1 Part I

1.1

R knows six different vector types, namely: logical, integer, real, complex, character (string) and raw. To give some examples for every type:

```
> # define logical object
> log <- TRUE
> is.logical(log)
[1] TRUE
> # define integer object
> int <- 1:5
> is.integer(int)
[1] TRUE
> # define real (numeric double) object
> real <- 2.5
> is.double(real)
[1] TRUE
> # define complex object
> comp <- 1+2i
> is.complex(comp)
[1] TRUE
> # define character (string) object
> char <- "a"
> is.character(char)
[1] TRUE
> # define raw object
> rawd <- as.raw(22) # corresponds to 16</pre>
> is.raw(rawd)
[1] TRUE
```

1.2

Difference between generic and numeric vector:

- An *atomic* vector contains only one single "atomic" data type in all entries. An example would be a vector which contains only integers.
- A generic vector (like a list) can contain different types of data. An example would be a vector which contains characters and numbers.

1.3

To explain: A data frame is a list, but not evey list is a data frame.

- A list is an object containing collections of objects. The types of the entries inside of the list can be different. It is for example allowed that a list contains a vector of real values (doubles) and a vector of characters. The length of the containing vectors can be different.
- A data frame is an object containing colletions of objects. The types of the entries inside of the list can be different. The length of the vectors have to be **the same**. The data frame has a matrix-like structure.

list and data frame are very similar, but the data frame has one more restriction (same length of all vectors). That's why a data frame is always a list, but a list is not always a data frame.

2 Part II