

atable package: Usage

Armin Ströbel

August 13, 2020

Contents

1	Context	1
2	Usage	2
2.1	Apply atable to datasets::ToothGrowth	2
2.2	Apply atable to datasets::mtcars	3
2.3	Extract specific values from the table	4
2.4	Language localisation	6
2.5	Word format	7
2.6	HTML format	7
2.7	Console	8
2.8	Mockup tables	8
2.9	Blocks	10
3	Scale of measurement, classes and atable	10
4	Modifying atable	10
4.1	Replace existing methods	12
4.2	Add new methods	13

List of Tables

1	ToothGrowth analysed by atable	3
2	mtcars analysed by atable	5
3	Localised atable	7
4	mockup table of the mtcars analysis	9
5	Blocking of the mtcars analysis	11
6	Classes and atable	11
7	Modified atable	14
8	atable with added methods for class Date	16

1 Context

The atable package supports the analysis and reporting of controlled clinical trials. Data of clinical trials can be stored in data.frames with rows representing 'patients' and columns representing 'measurements' on these patients or characteristics of the trial design like location or time point of measurement. Generally these data.frames will have some hundred rows and some dozen columns. The columns have different purposes:

- Grouping columns contain the treatment the patient received, e.g. new treatment, control group or placebo.
- Splitting columns contains strata of the patient, e.g. demographic data like age, gender or time point of measurement.
- Target columns are the actual measurements of interest, directly related to the objective of the trial. In the context of ICH E9 [ICH \(1999\)](#) these columns are called 'endpoints'.

The task is the comparison of the target columns between the groups, separately for every split column. This is often the first step of clinical trial analysis to get an impression of the distribution of data. The `atable` package solves this task by applying descriptive statistics and hypothesis tests and arranges the results in a table ready for printing.

Reporting of clinical trials is such a frequent task that guidelines have been written which recommend certain properties of clinical trial reports [Moher et al. \(2010\)](#). In particular Item 17a of CONSORT states that “Trial results are often more clearly displayed in a table rather than in the text”. And Item 15 suggests: “a table showing baseline demographic and clinical characteristics for each group”. The `atable` package is specifically designed to comply with these two items.

2 Usage

This sections contains examples for copy and paste for those readers in TL;DR-mode. The examples were created with RStudio in a Rnw-file and compiled to pdf with knitr [Xie \(2018\)](#) and \LaTeX [Mittelbach et al. \(2004\)](#). See folder `doc/inst/` for the Rnw-file of this vignette. The `atable` package only produces tables; it does not produce printable documents. To get a printable document, `atable`'s output must still be converted to other formats with e. g. `Hmisc::latex` [Harrell Jr et al. \(2018\)](#), `officer::body_add_table` [Gohel \(2018b\)](#) and `flextable::regulartable` [Gohel \(2018a\)](#), see examples below.

2.1 Apply `atable` to `datasets::ToothGrowth`

`datasets::ToothGrowth` contains data on tooth length (`len`) depending on three dose levels (`dose`) and two delivery methods of vitamin C (`supp` with levels orange juice or ascorbic acid) in 60 guinea pigs. The design of this experiment is a controlled trial. We use `atable` to test if tooth length depends on the delivery methods, separately for each dose level. See [table 1](#) for the results. This table satisfies the requirements of the CONSORT statement Item 17a [Moher et al. \(2010\)](#).

```
# apply atable
the_table <- atable::atable(ToothGrowth,
                             target_cols = "len",
                             group_col = "supp",
                             split_cols = "dose",
                             format_to = "Latex")

# send to LaTeX
Hmisc::latex(the_table,
              file = "",
              title = "",
              label = "tab:ToothGrowthatable",
              caption = "ToothGrowth analysed by atable.",
```

Table 1: ToothGrowth analysed by atable.

Group	OJ	VC	p	stat	Effect Size (CI)
0.5					
Observations	10	10			
len					
Mean (SD)	13 (4.5)	8 (2.7)	0.055	0.6	1.4 (0.37; 2.5)
valid (missing)	10 (0)	10 (0)			
1					
Observations	10	10			
len					
Mean (SD)	23 (3.9)	17 (2.5)	0.0033	0.8	1.8 (0.69; 2.9)
valid (missing)	10 (0)	10 (0)			
2					
Observations	10	10			
len					
Mean (SD)	26 (2.7)	26 (4.8)	0.99	0.2	-0.021 (-0.96; 0.92)
valid (missing)	10 (0)	10 (0)			

```
caption.lot = "ToothGrowth analysed by atable",
rowname = NULL)
```

In table 1 the categories of the grouping column `supp` (orange juice (OJ) and vitamin C (VC)) are arranged horizontally; the categories of the splitting column `dose` (0.5, 1, 2) are arranged vertically. The number of observations within each stratum defined by these categories is given. Descriptive statistics of the target column `len` are displayed. Also missing and valid values are counted. p-values and test statistics as well as effect sizes with a 95% confidence interval compare the target column `len` between the categories of the grouping column `dose`. The details about the p-values and confidence intervals can be found in section 3.

The number of observations was 10 in each stratum. There were no missing values. For dose 1 and `supp` OJ the mean (sd) of tooth length was 22.7 (3.91). For dose 1 and `supp` VC the mean (sd) of tooth length was 16.8 (2.52). This difference in tooth length is significant with a p-value of 0.0033. The effects size and its 95% confidence interval is -1.8 (-2.9; -0.69). So in stratum dose 1 the delivery method OJ shows greater tooth length than delivery method VC. For dose 0.5 the p-value is 0.055, just barely missed significance. For dose 2 there is no difference in length for the two delivery methods.

2.2 Apply atable to datasets::mtcars

`datasets::mtcars` comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles. We use `atable` to compare Miles per gallon (`mpg`), horse power (`hp`), number of forward gears (`gear`) and $\frac{1}{4}$ mile time (`qsec`) between number of cylinders (`cyl`) separately for V-shaped engines (`vs`) and non-V-shaped engines. See table 2 for the results. We can also add labels and units via R's attributes and also via `Hmisc`'s label.

```

# all columns of mtcars are numeric, although some are
# better represented as factors
mtcars <- within(datasets::mtcars, {gear <- factor(gear)})

# Add labels and units.
attr(mtcars$mpg, "alias") = "Consumption [Miles (US)/ gallon]"
Hmisc::label(mtcars$qsec) = "Quarter Mile Time"
units(mtcars$qsec) = "s"

# apply atable
the_table <- atable::atable(mpg + hp + gear + qsec ~ cyl | vs,
                           mtcars,
                           format_to = "Latex")

# atable also has a formula method.
# The left side contains the target columns, the right side contains grouping
# and splitting columns separated by the pipe |

# send to LaTeX
Hmisc::latex(the_table,
             file = "",
             title = "",
             label = "tab:mtcarsatable",
             caption = "mtcars analysed by atable.",
             caption.lot = "mtcars analysed by atable",
             rowname = NULL)

```

In table 2 the target columns mpg, hp, gear and qsec are arranged vertically. Statistics and tests are applied to all of them. The grouping columns gear is arranged horizontally. Number of observations was low; some groups only have 4 or less observations, there were empty groups.

For V-shaped engines (vs=0) cars with 8 cylinders have lower miles per gallon, more horse power and more gears than those cars with 4 or 6 cylinders. qsec ($\frac{1}{4}$ mile time) does not depend on number of cylinders. The same conclusion hold for straight engines (vs=0).

Notes:

- atable chooses the descriptive statistics and statistical tests depending on the class of the target column. See section 3 for details.
- atable can handle empty groups and gives appropriate results, see cyl=8 and vs=1.
- atable casts grouping and splitting columns to factors. Target columns are not casted.
- Effect size is not calculated as the grouping column gear has more than two categories.

2.3 Extract specific values from the table

Sometimes addressing a specific value of the table is necessary for reporting, but the values are all squeezed in a data.frame, rounded and formatted as characters. atable can also return all results unformatted.

Table 2: mtcars analysed by atable.

Group	4	6	8	p
0				
Observations	1	3	14	
Consumption [Miles (US)/ gallon]				
Mean (SD)	26 (NA)	21 (0.75)	15 (2.6)	0.011
valid (missing)	1 (0)	3 (0)	14 (0)	
hp				
Mean (SD)	91 (NA)	132 (38)	209 (51)	0.025
valid (missing)	1 (0)	3 (0)	14 (0)	
gear				
3	0% (0)	0% (0)	86% (12)	0.0022
4	0% (0)	67% (2)	0% (0)	
5	100% (1)	33% (1)	14% (2)	
missing	0% (0)	0% (0)	0% (0)	
Quarter Mile Time [s]				
Mean (SD)	17 (NA)	16 (0.77)	17 (1.2)	0.47
valid (missing)	1 (0)	3 (0)	14 (0)	
1				
Observations	10	4	0	
Consumption [Miles (US)/ gallon]				
Mean (SD)	27 (4.7)	19 (1.6)	NaN (NA)	0.0057
valid (missing)	10 (0)	4 (0)	0 (0)	
hp				
Mean (SD)	82 (22)	115 (9.2)	NaN (NA)	0.016
valid (missing)	10 (0)	4 (0)	0 (0)	
gear				
3	10% (1)	50% (2)	NaN% (0)	0.24
4	80% (8)	50% (2)	NaN% (0)	
5	10% (1)	0% (0)	NaN% (0)	
missing	0% (0)	0% (0)	NaN% (0)	
Quarter Mile Time [s]				
Mean (SD)	19 (1.6)	19 (0.82)	NaN (NA)	0.94
valid (missing)	10 (0)	4 (0)	0 (0)	

```

unformatted <- atable::atable(mpg + hp + gear + qsec ~ cyl | vs,
                             mtcars,
                             format_to = "Raw")
# format_to = "Raw" tells atable to skip formatting.

# Extract specific values
unformatted$statistics_result$mpg[[2]]$mean

## [1] 20.56667

unformatted$statistics_result$mpg[[2]]$sd

## [1] 0.7505553

```

Now single values are accessible `unformatted` and can be printed by `\Sexpr{}`. For example: The mean miles per gallon for V-shaped engines with 6 cylinder is 20.5666667 with a standard deviation of 0.7505553.

2.4 Language localisation

Sometimes reports in languages other than English are needed. `atable`'s output can be localised. We will set the language to German via the settings package [van der Loo \(2015\)](#). Note that we use the same dataset `mtcars` as above, which already has labels in english.

```

# Set german words for the table:
atable::atable_options(labels_TRUE_FALSE = c("Ja", "Nein"),
                       labels_Mean_SD = "Mittelwert (SD)",
                       labels_valid_missing = "Ok (fehlend)",
                       colname_for_observations = "N",
                       colname_for_value = "Wert",
                       colname_for_group = "",
                       replace_NA_by = "fehlend")

attr(mtcars$mpg, "alias German") = "Verbrauch [Miles (US)/ gallon]"
attr(mtcars$hp, "alias German") = "PS"

# Tell atable to look for attribute "alias German"
atable_options('get_alias.default' = function(x, ...)
  {attr(x, "alias German", exact = TRUE)})

# apply atable
the_table <- atable::atable(mtcars,
                           target_cols = c("mpg", "hp"))

# reset all options to default
atable_options_reset()

# send to LaTeX
Hmisc::latex(the_table,

```

Table 3: Localised atable. All identifiers produced by atable are now translated to german; also the user can add aliases to all variables for localisation.

	Wert
N	32
Verbrauch [Miles (US)/ gallon]	
Mittelwert (SD)	20 (6)
Ok (fehlend)	32 (0)
PS	
Mittelwert (SD)	147 (69)
Ok (fehlend)	32 (0)

```

file = "",
title = "",
label = "tab:Localisation",
caption = "Localised atable. All identifiers produced by atable are
          now translated to german; also the user can add aliases
          to all variables for localisation.",
caption.lot = "Localised atable",
rowname = NULL)

```

Table 3 shows a localised atable applied to test data shipped with the atable package.

2.5 Word format

atable can also produce printable tables for Word. To do this change the argument `format_to` to 'Word'. The actual print can be done by package `flextable` and `officer`:

```

for_Word <- atable::atable(mpg + hp + gear + qsec ~ cyl | vs, mtcars,
                          format_to = "Word")

# print in Word with packages flextable and officer

MyFTable <- flextable::regularatable(data = for_Word)
# left aligned first column:
MyFTable <- flextable::align(MyFTable, align = "left", j = 1)

# save on disc. Not run here:
# doc <- officer::read_docx()
# doc <- flextable::body_add_flextable(doc, value = MyFTable)
# print(doc, target = "atable and Word.docx")

```

2.6 HTML format

atable can also produce printable tables in HTML. To do this change the argument `format_to` to 'HTML', put the code in a Rmd-file in RStudio [RStudio Team \(2015\)](#) and click on knit to start the

magic. Code looks like this:

```
for_HTML <- atable::atable(mpg + hp + gear + qsec ~ cyl | vs,
                          mtcars,
                          format_to = "HTML")

options(knitr.kable.NA = '')
# knitr::kable(for_HTML, caption="HTML table with atable") # not run.
```

2.7 Console

For interactive analysis the results of `atable` can also be printed human readable in the console.

```
atable::atable(mpg + hp + gear + qsec ~ cyl | vs,
               mtcars,
               format_to = "Console")
```

Note that argument `format_to` may also be set globally via

```
atable_options(format_to = "Console")
```

2.8 Mockup tables

Create a table that contains placeholder instead of actual numbers; a mockup table.

When is such a table useful? The sponsor of the study should know how the trial report will look like before the complete data have been collected yet. This situation arises, when the protocol of the study has been written and the endpoints and some other variables of the study are known but no or only a handful of patient have been recruited.

`atable` provides tools to generate such a mockup table, see table 4. And when data collection is complete the code can be re-used to create the actual table.

```
# set the formatting of numbers so that only 'x' is returned instead of digits.

atable_options("format_p_values" = atable::mockup_format_numbers)
atable_options("format_numbers" = atable::mockup_format_numbers)
atable_options("format_percent" = atable::mockup_format_numbers)

# now apply atable to mtcars as above
the_table <- atable::atable(mpg + hp + gear + qsec ~ cyl | vs,
                           mtcars,
                           format_to = "Latex")

# send to LaTeX
Hmisc::latex(the_table,
             file = "",
             title = "",
             label = "tab:mtcarsatablemockup",
             caption = "mockup table of the mtcars analysis, filled with xxx instead of num
```


Table 4: mockup table of the mtcars analysis, filled with xxx instead of numbers. Compare with table 2.

Group	4	6	8	p
0				
Observations	x	x	xx	
Consumption [Miles (US)/ gallon]				
Mean (SD)	xx (x)	xx.x (x.xx)	xx.x (x.xx)	x.xx
valid (missing)	x (x)	x (x)	xx (x)	
hp				
Mean (SD)	xx (x)	xxx. (xx.x)	xxx. (xx.x)	x.xx
valid (missing)	x (x)	x (x)	xx (x)	
gear				
3	x% (x)	x% (x)	xx.x% (xx)	x.xx
4	x% (x)	xx.x% (x)	x% (x)	
5	xxx% (x)	xx.x% (x)	xx.x% (x)	
missing	x% (x)	x% (x)	x% (x)	
Quarter Mile Time [s]				
Mean (SD)	xx.x (x)	xx.x (x.xx)	xx.x (x.xx)	x.xx
valid (missing)	x (x)	x (x)	xx (x)	
1				
Observations	xx	x	x	
Consumption [Miles (US)/ gallon]				
Mean (SD)	xx.x (x.xx)	xx.x (x.xx)	x (x)	x.xx
valid (missing)	xx (x)	x (x)	x (x)	
hp				
Mean (SD)	xx.x (xx.x)	xxx. (x.xx)	x (x)	x.xx
valid (missing)	xx (x)	x (x)	x (x)	
gear				
3	xx% (x)	xx% (x)	x% (x)	x.xx
4	xx% (x)	xx% (x)	x% (x)	
5	xx% (x)	x% (x)	x% (x)	
missing	x% (x)	x% (x)	x% (x)	
Quarter Mile Time [s]				
Mean (SD)	xx.x (x.xx)	xx.x (x.xx)	x (x)	x.xx
valid (missing)	xx (x)	x (x)	x (x)	

```
Compare with table \\ref{tab:mtcarsatable}.",
caption.lot = "mockup table of the mtcars analysis",
rowname = NULL)
```

```
# back to normal:
atable_options_reset()
```

2.9 Blocks

In `datasets::mtcars` the variables `cyl`, `disp` and `mpg` are related to the engine and `am` and `gear` are related to the gearbox, so grouping them together is desirable. Table 5 is an example of blocking with `datasets::mtcars`.

```
the_table <- atable::atable(datasets::mtcars,
                             target_cols = c("cyl", "disp", "hp", "am", "gear", "qsec") ,
                             blocks = list("Engine" = c("cyl", "disp", "hp"),
                                           "Gearbox" = c("am", "gear")),
                             format_to = "Latex")

# send to LaTeX
Hmisc::latex(
  the_table,
  file = "",
  title = "",
  label = "tab:mtcarsblocking",
  caption = "Blocking shown with datasets::mtcars: Variables cyl, disp and mpg are
in block Engine and variables am and gear in block gearbox. Variable qsec is not
blocked and thus not indented.",
  caption.lot = "Blocking of the mtcars analysis",
  rowname = NULL)
```

3 Scale of measurement, classes and atable

Scale of measurement [Stevens \(1946\)](#) is a well known concept in statistics. The scales are: nominal, ordinal and interval. The scale of measurement narrows operations, statistics and tests that are applicable and meaningful for a variable. Some classes in R have the same properties as these scales of measurement. E.g. class `factor` matches the nominal scale, class `ordered` matches scale ordinal and class `numeric` matches the interval scale. `atable` builds on this matching: Depending on the class of a variable suitable descriptive statistics and hypothesis tests are chosen. See table 6 for details.

The statistical tests in table 6 are meant for two or more independent samples, which arise in parallel group controlled trials. The statistical tests are all non-parametric. Parametric alternatives exist which have greater statistical power if their requirements are met by the data, but non-parametric tests are chosen for their broader field of application. Additionally just because this random package here uses these tests, does not mean that these tests are suitable to analyse a specific study.

4 Modifying atable

The current implementation of tests and statistics (see table 6) is not suitable for all possible data sets. For example the parametric t-test or the robust estimator median may be more adequate for some datasets. Also dates and times are currently not handled by `atable`.

It is intended that some parts of the `atable` package can be altered by the user. This modification is accomplished by replacing the underlying methods or adding new ones, while preserving the structure of arguments and results of the old functions. The workflow of `atable` (and the corresponding function in brackets) is as follows:

Table 5: Blocking shown with datasets::mtcars: Variables cyl, disp and mpg are in block Engine and variables am and gear in block gearbox. Variable qsec is not blocked and thus not indented.

Group	value
Observations	32
Engine	
cyl	
Mean (SD)	6.2 (1.8)
valid (missing)	32 (0)
disp	
Mean (SD)	231 (124)
valid (missing)	32 (0)
hp	
Mean (SD)	147 (69)
valid (missing)	32 (0)
Gearbox	
am	
Mean (SD)	0.41 (0.5)
valid (missing)	32 (0)
gear	
Mean (SD)	3.7 (0.74)
valid (missing)	32 (0)
qsec	
Mean (SD)	18 (1.8)
valid (missing)	32 (0)

Table 6: Classes and atable. Table shows the descriptive statistics and hypothesis tests, that are applied to the three R classes factor, ordered and numeric. Table also shows the appropriate scale of measurement. Class character and logical are treated as nominal scaled variables.

R class	factor	ordered	numeric
scale of measurement	nominal	ordinal	interval
statistic	counts occurrences of every level	as factor	Mean and standard deviation
two sample test	χ^2 test	Wilcoxon test	Rank-Sum Test
effect size	two levels: odds ratio, else Cramér's ϕ	Cliff's Δ	Kolmogorov-Smirnov Test
multi sample test	χ^2 test	Kruskal-Wallis test	Cohen's d
			Kruskal-Wallis test

1. calculate statistics (`statistics`)
2. apply hypothesis tests (`two_sample_hstest` and `multi_sample_hstest`)
3. format statistics results (`format_statistics`)
4. format hypothesis test results (`format_tests`).

These four functions may be altered by the user by replacing existing or adding new methods to already existing S3-generics. Here are two examples:

4.1 Replace existing methods

This example replaces `two_sample_hstest.numeric` with a new function that applies `t.test`, `ks.test` and `cohen.d` simultaneously. See the documentation of `two_sample_hstest`: the function has two arguments called `value` and `group` and returns a named list. First create a new function that does the desired tests:

```
# write a new function:
new_two_sample_hstest <- function(value, group, ...){

  d <- data.frame(value = value, group = group)

  group_levels <- levels(group)
  x <- subset(d, group %in% group_levels[1], select = "value", drop = TRUE)
  y <- subset(d, group %in% group_levels[2], select = "value", drop = TRUE)

  ks_test_out <- stats::ks.test(x, y)
  t_test_out <- stats::t.test(x, y)
  cohen_d_out <- effsize::cohen.d(x, y, na.rm = TRUE)

  # return p-values of both tests
  out <- list(p_ks = ks_test_out$p.value,
             p_t = t_test_out$p.value,
             cohens_d = cohen_d_out$estimate)

  return(out)
}
```

Now create a new version of `statistics.numeric` that calculates the median, MAD, mean and `sd`. See the documentation of `statistics`: the function has one argument called `x` and the ellipsis `...`. The function must return a named list.

```
new_stats <- function(x, ...){

  statistics_out <- list(Median = median(x, na.rm = TRUE),
                       MAD = mad(x, na.rm = TRUE),
                       Mean = mean(x, na.rm = TRUE),
                       SD = sd(x, na.rm = TRUE))

  return(statistics_out)
}
```

```
}
```

These new function currently live in the user's workspace. But they must replace the already existing methods. `atable_options` allows to replace already existing methods globally:

```
atable_options("statistics.numeric" = new_stats)
```

Also `atable` has arguments to allow this replacement:

```
the_table <- atable::atable(atable::test_data,
                           target_cols = "Numeric",
                           group_col = "Group",
                           split_cols = "Split1",
                           format_to = "Latex",
                           two_sample_hptest.numeric = new_two_sample_hptest)
```

Then print the results:

```
Hmisc::latex(the_table,
             file = "",
             title = "",
             label = "tab:modifynumeric",
             caption = "Modified atable also calculates the median, MAD,
             t-test and KS-test.",
             caption.lot = "Modified atable",
             rowname = NULL)
```

See table 7 for the results. `atable` now calculates the median, MAD, mean, sd, cohen's d and performs t- and Kolmogorov-Smirnov tests.

All methods listed above may be altered. see also the documentation of `atable` and `atable_options` for a complete list.

4.2 Add new methods

Currently the generic `statistics` has no method for class `Date` (see `methods(statistics)`). We will define one:

```
statistics.Date <- function(x, ...){

  out <- list(
    Min = min(x, na.rm = TRUE),
    Median = median(x, na.rm = TRUE),
    Max = max(x, na.rm = TRUE)
  )

  class(out) <- c("statistics_Date", class(out))
  # We will need this new class later to specify the format

  return(out)
}
```

Table 7: Modified atable also calculates the median, MAD, t-test and KS-test.

Group	Treatment	Control	p_ks	p_t	cohens_d
C					
Observations	28	21			
Numeric					
Median	0.39	-0.27	0.0021	0.049	0.57
MAD	0.75	0.28			
Mean	0.13	-0.31			
SD	0.92	0.53			
B					
Observations	28	17			
Numeric					
Median	0.039	0.4	0.24	0.78	-0.11
MAD	0.51	1.9			
Mean	-0.031	0.095			
SD	0.93	1.5			
A					
Observations	20	15			
Numeric					
Median	-0.3	0.032	0.38	0.24	-0.45
MAD	0.75	1.3			
Mean	-0.17	0.25			
SD	0.85	1			

It is not necessary to add this method in `atable`'s namespace (as in section 4.1) as R will find the method (only) in the global environment.

We can also alter the formatting of the new method: the minimum and maximum should be next to each other, separated by a semicolon; the median should go below them. See the documentation of `format_statistics`: the function has one argument called `x` and the ellipsis `...`. The function must return a `data.frame` with names `tag` and `value` with class `factor` and `character` respectively.

```
format_statistics.statistics_Date <- function(x, ...){

  min_max <- paste0(x$Min, "; ", x$Max)
  Median <- as.character(x$Median)

  out <- data.frame(
    tag = factor(c("Min Max", "Median"), levels = c("Min Max", "Median")),
    value = c(min_max, Median),
    stringsAsFactors = FALSE)
  # the factor needs levels for the non-alphabetic order
  return(out)
}
```

Note that there is also a default method for `format_statistics`, that just returns the names and values of `x` as a `data.frame`, see table 7 for the result of the default formatting. Now print the table:

```
the_table <- atable::atable(atable::test_data,
                           target_cols = "Date",
                           format_to = "Latex")

Hmisc::latex(the_table,
             file = "",
             title = "",
             label = "tab:addedDate",
             caption = "atable with added methods for class Date. Now calculates
             minimum, maximum and median for this class",
             caption.lot = "atable with added methods for class Date",
             rowname = NULL)
```

Table 8 shows the application the new methods for class `Date`. The statistics and their format are as specified.

Adding new user-defined methods to `atable` (as described above) can introduce errors to the code. To prevent some of these possible errors, functions that check the results of `statistics` and `format_statistics` etc. were implemented in `atable`; these function are called `check_...`. Also the user is advised to read the documentation of the generic that she/he wants to modify.

References

ICH Harmonised Tripartite Guideline. Statistical principles for clinical trials. International Conference on Harmonisation E9 Expert Working Group. *Statistics in medicine*, 18:1905–1942, Aug. 1999. ISSN 0277-6715.

Table 8: atable with added methods for class Date. Now calculates minimum, maximum and median for this class

Group	value
Observations	129
Date	
Min Max	2001-06-21; 2001-07-29
Median	2001-07-07

- D. Gohel. *flextable: Functions for Tabular Reporting*, 2018a. URL <https://CRAN.R-project.org/package=flextable>. R package version 0.4.4.
- D. Gohel. *officer: Manipulation of Microsoft Word and PowerPoint Documents*, 2018b. URL <https://CRAN.R-project.org/package=officer>. R package version 0.3.2.
- F. E. Harrell Jr, with contributions from Charles Dupont, and many others. *Hmisc: Harrell Miscellaneous*, 2018. URL <https://CRAN.R-project.org/package=Hmisc>. R package version 4.1-1.
- F. Mittelbach, M. Goossens, J. Braams, D. Carlisle, and C. Rowley. *The LaTeX Companion (Tools and Techniques for Computer Typesetting)*. Addison-Wesley Professional, 2004. ISBN 0-201-36299-6. URL <https://www.amazon.com/LaTeX-Companion-Techniques-Computer-Typesetting/dp/0201362996?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbiori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0201362996>.
- D. Moher, S. Hopewell, K. F. Schulz, V. Montori, P. C. Gotzsche, P. J. Devereaux, D. Elbourne, M. Egger, and D. G. Altman. CONSORT 2010 explanation and elaboration: updated guidelines for reporting parallel group randomised trials. *BMJ*, 340(mar23 1):c869–c869, mar 2010. doi: 10.1136/bmj.c869.
- RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, Inc., Boston, MA, 2015. URL <http://www.rstudio.com/>.
- S. S. Stevens. On the theory of scales of measurement. *Science*, 103(2684):677–680, jun 1946. doi: 10.1126/science.103.2684.677.
- M. van der Loo. *settings: Software Option Settings Manager for R*, 2015. URL <https://CRAN.R-project.org/package=settings>. R package version 0.2.4.
- Y. Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2018. URL <https://yihui.name/knitr/>. R package version 1.20.