rtables - Reporting tables with R : : **CHEAT SHEET**

Basics

The **rtables** R package is designed to create and display complex tables with R.

Every rtable layout is constructed starting with **basic_table** and is rendered using **build_table**.

CODE

```
tbl_a <- basic_table() %>%
  split_cols_by("ARM") %>%
  split_rows_by("STRATA1") %>%
  analyze("AGE") %>%
  build_table(adsl)
```

TABLE OUTPUT

	ARM X	ARM Y
A		
Mean	33.32	35.86
B		
Mean	33.65	38.00

Layout & Tabulation

ANALYZE & SUMMARIZE FUNCTIONS

analyze()

analyze_colvars()

summarize_row_groups()

LAYOUT MODIFIERS

append_topleft()

add_colcounts()

add_overall_col()

CUSTOMIZED TABLE CODE

```
basic_table(show_colcounts = TRUE) %>%
  split_cols_by("ARM") %>%
  add_overall_col("TOTAL") %>%
  split_rows_by("BMRKR2",
                split_label = "Biomarker 2 Level",
                label_pos = "topleft") %>%
  summarize_row_groups() %>%
  analyze("AGE", var_labels = "Age (yrs)",
          afun = mean, format = "xx.x") %>%
  analyze("STRATA1", var_labels = "Stratif. Term",
          afun = function(x, .N_col) lapply(
            table(x),
            function(xi) rcell(
              xi * c(1, 1 / .N_col),
              format = "xx (xx.xx%)"
            ))) %>%
  append_topleft("  Attribute") %>%
  build_table(adsl)
```

CUSTOMIZED TABLE OUTPUT

Biomarker 2 Level Attribute	ARM X (N=42)	ARM Y (N=40)	TOTAL (N=82)
LOW	22 (52.4%)	25 (62.5%)	47 (57.3%)
Age (yrs)			
mean	33.5	36.4	35.1
Stratif. Term			
A	9 (21.43%)	12 (30.00%)	21 (25.61%)
B	13 (30.95%)	13 (32.50%)	26 (31.71%)
HIGH	20 (47.6%)	15 (37.5%)	35 (42.7%)
Age (yrs)			
mean	33.5	37.7	35.3
Stratif. Term			
A	10 (23.81%)	9 (22.50%)	19 (23.17%)
B	10 (23.81%)	6 (15.00%)	16 (19.51%)

Customization Options

ANALYZE & SUMMARIZE FUNCTIONS

Argument	Input	Effect on Table
afun/cfun	Analysis function	The function is used to calculate cell values
var_labels	Labels for variables being analyzed	Labels are printed in the leftmost column
format	Format string or function	Format is applied to render cell values
na_str	String to represent NA values	String is printed in place of missing values
inclNAs	TRUE or FALSE	Changes whether records with NA are included in analysis
show_labels	"default", "visible", or "hidden"	var_labels are printed or hidden in the table
indent_mod	Number of spaces to indent by	Current analysis rows are indented
section_div	String to divide split sections by	String is printed between groups defined by current split

Simple Tabulation

Quick tables with **qtable** – an extension of base::table for exploratory work & data summarization.

CODE

```
qtable(
  adsl,
  row_vars = c(
    "STRATA1", "STRATA2"
  ),
  col_vars = c("ARM"),
  avar = "AGE",
  afun = mean
)
```

TABLE OUTPUT

AGE - mean	ARM X (N=42)	ARM Y (N=40)
A		
X	33.00	33.44
Z	34.33	39.25
Y	32.75	34.50
B		
X	34.29	34.50
Z	26.25	47.50
Y	35.75	36.57

Titles & Footers

CODE

```
main_title(tbl_a) <- "My Title"
subtitles(tbl_a) <- c("A subtitle")
main_footer(tbl_a) <- c("A footnote")
prov_footer(tbl_a) <- c("A provenance footer")
fnotes_at_path(tbl_a,
               rowpath = c("STRATA1", "A", "AGE", "Mean"),
               colpath = c("ARM", "ARM X")
) <- "Mean age for arm X"
```

TABLE OUTPUT

```
My Title
A subtitle

                ARM X      ARM Y
A
  Mean      33.32 {1}      35.86
B
  Mean        33.65        38.00
{1} - Mean age for arm X

A footnote
A provenance footer
```

 Your Name • your@email.com • Learn more at the rtables website • Visit the rtables GitHub repository • package version 0.6.6 • Updated: 2024-01

Split Functions

Split functions are used to **add, remove, or transform** the levels of the variable used in a split.

ROW SPLITS

split_rows_by()
split_rows_by_multivar()
split_rows_by_cuts()
split_rows_by_cutfun()
split_rows_by_quartiles()

COLUMN SPLITS

split_cols_by()
split_cols_by_multivar()
split_cols_by_cuts()
split_cols_by_cutfun()
split_cols_by_quartiles()

SPLIT FUNCTIONS

remove_split_levels()
keep_split_levels()
drop_split_levels()
drop_and_remove_levels()
add_overall_level()
add_combo_levels()
reorder_split_levels()
trim_levels_in_group()
trim_levels_to_map()

CODE

```
basic_table() %>%
  split_cols_by(
    "ARM",
    split_fun = remove_split_levels(c("ARM Y"))
  ) %>%
  split_rows_by(
    "STRATA1",
    split_fun = reorder_split_levels(c("B", "A"))
  ) %>%
  analyze("AGE") %>%
  build_table(adsl)
```

TABLE OUTPUT

```
                ARM X
-----------------------
B
  Mean          33.65
A
  Mean          33.32
```

For information on custom split functions, see ?custom_split_funs

Sorting & Pruning

Sorting functions are used to **reorder table rows** according to a given criteria function. Pruning functions are used to **remove table rows** according to a given criteria.

SORTING

sort_at_path()
cont_n_allcols()
cont_n_onecol()

PRUNING

prune_table()
all_zero_or_na()
all_zero()
content_all_zeros_nas()
prune_empty_level()
prune_zeros_only()
low_obs_pruner()

```
tbl <- basic_table() %>%
  split_cols_by("ARM") %>%
  split_rows_by("STRATA2") %>%
  summarize_row_groups() %>%
  build_table(adsl)
```

```
      ARM X          ARM Y
-----------------------------
X   19 (45.2%)   19 (47.5%)
Y   23 (54.8%)   21 (52.5%)
Z    0 (0.0%)    0 (0.0%)
```

```
tbl %>%
  sort_at_path(
    "STRATA2",
    scorefun = cont_n_allcols
  ) %>%
  prune_table()
```

```
      ARM X          ARM Y
-----------------------------
Y   23 (54.8%)   21 (52.5%)
X   19 (45.2%)   19 (47.5%)
```

Access & Modify

ACCESSORS

head(tbl, n)
tail(tbl, n)
tbl[x, y]
cell_values(tbl, rowpath, colpath)
value_at(tbl, rowpath, colpath)
top_left(tbl)

MODIFIERS

tbl[x, y] <- rcell(...)
top_left(tbl) <- "XXX"
rbind(tbl_1, tbl_2)
cbind_rtables(tbl_1, tbl_2)

```
head(tbl_a, 2)
```

```
                ARM X    ARM Y
-----------------------------
A
  Mean          33.32    35.86
```

```
tail(tbl_a, 2)
```

```
                ARM X    ARM Y
-----------------------------
B
  Mean          33.65    38.00
```

```
tbl_a[3:4, 1]
```

```
                ARM X
-----------------------
B
  Mean          33.65
```

```
cell_values(
  tbl_a, rowpath = c(
    "STRATA1", "A", "AGE", "Mean"
  ),
  colpath = c("ARM", "ARM Y"))
```

```
$`ARM Y`
[1] 35.86
```

Rendering

rtables prints output in ASCII format in the R console. **rtable** objects can also be paginated or converted to different output types in the console, or exported to various file types.

R SESSION OUTPUT

Viewer(tbl)
toString(tbl)
as_html(tbl)
tt_to_flextable(tbl)

PAGINATION

```
paginate_table(
  tbl,
  page_type = "letter",
  font_family = "Courier",
  font_size = 8,
  landscape = FALSE
)
```

EXPORT

export_as_docx(tbl, "tbl.docx")
export_as_pdf(tbl, "tbl.pdf")
export_as_rtf(tbl, "tbl.rtf")
export_as_tsv(tbl, "tbl.tsv")
export_as_txt(tbl, "tbl.txt")

 Your Name • your@email.com • Learn more at the rtables website • Visit the rtables GitHub repository • package version 0.6.6 • Updated: 2024-01