Chapter 6
Step 4: Exploring Data

6.1 A First Glimpse at the Data

After data collection, exploratory data analysis cleans and – if necessary – pre-processes the data. This exploration stage also offers guidance on the most suitable algorithm for extracting meaningful market segments.

At a more technical level, data exploration helps to (1) identify the measurement levels of the variables; (2) investigate the univariate distributions of each of the variables; and (3) assess dependency structures between variables. In addition, data may need to be pre-processed and prepared so it can be used as input for different segmentation algorithms. Results from the data exploration stage provide insights into the suitability of different segmentation methods for extracting market segments.

To illustrate data exploration using real data, we use a travel motives data set. This data set contains 20 travel motives reported by 1000 Australian residents in relation to their last vacation. One example of such a travel motive is: I AM INTERESTED IN THE LIFE STYLE OF LOCAL PEOPLE. Detailed information about the data is provided in Appendix C.4. A comma-separated values (CSV) file of the data is contained in the R package MSA and can be copied to the current working directory using the command

```
R> vaccsv <- system.file("csv/vacation.csv",
+   package = "MSA")
R> file.copy(vaccsv, ".")
```

Alternatively, the CSV file can be downloaded from the web page of the book (http://www.MarketSegmentationAnalysis.org). The CSV file can be explored with a spreadsheet program before commencing analyses in R.

S. Dolnicar et al., *Market Segmentation Analysis*, Management for Professionals,
https://doi.org/10.1007/978-981-10-8818-6_6

To read the data set into R, we use the following command:

```
R> vac <- read.csv("vacation.csv", check.names = FALSE)
```

`check.names = FALSE` prevents `read.csv()` to convert blanks in column names to dots (which is the default). After reading the data set into R, we store it in a data frame named `vac`.

We can inspect the the `vac` object, and learn about column names, and the size of the data set using the commands:

```
R> colnames(vac)
```

```
 [1] "Gender"
 [2] "Age"
 [3] "Education"
 [4] "Income"
 [5] "Income2"
 [6] "Occupation"
 [7] "State"
 [8] "Relationship.Status"
 [9] "Obligation"
[10] "Obligation2"
[11] "NEP"
[12] "Vacation.Behaviour"
[13] "rest and relax"
[14] "luxury / be spoilt"
[15] "do sports"
[16] "excitement, a challenge"
[17] "not exceed planned budget"
[18] "realise creativity"
[19] "fun and entertainment"
[20] "good company"
[21] "health and beauty"
[22] "free-and-easy-going"
[23] "entertainment facilities"
[24] "not care about prices"
[25] "life style of the local people"
[26] "intense experience of nature"
[27] "cosiness/familiar atmosphere"
[28] "maintain unspoilt surroundings"
[29] "everything organised"
[30] "unspoilt nature/natural landscape"
[31] "cultural offers"
[32] "change of surroundings"
```

```
R> dim(vac)
```

```
[1] 1000   32
```

`summary(vac)` generates a full summary of the data set. Below we select only four columns to show `Gender` (column 1 of the data set), `Age` (column 2), `Income` (column 4), and `Income2` (column 5).

```
R> summary(vac[, c(1, 2, 4, 5)])
   Gender         Age                                 Income
 Female:488   Min.   : 18.00   $30,001 to $60,000   :265
 Male  :512   1st Qu.: 32.00   $60,001 to $90,000   :233
              Median : 42.00   Less than $30,000    :150
              Mean   : 44.17   $90,001 to $120,000  :146
              3rd Qu.: 57.00   $120,001 to $150,000: 72
              Max.   :105.00   (Other)             : 68
                               NA's                : 66
   Income2
 <30k   :150
 >120k  :140
 30-60k :265
 60-90k :233
 90-120k:146
 NA's   : 66
```

As can be seen from this summary, the Australian travel motives data set contains answers from 488 women and 512 men. The age of the respondents is a metric variable summarised by the minimum value (Min.), the first quartile (1st Qu.), the median, the mean, the third quartile (3rd Qu.), and the maximum (Max.). The youngest respondent is 18, and the oldest 105 years old. Half of the respondents are between 32 and 57 years old. The summary also indicates that the Australian travel motives data set contains two income variables: `Income2` consists of fewer categories than `Income`. `Income2` represents a transformation of `Income` where high income categories (which occur less frequently) have been merged. The summary of the variables `Income` and `Income2` indicates that these variables contain missing data. This means that not all respondents provided information about their income in the survey. Missing values are coded as `NA`s in R. `NA` stands for "not available". The summary shows that 66 respondents did not provide income information.

6.2 Data Cleaning

The first step before commencing data analysis is to clean the data. This includes checking if all values have been recorded correctly, and if consistent labels for the levels of categorical variables have been used. For many metric variables, the range of plausible values is known in advance. For example, age (in years) can be expected to lie between 0 and 110. It is easy to check whether any implausible values are contained in the data, which might point to errors during data collection or data entry.

Similarly, levels of categorical variables can be checked to ensure they contain only permissible values. For example, gender typically has two values in surveys: female and male. Unless the questionnaire did offer a third option, only those two should appear in the data. Any other values are not permissible, and need to be corrected as part of the data cleaning procedure.

Returning to the Australian travel motives data set, the summary for the variables Gender and Age indicates that no data cleaning is required for these variables. The summary of the variable Income2 reveals that the categories are not sorted in order. This is a consequence of how data is read into R. R functions like read.csv() or read.table() convert columns containing information other than numbers into factors. Factors are the default format for storing categorical variables in R. The possible categories of these variables are called levels. By default, levels of factors are sorted alphabetically. This explains the counter-intuitive ordering of the income variable in the Australian travel motives data set. The categories can be re-ordered. One way to achieve this is to copy the column to a helper variable inc2, store its levels in lev, find the correct re-ordering of the levels, and then convert the variable into an ordinal variable (an ordered factor in R):

```
R> inc2 <- vac$Income2
R> levels(inc2)

[1] "<30k"    ">120k"   "30-60k"  "60-90k"  "90-120k"

R> lev <- levels(inc2)
R> lev

[1] "<30k"    ">120k"   "30-60k"  "60-90k"  "90-120k"

R> lev[c(1, 3, 4, 5, 2)]

[1] "<30k"    "30-60k"  "60-90k"  "90-120k" ">120k"

R> inc2 <- factor(inc2, levels = lev[c(1, 3, 4, 5, 2)],
+   ordered = TRUE)
```

Before overwriting the – oddly ordered – column of the original data set, we double-check that the transformation was implemented correctly. An easy way to do this is to cross-tabulate the original column with the new, re-ordered version:

```
R> table(orig = vac$Income2, new = inc2)

         new
orig      <30k 30-60k 60-90k 90-120k >120k
  <30k     150      0      0       0     0
  >120k      0      0      0       0   140
  30-60k     0    265      0       0     0
  60-90k     0      0    233       0     0
  90-120k    0      0      0     146     0
```

As can be seen, all row values in this cross-tabulation have exactly one corresponding column value, and the names coincide. It can be concluded that no errors were introduced during re-ordering, and the original column of the data set can safely be overwritten:

```
R> vac$Income2 <- inc2
```

We can re-order variable `Income` in the same way. We keep all R code relating to data transformations to ensure that every step of data cleaning, exploration, and analysis can be reproduced in future. Reproducibility is important from a documentation point of view, and enables other data analysts to replicate the analysis. In addition, it enables the use of the exact same procedure when new data is added on a continuous basis or in regular intervals, as is the case when we monitor segmentation solutions on an ongoing basis (see Step 10). Cleaning data using code (as opposed to clicking in a spreadsheet), requires time and discipline, but makes all steps fully documented and reproducible. After cleaning the data set, we save the corresponding data frame using function `save()`. We can easily re-load this data frame in future R work sessions using function `load()`.

6.3 Descriptive Analysis

Being familiar with the data avoids misinterpretation of results from complex analyses. Descriptive numeric and graphic representations provide insights into the data. Statistical software packages offer a wide variety of tools for descriptive analysis. In R, we obtain a numeric summary of the data with command `summary()`. This command returns the range, the quartiles, and the mean for numeric variables. For categorical variables, the command returns frequency counts. The command also returns the number of missing values for each variable.

Helpful graphical methods for numeric data are histograms, boxplots and scatter plots. Bar plots of frequency counts are useful for the visualisation of categorical variables. Mosaic plots illustrate the association of multiple categorical variables. We explain mosaic plots in Step 7 where we use them to compare market segments.

Histograms visualise the distribution of numeric variables. They show how often observations within a certain value range occur. Histograms reveal if the distribution of a variable is unimodal and symmetric or skewed. To obtain a histogram, we first need to create categories of values. We call this binning. The bins must cover the entire range of observations, and must be adjacent to one another. Usually, they are of equal length. Once we have created the bins, we plot how many of the observations fall into each bin using one bar for each bin. We plot the bin range on the x-axis, and the frequency of observations in each bin on the y-axis.

A number of R packages can construct histograms. We use package lattice (Sarkar 2008) because it enables us to create histograms by segments in Step 7. We can construct a histogram for variable AGE using:

```
R> library("lattice")
R> histogram(~ Age, data = vac)
```

The left plot in Fig. 6.1 shows the resulting histogram.

By default, this command automatically creates bins. We can gain a deeper understanding of the data by inspecting histograms for different bin widths by specifying the number of bins using the argument `breaks`:

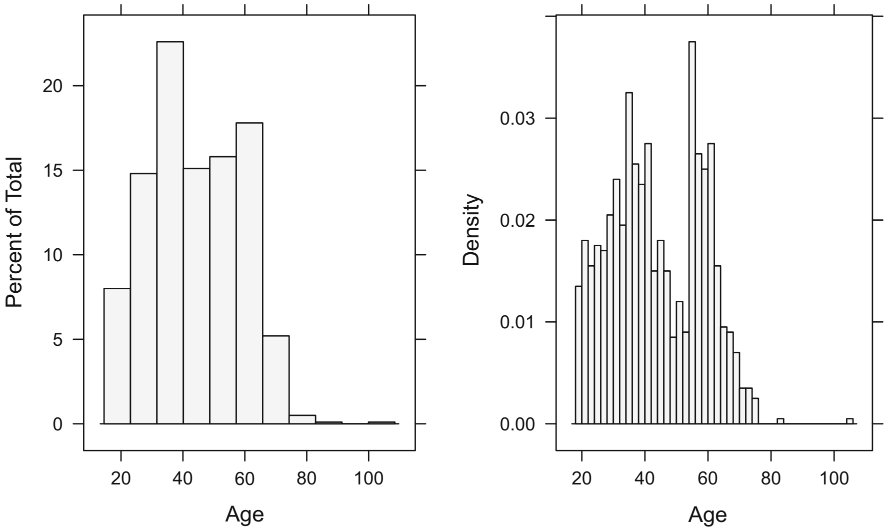

Fig. 6.1 Histograms of tourists' age in the Australian travel motives data set

```
R> histogram(~ Age, data = vac, breaks = 50,
+   type = "density")
```

This command leads to finer bins, as shown in the right plot of Fig. 6.1. The finer bins are more informative, revealing that the distribution is bi-modal with many respondents aged around 35–40 and around 60 years.

Argument `type = "density"` rescales the y-axis to display density estimates. The sum of the areas of all bars in this plot ads up to 1. Plotting density estimates allows us to superimpose probability density functions of parametric distributions. This scaling is in general viewed as the default representation for a histogram.

We can avoid selecting bin widths by using the *box-and-whisker* plot or boxplot (Tukey 1977). The boxplot is the most common graphical visualisation of unimodal distributions in statistics. It is widely used in the natural sciences, but does not enjoy the same popularity in business, and the social sciences more generally. The simplest version of a boxplot compresses a data set into minimum, first quartile, median, third quartile and maximum. These five numbers are referred to as the *five number summary*. R uses the five number summary, and the mean by default to create a numeric summary of a metric variable:

```
R> summary(vac$Age)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  18.00   32.00   42.00   44.17   57.00  105.00 
```

As can be seen from the output generated by this command, the youngest survey participant in the Australian travel motives study is 18 years old. One quarter of respondents are younger than 32; half of the respondents are younger than 42; and

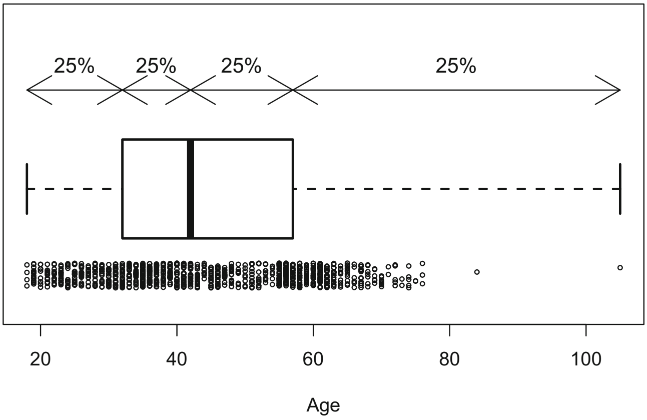

Fig. 6.2 Construction principles for box-and-whisker plots (tourists' age distribution)

three quarters of respondents are younger than 57. The oldest survey respondent is either an astonishing 105 years old, or has made a mistake when completing the survey. The minimum, first quartile, median, third quartile, and maximum are used to generate the boxplot. An illustration of how this is done is provided in Fig. 6.2.

The box-and-whisker plot itself is shown in the middle row of Fig. 6.2. The bottom row plots actual respondent values. Each respondent is represented by a small circle. The circles are jittered randomly in y-axis direction to avoid overplotting in regions of high density. The top row shows the quartiles. The inner box of the box-and-whisker plot extends from the first quartile at 32 to the third quartile at 57. The median is at 42 and depicted by a thick line in the middle of the box. The inner box contains half of the respondents. The whiskers mark the smallest and largest values observed among the respondents, respectively.

Such a simple box-and-whisker plot provides insight into several distributional properties of the sample assuming unimodality. For the Australian travel motives data set, the boxplot shows that the data is right skewed with respect to age because the median is not in the middle of the box but located more to the left. A symmetric distribution would have the median located in the middle of the inner box.

As can also be seen from Fig. 6.2, the 105-year old respondent is solely responsible for the whisker reaching all the way to a value of 105. This, obviously is not an optimal representation of the data, given most other respondents are 70 or younger. The 105-year old respondent is clearly an outlier. The version of the box-and-whisker plot used in Fig. 6.2 is heavily outlier-dependent. To get rid of this dependency on outliers, most statistical packages do not draw whiskers all the way to the minimum and maximum values contained in the data. Rather, they impose a restriction on the length of the whiskers. In R, whiskers are, by default, no longer than 1.5 times the size of the box. This length corresponds approximately to a 99% confidence interval for the normal distribution. Values outside of this range appear as circles. Depicting outliers as circles ensures that information about outliers in the data does not get lost in the box-and-whisker plot.

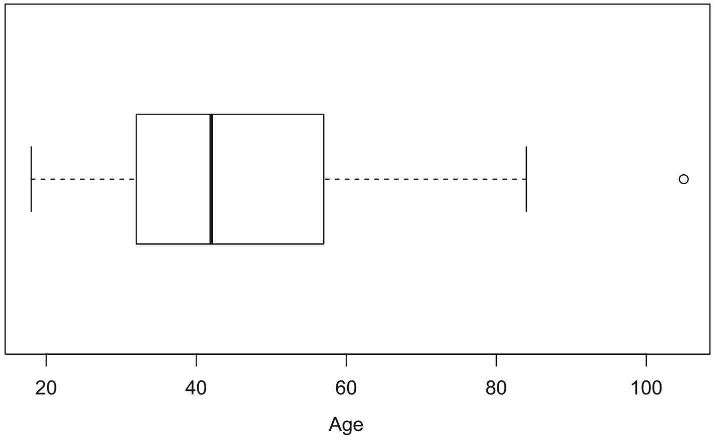

Fig. 6.3 Box-and-whisker plot of tourists' age in the Australian travel motives data set

The standard box-and-whisker plot for variable AGE in R results from:

```
R> boxplot(vac$Age, horizontal = TRUE, xlab = "Age")
```

`horizontal = TRUE` indicates that the box is horizontally aligned, otherwise it would be rotated by 90°. The result is shown in Fig. 6.3.

A comprehensive discussion of graphical methods for numeric data can be found in Putler and Krider (2012) and Chapman and Feit (2015).

To further illustrate the value of graphical methods, we visualise the percentages of agreement with the travel motives contained in the last 20 columns of the Australian travel motives data set. The numeric summaries introduced earlier offer some insights into the data, but they fail to provide an overview of the structure of the data that is intuitively easy and quick to understand. Using R, a graphical representation of this data can be generated with only two commands. Columns 13 to 32 of the data set contain the travel motives, and `"yes"` means that the motive does apply. Searching for string `"yes"` returns `TRUE` or `FALSE` (for `"no"`), function `colMeans()` computes the mean number of `TRUE`s (that is, `"yes"`) for each column as a fraction between 0 and 1. Multiplying by 100 gives a percentage value between 0 and 100. The mean percentages are sorted, and a dot chart with a customised x-axis (argument `xlab` for the label and `xlim` for the range) is created:

```
R> yes <- 100 * colMeans(vac[, 13:32] == "yes")
R> dotchart(sort(yes), xlab = "Percent 'yes'",
+   xlim = c(0, 100))
```

The resulting chart in Fig. 6.4 shows – for the travel motives contained in the data set – the percentage of respondents indicating that each of the travel motives was important to them on the last vacation.

One look at this dot chart illustrates the wide range of agreement levels with the travel motives. The vast majority of tourists want to rest and relax, but realising one's

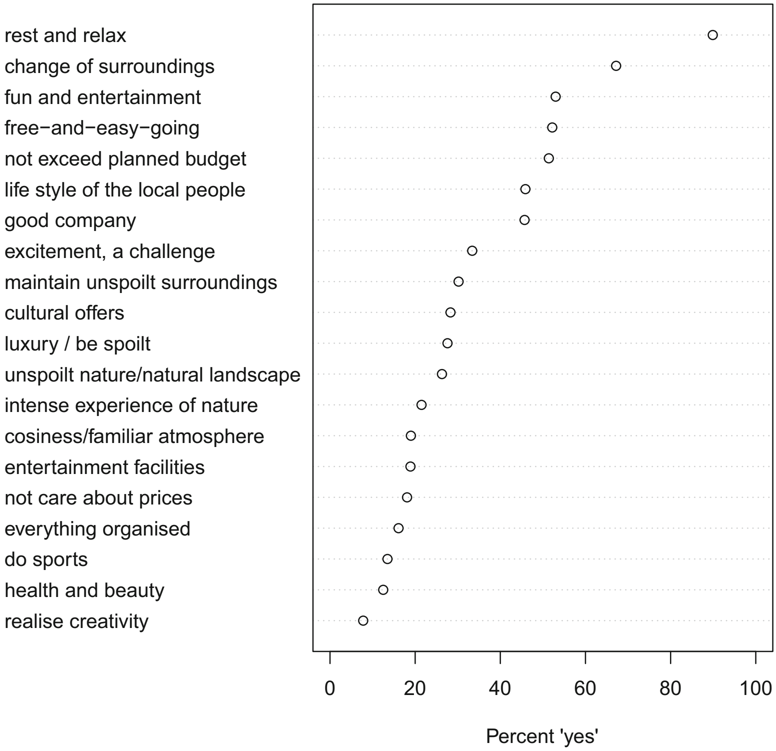

Fig. 6.4 Dot chart of percentages of YES answers in the Australian travel motives data set

creativity is important to only a very small proportion of respondents. The graphical inspection of the data also confirms the suitability of the Australian travel motives variables as segmentation variables because of the heterogeneity in the importance attributed to different motives. In other words: not all respondents say either YES or NO to most of those travel motives; differences exist between people. Such differences between people stand at the centre of market segmentation analysis.

6.4 Pre-Processing

6.4.1 Categorical Variables

Two pre-processing procedures are often used for categorical variables. One is merging levels of categorical variables before further analysis, the other one is converting categorical variables to numeric ones, if it makes sense to do so.

Merging levels of categorical variables is useful if the original categories are too differentiated (too many). Thinking back to the income variables, for example, the original income variable as used in the survey has the following categories:

```
R> sort(table(vac$Income))

$210,001 to $240,000   more than $240,001
                  10                   11
$180,001 to $210,000 $150,001 to $180,000
                  15                   32
$120,001 to $150,000  $90,001 to $120,000
                  72                  146
   Less than $30,000   $60,001 to $90,000
                 150                  233
  $30,001 to $60,000
                 265
```

The categories are sorted by the number of respondents. Only 68 people had an income higher than $150,000. The three top income categories contain only between 10 and 15 people each, which corresponds to only 1% to 1.5% of the observations in the data set with 1000 respondents. Merging all these categories with the next income category (72 people with an income between $120,001 and $150,000), results in the new variable `Income2`, which has much more balanced frequencies:

```
R> table(vac$Income2)

   <30k  30-60k  60-90k 90-120k   >120k
    150     265     233     146     140
```

Many methods of data analysis make assumptions about the measurement level or scale of variables. The distance-based clustering methods presented in Step 5 assume that data are numeric, and measured on comparable scales. Sometimes it is possible to transform categorical variables into numeric variables.

Ordinal data can be converted to numeric data if it can be assumed that distances between adjacent scale points on the ordinal scale are approximately equal. This is a reasonable assumption for income, where the underlying metric construct is classified into categories covering ranges of equal length.

Another ordinal scale or multi-category scale frequently used in consumer surveys is the popular agreement scale which is often – but not always correctly – referred to as Likert scale (Likert 1932). Typically items measured on such a multi-category scale are bipolar and offer respondents five or seven answer options. The verbal labelling is usually worded as follows: STRONGLY DISAGREE, DISAGREE, NEITHER AGREE NOR DISAGREE, AGREE, STRONGLY AGREE. The assumption is frequently made that the distances between these answer options are the same. If this can be convincingly argued, such data can be treated as numerical. Note, however, that there is ample evidence that this may not be the case due to response styles at both the individual and cross-cultural level (Paulhus 1991; Marin et al. 1992; Hui and Triandis 1989; Baumgartner and Steenkamp 2001; Dolnicar and Grün 2007). It is therefore important to consider the consequences of the chosen survey response options before collecting data in Step 3. Unless there is a strong argument for using multi-category scales (with uncertain distances between scale points), it may be preferable to use binary answer options.

Binary answer options are less prone to capturing response styles, and do not require data pre-processing. Pre-processing inevitably alters the data in some way. Binary variables can always be converted to numeric variables, and most statistical procedures work correctly after conversion if there are only two categories. Converting dichotomous ordinal or nominal variables to binary 0/1 variables is not a problem. For example, to use the travel motives as segmentation variables, they can be converted to a numeric matrix with 0 and 1 for NO and YES:

```
R> vacmot <- (vac[, 13:32] == "yes") + 0
```

Adding 0 to the logical matrix resulting from comparing the entries in the data frame to string `"yes"` converts the logical matrix to a numeric matrix with `0` for `FALSE` and `1` for `TRUE`. We will use matrix `vacmot` several times in the book. R package **flexclust** (Leisch 2006) contains it as a sample data set. We can load the data into R using `data("vacmot", package = "flexclust")`. This does not only load the data matrix containing the travel motives `vacmot`, but also the data frame `vacmotdesc` containing socio-demographic descriptor variables.

6.4.2 Numeric Variables

The range of values of a segmentation variable affects its relative influence in distance-based methods of segment extraction. If, for example, one of the segmentation variables is binary (with values 0 or 1 indicating whether or not a tourist likes to dine out during their vacation), and a second variable indicates the expenditure in dollars per person per day (and ranges from zero to $1000), a difference in spend per person per day of one dollar is weighted equally as the difference between liking to dine out or not. To balance the influence of segmentation variables on segmentation results, variables can be standardised. Standardising variables means transforming them in a way that puts them on a common scale.

The default standardisation method in statistics subtracts the empirical mean $\bar{x}$ and divides by the empirical standard deviation s:

$$z_i = \frac{x_i - \bar{x}}{s},$$

with

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2,$$

for the n observations of a variable $\boldsymbol{x} = \{x_1, \ldots, x_n\}$. This implies that the empirical mean and the empirical standard deviation of z are 0 and 1, respectively. Standardisation can be done in R using function `scale()`.

```
R> vacmot.scaled <- scale(vacmot)
```

Alternative standardisation methods may be required if the data contains observations located very far away from most of the data (outliers). In such situations, robust estimates for location and spread – such as the median and the inter quartile range – are preferable.

6.5 Principal Components Analysis

Principal components analysis (PCA) transforms a multivariate data set containing metric variables to a new data set with variables – referred to as principal components – which are uncorrelated and ordered by importance. The first variable (principle component) contains most of the variability, the second principle component contains the second most variability, and so on. After transformation, observations (consumers) still have the same relative positions to one another, and the dimensionality of the new data set is the same because principal components analysis generates as many new variables as there were old ones. Principal components analysis basically keeps the data space unchanged, but looks at it from a different angle.

Principal components analysis works off the covariance or correlation matrix of several numeric variables. If all variables are measured on the same scale, and have similar data ranges, it is not important which one to use. If the data ranges are different, the correlation matrix should be used (which is equivalent to standardising the data).

In most cases, the transformation obtained from principal components analysis is used to project high-dimensional data into lower dimensions for plotting purposes. In this case, only a subset of principal components are used, typically the first few because they capture the most variation. The first two principal components can easily be inspected in a scatter plot. More than two principal components can be visualised in a scatter plot matrix.

The following command generates a principal components analysis for the Australian travel motives data set:

```
R> vacmot.pca <- prcomp(vacmot)
```

In `prcomp`, the data is centered, but not standardised by default. Given that all variables are binary, not standardising is reasonable. We can inspect the resulting object `vacmot.pca` by printing it:

```
R> vacmot.pca
```

The print output shows the standard deviations of the principal components:

```
Standard deviations (1, .., p=20):

 [1] 0.81 0.57 0.53 0.51 0.47 0.45 0.43 0.42 0.41 0.38
[11] 0.36 0.36 0.35 0.33 0.33 0.32 0.31 0.30 0.28 0.24
```

These standard deviations reflect the importance of each principal component. The print output also shows the rotation matrix, specifying how to rotate the original data matrix to obtain the principal components:

```
Rotation (n x k) = (20 x 20):

                                      PC1     PC2     PC3
rest and relax                     -0.063  0.0120  0.1345
luxury / be spoilt                 -0.109  0.3932 -0.1167
do sports                          -0.095  0.1456 -0.0456
excitement, a challenge            -0.277  0.2227 -0.2103
not exceed planned budget          -0.286 -0.1561  0.5831
realise creativity                 -0.110 -0.0122 -0.0153
fun and entertainment              -0.279  0.5205  0.0865
good company                       -0.284 -0.0097  0.1291
health and beauty                  -0.140  0.0509  0.0039
free-and-easy-going                -0.317  0.0575  0.2445
entertainment facilities           -0.118  0.3207  0.0050
not care about prices              -0.049  0.2397 -0.2988
life style of the local people     -0.353 -0.2672 -0.3982
intense experience of nature       -0.241 -0.2133 -0.0763
cosiness/familiar atmosphere       -0.132 -0.0133  0.2017
maintain unspoilt surroundings     -0.307 -0.3361  0.0052
everything organised               -0.092  0.1649  0.0780
unspoilt nature/natural landscape -0.269 -0.1831 -0.0556
cultural offers                    -0.260 -0.1160 -0.4282
change of surroundings             -0.259  0.0919  0.1043
```

Only the part of the rotation matrix corresponding to the first three principal components is shown here. The column `PC1` indicates how the first principal component is composed of the original variables. This shows that the first principal component separates the two answer tendencies "almost no motives apply" and "all motives apply", and therefore is not of much managerial value. For the second principal component, the variables loading highest are FUN and ENTERTAINMENT, LUXURY / BE SPOILT and to MAINTAIN AN UNSPOILT SURROUNDING. For the third principal component not exceeding the planned budget, cultural offers, and the life style of the local people are important variables.

We can obtain further information on the fitted object with the `summary` function. For objects returned by function `prcomp`, the function `summary` gives:

```
R> print(summary(vacmot.pca), digits = 2)

Importance of components:
                        PC1  PC2   PC3   PC4  PC5   PC6
Standard deviation     0.81 0.57 0.529 0.509 0.47 0.455
Proportion of Variance 0.18 0.09 0.077 0.071 0.06 0.057
```

```
Cumulative Proportion  0.18 0.27 0.348 0.419 0.48 0.536
                         PC7   PC8   PC9  PC10  PC11  PC12
Standard deviation     0.431 0.420 0.405 0.375 0.364 0.360
Proportion of Variance 0.051 0.048 0.045 0.039 0.036 0.035
Cumulative Proportion  0.587 0.635 0.681 0.719 0.756 0.791
                        PC13 PC14 PC15  PC16  PC17  PC18
Standard deviation     0.348 0.33 0.33 0.320 0.306 0.297
Proportion of Variance 0.033 0.03 0.03 0.028 0.026 0.024
Cumulative Proportion  0.824 0.85 0.88 0.912 0.938 0.962
                        PC19  PC20
Standard deviation     0.281 0.243
Proportion of Variance 0.022 0.016
Cumulative Proportion  0.984 1.000
```

We interpret the output as follows: for each principal component (PC), the matrix lists standard deviation, proportion of explained variance of the original variables, and cumulative proportion of explained variance. The latter two are the most important pieces of information. Principal component 1 explains about one fifth (18%) of the variance of the original data; principal component 2 about one tenth (9%). Together, they explain 27% of the variation in the original data. Principal components 3 to 15 explain only between 8% and 3% of the original variation.

The fact that the first few principal components do not explain much of the variance indicates that all the original items (survey questions) are needed as segmentation variables. They are not redundant. They all contribute valuable information. From a projection perspective, this is bad news because it is not easy to project the data into lower dimensions. If a small number of principal components explains a substantial proportion of the variance, illustrating data using those components only gives a good visual representation of how close observations are to one another.

Returning to the Australian travel motives data set: we now want to plot the data in two-dimensional space. Usually we would do that by taking the first and second principal component. Inspecting the rotation matrix reveals that the first principal component does not differentiate well between motives because all motives load on it negatively. Principal components 2 and 3 display a more differentiated loading pattern of motives. We therefore use principal components 2 and 3 to create a perceptual map (Fig. 6.5):

```
R> library("flexclust")
R> plot(predict(vacmot.pca)[, 2:3], pch = 16,
+   col = "grey80")
R> projAxes(vacmot.pca, which = 2:3)
```

`predict(vacmot.pca)[, 2:3]` contains the rotated data and selects principal components 2 and 3. Points are drawn as filled circles (`pch = 16`) in light grey (`col`). Function `projAxes` plots how the principal components are composed of the original variables, and visualises the rotation matrix. As can be seen, NOT EXCEEDING THE PLANNED BUDGET (represented by the arrow pointing in the top slightly left direction) is a travel motive that is quite unique, whereas, for example,

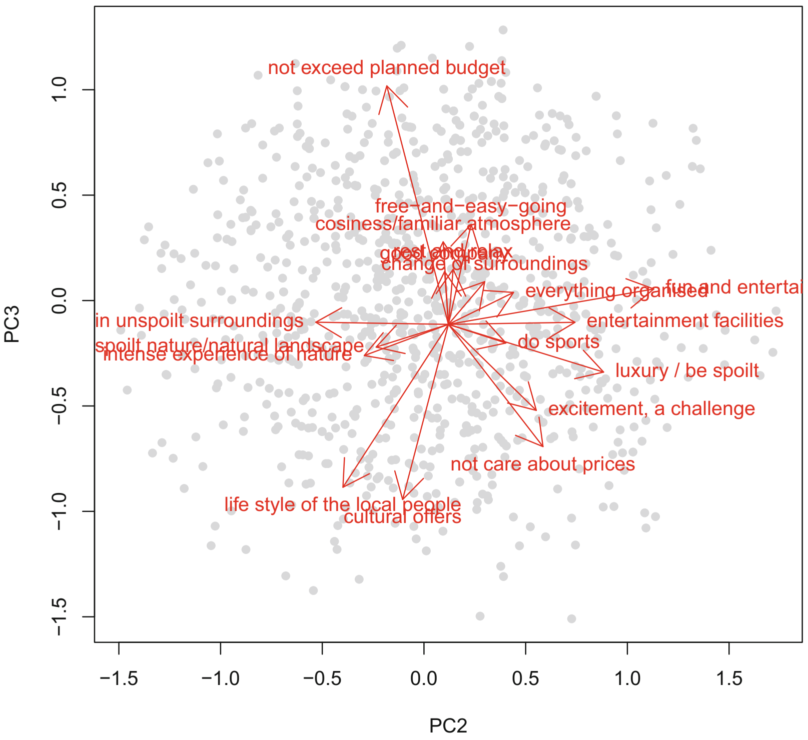

Fig. 6.5 Principal components 2 and 3 for the Australian travel motives data set

interest in the LIFESTYLE OF LOCAL PEOPLE, and interest in CULTURAL OFFERS available at destinations often occur simultaneously (as indicated by the two arrows both pointing to the left bottom of Fig. 6.5). A group of nature-oriented travel motives (arrows pointing to the left side of the chart) stands in direct contrast to the travel motives of LUXURY, EXCITEMENT, and NOT CARING ABOUT PRICES (arrows pointing to the right side of the chart).

Sometimes principal components analysis is used for the purpose of reducing the number of segmentation variables before extracting market segments from consumer data. This idea is appealing because more variables mean that the dimensionality of the problem the segment extraction technique needs to manage increases, thus making extraction more difficult and increasing sample size requirements (Dolnicar et al. 2014, 2016). Reducing dimensionality by selecting only a limited number of principal components has also been recommended in the early segmentation literature (Beane and Ennis 1987; Tynan and Drayton 1987), but has been since shown to be highly problematic (Sheppard 1996; Dolnicar and Grün 2008).

This will be discussed in detail in Sect. 7.4.3, but the key problem is that this procedure *replaces* original variables with a subset of factors or principal components. If all principal components would be used, the same data would be used; it would merely be looked at from a different angle. But because typically only a small subset of resulting components is used, a different space effectively serves as the basis for extracting market segments. While using a subset of principal components as segmentation variables is therefore not recommended, it is safe to use principal components analysis to explore data, and identify highly correlated variables. Highly correlated variables will display high loadings on the same principal components, indicating redundancy in the information captured by them. Insights gained from such an exploratory analysis can be used to remove some of the original – redundant – variables from the segmentation base. This approach also achieves a reduction in dimensionality, but still works with the original variables collected.

6.6 Step 4 Checklist

Task	Who is responsible?	Completed?
Explore the data to determine if there are any inconsistencies and if there are any systematic contaminations.		□
If necessary, clean the data.		□
If necessary, pre-process the data.		□
Check if the number of segmentation variables is too high given the available sample size. You should have information from a minimum of 100 consumers for each segmentation variable.		□
If you have too many segmentation variables, use one of the available approaches to select a subset.		□
Check if the segmentation variables are correlated. If they are, choose a subset of uncorrelated segmentation variables.		□
Pass on the cleaned and pre-processed data to Step 5 where segments will be extracted from it.		□

References

Baumgartner H, Steenkamp JBEM (2001) Response styles in marketing research: a cross-national investigation. J Mark Res 38(2):143–156

Beane TP, Ennis DM (1987) Market segmentation: a review. Eur J Mark 21(5):20–42

Chapman CN, McDonnell Feit E (2015) R for marketing research and analytics. UseR!. Springer International Publishing, Cham

Dolnicar S, Grün B (2007) Assessing analytical robustness in cross-cultural comparisons. Int J Tour Cult Hosp Res 1(2):140–160

Dolnicar S, Grün B (2008) Challenging “factor-cluster segmentation”. J Travel Res 47(1):63–71

Dolnicar S, Grün B, Leisch F, Schmidt K (2014) Required sample sizes for data-driven market segmentation analyses in tourism. J Travel Res 53(3):296–306

Dolnicar S, Grün B, Leisch F (2016) Increasing sample size compensates for data problems in segmentation studies. J Bus Res 69:992–999

Hui C, Triandis H (1989) Effects of culture and response format on extreme response style. J Cross Cult Psychol 20(3):2960–309

Leisch F (2006) A toolbox for k-centroids cluster analysis. Comput Stat Data Anal 51(2):526–544

Likert R (1932) A technique for the measurement of attitudes. Arch Psychol 140:1–55

Marin G, Gamba R, Marin B (1992) Extreme response style and acquiescence among hispanics – the role of acculturation and education. J Cross Cult Psychol 23(4):498–509

Paulhus D (1991) Measurement and control of response bias. In: Robinson J, Shaver P, Wrightsman L (eds) Measures of personality and social psychological attitudes. Academic, San Diego, pp 17–59

Putler DS, Krider RE (2012) Customer and business analytics: applied data mining for business decision making using R. Chapman&Hall/CRC, Boca Raton

Sarkar D (2008) Lattice: multivariate data visualization with R. Springer, New York

Sheppard AG (1996) The sequence of factor analysis and cluster analysis: differences in segmentation and dimensionality through the use of raw and factor scores. Tour Anal 1:49–57

Tukey J (1977) Exploratory data analysis. Addison-Wesley, Reading

Tynan AC, Drayton J (1987) Market segmentation. J Mark Manag 2(3):301–335