

Drawing Statistical Charts

The statcharts module

Tamara Kocurová, Adriana Kašparová, Tomáš Hála

This work focuses on the development of a module for creating statistical charts within ConT_EXt. The charts supported by the module have been chosen based on an analysis of those found in commonly used spreadsheets.

The main part of the work present examples of each chart type, showing the data and parameter values used to create them.

Introduction

Data can be represented in many ways, for example as text, tables, or charts – the latter being the simplest and easiest to understand.

There are many well-known and widely-used tools for chart creation, and a brief review is given in the next section.

There is no user-friendly way to draw charts in ConT_EXt (as it supports graphics but not charts) so users have two options: to insert a chart generated from another program in an image format, or to write their own code in some graphical language.

Therefore, the aim of this work was to create a module which would allow users to generate charts in ConT_EXt easily while maintaining the highest typographic quality.

The main part of the work is based on Tamara’s and Adriana’s successfully defended bachelor theses (Kocurová, 2020; Kašparová, 2020).

Brief review of graphing tools

While the programming language Lua is well known, quite simple and well documented (Lua.org, 2024), the situation with drawing tools is somewhat less clear.

The main source of information is undoubtedly the MetaPost manual by Hobby (2018) which describes more complex diagrams and goes beyond basic geometric figures. It also describes an extension to the MetaPost language implemented in the form of a package for drawing graphs. This allows the plot data to be read from a file consisting of multiple rows of pairs of values. The examples given include graphs showing just data points or with connecting lines which can be straight or smoothed and solid or dotted, with different line styles for different data series. The package can also fill the area under the line/lines which results in graph looking like the area chart.

Hagen (2017) explains a wide range of techniques for drawing not only simple elements but also more complicated figures (using MetaPost). Besides the solutions provided by Hobby (2019), Hagen (2017) adds how to transform (scale, shift and rotate) the shapes and how to define their colours. Filling shapes (closed paths) with any colour is discussed also by Henderson and Hening (2013).

Whereas Hobby (2019 etc.) does not specify any T_EX interface, Hagen (2017) enhances the code by integrating it with ConT_EXt. In order to use MetaPost code in ConT_EXt, he wraps it with start/stop commands (`\startMPcode`, `\startMPpage`, `\startMPdefinitions`, `\startMPinitialization`, etc.).

Besides the very simple vector graphics tool called `picture`, plenty of other packages have been written for L^AT_EX.

Morawski (2005) also used MetaPost to show how to plot various graphs. He used almost the same commands as Hobby (2019) but he also enhanced the code so that it is a valid L^AT_EX document. For that, he used the `latexMP` package.

For producing bar charts, there is a L^AT_EX package `bar` written by Joachim Bleser and Edmund Lang (Mittelbach and Goossens, 2004).

Casidy (2013) states that, for graphing with a coordinate system, the L^AT_EX environment `tikzpicture` can be used. Kottwitz (2015) recommends combining the `pgf-pie` package with `tikzpicture` for drawing pie charts.

While TikZ (written by Till Tantau) is a package developed for L^AT_EX (Crémer, 2011), it also – as a MetaPost alternative – supports ConT_EXt (ConT_EXt garden, 2019).

For drawing pie charts directly with MetaPost, Morawski (2002) states that the `piechartMP` package can be used. Integration with a document is shown for L^AT_EX, but it can probably be used within other T_EX engines.

Another package – `bpolynomial`, introduced by Hennig (2007) – is for plotting graphs in MetaPost and provides macros to calculate Bézier curves.

A different approach to using MetaPost for creating column charts, combined with the Lua lan-

guage for data processing operations, was shown by Hála (2019).

Other than these, there are programming languages which have robust graphics ability. For example, the Pic language (a language for specifying diagrams) provides shapes such as boxes, lines, arrows, circles, etc. (Kernighan, 1991). Tišnovský (2020) states that the Go language can be used for working with data (for data preparation and transmission of the values) and then a JavaScript graphing library such as `plotly` is used to create the charts. PostScript is also an option to create graphical shapes (Adobe Systems, 1999) including charts.

Material and methods

The above, albeit brief, overview of graph drawing tools shows a number of possibilities and different approaches.

However, they only provide solutions for specific issues encountered by the author in their work. No one, either in ConT_EXt or in L^AT_EX, has created a comprehensive tool that allows users to draw multiple types of graphs and parameterize them in a uniform way.

To be consistent with the ConT_EXt ethos and its other modules, it was clear that Lua and MetaPost had to be used for the implementation of our module.

Because of the very large number of different types of graphs in various scientific fields, we decided for the basic version of our module to restrict the number of chart types to those offered by the spreadsheets Calc and Excel.

The charts offered in these two spreadsheets were analyzed to get a list of types and subtypes, as well as their option settings.

contextgroup > context meeting 2024

A number of cited sources also provided inspiration on how to construct our module, both on how it should be structured, as well as MetaPost language examples.

- `\linechart`,
- `\piechart`,
- `\radarchart`,
- `\scatterchart`, and
- `\stockchart`

with only three parameters, e.g.:

Structure of the module

The module `t-statistical-charts` consists of three files:

- `t-statistical-charts.lua`
- `t-statistical-charts.mkiv`
- `t-readdata.lua`

The first file, `t-statistical-charts.lua`, contains the main chart drawing code.

The file `t-statistical-charts.mkiv` contains the definitions of the ConTeXt commands used to draw the charts.

Finally, the third file `t-readdata.lua`, provides data handling routines.

The main command to generate a chart is `\chart`:

```
\chart [1.] [2.] [3.] [4.]  
1 CHART_TYPE  
2 CHART_SUBTYPE  
3 SETTING_PARAMETERS  
4 DATA_INPUT
```

There are also individual versions of the above general command which are specific to each chart type:

- `\areachart`,
- `\barchart`,
- `\bubblechart`,
- `\columnchart`,

```
\areachart [1.] [2.] [3.]  
1 CHART_SUBTYPE  
2 inherits: \chart [#3]  
3 inherits: \chart [#4]
```

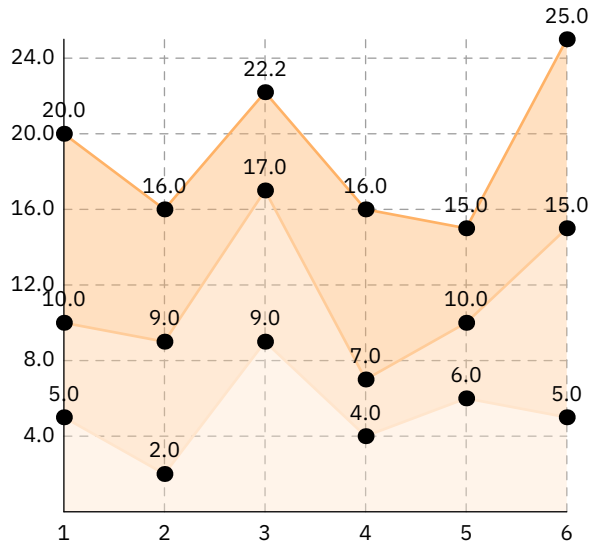
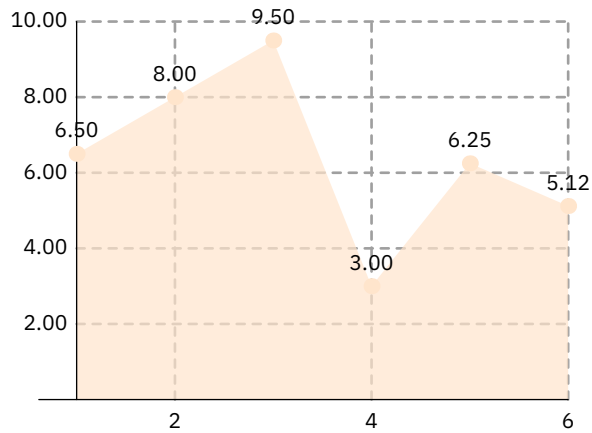
For a better understanding of the module, a manual *Drawing statistical charts* (Kocurová, Kašparová and Hála, 2020) has been published. It contains explanations of all parameter settings, data input options and examples of all available types and their subtypes. The continuously updated version of the manual is available at www.thala.cz/statcharts or via ConTeXt's module installer¹.

Charts

This section gives examples showing how all the implemented graph types and subtypes look. For clarity, we also include some examples of the code from which the graphs were obtained. The complete text of the commands including parameters can be found in the documentation.

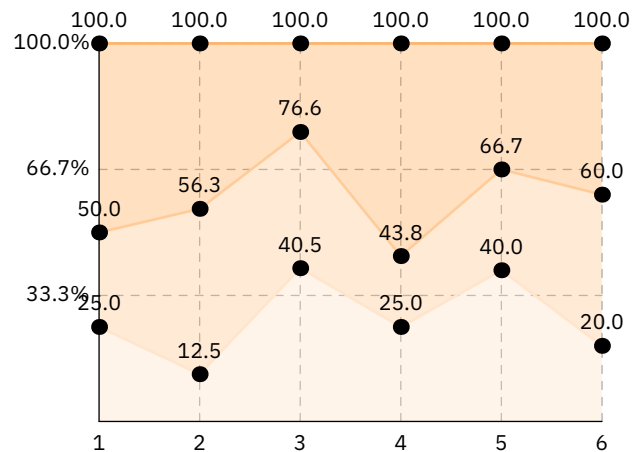
¹ `mtxrun --script install-modules install statistical-charts`

Area charts: basic, stacked, 100% stacked



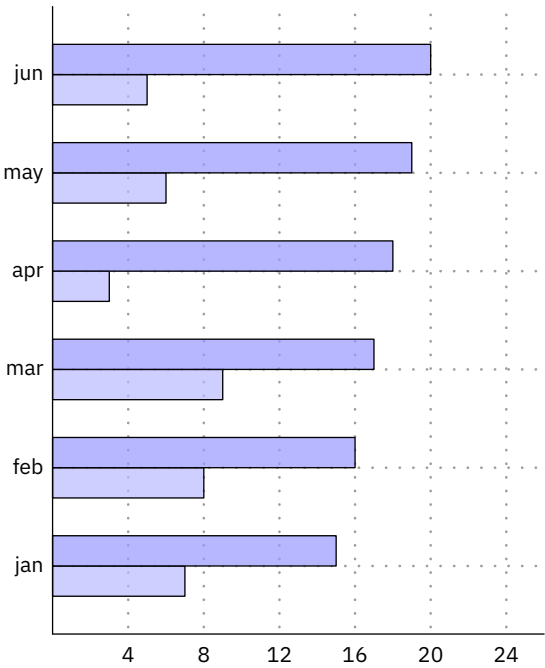
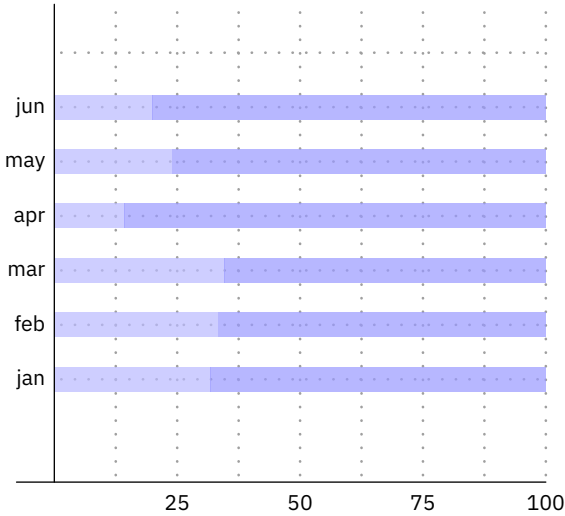
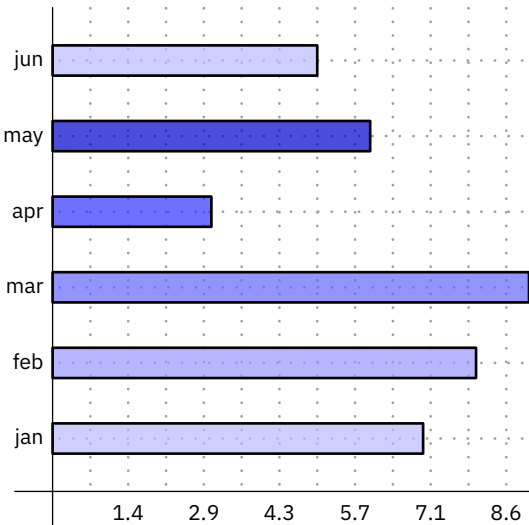
```
\chart[area][stacked][
  xlength=, ylength=4,
  xscale=1.4, yscale=.25,
  xaxislabels=yes, yaxislabels=yes,
  xaxislabelsnoval=yes, ylabelscount=3,
  unit=cm, distance=.95,
```

```
grid=on, gridcolor=middlegray,
gridline=dashed, gridlinewidth=0.5,
xgridlinesdensity=1,
ygridlinesdensity=1,
xgridlines=yes, ygridlines=yes,
xgridlinesnoval=yes, fillcolor=gray
,
fillpalette=orangepalette,
filltransparency=0.4,
line=yes, linecolor=black,
linepalette=orangepalette,
linewidth=1,
dots=yes, dotscolor=black,
dotpalette=, dotswidth=6,
dotslabels=yes,
equaldecimals=yes, decimals=1,
][method=struct,
data={ {5,2,9,4,6,5}, {5,7,8,3,4,10},
{10,7,5.2,9,5,10} }
```

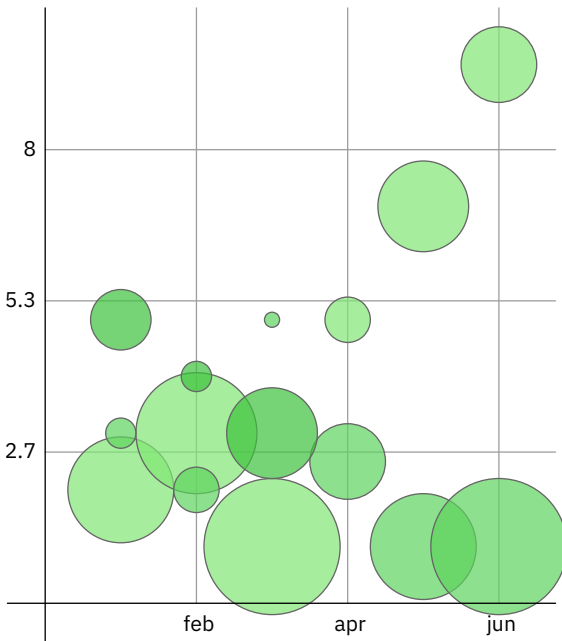


contextgroup > context meeting 2024

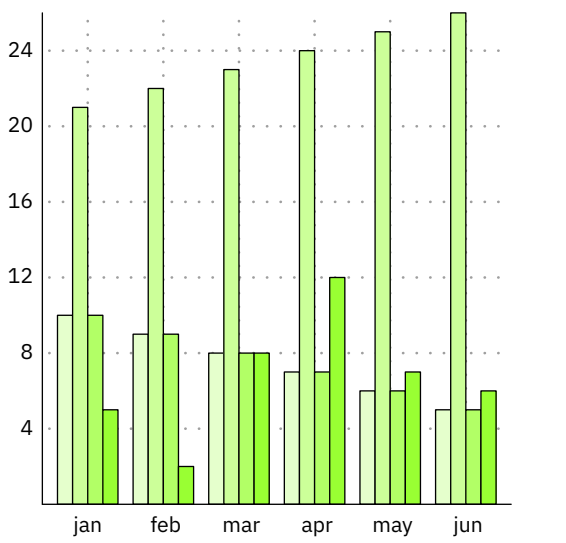
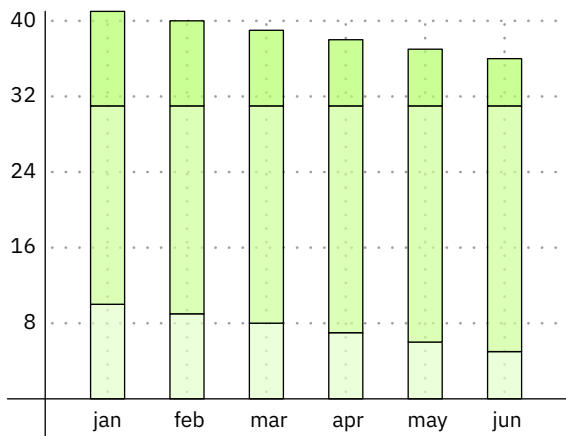
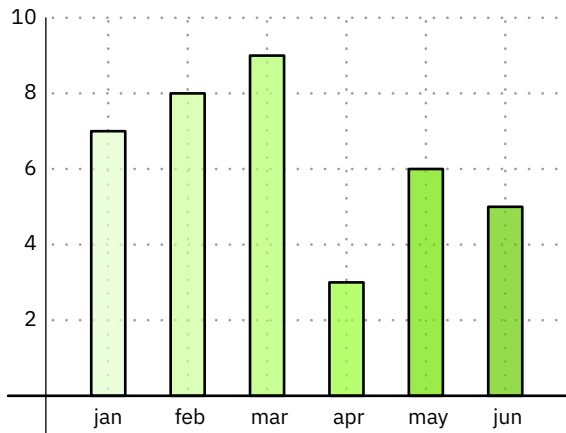
Bar charts:
basic, clustered, stacked, 100% stacked



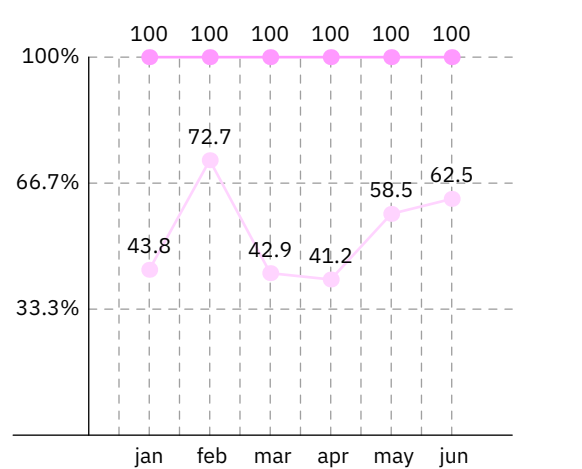
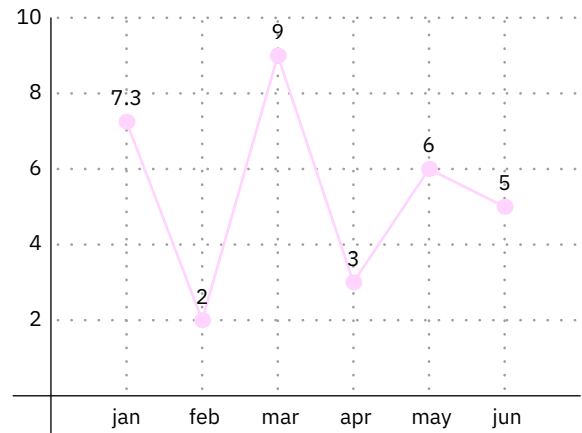
Bubble chart



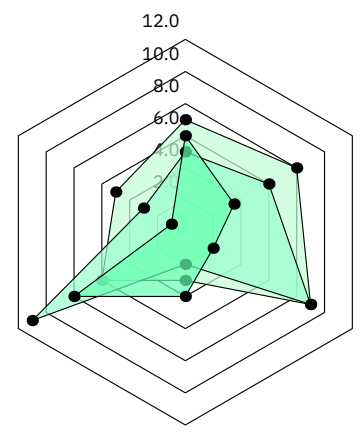
Column charts:
basic, clustered, stacked, 100% stacked



Line charts: basic, stacked

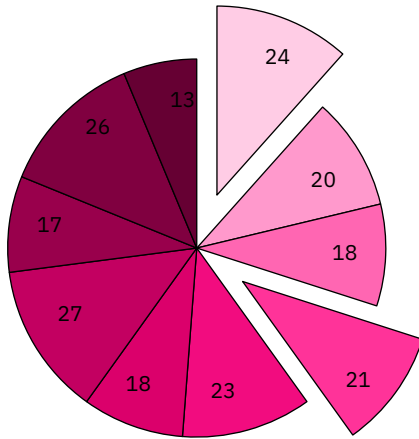


Radar chart



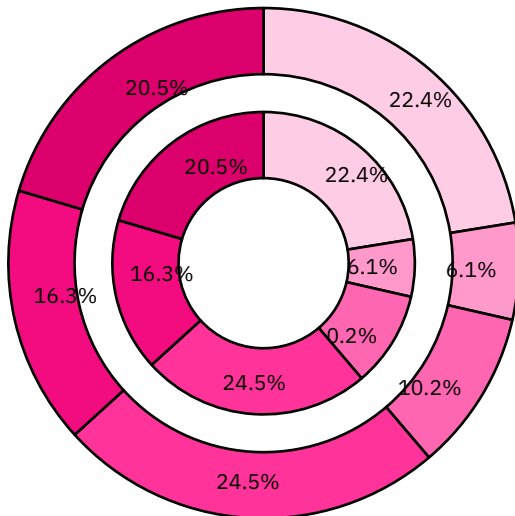
contextgroup > context meeting 2024

Pie charts: pie, doughnut

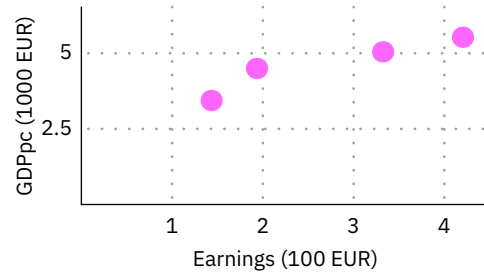


```
\def\timespentdata{%
  24,20,18,21,23,18,27,17,26,13}

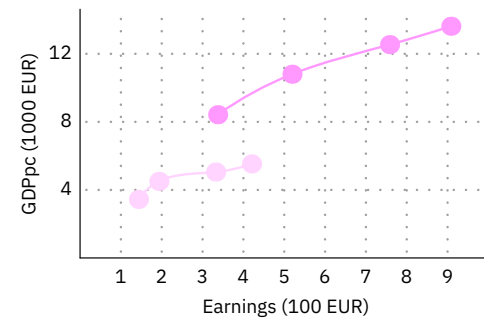
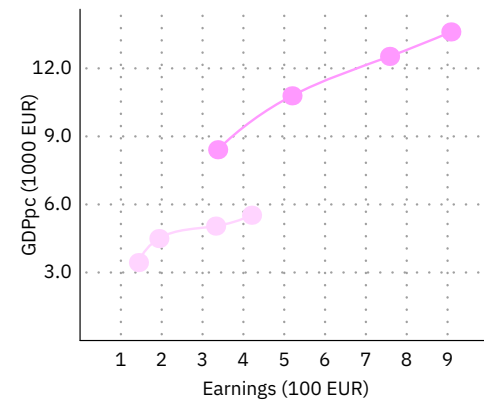
\chart[pie][basic][
  diameter=5, labelposition=in,
  fillpalette=pinkpalette,
  coef=.1, distanceout=.3,
  out={1,4},
][data={\timespentdata}]
```



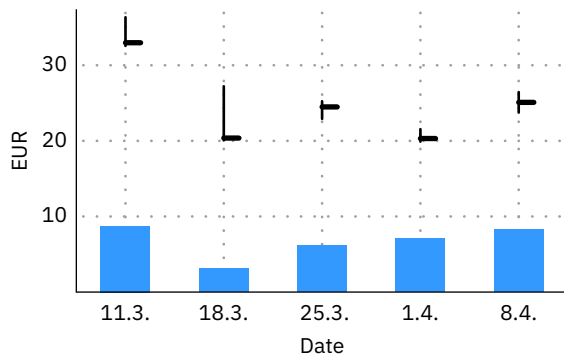
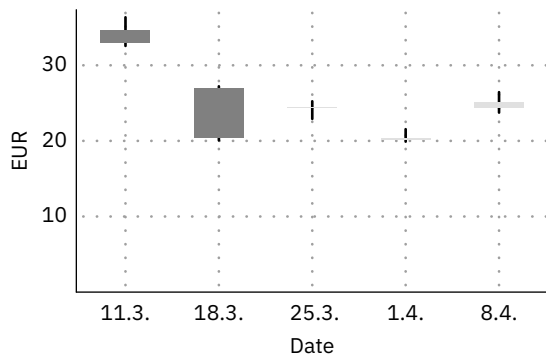
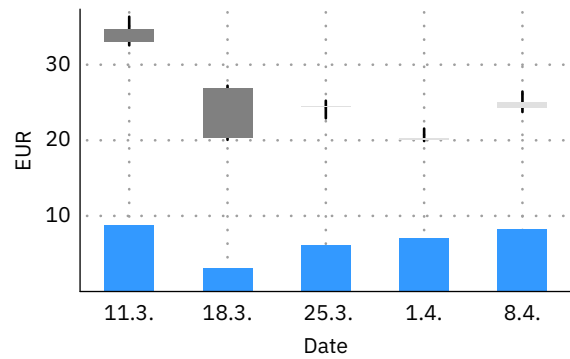
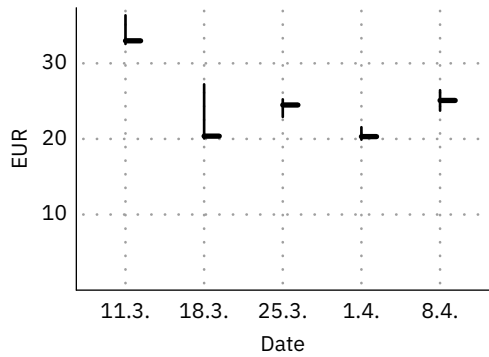
Scatter charts:
basic, with smooth lines, with straight lines



```
\def\gdpwagesdataone{{1.43,1.93,3.32,
  4.20},{3.440,4.500,5.050,5.530}}
\scatterchart[basic][
  xlength=5.4, xscale=1.2, yscale=.5,
  dotwidth=8, dotscolor=pinkflamingo,
  xcoor=yes, axesunits=yes,
  xunit=Earnings (100 EUR),
  yunit=GDPpc (1000 EUR)][
  method=struct, data={\gdpwagesdataone}
]
```



Stock charts:
 high-low-close, open-high-low-close, volume-
 high-low-close, volume-open-high-low-close



```
\def\sharesdatafour{{8.7465,34.62,
36.35,32.56,32.98},{3.0948,26.94,
27.22,20.06,20.37},{6.1873,24.37,
25.24,22.91,24.49},{7.0329,20.10,
21.55,19.90,20.31},{8.2355,24.30,
26.45,23.74,25.09}}
\def\datelabels{11.3.,18.3.,
25.3.,1.4.,8.4.}
\stockchart[vohlcv][
ylength=3.5,
xscale=.65,yscale=0.1,
left=0.5,
ogtcrectcolor=baloongray,
oltcrectcolor=gainsboro,
ocrectcontour=no,
vrectcolor=adriblue,
vrectcontour=no,
axesunits=yes,
xunit=Date,yunit=EUR][
method=struct,
data={\sharesdatafour},
xlabel={\datelabels}]
```

Conclusion

The main output of this work is the module `t-statistical-charts` the purpose of which is to draw charts according to the user's requirements. The user can choose from 25 chart types and set almost 100 parameters controlling the final appearance of the chart.

Another output is the publication *Drawing statistical charts* (Kocurová, Kašparová and Hála, 2020) which serves as a manual for the module.

We believe that this module and the corresponding publication will not only meet our personal requirements, but that it will also be used by other ConTeXt users and, therefore, bring a wider benefit.

Regarding further development of the module, the first obvious option is to extend it to support drawing other types of charts such as waterfall, funnel or combo. The last of these is a combination of two or more chart types in one, to make the data easier to read. Also, there could be an option to draw existing charts in a 3D view as well as displaying charts with 3D data.

Parameter enhancements could be added in future as well. For example a legend explaining the meaning of the colours used to distinguish the values (represented by dots, columns, bars, etc.) or the data series.

Now and future

Since Tamara's and Adriana's successful defenses, the code has been cleaned of some programming errors and some minor changes have been made over several iterations. For example, the `xunit` and `yunit` parameters were introduced to display units on the axes, the display of rounded values was adjusted and the names of the colour palettes were unified.

All requests received to date to modify or expand the user documentation have been resolved, which is now complete with respect to the current state of the module. Work has begun on the programming documentation.

The code is already available for download and use by potential interested parties, but it is stored on GitHub in a non-public repository for the time being. It will be published after a complete code cleanup and modification of the concept for component-based graph rendering.

Two additional bachelor's theses are currently assigned: the aim of the first one is to expand the module to implement the remaining two-dimensional graphs, while the second will add three-dimensional graphs.

References

- ADOBE SYSTEMS. *PostScript Language Reference*: third edition [on-line]. 1999. [cit. 2020-01-17]. Available at: www.adobe.com/jp/print/postscript/pdfs/PLRM.pdf.
- CASIDY, J. *L^AT_EX Graphics using TikZ: A Tutorial for Beginners (Part 1)—Basic Drawing* [on-line]. 2013. [cit. 2020-01-17]. Available at: [cs.overleaf.com/learn/latex/LaTeX_Graphics_using_TikZ:_A_Tutorial_for_Beginners_\(Part_1\)%E2%80%94Basic_Drawing](http://cs.overleaf.com/learn/latex/LaTeX_Graphics_using_TikZ:_A_Tutorial_for_Beginners_(Part_1)%E2%80%94Basic_Drawing).
- CONTEXT GARDEN. *MetaPost* [on-line]. 2014. [cit. 2020-01-17]. Available at: wiki.contextgarden.net/MetaPost.
- CONTEXT GARDEN. *TikZ* [on-line]. 2019. [cit. 2020-01-18]. Available at: wiki.contextgarden.net/TikZ.
- CRÉMER, J. *A very minimal introduction to TikZ* [on-line]. 2011. [cit. 2020-01-18]. Available at: cremeronline.com/LaTeX/minimalktikz.pdf.
- HAGEN, H. *Metafun: context mkiv* [on-line]. Hasselt: Boekplan, 2017. [cit. 2020-01-17]. 428 pp. ISBN 978-94-90688-02-8. Available at: www.pragma-ade.com/general/manuals/metafun-p.pdf.
- HÁLA, T. *Tabulky v ConTeXtu: přístupy, možnosti, algoritmy*. Zpravodaj Československého sdružení uživatelů T_EXu

- [on-line], 2019, 1–4, p. 24–43 [cit. 2020-01-18]. (ISSN 1213-8185.) Available at: www.cstug.cz/bulletin/pdf/2019-1-4.pdf.
- HENDERSON, T., HENNIG, S. *A Beginner's Guide to META-POST for Creating High-Quality Graphics* [on-line]. 2013. [cit. 2019-01-12]. Available at: www.tug.org/docs/metapost/mpintro.pdf.
- HENNIG, S. *The bpolynomial package* [on-line]. 2007. [cit. 2020-01-17]. Available at: mirrors.nic.cz/tex-archive/graphics/metapost/contrib/macros/bpolynomial/bpolynomial.pdf.
- HOBBY, J. D. *Drawing Graphs with MetaPost* [on-line]. 2018. [cit. 2020-01-17]. Available at: www.tug.org/docs/metapost/mpgraph.pdf.
- HOBBY, J. D. *MetaPost: A user's manual* [on-line]. 2019. [cit. 2020-01-17]. Available at: www.tug.org/docs/metapost/mpman.pdf.
- KAŠPAROVÁ, A. *Implementation of module drawing charts for ConTeXt*. (Bachelor thesis.) Brno: Mendel University in Brno, 2020. 45 pp. Supervisor: Tomáš Hála.
- KERNIGHAN, B. W. *PIC—A Graphics Language for Typesetting User Manual* [on-line]. 1991. [cit. 2020-01-17]. Available at: doc.cat-v.org/unix/v10/10thEdMan/pic.pdf.
- KOCUROVÁ, T. *Implementation of module drawing charts for ConTeXt*. (Bachelor thesis.) Brno: Mendel University in Brno, 2020. 42 pp. Supervisor: Tomáš Hála.
- KOCUROVÁ, T., KAŠPAROVÁ, A., HÁLA, T. *Drawing statistical charts* [on-line]. 1. ed. Brno: [s. n.], 2020. [cit. 2025-06-22]. 115 pp. Available at: www.thala.cz/statcharts/statistical-charts.pdf.
- KOTTWITZ, S. *L^AT_EX Cookbook*. Birmingham: Packt Publishing, 2015. 378 pp. ISBN 9781-78439-514-8.
- LUA.ORG. *Reference manuals*. In [Lua.org] [on-line]. 2024 [cit. 2025-07-16]. Available at: www.lua.org/manual/.
- MITTELBACH, F., GOOSSENS, M. *The L^AT_EX Companion: Second Edition*. 2.ed. Boston: Addison-Wesley Professional, 2004. 1120 pp. ISBN 0-201-36299-6.
- MORAWSKI, J.-U. *piechartMP: Drawing pie-charts with MetaPost* [on-line]. 2002. [cit. 2020-01-17]. Available at: ctan.math.illinois.edu/graphics/metapost/contrib/macros/piechartmp/piechartmp.pdf.
- MORAWSKI, J.-U. *latexMP* [on-line]. 2005. [cit. 2020-01-17]. Available at: ftp.cvut.cz/tex-archive/graphics/metapost/contrib/macros/latexmp/doc/latexmp.pdf.
- TIŠNOVSKÝ, P. *Tvorba grafů v jazyce Go: kreslení ve webovém klientu* [on-line]. 2020. [cit. 2020-01-17]. Available at: www.root.cz/clanky/tvorba-grafu-v-jazyce-go-kresleni-ve-webovem-klientu/.

Acknowledgement

We thank Hans for valuable comments on the organization of the code.

Tamara's and Adriana's participation in the conference was supported by the special scholarship of the Dean of the Faculty of Business and Economy, Mendel University in Brno.