Regression III

Bootstrapping

Dave Armstrong

Goals of the Lecture

- Introduce the general idea of Resampling - techniques to resample from the original data
- An extended example of bootstrapping local polynomial regression models.

Resampling: An Overview

- Resampling techniques sample from the original dataset
- Some of the applications of these methods are:
  - to compute standard errors and confidence intervals (either when we have small sample sizes, dubious distributional assumptions 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 GAMs)
- For the most part, this lecture will discuss resampling techniques in the context of computing confidence intervals and hypothesis tests for regression analysis.

Notes

Type notes here...

Resampling and Regression: A Caution

There is no need whatsoever for bootstrapping in regression analysis if the OLS assumptions are met

- In such cases, OLS estimates are unbiased and maximally efficient.

There are situations, however, where we cannot satisfy the assumptions and thus other methods are more helpful

- Robust regression (such as MM-estimation) often provides better estimates than OLS in the presence of influential cases, but only has reliable SEs asymptotically.
- Local polynomial regression is often "better" (in the RSS sense) in the presence of non-linearity, but because of the unknown df, only has approximate sampling distribution.

Notes

Type notes here...

Bootstrapping: General Overview

If we assume that a random variable X or statistic has a particular population value, 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 - Cronbach's α doesn't have an exact sampling distribution
- 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 standard errors and confidence intervals (sets) without making assumptions about the distributions that give rise to the data

Notes

Type notes here...

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"

Steps:

- Take several new samples from the original sample, calculating the statistic each time
- Calculate the average and standard error (and maybe quantiles) from the empirical distribution of the bootstrap samples
- We apply principles of inference similar to those employed when sampling from the population

Notes

Type notes here...

Bootstrapping: General Overview (3)

There are several Variants of the bootstrap. The two we are most interested in are:

- Nonparametric Bootstrap
  - No underlying population distribution is assumed
  - Most commonly used method
- Parametric Bootstrap
  - Assumes that the statistic has a particular parametric form (e.g., normal)

Notes

Type notes here...

Bootstrapping the Mean

- Imagine, unrealistically, that we are interested in finding the confidence interval for the mean of a sample of only 4 observations
- Specifically, assume that we are interested in the difference in income between husbands and wives
- We have four cases, with the following mean differences (in \$~1000's): 6, -3, 5, 3, for a mean of 2.75 and a standard deviation of 4.031

Notes

Type notes here...

Classical Theory Standard Error

- From classical theory, we can calculate the standard error:

$$\begin{aligned} SE &= \frac{S_x}{\sqrt{n}} \\ &= \frac{4.031129}{\sqrt{4}} \\ &= 2.015564 \end{aligned}$$

- Now we'll compare the confidence interval to the one calculated using bootstrapping

Notes

Type notes here...

Defining the Random Variable

Y^*	$p^*(Y^*)$
6	0.25
-3	0.25
5	0.25
3	0.25

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

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

Notes

Type notes here...

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 $n^n = 256$ different samples
- Since we found all possible samples, the mean of these samples is simply the original mean
- We then determine the standard error of $\bar{Y}$ from these samples

$$SE^*(\bar{Y}) = \sqrt{\frac{\sum_{b=1}^{n^n}(\bar{Y}_b^* - \bar{Y})^2}{n^n}} = 1.74553$$

We now adjust for the sample size:

$$\hat{SE}(\bar{Y}) = \sqrt{\frac{n}{n-1}} SE^*(\bar{Y}^*) = \sqrt{\frac{4}{3}} \times 1.74553 = 2.015564$$

Notes

Type notes here...

The Sample as the Population (2)

- In this example, because we used all possible resamples of our sample, the bootstrap standard error (2.015564) is exactly the same as the original standard error
- This 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 $\rightarrow$ original observations
  - Bootstrap Mean $\rightarrow$ original sample mean
  - Original sample mean $\rightarrow$ unknown population mean μ
  - Distribution of the bootstrap means $\rightarrow$ unknown sampling distribution from the original sample

Notes

Type notes here...

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:

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

where R is the number of bootstraps

Notes

Type notes here...

Bias and Variance

The bias of T can be seen as its deviation from the bootstrap average (i.e., it estimates $T - \theta$)

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

The estimated bootstrap variance of T^* is:

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

Notes

Type notes here...

Bootstrapping with Larger Samples

The larger the sample, 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 and impractical (e.g., it would take a long time 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 methods are commonly used

- Normal Theory Intervals
- Percentile Intervals
- Bias Corrected, accelerated Percentile Intervals

Notes

Type notes here...

Evaluating Confidence Intervals

Accuracy: how quickly do coverage errors go to zero?

- $\text{Prob}\{\theta < \hat{T}_{lo}\} = \alpha$ and $\text{Prob}\{\theta > \hat{T}_{up}\} = \alpha$
  - $\frac{1}{n}$ (second-order accurate)
  - $\frac{1}{\sqrt{n}}$ (first-order accurate)

Transformation Respecting:

- For any monotone transformation of θ, $\phi = m(\theta)$, can we obtain the right confidence interval on $\hat{\phi}$ with the confidence intervals on $\hat{\theta}$ mapped by $m()$? E.g.,

$$[\hat{\phi}_{lo}, \hat{\phi}_{up}] = [m(\hat{\theta}_{lo}), m(\hat{\theta}_{up})]$$

Notes

Type notes here...

Normal Theory Intervals

- Many statistics are asymptotically normally distributed
- Therefore, in large samples, we may be able to use a normality assumption to characterize the bootstrap distribution. E.g.,

$$\hat{T}^* \sim N(\hat{T}, \hat{se}^2)$$

where $\hat{se}$ is $\sqrt{\hat{V}(T^*)}$

- This approach works well for the bootstrap confidence interval, but only if the bootstrap sampling distribution is approximately normally distributed

Notes

Type notes here...

Percentile Intervals

Use percentiles of the bootstrap sampling distribution to find the end-points of the confidence interval

- The $(1-2\alpha)$ percentile interval can be approximated with:

$$\left[\hat{T}_{\%,lo}, \hat{T}_{\%,up}\right] \approx \left[T_B^{*(\alpha)}, T_B^{*(1-\alpha)}\right]$$

where $T_B^{*(\alpha)}$ and $T_B^{*(1-\alpha)}$ are the ordered (B) bootstrap replicates such that $100\alpha\%$ of them fall below the former and $100\alpha\%$ of them fall above the latter.

- 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

Notes

Type notes here...

BCa Intervals

The BC_a CI adjusts the confidence intervals for bias due to small samples by employing a normalizing transformation through two correction factors.

- Using strict percentile intervals, $\left[\hat{T}_{lo}, \hat{T}_{up}\right] \approx \left[T_B^{*(\alpha)}, T_B^{*(1-\alpha)}\right]$
- Here, $\left[\hat{T}_{lo}, \hat{T}_{up}\right] \approx \left[T_B^{*(\alpha_1)}, T_B^{*(\alpha_2)}\right]$
- $\alpha_1 \neq \alpha$ and $\alpha_2 \neq (1 - \alpha_1)$

$$\alpha_1 = \mathbf{\Phi}\left(\hat{z}_0 + \frac{\hat{z}_0 + z^{(\alpha)}}{1 - \hat{a}(\hat{Z}_0 + z^{(\alpha)})}\right)$$

$$\alpha_2 = \mathbf{\Phi}\left(\hat{z}_0 + \frac{\hat{z}_0 + z^{(1-\alpha)}}{1 - \hat{a}(\hat{Z}_0 + z^{(1-\alpha)})}\right)$$

Notes

Type notes here...

Bias correction

$$\hat{z}_0 = \Phi^{-1}\left(\frac{\#\left\{T_b^* \leq T\right\}}{B}\right)$$

- This just gives the inverse of the normal CDF for proportion of bootstrap replicates less than T.
- Note that if $\#\left\{T_b^* \leq T\right\}$ = 0.5, then $\hat{z}_0 = 0$.
- If T is unbiased, the proportion will be close to 0.5, meaning that the correction is close to 0.

Notes

Type notes here...

Acceleration:

$$\hat{a} = \frac{\sum_{i=1}^{n} \left(T_{(\cdot)} - T_{(i)}\right)^3}{6\left\{\sum_{i=1}^{n} \left(T_{(\cdot)} - T_{(i)}\right)^2\right\}^{\frac{3}{2}}}$$

- $T_{(i)}$ is the calculation of the original estimate T with each observation i jacknifed out in turn.
- $T_{(\cdot)}$ is $\sum_{i=1}^{n} \frac{T_{(i)}}{n}$
- The acceleration constant corrects the bootstrap sample for skewness.

Notes

Type notes here...

Evaluation Confidence Intervals

	Normal	**Percentile**	BC_a
Accuracy	1^{st} order	1^{st} order	2^{nd} order
Transofrmation-respecting	No	Yes	Yes

Notes

Type notes here...

Number of BS Samples

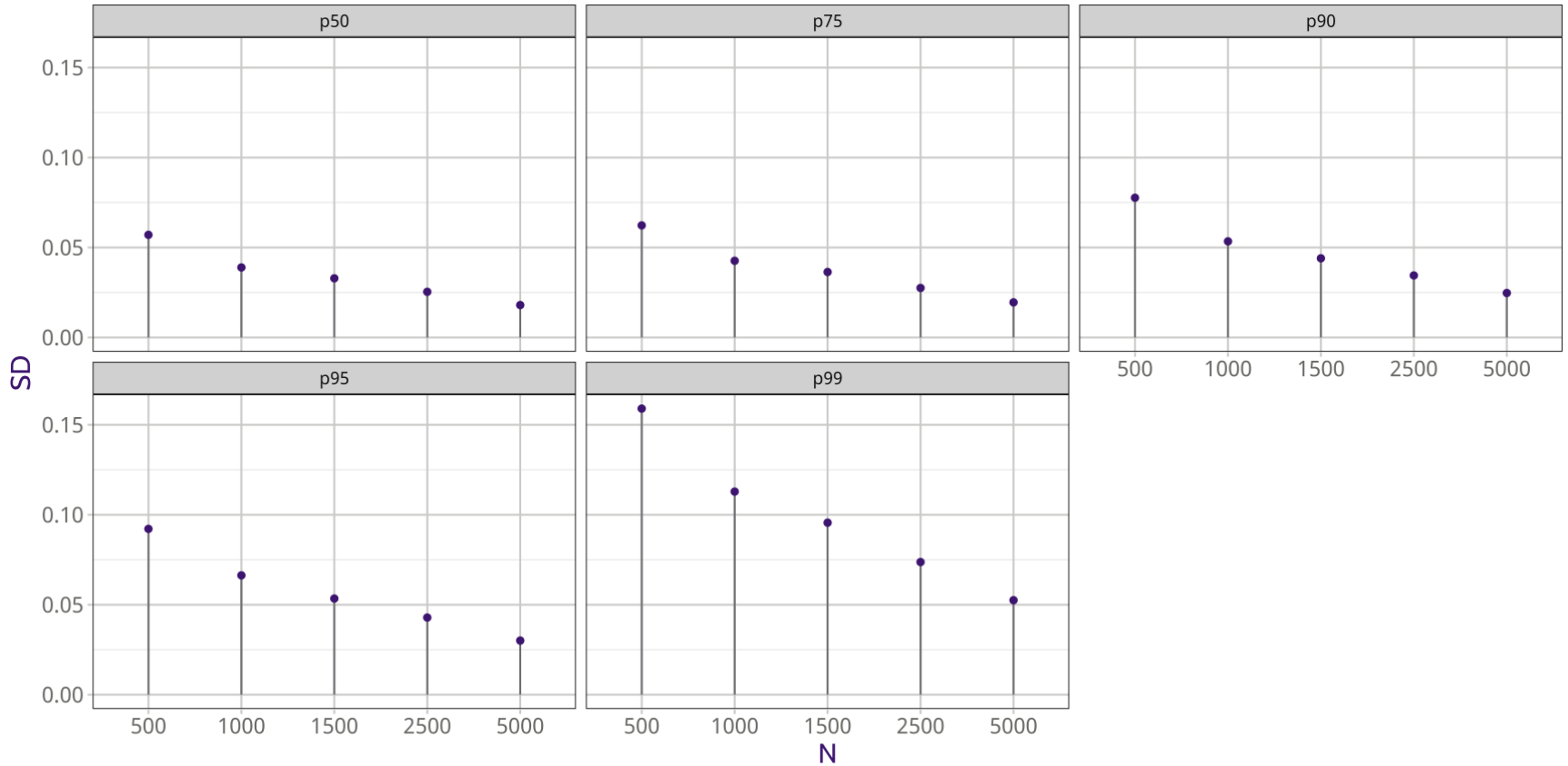
Now, we can revisit the idea about the number of bootstrap samples.

- The real question here is how precise to you want estimated quantities to be.
- Often, we're trying to estimate the 95% confidence interval (so the 2.5^{th} and 97.5^{th} percentiles of the distribution).
- Estimating a quantity in the tail of a distribution with precision takes more observations.

Notes

Type notes here...

Demonstration



Notes

Type notes here...

Bootstrapping the Median

```
set.seed(123)
n <- 25
x <- rchisq(n,1,1)
library(boot)
set.seed(123)
med.fun <- function(x, inds){
    med <- median(x[inds])
    med
}
boot.med <- boot(x, med.fun, R=1000)
median(x)
```

```
## [1] 1.206853
```

```
meds <- boot.med$t
t(summary(meds))
```

```
##
##       V1 Min.   :0.2838   1st Qu.:0.6693   Median :1.2069   Mean   :1.2799
##
##       V1 3rd Qu.:1.7855   Max.   :3.7749
```

Notes

Type notes here...

Make CIs

- Normal Theory CI:

```
se <- sd(meds)
bias <- mean(meds) - median(x)
norm.ci <- median(x)- bias  + qnorm((1+.95)/2)*se*c(-1,1)
```

- Percentile CI:

```
med.ord <- meds[order(meds)]
pct.ci <- med.ord[c(25,975)]
pct.ci <- quantile(meds, c(.025,.975))
```

Notes

Type notes here...

Make CIs (2)

- BC_a CI:

```
zhat <- qnorm(mean(meds < median(x)))
medi <- sapply(1:n, function(i)median(x[-i]))
Tdot <- mean(medi)
ahat <- sum((Tdot-medi)^3)/(6*sum((Tdot-medi)^2)^(3/2))
zalpha <- 1.96
z1alpha <- -1.96
a1 <- pnorm(zhat + ((zhat + zalpha)/(1-ahat*(zhat + zalpha))))
a2 <- pnorm(zhat + ((zhat + z1alpha)/(1-ahat*(zhat + z1alpha))))
bca.ci <- quantile(meds, c(a1, a2))
a1 <- floor(a1*1000)
a2 <- ceiling(a2*1000)
bca.ci <- med.ord[c(a2, a1)]
```

Notes

Type notes here...

Looking at CIs

```
mat <- rbind(norm.ci, pct.ci, bca.ci)
rownames(mat) <- c("Normal", "Percentile", "BCa")
colnames(mat) <- c("Lower", "Upper")
# print(xtable(mat), digits=4, type="html",
# html.table.attributes = "border = 0")
knitr::kable(mat, digits=3)
```

	Lower	Upper
Normal	-0.056	2.324
Percentile	0.473	2.829
BCa	0.473	2.211

```
boot.ci(boot.med)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.med)
##
## Intervals :
## Level      Normal              Basic
## 95%   (-0.056,  2.324 )   (-0.415,  1.940 )
##
## Level     Percentile            BCa
## 95%   ( 0.473,  2.829 )   ( 0.473,  2.211 )
## Calculations and Intervals on Original Scale
```

Notes

Type notes here...

BS Difference in Correlations

We can bootstrap the difference in correlations as well.

```
set.seed(123)
data(Mroz, package="carData")
cor.fun <- function(dat, inds){
    tmp <- dat[inds, ]
    cor1 <- with(tmp[which(tmp$hc == "no"), ],
    cor(age, lwg, use="complete"))
    cor2 <- with(tmp[which(tmp$hc == "yes"), ],
    cor(age, lwg, use="complete"))
    c(cor1, cor2, cor2-cor1)
}
boot.cor <- boot(Mroz, cor.fun, R=2000, strata=Mroz$hc)
```

Notes

Type notes here...

Bootstrap Confidence Intervals

```
library(DAMisc)
out1 <- tidy_boot_ci(boot.cor, type="perc",
             term_names = c("R no HC", "R HC", "Difference"))
out1
```

```
## # A tibble: 3 x 4
##   term       estimate conf.low conf.high
##   <chr>         <dbl>    <dbl>     <dbl>
## 1 R no HC    -0.00213 -0.0989     0.0823
## 2 R HC        0.0974  -0.00827    0.211
## 3 Difference  0.0996  -0.0442     0.241
```

```
out2 <- tidy_boot_ci(boot.cor, type="bca",
             term_names = c("R no HC", "R HC", "Difference"))
out2
```

```
## # A tibble: 3 x 4
##   term       estimate conf.low conf.high
##   <chr>         <dbl>    <dbl>     <dbl>
## 1 R no HC    -0.00213 -0.0981     0.0831
## 2 R HC        0.0974  -0.00994    0.209
## 3 Difference  0.0996  -0.0460     0.240
```

Notes

Type notes here...

Random-X Bootstrap

The Random-X bootstrap proceeds as follows:

- The regressors are treated as random
- Thus, we select bootstrap samples directly from the observations and calculate the statistic for each bootstrap sample
- also called bootstrap pairs

Notes

Type notes here...

Fixed-X Bootstrap

The Fixed-X Bootstrap proceeds as follows:

- 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 resampled from the residuals) to each $\hat{Y}$ 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

Notes

Type notes here...

WVS Data

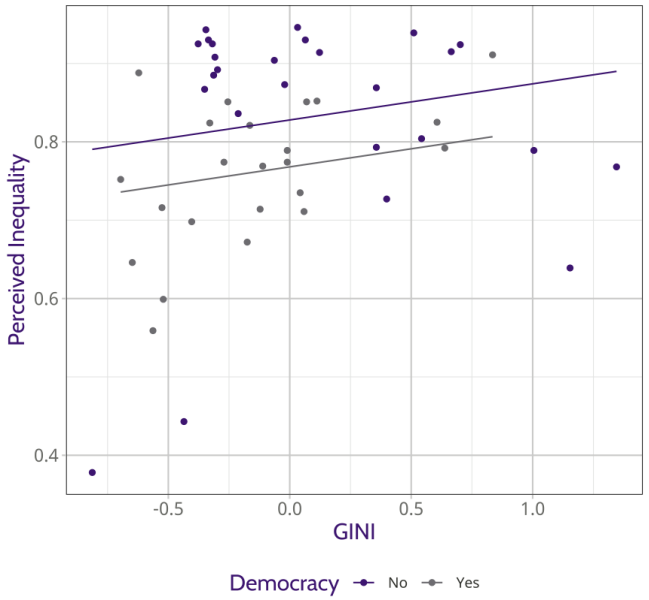
- `secpay`: Imagine two secretaries, of the same age, doing practically the same job. One finds out that the other earns considerably more than she does. The better paid secretary, however, is quicker, more efficient and more reliable at her job. In your opinion, is it fair or not fair that one secretary is paid more than the other?
- `gini` Observed inequality captured by GINI coefficient at the country level.
- `democrat` Coded 1 if the country maintained "partly free" or "free" rating on Freedom House's Freedom in the World meausre continuously from 1980 to 1995.

Notes

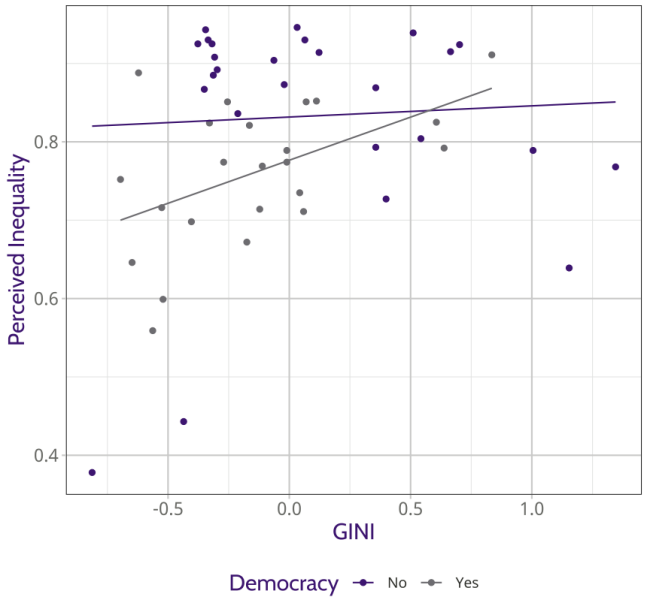
Type notes here...

Models

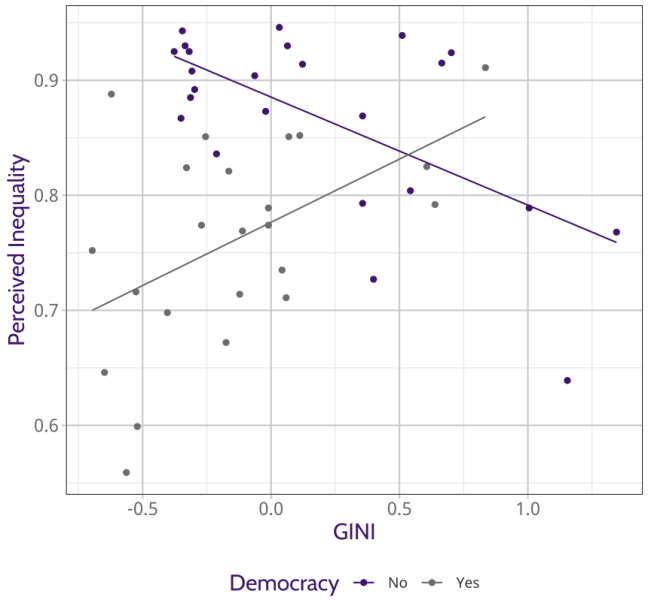
Additive, All Obs.



Multiplicative, All Obs



Multiplicative, No Outliers



Notes

Type notes here...

Bootstrapping Regression: Example (1)

- We fit a linear model of secpay on the interaction of gini and democracy.

```
library(car)
dat <- read.csv("http://quantoid.net/files/reg3/weakliem.txt", header=T)
dat <- dat[-c(25,49), ]
dat <- dat[complete.cases(
  dat[,c("secpay", "gini", "democrat")]), ]
mod <- lm(secpay ~ gini*democrat,data=dat)
S(mod, brief=TRUE)
```

```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.059234   0.059679  17.749  < 2e-16 ***
## gini          -0.004995   0.001516  -3.294  0.00198 **
## democrat      -0.486071   0.088182  -5.512 1.86e-06 ***
## gini:democrat  0.010840   0.002476   4.378 7.53e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.07118 on 43 degrees of freedom
## Multiple R-squared: 0.5176
## F-statistic: 15.38 on 3 and 43 DF,  p-value: 6.087e-07
##     AIC     BIC
## -109.20  -99.95
```

Notes

Type notes here...

Fixed-X Bootstrapping

- If we are relatively certain that the model is the "right" model (i.e., it is properly specified), then this makes sense.
- The `Boot` function in the `car` package will bootstrap regression models directly. - The code for today also shows the way to do this with the `boot` function from the `boot` package - a more flexible, but also more complicated method.

```
library(car)
boot.reg1 <- Boot(mod, R=500, method="residual")
```

```
est1 <- tidy_boot_ci(boot.reg1, type="perc",
             term_names = names(coef(mod)))
est2 <- tidy_boot_ci(boot.reg1, type="bca",
             term_names = names(coef(mod)))
est1
```

```
## # A tibble: 4 x 4
##   term          estimate conf.low conf.high
##   <chr>            <dbl>    <dbl>     <dbl>
## 1 (Intercept)     1.06     0.944     1.18
## 2 gini           -0.00500 -0.00808  -0.00193
## 3 democrat       -0.486   -0.665    -0.321
## 4 gini:democrat   0.0108   0.00620   0.0155
```

```
est2
```

```
## # A tibble: 4 x 4
##   term          estimate conf.low conf.high
##   <chr>            <dbl>    <dbl>     <dbl>
## 1 (Intercept)     1.06     0.937     1.17
## 2 gini           -0.00500 -0.00802  -0.00177
## 3 democrat       -0.486   -0.654    -0.291
## 4 gini:democrat   0.0108   0.00572   0.0154
```

Notes

Type notes here...

Random-X resampling (1)

- Random-X resampling is also referred to as observation resampling
- It selects B bootstrap samples (i.e., resamples) of the observations, fits the regression for each one, and determines the standard errors from the bootstrap distribution

```
boot.reg2 <- Boot(mod, R=500, method="case")
```

```
est1a <- tidy_boot_ci(boot.reg2, type="perc",
             term_names = names(coef(mod)))
est2a <- tidy_boot_ci(boot.reg2, type="bca",
             term_names = names(coef(mod)))
est1a
```

```
## # A tibble: 4 x 4
##   term          estimate conf.low conf.high
##   <chr>            <dbl>    <dbl>     <dbl>
## 1 (Intercept)    1.06     0.948     1.17
## 2 gini          -0.00500 -0.00817  -0.00164
## 3 democrat      -0.486   -0.665    -0.296
## 4 gini:democrat  0.0108   0.00558   0.0158
```

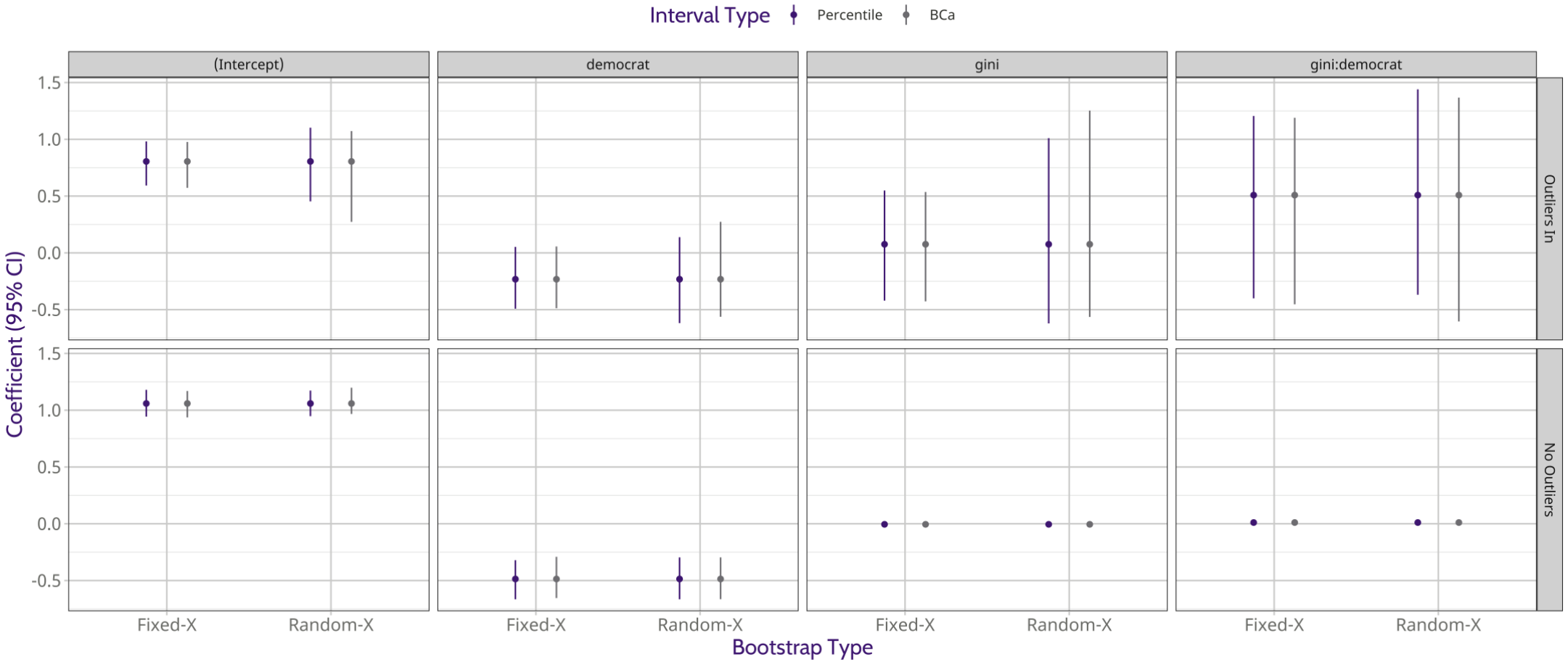
```
est2a
```

```
## # A tibble: 4 x 4
##   term          estimate conf.low conf.high
##   <chr>            <dbl>    <dbl>     <dbl>
## 1 (Intercept)    1.06     0.967     1.20
## 2 gini          -0.00500 -0.00945  -0.00228
## 3 democrat      -0.486   -0.664    -0.296
## 4 gini:democrat  0.0108   0.00557   0.0157
```

Notes

Type notes here...

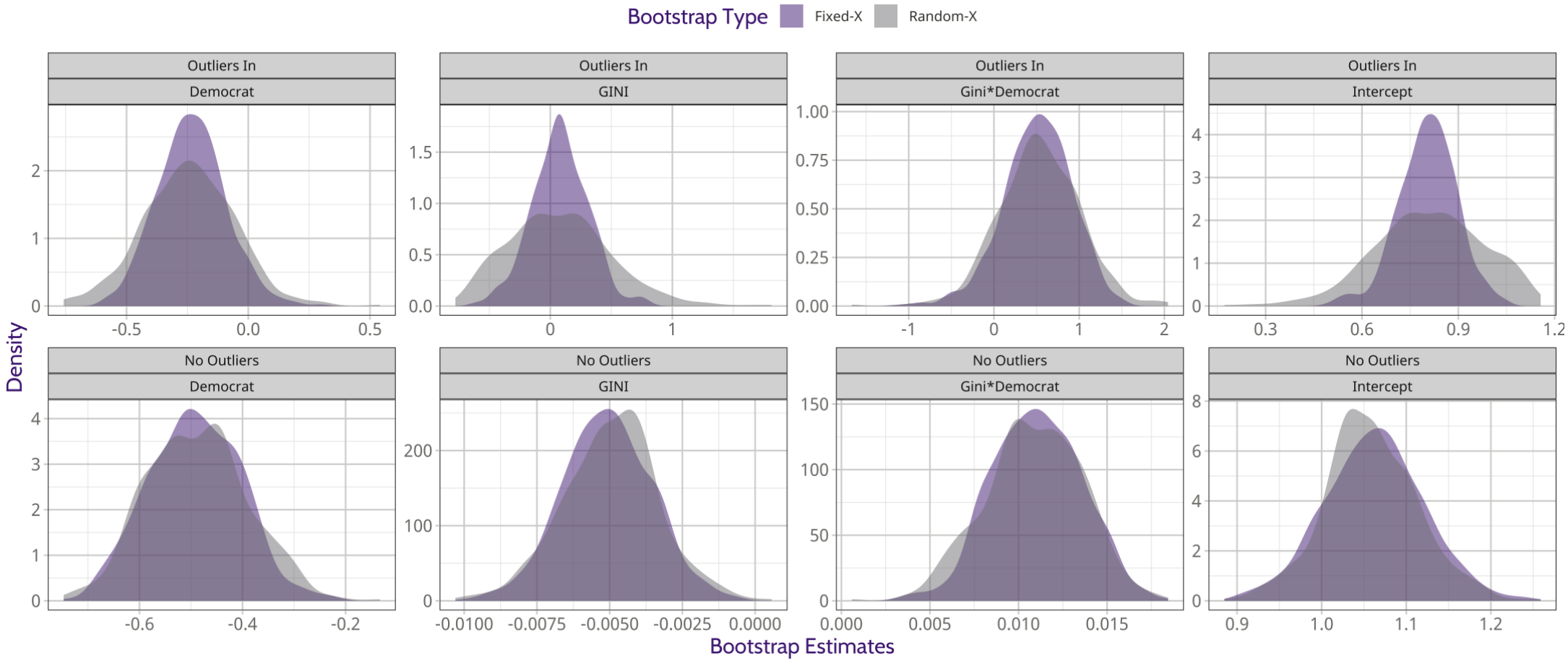
What if we Retained the Outliers?



Notes

Type notes here...

Bootstrap Disrtibution



Notes

Type notes here...

Jackknife-after-Bootstrap

The jackknife-after-bootstrap provides a diagnostic of the bootstrap by allowing us to examine what would happen to the density if particular cases were deleted

- Given that we know that there are outliers, this diagnostic is important
- The jackknife-after-bootstrap plot is produced using the `jack.after.boot` function in the `boot` package. Once again, the `index=2` argument asks for the results for the slope coefficient in the model

```
jack.after.boot(boot.reg1, index=2)
```

Notes

Type notes here...

Jackknife-after-Bootstrap (2)

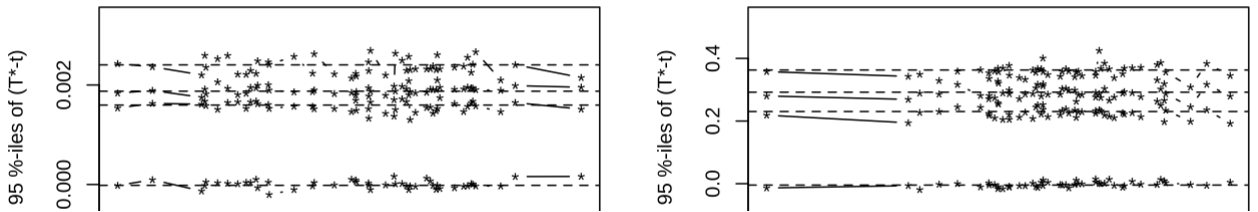
The `jack.after.boot` plot is constructed as follows:

- the horizontal axis represents the standardized jackknife value (the standardized value of the difference with that observation taken out)
- The vertical axis represents various quantiles of the bootstrap statistic
- Horizontal lines in the graph represent the bootstrap distribution at the various quantiles
- Case numbers are labeled at the bottom of the graph so that each observation can be identified

Notes

Type notes here...

Figure



Notes

Type notes here...

Fixed versus random

Why do the samples 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 un-modeled nonlinearity, heteroskedasticity or outliers) these attributes do *not* carry over to the bootstrap samples
- the effects of outliers was clear in the random-X case, but not with the fixed-X bootstrap

Notes

Type notes here...

When Might Nonparametric Bootstrapping Fail?

- Incomplete Data: since we are trying to estimate a population distribution from the sample data, we must assume that the missing data are not problematic. It is acceptable, however, to use bootstrapping if multiple imputation is used beforehand
- Dependent data: bootstrapping assumes independence when sampling with replacement. It should not be used if the data are dependent
- Outliers and influential cases: if obvious outliers are found, they should be removed or corrected before performing the bootstrap (especially for random-X). We do not want the simulations to depend crucially on particular observations

Notes

Type notes here...

Example from Class

We could bootstrap the ALSOS algorithm to get a sense of how variable the optimally scaled values are.

```
library(DAMisc)
library(progress)
load(file("https://quantoid.net/files/reg3/wvs.rda"))
wvs$class_num <- as.numeric(wvs$class)
wvs <- wvs %>%
  select(age, class_num, educ, income, sex, happy) %>%
  na.omit()
nreps <- 500
form <- as.formula(class_num ~ age + I(age^2) + educ + income + sex + happy)
```

Let's work through building the function together.

Notes

Type notes here...

Parametric Bootstrap

A parametric bootstrap re-samples from a known distribution to characterize uncertainty.

- This feels a lot like being Bayesian
  - In fact, in some cases you can even treat parametric bootstrapped values as draws from a posterior distribution.
- The way I almost always use this is to sample from the distribution of regression model coefficients.

$$\mathbf{B} \sim \mathcal{N}_k(\mathbf{b}, \mathbf{V}_b)$$

You can then use draws of $\mathbf{B}$ to make calculations.

Notes

Type notes here...

Example

Going back to the lecture on polynomial regression, what if we wanted to get a 95% confidence interval for the point at which the predicted prestige for women reached its lowest point?

```
data(Prestige, package="carData")
mod <- lm(prestige ~ log(income) + poly(women, 2, raw=TRUE) +
            poly(education, 3), data=Prestige)
b1 <- unname(coef(mod)[3])
b2 <- unname(coef(mod)[4])
minprest <- -b1/(2*b2)
minprest
```

```
## [1] 35.86337
```

Notes

Type notes here...

Both Types of Bootstrap

Non-parametric

```
boot.min <- function(data, inds){
  m <- update(mod, data=data[inds, ])
  b1 <- unname(coef(m)[3])
  b2 <- unname(coef(m)[4])
  -b1/(2*b2)
}
out1 <- boot(Prestige, boot.min, R=5000)
boot.ci(out1, type="perc")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 5000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = out1, type = "perc")
##
## Intervals :
## Level     Percentile
## 95%   (-1.74, 48.82 )
## Calculations and Intervals on Original Scale
```

Parametric

```
B <- MASS::mvrnorm(5000, coef(mod), vcov(mod))
out2 <- -B[,3]/(2*B[,4])
quantile(out2, c(.025, .975))
```

```
##      2.5%     97.5%
##  7.114885 47.818339
```

Notes

Type notes here...

Uses of the Parametric Bootstrap

I use the parametric bootstrap in the following settings.

1. Predicted response values from non-linear GLMs - both average case and average effect approaches.
    - Differences in predictions as well.
2. Importance or Relative Importance measures.
3. Simulations to understand the properties of some quantity.

Notes

Type notes here...

Assumptions Redux

Random-X

- Sample is representative of the population.
- Systematic component of the model is properly specified.

Fixed-X

- Random-X assumptions $+$
- Errors are iid.

Parametric

- Fixed-X assumptions $+$
- Random component of the model is properly specified (i.e., assumed error distribution is correct).

Notes

Type notes here...