FATHOM for Matlab

A Matlab Toolbox for Multivariate Ecological & Oceanographic Data Analysis

by

Dave Jones

http://www.rsmas.miami.edu/personal/djones/
Introduction

FATHOM Toolbox for Matlab

Description

FATHOM: Matlab Toolbox for Ecological & Oceanographic Data Analysis
by Dave Jones <djones@rsmas.miami.edu>

References

Ecological & Oceanographic Data Analysis. Available from:
http://www.rsmas.miami.edu/personal/djones/

Details

The FATHOM Toolbox is a collection of Matlab functions and scripts I’ve written for
my everyday use. I’m releasing them to the public in order to encourage the sharing of
code and prevent duplication of effort. If you find this toolbox useful, drop me a line.
I’d also appreciate bug reports and suggestions for improvements.

While I’ve made every attempt to write functions that provide accurate and precise
results, the functions in this toolbox are provided as is, with no guarantees and are
only intended for non-commercial use. These routines are currently being developed
and tested with Matlab 6.5 (Release 13) under Windows XP Pro SP1.

Please note that this users manual is currently under construction and is
presently incomplete.

Note

There is currently only one issue related to installation that you need to be aware of:

fnmds: NonMetric Multidimensional Scaling

This function calls Mark Steyver’s NMDS routine. For it to work you must install his
toolbox from:

http://www-psych.stanford.edu/ msteyver/programs_data/mdszip.zip

I’ve been able to obtain better results with this program by editing Steyver’s mds.m
file and changing

randn( 'state',seed );

to
rand( 'state',seed );

This allows you to draw from *Uniformly distributed random numbers* for initial configurations rather than *Normally distributed random numbers*.
Description

One-way Analysis of Similarity

Usage

\[ [r,p] = f\_anosim(x\text{Dis}, grps, rank, iter, pw, plt) \]

Arguments

- **x\text{Dis}**: symmetric distance matrix
- **grps**: row vector designating group membership for objects in x\text{Dis}
- **rank**: optionally rank distances in x\text{Dis} (default = 1)
- **iter**: # iterations for permutation test (default = 1000)
- **pw**: do pairwise tests (default = 1)
- **plt**: make diagnostic plot (default = 0)

Details

This function performs a multivariate ANOSIM by computing an unstandardized Mantel Statistic between an (optionally ranked) distance matrix and a model matrix; the model matrix is derived from a row vector designating group membership. Results are equivalent to Clarke’s method. The permutation test permutes the rows/columns of the distance matrix \text{xDis}. Pairwise tests between each group are also optionally run. Permutation tests are based on the complete permutation distribution when it is < 5000, otherwise it is randomly sampled the number of times specified by \text{iter}.

ANOSIM assumes that under the null hypothesis distances within groups are smaller than those between groups, thus significant differences can arise between groups having different dispersions but identical centroids. Diagnostic boxplots are provided as a way to check this and prevent Type I error.

Care must be taken when coding grouping factors (i.e., \text{grps} and \text{xDis} must be sorted ascending prior to running the function); see example below.

This program has been tested against Clarke’s *Primer 5 for Windows* and gives the same results.
Value

The function returns the following values:

- $r$ strength of relationship (ranges from -1 to 1)
- $p$ permutation-based probability of no difference between groups

Author(s)

Dave Jones

References


See Also

f_anosim2, f_npManova, f_mantel, and f_modelMatrix

Examples

Load the file, anosim.mat, in the data folder. This data is from Gray et al. (1990) and ships with Clarke’s Primer program. There is 1 response variable: species, representing abundances of 174 species (rows) from 39 collection sites (columns), and 1 factor, grps, a row vector specifying group membership (i.e. categories representing distance from the center of an oilfield).

```matlab
>> load anosim.mat
>> species2 = f_transform(species,1); % square-root transform species abundances:
>> dis = f_braycurtis(species2); % Bray-Curtis symmetric distance matrix
>> [r,p] = f_anosim(dis,grps)
```

====================================================================
1-way ANOSIM RESULTS:
====================================================================

Groupings:
1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4

Global Test:
R = 0.5413  p = 0.0010 (1000 of 408316439225392430000 possible perms)

Pair-Wise Tests:
1 2: R = 0.5539  p = 0.0020 (1000 of 8008 possible perms)
1 3: R = 0.8200  p = 0.0010 (1000 of 18564 possible perms)
1 4: R = 0.9277  p = 0.0010 (1000 of 12376 possible perms)
2 3: R = 0.1596  p = 0.0890 (1000 of 646646 possible perms)
2 4: R = 0.7635  p = 0.0010 (1000 of 352716 possible perms)
3 4: R = 0.5584  p = 0.0010 (1000 of 1352078 possible perms)

r = 0.54128
p = 0.001

f_anosim2  Two-way Analysis of Similarity

Description

Two-way crossed ANOSIM with no replication

Usage

\[ [r,p] = f\_anosim2(dis,fac1,fac2,\{rank\},iter) \]

Arguments

dis  symmetric distance matrix
fac1  vector of integers (or chars) specifying levels of factor 1 for rows/cols of distance matrix
fac2  vector of integers (or chars) specifying levels of factor 2 for rows/cols of distance matrix
rank  optionally rank distances in xDis (default = 1)
iter  # iterations for permutatin test (default = 1000)

Details

This program handles missing data in a 2-way layout design that would occur when one (or more) of the treatment levels is missing from a block. The one-tailed permutation-based significance test permutes the treatment levels separately within each block.
This program has been tested against Clarke’s Primer 5 for Windows and gives the same results.
Please be patient when running this program with large datasets and/or high values for iter.

Value

The function returns the following values:

\[ r \text{ strength of treatment effect (averaged across all blocks)} \]
\[ p \text{ permutation-based significance test} \]

Note

fac1 and fac2 must be equal to row/col size of dis.
Author(s)
Dave Jones

References

See Also
f_anosim, f_npManova, f_mantel, and f_modelMatrix

Examples
For an example of a Two-Way ANOSIM with no replication load the file, anosim2.mat, in the data folder. This data is from Warwick (1971) and ships with Clarke’s PRIMER program. There is 1 response variable, dis, representing a Bray-Curtis symmetric distance matrix from 4th root transformed species abundances, and 2 factors: site and time.

>> load anosim2.mat
>> tic;[r,p] = f_anosim2(dis,site,time,1,1000);toc
(Please wait...running 1000 permutations for each of 2 factors.)

====================================================================
Results of 2-way crossed ANOSIM (no replication):
====================================================================

Strength of 'TIME' effect (averaged across all 'SITE' blocks):
R = 0.0555 p = 0.2190 (1000 iterations)
# permuted statistics greater than or equal to R = 218

Strength of 'SITE' effect (averaged across all 'TIME' blocks):
R = 0.3574 p = 0.0010 (1000 iterations)
# permuted statistics greater than or equal to R = 0

====================================================================
elapsed_time = 121.74

**f_bartlett**  
*Bartlett’s Test*

**Description**

Bartlett’s Test for Homogeneity of Variances

**Usage**

```matlab
[pval, chisq, df] = f_bartlett(x)
```

**Arguments**

- **x**  
  column vector of input data
- **grps**  
  column vector specifying group membership

**Details**

Under the null hypothesis of equal variances, the test statistic `chisq` approximately follows a chi-square distribution with `df` degrees of freedom; `pval` is the p-value (1 minus the CDF of this distribution at `chisq`) of the test.

This program has been tested against the SAS code `bartlett.sas` and gives similar results.

**Value**

The function returns the following values:

- **pval**  
  probability that the null hypothesis is true
- **chisq**  
  test statistic
- **df**  
  degrees of freedom

**Author(s)**

Original Octave code `bartlett_test.m` by KH (Kurt.Hornik@ci.tuwien.ac.at). Ported to Matlab by Dave Jones.
Examples

Load the file, bartlett.mat, in the data folder.

```matlab
>> load bartlett.mat
>> [pval, chisq, df] = f_bartlett(x, grps)
pval =
    0.034381
chisq =
   8.6464
df =
    3
```
**f_bioenv**  
*Biotic - Environmental Correlation*

**Description**
Correlation of primary (biotic) symmetric distance matrix with all possible subsets of secondary (environmental) matrix.

**Usage**

\[
[res, resLabels] = f_bioenv(dis, matrix, labels, metric, trim, out)
\]

**Arguments**
- `dis`: symmetric distance matrix
- `matrix`: 2\textsuperscript{o} matrix (rows = variables, cols = samples)
- `labels`: cell array of variable labels of 2\textsuperscript{o} matrix; e.g., `labels = {'temp', 'sal', 'depth', 'O2'}`
- `metric`: distance metric to use for 2\textsuperscript{o} matrix; 0 = Euclidean (default); 1 = Bray-Curtis
- `trim`: return only this many of the top Rho’s per subset size class; 0 = return all (default)
- `out`: send results to screen (= 1, default) or cell array with filename; e.g., `out = {'results.txt'}`

**Value**
The function returns the following values:
- `res`: cell array, 1st column = Rho, 2nd:end are variable indices
- `resLabels`: cell array of variable names

**Note**
This function requires `combnk.m` from the *Matlab Statistics Toolbox*. The code could be modified to utilize the freely available `choosenk.m` instead.

For `out`, an existing file with the same name as specified will be **deleted**. Tabulated results are also sent to the screen or file, depending on the value of `out`.

The # of rows of `dis` must equal the # columns of `matrix`. 
Author(s)

Dave Jones

References


See Also

f_braycurtis, f_euclid

Examples

Load the file, spiders.mat, in the data folder. This is data from van der Aart & Smeenk-Enserink (1975) and is Hunting spider abundances for 12 species (spiders) taken from 28 sites (site_labels) and associated environmental data (env).

```matlab
>> load spiders.mat
>> spiders2 = f_transform(spiders',3)'; % log-transform abundances
>> dis = f_brayCurtis(spiders2'); % Bray-Curtis dissimilarity matrix
>> [res,resLabels] = f_bioenv(dis,env',env_labels,0,5,1);
```

There are 63 possible subsets of 6 variables
Processing 6 subsets of 1 variables
Processing 15 subsets of 2 variables
Processing 20 subsets of 3 variables
Processing 15 subsets of 4 variables
Processing 6 subsets of 5 variables
Processing 1 subsets of 6 variables

==========================================
Rho Variables
==========================================

1
0.7037 water
0.5486 light
0.5197 sand
0.4962 cover
0.4097 twigs
0.2611 herbs
2
0.7591 twigs water
0.7578 light water
0.7115 light sand
0.7073 cover water
0.7018 sand water

3
0.8110 sand twigs water
0.7870 light sand water
0.7561 cover light sand
0.7505 cover sand water
0.7496 cover twigs water

4
0.8159 cover sand twigs water
0.7995 cover light sand water
0.7982 light sand twigs water
0.7845 cover herbs sand water
0.7751 herbs light sand water

5
0.8065 cover light sand twigs water
0.8061 cover herbs light sand water
0.8047 cover herbs sand twigs water
0.7723 herbs light sand twigs water
0.7270 cover herbs light twigs water

6
0.7996 cover herbs light sand twigs water

**f_biplot**

*Eigenvecotor-based 2d distance biplot*

**Description**

This function is a general function used to distance biplot based on eigenvectors. Vectors are plotted for each variable comprising the original data matrix for which an eigen-analysis was performed. The direction each vector points indicates the direction of increase of each variable (i.e., gradient). Length of vectors indicate the relative contribution each variable contributes to the formation of the reduced space plotted (e.g., that space defined by the 1st 2 Principal Component axes).

**Usage**

```r
def f_biplotPca2(evects, scale, offset, sLabels);
```

**Arguments**

- **evects**: eigenvectors from f_pca
- **scale**: scaling factor for vectors (default =1)
- **offset**: label offset (default =0)
- **sLabels**: cell array of vector labels (if empty, autocreate); e.g., `sLabels = {'sal' 'tmp' 'elev'}`;

**Author(s)**

Dave Jones

**References**


**See Also**

f_pca, f_biplotEnv2, f_biplotSpecies, f_vectorfit
f_biplotEnv2  

**Biplot for 2-d Ordination**

**Description**

Create a distance biplot consisting of a 2-d ordination with environmental correlation vectors.

**Usage**

```matlab
[biplot, Rsq] = f_biplotEnv2(crds, env, special, iter, scale, offset, sLabels)
```

**Arguments**

- **crds**: matrix of ordination coordinates (rows = sites; cols = dimensions)
- **env**: matrix of (transformed) environmental variables (rows = sites, cols = variables)
- **special**: type of correlation; 0 = Pearson’s, 1 = Spearman’s (default)
- **iter**: # of iterations for permutation test (default = 0)
- **scale**: scaling factor for env vectors (default = 1)
- **offset**: offset of labels in plot (default = 0)
- **sLabels**: cell array of vector labels; if empty they are autocreated e.g., ```sLabels = {'sal' 'tmp' 'elev'}```

**Value**

The function returns the following values:

- **biplot**: 2 column matrix specifying x-y coordinates of endpoints for scaled env vectors to overlay on ordination
- **Rsq**: column matrix of correlation with each axis (a permutation test is performed when iter>0)

**Author(s)**

Dave Jones

**References**

See Also

f_biplotEnv3, f_biplotSpecies, f_vectorfit
f_biplotEnv3  

_Biplot for 3-d Ordination_

_Description_

Create a distance biplot consisting of a 3-d ordination with environmental correlation vectors.

_Usage_

```matlab
[biplot, Rsq] = f_biplotEnv3(crds, env, special, iter, scale, offset, sLabels, plotflag, minP)
```

_Arguments_

- **crds**: matrix of ordination coordinates (rows = sites; cols = dimensions)
- **env**: matrix of (transformed) environmental variables (rows = sites, cols = variables)
- **special**: type of correlation; 0 = Pearson’s, 1 = Spearman’s (default)
- **iter**: # of iterations for permutation test (default = 0)
- **scale**: scaling factor for env vectors (default = 1)
- **offset**: offset of labels in plot (default = 0)
- **sLabels**: cell array of vector labels; if empty they are autocreated e.g., `sLabels = {'sal' 'tmp' 'elev'}`
- **minP**: if p-value of correlation > minP, then correlation is NOT used (default = 0.05)

_Value_

The function returns the following values:

- **biplot**: 3 column matrix specifying xyz coordinates of endpoints for scaled env vectors to overlay on ordination
- **Rsq**: column matrix of correlation with each axis (permutation-based significance provided when iter>0)

_Author(s)_

Dave Jones
References


See Also

f_biplotEnv2, f_biplotSpecies, f_vectorfit
f_biplotSpecies  \hspace{1em} Species Biplot

Description
Create species vectors for ordination distance biplot.

Usage
\[ \text{[biplot,rsq]} = \text{f\_biplotSpecies}(\text{crds,species,special,iter,scale,offset,sLabels}); \]

Arguments
- \text{crds} \hspace{1em} \text{matrix of ordination coordinates (rows = sites; cols = eigenvectors or dimensions)}
- \text{species} \hspace{1em} \text{matrix of (transformed) species abundances (rows = sites, cols = variables)}
- \text{special} \hspace{1em} \text{type of correlation; 0 = Pearson's (default), 1 = Spearman's}
- \text{iter} \hspace{1em} \text{# of iterations for permutation test (default = 0)}
- \text{scale} \hspace{1em} \text{scaling factor for \text{species} vectors (default = 1)}
- \text{offset} \hspace{1em} \text{offset of labels in plot (default = 0)}
- \text{sLabels} \hspace{1em} \text{cell array of species labels (if empty autocreate) e.g., sLabels = \{‘sp1’ ‘sp2’ ‘sp3’\}}

Value
The function returns the following values:
- \text{biplot} \hspace{1em} 2 column matrix specifying x-y coordinates of endpoints for scaled \text{species} vectors to overlay on ordination
- \text{Rsq} \hspace{1em} \text{column matrix of correlation with each axis (permutation-based significance provided when iter>0)}

Author(s)
Dave Jones
References


See Also

f_biplotEnv2, f_biplotEnv3, f_vectorfit
Description
This function calculates a Bray-Curtis symmetric dissimilarity matrix from an input data matrix specifying species abundances per sample site.

Usage

dist = f_braycurtis(X);

Arguments

X  species x site data matrix

Details

Value
The function returns the following values:

dist  symmetric dissimilarity matrix

Note
The input data matrix should be codes as species (rows) versus sites (columns). An $n \times n$ symmetric dissimilarity matrix will be returned where $n$ is the number of sites.

Author(s)

=======================================================================
Copyright (c) 1997 B. Planque - Sir Alister Hardy Foundation for Ocean Science <bp@wpo.nerc.ac.uk>
Permission is granted to modify and re-distribute this code in any manner as long as this notice is preserved.
All standard disclaimers apply.
=======================================================================

Slightly modified by Dave Jones after distance.m in the EDAT Toolbox to only calculate a Bray–Curtis distance matrix between columns.
See Also

f_euclid
f_brokenstick  

*Broken- Stick model*

Description

This function is used to determine the # of significant ordination dimensions via the *Broken-Stick* model.

Usage

    f_brokenstick(nvars)

Arguments

   nvars     # of variables (e.g., sample sites)

Value

The function returns the following values:

- `varExplained`  % variance explained

Author(s)

Original Matlab code, `brokestk.m`, by R. E. Strauss, modified by Dave Jones.

References


See Also

    f_pca
Canonical Analysis of PCoA

Description
The function performs a Canonical Analysis of Principal Coordinates using any distance (dissimilarity) matrix.

Usage
[crds,trc,ccor,H,p1,p2,centroids] = f_cva(yDis,x,rank,iter,plt,verb)

Arguments
yDis  square symmetric distance matrix derived from response variables
x  (1) vector of integers specifying group membership for objects in yDis, (2) ANOVA design matrix specified by dummy coding, or (3) matrix of explanatory variables (rows = observations, cols = variables)
rank  optionally rank distances in yDis (default = 0)
iter  # iterations for permutation test (default = 0)
plt  optionally plot results (default = 1)
verb  optionally send results to display (default = 1)

Value
The function returns the following values:

crds  coordinates of canonical axes (= Qstar)
trc  trace statistic
ccor  canonical eigenvalues (1st value is greatest root statistic)
p1  randomized probability of trace statistic
p2  randomized probability of greatest root statistic
centroids  centroids of groups defined in x
Note

This program performs nonparametric Multiple Discriminant Analysis on any symmetric distance matrix when the input for x is (1) a vector specifying group membership. It performs generalized Canonical Variates Analysis when x is (2) an ANOVA design matrix or (3) a matrix of explanatory variables.

Use f_designMatrix to create an ANOVA design matrix for input as x; the matrix should be full rank (not singular) and do not include an intercept term (a column of all 1’s).

The program asks the user to specify how many axes of Q to retain for the analysis (m). Examine the EIGENVALUES and % VARIATION EXPLAINED output in the command window and try to include as much information in Q as possible with as few axes as possible.

This program has been tested against the numerical example in Legendre & Legendre (1998:p.626) and provides similar output. The % variation explained for each canonical axis, calculated by f_cap, are almost identical to that computed by manova1.m in the Matlab Statistical Toolbox (version 3) for the Iris data.

This program requires ortha.m by Andrew Knyazev <knyazev@na-net.ornl.gov> & Rico Argentati, which is included the present toolbox and available from:
http://www-math.cudenver.edu/~aknyazev/software/MATLAB/

Author(s)

Dave Jones

References

http://www.stat.auckland.ac.nz/PEOPLE/marti/


See Also

f_cda,f_npManova, f_anosim, f_anosim2, f_mantel, f_designMatrix
Examples

The example dataset, iris.mat, is Fisher’s Iris data and can be found in the data folder.

```matlab
>> load iris.mat
>> dis = f_euclid(iris);
>> [crds,trc,ccor,H,p1,p2,centroids] = f_cap(dis,grps,0,1000,1,1);
```

---

**Matrix Q:**

<table>
<thead>
<tr>
<th>Eigenvalues</th>
<th>% Explained</th>
<th>Axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>630.008</td>
<td>92.46187</td>
<td>1</td>
</tr>
<tr>
<td>36.15794</td>
<td>97.76852</td>
<td>2</td>
</tr>
<tr>
<td>11.65322</td>
<td>99.47878</td>
<td>3</td>
</tr>
<tr>
<td>3.551429</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>4.391243e-013</td>
<td>100</td>
<td>5</td>
</tr>
<tr>
<td>1.034584e-016</td>
<td>100</td>
<td>126</td>
</tr>
</tbody>
</table>

---

Specify # of Axes in Q to retain (1-126) ? 2

Permuting the data 999 times...

---

**Nonparametric Canonical Discriminant Analysis:**

- Trace Stat = 368.1700 p = 0.00100
- Greatest Root = 366.1265 p = 0.00100
- No. of permutations = 1000

No. of axes of Q used (m) = 2

Canonical Correlations:

```
366.1265  2.0435
```
Figure 1: Example of $f_{\text{cap}}$
f_cda  

*Canonical Discriminant Analysis*

**Description**

This function is used to perform a *classical* Canonical Discriminant Analysis.

**Usage**

```matlab
[scores,centroids,Cvects] = f_cda(x,y,method,pflag)
```

**Arguments**

- **x**: input data (rows = objects, cols = variables)
- **y**: column vector of integers specifying group membership
- **method**: center (=1, default) or standardize (=2)
- **pflag**: make canonical plot (default = 0)

**Details**

- **method** = 1: variables comprising matrix x are centered, so columns of the canonical eigenvectors (Cvects) are *Identification Functions*. Use this method when you plan to place new objects in the canonical space.
- **method** = 2: variables comprising matrix x are standardized, so columns of the canonical eigenvectors (Cvects) are *Discriminant Functions*. Use this method to assess the relative importance of the original variables of x in discriminating groups.

**Value**

The function returns the following values:

- **scores**: coordinates in canonical space (= canonical variates)
- **centroids**: group centers
- **cvects**: canonical eigenvectors
**Note**

*scores* are the coordinates of the original (centered, or standardized) data projected in new canonical space. They are obtained by multiplying the original data by the Canonical Eigenvectors.

*centroids* are the coordinates of the group means projected in the new canonical space.

*Cvects* (Canonical Eigenvectors) are the normalized orthogonal eigenvectors defining the canonical space of the discriminant analysis.

**Author(s)**

Dave Jones

**References**


**See Also**

*f_cap*

**Examples**

The example dataset, *iris.mat*, is Fisher’s Iris data and can be found in the *data* folder

```
>> load iris.mat
>> [scores,centroids,Cvects] = f_cda(iris,grps,2,1);
```
Figure 2: Example of f_cda
**f_figArea**  \hspace{1cm} \textit{Area of Figure Objects}

**Description**

This function is used to compute the Area of figure objects in a plot and optionally returns a list of handles sorted ascending by Area.

**Usage**

\[
\text{[area,h2]} = \text{f_figArea}(h);
\]

**Arguments**

- \( h \)  
  handle of figure object(s)

**Value**

The function returns the following values:

- \( \text{area} \)  
  area for object specified by \( h \)
- \( h2 \)  
  figure handles sorted by area (ascending)

**Note**

This is particularly useful when you’ve plotted a number of patches but, because of the stacking order, the smaller ones are obscured by the larger.

**Author(s)**

Dave Jones

**Examples**

After creating a plot with potentially overlapping patches, run the following commands:

\[
\begin{align*}
  h &= \text{get(gca,'Children')}; \quad \% \text{get stacking order of handles} \\
  [\text{null},h2] &= \text{f_figArea}(h); \quad \% \text{sort by area} \\
  \text{set(gca,'Children',h2);} & \quad \% \text{re-stack}
\end{align*}
\]
f_importSurfer  

Import Surfer Blanking File

Description
This program is used to import a Golden Software’s *Surfer for Windows* bln file into Matlab in a format that can be used by the M_Map Toolbox.

Usage

\[ [\text{ncst},k,\text{Area}] = \text{f_importSurfer}('\text{fname}') ; \]

Arguments

\text{fname}  
name of Surfer *.bln blanking file

Details
Be sure to save variables \text{ncst}, k, and \text{Area} as a usercoast.mat file.

Value
The function returns the following values:

\text{ncst},k,\text{Area}  
required components for M_Map usercoast.mat file

Note
Since Surfer can import ArcView Shape files, this is a good way to get shp data into Matlab. The suggested procedure is to turn off the display of axes in Surfer, then export the data as a bln file (select the option *Break Apart Compound Areas*), import using \text{f importSurfer}, save as a usercoast file, and plot using \text{m_usercoast}.m.

Author(s)
Dave Jones

References
Portions of this code are from the comments in \text{mu_coast}.m from Rich Pawłowicz’s <rich@ocgy.ubc.ca> M_Map Toolbox available from:

\url{http://www2.ocgy.ubc.ca/ rich/map.html}
Examples

Example code to plot as filled patches using M_Map toolbox:

```matlab
>> m_proj('mercator','longitudes',[-81 -80],'latitudes',[24.5 25.5]);
>> m_usercoast('fmri.mat','patch',[0 0 0],'edgecolor','none');
>> m_grid('box','fancy','fontsize',8,'linestyle','none','xtick',[-81:-80],'ytick',[24.5:25.5]);
```
Description

This program performs Multiple Linear Regression using Least Squares Estimation via Matlab’s QR factorization.

Usage

\[
[F,t,R2,yfit,b,resid] = f\_mregress(x,y,iter,perm,verb);
\]

Arguments

- \(x\): matrix of independent variables (column-wise)
- \(y\): column vector of dependent variable
- \(iter\): # of iterations for permutation test (default = 0)
- \(perm\): permute residuals instead of raw data (default = 1)
- \(verb\): verbose output of results to display (default = 1)

Details

This function solves the equation such that:
\[
y = b(0) + b(1)\*X(:,1)) + b(2)\*X(:,2))...+ b(k)\*X(:,k))
\]
where \(k\) = # of predictor variables.

Value

The function returns the following values (\(F\) and \(t\) are structures):

- \(F\).stat: F-statistic
- \(F\).para_p: parametric p-value for 1-tailed test of F
- \(F\).perm_p: permutation p-value for 1-tailed test of F
- \(t\).stat: t-statistic for partial regression coefficients
- \(t\).para_p: parametric p-value for 1-tailed test of t
- \(t\).perm_p: permutation p-value for 1-tailed test of t
- \(R2\): coefficient of multiple determination (\(R^2\), goodness-of-fit)
- \(yfit\): fitted values of \(y\)
- \(b\): regression coefficients (1st value is the y-intercept)
- \(resid\): residuals
Note

The regression coefficients are computed using Least Squares Estimation (via the \ operator), which is preferred over methods that require taking the inverse of a matrix. R2, the coefficient of multiple determination, is a measure of goodness-of-fit and gives the proportion of variance of Y explained by X.

Parametric (and optional permutation) tests of significance for the F- and t-statistics are performed. The permutation test is conducted when iter > 0 and allows for permutation of either the raw data or the residuals of the full regression model. Permutation of the raw data involves random permutation of the rows (= observations) of Y relative to the rows of X. The permutation test is preferred over the parametric test when the data are non-normal. Permutation of the residuals (vs. the raw data) is preferred when data have extreme values (i.e., outliers).

This function has been tested against Legendre & Casgrain’s regressn.exe program and gives similar output.

Calculation of parametric p-values for F and t require fpdf and tcdf from the Matlab Statistics Toolbox; these could be replaced by df and dt from the free Stixbox Toolbox.

Author(s)

Dave Jones with help from news://comp.soft-sys.matlab

References


Examples

The example dataset, mregress.mat, can be found in the data folder and is an excerpt from Table 16.1 of Sokal & Rohlf (1995).

>> load mregress.mat
>> tic;
>> [F,t,yfit,coefs,resid] = f_mregress(x,y,10000);
>> toc;
Permuting the data 9999 times...

Multiple Linear Regression via QR Factorization:

<table>
<thead>
<tr>
<th>R2</th>
<th>F-stat</th>
<th>parametric-p</th>
<th>permutation-p</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.51611</td>
<td>20.2653</td>
<td>0.00000</td>
<td>0.00010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>b</th>
<th>t-stat</th>
<th>parametric-p</th>
<th>permutation-p</th>
</tr>
</thead>
<tbody>
<tr>
<td>intercept</td>
<td>77.23671</td>
<td>3.59122</td>
<td>0.00045</td>
<td>0.00060</td>
</tr>
<tr>
<td>1</td>
<td>-1.04805</td>
<td>-2.80777</td>
<td>0.00384</td>
<td>0.00220</td>
</tr>
<tr>
<td>2</td>
<td>0.02430</td>
<td>5.07607</td>
<td>0.00000</td>
<td>0.00050</td>
</tr>
</tbody>
</table>

# permutations of residuals = 9999
All significance tests are One-Tailed

elapsed_time = 57.092
f_mst  Minimum Spanning Tree

Description
This function uses Kruskal’s algorithm to calculate a Minimum Spanning Tree for objects based on pair-wise distances specified in a symmetric distance matrix.

Usage
[mst,branch,scaled] = f_mst(dis,crds)

Arguments
dis  symmetric distance matrix
crds  Euclidean coordinates of an ordination based on dis

Value
The function returns the following values:
mst  pairs of objects connected by MST branches
branch  corresponding branch lengths
scaled  branch lengths rescaled to 0–100

Note
MST’s are particularly useful for checking or interpreting the results of an ordination by overlying a MST on an ordination plot.

Author(s)
Dave Jones

Examples
Load the file, spiders.mat, in the data folder. This is data from van der Aart & Smeenk-Enserink (1975) and is Hunting spider abundances for 12 species (spiders) taken from 28 sites (site_labels) and associated environmental data (env).

>> load spiders.mat
>> dis = f_braycurtis(spiders');
>> evects = f_pcoa(dis,0,0,0);
>> mst = f_mst(dis,evects);
f_nmds Non-metric Multidimensional Scaling

Description
This function is used to run Mark Steyvers’ Non-metric Multidimensional Scaling program.

Usage
config = f_nmds(dist, ndims, initial, plotflag, maxiter, conv, rotate);

Arguments
- dist: symmetric dissimilarity matrix
- ndims: number of dimensions of solution (default = 2)
- initial: initial config: 1 = Torgeson-Young scaling (default); 0 = random
- plotflag: plot results (default = 1)
- maxiter: max # iterations (default = 50)
- conv: convergence criterion (default = 0.001)
- rotate: rotate final configuration to Principal Coordinates (default = 1)

Value
The function returns the following values (config is a structure):
- config.mds: configuration of solution in (ndims) dimensions
- config.stress: final stress of solution
- config.dim: # dimensions of solution
- config.rsq: Mantel statistic comparing fitted distances with original dissimilarities

Note
This program requires the Matlab Optimization Toolbox and Mark Steyvers’ Nonmetric Scaling Toolbox, available from:

http://www-psych.stanford.edu/~msteyver/programs\us data/mdszip.zip
I've been able to obtain best results with this program by editing Steyver's mds.m file and changing `randn('state',seed)` to `rand('state',seed)`. This allows you to draw from uniformly distributed random numbers for initial configurations rather than normally distributed random numbers.

**Author(s)**

Modified after Mark Steyvers' original Matlab code, domds.m, by Dave Jones. Added support for variable random seed, rotate to PCA, Mantel statistic, and formatting of output.

**See Also**

f_pcoa, f_pca

**Examples**

Load the file, spiders.mat, in the data folder. This is data from van der Aart & Smeenk-Enserink (1975) and is Hunting spider abundances for 12 species (spiders) taken from 28 sites (site_labels) and associated environmental data (env).


```matlab
>> load spiders.mat
>> dis = f_braycurtis(spiders');
>> config = f_nmds(dis,2,0,0,50,0.001,1);
```

Values added to satisfy distance axioms 0 0
Minkowski metric r value 2
Number of objects in matrix 28
Number of useful similarity ratings 378
Number of dimensions: 2

Maximum number of iterations 50
Convergence Criterion: 0.001
Minimize function: stress1
Starting configuration is: random (seed=213210)

The following results apply to training file only:
(Rs is the rank order correlation coefficient between observed and predicted dissimilarities)

\[ t=0 \quad \text{Stress1}=0.42659 \quad \text{Stress2}=0.99278 \quad Rs=0.02823 \]
t=1 Stress1=0.60586 Stress2=0.99538 Rs=-0.05871

\( t=2 \) Stress1=0.40898 Stress2=0.98782 Rs=-0.03861

\( t=3 \) Stress1=0.38038 Stress2=0.96809 Rs=0.02299

\( t=4 \) Stress1=0.36543 Stress2=0.92466 Rs=0.13417

\( t=5 \) Stress1=0.33233 Stress2=0.82448 Rs=0.34933

\( t=6 \) Stress1=0.27412 Stress2=0.66189 Rs=0.62448

\( t=7 \) Stress1=0.18919 Stress2=0.43455 Rs=0.83975

\( t=8 \) Stress1=0.15189 Stress2=0.33059 Rs=0.91154

\( t=9 \) Stress1=0.13544 Stress2=0.28669 Rs=0.94157

\( t=10 \) Stress1=0.09710 Stress2=0.20584 Rs=0.97019

\( t=11 \) Stress1=0.09284 Stress2=0.19665 Rs=0.97251

\( t=12 \) Stress1=0.09224 Stress2=0.19559 Rs=0.97280

End of simulation - reason: convergence criterion reached

\[
\begin{align*}
\text{>> plot}(\text{config.mds(:,1)},\text{config.mds(:,2)},'bo'); \\
\text{>> title}([['\textbf{Nonmetric MDS (Stress = '}] \text{num2str(config.stress)} ['')]])]; \\
\text{>> xlabel}('\text{Dim 1}'); \\
\text{>> ylabel}('\text{Dim 2}');
\end{align*}
\]
Nonmetric MDS (Stress = 0.092239)

Figure 4: Example of \texttt{f_nmds}
Description

This program performs a nonparametric, permutation-based MANOVA on any symmetric distance (or dissimilarity) matrix.

Usage

result = f_npManova(yDis,x,type,rank,iter,pw,verb)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>yDis</td>
<td>square symmetric distance matrix derived from response variable(s)</td>
</tr>
<tr>
<td>x</td>
<td>matrix of integers specifying group membership for objects in yDis (column-wise) or a matrix of explanatory variables</td>
</tr>
<tr>
<td>type</td>
<td>x specifies group membership or factor levels (default = 0); x is a matrix of explanatory variables (=1)</td>
</tr>
<tr>
<td>rank</td>
<td>optionally rank distances in yDis (default = 0)</td>
</tr>
<tr>
<td>iter</td>
<td># iterations for permutation test (default = 0)</td>
</tr>
<tr>
<td>verb</td>
<td>optionally send results to display (default = 0)</td>
</tr>
</tbody>
</table>

Details

The permutation test in this program currently only supports permutation of the raw data.

Value

This function returns the following values (result is a structure):

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result.so</td>
<td>source of variation</td>
</tr>
<tr>
<td>result.df</td>
<td>degrees of freedom</td>
</tr>
<tr>
<td>result.SS</td>
<td>sum of squares</td>
</tr>
<tr>
<td>result.MS</td>
<td>mean square</td>
</tr>
<tr>
<td>result.F</td>
<td>F-statistics</td>
</tr>
<tr>
<td>result.p</td>
<td>permutation-based significance probabilities</td>
</tr>
</tbody>
</table>
Note

Special care must be taken when coding levels of nested factors, i.e. treatment levels of a nested factor must not be repeated across different levels of the main factor. For example use: main factor = [1 1 1 2 2 2 3 3 3]' nested factor = [1 2 3 4 5 6 7 8 9]'

instead of: nested factor = [1 2 3 1 2 3 1 2 3]'

More examples can be found in Appendix A.

This program takes a regression approach to ANOVA using the General Linear Model (GLM) and constructs F-ratios using the unrestricted form of the model. The F-ratios used for each type of test are provided in Appendix B of the User’s Manual. Some of these differ somewhat from those used in textbook examples, especially for balanced, mixed-model designs, but are the same as those used in most computer programs that use GLM (e.g., SAS and MINITAB).

To perform a classical (M)ANOVA use a symmetric Euclidean distance matrix as input for the response variable. The real power of this function, however, comes from its ability to use, say, a Bray-Curtis distance matrix derived from species abundances from a community ecology study.

After determining a significant factor effect, you may wish to use fnpManovaPW to perform a posteriori multiple comparison tests.

Author(s)

Dave Jones

References

http://www.stat.auckland.ac.nz/PEOPLE/marti/

http://www.stat.auckland.ac.nz/PEOPLE/marti/


See Also

f_npManovaPW, f_cap, f_anosim, f_anosim2, Appendix A, Appendix B

Examples

For an example of a One-Way Model I Anova with replication (balanced design) load the file, sr_09p5.mat, in the data folder. This is data from Box 9.5 of Sokal & Rohlf (1995) and has 1 response variable, age, and 1 fixed factor: clone.

>> load sr_09p5.mat
>> dis = f_euclid(age');
>> result = f_npManova(dis,[clone],0,0,1000,1);

Permuting the data 999 times...

=================================================================================
Nonparametric (Permutation-based) MANOVA:
=================================================================================

<table>
<thead>
<tr>
<th>'Source'</th>
<th>'df'</th>
<th>'SS'</th>
<th>'MS'</th>
<th>'F'</th>
<th>'p'</th>
</tr>
</thead>
<tbody>
<tr>
<td>'factor 1'</td>
<td>[ 1]</td>
<td>[0.0064286]</td>
<td>[0.0064286]</td>
<td>[0.014063]</td>
<td>[0.872]</td>
</tr>
<tr>
<td>'residual'</td>
<td>[12]</td>
<td>[5.4857]</td>
<td>[0.45714]</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>'total'</td>
<td>[13]</td>
<td>[5.4921]</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
</tbody>
</table>

# iterations = 1000

(Note: NaNs are placeholders for the ANOVA table)

(Data with replication)

See Appendix A for more examples.
Description

This function is used to perform PCA using either the covariance or the correlation matrix of the input data. Use the covariance matrix when the variables are of the same kind, type, and scale; otherwise, use the correlation matrix.

Usage

\[ [\text{scores}, \text{evects}, \text{evals}, \text{expl}] = \text{f_pca}(x, \text{pflag}, \text{method}); \]

Arguments

- \( x \): data matrix (rows = objects, cols = variables)
- \( \text{pflag} \): no plot (=0, default); scores (=1); scores +scree (=2); scores, scree, + equilibrium circle (=3)
- \( \text{method} \): use covariance (=1, default) or correlation matrix (=2)

Details

Principal Component scores are the coordinates of the objects from the input data matrix in the new space defined by the Principal Component Axes. The eigenvectors are the loadings of the original variables which, when multiplied by the original data, yield the Principal Component scores. The matrix of eigenvectors has a row for each variable of the original data matrix and a column for each Principal Component Axis. The eigenvalues give the variance of the scores along each Principal Component Axis.

The SCREE PLOT is a graphical method of evaluating how many PC Axes you need to retain in order to adequately represent the variation in the original data matrix—it provides the same information as expl.

The EQUILIBRIUM CIRCLE provides a graphical method of determining the relative contribution each variable makes to the formation of the reduced spaced defined by PC Axis I & II. Only variables whose vectors extend to or beyond the radius of the Equilibrium Circle are considered to have made a significant contribution to that reduced space.
Value

The function returns the following values:

- **scores**: coordinates of objects on each PC axis
- **evecs**: eigenvectors (= loadings)
- **evals**: eigenvalues for each PC axis
- **expl**: % percent variation explained by each PC axis (row 1); cumulative % variation explained (row 2)

Note

This function implements PCA via Singular Value Decomposition of an association matrix formed from the original data. This matrix is decomposed into object space (U), variable space (V), and a diagonal matrix with singular variables along its diagonal (D) such that [association matrix] = [U*D*V], where eigenvectors = U and eigenvalues = the diagonal elements along D.

Author(s)

Dave Jones

References


See Also

- `f_biplot`
- `f_nmds`
- `f_pcoa`
- `f_cap`

Examples

Load the file, `spiders.mat`, in the `data` folder. This is data from van der Aart & Smeenk-Enserink (1975) and is Hunting spider abundances for 12 species (`spiders`) taken from 28 sites (`site_labels`) and associated environmental data (`env`).


```matlab
>> load spiders.mat
>> f_pca(env,3,2);
```
Figure 5: Example of f_pca plot
Figure 6: Example of f pca Scree plot
Figure 7: Example of f_pca Equilibrium Circle plot
**f_procrustes**  
*Procrustes Analysis*

**Description**

Generalized orthogonal Procrustes rotation of matrix Y to X, maximizing similarity and minimizing sum-of-squared distances.

**Usage**

```
[Xscl,Yrot,m2,resid,prob] = f_procrustes(X,Y,stnd,iter,plotflag)
```

**Arguments**

- `X`  
  reference matrix (rows = observations, cols = variables)
- `Y`  
  target matrix to rotate
- `stnd`  
  standardize variables (default = 0)
- `iter`  
  # iterations for permutation test (default = 0)
- `plotflag`  
  plot fitted results (default = 0)

**Details**

This function performs an orthogonal least-squares Procrustes analysis on 2 rectangular data matrices (X & Y) by minimizing the sum-of-squared distances between corresponding elements of the 2 matrices. This is done by translating, scaling, mirroring, and rotating Y to fit X. The symmetric orthogonal Procrustes statistic ($m^2$) is a measure of goodness-of-fit of Y to X after rotation and provides the residual sum-of-squares, which varies from 0 to 1. Smaller values of $m^2$ indicate better fit.

**Value**

The function returns the following values (resid is a structure):

- `Xscl`  
  centered & scaled form of X
- `Yrot`  
  centered, scaled, & rotated form of Y
- `m2`  
  symmetric Procrustes statistic, ranges from 0–1 (smaller values indicate better fit)
- `resid`  
  structure of residuals (*.dim, *.obs, *.sse)
- `prob`  
  permutation-based significance of m2

51
Note

If the # of variables (columns) in X < Y, it is padded with 0’s and allows one to, say, rotate an ordination to an environmental variable.

When std = 1 each variable is standardized to mean 0 and variance 1 so they will contribute equal weight to the fitting process. This, however, may distort the final configurations.

An optional permutation-based significance test of $m^2$ is performed when iter > 0 to assess the statistical concordance between X & Y.

resid.dim provides the length of each observation along each dimension after rotation. This allows interpretation of the direction of increase when Y codes for, say, an environmental gradient.

resid.obs provides the total length of each observation, which is a measure of goodness-of-fit for each observation; (smaller values = better fit).

resid.sse (like $m^2$) is a measure of total concordance between X and Y (smaller values = better fit);

Author(s)

Dave Jones

References


http://cc.oulu.fi/~jarioksa/softhelp/vegan.html


See Also

f_mantel, f_bioenv

Examples

The first example follows Example 1 of Peres-Neto & Jackson (2001) and uses data from Table 1 of Losos (1990). Load the file, losos.mat, in the data folder.


```matlab
>> load losos.mat

% log transform variables:
>> morph_log = f_transform(morph,3);
>> perform_log = f_transform(perform,3);

% PCA on correlation matrix:
>> [morph_scores,morph_evects] = f_pca(morph_log,0,2);
>> [perform_scores,perform_evects] = f_pca(perform_log,0,2);

% scale variance of scores on each axis = 1:
>> morph_scores = f_transform(morph_scores',7)';

% Procrustes Analysis:
>> [morph_scl,perform_scl,m2,resid,prob] = f_procrustes(morph_scores(:,1:2),...
  perform_scores(:,1:2),1,1000,1);

Permuting the data 999 times...

>> prob

prob = 0.0030

% add labels to Superimposition plot:
>> f_labelplot([morph_scl(:,1) morph_scl(:,2)],sLabels);

% directions of variation:
>> figure;
>> f_biplotPca2(morph_evects,1,0,mLabels,0);
>> f_biplotPca2(perform_evects(:,1:2)*H,1,0,pLabels,0);
>> box on;
```
Figure 8: Example of \texttt{procrustes}
Figure 9: Example of the procrustes analysis.
f_shadeBox \hspace{1cm} Shade subsets of a time series plot

Description

This function is used to shade subsets of a time series plot in order to highlight specific time periods. It should be called after creating a time series plot and should use the same scaling factor.

Usage

f_shadebox(region)

Arguments

region \hspace{1cm} 2-d matrix defining regions along the Y-axis to shade (column 1 = start, column 2 = stop)

scale \hspace{1cm} scaling factor used in time series plot (default = 1)

Details

region specifies a variable currently loaded in the Matlab workspace.
'region' specifies a space-delimited ASCII file in the Matlab path.

Value

The function modifies the current figure.

Author(s)

Dave Jones

Examples

Load the file, shadeBox.mat, in the data folder.

```
>> load shadeBox.mat
>> plot(time,speed,'b-');
>> f_shadeBox(region);
```
Figure 10: Example of f\_shadeBox
f_smooth  

Moving Average Smoothing Filter

Description

This function is used to smooth columns of a matrix via a Moving Average. It utilizes the Matlab function smooth from the Curve Fitting Toolbox. Each column of data is smoothed separately but using the same span.

Usage

\[ y = f_{\text{smooth}}(x, \text{span}, \text{rep}); \]

Arguments

- \( x \) input matrix
- \( \text{span} \) size of filter (default = 5), should be odd
- \( \text{rep} \) # of times to repeat filter

Details

Blocks of data separated by rows of NaN’s are smoothed separately. The smooth function differs from similar filters, such as ones that utilize the Matlab function filter, as it preserves the endpoints of the data set.

Using the repeat option (rep) to run a smaller-sized filter multiple times may provide better results than running a larger-sized filter once.

Value

The function returns the following values:

- \( y \) smoothed data

Note

This function requires the Matlab Curve Fitting Toolbox.

Author(s)

Dave Jones
Examples

Load the file, smooth.mat, in the data folder. This is synthetic data representing, say, the geographical coordinates of the shorelines of two islands. The coordinates of the two islands are separated from one another in the matrix dat by a row of NaN's, thus they are smoothed independently.

```matlab
>> load smooth.mat
% smooth twice with a span of 5:
>> datsm = f_smooth(dat,5,2);
>> plot(dat(:,1),dat(:,2),'r-',datsm(:,1),datsm(:,2),'b-');
>> axis equal;box on;
>> legend('Original','Smoothed',2);
>> title('\bfSmoothing via Moving Average');
```
Figure 11: Example of $f_{\text{smooth}}$
**f_vecAngle**

*Counter-clockwise angle between 2 points*

**Description**

This function is used to determine the counter-clockwise angle (in degrees) between points A and B.

**Usage**

\[ \theta = f_{\text{vecAngle}}(a,b); \]

**Arguments**

- **a**: 2-d row vector for point a (= \([x_a \ y_a]\))
- **b**: 2-d row vector for point b (= \([x_b \ y_b]\))

**Value**

The function returns the following values:

- \(\theta\) angle between a & b in degrees

**Note**

The points are considered to be Cartesian coordinates of the heads of vectors starting at the origin.

Note that the counter-clockwise angle from (A -> B) is not necessarily equal to that from (B -> A).

This function is vectorized, so A and B may each be 2-d matrices specifying multiple pairs of points.

**Author(s)**

Dave Jones

**References**


[http://www.geocities.com/SiliconValley/2151/win95gpe.html](http://www.geocities.com/SiliconValley/2151/win95gpe.html)
See Also

f_vecMagDir, f_vecTrans, f_vecUV

Examples

>> a = [0 10; -10 -10];
>> b = [-10 -10; 0 10];
>> theta = f_vecAngle(a,b)
theta =
 135
 225
f_vecDiagram  Progressive Vector Diagrams

Description

This function is used to create Progressive Vector Diagrams from time series data of wind or moored current meter velocity vectors. This type of diagram is used to produce a Lagrangian display of Eulerian measurements.

Usage

f_vecDiagram(u,v,units)

Arguments

u,v  unrotated vector components
units  m/s (=1) or cm/s (=2) (default = 0)

Value

The function creates a new figure

Note

units is an optional parameter that allows calculation of the spatial units in the plot. A velocity vector specifying 1 m/s covers 3.6 km/hr (there is 3600 sec in an hour).

Author(s)

Dave Jones

See Also

f_vecPlot

Examples

Load the file vecPlot.mat from the data folder.

>> load vecPlot.mat
>> f_vecDiagram(u,v,1);
Figure 12: Example of f_vecPlot
**Description**

This function is used to obtain Polar coordinates (magnitude & direction) of a vector given its Cartesian coordinates (U & V vector components). The direction is the counter-clockwise angle of rotation.

**Usage**

\[
[mag, dir] = f\_vecMagDir(u, v);
\]

**Arguments**

\[ u, v \]

column vectors of Cartesian coordinates of heads of vector components

**Details**

The program uses the Matlab function ATAN2 which relies on the sign of both input arguments to determine the quadrant of the result.

**Value**

The function returns the following values:

\[ mag \]

length of vector

\[ dir \]

angle of rotation (in degrees between 0-360)

**Author(s)**

Dave Jones

**See Also**

f_vecAngle, f_vecTrans, f_vecUV,
Examples

```matlab
>> u = [0 -10]';
>> v = [10 -10]';
>> [mag,dir] = f_vecMagDir(u,v)
mag =
    10
   14.142

dir =
  90
 225
```
f_vecPlot  

Plot time series of velocity vectors

Description

This function is used to plot time series of wind or current meter velocity vectors using Matlab's quiver function.

Usage

f_vecPlot(jdate,u,v,scale,units,jRange);

Arguments

jdate  column vector of Julian dates  
u,v  corresponding vector components  
scale  scale factor (default = 1)  
units  Y-axis label; e.g., units = 'm/s' (default = none)  
jRange  limits of dates to plot; e.g., jRange = [min max] (default = auto)

Value

The function produces a new figure;

Note

This function is necessary in order to obtain vectors that have the proper length and angle of rotation. An optional scaling factor can be applied allowing the user control over the amount of overlap among vectors and/or the scaling of vectors relative to the overall time series. The X-axis is scaled accordingly. The Y-axis allows easy, visual interpretation of vector length.

U,V components of velocity vectors can be extracted from data specifying only Speed and Direction using f_vecUV.

Author(s)

Dave Jones

See Also

f_julian, f_vecUV, f_shadeBox
Figure 13: Example of f_vecPlot

Examples

Load the file vecPlot.mat from the data folder.

```
>> load vecPlot.mat
>> f_vecPlot(date,u,v,20,'tau',[140 182]);
>> axis tight;
>> f_shadeBox(subsets,20);
```
f_vecTrans  Transform 2-d vector coordinates

Description
This function is used to rotate, translate, and/or scale 2-dimensional Cartesian coordinates. If the input coordinates specify the heads of (velocity) vectors, they may additionally be converted to unit length.

Usage

\[ [tx,ty] = f\_vecTrans(x,y,\text{rot},\text{transl},\text{scale},\text{unit}); \]

Arguments

- \( \text{x,y} \): column vectors specifying coordinate pairs
- \( \text{rot} \): angle of rotation in degrees (default = 0)
- \( \text{transl} \): translation \([\text{dx dy}]\) (default = \([0 0]\))
- \( \text{scale} \): scaling factor \([\text{sx sy}]\) (default = \([1 1]\))
- \( \text{unit} \): convert vectors to unit length (default = 0)

Details
This function uses the following matrices for transforming coordinates:

Rotation matrix: \((\theta \text{ is in radians}):\)
\[
\begin{pmatrix}
\cos(\theta) & -\sin(\theta) & 0 \\
\sin(\theta) & \cos(\theta) & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

Translation matrix:
\[
\begin{pmatrix}
1 & 0 & tx \\
0 & 1 & ty \\
0 & 0 & 1
\end{pmatrix}
\]

Scaling matrix:
\[
\begin{pmatrix}
sx & 0 & 0 \\
0 & sy & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

69
Value

The function returns the following values:

\[ tx, ty \quad \text{transformed coordinates} \]

Note

X and Y coordinates can be translated or scaled asymmetrically if 2 values are specified for these parameters. If only 1 value is provided, coordinates are translated or scaled symmetrically.

Author(s)

Dave Jones

References


http://www.geocities.com/SiliconValley/2151/win95gpe.html

See Also

\[ f\_vecAngle, f\_vecMagDir, f\_vecTrans3d, f\_vecUV \]

Examples

\[
\begin{align*}
\gg & \quad x = [0 10]'; \\
\gg & \quad y = [-10 -10]'; \\
\gg & \quad [tx,ty] = f\_vecTrans(x,y,45,2,0.5,0) \\
& \quad tx = \\
& \quad \quad 3.5355 \\
& \quad \quad 7.0711 \\
& \quad ty = \\
& \quad \quad -0.70711 \\
& \quad \quad 2.8284 
\end{align*}
\]
f_vecTrans3d  

*Transform 3-d vector coordinates*

**Description**

This function is used to rotate, translate, and/or scale 3 dimensional Cartesian coordinates. If the input coordinates specify the heads of (velocity) vectors, they may additionally be converted to unit length.

**Usage**

\[
[t_{x}, t_{y}, t_{z}] = f\text{._vecTrans3d}(x, y, z, xrot, yrot, zrot, transl, scale, unit);
\]

**Arguments**

- *x*, *y*, *z*  
  column vectors specifying coordinate triplets
- *xrot*  
  rotation about X-axis in degrees (default = 0)
- *yrot*  
  rotation about Y-axis in degrees (default = 0)
- *zrot*  
  rotation about Z-axis in degrees (default = 0)
- *transl*  
  translation \([dx \ dy \ dz]\) (default = \([0 \ 0 \ 0]\))
- *scale*  
  scaling factor \([sx \ sy \ sz]\) (default = \([1 \ 1 \ 1]\))
- *unit*  
  convert vectors to unit length (default = 0)

**Details**

This function uses the following matrices for transforming coordinates:

**Rotation Matrices** \((\theta)\) is in radians:

**X-axis:**

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & \cos(\theta) & -\sin(\theta) & 0 \\
0 & \sin(\theta) & \cos(\theta) & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

**Y-axis:**

\[
\begin{pmatrix}
\cos(\theta) & 0 & \sin(\theta) & 0 \\
0 & 1 & 0 & 0 \\
-\sin(\theta) & 0 & \cos(\theta) & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]
Z-axis:
\[
\begin{pmatrix}
\cos(\theta) & -\sin(\theta) & 0 & 0 \\
\sin(\theta) & \cos(\theta) & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Translation matrix:
\[
\begin{pmatrix}
1 & 0 & 0 & tx \\
0 & 1 & 0 & ty \\
0 & 0 & 1 & tz \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Scaling matrix:
\[
\begin{pmatrix}
sx & 0 & 0 & 0 \\
0 & sy & 0 & 0 \\
0 & 0 & sz & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Value
The function returns the following values:

\(tx, ty\) transformed coordinates

Note
X, Y, & Z coordinates can be translated or scaled asymmetrically if 3 values are specified for these parameters. If only 1 value is provided, coordinates are translated or scaled symmetrically.

Author(s)
Dave Jones

References

http://www.geocities.com/SiliconValley/2151/win95gpe.html

See Also
f_vecAngle, f_vecMagDir, f_vecTrans, f_vecUV
f_vecUV  

U,V vector components from Magnitude & Direction

Description
This function is used to obtain Cartesian coordinates (the U,V vector components) of a vector given its Polar coordinates (Magnitude & Direction).

Usage

\[ [u,v] = f\_vecUV(mag,dir); \]

Arguments

mag  

column vector specifying magnitude of vectors (in arbitrary units)

dir  

column vector indicating angle of rotation (in degrees from 0–360)

Details
This function is used to obtain Cartesian coordinates (the U,V vector components) of a vector given its Polar coordinates (magnitude & direction). Direction is the counterclockwise angle of rotation.

Value
The function returns the following values:

\[ u,v \]

Cartesian coordinates of heads of vector components

Author(s)

Dave Jones

See Also

f_vecAngle, f_vecMagDir, f_vecTrans
Examples

>> mag = [10 14.142]';
>> dir = [90 225]';
>> [u,v] = f_vecUV(mag,dir)

u =
  6.1232e-016
  -9.9999

v =
  10
  -9.9999
Appendix A

(M)ANOVA Examples

Description

Here are a number of ANOVA Examples using f_npManova. These are provided to help the user become familiar with the syntax used for running the function and for coding of ANOVA factors. Note that f_npManova uses a GLM approach to ANOVA and relies on the unrestricted model for constructing proper F-ratios, so output may differ from that found in textbook examples, which typically use the restricted model with balanced designs. The GLM approach used here should provide output similar to that given by SAS and MINITAB.

Remember that the true power of this function is utilized when the response variable consists of a symmetric distance matrix based one of the metrics commonly used in community ecology (e.g., Bray-Curtis, etc.).

References


Examples

1. For an example of a Two-Way Model I ANOVA with replication, load the file, sr_11p1.mat, in the data folder. This data is from Table 11.1 of Sokal & Rohlf (1995) and has 1 response variable, food, and 2 fixed factors: fat and sex.

```matlab
>> load sr_11p1.mat
>> dis = f_euclid(food');
>> result = f_npManova(dis,[sex fat],0,0,1000,1);
```

References


Please specify the ANOVA model
for 2-way ANOVA factors 1 & 2:

1 & 2 are fixed [21]
1 is fixed or random, 2 is random [22]
1 is fixed or random, 2 is nested [23]

Select model...[O will cancel]
21

Permuting the data 999 times...

Nonparametric (Permutation-based) MANOVA:

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>factor 1</td>
<td>1</td>
<td>3780.8</td>
<td>3780.8</td>
<td>2.5925</td>
<td>0.139</td>
</tr>
<tr>
<td>factor 2</td>
<td>1</td>
<td>61204</td>
<td>61204</td>
<td>41.969</td>
<td>0.001</td>
</tr>
<tr>
<td>factor 1x2</td>
<td>1</td>
<td>918.75</td>
<td>918.75</td>
<td>0.63</td>
<td>0.446</td>
</tr>
<tr>
<td>residual</td>
<td>8</td>
<td>11667</td>
<td>1458.3</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>total</td>
<td>11</td>
<td>77570</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
</tbody>
</table>

# iterations = 1000

(Note: NaNs are placeholders for the ANOVA table)

(Data with replication)

2. For an example of a Two-Way Mixed-Model ANOVA without replication, load the file, sr_11p3.mat, in the data folder. This data is from Box 11.3 of Sokal & Rohlf (1995) and has 1 response variable, temp, and 2 factors: depth (fixed), and day (random).

```matlab
>> load sr_11p3.mat
>> dis = {f_euclid}(temp');
>> result = f_npManova(dis,[depth day],0,0,1000,1);
```

Please specify the ANOVA model
for 2-way ANOVA factors 1 & 2:

1 & 2 are fixed [21]
1 is fixed or random, 2 is random [22]
1 is fixed or random, 2 is nested [23]

Select model...[O will cancel]
22
Permuting the data 999 times...

========================================

Nonparametric (Permutation-based) MANOVA:

<table>
<thead>
<tr>
<th>'Source'</th>
<th>'df'</th>
<th>'SS'</th>
<th>'MS'</th>
<th>'F'</th>
<th>'p'</th>
</tr>
</thead>
<tbody>
<tr>
<td>'factor 1'</td>
<td>[9]</td>
<td>[2119.7]</td>
<td>[235.52]</td>
<td>[2835]</td>
<td>[0.001]</td>
</tr>
<tr>
<td>'factor 2'</td>
<td>[3]</td>
<td>[0.562]</td>
<td>[0.18733]</td>
<td>[2.255]</td>
<td>[0.134]</td>
</tr>
<tr>
<td>'residual'</td>
<td>[27]</td>
<td>[2.243]</td>
<td>[0.083074]</td>
<td>[NaN]</td>
<td>[NaN]</td>
</tr>
<tr>
<td>'total'</td>
<td>[39]</td>
<td>[2122.5]</td>
<td>[NaN]</td>
<td>[NaN]</td>
<td>[NaN]</td>
</tr>
</tbody>
</table>

# iterations = 1000

========================================

(Note: NaNs are placeholders for the ANOVA table)

(Data has NO replication)

3. For an example of a Two-Way Model II Nested ANOVA (balanced), load the file, sr_10p1.mat, in the data folder. This data is from Box 10.1 of Sokal & Rohlf (1995) and has 1 response variable, wing, and 2 factors: cage (random), and female (nested).

```matlab
>> load sr_10p1.mat
>> dis = f_euclid(wing');
>> result = f_npManova(dis,[cage female],0,0,1000,1);
```

Please specify the ANOVA model for 2-way ANOVA factors 1 & 2:

1 & 2 are fixed | [21]
1 is fixed or random, 2 is random | [22]
1 is fixed or random, 2 is nested | [23]

Select model...[0 will cancel]
23

Permuting the data 999 times...

========================================

Nonparametric (Permutation-based) MANOVA:

<table>
<thead>
<tr>
<th>'Source'</th>
<th>'df'</th>
<th>'SS'</th>
<th>'MS'</th>
<th>'F'</th>
<th>'p'</th>
</tr>
</thead>
<tbody>
<tr>
<td>'factor 1'</td>
<td>[2]</td>
<td>[665.68]</td>
<td>[332.84]</td>
<td>[1.7409]</td>
<td>[0.252]</td>
</tr>
<tr>
<td>'factor 2'</td>
<td>[9]</td>
<td>[1720.7]</td>
<td>[191.19]</td>
<td>[146.88]</td>
<td>[0.001]</td>
</tr>
<tr>
<td>'residual'</td>
<td>[12]</td>
<td>[15.62]</td>
<td>[1.3017]</td>
<td>[NaN]</td>
<td>[NaN]</td>
</tr>
<tr>
<td>'total'</td>
<td>[23]</td>
<td>[2402]</td>
<td>[NaN]</td>
<td>[NaN]</td>
<td>[NaN]</td>
</tr>
</tbody>
</table>

77
# iterations = 1000

(Note: NaNs are placeholders for the ANOVA table)

(Data with replication)

4. For an example of a Two-Way Model II Nested ANOVA (unbalanced), load the file, sr_10p6.mat, in the data folder. This data is from Box 10.6 of Sokal & Rohlf (1995) and has 1 response variable, \( ph \), and 2 factors: \( \text{dam} \) (random), and \( \text{sire} \) (nested).

```matlab
>> load sr_10p6.mat
>> dis = f_euclid(ph');
>> result = f_npManova(dis,[\text{dam sire}],0,0,1000,1);
```

Please specify the ANOVA model for 2-way ANOVA factors 1 & 2:

```
1 & 2 are fixed [21]
1 is fixed or random, 2 is random [22]
1 is fixed or random, 2 is nested [23]
```

Select model...[O will cancel]

23

Permuting the data 999 times...

Nonparametric (Permutation-based) MANOVA:

```
'Source' 'df' 'SS' 'MS' 'F' 'p'
'factor 1' [ 14] [1780.2] [127.16] [3.5383] [0.003]
'factor 2' [ 22] [ 790.6] [35.937] [1.4482] [0.107]
'residual' [123] [3052.2] [24.814] [ NaN] [ NaN]
'total' [159] [5622.9] [ NaN] [ NaN] [ NaN]
```

# iterations = 1000

(Note: NaNs are placeholders for the ANOVA table)

(Data with replication)

5. For an example of a Two-Way Model III ANOVA with no replication, load the file, zar_12p4.mat, in the data folder. This data is from Example 12.4 of Zar (1999) and has 1 response variable, \( \text{gain} \), and 2 random factors: \( \text{diet} \), and \( \text{block} \). This is also known as a randomized block design.
>> load zar_12p4.mat
>> dis = f_euclid(gain');
>> result = f_npManova(dis,[diet block],0,0,1000,1);

========================================
Please specify the ANOVA model
for 2-way ANOVA factors 1 & 2:
-----------------------------------------
1 & 2 are fixed [21]
1 is fixed or random, 2 is random [22]
1 is fixed or random, 2 is nested [23]
Select model...[O will cancel]
22
Permuting the data 999 times...

Nonparametric (Permutation-based) MANOVA:

       'Source'   'df'   'SS'   'MS'   'F'   'p'
       'factor 1'  [3]   [27.425]   [9.1418]   [11.825]   [0.004]
       'factor 2'  [4]   [62.647]   [15.662]   [20.259]   [0.001]
       'residual' [12]   [ 9.277]   [0.77308]   [ NaN]   [ NaN]
       'total'    [19]   [ 99.35]   [ NaN]   [ NaN]   [ NaN]

# iterations = 1000

(Data has NO replication)

6. For an example of a Three-Way Model I ANOVA with replication, load the file, zar_14p1.mat, in the data folder. This data is from Example 14.1 of Zar (1999) and has 1 response variable, rate, and 3 fixed factors: species, temp, and sex.

>> load zar_14p1.mat
>> dis = f_euclid(rate');
>> result = f_npManova(dis,[species temp sex],0,0,1000,1);

Please specify the ANOVA model
for 3-way ANOVA factors 1, 2, & 3:
-----------------------------------------
All factors fixed [31]
1 & 2 are fixed, 3 is random [32]
1 is fixed or random, 2 & 3 are random [33]
1 & 2 fixed, 3 nested in 1 (Cross-Nested) [34]
1 &/or 2 random, 3 nested in 1 (Cross-Nested) [35]
3 nested in 2 nested in 1 (Fully Nested) [36]

Select model...[O will cancel]
31

Permuting the data 999 times...

==================================================================
Nonparametric (Permutation-based) MANOVA:
==================================================================

<table>
<thead>
<tr>
<th>'Source'</th>
<th>'df'</th>
<th>'SS'</th>
<th>'MS'</th>
<th>'F'</th>
<th>'p'</th>
</tr>
</thead>
<tbody>
<tr>
<td>factor 1</td>
<td>[ 2]</td>
<td>[ 1.8175]</td>
<td>[ 0.90875]</td>
<td>[24.475]</td>
<td>[0.001]</td>
</tr>
<tr>
<td>factor 2</td>
<td>[ 2]</td>
<td>[ 24.656]</td>
<td>[ 12.328]</td>
<td>[332.02]</td>
<td>[0.001]</td>
</tr>
<tr>
<td>factor 3</td>
<td>[ 1]</td>
<td>[0.0088889]</td>
<td>[0.0088889]</td>
<td>[0.2394]</td>
<td>[0.626]</td>
</tr>
<tr>
<td>factor 1x2</td>
<td>[ 4]</td>
<td>[ 1.1017]</td>
<td>[ 0.27542]</td>
<td>[7.4177]</td>
<td>[0.002]</td>
</tr>
<tr>
<td>factor 1x3</td>
<td>[ 2]</td>
<td>[ 0.37028]</td>
<td>[ 0.18514]</td>
<td>[4.9863]</td>
<td>[0.013]</td>
</tr>
<tr>
<td>factor 2x3</td>
<td>[ 2]</td>
<td>[ 0.17528]</td>
<td>[ 0.087639]</td>
<td>[2.3603]</td>
<td>[0.114]</td>
</tr>
<tr>
<td>factor 1x2x3</td>
<td>[ 4]</td>
<td>[ 0.22056]</td>
<td>[ 0.055139]</td>
<td>[ 1.485]</td>
<td>[0.233]</td>
</tr>
<tr>
<td>residual</td>
<td>[54]</td>
<td>[ 2.005]</td>
<td>[ 0.03713]</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>total</td>
<td>[71]</td>
<td>[ 30.355]</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
</tbody>
</table>

# iterations = 1000

(Note: NaNs are placeholders for the ANOVA table)

(Data with replication)

7. For an example of a Three-Way Model I ANOVA with no replication, load the file, sr_12p1.mat, in the data folder. This data is from Box 12.1 of Zar (1999) and has 1 response variable, time, and 3 fixed factors: temp, cn, and o2.

>> load zar_14p1.mat
>> dis = f_euclid(rate');
>> result = f_npManova(dis,[species temp sex],0,0,1000,1);

==================================================================
Please specify the ANOVA model
for 3-way ANOVA factors 1, 2, & 3:
==================================================================
All factors fixed [31]
1 & 2 are fixed, 3 is random [32]
1 is fixed or random, 2 & 3 are random [33]
1 & 2 fixed, 3 nested in 1 (Cross-Nested) [34]
1 &/or 2 random, 3 nested in 1 (Cross-Nested) [35]
3 nested in 2 nested in 1 (Fully Nested) [36]
Select model...[O will cancel]
31
Permuting the data 999 times...

=================================================================
Nonparametric (Permutation-based) MANOVA:
=================================================================

<table>
<thead>
<tr>
<th>'Source'</th>
<th>'df'</th>
<th>'SS'</th>
<th>'MS'</th>
<th>'F'</th>
<th>'p'</th>
</tr>
</thead>
<tbody>
<tr>
<td>'factor 1'</td>
<td>[2]</td>
<td>[1.8175]</td>
<td>[0.90875]</td>
<td>[24.475]</td>
<td>[0.001]</td>
</tr>
<tr>
<td>'factor 2'</td>
<td>[2]</td>
<td>[24.656]</td>
<td>[12.328]</td>
<td>[332.02]</td>
<td>[0.001]</td>
</tr>
<tr>
<td>'factor 3'</td>
<td>[1]</td>
<td>[0.0088889]</td>
<td>[0.0088889]</td>
<td>[0.2394]</td>
<td>[0.612]</td>
</tr>
<tr>
<td>'factor 1x2'</td>
<td>[4]</td>
<td>[1.1017]</td>
<td>[0.27542]</td>
<td>[7.4177]</td>
<td>[0.001]</td>
</tr>
<tr>
<td>'factor 1x3'</td>
<td>[2]</td>
<td>[0.37028]</td>
<td>[0.18514]</td>
<td>[4.9863]</td>
<td>[0.012]</td>
</tr>
<tr>
<td>'factor 2x3'</td>
<td>[2]</td>
<td>[0.17528]</td>
<td>[0.087639]</td>
<td>[2.3603]</td>
<td>[0.093]</td>
</tr>
<tr>
<td>'factor 1x2x3'</td>
<td>[4]</td>
<td>[0.22056]</td>
<td>[0.055139]</td>
<td>[1.485]</td>
<td>[0.221]</td>
</tr>
<tr>
<td>'residual'</td>
<td>[54]</td>
<td>[2.005]</td>
<td>[0.03713]</td>
<td>[NaN]</td>
<td>[NaN]</td>
</tr>
<tr>
<td>'total'</td>
<td>[71]</td>
<td>[30.355]</td>
<td>[NaN]</td>
<td>[NaN]</td>
<td>[NaN]</td>
</tr>
</tbody>
</table>

# iterations = 1000
=================================================================
(Note: NaNs are placeholders for the ANOVA table)

(Data with replication)

8. For an example of a *Three-Way Model II ANOVA with replication*, load the file, `thick.mat`, in the `data` folder. This data ships with MINITAB and has 1 response variable, `thickness`, 2 fixed factors: `time` and `setting`, and 1 random factor: `operator`.

```matlab
>> load thick.mat
>> dis = f_euclid(thickness');
>> result = f_npManova(dis,[time setting operator],0,0,1000,1);
```

=================================================================
Please specify the ANOVA model
for 3-way ANOVA factors 1, 2, & 3:
=================================================================

All factors fixed [31]
1 & 2 are fixed, 3 is random [32]
1 is fixed or random, 2 & 3 are random [33]
1 & 2 fixed, 3 nested in 1 (Cross-Nested) [34]
1 &/or 2 random, 3 nested in 1 (Cross-Nested) [35]
3 nested in 2 nested in 1 (Fully Nested) [36]

Select model...[O will cancel]
Permuting the data 999 times...

Nonparametric (Permutation-based) MANOVA:

<table>
<thead>
<tr>
<th>'Source'</th>
<th>'df'</th>
<th>'SS'</th>
<th>'MS'</th>
<th>'F'</th>
<th>'p'</th>
</tr>
</thead>
<tbody>
<tr>
<td>'factor 1'</td>
<td>[ 1 ]</td>
<td>[ 9 ]</td>
<td>[ 9 ]</td>
<td>0.29032</td>
<td>0.648</td>
</tr>
<tr>
<td>'factor 2'</td>
<td>[ 2 ]</td>
<td>[ 15676]</td>
<td>[7838.2]</td>
<td>73.178</td>
<td>0.001</td>
</tr>
<tr>
<td>'factor 3'</td>
<td>[ 2 ]</td>
<td>[1120.9]</td>
<td>[560.44]</td>
<td>4.9114</td>
<td>0.052</td>
</tr>
<tr>
<td>'factor 1x2'</td>
<td>[ 2 ]</td>
<td>[ 114.5]</td>
<td>[ 57.25]</td>
<td>2.3854</td>
<td>0.209</td>
</tr>
<tr>
<td>'factor 1x3'</td>
<td>[ 2 ]</td>
<td>[ 62 ]</td>
<td>[ 31 ]</td>
<td>1.2917</td>
<td>0.363</td>
</tr>
<tr>
<td>'factor 2x3'</td>
<td>[ 4 ]</td>
<td>[428.44]</td>
<td>[107.11]</td>
<td>4.463</td>
<td>0.088</td>
</tr>
<tr>
<td>'factor 1x2x3'</td>
<td>[ 4 ]</td>
<td>[ 96 ]</td>
<td>[ 24 ]</td>
<td>7.082</td>
<td>0.004</td>
</tr>
<tr>
<td>'residual'</td>
<td>[18 ]</td>
<td>[ 61 ]</td>
<td>[ 3.889]</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>'total'</td>
<td>[35 ]</td>
<td>[17568]</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
</tbody>
</table>

# iterations = 1000

Please specify the ANOVA model for 3-way ANOVA factors 1, 2, & 3:

- All factors fixed [31]
- 1 & 2 are fixed, 3 is random [32]
- 1 is fixed or random, 2 & 3 are random [33]
- 1 & 2 fixed, 3 nested in 1 (Cross-Nested) [34]
- 1 &/or 2 random, 3 nested in 1 (Cross-Nested) [35]
- 3 nested in 2 nested in 1 (Fully Nested) [36]

Select model...[0 will cancel]
Permuting the data 999 times...

Nonparametric (Permutation-based) MANOVA:

<table>
<thead>
<tr>
<th>'Source'</th>
<th>'df'</th>
<th>'SS'</th>
<th>'MS'</th>
<th>'F'</th>
<th>'p'</th>
</tr>
</thead>
<tbody>
<tr>
<td>'factor 1'</td>
<td>[ 1]</td>
<td>0.34722</td>
<td>0.34722</td>
<td>0.14066</td>
<td>0.736</td>
</tr>
<tr>
<td>'factor 2'</td>
<td>[ 2]</td>
<td>15.361</td>
<td>7.6806</td>
<td>7.6903</td>
<td>0.003</td>
</tr>
<tr>
<td>'factor 3'</td>
<td>[11]</td>
<td>93.486</td>
<td>8.4987</td>
<td>3.6208</td>
<td>0.08</td>
</tr>
<tr>
<td>'factor 1x2'</td>
<td>[ 2]</td>
<td>1.3611</td>
<td>0.68056</td>
<td>0.60767</td>
<td>0.572</td>
</tr>
<tr>
<td>'factor 1x3'</td>
<td>[11]</td>
<td>27.153</td>
<td>2.4684</td>
<td>2.2041</td>
<td>0.048</td>
</tr>
<tr>
<td>'factor 2x3'</td>
<td>[22]</td>
<td>21.972</td>
<td>0.99874</td>
<td>0.89177</td>
<td>0.616</td>
</tr>
<tr>
<td>'residual'</td>
<td>[22]</td>
<td>24.639</td>
<td>1.1199</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>'total'</td>
<td>[71]</td>
<td>184.32</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
</tbody>
</table>

# iterations = 1000

(Note: NaNs are placeholders for the ANOVA table)

(Data has NO replication)

10. For an example of a *Three-Way Model III ANOVA with replication*, load the file, `exercise.mat`, in the `data` folder. This data is from Table 23.4 of Neter et al. (1996) and has 1 response variable, `tol`, and 3 random factors: `gender`, `fat`, and `smoke`.

```matlab
>> load exercise.mat
>> dis = f_euclid(tol');
>> result = f_npManova(dis,[gender fat smoke],0,0,1000,1);
```

Please specify the ANOVA model
for 3-way ANOVA factors 1, 2, & 3:

All factors fixed  [31]
1 & 2 are fixed, 3 is random  [32]
1 is fixed or random, 2 & 3 are random  [33]
1 & 2 fixed, 3 nested in 1 (Cross-Nested)  [34]
1 &/or 2 random, 3 nested in 1 (Cross-Nested)  [35]
3 nested in 2 nested in 1 (Fully Nested)  [36]

Select model...[0 will cancel]
33

Permuting the data 999 times...
Nonparametric (Permutation-based) MANOVA:

<table>
<thead>
<tr>
<th>'Source'</th>
<th>'df'</th>
<th>'SS'</th>
<th>'MS'</th>
<th>'F'</th>
<th>'p'</th>
</tr>
</thead>
<tbody>
<tr>
<td>'factor 1'</td>
<td>[1]</td>
<td>[176.58]</td>
<td>[176.58]</td>
<td>[7.7278]</td>
<td>[0.037]</td>
</tr>
<tr>
<td>'factor 2'</td>
<td>[1]</td>
<td>[242.57]</td>
<td>[242.57]</td>
<td>[2.8797]</td>
<td>[0.084]</td>
</tr>
<tr>
<td>'factor 3'</td>
<td>[1]</td>
<td>[70.384]</td>
<td>[70.384]</td>
<td>[0.86198]</td>
<td>[0.187]</td>
</tr>
<tr>
<td>'factor 1x2'</td>
<td>[1]</td>
<td>[13.65]</td>
<td>[13.65]</td>
<td>[7.2981]</td>
<td>[0.2]</td>
</tr>
<tr>
<td>'factor 1x3'</td>
<td>[1]</td>
<td>[11.07]</td>
<td>[11.07]</td>
<td>[5.9187]</td>
<td>[0.233]</td>
</tr>
<tr>
<td>'factor 2x3'</td>
<td>[1]</td>
<td>[72.454]</td>
<td>[72.454]</td>
<td>[38.737]</td>
<td>[0.089]</td>
</tr>
<tr>
<td>'factor 1x2x3'</td>
<td>[1]</td>
<td>[1.8704]</td>
<td>[1.8704]</td>
<td>[0.20036]</td>
<td>[0.694]</td>
</tr>
<tr>
<td>'residual'</td>
<td>[16]</td>
<td>[149.37]</td>
<td>[9.3354]</td>
<td>[NaN]</td>
<td>[NaN]</td>
</tr>
<tr>
<td>'total'</td>
<td>[23]</td>
<td>[737.95]</td>
<td>[NaN]</td>
<td>[NaN]</td>
<td>[NaN]</td>
</tr>
</tbody>
</table>

# iterations = 1000

(Note: NaNs are placeholders for the ANOVA table)

(Data with replication)

11. For an example of a **Three-Way Model III ANOVA with no replication**, load the file, milk.mat, in the data folder. This is from Brownlee (1960:p.516) and has 1 response variable, counts and 3 random factors: bottle, tube, and sample.

```matlab
>> load milk.mat
>> dis = f_euclid(counts');
>> result = f_npManova(dis,[bottle tube sample],0,0,1000,1);
```

Please specify the ANOVA model
for 3-way ANOVA factors 1, 2, & 3:

All factors fixed [31]
1 & 2 are fixed, 3 is random [32]
1 is fixed or random, 2 & 3 are random [33]
1 & 2 fixed, 3 nested in 1 (Cross-Nested) [34]
1 k/or 2 random, 3 nested in 1 (Cross-Nested) [35]
3 nested in 2 nested in 1 (Fully Nested) [36]

Select model...[O will cancel]
33

Permuting the data 999 times...
For an example of a *Three-Way, Fully Nested, Mixed Model ANOVA (balanced)*, load the file, `sr_10p5.mat`, in the *data* folder. This is from Box 10.5 of Sokal Rohlf (1995) has 1 response variable, *gly*, 1 fixed factor, *rx* and 2 nested factors: *rat* (nested in *rx*) and *prep* (nested in *rat*).

```matlab
>> load sr_10p5.mat
>> dis = f_euclid(gly');
>> result = f_npManova(dis,[rx rat prep],0,0,5000,1);
```

---

Please specify the ANOVA model for 3-way ANOVA factors 1, 2, & 3:

- All factors fixed
- 1 & 2 are fixed, 3 is random
- 1 is fixed or random, 2 & 3 are random
- 1 & 2 fixed, 3 nested in 1 (Cross-Nested)
- 1 &/or 2 random, 3 nested in 1 (Cross-Nested)
- 3 nested in 2 nested in 1 (Fully Nested)

Select model...[0 will cancel]

36

Permuting the data 4999 times...

---

Nonparametric (Permutation-based) MANOVA:

<table>
<thead>
<tr>
<th>'Source'</th>
<th>'df'</th>
<th>'SS'</th>
<th>'MS'</th>
<th>'F'</th>
<th>'p'</th>
</tr>
</thead>
<tbody>
<tr>
<td>'factor 1'</td>
<td>[ 2]</td>
<td>1557.6</td>
<td>778.78</td>
<td>2.929</td>
<td>0.192</td>
</tr>
<tr>
<td>'factor 2'</td>
<td>[ 3]</td>
<td>797.67</td>
<td>265.89</td>
<td>5.3715</td>
<td>0.0148</td>
</tr>
<tr>
<td>'factor 3'</td>
<td>[12]</td>
<td>594</td>
<td>49.5</td>
<td>2.3386</td>
<td>0.0484</td>
</tr>
</tbody>
</table>
13. For an example of a *Three-Way Cross-Nested ANOVA with replication*, load the file, `steel.mat`, in the *data* folder. This is from Brownlee (1960:p.530) and has 1 response variable, `qual`, 2 fixed factors: `ann` and `loc`, and 1 nested factor: `coil` (nested in `ann`). This design is also known as *partially nested* or *partially hierarchical*.

```
>> load steel.mat
>> dis = f_euclid(qual');
>> result = f_npManova(dis,[ann loc coil],0,0,1000,1);
```

Please specify the ANOVA model for 3-way ANOVA factors 1, 2, & 3:

- All factors fixed
- 1 & 2 are fixed, 3 is random
- 1 is fixed or random, 2 & 3 are random
- 1 & 2 fixed, 3 nested in 1 (Cross-Nested)
- 1 &/or 2 random, 3 nested in 1 (Cross-Nested)
- 3 nested in 2 nested in 1 (Fully Nested)

Select model...[O will cancel]

```
34
```

Permuting the data 999 times...

Nonparametric (Permutation-based) MANOVA:

```
'Source'  'df'   'SS'     'MS'    'F'      'p'
'factor 1' [ 1]    [2646] [2646]   [1.091] [0.346]
'factor 2' [ 1]    [1872.7] [1872.7] [35.389] [0.004]
'factor 3' [ 4]    [9701.3] [2425.3] [45.833] [0.001]
'factor 1x2' [ 1]   [16.667] [16.667] [0.31496] [0.718]
'factor 2x3' [ 4]   [211.67] [52.917] [0.50039] [0.718]
'residual' [12]    [1269] [105.75] [ NaN]   [ NaN]
'total'    [23]    [15717] [ NaN]   [ NaN]   [ NaN]
```

(Note: NaNs are placeholders for the ANOVA table)
# iterations = 1000

(Note: NaNs are placeholders for the ANOVA table)

(Data with replication)
Appendix B

F-ratios used by fnpManova

Description

Here are the F-ratios used in the various ANOVA designs supported by fnpManova.

References

This program uses a GLM-approach to (M)ANOVA, so for mixed models (when there are both fixed and random factors) an unrestricted model is used.

**Two-way ANOVA’s:**

A = fixed, B = fixed: \[f_{npManova2}\]

<table>
<thead>
<tr>
<th>Source</th>
<th>F-ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>MS_A/MS_{error}</td>
</tr>
<tr>
<td>B</td>
<td>MS_B/MS_{error}</td>
</tr>
<tr>
<td>AB</td>
<td>MS_{AB}/MS_{error}</td>
</tr>
</tbody>
</table>

A = fixed, B = random: \[f_{npManova2}\]

<table>
<thead>
<tr>
<th>Source</th>
<th>F-ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>MS_A/MS_{AB}</td>
</tr>
<tr>
<td>B</td>
<td>MS_B/MS_{AB}</td>
</tr>
<tr>
<td>AB</td>
<td>MS_{AB}/MS_{error}</td>
</tr>
</tbody>
</table>

A = random, B = random: \[f_{npManova2}\]

<table>
<thead>
<tr>
<th>Source</th>
<th>F-ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>MS_A/MS_B</td>
</tr>
<tr>
<td>B</td>
<td>MS_B/MS_{error}</td>
</tr>
</tbody>
</table>

**Two-way Nested ANOVA’s:**

A = fixed, B = nested in A:

A = random, B = nested in A: \[f_{npManova2n}\]

<table>
<thead>
<tr>
<th>Source</th>
<th>F-ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>MS_A/MS_B</td>
</tr>
<tr>
<td>B</td>
<td>MS_B/MS_{error}</td>
</tr>
</tbody>
</table>

**Three-way ANOVA’s:**

A = fixed, B = fixed, C = fixed: \[f_{npManova3}\]

<table>
<thead>
<tr>
<th>Source</th>
<th>F-ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>MS_A/MS_{error}</td>
</tr>
<tr>
<td>B</td>
<td>MS_B/MS_{error}</td>
</tr>
<tr>
<td>C</td>
<td>MS_C/MS_{error}</td>
</tr>
<tr>
<td>AB</td>
<td>MS_{AB}/MS_{error}</td>
</tr>
<tr>
<td>AC</td>
<td>MS_{AC}/MS_{error}</td>
</tr>
<tr>
<td>BC</td>
<td>MS_{BC}/MS_{error}</td>
</tr>
<tr>
<td>ABC</td>
<td>MS_{ABC}/MS_{error}</td>
</tr>
</tbody>
</table>

A = fixed, B = fixed, C = random: \[f_{npManova3}\]

<table>
<thead>
<tr>
<th>Source</th>
<th>F-ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>MS_A/MS_{AC}</td>
</tr>
<tr>
<td>B</td>
<td>MS_B/MS_{BC}</td>
</tr>
<tr>
<td>C</td>
<td>MS_C/(MS_{AC} + MS_{BC} - MS_{ABC})</td>
</tr>
<tr>
<td>AB</td>
<td>MS_{AB}/MS_{ABC}</td>
</tr>
</tbody>
</table>
A = fixed, B = random, C = random:
A = random, B = random, C = random: \([f_{npManova3}]\)

**Source**  
F-ratio

- A: \(\frac{MS_A}{(MS_{AB} + MS_{AC} - MS_{ABC})}\)
- B: \(\frac{MS_B}{(MS_{AB} + MS_{BC} - MS_{ABC})}\)
- C: \(\frac{MS_C}{(MS_{AC} + MS_{BC} - MS_{ABC})}\)
- AB: \(\frac{MS_{AB}}{MS_{ABC}}\)
- AC: \(\frac{MS_{AC}}{MS_{ABC}}\)
- BC: \(\frac{MS_{BC}}{MS_{ABC}}\)
- ABC: \(\frac{MS_{ABC}}{MS_{error}}\)

**Three-way Cross-Nested ANOVA's:**

**Three-way Fully-Nested ANOVA's:**

A = fixed, B = nested in A, C = nested in B:
A = random, B = nested in A, C = nested in B: \([f_{npManova3Nest2}]\)

**Source**  
F-ratio

- A: \(\frac{MS_A}{MS_{B}}\)
- B: \(\frac{MS_B}{MS_{C}}\)
- C: \(\frac{MS_C}{MS_{error}}\)