Hmisc Library Reference Card

Non-Graphics Functions

Category	Functions
Aggregating & Grouping Data	bystats, bystats2, cut2, dataRep, mApply, summary.formula, xy.group, summarize
Attributes	comment, label, units, contents
Computations	matxv, nomiss, rcspline.eval, rcspline.restate, solvet, trap.rule
Data Manipulation, Recoding	combine.levels, Cs, cut2, first.word, interaction, Lag, recode, reShape, score.binary, sedit, strmatch, subset, substi, tex, upData
Descriptive & Summary Stats	dataRep, describe, wtd.mean, wtd.var, wtd.rank, wtd.loess.noiter
File Import	cleanup.import, csv.get, getHdata, sas.get, spss.get, sasxport.get
Missing Data	aregImpute, fit.mult.impute, impute is.imputed, is.present, transcan
Modeling	areg.boot, rm.boot
Multivariate	pc1, rm.boot, transace, transcan, varclus
Nonparametric Methods	bootkm, hdquantile, hoeffd, plsmo, rcorr, rcorr.cens, rcorr.censp, somers2, spearman, spearman.test, spearman2, wtd.rank
Power, Sample Size, CLs	ballocation, binconf, bpower, bsamsize, ciapower, confbar, cpower, ftupwr, ftuss, ldBands, popower, rMultinom, samplesize.bin, spower
Predictive Accuracy	abs.error.pred, somers2
Programming	do, %in%, %nin%
Statistical Inference	deff, gbayes, james.stein, plotCorrPrecision, t.test.cluster
Survival Analysis, Censoring	bootkm, event.chart, event.history, rcorr.cens, rcorrp.cens
Table Making, LaTeX	format.df, latex, summary.formula, tex

Non-Graphics Functions, *continued*

Category	Functions
Utilities	all.is.numeric, approxExtrap, calltree, contents, ddmmmyy, eip, find.matches, list.tree, mask, matchCases, prn, src, store, sys, uncbind

Graphics Functions

Category	Functions
High-Level (Non-Trellis)	bpplot, datadensity, dot.chart, drawPlot, ecdf, errbar, event.chart, event.history, histbackback, hist.data.frame, mulbar.chart, plsmo, p.sunflowers, rcspline.plot, symbol.freq
Plot Annotations	labcurve, minor.tick, mtitle, pstamp, putKey, putKeyEmpty
Plot Setup & Devices	gs.slide, mgp.axis, mgp.axis.labels, mgp.axis.labels, ps.slide, setpdf, setps, setTrellis, trellis.strip.blank, win.slide, zoom
Plot/Symbol Info	character.table, show.col, show.pch
Plot Symbols & Additions	confbar, histSpike, scat1d
Trellis High-Level Graphics	Dotplot, ecdf, panel.bpplot, panel.plsmo, xYplot

Online Help

Use `?functionname` or `help(functionname)` from the command line to get detailed documentation on (most of the) individual functions. In S-PLUS on Windows you can also click on the `Help` button, then click on `Hmisc Library`. See especially the `Overview` entry — from the command line type `help(Overview, library='Hmisc')`. Type `help('data.frame.create.modify.check',lib='Hmisc')` to see a detailed template the data management steps described below.

Attaching Library

```
.First ← function()
   invisible(library(Hmisc,T)) # Omit ,T for R
```

You can attach Hmisc from the File menu in Windows but check the box Attach at top of search list (for S-PLUS).

Creating and Modifying Data Frames

The typical order of operations is as follows. SPn refers to search position n.

1. Import external data into a data frame (while _Data is in SP1)
2. Make global changes to a data frame (e.g., changing variable names) (_Data still in SP1)
3. Change attributes or values of variables within a data frame (_Data or data frame in SP1)
4. Do analyses involving the whole data frame (without attaching it) (Data frame still in _Data)
5. Do analyses of individual variables (after attaching the data frame in SP2 or later)

Hmisc functions respect the 'units' (for describe, Surv) and 'label' attribute (for describe, summary.formula, and high-level plotting functions).
Here are comments for creating a data frame Z by importing from an ascii file.

Step 1: Create initial draft of data frame

```
# S-Plus:
import.data(FileName = "/windows/temp/z.asc",
            FileType = "ASCII", DataFrame = "Z")
Z ← importData("/tmp/z.asc",  # depending on S version
     colNames=Cs(id,age,fev,height,sex,smoking))
# R: use csv.get, sas.get, sasxport.get, spss.get, read.csv, etc.
```

Step 2: Clean up data frame / make it be more efficiently stored

```
# Needed if not using an Hmisc import function
Z ← cleanup.import(Z)
```

Step 3: Make global changes to the data frame

```
names(Z) ← casefold(names(Z)) # names to lower case
names(Z)[6] ← 'smoke'   # assumes you know the positions!
names(Z)[names(Z)=='smoking'] ← 'smoke'
names(Z) ← edit(names(Z))
```

Step 4: Delete unneeded variables

```
Z$x1 ← NULL
Z[c('age','sex')] ← NULL   # delete 2 variables
Z[Cs(age,sex)]    ← NULL   # same thing
```

Step 5: Make changes to only a few variables without attaching

```
Z$sex   ← factor(Z$sex,   0:1, c('female','male'))
Z$smoke ← factor(Z$smoke, 0:1,
          c('non-current smoker','current smoker'))
units(Z$age)    ← 'years'
units(Z$fev)    ← 'L'
label(Z$fev)    ← 'Forced Expiratory Volume'
units(Z$height) ← 'inches'
```

Method using attach (not recommended)

```
attach(Z, pos=1, use.names=F)  # R does not have use.names
sex   ← factor(sex,   0:1, c('female','male'))
smoke ← factor(smoke, 0:1, c('non-current smoker',
                             'current smoker'))
units(age)    ← 'years'
units(fev)    ← 'L'
label(fev)    ← 'Forced Expiratory Volume'
units(height) ← 'inches'
# height ← NULL here would cause height to be deleted from Z
detach(1, 'Z2')          # MAKE SURE DATA FRAME NAME QUOTED!
```

Better approach for steps 3-5

```
Z2 <- upData(Z, rename=c(smoking='smoke'),
  drop=c('var1','var2'),
  levels=list(sex  =list(female=0,male=1),
              smoke=list('non-current smoker'=0,
                         'current smoker'=1)),
  units =c(age='years', fev='L', height='inches'),
  labels=c(fev='Forced Expiratory Volume'))
# or upData(Z, sex=factor(sex,0:1,c('female','male'))), etc.
# Note: upData like cleanup.import stores variables
# as efficiently as possible
```

Checking and Inspecting Data

```
page(describe(Z2), multi=T)
# multi=T allows that window to persist while
# control is returned to other windows

Z ← Z2; rm(Z2)     # once verify Z2 OK
```

Next, we can use a variety of other functions to check and describe all of the variables. As we are analyzing all or almost all of the variables, this is best done without attaching the data frame. Note that `plot.data.frame` plots inverted CDFs for continuous variables and dot plots showing frequency distributions of categorical ones.

```
summary(Z)
# basic summary function (summary.data.frame)
plot(Z)                    # plot.data.frame
datadensity(Z)             # rug plots and freq. bar charts for all var
hist.data.frame(Z)         # for R, just say hist(Z)

by(Z, Z$smoke, describe)    # stratified describe
```

Detailed Analyses Involving Individual Variables

Analyses based on the formula language can use data= so attaching the data frame may not be required. This saves memory. Here we use the `Hmisc summary.formula` function to compute 5 statistics on height, stratified separately by age quartile and by sex.

```
options(width=80)
summary(height ~ age + sex, data=Z,
        fun=function(y)c(smean.sd(y),
             smedian.hilow(y,conf.int=.5)))
# This computes mean height, S.D., median, outer quartiles

fit ← lm(height ~ age*sex, data=Z)
summary(fit)
```

For this analysis we could also have attached the data frame in search position 2. For other analyses, it is mandatory to attach the data frame unless `Z$` prefixes each variable name. Important: DO NOT USE `attach(Z, 1)` or `attach(Z, pos=1, ...)` if you are only analyzing and not changing the variables, unless you really need to avoid conflicts with variables in search position 1 that have the same names as the variables in `Z`. Attaching into search position 1 will cause S to be more of a memory hog.

```
attach(Z)
# Use e.g. attach(Z[,Cs(age,sex)]) if you only
# want to analyze a small subset of the variables
# Use e.g. attach(Z[Z$sex=='male',]) or
# attach(subset(Z, sex=='male')) or
# attach(subset(Z, sex=='male', c('age','race')))
# to analyze a subset of the observations

summary(height ~ age + sex,
        fun=function(y)c(smean.sd(y),
                         smedian.hilow(y,conf.int=.5)))
fit ← lm(height ~ age*sex)

# Run generic summary function on height and fev,
# stratified by sex
by(data.frame(height,fev), sex, summary)

# Cross-classify into 4 sex x smoke groups
by(Z, list(sex,smoke), summary)

# Plot 5 quantiles
s ← summary(fev ~ age + sex + height,
      fun=function(y)quantile(y,c(.1,.25,.5,.75,.9)))

plot(s, which=1:5, pch=c(1,2,15,2,1),
     main='A Discovery', xlab='Z')
# pch=c('=','[','o',']','='),

# Use the nonparametric bootstrap to compute a
# 0.95 confidence interval for the population mean fev
smean.cl.boot(fev)     # in Hmisc

# Use the Statistics ... Compare Samples ... One Sample
# menus to get a normal-theory-based C.I.  Then do it
# more manually.  The following method assumes that
# there are no NAs in fev

sd ← sqrt(var(fev))
xbar ← mean(fev)
xbar
sd
n ← length(fev)
qt(.975,n-1)
# prints 0.975 critical value of t dist. with n-1 d.f.

xbar + c(-1,1)*sd/sqrt(n)*qt(.975,n-1)
# prints confidence limits

# Fit a linear model
fit ← lm(fev ~ other variables ...)
```

```
detach()
```

The last command is only needed if you want to start operating on another data frame and you want to get Z out of the way.

For More Information

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

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

Version: March 30, 2004