.0	11922.135	63623.06	37772.60	51700.93	38325.96
E	49.5	-4281.555	74952.67	35335.56	79234.22	34439.20
D	25.0	-16300.287	89123.30	36411.51	105423.59	34359.62
C	13.0	-22262.376	95929.05	36833.33	118191.42	31725.29
B	7.0	-35743.338	108971.94	36614.30	144715.28	33594.17
A	4.0	-55172.530	130489.16	37658.31	185661.69	38398.58
Z	2.5	-58770.178	132872.75	37051.28	191642.92	36022.19
Y	1.5	-67112.727	140379.03	36633.15	207491.76	35952.53
X	1.0	-74105.962	146289.41	36091.73	220395.38	35578.98

Original data $F_L = 19000$, $F_U = 71600$ and $M = 39200$

Example: Household Income Matched Transform

Now we've found the sweet spot transform for the data by choosing $p = 1/9 \rightarrow T_p(x) = \frac{x^{1/9}-1}{1/9} = 9(x^{1/9} - 1)$. Now we need to further refine our transform data using

$$a + bT_p(x) = a + 9b(x^{1/9} - 1)$$

for some constants a and b.
Now let's match the values around the median by taking $x_0 = M = 39200$. According to notes we should choose

$$b = \frac{1}{T'(x_0)} = x_0^{8/9} = 12103.78 \text{ and,}$$

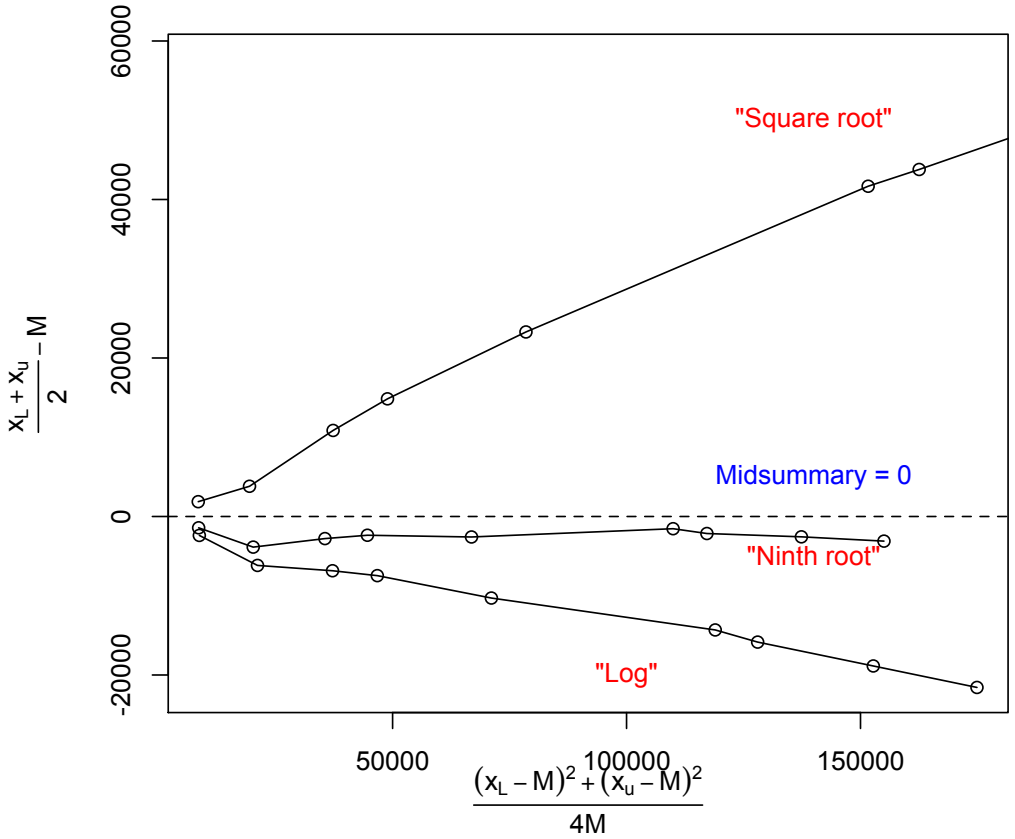
$$a = x_0 - \frac{T(x_0)}{T'(x_0)} = 39200 - 12103.78(T(x_0)) = 39200 - (12103.78)(20.15) = -204691.2$$

Thus the final matched transform is

$$\boxed{z = x_0 + (x^p - x_0^p)/px_0^{p-1} = 39200 + 108934(x^{1/9} - 3.24) = 108934(x^{1/9}) - 313746.2}$$

Example: Household Income

Matched midsummaries for the household income data:



Matched Transformations effect on Expected Value

From what we discussed if we use the matching algorithm the expected values do not match since

$$\boxed{\frac{\sum T(X_i)}{n} \neq T(\bar{x}), \quad E(T(x)) \neq T(EX)}$$

Matching can reduce the difference however, because under our matching rule

$$\mathrm{E}(Z) = a + b\mathrm{E}[T(X)] = x_0 + \frac{\mathrm{E}[T(X)] - T(x_0)}{T'(x_0)}$$

Now using the Taylor series expansion approximation

$$T(X) - T(x_0) \approx T'(x_0)(X - x_0) + \frac{1}{2}T''(x_0)(X - x_0)^2,$$

we see that

$$\begin{aligned}\mathrm{E}[Z] &\approx x_0 + (\mathrm{E}[X] - x_0) + \frac{1}{2}\mathrm{E}[(X - x_0)^2]\frac{T''(x_0)}{T'(x_0)} \\ &= \mathrm{E}(X) + \frac{1}{2}\mathrm{E}[(X - x_0)^2]\frac{T''(x_0)}{T'(x_0)}\end{aligned}$$

Matched transformations – Interpretation:

We can write
$E(X - x_0)^2 = E(X - EX)^2 + (EX - x_0)^2 = var(X) + (EX - x_0)^2$.

Hence, if we choose x_0 to be near the mean of X

$$\boxed{\mathrm{E}(Z) \approx \mathrm{E}(X) + \frac{1}{2} var(X) \frac{T''(x_0)}{T'(x_0)}}$$

If Z is a transformation of X matched at a point near the mean of X, the mean of Z is approximately equal to the mean of X plus a term that depends on the spread of X and the curvature of the transformation local to the median M.

Household Income Example

From before, in the household income data set we settled upon the transform

$$Z = x_0 + (X^p - x_0^p)/px_0^{p-1} = 39200 + 108934(X^{1/9} - 3.24) = 108934(X^{1/9}) - 313746.2$$

Hence, if we choose $x_0 = M$

$$T^{'}(x_0) = 12104x_0^{-8/9} = 1 \text{ and } T^{''}(x_0) = -10758.92x_0^{-17/9} = -2.267E-05$$

So according to the theory the new mean of the transformed data should be approximately

$$\mathrm{E}(Z) \approx \mathrm{E}(X) + \frac{1}{2}\mathrm{Var}(X)\frac{T''(x_0)}{T'(x_0)} = \mathrm{E}(X) - 0.0000113 * \mathrm{Var}(X)$$

Serendipitous effects of transformation

1. Data that are amounts or counts usually display both increasing spread with increasing level and right-skewness
   - Bounded below by zero: variation of spread with level; right-skewness
   - Poisson variable, mean λ, variance λ
2. Transformation for stable spread will necessarily compress the scale more for larger values than smaller values
3. Transformation for symmetry will also compress the scale more for larger values than for smaller values
4. Thus transformation for either spread or symmetry will usually help us towards both objective simultaneously.

When is transformation worthwhile?

1. Range consideration: Amounts or counts, when the range of batch is relatively large.

   Rule of thumb: transformation helpful when the ratio

   $$\frac{\text{largest data value}}{\text{smallest data value}} > 20$$

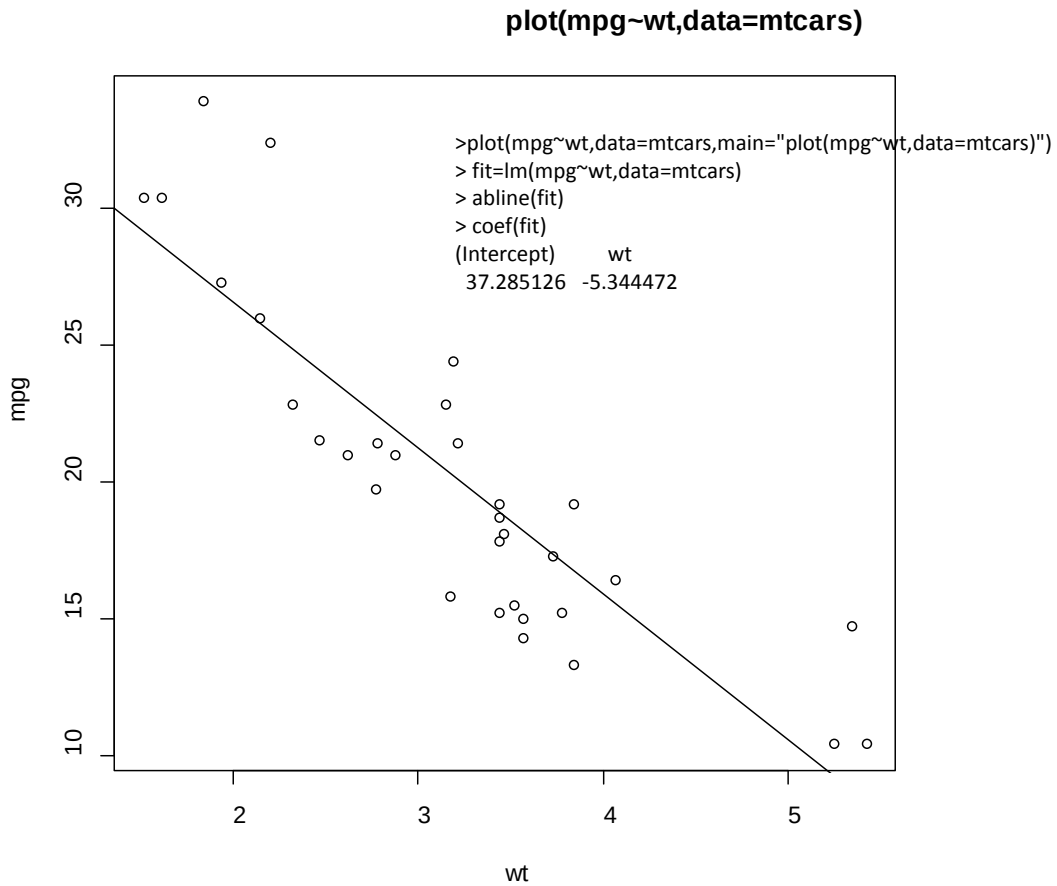
   not helpful if the ratio is smaller than 2
2. Looking at residuals (y-versus-x in Ch5, two-way table in Ch6)
3. Using transformation plots: slope ≈ 0, no transformation; if plot nearly linear & slope $\neq 0$, power; if nonlinear, complex transformation.
   - spread vs level plots
   - transformation plot for symmetry
   - additional one in Ch6
4. Trial and error

Transforming for Straightness

Advantage of linear relationship

- Easier interpretation
- Departure from fit are more easily detected
- Easy interpolation and extrapolation

Transformation for Straightness



Want to find a transform of Y which "straightens" or makes the Y vs X plot look linear.

Transforming for Straightness

Rules for Transforming for Straightness Ch 8 pp 264–267 of UREDA.

- Let $\{(x_i, y_i)\}$ be the paired observations between two variables x and y.
- Suppose M_x and M_y are the medians for x and y and let C be the slope of a resistant line.
- Transformation Rule is:
  1. Use $Y - M_y - C(X - M_x)$ as the vertical coordinates and
  2. use $\frac{C^2(X-M_x)^2}{2M_x}$ as the horizontal coordinates in a plot
  3. If the resulting graph is nearly linear with slope b then set $p = 1 - b$ and then
  4. Transform Y using $T_p(Y)$.

Why Does this Rule Work

We seek a power transformation $\phi(x)$ such that the points

$$[x_1, \phi(y_1)], [x_2, \phi(y_2)], \ldots, [x_n, \phi(y_n)]$$

fall approximately on a straight line. Suppose that the **true model** for the data were

$$y^p - M_y^p = K(X - M_x) \tag{1}$$

Now if $Z = Y^p = \phi(Y)$ then by using a Taylor series expansion of $Z^{1/p}$ around $M_z = M_y^{1/p}$ we have

$$Y = Z^{1/p} \approx M_z^{1/p} + \frac{1}{p} M_z^{(1-p)/p}(Z - M_z) + \frac{(1-p)}{2p^2} M_z^{(1-2p)/p}(Z - M_z)^2$$

But then plugging in (1) for the true model we see that

$$Y \approx M_y + \frac{KM_y}{pM_z}(X - M_x) + \frac{(1-p)}{2M_y}\left(\frac{KM_y}{pM_z}\right)^2 (X - M_x)^2$$

Why Does this Rule Work

If we replace $C = \frac{KM_y}{pM_z}$ then the equation becomes

$$Y \approx M_y + C(X - M_x) + \frac{(1-p)}{2M_y} C^2 (X - M_x)^2.$$

The numerical estimate for C can be roughly obtained by fitting the linear equation $Y - M_y = C(X - M_x)$ to raw data. We then have

$$Y - M_y - C(X - M_x) \approx (1-p) \frac{C^2 (X - M_x)^2}{2M_y},$$

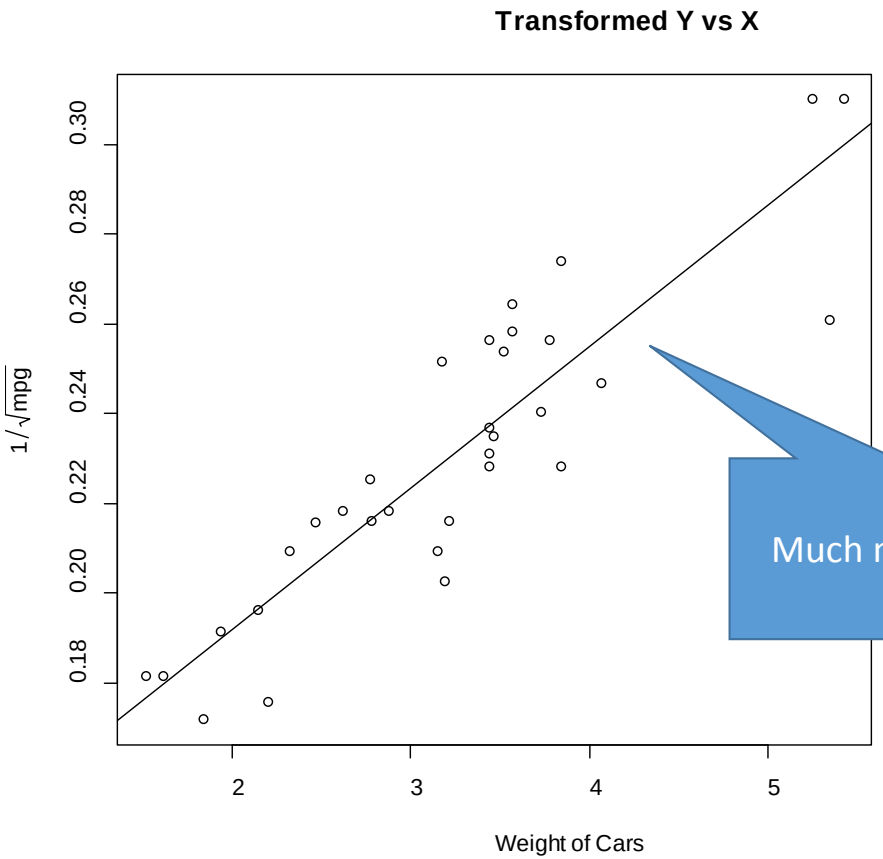
where only $(1 - p)$ is unknown. Notice this is the slope again!!

Mtcars Example

We see from the plot that the slope $C \approx -5.344$.

```
attach(mtcars)
y=mpg
x=wt
yy = y - median(y) + 5.344472*(x-median(x))
xx=(5.344472^2*(x-median(x))^2)/(2*median(y))
plot(xx,yy)
fit1=lm(yy~xx)
abline(fit1)
coef(fit1)
(Intercept)          xx
 -0.7586727   1.5367290
power = 1 - 1.5367
power
[1] -0.5367  #take 1/sqrt(y)
```

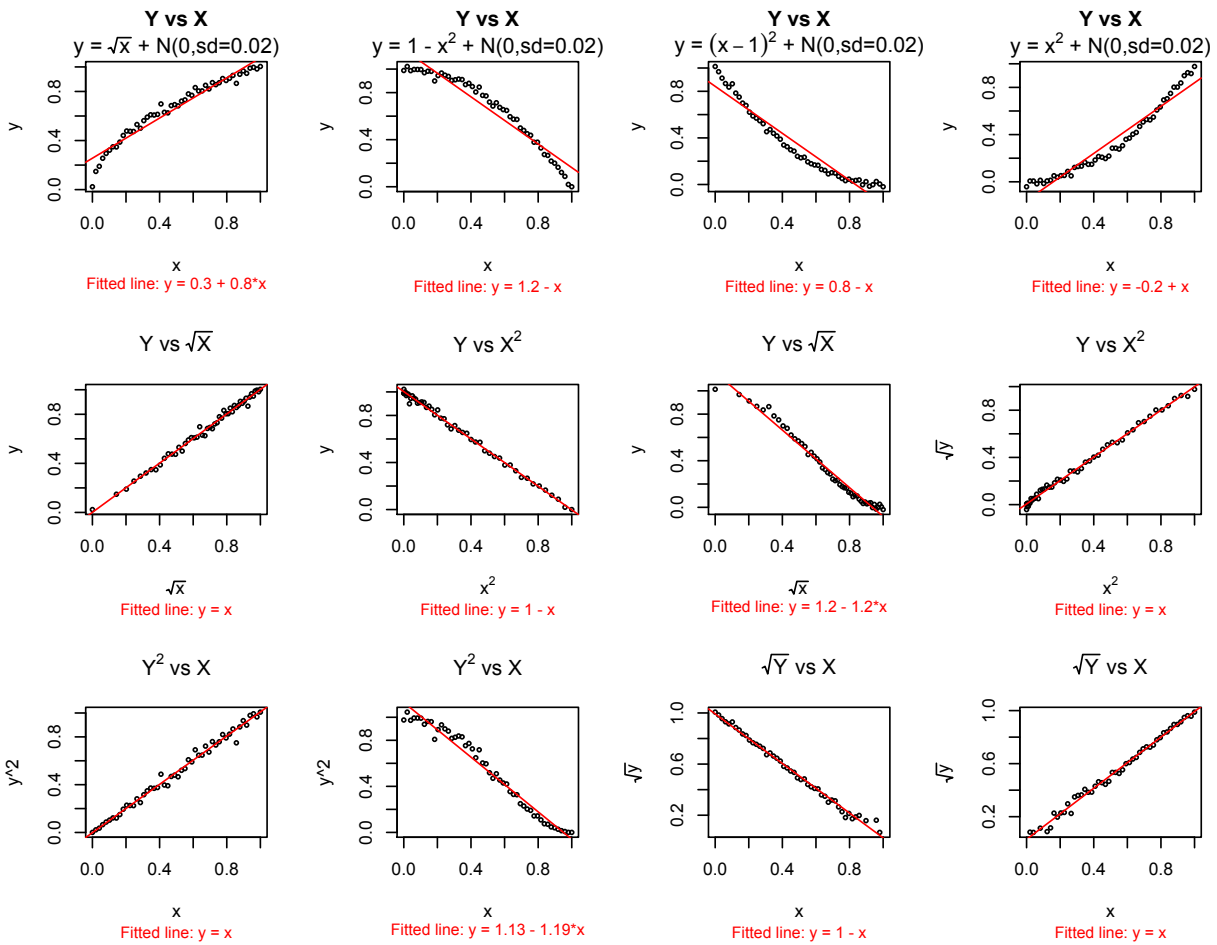
Mtcars Example




```
> transformY = 1/sqrt(y)
> plot(transformY~x,xlab="Weight of
Cars",ylab=expression(1/sqrt(mpg)),main="Transformed Y vs X")
> fit2=lm(transformY ~x)
> abline(fit2)
> coef(fit2)
(Intercept)         x
  0.1288965   0.0315443
```

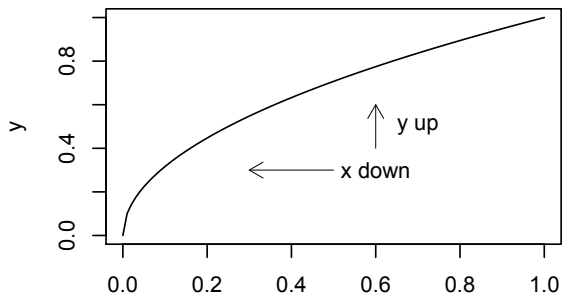
Much more linear!!

Transforming for straightness



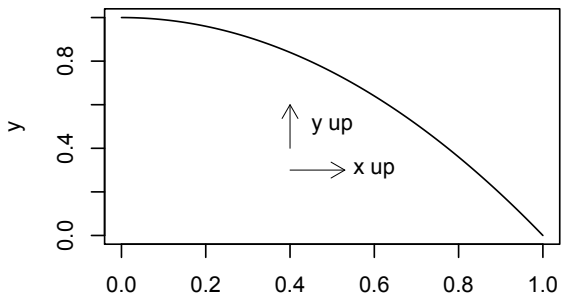
Transforming for straightness

$Y = \sqrt{X}$



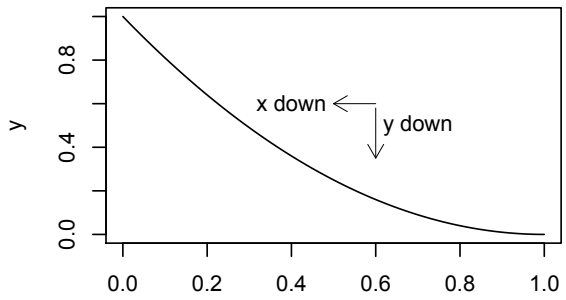
straighten with $\sqrt{x}$ or y^2

$Y = 1 - X^2$



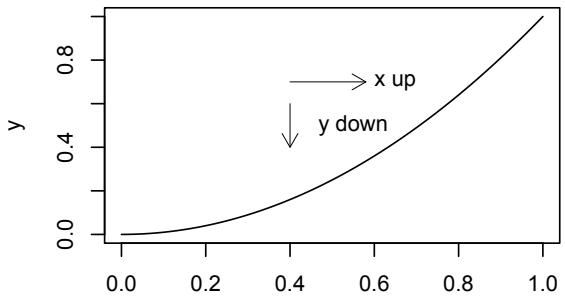
straighten with x^2 or y^2

$Y = (X - 1)^2$



straighten with $\sqrt{x}$ or $\sqrt{y}$

$Y = X^2$



straighten with x^2 or $\sqrt{y}$

Transformations for Two Way Tables

Let us say you have a two-way table of counts like the following contingency table

```
smoke <- matrix(c(51,43,22,92,28,21,68,22,9),ncol=3,byrow=TRUE)
colnames(smoke) <- c("High","Low","Middle")
rownames(smoke) <- c("current","former","never")
smoke <- as.table(smoke)
smoke
        High Low Middle
current   51  43     22
former    92  28     21
never     68  22      9
```

Transformations for Two Way Tables

A general $(R \times C)$ contingency table looks like this

	Column Variable			
Row Variable	Column Category 1	Column Category 2	$\cdots$	Column Category C
Row category 1	y_{11}	y_{12}	$\cdots$	y_{1C}
Row category 2	y_{21}	y_{22}	$\cdots$	y_{2C}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Row category R	y_{R1}	y_{R2}	$\cdots$	y_{RC}

A common additive way to model the data is the following model

$$\begin{aligned} y_{ij} &= m + a_i + b_j + r_{ij} \\ &= \text{common value} + \text{row effect} + \text{column effect} + \text{fluctuation} \end{aligned}$$

where m = common value, a_i = row effect, b_j = column effect and r_{ij} = *residuals*.

Diagnostic Plot

The **Diagnostic Plot for Contingency Tables** has the following procedure

1. Find estimates for m, a_i and, b_i (Ch 6 of UREDA) We'll go over this later....
2. Plot $\frac{a_i b_j}{m}$ on horizontal axis vs $r_{ij} = y_{ij} - (m + a_i + b_j)$ on vertical axis.
3. When the pattern is roughly linear, set $p = 1 - b = 1 - \text{slope}$.
4. Transform y_{ij} to $T_p(y_{ij})$ and fit additive model.

Why Does This Work?

Suppose that we can find a power transform where the fit of data would be exact, then

$$y_{ij}^{p} = m + a_i + b_j$$

Thus,

$$y_{ij}^{p} = m^{1/p}\left(1 + \frac{a_i}{m} + \frac{b_j}{m}\right)^{1/p}.$$

Now if we use the Taylor series expansion for $(1+t)^{1/p}$ we obtain

$$y_{ij}^{p} \approx m^{1/p}\left[1 + \frac{1}{p}\left(\frac{a_i}{m} + \frac{b_j}{m}\right) + \frac{(1-p)}{2p^2}\left(\frac{a_i}{m} + \frac{b_j}{m}\right)^2\right].$$

Why Does This Work?

Rearranging this into 4 terms: those that don't depend on i or j, those that depend only on i, those that depend on j and those taht depend on both gives

$$y_{ij}^p \approx m^{1/p}\left[1+\left(\frac{1}{p}\frac{a_i}{m}+\frac{(1-p)}{2p^2}\frac{a_i^2}{m^2}\right)+\left(\frac{1}{p}\frac{b_j}{m}+\frac{(1-p)}{2p^2}\frac{b_j^2}{m^2}\right)+\left(\frac{(1-p)}{2p^2}\frac{2a_ib_j}{m^2}\right)\right].$$

Now let

$$\begin{aligned}
D &= m^{1/p},\\
\frac{A_i}{D} &= \left(\frac{1}{p}\frac{a_i}{m}+\frac{(1-p)}{2p^2}\frac{a_i^2}{m^2}\right),\\
\frac{B_j}{D} &= \left(\frac{1}{p}\frac{b_j}{m}+\frac{(1-p)}{2p^2}\frac{b_j^2}{m^2}\right) \text{ and,}\\
\frac{C_{ij}}{D} &= \left(\frac{(1-p)}{2p^2}\frac{2a_ib_j}{m^2}\right)=\frac{(1-p)}{p^2}\frac{a_i}{m}\frac{b_j}{m}
\end{aligned}$$

so,

$$y_{ij} \approx D + A_i + B_j + C_{ij}.$$

Why Does This Work?

Now, in the second order approximation, the product of the two middle terms is

$$\frac{A_i}{D}\frac{B_j}{D} \approx \frac{1}{p^2}\frac{a_i}{m}\frac{b_j}{m}$$

and the last term is

$$\boxed{\frac{C_{ij}}{D} = \frac{1}{p^2}\frac{a_i}{m}\frac{b_j}{m} \approx (1-p)\frac{A_i}{D}\frac{B_j}{D}.}$$

Thus to second order approximation

$$y_{ij} \approx D + A_i + B_j + (1-p)\frac{A_i B_j}{D}.$$

Why Does This Work?

So to the extent that the model is any good, the residuals would be given by

$$R_{ij} \approx (1 - p)\frac{A_i B_j}{D}$$

and plotting R_{ij} against $\frac{A_i B_j}{D}$ provides the diagnostic plot.
We will go into this more later.

Summary of Transformations

			Plotting Coordinates	
Structure	Objective	Notation	Horizontal	Vertical
Single batch	Symmetry	M = median y_q = lower qth quartile y_{1-q} = upper qth quartile	$\frac{(y_{1-q}-M)^2+(M-y_q)^2}{4M}$	$\frac{(y_{1-q}+y_q)}{2} - M$
Several batches	Equal Spread	M_i = median for ith batch Q_i = IQR for ith batch	$\log(M_i)$	$\log(Q_i)$
y vs x	Straight Line	M_x = median for X M_y = Median of Y C = Slope of Y vs X	$\frac{C^2(X-M_X)^2}{2M_y}$	$Y - M_y - C(X - m_x)$
two-way table	Additivity	$y_{ij} = m + a_i + b_j + r_{ij}$	$\frac{a_i b_j}{m}$	$r_{ij} = y_{ij} - (m + a_i + b_j)$

More on Data and Transformations

Types of data:

1. Counts
2. Amounts (quantitative)
3. Balances (differences, ratios, log(ratios))
4. Counted fractions and percentages
5. Categories $\rightarrow$ Ranks

Possible re-expressions:

1. Counts: Square roots
2. Amounts: nothing if nearly symmetric; else Ladder
3. Balances: often none; but re-expression of amounts or counts before subtraction sometimes helps
4. Counted fractions and percentages: re-expression is often helpful, special techniques
5. Ranks: Normal scores $\Phi^{-1}(p_i)$, $p_i = \dfrac{i - 1/3}{n + 1/3}$

Transformations for Fractions ($0 \leq p \leq 1$) (EDA Ch15)

- Counted fraction: $p = (\#\text{ successes}) / (\text{total }\#\text{ observations})$
- Objective: compare, analyze, describe distributions (sequence of counted fractions)
- Cutting value or cut: a value that divides "below" from "above"
- Instead of (count below cut-off)/n, use

$$(\text{count below} + \frac{1}{2}\text{count equal} + \frac{1}{6})/(\text{total count} + 1/3)$$

- Example: $n = 6$, $\{X\} = \{0, 1, 2, 2, 3, 7\}$
  1. cut $= 2$, $P\{x \leq 2\} = (2 + 1/2 \times 2 + 1/6)/(6 + 1/3) = 0.5$
  2. cut $= 3$, $P\{x \leq 3\} = (4 + 1/2 \times 1 + 1/6)/(6 + 1/3) = 0.737$
  3. cut $= 4$, $P\{x \leq 4\} = (5 + 1/6)/(6 + 1/3) = 0.816$

Three Matched Scales (EDA Ch15)

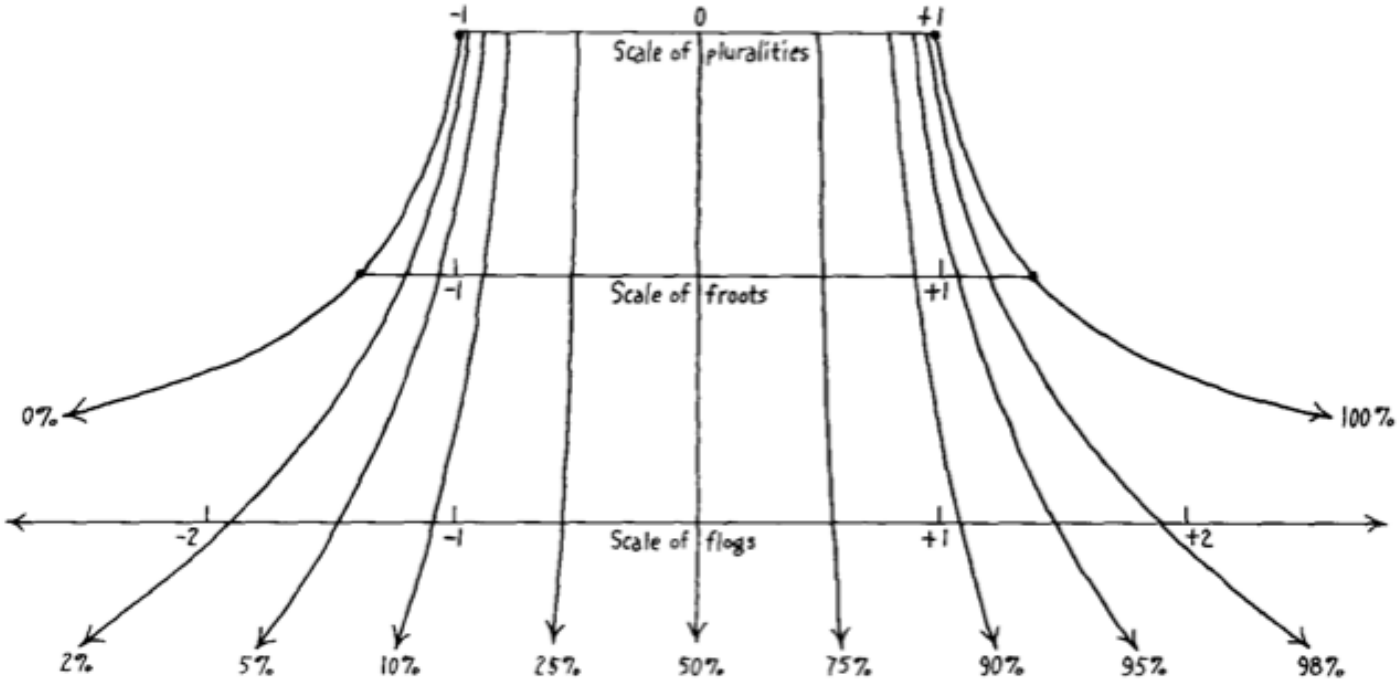
Suppose f is the fraction concerned, we want new scales to have nice properties:

- $50\% \rightarrow 0$
- swap of f with $1 - f \rightarrow$ change the sign, not the size

Matched transformations for f:

- plurality: $f - (1 - f)$
  Or cents (folded percents): (% yest) - (% no)
- froots (folded roots): $\sqrt{2f} - \sqrt{2(1 - f)}$
- flogs (folded logs): $(1/2)\ln(f) - (1/2)\ln(1 - f)$
- Both (froots and flogs) stretch ends of scales

Illustration (EDA Ch15)



Example: Comparison Within one Dataset

Example: Gallup Poll results (Washington Post, Feb 62)

Protestants shift support to Kennedy

	Protestants		Catholics	
Date	Nov 1960	Jan 1962	Nov 1960	Jan 1962
Kennedy	38%	59%	78%	89%
Nixon	62%	41%	22%	11%

"One of the major reasons for the President's popularity has been his success in allaying anti-Catholic sentiment while not losing the support of fellow Catholitcs" (Protestants: 21% increase; Catholics: 11% increase).

Example

What is really going on?

- Scale stretching near the end (38% $\rightarrow$ 42%, 89% $\rightarrow$ 93%)
- Take flogs:

Data	Protestant	Catholics	Difference
Nov' 60	-0.24	+0.63	+0.87
Jan' 62	+0.18	+1.05	+0.87
Change	+0.42	0.42	

- JWT: "JFK's popularity increased by 0.42 on the flog scale, regardless of whether you asked Protestants or Catholics".
- flog: not this good all the time, but helps often

Resistant Regression

Resistant Regression

Motivating Example: Test Scores

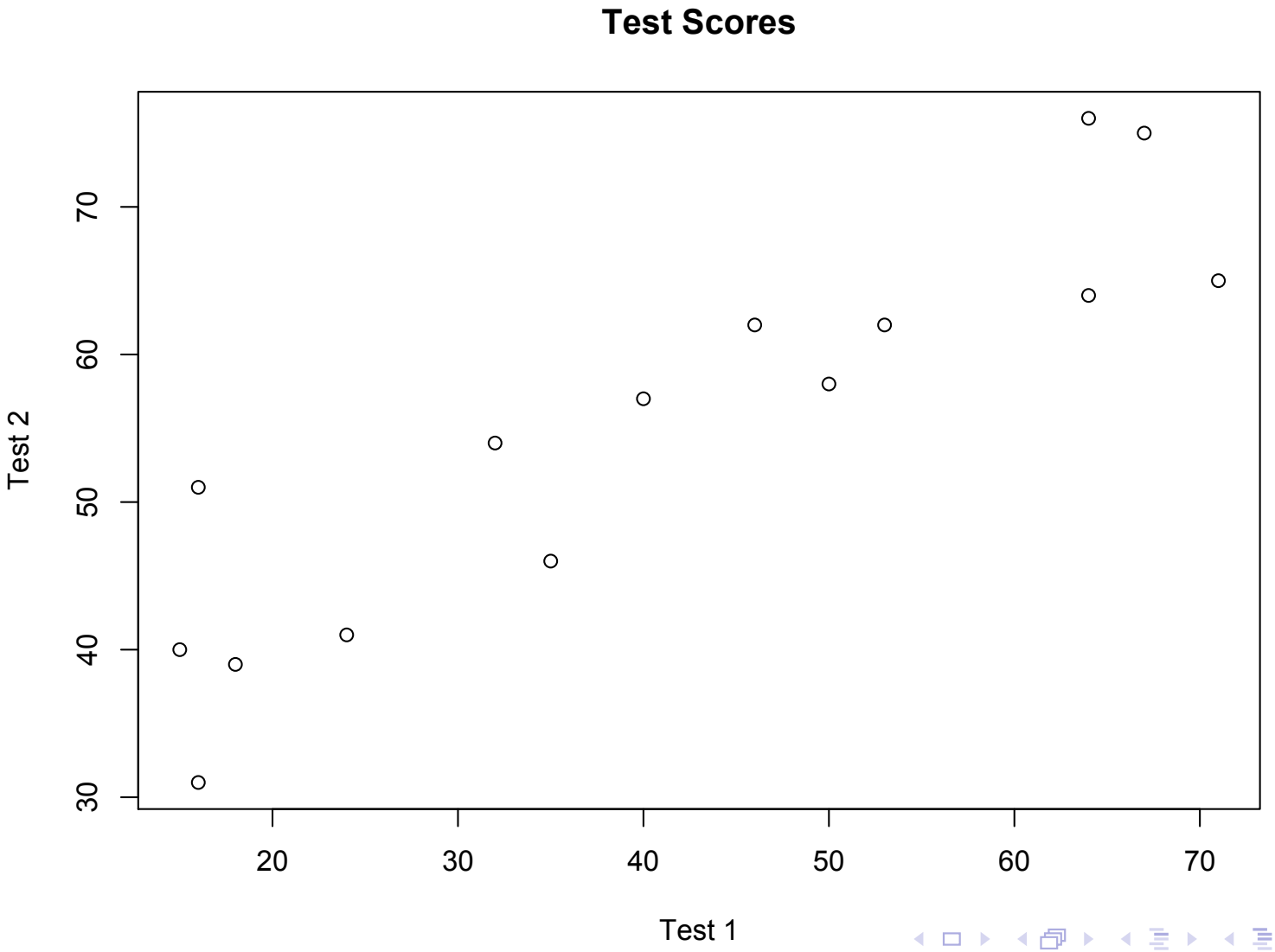
Example: Use productivity measures to grade 15 employees who work on an assembly line. Each employee was tested once, then again a month later.

Test 1	Test 2
50	58
35	46
15	40
64	76
53	62
18	39
40	57
24	41
16	31
67	75
46	62
64	64
32	54
71	65
16	51

Goal: Exploration of how y may vary with x or depend on x

1. Plot y against x
2. When the plot reveals clear curvature, re-expression of y or of x or both y and x.
3. If the pattern of points is fairly straight, fit a straight line $y = a + bx$;
4. Examine residuals.

Motivating Example: Test Scores



Least Squares Regression

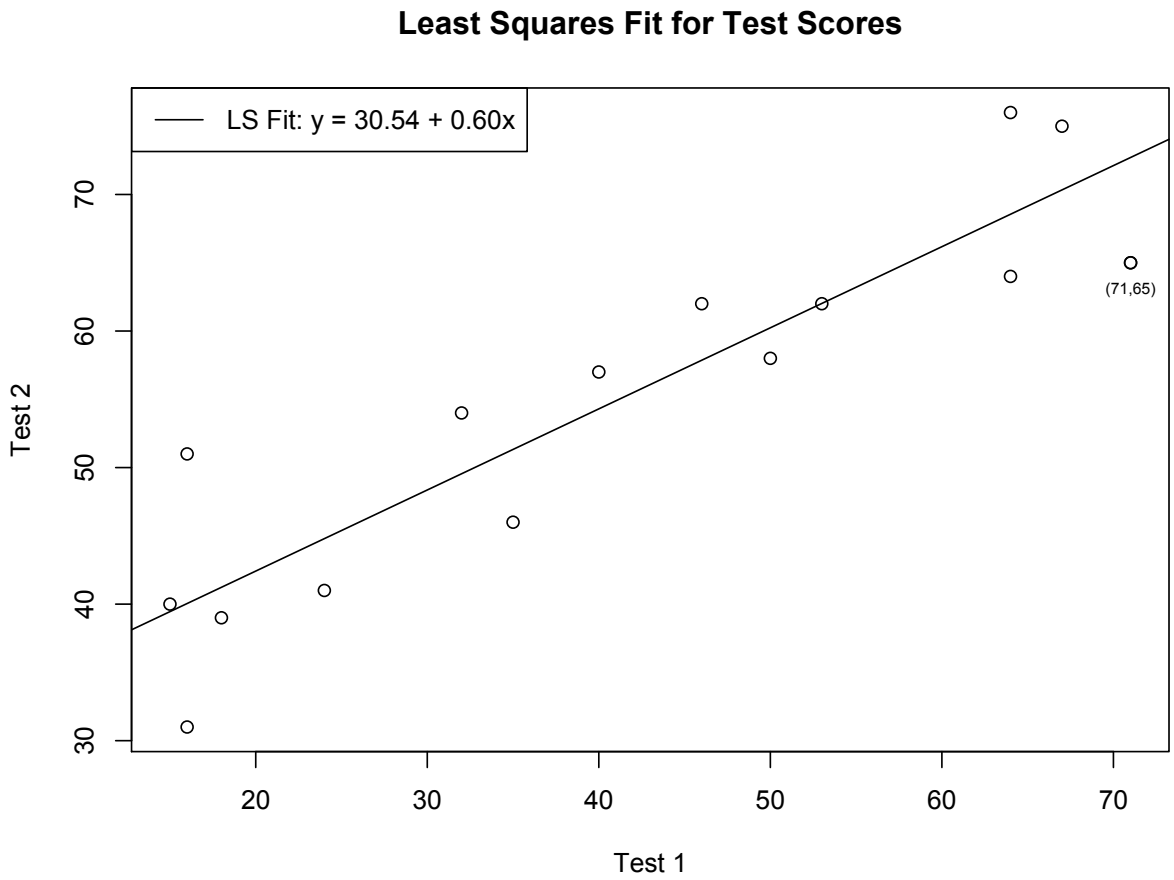
How to fit straight lines? $\hat{y} = a + bx$

1. Classical methods: least-squares regression.
   - Algebraically simple calculation:

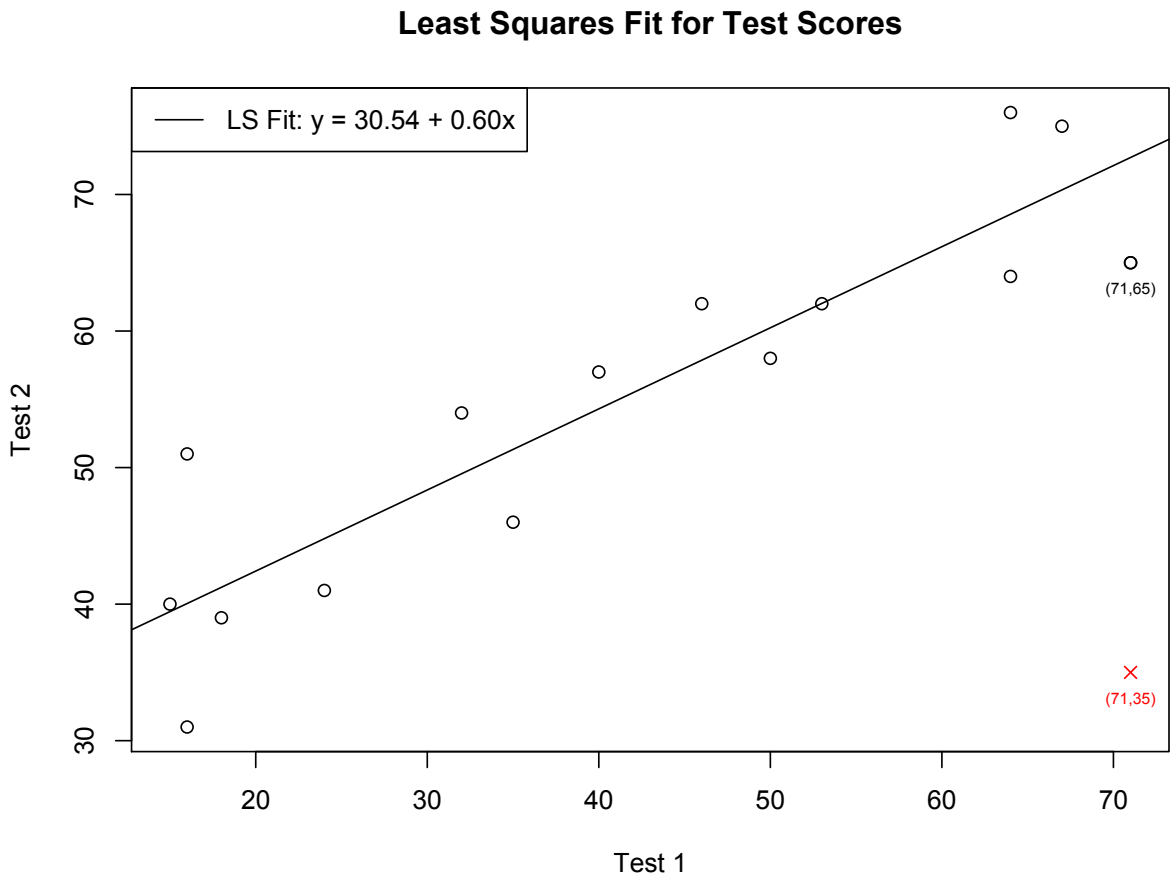
$$b = \sum_{i=1}^{n} c_i y_i, \quad c_i = \frac{x_i - \bar{x}}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \quad a = \bar{y} - b\bar{x} = \text{mean}\{y_i - bx_i\}$$

   - Fits inference built on the Gaussian distribution (LS estimators are also MLEs)
   - Minimizes sum of squared distances (residuals)
2. R functions
   - lm(y∼x)
   - summary(lm(y∼x))
   - abline(lm(y∼x))

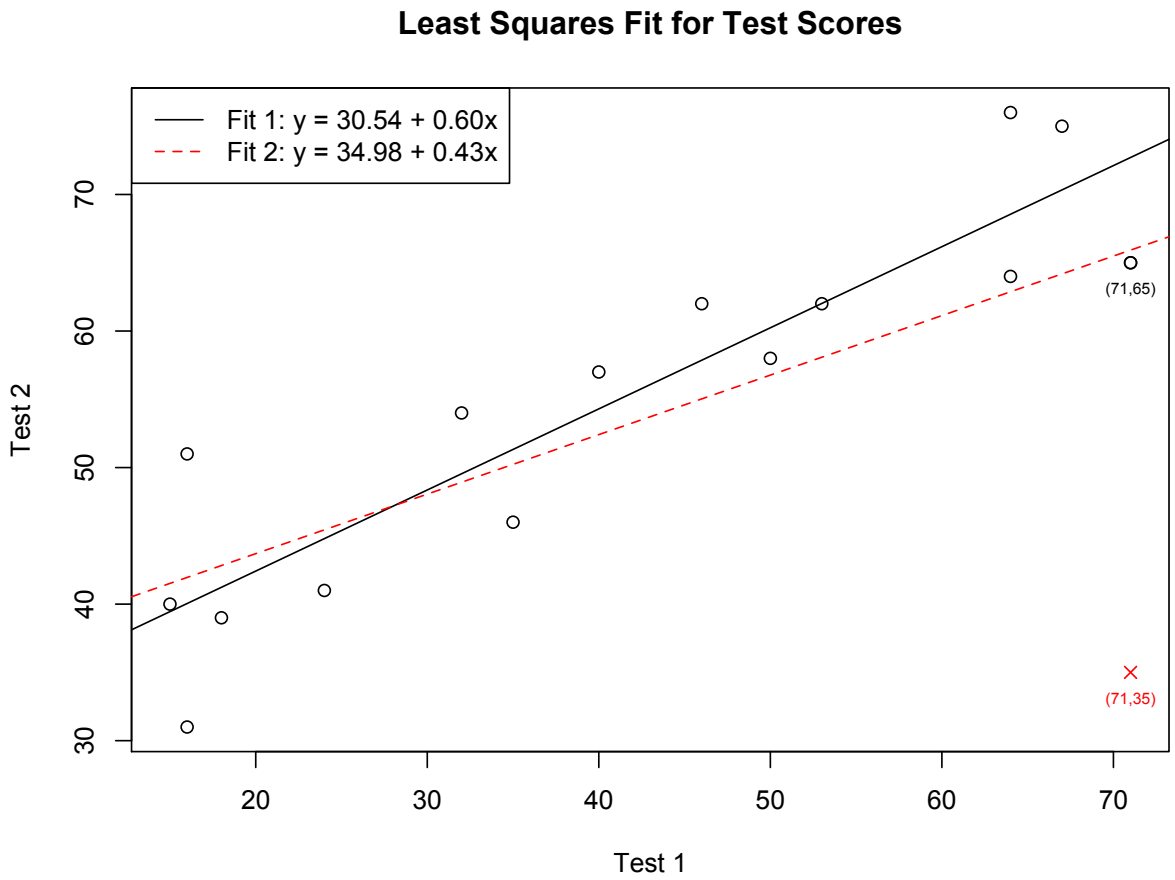
Motivating Example: LS fit



Motivating Example: outlier

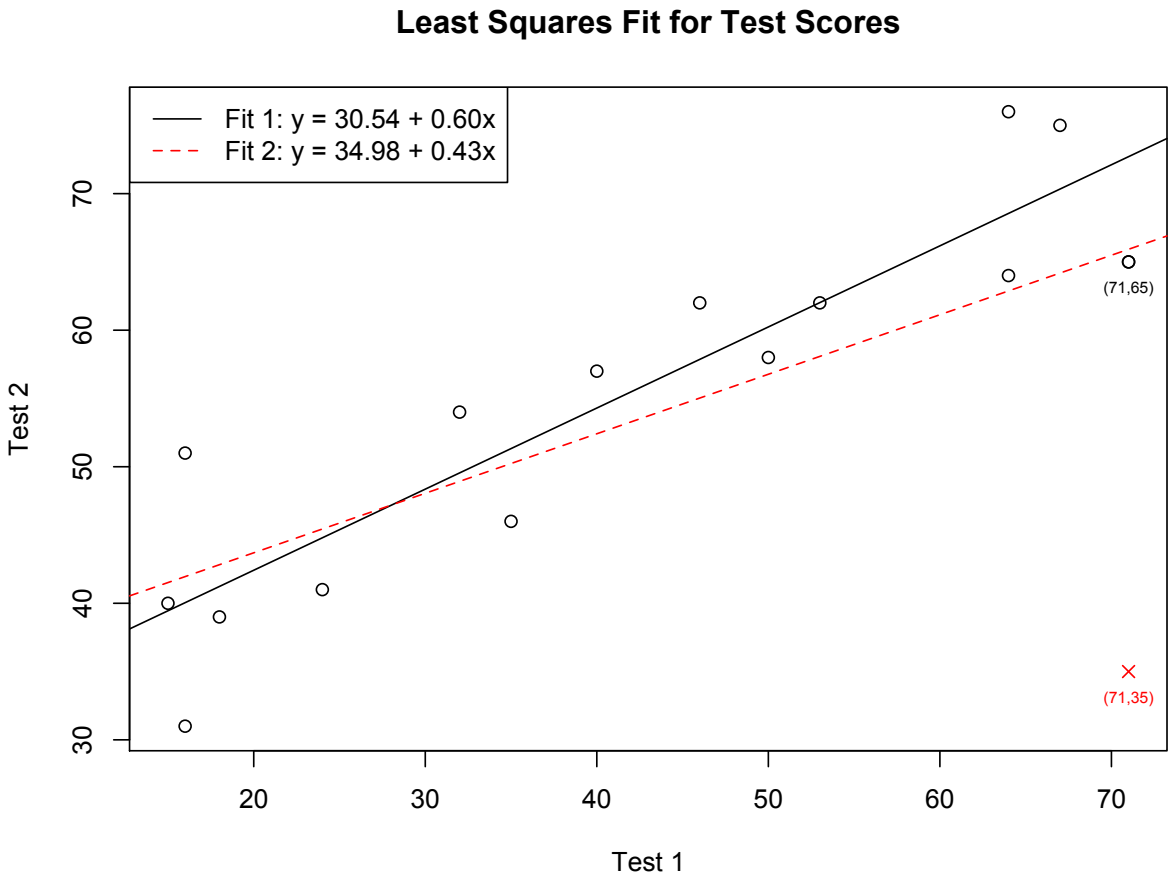


Motivating Example: LS fit



Motivating Example: LS fit with an outlier

Least Squres Fit: No resistance! Outliers can destroy LS estimates, misleading summary of the relationship between y and x.



Example 1: State data (in R: state.x77)

Matrix with 50 rows and 8 columns giving the following statistics in the respective columns.

```
state.x77[1:4,]
             Pop Income Illit LifeExp Murder HSGrad Frost Area
Alabama     3615   3624   2.1   69.05   15.1   41.3    20   50708
Alaska       365   6315   1.5   69.31   11.3   66.7   152  566432
Arizona     2212   4530   1.8   70.55    7.8   58.1    15  113417
Arkansas    2110   3378   1.9   70.66   10.1   39.9    65   51945
```

- Pop = Population (1000s, 7/1/75)
- Income: per capita (1974)
- Illit = Illiteracy (1970, % of population)
- LifeExp = Life Expectancy in years (1969–71)
- Murder: # murder per 100,000 population (1976)
- HSGrad: % HS graduates (1970)
- Frost: mean # days with min temp $< 32^\circ F$ (1931–1960)
- Area: land area in square miles

Example 1: stem-and-leaf

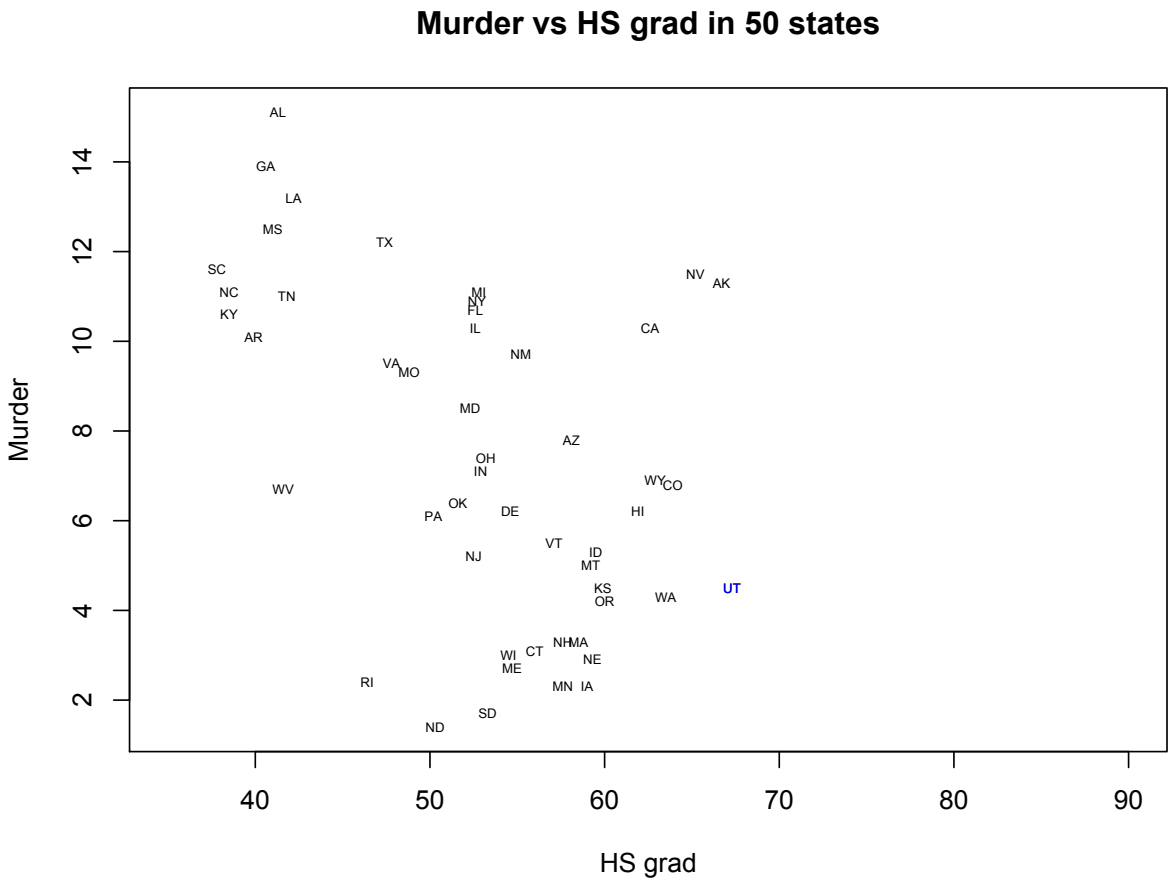
Is there a connection between %HS graduates and murder rate?

Plot *murder* vs *HSgrad*; first, stem-and-leaf:

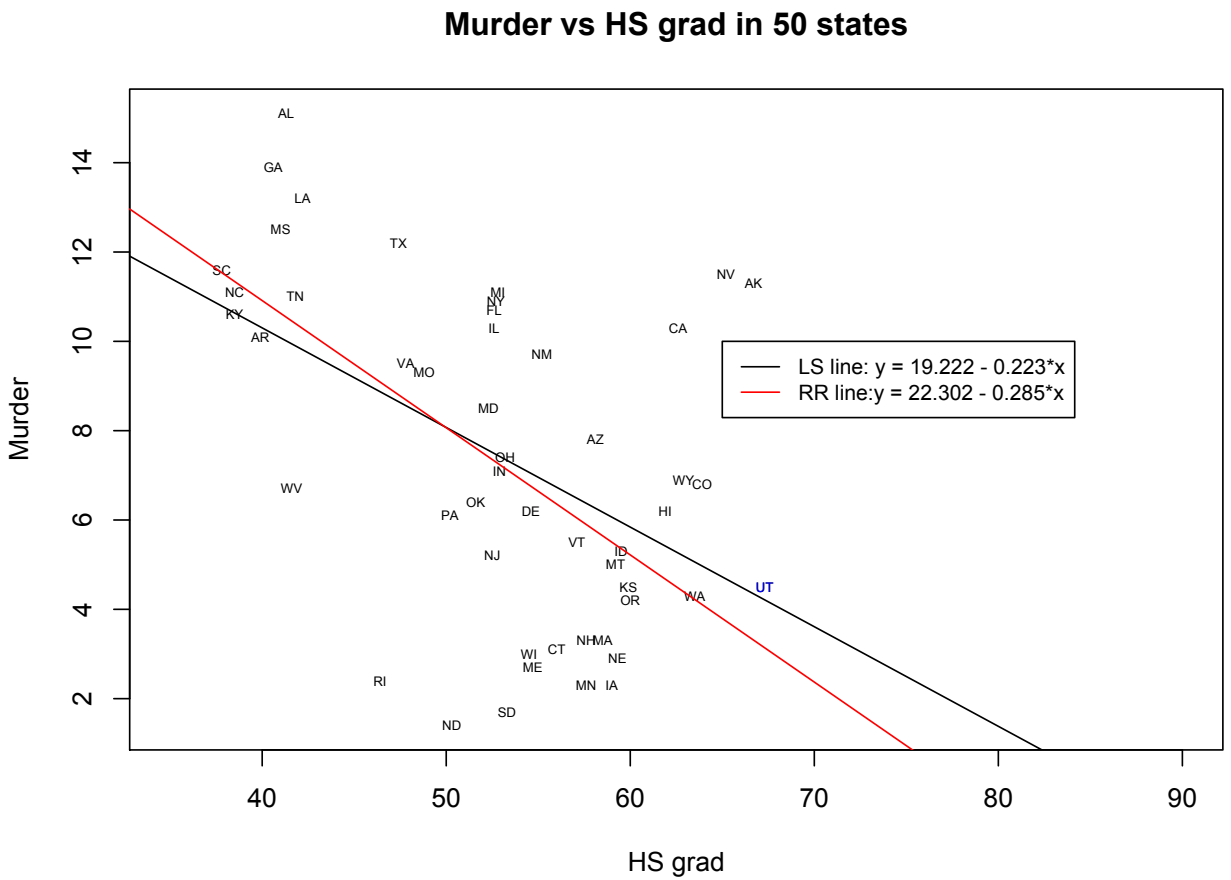
```
stem(murder)
  The decimal point is at the |

   0 | 47
   2 | 334790133
   4 | 23550235
   6 | 1224789148
   8 | 5357
  10 | 133679011356
  12 | 2529
  14 | 1
```

```
stem(HSgrad)
  The decimal point is 1 digit(s)
  to the right of the |
  3+ | 899
  4  | 0111222
  4+ | 6789
  5  | 002233333333
  5+ | 5555678889999
  6  | 00023344
  6+ | 577
```

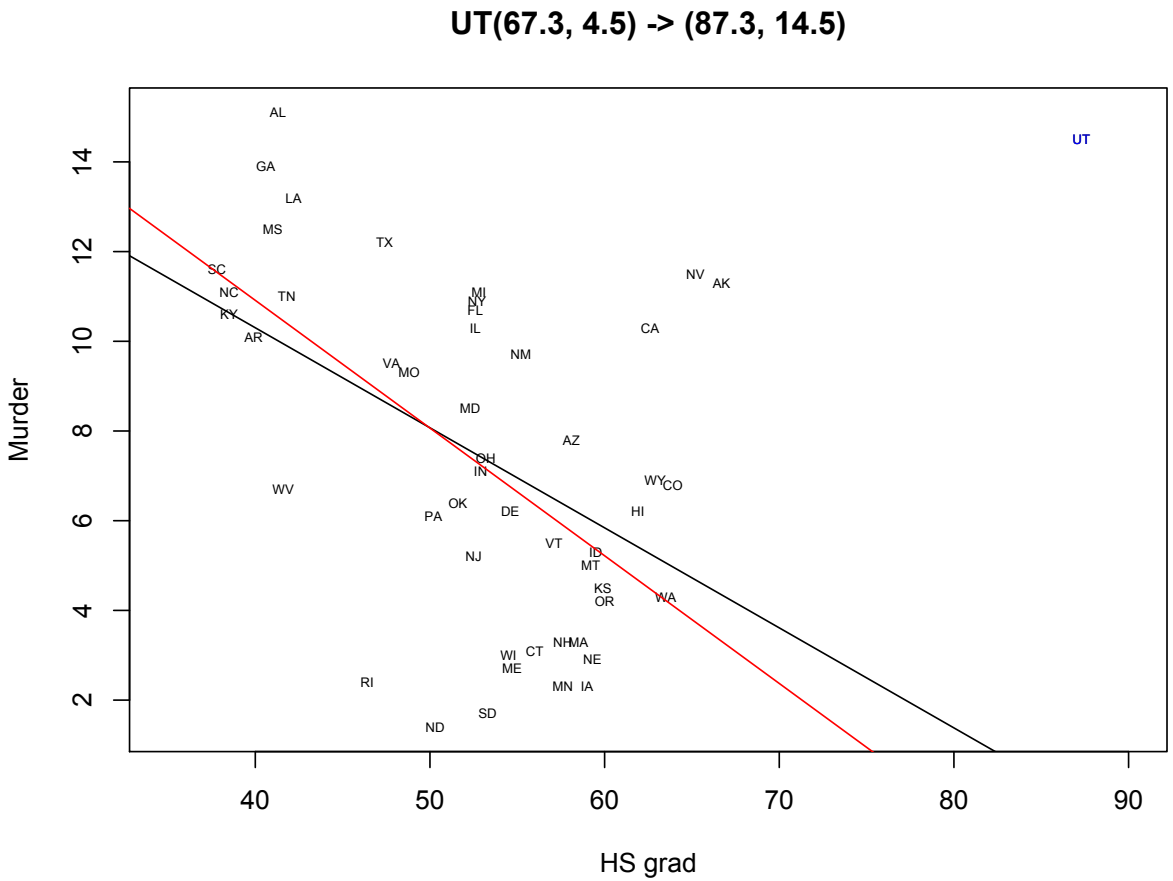
Example 1: Scatterplot of data



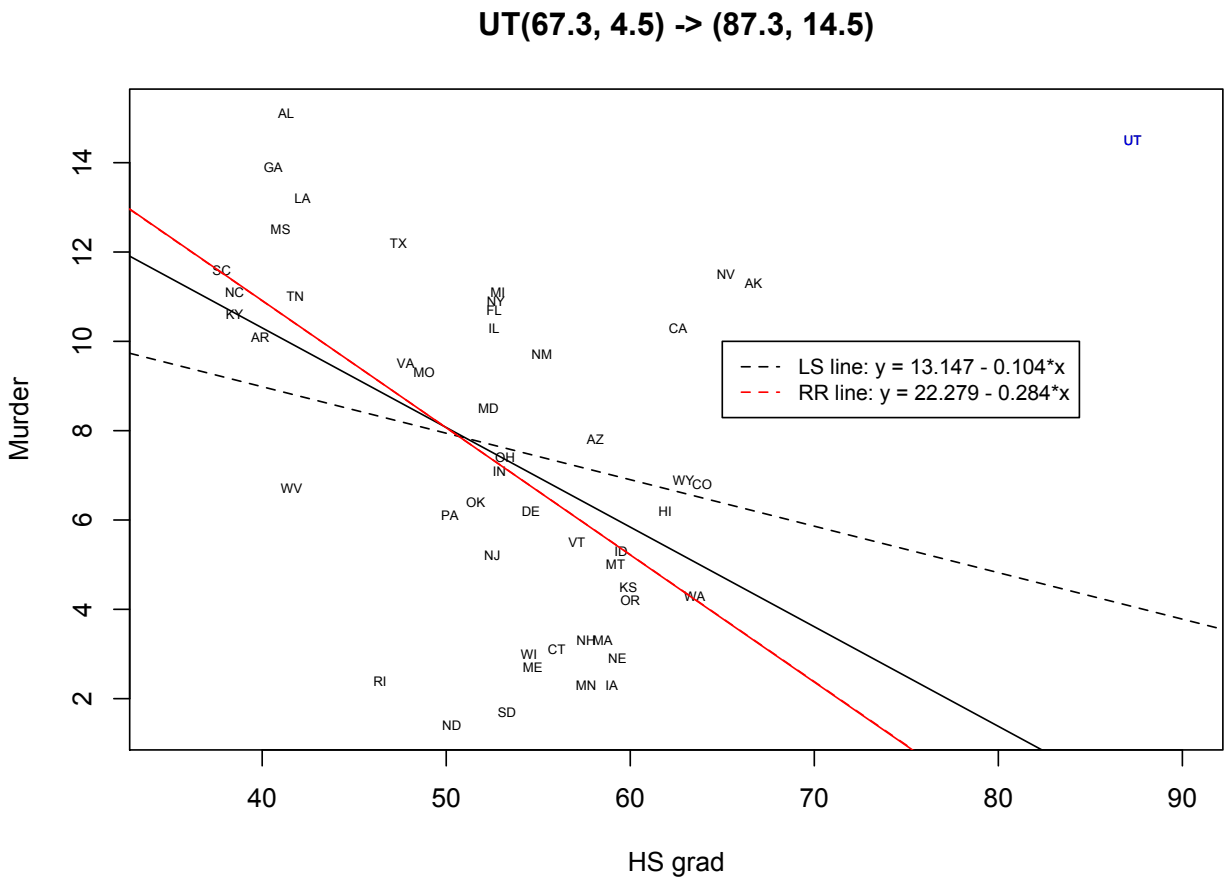
Example 1: Fit lines



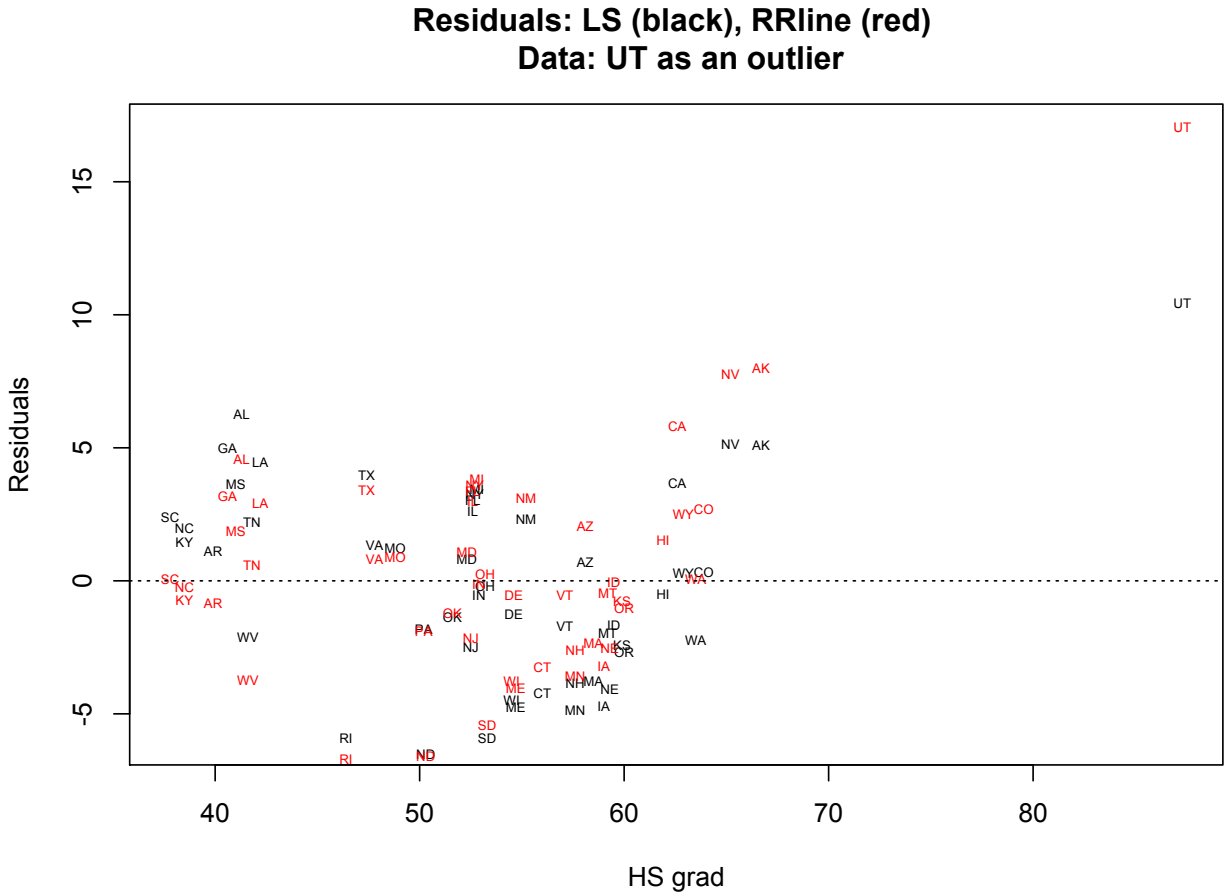
Example 1: Outlier



Example 1: Fit lines with an outlier



Example 1: Compare residuals



Example 1: Compare residuals (stem-and-leaf)

```
  LS
-6 | 5
-4 | 99987521
-2 | 98754210
-0 | 87743552
 0 | 33781235
 2 | 0234602467
 4 | 04011
 6 | 3
 8 |
10 | 4
```

```
RRline
-6 | 76
-4 | 40
-2 | 886326542
-0 | 920887655211
 0 | 112689159
 2 | 05790124468
 4 | 68
 6 | 8
 8 | 0
10 |
12 |
14 |
16 | 0
```

Robust-resistant line

How to fit straight lines? $\hat{y} = a + bx$

- Robust-resistant line:
  1. Sort the values of x, $(x_1 \leq x_2 \leq \cdots \leq x_n)$, and divide the n data points (x_i, y_i) into 3 nearly equal groups, according to the x values.
  2. Assume no ties among x_i's

	$n = 3k$	$n = 3k+1$	$n = 3k+2$
L=left group	k	k	$k+1$
M=middle group	k	$k+1$	k
R=right group	k	k	$k+1$

  3. Find summary points (median x-value, median y-value) in outer groups (x_L, y_L), (x_R, y_R)

$$\boxed{\text{slope } b = \frac{y_R - y_L}{x_R - x_L}}$$

  4. Intercept $\boxed{a = \text{median}\{y_i - bx_i\}}$

Example 1: Compare fits (residuals and coefficients)

Compare differences in slope and intercept

```
      UT = (67.3, 4.5)

  Min   1Q   Med  3Q   Max
-6.60 -2.22 0.00 2.27 6.95

           Estimate SE  t-stat
Intercept 19.222 3.092  6.22*
HSgrad    -0.223 0.057 -3.87*

R-sq=0.238 Adj R-sq=0.222

RRline:
Int=22.3020 Slope=-0.2847
```

```
      UT = (87.3, 14.5)

  Min   1Q   Med  3Q   Max
-6.51 -2.46 0.04 2.57 10.44

    Estimate SE  t-stat
    13.147 3.129  4.20*
    -0.104 0.058 -1.80

R-sq=0.064 Adj R-sq=0.044


Int=22.2789 Slope=-0.2843
```

LS line vs RR line

Compare:

- Difference in slope and intercept
- LS line: b is a weighted average of y_i's, weights depend on distance of x_i from $\bar{x}$. The further x_i is from $\bar{x}$, the greater its impact on slope. An outlier in y_i also affects the slope.

$$b = \sum_{i=1}^{n} c_i y_i, \quad c_i = \frac{x_i - \bar{x}}{\sum_{i=1}^{n} (x_i - \bar{x})^2}.$$

- RR line: A single x_i or y_i enters slope calculation only as a median, so value of an extreme x_i or y_i is not used.

LS line vs RR line

Compare:

- Difference in slope and intercept
- LS line: b is a weighted average of y_i's, weights depend on distance of x_i from $\bar{x}$. The further x_i is from $\bar{x}$, the greater its impact on slope. An outlier in y_i also affects the slope.

$$b = \sum_{i=1}^{n} c_i y_i, \quad c_i = \frac{x_i - \bar{x}}{\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

- RR line: A single x_i or y_i enters slope calculation only as a median, so value of an extreme x_i or y_i is not used.
- LS line: An outlier in y_i affects the intercept.

$$a = \begin{cases} \text{mean}\{y_i - bx_i\} - LS \\ \text{median}\{y_i - bx_i\} - RR \end{cases}$$

LS line vs RR line

Compare:

- Difference in slope and intercept
- LS line: b is a weighted average of y_i's, weights depend on distance of x_i from $\bar{x}$. The further x_i is from $\bar{x}$, the greater its impact on slope. An outlier in y_i also affects the slope.

$$b = \sum_{i=1}^{n} c_i y_i, \quad c_i = \frac{x_i - \bar{x}}{\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

- RR line: A single x_i or y_i enters slope calculation only as a median, so value of an extreme x_i or y_i is not used.
- LS line: An outlier in y_i affects the intercept.

$$a = \begin{cases} \text{mean}\{y_i - bx_i\} - LS \\ \text{median}\{y_i - bx_i\} - RR \end{cases}$$

- RR may need to iterate: fit line to residuals

Example 2: Brain and Body Weights for 28 Species

An example from R data library **MASS**

```
> library(MASS)
> data(Animals)
> help(Animals)
```

```
Description
Average brain and body weights for 28 species of land animals.

Format
body weight in kg.
brain weight in g.

Source: Rousseeuw and Leroy (1987) Robust Regression and
Outlier Detection. Wiley, p. 57.
```

Example 2: Data

	Mountain beaver	Cow	Grey wolf	Goat	Guinea pig
body	1.35	465	36.33	27.66	1.04
brain	8.10	423	119.50	115.00	5.50

	Dipliodocus	Asian elephant	Donkey	Horse	Potar monkey	Cat
body	11700	2547	187.1	521	10	3.3
brain	50	4603	419.0	655	115	25.6

	Giraffe	Gorilla	Human	African elephant	Triceratops
body	529	207	62	6654	9400
brain	680	406	1320	5712	70

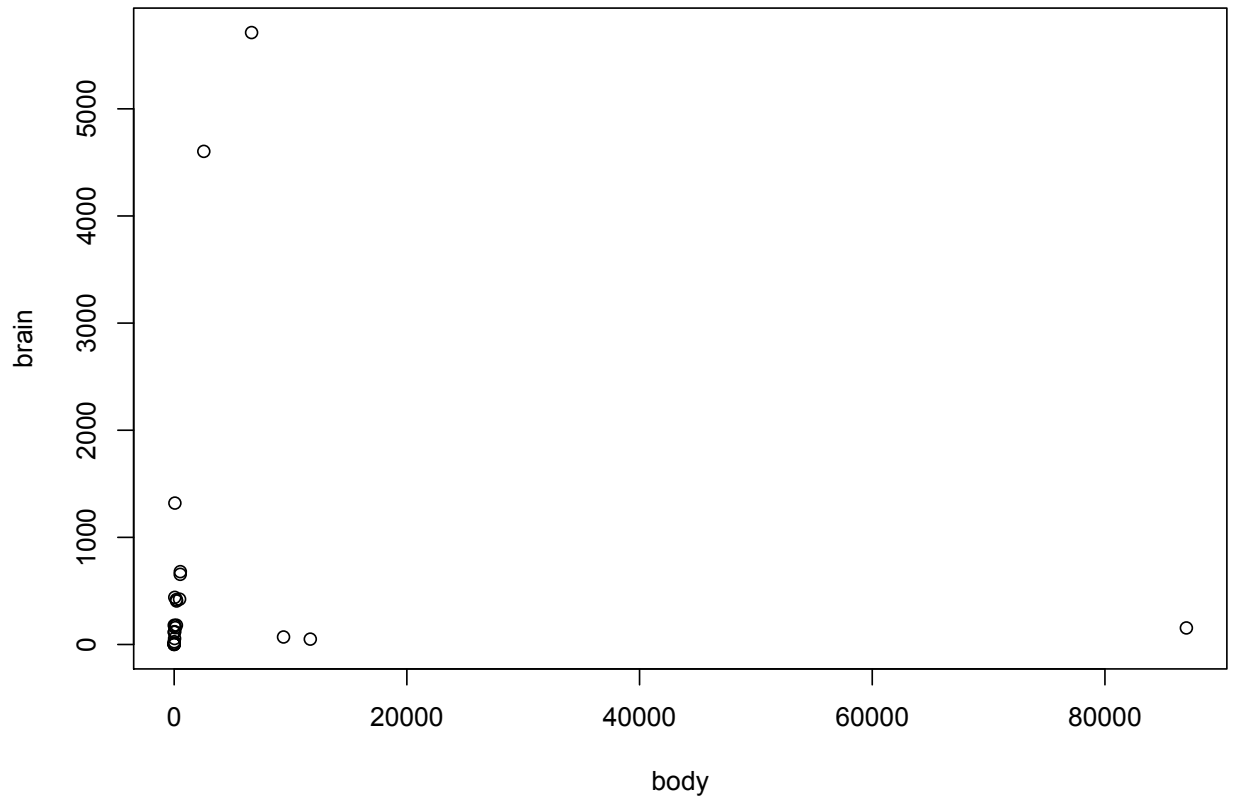
	Rhesus monkey	Kangaroo	Golden hamster	Mouse	Rabbit	Sheep
body	6.8	35	0.12	0.023	2.5	55.5
brain	179.0	56	1.00	0.400	12.1	175.0

	Jaguar	Chimpanzee	Rat	Brachiosaurus	Mole	Pig
body	100	52.16	0.28	87000.0	0.122	192
brain	157	440.00	1.90	154.5	3.000	180

Example 2: Plot data

```
Animals.abb <- c("Bvr","Cow","Wolf","Goat","Gpig","Dipl",
"AsEl","Dnky","Hors","Pmky","Cat","Grff","Grll","Humn",
"AfEl","Tric","Rmky","Kngr","Hmst","Mous","Rbbt","Shep",
"Jagr","Cmpz","Rat","Brac","Mole","Pig")

body <- Animals[,1]

brain <- Animals[,2]

plot(body,brain)
```

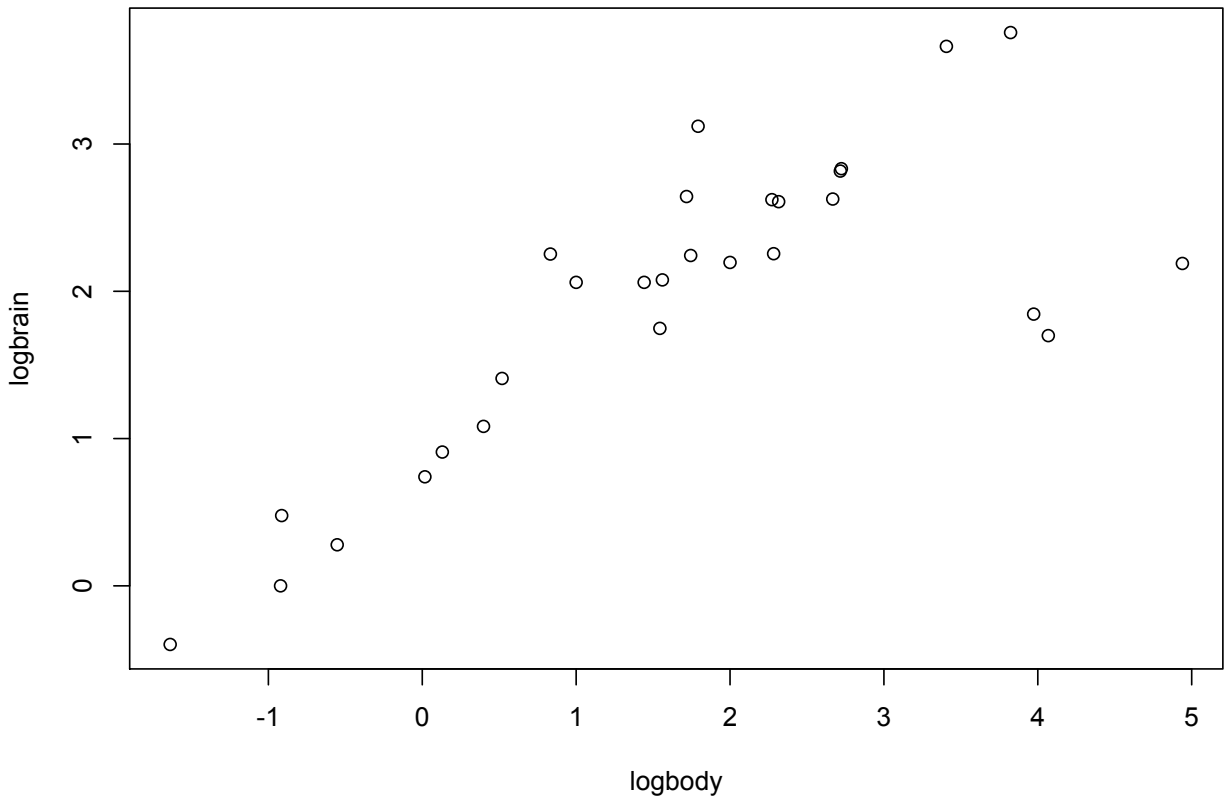
Example 2: Scatterplot



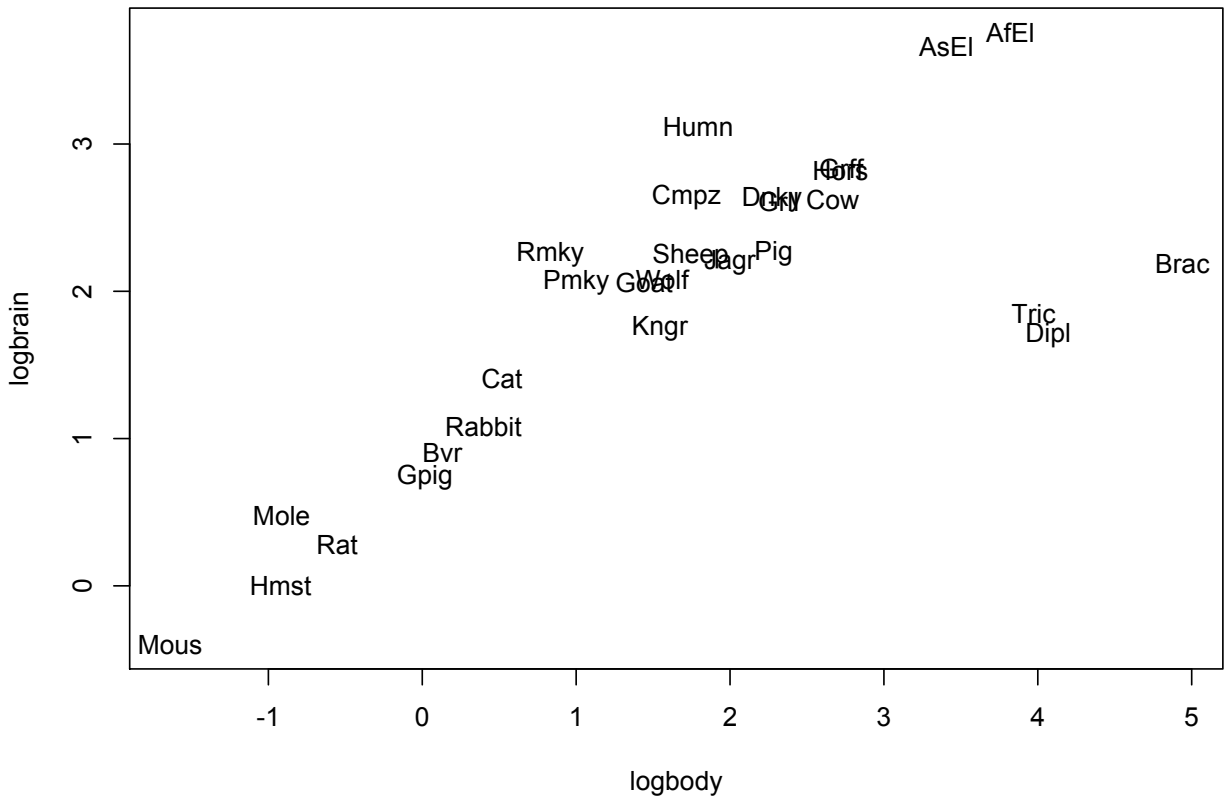
Example 2: Log Transformation

```
logbody <- log10(body)
logbrain <- log10(brain)

plot(logbody, logbrain)

plot(logbody, logbrain, type='n')
text(logbody, logbrain, Animals.abb)
```

Example 2: Plot 1 of transformed data



Example 2: Plot 2 of transformed data



Example 2: Sort transformed data

```
sort.logbb <- cbind(sort(logbody), logbrain[order(logbody)])

dimnames(sort.logbb) <- list(Animals.abb[order(logbody)],
c("logbody","logbrain"))

round(t(sort.logbb), 3)
```

Example 2: Sorted data

	Mous	Hmst	Mole	Rat	Gpig	Bvr	Rabbit
logbody	-1.638	-0.921	-0.914	-0.553	0.017	0.130	0.398
logbrain	-0.398	0.000	0.477	0.279	0.740	0.908	1.083

	Cat	Rmky	Pmky	Goat	Kngr	Wolf	Cmpz
logbody	0.519	0.833	1.000	1.442	1.544	1.560	1.717
logbrain	1.408	2.253	2.061	2.061	1.748	2.077	2.643

	Sheep	Humn	Jagr	Dnky	Pig	Grll	Cow
logbody	1.744	1.792	2.000	2.272	2.283	2.316	2.667
logbrain	2.243	3.121	2.196	2.622	2.255	2.609	2.626

	Hors	Grff	AsEl	AfEl	Tric	Dipl	Brac
logbody	2.717	2.723	3.406	3.823	3.973	4.068	4.940
logbrain	2.816	2.833	3.663	3.757	1.845	1.699	2.189

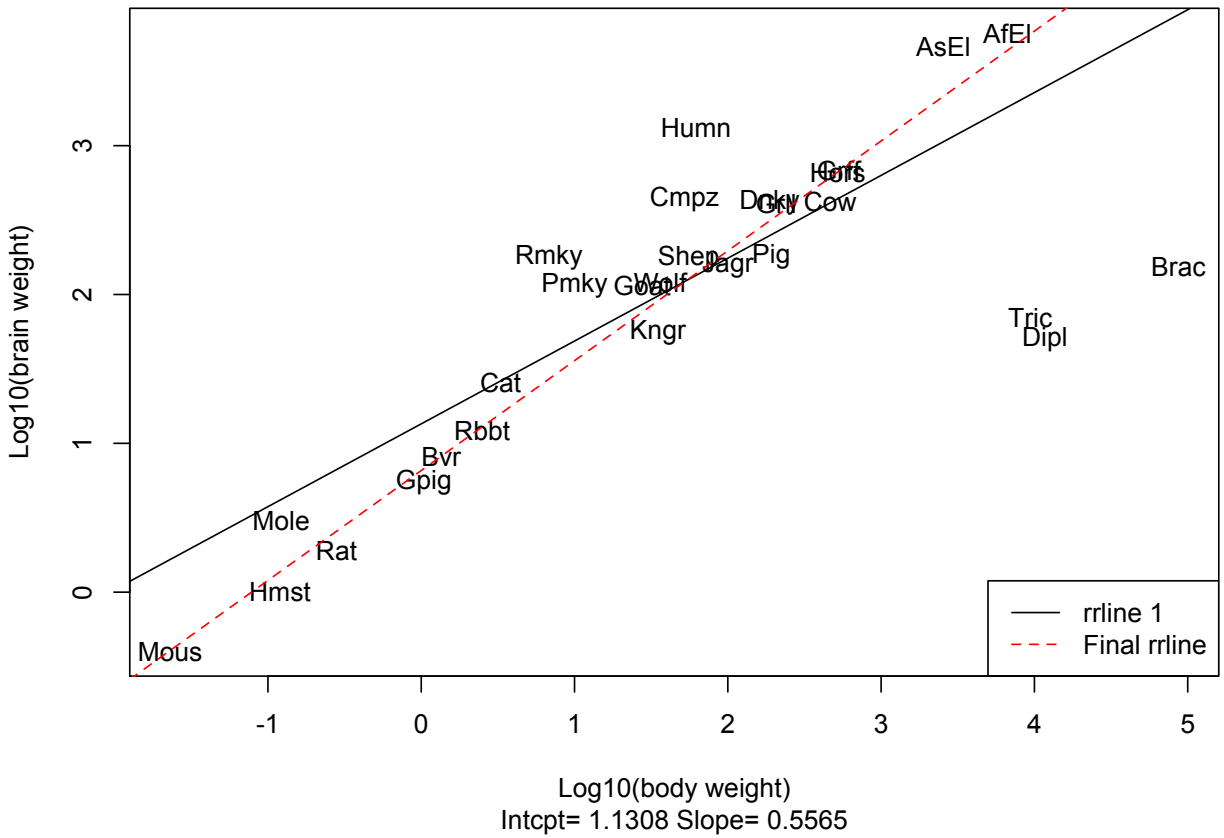
Example 2: rrline1

```
rrline1 <- function(x,y) {
  n <- length(x); nmod3 <- n%%3;
  if(nmod3 == 0) n3 <- n/3;
  if(nmod3 == 1) n3 <- (n-1)/3;
  if(nmod3 == 2) n3 <- (n+1)/3;
#  n3 <- floor((length(x)+1.99)/3)
  x.order <- order(x)
  medxL <- median(x[x.order][1:n3])
  medxR <- median(rev(x[x.order])[1:n3])
  medyL <- median(y[x.order][1:n3])
  medyR <- median(rev(y[x.order])[1:n3])
  slope1 <- (medyR - medyL)/(medxR - medxL)
  int1 <- median(y - slope1 * x)
  newy <- y - slope1*x - int1
  sumres <- sum(abs(newy))
  list(a=int1, b=slope1, sumres = sumres, res=newy)
}
```

Example 2: rrline1

```
rr1 <- rrline1(logbody, logbrain)

plot(logbody, logbrain, xlab="Log10(body weight)",
ylab="Log10(brain weight)", type="n", sub=
paste(paste("Intcpt=", format(round(rr1$a,4))),
      paste("Slope=",format(round(rr1$b,4)))))


text(logbody,logbrain,Animals.abb)
abline(rr1$a,rr1$b)

rr.bb <- run.rrline(logbody, logbrain)
abline(rr.bb$coef[6,1], rr.bb$coef[6,2], col=2, lty=2)
```

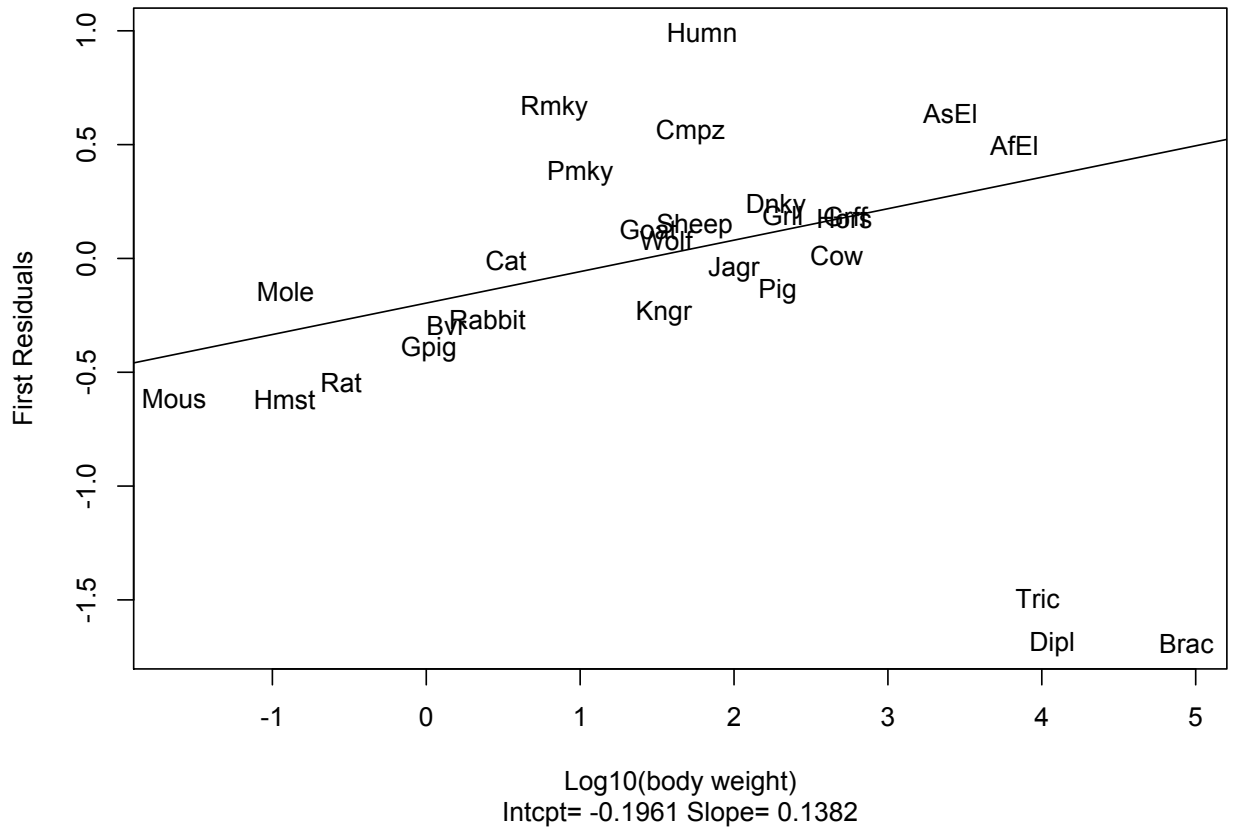
Example 2: rrline1



Example 2: rrline 2 fits first residuals

```
rr2 <- rrline1(logbody, rr1$res)

plot(logbody, rr1$res, xlab="Log10(body weight)",
ylab="First Residuals", type="n", sub=
paste(paste("Intcpt=", format(round(rr2$a,4))),
      paste("Slope=",format(round(rr2$b,4)))))

text(logbody,rr1$res,Animals.abb)
abline(rr2$a,rr2$b)
```

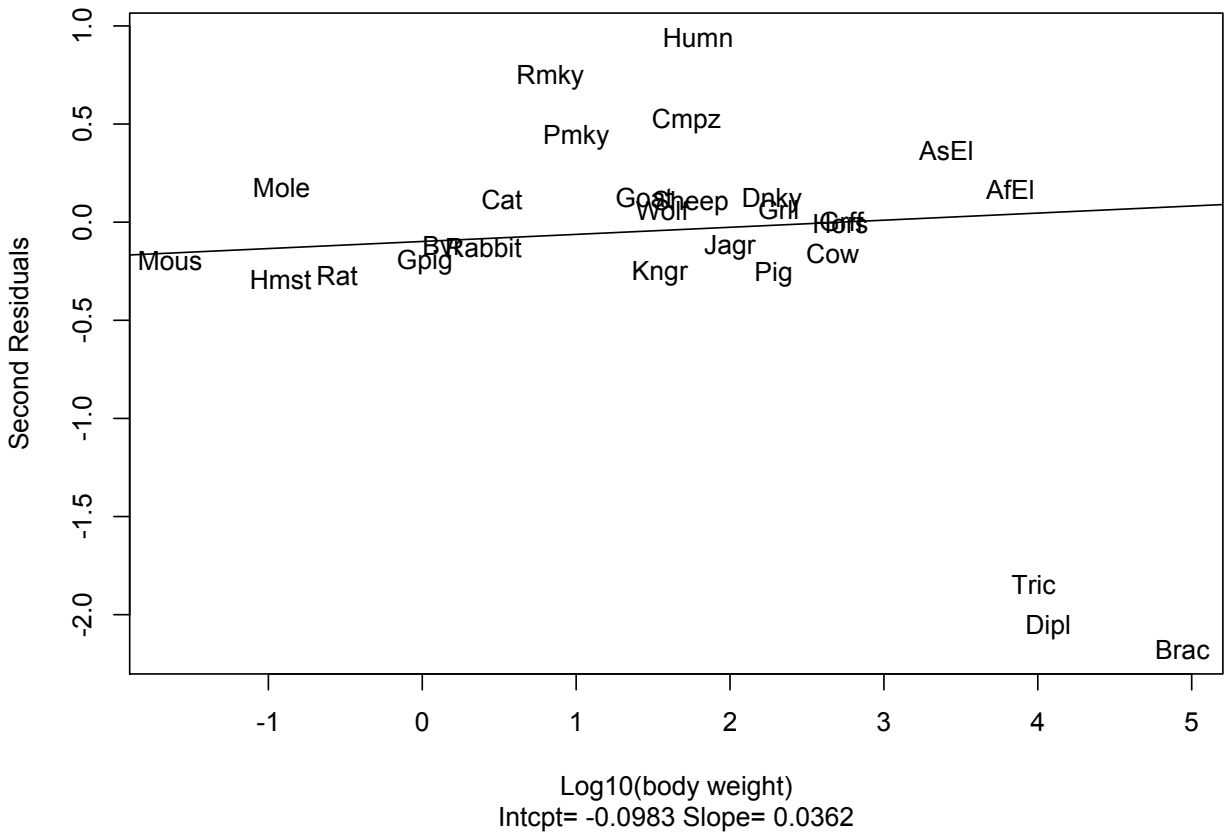
Example 2: rrline 2 fits first residuals



Example 2: rrline 3 fits second residuals

```
rr3 <- rrline1(logbody, rr2$res)

plot(logbody, rr2$res, xlab="Log10(body weight)",
ylab="Second Residuals", type="n", sub=
paste(paste("Intcpt=", format(round(rr3$a,4))),
      paste("Slope=",format(round(rr3$b,4)))))

text(logbody,rr2$res,Animals.abb)
abline(rr3$a,rr3$b)
```

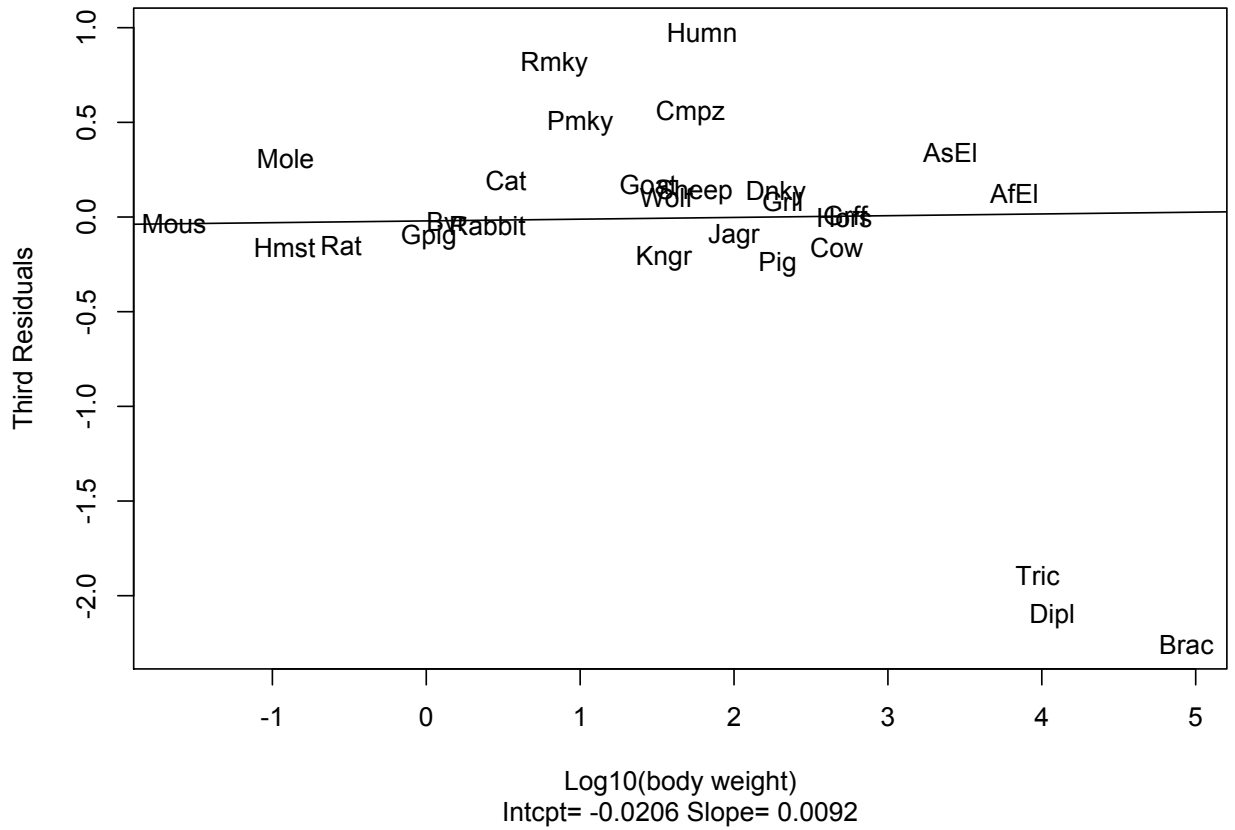
Example 2: rrline 3 fits second residuals



Example 2: rrline 4 fits third residuals

```
rr4 <- rrline1(logbody, rr3$res)

plot(logbody, rr3$res, xlab="Log10(body weight)",
ylab="Third Residuals", type="n", sub=
paste(paste("Intcpt=", format(round(rr4$a,4))),
      paste("Slope=",format(round(rr4$b,4)))))

text(logbody,rr3$res,Animals.abb)
abline(rr4$a,rr4$b)
```

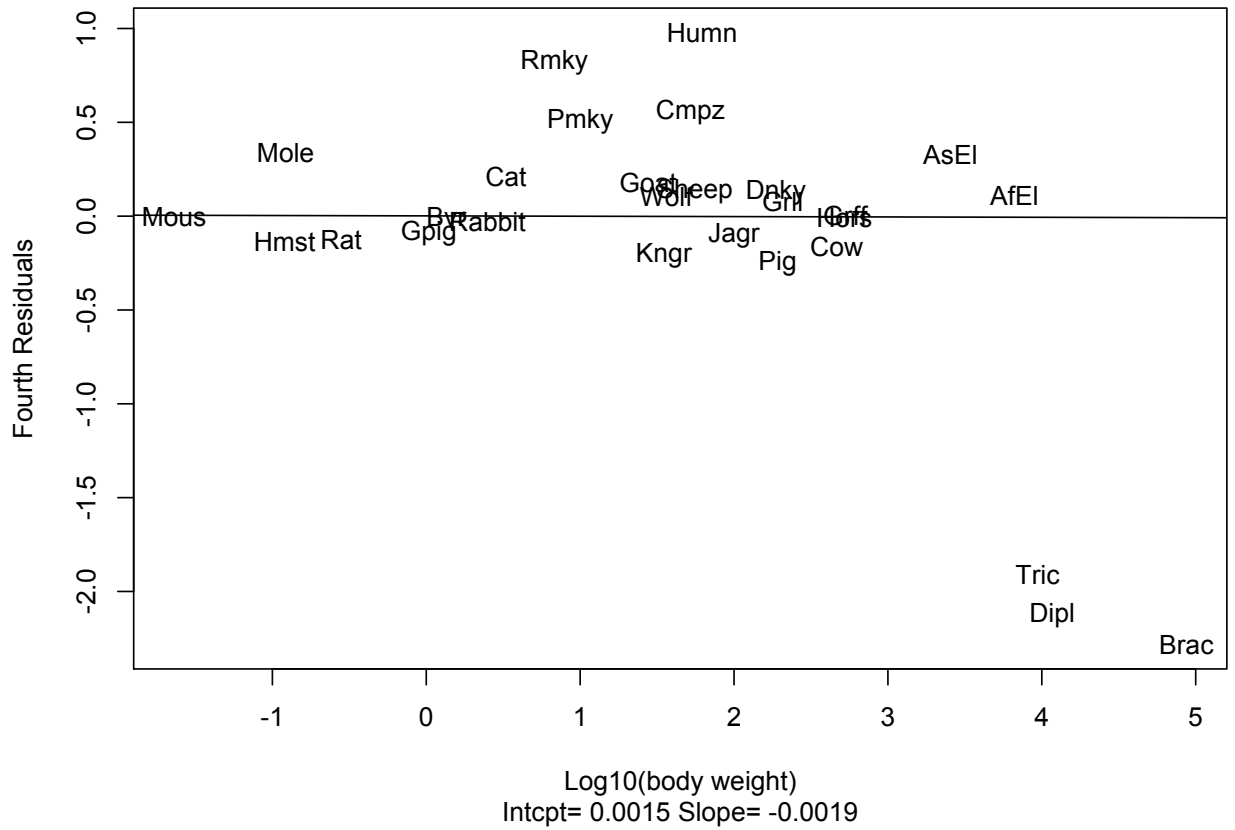
Example 2: rrline 4 fits third residuals



Example 2: rrline 5 fits fourth residuals

```
rr5 <- rrline1(logbody, rr4$res)

plot(logbody, rr4$res, xlab="Log10(body weight)",
ylab="Fourth Residuals", type="n", sub=
paste(paste("Intcpt=", format(round(rr5$a,4))),
      paste("Slope=",format(round(rr5$b,4)))))

text(logbody,rr4$res,Animals.abb)
abline(rr5$a,rr5$b)
```

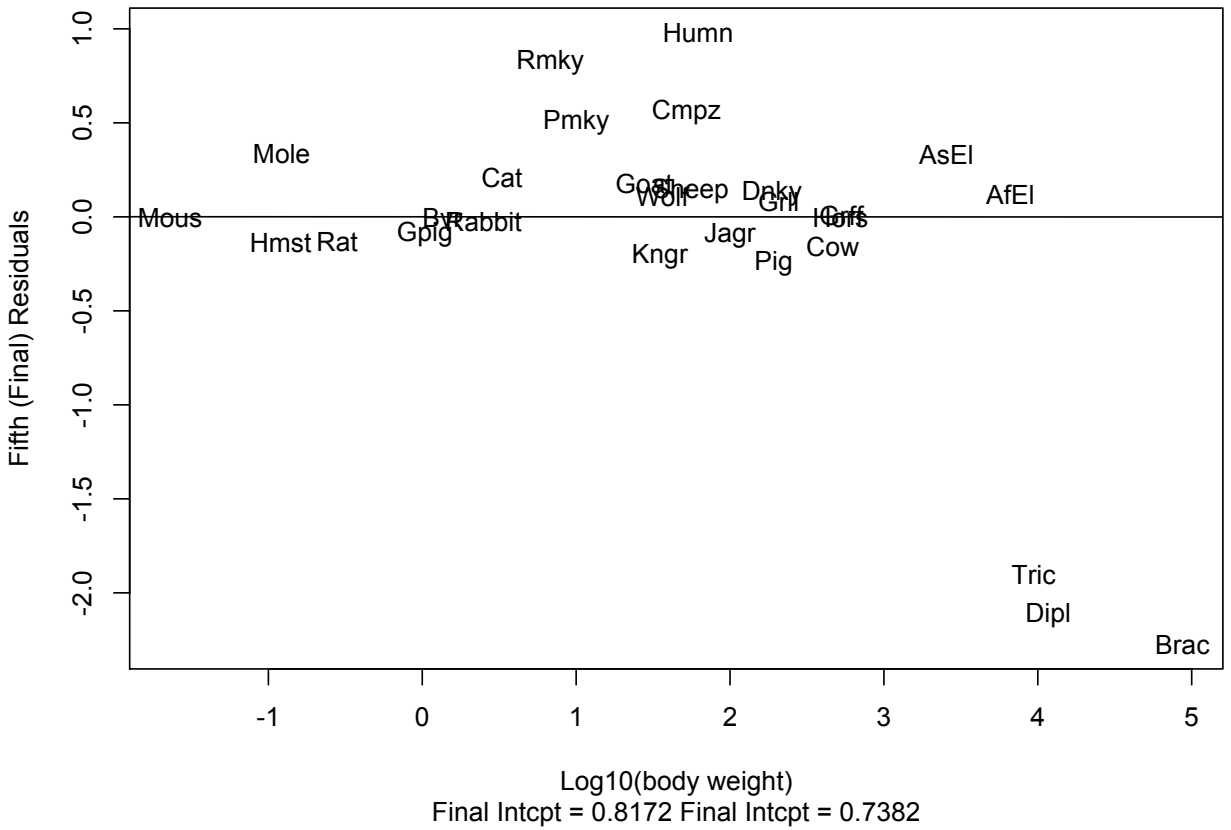
Example 2: rrline 5 fits fourth residuals



Example 2: final residuals

```
plot(logbody, rr5$res, xlab="Log10(body weight)",
   ylab="Fifth (Final) Residuals", type="n", sub=
   paste(
paste("Final Intcpt =",format(round(rr.bb$coef[6,1],4))),
paste("Final Intcpt =",format(round(rr.bb$coef[6,2],4)))))

text(logbody,rr.bb$res,Animals.abb)
abline(0.00027,-0.00019)
```

Example 2: final residuals



Example 2: rrline

```
run.rrline <- function(xx,yy,iter=5) {
  out.coef <- matrix(0,iter,3)
  l <- (1:length(xx))[!is.na(xx) & !is.na(yy)]; n <- length(l)
  x <- xx[l];  y <- yy[l]; newy <- y
  for (i in 1:iter) {
    rr <- rrline1(x,newy)
    out.coef[i,] <- c(rr$a,rr$b,rr$sumres)
    newy <- rr$res
  }
  dimnames(out.coef) <- list(format(1:iter),c("a","b","|res|"))
  aa <- sum(out.coef[,1])
  bb <- sum(out.coef[,2])
  cc <- sum(abs(y - aa - bb*x))
  res <- y - aa - bb*x
  out.coef <- rbind(out.coef,c(aa,bb,cc))
  print(round(out.coef,5))
  list(a = aa, b = bb, res = res, coef=out.coef)
}
```

Example 2: rrline, coefficients and residuals

```
run.rrline(logbody, logbrain)

$res
 [1] -0.00493 -0.15990  0.10842  0.17916 -0.08942 -2.12125
 [7]  0.33161  0.12783 -0.00645  0.50532  0.20828  0.00493
[13]  0.08174  0.98028  0.11750 -1.90495  0.82111 -0.20880
[19] -0.13748 -0.00582 -0.02817  0.13825 -0.09765  0.55856
[25] -0.13036 -2.27448  0.33434 -0.24740
$coef
              a            b    |res|
1  1.130794115  0.556500479 13.07018
2 -0.196111359  0.138201888 12.02777
3 -0.098333309  0.036150002 11.93348
4 -0.020620942  0.009217754 11.90944
5  0.001476957 -0.001899684 11.91439
   0.817205463  0.738170440 11.91439
```

Some notes

- Difference between book and above in estimating intercept!

$$a = \frac{1}{3}\left[(y_L - bx_L) + (y_M - bx_M) + (y_R - bx_R)\right]$$

$$a = \text{median}\{y_i - bx_i\}$$

Very similar performance – use whichever is easiest.

- RRline in transformation plots

Other Robust Lines

1. The Brown-Mood line: $\hat{y} = a_{BM} + b_{BM}x$
   - Two groups (spilt at median)
   - b_{BM} and a_{BM} are chosen to yield zero median residual in each of the two groups

$$\underset{x_i \leq M_x}{\text{med}} \{y_i - a_{BM} - b_{BM}x_i\} = 0$$

$$\underset{x_i > M_x}{\text{med}} \{y_i - a_{BM} - b_{BM}x_i\} = 0$$

   - Calculate b_{BM} using an iterative procedure

$$a_{BM} = \text{med}\{y_i - b_{BM}x_i\}$$

Other Robust Lines

2. Bartlett's method
   - 3 groups
   - 3 summary points: mean
   - $$b_B = \frac{\bar{y}_U - \bar{y}_L}{\bar{x}_U - \bar{x}_L}, \quad a_B = \bar{y} - b_B \bar{x}$$

Other Robust Lines

3. Wald's method
   - Two groups
   - 2 summary points: mean
   -

$$b_W = \frac{\bar{y}_U - \bar{y}_L}{\bar{x}_U - \bar{x}_L}, \quad a_W = \bar{y} - b_W \bar{x}$$

	Summary measure	
# groups	Mean	Median
2	Wald	Brown and Mood
3	Barlett	3-group RR line

Breakdown bound[1]: $LS = 0$, $RR = 1/6$

[1] The *breakdown bound* of a procedure for fitting a line to n pairs of y-versus-x data is k/n, where k is the greatest number of data points that can be replaced by arbitrary values while always leaving the slope and intercept bounded.

Other Robust Lines

4. Least absolute residuals (LAR)

$$\min \sum |y_i - a - bx_i|$$

- No explicit formulas for $\hat{\alpha}$, $\hat{\beta}$, solve computationally
- May not even be unique
- Not robust, but less sensitive to moderate disturbance than LS

Other Robust Lines

5. Median of pairwise slopes (Theil 1950)

$$\text{median } b_{ij} = (y_j - y_i)/(x_j - x_i), \quad 1 \le i < j \le n, \quad b_T = \text{med}\{b_{ij}\}$$

- $n(n-1)$ possible b_{ij}
- If k points are "wild" $\Rightarrow k(k-1)/2 + k(n-k)$ "wild" slopes (affected by 2 or 1 of k wild points)
- Need $k(k-1)/2 + k(n-1) < 0.5n(n-1)/2$ $(k/n \approx 0.29)$; Otherwise, b_T is wild

Other Robust Lines

6. Repeated Median Line

$$b_{RM} = \text{med}_i\{\text{med}_{j\neq i}\{b_{ij}\}\}$$

- For each point $\{x_i, y_i\}$, find $s_i \equiv$ median of all slopes through it
- Take median of s_i's as slope
- $a_{RM} = \text{median}_i\{y_i - b_{RM}x_i\}$
- Use each b_{ij} twice, think of a matrix of slopes (first find medians of rows, and then take median; # 5: median of the matrix)

$$\begin{pmatrix} & b_{12} & b_{13} & \cdots & b_{1n} \\ b_{21} & & b_{23} & \cdots & b_{2n} \\ \vdots & \ddots & \ddots & \cdots & \vdots \\ b_{n1} & \cdots & \cdots & \cdots & \end{pmatrix}$$

- $k \leq n/2$ wild points

Other Robust Lines

7. Least median of squares $b_{LMS} = \text{argmin}\{\text{median}(y_i - bx_i)\}$
   - minimize median, not mean/sum
   - very robust, breakdown $\approx 1/2$
   - computationally cumbersome
   - requires two sorts on n observations, $2O(n \log n)$ operations

Other Robust Lines

8. General M-estimation

$$b_M = \text{argmin} \sum_{i=1}^{n} \rho(y_i - \beta x_i)$$

- $\rho(\cdot) = (\cdot)^2 \Rightarrow$ LS too sensitive to outliers
- $\rho(\cdot) = |\cdot| \Rightarrow$ LAR too sensitive to the middle observations
- Huber estimator: quadratic in the center and linear in the tails (Ch11, P371)

$$\rho(u) = u^2 I_{[-k,k]} + (2ku\text{sign}(u) - k^2) I_{[k,\infty]}(|u|)$$

- Redescending estimators (or Hampels), very robust, breakdown: 0.5
- Find iteratively (w-iteration)

Resistance and efficiency

How to choose from available methods?

1. Degrees of resistance
2. Relative precision of the slope estimates
3. Increased resistance requires more computational effort (sort)

Announcements

- In-class Quiz Next Week (Monday or Wednesday?)
- Homework 4??
- Bring laptop next Monday, 9/28
- Quiz is on Lectures up to This Material.
- Get help:
  – Your own reference list (R functions posted on Canvas)
  – R built-in help system (help(lm), ?lm)
  – R reference card
  `http://cran.r-project.org/doc/contrib/Short-refcard.pdf`
- No discussion during quiz
- Quiz Submission: use Canvas
- Quiz Dataset sent out on Monday morning