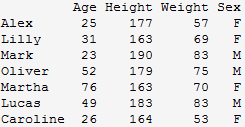
**Exercise 1**  
Create the following data frame, afterwards invert Sex for all individuals.  


# Exercise 1

Name <- c("Alex", "Lilly", "Mark", "Oliver", "Martha", "Lucas", "Caroline")

Age <- c(25, 31, 23, 52, 76, 49, 26)

Height <- c(177, 163, 190, 179, 163, 183, 164)

Weight <- c(57, 69, 83, 75, 70, 83, 53)

Sex <- as.factor(c("F", "F", "M", "M", "F", "M", "F"))

df <- data.frame (row.names = Name, Age, Height, Weight, Sex)

levels(df$Sex) <- c("M", "F")

df

## Age Height Weight Sex

## Alex 25 177 57 M

## Lilly 31 163 69 M

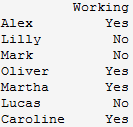
## Mark 23 190 83 F

## Oliver 52 179 75 F

## Martha 76 163 70 M

## Lucas 49 183 83 F

## Caroline 26 164 53 M

**Exercise 2**  
Create this data frame (make sure you import the variable Working as character and not factor).  
  
Add this data frame column-wise to the previous one.  
a) How many rows and columns does the new data frame have?  
b) What class of data is in each column?

# Exercise 2

Name <- c("Alex", "Lilly", "Mark", "Oliver", "Martha", "Lucas", "Caroline")

Working <- c("Yes", "No", "No", "Yes", "Yes", "No", "Yes")

dfa <- data.frame(row.names = Name, Working)

# a)

dfa <- cbind (df,dfa)

dim(dfa)

## [1] 7 5

# or:

nrow(dfa)

## [1] 7

ncol(dfa)

## [1] 5

# b)

sapply(dfa, class)

## Age Height Weight Sex Working

## "numeric" "numeric" "numeric" "factor" "factor"

str(dfa) # alternative solution

## 'data.frame': 7 obs. of 5 variables:

## $ Age : num 25 31 23 52 76 49 26

## $ Height : num 177 163 190 179 163 183 164

## $ Weight : num 57 69 83 75 70 83 53

## $ Sex : Factor w/ 2 levels "M","F": 1 1 2 2 1 2 1

## $ Working: Factor w/ 2 levels "No","Yes": 2 1 1 2 2 1 2

**Exercise 3**  
Check what class of data is the (built-in data set) state.center and convert it to data frame.

# Exercise 3

class (state.center)

## [1] "list"

df <- as.data.frame(state.center)

**Exercise 4**  
Create a simple data frame from 3 vectors. Order the entire data frame by the first column.

# Exercise 4

# Example vectors

v <- c(45:41, 30:33)

b <- LETTERS[rep(1:3, 3)]

n <- round(rnorm(9, 65, 5))

df <- data.frame(Age = v, Class = b, Grade = n)

df[with (df, order(Age)),]

## Age Class Grade

## 6 30 C 57

## 7 31 A 64

## 8 32 B 59

## 9 33 C 73

## 5 41 B 61

## 4 42 A 71

## 3 43 C 70

## 2 44 B 63

## 1 45 A 62

df[order(df$Age), ] # alternative solution

## Age Class Grade

## 6 30 C 57

## 7 31 A 64

## 8 32 B 59

## 9 33 C 73

## 5 41 B 61

## 4 42 A 71

## 3 43 C 70

## 2 44 B 63

## 1 45 A 62

**Exercise 5**  
Create a data frame from a matrix of your choice, change the row names so every row says id\_i (where i is the row number) and change the column names to variable\_i (where i is the column number). I.e., for column 1 it will say variable\_1, and for row 2 will say id\_2 and so on.

# Exercise 5

matr <- matrix(1:20, ncol = 5) # Example matrix

df <- as.data.frame(matr)

colnames(df) <- paste("variable\_", 1:ncol(df))

rownames(df) <- paste("id\_", 1:nrow(df))

df

## variable\_ 1 variable\_ 2 variable\_ 3 variable\_ 4 variable\_ 5

## id\_ 1 1 5 9 13 17

## id\_ 2 2 6 10 14 18

## id\_ 3 3 7 11 15 19

## id\_ 4 4 8 12 16 20

**Exercise 6**  
For this exercise, we’ll use the (built-in) dataset VADeaths.

a) Make sure the object is a data frame, if not change it to a data frame.  
b) Create a new variable, named Total, which is the sum of each row.  
c) Change the order of the columns so total is the first variable.

# Exercise 6

#a)

class(VADeaths)

## [1] "matrix"

df <- as.data.frame(VADeaths)

#b)

df$Total <- df[, 1] + df[, 2] + df[, 3] + df[, 4]

df$Total <- rowSums(df[1:4]) # alternative solution

#c)

df <- df[, c(5, 1:4)]

**Exercise 7**  
For this exercise we’ll use the (built-in) dataset state.x77.

a) Make sure the object is a data frame, if not change it to a data frame.  
b) Find out how many states have an income of less than 4300.  
c) Find out which is the state with the highest income.

# Exercise 7

#a)

class (state.x77)

## [1] "matrix"

df <- as.data.frame(state.x77)

#b)

nrow(subset(df, df$Income < 4300))

## [1] 20

#c)

row.names(df)[(which(max(df$Income) == df$Income))]

## [1] "Alaska"

**Exercise 8**  
With the dataset swiss, create a data frame of only the rows 1, 2, 3, 10, 11, 12 and 13, and only the variables Examination, Education and Infant.Mortality.  
a) The infant mortality of Sarine is wrong, it should be a NA, change it.  
b) Create a row that will be the total sum of the column, name it Total.  
c) Create a new variable that will be the proportion of Examination (Examination / Total)

# Exercise 8

df <- swiss[c(1:3, 10:13), c("Examination", "Education", "Infant.Mortality")]

#a)

df[4,3] <- NA

#b)

df["Total",] <- c(sum(df$Examination), sum(df$Education), sum(df$Infant.Mortality, na.rm = TRUE))

#c)

df$proportion <- round(df$Examination / df["Total", "Examination"], 3)

**Exercise 9**  
Create a data frame with the datasets state.abb, state.area, state.division, state.name, state.region. The row names should be the names of the states.

a) Rename the column names so only the first 3 letters after the full stop appear (e.g. States.abb will be abb).

# Exercise 9

df <- data.frame(state.abb, state.area, state.division, state.region, row.names = state.name)

#a)

names(df) <- substr(names(df), 7, 9)

**Exercise 10**  
Add the previous data frame column-wise to state.x77  
a) Remove the variable div.  
b) Also remove the variables Life Exp, HS Grad, Frost, abb, and are.  
c) Add a variable to the data frame which should categorize the level of illiteracy:  
[0,1) is low, [1,2) is some, [2, inf) is high.  
d) Find out which state from the west, with low illiteracy, has the highest income, and what that income is.

# Exercise 10

dfa <- cbind(state.x77, df)

#a)

dfa$div <- NULL

#b)

dfa <- subset(dfa, ,-c(4, 6, 7, 9, 10))

# c)

dfa$illi <- ifelse(dfa$Illiteracy < 1,

"Low Illiteracy",

ifelse(dfa$Illiteracy >= 1 & dfa$Illiteracy < 2,

"Some Illiteracy",

"High Illiteracy"))

# Or:

dfa$illi <- cut(dfa$Illiteracy,

c(0, 1, 2, 3),

include.lowest = TRUE,

right = FALSE,

labels = c("Low Illiteracy", "Some Illiteracy", "High Illliteracy"))

# d)

sub <- subset(dfa, illi == "Low Illiteracy" & reg == "West")

max <- max(sub$Income)

stat <- row.names(sub)[which (sub$Income == max)]

cat("Highest income from the West is", max , "the state where it's from is", stat, "\n")

## Highest income from the West is 5149 the state where it's from is Nevada

3.1 Factors

**3.1.1** If x = c(1, 2, 3, 3, 5, 3, 2, 4, NA), what are the levels of factor(x)?

x = c(1, 2, 3, 3, 5, 3, 2, 4, NA)

xf <- factor(x)

xf # Levels are 1,2,3,4,5

## [1] 1 2 3 3 5 3 2 4 <NA>

## Levels: 1 2 3 4 5

**3.1.2** Let x <- c(11, 22, 47, 47, 11, 47, 11). If an R expression factor(x, levels=c(11, 22, 47), ordered=TRUE) is executed, what will be the 4th element in the output?

x <- c(11, 22, 47, 47, 11, 47, 11)

factor(x, levels=c(11, 22, 47), ordered=TRUE) # The levels are 11, 22 and 47 and 4th element is 47.

## [1] 11 22 47 47 11 47 11

## Levels: 11 < 22 < 47

**3.1.3** If z <- c(“p”, “a” , “g”, “t”, “b”), then which of the following R expressions will replace the third element in z with “b”.

z <- c("p", "a" , "g", "t", "b")

factor(z) # as we can see, "g" is the 3rd element and also the 3rd factor. 3rd expression will replace 3rd element with "b" obviously.

## [1] p a g t b

## Levels: a b g p t

z[3] <- "b"

**3.1.4** If z <- factor(c(“p”, “q”, “p”, “r”, “q”)) and levels of z are “p”, “q” ,“r”, write an R expression that will change the level “p” to “w” so that z is equal to: “w”, “q” , “w”, “r” , “q”.

z <- factor(c("p", "q", "p", "r", "q")) # now we have the z with p,q,r factors.

levels(z)[1] <- "w"

z

## [1] w q w r q

## Levels: w q r

**3.1.5** If:  
s1 <- factor(sample(letters, size=5, replace=TRUE)) and  
s2 <- factor(sample(letters, size=5, replace=TRUE)),  
write an R expression that will concatenate s1 and s2 in a single factor with 10 elements.

s1 <- factor(sample(letters, size=5, replace=TRUE))

s2 <- factor(sample(letters, size=5, replace=TRUE))

factor(c(levels(s1)[s1], levels(s2)[s2])) # if your random samples have similar letters, note that it affects the number of levels.

## [1] g q o v j y q k t v

## Levels: g j k o q t v y

**3.1.8** Consider the factor responses <- factor(c(“Agree”, “Agree”, “Strongly Agree”, “Disagree”, “Agree”)), with the following output:

#[1] Agree Agree Strongly Agree Disagree Agree

#Levels: Agree Disagree Strongly Agree

responses <- factor(c("Agree", "Agree", "Strongly Agree", "Disagree", "Agree"))

Later it was found that new a level “Strongly Disagree” exists. Write an R expression that will include “strongly disagree” as new level attribute of the factor and returns the following output:

#[1] Agree Agree Strongly Agree Disagree Agree

#Levels: Strongly Agree Agree Disagree Strongly Disagree

factor(responses, levels=c("Strongly Agree", "Agree", "Disagree", "Strongly Disagree")) # added the last one.

## [1] Agree Agree Strongly Agree Disagree

## [5] Agree

## Levels: Strongly Agree Agree Disagree Strongly Disagree

**3.1.9** Let x <- data.frame(q=c(2, 4, 6), p=c(“a”, “b”, “c”)). Write an R statement that will replace levels a, b, c with labels “fertiliser1”, “fertliser2”, “fertiliser3”.

x <- data.frame(q=c(2, 4, 6), p=c("a", "b", "c"))

x$p <- factor(x$p, levels=c("a", "b", "c"), labels=c("fertiliser1", "fertiliser2", "fertiliser3"))

levels(x$p)

## [1] "fertiliser1" "fertiliser2" "fertiliser3"

**3.1.10** If x <- factor(c(“high”, “low”, “medium”, “high”, “high”, “low”, “medium”)), write an R expression that will provide unique numeric values for various levels of x with the following output:

# levels value

#1 high 1

#2 low 2

#3 medium 3

x <- factor(c("high", "low", "medium", "high", "high", "low", "medium"))

unique(x) # this expression gives the unique values in x

## [1] high low medium

## Levels: high low medium

data.frame(levels = unique(x), value = as.numeric(unique(x)))

## levels value

## 1 high 1

## 2 low 2

## 3 medium 3