Regression III: Advanced Methods

Lecture 11. Resampling and Regression

Bob Andersen
McMaster University

http://socserv.mcmaster.ca/andersen

Goals of the Lecture

- Introduce the general idea of ***Resampling:*** Techniques to resample from the original sample
  - ***Bootstrapping***
  - ***Jackknife***
  - ***Cross-validation***
- An extended example of bootstrapping after a robust regression using **R**
- I'll also briefly discuss how cross-validation can be used to validate predictions from a regression model

Resampling: An Overview

- Resampling techniques sample from the original data set
- Some of the applications of these methods are:
  - to compute standard errors and confidence intervals (either when we have small sample sizes or for a statistic that does not have an easily derivable asymptotic standard error)
  - subset selection in regression
  - handling missing data
  - selection of degrees of freedom in nonparametric regression (especially generalized additive models)
- For the most part, this lecture will discuss resampling techniques in the context of computing confidence intervals and hypothesis tests for regression analysis

Resampling and Regression A Caution

- There is no need whatsoever for ***bootsrapping*** in regression analysis if the OLS assumptions are met
  - In such cases, OLS estimates are the most unbiased and efficient of estimates
- There are situations, however, when we cannot satisfy the assumptions and thus other methods are more helpful
  - For example, robust regression (such as M-estimation) often provides better estimates than OLS in the presence of influential cases. Nonetheless, the SEs from robust regression are reliable only for large sample sizes
- ***Crossvalidation*** is particularly helpful for validating models and choosing model fitting parameters, such as smoothing parameters and degrees of freedom, in nonparamteric regression

Bootstrapping: General Overview (1)

- If we assume that a random variable X or statistic has a particular population distribution we can study how a statistical estimator computed from samples behaves
- We don't always know, however, how a variable or statistic is distributed in the population
  - For example, there may be a statistic for which standard errors have not been formulated (e.g., imagine we wanted to test whether two additive scales have significantly different levels of internal consistency—to my knowledge there is no formula for the standard errors for Cronbach's α)
  - Another example is the impact of missing data on a distribution—we don't know how the missing data differ from the observed data
- Bootstrapping is a technique for estimating confidence intervals of statistical estimators without making assumptions about the distribution giving rise to the data

Bootstrapping: General Overview (2)

- Assume that we have a sample of size n for which we require more reliable standard errors for our estimates
  - Perhaps n is small, or alternatively, we have a statistic for which there is no known sampling distribution
- The ***bootstrap*** provides one "solution"
  - Take several ***new samples*** from the ***original sample***, calculating the statistic each time
  - Calculate the average and its standard error from the empirical distribution of the bootstrap samples
  - In other words, we find a standard error based on sampling the original sample
- We apply principles of inference similar to those employed when sampling from the population
  - ***The population is to the sample as the sample is to the bootstrap samples***

Bootstrapping: General Overview (3)

- There are several variants of the bootstrap:

1. Nonparametric Bootstrap
  - No underlying population distribution is assumed
  - Most commonly used method

2. Smoothed Bootstrap
  - Smooths the sample distribution and then samples from it

3. Parametric Bootstrap
  - Equivalent to maximum likelihood
  - Assumes that the statistic has a particular parametric form (such as the normal distribution)
  - Of little help in real problems

4. Bayesian Bootstrap
  - Rather than give equal probability of being selected, assigns different probabilities to each case based on some prior knowledge

Jackknifing

- Jackknife resampling also resamples the data
- It differs from bootstrapping in that rather than take several new random samples of the data with replacement, the jackknife procedure resamples the data by ***randomly taking a single observation out***
  - In other words, new estimates of the statistic are calculated with a different observation deleted each time.
- Unless the sample is very large the ***number of jackknife samples*** used is usually equal to the number of cases in the original sample
- The average of the new estimates is then calculated to find the jackknife estimate
- Simulation studies generally show that the jackknife works well for robust estimators of location, but not as well as the bootstrap for estimating the standard deviation (see Chernick, 1982:103)

Bootstrapping the Mean: A simple example

- Imagine, unrealistically, that we are interested in finding the confidence interval for a mean from a sample of only 4 observations. Assume that we are interested in the difference in income between husbands and wives
  - We have four cases, with the following income differences (in $1000s): 6, -3, 5, 3, for a mean of 2.75, and standard deviation of 4.031
- From classical theory we can calculate the confidence interval:

$$\begin{aligned} u &= \bar{Y} \pm t_{n-1,.025}\frac{S}{\sqrt{n}} \\ &= 2.75 \pm 4.30 \times \frac{4.031}{\sqrt{4}} \\ &= 2.75 \pm 4.30 \times 2.015 \\ &= 2.75 \pm 8.66 \end{aligned}$$

- Now we'll compare this confidence interval to one found using bootstrapping

Defining the Random Variable

- The first thing that bootstrapping does is estimate the population distribution of Y from the four observations in the sample
- In other words, the random variable Y* is defined:

y^*	$p^*(y^*)$
6	.25
-3	.25
5	.25
3	.25

- The mean of Y* is then simply the mean of the sample:

$$\begin{aligned} E^*(Y^*) &= \sum y^* p(y^*) \\ &= 2.75 \\ &= \bar{Y} \end{aligned}$$

The *Sample* as the *Population (1)*

- We now treat the sample as if it were the population, and resample from it
- In this case we take ***all possible samples with replacement,*** meaning that we take n^n=4^4=256 different samples
- Since we found all possible samples, the mean of these samples is simply the original mean
- The standard error of Y-bar from these samples is:

$$SE^*(\bar{Y}^*) = \sqrt{\frac{\sum_{b=1}^{n^n}(\bar{Y}_b^* - \bar{Y})^2}{n^n}} = 1.745$$

We now make an adjustment for the sample size:

$$\widehat{SE(\bar{Y})} = \sqrt{\frac{n}{n-1}}SE^*(\bar{Y}*) = \sqrt{\frac{4}{3}} \times 1.745 = 2.015$$

The *Sample* as the *Population (2)*

- In this example, because we used ***all possible resamples of our sample***, the bootstrap standard error (2.015) is exactly the same as the original standard error
- Still, the same approach can be used for statistics for which we do not have standard error formulas, or we have small sample sizes
- In summary, the following analogies can be made to sampling from the population:
  - Bootstrap observations→ original observations
  - Bootstrap mean → original sample mean
  - Original sample mean→unknown population mean μ
  - ***Distribution of the bootstrap means→ unknown sampling distribution from the original sample***

Characteristics of the Bootstrap Statistic

- The bootstrap sampling distribution around the original estimate of the statistic T is analogous to the sampling distribution of T around the population parameter θ
- The average of the bootstrapped statistics is simply

$$\bar{T}^* = \hat{E}^*(T^*) = \frac{\sum_{b=1}^{R} T_b^*}{R}$$

where R is the number of bootstraps

- The ***bias of T*** can then be seen as its deviation from the bootstrap average (*i.e.*, it estimates T-θ):

$$\hat{B}^* = \bar{T}^* - T$$

- The estimated bootstrap variance of T* is

$$\hat{V}^*(T^*) = \frac{\sum_{b=1}^{R}(T_b^* - \bar{T}^*)^2}{R-1}$$

Bootstrapping with Larger Samples

- The larger the sample is, the more effort it is to calculate the bootstrap estimates
  - With large sample sizes the possible number of bootstrap samples n^n gets very large impractical (*e.g.*, it would be foolish to calculate 1000^{1000} bootstrap samples)
  - Typically we want to take somewhere between 1000 and 2000 bootstrap samples in order to find a confidence interval of a statistic
- After calculating the standard error we can easily find the confidence interval. Three commonly used methods are:
  1. ***Normal Theory Intervals***
  2. ***Percentile Intervals***
  3. ***Bias Corrected Percentile Intervals***

Bootstrap Confidence Intervals (1) *Normal theory intervals*

- Most statistics are asymptotically normally distributed
- Therefore in large samples the bootstrap estimate along with the normal distribution can produce a 100(1-α)% confidence interval for the population parameter θ

$$\theta = \hat{T} \pm z_{\alpha/2}\widehat{SE}^*(\hat{T}^*)$$

- This approach works well for the bootstrap confidence interval but ***only if the bootstrap sampling distribution is approximately normally distributed***
  - In other words, it is important to ***look at the distribution*** before relying on the Normal theory interval

Bootstrap Confidence Intervals (2) *Bootstrap percentile intervals*

- Uses ***percentiles of the bootstrap sampling distribution*** to find the end points of the confidence interval
- With large bootstrap samples *r*, the confidence interval is bounded at the $\alpha/2$ and 1-$\alpha/2$ percentiles

$$\hat{T}^*_{(\text{lower})} < \theta < \hat{T}^*_{(\text{upper})}$$

where $\hat{T}^*_{(\text{lower})}$ and $\hat{T}^*_{(\text{upper})}$ are the ordered boostrap replicates, with the lower equal to $.025 \times r$, and upper qual to $.975 \times r$ for a 95% confidence interval

- These intervals do not assume a normal distribution, but they do not perform well unless we have a large original sample and at least 1000 bootstrap samples

Bootstrap Confidence Intervals (3)
Bias-corrected, accelerated percentile interval (BC_a)

- The BC_a CI adjusts the confidence intervals for bias due to small samples by employing a normalizing transformation through ***two correction factors, Z and A***
- ***Z*** is defined as:

$$Z = \Phi^{-1}\left[\frac{\#_{b=1}^{R}(T_b^* \leq T)}{R+1}\right]$$

where Φ^{-1} is the inverse of the standard normal function, and $\#(T_b^* \leq T)/(R+1)$ is the adjusted proportion of the bootstrap replicates at or below the original sample estimate T of θ

- If T is unbiased, the proportion will be close to .5, meaning that the correction is close to 0.

Bootstrap Confidence Intervals (4)
Bias-corrected, accelerated percentile interval (BC_a)

- To find the A correction factor we start by finding the average of the n ***jackknife values*** $T_{(-i)}$ for the estimate of T:

$$\bar{T} = \sum_{i=1}^{n} T_{(-i)}/n$$

- We then calculate the A correction factor as follows:

$$A = \frac{\sum_{i=1}^{n}\left(T_{(-i)} - \bar{T}\right)^3}{6\left[\sum_{i=1}^{n}\left(T_{(-i)} - \bar{T}^2\right)\right]^{\frac{2}{3}}}$$

- The A correction factor causes the standard error to grow as the sample size decreases

Bootstrap Confidence Intervals (5)
BC_a (c)

- Z and A are then used to calculate:

$$A_{1\text{lower}} = \Phi\left[Z + \frac{Z - z_{\alpha/2}}{1 - A\left(Z - z_{\alpha/2}\right)}\right]$$

$$A_{2\text{upper}} = \Phi\left[Z + \frac{Z + z_{\alpha/2}}{1 - A\left(Z + z_{\alpha/2}\right)}\right]$$

where Φ is the cumulative standard normal distribution function

- R^*A_{1lower} and R^*A_{2upper} are then the lower and upper limits of the confidence interval (R is the number of bootstrap samples)

Bootstrapping Regression Models

- There are two ways to do this:
- ***Random–X Bootstrapping***
  - The regressors are treated as ***random***
  - Thus we select bootstrap samples directly from the observations and calculate the statistic for each bootstrap sample
- ***Fixed-X Bootstrapping***
  - The regressors are treated as ***fixed***—implies that the regression model fit to the data is correct
  - The fitted values of Y are then the expectation of the bootstrap
    - We attach a ***random error*** (usually the resampled residuals) to each Y-hat which produces the fixed-x bootstrap sample, $\mathbf{Y}_b^*$.
    - To obtain bootstrap replications of the coefficients, we regress $\mathbf{Y}_b^*$ on the fixed model matrix for each bootstrap sample

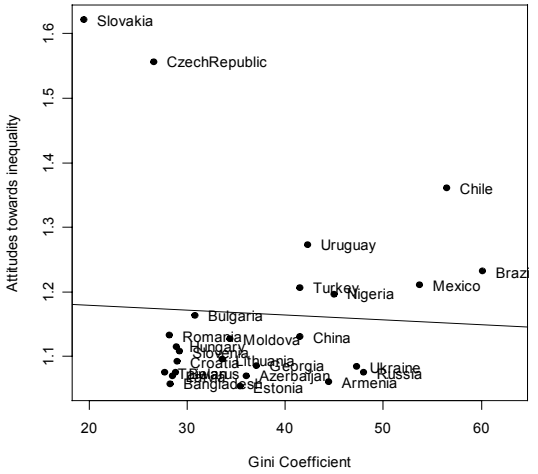
Bootstrapping Regression Example: Inequality data

- Recall that we used robust regression on the Weakliem data, regressing average attitudes towards pay inequality in nondemocracies on the level of income inequality (gini coefficient) in those countries
- An OLS model fit poorly, and diagnostics (Cook's D) indicated that Slovakia and the Czech Republic were highly influential

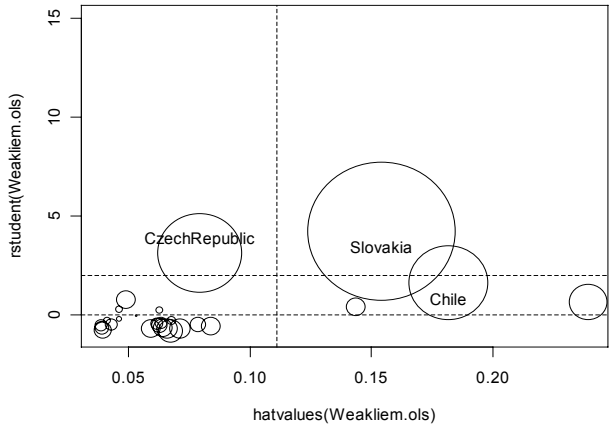
```
> Weakliem.ols<-lm(secpay~gini)
> summary(Weakliem.ols)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.1947705  0.1106550  10.797 1.08e-10 ***
gini        -0.0007586  0.0028843  -0.263    0.795

Residual standard error: 0.1485 on 24 degrees of freedom
Multiple R-Squared: 0.002874,   Adjusted R-squared: -0.03867
```


Bootstrapping Regression Example: Inequality data (2)



Bootstrapping Regression Example: Inequality data (3)



Slovakia(26); Czech Republic (7); Chile (5)

Bootstrapping Regression Example: Inequality data (4)

- ***Robust regression*** (M-Estimation with Huber weights) gave a significantly better fit to the data than the OLS
  - The slope for the Gini coefficient didn't only reach statistical significance, but it actually changed direction
  - The residual standard error also decreased significantly

```
> Weakliem.huber<-rlm(secpay~gini)
> summary(Weakliem.huber)

Coefficients:
            Value   Std. Error t value
(Intercept)  1.0169  0.0563    18.0643
gini         0.0031  0.0015     2.0837

Residual standard error: 0.06363 on 24 degrees of freedom
```

Bootstrapping Regression Example: Inequality data (5)

- Despite that the slope coefficient is now statistically significant, ***the standard errors are not reliable because of the small sample size***
  - Standard errors produced by the `rlm` rely on asymptotic approximations, and thus are not trustworthy for a sample size of 26
- The bootstrap provides an alternative way to get standard errors
- I could proceed to bootstrap the regression in two ways:
  - ***Random-x resampling*** or ***observation resampling***
  - ***Fixed-x resampling***
- I'll consider both in this example

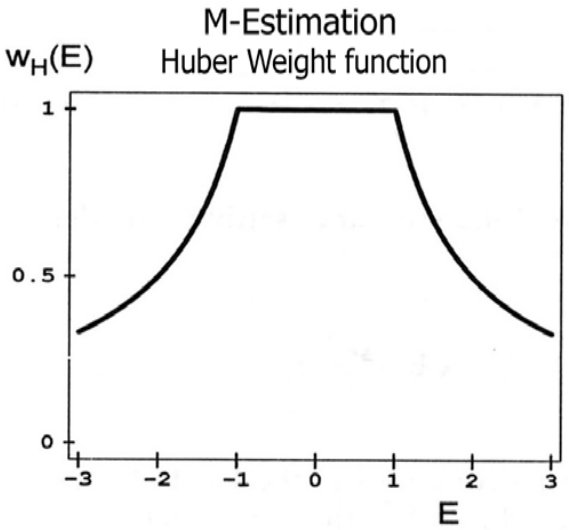
Random-x Resampling (1)

- Also referred to as ***observation resampling***
- Random x-resampling selects R boostrap samples (*i.e.*, resamples) of the ***observations***, fits the regression for each one, and determines the standard errors from the bootstrap distribution
- This is done fairly simply in **R** using the `boot` function of the `boot` library
- The function takes several arguments, but only three are required:
  - `data`: The data to which the bootstrapping will be applied
  - `statistic`: a function that returns the statistic to be bootstrapped (This will usually require that you write a function)
  - `R`: the number of bootstraps to take

Random-x Resampling (2)

- I start by creating a function that extracts the Huber weights
- Recall that the Huber weight behaves like OLS in the centre of the distribution, but similar to absolute values regression on the tales, giving low weight to observations with very extreme residuals

```
> boot.huber<-function(data, indices, maxit=20){
+     data<-data[indices,]
+     #selects the observations in bootstrap sample
+     Weakliem.huber<-rlm(secpay~gini, data=data, maxit=maxit)
+     coefficients(Weakliem.huber)
+     #returns coefficients vector
+     }
```

Random-x Resampling (3)

- The `boot` function in the `boot` package returns the original robust coefficients (from the `rlm` model) plus ***bootstrap estimates of bias and the bootstrap standard errors***

```
> Weakliem.boot<-boot(data=Nondemo,
    statistic=boot.huber, R=1500, maxit=1000)
> Weakliem.boot

ORDINARY NONPARAMETRIC BOOTSTRAP
Bootstrap Statistics :
       original        bias    std. error
t1* 1.016927104  0.0299164207  0.13839081
t2* 0.003057488 -0.0006629876  0.00345363
```

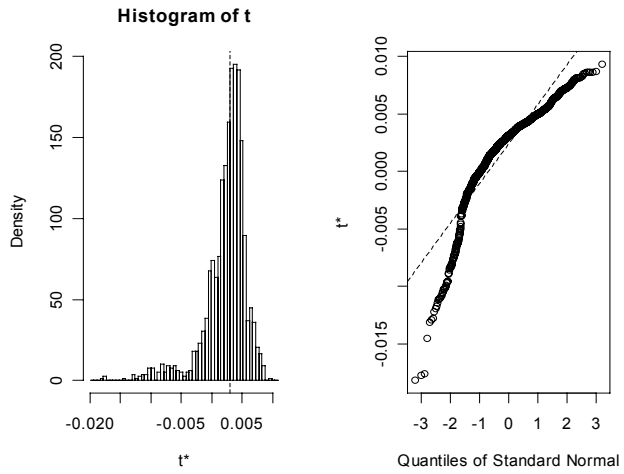
- t1* represents the first coefficient in the model frame (the intercept); t2 is the second (gini).

Random-x Resampling (4)

- We see below that the bootstrap standard error for the Gini coefficient is more than twice as large as the original standard error (.00345 versus .0015)
  - Given that the sample size is extremely small, it is not surprising that there is a difference between the two
- We also see that the bootstrap coefficient has little bias
  - The difference between the average bootstrap coefficient (b=.0031) and the original sample value (b=.00305) is not large (-.00066)
- We now examine the distribution of the bootstrap statistic by simply plotting the `boot` object (`index=2` means plot the second coefficient)

```
> #Plotting bootstrap sample for gini#
> #gives histogram and qq-plot
> plot(Weakliem.boot, index=2)
```


The Bootstrap Distribution



The Bootstrap Distribution (2)

- The dashed line on the previous histogram indicates the value of the statistic from the model fit to the original sample (in this case the coefficient for Gini from the robust regression)
- We can see from both plots that the bootstrap distribution has a strong negative skew, ***reflecting the outlying cases***
  - This suggests that we should use one of the precentile confidence intervals rather than the normal-theory interval
  - Given the small sample size, I choose the bias-corrected, accelerated percentile interval (BC_a)

Bootstrap Confidence Interval

- The `boot.ci` function will return several types of confidence intervals for the statistic
  - I request the normal-theory (`"norm"`), the Percentile interval (`"perc"`), and the bias-corrected, accelerated percentile interval (`"bca"`)
- According to all three measures, the coefficient is not statistically significant (the default for `boot.ci` is a 95% CI)—still, the BC_a is slightly larger and centred differently than the normal

```
> boot.ci(Weakliem.boot, index=2, type=c("norm", "perc", "bca"))
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1500 bootstrap replicates

Intervals :
Level      Normal             Percentile            BCa
95%   (-0.0030,  0.0105 )   (-0.0081,  0.0072 )   (-0.0103,  0.0068 )
Calculations and Intervals on Original Scale
```


Jackknife-after-Bootstrap (1)

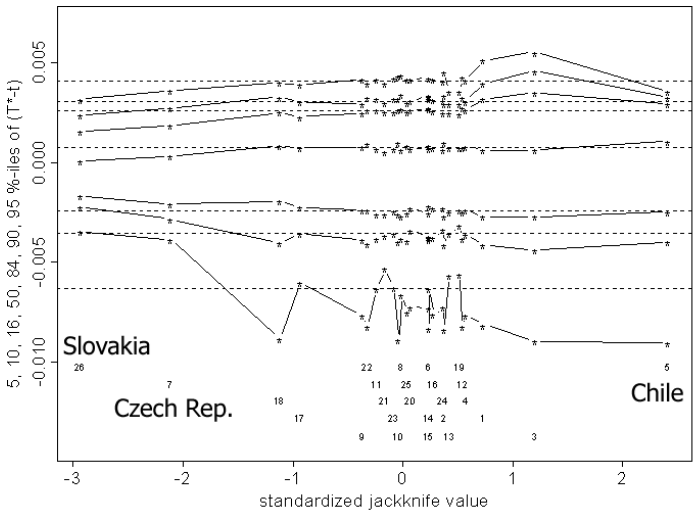
- The ***jackknife-after-bootstrap*** provides a diagnostic of the bootstrap by allowing us to ***examine what would happen to the distribution if particular cases were deleted***
  - Given that we know that there are outliers (evidence of which was also seen in the bootstrap distribution plots) this diagnostic is important
- The jackknife-after-bootstrap plot is produced using the `jack.after.boot` function (`boot` package). Once again, the `index=2` argument specifies that I want the plot for the second coefficient (the intercept is always the first)

```
> jack.after.boot(Weakliem.boot, index=2)
```


Jackknife-after-Bootstrap Plot

- The `jack.after.boot` plot is constructed as follows:
  - The horizontal axis represents the ***standardized jackknife value***
  - The vertical axis represents various quantiles of the bootstrap statistics
    - Horizontal lines in the graph represent the bootstrap distribution at the various quantiles
  - Case numbers are labelled at the bottom of the graph so that each observation can be identified—the asterisks above them represent the influence of the case
- We can see clearly from the following graph that Slovakia, the Czech Republic and Chile had a strong influence on the bootstrap estimates—Slovakia and the Czech Republic decreased the coefficient, but Chile increases it

Jackknife-after-Bootstrap (2)



Statistical significance after removing outliers

- ***Why did we get much larger standard errors with the bootstrap?***
  - The output for the robust `rlm` function shows standard errors based on asymptotic theory—in other words, the sample size was too small for this to apply
  - Robust regression down-weights the outliers, in essence nearly deleting them from the analysis and then calculates standard errors
  - In the resampling process the influential cases may get replicated several times in one sample, and thus overwhelm the robust regression—*i.e.,* ***it will not eliminate them all***

Fixed-x Resampling (1) (residual resampling)

- The observations in the Inequality data represent countries. Since we can't really resample countries, it is sensible to think of them as fixed
- Fixed-x resampling involves generating bootstrap replications with a fixed model-matrix $\boldsymbol{X}$
- We proceed in the following fashion:
  - Treat the fitted values of Y from the model as the expectation of the response from the bootstrap.
  - We then attach a random error to each Y-hat which produces the fixed-x bootstrap sample, $\mathbf{Y}_b$*.
  - We then regress $\mathbf{Y}_b$* on the fixed model matrix to obtain bootstrap replications of the coefficients

Fixed-x Resampling (2) R-script

- Fixed-x Resampling is also done relatively easily in R
- We must start, however, by defining the fitted values, residuals and the model matrix

```
> boot.huber.fixed<-function(data, indices, maxit=20){
+     fit<-fitted(Weakliem.huber)
+     e<-residuals(Weakliem.huber)
+     X<-model.matrix(Weakliem.huber)
+     y<-fit+e[indices]
+     mod<-rlm(y~X-1, maxit=maxit)
+     coefficients(mod)
+     }
```

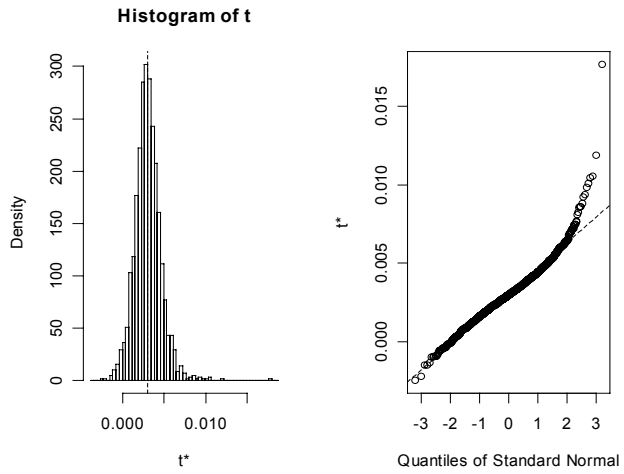

Fixed-x Resampling (3)

- The standard error in the original data was .0015, essentially the same as the fixed-x bootstrap. This is significantly less than the standard error of .00333 found in the random X bootstrap
- This result is also confirmed in the distribution of the bootstrap—although there is still a skew, is no where near as much as in the random-x resampling

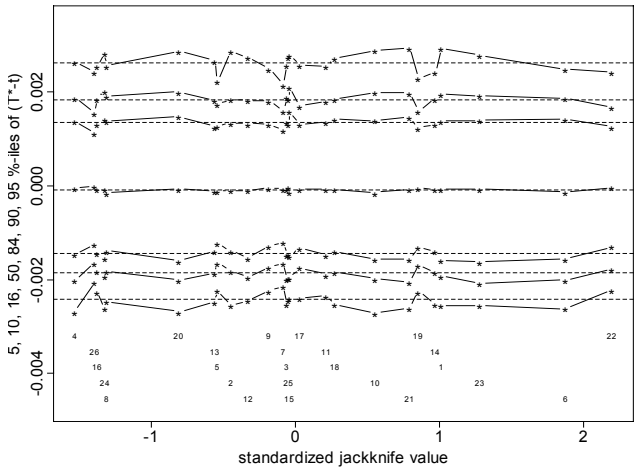
```
> Weakliem.huber.fixed<-boot(Nondemo, boot.huber.fixed,
    R=1500, maxit=100)
> Weakliem.huber.fixed

Bootstrap Statistics :
       original       bias    std. error
t1* 1.016927104 3.950600e-03 0.062763490
t2* 0.003057488 9.373759e-06 0.001625366
```

The Bootstrap Distribution



Fixed-x Resampling



Fixed versus not Fixed: Why do the answers differ?

- Fixed-x resampling enforces the assumption that errors are randomly distributed by ***resampling the residuals from a common distribution***
- As a result, if the model is not specified correctly—*i.e.*, there is unmodeled nonlinearity, non-constant error variance, or outliers—***these attributes do not carry over to the bootstrap samples***
- In this case, then, where there are outliers, it makes sense to stick with the random x resampling
- The effects of outliers was clear in the random-X case, but not with the fixed-X bootstrap

When Might Bootstrapping Fail?

1. Incomplete Data

- Since we are trying to estimate a population distribution from the sample data, we must assume that missing data are not problematic. It is acceptable, however, to use bootstrapping if multiple imputation is used beforehand.

2. Dependent Data

- Simple bootstrapping imposes mutual dependence on the Y_j, and thus their joint distribution is $F(y_1) \times \cdots \times F(y_n)$. This is incorrect for dependent data. As a result, bootstrapping should not be used in these circumstances.

3. Outliers and Influential Cases

- If obvious outliers are found, they should be removed or corrected before performing the bootstrap. We do not want the simulations to depend crucially on particular observations.

Cross-Validation (1)

- If no two observations have the same Y, a p-variable model fit to p+1 observations will fit the data perfectly
- This implies, then, that discarding much of a dataset can lead to better fitting models
  - Of course, this will lead to biased estimators that are likely to give quite different predictions on another dataset
- Model validation allows us to assess whether the model is likely to predict accurately on future observations or observations not used to develop the model
- Three major factors that may lead to model failure are: (1) over-fitting, (2) changes in measurement, (3) changes in sampling.
  - ***External validation*** involves retesting the model on new data collected at a different point in time or from a different population
  - ***Internal validation*** (or ***cross-validation***) involves fitting and evaluating the model carefully using only one sample

Cross-Validation (2)

- A basic but powerful tool for statistical model building
- Cross-validation is similar to bootstrapping in that it resamples the original sample
- Basic form involves randomly dividing the sample into two subsets:
  - The first subset of the data (***screening or training sample***) is used to select or estimate a statistical model
  - We then test our findings on the second subset (***confirmatory or test sample***)
- Can be helpful in avoiding capitalizing on chance and over-fitting the data—*i.e.*, findings from the first subset may not be confirmed by the second subset
- Crossvalidation is often extended to use several subsets (either a preset number chosen by the researcher or ***leave-one-out*** cross-validation)

Cross-Validation (3) Several Subsets

- The data are split into k subsets (usually $3 \leq k \geq 10$; but can also use ***leave-one-out*** cross-validation)
- Each of the subsets are left out in turn, with the regression run on the remaining data
- ***Prediction error*** is then calculated as the sum of the squared errors:

$$RSS = \sum (Y_i - \hat{Y}_i)^2$$

- We choose the model with the smallest average "error" (*i.e.*, mean squared error)

$$MSE = \frac{\sum (Y_i - \hat{Y}_i)^2}{n}$$

- We could also look to the model with the largest average R^2

Cross-Validation (4): Other Considerations

- ***Deciding on the number of observations to leave out from each fit***
  - There is no rule on how many cases to leave out, but Efron (1983) suggests that grouped cross-validation (with approximately 10% of the data left out each time) is better than leave-one-out cross-validation
- ***Number of repetitions***
  - Harrell (2001:93) suggests that one may need to leave 1/10 of the sample out 200 times to get accurate estimates
- ***Cross-validation does not validate the complete sample***
  - External validation, on the other hand, validates the model on a new sample
  - Of course, limitations in resources usually prohibit external validation in a single study

Cross-Validation in R (1)

- Cross-validation is done easily using the `validate` function in the `Design` package:

```
> library(car)
> library(Design)
> data(Prestige)
> mod1<-ols(prestige~income+type+education, x=TRUE, y=TRUE, data=Prestige)
> validate(mod1, method="crossvalidation", B=4)
          index.orig   training      test     optimism index.corrected n
R-square   0.8348574  0.8396671  0.782345   0.05732221       0.7775352 4
MSE       47.7681243 46.0081432 60.095004 -14.08686103      61.8549853 4
```

- Notice that you must use the model functions specific to the `Design` package (*i.e.*, `ols`) for the function to work
- Here I chose to split the sample into 4 (1 ***training*** sample and 2 *test* samples) with the `B=4` argument
- Specifying the argument `pr=T` after the `method` argument will prints the results of each repetition

Cross-Validation in R (2)

- You must be careful not to over-fit the model—*i.e.*, you must make sure that the sub-samples are large enough
- Below are perform crossvalidation using the default number of sample splits (B=40)
- Notice that the R^2 for the test sample is negative—this implies that predictions from the model are worse than from using Y-bar alone. Simply put, it indicates over-fitting and suggests reducing the number of samples (after all, we only have 102 observations).

```
> library(car)
> library(Design)
> data(Prestige)
> mod1<-ols(prestige~income+type+education, x=TRUE, y=TRUE, data=Prestige)
> validate(mod1, method="crossvalidation")
          index.orig training        test    optimism index.corrected  n
R-square   0.8348574  0.83478 -41.8923685  42.7271485     -41.8922911 40
MSE       47.7681243 47.69236  54.2520728  -6.5597087      54.3278329 40
```


Summary & Conclusions

- Resampling techniques are powerful tools for estimating standard errors from small samples or when the statistics we are using do not have easily determined standard errors
- Bootstrapping involves taking 'new' random samples (with replacement) from the original data
  - We then calculate bootstrap standard errors and statistical tests from the average of the statistic from the bootstrap samples
- Jackknife resampling takes new samples of the data by omitting each case individually and recalculating the statistic each time
- Cross-validation randomly splits the sample into two groups, comparing the model results from one sample to the results from the other

Further Reading

Chernick, M. R. (1999) *Bootstrap Methods: A Practitioner's Guide.* New York: John Wiley & Sons.

Davison, A.C. and D.V. Hinkley (1997) *Bootstrap methods and their application*. Cambridge University Press.

Efron, B. (1983) "Estimating the error rate of a prediction rule: Improvement on cross-validation," *Journal of the American Statistical Association*, 81:461-470.

Harrell, F.E. Jr. (2001) *Regression Modeling Strategies. With Applications to Linear Models, Logistic Regression, and Survival Analysis.* New York: Springer.