Exploratory Data Analysis

Robust Regression and Median Polish

David B King, Ph.D.

October 5, 2015

Resistant Regression

Resistant Regression

Some Concepts on Influence

In Basic OLS Regression, the prediction $\hat{\mathbf{y}}$ is given by

$$\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{H}\mathbf{y}$$

with $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ denoting the hat matrix.

Leverage

The i^{th} diagonal element of $\mathbf{H}$, denoted h_{ii} is defined to be the leverage of an observation. If an observation has high leverage it has a large ability to change the regression parameters $\hat{\beta}$ and is generally thought of as an x outlier.

Bounds on Influence

The values of leverage are confined between 0 and 1.

$$0 \geq h_{ii} \leq 1$$

Proof: Clearly **H** is idempotent (and positive definite) since

$$\mathbf{HH} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{X})(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T = \mathbf{H}$$

Hence from matrix multiplication if I multiply the i^{th} row by the ith column of **H** this implies that

$$h_{ii} = \sum_{j=1}^{n} h_{ij}h_{ji} = h_{ii}^2 + \sum_{j \neq i} h_{ij}^2 \geq 0$$

and so

$$h_{ii} \geq h_{ii}^2 \implies 0 \leq h_{ii} \leq 1.$$

Pedagogical Example

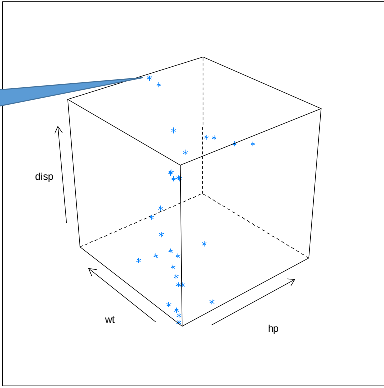
```
library(car)
fit = lm(mpg~disp+hp+wt+drat,data=mtcars)
```

Y function of 4 variables

Let’s tray to visualize the 4 dimensional X space:

```
library(car)
library(MASS)
library(lattice)
p1=cloud(disp~hp*wt,data=mtcars)
p2=cloud(disp~hp*drat,data=mtcars)
p3=cloud(hp~wt*drat,data=mtcars)
p4=cloud(hp~wt*disp,data=mtcars)

print(p1, position = c(0,0,.4,.4), more = TRUE)
print(p2, position = c(1,0,.6,.4), more = TRUE)
print(p3, position = c(1,1,.6,.6), more = TRUE)
print(p4, position = c(0,1,.4,.6))
```



X outliers are near edge of data cloud

Pedagogical Example

```
> X=model.matrix(fit)
> H=X%*%solve(t(X)%*%X)%*%t(X)
> h=diag(H)
> h
          Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4 Drive   Hornet Sportabout
         0.04601982          0.04985367          0.06740049          0.12333254          0.17158433
            Valiant          Duster 360           Merc 240D            Merc 230            Merc 280
         0.20140186          0.14458158          0.12645382          0.10701248          0.12873683
          Merc 280C          Merc 450SE          Merc 450SL         Merc 450SLC  Cadillac Fleetwood
         0.12873683          0.11611093          0.08400892          0.08623978          0.21951283
Lincoln Continental   Chrysler Imperial            Fiat 128         Honda Civic      Toyota Corolla
         0.22936687          0.23636908          0.08435766          0.31906469          0.10355543
      Toyota Corona    Dodge Challenger         AMC Javelin          Camaro Z28    Pontiac Firebird
         0.07049766          0.17041022          0.08905695          0.13753290          0.19562332
          Fiat X1-9       Porsche 914-2        Lotus Europa      Ford Pantera L        Ferrari Dino
         0.09294851          0.12737878          0.21234451          0.36351495          0.16823066
      Maserati Bora          Volvo 142E
         0.50833966          0.09042143
```

Observations With high leverage have more ability to affect the regression coefficients

The Effect of Leverage on Variance

The predicted value of $\mathbf{y}$ is given by

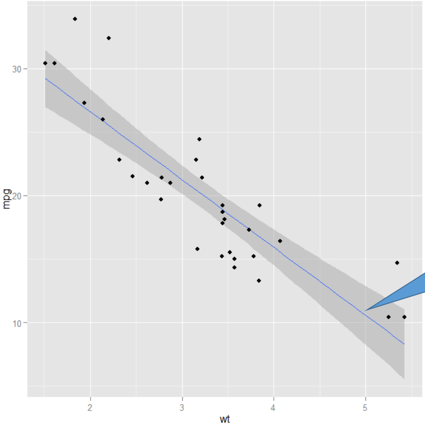
$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$$

hence

$$\begin{aligned} \mathrm{Var}[\hat{\mathbf{y}}] &= \mathbf{H}\mathrm{Var}[\mathbf{y}]\mathbf{H}^T = \sigma^2\mathbf{H} \\ \Longrightarrow \mathrm{Var}[\hat{\mathbf{y}}_i] &= \sigma^2 h_{ii} \end{aligned}$$

This implies that as the we get near the edges of the data uncertainty in the prediction grows!!

Illustration



Uncertainty in prediction grows near the edges of data

The Effect of Leverage on Variance

The effect of leverage on the variance of the residuals $\hat{\varepsilon}$ is just the opposite since

$$\hat{\varepsilon} = (\mathbf{I} - \mathbf{H})\mathbf{y}$$

hence

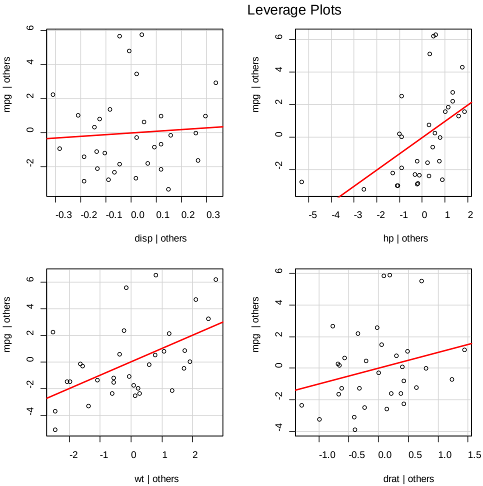
$$\begin{aligned} \mathrm{Var}[\hat{\varepsilon}] &= (\mathbf{I} - \mathbf{H})\mathrm{Var}[\hat{\varepsilon}](\mathbf{I} - \mathbf{H})^T = \sigma^2(\mathbf{I} - \mathbf{H}) \\ \Longrightarrow \mathrm{Var}[\hat{\varepsilon}_i] &= \sigma^2(1 - h_{ii}) \end{aligned}$$

This implies that as the we get near the edges of the data, the uncertainty in the residual decreases!!

Leverage Plots

leveragePlots(fit)

Leverage Plots can quantify each variables contribution to the overall leverage value



Influence vs Leverage

- **Leverage** measures a data points capacity or ability to change the direction of a regression line.
- But, just because you have the ability or capacity to do something doesn't mean you actually will do it!!
- **Influence** measures a data points actual effect on the direction of a regression line.
- How do we measure this? Answer: Take away the data point and observe how the regression line changes. (This is Jackknifing).

Influence vs Leverage

Let $\hat{Y}_{i(j)}$ denote the prediction at point i given that data point j is removed from the data set.

Cook's Distance

Cook's distance measures the effect of deleting a given observation and it is calculated as

$$D_i = \frac{\sum_{j=1}^{n}(\hat{Y}_j - \hat{Y}_{j(i)})}{pMSE} = \frac{(\hat{\epsilon}_i^2)}{pMSE}\left[\frac{h_{ii}}{(1-h_{ii}^2)}\right]$$

where p is the number of variables in regression, MSE is the mean squared error.

- Cook's distance is a measure of **influence** and measures how much the prediction would change if we we removed the observation.
- D_i can be interpreted as the distance one's estimates move within the confidence ellipsoid of the parameters. if an observation is removed from the data.

What data points are influential?

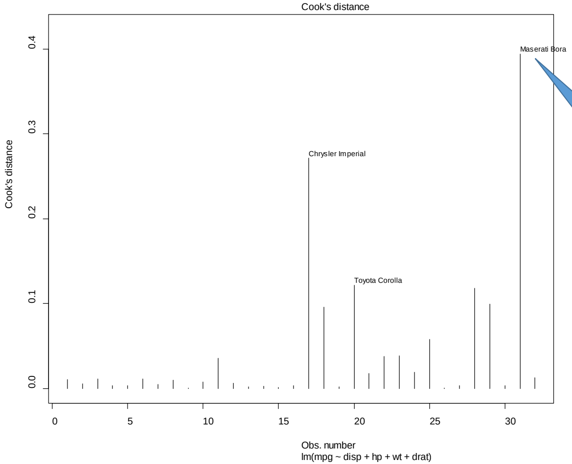
We know that asymptotically as $n \to \infty$

$$\boxed{D_i \sim F_{p,n-p}}$$

hence a conservative cutoff is qf$(0.95, p, n-p)$, others have suggested a more modest $4/(n-p+1)$.

```
conservativecutoff = qf(0.95,length(fit$coefficients)-1,
nrow(mtcars)-length(fit$coefficients)+1)
cutoff <- 4/((nrow(mtcars)-length(fit$coefficients)-2))
plot(fit, which=4, cook.levels=cutoff)
```

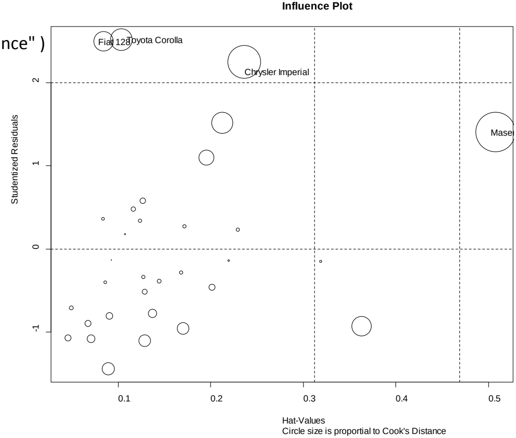
Influence Example



Beware of points that are both influential and have high leverage

Influence Plots

```
influencePlot(fit, id.method="identify",
main="Influence Plot",
sub="Circle size is proportial to Cook's Distance" )
```



With an influence plot you can see

h_{ii} =influence on x-axis
$t_i = e_i/s(1-h_{ii})$ studentized residuals on y-axis

Size of the circle is Cook's Distance D_i

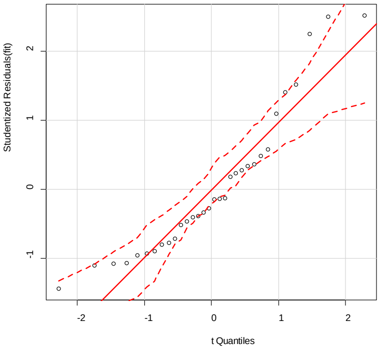
Outlier Test Example

Outlier and Normality Plots

```
> outlierTest(fit)

No Studentized residuals with Bonferonni p < 0.05
Largest |rstudent|:
           rstudent unadjusted p-value Bonferonni p
Toyota Corolla  2.51597          0.01838      0.58816

qqPlot(fit, main="QQ Plot") #qq plot for studentized resid
```



Resistant Regression

What do you do about influential outliers?
Modern Methods of Resistant Regression

Iteratively Reweighted LS

Robust regression: “iteratively re-weighted LS”

$$y = x\beta + \epsilon = \mathit{fit} + \mathit{rough}$$

$$\hat{\beta} = (X'X)^{-1}X'y, \quad \text{linear function of } y$$

weighted LS: $\hat{\beta}_W = (X'WX)^{-1}X'Wy$, linear function of y, where weight matrix $W = \mathrm{diag}(w_1, \cdots, w_n)$.
w_i = Big if residual is small (point i is “consistent” with model”)
w_i = Small if residual is large (point i is “inconsistent” with model)
S_r= measure of spread in residuals
Big residual if $|\mathit{resid}| > 6S_r$

$$(*) \quad w_i = \begin{cases} 1 - (\hat{\epsilon}_i/(6S_r)^2)^3, & \text{if } |\hat{\epsilon}_i| \le 6S_r \\ 0 & \text{else} \end{cases}$$

Iteratively Reweighted LS

Iteratively re-weighted LS

1. Fit $OLS \rightarrow \hat{\beta}^{(0)} \rightarrow r^{(0)} = y - X\hat{\beta}^{(0)}$
2. Calculate $W^{(1)}$ using $r_i^{(0)}$ (weights from (*))
3. Fit $\hat{\beta}^{(1)}$ vis WLS $\rightarrow \hat{\beta}^{(1)} \rightarrow r^{(1)} = y - X\hat{\beta}^{(1)}$
4. Calculate $W^{(2)}$ using $r_i^{(1)}$ (weights from (*))
5. Fit $\hat{\beta}^{(2)}$ via WLS $\rightarrow \hat{\beta}^{(2)} \rightarrow r^{(2)} = y - X\hat{\beta}^{(2)}$
6. Until $|\hat{\beta}^{(k+1)} - \hat{\beta}^{(k)}| < tolerance$

Iteratively Reweighted LS

Notes on robust regression:

1. Starting with OLS is not robust
2. Eqn (*) is called bisque weight function
3. The "6" is a "tuning parameter" often b/w 4–6
4. Any weight function that falls smoothly to 0 yields robust estimate of β. Biweight regression is especially efficient
5. Easy to do computationally. See library (MASS), help (rlm).

Iteratively Reweighted LS

Example of IRLS:

```
IRWLS<-function(data,epsilon=10^{-6}){
n = nrow(data);
w = rep(1,n); # weights set equal to 1 initially
eps = 1; loops = 0;
while(eps>= epsilon){
if(loops > 0){betaold=betahat; w=1/residfit$fitted;}
fit = lm(y~x,data=data,weights=w);betahat = fit$coef;
resid=fit$residuals;resid2 = resid^2;
residfit = lm(resid2~x,data=data);gamma=residfit$coef;
if(loops>0){eps=norm(betahat-betaold,type="2")}
loops=loops+1;
}
output=list(fit=fit,residfit=residfit,loops=loops);
}
```

Some Theory of M estimation

It is well known that we compute the least squares estimator β by minimizing

$$\beta = \arg\min \left[\sum_{i=1}^{n} \rho \left(\frac{(y_i - \mathbf{x}_i^T \beta)}{\sigma} \right) \right]$$

where for OLS regression we choose $\rho(u) = u^2$.
But other choices for $\rho(\cdot)$ are possible. For example we define Minimum Absolute Deviation (MAD) to be

$$\beta = \arg\min \left[\sum_{i=1}^{n} \left| \frac{(y_i - \mathbf{x}_i^T \beta)}{\sigma} \right| \right]$$

so the choice for MAD is $\rho(u) = |u|$.

Some Theory of M estimation

- The theory of M-estimation concerns itself with what happens to regression if we change the way we penalize the residuals for arbitrary $\rho(\cdot)$.
- Certain choices of ρ lead to more robust regression methods.(see UREDA Ch. 11)

M estimator

The *M*-estimator $T_n(x_1, \ldots, x_n)$ for a function ρ and the sample $x_1, \ldots, x_n$ is the value of t which minimizes some objective function

$$\sum_{i=1}^{n} \rho(x_i, t).$$

M estimator

If the derivative

$$\psi(x,t) = \frac{\partial \rho(x_i, t)}{\partial t}$$

is known then the M-estimator $T_n(x_1, \ldots x_n)$ can be finding the value of t that satisfies the score equation

$$\sum_{i=1}^{n} \psi(x_i, t) = 0$$

Bi-Weight Function

Different objective functions have been chosen for robust regression purposes. For example, one such function is the bi-weight, or bi-square function.

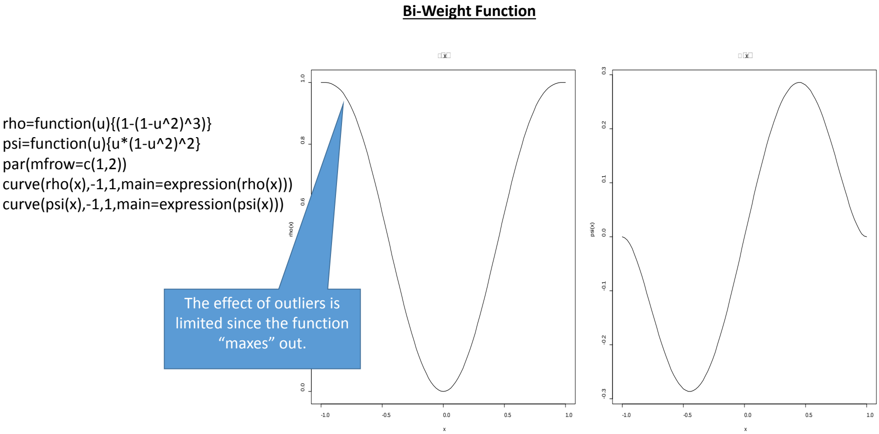
$$\rho(u) = \begin{cases} \frac{1}{6}\left[1-(1-u^2)^3\right] & |u| \leq 1 \\ \frac{1}{6} & |u| \geq 1 \end{cases}$$

$$\psi(u) = \begin{cases} u(1-u^2)^2 & |u| \leq 1 \\ 0 & |u| \geq 1 \end{cases}$$

The M estimator is found through solution to score equations

$$\sum_{i=1}^{n} \psi(u_i) = 0 \text{ with } u_i = \frac{(x_i - t)}{\hat{\sigma}}.$$

Bi-Weight Function



Bi-Weight Function

```
rho=function(u){(1-(1-u^2)^3)}
psi=function(u){u*(1-u^2)^2}
par(mfrow=c(1,2))
curve(rho(x),-1,1,main=expression(rho(x)))
curve(psi(x),-1,1,main=expression(psi(x)))
```

The effect of outliers is limited since the function “maxes” out.

Huber's Loss Function

Huber's Loss Function is another popular method of minimizing the effect of outliers and influential data. For some user defined k we define

$$\rho_k(u) = \begin{cases} \frac{1}{2}u^2 & |u| \leq k \\ k(|u| - \frac{1}{2}k) & |u| \geq k \end{cases}$$

$$\psi_k(u) = \max(-k, \min(k, u)).$$

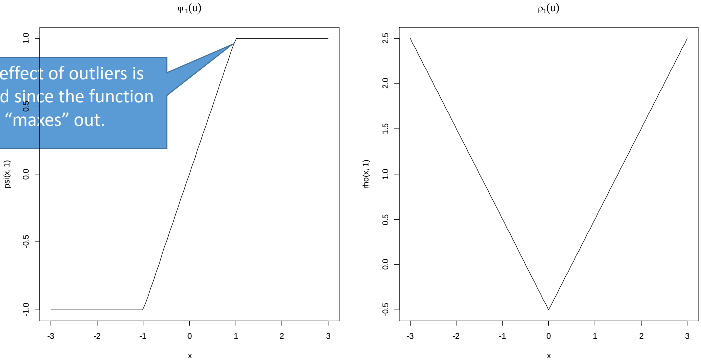
The M estimator is found through solution to score equations

$$\sum_{i=1}^{n} \psi_k(u_i) = 0 \text{ with } u_i = \frac{(x_i - t)}{\hat{\sigma}}.$$

Huber's Loss Function

Huber's Loss Function

```
rho=function(u,k){if(abs(u)<=k){u^2/2}else{k*(abs(u)-k/2)}}
psi=function(u,k){pmax(-k,pmin(u,k))}
par(mfrow=c(1,2))
curve(psi(x,1),-3,3,main=expression(psi[1](u)))
curve(rho(x,1),-3,3,main=expression(rho[1](u)))
```



The effect of outliers is limited since the function "maxes" out.

RLM Regression in R

In the MASS package the robust regression methods rlm() uses the **Loss function** $\rho(u)$ **to adjust the weights** and then uses iterative reweighted least squares methods. Let's practice:

```
library(foreign)
cdata <- read.dta("http://www.ats.ucla.edu/stat/data/crime.dta")
summary(ols <- lm(crime ~ poverty + single, data = cdata))
d1 <- cooks.distance(ols)
r <- stdres(ols)
a <- cbind(cdata, d1, r)
a[d1 > 4/51, ]
```

RLM Regression in R

Let's practice:

```
rabs <- abs(r)
a <- cbind(cdata, d1, r, rabs)
asorted <- a[order(-rabs), ]
asorted[1:10, ]
summary(rr.huber <- rlm(crime ~ poverty + single, data = cdata))
hweights <- data.frame(state = cdata$state,
resid = rr.huber$resid, weight = rr.huber$w)
hweights2 <- hweights[order(rr.huber$w), ]
hweights2[1:15, ]
rr.bisquare <- rlm(crime ~ poverty + single,
data=cdata, psi = psi.bisquare)
summary(rr.bisquare)
```

Median Polish

Analysis of Two-Way Tables by Median Polish
Chapter 6 of UREDA and Chapter 3 of EDTTS

Median Polish

1. Two-way analysis: Example
2. Morals: Median Polish vs Two-way analysis by means
3. Examples
4. Diagnostic plot
5. Improving the fit

Example (EDTTS table 7-4, page 267)

Data on yield of six varieties (A, B, C, D, E, F) of corn in four randomized blocks (1, 2, 3, 4).

Response variable: the number of plants.

	1	2	3	4
A	28	22	27	19
B	23	26	28	24
C	27	24	27	28
D	24	28	30	30
E	30	26	26	29
F	30	25	27	24

	1	2	3	4
A	288	22	27	19
B	23	26	28	24
C	27	24	27	28
D	24	28	30	30
E	30	26	26	29
F	30	25	27	24

Tukey's Median Polish Procedure

Want to fit tabular data to the model

$$y_{ij} = m + a_i + b_j + \epsilon_{ij}$$

So we set up a iterative procedure and denote the fit and residuals at the end of n iterations by

$$y_{ij} = m^{(n)} + a_i^{(n)} + b_j^{(n)} + \epsilon_{ij}^{(n)}$$

The initial conditions before the first iteration $(n = 0)$ are

$$\begin{aligned} m^{(n)} &= 0 \\ a_i^{(n)} &= 0 \quad i = 1, \ldots, R \\ b_j^{(n)} &= 0 \quad j = 1, \ldots, C \\ \epsilon_{ij}^{(0)} &= y_{ij} \end{aligned}$$

Tukey's Median Polish Procedure

The steps to the iteration procedure go as follows:

Rows Update:

$$\begin{aligned}\Delta a_i^{(n)} &= \text{med}\{\epsilon_{ij}^{(n-1)} | j = 1, \ldots, C\} \text{ row median for } i = 1, \ldots, R \\ \Delta m_b^{(n)} &= \text{med}\{b_j^{(n-1)} | j = 1, \ldots, C\} \\ d_{ij}^{(n)} &= \epsilon_{ij}^{(n-1)} - \Delta a_i^{(n)} \text{ for } i = 1, \ldots, R \text{ and } j = 1, \ldots, C\end{aligned}$$

Column Update:

$$\begin{aligned}\Delta b_j^{(n)} &= \text{med}\{d_{ij}^{(n-1)} | i = 1, \ldots, R\} \text{ column median for } j = 1, \ldots, C \\ \Delta m_a^{(n)} &= \text{med}\{a_i^{(n-1)} + \Delta a_i^{(n-1)} | i = 1, \ldots, R\} \\ \epsilon_{ij}^{(n)} &= d_{ij}^{(n-1)} - \Delta b_j^{(n)} \text{ for } i = 1, \ldots, R \text{ and } j = 1, \ldots, C\end{aligned}$$

Common Value and Effects Update:

$$\begin{aligned}m^{(n)} &= m^{(n-1)} + \Delta m_a^{(n)} + \Delta m_b^{(n)} \\ a_i^{(n)} &= a_i^{(n-1)} + \Delta a_i^{(n-1)} - \Delta m_a^{(n)} \text{ for } i = 1, \ldots, R \\ b_j^{(n)} &= b_j^{(n-1)} + \Delta b_j^{(n-1)} - \Delta m_b^{(n)} \text{ for } j = 1, \ldots, C.\end{aligned}$$

Row Median Polish at Iteration n

After the row values are updated on iteration n the table looks like:

Row	Column j			New	Previous
i	1	$\cdots$	C	Median	Median
1	$\epsilon_{11}^{(n-1)}$	$\cdots$	$\epsilon_{1C}^{(n-1)}$	$\Delta a_1^{(n)}$	$a_1^{(n-1)}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
R	$\epsilon_{R1}^{(n-1)}$	$\cdots$	$\epsilon_{RC}^{(n-1)}$	$\Delta a_R^{(n)}$	$a_R^{(n-1)}$
previous	$b_1^{(n-1)}$	$\cdots$	$b_C^{(n-1)}$	$\Delta m_b^{(n)}$	$m^{(n-1)}$

Column Median Polish at Iteration n

After the column values are updated on iteration n using the previous table the new table looks like:

Row	Column j			Previous Median
i	1	$\cdots$	C	Median
1	$d_{11}^{(n)}$	$\cdots$	$d_{1C}^{(n)}$	$a_1^{(n-1)} + \Delta a_1^{(n)}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
R	$d_{R1}^{(n)}$	$\cdots$	$d_{RC}^{(n)}$	$a_R^{(n-1)} + \Delta a_R^{(n)}$
New Median	$\Delta b_1^{(n)}$	$\cdots$	$\Delta b_1^{(n)}$	$\Delta m_a^{(n)}$
Previous Median	$b_1^{(n-1)} - \Delta m_b^{(n)}$	$\cdots$	$b_C^{(n-1)} - \Delta m_b^{(n)}$	$m^{(n-1)} + \Delta m_b^{(n)}$

Example (EDTTS table 7-4, page 267)

```
# R function for LS analysis:
twoway.mean <- function(mat){
  meff.LS <- mean(mat)
  aeff.LS <- apply(mat,1,mean,na.rm=T) - meff.LS #row
   beff.LS <- apply(mat,2,mean,na.rm=T) - meff.LS #column
   res.LS <- mat - meff.LS -
    matrix(rep(aeff.LS, ncol(mat)), byrow=F,ncol=ncol(mat)) -
    matrix(rep(beff.LS, nrow(mat)), byrow=T,ncol=ncol(mat))
  list(overall=meff.LS, row=aeff.LS, col=beff.LS, res=res.LS)
}
# LS of both
tbl74.LS <- twoway.mean(tbl74)
tbl74.out.LS <- twoway.mean(tbl74.out)
```

Example (EDTTS table 7-4, page 267)

After 1 iteration:

	1	2	3	4	Row
A	3.3	-0.8	1.8	-4.3	-2.3
B	-2.9	1.9	1.6	-0.6	-1.1
C	-0.2	-1.3	-0.7	2.2	0.2
D	-4.7	1.2	0.8	2.7	1.7
E	1.6	-0.6	-2.9	1.9	1.4
F	2.8	-0.3	-0.7	-1.8	0.2
Col	0.7	-1.2	1.2	-0.7	26.3

	1	2	3	4	Row
A	165.8	-55.0	-52.3	-58.5	51.8
B	-35.4	12.7	12.4	10.2	-11.9
C	-32.7	9.5	10.2	13.0	-10.7
D	-37.2	12.0	11.7	13.5	-9.2
E	-30.9	10.2	7.9	12.7	-9.4
F	-29.7	10.5	10.2	9.0	-10.7
Col	33.2	-12.0	-9.7	-11.5	37.2

Example (EDTTS table 7-4, page 267)

```
# medpol of both
tbl74.MP <- medpolish(tbl74)
tbl74.out.MP <- medpolish(tbl74.out)
```

```
       1    2    3    4  Row
A    1.6 -1.2  1.2 -5.3 -1.2
B   -4.3  1.8  1.3 -1.2 -0.3
C   -0.4 -0.2  0.2  2.7 -0.2
D   -6.9  0.2 -0.2  1.2  3.2
E    0.4 -0.5 -3.0  1.5  2.0
F    2.1  0.2 -0.2 -1.8  0.2
Col  1.3 -1.8  0.7 -0.8 26.3
```

```
         1    2    3    4  Row
A    261.8 -1.1  1.4 -5.3 -1.4
B     -4.1  2.0  1.5 -1.2 -0.5
C     -0.4 -0.4  0.1  2.4 -0.1
D     -6.7  0.4 -0.1  1.2  3.1
E      0.4 -0.5 -3.0  1.3  2.0
F      2.3  0.4 -0.1 -1.8  0.1
Col    1.1 -2.0  0.5 -0.8 26.5
```

Example (EDTTS table 7-4, page 267)

Compare row effects:

```
                    m       A       B       C      D      E      F
LS: original   26.333  -2.333  -1.083   0.167  1.667  1.417   0.167
LS: outlier    37.167  51.833 -11.917 -10.667 -9.167 -9.417 -10.667

MP: original   26.344  -1.250  -0.344  -0.250  3.250  1.969   0.250
MP: outlier    26.500  -1.375  -0.500  -0.125  3.125  2.000   0.125
```

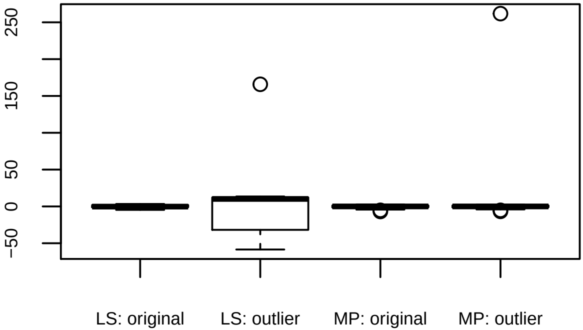
Compare column effects:

```
                    1       2      3       4
LS: original    0.667  -1.167  1.167  -0.667
LS: outlier    33.167 -12.000 -9.667 -11.500

MP: original    1.297  -1.844  0.656  -0.797
MP: outlier     1.062  -2.000  0.500  -0.812
```

Example (EDTTS table 7-4, page 267)

Compare residuals:



Morals

Corn yield data:

1. Median polish is more robust than classical ANOVA.
2. A single outlier can spread throughout a two-way table in a means analysis.
3. Always look at residuals (stem-and-leaf, boxplot).
4. Means analysis: mean(effects) = mean(residuals)=0
   Median polish: median(effects)=median(residuals)=0
5. Means analysis requires no iteration;
   medpolish does.

Median Polish: Temperature Example

Average Monthly Temperatures ×10 for 3 cities:

	Jan	Feb	Mar	Apr	May	Jun	Jul
Caribou	87	98	217	347	485	584	640
Washington	362	371	453	544	647	734	773
Laredo	576	619	684	759	812	858	877

Calculate row medians:

	Jan	Feb	Mar	Apr	May	Jun	Jul	Eff
Caribou	87	98	217	347	485	584	640	347
Washington	362	371	453	544	647	734	773	544
Laredo	576	619	684	759	812	858	877	759

Median Polish: Temperature Example

Subtract row medians;

	Jan	Feb	Mar	Apr	May	Jun	Jul	Eff
Caribou	-260	-249	-130	0	138	237	293	347
Washington	-182	-173	-91	0	103	190	229	544
Laredo	-183	-140	-75	0	53	99	118	759

Calculate column medians:

	Jan	Feb	Mar	Apr	May	Jun	Jul	Eff
Caribou	-260	-249	-130	0	138	237	293	347
Washington	-182	-173	-91	0	103	190	229	544
Laredo	-183	-140	-75	0	53	99	118	759
Col Eff	-183	-173	-91	0	103	190	229	544

Median Polish: Temperature Example

Subtract column medians:

	Jan	Feb	Mar	Apr	May	Jun	Jul	Eff
Caribou	-77	-76	-39	0	35	47	64	-197
Washington	1	0	0	0	0	0	0	0
Laredo	0	33	16	0	-50	-91	-111	215
Col Eff	-183	-173	-91	0	103	190	229	544

Slight difference between starting
with rows or with columns; usually unimportant. See UREDA, p.185.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Eff
Caribou	-78	-76	-39	0	35	47	64	-197
Washington	0	0	0	0	0	0	0	0
Laredo	0	34	17	1	-49	-90	-110	214
Col Eff	-182	-173	-91	0	103	190	229	544

Median Polish: Temperature Example

“Almost as good”:

(2)	Jan	Feb	Mar	Apr	May	Jun	Jul	Eff
Caribou	-63	-102	-33	-3	35	34	40	-200
Washington	12	-29	3	-6	-3	-16	-27	0
Laredo	26	19	34	9	-38	-92	-123	200
Col Eff	-200	-150	-100	0	100	200	250	550

Original temperature units (patterns in residuals?):

	Jan	Feb	Mar	Apr	May	Jun	Jul	Eff
C	-6.3	-10.2	-3.3	-0.3	3.5	3.4	4.0	-20
W	1.2	-2.9	0.3	-0.6	-0.3	-1.6	-2.7	0
L	2.6	1.9	3.4	0.9	-3.8	-9.2	-12.3	20
Eff	-20	-15	-10	0	10	20	25	55

Median Polish (R^2) Temperature Example

Note: In classical analysis of variance,

- SS(row) $= C\sum_{i=1}^{R}(\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot})^2$
- SS(column) $= R\sum_{j=1}^{C}(\bar{y}_{\cdot j} - \bar{y}_{\cdot\cdot})^2$
- SS(resid) $= \sum_{j=1}^{C}\sum_{i=1}^{R}(y_{ij} - \bar{y}_{i\cdot} - \bar{y}_{\cdot j} + \bar{y}_{\cdot\cdot})^2$
- SS(total) $= \sum_{j=1}^{C}\sum_{i=1}^{R}(y_{ij} - \bar{y}_{\cdot\cdot})^2$

Assessing quality of fit:

- Classical ANOVA $R^2 = 1 - \frac{\text{SS(resid)}}{\text{SS(resid)}}$
- Analog: $R^2 = 1 - \frac{\sum_{ij}|r_{ij}|}{\sum_{ij}|y_{ij} - \tilde{y}_{\cdot\cdot}|}$
- Temperature data: Analog $R^2 = 1–640/3863 = 83.4\%$

Two-way plot

Fitted values: $\widehat{y}_{ij} = m + a_i + b_j$

	Jan	Feb	Mar	Apr	May	Jun	Jul
Caribou	15	20	25	35	45	55	60
Washington	35	40	45	55	65	75	80
Laredo	55	60	65	75	85	95	100

Plotting the fit:

- Show relative effects of rows (e.g., cities)
- Show relative effects of columns (e.g., months)
- Show approximate fitted value $\hat{y}_{ij}$

Process:

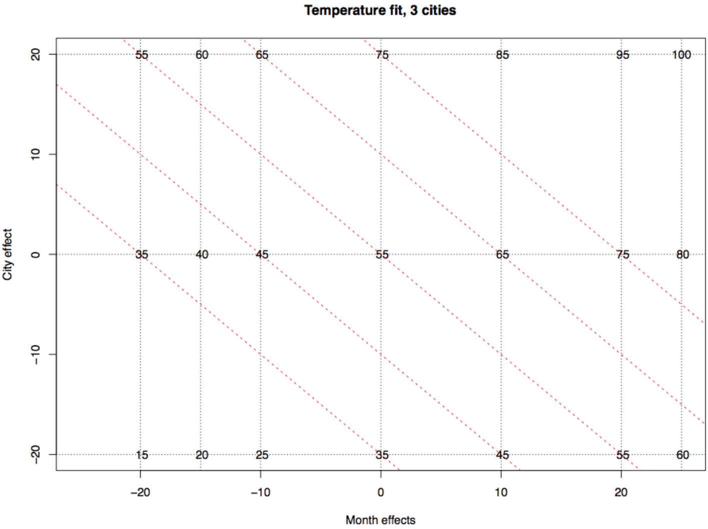
- Mark off on horizontal axis: row effects
- Mark off on vertical axis: column effects
- Rotate the page -45°
- Place y-axis corresponding to fitted values
- vertical line segments for residuals

Two-way plot

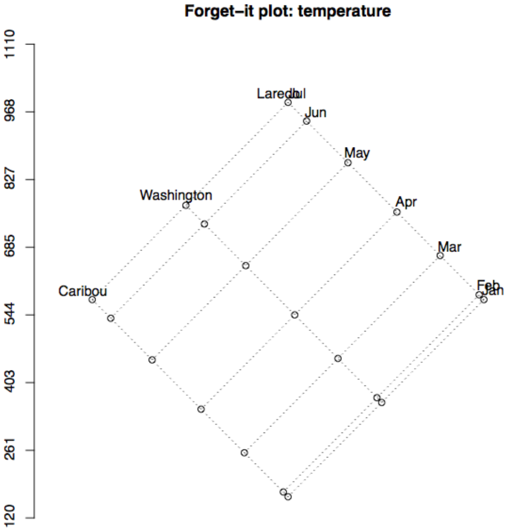
Example: *Forget-it Plot* on “Almost as good” fit:

- Draw horizontal line and mark off effects for columns
  Cols: -20, -15, -10, 0, 10, 20, 25
- Draw vertical line and mark off effects for rows
  Rows: -20, 0, 20
- Draw horizontal/vertical lines for each row/col effect
- At each intersection, write fitted value $= m + a_i + b_j$
- Rotate plot by 45°, place “fit” scale

Forget-it plot



Forget-it plot



Median Polish: Example 2

Data: Olympic distance winning times (seconds, ×10)

	1948	1952	1956	1960	1964	1968	1972
100m	103	104	105	102	100	99	101
200m	211	207	206	205	203	198	200
400m	462	459	467	449	451	438	447
800m	1092	1092	1077	1063	1051	1043	1059
1500m	2298	2252	2212	2156	2181	2149	2163

```
run.MP <- medpolish(run)
```

Median Polish: Example 2

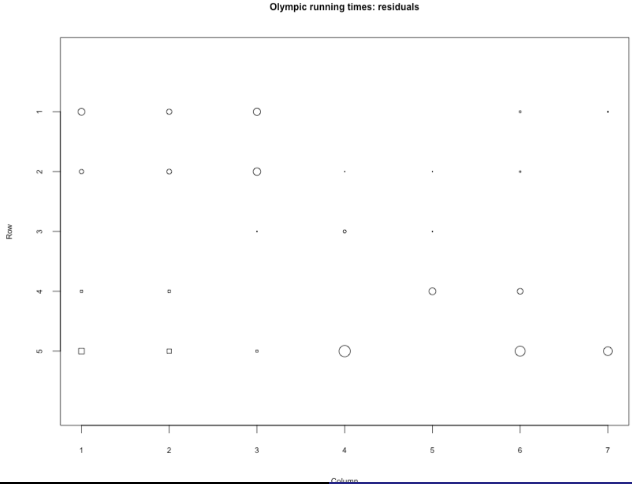
Median Polish:

	1948	1952	1956	1960	1964	1968	1972	Row
100m	-10	-6	-11	0	0	10	3	-349
200m	-4	-5	-12	1	1	7	0	-247
400m	0	0	2	-2	2	0	0	0
800m	18	21	0	0	-10	-7	9	612
1500m	104	61	15	-27	0	-21	-16	1732
Col	11	8	14	0	-2	-13	-4	451

$$R^2 = 1 - \sum |res| / \sum |data - 451| = 1 - 376/20798 = 0.982$$

Example 2: Plot residuals

Symbols plot of $-/+$ residuals: circle $(-)$ radii or square $(+)$ sides $\propto \sqrt{|res|}$



Median Polish: Diagnosing patterns in residuals

Model for y_{ij} is $\hat{y}_{ij} = m + a_i + b_j$

Suppose residuals show structure

And suppose that data fit this model perfectly:

$$y_{ij} = m + a_i + b_j + (a_i b_j)/m \quad (1)$$

Algebra: (1) can be written as

$$y_{ij} = m(1 + a_i/m)(1 + b_j/m) \quad (2)$$

What does (2) suggest?

Median Polish: Diagnosing patterns in residuals

$$y_{ij} = m(1 + a_i/m)(1 + b_j/m) \quad (2)$$

(2) is a product of three terms:

- m (no i,j)
- $1 + a_i/m$ (involves only row i, no columns)
- $1 + b_j/m$ (involves only col j, no rows)

How can we change y_{ij} from a *multiplicative* model to an *additive* model?

Median Polish: Transformation

$$y_{ij} = m + a_i + b_j + (a_i b_j)/m$$

Plot **residual** res_{ij} vs **comparison value** $cv_{ij} \equiv a_i b_j / m$.

If the plot has slope —, then —

Do this for Olympic Run data

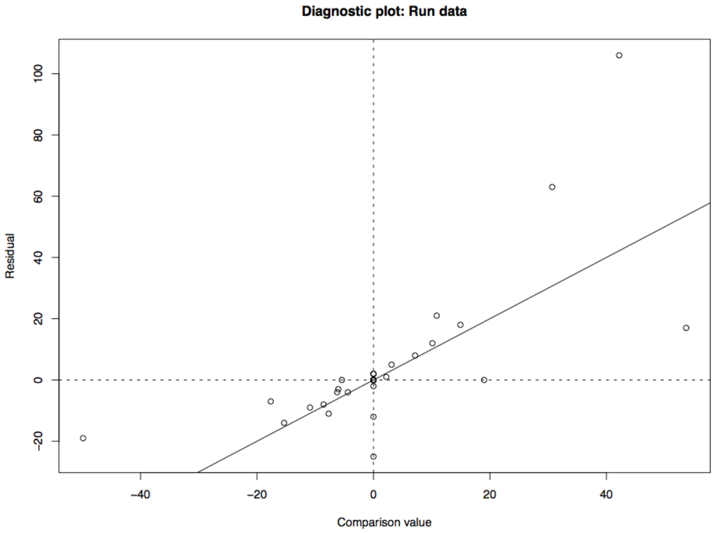
Median Polish: Diagnostic plot

Matrix of comparison values:$cv_{ij} = a_i b_j / m$

	1948	1952	1956	1960	1964	1968	1972
100m	-8.51	-6.19	-10.83	0	1.55	10.06	3.10
200m	-6.02	-4.38	-7.67	0	1.10	7.12	2.19
400m	0.00	0.00	0.00	0	0.00	0.00	0.00
800m	14.93	10.86	19.00	0	-2.71	-17.64	-5.43
1500m	42.24	30.72	53.76	0	-7.68	-49.92	-15.36

Plot res_{ij} (y-axis) vs cv_{ij} (x-axis)

Median Polish: Diagnostic plot



Example 2: Transformed data

Repeat on logarithms:

	1948	1952	1956	1960	1964	1968	1972
100m	4.63	4.64	4.65	4.62	4.61	4.60	4.62
200m	5.35	5.33	5.33	5.32	5.31	5.29	5.30
400m	6.14	6.13	6.15	6.11	6.11	6.08	6.10
800m	7.00	7.00	6.98	6.97	6.96	6.95	6.97
1500m	7.74	7.72	7.70	7.68	7.69	7.67	7.68

Example 2: Transformed data

Median polish on logs (times 1000)

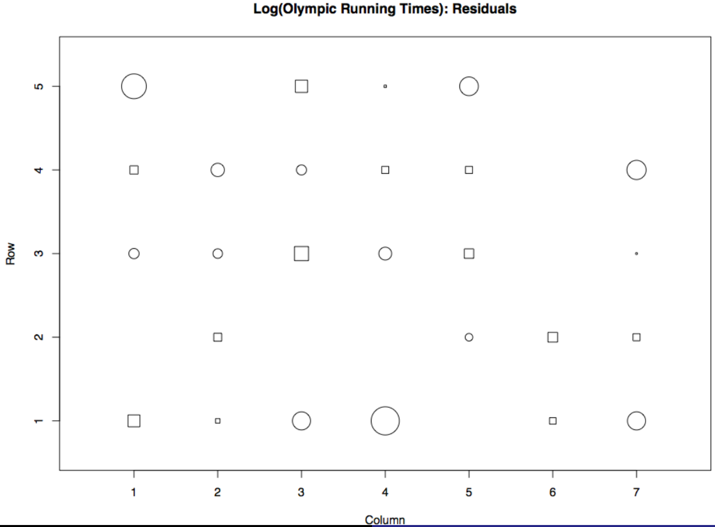
	1948	1952	1956	1960	1964	1968	1972	Row
100m	-17	0	17	1	-9	0	0	-1487
200m	7	-5	-3	5	5	0	-10	-794
400m	-3	-2	22	-4	10	0	0	0
800m	0	7	0	0	-2	10	5	857
1500m	16	2	-9	-21	0	5	-9	1586
Col	27	20	13	0	-10	-29	-9	6111

Analog $R^2 : 1 - 0.2061/33.2018 = 0.994$

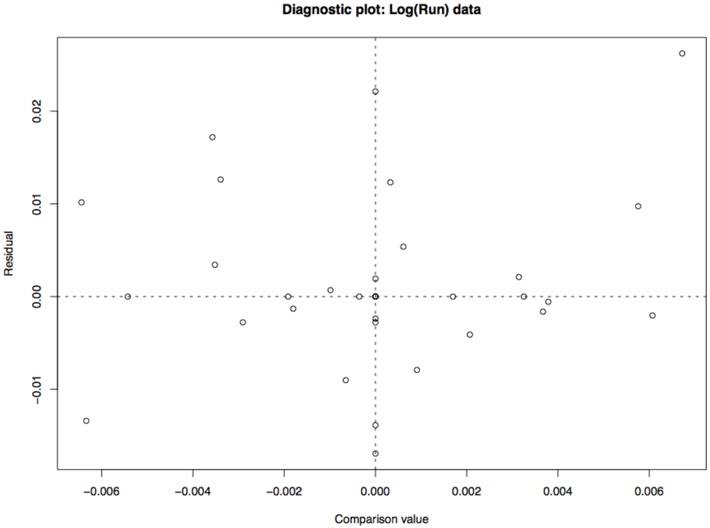
Pattern in residuals?

Example 2: Transformed data, plot residuals

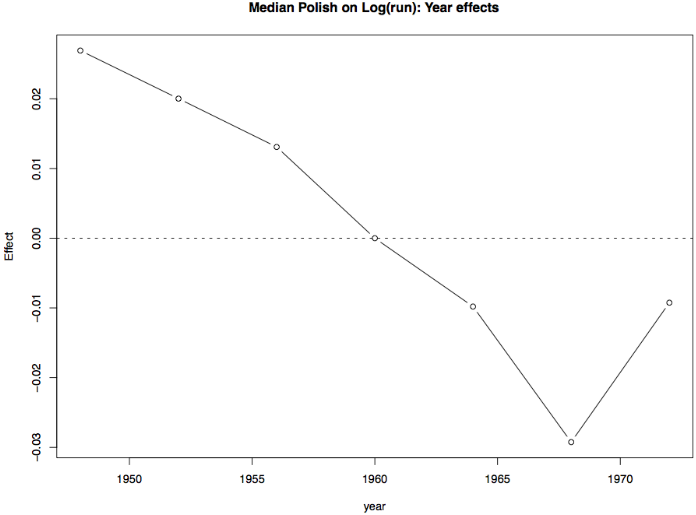
No pattern in residuals; check diagnostic plot



Example 2: Transformed data, diagnostic plot



Example 2: Transformed data, column effects



Transformation for two-way table

Other values of diagnostic plot slope besides 1?

Suppose y_{ij} do not fit additive model $m + a_i + b_j$, *but transformed data* y_{ij} *do:*

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

$$y_{ij} = (m + a_i + b_j)^{1/p}$$

$$y_{ij} = m^{1/p}(1 + a_i/m + b_j/m)^{1/p}$$

Taylor's series about $t_0 = 0$:

$$f(t) = (1+t)^u \approx f(0) + f'(0) \cdot t + (f''(0)/2) \cdot t^2$$

where $t = (a_i/m + b_j/m)$ and $u = 1/p$

Diagnostic Plot for Additivity

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 $p = 1 - b = 1 -$ 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

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

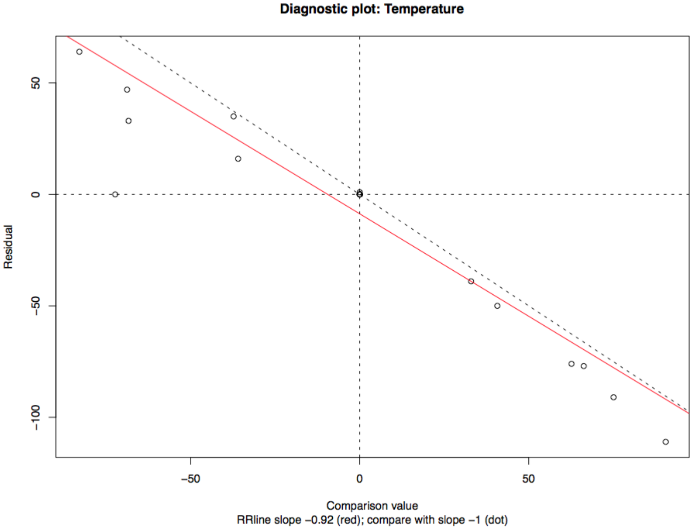
Why Does This Work?

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.

Temperature Example: Diagnostic plot



Temperature Example: Transformed data

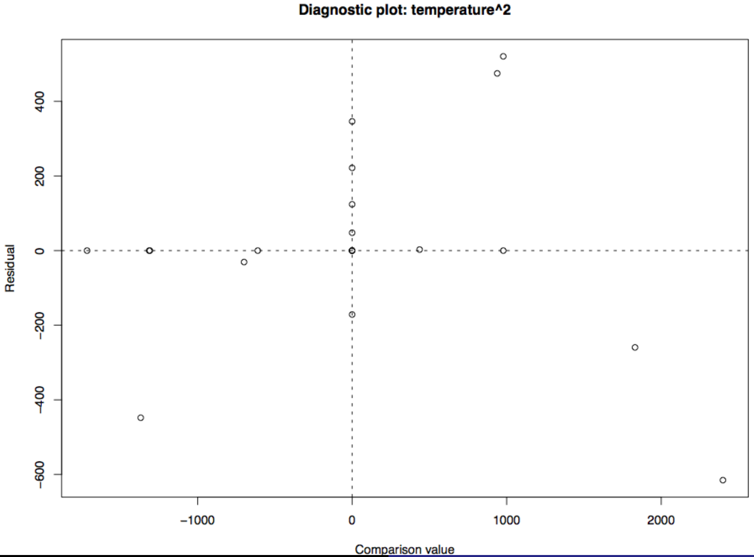
temp^2:	Jan	Feb	Mar	Apr	May	Jun	Jul
Caribou	76	96	471	1204	2352	3411	4096
Washington	1310	1376	2052	2959	4186	5388	5975
Laredo	3318	3832	4679	5761	6593	7362	7691

Median polish on temp^2:

	Jan	Feb	Mar	Apr	May	Jun	Jul	Row
1	521	475	3	0	-31	0	0	-1755
2	0	0	-171	0	48	222	124	0
3	-448	0	0	346	0	-259	-615	2455
Col	-1649	-1583	-736	0	1179	2206	2892	2959

$$R^2 = 1 - \sum |res| / \sum |y_{ij} - 3410.56| = 0.919 \quad \text{(cf. 0.834)}$$

Temperature Example: Transformed data, diagnostic plot



Transformation for two-way table

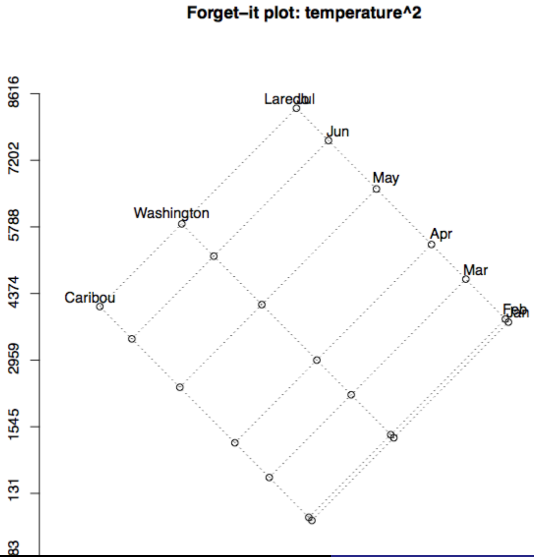
- Sometimes p is not -2 (reciprocal square), -1 (reciprocal), -1/2 (reciprocal square root), -1/3 (reciprocal cube root), 0 (log), 1/2 (square root), 2 (square), . . .
- Suppose $\mathit{slope} = -5.2 \Rightarrow p = 6.2$

  Don't even try transformation
- Instead, fit data as

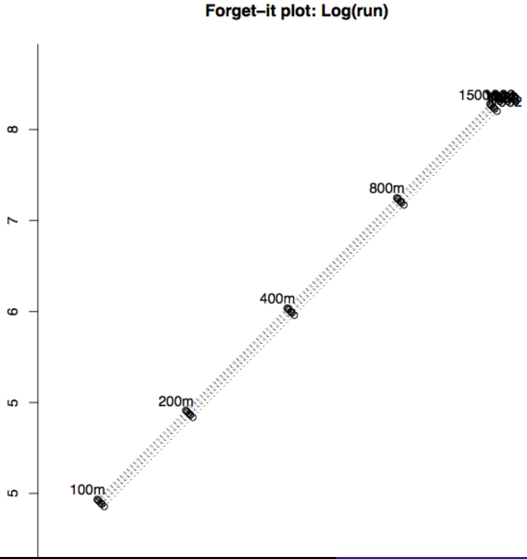
$$\hat{y}_{ij} = m + a_i + b_j + k \cdot a_i b_j$$

where $k = -5.2/m$

Example 1: Transformed data, Forget-it plot



Example 2: Forget-it plot



Example 2: Standardize data

Better: Standardize run data: #seconds per 100m
(divide 200m times by 2; . . .; 1500m times by 15)

```
run.std
          1948    1952    1956    1960    1964    1968    1972
 100m   103.00  104.00  105.00  102.00  100.00   99.00  101.00
 200m   105.50  103.50  103.00  102.50  101.50   99.00  100.00
 400m   115.50  114.75  116.75  112.25  112.75  109.50  111.75
 800m   136.50  136.50  134.63  132.88  131.38  130.38  132.38
1500m   153.20  150.13  147.47  143.73  145.40  143.27  144.20
```

Example 2: Standardize data

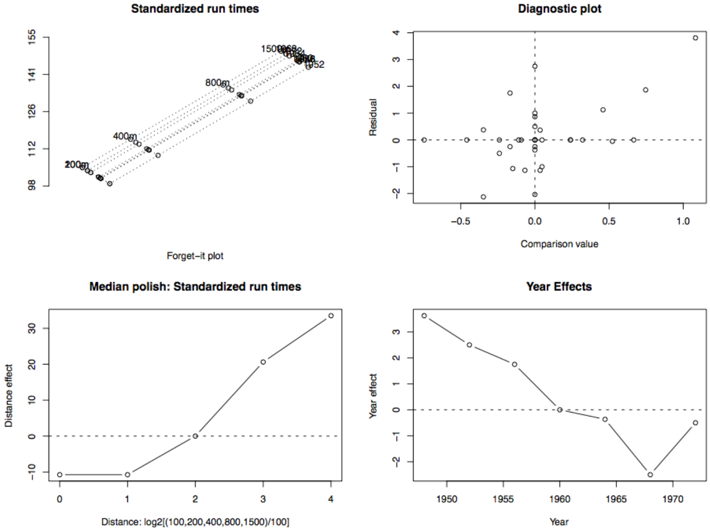
Median polish on standardized running times:

	1948	1952	1956	1960	1964	1968	1972	Row
100m	-2.13	0.00	1.75	0.50	-1.13	0.00	0.00	-10.7
200m	0.38	-0.50	-0.25	1.00	0.37	0.00	-1.00	-10.7
400m	-0.38	0.00	2.75	0.00	0.87	-0.25	0.00	0.00
800m	0.00	1.13	0.00	0.00	-1.13	0.00	0.00	20.6
1500m	3.81	1.87	-0.05	-2.03	0.00	0.00	-1.07	33.5
Col	3.63	2.50	1.75	0.00	-0.37	-2.50	-0.50	112.2

Analog

$$R^2 = \sum |res| / \sum |data - 112.75| = 1 - 24.325/546.525 = 0.9555$$

Summary Table



Standardized run times

Diagnostic plot

Median polish: Standardized run times

Year Effects