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 penalization

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


\(^1\)lrm fits the proportional odds model. In conjunction with the cr.setup function it fits the continuation ratio model.
Transformations of Predictors

rcs(x1, 4) : restricted cubic spline with 4 default knots
rcs(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), rcs(pmax(x1, 0), 4); plots will have innermost variables on axes

restricted interactions : %ia%

Help : ?rcs 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 limits\(^2\). For ols fits can get CLs for individuals and means.

\(^2\)In 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)

\[ g \leftarrow \text{Function}(f) \]
\[ g(x1=5:9, x2='drug') \quad \# x3 \text{ defaults to median} \]

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
\[ f \leftarrow \text{update}(f, x=T, y=T) \]

\[ f2 \leftarrow \text{bootcov}(f, \text{subject.id}, B=100) \]
\[ \text{anova}(f2) \]
\# all functions on f2 use new covariance matrix

Partial Wald $\chi^2$ and $F$ (for ols) Statistics

\[ f \leftarrow \text{lrm}(y \sim x1+x3*\text{rcs}(x2,4)) \]

\[ \text{specs}(f,T) \quad \# \text{shows knots chosen for } x2 \]

<table>
<thead>
<tr>
<th>Assumption</th>
<th>Parameters</th>
<th>d.f.</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1 category</td>
<td>drug</td>
<td>1</td>
</tr>
<tr>
<td>x3 asis</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>x2 rcspline</td>
<td>0.0417</td>
<td>0.3570</td>
</tr>
<tr>
<td>x3 * x2 interaction</td>
<td>linear</td>
<td>x nonlinear</td>
</tr>
</tbody>
</table>

\[ x1 \quad x3 \quad x2 \]
\[ \text{Low:effect} \quad NA \quad 0 \quad 0.2566 \]
\[ \text{Adjust to drug} \quad 0 \quad 0.5034 \]
\[ \text{High:effect} \quad NA \quad 1 \quad 0.7721 \]
\[ \text{Low:prediction drug} \quad 0 \quad 0.0141 \]
\[ \text{High:prediction placebo} \quad 1 \quad 0.9815 \]
\[ \text{Low drug} \quad 0 \quad 0.0059 \]
\[ \text{High placebo} \quad 1 \quad 0.9989 \]

\[ \text{print(anova}(f,x2,x3),'names') \quad \# \text{combined test of } x2,x3 \]
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 * x2,x3 * x2',x3 * x2''
x3,x2,x2',x2'',x3 * x2',x3 * x2''
x3,x2,x2',x2'',x3 * x2',x3 * x2''
x3,x2,x2',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 built-in 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 \textit{psm} and \textit{cph} (the latter working fastest if \texttt{surv=T} was specified). \textit{survplot} will also plot results from \textit{survfit}. Here \texttt{etime} is a right-censored event time variable and \texttt{event} is an event/censoring indicator.

\begin{verbatim}
f ← \texttt{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
\end{verbatim}

Help: \texttt{?survest.cph}, \texttt{?survest.psm}, \texttt{?survplot}

Charts Depicting Odds Ratios
Hazard Ratios, Differences

\begin{verbatim}
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 \texttt{lrm},
# log scale
\end{verbatim}

Help: \texttt{?summary.Design}

Nomogram

\begin{verbatim}
# 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=plogis, funlabel='Prob[Y=1]')
\end{verbatim}
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.

```r
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.

```r
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.

```r
c ← contrast(f, list(x1=1:10, x2='drug'), list(x1=1:10, x2='placebo'))
xYplot(Cbind(Contrast, Lower, Upper) ~ x1, data=c, ylab='Drug - Placebo') # xYplot in Hmisc
```

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.

```r
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`.

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

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

### 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](http://biostat.mc.vanderbilt.edu/s/Design.html).

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

Version: March 30, 2004