NAUFWP Faculty Benchmarking Analysis  
Kenneth F. Kellner and Robert K. Swihart

Introduction

This document provides instructions for calculation of standardized deviance residuals for the original data presented in the PLoS One paper by Swihart et al., with the option of adding new individual(s) to benchmark.

Data Input

First, set the working directory to the location of the full faculty database (‘S1_Data.csv’) and make sure the required R packages are installed and loaded. If any of the packages are missing, they will be downloaded and installed automatically.

```r
setwd(choose.dir())
if(!'MASS' %in% installed.packages()[,'Package']){install.packages('MASS')}
if(!'aplpack' %in% installed.packages()[,'Package']){install.packages('aplpack')}
require(MASS); require(aplpack)
```

Next, read in the database, and inspect the result in R:

```r
alldata <- read.csv('S1_Data.csv',header=TRUE)
```

The database contains a single row for each scientist. Columns 1-11 are the independent variables:

```
head(alldata[,1:11])
```

<table>
<thead>
<tr>
<th>age.phd</th>
<th>Male</th>
<th>Re</th>
<th>health</th>
<th>genetics</th>
<th>social</th>
<th>applied</th>
<th>ecology</th>
<th>quant</th>
<th>conservation</th>
<th>aquaticsci</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>41</td>
<td>1</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>1</td>
<td>50</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1</td>
<td>65</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>19</td>
<td>0</td>
<td>50</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td>70</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

1. **age.phd**: Academic age, i.e., the number of years since conferral of the Ph.D.
2. **Male**: Sex (0 = female, 1 = male)
3. **Re**: Percent of appointment allocated to research (0-100)
4. **health**: Disease-related subdiscipline? 0 if no, 1 if yes
5. **genetics**: Genetics subdiscipline? 0 if no, 1 if yes
6. **social**: Social sciences? 0 if no, 1 if yes
7. **applied**: Management-related subdiscipline? 0 if no, 1 if yes
8. **ecology**: Ecology subdiscipline? 0 if no, 1 if yes
9. **quant**: Quantitative subdiscipline? 0 if no, 1 if yes
10. conservation: Conservation-related subdiscipline? 0 if no, 1 if yes
11. aquaticsci: Aquatic sciences subdiscipline? 0 if no, 1 if yes

Columns 12-19 are response variables:

```r
head(alldata[,12:19])
```

<table>
<thead>
<tr>
<th></th>
<th>nPubs</th>
<th>h</th>
<th>mqPhD</th>
<th>hb</th>
<th>citesperyPhD</th>
<th>nCites</th>
<th>mindex</th>
<th>rindex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>39</td>
<td>8</td>
<td>0.195</td>
<td>18.392</td>
<td>4.195</td>
<td>172</td>
<td>11.0</td>
<td>10.536</td>
</tr>
<tr>
<td>2</td>
<td>34</td>
<td>9</td>
<td>0.750</td>
<td>20.091</td>
<td>17.000</td>
<td>204</td>
<td>12.0</td>
<td>12.000</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>8</td>
<td>2.000</td>
<td>22.526</td>
<td>68.750</td>
<td>275</td>
<td>27.5</td>
<td>15.811</td>
</tr>
<tr>
<td>4</td>
<td>17</td>
<td>7</td>
<td>1.750</td>
<td>16.695</td>
<td>35.750</td>
<td>143</td>
<td>21.0</td>
<td>11.358</td>
</tr>
<tr>
<td>5</td>
<td>42</td>
<td>15</td>
<td>0.789</td>
<td>34.799</td>
<td>32.474</td>
<td>617</td>
<td>37.0</td>
<td>23.388</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>1</td>
<td>0.167</td>
<td>4.162</td>
<td>1.833</td>
<td>11</td>
<td>8.0</td>
<td>2.828</td>
</tr>
</tbody>
</table>

12. npubs: Number of publications (based on Web of Science using all databases)
13. h: Hirsch’s h-index (the number of publications cited at least h times each)
14. mqPhD: m quotient, which is h-index divided by academic age. Note that we define academic age as the number of years since conferral of the Ph.D. degree, which differs from the age used by Hirsch.
15. hb: Brown’s hb-index (see manuscript for computation)
16. citesperyPhD: Citations per year since Ph.D. (i.e., number of citations/academic age)
17. nCites: Total number of citations (based on Web of Science using all databases)
18. m-index: The median number of citations for papers in the h-core
19. r-index: The square root of the sum of citations for papers in the h-core

Next, read in the data file containing information about the potential new faculty member(s) you wish to evaluate. This file should be nearly identical in structure to the database above (one row per scientist). The file ‘S2_Data.csv’ is provided as an example, containing three potential faculty members. This input file can potentially contain any number of individuals to evaluate. The rows of this file are named to uniquely identify the faculty candidates.

```r
newdata <- read.csv('S2_Data.csv',header=TRUE)
rownames(newdata) <- c('candidate.x','candidate.y','candidate.z')
```

In the provided file, candidate.x represents the hypothetical candidate discussed in the paper. This individual is a male with a 25% research appointment, 15 years post-PhD, who works on wildlife diseases and ecology by developing quantitative tools.

```r
newdata[,1:9]
```

<table>
<thead>
<tr>
<th></th>
<th>age.phd</th>
<th>Male</th>
<th>Re</th>
<th>health</th>
<th>genetics</th>
<th>social</th>
<th>applied</th>
<th>ecology</th>
<th>quant</th>
</tr>
</thead>
<tbody>
<tr>
<td>candiatex</td>
<td>15</td>
<td>1</td>
<td>25</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>candidate.y</td>
<td>15</td>
<td>0</td>
<td>60</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>candidate.z</td>
<td>20</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

candidate.x has the following values for the 8 response variables or metrics:

- Total publications = 30
- h-index = 10
- m-quotient (or rate of accumulation of h) is 10/15 = 0.67 from PhD date
- hb-index = 21
- Citations per year since PhD date = 18
- Total number of citations = 270
- m-index = 17 = median number of cites for papers in h-core
- r-index = 15 = square root of sum of citations in h-core

\[
\begin{array}{cccccccc}
\text{nPubs} & \text{h} & \text{mqPhD} & \text{hb} & \text{citesperyPhD} & \text{nCites} & \text{mindex} & \text{rindex} \\
\text{candidate.x} & 30 & 10 & 0.670 & 21.000 & 18.000 & 270 & 17 & 15.000 \\
\text{candidate.y} & 53 & 15 & 1.364 & 34.313 & 54.364 & 598 & 26 & 21.471 \\
\text{candidate.z} & 29 & 16 & 0.800 & 52.000 & 77.600 & 1552 & 50 & 38.756 \\
\end{array}
\]

**Fit Regression Models**

The faculty database is split into independent variables (\(X\)) and response variables (\(Y\)), and the appropriate information for the potential new faculty members is appended to \(X\) and \(Y\) to create \(X.new\) and \(Y.new\). These appended rows are named according to the provided ID values.

\[
X \leftarrow \text{alldata[,1:11]}
\]

\[
Y \leftarrow \text{alldata[,12:19]}
\]

\[
X.new \leftarrow \text{as.matrix(rbind(X,newdata[,1:11]))}
\]

\[
\text{row.names}(X.new) \leftarrow \text{c(1:dim(alldata)[1],as.character(newdata[,1])})
\]

\[
Y.new \leftarrow \text{as.matrix(rbind(Y,newdata[,12:19]))}
\]

\[
\text{row.names}(Y.new) \leftarrow \text{c(1:dim(alldata)[1],as.character(newdata[,1])})
\]

A quadratic term for \(age.phd\) is computed (\(age.phd\) is centered and squared):

\[
age.2n \leftarrow \text{scale(X.new[,1],center=TRUE, scale=FALSE)}
\]

\[
age.2n \leftarrow age.2n^2
\]

Finally, a series of unique generalized linear models (one per response variable or metric) are fit. Best fit models for each response variable were chosen as described in Swihart et al. (*in press*).

\[
\text{models} \leftarrow \text{list()}
\]

\[
\text{models$np.best} \leftarrow \text{glm.nb(Y.new[,1]-X.new[,1]+age.2n+X.new[,2:11])}
\]

\[
\text{models$h.best} \leftarrow \text{glm.nb(Y.new[,2]-X.new[,1]+age.2n+X.new[,3:11])}
\]

\[
\text{models$mq.best} \leftarrow \text{glm(Y.new[,3]-log(X.new[,1]+.5)+age.2n+X.new[,3:11])}
\]

\[
\text{models$hb.best} \leftarrow \text{glm.nb(round(Y.new[,4])-X.new[,1]+age.2n+X.new[,3:11])}
\]

\[
\text{models$c.best} \leftarrow \text{glm.nb(round(Y.new[,5])-X.new[,1]+X.new[,3:11])}
\]

\[
\text{models$nc.best} \leftarrow \text{glm.nb(Y.new[,6]-X.new[,1]+age.2n+X.new[,3:11])}
\]

\[
\text{models$m.best} \leftarrow \text{glm.nb(round(Y.new[,7])-X.new[,1]+age.2n+X.new[,3]+X.new[,4:11])}
\]

\[
\text{models$r.best} \leftarrow \text{glm.nb(round(Y.new[,8])-X.new[,1]+age.2n+X.new[,3]+X.new[,4:11])}
\]
Calculate Standardized Deviance Residuals

After all models are fit, standardized deviance residuals are calculated for each model and stored in the `stdres` object.

```r
stdres <- sapply(models,rstandard)
```

We then calculate a series of summary statistics (mean, range, and 95% confidence interval) across all 8 metrics for each of the faculty included. These values will be used later.

```r
r.mean <- rowMeans(stdres)
r.min <- apply(stdres,1,min)
r.max <- apply(stdres,1,max)
r.ci <- 1.96*apply(stdres,1,sd)/sqrt(dim(Y.new)[2])
```

A matrix `set` is defined to store quantile values for each faculty × metric combination for the full dataset. Summary statistics (mean, SD, and range) for each faculty calculated across the 8 metrics are stored in `qstat`.

```r
set <- matrix(data=NA,nrow=dim(Y.new)[1],ncol=dim(Y.new)[2])
for(i in 1:dim(Y.new)[1]){ for(j in 1:dim(Y.new)[2]){ set[i,j] <- sum(stdres[,j] <= stdres[i,j])/dim(Y.new)[1] }
}
qstat <- matrix(data=NA,nrow=dim(Y.new)[1],4)
for(i in 1:dim(Y.new)[1]){ qstat[i,1] <- mean(set[i,]) qstat[i,2] <- sd(set[i,]) qstat[i,3] <- min(set[i,]) qstat[i,4] <- max(set[i,]) }
colnames(qstat) <- c('Mean','SD','Min','Max')
rownames(qstat) <- c(1:dim(alldata)[1],rownames(newdata))

head(qstat)

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>SD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.176</td>
<td>0.123</td>
<td>0.098</td>
<td>0.468</td>
</tr>
<tr>
<td>2</td>
<td>0.116</td>
<td>0.064</td>
<td>0.055</td>
<td>0.255</td>
</tr>
<tr>
<td>3</td>
<td>0.606</td>
<td>0.136</td>
<td>0.423</td>
<td>0.852</td>
</tr>
<tr>
<td>4</td>
<td>0.347</td>
<td>0.101</td>
<td>0.277</td>
<td>0.584</td>
</tr>
<tr>
<td>5</td>
<td>0.467</td>
<td>0.129</td>
<td>0.373</td>
<td>0.770</td>
</tr>
<tr>
<td>6</td>
<td>0.036</td>
<td>0.034</td>
<td>0.009</td>
<td>0.116</td>
</tr>
</tbody>
</table>
```

Thus, for example, the first faculty member in the database has an average quantile value of 0.176 with a standard deviation of 0.123 across the 8 metrics.
Evaluation of Individual Faculty

In this section, we focus on the set of faculty members being evaluated based on the existing database (that is, the individuals contained in the ‘Inputfaculty.csv’ file, 3 individuals in total). Hereafter, we refer to these 3 individuals as ‘candidates’.

First, using `stdres`, we obtain quantile values for each candidate × metric combination assuming residuals come from a standard normal distribution:

```r
first <- dim(alldata)[1] + 1 # index of first candidate
last <- dim(alldata)[1] + dim(newdata)[1] # index of last candidate
quantile.normal <- round(t(apply(stdres[first:last,],1,pnorm)),3)
colnames(quantile.normal) <- colnames(Y.new)
row.names(quantile.normal) <- row.names(newdata)
```

<table>
<thead>
<tr>
<th></th>
<th>nPubs</th>
<th>h</th>
<th>mqPhD</th>
<th>hb</th>
<th>citesperyrPhD</th>
<th>nCites</th>
<th>mindex</th>
<th>rindex</th>
</tr>
</thead>
<tbody>
<tr>
<td>candidate.x</td>
<td>0.155</td>
<td>0.146</td>
<td>0.169</td>
<td>0.079</td>
<td>0.068</td>
<td>0.085</td>
<td>0.070</td>
<td>0.088</td>
</tr>
<tr>
<td>candidate.y</td>
<td>0.487</td>
<td>0.387</td>
<td>0.596</td>
<td>0.289</td>
<td>0.283</td>
<td>0.223</td>
<td>0.246</td>
<td>0.247</td>
</tr>
<tr>
<td>candidate.z</td>
<td>0.222</td>
<td>0.653</td>
<td>0.610</td>
<td>0.821</td>
<td>0.831</td>
<td>0.810</td>
<td>0.852</td>
<td>0.881</td>
</tr>
</tbody>
</table>

Using the summary statistics calculated earlier (mean, 95% confidence interval, range) for each candidate across residuals from all metrics we calculate a quantile value for each candidate × statistic combination (still assuming a standard normal distribution for residuals):

```r
summary.normal <- matrix(NA,nrow=dim(newdata)[1],ncol=5)
colnames(summary.normal) <- c('Mean', '2.5%', '97.5', 'Min', 'Max')
rownames(summary.normal) <- rownames(newdata)
summary.normal[,1] <- pnorm(r.mean[first:last])
summary.normal[,2] <- pnorm(r.mean[first:last] - r.ci[first:last])
summary.normal[,3] <- pnorm(r.mean[first:last] + r.ci[first:last])
summary.normal[,4] <- pnorm(r.min[first:last])
summary.normal[,5] <- pnorm(r.max[first:last])
summary.normal <- round(summary.normal,3)
```

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>2.5%</th>
<th>97.5</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>candidate.x</td>
<td>0.103</td>
<td>0.078</td>
<td>0.133</td>
<td>0.068</td>
<td>0.169</td>
</tr>
<tr>
<td>candidate.y</td>
<td>0.338</td>
<td>0.252</td>
<td>0.433</td>
<td>0.223</td>
<td>0.596</td>
</tr>
<tr>
<td>candidate.z</td>
<td>0.729</td>
<td>0.567</td>
<td>0.854</td>
<td>0.222</td>
<td>0.881</td>
</tr>
</tbody>
</table>

Next, we take an alternative approach to calculation of quantiles We calculate quantile values for each candidate × metric combination in relation to the larger faculty database (instead of assuming a normal distribution).
quantile.database <- round(set[first:last,],3)
colnames(quantile.database) <- colnames(Y.new)
row.names(quantile.database) <- row.names(newdata)

summary.database <- matrix(NA,nrow=dim(newdata)[1],ncol=5)
colnames(summary.database) <- c('Mean','2.5%','97.5%','Min','Max')
row.names(summary.database) <- rownames(newdata)
summary.database[,1] <- qstat[first:last,1]
summary.database[,3] <- summary.database[,1] + rp.ci
summary.database[,4] <- qstat[first:last,3]
summary.database <- round(summary.database,3)

Multivariate Performance Benchmarking

Here we demonstrate a factor analysis approach for benchmarking the performance of individual faculty candidates. First, conduct the factor analysis and examine the results:

fa <- factanal(stdres,factors=2,scores='reg',rotation="varimax")

Using the results from the factor analysis, plot convex hulls for 100, 95, 90, 75, and 50% of the factor scores. The locations of the three candidates along the factor axes are highlighted.

p100 <- plot.hulls(fa$scores,fraction=1.0,main='',pch=20,cex=.65,lwd=2,col="gray",
xlab="Factor 1",ylab="Factor 2")
p95  <- plot.hulls(fa$scores,fraction=0.95,add=TRUE,lwd=2,main='',pch=20,cex=.65,col="gray")
p90  <- plot.hulls(fa$scores,fraction=0.9,add=TRUE,lwd=2,main='',pch=20,cex=.65,col="gray")
Next, calculate the quantiles for the factor scores (considered separately) of the candidates relative to the faculty database:

```r
q <- matrix(data=NA,nrow=dim(Y.new)[1],ncol=dim(fa$scores)[2])
for(i in 1:dim(Y.new)[1]){
  for(j in 1:dim(fa$scores)[2]){
    q[i,j] <- sum(fa$scores[,j] <= fa$scores[i,j])/(dim(Y.new)[1])
  }
}
```

Next, calculate the quantiles for the factor scores (considered separately) of the candidates relative to the faculty database:

```r
q <- matrix(data=NA,nrow=dim(Y.new)[1],ncol=dim(fa$scores)[2])
for(i in 1:dim(Y.new)[1]){
  for(j in 1:dim(fa$scores)[2]){
    q[i,j] <- sum(fa$scores[,j] <= fa$scores[i,j])/(dim(Y.new)[1])
  }
}
```
factor.quantiles <- `round(q[first:last,],3)
colnames(factor.quantiles) <- c('Factor 1','Factor 2')
row.names(factor.quantiles) <- row.names(newdata)

<table>
<thead>
<tr>
<th></th>
<th>Factor 1</th>
<th>Factor 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>candidate.x</td>
<td>0.116</td>
<td>0.361</td>
</tr>
<tr>
<td>candidate.y</td>
<td>0.202</td>
<td>0.691</td>
</tr>
<tr>
<td>candidate.z</td>
<td>0.968</td>
<td>0.186</td>
</tr>
</tbody>
</table>

Now we obtain a single score for each candidate, by averaging the factor scores together while weighting the scores by the corresponding eigenvalues (i.e., proportional to the amount of variation each factor explains). We then rank each candidate (out of 440 total faculty) according to the averaged factor score.

lam <- colSums(fa$loading*fa$loading)/dim(fa$loading)[1] #calculate prop var explained
f <- matrix(data=NA,nrow=2,ncol=1)
f[1] <- lam[1]/sum(lam) #weight for factor 1
f[2] <- lam[2]/sum(lam) #weight for factor 2
s.w <- fa$scores%*%f
factor.average <- matrix(NA,nrow=dim(newdata)[1],ncol=2)
colnames(factor.average) <- c('Score','Rank')
row.names(factor.average) <- row.names(newdata)

factor.average[,1] <- `round(s.w[first:last],3)
sw.rank <- rank(s.w) #rank factor scores
factor.average[,2] <- `round(sw.rank[first:last])

<table>
<thead>
<tr>
<th></th>
<th>Score</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>candidate.x</td>
<td>-0.807</td>
<td>48</td>
</tr>
<tr>
<td>candidate.y</td>
<td>-0.323</td>
<td>129</td>
</tr>
<tr>
<td>candidate.z</td>
<td>0.855</td>
<td>388</td>
</tr>
</tbody>
</table>