Week 5, Lecture 09
Advanced statistical methods, part I: Ecological analyses, ordinal data, and dimensionality reduction

Richard E.W. Berl

Spring 2019

Ecological analyses

Download the data from the annual Audubon Christmas Bird Count here: http://netapp.audubon.org/CBCObservation/Historical/ResultsByCount.aspx

Let's get all of the data available for Fort Collins: For "Start Year," select Count 1 in 1900; leave "End Year" at 2017; select "United States" and "Colorado" and flip through the pages until you find "Fort Collins" (was at the bottom of page 2 for me). Click the bubble, select CSV and Export.

Place the data in your `/data` folder.

```
library(readr)
```

```
fcbird = as.data.frame(read_csv("./data/HistoricalResultsByCount [COFC-1901-2018].csv",
                    skip=208, n_max=18031))
```

```
## Parsed with column specification:
## cols(
##   COM_NAME = col_character(),
##   CountYear = col_character(),
##   how_manyCW = col_character(),
##   NumberByPartyHours = col_double(),
##   Flags = col_character()
## )
```

```
## Warning: 2 parsing failures.
## row # A tibble: 2 x 5 col     row col          expected        actual   file
```

```
head(fcbird)
```

```
##                                             COM_NAME
## 1 Greater White-fronted Goose\r\n[Anser albifrons]
## 2 Greater White-fronted Goose\r\n[Anser albifrons]
## 3 Greater White-fronted Goose\r\n[Anser albifrons]
## 4 Greater White-fronted Goose\r\n[Anser albifrons]
## 5 Greater White-fronted Goose\r\n[Anser albifrons]
## 6 Greater White-fronted Goose\r\n[Anser albifrons]
##                                                                                                                 CountYea
## 1       1926 [27]\r\nCount Date: 12/25/1926\r\n# Participants: \r\n# Species Reported: \r\nTotal Hrs
## 2       1927 [28]\r\nCount Date: 12/23/1927\r\n# Participants: \r\n# Species Reported: \r\nTotal Hrs
## 3  1947 [48]\r\nCount Date: 12/27/1947\r\n# Participants: \r\n# Species Reported: \r\nTotal Hrs.: 8.5
## 4 1948 [49]\r\nCount Date: 12/30/1948\r\n# Participants: \r\n# Species Reported: \r\nTotal Hrs.: 28.0
```

```
## 5 1949 [50]\r\nCount Date: 12/29/1949\r\n# Participants: \r\n# Species Reported: \r\nTotal Hrs.: 25.
## 6 1950 [51]\r\nCount Date: 12/29/1950\r\n# Participants: \r\n# Species Reported: \r\nTotal Hrs.: 18.
##   how_manyCW NumberByPartyHours Flags
## 1       <NA>                 NA  <NA>
## 2       <NA>                 NA  <NA>
## 3       <NA>                 NA  <NA>
## 4       <NA>                 NA  <NA>
## 5       <NA>                 NA  <NA>
## 6       <NA>                 NA  <NA>
```

```r
tail(fcbird)
```

```
##                                    COM_NAME
## 18026 House Sparrow\r\n[Passer domesticus]
## 18027 House Sparrow\r\n[Passer domesticus]
## 18028 House Sparrow\r\n[Passer domesticus]
## 18029 House Sparrow\r\n[Passer domesticus]
## 18030 House Sparrow\r\n[Passer domesticus]
## 18031 House Sparrow\r\n[Passer domesticus]
##
## 18026  2012 [113]\r\nCount Date: 12/15/2012\r\n# Participants: 71\r\n# Species Reported: 94\r\nTotal
## 18027  2013 [114]\r\nCount Date: 12/14/2013\r\n# Participants: 71\r\n# Species Reported: 84\r\nTotal
## 18028  2014 [115]\r\nCount Date: 12/20/2014\r\n# Participants: 75\r\n# Species Reported: 95\r\nTotal
## 18029 2015 [116]\r\nCount Date: 12/19/2015\r\n# Participants: 77\r\n# Species Reported: 100\r\nTotal
## 18030  2016 [117]\r\nCount Date: 12/17/2016\r\n# Participants: 82\r\n# Species Reported: 88\r\nTotal
## 18031 2017 [118]\r\nCount Date: 12/16/2017\r\n# Participants: 90\r\n# Species Reported: 100\r\nTotal
##       how_manyCW NumberByPartyHours Flags
## 18026       2462            18.2370   HC,
## 18027       1694            11.4537  <NA>
## 18028       1409             9.6972  <NA>
## 18029       1443             9.5880  <NA>
## 18030        760             5.7445  <NA>
## 18031       1022             7.1469  <NA>
```

```r
library(stringr)
```

```r
fcbird$SPEC_NAME = str_split_fixed(fcbird$COM_NAME, "\\r\\n", 2)[,2]
fcbird$SPEC_NAME = gsub("\\[|\\]", "", fcbird$SPEC_NAME)
fcbird$COM_NAME = str_split_fixed(fcbird$COM_NAME, "\r\n", 2)[,1]
fcbird$CountYear = as.integer(substr(fcbird$CountYear, 1, 4))

fcbird = fcbird[,c("COM_NAME","SPEC_NAME","CountYear","how_manyCW")]
```

```r
head(fcbird)
```

```
##                        COM_NAME       SPEC_NAME CountYear how_manyCW
## 1 Greater White-fronted Goose Anser albifrons      1926       <NA>
## 2 Greater White-fronted Goose Anser albifrons      1927       <NA>
## 3 Greater White-fronted Goose Anser albifrons      1947       <NA>
## 4 Greater White-fronted Goose Anser albifrons      1948       <NA>
## 5 Greater White-fronted Goose Anser albifrons      1949       <NA>
## 6 Greater White-fronted Goose Anser albifrons      1950       <NA>
```

```r
tail(fcbird)
```

```
##            COM_NAME         SPEC_NAME CountYear how_manyCW
## 18026 House Sparrow Passer domesticus      2012       2462
## 18027 House Sparrow Passer domesticus      2013       1694
```

```
## 18028 House Sparrow Passer domesticus      2014       1409
## 18029 House Sparrow Passer domesticus      2015       1443
## 18030 House Sparrow Passer domesticus      2016        760
## 18031 House Sparrow Passer domesticus      2017       1022
```

Cheat Sheet

```
library(tidyr)
```

```
fcbirdW = spread(fcbird[,-2], "COM_NAME", "how_manyCW")
```

`vegan` doesn't accept missing values :(

```
l1 = combn(2:length(fcbirdW[,-1]), 2, function(x) fcbirdW[,-1][x[1]:x[2]], simplify = FALSE)
# If you also need "combinations" of only single columns, then uncomment the next line
# l1 = c(d[-1], l1)
l2 = sapply(l1, function(x) sum(complete.cases(x)))

score = sapply(1:length(l1), function(i) NCOL(l1[[i]]) * l2[i])
best_score = which.max(score)
best = l1[[best_score]]
```

Source: dww on StackOverflow, 12/4/18

```
rownames(best) = fcbirdW$CountYear
best = best[complete.cases(best),]
# best = apply(best, as.numeric)
best = data.frame(lapply(best, function(x) as.numeric(as.character(x))),
                  check.names=F, row.names=rownames(best))

head(best)
```

```
##      American Crow American Dipper American Goldfinch American Kestrel
## 1952           353              12                 16                2
## 1956             5               2                 36                2
## 1957             3               3                  7                3
## 1958           168               8                  3                7
## 1960             2               5                  3                6
## 1962           590              20                  6                2
```

```
str(best)
```

```
## 'data.frame':    60 obs. of  4 variables:
##  $ American Crow     : num  353 5 3 168 2 590 130 13 100 390 ...
##  $ American Dipper   : num  12 2 3 8 5 20 15 10 2 5 ...
##  $ American Goldfinch: num  16 36 7 3 3 6 1 3 6 32 ...
##  $ American Kestrel  : num  2 2 3 7 6 2 5 4 2 12 ...
```

```
install.packages("vegan")
```

```
library(vegan)
```

```
## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.5-4
```

Diversity

```
?diversity
```

```r
diversity(best, index="shannon")
```

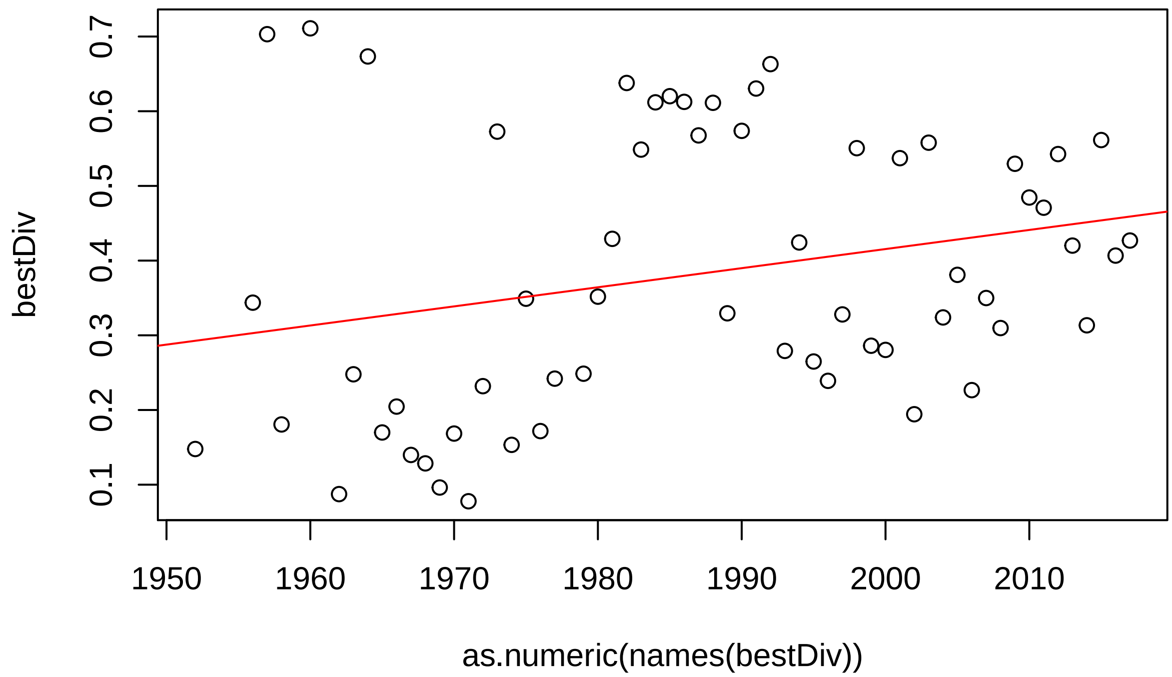
```
##      1952      1956      1957      1958      1960      1962      1963
## 0.3437796 0.6994078 1.3032836 0.4172591 1.3050964 0.2188448 0.5043830
##      1964      1965      1966      1967      1968      1969      1970
## 1.2274905 0.3910243 0.4453950 0.3129922 0.3027719 0.2456579 0.3681573
##      1971      1972      1973      1974      1975      1976      1977
## 0.2114731 0.5172964 1.0273063 0.3687373 0.7055312 0.3574685 0.5070725
##      1979      1980      1981      1982      1983      1984      1985
## 0.5238491 0.6494850 0.8190258 1.1390141 0.9851331 1.1136518 1.1115508
##      1986      1987      1988      1989      1990      1991      1992
## 1.0876915 1.0321125 1.0849486 0.6690217 0.9653766 1.1289526 1.2249457
##      1993      1994      1995      1996      1997      1998      1999
## 0.5457064 0.7918858 0.5299561 0.5084034 0.6102280 0.9501821 0.5673978
##      2000      2001      2002      2003      2004      2005      2006
## 0.5532326 0.8874201 0.4275090 0.9922356 0.6638118 0.7163728 0.4833601
##      2007      2008      2009      2010      2011      2012      2013
## 0.6654255 0.6276081 0.9762043 0.8541593 0.8501107 0.9471467 0.7681333
##      2014      2015      2016      2017
## 0.6260655 0.9791286 0.7936965 0.7690246
```

```r
diversity(best, index="simpson")
```

```
##       1952       1956       1957       1958       1960       1962
## 0.14776841 0.34370370 0.70312500 0.18065672 0.71093750 0.08741006
##       1963       1964       1965       1966       1967       1968
## 0.24779615 0.67333333 0.16991736 0.20458590 0.13981213 0.12852485
##       1969       1970       1971       1972       1973       1974
## 0.09622533 0.16844073 0.07785600 0.23192323 0.57269965 0.15336187
##       1975       1976       1977       1979       1980       1981
## 0.34899996 0.17176848 0.24199691 0.24858277 0.35173546 0.42913703
##       1982       1983       1984       1985       1986       1987
## 0.63781217 0.54863182 0.61186583 0.62013317 0.61254071 0.56760808
##       1988       1989       1990       1991       1992       1993
## 0.61126005 0.32947021 0.57373279 0.63037522 0.66306406 0.27912875
##       1994       1995       1996       1997       1998       1999
## 0.42428440 0.26492143 0.23901937 0.32795545 0.55056497 0.28605894
##       2000       2001       2002       2003       2004       2005
## 0.28045643 0.53715014 0.19434426 0.55787305 0.32392225 0.38094189
##       2006       2007       2008       2009       2010       2011
## 0.22659745 0.34990480 0.30970734 0.52964575 0.48446848 0.47083788
##       2012       2013       2014       2015       2016       2017
## 0.54263525 0.42010744 0.31339904 0.56141183 0.40671627 0.42687500
```

```r
bestDiv = diversity(best, index="simpson")

plot(as.numeric(names(bestDiv)), bestDiv)
abline(lm(bestDiv ~ as.numeric(names(bestDiv))), col="red")
```




```
cor.test(as.numeric(names(bestDiv)), bestDiv)
```

```
##
##  Pearson's product-moment correlation
##
## data:  as.numeric(names(bestDiv)) and bestDiv
## t = 2.01, df = 58, p-value = 0.04909
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.001355045 0.478133794
## sample estimates:
##       cor
## 0.2551919
```

Evenness

```
diversity(best, index="shannon") / log(specnumber(best))
```

```
##      1952      1956      1957      1958      1960      1962      1963
## 0.2479846 0.5045161 0.9401204 0.3009888 0.9414280 0.1578631 0.3638354
##      1964      1965      1966      1967      1968      1969      1970
## 0.8854472 0.2820644 0.3212846 0.2257761 0.2184037 0.1772047 0.2655694
##      1971      1972      1973      1974      1975      1976      1977
## 0.1525456 0.3731505 0.7410449 0.2659878 0.5089332 0.2578590 0.3657755
##      1979      1980      1981      1982      1983      1984      1985
## 0.3778773 0.4685044 0.5908022 0.8216250 0.7106233 0.8033300 0.8018144
```

```
##      1986      1987      1988      1989      1990      1991      1992
## 0.7846036 0.7445118 0.7826250 0.4825972 0.6963720 0.8143671 0.8836115
##      1993      1994      1995      1996      1997      1998      1999
## 0.3936440 0.5712249 0.3822826 0.3667355 0.4401865 0.6854115 0.4092910
##      2000      2001      2002      2003      2004      2005      2006
## 0.3990730 0.6401383 0.3083826 0.7157467 0.4788390 0.5167537 0.3486706
##      2007      2008      2009      2010      2011      2012      2013
## 0.4800030 0.4527236 0.7041825 0.6161457 0.6132252 0.6832219 0.5540911
##      2014      2015      2016      2017
## 0.4516108 0.7062920 0.5725310 0.5547340
```

Richness

```
?rarefy
```

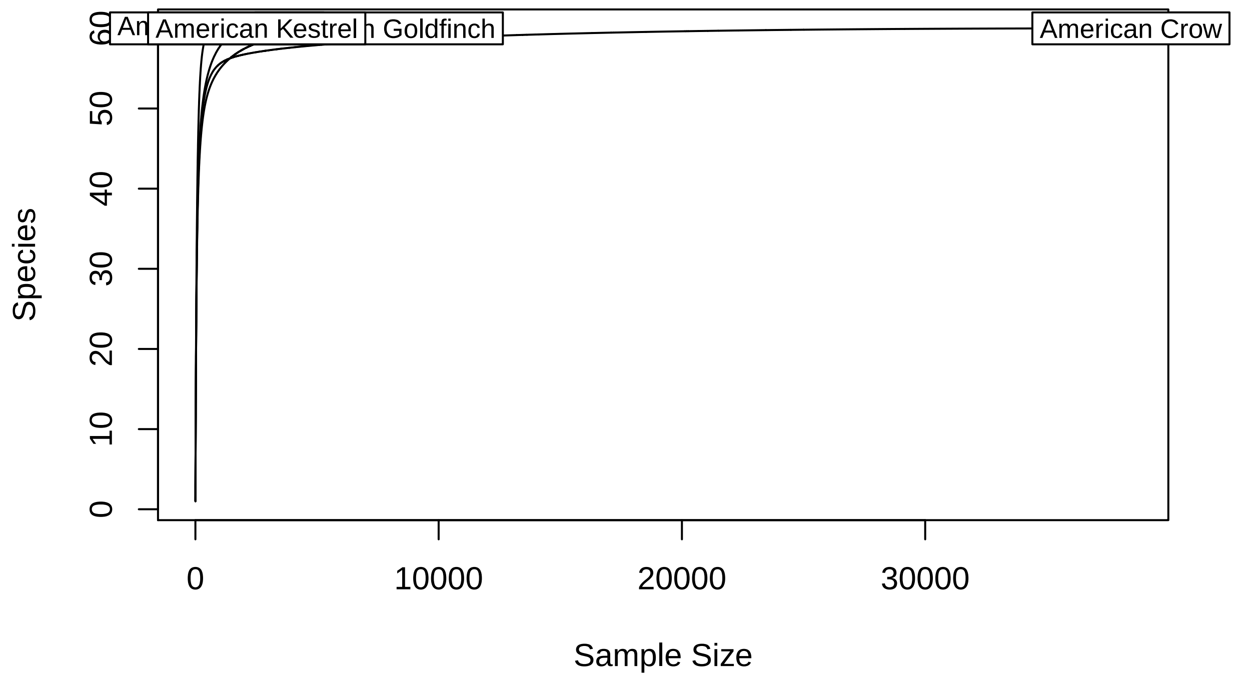
```
rarefy(best, sample=10)
```

```
##     1952     1956     1957     1958     1960     1962     1963     1964
## 1.677800 2.532271 3.892857 1.841751 3.837787 1.407843 2.020324 3.535623
##     1965     1966     1967     1968     1969     1970     1971     1972
## 1.792072 1.888261 1.607926 1.580720 1.454925 1.721593 1.378540 2.059964
##     1973     1974     1975     1976     1977     1979     1980     1981
## 2.963610 1.719705 2.448158 1.700102 2.021976 2.065888 2.229997 2.630100
##     1982     1983     1984     1985     1986     1987     1988     1989
## 3.167738 2.858432 3.165771 3.121535 3.038983 3.003950 3.026253 2.362203
##     1990     1991     1992     1993     1994     1995     1996     1997
## 2.658250 3.145499 3.461613 2.057313 2.557292 2.051739 2.021945 2.158362
##     1998     1999     2000     2001     2002     2003     2004     2005
## 2.683295 2.116202 2.104663 2.536735 1.850560 2.841343 2.356356 2.375419
##     2006     2007     2008     2009     2010     2011     2012     2013
## 1.972593 2.282606 2.260685 2.904340 2.573524 2.595579 2.762594 2.485045
##     2014     2015     2016     2017
## 2.259514 2.778704 2.597883 2.445090
## attr(,"Subsample")
## [1] 10
```

```
head(rarefy(best, sample=c(5, 15)))
```

```
##            N5      N15
## [1,] 1.366906 1.941366
## [2,] 1.885565 3.004572
## [3,] 3.087225 4.000000
## [4,] 1.454503 2.171092
## [5,] 3.083562 4.000000
## [6,] 1.216024 1.578600
```

```
rarecurve(t(best))
```




```
?specaccum
```

`diverse` package

Ordinal data

Let's get some Human Dimensions data for once! Go to the US Forest Service page for the 2004 visitor preference and usage data set for the Bob Marshall Wilderness Complex in Montana: https://www.fs.usda.gov/rds/archive/Product/RDS-2017-0016

At the bottom, click "Download data publication," which gives you a ZIP archive. Open it up, go into the "Data" folder and pull out both CSVs for your `/data` directory. You can hang on to the other files in the archive as well, for the metadata.

For now, let's load in the onsite data:

```
bm = read.csv("./data/BMWC2004_onsitedata.csv", header=T, na.strings="88",
              stringsAsFactors=F)
```

```
head(bm)
```

```
##    id. newweigh    first_ma   reminder      resend date_ret group_.
## 1 2000    1.215 13-JUL-2004 24.07.2004 07-AUG-2004  9/16/04       1
## 2 2001    1.215 13-JUL-2004 24.07.2004 07-AUG-2004  9/16/04       1
## 3 2002    1.215 13-JUL-2004                         7/19/04       2
## 4 2003    1.215 13-JUL-2004 24.07.2004              7/26/04       2
## 5 2004    1.215 13-JUL-2004 24.07.2004 07-AUG-2004   8/9/04       3
## 6 2005    1.215 13-JUL-2004 24.07.2004 07-AUG-2004                3
```

```
##         city st stcode poolstcd zip_code trailhea    date_con sumfall
## 1       Troy MT      1        1    59935       12 18-JUN-2004       1
## 2       Troy MT      1        1    59935       12 18-JUN-2004       1
## 3 Kalispell MT      1        1    59901       12 18-JUN-2004       1
## 4 Kalispell MT      1        1    59901       12 18-JUN-2004       1
## 5  Florance MT      1        1    59833       12 18-JUN-2004       1
## 6  Missoula MT      1        1    59801       12 18-JUN-2004       1
##   time_of entering wilderne overnigh length_o lengcats outfitte type_of
## 1    1900        2        1        1        7        5        2       2
## 2    1900        2        1        1        7        5        2       2
## 3    2000        1        1        1        2        2        2       1
## 4    2000        1        1        1        2        2        2       1
## 5    2030        2        1        1        1        2        2       2
## 6    2030        2        1        1        1        2        2       2
##   hikehors stocknum stockcat numnons         reason_f visitbef prvsvist
## 1        2        7        3       1 Mentally impared        2        0
## 2        2       NA       NA      NA                         2        0
## 3        1        0        0       0                         1       12
## 4        1       NA       NA      NA                         1       10
## 5        2        5        2       0                         2        0
## 6        2       NA       NA      NA                         1        3
##   aware_of affect_p
## 1        1        2
## 2        1        2
## 3        1        1
## 4        1        2
## 5        1        2
## 6        1        2
##                                                            how v28 v29
## 1                                                                2
## 2                                                                2
## 3 The area was basically shut down there was so much caution   2
## 4                                                                2
## 5                                                                2
## 6                                                                2
##   natural remotnes scenic_b hunting fishing recent_f test_ski familiar
## 1       1        1        2       1       1        1        3        2
## 2       1        1        2       1       1        1        3        2
## 3       3        3        3       2       3        1        2        3
## 4       3        3        3       3       3        1        2        2
## 5       2        3        3       3       3        2        2        2
## 6       3        3        3       1       3        1        1        3
##   variety friend_s date_of age agecats educatio female filter_.
## 1       2        1      50  54      54       NA      2        1
## 2       2        1      52  52      52       NA      1        1
## 3       1        2      81  23      23       16      2        0
## 4       2        2      82  22      22       16      2        0
## 5       2        2      61  43      43       14      2        1
## 6       1        1      63  41      41       16      2        1
```

Likert data

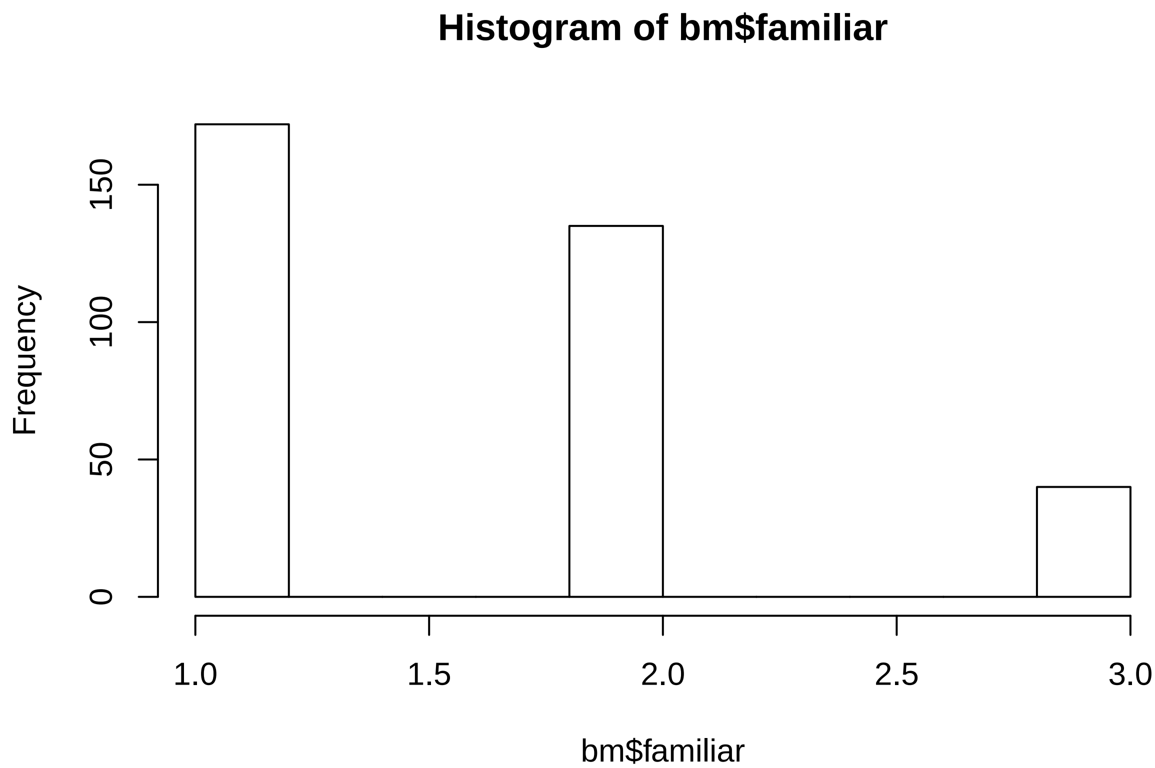
```
summary(bm[,36:45])
```

```
##     natural         remotnes         scenic_b         hunting     
##  Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.000  
##  1st Qu.:2.00   1st Qu.:3.000   1st Qu.:3.000   1st Qu.:1.000  
##  Median :3.00   Median :3.000   Median :3.000   Median :1.000  
##  Mean   :2.67   Mean   :2.768   Mean   :2.844   Mean   :1.554  
##  3rd Qu.:3.00   3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:2.000  
##  Max.   :3.00   Max.   :3.000   Max.   :3.000   Max.   :3.000  
##  NA's   :57     NA's   :56      NA's   :56      NA's   :73     
##     fishing         recent_f         test_ski         familiar   
##  Min.   :1.000   Min.   :0.000   Min.   :1.000   Min.   :1.00  
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.00  
##  Median :3.000   Median :1.000   Median :2.000   Median :2.00  
##  Mean   :2.221   Mean   :1.494   Mean   :1.744   Mean   :1.62  
##  3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.00  
##  Max.   :3.000   Max.   :3.000   Max.   :3.000   Max.   :3.00  
##  NA's   :61      NA's   :61      NA's   :62      NA's   :62    
##     variety         friend_s    
##  Min.   :1.000   Min.   :1.000  
##  1st Qu.:2.000   1st Qu.:1.000  
##  Median :2.000   Median :2.000  
##  Mean   :2.156   Mean   :1.835  
##  3rd Qu.:3.000   3rd Qu.:3.000  
##  Max.   :3.000   Max.   :3.000  
##  NA's   :62      NA's   :70     
```

You can't take the mean of an ordinal variable!

But you can take the median.

```
hist(bm$familiar)
```



Hypothesis testing

Permutation tests

```
install.packages("coin")
```

```
library(coin)
```

```
## Loading required package: survival
```

```
bmLik = bm
bmLik$st = factor(ifelse(bmLik$st != "MT", "Not MT", "MT"))
bmLik$familiar = ordered(bmLik$familiar)
```

```
table(bmLik$st, bmLik$familiar)
```

```
##
##           1  2  3
##   MT     94 87 26
##   Not MT 78 48 14
```

```
?independence_test
```

```
independence_test(familiar ~ st, data=bmLik)
```

```
##
##  Asymptotic General Independence Test
##
## data:  familiar (ordered) by st (MT, Not MT)
```

```
## Z = 1.7192, p-value = 0.08558
## alternative hypothesis: two.sided
```

Two-way tests, regression, etc. available on Mangiafico page

Polychoric correlations

```r
install.packages("lavaan")
```

```r
library(lavaan)
```

```
## This is lavaan 0.6-3
```

```
## lavaan is BETA software! Please report any bugs.
```

```r
?lavCor
```

```r
?psych::tetrachor
```

```r
bmLik[,36:45] = lapply(bmLik[,36:45], function(x) ordered(x))
str(bmLik)
```

```
## 'data.frame':    409 obs. of  51 variables:
##  $ id.     : int  2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ...
##  $ newweigh: num  1.22 1.22 1.22 1.22 1.22 ...
##  $ first_ma: chr  "13-JUL-2004" "13-JUL-2004" "13-JUL-2004" "13-JUL-2004" ...
##  $ reminder: chr  "24.07.2004" "24.07.2004" "" "24.07.2004" ...
##  $ resend  : chr  "07-AUG-2004" "07-AUG-2004" "" "" ...
##  $ date_ret: chr  "9/16/04" "9/16/04" "7/19/04" "7/26/04" ...
##  $ group_. : int  1 1 2 2 3 3 3 4 4 6 ...
##  $ city    : chr  "Troy" "Troy" "Kalispell" "Kalispell" ...
##  $ st      : Factor w/ 2 levels "MT","Not MT": 1 1 1 1 1 1 1 1 1 1 ...
##  $ stcode  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ poolstcd: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ zip_code: chr  "59935" "59935" "59901" "59901" ...
##  $ trailhea: int  12 12 12 12 12 12 12 12 12 12 ...
##  $ date_con: chr  "18-JUN-2004" "18-JUN-2004" "18-JUN-2004" "18-JUN-2004" ...
##  $ sumfall : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ time_of : int  1900 1900 2000 2000 2030 2030 2030 900 900 1215 ...
##  $ entering: int  2 2 1 1 2 2 2 1 1 1 ...
##  $ wilderne: int  1 1 1 1 1 1 1 1 1 2 ...
##  $ overnigh: int  1 1 1 1 1 1 1 1 1 2 ...
##  $ length_o: int  7 7 2 2 1 1 1 7 7 0 ...
##  $ lengcats: int  5 5 2 2 2 2 2 5 5 1 ...
##  $ outfitte: int  2 2 2 2 2 2 2 1 1 2 ...
##  $ type_of : int  2 2 1 1 2 2 2 4 4 1 ...
##  $ hikehors: int  2 2 1 1 2 2 2 0 0 1 ...
##  $ stocknum: int  7 NA 0 NA 5 NA NA 0 NA 0 ...
##  $ stockcat: int  3 NA 0 NA 2 NA NA 0 NA 0 ...
##  $ numnons : int  1 NA 0 NA 0 NA NA 2 NA 2 ...
##  $ reason_f: chr  "Mentally impared" "" "" "" ...
##  $ visitbef: int  2 2 1 1 2 1 1 2 1 1 ...
##  $ prvsvist: int  0 0 12 10 0 3 10 0 6 9 ...
##  $ aware_of: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ affect_p: int  2 2 1 2 2 2 2 2 2 1 ...
##  $ how     : chr  "" "" "The area was basically shut down there was so much caution" "" ...
##  $ v28     : int  2 2 2 2 2 2 2 2 2 2 ...
```

```
##  $ v29     : chr  "" "" "" "" ...
##  $ natural : Ord.factor w/ 3 levels "1"<"2"<"3": 1 1 3 3 2 3 3 3 3 3 ...
##  $ remotnes: Ord.factor w/ 3 levels "1"<"2"<"3": 1 1 3 3 3 3 3 3 3 3 ...
##  $ scenic_b: Ord.factor w/ 3 levels "1"<"2"<"3": 2 2 3 3 3 3 3 3 3 3 ...
##  $ hunting : Ord.factor w/ 3 levels "1"<"2"<"3": 1 1 2 3 3 1 3 2 3 3 ...
##  $ fishing : Ord.factor w/ 3 levels "1"<"2"<"3": 1 1 3 3 3 3 3 3 3 3 ...
##  $ recent_f: Ord.factor w/ 4 levels "0"<"1"<"2"<"3": 2 2 2 2 3 2 3 2 2 4 ...
##  $ test_ski: Ord.factor w/ 3 levels "1"<"2"<"3": 3 3 2 2 2 1 2 2 2 3 ...
##  $ familiar: Ord.factor w/ 3 levels "1"<"2"<"3": 2 2 3 2 2 3 2 1 1 3 ...
##  $ variety : Ord.factor w/ 3 levels "1"<"2"<"3": 2 2 1 2 2 1 2 3 3 3 ...
##  $ friend_s: Ord.factor w/ 3 levels "1"<"2"<"3": 1 1 2 2 2 1 2 1 1 3 ...
##  $ date_of : int  50 52 81 82 61 63 77 65 63 82 ...
##  $ age     : int  54 52 23 22 43 41 27 39 41 22 ...
##  $ agecats : int  54 52 23 22 43 41 27 39 41 22 ...
##  $ educatio: int  NA NA 16 16 14 16 12 16 13 13 ...
##  $ female  : int  2 1 2 2 2 2 2 1 2 2 ...
##  $ filter_.: int  1 1 0 0 1 1 1 NA NA 0 ...
```

```
bmLik$recent_f
```

```
##   [1] 1    1    1    1    2    1    2    1    1    3    1    2    1    1   
##  [15] 1    1    1    1    1    1    1    1    1    2    3    1    1    3   
##  [29] 1    1    2    1    2    2    3    2    1    3    2    1    3    1   
##  [43] 3    1    3    1    1    1    3    2    1    1    1    1    1    1   
##  [57] 2    1    1    1    1    2    1    1    2    1    2    2    1    1   
##  [71] 2    1    1    1    2    1    1    2    2    3    1    2    1    1   
##  [85] 2    3    3    3    2    2    1    1    3    1    3    1    1    3   
##  [99] 1    1    1    2    1    2    1    2    1    1    1    <NA> 1    1   
## [113] 1    1    1    <NA> 2    <NA> 2    <NA> 1    1    1    1    2    1   
## [127] 2    3    <NA> <NA> 3    3    1    1    1    2    2    1    1    1   
## [141] 1    2    2    2    1    1    1    2    1    1    2    2    2    2   
## [155] 1    1    1    2    2    1    1    1    <NA> 1    1    1    1    2   
## [169] <NA> <NA> 3    1    1    2    <NA> 1    <NA> 1    2    2    1    2   
## [183] <NA> 2    1    1    1    1    2    1    1    1    2    <NA> <NA> 1   
## [197] <NA> 1    1    <NA> 2    <NA> <NA> 2    2    1    2    1    3    1   
## [211] 2    1    <NA> 2    1    2    <NA> 1    3    1    2    3    2    1   
## [225] 2    1    1    3    0    2    1    1    1    1    1    3    1    1   
## [239] 2    1    2    2    1    1    2    1    1    3    1    1    1    2   
## [253] 2    2    1    1    1    <NA> 1    2    1    3    <NA> 2    2    2   
## [267] 1    1    2    2    1    1    1    1    2    1    <NA> 1    1    1   
## [281] 2    1    1    2    2    <NA> 1    2    1    1    1    1    1    2   
## [295] 2    1    1    <NA> <NA> 2    <NA> <NA> 3    3    1    <NA> <NA> <NA>
## [309] <NA> 1    <NA> 2    <NA> 1    2    2    1    1    1    1    1    1   
## [323] 3    2    1    2    <NA> <NA> <NA> <NA> 1    <NA> 2    1    1    1   
## [337] 1    2    1    2    2    2    2    2    <NA> <NA> <NA> 2    <NA> <NA>
## [351] 1    2    2    1    2    1    2    2    <NA> <NA> <NA> <NA> <NA> 1   
## [365] 1    1    1    1    1    1    2    <NA> <NA> 1    1    1    <NA> <NA>
## [379] <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> 1    2    1    1    2    2   
## [393] 2    2    2    2    1    2    2    1    1    1    2    1    1    1   
## [407] 1    3    1   
## Levels: 0 < 1 < 2 < 3
```

```
bmLikCor = lavCor(bmLik[,36:45])
bmLikCor
```

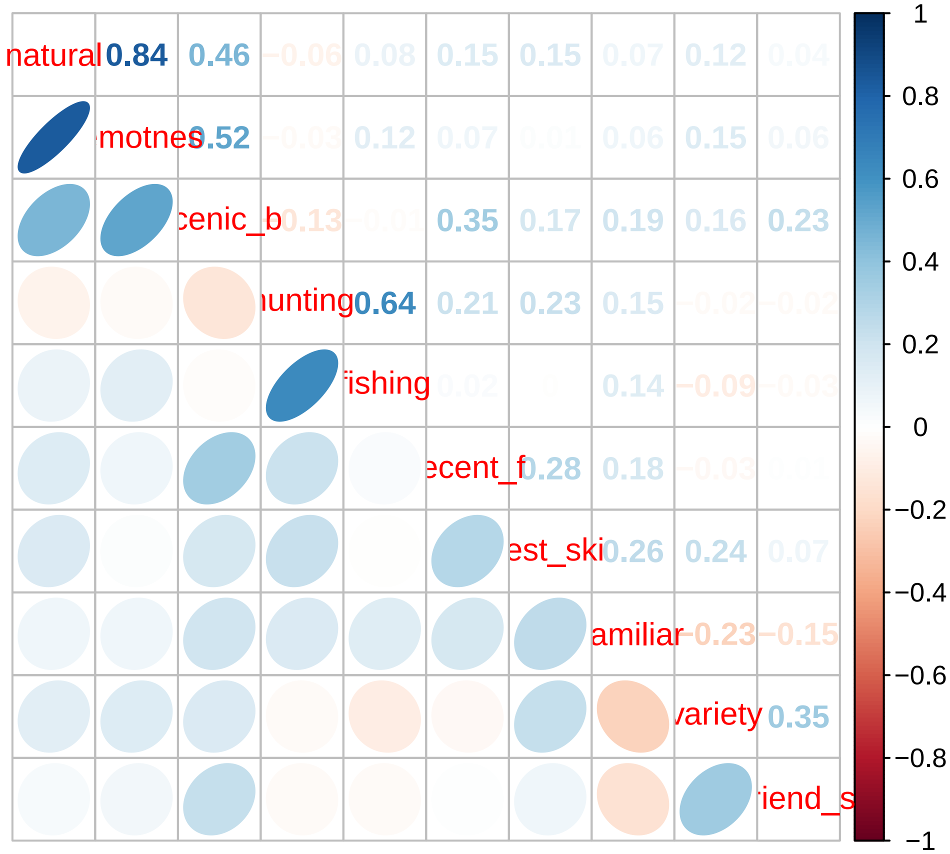
```
##          naturl remtns scnc_b huntng fishng rcnt_f tst_sk familr varity
```

```
## natural   1.000
## remotnes  0.839  1.000
## scenic_b  0.456  0.524  1.000
## hunting  -0.063 -0.029 -0.133  1.000
## fishing   0.083  0.122 -0.014  0.635  1.000
## recent_f  0.146  0.065  0.350  0.213  0.025  1.000
## test_ski  0.154  0.014  0.173  0.229 -0.003  0.280  1.000
## familiar  0.069  0.063  0.192  0.153  0.137  0.180  0.256  1.000
## variety   0.123  0.146  0.156 -0.020 -0.090 -0.035  0.238 -0.226  1.000
## friend_s  0.035  0.056  0.234 -0.022 -0.026  0.008  0.069 -0.153  0.353
##          frnd_s
## natural
## remotnes
## scenic_b
## hunting
## fishing
## recent_f
## test_ski
## familiar
## variety
## friend_s  1.000
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corrplot.mixed(bmLikCor, lower="ellipse", upper="number")
```



Treating ordinal data as continuous

If you have at least 6 levels and good sample size, you're usually okay.

See:

Rhemtulla, M., et al. (2012). When can categorical variables be treated as continuous? A comparison of robust continuous and categorical SEM estimation methods under suboptimal conditions. Psychological Methods, 17(3), 354. doi: 10.1037/a0029315

Dimensionality reduction

Multidimensional scaling

```
install.packages("psych")
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following object is masked from 'package:lavaan':
##
##     cor2cov
```

```
?cor2dist
```

```
bmLikDist = as.dist(cor2dist(bmLikCor))
bmLikDist
```

```
##            natural  remotnes  scenic_b   hunting   fishing  recent_f
## remotnes 0.5677948
## scenic_b 1.0434163 0.9758939
## hunting  1.4583126 1.4346551 1.5050812
## fishing  1.3545159 1.3249712 1.4242586 0.8543437
## recent_f 1.3070717 1.3674042 1.1404685 1.2546118 1.3965050
## test_ski 1.3007151 1.4041365 1.2860976 1.2416681 1.4164165 1.1998816
## familiar 1.3643657 1.3692839 1.2715226 1.3013499 1.3137339 1.2809036
## variety  1.3242858 1.3071400 1.2990552 1.4283631 1.4767730 1.4384538
## friend_s 1.3890874 1.3737328 1.2374257 1.4298405 1.4323002 1.4084506
##           test_ski  familiar   variety
## remotnes
## scenic_b
## hunting
## fishing
## recent_f
## test_ski
## familiar 1.2201762
## variety  1.2344690 1.5660747
## friend_s 1.3645824 1.5186283 1.1375152
```

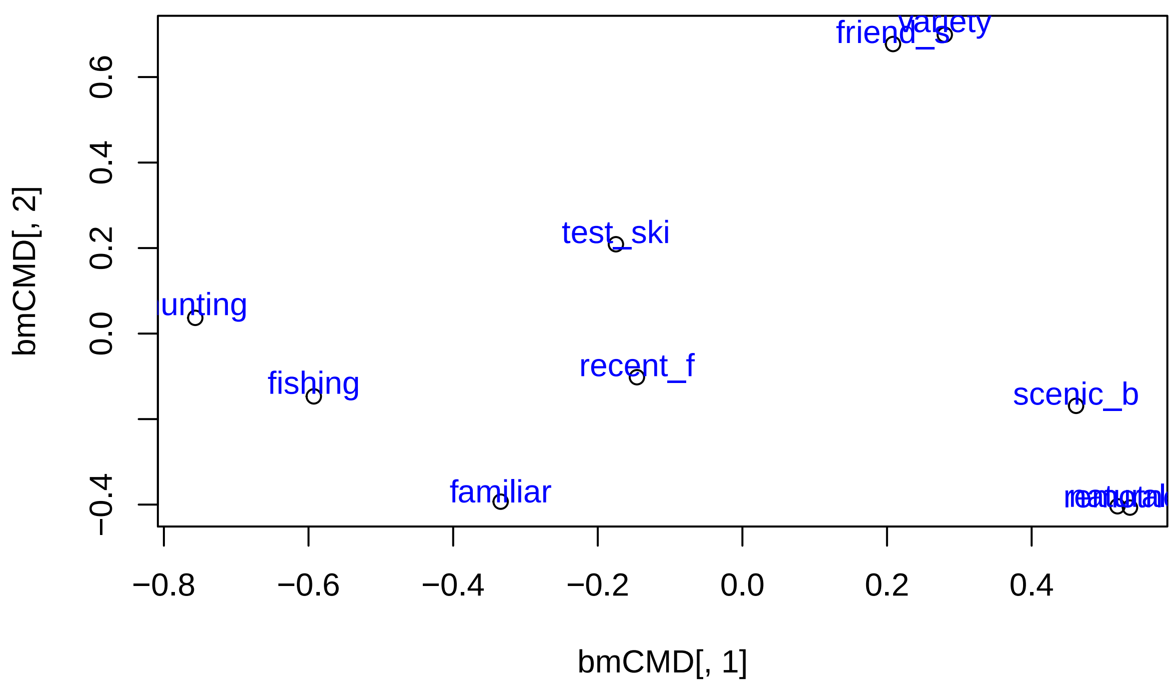
Classical

```
?cmdscale
```

```
bmCMD = cmdscale(bmLikDist)
bmCMD
```

```
##                [,1]        [,2]
## natural   0.5187538 -0.40388770
## remotnes  0.5359868 -0.40700986
## scenic_b  0.4614625 -0.16901151
## hunting  -0.7566374  0.03678534
## fishing  -0.5926913 -0.14685552
## recent_f -0.1458461 -0.10204178
## test_ski -0.1749012  0.20894225
## familiar -0.3343000 -0.39312997
## variety   0.2798524  0.69901635
## friend_s  0.2083204  0.67719239
```

```r
plot(bmCMD[,1], bmCMD[,2])
text(bmCMD[,1], bmCMD[,2] + 0.025, labels=row.names(bmCMD), col="blue")
```



Nonmetric

Tries to reproduce ranks of distances rather than distance values themselves

```r
library(MASS)
```

```r
?isoMDS
```

```r
bmNMD = isoMDS(bmLikDist)
```

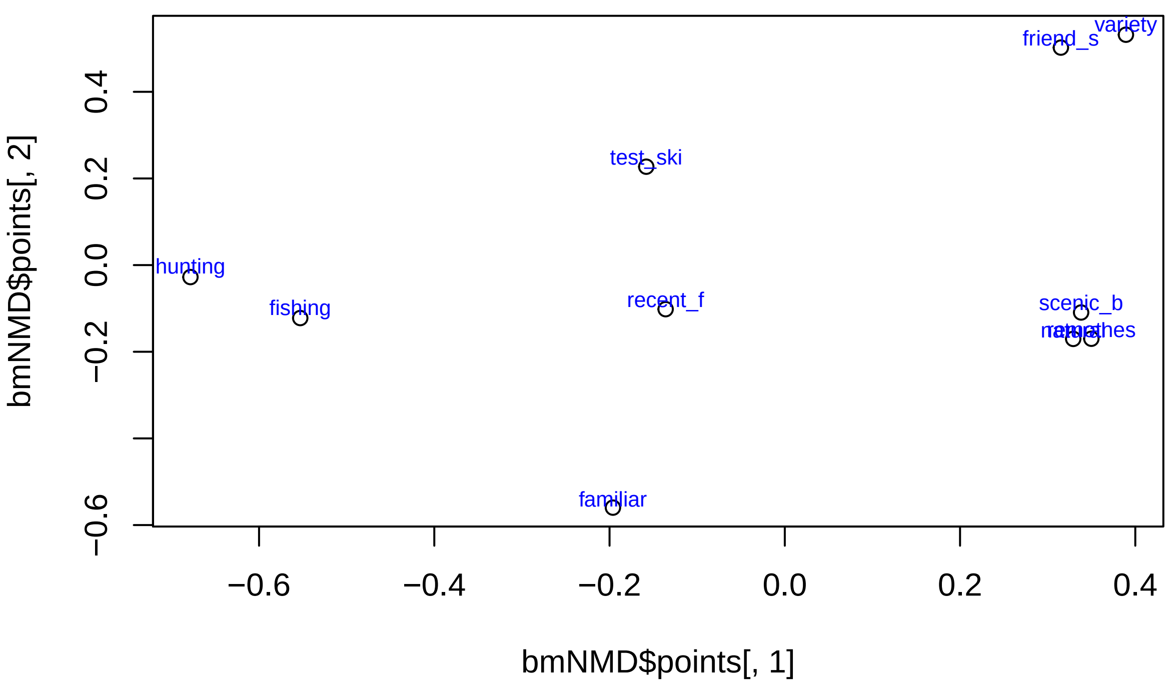
```
## initial  value 20.430062
## iter   5 value 14.383104
## iter  10 value 13.790265
```

```
## iter  10 value 13.776872
## final  value 13.658003
## converged
```

```
bmNMD
```

```
## $points
##                 [,1]        [,2]
## natural   0.3291576 -0.17033374
## remotnes  0.3498663 -0.16999471
## scenic_b  0.3381752 -0.10932250
## hunting  -0.6783328 -0.02761269
## fishing  -0.5530184 -0.12219738
## recent_f -0.1359963 -0.10171052
## test_ski -0.1581010  0.22715111
## familiar -0.1961495 -0.55973237
## variety   0.3893220  0.53172442
## friend_s  0.3150770  0.50202839
##
## $stress
## [1] 13.658
```

```
plot(bmNMD$points[,1], bmNMD$points[,2])
text(bmNMD$points[,1], bmNMD$points[,2] + 0.02,
     labels=row.names(bmNMD$points), col="blue", cex=0.7)
```




```
?vegan::metaMDS
```

```r
head(best)
```

```
##      American Crow American Dipper American Goldfinch American Kestrel
## 1952           353              12                 16                2
## 1956             5               2                 36                2
## 1957             3               3                  7                3
## 1958           168               8                  3                7
## 1960             2               5                  3                6
## 1962           590              20                  6                2
```

```r
bestNMD = metaMDS(best)
```

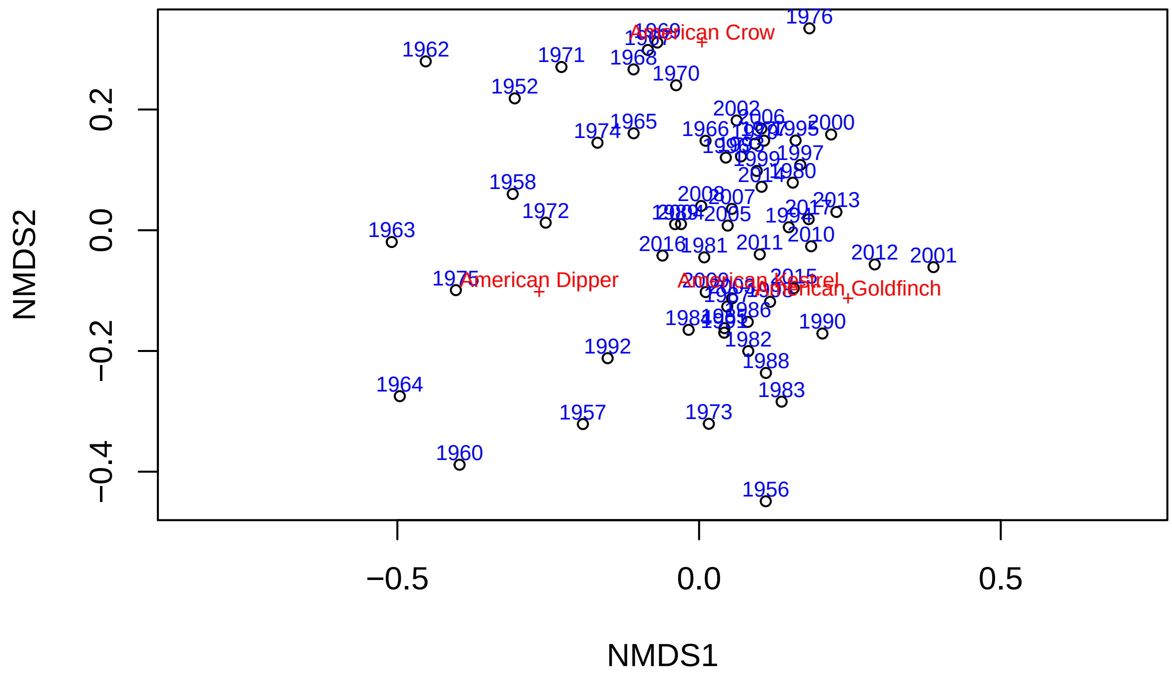
```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.08931323
## Run 1 stress 0.08931331
## ... Procrustes: rmse 4.477757e-05  max resid 0.0002967684
## ... Similar to previous best
## Run 2 stress 0.08931324
## ... Procrustes: rmse 5.45093e-05  max resid 0.0003518054
## ... Similar to previous best
## Run 3 stress 0.0892889
## ... New best solution
## ... Procrustes: rmse 0.001499923  max resid 0.008328837
## ... Similar to previous best
## Run 4 stress 0.08928883
## ... New best solution
## ... Procrustes: rmse 3.714837e-05  max resid 0.0002477234
## ... Similar to previous best
## Run 5 stress 0.08928882
## ... New best solution
## ... Procrustes: rmse 2.027499e-05  max resid 0.0001269479
## ... Similar to previous best
## Run 6 stress 0.08931323
## ... Procrustes: rmse 0.001499376  max resid 0.008330408
## ... Similar to previous best
## Run 7 stress 0.08928928
## ... Procrustes: rmse 0.0001214977  max resid 0.0007935873
## ... Similar to previous best
## Run 8 stress 0.08928953
## ... Procrustes: rmse 0.0001655092  max resid 0.001094084
## ... Similar to previous best
## Run 9 stress 0.08928881
## ... New best solution
## ... Procrustes: rmse 7.628223e-06  max resid 2.70645e-05
## ... Similar to previous best
## Run 10 stress 0.08931327
## ... Procrustes: rmse 0.001494348  max resid 0.008335853
## ... Similar to previous best
## Run 11 stress 0.0892889
## ... Procrustes: rmse 3.280793e-05  max resid 0.0001799034
## ... Similar to previous best
## Run 12 stress 0.08928892
## ... Procrustes: rmse 6.047813e-05  max resid 0.0004031535
## ... Similar to previous best
```

```
## Run 13 stress 0.08931323
## ... Procrustes: rmse 0.001499246  max resid 0.008329029
## ... Similar to previous best
## Run 14 stress 0.08928889
## ... Procrustes: rmse 8.373068e-05  max resid 0.0005474631
## ... Similar to previous best
## Run 15 stress 0.08931351
## ... Procrustes: rmse 0.001496325  max resid 0.008336517
## ... Similar to previous best
## Run 16 stress 0.08928884
## ... Procrustes: rmse 6.547815e-05  max resid 0.0004255845
## ... Similar to previous best
## Run 17 stress 0.08931324
## ... Procrustes: rmse 0.001499752  max resid 0.00832863
## ... Similar to previous best
## Run 18 stress 0.08928882
## ... Procrustes: rmse 5.671286e-06  max resid 2.643498e-05
## ... Similar to previous best
## Run 19 stress 0.08928881
## ... New best solution
## ... Procrustes: rmse 3.982887e-05  max resid 0.0002643959
## ... Similar to previous best
## Run 20 stress 0.08928881
## ... New best solution
## ... Procrustes: rmse 7.969902e-06  max resid 5.13065e-05
## ... Similar to previous best
## *** Solution reached
```

```
bestNMD
```

```
##
## Call:
## metaMDS(comm = best)
##
## global Multidimensional Scaling using monoMDS
##
## Data:     wisconsin(sqrt(best))
## Distance: bray
##
## Dimensions: 2
## Stress:     0.08928881
## Stress type 1, weak ties
## Two convergent solutions found after 20 tries
## Scaling: centring, PC rotation, halfchange scaling
## Species: expanded scores based on 'wisconsin(sqrt(best))'
```

```
plot(bestNMD)
text(bestNMD$points[,1], bestNMD$points[,2] + 0.02,
     labels=row.names(bestNMD$points), col="blue", cex=0.7)
text(bestNMD$species[,1], bestNMD$species[,2] + 0.02,
     labels=row.names(bestNMD$species), col="red", cex=0.7)
```



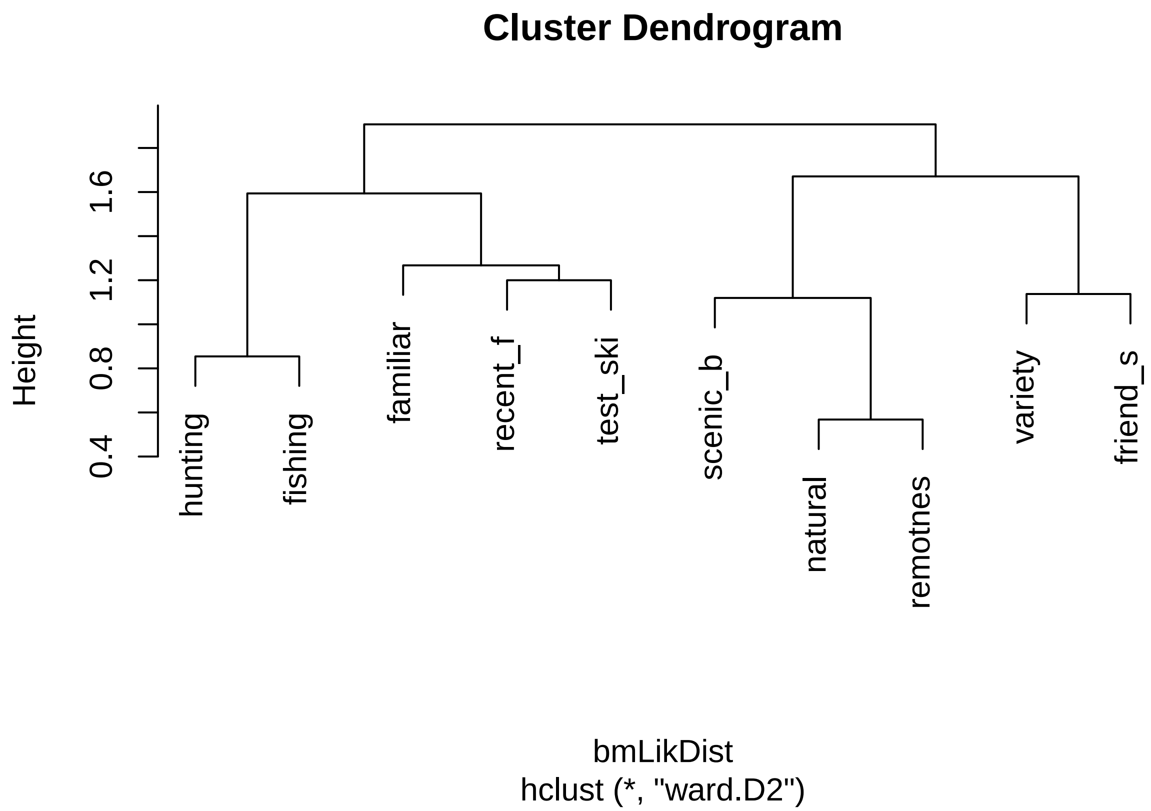
Cluster analysis

Hierarchical clustering

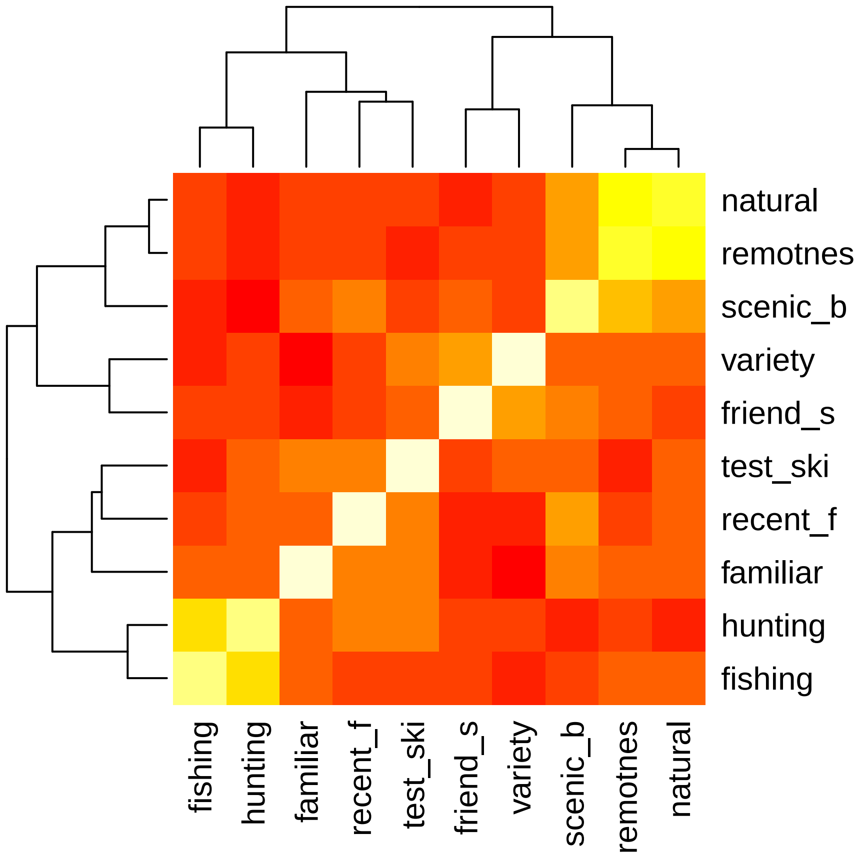
```
?hclust

bmHC = hclust(bmLikDist, method="ward.D2")

plot(bmHC)
```




```
heatmap(bmLikCor, hclustfun=function(x) hclust(x, method="ward.D2"))
```




```
install.packages("pvclust")
```

```
library(pvclust)
```

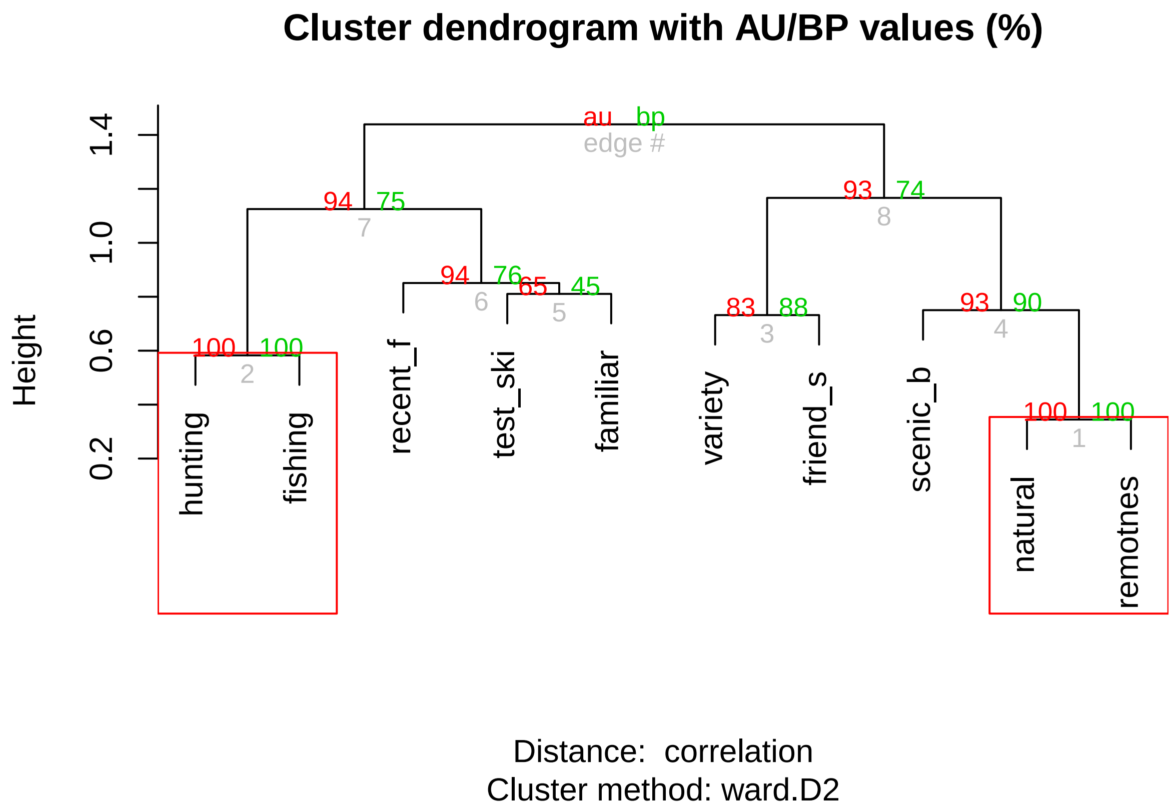
```
?pvclust
```

Needs raw data; does not allow distance matrix as input

```
bmPVHC = pvclust(bm[,36:45], method.hclust="ward.D2")
```

```
## Bootstrap (r = 0.5)... Done.
## Bootstrap (r = 0.6)... Done.
## Bootstrap (r = 0.7)... Done.
## Bootstrap (r = 0.8)... Done.
## Bootstrap (r = 0.9)... Done.
## Bootstrap (r = 1.0)... Done.
## Bootstrap (r = 1.1)... Done.
## Bootstrap (r = 1.2)... Done.
## Bootstrap (r = 1.3)... Done.
## Bootstrap (r = 1.4)... Done.
```

```
plot(bmPVHC)
pvrect(bmPVHC)
```



Red = "AU" (Approximately Unbiased) _p_value: 1 - *p*-value (>95 is "significant")

Green = "BP" (Bootstrap Probability): percent of times the tree-building algorithm produced that branch

(pdf / Rmd)