DEUTSCHE
BUNDESBANK
EUROSYSTEM

bbkplot

Deutsche Bundesbank Corporate Design in R

Package version 1.10
Available in Bundesbank environment

Deutsche Bundesbank, Research Data and Service Centre

Hendrik Christian Doll
Daniel Ollech
Frederik Hering
Siddharth Marya

Research Data and
Service Centre

Abstract

We introduce the *bbkplot* package, in order to facilitate user-friendly and reliable production of data-driven graphics in accordance with Deutsche Bundesbank's corporate design. The presented package allows conversion of figures and tables into graphics precisely adhering to corporate design standards. These standards specify formatting requirements for graphics in great detail in order to ensure consistency in public brand appearance. *bbkplot* automates the exact application of these requirements and enables quick and easy production of high-quality graphics by the user. We see two value propositions from the package. (1) By advancing from pre-specified graphic types that traditional graphics suites implement to flexible plot creation, *bbkplot* efficiently allows creation of a wider range of graphics in corporate design; (2) by facilitating low-threshold user creation of simple graphics for presentations, we support a more comprehensive external brand appearance.

Keywords: R, R-package, data visualization, corporate design

Citation: Doll, H.C., Hering, F., Marya, S. and D. Ollech (2020). "bbkplot. Deutsche Bundesbank Corporate Design in R", Technical Report 2020-03, Deutsche Bundesbank, Frankfurt a.M.

Contents

1 Introduction

Deutsche Bundesbank regularly publishes official reports, research papers, and gives presentations as part of its mandate, research and outreach activities. Many of these releases disseminate results of data-driven analysis. An important aspect of these publications and of communicating data-based results in general is visualisation. Visualisations are important to facilitate better understanding of the data and thus, Bundesbank publications are replete with figures and tables.[1)]

Naturally, to guarantee that these visualisations are standardised across publications, Bundesbank corporate design standards specify detailed formatting requirements for graphics in order to ensure consistency in public brand appearance. Specialised units set figures and graphs manually using commercial software. This process ensures high-quality visualisations in Bundesbank print publications.

Figures included in Bundesbank flagship print publications, such as monthly reports, are often quite specific and the result of a complex process of coordination. In contrast, the creation of graphics for presentations involves less parties. In these cases, easy-to-use open source software that produces graphics exactly adhering to corporate design has a great utility attached to it. It allows fast and independent creation and editing of graphics of high quality and on-the-fly.

Until now, the possibility of user-created graphics adhering to the Bundesbank corporate design is limited, as commercial graphical software programs are complex and not readily available to analysts. We introduce the possibility of easy user-friendly creation of data-driven visualisations, which adhere to corporate design standards. For this purpose, we implement the *bbkplot* package in the widely used open-source software R, which is especially tailored to statistical analysis and data visualisation.

We see two value propositions from our package. First, by advancing from pre-specified graphic types that traditional graphics suites implement to flexible plot creation, *bbkplot* efficiently allows creation of a wider range of graphics in corporate design. Such types of visualisations include traditional boxplots, barplots, scatterplots, etc. but also new forms of network plots, maps and interactive plots. Second, we aspire to improve external brand appearance by facilitating low-threshold user creation of graphics for presentations and thereby a wider usage of corporate design.

[1] The views expressed in this technical report are personal views of the authors and do not necessarily reflect the views of Deutsche Bundesbank ore the Eurosystem.

2 Corporate Design Guides

2.1 Bundesbank Corporate Design Manual

It is increasingly important to mandate a corporate design that ensures a consistent public appearance in published visualisations, since data-driven results become increasingly important in communication. Corporate design is an effective tool for both publishers and consumers of information as it enables publishers to establish a visual identity. At the same time it facilitates users to verify the source of information. The benefit of using such guides is not limited to generating consistency, it also leads to efficiency gains by saving time in generating documents and to create a professional look in documents (Allen, 1996).

To ensure brand recognition, it is important for the data visualisations at Bundesbank to be standardised and consistent as they act as an indicator of its mission and corporate identity. Elements of Bundesbank's corporate design include its trade marks, logo and consistent design elements, such as colours, fonts, standardised wording (such as common abbreviations, etc.), layout (such as tables of content, page layout, info boxes, etc.), images, and charts. These are recorded in the Bundesbank's Corporate Design Manual (Deutsche Bundesbank, 2011).

Consistency of data-based figures, plots, tables and diagrams is key in creating a uniform corporate communication, since they are a common form of illustration in Bundesbank publications. The Corporate Design Manual ensures this by defining a broad set of rules for designing figures and tables in Bundesbank publications. The Corporate Design Manual also lists the particular colour shades that should be used. These are referred to as Bundesbank colours hereinafter. The manual ensures that all figures have the same structure. This structure can be divided into three parts, namely:

- The header: This section contains the title, subtitle and number. These elements are optional and the background colour of the header is Bundesbank grey. The size of the header must be at least two times the format size and can be increased if more space is required.
- The plotting area: The plotting area contains at least one figure and optionally the legend.
- The footer: The footer contains a label with ”Deutsche Bundesbank”, which is mandated by the manual and the source which is optional. It is further possible to move the legend into the footer.

The three distinct parts of a figure are divided by a white line of width 2 points. The manual also lists the requirements regarding the font and font size of the different elements. Some of these are listed below for reference. Note that this enumeration is not comprehensive, but serves to showcase the degree of detail of the guidelines:

- The title of the graphic is set in the size of the continuous text. This can range from 9 to 12 pt. All other text element sizes are relative to the title. The size of supplementary text is scaled to 70 per cent of the title size. However, this should be at least 6 pt.
- The font style of the title is Frutiger Next Medium. The font style for the supplementary text is Frutiger Next Light.
- The style guide further requires 2 mm space on the borders of every element and certain margins between text lines.

Further extended guidelines are set for exact color codes and their mandated shadings, axis formatting, grid lines, legend placement and various special types of charts (such as box plots, bar charts, maps, etc.).

2.2 Existing Corporate Design Packages

Given the thrust towards digitisation and disruption from open source technologies, it is not surprising that several authors have developed software and libraries that accommodate their respective corporate designs using open source software R. Given that users, such as analysts and journalists, become increasingly data-savvy, there is a large efficiency gain in giving users the proper tools to convert their figures right away.

We particularly draw inspiration from *bbplot* (Stylianou et al., 2018), which is an open source R package built by the British Broadcasting Company (BBC) for production of figures. The BBC team went on to publish their charts using their package in 2018. A starting point for the package was the implementation of corporate colour palettes as described by Jackson (2018). Bion, Chang, and Goodman (2017) outline how Airbnb uses R for data visualisation to create branded graphs in addition to other stages of the analysis pipeline.

There even exists a dedicated *ggthemes* package that allows plotting in the style of the Economist, FiveThirtyEight, Wall Street Journal, among others (Arnold et al., 2020). Another more general example is the *ggCorpIdent* package (Klein and Wagner, 2018) which in theory can be used to implement any corporate design for ggplot2 graphics. Similarly, Neth and Gradwohl (2019) developed the *unikn* package that allows to either use the corporate design colours and text functions of the University of Konstanz or adapt these functions to the needs of other institutions.

The Bundesbank use case however incorporates rules that go beyond a simple *ggplot2* implementation. Therefore, we could not entirely rely on any of the above packages. Due to concerns of unauthorized publication of graphics in corporate design, we are unable to share the package source code unfortunately, hence we limit ourselves to describing the package and showcasing its functionality in the current paper.

3 The package: bbkplot

3.1 Design

The primary goal of the *bbkplot* package is to develop an open source solution to visualise data adhering to the Corporate Design Manual of the Bundesbank. It should fulfil the various corporate design requirements outlined above and enable Bundesbank employees to easily produce high quality graphics and include them in documents and presentations automatically.

We developed the *bbkplot* package with a set of design objectives in mind: First, the functions should be easy to use, even for inexperienced R users. Second, the package should integrate well in the working process of R users, i.e. the notation and structure should be in line with the customs in R. Third, the package should strike a good balance between including as many plot types and relying on as few functions as possible.

With this in mind, a fruitful starting point is the *ggplot2* package developed by Wickham et al. (2019). The package is probably the most widely-used and reputable visualisation package in R and allows far-reaching customisation of visualisations, which include all important plot types. Additionally, a large number of packages have been developed that extend *ggplot2* graphical capacities even further, including network, radar, geographic, and interactive graphics.

Basically, the *bbkplot* package is an add-on to *ggplot2*. Users can create graphics using the ggplot2 infrastructure and translate the layout to the Bundesbank design using designated functions of the *bbkplot* package. However, we advance from *ggplot2*, by allowing for grid graphics and, to a limited extent, even base graphics. Furthermore, for time series, a function is available to directly plot the data.

3.2 Functions

Currently, the *bbkplot* package includes functions for creating Bundesbank tables and graphics, changing the layout of a plot to the Bundesbank corporate design and helper functions, that import the Bundesbank font, define Bundesbank colour scheme and save plots and tables.[2)]

3.2.1 bbkplot

The function *bbkplot()* is designed to convert *ggplot* objects into Bundesbank style graphics.[3)] The function primarily uses the *ggplot2*, *grid* and *gridExtra* package and arranges the three distinct areas of a graph as mentioned above namely, the header, the plotting area and the footer.

Adding the header and footer to the plotting area is achieved using the gridExtra package, the class of the resulting object changes from *ggplot2* to *grid*. Consequently, after applying the *bbkplot()*

[2] Though the Bundesbank corporate style guide mandates the use of British English, which we aspire to use in this paper, the package and parameter names are in American English, as this is the lingua franca of R.

[3] The function also adds Bundesbank style header and footer to objects of class grid. These are often multiple plots generated by ggplot2. This grid objects need to be in the Bundesbank style for the whole plot generated by bbkplot to be in the Bundesbank style. This can usually be achieved by adding theme_bbk() to parts of the grid object.

function, it is not possible to change the design of the plot using *ggplot2* functions. Put differently, all changes to the non-bbk specific layout and plot elements should be added before calling the *bbkplot()* function. Still, some additional changes can also be achieved using the *bbkplot()* function itself.

There are several arguments to the function call and these can be used to personalise the graphical object, e.g. by adding title and subtitle to the plot. Table 1 (see appendix) summarises the arguments of *bbkplot()*. Figure 1 shows the converted bbkplot object (left) and the original ggplot (right). Further customisations and usage examples are presented in the following sections.

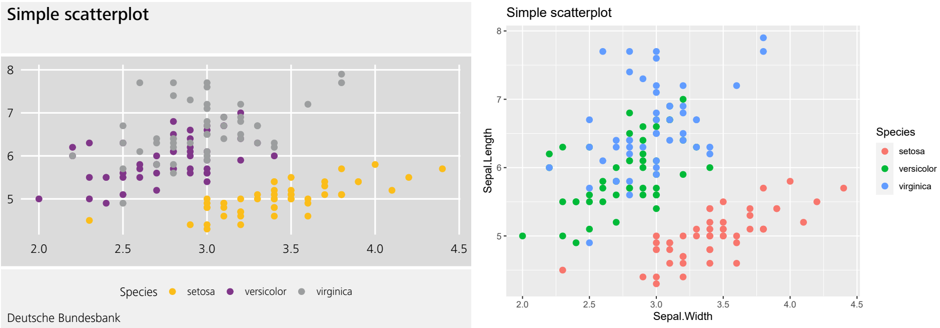


Figure 1: On the left is the bbkplot result, on the right the original plot

3.2.2 theme_bbk

In *ggplot2* and other packages, there exist a wide array of *theme_x()* type functions that define the general layout of a plot. The above mentioned *ggthemes* package incorporates various corporate styles via specifying themes. Similarly, the bbkplot package includes the *theme_bbk()* function, which can be used to change the style of the plotting area in accordance with Bundesbank's corporate design. The aforementioned *bbkplot()* function is the workhorse for adapting the plotting area.

In contrast to calling *bbkplot()* directly, using *theme_bbk()* enables extra calls to *ggplot2::theme()* to further personalise the plot. To add the Bundesbank corporate design specific footer and header, it is then necessary to call *bbkplot()* afterwards with the parameter setting *add_theme_bbk = FALSE*. This setting suppresses another call to *theme_bbk()*, which would potentially replace the plot adjustments made.

This is useful for instance, when one may need very specific settings, which usually would be overriden by *theme_bbk()*. Such examples could potentially include particular customised grid lines, axis labels or legend positioning. Also, it is often preferable to use *theme_bbk()* when working with other packages extending the possibilities of ggplot2.[4] The arguments of *theme_bbk()* are presented in Table 3 (see appendix).

[4] Examples include the packages *ggiraph* and *plotly* that allow interactive plotting. Up-to-date examples of creating interactive Bundesbank graphics can be found in the vignette of the *bbkplot* package.

3.2.3 tsplot

Due to the relevance of time series to economists, a function is specifically dedicated to displaying and manipulating time series. For visualising the data the *tsplot()* function relies heavily on the *bbkplot()* function. The additional benefit is twofold. First, as a consequence of the peculiarities of the time axis, reformatting time series is often quite cumbersome. Therefore, *tsplot()* accepts time series as input instead of a ggplot object as in *bbkplot()*. Second, *tsplot()* adds several functionalities for transforming the input and the output.

By default, the time series is depicted as a line plot and the function uses sensible decision rules to identify a meaningful date format for the x-axis. Using the parameters transform and type, it is easy to produce a figure of the period-to-period rate of change as a bar plot instead. As date axis ticks and labels follow very detailed style guidelines, special care is taken to ensure easy and user-friendly implementation of these. The arguments for *tsplot()* are summarised in Table 4 (see appendix).

The input to *tsplot()* needs to be (multiple) time series, ideally of class xts from the *xts* package.[5) The xts format is very flexible with respect to the frequency and date format of the time series. Additionally, the Bundesbank's *rZISDB* package, which is used by Bundesbank analysts to load time series from the data base into R, uses the *xts* format as well. This allows a seamless interaction between loading and displaying the data.

3.2.4 bbktable

The function *bbktable()* can be used to create tables in the Bundesbank design. It accepts data.frames and matrices as input. Though the output is a table, the format is a plot. This has the disadvantage that the table cannot be changed after its creation. However, on the positive side, such plots can be flexibly included in any publications and documents, i.e. LaTex, Excel and PowerPoint. Table 5 (see appendix) summarises the arguments of this function.

3.2.5 bbkcolor

In general, there are 13 specific colours defined in the Corporate Design Manual, which are Bundesbank-blue, Bundesbank-grey as so-called brand colours, and dark grey, yellow, orange, brown, red, pink, violet, light blue, petrol, dark green and light green as design colours for visualisations. For most of the colours, the Corporate Design Manual specifies approved shades.[6)

The *bbkcolor()* function defines the colours using the hexadecimal system that is used by *ggplot2* and *bbkplot*. The user can chose colours and, if needed, the shade. Each colour has up to five shades. If less than five shades are defined in the Corporate Design Manual, the shades are repeated, e.g. in *bbkcolor()* the fourth and fifth shade of green are identical (see overview in appendix). The function allows the specification of multiple colours and shades and understands American and British spelling of the colors, e.g. grey and gray.

[5] Objects of class ts are transformed to xts using the tsbox package inside of the function.

[6] For the shades, corporate design defines the colours as RGB and CMYK colours. We translate these into hexadecimals also known as HTML colour codes. A detailed overview is given in Figure 9 in the Appendix.

3.2.6 save_plot

To export the plot, the *save_plot()* function can be used. The recommended output format are PNG and PDF (*device* = *"png"* or *device* = *"pdf"*). By default, the plot is exported as a PNG image to the current working directory. A PDF can be created by setting *device* = *"pdf"*. Internally, the *grDevices* function *cairo_pdf()* is used.

For all formats but PDF, *ggsave()* of the *ggplot2* package is used. Currently, it allows exporting plots as EPS, PS, TEX, JPEG, TIFF, BMP, SVG and WMF. The function *save_plot()* allows to specify a path argument, to save the plot in any specified folder and a file name argument.

4 Usage examples

This section provides examples for using the functions described above. Further, these examples also illustrate how several types of plots can be created using the *ggplot()* function of the *ggplot2* package and then easily converted to Bundesbank style using *bbkplot*. More and constantly updated examples can be found in the vignette of *bbkplot*.

For all examples, the *ggplot2* and *bbkplot* package need to be loaded as follows:

```
library(ggplot2)
library(bbkplot)
```

Generally, plots are created using *ggplot2*. Afterwards, the style is transformed according to the Corporate Design Manual using a call to functions in *bbkplot*. While these functions try to preserve all other settings defined in the original plot, some settings can be defined by the user both in *ggplot2* and in *bbkplot*. If for example, a title is defined in the original call to ggplot and in bbkplot, the title defined using bbkplot will have precedence.[7]

4.1 Scatterplot

Scatterplots are a simple visualisation of a bivariate relationship. Using dots or similar symbols the value pairs of two variables are displayed in a diagram with cartesian coordinates. Often these plots are augmented by lines that aim to clarify the relationship. These lines can be derived by standard statistical procedures such as linear regression, LOESS or splines.

Using the classic "Motor Trend Car Road Tests" (mtcars) dataset, the following code creates a simple scatterplot of the miles per gallon (mpg) and weight of 32 automobiles:

```
# Define the scatterplot
p <- ggplot(mtcars, aes(mpg, wt)) +
  geom_point(shape = 19) +
  aes(color = factor(cyl)) +
  labs(title = "A scatterplot")
```

Then, the *bbkplot()* function is used to convert the plot into the Bundesbank style (see figure 2) :

```
bbkplot(p)
```

As mentioned earlier, further customisation within the corporate design scope is possible. For instance, the following modifications to the above code allow us to draw the same plot with the legend positioned inside (see figure 3):

[7] This is also the case for subtitles, where a subtitle defined using the bbkplot will have precedence.

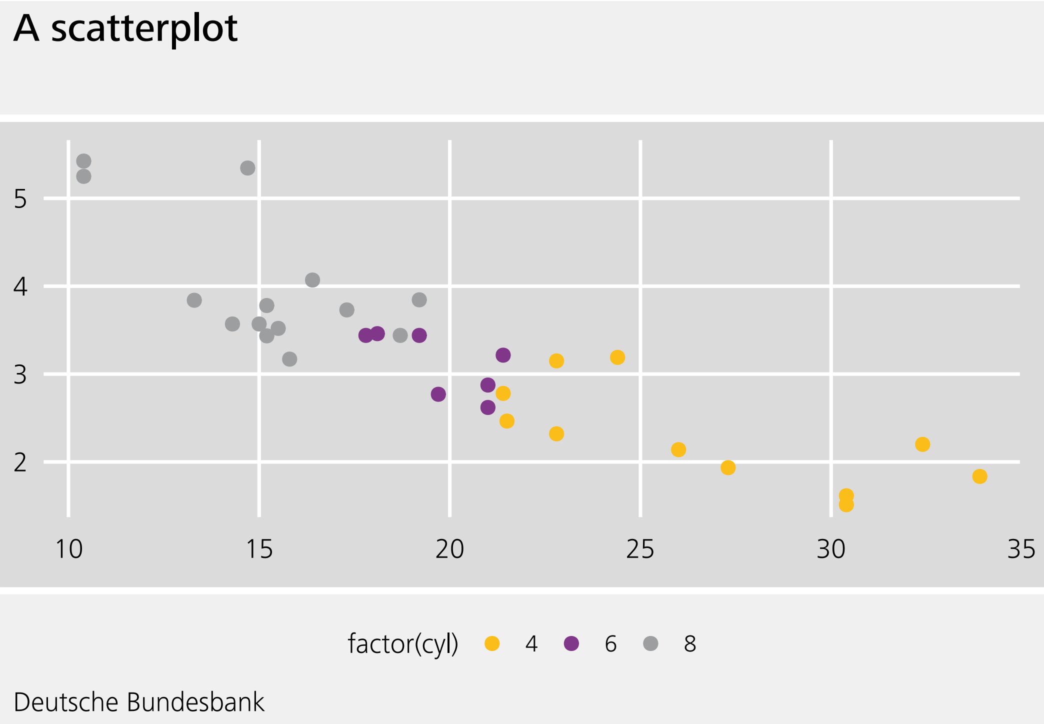


Figure 2: Scatterplot converted to bbkplot

```
p <- ggplot(mtcars, aes(mpg, wt)) +
  geom_point(aes(color = factor(cyl))) +
  labs(title = "Test") +
  theme(legend.position = c(.95, .95),
        legend.justification = c("right", "top"),
        legend.box.just = "right",
        legend.margin = margin(6, 6, 6, 6))


bbkplot(p, title = "A scatterplot with mtcars in R",
        subtitle = "Legend in plot area",
        legend_below = FALSE)
```

4.2 Histogram

A histogram is a visualisation of the frequency distribution of grouped data (Diaz-Bone, 2006). While the height of the bars in barplots correspond to the number of observations in each group, for histograms the area of each bar is equal to the frequency density. It is therefore often used to inspect the empirical distribution of continuous data. Sometimes parametric density estimates are added to histograms.

Histograms are easily created in ggplots. Here, we present an example of a histogram, where two subgroups are highlighted in colour. We draw data from two normal distributions with different

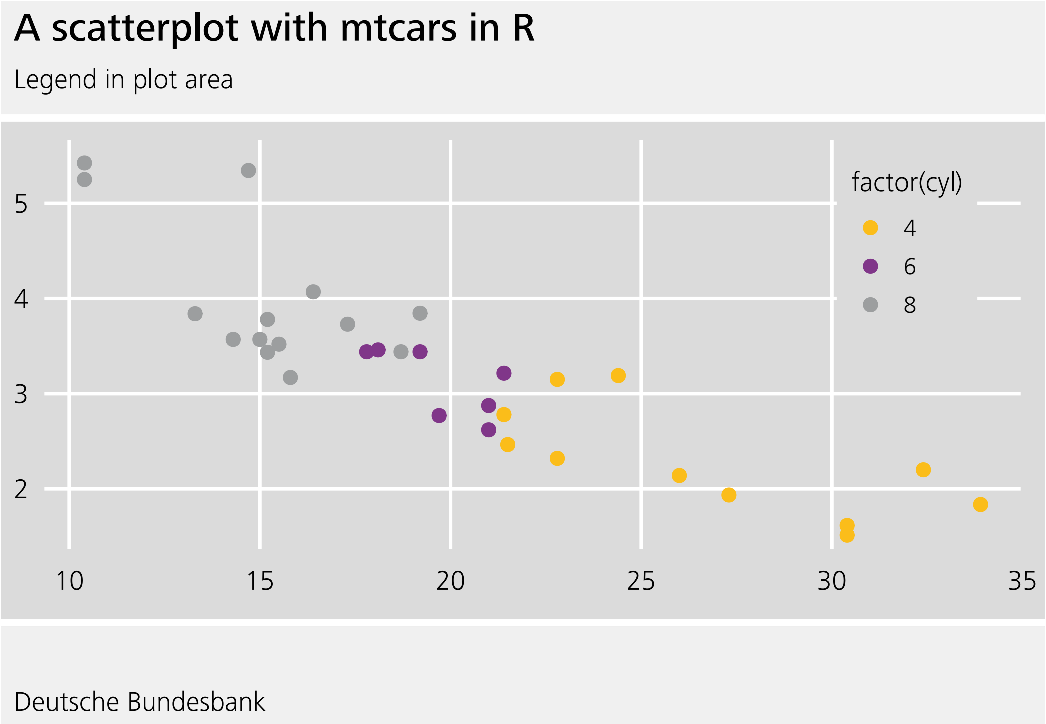


Figure 3: Scatterplot

mean and then plot these data using *ggplot()*:

```
set.seed(1234)

# Generate artificial data
df <- data.frame(
  Object_Type = factor(rep(c("A", "B"), each=200)),
  Weight = round(c(rnorm(200, mean=50, sd=2),
                   rnorm(200, mean=65, sd=2))))

# Create histogras with bins
p2 <- ggplot(df, aes(x = Weight, fill = Object_Type)) +
  geom_histogram(bins = 25 )  +
  theme(legend.position = c(.55, .9),
        legend.justification = c("right", "top"),
        legend.box.just = "right",
        legend.margin = margin(6, 6, 6, 6))
```

Again, the ggplot object can simply be transformed to the Bundesbank style using *bbkplot()*. Here, we set a title and a source using the latter function (see figure 4):

```
bbkplot(p2, title = "Histogram of generated data",
        source = "Data generated with calls to rnorm()",
        legend_below = F)
```

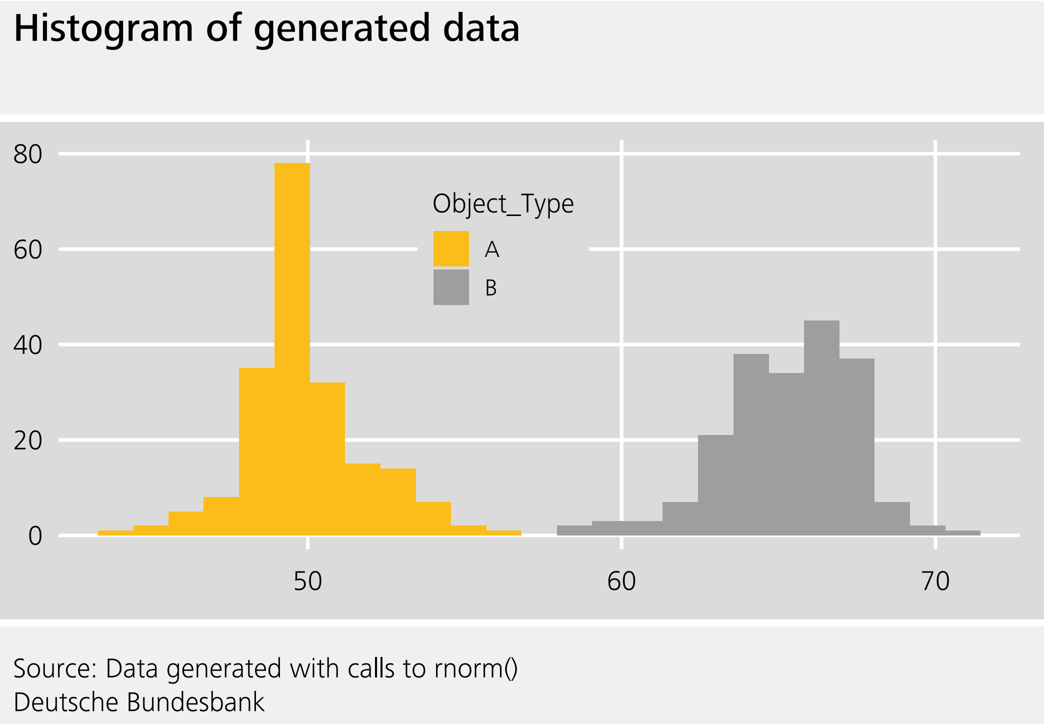


Figure 4: Histogram created using ggplot2

4.3 Boxplot

Boxplots are another option to represent the conditional or unconditional distribution of the data. They have been developed by Tukey (1977) as a simple yet meaningful descriptive representation of the data that even can be drawn by hand. Traditionally, a boxplot comprises the median as a line, a box stretching from the lower to the upper quartile, two whiskers that depict extremes and single outliers outside of the extremes. There exist numerous variations of this standard configuration, especially with respect to the whiskers (cf. Wickham and Stryjewski, 2011).

In the Bundesbank design, the mean is included alongside the median in the boxplot. The mean is shown as a line and the median as a filled green circle. The outliers are usually not depicted. The necessary transformations of the boxplot are computed inside of bbkplot. To create an example of a boxplot, we use the 'mtcars' data in R. The mileage per gallon (mpg) is broken down by the number of cylinders of the automobile (see figure 5). As before, the boxplot can afterwards be transformed into Bundesbank style using a single line of code.

```
p3 <- ggplot(mtcars,aes(x=factor(cyl),y=mpg)) +
  geom_boxplot() +
  theme(legend.position = "None") +
  labs(x = "cylinders", y = "mpg")

bbkplot(p3, title="Boxplot for mtcars data")
```

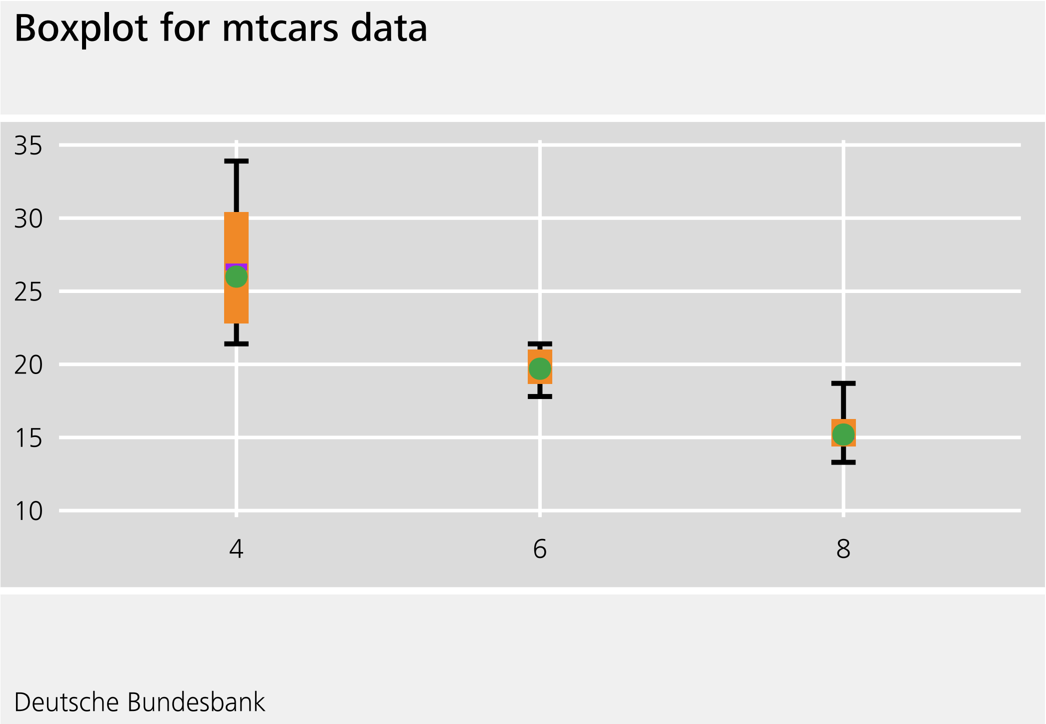


Figure 5: Boxplot in Bundesbank design

4.4 Map graphic

Heatmaps were originally developed by financial software designer Cormac Kinney in 1991, who used this colour-coded cluster analysis system to visualise financial market information (Akers, 2015). Each individual value of data is represented as a gradation of colour in such graphical presentations.

In our example below, We present a map graphic as a heatmap which represents the happiness index (Deutsche Post, 2019) across different German federal states. Maps are an effective way to represent data pertaining to geographical locations. Combining this with a heatmap allows us to highlight the distinct values across space as a continous spectrum thus identifying even small changes.

We use the GADM maps and spatial data for our purpose. This data can be downloaded using the *GADMTools* package in R and is treated as "simple features" object as implemented by the *sf* package in R. The *ggplot2* package allows the use of "simple features" as graphical layers. The following code follows our familiar outline and converts the ggplot with the map graphic into a Bundesbank style plot:

```r
# Add the happiness index to the GADM data
fll <- c(7.44, 7.27, 7.19, 7.18, 7.17, 7.31, 7.21, 7.21,
         7.27, 7.21, 6.93, 6.76, 6.87, 6.98, 6.92, 7.09)
happy_map <- ggplot(geo_data_final, aes(fill = fll)) +
  geom_sf() + xlab(label = "Latitude") +
  ylab(label = "Longitude") + labs(fill = "Happiness Index")
```

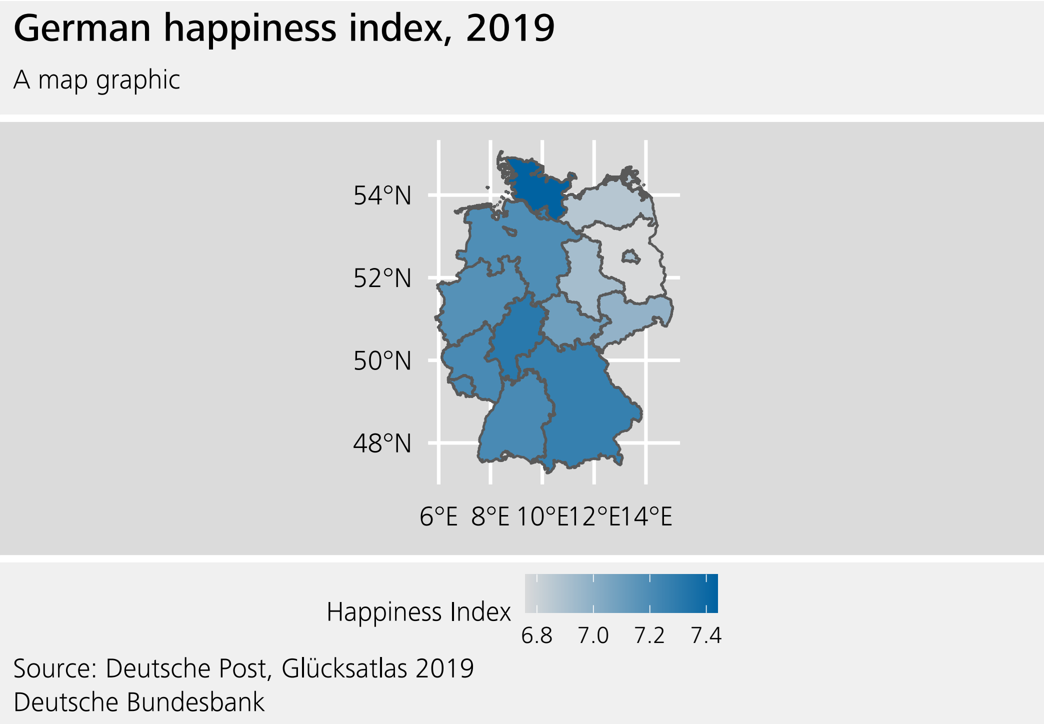


Figure 6: Line plot

```
# Call bbkplot function
# with parameters discrete and scale_y_continous set as FALSE

bbkplot(happy_map, scale_y_continuous = FALSE,
        source = "Deutsche Post, Glücksatlas 2019",
        title = "German happiness index, 2019", discrete = FALSE,
        subtitle = "A map graphic",  palette = "bichromatic")
```

4.5 Time series plot

As discussed above, a special function dedicated to plotting time series is included in the *bbkplot* package in order to facilitate an efficient visualisation of this data type. Using the packages within the Bundesbank environment, it is possible to directly load specified time series data from the Bundesbank database into R using the rZISDB package and then plot them in Bundesbank design:

```
tsplot(rZISDB::load(ids = c("BBNZ1.Q.DE.N.H.0000.A",
                          "BBNZ1.Q.DE.Y.H.0000.A"))["2011/"],
       title = "GDP Germany, chain-linked", subtitle = "2015=100",
       names = c("Original", "Seasonally Adjusted"))
```

```
## [1] "BBNZ1.Q.DE.N.H.0000.A has dimension factor  1"
## [1] "BBNZ1.Q.DE.Y.H.0000.A has dimension factor  1"
```

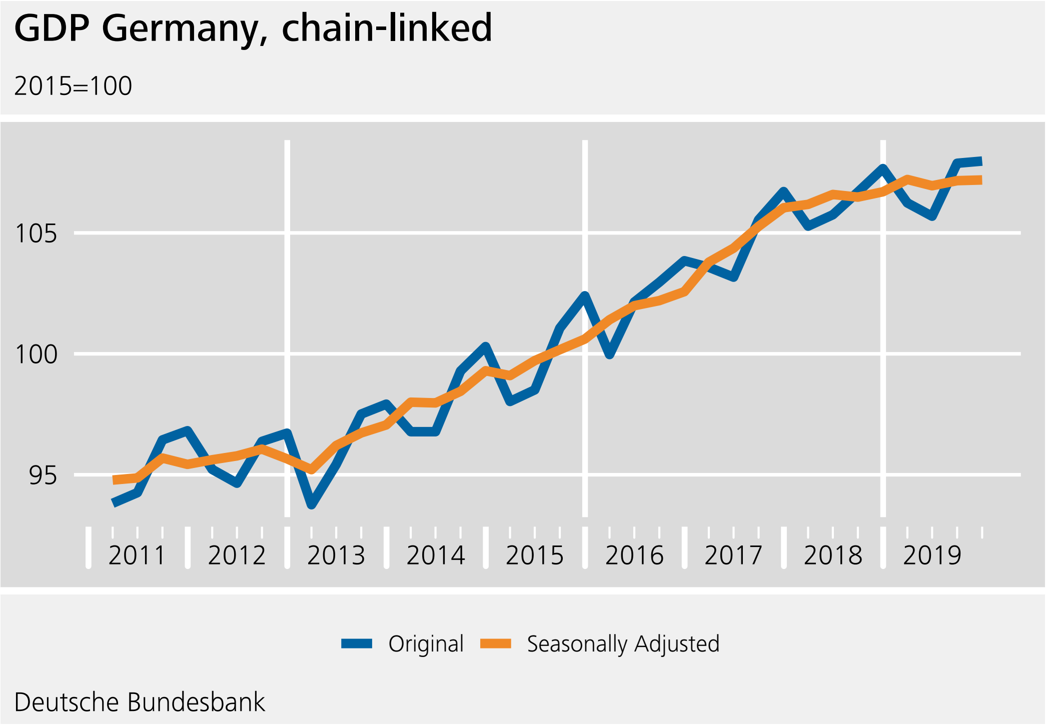


Figure 7: Time series plot

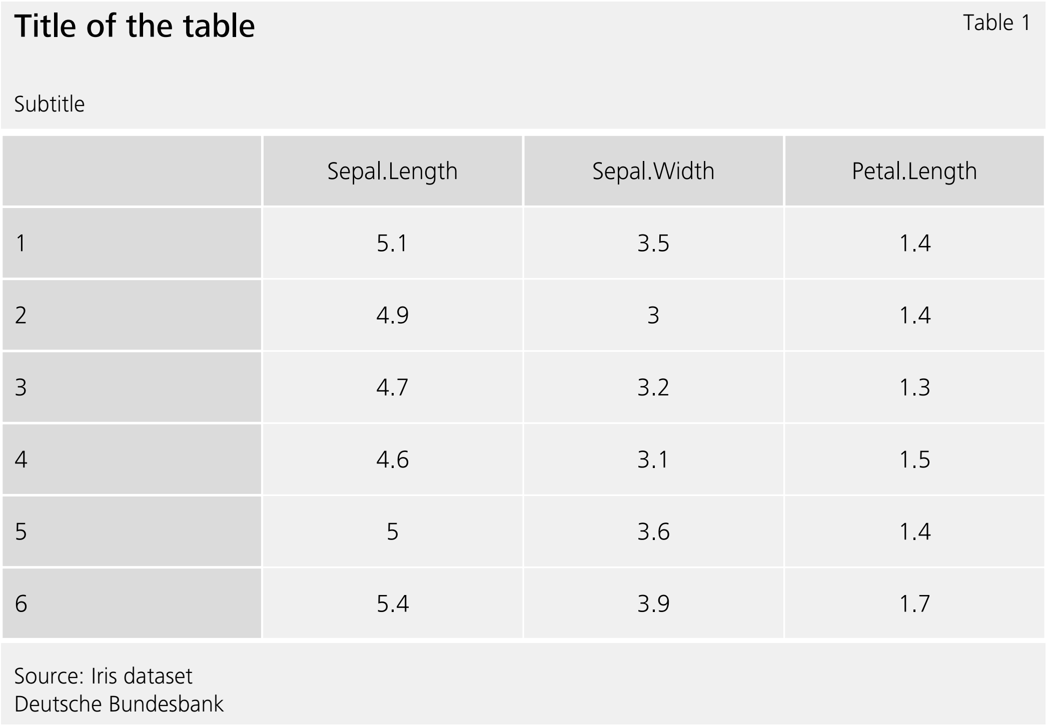
A single call to *tsplot()* that includes the time series object suffices to create a Bundesbank style plot. This can of course be augmented by including title, subtitle and specific names for the time series.

4.6 Tables

In addition to plots, the *bbkplot* package allows creating Bundesbank style tables using *bbktable()*. This function takes a pre-defined data set (such as a data.frame) as an input. The resulting table can be tweaked by adding a title, numbering or changing the text size (see figure 8):

```
d <- head(iris[,1:3])

bbktable(d,
         title = "Title of the table",
         footer_label = NULL,
         subtitle = "Subtitle",
         source = "Iris dataset",
         fontsize = 15,
         number = "Table 1")
```



Title of the table Table 1

Subtitle

	Sepal.Length	Sepal.Width	Petal.Length
1	5.1	3.5	1.4
2	4.9	3	1.4
3	4.7	3.2	1.3
4	4.6	3.1	1.5
5	5	3.6	1.4
6	5.4	3.9	1.7

Source: Iris dataset
Deutsche Bundesbank

Figure 8: Table with adapted fontsize parameter

5 Conclusion

The past decade has witnessed a growth in the amount of data being collected and its use for analysis. As the digitisation trend continues, there will be more data-driven analysis, decision-making, and dissemination of results. Along with this comes an increasing number of data-savvy employees carrying out data analysis and using programming languages such as R.

There has been a rise in evidence based policy-making in different areas. It may be defined as an attempt to base policy decisions on scientific and empirical evidence, including impact studies, cost-benefit analysis, program evaluation and academic research in general. Especially in economics, the availability of new data has raised challenges and opportunities stimulating numerous developments (Heckman, 2001). Thus, the need to communicate these data-driven decisions has also gained precedence.

Adherence to corporate design lends the authenticity and trustworthiness of a central bank to its communication with the public. To add to this, the spread of information on digital platforms has surpassed the traditional platforms and this means that the time lag from production to publication has also been reduced. This adds to the time pressure to be able to produce results and communicate them.

Our package *bbkplot* is developed to meet this contingency keeping in mind the tech- and data-savvy employees who can use it at scale to produce graphics for their presentations and communication. It further enables them to produce interactive graphics including network and map graphics for digital presentations, an accomplishment that was not possible in the past.

In a modern, data-friendly paradigm, it makes sense to develop a package like *bbkplot* on an open source platform. This enables a large number of analysts to produce graphics that complement their analysis and are fit to be included in reports and presentations on the go. The clear benefit is access to means of producing quality graphs directly by the responsible expert carrying out the analysis at no extra cost to the organisation.

An easy way to produce these visualisations promotes economical use of resources and is designed to allow users not expertly experienced in the use of R to create graphics in corporate design. In addition to the development of the package, we supplement researchers and analysts within Bundesbank with training in production of graphics using our package to promote its use and the use of R for data analysis in general.

Easy and low-barrier access to high-quality visualisations is meaningful from an organisational point of view as it will facilitate better adherence to corporate design. By helping to consequently including corporate design in all communication, we support brand communication and spread of messages from a trusted brand.

References

Allen, P. (1996). User Attitudes Toward Corporate Style Guides: A Survey. Technical Communication, 43 (3), 237-243.

Arnold, J.B, Daroczi, G., Werth, B., Weitzner, B., Kunst, J., Auguie, B. Rudis, B., Wickham, H., Talbot, J., and J. London (2020). ggthemes: Extra Themes, Scales and Geoms for ggplot2. https://CRAN.R-project.org/package=ggthemes

Bion, R., Chang, R., and J. Goodman (2017). How R helps Airbnb make the most of its data. The American Statistician, 72(1), 46-52.

Deutsche Bundesbank (2011). Corporate Design Manual. October 2011.

Diaz-Bone, R. (2018). Statistik für Soziologen. Konstanz. UTB Verlag.

Global Administrative Areas (2012). GADM database of Global Administrative Areas, version 2.0. www.gadm.org.

Heckman, J.J. (2001). Micro data, heterogeneity, and the evaluation of public policy: Nobel lecture. Journal of Political Economy, 109 (4), 673-748.

Jackson, S. (2018). Creating corporate colour palettes for ggplot2. https://drsimonj.svbtle.com/creating-corporate-colour-palettes-for-ggplot2

Klein M. and S. Wagner (2018). ggCorpIdent: INWTlab wrapper for ggplot2 to create plots matching a corporate identity. INWT Statistics GmbH. https://github.com/INWTlab/ggCorpIdent

Neth, H. and N. Gradwdohl (2019). unikn: Graphical elements of the University of Konstanz's corporate design. https://CRAN.R-project.org/package=unikn

Grimm, R. and B. Raffelhüschen (2019). Glücksatlas. Deutsche Post AG

Stylianou, N., Guibourg, C., Dahlgreen, W., Cuffe, R., Calver, T., and R. Mpini (2018). bbplot: Making ggplot graphics in BBC news style. British Broadcasting Company.

Tuckey, J.W. (1977). Exploratory Data Analysis. Addison-Wesley.

Akers, W. (2015). Visual resource monitoring for complex multi-project environments. Int. J. System of Systems Engineering, Vol. 6, Nos. 1/2, pp.112–126

Wickham, H, Chang, W., Henry, L., Takahashi, K., Wilke, C., Woo, K., and H. Yutani (2019). ggplot2: Create elegant data visualisations using the grammar of graphics.
https://CRAN.R-project.org/package=ggplot2

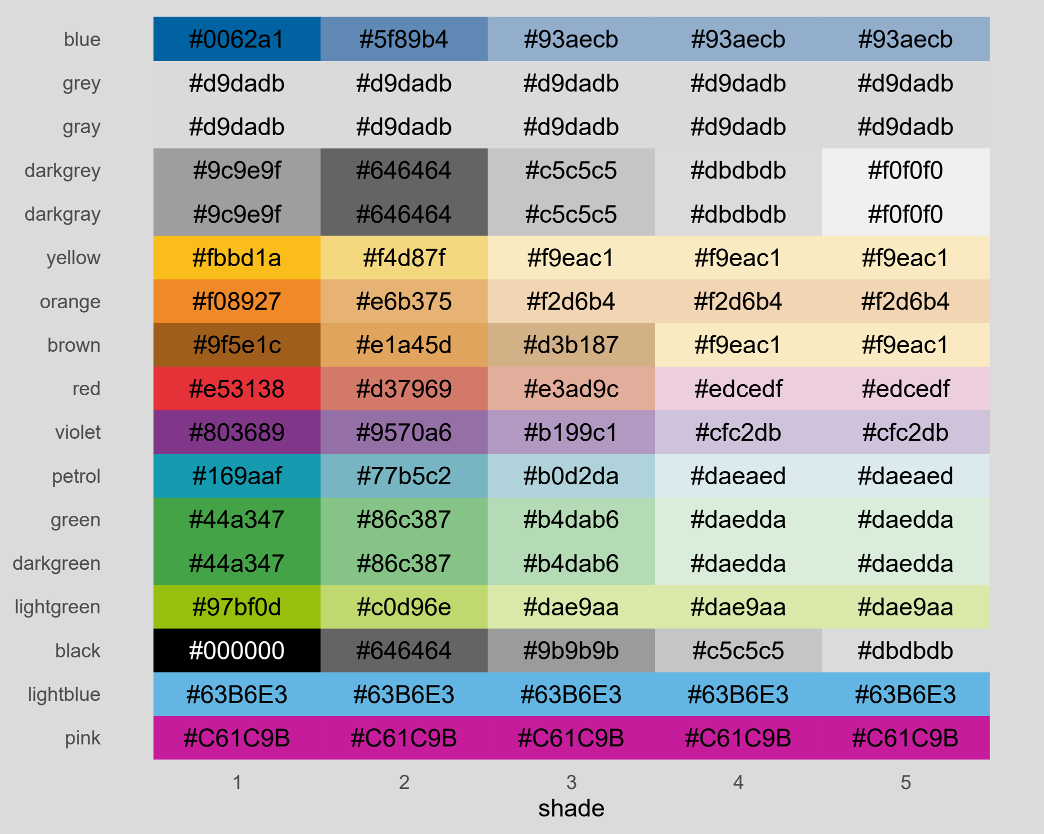
Wickham, H, and L. Stryjewski (2011). 40 years of boxplots. American Statistician.

A Appendix

Deutsche Bundesbank Corporate Design Colors

Figure 9

bbkcolor(): color name and desired shade as input, exact hexadecimal codes as output



	1	2	3	4	5
blue	#0062a1	#5f89b4	#93aecb	#93aecb	#93aecb
grey	#d9dadb	#d9dadb	#d9dadb	#d9dadb	#d9dadb
gray	#d9dadb	#d9dadb	#d9dadb	#d9dadb	#d9dadb
darkgrey	#9c9e9f	#646464	#c5c5c5	#dbdbdb	#f0f0f0
darkgray	#9c9e9f	#646464	#c5c5c5	#dbdbdb	#f0f0f0
yellow	#fbbd1a	#f4d87f	#f9eac1	#f9eac1	#f9eac1
orange	#f08927	#e6b375	#f2d6b4	#f2d6b4	#f2d6b4
brown	#9f5e1c	#e1a45d	#d3b187	#f9eac1	#f9eac1
red	#e53138	#d37969	#e3ad9c	#edcedf	#edcedf
violet	#803689	#9570a6	#b199c1	#cfc2db	#cfc2db
petrol	#169aaf	#77b5c2	#b0d2da	#daeaed	#daeaed
green	#44a347	#86c387	#b4dab6	#daedda	#daedda
darkgreen	#44a347	#86c387	#b4dab6	#daedda	#daedda
lightgreen	#97bf0d	#c0d96e	#dae9aa	#dae9aa	#dae9aa
black	#000000	#646464	#9b9b9b	#c5c5c5	#dbdbdb
lightblue	#63B6E3	#63B6E3	#63B6E3	#63B6E3	#63B6E3
pink	#C61C9B	#C61C9B	#C61C9B	#C61C9B	#C61C9B

Source: Corporate Design Manual, own conversion
Deutsche Bundesbank

Table 1: Arguments for bbkplot( )

	Argument	Definition
1	object	a ggplot object to be put into bbk design. Must be specified
2	title	the title of the plot, a string. Default=NULL
3	main	alternative argument name to pass the title of the plot, a string. Default=NULL. When both title and main are not NULL, title will be used
4	subtitle	the subtitle of the plot, a string. Default=NULL
5	submain	subtitle of the plot, a string. Default=NULL
6	number	an optional number for the plot (as a string), in case multiple plots are produced and should be numbered (in the upper right corner). Default=NULL
7	footer_label	label of the plot, a string, e.g. usefull for adding datetime or the name of the file itself to the plot. Default=NULL
8	color	the color of the plot, a string (one of c(”blue”, ”grey”, ”darkgrey”, ”yellow”, ”orange”, ”brown”, ”red”, ”violet”, ”petrol”, ”green”, ”darkgreen”, ”lightgreen”, ”black”), i.e. the bbk corporate style guide colors). Default=NULL
9	fill	the color to fill areas
10	source	the source of the plot, a string
11	include_source	should the text component ”Source:” be included
12	legend_below	boolean, whether the legend should be extracted and plotted below the plot area itself. If FALSE, the user has to adjust legend position within the plotting area. Both are valid according to the Bundesbank style guide. Default=TRUE
13	add_theme_bbk	whether to add the bbk theme to the plotting surface. Default=TRUE. Useful to be set to FALSE in non-standard plots, if different theme settings were manually set before, this option may override them)
14	fontsize	the size of the font to be used for the title. Everything else is scaled to 0.7*fontsize according to the styleguide. Default=18
15	discrete	boolean, should color values used for plotting be discrete (or interpolated). Default=TRUE
16	palette	the color palette from the range of Bundesbank design colors (see details). Default=”main”. One of c(”main”, ”main_raster_1”, ”main_raster_2”, ”main_raster_3”, ”main_raster_4”, mixed”, ”mixed_raster_1”, ”mixed_raster_2”, hot”, ”cold”, ”bichromatic”, ”blackwhite”, ”random”)
17	axis_title	Whether to draw the axis titles. Default=TRUE
18	legend_title	Whether to draw the legend titles. Default=TRUE
19	distance_subtitle	The size of the space between title and subtitle. Default=1. May be useful to increase in case of multiline titles
20	plain_background	Is it a plain background?
21	y_title_direction	Defines the position of the y-axis title. Default=”vertical”. Other options are ”top”, to align the title on the top of the plotting area (horizontal) and ”subtitle” to align the title into the header
22	log_y	What kind of logarithmic scaling the y axis should have. log_y_scale takes the value of the base of the logarithm. It will usually be a value of c(1/2, 2, 10,exp(1)) but can take other numeric values. default=NULL (no logarithmic scale)

Table 2: Arguments for bbkplot( ) continued

	Argument	Definition
1	n_breaks	number of grid labels, numeric value, default=5
2	scale_y_position	position of the y axis scale, default = ”left”, can be set to ”right” e.g. to arrange multiple plots
3	horizontal_line	whether to add a highlighted horizontal line, can be any numeric (usually at zero), default=0
4	expand_x_axis	whether the x -axis should be expanded (the largest value by 10 percent) in order to open the grid, default=FALSE
5	legend_height	height of the legend (if plotted below the plot area), default = 1, increase legend_height to increase vertical space for legend
6	scale_y_continuous	whether the y_scale is continuous (TRUE) or discrete (FALSE), default = TRUE
7	secondary_axis	creares a secondary axis based on a one to one transformation of the primary axis
8	secondary_axis_name	name of the secondary axis
9	...	additional arguments passed to ggplot2::theme()

Table 3: Arguments for theme_bbk( )

	Argument	Definition
1	fontsize	set the fontsize for axis title (lables, legend, etc. are scaled with a factor of 0.7 of fontsize). Default=18.
2	discrete	Wether color values used for plotting should be discrete. Default=TRUE.
3	palette	the color palette from the range of Bundesbank design colors (one of c(”main”, ”main_raster_1”, ”main_raster_2”, ”main_raster_3”, ”main_raster_4”, mixed”, ”mixed_raster_1”, ”mixed_raster_2”, hot”, ”cold”, ”bichromatic”, ”blackwhite”, ”random”))
4	axis_title	Whether to draw the axis titles. Default=TRUE.
5	color	Color parameter to select any of the Bundesbank corporate design colors.
6	fill	Fill parameter to select any of the Bundesbank corporate design colors.
7	plain_background	Is it a plain background?
8	y_title_direction	Defines the position of the y-axis title. Default=”vertical”. Other options are ”top”, to align the title on the top of the plotting area (horizontal) and ”subtitle” to align the title into the header
9	log_y	What kind of logarithmic scaling the y axis should have. log_y_scale takes the value of the base of the logarithm. It will usually be a value of c(1/2, 2, 10,exp(1)) but can take other numeric values. default=NULL (no logarithmic scale)
10	n_breaks	number of grid labels, numeric value, default=5
11	scale_y_position	position of the y axis scale, default = ”left”, can be set to ”right” e.g. to arrange multiple plots
12	horizontal_line	whether to add a highlighted horizontal line, can be any numeric (usually at zero), default=0
13	expand_x_axis	whether the x -axis should be expanded (the largest value by 10 percent) in order to open the grid, default=FALSE
14	scale_y_continuous	whether the y_scale is continuous (TRUE) or discrete (FALSE), default = TRUE
15	secondary_axis	creates a secondary axis based on a one to one transformation of the primary axis.
16	secondary_axis_name	name of the secondary axis
17	...	additional arguments passed to ggplot2::theme()

Table 4: Arguments for tsplot( )

	Argument	Definition
1	series	one or multiple time series
2	transform	one of ”none”,”diff”, ”change” (can be abbreviated)
3	type	either ”bar”, ”bar2” or ”line”
4	years	number of years to include
5	names	change names of series
6	scale	by what factor should data be scaled.
7	color	color of the series
8	main	title of the plot
9	submain	subtitle of the plot
10	title	title of the plot
11	subtitle	subtitle of the plot
12	linesize	scale the size of the lines
13	linetype	type of line
14	axis_title	Whether to draw the axis titles. Default=FALSE.
15	legend_position	Position of legend inside of plot
16	legend_direction	”horizontal” or ”vertical”
17	date_breaks	distance between labels (see examples)
18	vlines	vertical lines to be drawn. Either a date or an integer
19	max_vlines	maximum number of vertical lines
20	date_labels	format of the date label for x-axis
21	all_years	Should all years be displayed on x-axis (only for longer series)
22	between	move date labels between years, dominates all_years
23	between_5y	move date labels between 5 years
24	ylim	Change limits of y-axis
25	source	the source of the plot, a string
26	add_theme_bbk	whether to add the bbk theme to the plotting surface. Default=TRUE. Useful to be set to FALSE in non-standard plots, if different theme settings were manually set before, this option may override them).
27	log_y	What kind of logarithmic scaling the y axis should have. log_y_scale takes the value of the base of the logarithm. It will usually be a value of c(1/2, 2, 10,exp(1)) but can take other numeric values. default=NULL (no logarithmic scale)
28	n_breaks	number of grid labels, numeric value, default=5
29	expand_x_axis	whether the x -axis should be expanded (the largest value by 10 percent) in order to open the grid, default=FALSE
30	secondary_axis	creates a secondary axis based on a one to one transformation of the primary axis
31	secondary_axis_name	name of the secondary axis
32	week_of_year	should x axis be week of year
33	...	other settings for bbkplot

Table 5: Arguments for bbktable( )

	Argument	Definition
1	object	a dataframe object to be wrapped with header, footer, etc..
2	title	the title of the table, a string
3	main	alternative argument name to pass the title of the table, a string. When both title and main are not NULL, title will be used
4	subtitle	the title of the table, a string
5	submain	subtitle of the plot, a string. Default=NULL
6	number	an optional number for the table (as a string), in case multiple tables are produced and should be numbered,
7	source	the source of the table, a string
8	fontsize	the size of the font to be used for the title. Everything else is scaled to 0.7*fontsize according to the styleguide
9	footer_label	label of the plot, a string, e.g. usefull for adding datetime or the name of the file itself to the plot. Default=NULL
10	include_source	should the text component ”Source:” be included
11	index_col	argument for setting the index column on/off
12	index_text	the text of the index column, a string.
13	col_widths	argument for manipulating the column widths. Needs an inputvector, which specifys the relative width of each column. Sum of elements must be 1.
14	use_rownames	argument of logical nature for using the row names. Default value is TRUE
15	...	additional settings for theme_bbk()