

## Chapter 6

# Categorical Data

The presentation of simple frequencies or of parameters such as percentages or averages by bar and column charts is certainly one of the most widely used visualisations. Therefore, this is where we want to start. Sometimes though, one might want to also plot parameters that are located on a scale between two statements. In such cases, the use of a profile plot makes sense. If there are a lot of attributes in a bar chart and/or you want to plot several variables simultaneously, then a different illustration form is needed. In such cases, a dot chart comes in handy. The next section explains regular pie charts, but also gives examples of spie charts, radial polygons, and radial column charts. “Chart tables” refers to illustration types in which the arrangement of information has table character. We start with suggestions for two variants of so-called Gantt-charts, then follow with examples for a bump chart, a heat map, a mosaic plot and two examples for tree maps.

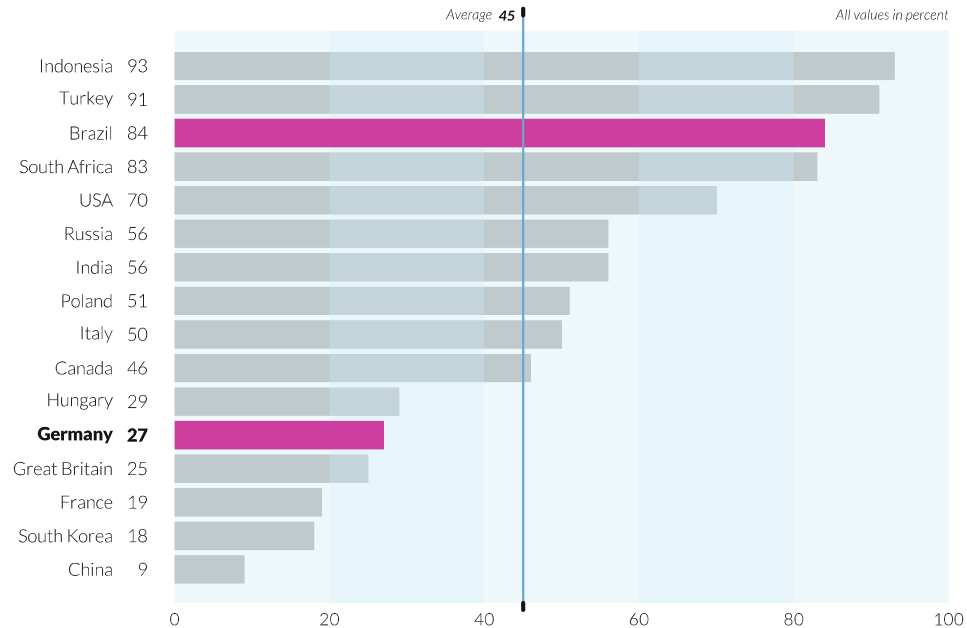
### 6.1 Bar and Column Charts

Attributes of nominal values (frequencies, fractions, etc.) should be presented as bars, while columns should be reserved for ordinal or metrical variables. However, even this should not be considered a rule set in stone. If you think that your concrete case calls for a column chart rather than a bar chart, then you may of course use that. Even though bar charts are omnipresent, their design is anything but easy. The following, seemingly easy examples give an adequate representation of how R can match almost any imaginable design requirements. Starting with the first example, we explain a series of basic principles that we follow throughout this book.

### 6.1.1 Bar Chart Simple

#### 'I Definitely Believe in God or a Supreme Being'

was said in 2010 in:



Source: www.ipsos-na.com. Design: Stefan Fichtel, iextract

The figure shows the results of a 2010 survey carried out in different countries: How many percent of the respondents agreed with the statement “I Definitely Believe in God or a Supreme Being”? There is no common rule about the aspect ratio of bar or column charts; the most aesthetically pleasing presentation depends on several factors such as the actual values, the number of attributes, and their range and variance, so that the best value has to be found by trial and error for each individual case. With bar charts, it is usually best to sort bars in descending order. However, this is not obligatory. One could as well sort the bars in alphabetical order. It completely depends on the statement one wants to make. Bars should be styled in one colour. Individual bars/values can be highlighted in another colour. If the bars are left-justified, then the labelling should be right-justified, using visually appealing spacing. Text size should match bar thickness, unless very long labels are used, therefore being wrapped over multiple lines (see Sect. 6.1.3). It is often useful to present the actual values next to the labels. If percent values are plotted instead of numbers, then the repetitive use of “%” can be forgone as long as a single reference reading “all values in percent” is added at a suitable location. A part of

the values can be specifically highlighted, here in magenta colour. The colour of the highlight should be clearly distinguishable from the other bars' colour. It is also useful to indicate the highlight in the labelling. To do this, one should choose a typeface that does not only have a regular and a bold cut, but several grades. Here, we use the "light" and "black" cuts of the Lato typeface, which offers five weights. Additionally, it is sometimes useful to also include a trend line as a reference value. This variant is preferred over the use of an artificial category "average" as an extra bar. The line should be explained in a label. As scaling for an x-axis ranging from 0 to 100%, either 20 or 25% steps can be chosen. With either option, a background labelling, whose area matches the markings on the axis, is recommended to facilitate orientation. Alternatively, ledger lines can be used, as shown in the subsequent examples. In the present case, the background area is alternately coloured in light shades of blue. Data are derived from an Ipsos survey that was ordered by the Thompson Reuters News Service and performed between 7 and 23 September 2010 in 24 countries. The participating countries were Argentina, Australia, Belgium, Brazil, Canada, China, France, Germany, Great Britain, Hungary, India, Indonesia, Italy, Japan, Mexico, Poland, Russia, Saudi Arabia, South Africa, South Korea, Spain, Sweden, Turkey, and the USA. The study used an international sample of adults aged between 18 and 64 years from the USA and Canada, and aged between 16 and 64 years from all other countries. The unweighted basis of the respondents numbered 18,531 people. The survey included approximately 1000 people from each country, excluding Argentina, Indonesia, Mexico, Poland, Saudi Arabia, South Africa, South Korea, Sweden, Russia, and Turkey, where the sample size was approximately 500. Data were taken from the Open Mind Journal website and manually entered into an XLSX table.

```
pdf_file<-"pdf/barcharts_simple.pdf"
cairo_pdf(bg="grey98", pdf_file,width=9,height=6.5)

par(omi=c(0.65,0.25,0.75,0.75),mai=c(0.3,2,0.35,0),mgp=c(3,3,0),
     family="Lato Light", las=1)

# Import data and prepare chart

library(gdata)
ipsos<-read.xls("myData/ipsos.xlsx", encoding="latin1")
sort.ipsos<-ipsos[order(ipsos$Percent) ,]
attach(sort.ipsos)

# Create chart

x<-barplot(Percent,names.arg=F,horiz=T,border=NA,xlim=c(0,100),
           col="grey", cex.names=0.85,axes=F)

# Label chart

for (i in 1:length(Country))
{
if (Country[i] %in% c("Germany","Brasil"))
```

```

    {myFont<-"Lato Black"} else {myFont<-"Lato Light"}
text(-8,x[i],Country[i],xpd=T,adj=1,cex=0.85,family=myFont)
text(-3.5,x[i],Percent[i],xpd=T,adj=1,cex=0.85,family=myFont)
}

# Other elements

rect(0,-0.5,20,28,col=rgb(191,239,255,80,maxColorValue=255),\
     border=NA)
rect(20,-0.5,40,28,col=rgb(191,239,255,120,maxColorValue=255),\
     border=NA)
rect(40,-0.5,60,28,col=rgb(191,239,255,80,maxColorValue=255),\
     border=NA)
rect(60,-0.5,80,28,col=rgb(191,239,255,120,maxColorValue=255),\
     border=NA)
rect(80,-0.5,100,28,col=rgb(191,239,255,80,maxColorValue=255),\
     border=NA)

myValue2<-c(0,0,0,0,27,0,0,0,0,0,0,0,0,0,84,0,0)
myColour2<-rgb(255,0,210,maxColorValue=255)
x2<-barplot(myValue2,names.arg=F,hORIZ=T,border=NA,xlim=c(0,100)\
            ,col=myColour2,cex.names=0.85,axes=F,add=T)

arrows(45,-0.5,45,20.5,lwd=1.5,length=0,xpd=T,col="skyblue3")
arrows(45,-0.5,45,-0.75,lwd=3,length=0,xpd=T)
arrows(45,20.5,45,20.75,lwd=3,length=0,xpd=T)
text(41,20.5,"Average",adj=1,xpd=T,cex=0.65,font=3)
text(44,20.5,"45",adj=1,xpd=T,cex=0.65,family="Lato",font=4)
text(100,20.5,"All values in percent",adj=1,xpd=T,cex=0.65,font\
      =3)
mtext(c(0,20,40,60,80,100),at=c(0,20,40,60,80,100),1,line=0,cex\
      =0.80)

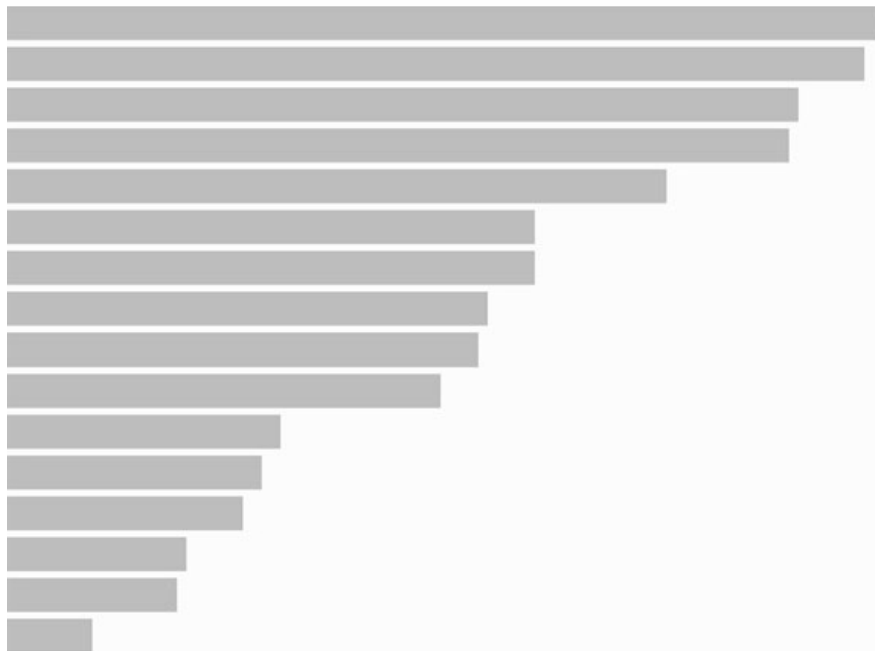
# Titling

mtext("'I Definitely Believe in God or a Supreme Being'",3,line\
      =1.3,adj=0,cex=1.2,family="Lato Black",outer=T)
mtext("was said in 2010 in:",3,line=-0.4,adj=0,cex=0.9,outer=T)
mtext("Source: www.ipsos-na.com, Design: Stefan Fichtel, iextract\
      ",1,line=1,adj=1.0,cex=0.65,outer=T,font=3)
dev.off()

```

The script begins by setting the window size to a width of 9 inches and a height of 6.5 inches, and setting outer (omi) and inner (mai) margins and the distance of the axis labels (mfp). The left margin is chosen a little wider so we have more space for the labels. Data are read from an XLSX table and assigned to the ipsos variable. Then we sort the values using ipsos[order(ipsos\$value),]. This does appear uncommon at first glance: sorting is done by creating a new data area that results

from the old one and rearranges the index values. The `order()` function with the argument `ipsos$value` returns the order of the sorted rows, here 24, 23, 21, 22. . . . Since the function is called from within the square brackets of `ipsos`, followed by a comma and a space, the entire data frame, i.e., all variables, is returned in the sorted order. Note that the `barplot()` function will plot the bars from bottom to top. This is against our intuition, but does not differ from spreadsheet software. Using `attach()`, we can append the sorted data frame to R's search path, so that the variables of the data frame can be directly addressed in the subsequent course of the script. This means that the statement `Country` instead of `sort.ipsos$Country` suffices. Now, the actual bar chart can be created. R uses the `barplot()` function for both bar and column charts. The latter is the default setting; to plot bars rather than columns, `horiz=T` has to be specified. Since we will need the values of the vertical positions of the bars further down the script, we assign the result of `barplot()` as an object to the variable `x`. The parameters and values used in `barplot` are quickly explained: The actual numbers are given first via the value vector, and labels are suppressed by assigning the logical value `F` to the `names.arg` parameter. As previously explained, `horiz=T` creates bars instead of columns, then `border=NA` means no frames are drawn around the bars; the value axis ranges from 0 to 100, the bar colour is grey, the font size is reduced by the factor 0.85, and no axes are to be drawn. Now we have created a fairly basic bar chart:

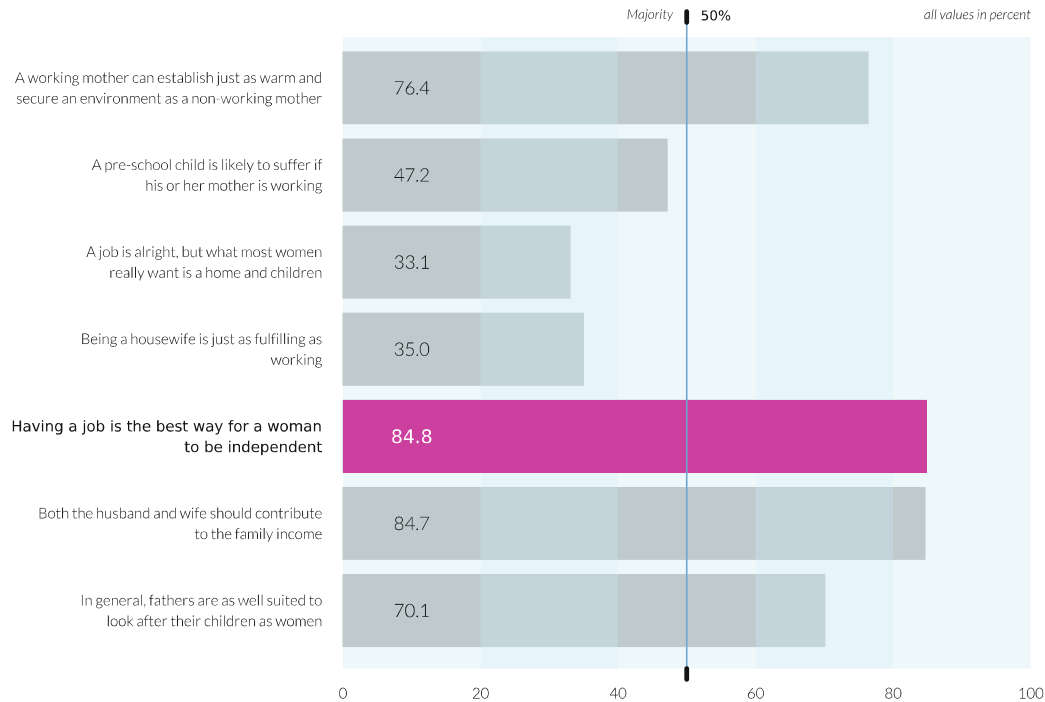


This, however, gives us the opportunity to individualise the missing elements before adding them. First comes the labelling of the bars. To do this, we define a loop that runs through every value of country and changes the font to “Lato Black” if the country is Germany or Brazil. We use the `text()` function for the labels. The respective country name and the percent value are positioned horizontally at the coordinates  $-8$  and  $-3.5$ . The vertical positioning is achieved by referring to the object `x`, in which we previously saved the result of the `barplot()` call. The  $y$ -position of the  $i^{\text{th}}$  bar is simply the value `x[i]`. R automatically extracts the required value from the object, i.e., the vertical position of the bar. Since the `x` values are located outside of the actual chart (the percent values only start at 0), we have to explicitly enable labelling outside of the chart with `xpd=T`. Otherwise, R would suppress the output. We also specify right justification (`adj=1`), a font size, which in turn is decreased by a factor of 0.85, and the font as a vector that contains either the value “Lato Light” or “Lato Black”. Background areas are plotted next. This is done through use of the `rect()` function, which places a shade of blue in different, alternating transparencies over the bar chart. Afterwards, a second co-domain is specified, and a magenta hue as a second colour. To maintain clarity, we use `rgb()` to assign the colour specification to a variable `myColour2`. `myValue2` only contains the respective percent value at the positions of Germany and Brazil, otherwise its value is zero. `x2` causes a second `barplot()` call that complements the existing bar chart with `add=T`. Why the detour? Could we not have just plotted two magenta bars in the first bar chart? Technically, this would have been easily feasible; in the first `barplot` call, the constant `grey` following the `col=...` parameter would have to be replaced with a vector containing the colour specifications for each individual bar. However, those bars would then have also been covered by the transparent blue areas. In our case, we first plotted the grey bars, then overlaid them with the transparent blue areas, which were then overlaid by the two non-transparent magenta bars. This is where the basic principle of R’s traditional approach towards charts shows, opening up an almost unlimited scope for design. After that, there are only a series of annotations: with three calls of the `arrows()` function, a blue trend line is plotted at 45%, with black ends for aesthetic reasons. The line is labelled, and “all values in percent” is positioned at the top right. The statement `font=3` causes an italic font, `font=4` a bold one. Once again, we have to use `xpd=T` to enable plotting outside the chart area. Labelling of the value axis is done via the `mtext()` function, which requires specification of the values that are to be plotted and their positions. Finally, the title and subtitle to be plotted above, and the source to be plotted below the chart, have to be defined, once again with the `mtext()` function. All three are returned in the outer margin. The space for this was initially set with the parameter `omi=c(0.65,0.25,0.75,0.75)` of the `par()` function. At the very end, `dev.off()` completes the graphic output, and then the figure is saved into the pre-defined PDF file.

### 6.1.2 Bar Chart for Multiple Response Questions: First Two Response Categories

#### It is often said that attitudes towards gender roles are changing

Agree strongly / agree



Source: European Values Study 2008 Germany, ZA4753. www.gesis.org. Design: Stefan Fichtel, iextract

The figure shows the results of a survey. The data are based on the European Values Study, a long-term study on values and attitudes of male and female Europeans; shown here is a survey from 2008 to 2010. The study has been conducted since the early 1980s and is repeated every 9 years. Aside from a series of questions concerning value orientation, socio-economic data are also collected. On the topic of “It is often said that attitudes towards gender roles are changing”, the respondents were presented with a series of statements. They could respond to each statement with “Agree strongly”, “Agree”, “Disagree”, “Disagree strongly” and “Don’t know”. The look of the figure almost matches the previous example. However, there are a few differences. The first difference lies in the data: while before the diagram was defined by the individual attributes of a variable, here several variables are combined: each bar shows a variable’s value. Such a group of questions on one topic usually means longer labels for the bars, since one wants to show individual statements. The message of the thematic cluster acts perfectly as the title, while the subtitle shows the selections that were chosen from the answers. In the current example, these are the percent values of the first two categories “agree strongly” and

“agree”. Aside from the repetition of the complete statements that the respondents agreed to, given the extensive labels, it is in this case also useful to write the percent value in the bars. As in the previous examples, the bars are also again complemented by alternating blue areas. For illustration purposes, one question is once more especially highlighted.

Data: see appendix A, ZA4753: European Values Study 2008: Germany (EVS 2008).

First, we save the labels into a separate file called scripts/inc\_labels\_za4753.r (slightly shortened for print):

```
f_v159<-"A working mother can establish just\nas warm and secure\n
..."
f_v160<-"A pre-school child is likely to suffer if\nhis or her..."
f_v161<-"A job is alright but what most women\nreally want ..."
f_v162<-"Being a housewife is just as fulfilling as\nworking..."

f_v163<-"Having a job is the best way for a woman\nto be..."
f_v164<-"Both the husband and wife should contribute\nto ..."
f_v165<-"In general, fathers are as well suited to\nlook after ..."
"
f_v166<-"Men should take as much responsibility\nas women ..."
names<-c(f_v165, f_v164, f_v163, f_v162, f_v161, f_v160, f_v159)
```

Now comes the script for the figure:

```
pdf_file<-"pdf/barcharts_multiple.pdf"
cairo_pdf(bg="grey98", pdf_file, width=13,height=10.5)

par(omi=c(0.65,0.75,1.25,0.75),mai=c(0.9,3.85,0.55,0),lheight\
    =1.15,family="Lato Light",las=1)
source("scripts/inc_labels_za4753.r")
library(memisc)

# Read data and prepare chart

ZA4753<-spss.system.file("myData/ZA4753_v1-1-0.sav")
myData<-subset(ZA4753,select=c(v159,v160,v161,v162,v163,v164,\
    v165))
attach(myData)
z<-NULL
y<-table(as.matrix(v165))
z<-c(z,100*(y["1"]+y["2"])/sum(y))
y<-table(as.matrix(v164))
z<-c(z,100*(y["1"]+y["2"])/sum(y))
y<-table(as.matrix(v163))
z<-c(z,100*(y["1"]+y["2"])/sum(y))
y<-table(as.matrix(v162))
z<-c(z,100*(y["1"]+y["2"])/sum(y))
y<-table(as.matrix(v161))
z<-c(z,100*(y["1"]+y["2"])/sum(y))
y<-table(as.matrix(v160))
z<-c(z,100*(y["1"]+y["2"])/sum(y))
y<-c(0,table(as.matrix(v159)))
```



```

z<-c(z,100*(y["1"]+y["2"])/sum(y))

# Create chart

bp<-barplot(z,names.arg=F,horiz=T,border=NA,xlim=c(0,100),
  col="grey",axes=F,family="Lato")
myColour<-rgb(255,0,210,maxColorValue=255)
rect(0,-0.1,20,8.6,col=rgb(191,239,255,80,maxColorValue=255),\
  border=NA)
rect(20,-0.1,40,8.6,col=rgb(191,239,255,120,maxColorValue=255),\
  border=NA)
rect(40,-0.1,60,8.6,col=rgb(191,239,255,80,maxColorValue=255),\
  border=NA)
rect(60,-0.1,80,8.6,col=rgb(191,239,255,120,maxColorValue=255),\
  border=NA)
rect(80,-0.1,100,8.6,col=rgb(191,239,255,80,maxColorValue=255),\
  border=NA)
z2<-c(0,0,84.81928,0,0,0,0)
bp<-barplot(z2,names.arg=F,horiz=T,border=NA,xlim=c(0,100),
  col=myColour,axes=F,add=T)

# Other elements

for (i in 1:length(mynames))
{
  if (i == 3) {myFont<-"Lato Bold"} else {myFont<-"Lato Light"}
  text(-3,bp[i],mynames[i],xpd=T,adj=1,family=myFont,cex=1)
  text(10,bp[i],format(round(z[i],1),nsmall=1),family=myFont,cex\
    =1.25,
    col=ifelse(i==3,"white","black"))
}
arrows(50,-0.1,50,8.8,lwd=1.5,length=0,xpd=T,col="skyblue3")
arrows(50,-0.25,50,-0.1,lwd=5,length=0,xpd=T)
arrows(50,8.8,50,8.95,lwd=5,length=0,xpd=T)
text(48,8.9,"Majority",adj=1,xpd=T,cex=0.9,font=3)
text(52,8.9,"50%",adj=0,xpd=T,cex=0.9,family="Lato Bold",font=3)
text(100,8.9,"all values in percent",adj=1,xpd=T,cex=0.9,font=3)
mtext(c(0,20,40,60,80,100),at=c(0,20,40,60,80,100),1,line=0.75)

# Titling

mtext("It is often said that attitudes towards gender roles are \
  changing",3,line=2.2,adj=0,cex=1.8,family="Lato Black",\
  outer=T)
mtext("Agree strongly / agree ",3,line=0,adj=0,cex=1.5,outer=T)
mtext("Source: European Values Study 2008 Germany, ZA4753. www.\
  gesis.org. Design: Stefan Fichtel, iextract",1,line=0,adj\
    =1,cex=0.95,outer=T,font=3)
dev.off()

```

In the script, a file containing the actual questions that we need for the labelling of the bars has to be read first to create the figures. The questions are not contained in the SPSS file, but only in the original survey available on the GESIS ZACAT portal. We have copy/pasted those questions, assigned them to compact variables (`f_v159`, `f_v160`, ...), and fitted a line break where appropriate. The actual data are imported using Martin Elff's `memisc` package. The advantage of this package lies in the fact that the import does not actually import the entire SPSS data frame, but only the required part. We use the `select` parameter of the `subset()` function to import the variables `v159`, `v160`, `v161`, `v162`, `v163`, `v164`, and `v165`. The attributes of the variables can be viewed using the `codebook()` function, for example,

```
>codebook(v159)
=====
v159 'working mother warm relationship with children (Q48A)'
-----

Storage mode: double
Measurement: nominal
Missing values: -5--1

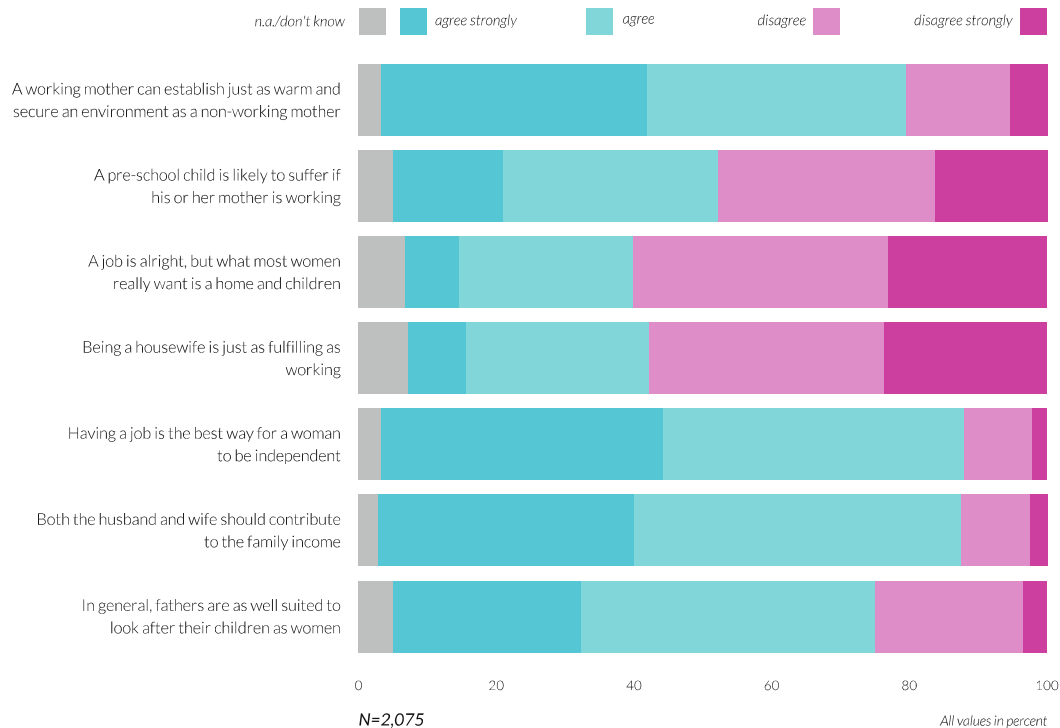
      Values and labels      N      Percent
-----
-5 M 'other missing'        0         0.0
-4 M 'question not asked'   0         0.0
-3 M 'nap'                   0         0.0
-2 M 'na'                     0         0.0
-1 M 'dk'                     67         3.2
 1  'agree strongly'        803        40.0  38.7
 2  'agree'                  782        38.9  37.7
 3  'disagree'               311        15.5  15.0
 4  'disagree strongly'     112         5.6   5.4
```

The values `-5`, `-4`, `-3`, `-2`, and `-1` are defined as missing values and are therefore not taken into account during statistical calculations. With the exception of `dk` (don't know) and `na` (no answer), none of these characteristics come to pass in the selected seven variables. If we want to show a bar chart across several variables, we have to initially generate a vector that contains the required values. First, we define a variable `z` that is pre-assigned the value `NIL`. Using `table()`, we can then create a one-row "table" for each variable of our data frame, containing the frequencies of the attributes `-2`, `-1`, `1`, `2`, `3`, `4`. To do this, we first have to convert the SPSS variable into a matrix using `as.matrix()`. This also counts `-2` and `-1` as values. In my opinion, this is useful: the calculation of agreement should also be based on cases that answered "don't know" or are set to `na`. If you do not share my view, then you would have to exclude those attributes from the calculation. This will not make a visible difference, at least for the latter, since the `na` attributes are limited

to a few cases. A one-row table is now created for every individual variable, the table is assigned the variable `y`, and the portion of the first two attributes on all attributes is calculated using `z<-c(z,100*(y[“1”]+y[“2”])/sum(y))`, and attached to `z` as an additional value. Note that we start at the back when creating tables, i.e., `v165`, then `v164`, etc; this is because the bar chart is plotted from bottom to top (see also Sect. 6.1.1). In `v159`, the attribute `-2` does not occur, which means we have to define this “frequency” as 0 as the first value of the vector `y`. The call of `barplot()` is done as usual: we want to plot `z`, the title is saved in `names`, horizontal layout, no frames around the bars, x-axis range from 0 to 100, colour “orange”, no axes. The object created by the `barplot` function is saved in the `bp` variable, so we can use the coordinates of the columns for the `text()` function in the next line. Each bar’s respective value is written as text at position 10, rounded to one digit after the decimal print and complemented with a percent symbol `%`, in white colour, magnified by a factor of 2. Now all that is left to do, as it were, is to put the titles and captions into the outer margins using `mtext()`.

### 6.1.3 Bar Chart for Multiple Response Questions: All Response Categories

#### It is often said that attitudes towards gender roles are changing



Source: European Values Study 2008 Germany, ZA4753. www.gesis.org. Design: Stefan Fichtel, iextract

If you want the figure to show all categories and not only the first two, then you will end up with a bar chart in which all bars have the same length. They only differ in their subdivisions. In this type of chart, colour selection is important. Sensible choices are colour combinations that are easily associated with agreement and disagreement. The category “na/don’t know” should be set apart from the selected colour scheme; grey is a good choice here. Since there are now five categories, we also need a legend. In the present chart type, it should be placed above the first bar. The alternative is a separate legend as shown in Sect. 7.1.2 or a “reference bar”, similar to that shown in Sects. 7.3.4 and 7.3.5. The width of the legend should correspond to the width of the bars, and it should be easy to associate the elements with the actual data. This means that some category labels are placed to the left and some to the right of the colours.

Data: see annex A, ZA4753: European Values Study 2008: Germany (EVS 2008). We need two files to generate this figure. First, the data are imported. This part was transferred, since we will need it in two other examples.

```
# inc_data_za4753.r
library(memisc)
ZA4753<-spss.system.file("mydata/ZA4753_v1-1-0.sav")
mydata<-subset(ZA4753,select=c(v106,v159,v160,v161,v162,v163,\
v164,v165))
attach(mydata)
z<-NULL
y<-100*table(as.matrix(v165))/length(v165)
z<-rbind(z,y)
y<-100*table(as.matrix(v164))/length(v164)
z<-rbind(z,y)
y<-100*table(as.matrix(v163))/length(v163)
z<-rbind(z,y)
y<-100*table(as.matrix(v162))/length(v162)
z<-rbind(z,y)
y<-100*table(as.matrix(v161))/length(v161)
z<-rbind(z,y)
y<-100*table(as.matrix(v160))/length(v160)
z<-rbind(z,y)
y<-c(0,100*table(as.matrix(v159))/length(v159))
z<-rbind(z,y)
myresponses<-c("n.a./don't know","agree strongly","agree","\
disagree","disagree strongly")
```

Then comes the actual script that generates the figure:

```
pdf_file<-"pdf/barcharts_multiple_all.pdf"
cairo_pdf(bg="grey98", pdf_file, width=13,height=10.5)

par(omi=c(0.0,0.75,1.25,0.75),mai=c(1.6,3.75,0.5,0),lheight\
=1.15,
family="Lato Light",las=1)

# Import data and prepare chart
```

```

source("scripts/inc_labels_za4753.r", encoding="UTF-8")
source("scripts/inc_data_za4753.r", encoding="UTF-8")

myC1<-rgb(0, 208, 226, maxColorValue=255)
myC2<-rgb(109, 221, 225, maxColorValue=255)
myC3<-rgb(255, 138, 238, maxColorValue=255)
myC4<-rgb(255, 0, 210, maxColorValue=255)
mycolours<-c("grey", myC1, myC2, myC3, myC4)

myData0<-cbind(z[, 1]+z[, 2], z[, 3], z[, 4], z[, 5], z[, 6])
myData1<-t(myData0)

# Create chart

x<-barplot(myData1, names.arg=mynames, cex.names=1.1, horiz=T,
           border=NA, xlim=c(0, 100), col=mycolours, axes=F)

# Other elements

px<-c(2, 8, 35, 68, 98); py<-rep(9, 5); tx<-c(-2, 23, 43, 65, 95); ty<-
  rep(9, 5)
points(px, py, pch=15, cex=4, col=mycolours, xpd=T)
text(tx, ty, myresponses, adj=1, xpd=T, family="Lato Light", font=3)
mtext(c(0, 20, 40, 60, 80, 100), at=c(0, 20, 40, 60, 80, 100), 1, line=0, cex
      =0.90)

# Titling

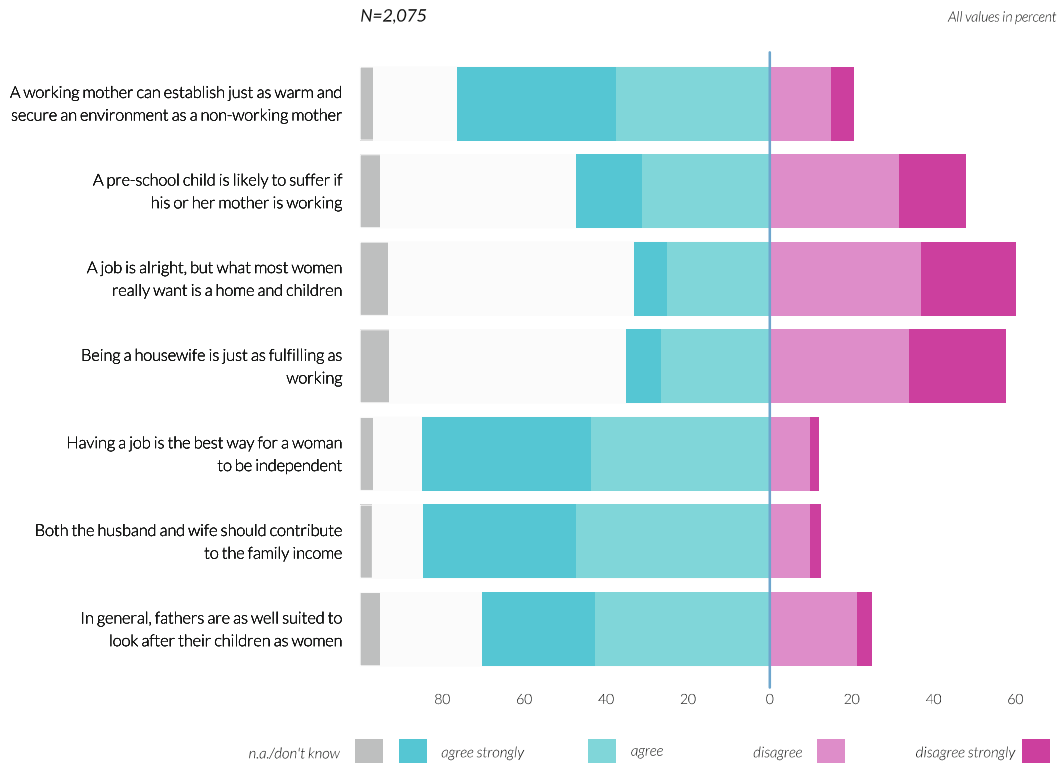
mtext("It is often said that attitudes towards gender roles are
      changing", 3, line=2.2, adj=0, cex=1.8, outer=T, family="Lato
      Black")
mtext("All values in percent", 1, line=2, adj=1, cex=0.95, font=3)
mtext("Source: European Values Study 2008 Germany, ZA4753. www.
      gesis.org. Design: Stefan Fichtel, iextract", 1, line=4.5, adj
      =1, cex=0.95, font=3)
mtext("N=2,075", 1, line=2, adj=0, cex=1.15, family="Lato", font=3)
dev.off()

```

The script starts with the settings for window size and margins, followed by import of first the labels and then the data. Colour definitions match those from the previous example. With the data, the first and second column are combined, then transposed for plotting in the `barplot()` function. At the end, positions for the legend are specified. Because of the exact placement, we do not use the `legend()` function; instead, we add the colours and texts using the functions `points()` and `text()`. The last step is to add the axes labels, titles, and captions.

### 6.1.4 Bar Chart for Multiple Response Questions: All Response Categories, Variant

#### It is often said that attitudes towards gender roles are changing



The figure is a variation of the one from Sect. 6.1.3 and centres the bars around zero. This means that agreements are right-justified as stacked bars, while disagreements are adjacent to those and left-justified. The “no answer/don’t know” category should stay separated on the left side. In this case, the legend has been placed below the chart instead of above it. Its layout matches the previous example.

Data: see annex A, ZA4753: European Values Study 2008: Germany (EVS 2008).

```
pdf_file<-"pdf/barcharts_multiple_all_2.pdf"
cairo_pdf(bg="grey98", pdf_file, width=13,height=10.5)

par(omi=c(0.25,0.75,1,0.75),mai=c(1.8,3.75,0.25,0),lheight=1.15,
     family="Lato Light",las=1)
library(RColorBrewer)

# Import data and prepare chart

source("scripts/inc_labels_z4753.r",encoding="UTF-8")
source("scripts/inc_data_z4753.r",encoding="UTF-8")

myC1<-rgb(0,208,226,maxColorValue=255)
```

```

myC2<-rgb(109, 221, 225, maxColorValue=255)
myC3<-rgb(255, 138, 238, maxColorValue=255)
myC4<-rgb(255, 0, 210, maxColorValue=255)
colours<-c("grey", myC1, myC2, myC3, myC4)

myData0<-cbind(z[,1]+z[,2], z[,3], z[,4], z[,5], z[,6])
myData1<-t(myData0)

# Create chart

barplot(-rep(100, 7), names.arg=mynames, cex.names=1.1, horiz=T, \
        border=par("bg"), xlim=c(-100, 70), col=colours[1], axes=F)
barplot(-(100-myData1[1, ]), names.arg=mynames, cex.names=1.1, horiz=\
        =T, border=par("bg"), xlim=c(-100, 70), col=par("bg"), axes=F, \
        add=T)
barplot(-myData1[3:2, ], names.arg=mynames, cex.names=1.1, horiz=T, \
        border=NA, xlim=c(-100, 70), col=colours[3:2], axes=F, add=T)
barplot(myData1[4:5, ], names.arg=mynames, cex.names=1.1, horiz=T, \
        border=NA, xlim=c(-100, 70), col=colours[4:5], axes=F, add=T)

# Other elements

arrows(0, -0.1, 0, 8.6, lwd=2.5, length=0, xpd=T, col="skyblue3")
px<-c(-98, -87, -41, 15, 65); tx<-c(-105, -60, -26, 8, 60); y<-rep(-1, 5)
points(px, y, pch=15, cex=4, col=colours, xpd=T)
text(tx, y, myresponses, adj=1, xpd=T, font=3)
mtext(c(80, 60, 40, 20, 0, 20, 40, 60), at=c(-80, -60, -40, -20, 0, 20, 40, 60)\
      , 1, line=0, cex=0.95)

# Titling

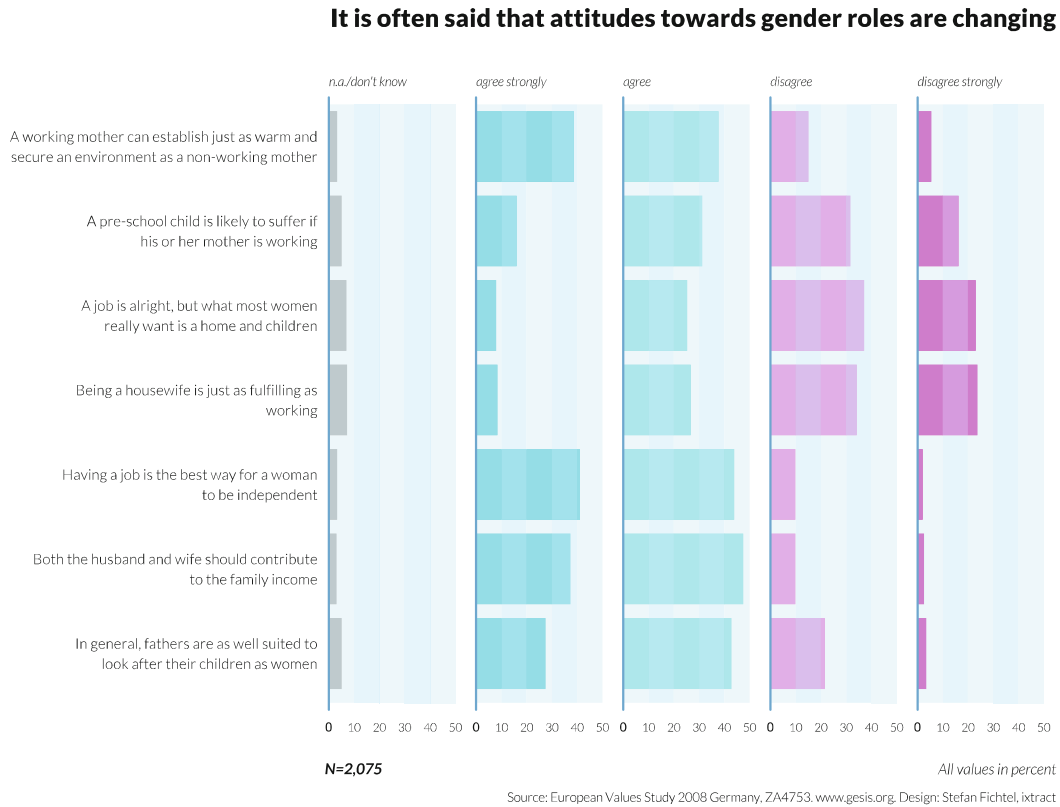
mtext("It is often said that attitudes towards gender roles are \
      changing", 3, line=2.2, adj=0, cex=1.8, outer=T, family="Lato \
      Black")
mtext("All values in percent", 3, line=1, adj=1, cex=0.95, font=3)
mtext("Source: European Values Study 2008 Germany, ZA4753. www.\
      gesis.org. Design: Stefan Fichtel, iextract", 1, line=5.2, adj\
      =1, cex=0.95, font=3)
mtext("N=2,075", 3, line=1, adj=0, cex=1.15, family="Lato", font=3)
dev.off()

```

The first half of the script matches the previous example. The most important difference is that the `barplot()` function is not called once, but four times. First, seven grey bars ranging from 0 to  $-100$  are plotted. Using `add=T`, these bars are overlaid with the percent values of the “na/don’t know” category as negative difference from

100 in the background colour. After that, only the exact “na/don’t know” portion of the grey bars remains visible. The third barplot() call plots the third and second category as stacked bars, again as negatives; the fourth barplot() plots the fourth and fifth category, again as stacked bars, but now in positive direction. Using arrows(), the zero line is highlighted in the colour “skyblue3”. The rest matches the previous example; however, the negative labelling of the x-axis has to be replaced with the respective positive values in this example.

### 6.1.5 Bar Chart for Multiple Response Questions: All Response Categories (Panel)





The figure shows the use of a panel as another alternative, in which the bars for the individual attributes are each drawn separately. In this type of presentation, the differences between the individual attributes stand out clearly. The labels of the individual attributes are now titles, and instead of a single x-axis, a separate one should be provided for each response category.

Data: see annex A, ZA4753: European Values Study 2008: Germany (EVS 2008).

```
pdf_file<-"pdf/barcharts_multiple_all_panel.pdf"
cairo_pdf(bg="grey98", pdf_file, width=13,height=10.5)

par(omi=c(1.25,1.25,1.25,0.25),lheight=1.15,family="Lato Light",\
    las=1)
library(RColorBrewer)

# Import data and prepare chart

source("scripts/inc_labels_z4753.r")
source("scripts/inc_data_z4753.r")
layout(matrix(data=c(1,2,3,4,5),nrow=1,ncol=5), widths=c(\
    2.5,1,1,1,1),heights=c(1,1))

myData1<-cbind(z[,1]+z[,2],z[,3],z[,4],z[,5],z[,6])
tmyData<-myData1
DD_pos<-c(45,45,45,45,35)
myC1<-rgb(0,208,226,maxColorValue=255)
myC2<-rgb(109,221,225,maxColorValue=255)
myC3<-rgb(255,138,238,maxColorValue=255)
myC4<-rgb(255,0,210,maxColorValue=255)
colours<-c("grey",myC1,myC2,myC3,myC4)

# Create chart

for (i in 1:5) {
  if (i == 1)
  {
    par(mai=c(0.25,2.75,0.25,0.15))
    bp1<-barplot(tmyData[,i],horiz=T,cex.names=1.6,names.arg=\
        mynames,
        xlim=c(0,50),col=colours[i],border=NA,axes=F)
  } else
  {
    par(mai=c(0.25,0.1,0.25,0.15))
    bp2<-barplot(tmyData[,i],horiz=T,axisnames=F,xlim=c(0,50),col=\
        colours[i],border=NA,axes=F)
  }
}

# Other elements

rect(0,0,10,8.5,col=rgb(191,239,255,80,maxColorValue=255),border\
    =NA)
rect(10,0,20,8.5,col=rgb(191,239,255,120,maxColorValue=255),\
    border=NA)
```

```

rect(20, 0, 30, 8.5, col=rgb(191, 239, 255, 80, maxColorValue=255), \
     border=NA)
rect(30, 0, 40, 8.5, col=rgb(191, 239, 255, 120, maxColorValue=255), \
     border=NA)
rect(40, 0, 50, 8.5, col=rgb(191, 239, 255, 80, maxColorValue=255), \
     border=NA)

mtext(myresponses[i], 3, adj=0, line=0, cex=0.95, font=3)
mtext(c(10, 20, 30, 40, 50), at=c(10, 20, 30, 40, 50), 1, line=1, cex=0.85)
mtext(0, at=0, 1, line=1, cex=0.90, family="Lato Bold")
arrows(0, -0.1, 0, 8.6, lwd=2.5, length=0, xpd=T, col="skyblue3")
}

# Titling

mtext("It is often said that attitudes towards gender roles are \
      changing", 3, line=3.5, adj=1, cex=1.8, family="Lato Black", \
      outer=T)
mtext("N=2,075", 1, line=3, adj=0.25, cex=1.1, family="Lato", font=4, \
      outer=T)
mtext("All values in percent", 1, line=3, adj=1, cex=1.1, font=3, \
      outer=T)
mtext("Source: European Values Study 2008 Germany, ZA4753. www.\
      gesis.org. Design: Stefan Fichtel, iextract", 1, line=5.5, adj\
      =1.0, cex=0.95, outer=T)
dev.off()

```

As in the previous bar chart variants, the script starts with the import of labels and data. Then we define five windows. We use the `layout()` function for this, since the first window has to be wider than the other four to leave space for the item texts. Otherwise, we could also use the `mfcpl` function. The output window is split so that it is made up of one row and five columns, the first being twice as wide as the others and all being of equal height. The data are compiled as in the previous example. The five bar charts are plotted within the `for()` loop. Here, we distinguish between the first and the other four bar charts: we require a wider margin for the labels in the first one, while 0.1 suffices for the others. In this case, the first argument of the `barplot()` function, the data, is column `i` of the `tmyData` matrix. The background is defined as in the first example (Sect. 6.1.1). The labels are once more added at the end of the script; the attributes from the response vector are simply left-justified and laid over the existing figure. Using `arrows()`, a blue line is set at the location of the zero line as visual separator. This finalises the loop; titles and subtitles follow.

### 6.1.6 Bar Chart for Multiple Response Questions: Symbols for Individuals

#### It is often said that attitudes towards gender roles are changing

Agree strongly / agree



About the figure: Otto Neurath is probably the most adept in the art of visualising statistics in a way that shows the massiveness of the facts. A characteristic of Otto Neurath's figures is that everything is very orderly. All the persons are always lined up so they are easily countable. Now, while statistics is always an orderly mass in its end result, it does also contain elements that are varied, random, and indistinct. This can be visualised by, e.g., persons that are randomly spread over an area. The number of persons corresponds to the frequency of the measured variable, but one also gets an impression of the statistical character of the presented parameter. The example from Sect. 6.1.1 is reused in the present example, but we limit ourselves to the first three from the group of questions. For explanatory reasons, we also distinguish between men and women. The persons are placed randomly, but so that their number is visible. The legend has to state that every figure represents 10 people and, obviously, also which symbol was chosen for which representation—even if it is apparent from the form of the symbol.

Data: see annex A, ZA4753: European Values Study 2008: Germany (EVS 2008).

```

pdf_file<-"pdf/barcharts_multiple_symbols.pdf"
cairo_pdf(bg="grey98", pdf_file,width=13,height=10.5)

par(omi=c(0.65,0.65,0.85,0.85),mai=c(1.1,5.85,1.55,0),family="\
  Lato Light",las=1)

# Prepare chart

col_f<-rgb(255,97,0,190,maxColorValue=255)
col_m<-rgb(68,90,111,190,maxColorValue=255)
source("scripts/inc_labels_z4753.r")

# Create chart

plot(1:5,type="n",axes=F,xlab="",ylab="",xlim=c(0,20),ylim=c\
  (1,6))
mySymbols<-function(n_f,n_m,y,labelling,... ){
  par(family="Symbol Signs")
  for (i in 1:n_f)
  {
    text(runif(1,0,(n_f+ n_m)/10),runif(1,y,y+1),"F",cex=3.25,
      col=col_f)
  }
  for (i in 1:n_m)
  {
    text(runif(1,0,(n_f+ n_m)/10),runif(1,y,y+1),"M",cex=3.25,
      col=col_m)
  }
  par(family="Lato Light")
  text(-3,y+0.5,labelling,xpd=T,cex=1.45,adj=1)
}
mySymbols(round(336/10),round(350/10),1,myC_v161)
mySymbols(round(454/10),round(525/10),3,myC_v160)
mySymbols(round(865/10),round(720/10),5,myC_v159)
axis(1,at=c(0,5,10,15,20),labels=c("0","500","1,000","1,500","\
  2,000"),col=par("bg"),col.ticks="grey81",lwd.ticks=0.5,tck\
  =-0.025)

# Other elements

abline(v=c(0,5,10,15,20),lty="dotted")

# Titling

mtext("It is often said that attitudes towards gender roles are \
  changing",3,line=-0.5,adj=0,cex=1.8,family="Lato Black",\
  outer=T)
mtext("Agree strongly / agree",3,line=-3,adj=0,cex=1.8,outer=T,\
  font=3)
mtext("Source: EVS 2008 Germany, ZA4753",1,line=0,adj=1,cex=1.5,\
  outer=T,font=3)
mtext("2,075 respondents. Every figure represents 10 people ",1,\
  line=-2,adj=0,cex=1.5,outer=T,font=3)

```

```

par(family="Lato Light")
mtext(" Women", 1, line=1, adj=0.02, cex=1.5, outer=T, font=3)
mtext(" Men", 1, line=1, adj=0.12, cex=1.5, outer=T, font=3)
par(family="Symbol Signs")
mtext("F", 1, line=1, adj=0, cex=2.5, outer=T, font=3, col=col_f)
mtext("M", 1, line=1, adj=0.1, cex=2.5, outer=T, font=3, col=col_m)
dev.off()

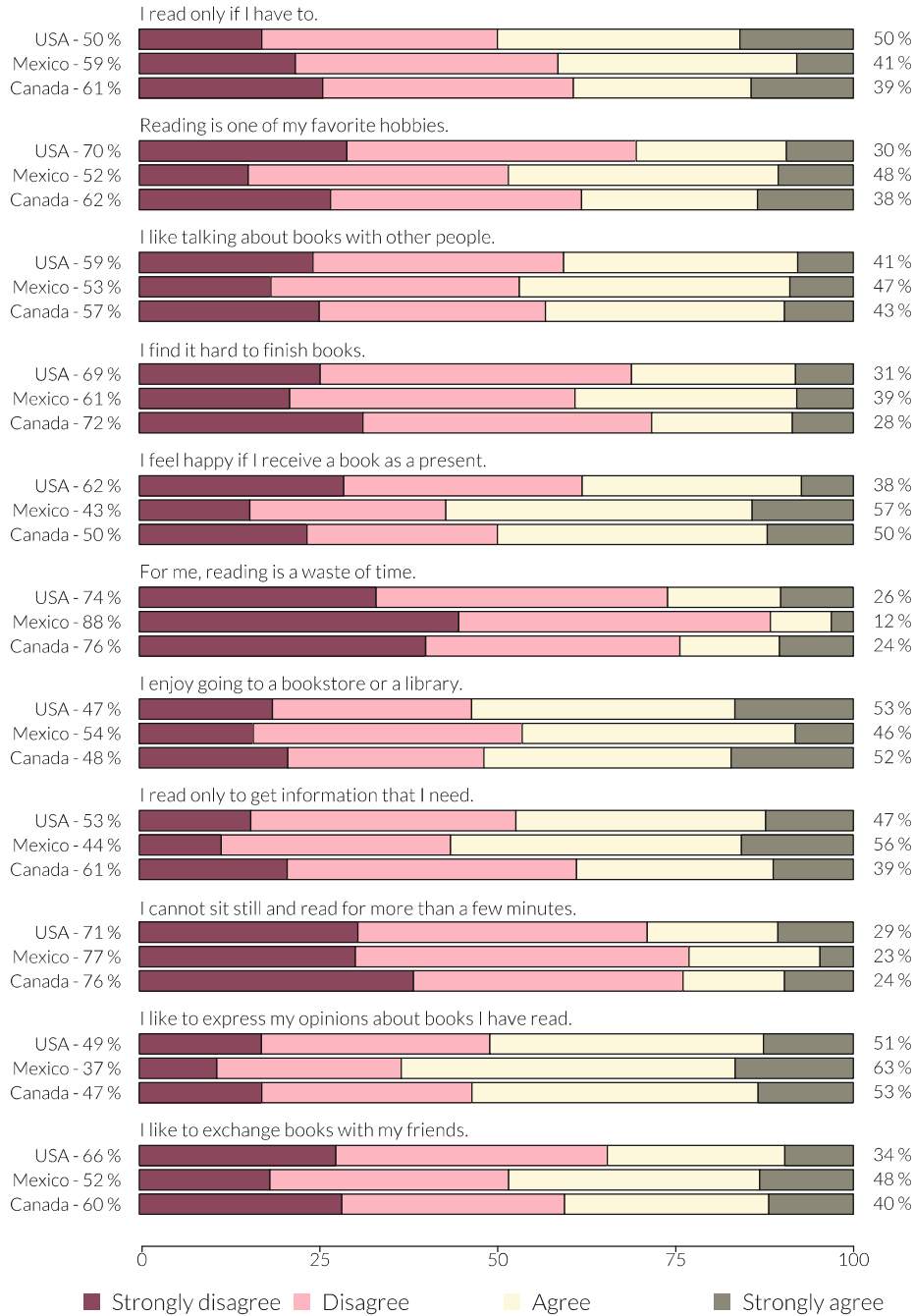
```

Within the script, programming differs from the previous examples, because we cannot use the high-level function `barplot()` or `barp()`. Instead, we import the labels and then first create an “empty” chart using `plot()` and the parameter `type="n"` (that means we only “define” the chart). We do not want to plot axes or axes labels; however, the range of values is specified as 0–20 in x-direction, and as 1–6 in y-direction. This is followed by the definition of a function `symbols` to which we transfer the parameters `n_f`, `n_m`, `y`, and labelling. Further parameters can be transferred with the “...” expression. These are then passed through to the functions used within the function. Within the `symbols` function, the font “Symbol Signs” is first selected. Then we use `text()` in two loops to create text output at positions generated with random numbers. We use the `runif()` function for number generation. The function is usually called in the form of `runif(NumberOf random numbers, interval start, interval end)`. Since we always use the function within a loop, the amount of numbers to be generated per run is always 1. The x-position is then a uniformly distributed random number in the interval from 0 to the sum of men and women, divided by 10; the y-position is a uniformly distributed random number in the interval from `y` to `y + 1`, with `y` being a value transferred to the function. Font size is greatly increased using `cex=2.75`; colour is red for women and blue for men, both transparent. After the two loops, the font is switched back to “Lato Light” and the labelling that was also transferred to the function is inserted at position `-3, y+0.5`. Now we can use the `symbols` function to call up the frequencies divided by ten of men and women as integer values for each variable of the group of questions that is to be displayed. The only reason for the division by 10 is scaling. The transferred parameters are the number of men and women, the y-position, and the labels. The y-positions for the three variables are 1, 3, and 5. Last but not least, the labels and markers have to be attached. We first draw the x-axis with the labels “0”, “500”, “1,000”, “1,500”, and “2,000” at the positions 0, 5, 10, 15, and 20, respectively. Dotted lines are added for guidance. Then, titles and legends are attached.

### 6.1.7 Bar Chart for Multiple Response Questions: All Response Categories, Grouped

## Reading attitude

How much do you disagree or agree with these statements about reading?



Source: PISA 2009 Assessment Framework - Key Competencies in Reading, Mathematics, and Science

© OECD 2009, Data: bryer.org

The figure is a useful extension compared with Sect. 6.1.3, due to the additional consideration of a grouping variable. The present example again deals with presenting a group of questions. Here, it revolves around reading behaviour derived from the PISA study from 2009. There were 11 statements that could once more be answered with “strongly disagree”, “disagree”, “agree”, “strongly agree”. Presented are the respective percentages of the response categories, differentiated between responses from US, Mexican, and Canadian students. In addition to Sect. 6.1.3, percentage values are printed to the left and right of bars to show the agreement or disagreement for a certain question and a certain country. In my opinion, the fact that this corresponds to the sums of “strongly disagree” and “disagree” or “agree” and “strongly agree” does not need to be explicitly stated; it is clear from the figure. In contrast to the preceding figures, we have chosen red/light red for the negative and yellow/brown for the positive attributes. The texts of the individual questions of the group are written directly above each bar group. The illustration form is based on a suggestion by Jason Bryer.

The data originate from a survey on the international comparison of school performance study by the Programme for International Student Assessment (PISA) from 2009. This survey, conducted on behalf of the Organisation for Economic Co-operation and Development (OECD), recorded the abilities of 15-year-old students in the areas of reading competency, mathematical competency, and scientific competency in the OECD states and in 33 OECD partner countries. Since 2000, the survey has been conducted every 3 years. Data for the USA, Canada and Mexico were provided by Jason Bryer in R file format. The data record comprised 305 variables and 66,690 respondents.

```
pdf_file<-"pdf/barcharts_multiple_all_grouped.pdf"
cairo_pdf(bg="grey98", pdf_file,width=12,height=19)

par(omi=c(1.0,0.5,1.75,0.5),mai=c(0.1,1.45,0.35,0.8),
    family="Lato Light",las=1)

# Import data and prepare chart

load("myData/pisana.rda")
items28=pisana[,substr(names(pisana),1,5) == 'ST24Q']
source("scripts/inc_names_item28.r")

for(i in 1:ncol(items28))
{
items28[,i]=factor(items28[,i],levels=1:4,ordered=T)
}
source("scripts/functions/lickert.r")
library(reshape)
lik=likert(items28,grouping=pisana$CNT)
x<-print(lik); y<-cbind(x[,1],x[,3],x[,4],x[,5],x[,6])
colours<-c("palevioletred4","lightpink","cornsilk1","cornsilk4")
k<-length(y[,1])/length(unique(y[,1]))
par(mfcol=c(k+1,1),las=1)
```

```

for (i in 1:k)
{
z<-y[c(i,i+k,i+2*k),]
prozcan_l<-format(round(z[1,2]+z[1,3],0),nsmall=0)
prozmex_l<-format(round(z[2,2]+z[2,3],0),nsmall=0)
prozusa_l<-format(round(z[3,2]+z[3,3],0),nsmall=0)
prozcan_r<-format(round(z[1,4]+z[1,5],0),nsmall=0)
prozmex_r<-format(round(z[2,4]+z[2,5],0),nsmall=0)
prozusa_r<-format(round(z[3,4]+z[3,5],0),nsmall=0)
b1<-paste("Canada","-",prozcan_l,"%",sep=" ")
b2<-paste("Mexico","-",prozmex_l,"%",sep=" ")
b3<-paste("USA","-",prozusa_l,"%",sep=" ")

# Create chart

barplot(t(z[,2:5]),names.arg=c(b1,b2,b3),cex.names=2,
        col=colours,horiz=T,axes=F)
text(105.5,1.0-0.25,paste(prozcan_r,"%",sep=" "),xpd=T,cex=2)
text(105.5,2.2-0.25,paste(prozmex_r,"%",sep=" "),xpd=T,cex=2)
text(105.5,3.4-0.25,paste(prozusa_r,"%",sep=" "),xpd=T,cex=2)
text(0,4.3,names(items28)[i],cex=2.1,xpd=T,adj=0)
}

# Other elements

par(mai=c(1.1,1.225,0,0.45))
plot(1:2,typ="n",axes=F,xlim=c(0,100),xlab="",ylab="")
axis(1,at=c(0,25,50,75,100),cex.axis=2)
legend(-10,-0.5,pt.cex=4,cex=2.5,pch=15,col=colours,ncol=4,c("\
  Strongly disagree","Disagree","Agree","Strongly agree"),\
  bty="n",xpd=T)

# Titling

mtext("Reading attitude",3,line=5.5,adj=0,cex=3.8,family="Lato \
  Black",outer=T)
mtext("How much do you disagree or agree with these statements \
  about reading?",3,line=2.2,adj=0,cex=2.0,outer=T)
mtext("Source: PISA 2009 Assessment Framework - Key Competencies \
  in Reading, Mathematics, and Science",1,line=1,adj=1.0,\
  cex=1.25,outer=T)
mtext("© OECD 2009, Data: bryer.org",1,line=3.5,adj=1.0,cex\
  =1.25,outer=T)
dev.off()

```



The script first defines a large window (12 by 19 inches) and individual margin settings. The data record is already available in binary R format and is loaded using `load()`. The following rows are taken from the example script by Jason Bryer for the illustration of his visualisation of a Likert scale: first, the items of question group 28 are extracted by limiting the data record to variables whose name starts with “ST24Q”. This is done by addressing the data record with square brackets as a matrix: `items28 = pisana[,substr(names(pisana), 1,5) ==, ST24Q’]`. Within the square brackets, the first character is a comma; this ensures that all rows are carried over into the new selection. Column selection is made via the `substr()` function containing the stated condition. The variables thereby selected are then assigned the statements of the question group 28 as names using `names()`; then all the variables are converted to “factors” or “categorical variables” in a loop. Now, the `likert()` function is read, which we downloaded from <http://github.com/jbryer/pisa>. We still need Hadley Wickham’s `reshape` package for the `cast()` function that is called by the `likert()` function. Using `print()`, the result is saved in the variable `x`; then `y` is used to create a matrix that only contains the groups and percent values:

```
>y
  [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1 25.69810 35.12856 24.883834 14.289507
[2,] 1 26.77758 35.18871 24.636078 13.397637
[3,] 1 25.22917 31.68150 33.470617  9.618706
[4,] 1 31.33106 40.44088 19.698110  8.529946
[5,] 1 23.48092 26.65869 37.827417 12.032974
.
.
.
```

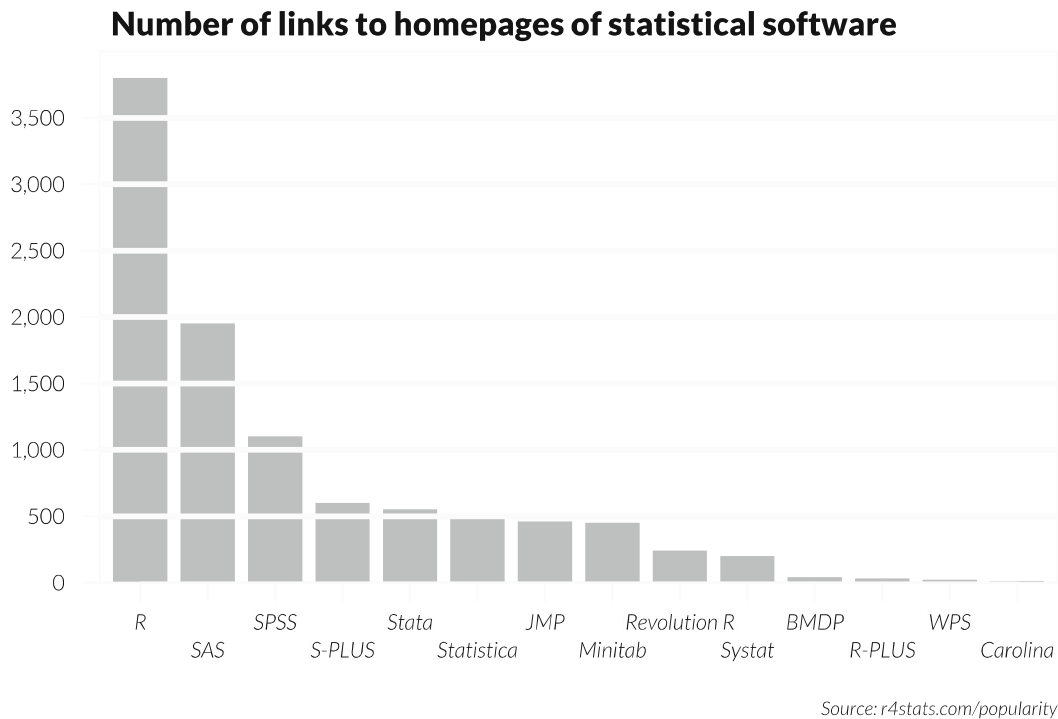
In the next step, every variable is assigned a colour. Here, we use known colours that are included in R: “palevioletred4”, “lightpink”, “cornsilk1”, and “cornsilk4”. `k` is the number of items, here exemplarily calculated from the number of `y` rows divided by the number of different attributes of the grouping variable. The basic idea of the figure is to draw an individual chart for each item. Additionally, an extra chart window is required to draw the value axes and a legend. Therefore, we use `par(mfcol=c(k+1,1), las=1)` to split the illustration window into `k+1=12` rows of equal size and one column. `las=1` causes horizontal labelling. Now follows a loop with `k=11` passes. Since the start matrix `y` is sorted, `z` is a vector with the three values for each question for Canada, Mexico, and the USA, respectively: in the first run, every 1st, 11th, and 21st value, in the second every 2nd, 12th, and 22nd, etc. `b1`, `b2`, and `b3` then result in the labels that are made up of the country’s name followed by a dash and the sum of the first two item percentages. Numbers are rounded and fitted with a percent symbol. This is followed by the `barplot()` call. The second

to fifth column of  $z$ , i.e. the 4% columns, of the transposed matrix are plotted—otherwise the rows would be plotted instead. As per definition, the bars go to 100%, and therefore the sum of the two right percent columns can be written directly after the bar, at position 105.5, and at the height of the centre of the bar using `text()`. The height of the centre of the bar is: bar width (1), from the second bar onwards additionally the bar distance (0.2), and then back 0.25. The text is positioned outside of the actual chart, so it is important to set the parameter `xpd=TRUE`, or the text will not appear. Finally, the item is written horizontally at the start of the bar and vertically above the highest bar at position 4.3. Here again, `xpd=TRUE` has to be set, the font is increased by a factor of 2.1, and text has to be left justified using `adj=0` to make sure it really starts at  $x=0$ . With the loop, everything is repeated 11 times. In the 12th window, an invisible `plot()` is drawn—to do this, simply set 1:2 as value, no axes, and `type="n"`—then an axis with percentages at 0, 25, 50, 75, and 100 is added. Underneath goes a legend for the entire figure. Since this is the first time we have used the `legend()` function, there should be a short explanation: we are positioning it at coordinates of the invisible plot,  $-10$  and  $-0.5$ . These are coordinates outside of the plot area, so once more `xpd=TRUE` has to be set. The vector `c()` contains the four attributes of the items, point size is to be increased by a factor of 4 from the preceding value, and the font by a factor of 2.5. We use 15 as a symbol, which is a filled square. Obviously, the colours have to correspond to the items, so the vector `colours` is used again; `ncol=4` means that the legend should be quadrifid. The default is `ncol=1`, a vertical arrangement. With the box type `bty="n"`, no frame is drawn around the legend; this has to be explicitly stated. In this last window, the margins are set so the plotted x-axis ends flush left- and right-justified with the bars of the other windows. In this case, suitable values were found by trial and error. Finally, as in the previous examples, come the titles—again using `mtext()` for flexibility's sake, and in the outer margin with the `outer=TRUE` parameter. The best settings for font size, line position, alignment and, above, for the margins, are once more a matter of trial and error. If we want to first sort according to country and within those according to the item (therefore not 11 times 3, but 3 times 11 bars), then we would have to set `k<-unique(y[,1])` and rewrite the loop slightly:

```
for (i in 1:length(k))
{
z<-subset(y,y[,1]==i)
barplot(t(z[,2:5]), names.arg=names(items28), horiz=TRUE)
}
```

The preceding examples should cover a large part of the practical applications for bar charts. Now follow examples that each illustrate a single aspect of representation.

### 6.1.8 Column Chart with Two-Line Labelling



The figure shows the number of links to homepages of statistical software as a column chart, for illustrative purposes. However, the labels would be too long here and would overlap; alternatively, the bars would have to be spaced far apart, so the figure would not look nice. We have two options: We organise our labels either in two rows or at a 45° angle. Other than that, the chart was purposefully kept simple and follows the “reduced appearance” proposal by Edward Tufte. With bar charts, we should go without grid lines and rather “break” the columns with white lines. Once more, there are no aspect ratios resulting from the presented numbers. Sometimes, the size and ratio has to be chosen dependent on the page layout; that should be the exception though. If the figure is embedded in a text, then an aspect ratio of, e.g., 7:5 is a good choice. With two-line labelling of the x-axis, the individual labels should obviously not be too long. In the present case, this does not pose a problem. Since the largest column is against the y-axis, and the columns are visually subdivided, we can do without tick marks on the axis. Values exceeding 1000 are frequently formatted with a “thousands comma”. The common use of a space instead of a comma, especially in official statistical publications, is generally

less suited for figures. Scaling of the y-axis does not necessarily have to include the maximal attributes of the values. Data are derived from a figure on <http://r4stats.com/popularity>.

```
pdf_file<-"pdf/columncharts_labels_tworows.pdf"
cairo_pdf(bg="grey98", pdf_file,width=7,height=5)

library(plotrix)
library(gdata)

par(mai=c(0.95,0.5,0.0,0.5),omi=c(0,0.5,1.0,0),fg=par("bg"),\
    family="Lato Light",las=1)

# Import data and prepare chart

links<-read.xls("myData/listserv_discussion_traffic.xlsx",sheet\
    =2, encoding="latin1")
attach(links)
sort.links<-links[order(-Number) ,]
myNames<-sort.links$Software
myNumber<-sort.links$Number
py<-c(0,500,1000,1500,2000,2500,3000,3500)
fpy<-format(py,big.mark=",")

# Create chart and other elements

barp(myNumber,cex.axis=0.75,names.arg=myNames,border=NA,col="\
    grey",staxx=T,ylim=c(0,4000),height.at=py,height.lab=fpy)
par(col="black")
staxlab(1,1:length(myNames),myNames,nlines=2,top.line=0.55,font\
    =3,cex=0.75)
abline(h=c(500,1000,1500,2000,2500,3000,3500),col=par("bg"),lwd\
    =3)

# Titling

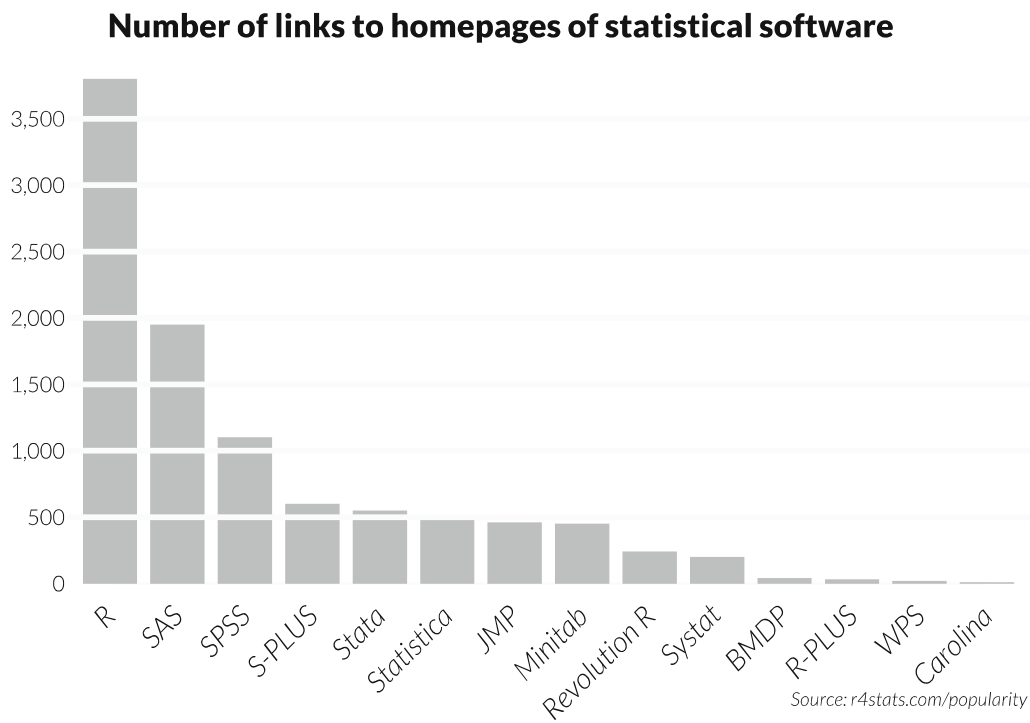
mtext("Number of links to homepages of statistical software",3,\
    line=2,adj=0,cex=1.4,family="Lato Black",outer=T)
mtext("Source: r4stats.com/popularity",1,line=3,adj=1.0,cex\
    =0.65,font=3)
dev.off()
```

In the script, we directly define the aspect ratio of 7:5 as a parameter in inches. The procedure of data import and sorting is the same as described in Sect. 6.1.1. With the minus, sorting is done in descending order. To achieve two-line labelling, we use the plotrix package by Jim Lemon. On the one hand, this package provides the extended barplot function `barp()`, and on the other hand, the `staxlab()` function

that allows multiple-line axis labels. The `barp()` arguments largely match those of `barplot()`. It is important to set `staxx=TRUE`, which does not output any x-axis labelling, but leaves this to the subsequent `staxlab()` call. We only have to specify that the parameters refer to axis 1 (x-axis), the positions of the labels (`1:length(names)`), that the labels are contained within the `names` vector, that the number of rows equals two, and finally that the distance between the axis and the first row is to be 0.75. The parameter `big.mark=","` of the `format` function causes a thousands separator. We used `fg=par("bg")` at the beginning to hide the line and x-axis labels that the `barp()` function draws by default. Specification of the background colour using `fg` also impacts on the `staxlab` output, so we now have to again set the colour using `col="black"`. Using `abline`, lines are drawn over the bars at the position of the y-labels (except the zero) in the background colour.

### 6.1.9 Column Chart with 45° Labelling

The figure is a variant of the reduced illustration from the previous example using 45° labels. Such a form of representation will not convince everybody. In my opinion, it may well be used though—as opposed to vertically aligned labels that completely run against the flow of reading. The advantage over the previous variant is that the labels can be larger. In a confined space for the figure, this type might be an alternative.



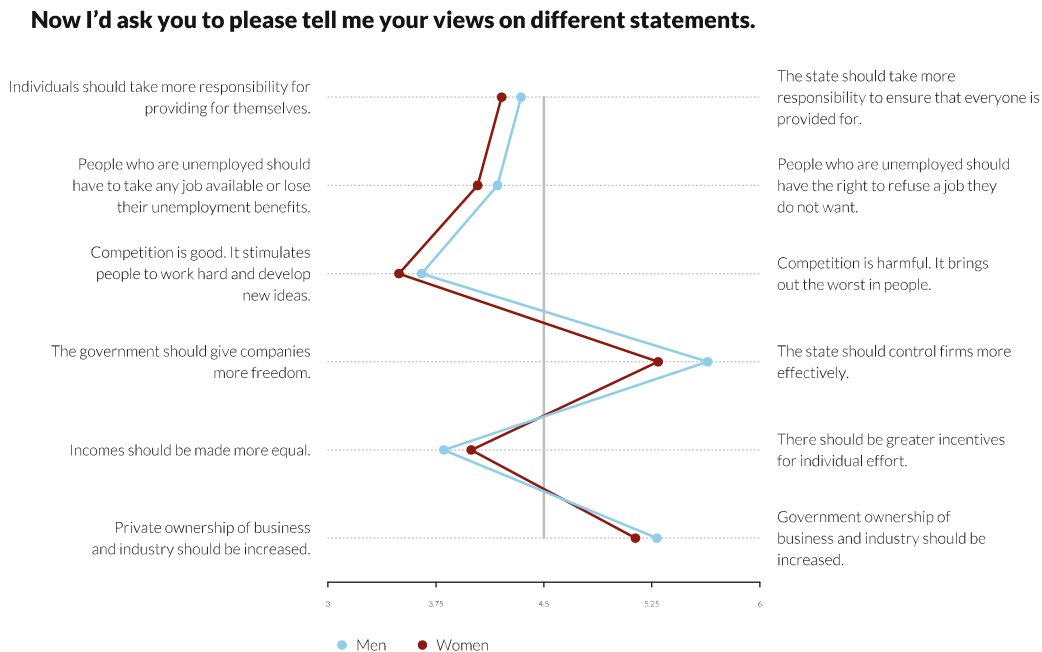
Data are again taken from the respective figure on <http://r4stats.com/popularity>. The script matches the previous example; the only difference is the `staxlab` row, which is changed to

```
staxlab(1, 1:length(names), names, srt=45, top.line=1.25)
```

and the file name is adjusted in the first row.

### 6.1.10 Profile Plot for Multiple Response Questions: Mean Values of the Responses

About the figure: So far, the figures have had in common that they had one label per item. Sometimes though, one might want to also plot parameters that are located on a scale between two statements. In such cases, the use of a profile plot makes sense. The example shows such a profile plot for a series of average estimates that lie between two statements. Here, we are again using a group of questions from the European Values Study 2008. The task was to voice one's opinion on different statements. Two opposing statements were presented to each respondent, and the respondents were asked to rank their answer on a scale between the statements. The connection of the averages plotted as a line chart shows the profile, e.g., for a selected group. Such profiles can be stacked so several groups can be compared. The figure shows profiles for women and men with respect to statements concerning the economy.



Source: ZA4753: European Values Study 2008: Germany (EVS 2008), N=2,075

Data: see annex A, ZA4753: European Values Study 2008: Germany (EVS 2008).

```
pdf_file<-"pdf/profile_chart.pdf"
cairo_pdf(bg="grey98", pdf_file, width=12, height=8)

source("scripts/functions/profile.plot.r")
par(lheight=1.15,mai=c(0.25,0.25,0.25,0.25), omi=c(
  0.5,0.5,1.1,0.5),
  family="Lato Light",las=1)

# Import labels

text.left<-NULL
text.right<-NULL
text.left<-c(text.left,"Individuals should take more \
responsibility for\nproviding for themselves.")
text.right<-c(text.right,"The state should take more\
nresponsibility to ensure that everyone is\nprovided for.\
")
text.left<-c(text.left,"People who are unemployed should\nhave \
to take any job available or lose\ntheir unemployment \
benefits.")
text.right<-c(text.right,"People who are unemployed should\nhave\
the right to refuse a job they\ndo not want.")
text.left<-c(text.left,"Competition is good. It stimulates\
npeople to work hard and develop\nnew ideas.")
text.right<-c(text.right,"Competition is harmful. It brings\nout\
the worst in people.")
text.left<-c(text.left,"The government should give companies\
nmore freedom.")
text.right<-c(text.right,"The state should control firms more\
neffectively.")
text.left<-c(text.left,"Incomes should be made more equal.")
text.right<-c(text.right,"There should be greater incentives\
nfor individual effort.")
text.left<-c(text.left,"Private ownership of business\nand \
industry should be increased.")
text.right<-c(text.right,"Government ownership of\nbusiness and \
industry should be\nincreased.")

# Import and prepare data

library(Hmisc)
ZA4753<-spss.get("myData/ZA4753_v1-1-0.sav",use.value.labels=F)
myVariables<-c("v302","v194","v195", "v196","v197","v198","v199"\
)
myResult<-dim(2)
myData<-ZA4753[, myVariables]
for (i in 2:length(myVariables))
{
  mySelection<-subset(myData[, c(1, i)], myData [, i] >= 1 & \
myData [, i] <= 10)
```

```

myValues<-t(aggregate(mySelection [, 2], by=list(mySelection \
[, 1]), FUN=mean, na.rm=T))
myResult<-rbind(myResult, myValues[2, ])
}

# Create chart

colnames(myResult)=c("Men", "Women")
myC1<-"skyblue"
myC2<-"darkred"
profile.plot(myResult, text.left, text.right, colours=c(myC1, myC2), \
legend.n.col=2)

# Titling

mtext("Now I'd ask you to please tell me your views on different \
statements.", 3, line=3, adj=0, cex=1.5, family="Lato Black", \
outer=T)
mtext("Source: ZA4753: European Values Study 2008: Germany (EVS \
2008). N=2,075", 1, line=1, adj=1.0, cex=1.1, font=3, outer=T)
dev.off()

```

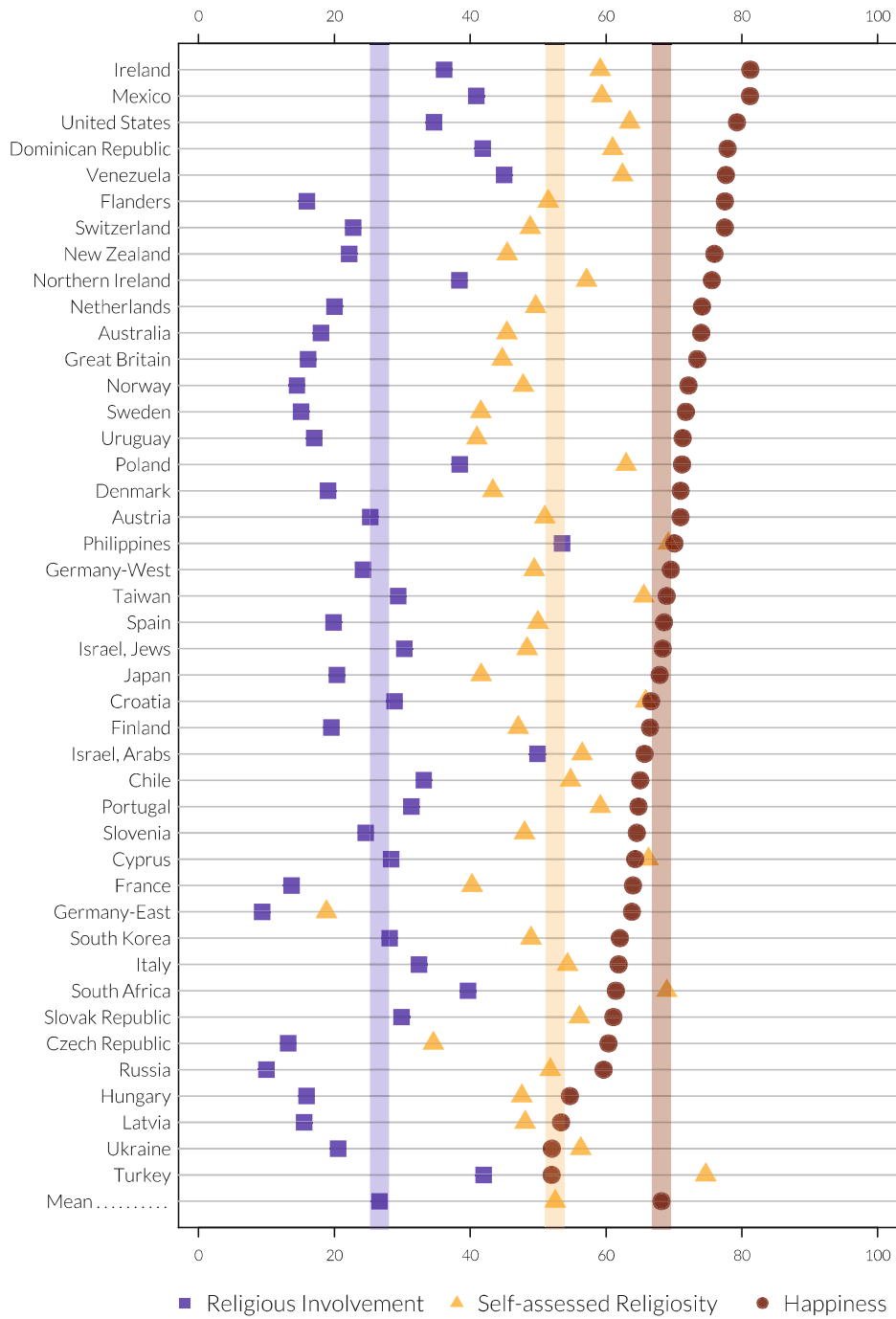
The script is essentially confined to a few lines, since we use the function `profile.plot()` created by Patrick R. Schmid. We have only slightly adjusted the margin settings within the function. After importing the source file and the function, a large part of the script is made up of defining two vectors containing the statement texts. Both, `text.left` and `text.right`, are first initialised with `NULL` and then extended statement by statement, by appending the statements as additional elements to the vector. After import of the data, a list of variables is processed in a loop. For every variable, the `aggregate()` function calculates the mean value for men and women and appends it to result. After specification of a colour for each group, the actual function is then called, data, texts, and colours are handed over as parameters, and a two-column legend is created. At the end come the titles and captions.



### 6.1.11 Dot Chart for Three Variables

#### Religious Involvement, Self-assessed Religiosity, and Happiness

(Mean values scaled from 0 to 100, Data from 2008)



Source: ISSP Data Report Religious Attitudes and Religious Change

About the figure: if there are a lot of attributes in a bar chart and/or you want to plot several variables simultaneously, then a different illustration form is needed. In such cases, the dot chart that was already used as an example in Fig. 2.5 comes in handy. In the variant presented here, three parameters are plotted simultaneously. Data are once more derived from a survey data record, the International Social Survey Programme: Religion III-ISSP 2008. The chart shows three indicators for each country: “Religious Involvement”, “Self-assessed Religiosity”, and “Happiness”. The average value for all countries is shown by a transparent line for each indicator, the individual values as red, blue, and yellow dots. The data record was sorted by the third variable “Happiness”. We can see from the figure that there is no linear correlation between religiosity—either personal or ecclesiastic—and personal happiness. The data are derived from the data record “International Social Survey Programme: Social Inequality IV—ISSP 2009”. The ISSP is a social-economic survey program that has conducted yearly surveys since 1985. These days, institutions from 48 countries take part in its implementation. Apart from social and demographic questions, different main topics that can be repeated every few years are chosen on a yearly basis. In 2009, the main focus was the topic “social inequities”. The data record contains 350 variables and 55,238 units (respondents). The record is available from [gesis.org](http://www.gesis.org) (requires registration) under the title ZA5400 (doi:10.4232/1.11506). It is supplied in different formats; for our example, the record was downloaded in SPSS format. The data record is access class A (data and documents are free for academic research and teaching).

```
pdf_file<-"pdf/dotcharts_overlay.pdf"
cairo_pdf(bg="grey98", pdf_file,width=7,height=10)

library(Hmisc) # because of dotchart2, only there add=T can be \
used
par(omi=c(0.15,0.75,0.95,0.75),mai=c(0.9,1.75,0.25,0),family="\
Lato Light",las=1)

# Import data and prepare chart

myData<-read.xls("myData/Bechert_Graph.xlsx",sheet=1, encoding="\
latin1")
row.names(myData)<- myData$Countries
myData$Countries<-NULL

total<- myData["Mean . . . . . ",]
myData<- myData[rownames(myData)!="Mean . . . . . ",]

myDatasort <- myData [order(-myData$Happiness),]
myDatasort <-rbind(myDatasort,total)
attach(myDatasort)

myC1<-rgb(255,165,0,190,maxColorValue=255)
myC2<-rgb(0,0,139,190,maxColorValue=255)
myC3<-rgb(100,0,0,190,maxColorValue=255)

myC1g<-rgb(255,165,0,60,maxColorValue=255)
```

```

myC2g<-rgb(0,0,139,60,maxColorValue=255)
myC3g<-rgb(100,0,0,60,maxColorValue=255)

# Create chart and other elements

dotchart2(Religiosity., labels=row.names(myDatasort), pch=17, \
  dotsize=4, cex=0.6, cex.labels=0.75, xlab="", col=myC1, xaxis=F, \
  , xlim=c(1,100))
dotchart2(Involvement, labels=row.names(myDatasort), pch=15, \
  dotsize=4, cex=0.6, xlab="", col=myC2, xaxis=F, add=T)
dotchart2(Happiness, labels=row.names(myDatasort), pch=19, dotsize=
  =4, cex=0.6, xlab="", col=myC3, xaxis=F, add=T)

axis(1)
axis(3)

abline(v=total, col=c(myC1g, myC2g, myC3g), lwd=12)
legend(-5, -2.6, c("Religious Involvement"), ncol=1, pch=15, col=myC2, \
  , bty="n", cex=1.5, pt.cex=1.5, xpd=T)
legend(35, -2.6, c("Self-assessed Religiosity"), ncol=1, pch=17, col=\
  myC1, bty="n", cex=1.5, pt.cex=1.5, xpd=T)

legend(80, -2.6, c("Happiness"), ncol=1, pch=19, col=myC3, bty="n", cex=
  =1.5, pt.cex=1.5, xpd=T)

# Titling

mtext("Religious Involvement, Self-assessed Religiosity, and \
  Happiness ", 3, line=3.75, adj=0, cex=1.05, family="Lato Black", \
  , outer=T)
mtext("(Mean values scaled from 0 to 100, Data from 2008)", 3, \
  line=1.25, adj=0, cex=0.90, font=3, outer=T)
mtext("Source: ISSP Data Report Religious Attitudes and \
  Religious Change", 1, line=-1, adj=1, cex=0.90, font=3, outer=T)

dev.off()

```

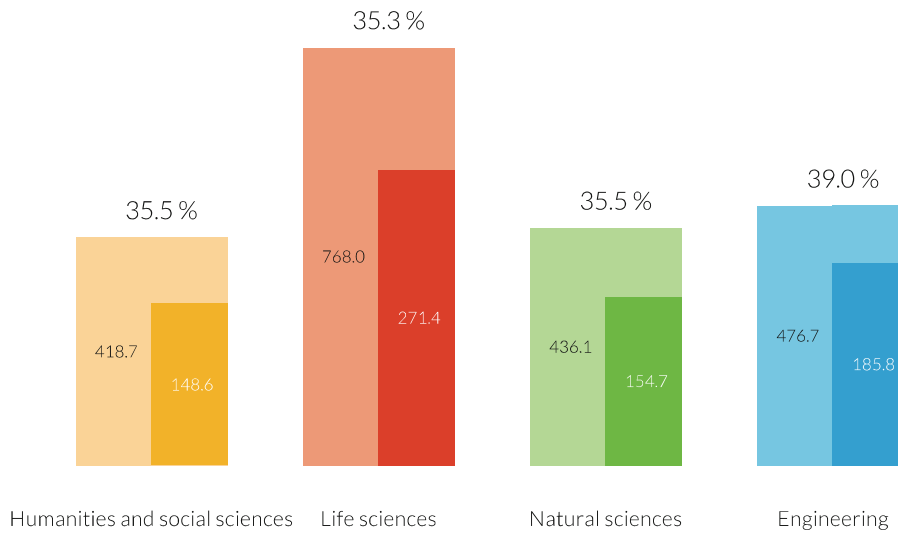
The script first defines an illustration window that is twice as high as wide, and specific margin settings. The Hmisc library by Frank E. Harrell Jr is embedded because we require its dotchart2() function. The advantage this has over the original dotchart() function is that it lets us specify dot size, x-axis layout, and above all the stacking of multiple dot charts. We import data from an XLS sheet, define the content of the “Country” column as row name, and the vector total that contains the row of means. Once this is done, this row is removed from the data, the data record is sorted in descending order by the “Religiosity” variable, and the total row is appended. The attach() function saves us from having to spell out the name of the data record before the variables in the future. This is followed by three definitions of transparent colours for the dots and another three even more transparent ones for the mean lines. The dotchart2() function then plots the dot charts. The function is called three times, with add=TRUE for the second and third call so they are stacked

on top of the existing one. For better orientation, we draw x-axes at both the top and bottom using `axis(1)` and `axis(3)`. This is followed by the three vertical lines specifying the means using `abline()`. “v” means “vertical”; `total` is the vector with the position of the means. We use the value 12, thereby choosing very wide lines, because their width will then approximately match the dots’ diameter. The legend is essentially the same as in Sect. 6.1.7. As before, the script ends with the titles and captions.

### 6.1.12 Column Chart with Shares

#### DFG grants in 2010

*Individual grants by science sector, values in million Euro. Percent value: approval quota*



*Source: DFG Information Cards, [www.dfg.de](http://www.dfg.de)*

About the figure: With bar charts, there is also the option to show shares. However, there is one thing to be mindful of: in the commonly used illustration form of “stacked” areas, a share cannot be distinguished from an addition without further explanation. In these cases, it makes sense not to use the entire width of the bars or columns for the presentation of the share, but only half of them. However, to maintain proportions, the height of this area has to be doubled. This example shows such an illustration form for the sums of requested and granted applications for research grants submitted to the German Research Foundation [Deutsche Forschungsgemeinschaft (DFG)], split into four scientific sectors. Since the granted amounts are real parts of the resources requested—not one Euro that wasn’t requested is granted—the granted sums can be plotted as parts of the sums requested. In the figure, column height reflects the amount requested, and the

lower area the fraction of that amount that was granted. For better orientation, the respective sums are written inside the columns, and the grant rates (granted divided by requested) over the columns. The figure's essential messages are that the grant rates are almost identical across all four science sectors and that significantly more is requested and therefore also granted in the life sciences than in the other three fields. The data can be directly extracted from a PDF publication "DFG Information Cards 2011" (card 9, "Research Grants") on the DFG website and was manually entered into the R script.

```
pdf_file<-"pdf/columncharts_shares_1x4.pdf"
cairo_pdf(bg="grey98", pdf_file,width=11,height=7)

par(cex=0.9,omi=c(0.75,0.5,1.25,0.5),mai=c(0.5,1,0.75,1),mgp=c(
  (3,2,0),family="Lato Light",las=1)

# Import data

source("scripts/inc_data_dfg.r", encoding="latin1")

# Create charts and other elements

barplot(x,col=c(myC1a,myC1a,myC2a,myC2a,myC3a,myC3a,myC4a,myC4a)\
  ,beside=T,border=NA,axes=F,names.arg=c("","","",""))
barplot(2*y,col=c(myC1a,myC1b,myC2a,myC2b,myC3b,myC3b,myC4a,\
  myC4b),beside=T,border=NA,axes=F,add=T,names.arg=labelling\
  ,cex.names=1.25)

z<-1
for (i in 1:4)
{
text(z+0.25,x[1,i]/2,format(round(x[1,i],1),nsmall=1),adj=0)
text(z+1.25,y[2,i],format(round(y[2,i],1),nsmall=1),adj=0,col="\
  white")
text(z+0.65,x[1,i]+50,paste(format(round(100*y[2,i]/x[1,i],1),
  nsmall=1),"%",sep=" "),adj=0,cex=1.5,xpd=T)
z<-z+3
}

# Titling

mtext("DFG grants in 2010",3,line=4,adj=0,family="Lato Black",\
  outer=T,cex=2)
mtext("Individual grants by science sector, values in million \
  Euro. Percent value: approval quota",3,line=1,adj=0,cex\
  =1.35,font=3,outer=T)
mtext("Source: DFG Information Cards, www.dfg.de",1,line=2,adj\
  =1.0,cex=1.1,font=3,outer=T)
dev.off()
```

The script first embeds the data: first four colours, twice, then the actual data, and finally the labels. Note that the application sums are read twice. This makes the bars twice as wide. The grant data on the other hand are only read once; the respective first values are set to zero. This is followed by two `barplot()` calls. Since we do not set `horiz=TRUE` this time, not bar, but column charts are plotted. The second call is done with `add=TRUE`, so that these columns are plotted on top of the first ones. To maintain correct proportions, grant data (y) have to be multiplied by 2. Alternatively, column width could have been adjusted accordingly within the `barplot()` calls. Finally follow four times three labels: halfway up, the values of the requested and granted sums, respectively (requested sums in the left columns), centrally over the columns the ratio of granted to requested sums in percent. The term `format(round(x[1,i],1),nsmall=1),adj=0)` ensures output of a decimal place even with integers. Further examples for bar charts will follow in Sect. 7.3.

## 6.2 Pie Charts and Radial Diagrams

Pie charts are pretty to look at—nothing more. But no less for that, which is why I do not want to discourage their use. The frequently used argument that human perception finds it easier to compare, e.g., points on lines (dot charts) remains unchallenged. However, if the magnitude of individual data is so close together that they are hard to sort by their exact difference in magnitude, then the message of these data is that they are similar and small differences are therefore unimportant. In this section, we will not only explain regular pie charts, but also give examples of spie charts and radial polygons, and in Chap. 11 also radial column charts. The types differ in the variability of their radii and/or angles:

Type	Radius	Angle
Pie chart	Constant	Variable
Spie chart	Variable	Variable
Radial polygon	Variable	Constant
Radial column chart	Variable	Constant