CompStat/R - Paper 2

Group 2: Carlo Michaelis, Patrick Molligo, Lukas Ruff

21 June 2016

Part I: Functions

Functions I

Below we define a function `dropNa` which given an atomic vector `x` as argument, returns `x` after removing missing values.

```
dropNa <- function(x) {
  # expects an atomic vector as an argument and returns it without missing
  # values
  #
  # Args:
  #   x: atomic vector
  #
  # Returns:
  #   The atomic vector x without missing values

  # To remove the NAs, we use simple logical subsetting
  y <- x[!is.na(x)]

  # Return y
  y
}
```

Let's test our implementation with the following line of code:

```
all.equal(dropNa(c(1, 2, 3, NA, 1, 2, 3)), c(1, 2, 3, 1, 2, 3))
```

```
## [1] TRUE
```

As we can see from this positive test, our implementation was successful.

Functions II

Below we define a function `meanVarSdSe` which given a numeric vector `x` as argument, returns the mean, the variance, the standard deviation and the standard error of `x`.

```
meanVarSdSe <- function(x) {
  # expects a numeric vector as an argument and returns the mean,
  # the variance, the standard deviation and the standard error
  #
  # Args:
  #   x: numeric vector
  #
  # Returns:
```

```
  #   a numerical vector containing mean, variance, standard deviation
  #   and standard error of x

  # First we check if we have a numerical vector
  # If not: stop and throw error
  if( !is.numeric(x) )
    stop("Argument need to be numeric.")

  # Create vector object
  y <- vector()

  # Calculate mean, variance, standard deviation and standard error
  y[1] <- mean(x)
  y[2] <- var(x)
  y[3] <- sd(x)
  y[4] <- y[3]/sqrt(length(x))

  # Set names to vector entries
  names(y) <- c("mean", "var", "sd", "se")

  # Return the numeric vector y
  y
}
```

To test the function, we define a numeric vector, which contains numbers from 1 to 100 and use it as an argument for our function `meanVarSdSe`:

```
x <- 1:100
meanVarSdSe(x)
```

```
##       mean        var         sd         se
##  50.500000 841.666667  29.011492   2.901149
```

Finally we can confirm, that the result is of type `numeric`:

```
class(meanVarSdSe(x))
```

```
## [1] "numeric"
```

Functions III

Part II: Scoping and related topics

Scoping I

Scoping II

Scoping III

Dynamic lookup