# Package ‘SK4FGA’

January 30, 2023

**Type** Package

**Title** Scott-Knott for Forensic Glass Analysis

**Version** 0.1.1

**Maintainer** Toby Hayward <tobyhayward13@gmail.com>

**Description** In forensics, it is common and effective practice to analyse glass fragments from the scene and suspects
to gain evidence of placing a suspect at the crime scene.
This kind of analysis involves comparing the physical and chemical attributes of glass fragments that exist on both the
person and at the crime scene, and assessing the significance in a likeness that they share.
The package implements the Scott-
Knott Modification 2 algorithm (SKM2) (Christopher M. Triggs and
James M. Curran and John S. Buckleton and Kevan A.J. Walsh (1997) <doi:10.1016/S0379-0738(96)02037-3>
``The grouping problem in forensic glass analysis: a divisive approach'', Forensic Science International, 85(1), 1--14)
for small sample glass fragment analysis using the refractive index (ri) of a set of glass samples.
It also includes an experimental multivariate analog to the Scott-
Knott algorithm for similar analysis on glass samples
with multiple chemical concentration variables and multiple samples of the same item; testing against the Hotellings T^2
distribution (J.M. Curran and C.M. Triggs and J.R. Almirall and J.S. Buckleton and K.A.J. Walsh (1997)
<doi:10.1016/S1355-0306(97)72197-X> ``The interpretation of elemental composition measurements from forensic
glass evidence'', Science & Justice, 37(4), 241--244).

**License** GPL (>= 2)

**BugReports** https://github.com/tobyhayward13/SCI118UOA_ForensicGlassAnalysis/issues

**URL** https://github.com/tobyhayward13/SCI118UOA_ForensicGlassAnalysis

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 2.10)

**RoxygenNote** 7.2.3

**LinkingTo** Rcpp

**Imports** Rcpp

**NeedsCompilation** yes

**Author** Toby Hayward [aut, cre] (Main developer and maintainer of the package.),
James Curran [aut, ctb] (Supervised and contributed to the development of the package.),
Lewis Kendall-Jones [ctb] (Wrote and supported the development of the C++ code.)

**Repository** CRAN

**Date/Publication** 2023-01-30 13:00:02 UTC

# R topics documented:

---

`calculate_lambda_threshold`
*Calculate the significance threshold for Lambda.*

---

### Description

For a given significance value, this function uses critical values determined from simulated data formed on 1 million arrays, and returns the quantile estimated at that significance level. For values of k > 20, it assumes a chi squared distribution with k/(pi - 2) degrees of freedom.

### Usage

```
calculate_lambda_threshold(k, alpha)
```

### Arguments

| | |
|---|---|
| `k` | Number of indices. |
| `alpha` | Level of significance. |

### Value

A 100(1-alpha)% quantile estimate from the distribution of Lambda.

## `find_B0` *Calculate B0*

### Description

Calculates the B0 value for a given numeric vector of values; assuming they're appropriate values corresponding to glass fragment refractive indices.

### Usage

```
find_B0(arr)
```

### Arguments

| | |
|---|---|
| `arr` | vector of refractive indices. |

### Value

A numeric corresponding to the maximum between-sum-of-squares estimate from the sample.

## `find_T0` *Calculate T0 for a given list of glass fragments and features.*

### Description

Calculates the "T0" value (the split corresponding to the maximum value of T^2) for a given list of data sets corresponding to glass fragment features assuming they're appropriate values corresponding to glass fragment features.

### Usage

```
find_T0(data, i = 1, j = length(data))
```

### Arguments

| | |
|---|---|
| `data` | list of glass fragment chemical (or otherwise) features. |
| `i` | Starting element (default = 1) |
| `j` | Ending element (default = length(array)) |

### Value

A numeric corresponding to the maximum between-sum-of-squares estimate from the sample.

## `find_T2` *Calculate Hotelling's T^2 Statistic for two independent multivariate samples.*

### Description

Calculate Hotelling's T^2 Statistic for two independent multivariate samples.

### Usage

```
find_T2(d1, d2)
```

### Arguments

`d1` matrix or data.frame type object containing the multivariate data for the first sample.

`d2` matrix or data.frame type object containing the multivariate data for the second sample.

### Value

T^2 value for the two objects.

## `generate_indices` *Generate a test array of Glass-Fragment Refractive Indices.*

### Description

Returns a vector of randomly generated refractive indices from a expected normal distribution of glass fragments.

### Usage

```
generate_indices(n = 10, .sd_multi = 1)
```

### Arguments

`n` Number of refractive indices to generate.

`.sd_multi` Scale factor of the standard deviation. Greater values imply more variance in the random sample.

### Value

A vector of randomly generated RIs.

### Examples

```
test_ris = generate_indices(8)
partition(test_ris)

test_ris_varied = generate_indices(.sd_multi = 5)
partition(test_ris_varied)
```

## glass *glass*

### Description

Glass composition data for seven elements from 200 glass items.

### Usage

```
data(glass)
```

### Format

a ‘data.frame‘ with 2400 rows and 9 columns.

**item** factor200 levels - which item the measurements came from

**fragment** factor4 levels - which of the four fragments from each item the observations were made upon

**logNaO** numericlog of sodium concentration to oxygen concentration

**logMgO** numericlog of magnesium concentration to oxygen concentration

**logAlO** numericlog of aluminium concentration to oxygen concentration

**logSiO** numericlog of silicon concentration to oxygen concentration

**logKO** numericlog of potassium concentration to oxygen concentration

**logCaO** numericlog of calcium concentration to oxygen concentration

**logFeO** numericlog of iron concentration to oxygen concentration

### Details

These data are from Grzegorz (Greg) Zadora at the [Institute of Forensic Research](http://ies.krakow.pl/) in Krakow, Poland. They are the log of the ratios of each element to oxygen, so logNaO is the log(10) of the Sodium to Oxygen ratio, and logAlO is the log of the Aluminium to Oxygen ratio. The instrumental method was SEM-EDX.

The ‘item‘ indicates the object the glass came from. The levels for each item are unique to that item. The ‘fragment‘ can be considered a sub-item. When collecting these observations Greg took a glass object, say a jam jar, he would then break it, and extract four fragments. Each fragment would be measured three times upon different parts of that fragment. The fragment labels are repeated, so, for example, fragment "f1" from item "s2" has nothing whatsoever to do with fragment "f1" from item "s101".

For two level models use ‘item‘ as the lower level - three level models can use the additional information from the individual fragments.

### Source

Grzegorz Zadora [Institute of Forensic Research](http://ies.krakow.pl/), Krakow, Poland.

### References

Aitken, C.G.G. Zadora, G. & Lucy, D. (2007) A Two-Level Model for Evidence Evaluation. *Journal of Forensic Sciences*: **52**(2); 412-419.

---

## `glass2` *glass2*

---

### Description

Glass Fragment Elemental Composition Data on 15 variables.

### Usage

```
data(glass2)
```

### Format

a ‘data.frame‘ with 16 rows and 16 columns.

**item** factor761 levels - which item the measurements came from

**Li7** numericlog of lithium concentration

**Mg25** numericlog of magnesium concentration

**Al27** numericlog of aluminium concentration

**K39** numericlog of potassium concentration

**Ti49** numericlog of titanium concentration

**Mn55** numericlog of manganese concentration

**Fe57** numericlog of iron concentration

**Rb85** numericlog of rubidium concentration

**Sr88** numericlog of strontium concentration

**Zr90** numericlog of zirconium concentration

**Ba137** numericlog of barium concentration

**La139** numericlog of lanthanum concentration

**Ce140** numericlog of cerium concentration

**Nd146** numericlog of neodymium concentration

**Pb208** numericlog of lead concentration

### Details

Log transformed example casework data

### Source

Almirall, Jose; Akmeemana, Anuradha, 2022, "casework.tab", Shiny Glass Application, https://doi.org/10.34703/gzx1-9v95/OB8BS9/CP6WXP, FIU Research Data Portal, V2, UNF:6:jQxEQCGZVvlWtc6owbtp+A== [fileUNF]

### References

Anuradha Akmeemana, R. C., Jose Almirall, The Calculation of Calibrated Likelihood Ratios (LRs) for Glass Using a Multivariate Kernel Density Model: Introducing a User-Friendly Graphical User Interface (GUI). In American Academy of Forensic Science, Anaheim, CA, 2020.

## `has.children` *has.children*

### Description

For internal use only. Determines if a node in the Partition tree has a child.

### Usage

```
has.children(part)
```

### Arguments

| | |
|---|---|
| `part` | Node in partition Tree. |

### Value

Logical determining if the node has any children.

## `order_euclid` *Order a list of data frames containing numerical columns by their euclidean distance to the mean.*

### Description

Meant for internal use only.

### Usage

```
order_euclid(alist)
```

### Arguments

| | |
|---|---|
| `alist` | A list of data frames. |

### Value

A list of data frames.

## `partition` *Create Partitions of an RI array.*

### Description

Partitions the array of assumed glass fragment refractive indices into statistically significant groups.

### Usage

```
partition(array, alpha = 0.05, .debug = FALSE)
```

### Arguments

| | |
|---|---|
| `array` | Vector of refractive indices. |
| `alpha` | Significance parameter "[0,1]". Higher values are more likely to partition the array further. |
| `.debug` | Runs debugging. |

### Value

sk_partition_tree

## Examples

```
set.seed(123)
ris = generate_indices(8, 4)
part = partition(ris)
plot(part)
part$groups
```

---

## partition.multi *Create Partitions of a multivariate array of objects.*

---

### Description

Partitions the array of assumed glass fragment chemical compositions and features into statistically significant groups.

### Usage

```
partition.multi(data, alpha = 0.05, .debug = FALSE)
```

### Arguments

| | |
|---|---|
| data | A list of data.frames or matrices corresponding to individual observations of glass fragment features. |
| alpha | Significance parameter "[0,1]". Higher values are more likely to partition the array further. |
| .debug | Runs debugging. |

### Value

A list of groupings and the tree formed.

### Examples

```
test.data = prepare_data(glass, 1)[1:3]
part = partition.multi(test.data)
plot(part)

set.seed(123)
test.data.random = prepare_data(glass, 1)
test.data.random = test.data.random[sample(1:length(test.data.random), 5)]
part = partition.multi(test.data.random)
part$groups
```

## plot.sk_partition_tree

*Plot S3method for objects of type "sk_partition_tree".*

### Description

S3method for plotting the resulting tree formed by the partitioning algorithms in the SK4FGA package.

### Usage

```
## S3 method for class 'sk_partition_tree'
plot(x, ...)
```

### Arguments

| | |
|---|---|
| `x` | Output from the function "partition()" |
| `...` | Extra details for the plot. Unused. |

### Value

Plot of the decision tree that is formed by the sk_partition_tree object returned by partition and partition.multi.

### Examples

```
data = generate_indices()
part = partition(data)
plot.sk_partition_tree(part)

data(glass)
data.multi = prepare_data(glass, 1)[1:3]
part = partition.multi(data.multi)
plot(part)
```

## prepare_data

*prepare_data*

### Description

Prepare a data file that is in standard form for partition.multi.

### Usage

```
prepare_data(data, label = NA)
```

### Arguments

| | |
|---|---|
| `data` | Inputted data.frame. |
| `label` | Column corresponding to the label to be grouped by. |

### Value

A list of split data.

## `ptsquared` *Calculate the Probability for a given T^2 statistic.*

### Description

Calculate the Probability for a given T^2 statistic.

### Usage

```
ptsquared(t, n1, n2, p)
```

### Arguments

| | |
|---|---|
| `t` | T^2 statistic. |
| `n1` | Number of observations in first sample. |
| `n2` | Number of observations in second sample. |
| `p` | Number of parameters. |

### Value

A probability corresponding to a given T^2 statistic and for given arguments.

## `ungroup.partition` *ungroup.partition*

### Description

Ungroups the tree object in the output from partition()

### Usage

```
ungroup.partition(tree)
```

### Arguments

| | |
|---|---|
| `tree` | tree object returned from partition() |

### Value

A list object containing the indices of the `vehicle.glass` *vehicle.glass*

## Description

FIU Vehicle Glass Database V2.0

## Usage

```
data(vehicle.glass)
```

## Format

a ‘data.frame‘ with 6858 rows and 16 columns.

**item** factor761 levels - which item the measurements came from

**Li7** numericlog of lithium concentration

**Mg25** numericlog of magnesium concentration

**Al27** numericlog of aluminium concentration

**K39** numericlog of potassium concentration

**Ti49** numericlog of titanium concentration

**Mn55** numericlog of manganese concentration

**Fe57** numericlog of iron concentration

**Rb85** numericlog of rubidium concentration

**Sr88** numericlog of strontium concentration

**Zr90** numericlog of zirconium concentration

**Ba137** numericlog of barium concentration

**La139** numericlog of lanthanum concentration

**Ce140** numericlog of cerium concentration

**Nd146** numericlog of neodymium concentration

**Pb208** numericlog of lead concentration

## Details

This freely available research-based database consists of 762 samples of various vehicle glass (windshield, passenger side, driver side, etc.). The samples span various makes and models, and range in year from 2004-2019. All samples were collected from the M&M salvage yard in Ruckersville, VA.

## Source

Almirall, Jose; Akmeemana, Anuradha, 2022, "FIU Vehicle Glass Database V2.0.tab", Shiny Glass Application, https://doi.org/10.34703/gzx1-9v95/OB8BS9/XGH0IO, FIU Research Data Portal, V2, UNF:6:YDbwWISU04S+UCtb7aRoBQ== [fileUNF]

## References

Anuradha Akmeemana, R. C., Jose Almirall, The Calculation of Calibrated Likelihood Ratios (LRs) for Glass Using a Multivariate Kernel Density Model: Introducing a User-Friendly Graphical User Interface (GUI). In American Academy of Forensic Science, Anaheim, CA, 2020.

# Index