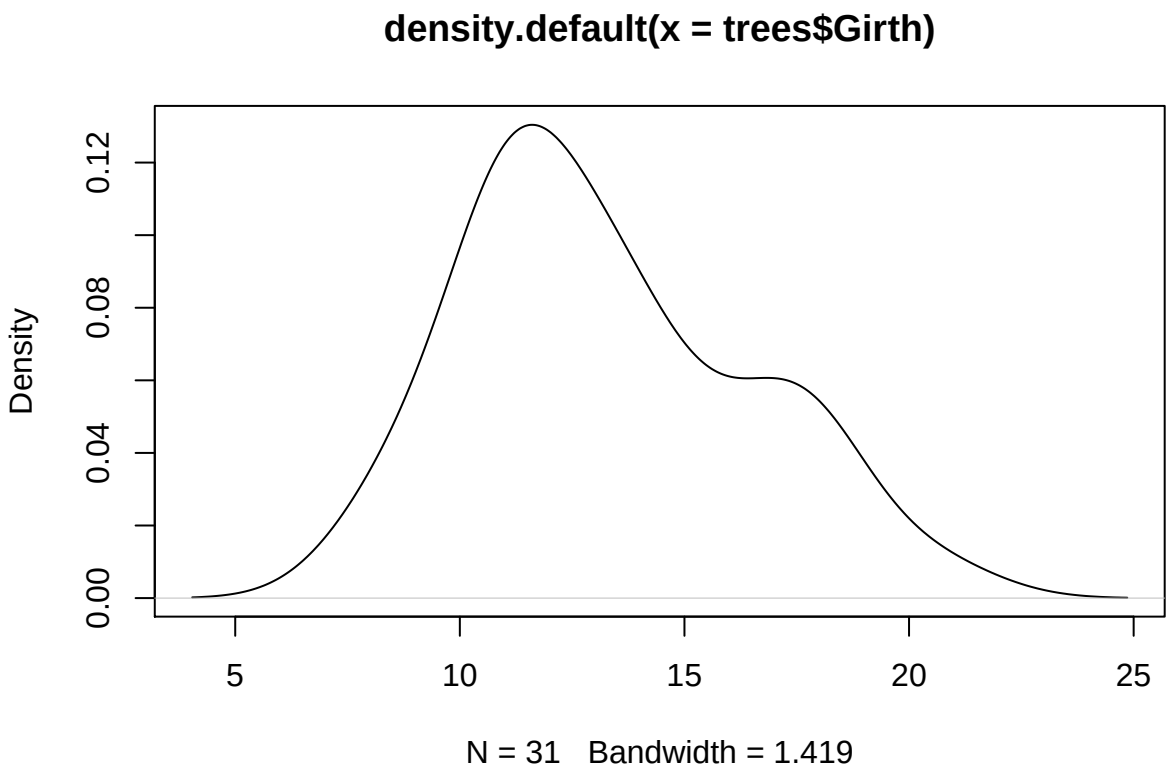
Simulating Correlated Random Variables

Simulation

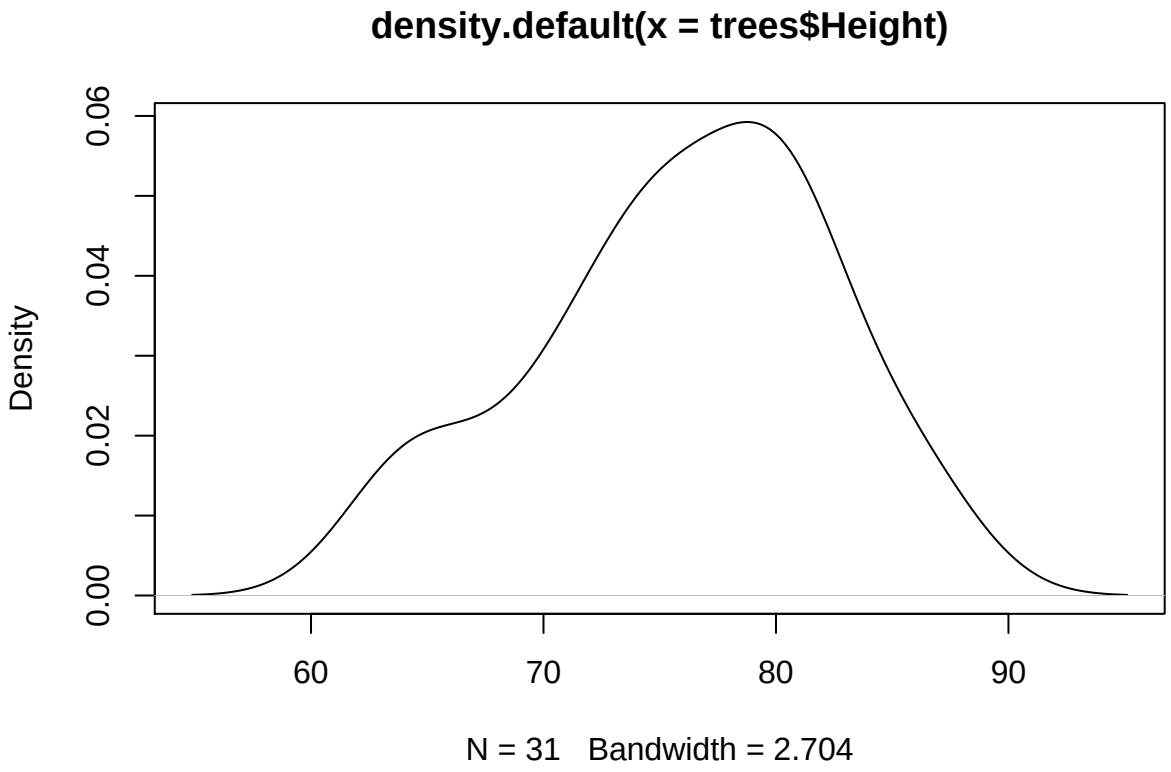
I'm going to describe two approaches in R for simulating correlated random variables. I'll use the dataset `trees`, which has measurements of girth (also called diameter at base height (DBH)), height, and volume. Take a look at the data.

Let's say that I'm interested in using this dataset to simulate trees with similar correlations between girth, height, and volume. **Why simulation?** Our first step might be to look at the histograms and density plots for each variable.

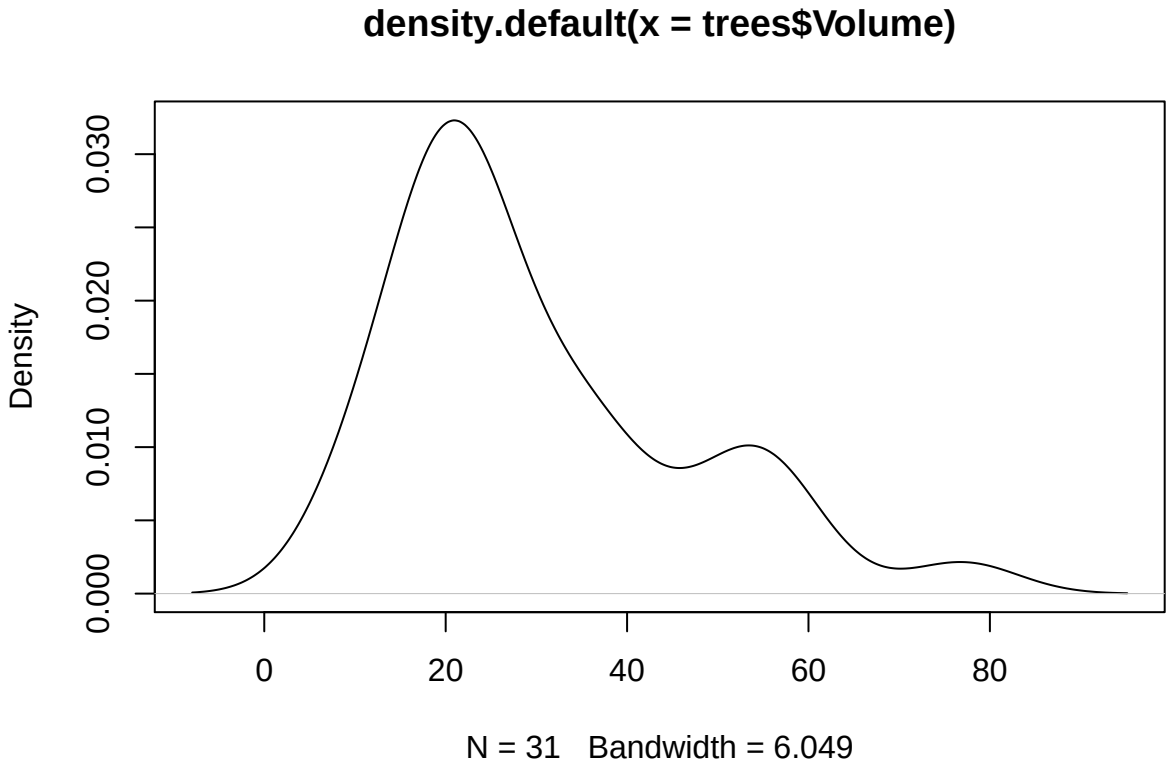
```
plot(density(trees$Girth))
```




```
plot(density(trees$Height))
```




```
plot(density(trees$Volume))
```



The data are a bit skewed, sure, but they seem roughly normal. Maybe we can estimate the mean and standard deviation, and use those to simulate the variables? Let's try it.

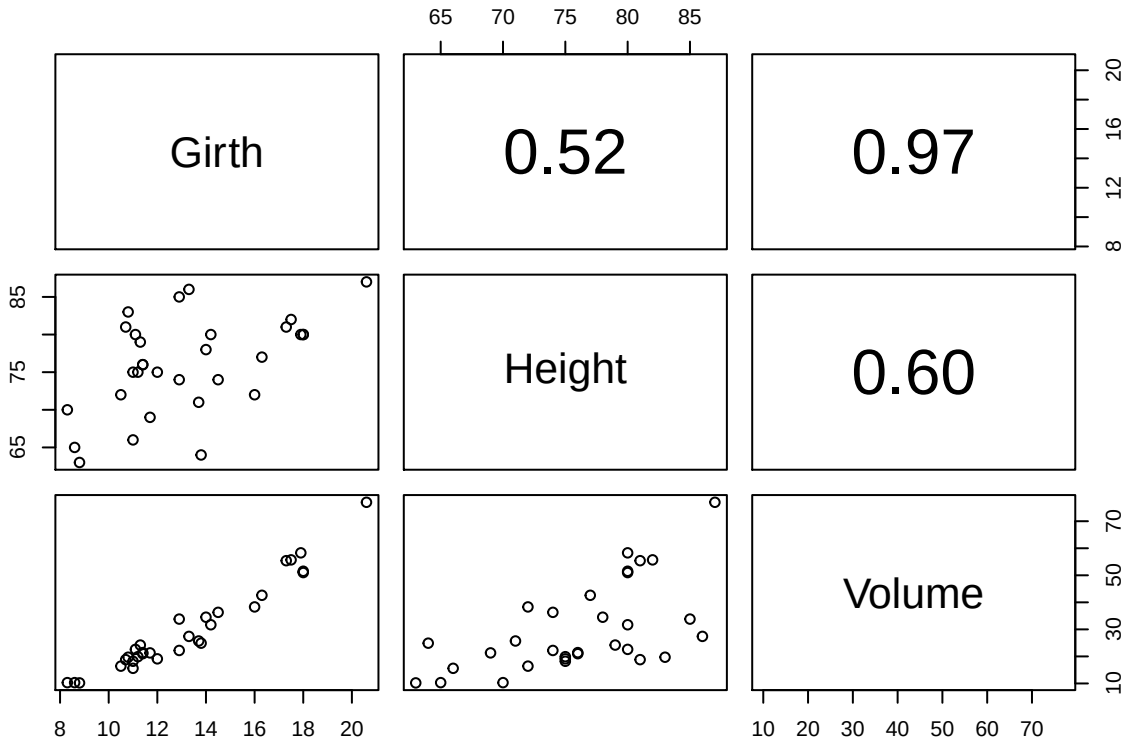
```
normal.par<-function(x) c(mean(x),sd(x))

nVars=3;nSim=100;
dt<-matrix(nrow=nSim,ncol=nVars,dimnames=list(c(),names(trees)))
for(i in 1:nVars){
  dt[,i]<-rnorm(nSim,mean=normal.par(trees[,i])[1],normal.par(trees[,i])[2])
}
```

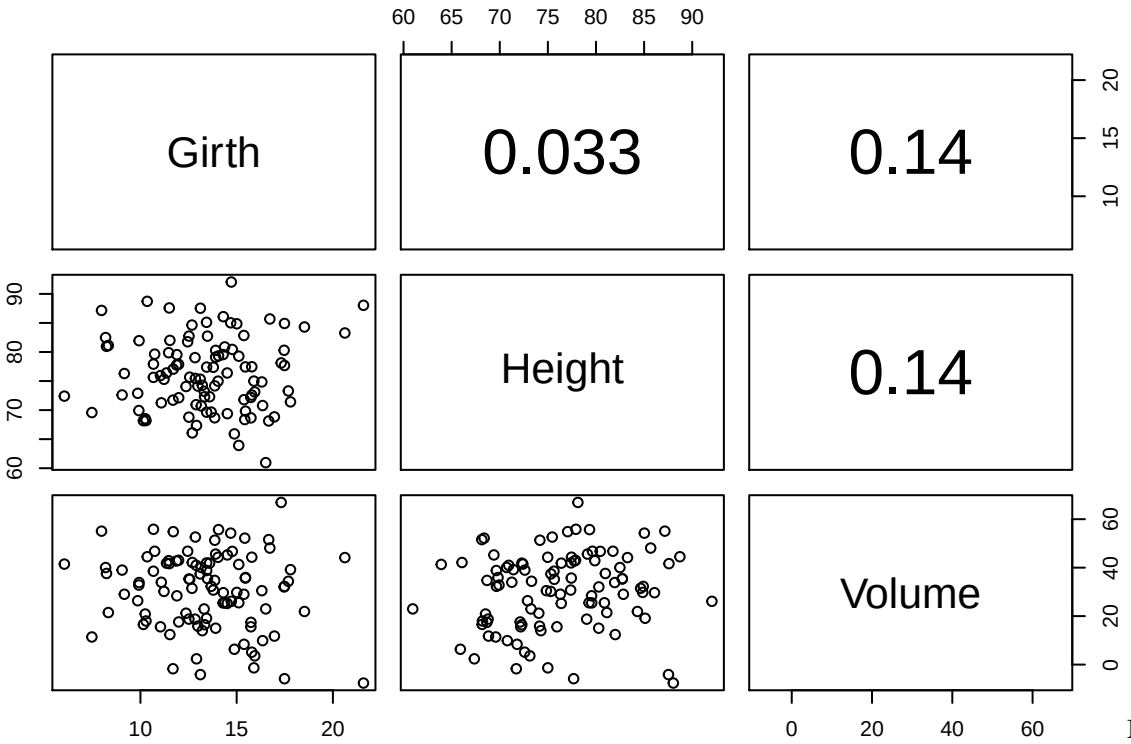
How well does this work? The function `panel.cor` lets us plot a pairs plot with correlation values added.

```
panel.cor <- function(x, y, digits=2, prefix="", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  txt <- paste(prefix, txt, sep="")
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = 3)
}

pairs(trees,upper.panel=panel.cor)
```




```
pairs(dt,upper.panel=panel.cor)
```



Not so well.

The simulation does not take into account correlation between variables, so generates values that are independently, normally distributed. The normal distribution is also unbounded, which is why there are negative values. But the key point here is that if variables are correlated, that correlation should be accounted for in a simulation. But how do we do that?

Cholesky Decomposition

nVars=2;n.gen=1000; random.normal = matrix(rnorm(nVars*n.gen), nrow=nVars, ncol=n.gen) plot(density(trees[,1]));plot(density(trees[,2]));plot(density(trees[,3]))

Perform the test

shapiro.test(v1); shapiro.test(v2)

Plot using a qqplot

qqnorm(trees[,1]);qqline(trees[,1], col = 2) qqnorm(trees[,2]);qqline(trees[,2], col = 2) qqnorm(trees[,3]);qqline(trees[,3], col = 2)

x <- rgamma(1e5, 2, .2) plot(density(x))

normalize the gamma so it's between 0 & 1

.0001 added because having exactly 1 causes fail

xt <- trees[,3] #/ (max(trees[,3]) + .0001)

fit a beta distribution to xt

library(MASS) fit.lnorm <- fitdistr(xt, "log-normal") fit.norm <- fitdistr(xt, "normal")

x.lnorm <- rlnorm(1e5,fit.lnorm$estimate[[1]], $fit.lnorm$estimate[[2]]) x.norm <- rnorm(1e5,fit.norm$estimate[[1]], $fit.norm$estim

plot the pdfs on top of each other

plot(density(xt)) lines(density(x.lnorm), col="red") lines(density(x.norm), col="blue")

cor(trees[,1:2])

nVars=2;n.gen=1000; random.normal = matrix(rnorm(nVars*n.gen), nrow=nVars, ncol=n.gen) treesCholesky<-trees treesCholesky[,3]<-log(treesCholesky[,3]) ## Cholesky Decomposition Method # calculate the correlation for each site correlationMatrix=cor(treesCholesky[,2:3],use="complete.obs") # decompose the correlation matrix using Cholesky decomposition decomposedMatrix=t(chol(correlationMatrix)) # generate correlation structure by multiplying decomposed matrix # by values drawn from random normal distribution simulatedValues<-t(decomposedMatrix %% random.normal)

for(j in 1:nVars){ simulatedValues[,j]<-(simulatedValues[,j]*(sd(treesCholesky[,j+1]))+(mean(treesCholesky[,j+1]))) }

exponential(precipitation values)

simulatedValues[,2]=exp(simulatedValues[,2])

pairs(simulatedValues) plot(simulatedValues[,1],simulatedValues[,2]) points(trees[,2],trees[,3],col="red")

Copula method

reps=1000 #Defines the temporal correlation between the two parameters. library(MASS) Z <- mvrnorm(n=reps,mu=c(0,0), matrix(data=cor(trees[,2:3]),nrow=2,ncol=2)) #Generates standard multivariate normal data with correlation structure defined by rho. U = pnorm(Z) #Apply the Normal CDF function to Z to obtain data that is uniform on the interval [0,1], but still correlated. x <- qnorm(U[,1],mean=mean(trees[,2]),sd=sd(trees[,2])) y <- qlnorm(U[,2],fit.lnorm$estimate[[1]], $fit.lnorm$estimate[[2]])

xvar <- x yvar <- y xy<-(data.frame(x,y)) zvar <- as.factor(c(rep(1, reps), rep(2, reps))) xy <- data.frame(xy, zvar)

panel.cor <- function(x, y, digits=2, prefix="", cex.cor, ...) { usr <- par("usr"); on.exit(par(usr)) par(usr = c(0, 1, 0, 1)) r <- abs(cor(x, y)) txt <- format(c(r, 0.123456789), digits=digits)[1] txt <- paste(prefix, txt, sep="") if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt) text(0.5, 0.5, txt, cex = 3) }

plot(x,y) points(trees[,2],trees[,3],col="red") points(simulatedValues[,1],simulatedValues[,2],col="blue")

pairs(trees[,2:3],upper.panel=panel.cor); pairs(matrix(c(x,y),reps,2),upper.panel=panel.cor) pairs(simulatedValues,upper.panel=

```
twonorms<-function(val=val){
reps=10000
#Defines the  sequence for stochastic trials.
a= 0
b = 1
#a and b are hyperparameters of the gamma distribution
#that define both the expected value and variance.
alpha = 0
```

```
beta =  1
#alpha and beta are hyperparameters of the beta distribution that define both the expected value and
#variance.
rho = val
#Defines the temporal correlation between the two parameters.
library(MASS)
Z <- mvrnorm(n=reps,mu=c(0,0), matrix(data=c(1,rho,rho,1),nrow=2,ncol=2))
#Generates standard multivariate normal data with correlation structure defined by rho.
U = pnorm(Z)
#Apply the Normal CDF function to Z to obtain data that is uniform on the interval [0,1], but still cor
x <- qnorm(U[,1],a,b) #x is gamma distributed
y <- qnorm(U[,2],alpha,beta) #y is beta distributed

xvar <- x
yvar <- y
xy<-(data.frame(x,y))
zvar <- as.factor(c(rep(1, 1000), rep(2, 1000)))
xy <- data.frame(xy, zvar)

#scatterplot of x and y variables
scatter <- ggplot(xy,aes(x, y)) +
  geom_point() +
  scale_color_manual(values = c("", "")) +
  theme(legend.position=c(1,1),legend.justification=c(1,1))

# Scatter plot of x and y variables and color by groups
sp2 <- ggplot(xy,aes(x, y)) + geom_point()

library(ggExtra)

# Marginal density plot
ggMarginal(sp2 + theme_gray())
}
```