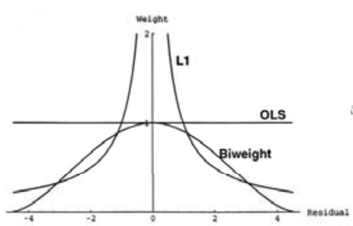
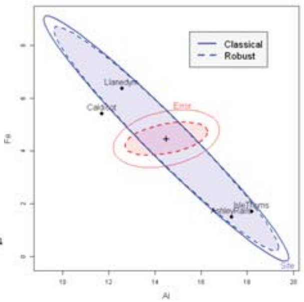
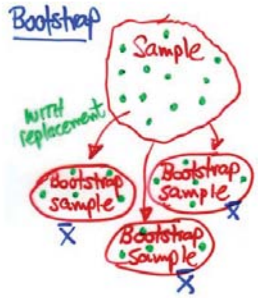






Robust statistical inference & bootstrapping

(What to do when you're not feeling Normal?)

Michael Friendly

Psychology 6140

Classical statistical inference

- Relies on distributional assumptions
  - GLM: $\varepsilon_i \sim N(0, \sigma^2)$
  - *generalized* LMs: allow other *assumed* distributions (e.g., Poisson for counts, binomial for binary data)
  - $\rightarrow b_i \sim N(\beta_i, \sigma^2 (X'X)^{-1}_{ii})$ but *only* under assumptions
- In some cases, all we have: asymptotic results
  - CFA: minimize $F(S, \Sigma)$
  - $(n\text{-}1)\ F_{min} \sim \chi^2$ as $n \rightarrow \infty$.
  - Cold comfort with small *n*.
- Robust methods & bootstrapping substitute computation for assumptions
  - Good news: These are general ideas, that apply to *all* statistical methods.
  - Bad news: sometimes requires specialized software (but SAS, R, SPSS are catching up)

Two kinds of robustness

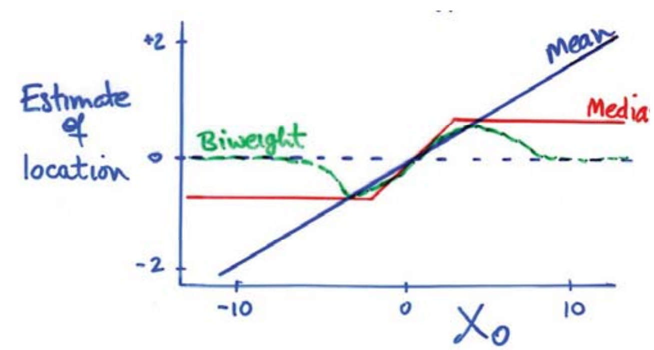
- Robustness of validity (Type 1 error)
  - Is the *p*-value for a test approx. correct over a range of data distributions?
  - OLS: OK– *p*-values not seriously affected by (moderate) non-normality
  - More complex models (e.g., CFA): How are tests affected?
- Robustness of efficiency (Type 2 error)
  - Is power high over a wide range of distributions?
  - OLS not robust in this sense– efficiency seriously degraded for heavy-tailed distributions (decrease in power)
  - Related idea: resistance--- lack of influence of small # of outliers

Trivial example: measures of location

- Sample: x = -2, -1, 0, 1, 2
  - Mean = median = 0
- What happens as we add one new observation, x_0, over range of all values?

Different estimators can be considered in terms of their **influence function**

A given estimate can be made **robust** by restricting influence of a given observation



Weighted least squares

- One useful method for correcting a variety of problems in linear models is to estimate parameters by *weighted least squares*, i.e., minimize

$$Q(\boldsymbol{\beta}) = \sum w_i e_i^2$$

for some **specified** weights, $w_1, w_2, \ldots, w_n$

- This idea provides the basis for a large class of robust methods

Weighted least squares

- WLS solution:
  - Let $\mathbf{W} = \text{diag}(w_1, w_2, \ldots, w_n)$
  - then

$$\mathbf{b} = (\mathbf{X'WX})^{-1}\mathbf{X'Wy}$$

  - minimizes

$$Q(\boldsymbol{\beta}) = \sum w_i e_i^2 = (\mathbf{y} - \mathbf{Xb})'\mathbf{W}(\mathbf{y} - \mathbf{Xb})$$

- SAS:

```
proc reg;
  weight w;
  model y = x1 x2 x3;
```

R:

```
lm(y~x1+x2+x3,
   weights=w)
```


M-estimators for robust linear models

- Idea: generalize OLS and WLS by minimizing a symmetric function of the residuals

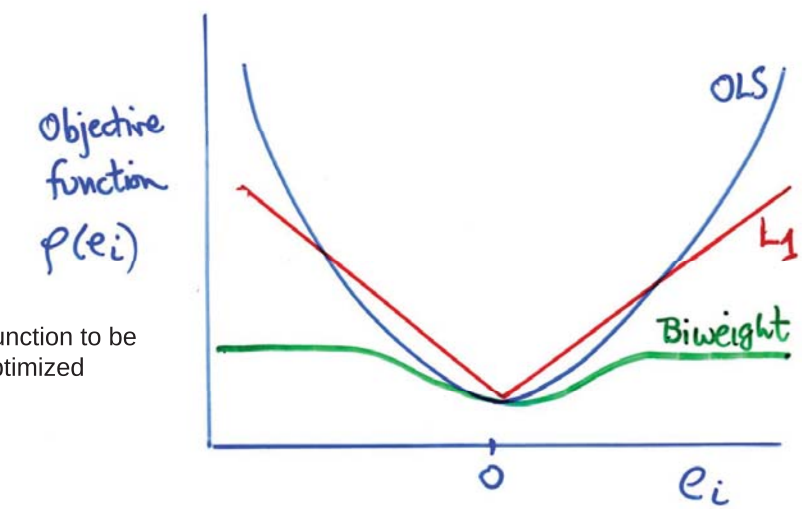
$$Q(e_i, \rho) = \sum \rho(e_i)$$

where the function, $\rho(e_i)$ can reduce influence of outliers

  - OLS: $\rho(e_i) = e_i^2$
  - L_1 estimation: $\rho(e_i) = |e_i|$ (least absolute value)
  - Bi-weight:

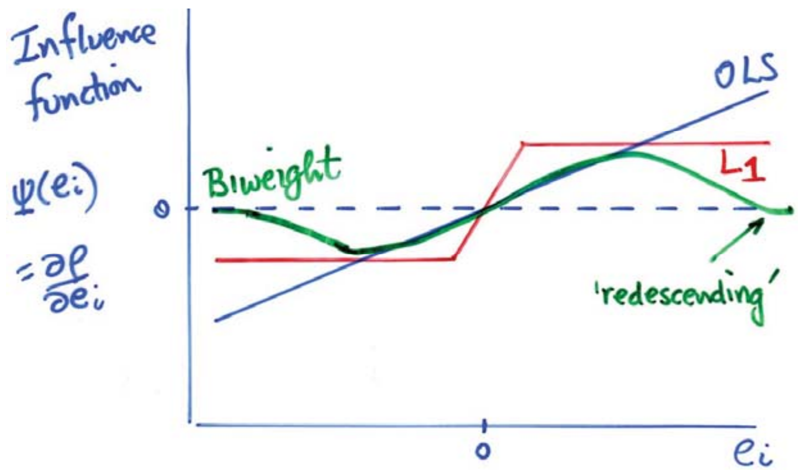
$$\rho(e_i) = \begin{cases} [1-(e_i/c)^2]^2 & |e_i| \le c \\ 1 & |e_i| > c \end{cases}$$

M-estimators: objective functions



Function to be optimized

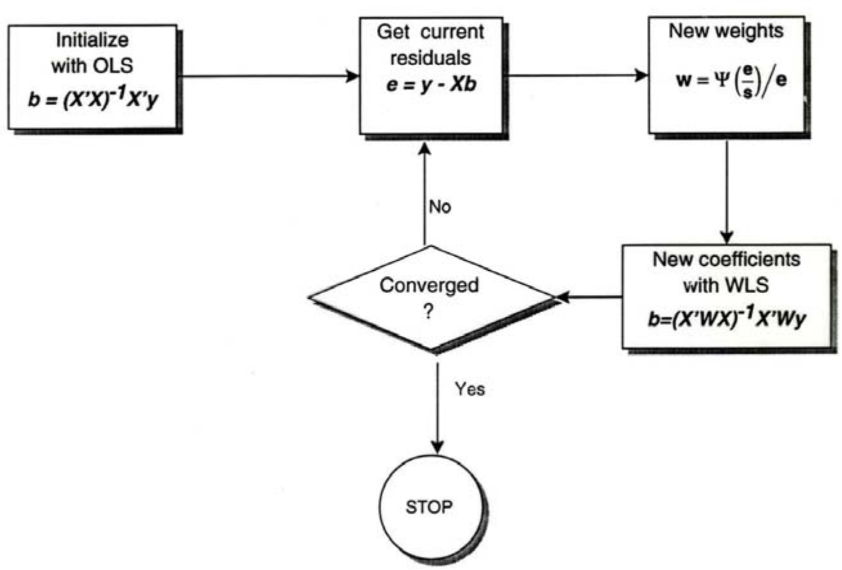
M-estimators: influence functions



Finding M-estimates by IRLS

- Minimize $Q = \Sigma\rho(e_i) = \Sigma\rho(y_i - \mathbf{x}_i'\mathbf{b})$ by WLS, using the weight function $w(e_i) = \frac{\psi(e_i)}{e_i} = \frac{\rho'(e_i)}{e_i}$
- then
  $$\mathbf{b} = (\mathbf{X'WX})^{-1}\mathbf{X'Wy}$$
- where $\mathbf{W} = \text{diag}\,(w(e_1), \ldots w(e_n))$
- But: weights, $w(e_i)$ and coefficients **b** depend on each other. Therefore:
  - Iterate: compute new weights, new coefficients
  - Until: coefficients don't change (IRLS)

Iteratively Reweighted Least Squares



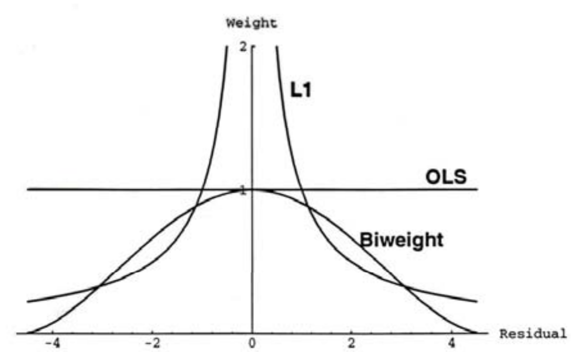
Weight functions for M-estimators

OLS: $w(e_i) = \frac{\Psi(e_i)}{e_i} = \frac{e_i}{e_i} = 1$ all equally weighted

L1: $w(e_i) = \frac{\text{sign}(e_i)}{e_i} = \frac{1}{|e_i|}$

Biweight: $w(e_i) = \begin{cases} \left[1 - \left(\frac{e_i}{c}\right)^2\right]^2 & \text{if } |e_i| \le c \\ 0 & |e_i| > c \end{cases}$

A graph of different weight functions shows that we might want a function like the biweight that gives smaller weights as residuals get large

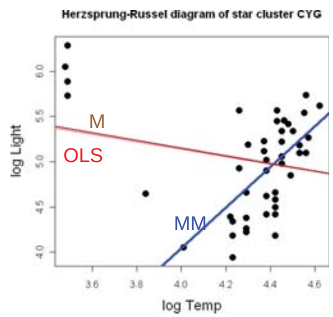


Modern robust methods

- Robust methods in statistics is a growth topic
  - Good for univariate models; multivariate models still need work
- New classes feature high *breakdown-bound* –proportion of unusual cases before estimates are affected
  - M estimators not resistant to leverage points
  - MM, LTS, S methods– high breakdown

LTS: Least trimmed squares– minimize SS of smallest *h*% of residuals (50 ≤*h*≤75)

M estimator can't resist the cluster of 4 red giants



Inference & hypothesis tests

- How to calculate robust standard errors?
- Asymptotic var-cov matrix of the M-estimator, b, is given by

$$Var(\mathbf{b})_{p\times p} = \frac{\sum[\psi(e_i)]^2 / n}{[\sum \psi'(e_i)]^2} (\mathbf{X}'\mathbf{X})^{-1}$$

- For comparison, with OLS: $\psi(e_i)=e_i$, $\psi'(e_i)=1$, so this gives

$$Var(\mathbf{b})_{p\times p} = \frac{SSE}{n} (\mathbf{X}'\mathbf{X})^{-1}$$

- CIs & hypothesis tests: $z = b_j / ASE(b_j)$
- Caveat: Asymptotic theory depends on large *n*

Robust tools

- `robust` macro (http://datavis.ca/sasmac/)
  - Fits models with proc GLM, REG or LOGISTIC
  - Weight functions: BISQUARE, HUBER, LAV, OLS
- PROC ROBUSTREG
  - Fits all general linear models (ANOVA, regression)
  - Calculates asymptotic standard errors: CIs & hypothesis tests
  - Provides M-estimation, MM-estimation, LTS (least trimmed squares) and other methods
  - These have high-breakdown property– can tolerate a large proportion of outliers
- R: lots of robust stuff– See CRAN *Robust Task View*
  - `rlm()` in MASS package: M-estimation
  - `lmrob()` in robustbase package: highly robust and highly efficient MM estimator (95% efficiency for normal errors)
  - `robmlm()` in heplots package: multivariate LMs

Example: Duncan occupational prestige

```
%include data(duncan);
title 'Robust Regression - Duncan data';
%robust(data=duncan,
   response=prestige,
   model=income educ,
   id=job,
   proc=reg,
   function=bisquare,
   out=resids);
proc plot data=resids;
   plot _weight_ * case = job; run;
```

NB: The %robust macro is now deprecated, in favor of PROC ROBUSTREG, but I retain these examples to illustrate the details.

Example: Duncan occupational prestige

Robust Regression - Duncan data
Iteration history and parameter estimates

ITER	_RMSE_	INTERCEP	INCOME	EDUC	MAXDIF
1	13.369	-6.0647	0.5987	0.5458	0.9443
2	8.871	-7.6649	0.7213	0.4754	0.1580
3	8.951	-7.6916	0.7704	0.4383	0.1427
4	8.644	-7.6731	0.7966	0.4209	0.0637
5	8.496	-7.6112	0.8104	0.4115	0.0440

Note how coefficients for income and education change

Example: Duncan occupational prestige

Residuals, fitted values and weights

JOB	PRESTIGE	_FIT_	_WEIGHT_	_RESID_	_HAT_
Accountant	82	78.024	0.9784	3.975	0.0596
Chemist	90	79.645	0.8582	10.354	0.0523
Minister	87	43.975	0.0000	43.025	0.0000
Professor	93	82.526	0.8551	10.474	0.0668
Dentist	90	98.373	0.9061	-8.373	0.0930
Reporter	52	82.488	0.1311	-30.488	0.0099
Civil Eng.	88	86.128	0.9952	1.871	0.0696
Undertaker	57	56.878	0.9999	0.122	0.0619
Lawyer	89	94.308	0.9617	-5.308	0.0889
Physician	97	93.897	0.9868	3.103	0.0889
PS Teacher	73	68.736	0.9752	4.264	0.1085
RR Conductor	38	67.971	0.1471	-29.971	0.0557
...	..	...			

Example: fuel consumption data

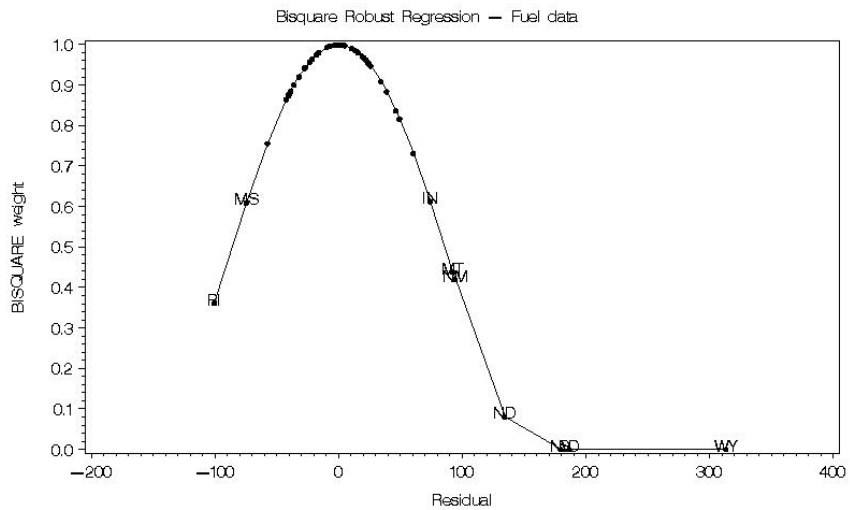
```
%include data(fuel);
title 'Bisquare Robust Regression - Fuel data';
%robust(data=fuel,
      response=fuel,
      model=tax drivers road inc,
      id=state,
      proc=reg,
      out=resids);

proc gplot data=resids;
      plot  _weight_ * _resid_;
```


Example: fuel consumption data

Bisquare Robust Regression - Fuel data
Iteration history and parameter estimates

iter	_RMSE_	Intercept	tax	drivers	road	inc	_maxdif_
1	66.306	377.291	-34.7901	1336.45	-.002425889	-0.066589	0.9957
2	48.214	437.749	-31.2176	1154.88	-.001636757	-0.065316	0.3940
3	43.741	470.093	-28.2648	1059.45	-.001200214	-0.066819	0.1861
4	41.463	485.838	-25.4150	1002.13	-.000636108	-0.069252	0.1098
5	40.631	489.507	-23.2760	975.41	-.000139617	-0.071349	0.1405
6	37.879	490.060	-20.3267	942.66	0.000479679	-0.073647	0.1287
7	34.875	472.821	-16.8588	928.44	0.001175523	-0.075289	0.0806
8	34.172	455.703	-15.4570	934.60	0.001460715	-0.075185	0.0349

Example: fuel consumption data



Example: proc robustreg

```
ods rtf file='robdunc0.rtf' style=journal;
ods graphics on;
proc robustreg data=duncan
         plots=(ddplot(label=leverage) rdplot(label=leverage)) ;
   model prestige = income educ / diagnostics itprint ;
   id job;
   output out=resids r=residual weight=weight outlier=outlier;
   run;
ods graphics off;
ods rtf close;
```

Using ODS Graphics, a variety of useful plots are produced, including:

- ddplot: Leverage plot: robust distances vs. Mahalanobis distances for Xs
- rdplot: Influence plot: robust residuals vs. robust distances

Example: proc robustreg

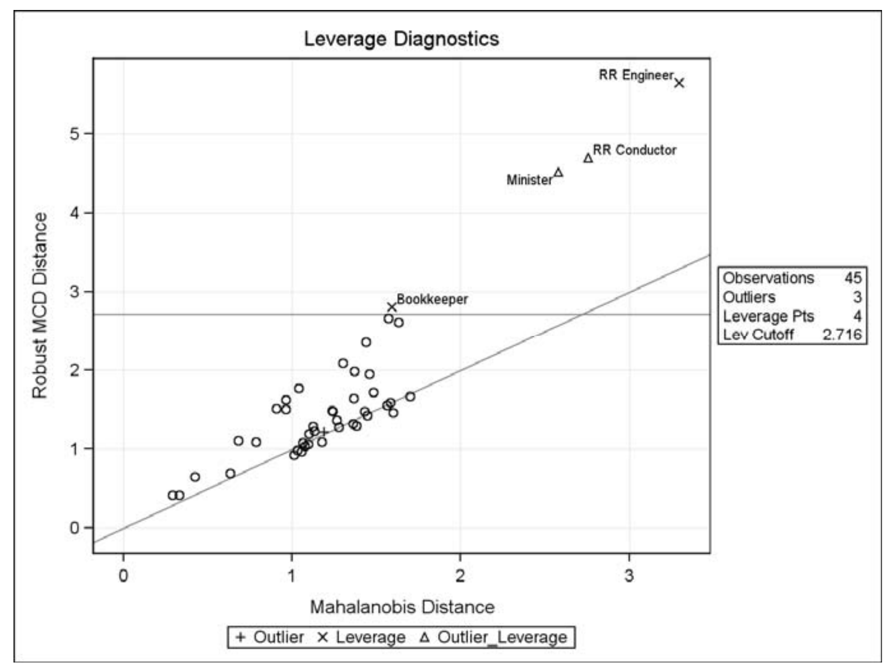
Parameter Estimates							
Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-7.4120	3.8733	-15.0036	0.1796	3.66	0.0557
income	1	0.7903	0.1085	0.5777	1.0030	53.06	<.0001
educ	1	0.4185	0.0891	0.2439	0.5931	22.07	<.0001
Scale	1	9.5553					

CIs and hypothesis tests are based on Wald χ^2

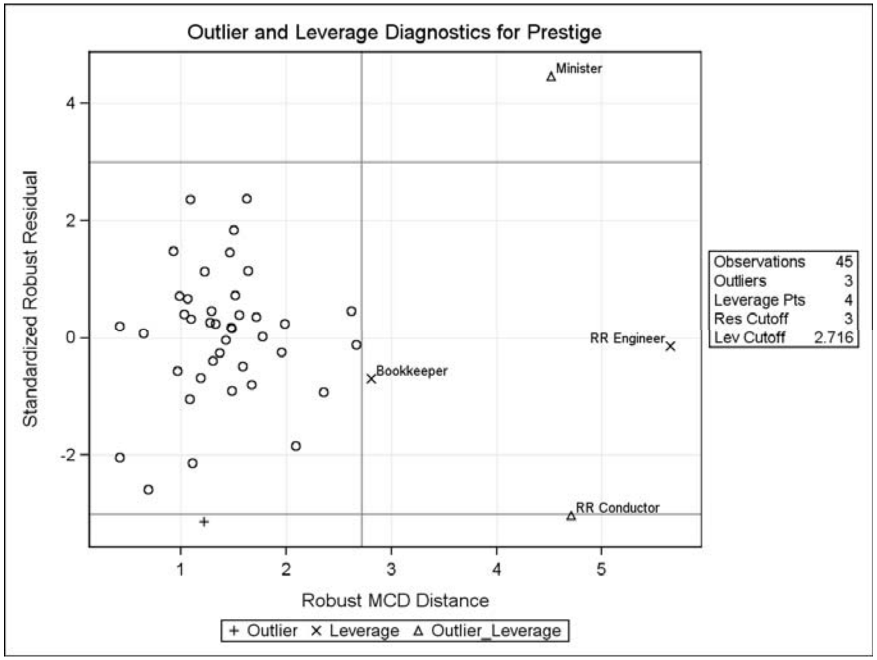
Diagnostics			
Obs	job	Standardized Robust Residual	Outlier
6	Minister	4.4647	*
9	Reporter	-3.1344	*
16	RR Conductor	-3.0227	*

The same 3 suspects are identified as outliers

ddplot: Leverage plot: robust distances vs. Mahalanobis distances for Xs



rdplot: Influence plot: robust residuals vs. robust distances



R: rlm() and lmrob()

```
> library(car)
> data(Duncan)
> library(MASS)
> dunc.robust <- rlm(prestige ~ income+education, data=Duncan)
> summary(dunc.robust)
```

```
Call: rlm(formula = prestige ~ income + education, data = Duncan)
Coefficients:
            Value  Std. Error t value
(Intercept) -7.111  3.881     -1.832
income       0.701  0.109      6.452
education    0.485  0.089      5.438

Residual standard error: 9.89 on 42 degrees of freedom
```

Which cases have small weights?

```
> cbind(Duncan,dunc.robust$w)[dunc.robust$w < .5,]
```

```
          type income education prestige dunc.robust$w
minister  prof     21        84       87      0.344664
reporter    wc     67        87       52      0.441727
```


Example: Robust ANOVA

An experiment was carried out to study the effects of two successive treatments (*Treat1*, *Treat2*) on the recovery time of mice with certain diseases.

Sixteen mice were randomly assigned into four groups for the four different combinations of the treatments.

The recovery times (*time*) were recorded (in hours) as shown in the following data set *recover*.

```
data recover;
      input  Treat1 $ Treat2 $ time @@;
   datalines;
   0 0 20.2  0 0 23.9  0 0 21.9  0 0 42.4
   1 0 27.2  1 0 34.0  1 0 27.4  1 0 28.5
   0 1 25.9  0 1 34.5  0 1 25.1  0 1 34.2
   1 1 35.0  1 1 33.9  1 1 38.3  1 1 39.9
   ;
```


Standard ANOVA:

```
proc glm data=recover;
    class Treat1 Treat2;
    model time = Treat1 | Treat2;
run;
```

Resuts are disappointing!

Source	DF	Squares	Mean Square	F Value	Pr > F
Model	3	209.9118750	69.9706250	1.86	0.1905
Error	12	451.9225000	37.6602083		
Corrected Total	15	661.8343750			

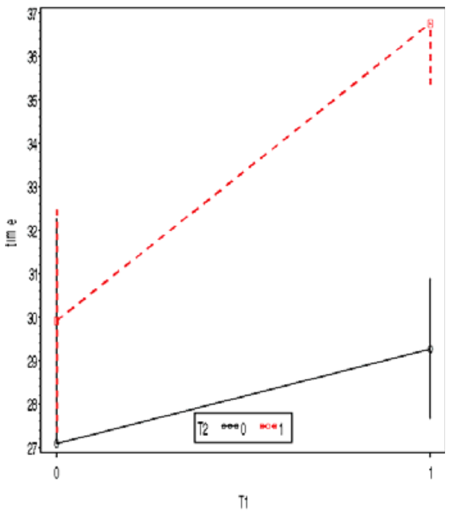
PERISH!

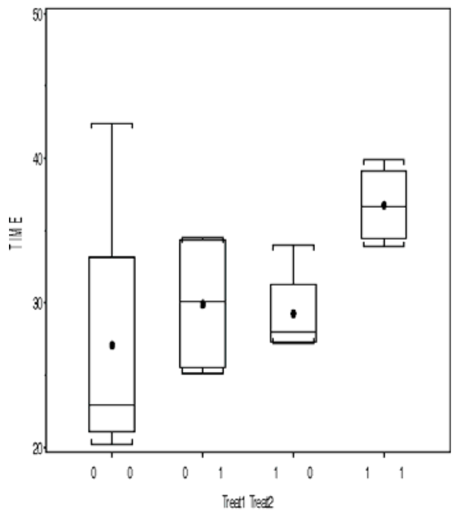
Source	DF	Type III SS	Mean Square	F Value	Pr > F
Treat1	1	81.4506250	81.4506250	2.16	0.1671
Treat2	1	106.6056250	106.6056250	2.83	0.1183
Treat1*Treat2	1	21.8556250	21.8556250	0.58	0.4609

Publish or perish?

Wait … it's time to plot the data!

- Plot of means seems to show an interaction, maybe main effect of T2
- Boxplot shows large variance in cell (0,0), but no outliers





PROC ROBUSTREG to the rescue:

```
ods rtf file='robust-anova.rtf';
ods graphics on;
proc robustreg data=recover  plot=histogram;
   class Treat1 Treat2;
   model time = Treat1 | Treat2 / diagnostics;
   T1_T2: test Treat1*Treat2;
   output out=robout r=resid sr=stdres;
run;
ods graphics off;
ods rtf close;
```

Robust versions of the F and Wald tests

Outlier and leverage diagnostics

Histogram of std. robust residuals

Parameter estimates show that both treatment main effects are significant at the 5% level:

Parameter Estimates

Parameter			DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept			1	36.7655	2.0489	32.7497	40.7814	321.98	<.0001
Treat1	0		1	-6.8307	2.8976	-12.5100	-1.1514	5.56	0.0184
Treat1	1		0	0.0000	.	.	.	.	.
Treat2	0		1	-7.6755	2.8976	-13.3548	-1.9962	7.02	0.0081
Treat2	1		0	0.0000	.	.	.	.	.
Treat1*Treat2	0	0	1	-0.2619	4.0979	-8.2936	7.7698	0.00	0.9490
Treat1*Treat2	0	1	0	0.0000	.	.	.	.	.
Treat1*Treat2	1	0	0	0.0000	.	.	.	.	.
Treat1*Treat2	1	1	0	0.0000	.	.	.	.	.
Scale			1	3.5346					

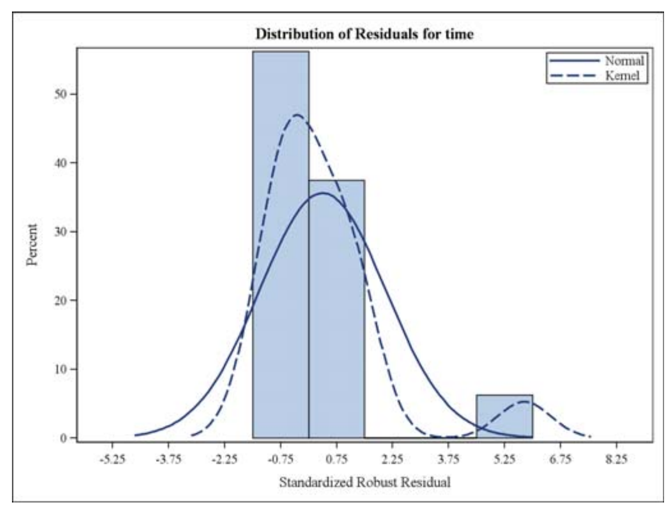
Diagnostics:

Obs	Standardized Robust Residual	Outlier
4	5.7722	*

Further investigation showed that the original value of 24.4 for the fourth observation was recorded incorrectly.

- Who published?
- Who perished?

The histogram plot of standardized robust residuals clearly show this as an outlier



Robust MLMs

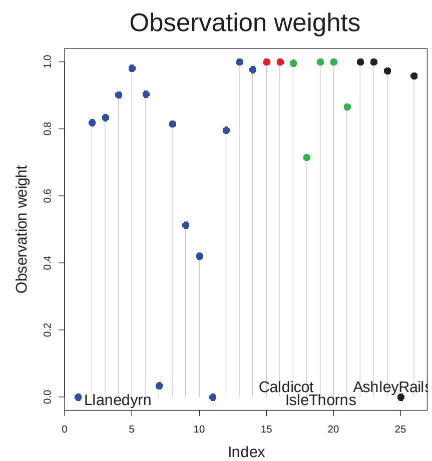
- Robust methods for univariate LMs are now well-developed and implemented
  - → proper SEs, CIs and hypothesis tests
- Analogous methods for multivariate LMs are a current hot research topic
- The heplots package now provides `robmlm()` for the fully general MLM (MANOVA, MMReg)
  - Uses simple M-estimator via IRLS
  - Weights: calculated from Mahalanobis D^2, a robust covariance estimator and weight function, ψ(D^2)

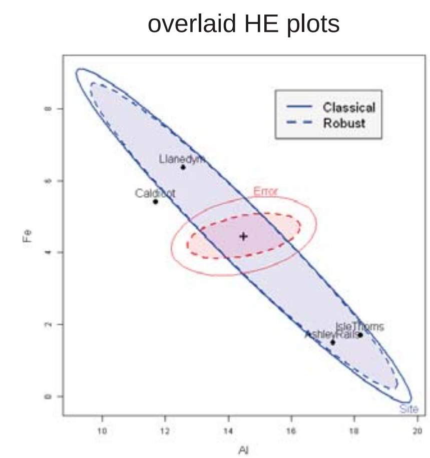
$$D^2 = (\mathbf{Y} - \hat{\mathbf{Y}})^T \mathbf{S}^{-1}_{\text{robust}} (\mathbf{Y} - \hat{\mathbf{Y}}) \sim \chi^2_p$$

  - Downside: SEs, *p*-values only approximate

Robust MLMs: Example

```
> pottery.mod <- lm(cbind(Al,Fe,Mg,Ca,Na)~Site, data=Pottery)
> pottery.rmod <- robmlm(cbind(Al,Fe,Mg,Ca,Na)~Site, data=Pottery)
```





Bootstrapping

- Classical statistical inference relies on
  - Distributional assumptions
  - Asymptotic results
- Bootstrapping is a non-parametric approach to inference that substitutes computation for assumptions

Functional bootstraps: help to pull you up from where you are, to where you want to be

bootstrap (v): help oneself, often through improvised means

Decorative bootstraps: we don't need these

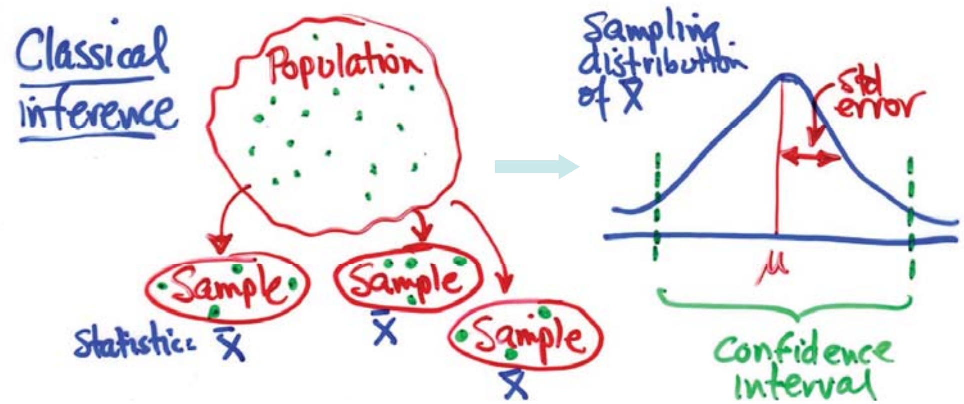
Bootstrapping

- Can provide more accurate inferences when data is badly behaved or *n* is small
- Can be applied when *no* sampling theory is available
  - Tests of equality of ratios (y/x)
  - fMRI studies: differences among patterns of brain activation
  - Joe Jackson: how did he hit in clutch situations?
- Can be applied to complex data-collection plans (stratified/clustered samples)

More general ideas: Resampling

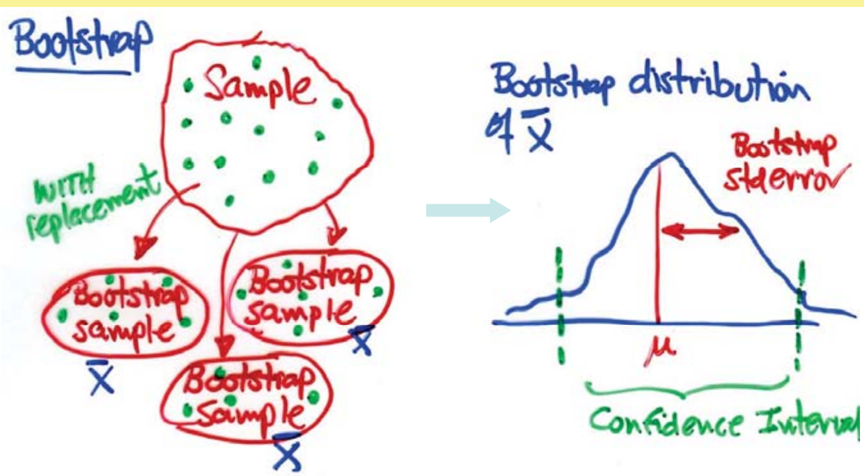
- The bootstrap is an example of the general idea of ***resampling*** from an original data set for statistical inference
- Other examples:
  - Jackknife: leave-one-out analysis
  - Cross-validation: choosing optimal model fitting parameters
  - Permutation tests: totally non-parametric
- Uses:
  - Std errors, CIs with small samples
  - Subset selection in linear models (PROC GLMSELECT)
  - Dealing with missing data

Classical statistical inference



Here, we rely on statistical theory (CLT) & assumptions (independence, normality, constant variance) to take us to the sampling distribution of the statistic of interest.

Bootstrap



Key idea:

Population is to the sample	AS	Sample is to the bootstrap sample

Bootstrap: general method

- Repeat *b*=1, …, B times (B > 200-1000+):
  - Generate random resample (w/ replacement)
  - The bootstrap sample **must** replicate conditions of original data
  - Calculate estimates of parameters, $\boldsymbol{\theta}_b{}^*$
- Estimate standard errors as the standard deviation of **θ*** over the B bootstrap samples

$$SE_{boot}(\boldsymbol{\theta}) = \left(\sum_{b=1}^{B} (\boldsymbol{\theta}_b^* - \bar{\boldsymbol{\theta}}^*) / (B-1) \right)^{1/2}$$

- Calculate bootstrap CIs by finding the lower and upper (α/2) percentiles of the bootstrap distributions
- Other methods for calculating bootstrap CIs provide bias correction

Bootstrap: trivial example

TABLE 16.1 Contrived "Sample" of Four Married Couples, Showing Husbands' and Wives' Incomes in Thousands of Dollars.

Observation	*Husband's Income*	*Wife's Income*	*Difference* Y_i
1	24	18	6
2	14	17	−3
3	40	35	5
4	44	41	3

$\bar{Y} = 2.75$

Test H_0: $\mu_H - \mu_W = 0$ by bootstrap

For this example, all of the possible 256 bootstrap samples of size n=4 can be enumerated

TABLE 16.2 A Few of the 256 Bootstrap Samples for the Dataset [6, −3, 5, 3], and the Corresponding Bootstrap Means, $\bar{Y}^*_b$

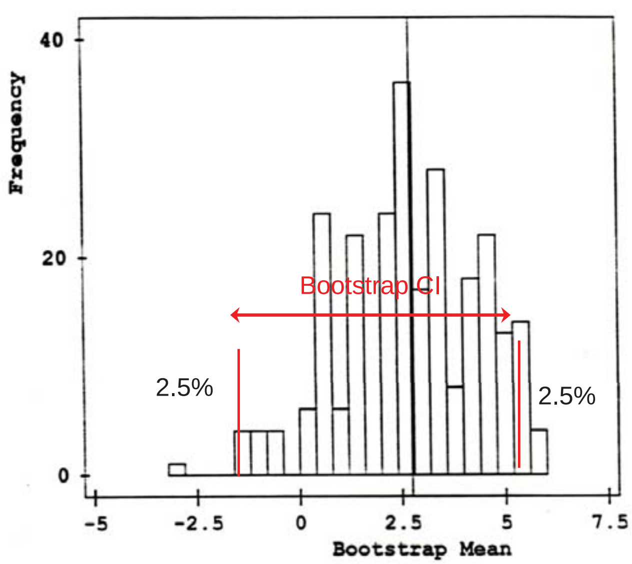
Bootstrap Sample *b*	Y^*_{b1}	Y^*_{b2}	Y^*_{b3}	Y^*_{b4}	$\bar{Y}^*_b$
1	6	6	6	6	6.00
2	6	6	6	−3	3.75
3	6	6	6	5	5.75
⋮	⋮				⋮
100	−3	5	6	3	2.75
101	−3	5	−3	6	1.25
⋮	⋮				⋮
255	−3	3	3	5	3.50
256	3	3	3	3	3.00

Bootstrap sampling distribution

Sample mean: Y=2.75

CI: includes 0, so we cannot reject H_0

More generally: we have an SE and CI that does not rely on assumptions or large N!



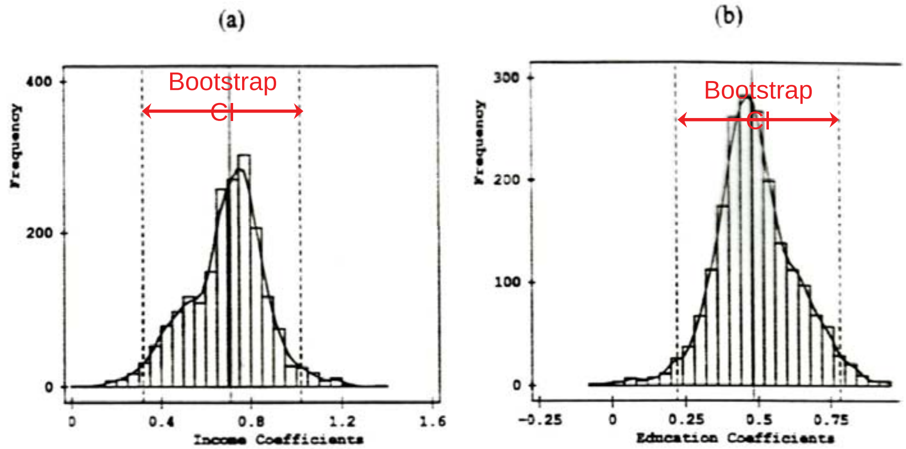
Bootstrapping linear models

- Random-X resampling
  - Regressors are treated as random
  - Select bootstrap samples from the data
- Fixed-X resampling
  - Regressors treated as fixed: implies that model is correct
  - Select bootstrap samples from the residuals
  - Add resampled residuals to fitted values to give the bootstrap sample

Bootstrapping the Duncan M-estimator

B=2000 bootstrap samples of size n=45 were generated randomly with replacement

M-estimates of the coefficients for Income and Education calculated for each



Bootstrapping: the boot macro

The boot macro can be used to do a bootstrap analysis of almost any statistical method. You need to write a macro to do the analysis for one sample.

```
%include data(duncan);
title 'Bootstrap OLS Regression - Duncan data';
*-- Macro to do one regression, called by %BOOT;
%macro reg(data=, out=);
proc reg noprint data=&data  outest=&out(drop=prestige _rmse_);
     model prestige = income educ;
     %bystmt;        *-- analyze BY _sample_;
     run;
%mend;

%boot(data=duncan, random=123, samples=1000, analyze=reg);
%bootci(stat=income educ, method=pctl);
```

data = name of input data set
out = name of output data set containing statistics

%bootci requires ~1000 for a 90% CI, more for greater confidence

Some of the 1000 bootstrap estimates

Obs	_sample_	Intercept	income	educ
1	1	-5.7582	0.54098	0.58980
2	2	-8.3689	0.97551	0.27158
3	3	-9.5888	0.72040	0.46339
4	4	-4.5667	0.49814	0.60600
5	5	-7.8326	0.76440	0.42816
6	6	-4.1156	0.56472	0.50862
7	7	-6.2003	0.54978	0.59809

Partial output from the bootci macro

Name	Observed Statistic	Bootstrap Mean	Approximate Bias	Approximate Standard Error	Approximate Lower Confidence Limit
educ	0.54583	0.52897	-0.016865	0.13927	0.28973
income	0.59873	0.61985	0.021113	0.17253	0.23947

Name	Bias-Corrected Statistic	Approximate Upper Confidence Limit	Confidence Level (%)	Method for Confidence Interval
educ	0.56270	0.83566	95	Bootstrap Normal
income	0.57762	0.91577	95	Bootstrap Normal

Now, do the same for a robust regression:

```
title 'Bootstrap robust regression';
%macro robreg(data=, out=);
   ods listing close;
   proc robustreg data=&data  outest=&out(drop=prestige);
        model prestige = income educ ;
        %bystmt;        *-- analyze BY _sample_;
       run;
   ods listing;
   %mend;

%boot(data=duncan, random=123, samples=1000, stat=income educ,
         analyze=robreg);
%bootci(stat=income educ, method=pctl);
```

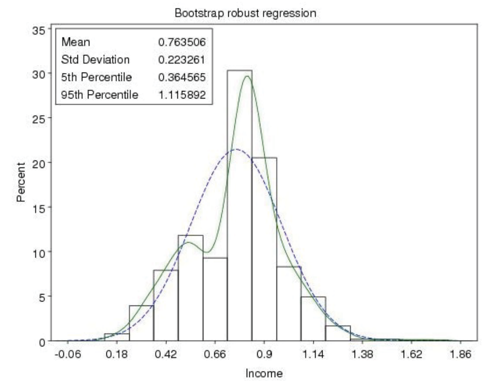
Partial output from the bootci macro

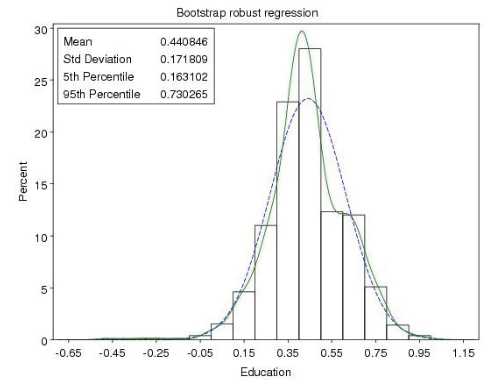
Name	Observed Statistic	Bootstrap Mean	Approximate Bias	Approximate Standard Error	Approximate Lower Confidence Limit
educ	0.41849	0.44085	0.022356	0.17181	0.05939
income	0.79035	0.76351	-0.026841	0.22326	0.37961

Name	Bias-Corrected Statistic	Approximate Upper Confidence Limit	Confidence Level (%)	Method for Confidence Interval
educ	0.39613	0.73287	95	Bootstrap Normal
income	0.81719	1.25477	95	Bootstrap Normal

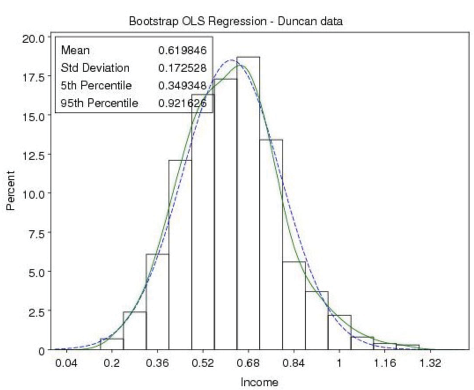
Name	Minimum Resampled Estimate	Maximum Resampled Estimate	Number of Resamples	LABEL OF FORMER VARIABLE
educ	-0.48286	0.98063	1000	Education
income	0.16942	1.74531	1000	Income

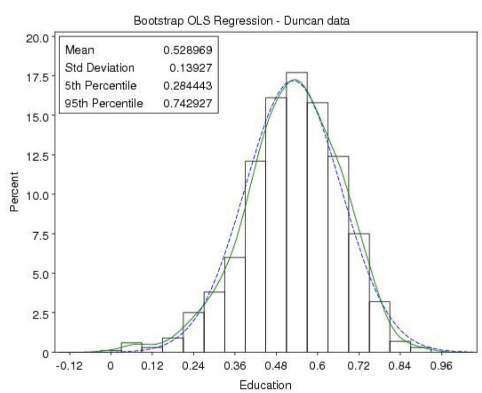
Graphs of the bootstrap distribution of M-estimates of for Income and Education





Compare with the bootstrap distribution of OLS estimates of for Income and Education





The CIs for robust regression are wider, but more realistic

Complex example: Bootstrapping a SEM

- Data from the Canadian National Election Survey, 1977
  - Items: 4-point Likert scales
  - MBSA2: We should be more tolerant of people who choose to live according to their own standards
  - MBSA7: Newer lifestyles are contributing to the breakdown of our society
  - MBSA8: The world is always changing and we should adapt our view of moral behaviour to these changes
  - MBSA9: This country would have many fewer problems if there were more emphasis on traditional family values

The data:

```
> str(CNES)
'data.frame':   1529 obs. of  4 variables:
 $ MBSA2: Ord.factor w/ 4 levels "StronglyDisagree"<..: 4 3 3 4 3 3 2 3 2 3 ...
 $ MBSA7: Ord.factor w/ 4 levels "StronglyDisagree"<..: 3 4 2 3 1 2 1 1 3 3 ...
 $ MBSA8: Ord.factor w/ 4 levels "StronglyDisagree"<..: 2 1 2 1 3 3 2 2 1 3 ...
 $ MBSA9: Ord.factor w/ 4 levels "StronglyDisagree"<..: 2 4 3 4 2 3 3 2 4 4 ...
```

Because the items are polytomous, we compute polychoric correlations:

```
> library(polycor)
> R.cnes <- hcor(CNES)
> R.cnes
           MBSA2      MBSA7      MBSA8      MBSA9
MBSA2  1.0000000 -0.3017953  0.2820608 -0.2230010
MBSA7 -0.3017953  1.0000000 -0.3422176  0.5449886
MBSA8  0.2820608 -0.3422176  1.0000000 -0.3206524
MBSA9 -0.2230010  0.5449886 -0.3206524  1.0000000
```

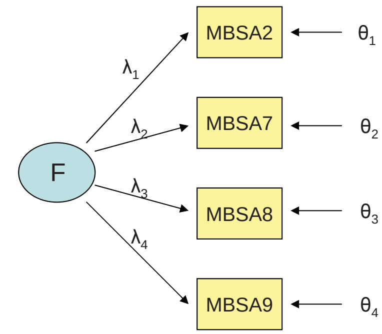
However, this will cause problems for a SEM:

- Std errors of polytomous correlations are complex
- Std errors of the SEM analysis will be incorrect (Pearson cor. assumed)

The model: One factor CFA

The sem package in R provides simple `cfa()` notation to specify the model:

```
> model.cnes <-
cfa(reference.indicators=FALSE)
1: F: MBSA2, MBSA7, MBSA8, MBSA9
2:
Read 1 item
NOTE: adding 4 variances to the model
```



Fitting the model:

```
> sem.cnes <- sem(model.cnes, R.cnes, N=1529)
> summary(sem.cnes)

 Model Chisquare =  33.211   Df =  2 Pr(>Chisq) = 6.1407e-08
 Chisquare (null model) =  984.33   Df =  6
 Goodness-of-fit index =  0.98934
 Adjusted goodness-of-fit index =  0.94668
 RMSEA index =  0.10106   90% CI: (0.07261, 0.13261)
 Bentler-Bonnett NFI =  0.96626
 Tucker-Lewis NNFI =  0.9043
 Bentler CFI =  0.9681
 SRMR =  0.035365
 BIC =  18.547

Parameter Estimates
     Estimate Std Error z value Pr(>|z|)
lam1 -0.38933 0.028901  -13.471 0        MBSA2 <--- F
lam2  0.77792 0.029357   26.498 0        MBSA7 <--- F
lam3 -0.46868 0.028845  -16.248 0        MBSA8 <--- F
lam4  0.68680 0.028409   24.176 0        MBSA9 <--- F
the1  0.84842 0.032900   25.788 0        MBSA2 <--> MBSA2
the2  0.39485 0.034436   11.466 0        MBSA7 <--> MBSA7
the3  0.78033 0.031887   24.472 0        MBSA8 <--> MBSA8
the4  0.52831 0.030737   17.188 0        MBSA9 <--> MBSA9

 Iterations =  12
```

Seems to fit well

But: can we trust these results?

Asymptotic std errors are not to be trusted here

Doing the bootstrap analysis:

```
> # Define a function to return correlations for a bootstrap sample
> hcor <- function(data) hetcor(data, std.err=FALSE)$correlations
>
> boot.cnes <- bootSem(CNES, sem.cnes, R=100, cov=hcor)
```

~ 48 sec. to do R=100 samples

```
> summary(boot.cnes, type="norm")
Call: boot.sem(data = CNES, model = sem.cnes, R = 100, cov = hcor)

Lower and upper limits are for the 95 percent norm confidence interval

      Estimate          Bias  Std.Error      Lower      Upper
lam1 -0.3893278  0.0017661057 0.03480571 -0.4593118 -0.3228759
lam2  0.7779153  0.0054905355 0.03456017  0.7046881  0.8401615
lam3 -0.4686838  0.0078697011 0.03627866 -0.5476584 -0.4054486
lam4  0.6867992 -0.0015019493 0.02937082  0.6307354  0.7458669
the1  0.8484245  0.0001720356 0.02719461  0.7949520  0.9015530
the2  0.3948479 -0.0097552534 0.05425617  0.2982630  0.5109433
the3  0.7803349  0.0060123631 0.03360386  0.7084602  0.8401849
the4  0.5283057  0.0012078609 0.04055924  0.4476032  0.6065925
> # cf., standard errors to those computed by summary(sem.cnes)
```

Summary

- Robust methods
  - General solutions to problems of “messy” data
  - Weighted analysis, using weights = f(residuals)
  - Iterative method: IRLS
  - Now get asymptotic std errors, robust tests, etc.
  - More exact methods now for univariate (G)LMs
- Bootstrapping (resampling) methods
  - General solutions to problems of “messy” analysis
  - Generate sampling distribution from the data
  - Substitutes computation for assumptions