



Packages & Libraries

Packages

- R packages are a collection of R functions, compiled code and sample data.
- They are stored under a directory called "**library**" in the R environment.
- By default, R installs a set of packages during installation. More packages are added later, when they are needed for some specific purpose.
- When we start the R console, only the default packages are available by default. Other packages which are already installed have to be loaded explicitly to be used by the R program that is going to use them.
- All the packages available in R language are listed at [R Packages](#).

Packages

Command Name	Description
<code>.libPaths()</code>	To Check Available R Package
<code>library()</code>	To load the Packages use the command / Get the list of all the packages installed. Example: <code>>library(MASS)</code>
<code>data()</code>	It will list all the datasets in loaded packages. Example: <code>>data()</code>
<code>search()</code>	Get all packages currently loaded in the R environment. Example: <code>>search()</code>
<code>installed.packages()</code>	It returns a detailed data frame about installed packages, not only containing names, but also licenses, versions, dependencies and more.

Packages

- **Examples of libraries that come as a part of base package in R.:**

- **MASS** : package associated with venables and Ripley's book
- **Mgcv** : Mixed GAM Computation Vehicle with GCV/AIC/REML Smoothness Estimation.

Contents of Libraries:

- It is easy to use the help function to discover the contents of library packages.
- Here is how find out about the contents of the **spatial library** :
library(help=spatial)

Packages

•Installing Packages and Libraries:

- The base R package contains programs for basic operations.
- It does not contain some of the libraries necessary for advanced statistical work.
- Specific requirements are met by special packages.
- They are downloaded and their downloading is very simple.
- To install any packages,
 - Run the R Program,
 - Then on the command line, use the **install. Packages** function to download the libraries we want.

Packages

•Installing Packages and Libraries:

Example:

- The package meta contains the statistical tolls for meta analysis.
- The package Agreement contains statistical tolls for measuring agreement.
- The packages meta or agreement can be installed by
 - `install.packages("meta")`
 - `install.packages("agreement")`

Exp. 5: Demonstrate Data Frames in R

- The commands `c`, `cbind`, `vector` and `matrix` functions combine data.
- Another option is the **data frame**.
- In a data frame, we can combine **variables of equal length, with each row in the data frame containing observations** on the same unit.
- Hence, It is similar to the `matrix` or `cbind` functions.
- **Advantage** is that one can make changes to the data without affecting the original data

Exp. 5: Demonstrate Data Frames in R

- One can also combine numerical variables, character strings as well as factors in data frame.
- **For example, `cbind` and `matrix` functions can not be used** to combine different types of data.
- Data frames are special types of objects in R designed for data sets.
- The data frame format is similar to a spreadsheet, where **columns contain variables and observations are contained in rows.**

Exp. 5: Demonstrate Data Frames in R

- Data frames contain complete data sets that are mostly created with other programs (spreadsheet-files, software SPSS-files, Excel-files etc.).
- **Variables** in a data frame **may be numeric (numbers) or categorical (characters or factors).**

Exp. 5: Demonstrate Data Frames in R

- Data Frames are data displayed in a format as a table.
- Data Frames can have different types of data inside it. While the first column can be **character**, the second and third can be **numeric or logical**. However, each column should have the same type of data.

E_name	E_age	E_salary
A	30	20000
B	40	30000
C	50	40000
D	60	50000
E	70	60000

Exp. 5: Demonstrate Data Frames in R

Package “MASS” describes functions and datasets to support venables and Ripley, “Modern Applied Statistics with S” (4th edition 2002), Springer.

Example :

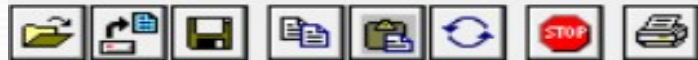
An example data frame painters is available in the library. MASS:

```
>library(MASS)      # load the data  
>painters           #data set
```

Here, the names of the painters serve as row identification, i.e., every row is assign to the name of the corresponding the painter.

R RGui (64-bit)

File Edit View Misc Packages Windows Help



R R Console

```
> library(MASS)
> painters
```

	Composition	Drawing	Colour	Expression	School
Da Udine	10	8	16	3	A
Da Vinci	15	16	4	14	A
Del Piombo	8	13	16	7	A
Del Sarto	12	16	9	8	A
Fr. Penni	0	15	8	0	A
Guilio Romano	15	16	4	14	A
Michelangelo	8	17	4	8	A
Perino del Vaga	15	16	7	6	A
Perugino	4	12	10	4	A
Raphael	17	18	12	18	A
F. Zucarro	10	13	8	8	B
Fr. Salviata	13	15	8	8	B
Parmigiano	10	15	6	6	B
Primaticcio	15	14	7	10	B
T. Zucarro	13	14	10	9	B
Volterra	12	15	5	8	B
Barocci	14	15	6	10	C
Cortona	16	14	12	6	C
Josepin	10	10	6	2	C
L. Jordaens	13	12	9	6	C
Testa	11	15	0	6	C
Vanius	15	15	12	13	C
Bassano	6	8	17	0	D
Ballini	4	6	14	0	D

Exp. 5: Demonstrate Data Frames in R

- **Functions on Data frames:**

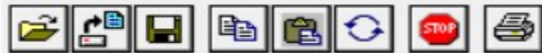
The `rownames()` and `colnames()` functions in R are used to obtain or set the names of the row and column of a matrix-like object, respectively.

`>rownames(painters)`

`>colnames(painters)`

RGui (64-bit) - [R Console]

File Edit View Misc Packages Windows Help



```
> rownames(painters)
 [1] "Da Udine"      "Da Vinci"      "Del Piombo"    "Del Sarto"
[11] "F. Zucarro"    "Fr. Salviata"  "Parmigiano"    "Primaticcio"
[21] "Testa"        "Vanius"        "Bassano"       "Bellini"
[31] "Titian"        "Veronese"      "Albani"        "Caravaggio"
[41] "Holbein"       "Pourbus"       "Van Leyden"    "Diepenbeck"
[51] "Bourdon"       "Le Brun"       "Le Suer"       "Poussin"
> colnames(painters)
[1] "Composition" "Drawing"      "Colour"       "Expression"   "School"
> |
```

Exp. 5: Demonstrate Data Frames in R

- **Functions on Data frames:**

is.numeric(): returns a logical value, TRUE or FALSE, indicating if the argument passed to it has a base type of the class double or integer and the values are regarded as numeric.

Syntax: name of function(**Name of data set \$Name of variable**)

Example: The data set painters contains four numerical variables (Composition, Drawing, colour and Expression), as well as one factor variable (school).

>is.numeric (painters\$School)

[1] FALSE

```
> is.numeric(painters$School)
[1] FALSE
> |
```

>is.numeric (painters\$composition)

[1] TRUE

```
> is.numeric(painters$Composition)
[1] TRUE
> |
```

NOTE: The \$ operator is used to extract or subset a specific part of a data object in R.

Exp. 5: Demonstrate Data Frames in R

- **Functions on Data frames:**

is.factor(): It is used to return a logical value, TRUE or FALSE, indicating whether an argument passed to it is of type factor or not..

Example: The data set painters contains four numerical variables (Composition, Drawing, colour and Expression), as well as one factor variable (school).

```
>is.factor (painters$School)  
[1] TRUE
```

```
>is.numeric (painters$Drawing)  
[1] FALSE
```

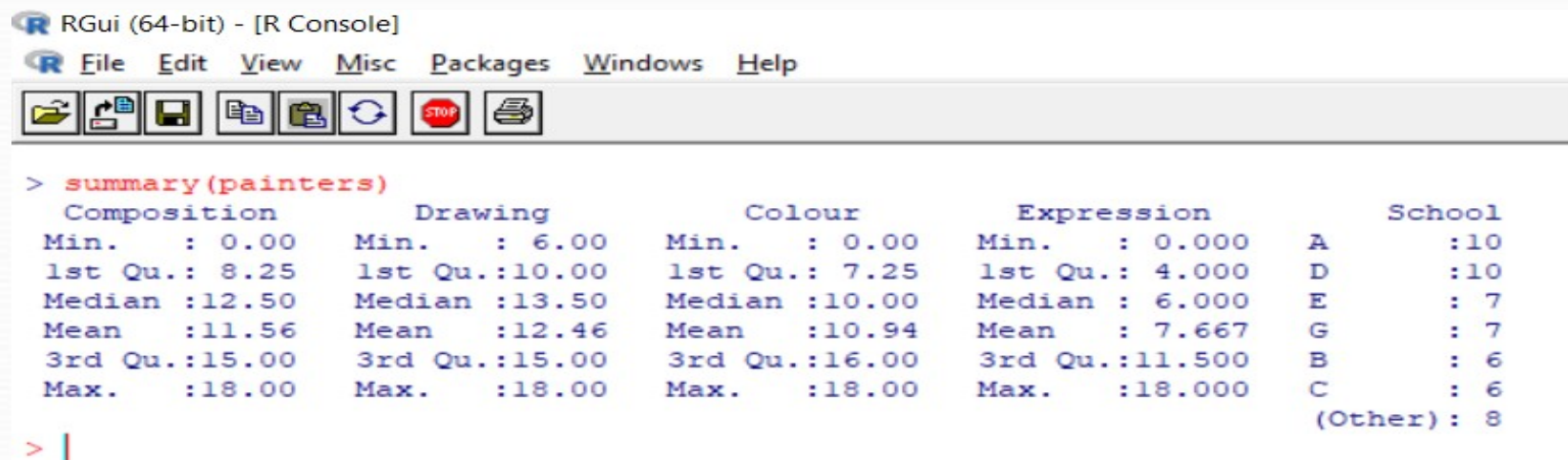

Exp. 5: Demonstrate Data Frames in R

- Functions on Data frames:**

summary(): Using this function, we can get a quick overview of descriptive measures for each variable.

Example:

`>summary(painters)`



```
> summary(painters)
  Composition      Drawing      Colour      Expression      School
Min.   : 0.00   Min.   : 6.00   Min.   : 0.00   Min.   : 0.000   A       :10
1st Qu.: 8.25   1st Qu.:10.00   1st Qu.: 7.25   1st Qu.: 4.000   D       :10
Median :12.50   Median :13.50   Median :10.00   Median : 6.000   E       : 7
Mean   :11.56   Mean   :12.46   Mean   :10.94   Mean   : 7.667   G       : 7
3rd Qu.:15.00   3rd Qu.:15.00   3rd Qu.:16.00   3rd Qu.:11.500   B       : 6
Max.   :18.00   Max.   :18.00   Max.   :18.00   Max.   :18.000   C       : 6
                                           (Other): 8
> |
```

The categories F and H, each present 4 times in the variable “School”, are summed under the category other as 8 with the corresponding frequency. I.E., only the 6 most frequent values are displayed.

Exp. 5: Demonstrate Data Frames in R

Test if we are dealing with a Data Frame or not:

In order to check whether a given dataset corresponds to the structure of the data frame, use the following command:

```
>is.data.frame(painters)  
[1] TRUE
```



```
RGui (64-bit) - [R Console]  
File Edit View Misc Packages Windows Help  
[Icons]  
  
> is.data.frame(painters)  
[1] TRUE  
> |
```

Exp. 5: Demonstrate Data Frames in R

Test if we are dealing with a Data Frame or not:

In order to check whether a given dataset corresponds to the structure of the data frame, use the following command:

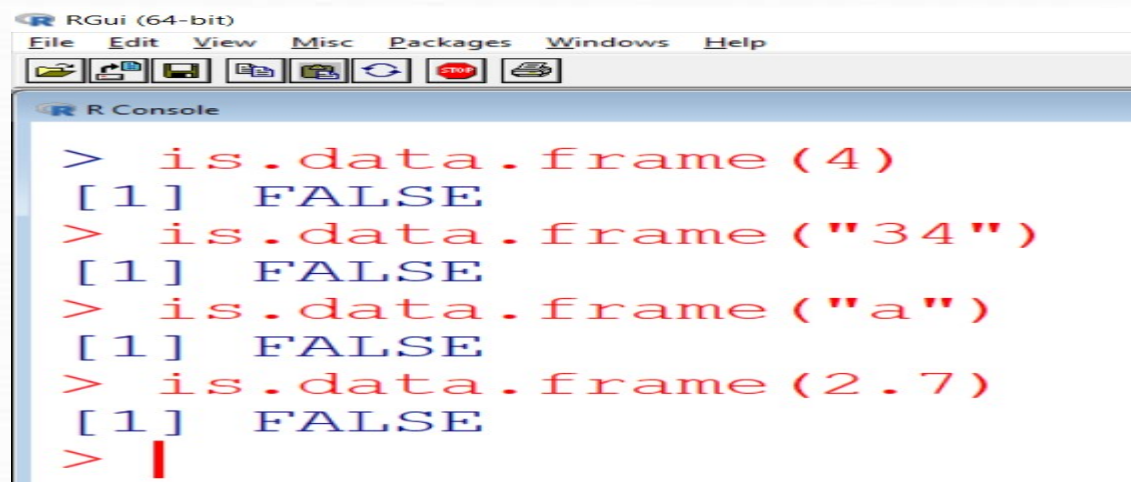
over different data types

```
>is.data.frame(4)
```

```
>is.data.frame("34")
```

```
>is.data.frame("a")
```

```
>is.data.frame(2.7)
```



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]
R Console
> is.data.frame(4)
[1] FALSE
> is.data.frame("34")
[1] FALSE
> is.data.frame("a")
[1] FALSE
> is.data.frame(2.7)
[1] FALSE
> |
```

Exp. 5: Demonstrate Data Frames in R

Creating a Data Frame:

Use the **data.frame()** function to create a data frame.

```
mydata <- data.frame(col1, col2, col3,...)
```

where col1, col2, col3, ... are column vectors of any type (such as character, numeric, or logical).

Exp. 5: Demonstrate Data Frames in R

Example:

```
>x=1:16
```

```
#vector
```

```
>y=matrix(x,nrow=4,ncol=4)
```

```
#4x4 matrix
```

```
>z=letters[1:16]
```

```
# lower case Alphabets
```

```
> x
```

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
```

```
> y
```

```
  [,1] [,2] [,3] [,4]  
[1,]  1  5  9 13  
[2,]  2  6 10 14  
[3,]  3  7 11 15  
[4,]  4  8 12 16
```

```
>z
```

```
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p"
```

Exp. 5: Demonstrate Data Frames in R

Example (continued):

```
df=data.frame(x,y,z)
```

```
> df
```

	x	X1	X2	X3	X4	z
1	1	1	5	9	13	a
2	2	2	6	10	14	b
3	3	3	7	11	15	c
4	4	4	8	12	16	d
5	5	1	5	9	13	e
6	6	2	6	10	14	f
7	7	3	7	11	15	g
8	8	4	8	12	16	h
9	9	1	5	9	13	i
10	10	2	6	10	14	j
11	11	3	7	11	15	k
12	12	4	8	12	16	l
13	13	1	5	9	13	m
14	14	2	6	10	14	n
15	15	3	7	11	15	o
16	16	4	8	12	16	p

Exp. 5: Demonstrate Data Frames in R

str() function:

- Display information about the **structure** of the data frame .
- The result of **str** gives the dimension as well as the name and type of each variable.

Example 1:

```
>str(df)
```

```
'data.frame': 16 obs. of 6 variables:
```

```
$ x : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
$ X1: int  1 2 3 4 1 2 3 4 1 2 ...
```

```
$ X2: int  5 6 7 8 5 6 7 8 5 6 ...
```

```
$ X3: int  9 10 11 12 9 10 11 12 9 10 ...
```

```
$ X4: int 13 14 15 16 13 14 15 16 13 14 ...
```

```
$ z : chr  "a" "b" "c" "d" ...
```

Exp. 5: Demonstrate Data Frames in R

str() function:

- Display information about the **structure** of the data frame .
- The result of **str** gives the dimension as well as the name and type of each variable.

Example 1:

> str(painters)

'data.frame': 54 obs. of 5 variables:

\$ Composition: int 10 15 8 12 0 15 8 15 4 17 ...

\$ Drawing : int 8 16 13 16 15 16 17 16 12 18 ...

\$ Colour : int 16 4 16 9 8 4 4 7 10 12 ...

\$ Expression : int 3 14 7 8 0 14 8 6 4 18 ...

\$ School : Factor w/ 8 levels "A","B","C","D",...: 1 1 1 1 1 1 1 1 1 1 ...

Exp. 5: Demonstrate Data Frames in R

Extract a variable from data frame using \$:

- Variables can be extracted using the \$ operator followed by the name of the variable.

Syntax:

Name of data set \$Name of variable

Example: suppose we want to extract information on school from the data set painters.

```
> painters$School
```

```
[1] AAAAAAAAAAABBBBBBBCCCCCCDDDDDDDDDDDEEEEEEEFF  
FFGGGGGGGGHHHH
```

```
Levels: A B C D E F G H
```

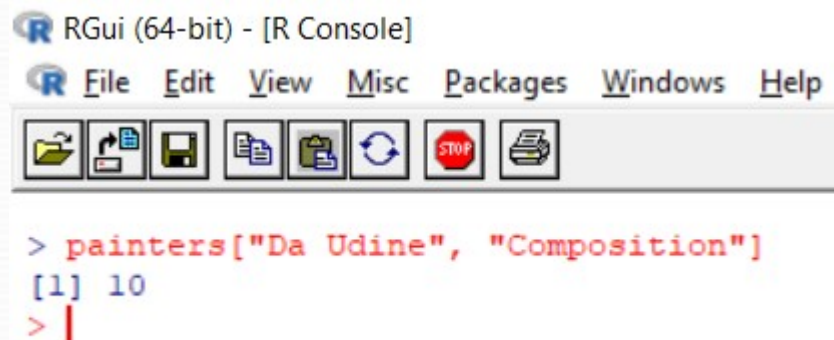

Exp. 5: Demonstrate Data Frames in R

Extract data from data frame [row, column]:

- The data from a data frame can be extracted using the matrix style [row, column] indexing.

Example: suppose we want to extract information on the first painter **Da Uline** on the variable **Composition** from the data set **painters**.

```
> painters["Da Uline", "Composition"]  
[1] 10
```



```
RGui (64-bit) - [R Console]  
File Edit View Misc Packages Windows Help  
[Icons]  
  
> painters["Da Uline", "Composition"]  
[1] 10  
> |
```

Exp. 5: Demonstrate Data Frames in R

Summary function:

Summary function for a categorical variable returns a detailed frequency table.

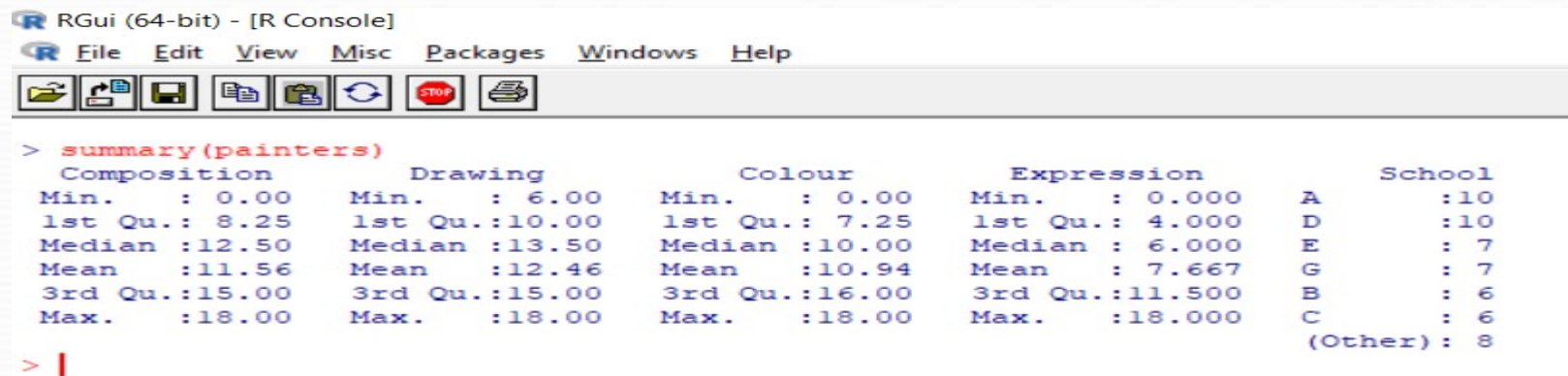
Example 1:

```
> summary painters$School
```

A	B	C	D	E	F	G	H
10	6	6	10	7	4	7	4

Example 2:

```
> summary(painters)
```



```
> summary(painters)
```

Composition		Drawing		Colour		Expression		School	
Min. :	0.00	Min. :	6.00	Min. :	0.00	Min. :	0.000	A	:10
1st Qu.:	8.25	1st Qu.:	10.00	1st Qu.:	7.25	1st Qu.:	4.000	D	:10
Median :	12.50	Median :	13.50	Median :	10.00	Median :	6.000	E	: 7
Mean :	11.56	Mean :	12.46	Mean :	10.94	Mean :	7.667	G	: 7
3rd Qu.:	15.00	3rd Qu.:	15.00	3rd Qu.:	16.00	3rd Qu.:	11.500	B	: 6
Max. :	18.00	Max. :	18.00	Max. :	18.00	Max. :	18.000	C	: 6
								(Other)	: 8

```
> |
```

Exp. 5: Demonstrate Data Frames in R

Plot and Graphics of the Data:

Examples:

```
>plot(painters$School) #factor variable
```



Exp. 5: Demonstrate Data Frames in R

attach() function:

- With a command **attach()** over the data frame, the variables can be referenced directly by name.
- It can address the names of the data frame directly, without the prefix dollar sign operator. **Example:** painters\$

•Example:

```
>attach(painters)
```

Variable names are

- Composition,
- Drawing,
- Colour,
- Expression,
- School,

Exp. 5: Demonstrate Data Frames in R

Examples:

>summary(School)

#character variable

```
R Console  
  
> summary(School)  
  A   B   C   D   E   F   G   H  
10   6   6  10   7   4   7   4  
> |
```

>summary(Composition)

#Numeric variable

```
R Console  
  
> summary(Composition)  
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   
 0.00   8.25   12.50   11.56   15.00   18.00   
> |
```

Exp. 5: Demonstrate Data Frames in R

detach() function:

It recovers the default setting and then we have to use painters\$ again.

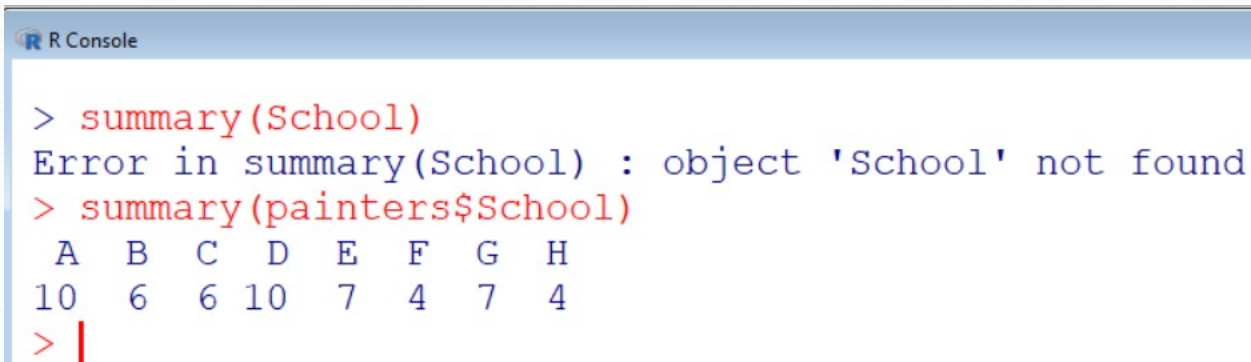
•Syntax:

>detach(Name of data set)

Example:

>detach(painters)

>summary(School)



```
R Console
> summary(School)
Error in summary(School) : object 'School' not found
> summary(painters$School)
  A  B  C  D  E  F  G  H
10  6  6 10  7  4  7  4
> |
```

Exp. 5: Demonstrate Data Frames in R

subset() function: subset of a data frame can be obtained by using this function.

Example:

`>subset(painters,School=='F')`

```
R Console
> subset(painters,School=='F')
      Composition Drawing Colour Expression School
Durer             8     10     10           8      F
Holbein            9     10     16          13      F
Pourbus            4     15      6           6      F
Van Leyden         8      6      6           4      F
> |
```


Exp. 5: Demonstrate Data Frames in R

Similar outcome can be also obtained from:

Example:

```
> painters[painters[["School"]]=='F',]
```

R Console

```
> painters[painters[["School"]]=='F',]
```

	Composition	Drawing	Colour	Expression	School
Durer	8	10	10	8	F
Holbein	9	10	16	13	F
Pourbus	4	15	6	6	F
Van Leyden	8	6	6	4	F

```
> |
```


Exp. 5: Demonstrate Data Frames in R

More calculations on subset() function:

Example:

```
>subset(painters,Composition<=6)
```

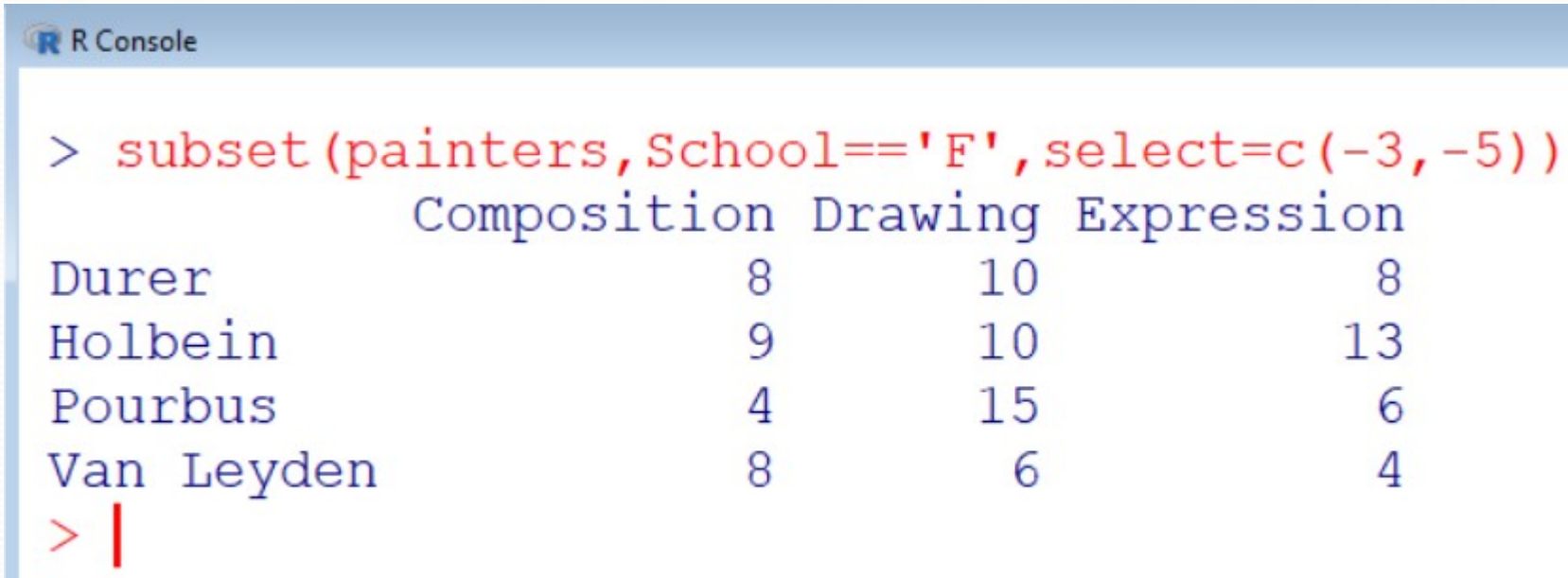
```
R Console
> subset(painters,Composition<=6)
      Composition Drawing Colour Expression School
Fr. Penni         0     15      8          0      A
Perugino          4     12     10          4      A
Bassano           6      8     17          0      D
Bellini           4      6     14          0      D
Murillo           6      8     15          4      D
Palma Vecchio     5      6     16          0      D
Caravaggio        6      6     16          0      E
Pourbus           4     15      6          6      F
> |
```

Exp. 5: Demonstrate Data Frames in R

Elimination of uninteresting Columns::

Example:

```
>subset painters, School=='F',select=c(-3,-5))
```



The screenshot shows an R console window with the following content:

```
> subset(painters, School=='F', select=c(-3, -5))
```

	Composition	Drawing	Expression
Durer	8	10	8
Holbein	9	10	13
Pourbus	4	15	6
Van Leyden	8	6	4

```
> |
```

The third and the fifth column (Colour and School) are not shown.

Exp. 5: Demonstrate Data Frames in R

Getting to know a Data frame:

Function Name	Description	Example
str()	The structure of the data frame can be seen by using function	>str(emp.data)
summary()	The statistical summary and nature of the data can be obtained by applying function	>summary(emp.data)
names()	This function returns the Column Names of a data frame	>names(emp.data)
rownames()	This function returns the Row Names of a data frame	>rownames(emp.data)
dim()	This function returns the Dimension of a data frame	>dim(emp.data)
nrow()	This function returns the No. of Rows of a data frame	>nrow(emp.data)
ncol()	This function returns the No. of Columns of a data frame	>ncol(emp.data)
head()	By default, the head() function returns the first 6 rows by default.	>head(emp.data) >head(emp.data,11)
tail()	By default, the tail() function returns the last 6 rows by default.	>tail(emp.data) >tail(emp.data,9)