Modern Beamer Presentations with the **metropolis** package

Matthias Vogelgesang
matthias.vogelgesang@gmail.com

v1.1 — 2016/02/06

Contents

1 Introduction

Beamer is an awesome way to make presentations with LaTeX, but its theme selection is surprisingly sparse. The stock themes share an aesthetic that is can be a little cluttered, and the few distinctive custom themes available are often specialized for a particular corporate or institutional brand.

The goal of **metropolis** is to provide a simple, modern Beamer theme suitable for anyone to use. It tries to minimize noise and maximize space for content; the only visual flourish it offers is an (optional) progress bar added to each slide or to the section slides.

By default, **metropolis** uses Fira Sans, a gorgeous typeface commissioned by Mozilla and designed by Carrois. For best results, you will need the Fira typeface installed and use XeLaTeX to typeset your slides. However, **metropolis** can also be used with other typefaces and LaTeX build systems.

metropolis's codebase is maintained on GitHub. If you have issues, find mistakes in the manual or want to help make the theme even better, please get in touch there. The full list of contributors already contains over a dozen names!

2 Getting Started

2.1 Installing from CTAN

For the regular user it is recommended to install **metropolis** from CTAN. In case you keep your TeX distribution up-to-date, chances are good that **metropolis** is already installed. If it is not, you need to update your packages. For TeX Live (or MacTeX on OS X) the following command updates all packages.

```
sudo tlmgr update --all
```

For any other distribution please refer to its documentation on how to update your packages.

To get the most out of the theme you should also install the `Fira` fonts. Yet this is not mandatory. **metropolis** also works with the standard fonts.

2.2 Installing from GitHub

Installing **metropolis**, like any Beamer theme, involves four easy steps:

Download the source with a `git clone` of the **metropolis** repository or as a zip archive of the latest development version.

Compile the style files by running `make sty` inside the downloaded directory. (Or run LaTeX directly on `source/metropolistheme.ins`.)

Move the resulting `*.sty` files to the folder containing your presentation. To use **metropolis** with many presentations, run `make install` or move the `*.sty` files to a folder in your TeX path instead.

Use the theme for your presentation by declaring `\usetheme{metropolis}` in the preamble of your Beamer document.

metropolis uses the Make build system to offer the following installation options for advanced users:

`make sty` builds the theme style files.

`make doc` builds this documentation manual.

`make demo` builds a demo presentation to test the features of **metropolis**.

`make all` builds the theme, manual, and demo presentation.

`make clean` removes the files generated by `make all`.

`make install` installs the theme into your local texmf folder.

`make uninstall` removes the theme from your local texmf folder.

`make ctan` creates a package for CTAN distribution.

2.3 Installing the Debian Package

As an alternative users of Debian or Ubuntu can also install this .deb package containing the theme files as well as the Fira Sans font files.

2.4 A Minimal Example

The following code shows a minimal example of a Beamer presentation using **metropolis**.

```
\documentclass{beamer}
\usetheme{metropolis}       % Use metropolis theme
\title{A minimal example}
\date{\today}
\author{Matthias Vogelgesang}
\institute{Centre for Modern Beamer Themes}
\begin{document}
  \maketitle
  \section{First Section}
  \begin{frame}{First Frame}
    Hello, world!
  \end{frame}
\end{document}
```

2.5 Dependencies

metropolis depends on the `beamer` class and the following standard packages:

- `tikz`
- `pgfopts`
- `etoolbox`
- `calc`
- `ifxetex`
- `ifluatex`

For best results, we recommend installing the fonts `Fira Sans` and `Fira Mono` and compiling with **metropolis** using XeLaTeX or LuaTeX. These are optional dependencies; **metropolis** is compatible with (e.g.) pdfLaTeX and will fall back to standard fonts if `Fira Sans` or `Fira Mono` is not installed.

The packaged name of `Fira Sans` is `Fira Sans OT` in some Linux distributions; this case is automatically handled by **metropolis**.

2.6 Pandoc

To use this theme with Pandoc-based presentations, you can run the following command

```
$ pandoc -t beamer --latex-engine=xelatex -V theme:
   metropolis -o output.pdf input.md
```

3 Customization

3.1 Package options

The theme provides a number of options, which can be set using a key=value interface. The primary way to set options is to provide a comma-separated list of option-value pairs when loading **metropolis** in the preamble:

```
\usetheme[option1=value1, option2=value2, ...]{metropolis}
```

Options can be changed at any time — even mid-presentation! — with the `\metroset` macro.

```
\metroset{option1=newvalue1, option2=newvalue2, ...}
```

The list of options is structured as shown in the following example.

`option key` ***list of possible values*** . . . . default

A short description of the option.

3.1.1 Main theme

`titleformat` *regular, smallcaps, allsmallcaps, allcaps* . . . . regular

Changes the format of titles, subtitles, section titles, frame titles, and the text on standout "plain" frames. The available options produce Regular, SmallCaps, allsmallcaps, or ALLCAPS titles. Please refer to Section 4.1 for known issues with these options.

`titleformat-plain` *regular, smallcaps, allsmallcaps, allcaps* . . . . regular

Changes the format of standout "plain" frames (see `titleformat`, above).

3.1.2 Inner theme

`block` *transparent, fill* . . . . transparent

Optionally adds a light grey background to block environments like `theorem` and `example`.

`sectionpage` *none, simple, progressbar* . . . . progressbar

Adds a slide at the start of each section (`simple`) with an optional thin progress bar below the section title (`progressbar`). The `none` option disables the section page.

3.1.3 Outer theme

`numbering` *none, counter, fraction* . . . . counter

Controls whether the frame number at the bottom right of each slide is omitted (`none`), shown (`counter`) or displayed as a fraction of the total number of frames (`fraction`).

`progressbar` *none, head, frametitle, foot* . . . . none

Optionally adds a progress bar to the top of each frame (`head`), the bottom of each frame (`foot`), or directly below each frame title (`frametitle`).

3.1.4 Color theme

`block` *transparent, fill* . . . . transparent

Optionally adds a light grey background to block environments like `theorem` and `example`.

`background` *dark, light* . . . . light

Provides the option to have a dark background and light foreground instead of the reverse.

3.1.5 Font theme

`titleformat-title`
`titleformat-subtitle`
`titeformat-section`
`titleformat-frame`

regular, smallcaps, allsmallcaps, allcaps . . . . regular

Individually controls the format of titles, subtitles, section titles, and frame titles (see `titleformat`, above).

3.2 Color Customization

The included **metropolis** color theme is used by default, but its colors can be easily changed to suit your tastes. All of the theme's styles are defined in terms of three beamer colors:

- `normal text` (dark fg, light bg)
- `alerted text` (colored fg, should be visible against dark or light)
- `example text` (colored fg, should be visible against dark or light)

An easy way to customize the theme is to redefine these colors using

```
\setbeamercolor{ ... }{ fg= ... , bg= ... }
```

in your preamble. For greater customization, you can redefine any of the other stock beamer colors. In addition to the stock colors the theme defines a number of **metropolis** specific colors, which can also be redefined to your liking.

```
\setbeamercolor{progress bar}{ ... }
\setbeamercolor{title separator}{ ... }
\setbeamercolor{progress bar in head/foot}{ ... }
```

```
\setbeamercolor{progress bar in section page}{ ... }
```

3.3 Font Customization

The default font for **metropolis** is `Fira`. This can be easily changed using the standard font selection commands of the fontspec package. So if you prefer, for example, the `Ubuntu` font family, just add the following two commands after loading the **metropolis** theme.

```
\setsansfont{Ubuntu}
\setmonofont{Ubuntu Mono}
```

3.3.1 Old style figures

The regular fontspec mechanism for changing glyph appearance applies also to this theme. If you want to have old style figures in the text but regular lined figures for math, you could add the following to your preamble:

```
\usefonttheme{professionalfonts}   % required for mathspec
\usepackage{mathspec}
\setsansfont[BoldFont={Fira Sans},
             Numbers={OldStyle}]{Fira Sans Light}
\setmathsfont(Digits)[Numbers={Lining, Proportional}]{Fira
   Sans Light}
```

3.4 Commands

`\plain` The `\plain{title=[]}{<body>}` command sets a slide in with a plain dark background, which can be useful to focus attention on a single sentence or image.

3.5 Paul Tol's colors: a `pgfplots` theme

A good presentation uses colors that are

- distinct from each other as much as possible, and

- distinct from black and white,
- under many different lighting and display environments, and
- to color-blind viewers,
- all while matching well together.

In a technical note for SRON, Paul Tol proposed a palette of colors satisfying these constraints. The sub-package `pgfplotsthemetol` defines palettes for `pgfplots` charts based on Tol's work. Use the `mlineplot` key to plot line data and `mbarplot` or horizontal `mbarplot` to plot bar charts.

4 Known Issues

4.1 Title formats

Be aware that not every font supports small caps, so the `smallcaps` or `allsmallcaps` options may not work if you use a font other than `Fira Sans`. In particular, the Computer Modern sans-serif typeface, which is used when **metropolis** is compiled with pdfLaTeX, does not have a small-caps variant.

The title format options `allsmallcaps` and `allcaps` are quite nice from an aesthetic point of view, but their use of `\MakeLowercase` and `\MakeUppercase` can cause unexpected problems. For example:

- Some commands, like `\\`, do not work inside `\MakeLowercase` and `\MakeUppercase`. (See #125)
- Only alphabetic characters are affected by `\MakeLowercase`, so numerals and punctuation remain at full height. This can spoil some of the aesthetic benefits of `allsmallcaps`. (See #33)
- `\MakeLowercase` and `\MakeUppercase` apply to math mode and `\scshape` does not. This can easily introduce mathematical errors that are hard to catch.
- It is impossible to typeset symbols which are encoded as uppercase letters in a different font. In particular, `\mathbb` and `\mathcal` letters will be replaced by other math glyphs. (See #153)

The `allsmallcaps` and `allcaps` options are safe to use if your titles contain only alphabetic characters and do not require the expansion of any macros.

4.2 Plain Frame

The `\plain` command does not work if you override the **metropolis** color theme with the default beamer color theme `fly`.

5 License

metropolis is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. This means that if you change the theme and re-distribute it, you must retain the copyright notice header and license it under the same CC-BY-SA license. This does not affect any presentations that you create with the theme.

6 Implementation

6.1 metropolis parent theme

The primary job of this package is to load the component sub-packages of the **metropolis** theme and route the theme options accordingly. It also provides some custom commands and environments for the user.

6.1.1 Package dependencies

```
\RequirePackage{etoolbox}
\RequirePackage{pgfopts}
```

6.1.2 Options

Most options are passed off to the component sub-packages.

```
\pgfkeys{/metropolis/.cd,
  .search also={
    /metropolis/inner,
    /metropolis/outer,
    /metropolis/color,
    /metropolis/font,
  },
```

Currently, the `block` option affects two subthemes and has to be handled separately.

```
  block/.code=\pgfkeysalso{
    inner/block=#1,
    color/block=#1,
  },
}
```

`titleformat plain` Controls the formatting of the text on standout “plain” frames.

```
\pgfkeys{
  /metropolis/titleformat plain/.cd,
    .is choice,
    regular/.code={%
      \let\metropolis@plaintitleformat\@empty%
      \setbeamerfont{plain title}{shape=\normalfont}%
    },
    smallcaps/.code={%
      \let\metropolis@plaintitleformat\@empty%
      \setbeamerfont{plain title}{shape=\scshape}%
    },
    allsmallcaps/.code={%
      \let\metropolis@plaintitleformat\MakeLowercase%
      \setbeamerfont{plain title}{shape=\scshape}%
      \PackageWarning{beamerthememetropolis}{%
       Be aware that titleformat plain=allsmallcaps can lead to problems%
      }
    },
    allcaps/.code={%
      \let\metropolis@plaintitleformat\MakeUppercase%
      \setbeamerfont{plain title}{shape=\normalfont}%
      \PackageWarning{beamerthememetropolis}{%
        Be aware that titleformat plain=allcaps can lead to problems%
      }
    },
}
```

`titleformat` Sets a standard format for titles, subtitles, section titles, frame titles, and the text on standout “plain” frames.

```
\pgfkeys{
  /metropolis/titleformat/.code=\pgfkeysalso{
      font/titleformat title=#1,
      font/titleformat subtitle=#1,
      font/titleformat section=#1,
      font/titleformat frame=#1,
      titleformat plain=#1,
    }
}
```

For backwards compatibility with earlier betas of the theme, we implement deprecated option names as aliases to the corresponding `key=value` options.

```
\pgfkeys{/metropolis/.cd,
  usetitleprogressbar/.code=\pgfkeysalso{outer/progressbar=frametitle},
  noslidenumbers/.code=\pgfkeysalso{outer/numbering=none},
  usetotalslideindicator/.code=\pgfkeysalso{outer/numbering=fraction},
  nosectionslide/.code=\pgfkeysalso{inner/sectionpage=none},
  darkcolors/.code=\pgfkeysalso{color/background=dark},
  blockbg/.code=\pgfkeysalso{color/block=fill, inner/block=fill},
}
```

Set default values for options.

```
\newcommand{\metropolis@setdefaults}{
  \pgfkeys{/metropolis/.cd,
    titleformat plain=regular,
  }
}
```

6.1.3 Component sub-packages

Having processed the options, we can now load the component sub-packages of the theme.

```
\useinnertheme{metropolis}
\useoutertheme{metropolis}
\usecolortheme{metropolis}
\usefonttheme{metropolis}
```

The tol theme for pgfplots is only loaded if pgfplots is used.

```
\AtEndPreamble{%
  \@ifpackageloaded{pgfplots}{%
    \RequirePackage{pgfplotsthemetol}
  }{}
}
```

6.1.4 Custom commands

The parent theme defines custom commands as their proper usage may depend on multiple sub-packages.

\metroset Allows the user to change options midway through a presentation.

```
\newcommand{\metroset}[1]{\pgfkeys{/metropolis/.cd,#1}}
```

\plain Creates a plain frame with dark background, suitable for displaying images or a few words. The format of the text can be set with the titleformat plain option.

```
\def\metropolis@plaintitleformat#1{#1}
\newcommand{\plain}[2][]{%
  \begingroup
    \setbeamercolor{background canvas}{
      use=palette primary,
      parent=palette primary
    }
    \begin{frame}[c]{#1}
      \begin{center}
        \usebeamercolor[fg]{palette primary}
        \usebeamerfont{plain title}
        \metropolis@plaintitleformat{#2}
      \end{center}
    \end{frame}
  \endgroup
}
```

\mreducelistspacing

```
\newcommand{\mreducelistspacing}{\vspace{-\topsep}}
```

6.1.5 Process package options

```
\metropolis@setdefaults
\ProcessPgfOptions{/metropolis}
```

6.2 metropolis inner theme

A `beamer` inner theme dictates the style of the frame elements traditionally set in the “body” of each slide. These include:

- title, part, and section pages;
- itemize, enumerate, and description environments;
- block environments including theorems and proofs;
- figures and tables; and
- footnotes and plain text.

6.2.1 Package dependencies

```
\RequirePackage{etoolbox}
\RequirePackage{calc}
\RequirePackage{pgfopts}
\RequirePackage{tikz}
```

6.2.2 Options

`block` This option controls the block style.

```
\pgfkeys{
  /metropolis/inner/block/.cd,
    .is choice,
    transparent/.code=\setlength{\metropolis@blockskip}{0ex},
    fill/.code=\setlength{\metropolis@blockskip}{1ex},
}
```

`sectionpage` The `sectionpage` option defines the behaviour of the sectionpage.

```
\pgfkeys{
  /metropolis/inner/sectionpage/.cd,
    .is choice,
    none/.code=\metropolis@sectionpage@none,
    simple/.code=\metropolis@sectionpage@simple,
    progressbar/.code=\metropolis@sectionpage@progressbar,
}
```

\metropolis@inner@setdefaults Set default values for inner theme options.

```
\newcommand{\metropolis@inner@setdefaults}{
  \pgfkeys{/metropolis/inner/.cd,
    sectionpage=progressbar,
    block=transparent,
  }
}
```

6.2.3 Title page

title page Template for the title page. Each element is only typeset if it is defined by the user. If \subtitle is empty, for example, it won't leave a blank space on the title slide.

```
\setbeamertemplate{title page}{
  \begin{minipage}[b][\paperheight]{\textwidth}
   \ifx\inserttitlegraphic\@empty\else\usebeamertemplate*{title graphic}\fi
    \vfill%
    \ifx\inserttitle\@empty\else\usebeamertemplate*{title}\fi
    \ifx\insertsubtitle\@empty\else\usebeamertemplate*{subtitle}\fi
    \usebeamertemplate*{title separator}
```

Beamer's definition of \insertauthor is always nonempty, so we have to test another macro initialized by \author{...} to see if the user has defined an author. This solution was suggested by Enrico Gregorio in an answer to this Stack Exchange question.

```
    \ifx\beamer@shortauthor\@empty\else\usebeamertemplate*{author}\fi
    \ifx\insertdate\@empty\else\usebeamertemplate*{date}\fi
    \ifx\insertinstitute\@empty\else\usebeamertemplate*{institute}\fi
    \vfill
    \vspace*{1mm}
  \end{minipage}
}
```

Normal people should use \maketitle or \titlepage instead of using the title page beamer template directly. Beamer already defines these macros, but we patch them here to make the title page `[plain]` by default, remove `\@thanks`, and ensure the title frame number doesn't count.

`\maketitle` `\titlepage` Inserts the title frame, or causes the current frame to use the `title page` template.

```
\def\maketitle{%
  \ifbeamer@inframe
    \titlepage
  \else
    \frame[plain,noframenumbering]{\titlepage}
  \fi
}
\def\titlepage{%
  \usebeamertemplate{title page}
}
```

`title graphic` Set the title graphic in a zero-height box, so it doesn't change the position of other elements.

```
\setbeamertemplate{title graphic}{
  \vbox to 0pt {
    \vspace*{2em}
    \inserttitlegraphic%
  }%
  \nointerlineskip%
}
```

`title` Set the title on the title page.

```
\setbeamertemplate{title}{
  \raggedright%
  \linespread{1.0}%
  \inserttitle%
  \par%
  \vspace*{0.5em}
}
```

`subtitle` Set the subtitle on the title page.

```
\setbeamertemplate{subtitle}{
  \insertsubtitle%
  \par%
  \vspace*{0.5em}
}
```

`title separator` Template to set the title graphic in a zero-height box. (It won’t change the position of other elements.)

```
\setbeamertemplate{title separator}{
  \begin{tikzpicture}
    \draw[fg, fill=fg] (0,0) rectangle (\textwidth, 0.4pt);
  \end{tikzpicture}%
  \par%
}
```

`author` Set the author on the title page.

```
\setbeamertemplate{author}{
  \vspace*{2em}
  \insertauthor%
  \par%
  \vspace*{0.25em}
}
```

`date` Set the date on the title page.

```
\setbeamertemplate{date}{
  \insertdate%
  \par%
}
```

`institute` Set the institute on the title page.

```
\setbeamertemplate{institute}{
  \vspace*{3mm}
  \insertinstitute%
  \par%
}
```

6.2.4 Section page

section page Template for the section title slide at the beginning of each section.

```
\newcommand{\metropolis@sectionpage@none}{
  \AtBeginSection{
    % intenionally empty
  }
}
\defbeamertemplate{section page}{simple}{
  \centering
  \usebeamercolor[fg]{section title}
  \usebeamerfont{section title}
  \insertsectionhead\\
}
\newcommand{\metropolis@sectionpage@simple}{
  \setbeamertemplate{section page}[simple]
  \AtBeginSection{
    \ifbeamer@inframe
      \sectionpage
    \else
      \frame[plain,c,noframenumbering]{\sectionpage}
    \fi
  }
}
\defbeamertemplate{section page}{progressbar}{
  \centering
  \begin{minipage}{22em}
    \usebeamercolor[fg]{section title}
    \usebeamerfont{section title}
    \insertsectionhead\\[-1ex]
    \usebeamertemplate*{progress bar in section page}
  \end{minipage}
  \par
}
\newcommand{\metropolis@sectionpage@progressbar}{
  \setbeamertemplate{section page}[progressbar]
  \AtBeginSection{
    \ifbeamer@inframe
      \sectionpage
    \else
      \frame[plain,c,noframenumbering]{\sectionpage}
    \fi
  }
}
```

`progress bar in section page`

Template for the progress bar displayed by default on the section page. This code is duplicated in large part in the outer theme's template `progress bar in head/foot`.

```
\newlength{\metropolis@progressonsectionpage}
\setbeamertemplate{progress bar in section page}{
  \setlength{\metropolis@progressonsectionpage}{%
   \textwidth * \ratio{\insertframenumber pt}{\inserttotalframenumber pt}%
  }%
  \begin{tikzpicture}
    \draw[bg, fill=bg] (0,0) rectangle (\textwidth, 0.4pt);
   \draw[fg, fill=fg] (0,0) rectangle (\metropolis@progressonsectionpage, 0.4pt);
  \end{tikzpicture}%
}
```

The above code assumes that `\insertframenumber` is less than or equal to `\inserttotalframenumber`. However, this is not true on the first compile; in the absence of an `.aux` file, `\inserttotalframenumber` defaults to 1. This behaviour could cause fatal errors for long presentations, as `\metropolis@progressonsectionpage` would exceed TEX's maximum length (16383.99999pt, roughly 5.75 metres or 18.9 feet). To avoid this, we increase the default value for `\inserttotalframenumber`; presentations with over 4000 slides will still break on first compile, but users in that situation likely have deeper problems to solve.

```
\def\inserttotalframenumber{100}
```

6.2.5 Block environments

Regular block environment

```
\newlength{\metropolis@blockskip}
\setbeamertemplate{block begin}{%
  \setlength{\parskip}{\metropolis@parskip}
  \vspace*{1ex}
  \begin{beamercolorbox}[%
```

```
    ht=2.4ex,
    dp=1ex,
    leftskip=\metropolis@blockskip,
    rightskip=\metropolis@blockskip]{block title}
      \usebeamerfont*{block title}\insertblocktitle%
  \end{beamercolorbox}%
  \vspace*{-1pt}
  \usebeamerfont{block body}%
  \begin{beamercolorbox}[%
    dp=1ex,
    leftskip=\metropolis@blockskip,
    rightskip=\metropolis@blockskip,
    vmode]{block body}%
}
\setbeamertemplate{block end}{%
  \end{beamercolorbox}
  \vspace*{0.2ex}
}
```

Alerted block environment

```
\setbeamertemplate{block alerted begin}{%
  \setlength{\parskip}{\metropolis@parskip}
  \vspace*{1ex}
  \begin{beamercolorbox}[%
    ht=2.4ex,
    dp=1ex,
    leftskip=\metropolis@blockskip,
    rightskip=\metropolis@blockskip]{block title alerted}
      \usebeamerfont*{block title alerted}\insertblocktitle%
  \end{beamercolorbox}%
  \vspace*{-1pt}
  \usebeamerfont{block body alerted}%
  \begin{beamercolorbox}[%
    dp=1ex,
    leftskip=\metropolis@blockskip,
    rightskip=\metropolis@blockskip,
    vmode]{block body alerted}%
}
\setbeamertemplate{block alerted end}{%
  \end{beamercolorbox}
  \vspace*{0.2ex}
}
```

Example block environment

```
\setbeamertemplate{block example begin}{%
  \setlength{\parskip}{\metropolis@parskip}
  \vspace*{1ex}
  \begin{beamercolorbox}[%
    ht=2.4ex,
    dp=1ex,
    leftskip=\metropolis@blockskip,
    rightskip=\metropolis@blockskip]{block title example}
      \usebeamerfont*{block title example}\insertblocktitle%
  \end{beamercolorbox}%
  \vspace*{-1pt}
  \usebeamerfont{block body example}%
  \begin{beamercolorbox}[%
    dp=1ex,
    leftskip=\metropolis@blockskip,
    rightskip=\metropolis@blockskip,
    vmode]{block body example}%
}
\setbeamertemplate{block example end}{%
  \end{beamercolorbox}
  \vspace*{0.2ex}
}
```

6.2.6 Lists and floats

```
\setbeamertemplate{itemize items}{\textbullet}
\setbeamertemplate{caption label separator}{: }
\setbeamertemplate{caption}[numbered]
```

6.2.7 Footnotes

```
\setbeamertemplate{footnote}{%
  \parindent 0em\noindent%
  \raggedright
  \usebeamercolor{footnote}\hbox to 0.8em{\hfil\insertfootnotemark}\insertfootnotetext\par%
}
```

6.2.8 Text and spacing settings

```
\newlength{\metropolis@parskip}
\setlength{\metropolis@parskip}{0.5em}
\setlength{\parskip}{\metropolis@parskip}
\linespread{1.15}
```

By default, Beamer frames offer the `c` option to *almost* vertically center the text, but the placement is a little too high. To fix this, we redefine the `c` option to equalize `\beamer@frametopskip` and `\beamer@framebottomskip`. This solution was suggested by Enrico Gregorio in an answer to this Stack Exchange question.

```
\define@key{beamerframe}{c}[true]{% centered
  \beamer@frametopskip=0pt plus 1fill\relax%
  \beamer@framebottomskip=0pt plus 1fill\relax%
  \beamer@frametopskipautobreak=0pt plus .4\paperheight\relax%
  \beamer@framebottomskipautobreak=0pt plus .6\paperheight\relax%
  \def\beamer@initfirstlineunskip{}%
}
```

Process package options

```
\metropolis@inner@setdefaults
\ProcessPgfPackageOptions{/metropolis/inner}
```

6.3 metropolis outer theme

A `beamer` outer theme dictates the style of the frame elements traditionally set outside the body of each slide: the head, footline, and frame title.

6.3.1 Package dependencies

```
\RequirePackage{etoolbox}
\RequirePackage{calc}
\RequirePackage{pgfopts}
```

6.3.2 Options

`numbering` Adds slide numbers to the bottom right of each slide.

```
\pgfkeys{
  /metropolis/outer/numbering/.cd,
    .is choice,
    none/.code=\setbeamertemplate{frame numbering}[none],
    counter/.code=\setbeamertemplate{frame numbering}[counter],
    fraction/.code=\setbeamertemplate{frame numbering}[fraction],
}
```

`progressbar` Adds a progress bar to the top, bottom, or frametitle of each slide.

```
\pgfkeys{
  /metropolis/outer/progressbar/.cd,
    .is choice,
    none/.code={%
      \setbeamertemplate{headline}[plain]
      \setbeamertemplate{frametitle}[plain]
      \setbeamertemplate{footline}[plain]
    },
    head/.code={\pgfkeys{/metropolis/outer/progressbar=none}
      \addtobeamertemplate{headline}{}{%
        \usebeamertemplate*{progress bar in head/foot}
      }
    },
    frametitle/.code={\pgfkeys{/metropolis/outer/progressbar=none}
      \addtobeamertemplate{frametitle}{}{%
        \usebeamertemplate*{progress bar in head/foot}
      }
    },
    foot/.code={\pgfkeys{/metropolis/outer/progressbar=none}
      \addtobeamertemplate{footline}{}{%
        \usebeamertemplate*{progress bar in head/foot}%
      }
    },
}
```

`\metropolis@outer@setdefaults` Sets default values for outer theme options.

```
\newcommand{\metropolis@outer@setdefaults}{
  \pgfkeys{/metropolis/outer/.cd,
    numbering=counter,
    progressbar=none,
  }
}
```

6.3.3 Head and footline

All good `beamer` presentations should already remove the navigation symbols, but **metropolis** removes them automatically (just in case).

```
\setbeamertemplate{navigation symbols}{}
```

`frame numbering`

Templates for the frame number. Can be omitted, shown or displayed as a fraction of the total frames.

```
\defbeamertemplate{frame numbering}{none}{}
\defbeamertemplate{frame numbering}{counter}{\insertframenumber}
\defbeamertemplate{frame numbering}{fraction}{
  \insertframenumber/\inserttotalframenumber
}
```

`headline`
`footline`

Templates for the head- and footline at the top and bottom of each frame.

```
\defbeamertemplate{headline}{plain}{}
\defbeamertemplate{footline}{plain}{%
  \begin{beamercolorbox}[wd=\textwidth, sep=3ex]{footline}%
    \hfill%
    \usebeamerfont{page number in head/foot}%
    \usebeamertemplate*{frame numbering}
  \end{beamercolorbox}%
}
```

6.3.4 Frametitle

`frametitle`

Templates for the frame title, which is optionally underlined with a progress bar.

```
\newcommand{\metropolis@frametitlestrut}{
  \vphantom{ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz()}%
}
\defbeamertemplate{frametitle}{plain}{%
  \nointerlineskip%
  \begin{beamercolorbox}[%
      wd=\paperwidth,%
      sep=1.5ex,%
    ]{frametitle}%
  \metropolis@frametitlestrut\insertframetitle\metropolis@frametitlestrut%
  \end{beamercolorbox}%
}
```

`progress bar in head/foot` Template for the progress bar optionally displayed below the frame title on each page. Much of this code is duplicated in the inner theme's template `progress bar in section page`.

```
\newlength{\metropolis@progressinheadfoot}
\setbeamertemplate{progress bar in head/foot}{
  \nointerlineskip
  \setlength{\metropolis@progressinheadfoot}{%
   \paperwidth * \ratio{\insertframenumber pt}{\inserttotalframenumber pt}%
  }%
  \begin{beamercolorbox}[wd=\paperwidth]{progress bar in head/foot}
    \begin{tikzpicture}
      \draw[bg, fill=bg] (0,0) rectangle (\paperwidth, 0.4pt);
    \draw[fg, fill=fg] (0,0) rectangle (\metropolis@progressinheadfoot, 0.4pt);
    \end{tikzpicture}%
  \end{beamercolorbox}
}
```

6.3.5 Process package options

```
\metropolis@outer@setdefaults
\ProcessPgfPackageOptions{/metropolis/outer}
```

6.4 metropolis font theme

A `beamer` font theme sets the style of the font used in the document.

6.4.1 Package dependencies

```
\RequirePackage{etoolbox}
\RequirePackage{ifxetex}
\RequirePackage{ifluatex}
\RequirePackage{pgfopts}
```

6.4.2 Load Fira fonts

If the presentation is compiled with XeLaTeX or LuaLaTeX, the fontspec package is loaded and we search for the `Fira` fonts.

```
\ifboolexpr{bool {xetex} or bool {luatex}}{
  \RequirePackage[no-math]{fontspec}
```

`\checkfont` Checks if a font is installed; if not, `fontsnotfound` is increased.

```
  \newcounter{fontsnotfound}
  \newcommand{\checkfont}[1]{%
    \suppressfontnotfounderror=1%
    \font\x = "#1" at 10pt
    \selectfont
    \ifx\x\nullfont%
      \stepcounter{fontsnotfound}%
    \fi%
    \suppressfontnotfounderror=0%
  }

```

`\iffontsavailable` Resets the `fontsnotfound` counter and calls `\checkfont` for each font in the comma separated list in the first argument.

```
  \newcommand{\iffontsavailable}[3]{%
    \setcounter{fontsnotfound}{0}%
    \expandafter\forcsvlist\expandafter%
    \checkfont\expandafter{#1}%
    \ifnum\value{fontsnotfound}=0%
      #2%
    \else%
      #3%
    \fi%
  }
```

We search for regular, italic, light, light italic, mono, and mono bold fonts under the default `Fira Sans` and `Fira Mono` names. If this fails, the suffix OT — used by some Linux distributions — will be tried. If this also fails, a warning will be displayed and the standard fonts will be used.

```
  \iffontsavailable{Fira Sans Light,%
                Fira Sans Light Italic,%
                Fira Sans,%
                Fira Sans Italic}{%
    \setsansfont[BoldFont={Fira Sans}]{Fira Sans Light}%
  }{%
    \iffontsavailable{Fira Sans Light OT,%
                  Fira Sans Light Italic OT,%
                  Fira Sans OT,%
                  Fira Sans Italic OT}{%
      \setsansfont[BoldFont={Fira Sans OT}]{Fira Sans Light OT}%
    }{%
      \PackageWarning{beamerthememetropolis}{%
        Could not find Fira Sans fonts%
      }
    }
  }
  \iffontsavailable{Fira Mono, Fira Mono Bold}{%
    \setmonofont{Fira Mono}%
  }{%
    \iffontsavailable{Fira Mono OT, Fira Mono Bold OT}{%
      \setmonofont{Fira Mono OT}%
    }{%
      \PackageWarning{beamerthememetropolis}{%
        Could not find Fira Mono fonts%
      }
    }
  }
  \AtBeginEnvironment{tabular}{%
    \addfontfeature{Numbers={Monospaced}}%
  }
}{%
  \PackageWarning{beamerthememetropolis}{%
    You need to compile with XeLaTeX or LuaLaTeX to use the Fira fonts%
  }
}
```

This concludes the portion of the code which is only run when compiled with XeLATEX or LuaLATEX. The remainder of this package applies regardless of the compiling engine.

6.4.3 General font definitions

```
\setbeamerfont{title}{size=\Large,%
                      series=\bfseries}
\setbeamerfont{author}{size=\small}
\setbeamerfont{date}{size=\small}
\setbeamerfont{section title}{size=\Large,%
                              series=\bfseries}
\setbeamerfont{plain title}{size=\Large,%
                            series=\bfseries}
\setbeamerfont{block title}{size=\normalsize,%
                            series=\bfseries}
\setbeamerfont{block title alerted}{size=\normalsize,%
                                    series=\bfseries}
\setbeamerfont*{subtitle}{size=\large}
\setbeamerfont{frametitle}{size=\large,%
                           series=\bfseries}
\setbeamerfont{caption}{size=\small}
\setbeamerfont{caption name}{series=\bfseries}
\setbeamerfont{description item}{series=\bfseries}
\setbeamerfont{page number in head/foot}{size=\scriptsize}
\setbeamerfont{bibliography entry author}{size=\normalsize,%
                                          series=\normalfont}
\setbeamerfont{bibliography entry title}{size=\normalsize,%
                                         series=\bfseries}
\setbeamerfont{bibliography entry location}{size=\normalsize,%
                                            series=\normalfont}
\setbeamerfont{bibliography entry note}{size=\small,%
                                        series=\normalfont}
```

6.4.4 Title format options

`titleformat title` Controls the format of the title.

```
\pgfkeys{
  /metropolis/font/titleformat title/.cd,
    .is choice,
    regular/.code={%
      \let\metropolis@titleformat\@empty%
      \setbeamerfont{title}{shape=\normalfont}%
    },
    smallcaps/.code={%
      \let\metropolis@titleformat\@empty%
      \setbeamerfont{title}{shape=\scshape}%
    },
    allsmallcaps/.code={%
      \let\metropolis@titleformat\lowercase%
      \setbeamerfont{title}{shape=\scshape}%
      \PackageWarning{beamerthememetropolis}{%
       Be aware that titleformat title=allsmallcaps can lead to problems%
      }
    },
    allcaps/.code={%
      \let\metropolis@titleformat\uppercase%
      \setbeamerfont{title}{shape=\normalfont}
      \PackageWarning{beamerthememetropolis}{%
        Be aware that titleformat title=allcaps can lead to problems%
      }
    },
}
```

`titleformat subtitle` Control the format of the subtitle.

```
\pgfkeys{
  /metropolis/font/titleformat subtitle/.cd,
    .is choice,
    regular/.code={%
      \let\metropolis@subtitleformat\@empty%
      \setbeamerfont{subtitle}{shape=\normalfont}%
    },
    smallcaps/.code={%
      \let\metropolis@subtitleformat\@empty%
      \setbeamerfont{subtitle}{shape=\scshape}%
    },
    allsmallcaps/.code={%
      \let\metropolis@subtitleformat\lowercase%
      \setbeamerfont{subtitle}{shape=\scshape}%
      \PackageWarning{beamerthememetropolis}{%
      Be aware that titleformat subtitle=allsmallcaps can lead to problems%
      }
    },
    allcaps/.code={%
      \let\metropolis@subtitleformat\uppercase%
      \setbeamerfont{subtitle}{shape=\normalfont}%
      \PackageWarning{beamerthememetropolis}{%
        Be aware that titleformat subtitle=allcaps can lead to problems%
      }
    },
}
```

`titleformat section` Controls the format of the section title.

```
\pgfkeys{
  /metropolis/font/titleformat section/.cd,
    .is choice,
    regular/.code={%
      \let\metropolis@sectiontitleformat\@empty%
      \setbeamerfont{section title}{shape=\normalfont}%
    },
    smallcaps/.code={%
      \let\metropolis@sectiontitleformat\@empty%
      \setbeamerfont{section title}{shape=\scshape}%
    },
    allsmallcaps/.code={%
      \let\metropolis@sectiontitleformat\MakeLowercase%
      \setbeamerfont{section title}{shape=\scshape}%
      \PackageWarning{beamerthememetropolis}{%
       Be aware that titleformat section=allsmallcaps can lead to problems%
      }
    },
    allcaps/.code={%
      \let\metropolis@sectiontitleformat\MakeUppercase%
      \setbeamerfont{section title}{shape=\normalfont}%
      \PackageWarning{beamerthememetropolis}{%
        Be aware that titleformat section=allcaps can lead to problems%
      }
    },
}
```

`frametitleformat` Control the format of the frame title.

```
\pgfkeys{
  /metropolis/font/titleformat frame/.cd,
    .is choice,
    regular/.code={%
      \let\metropolis@frametitleformat\@empty%
      \setbeamerfont{frametitle}{shape=\normalfont}%
    },
    smallcaps/.code={%
      \let\metropolis@frametitleformat\@empty%
      \setbeamerfont{frametitle}{shape=\scshape}%
    },
    allsmallcaps/.code={%
      \let\metropolis@frametitleformat\MakeLowercase%
      \setbeamerfont{frametitle}{shape=\scshape}%
      \PackageWarning{beamerthememetropolis}{%
       Be aware that titleformat frame=allsmallcaps can lead to problems%
      }
    },
    allcaps/.code={%
      \let\metropolis@frametitleformat\MakeUppercase%
      \setbeamerfont{frametitle}{shape=\normalfont}
      \PackageWarning{beamerthememetropolis}{%
        Be aware that titleformat frame=allcaps can lead to problems%
      }
    },
}
```

`titleformat aliases` Allows `titleformat title` et al. to be used in the `\usetheme` declaration, where LaTeX automatically removes all spaces.

```
\pgfkeys{
  /metropolis/font/.cd,
  titleformattitle/.code=\pgfkeysalso{titleformat title=#1},
  titleformatsubtitle/.code=\pgfkeysalso{titleformat subtitle=#1},
  titleformatsection/.code=\pgfkeysalso{titleformat section=#1},
  titleformatframe/.code=\pgfkeysalso{titleformat frame=#1},
}
```

`\metropolis@font@setdefaults` Sets default values for font theme options.

```
\newcommand{\metropolis@font@setdefaults}{
  \pgfkeys{/metropolis/font/.cd,
    titleformat title=regular,
    titleformat subtitle=regular,
    titleformat section=regular,
    titleformat frame=regular,
  }
}
```

We first define hooks to change the case format of the titles.

```
\def\metropolis@titleformat#1{#1}
\def\metropolis@subtitleformat#1{#1}
\def\metropolis@sectiontitleformat#1{#1}
\def\metropolis@frametitleformat#1{#1}
```

To make the uppercase and lowercase macros work in the title, subtitle, etc., we have to patch the appropriate `beamer` commands that set their values. This solution was suggested by Enrico Gregorio in an answer to this StackExchange question.

```
\patchcmd{\beamer@title}%
  {\def\inserttitle{#2}}%
  {\def\inserttitle{\metropolis@titleformat{#2}}}%
  {}%
  {\PackageError{beamerfontthememetropolis}{Patching title failed}}
\patchcmd{\beamer@subtitle}%
  {\def\insertsubtitle{#2}}%
  {\def\insertsubtitle{\metropolis@subtitleformat{#2}}}%
  {}%
  {\PackageError{beamerfontthememetropolis}{Patching subtitle failed}}
\patchcmd{\sectionentry}
  {\def\insertsectionhead{#2}}
  {\def\insertsectionhead{\metropolis@sectiontitleformat{#2}}}
  {}
  {\PackageError{beamerfontthememetropolis}{Patching section title failed}}
\patchcmd{\beamer@section}
  {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
  {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{%
    \metropolis@sectiontitleformat{#1}}}}
  {}
  {\PackageError{beamerfontthememetropolis}{Patching section title failed}}
```

Similarly, to make the `\MakeLowercase` and `\MakeUppercase` macros work in the frame title we have to patch `\beamer@@frametitle`.

```
\patchcmd{\beamer@@frametitle}
  {\beamer@ifempty{#2}{}{%
    \gdef\insertframetitle{{#2\ifnum\beamer@autobreakcount>0\relax{}\space%
      \usebeamertemplate*{frametitle continuation}\fi}}%
    \gdef\beamer@frametitle{#2}%
    \gdef\beamer@shortframetitle{#1}%
    }}
  {\beamer@ifempty{#2}{}{%
      \gdef\insertframetitle{{\metropolis@frametitleformat{#2}\ifnum%
      \beamer@autobreakcount>0\relax{}\space%
      \usebeamertemplate*{frametitle continuation}\fi}}%
    \gdef\beamer@frametitle{#2}%
    \gdef\beamer@shortframetitle{#1}%
    }}
  {}
  {\PackageError{beamerfontthememetropolis}{Patching frame title failed}}
```

6.4.5 Process package options

```
\metropolis@font@setdefaults
\ProcessPgfPackageOptions{/metropolis/font}
```

6.5 metropolis color theme

6.5.1 Package dependencies

```
\RequirePackage{pgfopts}
```

6.5.2 Options

`block` Controls whether block environments are filled or transparent.

```
\pgfkeys{
  /metropolis/color/block/.cd,
    .is choice,
    transparent/.code=\metropolis@block@transparent,
    fill/.code=\metropolis@block@fill,
}
```

`colors` Provides the option to have a dark background and light foreground instead of the reverse.

```
\pgfkeys{
  /metropolis/color/background/.cd,
    .is choice,
    dark/.code=\metropolis@colors@dark,
    light/.code=\metropolis@colors@light,
}
```

`\metropolis@color@setdefaults` Sets default values for color theme options.

```
\newcommand{\metropolis@color@setdefaults}{
  \pgfkeys{/metropolis/color/.cd,
    background=light,
    block=transparent,
  }
}
```

6.5.3 Base colors

```
\definecolor{mDarkBrown}{HTML}{604c38}
\definecolor{mDarkTeal}{HTML}{23373b}
\definecolor{mLightBrown}{HTML}{EB811B}
\definecolor{mLightGreen}{HTML}{14B03D}
```

6.5.4 Base styles

All colors in **metropolis** are derived from the definitions of `normal text`, `alerted text`, and `example text`.

```
\newcommand{\metropolis@colors@dark}{
  \setbeamercolor{normal text}{%
    fg=black!2,
    bg=mDarkTeal
  }
}
\newcommand{\metropolis@colors@light}{
  \setbeamercolor{normal text}{%
    fg=mDarkTeal,
    bg=black!2
  }
}
\setbeamercolor{alerted text}{%
  fg=mLightBrown
}
\setbeamercolor{example text}{%
  fg=mLightGreen
}
```

6.5.5 Derived colors

The titles and structural elements (e.g. `itemize` bullets) are set in the same color as `normal text`. This would ideally done by setting `normal text` as a parent style, which we do to set `titlelike`, but this doesn't work for `structure` as its foreground is set explicitly in `beamercolorthemedefault.sty`.

```
\setbeamercolor{titlelike}{use=normal text, parent=normal text}
\setbeamercolor{author}{use=normal text, parent=normal text}
\setbeamercolor{date}{use=normal text, parent=normal text}
\setbeamercolor{institute}{use=normal text, parent=normal text}
\setbeamercolor{structure}{use=normal text, fg=normal text.fg}
```

The "primary" palette should be used for the most important navigational elements, and possibly of other elements. **metropolis** uses it for frame titles and slides.

```
\setbeamercolor{palette primary}{%
  use=normal text,
  fg=normal text.bg,
  bg=normal text.fg
}
\setbeamercolor{frametitle}{%
  use=palette primary,
  parent=palette primary
}
```

The **metropolis** inner or outer themes optionally display progress bars in various locations. Their color is set by `progress bar` but the two different kinds can be customized separately. The horizontal rule on the title page is also set based on the progress bar color and can be customized with `title separator`.

```
\setbeamercolor{progress bar}{%
  use=alerted text,
  fg=alerted text.fg,
  bg=alerted text.fg!50!black!30
}
\setbeamercolor{title separator}{
  use=progress bar,
  parent=progress bar
}
\setbeamercolor{progress bar in head/foot}{%
  use=progress bar,
  parent=progress bar
}
\setbeamercolor{progress bar in section page}{
  use=progress bar,
  parent=progress bar
}
```

Blocks

```
\newcommand{\metropolis@block@transparent}{
  \setbeamercolor{block title}{use=normal text, parent=normal text}
}
\newcommand{\metropolis@block@fill}{
  \setbeamercolor{block title}{%
    use=normal text,
    fg=normal text.fg,
    bg=normal text.bg!80!fg
  }
}
\setbeamercolor{block title alerted}{%
    use={block title, alerted text},
    bg=block title.bg,
    fg=alerted text.fg
}
\setbeamercolor{block title example}{%
    use={block title, example text},
    bg=block title.bg,
    fg=example text.fg
}
\setbeamercolor{block body alerted}{use=block body, parent=block body}
\setbeamercolor{block body example}{use=block body, parent=block body}
\setbeamercolor{block body}{
  use={block title, normal text},
  bg=block title.bg!50!normal text.bg
}
```

Footnotes

```
\setbeamercolor{footnote}{fg=normal text.fg!90}
\setbeamercolor{footnote mark}{fg=.}
```

Process package options

```
\metropolis@color@setdefaults
\ProcessPgfPackageOptions{/metropolis/color}
```

```
\mode<all>
```

6.6 Tol pgfplots theme

Paul Tol's 12-color palette[1] is as follows:

```
\definecolor{TolDarkPurple}{HTML}{332288}
\definecolor{TolDarkBlue}{HTML}{6699CC}
\definecolor{TolLightBlue}{HTML}{88CCEE}
\definecolor{TolLightGreen}{HTML}{44AA99}
\definecolor{TolDarkGreen}{HTML}{117733}
\definecolor{TolDarkBrown}{HTML}{999933}
\definecolor{TolLightBrown}{HTML}{DDCC77}
\definecolor{TolDarkRed}{HTML}{661100}
\definecolor{TolLightRed}{HTML}{CC6677}
\definecolor{TolLightPink}{HTML}{AA4466}
\definecolor{TolDarkPink}{HTML}{882255}
\definecolor{TolLightPurple}{HTML}{AA4499}
```

To use these colors, we describe "cycle lists" from which PGF chooses styles for the different series in a chart.

[1] Tol actually describes several palettes; these colours are taken from the bottom row of Figure 3 in his technical note.

mbarplot cycle Colors and styles intended for bar charts with up to 12 series.

```
\pgfplotscreateplotcyclelist{mbarplot cycle}{%
  {draw=TolDarkBlue,    fill=TolDarkBlue!70},
  {draw=TolLightBrown,  fill=TolLightBrown!70},
  {draw=TolLightGreen,  fill=TolLightGreen!70},
  {draw=TolDarkPink,    fill=TolDarkPink!70},
  {draw=TolDarkPurple,  fill=TolDarkPurple!70},
  {draw=TolDarkRed,     fill=TolDarkRed!70},
  {draw=TolDarkBrown,   fill=TolDarkBrown!70},
  {draw=TolLightRed,    fill=TolLightRed!70},
  {draw=TolLightPink,   fill=TolLightPink!70},
  {draw=TolLightPurple, fill=TolLightPurple!70},
  {draw=TolLightBlue,   fill=TolLightBlue!70},
  {draw=TolDarkGreen,   fill=TolDarkGreen!70},
}
```

mlineplot cycle Colors and styles intended for line charts with up to 4 series.

```
\pgfplotscreateplotcyclelist{mlineplot cycle}{%
  {TolDarkBlue, mark=*, mark size=1.5pt},
  {TolLightBrown, mark=square*, mark size=1.3pt},
  {TolLightGreen, mark=triangle*, mark size=1.5pt},
  {TolDarkBrown, mark=diamond*, mark size=1.5pt},
}
```

However, the above cycle lists are not applied automatically. We still need to define styles — `mlineplot` and `mbarplot` — that the user can apply to the axis of a `pgfplots` chart to use the colors. We'll also take the opportunity to adjust the display of chart axes when these styles are used.

```
\pgfplotsset{
  compat=1.9,
```

mlineplot A style to apply to the axis of a PGF line plot.

```
  mlineplot/.style={
    mbaseplot,
    xmajorgrids=true,
    ymajorgrids=true,
    major grid style={dotted},
  axis x line=bottom,
  axis y line=left,
  legend style={
    cells={anchor=west},
    draw=none
  },
  cycle list name=mlineplot cycle,
},
```

mbarplot
horizontal mbarplot

A style to apply to the axis of a PGF bar chart. mbarplot uses vertical bars by default, while horizontal mbarplot has horizontal bars as the name implies. Their shared properties are factored out into the internal style mbarplot base.

```
mbarplot base/.style={
  mbaseplot,
  bar width=6pt,
  axis y line*=none,
},
mbarplot/.style={
  mbarplot base,
  ybar,
  xmajorgrids=false,
  ymajorgrids=true,
  area legend,
  legend image code/.code={%
    \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
  },
  cycle list name=mbarplot cycle,
},
horizontal mbarplot/.style={
  mbarplot base,
  xmajorgrids=true,
  ymajorgrids=false,
  xbar stacked,
  area legend,
  legend image code/.code={%
    \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
  },
  cycle list name=mbarplot cycle,
},
```

`mbaseplot` Adjusts the appearance of the axes in a PGF chart.

```
 mbaseplot/.style={
   legend style={
     draw=none,
     fill=none,
     cells={anchor=west},
   },
   x tick label style={
     font=\footnotesize
   },
   y tick label style={
     font=\footnotesize
   },
   legend style={
     font=\footnotesize
   },
   major grid style={
     dotted,
   },
   axis x line*=bottom,
 },
 disable thousands separator/.style={
   /pgf/number format/.cd,
     1000 sep={}
 },
}
```