

# gClinBiomarker: example use cases

*Ning Leng, Alexey Pronin, Doug Kelkhoff*

*2017-09-27*

## Contents

<b>1</b>	<b>Loading packages and data</b>	<b>4</b>
<b>2</b>	<b>Example data set</b>	<b>4</b>
<b>3</b>	<b>SummaryVars(): Summarize demographics variables (check demographics imbalance)</b>	<b>5</b>
3.1	Default . . . . .	5
3.2	Compare to non-BEP instead of ITT . . . . .	6
3.3	Allow testing . . . . .	6
3.4	Reorder TRT/CTRL arm in display . . . . .	6
3.5	Combine trt arms . . . . .	7
<b>4</b>	<b>PlotProperty(): Plot biomarker, clinical covariate property and their association</b>	<b>7</b>
4.1	Numeric biomarker variable. No clinical variables. Log transformation. . . . .	7
4.2	No biomarker variable. Two numeric clinical variables. Log transformation for one variable. . .	8
4.3	Numeric biomarker variable. Two clinical categorical variables. Log transformation for numeric variable. . . . .	8
4.4	Numeric biomarker variable. Two clinical variables: one is categorocal, second is numeric. Log transformation for biomarker (numeric) variable. . . . .	9
4.5	Categorical biomarker variable. Categorical clinical variable. . . . .	10
4.6	Categorical biomarker variable. Categorical and numeric clinical variables. No log transformation. .	11
4.7	Categorical biomarker variable. Categorical and numeric clinical variables. No log transformation. Don't plot biomarker variable but plot clinical variables and association between biomarker and clinical variables. . . . .	12
4.8	Categorical biomarker variable. Categorical and numeric clinical variables. No log transformation. Don't plot clinical variables but plot biomarker variable and association between biomarker and clinical variables. . . . .	13
4.9	Categorical biomarker variable. Categorical and numeric clinical variables. No log transformation. Don't plot association between variables but plot biomarker and clinical variables. . . .	14
4.10	Categorical biomarker variable. Categorical and numeric clinical variables. No log transformation. Plot only association between variables. . . . .	15
4.11	Categorical biomarker variable. Categorical and numeric clinical variables. No log transformation. Plot only clinical variables. . . . .	16
<b>5</b>	<b>PlotRspBar(): Compare response outcome across different population</b>	<b>17</b>
5.1	Plot response of one population . . . . .	17
5.2	Plot response of one population, binarize outcome (classify classes to responder vs non responder)	18
5.3	Change color . . . . .	18
5.4	Plot horizontally . . . . .	19
5.5	By arm . . . . .	20
5.6	Compare full population vs BEP . . . . .	22
5.7	Compare subgroups . . . . .	23
5.8	Plot count instead of percentage . . . . .	24
<b>6</b>	<b>PlotKM(): Plot KM curves for subpopulations</b>	<b>25</b>
6.1	Without subgroup . . . . .	25

6.2	Without subgroup, BEP	26
6.3	By TRT	27
6.4	By TRT, change color, line type	28
6.5	Mark median PFS, no grid	29
6.6	By KRAS.mutant only	30
6.7	By TRT and KRAS.mutant	32
6.8	By TRT and KRAS.mutant, change color and line type	33
6.9	By TRT and KRAS.expression (continuous biomarker, cut at median)	34
6.10	By TRT and KRAS.expression (continuous biomarker, cut at a numerical cutoff)	35
6.11	By TRT and KRAS.expression (continuous biomarker, cut at a numerical cutoff). High group defined as “greater than”, instead of “greater than or equal to”	36
6.12	By TRT and KRAS.expression , more than 2 groups	37
6.13	More flexibility, reorder subgroups	39
6.14	By TRT and KRAS.mutant, reorder and rename	40
6.15	Legend location	41
<b>7</b>	<b>PlotTabForestBiomarker() : forest plot and summary statistics table for a single biomarker (include cutoff exploration for cont. biomarker )</b>	<b>42</b>
7.1	Survival outcome, 2-arm, categorical variable	42
7.2	Survival outcome, another format	44
7.3	Survival outcome, 2-arm, categorical variable, don't show ITT, BEP, rename variable name in display	45
7.4	Survival outcome, 2-arm, continuous variable, greater than some percentage cutoffs	47
7.5	Survival outcome, 2-arm, continuous variable, less than some percentage cutoffs	48
7.6	Survival outcome, 2-arm, continuous variable, greater and less than some percentage cutoffs	50
7.7	Survival outcome, 2-arm, continuous variable, within cutoff bin	53
7.8	Survival outcome, 2-arm, continuous variable, greater than some numerical cutoffs	54
7.9	Survival outcome, 2-arm, continuous variable, adjust for covariates	55
7.10	Survival outcome, 2-arm, continuous variable, stratification	57
7.11	Survival outcome, 2-arm, continuous variable, adjust for covariates and stratified by strata	58
7.12	Survival outcome, 1arm	59
7.13	Survival outcome, 2arm, flip TRT/CTRL order (calculate ctrl over trt HR)	60
7.14	Survival outcome, 2-arm, continuous variable, greater than some percentage cutoffs, show across-arm and within-arm results in the same plot	62
7.15	Response outcome, 2-arm, categorical variable	64
7.16	Response outcome, another format	66
7.17	Continuous outcome, 2-arm, categorical variable	67
<b>8</b>	<b>PlotTabForestMulti(): Forest plots to compare multiple variates' effect in ITT vs BEP, or compare multiple variates' effect in subpopulations</b>	<b>69</b>
8.1	2-arm, compare BEP vs ITT	69
8.2	2-arm, compare subgroups defined by biomarker	71
8.3	2-arm, compare subgroups defined by biomarker, also show ITT	71
8.4	2-arm, compare subgroups defined by biomarker, also show BEP	72
8.5	1-arm, compare ITT vs BEP	76
8.6	2-arm, compare subgroups defined by biomarker, multiple cutoffs for continuous biomarker	76
8.7	2-arm, compare subgroups defined by biomarker, multiple cutoffs for continuous biomarker, calculate “< cutoff”	78
8.8	2-arm, compare subgroups defined by biomarker, multiple cutoffs for continuous biomarker, calculate within bin statistics	80
8.9	Basic forest plot without comparison	81
8.10	Response outcome	81
8.11	Continuous endpoint	83

<b>9</b>	<b>PlotSTEPP(): STEPP (Subpopulation Treatment Effect Pattern Plot) from the given point estimates and confidence intervals at desired percentiles.</b>	<b>85</b>
9.1	Survival outcome . . . . .	85
9.2	Continuous outcome . . . . .	87
9.3	Binary outcome variable . . . . .	88
<b>10</b>	<b>CoxTab(): fit cox proportional model for multiple covariates (additive model or separate models)</b>	<b>90</b>
10.1	single covariate . . . . .	90
10.2	multiple covariates, ITT . . . . .	90
10.3	BEP . . . . .	90
10.4	Reorder Sex: M as reference . . . . .	91
10.5	Fit separate models for each variable instead of fitting an additive model . . . . .	91
<b>11</b>	<b>StatSummary()</b>	<b>92</b>
11.1	Contunuous outcome . . . . .	92
11.2	Binary outcome . . . . .	93
11.3	Survival variable . . . . .	93
<b>12</b>	<b>LogRankTab(): log rank test for subgroup analysis</b>	<b>94</b>
<b>13</b>	<b>BoxPlot(): advanced box plot function</b>	<b>94</b>
13.1	Generate example dataset . . . . .	94
13.2	Specify data as named list . . . . .	94
13.3	Same plot, now horizontally plotted . . . . .	95
13.4	Specify data as numeric matrix . . . . .	95
13.5	Specify data as numeric (unnamed) vectors . . . . .	96
13.6	... horizontally plotted (no axis-label rotation done here) . . . . .	96
13.7	Plot values 'y' according to time 'time' (factor levels are automatically ordered as e.g. function sort does) . . . . .	97
13.8	Now as box-percentile plot . . . . .	97
13.9	With custom main title . . . . .	98
13.10	The same plot with trend line (connects either means or medians) and Y-axis label . . . . .	98
13.11	Use an addition grouping variable to color points in the stripchart . . . . .	99
13.12	Use yet another grouping factor for plotting symbols in the stripchart . . . . .	99
13.13	Generate new dataset with different stucture . . . . .	100
13.14	Boxplot with trend lines (mean) and grid added to the plot . . . . .	100
13.15	Now as box-percentile plot . . . . .	101
13.16	Use yet another way to specify the data . . . . .	101
13.17	Specifying data as numeric matrix and using a trend-line connecting the means . . . . .	102
13.18	Multiple grouping factors can be specified via the formula interface which is exemplified using the mtcars dataset, of interest is miles per gallon (mpg) depending on number of gears and on the number of cylinders . . . . .	102
13.19	Now 'cyl' is explicitly nested within 'gear' which changes the ordering of combined grouping factors (which is identical to using formula 'mpg-cyl:gear'). . . . .	103
13.20	More meaningful group-labels are best specified using custom factor-level names (increase width of the plot window) . . . . .	103
13.21	One can use a table as Xaxis label representing the factor-level combination defining sub-classes	104
13.22	With the original factor levels and as horizontal plot . . . . .	104
13.23	Using smaller bottom margin will result in smaller table height . . . . .	105
13.24	One can use different font-settings for rownames and cells of the table . . . . .	105
13.25	Use more crossed factors . . . . .	106
13.26	Alternatively one can use the 'Xaxis' argument, but the ordering of these labels is not checked which is not important for automatically generated group-labels as shown in the previous example . . . . .	106

13.27	The same plot with some fancy options . . . . .	107
13.28	Horizontal fancy plot . . . . .	108
<b>14</b>	<b>Longitudinal analysis</b>	<b>108</b>
14.1	Biomarker Endpoint Progression . . . . .	108
14.2	Plotting Subpopulations . . . . .	110
14.3	Including sample counts . . . . .	111
14.4	Plotting Deviations as Errorbars . . . . .	112

## 1 Loading packages and data

```
library(knitr)
library(devtools)

## Warning: package 'devtools' was built under R version 3.3.2

install_github("RPackages/gClinBiomarker", host="https://github.roche.com/api/v3")
library(gClinBiomarker)

## Warning: package 'survival' was built under R version 3.3.2

library(ggplot2)
data(input)
sample.data <- input
```

## 2 Example data set

```
head(sample.data)
```

##	Patient.ID	Sample.ID	Arm	Age	Weight	ECOG	Sex	Baseline.SLD	Country
## 1	PID001	SID001	TRT	40	61.0	0	F	550.4970	Western Europe
## 2	PID002	SID002	TRT	60	87.0	1	F	549.7751	USA
## 3	PID003	SID003	TRT	53	57.6	1	M	548.8829	Western Europe
## 4	PID004	SID004	TRT	44	57.0	0	F	549.6050	USA
## 5	PID005	SID005	TRT	49	72.5	2	F	551.5498	Other
## 6	PID006	SID006	TRT	62	76.0	0	M	549.2565	Other

##	CD8.ihc	KRAS.mutant	KRAS.exprs	OS	PFS	Response	BEP	OS.event
## 1	3	Mutant	171.25	9.691992	5.782341	PR	1	0
## 2	2	Wild Type	2.77	4.435318	1.412731	PD	1	1
## 3	3	Wild Type	61.82	9.691992	1.215606	PD	1	0
## 4	1	Mutant	71.51	1.905544	1.905544	PD	1	0
## 5	NA	<NA>	NA	2.628337	2.628337	SD	0	1
## 6	2	Mutant	44.63	4.829569	4.139630	NON CR/PD	1	0

##	PFS.event	Lab_ontrt
## 1	0	28.993922
## 2	1	9.890427
## 3	1	26.896020
## 4	1	5.421392
## 5	1	14.741382
## 6	0	16.957035

```
str(sample.data)
```

```
## 'data.frame':   550 obs. of  19 variables:
## $ Patient.ID   : chr  "PID001" "PID002" "PID003" "PID004" ...
## $ Sample.ID    : chr  "SID001" "SID002" "SID003" "SID004" ...
## $ Arm          : Factor w/ 2 levels "CTRL","TRT": 2 2 2 2 2 2 2 2 1 ...
## $ Age          : int   40 60 53 44 49 62 70 49 62 45 ...
## $ Weight       : num   61 87 57.6 57 72.5 76 46 69.3 65 54 ...
## $ ECOG         : int    0 1 1 0 2 0 0 1 0 1 ...
## $ Sex          : chr   "F" "F" "M" "F" ...
## $ Baseline.SLD : num   550 550 549 550 552 ...
## $ Country      : Factor w/ 3 levels "Other","USA",...: 3 2 3 2 1 1 3 3 1 ...
## $ CD8.ihc      : num    3 2 3 1 NA 2 NA 1 NA NA ...
## $ KRAS.mutant  : Factor w/ 2 levels "Mutant","Wild Type": 1 2 2 1 NA 1 NA 2 NA NA ...
## $ KRAS.exprs   : num   171.25 2.77 61.82 71.51 NA ...
## $ OS           : num    9.69 4.44 9.69 1.91 2.63 ...
## $ PFS          : num    5.78 1.41 1.22 1.91 2.63 ...
## $ Response     : Factor w/ 6 levels "CR","NE","NON CR/PD",...: 5 4 4 4 6 3 4 3 4 6 ...
## $ BEP          : num    1 1 1 1 0 1 0 1 0 0 ...
## $ OS.event     : int    0 1 0 0 1 0 0 0 1 0 ...
## $ PFS.event    : int    0 1 1 1 1 0 1 0 1 0 ...
## $ Lab_ontrt    : num   28.99 9.89 26.9 5.42 14.74 ...
```

### 3 SummaryVars(): Summarize demographics variables (check demographics imbalance)

#### 3.1 Default

```
kable(
  SummaryVars(data=sample.data, trt='Arm', subgroup='BEP', var=c('Age', 'Sex'),
    var.class=c("numeric", "categorical"))
)
```

	All(CTRL)	BEP(CTRL)	All(TRT)	BEP(TRT)
Age				
N	182	81	368	176
Mean	52.54	53.1	54.03	54.42
Median	51.5	52	54	54
Min-Max	27...85	32...85	30...89	33...89
NA's	0	0	0	0
Sex				
Total	182	81	368	176
NA's	0	0	0	0
F	89 (48.9%)	39 (48.15%)	184 (50%)	90 (51.14%)
M	93 (51.1%)	42 (51.85%)	184 (50%)	86 (48.86%)

### 3.2 Compare to non-BEP instead of ITT

```
kable(
  SummaryVars(data=sample.data, trt='Arm', subgroup='BEP', var=c('Age', 'Sex'),
    var.class=c("numeric", "categorical"), compare.subgroup=TRUE)
)
```

	BEP_1(CTRL)	BEP_0(CTRL)	BEP_1(TRT)	BEP_0(TRT)
Age				
N	81	101	176	192
Mean	53.1	52.09	54.42	53.67
Median	52	50	54	54
Min-Max	32...85	27...82	33...89	30...79
NA's	0	0	0	0
Sex				
Total	81	101	176	192
NA's	0	0	0	0
F	39 (48.15%)	50 (49.5%)	90 (51.14%)	94 (48.96%)
M	42 (51.85%)	51 (50.5%)	86 (48.86%)	98 (51.04%)

### 3.3 Allow testing

```
kable(
  SummaryVars(data=sample.data, trt='Arm', subgroup='BEP', var=c('Age', 'Sex'),
    var.class=c("numeric", "categorical"), test.subgroup=TRUE)
)
```

	All(CTRL)	BEP(CTRL)	pvalue(CTRL)	All(TRT)	BEP(TRT)	pvalue(TRT)
Age						
N	182	81	0.5	368	176	0.69
Mean	52.54	53.1		54.03	54.42	
Median	51.5	52		54	54	
Min-Max	27...85	32...85		30...89	33...89	
NA's	0	0		0	0	
Sex						
Total	182	81	0.88	368	176	0.75
NA's	0	0		0	0	
F	89 (48.9%)	39 (48.15%)		184 (50%)	90 (51.14%)	
M	93 (51.1%)	42 (51.85%)		184 (50%)	86 (48.86%)	

### 3.4 Reorder TRT/CTRL arm in display

```
kable(
  SummaryVars(data=sample.data, trt='Arm', subgroup='BEP', var=c('Age', 'Sex'),
    var.class=c("numeric", "categorical"), trt.order=c("TRT", "CTRL"))
)
```

	All(TRT)	BEP(TRT)	All(CTRL)	BEP(CTRL)
Age				
N	368	176	182	81
Mean	54.03	54.42	52.54	53.1
Median	54	54	51.5	52
Min-Max	30...89	33...89	27...85	32...85
NA's	0	0	0	0
Sex				
Total	368	176	182	81
NA's	0	0	0	0
F	184 (50%)	90 (51.14%)	89 (48.9%)	39 (48.15%)
M	184 (50%)	86 (48.86%)	93 (51.1%)	42 (51.85%)

### 3.5 Combine trt arms

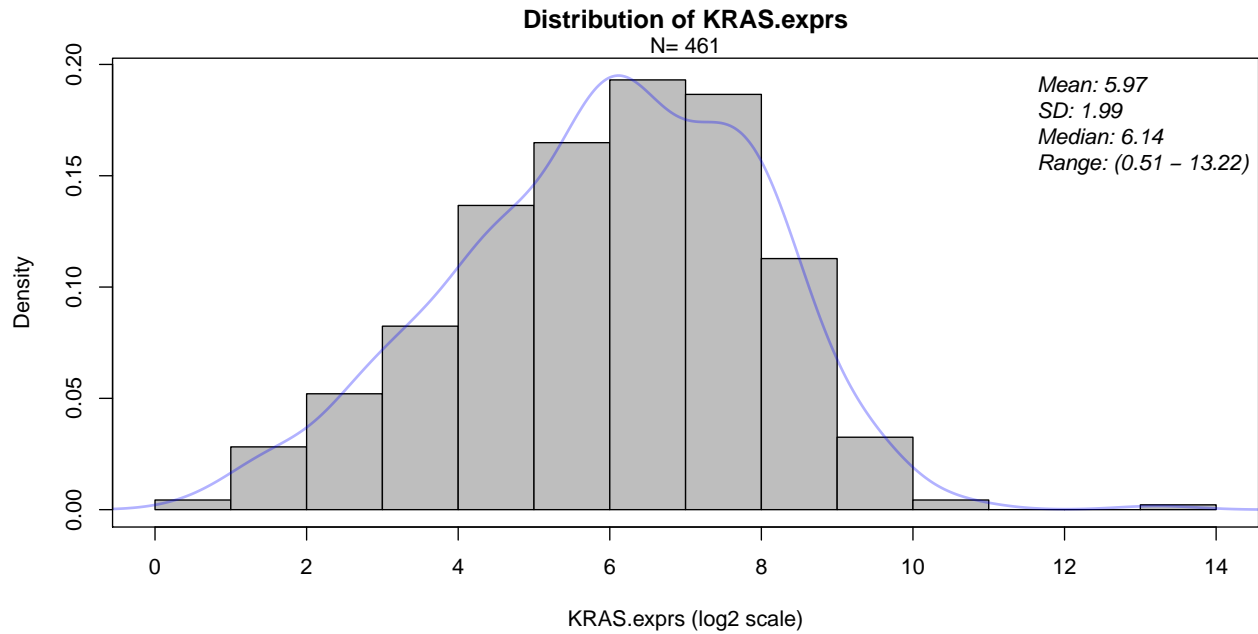
```
kable(
  SummaryVars(data=sample.data, subgroup='BEP', var=c('Age', 'Sex'),
    var.class=c("numeric", "categorical"))
)
```

	All	BEP
Age		
N	550	257
Mean	53.53	54
Median	53	54
Min-Max	27...89	32...89
NA's	0	0
Sex		
Total	550	257
NA's	0	0
F	273 (49.64%)	129 (50.19%)
M	277 (50.36%)	128 (49.81%)

## 4 PlotProperty(): Plot biomarker, clinical covariate property and their association

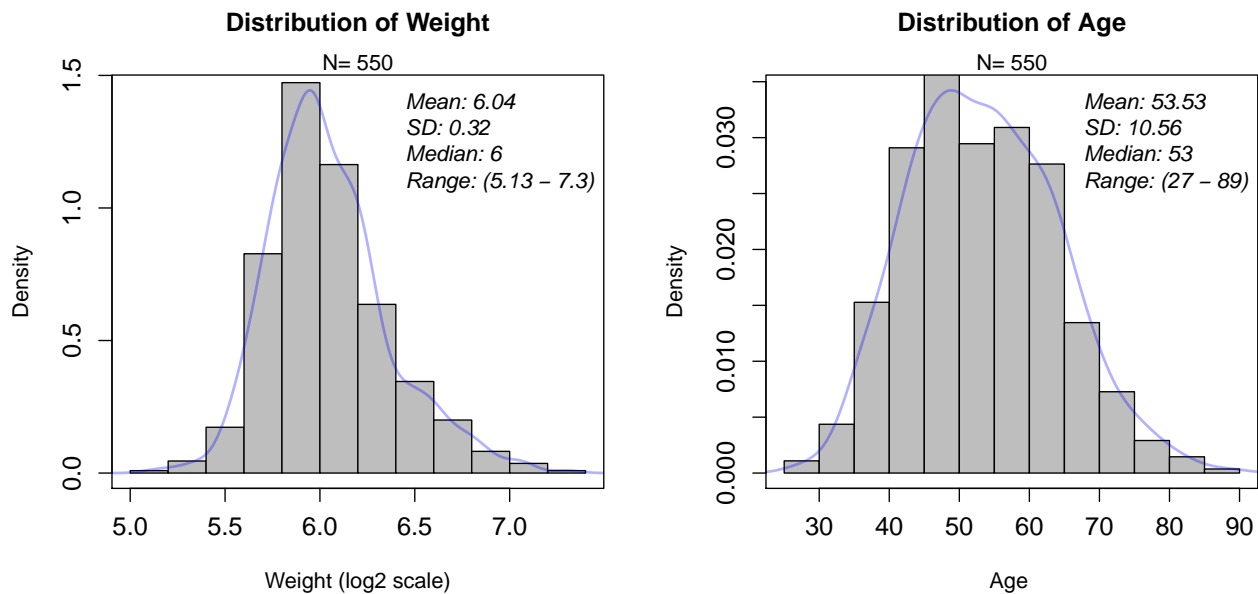
### 4.1 Numeric biomarker variable. No clinical variables. Log transformation.

```
PlotProperty(data=input, biomarker.var="KRAS.exprs", biomarker.class="numeric", log2=TRUE)
```



#### 4.2 No biomarker variable. Two numeric clinical variables. Log transformation for one variable.

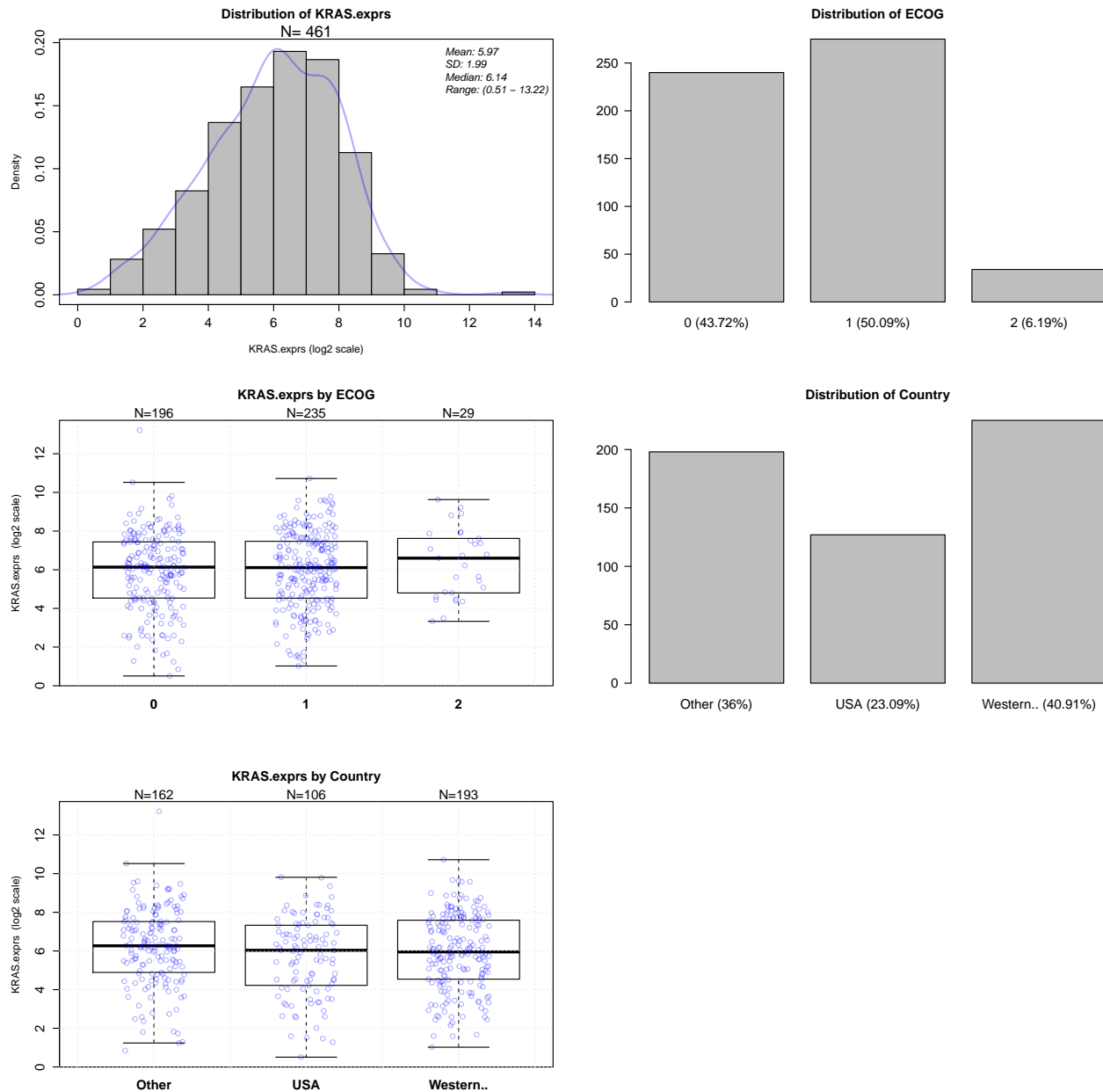
```
PlotProperty(data=input, biomarker.var=NULL, biomarker.class=NULL,
var=c("Weight", "Age"), var.class=c("numeric", "numeric"),
log2=c(TRUE, FALSE), par.param = list(mfrow=c(1,2), cex.axis=1.2))
```



#### 4.3 Numeric biomarker variable. Two clinical categorical variables. Log transformation for numeric variable.

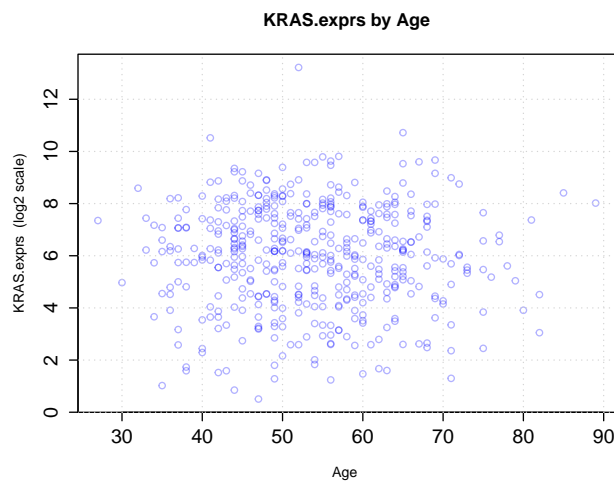
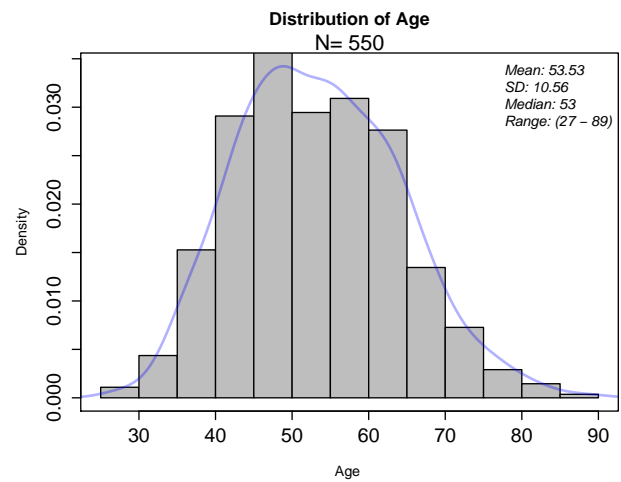
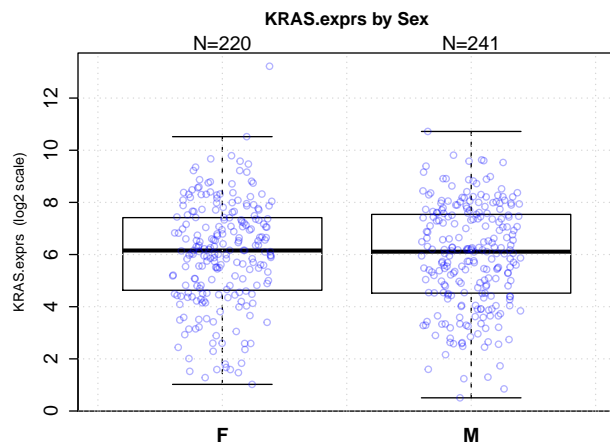
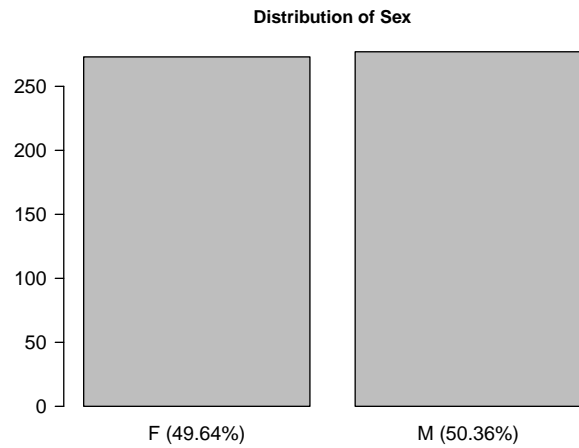
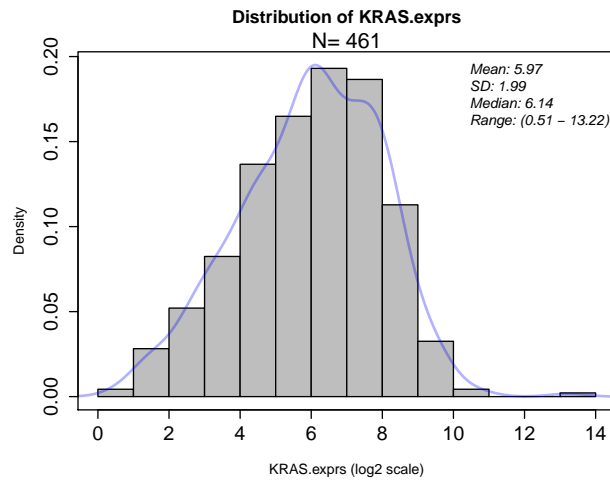


```
PlotProperty(data=input, biomarker.var="KRAS.exprs", biomarker.class="numeric",
var=c("ECOG", "Country"), var.class=c("categorical", "categorical"),
log2=TRUE, par.param = list(mfrow=c(3,2), cex.axis=1.2),
show.clinical.uni=TRUE)
```



#### 4.4 Numeric biomarker variable. Two clinical variables: one is categorical, second is numeric. Log transformation for biomarker (numeric) variable.

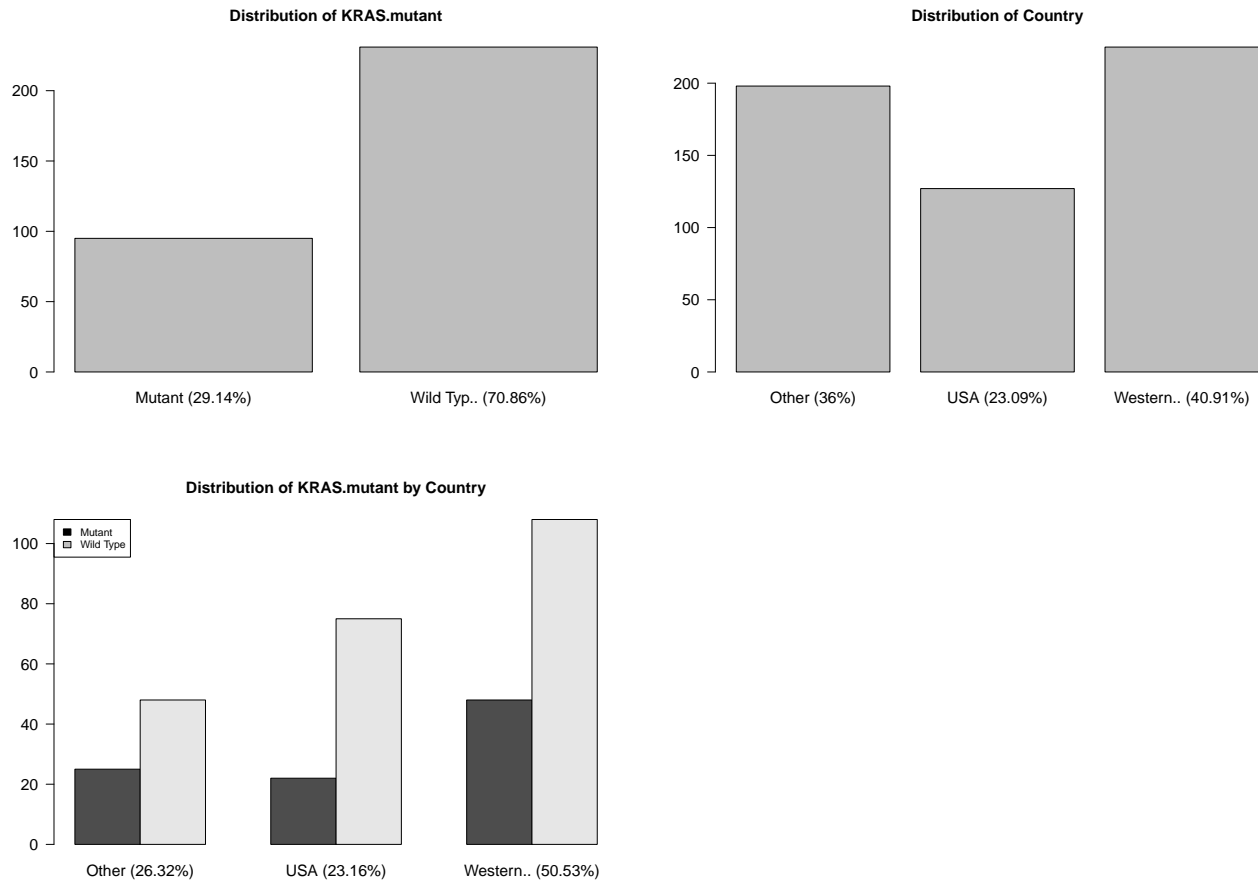
```
PlotProperty(data=input, biomarker.var="KRAS.exprs", biomarker.class="numeric",
var=c("Sex", "Age"), var.class=c("categorical", "numeric"),
log2=c(TRUE, FALSE, FALSE), par.param = list(mfrow=c(3,2), cex.axis=1.4),
show.clinical.uni=TRUE)
```



#### 4.5 Categorical biomarker variable. Categorical clinical variable.

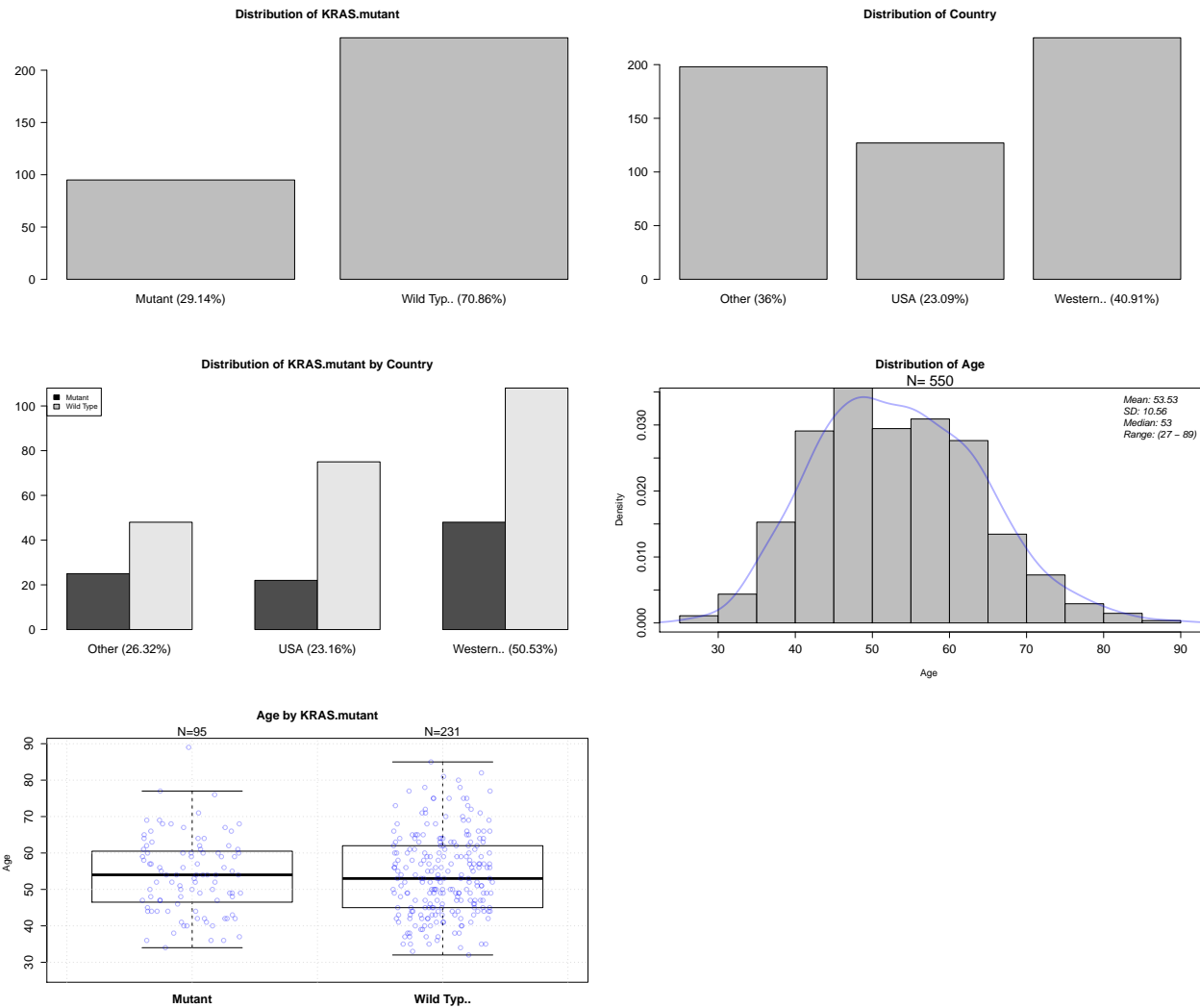
```
PlotProperty(data=input, biomarker.var="KRAS.mutant", biomarker.class="categorical",
             var="Country", var.class="categorical",
```

```
par.param = list(mfrow=c(2,2), cex.axis=1.2),
show.clinical.uni=TRUE)
```



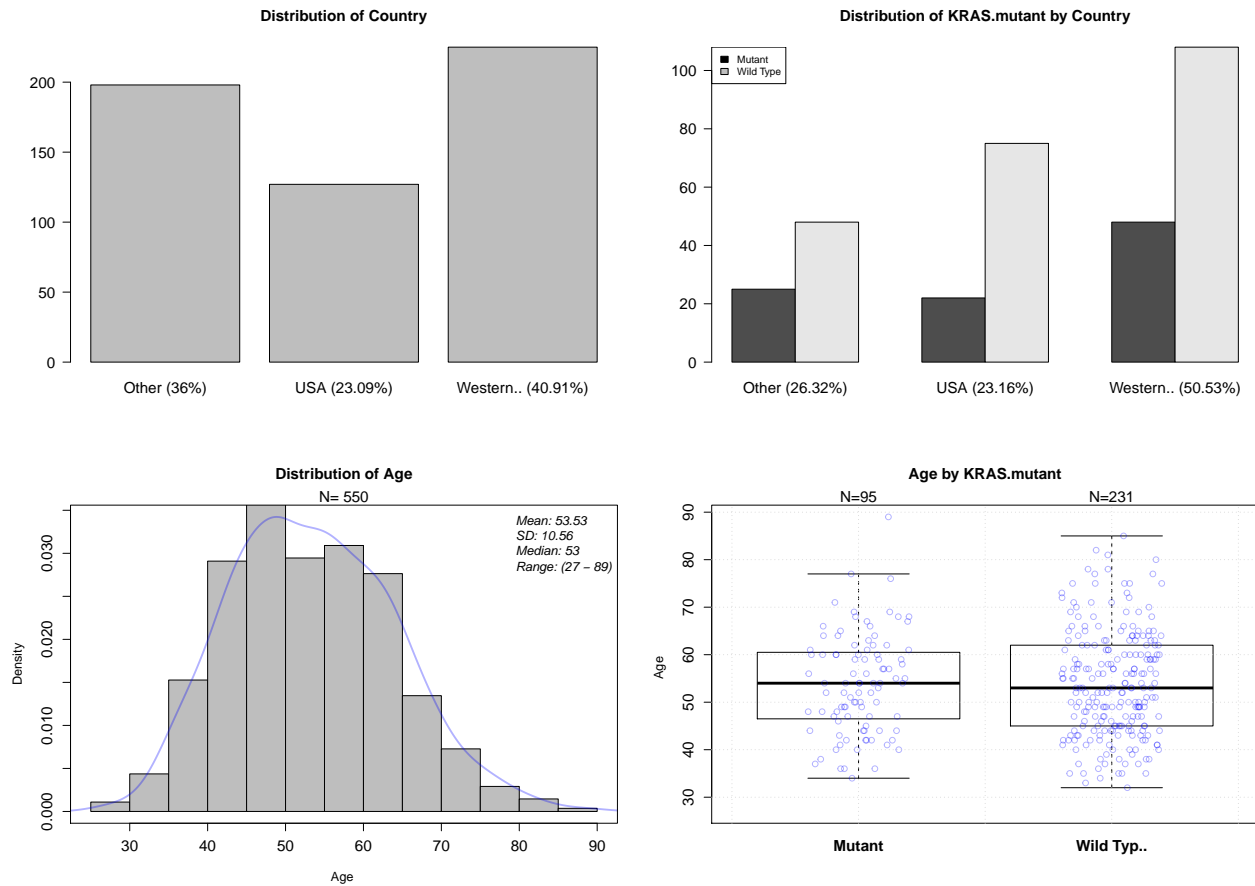
#### 4.6 Categorical biomarker variable. Categorical and numeric clinical variables. No log transformation.

```
PlotProperty(data=input, biomarker.var="KRAS.mutant", biomarker.class="categorical",
var=c("Country", "Age"), var.class=c("categorical", "numeric"),
par.param = list(mfrow=c(3,2), cex.axis=1.2),
show.clinical.uni=TRUE)
```



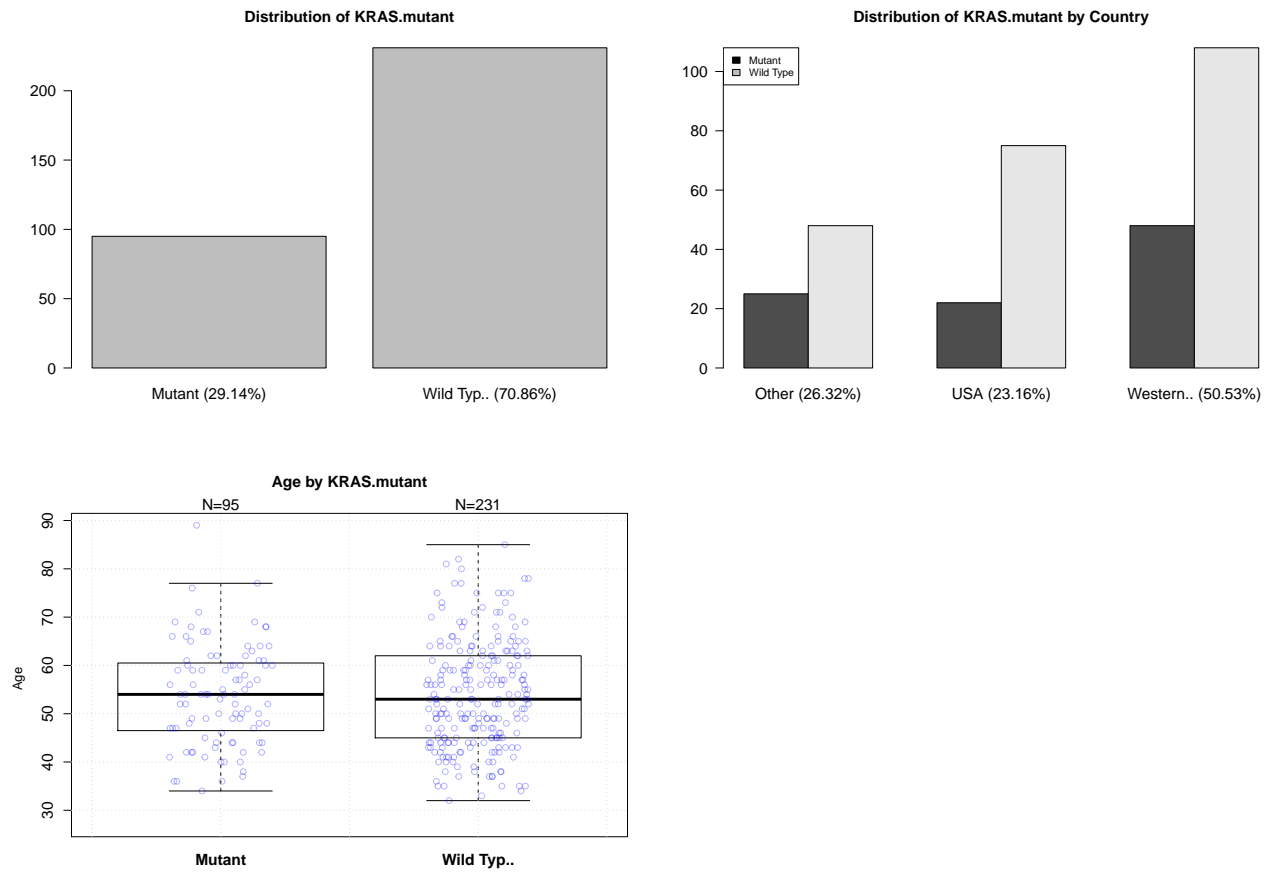
**4.7 Categorical biomarker variable. Categorical and numeric clinical variables. No log transformation. Don't plot biomarker variable but plot clinical variables and association between biomarker and clinical variables.**

```
PlotProperty(data=input, biomarker.var="KRAS.mutant", biomarker.class="categorical",
var=c("Country", "Age"), var.class=c("categorical", "numeric"),
par.param = list(mfrow=c(2,2), cex.axis=1.2), show.biomarker.uni = FALSE,
show.clinical.uni=TRUE)
```



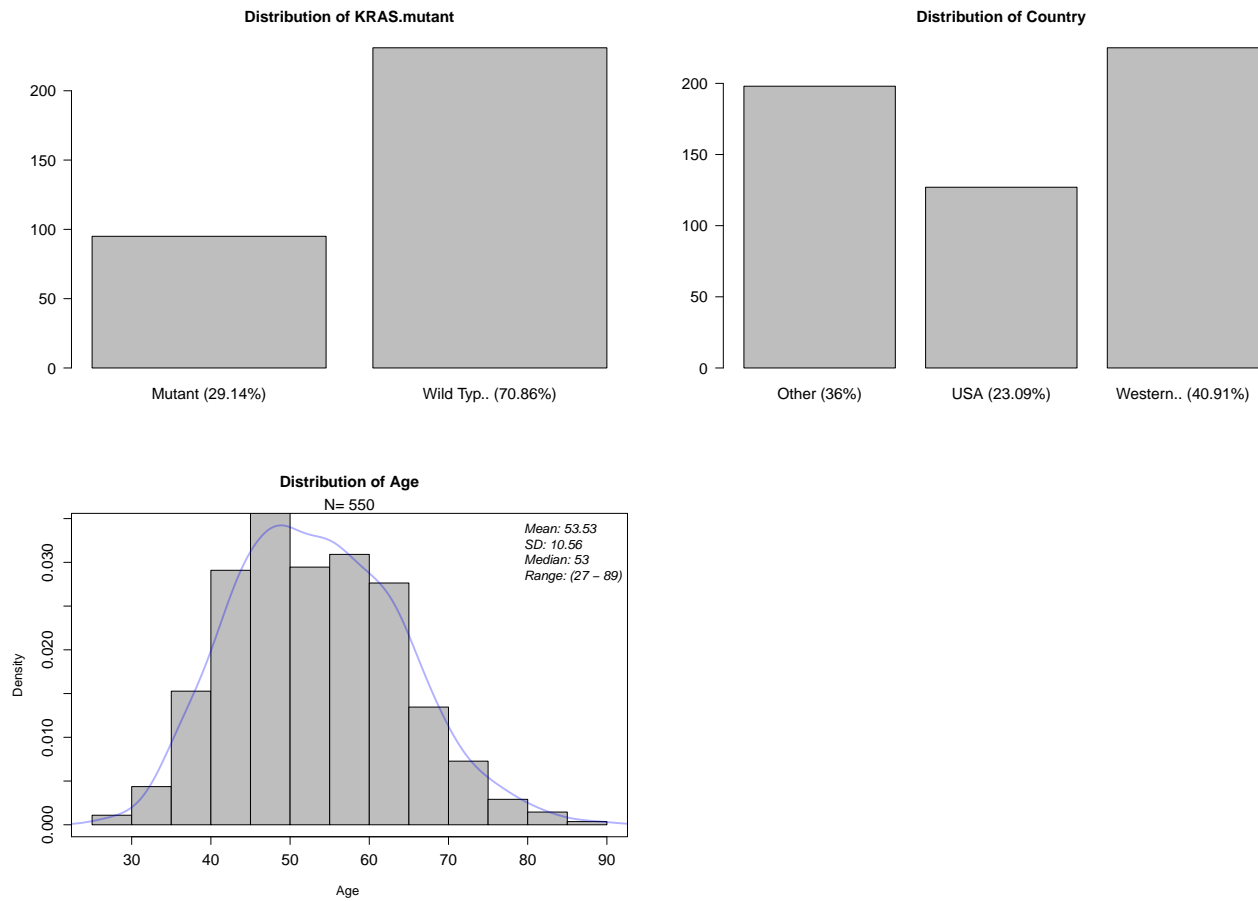
4.8 Categorical biomarker variable. Categorical and numeric clinical variables. No log transformation. Don't plot clinical variables but plot biomarker variable and association between biomarker and clinical variables.

```
PlotProperty(data=input, biomarker.var="KRAS.mutant", biomarker.class="categorical",
var=c("Country", "Age"), var.class=c("categorical", "numeric"),
par.param = list(mfrow=c(2,2), cex.axis=1.2))
```



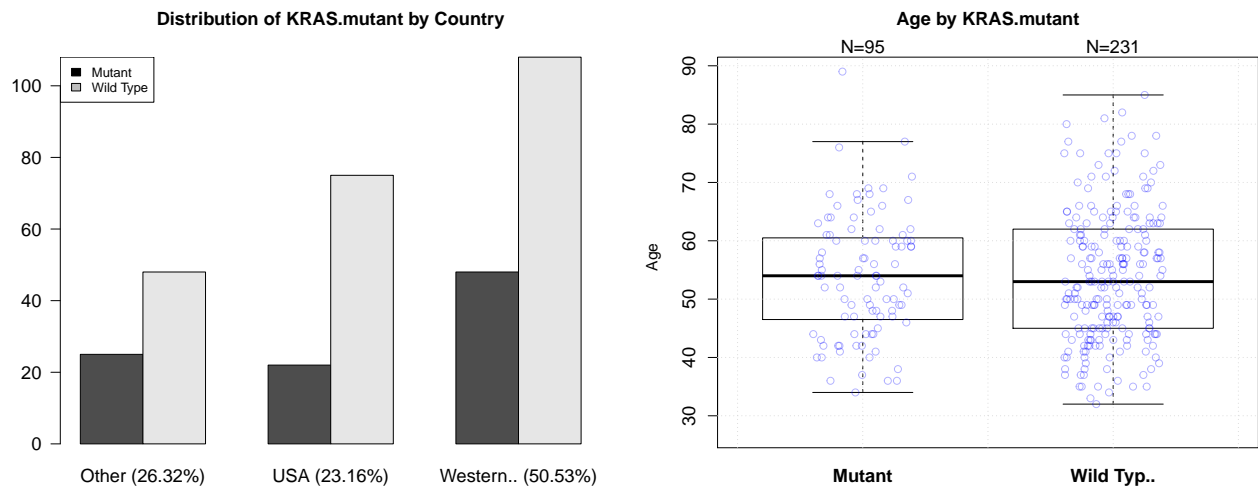
4.9 Categorical biomarker variable. Categorical and numeric clinical variables. No log transformation. Don't plot association between variables but plot biomarker and clinical variables.

```
PlotProperty(data=input, biomarker.var="KRAS.mutant", biomarker.class="categorical",
var=c("Country", "Age"), var.class=c("categorical", "numeric"),
par.param = list(mfrow=c(2,2), cex.axis=1.2), show.association = FALSE,
show.clinical.uni=TRUE)
```



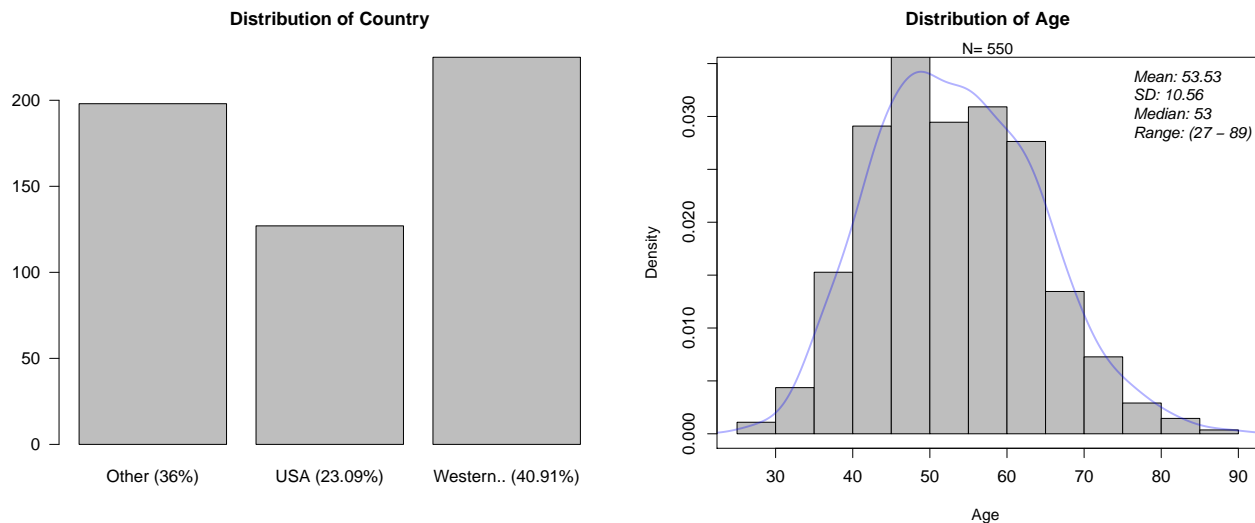
#### 4.10 Categorical biomarker variable. Categorical and numeric clinical variables. No log transformation. Plot only association between variables.

```
PlotProperty(data=input, biomarker.var="KRAS.mutant", biomarker.class="categorical",
var=c("Country", "Age"), var.class=c("categorical", "numeric"),
par.param = list(mfrow=c(1,2), cex.axis=1.2),
show.biomarker.uni = FALSE)
```



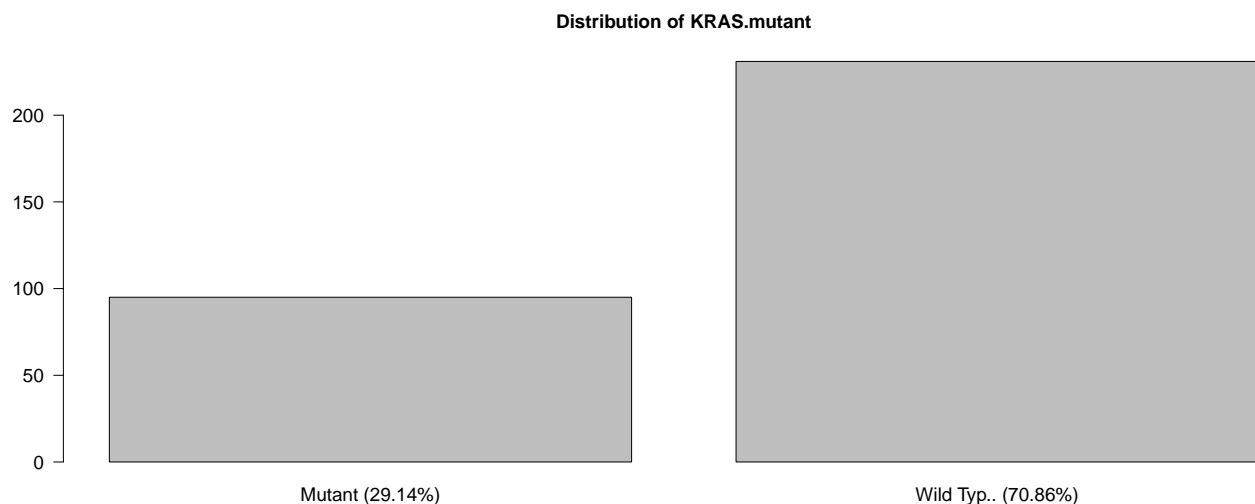
#### 4.11 Categorical biomarker variable. Categorical and numeric clinical variables. No log transformation. Plot only clinical variables.

```
PlotProperty(data=input, biomarker.var="KRAS.mutant", biomarker.class="categorical",
  var=c("Country", "Age"), var.class=c("categorical", "numeric"),
  par.param = list(mfrow=c(1,2), cex.axis=1.1),
  show.biomarker.uni = FALSE, show.association = FALSE,
  show.clinical.uni = TRUE)
```



## Categorical biomarker variable. Categorical and numeric clinical variables. No log transformation. Plot only biomarker variable.

```
PlotProperty(data=input, biomarker.var="KRAS.mutant", biomarker.class="categorical",
  var=c("Country", "Age"), var.class=c("categorical", "numeric"),
  par.param = list(cex.axis=1.2),
  show.association = FALSE)
```





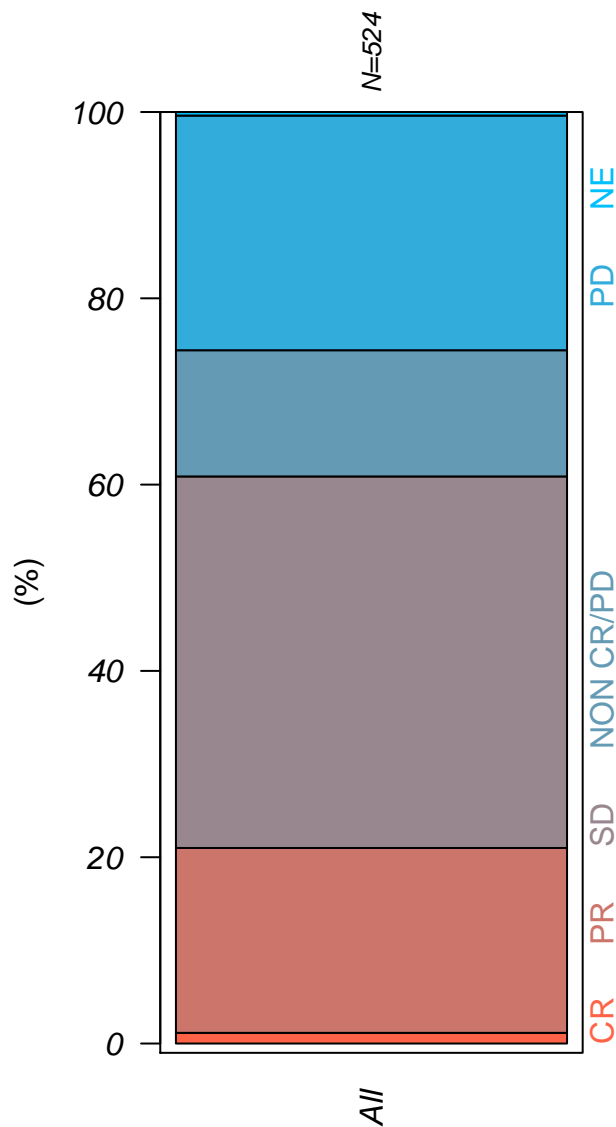
## 5 PlotRspBar(): Compare response outcome across different population

### 5.1 Plot response of one population

```
PlotRspBar (input, outcome.var="Response",  
            rsp.levels=c("CR", "PR", "SD", "NON CR/PD", "PD", "NE"))
```

## entries with missing outcome.var are removed! 524 entries left

#### Association of response rate

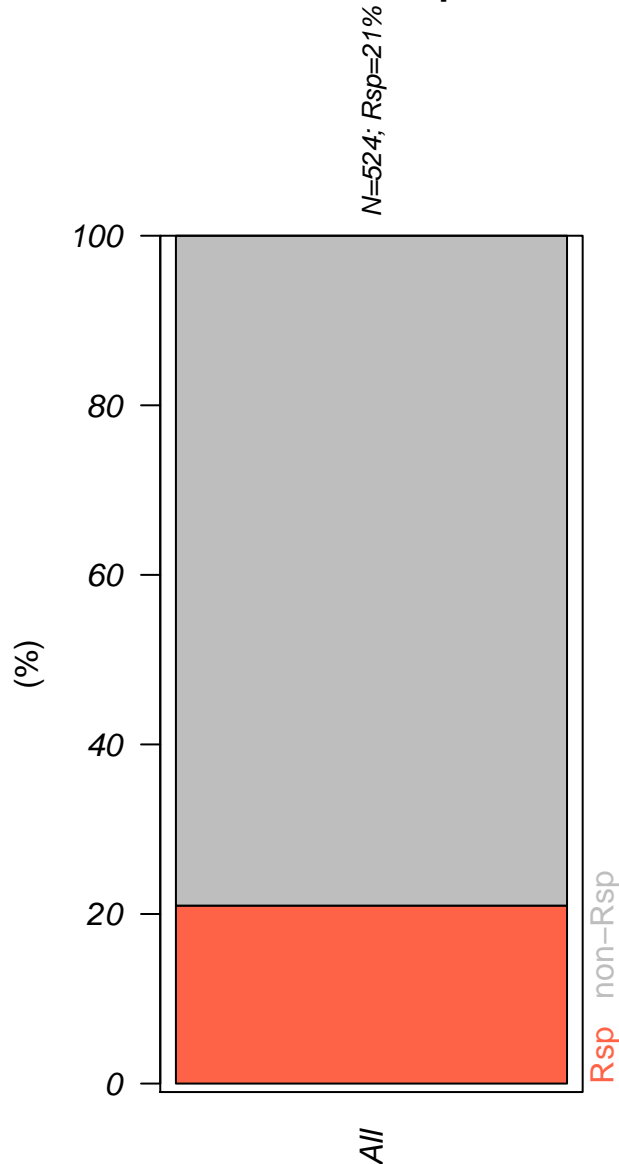


## 5.2 Plot response of one population, binarize outcome (classify classes to responder vs non responder)

```
PlotRspBar (input, outcome.var="Response",  
            binary=TRUE,  
            rsp.response = c("CR", "PR"),  
            rsp.nonresponse = c("SD", "PD", "NON CR/PD", "NE", NA))
```

## entries with missing outcome.var are removed! 524 entries left

### Association of response rate



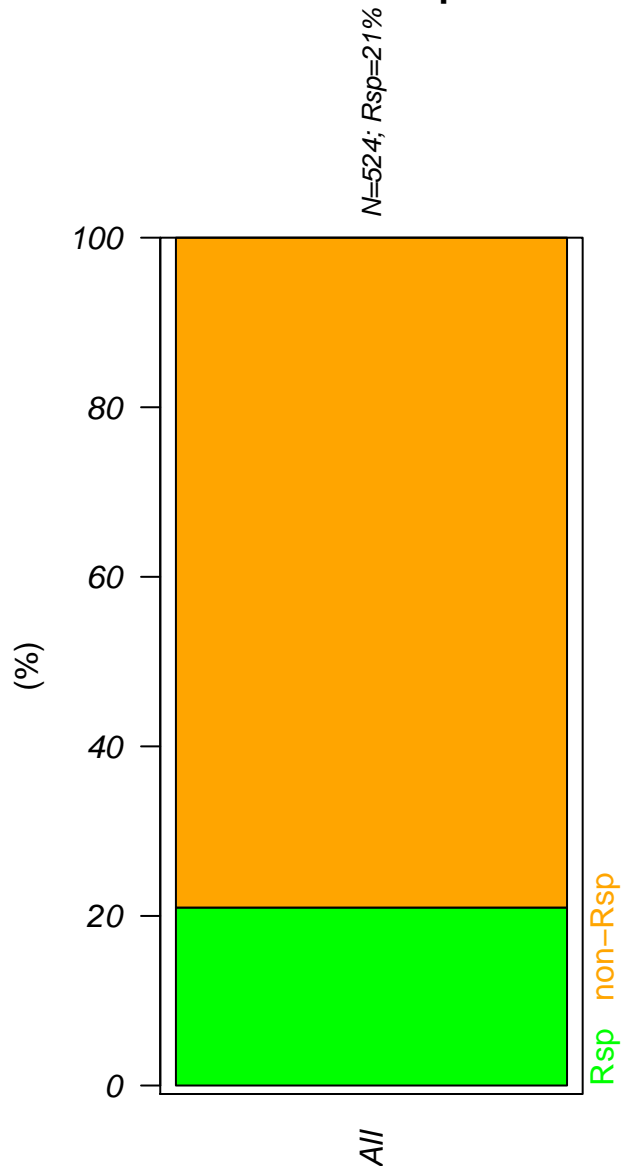
## 5.3 Change color

```
PlotRspBar (input, outcome.var="Response",  
            binary=TRUE,
```

```
rsp.response = c("CR","PR"),
rsp.nonresponse = c("SD", "PD","NON CR/PD","NE",NA),
col=c("green","orange"))
```

## entries with missing outcome.var are removed! 524 entries left

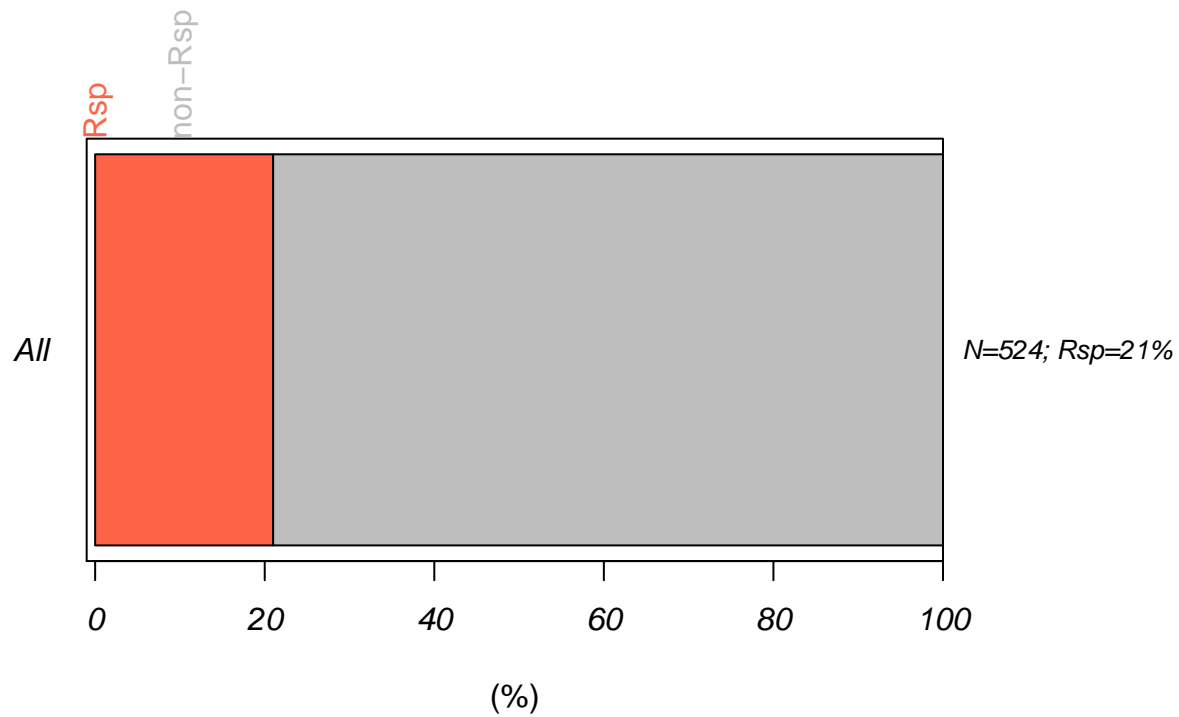
## Association of response rate



### 5.4 Plot horizontally

```
PlotRspBar (input, outcome.var="Response",
  binary=TRUE,
  rsp.response = c("CR","PR"),
  rsp.nonresponse = c("SD", "PD","NON CR/PD","NE",NA),
  horiz=TRUE)
```

```
## entries with missing outcome.var are removed! 524 entries left
```

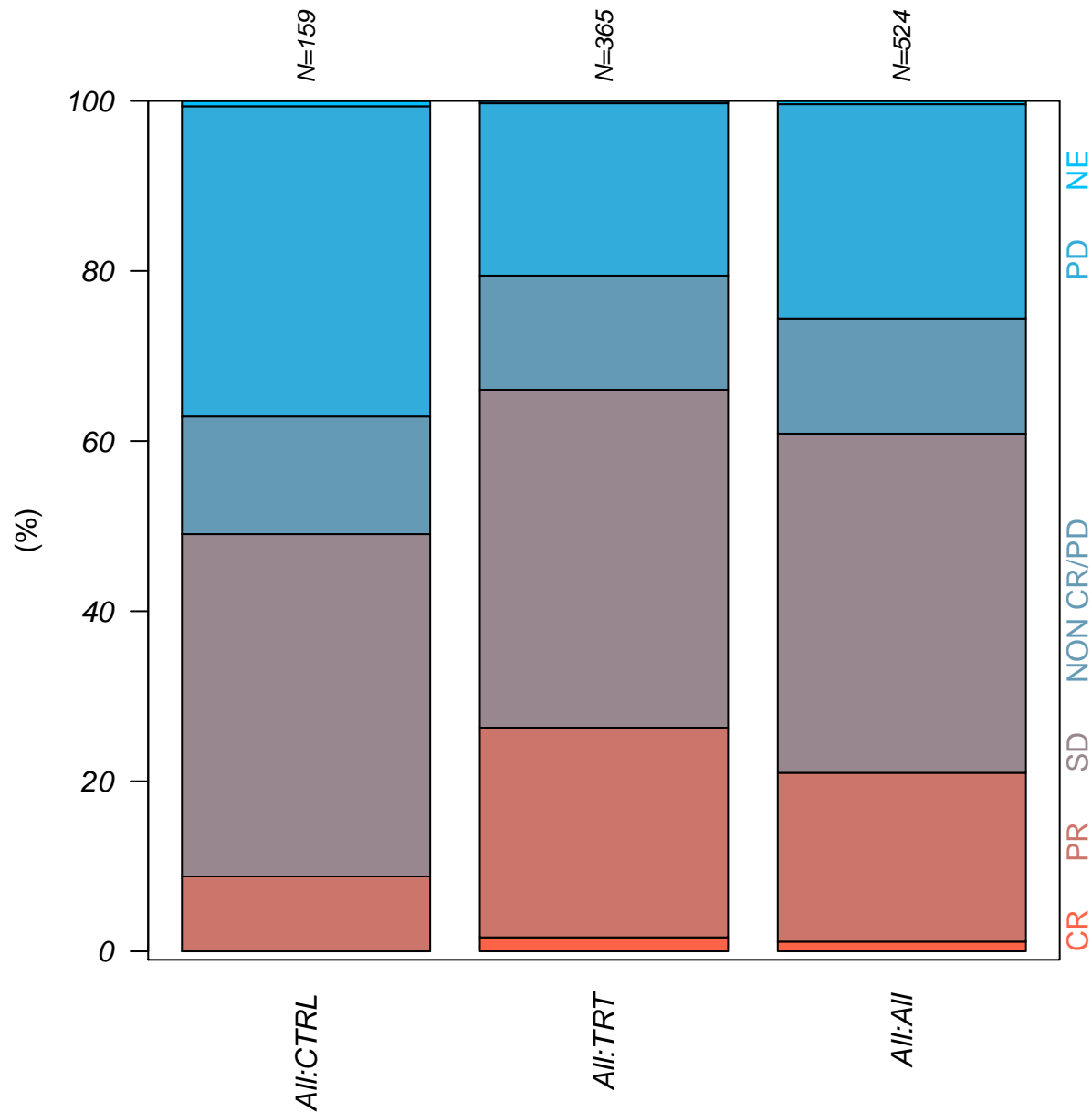


## 5.5 By arm

```
PlotRspBar (input, outcome.var="Response",  
            binary=FALSE,  
            rsp.levels=c("CR", "PR", "SD", "NON CR/PD", "PD", "NE"),  
            trt="Arm")
```

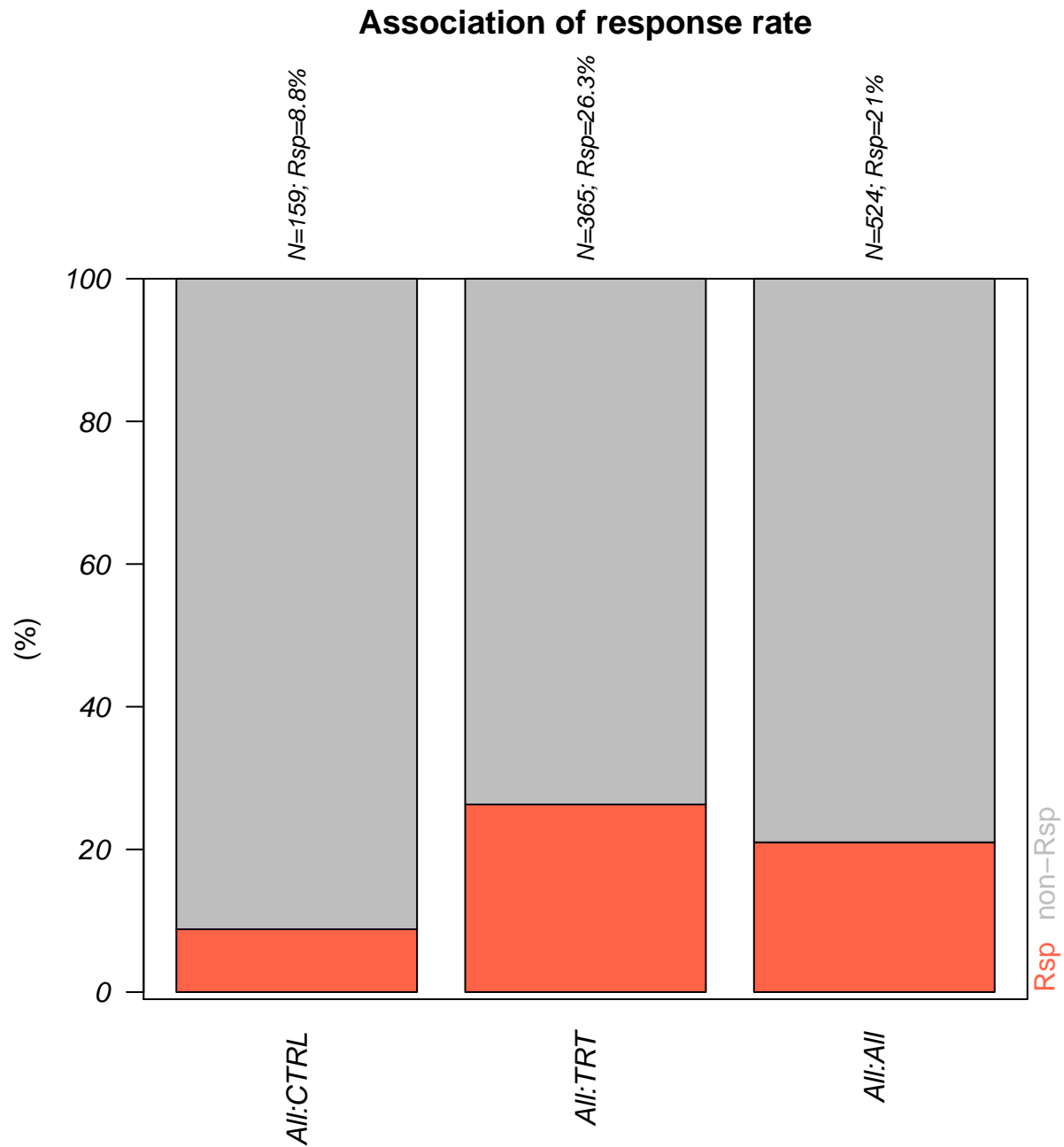
```
## entries with missing outcome.var are removed! 524 entries left
```

## Association of response rate



```
PlotRspBar (input, outcome.var="Response",
  binary=TRUE,
  rsp.response = c("CR","PR"),
  rsp.nonresponse = c("SD", "PD", "NON CR/PD","NE",NA),
  trt="Arm")
```

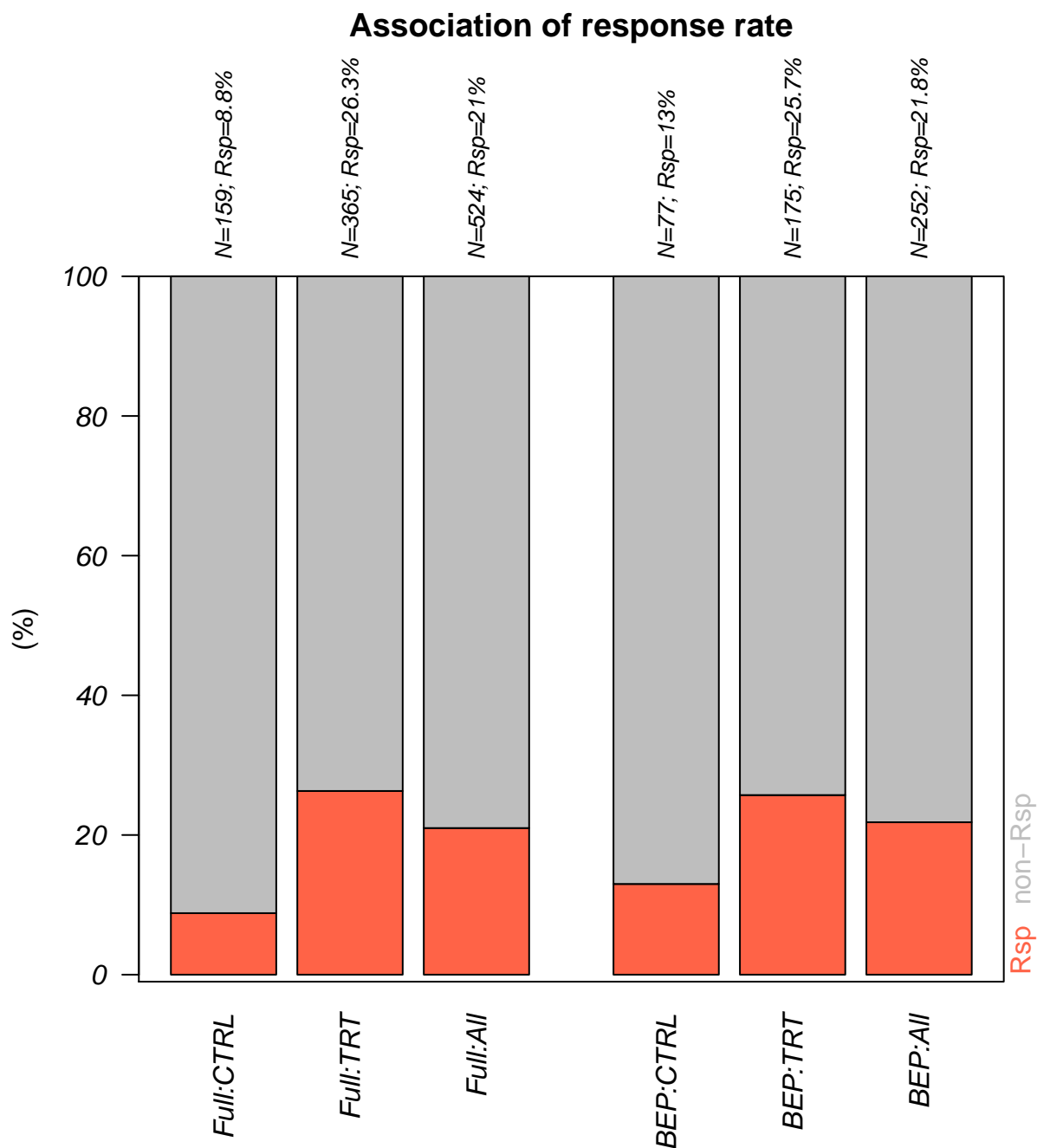
## entries with missing outcome.var are removed! 524 entries left



## 5.6 Compare full population vs BEP

```
PlotRspBar (input, outcome.var="Response",
  binary=TRUE,
  rsp.response = c("CR", "PR"),
  rsp.nonresponse = c("SD", "PD", "NON CR/PD", "NE", NA),
  trt="Arm", compare.bep=TRUE, bep="BEP")
```

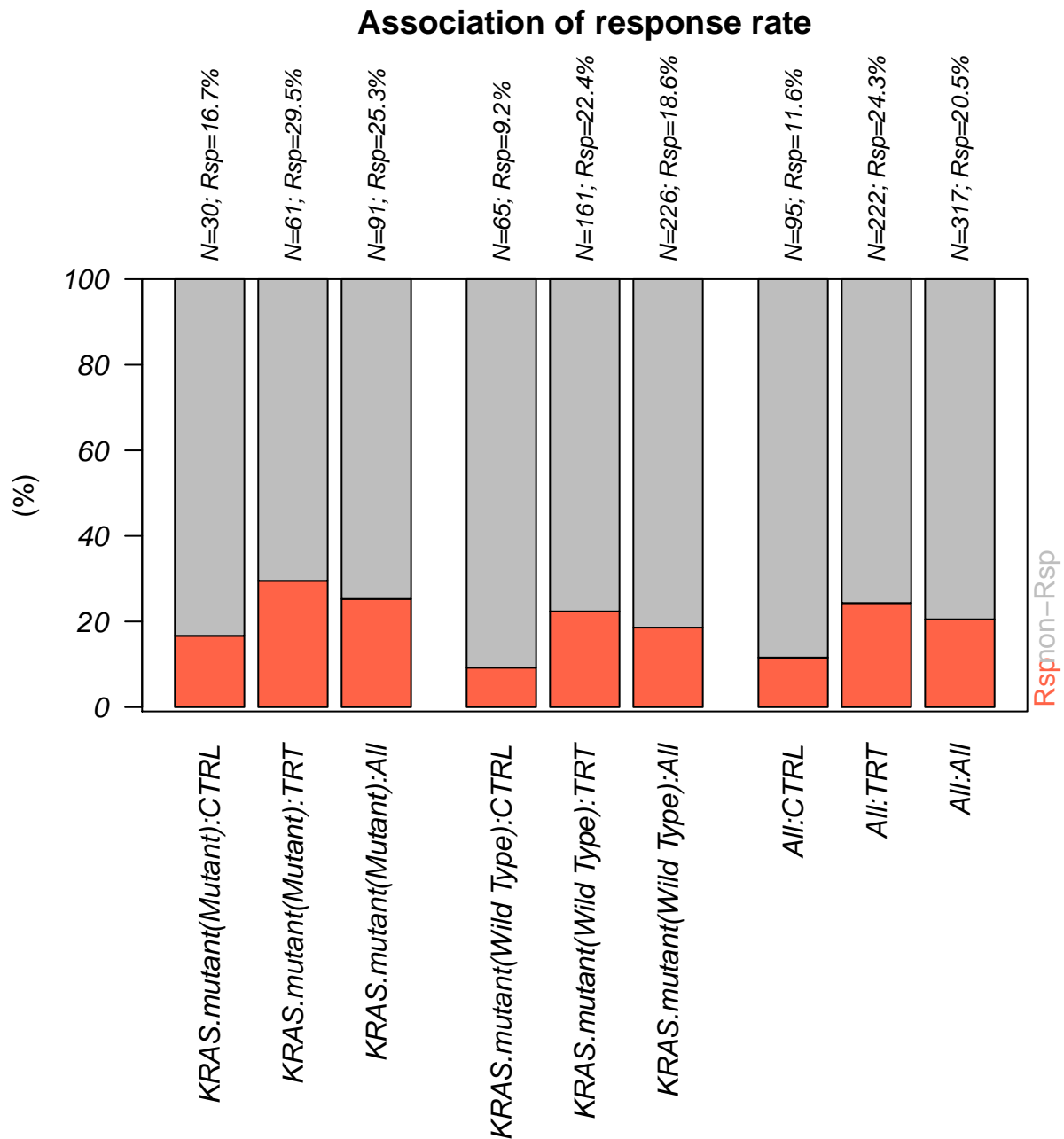
## entries with missing outcome.var are removed! 524 entries left



## 5.7 Compare subgroups

```
PlotRspBar (input, outcome.var="Response",
            binary=TRUE,
            rsp.response = c("CR", "PR"),
            rsp.nonresponse = c("SD", "PD", "NON CR/PD", "NE", NA),
            trt="Arm", compare.var=TRUE, var="KRAS.mutant")
```

## entries with missing outcome.var are removed! 524 entries left

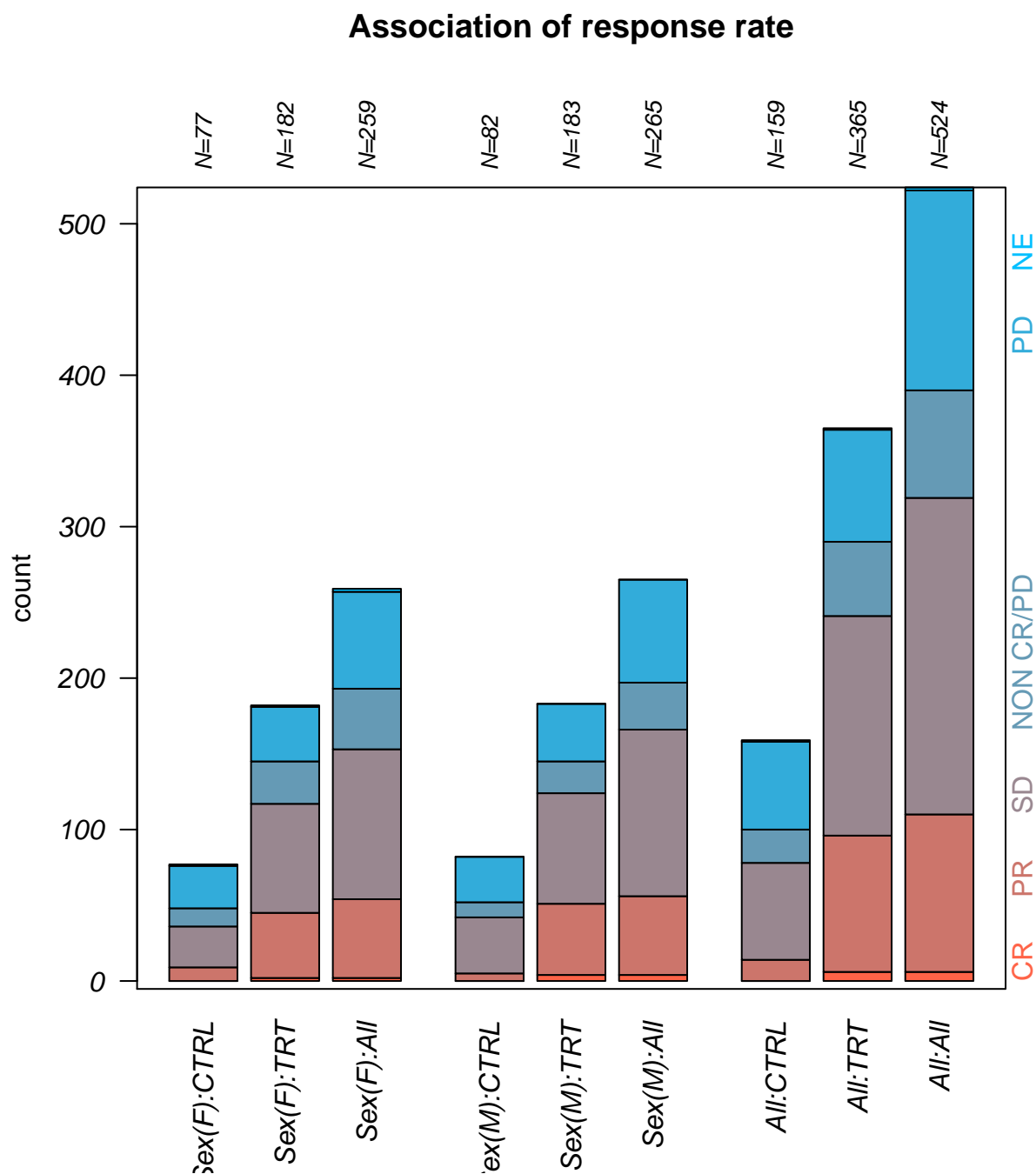


## 5.8 Plot count instead of percentage

```
PlotRspBar (input, outcome.var="Response",
            binary=FALSE,
            rsp.levels=c("CR", "PR", "SD", "NON CR/PD", "PD", "NE"),
            trt="Arm",
            compare.var=TRUE, var="Sex", plot.count = TRUE)
```

## entries with missing outcome.var are removed! 524 entries left

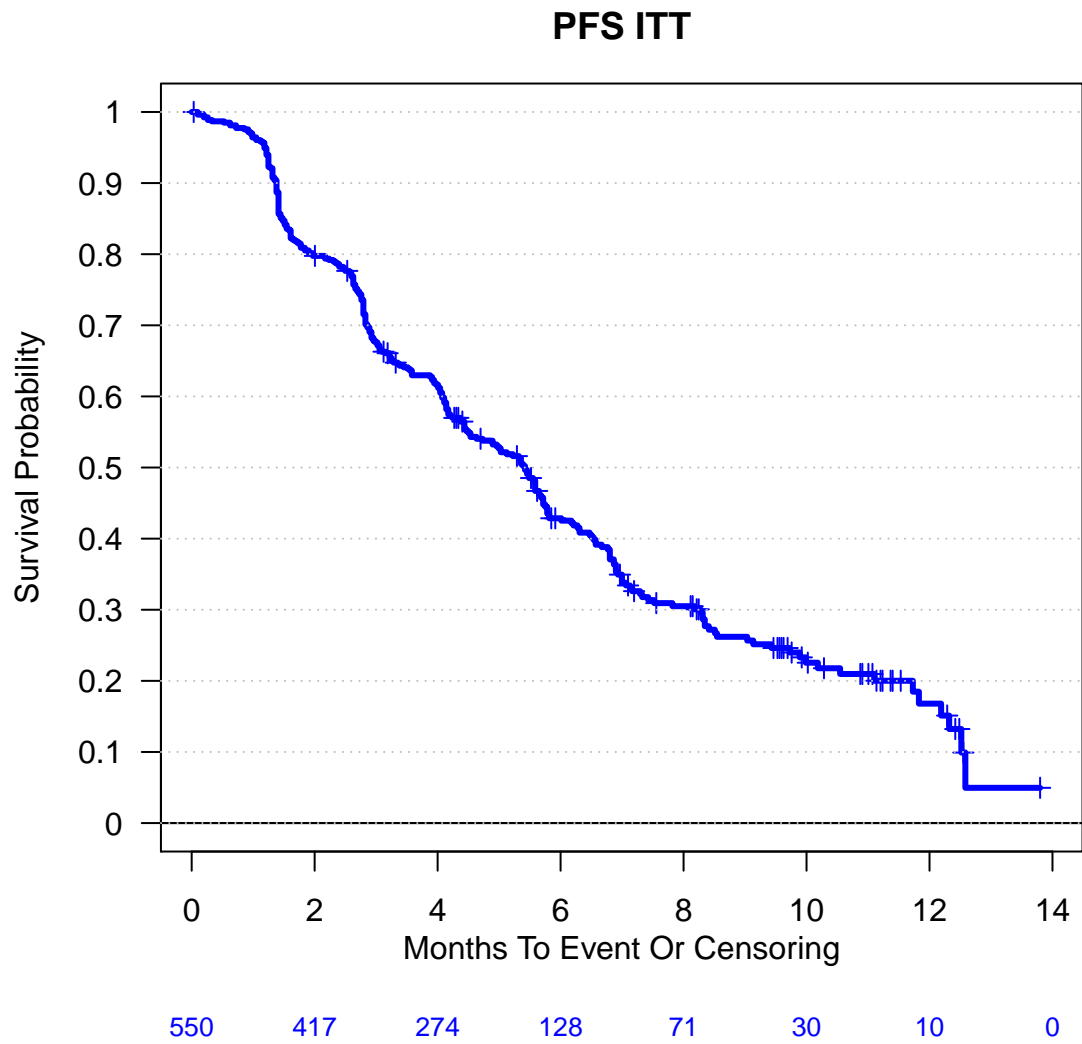




## 6 PlotKM(): Plot KM curves for subpopulations

### 6.1 Without subgroup

```
print(PlotKM(data=sample.data, tte="PFS", cen="PFS.event",
             main="PFS ITT"))
```

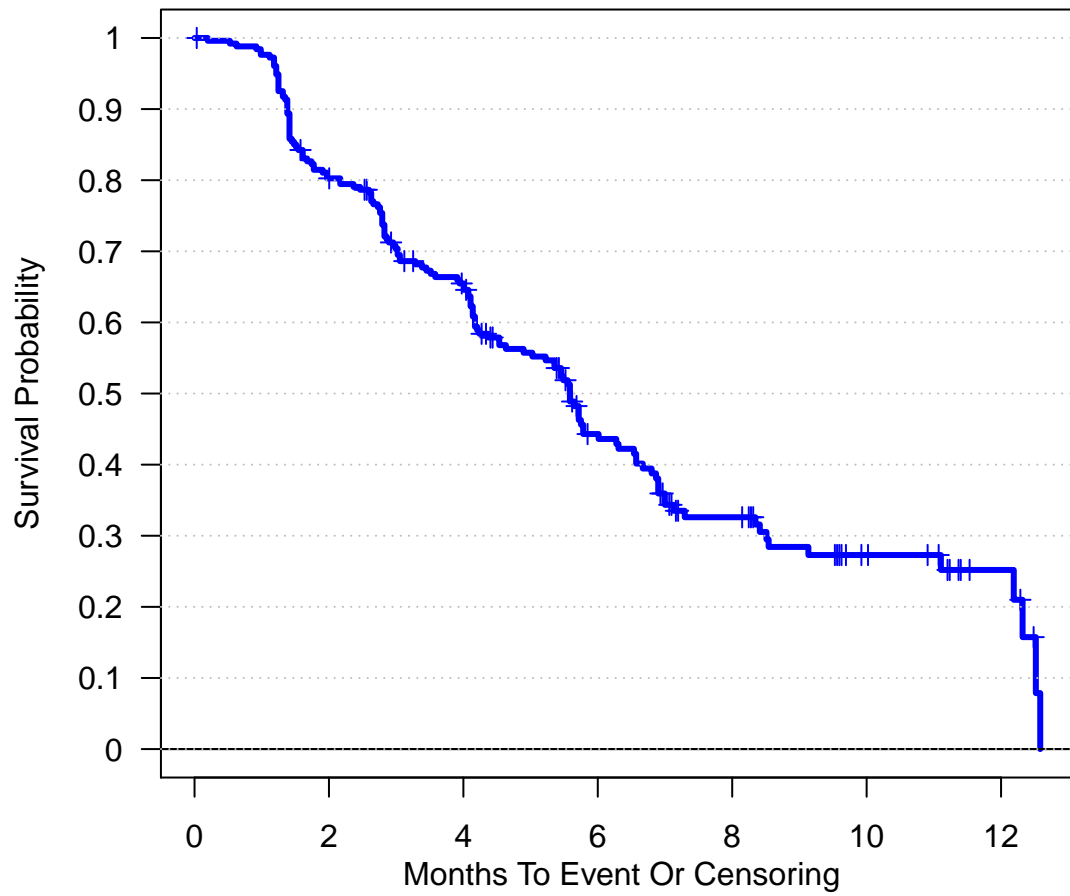


```
## [1] ""
```

## 6.2 Without subgroup, BEP

```
print(PlotKM(data=sample.data, bep="BEP",
             tte="PFS", cen="PFS.event", main="PFS BEP"))
```

## PFS BEP



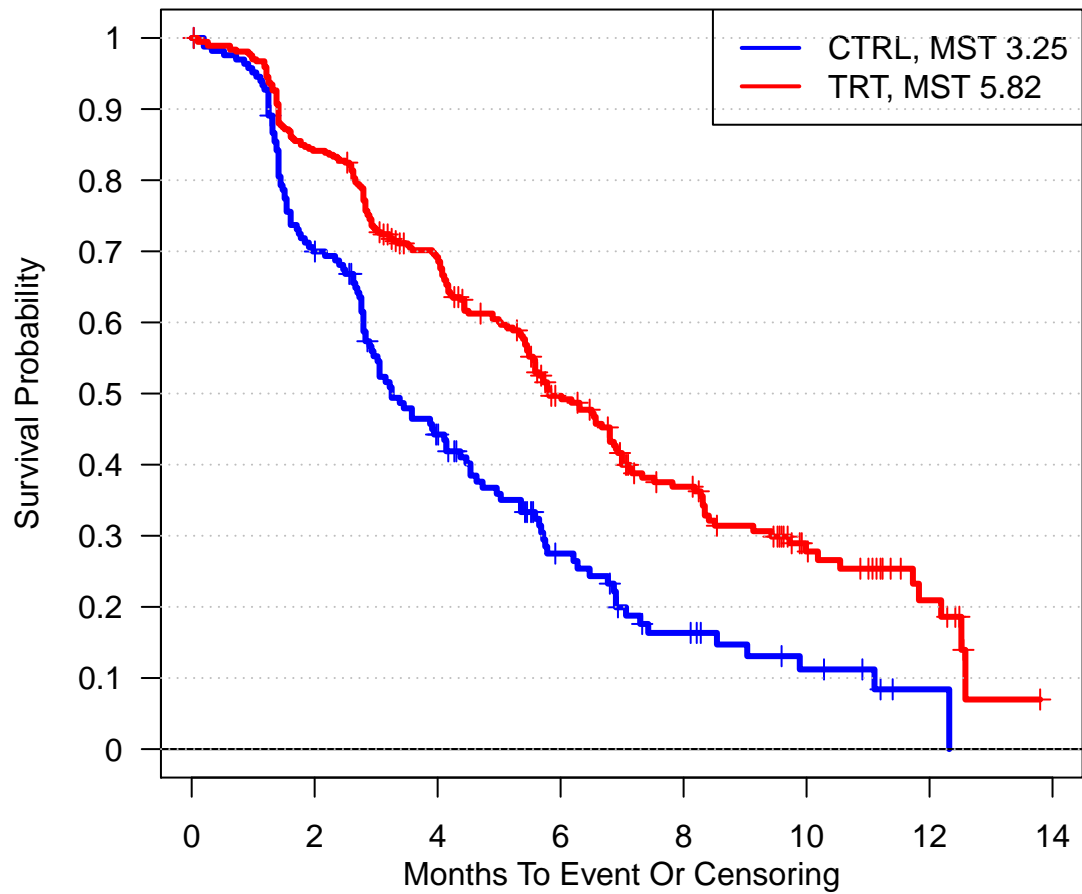
257 201 145 64 36 16 6

## [1] ""

### 6.3 By TRT

```
print(PlotKM(data=sample.data, tte="PFS", cen="PFS.event",
             main="PFS ITT by treatment", trt="Arm"))
```

## PFS ITT by treatment



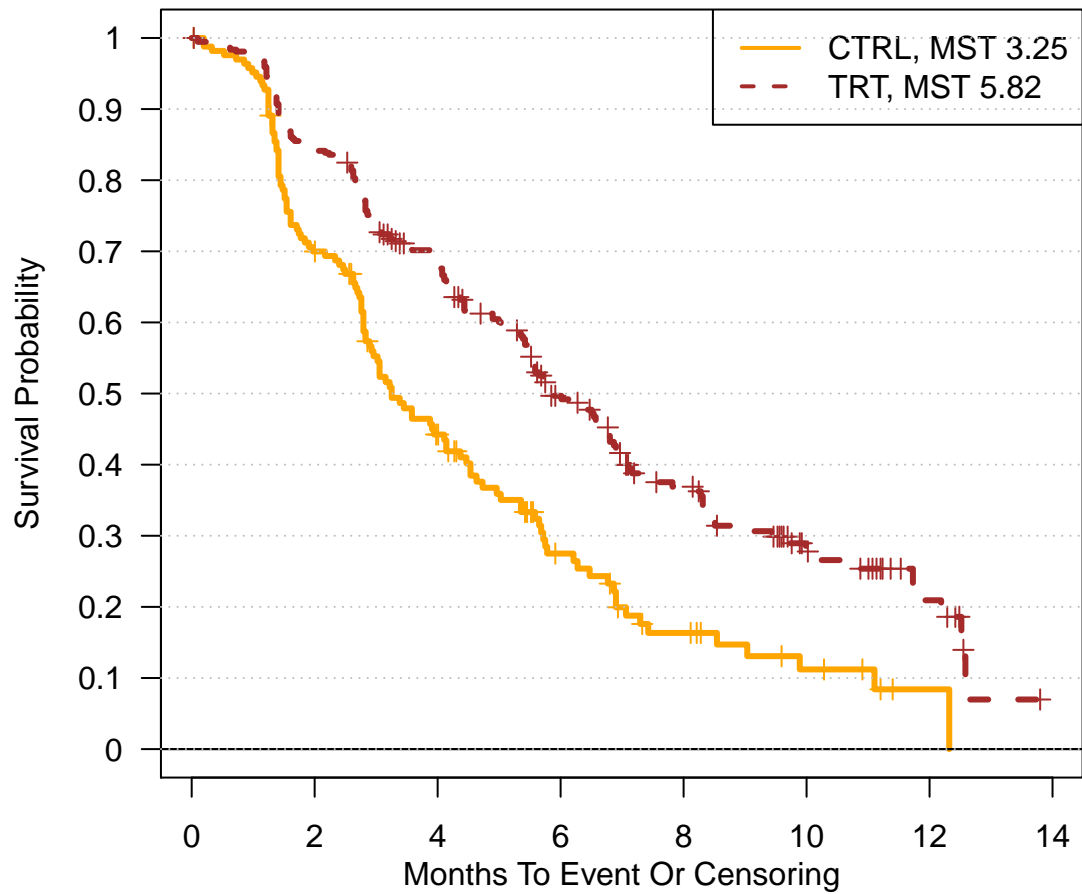
	0	2	4	6	8	10	12	14
CTRL	182	112	58	26	13	6	1	0
TRT	368	305	216	102	58	24	9	0

```
## [1] ""
```

### 6.4 By TRT, change color, line type

```
print(PlotKM(data=sample.data, tte="PFS", cen="PFS.event",
             main="PFS ITT by treatment", trt="Arm",
             col=c("orange", "brown"), lty=c(1,2)))
```

## PFS ITT by treatment



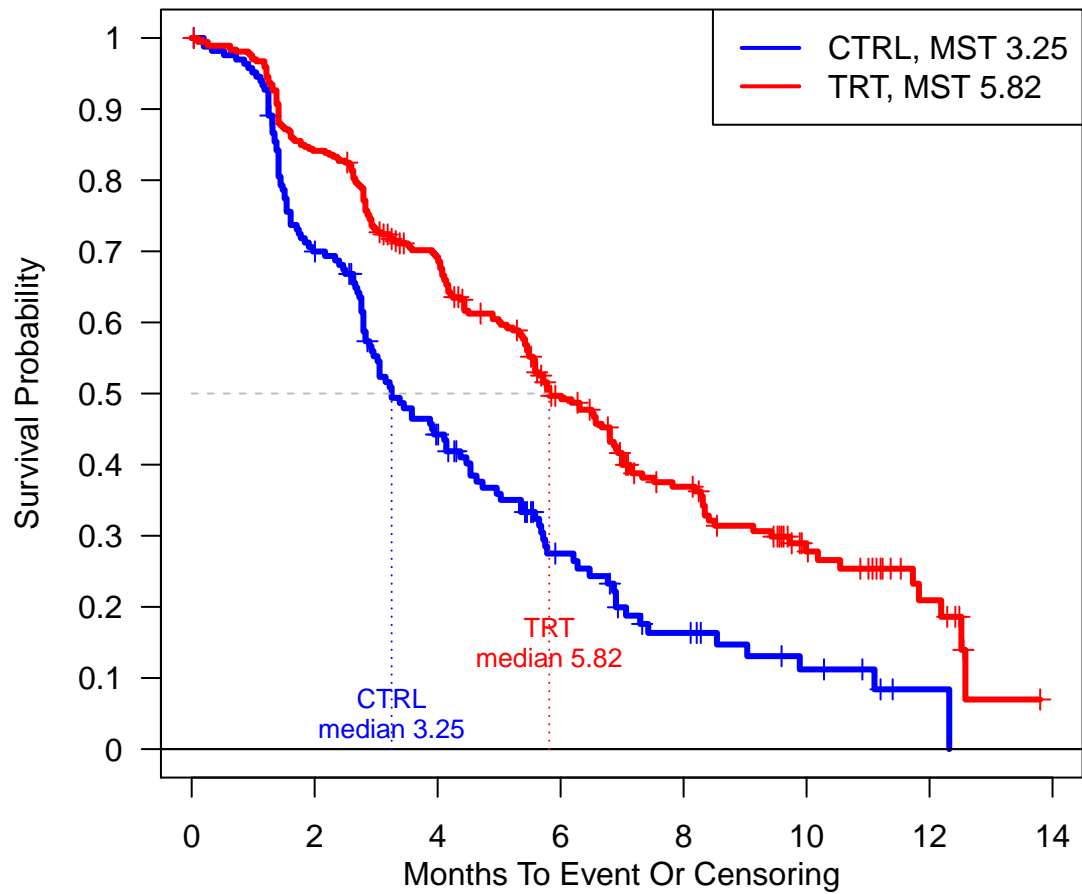
CTRL	182	112	58	26	13	6	1	0
TRT	368	305	216	102	58	24	9	0

```
## [1] ""
```

### 6.5 Mark median PFS, no grid

```
print(PlotKM(data=sample.data, tte="PFS", cen="PFS.event",
             main="PFS ITT by treatment",
             trt="Arm",
             plot.grid = FALSE,
             plot.median=T))
```

## PFS ITT by treatment



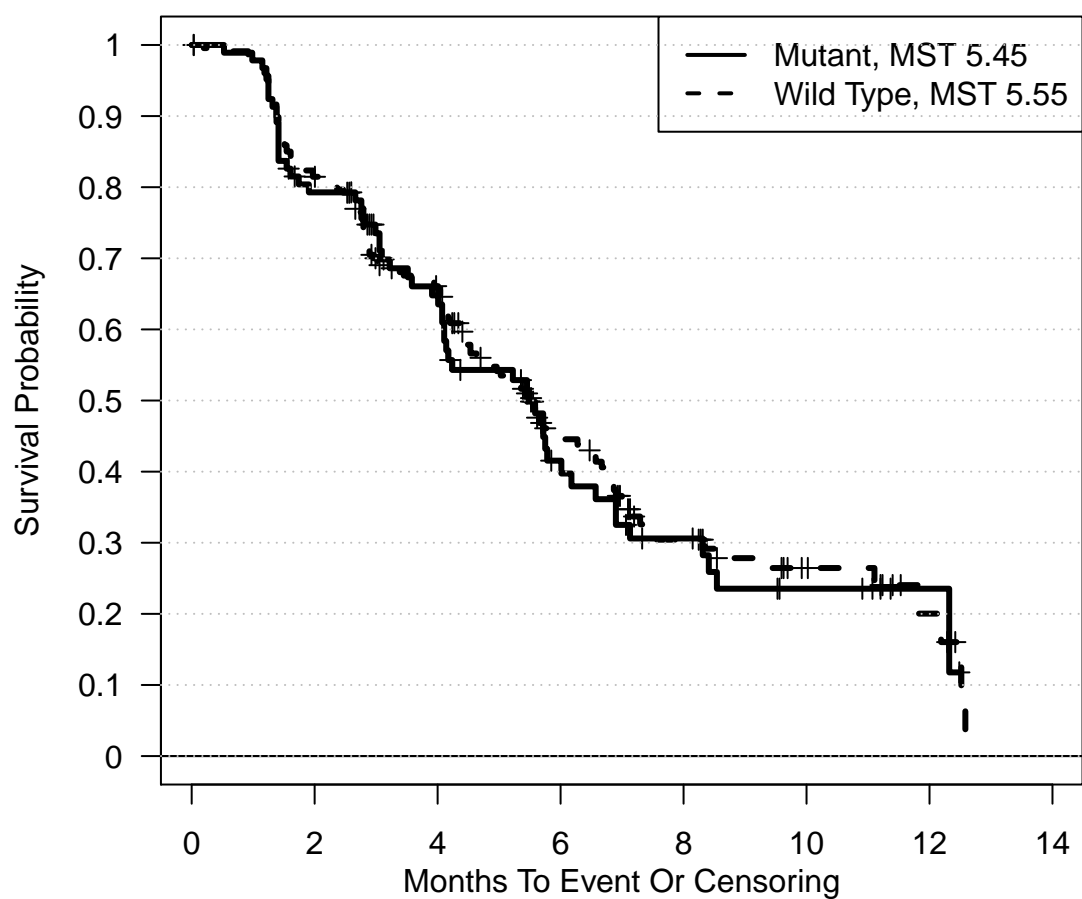
CTRL	182	112	58	26	13	6	1	0
TRT	368	305	216	102	58	24	9	0

```
## [1] ""
```

## 6.6 By KRAS.mutant only

```
print(PlotKM(data=sample.data, tte="PFS", cen="PFS.event",
  main="OS BEP by treatment, by KRAS mutation",
  var="KRAS.mutant"))
```

## OS BEP by treatment, by KRAS mutation

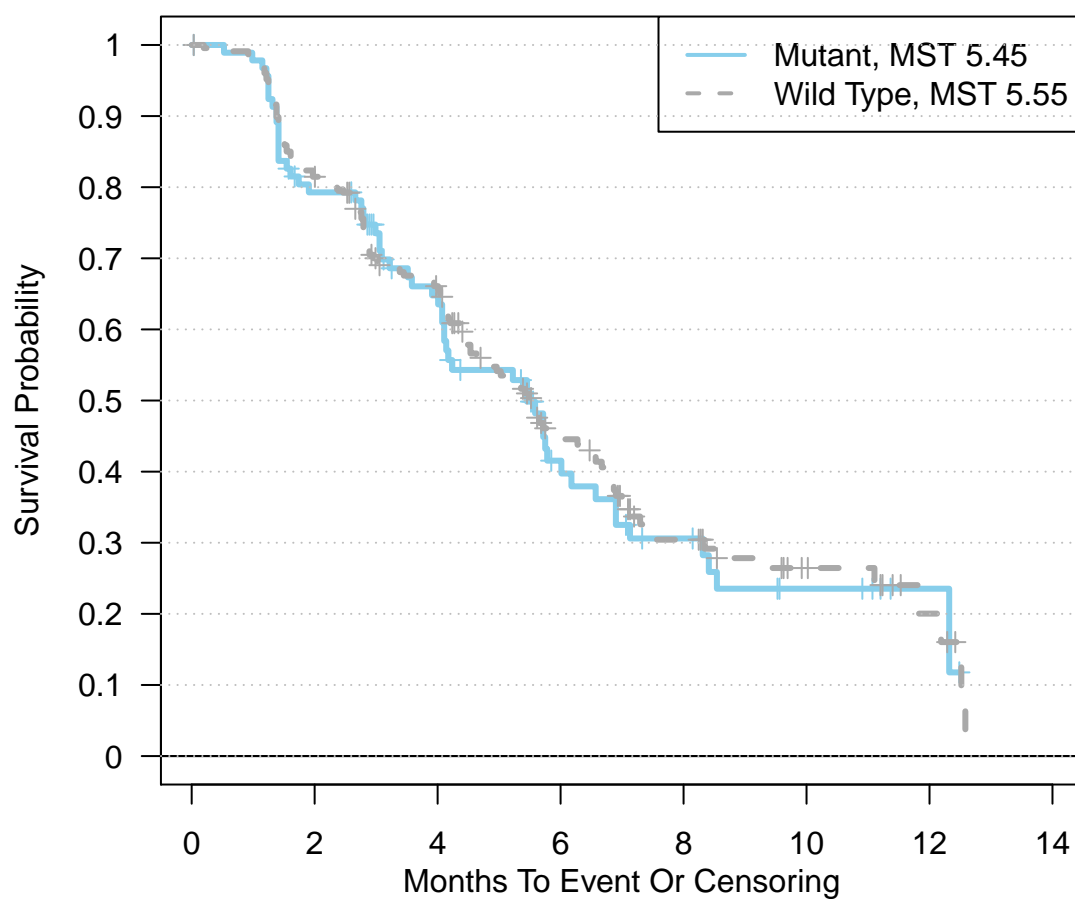


Mutant	95	71	51	23	14	7	2	0
Wild Type	231	183	133	57	28	12	5	0

```
## [1] ""
```

```
print(PlotKM(data=sample.data, tte="PFS", cen="PFS.event",
  main="OS BEP by treatment, by KRAS mutation",
  var="KRAS.mutant", col=c("skyblue", "darkgray")))
```

## OS BEP by treatment, by KRAS mutation



	0	2	4	6	8	10	12	14
Mutant	95	71	51	23	14	7	2	0
Wild Type	231	183	133	57	28	12	5	0

```
## [1] ""
```

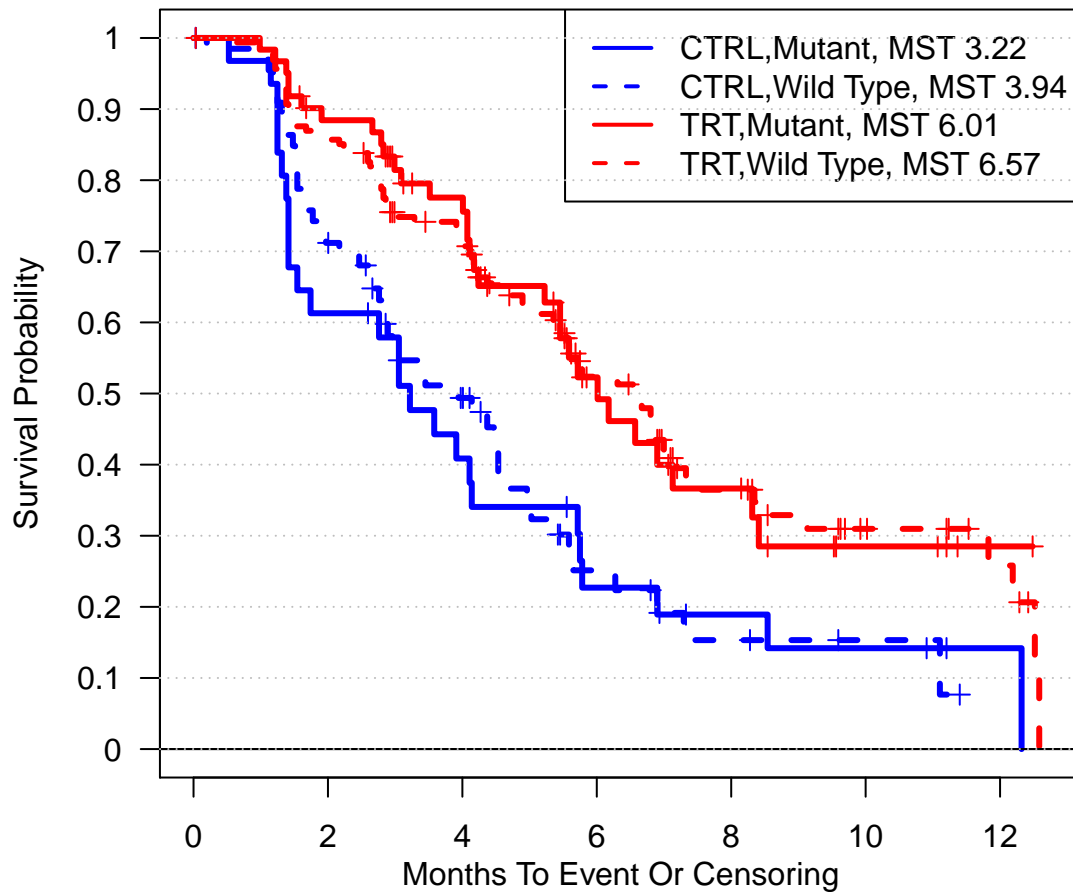
### 6.7 By TRT and KRAS.mutant

```
print(PlotKM(data=sample.data, tte="PFS", cen="PFS.event",
             main="OS BEP by treatment, by KRAS mutation",
             trt="Arm", var="KRAS.mutant"))
```

```
## entries who have NA in trt, var, or varlist are removed
```



## OS BEP by treatment, by KRAS mutation



CTRL, Mutant	34	19	12	6	4	3	1
CTRL, Wild Type	69	46	27	9	4	2	0
TRT, Mutant	61	52	39	17	10	4	1
TRT, Wild Type	162	137	106	48	24	10	5

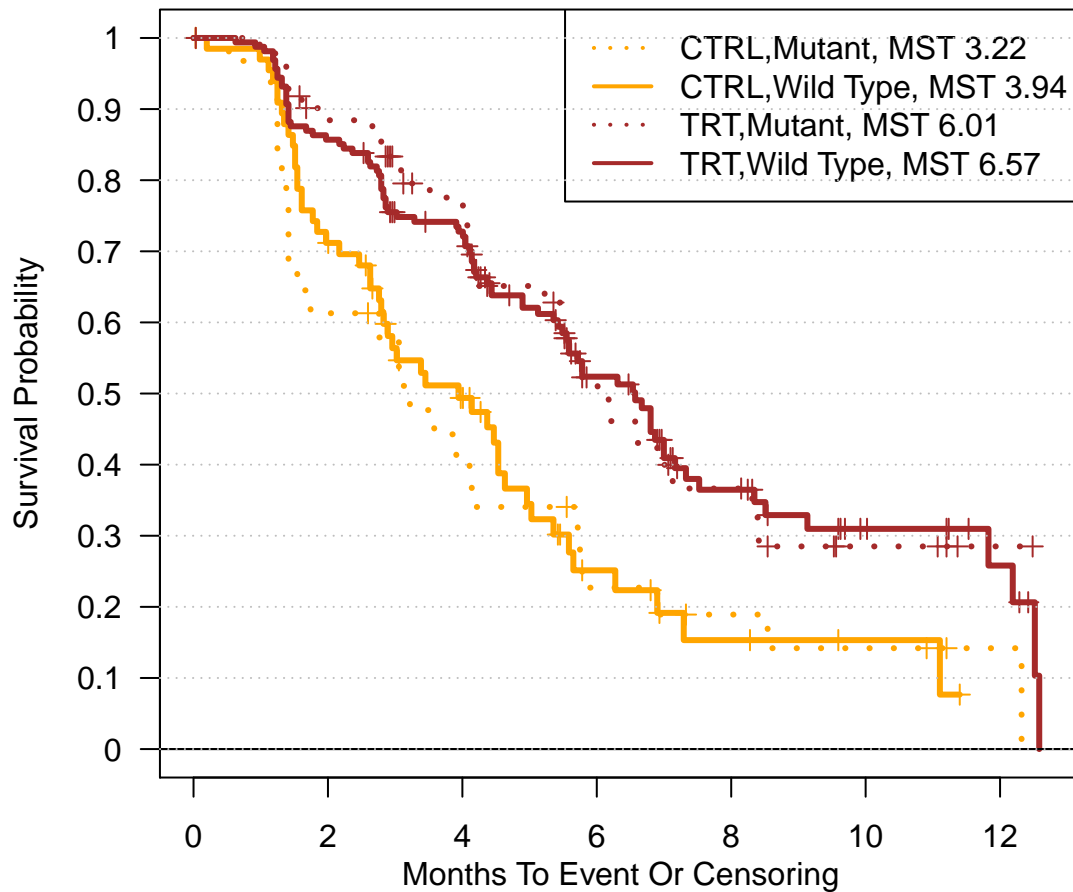
```
## [1] ""
```

### 6.8 By TRT and KRAS.mutant, change color and line type

```
print(PlotKM(data=sample.data, tte="PFS", cen="PFS.event",
  main="PFS BEP by treatment, by KRAS mutation",
  trt="Arm", var="KRAS.mutant",
  col=c("orange", "orange", "brown", "brown"),
  lty=c(3, 1, 3, 1)))
```

```
## entries who have NA in trt, var, or varlist are removed
```

## PFS BEP by treatment, by KRAS mutation



CTRL, Mutant	34	19	12	6	4	3	1
CTRL, Wild Type	69	46	27	9	4	2	0
TRT, Mutant	61	52	39	17	10	4	1
TRT, Wild Type	162	137	106	48	24	10	5

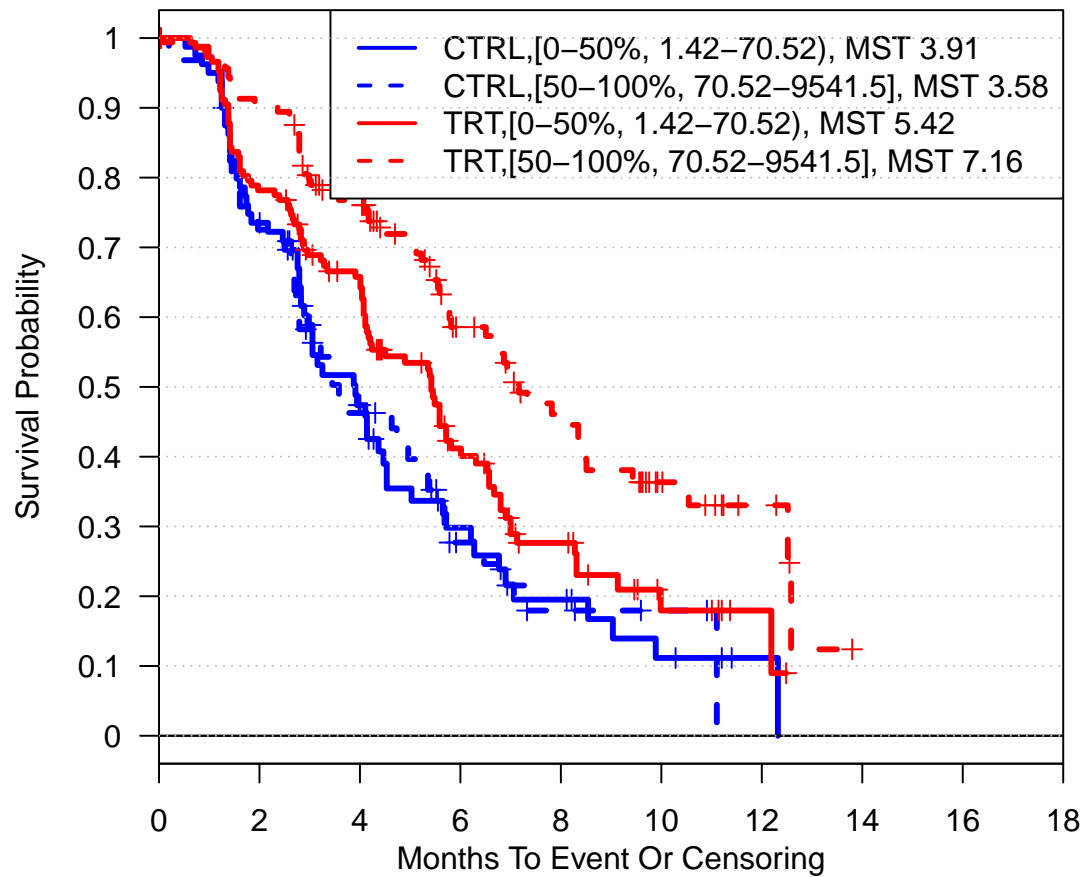
```
## [1] ""
```

### 6.9 By TRT and KRAS.expression (continuous biomarker, cut at median)

```
print(PlotKM(data=sample.data, tte="PFS", cen="PFS.event",
  main="PFS BEP by treatment, by KRAS expression",
  trt="Arm", var="KRAS.exprs", var.class="continuous",
  percentile.cutoff=0.5, xlim=c(0,18))
)
```

```
## entries who have NA in trt, var, or varlist are removed
```

## PFS BEP by treatment, by KRAS expression



CTRL,[0-50%, 1.42-70.52)	82	58	31	15	9	4	1	0	0	0
CTRL,[50-100%, 70.52-9541.5]	70	44	23	9	4	2	0	0	0	0
TRT,[0-50%, 1.42-70.52)	147	113	83	38	20	6	2	0	0	0
TRT,[50-100%, 70.52-9541.5]	162	145	105	47	30	12	5	0	0	0

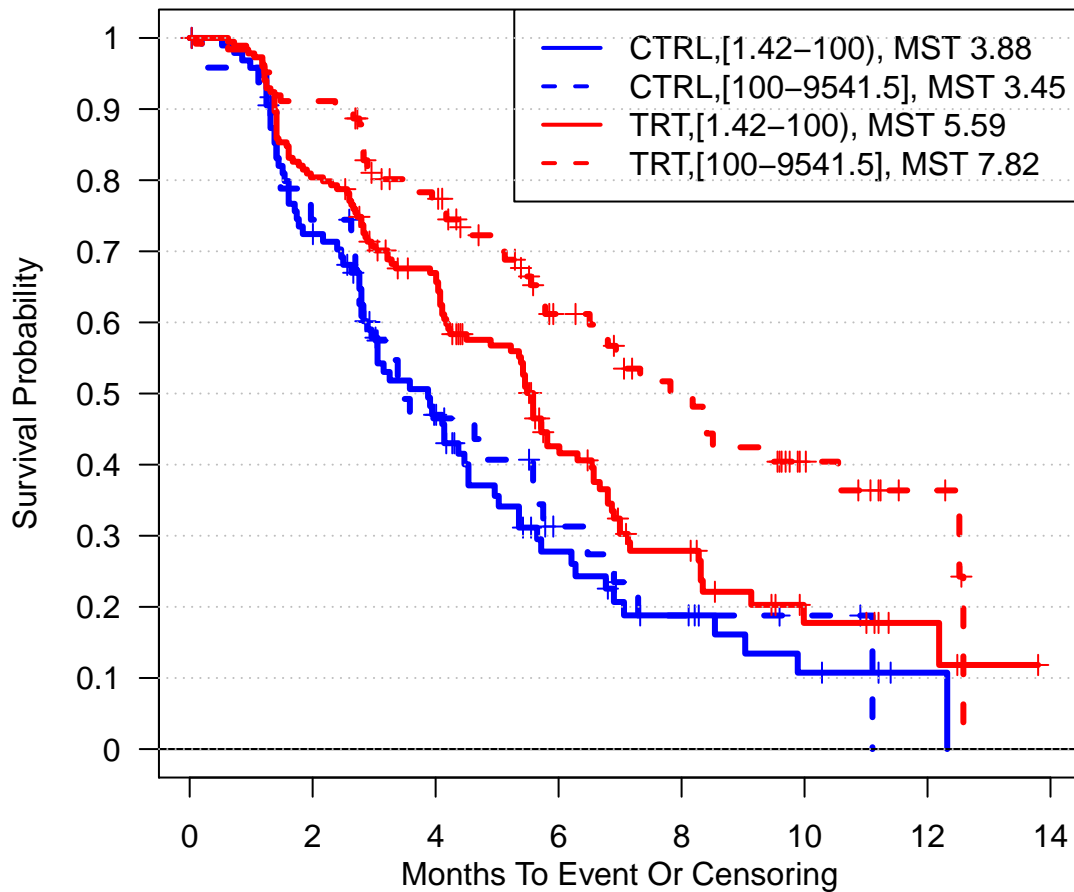
```
## [1] ""
```

### 6.10 By TRT and KRAS.expression (continuous biomarker, cut at a numerical cutoff)

```
print(PlotKM(data=sample.data, tte="PFS", cen="PFS.event",
  main="PFS BEP by treatment, by KRAS expression",
  trt="Arm", var="KRAS.exprs", var.class="continuous",
  numerical.cutoff=100)
)
```

```
## entries who have NA in trt, var, or varlist are removed
```

## PFS BEP by treatment, by KRAS expression



CTRL,[1.42-100)	99	68	37	16	9	4	1	0
CTRL,[100-9541.5]	53	34	17	8	4	2	0	0
TRT,[1.42-100)	184	146	104	43	22	7	3	0
TRT,[100-9541.5]	125	112	84	42	28	11	4	0

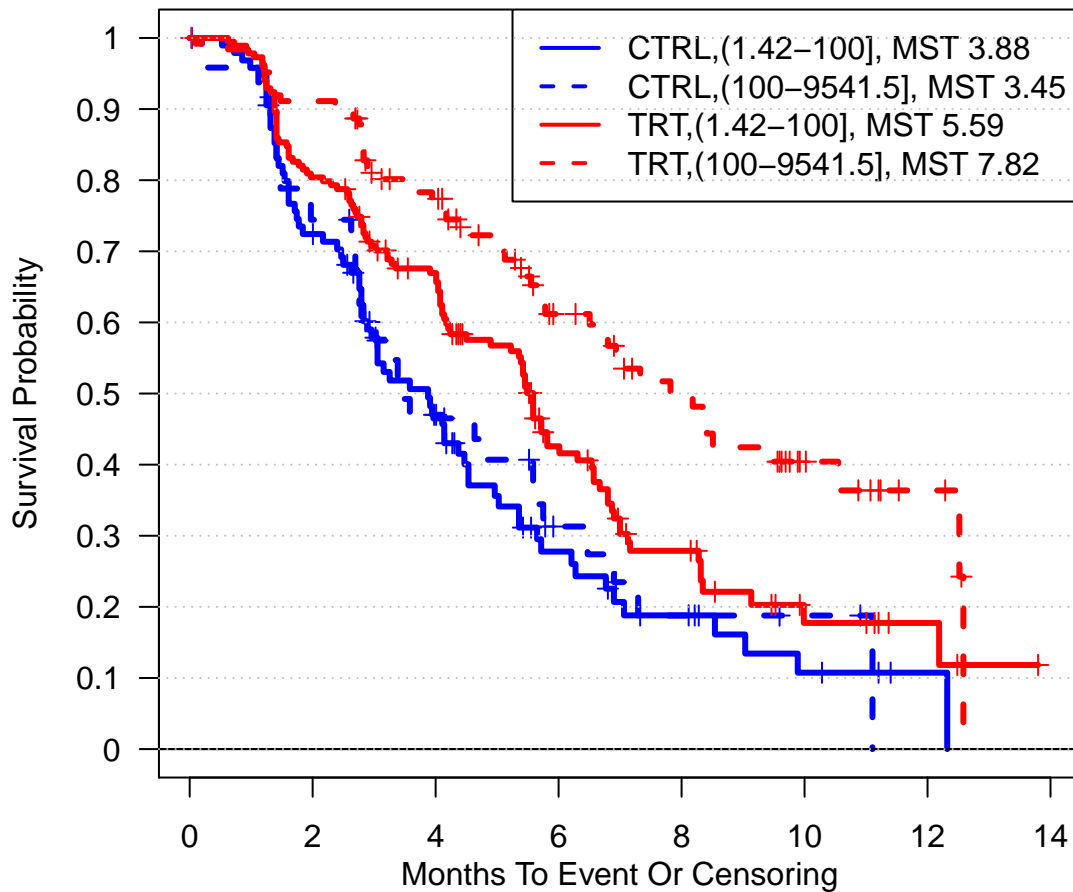
```
## [1] ""
```

6.11 By TRT and KRAS.expression (continuous biomarker, cut at a numerical cutoff). High group defined as “greater than”, instead of “greater than or equal to”

```
print(PlotKM(data=sample.data, tte="PFS", cen="PFS.event",
  main="PFS BEP by treatment, by KRAS expression",
  trt="Arm", var="KRAS.exprs", var.class="continuous",
  numerical.cutoff=100, equal.in.high = F)
)
```

```
## entries who have NA in trt, var, or varlist are removed
```

## PFS BEP by treatment, by KRAS expression



CTRL,(1.42-100]	99	68	37	16	9	4	1	0
CTRL,(100-9541.5]	53	34	17	8	4	2	0	0
TRT,(1.42-100]	184	146	104	43	22	7	3	0
TRT,(100-9541.5]	125	112	84	42	28	11	4	0

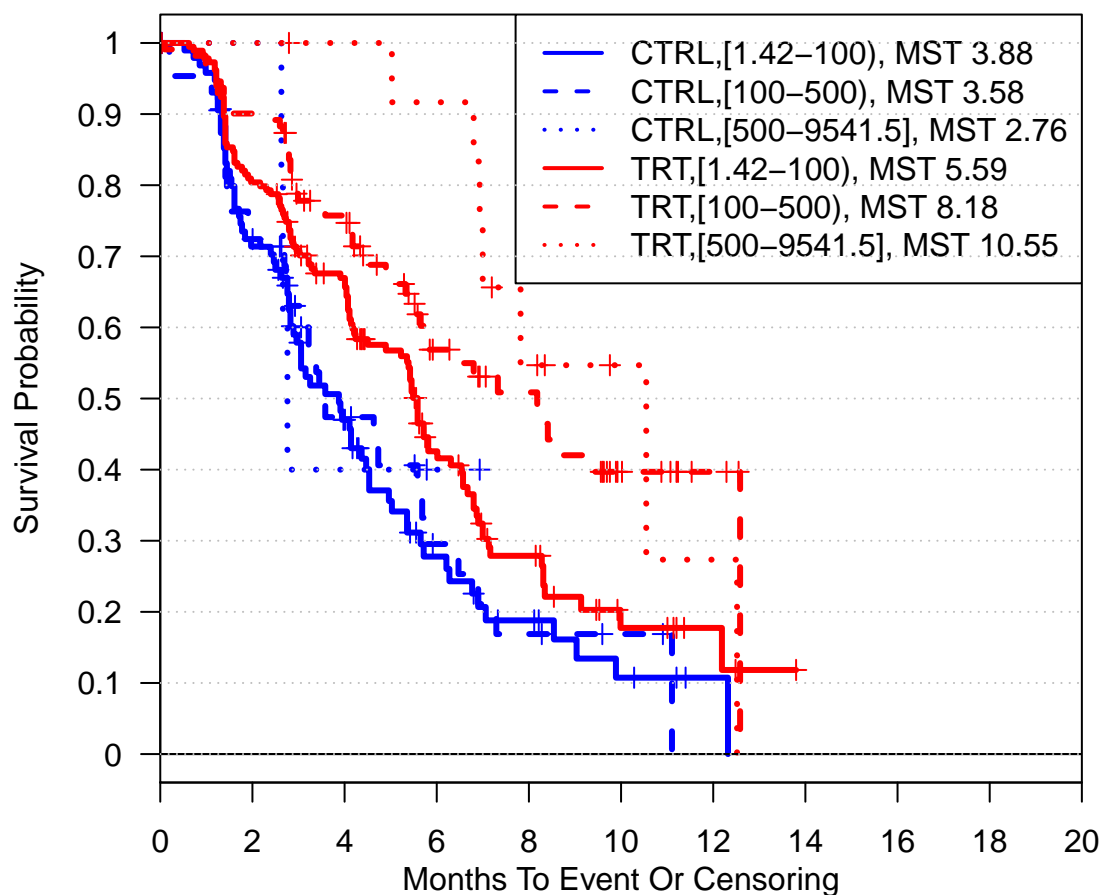
```
## [1] ""
```

### 6.12 By TRT and KRAS.expression , more than 2 groups

```
print(PlotKM(data=sample.data, tte="PFS", cen="PFS.event",
  main="PFS BEP by treatment, by KRAS expression",
  trt="Arm", var="KRAS.exprs", var.class="continuous",
  numerical.cutoff=c(100,500), xlim=c(0, 20))
)
```

```
## entries who have NA in trt, var, or varlist are removed
```

## PFS BEP by treatment, by KRAS expression



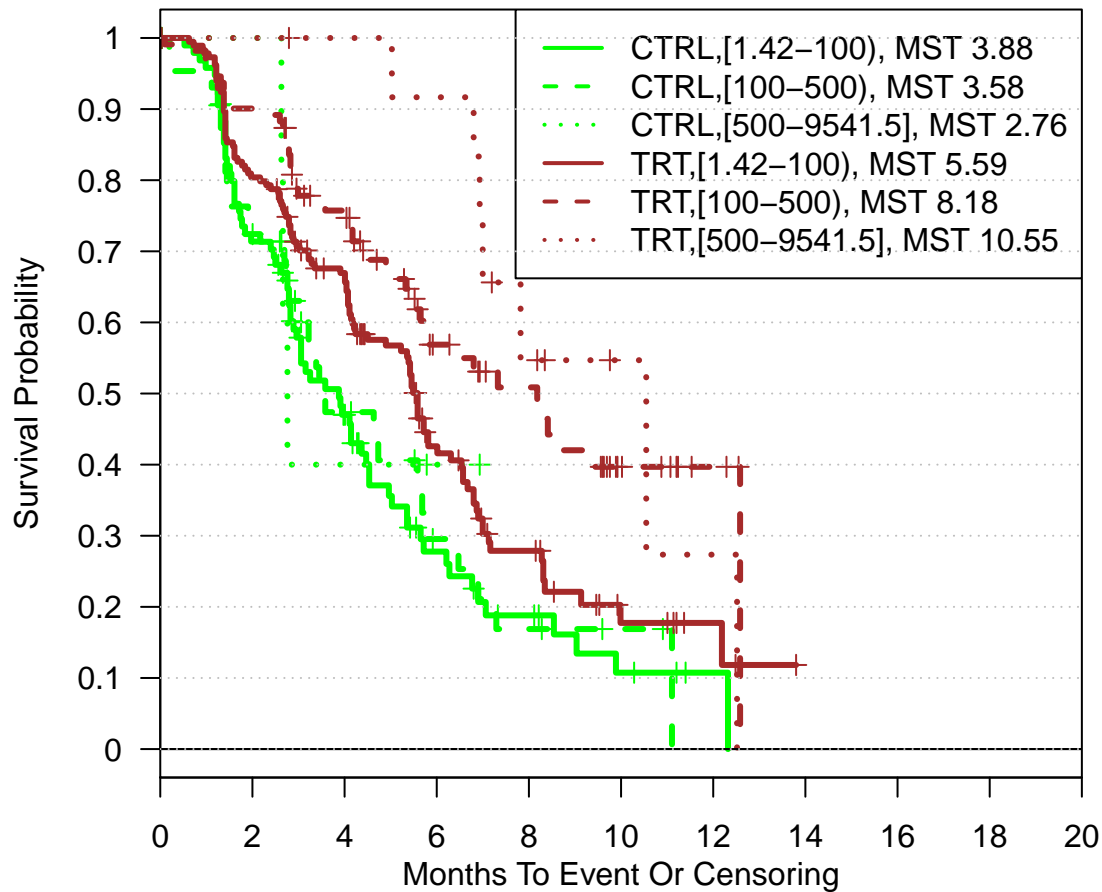
CTRL,[1.42-100)	99	68	37	16	9	4	1	0	0	0	0
CTRL,[100-500)	47	29	15	7	4	2	0	0	0	0	0
CTRL,[500-9541.5]	6	5	2	1	0	0	0	0	0	0	0
TRT,[1.42-100)	184	146	104	43	22	7	3	0	0	0	0
TRT,[100-500)	112	99	72	31	23	9	3	0	0	0	0
TRT,[500-9541.5]	13	13	12	11	5	2	1	0	0	0	0

```
## [1] ""
```

```
print(PlotKM(data=sample.data, tte="PFS", cen="PFS.event",
  main="PFS BEP by treatment, by KRAS expression",
  trt="Arm", var="KRAS.exprs", var.class="continuous",
  numerical.cutoff=c(100,500),
  col=c("green","green","green","brown","brown","brown"),
  xlim=c(0,20))
)
```

```
## entries who have NA in trt, var, or varlist are removed
```

## PFS BEP by treatment, by KRAS expression



CTRL,[1.42-100)	99	68	37	16	9	4	1	0	0	0	0
CTRL,[100-500)	47	29	15	7	4	2	0	0	0	0	0
CTRL,[500-9541.5]	6	5	2	1	0	0	0	0	0	0	0
TRT,[1.42-100)	184	146	104	43	22	7	3	0	0	0	0
TRT,[100-500)	112	99	72	31	23	9	3	0	0	0	0
TRT,[500-9541.5]	13	13	12	11	5	2	1	0	0	0	0

```
## [1] ""
```

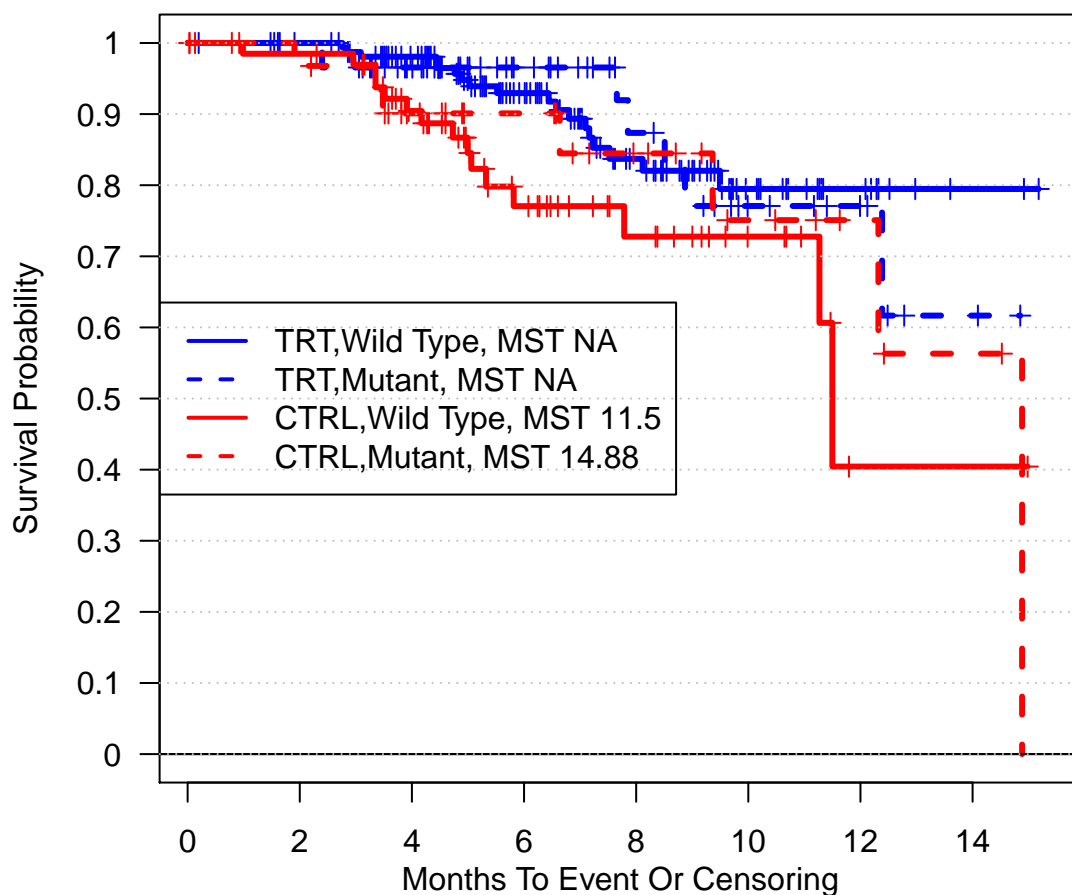
### 6.13 More flexibility, reorder subgroups

```
print(PlotKM(data=sample.data, tte="OS",cen="OS.event",
  main="OS BEP by treatment, by KRAS mutation",
  varlist=c("Arm","KRAS.mutant"),
  varlist.levels=list(c("TRT","CTRL"),c("Wild Type","Mutant")),
  legend.loc="left"))
```

```
## 'varlist' is specified, trt and var parameters will be ignored
```

```
## entries who have NA in trt, var, or varlist are removed
```

## OS BEP by treatment, by KRAS mutation



TRT, Wild Type	162	159	137	87	50	22	10	2
TRT, Mutant	61	58	48	31	19	10	6	2
CTRL, Wild Type	69	65	52	28	17	9	1	1
CTRL, Mutant	34	30	23	18	12	7	4	2

## [1] ""

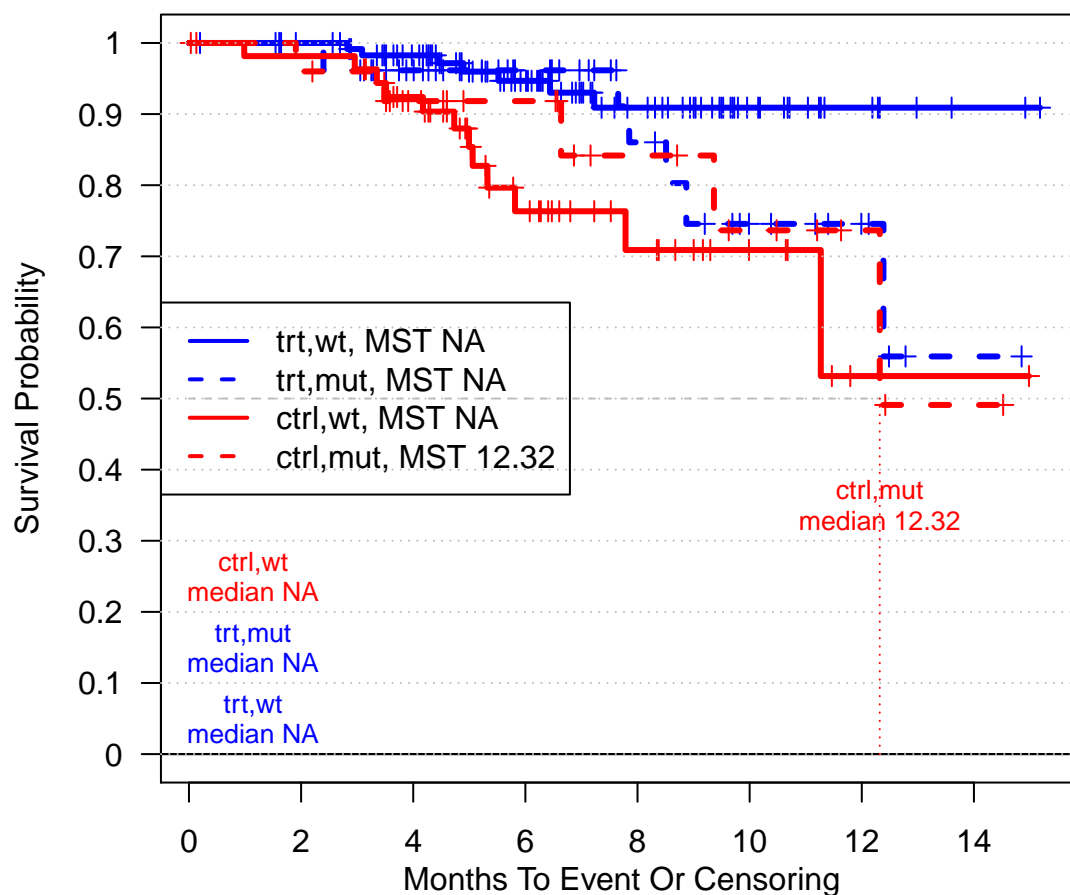
### 6.14 By TRT and KRAS.mutant, reorder and rename

```
print(PlotKM(data=sample.data, tte="OS", cen="OS.event", bep="BEP",
  main="OS BEP by treatment, by KRAS mutation", varlist=c("Arm", "KRAS.mutant"),
  varlist.levels=list(c("TRT", "CTRL"), c("Wild Type", "Mutant")),
  varlist.labels=list(c("trt", "ctrl"), c("wt", "mut")),
  plot.median=T, legend.loc="left"))
```

## 'varlist' is specified, trt and var parameters will be ignored



## OS BEP by treatment, by KRAS mutation



trt,wt	121	119	101	66	38	17	7	2
trt,mut	55	52	43	28	17	9	5	1
ctrl,wt	56	53	45	23	13	6	1	1
ctrl,mut	25	24	18	14	9	6	3	1

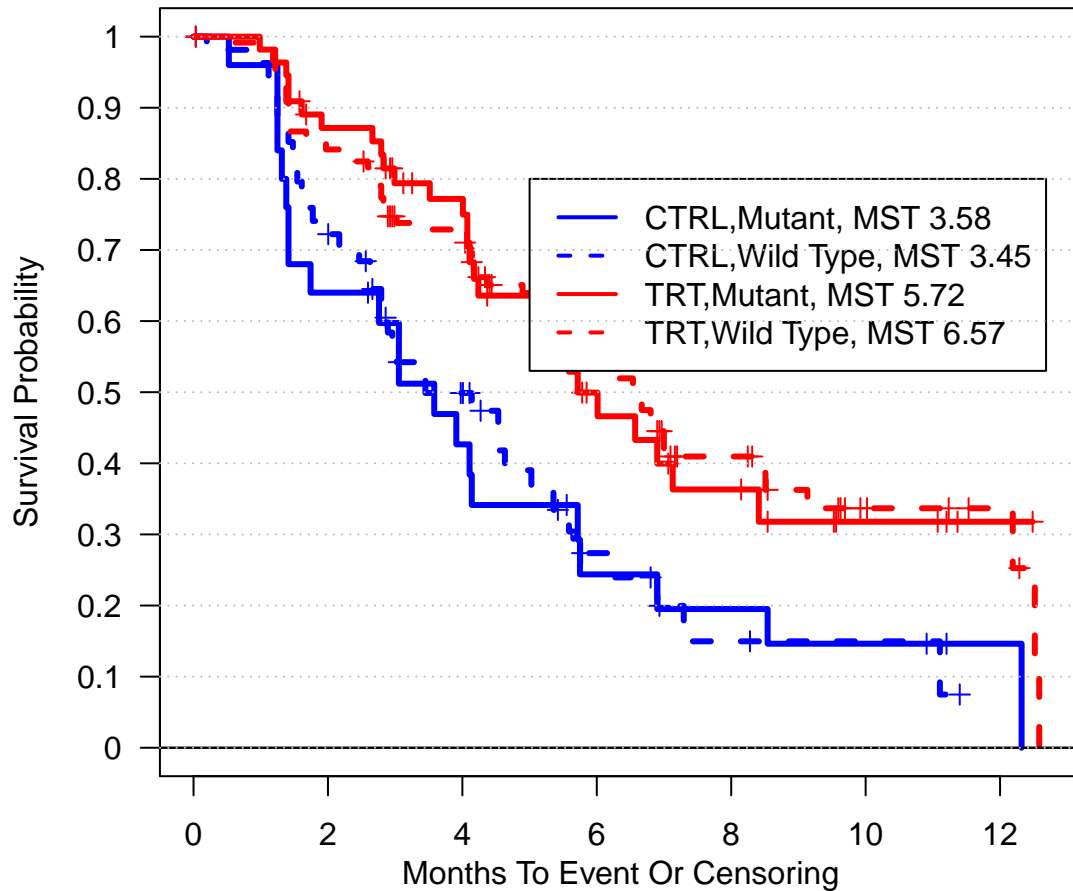
```
## [1] ""
```

### 6.15 Legend location

```
print(PlotKM(data=sample.data, tte="PFS", cen="PFS.event",
  var=c("Arm", "KRAS.mutant"),
  bep="BEP", legend.loc=NULL, legend.x=5, legend.y=.8))
```

```
## more than one elements in 'var', trt parameter will be ignored
```

```
## 'varlist' is specified, trt and var parameters will be ignored
```



CTRL,Mutant	25	16	10	5	4	3	1
CTRL,Wild Type	56	39	22	8	3	2	0
TRT,Mutant	55	46	35	15	9	4	1
TRT,Wild Type	121	100	78	36	20	7	4

```
## [1] ""
```

## 7 PlotTabForestBiomarker() : forest plot and summary statistics table for a single biomarker (include cutoff exploration for cont. biomarker )

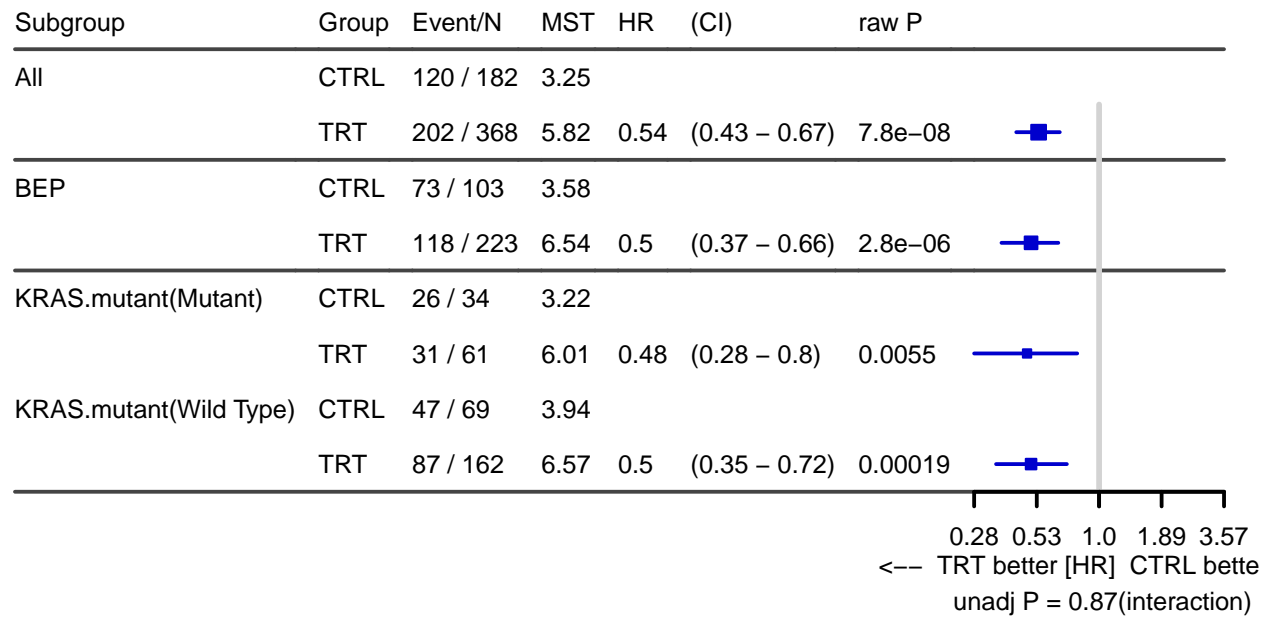
### 7.1 Survival outcome, 2-arm, categorical variable

by default BEP will be defined as patients non-NA biomarker measurement

```
PlotTabForestBiomarker(data=input,
                        outcome.class="survival",
                        outcome.var=c("PFS","PFS.event"),
                        trt="Arm",
                        var="KRAS.mutant",
                        var.class="categorical")
```

## Some NAs in var column, will define the non NA entries as BEP

**Across-arm Effect of Biomarker**  
**PFS, KRAS.mutant**  
**Unadjusted, unstratified analysis**



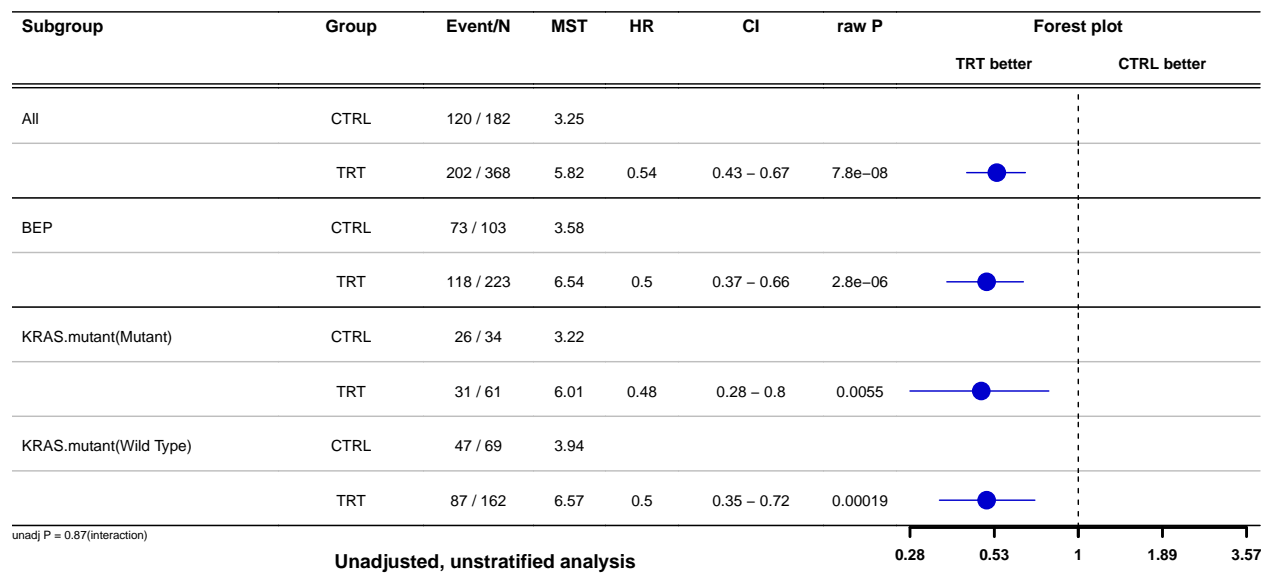
```
##
## code.v
## [1,] "Subgroup"      "Group" "Event/N" "MST" "HR"
## [2,] "All"           "CTRL"  "120 / 182" "3.25" ""
## [3,] ""              "TRT"   "202 / 368" "5.82" "0.54"
## [4,] "BEP"          "CTRL"  "73 / 103" "3.58" ""
## [5,] ""              "TRT"   "118 / 223" "6.54" "0.5"
## [6,] "KRAS.mutant(Mutant)" "CTRL" "26 / 34" "3.22" ""
## [7,] ""              "TRT"   "31 / 61" "6.01" "0.48"
## [8,] "KRAS.mutant(Wild Type)" "CTRL" "47 / 69" "3.94" ""
## [9,] ""              "TRT"   "87 / 162" "6.57" "0.5"
##
## [1,] "CI"           "raw P"
## [2,] ""              ""
## [3,] "0.43 - 0.67" "7.8e-08"
## [4,] ""              ""
## [5,] "0.37 - 0.66" "2.8e-06"
## [6,] ""              ""
## [7,] "0.28 - 0.8"  "0.0055"
## [8,] ""              ""
## [9,] "0.35 - 0.72" "0.00019"
```

## 7.2 Survival outcome, another format

```
PlotTabForestBiomarker(data=input,
                        outcome.class="survival",
                        outcome.var=c("PFS","PFS.event"),
                        trt="Arm",
                        var="KRAS.mutant",
                        var.class="categorical",
                        tabforest = TRUE)
```

## Some NAs in var column, will define the non NA entries as BEP

### Across-arm Effect of Biomarker PFS, KRAS.mutant



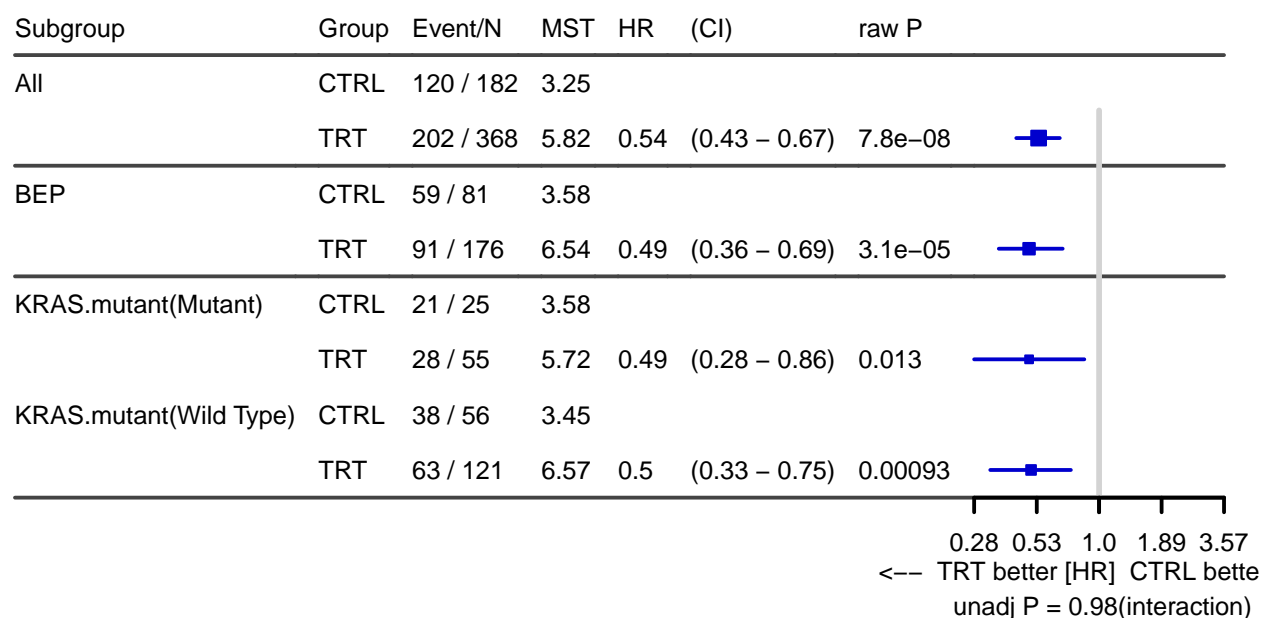
```
## code.v
## [1,] "Subgroup"      "Group" "Event/N" "MST" "HR"
## [2,] "All"          "CTRL"  "120 / 182" "3.25" ""
## [3,] ""             "TRT"   "202 / 368" "5.82" "0.54"
## [4,] "BEP"         "CTRL"  "73 / 103" "3.58" ""
## [5,] ""             "TRT"   "118 / 223" "6.54" "0.5"
## [6,] "KRAS.mutant(Mutant)" "CTRL"  "26 / 34" "3.22" ""
## [7,] ""             "TRT"   "31 / 61" "6.01" "0.48"
## [8,] "KRAS.mutant(Wild Type)" "CTRL"  "47 / 69" "3.94" ""
## [9,] ""             "TRT"   "87 / 162" "6.57" "0.5"
##
## [1,] "CI"           "raw P"
## [2,] ""             ""
## [3,] "0.43 - 0.67" "7.8e-08"
## [4,] ""             ""
## [5,] "0.37 - 0.66" "2.8e-06"
## [6,] ""             ""
## [7,] "0.28 - 0.8"  "0.0055"
## [8,] ""             ""
## [9,] "0.35 - 0.72" "0.00019"
```

### 7.3 Survival outcome, 2-arm, categorical variable, don't show ITT, BEP, re-name variable name in display

user can also define BEP column

```
PlotTabForestBiomarker(data=input,
                        outcome.class="survival",
                        outcome.var=c("PFS", "PFS.event"),
                        trt="Arm",
                        var="KRAS.mutant",
                        var.class="categorical",
                        bep = 'BEP',
                        bep.indicator=1)
```

#### Across-arm Effect of Biomarker PFS, KRAS.mutant Unadjusted, unstratified analysis



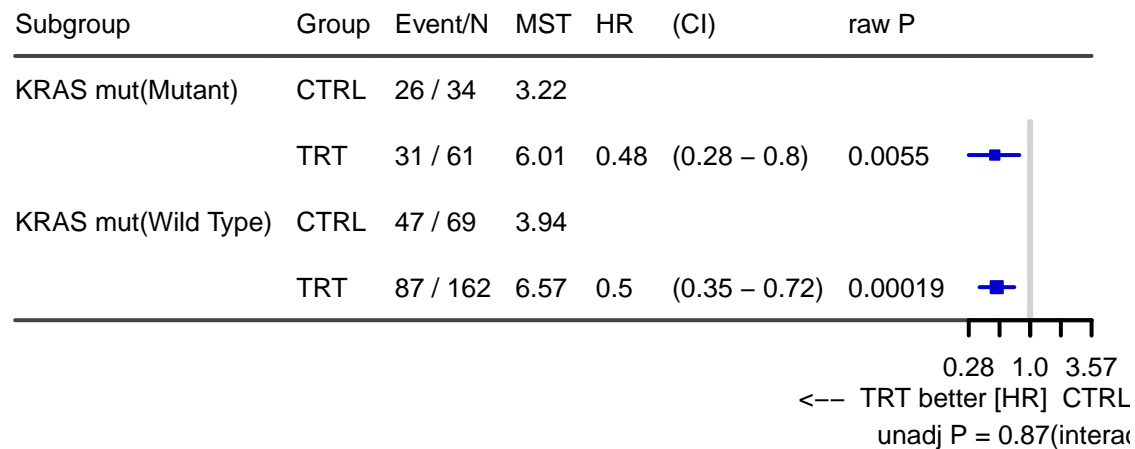
```
##                                     code.v
## [1,] "Subgroup"                    "Group" "Event/N"  "MST"  "HR"
## [2,] "All"                        "CTRL"  "120 / 182" "3.25" ""
## [3,] ""                            "TRT"   "202 / 368" "5.82" "0.54"
## [4,] "BEP"                        "CTRL"  "59 / 81"  "3.58" ""
## [5,] ""                            "TRT"   "91 / 176"  "6.54" "0.49"
## [6,] "KRAS.mutant(Mutant)"        "CTRL"  "21 / 25"  "3.58" ""
## [7,] ""                            "TRT"   "28 / 55"  "5.72" "0.49"
## [8,] "KRAS.mutant(Wild Type)"     "CTRL"  "38 / 56"  "3.45" ""
## [9,] ""                            "TRT"   "63 / 121" "6.57" "0.5"
##
## [1,] "CI"                          "raw P"
## [2,] ""                            ""
## [3,] "0.43 - 0.67"                "7.8e-08"
```

```
## [4,] "" ""
## [5,] "0.36 - 0.69" "3.1e-05"
## [6,] "" ""
## [7,] "0.28 - 0.86" "0.013"
## [8,] "" ""
## [9,] "0.33 - 0.75" "0.00093"
```

```
PlotTabForestBiomarker(data=input,
                        outcome.class="survival",
                        outcome.var=c("PFS","PFS.event"),
                        trt="Arm",
                        var="KRAS.mutant",
                        var.class="categorical",
                        var.name="KRAS mut",
                        show.itt=FALSE,
                        show.bep=FALSE)
```

## Some NAs in var column, will define the non NA entries as BEP

### Across-arm Effect of Biomarker PFS, KRAS mut Unadjusted, unstratified analysis



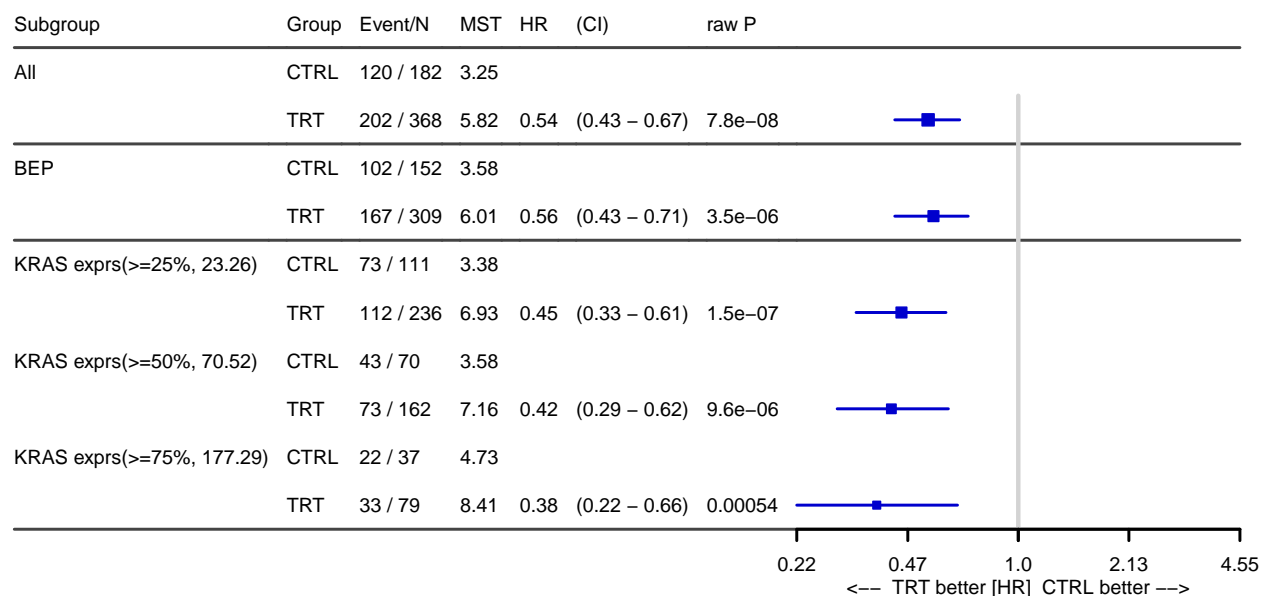
```
## code.v
## [1,] "Subgroup" "Group" "Event/N" "MST" "HR" "CI"
## [2,] "KRAS mut(Mutant)" "CTRL" "26 / 34" "3.22" "" ""
## [3,] "" "TRT" "31 / 61" "6.01" "0.48" "0.28 - 0.8"
## [4,] "KRAS mut(Wild Type)" "CTRL" "47 / 69" "3.94" "" ""
## [5,] "" "TRT" "87 / 162" "6.57" "0.5" "0.35 - 0.72"
##
## [1,] "raw P"
## [2,] ""
## [3,] "0.0055"
## [4,] ""
## [5,] "0.00019"
```

## 7.4 Survival outcome, 2-arm, continuous variable, greater than some percentage cutoffs

```
PlotTabForestBiomarker(data=input,
                        outcome.class="survival",
                        outcome.var=c("PFS", "PFS.event"),
                        trt="Arm",
                        var="KRAS.exprs",
                        var.class="numeric", var.name="KRAS exprs",
                        percentile.cutoff=c(.25,.5,.75),
                        # cols=c("black", "black", "darkgreen", "darkgreen", "darkgreen"),
                        numerical.cutoff=NULL,
                        greater=TRUE, less=FALSE,
                        within.bin=FALSE,
                        show.itt=TRUE, show.bep=TRUE)
```

## Some NAs in var column, will define the non NA entries as BEP

### Across-arm Effect of Biomarker PFS, KRAS exprs Unadjusted, unstratified analysis



```
##
## [1,] "Subgroup"
## [2,] "All"
## [3,] ""
## [4,] "BEP"
## [5,] ""
## [6,] "KRAS exprs(>=25%, 23.26)"
## [7,] ""
## [8,] "KRAS exprs(>=50%, 70.52)"
## [9,] ""

code.v
"Group" "Event/N" "MST" "HR"
"CTRL" "120 / 182" "3.25" ""
"TRT" "202 / 368" "5.82" "0.54"
"CTRL" "102 / 152" "3.58" ""
"TRT" "167 / 309" "6.01" "0.56"
"CTRL" "73 / 111" "3.38" ""
"TRT" "112 / 236" "6.93" "0.45"
"CTRL" "43 / 70" "3.58" ""
"TRT" "73 / 162" "7.16" "0.42"
```

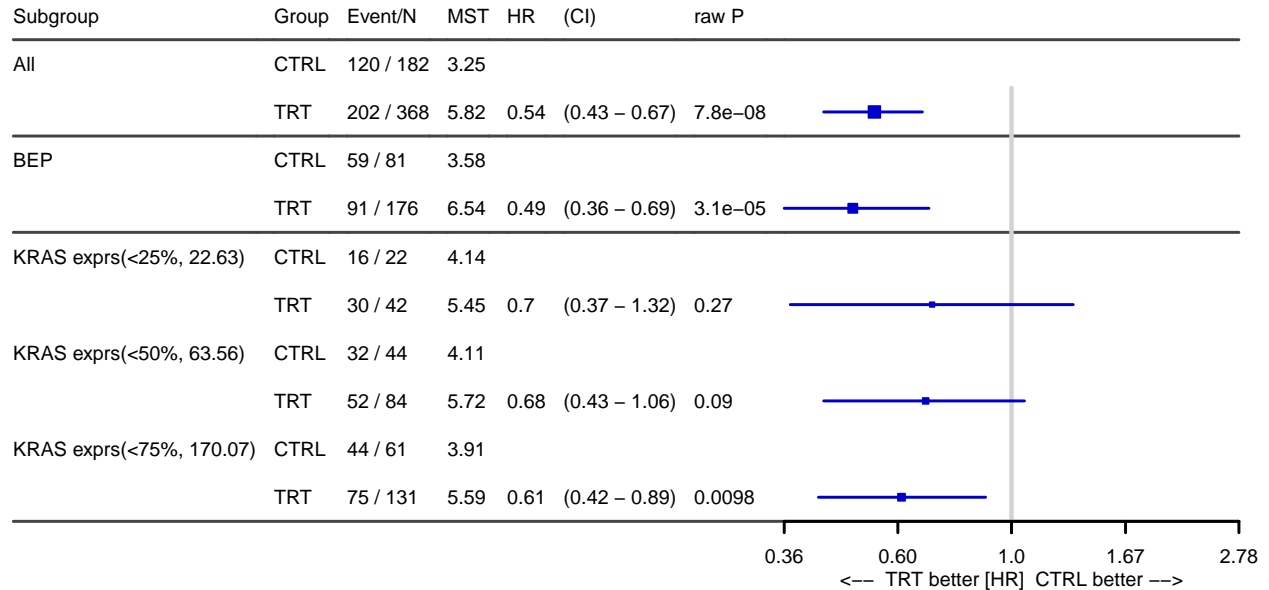
```
## [10,] "KRAS exprs(>=75%, 177.29)" "CTRL" "22 / 37" "4.73" ""
## [11,] "" "TRT" "33 / 79" "8.41" "0.38"
##
## [1,] "CI" "raw P"
## [2,] "" ""
## [3,] "0.43 - 0.67" "7.8e-08"
## [4,] "" ""
## [5,] "0.43 - 0.71" "3.5e-06"
## [6,] "" ""
## [7,] "0.33 - 0.61" "1.5e-07"
## [8,] "" ""
## [9,] "0.29 - 0.62" "9.6e-06"
## [10,] "" ""
## [11,] "0.22 - 0.66" "0.00054"
```

## 7.5 Survival outcome, 2-arm, continuous variable, less than some percentage cutoffs

```
PlotTabForestBiomarker(data=input,
                        outcome.class="survival",
                        outcome.var=c("PFS", "PFS.event"),
                        trt="Arm",
                        var="KRAS.exprs",
                        var.class="numeric", var.name="KRAS exprs",
                        percentile.cutoff=c(.25,.5,.75),
                        numerical.cutoff=NULL,
                        greater=FALSE, less=TRUE,
                        within.bin=FALSE,
                        show.itt=TRUE, show.bep=TRUE,
                        bep = 'BEP', bep.indicator=1)
```



**Across-arm Effect of Biomarker**  
**PFS, KRAS exprs**  
**Unadjusted, unstratified analysis**



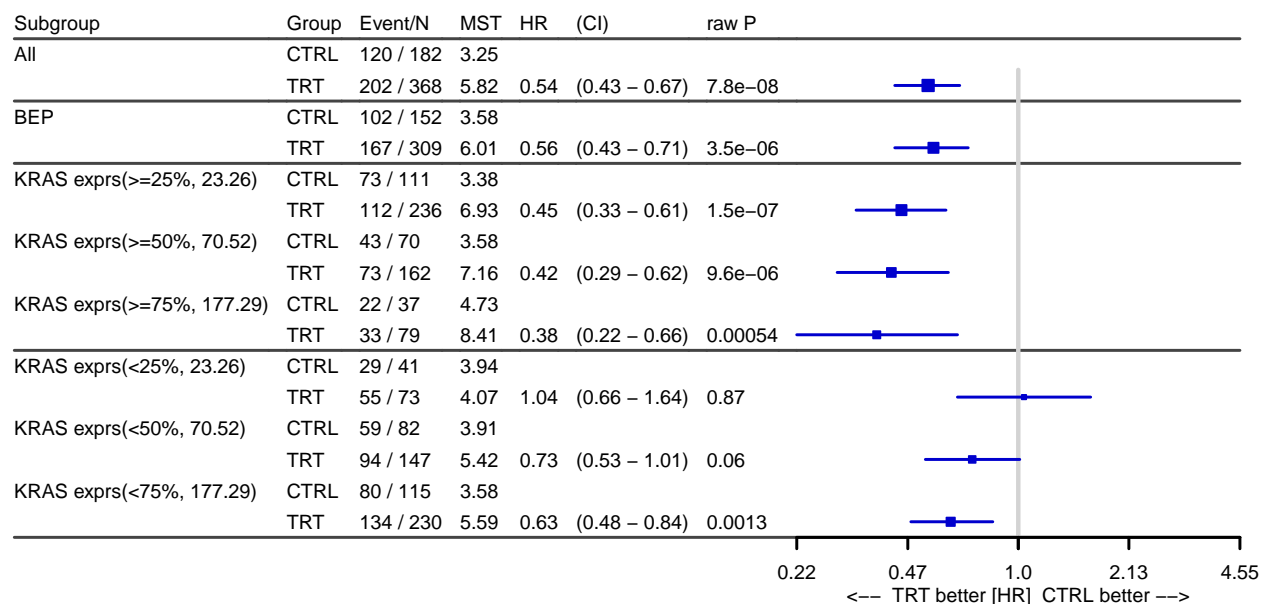
```
##
## [1,] "Subgroup"      code.v
## [2,] "All"           "Group" "Event/N"  "MST"  "HR"
## [3,] ""              "CTRL"  "120 / 182" "3.25" ""
## [4,] "BEP"           "TRT"   "202 / 368" "5.82" "0.54"
## [5,] ""              "CTRL"  "59 / 81"  "3.58" ""
## [6,] "KRAS exprs(<25%, 22.63)" "TRT"   "91 / 176" "6.54" "0.49"
## [7,] ""              "CTRL"  "16 / 22"  "4.14" ""
## [8,] "KRAS exprs(<50%, 63.56)" "TRT"   "30 / 42"  "5.45" "0.7"
## [9,] ""              "CTRL"  "32 / 44"  "4.11" ""
## [10,] "KRAS exprs(<75%, 170.07)" "TRT"   "52 / 84"  "5.72" "0.68"
## [11,] ""             "CTRL"  "44 / 61"  "3.91" ""
## [12,] ""             "TRT"   "75 / 131" "5.59" "0.61"
##
## [1,] "CI"           "raw P"
## [2,] ""             ""
## [3,] "0.43 - 0.67"  "7.8e-08"
## [4,] ""             ""
## [5,] "0.36 - 0.69"  "3.1e-05"
## [6,] ""             ""
## [7,] "0.37 - 1.32"  "0.27"
## [8,] ""             ""
## [9,] "0.43 - 1.06"  "0.09"
## [10,] ""            ""
## [11,] "0.42 - 0.89" "0.0098"
```

## 7.6 Survival outcome, 2-arm, continuous variable, greater and less than some percentage cutoffs

```
PlotTabForestBiomarker(data=input,
                        outcome.class="survival",
                        outcome.var=c("PFS", "PFS.event"),
                        trt="Arm",
                        var="KRAS.exprs",
                        var.class="numeric", var.name="KRAS exprs",
                        percentile.cutoff=c(.25,.5,.75),
                        numerical.cutoff=NULL,
                        greater=TRUE, less=TRUE,
                        within.bin=FALSE,
                        show.itt=TRUE, show.bep=TRUE)
```

## Some NAs in var column, will define the non NA entries as BEP

### Across-arm Effect of Biomarker PFS, KRAS exprs Unadjusted, unstratified analysis



```
##
## [1,] "Subgroup"
## [2,] "All"
## [3,] ""
## [4,] "BEP"
## [5,] ""
## [6,] "KRAS exprs(>=25%, 23.26)"
## [7,] ""
## [8,] "KRAS exprs(>=50%, 70.52)"
## [9,] ""
## [10,] "KRAS exprs(>=75%, 177.29)"

code.v
"Group" "Event/N" "MST" "HR"
"CTRL" "120 / 182" "3.25" ""
"TRT" "202 / 368" "5.82" "0.54"
"CTRL" "102 / 152" "3.58" ""
"TRT" "167 / 309" "6.01" "0.56"
"CTRL" "73 / 111" "3.38" ""
"TRT" "112 / 236" "6.93" "0.45"
"CTRL" "43 / 70" "3.58" ""
"TRT" "73 / 162" "7.16" "0.42"
"CTRL" "22 / 37" "4.73" ""
"TRT" "33 / 79" "8.41" "0.38"
```

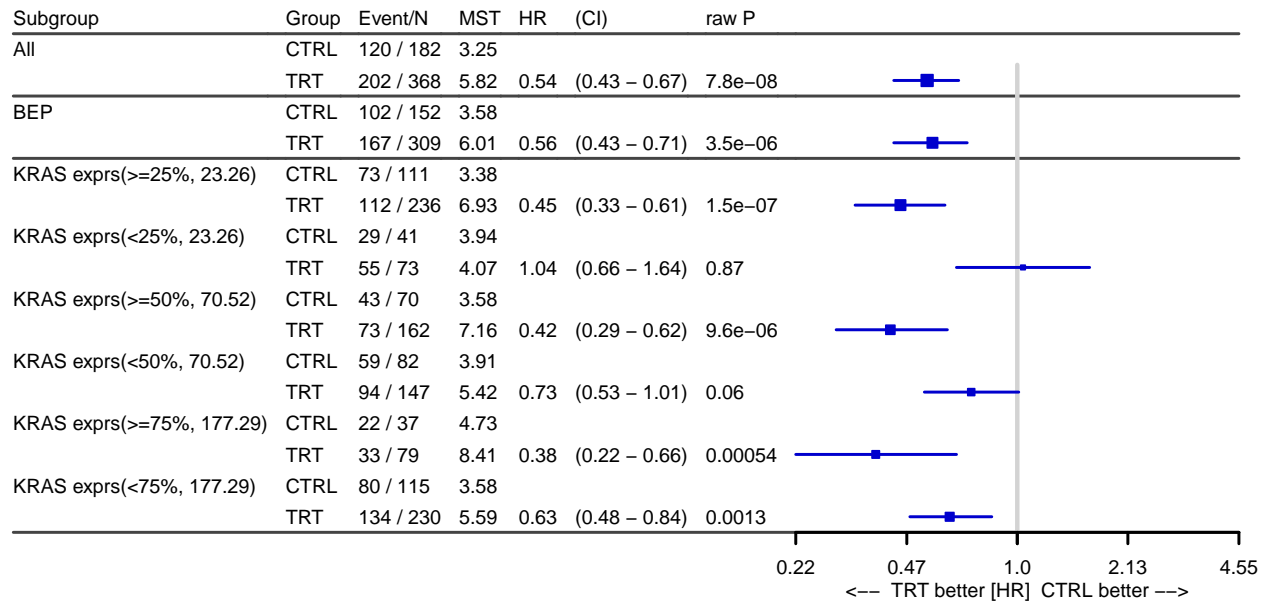
```
## [11,] "" "TRT" "33 / 79" "8.41" "0.38"
## [12,] "KRAS exprs(<25%, 23.26)" "CTRL" "29 / 41" "3.94" ""
## [13,] "" "TRT" "55 / 73" "4.07" "1.04"
## [14,] "KRAS exprs(<50%, 70.52)" "CTRL" "59 / 82" "3.91" ""
## [15,] "" "TRT" "94 / 147" "5.42" "0.73"
## [16,] "KRAS exprs(<75%, 177.29)" "CTRL" "80 / 115" "3.58" ""
## [17,] "" "TRT" "134 / 230" "5.59" "0.63"
##
## [1,] "CI" "raw P"
## [2,] "" ""
## [3,] "0.43 - 0.67" "7.8e-08"
## [4,] "" ""
## [5,] "0.43 - 0.71" "3.5e-06"
## [6,] "" ""
## [7,] "0.33 - 0.61" "1.5e-07"
## [8,] "" ""
## [9,] "0.29 - 0.62" "9.6e-06"
## [10,] "" ""
## [11,] "0.22 - 0.66" "0.00054"
## [12,] "" ""
## [13,] "0.66 - 1.64" "0.87"
## [14,] "" ""
## [15,] "0.53 - 1.01" "0.06"
## [16,] "" ""
## [17,] "0.48 - 0.84" "0.0013"
```

“Less” rows next to “Greater” rows:

```
PlotTabForestBiomarker(data=input,
                        outcome.class="survival",
                        outcome.var=c("PFS", "PFS.event"),
                        trt="Arm",
                        var="KRAS.exprs",
                        var.class="numeric", var.name="KRAS exprs",
                        percentile.cutoff=c(.25,.5,.75),
                        numerical.cutoff=NULL,
                        greater=TRUE, less=TRUE, greater.by.less = TRUE,
                        within.bin=FALSE,
                        show.itt=TRUE, show.bep=TRUE)
```

```
## Some NAs in var column, will define the non NA entries as BEP
```

**Across-arm Effect of Biomarker**  
**PFS, KRAS exprs**  
**Unadjusted, unstratified analysis**



```
##
## [1,] "Subgroup"
## [2,] "All"
## [3,] ""
## [4,] "BEP"
## [5,] ""
## [6,] "KRAS exprs(>=25%, 23.26)"
## [7,] ""
## [8,] "KRAS exprs(<25%, 23.26)"
## [9,] ""
## [10,] "KRAS exprs(>=50%, 70.52)"
## [11,] ""
## [12,] "KRAS exprs(<50%, 70.52)"
## [13,] ""
## [14,] "KRAS exprs(>=75%, 177.29)"
## [15,] ""
## [16,] "KRAS exprs(<75%, 177.29)"
## [17,] ""
##
## [1,] "CI"
## [2,] ""
## [3,] "0.43 - 0.67"
## [4,] ""
## [5,] "0.43 - 0.71"
## [6,] ""
## [7,] "0.33 - 0.61"
## [8,] ""
## [9,] "0.66 - 1.64"

code.v
"Group" "Event/N" "MST" "HR"
"CTRL" "120 / 182" "3.25" ""
"TRT" "202 / 368" "5.82" "0.54"
"CTRL" "102 / 152" "3.58" ""
"TRT" "167 / 309" "6.01" "0.56"
"CTRL" "73 / 111" "3.38" ""
"TRT" "112 / 236" "6.93" "0.45"
"CTRL" "29 / 41" "3.94" ""
"TRT" "55 / 73" "4.07" "1.04"
"CTRL" "43 / 70" "3.58" ""
"TRT" "73 / 162" "7.16" "0.42"
"CTRL" "59 / 82" "3.91" ""
"TRT" "94 / 147" "5.42" "0.73"
"CTRL" "22 / 37" "4.73" ""
"TRT" "33 / 79" "8.41" "0.38"
"CTRL" "80 / 115" "3.58" ""
"TRT" "134 / 230" "5.59" "0.63"
"raw P"
""
"7.8e-08"
""
"3.5e-06"
""
"1.5e-07"
""
"0.87"
```

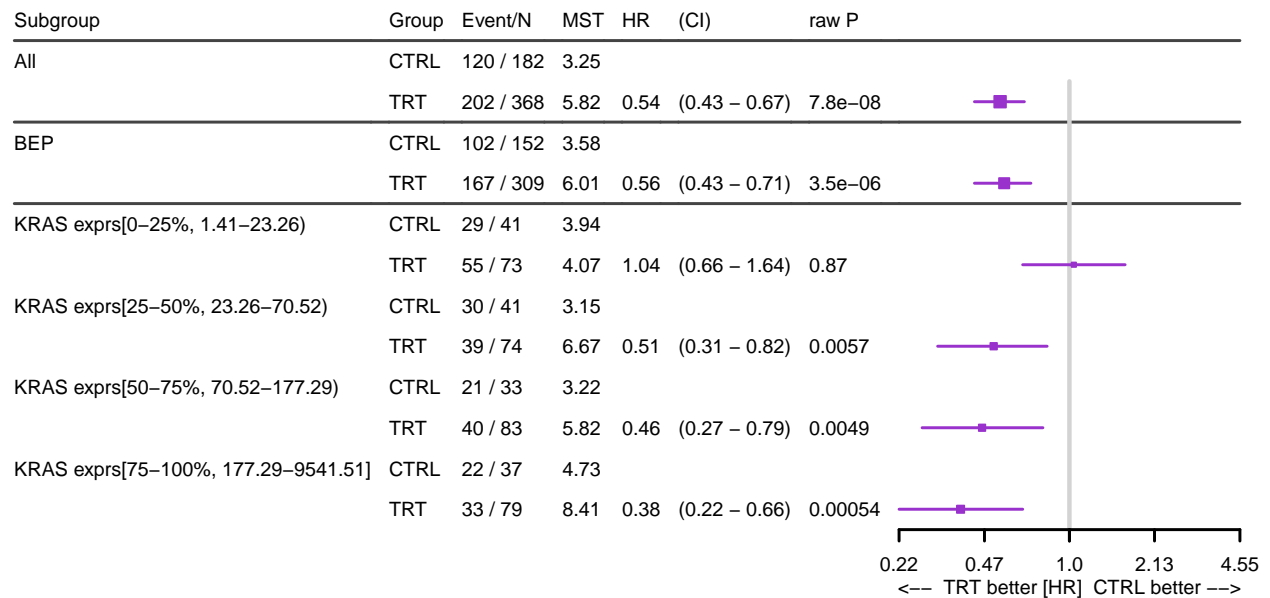
```
## [10,] "" ""
## [11,] "0.29 - 0.62" "9.6e-06"
## [12,] "" ""
## [13,] "0.53 - 1.01" "0.06"
## [14,] "" ""
## [15,] "0.22 - 0.66" "0.00054"
## [16,] "" ""
## [17,] "0.48 - 0.84" "0.0013"
```

## 7.7 Survival outcome, 2-arm, continuous variable, within cutoff bin

```
PlotTabForestBiomarker(data=input,
                        outcome.class="survival",
                        outcome.var=c("PFS","PFS.event"),
                        trt="Arm",
                        var="KRAS.exprs",
                        var.class="numeric", var.name="KRAS exprs",
                        percentile.cutoff=c(.25,.5,.75),
                        numerical.cutoff=NULL,
                        greater=FALSE, less=FALSE,
                        within.bin=TRUE,
                        show.itt=TRUE, show.bep=TRUE)
```

## Some NAs in var column, will define the non NA entries as BEP

### Across-arm Effect of Biomarker PFS, KRAS exprs Unadjusted, unstratified analysis



```
##
## [1,] "Subgroup"
## [2,] "All"

code.v
"Group" "Event/N" "MST"
"CTRL" "120 / 182" "3.25"
```

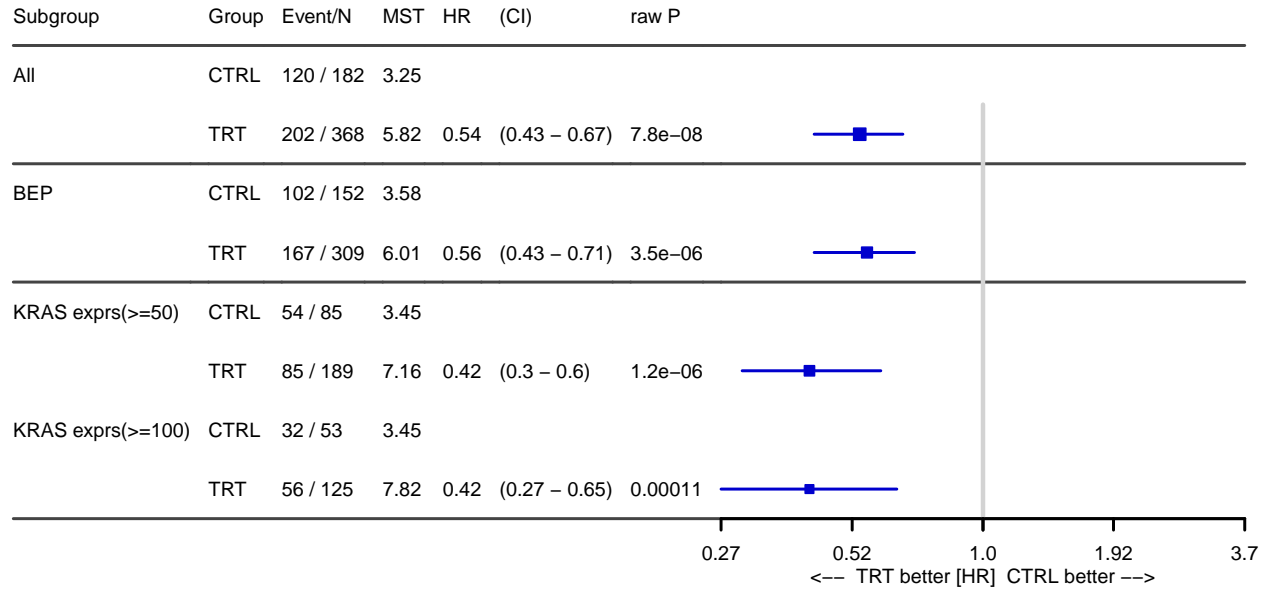
```
## [3,] "" "TRT" "202 / 368" "5.82"
## [4,] "BEP" "CTRL" "102 / 152" "3.58"
## [5,] "" "TRT" "167 / 309" "6.01"
## [6,] "KRAS exprs[0-25%, 1.41-23.26)" "CTRL" "29 / 41" "3.94"
## [7,] "" "TRT" "55 / 73" "4.07"
## [8,] "KRAS exprs[25-50%, 23.26-70.52)" "CTRL" "30 / 41" "3.15"
## [9,] "" "TRT" "39 / 74" "6.67"
## [10,] "KRAS exprs[50-75%, 70.52-177.29)" "CTRL" "21 / 33" "3.22"
## [11,] "" "TRT" "40 / 83" "5.82"
## [12,] "KRAS exprs[75-100%, 177.29-9541.51]" "CTRL" "22 / 37" "4.73"
## [13,] "" "TRT" "33 / 79" "8.41"
##
## [1,] "HR" "CI" "raw P"
## [2,] "" "" ""
## [3,] "0.54" "0.43 - 0.67" "7.8e-08"
## [4,] "" "" ""
## [5,] "0.56" "0.43 - 0.71" "3.5e-06"
## [6,] "" "" ""
## [7,] "1.04" "0.66 - 1.64" "0.87"
## [8,] "" "" ""
## [9,] "0.51" "0.31 - 0.82" "0.0057"
## [10,] "" "" ""
## [11,] "0.46" "0.27 - 0.79" "0.0049"
## [12,] "" "" ""
## [13,] "0.38" "0.22 - 0.66" "0.00054"
```

## 7.8 Survival outcome, 2-arm, continuous variable, greater than some numerical cutoffs

```
PlotTabForestBiomarker(data=input,
                        outcome.class="survival",
                        outcome.var=c("PFS","PFS.event"),
                        trt="Arm",
                        var="KRAS.exprs",
                        var.class="numeric", var.name="KRAS exprs",
                        percentile.cutoff=NULL,
                        numerical.cutoff=c(50,100),
                        greater=TRUE, less=FALSE,
                        within.bin=FALSE,
                        show.itt=TRUE, show.bep=TRUE)
```

```
## Some NAs in var column, will define the non NA entries as BEP
```

**Across-arm Effect of Biomarker**  
**PFS, KRAS exprs**  
**Unadjusted, unstratified analysis**



```
##                               code.v
## [1,] "Subgroup"              "Group" "Event/N"  "MST"  "HR"  "CI"
## [2,] "All"                   "CTRL" "120 / 182" "3.25" ""    ""
## [3,] ""                      "TRT"  "202 / 368" "5.82" "0.54" "0.43 - 0.67"
## [4,] "BEP"                   "CTRL" "102 / 152" "3.58" ""    ""
## [5,] ""                      "TRT"  "167 / 309" "6.01" "0.56" "0.43 - 0.71"
## [6,] "KRAS exprs(>=50)"      "CTRL" "54 / 85"  "3.45" ""    ""
## [7,] ""                      "TRT"  "85 / 189"  "7.16" "0.42" "0.3 - 0.6"
## [8,] "KRAS exprs(>=100)"    "CTRL" "32 / 53"  "3.45" ""    ""
## [9,] ""                      "TRT"  "56 / 125"  "7.82" "0.42" "0.27 - 0.65"
##
## [1,] "raw P"
## [2,] ""
## [3,] "7.8e-08"
## [4,] ""
## [5,] "3.5e-06"
## [6,] ""
## [7,] "1.2e-06"
## [8,] ""
## [9,] "0.00011"
```

## 7.9 Survival outcome, 2-arm, continuous variable, adjust for covariates

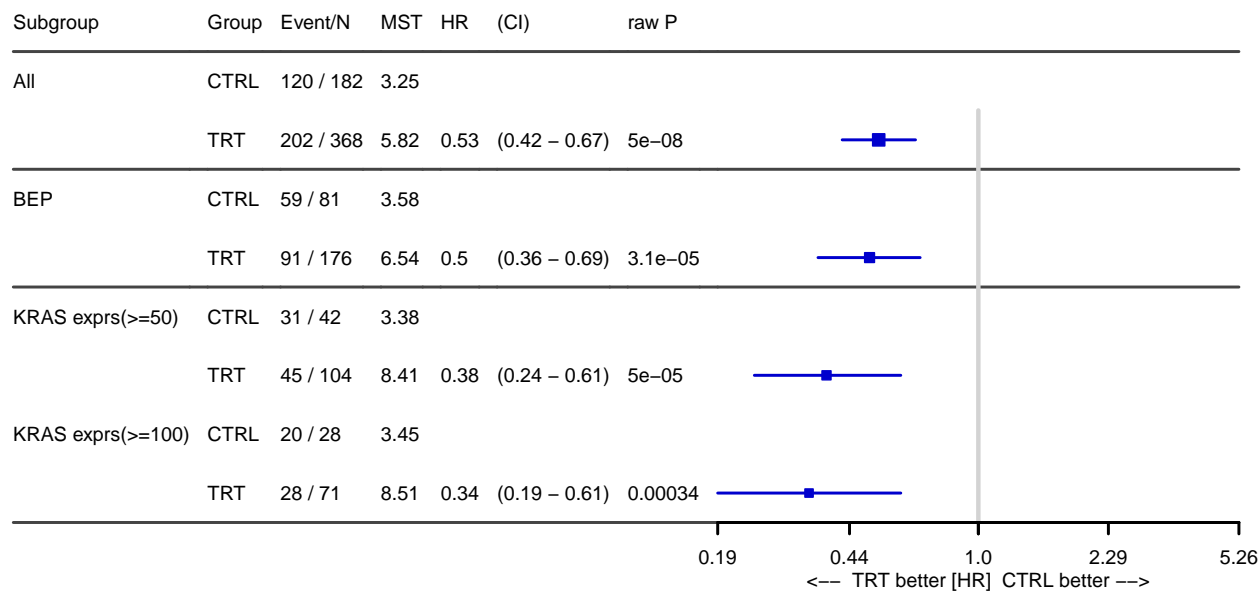
```
PlotTabForestBiomarker(data=input,
                        outcome.class="survival",
                        outcome.var=c("PFS", "PFS.event"),
                        trt="Arm",
```

```

var="KRAS.exprs",
var.class="numeric", var.name="KRAS exprs",
percentile.cutoff=NULL,
numerical.cutoff=c(50,100),
greater=TRUE, less=FALSE,
within.bin=FALSE,
show.itt=TRUE, show.bep=TRUE,
bep = 'BEP', bep.indicator=1, covariate="Age")

```

**Across-arm Effect of Biomarker**  
**PFS, KRAS exprs**  
**Results adjusted by Age;**



```

##
##      code.v
## [1,] "Subgroup"      "Group" "Event/N"  "MST"  "HR"  "CI"
## [2,] "All"           "CTRL" "120 / 182" "3.25" ""   ""
## [3,] ""              "TRT"  "202 / 368" "5.82" "0.53" "0.42 - 0.67"
## [4,] "BEP"          "CTRL" "59 / 81"  "3.58" ""   ""
## [5,] ""              "TRT"  "91 / 176" "6.54" "0.5"  "0.36 - 0.69"
## [6,] "KRAS exprs(>=50)" "CTRL" "31 / 42"  "3.38" ""   ""
## [7,] ""              "TRT"  "45 / 104" "8.41" "0.38" "0.24 - 0.61"
## [8,] "KRAS exprs(>=100)" "CTRL" "20 / 28"  "3.45" ""   ""
## [9,] ""              "TRT"  "28 / 71"  "8.51" "0.34" "0.19 - 0.61"
##
## [1,] "raw P"
## [2,] ""
## [3,] "5e-08"
## [4,] ""
## [5,] "3.1e-05"
## [6,] ""
## [7,] "5e-05"
## [8,] ""

```

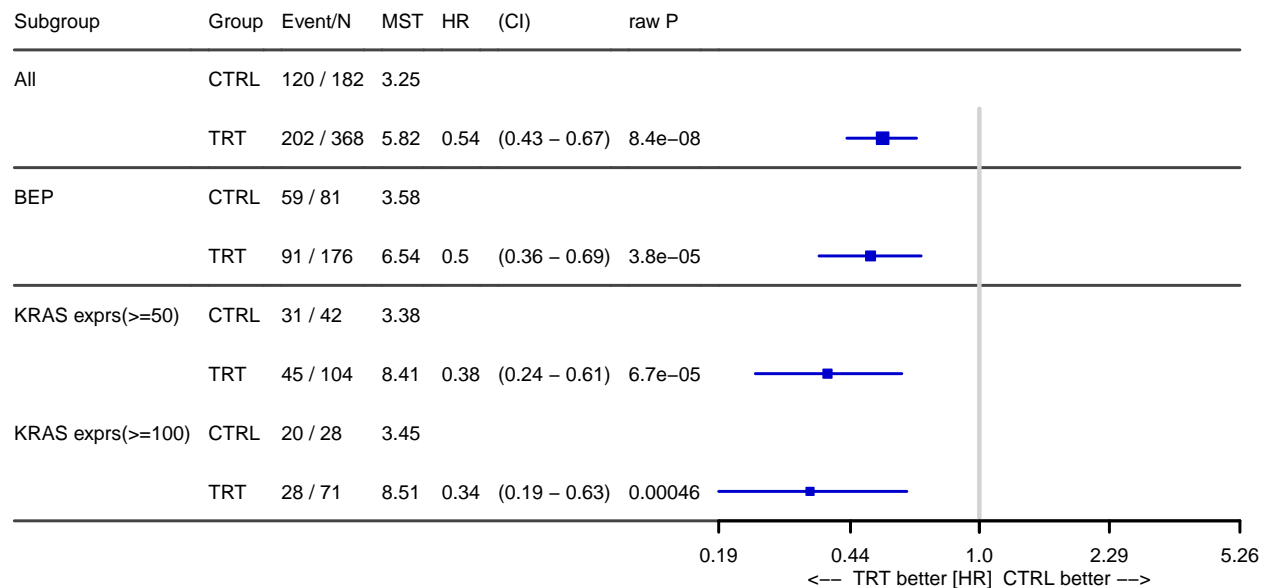


```
## [9,] "0.00034"
```

## 7.10 Survival outcome, 2-arm, continuous variable, stratification

```
PlotTabForestBiomarker(data=input,
                        outcome.class="survival",
                        outcome.var=c("PFS","PFS.event"),
                        trt="Arm",
                        var="KRAS.exprs",
                        var.class="numeric", var.name="KRAS exprs",
                        percentile.cutoff=NULL,
                        numerical.cutoff=c(50,100),
                        greater=TRUE, less=FALSE,
                        within.bin=FALSE,
                        show.itt=TRUE, show.bep=TRUE,
                        bep = 'BEP', bep.indicator=1, strata="Sex")
```

**Across-arm Effect of Biomarker**  
**PFS, KRAS exprs**  
**;Results stratified by Sex**



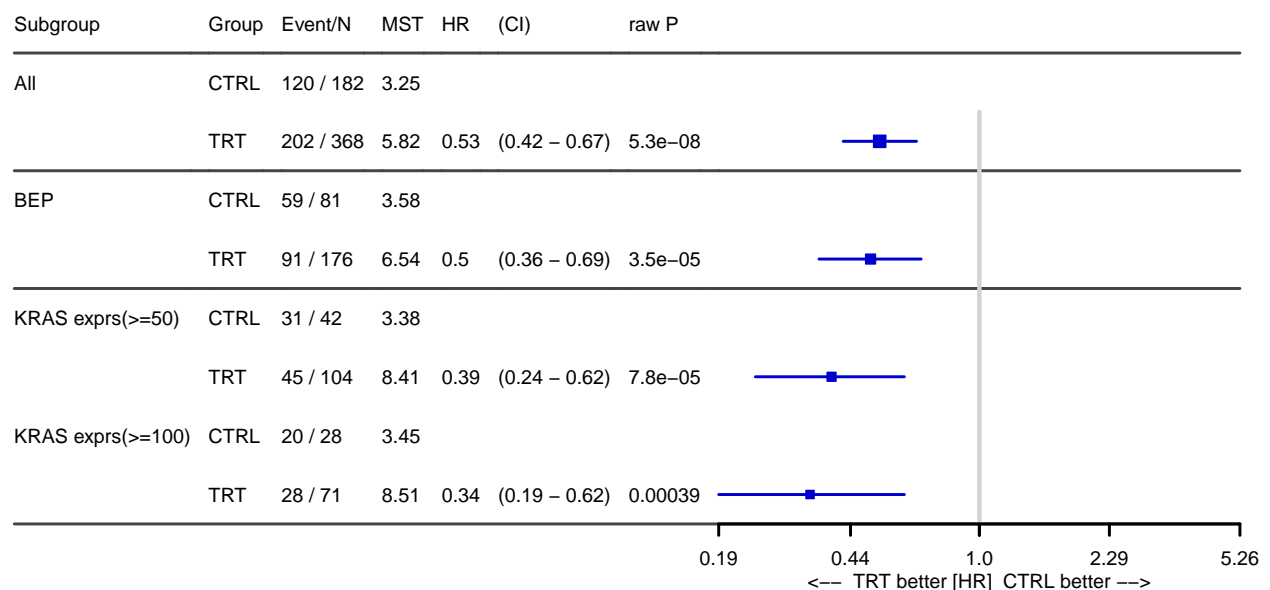
```
##
## [1,] "Subgroup"      code.v
## [2,] "All"          "CTRL" "120 / 182" "3.25" "" ""
## [3,] ""              "TRT"  "202 / 368" "5.82" "0.54" "0.43 - 0.67"
## [4,] "BEP"           "CTRL" "59 / 81"  "3.58" "" ""
## [5,] ""              "TRT"  "91 / 176" "6.54" "0.5"  "0.36 - 0.69"
## [6,] "KRAS exprs(>=50)" "CTRL" "31 / 42"  "3.38" "" ""
## [7,] ""              "TRT"  "45 / 104" "8.41" "0.38" "0.24 - 0.61"
## [8,] "KRAS exprs(>=100)" "CTRL" "20 / 28"  "3.45" "" ""
## [9,] ""              "TRT"  "28 / 71"  "8.51" "0.34" "0.19 - 0.63"
```

```
##
## [1,] "raw P"
## [2,] ""
## [3,] "8.4e-08"
## [4,] ""
## [5,] "3.8e-05"
## [6,] ""
## [7,] "6.7e-05"
## [8,] ""
## [9,] "0.00046"
```

## 7.11 Survival outcome, 2-arm, continuous variable, adjust for covariates and stratified by strata

```
PlotTabForestBiomarker(data=input,
                        outcome.class="survival",
                        outcome.var=c("PFS","PFS.event"),
                        trt="Arm",
                        var="KRAS.exprs",
                        var.class="numeric", var.name="KRAS exprs",
                        percentile.cutoff=NULL,
                        numerical.cutoff=c(50,100),
                        greater=TRUE, less=FALSE,
                        within.bin=FALSE,
                        show.itt=TRUE, show.bep=TRUE,
                        bep = 'BEP', bep.indicator=1,covariate="Age",strata="Sex")
```

### Across-arm Effect of Biomarker PFS, KRAS exprs Results adjusted by Age;Results stratified by Sex



```
##
## [1,] "Subgroup"      code.v
## [2,] "All"          "Group" "Event/N"  "MST"  "HR"  "CI"
## [3,] ""             "CTRL"  "120 / 182" "3.25" ""   ""
## [4,] "BEP"         "TRT"   "202 / 368" "5.82" "0.53" "0.42 - 0.67"
## [5,] ""            "CTRL"  "59 / 81"   "3.58" ""   ""
## [6,] "KRAS exprs(>=50)" "TRT"   "91 / 176"  "6.54" "0.5"  "0.36 - 0.69"
## [7,] ""            "CTRL"  "31 / 42"   "3.38" ""   ""
## [8,] "KRAS exprs(>=100)" "TRT"   "45 / 104"  "8.41" "0.39" "0.24 - 0.62"
## [9,] ""            "CTRL"  "20 / 28"   "3.45" ""   ""
## [9,] ""            "TRT"   "28 / 71"   "8.51" "0.34" "0.19 - 0.62"
##
## [1,] "raw P"
## [2,] ""
## [3,] "5.3e-08"
## [4,] ""
## [5,] "3.5e-05"
## [6,] ""
## [7,] "7.8e-05"
## [8,] ""
## [9,] "0.00039"
```

## 7.12 Survival outcome, 1arm

within.bin, show.itt, show.bep will be ignored

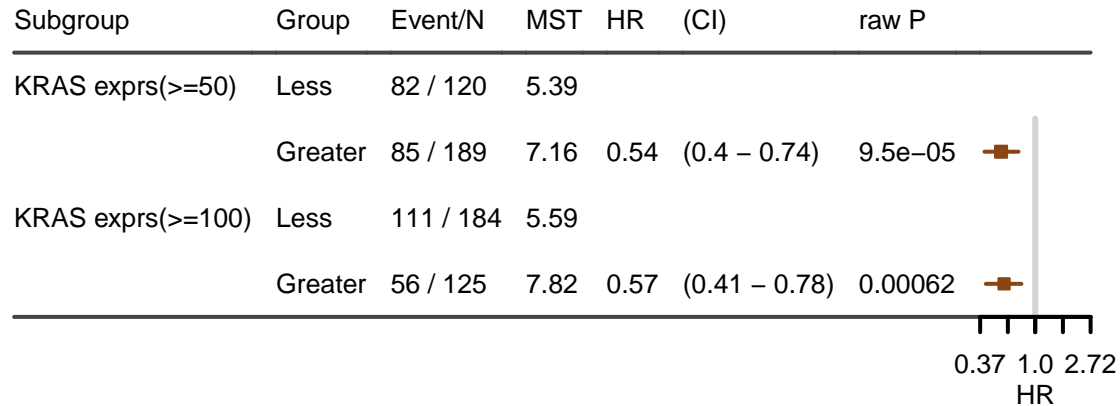
```
PlotTabForestBiomarker(data=subset(input, Arm=="TRT"),
                        outcome.class="survival",
                        outcome.var=c("PFS", "PFS.event"),
                        trt=NULL,
                        var="KRAS.exprs",
                        var.class="numeric", var.name="KRAS exprs",
                        percentile.cutoff=NULL,
                        numerical.cutoff=c(50,100),
                        greater=TRUE, less=FALSE,
                        within.bin=FALSE,
                        show.itt=TRUE, show.bep=TRUE, covariate="Age", strata="Sex")
```

```
## Some NAs in var column, will define the non NA entries as BEP
```

```
## only 1 arm; show.itt is set to FALSE
```

```
## only 1 arm; show.bep is set to FALSE
```

**Within-arm Effect of Biomarker**  
**PFS, KRAS exprs**  
**Results adjusted by Age; Results stratified by Sex**



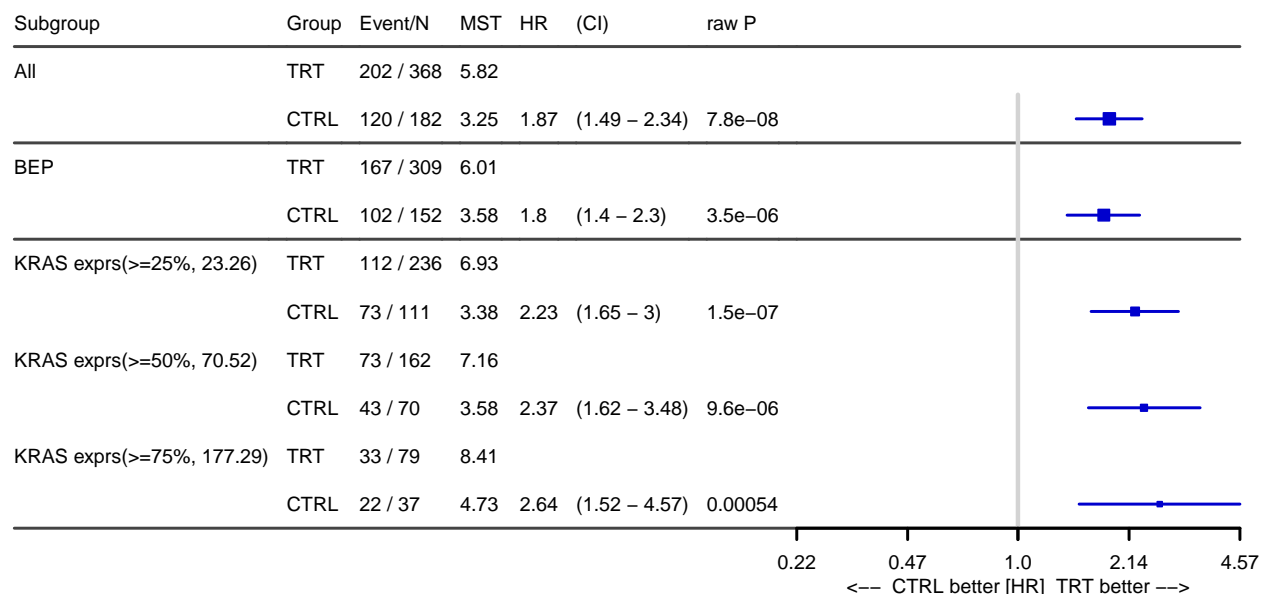
```
##
## [1,] "Subgroup"      code.v
## [2,] "KRAS exprs(>=50)" "Group"  "Event/N"  "MST"  "HR"  "CI"
## [3,] ""              "Greater" "85 / 189" "7.16" "0.54" "0.4 - 0.74"
## [4,] "KRAS exprs(>=100)" "Less"    "111 / 184" "5.59" ""
## [5,] ""              "Greater" "56 / 125" "7.82" "0.57" "0.41 - 0.78"
##
## [1,] "raw P"
## [2,] ""
## [3,] "9.5e-05"
## [4,] ""
## [5,] "0.00062"
```

### 7.13 Survival outcome, 2arm, flip TRT/CTRL order (calculate ctrl over trt HR)

```
PlotTabForestBiomarker(data=input,
  outcome.class="survival",
  outcome.var=c("PFS", "PFS.event"),
  trt="Arm",
  var="KRAS.exprs",
  var.class="numeric", var.name="KRAS exprs",
  percentile.cutoff=c(.25,.5,.75),
  numerical.cutoff=NULL,
  greater=TRUE, less=FALSE,
  within.bin=FALSE,
  show.itt=TRUE, show.bep=TRUE,
  covariate=NULL, #Sex
  strata=NULL, #Age
  placebo.code='TRT',
  active.code='CTRL')
```

## Some NAs in var column, will define the non NA entries as BEP

**Across-arm Effect of Biomarker**  
**PFS, KRAS exprs**  
**Unadjusted, unstratified analysis**



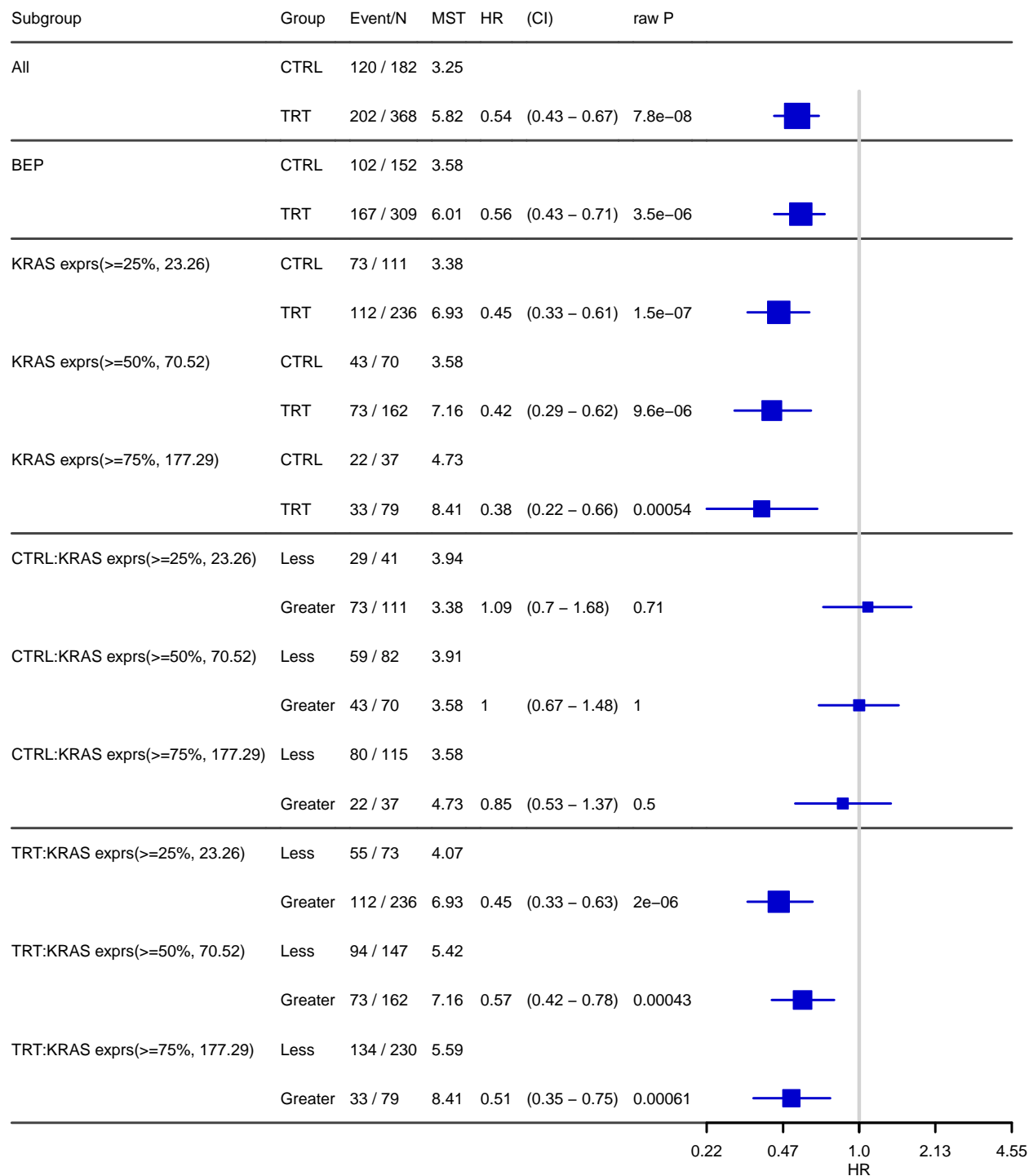
```
##
##      code.v
## [1,] "Subgroup"      "Group" "Event/N"  "MST"  "HR"
## [2,] "All"           "TRT"  "202 / 368" "5.82" ""
## [3,] ""              "CTRL" "120 / 182" "3.25" "1.87"
## [4,] "BEP"           "TRT"  "167 / 309" "6.01" ""
## [5,] ""              "CTRL" "102 / 152" "3.58" "1.8"
## [6,] "KRAS exprs(>=25%, 23.26)" "TRT"  "112 / 236" "6.93" ""
## [7,] ""              "CTRL" "73 / 111"  "3.38" "2.23"
## [8,] "KRAS exprs(>=50%, 70.52)" "TRT"  "73 / 162"  "7.16" ""
## [9,] ""              "CTRL" "43 / 70"   "3.58" "2.37"
## [10,] "KRAS exprs(>=75%, 177.29)" "TRT"  "33 / 79"   "8.41" ""
## [11,] ""             "CTRL" "22 / 37"   "4.73" "2.64"
##
## [1,] "CI"           "raw P"
## [2,] ""             ""
## [3,] "1.49 - 2.34"  "7.8e-08"
## [4,] ""             ""
## [5,] "1.4 - 2.3"    "3.5e-06"
## [6,] ""             ""
## [7,] "1.65 - 3"     "1.5e-07"
## [8,] ""             ""
## [9,] "1.62 - 3.48"  "9.6e-06"
## [10,] ""            ""
## [11,] "1.52 - 4.57" "0.00054"
```

### 7.14 Survival outcome, 2-arm, continuous variable, greater than some percentage cutoffs, show across-arm and within-arm results in the same plot

```
PlotTabForestBiomarker(data=input,
                        outcome.class="survival",
                        outcome.var=c("PFS", "PFS.event"),
                        trt="Arm",
                        var="KRAS.exprs",
                        var.class="numeric", var.name="KRAS exprs",
                        percentile.cutoff=c(.25,.5,.75),
                        numerical.cutoff=NULL,
                        greater=TRUE, less=FALSE,
                        within.bin=FALSE,
                        show.itt=TRUE, show.bep=TRUE,
                        across.and.within = TRUE)
```

```
## Some NAs in var column, will define the non NA entries as BEP
```

**Across-arm and Within-arm Effect of Biomarker**  
**PFS, KRAS exprs**  
**Unadjusted, unstratified analysis**



```
##
## [1,] "Subgroup"
## [2,] "All"

code.v
"Group" "Event/N" "MST" "HR"
"CTRL" "120 / 182" "3.25" ""
```

```
## [3,] "" "TRT" "202 / 368" "5.82" "0.54"
## [4,] "BEP" "CTRL" "102 / 152" "3.58" ""
## [5,] "" "TRT" "167 / 309" "6.01" "0.56"
## [6,] "KRAS exprs(>=25%, 23.26)" "CTRL" "73 / 111" "3.38" ""
## [7,] "" "TRT" "112 / 236" "6.93" "0.45"
## [8,] "KRAS exprs(>=50%, 70.52)" "CTRL" "43 / 70" "3.58" ""
## [9,] "" "TRT" "73 / 162" "7.16" "0.42"
## [10,] "KRAS exprs(>=75%, 177.29)" "CTRL" "22 / 37" "4.73" ""
## [11,] "" "TRT" "33 / 79" "8.41" "0.38"
## [12,] "CTRL:KRAS exprs(>=25%, 23.26)" "Less" "29 / 41" "3.94" ""
## [13,] "" "Greater" "73 / 111" "3.38" "1.09"
## [14,] "CTRL:KRAS exprs(>=50%, 70.52)" "Less" "59 / 82" "3.91" ""
## [15,] "" "Greater" "43 / 70" "3.58" "1"
## [16,] "CTRL:KRAS exprs(>=75%, 177.29)" "Less" "80 / 115" "3.58" ""
## [17,] "" "Greater" "22 / 37" "4.73" "0.85"
## [18,] "TRT:KRAS exprs(>=25%, 23.26)" "Less" "55 / 73" "4.07" ""
## [19,] "" "Greater" "112 / 236" "6.93" "0.45"
## [20,] "TRT:KRAS exprs(>=50%, 70.52)" "Less" "94 / 147" "5.42" ""
## [21,] "" "Greater" "73 / 162" "7.16" "0.57"
## [22,] "TRT:KRAS exprs(>=75%, 177.29)" "Less" "134 / 230" "5.59" ""
## [23,] "" "Greater" "33 / 79" "8.41" "0.51"
##
## [1,] "CI" "raw P"
## [2,] "" ""
## [3,] "0.43 - 0.67" "7.8e-08"
## [4,] "" ""
## [5,] "0.43 - 0.71" "3.5e-06"
## [6,] "" ""
## [7,] "0.33 - 0.61" "1.5e-07"
## [8,] "" ""
## [9,] "0.29 - 0.62" "9.6e-06"
## [10,] "" ""
## [11,] "0.22 - 0.66" "0.00054"
## [12,] "" ""
## [13,] "0.7 - 1.68" "0.71"
## [14,] "" ""
## [15,] "0.67 - 1.48" "1"
## [16,] "" ""
## [17,] "0.53 - 1.37" "0.5"
## [18,] "" ""
## [19,] "0.33 - 0.63" "2e-06"
## [20,] "" ""
## [21,] "0.42 - 0.78" "0.00043"
## [22,] "" ""
## [23,] "0.35 - 0.75" "0.00061"
```

## 7.15 Response outcome, 2-arm, categorical variable

by default BEP will be defined as patients non-NA biomarker measurement

```
PlotTabForestBiomarker(data=input,
                        outcome.class="binary",
                        outcome.var=c("Response"),
```



```





rsp.cat = TRUE,
rsp.response = c("CR","PR"),
rsp.nonresponse = c("SD", "PD","NON CR/PD","NE",NA),
trt="Arm",
var="KRAS.mutant",
var.class="categorical")

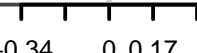
```

## Covariate adjustment and stratification are not supported for binary outcome

## Some NAs in var column, will define the non NA entries as BEP

### **Across-arm Effect of Biomarker** **Response, KRAS.mutant** **Unadjusted, unstratified analysis**

Subgroup	Group	nRsp/N	Rsp Rate	deltaRR	(CI)	raw P	
All	CTRL	14 / 182	0.08				
	TRT	96 / 368	0.26	0.18	(0.12 – 0.25)	7e-07	
BEP	CTRL	11 / 103	0.11				
	TRT	54 / 223	0.24	0.14	(0.05 – 0.22)	0.007	
KRAS.mutant(Mutant)	CTRL	5 / 34	0.15				
	TRT	18 / 61	0.3	0.15	(-0.04 – 0.34)	0.17	
KRAS.mutant(Wild Type)	CTRL	6 / 69	0.09				
	TRT	36 / 162	0.22	0.14	(0.03 – 0.24)	0.024	

  
 <--- CTRL better [deltaRR] TRT  
 \* Unadj P = 0.77 (interact)

```

##
## code.v
## [1,] "Subgroup"      "Group" "nRsp/N"  "Rsp Rate" "deltaRR"
## [2,] "All"           "CTRL"  "14 / 182" "0.08"     ""
## [3,] ""              "TRT"   "96 / 368" "0.26"     "0.18"
## [4,] "BEP"           "CTRL"  "11 / 103" "0.11"     ""
## [5,] ""              "TRT"   "54 / 223" "0.24"     "0.14"
## [6,] "KRAS.mutant(Mutant)" "CTRL"  "5 / 34"   "0.15"     ""
## [7,] ""              "TRT"   "18 / 61"   "0.3"      "0.15"
## [8,] "KRAS.mutant(Wild Type)" "CTRL"  "6 / 69"   "0.09"     ""
## [9,] ""              "TRT"   "36 / 162"  "0.22"     "0.14"
##
## [1,] "CI"           "raw P"
## [2,] ""             ""
## [3,] "0.12 - 0.25"  "7e-07"
## [4,] ""             ""
## [5,] "0.05 - 0.22"  "0.007"
## [6,] ""             ""

```

```
## [7,] "-0.04 - 0.34" "0.17"
## [8,] "" ""
## [9,] "0.03 - 0.24" "0.024"
```

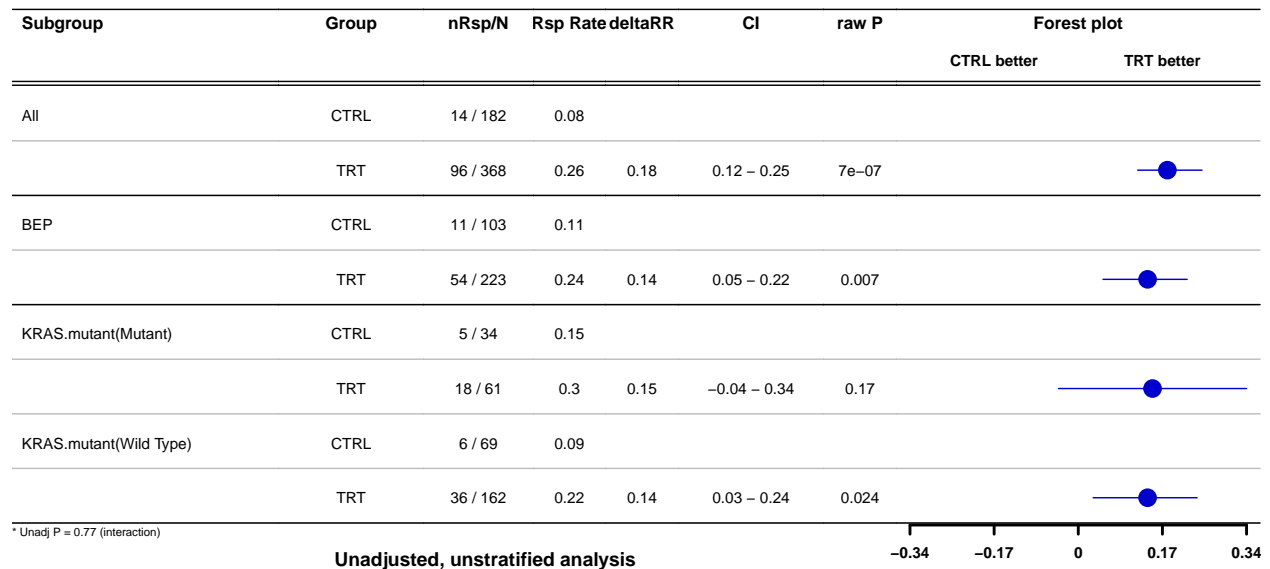
## 7.16 Response outcome, another format

```
PlotTabForestBiomarker(data=input,
                        outcome.class="binary",
                        outcome.var=c("Response"),
                        rsp.cat = TRUE,
                        rsp.response = c("CR","PR"),
                        rsp.nonresponse = c("SD", "PD","NON CR/PD","NE",NA),
                        trt="Arm",
                        var="KRAS.mutant",
                        var.class="categorical",
                        tabforest = TRUE)
```

## Covariate adjustment and stratification are not supported for binary outcome

## Some NAs in var column, will define the non NA entries as BEP

### Across-arm Effect of Biomarker Response, KRAS.mutant



```
## code.v
## [1,] "Subgroup" "Group" "nRsp/N" "Rsp Rate" "deltaRR"
## [2,] "All" "CTRL" "14 / 182" "0.08" ""
## [3,] "" "TRT" "96 / 368" "0.26" "0.18"
## [4,] "BEP" "CTRL" "11 / 103" "0.11" ""
## [5,] "" "TRT" "54 / 223" "0.24" "0.14"
## [6,] "KRAS.mutant(Mutant)" "CTRL" "5 / 34" "0.15" ""
## [7,] "" "TRT" "18 / 61" "0.3" "0.15"
## [8,] "KRAS.mutant(Wild Type)" "CTRL" "6 / 69" "0.09" ""
## [9,] "" "TRT" "36 / 162" "0.22" "0.14"
##
```

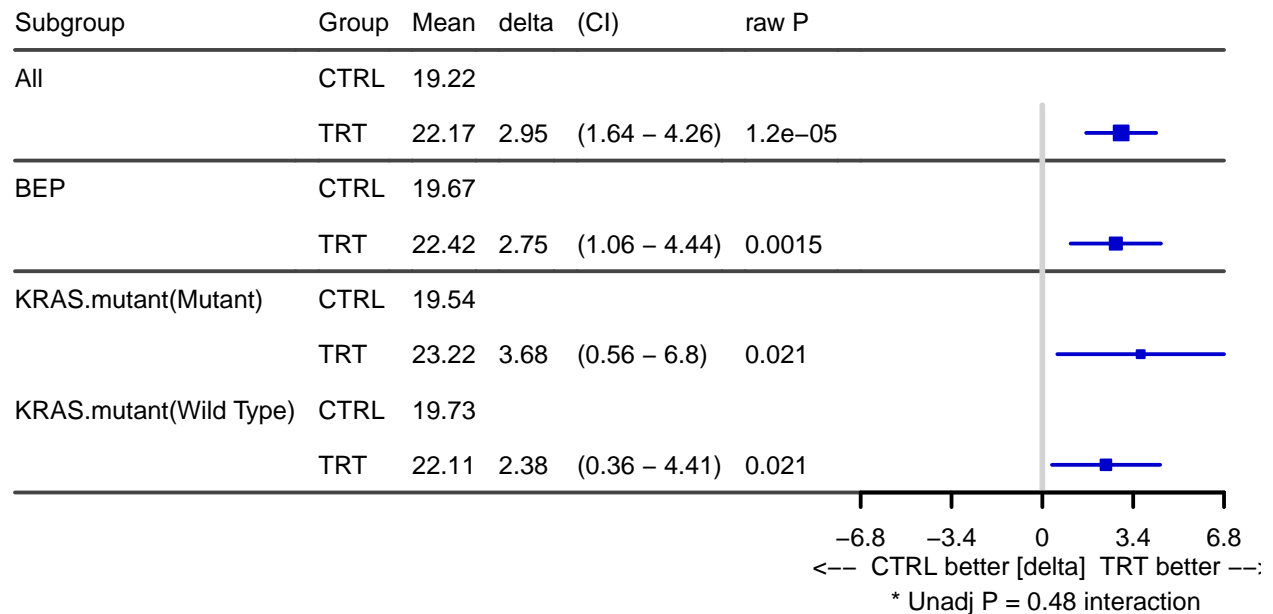
```
## [1,] "CI"          "raw P"
## [2,] ""           ""
## [3,] "0.12 - 0.25" "7e-07"
## [4,] ""           ""
## [5,] "0.05 - 0.22" "0.007"
## [6,] ""           ""
## [7,] "-0.04 - 0.34" "0.17"
## [8,] ""           ""
## [9,] "0.03 - 0.24" "0.024"
```

## 7.17 Continuous outcome, 2-arm, categorical variable

```
PlotTabForestBiomarker(data=input,
                        outcome.class="continuous",
                        outcome.var=c("Lab_ontrt"),
                        trt="Arm",
                        var="KRAS.mutant",
                        var.class="categorical")
```

```
## Stratification is not supported for continuous outcome
## Some NAs in var column, will define the non NA entries as BEP
```

### Across-arm Effect of Biomarker Lab\_ontrt, KRAS.mutant Unadjusted, unstratified analysis



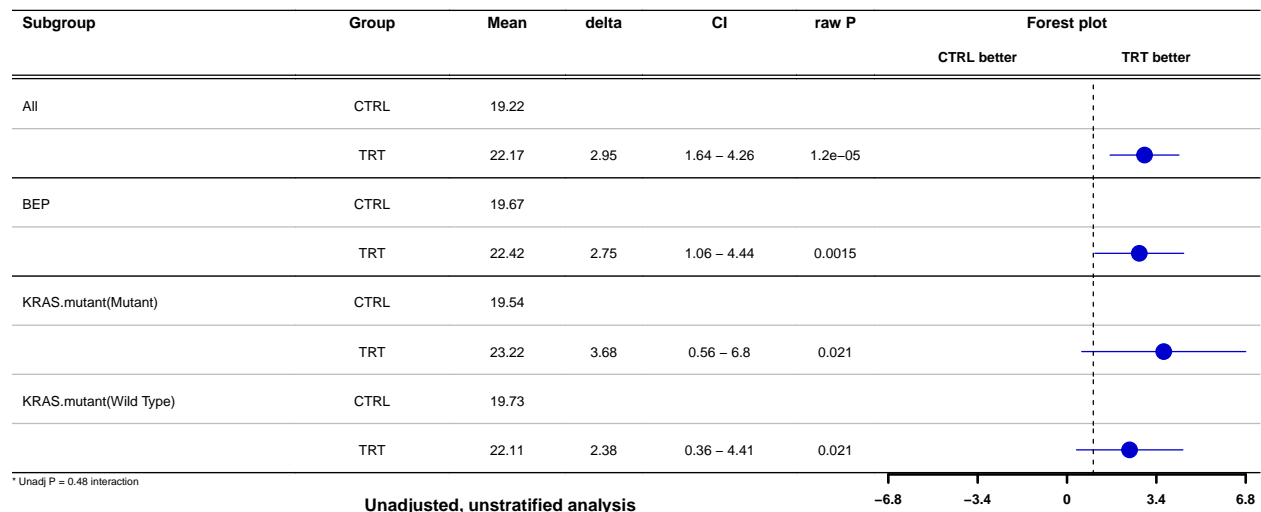
```
## code.v
## [1,] "Subgroup"      "Group" "Mean" "delta" "CI"
## [2,] "All"          "CTRL"  "19.22" ""      ""
## [3,] ""              "TRT"   "22.17" "2.95" "1.64 - 4.26"
```

```
## [4,] "BEP" "CTRL" "19.67" "" ""
## [5,] "" "TRT" "22.42" "2.75" "1.06 - 4.44"
## [6,] "KRAS.mutant(Mutant)" "CTRL" "19.54" "" ""
## [7,] "" "TRT" "23.22" "3.68" "0.56 - 6.8"
## [8,] "KRAS.mutant(Wild Type)" "CTRL" "19.73" "" ""
## [9,] "" "TRT" "22.11" "2.38" "0.36 - 4.41"
##
## [1,] "raw P"
## [2,] ""
## [3,] "1.2e-05"
## [4,] ""
## [5,] "0.0015"
## [6,] ""
## [7,] "0.021"
## [8,] ""
## [9,] "0.021"
```

```
PlotTabForestBiomarker(data=input,
                        outcome.class="continuous",
                        outcome.var=c("Lab_ontrt"),
                        trt="Arm",
                        var="KRAS.mutant",
                        var.class="categorical",tabforest=T)
```

```
## Stratification is not supported for continuous outcome
## Some NAs in var column, will define the non NA entries as BEP
```

### Across-arm Effect of Biomarker Lab\_ontrt, KRAS.mutant



```
## code.v
## [1,] "Subgroup" "Group" "Mean" "delta" "CI"
## [2,] "All" "CTRL" "19.22" "" ""
## [3,] "" "TRT" "22.17" "2.95" "1.64 - 4.26"
## [4,] "BEP" "CTRL" "19.67" "" ""
## [5,] "" "TRT" "22.42" "2.75" "1.06 - 4.44"
## [6,] "KRAS.mutant(Mutant)" "CTRL" "19.54" "" ""
## [7,] "" "TRT" "23.22" "3.68" "0.56 - 6.8"
```

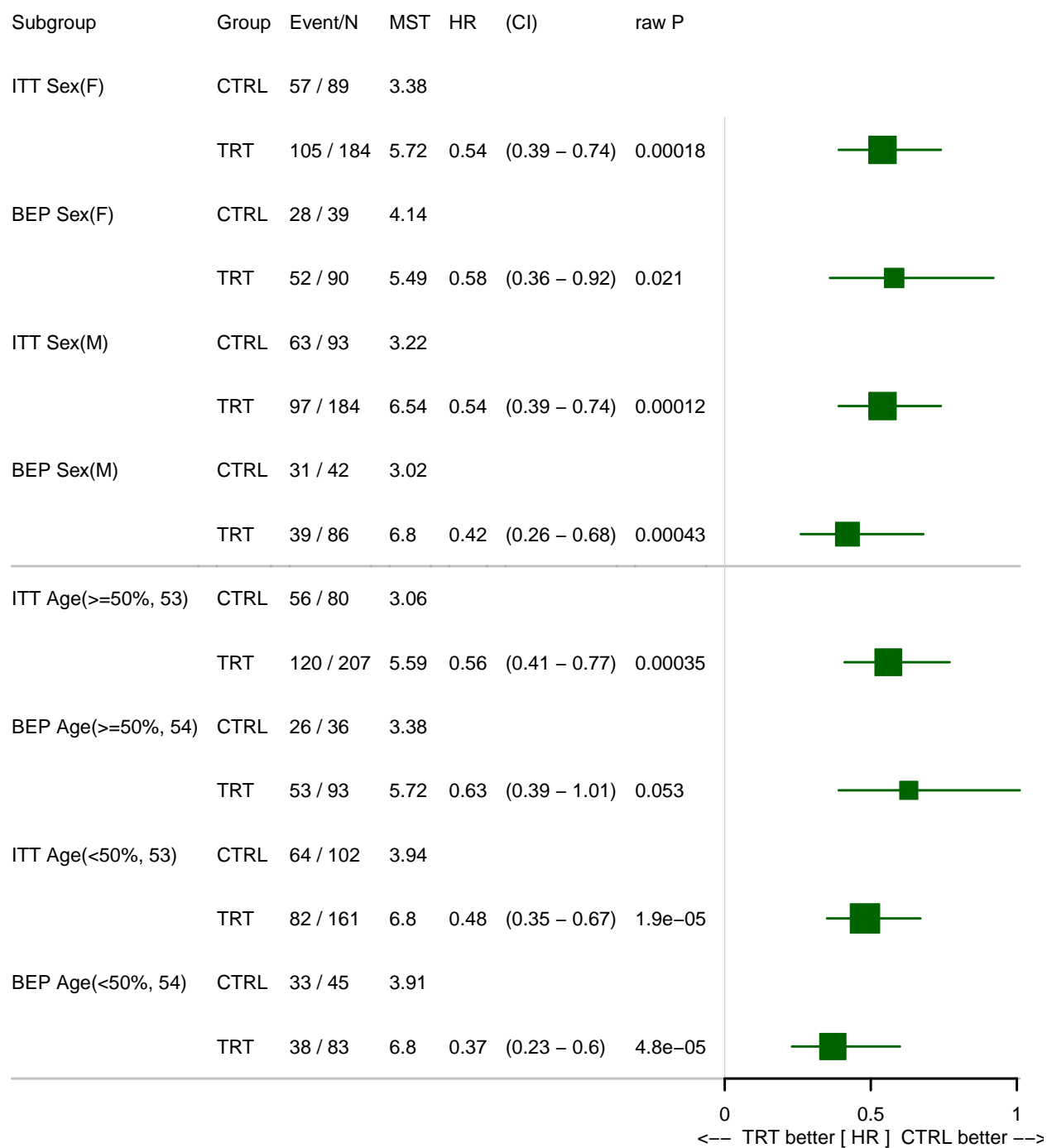
```
## [8,] "KRAS.mutant(Wild Type)" "CTRL" "19.73" "" ""
## [9,] "" "TRT" "22.11" "2.38" "0.36 - 4.41"
##
## [1,] "raw P"
## [2,] ""
## [3,] "1.2e-05"
## [4,] ""
## [5,] "0.0015"
## [6,] ""
## [7,] "0.021"
## [8,] ""
## [9,] "0.021"
```

## 8 PlotTabForestMulti(): Forest plots to compare multiple variates' effect in ITT vs BEP, or compare multiple variates' effect in subpopulations

### 8.1 2-arm, compare BEP vs ITT

```
PlotTabForestMulti(data=input,
                    outcome.class="survival",
                    outcome.var=c("PFS", "PFS.event"),
                    trt="Arm",
                    var=c("Sex", "Age"),
                    bep="BEP", bep.indicator=1,
                    compare.bep.itt=TRUE
)
```

**Across arm, Compare BEP vs. All  
PFS  
Unadjusted, unstratified analysis**



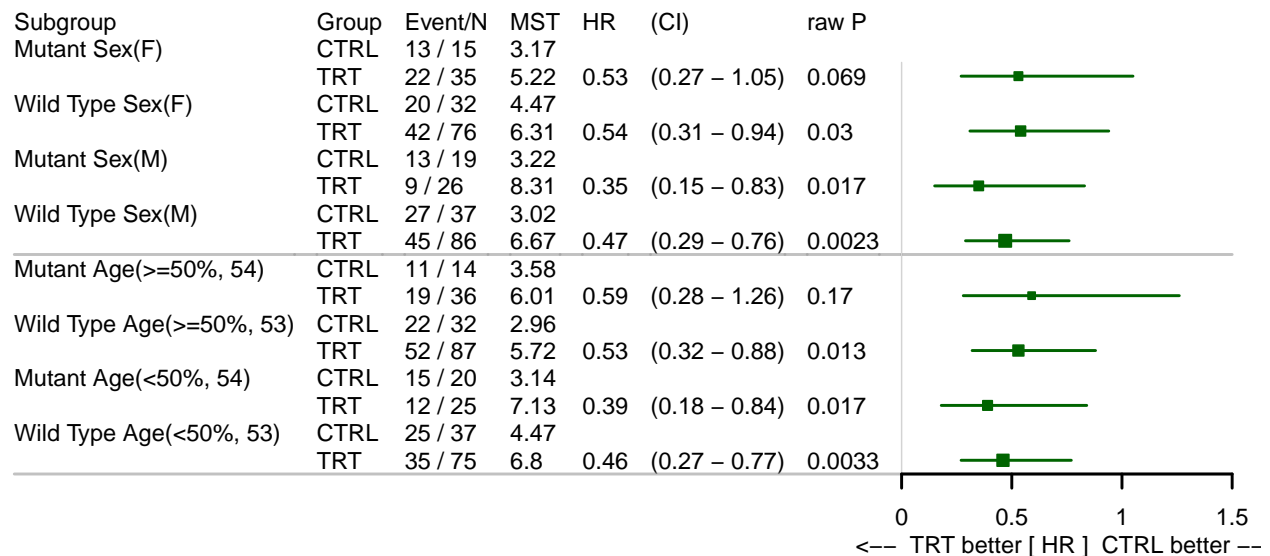
## 8.2 2-arm, compare subgroups defined by biomarker

```
PlotTabForestMulti(data=input,
                    outcome.class="survival",
                    outcome.var=c("PFS","PFS.event"),
                    trt="Arm",
                    var=c("Sex","Age"),
                    bep="BEP",bep.indicator=1, compare.bep.itt=FALSE,
                    compare.subgroup=TRUE, subgroup="KRAS.mutant"
)
```

### Across arm, Compare KRAS.mutant subgroup

#### PFS

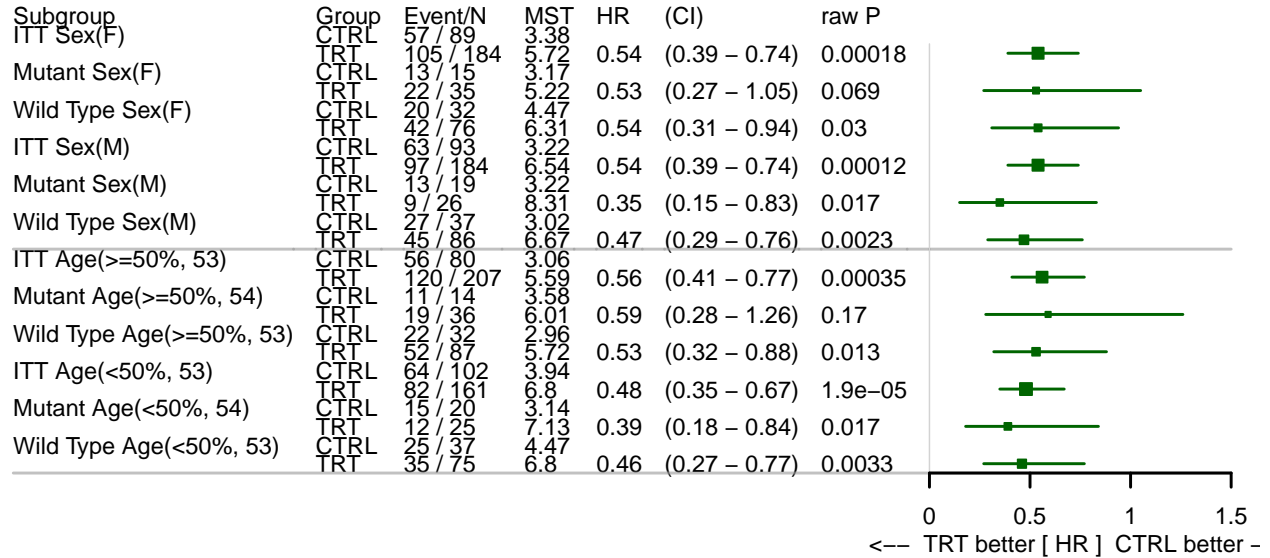
#### Unadjusted, unstratified analysis



## 8.3 2-arm, compare subgroups defined by biomarker, also show ITT

```
PlotTabForestMulti(data=input,
                    outcome.class="survival",
                    outcome.var=c("PFS","PFS.event"),
                    trt="Arm",
                    var=c("Sex","Age"),
                    bep="BEP",bep.indicator=1, compare.bep.itt=FALSE,
                    compare.subgroup=TRUE, subgroup="KRAS.mutant", show.itt=TRUE
)
```

**Across arm, Compare KRAS.mutant subgroup**  
**PFS**  
**Unadjusted, unstratified analysis**

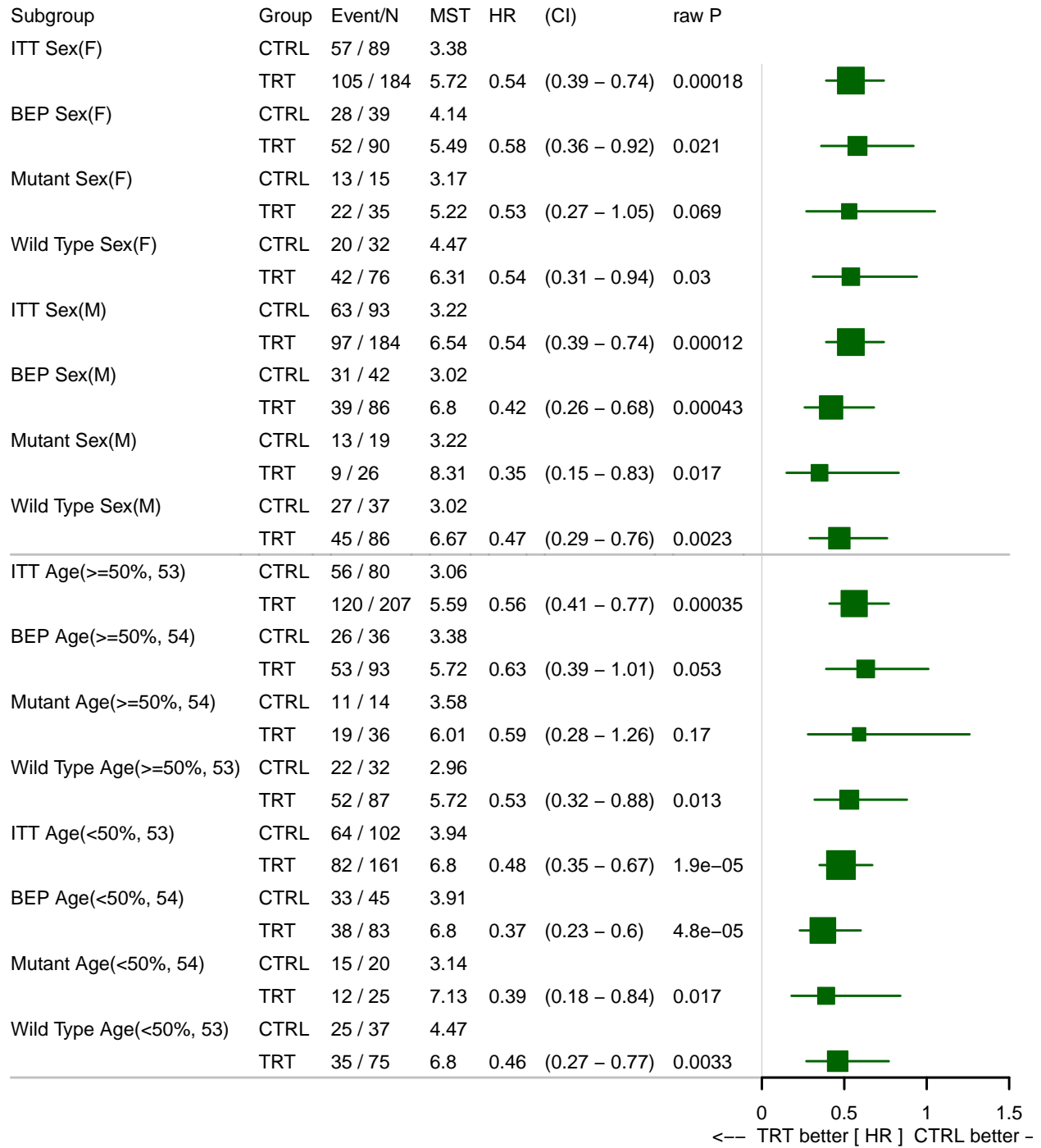


#### 8.4 2-arm, compare subgroups defined by biomarker, also show BEP

```
PlotTabForestMulti(data=input,
                    outcome.class="survival",
                    outcome.var=c("PFS", "PFS.event"),
                    trt="Arm",
                    var=c("Sex", "Age"),
                    bep="BEP", bep.indicator=1, compare.bep.itt=FALSE,
                    compare.subgroup=TRUE, subgroup="KRAS.mutant", show.itt=TRUE, show.bep=TRUE
)
```



**Across arm, Compare KRAS.mutant subgroup**  
**PFS**  
**Unadjusted, unstratified analysis**



If bep column is not defined, the program will take the non NA entries in subgroups column as BEP

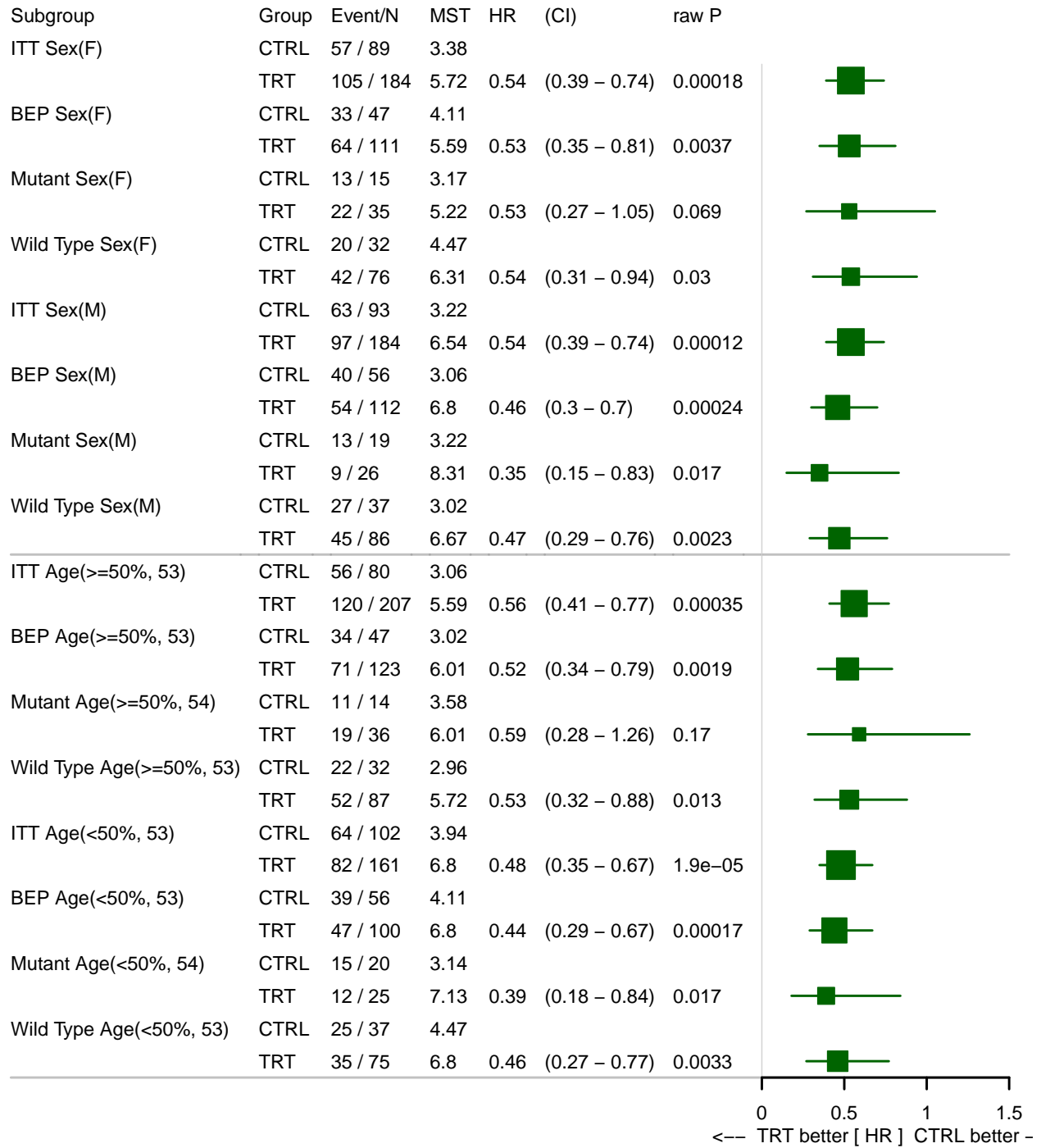
```

PlotTabForestMulti(data=input,
                    outcome.class="survival",
                    outcome.var=c("PFS", "PFS.event"),
                    trt="Arm",
                    var=c("Sex", "Age"), compare.bep.itt=FALSE,
                    compare.subgroup=TRUE, subgroup="KRAS.mutant", show.itt=TRUE, show.bep=TRUE
                    )

```

## show.bep is TRUE but bep is not specified, will define the non NA entries in subgroup column as BEP

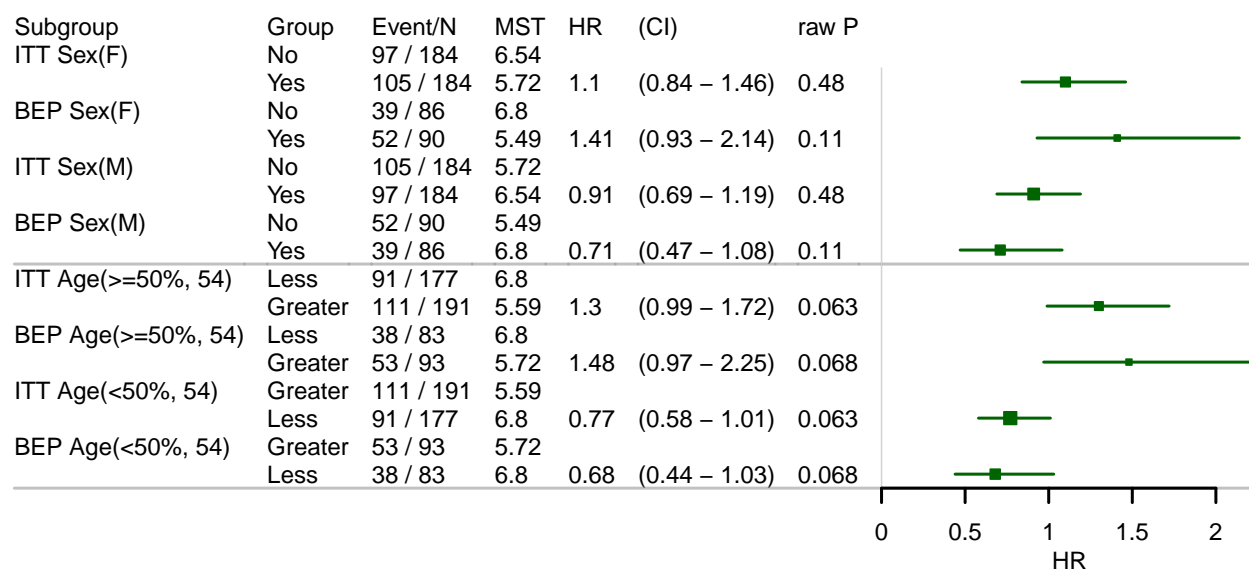
**Across arm, Compare KRAS.mutant subgroup**  
**PFS**  
**Unadjusted, unstratified analysis**



## 8.5 1-arm, compare ITT vs BEP

```
PlotTabForestMulti(data=subset(input,Arm=="TRT"),
  outcome.class="survival",
  outcome.var=c("PFS","PFS.event"),
  trt=NULL,
  var=c("Sex","Age"),
  bep="BEP",bep.indicator=1, compare.bep.itt=TRUE
)
```

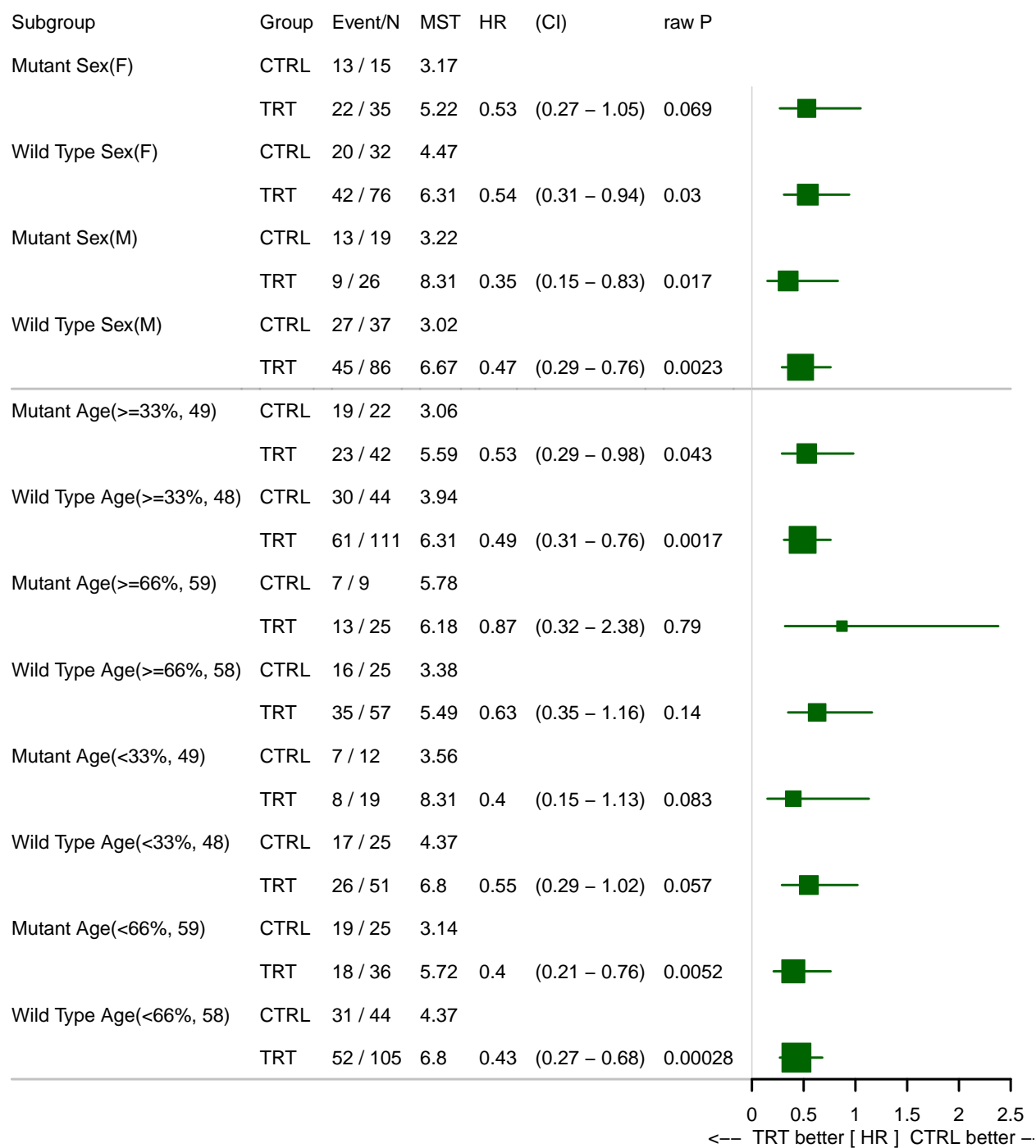
### Within arm, Compare BEP vs. All PFS Unadjusted, unstratified analysis



## 8.6 2-arm, compare subgroups defined by biomarker, multiple cutoffs for continuous biomarker

```
PlotTabForestMulti(data=input,
  outcome.class="survival",
  outcome.var=c("PFS","PFS.event"),
  trt="Arm", percentile.cutoff=c(.33,.66),
  var=c("Sex","Age"),
  bep="BEP",bep.indicator=1, compare.bep.itt=FALSE,
  compare.subgroup=TRUE, subgroup="KRAS.mutant"
)
```

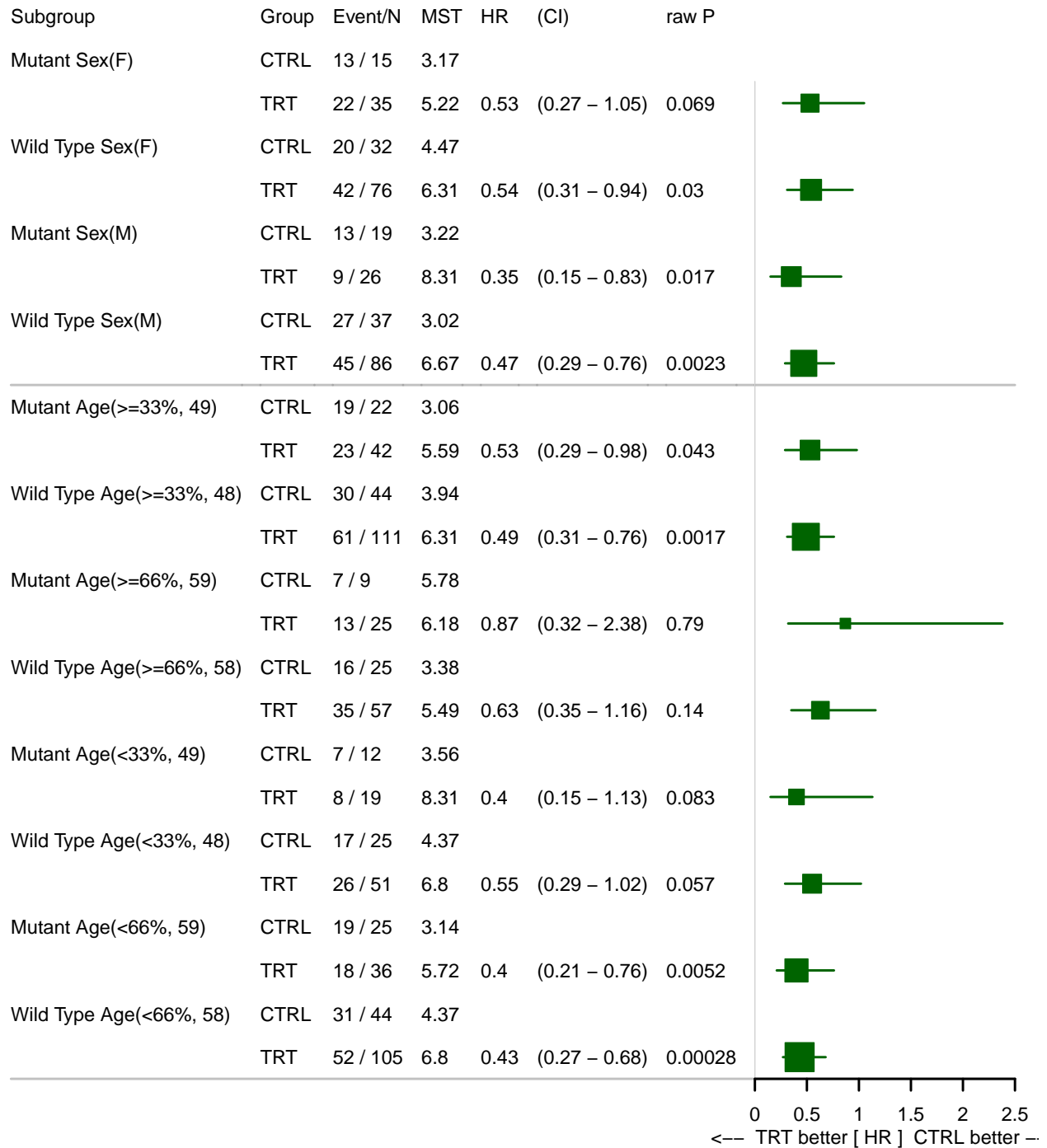
**Across arm, Compare KRAS.mutant subgroup**  
**PFS**  
**Unadjusted, unstratified analysis**



## 8.7 2-arm, compare subgroups defined by biomarker, multiple cutoffs for continuous biomarker, calculate “< cutoff”

```
PlotTabForestMulti(data=input,
                    outcome.class="survival",
                    outcome.var=c("PFS", "PFS.event"),
                    trt="Arm", percentile.cutoff=c(.33,.66),less=TRUE,greater=FALSE,
                    var=c("Sex", "Age"),
                    bep="BEP",bep.indicator=1, compare.bep.itt=FALSE,
                    compare.subgroup=TRUE, subgroup="KRAS.mutant"
                    )
```

**Across arm, Compare KRAS.mutant subgroup**  
**PFS**  
**Unadjusted, unstratified analysis**



## 8.8 2-arm, compare subgroups defined by biomarker, multiple cutoffs for continuous biomarker, calculate within bin statistics

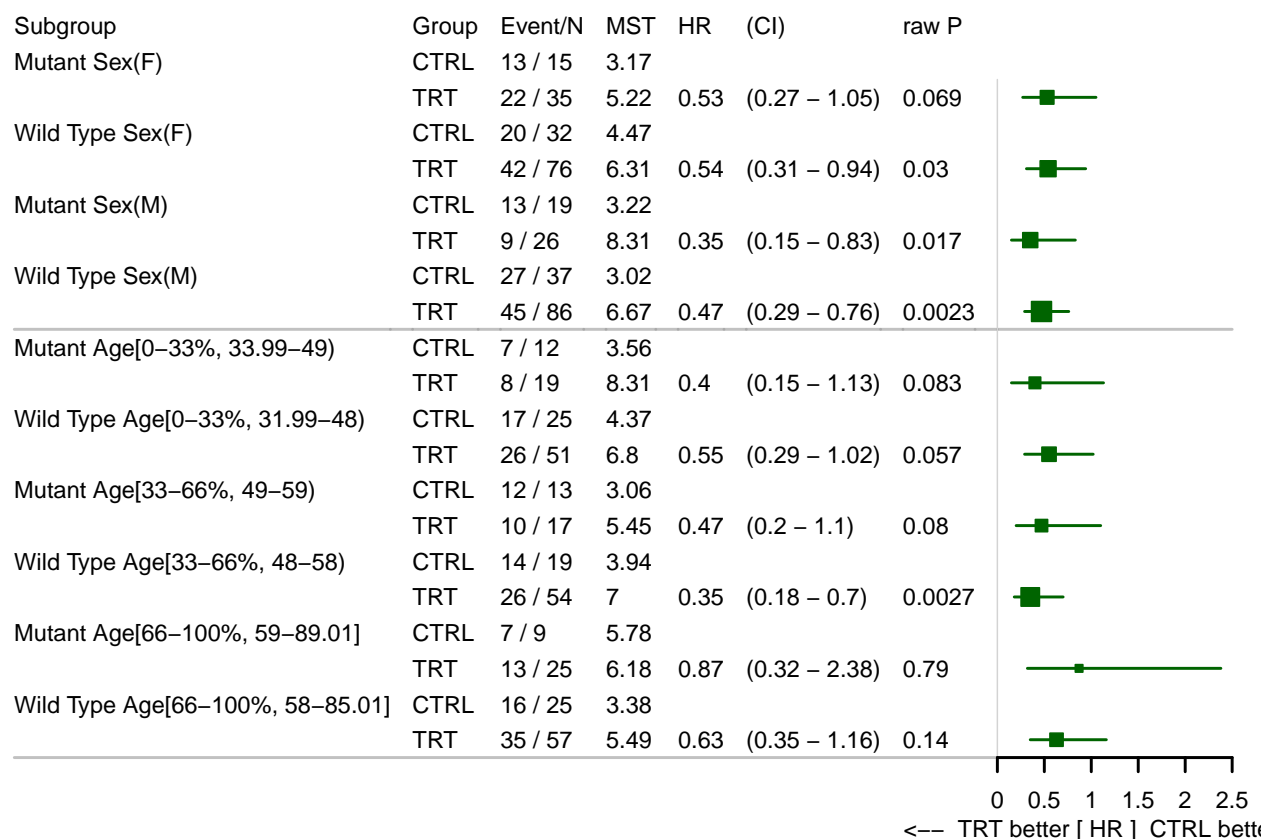
```
PlotTabForestMulti(data=input,
                    outcome.class="survival",
                    outcome.var=c("PFS", "PFS.event"),
                    trt="Arm", percentile.cutoff=c(.33,.66),within.bin=TRUE,
                    var=c("Sex","Age"),
                    bep="BEP",bep.indicator=1, compare.bep.itt=FALSE,
                    compare.subgroup=TRUE, subgroup="KRAS.mutant"
)
```

```
## within.bin is TRUE, greater and less will be ignored
## within.bin is TRUE, greater and less will be ignored
## within.bin is TRUE, greater and less will be ignored
## within.bin is TRUE, greater and less will be ignored
```

### Across arm, Compare KRAS.mutant subgroup

#### PFS

#### Unadjusted, unstratified analysis

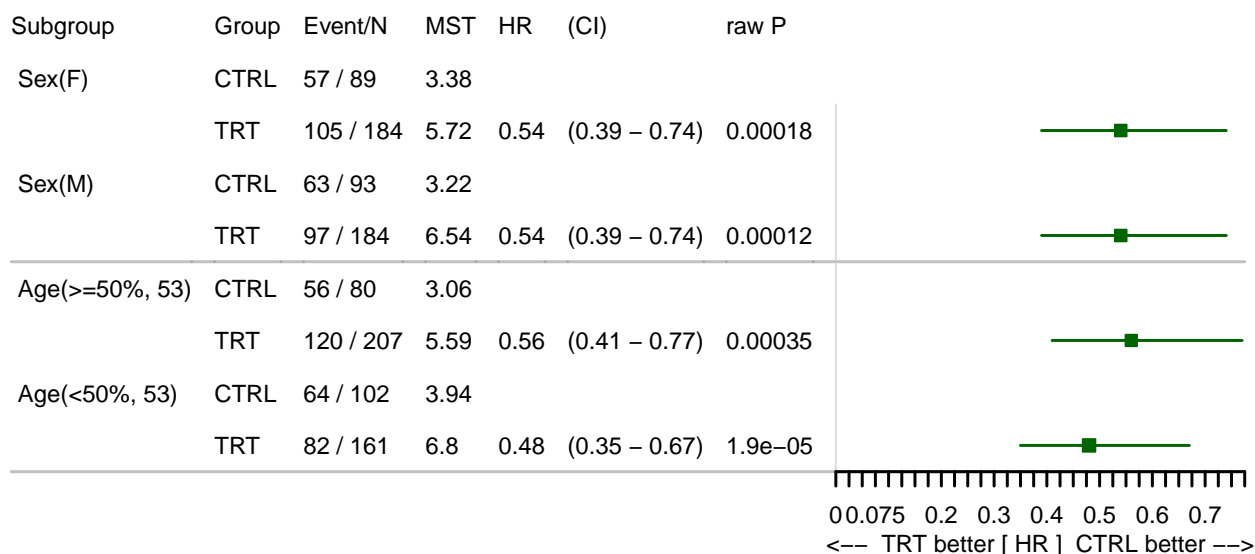




## 8.9 Basic forest plot without comparison

```
PlotTabForestMulti(data=input,
                    outcome.class="survival",
                    outcome.var=c("PFS","PFS.event"),
                    trt="Arm",
                    var=c("Sex","Age"),
                    compare.bep.itt=FALSE, compare.subgroup=FALSE,itt.name=""
)
```

### Across arm PFS Unadjusted, unstratified analysis

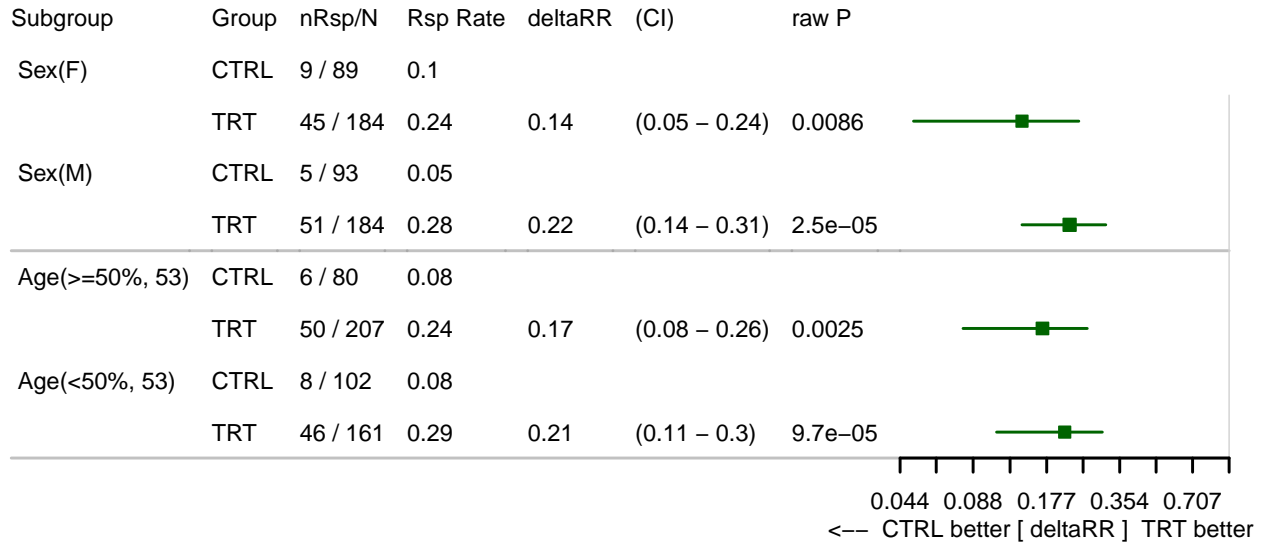


## 8.10 Response outcome

```
PlotTabForestMulti(data=input,
                    outcome.class="binary",
                    outcome.var=c("Response"),
                    rsp.cat = TRUE,
                    rsp.response = c("CR","PR"),
                    rsp.nonresponse = c("SD", "PD","NON CR/PD","NE",NA),
                    trt="Arm",
                    var=c("Sex","Age"),
                    compare.bep.itt=FALSE, compare.subgroup=FALSE,itt.name=""
)
```

```
## Covariate adjustment and stratification are not supported for binary outcome
## Covariate adjustment and stratification are not supported for binary outcome
```

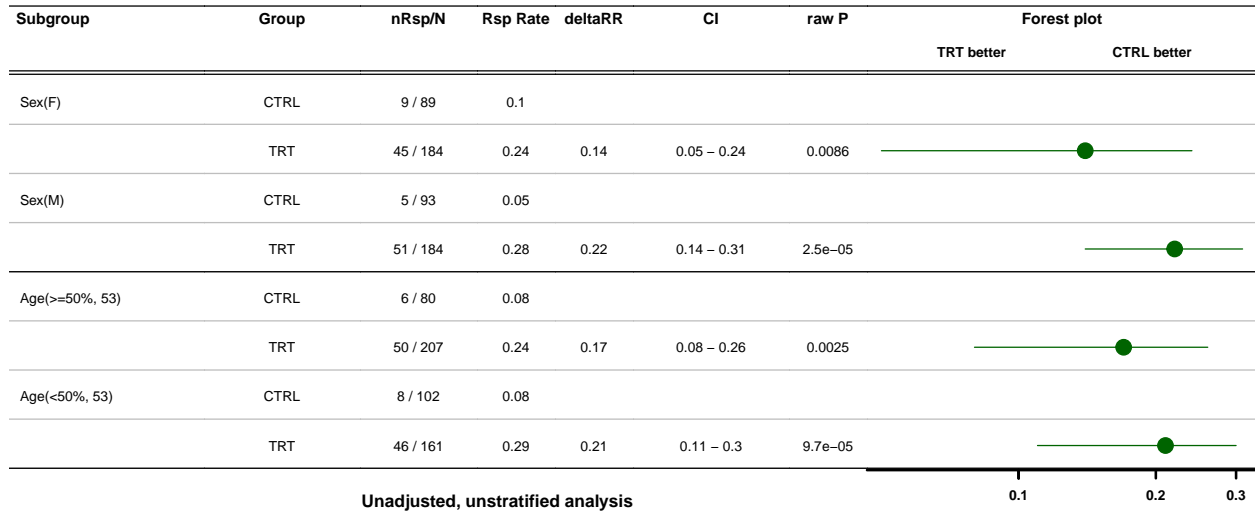
**Across arm  
Response  
Unadjusted, unstratified analysis**



```
PlotTabForestMulti(data=input,
                    outcome.class="binary",
                    outcome.var=c("Response"),
                    rsp.cat = TRUE,
                    rsp.response = c("CR","PR"),
                    rsp.nonresponse = c("SD", "PD","NON CR/PD","NE",NA),
                    trt="Arm",
                    var=c("Sex","Age"),
                    compare.bep.itt=FALSE, compare.subgroup=FALSE,itt.name="", tabforest=T
)
```

```
## Covariate adjustment and stratification are not supported for binary outcome
## Covariate adjustment and stratification are not supported for binary outcome
## Warning in log(clip): NaNs produced
```

## Across arm Response

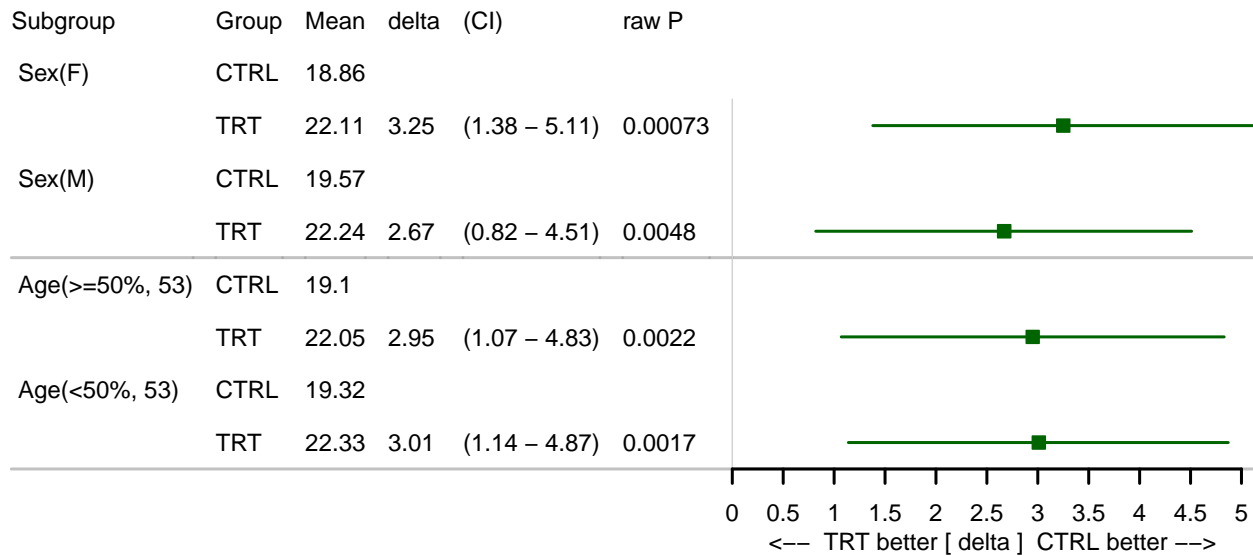


### 8.11 Continuous endpoint

```
PlotTabForestMulti(data=input,
                    outcome.class="continuous",
                    outcome.var=c("Lab_ontrt"),
                    trt="Arm",
                    var=c("Sex", "Age"),
                    compare.bep.itt=FALSE, compare.subgroup=FALSE, itt.name=""
)
```

```
## Stratification is not supported for continuous outcome
## Stratification is not supported for continuous outcome
```

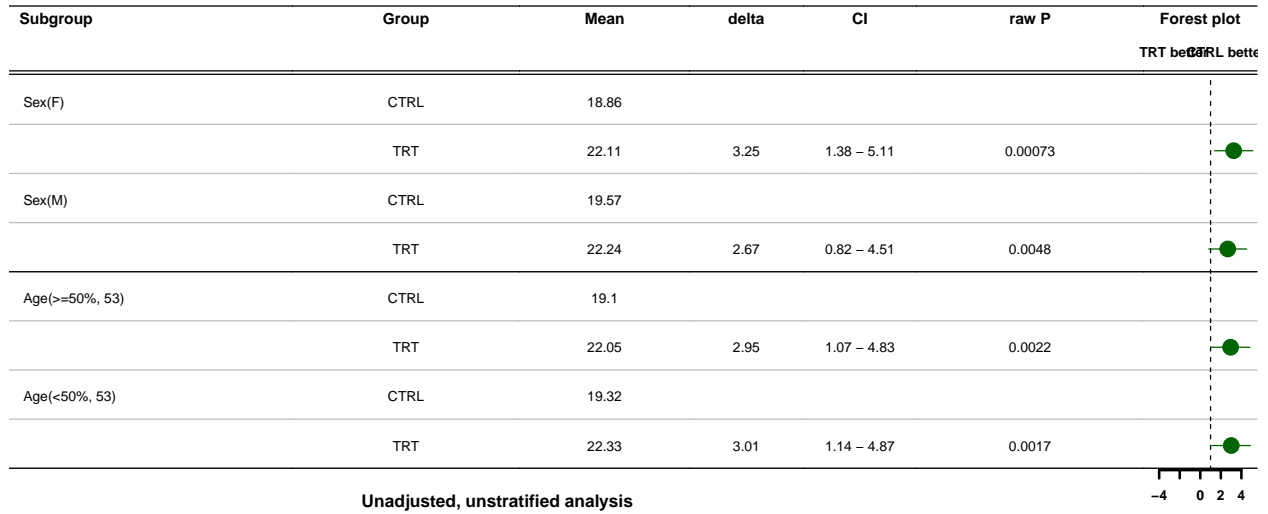
**Across arm**  
**Lab\_ontrt**  
**Unadjusted, unstratified analysis**



```
PlotTabForestMulti(data=input,
  outcome.class="continuous",
  outcome.var=c("Lab_ontrt"),
  trt="Arm",
  var=c("Sex", "Age"),
  compare.bep.itt=FALSE, compare.subgroup=FALSE, itt.name="", tabforest=T
)
```

```
## Stratification is not supported for continuous outcome
## Stratification is not supported for continuous outcome
```

## Across arm Lab\_ontrt



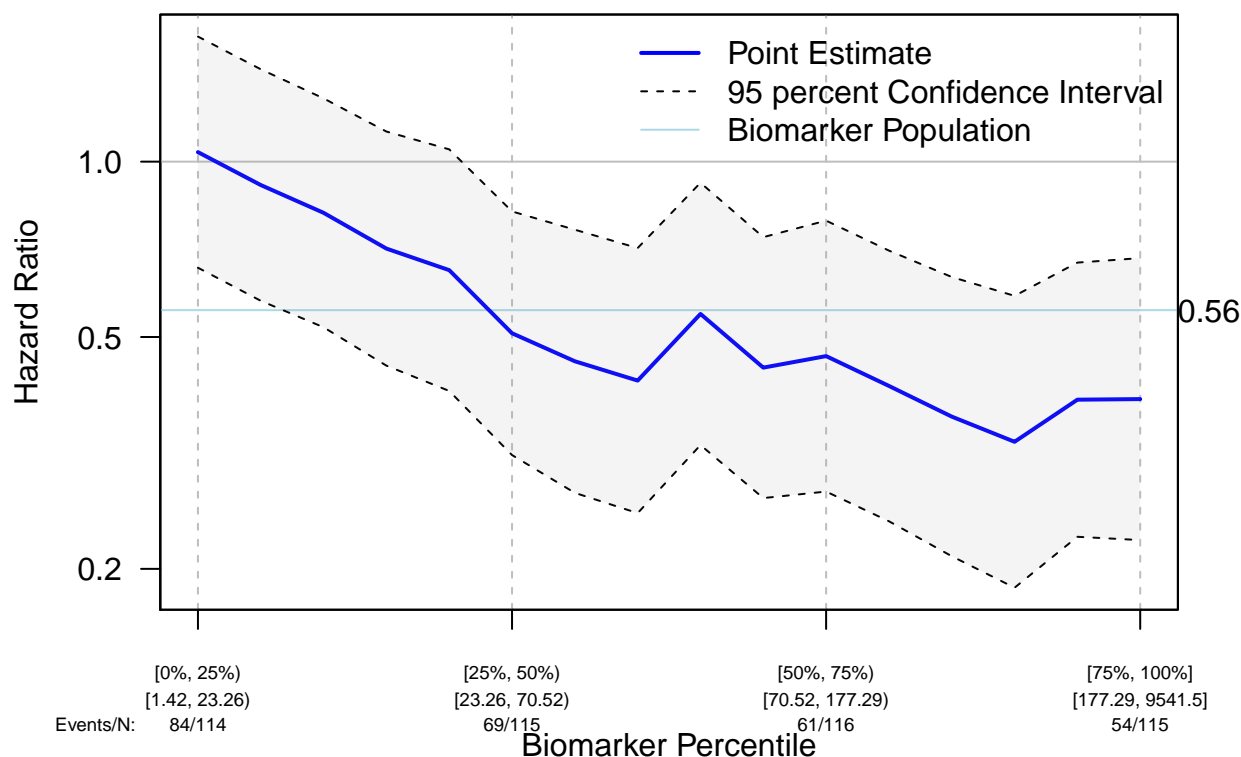
## 9 PlotSTEPP(): STEPP (Subpopulation Treatment Effect Pattern Plot) from the given point estimates and confidence intervals at desired percentiles.

### 9.1 Survival outcome

```
PlotSTEPP(data = input,
  outcome.var = c("PFS", "PFS.event"),
  outcome.class = "survival",
  trt = "Arm",
  var = "KRAS.exprs",
  placebo.code = "CTRL",
  active.code = "TRT",
  csv.name = NULL,
  pdf.name = NULL
)
```

## some NA in var column, will ignore NA entries

## STEPP: Subgroup Treatment Effect Pattern Plot



##	Window Center	Hazard Ratio	CI Lower	CI Upper	BM Lower	BM Upper
## [1,]	0.125	1.0382905	0.6576013	1.6393629	1.42	23.26
## [2,]	0.175	0.9116623	0.5769829	1.4404727	5.98	32.00
## [3,]	0.225	0.8168277	0.5196594	1.2839325	9.51	42.81
## [4,]	0.275	0.7091655	0.4465512	1.1262218	14.03	48.84
## [5,]	0.325	0.6512376	0.4040444	1.0496628	19.70	59.30
## [6,]	0.375	0.5074886	0.3136041	0.8212413	23.26	70.52
## [7,]	0.425	0.4542622	0.2700666	0.7640860	32.00	77.71
## [8,]	0.475	0.4208886	0.2493019	0.7105732	42.81	95.01
## [9,]	0.525	0.5477689	0.3262293	0.9197544	48.84	121.94
## [10,]	0.575	0.4430307	0.2645633	0.7418874	59.30	151.17
## [11,]	0.625	0.4636518	0.2714610	0.7919111	70.52	177.29
## [12,]	0.675	0.4121338	0.2414306	0.7035327	77.71	213.78
## [13,]	0.725	0.3650580	0.2102587	0.6338257	95.01	259.57
## [14,]	0.775	0.3304610	0.1856466	0.5882385	121.94	317.37
## [15,]	0.825	0.3903051	0.2270411	0.6709713	151.17	467.88
## [16,]	0.875	0.3912302	0.2241152	0.6829571	177.29	9541.50
##	BM Center	Window Left	Window Right	N	Events	
## [1,]	11.010	0.00	0.25	114	84	
## [2,]	16.450	0.05	0.30	115	84	
## [3,]	21.860	0.10	0.35	115	85	
## [4,]	27.860	0.15	0.40	115	78	
## [5,]	36.250	0.20	0.45	115	73	
## [6,]	45.250	0.25	0.50	115	69	
## [7,]	53.265	0.30	0.55	114	59	
## [8,]	64.450	0.35	0.60	115	59	
## [9,]	73.010	0.40	0.65	115	63	
## [10,]	88.030	0.45	0.70	115	62	

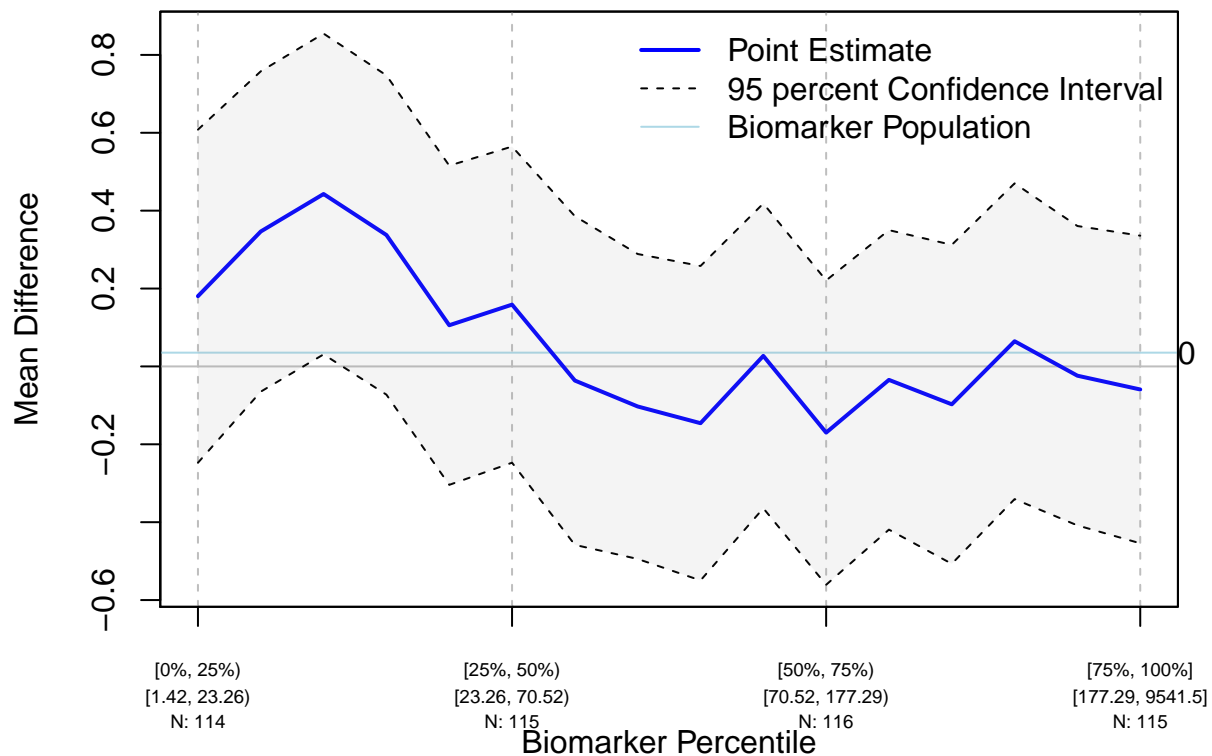
```
## [11,] 101.480      0.50      0.75 116      61
## [12,] 133.900      0.55      0.80 116      61
## [13,] 165.420      0.60      0.85 115      55
## [14,] 195.360      0.65      0.90 115      51
## [15,] 238.860      0.70      0.95 115      56
## [16,] 292.040      0.75      1.00 115      54
```

## 9.2 Continuous outcome

```
PlotSTEPP(data = input,
  outcome.var = "Baseline.SLD",
  outcome.class = "continuous",
  trt = "Arm",
  var = "KRAS.exprs",
  covariate= "Sex",
  placebo.code = "CTRL",
  active.code = "TRT",
  csv.name = NULL,
  pdf.name = NULL
)
```

## some NA in var column, will ignore NA entries

### STEPP: Subgroup Treatment Effect Pattern Plot



##	Window	Center	Mean Difference	CI Lower	CI Upper	BM Lower
##	[1,]	0.125	0.18035575	-0.24729997	0.6080115	1.42
##	[2,]	0.175	0.34651479	-0.06475305	0.7577826	5.98
##	[3,]	0.225	0.44278821	0.03092580	0.8546506	9.51

```
## [4,]      0.275      0.33749063 -0.07236992 0.7473512      14.03
## [5,]      0.325      0.10572728 -0.30412158 0.5155761      19.70
## [6,]      0.375      0.15866042 -0.24724256 0.5645634      23.26
## [7,]      0.425     -0.03630432 -0.45831952 0.3857109      32.00
## [8,]      0.475     -0.10286329 -0.49436652 0.2886399      42.81
## [9,]      0.525     -0.14583073 -0.54953031 0.2578689      48.84
## [10,]     0.575      0.02704796 -0.36419706 0.4182930      59.30
## [11,]     0.625     -0.16994763 -0.56071364 0.2208184      70.52
## [12,]     0.675     -0.03456171 -0.41907553 0.3499521      77.71
## [13,]     0.725     -0.09734050 -0.50684711 0.3121661      95.01
## [14,]     0.775      0.06465414 -0.34072771 0.4700360     121.94
## [15,]     0.825     -0.02383926 -0.40821143 0.3605329     151.17
## [16,]     0.875     -0.05926090 -0.45420790 0.3356861     177.29
##      BM Upper BM Center Window Left Window Right      N
## [1,]    23.26    11.010      0.00      0.25 114
## [2,]    32.00    16.450      0.05      0.30 115
## [3,]    42.81    21.860      0.10      0.35 115
## [4,]    48.84    27.860      0.15      0.40 115
## [5,]    59.30    36.250      0.20      0.45 115
## [6,]    70.52    45.250      0.25      0.50 115
## [7,]    77.71    53.265      0.30      0.55 114
## [8,]    95.01    64.450      0.35      0.60 115
## [9,]   121.94    73.010      0.40      0.65 115
## [10,]   151.17    88.030      0.45      0.70 115
## [11,]   177.29   101.480      0.50      0.75 116
## [12,]   213.78   133.900      0.55      0.80 116
## [13,]   259.57   165.420      0.60      0.85 115
## [14,]   317.37   195.360      0.65      0.90 115
## [15,]   467.88   238.860      0.70      0.95 115
## [16,]  9541.50   292.040      0.75      1.00 115
```

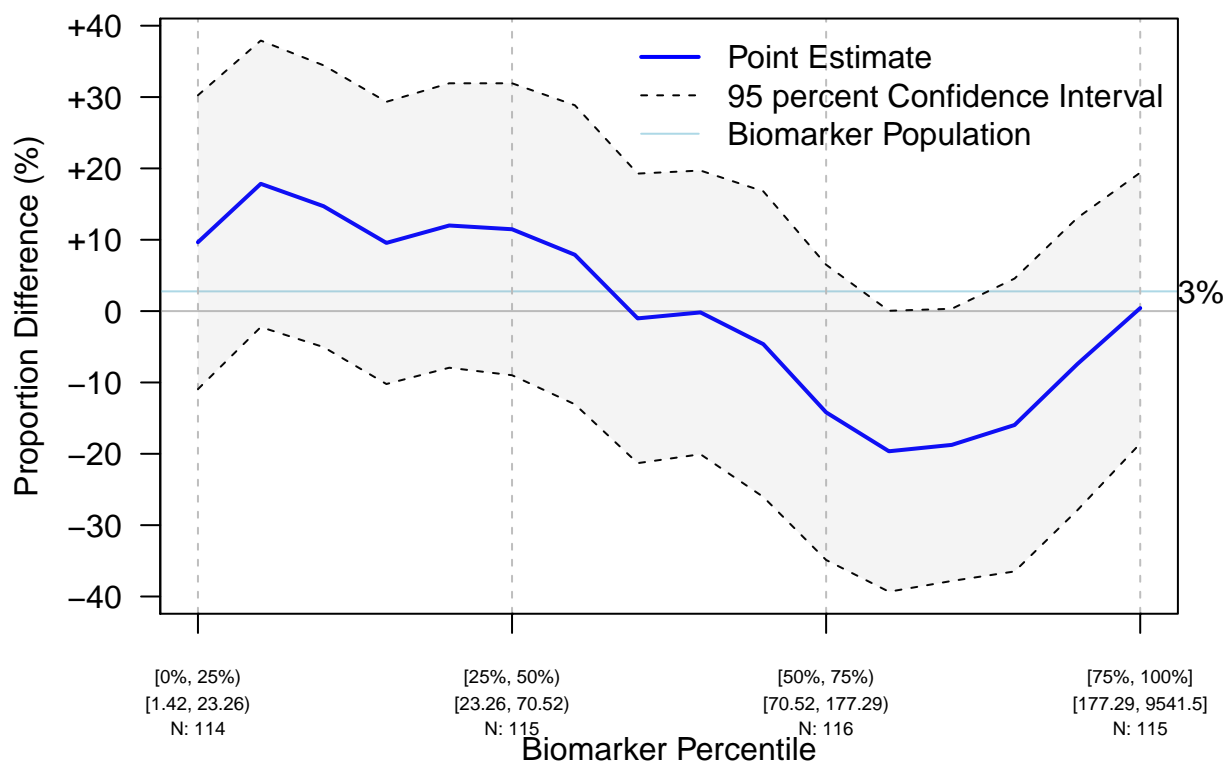
### 9.3 Binary outcome variable

```
PlotSTEPP(data = input,
  outcome.var = "ECOG",
  outcome.class = "binary",
  trt = "Arm",
  var = "KRAS.exprs",
  placebo.code = "CTRL",
  active.code = "TRT",
  csv.name = NULL,
  pdf.name = NULL
)
```

```
## some NA in var column, will ignore NA entries
```



## STEPP: Subgroup Treatment Effect Pattern Plot



##	Window	Center	Proportion	Difference	CI Lower	CI Upper	
##	[1,]	0.125		0.096558637	-0.10936788	0.3024851571	
##	[2,]	0.175		0.178408350	-0.02235039	0.3791670900	
##	[3,]	0.225		0.147000659	-0.05041839	0.3444197101	
##	[4,]	0.275		0.095583388	-0.10218602	0.2933527961	
##	[5,]	0.325		0.119973632	-0.07934753	0.3192947989	
##	[6,]	0.375		0.114700066	-0.08986157	0.3192616985	
##	[7,]	0.425		0.078947368	-0.13057600	0.2884707407	
##	[8,]	0.475		-0.010179010	-0.21308461	0.1927265895	
##	[9,]	0.525		-0.001808318	-0.20059338	0.1969767445	
##	[10,]	0.575		-0.046292948	-0.26043294	0.1678470400	
##	[11,]	0.625		-0.141906874	-0.34878874	0.0649749952	
##	[12,]	0.675		-0.196428571	-0.39328032	0.0004231801	
##	[13,]	0.725		-0.187500000	-0.37820558	0.0032055783	
##	[14,]	0.775		-0.159638554	-0.36486679	0.0455896813	
##	[15,]	0.825		-0.074191280	-0.27905945	0.1306768944	
##	[16,]	0.875		0.004219409	-0.18560507	0.1940438921	
##	BM Lower	BM Upper	BM Center	Window	Left	Window Right	N
##	[1,]	1.42	23.26	11.010	0.00	0.25	114
##	[2,]	5.98	32.00	16.450	0.05	0.30	115
##	[3,]	9.51	42.81	21.860	0.10	0.35	115
##	[4,]	14.03	48.84	27.860	0.15	0.40	115
##	[5,]	19.70	59.30	36.250	0.20	0.45	115
##	[6,]	23.26	70.52	45.250	0.25	0.50	115
##	[7,]	32.00	77.71	53.265	0.30	0.55	114
##	[8,]	42.81	95.01	64.450	0.35	0.60	115
##	[9,]	48.84	121.94	73.010	0.40	0.65	115
##	[10,]	59.30	151.17	88.030	0.45	0.70	115

```
## [11,]    70.52   177.29   101.480         0.50         0.75 116
## [12,]    77.71   213.78   133.900         0.55         0.80 116
## [13,]    95.01   259.57   165.420         0.60         0.85 115
## [14,]   121.94   317.37   195.360         0.65         0.90 115
## [15,]   151.17   467.88   238.860         0.70         0.95 115
## [16,]   177.29  9541.50   292.040         0.75         1.00 115
```

## 10 CoxTab(): fit cox proportional model for multiple covariates (additive model or separate models)

### 10.1 single covariate

```
CoxTab(data=sample.data, tte="OS", cens="OS.event", bep='BEP', var='Sex' )
```

```
##           HR      CI.low CI.high p-value n.trt n.ref
## Sex (M/F) "0.85" "0.42" "1.72"  "0.65"  "128" "129"
```

### 10.2 multiple covariates, ITT

Additive model will be fitted.

```
kable(
CoxTab(data=sample.data,tte="OS", cens="OS.event", var=c('Sex',"Country","Age"),
var.class=c("categorical","categorical","numeric"))
)
```

	HR	CI.low	CI.high	p-value	n.trt	n.ref
Sex (M/F)	0.72	0.48	1.08	0.11	277	273
Country (USA/Other)	0.65	0.36	1.2	0.17	127	198
Country (Western Europe/Other)	0.77	0.5	1.2	0.24	225	198
Age	1.01	0.99	1.03	0.43		

Simplify: automatically learn var.class from column class:

```
kable(
CoxTab(data=sample.data,tte="OS", cens="OS.event", var=c('Sex',"Country","Age"))
)
```

	HR	CI.low	CI.high	p-value	n.trt	n.ref
Sex (M/F)	0.72	0.48	1.08	0.11	277	273
Country (USA/Other)	0.65	0.36	1.2	0.17	127	198
Country (Western Europe/Other)	0.77	0.5	1.2	0.24	225	198
Age	1.01	0.99	1.03	0.43		

### 10.3 BEP

```
kable(
CoxTab(data=sample.data,tte="OS", cens="OS.event", var=c('Sex',"Country","Age"),
var.class=c("categorical","categorical","numeric"), bep="BEP")
)
```

	HR	CI.low	CI.high	p-value	n.trt	n.ref
Sex (M/F)	0.85	0.42	1.73	0.65	128	129
Country (USA/Other)	1.17	0.4	3.38	0.77	76	57
Country (Western Europe/Other)	1.11	0.44	2.83	0.83	124	57
Age	1.01	0.98	1.04	0.48		

## 10.4 Reorder Sex: M as reference

```
kable(
CoxTab(data=sample.data,tte="OS", cens="OS.event", var=c('Sex',"Country","Age"),
var.class=c("ordered.factor","categorical","numeric"),
ordered.factor.levels.list=list(Sex=c("M","F")),bep="BEP")
)
```

	HR	CI.low	CI.high	p-value	n.trt	n.ref
Sex (F/M)	1.12	0.68	1.85	0.65	129	128
Country (USA/Other)	1.17	0.4	3.38	0.77	76	57
Country (Western Europe/Other)	1.11	0.44	2.83	0.83	124	57
Age	1.01	0.98	1.04	0.48		

## 10.5 Fit separate models for each variable instead of fitting an additive model

```
kable(
CoxTab(data=sample.data,tte="OS", cens="OS.event", var=c('Sex',"Country","Age"),
additive=FALSE)
)
```

	HR	CI.low	CI.high	p-value	n.trt	n.ref
Sex (M/F)	0.71	0.48	1.07	0.099	277	273
Country (USA/Other)	0.67	0.37	1.22	0.19	127	198
Country (Western Europe/Other)	0.77	0.5	1.2	0.25	225	198
Age	1.01	0.99	1.02	0.54		

## 11 StatSummary()

### 11.1 Continuous outcome

#### 11.1.1 Create summary statistics

```
StatSummary(outcome.var = input$OS, treatment.var = input$Arm,  
            placebo.code = "CTRL", active.code = "TRT",  
            outcome.class = "continuous")
```

```
## Effect.Size      Lower      Upper      P Mean.Placebo  
## 0.79113129 0.19464513 1.38761744 0.00942898 5.77854999  
## Mean.Active  
## 6.56968128
```

#### 11.1.2 Add a covariate variable

```
StatSummary(outcome.var = input$OS, treatment.var = input$Arm,  
            placebo.code = "CTRL", active.code = "TRT",  
            outcome.class = "continuous", covariate.var = input$Sex)
```

```
## Effect.Size      Lower      Upper      P Mean.Placebo  
## 0.795751256 0.199865806 1.391636707 0.008954882 5.563721382  
## Mean.Active  
## 6.359472638
```

#### 11.1.3 Add return.fit = TRUE to return a table of summary statistics

```
StatSummary(outcome.var = input$OS, treatment.var = input$Arm,  
            placebo.code = "CTRL", active.code = "TRT",  
            outcome.class = "continuous", covariate.var = input$Sex,  
            return.fit = TRUE)
```

```
## [[1]]  
## Effect.Size      Lower      Upper      P Mean.Placebo  
## 0.795751256 0.199865806 1.391636707 0.008954882 5.563721382  
## Mean.Active  
## 6.359472638  
##  
## [[2]]  
##  
## Call:  
## lm(formula = outcome.var[subgroup.var] ~ covariate.var[subgroup.var] +  
##     treatment.var[subgroup.var])  
##  
## Coefficients:  
##              (Intercept)      covariate.var[subgroup.var]M  
##                5.5637                0.4204  
## treatment.var[subgroup.var]TRT  
##                0.7958
```

## 11.2 Binary outcome

### 11.2.1 Create summary statistics

```
StatSummary(outcome.var = input$OS.event, treatment.var = input$Arm,
             placebo.code = "CTRL", active.code = "TRT",
             outcome.class = "binary")
```

```
## Effect.Size      Lower      Upper      P Rsp.Placebo
## -0.04846512 -0.12251455  0.02558430  0.19901328  0.20879121
## Rsp.Active      N.Placebo      N.Active nRsp.Placebo nRsp.Active
## 0.16032609 182.00000000 368.00000000 38.00000000 59.00000000
```

## 11.3 Survival variable

### 11.3.1 Create summary statistics

```
StatSummary(outcome.var = input[, c("OS", "OS.event")],
             treatment.var = input$Arm, placebo.code = "CTRL",
             active.code = "TRT", outcome.class = "survival")
```

```
## CTRL.events      CTRL.n      CTRL.MST TRT.events      TRT.n      TRT.MST
## 38.00000000 182.00000000 14.8829569 59.00000000 368.00000000 NA
## Effect.Size      Lower      Upper      P
## 0.6667451 0.4432032 1.0030365 0.0517253
```

### 11.3.2 Add a covariate variable

```
StatSummary(outcome.var = input[, c("OS", "OS.event")],
             treatment.var = input$Arm, placebo.code = "CTRL",
             active.code = "TRT", outcome.class = "survival",
             covariate.var = input$Sex)
```

```
## CTRL.events      CTRL.n      CTRL.MST TRT.events      TRT.n
## 38.00000000 182.00000000 14.88295688 59.00000000 368.00000000
## TRT.MST Effect.Size      Lower      Upper      P
## NA 0.63926120 0.42390908 0.96401541 0.03277593
```

### 11.3.3 Add a stratification variable

```
StatSummary(outcome.var = input[, c("OS", "OS.event")],
             treatment.var = input$Arm, placebo.code = "CTRL",
             active.code = "TRT", outcome.class = "survival",
             covariate.var = input$Sex, strat.factor.var = input$Age)
```

```
## CTRL.events      CTRL.n      CTRL.MST TRT.events      TRT.n
## 38.00000000 182.00000000 14.88295688 59.00000000 368.00000000
## TRT.MST Effect.Size      Lower      Upper      P
## NA 0.60116855 0.37830923 0.95531273 0.03128551
```

## 12 LogRankTab(): log rank test for subgroup analysis

```
kable(  
  LogRankTab(data=input,tte="PFS",cens="PFS.event",var="Arm")  
)
```

	CTRL	TRT
	N=182	N=368
Patients with event	120 (65.9%)	202 (54.9%)
Patients without event	62 (34.1%)	166 (45.1%)
Time to event (month)		
Median (KM)	3.25	5.82
95% CI Median	(2.83;4.14)	(5.49;6.8)
25% and 75%-ile	1.61;6.47	2.89;11.73
Range (inc. cens.)	0.03 to 12.32	0.03 to 13.8
p-value (Log-Rank Test)		0
Hazard Ratio		0.54
95% CI		(0.43;0.67)

## 13 BoxPlot(): advanced box plot function

### 13.1 Generate example dataset

```
example <- data.frame( y=c(rnorm(30)+10, rnorm(4)+20, rnorm(15)+15, NA),  
  time=c(rep("t2", 30), rep("t4",4), rep("t1", 15), "t3"),  
  grp=sample(1:3, 50, TRUE), sex=sample(1:2, 50, TRUE))  
head(example)
```

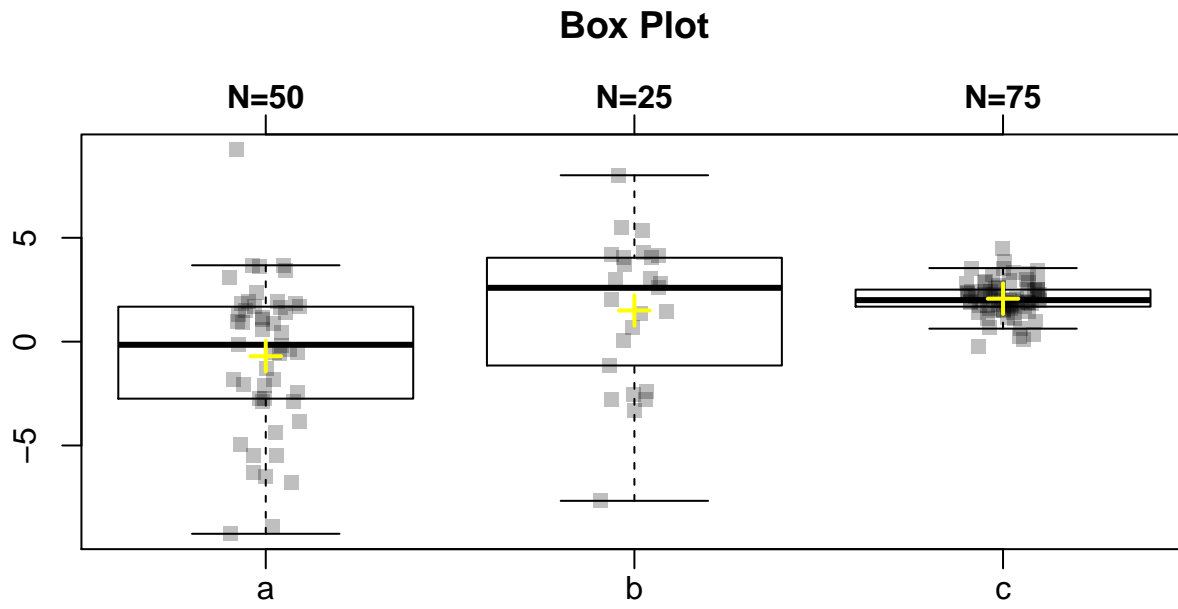
```
##           y time grp sex  
## 1  9.638091  t2   1   2  
## 2  9.954756  t2   2   1  
## 3 10.958392  t2   2   1  
## 4 10.735416  t2   2   2  
## 5 11.269834  t2   3   2  
## 6 10.484932  t2   1   2
```

```
str(example)
```

```
## 'data.frame':   50 obs. of  4 variables:  
## $ y : num  9.64 9.95 10.96 10.74 11.27 ...  
## $ time: Factor w/ 4 levels "t1","t2","t3",...: 2 2 2 2 2 2 2 2 2 2 ...  
## $ grp : int   1 2 2 2 3 1 2 1 1 3 ...  
## $ sex : int   2 1 1 2 2 2 2 2 1 2 ...
```

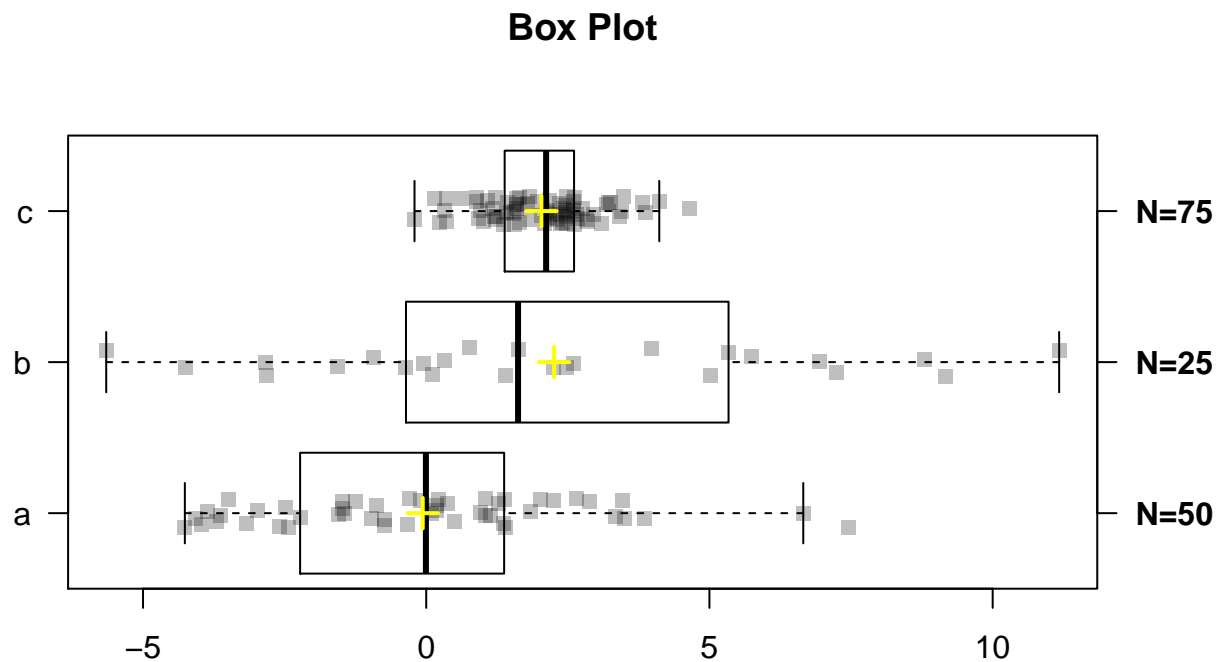
### 13.2 Specify data as named list

```
BoxPlot(list(a=rnorm(50,,3), b=rnorm(25,1,4), c=rnorm(75,2,1)))
```



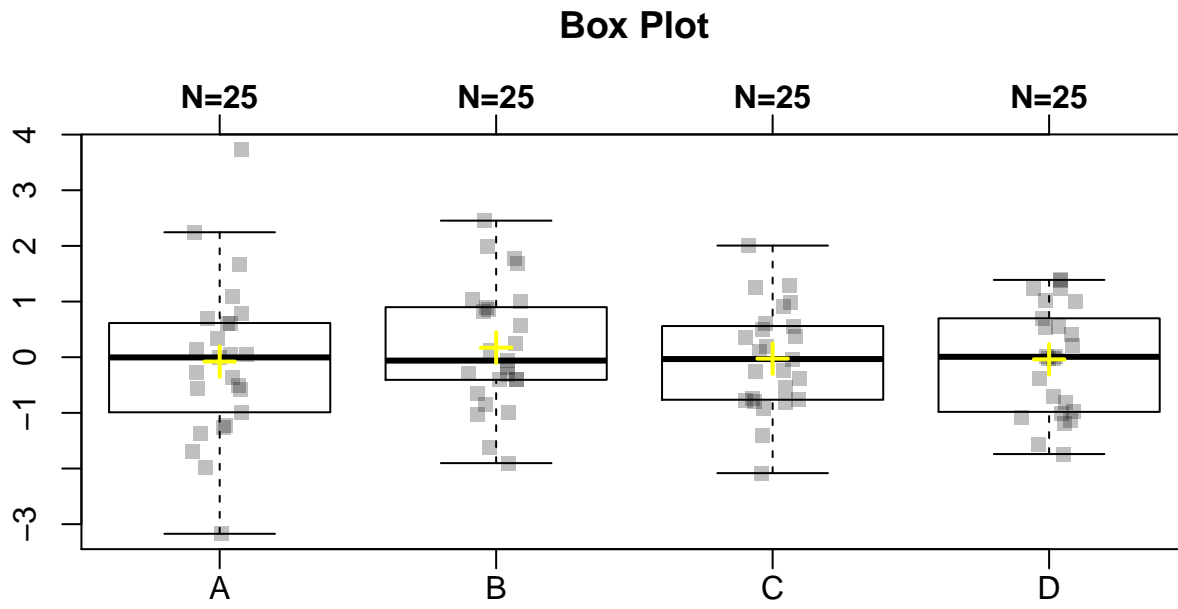
### 13.3 Same plot, now horizontally plotted

```
BoxPlot(list(a=rnorm(50,,3), b=rnorm(25,1,4), c=rnorm(75,2,1)), horizontal=TRUE,
        Xaxis=list(las=2, hadj=2), Xaxis2=list(las=2, hadj=-.25))
```



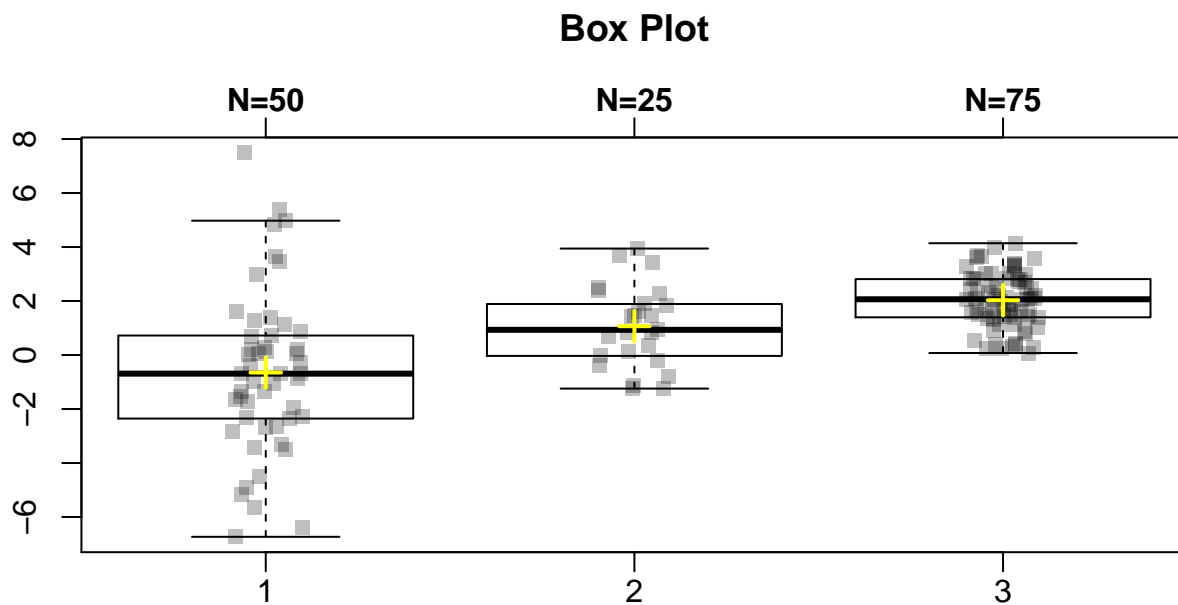
### 13.4 Specify data as numeric matrix

```
BoxPlot(matrix(rnorm(100), ncol=4, dimnames=list(NULL, LETTERS[1:4])) )
```



### 13.5 Specify data as numeric (unnamed) vectors

```
BoxPlot(rnorm(50,,3), rnorm(25,1,2), rnorm(75,2,1))
```

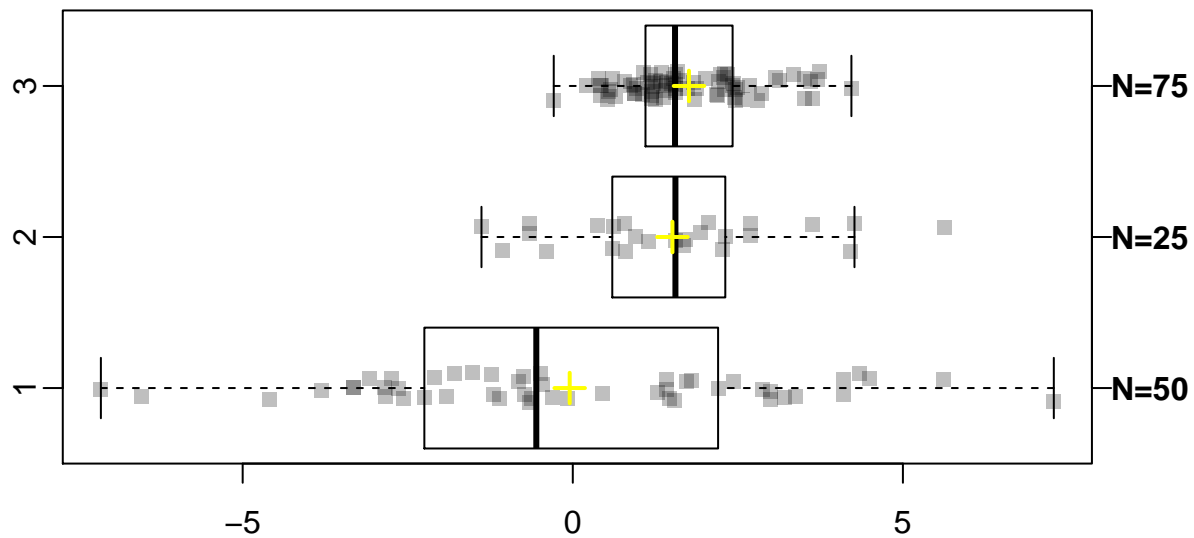


### 13.6 ... horizontally plotted (no axis-label rotation done here)

```
BoxPlot(rnorm(50,,3), rnorm(25,1,2), rnorm(75,2,1), horizontal=TRUE)
```



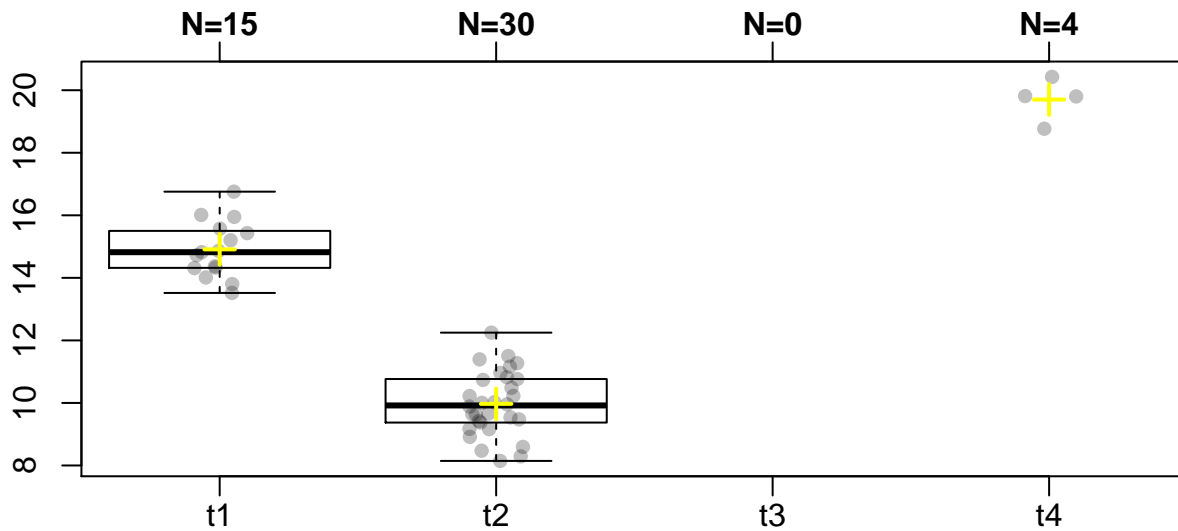
## Box Plot



13.7 Plot values 'y' according to time 'time' (factor levels are automatically ordered as e.g. function sort does)

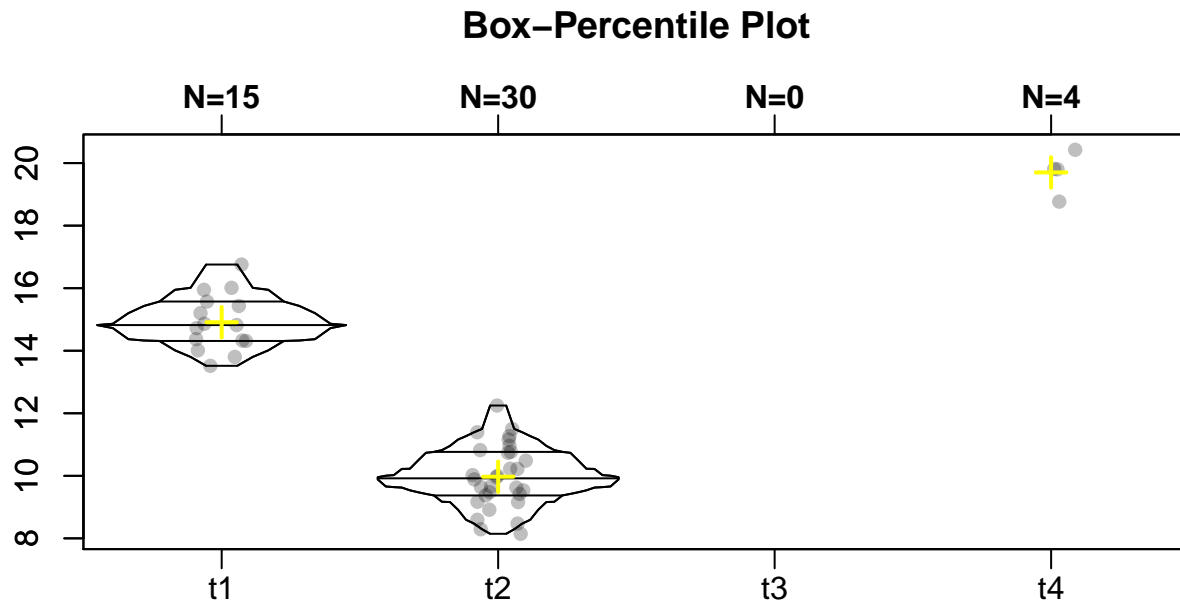
```
BoxPlot(example, y~time, sc.pch=16)
```

## Box Plot



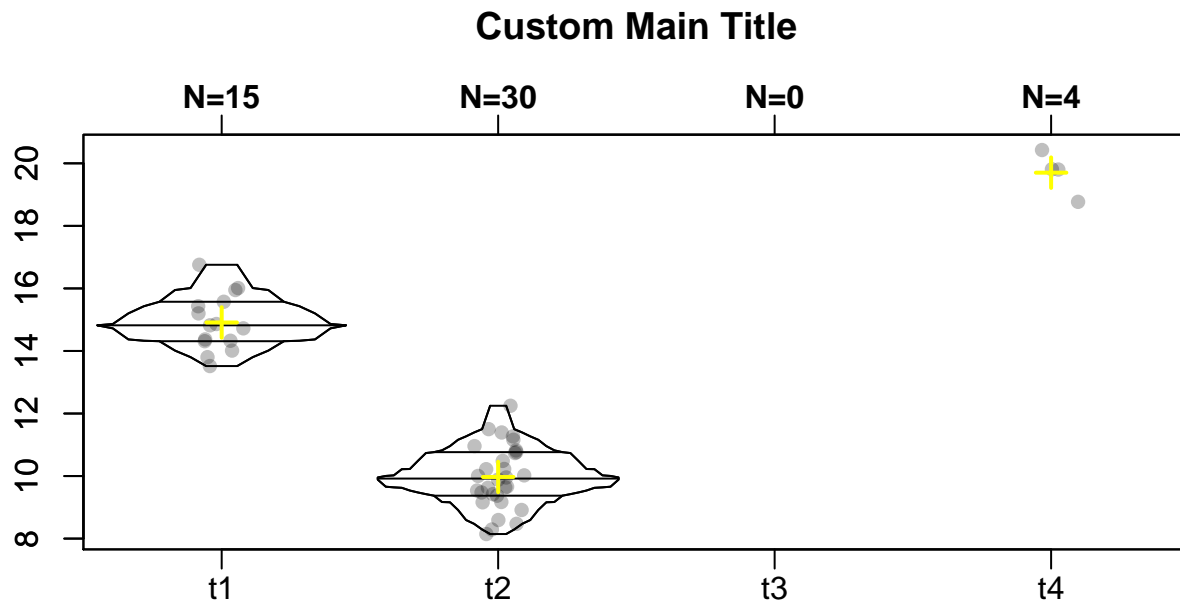
13.8 Now as box-percentile plot

```
BoxPlot(example, y~time, sc.pch=16, box.type="bp")
```



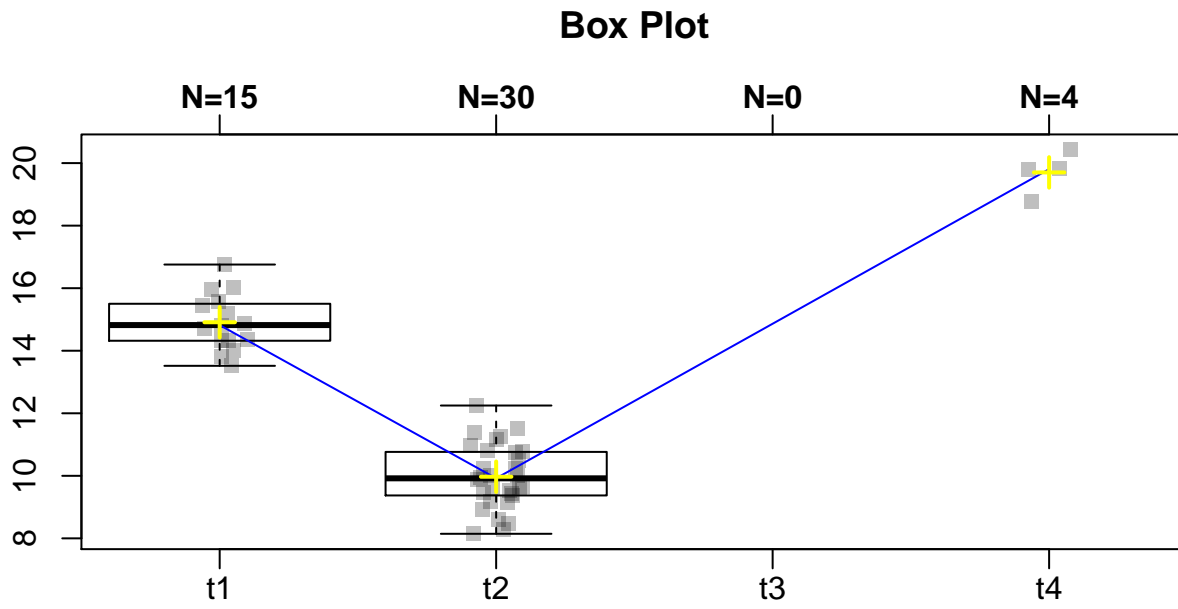
#### 13.9 With custom main title

```
BoxPlot(example, y~time, sc.pch=16, box.type="bp", Title=list(main="Custom Main Title"))
```



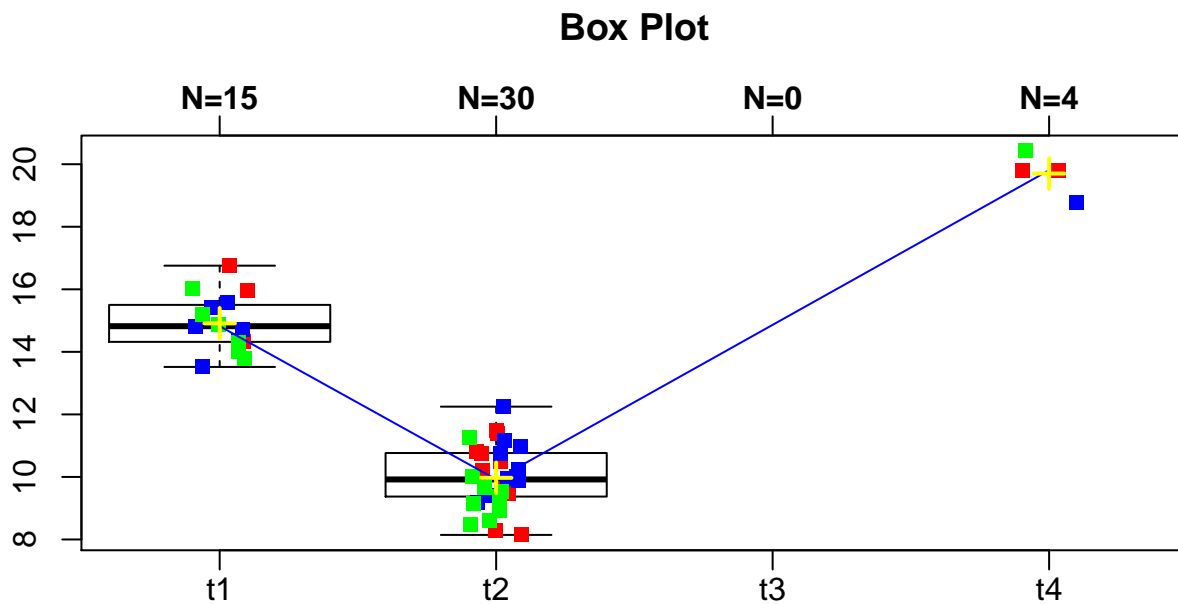
#### 13.10 The same plot with trend line (connects either means or medians) and Y-axis label

```
BoxPlot(example, y~time, trend="median", ylab="Y-Axis Label")
```



13.11 Use an addition grouping variable to color points in the stripchart

```
BoxPlot(example, y~time, trend="median", ylab="Y-Axis Label",
        sc.col=c("red", "blue", "green")[example$grp] )
```



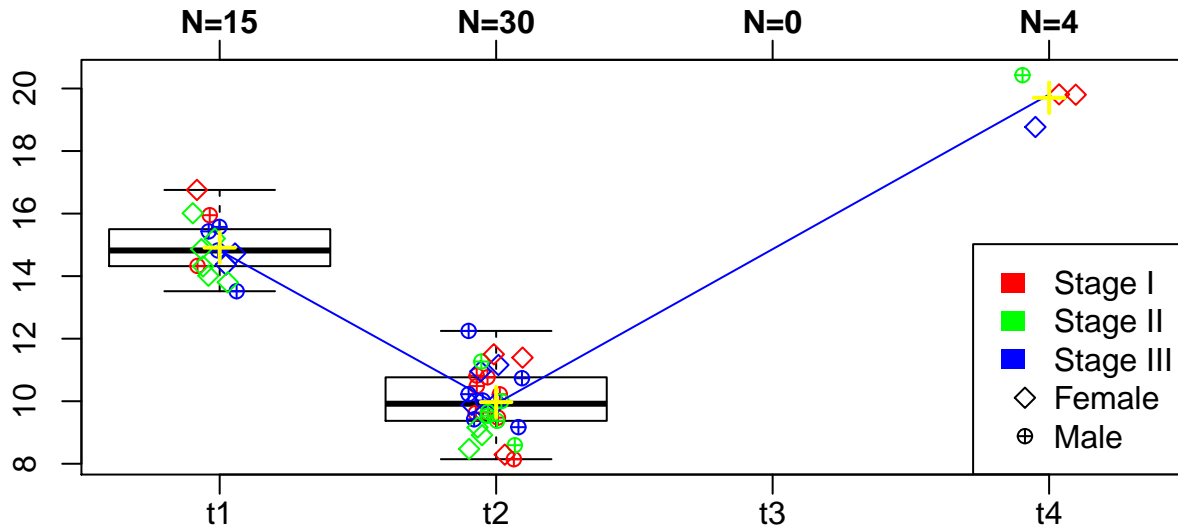
13.12 Use yet another grouping factor for plotting symbols in the stripchart

```
BoxPlot(example, y~time, trend="median", ylab="Y-Axis Label",
        sc.col=c("red", "blue", "green")[example$grp],
        sc.pch=c(5,10)[example$sex] )

legend("bottomright", fill=c("red", "green", "blue", "white", "white"),
```

```
legend=c("Stage I", "Stage II", "Stage III", "Female", "Male"),
pch=c(-1, -1, -1, 5, 10), border=NA)
```

### Box Plot



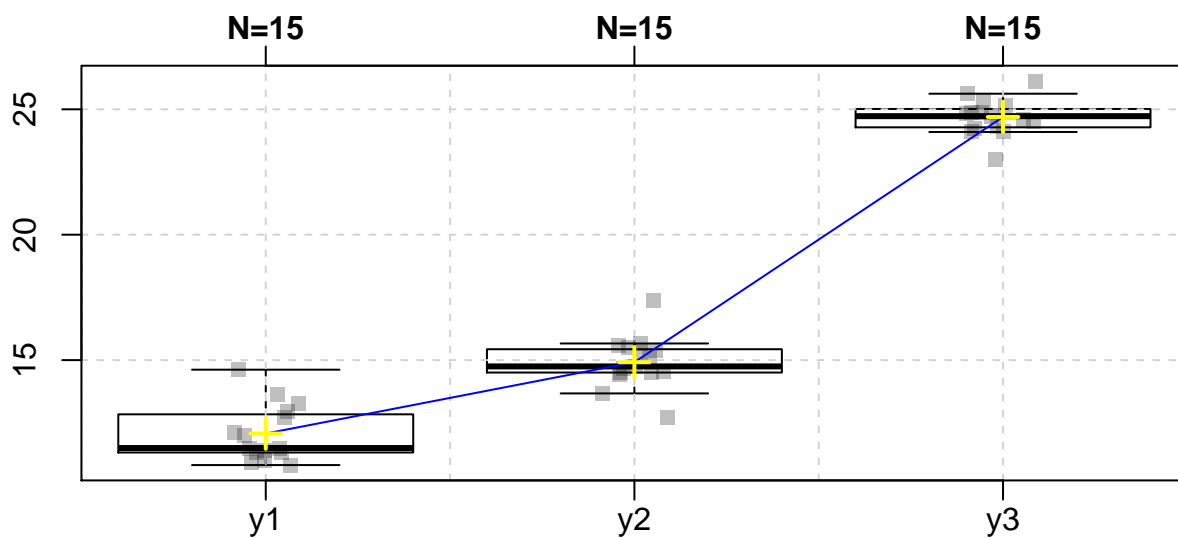
#### 13.13 Generate new dataset with different structure

```
example2 <- data.frame(y1=12+rnorm(15), y2=15+rnorm(15), y3=25+rnorm(15))
```

#### 13.14 Boxplot with trend lines (mean) and grid added to the plot

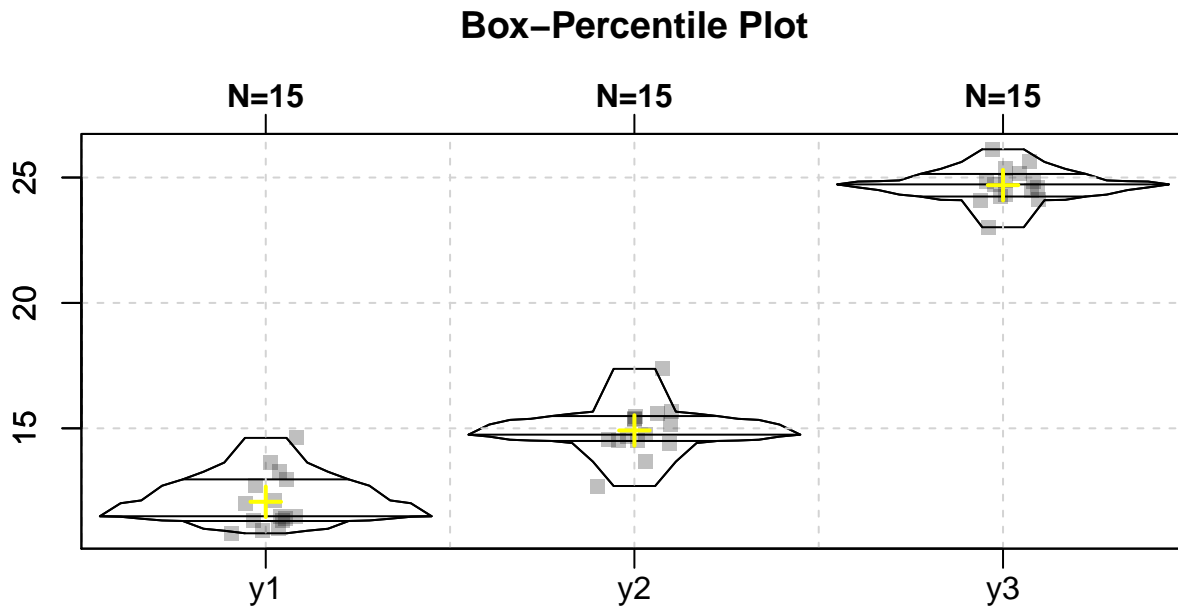
```
BoxPlot(example2, var=c("y1", "y2", "y3"), Grid=TRUE, trend="mean")
```

### Box Plot



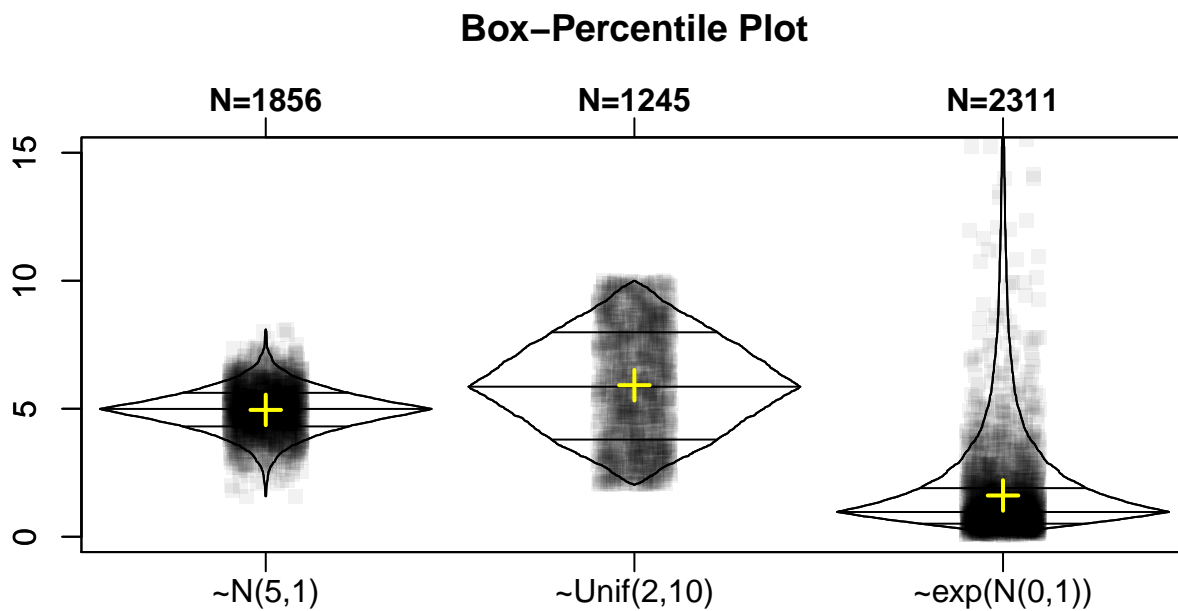
### 13.15 Now as box-percentile plot

```
BoxPlot(example2, var=c("y1", "y2", "y3"), box.type="bp", Grid=TRUE)
```



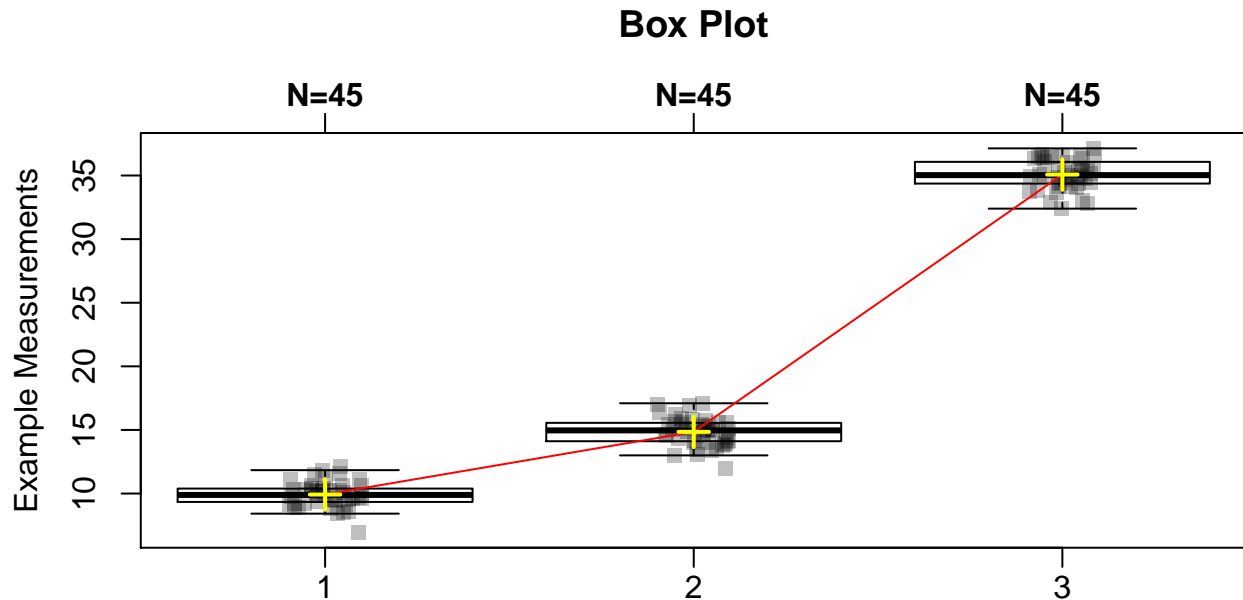
### 13.16 Use yet another way to specify the data

```
BoxPlot(rnorm(1856, 5), runif(1245, 2,10), exp(rnorm(2311)), sc.col=as.rgb("black", .05),
        box.type="bp", Xaxis=list(labels=c("~N(5,1)", "~Unif(2,10)", "~exp(N(0,1))")), ylim=c(0,15))
```



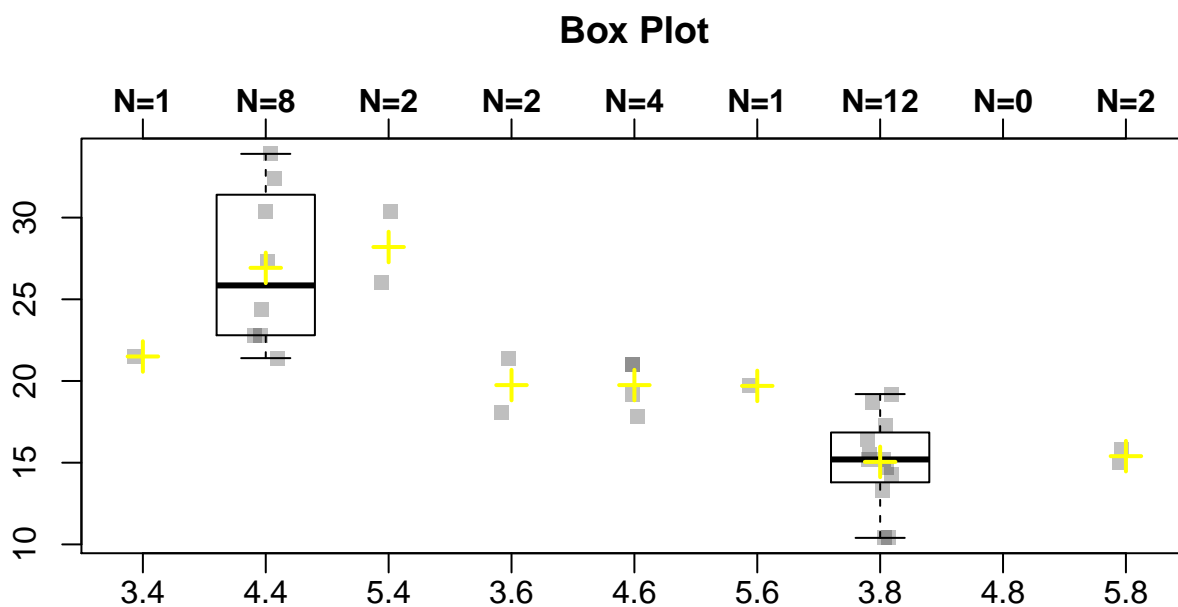
### 13.17 Specifying data as numeric matrix and using a trend-line connecting the means

```
mat <- matrix(c(rep(10,45), rep(15,45), rep(35,45))+rnorm(135), ncol=3)
BoxPlot(mat, trend="mean", trend.col="red", Ylabel=list(text="Example Measurements"))
```



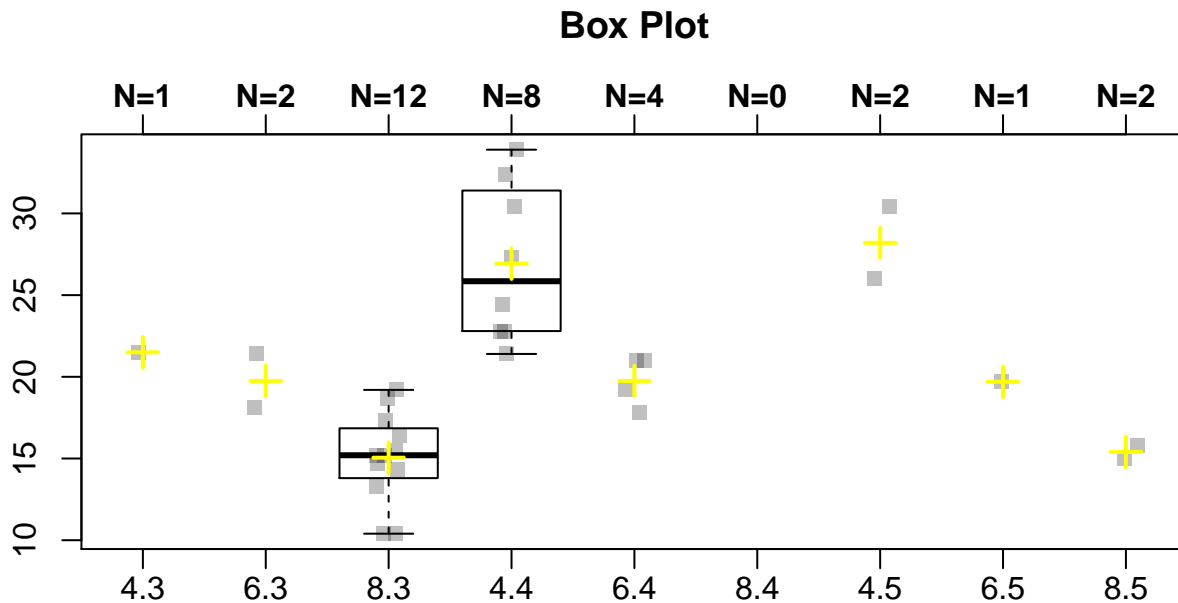
### 13.18 Multiple grouping factors can be specified via the formula interface which is exemplified using the mtcars dataset, of interest is miles per gallon (mpg) depending on number of gears and on the number of cylinders

```
data(mtcars)
BoxPlot(mtcars, mpg~gear:cyl)
```



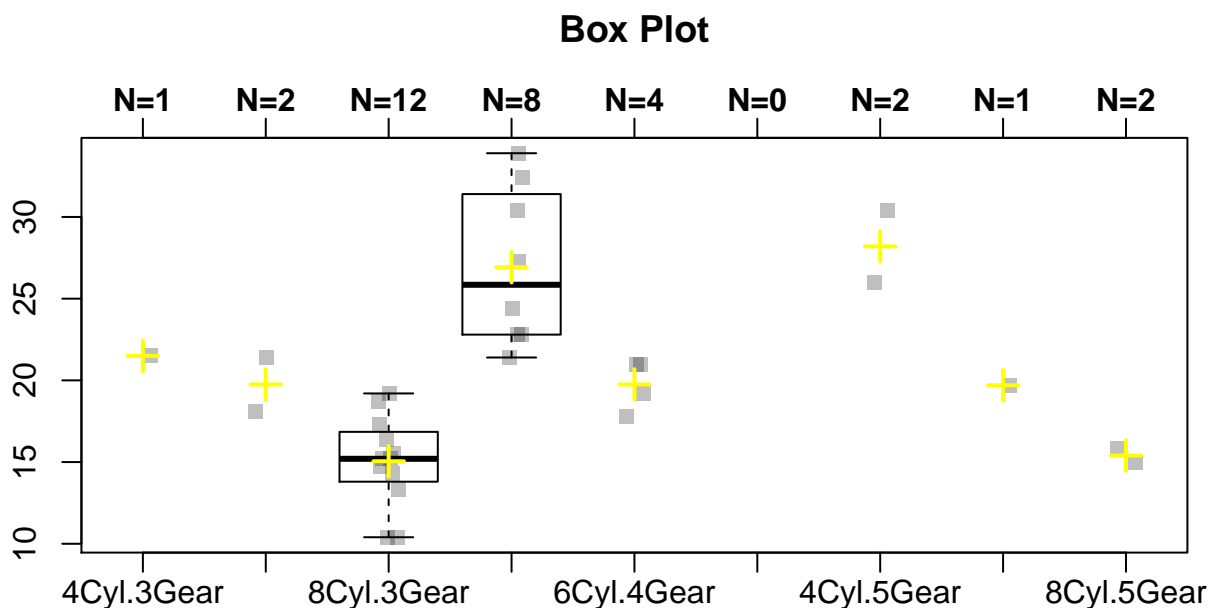
13.19 Now 'cyl' is explicitly nested within 'gear' which changes the ordering of combined grouping factors (which is identical to using formula 'mpg~cyl:gear').

```
BoxPlot(mtcars, mpg~cyl %in% gear)
```



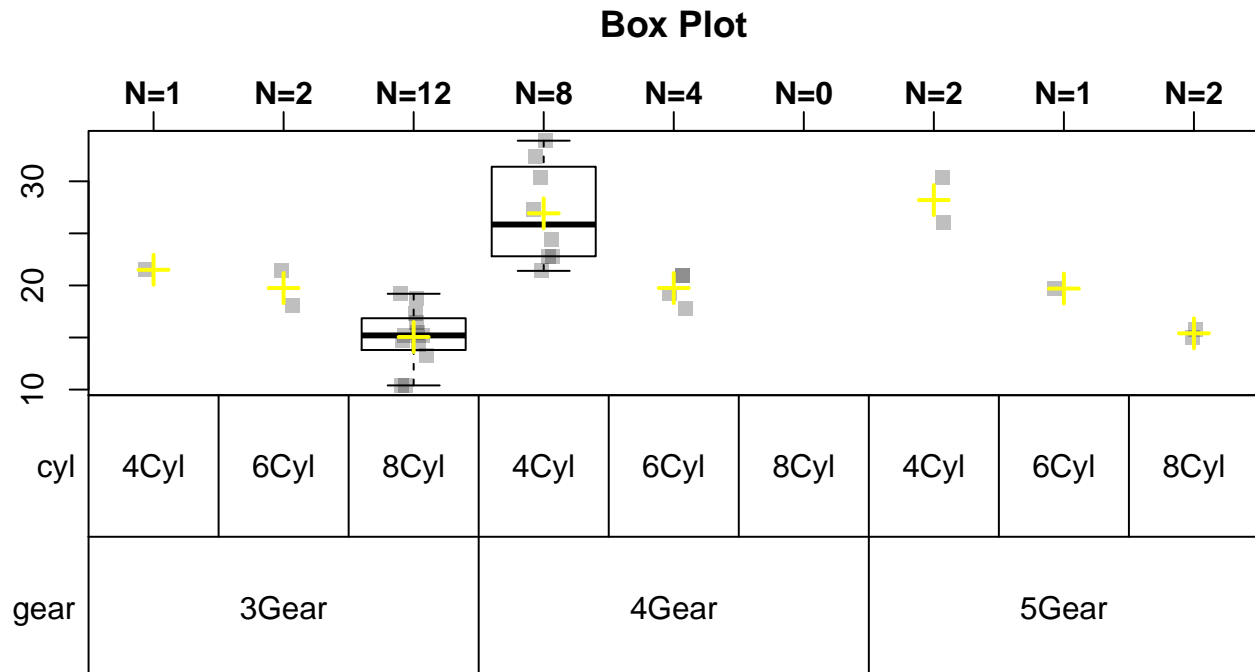
13.20 More meaningful group-labels are best specified using custom factor-level names (increase width of the plot window)

```
dat <- mtcars
dat$cyl <- factor(dat$cyl, levels=c(4,6,8), labels=c("4Cyl", "6Cyl", "8Cyl"))
dat$gear <- factor(dat$gear, levels=c(3,4,5), labels=c("3Gear", "4Gear", "5Gear"))
BoxPlot(dat, mpg~cyl %in% gear)
```



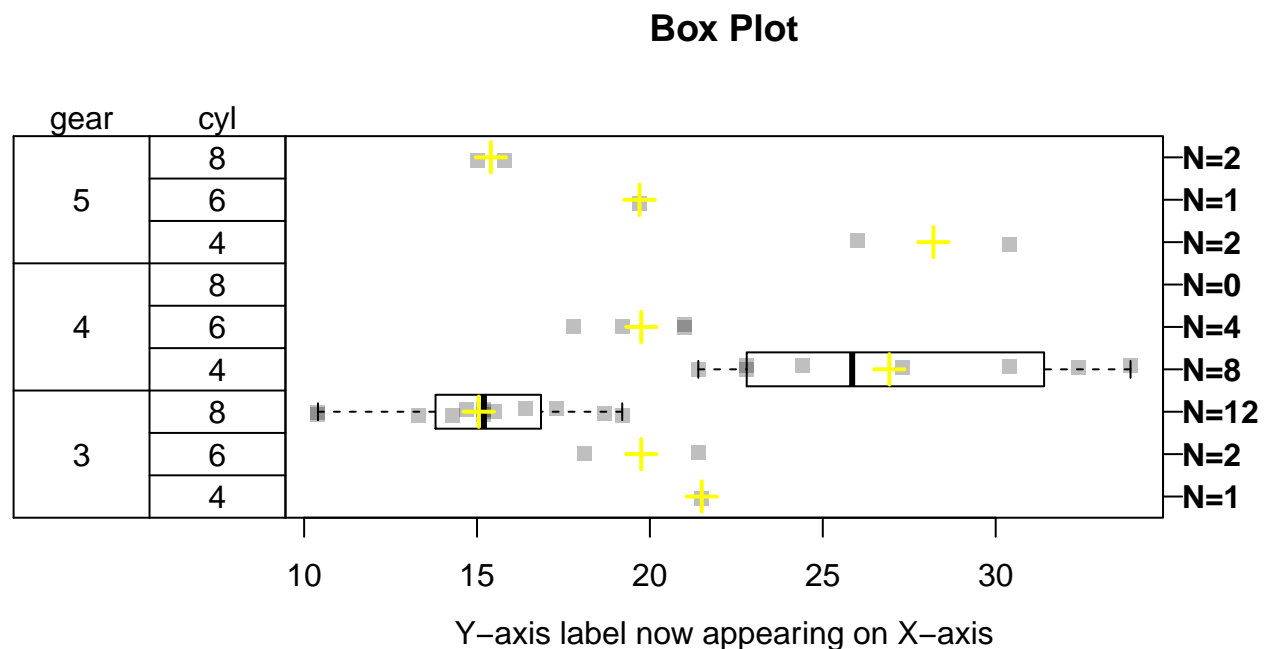
### 13.21 One can use a table as Xaxis label representing the factor-level combination defining sub-classes

```
BoxPlot(dat, mpg~cyl:gear, XaxisTab=list(),mar=c(8,3,5,1))
```



### 13.22 With the original factor levels and as horizontal plot

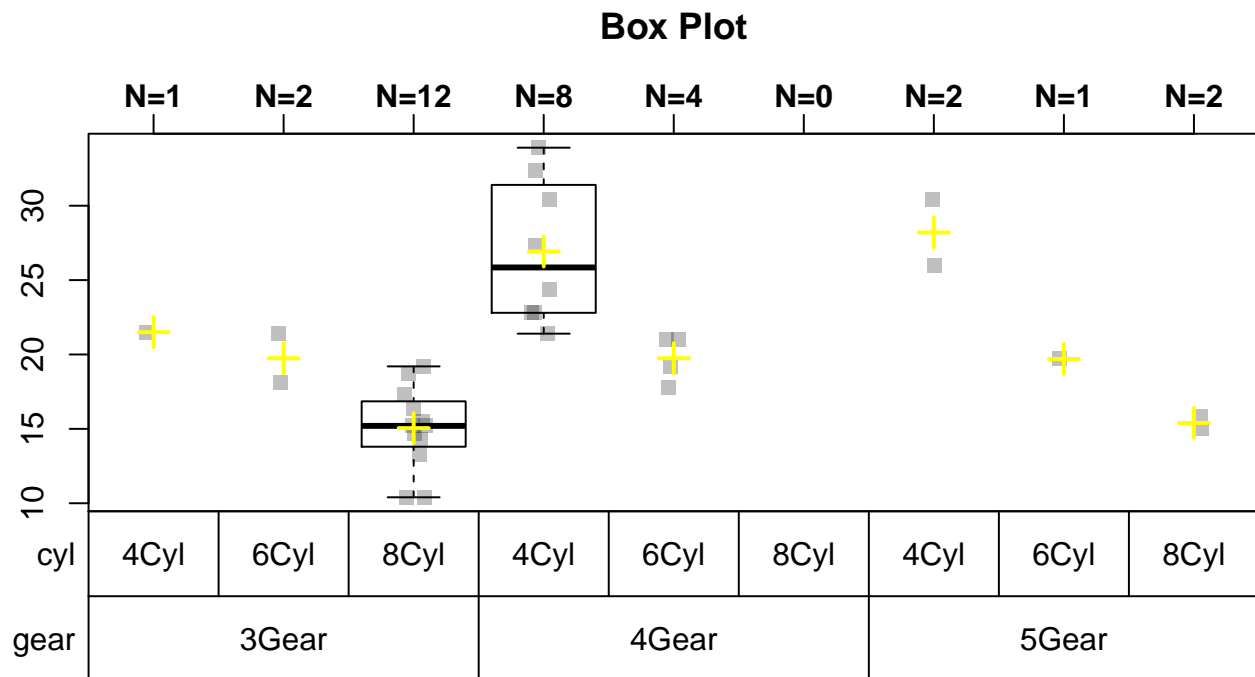
```
BoxPlot(mtcars, mpg~cyl:gear, XaxisTab=list(), mar=c(5,8,5,4),
        horizontal=TRUE, Ylabel=list(text="Y-axis label now appearing on X-axis"))
```





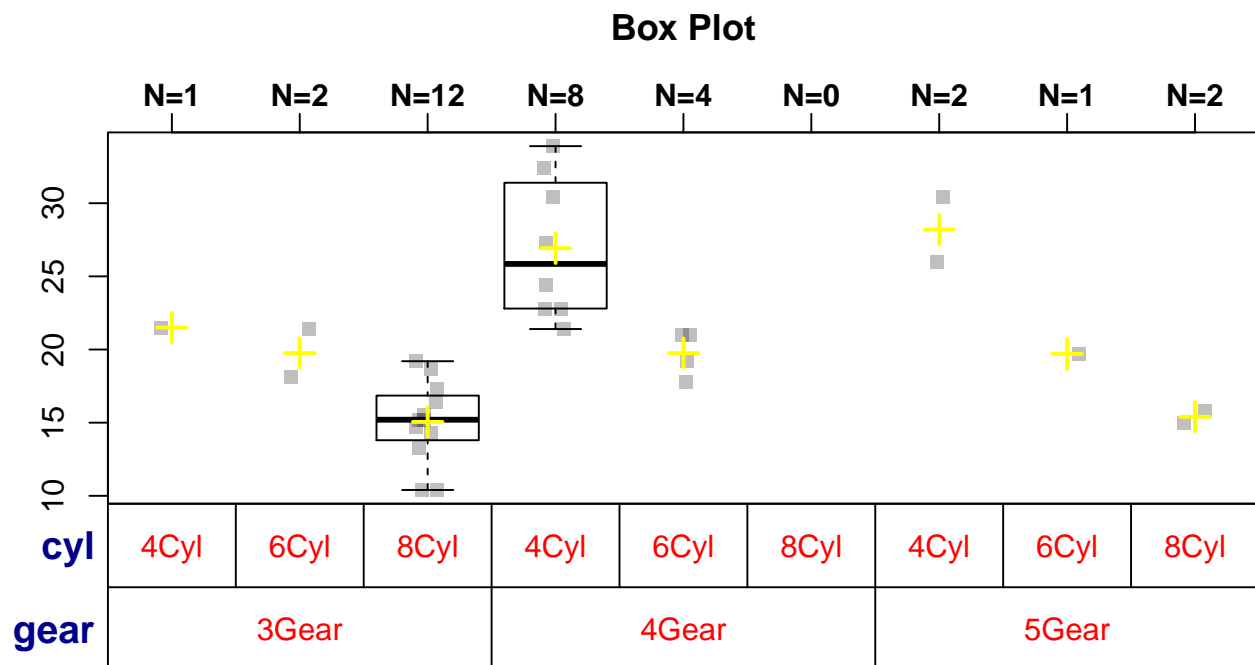
### 13.23 Using smaller bottom margin will result in smaller table height

```
BoxPlot(dat, mpg~cyl:gear, XaxisTab=list(font=2, col="darkblue", cex=1.25), mar=c(5,3,5,1))
```



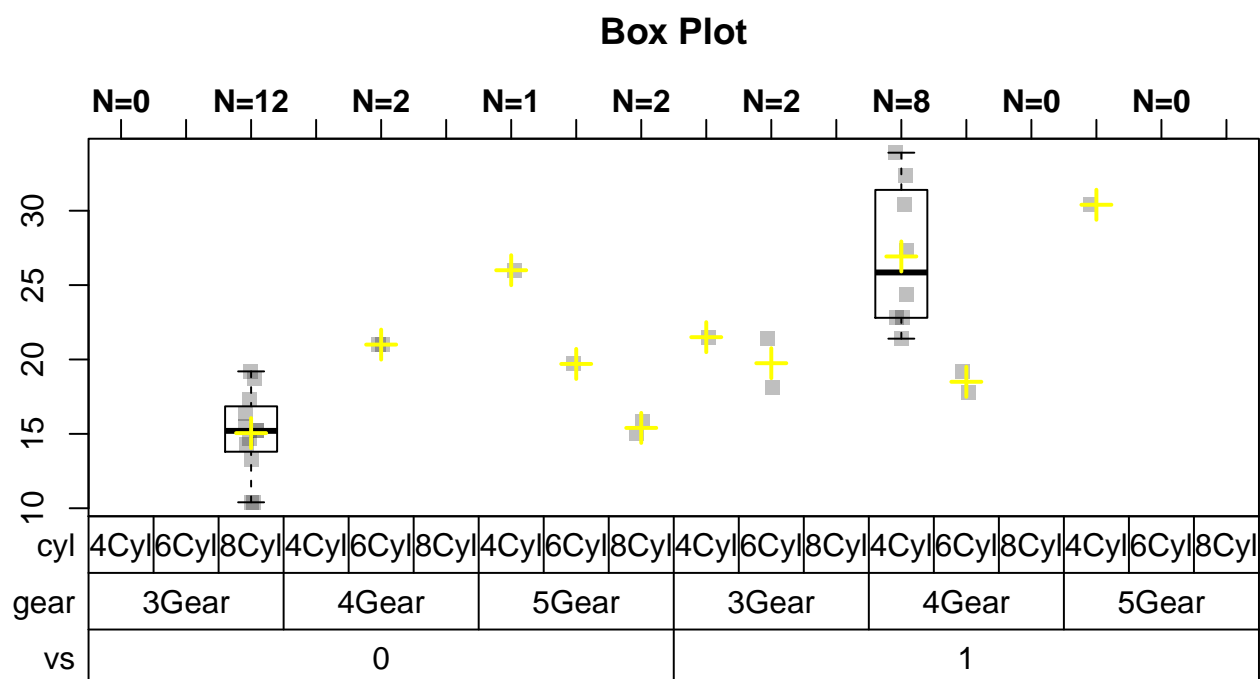
### 13.24 One can use different font-settings for rownames and cells of the table

```
BoxPlot(dat, mpg~cyl:gear, XaxisTab=list(Label=list(font=2, col="darkblue", cex=1.25),
Text=list(col="red")), mar=c(5,3,5,1))
```



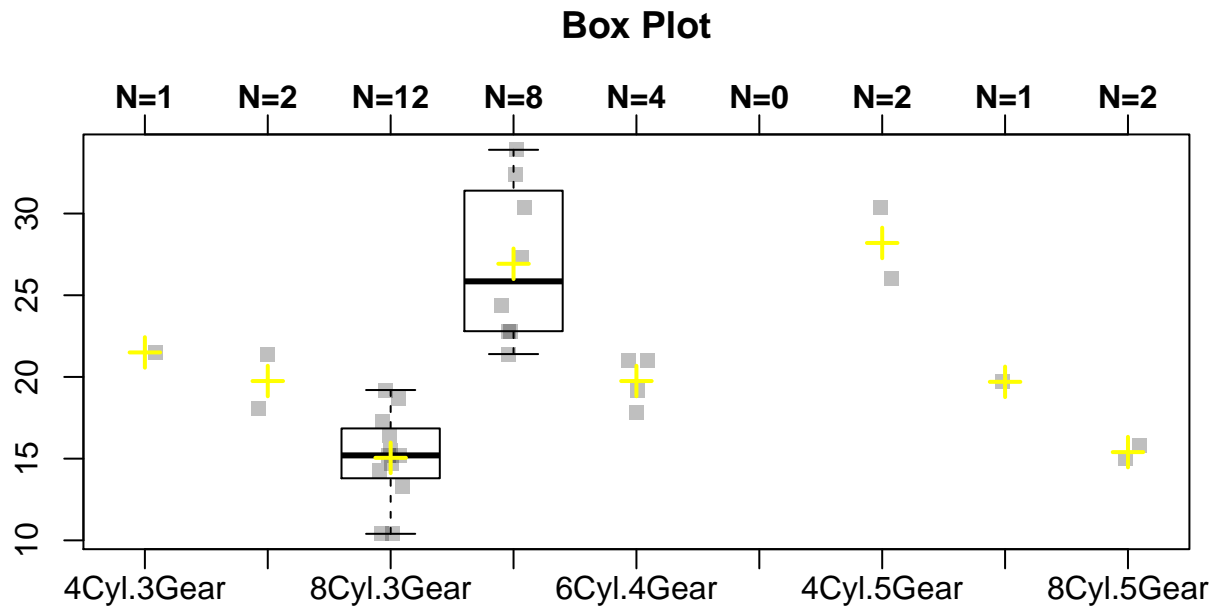
### 13.25 Use more crossed factors

```
BoxPlot(dat, mpg~cyl:gear:vs, XaxisTab=list(), mar=c(5,3,5,1))
```



13.26 Alternatively one can use the 'Xaxis' argument, but the ordering of these labels is not checked which is not important for automatically generated group-labels as shown in the previous example

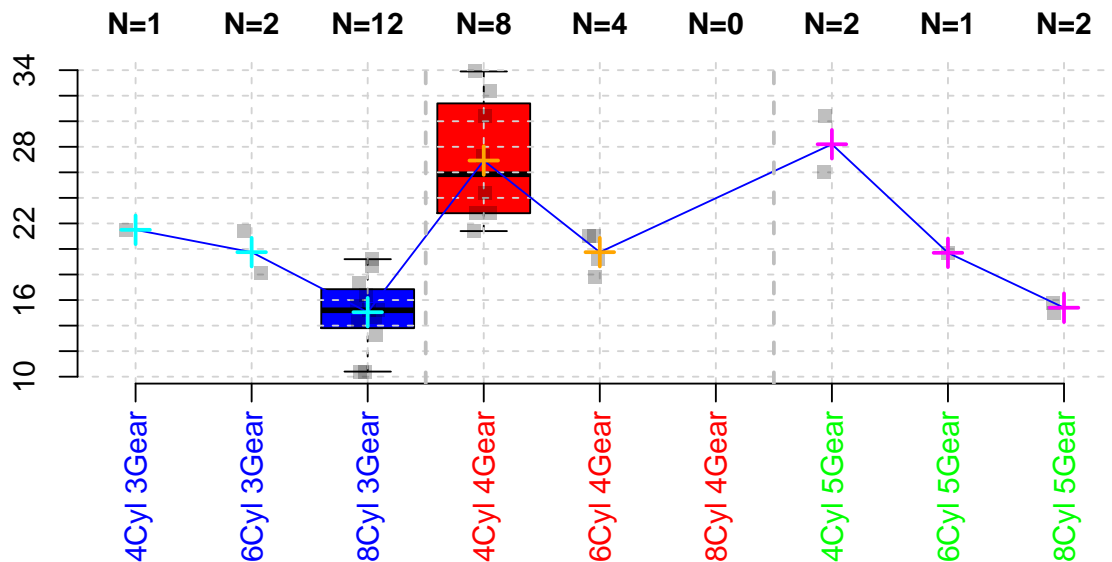
```
BoxPlot(mtcars, mpg~cyl %in% gear,
        Xaxis=list(labels=paste(rep(c("4Cyl", "6Cyl", "8Cyl"),3),
                                c(rep("3Gear",3), rep("4Gear",3), rep("5Gear",3))), sep=".")))
```



13.27 The same plot with some fancy options

```
BoxPlot(dat, mpg~cyl %in% gear, Title=list(main="Miles per Gallon by Number of Gears",
  col.main="Green", cex.main=2.5), vline=c(3.5, 6.5), vl.lty=2, vl.col="gray", vl.lwd=2,
  Xaxis=list(labels=NA, at=1:9, tick=TRUE), col=c(rep("blue", 3), rep("red", 3), rep("green", 3)),
  Xaxis2=list(tick=FALSE), Yaxis=list(at=seq(10,34,2)), Grid=list(x=1:9, y=seq(10,34,2)),
  Xlabel=list(text=paste(rep(c("4Cyl", "6Cyl", "8Cyl"),3), c(rep("3Gear",3), rep("4Gear",3), rep("5Gear",3))),
    at=1:9, las=2, adj=1, line=0.75, col=c(rep("blue", 3), rep("red", 3), rep("green", 3)),
    mean.col=c(rep("cyan", 3), rep("orange", 3), rep("magenta", 3)), Box=FALSE, trend="mean", mar=c(1,1,1,1))
```

## Miles per Gallon by Number of Gears

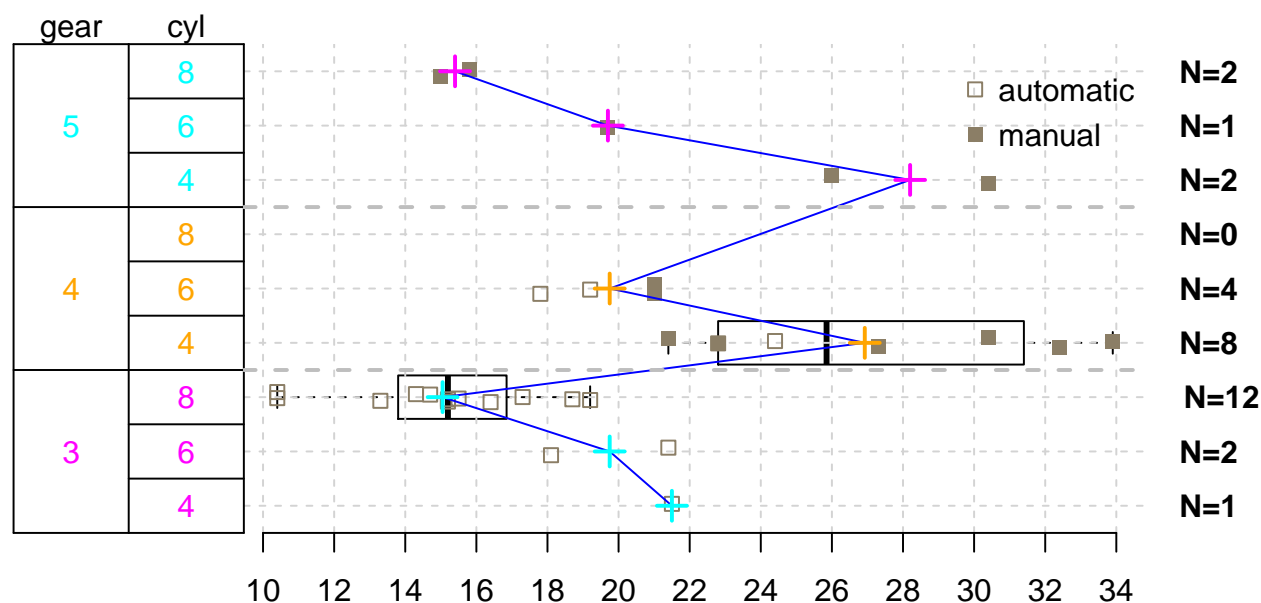


## 13.28 Horizontal fancy plot

```
BoxPlot(mtcars, mpg~cyl %in% gear, Title=list(main="Miles per Gallon by Number of Gears", col.main="#84
      vline=c(3.5, 6.5), vl.lty=2, vl.col="gray", vl.lwd=2,
      Axis2=list(tick=FALSE, las=2, hadj=-.25), Yaxis=list(at=seq(10,34,2)), Grid=list(x=1:9, y=seq(
      mean.col=c(rep("cyan", 3), rep("orange", 3), rep("magenta", 3)), Box=FALSE, trend="mean",
      mar=c(3, 7, 4, 4), horizontal=TRUE, sc.pch=c(0, 15)[dat$am+1], sc.col="wheat4",
      AxisTab=list(Text=list(col=c(rep("cyan", 3), rep("orange", 3), rep("magenta", 3)))) )

legend(x="topright", pch=c(0, 15), legend=c("automatic", "manual"), box.lty=0, col="wheat4")
```

### Miles per Gallon by Number of Gears



## 14 Longitudinal analysis

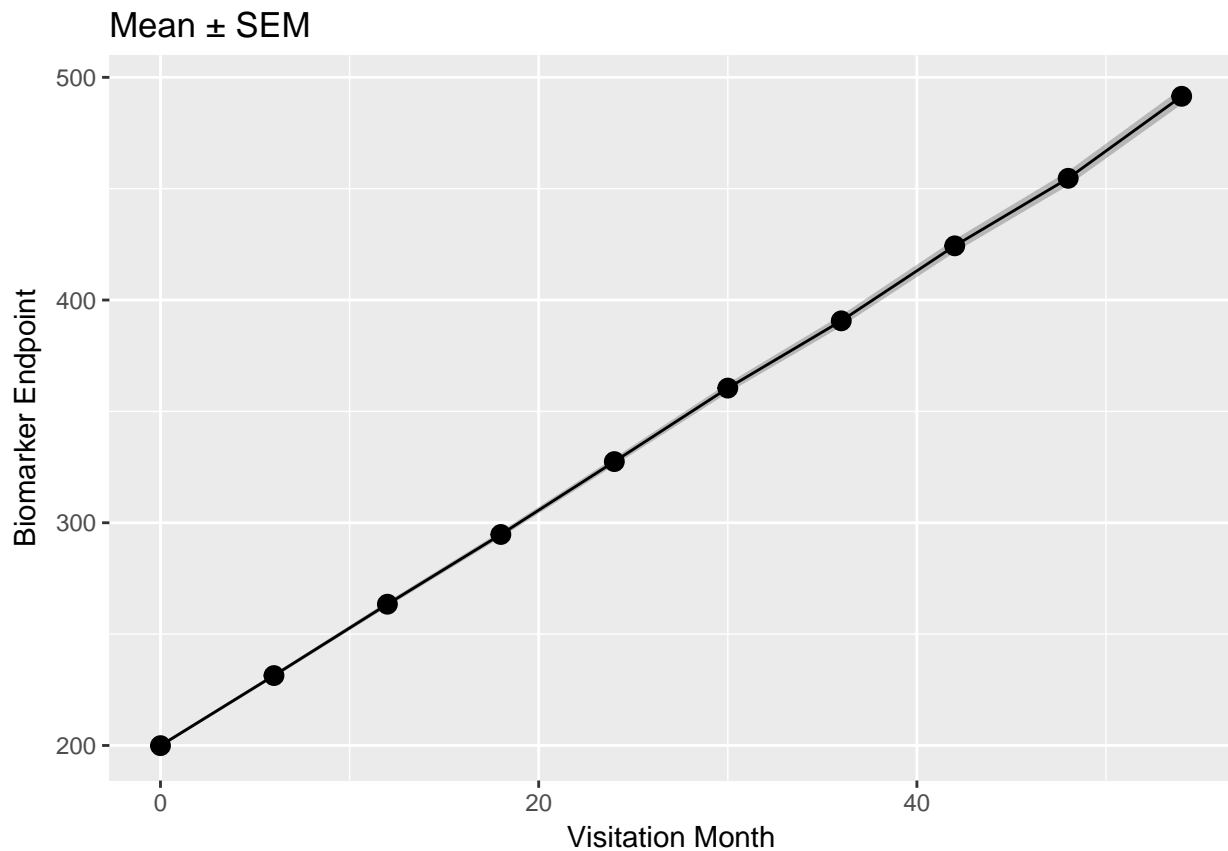
In the context of biomarker analysis, the `PlotLong` function can be used to plot the longitudinal measurement of a biomarker endpoint for one or more patient subpopulations. Here we'll use the packaged dataset `bmkr` (biomarker) which has columns for patient ID (`pid`), treatment (`trt`), gender (`sex`), visitation month (`vm`) and endpoint reading (`ep`)

### 14.1 Biomarker Endpoint Progression

#### 14.1.1 Mean $\pm$ SEM Timecourse

Here we plot the mean and standard error of the mean over time. We can see clearly that the mean biomarker endpoint increases throughout the study.

```
PlotLong(longbmkr, aes(x=vm, y=ep),
      xlab = 'Visitation Month',
      ylab = 'Biomarker Endpoint',
      labs.title = 'Mean  $\pm$  SEM')
```

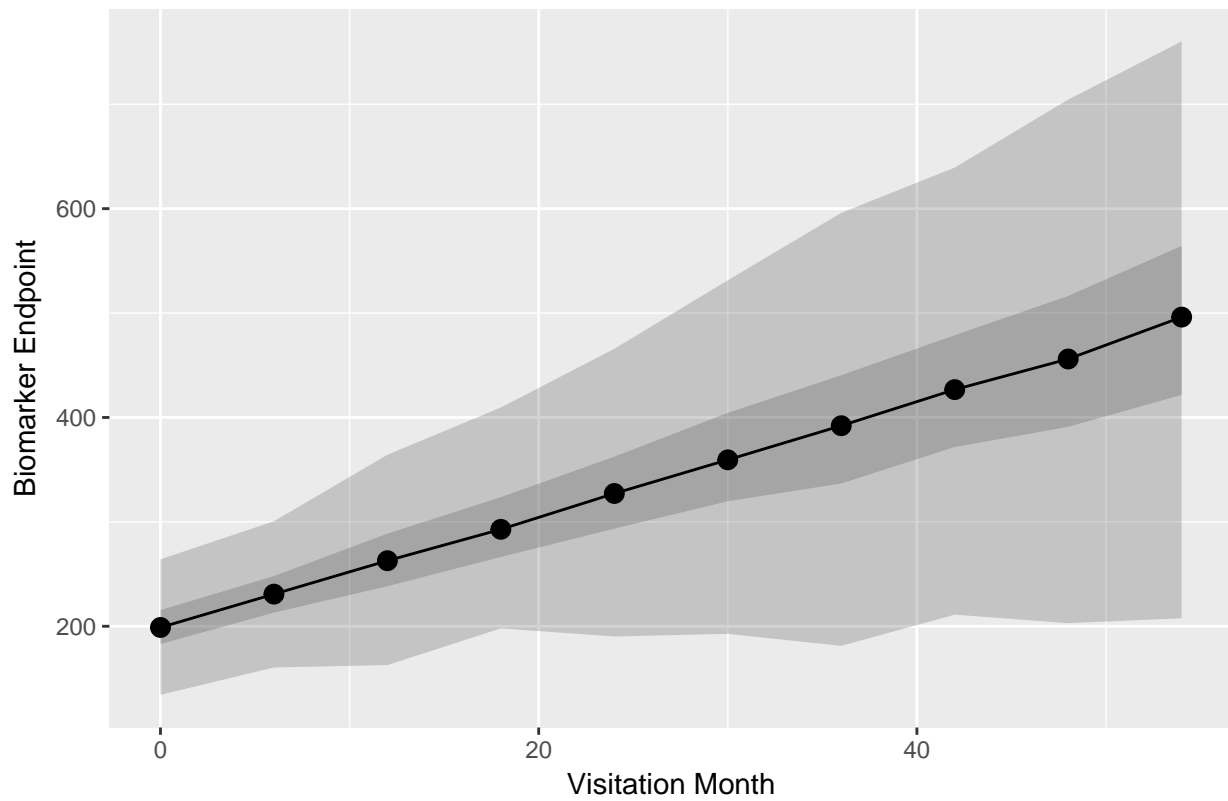


#### 14.1.2 Tukey Hinges and Whiskers

However, we may be more interested in the population distribution progression where we'd want to plot the Tukey boxplot hinges and whiskers over time. From this plot, we can see that although there is a slow rise in population mean, the patient response also becomes more distributed.

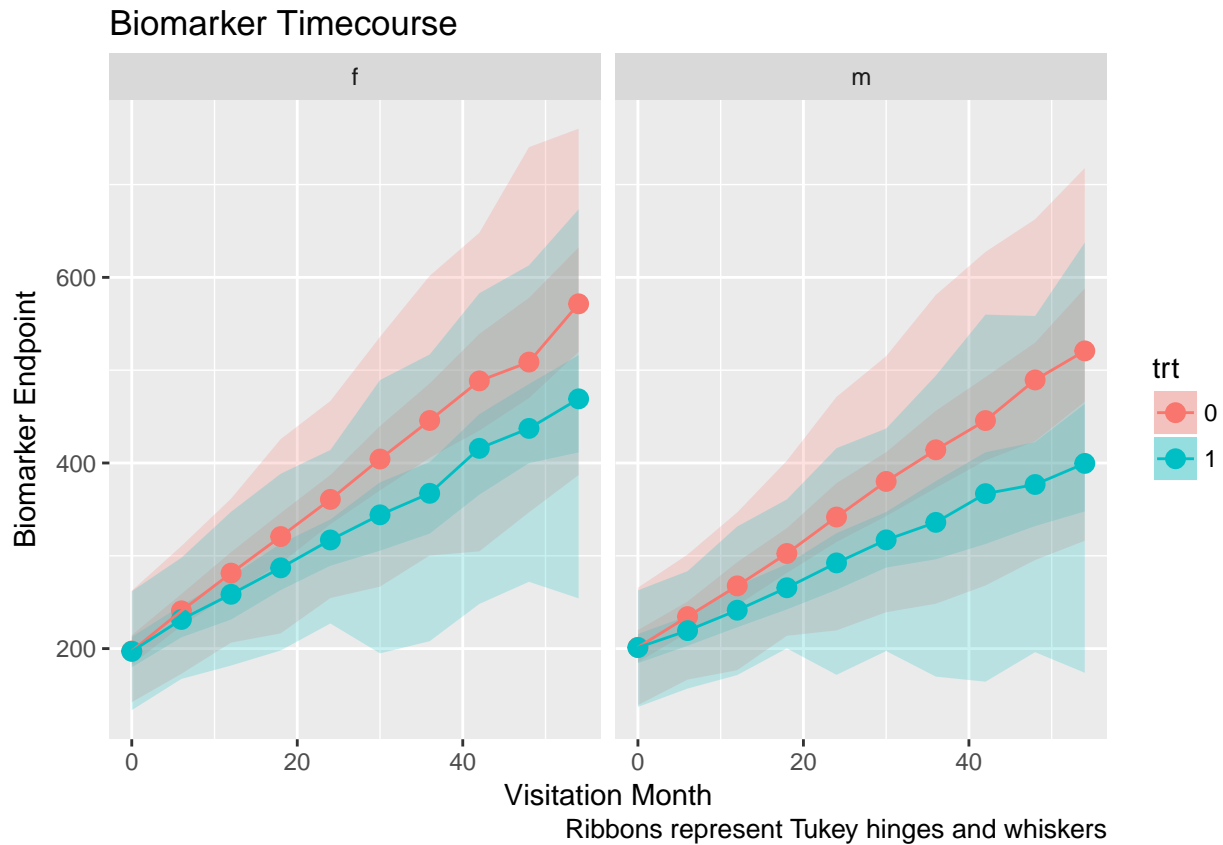
```
PlotLong(longbmr, aes(x=vm, y=ep), fun.data = 'tukey',
         xlab = 'Visitation Month',
         ylab = 'Biomarker Endpoint',
         labs.title = 'Tukey Whiskers and Hinges')
```

## Tukey Whiskers and Hinges



## 14.2 Plotting Subpopulations

```
PlotLong(longbmkr, aes(x=vm, y=ep, group=trt, color=trt, fill=trt),  
  fun.data = 'tukey', facet.fun = . ~ sex,  
  xlab = 'Visitation Month',  
  ylab = 'Biomarker Endpoint',  
  labs.title = 'Biomarker Timecourse',  
  labs.caption = 'Ribbons represent Tukey hinges and whiskers')
```



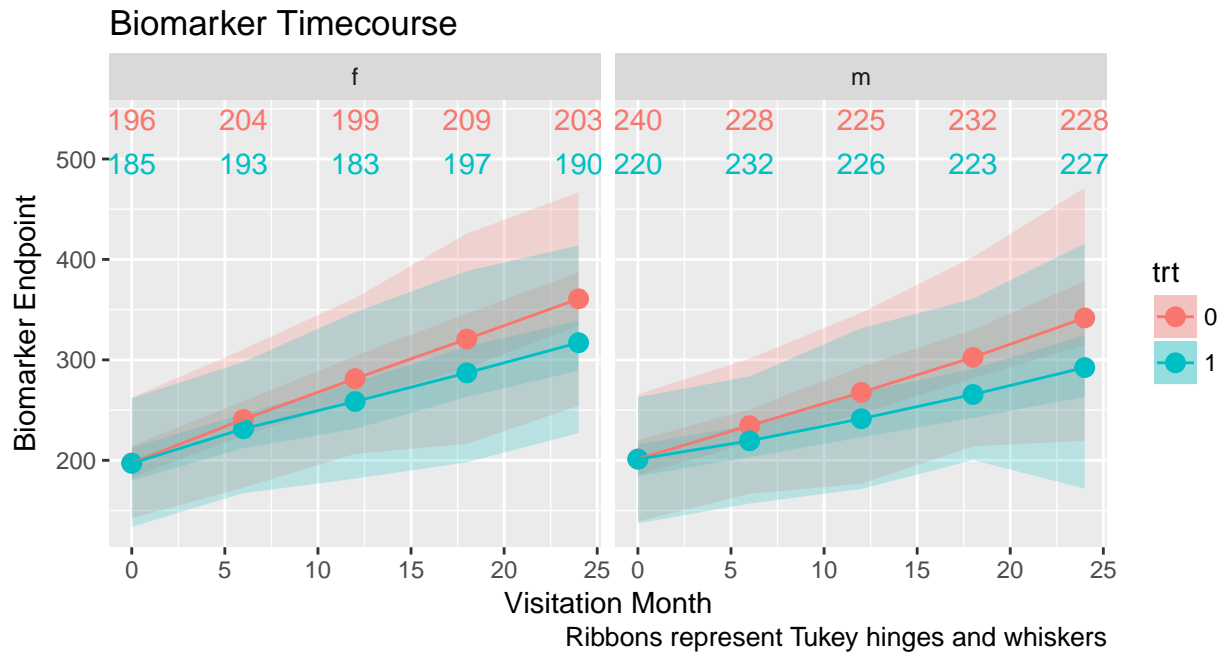
## 14.3 Including sample counts

### 14.3.1 Sample counts as table above plot

```
library(dplyr)

## Warning: package 'dplyr' was built under R version 3.3.2
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
PlotLong(longbmkrr %>% filter(vm <= 24),
  aes(x=vm, y=ep, group=trt, color=trt, fill=trt),
  fun.data = 'tukey', facet.fun = . ~ sex,
  show.counts = 'table',
  xlab = 'Visitation Month',
  ylab = 'Biomarker Endpoint',
  labs.title = 'Biomarker Timecourse',
  labs.caption = 'Ribbons represent Tukey hinges and whiskers')
```

## Warning: package 'bindrcpp' was built under R version 3.3.2



#### 14.4 Plotting Deviations as Errorbars

```
PlotLong(longbmr %>% filter(vm <= 24),
  aes(x=vm, y=ep, group=trt, color=trt, fill=trt),
  fun.data = 'tukey', facet.fun = . ~ sex,
  show.counts = 'table',
  plot.style = 'errorbars',
  xlab = 'Visitation Month',
  ylab = 'Biomarker Endpoint',
  labs.title = 'Biomarker Timecourse',
  labs.caption = 'Ribbons represent Tukey hinges and whiskers')
```



