Darling Judah Hsu

An Investigation into ANOVA Assumptions and its Application to Heart Disease

Investigation into ANOVA Assumptions

Size Imbalances Between Samples

When it comes to huge imbalances in sample sizes, regarding anova, we've found that huge imbalances in sample sizes don't seem to be that much of an issue. In this study, I took 9 samples, each of 100,000 data points and then took one sample of only 100 data points. Samples were taken from r using the "rnorm" function, assuming normality and equal variance of 1 for all samples. The 9 samples of 100,000 data points each had a mean of 0 while the one sample of 100 data points had a mean of 0.5. From there, I ran the one-way ANOVA test, counted whether the p-value was significant enough to "reject the null," and repeated this process, in r, 100 times. In this case, the power was 0.94, meaning I correctly rejected the null 94 times. To test again, I repeated the process except that in our next test, the smaller sample had a mean of 1. In this case, I obtained a power of 1, meaning the one-way ANOVA correctly rejected the null hypothesis every time. So, huge imbalances in sample sizes don't seem to be that large of a problem, especially if the "different sample" is at least 0.5 standard deviations away from the mean of the other samples. For good measure, I also tested to see what would happen if the larger sample had a different mean and of course, I obtained a power of 1.

The only issue, I found, is when the smaller sample size is extremely small. Regardless of imbalance, for example, when I obtained 9 other samples of 100,000 data points or only 2 others

of 100 data points, the power decreases by a large margin. For example, when testing 1 normal sample of mean 1, variance 1, and size 10 against 9 other normal samples of size 100,000, mean 0, and variance 1, I obtained a power of 0.55- that is, only 55 out of 100 of the sampling distributions were correctly rejected. Testing 1 normal sample of 10 data points, mean 1, and variance 1, against 2 other normal sample sizes of 100, mean 0, and variance 1, I still obtained powers ranging from 0.71 - 0.86.

Conclusion: although huge power imbalances didn't seem to have as much of an effect when testing sufficiently large sample sizes, we see that when the "odd sample out" isn't sufficiently large, size imbalances make a major difference (causing power decrease from about 0.82 to 0.55 from huge size imbalances to smaller ones). However, we also find that when the "odd sample out" isn't sufficiently large, there's also an inherent loss of power from about 1 (when everything is equal except means) to 0.82 when one sample is different and not sufficiently large. It's also important to note that this is on the assumption of normality and equal variance.

Equal Variance Assumption Violated

When testing a violation of the equal variance assumption in this study, I did so under the assumption of normality and equal sample sizes. This study was conducted using the "rnorm" function in r, sampling from 3 normally distributed samples, each of sample size 100. As will be the case for every subsequent test in this part of the project, I once again found 100 sampling distributions and calculated the power as a metric for how well the one-way ANOVA test works (basic code outline for testing size imbalances, as well as unequal variance, can be found at the end of part 1).

First of all, I wanted to confirm that equal, but larger, variance of each sample would result in about the same power as a test run on samples with equal, but smaller, variance. So, with one sample mean being 0.5 units (not standard deviations) away from other means, an equal variance of 1 for each sample, and repeating the procedure multiple times, I found that the anova test resulted in powers ranging from 0.95-0.99. I then continued the procedure for equal variances of 4 and one sample meant being 0.5 units away from the others. Surprisingly (and unsurprisingly at the same time), I found that the powers now ranged from 0.38-0.46. This difference in power, I suspect, is likely because a mean 0.5 standard deviations away from the others has a larger effect, and is thus easier to detect, when variances are smaller. So, my first finding is that when variance within each sample is inherently large, differences in the means of the samples become harder to detect. When I changed the test sample (the one with an unequal location) to be 1 unit away from other samples, with equal variances of 4, I once again obtained powers ranging from 0.95-1. So, it's important to note that when testing equal variances, location differences should be scaled by standard deviation & variance.

Next, I found 2 primary situations to consider: when the test sample had a smaller variance than the other 2 and when the test sample had a larger variance than the other 2. My initial hypothesis was that power would be greater when the test sample had a smaller variance than the others, as opposed to the test sample having the greater variance. So, using the same procedure as earlier, I tested one sample (of mean 1 and standard deviation 9) against 2 other samples (of mean 0 and standard deviation 4). As it so happened, I obtained powers ranging from 0.73 to 0.89. Next, I tested one sample of mean 1 and standard deviation 1 against 2 other samples of mean 0 and standard deviation 4. As per my expectations, I obtained powers ranging from 0.99 to 1.

However I then wanted to test whether the effect of the "test sample having less variance than the other samples" would be more pronounced (that is, the power would be greater) as imbalances in variance became more pronounced. So, I tested a sample of mean 1 and standard deviation of 1 against 2 other samples of mean 0 and standard deviations of 9; to my surprise, I got less powers ranging from 0.81 to 0.93. From this, I then decided to test greater imbalances of variance when the test sample had greater variance and obtained seemingly better powers ranging from 0.91 to 0.96. Then, I tested unequal variances in general, where variances were either 20 or 1; then I obtained powers ranging from around 0.48 to 0.55 when the test sample had smaller variance and powers ranging from 0.68 to 0.73 when the test sample had larger variance.

Conclusion: unequal variances tend to lead to a loss of power. In general, there didn't seem to be a specific trend associated with when the test sample had smaller variance or larger variance. However, it was evident that greater imbalance between variances of the samples was associated with greater loss of power.

Another thing to note, however, is that there seems to be an interaction between unequal variance and imbalance sample size. For example, when we change the test sample (with mean 1) to have sample size 1000 (with mean 1 and variance 1), while the other two samples have sample size 100 (with means 0 and variance 1), we get power ranging from 0.96 to 0.99. Also, when we set the sample sizes equal but set the test sample to variance 4, we nonetheless get power essentially staying at 1. However, when we set the test sample size to 1000 with variance 4, the power falls to range from 0.42 to 0.58. Likewise, when we keep the test sample at sample size 100, variance 4, and mean 1, while setting the other 2 samples to size 1000 and variance 1, we find that the power ranges from 0.23 to 0.28. When we set the test sample to have size 1000, mean 1, and variance 1 while the other samples are set to size 100, mean 0, and variance 4, the power is a lot higher -but still lower than usual- ranging from 0.86 to 0.89.

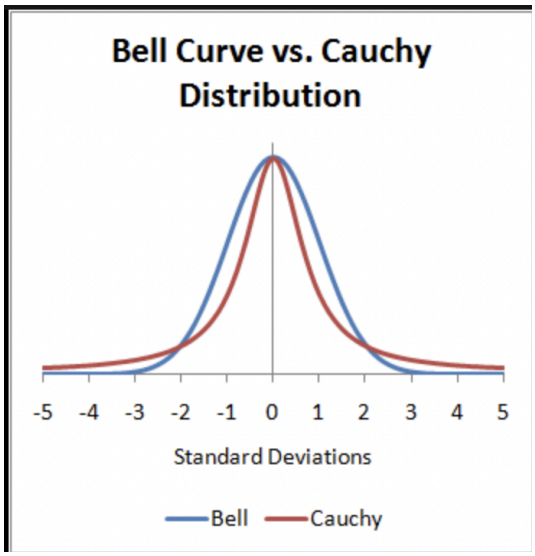
In this way, it's also important to note that though size imbalance might not have a grand effect on power; mixed with unequal variance, it could affect the power of the test.

Normality Assumption Violated

To test violation of the normal assumption, equal variances and sample sizes were assumed equal (code outline may be found at the end of part 1). As per usual, 3 samples were used, each of 100 data points.

To first test violation of the normality assumption, I changed the distribution of the samples to represent a cauchy distribution via the r function rcauchy. Essentially, a cauchy distribution looks very similar to a normal distribution except that it has thicker tails. To my surprise, results showed rather immediately. Using 3 cauchy distributions, where the test sample had a mean of 0.5 and the other 2 samples had means of 0- with scales of 1 for each sample- I obtained powers ranging from 0.01 to 0.04. Wondering whether it was the fault of the scale (since the cauchy distribution *does* look rather similar to the null distribution), I reran the same test except that each sample had a scale of 3; nothing changed. In this way, it's clear to see that distributions that differ from the normal distribution have a relatively large effect on the power of the one-way ANOVA test- even if the distribution may look similar to normal distributions.

Cauchy Distribution



Community Structures among K-samples

In the case of looking at community structures among k-samples, our primary tool will be hierarchical clustering; in r, it's a simple task. Using the code that can be found at the end of the first part of this project, we find a hierarchical clustering tree. From there, we use the gap statistic to find an estimate for the optimal number of clusters. If we happen to have more than one optimal cluster and the clusters *don't* have a relatively even distribution of data from each sample, then our data would support the notion that not all the samples come from the same distribution. To test the gap statistic and hierarchical clustering, we run some experiments.

In the first we take 3 normally distributed samples of variance 1 and sample size 30 (size 30 for the sake of visualization via graphs). As can be seen in the gap statistic graph, when the means are all the same, the estimated optimal cluster is 1 and the table of data shows an approximately even distribution, within each cluster, of data from each sample.

However, when we change one of the samples to have mean 5, we get a new story. When we use a mean of 5 for one of the samples, we find that the new "optimal number of clusters" by the gap statistic is 2; looking at the table, we also find that the sample with a mean 5 essentially controls its own cluster. In this way, we find that hierarchical clustering can be used to find the community structures among k-samples.

One important thing to note, however, is that when the mean was less than 5 for the test sample, the gap statistic was unable to pick up on the number of optimal clusters (which should've been 2 assuming since we know that 2 of the samples had the same mean while the other had a different mean). So, for the purpose of more tests involving hierarchical clustering

and the gap statistic, I'll focus on the gap statistic but more so on the number of samples used and the table displaying the distribution of which samples the data in each cluster came from.

Code & Images

Testing Size Imbalance & Unequal Variance

```
mu = 0
numSamples = 100
powerDeterm = function(mu, sigma){
  rejectCount = 0
  #correctCount = 0
  for(x in 1:numSamples){
    #determine sample based on given distributions with a mean parameter

    #WEIRD SIZE (and in this case, dif mu)
    #sample1 = rnorm(100,mean=mu, sd=sqrt(1))
    #sample2 = rnorm(100,mean=mu+1, sd=sqrt(20))
    #(and mu gets weirder)
    #sample3 = rnorm(100,mean=mu, sd=sqrt(1))
    #power drops significantly
    #but when mu >1, it's a lot more clearly defined
    #sample4 = rnorm(100000,mean=mu, sd=sqrt(1))
    #sample5 = rnorm(100000,mean=mu, sd=sqrt(1))
    #sample6 = rnorm(100000,mean=mu, sd=sqrt(1))
    #sample7 = rnorm(100000,mean=mu, sd=sqrt(1))
    #sample8 = rnorm(100000,mean=mu, sd=sqrt(1))
    #sample9 = rnorm(100000,mean=mu, sd=sqrt(1))
    #sample10 = rnorm(100000,mean=mu, sd=sqrt(1))

    hist(sample1)
    samples = c(rep("sample1", 100), rep("sample2", 100), rep("sample3", 100))
    #rep("sample4", 100000), rep("sample5", 100000), rep("sample6", 100000),
    #rep("sample7", 100000), rep("sample8", 100000), rep("sample9", 100000),
    #rep("sample10", 100000))
    bigSample = c(sample1, sample2, sample3)
    #sample4, sample5, sample6,
    #sample7, sample8, sample9, sample10)
    df <- data.frame(samples, bigSample)
    anovaInfo = oneway.test(bigSample ~ samples, data=df, var.equal = TRUE)
    if(anovaInfo$p.value < 0.05){
      rejectCount = rejectCount + 1
    }
  }
  return(rejectCount/numSamples)

  #return proportion of correct rejection
  #RN since mu is the same, if it rejects it it's wrong
  #power = rejectCount/numSamples
  #return(power)
}

powerCount = 0
for(x in 1:numDist){
  check1 = powerDeterm(mu, sigma)
  powerCount = powerCount + check1
}
power = powerCount/numDist
power
```

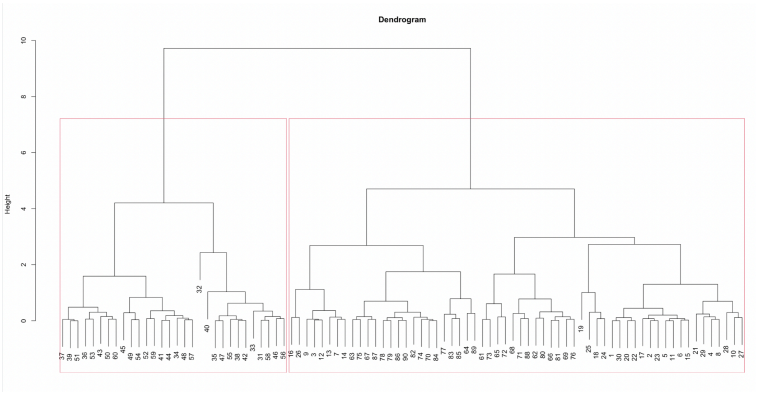
Testing Normality Assumption Violation

```r
powerDeterm = function(mu, sigma){
  rejectCount = 0
  #correctCount = 0
  for(x in 1:numSamples){
    #determine sample based on given distributions with a mean parameter

    #sample1 = rcauchy(100,location=mu+0.5, scale=3)
    #sample2 = rcauchy(100,location=mu, scale=3)
    #sample3 = rcauchy(100,location=mu, scale=3)

    samples = c(rep("sample1", 100), rep("sample2", 100), rep("sample3", 100))
    bigSample = c(sample1, sample2, sample3)
    df <- data.frame(samples, bigSample)
    anovaInfo = oneway.test(bigSample ~ samples, data=df, var.equal = TRUE)
    if(anovaInfo$p.value < 0.05){
      rejectCount = rejectCount + 1
    }
  }
  return(rejectCount/numSamples)

  #return proportion of correct rejection
  #RN since mu is the same, if it rejects it it's wrong
  #power = rejectCount/numSamples
  #return(power)
}

powerCount = 0
for(x in 1:numDist){
  check1 = powerDeterm(mu, sigma)
  powerCount = powerCount + check1
}
Power = powerCount/numDist
Power
```

Hierarchical Clustering Code

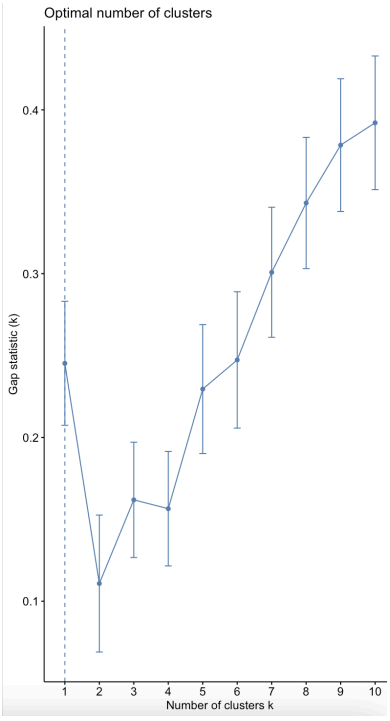
Finding HC Tree

```r
sample1 = rnorm(30,mean=mu, sd=sqrt(1))
sample2 = rnorm(30,mean=mu + 5, sd=sqrt(1))
sample3 = rnorm(30,mean=mu, sd=sqrt(1))
sample1 = cbind(sample1, 1)
sample2 = cbind(sample2, 2)
sample3 = cbind(sample3, 3)
mydata = rbind(sample1, sample2, sample3)
colnames(mydata) = c("sample data", "sample")
mydata = as.data.frame(mydata)
mydata
clusters = hclust(dist(mydata))

gap_stat <- clusGap(mydata, FUN=hcut, K.max=10)
fviz_gap_stat(gap_stat)

plot(clusters, xlab='', main='Dendrogram')
clusterCut <- cutree(clusters, 3)
rect.hclust(clusters, k=3)

table(clusterCut, mydata$sample)

```

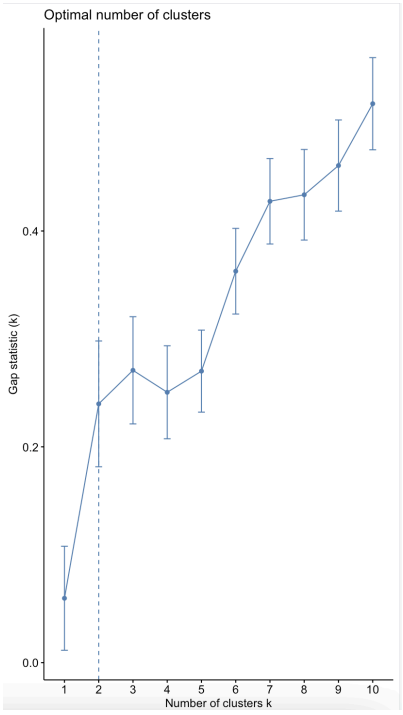


Gap Statistic Graphs

(Same Samples)



(Test sample mean 5)



Tables

For test sample of equal mean & test sample of mean 5 (respectively)

clusterCut	1	2	3
1	18	17	7
2	12	3	0
3	0	10	23

clusterCut	1	2	3
1	30	4	30
2	0	26	0

Application to Heart Disease

As part of this study, our primary focus will be on the interactions and effects on response variable BMI by explanatory variables “Education” (measured on a scale from 1-6 where 1 represents someone who’s never been to kindergarten and 6 is someone who’s gone through college or more), “Smoker” (whether or not the experimental subject was a smoker), “HvyAlcoholConsump” (whether or not the experimental subject had heavy alcohol consumption), and “HighBP” (whether or not the experimental subject had high blood pressure). In total, we looked at 48 exhaustive variables, each carrying vectors of BMI taken from experimental subjects. BMI data was distributed based on names “EdSmAlcBP####” where the first “#” represents education level, the second # represents whether or not they were a smoker, the third “#” representing whether they had heavy alcohol consumption, and the fourth “#” representing whether or not they had high blood pressure. For example, variable EdSmAlcBP4001 carries a vector of BMIs for experimental subjects at education level 4, who weren’t smokers, didn’t indulge in alcohol, and had high blood pressure.

Straight away, after conducting the one-way anova, we find that there is a significant difference; using the r code “oneway.test()”, we get a p-value of 2.2 x (10^-16). However, upon closer examination, by first looking at sample size, we find that various plots have sample sizes below 30. As seen in the first part of the project, samples with sample sizes below a certain threshold (often most will say 30), the power of the anova test goes down as the sample sizes become too small to be assumed normal. Of the samples with sizes too small, we find vectors EdSmAlcBP1010, EdSmAlcBP1011, EdSmAlcBP1110, EdSmAlcBP1111, EdSmAlcBP2010,

EdSmAlcBP2011, EdSmAlcBP2110, and EdSmAlcBP3011. Since samples were taken from America -one of the richest and, thus, more educated countries in the world- the number of experimental subjects with lower education levels tended to be less. Next, just to ensure normality, I plotted the remaining histograms of each variable,
and found that certain more variables inherently violate the normality assumption, such as EdSmAlcBP1100 (with a strong right skew) and EdSmAlcBP1101 (looking approximately uniform). Among the rest of the variables (38), we find that most are approximately normal with a slight skew (pictures may be found at the end of part 2). For the sake of equal variance, I conducted the nonparametric Fligner test and obtained an insignificant p-value, indicating that the equal variance assumption holds.

After determining that the equal variance and normality assumptions hold, we are ready to run the ANOVA test. In it, we obtain a p-value (2.2 x 10^-16, the smallest possible p-value in r), which is less than the significance level 0.05. So, our next step is to find which pairs, exactly, are considered significantly different from each other. To do so, we'll be using Tukey-Kramer's pairwise comparison. However, due to the large amount of data, we'll find pairwise comparisons within each level of education.

From the data, and Tukey-Kramer comparisons, we notice a few things: pairs 111 and 011 (SmAlcBP) have been deemed insignificant at every level of education; in education level 2, changes in HighBP are all deemed significant; in education level 3, every comparison involving 3001 was deemed significant; in education level 4, every comparison involving 4001 and 4101 was deemed significant; practically every comparison at education levels 5 and 6 were deemed significant; the ratio of significant pairs to total pairs increases at each level of education (*Note that none of the significant Tukey-Kramer confidence intervals were positive and when I say that*

a comparison involving #### is significant, I mean to say that that variable had a significant increasing effect on BMI). To summarize, the HighBP seems to be the variable with the most consistent and significant effect while other interaction variables don't seem to have significant effects. Many pairwise comparisons show significant differences at education levels 4, 5, 6 but not as much at education levels 1, 2, or 3. This could possibly be because factors such as heavy alcohol consumption or smoking might be enhanced at higher education levels; higher education is generally associated with higher income. With higher income, one has more potential to buy alcohol or cigarettes. At the same time, most factor variables at education levels 1, 2, and 3 are inconsistent in that they may be significant compared to some variables but insignificant with respect to others; the only consistent variable that seemed to have a great effect on BMI, across the different education levels, was high blood pressure.

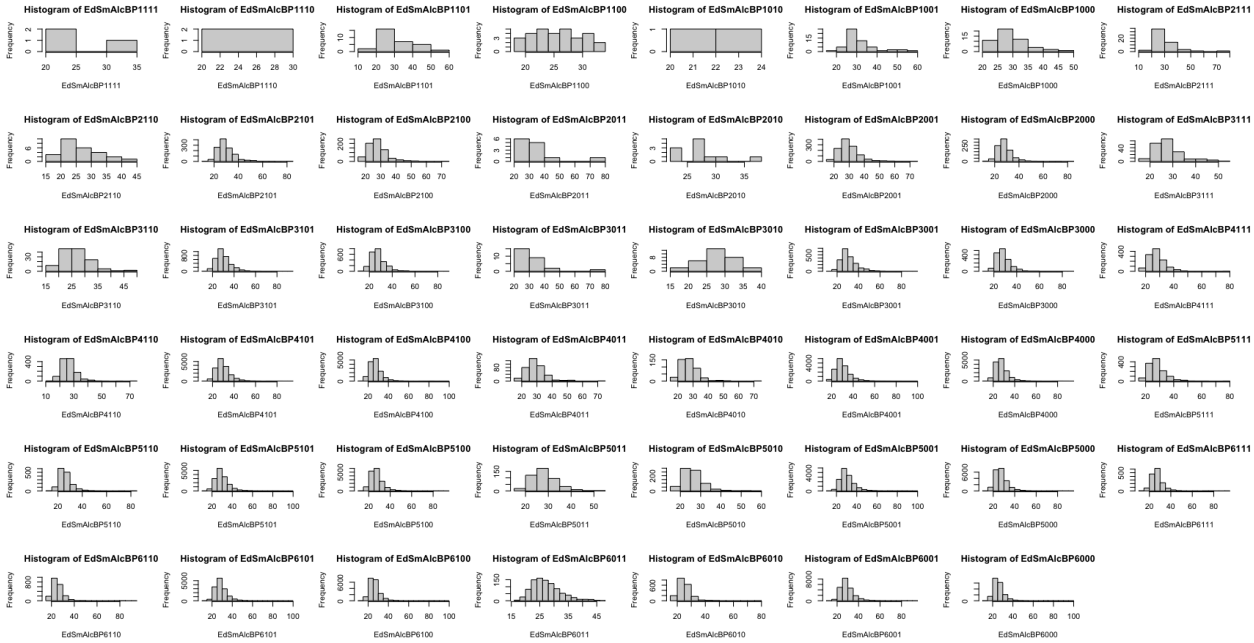
Another method we can use is hierarchical clustering, as stated in the first part. However, due to the large nature of the data, it's impossible to create a hierarchical tree of the whole dataset with my current technology (as well as use the gap statistic clustering method). So, for the sake of simplicity, I'll be creating HC (hierarchical clustering) trees based on only certain comparisons involving education levels 1, 2, and 3 (education levels 4, 5, and 6 contain data too large) that can be found at the end of part 2. As it so happens, none of the interactions within education levels 1-3 seem to have a significant effect. Within each cluster found in the tables, it appears as though the distribution of sample interactions is relatively even, where there's no instance of a single sample fully dominating a single cluster. This seems to fit in relatively well with the ANOVA results as we know that interactions in education levels 1, 2, and 3 (based on Tukey's pairwise comparison) didn't seem to be as significant as those in the higher-up levels of education.

Compared with ANOVA & Tukey-Kramer's pairwise comparisons, the hierarchical clustering trees and tables fare far better in terms of visualization. When it comes to such large amounts of data, it becomes increasingly difficult to test for normality and equal variance with ANOVA. Since there are such large amounts of samples, it's also very likely that a sample might be considered significant and so, another question becomes whether or not that sample is significant because of the explanatory variables or because of other unintended factors. Then, it becomes even harder to determine what significant Tukey-Kramer comparisons mean as there could potentially be many comparisons that are significantly different, insignificant, and contradicting. In terms of ease in interpretation and understanding, hierarchical clustering shines much better as one can clearly see the different groups made (though it may be difficult to identify in the tree) and the distributions of each sample in the tables.

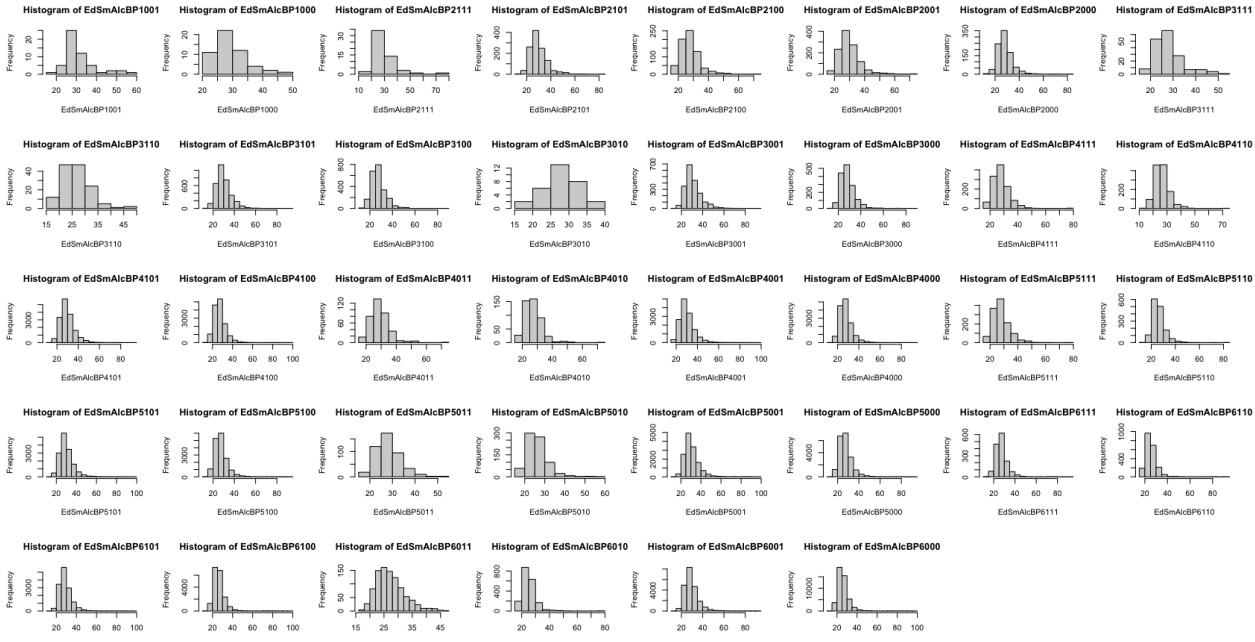
However, one potential (and perhaps the primary) issue with hierarchical clustering is that it requires a large amount of memory storage, as well as a good computer. ANOVA, on the other hand, works with enormous amounts of data and runs relatively smoothly (though it may be hard to understand). Hierarchical clustering, on the other hand, is far easier to understand- but it requires more resources to run and create. I didn't even have enough storage or memory to compute the gap statistic for any of the individual education levels.

Code & Images

Original Set of Graphs



Set of Graphs (filtered for normality)



Tukey Pairwise Comparisons

Within Education Level 1

```
$bigSampleNames
                                 diff        lwr     upr     p adj
EdSmAlcBP1001-EdSmAlcBP1000 2.356125 -0.3647791 5.07703 0.0889227
```

Within Education Level 2

```
$bigSampleNames
                                  diff        lwr        upr     p adj
EdSmAlcBP2001-EdSmAlcBP2000  1.6972659  0.8553098  2.5392220 0.0000004
EdSmAlcBP2100-EdSmAlcBP2000 -0.5782086 -1.5309617  0.3745445 0.4614793
EdSmAlcBP2101-EdSmAlcBP2000  1.5013067  0.6649336  2.3376799 0.0000100
EdSmAlcBP2111-EdSmAlcBP2000  0.9693728 -1.6664285  3.6051740 0.8537948
EdSmAlcBP2100-EdSmAlcBP2001 -2.2754745 -3.1862901 -1.3646589 0.0000000
EdSmAlcBP2101-EdSmAlcBP2001 -0.1959592 -0.9842272  0.5923089 0.9611306
EdSmAlcBP2111-EdSmAlcBP2001 -0.7278931 -3.3488270  1.8930408 0.9424816
EdSmAlcBP2101-EdSmAlcBP2100  2.0795153  1.1738581  2.9851726 0.0000000
EdSmAlcBP2111-EdSmAlcBP2100  1.5475814 -1.1110166  4.2061793 0.5047962
EdSmAlcBP2111-EdSmAlcBP2101 -0.5319339 -3.1510797  2.0872118 0.9814387
```

Within Education Level 3

```
$bigSampleNames
                                  diff        lwr        upr     p adj
EdSmAlcBP3010-EdSmAlcBP3001 -3.288632486 -7.0188764  0.4416114 0.1203526
EdSmAlcBP3100-EdSmAlcBP3001 -3.302420780 -3.9359165 -2.6689251 0.0000000
EdSmAlcBP3101-EdSmAlcBP3001 -0.998912859 -1.5837298 -0.4140959 0.0000169
EdSmAlcBP3110-EdSmAlcBP3001 -2.891013850 -3.5745562 -2.2074715 0.0000000
EdSmAlcBP3111-EdSmAlcBP3001 -2.887631373 -4.5170194 -1.2582434 0.0000066
EdSmAlcBP3100-EdSmAlcBP3010 -0.013788295 -3.7404629  3.7128863 1.0000000
EdSmAlcBP3101-EdSmAlcBP3010  2.289719626 -1.4289895  6.0084287 0.4952278
EdSmAlcBP3110-EdSmAlcBP3010  0.397618635 -3.3378889  4.1331262 0.9996578
EdSmAlcBP3111-EdSmAlcBP3010  0.401001112 -3.6166713  4.4186736 0.9997504
EdSmAlcBP3101-EdSmAlcBP3100  2.303507921  1.7419077  2.8651081 0.0000000
EdSmAlcBP3110-EdSmAlcBP3100  0.411406930 -0.2523808  1.0751947 0.4876053
EdSmAlcBP3111-EdSmAlcBP3100  0.414789407 -1.2064105  2.0359893 0.9783843
EdSmAlcBP3110-EdSmAlcBP3101 -1.892100991 -2.5096026 -1.2745994 0.0000000
EdSmAlcBP3111-EdSmAlcBP3101 -1.888718514 -3.4915233 -0.2859137 0.0102085
EdSmAlcBP3111-EdSmAlcBP3110  0.003382477 -1.6380201  1.6447851 1.0000000
```

Within Education Level 4

```
$bigSampleNames
                                    diff           lwr         upr     p adj
EdSmAlcBP4010-EdSmAlcBP4001 -2.89650071 -3.8189376475 -1.97406378 0.0000000
EdSmAlcBP4011-EdSmAlcBP4001 -1.11176895 -2.1411158399 -0.08242206 0.0244566
EdSmAlcBP4100-EdSmAlcBP4001 -2.94973715 -3.1807055950 -2.71876871 0.0000000
EdSmAlcBP4101-EdSmAlcBP4001 -0.53836440 -0.7643554200 -0.31237338 0.0000000
EdSmAlcBP4110-EdSmAlcBP4001 -2.72645638 -2.9533836407 -2.49952911 0.0000000
EdSmAlcBP4111-EdSmAlcBP4001 -2.22353388 -2.8105669474 -1.63650082 0.0000000
EdSmAlcBP4011-EdSmAlcBP4010  1.78473176  0.4224587853  3.14700474 0.0021582
EdSmAlcBP4100-EdSmAlcBP4010 -0.05323644 -0.9749625727  0.86848970 0.9999980
EdSmAlcBP4101-EdSmAlcBP4010  2.35813632  1.4376448220  3.27862781 0.0000000
EdSmAlcBP4110-EdSmAlcBP4010  0.17004434 -0.7506774616  1.09076614 0.9981584
EdSmAlcBP4111-EdSmAlcBP4010  0.67296683 -0.3951345054  1.74106817 0.5087978
EdSmAlcBP4100-EdSmAlcBP4011 -1.83796820 -2.8666781630 -0.80925823 0.0000029
EdSmAlcBP4101-EdSmAlcBP4011  0.57340455 -0.4541993150  1.60100842 0.6527738
EdSmAlcBP4110-EdSmAlcBP4011 -1.61468742 -2.6424975978 -0.58687725 0.0000741
EdSmAlcBP4111-EdSmAlcBP4011 -1.11176493 -2.2734471533  0.04991729 0.0712633
EdSmAlcBP4101-EdSmAlcBP4100  2.41137275  2.1883007490  2.63444475 0.0000000
EdSmAlcBP4110-EdSmAlcBP4100  0.22328077 -0.0007396715  0.44730122 0.0514182
EdSmAlcBP4111-EdSmAlcBP4100  0.72620327  0.1402877484  1.31211879 0.0048154
EdSmAlcBP4110-EdSmAlcBP4101 -2.18809198 -2.4069770606 -1.96920690 0.0000000
EdSmAlcBP4111-EdSmAlcBP4101 -1.68516948 -2.2691408150 -1.10119815 0.0000000
EdSmAlcBP4111-EdSmAlcBP4110  0.50292250 -0.0814117909  1.08725678 0.1459335
```

Within Education Level 5

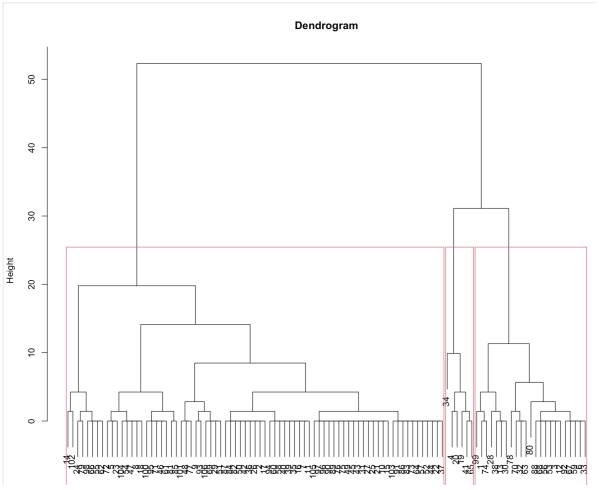
```
$bigSampleNames
                                   diff         lwr          upr     p adj
EdSmAlcBP5010-EdSmAlcBP5001 -4.3925346 -5.11642538 -3.668643813 0.0000000
EdSmAlcBP5011-EdSmAlcBP5001 -2.2904721 -3.22062416 -1.360320127 0.0000000
EdSmAlcBP5100-EdSmAlcBP5001 -3.2164986 -3.44042135 -2.992575953 0.0000000
EdSmAlcBP5101-EdSmAlcBP5001 -0.7657606 -0.99408801 -0.537433143 0.0000000
EdSmAlcBP5110-EdSmAlcBP5001 -3.2285729 -3.43952779 -3.017618106 0.0000000
EdSmAlcBP5111-EdSmAlcBP5001 -2.5843341 -3.15656159 -2.012106585 0.0000000
EdSmAlcBP5011-EdSmAlcBP5010  2.1020625  0.94662281  3.257502107 0.0000017
EdSmAlcBP5100-EdSmAlcBP5010  1.1760359  0.45492612  1.897145780 0.0000313
EdSmAlcBP5101-EdSmAlcBP5010  3.6267740  2.90428428  4.349263766 0.0000000
EdSmAlcBP5110-EdSmAlcBP5010  1.1639617  0.44677273  1.881150582 0.0000352
EdSmAlcBP5111-EdSmAlcBP5010  1.8082005  0.91528264  2.701118385 0.0000000
EdSmAlcBP5100-EdSmAlcBP5011 -0.9260265 -1.85401589  0.001962872 0.0509046
EdSmAlcBP5101-EdSmAlcBP5011  1.5247116  0.59564949  2.453773634 0.0000270
EdSmAlcBP5110-EdSmAlcBP5011 -0.9381008 -1.86304667 -0.013154936 0.0442633
EdSmAlcBP5111-EdSmAlcBP5011 -0.2938619 -1.36085175  0.773127852 0.9838404
EdSmAlcBP5101-EdSmAlcBP5100  2.4507381  2.23138690  2.670089241 0.0000000
EdSmAlcBP5110-EdSmAlcBP5100 -0.0120743 -0.21327932  0.189130731 0.9999974
EdSmAlcBP5111-EdSmAlcBP5100  0.6321646  0.06345916  1.200869963 0.0181098
EdSmAlcBP5110-EdSmAlcBP5101 -2.4628124 -2.66890823 -2.256716499 0.0000000
EdSmAlcBP5111-EdSmAlcBP5101 -1.8185735 -2.38902761 -1.248119412 0.0000000
EdSmAlcBP5111-EdSmAlcBP5110  0.6442389  0.08051339  1.207964327 0.0133069
```

Within Education Level 6

```
$bigSampleNames
                                    diff        lwr        upr     p adj
EdSmAlcBP6010-EdSmAlcBP6001 -4.1121114 -4.5367530 -3.6874698 0.0000000
EdSmAlcBP6011-EdSmAlcBP6001 -1.8414307 -2.4402033 -1.2426580 0.0000000
EdSmAlcBP6100-EdSmAlcBP6001 -2.8429616 -3.0208485 -2.6650747 0.0000000
EdSmAlcBP6101-EdSmAlcBP6001 -0.3763046 -0.5690112 -0.1835980 0.0000002
EdSmAlcBP6110-EdSmAlcBP6001 -3.2384048 -3.3846710 -3.0921386 0.0000000
EdSmAlcBP6111-EdSmAlcBP6001 -2.0392590 -2.5148996 -1.5636184 0.0000000
EdSmAlcBP6011-EdSmAlcBP6010  2.2706807  1.5570504  2.9843111 0.0000000
EdSmAlcBP6100-EdSmAlcBP6010  1.2691498  0.8420861  1.6962134 0.0000000
EdSmAlcBP6101-EdSmAlcBP6010  3.7358068  3.3023608  4.1692528 0.0000000
EdSmAlcBP6110-EdSmAlcBP6010  0.8737066  0.4588169  1.2885962 0.0000000
EdSmAlcBP6111-EdSmAlcBP6010  2.0728524  1.4588707  2.6868341 0.0000000
EdSmAlcBP6100-EdSmAlcBP6011 -1.0015310 -1.6020237 -0.4010382 0.0000181
EdSmAlcBP6101-EdSmAlcBP6011  1.4651261  0.8600776  2.0701746 0.0000000
EdSmAlcBP6110-EdSmAlcBP6011 -1.3969741 -1.9888708 -0.8050775 0.0000000
EdSmAlcBP6111-EdSmAlcBP6011 -0.1978283 -0.9429335  0.5472769 0.9866522
EdSmAlcBP6101-EdSmAlcBP6100  2.4666571  2.2686704  2.6646437 0.0000000
EdSmAlcBP6110-EdSmAlcBP6100 -0.3954432 -0.5485989 -0.2422875 0.0000000
EdSmAlcBP6111-EdSmAlcBP6100  0.8037026  0.3258985  1.2815068 0.0000146
EdSmAlcBP6110-EdSmAlcBP6101 -2.8621002 -3.0322436 -2.6919569 0.0000000
EdSmAlcBP6111-EdSmAlcBP6101 -1.6629544 -2.1464716 -1.1794372 0.0000000
EdSmAlcBP6111-EdSmAlcBP6110  1.1991458  0.7321909  1.6661007 0.0000000
```

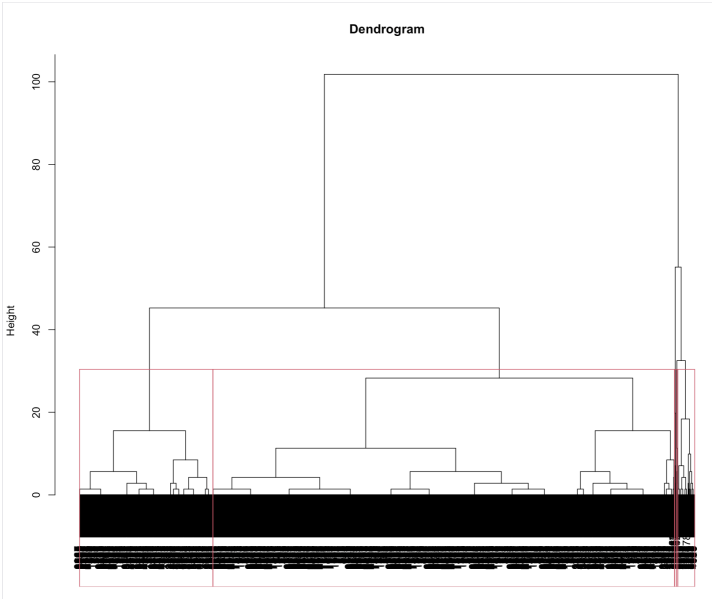
Hierarchical Clustering Trees & Tables

Education Level 1



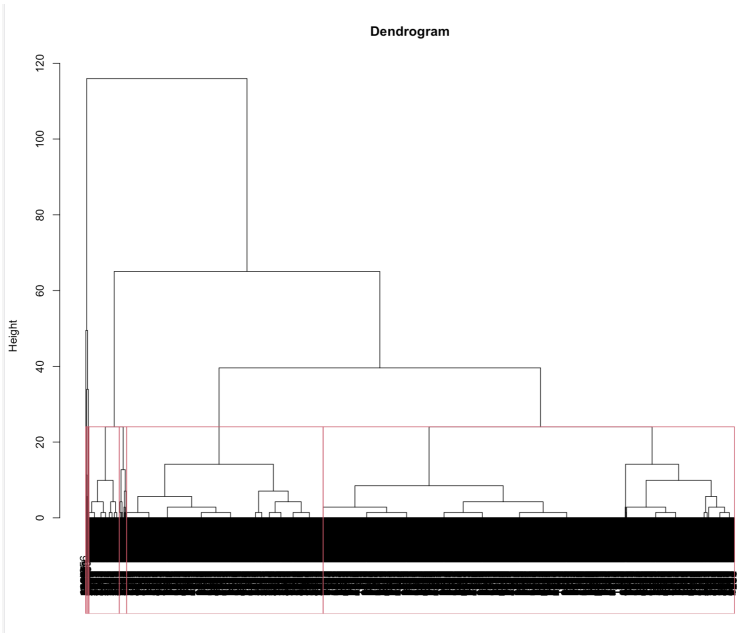
clusterCut	EdSmAlcBP1001	EdSmAlcBP1110
1	11	12
2	38	39
3	5	1

Education Level 2



clusterCut	EdSmAlcBP2000	EdSmAlcBP2001	EdSmAlcBP2100	EdSmAlcBP2101	EdSmAlcBP2111
1	746	797	577	834	42
2	150	303	107	295	11
3	3	4	1	2	1
4	15	34	16	44	1
5	2	5	0	3	0

Education Level 3



clusterCut	EdSmAlcBP3001	EdSmAlcBP3010	EdSmAlcBP3100	EdSmAlcBP3101	EdSmAlcBP3110	EdSmAlcBP3111
1	1089	21	1653	1932	1170	127
2	703	10	535	1138	442	35
3	140	0	51	193	48	10
4	38	0	19	37	11	2
5	10	0	2	11	2	0
6	3	0	6	3	3	0
7	2	0	1	3	1	0