An Example of the Tufte-Handout Style[1]

The Tufte-LaTeX Developers

2025-04-07

[1] Inspired by Edward Tufte!

This document describes the Tufte-LaTeX `tufte-handout` document class style. It also provides examples and comments on the style's use. Only a brief overview is presented here; for a complete reference, see the sample book.

As mentioned in the sample book, some of the changes done in the Tufte-LaTeX classes are not in spirit with Tufte's designs. The changes were made to make the produced documents more accessible to my personal needs and preferences. However I have done my best to provide a way of disabling or easily overwriting these changes.

1 *Tufte-LaTeX Design*

The Tufte-LaTeX document classes define a style similar to the style Edward Tufte uses in his books and handouts. Tufte's style is known for its extensive use of sidenotes, tight integration of graphics with text, and well-set typography. This document aims to be at once a demonstration of the features of the Tufte-LaTeX document classes and a style guide to their use.

1.1 *Headings*

This style provides A- and B-heads (that is, `\section` and `\subsection`), demonstrated above.

Paragraph The `\paragraph` headings (as shown here) are introduced by italicized text and separated from the main paragraph by a bit of space.

By default `\subsubsection` and `\subparagraph` headings are not defined in the Tufte-LaTeX classes. They will emit an error if you try to use them and any smaller heading. For more information about these choices, see the *"Sample Book"* in subsection called *"Headings"* in the *"On the Use of the `tufte-book` Document Class"* chapter. There is also an informative section called *"Numbered Section Headings"* in the *"Customizing Tufte-LaTeX"* chapter.

In his later books,[2] Tufte starts each section with a "new thought". It has bit of vertical space, a non-indented paragraph, and sets the first few words in small caps. To accomplish this effect using this style, use the `\newthought` command:

[2] Tufte 2006.

```
\newthought{In his later books}, Tufte starts…
```

1.2 *Sidenotes*

One of the most prominent and distinctive features of this style is the extensive use of sidenotes. The wide margin on the right side provides ample room for sidenotes and small figures. Any `\footnote` will automatically be converted into a `\sidenote`.[3] If you'd like to place ancillary

[3] This is a sidenote that was entered using the `\footnote` command.

information in the margin without the sidenote mark (the superscript number), you can use the `\marginnote` command.

This is a margin note. Notice that there isn't a number preceding the note, and there is no number in the main text where this note was written.

In his books Tufte places margin on the right side of the page, regardless whether it's an even or odd page. If you prefer to alternate the placement of margins, so they fall on outer edge you can use the `symmetric` class option.

The specification of the `\sidenote` command is:

`\sidenote[`⟨*number*⟩`][`⟨*offset*⟩`]{`*Sidenote text.*`}`

Both the ⟨*number*⟩ and ⟨*offset*⟩ arguments are optional. If you provide a ⟨*number*⟩ argument, then that number will be used as the sidenote number. It will only change the number of the current sidenote, and will not affect the numbering sequence of subsequent sidenotes.

You can adjust the vertical position of the sidenote by providing a dimension in the ⟨*offset*⟩ argument. Some examples of valid dimensions are:

```
1.0in    2.54cm    254mm    6\baselineskip
```

If the dimension is positive, it will push the sidenote down the page; if the dimension is negative, it will pull the sidenote up the page.

The `\marginnote` command has a similar *offset* argument:

`\marginnote[`⟨*offset*⟩`]{`*Margin note text.*`}`

1.3 References

References are placed alongside their citations as sidenotes, as well. This can be accomplished using the normal `\cite` command or the `\autocite` command, which functions similarly.[4]

[4] If you use the `\cite` command within a sidenote, it will render as an in-line parenthetical citation, as demonstrated here (Tufte 2001).

You will need to specify a bibliography resource file in the preamble of your document using `\addbibresource{`*bibliography-file.bib*`}` command. The complete list of references may be printed automatically by using the `\printbibliography` command. See the end of this document for an example, and the BibLaTeX documentation for more information. Bibliography can be turned off with the help of `nobib` class option.

To enter multiple citations at one location,[5] you can provide a list of keys separated by commas:

[5] Tufte 1990, 2006.

`\cite{`*bibkey1,bibkey2,...*`}`

In the new version of Tufte-LaTeX, it's impossible to offset citation's position the same way sidenotes can be moved up or down the margin. You can find more information about the reasons for it in the *"Sample Book"*

1.4 Figures and Tables

Images and graphics play an integral role in Tufte's work. In addition to the standard `figure` and `tabular` environments, this style provides special figure and table environments for full-width floats.

Full page-width figures and tables may be placed in `figure*` or `table*` environments. To place figures or tables in the margin, use the `marginfigure` or `margintable` environments as follows (see figure 1):

```
\begin{marginfigure}
  \includegraphics{margin-figure}
  \caption{Margin figure caption}%
  \label{fig:margin-figure-label}
\end{marginfigure}
```

The `marginfigure` and `margintable` environments accept an optional parameter ⟨*offset*⟩ that adjusts the vertical position of the figure or table. See the "Sidenotes" section above for examples of how to use offsets. The specifications are:

```
\begin{marginfigure}[⟨offset⟩] % or margintable
  …
\end{marginfigure}
```

Figure 2 is an example of the `figure*` environment and figure 3 is an example of the normal `figure` environment.

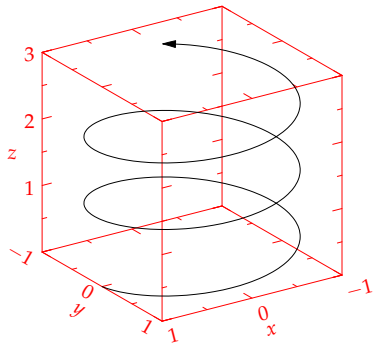

Figure 1: This is a margin figure. The helix is defined by $x = \cos(2\pi z)$, $y = \sin(2\pi z)$, and $z = [0, 2.7]$. The figure was drawn using Asymptote (`http://asymptote.sf.net/`).

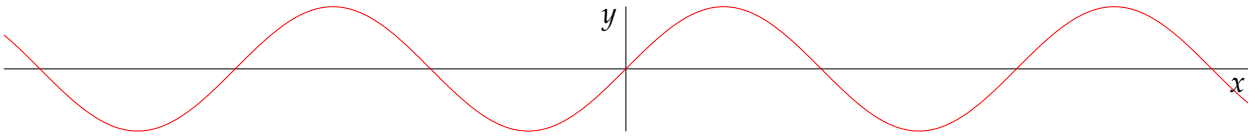

Figure 2: This graph shows $y = \sin x$ from about $x = [-10, 10]$. *Notice that this figure takes up the full page width.*

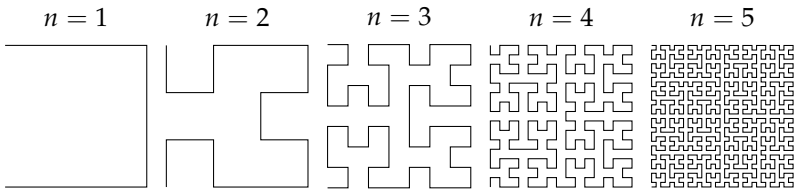

Figure 3: Hilbert curves of various degrees n. *Notice that this figure only takes up the main textblock width.*

As with sidenotes and marginnotes, a caption may require vertical adjustment. The `\caption` command can take a second optional argument which enables you to do this by providing a dimension ⟨*offset*⟩. You may specify the caption in any one of the following forms:

```
\caption{long caption}
\caption[short caption]{long caption}
\caption[][⟨offset⟩]{long caption}
\caption[short caption][⟨offset⟩]{long caption}
```

A positive ⟨*offset*⟩ will push the caption down the page. The short caption, if provided, is what appears in the list of figures/tables, otherwise the "long" caption appears there. Note that although the arguments ⟨*short caption*⟩ and ⟨*offset*⟩ are both optional, they must be provided in order. Thus, to specify an ⟨*offset*⟩ without specifying a ⟨*short caption*⟩, you must include the first set of empty brackets `[]`, which tell `\caption` to use the default "long" caption. As an example, the caption to figure 3 above was given in the form:

```
\caption[Hilbert curves…][1em]{Hilbert curves…}
```

CAPTION OFFSET IS UNAVAILABLE for `marginfigure` and `margintable` environments. In these cases you need to offset the whole figure or table. Captions in `marginfigure` and `margintable` still support short captions.

TUFTE STYLE TABLES ARE SIMPLE and should be styled with the `booktabs` package. Table 1 shows table created with the `booktabs` package. Notice the lack of vertical rules—they serve only to clutter the table's data. Hence Tufte style tables use only horizontal rules. In cases where a table has many rows, `colortbl` can be used to make rows stand out visually from each other. Colors can be used to group related rows, highlight important data, or make one row stand out from the others.

Margin	Length
Paper width	8 1/2 inches
Paper height	11 inches
Textblock width	6 1/2 inches
Textblock/sidenote gutter	3/8 inches
Sidenote width	2 inches

Table 1: Here are the dimensions of the various margins used in the Tufte-handout class.

1.5 *Too Many Floats*

OCCASIONALLY LaTeX will generate an error message:

```
Error: Too many unprocessed floats
```

LaTeX tries to place floats in the best position on the page. Until it's finished composing the page, however, it won't know where those positions are. If you have a lot of floats on a page (including sidenotes, margin notes, figures, tables, etc.), LaTeX may run out of "slots" to keep track of them and will generate the aforementioned error.

LaTeX initially allocates 18 slots for storing floats. To work around this limitation, the Tufte-LaTeX document classes provide a `\morefloats` command that will reserve more slots.

The first time `\morefloats` is called, it allocates an additional 34 slots. The second time `\morefloats` is called, it allocates another 26 slots.

The `\morefloats` command may only be used two times. Calling it a third time will generate an error message:

```
You may only call \morefloats twice
See the Tufte-LaTeX documentation for alternatives
```

This is because allocating more floats may lead LaTeX to run out of memory.

See the *"Too Many Floats"* section *"Sample Book"* for more information on how to handle this error.

1.6 *Captions*

You may notice that the captions are sometimes misaligned. Due to the way LaTeX's floats works, it's hard to know for sure where it decided

to put the float. Therefore, the Tufte-LaTeX document classes provide commands to override the caption position.

Vertical alignment In cases where the caption is too high or too low on the page, you can adjust its vertical position. To override the caption's vertical alignment, use the provided `\setfloatalignment` command inside the float environment. For example:

```
\begin{figure}
  \includegraphics{vertical-figure}
  \caption{vertical-figure-caption}%
  \label{fig:vertical-figure-label}
  \setfloatalignment{b} % forces caption to be bottom-aligned
\end{figure}
```

The syntax of the `\setfloatalignment` command is:

```
\setfloatalignment{⟨pos⟩}
```

where ⟨*pos*⟩ can be either `b` for bottom-aligned captions, or `t` for top-aligned captions.

Horizontal alignment To override the horizontal alignment, use either the `\forceversofloat` or the `\forcerectofloat` command inside of the float environment. Note that these commands only work when the `symmetric` option is enabled. For example:

```
\begin{figure}
  \includegraphics{horizontal-figure}
  \caption{horizontal-figure-caption}%
  \label{fig:horizontal-figure-label}
  \forceversofloat % forces caption to the left of the float
\end{figure}
```

The `\forceversofloat` command causes the algorithm to assume the float has been placed on a verso page—that is, a page on the left side of a two-page spread. Conversely, the `\forcerectofloat` command causes the algorithm to assume the float has been placed on a recto page—that is, a page on the right side of a two-page spread.

1.7 *Full-width text blocks*

In addition to the new float types, there is a `fullwidth` environment. This environment stretches across the main text block and the sidenotes area.

```
\begin{fullwidth}
  Lorem ipsum dolor sit amet…
\end{fullwidth}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1.8 Typography

Typefaces When using XeLaTeX or LuaLaTeX, the Tufte-LaTeX classes will load the `fontspec` package. This package allows you to set the typeface to any installed font, any local font files, or to any font files you have installed in your `TEXMF` tree.

By default the Tufte-LaTeX classes will use the ET-Bembo font from the `ETbb` package, as the main typeface. If it's unavailable, the TeX Gyre Pagella from the `tex-gyre-pagella` package will be used as fallback serif font. For math fonts it tries to use the Palatino font from the `mathpazo` package. For sans serif text the Gillius No. 2 font from the `gillius` package will be used. If this one is unavailable, the TeX Gyre Heros font from the `tex-gyre-heros` package will be used. In case of monospaced text the `Fira Mono` font from the `FiraMono` package will be used. If it's not present, the TeX Gyre Cursor font from the `tex-gyre-cursor` package will be used. However the provided `custom-tufte-common.tex`common file hook overrides the default monospaced font with `RecursiveMono` font. This file shows how you can override the default fonts, and how the file hooks can be used.

The TeX Gyre faces are usually included with TeX Live distributions, hence why they are used as fallback fonts. If any of the selected fonts don't suit you, you can easily change them using the `fontspec` package.

When using the pdfLaTeX engine, the Tufte-LaTeX classes will try to use the same default fonts, but will fall back to the default Computer Modern fonts if they are unavailable. The `fontspec` package is not available under pdfLaTeX, so it uses the `fontenc` package to set the font encoding. This package doesn't make it easy to use non-standard fonts, so it's recommended to use XeLaTeX or LuaLaTeX for the best results. Alternatively install and use font packages that are compatible with pdfLaTeX.

In cases where `nofonts` option is used, the Tufte-LaTeX classes will not load any fonts. In pdfLaTeX will not load the `fontenc` package. In other engines `nols` option must also be used to disable loading of `fontspec` package.

Letterspacing This document class includes two new commands and some improvements on existing commands for letterspacing.

When setting strings of ALL CAPS or small caps commands, the letterspacing—that is, the spacing between the letters—should be increased slightly.[6] The `\allcaps` command was modified with proper letterspacing for strings of FULL CAPITAL LETTERS, and the `\smallcaps` command was modified with spacing for small capital letters. These commands will also automatically convert the case of the text to upper- or lowercase, respectively. You can see that in the source code of this document.

The `\textsc` command has also been redefined to include proper letterspacing. However, the case of the `\textsc` argument is left as is. This allows one to use both uppercase and lowercase letters: The Initial Letters Of The Words In This Sentence Are Capitalized.

[6] Bringhurst 2005.

2 Document Class Options

The `tufte-book` class is based on the LaTeX `book` document class. Conversely the `tufte-handout` class is based on the `article` document class. Therefore, you can pass any of the typical book or article options to them. Tufte-LaTeX offers a few additional options that are specific to the `tufte-book` and `tufte-handout` document classes. Besides the `nomodern-titles` options, which is only applicable to the `tufte-handout` class, all other options are available for both classes.

2.1 Paper Size and Layout Options

The `a4paper` option will set the paper size to A4 instead of the default US letter size.

The `b5paper` option will set the paper size to B5 instead of the default US letter size.

The `a5paper`, `executivepaper`, and `legalpaper` options are unavailable in the Tufte-LaTeX classes.

The `twoside` option will modify the running heads so that the page number is printed on the outside edge.In other words it will be placed on the right side of the odd pages, and on the left side of the even pages. When it comes to books, the head on the left side will also contain book title, and right side will contain chapter title. While in case of the handouts the left side head will use the author name, and right side will use the handout title. By default the Tufte-LaTeX classes use the `twoside` option, as Tufte's *BE* book has done.[7] If you wish to disable it you can use the `oneside` option on a case by case basis.

[7] Tufte 2006.

The `symmetric` option typesets the sidenotes on the outside edge of the page, same way the `twoside` option does for the heads. This is the way books are traditionally printed, but Tufte's book design places the sidenotes on the right side of every page. This option implicitly sets the `twoside` option.

The `landscape`, `onecolumn`, and `twocolumn` options are not available in the Tufte-LaTeX classes.

2.2 Font and Text Options

The `sftitle` option will set the title page or block in a sans serif typeface. The `nosftitle` option will set the title page or block in a serif typeface. In case of `tufte-handout` these options also affect the abstract while in `tufte-book` they affect the epigraphs. By default the `tufte-book` class uses `sftitle` and the `tufte-handout` class uses `nosftitle`.

The `sfmarginals` option makes all marginal material use sans serif typeface instead of the default serif typeface.

The `justified` option fully justifies the main text (flush left and right). By default the text is ragged right, just as the body text of Tufte's books is ragged right. This prevents needless hyphenation and makes it easier to read the text in the slightly narrower column.

The `10pt`, `11pt`, and `12pt` options are unavailable in the Tufte-LaTeX classes.

The nofonts option prevents the Tufte-LaTeX classes from automatically loading the Tufte typefaces. You should use this option if you wish to load your own fonts in pdfLaTeX. If you're using XeLaTeX or LuaLaTeX, you can use fontspec to set fonts, so this option is not necessary, but is available if you wish to use it. If you aren't using the nols option, the fontspec package will still be loaded as it is required for letterspacing.

The nols option inhibits loading the code that modifies the letterspacing. The Tufte-LaTeX classes try to load the appropriate letterspacing package to adjust spacing of letters in all-caps environments. It uses letterspace or the soul under pdfLaTeX. In case of XeLaTeX and LuaLaTeX it uses fontspec.

The bidi option loads the bidi package which is used with XeLaTeX to typeset bi-directional text. Since the bidi package needs to be loaded before the sidenotes and cite commands are defined, it can't be loaded in the document preamble. Hence this option exists to load it in the class file.

2.3 Title Page Options

The notitlepage option causes \maketitle to generate a title block instead of a title page. While the analogous titlepage option causes \maketitle to generate a full title page. By default the tufte-book class uses titlepage option and the tufte-handout class uses the notitlepage.

2.4 Toggle Options

The nobib option inhibits loading of the natbib and bibtex packages and modifying the \cite command.

The notoc option suppresses Tufte-LaTeX's custom table of contents (TOC) design. The current TOC design only shows unnumbered chapter titles in books; it doesn't show sections or subsections. The notoc option will revert to LaTeX's TOC design.

The nohyper option prevents the hyperref package from being loaded. The default is to load the hyperref package and use the \title and \author contents as metadata for the generated PDF.

The nonotes option inhibits loading packages used for shadednote and framednote environments. By default it loads the amsthm, cleveref, and thmtools packages. Using those it defines environments for notes that are either shaded or framed.

The nomoderntitles is a new option added in the latest version of Tufte-LaTeX. It only works in the tufte-handout class. It disables coloring and styling of the section, subsection, and paragraph titles. The default is to color the titles and add a colored box to the left with section numbers inside it.

2.5 Marginal Options

In the Tufte-LaTeX classes there are four types of marginal materials, which are: `sidenote`, `marginnote`, `caption`, and `citation`. Each of those can have their justification set to one of the following options:

`justified` Sets the text to be justified (sets it flush left and right).

`raggedleft` Sets the text to be ragged left.

`raggedright` Sets the text to be ragged right.

`raggedouter` Sets the text to be ragged left on the left (verso) page, and ragged right on the right (recto) page. This is useful in conjunction with the `symmetric` document class option.

`auto` Justified the text if `justified` class option is on, otherwise used default ragged right text. This is the default justification option for marginal material.

Additionally, the `marginals` option can be used to set the justification settings for all marginal materials. See the Customizing Marginal Material section for more information on marginal material.

2.6 Debugging Options

The `debug` option causes the Tufte-LaTeX classes to output debug information to the log file which is useful in troubleshooting bugs. It prints list of options and their values under the `Tufte-LaTeX settings` section. Additionally the `tufte-handout` will print out a dedicated `Tufte-LaTeX Handout settings` section. It will also cause the graphics to be replaced by outlines. When combined with `\geometry`⟨*showframe*⟩ command it will show margins for debugging page layout issues.

3 Customizing Tufte-LaTeX

The Tufte-LaTeX document classes are designed to closely emulate Tufte's book design by default. However, each document is different and you may encounter situations where the default settings are insufficient. This chapter explores many of the ways you can adjust the Tufte-LaTeX document classes to better fit your needs.

3.1 File Hooks

When creating many documents using the Tufte-LaTeX classes, it's easier to store common customizations in one file. Otherwise they would need to be copied into the preamble of each document. The Tufte-LaTeX classes provide three file hooks: `custom-tufte-common.texcommon`, `custom-tufte-book.texbook`, and `custom-tufte-handout.texhandout`.

`custom-tufte-common.tex` If this file exists, it will be loaded by all of the Tufte-LaTeX document classes, just prior to any class-specific code. If your customizations or code should be included in both the book and handout classes, use this file hook.

`custom-tufte-book.tex` If this file exists, it will be loaded after all of the common and book-specific code has been read. If your customizations apply only to the book class, use this file hook.

`custom-tufte-handout.tex` If this file exists, it will be loaded after all of the common and handout-specific code has been read. If your customizations apply only to the handout class, use this file hook.

This project comes with a `custom-tufte-common.texcommon` file hook that demonstrates how to use the file hooks. It shows how to change the monospaced font to `RecursiveMono`. You can use it as a starting point for your own customizations.

3.2 Numbered Section Headings

While Tufte dispenses with numbered headings in his books, if you require them, they can be enabled by changing the value of the `secnumdepth` counter. From the table below, select the heading level at which numbering should stop and set the `secnumdepth` counter to that value. For example, if you want parts and chapters numbered, but don't want numbering for sections or subsections, use the command:

```
\setcounter{secnumdepth} {0}
```

The default value of `secnumdepth` for the `tufte-book` class is -1. Note that this makes it impossible to use the `cleveref` package's `\cref` command with sections and subsections. This version of `tufte-handout` class sets the counter to 2 so sections and subsections are numbered. This change was made to make the sections stand out more as I found it hard to distinguish them from the body text. If you wish to revert to no numbering, set the counter to -1. You can also pass the `nomoderntitles` option to the `tufte-handout` class to disable the coloring and styling of the section and paragraph titles.

Heading level	Value
Part (in `tufte-book`)	−1
Part (in `tufte-handout`)	0
Chapter (only in `tufte-book`)	0
Section	1
Subsection	2
Subsubsection	3
Paragraph	4
Subparagraph	5

Table 2: Heading levels used with the `secnumdepth` counter.

3.3 Changing the Paper Size

The Tufte-LaTeX classes currently only provide three paper sizes: A4, B5, and US letter. To specify a different paper size (and/or margins), use the `\geometry` command in the preamble of your document (or one of the file hooks). The full documentation of the `\geometry` command may be found in the `geometry` package documentation.[8]

[8] Umeki 2008.

3.4 Customizing Marginal Material

Marginal material includes sidenotes, citations, margin notes, and captions. Normally, the justification of the marginal material follows the justification of the body text. If you specify the `justified` document class option, all of the margin material will be fully justified as well. If you don't specify the `justified` option, then the marginal material will be set ragged right.

You can set the justification of the marginal material separately from the body text using the following document class options: `sidenote`, `marginnote`, `caption`, `citation`, and `marginals`. The `marginals` option simultaneously sets the justification on all four marginal material types. For example:

```
\documentclass[symmetric,justified,marginals=raggedouter]{tufte-book}
```

will set the body text of the document to be fully justified. All of the margin material (sidenotes, margin notes, captions, and citations) to be flush against the body text with ragged outer edges.

The font and style of the marginal material may also be modified using the following commands:

```
\setsidenotefont{⟨font commands⟩}
\setcaptionfont{⟨font commands⟩}
\setmarginnotefont{⟨font commands⟩}
\setcitationfont{⟨font commands⟩}
```

The ⟨*font commands*⟩ can contain font size changes (e.g., `\footnotesize`, `\Huge`, etc.), font style changes (e.g., `\sffamily`, `\ttfamily`, `\itshape`, etc.), color changes (e.g., `\color{tufte-blue}`), and many other adjustments.

If, for example, you wanted the captions to be set in italic sans serif, you could use:

```
\setcaptionfont{\itshape\sffamily}
```

4 New Features in Tufte-LaTeX Classes

4.1 Custom Colors

The new Tufte-LaTeX document classes define a number of custom colors. They use these colors for things like links, citations, links, etc. In case of `tufte-handout` class it also uses them for the section titles. The common class uses the `xcolor` package to define the colors. You can choose to use these colors in your own documents as you see fit, redefine them, override them, or not use them at all. Here are the colors available in the Tufte-LaTeX document classes:

tufte-black	tufte-grey
tufte-red	tufte-pastel-red
tufte-orange	tufte-pastel-orange
tufte-yellow	tufte-pastel-yellow
tufte-green	tufte-pastel-green
tufte-blue	tufte-pastel-blue
tufte-purple	tufte-pastel-purple

The colors are defined with following values:

```
\definecolor{tufte-black}{HTML}{282828}
\definecolor{tufte-grey}{HTML}{F6F6F6}
\definecolor{tufte-white}{HTML}{FFFFFF}
\definecolor{tufte-red}{HTML}{E74C3C}
\definecolor{tufte-pastel-red}{HTML}{FADBD8}
\definecolor{tufte-orange}{HTML}{E67E22}
\definecolor{tufte-pastel-orange}{HTML}{FAE5D3}
\definecolor{tufte-yellow}{HTML}{F1C40F}
\definecolor{tufte-pastel-yellow}{HTML}{FCF3CF}
\definecolor{tufte-green}{HTML}{27AE60}
\definecolor{tufte-pastel-green}{HTML}{D4EFDF}
\definecolor{tufte-blue}{HTML}{3498DB}
\definecolor{tufte-pastel-blue}{HTML}{D6EAF8}
\definecolor{tufte-purple}{HTML}{9B59B6}
\definecolor{tufte-pastel-purple}{HTML}{EBDEF0}
```

4.2 Note Environments

Another feature common to the new Tufte-LaTeX classes are two environments for notes. These notes can be used to highlight important information, provide references, or to simply make a note. It's useful for making important informations stand out, without risk of being lost in the margins.

Both note environments provide a title, a label, and a continuation option. The title and label are optional, the label is used for referencing the note or for continuing a note later in the text. The `shadednote` environment creates a note with a shaded background. While the `framednote` environment places a frame to the left of the note. Both environments use the same counter as they are similar enough so it doesn't make sense to separate them.

Note 1. *This is an example of the* `shadednote` *environment. It provides a shaded background for the note text. The note text can be long or short, although they should be short and to the point. Notes should be used to crucial information, not be a substitute for a paragraph*

Note 2 (*Note Title*)

> *This is an example of the* `framednote` *environment. Frames the note text with to the left of note and is more muted than the* `shadednote`*. In both cases the note text is italicized.*

If you label an note, you can reference it using the `\cref` command. For example, Note 2 showcases the `framednote` environment. You can also use the ⟨*continues*⟩ option to continue an note:

Note 2 (*continuing from p. 13*)

> *This is a continuation of the previous note. It will be displayed in a new frame, but will have the same label, title, and number. Useful if you want to refer to or expand on previous note without having to repeat the information.*

The way to use the note environments is as follows:

```
\begin{shadednote}[
  title={Optional title},
  label={Optional label},
  continues={Optional label}
  ]
  Note text here
\end{shadednote}
```

4.3 Modified Section Headings

The `tufte-handout` class provides a new section heading style. Every section, subsection and paragraph will have their names colored, in addition the the usual styling. Additionally section and subsection titles will have a colored box placed to the left of the title. Section number will be placed inside the box. This is done to make the sections stand out more, as I found it hard to distinguish them from the body text. If you wish to revert to the default styling, you can pass the `nomoderntitles` option to the `tufte-handout` class.

The following figure shows the difference between the default and the new section heading style.

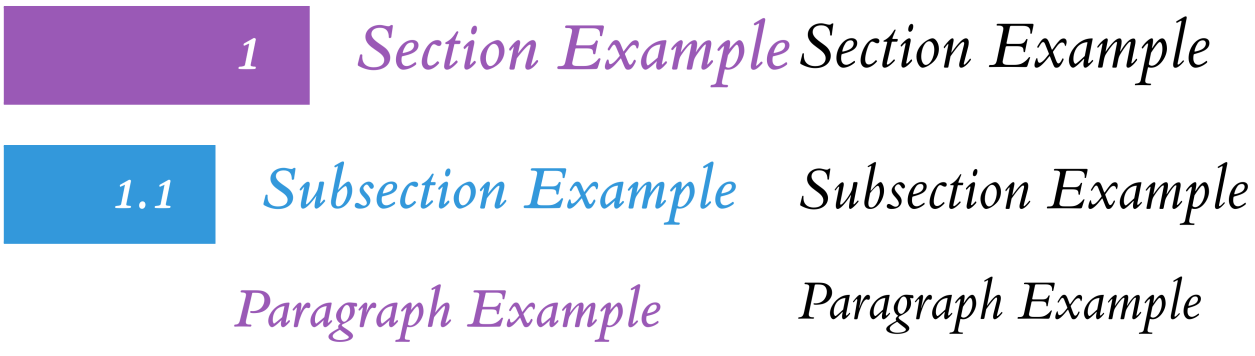

Figure 4: Comparison between modern and old style section headings. *Notice that the spacing between the sections is a little bit different. This was motivated by wanting to equalize the spacing between colored boxes.*

5 Installing and Compiling Tufte-LaTeX

5.1 Local Installation

To install the Tufte-LaTeX classes locally, download the source files from the GitHub repository[9]. Then you can place the following class files in the same directory as your TeX document:

- `tufte-common.def`
- `tufte-handout.cls`
- `tufte-book.cls`
- `custom-tufte-common.tex` *(optional)*

You can then either compile the document using `latexmk` or use the provided `makefile` to compile the document. See the Compiling Your Document section for more information on compiling the document.

[9] Repository can be found at `https://github.com/MormonJesus69420/Modernized-Tufte-LaTeX`

5.2 Global Installation

To install the Tufte-LaTeX classes globally, you can place the class files in your TeX distribution's `tex/latex/tufte-latex` directory. When doing it by hand use the `kpsewhich -var-value TEXMFHOME` to find the correct directory, and copy the files there. After copying the files, you will need to update the TeX distribution's file database. To do that run the `texhash` command in the terminal.

You can also use the `install` target from provided `makefile`. This will do all these steps for you, installing additional documentation files and more.

5.3 Installing from CTAN

TODO! Provide installation guide from CTAN when package is published there

6 Compiling Your Document

The Tufte-LaTeX classes can be compiled using pdfLaTeX, XeLaTeX, or LuaLaTeX. I personally recommend using LuaLaTeX, as it provides the best support for Tufte-LaTeX functionality without and quirks that XeLaTeX has.

The Tufte-LaTeX classes are designed to work with the `biblatex` package for bibliographies. As such you will need to have `biber` installed on your system to compile the bibliography files.

6.1 Compiling with `makefile`

As mentioned before you can use the provided `makefile` to compile the sample documents with either `make compile-lua`, `make compile-xe`, or `make compile-pdf`. It is up to you which engine you prefer to use, but from my experience LuaLaTeX provides the best results and is the

most reliable of them all. In case of single files use the `make compile-only FILE="<file>.tex"` target. It is set up to use LuaLaTeX by default, but you can change it to use XeLaTeX or pdfLaTeX by changing the `-lualatex` option to `-xelatex` or `-pdflatex`.

In general I would recommend changing the `-outdir=samples` in the `LMKFLAGS` variable to a directory of your choice, for example: `-outdir=out`.

If you wish to clean up the auxiliary files, use the `make clean` target. And the remove both auxiliary files and compiled PDFs use the `make clean-all` target.

The "Sample Book" and the project's readme provide a simplified `makefile`, with only the necessary targets for most use-cases.

6.2 *Compiling manually*

You may also compile the documents with `latexmk` directly using the following command: Just substitute "`«engine»`" with your preferred engine, and "`«source.tex»`" with the name of your source file.

```
latexmk -outdir=out -auxdir=aux -bibtex -<<engine>> -jobname=%A-lualatex <<source.tex>>
```

You may also do things fully manually. To compile `tufte-handout`, you should do something like this:

```
lualatex  <<source.tex>> # Replace <<source.tex>> with the name of your source file
biber     <<source>>     # Note that the file extension is omitted
lualatex  <<source.tex>>
lualatex  <<source.tex>>
```

To compile `tufte-handout`, you should do something like this:

```
lualatex  <<source.tex>> # Replace <<source.tex>> with the name of your source file
biber     <<source>>     # Note that the file extension is omitted
makeindex <<source.idx>> # If you are using an index
lualatex  <<source.tex>>
lualatex  <<source.tex>>
lualatex  <<source.tex>>
```

7 Compatibility Issues

When switching an existing document from one document class to a Tufte-LaTeX document class, a few changes to the document may have to be made.

7.1 Converting from `article` to `tufte-handout`

The following `article` class options are unsupported: `10pt`, `11pt`, `12pt`, `a5paper`, `b5paper`, `executivepaper`, `legalpaper`, `landscape`, `onecolumn`, and `twocolumn`.

The following headings are not supported: `\subsubsection` and `\subparagraph`.

7.2 Converting from `book` to `tufte-book`

The following `book` class options are unsupported: `10pt`, `11pt`, `12pt`, `a5paper`, `b5paper`, `executivepaper`, `legalpaper`, `landscape`, `onecolumn`, and `twocolumn`.

The following headings are not supported: `\subsubsection` and `\subparagraph`.

8 Troubleshooting and Support

8.1 Tufte-LaTeX Website

The original website for the Tufte-LaTeX packages is located at https://tufte-latex.github.io/tufte-latex/. On that website, you'll find links to the GIT repository, mailing lists, bug tracker, and documentation.

However as the project seems to be abandoned as of time of writing, the website may not be available in the future. Additionally some of the links there seem to have already been victim of link rot. You can find more help and information on the current development of the Tufte-LaTeX classes at my GitHub repository. https://github.com/MormonJesus69420/Modernized-Tufte-LaTeX

I have also set up a custom website for the project which is hosted under https://salwerowi.cz/Modernized-Tufte-LaTeX.

If for some reason this address does not work it should also be available under https://mormonjesus69420.github.io/Modernized-Tufte-LaTeX.

8.2 Tufte-LaTeX Mailing Lists

There is only one surviving mailing list for the Tufte-LaTeX project:

Discussion list The `tufte-latex` discussion list is for asking questions, getting assistance with problems, and help with troubleshooting. Release announcements were also posted to this list.

You can subscribe to the `tufte-latex` discussion list at http://groups.google.com/group/tufte-latex

Commits list The `tufte-latex-commits` list used to exist as well as a read-only mailing list. Messages were sent to the list any time the Tufte-LaTeX code had been updated.

This list was available at http://groups.google.com/group/tufte-latex-commits

A more modern way to keep up with the development of the Tufte-LaTeX classes is to follow the GitHub repository. You can use the *Discussions* feature to ask questions, suggest improvements, and interact with other users. In case you notice and bugs with the classes, or notice any issues with the documentation, you can open an issue on the GitHub repository. In case you want to contribute to the project, you can open a pull request on the GitHub repository.

https://github.com/MormonJesus69420/Modernized-Tufte-LaTeX/discussions

https://github.com/MormonJesus69420/Modernized-Tufte-LaTeX/issues

https://github.com/MormonJesus69420/Modernized-Tufte-LaTeX/pulls

I DON'T PLAN ON USING MAILING LISTS in my project, as I never got used to them, and feel that GitHub provides a more user friendly way to interact with the project.

8.3 Getting Help

If you've encountered a problem with one of the Tufte-LaTeX document classes, have a question, or would like to report a bug, please create an issue or start a discussion on the GitHub repository.

To help with troubleshooting the problem more quickly, please try to compile your document using the `debug` class option. When asking for help include the generated `.log` file in the issue, along with a brief description of the problem, and if possible a minimal working example that reproduces the issue. You can also upload your whole document if you're not sure what's causing the issue, and you are comfortable with sharing the document.

8.4 Package Dependencies

The following is a list of packages that the Tufte-LaTeX document classes rely upon. Packages marked with an asterisk are either optional or only loaded for some class options.

- ams{math,symb,thm,xtra} * *for* `Note` *environments*
- biblatex * *only if* `nobib` *is off, requires* `biber` *backend*
- bidi * *only if using* `bidi` *option*
- changepage
- chngpage * *only if* `changepage` *is not available*
- cleveref * *for* `Note` *environments*
- ETbb * *if available, and* `nofonts` *is off*
- fancyhdr
- FiraMono * *if available, and* `nofonts` *is off*
- fontenc * *only with pdfLaTeX, and* `nofonts` *is off*
- fontspec * *only with XeLaTeX or LuaLaTeX, and* `nofonts` *is off*
- geometry
- gillius2 * *if available, and* `nofonts` *is off*
- hardwrap
- hyperref * *only if* `nohyper` *is off*
- iftex * *if not it assumes pdfLaTeX*
- letterspace * *only if* `nols` *is off*
- mathpazo * *if available, and* `nofonts` *is off*
- multicol
- optparams
- paralist
- placeins
- ragged2e
- sectsty
- setspace
- soul * *only with pdfLaTeX*
- textcase
- textcomp * *only with pdfLaTeX, and* `nofonts` *is off*
- thmtools * *for* `Note` *environments*
- titlesec
- titletoc
- transparent
- xcolor
- xifthen
- xkeyval

8.5 More Documentation

For more documentation on the Tufte-LaTeX document classes (including commands not mentioned in this handout), please see the sample book.

8.6 Support

The website for the Tufte-LaTeX packages is located at `https://github.com/Tufte-LaTeX/tufte-latex`. On our website, you'll find links to our SVN repository, mailing lists, bug tracker, and documentation.

9 *References*

Bringhurst, Robert (2005). *The Elements of Typography*. 3.1. Vancouver, British Columbia: Hartley & Marks. ISBN: 0-88179-205-5 (cit. on p. 6).

Tufte, Edward R. (1990). *Envisioning Information*. Cheshire, Connecticut: Graphics Press. ISBN: 0-9613921-1-8 (cit. on p. 2).

– (2001). *The Visual Display of Quantitative Information*. Cheshire, Connecticut: Graphics Press. ISBN: 0-9613921-4-2 (cit. on p. 2).

– (2006). *Beautiful Evidence*. First. Cheshire, Connecticut: Graphics Press, LLC. ISBN: 0-9613921-7-7 (cit. on pp. 1, 2, 7).

Umeki, Hideo (Dec. 2008). *The `geometry` package*. `http://ctan.org/pkg/geometry` (cit. on p. 11).