Chapter 7
Distributions

In addition to purely statistical presentation forms, such as histograms and box plots, visualisations of distributions include traditional presentation forms such as population pyramids or Lorenz curves. Obviously, not only populations can be shown in pyramids, and obviously an inequality, which is what a Lorenz curve commonly depicts, can be shown in a different form, too. We will therefore show alternative examples for both.

7.1 Histograms and Box Plots

7.1.1 Histograms Overlay

The figure shows the distribution of the women-to-men ratio in the German federal states of Brandenburg and Rhineland-Palatinate (Rheinland-Pfalz). With histograms, the choice of classification is vital. If we assume a normal or at least symmetrical distribution, then the mean should be plotted in the centre of the figure and the x-axis should span equal distances in both directions. If two distributions are stacked, then the mean of one of the distributions should be taken as reference point. The x-axis length then has to be chosen so it covers both distributions in their entirety. If the figures are stacked, then transparent colouring is most useful, as the overlap is then discernible as a third colour. Axis lines can be omitted. Since we are dealing with stable variables, graduation lines on the x-axis are not necessarily attached at the centre of the classes.
Data: see annex A, v_frauen_maenner.

```r
pdf_file<-'pdf/histograms_overlay.pdf'
cairo_pdf(bg="grey98", pdf_file,width=11,height=7)
source("scripts/inc_datadesign_dbconnect.r")
par(omi=c(0.75,0.2,0.75,0.2),mai=c(0.25,1.25,0.25,0.25),family="\nLato Light",las=1)

# Import data and prepare chart
sql<-'select * from v_women_men"
myDataset<-dbGetQuery(con,sql)
attach(myDataset)
myCol1<='rgb(191,239,255,180,maxColorValue=255)
myCol2<='rgb(255,0,210,80,maxColorValue=255)
brandenburg<-'subset(myDataset,bundesland == 'Brandenburg')

# Create chart
hist(b

```
7.1 Histograms and Box Plots

In the script, data from a MySQL database are integrated by importing the view `v_women_men`. Since we want to stack two histograms, we have to define transparent colours; this will also make the overlap visible. We define two partial datasets: on the one hand all men/women ratios of the German federal state of Brandenburg, on the other hand all of the state Rhineland-Palatinate. A histogram is plotted for both databases using `hist()`, and the second one is added to the first using `add=TRUE`. It is important that we define the same display area on the x-axis for both figures. We do not explicitly specify the number of classes in the histograms; the default setting already provides an optimal result for our data.

7.1.2 Column Charts Coloured with ColorBrewer (Panel)

About the figure: the visualisation of frequency distributions can also be useful for rough classifications. In this example, we look at four variables from the Eurobarometer. The question we are asking ourselves is the distribution of four questions concerning life satisfaction. In 2009, the Eurobarometer 71.2 asked almost 30,000 people a series of questions relating to this topic. The possible responses to the two questions “On the whole, are you very satisfied, fairly satisfied, not very
satisfied or not at all satisfied with the life you lead?” and “How would you judge
the current situation in each of the following?” ranged between 1 (“Very satisfied”
or “very good”) and 4 (“Not at all satisfied” or “very bad”). To the two questions
“What are your expectations for the next twelve months: will the next twelve months
be better, worse or the same, when it comes to…” and “Compared with 5 years ago,
would you say things have improved, gotten worse or stayed about the same when
it comes to …” the possible responses were 1 “better”, 2 “worse” and 3 “same”,
or 1 “improved”, 2 “got worse” and 3 “stayed about the same”. “Don’t know” was
another optional response for all questions. If we are interested in the responses to
those questions in relation to countries, not individual persons, then we first have
to decide on an indicator. One option would be the proportion of people that have
chosen categories 1 and 2. Another would be to assume that the possible responses
correspond to points on a continuous scale and calculate averages. The latter is a
process commonly used for school grades, too. That means the arithmetic mean for
each of the 30 countries and each of the four questions is calculated first, and then
the distribution of those means is considered. One can arrive at useful classifications
by looking at the value distribution in histograms. The simplest way to do this is with

sql<-“select * from v_zza4972_countries”
dataframe<-dbGetQuery(con,sql)
attach(dataframe)
par(mfcol=c(1,4))
for (ii in 2:5) hist(dataframe[,ii],main=names(dataframe[ii]),xlab="")

which gives the following result (Fig. 7.1).

![Fig. 7.1 Histogram of the four variables of the Eurobarometer](image)

Since we want to compare four variables in the present case, we define the same
number of classes in each case. In the intended compact and comparative overview,
labels with the actual values would be confusing. This is why we use a different
form. First, we plot a bar chart instead of a histogram for each case. The difference
lies in the fact that the bars do not border each other. We choose the respective
frequency of the countries as labels for the bars, and for the colours a continuous
Brewer palette per diagram. This will also serve as our legend. Underneath the
individual figures, the range of values is listed. Since we are now able to plot them
one below the other, they are much easier to read. The classifications are the same
for the first two variables, and slightly different for the last two, which also do not measure the same things. We will get back to this figure in Sect. 10.3.4.


```r
par(mfrow=c(2,4),omi=c(0.5,0.5,0.75,0.5),mai=c(0.5,0.5,0.5,0.5),
   cex=1.1,family="Lato Light",las=1)

source("scripts/inc_datadesign_dbconnect.r")
sql<-'select ETX from v_za4972_laender'
myDataset<-dbGetQuery(con,sql)
attach(myDataset)

# Create chart
myCuts<-c(0,1.5,2,2.5,3)
barplot(table(cut(v84,myCuts)),col=brewer.pal(4,"Reds"), ylim=c(0,20), names.arg=table(cut(v84,myCuts)),axes=F,main="Life Satisfaction")
myCuts<-c(0,1.5,2,2.5,3)
barplot(table(cut(v85,myCuts)),col=brewer.pal(4,"Greens"), ylim=c(0,20),names.arg=table(cut(v85,myCuts)),axes=F,main=" Situation: Area you live in")
myCuts<-c(-0.1,0,0.1,0.2,0.3)
barplot(table(cut(v100,myCuts)),col=brewer.pal(4,"Blues"), ylim=c(0,20),names.arg=table(cut(v100,myCuts)),axes=F,main=" Expectations: Area you live in")
myCuts<-c(-0.6,-0.4,0,0.2,0.4)
barplot(table(cut(v114,myCuts)),col=brewer.pal(4,"Purples"), ylim=c(0,20),names.arg=table(cut(v114,myCuts)),axes=F,main=" Change 5 years: Life in general")
```

```r
par(family="Lato", cex=0.7)
plot.new()
bez<-c("0.0 to 1.5","1.5 to 2.0","2.0 to 2.5","2.5 to 3.0")
legend("center", bez,cex=2.05,border=F,bty="n",fill= brewer.pal(4,"Reds"),y.intersp=1.3)
plot.new()
bez<-c("0.0 to 1.5","1.5 to 2.0","2.0 to 2.5","2.5 to 3.0")
legend("center", bez,cex=2.05,border=F,bty="n",fill= brewer.pal(4,"Greens"),y.intersp=1.3)
plot.new()
bez<-c("-0.1 to ±0.0","±0.0 to +0.1","+0.1 to +0.2","+0.2 to +0.3")
legend("center", bez,cex=2.05,border=F,bty="n",fill= brewer.pal(4,"Blues"),y.intersp=1.3)
plot.new()
bez<-c("-0.6 to -0.4","-0.4 to ±0.0","±0.0 to +0.2","+0.2 to +0.4")
legend("center", bez,cex=2.05,border=F,bty="n",fill= brewer.pal(4,"Purples"),y.intersp=1.3)
```
In the script, we first use mfrow=c(2,4) to split the figure into eight areas of equal size, two rows and four columns. The view v_za_4972 already contains the necessary aggregation of the variable means for the individual countries. This is followed by four calls of the barplot() function. To this end, we first define the previously determined classifications within the vector limits. With these classifications and the cut() function, the variables are organised into the respective classes; the class frequencies are simultaneously established using the table() function. The table call is utilised a second time to use the frequencies as axis labels. Four Brewer palettes are employed for the colours. Legends are then plotted into separate windows. Since legend() is a low-level function, we first need one plot.new() call, respectively, to create an empty figure window in which the legend can be written. The structure of the legend corresponds to the previous examples. With y.intersp=1.3, the line spacing is slightly increased to improve readability. Since we are using a panel illustration with multiple figures, we have to write headings and captions into the outer margin using outer=T. Please note that the ‘minus’ symbol is used for the labels, and not the dash shown on the keyboard.

### 7.1.3 Histograms (Panel)

The main question asked in the present and following two figures is what presentation options there are for distributions of a variable that are differentiated by a relatively large number of attributes of a factor or another categorical variable.
About the figure: The figure shows the distribution of monthly net household income of respondents in selected countries. To allow a direct comparison between the countries, purchasing power parity (PPP) weighted values were used. Please note that the distributions are based on the results of a survey. Especially with statements concerning income it should be kept in mind that, without doubt, voluntary disclosures will not give a perfect representation. However, income values are usually a good reference point and are used for the analysis of socioeconomic questions in which an impact of the amount of income is suspected. The figure shows the distribution of the income disclosures for 46 countries. The same y-axis scaling was used for all countries, so frequencies are directly comparable. Similarly, x-axis span and class width are each identical. Data are sorted in descending order of their median, so that countries judged “poorest” by these criteria come first. The
logic of the definition of histograms dictates that the distribution for the countries begins very left-heavy and then continuously extends further to the right and flattens out. Another fact that can be deduced from the histograms is that the data are strongly classed, especially those in the higher income areas.

Data: See annex A, ZA4804: European Values Study Longitudinal Data File 1981–2008 (EVS 1981–2008). Data are survey data from the third and fourth wave of the European Values Study. The fourth wave was conducted between 2008 and 2010, the third wave from 1999 to 2001. In most countries included in the third wave, 1000–1200 persons were questioned, in the fourth wave approximately 1500 persons, sometimes fewer in smaller countries.

---

```r
# Import data and prepare chart

ZA4804<–spss.system.file("mydata/ZA4804_v3-0-0.sav")
myData<–subset(ZA4804,select=c(s002evs,s003,x047d))
attach(myData)
t<–subset(myData,x047d>0 & s002evs=="2008–2010")
tMedians<–aggregate(as.numeric(x047d),list(as.factor(s003)),median,na.rm =T)
tCountries<–tMedians[order(tMedians$x),1]

# Create chart

attach(t)
for (i in 1:(length(tCountries)-2))
{
    Country<–subset(t,s003==tCountries[i])
    hist(Country$x047d,main="",axes=F,xlab="",ylab="",xlab=c(0,8),ylab=c(0,1000),border="white",col="red",breaks=seq(from=-2,to=16,by=0.5))
    text(4,900,tCountries[i],cex=3.0)
    box(lty='dotdash',col='grey')
    if (i==43) axis(1,cex.axis=3,at=c(0,8),labels=c("Less
than €150\,"","€8,000"),mgp=c(0,8,1))
}

# Titling

mtext("European Value Study 2008–2010: Income Distribution of 47 Countries",3,line=10,adj=0,cex=3.8,family="Lato Black",outer=T)
```
In the script, we first define a very large window of 25 by 25 inches, a little larger than the short side of the A1 format. Data are read from an SPSS file using Martin Elff’s memisc package, and limited to the required data. Here,

- s002evs is the EVS wave
- s003 the country
- x047d the purchasing power parity-weighted household income

After data import, we create a list of all featured countries and their respective medians. To do this, we use the aggregate() function. Aggregation functions have to be transferred as a list. The country list is sorted in ascending order by the median. In a loop, we now draw a histogram for each country, always using the same x- and y-axis areas and class widths. Once this is done, the heading is written above the histograms using text(), and a box of “dotdash” type is created. An x-axis label is attached in the lower left chart (the 43rd).

### 7.1.4 Box Plots for Groups: Sorted in Descending Order

Histograms are without doubt a very suitable form to present distributions. With limited space, the box plot is an alternative that can also explicitly plot a series of key quantities of a distribution. Here, the interquartile distance, the span of the first and third quartile, is depicted as a bar (a “box”) in which the median (the second quartile) is added as a marker. In an extended variant, lines are joined on the left and right, so that it becomes a box-and-whisker plot. There are different definitions of these lines: the most frequently used is that their length is defined by the points that are less than 1.5 interquartile distances from the first or third quartile. This definition is also the default setting for R’s boxplot function.
The figure again shows a comparative income distribution in different countries, as in Sect. 7.1.3, but in the compact form of simplified box plots. In contrast to the complete box plot version, only the median and the interquartile distance (as span) are shown. Since we do not want to group anything, a continuous Brewer gamut in steps of ten is used as colour. Looking at the figure, we can derive that income disclosures from countries with a higher monthly net household income sometimes show a larger span than disclosures from countries with a lower income (each PPP.
weighted). However, when interpreting the results, it should be kept in mind that we present the quartile distance, and not the entire income. This means the figure shows that area around the median in which the inner 50% of the characteristic distribution lies. The selected illustration form is similar to the “dot charts” that we got to know in Sect. 6.1.11.


```
pdf_file<="pdf/boxplots_multiple.pdf"
cairo_pdf(bg="grey98", pdf_file,width=7,height=9)
par(omi=c(0.35,0.25,0.75,0.75), mai=c(0.95,1.75,0.25,0),family="\nLato Light", las=1)
library(RColorBrewer)
library(memisc)

# Import data and prepare chart
myDataFile<="myData/ZA4804_v2-0-0.sav"
ZA4804<–spss.system.file(myDataFile)
myData<–subset(ZA4804, select=c(s002evs,s003,x047d))
attach(myData)
x<–subset(myData,x047d>0 & s002evs=="2008–2010")
attach(x)

# Import data and prepare chart
myDataFile<="myData/ZA4804_v2-0-0.sav"
ZA4804<–spss.system.file(myDataFile)
myData<–subset(ZA4804, select=c(s002evs,s003,x047d))
attach(myData)
x<–subset(myData,x047d>0 & s002evs=="2008–2010")
attach(x)

# Import data and prepare chart
myDataFile<="myData/ZA4804_v2-0-0.sav"
ZA4804<–spss.system.file(myDataFile)
myData<–subset(ZA4804, select=c(s002evs,s003,x047d))
attach(myData)
x<–subset(myData,x047d>0 & s002evs=="2008–2010")
attach(x)

# Import data and prepare chart
myDataFile<="myData/ZA4804_v2-0-0.sav"
ZA4804<–spss.system.file(myDataFile)
myData<–subset(ZA4804, select=c(s002evs,s003,x047d))
attach(myData)
x<–subset(myData,x047d>0 & s002evs=="2008–2010")
attach(x)

# Import data and prepare chart
myDataFile<="myData/ZA4804_v2-0-0.sav"
ZA4804<–spss.system.file(myDataFile)
myData<–subset(ZA4804, select=c(s002evs,s003,x047d))
attach(myData)
x<–subset(myData,x047d>0 & s002evs=="2008–2010")
attach(x)

# Import data and prepare chart
myDataFile<="myData/ZA4804_v2-0-0.sav"
ZA4804<–spss.system.file(myDataFile)
myData<–subset(ZA4804, select=c(s002evs,s003,x047d))
attach(myData)
x<–subset(myData,x047d>0 & s002evs=="2008–2010")
attach(x)

# Import data and prepare chart
myDataFile<="myData/ZA4804_v2-0-0.sav"
ZA4804<–spss.system.file(myDataFile)
myData<–subset(ZA4804, select=c(s002evs,s003,x047d))
attach(myData)
x<–subset(myData,x047d>0 & s002evs=="2008–2010")
attach(x)

# Import data and prepare chart
myDataFile<="myData/ZA4804_v2-0-0.sav"
ZA4804<–spss.system.file(myDataFile)
myData<–subset(ZA4804, select=c(s002evs,s003,x047d))
attach(myData)
x<–subset(myData,x047d>0 & s002evs=="2008–2010")
attach(x)

# Import data and prepare chart
myDataFile<="myData/ZA4804_v2-0-0.sav"
ZA4804<–spss.system.file(myDataFile)
myData<–subset(ZA4804, select=c(s002evs,s003,x047d))
attach(myData)
x<–subset(myData,x047d>0 & s002evs=="2008–2010")
attach(x)

# Import data and prepare chart
myDataFile<="myData/ZA4804_v2-0-0.sav"
ZA4804<–spss.system.file(myDataFile)
myData<–subset(ZA4804, select=c(s002evs,s003,x047d))
attach(myData)
x<–subset(myData,x047d>0 & s002evs=="2008–2010")
attach(x)

# Import data and prepare chart
myDataFile<="myData/ZA4804_v2-0-0.sav"
ZA4804<–spss.system.file(myDataFile)
myData<–subset(ZA4804, select=c(s002evs,s003,x047d))
attach(myData)
x<–subset(myData,x047d>0 & s002evs=="2008–2010")
attach(x)

# Import data and prepare chart
myDataFile<="myData/ZA4804_v2-0-0.sav"
ZA4804<–spss.system.file(myDataFile)
myData<–subset(ZA4804, select=c(s002evs,s003,x047d))
attach(myData)
x<–subset(myData,x047d>0 & s002evs=="2008–2010")
attach(x)
```

```
# Create chart and other elements
boxplot(1000*x047d ~ s003f, horizontal=T, ylim=c(0,4000), border=NA, boxwex=0.25, las=1, col=myColour, outline=F, cex.axis=0.7)
points(sort(1000*tM$x, decreasing=T), length(unique(s003)):1, pch=19, cex=1.15, col=rev(myColour))
abline(v=2000)
abline(h=seq(7.5,37.5,by=10))
par(fg="black")
mtex("25 \%", 3, at=1300, line=-2)
mtex("75 \%", 3, at=3000, line=-2)
```
In the script, we compile a list of all featured countries and their respective medians as in the previous example. We want to aggregate by variable s003, with representation by the medians of x047d, sorted in descending order. To this end, we create the variable s003f as a factor with the country names from the median list and organised by levels (categories). To achieve the descending order by the median, we use a procedure that was suggested by Jim Porzak in an R mailing list.

That done, the box plots can be drawn. R expects a formula notation for this. Since the income is saved in thousands in the data frame, we multiply it by 1000 for a correct x-axis label. Using the boxwex parameter, the box plots are drawn narrower than usual. outline=F ensures that no rogue values are plotted. This means the representation of the box plot is limited to the span of the two quartiles. We choose a Brewer palette with six values as colour, and we use the second to sixth (the first is too light for our purposes). This means each of ten countries, at the end 7, are coloured. For ease of orientation, we also plot a vertical auxiliary line in the middle, and horizontal guide lines every ten rows.

The necessary labels come at the end of the script.
7.1.5 *Box Plots for Groups: Sorted in Descending Order, Comparison of Two Polls*

**Income Distribution 1999 and 2008**

*European Values Study*

About the figure: in contrast to the previous example, an additional temporal comparison is included here: How has the income distribution in the individual countries changed when compared with an earlier point in time? The survey wave from 1999 to 2001 was used for comparison. It is important to design the figure in such a way that allows a comparison not only of countries but also times. Here, quartile spans were represented such that the distance between the two lines for one country is smaller than the distance between lines for different countries. Additionally, the country lines were highlighted with a background colour for better visual differentiation between the countries. The median mark—a triangle—was chosen to take up little space, but be clearly visible. The structure is explained in a legend at the top right.

The figure compares the monthly net household income of the respondents, weighted with purchasing power parities, for the 1999 and 2008 waves. Be aware though that purchasing power parities are designed for a specific year and temporal changes for a certain country should only be interpreted with caution.


Data preparation and chart creation:

```R
pdf_file<-'pdf/boxplots_multiple_compare.pdf'
cairo_pdf(bg='grey98', pdf_file, width=7, height=10)

par(omi=c(0.35,0.25,0.75,0.25), mai=c(0.75,1.75,0.55,0), family="Lato Light", las=1)
library(memisc)

# Import data and prepare chart
myDatafile<-'myData/ZA4804_v2-0-0.sav'
ZA4804<-spss.system.file(myDatafile)
myData<-subset(ZA4804, select=c(s002evs,s003,x047d))
attach(myData)
t1<-subset(myData,x047d>0 & s002evs=="1999–2001")
t1_countries<-unique(t1$s003)
t2<-subset(myData,x047d>0 & s002evs=="2008–2010" & is.element(\s003,t1_countries))

attach(t1)
a1<-aggregate(as.numeric(x047d),list(as.factor(s003)), quantile, na.rm =T)

attach(t2)
a2<-aggregate(as.numeric(x047d),list(as.factor(s003)), quantile, na.rm =T)
a1.sorted<-a1[order(a1$x[,3]),]

# Define chart
plot(1:1, type="n", xlim=c(0,4.25), ylim=c(0.5,51.5), axes=F,xlab="",ylab="",yaxs="i")
```

# Other elements

abline(v=c(0,1,2,3,4), lty="dotted", col="grey70")

myC1<-"gray55"
myC2<-"deeppink"
myBckgrnd<-rgb(191,239,255,70,maxColorValue=255)

for (i in 1:25)
  {
    rect(0,2*i-0.9,4.25,2*i+0.9, col=myBckgrnd, border=NA)
    segments(a1.sorted$x[i,2],2*i-0.2,a1.sorted$x[i,4],2*i-0.2, lwd=4, col=myC1)
    segments(a2$x[a2$Group.1==a1.sorted$Group.1[i],2],2*i+0.2, a2$x[a2$Group.1==a1.sorted$Group.1[i],4],2*i+0.2, col=myC2, lwd=4)
  }

mtext(c(0, "1.000", "2.000", "3.000", "4.000"),1,at=c(0:4),cex=0.85)

mtext("Monthly Household Income (PPP adjusted) in Euro",1,adj=0.5,line=1.5)

# Titling

mtext("Income Distribution 1999 and 2008",3,line=1.6,adj=0,cex=1.8,family="Lato Black",outer=T)
mtext("European Values Study",3,line=-0.2,adj=0,cex=1.5,font=3,outer=T)
mtext("Source: ZA4804 European Values Study Longitudinal Data File 1981–2008, www.gesis.org",1,line=0,adj=1.0,cex=0.95,font=3,outer=T)

# Legend

par(new=T, omi=c(0,0,0,0), mai=c(8.5,5.5,0.5,0.55))

plot(0:1, xlab="",ylab="")

segments(0,0.42,1,0.42,col=myC1,xpd=T,lwd=4)
segments(0,0.57,1,0.58,col=myC2,xpd=T,lwd=4)

text(0,0.75,"25",adj=0.5,cex=0.7,xpd=T,font=3)
text(1,0.75,"75",adj=0.5,cex=0.7,xpd=T,font=3)
text(0.5,0.75,"Median",adj=0.5,cex=0.7,font=3)
text(-0.1,0.42,"1999–2001",adj=1,cex=0.65,xpd=T,font=3)
text(-0.1,0.58,"2008–2010",adj=1,cex=0.65,xpd=T,font=3)

par(family="Symbola")
The script begins with the loading of the data as in the previous example. Here, we create two partial data frames. The first, t1, comprises the survey data from 1999 to 2001, the second, t2, those from 2008 to 2010. With is.element(s003, t1_countries), only values from countries that also feature in the first data frame are chosen for the second. The boxplot() function used in the previous example does not provide a parameter panel.first that allows for plotting elements of the figure, such as a coloured area or a grid, first. For this reason, we use a different approach in this example, drawing the box plots with the low-level functions segments() and points().

We first need aggregated data. This is easiest to achieve with the aggregate() function and the aggregation function quantile. The result of the first aggregation looks like this (the first five rows):

```
> a1[1:5,]
Group.1     x.0%     x.25%     x.50%     x.75%     x.100%
1  Austria  0.3726296  1.2669406  1.8631480  2.5338812  5.5149180
2   Belgium  0.6957128  0.9486993  1.3914256  2.2768783  3.2888241
3  Bulgaria  0.1688909  0.1688909  0.3377817  0.5477541  1.0133452
4   Croatia  0.3280854  0.7217878  1.1154903  1.9028952  3.4777050
5 Czech Republic  0.5065086  0.6014790  0.8389049  1.0605024  2.0102061
```

The output of the function is therefore a variable group.1 and a variable x that contains the minimum, the first quartile, the median, the second quartile, and the maximum in five columns, respectively. We create such an aggregated data frame for both survey periods, calling these a1 and a2. In the next step, we use a1 to create a sorted data frame a1.sorted, sorting by the third column of x (the median). We now define a common plot() that extends to the range of values of all quartiles in its x-axis. In the y-direction, we need twice the number of featured countries (25) and a “margin allowance” at the top and bottom, so that the y-axis range spans 0.5–51.5. To keep R from adding a standard 4%, yaxs=“i” has to be set. This is followed by five vertical dotted guide lines and the specification of colours for the box plots and the background. Plotting of the actual data is done within a loop. First, a light-blue rectangle is laid behind each row. On this, the quartile spans are plotted using segments(). The first segment is processed row by row and contains the range of the value from the second column to the value of the fourth column of x within the data frame a1.sorted. The y-position is the numerator, shifted down by 0.2, respectively. The second segment has an almost identical structure. The difference is that the second data frame a2 is not processed row by row. Instead, the row, whose country corresponds to the current country of a1.sorted is picked out. Additionally, the y-position is shifted up by 0.2. Now we are only missing a marker for the median. Triangles with the tip pointing up and down are a good option. The plot symbol of filled triangles is available in R, but only with the tip pointing up. This is why we use the text() function for the image, and the Unicode characters “BLACK UP-POINTING TRIANGLE” and “BLACK DOWN-POINTING TRIANGLE” from the Unicode block “Geometric Shapes”. Since the Lato font does not contain these characters, we choose the Symbola font. We shift the position of the triangles up or
down by 0.4, so they protrude from the lines. Finally, we change the font back to Lato Light and then, in the last command within the loop, set the respective country’s name left of the figure. After the loop come axes labels, title, and source details. At the end, we have to plot a legend at the top right; this is a bit more complex than before. We use an individual plot() call after calling par() with new=T and setting the margins so that the figure is plotted in the top left corner.

7.2 (Population) Pyramids

Population pyramids are a special form of distribution. They are very descriptive figures that are frequently found in publications, but are apparently not part of the standard repertoire of the many graphic software packages or spreadsheet analyses. The Web does offer a series of “workarounds”, but with almost all of these, the misappropriation of standard figures is obvious. What constitutes a population pyramid? Generally, such figures illustrate the age distribution of a population differentiated by gender as bars arranged back-to-back. The expression “pyramid” is a relic from times in which populations presented in this way did actually resemble a pyramid: many young and few old people. However, this is no longer the case in advanced industrial nations. Here, the younger generations are much less represented than the older ones. Therefore, the distribution of modern populations resembles a mushroom rather than a pyramid. However, the term “population pyramid” is still valid today. With the help of such a population pyramid, a population’s structure can be presented very clearly. With the appropriate illustration, its general structure, war-related losses of individual years, surplus of women or men, or portions of certain population groups are evident at a glance. The following examples show different aspects that can be described with population pyramids. The first example (Sect. 7.2.1) shows a variant in which different age classes are colour-coded; Sect. 7.2.2 shows two pyramids for comparison, with the surpluses of women and men mirrored on the opposing sides. In contrast, in Sect. 7.2.3, proportions of the population are suitably highlighted. The last two examples show the use of the pyramid for summarised data and for an estimation scale—since they can obviously also be useful for the representation of data other than age classes. In R, population pyramids can be created with individual packages, for example with the pyramid package by Minato Nakazawa. More flexible options are offered by the pyramid.plot() function from the plotrix package by Jim Lemon. The pyramids introduced here use the barplot() function, which—when extended with a few low-level functions—yields appealing results.
Data Visualisation with R
100 Examples
Rahlf, Th.
2017, XV, 385 p. 181 illus., 162 illus. in color.,
Hardcover
ISBN: 978-3-319-49750-1