The subcaption package*

Axel Sommerfeldt
https://gitlab.com/axelsommerfeldt/caption

2023/02/19

Abstract

This package offers an user interface to typeset sub-captions.

⚠️ At the end of each section, text marked with the mountain symbol will contain background knowledge on how the particular command or environment is actually implemented. If you just want to use this package as it is, you don’t have to read or understand them.

⚠️ Since version 3.1 the caption package offers a low-level interface to typeset sub-captions: \DeclareCaptionSubType defines the required counters and internal commands, \setcaptionsubtype switches to the sub-caption mode, and \caption@subtypehook could be extended to apply own code when a switch to the sub-caption mode is in progress.

This package demonstrates its usage by offering a high-level user interface additionally.

*This package has version number v1.6.
Contents

1 Loading the package 4
2 Setting options 4
3 The \subcaptionbox command 5
   3.1 Comparison with \captionbox .......................... 8
4 The subcaptionblock environment 8
5 The subcaptiongroup environment 10
6 The \DeclareCaptionSubType command 12
7 Cross Referencing 13
   7.1 The \subref command .................................. 14
   7.2 The subrefformat= option .............................. 14
   7.3 Referencing sub-figures without sub-captions .......... 15
   7.4 Typesetting sub-captions without generating a (new) reference ....... 16
   7.5 Where to place the \label command? .................. 17
   7.6 Where do hyperlinks jump? ............................ 17
8 Captions inside sub-figures 18
   8.1 Using the overpic package ......................... 18
   8.2 Using the stackengine package ......................... 19
   8.3 Using the tikz package ................................ 20
   8.4 Using the skip and margin options .................. 22
9 Numbering 23
   9.1 Pitfall #1: Using multiple main captions ............. 23
   9.2 Pitfall #2: Expecting \caption to increment the counter ....... 25
10 Abbreviatory commands 25
11 The \subfloat command 26
12 Required packages 27
13 Other packages with similar offerings 27
14 Beyond this package 28
15 Thanks 28
### 16 The implementation

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.1 Identification</td>
<td>29</td>
</tr>
<tr>
<td>16.2 Initial code</td>
<td>29</td>
</tr>
<tr>
<td>16.3 Main code</td>
<td>29</td>
</tr>
<tr>
<td>16.4 Execution of options</td>
<td>34</td>
</tr>
<tr>
<td>16.5 Bonus material</td>
<td>35</td>
</tr>
</tbody>
</table>
1 Loading the package

Load this package using

\usepackage[⟨options⟩]{subcaption}.

The options for the subcaption package are the same ones as for the caption package, but specify settings which are used for sub-captions additionally. In fact

\usepackage[⟨options⟩]{subcaption}

is identical to

\usepackage{subcaption}
\subcaptionsetup{⟨options⟩}.

The default settings for sub-captions are:

margin=0pt,font+=smaller,labelformat=pars,labelsep=space,
skip=6pt,list=false,hypcap=false\footnote{This means that sub-captions are not listed in the List of Figures or Tables by default, but you can enable it by specifying the option list=true.}

2 Setting options

The \subcaptionsetup command sets options specifically for sub-captions.

\subcaptionsetup{⟨options⟩}

sets options valid for all sub-captions.

An example:

\subcaptionsetup{font=it}

sets the font to “italic” for all sub-captions.

\subcaptionsetup[⟨environment⟩]{⟨options⟩}

is an alternative syntax to

\captionsetup[sub⟨environment⟩]{⟨options⟩}.
An example:

\subcaptionsetup{table}{labelformat=simple}

is identical to

\captionsetup[subtable]{labelformat=simple}

and sets the labelformat to “simple” for all sub-captions inside tables.

Options specified with \usepackage{subcaption} and \subcaptionsetup{...} will override the ones specified by \captionsetup{...} and \captionsetup[figure]{...}, but are again overwritten by \subcaptionsetup[figure]{...} (same for ‘table’). So finally we have the following order how settings for sub-captions are applied:

1. Global settings (\usepackage{...}{caption} and \captionsetup{...})
2. Environmental settings (\captionsetup[figure-or-table]{...})
3. Local settings (\captionsetup{...} inside figure or table environment)
4. Default ‘sub’ settings (margin=0pt,font=smaller,...,see above)
5. Custom ‘sub’ settings (\usepackage{...}{subcaption} and \subcaptionsetup{...})
6. Environmental ‘sub’ settings (\subcaptionsetup[figure-or-table]{...})
7. Local ‘sub’ settings (\captionsetup{...} inside subfigure or subtable)

An example:

\usepackage[labelsep=quad,indentation=10pt]{caption}
\usepackage[labelfont=bf,list=true]{subcaption}
\captionsetup[table]{textfont=it,position=top}
\subcaptionsetup[table]{textfont=sf}

causes sub-captions inside table environments to be typeset with the settings

indentation=10pt,position=top,margin=0pt,font=smaller,labelformat=parses,labelsep=space,skip=6pt,hypcap=false,labelfont=bf,list=true,textfont=sf.

3 The \subcaptionbox command

\subcaptionbox The \subcaptionbox command typesets given content and caption. It automatically aligns the sub-figures resp. sub-tables by their very first caption line.
Its syntax is:

\subcaptionbox\[⟨list entry⟩\]{{{heading}}}\[⟨width⟩\][⟨inner-pos⟩]{{{contents}}}\]
\subcaptionbox*{⟨heading⟩}[[width]][[inner-pos]]{{contents}}

The arguments \(⟨list\ entry⟩\) & \(⟨heading⟩\) will be used for typesetting the \caption. \(⟨width⟩\) is the width of the resulting \parbox; the default value is the width of the contents. \(⟨inner-pos⟩\) specifies how the contents will be justified inside the resulting \parbox; it can be either ‘c’ (centered, using \centering), ‘l’ (left-justified, using \raggedright), ‘r’ (right-justified, using \raggedleft), or ‘s’ (for no special justification). The default is ‘c’. (But you can use any justification defined with \DeclareCaptionJustification as well, e.g.: ‘centerlast’)

When using \subcaptionbox, the baseline of the resulting box will be placed right between contents and heading. So usually you don’t have to care about the vertical alignment of the sub-figures for yourself. Also the hyperlink anchor is placed properly with respect to the hypcap= setting.

An example:

\begin{figure}
\centering
\subcaptionbox{A cat\label{cat}}{\includegraphics{cat}}
\subcaptionbox{An elephant\label{elephant}}{\includegraphics{elephant}}
\caption{Two animals}\label{animals}
\end{figure}

(a) A cat  (b) An elephant

Figure 1: Two animals

As you see the result is not satisfying; the caption below the cat looks ugly because of the small width of the graphic. This can be solved by using the optional arguments of \subcaptionbox, increasing the width of the resulting box:

---

\textsuperscript{2}The pictures were taken with permission from the \LaTeX\ Companion\cite{1} examples.
Furthermore the main caption, which is centered with respect to the \textwidth, looks mis-aligned with respect to the sub-captions. This can (again) be solved by using the optional arguments of \subcaptionbox, giving both boxes the same width, for example:

\subcaptionbox{A cat\label{cat}}{2.5cm}{\includegraphics{cat}}
\subcaptionbox{An elephant\label{elephant}}{.4\textwidth}{\includegraphics{elephant}}

(a) A cat  (b) An elephant

Figure 3: Two animals

The \subcaptionbox is a \parbox with \setcaptionsubtype as first contents line.
3.1 Comparison with \captionbox

Both, \captionbox (offered by the caption package) and \subcaptionbox, put its contents and caption into a \parbox of either natural or given width and share the same (mandatory and optional) arguments, but while \captionbox uses a regular caption, \subcaptionbox uses a sub-caption instead, like “(a)” or “2.1”. So for example the last example would look like this when using \captionbox instead of \subcaptionbox:

\begin{figure}
  \centering
  \captionbox{A cat\label{cat}}\[.4\textwidth\]{\includegraphics{cat}}%  \\
  \captionbox{An elephant\label{elephant}}\[.4\textwidth\]{\includegraphics{elephant}}
\end{figure}

Figure 4: A cat  
Figure 5: An elephant

4 The subcaptionblock environment

The subcaptionblock environment makes a box with given width. Inside this box the regular caption commands (like \caption, \phantomcaption, ...) could be used to typeset sub-captions. subcaptionblock has the same (optional & mandatory) arguments as the minipage environment:

\begin{subcaptionblock}{⟨outer-pos⟩}{⟨height⟩}{⟨inner-pos⟩}{⟨width⟩}...
\end{subcaptionblock}

The default value for ⟨outer-pos⟩ is either ‘b’ (if no main caption was typeset so far) or ‘t’ (if a main caption was already typeset), and the default value for ⟨inner-pos⟩ is ‘s’.

Beside the ⟨outer-pos⟩ values of ‘c’, ‘t’, and ‘b’, the subcaption package also offers the values ‘T’ and ‘B’ additionally which align the contents at the very top resp. bottom. (In contrast ‘t’ and ‘b’ align the contents at the top resp. bottom baseline.)
The same example as Figure 3, but this time using the \texttt{subcaptionblock} environment instead of \texttt{subcaptionbox}:

\begin{figure}
\centering
\begin{subcaptionblock}{.4\textwidth}
\centering
\includegraphics{cat}
\caption{A cat}\label{cat}
\end{subcaptionblock}\
\begin{subcaptionblock}{.4\textwidth}
\centering
\includegraphics{elephant}
\caption{An elephant}\label{elephant}
\end{subcaptionblock}
\caption{Two animals}\label{animals}
\end{figure}

(a) A cat
(b) An elephant

Figure 6: Two animals

Some additional notes:

- You can override the settings for a specific subcaption with a \texttt{captionsetup} inside the \texttt{subcaptionblock}, e.g.:

\begin{subcaptionblock}{.4\textwidth}
\centering
\includegraphics{owl}
\captionsetup{skip=3pt}
\caption{An owl}\label{owl}
\end{subcaptionblock}

- Just like \texttt{figure} or \texttt{table}, a \texttt{subcaptionblock} could have multiple captions, e.g.:

\begin{subcaptionblock}{.4\textwidth}
\centering
\includegraphics{cat}
\caption{A cat}\label{cat}
\end{subcaptionblock}
Hyperlinks targeted to this sub-figure will jump to the beginning of the subcaptionblock, and not to the \caption inside it (if hypcap=true is set for sub-captions). (See section 7.6: Where do hyperlinks jump?)

The subcaptionblock environment is also offered as subfigure resp. subtable. (And prior version 1.5 of the subcaption package it was only available as subfigure resp. subtable.)

There is no difference in them except the environment name should match the current floating environment, i.e. inside a figure a subfigure should be used, and inside a table a subtable should be used. Using the wrong sub-environment will cause a warning since v1.5 of the subcaption package.

So if in doubt, or when writing own \LaTeXX commands which should work in every floating environment, using subcaptionblock is the correct choice.

(Note that the default value for (outer-pos) has changed from ‘c’ to either ‘b’ or ‘t’ in version 1.5 of the subcaption package.)

The subcaptionblock, subfigure, and subtable environments are minipage environments with \setcaptionsubtype as first contents line.

subfigure and subtable are defined with the help of \ForEachCaptionSubType offered by the caption package, which executes code for every sub-type declared with \DeclareCaptionSubType.

5 The \texttt{subcaptiongroup} environment

The subcaptiongroup environment is only switching to the sub-caption mode inside an own \LaTeXX group. Inside this environment the regular caption commands (like \caption, \phantomcaption,...) could be used to typeset sub-captions.

Its syntax is:

\begin{subcaptiongroup}
...
\end{subcaptiongroup}

There is a starred variant of this environment as well which uses \setcaptionsubtype* instead of \setcaptionsubtype internally:

\begin{subcaptiongroup*}
...
\end{subcaptiongroup*}

While this gives you great flexibility, it also offers you no help formatting its contents.
The same example as Figure 6, but this time using the `subcaptiongroup` environment instead of `subcaptionblock`:

\begin{figure}
\centering
\begin{subcaptiongroup}
\centering
\parbox[b]{.4\textwidth}{% 
\centering
\includegraphics{cat}
\caption{A cat}\label{cat}}%
\begin{minipage}[b]{.4\textwidth}
\centering
\includegraphics{elephant}
\caption{An elephant}\label{elephant}
\end{minipage}
\end{subcaptiongroup}
\caption{Two animals}\label{animals}
\end{figure}

–or–

\begin{figure}
\centering
\begin{subcaptiongroup}
\centering
\begin{minipage}[b]{.4\textwidth}
\centering
\includegraphics{cat}
\caption{A cat}\label{cat}
\end{minipage}%
\begin{minipage}[b]{.4\textwidth}
\centering
\includegraphics{elephant}
\caption{An elephant}\label{elephant}
\end{minipage}
\end{subcaptiongroup}
\caption{Two animals}\label{animals}
\end{figure}
The subcaptiongroup environment is a \LaTeX{} environment with \texttt{\setcaptionsubtype} as first contents line.

6 The \texttt{\DeclareCaptionSubType} command

\texttt{\ DeclareCaptionSubType} For using the sub-caption feature of the \texttt{caption} package some commands and counters must be prepared. This is done with

\begin{verbatim}
\DeclareCaptionSubType[⟨numbering scheme⟩]{⟨type⟩}
\DeclareCaptionSubType*[⟨numbering scheme⟩]{⟨type⟩}
\end{verbatim}

For the environments \texttt{figure} & \texttt{table}, and all the ones defined with \texttt{\ DeclareFloatingEnvironment} offered by the \texttt{newfloat} package, this will be done automatically, but for other environments (e.g. the ones defined with \texttt{\ newfloat} offered by the float package or \texttt{\ DeclareNewFloatType} offered by the floatrow package) this has to be done manually.

The starred variant provides the sub-caption numbering format \texttt{⟨type⟩.⟨subtype⟩} (for example ‘1.2’) while the non-starred variant simply uses \texttt{⟨subtype⟩} (for example ‘a’).

Own numbering formats can be created by redefining \texttt{\thesub⟨type⟩}, e.g.:

\begin{verbatim}
\DeclareCaptionSubType*[figure]
\renewcommand\thesubfigure{\thefigure\alph{subfigure}}
\end{verbatim}

would give you sub-caption numbers like ‘1b’.

The default numbering scheme is \texttt{alph}, but you can use any \LaTeX{} (or self-defined) command name here which converts a counter to a text value, e.g.: \texttt{arabic}, \texttt{roman}, \texttt{Roman}, \texttt{alph}, \texttt{Alph}, \texttt{fnsymbol}, ...

But \texttt{\DeclareCaptionSubType} is not only for defining new sub-caption types, you can use this command for re-definitions as well, e.g.:

\begin{verbatim}
\DeclareCaptionSubType*[arabic]{table}
\subcaptionsetup{table}{labelformat=simple,labelsep=colon}
\end{verbatim}
will give you sub-captions in tables like these ones:

<table>
<thead>
<tr>
<th>1.1: Table one</th>
<th>1.2: Table two</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B</td>
<td>E F</td>
</tr>
<tr>
<td>C D</td>
<td>G H</td>
</tr>
</tbody>
</table>

\texttt{\DeclareCaptionSubType} and \texttt{\ForEachCaptionSubType} are integral parts of the caption package kernel.

7 Cross Referencing

The macro \texttt{\the\langle\text{counter}\rangle} is not only responsible for the look of the \texttt{\langle\text{counter}\rangle}, but for the look of the references typeset with \texttt{\ref}, too. References will be prefixed by \texttt{I\TeX} with the internal macro \texttt{\p@\langle\text{counter}\rangle}.

\texttt{\DeclareCaptionSubType} will define both of them for sub-captions (e.g. sub-figure and subtable), and as you have seen in the last section \texttt{\DeclareCaptionSubType} will give you some options to control the internal (re-)definition of \texttt{\the\langle\text{counter}\rangle} and \texttt{\p@\langle\text{counter}\rangle}.

\texttt{\thesubfigure} For example \texttt{\thesubfigure} and \texttt{\p@subfigure} are (as default) internally defined \texttt{\p@subfigure} as

\begin{verbatim}
newcommand\thesubfigure{\alph{subfigure}}
newcommand\p@subfigure{\thefigure}
\end{verbatim}

so the label of sub-captions will look like ‘a’ (decorated by the selected label format), while references will look like ‘1a’ since they are prefixed by \texttt{\p@subfigure = \thefigure}.

After \texttt{\DeclareCaptionSubType*[arabic]{figure}}, \texttt{\thesubfigure} and \texttt{\p@subfigure} will look like

\begin{verbatim}
renewcommand\thesubfigure{\thefigure.\arabic{subfigure}}
renewcommand\p@subfigure{}
\end{verbatim}

But if you want detailed control on how the references will look like, the options of \texttt{\DeclareCaptionSubType} are potentially not sufficient. In this case one need to redefine these two macros on his/her own. Some examples:

If you want parentheses around the sub-figure part of the reference, so they will look like ‘1(a)’, you may get them this way:

\begin{verbatim}
usepackage[labelformat=simple]{subcaption}
renewcommand\thesubfigure{((\alph{subfigure}))}
\end{verbatim}

(\textbf{Note:} Since \texttt{parens} is the default label format you will get double parentheses in sub-captions when not specifying a different label format, e.g. \texttt{simple}.)

But if you want only a closing parenthesis, so references will look like ‘1a)’, but the sub-captions itself should still look like ‘(a)’, this would be a possible solution:
\usepackage{subcaption}
\renewcommand\thesubfigure{\textup{\alph{subfigure})}}
\DeclareCaptionLabelFormat{opening}{(#2}
\subcaptionsetup[figure]{labelformat=opening}

Please note that you need to surround redefinitions of \p@⟨counter⟩ with \makeatletter and \makeatother. See \url{http://tex.stackexchange.com/questions/8351/} for details.

7.1 The \subref command

While \ref{⟨key⟩} (and \ref*{⟨key⟩}, if the hyperref package is used) usually gives a combined result representing the main caption counter and the sub-caption one, it is sometimes useful to have a reference to the sub-caption only. For this purpose you can use

\subref{⟨key⟩}
\subref*{⟨key⟩}

So for example \ref{cat} gives the result ‘1a’ but \subref{cat} gives ‘a’.

Note: If the sub-caption was (re-)defined with the starred variant \DeclareCaptionSubType*, both \ref and \subref usually gives the same result.

The \subref command demonstrates the usage of \caption@subtypehook which will be called during \captionsetup{subtype}.

7.2 The subrefformat= option

By applying \DeclareCaptionSubType, or by redefining \the⟨counter⟩ and \p@⟨counter⟩, you will change the look of references typeset with \ref and \subref.

But maybe you only want to change the output of \subref without affecting the references typeset with \ref?

This is possible, too, by using the option subrefformat:

\captionsetup{subrefformat={⟨label format⟩}}

This one will choose a label format (either a pre-defined one, or a one defined with \DeclareCaptionLabelFormat) as decorative element to sub-references. The default one is simple which has no decorative elements but simply typeset the reference as it is.

For example

\captionsetup{subrefformat=parsens}

will result in references (typeset with \ref) like ‘1a’ but sub-references (typeset with \subref) like ‘(a)’.
7.3 Referencing sub-figures without sub-captions

\phantomcaption  If you don’t want to give a sub-figure a caption (yet), because the picture itself already contains the caption, or for some other reason, you could use the command

\phantomcaption

instead of \caption. \phantomcaption is offered by the caption package since version 3.2 and does not generate any output but increases the sub-figure resp. sub-table counter and gives you an anchor for a \label command which can be placed after it.

An example:

\begin{figure}
  \centering
  \begin{subcaptiongroup}
    \includegraphics{cat_with_a}
    \phantomcaption\label{cat}
    \includegraphics{elephant_with_b}
    \phantomcaption\label{elephant}
  \end{subcaptionblock}
  \captionsetup{subrefformat=parens}
  \caption{Two animals: \subref{cat} a cat, and \subref{elephant} an elephant}
  \label{animals}
\end{figure}

(a) (b)

Figure 8: Two animals: (a) a cat, and (b) an elephant

\captionlistentry  If you don’t want to give a sub-figure a caption (yet), because the picture itself already contains the caption, or for some other reason, you could also use the command

\captionlistentry{⟨list entry⟩}

instead of \caption. \captionlistentry is offered by the caption package since version 3.3 and (just like \phantomcaption) does not generate any output but increases the sub-figure resp. sub-table counter and gives you an anchor for a \label command which can be placed after it. Additionally to \phantomcaption this command puts an entry into the list of figures resp. tables.

\footnote{Like \ref*, \subref* is only available if the hyperref package\cite{hyperref} is used.}
An example:

\begin{figure}
\centering
\begin{subcaptiongroup}
\includegraphics{cat_with_a}
\captionlistentry{A cat}
\label{cat}
\includegraphics{elephant_with_b}
\captionlistentry{An elephant}
\label{elephant}
\end{subcaptionblock}
\captionsetup{subrefformat=parens}
\caption{Two animals: \subref{cat} a cat, and \subref{elephant} an elephant}
\label{animals}
\end{figure}

7.4 Typesetting sub-captions without generating a (new) reference

The \caption command is a multi-purpose command:

1. It increments the sub-figure resp. sub-table counter and generates an internal reference which could be used with \label
2. It puts an entry into the list of figures resp. tables
3. It finally typesets a caption

When put into a command or into an environment which either evaluates its content more than once or does not like one of the first two actions (for whatever reason), the result could be either an error message or an incorrect result, for example a sub-figure resp. sub-table counter which was incremented more than once.

In these cases the \caption command could be split into \phantomcaption which performs step one only (or \captionlistentry which performs steps one and two), and \captiontext which performs step three only. This way critical steps could be out-sourced from the target command or environment, for example by prepending \phantomcaption and using \captiontext inside.

The syntax of \captiontext is

\captiontext\[(\text{number})\]{\langle\text{text of sub-caption}\rangle}
\captiontext*\[(\text{number})\]{\langle\text{text of sub-caption}\rangle}

\captiontext is offered by the caption package since version 3.6 and as opposite to \caption it does not increase the sub-figure resp. sub-table counter and does not give you an anchor for a \label command. It typesets the caption only, using existing counter values unless a \langle number\rangle is given explicitly.

For example code please take a look at section 8: Captions inside sub-figures.
7.5 Where to place the \label command?

When \caption inside a captionsubblock, subfigure, subtable, or caption-subgroup environment, the \label can be either placed inside the caption text or right after the command, e.g.:

\caption{Some text here\label{text}}
...
\caption{Some other text}\label{othertext}
...
\caption{Something completely different}
\label{differenttext}

When using \phantomcaption or \captionlistentry inside a captionsubblock, subfigure, subtable, or captionsubgroup environment, the \label should be placed right after the command, e.g.:

\phantomcaption\label{this}
...
\phantomcaption
\label{that}

But when using the \subcaptionbox command, the \label should be placed inside the caption text, e.g.:

\subcaptionbox{A description here\label{todo1}}
\{Some content here\}
...
\subcaptionbox[List-of-Figures entry]{A description here\label{todo2}}
\{Some content here\}

Placing \label outside the \subcaptionbox would produce an incorrect reference.

7.6 Where do hyperlinks jump?

For the captionsubblock, subfigure, subtable, and captionsubgroup environments, and for the \subcaptionbox command (and all other constructs which use \setcaptionsubtype) the hyperlink anchors will be placed in respect to the hypcap= setting. While usage of this option is straight-forward for ordinary captions, the usage for sub-captions depends on the setting regarding the main captions.

This table gives you an overview where the hyperlinks will jump:

<table>
<thead>
<tr>
<th>hypcap= \null</th>
<th>hypcap=true</th>
</tr>
</thead>
<tbody>
<tr>
<td>hypcap= \null</td>
<td>sub-caption</td>
</tr>
<tr>
<td>sub-caption</td>
<td>figure or</td>
</tr>
<tr>
<td></td>
<td>sub-figure</td>
</tr>
<tr>
<td></td>
<td>sub-figure</td>
</tr>
<tr>
<td></td>
<td>sub-table</td>
</tr>
<tr>
<td></td>
<td>sub-table</td>
</tr>
</tbody>
</table>
But if the captionsubgroup environment is used (or a different construct which uses \setcaptionsubtype* and hypcap=true is set for sub-captions, the sub-caption package does not know where the sub-figure or sub-table actually begins, so it will jump to the sub-caption instead.

Remember: If you use the hypcap package[5], it controls the placement of the hyperlink anchors, making the rules above invalid.

(See also the documentation of the caption package, sections about hyperref & hypcap.)

8 Captions inside sub-figures

Nearly all code examples so far have placed the caption either above or below the sub-figure. But it is possible to put the caption inside the sub-figure, too:

• The caption could already be part of the image. (This case is already handled in section 7.3: Referencing sub-figures without sub-captions.)

• \LaTeXX packages like overpic, stackengine, or tikz could be used. Note that the commands resp. environments offered by these packages usually evaluate their content more than once, resulting in either errors or wrong reference counters. For this reason \caption should not be used here, instead \phantomcaption or \captionlistentry should be used outside the command resp. environment and \captiontext should be used inside it. See also section 7.4: Typesetting sub-captions without generating a (new) reference

• The options skip= and margin= could be used to place the caption onto the image.

8.1 Using the overpic package

The same example as in section section 7.3: Referencing sub-figures without sub-captions, but using the overpic environment offered by the overpic package [7] to place the captions inside the pictures:

\usepackage{overpic}
...
\begin{figure}
  \centering
  \begin{subcaptiongroup}
  \captionlistentry{A cat}
  \label{cat}
  \begin{overpic}[width=60pt]{cat}
    \put(40,34){\captiontext*{}}
  \end{overpic}
  \captionlistentry{An elephant}
  \label{elephant}
  \begin{overpic}[width=140pt]{elephant}
    \captiontext*{}
  \end{overpic}
\end{subcaptiongroup}
\end{figure}
If neither a reference to the sub-figures nor an entry in the List of Figures is needed, the usage of \phantomcaption resp. \captionlistentry could be dropped. Since this leaves \captiontext without a valid sub-figure number value it must be given to it explicitly as optional argument.

Furthermore the usage of the subcaptiongroup environment could be dropped here, instead it would be sufficient to replace \captiontext with \subcaptiontext. (See section 10: Abbreviatory commands)

8.2 Using the stackengine package

The same example as in section section 7.3: Referencing sub-figures without sub-captions, but using the \stackinset command offered by the stackengine package [8] to place the captions inside the pictures:
\usepackage{stackengine}
...
\begin{figure}
\centering
\begin{subcaptiongroup}
\subcaptionlistentry{A cat}
\label{cat}
\stackinset{l}{25pt}{b}{20pt}{\captiontext*{}}
{\includegraphics[width=60pt]{cat}}
\subcaptionlistentry{An elephant}
\label{elephant}
\stackinset{l}{80pt}{b}{60pt}{\captiontext*{}}
{\includegraphics[width=140pt]{elephant}}
\end{subcaptiongroup}
\captionsetup{subrefformat=parens}
\caption{Two animals: \subref{cat} a huge cat, and \subref{elephant} an elephant}
\end{figure}

Figure 10: Two animals: (a) a huge cat, and (b) an elephant

If neither a reference to the sub-figures nor an entry in the List of Figures is needed, this code could be simplified to:
\usepackage{stackengine}
...
\begin{figure}
\centering
\stackinset{l}{25pt}{b}{20pt}{\subcaptiontext*[1]{}}
{\includegraphics[width=60pt]{cat}}
\stackinset{l}{80pt}{b}{60pt}{\subcaptiontext*[2]{}}
{\includegraphics[width=140pt]{elephant}}
\caption{Two animals: A huge cat and an elephant}
\end{figure}

8.3 Using the tikz package

The same example as in section section 7.3: Referencing sub-figures without sub-captions, but using the tikzpicture environment offered by the tikz package [9] to
place the captions inside the pictures:

\usepackage{tikz}
...
\begin{figure}
\centering
\begin{subcaptiongroup}
  \subcaptionlistentry{A cat}
  \label{cat}
  \begin{tikzpicture}
    \node (cat) at (0,0)
    {
      \includegraphics[width=60pt]{cat}};
    \node at (0.1,-0.1) \captiontext*{};
  \end{tikzpicture}
  \subcaptionlistentry{An elephant}
  \label{elephant}
  \begin{tikzpicture}
    \node (elephant) at (0,0)
    {
      \includegraphics[width=140pt]{elephant}};
    \node at (0.5,-0.1) \captiontext*{};
  \end{tikzpicture}
\end{subcaptiongroup}
\captionsetup{subrefformat=parens}
\caption{Two animals: \subref{cat} a huge cat, and \subref{elephant} an elephant}
\end{figure}

Figure 11: Two animals: (a) a huge cat, and (b) an elephant

If neither a reference to the sub-figures nor an entry in the List of Figures is needed, this code could be simplified to:

\usepackage{tikz}
...
\begin{figure}
\centering
\begin{tikzpicture}
\node (cat) at (0,0)
{
  \includegraphics[width=60pt]{cat}};
\node at (0.1,-0.1) \captiontext*{};
\end{tikzpicture}
\node (elephant) at (0,0)
{
  \includegraphics[width=140pt]{elephant}};
\node at (0.5,-0.1) \captiontext*{};
\end{figure}
8.4 Using the skip and margin options

For a particular sub-caption the `skip=...` could be set to a negative value so it will overlap with the image. Combined with `singlelinecheck=off` (to switch off the centering of short captions) and `margin=...` it could be placed at a specific horizontal position within the image, too.

The same example as in section section 7.3: Referencing sub-figures without sub-captions, but using the `skip=...` and `margin=...` options:

```
\begin{figure}
  \centering
  \subcaptionsetup[figure]{skip=-28pt,slc=off,margin={25pt,0pt}}
  \subcaptionbox{\label{cat}}{\includegraphics[width=60pt]{cat}}
  \subcaptionsetup[figure]{skip=-60pt,slc=off,margin={80pt,0pt}}
  \subcaptionbox{\label{elephant}}{\includegraphics[width=140pt]{elephant}}
  \captionsetup{subrefformat=parens}
  \caption{Two animals: \subref{cat} a huge cat, and \subref{elephant} an elephant}
\end{figure}
```

![Figure 12: Two animals: (a) a huge cat, and (b) an elephant](image)

22
9 Numbering

9.1 Pittfall #1: Using multiple main captions

When multiple main captions are used within a figure or table, and sub-captions are used as well, how does the subcaption package know which sub-captions belong to which caption, i.e. what is the main counter value for the sub-captions?

Let’s illustrate this problem with an example document:

\documentclass{article}
\usepackage{graphicx,subcaption}
\begin{document}
\begin{figure}
\centering
\includegraphics[width=4cm]{example-image-c}
\caption{Caption no. 1}
\bigskip
\subcaptionbox{\label{fig:2a}}{\includegraphics[width=2cm]{example-image-a}}
\subcaptionbox{\label{fig:2b}}{\includegraphics[width=2cm]{example-image-b}}
\caption{Caption no. 2}
\end{figure}

Look at sub-figures \ref{fig:2a} and \ref{fig:2b}.

It’s obvious that the sub-captions belong to 2nd caption, and therefore \ref{fig:2a} will become “2a”, isn’t it? But since the subcaption package is only involved when using LATEX commands either defined or patched by the caption or subcaption package, this is what the caption package is aware of:

\usepackage{subcaption}
\begin{document}
\begin{figure}
\caption{Caption no. 1}
\subcaptionbox{\label{fig:2a}}{(unknown content)}
\subcaptionbox{\label{fig:2b}}{(unknown content)}
\caption{Caption no. 2}
\end{figure}
\end{document}

So from captions point of view it’s not easy to decide if the sub-captions belong to the 1st or 2nd main caption since they are placed between them. (Note: The subcaption package is only offering an user interface to the sub-caption feature of the caption package, and therefore this decision is the responsibility of the caption package.)
But how does the caption package make a decision? If in doubt, it rather clings to the past than to the future, i.e. in this case it decides that the sub-captions belong to the 1st caption and therefore the result of \ref{fig:2a} is not “2a” but “1a”. If the caption package is unsure about its decision (like in this case), a warning will be issued:

Package caption Warning: Ambiguous sub-caption(s), use \nextfloat on input line 15.

See the caption package documentation for explanation.

How to fix it? Use \nextfloat to tell the caption package where the 2nd figure within the figure environment starts:

\documentclass{article}
\usepackage{graphicx,subcaption}
\begin{document}
\begin{figure}
\centering
\includegraphics[width=4cm]{example-image-c}
\caption{Caption no. 1}
\bigskip
\nextfloat
\subcaptionbox[\label{fig:2a}]{\includegraphics[width=2cm]{example-image-a}}
\subcaptionbox[\label{fig:2b}]{\includegraphics[width=2cm]{example-image-b}}
\caption{Caption no. 2}
\end{figure}
Look at sub-figures \ref{fig:2a} and \ref{fig:2b}.
\end{document}

Note that the \nextfloat command was introduced in caption package v3.6. Previous versions of the caption package have bound the decision to the position= setting of the floating environment instead, i.e. sub-captions belonged to the caption above if position=top was set and they belonged to the caption below if position=below was set. While this would be beneficial in this case, there were several cases were it was not and especially it was not always comprehensible to the user why his references got an incorrect numbering.

Furthermore the new decision algorithm always succeeds if there is only one caption within the figure or table (which is the case most of the time) while the old one did not. If you still prefer the old decision algorithm (for example because you want to process an already existing document), you need to specify the caption package version explicitly, for example:

\documentclass{article}
\usepackage{caption}[=v3.5]
\usepackage{graphicx,subcaption}
...

See also: section 12: Required packages
9.2 Pittfall #2: Expecting \caption to increment the counter

Usually \caption increments the figure resp. table counter and therefore it is usually safe to assume that the counter was not incremented yet in code used before \caption:

\documentclass{article}
\begin{document}
\begin{figure}
  Figure counter before caption: \thefigure
  \caption{Some text}
  Figure counter after caption: \thefigure
\end{figure}
\end{document}

Unsurprisingly the results are 0 and 1 for the counter values.
But this happens if we use the subcaption package:

\documentclass{article}
\usepackage{subcaption}
\begin{document}
\begin{figure}
  \subcaptionbox{}{some content}
  \subcaptionbox{}{some content}
  Figure counter before caption: \thefigure
  \caption{Some text}
  Figure counter after caption: \thefigure
\end{figure}
\end{document}

Here the results are 1 and 1 for the counter values. Why?
Since the caption package assumes that there will be a \caption following \subcaptionbox it decides that the sub-captions belong to the upcoming main caption and therefore share the same main counter value which needs to be incremented before its use. Therefore the first \subcaptionbox in the figure increments the figure counter while the second one and especially the \caption does not.
Usually this is no problem at all but keep this in mind if you are doing tricky stuff with the figure resp. table counter within figures resp. tables.

10 Abbreviatory commands

As we have seen in sections section 8.1: Using the overpic package, section 8.2: Using the stackengine package, and section 8.3: Using the tikz package it’s sometimes inconvenient to use the commands and environments described so far.
For example it would be inconvenient to use captiongroup just for a single \caption-text in this particular case:

\begin{overpic}[width=60pt]{cat}
  \put(40,34)
Since `\captiontext` is already used in an extra environment (\overpic) we don’t need an extra \captiongroup* here, instead using `\setcaptionsubtype*` (which switches into the sub-caption mode without making an \hyperref anchor) would be sufficient:

```latex
\begin{overpic}[width=60pt]{cat}
  \put(40,34){\setcaptionsubtype* \captiontext*[1]{}}
\end{overpic}
```

But this is still inconvenient when used many times, and therefore the subcaption package defines several extra commands which are prefixed with `\setcaptionsubtype*`:

<table>
<thead>
<tr>
<th>Regular command</th>
<th>using <code>\setcaptionsubtype*</code> available since</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\caption</code></td>
<td><code>\subcaption</code> v1.0</td>
</tr>
<tr>
<td><code>\phantomcaption</code></td>
<td><code>\phantomsubcaption</code> v1.1</td>
</tr>
<tr>
<td><code>\captionlistentry</code></td>
<td><code>\subcaptionlistentry</code> v1.5</td>
</tr>
<tr>
<td><code>\captiontext</code></td>
<td><code>\subcaptiontext</code> v1.5</td>
</tr>
</tbody>
</table>

This way our code snipped above could be simplified to:

```latex
\begin{overpic}[width=60pt]{cat}
  \put(40,34){\subcaptiontext*[1]{}}
\end{overpic}
```

Since `\setcaptionsubtype` should only be used within an extra group or environment, the same applies to these commands as well. (In this case `\subcaptiontext` is encapsulated by the \overpic environment, so we are ok here.

\textbf{⚠️} The `\subcaption` command is just a simple combination of `\setcaptionsubtype*` and `\caption`. Same for all other abbreviatory commands here.

## 11 The `\subfloat` command

`\subfloat`\footnote{New feature v1.3}

To allow a smoother transition from the subfig package [10] (which is unmaintained for over 16 years) this package also offers `\subfloat` with the same syntax:

```latex
\subfloat[⟨list_entry⟩]{⟨sub-caption⟩}{⟨body⟩}
```
12 Required packages

Starting with version 1.4 the subcaption package requires at least version 3.1 of the caption package and loads it automatically. (Older versions of the subcaption package have required exactly the version of the caption package which was released with it.)

If you need to use a specific version of the caption package you need to load it before the subcaption package, e.g.:

```latex
\usepackage[...]{caption}[=v3.5]
\usepackage[...]{subcaption}
```

Note that there are limitations if an older version of the caption package is used:

- The \phantomsbcommand need at least \caption v3.2.
- The \subcaptionlistentry command need at least \caption v3.3.
- The \subcaptiontext command need at least \caption v3.6.
- The \subfloat emulation needs at least \caption v3.4.

13 Other packages with similar offerings

The subcaption package is not the only one defining sub-figure counters and offering commands or environments to typeset sub-figures. The most established are:

- The memoir document class
- The subfigure package and its successor, the subfig package

Prior subcaption version 1.6 an error message was issued if the ‘subfigure’ or ‘subtable’ counter was already defined by any of them. Since subcaption version 1.6 the ‘subfigure’ and ‘subtable’ counters will only be defined by the subcaption package if they are not defined yet, but the usage of any of the commands or environments described in this document will result in an error message “The counter sub... was defined by ...” if the corresponding counter was not defined by the subcaption package. Equally the commands or environments offered by the above document classes or packages should not be used with counters defined by the subcaption package, this will result in either \LaTeX errors or unwanted behaviour.

So the rule of thumb is: If the counter ‘subfigure’ or ‘subtable’ is defined by document class or package xyz, then the corresponding commands and environments offered by the document class or package xyz should be used.

An example document:

```latex
\documentclass[memoi]
\newsubfloat{table} % defines the ‘subtable’ counter
\usepackage{subcaption} % defines the ‘subfigure’ counter
\begin{document}
\begin{figure}
\subcaptionbox{...}{...} % <- This is fine
\caption{...}
```
14 Beyond this package

For a more advanced usage of the sub-caption feature of the `caption` package, please take a look at the excellent `keyfloat` package[6] which provides the environments `keysfigs`, `keyssubtabs`, and `keyssubfloats` for typesetting sub-figures and sub-tables. Furthermore the `floatrow` package[3] provides the `subfloatrow` environment for typesetting sub-figures.

15 Thanks

I would like to thank Stephen Dalton who helped to make this package a better one.
16 The implementation

16.1 Identification

We need at least LaTeX2e version 1994/12/01.
\NeedsTeXFormat{LaTeX2e}[1994/12/01]

Bypass the release declarations in case the LaTeX kernel doesn’t know how to deal with them (as suggested by https://www.latex-project.org/publications/2018-FMi-TUB-tbl22mitt-version-rollback.pdf).
\providecommand\DeclareRelease[3]{}, \providecommand\DeclareCurrentRelease[2]{}

Declare all supported releases.
\DeclareCurrentRelease{v1}{2007/12/06}

Identify the current version of the package.
\ProvidesPackage{subcaption}[2023/02/19 v1.6-dev Sub-captions (AR)]

Since we base on the caption package we load it here.
\RequirePackage{caption}[2010/01/09] % we need at least v3.1m

16.2 Initial code

\subcaption@Info \subcaption@Info{⟨message⟩}
issues an info message (with code line indication).
\newcommand*\subcaption@Info{\PackageInfo{subcaption}}

\subcaption@Warning \subcaption@Warning{⟨message⟩}
issues an warning message (with code line indication).
\newcommand*\subcaption@Warning{\PackageWarning{subcaption}}

\subcaption@Error \subcaption@Error{⟨message⟩}
issues an error message (with code line indication).
\newcommand*\subcaption@Error[1]{\PackageError{subcaption}{#1}{\caption@@eh{subcaption}}}
\providecommand*\caption@@eh[1]{\IfYouDoNotUnderstandThisError, please take a closer look at the documentation of the ‘#1’ package, especially the section about errors.\MessageBreak\@ehc}

\subcaption@OutsideFloat \subcaption@OutsideFloat{⟨command⟩ outside float} issues a "⟨command⟩ outside float" error.
\newcommand*\subcaption@OutsideFloat[1]{\subcaption@Error{\string#1 outside float}}

16.3 Main code

\subcaptionsetup \subcaptionsetup*{⟨environment⟩}{⟨options⟩} is an abbreviation for \captionsetup*{⟨environment⟩}{⟨options⟩}, especially \subcaptionsetup⟨options⟩
sets options valid for all sub-captions.
\newcommand*\subcaptionsetup[1][]{\ifstar\subcaptionsetup*[⟨#1⟩]!}{\subcaptionsetup*[⟨#1⟩]!}{\subcaptionsetup*[⟨#1⟩]!}{\subcaptionsetup*[⟨#1⟩]!}
\setcaptionsubtype\setcaptionsubtype is available since caption package v3.2, so we need to define it first if only an older version is loaded (in fallback mode).
\providecommand\setcaptionsubtype{%
  \caption@iftype\relax
  \if\@ifstar\captionsetup{subtype*}{\captionsetup{subtype}}\relax
  \setcaptionsubtype
}\caption@iftype\relax
\setcaptionsubtype\setcaptionsubtype}

16.3.1 The subcaptiongroup environment

\begin{ fov. } subcaptiongroup (em.) subcaptiongroup is the simplest of all environments or commands this package has to offer. It’s simply an environment containing \setcaptionsubtype, that’s all. A starred variant of this environment is available, too, which uses \setcaptionsubtype*.
\begin{\newenvironment{subcaptiongroup}{\caption@iftype\relax}{\subcaption@OutsideFloat{subcaptiongroup}}}\end{\newenvironment} \begin{\newenvironment{subcaptiongroup*}{\caption@iftype\setcaptionsubtype*}{\subcaption@OutsideFloat{subcaptiongroup*}}}\end{\newenvironment}

16.3.2 The subcaptionblock environment (and aliases)

\begin{ fov. } subcaption@minipage This is a minipage with \setcaptionsubtype as first contents line.
\begin{\newcommand*{\subcaption@minipage}{\kernel@ifnextchar[\subcaption@iminipage\setcaptionsubtype\setcaptionsubtype\relax\if\@ifstar\captionsetup{subtype*}{\captionsetup{subtype}}\relax\setcaptionsubtype\setcaptionsubtype\relax\subcaption@iminipage\setcaptionsubtype\setcaptionsubtype\relax\subcaption@endminipage@hook\@empty\if#1B\minipage[b]{#2}{#3}\vspace{0pt}\else\if#1T\minipage[t]{#2}{#3}\vspace{0pt}\else\minipage[1]{#2}{#3}\fi\fi\@subfloatboxreset}}\end{\newcommand*} \begin{\def\subcaption@iminipage[#3]{\caption@withoptargs\subcaption@iiminipage{#1}, all other values (and additional optional arguments) will be passed unseen to the minipage environment.}
\begin{\newcommand*{\subcaption@iminipage}{\setcaptionsubtype\setcaptionsubtype\relax\if\@ifstar\captionsetup{subtype*}{\captionsetup{subtype}}\relax\setcaptionsubtype\setcaptionsubtype\relax\subcaption@endminipage@hook\@empty\if#1B\minipage[b]{#2}{#3}\vspace{0pt}\else\if#1T\minipage[t]{#2}{#3}\vspace{0pt}\else\minipage[1]{#2}{#3}\fi\fi\@subfloatboxreset}}\end{\newcommand*} \begin{\let\subcaption@oiniginpage@hook@empty}
\begin{\if\@state\vspace{0pt}\else\vspace{0pt}\fi%
subcaptionblock is a minipage with \setcaptionsubtype as first contents line.

\subcaption@newminipage\subcaption@newminipage{⟨name of new environment⟩}
defines a new environment containing \subcaption@newminipage and \subcaption@newminipage. Furthermore a test will be included which checks if the environment name matches sub\@captype.

\subcaption@newminipage[1]\subcaption@newminipage[#1]{\subcaption@newminipage[1]}\subcaption@newminipage[1]{\subcaption@newminipage[1]}

\caption@iftype
{\edef\caption@tempa{#1}\
  \edef\caption@tempb{sub\@captype}\
  \ifx\caption@tempa\caption@tempb \else\
    \subcaption@Warning{\caption@tempa is treated as \caption@tempb\MessageBreak}%
  \fi}%
{\subcaption@OutsideFloat[#1]}%
{\subcaption@OutsideFloat[#1]}%

\subcaptionbox
A \parbox with contents and sub-caption, separated by an invisible \hrule.
The code of this macro was moved to the caption package v3.2 (so it could offer it as \captionbox as well), so since then it’s sufficient to simply use \captionbox here. But if only caption package v3.1 is available (since the user decided to use this particular fallback version), we still must define the code on our own.
\subcaptionbox
\captionbox
\caption@withoptargs
16.3.4 The \texttt{\subref} command

We redefine \texttt{\label} to \texttt{\subcaption@label} at \texttt{\setcaptionsubtype}.

\begin{verbatim}
\g@addto@macro\caption@subtypehook{% 
  \ifx\label\subcaption@label 
    \let\subcaption@ORI@label\label 
    \let\label\subcaption@label 
  \fi}
\subcaption@label
\end{verbatim}

When a label will be placed for a sub-caption, we automatically place a second one for \texttt{\subref}, too. This second label will contain the sub-type counter only.

\begin{verbatim}
\newcommand*{\subcaption@label}{% 
  \caption@withoptargs{\subcaption@@label}
\newcommand*{\subcaption@@label}[2]{% 
  \@bsphack\begingroup
    \let\@bsphack\relax
    \let\@esphack\relax
  \@currentlabel by expanding the original \texttt{\label} code.
  \subcaption@ORI@label#1{#2}% 
  \subcaption@prepare@label
  \protected@edef\@currentlabel{\csname thesub@captype\endcsname} \subcaption@ORI@label{sub@#2}% 
  \endgroup\@esphack}
\subcaption@prepare@label
\end{verbatim}

Adaption to the \texttt{showkeys} package: Hide the ‘sub’ label from it.

\begin{verbatim}
\newcommand*{\subcaption@prepare@label}{% 
  \let\SK@\@gobbletwo
\end{verbatim}

Adaption to the \texttt{showlabels} package: Hide the ‘sub’ label from it by expanding to the original definition saved to $\texttt{\SL@orig#1}$ (with $#1 = \texttt{\SL@origlabel}$, ...).

\begin{verbatim}
  \def\SL@showlabels##1{\@nameuse{SL@orig##1}}
\end{verbatim}

\texttt{\subref} This one calls \texttt{\ref} with the second label. (see \texttt{\subcaption@label})

\begin{verbatim}
\DeclareRobustCommand*{\subref}{% 
  \@ifstar{\caption@withoptargs{\subcaption@ref*}}{% 
    \caption@withoptargs{\subcaption@ref}}
\newcommand*{\subcaption@ref}{% 
  \caption@withoptargs{\subcaption@ref}} \@ifundefined{hyperref}{% 
  {\subcaption@ref[#1](#2)}}{% 
    {\hyperref[#2][\subcaption@ref(#1)]}}
\end{verbatim}
Selecting a subref format simply means saving the code (in \subcaption@reffmt).

To offer a smooth transition from the subfig to the subcaption package we offer the options subrefformat=subsimple and subrefformat=subparens, too.

16.4 Execution of options

We use \caption@ExecuteOptions and \caption@ProcessOptions here to add the options to the ‘sub’ option list instead of executing them immediately.

\subcaption@DeclareType \subcaption@DeclareType{(type)}

defines the sub-counter via \DeclareCaptionSubType offered by the caption package.
We call \DeclareCaptionSubType for figure and table, and each floating environment declared with \DeclareFloatingEnvironment here.

```
197 \@ifundefined{ForEachCaptionType} \% caption3 v1.13
198 \{\@ifundefined{caption@ForEachType} \% caption3 v1.4a
199 \{\@ifundefined{c@figure}{}\subcaption@DeclareType{figure}\
200 \@ifundefined{c@table}{}\subcaption@DeclareType{table}\
201 \caption@For{typelist}\subcaption@DeclareType{#1}}\}
202 \{\caption@ForEachType\subcaption@DeclareType{#1}}\}
```

16.5 Bonus material

16.5.1 The \subcaption, \phantomsubcaption, \subcaptionlistentry, and \subcaptiontext commands

\subcaption@newabbreviation \subcaption@newabbreviation{(new command)}{(existing command)}{(extra code in case of error)}
defines a new command as abbreviation of \setcaptionsubtype* plus \(\text{(command)}\). (With a prefacing \setcaptionsubtype, \(\text{new command}\) is reduced to \(\text{(command)}\). Unfortunately we have to do this on our own since using \setcaptionsubtype multiple times is not suppressed until caption package v3.6.)

Note: Since \setcaptionsubtype is used, the new command is designed to be used inside an own group!

```
204 newcommand*\subcaption@newabbreviation[3]{% 205 \newcommand*#1{% 206 \caption@iftype 207 \{\setcaptionsubtype*#2\} 208 \{\subcaption@OutsideFloat#1#3\}% 209 \g@addto@macro\caption@subtypehook\{\let#1#2\}% needed for caption < 3.6
```

\caption@gooble \caption@gooble*{(arg)}{(...)}{(arg)}
is similar to \@gooble but gobbles a star and optional arguments as well.

```
211 \@ifundefined{caption@gooble}{% 212 \DeclareRobustCommand*\caption@gooble{% 213 \caption@withoptargs\gobbletwo}\% caption3 < v1.4 214 \caption@withoptargs\gobble\} 215 \{\}
```

\subcaption \subcaption is an abbreviation of \setcaptionsubtype* plus \caption.

```
215 \@ifclassloaded{memoir}\{\let\subcaption\undefined\} 216 \subcaption@newabbreviation\subcaption\caption\caption@gooble
```

\phantomsubcaption \phantomsubcaption is an abbreviation of \setcaptionsubtype* plus \phantomcaption.

Note: This commands needs at least caption package v3.2.

```
217 \subcaption@newabbreviation\phantomsubcaption\phantomsubcaption\relax
```

\subcaptionlistentry \subcaptionlistentry is an abbreviation of \setcaptionsubtype* plus \captionlistentry.

Note: This commands needs at least caption package v3.3.
\subcaptiontext is an abbreviation of \setcaptionsubtype* plus \captiontext.

Note: This command needs at least \caption package v3.6.

Inside the \subfloatrow environment offered by the floatrow package the usage of \caption etc. needs to be mapped to \subcaption etc.

\providecommand*{\caption@subfloatrow@hook}{} % caption < v3.7
\g@addto@macro{\caption@subfloatrow@hook}{% \let\caption\subcaption \let\phantomcaption\phantomsubcaption \let\captionlistentry\subcaptionlistentry \let\captiontext\subcaptiontext}

16.5.2 The \subfloat command

\subfloat \subfloat{\langle\textit{list	extunderscore entry}\rangle}{\langle\textit{sub	extunderscore caption}\rangle}{\langle\textit{body}\rangle}

If \langle\textit{sub	extunderscore caption}\rangle is given, we map this to \subcaptionbox but transfer the \label from \langle\textit{body}\rangle to \langle\textit{sub	extunderscore caption}\rangle. If not, we do the same as \subcaptionbox does, but use \phantomcaption instead of \caption. In both cases we do a \ignore-spaces at the end since the original implementation of \subfloat does this, too.

\begin{Verbatim}
\caption{AtBeginDocument{\%}
\begin{verbatim}
\newenvironment{subcaption@memoir@subfloat}{}{}%\iffalse\subfloat defined by memoir\fi
\if\subfloat\subcaption@memoir@subfloat\else
\renewcommand*{\subfloat}{\def{\subcaption@currenvir}{\subfloat}\if\@currenvir{\subcaption@currenvir}\else\subcaption@subfloat\fi}%
\else\endgroup\providecommand*{\subfloat}{\subcaption@subfloat}{}\fi
\end{verbatim}
\end{verbatim}
\end{Verbatim}
\subcaption@clrlabel
\subcaption@ignorespaces
\def\subcaption@subfloat@#1{%
  \setbox\@tempboxa\hbox{#1}%
  \caption@iiiibox
  \setcaptionsubtype
  \{% no optional arguments for \caption
  \}% no sub-captions
  \[\wd\@tempboxa]\[\captionbox@innerpos@default]%
  \{% unhbox\@tempboxa\%
  \}% no optional arguments for \caption
\}
\subcaption@clrlabel
\subcaption@clrlabel
\subcaption@getlabel\langle\text\rangle
gets the label command out of the given caption text and stores it to \caption@the-
label. It uses \caption@getlabel for this purpose which interface unfortunately
has changed over time. (Changing the interface was a bad idea in the first place but now
it is as it is).
\@ifundefined{caption@getlabel}{% caption3 < v1.7
  \newcommand\subcaption@getlabel[1]{% Cannes3 < v1.7
    \caption@Error\langle\noexpand\subfloat needs at least caption v3.4\rangle%
    \let\caption@thelabel\relax
  }\{@ifundefined{caption@getlabel}{% caption3 >= v1.7
    \newcommand\subcaption@getlabel[1]{% Cannes3 >= v1.7
      \caption@getlabel#1\label{}\@nil}
  }{%
    \newcommand\subcaption@getlabel{% Cannes3 >= v2.0
      \caption@getlabel}%
  }
}\}
\subcaption@clrlabel
\subcaption@clrlabel
\subcaption@clrlabel
\caption@the-
label to \relax.
\@ifundefined{caption@getlabel}{% caption3 < v2.3
  \newcommand\subcaption@clrlabel{% Cannes3 < v2.3
    \let\caption@thelabel\relax}
  }{%
    \newcommand\subcaption@clrlabel{% Cannes3 >= v2.3
    \caption@clrlabel}%
  }
}

References

[1] Frank Mittelbach and Michel Goossens:
*The \LaTeX Companion (2nd. Ed.*),

[2] Axel Sommerfeldt:
*Customizing captions of floating environments,*
2022/01/07
[3] Olga Lapko:  
*The floatrow package documentation*,  
2007/12/24

[4] Sebastian Rahtz & Heiko Oberdiek:  
*Hypertext marks in \LaTeX*,  
November 12, 2007

[5] Heiko Oberdiek:  
*The hypcap package – Adjusting anchors of captions*,  
2007/04/09

[6] Brian Duun:  
*The \LaTeX\ keyfloat Package*,  
2019/09/23

[7] Rolf Niepraschk:  
*The overpic package*,  
2020/02/22

[8] Steven B. Segletes:  
*The stackengine Package*,  
July 22, 2021

[9] Till Tantau:  
*The Ti\kZ and PGF Packages*,  
May 15, 2021

[10] Steven D. Cochran:  
*The subfig package*,  
2005/07/05