Package ‘vdg’

October 18, 2016

Type Package

Title Variance Dispersion Graphs and Fraction of Design Space Plots

Version 1.2.0

Date 2016-10-18

Author Pieter Schoonees [aut, cre, cph], Niel le Roux [ctb]

Maintainer Pieter Schoonees <schoonees@gmail.com>

Depends R (>= 3.3.1), parallel, ggplot2, quantreg

Imports proxy, splines, gridExtra, grDevices, methods, stats, utils

Suggests rsm, Vdgraph, AlgDesign, knitr, lhs

Description Facilities for constructing variance dispersion graphs, fraction-of-design-space plots and similar graphics for exploring the properties of experimental designs. The design region is explored via random sampling, which allows for more flexibility than traditional variance dispersion graphs. A formula interface is leveraged to provide access to complex model formulae. Graphics can be constructed simultaneously for multiple experimental designs and/or multiple model formulae. Instead of using pointwise optimization to find the minimum and maximum scaled prediction variance curves, which can be inaccurate and time consuming, this package uses quantile regression as an alternative.

VignetteBuilder knitr

License GPL (>= 2)

LazyData yes

NeedsCompilation yes

RoxygenNote 5.0.1

Repository CRAN

Date/Publication 2016-10-18 10:20:56

R topics documented:

vdg-package ......................................................... 2
Description

This package provides functionality for producing variance dispersion graphs (VDGs), fraction-of-design (FDS) plots and related graphics for assessing the prediction variance properties of experimental designs. Random sampling is used to assess the distribution of the prediction variance throughout the design region. Multiple design and/or model formulae can be assessed at the same time. Graphics are produced by the \texttt{ggplot2} package.

Details

The workhorse function in the package is \texttt{spv}, which takes lists of experimental designs and/or model formulae and produces samples throughout the design region at which the prediction variance is evaluated. Depending on the type of input for the design and formula arguments, \texttt{spv} creates output objects of S3 classes \texttt{spv}, \texttt{spvlist}, \texttt{spvforlist} or \texttt{spvlistforlist}. The graphical output are obtained with the \texttt{plot} methods of these classes, and the which argument can be used to control the type of plots produced.

The design regions allowed for are typically spherical or cuboidal in nature, but the \texttt{keepfun} argument to \texttt{spv} can be used for rejection sampling. In this way nonstandard design regions can be allowed for. See also the \texttt{type} argument of \texttt{spv}. The output from the \texttt{plot} methods for objects created by \texttt{spv} are typically named lists of graphical objects created by \texttt{ggplot2}. These are best stored in an object and recreated by printing the required plot. Storing such graphical objects also enables post-hoc manipulation of the plots, such as changing the background colour by using \texttt{ggplot2}'s \texttt{theme} function.

Author(s)

Pieter C. Schoonees <schoonees@gmail.com>

References

See Also

`spv`, `plot.spv`, and `vignette(topic = "vdg")`.

---

**GJ54**

*Design from Goos & Jones*

**Description**

This data frame contains the design of Table 5.4 in Goos & Jones (2011).

**Usage**

GJ54

**Format**

a data frame of 15 runs in two variables: Time (seconds) and Temperature (Kelvin)

**References**


---

**LHS**

*Latin Hypercube Sampling*

**Description**

Different versions of latin hypercube sampling (LHS): ordinary LHS, midpoint LHS, symmetric LHS or randomized symmetric LHS. LHS is a method for constructing space-filling designs. They can be more efficient for hypercuboidal design regions than other sampling methods.

**Usage**

```r
LHS(n, m = 3, lim = c(-1, 1))
MLHS(n, m = 3, lim = c(-1, 1))
SLHS(n, m = 3, lim = c(-1, 1))
RSLHS(n, m = 3, lim = c(-1, 1))
```
Arguments

- \texttt{n} number of design points to generate
- \texttt{m} number of design factors
- \texttt{lim} limits of the coordinates in all dimensions

Value

Matrix with samples as rows.

Author(s)

Pieter C. Schoonees

References


Examples

```r
set.seed(1234)
pts <- seq(-1, 1, length = 11)

# Ordinary LHS
samp <- lhs\(n = 10, m = 2\)
plot(samp, main = "LHS")
abline(h = pts, v = pts, col = "lightgrey")

# Midpoint LHS
samp <- mlhs\(n = 10, m = 2\)
plot(samp, main = "MLHS")
abline(h = pts, v = pts, col = "lightgrey")

# Symmetric LHS
samp <- slhs\(n = 10, m = 2\)
plot(samp, main = "SLHS")
abline(h = pts, v = pts, col = "lightgrey")

# Randomized Symmetric LHS
samp <- rslhs\(n = 10, m = 2\)
plot(samp, main = "RSLHS")
abline(h = pts, v = pts, col = "lightgrey")
```
meanspv

Compute Mean Spherical SPV

Description

Computes the matrix of spherical region moments for a given model formula and a vector of radii, and uses this to calculate the mean spherical SPV for each of the radii. The function expmat calculates the matrix containing the exponents of each model factor within each model term as columns. Only simple formulae are allowed for. Only products of terms should be included in calls to I. The power operator ^ should be used instead of sqrt. Models should contain only monomial terms.

Usage

meanspv(formula, radii, FtF.inv, n)

expmat(formula)

Arguments

formula: model formula
radii: numeric vector or radii at which to calculate the matrix of spherical region moments
FtF.inv: inverse of F'F, where F is the design matrix
n: integer giving the number of design runs

Author(s)

Pieter C. Schoonees

References


Examples

f1 <- formula(~ x1*x2)
expmat(f1)
f2 <- update(f1, - . + I(x1^2) + I(x2^2))
expmat(f2)
f3 <- update(f2, - . + I(x2^0.4))
expmat(f3)
f4 <- update(f3, - . + I(x1^2):I(x2^2))
expmat(f4)
f5 <- update(f4, - . + I(x1^3*x2^0.5))
expmat(f5)
plot.spv

Plot VDGs or FDS plots

Description

Produce Variance Dispersion Graphs and/or Fraction of Design Space plots for experimental designs. There are methods for the S3 classes spv, spvlist, spvforlist and spvlistforlist – see spv.

Usage

```r
## S3 method for class 'spv'
plot(x, which = c("fds", "vdgsim", "vdgquantile", "vdgboth", 
               "boxplots"), np = 50, alpha = 7/sqrt(length(x$spv)),
       points.colour = "#39BEB1", points.size = 2, tau = c(0.05, 0.95),
       radii = 21, hexbin = FALSE, bins = 80, df = 10, lines.size = 1,
       origin = rep(0, ncol(x$sample)), method, arrange = FALSE, ...)

## S3 method for class 'spvforlist'
plot(x, which = c("fds", "vdgsim", "vdgquantile", 
               "vdgboth", "boxplots"), np = 50, alpha = 7/sqrt(length(x[[1]]$spv)),
       points.colour = "#39BEB1", points.size = 2, tau = c(0.05, 0.95),
       radii = 21, hexbin = FALSE, bins = 80, df = 10, lines.size = 1,
       origin = rep(0, ncol(x[[1]]$sample)), method, arrange = FALSE, ...)

## S3 method for class 'spvlist'
plot(x, which = c("fds", "vdgsim", "vdgquantile", "vdgboth", 
               "boxplots"), np = 50, alpha = 7/sqrt(length(x[[1]]$spv)),
       points.colour = "#39BEB1", points.size = 2, tau = c(0.05, 0.95),
       radii = 21, hexbin = FALSE, bins = 80, vrfds = FALSE, df = 10,
       lines.size = 1, origin = rep(0, ncol(x[[1]]$sample)), method,
       arrange = FALSE, ...)

## S3 method for class 'spvlistforlist'
plot(x, which = c("fds", "vdgsim", "vdgquantile", 
               "vdgboth", "boxplots"), np = 50, alpha = 7/sqrt(length(x[[1]][[1]]$spv)),
       points.colour = "#39BEB1", points.size = 2, tau = c(0.05, 0.95),
       radii = 21, hexbin = FALSE, bins = 80, df = 10, lines.size = 1,
       origin = rep(0, ncol(x[[1]][[1]]$sample)), method, arrange = FALSE, ...)
```

Arguments

- `x`: an object of type spv for a single experimental design or an object of type spvlist for multiple designs.
- `which`: either a numeric vector of integers or a character vector indicating which plots to produce. The possible plots are:
plot.spv

1 or "fds" A (variance ratio) FDS plot
2 or "vdgsim" A VDG with only the simulated prediction variance points plotted
3 or "vdgquantile" A VDG with only the quantile regression lines corresponding to tau shown
4 or "vdgboth" A combination of 2 and 3
5 or "boxplots" Parallel boxplots of the prediction variance

np scalar; the number of points to use for calculating the fraction of design space criterion.
alpha the alpha transparency coefficient for the plots
points.colour colour for points in scatterplot of SPV against the radius
points.size size for points in scatterplot of SPV against the radius
tau the tau parameter for rq (quantile regression)
radii either a numeric vector containing the radii to use for calculating the mean spherical SPV over the spherical design space, or an integer (length one vector) giving the number of radii to use for calculating the mean spherical SPV. If missing, the mean spherical SPV is not used.
hexbin logical indicating whether hexagonal binning should be used to display density instead of colour transparency
bins argument passed to stat_binhex to determine the number of hexagons used for binning.
df degrees-of-freedom parameter passed to bs
lines.size line size passed to geom_line
origin numeric vector specifying the origin of the design space
method optional; passed to dist to overwrite defaults of "Euclidean" for spherical regions or "supremum" for cubiodal regions
arrange Logical indicating whether to return a single graphical object arranging the resulting plots in a single plot window via grid.arrange, or whether to return the list of graphical objects containing the plots.
VRFDS logical indicating whether to construct a variance ratio FDS plot or not (only for class spvlist). The first design is used as reference design in case of VRFDS is TRUE
...
additional arguments passed to dist

Value

Returns a list of ggplot graphical objects (or grobs) with names corresponding to the character version of which. These plot objects can be manipulated by changing plot aesthetics and theme elements.

Author(s)

Pieter C. Schoonees
References


Examples

```r
# Single design (class 'spv')
# Larger n should be used in actual cases
library(rsm)
bbd3 <- as.data.frame(bbd(3)[,3:5])
colnames(bbd3) <- paste0("x", 1:3)
quadr <- formula(~ x1*x2*x3 - x1:x2:x3 + I(x1^2) + I(x2^2) + I(x3^2))
set.seed(1234)
out <- spv(n = 1000, design = bbd3, type = "spherical", formula = quadr)
plot(out)

# List of designs (class 'spvlist')
## Not run:
library(Vdggraph)
data(SCDH5); data(SCDDL5)
des.list <- list(SCDH5 = SCDH5, SCDDL5 = SCDDL5)
quadr5 <- formula(~ x1 + x2 + x3 + x4 + x5 + x1:x2 + x1:x3 + x1:x4 + x1:x5
                   + x2:x3 + x2:x4 + x2:x5 + x3:x4 + x3:x5 + x4:x5
                   + I(x1^2) + I(x2^2) + I(x3^2) + I(x4^2) + I(x5^2))
out2 <- spv(n = 500, design = des.list, type = "spherical", formula = quadr5)
plot(out2)

## End(Not run)

# List of formulae (class 'spvforlist')
## Not run:
fact3 <- expand.grid(x1 = c(-1,1), x2 = c(-1, 1), x3 = c(-1,1))
lin3f <- formula(~ x1 + x2 + x3)
int3f <- formula(~ (x1+x2+x3)^2)
set.seed(4312)
out3 <- spv(n = 500, design = fact3, type = "cuboidal",
           formula = list(linear = lin3f, interaction = int3f))
plot(out3)

## End(Not run)

# List of formulae and designs (class 'spvlistforlist')
## Not run:
fact3.n <- rbind(fact3, 0, 0, 0)
set.seed(4312)
out4 <- spv(n = 200, design = list(factorial = fact3, factorial.with.cntr = fact3.n),
           type = "cuboidal", formula = list(linear = lin3f, interaction = int3f))
```
**print.spv**

```r
out4
plot(out4)
```

```r
## End(Not run)
```

---

**print.spv** | *Print Method for S3 spv classes*

---

**Description**

Simple print methods for S3 classes `spv`, `spvlist`, `spvforlist` and `spvlistforlist`. See `plot.spv` for examples.

**Usage**

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

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

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

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

**Arguments**

- `x` Object of class `spv` or `spvlist`
- `...` Unimplemented

**Author(s)**

Pieter C. Schoonees

**References**

runif_cube  

Sampling for hyperspheres/hypercubes

Description

Sample uniformly in or on a hyperspheres or hypercubes.

Usage

runif_cube(n, m = 2, max.dist = 1, at = FALSE, nr.dist = 21)

runif_sphere(n, m = 2, max.radius = sqrt(m), at = FALSE, nr.rad = 21)

Arguments

- n  
  number of points to sample
- m  
  number of design factors
- max.dist  
  maximum distance from origin (L-infinity norm/supremum distance) for the hypercuboidal design region (enveloping hypercube)
- at  
  logical indicating whether to sample on concentric hyperspheres/hypercubes or not. With this option n is distributed proportionally across radii / supremum distances so that the density of samples on each concentric hypercube / hypersphere are uniform across the different hyperspheres / hypercubes.
- nr.dist  
  the number of concentric hypercubes to use in case at is TRUE
- max.radius  
  maximum radius of the hyperspherical design region (enveloping hypersphere)
- nr.rad  
  number of concentric hyperspheres to sample on in case of at being TRUE

Author(s)

Pieter C. Schoonees

References


Examples

set.seed(1234)
runif_sphere(n = 10)

set.seed(1234)
samp <- runif_sphere(n = 500, at = TRUE)
plot(samp, asp = 1)
sampler

Sampler Function

Description

This is a wrapper for the sampling functions of the \texttt{vdg} package. It extracts design properties from the design passed to it to take appropriate samples.

Usage

\begin{verbatim}
Sampler(n, design, type = c("spherical", "cuboidal", "lhs", "mlhs", "slhs", "rslhs", "custom"), at = FALSE, custom.fun = NULL, ...)
\end{verbatim}

Arguments

\begin{itemize}
\item \texttt{n} number of points to sample
\item \texttt{design} design for which the sample is required (either a matrix or data frame)
\item \texttt{type} type of design region/sampling method. One of "spherical", "cuboidal", "lhs", "mlhs", "slhs", "rslhs" or "custom". Option "custom" requires \texttt{custom.fun} to be non-NULL.
\item \texttt{at} logical; should sampling be done on the surface of hyperspheres or hypercubes? Not used for LHS methods.
\item \texttt{custom.fun} A custom sampling function, used in conjunction with \texttt{type = "custom"}. The first and second arguments must be the sample size and dimension respectively.
\item ... other arguments passed to the underlying sampling functions.
\end{itemize}

Value

Matrix with samples as rows, with S3 class \texttt{smpl}

Author(s)

Pieter C. Schoonees

References


See Also

\texttt{runif_sphere, runif_cube, LHS, MLHS, SLHS, RSLHS}
Examples

```r
## Default spherical design region
set.seed(1896)
samp1 <- sampler(n = 100, design = expand.grid(x = -1:1, y = -1:1))
plot(samp1)

## Supplying a custom sampling function based on lhs::improvedLHS()
library("lhs")
sfun <- function(n, k, dup = 1) 2 * improvedLHS(n, k, dup = dup) - 1
samp2 <- sampler(n = 100, design = expand.grid(x = -1:1, y = -1:1),
                 type = "custom", custom.fun = sfun)
plot(samp2)
```

spv

*Calculate the Scaled Prediction Variance (or SPV)*

Description

Calculates the SPV for a sample of points in a design region of specified type. Sampling is done by calling `sampler`.

Usage

```r
spv(n, design, type = "spherical", formula, at = FALSE, keepfun, sample,
    unscaled = FALSE, ...)
```

## S3 method for class 'data.frame'
```r
spv(n, design, type = c("spherical", "cuboidal", "lhs", "mlhs", "slhs", "rslhs", "custom"), formula, at = FALSE, keepfun, sample,
    unscaled = FALSE, ...)
```

## S3 method for class 'list'
```r
spv(n, design, type = c("spherical", "cuboidal", "lhs", "mlhs", "slhs", "rslhs", "custom"), formula, at = FALSE, keepfun, sample,
    unscaled = FALSE, ...)
```

## S3 method for class 'matrix'
```r
spv(n, design, type = c("spherical", "cuboidal", "lhs", "mlhs", "slhs", "rslhs", "custom"), formula, at = FALSE, keepfun, sample,
    unscaled = FALSE, ...)
```

Arguments

- `n` number of samples to take
- `design` a design or list of designs. Each design must be either a matrix or a data.frame or coercible to a data.frame.
- `type` type of sampling passed to `sampler`
formula either a single model formula of a list of formulae
at only used when type is 'spherical' or 'cuboidal'
keepfun optional; function operating on the columns of a matrix with the same number of 
columns as design which return a logical value for including a specific point in 
the sample or not. Useful for rejection sampling for nonstandard design regions.
sample optional; if not missing it should contain a matrix or data.frame containing 
points sampled over the required design region. If it is not missing, no further 
sampling will be done: the SPV is simply evaluated at these points.
unscaled logical indicating whether to use the unscaled prediction variance (UPV) instead 
of the scale prediction variance (SPV)
... additional arguments passed to sampler. This enables the used of user-specified 
sampling functions via the custom.fun argument to sampler.

Value
Object of class ‘spv’, ‘spvlist’, ‘spvforlist’ or ‘spvlistforlist’, depending on whether single de-
signs/formulas are passed or lists of these.

Author(s)
Pieter C. Schoonees

References
doi: 10.18637/jss.v074.i03.

See Also
plot.spv for more examples

Examples

```r
# Single design (class 'spv')
library(rsm)
bbd3 <- as.data.frame(bbd(3)[,3:5])
colnames(bbd3) <- paste0("x", 1:3)
quad.3f <- formula(~(x1 + x2 + x3)^2 - x1:x2:x3)
out <- spv(n = 1000, design = bbd3, type = "spherical", formula = quad.3f)
out
```
stdrange

Standardize or Unstandardize the Column Range

Description

Simple functions for rescaling a data matrix to a coded design and back. stdrange converts the design in actual measurements into a coded design, while ustdrange reverses the process (if the correct arguments are given).

Usage

stdrange(x, mins = apply(x, 2, min), maxs = apply(x, 2, max))

ustdrange(x, mins, maxs)

Arguments

x matrix containing the design, or an object coercible to a matrix.
mins vector of original values, one for each column, which should be recoded to the value -1; or which have already been recoded to -1. This and the next argument are both recycled if not of the correct length.
maxs vector of original values which should be recoded as 1, or which have already been recoded to 1.

Author(s)

Pieter C. Schoonees
# Index

<table>
<thead>
<tr>
<th>Topic</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LHS, 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hplot</td>
<td></td>
<td></td>
</tr>
<tr>
<td>plot.spv, 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>multivariate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spv, 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>package</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vdg-package, 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>print</td>
<td></td>
<td></td>
</tr>
<tr>
<td>print.spv, 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>^, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bs, 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dist, 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>expmat (meanspv), 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>geom_line, 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ggplot, 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GJ54, 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>grid.arrange, 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LHS, 3, 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>meanspv, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLHS, 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLHS (LHS), 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>plot, 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>plot.spv, 3, 6, 9, 13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>plot.spvforlist (plot.spv), 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>plot.spvlist (plot.spv), 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>plot.spvlistforlist (plot.spv), 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>print.spv, 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>print.spvforlist (print.spv), 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>print.spvlist (print.spv), 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>print.spvlistforlist (print.spv), 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rq, 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSLHS, 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSLHS (LHS), 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>runif_cube, 10, 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>runif_sphere, 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>runif_sphere (runif_cube), 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sampler, 11, 12, 13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLHS, 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLHS (LHS), 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spv, 2, 3, 6, 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sqrt, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stat_binhex, 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stdrange, 14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>theme, 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ustdrange (stdrange), 14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vdg (vdg-package), 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vdg-package, 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>