WS 2023/24

UE - Regression Methods and Computational Statistics
Sebastian Fuchs, Jonas Beck

Exercise sheet No. 1

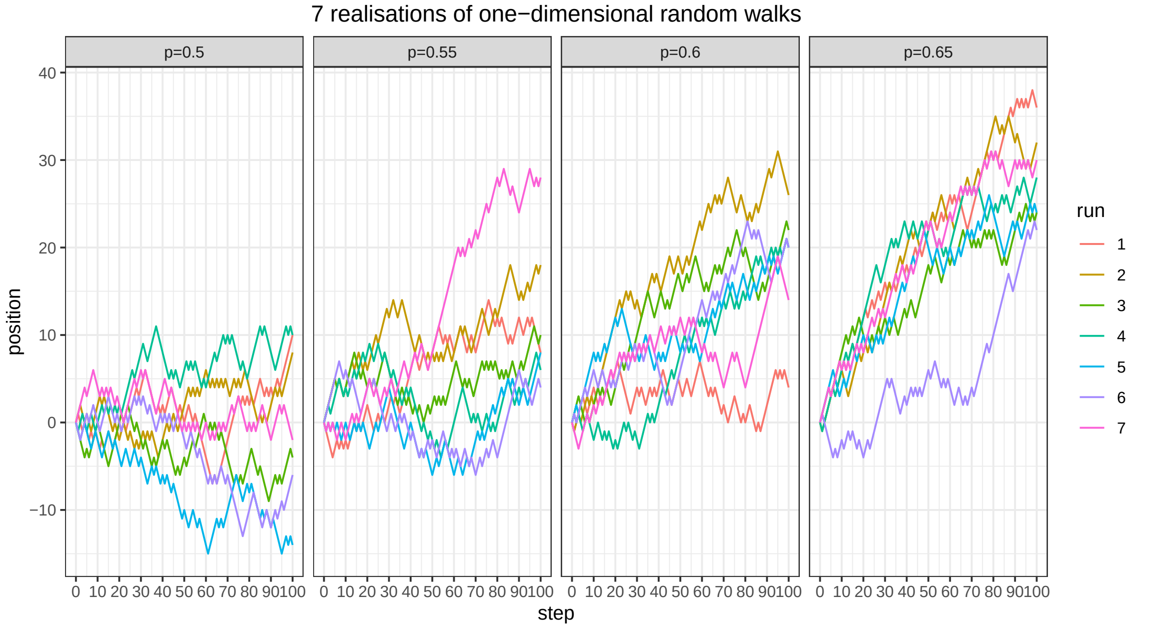
1. **R Warm-up:**

   (a) Use a double loop to write $i * j$ in row number i and column number j of the $k \times n$ matrix M (e.g. for $k = 10$ and $n = 5$). Find an alternative way to construct M without double loops or any loop at all.

   (b) What does the following code snippet do? Try to explain each step and command line. If you need help, use the R help function (or F1).

```
f <- function(x){
  N <- length(x)
  x <- sort(x)
  y <- cumsum(table(x))/N
  x <- unique(x)
  return(approxfun(x = x, y = y, method = "constant", yleft = 0, yright = 1))
}
n <- 10
x <- sample(1:10, n, replace = T)
grid <- seq(0,10,0.01)
plot(grid, f(x)(grid), type = "l")
lines(grid, ecdf(x)(grid), col = "green")
```

   (c) Implement the following random walk(s) and illustrate the results (i.e. the position x_i at step i) with the package `ggplot2`. A disoriented man walks with probability p one step ahead and with probability $q = 1 - p$ one step back. He starts at $x_0 = 0$. Use the `ggplot2` function `facet_wrap` to align versions of the random walk(s) side by side for different probabilities p. What can be observed?



2. **S3-methods:** Find out about S3-methods using the following code snippet. What do the functions `UseMethod()` and `class()` do?

```
f <- function(x) {
  UseMethod("f")
}
f.default <- function(x){
  print("default")
}
f.foo <- function(x){
  print("foo")
}

x <- 10
class(x) <- "foo"
f(x)

## [1] "foo"

class(x) <- "bar"
f(x)

## [1] "default"

class(x) <- c("foo", "bar")
f(x)

## [1] "foo"
```

Implement your own S3-method, called `myMean()`, which should do the following:

```
# input: numeric vector; output: mean
myMean(c(1,4,3,8,7,2,6))

## [1] 4.428571

# input: data.frame with numeric columns; output: colwise mean
df <- data.frame(x=runif(10), y=rnorm(10,4), z=rexp(10))
myMean(df)

##         x         y         z
## 0.4880968 3.9884324 1.1660806

# otherwise
myMean(c("a","b"))

## [1] "myMean only works with numeric vectors or numeric data.frames"

myMean(data.frame(x=c(1,2), y = c("1","2")))

## [1] "myMean only works with numeric vectors or numeric data.frames"
```

3. **tidyr, dplyr and knitr:** Find out about the packages `tidyr` and `dplyr`. Use the attached data set `measurement.txt` and solve the following exercises with the help of these two R-packages (alternatives are possible). Furthermore, inform yourself about *knitr* and illustrate all results in a nice (and informative) *knitr* document.

   (a) Sort the data w.r.t. to `T1.grp1` (decreasing) and depict the first 8 rows of the so obtained data.frame in an appropriate table (centered with labels and optional with colors)

   (b) Compute the mean values, medians and standard deviations for all groups and measurements and report the results.

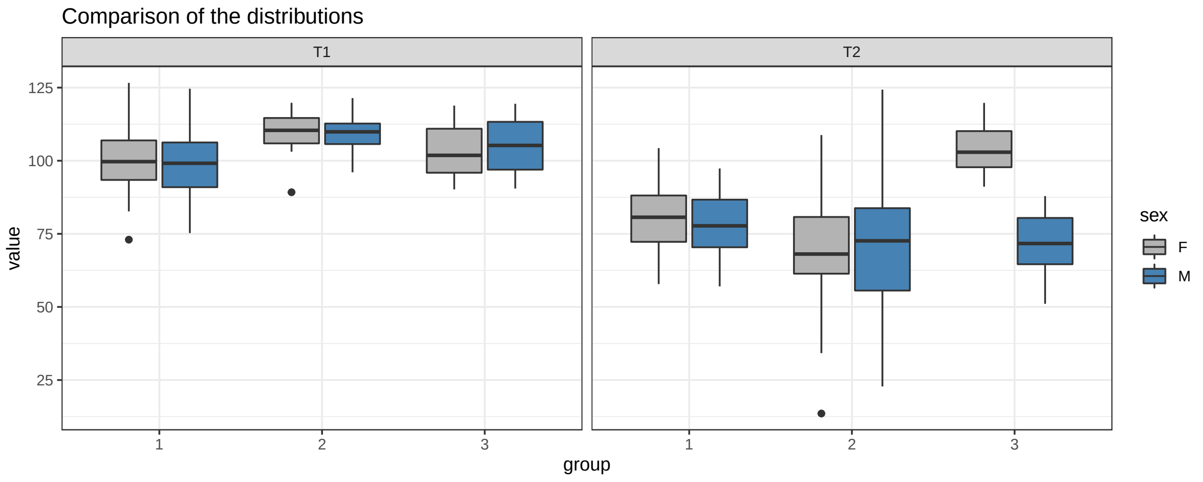
   (c) Convert the data into a long-table format (see Table 1) and depict the first and last 8 lines of the data set.

   (d) Compute the variance of all subjects for each time (T1 and T2) and report the results.

   (e) Use boxplots (`ggplot2`) to compare the distributions for the variables sex, time and group (see example). Give an interpretation.

Table 1: Long table format

subject	sex	time	group	value
1	F	T1	1	82.67
1	F	T1	2	105.88
1	F	T1	3	100.24
1	F	T2	1	80.66
1	F	T2	2	47.20
1	F	T2	3	102.91
100	F	T1	1	104.55
100	F	T1	2	110.39
100	F	T1	3	101.84
100	F	T2	1	87.66
100	F	T2	2	68.08
100	F	T2	3	109.54


Comparison of the distributions
T1
T2
value
group
sex
F
M