The **dataref** package

Christian Dietrich

stettberger@dokucode.de

v0.1 from 2013/12/06

1   Introduction

Writing scientific texts is a craft. It is the craft of communicating your results to your colleagues and to the curious world public. Often your conclusions are based upon facts and numbers that you gathered during your research for the specific topic. You might have done many experiments and produced lot of data. The craft of writing is to guide your reader through a narrative that is based upon that data. But there may be many versions of that data. Perhaps you found a problem in your experiment, while already writing, that forces you back into the laboratory. After a while, the moon has done its circle many times, you return from that dark place and your methodology has improved as significantly as your data has. But now you have to rewrite that parts of the data, that reference the old data points.

The **dataref** is here to help you with managing your data points. It provides you with macro style keys, that represent symbolic names for your datapints. You can reference those symbolic names with `\dref`, use them in calculations to have always up-to-date percentage values, define projections between sets of data points and document them. **dataref** also introduces the notion of assertions (`\drefassert`) for your results to ensure that your prosa text references fit the underlying data.

2   Usage

The **dataref** package heavily uses `pgfkeys` and `pgfmath` to perform storage and operations upon data points. See `texdoc pgfmanual` for further informations about those topics.

`\drefset{⟨name⟩}{⟨value⟩}`

The `\drefset` command is used to define the symbolic data points. The first argument is the symbolic name, the second argument is the value. The value can be a number, but it can also be arbitrary text. The key may contain virtually all characters, including spaces and slashes. It is good practice use a hierarchy to structure you data point names.

```
\drefset{/control group/mice race}{Black Six}
\drefset{/control group/mice count}{32}
\drefset{/control group/dead after 24h}{3}
\drefset{/control group/dead after 48h}{7}
\drefset{/control group/recovered}{6}

\drefset{/med A/mice race}{Black Six}
\drefset{/med A/mice count}{32}
\drefset{/med A/dead after 24h}{6}
\drefset{/med A/dead after 48h}{1}
\drefset{/med A/recovered}{9}
```

The code snippet, that is best stored in an external file, and which might be auto-generated, is best read with \input. It defines 10 symbolic names, that are partitioned into two "directories" (control group and medicament A).

\dref{⟨*name*⟩}

This macro is used to reference a single symbolic data point. The value stored in that datapoint is inserted into the text. \dref does additonally mark the data point as used. It will then appear in the dref usage report. For undefined keys the default behaviour is to abort the compilation. But the package option ignoremissing just outputs a warning. All referenced/missing/found datapoints are noted in the aux file.

Macro	Expansion
\dref{/control group/mice race}	Black Six
\dref{/control group/mice count}	32
\dref[sci,precision=2,zerofill=true]{/med A/recovered}	$9.00 \cdot 10^0$

\dref additionally takes an optional argument. This argument is interpreted as /pgf/number format/ argument. See the pgf/tikz manual for more information. Only if the optional argument is present the value is printed as a number, so [] is a useful option, since it enforces printing it as a number with the proper number format applied. Be aware that \dref is **not** expandable.

\drefvalueof{⟨*name*⟩}

Since \dref is not expandable, this macro can be used to get the bare value of a symbolic data point. But use it with caution, since it bypasses all internal book keeping.

\drefvalueof{/med A/mice race}

\drefref{⟨*name*⟩}

This is complement of \drefvalueof, it does *only* the book keeping for a key (marking it as referenced etc.) So it might be used to compensate the usage of its bad sibling.

\drefref{/med A/mice race}

[ignoremissing]
[defaultvalue=1.0]

This two package options influence the behaviour regarding to unknown keys.

With `ignoremissing` each missing symbolic datapoint is replaced by the default-value. This behaviour might be useful, when you use the .aux file, where the unknown keys are noted to extract data points from a third source (e.g. database, wikidata, etc). In the future a secondary tool will be provided to resolve those references.

`\drefsethelp{⟨pattern⟩}{⟨text⟩}`
`\drefhelp{⟨name⟩}`   dataref comes with a simple method for defining documentation for data points. This help for example be used to communicate what is the concrete semantiv of the data point. This is of special interest when writter and data gatherer are not the same person. `\drefsethelp` takes two arguments: first a regular expression that matches the symbolic data point, second the help text.

```
\drefsethelp{.*/mice race}{The mice race used for experiments heavily
    influences the outcome of the results}
```

The helptext for a key is obtained by using `\drefhelp` macro. It checks all defined helps (in linear order, first defined, first matched), and prints the first matching help text.

> `\drefhelp{/med A/mice race}`

`\drefcalc[⟨lets⟩]{⟨expr⟩}[⟨format⟩]`
    The `\drefcalc` command is the core function of calculating with data points. It is based on the pgfmath engine. It uses the required argument as an mathematical expression, but has additional features, that can be used.

> `\drefcalc{(4+7)/12 * 100}` $\Rightarrow 91.67$

It adds support for the `data` function within pgfmath, that references symbolic data points. The keyname has to be in double quotes to indicate a string, but you can easily define an appropriate macro, that abstracts from `data("")`.

> `\drefcalc{data("/control group/recovered") /`
>   `data("/control group/mice count")}`
> $\Rightarrow 0.19$
> `\newcommand{\cg}[1]{data("/control group/#1")}`
> `\drefcalc{\cg{recovered}/\cg{mice count}}` $\Rightarrow 0.19$

The first optional argument lets you define constants within pgfmath (zero arity functions, that can be called without parenthesis). Those bindings are only valid for the current `\datarefcalc` call.

> `\drefcalc[A=\cg{recovered},B=12*20]{B/A}` $\Rightarrow 40$
> `\drefcalc[X=100]{30/X}` $\Rightarrow 0.3$

The second optional argument, that appears after the required argument does define the pgfmath's `number format`.

> `\drefcalc{1/3}[precision=5,fixed]` $\Rightarrow 0.33333$

`\drefcalc*`
`\drefresult`
`\drefformat{`⟨*number*⟩`}`
When you get confused of the which optional argument does what, just think of a pipe. First you define bindings, then you calculate, then you emit stuff. When drefcalc is called with an star argument, it does not print the result, but does only set `\drefresult`. `\drefformat` is used to format a number.

> `\drefcalc*{1/3} ABC: \drefresult` ⇒ ABC: 0.33333
> `\drefformat[fixed,precision=1]{\drefresult}`⇒ 0.3
> `\drefformat[sci]{100000}` ⇒ $1 \cdot 10^5$

`\dreflet{`⟨*lets*⟩`}`

The bindings for `\drefcalc` are only local to that macro call. Defining a binding for the current group can be done with `\dreflet`.

> `\newcommand{\cg}[1]{data("/control group/#1")}`
> `\dreflet{percent=data("/med A/mice count")/100}`

> The result cleary shows that a lorem ipsum kills
> `\drefcalc{\cg{dead after 24h}/percent}` percent within 24 and
> `\drefcalc{\cg{dead after 28h}/percent}` percent within 48 hours.

> The result cleary shows that a lorem ipsum kills 9.38 percent within 24 and 21.88 percent within 48 hours.

`\drefprojection{`⟨*from*⟩`}{`⟨*to*⟩`}{`⟨*projection*⟩`}`
Sometimes one or multiple sets of data have to projected/mixed into a new set of data, that is fully depended on those values. This can be done with t projects one data set (subdirectoy) into another one. within the projection three different operations are possible: `\id`, `\rename` and `\alc`.
identity function renaming of points

> `\drefprojection{/control group}{/projection}{`
> `    \id{mice race} % identity function`
> `    \rename{mice count}{count} % renaming of points`
> `    \calc{data("/dead after 24h")+data("/dead after 48h")}{died}`
> `  }`

> `\dref{/projection/died}` ⇒ 10.0
> `\dref{/projection/mice race}` ⇒ Black Six
> `\dref{/projection/count}` ⇒ 32

`\drefrow{`⟨*list*⟩`}{`⟨*macro*⟩`}`
`\drefrow*`
Often different columns in a table have to be obtained from your data points. Often those rows and columns are similar. Generating parts of tables within LATEXis very tricky, so dataref provides you with `\drefrow`. This macro iterates over a comma-separated list of values and fills out a macro which is interpreted as an symbolic data point. The entries are seperated with & and printed. In the starred variant the resulting text is not interpreted as symbolic name, but as a macro.

The second argument is the macro, and can have two macro replacements. The first replacement `#1` is the value of the list item, the second `#2` is the index in the list.

```
\begin{tabular}{lccc}
   Group & $<$ 24h & $<$48h & recovered\\ \hline
   Control Group & \drefrow{dead after 24h,dead after 48h,recovered}%
                       {/control group/#1}\\
   Medicament A & \drefrow{dead after 24h,dead after 48h,recovered}%
                       {/med A/#1}\\
   Starred Variant & \drefrow*{B,C,D}{\#1=#1,\#2=#2}\\
 \end{tabular}
```

Group	< 24h	<48h	recovered
Control Group	3	7	6
Medicament A	6	1	9
Starred Variant	#1=B,#2=1	#1=C,#2=2	#1=D,#2=3

`\drefassert{⟨expr⟩}`
`[noassert]`

Sometimes the underlying data changes, while you are writing. But what when your prose text relies on certain characteristics of the data (At least this should be goal!). `\drefassert` uses a pgfmath expression that evaluates to `true` or `false`. When the assertion holds (`true`) nothing happens, only a terminal message is printed. When it does not hold (`false`) the compilation is aborted.

```
\drefassert{data("/control group/mice count") > 30}
 Of the more than thirty infected mice...
```

The **noassert** package options disables the latex abortion. In that case only a terminal message is printed.

`[annotate=none]`
`[annotate=footnote]`
`[annotate=pdfcomment]`
`\drefannotate{⟨style⟩}`

While writing a document it is desirable to know, what key is used, while writing the text and generating the document. Therefore **dataref** provides the possibility to annotate values. The default package option **none** disables this kind of annotation. For the **pdfcomment** option uses pdf annotations. Be aware that those annotations work only on a few selected PDF readers properly. `\drefannotate` sets the annoation style for the current group.

```
\drefannotate{none}
```
Black Six, 32, 33.33

```
\drefannotate{footnote}
```
Black Six[1], 32[2], 33.33[3]

---

[1] /control group/mice race
[2] /control group/mice count
[3] 100/3

`\drefusagereport`

`[usagereport]`

With the **usagereport** package option enabled, `\drefusagereport` generates a usagereport of all referenced keys. The usagereport groups the keys by the help texts.

Datagraphy

The mice race used for experiments heavily influences the outcome of the results

	Page(s)	Value
/control group/mice race	2, 4, 5	Black Six
/projection/mice race	4	Black Six

Of all infected mice, a certain number died within a specified period of time. A certain recovered from the infection. Each mouse falls into one this category.

	Page(s)	Value
/med A/recovered	2, 5	9
/control group/recovered	3, 5	6
/control group/dead after 24h	4, 5	3
/control group/dead after 48h	4, 5	7
/med A/dead after 24h	5	6
/med A/dead after 48h	5	1

Keys without Help	Page(s)	Value
/control group/mice count	2, 3, 4, 5	32
/med A/mice count	4	32
/projection/died	4	10.0
/projection/count	4	32

3 Implementation

Guard against reading twice

```
1 \ifx\drefloaded\undefined
2   \let\drefloaded=\relax
3 \else
4   \expandafter\endinput
5 \fi
6 \ifx\PackageError\undefined
7   \def\dref@error#1{\immediate\write-1{Package dref: Error! #1.}}%
8 \else
9   \def\dref@error#1{\PackageError{dref}{#1}{}}%
10 \fi
11 % \end{macrocode}
12 %
13 % \begin{macrocode}
14 \RequirePackage{pgf}
15 \RequirePackage{kvoptions}
16 \RequirePackage{xparse}
17 \RequirePackage{etoolbox}
18 \RequirePackage{etextools}

19 \SetupKeyvalOptions{
20   family=dref,
21   prefix=dref@
22 }
23 \DeclareStringOption[/data]{datapath}
24 \DeclareStringOption[1]{defaultvalue}
25 \DeclareStringOption[none]{annotate}
26 \DeclareBoolOption{usagereport}
27 \DeclareBoolOption{ignoremissing}
28 \DeclareBoolOption{noassert}
29 \ProcessKeyvalOptions*
```

\dref@set

```
30 \newcommand{\dref@set}[2]{%
31     \pgfkeys@temptoks{#2}%
32     \expandafter\xdef\csname pgfk@\dref@datapath#1\endcsname{\the\pgfkeys@temptoks}%
33 }
```

\drefset

```
34 \def\drefset#1#2{\dref@set{#1}{#2}}
```

\dref@expandable

```
35 \long\def\dref@expandable#1{%
36   \pgfkeysifdefined{\dref@datapath#1}{%
37     \pgfkeysvalueof{\dref@datapath#1}%
38   }{%
39     \dref@defaultvalue%
40   }%
```

```
41 }
```

\dref@unexpandable

```
42 \long\def\dref@unexpandable#1{%
43   \def\drefcurrentkey{\dref@datapath#1}%
44   \pgfkeysifdefined{\drefcurrentkey}{%
45     \immediate\write\@auxout{\noexpand\dref@found{\drefcurrentkey}{\thepage}}%
46   }{%
47     \immediate\write\@auxout{\noexpand\dref@notfound{\drefcurrentkey}{\thepage}}%
48     \ifdref@ignoremissing%
49       \typeout{Dref warning: undefined key '\drefcurrentkey'}%
50       \dref@mkannotate{UNDEFINED: \drefcurrentkey}%
51     \else%
52       \dref@error{Dref error: undefined key '\drefcurrentkey'}%
53     \fi%
54   }%
55   \immediate\write\@auxout{\noexpand\dref@referenced{\drefcurrentkey}{\thepage}}%
56 }
```

\dref

```
57 \DeclareDocumentCommand{\dref}{o m}{%
58   \dref@unexpandable{#2}%
59   \IfNoValueTF {#1}{%
60     \gdef\dref@dref@output{\dref@expandable{#2}}%
61   }{%
62     \gdef\dref@dref@output{%
63       \pgfmathparse{\dref@expandable{#2}}%
64       \dref@format[#1]{\pgfmathresult}%
65     }%
66   }%
67   \dref@dref@output%
68   \dref@mkannotate{#2}%
69 }
```

\drefvalueof

```
70 \def\drefvalueof#1{%
71   \dref@expandable{#1}%
72 }
```

\drefref

```
73 \def\drefref#1{%
74   \dref@unexpandable{#1}%
75 }
```

\dref@help@match

```
76 \newcommand{\dref@help@match}[2]{%
77   \ifstrmatch{#1}{#2}%
78 }
```

```
79 \newcommand{\dref@help}[2][]{%
80   \pgfkeysifdefined{#2/help}{%
81     \pgfkeysvalueof{#2/help}%
82   }{#1}%
83 }
```

```
84 \csdef{dref@helps}{}
85 \newcommand{\drefsethelp}[2]{
86   \csdef{dref@help@#1}{#2}%
87   \listcsadd{dref@helps}{#1}%
88 }
```

```
89 \newcommand{\drefhelp}[1]{
90   \renewcommand{\do}[1]{%
91     \dref@help@match{##1}{#1}{%
92       \csuse{dref@help@##1}%
93     \listbreak}{}%
94   }%
95   \ifcsvoid{dref@helps}{}{%
96     \dolistcsloop{dref@helps}%
97   }%
98 }
```

```
99 \long\def\dref@notfound#1#2{
100   \ifdref@usagereport%
101     \dref@usagereport@notfound{#1}{#2}%
102   \else\relax\fi%
103 }
104 \long\def\dref@found#1#2{
105   \ifdref@usagereport%
106     \dref@usagereport@found{#1}{#2}%
107   \else\relax\fi%
108 }
109 \long\def\dref@referenced#1#2{
110   \ifdref@usagereport%
111     \dref@usagereport@referenced{#1}{#2}%
112   \else\relax\fi%
113 }
```

```
114 \def\dref@let#1{%
115   \def\@tmp##1=##2;{\pgfmathdeclarefunction*{##1}{0}{\pgfmathparse{##2}}}%
116   \renewcommand*{\do}[1]{\@tmp##1;}%
117   \docsvlist{#1}%
118 }
```

```
119 \def\dreflet#1{%
120   \dref@let{#1}%
121 }
```

```
122 \DeclareDocumentCommand{\dref@calc}{o m}{%
123   \IfNoValueTF {#1}{}{%
124     \dref@let{#1}%
125   }%
126   \pgfmathparse{#2}%
127 }
128 \def\drefresult{0.0}
129 \DeclareDocumentCommand{\drefcalc}{s O{} m O{}}{%
130   \begingroup%
131   \dref@calc[#2]{#3}%
132   \xdef\drefresult{\pgfmathresult}%
133   \IfBooleanTF {#1} {}% Wit star do not print anything
134               {%
135                 \dref@format[#4]{\pgfmathresult}%
136                 \dref@mkannotate{#3}%
137               }%
138   \endgroup%
139 }
```

```
140 \newcommand{\dref@format}[2][]{%
141   \pgfmathprintnumber[#1]{#2}%
142 }
143 \DeclareDocumentCommand{\drefformat}{O{} m}{%
144   \dref@format[#1]{#2}%
145 }
```

```
146 \gdef\dref@data@math@prefix{}
147 \pgfmathdeclarefunction{data}{1}{%
148       \begingroup%
149             \dref@unexpandable{\dref@data@math@prefix#1}%
150             \pgfmathparse{\dref@expandable{\dref@data@math@prefix#1}}%
151             \pgfmath@smuggleone\pgfmathresult%
152       \endgroup%
153 }
154 \DeclareDocumentCommand{\drefprojection}{m m m}{%
155   \begingroup%
156     \def\dref@data@math@prefix{#1}%
157     \def\rename##1##2{\dref@unexpandable{#1/##1}\drefset{#2/##2}{\dref@expandable{#1/##1}}}%
158     \def\id##1{\rename{##1}{##1}}%
159     \def\calc##1##2{%
160       \begingroup%
```

10

```
161            \dref@calc{##1}%
162            \xdef\dref@project@result{\pgfmathresult}
163          \endgroup%
164          \drefset{#2/##2}{\dref@project@result}%
165        }%
166      #3%
167      \endgroup%
168 }
```

\dref@makerow

```
169
170 \newtoks\dref@toks
171
172 \newcommand{\dref@makerow}[2]{%
173   {\global\dref@toks={}%
174     \@tempcnta=\z@%
175     \def\inner##1##2{#2}%
176     \renewcommand*{\do}[1]{%
177       \advance\@tempcnta\@ne%
178       \csdef{@cell\number\@tempcnta}{\inner{##1}{\number\@tempcntb}}%
179     }%
180     \expandafter\def\expandafter\arglist\expandafter{#1}%
181     \expandafter\docsvlist\expandafter{\arglist}%
182     \@tempcntb=\z@
183        {\loop\ifnum\@tempcntb<\@tempcnta
184           \advance\@tempcntb\@ne
185           \edef\next{%
186             \ifnum\@tempcntb=\@ne\else&\fi
187             \csuse{@cell\number\@tempcntb}}%
188           \global\dref@toks=\expandafter{\the\expandafter\dref@toks\next}%
189           \repeat}%
190   }%
191   \the\dref@toks}
192 \DeclareDocumentCommand{\drefrow}{s m m}{%
193   \IfBooleanTF {#1} {%
194     \dref@makerow{#2}{#3}%
195   }{% Wit star do not print anything
196     \dref@makerow{#2}{\dref[]{#3}}%
197   }%
198 }
```

\dref@mkannotate

```
199
200 \expandafter\ifstrequal\expandafter{\dref@annotate}{pdfcomment}{
201   \RequirePackage{pdfcomment}
202 }
203
204 \newcommand{\dref@mkannotate}[1]{%
205   \expandafter\ifstrequal\expandafter{\dref@annotate}{none}%
206     {\relax}%
```

```
207     {\expandafter\ifstrequal\expandafter{\dref@annotate}{footnote}%
208       {\footnote{#1}}%
209       {\expandafter\ifstrequal\expandafter{\dref@annotate}{pdfcomment}%
210         {\pdfcomment[opacity=0.4,voffset=2ex]{#1}}%
211         {\dref@error{Value for annotate not supported: '\dref@annotate'}%
212          }}}}%
213
214 \newcommand{\drefannotate}[1]{%
215   \renewcommand{\dref@annotate}{#1}%
216 }
```

Usagereport

```
217 \ifdref@usagereport
218   \RequirePackage{longtable}
219   \RequirePackage{booktabs}
220 \fi
```

\dref@usagereport@referenced

```
221 \newcommand{\dref@usagereport@notfound}[2]{}
222 \newcommand{\dref@usagereport@found}[2]{}
223
224 \csdef{pgfdat@usagereport@keys}{}
225 \csdef{pgfdat@usagereport@matchedkeys}{}
226
227 \newcommand{\dref@usagereport@referenced}[2]{
228   \ifinlistcs{#2}{dref@usagereport@referenced@#1}{}{
229     \listcsgadd{dref@usagereport@referenced@#1}{#2}
230   }
231   \ifinlistcs{#1}{dref@usagereport@keys}{}{
232     \listcsgadd{dref@usagereport@keys}{#1}
233   }
234 }
```

\dref@usagereport@strippath

```
235 \expandafter\def\expandafter\dref@usagereport@strippath@\dref@datapath#1\blanktest{#1}
236
237 \newcommand{\dref@usagereport@strippath}[1]{%
238   \expandafter\ifstrmatch\expandafter{\expandafter^\dref@datapath.*$}{#1}%
239     {\dref@usagereport@strippath@#1\blanktest}%
240     {#1}%
241 }
```

usagereport@formatreferencelist

```
242 \newcommand{\dref@usagereport@formatreferencelist}[1]{%
243   \begingroup%
244   \def\sep{}%
245   \renewcommand{\do}[1]{\sep\ifdef{\hyperlink}{\hyperlink{page.##1}{##1}}{##1}\def\sep{, }}%
246   \dolistcsloop{dref@usagereport@referenced@#1}%
247   \endgroup%
248 }
```

```
249 \newcommand{\dref@usagereport@keyheader}[1]{%
250   \textbf{\ifdef{\hypertarget}%
251     {\hypertarget{#1}{\dref@usagereport@strippath{#1}}}%
252     {\dref@usagereport@strippath{#1}}}%
253   & \dref@usagereport@formatreferencelist{#1}%
254   & \pgfkeysifdefined{#1}{\pgfkeysvalueof{#1}}{\textbf{\red{undefined}}} \\%
255 }
```

```
256 \newcommand{\dref@usagereport@forhelp}[1]{%
257   \begingroup%
258   \noindent\csuse{dref@help@#1}
259   \renewcommand{\do}[1]{%
260     \dref@help@match{#1}{##1}{%
261       \dref@usagereport@keyheader{##1}%
262       \ifinlistcs{##1}{dref@usagereport@matchedkeys}{}{%
263         \listcsgadd{dref@usagereport@matchedkeys}{##1}%
264       }%
265     }{}%
266   }%
267   \begin{longtable}{lll}\toprule%
268     & Page(s) & Value \\ \midrule
269   \dolistcsloop{dref@usagereport@keys}%
270   \end{longtable}%
271   \endgroup%
272 }
```

```
273 \newcommand{\dref@usagereport@withouthelp}{%
274   \renewcommand{\do}[1]{%
275     \ifinlistcs{##1}{dref@usagereport@matchedkeys}{}{%
276       \dref@usagereport@keyheader{##1}%
277     }%
278   }%
279   \begin{longtable}{lll}\toprule%
280     Keys without Help  & Page(s) & Value \\\midrule
281     \endhead
282     \dolistcsloop{dref@usagereport@keys}%
283   \end{longtable}%
284 }
```

```
285 \newcommand{\drefusagereport}{%
286   \ifdref@usagereport%
287   \ifcsvoid{dref@usagereport@keys}{\typeout{EMPTY}}{%
288   \begingroup%
289   \setlength{\LTleft}{2em}%
290   \setlength{\LTright}{0pt}%
```

13

```
291    \renewcommand{\do}[1]{%
292      \ifinlistcs{##1}{dref@usagereport@matchedkeys}{}{%
293        \dref@usagereport@forhelp{##1}%
294      }%
295    }%
296    \dolistcsloop{dref@helps} % For all help text
297    \setlength{\LTleft}{0em}%
298    \dref@usagereport@withouthelp\relax
299    \endgroup%
300    }% csempty @keys
301    \fi%
302 }
```

\drefassert

```
303 \newcommand{\drefassert}[1]{%
304    \begingroup%
305      \pgfmathsetmacro{\result}{(#1) ? 1 : 0}
306      \expandafter\ifstrequal\expandafter{\result}{1.0}{%
307        \typeout{Assertion holds: #1}%
308      }{%
309        \ifdref@noassert%
310          \typeout{Assertion failed: #1}%
311        \else%
312          \dref@error{Assertion failed: #1}%
313        \fi%
314    }%
315    \endgroup%
316 }
```