Drawing statistical charts

Tamara Kocurová
Adriana Kašparová
Tomáš Hála

ConTEXt is a system for typesetting documents based on TEX and MetaPost, main author is Hans Hagen.
Lua is a programming language primarily for embedded use in applications, by Roberto Ierusalimschy.
MetaPost is a tool for creating graphics, written by John D. Hobby.

Acknowledgement:

Hans Hagen—for ideas how to improve the code,
Wolfgang Schuster—for the help with syntax setup.

Preface

This book was created to make the module for drawing charts easier to use for everyone interested. The reason of programming the module itself, was to create a tool to draw charts (as it is possible in the spreadsheets) in ConTEXt, using Lua and MetaPost languages. This option was something which was missing, as we got introduced to and then used the ConTEXt typographic system at the university. Shortly, it became a familiar language for document typesetting to us. We were sure to use this environment for typesetting our final thesis, so the idea of actually creating a module for ConTEXt was a combination of a useful and enjoyable thing at the same time. However, this would not be possible without a cooperation and a lot of work done by Tomáš Hála. Our gratitude also belongs to Hans Hagen and John D. Hobby for their very useful manuals, which helped us to understand the basics of drawing in MetaPost combined with ConTEXt.

With this module, the user will be able to draw area, bar, bubble, column, line, pie, radar, scatter and stock charts. This document should be a sufficient guide for everyone who will be willing to use the module to create their own charts in ConTEXt.

Tamara & Adriana

Content

1 About the module

1.1 Parts of the module

The module consists of three files:

t-statistical-charts.mkiv (v0.14)—containing basic definitions of commands,
t-statistical-charts.lua (v0.42)—with the complete implementation of drawing, and
t-readdata.lua (v0.14)—used for reading and processing of user data.

The module can be joined a typical way:

```
\usemodule[statistical-charts]
```

1.2 Chart frames

All charts are shipped out as a context of a special frame `\statisticalchart` which was derived from a standard `\framed`. Corresponding options, e.g. offsets, style etc., are available via

```
\setupframed[statisticalchart][...]
```

All the text parts of charts can be controlled by specific pre-defined framed elements. One can use the following commands:

```
\setupframed[xaxislabels][...]
```

can be used in all charts except pie and radar charts.

```
\setupframed[yaxislabels][...]
```

can be used in all charts except pie and radar charts.

```
\setupframed[dotslabels][...]
```

can be used in area, line, radar and scatter charts.

```
\setupframed[layerslabels][...]
```

can be used in radar chart.

```
\setupframed[partslabels][...]
```

can be used in pie chart.
The example of setting the **xaxislabels** follows.

```
\setupframed[xaxislabels][
  foregroundcolor=red,
  frame=on,
  toffset=0.5cc,
  foregroundstyle={\setupbodyfont[1cc]}]
```

1.3 Palettes

Unlike the expected use of a standard system of palettes used in ConTEXt, which is based on named colors, for drawing charts, the palettes with *indexed* colors are needed. Therefore, this approach has been implemented. This tool is quite easy for user-defined palettes.

1.3.1 Indexed palettes

To define a palette, the command `\definepalette` is used (see below). The command calls the function (d.definepalette(name, colors)) in Lua.

```
\def\definepalette[#name][#colors]{
   \ctxlua{d.definepalette('#name',
   '#colors')}
}

function d.definepalette(name, colors)
   d.palette = d.palette or {}
   d.palette[name] =
   utilities.parsers.settings_to_array(
   colors)
end
```

1.3.2 Pre-defined palettes for charts

The palettes used in the chart examples are defined this way.

```
\definepalette [redpalette] [
   cosmos, monalisa, bittersweet,
   redorange, freespeechred]

\definecolor [cosmos]        [x=ffcccc]
\definecolor [monalisa]      [x=ff9999]
\definecolor [bittersweet]   [x=ff6666]
\definecolor [redorange]     [x=ff3333]
\definecolor [freespeechred] [x=cc0000]
```

```
\definepalette [orangepalette] [
  bisque, peachorange, rajah,
  neoncarrot, tamarange]

\definecolor [bisque]        [x=ffe5cc]
\definecolor [peachorange]   [x=ffcc99]
\definecolor [rajah]         [x=ffb266]
\definecolor [neoncarrot]    [x=ff9933]
\definecolor [tamarange]     [x=ff8000]
```

```
\definepalette [lightgreenpalette] [
  snowflurry, reef, sulu, greenyellow,
  lawngreen, kellygreen, scarfgreen,
  lapalma, bilbao, verdungreen]

\definecolor [snowflurry]    [x=e5ffcc]
\definecolor [reef]          [x=ccff99]
\definecolor [sulu]          [x=b2ff66]
\definecolor [greenyellow]   [x=99ff33]
\definecolor [lawngreen]     [x=72e300]
\definecolor [kellygreen]    [x=66cc00]
\definecolor [scarfgreen]    [x=55a800]
\definecolor [lapalma]       [x=478f00]
\definecolor [bilbao]        [x=3a7300]
\definecolor [verdungreen]   [x=326300]
```

```
\definepalette [darkgreenpalette] [
  pastelgreen, fern, limegreen,
  islamicgreen, green]

\definecolor [pastelgreen]   [x=80e874]
\definecolor [fern]          [x=5dd45d]
\definecolor [limegreen]     [x=3cc23c]
\definecolor [islamicgreen]  [x=009900]
\definecolor [green]         [x=006e00]
```

```
\definepalette [turquoisepalette] [
  grannyapple, magicmint, aquamarine,
  mediumspringgreen, shamrock]

\definecolor [grannyapple]   [x=c0ffd3]
\definecolor [magicmint]     [x=99ffcc]
\definecolor [aquamarine]    [x=66ffb2]
\definecolor [mediumspringgreen]
                             [x=33ff99]
\definecolor [shamrock]      [x=4bd691]
```

```
\definepalette [cyanpalette] [
   electricblue, babyblue,
  aqua, lightcyan, darkturquoise]

\definecolor [lightcyan]     [x=ccffff]
\definecolor [electricblue]  [x=99ffff]
\definecolor [babyblue]      [x=66ffff]
\definecolor [aqua]          [x=00f3f3]
\definecolor [darkturquoise] [x=00dada]
```

```
\definepalette [bluepalette] [
  palecornflowerblue, lightskyblue,
  mayablue, adriblue, dodgerblue]
```

```
\definecolor [palecornflowerblue]
                          [x=b0d7ff]
\definecolor [lightskyblue]   [x=99ccff]
\definecolor [mayablue]       [x=66b2ff]
\definecolor [adriblue]       [x=3399ff]
\definecolor [dodgerblue]     [x=0080ff]
```

```
\definepalette [indigopalette] [
  lavenderblue, portage, mediumslateblue,
  neonblue, mediumblue]

\definecolor [lavenderblue]   [x=babaff]
\definecolor [portage]        [x=9999ff]
\definecolor [mediumslateblue]
                          [x=6666ff]
\definecolor [neonblue]       [x=3333ff]
\definecolor [mediumblue]     [x=0000cc]
```

```
\definepalette [purplepalette] [
  mauve, coctailpurple, mediumpurple,
  blueviolet, electricindigo]

\definecolor [mauve]          [x=e0c2ff]
\definecolor [coctailpurple]  [x=cc99ff]
\definecolor [mediumpurple]   [x=b266ff]
\definecolor [blueviolet]     [x=9933ff]
\definecolor [electricindigo] [x=7f00ff]
```

```
\definepalette [magentapalette] [
  snuff, violet, pinkflamingo,
  razzledazzlerose, deepmagenta]

\definecolor [snuff]          [x=ffd4ff]
\definecolor [violet]         [x=ff99ff]
\definecolor [pinkflamingo]   [x=ff66ff]
\definecolor [razzledazzlerose]
                          [x=f331f3]
\definecolor [deepmagenta]    [x=da00da]
```

```
\definepalette [pinkpalette] [
  classicrose, carnationpink, hotpink,
  wildstrawberry, ruby, razzmatazz,
  jazzberryjam, crayonflower,
  tyrianpurple, sweettreat]

\definecolor [classicrose]    [x=ffcce5]
\definecolor [carnationpink]  [x=ff99cc]
\definecolor [hotpink]        [x=ff66b2]
\definecolor [wildstrawberry] [x=ff3399]
\definecolor [ruby]           [x=f20c7f]
\definecolor [razzmatazz]     [x=db006b]
\definecolor [jazzberryjam]   [x=c30061]
\definecolor [crayonflower]   [x=99004c]
\definecolor [tyrianpurple]   [x=800040]
```

```
\definecolor [sweettreat]     [x=660033]
```

Moreover, there are these two following palettes available:


```
\definepalette [yellowpalette] [
  cream, canary, laserlemon,
  gorse, chartreuseyellow]

\definecolor [cream]          [x=ffffcc]
\definecolor [canary]         [x=ffff99]
\definecolor [laserlemon]     [x=ffff66]
\definecolor [gorse]          [x=ffff33]
\definecolor [chartreuseyellow]
                              [x=e3e300]
```

```
\definepalette [graypalette] [
  gainsboro, silver, nobel,
  baloongray, zambezi]

\definecolor [gainsboro]      [x=e0e0e0]
\definecolor [silver]         [x=c0c0c0]
\definecolor [nobel]          [x=a0a0a0]
\definecolor [baloongray]     [x=808080]
\definecolor [zambezi]        [x=606060]
```

2 Parameters

All of the parameters which are used in the charts are explained in this chapter. The parameters can be found in alphabetical order in *Chapter* .

Unit

unit: unit of xlength, ylength, xshift, yshift, xscale, yscale, distance, barwidth, columnwidth, left, bottom, bubblediameter, diameter, holediameter, distanceout, distancebetween

Axes

xlength: length of *x*-axis(if the entered value of length is lower than required according to data, then minimum length needed will be automatically set)
ylength: length of *y*-axis(if the entered value of length is lower than required according to data, then minimum length needed will be automatically set)
xshift: change of *x*-axis from point 0,0 (positive right, negative left)
yshift: change of *y*-axis from point 0,0 (positive up, negative down)
xwidth: width of the *x*-axis in pt
ywidth: width of the *y*-axis in pt
xaxislabels: labels on *x*-axis
yaxislabels: labels on *y*-axis
axesunits: labels of axes' units
xunit: *x*-axis unit
yunit: *y*-axis unit
xaxislabelsnoval: labels on *x*-axis, starting from the last value to the end of axis
yaxislabelsnoval: labels on *y*-axis, starting from the last value to the end of axis
xlabelscount: number of *x*-axis labels
ylabelscount: number of *y*-axis labels
xaxislabinit: initial *x*-axis label (only if *x*-coordinate is entered)
xaxislabinc: incremental value of *x*-axis labels (only if *x*-coordinate is entered)
axes: axes emanating from the center to the polygon vertices
axiswidth: width of axes in pt
axisline: line pattern of axes
axiscolor: color of axes
xcoor: if *x*-coordinate entered—only for bubble chart
zcoor: if *x*-coordinate entered—only for bubble chart (values will be represented as bubble diameter—multiplied by bubblediameter)

Numbers

equaldecimals: equalizes the number of decimal places in labels according to maximum number of decimal places in the given numbers (not for *x*-axis labels in area, bubble, column, line, scatter and stock charts and *y*-axis in bar charts)
decimals: maximum number of decimals (eg. 1, 2, 3, ...) in labels (not for *x*-axis labels in area, bubble, column, line, scatter and stock charts and *y*-axis

in bar charts)
rounding: rounding method
perc: labels displayed in percents

Labels

layerslabels: labels of layers—only for radar chart
labels: labels of the parts—only for pie chart
labelposition: position of the labels—only for pie chart

Scale of the chart

xscale: change of the scale of *x*-axis (the higher the number, the larger the chart)
yscale: change of the scale of *y*-axis (the higher the number, the larger the chart)

Positioning

distance: distance between values or layers (if *x*-coordinate is entered, the value of distance will be always set to 1)
left: left margin of the first value from *y*-axis(if *x*-coordinate is entered, the value of left will be always set to 1)
bottom: bottom margin of the first value from *x*-axis

Grid

grid: grid of the chart
gridcolor: color of the grid lines
gridline: line pattern of grid lines
gridlinewidth: width of grid lines in pt
xgridlines: grid lines perpendicular to *x*-axis
ygridlines: grid lines perpendicular to *y*-axis
xgridlinesdensity: density of grid lines perpendicular to *x*-axis(does not work for numbers lower than 1 in 100% stacked bar chart)
ygridlinesdensity: density of grid lines perpendicular to *y*-axis(does not work for numbers lower than 1 in 100% stacked area, column and line chart)
xgridlinesnoval: grid lines perpendicular to *x*-axis, starting from the last value to the end of axis
ygridlinesnoval: grid lines perpendicular to *y*-axis, starting from the last value to the end of axis
xgridlinesleft: grid lines perpendicular to *x*-axis starting from the beginning of *x*-axis to the first value
layerline: pattern of layer lines—only for radar chart
layerwidth: width of layer lines in pt—only for radar chart

Dots

dots: dots representing values
dotscolor: color of dots (used if no dotspalette entered)
dotspalette: color of dots

dotswidth: width of dots in pt
dotslabels: labels of dots

Fill

fill: fill of the area
fillcolor: color of the filled area (used if no fillpalette entered)
fillpalette: palette of fill color
filltransparency: transparency of fill color (0—total transparency, 1—no transparency)

Contour

contour: contour of objects
contourcolor: color of contour (used if no contourpalette entered)
contourpalette: palette color of contour
contourwidth: width of contour in pt
ocrectcontour: contour of rectangle representing open-close
ocrectcontourcolor: color of open-close rectangle contour
ocrectcontourwidth: width of open-close rectangle contour in pt
vrectcontour: contour of rectangle representing volume
vrectcontourcolor:color of volume rectangle contour
vrectcontourwidth: width of volume rectangle contour in pt

Objects

line: area delimiting line
linecolor: color of line connecting values (used if no linepalette entered)
linepalette: palette of line color
linewidth: width of line connecting values in pt
barcolor: color of bar (used if no barpalette entered)
barpalette: palette of bar color
bartransparency: transparency of bar color
barwidth: width of bar
columncolor: color of bar (used if no columnpalette entered)
columnpalette: palette of column color
columntransparency: transparency of column color
columnwidth: width of column
bubblediameter: diameter of bubble (in case the *z*-coordinate is entered, those values will be multiplied by bubblediameter)
bubblecolor: color of bubble (used if no bubblepalette entered)
bubblepalette: palette of bubble color
bubbletransparency: transparency of bubble color
diameter: diameter of pie/doughnut
holediameter: diameter of doughnut hole
rightlinecolor: color of right line representing close value—only for stock chart
rightlinewidth: width of right line representing close value in pt
rightlinelength: length of right line representing close value in pt
rectanglewidth: width of rectangle representing

open-close and volume

ogtcrectcolor: color of rectangle representing open-close if open is greater than close

oltcrectcolor: color of rectangle representing open-close if open is less than close

ocrecttransparency: transparency of rectangle representing open-close (0—total transparency, 1—no transparency)

vrectcolor: color of rectangle representing volume

vrecttransparency: transparency of rectangle representing volume (0—total transparency, 1—no transparency)

Other

step: the value represented by one layer

layers: number of layers (if the entered number of layers is lower than required according to data, then minimum number needed will be automatically set)

layercolor: color of layers

distanceout: distance between the center and the inner point of the parts which will be out

distancebetween: distance between the doughnuts which represent the data series

coef: coefficient to adjust the distance between labels and the parts

out: list of indexes of parts which will be out

3 Area charts

The area chart is similar to the line chart, except the area between the horizontal axis and the line (lines) is colored (Chesnutt, 2016). This type of chart is mostly used when it is suitable to show the trend (alternatively trends if there is more than one series) of the data, where all of the data have to occur either in positive or in negative values. The subtypes of the area chart are:

- basic,
- stacked and
- 100% stacked.

Area chart parameters

The following parameters (and their default values) can be set in the area chart.

To work with the axes:

xlength: minimum length needed according to data
ylength: minimum length needed according to data
xshift: 0
yshift: 0
xwidth: 0.5
ywidth: 0.5

To scale the entire chart:

xscale: 1
yscale: 1

To set the unit:

unit: cm

To set the distance between the values:

distance: 1

To adjust the grid:

grid: on
gridcolor: middlegray
gridline: dotted
gridlinewidth: 1
ygridlinesdensity: 1
xgridlines: yes
ygridlines: yes
xgridlinesnoval: yes

To set the fill:

fillcolor: gray
fillpalette:
filltransparency: 1

To set the line:

line: yes
linecolor: black
linepalette:
linewidth: 1

To set the dots:

dots: yes
dotscolor: black
dotspalette:
dotswidth: 5

To set the labels:

xaxislabels: yes
yaxislabels: yes
axesunits: yes
xunit:
xunit:
xunit: yes
dotslabels: no
decimals: 1
rounding: halfup
equaldecimals: no

3.1 Basic area chart

The basic area chart can represent one or more data series. The example in *Chart 3.1* shows the acquisition of citizenship in 2014–218 in Slovakia—see the data in *Table 3.1*. When more data series are shown, then the lines and filled areas will be displayed on top of each other. An example can be seen in *Chart 3.2*, where the acquisition in Slovakia, Iceland, and Estonia is displayed (*Table 3.2*).

Year	2014	2015	2016	2017	2018
Citizenship	13	19	34	20	36

Table 3.1 Number of citizenship acquisition in age group 65 and over in 2014–2018 in Slovakia (Eurostat, 2020a)

To set the area chart attributes, the parameters in *Chapter* can be used. The *Chart 3.1* was created by using the following commands—see the parameters set (changed from default).

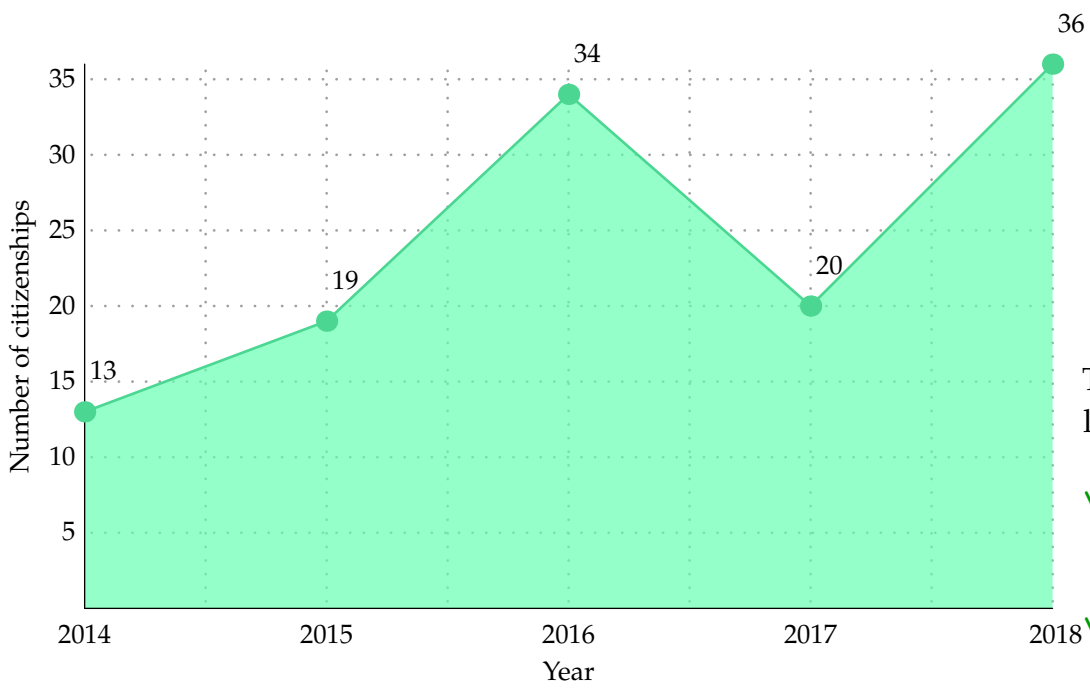

Chart 3.1 Basic area chart with one data series

```
\def\citizenshipdataone{13,19,34,20,36}

\def\yearlabelsone{2014,2015,2016,2017,
   2018}

\setupframed[dotslabels][loffset=1.1cc,
   boffset=0.4cc]

\areachart[basic][
   yscale=0.2, xscale=1.6,
   distance=2,
   xgridlinesdensity=2,
   fillcolor=aquamarine,
   line=yes, linecolor=shamrock,
   dotscolor=shamrock,
   dotswidth=8, dotslabels=yes,
   axesunits=yes, xunit=Year,
   yunit=Number of citizenships][
   data={\citizenshipdataone},
   xlabels={\yearlabelsone}]
```

The *Chart 3.2* was created by using the commands below—see the parameters set (changed from default).

```
\def\citizenshipdatatwo{{13,19,34,20,36
   },{10,19,11,10,11},{21,10,18,10,10}}

\def\yearlabelsone{2014,2015,2016,2017,
   2018}

\areachart[basic][
   yscale=0.2, xscale=1.6,
   distance=2,
   xgridlinesdensity=2,
   fillpalette=turquoisepalette,
   line=no,
   dotscolor=shamrock,
   dotswidth=8,
   axesunits=yes,
   xunit=Year,
   yunit=Number of citizenships][
   method=struct,
   data={\citizenshipdatatwo},
   xlabels={\yearlabelsone}]
```

Year	2014	2015	2016	2017	2018
Slovakia	13	19	34	20	36
Iceland	10	19	11	10	11
Estonia	21	10	18	10	10

Table 3.2 Number of citizenship acquisition in age group 65 and over in 2014–2018 in Slovakia, Iceland and Estonia (Eurostat, 2020a)

3.2 Stacked area chart

The stacked area chart is convenient if the author wants to illustrate the contribution of each part to the whole. The line at the top of the graph represents the sum of all values in each particular point because the data are shown cumulatively (The Data Visualisation Catalogue, 2019a).

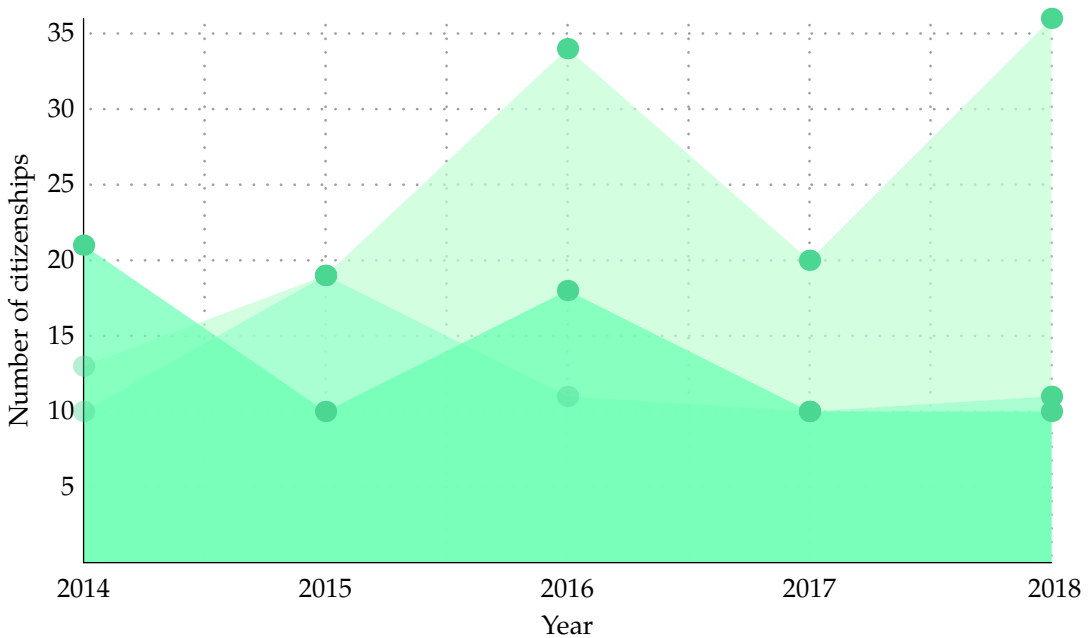

Chart 3.2 Basic area chart with multiple data series

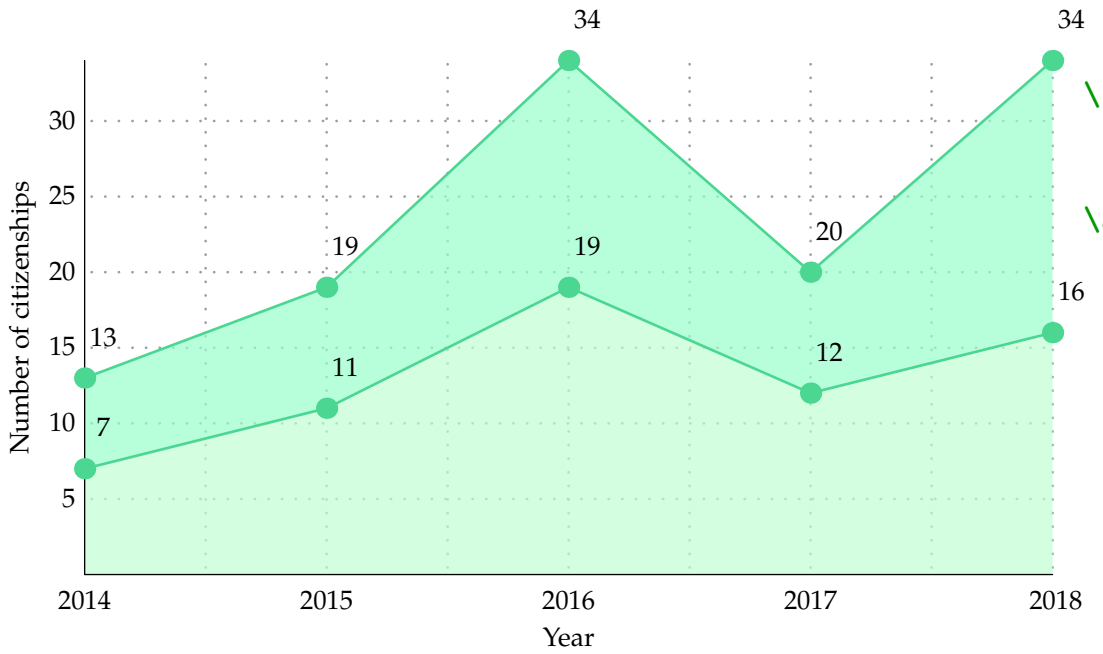

Chart 3.3 Stacked area chart with multiple data series

The stacking can be done only when there is more than one series of data. For example (see *Chart 3.3*), this subtype can show the acquisition of citizenship according to sex in multiple years for a specific country—data in Table ??.
The *Chart 3.3* was created by using the commands below—see the parameters set (changed from default).

```
\def\citizenshipdatathree{{7,11,19,12,16
  },{6,8,15,8,18}}

\def\yearlabelsone{2014,2015,2016,2017,
  2018}

\setupframed[dotslabels][loffset=1.1cc,
  boffset=0.4cc]

\areachart[stacked][
  yscale=0.2, xscale=1.6,
  distance=2,
  xgridlinesdensity=2,
  fillpalette=turquoisepalette,
  line=yes, linecolor=shamrock,
  dotscolor=shamrock,
  dotswidth=8, dotslabels=yes,
  axesunits=yes, xunit=Year,
  yunit=Number of citizenships][
  method=struct,
  data={\citizenshipdatathree},
  xlabels={\yearlabelsone}]
```

3.3 100% stacked area chart

The lines which represent the values are always plotted relatively, so all of the series shown sum to 100%, which means the entire graph area is filled with various colors. Hence, the absolute total at each point cannot be read from the graph (AnyChart, c2020). See the example *Chart 3.4* using the data in Table ??. To manipulate with 100% stacked area chart, in addi-

tion to the parameters listed in *Chapter* and explained in *Chapter 2*—except the **yscale** parameter, also the following one (and its default value) can be used.

To set the number of *y*-axis labels:

ylabelscount: 5

The *Chart 3.4* was created by using the commands below—see the parameters set (changed from default).

```
\def\citizenshipdatathree{{7,11,19,12,16
   },{6,8,15,8,18}}

\def\yearlabelsone{2014,2015,2016,2017,
   2018}

\areachart[stacked100][
   ylength=7, xscale=1.55,
   distance=2, grid=off,
   fillpalette=turquoisepalette,
   line=yes, linecolor=shamrock,
   dotscolor=shamrock,
   dotswidth=8,  axesunits=yes,
   xunit=Year, yunit=Number of citizenships][
   method=struct, data={\citizenshipdatathree},
   xlabels={\yearlabelsone}]
```

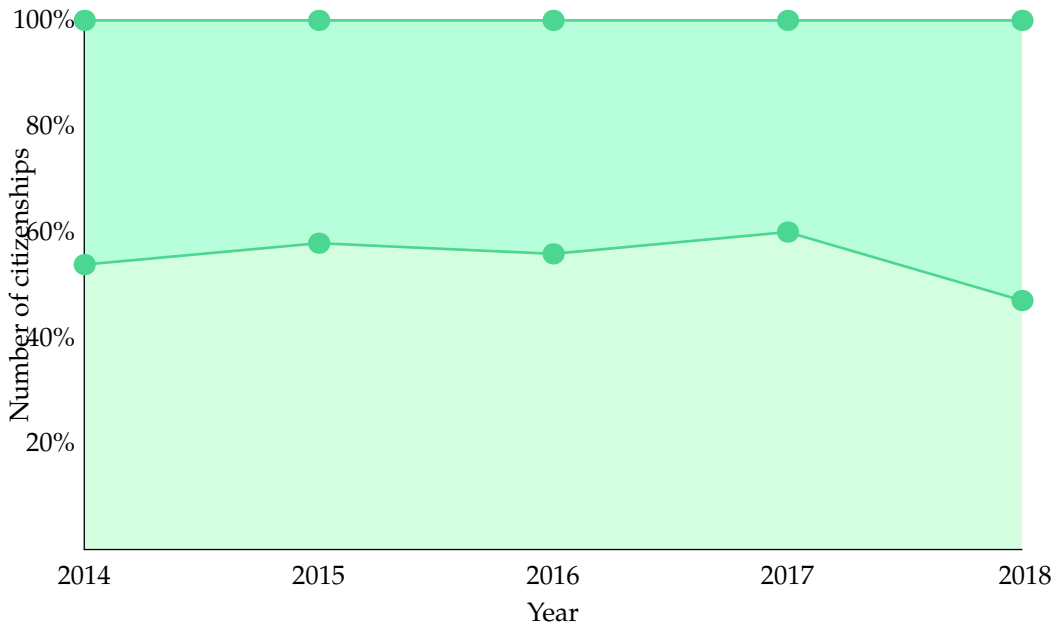

Chart 3.4 100% stacked area chart with multiple data series

4 Bar charts

In bar chart, the values are represented by the horizontal axis (x-axis) and the categories, on the other hand, are placed along the vertical axis (Keynote Support, c2008–2020). This type of chart, as well as the column chart, is very common in marketing because it provides clear image of data comparison (Lile, 2017). These charts also offer the advantage of being easily understandable by anyone with a very little knowledge. The bar chart can be also (as well as the column chart) represented by four different subtypes which are:

- basic,
- clustered,
- stacked and
- 100% stacked.

Bar chart parameters

The following parameters (and their default values) can be set in the bar chart.

To work with the axes:

xlength: minimum length needed according to data (3 in 100% stacked bar chart)
ylength: minimum length needed according to data + 1 bottom
xshift: 0
yshift: 0
xwidth: 0.5
ywidth: 0.5

To scale the entire chart:

xscale: 1
yscale: 1

To set the unit:

unit: cm

To set the position of the initial and all other values:

distance: 1
bottom: 1

To adjust the grid:

grid: on
gridcolor: middlegray
gridline: dotted
gridlinewidth: 1
xgridlinesdensity: 1
xgridlines: yes
ygridlines: yes
ygridlinesnoval: yes

To set the bars:

barcolor: gray
barpalette:
bartransparency: 1
barwidth: 1
contour: yes
contourcolor: black
contourpalette:
contourwidth: 0.5

To set the labels:

xaxislabels: yes
yaxislabels: yes
axesunits: yes
xunit:
xunit:
xunit: yes
decimals: 1
rounding: halfup
equaldecimals: no

4.1 Basic bar chart

The basic bar chart visualizes data as a set of simple rectangular bars with different lengths. In contrast with the column chart example (*Chart 6.1*), the axes are swapped, so the vertical axis can represent age groups and the horizontal axis can display the extent of unemployment—see the data in *Table 4.1*). The example can be seen in *Chart 4.1*.
To set the bar chart attributes, the parameters in *Chapter* can be used. Even though basic bar chart works with just one-dimensional data series, even if multiple series will be inserted, the command will work but only the first data series will be shown. The *Chart 4.1* was created by using the commands below—see the parameters set (changed from default).

```
\def\unemploydataone{20,17.2,14.2,29.3,
   22.5,18.4}

\def\agelabels{15--24,25--29,30--34,
   35--44,45--54,55+}

\barchart[basic][
   xscale=0.5, yscale=0.6,
   barcolor=redorange,
   bartransparency=0.7,
   axesunits=yes,
   xunit=Number of unemployed
            persons (thousands),
```

```
yunit=Age][
data={\unemploydataone},
ylabels={\agelabels}]
```

Age group	15–24	25–29	30–34	35–44	45–54	55+
Unemployment	20.0	17.2	14.2	20.3	22.5	18.4

Table 4.1 Average unemployment according to age groups in thousands in 2018 in the Czech Republic (CZSO, 2020)

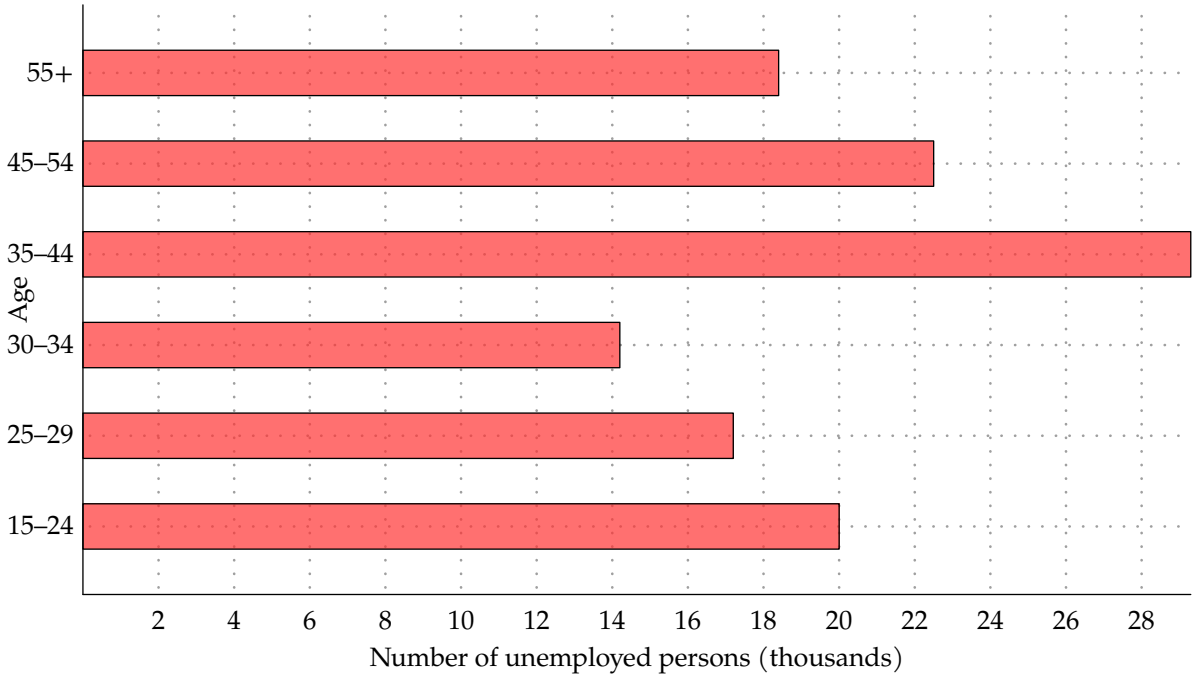

Chart 4.1 Basic bar chart with one data series

4.2 Clustered bar chart

Clustered bar chart has essentially the same purpose as the clustered column chart. It is used when the author works with more than one data series, so they can be compared to each other. According to WallStreetMojo (c2020), a clustered bar chart is actually putting next to each other bars from different charts. To show an example (see *Chart 4.2*), instead of one value for a whole year as in the previous subtype, the values for each quarter were used (see *Table 4.2*).

Age group	15–24	25–29	30–34	35–44	45–54	55+
Q1	18.9	16.3	14.0	35.5	25.0	19.9
Q2	20.1	17.6	15.3	26.5	20.6	18.2
Q3	23.0	19.4	13.9	29.4	22.4	19.2
Q4	17.9	15.3	13.7	25.8	22.0	16.4

Table 4.2 Average unemployment according to age groups in thousands in quarters of 2018 in the Czech Republic (CZSO, 2020)

The *Chart 4.2* was created by using the commands below—see the parameters set (changed from default).

```
\def\unemploydatatwo{{18.9,16.3,14,35.5,
   25,19.9},{20.1,17.6,15.3,16.5,20.6,
   18.2},{23,19.4,13.9,29.4,22.4,19.2},{
   17.9,15.3,13.7,25.8,22,16.4}}

\def\agelabels{15--24,25--29,30--34,
   35--44,45--54,55+}

\barchart[clustered][
   xscale=0.33, yscale=0.34,
   barpalette=redpalette,
   bartransparency=0.7,
   axesunits=yes,
   xunit=Number of unemployed
                  persons (thousands),
   yunit=Age][
   method=struct,
   data={\unemploydatatwo},
   ylabels={\agelabels}]
```

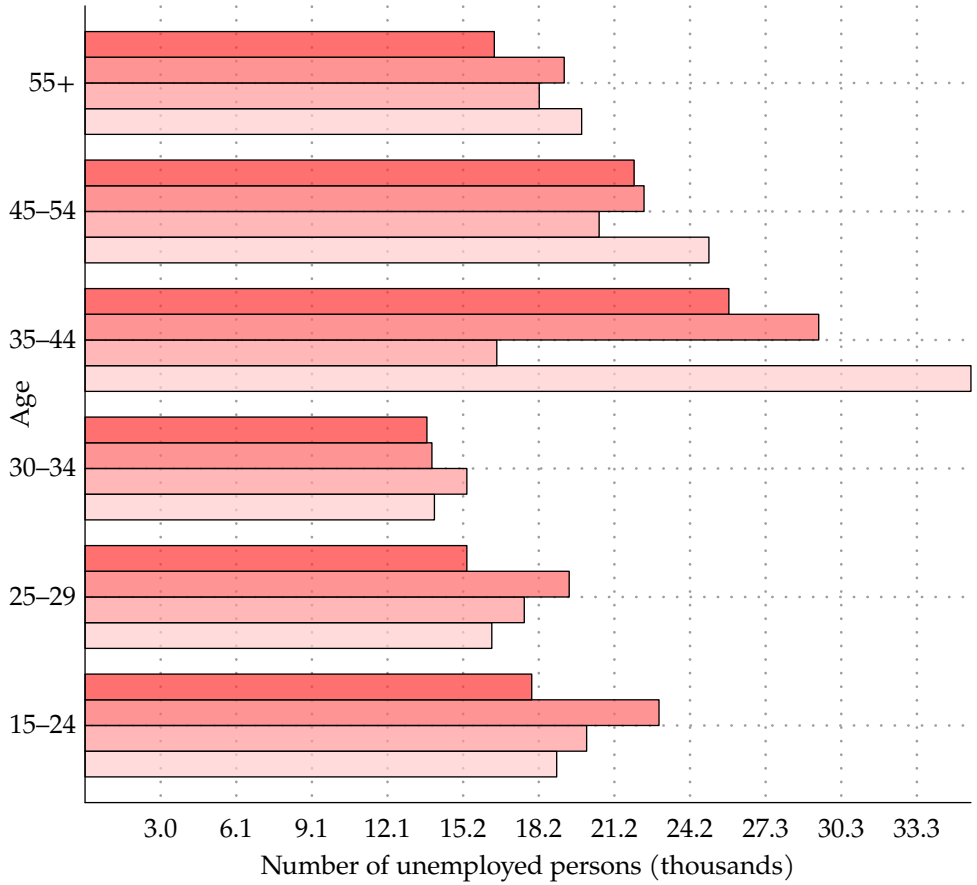

Chart 4.2 Clustered bar chart with multiple data series

4.3 Stacked bar chart

As The Data Visualisation Catalogue (2019b) states, in the stacked bar chart, the values of one category are arranged one after another. This type is especially useful for comparing the categories (segments) with each other because the values of one segment will add up to its total value. The data used in example (*Chart 4.3*) are shown in *Table 4.3*.

Age group	2014	2015	2016	2017	2018
15–24	56.4	43.7	34.6	25.1	20.0
25–29	42.2	37.5	30.6	19.2	17.2
30–34	41.5	35.7	26.4	17.5	14.2
35–44	78.9	65.3	48.7	41.0	29.3
45–54	63.6	49.4	38.6	31.3	22.5
55+	41.0	36.5	32.6	21.5	18.4

Table 4.3 Unemployment according to age groups in thousands in 2014–2018 in the Czech Republic (CZSO, 2020)

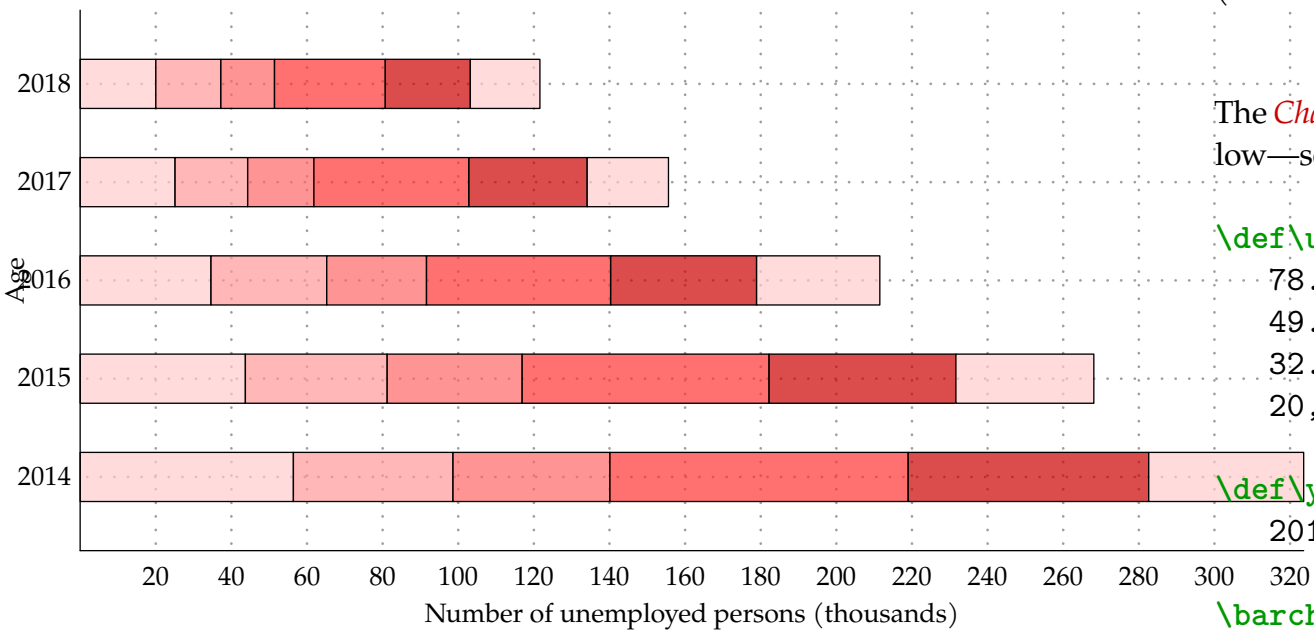

Chart 4.3 Stacked bar chart with multiple data series

The *Chart 4.3* was created by using the commands below—see the parameters set (changed from default).

```
\def\unemploydatathree{{56.4,42.2,41.5,
   78.9,63.6,41},{43.7,37.5,35.7,65.3,
   49.4,36.5},{34.6,30.6,26.4,48.7,38.6,
   32.6},{25.1,19.2,17.5,41,31.3,21.5},{
   20,17.2,14.2,29.3,22.5,18.4}}

\def\yearlabelsone{2014,2015,2016,2017,
   2018}

\barchart[stacked][
   xscale=0.05, yscale=0.65,
   barpalette=redpalette,
```

```
bartransparency=0.7,
decimals=-1,
axesunits=yes,
xunit=Number of unemployed persons (thousands),
yunit=Age][
method=struct,
data={\unemploydatathree}, c=r,
ylabels={\yearlabelsone}]
```

The *Chart 4.4* was created by using the following commands—see the parameters set (changed from default).

```
\def\unemploydatathree{{56.4,42.2,41.5,
   78.9,63.6,41},{43.7,37.5,35.7,65.3,
   49.4,36.5},{34.6,30.6,26.4,48.7,38.6,
   32.6},{25.1,19.2,17.5,41,31.3,21.5},{
   20,17.2,14.2,29.3,22.5,18.4}}
```

4.4 100% stacked bar chart

This type of chart does not show the absolute values, but their percentage out of 100% (Keynote Support, c2008–2020). Therefore, also the *x*-axis is not marked with the values themselves, but with the percentages up to 100% (see the *Chart 4.4*). The data in *Table 4.3* were used to show this example.

To manipulate with 100% stacked bar chart, in addition to the parameters listed in *Chapter* and explained in *Chapter 2*—except the **xscale** parameter, also the following one (and its default value) can be used.

To set the number of *y*-axis labels:

xlabelscount: 5

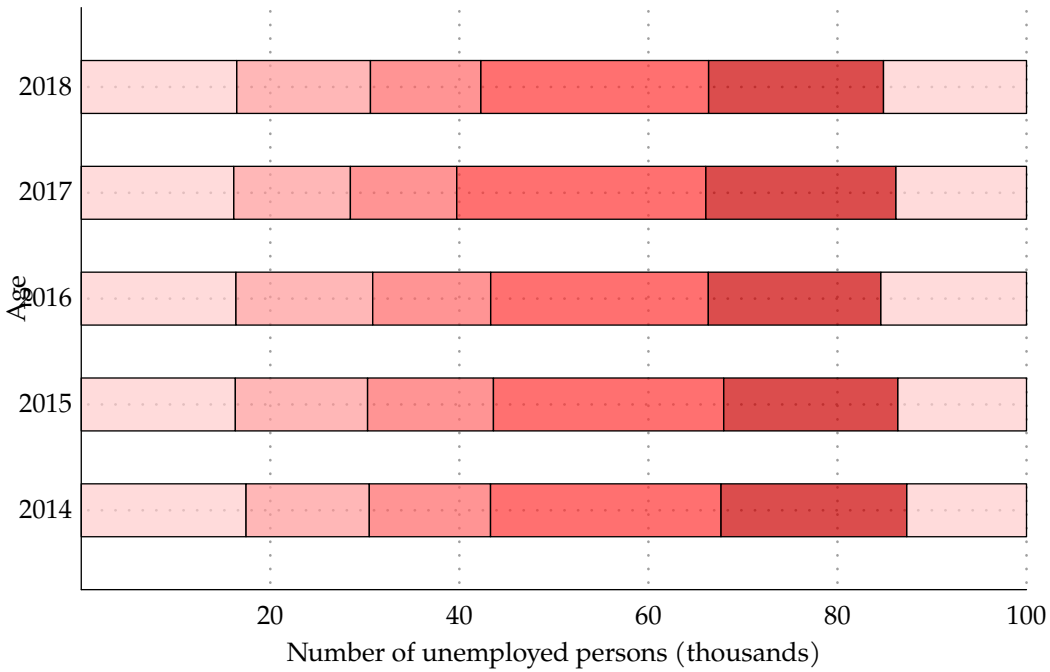

Chart 4.4 100% stacked bar chart with multiple data series

```
\def\yearlabelsone{2014,2015,2016,2017,
   2018}

\barchart[stacked100][
   xlength=12.5,
   yscale=0.7,
   xlabelscount=5,
   barpalette=redpalette,
   bartransparency=0.7,
   axesunits=yes,
   xunit=Number of unemployed persons (thousands),
   yunit=Age][
   method= struct,
   data={\unemploydatathree}, c=r,
   ylabels={\yearlabelsone}]
```

5

Bubble charts

The bubble chart is similar to the scatter chart, except the dots showing the data values are replaced with bubbles of a various (or the same) size (Yi, 2019). If the bubbles differ in their size, the data needs to be enlarged by third dimension. The size helps the reader to think about the data with the additional dimension, which can on the other hand cause putting the focus mainly on the larger bubbles. This is the reason why the author needs to carefully consider the size differences.

Bubble chart parameters

The following parameters (and their default values) can be set in the bubble chart.

To work with the axes:

xlength: minimum length needed according to data + left
ylength: minimum length needed according to data + 1 unit
xshift: 0
yshift: 0
xwidth: 0.5
ywidth: 0.5

To scale the entire chart:

xscale: 1
yscale: 1

To set the unit:

unit: cm

To set the distance between the values:

distance: 1
left: 1

To adjust the grid:

grid: on
gridcolor: middlegray
gridline: dotted
gridlinewidth: 1
xgridlinesdensity: 1
ygridlinesdensity: 1
xgridlines: yes
ygridlines: yes
xgridlinesnoval: yes
xgridlinesleft: yes

To set the bubbles:

bubblecolor: gray
bubblepalette:
bubblediameter: 1

filltransparency: 0.7
contour: yes
contourcolor: black
contourpalette:
contourwidth: 0.5

To set the whether the coordinates will be inserted:

xcoor: no
zcoor: no

To set the initial and increment value:

xaxislabinit: 1
xaxislabinc: 1

To set the labels:

xaxislabels: yes
yaxislabels: yes
axesunits: yes
xunit:
xunit:
xunit: yes
decimals: 1
rounding: halfup
equaldecimals: no

5.1 Bubble chart

To show examples of bubble charts, the data describing various variables (number of teachers, number of graduates, area) will be used—see the data in *Table 5.1*.

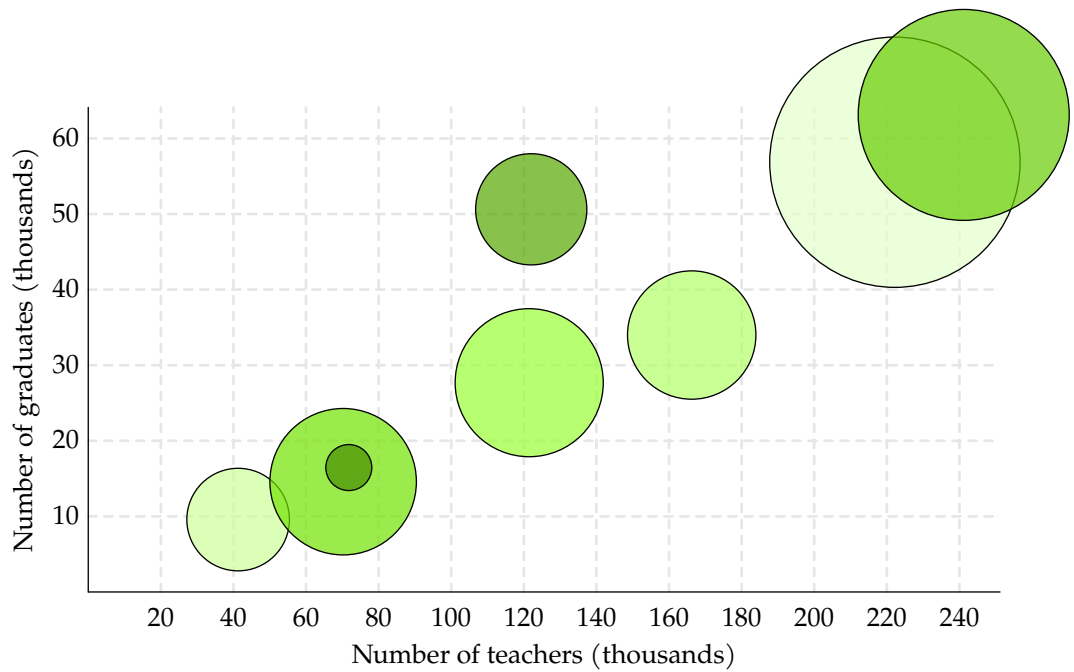

Chart 5.1 Bubble chart with *x*-, *y*- and *z*-coordinate with one data series

The example which uses all three coordinatecoordinates (x, y, z) is shown in *Chart 5.1*, and in *Chart 5.2* with multiple series, where one year was added to the data (see *Table 5.2*). Another example (*Chart 5.3*) shows the chart, where the z coordinate was not inserted, so the data does not use the area variable in the *Table 5.1*—therefore, the size of the bubbles is the same. The last example (*Chart 5.4*) of bubble chart uses only y and z coordinate, which means that bubbles will have different size, but at the same time the distance between them will be the same (according to the **distance** parameter)—the number of teachers will not be considered.

Country	Teachers	Graduates	Area
Bulgaria	22.223	56.851	110.370
Estonia	4.129	9.577	45.227
Croatia	16.625	33.989	56.594
Lithuania	12.148	27.684	65.286
Latvia	7.022	14.587	64.573
Hungary	24.119	63.109	93.011
Slovakia	12.204	50.622	49.035
Slovenia	7.178	16.458	20.273

Table 5.1 Number of teachers (in thousands), graduates (in thousands) and area (in thousands km^2) in 2017 in chosen countries (Eurostat, 2020bcd)

To set the bubble chart attributes, the parameters in *Chapter* can be used. The *Chart 5.1* was created by using the following commands—see the parameters set (changed from default).

```
\def\educountriesdataone{{22.223,4.129,
   16.625,12.148,7.022,24.119,12.204,
   7.178},{56.851,9.577,33.989,27.684,
   14.587,63.109,50.622,16.458},{
   110.370,45.227,56.594,65.286,64.573,
   93.011,49.035,20.273}}

\bubblechart[basic][
   gridcolor=gray,
   xscale=0.48, yscale=0.1,
   gridline=dashed,
   xgridlinesdensity=0.5,
   bubblepalette=lightgreenpalette,
   bubblediameter=0.03,
   xcoor=yes, zcoor=yes,
   xaxislabinit=10, xaxislabinc=10,
   axesunits=yes,
   xunit=Number of teachers (thousands),
   yunit=Number of graduates (thousands)][
   method=struct,
   data={\educountriesdataone}]
```

Country	Year	Teachers	Graduates	Area
Bulgaria	2014	23.012	63.373	110.370
	2017	22.223	56.851	110.370
Estonia	2014	4.708	10.190	45.227
	2017	4.129	9.577	45.227
Croatia	2014	16.842	35.191	56.594
	2017	16.625	33.989	56.594
Lithuania	2014	14.067	33.130	65.286
	2017	12.148	27.684	65.286
Latvia	2014	6.894	17.345	64.573
	2017	7.022	14.587	64.573
Hungary	2014	21.778	72.465	93.011
	2017	24.119	63.109	93.011
Slovakia	2014	12.798	66.194	49.035
	2017	12.204	50.622	49.035
Slovenia	2014	7.142	18.824	20.273
	2017	7.178	16.458	20.273

Table 5.2 Number of teachers (in thousands), graduates (in thousands) and area (in thousands km^2) in 2014 and 2017 in chosen countries (Eurostat, 2020bcd)

The *Chart 5.2* was created by using the following commands—see the parameters set (changed from default).

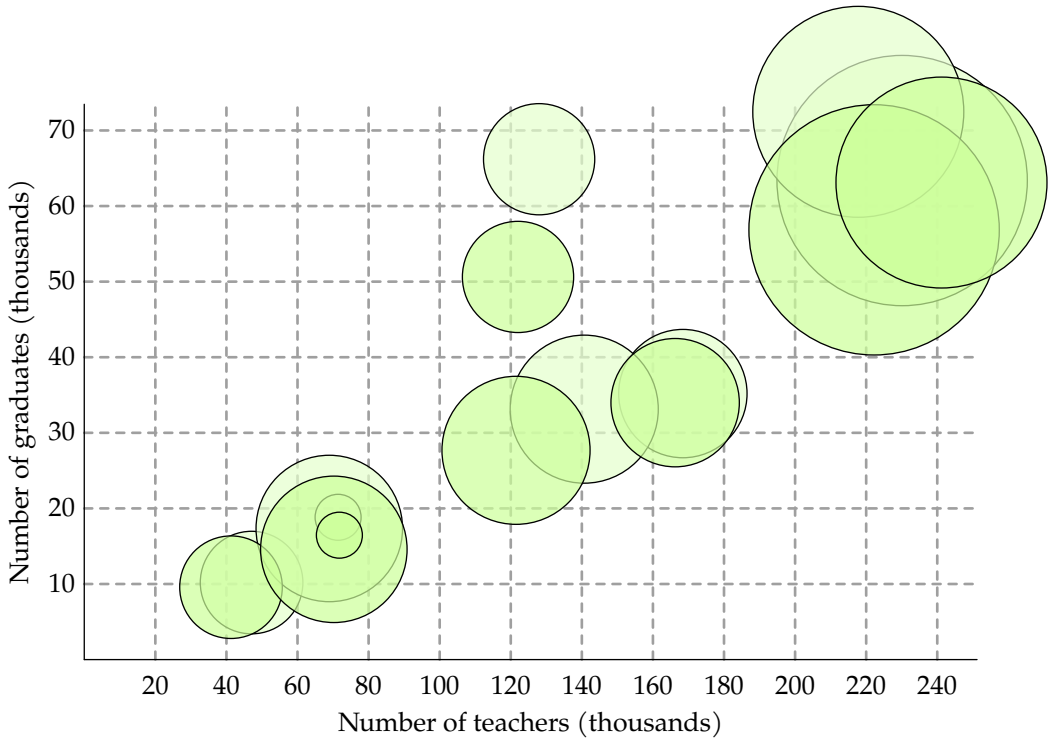

Chart 5.2 Bubble chart with x-, y- and z-coordinate with multiple data series

```
\def\educountriesdatafour{{23.012,
   4.708,16.842,14.067,6.894,21.778,
   12.798,7.142},{63.373,10.190,35.191,
   33.130,17.345,72.465,66.194,18.824
   },{110.370,45.227,56.594,65.286,
   64.573,93.011,49.035,20.273},{22.223,
   4.129,16.625,12.148,7.022,24.119,
   12.204,7.178},{56.851,9.577,33.989,
   27.684,14.587,63.109,50.622,16.458
   },{110.370,45.227,56.594,65.286,
   64.573,93.011,49.035,20.273}}

\bubblechart[basic][distance=2,
   xscale=0.47, yscale=0.1,
   gridline=dashed,
   xgridlinesdensity=0.5,
   bubblepalette=lightgreenpalette,
   bubblediameter=0.03,
   xcoor=yes, zcoor=yes,
   xaxislabinit=10, xaxislabinc=10,
   axesunits=yes,
   xunit=Number of teachers (thousands),
   yunit=Number of graduates (thousands)][
   method=struct,
   data={\educountriesdatafour}]
```

The *Chart 5.3* was created by using the commands below—see the parameters set (changed from default).

```
\def\educountriesdatatwo{{22.223,
   4.129,16.625,12.148,7.022,24.119,
   12.204,7.178},{56.851,9.577,33.989,
   27.684,14.587,63.109,50.622,16.458}}

\bubblechart[basic][
   distance=2,
   xscale=0.48, yscale=0.1,
   gridline=dashed,
   xgridlinesdensity=0.5,
   bubblepalette=lightgreenpalette,
   bubblediameter=1.5,
   xcoor=yes, zcoor=no,
   xaxislabinit=10, xaxislabinc=10,
   axesunits=yes,
   xunit=Number of teachers (thousands),
   yunit=Number of graduates (thousands)][
   method=struct,
   data={\educountriesdatatwo}]
```

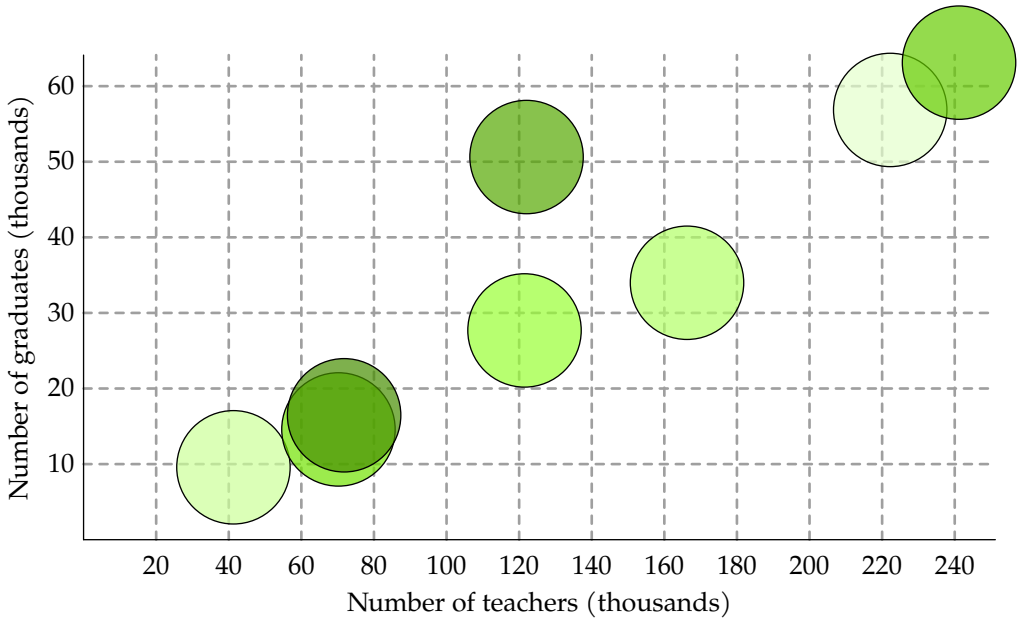

Chart 5.3 Bubble chart with *x*- and *y*-coordinate with one data series

The *Chart 5.4* was created by using the following commands—see the parameters set (changed from default).

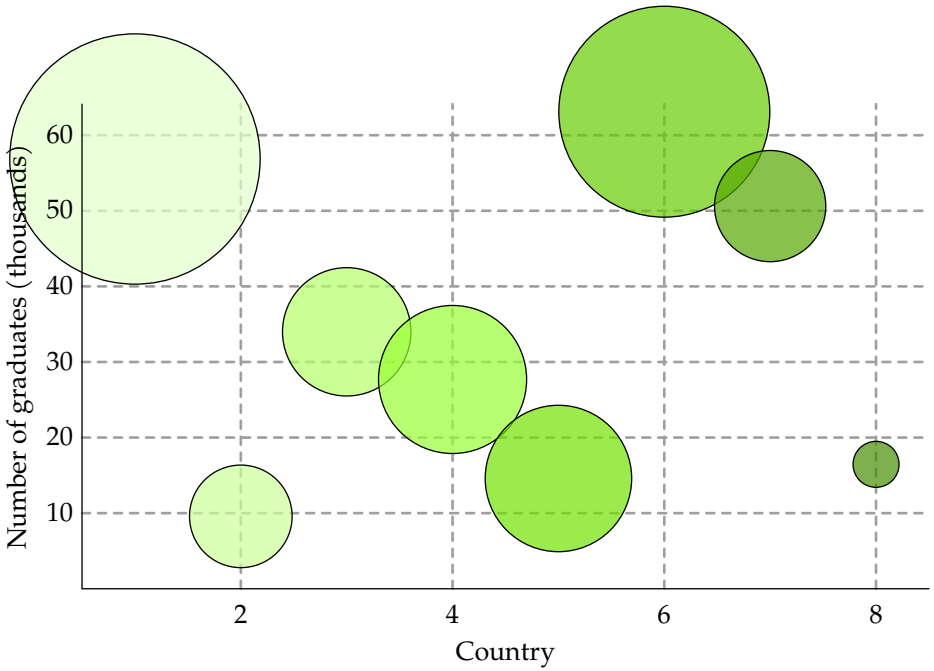

Chart 5.4 Bubble chart with *x*- and *z*-coordinate with one data series

```
\def\educountriesdatathree{{56.851,
   9.577,33.989,27.684,14.587,63.109,
   50.622,16.458},{110.370,45.227,
   56.594,65.286,64.573,93.011,49.035,
   20.273}}

\bubblechart[basic][
   distance=2,
   xscale=0.7, yscale=0.1,
   gridline=dashed,
   xgridlinesdensity=0.5,
   bubblepalette=lightgreenpalette,
   bubblediameter=0.03,
   zcoor=yes,
   xaxislabinit=10, xaxislabinc=10,
   axesunits=yes,
   xunit=Country,
   yunit=Number of graduates (thousands)][
   method=struct,
   data={\educountriesdatathree}]
```

6 Column charts

The column chart is often referred to as the simplest and most straightforward from the wide spectrum of all chart types. It is basically almost the same as the bar chart, except the axes are swapped. According to Lile (2017), the universal column chart contains a series of bars which lengths vary. In this type of chart, the horizontal axis represents the categories, the author wants to compare. The second (vertical) axis symbolizes the values of each category. This type of chart can be represented by four different subtypes which are:

- basic,
- clustered,
- stacked and
- 100% stacked.

Column chart parameters

The following parameters (and their default values) can be set in the column chart.

To work with the axes:

xlength: minimum length needed according to data + left
ylength: minimum length needed according to data (3 in 100% stacked column chart)
xshift: 0
yshift: 0
xwidth: 0.5
ywidth: 0.5

To scale the entire chart:

xscale: 1
yscale: 1

To set the unit:

unit: cm

To set the position of the initial and all other values:

distance: 1

left: 1

To adjust the grid:

grid: on
gridcolor: middlegray
gridline: dotted
gridlinewidth: 1
ygridlinesdensity: 1
xgridlines: yes
ygridlines: yes
xgridlinesnoval: yes

To set the columns:

columncolor: gray
columnpalette:
columntransparency: 1
columnwidth: 1
contour: yes
contourcolor: black
contourpalette:
contourwidth: 0.5

To set the labels:

xaxislabels: yes
yaxislabels: yes
axesunits: yes
xunit:
xunit:
xunit: yes
decimals: 1
rounding: halfup

equaldecimals: no

6.1 Basic column chart

This subtype of column chart is the simplest and can be used only in case of working with one-dimensional data (or one series of data). For example (*Chart 6.1*), the horizontal axis could be used for representing age groups, while the vertical axis would display the unemployment in thousand persons.

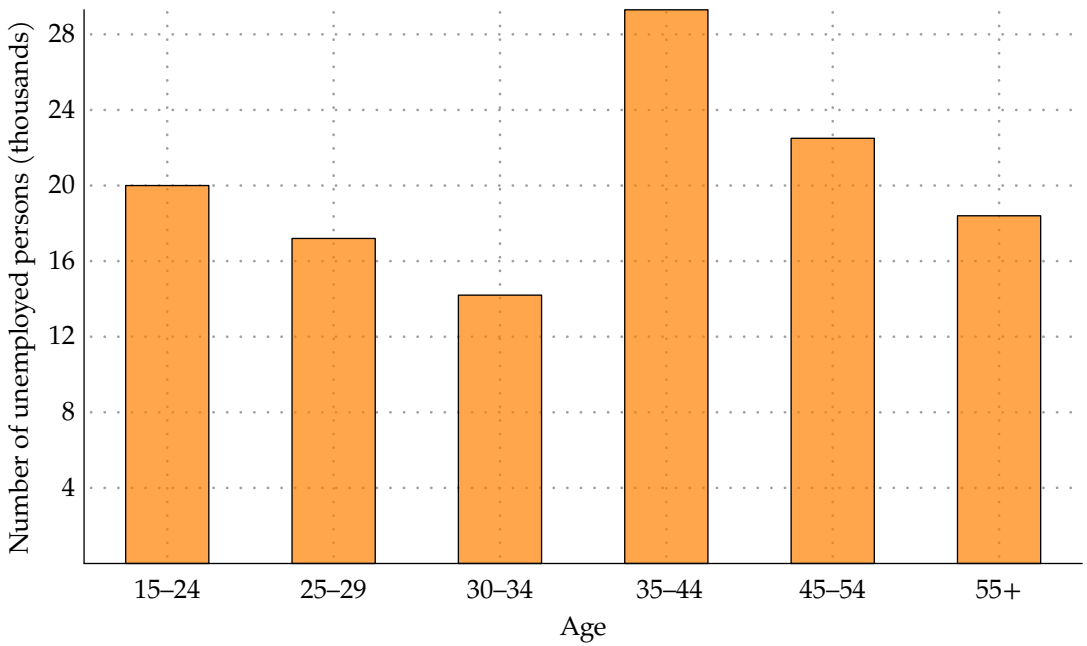

Chart 6.1 Basic column chart with one data series

The data used are the same as in the basic bar chart example (*Chart 4.1*)—see *Table 4.1*). Each column will have a different height according to the number of unemployed in the specific age group.

To set the column chart attributes, the parameters in *Chapter* can be used. Even though basic column chart works with just one-dimensional data series, even if multiple series will be inserted, the command will work but only the first one data series will be shown. The *Chart 6.1* was created by using the commands below—see the parameters set (changed from default).

```
\def\unemploydataone{20,17.2,14.2,29.3,
   22.5,18.4}

\def\agelabels{15--24,25--29,30--34,
   35--44,45--54,55+}

\columnchart[basic][
   xscale=1.1, yscale=0.25,
   left=0.5,
   columncolor=tamarange,
   columntransparency=0.7,
   axesunits=yes,
   xunit=Age,
   yunit=Number of unemployed
              persons (thousands)][
   data={\unemploydataone},
   xlabels={\agelabels}]
```

6.2 Clustered column chart

This type of column chart can be very effectively used to compare data when there is more than one series (Keynote Support, c2020). The advantage of this type is that not only the values within one series can be compared, but also the comparison between multiple series in one category can be easily visible. To show an example, the same data were used. However, more than one series is needed, therefore the data for each quarter were displayed separately. See the (*Chart 6.2*) and the data used in *Table 4.2*.

The *Chart 6.2* was created by using the commands below—see the parameters set (changed from default).

```
\def\unemploydatatwo{{18.9,16.3,14,35.5,
   25,19.9},{20.1,17.6,15.3,16.5,20.6,
   18.2},{23,19.4,13.9,29.4,22.4,19.2},{
   17.9,15.3,13.7,25.8,22,16.4}}

\def\agelabels{15--24,25--29,30--34,
   35--44,45--54,55+}

\columnchart[clustered][
   xscale=0.44, yscale=0.25,
   left=0,
```

```
columnpalette=redpalette,
columntransparency=0.7,
axesunits=yes,
xunit=Age,
yunit=Number of unemployed
            persons (thousands)][
method=struct,
data={\unemploydatatwo},
xlabels={\agelabels}]
```

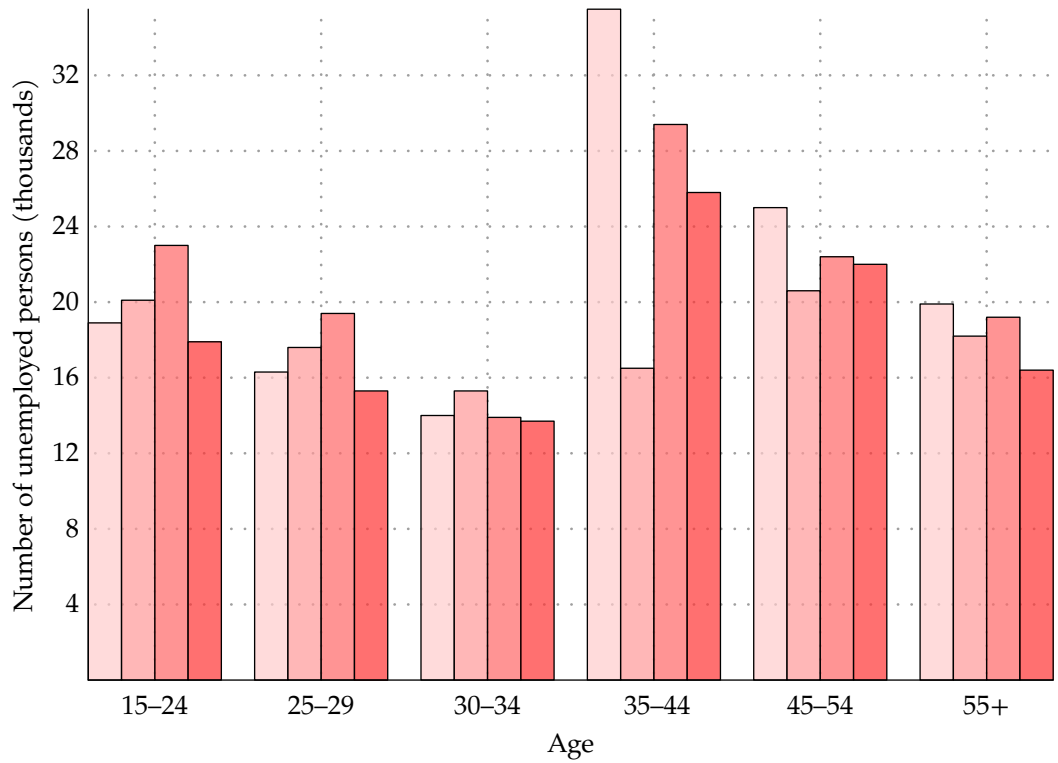

Chart 6.2 Clustered column chart with one data series

6.3 Stacked column chart

Stacked column chart can be used only when multiple series are displayed. The benefit of this type is that the author can put focus on the total of all series in one category (Rost, 2018).

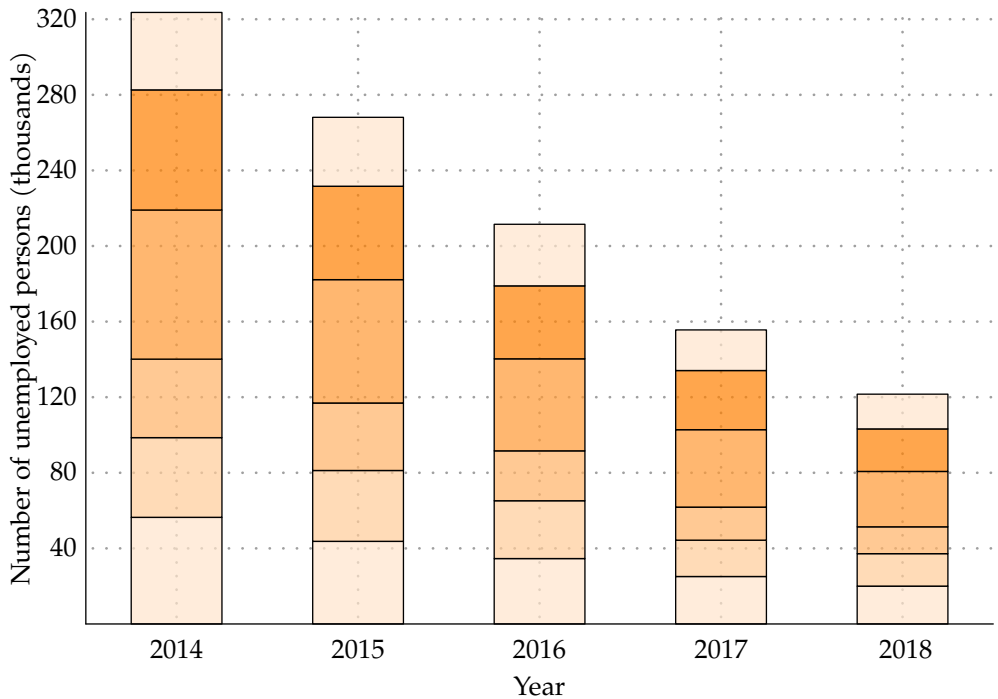

Chart 6.3 Stacked column chart with one data series

For example, this chart can be used to visualize the total unemployment in years 2014–2018 and at the same time, to distinguish between the numbers of each age group. The data in *Table 4.3* were used. The *Chart 6.3* was created by using the commands below—see the parameters set (changed from default).

```
\def\unemploydatathree{{56.4,42.2,41.5,
   78.9,63.6,41},{43.7,37.5,35.7,65.3,
   49.4,36.5},{34.6,30.6,26.4,48.7,38.6,
   32.6},{25.1,19.2,17.5,41,31.3,21.5},{
   20,17.2,14.2,29.3,22.5,18.4}}

\def\yearlabelsone{2014,2015,2016,2017,
   2018}

\columnchart[stacked][
   xscale=1.2, yscale=0.025,
   left=0.5,
   columnpalette=orangepalette,
   columntransparency=0.7,
   axesunits=yes,
   xunit=Year,
   yunit=Number of unemployed persons (thousands)][
   method=struct,
   data={\unemploydatathree}, c=r,
   xlabels={\yearlabelsone}]
```

6.4 100% stacked column chart

According to Keynote Support (c2008–2020), in 100% stacked column chart, values of all series within one category are expressed as a portion of 100%. This chart is suitable when the author wants to show the change in time within the categories of a whole (amCharts, c2006–2020). Except of displaying the values in relative way, this type of chart is very similar to the stacked column chart. See the example with the same data as in *Chart 6.4* chart—*Table 4.3*.

To manipulate with 100% stacked column chart, in addition to the parameters listed in *Chapter* and explained in *Chapter 2*—except the **yscale** parameter, also the following one (and its default value) can be used.

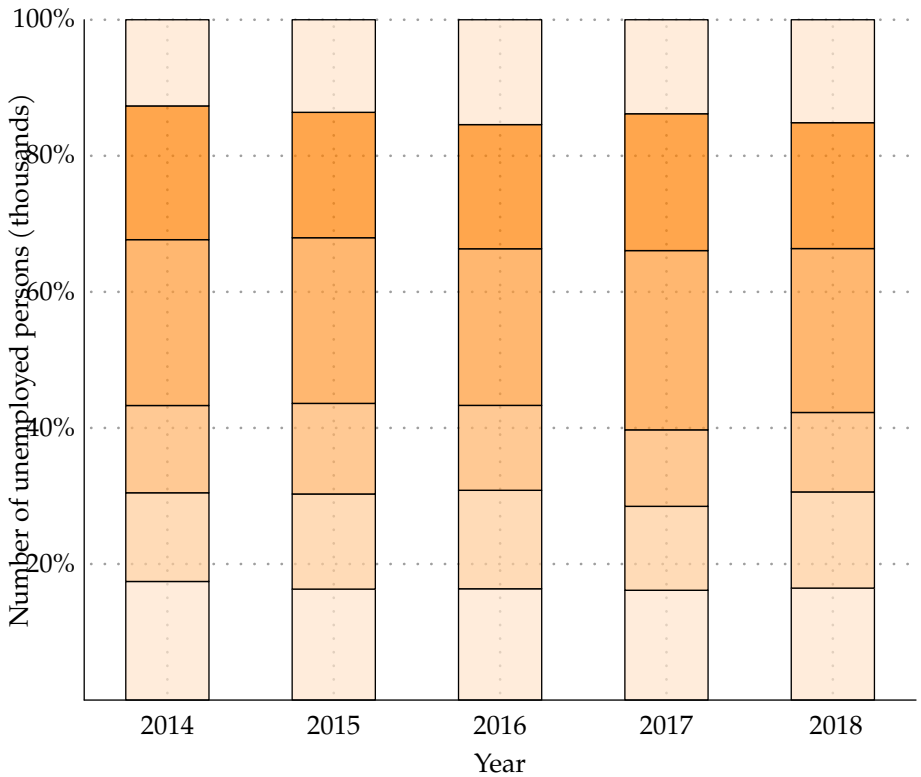

Chart 6.4 100% stacked column chart with one data series

To set the number of *y*-axis labels:

ylabelscount: 5

The *Chart 6.4* was created by using the following commands—see the parameters set (changed from default).

```
\def\unemploydatathree{{56.4,42.2,41.5,
   78.9,63.6,41},{43.7,37.5,35.7,65.3,
   49.4,36.5},{34.6,30.6,26.4,48.7,38.6,
   32.6},{25.1,19.2,17.5,41,31.3,21.5},{
   20,17.2,14.2,29.3,22.5,18.4}}

\def\yearlabelsone{2014,2015,2016,2017,
   2018}
```

```
\columnchart[stacked100][
   ylength=9,
   xscale=1.1,
   left=0.5,
   ylabelscount=5,
   columnpalette=orangepalette,
   columntransparency=0.7,
   axesunits=yes,
   xunit=Year,
   yunit=Number of unemployed persons (thousands)][
   method=struct,
   data={\unemploydatathree}, c=r,
   xlabels={\yearlabelsone}]
```

7 Line charts

As well as the column chart, the axes of the line chart have the same meaning, where the x-axis is used for categories/segments and the y-axis marks the values or percentages. This type differs from column and bar charts in the data because it displays dependent data in different time periods, whereas column and bar charts can also show the separate data. For example, column and bar chart can represent the annual sales of various companies within one industry (and obviously, those values do not depend on each other). Hence, the line chart is commonly used to show data trends (SkillsYouNeed, c2011–2020). The trend is expressed by the line which connects all the values. As well as the types above, the line chart also has its subtypes:

- basic,
- stacked and
- 100% stacked.

Line chart parameters

The following parameters (and their default values) can be set in the line chart.

To work with the axes:

xlength: minimum length needed according to data + left
ylength: minimum length needed according to data + 1 unit (3 in 100% stacked line chart)
xshift: 0
yshift: 0
xwidth: 0.5
ywidth: 0.5

To scale the entire chart:

xscale: 1
yscale: 1

To set the unit:

unit: cm

To set the position of the initial and all other values:

distance: 1

left: 1

To adjust the grid:

grid: on
gridcolor: middlegray
gridline: dotted
gridlinewidth: 1
ygridlinesdensity: 1
xgridlines: yes
ygridlines: yes
xgridlinesnoval: yes
xgridlinesleft: yes

To set the line:

linecolor: gray
linepalette:
linewidth: 1

To set the dots:

dots: yes
dotscolor: black
dotspalette:
dotswidth: 5

To set the labels:

xaxislabels: yes
yaxislabels: yes
axesunits: yes
xunit:
xunit:
xunit: yes
dotslabels: no
decimals: 1
rounding: halfup
equaldecimals: no

7.1 Basic line chart

The basic line chart can be displayed with or without the markers. In comparison to basic column and bar chart, the basic line chart shows both one-dimensional or two-dimensional data. See the examples of a basic line chart with markers, where just one series is used (*Chart 7.1*) and where multiple series are used (*Chart 7.2*). The data used describe the number of bachelor graduates in years 2013–2017 in Bulgaria in *Chart 7.1* and in Bulgaria and Ireland in (*Chart 7.2*) chart—see data in *Table 7.1* and *Table 7.2*.

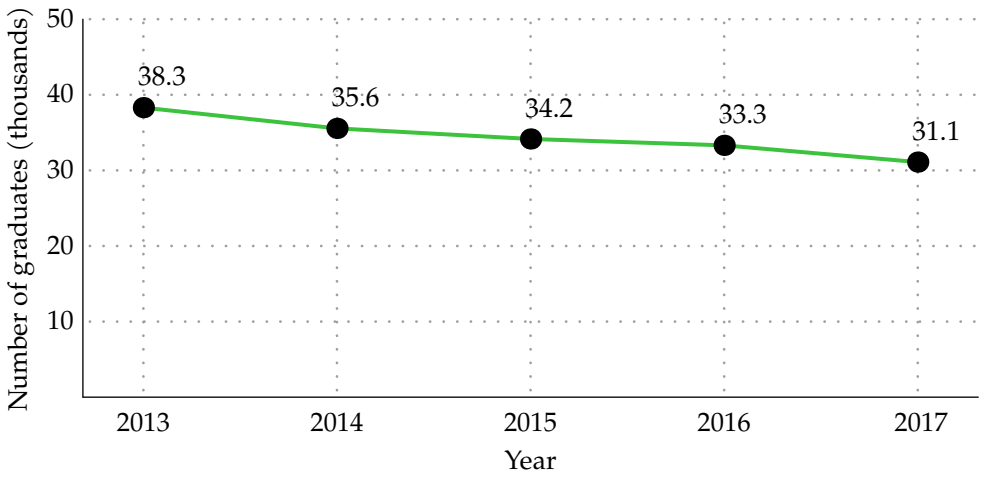

Chart 7.1 Basic line chart with one data series

To set the line chart attributes, the parameters in *Chapter* can be used. The *Chart 7.1* was created by using the commands below—see the parameters set (changed from default).

```
\def\edudataone{38.303,35.556,34.158,
   33.313,31.101}

\def\yearlabelstwo{2013,2014,2015,2016,
   2017}

\linechart[basic][
   ylength=5,
   xscale=1.6, yscale=0.1,
   left=0.5,
   distance=1.6,
   linecolor=limegreen, linewidth=1.5,
   dotscolor=tomgreen,
   dotswidth=8, dotslabels=yes,
   axesunits=yes,
   xunit=Year,
   yunit=Number of graduates (thousands)][
   data={\edudataone},
   xlabels={\yearlabelstwo}]
```

Year	2013	2014	2015	2016	2017
Bachelor graduates	38.303	35.556	34.158	33.313	31.101

Table 7.1 Number of bachelor graduates in thousands in 2013–2017 in Bulgaria (Eurostat, 2020c)

Year	2013	2014	2015	2016	2017
Bulgaria	38.303	35.556	34.158	33.313	31.101
Ireland	29.391	40.856	43.036	44.287	44.485

Table 7.2 Number of bachelor graduates in thousands in year 2013–2017 in Bulgaria and Ireland (Eurostat, 2020c)

The *Chart 7.2* was created by using the commands below—see the parameters set (changed from default).

```
\def\edudatatwo{{38.303,35.556,34.158,
    33.313,31.101},{29.391,40.856,43.036,
    44.287,44.485}}

\def\yearlabelstwo{2013,2014,2015,2016,
    2017}
```

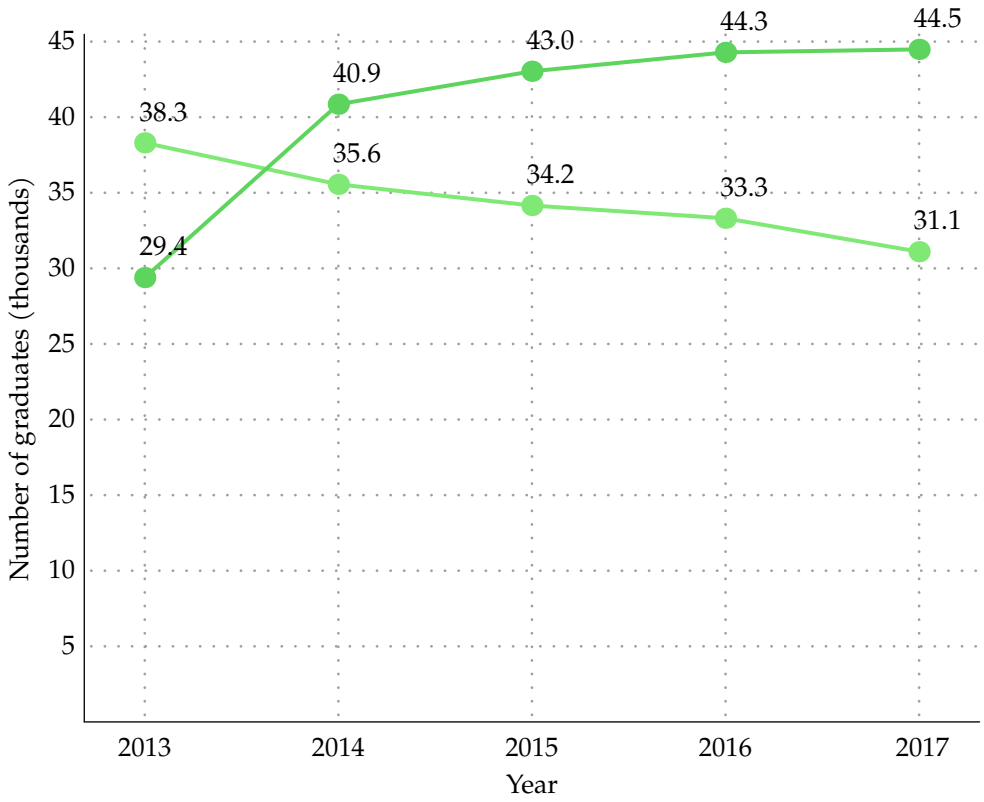

Chart 7.2 Basic line chart with multiple data series

7.2 Stacked line chart

Unlike in the basic line chart with more data series used, the lines in the stacked line chart do not cross each other because they are displayed in a cumulative way. Therefore, the data must occur in either positive or negative values.

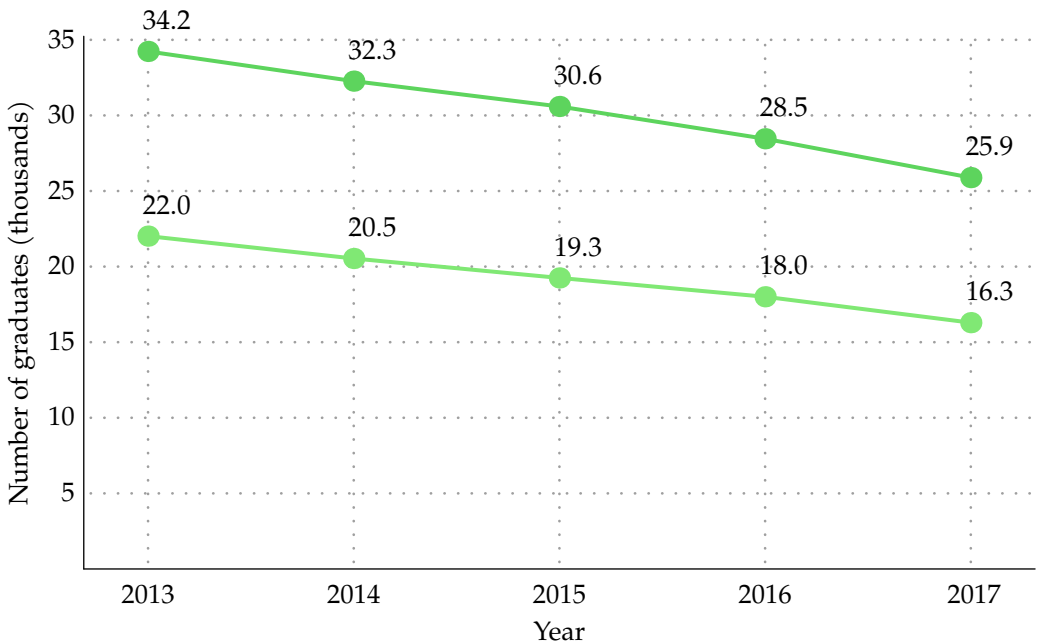

Chart 7.3 Stacked line chart with multiple data series

The stacked line chart always represents more than one series as line-connected sets of points (Infragistics, c2008–2011). The example (*Chart 7.3*) shows the number of female and male master graduates in 2013–2017 in Slovakia—see the data *Table 7.3*.

The *Chart 7.3* was created by using the commands below—see the parameters set (changed from default).

```
\def\edudatathree{{22.017,20.536,19.251,
   18.005,16.292},{12.216,11.732,11.341,
   10.451,9.595}}

\def\yearlabelstwo{2013,2014,2015,2016,
   2017}

\linechart[stacked][
   xscale=1.7, yscale=0.2,
   distance=1.6,
   left=0.5,
   linepalette=darkgreenpalette,
   linewidth=1.5,
   dotspalette=darkgreenpalette,
   dotswidth=8, dotslabels=yes,
   axesunits=yes,
   xunit=Year,
   yunit=Number of graduates (thousands)][
   method=struct,
   data={\edudatathree},
   xlabels={\yearlabelstwo}]
```

100% stacked line chart

This type of chart is similar to the stacked line chart, but in addition, the values are represented in a relative way and for each point they make up 100% all together (Infragistics, c2008–2011). This chart also shows the contribution of each series at a certain point. The example (*Chart 7.4*) is similar to the stacked line chart (*Chart 7.3*)—the data are the same (in *Table 7.3*), except they are displayed in a relative way, so the ratio between females and males can be easily seen.

To manipulate with 100% stacked line chart, in addition to the parameters listed in *Chapter* and explained in *Chapter 2*—except the **yscale** parameter, also the following one (and its default value) can be used.

To set the number of *y*-axis labels:

ylabelscount: 5

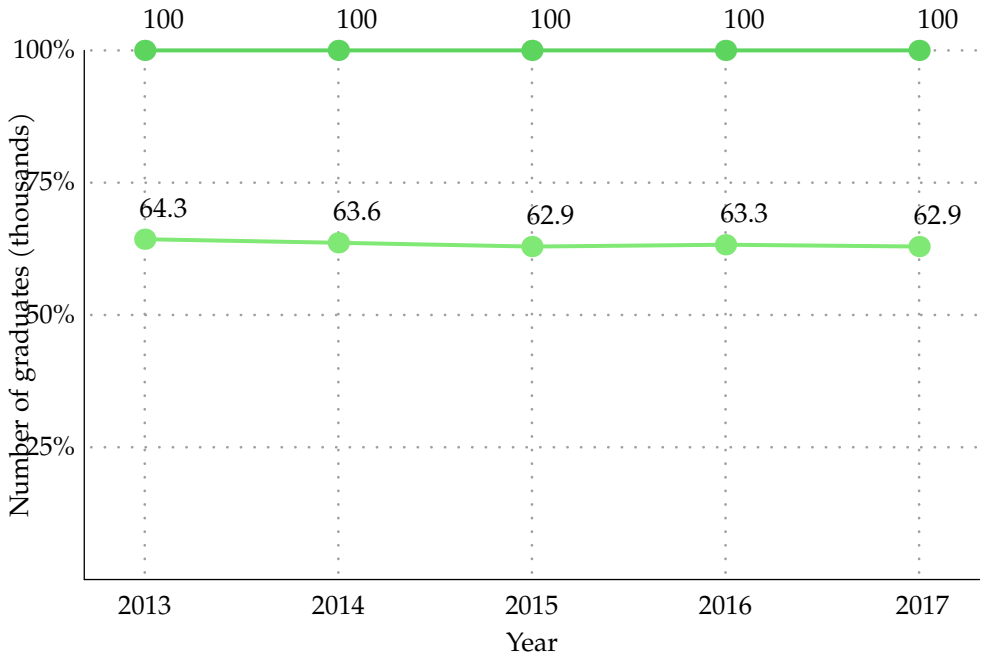

Chart 7.4 100% stacked line with multiple data series

Year	2013	2014	2015	2016	2017
Females	22.017	20.536	19.251	18.005	16.292
Males	12.216	11.732	11.341	10.451	9.595

Table 7.3 Number of master graduates according to sex in thousands in 2013–2017 in Slovakia (Eurostat, 2020c)

The *Chart 7.4* was created by using the commands below—see the parameters set (changed from default).

```
\def\edudatathree{{22.017,20.536,19.251,
   18.005,16.292},{12.216,11.732,11.341,
   10.451,9.595}}

\def\yearlabelstwo{2013,2014,2015,2016,
   2017}

\linechart[stacked100][
   ylength=7,
   xscale=1.6, yscale=0.22,
   ylabelscount=4,
   distance=1.6,
   left=0.5,
   linepalette=darkgreenpalette,
   linewidth=1.5,
   dotscolor=tomgreen,
   dotspalette=darkgreenpalette,
   dotswidth=8, dotslabels=yes,
   axesunits=yes,
   xunit=Year,
   yunit=Number of graduates (thousands)][
   method=struct,
   data={\edudatathree},
   xlabels={\yearlabelstwo}]
```

8

Pie charts

There are two variations of the pie chart:

- pie and
- doughnut

which have essentially the same function—to show the data in the relative form. The more common one is probably the pie chart itself, but if more than one series needs to be shown, then it is necessary to use the doughnut chart.

Pie chart parameters

The following parameters (and their default values) can be set in the pie chart.

To set the diameters:

diameter: 5

To set the unit:

unit: cm

To set the parts:

fillcolor:
fillcolor: orange
contour: yes
contourcolor: black
contourpalette:
contourwidth: 0.5

To set the labels:

labels: yes
perc: no
decimals: 0.1
coef: 0
decimals: 1
rounding: halfup
equaldecimals: no

8.1 Pie chart

Lile (2017) mentioned that visualizing data trough the pie chart is the simplest and the most effective way to compare particular parts of a whole. The pie chart is represented by a circle which is divided into parts—illustrating the data. The parts must be either labelled or the chart must contain a legend, explaining the meaning of each part. The example (*Chart 8.1*) illustrates the average time spent on shopping and services by men—see the data used in *Table 8.1*. The country with the most minutes spent on those activities is emphasized by the corresponding part being shifted out.

To manipulate with pie chart, in addition to the parameters listed in *Chapter* and explained in *Chapter 2*, also the following ones (and their default values) can be used.

To set the parts which will be out:

out:
distanceout: 0.2

To set the label position:

labelposition: out

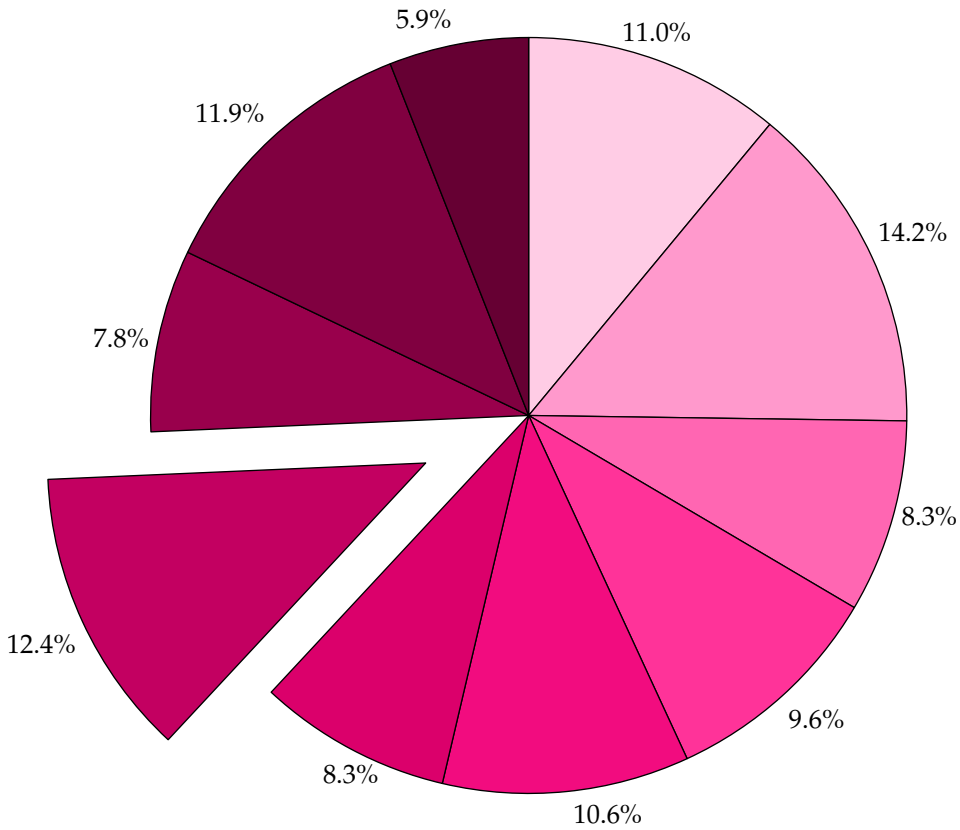

Chart 8.1 Pie chart with one data series

The *Chart 8.1* was created by using the commands below—see the parameters set (changed from default).

```
\def\timespentdataone{24,31,18,21,23,18,
   27,17,26,13}

\piechart[basic][
   diameter=10,
```

```
perc=yes,
fillpalette=pinkpalette,
distanceout=0.3, out={7}][
data={\timespentdataone}]
```

Countries	Time
Belgium	24
Germany	31
Greece	18
Spain	21
France	23
Hungary	18
Netherlands	27
Romania	17
Finland	26
Turkey	13

Table 8.1 Time spent on shopping and services (average per day) by men in minutes in 2010 in chosen countries (Eurostat, 2020e)

8.2 Doughnut chart

A doughnut chart is a form of the pie chart, but it usually has a blank space inside of the circle. The benefit of this chart is that it allows the user to display more than one data series, which makes it easy to compare the series with each other. If there are too many categories, this chart should not be used, because it can simply get chaotic. See the example in *Chart 8.2*, which uses similar data as the previous example (*Chart 8.1*), but one more series is used (the average time spent by women). The data are shown in *Table 8.2*.

To manipulate with doughnut chart, in addition to the parameters listed in *Chapter* and explained in *Chapter 2*, also the following ones (and their default values) can be used.

To set the hole diameter:

holediameter: 3

To set the distance between the doughnuts:

distancebetween: 0

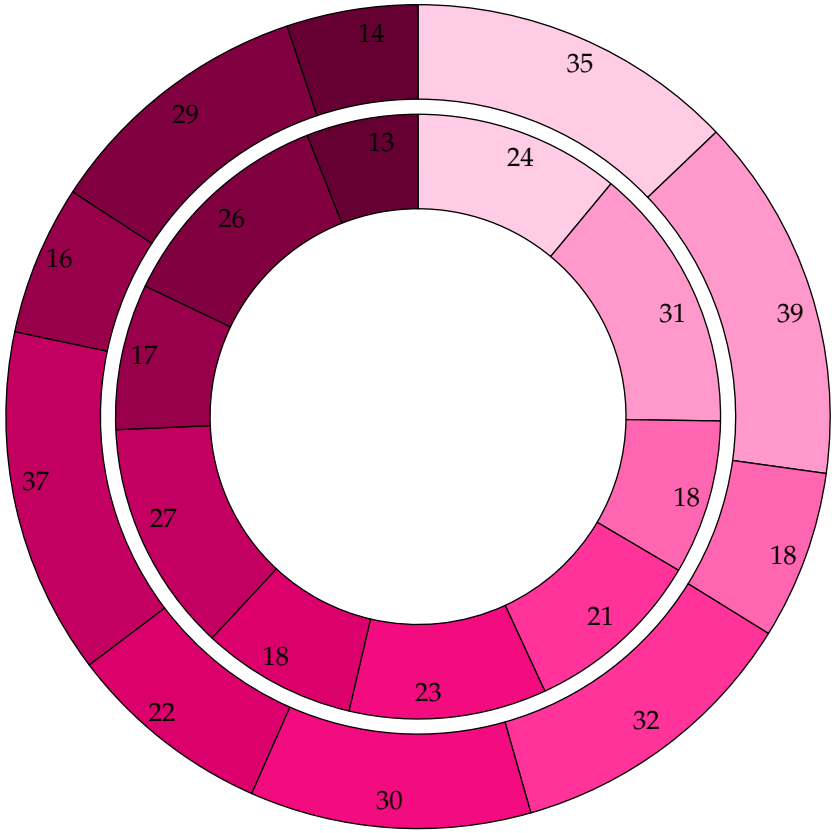

Chart 8.2 Doughnut chart with multiple data series

The *Chart 8.2* was created by using the commands below—see the parameters set (changed from default).

```
\def\timespentdatatwo{{24,31,18,21,23,
   18,27,17,26,13},{35,39,18,32,30,22,
   37,16,29,14}}

\piechart[doughnut][
   diameter=8, holediameter=5.5,
   fillpalette=pinkpalette,
   distancebetween=0.2][
   method=struct,
   data={\timespentdatatwo}]
```

Countries	Males	Females
Belgium	24	35
Germany	31	39
Greece	18	18
Spain	21	32
France	23	30
Hungary	18	22
Netherlands	27	37
Romania	17	16
Finland	26	29
Turkey	13	14

Table 8.2 Time spent on shopping and services (average per day) by men and women in minutes in 2010 (EUROSTAT, 2020e)

9 Radar charts

This type of chart is suitable for visualizing data with multiple variables. The coordinate system of a radar chart, often called a net or spider chart, consists of regular polygons with axes emanating from the center. The number of polygon vertices is given by the number of data (in each series if there is more than one). Each data series is displayed in form of a polygon and in case of multiple series, the polygons overlay each other. Therefore, the reader can see the differences (shown by shape and size of polygons) between the data series very clearly (Nowicki and Merenstein, 2016).

Radar chart parameters

The following parameters (and their default values) can be set in the radar chart.

To set the axes:

axes: yes
axiswidth: 0.5
axisline: dashed
axiscolor: middlegray

To set the unit:

unit: cm

To set the distance between the layers:

distance: 2

To work with the layers:

step: 2
layers: minimum number of layers needed according to data
layercolor: black
layerline: dashed
layerwidth: 0.5

To set the line:

linecolor: black
linepalette:
linewidth: 1

To set the fill:

fill: no
fillcolor: gray
fillpalette:
filltransparency: 1

To set the dots:

dotscolor: black
dotspalette:
dotswidth: 5

To set the labels:

layerslabels: yes
dotslabels: no
decimals: 1

rounding: halfup
equaldecimals: no

9.1 Radar chart

The first example of radar chart (*Chart 9.1*) shows the total government expenditure according to specific classifications in 2018 in Spain—see the data used in *Table 9.1*.

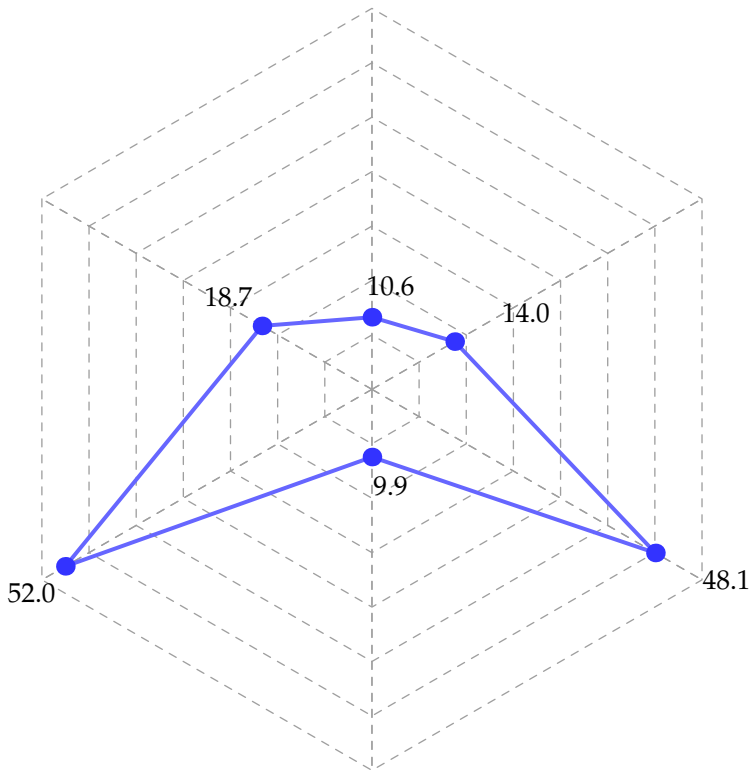

Chart 9.1 Radar chart with one data series

To see the difference, the second example (*Chart 9.2*) uses multiple data series—with Belgium and Poland as additional countries (see the data in *Table 9.2*).

To set the radar chart attributes, the parameters in *Chapter* can be used. The *Chart 9.1* was created by using the commands below—see the parameters set (changed from default).

```
\def\expendituredataone{10.6040,14.0410,
   48.0950,9.9460,52.0170,18.6530}

\radarchart[basic][
   layercolor=middlegray,
   distance=0.72,
   step=8,
   layerslabels=no,
   linecolor=mediumslateblue,
   linewidth=1.5,
   dotscolor=neonblue,
   dotswidth=7, dotslabels=yes,
   equaldecimals=yes][
   data={\expendituredataone}]
```

Classification	Spain
Environmental protection	10.6040
Police services	14.0410
Education	48.0950
Family and children	9.9460
Health	52.0170
Transport	18.6530

Table 9.1 Total general government expenditure according to classifications of the functions of government in billions in 2018 in Spain (Eurostat, 2020f)

Classification	Spain	Belgium	Poland
Environmental protection	10.6040	5.8602	2.4418
Police services	14.0410	4.6305	5.3627
Education	48.0950	28.6181	24.8754
Family and children	9.9460	9.9378	12.6925
Health	52.0170	34.9064	23.9082
Transport	18.6530	12.7486	17.7386

Table 9.2 Total general government expenditure according to classifications of the functions of government in billions in 2018 in Spain, Belgium and Poland (Eurostat, 2020f)

The *Chart 9.2* was created by using the following commands—see the parameters set (changed from default).

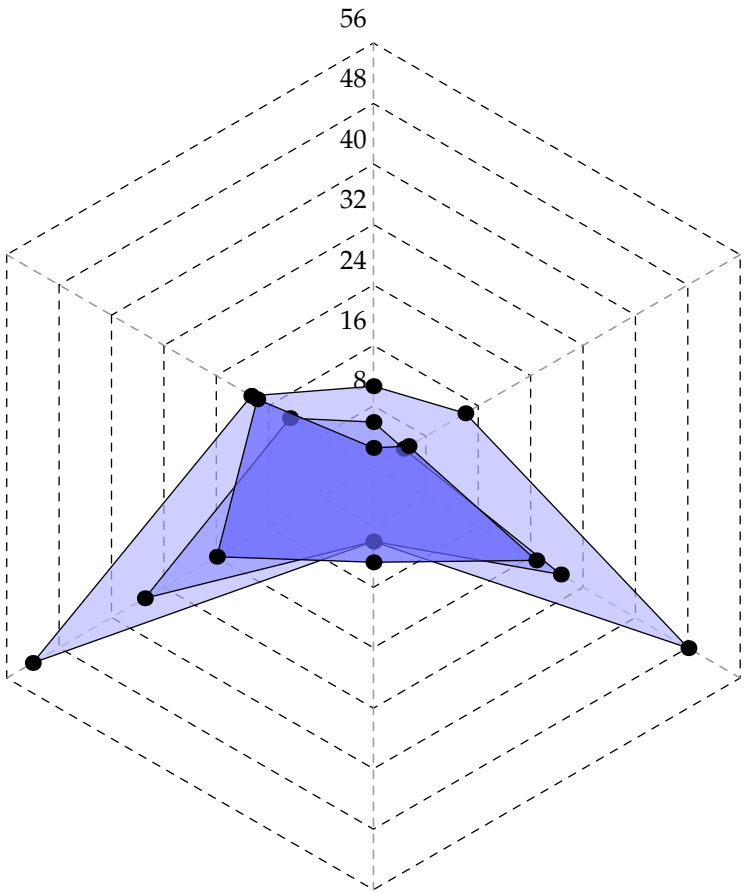

Chart 9.2 Radar chart with multiple data series

```
\def\expendituredatatwo{{10.6040,
   14.0410,48.0950,9.9460,52.0170,
   18.6530},{5.8602,4.6305,28.6181,
   9.9378,34.9064,12.7486},{2.4418,
   5.3627,24.8754,12.6925,23.9082,
   17.7386}}

\radarchart[basic][
   distance=0.8,
   step=8,
   fill=yes,
   linecolor=black, linewidth=0.5,
   fillpalette=indigopalette,
   filltransparency=0.7,
   dotswidth=6, dotslabels=no ][
   method=struct,
   data={\expendituredatatwo}]
```

10 Scatter charts

The scatter chart is useful to show a possible correlation between the variables (each represented by one axis). The presence of correlation is indicated by the position of points. If the points are distributed randomly, there is no correlation and if they remind a line or a curve, the correlation between the variables exists (Rouse, 1999–2020). The benefit of this chart is that the user can illustrate one or multiple data series. Depending on displaying the line and whether it is smooth or straight, various subtypes can be recognized:

- basic,
- with smooth lines and
- with straight lines.

Scatter chart parameters

The following parameters (and their default values) can be set in the scatter chart.

To work with the axes:

xlength: minimum length needed according to data + left
ylength: minimum length needed according to data + 1 unit
xshift: 0
yshift: 0
xwidth: 0.5
ywidth: 0.5

To scale the entire chart:

xscale: 1
yscale: 1

To set the unit:

unit: cm

To set the position of the initial and all other values:

distance: 1
left: 1

To adjust the grid:

grid: on
gridcolor: middlegray
gridline: dotted
gridlinewidth: 1
xgridlinesdensity: 1
ygridlinesdensity: 1
xgridlines: yes
ygridlines: yes
xgridlinesnoval: yes
xgridlinesleft: yes

To set the dots:

dots: yes
dotscolor: black
dotspalette:
dotswidth: 5

To set whether the x coordinate will be inserted:

xcoor: no

To set the initial and increment value:

xaxislabinit: 1
xaxislabinc: 1

To set the labels:

xaxislabels: yes
yaxislabels: yes
axesunits: yes
xunit:
xunit:
xunit: yes
dotslabels: no
decimals: 1
rounding: halfup
equaldecimals: no

10.1 Basic scatter chart

This type of scatter chart does not connect any plotted points—only the markers are shown. The basic scatter chart can be for example used for displaying dependence of monthly earnings on the real GDP in chosen years (*Chart 10.1*)—see the data in *Table 10.1*. The example of displaying multiple data series is in *Chart 10.2*, where another country was added (see *Table 10.2*).

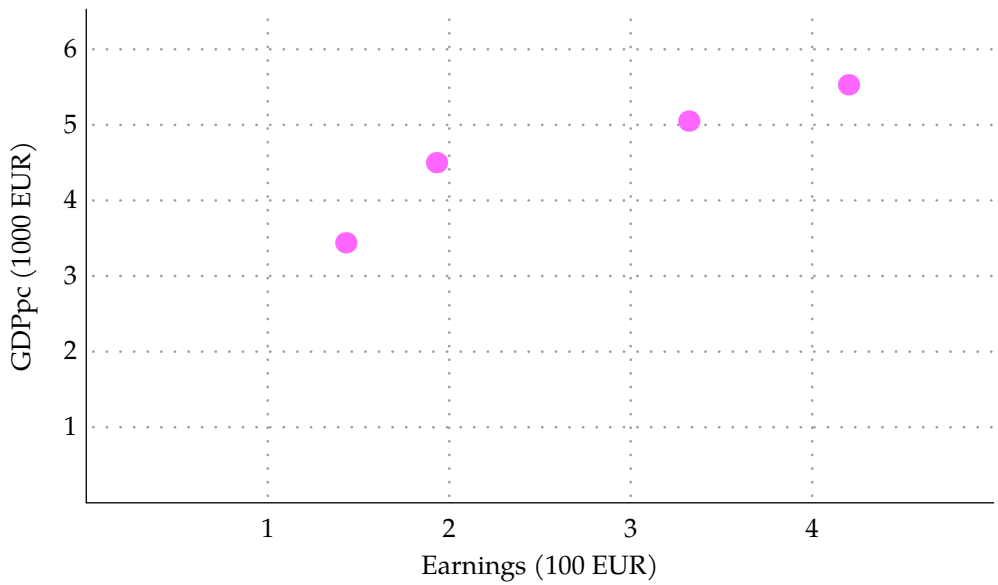

Chart 10.1 Basic scatter chart with one data series

Variable	2002	2006	2010	2014
GDP	3.44	4.50	5.05	5.53
Earnings	1.43	1.93	3.32	4.20

Table 10.1 Dependence of monthly earnings (in 100 EUR) on the real GDP (in 1000 EUR per capita) in years 2002, 2006, 2010 and 2014 in Bulgaria (EUROSTAT, 2020gh)

To set the scatter chart attributes, the parameters in *Chapter* can be used. The *Chart 10.1* was created by using the commands below—see the parameters set (changed from default).

```
\def\gdpwagesdataone{{1.43,1.93,3.32,
   4.20},{3.440,4.500,5.050,5.530}}

\scatterchart[basic][
   xlength=12,
   xscale=2.4, yscale=1,
   dotswidth=8,
   dotscolor=pinkflamingo,
   xcoor=yes,
   axesunits=yes,
   xunit=Earnings (100 EUR),
   yunit=GDPpc (1000 EUR)][
   method=struct,
   data={\gdpwagesdataone}]
```

Country	Variable	2002	2006	2010	2014
Bulgaria	GDP	3.44	4.50	5.05	5.53
	Earnings	1.43	1.93	3.32	4.20
Slovakia	GDP	8.42	10.80	12.54	13.62
	Earnings	3.37	5.19	7.58	9.08

Table 10.2 Dependence of monthly earnings (in 100 EUR) on the real GDP (in 1000 EUR per capita) in years 2002, 2006, 2010 and 2014 in Bulgaria and Slovakia (Eurostat, 2020gh)

The *Chart 10.2* was created by using the commands below—see the parameters set (changed from default).

```
\def\gdpwagesdatatwo{{1.43,1.93,3.32,
   4.20},{3.440,4.500,5.050,5.530},{
   3.37,5.19,7.58,9.08},{8.420,
   10.800,12.540,13.620}}

\scatterchart[basic][
   xscale=1.2, yscale=0.5,
   dotspalette=magentapalette,
   dotswidth=8,
   xcoor=yes,
   axesunits=yes,
   xunit=Earnings (100 EUR),
   yunit=GDPpc (1000 EUR)][
   method=struct,
   data={\gdpwagesdatatwo}]
```

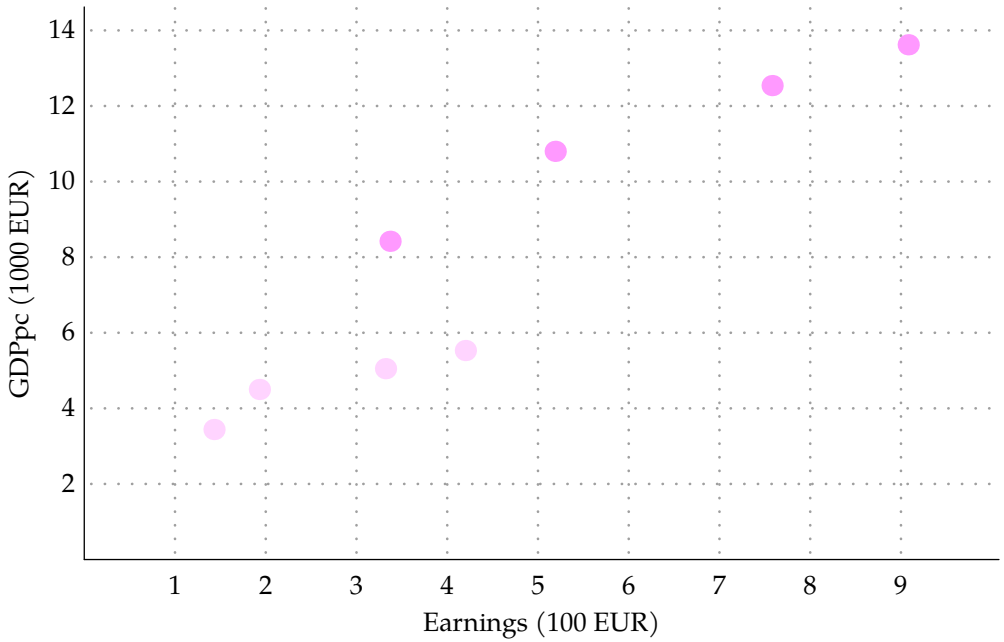

Chart 10.2 Basic scatter chart with multiple data series

10.2 Scatter chart with smooth lines

This subtype of scatter chart connects the points with smooth lines. The example can be seen in *Chart 10.3* where the data used are the same as in *Chart 10.2* (see *Table 10.2*).

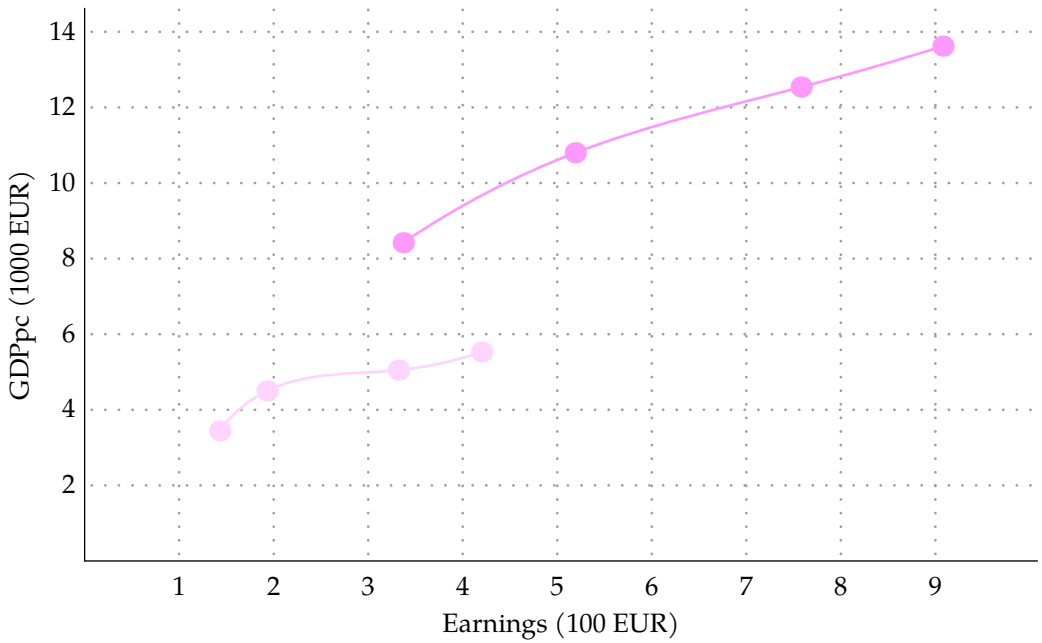

Chart 10.3 Scatter chart with smooth lines with multiple series

To manipulate with scatter chart with smooth lines, in addition to the parameters listed in *Chapter* and explained in *Chapter 2*, also the following ones (and their default values) can be used.

To set the line:

linecolor: gray
linepalette:
linewidth: 1

The chart in *Chart 10.3* was created by using the commands below—see the parameters set (changed from default).

```
\def\gdpwagesdatatwo{{1.43,1.93,3.32,
   4.20},{3.440,4.500,5.050,5.530},{
   3.37,5.19,7.58,9.08},{8.420,
   10.800,12.540,13.620}}

\scatterchart[smoothlines][
   xscale=1.25, yscale=0.5,
   linepalette=magentapalette,
   dotspalette=magentapalette,
   dotswidth=8,
   xcoor=yes,
   axesunits=yes,
   xunit=Earnings (100 EUR),
   yunit=GDPpc (1000 EUR)][
   method=struct,
```

```
data={\gdpwagesdatatwo}]
```

10.3 Scatter chart with straight lines

The only difference between the scatter chart with smooth lines and scatter chart with straight lines is that the lines, which connect the points, are either straight or smooth. See the example of this subtype in *Chart 10.4*—the data used are in *Table 10.2*.

To manipulate with scatter chart with straight lines, in addition to the parameters listed in *Chapter* and explained in *Chapter 2*, also the following ones (and their default values) can be used.

To set the line:

linecolor: gray
linepalette:
linewidth: 1

The *Chart 10.4* was created by using the commands below—see the parameters set (changed from default).

```
\def\gdpwagesdataone{{1.43,1.93,3.32,
   4.20},{3.440,4.500,5.050,5.530}}
```

```
\scatterchart[smoothlines][
   xscale=1.25, yscale=0.5,
   linepalette=magentapalette,
   dotspalette=magentapalette,
   dotswidth=8,
   xcoor=yes,
   axesunits=yes,
   xunit=Earnings (100 EUR),
   yunit=GDPpc (1000 EUR)][
   method=struct,
   data={\gdpwagesdatatwo}]
```

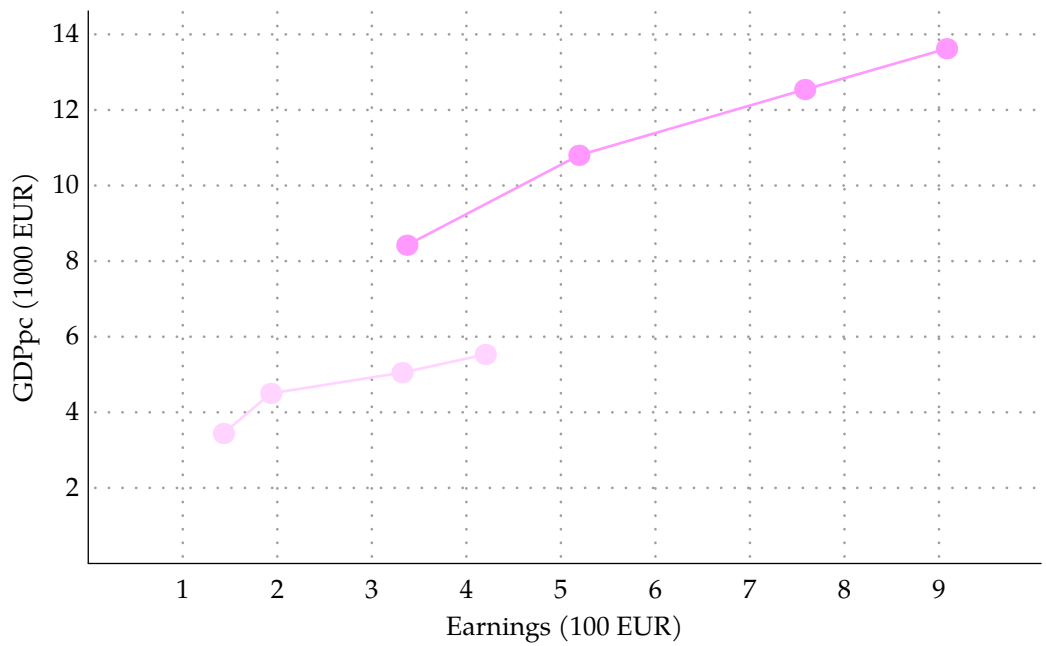

Chart 10.4 Scatter chart with straight lines with multiple series

11 Stock charts

The stock chart (or a price chart) generally shows a change of stock's price over a certain time period. This chart is often used by (potential) investors as it is a great tool to analyze the current stock market situation. The price is displayed along the y-axis and the x-axis represents the change of time. There are various possibilities to visualize data through a stock chart. The chart can consist of lines, bars, candlesticks and points & figures (StockCharts, 2019). Each of those is suitable for different subtypes. The subtypes are named so the reader can easily know which information will the chart contain. There are four of them:

- high-low-close,
- open-high-low-close,
- volume-high-low-close and
- volume-open-high-low-close.

Stock chart parameters

The following parameters (and their default values) can be set in the stock chart.

To work with the axes:

xlength: minimum length needed according to data + left
ylength: minimum length needed according to data + 1 unit
xshift: 0
yshift: 0
xwidth: 0.5
ywidth: 0.5

To scale the entire chart:

xscale: 1
yscale: 1

To set the unit:

unit: cm

To set the position of the initial and all other values:

distance: 2

left: 1

To adjust the grid:

grid: on
gridcolor: middlegray
gridline: dotted
gridlinewidth: 1
xgridlinesdensity: 1
ygridlinesdensity: 1
xgridlines: yes
ygridlines: yes
xgridlinesnoval: yes

To set the line:

linecolor: black
linewidth: 1

To set the labels:

xaxislabels: yes
yaxislabels: yes
axesunits: yes
xunit:
xunit:
xunit: yes
decimals: 1
rounding: halfup
equaldecimals: no

11.1 High-low-close chart

This subtype displays a maximum, minimum, and a closing price of the stock for a particular time period. Elements of this chart are a vertical bar, where the top of the bar represents the maximum price and the bottom the minimum price, and a horizontal line which shows the closing value.

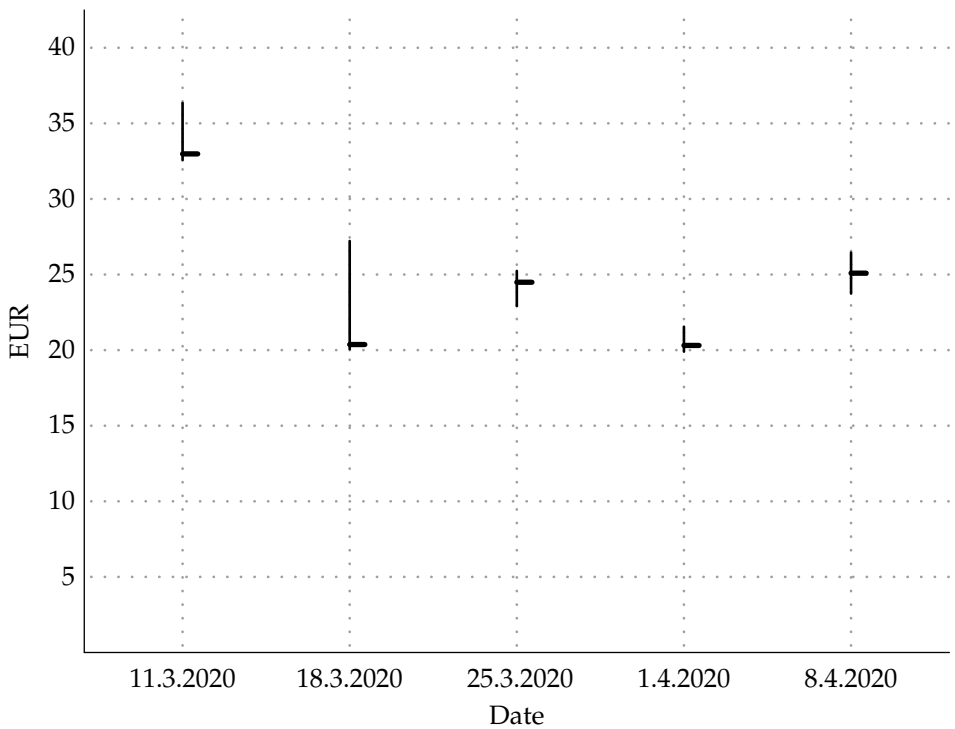

Chart 11.1 High-low-close stock chart

The example (*Chart 11.1*) shows the high, low and close price of crude oil for each of the five chosen days of 2020—see the data used in *Table 11.1*.

Date	11.3.	18.3.	25.3.	1.4.	8.4.
High	36.35	27.22	25.24	21.55	26.45
Low	32.56	20.06	22.91	19.90	19.90
Close	32.98	20.37	24.49	20.31	20.31

Table 11.1 Crude oil historical data (high, low, close) in specific dates of 2020 (INVESTING.COM, 2020)

To manipulate with high-low-close chart, in addition to the parameters listed in *Chapter* and explained in *Chapter 2*, also the following ones (and their default values) can be used.

To set the right line:

rightlinecolor: black
rightlinewidth: 2
rightlinelength: 0.2

The *Chart 11.1* was created by using the following commands—see the parameters set (changed from default).

```
\def\sharesdataone{{36.35,32.56,32.98
   },{27.22,20.06,20.37},{25.24,22.91,
   24.49},{21.55,19.90,20.31},{26.45,
   23.74,25.09}}

\def\datelabels{11.3.2020,18.3.2020,
   25.3.2020,1.4.2020,8.4.2020}

\stockchart[hlc][
   ylength=8.5,
   xscale=1.3, yscale=0.2,
   distance=1.7,
   axesunits=yes,
   xunit=Date, yunit=EUR][
   method=struct,
   data={\sharesdataone},
   xlabels={\datelabels}]
```

11.2 Open-high-low-close chart

Compared to the high-low-close chart, this subtype differs in an additional value—the opening price. However, for visualizing the data, instead of bars and lines, this chart uses the candlesticks. Again, the main vertical line symbolizes the minimum and maximum value and the opening and closing price is represented by a rectangle which is placed along the vertical line. The color of the rectangle depends on whether the

closing price is higher or lower than the opening price—in general when the closing value is higher, the color of those rectangles will be brighter than of those, which closing price is lower than the opening price (where the darker color indicates the fall of price during the specific period). The example (*Chart 11.2*) uses similar data as the high-low-close chart example (*Chart 11.1*), but the data are enriched by the opening price (see *Table 11.2*).

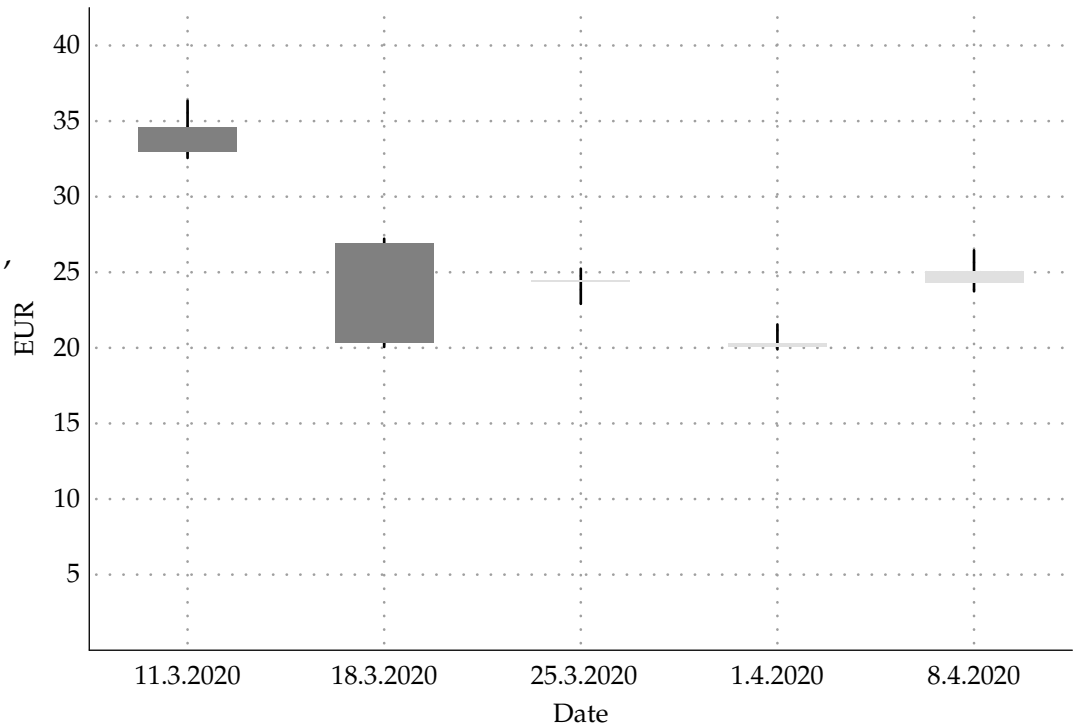

Chart 11.2 Open-high-low-close stock chart

To manipulate with open-high-low-close chart, in addition to the parameters listed in *Chapter* and explained in *Chapter 2*, also the following ones (and their default values) can be used.

Date	11.3.	18.3.	25.3.	1.4.	8.4.
Open	34.62	26.94	24.37	20.10	24.30
High	36.35	27.22	25.24	21.55	26.45
Low	32.56	20.06	22.91	19.90	19.90
Close	32.98	20.37	24.49	20.31	20.31

Table 11.2 Crude oil historical data (open, high, low, close) in specific dates of 2020 (INVESTING.COM, 2020)

To set the rectangle width:

rectanglewidth: 1

To set the open-close rectangle:

ogtcrectcolor: black
oltcrectcolor: white
ocrecttransparency: 1
ocrectcontour: yes
ocrectcontourcolor: black
ocrectcontourwidth: 0.5

The *Chart 11.2* was created by using the commands below—see the parameters set (changed from default).

```
\def\sharesdatatwo{{34.62,36.35,
   32.56,32.98},{26.94,27.22,20.06,
   20.37},{24.37,25.24,22.91,24.49
   },{20.10,21.55,19.90,20.31},{
   24.30,26.45,23.74,25.09}}

\def\datelabels{11.3.2020,18.3.2020,
   25.3.2020,1.4.2020,8.4.2020}

\stockchart[ohlc][
   ylength=8.5,
   xscale=1.3, yscale=0.2,
   left=0.5,
   ogtcrectcolor=baloongray,
   oltcrectcolor=gainsboro,
   ocrectcontour=no,
   axesunits=yes,
   xunit=Date, yunit=EUR][
   method=struct,
   data={\sharesdatatwo},
   xlabels={\datelabels}]
```

11.3 Volume-high-low-close chart

Volume-high-low-close chart, as the name suggests,

adds a volume value to the high-low-close chart. The volume represents the amount of traded stocks within a certain time period (Desjardins, 2016). As in all other stock charts, the vertical line shows the lowest and the highest price. The closing price is represented by a short horizontal line and the rectangle in this chart represents the volume. The example (*Chart 11.3*) shows the volume, high, low, and close values of crude oil in specific dates—see the data used in *Table 11.3*.

Date	11.3.	18.3.	25.3.	1.4.	8.4.
Volume	8.7465	3.0948	6.1873	7.0329	8.2355
High	36.35	27.22	25.24	21.55	26.45
Low	32.56	20.06	22.91	19.90	19.90
Close	32.98	20.37	24.49	20.31	20.31

Table 11.3 Crude oil historical data (volume, high, low, close) in specific dates of 2020 (Investing.com, 2020)

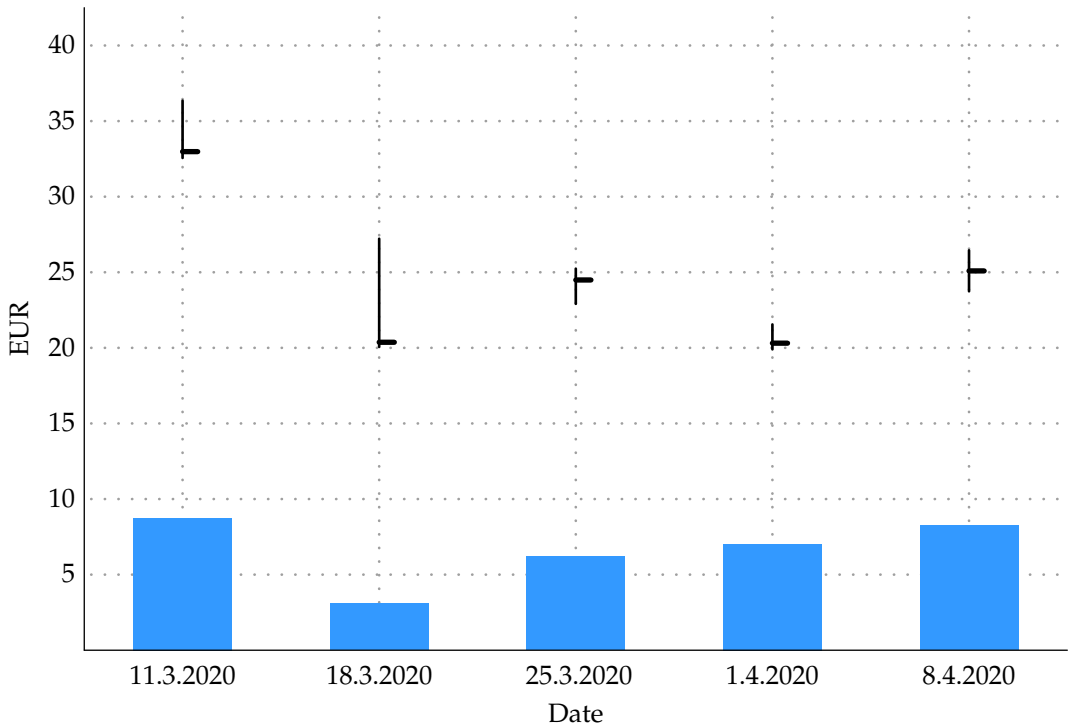

Chart 11.3 Volume-high-low-close stock chart

To manipulate with volume-high-low-close chart, in addition to the parameters listed in *Chapter* and explained in *Chapter 2*, also the following ones (and their default values) can be used.

To set the right line:

rightlinecolor: black
rightlinewidth: 2
rightlinelength: 0.2

To set the rectangle width:

rectanglewidth: 1

To set the volume rectangle:

vrectcolor: gray
vrecttransparency: 1
vrectcontour: yes
vrectcontourcolor: black
vrectcontourwidth: 0.5

The *Chart 11.3* was created by using the commands below—see the parameters set (changed from default).

```
\def\sharesdatathree{{8.7465,36.35,
   32.56,32.98},{3.0948,27.22,20.06,
   20.37},{6.1873,25.24,22.91,24.49
   },{7.0329,21.55,19.90,20.31},{
   8.2355,26.45,23.74,25.09}}

\def\datelabels{11.3.2020,18.3.2020,
   25.3.2020,1.4.2020,8.4.2020}

\stockchart[vhlc][
   ylength=8.5,
   xscale=1.3, yscale=0.2,
   left=0.5,
   vrectcolor=adriblue,
   vrectcontour=no,
   axesunits=yes,
   xunit=Date, yunit=EUR][
   method=struct,
   data={\sharesdatathree},
   xlabels={\datelabels}]
```

11.4 Volume-open-high-low-close chart

This stock chart is a combination of the two previous subtypes, as it contains information about all values—volume, opening, maximum, minimum, and closing price (see the chart in *Chart 11.4* and *Table 11.4*).

The volume and opening/closing price will be both displayed by rectangles which can, in some cases, overlay each other. The maximum and the minimum price is still represented by a vertical line.
To manipulate with volume-open-high-low-close chart, in addition to the parameters listed in *Chapter* and explained in *Chapter 2*, also the following ones (and their default values) can be used.

Date	11.3.	18.3.	25.3.	1.4.	8.4.
Volume	8.7465	3.0948	6.1873	7.0329	8.2355
Open	34.62	26.94	24.37	20.10	24.30
High	36.35	27.22	25.24	21.55	26.45
Low	32.56	20.06	22.91	19.90	19.90
Close	32.98	20.37	24.49	20.31	20.31

Table 11.4 Crude oil historical data (volume, open, high, low, close) in specific dates of 2020 (Investing.com, 2020)

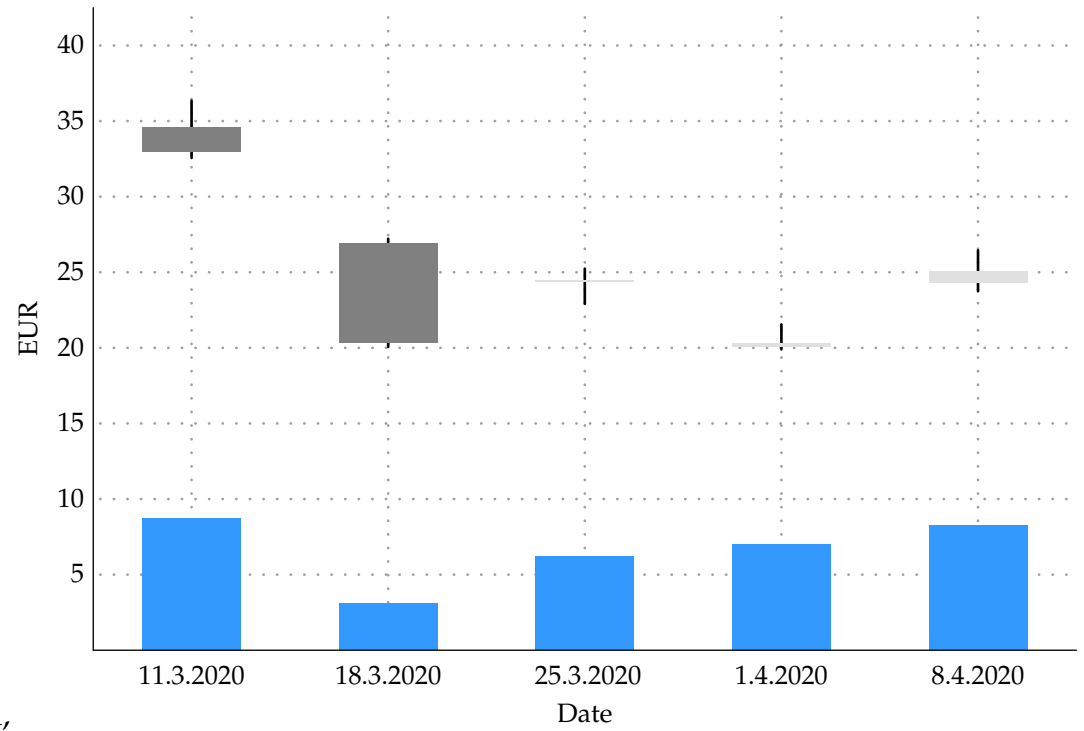

Chart 11.4 Volume-open-high-low-close stock chart

To set the rectangle width:

rectanglewidth: 1

To set the open-close rectangle:

ogtcrectcolor: black
oltcrectcolor: white
ocrecttransparency: 1
ocrectcontour: yes
ocrectcontourcolor: black
ocrectcontourwidth: 0.5

To set the volume rectangle:

vrectcolor: gray
vrecttransparency: 1
vrectcontour: yes
vrectcontourcolor: black
vrectcontourwidth: 0.5

The *Chart 11.4* was created by using the commands below—see the parameters set (changed from default).

```
\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{11.3.2020,18.3.2020,
   25.3.2020,1.4.2020,8.4.2020}

\stockchart[vohlc][
   ylength=8.5,
   xscale=1.3, yscale=0.2,
   left=0.5,
   ogtcrectcolor=baloongray,
   oltcrectcolor=gainsboro,
   ocrectcontour=no,
   vrectcolor=adriblue,
   vrectcontour=no,
   axesunits=yes,
   xunit=Date, yunit=EUR][
   method=struct,
   data={\sharesdatafour},
   xlabels={\datelabels}]
```

12 Where can data come from

No data, no charts. Therefore, this chapter shows various formats which can be used for data input. All formats are based on previous experience with data processing; some of them have been described at ConTEXt Meeting 2016 and published later (Hála, 2017).

12.1 Concept of data storage

For all kinds of chart, we need a unified and easily available data structure.

```
documentdata = documentdata or {}
documentdata.STATCHARTS =
   documentdata.STATCHARTS or {}
d=documentdata.STATCHARTS
```

For data storing, there are two independent table structures:

```
d.data = {}
d.dataset = {}
```

While the first one is used by all charts for getting data necessary for drawing, the second one is used for long-time data storage and it will be described later (see *Section 12.3*).

12.2 Direct data

When more charts are used with method data, the value of table data is always overwritten and the older

values are never stored anywhere.

12.2.1 Plain data

Method `plain` is the basic and default method for one-dimensional data.

```
\chart[..][..][..][
  data={5,6,8,8,10,11}
]
\chart[..][..][..][%
  method=plain,
  data={5,6,8,8,10,11}
]

table={
 ["data"]={
  { 5, 6, 8, 8, 10, 11 },
 },
```

12.2.2 Joined data

When structured data comes as one unstructured stream of values one must use the method `joined`. This method has the same form as the method `plain` and the internal structure information is expressed by the key `per` containing the number of values for one line.

```
\chart[..][..][..][%
  method=joined,
  data={5,3,5,8,9,15,13,15,18,19},
  per=5,
]

table={
 ["data"]={
  { 5, 3, 5, 8, 9 },
  { 15, 13, 15, 18, 19 },
 },
```

12.2.3 Structured data

If we need the same arrangement of data, we can also use the method `struct` which differs in the way how data is coded in the source text.

```
\chart[..][..][..][%
  method=struct,
  data={{5,3,5,8,9},{15,13,15,18,19}},
]

table={
 ["data"]={
  { 5, 3, 5, 8, 9 },
  { 15, 13, 15, 18, 19 },
  { 5, 15 },
  { 8, 18 },
  { 9, 19 },
 },
```

12.2.4 Swapped structured data

The previous way works suppose that from input comes data arranged by rows. The key `c` with value `r` swaps rows and columns:

```
\chart[..][..][..][%
  method=struct,
  data={{5,3,5,8,9},{15,13,15,18,19}},
  c=r,
]

table={
 ["data"]={
  { 5, 15 },
  { 3, 13 },
```

12.2.5 Labels for marking axes

When needed, axes lables can be marked by another data source. For this purpose, keys `xlabels` and/or `ylabels` serve. They contain a list of comma-separated values, see e.g. ...., which will be used for marking axes.

12.3 Datasets

For permanent storage and repeated use, the table `dataset` has been created. Because it is arranged as a hash, one can store any number of datasets. Unlike the table `data`, the given dataset is stored permanently and it can be overwritten only in case the user explicitly uses the same dataset name while he/she defines the file name for loading data.

12.3.1 CSV files

Format CSV (comma separated values) is very frequently used text format for data transfer. Among other things, all contemporary spreadsheets enable data export to CSV. Therefore, it became as the first choice for addition reading tools. The key `dsname` stands for unique name of dataset which will contain data from the file defined with the key `filename`. Columns are specified by `cx` and `cy` for *x*-axis and *y*-axis, respectively. Similarly, one can use `rx` and `ry` for data with transposed arrangement. (In the following examples, let's assume two datasets – EU for European Union and CNB for Czech National Bank.)

```
\chart[..][..][..][method=dataset,
   filename=eu.csv, dsname=EU, cx=2, cy=3]
```

12.3.2 Notation of rows and columns

We can look at datasets as sheets from a spreadsheet. Therefore, columns can be referred with spreadsheet column notation. The following four examples show which part of data will be copied from the existing dataset `EU` to the table `data` for drawing charts.

```
\chart[..][..][..][method=dataset,
   dsname=CNB,
   cx=D, cy=AE]
\chart[..][..][..][method=dataset,
   dsname=EU,
   cx=B2:B5, cy=C3:C6]
\chart[..][..][..][method=dataset,
   dsname=EU,
   rx=D, ry=E]
\chart[..][..][..][method=dataset,
   dsname=CNB,
   rx=AA10:A15, ry=AF10:AF15]
```

The first way stands for the whole column, the second one uses the popularly known spreadsheet notation.
Keys `rx` and `ry` work similarly as `cx` and `cy` but with swapping rows and columns.

12.3.3 Install a dataset

The dataset can be loaded independently on the drawing which could make writing chart commands more comfortable. When the dataset is installed, the key `filename` can be skipped at chart commands.

```
\installdataset[filename=eu.csv, dsname=
xxxx, sep=|]
```

12.3.4 Separators

Generally, the comma is set as a default separator. However as definition of CSV format states, data can be separated by any other character. The following code shows how to do it.

```
\chart[..][..][..][
  method=dataset,
  dsname=cnb20,
  sep=|,
  decimal=comma,
  rx=2, ry=3]
```

12.3.5 Decimal comma versus dot

Data originated in language regions where decimal comma is used, could not be read as numbers by default. As the previous snippet of the code shows, decimal comma will be accepted by the key `decimal`. This feature is applied only to such a piece of data which meets the rules for writing numbers, others are left untouched.

References

AnyChart. 100 Percent Stacked Area Chart. In *AnyChart* [on-line!]. c2020 [cit. 2020-01-13]. Available at: https://www.anychart.com/products/anychart/gallery/Area_Charts/100_Percent_Stacked_Area_Chart.php.

amCharts. 100% Stacked Column Chart. In *amCharts* [on-line!]. c2006–2020 [cit. 2020-01-15]. Available at: https://www.amcharts.com/demos/100-stacked-column-chart/.

Chesnutt, B. What is an Area Chart?—Definition & Examples : What Is an Area Chart?. In *Study* [on-line!]. 2016 [cit. 2020-01-13]. Available at: https://study.com/academy/lesson/what-is-an-area-chart-definition-examples.html.

Desjardins, J. What is a Stock Chart?. In *Visual Capitalist* [on-line!]. 2016 [cit. 2020-02-05]. Available at: https://www.visualcapitalist.com/what-is-a-stock-chart/.

Hagen, H. *Metafun : context mkiv* [on-line!]. Hasselt : Boekplan, 2017. [cit. 2020-01-17]. 428 pp. ISBN 978-94-90688-02-8. Available at: http://www.pragma-ade.com/general/manuals/metafun-p.pdf.

Hála, T. Processing of Delimiter Separated Values in ConTeXt. In *Proceedings : 10th ConTEXt Meeting.* 1. ed. [(Netherlands)] : ConTEXt Group, 2017, p. 58–69.

Hobby, J. D. MetaPost : A user's manual. In *TEXUsers Group* [on-line!]. 2019 [cit. 2020-01-17]. Available at: https://www.tug.org/docs/metapost/mpman.pdf.

Infragistics. Stacked Line Chart. In *Infragistics* [on-line!]. c2008–2011a [cit. 2020-01-13]. Available at: https://help.infragistics.com/Help/Doc/Silverlight/2011.1/CLR4.0/html/xamWebChart_Stacked_Line_Chart.html.

Infragistics. Stacked 100 Line Chart. In *Infragistics* [on-line!]. c2008–2011b [cit. 2020-01-13]. Available at: https://help.infragistics.com/Help/Doc/Silverlight/2011.1/CLR4.0/html/xamWebChart_Stacked_100_Line_Chart.html.

Keynote Support. Excel Chart Types: Pie, Column, Line, Bar, Area, and Scatter. In *Keynote Support* [on-line!]. c2008–2020 [cit. 2020-01-13]. Available at: https://www.keynotesupport.com/excel-basics/excel-chart-types.shtml.

Lile, S. 44 Types of Graphs Perfect for Every Top Industry. In *Visme* [on-line!]. c2020 [cit. 2020-01-15]. Available at: https://visme.co/blog/types-of-graphs/.

Nowicki, H., Merenstein, C. Radar Chart : CS 465: Information Visualization. In *Radar Chart* [on-line!]. 2016 [cit. 2020-02-05]. Available at: https://skylight.middlebury.edu/~candrews/showcase/infovis_techniques_s16/radar_chart/.

Rost, L. C. What to consider when creating stacked column charts. In *Chartable* [on-line!]. 2018 [cit. 2020-01-13]. Available at: https://blog.datawrapper.de/stacked-column-charts/.

Rouse, M. scatter plot. In *WhatIs* [on-line!]. 1999–2020 [cit. 2020-01-15]. Available at: https://whatis.techtarget.com/definition/scatter-plot.

SkillsYouNeed. Graphs and Charts : Types of Charts. In *SkillsYouNeed* [on-line!]. c2011–2020 [cit. 2020-01-15]. Available at: https://www.skillsyouneed.com/num/graphs-charts.html.

StockCharts. What Are Charts?. In *StockCharts* [on-line!]. 2019 [cit. 2020-01-16]. Available at: https://school.stockcharts.com/doku.php?id=chart_analysis:what_are_charts.

The Data Visualisation Catalogue. Stacked Area Graph. In *The Data Visualisation Catalogue* [on-line!]. 2019a [cit. 2020-01-15]. Available at: https://datavizcatalogue.com/methods/stacked_area_graph.html.

The Data Visualisation Catalogue. Stacked Bar Graph. In *The Data Visualisation Catalogue* [on-line!]. 2019b [cit. 2020-01-15]. Available at: https://datavizcatalogue.com/methods/stacked_bar_graph.html.

WallStreetMojo. Clustered Bar Chart in Excel. In *WallStreetMojo* [on-line!]. c2020 [cit. 2020-01-13]. Available at: https://www.wallstreetmojo.com/clustered-bar-chart-excel/.

Yi, M. A Complete Guide to Bubble Charts : What is a bubble chart?. In *Chartio* [on-line!]. 2019 [cit. 2020-01-16]. Available at: https://chartio.com/learn/charts/bubble-chart-complete-guide/.

Data sources

CZSO. Zaměstnanost, nezaměstnanost – časové řady [Employment, Unemployment]. In *Czech Statistical Office* [on-line!]. 2020 [cit. 2020-05-19]. Available at: https://www.czso.cz/csu/czso/zam_cr.

Eurostat. Acquisition of citizenship by age group, sex and former citizenship. In *Eurostat* [on-line!]. 2020a [cit. 2020-05-19]. Available at: https://ec.europa.eu/eurostat/en/web/products-datasets/-/MIGR_ACQ.

Eurostat. Area by NUTS 3 region. In *Eurostat* [on-line!]. 2020b [cit. 2020-05-19]. Available at: https://ec.europa.eu/eurostat/en/web/products-datasets/-/DEMO_R_D3AREA.

Eurostat. Graduates by education level, programme orientation, sex and field of education. In *Eurostat* [on-line!]. 2020c [cit. 2020-05-19]. Available at: https://ec.europa.eu/eurostat/web/products-datasets/product?code=educ_uoe_grad02.

Eurostat. Classroom teachers and academic staff by education level, programme orientation, sex and age groups. In *Eurostat* [on-line!]. 2020d [cit. 2020-05-19]. Available at: https://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=educ_uoe_perp01&lang=en.

Eurostat. Time spent, participation time and participation rate in the main activity by sex and household composition. In *Eurostat* [on-line!]. 2020e [cit. 2020-05-19]. Available at: https://ec.europa.eu/eurostat/web/products-datasets/-/tus_00hhstatus.

Eurostat. General government expenditure by function (COFOG). In *Eurostat* [on-line!]. 2020f [cit. 2020-05-19]. Available at: https://ec.europa.eu/eurostat/web/products-datasets/product?code=gov_10a_exp.

Eurostat. Real GDP per capita. In *Eurostat* [on-line!]. 2020g [cit. 2020-05-19]. Available at: https://ec.europa.eu/eurostat/web/products-datasets/-/sdg_08_10.

Eurostat. Structure of earnings survey: monthly earnings. In *Eurostat* [on-line!]. 2020h [cit. 2020-05-19]. Available at: https://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=earn_ses_monthly&lang=en.

Investing.com. Crude Oil WTI Futures—Jul 20 (CLN0). In *Investing.com* [on-line!]. 2020 [cit. 2020-05-19]. Available at: https://www.investing.com/commodities/crude-oil-historical-data.

Index

Review of the syntax

```
\chart [..[1]..] [..[2]..] [..,..[3]=..,..] [..,..[4]=..,..]

1   CHART_TYPE

2   CHART_SUBTYPE

3   unit                    =  DIMENSION cm
    xlength                 =  NUMBER
    ylength                 =  NUMBER
    xshift                  =  NUMBER 0
    yshift                  =  NUMBER 0
    xwidth                  =  NUMBER 0.5
    ywidth                  =  NUMBER 0.5
    xscale                  =  NUMBER 1
    yscale                  =  NUMBER 1
    xaxislabels             =  yes no
    yaxislabels             =  yes no
    axesunits               =  yes no
    xunit                   =  TEXT
    yunit                   =  TEXT
    xaxislabelsnoval        =  yes no
    yaxislabelsnoval        =  yes no
    xlabelscount            =  NUMBER 5
    ylabelscount            =  NUMBER 5
    distance                =  NUMBER 1
    left                    =  NUMBER 1
    bottom                  =  NUMBER 1
    grid                    =  on off
    gridcolor               =  COLOR middlegray
    gridline                =  dashed dotted full
    gridlinewidth           =  NUMBER 1
    xgridlines              =  yes no
    ygridlines              =  yes no
    xgridlinesdensity       =  NUMBER 1
    ygridlinesdensity       =  NUMBER 1
    xgridlinesnoval         =  yes no
    ygridlinesnoval         =  yes no
    xgridlinesleft          =  yes no
    fill                    =  yes no
    fillcolor               =  COLOR
```

```
    fillpalette             =  TEXT
    filltransparency        =  NUMBER
    line                    =  yes no
    linecolor               =  COLOR black
    linepalette             =  TEXT
    linewidth               =  NUMBER 1
    barcolor                =  COLOR gray
    barpalette              =  TEXT
    bartransparency         =  NUMBER 1
    barwidth                =  NUMBER 1
    columncolor             =  COLOR gray
    columnpalette           =  TEXT
    columntransparency      =  NUMBER 1
    columnwidth             =  NUMBER 1
    bubblediameter          =  NUMBER 1
    bubblecolor             =  COLOR gray
    bubblepalette           =  TEXT
    bubbletransparency      =  NUMBER 0.7
    dots                    =  yes no
    dotscolor               =  COLOR black
    dotspalette             =  TEXT
    dotswidth               =  NUMBER 5
    dotslabels              =  yes no
    contour                 =  yes no
    contourcolor            =  COLOR black
    contourpalette          =  TEXT
    contourwidth            =  NUMBER 0.5
    equaldecimals           =  yes no
    decimals                =  NUMBER 1
    rounding                =  up down halfup halfdown halfabsup halfabsdown halfeven halfodd
    xcoor                   =  yes no
    zcoor                   =  yes no
    xaxislabinit            =  NUMBER 1
    xaxislabinc             =  NUMBER 1
    axes                    =  yes no
    axiswidth               =  NUMBER 0.5
    axisline                =  dashed dotted full
    axiscolor               =  COLOR middlegray
    step                    =  NUMBER 2
```

```
    layers                  =  NUMBER
    layercolor              =  COLOR black
    layerline               =  dashed dotted full
    layerwidth              =  NUMBER 0.5
    layerslabels            =  yes no
    diameter                =  NUMBER 5
    holediameter            =  NUMBER 3
    perc                    =  yes no
    distanceout             =  NUMBER 0.2
    distancebetween         =  NUMBER 0
    labels                  =  yes no
    labelposition           =  in out
    coef                    =  NUMBER 0
    out                     =  NUMBER {NUMBER_LIST}
    rightlinecolor          =  COLOR black
    rightlinewidth          =  NUMBER 2
    rightlinelength         =  NUMBER 0.2
    rectanglewidth          =  NUMBER 1
    ogtcrectcolor           =  COLOR black
    oltcrectcolor           =  COLOR white
    ocrecttransparency      =  NUMBER 1
    ocrectcontour           =  yes no
    ocrectcontourcolor      =  COLOR black
    ocrectcontourwidth      =  NUMBER 0.5
    vrectcolor              =  COLOR gray
    vrecttransparency       =  NUMBER 1
    vrectcontour            =  yes no
    vrectcontourcolor       =  COLOR black
    vrectcontourwidth       =  NUMBER 0.5

4   method          =  plain joined struct dataset
    data            =  STRUCTURED_DATA
    per             =  NUMBER (default 0)
    xlabels         =  TEXT
    ylabels         =  TEXT
    c               =  r
    cx              =  NUMBER COLUMN_NAME COLUMN_INTERVAL
    cy              =  NUMBER COLUMN_NAME COLUMN_INTERVAL
    cz              =  NUMBER COLUMN_NAME COLUMN_INTERVAL
```

```
    decimal       =  comma dot
    rx            =  NUMBER COLUMN_NAME COLUMN_INTERVAL
    ry            =  NUMBER COLUMN_NAME COLUMN_INTERVAL
    inherits: \installdataset
```

```
\installdataset [..,..*=..,..]

*   sep           =  CHAR
    filename      =  FILENAME
    dsname        =  NAME
```

```
\areachart [.1..] [.2..] [.3..]

1   CHART_SUBTYPE

2   inherits: \chart [##3]

3   inherits: \chart [##4]
```

```
\barchart [.1..] [.2..] [.3..]

1   CHART_SUBTYPE

2   inherits: \chart [##3]

3   inherits: \chart [##4]
```

```
\bubblechart [.1..] [.2..] [.3..]

1   CHART_SUBTYPE

2   inherits: \chart [##3]

3   inherits: \chart [##4]
```

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

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

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

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

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

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