

# Package ‘ODA’

December 6, 2020

**Maintainer** Nathaniel Rhodes <nrhode@midwestern.edu>

**License** GPL-3

**Title** a package and interface for the MegaODA software suite

**Version** 1.1.2

**Authors** Nathaniel Rhodes [aut, cre] and Paul Yarnold [ctb, cph]

**Description** This package contains the functions needed to run and evaluate the output from MegaODA software. For any statistical hypothesis this non-parametric statistically-motivated machine-learning algorithm explicitly obtains the model which maximizes the (weighted) predictive accuracy for the sample. Many model validation methods are available. Users are encouraged to read about the ODA paradigm in Maximizing Predictive Accuracy (Paul Yarnold and Robert Soltysik, 2016), or on the ODA website. This package was developed by Nathaniel J. Rhodes to interface with ODA to assist the user in developing, evaluating, and validating maximum-accuracy ODA models.

**URL** <https://odajournal.com/>

**Date** 2020-12-06

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Depends** R (>= 3.5.0), utils, stats

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Imports** epitools,  
filesstrings,  
statmod,  
stringr,  
benchmarkme

## R topics documented:

.onAttach . . . . .	2
.onLoad . . . . .	2
NOVOboot . . . . .	3
ODAclean . . . . .	4
ODAlload . . . . .	6
ODAmannual . . . . .	7

ODAParse . . . . .	7
ODAPower . . . . .	9
ODARun . . . . .	10
ODATree . . . . .	13

<b>Index</b>	<b>15</b>
--------------	-----------

---

.onAttach	<i>.onAttach start message</i>
-----------	--------------------------------

---

**Description**

.onAttach start message

**Usage**

.onAttach(libname, pkgname)

**Arguments**

libname	defunct
pkgname	defunct

**Value**

invisible()

---

.onLoad	<i>.onLoad getOption package settings</i>
---------	---

---

**Description**

.onLoad getOption package settings

**Usage**

.onLoad(libname, pkgname)

**Arguments**

libname	defunct
pkgname	defunct

**Value**

invisible()

NOVOboot

*Perform novometric bootstrap analysis using a classification model***Description**

Perform novometric bootstrap analysis using a classification model

**Usage**

```
NOVOboot(
  data = "",
  run = "",
  predictor = "",
  outcome = "",
  nboot = "",
  seed = ""
)
```

**Arguments**

data	The name of a valid object created by <a href="#">ODAParse</a> that begins with <code>oda.list.x</code> where x is the run.
run	The run number of the ODA model created using <a href="#">ODArun</a> and parsed using <a href="#">ODAParse</a> .
predictor	The model number within the <a href="#">ODArun</a> which is read using <a href="#">ODAParse</a> . This is the model that is compared to chance.
outcome	The outcome number, when more than one outcome was evaluated. For example, if three outcomes were evaluated simultaneously and the impact of outcome one was of interest, then enter 1.
nboot	The number of bootstrap replicates. Both model and chance will be evaluated using this number of replicates using a 50% resampling with replacement. The default value is 25,000 replicates.
seed	The seed number passed to <a href="#">set.seed</a> that serves as the origin of the pseudorandom numbers generated for the bootstrap resampling. The default seed number is the current system time.

**Details**

The first axiom of novometric theory states that, for a random statistical sample "S" consisting of a class variable, one or more attributes, and a weight, the corresponding exact discrete confidence intervals, CIs, for model and for chance do NOT overlap. That is, a significant model exists. If the CIs overlap, the effect is not statistically significant. However, if the CIs do not overlap, then the effect strength of the model is statistically significant at the confidence level selected by the user.

NOVOboot reports the distribution of ESS in quantiles from 0% to 100%. This provides a mechanism to evaluate the validity of an ODA model through bootstrap resampling with replacement. The same methodology can be applied to any classification model wherein a two class and two attribute confusion matrix can be formulated. It is axiomatic that the exact discrete confidence intervals for model and chance define the boundry between "signal" and "noise". A distribution of p values from Fisher's Exact tests conducted for each bootstrap replicate are also supplied and can be graphically evaluated, as in the example below.

## Value

An array of percentiles from 0% to 100% capturing the resampled model and chance performance metrics. The simulated bootstraps are stored within the object `novo.boot.x` which is appended with the current run number `x`.

## References

Yarnold, P.R. (2020). Reformulating the First Axiom of Novometric Theory: Assessing Minimum Sample Size in Experimental Design *Optimal Data Analysis* **9**, 7-8. <https://odajournal.files.wordpress.com/2020/01/v9a2.pdf>

Yarnold, P.R. and Soltysik, R.C. (2016). *Maximizing Predictive Accuracy*. ODA Books. DOI: 10.13140/RG.2.1.1368.3286

## See Also

[ODArun](#) [ODAparseset.seed](#)

## Examples

```
## Not run
## NOV0boot(data=oda.list.1,run=1,predictor=8,outcome=1,seed=1234)

## Example of a moderate effect size (ESS) Novometric bootstrap confidence interval
ess <- (((14/14)+(20/64))/2)-0.5)/.5 # 31.25% ESS, a moderate effect
data <- matrix(c(20,0,44,14),ncol=2,nrow=2,dimnames=list(c(0,1),c(0,1)))
data.raw <- epitools::expand.table(data)
data.tab <- list(table(cbind(data.raw[1],data.raw[2]),dnn=c("v1","x")))
oda.list.1 <- list() # supply list formatted for NOV0boot
oda.list.1[[1]] <- do.call("list",data.tab) # supply data for NOV0boot

# NOV0boot(data=oda.list.1,run=1,predictor=1,outcome=1,seed=1234, nboot=25000)
# hist(novo.boot.1$p, main="Distribution of P-values \n by Fisher's Exact Test")
```

---

ODAclean

*Read and clean a data.csv input file and transform variables for  
ODArun()*

---

## Description

A valid .csv file is imported, cleaned, and moved to output folder. Data frame objects called `key` and `data` are loaded in the environment.

## Usage

```
ODAclean(
  data = "",
  output = "",
  miss = "",
  ipsative = "",
  normative = "",
  id = "",
```

```
    overwrite = FALSE
  )
```

### Arguments

data	The character name of the .csv file to be loaded and cleaned. The current working directory must be set to the Runs folder.
output	An integer that specifies of the Runs subdirectory folder in which to export the cleaned data. If the subdirectory does not exist, it will be created. If it does exist, the user will be asked whether the files should be overwritten.
miss	A numeric value e.g., -9 that is substituted for all NA values in the imported dataframe where missing or NA values exist.
ipsative	A character vector of variable names x in the data which will be ipsatively standardized within id groups i.e., $x - mean(x)/sd(x)$ . If ipsative standardization is desired, an id variable must be supplied.
normative	A character vector of variable names x in the cleaned data which will be normatively standardized $x - mean(x)/sd(x)$ .
id	A character vector that represents a block of id variables within which ipsative standardization can be completed. More than 1 observation per subject is needed for ipsative standardization.
overwrite	Logical value specifying whether files in output directory should be overwritten. Can be overridden by specifying TRUE or FALSE.

### Value

Cleaned data is moved to the Runs folder as both .txt and .csv files.

data.txt	A cleaned data file is moved to the output directory. Row and column names are removed from the .txt file. If specified, missing value replacements and standardized variables are also passed to this file.
data.csv	The data.csv file is moved to the output dir as a reference. The column names are maintained for use with <a href="#">ODALoad</a> and <a href="#">ODAParse</a>

### Author(s)

Nathaniel J. Rhodes

### Examples

```
# Not Run
# ODAclean(data="data.csv",output=1, miss=-9)
```

ODALoad

*Load data files and variable key for ODA***Description**

Loads the primary data file from the specified run folder and generates an ODA-friendly key for the user.

**Usage**

```
ODALoad(run = "", path = getwd(), ...)
```

**Arguments**

run	The numerical value of the folder number containing the data specified by the run number. This number will also be used to name objects uniquely by appending run to the object e.g. data.1 for Run 1. At least one value for this parameter must be supplied by the user as no defaults are supplied.
path	The working directory of the project stored. the working directory should be set above the level of the Runs folder.
...	Additional data files to load and review, if desired.

**Details**

The current working directory is stored as the path and the files to be loaded must be located in the Run folder.

**Value**

The following objects are loaded into R

data	Data frame of data.csv from specified run folder.
key	Data frame containing 2 columns: the variable names from the data and an ODA-friendly alias e.g., v1 v2

**Author(s)**

Nathaniel J. Rhodes

**Examples**

```
# Not run:
# ODALoad(1)
```

ODAManual

*Open user and function manuals.***Description**

Opens the ODA User Manual and function libraries

**Usage**

```
ODAManual()
```

**Details**

Help for using ODA package

ODAParse

*Parse ODA output files.***Description**

Parses model details, model predictions, and loads objects.

**Usage**

```
ODAParse(run = "", path = getwd(), out = "")
```

**Arguments**

run	The numerical value of the run folder in which the data file is stored
path	The working directory of the project stored as path which should be set above the level of the Runs folder
out	The character name of the output file, when not specified it is assumed to be MODEL.OUT and must be the same name as was specified in <a href="#">ODARun</a>

**Details**

When run, ODAParse will return model performance metrics and data loaded within the global environment, which can be further evaluated.

The path to the current project must be stored as path and the files to be loaded must be located in the Run folder.

Categorical ODA models are not fully supported. If a categorical model is detected, a model summary and statistics will be returned and loaded.

Multisample generalizability (GEN) analyses are partially supported. If GEN models are detected, the user will be warned.

For each ODA model, ODAParse will return the **Effect Strength for Sensitivity** or ESS and the **Effect Strength for Predictive Value** or ESP. Mean **Percent Accuracy in Classification** (PAC) and **Mean Predictive Value** (MPV) are reported, as these are a common metrics for predictive model performance.

In binary classification problems the following relationships are defined:

$$PAC = \frac{(Sensitivity + Specificity)}{2} \times 100$$

$$ESS = \frac{(PAC - 50)}{(100 - 50)} \times 100$$

$$MPV = \frac{(PPV + NPV)}{2} \times 100$$

$$ESP = \frac{(MPV - 50)}{(100 - 50)} \times 100$$

Unlike mean PAC and MPV, both ESS and ESP are normed against chance (Yarnold *et al.* 2005 and 2016) providing an intuitive scaling of model accuracy within the ODA paradigm.

The significance values reported are from 1. Monte Carlo simulations for all observations and 2. Fisher's Exact tests for LOO analysis. Depending on whether a GEN, CAT, or univariate ODA model is detected, and if LOO jackknife is performed, the exact `oda.model` output will vary.

The user is referred to the ODA User guide [ODAManual](#) and is encouraged to review the data contained within the MODEL.OUT file for more information.

#### Value

The following objects with the run number appended are returned for non categorical ODA models:

<code>oda.data</code>	Data frame based on the <code>data.csv</code> file from specified run folder.
<code>oda.key</code>	Data frame containing 2 columns: the variable names from the <code>oda.data</code> and an ODA friendly alias e.g., <code>v1 v2</code> for each attribute variable included in the ODA model.
<code>oda.list</code>	Data frame containing the confusion matrix for each ODA model contained in the <code>model.out</code> file. This output can be used to evaluate the reproducibility of the model results using a Novometric bootstrap analysis, see <a href="#">NOVObot</a> .
<code>oda.model</code>	Data frame containing the parsed model output and the ESS, ESP, and significance for ODA models.
<code>oda.perf</code>	Data frame containing the overall accuracy, sensitivity, specificity, positive predictive value, and negative predictive value for each attribute included in the ODA model. Univariate OR and 95% CI are also presented, but caution is needed with observed cell counts of zero.
<code>oda.stats</code>	Data frame containing the classification summary for each ODA model contained in the <code>model.out</code> file.
<code>oda.sda</code>	Data frame containing predictions based on the model attributes and observed classifications. Correct and incorrect classifications, e.g., False positive and false negative status, are captured for each observation in the dataset. Structural decomposition can be completed using these data in subsequent ODA models.

#### Author(s)

Nathaniel J. Rhodes



## References

Yarnold P.R. and Soltysik R.C. (2005). *Optimal data analysis: Guidebook with software for Windows*. APA Books.

Yarnold, P.R. and Soltysik, R.C. (2016). *Maximizing Predictive Accuracy*. ODA Books. DOI: 10.13140/RG.2.1.1368.3286.

## See Also

[ODAmanual](#) [ODAclean](#) [ODAlload](#) [ODArun](#) [NOV0boot](#)

## Examples

```
# Not run:
# ODAparse(1)
```

---

ODApower	<i>Estimate power for an ODA model.</i>
----------	---

---

## Description

Statistical power is estimated for a unit weighted binary application with balanced samples in each of 2 groups

## Usage

```
ODApower(n1, n2, p1, p2, comp, alpha, nsim)
```

## Arguments

n1	A numeric vector that contains the number of subjects in group 1
n2	A numeric vector that contains the number of subjects in group 2
p1	A numeric value that is the proportion of group 1 with the outcome
p2	A numeric value that is the proportion of group 2 with the outcome
comp	An integer value specifying the number of experiment wise comparisons for a Sidak type adjustment to alpha
alpha	Numeric value specifying the a priori level of significance assumed
nsim	An integer value specifying the number of Fisher's Exact Tests to simulate.

## Details

A default of 10,000 Monte Carlo Fisher's Exact tests are simulated and compared to alpha to estimate power.

The resulting power estimate represents a "worst case scenario" for the lowest level of measurement accuracy, i.e., a two class and two attribute problem, see Rhodes 2020.

For unit weighted applications, a Fisher's Exact test is isomorphic to the power of an ODA model, see Yarnold *et al.* 2005.

The *a priori* significance level alpha is adjusted based on number of comparisons (comp) as follows:  
 $\alpha_{adjusted} = 1 - (1 - \alpha)^{1/comp}$ .

**Value**

An array of power estimates with nrow of length n1 and ncol of length comp

**Author(s)**

Nathaniel J. Rhodes

**References**

Rhodes N.J. (2020). Statistical Power Analysis in ODA, CTA and Novometrics. *Optimal Data Analysis* 9, 21-25. <https://odajournal.files.wordpress.com/2020/02/v9a5.pdf>

Yarnold P.R. and Soltysik R.C. (2005). *Optimal data analysis: Guidebook with software for Windows*. APA Books.

**See Also**

[fisher.test](#) [power.fisher.test](#)

**Examples**

```
n1 <- seq(15,50,5)
n2 <- seq(15,50,5)
p1 <- 0.74
p2 <- 0.26
alpha <- 0.05
comp <- 1
nsim <- 100
#Power for an analysis with an ESS of 48% (a moderate effect)
ess <- 100*((0.74+0.74)/2)-0.5)/0.5
ODArun(n1=n1,n2=n2,comp=comp,p1=p1,p2=p2,alpha=alpha,nsim=nsim)
```

---

ODArun

*Execute an ODA model run.*

---

**Description**

Creates an command file using the parameters below and calls MegaODA

**Usage**

```
ODArun(
  run = "",
  path = getwd(),
  data = "data.txt",
  out = "model.out",
  hold = "",
  vstart = "",
  vend = "",
  class = "",
  attribute = "",
  categorical = "",
```

```

include = "",
exclude = "",
direction = "",
degen = "",
gen = "",
primary = "",
secondary = "",
nopriors = F,
miss = "",
weight = "",
mcarlo = T,
iter = "1000",
target = "",
sidak = "",
stop = "",
adjust = F,
setseed = "",
loooff = F,
overwrite = FALSE
)

```

### Arguments

run	A numerical value specifying the run folder containing the data
path	The working directory of the project stored as path
data	A character name specifying the data.txt file in the runs folder
out	A character name specifying the output file with "model.out" as the default
hold	A character name specifying the holdout data file, omitted by default
vstart	A character name specifying the start variable, see key object from <a href="#">ODALoad</a>
vend	A character name specifying the end variable, see key object from <a href="#">ODALoad</a>
class	A character name specifying the class variable, see key object from <a href="#">ODALoad</a>
attribute	A character name specifying the attributes, see key object from <a href="#">ODALoad</a>
categorical	An optional character name specifying categorical attributes, see key object from <a href="#">ODALoad</a>
include	An optional character name specifying the variable and a value of observations that are included e.g., "v2=2" or "v3>=50"
exclude	An optional character name specifying the variable and a value that are excluded
direction	An optional character name and direction e.g., $v_2 < 1$ or $v_2 \leq 1$ specifying a directional hypothesis
degen	An optional character name specifying attributes for which degenerate solutions are allowed, off by default
gen	An optional character name specifying the variable whose values denote groups for a multisample generalizability analysis, off by default
primary	An optional character vector specifying the primary criterion for choosing among optimal solutions, see <a href="#">Details</a>
secondary	An optional character vector specifying the secondary criterion for choosing among optimal solutions, see <a href="#">Details</a>

nopriors	An optional logical value specifying whether the ODA criterion is weighted by the reciprocal of class membership, with <code>nopriors = FALSE</code> as the default
miss	A numeric value specifying a missing or NA value in the data with a default value set at -9, see <a href="#">ODAclean</a>
weight	An optional character name of a variable containing a positive, non zero, weight value; cannot be the same as variables declared as <code>class attribute categorical</code> or <code>gen</code>
mcarlo	A logical value specifying whether Monte Carlo analysis should be used to estimate Type I error with <code>mcarlo = TRUE</code> as the default
iter	An integer value specifying the maximum number of Monte Carlo iterations to be reached before halting and must be specified if <code>mcarlo = TRUE</code>
target	An optional numerical value specifying the target level of $\alpha < \text{target}$ to be reached before halting and must be specified if <code>sidak</code> or <code>stop</code> are utilized
sidak	An optional integer value specifying an adjustment to target based on the number of experiment wise comparisons and must be combined with <code>target</code>
stop	An optional numerical value specifying the confidence level that Type I error is less than <code>target</code> to be reached before halting and must be combined with <code>target</code>
adjust	An optional logical value specifying whether tied Monte Carlo iterations are to be split in half with <code>adjust = FALSE</code> as the default
setseed	An optional integer value specifying a seed number for Monte Carlo analysis with the current system time as the default
loooff	An optional logical value specifying whether leave one out or LOO analysis should be turned off with <code>loooff = FALSE</code> as the default
overwrite	An optional logical value specifying whether previous files in the Run folder should be overwritten with <code>overwrite = FALSE</code> as the default

## Details

The working directory of the project is stored as path. All files needed for the run must be located in the appropriate run folder beneath this level. See [ODAclean](#).

ODArun will produce a command file with an extension of .pgm and two files with .out extensions. Resulting MODEL.OUT files can be parsed using [ODAParse](#).

The ODA algorithm explicitly maximizes the classification accuracy which is achieved for the training sample see Yarnold, 2005.

The use of Optimal in Optimal Data Analysis means that an ODA model achieves the theoretically maximum possible level of accuracy in any given application. For more information see [ODAmanual](#).

ODA utilizes primary and secondary criteria for selecting among multiple optimal solutions. By default, when not specified and when `priorsoff = FALSE` primary is set to `maxsens`. By default, when not specified secondary is set to `samplerrep`.

When `gen` is not active, other options include: `maxsens` `meansens` `samplerrep` `balanced` `distance` `random` `sens(attribute)`

When `gen` is active, other options include: `genmean` and `gensens(attribute)`

There are several disallowed specifications. Error checking is automatically performed on the user inputs. However, if the `to` keyword is used with a range of variables, it is suggested that the user confirm that the desired analysis was performed as some error checking is not available. The following cannot be combined in ODArun: `weight` cannot be both declared and listed as any of the following `class attribute` `gen`. Likewise, `gen` cannot be both declared and listed as any of the following `attribute categorical` `class` `weight`.

**Value**

Nothing is returned. Three files are created in the Run folder:

MODEL.out	The model file that contains the commands from MEGAODA syntax and analysis results, see <a href="#">ODAmanual</a> . This file is required for <a href="#">ODAparsed</a>
OUT.out	The echo file that contains the initial commands for OPEN and OUTPUT from MEGAODA syntax, see <a href="#">ODAmanual</a> .
OUT.pgm	The command file that contains all of the commands for MEGAODA passed from R based on user input to ODArun.

**Author(s)**

Nathaniel J. Rhodes

**References**

- Yarnold P.R. and Soltysik R.C. (2005). *Optimal data analysis: Guidebook with software for Windows*. APA Books.
- Yarnold, P.R. and Soltysik, R.C. (2016). *Maximizing Predictive Accuracy*. ODA Books. DOI: 10.13140/RG.2.1.1368.3286

**See Also**

[ODAmanual](#) [ODAclean](#) [ODALoad](#)

**Examples**

```
# Not run:
# ODArun(run=1, vstart="v1", vend="v45", class="v45", attribute="v1 to v44")
```

---

ODAtree

---

*Generate a folder tree for an ODA project*


---

**Description**

Establishes a subdirectory for an ODA project within a given working directory.

**Usage**

```
ODAtree(project = "NewProject", folder = getwd())
```

**Arguments**

project	A character string of a new project, e.g. "New Project"
folder	The full path to the root folder for the new project. Default is the current working directory.

**Value**

A new folder named as project containing the following subfolders:

Rscript	The folder for the Rscript file with templated syntax as a skeleton Rscript for the project.
Runs	The folder for the <a href="#">ODArun</a> outputs and data files for cleaning and analysis.
Program	The folder containing the executable program used for all <a href="#">ODArun</a> analyses.

**Examples**

```
##Not run  
##ODAtree("NewProject")
```

# Index

`.onAttach`, [2](#)

`.onLoad`, [2](#)

`fisher.test`, [10](#)

`NOVObot`, [3](#), [8](#), [9](#)

`ODAclean`, [4](#), [9](#), [12](#), [13](#)

`ODALoad`, [5](#), [6](#), [9](#), [11](#), [13](#)

`ODAmanual`, [7](#), [8](#), [9](#), [12](#), [13](#)

`ODAParse`, [3–5](#), [7](#), [12](#), [13](#)

`ODAPower`, [9](#)

`ODARun`, [3](#), [4](#), [7](#), [9](#), [10](#), [14](#)

`ODAtree`, [13](#)

`power.fisher.test`, [10](#)

`set.seed`, [3](#), [4](#)