The dataref package

Christian Dietrich 2013 stettberger@dokucode.de
https://github.com/stettberger/dataref

2014/08/20 v0.2

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 data points. 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 information 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 to 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, which 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⟩}`
`\dref[⟨format⟩]{⟨name⟩}`

This macro is used to reference a single symbolic data point. The value stored in that datapoint is inserted into the text. `\dref` additionally marks 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 in the unstarred version this macro parses the value as a number. 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]`

These two package options influence the behaviour regarding unknown keys. With `ignoremissing` each missing symbolic datapoint is replaced by the defaultvalue. 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 can for example be used to communicate what is the concrete semantics 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 the `\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[⟨format⟩]{⟨expr⟩}`
`data("⟨key⟩")`
`d(⟨key⟩)`

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 a mathematical expression, but has additional features, that can be used.

`\drefcalc{(4+7)/12 * 100}` ⇒ 91.67

It adds support for the `data` function within pgfmath, which 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("")`. As a quote-free alternative to the data command, `\drefcalc` provides also `d(<key>)`.

`\drefcalc{data("/med A/mice count") * 100}` ⇒ 3,200 `\drefcalc{d(/med A/mice count) *`
⇒ 3,200

The optional argument lets you give a number format, which is used for printing the result number (`/pgf/number format`).

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

`\drefcalc` works as well in a **/pgf/fpu** environment or a normal one. The FPU feature of pgfmath is used to handle large numbers, which may occur often when handling experiment data points.

`\dreflet{A=123456789, B=987654321, a=12, b=98}`

Macro	Inserted Text	`\drefresult`
`\drefcalc[/pgf/fpu]{A/B}`	0.12	0.1241
`\drefcalc{a/b}`	0.12	0.12244
`\drefcalc*[/pgf/fpu]{A/B}`		0.1241
`\drefcalc*{a/b}`		0.12244

`\drefcalc*`
`\drefresult`
`\drefformat{⟨number⟩}`

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

`/dref/let={⟨lets⟩}`
`\dreflet{⟨lets⟩}`

Since symbolic key names can get long, dataref has the possibility to define variables for use within mathematical expression from other expressions. These "let"-bindings can either be defined locally for a `\drefcalc` commando with a pgf key or globally with `\dreflet`.

`\drefcalc[/dref/let{A=12*20,B=\cg{recovered}}]{A/B}` ⇒ 40
`\drefcalc[/dref/let={X=100}]{30/X}` ⇒ 0.3

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 clearly 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.

`\drefrel*[⟨opts⟩]{⟨key⟩}`
`\drefrel[⟨opts⟩]{⟨key⟩}`

The `\drefrel` macro is used to calculate relations between a base value and a concrete key. A prominent example of such a relation is the percent relation. `\drefrel` allows you to write down intentionally what relation you want to express without thinking about a concrete formula. The starred version of this macro does not print anything, but sets only `\drefresult`.

`\drefrel[base=/med A/mice count,factor]{/med A/recovered}`
⇒ 28.13

The type of relation can be manipulated with various keys. Almost always the given argument key will be set in relation to a base value. The type of relation can be given as well as post-processing steps.

Like `\drefcalc`, `\drefrel` sets the `\drefresult` macro accordingly.

`/dref/base`
`/dref/base plain`
`/dref/value plain`

This specifies the key that will be used as a base. Without the **base plain** option, the value will be interpreted as a symbolic datapoint. With the option, base contains the plain value. When **value plain** is given, the mandatory argument is interpreted as a number and not as a symbolic name.

`\drefrel[factor,base=50,base plain]{/med A/mice count}` ⇒ 0.64
`\drefrel[factor,base=50,base plain,value plain]{45}` ⇒ 0.9

`/dref/factor`

Is a base relation type, which cannot be mixed with other relation types. It simply divides the given value by the base value.

$$\mathtt{\backslash drefresult} = \frac{\mathtt{value}}{\mathtt{base}}$$

`/dref/increase`
`/dref/overhead`

Is a base relation type. It calculates the overhead factor a value show toward the base value. `increase` and `overhead` are synonyms.

$$\mathtt{\backslash drefresult} = \frac{\mathtt{value} - \mathtt{base}}{\mathtt{base}}$$

`\drefrel[overhead,base=50,base plain,value plain]{45}` ⇒ -0.1

`/dref/delta`

Is a base relation type. It calculates the difference between value and base.

$$\mathtt{\backslash drefresult} = \mathtt{value} - \mathtt{base}$$

`\drefrel[delta,base=50,base plain,value plain]{45}` ⇒ -5

`/dref/scale`
`/dref/product`

Is a base-relation type. It calculates the product of value and base.

$$\mathtt{\backslash drefresult} = \mathtt{value} \cdot \mathtt{base}$$

`\drefrel[scale,base=50,base plain,value plain]{45}` ⇒ 45

`/dref/percent`

Is a post-processing type. It calculates the percent value from a fraction.

$$\mathtt{\backslash drefresult} = \mathtt{\backslash drefresult} \cdot 100.0$$

`\drefrel[factor,percent,base=/med A/mice count]{/med A/recovered}`
⇒ 28.13

`/dref/abs`

Is a post-processing type. It takes the absolute value.

$$\mathtt{\backslash drefresult} =| \mathtt{\backslash drefresult} |$$

`\drefrel[overhead,abs,base=50,base plain,value plain]{45}` ⇒ 0.1

`/dref/negate`

Is a post-processing type. It negates the value.

$$\mathtt{\backslash drefresult} = \mathtt{\backslash drefresult} \cdot -1.0$$

`\drefrel[factor,negate,base=/med A/mice count]{/med A/recovered}`
⇒ -0.28

`/dref/divide`

Is a post-processing type. Divides the result by a contant factor. The argument must be a plain number.

$$\texttt{\textbackslash drefresult} = \texttt{\textbackslash drefresult} \cdot \{divide\}$$

`\drefrel[value plain,divide=1e6]{1453342654}` ⇒ 1,453.34

`\drefprojection{⟨from⟩}{⟨to⟩}{⟨projection⟩}`

Sometimes one or multiple sets of data have to be projected/mixed into a new set of data that is fully dependent on those values. This is achieved with `\drefprojection`. It projects one data set (subdirectoy) into another one. Tithin the projection three different operations are possible: `\id`, `\rename` and `\calc`.

identity function renaming of points 10

```
\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
`\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 a 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 symbolic name is expanded with `\drefvalueof`.

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 if your prose text relies on certain characteristics of the data. `\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 warning message is printed on the terminal.

`[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. The **pdfcomment** option uses pdf annotations. Be aware that those annotations work properlyy only on a few selected PDF readers[1]. `\drefannotate` sets the annoation style for the current group.

> `\drefannotate{none}`
> Black Six, 32, 33.33
>
> `\drefannotate{footnote}`
> Black Six[2], 32[3], 33.33[4]
>
> `\drefannotate{pdfcomment}`
> Black Six, 32, 33.33

`\drefusagereport`
`[usagereport]`
`[refall]`

With the **usagereport** package option enabled, `\drefusagereport` generates a usagereport of all referenced keys. The usage report groups the keys by the help texts. If the refall package option is given, all keys are marked as referenced.

Datagraphy

	Page	Value
/control group/mice race	2, 6, 7	Black Six
/projection/mice race	6	Black Six
The mice race used for experiments heavily influences the outcome of the results		

	Page	Value
/med A/recovered	2, 4, 5, 6	9

[1] In doubt use Acrobat
[2] `\dref*{/control group/mice race}`
[3] `\dref{/control group/mice count}`
[4] `\drefcalc{100/3}`

	Page	Value
/control group/recovered	4, 6	6
/control group/dead after 24h	6	3
/control group/dead after 48h	6	7
/med A/dead after 24h	6	6
/med A/dead after 48h	6	1
Of all infected mice, a certain number died within a specified period of time. A certain recovered from the infection. Each mouse is in exactly one category.		

Keys without Help	Page	Value
/control group/mice count	2, 6, 7	32
/med A/mice count	4, 5	32
/.DUMMY	5	1
/projection/died	6	10
/projection/count	6	32

3 Implementation

Guard against reading twice

```
\ifx\drefloaded\undefined
  \let\drefloaded=\relax
\else
  \expandafter\endinput
\fi
\ifx\PackageError\undefined
  \def\dref@error#1{\immediate\write-1{Package dref: Error! #1.}}%
\else
  \def\dref@error#1{\PackageError{dref}{#1}{}}%
\fi
% \end{macrocode}
%
% \begin{macrocode}
\RequirePackage{pgf}
\RequirePackage{kvoptions}
\usepgflibrary{fpu}
\usepackage{etoolbox}
\let\origforlistloop\forlistloop
\usepackage{etextools}
\let\forlistloop\origforlistloop
\RequirePackage{xcolor}

\SetupKeyvalOptions{
  family=dref,
  prefix=dref@
}
\DeclareStringOption[/data]{datapath}
\DeclareStringOption[1]{defaultvalue}
\DeclareStringOption[none]{annotate}
\DeclareBoolOption{usagereport}
\DeclareBoolOption{refall}
\DeclareBoolOption{ignoremissing}
\DeclareBoolOption{noassert}
\ProcessKeyvalOptions*
```

\dref@set

```
\def\dref@set#1#2{%
    \pgfkeys@temptoks{#2}%
    \expandafter\xdef\csname
    pgfk@\dref@datapath#1\endcsname{\the\pgfkeys@temptoks}%
    \ifdref@refall%
      \expandafter\dref@found\expandafter{\dref@datapath#1}{0}
      \expandafter\dref@referenced\expandafter{\dref@datapath#1}{0}%
    \fi%
}
```

\drefset

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

\dref@expandable

```
\def\dref@expandable#1{%
  \pgfkeysifdefined{\dref@datapath#1}{%
    \pgfkeysvalueof{\dref@datapath#1}%
  }{%
    \ifdref@ignoremissing%
      \dref@defaultvalue%
    \else%
      \typeout{Dref error: undefined key ‘#1’}\QUIT%
    \fi%
  }%
}
```

\dref@unexpandable

```
\def\dref@unexpandable#1{%
  \def\drefcurrentkey{\dref@datapath#1}%
  \pgfkeysifdefined{\drefcurrentkey}{%
    \edef\dref@thepage{\arabic{page}}%
    \immediate\write\@auxout{\noexpand\dref@found{\drefcurrentkey}{\dref@thepage}}%
  }{%
    \immediate\write\@auxout{\noexpand\dref@notfound{\drefcurrentkey}{\dref@thepage}}%
    \ifdref@ignoremissing%
       \typeout{Dref warning: undefined key ‘\drefcurrentkey’}%
       \dref@mkannotate{UNDEFINED: \drefcurrentkey}%
    \else%
       \dref@error{Dref error: undefined key ‘\drefcurrentkey’}%
    \fi%
  }%
  \immediate\write\@auxout{\noexpand\dref@referenced{\drefcurrentkey}{\dref@thepage}}%
}
```

\drefifdefined

```
\newcommand{\drefifdefined}[3]{
  \def\drefcurrentkey{\dref@datapath#1}%
  \pgfkeysifdefined{\drefcurrentkey}{#2}{#3}%
}
```

\dref

```
\def\dref{\@ifstar\@@dref\@dref}
\newcommand{\@dref}[2][]{% Unstarred
  \edef\dref@argument{#2}%
  \expandafter\dref@unexpandable\expandafter{\dref@argument}%
  \pgfmathparse{\dref@expandable{#2}}%
  \dref@format[#1]{\pgfmathresult}%
  \dref@mkannotate{\textbackslash dref\{#2\}}%
}
```

```
\newcommand{\@@dref}[2][]{% Starred
  \edef\dref@argument{#2}%
  \expandafter\dref@unexpandable\expandafter{\dref@argument}%
  \expandafter\gdef\expandafter\dref@dref@output\expandafter{\expandafter\dref@expandable\expand
  \dref@dref@output%
  \dref@mkannotate{\textbackslash dref*\{#2\}}%
}
```

\drefvalueof

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

\drefref

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

\dref@help@match

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

\dref@help

```
\newcommand{\dref@help}[2][]{%
  \pgfkeysifdefined{#2/help}{%
    \pgfkeysvalueof{#2/help}%
  }{#1}%
}
```

\drefsethelp

```
\csdef{dref@helps}{}
\newcommand{\drefsethelp}[2]{
  \csdef{dref@help@#1}{#2}%
  \listcsadd{dref@helps}{#1}%
}
```

\drefhelp

```
\newcommand{\drefhelp}[1]{
  \renewcommand{\do}[1]{%
    \dref@help@match{##1}{#1}{%
      \csuse{dref@help@##1}%
    \listbreak}{}%
  }%
  \ifcsvoid{dref@helps}{}{%
    \dolistcsloop{dref@helps}%
  }%
}
```

\dref@referenced

```
\def\dref@notfound#1#2{
  \ifdref@usagereport%
    \dref@usagereport@notfound{#1}{#2}%
  \else\relax\fi%
}
\def\dref@found#1#2{
  \ifdref@usagereport%
    \dref@usagereport@found{#1}{#2}%
  \else\relax\fi%
}
\def\dref@referenced#1#2{
  \ifdref@usagereport%
    \dref@usagereport@referenced{#1}{#2}%
  \else\relax\fi%
}
```

\dref@let

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

% \end{macro}
%
%
% \begin{macro}{\dreflet}
%    \begin{macrocode}
\def\dreflet#1{%
  \dref@let{#1}%
}
```

\drefcalc

```

\def\dref@parser@parse#1#2#3\@nnil{%
  %\typeout{’#1’ ’#2’ ’#3’}%
  \ifcsdef{dref@parser@#1@#2}{%
    \csuse{dref@parser@#1@#2}#3\@nnil%
  }{%
    #1#2\ifblank{#3}{}{%
      \dref@parser@parse{}#3\@nnil%
    }%
  }%
}

\csdef{dref@parser@@d}{\dref@parser@parse{d}}
```

```
\csdef{dref@parser@d@a}{\dref@parser@parse{da}}
\csdef{dref@parser@da@t}{\dref@parser@parse{dat}}
\csdef{dref@parser@dat@a}{\dref@parser@parse{data}}
\csdef{dref@parser@data@(}{\dref@parser@parse{data(}}
\csdef{dref@parser@data(@"}{\dref@parser@tillquote}
\csdef{dref@parser@d@(}{\dref@parser@tillparen}

\def\dref@parser@tillquote#1")#2\@nnil{%
  (\drefvalueof{\dref@data@math@prefix #1})\ifblank{#2}{}{\dref@parser@parse{}#2\@nnil}%
}
\def\dref@parser@tillparen#1)#2\@nnil{%
  (\drefvalueof{\dref@data@math@prefix #1})\ifblank{#2}{}{\dref@parser@parse{}#2\@nnil}%
}

\def\dref@parser@end#1#2\@nnil{}
\csdef{dref@parser@@}{\typeout{end}\dref@parser@end}

\newcommand{\dref@calc}[1]{%
  \xdef\dref@calc@argA{#1}%
  \xdef\dref@calc@@argA{\expandafter\dref@parser@parse%
    \expandafter{\expandafter}%
    \dref@calc@argA\@nnil}%%
  %\typeout{\dref@calc@@argA}%
  \pgfmathparse{\dref@calc@@argA}
}

\pgfset{/dref/let/.code={\dref@let{#1}}}
\def\drefresult{0}
\def\drefcalc{\@ifstar\@@drefcalc\@drefcalc}
\newcommand{\@drefcalc}[2][]{% Unstarred
  \begingroup%
  \pgfset{/pgf/number format/.cd, #1}%
  \dref@calc{#2}%
  \pgfmathprintnumberto[fixed,assume math mode=true,precision=10,1000 sep={}]{\pgfmathresult}{\
  \xdef\drefresult{\drefresult}%
  \dref@format{\pgfmathresult}%
  \dref@mkannotate{\textbackslash drefcalc\{#2\}}%
  \endgroup%
}
\newcommand{\@@drefcalc}[2][]{ % Starred
   \begingroup%
   \pgfset{/pgf/number format/.cd, #1}%
   \dref@calc{#2}%
   \pgfmathprintnumberto[fixed,assume math mode=true,precision=10,1000 sep={}]{\pgfmathresult}{
   \xdef\drefresult{\drefresult}%
   \endgroup%
}
```

\drefformat

```
\newcommand{\dref@format}[2][]{%
```

```
  \pgfmathprintnumber[#1]{#2}%
}
\newcommand{\drefformat}[2][]{\dref@format[#1]{#2}}
```

data()

```
\gdef\dref@data@math@prefix{}
\pgfmathdeclarefunction{data}{1}{%
        \begingroup%
                \dref@unexpandable{\dref@data@math@prefix#1}%
                \pgfmathparse{\dref@expandable{\dref@data@math@prefix#1}}%
                \pgfmath@smuggleone\pgfmathresult%
        \endgroup%
}
\long\def\drefprojection#1#2#3{%
  \begingroup%
     \def\dref@data@math@prefix{#1}%
     \def\rename##1##2{\dref@unexpandable{#1/##1}\drefset{#2/##2}{\dref@expandable{#1/##1}}}%
     \def\id##1{\rename{##1}{##1}}%
     \def\calc##1##2{%
       \begingroup%
          \drefcalc{##1}%
          \xdef\dref@project@result{\drefresult}
       \endgroup%
       \drefset{#2/##2}{\dref@project@result}%
      }%
     #3%
     \endgroup%
}
```

\dref@makerow

```

\newtoks\dref@toks
\newcount\drefcellcount

\newcommand{\dref@makerow}[2]{%
  {\global\dref@toks={}%
    \drefcellcount=\z@%
    \def\do##1{%
      \advance\drefcellcount\@ne%
      \def\@tempa{\doX{##1}}%
      \expandafter\@tempa\expandafter{\the\drefcellcount}%
    }%
    \def\doX##1##2{%
      \csxdef{@cell\the\drefcellcount}{\detokenize{%
          #2%
        }}%
    }%
    \expandafter\def\expandafter\arglist\expandafter{#1}%
    \expandafter\docsvlist\expandafter{\arglist}%
    \@tempcntb=0\relax
    {\loop\ifnum\@tempcntb<\drefcellcount
      \advance\@tempcntb by 1\relax%
      \ifnum \@tempcntb = 1%
        \edef\@@next{\csuse{@cell\the\@tempcntb}}%
      \else%
        \edef\@@next{&\csuse{@cell\the\@tempcntb}}%
      \fi%
      \global%
      \dref@toks%
      \expandafter=%
      \expandafter{%
        \the%
        \expandafter\dref@toks%
        \@@next}%
      \repeat}%
  }%
  \typeout{LINE: \the\dref@toks}%
  \expandafter\scantokens\expandafter{\the\dref@toks}}

\long\def\drefrow{\@ifstar\@@drefrow\@drefrow}
\def\@drefrow#1#2{\dref@makerow{#1}{\dref{#2}}} % Unstarred
\def\@@drefrow#1#2{\dref@makerow{#1}{#2}} % Starred
```

\dref@mkannotate

```

\expandafter\ifstrequal\expandafter{\dref@annotate}{pdfcomment}{
  \RequirePackage{pdfcomment}
}

\def\dref@mkannotate@none#1{\relax}
\def\dref@mkannotate@footnote#1{\footnote{\texttt{#1}}}
\def\dref@mkannotate@pdfcomment#1{\pdfcomment[opacity=0.4,voffset=2ex]{#1}}

\newcommand{\dref@mkannotate}[1]{%
  \ifcsdef{dref@mkannotate@\dref@annotate}{%
    \csuse{dref@mkannotate@\dref@annotate}{#1}%
    }{%
      \dref@error{Value for annotate not supported: '\dref@annotate'}%
    }%
}

\newcommand{\drefannotate}[1]{%
  \renewcommand{\dref@annotate}{#1}%
}
```

Usagereport

```
\ifdref@usagereport
  \RequirePackage{xtab}
  \RequirePackage{booktabs}
\fi
```

\dref@usagereport@referenced

```
\newcommand{\dref@usagereport@notfound}[2]{}
\newcommand{\dref@usagereport@found}[2]{}

\csdef{pgfdat@usagereport@keys}{}
\csdef{pgfdat@usagereport@matchedkeys}{}

\newcommand{\dref@usagereport@referenced}[2]{
  \ifinlistcs{#2}{dref@usagereport@referenced@#1}{}{
    \listcsgadd{dref@usagereport@referenced@#1}{#2}
  }
  \ifinlistcs{#1}{dref@usagereport@keys}{}{
    \listcsgadd{dref@usagereport@keys}{#1}
  }
}
```

\dref@usagereport@strippath

```
\expandafter\def\expandafter\dref@usagereport@strippath@\dref@datapath#1\blanktest{#1}

\newcommand{\dref@usagereport@strippath}[1]{%
  \expandafter\ifstrmatch\expandafter{\expandafter^\dref@datapath.*$}{#1}%
    {\dref@usagereport@strippath@#1\blanktest}%
    {#1}%
}
```

usagereport@formatreferencelist

```
\newcommand{\dref@usagereport@formatreferencelist}[1]{%
  \begingroup%
  \def\sep{}%
  \renewcommand{\do}[1]{\sep\ifdef{\hyperlink}{\hyperlink{page.##1}{##1}}{##1}\def\sep{, }}%
  \dolistcsloop{dref@usagereport@referenced@#1}%
  \endgroup%
}
```

\dref@usagereport@keyheader

```
\newif\ifdref@usagereport@keyheader@first
\dref@usagereport@keyheader@firsttrue
\newcommand{\dref@usagereport@keyheader}[1]{%
  \ifdref@usagereport@keyheader@first%
  \global\dref@usagereport@keyheader@firstfalse%
  \else%
    \\%
  \fi%
  \textbf{\ifdef{\hypertarget}%
    {\hypertarget{#1}{\dref@usagereport@strippath{#1}}}%
    {\dref@usagereport@strippath{#1}}}%
  & \dref@usagereport@formatreferencelist{#1}%
  & \pgfkeysifdefined{#1}{\pgfkeysvalueof{#1}}{\textbf{\color{red}undefined}}%
}
```

\dref@usagereport@forhelp

```
\newif\ifdref@withhelp
\errorcontextlines=23
\newlength{\dreflinewidth}%
\newcommand{\dref@usagereport@forhelp}[1]{%
  \begingroup%
  \dref@withhelpfalse%
  \renewcommand{\do}[1]{%
    \dref@help@match{#1}{##1}{%
      \dref@withhelptrue%
    }{}%
  }%
  \dolistcsloop{dref@usagereport@keys}%
  \dref@usagereport@keyheader@firsttrue%
  \renewcommand{\do}[1]{%
    \dref@help@match{#1}{##1}{%
      \dref@usagereport@keyheader{##1}%
      \ifinlistcs{##1}{dref@usagereport@matchedkeys}{}{%
        \listcsgadd{dref@usagereport@matchedkeys}{##1}%
      }%
    }{}%
  }%
  \ifdref@withhelp
    \tablehead{\hline      & Page  & Value \\\hline}%
    \setlength\tabcolsep{3pt}%
    \dreflinewidth=\linewidth%
    \advance\dreflinewidth by -6\tabcolsep%
    \begin{xtabular}{|p{0.7\dreflinewidth}|p{0.15\dreflinewidth}|p{0.15\dreflinewidth}|}%
      \dolistcsloop{dref@usagereport@keys}\\\hline
      \multicolumn{3}{|p{\linewidth}|}{\csuse{dref@help@#1}}\\\hline
    \end{xtabular}%
  \fi%

  \endgroup%
}
```

\dref@usagereport@withouthelp

```
\newif\ifdref@withouthelp
\newcommand{\dref@usagereport@withouthelp}{%
  \begingroup%
  \dref@withouthelpfalse%
  \renewcommand{\do}[1]{%
    \ifinlistcs{##1}{dref@usagereport@matchedkeys}{}{%
      \dref@withouthelptrue%
    }%
  }%
  \dolistcsloop{dref@usagereport@keys}%
  \renewcommand{\do}[1]{%
    \ifinlistcs{##1}{dref@usagereport@matchedkeys}{}{%
      \dref@usagereport@keyheader{##1}%
    }%
  }%
  \ifdref@withouthelp%
    \setlength\tabcolsep{0pt}%
    \tablehead{\toprule Keys without Help & Page & Value \\ \midrule}%
    \begin{xtabular}{p{0.7\linewidth}p{0.15\linewidth}p{0.15\linewidth}}
      \dolistcsloop{dref@usagereport@keys}\\
      \bottomrule
    \end{xtabular}%
   \fi%
   \endgroup%
}
```

\drefusagereport

```
\newcommand{\drefusagereport}{%
  \ifdref@usagereport%
  \ifcsvoid{dref@usagereport@keys}{\typeout{EMPTY}}{%
  \begingroup%
  \renewcommand{\do}[1]{%
    \ifinlistcs{##1}{dref@usagereport@matchedkeys}{}{%
      \dref@usagereport@forhelp{##1}%
    }%
  }%
  \dolistcsloop{dref@helps} % For all help text
  \dref@usagereport@withouthelp\relax
  \endgroup%
  }% csempty @keys
  \fi%
}
```

\drefassert

```
\newcommand{\drefassert}[1]{%
  \begingroup%
    \drefcalc*{#1}%
    \expandafter\ifstrequal\expandafter{\drefresult}{1}{%
      \typeout{Assertion holds: #1}%
    }{%
      \ifdref@noassert%
        \typeout{Assertion failed: #1}%
      \else%
        \dref@error{Assertion failed: #1}%
      \fi%
   }%
  \endgroup%
}
```

\drefrel

```
\newif\if@dref@valuemustderef%
\newif\if@dref@basemustderef%
```

```
\newif\if@dref@increase%
\newif\if@dref@product%
\newif\if@dref@factor%
\newif\if@dref@delta%
\newif\if@dref@percent%
\newif\if@dref@abs%
\newif\if@dref@neg%
\pgfkeys{%
  \dref@datapath/.DUMMY/.initial=1
}
\pgfkeys{%
  /dref/.cd,%
  value/.initial = /.DUMMY,%
  base/.initial  = /.DUMMY,%
  divide/.initial = 1,%
  value plain/.is if=@dref@valuemustderef,%
  value plain/.default=false,%
  value plain=true,%
  base plain/.is if=@dref@basemustderef,%
  base plain/.default=false,%
  base plain=true,%
  factor/.is if=@dref@factor,%
  factor/.default=true,%
  factor=false,%
  delta/.is if=@dref@delta,%
  delta/.default=true,%
  delta=false,%
  scale/.is if=@dref@product,%
  scale/.default=true,%
  scale=false,%
  product/.is if=@dref@product,%
  product/.default=true,%
  product=false,%
  increase/.is if=@dref@increase,%
  increase/.default=true,%
  increase=false,%
  overhead/.is if=@dref@increase,%
  overhead/.default=true,%
  overhead=false,%
  percent/.is if=@dref@percent,%
  percent/.default=true,%
  percent=false,%
  abs/.is if=@dref@abs,%
  abs/.default=true,%
  abs=false,%
  negate/.is if=@dref@neg,%
  negate/.default=true,%
  negate=false,%
}

```

```
\def\drefrel{\@ifstar\@@drefrel\@drefrel}

\newcommand{\@drefrel}[2][]{%
  \@@drefrel[#1]{#2}%
  \@@drefrel@result%
  \dref@mkannotate{\textbackslash{}drefrel[#1]\{#2\}}%
}

\newcommand{\@@drefrel}[2][]{%
  \begingroup%
  \pgfkeys{/pgf/fpu}%
  \pgfkeys{/dref/.cd,#1}%
  \pgfkeys{/dref/value=#2}%
  \if@dref@valuemustderef%
    \drefref{\pgfkeysvalueof{/dref/value}}%
    \edef\drefvalue{\drefvalueof{\pgfkeysvalueof{/dref/value}}}%
  \else%
    \def\drefvalue{\pgfkeysvalueof{/dref/value}}%
  \fi%
  \if@dref@basemustderef%
    \drefref{\pgfkeysvalueof{/dref/base}}%
    \def\drefbase{\drefvalueof{\pgfkeysvalueof{/dref/base}}}%
  \else%
    \def\drefbase{\pgfkeysvalueof{/dref/base}}%
  \fi%
  \xdef\drefresult{\drefvalue}%
  \if@dref@increase%
     \pgfmathparse{((\drefvalue) - (\drefbase)) / (\drefbase)}%
     \def\drefresult{\pgfmathresult}%
  \else%
     \if@dref@factor%
         \pgfmathparse{(\drefvalue) / (\drefbase)}%
         \def\drefresult{\pgfmathresult}%
     \else%
         \if@dref@delta%
             \pgfmathparse{(\drefvalue) - (\drefbase)}%
             \def\drefresult{\pgfmathresult}%
         \else%
            \if@dref@product%
               \pgfmathparse{(\drefvalue) * (\drefbase)}%
               \def\drefresult{\pgfmathresult}%
            \else
               \def\drefresult{\drefvalue}%
            \fi
         \fi%
     \fi%
  \fi%
  % Percent
  \if@dref@percent%
      \pgfmathparse{(\drefresult)*100.0}%
      \def\drefresult{\pgfmathresult}%
  \fi%
  % Absolute Value
  \if@dref@abs%
      \pgfmathparse{abs(\drefresult)}%
      \def\drefresult{\pgfmathresult}%
  \fi%
  % Negative Value
  \if@dref@neg%
      \pgfmathparse{-1.0*(\drefresult)}%
      \def\drefresult{\pgfmathresult}%
  \fi%
  \pgfmathparse{\drefresult/\pgfkeysvalueof{/dref/divide}}%
  \pgfmathprintnumberto[fixed,assume math mode=true,precision=10,1000 sep={}]{\pgfmathresult}{\
  \pgfmathprintnumberto{\pgfmathresult}{\@@drefrel@result}%
  \xdef\drefresult{\drefresult}%
  \xdef\@@drefrel@result{\@@drefrel@result}%
  \endgroup%
}

%
```