Razorback Sucker Elemental Isotope Analysis

Alejandro Aragon

12/21/2021

Contents

1 San Juan River Razorback Suckers

Razorback Suckers were collected in 2014 on the San Juan River. Elemental isotopic ratios from razorback suckers were analyzed for Ba (Barium 56), Ca (Calcium 20), Mg (Magnesium 12), and Sr (Strontium 38). razorback suckers are non-lethally obtained and are used to detect natal origin since the material in the suckers are partially developed early in life.

The issue is that hatchery fish can get into the river and lose their tags. It is important for environmental resource managers to know whether untagged fish are wild or hatchery fish. There are five fish sources in the dataset.

```
5 Sources

Hatchery
  DEX = Dexter National Fish Hatchery
  GJH = Ouray National Fish Hatchery, Grand Valley Unit

Wild
  NAP = NAPI ponds
  SJR = San Juan River

Unknown
  UNK = untagged Razorback Suckers captured in the San Juan River
        these could be from any of the above sources
```

Goal is to test whether the known source populations have different multivariate means, and if so, if it is possible to build a model that predicts the origin of razorback suckers.

2 Data

2.1 Summary

```
sjrs.full <- read.csv("F:/Data Analysis/SanJuanRazorbackSuckers_data2014.csv")
str(sjrs.full)
```

```
'data.frame':   1512 obs. of  13 variables:
 $ Sort.Key: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Station : Factor w/ 67 levels "GJHFR-127A","GJHFR-130A",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Source  : Factor w/ 5 levels "DEX","GJH","NAP",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ Type    : Factor w/ 1 level "FR": 1 1 1 1 1 1 1 1 1 1 ...
 $ Ba137   : num  0.00193 0.00194 0.00194 0.00183 0.00178 ...
 $ Ba138   : num  0.012 0.0118 0.0122 0.012 0.0111 ...
 $ Ca43    : num  0.664 0.652 0.691 0.654 0.65 ...
 $ Mg24    : num  1.83 1.86 1.91 1.93 1.98 ...
 $ Mg25    : num  0.237 0.247 0.25 0.254 0.252 ...
 $ Mg26    : num  0.27 0.277 0.289 0.284 0.284 ...
 $ Sr86    : num  0.155 0.157 0.16 0.158 0.153 ...
 $ Sr87    : num  0.112 0.11 0.115 0.109 0.107 ...
 $ Sr88    : num  1.34 1.28 1.34 1.32 1.31 ...
```

```
summary(sjrs.full)
```

```
   Sort.Key            Station     Source    Type          Ba137
 Min.   :   1.0   SJEFR-144A:  89   DEX:224   FR:1512   Min.   :0.00023
 1st Qu.: 378.8   SJRFR-145A:  59   GJH:199             1st Qu.:0.00148
 Median : 763.5   SJRFR-153A:  54   NAP:133             Median :0.00257
 Mean   : 744.4   GJHFR-134A:  53   SJR:244             Mean   :0.00676
 3rd Qu.:1109.2   NAPFR-114A:  51   UNK:712             3rd Qu.:0.00836
 Max.   :1454.0   GJHFR-127A:  48                       Max.   :0.04396
                  (Other)   :1158                       NA's   :65
     Ba138               Ca43             Mg24            Mg25
 Min.   :0.001496   Min.   :0.3900   Min.   :1.572   Min.   :0.2124
 1st Qu.:0.009572   1st Qu.:0.6552   1st Qu.:2.105   1st Qu.:0.2739
 Median :0.018626   Median :0.6630   Median :2.245   Median :0.2917
 Mean   :0.044593   Mean   :0.6608   Mean   :2.246   Mean   :0.2915
 3rd Qu.:0.055527   3rd Qu.:0.6706   3rd Qu.:2.391   3rd Qu.:0.3103
 Max.   :0.332417   Max.   :0.7069   Max.   :3.110   Max.   :0.3921
                                     NA's   :65      NA's   :65
      Mg26             Sr86             Sr87              Sr88
 Min.   :0.2410   Min.   :0.05806   Min.   :0.04093   Min.   :0.4031
 1st Qu.:0.3125   1st Qu.:0.15245   1st Qu.:0.10864   1st Qu.:1.2805
 Median :0.3328   Median :0.16947   Median :0.12117   Median :1.4344
 Mean   :0.3325   Mean   :0.19910   Mean   :0.14172   Mean   :1.7153
 3rd Qu.:0.3536   3rd Qu.:0.23823   3rd Qu.:0.17012   3rd Qu.:2.1281
 Max.   :0.4450   Max.   :0.57698   Max.   :0.41333   Max.   :4.8600
 NA's   :65       NA's   :65        NA's   :65
```

Adding `Source.Type` as a variable for a more general view of the sources.

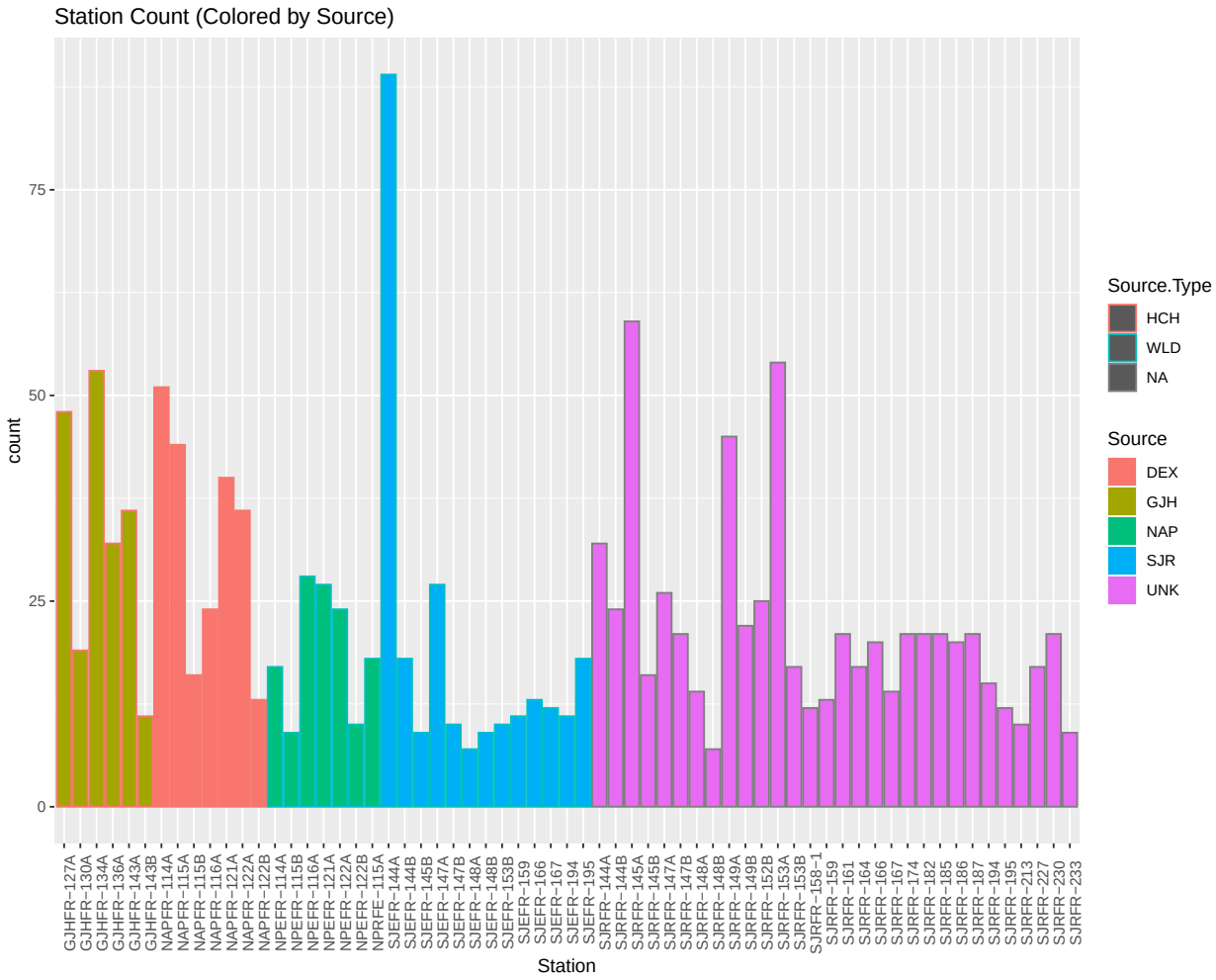
```
Source Type

HCH = Hatchery
    DEX = Dexter National Fish Hatchery
    GJH = Ouray  National Fish Hatchery, Grand Valley Unit

WLD = Wild
    NAP = NAPI ponds
    SJR = San Juan River

Unknown
  UNK = untagged Razorback Suckers captured in the San Juan River
        these could be from any of the above sources
```

```
sjrs.full <- na.omit(sjrs.full)
sjrs.full$Source.Type <- rep(NA, nrow(sjrs.full))
sjrs.full$Source.Type[ sjrs.full$Source %in% c("DEX", "GJH") ] <- "HCH"
sjrs.full$Source.Type[ sjrs.full$Source %in% c("NAP", "SJR") ] <- "WLD"
sjrs.full$Source.Type <- as.factor(sjrs.full$Source.Type)
```

2.2 Preliminary Visualization

`Station` has 67 levels making a pairing table challenging to read.

```
library(ggplot2)

p <- ggplot(sjrs.full, aes(Station, fill = Source, color = Source.Type))
p <- p + geom_bar(stat = "count")
p <- p + theme(axis.text.x = element_text(angle = 90))
p <- p + labs( title = "Station Count (Colored by Source)")
print(p)
```



Subsetting dataset to remove `Sort.Key`, `Station`, and `Type`. Suckers from an unknown source is also removed here as it provides no information to the goal (identifying source by isotopes).

```
sjrs <- subset(sjrs.full, subset = (Source != "UNK"), select = c(Source, Ba137:Source.Type))

sjrs.t <- subset(sjrs.full, select = c(Source, Ba137:Source.Type))

str(sjrs)
```

```
'data.frame':   800 obs. of  11 variables:
 $ Source     : Factor w/ 5 levels "DEX","GJH","NAP",..: 2 2 2 2 2 2 2 2 2 2 ...
```

```
 $ Ba137      : num  0.00193 0.00194 0.00194 0.00183 0.00178 ...
 $ Ba138      : num  0.012 0.0118 0.0122 0.012 0.0111 ...
 $ Ca43       : num  0.664 0.652 0.691 0.654 0.65 ...
 $ Mg24       : num  1.83 1.86 1.91 1.93 1.98 ...
 $ Mg25       : num  0.237 0.247 0.25 0.254 0.252 ...
 $ Mg26       : num  0.27 0.277 0.289 0.284 0.284 ...
 $ Sr86       : num  0.155 0.157 0.16 0.158 0.153 ...
 $ Sr87       : num  0.112 0.11 0.115 0.109 0.107 ...
 $ Sr88       : num  1.34 1.28 1.34 1.32 1.31 ...
 $ Source.Type: Factor w/ 2 levels "HCH","WLD": 1 1 1 1 1 1 1 1 1 1 ...
```

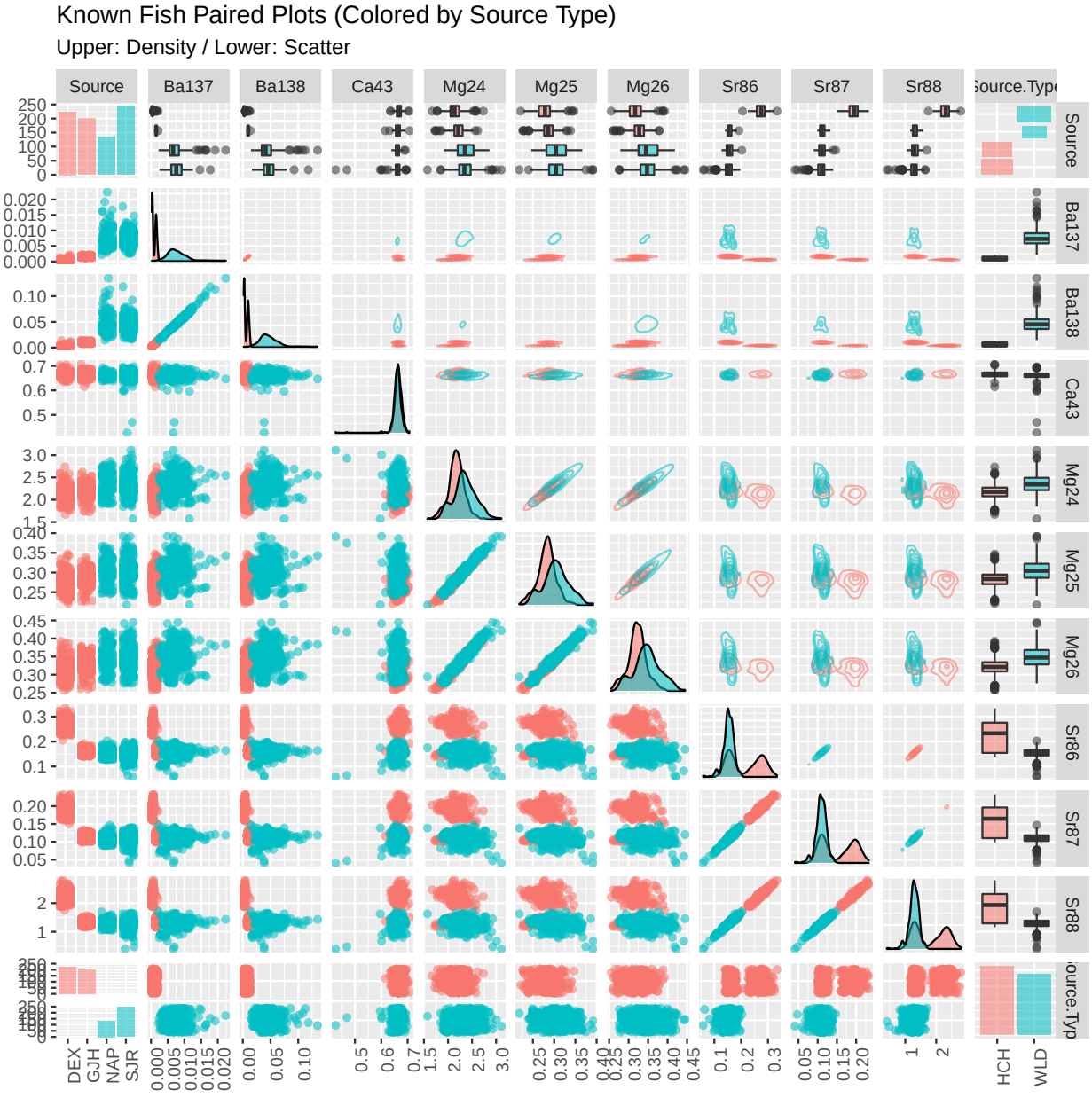
```
summary(sjrs)
```

```
 Source        Ba137               Ba138               Ca43
 DEX:224   Min.   :0.0002266   Min.   :0.001496   Min.   :0.4265
 GJH:199   1st Qu.:0.0006141   1st Qu.:0.003840   1st Qu.:0.6559
 NAP:133   Median :0.0019209   Median :0.012029   Median :0.6634
 SJR:244   Mean   :0.0041092   Mean   :0.025309   Mean   :0.6629
 UNK:  0   3rd Qu.:0.0071076   3rd Qu.:0.043307   3rd Qu.:0.6708
           Max.   :0.0223684   Max.   :0.134979   Max.   :0.7069
      Mg24            Mg25             Mg26            Sr86
 Min.   :1.572   Min.   :0.2179   Min.   :0.2553   Min.   :0.05806
 1st Qu.:2.123   1st Qu.:0.2750   1st Qu.:0.3137   1st Qu.:0.14954
 Median :2.241   Median :0.2911   Median :0.3317   Median :0.16300
 Mean   :2.254   Mean   :0.2925   Mean   :0.3338   Mean   :0.18738
 3rd Qu.:2.382   3rd Qu.:0.3094   3rd Qu.:0.3524   3rd Qu.:0.23578
 Max.   :3.110   Max.   :0.3921   Max.   :0.4450   Max.   :0.33371
      Sr87              Sr88        Source.Type
 Min.   :0.04093   Min.   :0.4031   HCH:423
 1st Qu.:0.10636   1st Qu.:1.2505   WLD:377
 Median :0.11610   Median :1.3585
 Mean   :0.13333   Mean   :1.5681
 3rd Qu.:0.16835   3rd Qu.:1.9651
 Max.   :0.23324   Max.   :2.7958
```

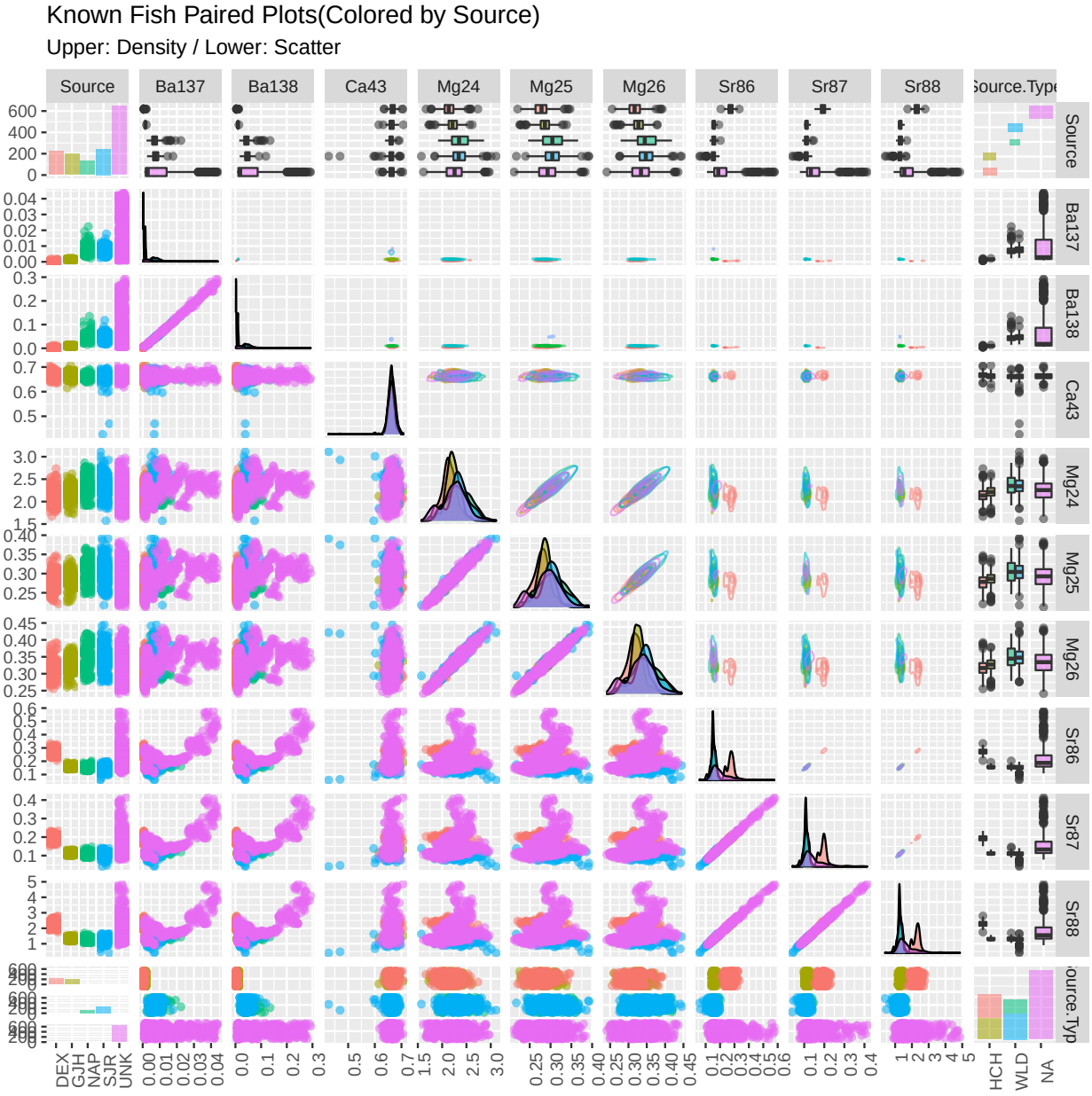
Subsetting data has removed any missing information.

```
# Scatterplot matrix
library(ggplot2)
library(GGally)

p <- ggpairs(sjrs
             , mapping = ggplot2::aes(colour = Source.Type, alpha = 0.5)
             , upper = list(continuous = "density", combo = "box")
             , lower = list(continuous = "points", combo = "dot")
             #, lower = list(continuous = "cor")
             )
p <- p + theme(axis.text.x = element_text(angle = 90))
p <- p + labs(title = "Known Fish Paired Plots (Colored by Source Type)"
              , subtitle = "Upper: Density / Lower: Scatter")

p1 <- ggpairs(sjrs.t
             , mapping = ggplot2::aes(colour = Source, alpha = 0.5)
```

```
            , upper = list(continuous = "density", combo = "box")
            , lower = list(continuous = "points", combo = "dot")
            #, lower = list(continuous = "cor")
            )
p1 <- p1 + theme(axis.text.x = element_text(angle = 90))
p1 <- p1 + labs(title = "Known Fish Paired Plots(Colored by Source)"
                , subtitle = "Upper: Density / Lower: Scatter")

print(p)
```




```
print(p1)
```



2.2.1 Preliminary Scatter Plot Observations

- High correlation between among isotopes of the same element.
  - Dropping isotopes so one isotope of each element is represented in the model
  - a principle component of each element to be considered
- Barium isotopes would benefit from a transformation to normalize the distribution
- Calcium isotope has two observations that are skewing the distribution.
- Strontium isotopes are indicative of razorback suckers from Dexter National Fish Hatchery
- Source Type adds no obvious information for preliminary inspection
  - Drop Source Type Variable

2.3 Clean and Transforming data

Transformations

```
sjrs$Ba137lg <- log10(sjrs$Ba137)
sjrs$Ba138lg <- log10(sjrs$Ba138)
```

Restriction

```
sjrs <- sjrs[!(sjrs$Ca43 < .55),]
```

2.3.1 Partitioning Data For Training

```
library(caret)

set.seed(1122)
trainIndex <- createDataPartition(sjrs$Source, p = .7, list = FALSE, times = 1)
train.dat <- sjrs[ trainIndex,]
test.dat <- sjrs[-trainIndex,]
```

Train Data Visualization (After Transformations)

Unnecessary step, only done to confirm log transformation is appropriate for training data.

2.4 Data Summary

Razorback suckers from the Dexter National Fish Hatchery can be easily identified by most combinations of the predictor variables. Ba138 in combination with any other predictor variable allows suckers from the Ouray National Fish Hatchery, Grand Valley Unit to be visually categorized.

3 Multivariate Analysis of Variance (MANOVA)

3.0.1 Assumptions

Shapiro-Wilk test for multivariate normality, as well as QQ-plots comparing the Mahalanobis D2 distance to a chi-squared distribution.

Test Hypothesis

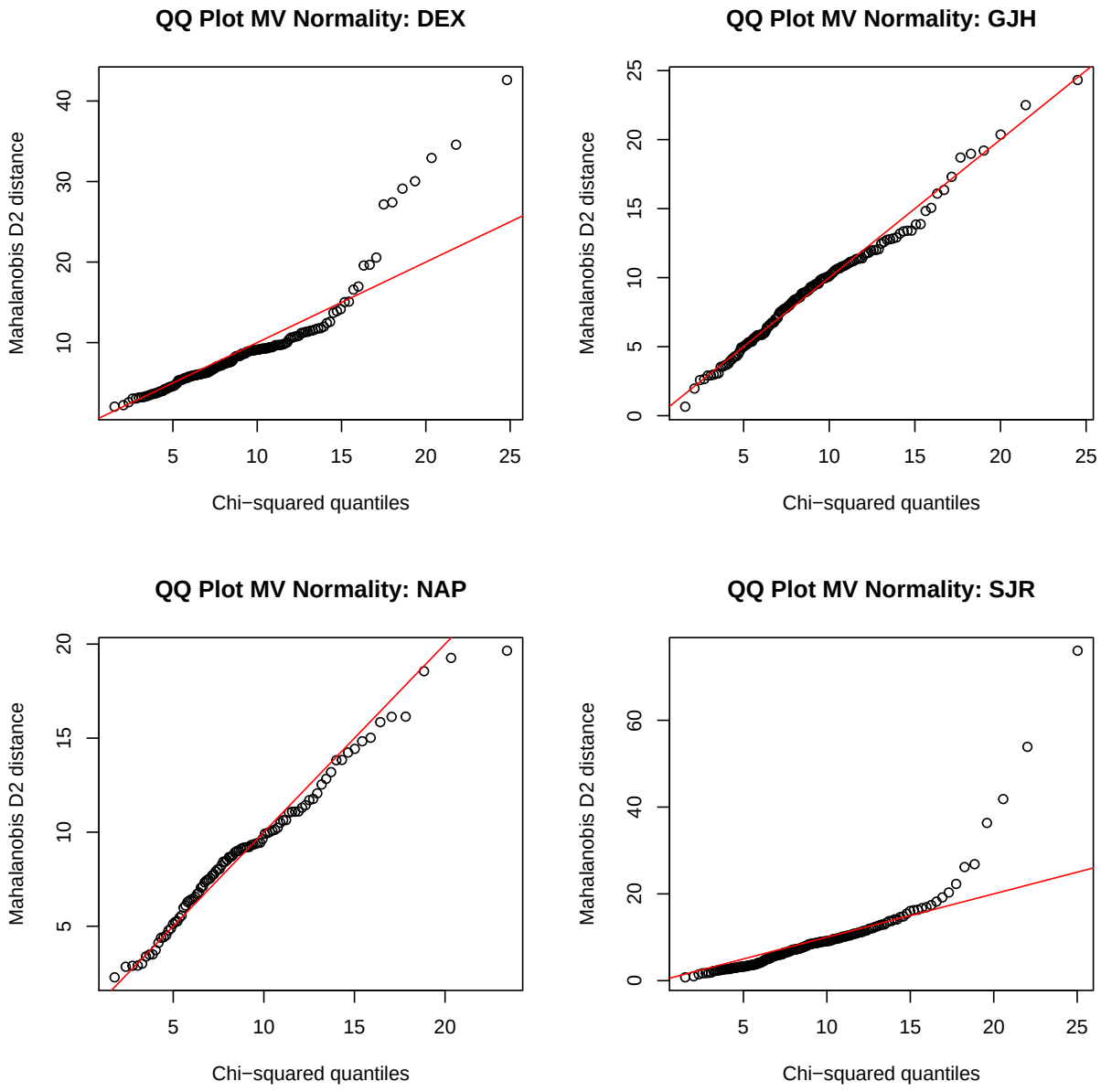
Null hypothesis: The data is normally distributed.

Alternative hypothesis: Data is not normally distributed.

If p-value $< .05$ reject H_0.

```
library(mvnormtest)

iso.vars <- c("Ca43", "Mg24", "Mg25", "Mg26", "Sr86"
              , "Sr87", "Sr88", "Ba137lg", "Ba138lg" )

# The data needs to be transposed t() so each variable is a row with observations as columns.
sw.dex <- mshapiro.test(t(subset(train.dat, subset = (Source == "DEX"), select = iso.vars)))
sw.gjh <- mshapiro.test(t(subset(train.dat, subset = (Source == "GJH"), select = iso.vars)))
sw.nap <- mshapiro.test(t(subset(train.dat, subset = (Source == "NAP"), select = iso.vars)))
sw.sjr <- mshapiro.test(t(subset(train.dat, subset = (Source == "SJR"), select = iso.vars)))
sw.results <- list(sw.dex, sw.gjh, sw.nap, sw.sjr)
```

```
library(mvnormtest)

# Graphical Assessment of Multivariate Normality
f.mnv.norm.qqplot <- function(x, name = "") {
  x <- as.matrix(x)
  center <- colMeans(x)
  n <- nrow(x);
  p <- ncol(x);
  cov <- cov(x);
  d <- mahalanobis(x, center, cov) # distances
  qqplot(qchisq(ppoints(n), df=p), d
    , main=paste("QQ Plot MV Normality:", name)
    , ylab="Mahalanobis D2 distance"
    , xlab="Chi-squared quantiles")
  abline(a = 0, b = 1, col = "red")
}

par(mfrow=c(2,2))
f.mnv.norm.qqplot(subset(train.dat, subset = (Source == "DEX"), select = iso.vars), "DEX")
f.mnv.norm.qqplot(subset(train.dat, subset = (Source == "GJH"), select = iso.vars), "GJH")
f.mnv.norm.qqplot(subset(train.dat, subset = (Source == "NAP"), select = iso.vars), "NAP")
f.mnv.norm.qqplot(subset(train.dat, subset = (Source == "SJR"), select = iso.vars), "SJR")
```



Results

p-value for the Shapiro-Wilk test for Dexter National Fish Hatchery (DEX) is 4.33342×10^{-13}. $4.33342 \times 10^{-13} < .05$, reject H_0 and conclude that the observations for DEX are not normally distributed.

p-value for the Shapiro-Wilk test for Ouray National Fish Hatchery, Grand Valley Unit (GJH) is 9.6302×10^{-5}. $9.6302 \times 10^{-5} < .05$, fail to reject H_0 and conclude that the observations for GJH are not normally distributed.

p-value for the Shapiro-Wilk test for NAPI ponds (NAP) is 7.97858×10^{-4}. $7.97858 \times 10^{-4} < .05$ fail to reject H_0 and conclude that the observations for NAP are not normally distributed.

p-value for the Shapiro-Wilk test for San Juan River (SJR) is 9.7066×10^{-16}. $9.7066 \times 10^{-16} < .05$ reject H_0 and conclude that the observations for SJR are not normally distributed.

Conclusion on Assumption of Normality

Since the assumption of normality is not met a non-parametric measure of MANOVA will need to be done.

3.1 MANOVA Model

MANOVA will be measured by the Hotelling-Lawley trace test.

3.1.1 Comparison Across Sources

Test Hypothesis

$$H_0 : \mu_{DEX} = \mu_{GJH} = \mu_{NAP} = \mu_{SJR} \text{ versus } H_\alpha : \mu_{DEX} \neq \mu_{GJH} \neq \mu_{NAP} \neq \mu_{SJR}$$

Null hypothesis: The fish from the four sources are the same across isotopes.

Alternative hypothesis: The fish from the four sources differ in isotopes.

```
library(car)

lm.man <- lm(cbind(Ca43, Mg24, Mg25, Mg26, Sr86, Sr87, Sr88, Ba137lg, Ba138lg)
             ~ Source, data = train.dat)
man.train <- Manova(lm.man, test = "Hotelling-Lawley")
man.train
```

```
Type II MANOVA Tests: Hotelling-Lawley test statistic
       Df test stat approx F num Df den Df    Pr(>F)
Source  3    29.371   595.77     27   1643 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Results

The results from the Hotelling-Lawley test show that we have sufficient evidence to reject the null hypothesis and conclude that at least one of the sources differs from the other.

3.1.2 Multiple Comparisons (Between Sources)

Test Hypothesis

$$H_0 : \mu_i = \mu_j \text{ versus } H_\alpha : \mu_i \neq \mu_j$$

Null hypothesis: mean of `source i` is equal to the mean of `source j`.

Alternative hypothesis: mean of `source i` is not equal to the mean of `source j`.

This test is repeated for all sources so that the mean of each source has been tested against each of the others.

```
library(car)
# Multivariate MANOVA test

Source.list <- sort(unique(train.dat$Source))
```

```r
for (i in 1:(length(Source.list) - 1)) {
  for (j in (i + 1):length(Source.list)) {
    # print a header to indicate which comparisons are being made
    cat("\n\n")
    cat(paste("***** Comparison between", i, Source.list[i], "and", j, Source.list[j]))

    # perform pairwise comparison

    man.pair <- Manova(lm(cbind(Ba137, Ba138, Ca43, Mg24, Mg25, Mg26, Sr86, Sr87, Sr88)
        ~ Source, data = subset(train.dat, (Source %in% Source.list[c(i, j)]))
          ))
    # print result
    print(man.pair)
  }
}
```

```
***** Comparison between 1 DEX and 2 GJH
Type II MANOVA Tests: Pillai test statistic
       Df test stat approx F num Df den Df    Pr(>F)
Source  1   0.93967   496.67      9    287 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


***** Comparison between 1 DEX and 3 NAP
Type II MANOVA Tests: Pillai test statistic
       Df test stat approx F num Df den Df    Pr(>F)
Source  1   0.91252   279.31      9    241 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


***** Comparison between 1 DEX and 4 SJR
Type II MANOVA Tests: Pillai test statistic
       Df test stat approx F num Df den Df    Pr(>F)
Source  1   0.94984   667.04      9    317 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


***** Comparison between 2 GJH and 3 NAP
Type II MANOVA Tests: Pillai test statistic
       Df test stat approx F num Df den Df    Pr(>F)
Source  1   0.72957   67.144      9    224 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


***** Comparison between 2 GJH and 4 SJR
Type II MANOVA Tests: Pillai test statistic
       Df test stat approx F num Df den Df    Pr(>F)
Source  1   0.81383   145.72      9    300 < 2.2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


***** Comparison between 3 NAP and 4 SJR
Type II MANOVA Tests: Pillai test statistic
       Df test stat approx F num Df den Df Pr(>F)
Source  1  0.034732   1.0155      9    254 0.4279
```

Results

At $\alpha = .05$ reject H_0 and conclude the means between the following pairs are not equal: DEX and GJH DEX and NAP DEX and SJR GJH and NAP and GJH and SJR

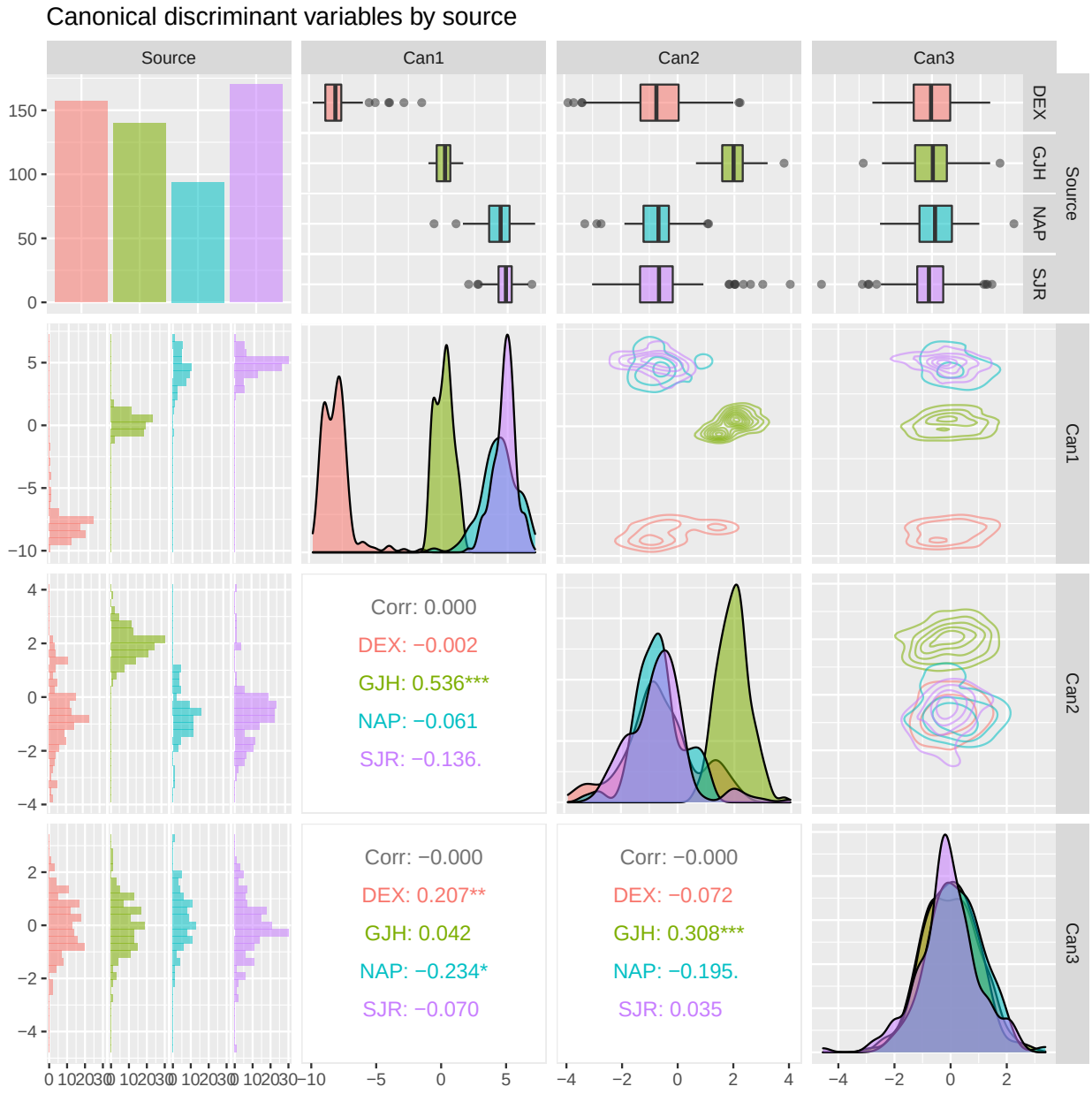
Fail to reject the null and conclude that the means of NAP and SJR are equal.

3.2 Canonical Discriminant Functions (Further Visualizations)

The canonical discriminant analysis will indicate the directions that provide the greatest ability to distinguish between the groups.

```
library(candisc)

can.train <- candisc(lm.man)
summary(can.train)
```

```
Canonical Discriminant Analysis for Source:

    CanRsq Eigenvalue Difference   Percent Cumulative
1 0.965606  28.075036      26.79 95.586725     95.587
2 0.562430   1.285351      26.79  4.376217     99.963
3 0.010767   0.010884      26.79  0.037058    100.000

Class means:

        Can1     Can2        Can3
DEX -7.96609 -0.61408 -0.00361022
GJH  0.20224  1.95851 -0.00060853
NAP  4.37741 -0.66474  0.20619880
SJR  4.76992 -0.67820 -0.11018052

 std coefficients:
              Can1     Can2     Can3
Ca43    -0.0072060  0.16324 -0.13670
Mg24    -0.0630143 -0.12663  3.79420
Mg25    -0.0048806 -0.40953 -4.06915
Mg26    -0.0527532  0.24634  0.40799
Sr86    -0.1542524 -0.45796 -1.78681
Sr87     0.0544305 -0.13012  1.29066
Sr88    -0.5905329 -0.19757  0.61299
Ba137lg  0.2725186 -2.48892 -4.68979
Ba138lg  0.6148864  2.00359  4.74071
```



Results

The discriminant variables all except $Can3$ show some sort of difference between the sources because of their distributions but $Can1$ shows the clearest differences and similarities making it the most useful.

3.3 MANOVA Final Results

Most of the sources can be differentiated by the mean of the isotopes. Razorback suckers that lose their tags can be traced to their source by analyzing the elemental isotopes in the fish.

4 Cluster Analysis

Goal is to find a clustering method that creates a dendrogram that seems to have a relatively small number of clusters that are different between clusters but similar within clusters.

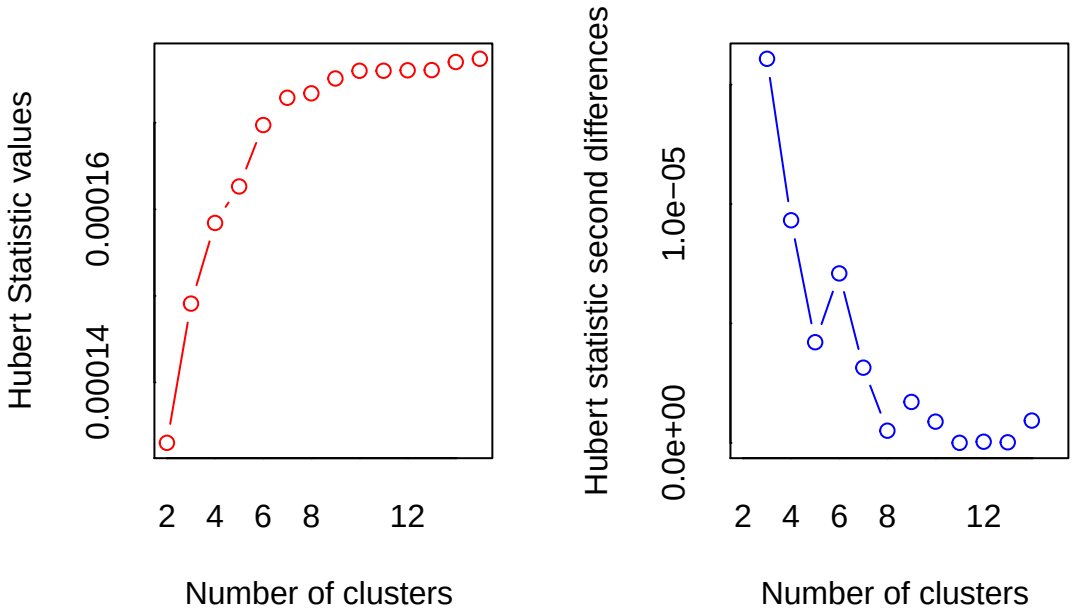
4.0.1 Linkage Method Selection

This section is unnecessary in understanding the research question. The cost to run this code is more expensive than its benefit and it's included here for future analysis and ease of examining the efficacy of other linkage methods.

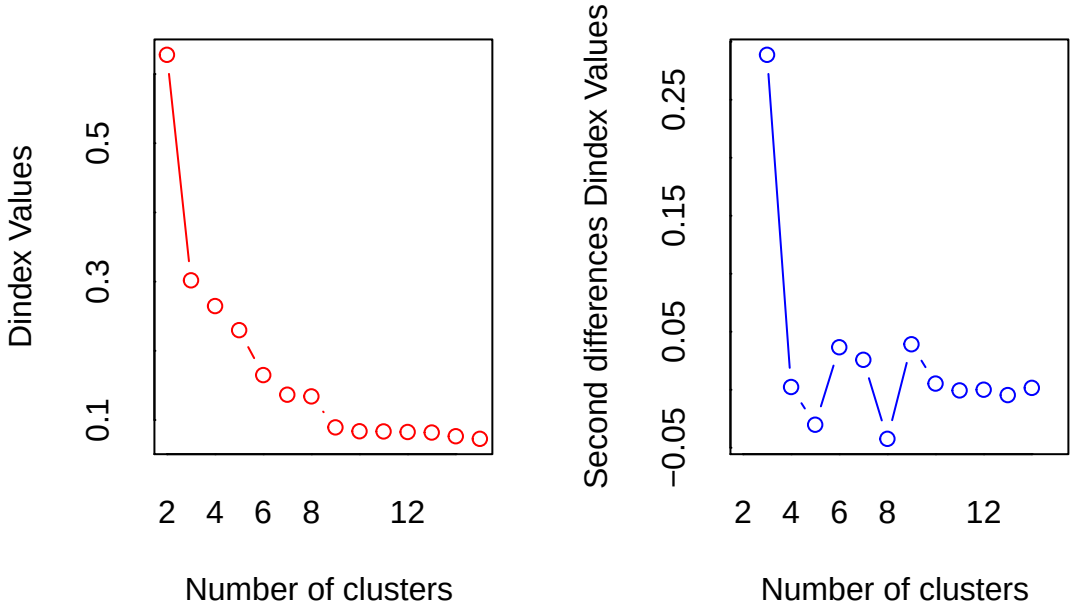
```
# eval=FALSE so this chunk doesn't evaluate

wd.clust <- NbClust(train.num, method = "ward.D", index = "all")
wd.clust$Best.nc

wd2.clust <- NbClust(train.num, method = "ward.D2", index = "all")
wd2.clust$Best.nc

sgl.clust <- NbClust(train.num, method = "single", index = "all")
sgl.clust$Best.nc

com.clust <- NbClust(train.num, method = "complete", index = "all")
com.clust$Best.nc

aveclust <- NbClust(train.num, method = "average", index = "all")
aveclust$Best.nc

mcq.clust <- NbClust(train.num, method = "mcquitty", index = "all")
mcq.clust$Best.nc

med.clust <- NbClust(train.num, method = "median", index = "all")
med.clust$Best.nc

cen.clust <- NbClust(train.num, method = "centroid", index = "all")
cen.clust$Best.nc
```

4.1 Average Linkage Method

```
library(NbClust)

# Removing factor variables
train.sub <- subset(train.dat, select = iso.vars)
train.num <- as.numeric(as.matrix(train.sub))

aveclust <- NbClust(train.num, method = "average", index = "all")
```



*** : The Hubert index is a graphical method of determining the number of clusters.
In the plot of Hubert index, we seek a significant knee that corresponds to a significant increase of the value of the measure i.e the significant peak in Hubert index second differences plot.




```
*** : The D index is a graphical method of determining the number of clusters.
               In the plot of D index, we seek a significant knee (the significant peak in Dindex
               second differences plot) that corresponds to a significant increase of the value of
               the measure.

*******************************************************************
* Among all indices:
* 2 proposed 3 as the best number of clusters
* 1 proposed 4 as the best number of clusters
* 1 proposed 8 as the best number of clusters
* 1 proposed 9 as the best number of clusters
* 1 proposed 14 as the best number of clusters

                   ***** Conclusion *****

* According to the majority rule, the best number of clusters is  3


*******************************************************************
```

```
aveclust$Best.nc
```

```
                     KL       CH Hartigan     CCC    Scott  Marriot TrCovW
Number_clusters  4.0000    14.00    8.000  9.0000    3.000    3.000   -Inf
Value_Index     12.0326 51807.33 7606.475 35.4504 6057.105 6155.765      3
                  TraceW Friedman    Rubin Cindex     DB Silhouette  Duda
Number_clusters 1799.481  91.1682 -59.5124 0.1164 0.3697     0.7635 0.542
Value_Index        9.000   9.0000   3.0000 7.0000 3.0000     3.0000 3.000
```

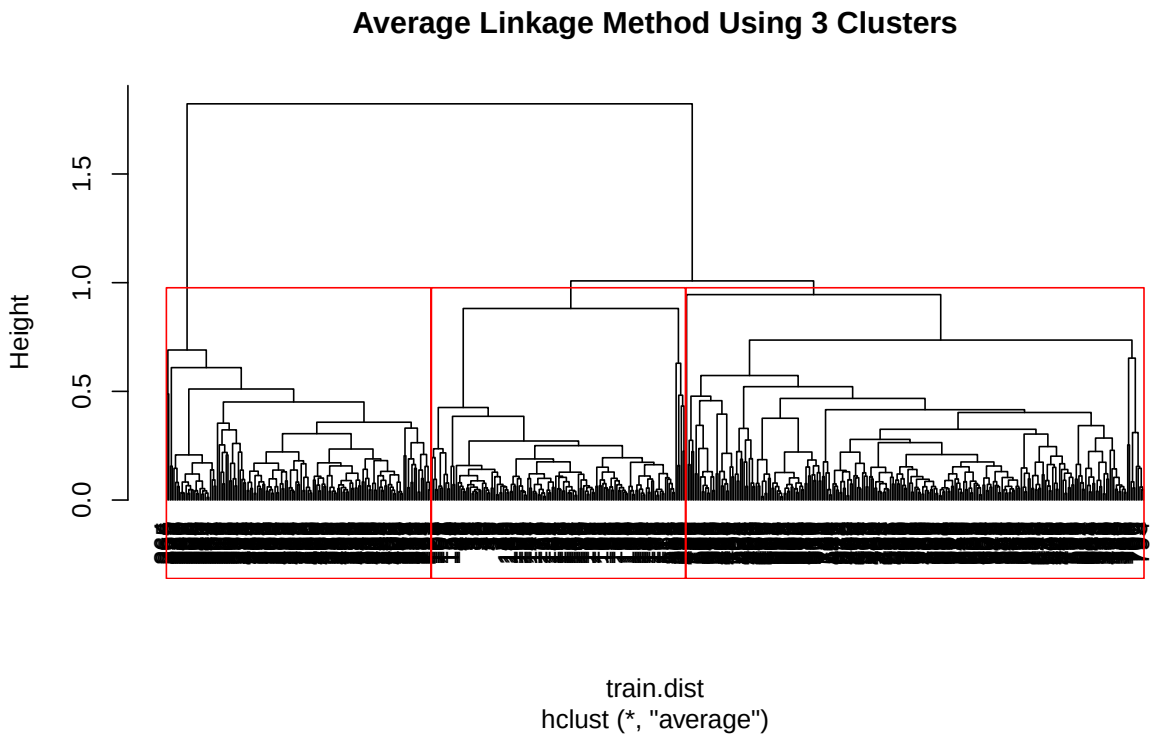
```
                PseudoT2  Beale Ratkowsky    Ball PtBiserial   Frey McClain
Number_clusters 946.3429 1.8413    0.5955 1114.07     0.7787 1.2412  0.1383
Value_Index       2.0000 2.0000    3.0000    2.00     2.0000 2.0000  2.0000
                  Dunn Hubert SDindex Dindex  SDbw
Number_clusters 0.3414      0  2.4739      0 0.025
Value_Index     0.0000      4  0.0000      9 4.000
```

4.2 Plotting

4.2.1 Dendrogram

```
# create distance matrix between points
train.dist <- dist(train.sub)
# create dendrogram
hc.complete  <- hclust(train.dist, method = "average")

# create a column with group membership
train.dat$cut.comp <- factor(cutree(hc.complete, k = 3))
```

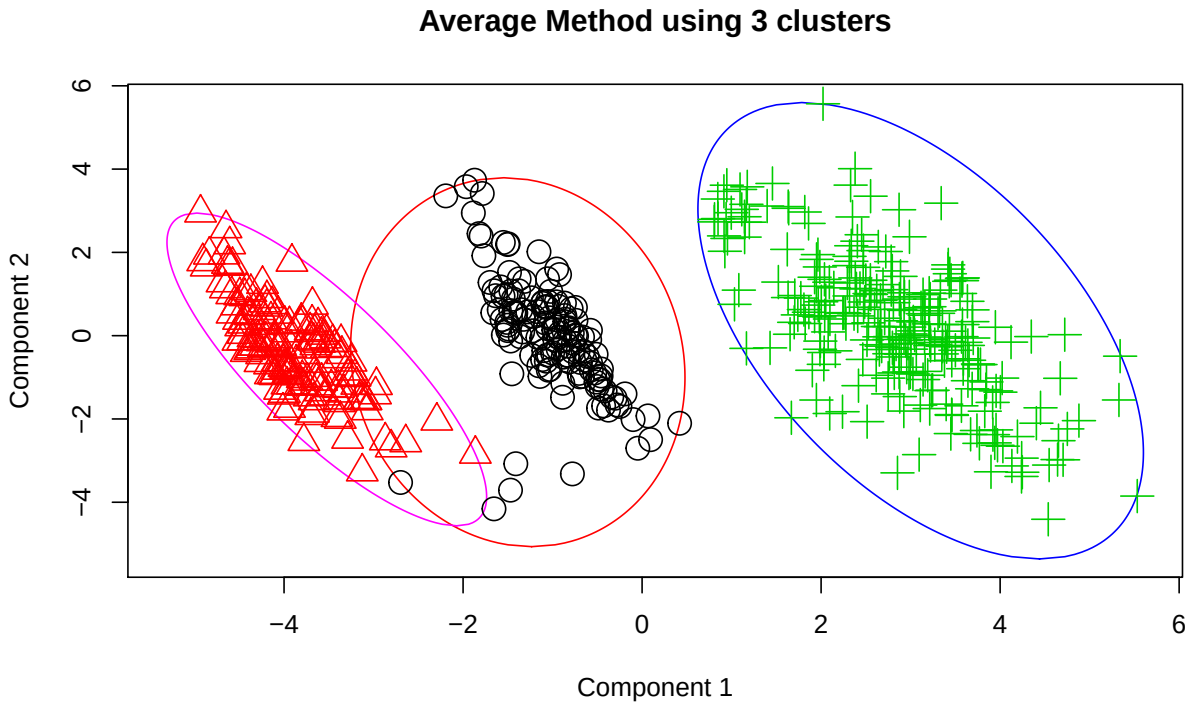
```
plot(hc.complete
     , hang = -1
     , main = "Average Linkage Method Using 3 Clusters"
#     , labels = train.sub[,1]
     )
rect.hclust(hc.complete, k = 3)
```



The density of the dendrogram makes this plot hard to understand. Plotting against principal components gives a better visual understanding of the classifications.

4.2.2 PCA

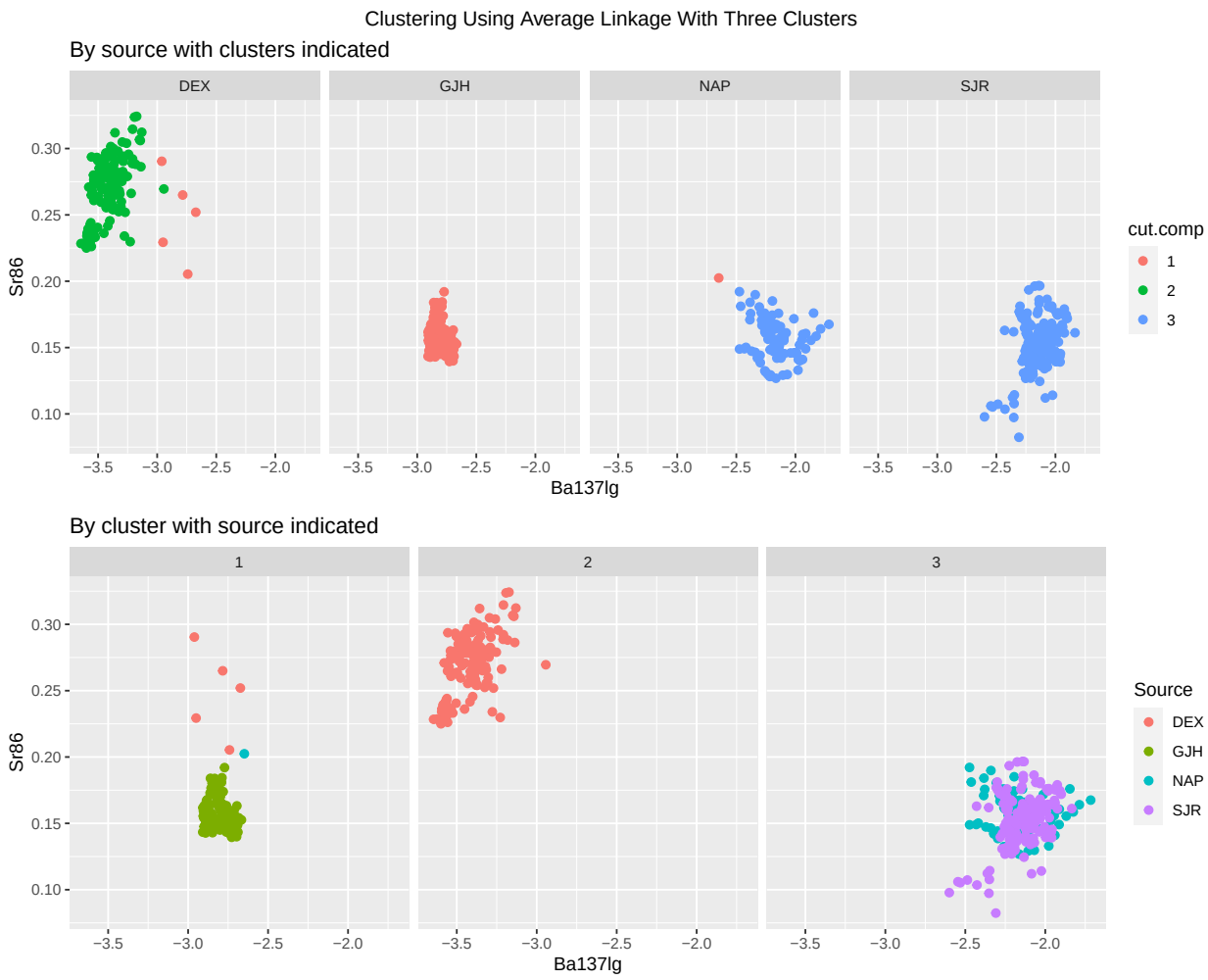
```
library(cluster)

clusplot(train.dat, cutree(hc.complete, k = 3)
        , color = TRUE, labels = 0, lines = 0
        , cex = 2, cex.txt = 1, col.txt = "gray20"
        , main = "Average Method using 3 clusters"
        , sub = NULL
        , col.p = train.dat$cut.comp)
```



4.3 Cluster Analysis Results

4.3.1 Visualizing Efficacy

Efficacy By Source



Efficacy By Source Type

The plots show there would be clear errors in classification of the origin of the razorback suckers particularly when trying to differentiate fish sourced from NAPI ponds and the San Juan River. This analysis is not unexpected given the results of the MANOVA analysis.

5 Prediction

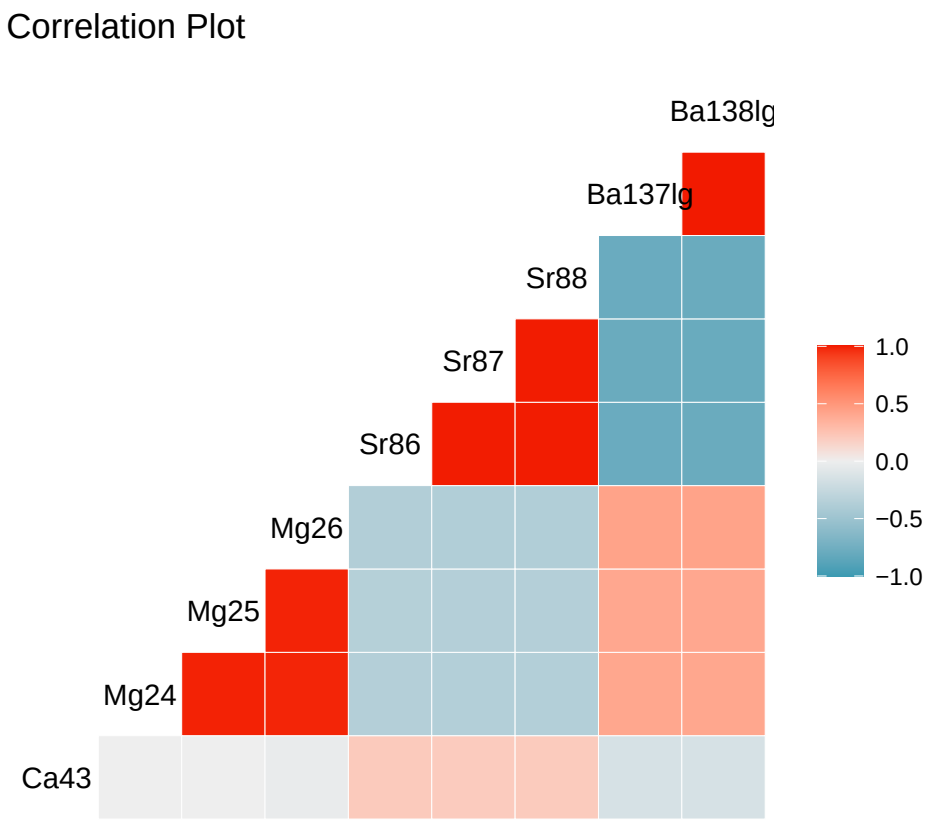
5.1 Principal Components Analysis (PCA)

The model can be simplified without sacrificing information gain by taking a principal component of correlated isotopes.

The correlation can be seen on previous pairs plots but will be shown again here.

```
library(ggplot2)
library(GGally)

train.x <- subset(train.dat, select = iso.vars)

p <- ggcorr(train.x, method = c("everything", "pearson"))
p <- p + labs(title = "Correlation Plot")
```

```
print(p)
```



Barium Isotope Variables:

```
pca.Ba <- princomp( ~ Ba137lg + Ba138lg, data = train.dat, cor = FALSE)
train.dat$PC1.Ba <- pca.Ba$scores[,"Comp.1"]

summary(pca.Ba)
```

```
Importance of components:
                          Comp.1       Comp.2
Standard deviation     0.7684660 0.0099557874
Proportion of Variance 0.9998322 0.0001678143
Cumulative Proportion  0.9998322 1.0000000000
```

```
print(loadings(pca.Ba), cutoff = 0)
```

```
Loadings:
        Comp.1 Comp.2
Ba137lg  0.711  0.703
```

```
Ba138lg  0.703 -0.711

               Comp.1 Comp.2
SS loadings       1.0    1.0
Proportion Var    0.5    0.5
Cumulative Var    0.5    1.0
```

```
# Adding the PC to the test dataset
pca.Ba <- princomp( ~ Ba137lg + Ba138lg, data = test.dat, cor = FALSE)
test.dat$PC1.Ba <- pca.Ba$scores[,"Comp.1"]
```

`Comp.1` explains 99.985% of the variability of both the Barium isotopes.

Calcium Isotope Variables:

```
pca.Ca <- princomp( ~ Ca43, data = train.dat, cor = FALSE)
train.dat$PC1.Ca <- pca.Ca$scores[,"Comp.1"]

summary(pca.Ca)
```

```
Importance of components:
                          Comp.1
Standard deviation     0.01271343
Proportion of Variance 1.00000000
Cumulative Proportion  1.00000000
```

```
print(loadings(pca.Ca), cutoff = 0)
```

```
Loadings:
     Comp.1
Ca43 1

               Comp.1
SS loadings         1
Proportion Var      1
```

`Comp.1` explains 100% of the variability of the Calcium isotope.

Magnesium Isotope Variables:

```
pca.Mg <- princomp( ~ Mg24 + Mg25 + Mg26, data = train.dat, cor = FALSE)
summary(pca.Mg)
```

```
Importance of components:
                          Comp.1      Comp.2       Comp.3
Standard deviation     0.2208144 0.007054840 0.0036365844
Proportion of Variance 0.9987097 0.001019433 0.0002708768
Cumulative Proportion  0.9987097 0.999729123 1.0000000000
```

```
print(loadings(pca.Mg), cutoff = 0)
```

```
Loadings:
     Comp.1 Comp.2 Comp.3
Mg24  0.982  0.185  0.043
Mg25  0.126 -0.464 -0.877
Mg26  0.142 -0.866  0.479

               Comp.1 Comp.2 Comp.3
SS loadings     1.000  1.000  1.000
Proportion Var  0.333  0.333  0.333
Cumulative Var  0.333  0.667  1.000
```

```
train.dat$PC1.Mg <- pca.Mg$scores[,"Comp.1"]
```

Note that `Comp.1` explains 99.871% of the variability of all three of the Magnesium isotopes.

Strontium Isotope Variables:

```
pca.Sr <- princomp( ~ Sr86 + Sr87 + Sr88, data = train.dat, cor = FALSE)
summary(pca.Sr)
```

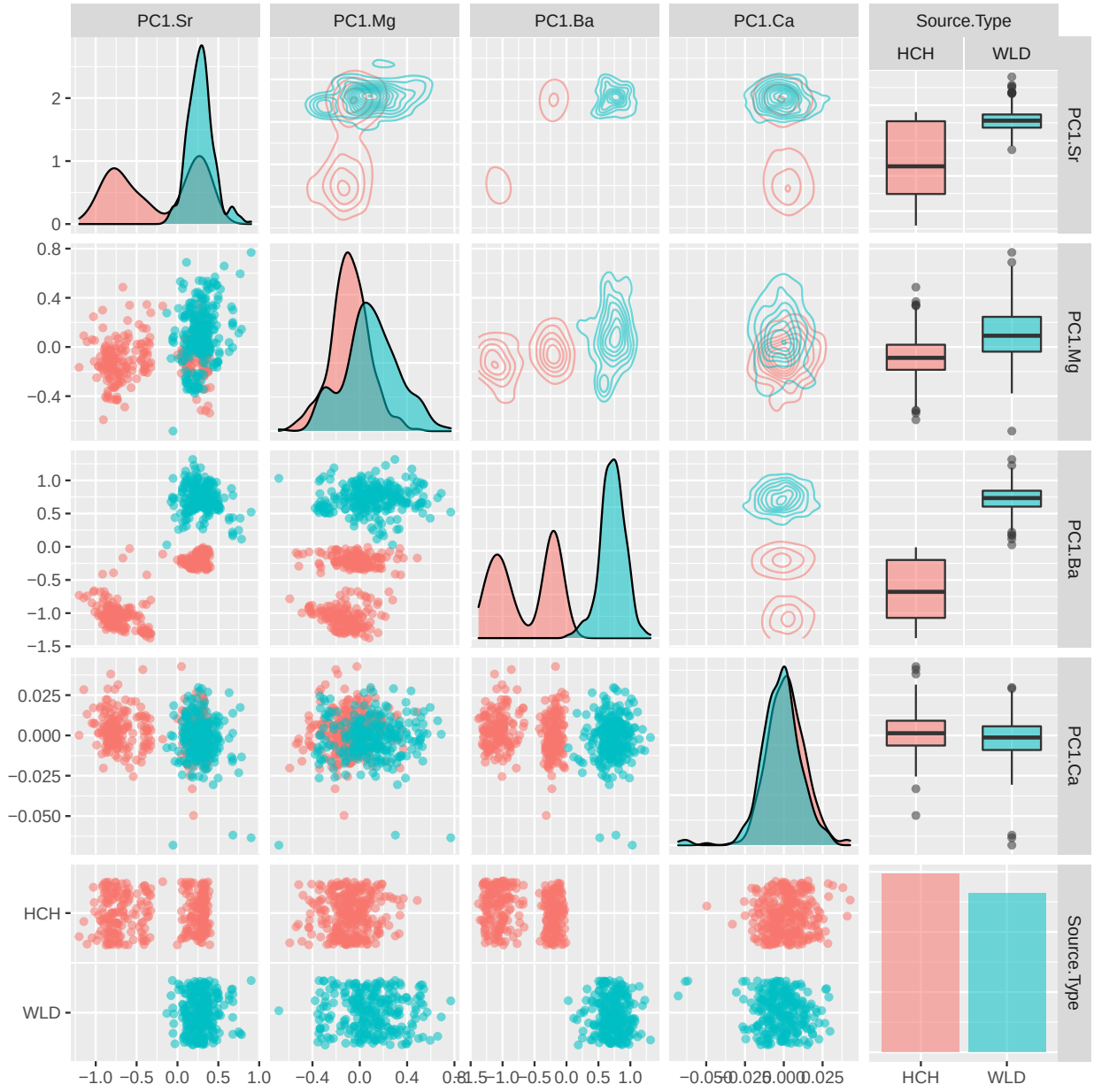
```
Importance of components:
                          Comp.1       Comp.2       Comp.3
Standard deviation     0.4668990 4.585043e-03 2.382171e-03
Proportion of Variance 0.9998775 9.642458e-05 2.602836e-05
Cumulative Proportion  0.9998775 9.999740e-01 1.000000e+00
```

```
print(loadings(pca.Sr), cutoff = 0)
```

```
Loadings:
     Comp.1 Comp.2 Comp.3
Sr86  0.117  0.891  0.439
Sr87  0.083  0.432 -0.898
Sr88  0.990 -0.142  0.023

               Comp.1 Comp.2 Comp.3
SS loadings     1.000  1.000  1.000
Proportion Var  0.333  0.333  0.333
Cumulative Var  0.333  0.667  1.000
```

```
train.dat$PC1.Sr <- -pca.Sr$scores[,"Comp.1"]
```

Note that `Comp.1` explains 99.988% of the variability of all three of the Strontium isotopes.




```
# Adding the PC to the test dataset
pca.Ba <- princomp( ~ Ba137lg + Ba138lg, data = test.dat, cor = FALSE)
test.dat$PC1.Ba <- pca.Ba$scores[,"Comp.1"]

pca.Ca <- princomp( ~ Ca43, data = test.dat, cor = FALSE)
test.dat$PC1.Ca <- pca.Ca$scores[,"Comp.1"]

pca.Mg <- princomp( ~ Mg24 + Mg25 + Mg26, data = test.dat, cor = FALSE)
test.dat$PC1.Mg <- pca.Mg$scores[,"Comp.1"]

pca.Sr <- princomp( ~ Sr86 + Sr87 + Sr88, data = test.dat, cor = FALSE)
test.dat$PC1.Sr <- -pca.Sr$scores[,"Comp.1"]
```

5.2 Regression Modeling

From visual inspection predicting hatchery and wild suckers will not be difficult so, both a binomial regression (fitted against `Source.Type`), and a multinomial regression (fitted on `Source`) will be done.

5.2.1 Binomial Model

```
library(nnet)

Bi.fit <- multinom(Source.Type ~ PC1.Ba + PC1.Ca + PC1.Mg + PC1.Sr, data = train.dat)
```

```
# weights:  6 (5 variable)
initial  value 388.855568
iter  10 value 17.078687
iter  20 value 0.017838
iter  30 value 0.005042
iter  40 value 0.004593
iter  50 value 0.004339
iter  60 value 0.004294
iter  70 value 0.004124
iter  80 value 0.003981
iter  90 value 0.003890
iter 100 value 0.003757
final  value 0.003757
stopped after 100 iterations
```

```
summary(Bi.fit)
```

```
Call:
multinom(formula = Source.Type ~ PC1.Ba + PC1.Ca + PC1.Mg + PC1.Sr,
    data = train.dat)

Coefficients:
               Values Std. Err.
(Intercept) -13.111890  92.53301
PC1.Ba      200.632991 713.88451
PC1.Ca       -5.045153  43.16170
PC1.Mg       55.359774 441.69477
PC1.Sr       12.432423 137.49361

Residual Deviance: 0.007514348
AIC: 10.00751
```

```
# Predicting the values for train dataset
test.dat$BiPred <- predict(Bi.fit, newdata = test.dat, "class")

# Building classification table
tab <- table(test.dat$Source.Type, test.dat$BiPred)
tab
```

```
     HCH WLD
  HCH 126   0
  UNK   0   0
  WLD   0 111
```

```
# Calculating accuracy - sum of diagonal elements divided by total obs
round((sum(diag(tab))/sum(tab))*100,2)
```

```
[1] 53.16
```

```
Bi.fit.red.AIC <- step(Bi.fit, direction="both", trace = 0)
```

5.2.1.1 Binomial Model Reduction

```
trying - PC1.Ba
trying - PC1.Ca
trying - PC1.Mg
trying - PC1.Sr
# weights:  5 (4 variable)
initial  value 388.855568
iter  10 value 17.190095
iter  20 value 0.005396
iter  30 value 0.004494
iter  40 value 0.004392
iter  50 value 0.004289
iter  60 value 0.004209
iter  70 value 0.004127
iter  80 value 0.004050
iter  90 value 0.003969
iter 100 value 0.003881
final  value 0.003881
stopped after 100 iterations
trying - PC1.Ba
trying - PC1.Mg
trying - PC1.Sr
trying + PC1.Ca
# weights:  4 (3 variable)
initial  value 388.855568
iter  10 value 0.025168
iter  20 value 0.020912
iter  30 value 0.019277
iter  40 value 0.018115
iter  50 value 0.016344
iter  60 value 0.014859
iter  70 value 0.014543
iter  80 value 0.013495
iter  90 value 0.012369
iter 100 value 0.011794
final  value 0.011794
```

```
stopped after 100 iterations
trying - PC1.Ba
trying - PC1.Mg
trying + PC1.Ca
trying + PC1.Sr
# weights:  3 (2 variable)
initial  value 388.855568
iter  10 value 0.061361
iter  20 value 0.059956
iter  30 value 0.058481
iter  40 value 0.057067
iter  50 value 0.055711
iter  60 value 0.054413
iter  70 value 0.053173
iter  80 value 0.051985
iter  90 value 0.050841
iter 100 value 0.049739
final  value 0.049739
stopped after 100 iterations
trying - PC1.Ba
trying + PC1.Ca
trying + PC1.Mg
trying + PC1.Sr
```

```
Bi.fit.red.AIC$anova
```

```
      Step Df     Deviance Resid. Df  Resid. Dev       AIC
1          NA           NA       556 0.007514348 10.007514
2 - PC1.Ca  1 0.0002467964       557 0.007761144  8.007761
3 - PC1.Sr  1 0.0158263288       558 0.023587473  6.023587
4 - PC1.Mg  1 0.0758910026       559 0.099478475  4.099478
```

```
summary(Bi.fit.red.AIC)
```

```
Call:
multinom(formula = Source.Type ~ PC1.Ba, data = train.dat)

Coefficients:
               Values  Std. Err.
(Intercept)  -2.074348   5.135809
PC1.Ba      202.565863 237.226327

Residual Deviance: 0.09947848
AIC: 4.099478
```

```
# Predicting the values for train dataset
test.dat$BiPred <- predict(Bi.fit.red.AIC, newdata = test.dat, "class")

# Building classification table
tab <- table(test.dat$Source.Type, test.dat$BiPred)
tab
```

```
     HCH WLD
  HCH 126   0
  UNK   0   0
  WLD   0 111
```

```
# Calculating accuracy - sum of diagonal elements divided by total obs
round((sum(diag(tab))/sum(tab))*100,2)
```

```
[1] 53.16
```

As expected the model accuracy in predicting `Source.Type` is 100%.

5.2.2 Multinomial Model

```
library(nnet)

mult.fit <- multinom(Source ~ PC1.Ba + PC1.Ca + PC1.Mg + PC1.Sr, data = train.dat)
```

```
# weights:  24 (15 variable)
initial  value 777.711137
iter  10 value 196.398870
iter  20 value 170.849378
iter  30 value 167.851251
iter  40 value 167.576823
iter  50 value 167.526809
iter  60 value 167.496035
final  value 167.493631
converged
```

```
summary(mult.fit)
```

```
Call:
multinom(formula = Source ~ PC1.Ba + PC1.Ca + PC1.Mg + PC1.Sr,
    data = train.dat)

Coefficients:
    (Intercept)   PC1.Ba     PC1.Ca    PC1.Mg    PC1.Sr
GJH    27.09645 13.43844  25.489710 -45.45781 103.81405
NAP    22.95161 81.18845 -13.975637 -23.96590  99.40827
SJR    21.96338 82.72289  -9.464595 -24.38296 101.30350

Std. Errors:
    (Intercept)   PC1.Ba   PC1.Ca   PC1.Mg    PC1.Sr
GJH    22.08520 55.53996 8.857935 47.69593 14.697525
NAP    18.58972 27.59492 6.740112 24.07147  7.331883
SJR    18.59439 27.59573 6.756452 24.07298  7.338307

Residual Deviance: 334.9873
AIC: 364.9873
```

```r
# Predicting the values for train dataset
test.dat$MultPred <- predict(mult.fit, newdata = test.dat, "class")

# Building classification table
tab <- table(test.dat$Source, test.dat$MultPred)
tab
```

```
     DEX GJH NAP SJR
  DEX  67   0   0   0
  GJH   0  59   0   0
  NAP   0   0   6  33
  SJR   0   1   1  70
  UNK   0   0   0   0
```

```r
# Calculating accuracy - sum of diagonal elements divided by total obs
round((sum(diag(tab))/sum(tab))*100,2)
```

```
[1] 85.23
```

```r
mult.fit.red.AIC <- step(mult.fit, direction="both", trace = 0)
```

5.2.2.1 Multinomial Model Reduction

```
trying - PC1.Ba
trying - PC1.Ca
trying - PC1.Mg
trying - PC1.Sr
# weights:  20 (12 variable)
initial  value 777.711137
iter  10 value 196.434864
iter  20 value 170.994293
iter  30 value 167.642852
iter  40 value 167.604238
iter  50 value 167.586596
iter  60 value 167.579521
iter  70 value 167.575446
final  value 167.575236
converged
trying - PC1.Ba
trying - PC1.Mg
trying - PC1.Sr
trying + PC1.Ca
# weights:  16 (9 variable)
initial  value 777.711137
iter  10 value 196.431288
iter  20 value 170.550077
iter  30 value 169.114702
iter  40 value 168.847265
```

```
iter  50 value 168.674754
iter  60 value 168.313679
iter  70 value 168.073719
iter  80 value 167.830693
iter  90 value 167.798635
final  value 167.798567
converged
trying - PC1.Ba
trying - PC1.Sr
trying + PC1.Ca
trying + PC1.Mg
```

```
mult.fit.red.AIC$anova
```

```
      Step Df  Deviance Resid. Df Resid. Dev      AIC
1          NA        NA       546   334.9873 364.9873
2 - PC1.Ca  3 0.1632080       549   335.1505 359.1505
3 - PC1.Mg  3 0.4466629       552   335.5971 353.5971
```

```
summary(mult.fit.red.AIC)
```

```
Call:
multinom(formula = Source ~ PC1.Ba + PC1.Sr, data = train.dat)

Coefficients:
    (Intercept)     PC1.Ba   PC1.Sr
GJH    10.59015   1.763627 95.56793
NAP    11.36755 162.162768 77.74643
SJR    10.40688 163.672970 79.43452

Std. Errors:
    (Intercept)   PC1.Ba   PC1.Sr
GJH    42.52134 432.4700 82.88347
NAP    22.10014 336.1247 90.01788
SJR    22.10400 336.1249 90.01694

Residual Deviance: 335.5971
AIC: 353.5971
```

```
# Predicting the values for train dataset
test.dat$MultPred <- predict(mult.fit.red.AIC, newdata = test.dat, "class")

# Building classification table
tab <- table(test.dat$Source, test.dat$MultPred)
#tab

# Calculating accuracy - sum of diagonal elements divided by total obs
round((sum(diag(tab))/sum(tab))*100,2)
```

```
[1] 84.39
```

The final multinomial model contains `PC1.Ba` and `PC1.Sr` as predictor variables and has an accuracy rate of 84.39%. The error is in predicting suckers originating from NAPI ponds and the San Juan River which is expected based on previous analysis.

5.3 Model Efficacy

```
library(caret)

conf.mat <- confusionMatrix(test.dat$MultPred, test.dat$Source)
conf.mat
```

```
Confusion Matrix and Statistics

          Reference
Prediction DEX GJH NAP SJR UNK
       DEX  67   0   0   0   0
       GJH   0  59   0   0   0
       NAP   0   0   4   2   0
       SJR   0   0  35  70   0
       UNK   0   0   0   0   0

Overall Statistics

               Accuracy : 0.8439
                 95% CI : (0.7913, 0.8876)
    No Information Rate : 0.3038
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.783

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: DEX Class: GJH Class: NAP Class: SJR Class: UNK
Sensitivity              1.0000     1.0000    0.10256     0.9722         NA
Specificity              1.0000     1.0000    0.98990     0.7879          1
Pos Pred Value           1.0000     1.0000    0.66667     0.6667         NA
Neg Pred Value           1.0000     1.0000    0.84848     0.9848         NA
Prevalence               0.2827     0.2489    0.16456     0.3038          0
Detection Rate           0.2827     0.2489    0.01688     0.2954          0
Detection Prevalence     0.2827     0.2489    0.02532     0.4430          0
Balanced Accuracy        1.0000     1.0000    0.54623     0.8801         NA
```

6 Conclusion

Wild razorback suckers and those from hatcheries can be differentiated by their elemental isotopic ratios and their origin can be predicted to 100% accuracy. Determining the specific source of the suckers is more difficult, with a final model accuracy of 84.39%. Fish from the Dexter National Fish Hatchery and Ouray National Fish Hatchery, Grand Valley Unit are easily differentiated from each other and those from the NAPI ponds and San Juan River, however, differentiating between fish from NAPI ponds and the San Juan River is not possible. The elemental isotopic ratio that has the most weight in classification of origin are the Barium isotopes, which was shown to be significant in classifying both hatchery vs wild fish and fish by their source.

Similarities in elemental isotopic ratios can be explained by proximity:

```
Ouray National Fish Hatchery: Northeaster Utah
Dexter National Fish Hatchery: Southeastern New Mexico
NAPI ponds: Northwestern New Mexico
San Juan River: spans from Southern Utah to Northern New Mexico
```

NAPI ponds and the San Juan River at a point are within 4 miles of each other while they are approximately 300 miles away from the Ouray and Dexter National Fish Hatcheries, which are 900 miles away from each other. Similarities elemental isotopic ratios can be explained by proximity.

Overall, tracking hatchery sourced razorback suckers is possible for biologists and conservationists to do and I assume it would be possible for other fish that are sourced from hatcheries that are not in close proximity of the natural populations they supplement.

7 Additional Reading

- Navajo Agricultural Products Industry (NAPI) 2014 report
- Center for Biological Diversity species notes
- The Southwestern Native Aquatic Resources & Recovery Center (formerly the Dexter National Fish Hatchery & Technology Center) reference page
- The Colorado River Fishery Project page