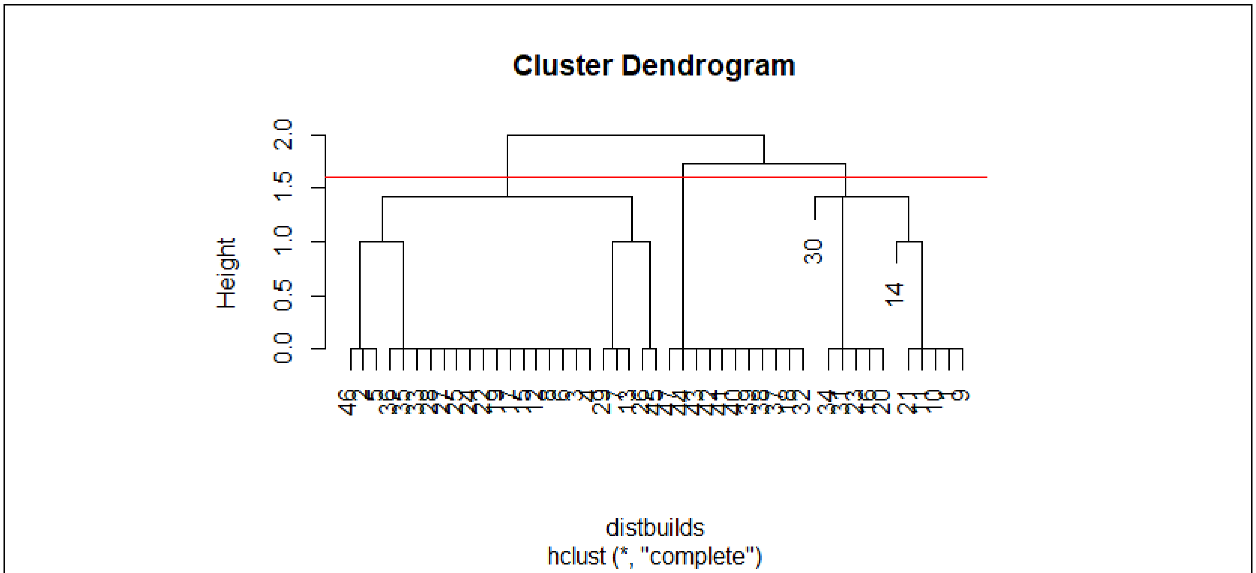
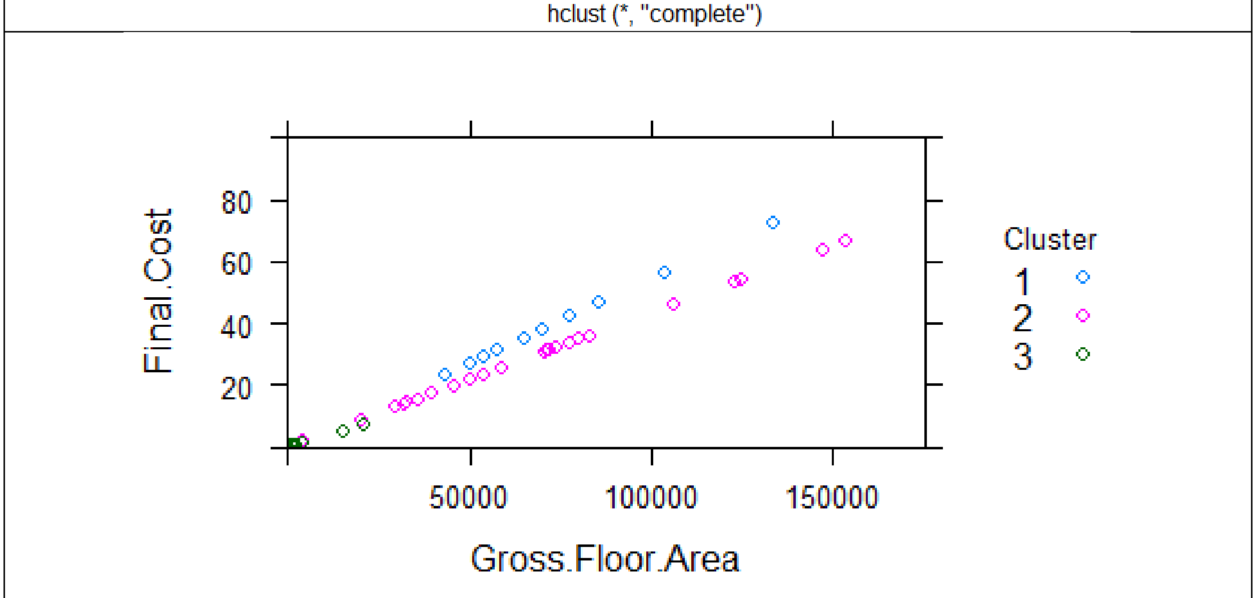
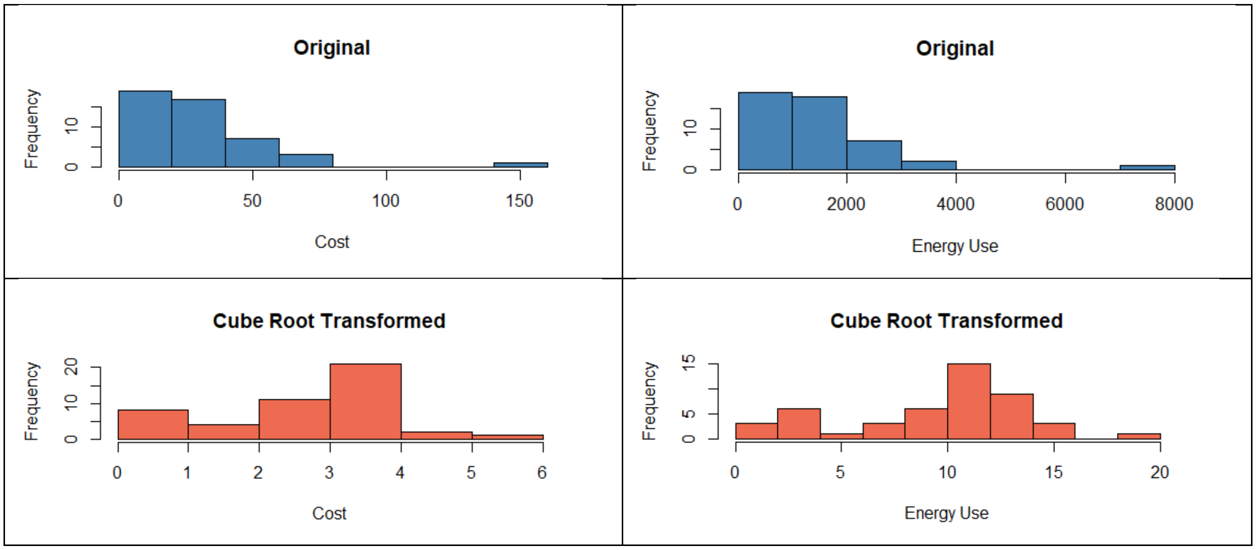
Clustering to determine classes:

Hierarchal Clustering with a complete linkage on the original building data resulted in 3 classes. These classes were determined to represent their sustainability score: low, average, or high. Low sustainability represented the buildings that had higher costs and were multiplied by a factor of 1.25 to represent their higher energy use on campus. High sustainability buildings had lower costs and were multiplied by a factor of 0.75 to represent this lower usage. The average building had no change.





Regression to model cost and energy of future buildings:



Transformation performed to ensure positive values as well as deal with skewedness (model overcompensation for larger values).

Let the variables $\mathrm{x_G}$ = gross floor area, $\mathrm{x_D}$ = description(dorm or university), $\mathrm{x_F}$ = provides food, $\mathrm{x_S}$ = stem or high tech use, $\mathrm{x_H}$ = heat, $\mathrm{x_A}$ = AC

Other than x_G, the variables are qualitative, represented with 1 or 0.

$$\mathrm{Cost} = (0.8330704 + 0.0000144\mathrm{x_G} + 0.2238008\mathrm{x_D} - 0.2465293\mathrm{x_F} + 0.1069584\mathrm{x_S} + 1.1789412\mathrm{x_H} + 0.0165981\mathrm{x_A})^3$$

$$\mathrm{Usage} = (2.9866395 + 0.0000517\mathrm{x_G} + 0.8023478\mathrm{x_D} - 0.8838317\mathrm{x_F} + 0.3834563\mathrm{x_S} + 4.2266201\mathrm{x_H} + 0.0595057\mathrm{x_A})^3$$

Adjusted R squared values for both models are .94

Approximately 94% of the observed variation can be explained by the model's inputs.

(how much variation of a dependent variable is explained by the independent variables)

Code:

```
library(lattice, lib.loc = "C:/Program Files/R/R-4.0.3/library")
##############################################

builds <- read.table(file='buildspecs.txt',header=TRUE, sep='\t')

buildsMX <- as.matrix(builds[,3:7])
distbuilds <- dist(buildsMX, method="euclidean")

clust <- hclust(distbuilds,method="complete")
plot(clust)
abline(h= 1.6,col = 'red')

clust3 <- cutree(clust, k=3)

labels<-cbind(builds[,1:7],as.factor(clust3))
colnames(labels)[8] <- "Cluster"

sum1 <- 0
for(i in 1:47)
{if(labels$Cluster[i]=="1")
  sum1 <- sum1 + 1}

sum2 <- 0
for(i in 1:47)
{if(labels$Cluster[i]=="2")
  sum2 <- sum2 + 1}

sum3 <- 0
for(i in 1:47)
{if(labels$Cluster[i]=="3")
  sum3 <- sum3 + 1}

sum1 # 12
sum2 # 24
sum3 # 11

for(i in 1:47){
```

```
  if(labels$Cluster[i]=="1"){labels[i,9]<-cbind(1.5)}
  if(labels$Cluster[i]=="2"){labels[i,9]<-cbind(1)}
  if(labels$Cluster[i]=="3"){labels[i,9]<-cbind(.5)}
}

colnames(labels)[9] <- "MulFac"

#REGRESSION

averages <- read.table(file='buildspecsAVGED.txt',header=TRUE, sep='\t')

costModel <- lm(Final.Cost ~ Gross.Floor.Area + DORM + FOOD + STEM + HEAT + AC, data = averages)
summary(costModel)
round(costModel$coefficients, digits=6)

energyModel <- lm(Final.Use ~ Gross.Floor.Area + DORM + FOOD + STEM + HEAT + AC, data = averages)
summary(energyModel)
round(energyModel$coefficients, digits=6)

#######################################
logCost <- log(averages$Final.Cost)
logEnergy <- log(averages$Final.Use)

logA<-cbind(averages, logCost, logEnergy)

costModelLog <- lm(logCost ~ Gross.Floor.Area + DORM + FOOD + STEM + HEAT + AC, data = logA)
summary(costModelLog)
round(costModelLog$coefficients, digits=7)

energyModelLog <- lm(logEnergy ~ Gross.Floor.Area + DORM + FOOD + STEM + HEAT + AC, data = logA)
summary(energyModelLog)
round(energyModelLog$coefficients, digits=7)

costO.res = resid(costModel)
plot(fitted(costModel), costO.res) #somewhat pos skew, so ln can be used
abline(0,0)

costL.res = resid(costModelLog)
```

```
plot(fitted(costModelLog), costL.res)
abline(0,0)

useO.res = resid(energyModel)
plot(fitted(energyModel), useO.res) #somewhat pos skew, so ln can be used

useL.res = resid(energyModelLog)
plot(fitted(energyModelLog), useL.res)
abline(0,0)

######################################
sqrtCost <- sqrt(averages$Final.Cost)
sqrtEnergy <- sqrt(averages$Final.Use)

sqrtA<-cbind(averages, sqrtCost, sqrtEnergy)

costModelSqRt <- lm(sqrtCost ~ Gross.Floor.Area + DORM + FOOD + STEM + HEAT + AC, data = sqrtA)
summary(costModelSqRt)
round(costModelSqRt$coefficients, digits=7)

energyModelSqRt  <- lm(sqrtEnergy ~ Gross.Floor.Area + DORM + FOOD + STEM + HEAT + AC, data =
sqrtA)
summary(energyModelSqRt)
round(energyModelSqRt$coefficients, digits=7)

costS.res = resid(costModelSqRt)
plot(fitted(costModelSqRt), costS.res)
abline(0,0)

useS.res = resid(energyModelSqRt)
plot(fitted(energyModelSqRt), useS.res)
abline(0,0)

################

hist(logA$logCost)
hist(sqrtA$sqrtCost)  ####best normalization of the 3
```

```
hist(sqrtA$sqrtEnergy)
################################################
#ulitmately best normalization

hist(averages$Final.Cost, col='steelblue', main='Original', xlab="Cost")
hist(averages$Final.Use, col='steelblue', main='Original', xlab="Energy Use")

hist(cubeCost, col='coral2', main='Cube Root Transformed', xlab="Cost")

hist(cubeEnergy, col='coral2', main='Cube Root Transformed', xlab="Energy Use")
#######################################

cubeCost <- averages$Final.Cost^(1/3)
cubeEnergy <- averages$Final.Use^(1/3)

cubeA<-cbind(averages, cubeCost, cubeEnergy)

costModelCube <- lm(cubeCost ~ Gross.Floor.Area + DORM + FOOD + STEM + HEAT + AC, data = cubeA)
summary(costModelCube)
round(costModelCube$coefficients, digits=7)

energyModelCube  <- lm(cubeEnergy ~ Gross.Floor.Area + DORM + FOOD + STEM + HEAT + AC, data =
cubeA)
summary(energyModelCube)
round(energyModelCube$coefficients, digits=7)

costC.res = resid(costModelCube)
plot(fitted(costModelCube), costC.res)
abline(0,0)

useC.res = resid(energyModelCube)
plot(fitted(energyModelCube), useC.res)
abline(0,0)

#################
final <- read.table(file='buildspecsAVGED.txt',header=TRUE, sep='\t')
```

```
xyplot(Final.Cost ~ Gross.Floor.Area, data=final, groups = Cluster, auto.key=list(space='right',
title = "Cluster",cex.title=.8),
       xlim=c(0,175000), ylim=c(0,100))

xyplot(Final.Cost ~ Gross.Floor.Area|Cluster, data=final, groups = Cluster,
auto.key=list(space='right', title = "Cluster",cex.title=.8),
       xlim=c(0,175000), ylim=c(0,100))
```