Chapter 01 of DSUR

Peter Baumgartner

2018-05-27

Contents

1 Basics: Measures of Central Tendency

1.1 How to get a small data set into R?

Number of friends of 11 Facebook user: 108, 103, 252, 121, 93, 57, 40, 53, 22, 116, 98.

1.1.1 Assign values to a vector

```
> # number of facebook friends = nff
> nff <- c(108, 103, 252, 121, 93, 57, 40, 53, 22, 116, 98)
> nff <- sort(nff)
> nff
```

```
 [1]  22  40  53  57  93  98 103 108 116 121 252
```

1.1.2 Read data values from keyboard

After running the following code you have to set your cursor into the console and provide the data. There are 2 possibilities: * Enter the data manually and separate each entry with >ENTER< * Copy a string of data (e.g. from a PDF table), where each data is spearated by a blank

In both cases: Terminate the input with an extra >ENTER<

```
> nff_scan <- as.vector(scan(file = ""))
> nff <- sort(nff_scan)
> nff_scan
```

1.2 The Mode

There is no mode-function in the base R module. See https://bit.ly/R-mode. But there are many possibilites to programm this function.

Before I will demonstrate this, I need to add another number into the data set in order to get the frequencey of one number higher than the others.

```
> nff_mode <- c(nff, 53)
```

1.2.1 Variante 1: Simple but good!

```
> Mode <- function(x) {
+   ux <- unique(x)
+   ux[which.max(tabulate(match(x, ux)))]
+ }
> Mode(nff_mode)
```

```
[1] 53
```

1.2.2 Version 2: with NA

```
> Mode <- function(x, na.rm = FALSE) {
+     if (na.rm) {
+         x = x[!is.na(x)]
+     }
+     ux <- unique(x)
+     ux[which.max(tabulate(match(x, ux)))]
+ }
> Mode(nff_mode)
```

```
[1] 53
```

1.2.3 Version 3: with package `modeest`

```
> library(modeest)
> mlv(nff_mode, method = "mfv")
```

```
Mode (most likely value): 53
Bickel's modal skewness: 0.5
Call: mlv.default(x = nff_mode, method = "mfv")
```

1.3 The Median

1.3.1 Just the function

```
> median(nff)
```

```
[1] 98
```

1.3.2 Without outlier

```
> median(nff[1:10])
```

```
[1] 95.5
```

1.4 The Mean

1.4.1 Just the function

```
> mean(nff)
```

```
[1] 96.63636
```

1.4.2 Without outlier

```
> mean(nff[1:10])
```

```
[1] 81.1
```

1.4.3 With NA Value not removed

```
> mean(nff[c(1:10, NA)])
```

```
[1] NA
```

1.4.4 With NA Value removed

```
> mean(nff[c(1:10, NA)], na.rm = TRUE)
```

```
[1] 81.1
```

1.4.5 With outlier but trimmed

```
> mean(nff, trim = 0.1)
```

```
[1] 87.66667
```

```
> mean(nff[2:10])
```

```
[1] 87.66667
```

1.5 The Range

1.5.1 Just the function

```
> x <- range(nff)
> xr <- x[2] - x[1]
> cat("Range:", x[2], "-", x[1], "=", xr)
```

```
Range: 252 - 22 = 230
```

1.5.2 Without outlier

```
> x <- range(nff[1:10])
> xr <- x[2] - x[1]
> cat("Range:", x[2], "-", x[1], "=", xr)
```

```
Range: 121 - 22 = 99
```

1.6 Upper and Lower Quartile

```
> quantile(nff, type = 1)
```

```
  0%  25%  50%  75% 100%
  22   53   98  116  252
```

1.7 The Interquartile Range

1.7.1 Just the function

```
> IQR(nff)
```

```
[1] 57
```

1.7.2 Computed with type = 1

This measure has 9 different calculation methods (quantile algorithms) which really matter because of their big differences. Standard is type = 7 (results in 57), whereas type = 1 results in 63.

```
> IQR(nff, type = 1)
```

```
[1] 63
```

1.8 Self-Tests

1.8.1 Self-test p.25

```
> treadmill <- c(18,16,18,24,23,22,22,23,26,29,32,34,34,36,36,43,42,49,46,46,57)
> Mode <- function(x, na.rm = FALSE) {
+     if (na.rm) {
+         x = x[!is.na(x)]
+     }
+     ux <- unique(x)
+     tab <- tabulate(match(x, ux));
+     ux[tab == max(tab)]
+
+ }
> Mode(treadmill)
```

```
[1] 18 23 22 34 36 46
```

```
> median(treadmill)
```

```
[1] 32
```

```
> mean(treadmill)
```

```
[1] 32.19048
```

```
> quantile(treadmill, type = 6)
```

```
  0%  25%  50%  75% 100%
16.0 22.5 32.0 42.5 57.0
```

```
> range(treadmill)
```

```
[1] 16 57
```

```
> IQR(treadmill, type = 6)
```

```
[1] 20
```

1.8.2 Self-Test p.27

What's the probability that someone wo threw themselves of Beachy Head was 30 years or older?

- First we convert 30 into a z-score. Suppose the mean of the suicide scores was 36, and the standard deviation 13; then 30 will bekomme (30-36)/13 = -0.4615385.
- We then look up this value in the column labelled "Bigger Portion" (i.e., the area above the value -0.4615385).
- I get the value of 32.28%, or put another way, there is a chance of 32.28% that a suicide victim was aged 30 or less. We can also say that there is a 67.72% chance that a suicide victim wa older than 30.