

# Package: opa (via r-universe)

September 5, 2024

**Type** Package

**Title** An Implementation of Ordinal Pattern Analysis

**Version** 0.8.2.033

**Description** Quantifies hypothesis to data fit for repeated measures and longitudinal data, as described by Thorngate (1987) <[doi:10.1016/S0166-4115\(08\)60083-7](https://doi.org/10.1016/S0166-4115(08)60083-7)> and Grice et al., (2015) <[doi:10.1177/2158244015604192](https://doi.org/10.1177/2158244015604192)>. Hypothesis and data are encoded as pairwise relative orderings which are then compared to determine the percentage of orderings in the data that are matched by the hypothesis.

**License** GPL (>= 3)

**URL** <https://timbeechey.github.io/opa/>

**BugReports** <https://github.com/timbeechey/opa/issues>

**Encoding** UTF-8

**LazyData** true

**LinkingTo** Rcpp, RcppArmadillo

**Imports** Rcpp, stats, lattice

**Suggests** knitr, rmarkdown, tinytest

**SystemRequirements** C++17

**Depends** R (>= 2.10)

**RoxygenNote** 7.3.1

**VignetteBuilder** knitr

**Repository** <https://timbeechey.r-universe.dev>

**RemoteUrl** <https://github.com/timbeechey/opa>

**RemoteRef** HEAD

**RemoteSha** ac39285a703e9403a63db0ede33d3a422eae97c4

## Contents

bees	2
compare_conditions	3
compare_groups	4
compare_hypotheses	4
correct_pairs	5
cval_plot	6
group_cvals	6
group_pccs	7
group_results	7
hypothesis	8
incorrect_pairs	9
individual_cvals	9
individual_pccs	10
individual_results	10
opa	11
pcc_plot	13
pituitary	14
plot.opafit	14
plot.opaGroupComparison	15
plot.opahypothesis	16
plot.opaHypothesisComparison	17
plot.oparandpccs	18
print.opafit	18
print.opaGroupComparison	19
print.opahypothesis	20
print.opaHypothesisComparison	20
print.pairwiseopafit	21
random_pccs	22
summary.opafit	23
summary.opaGroupComparison	23
summary.opaHypothesisComparison	24
<b>Index</b>	<b>26</b>

---

bees

*Bee data*

---

### Description

A data frame with 20 rows and 14 columns containing times between visits to a mechanical flower by bees in two experimental conditions.

### Usage

bees

**Format**

**bee** Unique identifier for each individual bee.

**condition** Factor identifying the two experimental conditions. In the frustrated condition bees were temporarily restricted from returning to the hive after collecting nectar, in the free condition bees were able to return to the hive without delay.

**t1-t12** Time between visits to the mechanical flower (in seconds) in each of 12 consecutive trials.

**Source**

Grice, J. W., Craig, D. P. A., & Abramson, C. I. (2015). A Simple and Transparent Alternative to Repeated Measures ANOVA. *SAGE Open*, 5(3), 215824401560419. <https://doi.org/10.1177/2158244015604192>

---

compare_conditions	<i>Calculates PCCs and c-values based on pairwise comparison of conditions.</i>
--------------------	---

---

**Description**

Calculates PCCs and c-values based on pairwise comparison of conditions.

**Usage**

```
compare_conditions(result, nreps = 1000L)
```

**Arguments**

result	an object of class "opafit" produced by a call to opa().
nreps	an integer

**Value**

compare\_conditions returns a list with the following elements

**pcc\_mat** A lower triangle matrix containing PCCs calculated from each pairing of data columns.

**cval\_mat** A lower triangle matrix containing c-values calculated from each pairing of data columns.

**pccs** A vector containing PCCs calculated from each pairing of data.

**cvals** A vector containing c-values calculated from each pairing of data.

**nreps** The number of permutations used to calculate the c-values.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11),
                 t4 = c(10, 5, 11, 12))
h <- hypothesis(1:4)
opamod <- opa(dat, h)
compare_conditions(opamod)
```

---

compare_groups	<i>Calculate the c-value of the difference in PCCs produced by two groups</i>
----------------	---

---

**Description**

Calculate the c-value of the difference in PCCs produced by two groups

**Usage**

```
compare_groups(m, group1, group2, two_tailed)
```

**Arguments**

m	an object of class "opafit" produced by a call to opa().
group1	a character string which matches a group level passed to opa().
group2	a character string which matches a group level passed to opa().
two_tailed	a boolean indicating whether the comparison is two-tailed.

**Value**

an object of class "opaGroupComparison".

**Examples**

```
dat <- data.frame(group = c("a", "b", "a", "b"),
                  t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11))
dat$group <- factor(dat$group, levels = c("a", "b"))
h <- hypothesis(1:3)
opamod <- opa(dat[,2:4], h, group = dat$group)
compare_groups(opamod, "a", "b")
```

---

compare_hypotheses	<i>Calculate the c-value of the difference in PCCs produced by two hypotheses</i>
--------------------	---

---

**Description**

Calculate the c-value of the difference in PCCs produced by two hypotheses

**Usage**

```
compare_hypotheses(m1, m2, two_tailed)
```

**Arguments**

m1                    an object of class "opafit" produced by a call to opa().  
 m2                    an object of class "opafit" produced by a call to opa().  
 two\_tailed          a boolean indicating whether the comparison is two-tailed.

**Value**

an object of class "opaHypothesisComparison".

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11),
                  t4 = c(10, 5, 11, 12))
h1 <- hypothesis(c(1, 2, 3, 4))
h2 <- hypothesis(c(1, 4, 2, 3))
opamod1 <- opa(dat, h1)
opamod2 <- opa(dat, h2)
compare_hypotheses(opamod1, opamod2, two_tailed = TRUE)
```

---

<code>correct_pairs</code>	<i>Return the number of pairs of observations matched by the hypothesis</i>
----------------------------	---

---

**Description**

Return the number of pairs of observations matched by the hypothesis

**Usage**

```
correct_pairs(m)
```

**Arguments**

m                    an object of class "opafit"

**Value**

a non-negative integer

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
correct_pairs(opamod)
```

cval\_plot

*Plot individual chance values*

---

**Description**

Plot individual chance values

**Usage**

```
cval_plot(m)
```

**Arguments**

m                    an object of class "opafit"

**Value**

No return value, called for side effects.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
cval_plot(opamod)
```

---

group\_cvals*Return the group chance values of the specified model*

---

**Description**

Return the group chance values of the specified model

**Usage**

```
group_cvals(m)
```

**Arguments**

m                    an object of class "opafit"

**Value**

a numeric vector

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
group_cvals(opamod)
```

---

**group\_pccs***Return the group PCCs of the specified model*

---

**Description**

Return the group PCCs of the specified model

**Usage**

```
group_pccs(m)
```

**Arguments**

**m** an object of class "opafit"

**Value**

a numeric vector

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
group_pccs(opamod)
```

---

**group\_results***Group-level PCC and chance values.*

---

**Description**

Group-level PCC and chance values.

**Usage**

```
group_results(m, digits)
```

**Arguments**

`m` an object of class "opafit" produced by `opa()`.  
`digits` a positive integer.

**Details**

If the model was fitted with no grouping variable, a single PCC and c-value are returned. If a grouping variable was specified in the call to `opa` then PCCs and c-values are returned for each factor level of the grouping variable.

**Value**

a matrix with 1 row per group.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
group_results(opamod)
```

---

hypothesis

*Create a hypothesis object*

---

**Description**

Create a hypothesis object

**Usage**

```
hypothesis(xs, type = "pairwise")
```

**Arguments**

`xs` a numeric vector  
`type` a string

**Value**

a list containing the following elements

**Examples**

```
h1 <- hypothesis(c(2, 1, 3, 4), type = "pairwise")
h2 <- hypothesis(c(2, 1, 3, 4), type = "adjacent")
```



---

incorrect_pairs	<i>Return the number of pairs of observations not matched by the hypothesis</i>
-----------------	---

---

**Description**

Return the number of pairs of observations not matched by the hypothesis

**Usage**

```
incorrect_pairs(m)
```

**Arguments**

m                    an object of class "opafit"

**Value**

a non-negative integer

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
incorrect_pairs(opamod)
```

---

individual_cvals	<i>Return the individual chance values of the specified model</i>
------------------	---

---

**Description**

Return the individual chance values of the specified model

**Usage**

```
individual_cvals(m)
```

**Arguments**

m                    an object of class "opafit"

**Value**

a numeric vector

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
individual_cvals(opamod)
```

---

individual\_pccs      *Return the individual PCCs of the specified model*

---

**Description**

Return the individual PCCs of the specified model

**Usage**

```
individual_pccs(m)
```

**Arguments**

m                    an object of class "opafit"

**Value**

a numeric vector

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
individual_pccs(opamod)
```

---

individual\_results      *Individual-level PCC and chance values.*

---

**Description**

Individual-level PCC and chance values.

**Usage**

```
individual_results(m, digits)
```

**Arguments**

`m` an object of class "opafit" produced by `opa()`  
`digits` an integer

**Details**

If the opa model was fitted with no grouping variable, a matrix of PCCs and c-values are returned corresponding to the order of rows in the data. If the opa model was fitted with a grouping variable specified, a table of PCCs and c-values is returned ordered by factor level of the grouping variable.

**Value**

a matrix containing a column of PCC values and a column of c-values with 1 row per row of data.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
individual_results(opamod)
```

---

 opa

*Fit an ordinal pattern analysis model*


---

**Description**

opa is used to fit ordinal pattern analysis models by computing the percentage of pair orderings in each row of data which are matched by corresponding pair orderings in an hypothesis, in addition the chance of a permutation of the data producing a percentage match as great.

**Usage**

```
opa(
  dat,
  hypothesis,
  group = NULL,
  pairing_type = "pairwise",
  diff_threshold = 0,
  nreps = 1000L,
  shuffle_across_individuals = FALSE
)
```

## Arguments

<code>dat</code>	a data frame
<code>hypothesis</code>	a numeric vector
<code>group</code>	an optional factor vector
<code>pairing_type</code>	a string
<code>diff_threshold</code>	a positive integer or floating point number
<code>nreps</code>	an integer, ignored if <code>cval_method = "exact"</code>
<code>shuffle_across_individuals</code>	a boolean indicating whether to randomize data across individuals in c-value computation.

## Details

Data is expected in **wide** format with 1 row per individual and 1 column per measurement condition. Data must contain only columns consisting of numerical values of the *dependent* variable.

The length of the `hypothesis` must be equal to the number of columns in the dependent variable data.frame `dat`.

Any *independent* variable must be passed separately as a vector with the `group` keyword. The grouping vector must be a *factor*.

`pairing_type` must be either "pairwise" or "adjacent". The "pairwise" option considered the relative ordering of every pair of observations in the data and every pair of elements of the hypothesis. The "adjacent" option considers the ordering of adjacent pairs only. If unspecified, the default is "pairwise".

`diff_threshold` may be a positive integer or double. If unspecified a default zero threshold is used. The `diff_threshold` is never applied to the hypothesis.

`nreps` specifies the number of random reorderings to use in the calculation of chance-values.

## Value

`opa` returns an object of class "opafit".

An object of class "opafit" is a list containing the following components:

**group\_pcc** the percentage of pairwise orderings from all pooled data rows which were correctly classified by the hypothesis.

**individual\_pccs** a vector containing the percentage of pairwise orderings that were correctly classified by the hypothesis for each data row.

**correct\_pairs** an integer representing the number of pairwise orderings pooled across all data rows that were correctly classified by the hypothesis.

**total\_pairs** an integer, the number of pair orderings contained in the data.

**group\_cval** the group-level chance value.

**individual\_cvals** a vector containing chance values for each data row

**rand\_pccs** A vector of PCCS calculated from each random ordering with length equal to `nreps`, a list of vectors if a group vector was passed to `opa()`.

**call** The matched call

**hypothesis** The hypothesis vector passed to opa()

**pairing\_type** A string indicating the method of pairing passed to opa().

**diff\_threshold** The numeric difference threshold used to calculate PCCs. If no value was passed in the `diff_threshold`, the default of 0 is used.

**data** The `data.frame` passed to opa().

**groups** The vector of groups passed to opa. If no group vector was passed to opa() the default of NULL is used.

**nreps** an integer, the number of random re-orderings of the data used to compute chance values.

## References

Grice, J. W., Craig, D. P. A., & Abramson, C. I. (2015). A Simple and Transparent Alternative to Repeated Measures ANOVA. *SAGE Open*, 5(3), 215824401560419. <<https://doi.org/10.1177/2158244015604192>>

Thorngate, W. (1987). Ordinal Pattern Analysis: A Method for Assessing Theory-Data Fit. *Advances in Psychology*, 40, 345–364. <[https://doi.org/10.1016/S0166-4115\(08\)60083-7](https://doi.org/10.1016/S0166-4115(08)60083-7)>

## Examples

```
dat <- data.frame(group = c("a", "b", "a", "b"),
                 t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
dat$group <- factor(dat$group, levels = c("a", "b"))
h <- hypothesis(1:3)
opamod <- opa(dat[,2:4], h, group = dat$group)
```

---

pcc\_plot

*Plot individual PCCs.*

---

## Description

Plot individual PCCs.

## Usage

```
pcc_plot(m)
```

## Arguments

`m` an object of class "opafit"

## Value

No return value, called for side effects.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
pcc_plot(opamod)
```

---

pituitary

*Childhood growth data*

---

**Description**

A data frame with 108 rows and 4 columns containing data on the distance from the the pituitary to the pteryo-maxillary fissure in children.

**Usage**

```
pituitary
```

**Format**

**distance** Distance in mm from the pituitary to the pteryo-maxillary fissure.

**age** Age in years.

**individual** Unique identifier for each individual.

**sex** Biological sex of each individual.

**Source**

Potthoff, R. F., & Roy, S. N. (1964). A Generalized Multivariate Analysis of Variance Model Useful Especially for Growth Curve Problems. *Biometrika*, 51(3/4), 313–326. <https://doi.org/10.2307/2334137>

---

plot.opafit

*Plots individual-level PCCs and chance-values.*

---

**Description**

Plots individual-level PCCs and chance-values.

**Usage**

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

**Arguments**

x                    an object of class "opafit" produced by opa()  
...                    ignored

**Value**

No return value, called for side effects.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
plot(opamod)
```

---

plot.opaGroupComparison

*Plot group comparison PCC replicates.*

---

**Description**

Plot group comparison PCC replicates.

**Usage**

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

**Arguments**

x                    an object of class "oparandpccs" produced by random\_pccs()  
...                    ignored

**Details**

Plot a histogram of PCCs computed from randomly reordered data used to calculate the chance-value for a group comparison.

**Value**

no return value, called for side effects only.

**Examples**

```

dat <- data.frame(group = c("a", "b", "a", "b"),
                  t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11))
dat$group <- factor(dat$group, levels = c("a", "b"))
h <- hypothesis(1:3)
opamod <- opa(dat[,2:4], h, group = dat$group)
z <- compare_groups(opamod, "a", "b")
plot(z)

```

---

plot.opahypothesis      *Plot a hypothesis.*

---

**Description**

Plot a hypothesis.

**Usage**

```

## S3 method for class 'opahypothesis'
plot(x, title = TRUE, ...)

```

**Arguments**

x	an object of class "opaHypothesis"
title	a boolean indicating whether to include a plot title
...	ignored

**Value**

No return value, called for side effects.

**Examples**

```

h <- hypothesis(c(1,2,3,3,3))
plot(h)

```



---

`plot.opaHypothesisComparison`*Plot hypothesis comparison PCC replicates.*

---

### Description

Plot hypothesis comparison PCC replicates.

### Usage

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

### Arguments

<code>x</code>	an object of class "oparandpccs" produced by <code>random_pccs()</code>
<code>...</code>	ignored

### Details

Plot a histogram of PCCs computed from randomly reordered data used to calculate the chance-value for a hypothesis comparison.

### Value

no return value, called for side effects only.

### Examples

```
dat <- data.frame(t1 = c(9, 4, 8, 10),  
                 t2 = c(8, 8, 12, 10),  
                 t3 = c(8, 5, 10, 11),  
                 t4 = c(10, 5, 11, 12))  
h1 <- hypothesis(c(1, 2, 3, 4))  
h2 <- hypothesis(c(1, 4, 2, 3))  
opamod1 <- opa(dat, h1)  
opamod2 <- opa(dat, h2)  
z <- compare_hypotheses(opamod1, opamod2)  
plot(z)
```

---

plot.oparandpccs      *Plot PCC replicates.*

---

### Description

Plot PCC replicates.

### Usage

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

### Arguments

x                    an object of class "oparandpccs" produced by random\_pccs()  
 ...                 ignored

### Details

Plot a histogram of PCCs computed from randomly reordered data used to calculate the chance-value.

### Value

no return value, called for side effects only.

### Examples

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
plot(random_pccs(opamod))
```

---

print.opafit              *Displays the call used to fit an ordinal pattern analysis model.*

---

### Description

Displays the call used to fit an ordinal pattern analysis model.

### Usage

```
## S3 method for class 'opafit'
print(x, ...)
```

**Arguments**

x                    an object of class "opafit".  
 ...                  ignored

**Value**

No return value, called for side effects.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
print(opamod)
```

---

```
print.opaGroupComparison
```

*Prints a summary of results from hypothesis comparison.*

---

**Description**

Prints a summary of results from hypothesis comparison.

**Usage**

```
## S3 method for class 'opaGroupComparison'
print(x, ...)
```

**Arguments**

x                    an object of class "opaHypothesisComparison".  
 ...                  ignored

**Value**

No return value, called for side effects.

**Examples**

```
dat <- data.frame(group = c("a", "b", "a", "b"),
                  t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11))
dat$group <- factor(dat$group, levels = c("a", "b"))
h <- hypothesis(1:3)
opamod <- opa(dat[,2:4], h, group = dat$group)
```

```
z <- compare_groups(opamod, "a", "b")
print(z)
```

---

```
print.opahypothesis Print details of a hypothesis
```

---

**Description**

Print details of a hypothesis

**Usage**

```
## S3 method for class 'opahypothesis'
print(x, ...)
```

**Arguments**

x	an object of type "opaHypothesis"
...	ignored

**Value**

No return value, called for side-effects.

**Examples**

```
h1 <- hypothesis(c(2, 1, 3, 4), type = "pairwise")
print(h1)
h2 <- hypothesis(c(2, 1, 3, 4), type = "adjacent")
print(h2)
```

---

```
print.opaHypothesisComparison
Prints a summary of results from hypothesis comparison.
```

---

**Description**

Prints a summary of results from hypothesis comparison.

**Usage**

```
## S3 method for class 'opaHypothesisComparison'
print(x, ...)
```

**Arguments**

x                    an object of class "opaHypothesisComparison".  
...                    ignored

**Value**

No return value, called for side effects.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11),
                 t4 = c(10, 5, 11, 12))
h1 <- hypothesis(c(1, 2, 3, 4))
h2 <- hypothesis(c(1, 4, 2, 3))
opamod1 <- opa(dat, h1)
opamod2 <- opa(dat, h2)
z <- compare_hypotheses(opamod1, opamod2)
print(z)
```

---

`print.pairwiseopafit`    *Displays the results of a pairwise ordinal pattern analysis.*

---

**Description**

Displays the results of a pairwise ordinal pattern analysis.

**Usage**

```
## S3 method for class 'pairwiseopafit'
print(x, ...)
```

**Arguments**

x                    an object of class "pairwiseopafit".  
...                    ignored

**Value**

No return value, called for side effects.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
pw <- compare_conditions(opamod)
print(pw, digits = 2)
```

---

random_pccs	<i>Return the random order generated PCCs used to calculate the group chance value</i>
-------------	--

---

**Description**

Return the random order generated PCCs used to calculate the group chance value

**Usage**

```
random_pccs(m)
```

**Arguments**

m                    an object of class "opafit"

**Value**

a numeric vector

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
h <- hypothesis(1:3)
opamod <- opa(dat, h)
random_pccs(opamod)
```

---

summary.opafit	<i>Prints a summary of results from a fitted ordinal pattern analysis model.</i>
----------------	--

---

**Description**

Prints a summary of results from a fitted ordinal pattern analysis model.

**Usage**

```
## S3 method for class 'opafit'  
summary(object, ..., digits = 2L)
```

**Arguments**

object	an object of class "opafit".
...	ignored
digits	an integer used for rounding values in the output.

**Value**

No return value, called for side effects.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),  
                 t2 = c(8, 8, 12, 10),  
                 t3 = c(8, 5, 10, 11))  
h <- hypothesis(1:3)  
opamod <- opa(dat, h)  
summary(opamod)  
summary(opamod, digits = 3)
```

---

summary.opaGroupComparison	<i>Prints a summary of results from hypothesis comparison.</i>
----------------------------	--

---

**Description**

Prints a summary of results from hypothesis comparison.

**Usage**

```
## S3 method for class 'opaGroupComparison'  
summary(object, ...)
```

**Arguments**

object            an object of class "opaHypothesisComparison".  
 ...                ignored

**Value**

No return value, called for side effects.

**Examples**

```
dat <- data.frame(group = c("a", "b", "a", "b"),
                  t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11))
dat$group <- factor(dat$group, levels = c("a", "b"))
h <- hypothesis(1:3)
opamod <- opa(dat[,2:4], h, group = dat$group)
z <- compare_groups(opamod, "a", "b")
summary(z)
```

---

```
summary.opaHypothesisComparison
```

*Prints a summary of results from hypothesis comparison.*

---

**Description**

Prints a summary of results from hypothesis comparison.

**Usage**

```
## S3 method for class 'opaHypothesisComparison'
summary(object, ...)
```

**Arguments**

object            an object of class "opaHypothesisComparison".  
 ...                ignored

**Value**

No return value, called for side effects.



**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11),
                 t4 = c(10, 5, 11, 12))
h1 <- hypothesis(c(1, 2, 3, 4))
h2 <- hypothesis(c(1, 4, 2, 3))
opamod1 <- opa(dat, h1)
opamod2 <- opa(dat, h2)
z <- compare_hypotheses(opamod1, opamod2)
summary(z)
```

# Index

## \* datasets

- bees, [2](#)
- pituitary, [14](#)

bees, [2](#)

compare\_conditions, [3](#)  
compare\_groups, [4](#)  
compare\_hypotheses, [4](#)  
correct\_pairs, [5](#)  
cval\_plot, [6](#)

group\_cvals, [6](#)  
group\_pccs, [7](#)  
group\_results, [7](#)

hypothesis, [8](#)

incorrect\_pairs, [9](#)  
individual\_cvals, [9](#)  
individual\_pccs, [10](#)  
individual\_results, [10](#)

opa, [11](#)

pcc\_plot, [13](#)  
pituitary, [14](#)  
plot.opafit, [14](#)  
plot.opaGroupComparison, [15](#)  
plot.opahypothesis, [16](#)  
plot.opaHypothesisComparison, [17](#)  
plot.oparandpccs, [18](#)  
print.opafit, [18](#)  
print.opaGroupComparison, [19](#)  
print.opahypothesis, [20](#)  
print.opaHypothesisComparison, [20](#)  
print.pairwiseopafit, [21](#)

random\_pccs, [22](#)

summary.opafit, [23](#)  
summary.opaGroupComparison, [23](#)  
summary.opaHypothesisComparison, [24](#)