MLE: Lab 4

Andy Ballard

February 3, 2017

Autocorrelation (most of this is cobbled together from a variety of sites I like)

What is it?

There are times, especially in time-series data, that the CLR assumption of

$$corr(\epsilon_t, \epsilon_{t-1}) = 0$$

is broken. This is known in econometrics as Serial Correlation or Autocorrelation. This means that

$$corr(\epsilon_t, \epsilon_{t-1}) \neq 0$$

and there is a pattern across the error terms. The error terms are then not independently distributed across the observations and are not strictly random.

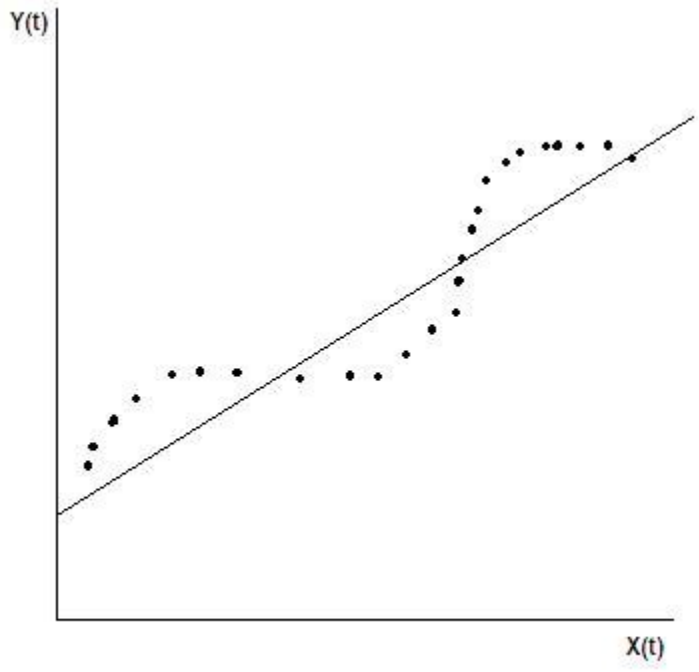


Figure 1: Positive Autocorrelation

When the error term is related to the previous error term, it can be written in an algebraic equation.

$$\epsilon_t = \rho\epsilon_{t-1} + u_t$$

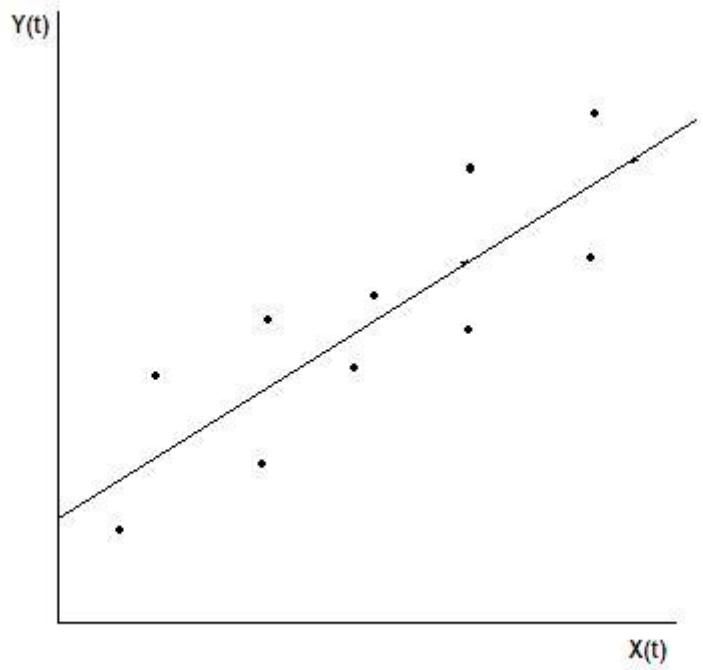


Figure 2: Negative Autocorrelation

where ρ is the autocorrelation coefficient between the two disturbance terms, and u is the disturbance term for the autocorrelation. This is known as an Autoregressive Process.

$$-1 < \rho = corr(\epsilon_t, \epsilon_{t-1}) < 1$$

The u is needed within the equation because although the error term is less random, it still has a slight random effect.

Examples

- Example: Consider stock market data. The price of a particular security or a stock market index at the close of successive days or during successive hours is likely to be serially correlated.
- Example: Consider the consumption of electricity during different hours of the day. Because the temperature patterns are similar between successive time periods, we can expect consumption patterns to be correlated between neighboring periods. If the model is not properly specified, this effect may show up as high correlation among errors from nearby periods.
- Systematic errors in measurement can also cause autocorrelation. For example, suppose a firm is updating its inventory in a given period. If there is a systematic error in the way it was measured, cumulative inventory stock will reflect accumulated measurement errors. This will show up as serial correlation.

Consequences

- Coefficients are still unbiased $E(\epsilon_t) = 0$, $cov(X_t, u_t) = 0$

- True variance of $\hat{\beta}$ is increased, by the presence of autocorrelations.
- Estimated variance of $\hat{\beta}$ is smaller due to autocorrelation (biased downward).
- A decrease in se($\hat{\beta}$) and an increase of the t-statistics; this results in the estimator looking more accurate than it actually is.
- R^2 becomes inflated.

Spatial Autocorrelation

Same issues as serial correlation.

- Example: The city of St. Paul has a spike of crime and so they hire additional police. The following year, they found that the crime rate decreased significantly. Amazingly, the city of Minneapolis, which had not adjusted its police force, finds that they have a increase in the crime rate over the same period.

Moran's statistic

Moran's I is a measure of spatial autocorrelation–how related the values of a variable are based on the locations where they were measured.

To calculate Moran's I, we will need to generate a matrix of inverse distance weights. In the matrix, entries for pairs of points that are close together are higher than for pairs of points that are far apart.

```
# from WDI lets get some data
wbVars <- c("BX.KLT.DINV.CD.WD","SP.POP.TOTL", "NY.GDP.DEFL.KD.ZG",
  "NY.GDP.PCAP.KD", "NY.GDP.MKTP.KD.ZG")
wbData <- WDI(country='all', indicator=wbVars,
  start=1988, end=2010, extra=T)
names(wbData)[4:8]=c('fdi', 'population', 'inflation', 'gdpCap', 'gdpGr')

# log everything
wbData$fdi <- log(wbData$fdi + abs(min(wbData$fdi, na.rm=T)) + 1)
wbData$population <- log(wbData$population + abs(min(wbData$population, na.rm=T)) + 1)
wbData$gdpCap <- log(wbData$gdpCap + abs(min(wbData$gdpCap, na.rm=T)) + 1)

# take a peak at your data
head(wbData)

# Lets throw out non country units
wbData$cname <- countrycode(wbData$iso2c, 'iso2c', 'country.name')
wbData <- wbData[!is.na(wbData$cname),]
head(wbData)
unique(wbData$cname)

# Load a helpful dataset
load(paste0(labPath, '/panel.rda'))

# Finding distances
capdist <- distmatrix(as.Date('2000-1-1'), type='capdist', tolerance=0.5, useGW=FALSE)
capdist[1:5, 1:5]
```

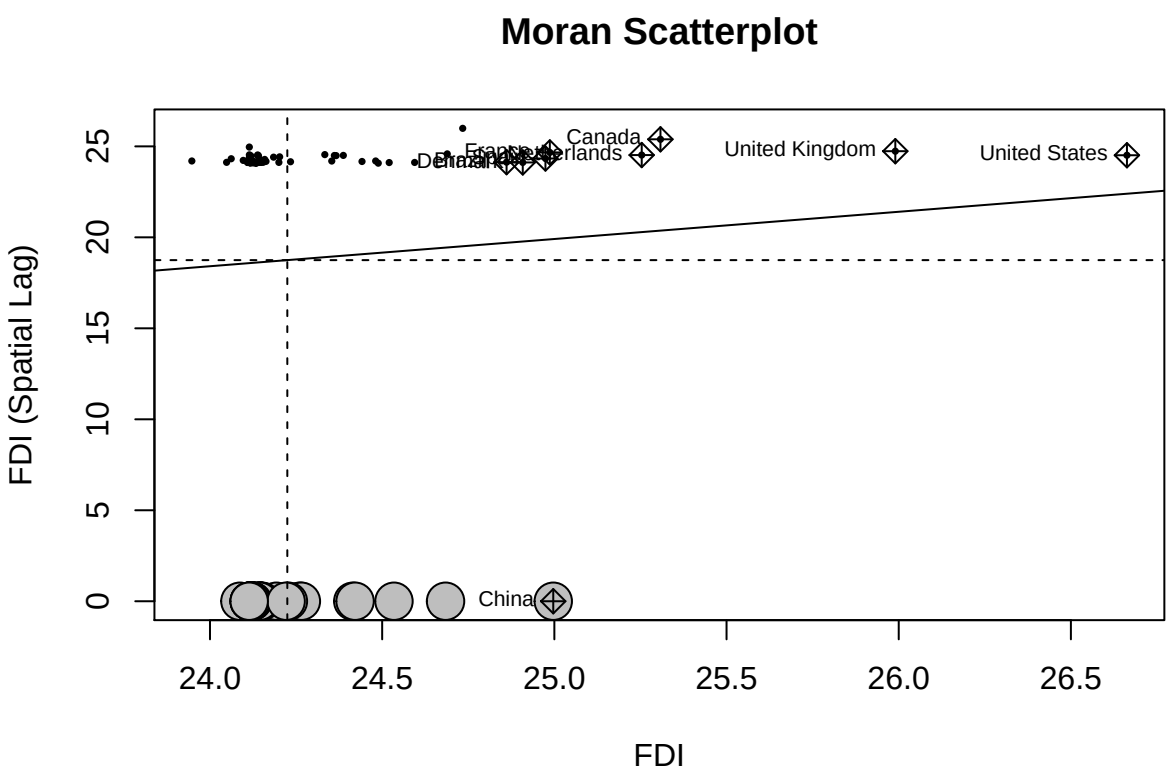
```
# lets see if we can calculate Moran's I for FDI
# lets focus on 2000 data
fdi <- wbData[which(wbData$year == 2000), c('cname', 'iso3c', 'fdi')]
fdi <- na.omit(fdi)

# Lets match countries with values in distance matrix
# First we need to get ccodes for the fdi dataset because that is what is used
# in our distance matrix
fdi$ccode <- panel$ccode[match(toupper(fdi$cname), panel$cname)]

# Some countries did not match, lets just drop 'em
sum(is.na(fdi$ccode))
fdi <- na.omit(fdi)

# Reorder rows of fdi dataframe
fdi <- fdi[order(fdi$ccode),]

# Now lets isolate the fdi and capdist matrix to only
# countries that exist in both objects
matches <- intersect(fdi$ccode, rownames(capdist))
fdi <- fdi[which(fdi$ccode %in% matches),]
capdist <- capdist[matches, matches]

# Like the example above we have to invert the distance matrix
invcapdist <- 1/capdist
diag(invcapdist) <- 0

# Now lets Moran's test
Moran.I(fdi$fdi, invcapdist)
```

Fun with spatial stats

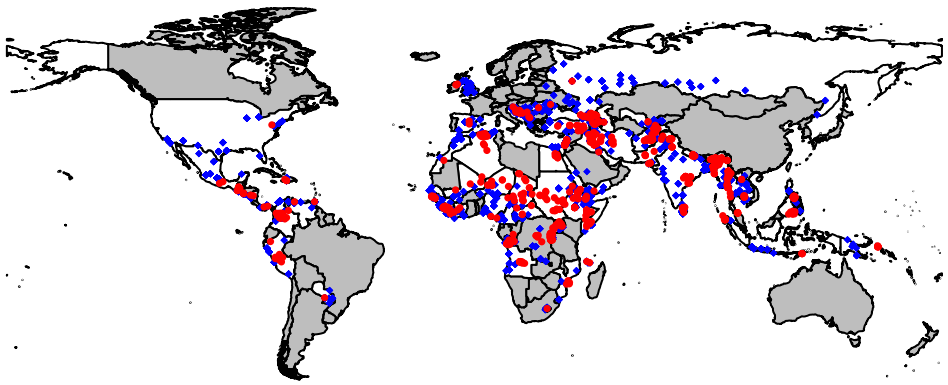
Some fancy stuff. Spatial statistics is a huge field. If you're interested in it follow the work of Roger Bivand.

```
# More fancy stuff
# Could also use spdep to make the Moran's plot
cshp00 <- cshp(date=as.Date('2000-1-1'), useGW=TRUE)

# Remove polygons from cshp00
cshp00 <- cshp00[which(cshp00$COWCODE %in% matches),]

# Centroid coordinates
coords <- coordinates(cshp00)

# Get list of neighbors
cshp00nb <- poly2nb(cshp00, queen=TRUE)
cshp00nbBin <- nb2listw(cshp00nb, style="W", zero.policy=TRUE)

# Plot centroid connections
plot(cshp00nbBin,coords=coords,pch=19, cex=0.1, col="gray")
```

```r
# Moran test again
moran.test(fdi$fdi, listw=cshp00nbBin, zero.policy=T)
geary.test(fdi$fdi, listw=cshp00nbBin, zero.policy=T)

# Moran plot
moran.plot(fdi$fdi, listw=cshp00nbBin, zero.policy=T, pch=16, col="black",cex=.5, quiet=F,
           labels=as.character(fdi$cname),xlab="FDI", ylab="FDI (Spatial Lag)", main="Moran Scatterplot
```


Moran Scatterplot
FDI (Spatial Lag)
FDI
0
5
10
15
20
25
24.0
24.5
25.0
25.5
26.0
26.5
Canada
United Kingdom
United States
China

Map Visualizations using cshapes



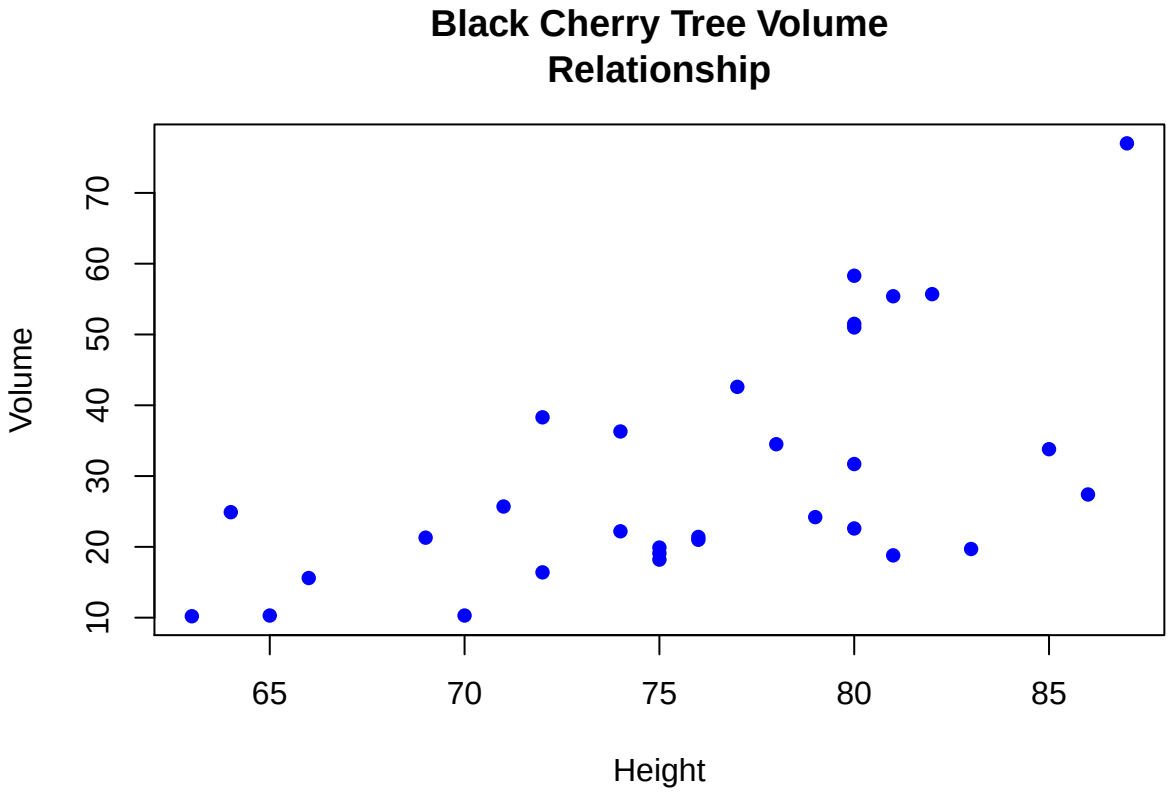
Bootstrapping

"The population is to the sample as the sample is to the bootstrap samples"

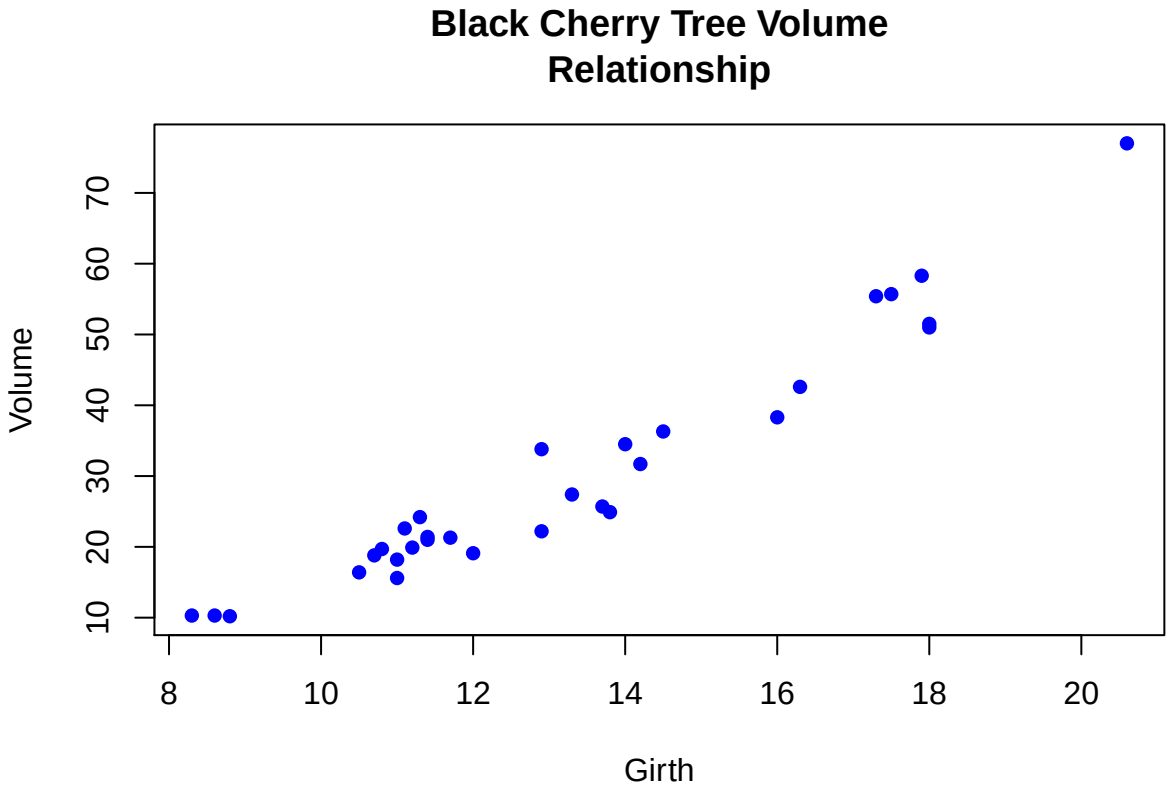
```
#We'll use a dataset native to R looking at black cherry trees

head(trees)
```

```
##   Girth Height Volume
## 1   8.3     70   10.3
## 2   8.6     65   10.3
## 3   8.8     63   10.2
## 4  10.5     72   16.4
## 5  10.7     81   18.8
## 6  10.8     83   19.7
```

```
plot(trees$Volume~trees$Height, main = 'Black Cherry Tree Volume
Relationship', xlab = 'Height', ylab = 'Volume', pch = 16, col =
'blue')
```




```
plot(trees$Volume~trees$Girth, main = 'Black Cherry Tree Volume
Relationship', xlab = 'Girth', ylab = 'Volume', pch = 16, col =
'blue')
```



Let's say we're interested in figuring out what affects a tree's volume. We'll write a function to calculate the quantities we're interested in, which are the model fit statistics (R^2) predicting volume based on:

- Height only
- Girth only
- Girth/height ratio
- Girth and height
- Girth, height, and girth/height ratio

This way, we can tell which model is truly the best fit.

```
volume_estimate <- function(data, indices){
 d <- data[indices, ]

 H_relationship <- lm(d$Volume~d$Height, data = d)
 H_r_sq <- summary(H_relationship)$r.square

 G_relationship <- lm(d$Volume~d$Girth, data = d)
 G_r_sq <- summary(G_relationship)$r.square

 G_H_ratio <- d$Girth / d$Height
 G_H_relationship <- lm(d$Volume~G_H_ratio, data = d)
 G_H_r_sq <- summary(G_H_relationship)$r.square

 combined_relationship <- lm(d$Volume~d$Height + d$Girth, data = d)
 combined_r_sq <- summary(combined_relationship)$r.square
```

```
 combined_2_relationship <- lm(d$Volume~d$Height +d$Girth + G_H_ratio, data = d)
 combined_2_r_sq <- summary(combined_2_relationship)$r.square

 relationships <- c(H_r_sq, G_r_sq, G_H_r_sq, combined_r_sq, combined_2_r_sq)
 return(relationships)
}
```

The `boot` function will take a sample from our data `trees` based on the `indices` input in the function (which the `boot` function automatically provides), and then runs our models and gives us the R^2 values for each model to check out which has the best fit.

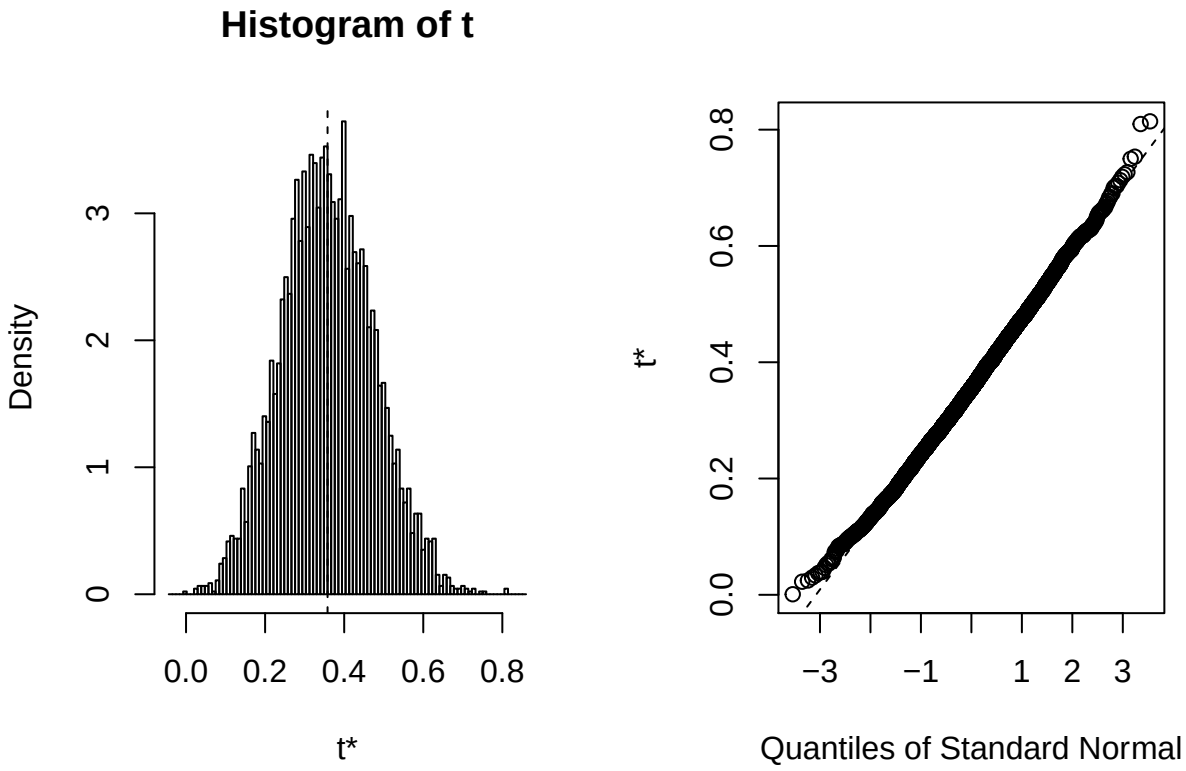
Now we can actually do the bootstrapping. We'll calculate all the model fit statistics for the models above on 5000 different samples from the full data.

```
results <- boot(data = trees, statistic = volume_estimate, R = 5000)
print(results)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = trees, statistic = volume_estimate, R = 5000)
##
##
## Bootstrap Statistics :
##      original        bias    std. error
## t1* 0.3579026  5.007886e-06  0.11624489
## t2* 0.9353199 -1.396852e-05  0.01796343
## t3* 0.7309204 -1.765554e-04  0.08272164
## t4* 0.9479500  3.136281e-03  0.01197627
## t5* 0.9732894  3.058722e-04  0.01033515
```

We had 5 total relationships that we tested, which are t1 (height only) through t5 (girth, height, and G/H). So if we wanted to look at a plot for the R^2 for the height-only model we would...

```
plot(results, index=1)
```



And if we want the 95% confidence interval for the R^2 of the height-only model, we would...

```
confidence_interval_H <- boot.ci(results, index = 1, conf = 0.95, type = 'bca')
print(confidence_interval_H)
```

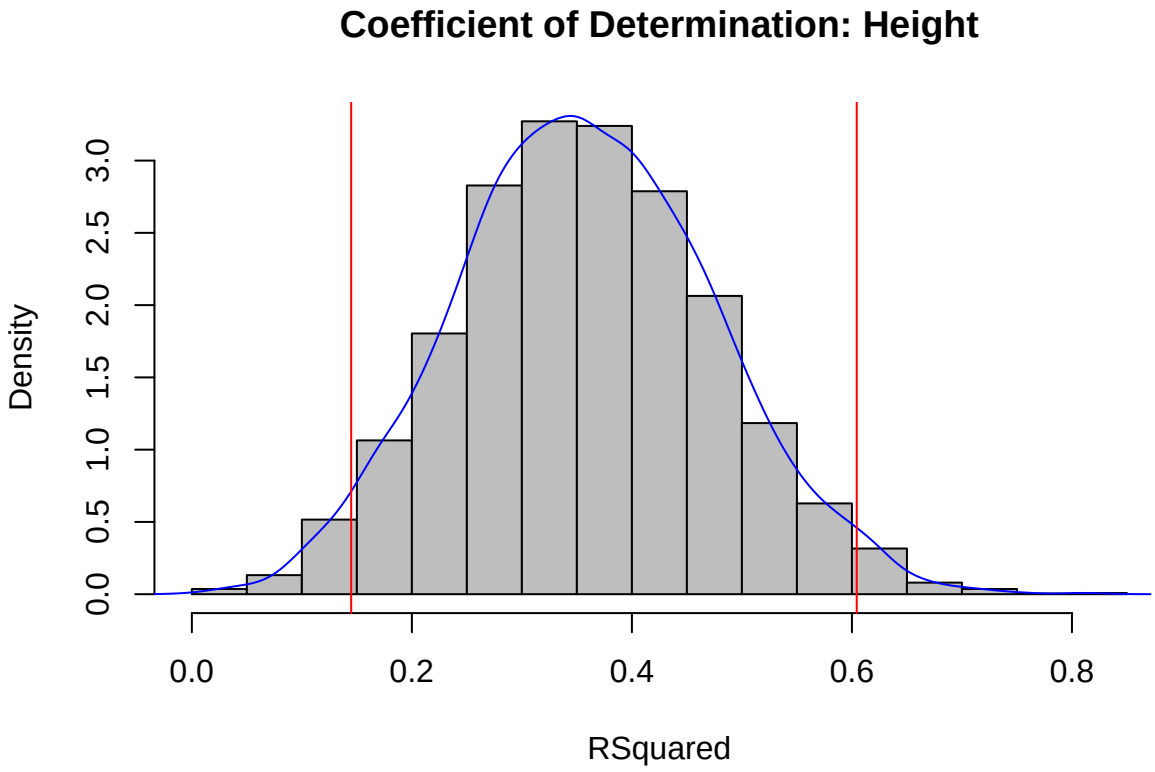
```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 5000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results, conf = 0.95, type = "bca", index = 1)
##
## Intervals :
## Level       BCa
## 95%   ( 0.1448,  0.6044 )
## Calculations and Intervals on Original Scale
```

```
ci_H <- confidence_interval_H$bca[ , c(4, 5)]
print(ci_H)
```

```
##
## 0.1448449 0.6043625
```

Now we can add our confidence interval bounds to the plots

```
hist(results$t[,1], main = 'Coefficient of Determination: Height', xlab = 'RSquared',
col = 'grey', prob = T)
lines(density(results$t[,1]), col = 'blue')
abline(v = ci_H, col = 'red')
```



Now the example we've used here is simple, and these data do generally fit the usual distributional assumptions and asymptotic results are valid and accurate. How can we tell this from the bootstrapped results?

Why might bootstrapping be useful?

What does the "bias" column in the bootstrapping results mean?

It's sort of weird to think of the R^2 as something with variance, isn't it? What does it actually mean?