

r-python-exercise1-ajinkyapdeshmukh

Ajinkya Deshmukh

Introduction to R

The current version of R used is listed below:

```
R.version
```

```
platform      _  
arch          x86_64-pc-linux-gnu  
os            linux-gnu  
system        x86_64, linux-gnu  
status  
major         4  
minor         3.1  
year          2023  
month         06  
day           16  
svn rev       84548  
language      R  
version.string R version 4.3.1 (2023-06-16)  
nickname      Beagle Scouts
```

Packages:

The Package which is used in Data Mining with R is “DMwR2”, this package is installed using below commands:

```
install.packages("DMwR2")
```

Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)

To see the contents of the packages we can use `help()`, as shown below:

```
help(package="DMwR2")
```

The packages are now installed in the system.

- There are two ways when we want to use a function in the installed packages, either of the two ways will work which are listed below:
1. Function used frequently: When we want to use the function frequently, we need to load it into the memory of the current session by using **library()** function.
 2. Function not used frequently: When we want to use the function once or twice, we can call the **function/dataset** by using **package::functionname**.

```
library(DMwR2)
```

Registered S3 method overwritten by 'quantmod':

```
method      from  
as.zoo.data.frame zoo
```

After using the above function, we can use any function or dataset which is provided in **DMwR2** by just referencing its name directly.

```
data("algae") #loading algae dataset  
algae
```

A tibble: 200 x 18

	season	size	speed	mxPH	mnO2	Cl	N03	NH4	oP04	P04	Chla	a1
	<fct>	<fct>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	winter	small	medium	8	9.8	60.8	6.24	578	105	170	50	0
2	spring	small	medium	8.35	8	57.8	1.29	370	429.	559.	1.3	1.4
3	autumn	small	medium	8.1	11.4	40.0	5.33	347.	126.	187.	15.6	3.3
4	spring	small	medium	8.07	4.8	77.4	2.30	98.2	61.2	139.	1.4	3.1
5	autumn	small	medium	8.06	9	55.4	10.4	234.	58.2	97.6	10.5	9.2
6	winter	small	high	8.25	13.1	65.8	9.25	430	18.2	56.7	28.4	15.1
7	summer	small	high	8.15	10.3	73.2	1.54	110	61.2	112.	3.2	2.4

```

 8 autumn small high      8.05  10.6  59.1  4.99  206.   44.7  77.4  6.9   18.2
 9 winter small medium  8.7    3.4  22.0  0.886 103.   36.3  71    5.54  25.4
10 winter small high    7.93   9.9   8     1.39   5.8  27.2  46.6  0.8   17
# i 190 more rows
# i 6 more variables: a2 <dbl>, a3 <dbl>, a4 <dbl>, a5 <dbl>, a6 <dbl>,
#   a7 <dbl>

```

```
manyNAs(algae) #finding rows with too many NAs in the algae dataset
```

```
[1] 62 199
```

library() without any parameters/arguments will provide the list of packages installed in different libraries on the system.

```
library()
```

Below command will show the packages loaded in the current session.

```
(.packages())
```

```
[1] "DMwR2"      "stats"      "graphics"   "grDevices"  "utils"      "datasets"
[7] "methods"    "base"
```

- When **library()** contains library of all installed packages. **library(packagename)** will check a package out.
- **.packages()** will show all checkout packages for the current session.
- If a package (e.g. **dbpylr**) is loaded in the current session by mistake, we can detach the package from the session using **detach**.

```
install.packages("dbpylr", repos = 'https://cloud.r-project.org') #assuming dbpylr is already installed
```

Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)

Warning: package 'dbplyr' is not available for this version of R

A version of this package for your version of R might be available elsewhere,
see the ideas at

<https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages>

```
#now trying to checkout dplyr, but we typed "dbplyr" accidentally
library(dbplyr)
(.packages())
```

```
[1] "dbplyr"      "DMwR2"      "stats"      "graphics"   "grDevices"  "utils"
[7] "datasets"   "methods"    "base"
```

```
#now realizing the mistake, we don't want this package to be active in the current session
#we can detach the package
```

```
detach("package:dbplyr", unload = TRUE)
(.packages())
```

```
[1] "DMwR2"      "stats"      "graphics"   "grDevices"  "utils"      "datasets"
[7] "methods"    "base"
```

```
library(dplyr) #load the correct library
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

filter, lag

The following objects are masked from 'package:base':

intersect, setdiff, setequal, union

Similarly, we can use another way to find what packages are installed in the system.

`installed.packages()`

	Package	LibPath
base64enc	"base64enc"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
bit	"bit"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
bit64	"bit64"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
blob	"blob"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
bslib	"bslib"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
cachem	"cachem"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
cli	"cli"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
clipr	"clipr"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
cpp11	"cpp11"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
crayon	"crayon"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
curl	"curl"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
DBI	"DBI"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
dbplyr	"dbplyr"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
digest	"digest"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
DMwR2	"DMwR2"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
dplyr	"dplyr"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
ellipsis	"ellipsis"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
evaluate	"evaluate"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
fansi	"fansi"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
fastmap	"fastmap"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
fontawesome	"fontawesome"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
fs	"fs"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
generics	"generics"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
gitcreds	"gitcreds"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
glue	"glue"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
highr	"highr"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
hms	"hms"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
htmltools	"htmltools"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
jquerylib	"jquerylib"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
jsonlite	"jsonlite"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
KernSmooth	"KernSmooth"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
knitr	"knitr"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
lifecycle	"lifecycle"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
magrittr	"magrittr"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
Matrix	"Matrix"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
memoise	"memoise"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
mgcv	"mgcv"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
mime	"mime"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"

nlme	"nlme"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
palmerpenguins	"palmerpenguins"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
pillar	"pillar"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
pkgconfig	"pkgconfig"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
prettyunits	"prettyunits"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
progress	"progress"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
purrr	"purrr"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
quantmod	"quantmod"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
R6	"R6"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
rappdirs	"rappdirs"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
readr	"readr"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
rlang	"rlang"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
rmarkdown	"rmarkdown"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
sass	"sass"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
spatial	"spatial"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
stringi	"stringi"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
stringr	"stringr"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
survival	"survival"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
tibble	"tibble"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
tidyr	"tidyr"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
tidyselect	"tidyselect"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
tinytex	"tinytex"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
TTR	"TTR"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
tzdb	"tzdb"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
utf8	"utf8"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
vctrs	"vctrs"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
vroom	"vroom"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
withr	"withr"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
xfun	"xfun"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
xts	"xts"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
yaml	"yaml"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
zoo	"zoo"	"/cloud/lib/x86_64-pc-linux-gnu-library/4.3"
base	"base"	"/opt/R/4.3.1/lib/R/library"
boot	"boot"	"/opt/R/4.3.1/lib/R/library"
class	"class"	"/opt/R/4.3.1/lib/R/library"
cluster	"cluster"	"/opt/R/4.3.1/lib/R/library"
codetools	"codetools"	"/opt/R/4.3.1/lib/R/library"
compiler	"compiler"	"/opt/R/4.3.1/lib/R/library"
datasets	"datasets"	"/opt/R/4.3.1/lib/R/library"
foreign	"foreign"	"/opt/R/4.3.1/lib/R/library"
graphics	"graphics"	"/opt/R/4.3.1/lib/R/library"
grDevices	"grDevices"	"/opt/R/4.3.1/lib/R/library"
grid	"grid"	"/opt/R/4.3.1/lib/R/library"

KernSmooth	"KernSmooth"	"/opt/R/4.3.1/lib/R/library"
lattice	"lattice"	"/opt/R/4.3.1/lib/R/library"
MASS	"MASS"	"/opt/R/4.3.1/lib/R/library"
Matrix	"Matrix"	"/opt/R/4.3.1/lib/R/library"
methods	"methods"	"/opt/R/4.3.1/lib/R/library"
mgcv	"mgcv"	"/opt/R/4.3.1/lib/R/library"
nlme	"nlme"	"/opt/R/4.3.1/lib/R/library"
nnet	"nnet"	"/opt/R/4.3.1/lib/R/library"
parallel	"parallel"	"/opt/R/4.3.1/lib/R/library"
rpart	"rpart"	"/opt/R/4.3.1/lib/R/library"
spatial	"spatial"	"/opt/R/4.3.1/lib/R/library"
splines	"splines"	"/opt/R/4.3.1/lib/R/library"
stats	"stats"	"/opt/R/4.3.1/lib/R/library"
stats4	"stats4"	"/opt/R/4.3.1/lib/R/library"
survival	"survival"	"/opt/R/4.3.1/lib/R/library"
tcltk	"tcltk"	"/opt/R/4.3.1/lib/R/library"
tools	"tools"	"/opt/R/4.3.1/lib/R/library"
utils	"utils"	"/opt/R/4.3.1/lib/R/library"
	Version	Priority
base64enc	"0.1-3"	NA
bit	"4.0.5"	NA
bit64	"4.0.5"	NA
blob	"1.2.4"	NA
bslib	"0.5.1"	NA
cachem	"1.0.8"	NA
cli	"3.6.1"	NA
clipr	"0.8.0"	NA
cpp11	"0.4.6"	NA
crayon	"1.5.2"	NA
curl	"5.0.2"	NA
DBI	"1.1.3"	NA
dbplyr	"2.3.3"	NA
digest	"0.6.33"	NA
DMwR2	"0.0.2"	NA
dplyr	"1.1.3"	NA
ellipsis	"0.3.2"	NA
evaluate	"0.21"	NA
fansi	"1.0.4"	NA
fastmap	"1.1.1"	NA
fontawesome	"0.5.2"	NA
fs	"1.6.3"	NA
generics	"0.1.3"	NA
gitcreds	"0.1.2"	NA

glue	"1.6.2"	NA
highr	"0.10"	NA
hms	"1.1.3"	NA
htmltools	"0.5.6"	NA
jquerylib	"0.1.4"	NA
jsonlite	"1.8.7"	NA
KernSmooth	"2.23-22"	"recommended"
knitr	"1.43"	NA
lifecycle	"1.0.3"	NA
magrittr	"2.0.3"	NA
Matrix	"1.6-1"	"recommended"
memoise	"2.0.1"	NA
mgcv	"1.9-0"	"recommended"
mime	"0.12"	NA
nlme	"3.1-163"	"recommended"
palmerpenguins	"0.1.1"	NA
pillar	"1.9.0"	NA
pkgconfig	"2.0.3"	NA
prettyunits	"1.1.1"	NA
progress	"1.2.2"	NA
purrr	"1.0.2"	NA
quantmod	"0.4.25"	NA
R6	"2.5.1"	NA
rappdirs	"0.3.3"	NA
readr	"2.1.4"	NA
rlang	"1.1.1"	NA
rmarkdown	"2.24"	NA
sass	"0.4.7"	NA
spatial	"7.3-17"	"recommended"
stringi	"1.7.12"	NA
stringr	"1.5.0"	NA
survival	"3.5-7"	"recommended"
tibble	"3.2.1"	NA
tidyr	"1.3.0"	NA
tidyselect	"1.2.0"	NA
tinytex	"0.46"	NA
TTR	"0.24.3"	NA
tzdb	"0.4.0"	NA
utf8	"1.2.3"	NA
vctrs	"0.6.3"	NA
vroom	"1.6.3"	NA
withr	"2.5.0"	NA
xfun	"0.40"	NA

xts	"0.13.1"	NA
yaml	"2.3.7"	NA
zoo	"1.8-12"	NA
base	"4.3.1"	"base"
boot	"1.3-28.1"	"recommended"
class	"7.3-22"	"recommended"
cluster	"2.1.4"	"recommended"
codetools	"0.2-19"	"recommended"
compiler	"4.3.1"	"base"
datasets	"4.3.1"	"base"
foreign	"0.8-84"	"recommended"
graphics	"4.3.1"	"base"
grDevices	"4.3.1"	"base"
grid	"4.3.1"	"base"
KernSmooth	"2.23-21"	"recommended"
lattice	"0.21-8"	"recommended"
MASS	"7.3-60"	"recommended"
Matrix	"1.5-4.1"	"recommended"
methods	"4.3.1"	"base"
mgcv	"1.8-42"	"recommended"
nlme	"3.1-162"	"recommended"
nnet	"7.3-19"	"recommended"
parallel	"4.3.1"	"base"
rpart	"4.1.19"	"recommended"
spatial	"7.3-16"	"recommended"
splines	"4.3.1"	"base"
stats	"4.3.1"	"base"
stats4	"4.3.1"	"base"
survival	"3.5-5"	"recommended"
tcltk	"4.3.1"	"base"
tools	"4.3.1"	"base"
utils	"4.3.1"	"base"
Depends		
base64enc	"R (>= 2.9.0)"	
bit	"R (>= 2.9.2)"	
bit64	"R (>= 3.0.1), bit (>= 4.0.0), utils, methods, stats"	
blob	NA	
bslib	"R (>= 2.10)"	
cachem	NA	
cli	"R (>= 3.4)"	
clipr	NA	
cpp11	"R (>= 3.5.0)"	
crayon	NA	

curl	"R (>= 3.0.0)"
DBI	"methods, R (>= 3.0.0)"
dbplyr	"R (>= 3.1)"
digest	"R (>= 3.3.0)"
DMwR2	"R(>= 3.0), methods"
dplyr	"R (>= 3.5.0)"
ellipsis	"R (>= 3.2)"
evaluate	"R (>= 3.0.2)"
fansi	"R (>= 3.1.0)"
fastmap	NA
fontawesome	"R (>= 3.3.0)"
fs	"R (>= 3.4)"
generics	"R (>= 3.2)"
gitcreds	"R (>= 3.4)"
glue	"R (>= 3.4)"
highr	"R (>= 3.3.0)"
hms	NA
htmltools	"R (>= 2.14.1)"
jquerylib	NA
jsonlite	"methods"
KernSmooth	"R (>= 2.5.0), stats"
knitr	"R (>= 3.3.0)"
lifecycle	"R (>= 3.4)"
magrittr	"R (>= 3.4.0)"
Matrix	"R (>= 3.5.0), methods"
memoise	NA
mgcv	"R (>= 3.6.0), nlme (>= 3.1-64)"
mime	NA
nlme	"R (>= 3.5.0)"
palmerpenguins	"R (>= 2.10)"
pillar	NA
pkgconfig	NA
prettyunits	NA
progress	NA
purrr	"R (>= 3.5.0)"
quantmod	"R (>= 3.2.0), xts(>= 0.9-0), zoo, TTR(>= 0.2), methods"
R6	"R (>= 3.0)"
rappdirs	"R (>= 3.2)"
readr	"R (>= 3.5)"
rlang	"R (>= 3.5.0)"
rmarkdown	"R (>= 3.0)"
sass	NA
spatial	"R (>= 3.0.0), graphics, stats, utils"

stringi	"R (>= 3.1)"
stringr	"R (>= 3.3)"
survival	"R (>= 3.5.0)"
tibble	"R (>= 3.4.0)"
tidyr	"R (>= 3.4.0)"
tidyselect	"R (>= 3.4)"
tinytex	NA
TTR	NA
tzdb	"R (>= 3.5.0)"
utf8	"R (>= 2.10)"
vctrs	"R (>= 3.5.0)"
vroom	"R (>= 3.4)"
withr	"R (>= 3.2.0)"
xfun	NA
xts	"R (>= 3.6.0), zoo (>= 1.7-12)"
yaml	NA
zoo	"R (>= 3.1.0), stats"
base	NA
boot	"R (>= 3.0.0), graphics, stats"
class	"R (>= 3.0.0), stats, utils"
cluster	"R (>= 3.5.0)"
codetools	"R (>= 2.1)"
compiler	NA
datasets	NA
foreign	"R (>= 4.0.0)"
graphics	NA
grDevices	NA
grid	NA
KernSmooth	"R (>= 2.5.0), stats"
lattice	"R (>= 4.0.0)"
MASS	"R (>= 4.0), grDevices, graphics, stats, utils"
Matrix	"R (>= 3.5.0), methods"
methods	NA
mgcv	"R (>= 3.6.0), nlme (>= 3.1-64)"
nlme	"R (>= 3.5.0)"
nnet	"R (>= 3.0.0), stats, utils"
parallel	NA
rpart	"R (>= 2.15.0), graphics, stats, grDevices"
spatial	"R (>= 3.0.0), graphics, stats, utils"
splines	NA
stats	NA
stats4	NA
survival	"R (>= 3.5.0)"

tcltk	NA
tools	NA
utils	NA
	Imports
base64enc	NA
bit	NA
bit64	NA
blob	"methods, rlang, vctrs (>= 0.2.1)"
bslib	"base64enc, cachem, grDevices, htmltools (>= 0.5.4), jquerylib\n(>= 0.1.3), j"
cachem	"rlang, fastmap (>= 1.1.1)"
cli	"utils"
clipr	"utils"
cpp11	NA
crayon	"grDevices, methods, utils"
curl	NA
DBI	NA
dbplyr	"blob (>= 1.2.0), cli (>= 3.4.1), DBI (>= 1.0.0), dplyr (>=\n1.1.0), glue (>=
digest	"utils"
DMwR2	"xts (>= 0.9-7), zoo (>= 1.7-10), class (>= 7.3-14), rpart (>=\n4.1-10), quan"
dplyr	"cli (>= 3.4.0), generics, glue (>= 1.3.2), lifecycle (>=\n1.0.3), magrittr ("
ellipsis	"rlang (>= 0.3.0)"
evaluate	"methods"
fansi	"grDevices, utils"
fastmap	NA
fontawesome	"rlang (>= 1.0.6), htmltools (>= 0.5.1.1)"
fs	"methods"
generics	"methods"
gitcreds	NA
glue	"methods"
highr	"xfun (>= 0.18)"
hms	"lifecycle, methods, pkgconfig, rlang (>= 1.0.2), vctrs (>=\n0.3.8)"
htmltools	"utils, digest, grDevices, base64enc, rlang (>= 0.4.12),\nfastmap (>= 1.1.0),
jquerylib	"htmltools"
jsonlite	NA
KernSmooth	NA
knitr	"evaluate (>= 0.15), highr, methods, tools, xfun (>= 0.39),\nyaml (>= 2.1.19)"
lifecycle	"cli (>= 3.4.0), glue, rlang (>= 1.0.6)"
magrittr	NA
Matrix	"grDevices, graphics, grid, lattice, stats, utils"
memoise	"rlang (>= 0.4.10), cachem"
mgcv	"methods, stats, graphics, Matrix, splines, utils"
mime	"tools"
nlme	"graphics, stats, utils, lattice"

palmerpenguins	NA
pillar	"cli (>= 2.3.0), fansi, glue, lifecycle, rlang (>= 1.0.2), utf8\n(>= 1.1.0), v
pkgconfig	"utils"
prettyunits	NA
progress	"hms, prettyunits, R6, crayon"
purrr	"cli (>= 3.6.1), lifecycle (>= 1.0.3), magrittr (>= 1.5.0),\nrlang (>= 1.1.1)
quantmod	"curl, jsonlite(>= 1.1)"
R6	NA
rappdirs	NA
readr	"cli (>= 3.2.0), clipr, crayon, hms (>= 0.4.1), lifecycle (>=\n0.2.0), methods
rlang	"utils"
rmarkdown	"bslib (>= 0.2.5.1), evaluate (>= 0.13), fontawesome (>=\n0.5.0), htmltools (>
sass	"fs (>= 1.2.4), rlang (>= 0.4.10), htmltools (>= 0.5.1), R6,\nrappdirs"
spatial	NA
stringi	"tools, utils, stats"
stringr	"cli, glue (>= 1.6.1), lifecycle (>= 1.0.3), magrittr, rlang\n(>= 1.0.0), str
survival	"graphics, Matrix, methods, splines, stats, utils"
tibble	"fansi (>= 0.4.0), lifecycle (>= 1.0.0), magrittr, methods,\npillar (>= 1.8.1)
tidyr	"cli (>= 3.4.1), dplyr (>= 1.0.10), glue, lifecycle (>= 1.0.3),\nrmagrittr, pur
tidyselect	"cli (>= 3.3.0), glue (>= 1.3.0), lifecycle (>= 1.0.3), rlang\n(>= 1.0.4), vc
tinytex	"xfun (>= 0.29)"
TTR	"xts (>= 0.10-0), zoo, curl"
tzdb	NA
utf8	NA
vctrs	"cli (>= 3.4.0), glue, lifecycle (>= 1.0.3), rlang (>= 1.1.0)"
vroom	"bit64, cli (>= 3.2.0), crayon, glue, hms, lifecycle (>=\n1.0.3), methods, rla
withr	"graphics, grDevices, stats"
xfun	"stats, tools"
xts	"methods"
yaml	NA
zoo	"utils, graphics, grDevices, lattice (>= 0.20-27)"
base	NA
boot	NA
class	"MASS"
cluster	"graphics, grDevices, stats, utils"
codetools	NA
compiler	NA
datasets	NA
foreign	"methods, utils, stats"
graphics	"grDevices"
grDevices	NA
grid	"grDevices, utils"
KernSmooth	NA

lattice	"grid, grDevices, graphics, stats, utils"
MASS	"methods"
Matrix	"graphics, grid, lattice, stats, utils"
methods	"utils, stats"
mgcv	"methods, stats, graphics, Matrix, splines, utils"
nlme	"graphics, stats, utils, lattice"
nnet	NA
parallel	"tools, compiler"
rpart	NA
spatial	NA
splines	"graphics, stats"
stats	"utils, grDevices, graphics"
stats4	"graphics, methods, stats"
survival	"graphics, Matrix, methods, splines, stats, utils"
tcltk	"utils"
tools	NA
utils	NA
	LinkingTo
base64enc	NA
bit	NA
bit64	NA
blob	NA
bslib	NA
cachem	NA
cli	NA
clipr	NA
cpp11	NA
crayon	NA
curl	NA
DBI	NA
dbplyr	NA
digest	NA
DMwR2	NA
dplyr	NA
ellipsis	NA
evaluate	NA
fansi	NA
fastmap	NA
fontawesome	NA
fs	NA
generics	NA
gitcreds	NA
glue	NA

highr	NA
hms	NA
htmltools	NA
jquerylib	NA
jsonlite	NA
KernSmooth	NA
knitr	NA
lifecycle	NA
magrittr	NA
Matrix	NA
memoise	NA
mgcv	NA
mime	NA
nlme	NA
palmerpenguins	NA
pillar	NA
pkgconfig	NA
prettyunits	NA
progress	NA
purrr	"cli"
quantmod	NA
R6	NA
rappdirs	NA
readr	"cpp11, tzdb (>= 0.1.1)"
rlang	NA
rmarkdown	NA
sass	NA
spatial	NA
stringi	NA
stringr	NA
survival	NA
tibble	NA
tidyr	"cpp11 (>= 0.4.0)"
tidyselect	NA
tinytex	NA
TTR	"xts"
tzdb	"cpp11 (>= 0.4.2)"
utf8	NA
vctrs	NA
vroom	"cpp11 (>= 0.2.0), progress (>= 1.2.1), tzdb (>= 0.1.1)"
withr	NA
xfun	NA
xts	"zoo"

yaml	NA
zoo	NA
base	NA
boot	NA
class	NA
cluster	NA
codetools	NA
compiler	NA
datasets	NA
foreign	NA
graphics	NA
grDevices	NA
grid	NA
KernSmooth	NA
lattice	NA
MASS	NA
Matrix	NA
methods	NA
mgcv	NA
nlme	NA
nnet	NA
parallel	NA
rpart	NA
spatial	NA
splines	NA
stats	NA
stats4	NA
survival	NA
tcltk	NA
tools	NA
utils	NA
	Suggests
base64enc	NA
bit	"testthat (>= 0.11.0), roxygen2, knitr, rmarkdown,\nmicrobenchmark, bit64 (>=
bit64	NA
blob	"covr, crayon, pillar (>= 1.2.1), testthat"
bslib	"bsicons, curl, fontawesome, ggplot2, knitr, magrittr,\nrapports, rmarkdown (>=
cachem	"testthat"
cli	"callr, covr, crayon, digest, glue (>= 1.6.0), grDevices,\nhtmltools, htmlwid
clipr	"covr, knitr, rmarkdown, rstudioapi (>= 0.5), testthat (>= \n2.0.0)"
cpp11	"bench, brio, callr, cli, covr, decor, desc, ggplot2, glue,\nknitr, lobster, m
crayon	"mockery, rstudioapi, testthat, withr"
curl	"spelling, testthat (>= 1.0.0), knitr, jsonlite, rmarkdown,\nmagrittr, httpuv

DBI	"blob, covr, DBItest, dbplyr, downlit, dplyr, glue, hms,\nknitr, magrittr, RMA"
dbplyr	"bit64, covr, knitr, Lahman, nycflights13, odbc, RMariaDB (>=\n1.0.2), rmarkd"
digest	"tinytest, simplermardown"
DMwR2	NA
dplyr	"bench, broom, callr, covr, DBI, dbplyr (>= 2.2.1), ggplot2,\nknitr, Lahman, "
ellipsis	"covr, testthat"
evaluate	"covr, ggplot2, lattice, rlang, testthat (>= 3.0.0), withr"
fansi	"unitizer, knitr, rmarkdown"
fastmap	"testthat (>= 2.1.1)"
fontawesome	"covr, dplyr (>= 1.0.8), knitr (>= 1.31), testthat (>= 3.0.0),\nrsvg"
fs	"covr, crayon, knitr, pillar (>= 1.0.0), rmarkdown, spelling,\ntestthat (>= 3"
generics	"covr, pkgload, testthat (>= 3.0.0), tibble, withr"
gitcreds	"codetools, covr, knitr, mockery, oskeyring, rmarkdown,\ntestthat (>= 3.0.0), "
glue	"covr, crayon, DBI, dplyr, forcats, ggplot2, knitr, magrittr,\nmicrobenchmark"
highr	"knitr, markdown, testit"
hms	"crayon, lubridate, pillar (>= 1.1.0), testthat (>= 3.0.0)"
htmltools	"markdown, testthat, withr, Cairo, ragg, shiny"
jquerylib	"testthat"
jsonlite	"httr, vctrs, testthat, knitr, rmarkdown, R.rsp, sf"
KernSmooth	"MASS, carData"
knitr	"bslib, codetools, DBI (>= 0.4-1), digest, formatR, gifski,\ngridSVG, htmlwidg"
lifecycle	"covr, crayon, knitr, lintr, rmarkdown, testthat (>= 3.0.1),\ntibble, tidyver"
magrittr	"covr, knitr, rlang, rmarkdown, testthat"
Matrix	"MASS, datasets, sfsmisc"
memoise	"digest, aws.s3, covr, googleAuthR, googleCloudStorageR, httr,\ntestthat"
mgcv	"parallel, survival, MASS"
mime	NA
nlme	"Hmisc, MASS, SASmixed"
palmerpenguins	"knitr, rmarkdown, tibble, ggplot2, dplyr, tidyr, recipes"
pillar	"bit64, DBI, debugme, DiagrammeR, dplyr, formattable, ggplot2,\nknitr, lubrida"
pkgconfig	"covr, testthat, disposables (>= 1.0.3)"
prettyunits	"codetools, covr, testthat"
progress	"Rcpp, testthat, withr"
purrr	"covr, dplyr (>= 0.7.8), httr, knitr, lubridate, rmarkdown,\ntestthat (>= 3.0"
quantmod	"DBI,RMySQL,RSQLite,timeSeries,xml2,downloader"
R6	"testthat, pryr"
rappdirs	"roxygen2, testthat (>= 3.0.0), covr, withr"
readr	"covr, curl, datasets, knitr, rmarkdown, spelling, stringi,\ntestthat (>= 3.1"
rlang	"cli (>= 3.1.0), covr, crayon, fs, glue, knitr, magrittr,\nmethods, pillar, m"
rmarkdown	"digest, dygraphs, fs, rsconnect, downlit (>= 0.4.0), katex\n(>= 1.4.0), sass"
sass	"testthat, knitr, rmarkdown, withr, shiny, curl"
spatial	"MASS"
stringi	NA

stringr	"covr, htmltools, htmlwidgets, knitr, rmarkdown, testthat (>=\n3.0.0)"
survival	NA
tibble	"bench, bit64, blob, brio, callr, cli, covr, crayon (>=\n1.3.4), DiagrammeR, c
tidyr	"covr, data.table, knitr, readr, repurrrsive (>= 1.1.0),\nrmarkdown, testthat
tidyselect	"covr, crayon, dplyr, knitr, magrittr, rmarkdown, stringr,\ntestthat (>= 3.1.
tinytex	"testit, rstudioapi"
TTR	"RUnit"
tzdb	"covr, testthat (>= 3.0.0)"
utf8	"cli, covr, knitr, rlang, rmarkdown, testthat (>= 3.0.0),\nwwithr"
vctrs	"bit64, covr, crayon, dplyr (>= 0.8.5), generics, knitr,\npillar (>= 1.4.4), p
vroom	"archive, bench (>= 1.1.0), covr, curl, dplyr, forcats, fs,\nggplot2, knitr, p
withr	"callr, covr, DBI, knitr, lattice, methods, rlang, rmarkdown\n(>= 2.12), RSQL
xfun	"testit, parallel, codetools, rstudioapi, tinytex (>= 0.30),\nmime, markdown
xts	"timeSeries, timeDate, tseries, chron, tinytest"
yaml	"RUnit"
zoo	"AER, coda, chron, ggplot2 (>= 3.0.0), mondate, scales,\nstinepack, strucchan
base	"methods"
boot	"MASS, survival"
class	NA
cluster	"MASS, Matrix"
codetools	NA
compiler	NA
datasets	NA
foreign	NA
graphics	NA
grDevices	"KernSmooth"
grid	NA
KernSmooth	"MASS, carData"
lattice	"KernSmooth, MASS, latticeExtra, colorspace"
MASS	"lattice, nlme, nnet, survival"
Matrix	"MASS, expm"
methods	"codetools"
mgcv	"parallel, survival, MASS"
nlme	"Hmisc, MASS, SASmixed"
nnet	"MASS"
parallel	"methods"
rpart	"survival"
spatial	"MASS"
splines	"Matrix, methods"
stats	"MASS, Matrix, SuppDists, methods, stats4"
stats4	NA
survival	NA
tcltk	NA

tools	"codetools, methods, xml2, curl, commonmark, knitr, xfun, mathjaxr, V8"
utils	"methods, xml2, commonmark, knitr"
	Enhances
base64enc	"png"
bit	NA
bit64	NA
blob	NA
bslib	NA
cachem	NA
cli	NA
clipr	NA
cpp11	NA
crayon	NA
curl	NA
DBI	NA
dbplyr	NA
digest	NA
DMwR2	NA
dplyr	NA
ellipsis	NA
evaluate	NA
fansi	NA
fastmap	NA
fontawesome	NA
fs	NA
generics	NA
gitcreds	NA
glue	NA
highr	NA
hms	NA
htmltools	"knitr"
jquerylib	NA
jsonlite	NA
KernSmooth	NA
knitr	NA
lifecycle	NA
magrittr	NA
Matrix	"SparseM, graph"
memoise	NA
mgcv	NA
mime	NA
nlme	NA
palmerpenguins	NA

pillar	NA
pkgconfig	NA
prettyunits	NA
progress	NA
purrr	NA
quantmod	NA
R6	NA
rappdirs	NA
readr	NA
rlang	"winch"
rmarkdown	NA
sass	NA
spatial	NA
stringi	NA
stringr	NA
survival	NA
tibble	NA
tidyr	NA
tidyselect	NA
tinytex	NA
TTR	"quantmod"
tzdb	NA
utf8	NA
vctrs	NA
vroom	NA
withr	NA
xfun	NA
xts	NA
yaml	NA
zoo	NA
base	NA
boot	NA
class	NA
cluster	NA
codetools	NA
compiler	NA
datasets	NA
foreign	NA
graphics	NA
grDevices	NA
grid	NA
KernSmooth	NA
lattice	"chron"

MASS	NA	
Matrix	"MatrixModels, SparseM, graph, igraph, maptools, sfsmisc, sp,\nspdep"	
methods	NA	
mgcv	NA	
nlme	NA	
nnet	NA	
parallel	"snow, Rmpi"	
rpart	NA	
spatial	NA	
splines	NA	
stats	NA	
stats4	NA	
survival	NA	
tcltk	NA	
tools	NA	
utils	NA	
	License	License_is_FOSS
base64enc	"GPL-2 GPL-3"	NA
bit	"GPL-2 GPL-3"	NA
bit64	"GPL-2 GPL-3"	NA
blob	"MIT + file LICENSE"	NA
bslib	"MIT + file LICENSE"	NA
cachem	"MIT + file LICENSE"	NA
cli	"MIT + file LICENSE"	NA
clipr	"GPL-3"	NA
cpp11	"MIT + file LICENSE"	NA
crayon	"MIT + file LICENSE"	NA
curl	"MIT + file LICENSE"	NA
DBI	"LGPL (>= 2.1)"	NA
dbplyr	"MIT + file LICENSE"	NA
digest	"GPL (>= 2)"	NA
DMwR2	"GPL (>= 2)"	NA
dplyr	"MIT + file LICENSE"	NA
ellipsis	"MIT + file LICENSE"	NA
evaluate	"MIT + file LICENSE"	NA
fansi	"GPL-2 GPL-3"	NA
fastmap	"MIT + file LICENSE"	NA
fontawesome	"MIT + file LICENSE"	NA
fs	"MIT + file LICENSE"	NA
generics	"MIT + file LICENSE"	NA
gitcreds	"MIT + file LICENSE"	NA
glue	"MIT + file LICENSE"	NA
highr	"GPL"	NA

hms	"MIT + file LICENSE"	NA
htmltools	"GPL (>= 2)"	NA
jquerylib	"MIT + file LICENSE"	NA
jsonlite	"MIT + file LICENSE"	NA
KernSmooth	"Unlimited"	NA
knitr	"GPL"	NA
lifecycle	"MIT + file LICENSE"	NA
magrittr	"MIT + file LICENSE"	NA
Matrix	"GPL (>= 2) file LICENSE"	NA
memoise	"MIT + file LICENSE"	NA
mgcv	"GPL (>= 2)"	NA
mime	"GPL"	NA
nlme	"GPL (>= 2)"	NA
palmerpenguins	"CC0"	NA
pillar	"MIT + file LICENSE"	NA
pkgconfig	"MIT + file LICENSE"	NA
prettyunits	"MIT + file LICENSE"	NA
progress	"MIT + file LICENSE"	NA
purrr	"MIT + file LICENSE"	NA
quantmod	"GPL-3"	NA
R6	"MIT + file LICENSE"	NA
rappdirs	"MIT + file LICENSE"	NA
readr	"MIT + file LICENSE"	NA
rlang	"MIT + file LICENSE"	NA
rmarkdown	"GPL-3"	NA
sass	"MIT + file LICENSE"	NA
spatial	"GPL-2 GPL-3"	NA
stringi	"file LICENSE"	"yes"
stringr	"MIT + file LICENSE"	NA
survival	"LGPL (>= 2)"	NA
tibble	"MIT + file LICENSE"	NA
tidyr	"MIT + file LICENSE"	NA
tidyselect	"MIT + file LICENSE"	NA
tinytex	"MIT + file LICENSE"	NA
TTR	"GPL (>= 2)"	NA
tzdb	"MIT + file LICENSE"	NA
utf8	"Apache License (== 2.0) file LICENSE"	NA
vctrs	"MIT + file LICENSE"	NA
vroom	"MIT + file LICENSE"	NA
withr	"MIT + file LICENSE"	NA
xfun	"MIT + file LICENSE"	NA
xts	"GPL (>= 2)"	NA
yaml	"BSD_3_clause + file LICENSE"	NA

zoo	"GPL-2 GPL-3"	NA
base	"Part of R 4.3.1"	NA
boot	"Unlimited"	NA
class	"GPL-2 GPL-3"	NA
cluster	"GPL (>= 2)"	NA
codetools	"GPL"	NA
compiler	"Part of R 4.3.1"	NA
datasets	"Part of R 4.3.1"	NA
foreign	"GPL (>= 2)"	NA
graphics	"Part of R 4.3.1"	NA
grDevices	"Part of R 4.3.1"	NA
grid	"Part of R 4.3.1"	NA
KernSmooth	"Unlimited"	NA
lattice	"GPL (>= 2)"	NA
MASS	"GPL-2 GPL-3"	NA
Matrix	"GPL (>= 2) file LICENCE"	NA
methods	"Part of R 4.3.1"	NA
mgcv	"GPL (>= 2)"	NA
nlme	"GPL (>= 2)"	NA
nnet	"GPL-2 GPL-3"	NA
parallel	"Part of R 4.3.1"	NA
rpart	"GPL-2 GPL-3"	NA
spatial	"GPL-2 GPL-3"	NA
splines	"Part of R 4.3.1"	NA
stats	"Part of R 4.3.1"	NA
stats4	"Part of R 4.3.1"	NA
survival	"LGPL (>= 2)"	NA
tcltk	"Part of R 4.3.1"	NA
tools	"Part of R 4.3.1"	NA
utils	"Part of R 4.3.1"	NA
	License_restricts_use OS_type MD5sum NeedsCompilation Built	
base64enc	NA NA NA "yes"	"4.3.0"
bit	NA NA NA "yes"	"4.3.0"
bit64	NA NA NA "yes"	"4.3.0"
blob	NA NA NA "no"	"4.3.0"
bslib	NA NA NA "no"	"4.3.0"
cachem	NA NA NA "yes"	"4.3.0"
cli	NA NA NA "yes"	"4.3.0"
clipr	NA NA NA "no"	"4.3.0"
cpp11	NA NA NA "no"	"4.3.0"
crayon	NA NA NA "no"	"4.3.0"
curl	NA NA NA "yes"	"4.3.0"
DBI	NA NA NA "no"	"4.3.0"

dbplyr	NA	NA	NA	"no"	"4.3.0"
digest	NA	NA	NA	"yes"	"4.3.0"
DMwR2	NA	NA	NA	"no"	"4.3.0"
dplyr	NA	NA	NA	"yes"	"4.3.1"
ellipsis	NA	NA	NA	"yes"	"4.3.0"
evaluate	NA	NA	NA	"no"	"4.3.0"
fansi	NA	NA	NA	"yes"	"4.3.0"
fastmap	NA	NA	NA	"yes"	"4.3.0"
fontawesome	NA	NA	NA	"no"	"4.3.0"
fs	NA	NA	NA	"yes"	"4.3.0"
generics	NA	NA	NA	"no"	"4.3.0"
gitcreds	NA	NA	NA	"no"	"4.3.1"
glue	NA	NA	NA	"yes"	"4.3.0"
highr	NA	NA	NA	"no"	"4.3.0"
hms	NA	NA	NA	"no"	"4.3.0"
htmltools	NA	NA	NA	"yes"	"4.3.0"
jquerylib	NA	NA	NA	"no"	"4.3.0"
jsonlite	NA	NA	NA	"yes"	"4.3.0"
KernSmooth	NA	NA	NA	"yes"	"4.3.1"
knitr	NA	NA	NA	"no"	"4.3.0"
lifecycle	NA	NA	NA	"no"	"4.3.0"
magrittr	NA	NA	NA	"yes"	"4.3.0"
Matrix	NA	NA	NA	"yes"	"4.3.1"
memoise	NA	NA	NA	"no"	"4.3.0"
mgcv	NA	NA	NA	"yes"	"4.3.1"
mime	NA	NA	NA	"yes"	"4.3.0"
nlme	NA	NA	NA	"yes"	"4.3.1"
palmerpenguins	NA	NA	NA	"no"	"4.3.0"
pillar	NA	NA	NA	"no"	"4.3.0"
pkgconfig	NA	NA	NA	"no"	"4.3.0"
prettyunits	NA	NA	NA	"no"	"4.3.0"
progress	NA	NA	NA	"no"	"4.3.0"
purrr	NA	NA	NA	"yes"	"4.3.0"
quantmod	NA	NA	NA	"no"	"4.3.0"
R6	NA	NA	NA	"no"	"4.3.0"
rappdirs	NA	NA	NA	"yes"	"4.3.0"
readr	NA	NA	NA	"yes"	"4.3.0"
rlang	NA	NA	NA	"yes"	"4.3.0"
rmarkdown	NA	NA	NA	"no"	"4.3.0"
sass	NA	NA	NA	"yes"	"4.3.0"
spatial	NA	NA	NA	"yes"	"4.3.1"
stringi	NA	NA	NA	"yes"	"4.3.0"
stringr	NA	NA	NA	"no"	"4.3.0"

survival	NA	NA	NA	"yes"	"4.3.1"
tibble	NA	NA	NA	"yes"	"4.3.0"
tidyr	NA	NA	NA	"yes"	"4.3.0"
tidyselect	NA	NA	NA	"no"	"4.3.0"
tinytex	NA	NA	NA	"no"	"4.3.0"
TTR	NA	NA	NA	"yes"	"4.3.0"
tzdb	NA	NA	NA	"yes"	"4.3.0"
utf8	NA	NA	NA	"yes"	"4.3.0"
vctrs	NA	NA	NA	"yes"	"4.3.0"
vroom	NA	NA	NA	"yes"	"4.3.0"
withr	NA	NA	NA	"no"	"4.3.0"
xfun	NA	NA	NA	"yes"	"4.3.0"
xts	NA	NA	NA	"yes"	"4.3.0"
yaml	NA	NA	NA	"yes"	"4.3.0"
zoo	NA	NA	NA	"yes"	"4.3.0"
base	NA	NA	NA	NA	"4.3.1"
boot	NA	NA	NA	"no"	"4.3.1"
class	NA	NA	NA	"yes"	"4.3.1"
cluster	NA	NA	NA	"yes"	"4.3.1"
codetools	NA	NA	NA	"no"	"4.3.1"
compiler	NA	NA	NA	NA	"4.3.1"
datasets	NA	NA	NA	NA	"4.3.1"
foreign	NA	NA	NA	"yes"	"4.3.1"
graphics	NA	NA	NA	"yes"	"4.3.1"
grDevices	NA	NA	NA	"yes"	"4.3.1"
grid	NA	NA	NA	"yes"	"4.3.1"
KernSmooth	NA	NA	NA	"yes"	"4.3.1"
lattice	NA	NA	NA	"yes"	"4.3.1"
MASS	NA	NA	NA	"yes"	"4.3.1"
Matrix	NA	NA	NA	"yes"	"4.3.1"
methods	NA	NA	NA	"yes"	"4.3.1"
mgcv	NA	NA	NA	"yes"	"4.3.1"
nlme	NA	NA	NA	"yes"	"4.3.1"
nnet	NA	NA	NA	"yes"	"4.3.1"
parallel	NA	NA	NA	"yes"	"4.3.1"
rpart	NA	NA	NA	"yes"	"4.3.1"
spatial	NA	NA	NA	"yes"	"4.3.1"
splines	NA	NA	NA	"yes"	"4.3.1"
stats	NA	NA	NA	"yes"	"4.3.1"
stats4	NA	NA	NA	NA	"4.3.1"
survival	NA	NA	NA	"yes"	"4.3.1"
tcltk	NA	NA	NA	"yes"	"4.3.1"
tools	NA	NA	NA	"yes"	"4.3.1"

utils	NA	NA	NA	"yes"	"4.3.1"
-------	----	----	----	-------	---------

Find out if the installed packages have a new version on CRAN:

```
old.packages()
```

	Package	LibPath	Installed	Built
KernSmooth	"KernSmooth"	"/opt/R/4.3.1/lib/R/library"	"2.23-21"	"4.3.1"
Matrix	"Matrix"	"/opt/R/4.3.1/lib/R/library"	"1.5-4.1"	"4.3.1"
mgcv	"mgcv"	"/opt/R/4.3.1/lib/R/library"	"1.8-42"	"4.3.1"
nlme	"nlme"	"/opt/R/4.3.1/lib/R/library"	"3.1-162"	"4.3.1"
spatial	"spatial"	"/opt/R/4.3.1/lib/R/library"	"7.3-16"	"4.3.1"
survival	"survival"	"/opt/R/4.3.1/lib/R/library"	"3.5-5"	"4.3.1"
	ReposVer	Repository		
KernSmooth	"2.23-22"	"http://rspm/default/__linux__/focal/latest/src/contrib"		
Matrix	"1.6-1"	"http://rspm/default/__linux__/focal/latest/src/contrib"		
mgcv	"1.9-0"	"http://rspm/default/__linux__/focal/latest/src/contrib"		
nlme	"3.1-163"	"http://rspm/default/__linux__/focal/latest/src/contrib"		
spatial	"7.3-17"	"http://rspm/default/__linux__/focal/latest/src/contrib"		
survival	"3.5-7"	"http://rspm/default/__linux__/focal/latest/src/contrib"		

Updating all the installed packages to the latest version:

```
#update.packages()
```

Updating the package without asking for confirmation for each package:

```
update.packages(ask = FALSE)
```

Warning: package 'KernSmooth' in library '/opt/R/4.3.1/lib/R/library' will not be updated

Warning: package 'Matrix' in library '/opt/R/4.3.1/lib/R/library' will not be updated

Warning: package 'mgcv' in library '/opt/R/4.3.1/lib/R/library' will not be updated

Warning: package 'nlme' in library '/opt/R/4.3.1/lib/R/library' will not be updated

Warning: package 'spatial' in library '/opt/R/4.3.1/lib/R/library' will not be updated

Warning: package 'survival' in library '/opt/R/4.3.1/lib/R/library' will not be updated

To find out which namespace/package a function belongs in the installed packages:

```
mean
```

```
function (x, ...)
UseMethod("mean")
<bytecode: 0x55a1fb2b7f58>
<environment: namespace:base>
```

Get help related to a function in an installed package, for example taking `mean()`:

```
help(mean)
```

- When two packages are providing function with the same name and we need to use both the functions, we can use `package::functionname()` to differentiate between the two functions.
- When we want to use a package which is already added, we can search for the package as below:

```
RSiteSearch('neural networks')
```

A search query has been submitted to <https://search.r-project.org>
The results page should open in your browser shortly

Project and Session Management

Including only save function as a comment

- `save(my.function, mydataset, file="path_to_my.session.RData")`
`load("path_to_my.session.RData")`

All objects are saved in **.RData** file in the current working directory to be loaded in future.

- `save.image()`

Get and set working directory

- `getwd()`
`setwd("/home/gchism/Documents/523")` # setwd using what you get from `getwd()`
`getwd()`

R Objects and Variables

Variables are the references to certain storage locations in the database or memory which holds some object ranging from simple number to complex model associating an object to a variable.

```
vat <- 0.2
```

Value vat holds

```
vat
```

```
[1] 0.2
```

Directly print the values when enclosed inside ()

```
(vat <- 0.2)
```

```
[1] 0.2
```

Certain examples are shown below:

1.

```
x <- 5
y <- vat * x
y
```

```
[1] 1
```

2.

```
z <- (y/2)^2
y
```

```
[1] 1
```

3.

```
z
```

```
[1] 0.25
```

All the variables created are alive until you delete it or when we exit R without saving them to list variables currently alive: **ls()** or **objects()** :

```
ls()
```

```
[1] "algae"          "algae.sols"      "has_annotatons"  "test.algae"
[5] "vat"            "x"               "y"               "z"
```

```
objects()
```

```
[1] "algae"          "algae.sols"      "has_annotatons"  "test.algae"
[5] "vat"            "x"               "y"               "z"
```

Remove a variable from to free the memory space:

```
rm(vat)
```

Functions

Functions are a special type of R object designed to carry out some operation.

Functions expects some input parameters and outputs results of the carried out operations.

R has many functions already available, libraries that are loaded contains functions we can use, new functions can also be created.

Examples:

```
max(4, 5, 6, 12, -4)
```

```
[1] 12
```

```
mean(4, 5, 6, 12, -4)
```

```
[1] 4
```

```
max(sample(1:100, 30))
```

```
[1] 98
```

```
mean(sample(1:100, 30))
```

```
[1] 45.6
```

help(sample) is used below to find out why the same function with same argument give different results in the above chunks:

```
help(sample)
```

```
set.seed(1) #the seed determines the starting point used in generating a sequence of pseud  
#there is a function to remove the seed:rm(.Random.seed, envir=.GlobalEnv)
```

```
rmnorm(1) #give me one number from a normal distribution
```

```
[1] -0.6264538
```

```
rmnorm(1)
```

```
[1] 0.1836433
```

```
set.seed(2)  
rmnorm(1)
```

```
[1] -0.8969145
```

```
rmnorm(1)
```

```
[1] 0.1848492
```

`set.seed()` is used to make sure multiple executions of a program involving random samples give the same result, used in short for debugging purposes.

To Create a new function, **se (standard error of means)**.
Test if `.se` exists in our environment.

```
exists("se")
```

```
[1] FALSE
```

`se` object not found, creating the function that computes the `se`:

```
se <- function(x){  
  variance <- var(x)  
  n <- length(x)  
  return(sqrt(variance/n))  
}
```

`se` Object has been created:

```
exists("se")
```

```
[1] TRUE
```

Creating another function with multiple arguments:

convMeters :- this function will convert meters to inches, feet, yards, and miles.

exists("convMeters")

```
convMeters <- function (x, to="inch"){  
  factor = switch(to, inch=39.3701, foot=3.28084, yard=1.09361, mile=0.000621371, NA)  
  if(is.na(factor)) stop ("unknown target unit")  
  else return (x*factor)  
}  
convMeters(23, "foot")
```

```
[1] 75.45932
```

When no argument is provided, the function will use default value as **'inch'**

```
convMeters(40)
```

```
[1] 1574.804
```

Factors

A factor can be seen as a categorical (i.e., nominal) variable factor levels are the set of unique values the nominal variable could have. Factors are different from characters.

To create a factor, use **factor()** :

```
g <-c('f', 'm', 'f', 'f', 'f', 'm', 'm', 'f')  
g <- factor(g)
```

```
other.g <-factor(c('m', 'm', 'm', 'm'), levels= c('f', 'm'))  
other.g
```



```
[1] m m m m
Levels: f m
```

Comparing the above with the following:

```
other.g <-factor(c('m', 'm', 'm', 'm'))
other.g
```

```
[1] m m m m
Levels: m
```

Using `table()` function:

```
g <- factor(c('f', 'm', 'f', 'f', 'f', 'm', 'm', 'f'))
table(g)
```

```
g
f m
5 3
```

Adding age factor to the table, table can have more than two factors:

```
a <- factor(c('adult', 'juvenile','adult', 'juvenile','adult', 'juvenile','juvenile', 'juv
table(a, g)
```

```
      g
a      f m
adult  3 0
juvenile 2 3
```

What if `a` factor is not the same length as the `g` factor:

```
a <- factor(c('adult', 'juvenile','adult', 'juvenile','adult', 'juvenile','juvenile'))
#table(a, g)
```

Bringing old `a` table back and create a new table with factor `g`

```
a <- factor(c('adult', 'juvenile','adult', 'juvenile','adult', 'juvenile','juvenile', 'juv
t <- table(a, g)
t
```

```
      g
a      f m
adult  3 0
juvenile 2 3
```

Finding marginal frequencies for a factor:

```
margin.table(t, 1)#1 refers to the first factor, a (age)
```

```
a
  adult juvenile
      3         5
```

```
margin.table(t, 2)# now find the marginal freq of the second factor g
```

```
g
f m
5 3
```

```
t
```

```
      g
a      f m
adult  3 0
juvenile 2 3
```

```
prop.table(t, 1) #use the margin generated for the 1st factor a
```

```
      g
a      f  m
adult  1.0 0.0
juvenile 0.4 0.6
```

```
prop.table(t, 2)
```

a	g	
	f	m
adult	0.6	0.0
juvenile	0.4	1.0

```
prop.table(t) #overall
```

a	g	
	f	m
adult	0.375	0.000
juvenile	0.250	0.375

```
prop.table(t) * 100
```

a	g	
	f	m
adult	37.5	0.0
juvenile	25.0	37.5

R Data Structures:

Vectors

- It is the most basic data object.
- One single number is a vector with a single element.
- All elements in one vector should be of one base datatype.

Creating a vector

```
v <- c(2, 5, 3, 4)
length(v)
```

```
[1] 4
```

Describes Datatype of element in vector **v**:

```
mode(v)
```

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

```
v <- c(2, 5, 3, 4, 'me')  
mode(v)
```

```
[1] "character"
```

```
v
```

```
[1] "2" "5" "3" "4" "me"
```

NA is often used to represent a missing value:

```
v <- c(2, 5, 3, 4, NA)  
mode(v)
```

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

```
v
```

```
[1] 2 5 3 4 NA
```

A boolean vector (**TRUE**, **FALSE**)

```
b <- c(TRUE, FALSE, NA, TRUE)  
mode(b)
```

```
[1] "logical"
```

```
b
```

```
[1] TRUE FALSE NA TRUE
```

Indexing is starting with 1 in vectors:

```
b[3]
```

```
[1] NA
```

```
b[3] <- TRUE  
b
```

```
[1] TRUE FALSE TRUE TRUE
```

```
b[10] <- FALSE  
b
```

```
[1] TRUE FALSE TRUE TRUE NA NA NA NA NA FALSE
```

Creating an empty vector:

```
e <-vector()  
mode(e)
```

```
[1] "logical"
```

```
e <- c()  
mode(e)
```

```
[1] "NULL"
```

```
length(e)
```

```
[1] 0
```

Using vector elements to construct another vector:

```
b2 <-c(b[1], b[3], b[5])  
b2
```

```
[1] TRUE TRUE  NA
```

Find the square root of all elements in **v**:

```
sqrt(v)
```

```
[1] 1.414214 2.236068 1.732051 2.000000      NA
```

Vector arithmetic

```
v1 <- c(3, 6, 9)  
v2 <- c(1, 4, 8)  
v1+v2 #addition
```

```
[1] 4 10 17
```

```
v1*v2 #dot product
```

```
[1] 3 24 72
```

```
v1-v2 #subtraction
```

```
[1] 2 2 1
```

```
v1/v2 #divsion
```

```
[1] 3.000 1.500 1.125
```

R uses recycling rule to make the shorter vector the same length as the longer vector, which makes R to enable arithmetic operations.

```
v3 <- c(1, 4)
v1+v3#the recycling rule makes v3 [1, 4, 1]
```

```
[1] 4 10 10
```

```
2*v1
```

```
[1] 6 12 18
```

Vector summary:

Using vector to illustrate “for” loop:

```
mysum <- function (x){
  sum <- 0
  for(i in 1:length(x)){
    sum <- sum + x[i]
  }
  return (sum)
}

(mysum (c(1, 2, 3)))
```

```
[1] 6
```

Print numbers from 1 to 10:

```
(x <-1:10)
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

Print numbers from 10 to 1:

```
(x <-10:1)
```

```
[1] 10 9 8 7 6 5 4 3 2 1
```

```
10:15-1
```

```
[1] 9 10 11 12 13 14
```

```
10:(15-1)
```

```
[1] 10 11 12 13 14
```

Using `seq()` to generate sequence with real numbers:

```
(seq(length=10, from=-2, by=0.5)) #10 values, starting from 2, interval/step = 0.5
```

```
[1] -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0 2.5
```

```
(seq(length=10, from=-2, by=0.5)) #10 values, starting from 2, interval/step = 0.5
```

```
[1] -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0 2.5
```

```
(rep(5, 10))
```

```
[1] 5 5 5 5 5 5 5 5 5 5
```

```
(rep("hi", 3))
```

```
[1] "hi" "hi" "hi"
```



```
(rep(1:2, 3))
```

```
[1] 1 2 1 2 1 2
```

```
(rep(TRUE:FALSE, 3))
```

```
[1] 1 0 1 0 1 0
```

```
(rep(1:2, each=3))
```

```
[1] 1 1 1 2 2 2
```

`gl()` for generating factor levels:

```
gl(3, 5) #three levels, each repeat 5 times
```

```
[1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3  
Levels: 1 2 3
```

```
gl(2, 5, labels= c('female', 'male'))#two levels, each level repeat 5 times
```

```
[1] female female female female female male   male   male   male   male  
Levels: female male
```

```
#first argument 2 says two levels.
```

```
#second argument 1 says repeat once
```

```
#third argument 20 says generate 20 values
```

```
gl(2, 1, 20, labels=c('female', 'male'))#10 alternating female and male pairs, a total of
```

```
[1] female male   female male   female male   female male   female male  
[11] female male   female male   female male   female male   female male  
Levels: female male
```

Labeling a dataset using **factor()** to convert number sequence to factor level labels:

```
n <- rep(1:2, each=3)
(n <- factor(n,
             levels = c(1, 2),
             labels = c('female', 'male')
             ))
```

```
[1] female female female male   male   male
Levels: female male
```

```
n
```

```
[1] female female female male   male   male
Levels: female male
```

Generate 10 values following a normal distribution with mean = 10 and standard deviation = 3:

```
(rnorm(10, mean=10, sd=3))
```

```
[1] 14.763536  6.608873  9.759245 10.397261 12.123864  9.280906 15.953422
[8]  9.583639 11.252952 12.945258
```

```
(rt(10, df=5)) #10 values following a Student T distribution with degree of freedom of 5
```

```
[1] -0.69423122 -1.90338399  0.02947754  0.27787640 -0.85885978  2.18155848
[7] -2.34536097 -0.52103705  0.25674735  0.25832186
```

Exercise:

1. Generate a random sample of normally distributed data of size 100, with a mean of 20 and standard deviation 4

```
sample <- (rnorm(100, mean=20, sd=4))
```

2. Compute the standard error of means of the dataset.

```
se(sample)
```

```
[1] 0.446371
```

Sub-setting

Use boolean operators:

```
x <- c(0, -3, 4, -1, 45, 90, -5)
#select all elements that is greater than 0
(gtzero <- x[x>0])
```

```
[1] 4 45 90
```

Use | (or), and & (and) operators:

```
x <- c(0, -3, 4, -1, 45, 90, -5)
(x[x<=-2 | x>5])
```

```
[1] -3 45 90 -5
```

```
(x[x>40 & x<100])
```

```
[1] 45 90
```

Use a vector index:

```
x <- c(0, -3, 4, -1, 45, 90, -5)
(x[c(4, 6)])#select the 4th and 6th elements in the vector
```

```
[1] -1 90
```

```
(y<-c(4,6)) #same as above
```

```
[1] 4 6
```

```
(x[y])
```

```
[1] -1 90
```

```
(x[1:3]) #select the 1st to the 3rd elements in the vector
```

```
[1] 0 -3 4
```

Use negative index to exclude elements:

```
x <- c(0, -3, 4, -1, 45, 90, -5)
(x[-1]) #select all but the first element
```

```
[1] -3 4 -1 45 90 -5
```

```
(x[-c(4, 6)])
```

```
[1] 0 -3 4 45 -5
```

```
(x[-(1:3)])
```

```
[1] -1 45 90 -5
```

Named elements

Elements in a vector can have names.

Assigning names to vector elements:

```
x <- c(0, -3, 4, -1, 45, 90, -5)
names(x) <- c('s1', 's2', 's3', 's4', 's5', 's6', 's7')
x
```

```
s1 s2 s3 s4 s5 s6 s7
0 -3 4 -1 45 90 -5
```

Creating a vector with named elements:

```
(pH <- c(area1=4.5, area2=5.7, area3=9.8, mud=7.2))
```

```
area1 area2 area3 mud
4.5 5.7 9.8 7.2
```

```
pH['mud']
```

```
mud
7.2
```

```
#x[-s1] #results in error
```

```
#x["-s1"] #results in error
```

```
#x[s1:s7] #results in error
```

```
#x[c('s1':'s7')] #results in error
```

```
pH[]
```

```
area1 area2 area3 mud
4.5 5.7 9.8 7.2
```

```
pH
```

```
area1 area2 area3 mud
4.5 5.7 9.8 7.2
```

```
pH[] <- 0
pH
```

```
area1 area2 area3 mud
      0      0      0  0
```

```
pH<- 0
pH
```

```
[1] 0
```

This is different from `pH <- 0`, why?

The first method zero the values because the empty `[]` selected all indices from the vector and changed their values to zero. The second assignment without the brackets is assigning the value of zero to `pH` and is now basically a variable and not a vector.

More R Data Structures

Matrices and Arrays

- Arrays and matrices are essentially long vectors **organized** by dimensions.
- Matrices are two dimensional, where as Arrays can be multiple dimensions, but both hold the same type of values.

Matrices

- Create a matrix

```
m <- c(45, 23, 66, 77, 33, 44, 56, 12, 78, 23)
is.vector(m)
```

```
[1] TRUE
```

```
is.matrix(m)
```

```
[1] FALSE
```

```
is.array(m)
```

```
[1] FALSE
```

```
#'organizing' the vector as a matrix  
dim(m) <-c(2, 5)#make the vector a 2 by 5 matrix, 2x5 must = length of the vector  
m
```

```
      [,1] [,2] [,3] [,4] [,5]  
[1,]   45   66   33   56   78  
[2,]   23   77   44   12   23
```

```
is.vector(m)
```

```
[1] FALSE
```

```
is.array(m)
```

```
[1] TRUE
```

Putting elements in matrix by row instead of columns:

```
(m <- matrix(c(45, 23, 66, 77, 33, 44, 56, 12, 78, 23), 2, 5, byrow = TRUE))
```

```
      [,1] [,2] [,3] [,4] [,5]  
[1,]   45   23   66   77   33  
[2,]   44   56   12   78   23
```

Exercise:

Create a matrix with two columns:

First columns hold age data for a group of students 11, 11, 12, 13, 14, 9, 8, and second columns hold grades 5, 5, 6, 7, 8, 4, 3.

```
test <-matrix(c(11, 11, 12, 13, 14, 9, 8, 5, 5, 6, 7, 8, 4, 3), 7, 2)
test
```

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

Access matrix elements using position indexes (again, index starting from 1):

```
m <- c(45, 23, 66, 77, 33, 44, 56, 12, 78, 23)
#then 'organize' the vector as a matrix
dim(m) <- c(2, 5)#make the vector a 2 by 5 matrix, 2x5 must = length of the vector
m
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]   45   66   33   56   78
[2,]   23   77   44   12   23
```

```
m[2, 3]#the element at row 2 and column 3
```

```
[1] 44
```

Sub-setting a matrix is similar to sub-setting on a vector.

The result is a value (a value is a vector), a vector, or a matrix:

```
(s<- m[2, 1]) # select one value
```

```
[1] 23
```

```
(m<- m [c(1,2), -c(3, 5)]) #select 1st row and 1st, 2nd, and 4th columns: result is a vect
```



```
      [,1] [,2] [,3]
[1,]   45   66   56
[2,]   23   77   12
```

```
(m[1,]) #select complete row or column: 1st row, result is a vector
```

```
[1] 45 66 56
```

```
(v<-m[,1]) # 1st column, result is a vector
```

```
[1] 45 23
```

```
is.vector(m)
```

```
[1] FALSE
```

```
is.matrix(m)
```

```
[1] TRUE
```

```
is.vector(s)
```

```
[1] TRUE
```

```
is.vector(v)
```

```
[1] TRUE
```

```
is.matrix(v)
```

```
[1] FALSE
```

Use `drop = FALSE` to keep the results as a matrix (not vectors like shown above)

```
m <- matrix(c(45, 23, 66, 77, 33, 44, 56, 12, 78, 23), 2, 5)
(m<-m[, 2, drop = FALSE])
```

```
      [,1]
[1,]    66
[2,]    77
```

```
is.matrix(m)
```

```
[1] TRUE
```

```
is.vector(m)
```

```
[1] FALSE
```

cbind() and **rbind()**: join together two or more vectors or matrices, by column, or by row:

```
cbind (c(1,2,3), c(4, 5, 6))
```

```
      [,1] [,2]
[1,]     1     4
[2,]     2     5
[3,]     3     6
```

```
rbind (c(1,2,3), c(4, 5, 6))
```

```
      [,1] [,2] [,3]
[1,]     1     2     3
[2,]     4     5     6
```

```
m <- matrix(c(45, 23, 66, 77, 33, 44, 56, 12, 78, 23), 2, 5)
(a <- rbind (c(1,2,3,4,5), m))
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1	2	3	4	5
[2,]	45	66	33	56	78
[3,]	23	77	44	12	23

```
is.array(a)
```

```
[1] TRUE
```

```
is.matrix(a)
```

```
[1] TRUE
```

Exercise:

What will m1-m4 look like?

```
m1 <- matrix(rep(10, 9), 3, 3)
m1
```

	[,1]	[,2]	[,3]
[1,]	10	10	10
[2,]	10	10	10
[3,]	10	10	10

```
m2 <- cbind (c(1,2,3), c(4, 5, 6))
m2
```

	[,1]	[,2]
[1,]	1	4
[2,]	2	5
[3,]	3	6

```
m3 <- cbind (m1[,1], m2[,2])
m3
```

```

      [,1] [,2]
[1,]   10   2
[2,]   10   5
[3,]   10   2

```

```

m4 <- cbind (m1[,1], m2[,2])
m4

```

```

      [,1] [,2]
[1,]   10   4
[2,]   10   5
[3,]   10   6

```

```

#m1-m4

```

Ans: Error in m1 - m4 : non-conformable arrays

Named rows and columns

```

sales <- matrix(c(10, 30, 40, 50, 43, 56, 21, 30), 2, 4, byrow=TRUE)
colnames(sales) <- c('1qrt', '2qrt', '3qrt', '4qrt')
rownames(sales) <- c('store1', 'store2')
sales

```

```

      1qrt 2qrt 3qrt 4qrt
store1   10   30   40   50
store2   43   56   21   30

```

Exercise:

Find store1 1qrt sale. 2. List store2's 1st and 4th quarter sales:

```

sales['store1', '1qrt']

```

```

[1] 10

```

```
sales['store2', c('1qrt', '4qrt')]
```

```
1qrt 4qrt
43    30
```

Arrays

Matrices and Arrays are similar, whereas arrays are more than 2 dimensions.

- 3-D array:

```
a <- array(1:48, dim= c(4, 3, 2))
a
```

```
, , 1
```

	[,1]	[,2]	[,3]
[1,]	1	5	9
[2,]	2	6	10
[3,]	3	7	11
[4,]	4	8	12

```
, , 2
```

	[,1]	[,2]	[,3]
[1,]	13	17	21
[2,]	14	18	22
[3,]	15	19	23
[4,]	16	20	24

Selecting array elements using the indexes, results may be a value, a vector, a matrix or an array, depending on the use of **drop=FALSE**:

```
a [1, 3, 2]
```

```
[1] 21
```

```
a [1, , 2]
```

```
[1] 13 17 21
```

```
a [1, , 2, drop=FALSE]
```

```
, , 1
```

```
      [,1] [,2] [,3]  
[1,]    13    17    21
```

```
a [4, 3, ]
```

```
[1] 12 24
```

```
a [c(2, 3), , -2]
```

```
      [,1] [,2] [,3]  
[1,]     2     6    10  
[2,]     3     7    11
```

Assigning names to dimensions of an array.

`[[]]` selects one dimension:

```
dimnames(a)[[1]] <-c("1qrt", "2qrt", "3qrt", "4qrt")  
dimnames(a)[[2]] <-c("store1", "store2", "store3")  
dimnames(a)[[3]] <-c("2017", "2018")  
a
```

, , 2017

	store1	store2	store3
1qrt	1	5	9
2qrt	2	6	10
3qrt	3	7	11
4qrt	4	8	12

, , 2018

	store1	store2	store3
1qrt	13	17	21
2qrt	14	18	22
3qrt	15	19	23
4qrt	16	20	24

We can use `list()` to specify names:

```
ar <- array(data      = 1:27,  
             dim       = c(3, 3, 3),  
             dimnames = list(c("a", "b", "c"),  
                               ar
```

, , g

	d	e	f
a	1	4	7
b	2	5	8
c	3	6	9

, , h

	d	e	f
a	10	13	16
b	11	14	17
c	12	15	18

, , i

	d	e	f
a	19	22	25

```
b 20 23 26
c 21 24 27
```

Splitting array into matrices:

Performing arithmetic operations on matrices, keeping in mind the recycling rules are applicable:

```
#matrix1 <- ar[,g]
#matrix1
```

```
matrix1 <- ar[, 'g']
matrix1
```

```
  d e f
a 1 4 7
b 2 5 8
c 3 6 9
```

```
matrix2 <- ar[, 'h']
matrix2
```

```
  d e f
a 10 13 16
b 11 14 17
c 12 15 18
```

```
sum <- matrix1 + matrix2
sum
```

```
  d e f
a 11 17 23
b 13 19 25
c 15 21 27
```

```
matrix1*3
```



```
      d   e   f
a 3 12 21
b 6 15 24
c 9 18 27
```

```
matrix1*c(2, 3)
```

```
      d   e   f
a 2 12 14
b 6 10 24
c 6 18 18
```

```
matrix1*c(2,3,2,3,2,3,2,3,2)
```

```
      d   e   f
a 2 12 14
b 6 10 24
c 6 18 18
```

```
matrix1*c(1, 2, 3)
```

```
      d   e   f
a 1   4   7
b 4 10 16
c 9 18 27
```

```
matrix1/c(1, 2, 3)
```

```
      d   e   f
a 1 4.0 7
b 1 2.5 4
c 1 2.0 3
```

```
matrix1/c(1, 2, 3, 1, 2, 3, 1, 2, 3)
```

```
      d    e f
a 1 4.0 7
b 1 2.5 4
c 1 2.0 3
```

Lists

- Lists are also vectors, but they are **recursive**.
- Lists can hold other lists, it can hold data of different types.
- Consists an ordered collection of objects known as their components.
- Components are not necessarily of same type.
- Components in the list are numbered and can also have a name attached to them.

Use `list$component_name` to access a component in a *list* can not be used on atomic vectors.

```
mylist <- list(stud.id=34453,
               stud.name="John",
               stud.marks= c(13, 3, 12, 15, 19)
               )
```

```
mylist$stud.id
```

```
[1] 34453
```

```
mylist[1]
```

```
$stud.id
[1] 34453
```

```
mylist[[1]]
```

```
[1] 34453
```

```
mylist["stud.id"]
```

```
$stud.id  
[1] 34453
```

```
handle <- "stud.id"  
mylist[handle]
```

```
$stud.id  
[1] 34453
```

```
mylist[["stud.id"]]
```

```
[1] 34453
```

Subset with [

- Subset extraction can be performed using both indices and names. To utilize names, the object should possess a name-related attribute like names, rownames, colnames, and so on.
- Negative integers can be employed to signify exclusion.
- Variables that are not enclosed in quotes are interpolated when placed within the brackets.

Extract one item with [[

- Double square brackets serve as a means to isolate a single element from a potentially larger collection.
- When applied to vectors, they produce single-value vectors; data frames result in column vectors, while for lists, you retrieve a single element.
- The outcome may not necessarily match the original container's object type. The dimension of the result corresponds to the dimension of the extracted item, which may not always be 1. Furthermore, it's worth reiterating that both names and indices can be employed for this purpose, and variable interpolation remains applicable.

Interact with "\$":

```
mylist <- list(stud.id=34453,  
              stud.name="John",  
              stud.marks= c(13, 3, 12, 15, 19)  
              )  
mylist$stud.marks
```

```
[1] 13  3 12 15 19
```

```
mylist$stud.marks[2]
```

```
[1] 3
```

Change names:

```
names(mylist)
```

```
[1] "stud.id"    "stud.name"  "stud.marks"
```

```
names(mylist) <- c('id','name','marks')  
names(mylist)
```

```
[1] "id"    "name"  "marks"
```

```
mylist
```

```
$id
```

```
[1] 34453
```

```
$name
```

```
[1] "John"
```

```
$marks
```

```
[1] 13  3 12 15 19
```

Add new component:

```
mylist$parents.names <- c('Ana', "Mike")
mylist
```

\$id

[1] 34453

\$name

[1] "John"

\$marks

[1] 13 3 12 15 19

\$parents.names

[1] "Ana" "Mike"

Use `c()` to concatenate two lists:

```
newlist <- list(age=19, sex="male");
expandedlist <-c(mylist, newlist)
expandedlist
```

\$id

[1] 34453

\$name

[1] "John"

\$marks

[1] 13 3 12 15 19

\$parents.names

[1] "Ana" "Mike"

\$age

[1] 19

\$sex

[1] "male"

```
length(expandedlist)
```

```
[1] 6
```

Remove list components using negative index, or using NULL

Exercise:

Starting with the expanded list given above, what will be the result of the following statement?
Consider the statement one by one.

```
expandedlist <- expandedlist[-5]  
expandedlist <- expandedlist[c(-1,-5)]  
expandedlist$parents.names <- NULL  
expandedlist[['marks']] <- NULL
```

```
mylist
```

```
$id
```

```
[1] 34453
```

```
$name
```

```
[1] "John"
```

```
$marks
```

```
[1] 13  3 12 15 19
```

```
$parents.names
```

```
[1] "Ana" "Mike"
```

```
unlist(mylist)
```

id	name	marks1	marks2	marks3
"34453"	"John"	"13"	"3"	"12"
marks4	marks5	parents.names1	parents.names2	
"15"	"19"	"Ana"	"Mike"	

```
mode(mylist)
```

```
[1] "list"
```

```
is.vector(unlist(mylist)) #atomic list with names
```

```
[1] TRUE
```

```
is.list(mylist)
```

```
[1] TRUE
```

```
is.atomic(mylist)
```

```
[1] FALSE
```

```
is.list(unlist(mylist))
```

```
[1] FALSE
```

Data Frames

- Special kind of list: each row is an observation, each column is an attribute.
- Column name should not be empty, row name should be unique.
- Data stored in Data Frames can be numeric, factor or character type.

Create a data frame

```
my.dataframe <- data.frame(site=c('A', 'B', 'A','A', 'B'),  
                           season=c('winter', 'summer', 'summer', 'spring', 'fall'),  
my.dataframe
```

```

  site season  ph
1    A winter 7.4
2    B summer 6.3
3    A summer 8.6
4    A spring 7.2
5    B  fall 8.9

```

Exercise:

Given ‘my.dataframes’, what values will the following statements access?

```

my.dataframe <- data.frame(site=c('A', 'B', 'A','A', 'B'),
                           season=c('winter', 'summer', 'summer', 'spring', 'fall'),
my.dataframe[3, 2]

```

```
[1] "summer"
```

```
my.dataframe[['site']]
```

```
[1] "A" "B" "A" "A" "B"
```

```
my.dataframe['site']
```

```

site
1    A
2    B
3    A
4    A
5    B

```

```
my.dataframe[my.dataframe$ph>7, ]
```

```

site season  ph
1    A winter 7.4
3    A summer 8.6
4    A spring 7.2
5    B  fall 8.9

```



```
my.dataframe[my.dataframe$ph>7, 'site']
```

```
[1] "A" "A" "A" "B"
```

```
my.dataframe[my.dataframe$ph>7, c('site', 'ph')]
```

```
  site ph
1    A 7.4
3    A 8.6
4    A 7.2
5    B 8.9
```

Use **subset()** to query a data frame

```
subset(my.dataframe, ph>7)
```

```
  site season ph
1    A winter 7.4
3    A summer 8.6
4    A spring 7.2
5    B  fall 8.9
```

```
subset(my.dataframe, ph>7, c("site", "ph"))
```

```
  site ph
1    A 7.4
3    A 8.6
4    A 7.2
5    B 8.9
```

```
subset(my.dataframe[1:2,], ph>7, c(site, ph))
```

```
  site ph
1    A 7.4
```

To change values in data frame - add 1 to **summer ph**:

```
my.dataframe[my.dataframe$season=='summer', 'ph'] <- my.dataframe[my.dataframe$season=='summer', 'ph']
my.dataframe[my.dataframe$season=='summer', 'ph']
```

```
[1] 7.3 9.6
```

```
my.dataframe[my.dataframe$season=='summer' & my.dataframe$ph>8, 'ph'] <- my.dataframe[my.dataframe$season=='summer' & my.dataframe$ph>8, 'ph']
my.dataframe[my.dataframe$season=='summer', 'ph']
```

```
[1] 7.3 10.6
```

Add a column

```
my.dataframe$N03 <- c(234.5, 123.4, 456.7, 567.8, 789.0)
my.dataframe
```

```

  site season  ph  N03
1    A winter 7.4 234.5
2    B summer 7.3 123.4
3    A summer 10.6 456.7
4    A spring 7.2 567.8
5    B  fall  8.9 789.0
```

Remove a column

```
#my.dataframe$N03<-NULL
my.dataframe <- my.dataframe[, -4]
my.dataframe
```

```

  site season  ph
1    A winter 7.4
2    B summer 7.3
3    A summer 10.6
4    A spring 7.2
5    B  fall  8.9
```

Structure of data frame:

```
str(my.dataframe)
```

```
'data.frame':  5 obs. of  3 variables:  
 $ site  : chr  "A" "B" "A" "A" ...  
 $ season: chr  "winter" "summer" "summer" "spring" ...  
 $ ph    : num  7.4 7.3 10.6 7.2 8.9
```

```
nrow(my.dataframe)
```

```
[1] 5
```

```
ncol(my.dataframe)
```

```
[1] 3
```

```
dim(my.dataframe)
```

```
[1] 5 3
```

```
#edit(my.dataframe) #this brings up a data editor
```

```
#View(my.dataframe) #this brings up a uneditable tab that display the data for you to view
```

```
names(my.dataframe)
```

```
[1] "site"  "season" "ph"
```

```
names(my.dataframe) <- c('area', 'season', 'P.h.')  
my.dataframe
```

```

      area season P.h.
1      A winter  7.4
2      B summer  7.3
3      A summer 10.6
4      A spring  7.2
5      B  fall  8.9

```

```

names(my.dataframe)[3] <- 'ph'
my.dataframe

```

```

      area season   ph
1      A winter  7.4
2      B summer  7.3
3      A summer 10.6
4      A spring  7.2
5      B  fall  8.9

```

Tibbles

- Tibbles are similar to data frame, but more convenient.
- Columns can be defined based on other columns already defined.
- Cannot convert categorical valued attributes to factors and cannot print an entire dataset.

```

install.packages("tibble")

```

Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)

```

library(tibble)

```

Create a Tibble

```

my.tibble <- tibble(TempCels = sample(-10:40, size=100, replace=TRUE),
                    TempFahr = TempCels*9/5+32,
                    Location = rep(letters[1:2], each=50))

my.tibble

```

```
# A tibble: 100 x 3
  TempCels TempFahr Location
  <int>     <dbl> <chr>
1      16      60.8 a
2      -5      23  a
3      31      87.8 a
4      -4      24.8 a
5       7      44.6 a
6      -3      26.6 a
7      12      53.6 a
8      25      77  a
9     -10      14  a
10     25      77  a
# i 90 more rows
```

```
install.packages("palmerpenguins")
```

Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)

```
library(palmerpenguins)
data(penguins)
dim(penguins)
```

```
[1] 344  8
```

```
class(penguins)
```

```
[1] "tbl_df"      "tbl"        "data.frame"
```

```
penguins
```

```
# A tibble: 344 x 8
  species island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
  <fct>   <fct>         <dbl>         <dbl>             <int>      <int>
1 Adelie  Torgu         182             43             190         3750
2 Adelie  Torgu         186             44             195         3800
3 Adelie  Torgu         185             44             193         3650
4 Adelie  Torgu         183             44             190         3600
5 Adelie  Torgu         182             43             190         3700
6 Adelie  Torgu         185             44             193         3750
7 Adelie  Torgu         183             44             190         3650
8 Adelie  Torgu         182             43             190         3700
9 Adelie  Torgu         185             44             193         3750
10 Adelie  Torgu         183             44             190         3650
# i 334 more rows
```

```

1 Adelie Torgersen      39.1      18.7      181      3750
2 Adelie Torgersen      39.5      17.4      186      3800
3 Adelie Torgersen      40.3       18      195      3250
4 Adelie Torgersen      NA        NA        NA        NA
5 Adelie Torgersen      36.7      19.3      193      3450
6 Adelie Torgersen      39.3      20.6      190      3650
7 Adelie Torgersen      38.9      17.8      181      3625
8 Adelie Torgersen      39.2      19.6      195      4675
9 Adelie Torgersen      34.1      18.1      193      3475
10 Adelie Torgersen      42        20.2      190      4250
# i 334 more rows
# i 2 more variables: sex <fct>, year <int>

```

Convert a data frame to a tibble

```

pe <-as_tibble(penguins)
class(pe)

```

```
[1] "tbl_df"      "tbl"        "data.frame"
```

```
pe
```

```

# A tibble: 344 x 8
  species island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
  <fct>   <fct>         <dbl>         <dbl>           <int>         <int>
1 Adelie Torgersen      39.1          18.7            181          3750
2 Adelie Torgersen      39.5          17.4            186          3800
3 Adelie Torgersen      40.3           18            195          3250
4 Adelie Torgersen      NA             NA             NA             NA
5 Adelie Torgersen      36.7          19.3            193          3450
6 Adelie Torgersen      39.3          20.6            190          3650
7 Adelie Torgersen      38.9          17.8            181          3625
8 Adelie Torgersen      39.2          19.6            195          4675
9 Adelie Torgersen      34.1          18.1            193          3475
10 Adelie Torgersen      42            20.2            190          4250
# i 334 more rows
# i 2 more variables: sex <fct>, year <int>

```

NOTE: print can be used to print the entire **pe** dataset.

```
x <- 1:16  
mode(x)
```

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

```
dim(x) <- c(4,4)  
class(x)
```

```
[1] "matrix" "array"
```

```
is.numeric(x)
```

```
[1] TRUE
```

```
mode(x) <- "character"  
mode(x)
```

```
[1] "character"
```

```
class(x)
```

```
[1] "matrix" "array"
```

Mode changed from 'numeric' to 'character', but class stays 'matrix' ... BUT

```
x <- factor(x)  
class(x)
```

```
[1] "factor"
```

```
mode(x)
```

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

Class changed from 'matrix' to 'factor', but mode stays 'numeric'

At this stage, even though x has mode numeric again, its new class, 'factor', prohibits it being used in arithmetic operations.

```
is.array(x)
```

```
[1] FALSE
```

```
is.list(x)
```

```
[1] FALSE
```

```
is.data.frame(x)
```

```
[1] FALSE
```

```
is.matrix(x)
```

```
[1] FALSE
```

```
is_tibble(x)
```

```
[1] FALSE
```

```
is.vector(x)
```

```
[1] FALSE
```

```
typeof(x)
```

```
[1] "integer"
```

Subsetting a tibble results in smaller tibble


```
class(pe[1:15, c("bill_length_mm", "bill_depth_mm")])
```

```
[1] "tbl_df"      "tbl"        "data.frame"
```

```
class(penguins[1:15, c("bill_length_mm", "bill_depth_mm")])
```

```
[1] "tbl_df"      "tbl"        "data.frame"
```

```
class(pe[1:15, c("bill_length_mm")])
```

```
[1] "tbl_df"      "tbl"        "data.frame"
```

```
class(penguins[1:15, c("bill_length_mm")])
```

```
[1] "tbl_df"      "tbl"        "data.frame"
```

Data wrangling cheatsheet

```
#dplyr
#install.packages("dplyr")
library(dplyr)
```

```
select(filter(pe, species=="Adelie"), bill_length_mm, bill_depth_mm)
```

```
# A tibble: 152 x 2
  bill_length_mm bill_depth_mm
      <dbl>         <dbl>
1         39.1          18.7
2         39.5          17.4
3         40.3           18
4          NA           NA
5         36.7          19.3
6         39.3          20.6
```

```

7           38.9           17.8
8           39.2           19.6
9           34.1           18.1
10          42             20.2
# i 142 more rows

```

```
filter(select(pe, bill_length_mm, bill_depth_mm, species), species=="Adelie")
```

```

# A tibble: 152 x 3
  bill_length_mm bill_depth_mm species
      <dbl>         <dbl> <fct>
1           39.1           18.7 Adelie
2           39.5           17.4 Adelie
3           40.3            18  Adelie
4            NA            NA  Adelie
5           36.7           19.3 Adelie
6           39.3           20.6 Adelie
7           38.9           17.8 Adelie
8           39.2           19.6 Adelie
9           34.1           18.1 Adelie
10          42             20.2 Adelie
# i 142 more rows

```

Exercise

How would you achieve the same result as the above but use tibble subsetting?

```
pe
```

```

# A tibble: 344 x 8
  species island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
  <fct>   <fct>         <dbl>         <dbl>         <int>         <int>
1 Adelie Torgersen      39.1           18.7           181           3750
2 Adelie Torgersen      39.5           17.4           186           3800
3 Adelie Torgersen      40.3            18           195           3250
4 Adelie Torgersen      NA            NA            NA            NA
5 Adelie Torgersen      36.7           19.3           193           3450
6 Adelie Torgersen      39.3           20.6           190           3650
7 Adelie Torgersen      38.9           17.8           181           3625

```

```

      8 Adelie  Torgersen          39.2          19.6          195          4675
      9 Adelie  Torgersen          34.1          18.1          193          3475
     10 Adelie  Torgersen          42           20.2          190          4250
# i 334 more rows
# i 2 more variables: sex <fct>, year <int>

```

```

pe[pe$species=='Adelie', c("bill_length_mm", "bill_depth_mm")]

```

```

# A tibble: 152 x 2
  bill_length_mm bill_depth_mm
      <dbl>         <dbl>
1         39.1         18.7
2         39.5         17.4
3         40.3          18
4          NA          NA
5         36.7         19.3
6         39.3         20.6
7         38.9         17.8
8         39.2         19.6
9         34.1         18.1
10        42          20.2
# i 142 more rows

```

```

subset(pe, pe$species=='Adelie', c("bill_length_mm", "bill_depth_mm"))

```

```

# A tibble: 152 x 2
  bill_length_mm bill_depth_mm
      <dbl>         <dbl>
1         39.1         18.7
2         39.5         17.4
3         40.3          18
4          NA          NA
5         36.7         19.3
6         39.3         20.6
7         38.9         17.8
8         39.2         19.6
9         34.1         18.1
10        42          20.2
# i 142 more rows

```

```
select(pe, bill_length_mm, bill_depth_mm, species) |> filter(species=="Adelie")
```

```
# A tibble: 152 x 3
  bill_length_mm bill_depth_mm species
      <dbl>         <dbl> <fct>
1         39.1         18.7 Adelie
2         39.5         17.4 Adelie
3         40.3          18  Adelie
4          NA          NA  Adelie
5         36.7         19.3 Adelie
6         39.3         20.6 Adelie
7         38.9         17.8 Adelie
8         39.2         19.6 Adelie
9         34.1         18.1 Adelie
10        42         20.2 Adelie
# i 142 more rows
```

Exercise

Pass the result from the filter to the select function and achieve the same result as shown above.

```
filter(pe, species=="Adelie") |> select(bill_length_mm, bill_depth_mm, species)
```

```
# A tibble: 152 x 3
  bill_length_mm bill_depth_mm species
      <dbl>         <dbl> <fct>
1         39.1         18.7 Adelie
2         39.5         17.4 Adelie
3         40.3          18  Adelie
4          NA          NA  Adelie
5         36.7         19.3 Adelie
6         39.3         20.6 Adelie
7         38.9         17.8 Adelie
8         39.2         19.6 Adelie
9         34.1         18.1 Adelie
10        42         20.2 Adelie
# i 142 more rows
```

Exercise

Create a data object to hold student names (Judy, Max, Dan) and their grades (78,85,99)
Convert number grades to letter grades:90-100:A;80-89:B;70-79:C; \<70:F

```
students <- list(names=c("Judy", "Max", "Dan"),  
                grades=c(78, 85, 99))  
print ("before:")
```

```
[1] "before:"
```

```
students
```

```
$names
```

```
[1] "Judy" "Max"  "Dan"
```

```
$grades
```

```
[1] 78 85 99
```

```
gradeConvertor<- function (grade){  
  grade = as.numeric(grade)  
  if(grade > 100 | grade < 0) print ("grade out of the range")  
  else if(grade >= 90 & grade <= 100) return ("A")  
  else if(grade >= 80 & grade < 90) return ("B")  
  else if(grade >= 70 & grade < 80) return ("C")  
  else return ("F")  
}  
  
#students$grades <-sapply(students$grades, gradeConvertor)  
  
for(i in 1:length(students$grades)){  
  students$grades[i] = gradeConvertor(students$grades[i])  
}  
  
print ("after:")
```

```
[1] "after:"
```

```
students
```

```
$names
```

```
[1] "Judy" "Max"  "Dan"
```

```
$grades
```

```
[1] "C" "B" "A"
```