IS441: Quality Assurance in Information Systems
Information Systems Department
Faculty of Computers & Information
Assiut University (Sheet 2)

c=rep(c("yes","control"),50)

1. Define the variable treatment as a vector of length 100 with elements ("yes","control","yes","control",: : :,"yes","control"):

c=rep(c("yes","yes","no","no","no"),20)

2. Define the variable smoker as a vector of length 100 with elements ("yes","yes","no","no","no",: : :,"yes","yes","no","no","no")

smoker=rep(c("yes","control"),50) treatment=rep(c("yes","yes","no","no","no"),20)

3. We assume the vectors smoker and treatment of Exercise 1 and 2 to be known. Define the vector

   lifespan <- abs(round(100*sin(1:100)))

   Suppose we have 100 individuals of which we have measured some quantity and stored in the vector lifespan. You may think of lifespan as the life span of the individuals but keep in mind that the data is self-generated and not meaningful. The vector smoker tells us which individual is smoker and which is non-smoker. Now we wish to study the measured quantity of all smokers.

   x=lifecpan[(smoke=="yes")] 40

   Define a new vector x which consists of all elements of lifespan at whose index in smoker is the element "yes". What is the maximum of lifespan over all smokers? y=lifecpan[(lifecpan%%2==0)&(lifecpan>=16)] length(y)=46

   - Define a new vector y of all even elements in lifespan which are greater than or equal to 16. What is the minimum of all these elements? y=lifecpan[(smoker=="yes")&(treatment=="yes")] 20
   - Half of the individuals got a certain treatment. Produce a new vector consisting of the lifespans of all individuals which are smokers and got the treatment. yy=lifecpan[(smoker=="no")|(treatment=="yes")] 40
   - Produce a new vector of the lifespans of individuals which are nonsmokers or got the treatment ('or' is not exclusive).

lapply(list1, summary)

list1=list(a,b,c)

Lists:

a=rnorm(10)

b=rexp(10)

c=c("yes","no","yes","no","yes","no")
c=as.factor(c)

4. Create a list with three elements. The first is a randomly generated set of numbers with a normal distribution. The second is a randomly generated set of numbers with an exponential distribution. The last is a set of factors. A summary is then performed on each element in the list.

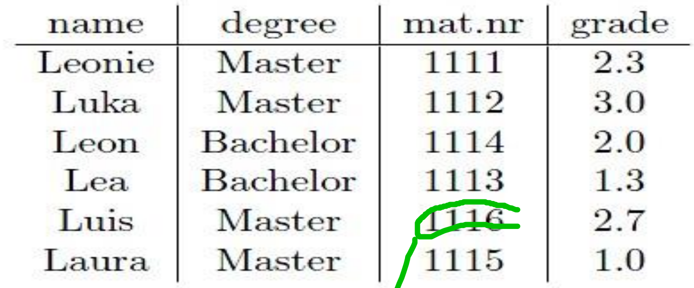
list2=list(a,b) sapply(list2, sum) tapply(a,b,sum)

5. Give an example of the **sapply** and the **tapply** functions. What do both of them apply?

```
student=data.frame(
 name=c("leonie","luka","leon","lea","luis","laura"),
 degree=c("master","master","bachelor","bachelor","master","master"),
 mat.nr=c(1111,1112,1114,1113,1116,1115),
 grade=c(2.3,3.0,2.0,1.3,2.7,1.0))
```

Data frames

6. are the typical R representation of data sets. Here we create a data frame \by hand" to become familiar with data frames. Use the command data.frame() to create a data frame students with the following entries:



name	degree	mat.nr	grade
Leonie	Master	1111	2.3
Luka	Master	1112	3.0
Leon	Bachelor	1114	2.0
Lea	Bachelor	1113	1.3
Luis	Master	~~1116~~	2.7
Laura	Master	1115	1.0

summary(student)

str(student)

student$name

- Get an overview of students with the commands names(), str() and summary().

student[5,3,1]

- Which command returns the fifth element of the vector 'mat.nr'?
- Check existence of the variable degree by entering it into the R command line. Now copy students into the search path with the command attach(). Check again whether degree is a known variable.
- Calculate the average grade of all students with a master degree
- Define a new data frame named 'ma.students' which consists of all students with degree Master (without using the command data.frame()). As all students in ma.students have degree Master the variable degree is not needed in ma.students.
- Write the data frame students into the file 'studentsfilele.txt'. Then read the data frame from this file into the new variable students2. If you used the right command, then students and students2 are identical. Check this using the command all().
- We wish to change 'degree' into 'deg' to save typing work. Use the command names() to this change. You might need to consult the help page ? names to find out how to do this.

7. Reading and Writing Frames

- Download the files 'olympic.txt', 'olympic0.dat', 'olympic1.txt', 'olympic2.txt' and 'olympic3.csv' from the course web page.
- Read the file 'olympic.txt' into a data frame.
- Read the file 'olympic0.dat' into a data frame. In that data frame, replace the first column by a column containing the respective years (integers between 1896 and 1992) and denote this column as \year". Then write this modified data frame to the file 'olympic1new.txt'.
- Read the file 'olympic1.txt' into a data frame. Produce a file 'olympicHighJump.txt' consisting only of the columns \Since1900" and \HighJump".
- Read the file 'olympic2.txt' into a data frame. Produce a file 'olympic2new.txt' containing a header with appropriate names (of your choice).
- Read the file 'olympic3.csv' into a data frame. Using this data frame, caculate the mean value of all long jump records between 1896 and 1972.

8. Merging data frames:

Download the files 'studentsfile.txt', 'studentgrades1.csv' and 'studentgrades2.txt' from the web page and read the data into R. Merge the data frames together and write the resulting data frame to the file 'studentgradesall.txt'.