

Design Library Reference Card

Notation

d : a data frame with nice `label()`s, `level()`s, and `units()`s for variables (see the `Overview` help file for the `Hmisc` library — type `help(Overview,library='Hmisc')`)

y : an uncensored response variable

x1,x2,x3 : predictor variables (binary, factor, character, continuous)

f : a fit from a `Design` fitting function

Help : tells how to get detailed documentation on individual functions from the S command line. You can also type `help(functionname)`. In Windows/NT you can also click on the `Help` button, then click on `Design Library`. See especially the `Overview` entry — from the command line type `help(Overview, library='Design')`. When there is no `Help` comment for a function below, type `?functionname` to obtain documentation.

Setting Up

Attaching Libraries

```
.First ← function() {  
  library(Hmisc,T)      # Omit ,T under R  
  library(Design,T)  
  invisible() }
```

Data From a Fully Prepared Data Frame

```
dd ← datadist(d)          # compute data distribution summary  
options(datadist='dd')   # for plotting  
f ← ols(y ~ x1 + x2*x3, data=d)  
# Save memory by not attaching d  
. . .
```

Data from a Data Frame with Some Changes or Additions Needed

```
dd ← datadist(d)
options(datadist='dd')
attach(d)      # attach(d[d$sex=='male',]) to access a subset
levels(x1) ← list(Other=list(levels(x1)[3:5]))
# combines levels 3-5 of x1
label(x2) ← 'A better label'
ratio ← x1/x2
S ← Surv(etime, event)
dd ← datadist(dd, x1, ratio) # update dd with x1, add ratio
f ← ols(y ~ x3 + log(ratio))
g ← cph(S ~ x3*log(ratio))
...
```

Data from a Collection of Vectors

```
dd ← datadist(x1, x2, x3)
options(datadist='dd')
f ← lrm(y ~ rcs(x1,4)*x2)
```

Help: `help(Overview, library='Design')`,
`?datadist, ?Design`

Special Model Fitting Functions

`ols` : ordinary and penalized least squares

`lrm` : binary and ordinal logistic regression with optional penalization¹

`cph` : Cox proportional hazards model

`psm` : parametric survival models

`bj` : Buckey-James right-censored least squares model

`glmD` : glm version that works with Design

`glsD` : gls version that works with Design

Help : `?ols, ?lrm, ?cph, ?psm, ?bj, ?Design`

¹`lrm` fits the proportional odds model. In conjunction with the `cr.setup` function it fits the continuation ratio model.

Transformations of Predictors

`rsc(x1, 4)` : restricted cubic spline with 4 default knots

`rsc(x1, c(1,2,6,9))` : rcs with user-specified knot locations

`lsp(x1, c(1,2,6))` : linear spline (knot locations mandatory)

`pol(x1, p)` : ordinary polynomial of degree p

`scored(x1)` : expand categorical predictor having k numeric levels into linear term and $k - 2$ dummy variables

`strat(x1)` : stratify on `x1` for `cph`

many S functions : e.g., `pmin(x1,4)`, `rsc(pmax(x1, 0), 4)`; plots will have innermost variables on axes

restricted interactions : `%ia%`

Help : `?rsc` etc., `help(Design.trans, library='Design')`

Functions Operating on Fit Objects

Basic Generic Functions & Predictions

`print` : print model fit

`coef` : print coefficient vector

`fitted` : extract predicted values

`resid` : extract residuals

`formula` : print model formula

`specs` : print details about model specification (e.g., knots, categories, d.f.). Add `,long=T` to see `datadist` info.

`predict` : predicted values and confidence limits². For `ols` fits can get CLs for individuals and means.
Help: `?predict.Design`

²In `Design` all predictions are “safe” as knots and categories are remembered.

Function : build an S function that computes predicted values (the linear combination of predictors)

```
g ← Function(f)
g(x1=5:9, x2='drug') # x3 defaults to median
```

Modifying the Covariance Matrix with Optional Allowance for Intra-Cluster Correlation

bootcov : bootstrap “nonparametric” covariance matrix

robcov : Huber-White robust covariance matrix

```
# Add raw data to fit for resampling if x=T,y=T not spec. to fit
f ← update(f, x=T, y=T)
```

```
f2 ← bootcov(f, subject.id, B=100)
anova(f2)
```

```
# all functions on f2 use new covariance matrix
```

Partial Wald χ^2 and F (for ols) Statistics

```
f ← lrm(y ~ x1+x3*rcs(x2,4))
```

```
specs(f,T) # shows knots chosen for x2
```

	Assumption	Parameters	d.f.
x1	category	drug placebo	1
x3	asis		1
x2	rcspline	0.0417 0.3570 0.6898 0.9563	3
x3 * x2	interaction	linear x nonlinear - Ag(B)	3

	x1	x3	x2
Low:effect	NA	0	0.2566
Adjust to	drug	0	0.5034
High:effect	NA	1	0.7721
Low:prediction	drug	0	0.0141
High:prediction	placebo	1	0.9815
Low	drug	0	0.0059
High	placebo	1	0.9989

```
print(anova(f,x2,x3), 'names') # combined test of x2,x3
```

Factor	Chi-Square	d.f.	P
x2	20.95	6	0.0019
All Interactions	16.81	3	0.0008
Nonlinear	2.45	4	0.6543
x3	56.90	4	<.0001
All Interactions	16.81	3	0.0008
TOTAL	59.75	7	<.0001

Tested

```
x2,x2',x2'',x3 * x2,x3 * x2',x3 * x2''
x3 * x2,x3 * x2',x3 * x2''
x2',x2'',x3 * x2',x3 * x2''
x3,x3 * x2,x3 * x2',x3 * x2''
x3 * x2,x3 * x2',x3 * x2''
x3,x2,x2',x2'',x3 * x2,x3 * x2',x3 * x2''
```

```
plot(anova(f))
lrtest(f, f2)      # likelihood ratio test for nested models
```

Help: ?anova.Design

Predictor Shape Plots

```
f ← lrm(y ~ rcs(x1,4)*rcs(x2,4)+x3)
# Plot showing effect of x1 (x-axis) on log odds
# 3 curves for 3 values of x2; x3 set to mode or median
plot(f, x1=NA, x2=c(2,4,6))
# NA causes plot to plot from 10th smallest to 10th
# largest value of x1 by default. Use x1=seq(...) otherwise.

# 3-D plot varying x1 and x2. Show prob. instead of logit.
plot(f, x1=NA, x2=NA, fun=function(x)1/(1+exp(-x)),
     ylab='Prob.', method='image')
# Here could use builtin function: fun=plogis

# x3 is discrete; make separate curve for each unique value
plot(f, x1=NA, x3=NA, conf.int=F)

# Show shape and strength of all predictors, setting others
# to reference values, by using common y-axis scale.
# ref.zero shifts y to zero when x=reference value.
plot(f, ref.zero=T, ylim=c(-2,2))

# Show two kinds of CLs for ols fits
g ← ols(y ~ rcs(x1,5)+x2)
for(type in c('individual','mean'))
  plot(g, x1=NA, conf.type=type, add=type=='mean')
```

Help: ?plot.Design

Survival Estimates and Curves

For fits from `psm` and `cph` (the latter working fastest if `surv=T` was specified). `survplot` will also plot results from `survfit`. Here `etime` is a right-censored event time variable and `event` is an event/censoring indicator.

```
f ← psm(Surv(etime, event) ~ x1 + log(x2+2) + x3,
        dist='gaussian')
# Compute survival curve for x1=10, x2=3, x3='male'
survest(f, data.frame(x1=10, x2=3, x3='male'))
# Add ,times=c(2,4) to get survival only at 2 and 4 years

# Plot cumulative mortality for x1=2 and 8 for males

survplot(f, x1=c(2,8), x3='male', n.risk=T,
        fun=function(y)1-y,
        ylab='Cumulative Probability')
# x2 defaults to median
```

Help: `?survest.cph`, `?survest.psm`, `?survplot`

Charts Depicting Odds Ratios Hazard Ratios, Differences

```
summary(f) # inter-quartile range differences
           # and anti-logs
summary(f, x1=c(2,6)) # effect of increasing x1 from 2 to 6
summary(f, x1=c(2,4,6)) # set x1 to 4 when examining x2,x3
                       # important if x1 interacts
plot(summary(f), log=T) # odds ratio chart if f from lrm,
                       # log scale
```

Help: `?summary.Design`

Nomogram

```
# Obtain predicted probabilities from logistic model
# for any values of predictors in the observed range
# Override default axis for one of the variables
nomogram(f, x2=c(1,3,5,7,9),
        fun=logis, funlabel='Prob[Y=1]')
```

General Contrasts and Confidence Limits for Effects

Compare a subject with $x_1=5$ on drug to a subject with $x_1=10$ on placebo, accounting for nonlinearity and interaction.

```
f ← lrm(y ~ rcs(x1,4)*x2 + x3)
contrast(f, list(x1=5, x2='drug' ),
         list(x1=10, x2='placebo'))
```

Compute drug effects separately for several values of x_1 . Also print the average effect over these levels of x_1 , with CLs.

```
for(type in c('individual','average'))
  print(contrast(f, list(x1=1:10, x2='drug'),
                list(x1=1:10, x2='placebo'),
                type=type, conf.int=0.99))
```

Plot drug effects over values of x_1 , with error bars.

```
k ← contrast(f, list(x1=1:10, x2='drug'),
             list(x1=1:10, x2='placebo'))
xYplot(Cbind(Contrast, Lower, Upper) ~ x1, data=k,
       ylab='Drug - Placebo')      # xYplot in Hmisc
# Use Cbind(exp(Contrast),exp(Lower),exp(Upper)) to get odds ratios
```

Use “no difference” contrasts to compute estimates of mean response for $x_1=5$, $x_3=2$ averaged over treatment (x_2) groups, using observed frequencies of treatments as weights.

```
contrast(f, list(x1=5, x3=2, x2=levels(x2)),
         type='average', weights=table(x2))
```

Help: ?contrast.Design

Model Validation

`f` must contain the raw data to allow resampling. `validate` estimates the likely future performance of the model based on statistical indexes. `calibrate` does likewise for computing overfitting-corrected calibration (predicted vs. observed) curves. Below `f` must be the most full model examined. To validate a model derived from backward stepdown, specify the full model and `bw=T` to `validate`, `calibrate`.

```
f ← update(f, x=T, y=T)
validate(f, B=140)
cal ← calibrate(f, B=150)
```

Other Functions

Function	Purpose
<code>gendata</code>	Generate data for obtaining predictions
<code>fastbw</code>	Fast backward step-down var. selection
<code>sensuc</code>	Sensitivity analysis for an unmeasured confounder in <code>lrm</code> model
<code>which.influence</code>	Which obs. are overly influential
<code>latex</code>	L ^A T _E X representation of fitted model
<code>Dialog</code>	Create menu to enter predictor values and get predictions from multiple fits
<code>Hazard</code>	S function analytic representation of fitted hazard function (for <code>psm</code>)
<code>Survival</code>	S function representation of fitted survival function (for <code>psm</code> , <code>cph</code>)
<code>Quantile</code>	S function representation of fitted function for quantiles of survival time (for <code>psm</code> , <code>cph</code>)
<code>Mean</code>	S function representation of fitted function for mean survival time
<code>Varcov</code>	Compute/retrieve var-cov matrix for fit
<code>vif</code>	Variance inflation factors for fit

For More Information

The central web page for the **Design** library, for updates to this card, and for information on statistical methodology is biostat.mc.vanderbilt.edu/s/Design.html.

Please communicate corrections and improvements to Frank Harrell at f.harrell@vanderbilt.edu.

Version: March 30, 2004