

## New Functions in the R Hmisc Package

### New Functions in the R Hmisc Package

[summaryM](#)

[summaryRc](#)

[summaryD](#)

[bpplotM](#)

[summaryP](#)

[summaryS](#)

[tabulr](#)

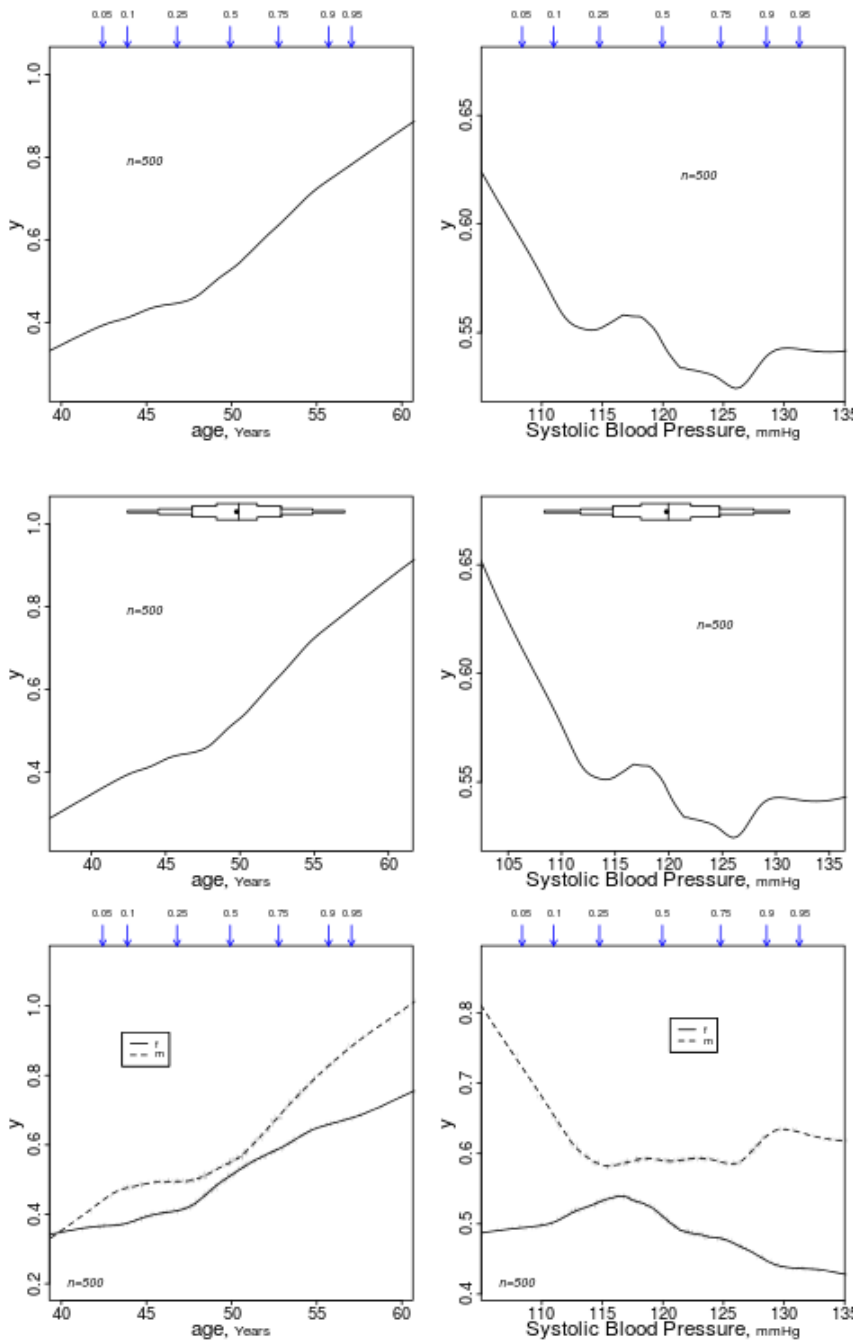
### summaryM

In place of `summary(group ~ a + b + c, method='reverse')` (which calls `summary.formula`) use `summaryM(a + b + c ~ group)`

### summaryRc

This is for graphing lowess nonparametric trend lines. This provides a much better summary than `summary.formula method = "response"` which in the example below would by default categorize age and bp into quartiles in order to get simple proportions of y.

```
set.seed(177)
sex <- factor(sample(c("m", "f"), 500, rep=TRUE))
age <- rnorm(500, 50, 5)
bp <- rnorm(500, 120, 7)
units(age) <- 'Years'; units(bp) <- 'mmHg'
label(bp) <- 'Systolic Blood Pressure'
L <- .5*(sex == 'm') + 0.1 * (age - 50)
y <- rbinom(500, 1, plogis(L))
png('/tmp/summaryRc.png', height=750)
spar(mfrow=c(3,2), top=2, cex.axis=1)
summaryRc(y ~ age + bp)
# For x limits use 1st and 99th percentiles to frame extended box plots
summaryRc(y ~ age + bp, bpplot='top', datadensity=FALSE, trim=.01)
summaryRc(y ~ age + bp + stratify(sex),
          label.curves=list(keys='lines'), nloc=list(x=.1,
y=.05))
dev.off()
```

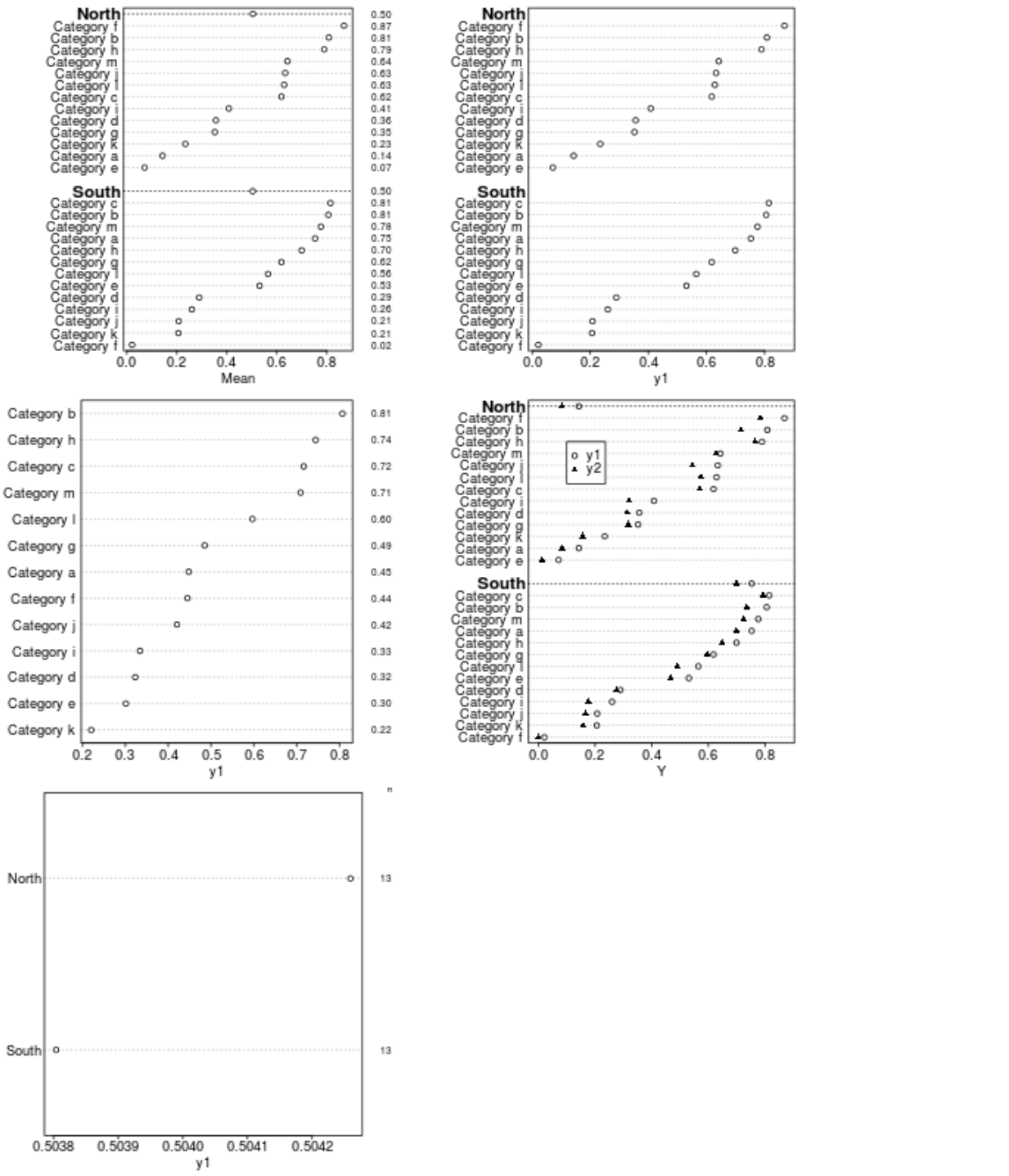


## summaryD

summaryD is for summarizing data using a user-specified function and produce a dot chart using the Hmisc dotchart3 function. This allows for major and minor categories, all in one panel.

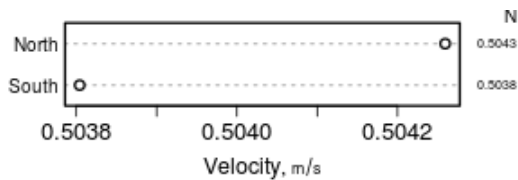
```
set.seed(135)
maj <- factor(c(rep('North',13),rep('South',13)))
g <- paste('Category',rep(letters[1:13],2))
n <- sample(1:15000, 26, replace=TRUE)
y1 <- runif(26)
y2 <- pmax(0, y1 - runif(26, 0, .1))
png('/tmp/summaryD.png', width=550, height=800)
spar(mfrow=c(3,2))
f <- function(x) sprintf('%4.2f', x)
summaryD(y1 ~ maj + g, xlab='Mean', auxtitle='', fmtvals=f)
summaryD(y1 ~ maj + g, groupsummary=FALSE)
summaryD(y1 ~ g, fmtvals=f, auxtitle='')
Y <- cbind(y1, y2)
summaryD(Y ~ maj + g, fun=function(y) y[1,], pch=c(1,17))
rlegend(.1, 26, c('y1', 'y2'), pch=c(1,17))

summaryD(y1 ~ maj, fun=function(y) c(mean(y), n=length(y)),
          auxvar='n')
dev.off()
```



```

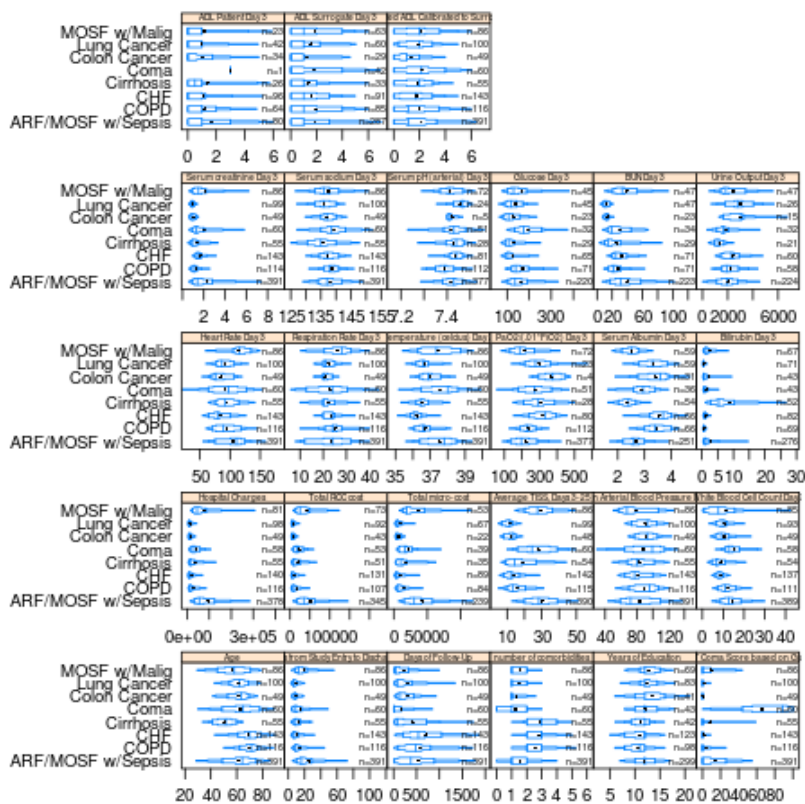
png('/tmp/summaryD2.png', width=300, height=100)
# Or: pdf('/tmp/z.pdf', width=3.5, height=1.25)
spar()
summaryD(y1 ~ maj, fmvls=function(x) round(x,4),
          xlab=labelPlotmath('Velocity', 'm/s'))
dev.off()
    
```



## bpplotM

This is for graphing extended box plots for multiple variables

```
getHdata(support)
# Automatically analyze all numeric variables with more than 5 unique values
bpplotM(data=support, groups='dzgroup', cex.strip=.4, cex.means=.3, cex.n=.45)
```



bpplotM also supports an R formula as the first argument, e.g. `bpplotM(age + weight + height ~ treatment * sex)`

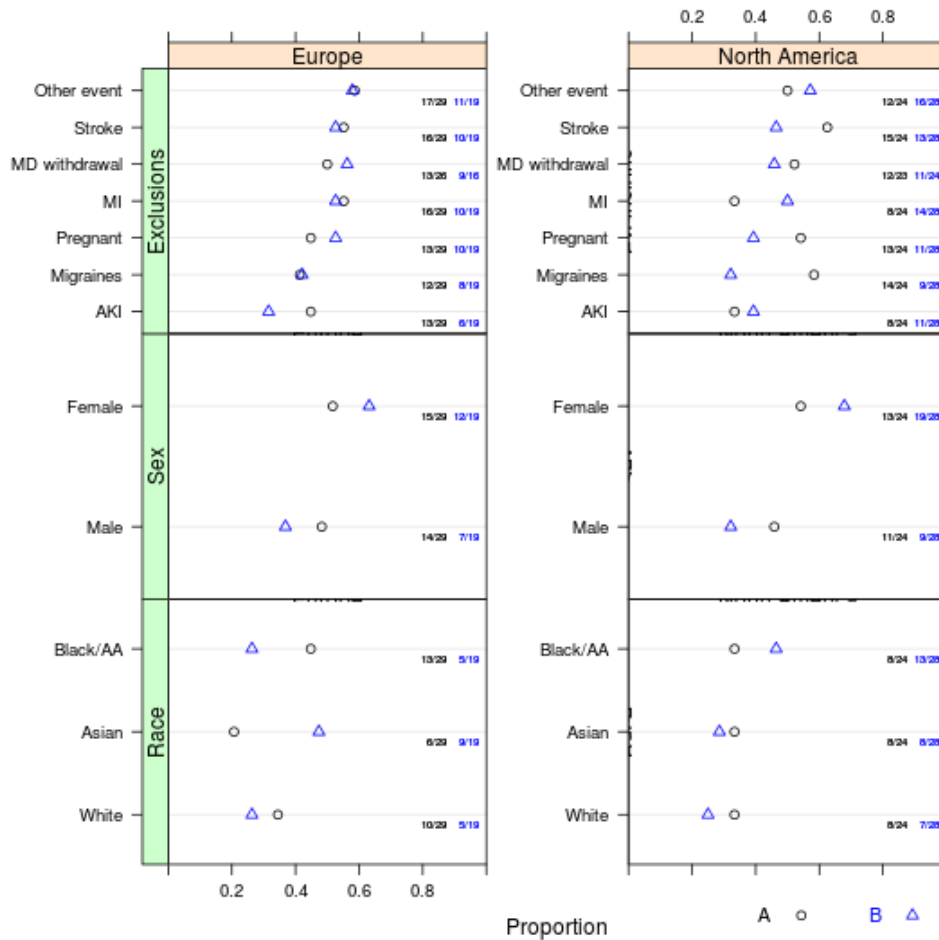
## summaryP

This is like bpplotM but for purely categorical data. It produces a "tall and thin" data frame with numerator and denominator frequencies. A plot method makes multi-panel dot plots using the R `trellis` `dotplot` function with a special `panel` function, to depict proportions by a series of cross-classifying variables, plus optionally a

superpositioning variable groups. `summaryP` provides special support for a series of "checklist" yes/no variables through an internal function `yn`. For `yn`, a positive response is taken to be `y`, `yes`, `present` (ignoring case) or a logical `TRUE`.

```
n <- 100
f <- function(na=FALSE) {
  x <- sample(c('N', 'Y'), n, TRUE)
  if(na) x[runif(100) < .1] <- NA
  x
}
set.seed(1)
d <- data.frame(x1=f(), x2=f(), x3=f(), x4=f(), x5=f(), x6=f(), x7=f(TRUE),
               age=rnorm(n, 50, 10),
               race=sample(c('Asian', 'Black/AA', 'White'), n, TRUE),
               sex=sample(c('Female', 'Male'), n, TRUE),
               treat=sample(c('A', 'B'), n, TRUE),
               region=sample(c('North America', 'Europe'), n, TRUE))
d <- upData(d, labels=c(x1='MI', x2='Stroke', x3='AKI', x4='Migraines',
                       x5='Pregnant', x6='Other event', x7='MD withdrawal',
                       race='Race', sex='Sex'))
dasna <- subset(d, region=='North America')
with(dasna, table(race, treat))

png('/tmp/summaryP.png', width=550, height=550)
s <- summaryP(race + sex + yn(x1, x2, x3, x4, x5, x6, x7, label='Exclusions')
~
               region, data=d)
# add exclude1=FALSE above to include female category
plot(s, val ~ freq | region * var, groups='treat') # best looking
# The above uses the latticeExtra package's useOuterStrips function to improve
output
# Default output without using latticeExtra: plot(s, groups='treat',
outerlabels=FALSE)
dev.off()
```



Also try:

```
plot(s, groups='treat')
# plot(s, groups='treat', outerlabels=FALSE) for standard lattice output
plot(s, groups='region', key=list(columns=2, space='bottom'))
plot(summaryP(race + sex ~ region, data=d, exclude1=FALSE), col='green')
```

## summaryS

`summaryS` summarizes multiple response variables and makes multipanel scatter or dot plots. An optional `fun` argument can specify a wide variety of summary statistics to compute.

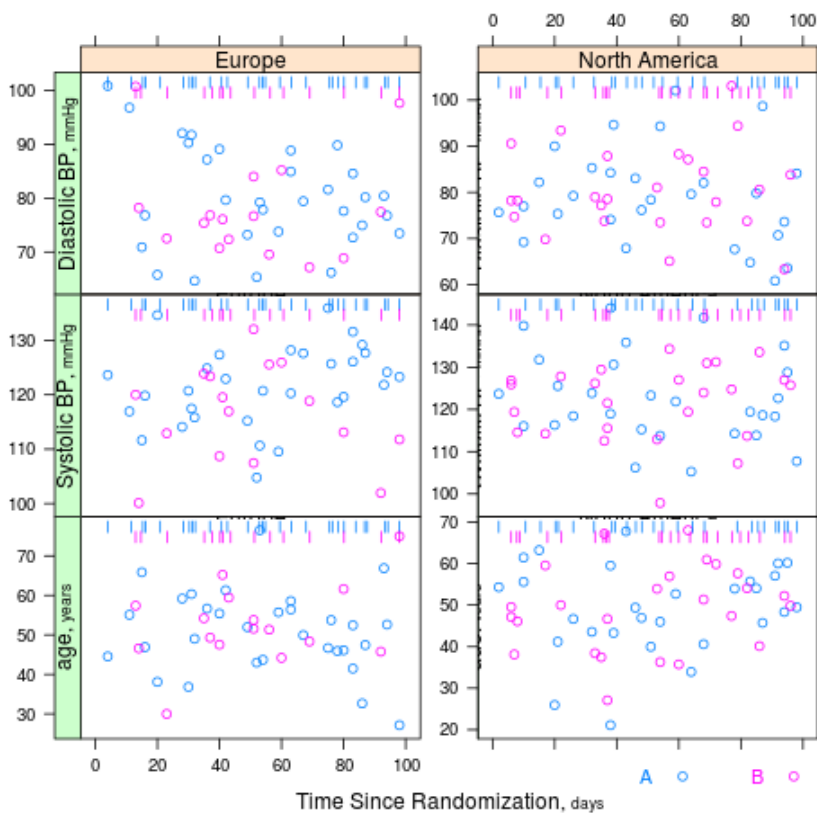
```

require(Hmisc)
n <- 100
set.seed(1)
d <- data.frame(sbp=rnorm(n, 120, 10),
                dbp=rnorm(n, 80, 10),
                age=rnorm(n, 50, 10),
                days=sample(1:n, n, TRUE),
                S1=Surv(2*runif(n)), S2=Surv(runif(n)),
                race=sample(c('Asian', 'Black/AA', 'White'), n, TRUE),
                sex=sample(c('Female', 'Male'), n, TRUE),
                treat=sample(c('A', 'B'), n, TRUE),
                region=sample(c('North America', 'Europe'), n, TRUE),
                meda=sample(0:1, n, TRUE), medb=sample(0:1, n, TRUE))

d <- upData(d, labels=c(sbp='Systolic BP', dbp='Diastolic BP',
                       race='Race', sex='Sex', treat='Treatment',
                       days='Time Since Randomization',
                       S1='Hospitalization', S2='Re-Operation',
                       meda='Medication A', medb='Medication B'),
            units=c(sbp='mmHg', dbp='mmHg', age='years', days='days'))

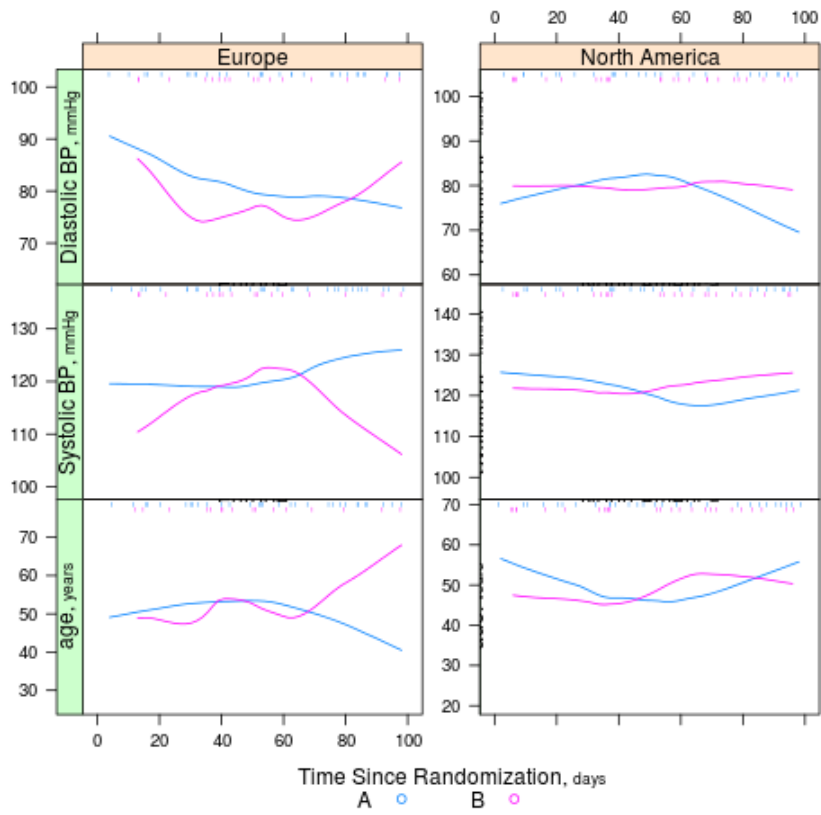
s <- summaryS(age + sbp + dbp ~ days + region + treat, data=d)
# plot(s) # 3 pages
plot(s, groups='treat', datadensity=TRUE,
     scat1d.opts=list(lwd=.5, nhistSpike=0))

```

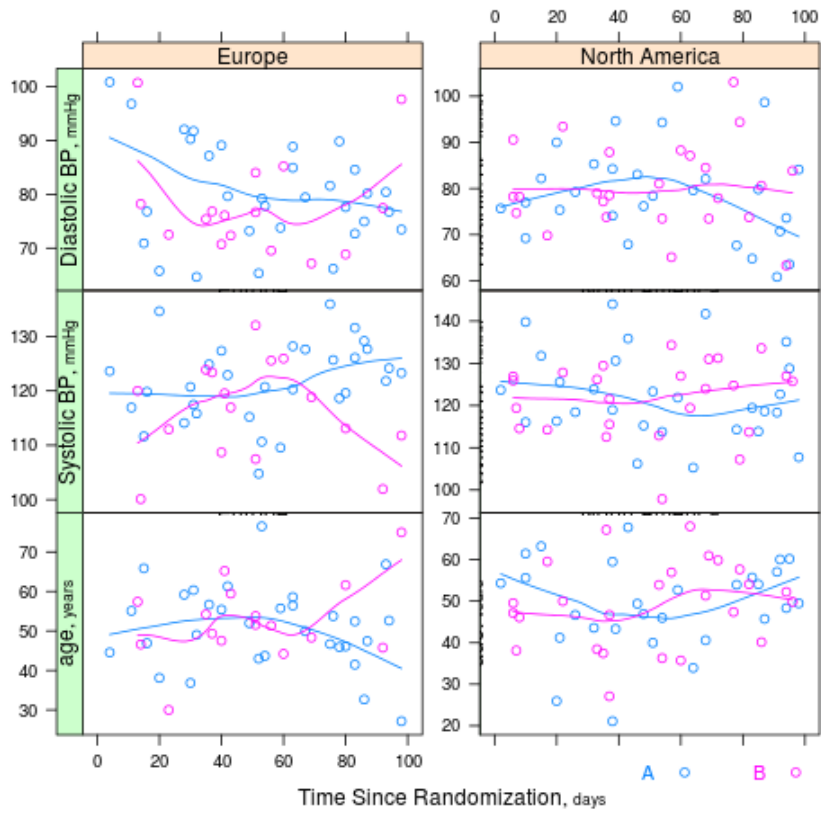




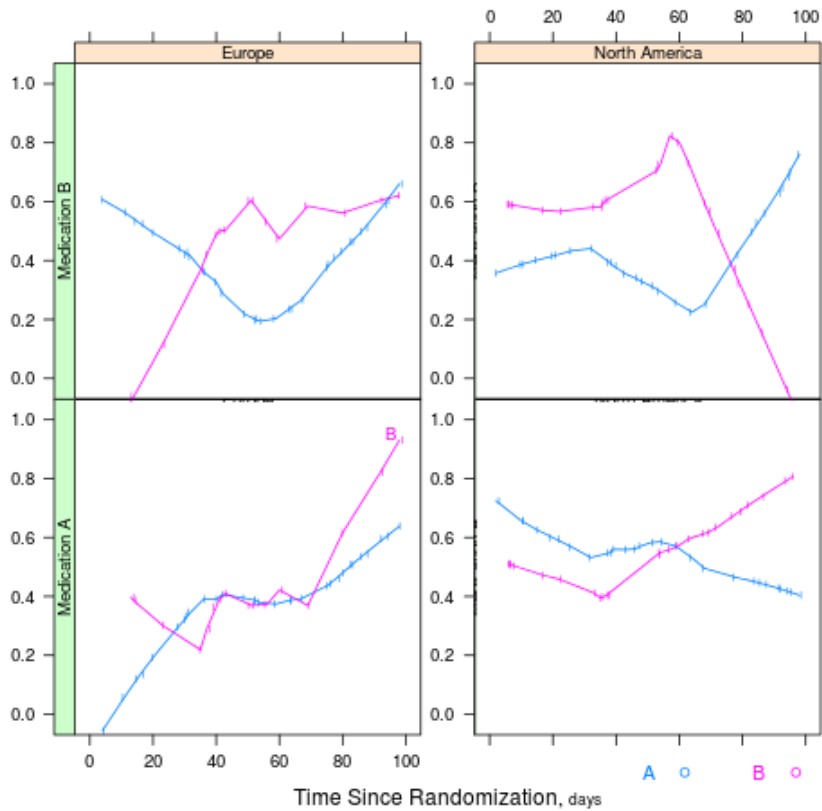
```
plot(s, groups='treat', panel=panel.loess,
     key=list(space='bottom', columns=2),
     datadensity=TRUE, scatld.opts=list(lwd=.5))
```



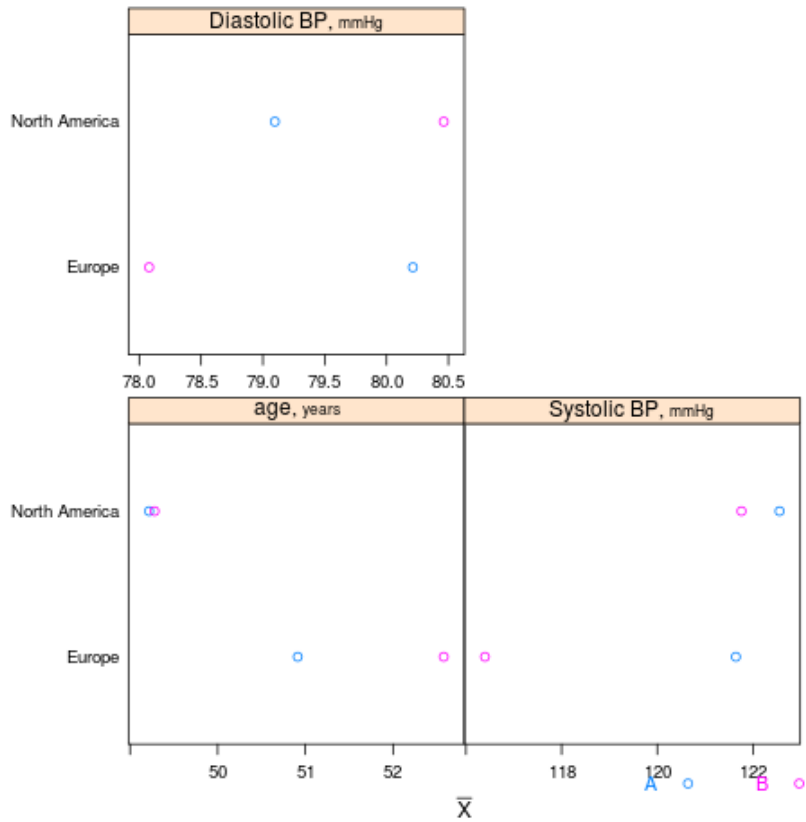
```
plot(s, groups='treat',
     panel=function(...) {panel.xyplot(...); panel.loess(...)})
```



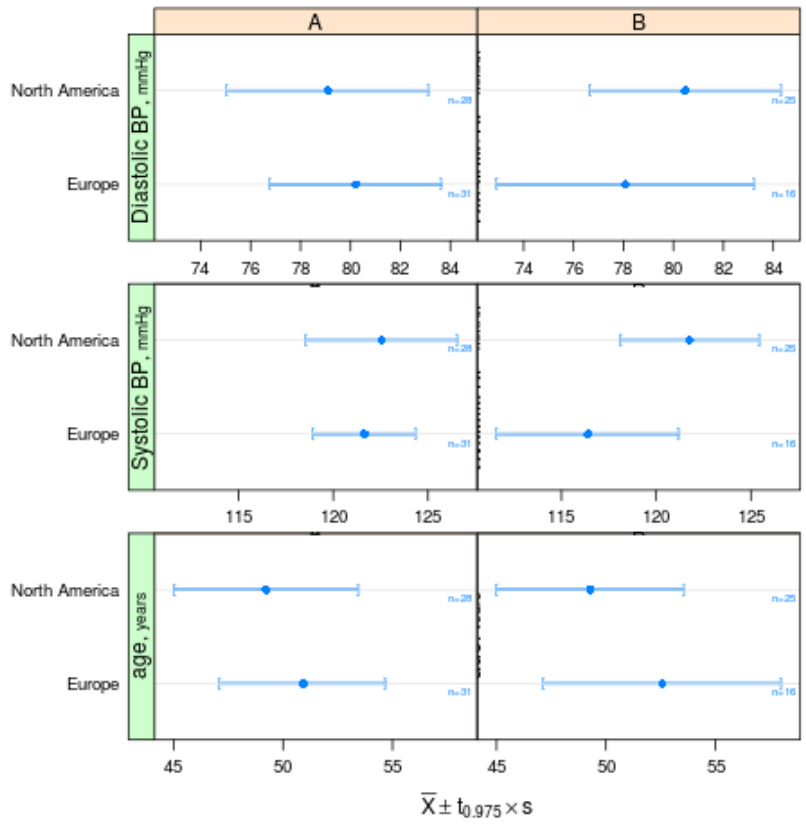
```
plot(s, groups='treat', panel=pan, paneldoesgroups=TRUE,  
     scatld.opts=list(lwd=.7), cex.strip=.8)
```



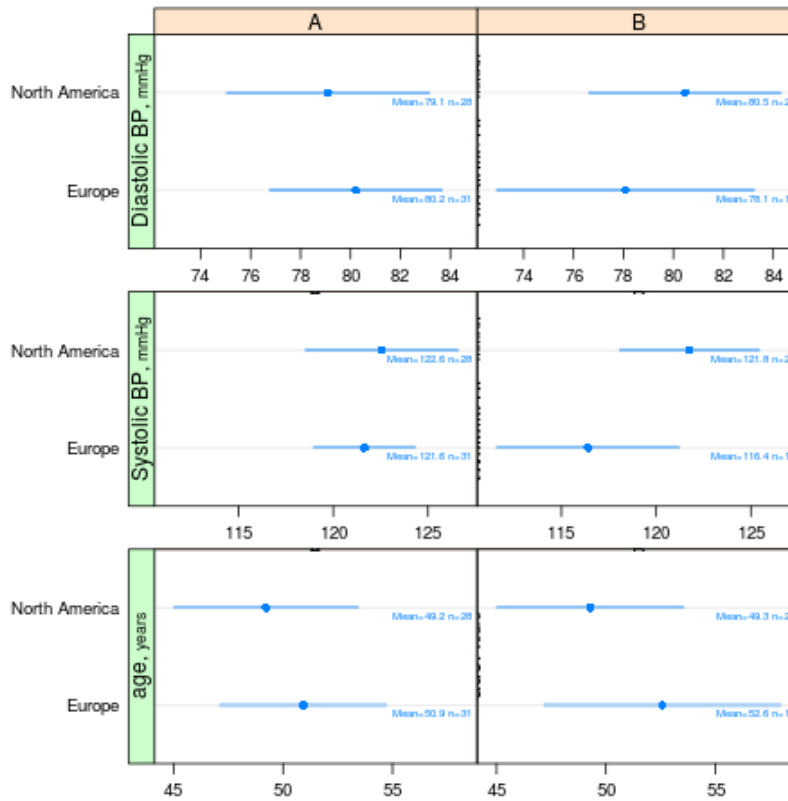
```
s <- summaryS(age + sbp + dbp ~ region + treat, data=d, fun=mean)
plot(s, groups='treat', funlabel=expression(bar(X)))
```



```
# Compute parametric confidence limits for mean, and include sample sizes
f <- function(x) {
  x <- x[! is.na(x)]
  c(smean.cl.normal(x, na.rm=FALSE), n=length(x))
}
s <- summaryS(age + sbp + dbp ~ region + treat, data=d, fun=f)
# Draw [ ] for lower and upper confidence limits in addition to thick line
plot(s, funlabel=expression(bar(X) %+-% t[0.975] %*% s),
     pch.stats=c(Lower=91, Upper=93)) # type show.pch() to see defs.
```



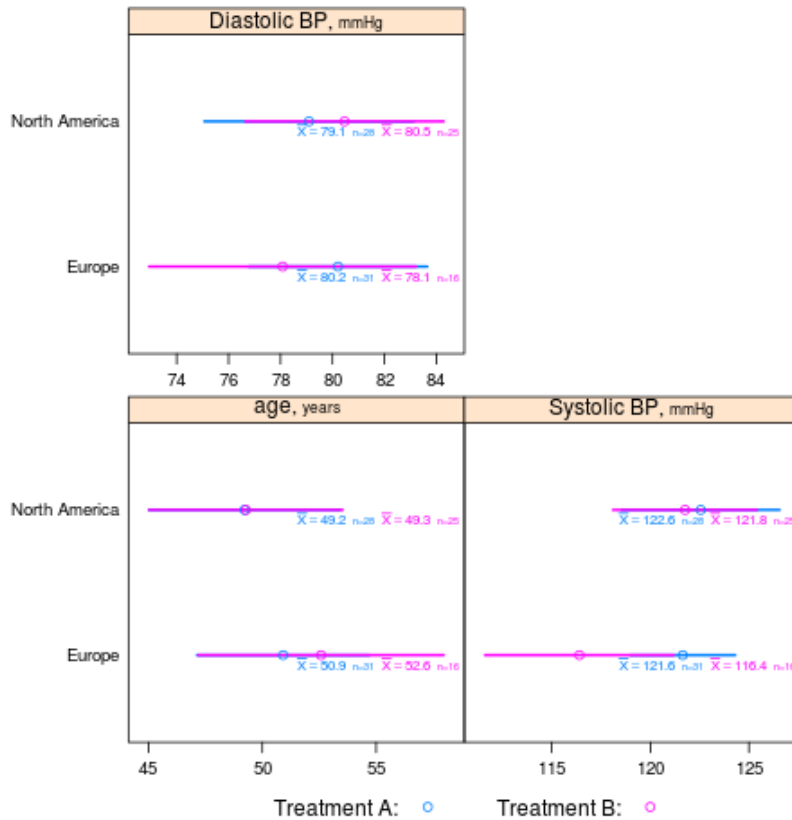
```
plot(s, textonly='n', textplot='Mean', digits=1)
```



```

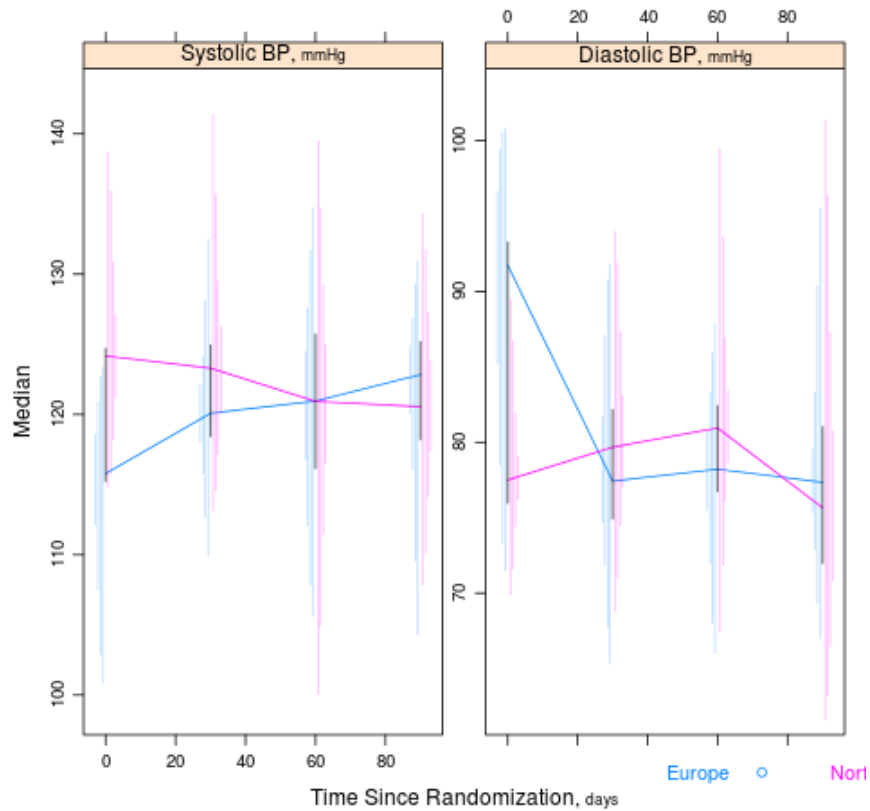
# Customize printing of statistics to use X bar symbol and smaller
# font for n=...
cust <- function(y) {
  means <- format(round(y[, 'Mean'], 1))
  ns <- format(y[, 'n'])
  simplyformatted <- paste('X=', means, ' n=', ns, ' ', sep='')
  s <- NULL
  for(i in 1:length(ns)) {
    w <- paste('paste(bar(X)==', means[i], ' ,~~scriptstyle(n==', ns[i],
              '))', sep='')
    s <- c(s, parse(text=w))
  }
  list(result=s,
        longest=simplyformatted[which.max(nchar(simplyformatted))])
}
plot(s, groups='treat', cex.values=.65,
      textplot='Mean', custom=cust,
      key=list(space='bottom', columns=2,
              text=c('Treatment A:', 'Treatment B:')))

```



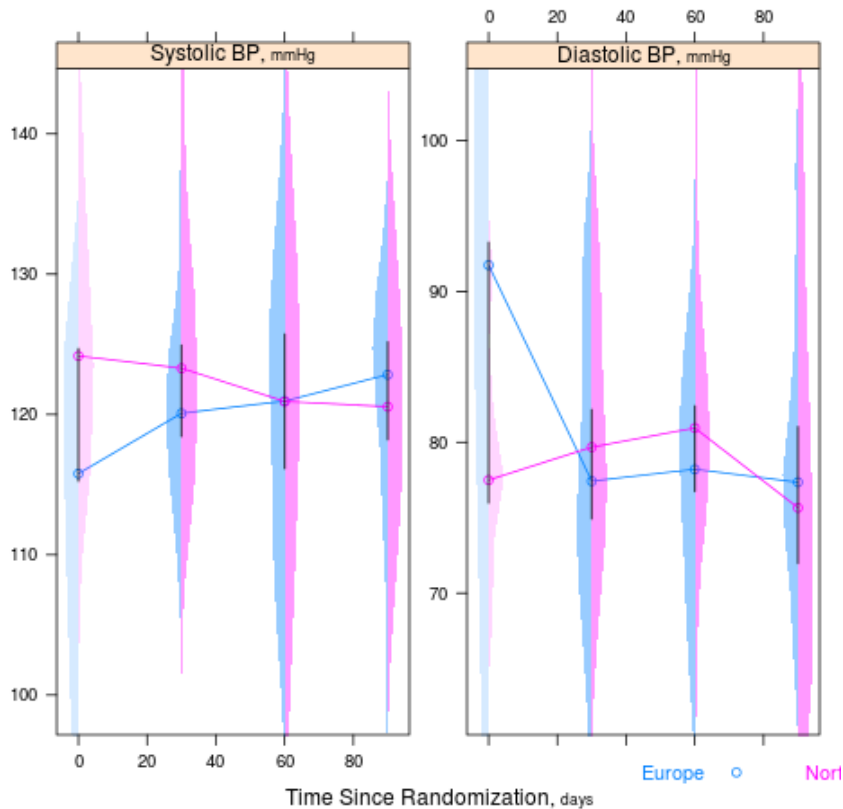
```
## Demonstrate simultaneous use of fun and panel
## First show the same quantile intervals used in panel.bppplot by
## default, stratified by region and day
```

```
d <- upData(d, days=round(days / 30) * 30)
g <- function(y) {
  probs <- c(0.05, 0.125, 0.25, 0.375)
  probs <- sort(c(probs, 1 - probs))
  y <- y[! is.na(y)]
  w <- hdquantile(y, probs)
  m <- hdquantile(y, 0.5, se=TRUE)
  se <- as.numeric(attr(m, 'se'))
  c(Median=as.numeric(m), w, se=se, n=length(y))
}
s <- summaryS(sbp + dbp ~ days + region, fun=g, data=d)
plot(s, groups='region', panel=mbarclPanel, paneldoesgroups=TRUE)
```

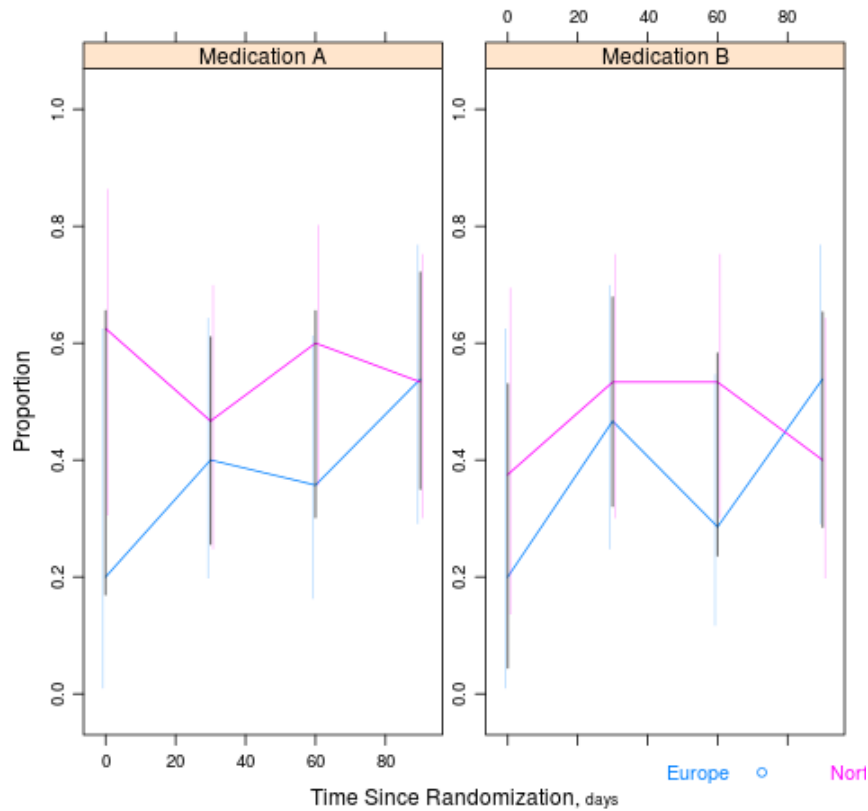


```
## Similar but use half-violin plots  
s <- summaryS(sbp + dbp ~ days + region, data=d)  
plot(s, groups='region', panel=medvPanel, paneldoesgroups=TRUE)
```








```
## Show Wilson confidence intervals for proportions, and confidence
## intervals for difference in two proportions
g <- function(y) {
  y <- y[!is.na(y)]
  n <- length(y)
  p <- mean(y)
  se <- sqrt(p * (1. - p) / n)
  structure(c(binconf(sum(y), n), se=se, n=n),
            names=c('Proportion', 'Lower', 'Upper', 'se', 'n'))
}
s <- summaryS(meda + medb ~ days + region, fun=g, data=d)
plot(s, groups='region', panel=mbarclPanel, paneldoesgroups=TRUE)
```



## tabulr

`tabulr` is a front-end to the `tabular` function in the `tables` package. `tabular` provides an elegant syntax for advanced multi-level LaTeX and regular text tables. `tabulr` makes use of `Hmisc` package variable attributes `label` and `units` for nicely labeling table components. By default, all variables appearing in the table that have labels have those labels used (and units of measurement, if present) in place of variable names. Also provided are some utility functions to mimic `summaryM` output for continuous variables (see `trio`) and functions for creating various LaTeX macro definitions that are useful in table making. See [this knitr output](#) for an example.

I	Attachment	Action	Size	Date	Who	Comment
	<a href="#">bpplotM.png</a>	<a href="#">manage</a>	89.0 K	30 Jul 2013 - 15:55	<a href="#">FrankHarrell</a>	Example of the Hmisc package bpplotM function
	<a href="#">summaryD.png</a>	<a href="#">manage</a>	18.4 K	08 Aug 2013 - 13:00	<a href="#">FrankHarrell</a>	Examples of Hmisc summaryD function
	<a href="#">summaryD2.png</a>	<a href="#">manage</a>	2.8 K	09 Aug 2013 - 15:02	<a href="#">FrankHarrell</a>	summaryD second example
	<a href="#">summaryP.png</a>	<a href="#">manage</a>	32.5 K	18 Dec 2013 - 15:23	<a href="#">FrankHarrell</a>	Example of the Hmisc package summaryP function
	<a href="#">summaryRc.png</a>	<a href="#">manage</a>	48.3 K	10 Aug 2013 - 13:04	<a href="#">FrankHarrell</a>	Example of Hmisc summaryRc function
	<a href="#">summaryS1.png</a>	<a href="#">manage</a>	53.3 K	28 Dec 2013 - 21:19	<a href="#">FrankHarrell</a>	summaryS example 1
	<a href="#">summaryS10.png</a>	<a href="#">manage</a>	19.0 K	04 Jan 2014 - 14:59	<a href="#">FrankHarrell</a>	summaryS example: confidence intervals for proportions and differences
	<a href="#">summaryS2.png</a>	<a href="#">manage</a>	35.2 K	28 Dec 2013 - 21:20	<a href="#">FrankHarrell</a>	summaryS ex. 2
	<a href="#">summaryS3.png</a>	<a href="#">manage</a>	56.5 K	28 Dec 2013 - 21:20	<a href="#">FrankHarrell</a>	summaryS ex. 3
	<a href="#">summaryS4.png</a>	<a href="#">manage</a>	30.0 K	28 Dec 2013 - 21:21	<a href="#">FrankHarrell</a>	
	<a href="#">summaryS5.png</a>	<a href="#">manage</a>	13.0 K	28 Dec 2013 - 21:21	<a href="#">FrankHarrell</a>	
	<a href="#">summaryS6.png</a>	<a href="#">manage</a>	20.6 K	28 Dec 2013 - 21:21	<a href="#">FrankHarrell</a>	
	<a href="#">summaryS7.png</a>	<a href="#">manage</a>	20.7 K	28 Dec 2013 - 21:22	<a href="#">FrankHarrell</a>	
	<a href="#">summaryS8.png</a>	<a href="#">manage</a>	17.7 K	28 Dec 2013 - 21:22	<a href="#">FrankHarrell</a>	

I	Attachment	Action	Size	Date	Who	Comment
	<a href="#">summaryS9.png</a>	<a href="#">manage</a>	19.4 K	04 Jan 2014 - 14:58	<a href="#">FrankHarrell</a>	summaryS example: multiple quantile intervals
	<a href="#">summaryS9v.png</a>	<a href="#">manage</a>	18.6 K	29 Jan 2014 - 16:34	<a href="#">FrankHarrell</a>	Example of summaryS: half-violin plots over time
	<a href="#">tabulr.pdf</a>	<a href="#">manage</a>	131.9 K	17 Aug 2013 - 12:44	<a href="#">FrankHarrell</a>	knitr output demonstrating tabulr function in Hmisc

This topic: Main > WebHome > StatComp > RS > Hmisc > HmiscNew

Topic revision: 29 Jan 2014, FrankHarrell

Copyright © 2013-2017 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding Vanderbilt Biostatistics Wiki? Send feedback

