atable package: Usage

Armin Ströbel

November 29, 2019

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

3 Scale of measurement, classes and atable 8

4 Modifying atable 9
  4.1 Replace existing methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
  4.2 Add new methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

List of Tables

  1 ToothGrowth analysed by atable . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
  2 mtcars analysed by atable . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
  3 Localised atable . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
  4 Classes and atable . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
  5 Modified atable . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
  6 atable with added methods for class Date . . . . . . . . . . . . . . . . . . . . . . . 13

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).

```r
# 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.",
             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 ¼ 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.
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 that those cars with 4 or 6 cylinders. qsec (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.

unformatted <- atable::atable(mpg + hp + gear + qsec ~ cyl | vs, 
mtcars, 
format_to = "Raw")
Table 2: mtcars analysed by atable.

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

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.

<table>
<thead>
<tr>
<th></th>
<th>Wert</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>32</td>
</tr>
<tr>
<td>Verbrauch [Miles (US)/ gallon]</td>
<td>Mittelwert (SD)</td>
</tr>
<tr>
<td></td>
<td>Ok (fehlend)</td>
</tr>
<tr>
<td>PS</td>
<td>Mittelwert (SD)</td>
</tr>
<tr>
<td></td>
<td>Ok (fehlend)</td>
</tr>
</tbody>
</table>

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:

```r
for_Word <- atable::atable(mpg + hp + gear + qsec ~ cyl | vs, mtcars,
                          format_to = "Word")
# print in Word with packages flextable and officer
MyFTable <- flextable::regulartable(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:
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")
```

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 maths 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 4 for details.

Table 4: 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.

<table>
<thead>
<tr>
<th>R class</th>
<th>factor</th>
<th>ordered</th>
<th>numeric</th>
</tr>
</thead>
<tbody>
<tr>
<td>scale of measurement statistic</td>
<td>counts occurences of every level</td>
<td>as factor</td>
<td>Mean and standard deviation</td>
</tr>
<tr>
<td>two sample test</td>
<td>(\chi^2) test</td>
<td>Wilcoxon Rank-Sum test</td>
<td>Kolmogorov-Smirnov Test</td>
</tr>
<tr>
<td>effect size</td>
<td>two levels: odds ratio, else Cramér’s (\phi)</td>
<td>Cliff’s (\Delta)</td>
<td>Cohen’s d</td>
</tr>
<tr>
<td>multi sample test</td>
<td>(\chi^2) test</td>
<td>Kruskal-Wallis test</td>
<td>Kruskal-Wallis test</td>
</tr>
</tbody>
</table>

The statistical tests in table 4 are meant for two or more independent samples, which arise in parallel group controlled trials. The statistical tests are all non-parametric. Parametric alternatives
exists 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 4) 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:

1. calculate statistics (statistics)
2. apply hypothesis tests (two_sample_htest and multi_sample_htest)
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_htest.numeric with a new function that applies t.test, ks.test and cohen.d simultaneously. See the documentation of two_sample_htest: the function has two arguments called value and group and returns a named list. First create a new function that does the desired tests:

```r
# write a new function:
new_two_sample_htest <- 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)
}
```
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.

```r
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:

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

Also `atable` has arguments to allow this replacement:

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

Then print the results:

```r
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 5 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 my be altered. see also the documentation of `atable` and `atable_options` for a complete list.
Table 5: Modified atable also calculates the median, MAD, t-test and KS-test.

<table>
<thead>
<tr>
<th>Group</th>
<th>Treatment</th>
<th>Control</th>
<th>p_ks</th>
<th>p_t</th>
<th>cohens_d</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Observations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Numeric</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Median</td>
<td>0.39</td>
<td>-0.27</td>
<td>0.0021</td>
<td>0.049</td>
</tr>
<tr>
<td></td>
<td>MAD</td>
<td>0.75</td>
<td>0.28</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>0.13</td>
<td>-0.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.92</td>
<td>0.53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Observations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Numeric</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Median</td>
<td>0.039</td>
<td>0.4</td>
<td>0.24</td>
<td>0.78</td>
</tr>
<tr>
<td></td>
<td>MAD</td>
<td>0.51</td>
<td>1.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>-0.031</td>
<td>0.095</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.93</td>
<td>1.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>Observations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Numeric</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Median</td>
<td>-0.3</td>
<td>0.032</td>
<td>0.38</td>
<td>0.24</td>
</tr>
<tr>
<td></td>
<td>MAD</td>
<td>0.75</td>
<td>1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>-0.17</td>
<td>0.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.85</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.2 Add new methods

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

\begin{verbatim}
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)
}
\end{verbatim}

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 \texttt{format_statistics}: the function has one argument called \texttt{x} and the ellipsis \texttt{...}. The function must return a data.frame with names \texttt{tag} and \texttt{value} with class factor and character respectively.

\begin{verbatim}
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)
}
\end{verbatim}

Note that there is also a default method for \texttt{format_statistics}, that just returns the names and values of \texttt{x} as a data.frame, see table 5 for the result of the default formatting. Now print the table:

\begin{verbatim}
the_table <- atable::atable(atable::test_data,
    target_cols = "Date",
    format_to = "Latex")
Hmisc::latex(the_table,
\end{verbatim}
Table 6: A table with added methods for class Date. Now calculates minimum, maximum and median for this class.

<table>
<thead>
<tr>
<th>Group</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observations</td>
<td>129</td>
</tr>
<tr>
<td>Date</td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>Median</td>
<td>2001-07-07</td>
</tr>
</tbody>
</table>

Table 6 shows the application of the new methods for class Date. The statistics and their format are as specified.

Adding new user-defined methods to a table (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 a table; these functions are called `check_...`. Also, the user is advised to read the documentation of the generic that she/he wants to modify.

References


