

# Bioinformatics III

## Seventh Assignment

Thibault Schowing (2571837)

Wiebke Schmitt (2543675)

June 11, 2018

### Exercise 7.1: Missing Data Imputation

#### (a) The script

Listing 1: Missing data imputation script

```
0 print("Assignment_7_-_Schmitt_Schowing")

#setwd("C:/Users/thsch/Desktop/Bioinformatics3/Assignments/Assignment7/Scripts
  ")

5 data = read.table("ms_toy.txt", header=TRUE)
  summary(data)
  # class(data) -> data.frame

10 # Takes a dataframe and the col name and calculate the values for the missing
    data
    # as if they were data "under the detection threshold"
    impute_missing_data <- function(data, colname, QUANTILE_VALUE, FRACTION,
      replace = FALSE){

      current_sd <- apply(data[colname], 2, sd, na.rm = TRUE)
15   current_mean <- apply(data[colname], 2, mean, na.rm = TRUE)

      str = sprintf("Current_sd: %f", current_sd)
      print(str)
      str = sprintf("Current_mean: %f", current_mean)
20   print(str)

      # Chosing the mean of the new distribution -> the 5% quantile for instance
        ?!?!?!
      #QUANTILE_VALUE <- 5

25   quant <- quantile(data[colname], QUANTILE_VALUE/100, na.rm = TRUE)
      str = sprintf("Current %d_percent_quantile: %f", QUANTILE_VALUE, quant)
      print(str)

30   # New mean equals the above quantile
      new_mean = quant

      # New sd -> fraction of the current sd
      #FRACTION <- 1/3
35   new_sd <- FRACTION * current_sd

      print(paste0("New_mean: ", new_mean))
      print(paste0("New_sd: ", new_sd))
```

```
40 # Display the distribution with the chosen mean
hist(as.matrix(data[colname]), main = "Distribution_of_the_Data", xlab = "
    Value")
abline(v=quant, col="red")

45 # We have sd and mean -> rnorm(nb, sd, mean) will generate numbers in the
    distribution

# Numbers of NA in the column (nb of data to generate):
nb_na <- sum(is.na(data[colname]))

50 generated_data = rnorm(nb_na, mean = new_mean, sd = new_sd)

# Trying to make the plots nice but will work for ONE distribution
55 p1 <- hist(generated_data, breaks = 30, freq = TRUE)
p2 <- hist(as.matrix(data[colname]), breaks = 60, freq = TRUE)

# Acceptable graphs for any distribution
#p1 <- hist(generated_data, freq = TRUE)
60 #p2 <- hist(as.matrix(data[colname]), freq = TRUE)

plot(p2, main = "Distribution_of_the_Data", xlab = "Value", col = "blue")
plot(p1, main = "Distribution_of_the_Data", xlab = "Value", col = "red", add
    =T)

65 # Replace the data a copy of the original dataframe if the parameter "
    replace" is set to TRUE.
# Also print the information of the data (copy) before and after the
    replacement
# By default it's FALSE.

70 if (replace){
    data.replace <- data
    print(summary(data.replace))
    data.replace[[colname]][which(is.na(data[colname]))] <- generated_data
75 #data[[colname]][which(is.na(data[colname]))] <- generated_data
p <- hist(as.matrix(data.replace[colname]), breaks = 60, freq = TRUE)
plot(p, main = "Distribution_of_the_data_after_replacement", xlab = "Value
    ", col = "blue")
    print(summary(data.replace))
80 }

}

85 # Playing with the new mean and sd values
impute_missing_data(data = data, 'ctrl.1', 1, 1/3)
impute_missing_data(data = data, 'ctrl.1', 5, 1/3)
impute_missing_data(data = data, 'ctrl.1', 10, 1/3)
90 impute_missing_data(data = data, 'ctrl.1', 15, 1/3)
impute_missing_data(data = data, 'ctrl.1', 20, 1/3)

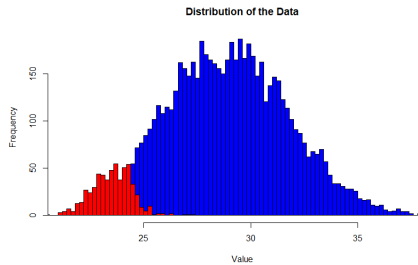
# 5 looks like the best new mean
impute_missing_data(data = data, 'ctrl.1', 5, 1)
95 impute_missing_data(data = data, 'ctrl.1', 5, 1/2)
impute_missing_data(data = data, 'ctrl.1', 5, 1/3)
impute_missing_data(data = data, 'ctrl.1', 5, 1/4)
impute_missing_data(data = data, 'ctrl.1', 5, 1/5)

100
```

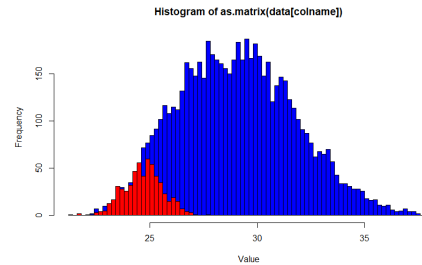
```
# We chose the 5% quartile for the mean and 1 third of the standard deviation  
for the new sd.  
impute_missing_data(data = data, 'ctrl.1', 5, 1/3)  
  
105 # Plot and information of the distribution after data replacement.  
impute_missing_data(data = data, 'ctrl.1', 5, 1/3, TRUE)  
  
# As we have to chose one of the 6, the plot are adapted especially for the  
first column of data.  
# If you want to have good looking plots for any data, switch the two  
commented lines  
110 # in the impute_missing_data function.  
  
# impute_missing_data(data = data, 'ctrl.2')  
# impute_missing_data(data = data, 'ctrl.3')  
# impute_missing_data(data = data, 'kout.1')  
115 # impute_missing_data(data = data, 'kout.2')  
# impute_missing_data(data = data, 'kout.3')
```

**(b) Playing with the new mean and new standard deviation**

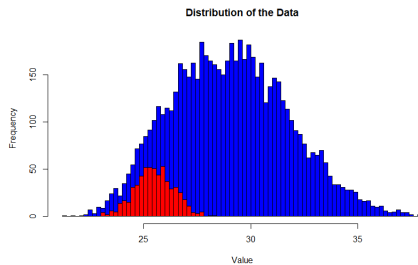
In figure 1 we vary the mean and standard deviation of the generated data distribution. We can see the generated data distribution sliding to the right when we change the new mean to a higher quartile and getting thinner when we take a smaller fraction of the original SD. The nicest fit is with one third of the original standard deviation and the 5% quartile as the new mean (figure 1b). In figure 2 we show the distribution of the data after having replaced the NAs with the values generated with the same parameters as in figure 1b.



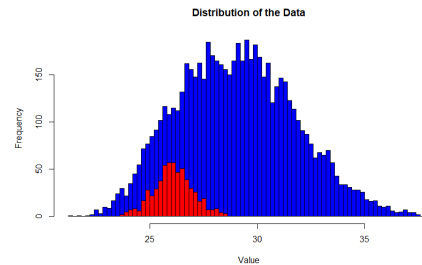
(a) New mean is the 1% quartile of the data distribution and SD is  $1/3$  of the original SD



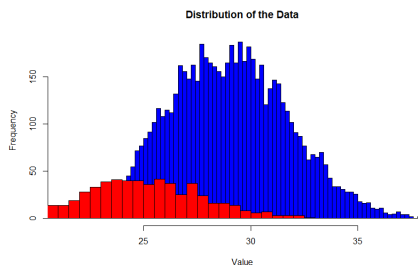
(b) New mean is the 5% quartile of the data distribution and SD is  $1/3$  of the original SD



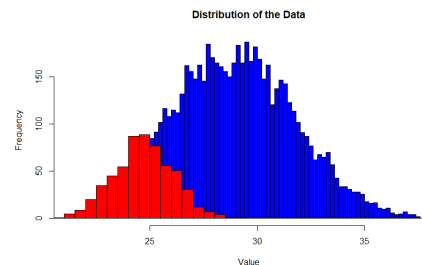
(c) New mean is the 10% quartile of the data distribution and SD is  $1/3$  of the original SD



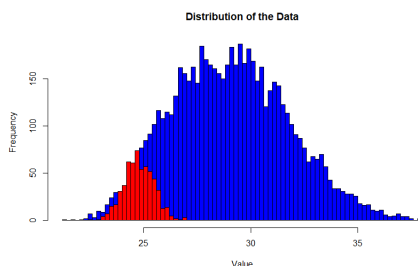
(d) New mean is the 15% quartile of the data distribution and SD is  $1/3$  of the original SD



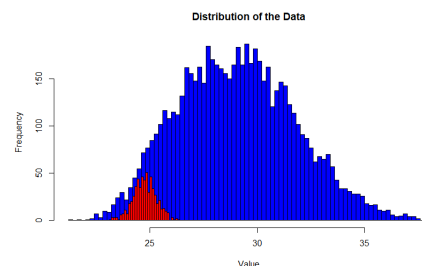
(e) New mean is the 5% quartile of the data distribution and SD is the original SD



(f) New mean is the 5% quartile of the data distribution and SD is  $1/2$  of the original SD



(g) New mean is the 5% quartile of the data distribution and SD is  $1/4$  of the original SD



(h) New mean is the 5% quartile of the data distribution and SD is  $1/5$  of the original SD

Figure 1: Variations of the new mean and the new standard deviation according to the one of the original distribution.

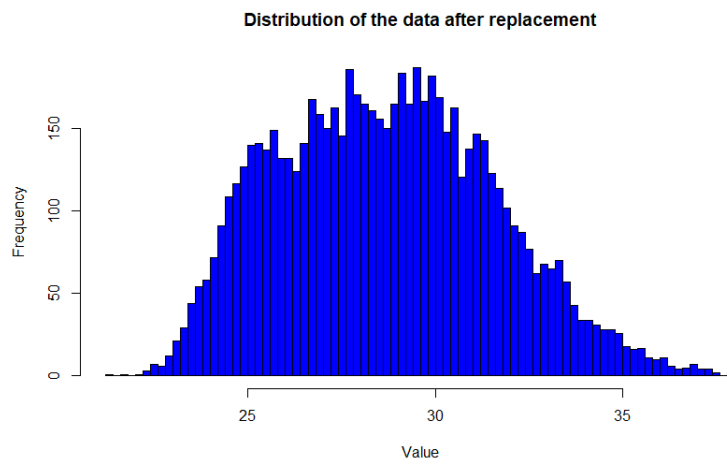


Figure 2: Data distribution after replacement of the NAs with the generated values.

### Exercise 7.2: DREAM challenge

(a) NOT IMPLEMENTED