13. Data preprocessing

Contents

Introduction

These notes consider Chapter 13 (pages 267 - 279) of the handbook on data preprocessing. Wide format data is assumed throughout this section.

Initial processing

Certain steps needs to be taken in order to ensure that the quality of the data is high enough for analysis. Test and analysis methods can be run to practically any data with R so it is the responsibility of the user to make sure that any results are drawn from sensible data.

Inspection of the raw data

The first step is to make sure that the data matrix has been correctly read in and contains all the data in the correct format. After this, the distribution and summary statistics should be checked.

- **Does the dataset look correct?**
  - Are all variables (columns) there with correct names and types?
  - Are expected number of observations (number of rows) there?
  - Check formats for e.g. decimal numbers and dates/times
- **Are the statistics as expected?**
  - Visually estimate the distribution using histograms and/or density functions.
    - Are there multiple peaks? Is the distribution asymmetric?
    - Are there outliers i.e. abnormally deviant values?
    - How much data is missing?
  - Calculate summary statistics
    - Are the numbers as expected?

Missing values

Oftentimes a data row may be missing one or more values. This is especially typical e.g. with long surveys where one might forget to answer all the questions or chooses not to answer some due to personal reasons. There are a few options how to handle missing values.

- **Dropping the whole row** is an easy and fast way to handle missing data. The downside is, of course, that the dataset can quickly get too small and in any case valuable information is lost. This should be done only if the dataset is large and only few rows are dropped.
- **Dropping the value(s)** is a milder version of the above method i.e. values of existing variables are used in the analysis but the missing ones are removed. The downside with this is that analyses of different variables using different amounts of values are as not comparable as with equal amount.
- **Imputation** i.e. replacement of missing values with e.g. the mean or median of the available values of the variable. Although this sounds tempting as it does not affect the mean/median, this could lead to type 2 error (see the notes in the previous chapter). This method is even worse if the missing data is due to a systematic effect.
  - Values may be systematically missing e.g. in a case where people tend to not answer a certain question in a survey.

There are many R packages to estimate the properties of missing data, such as `VIM`, `Mice`, `mitools`, and `Hmmc`.

Outlier values

In case of outliers, it might be tempting to drop them to decrease the dispersion in the data: even a single data point may significantly affect the mean and standard deviation. In worse case, they may also greatly affect the statistical tests and models as well.

- If there are irrefutable errors in data acquisition, it is possible just to throw these outlier values away. For instance, a human error when inputting values from physical survey forms to digital format or an obvious measurement error of some gadget can be discarded (or preferably corrected if the original material still exists).
- Some outlier values could still be dropped even if they are in some sense "real". For instance, in case where a test subject is not fully or at all focusing on the task at hand and, therefore, yields abnormal results. These scenarios are more difficult to spot.
- It still can happen that an outlier value cannot be plausibly dismissed and, therefore, is an important piece of information that can make the analysis more difficult (e.g. by violating the assumption of normality). It is easy to imagine that, say, some experimental treatment can be orders of magnitude more expensive than the commonly used ones.

Value combinations

Sometimes it can be useful to combine variables describing some common property. For instance, a questionnaire can contain dozens of questions on different symptoms caused by various reasons. In case of, say, anxiety a subset of these question could be combined to give a single value to describe this symptom on some scale. It can be easier to handle combined variables rather than multiple values but, on the other hand, some information about the original data is lost.

Common ways to combine values are by summing or by taking the mean.

- **Summed variables** are less prone to measurement errors than the original values. The problem with summed variables is that they can be greatly affected by missing values.
- **Mean variables** are more useful as they are less affected by missing values and have the same dimension as the original values.

Transformations

Many statistical tests assume that the data is approximately normally distributed. This is not always the case. Before analysis, the general shape and relevant summary values should be checked using histograms and box plots. In case the distribution is skewed, an attempt to correct this can be made using *invariant* transformations to the data. The idea is to change the dimension of the data so that the distribution would be closer to normal than the original.

This might sound tempting but there are caveats. The results of the tests are in the transformed distribution, not in the original so the interpretation can be difficult. Additionally, the exactly same transformation must be done to all variables used in the analysis in order for them to be comparable. Consider, if applicable, using some non-parametric methods rather than transformations.

Common transformation types for varying degrees of skewness are listed in the table below. In practice, it may take some trial and error to find out the best suited one for a given case. Note that the transformations are identical expect for the mirroring for left-skewed distributions.

Table 1: Common transformation functions.

Direction	Low	Moderate	High
Right	$\sqrt{x}$	$\ln(x)$	$\frac{1}{x}$
Left	$\sqrt{k-x}$	$\ln(k-x)$	$\frac{1}{k-x}$

Transformations using R

If not already done, install the example datasets (should be installed automatically with the R environment, though).

```
install.packages("remotes")
library(remotes)

install_url(
  "http://emotion.utu.fi/wp-content/uploads/2019/11/nummenmaa_1.0.tar.gz",
  dependencies=TRUE
)
```

Dataset *kipu* (pain) is needed here. This is a real-world dataset containing information about pain and its effects. See page 703 for full description. First, let's make sure it works.

```
library(nummenmaa)
names(kipu)
```

```
##  [1] "SUKUP"                "KATISUUS"             "KOULUTUS"
##  [4] "AKUUTTIKIPU"          "VIIMEAIKAINENKIPU"    "KROONINENKIPU"
##  [7] "MIGREENI"             "PAANSARKY"            "VATSAKIPU"
## [10] "SELKAKIPU"            "RAAJAKIPU"            "RESPTIVAPAAKIPULAAKE"
## [13] "RESEPTIKIPULAAKE"     "IKA"                  "PAINO"
## [16] "PITUUS"               "FYYSINENTYOPROS"      "ISTUMATYOPROS"
## [19] "PAHINKIPU"            "LIEVINKIPU"           "KESKIMKIPU"
## [22] "KIPUNYT"              "KIPULAAKEAUTTAA"      "KIPUVAIKYLEISESTI"
## [25] "KIPUVAIKMIELIALAAN"   "KIPUVAIKKAVELYYN"     "KIPUVAIKTYOHON"
## [28] "MASENNUS"             "AHDISTUS"             "ILO"
## [31] "SURU"                 "VIHA"                 "PELKO"
## [34] "HAMMASTYS"            "INHO"
```

Let's first add two new combined variables to the dataset. Original variables describe how pain affects certain aspects (on a scale from 0 to 10) so their sum and mean should give some estimate of the combined effect. Variables are *KIPUVAIKYLEISESTI* (generally), *KIPUVAIKMIELIALAAN* (mentally), and *KIPUVAIKKAVELYYN* (to walking).

```
# Summed
kipu$KIPUSUM <- with(kipu, (KIPUVAIKYLEISESTI + KIPUVAIKMIELIALAAN + KIPUVAIKKAVELYYN))
# Mean
kipu$KIPUMEAN <- with(kipu, (KIPUVAIKYLEISESTI + KIPUVAIKMIELIALAAN + KIPUVAIKKAVELYYN) / 3)
# Check
mean(kipu$KIPUSUM) ; mean(kipu$KIPUMEAN)
```

```
## [1] 10.49858
```

```
## [1] 3.499525
```

Now, let's use the new variable *KIPUMEAN* to test some transformations. Note that the scales start from 0 so a constant needs to be added in order to avoid infinities with logarithmic and division transformations.

```
# sqrt(x)
kipu$KIPUSQRT <- sqrt(kipu$KIPUMEAN)

# ln(x)
kipu$KIPULOG <- log(kipu$KIPUMEAN + 1)

# 1/x
kipu$KIPUINVERSE <- 1 / (kipu$KIPUMEAN + 1)
```

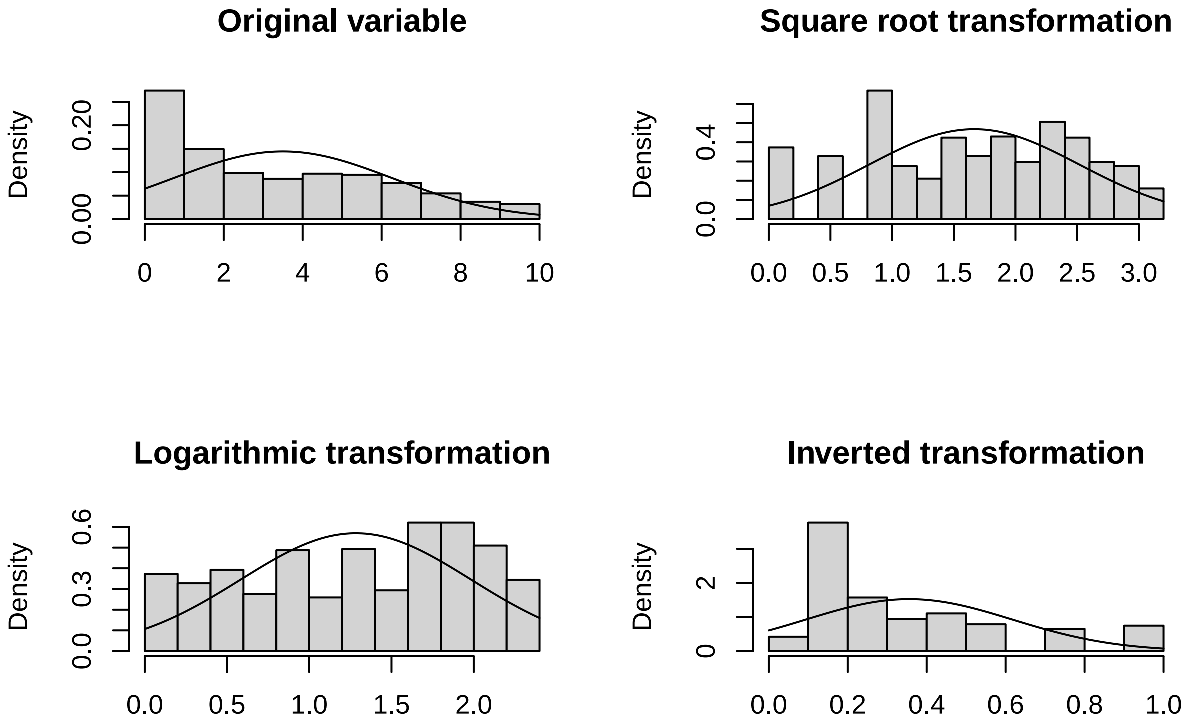
Plots to visualize these distributions in comparison to the normal distribution with the same mean and standard deviation.

```
par(mfrow=c(2, 2))
# Original
hist(kipu$KIPUMEAN, freq = FALSE, breaks = 12,
     xlab = NULL, main = "Original variable"
)
curve(dnorm(x, mean = mean(kipu$KIPUMEAN), sd = sd(kipu$KIPUMEAN)), add = TRUE)

# Sqrt
hist(kipu$KIPUSQRT, freq = FALSE, breaks = 12,
     xlab = NULL, main = "Square root transformation"
)
curve(dnorm(x, mean = mean(kipu$KIPUSQRT), sd = sd(kipu$KIPUSQRT)), add = TRUE)

# Log
hist(kipu$KIPULOG, freq = FALSE, breaks = 12,
     xlab = NULL, main = "Logarithmic transformation"
)
curve(dnorm(x, mean = mean(kipu$KIPULOG), sd = sd(kipu$KIPULOG)), add = TRUE)

# 1/x
hist(kipu$KIPUINVERSE, freq = FALSE, breaks = 12,
     xlab = NULL, main = "Inverted transformation"
)
curve(dnorm(x, mean = mean(kipu$KIPUINVERSE), sd = sd(kipu$KIPUINVERSE)), add = TRUE)
```



In this case, square root and logarithmic transformations yield better results than the inversion.

Testing normality

Visual comparison of observed or transformed distributions to their ideal counterparts is one way to estimate normality. There are also tests and other tools to do this. Some of them are introduced in this section.

Quantile-quantile (Q-Q) plots

Q-Q plot is a graphical method to compare two distributions by plotting their quantiles against each other. If the result is approximately the identity line ($y(x) = x$), the distributions are similar. More generally, if the result follows a linear curve, the distributions are linearly related. In other words, if an observed distribution and the normal distribution are compared, a linear plot would indicate normality in the observed distribution.

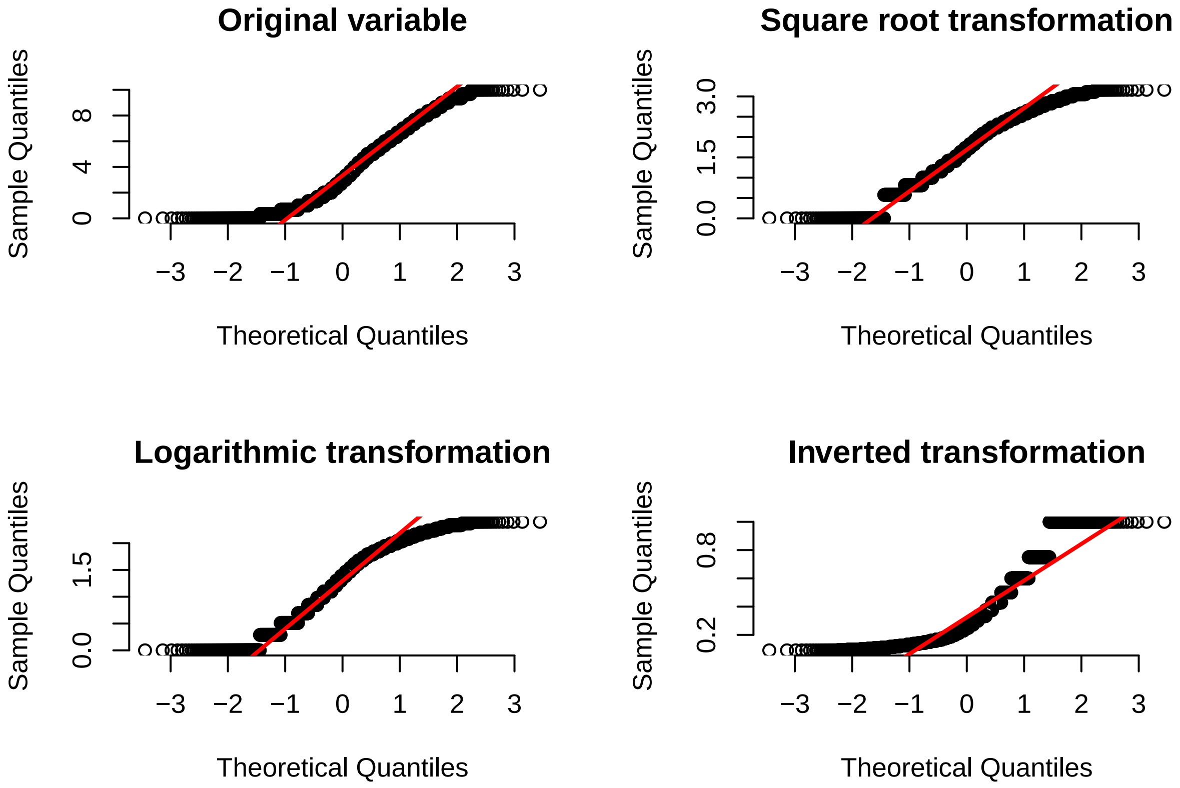
In R, this is done with the `qqnorm()` (plots the data) and `qqline()` (plots the theoretical ideal) functions.

```r
par(mfrow=c(2, 2))
# Original
qqnorm(kipu$KIPUMEAN, frame = FALSE, main = "Original variable")
qqline(kipu$KIPUMEAN, col = "red", lwd = 2)

# Sqrt
qqnorm(kipu$KIPUSQRT, frame = FALSE, main = "Square root transformation")
qqline(kipu$KIPUSQRT, col = "red", lwd = 2)
```

```r
# Log
qqnorm(kipu$KIPULOG, frame = FALSE, main = "Logarithmic transformation")
qqline(kipu$KIPULOG, col = "red", lwd = 2)

# 1/x
qqnorm(kipu$KIPUINVERSE, frame = FALSE, main = "Inverted transformation")
qqline(kipu$KIPUINVERSE, col = "red", lwd = 2)
```



Seems like none of these are normally distributed. This is probably due to the fact that zero values were overrepresented in the data (i.e. answers that pain did not cause any effects) meaning no transformation can help with this.

Numeric tests

There are some methods to numerically estimate normality e.g. the Kolmogorov–Smirnov test and the Shapiro–Wilk test. These test the null hypothesis that an observed distribution follows the normal distribution i.e. a small *p*-value from these tests would suggest against normality. Using the same examples as above.

```r
shapiro.test(kipu$KIPUMEAN)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  kipu$KIPUMEAN
## W = 0.92792, p-value < 2.2e-16
```

```
shapiro.test(kipu$KIPUSQRT)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  kipu$KIPUSQRT
## W = 0.96357, p-value < 2.2e-16
```

```
shapiro.test(kipu$KIPULOG)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  kipu$KIPULOG
## W = 0.94466, p-value < 2.2e-16
```

```
shapiro.test(kipu$KIPUINVERSE)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  kipu$KIPUINVERSE
## W = 0.83391, p-value < 2.2e-16
```

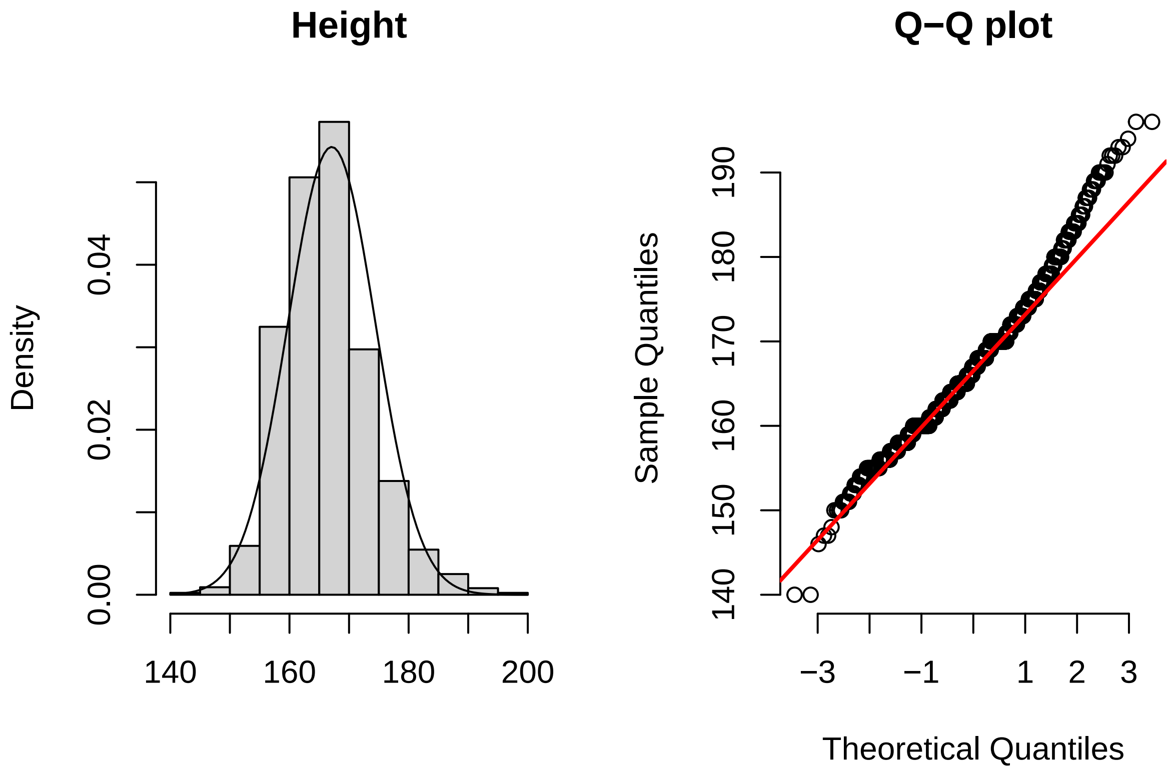
These p-values would also suggest that none of the distributions are normal. But should the results of these numeric tests be blindly accepted? If the result suggest normality, then yes. If this is not the case, additional inspection needs to be done. For example, the height *(PITUUS)* of people is known to follow the normal distribution in the population. What about in case of our dataset?

```
par(mfrow=c(1, 2))

# Histogram of heights
hist(kipu$PITUUS, freq = FALSE, breaks = 12, xlab = NULL, main = "Height")

# Ideal normal distribution based on mean and sd of the data
curve(dnorm(x, mean = mean(kipu$PITUUS), sd = sd(kipu$PITUUS)), add = TRUE)

# Q-Q plot
qqnorm(kipu$PITUUS, frame = FALSE, main = "Q-Q plot")
qqline(kipu$PITUUS, col = "red", lwd = 2)
```



Even in this data, the height seems roughly to follow the normal distribution. But the test

```
shapiro.test(kipu$PITUUS)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  kipu$PITUUS
## W = 0.98052, p-value = 1.022e-14
```

would indicate otherwise. This happens because the Shapiro–Wilk test is extremely powerful to find non-normality and, therefore, should not be trusted blindly when the p-value is small.

Other useful values are the *skewness* and *kurtosis* of a distribution. Absolute values less than 1 would indicate normality in these cases.

```
library(moments)
skewness(kipu$PITUUS)
```

```
## [1] 0.5040387
```

```
kurtosis(kipu$PITUUS)
```

```
## [1] 3.817965
```

These values would suggest that the distribution is rather symmetric but perhaps too heavy-tailed to be normal.