

R tips

Contents

1	Set-up	1
2	Overview	2
3	Data structure (S3)	2
3.1	'character'	2
3.2	'numeric'	3
3.3	'data.frame'	3
3.4	'list'	4
4	Packages and Functions	5
4.1	Package	5
4.2	Function	5
5	Regex match	6
5.1	grep	6
5.2	stringr::str_extract	7
6	Common Packages	7
7	* S4: Classes and Methods	8
7.1	Classes	8
7.2	Methods	9
7.2.1	Demo (Not Run)	9
7.2.2	Demo (Run)	9

1 Set-up

```
devtools::load_all("~/utils.tool")
```

```
## i Loading utils.tool
```

```
## Warning in .recacheSubclasses(def@className, def, env): undefined subclass "ggraph" of class
```

```
## "gg.obj"; definition not updated
```

```
library(ggplot2)
# knitr::opts_chunk$set(echo = F)
```

2 Overview

```
x <- seq(1, 10, by = .05)
work2time <- data.frame(Time = x, Work = -x^2 + 100)
p <- ggplot(work2time) +
  geom_line(aes(x = Work, y = Time)) +
  theme_classic() +
  theme(text = element_text(family = "Times"))
p
```

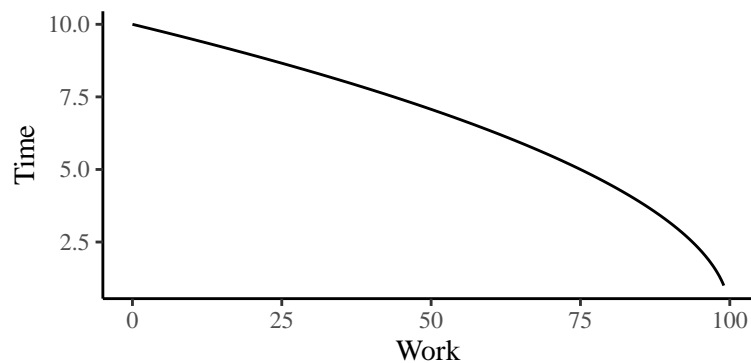


Figure 1: Learning Curve of R

3 Data structure (S3)

3.1 'character'

```
x <- "this is character"
y <- "this is \"character\""
print(x)
```

```
## [1] "this is character"
```

```
print(y)
```

```
## [1] "this is \"character\""
```

```
cat(x, "\n")
```

```
## this is character
```

```
cat(y, "\n")
```

```
## this is "character"
```

3.2 'numeric'

```
x <- 1
y <- 1:10
z <- seq(1, 10, by = .5)
x
```

```
## [1] 1
```

```
y
```

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

```
z
```

```
## [1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0 9.5 10.0
```

3.3 'data.frame'

```
data <- data.frame(x = 1:10, y = 10:1, z = rep("character", 10))
data
```

```
##      x  y      z
## 1    1 10 character
## 2    2  9 character
## 3    3  8 character
## 4    4  7 character
## 5    5  6 character
## 6    6  5 character
## 7    7  4 character
## 8    8  3 character
## 9    9  2 character
## 10 10  1 character
```

```
tibble::as_tibble(data)
```

```
## # A tibble: 10 x 3
##       x     y z
##   <int> <int> <chr>
## 1     1    10 character
## 2     2     9 character
## 3     3     8 character
## 4     4     7 character
## 5     5     6 character
## 6     6     5 character
## 7     7     4 character
## 8     8     3 character
```

```
## 9      9      2 character
## 10     10     1 character
```

```
## a inst data.frame
```

```
tibble::as_tibble(mtcars)
```

```
## # A tibble: 32 x 11
##      mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  21      6  160    110  3.9    2.62  16.5     0    1    4     4
##  2  21      6  160    110  3.9    2.88  17.0     0    1    4     4
##  3  22.8    4  108     93  3.85   2.32  18.6     1    1    4     1
##  4  21.4    6  258    110  3.08   3.22  19.4     1    0    3     1
##  5  18.7    8  360    175  3.15   3.44  17.0     0    0    3     2
##  6  18.1    6  225    105  2.76   3.46  20.2     1    0    3     1
##  7  14.3    8  360    245  3.21   3.57  15.8     0    0    3     4
##  8  24.4    4  147.    62  3.69   3.19  20      1    0    4     2
##  9  22.8    4  141.    95  3.92   3.15  22.9     1    0    4     2
## 10  19.2    6  168.   123  3.92   3.44  18.3     1    0    4     4
## # ... with 22 more rows
```

3.4 'list'

```
lst1 <- list(x = 1, y = 1:3)
```

```
lst1
```

```
## $x
```

```
## [1] 1
```

```
##
```

```
## $y
```

```
## [1] 1 2 3
```

```
lst2 <- list(
  x = 1:10,
  y = rep("character", 20), z = tibble::as_tibble(mtcars)
)
```

```
lst2
```

```
## $x
```

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

```
##
```

```
## $y
```

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

```
## [8] "character" "character" "character" "character" "character" "character" "character"
```

```
## [15] "character" "character" "character" "character" "character" "character"
```

```
##
## $z
## # A tibble: 32 x 11
##   mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  21     6  160   110  3.9   2.62  16.5    0   1    4    4
## 2  21     6  160   110  3.9   2.88  17.0    0   1    4    4
## 3 22.8    4  108    93  3.85  2.32  18.6    1   1    4    1
## 4 21.4    6  258   110  3.08  3.22  19.4    1   0    3    1
## 5 18.7    8  360   175  3.15  3.44  17.0    0   0    3    2
## 6 18.1    6  225   105  2.76  3.46  20.2    1   0    3    1
## 7 14.3    8  360   245  3.21  3.57  15.8    0   0    3    4
## 8 24.4    4  147    62  3.69  3.19  20      1   0    4    2
## 9 22.8    4  141    95  3.92  3.15  22.9    1   0    4    2
## 10 19.2    6  168   123  3.92  3.44  18.3    1   0    4    4
## # ... with 22 more rows
```

4 Packages and Functions

4.1 Package

4.2 Function

```
fun <- function(x = 1, y = 2) {
  x + y
}
res <- fun()
res
```

```
## [1] 3
```

```
fun2 <- function(x = seq(0.01, .99, length.out = 100)) {
  df <- data.frame(
    x = rep(x, 2),
    y = c(qlogis(x), 2 * qlogis(x)),
    group = rep(c("a", "b"),
      each = 100)
  )
  p <- ggplot(df, aes(x=x, y=y, group=group))
  # These work
  p + geom_line(linetype = 2)
}
p <- fun2()
```

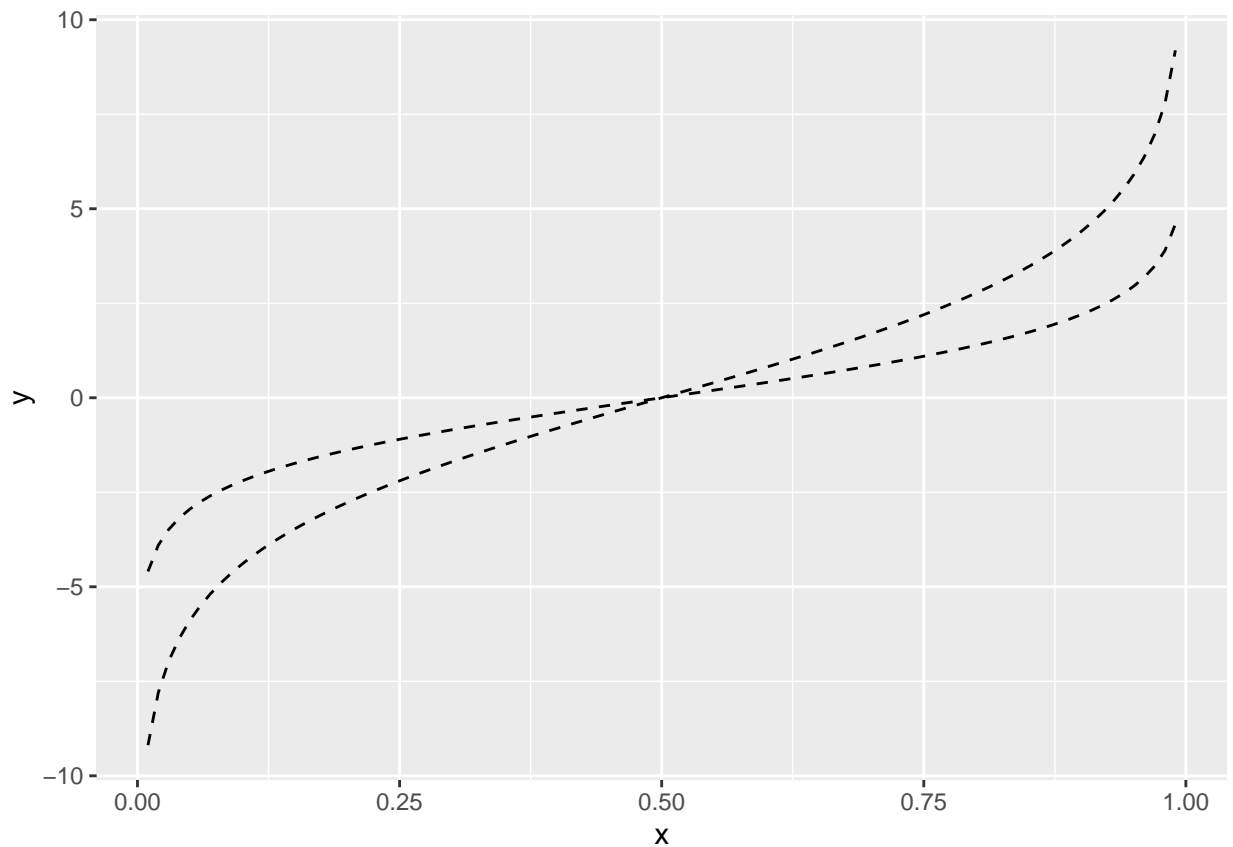


Figure 2: Demo figure

5 Regex match

5.1 grep

```
letters
```

```
## [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "w"
## [24] "x" "y" "z"
```

```
grep("[a-z]", letters)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
```

```
txt <- c("arm","foot","lefroo", "bafoobar")
```

```
if(length(i <- grep("foo", txt)))
```

```
  cat("'foo' appears at least once in\n\t", txt, "\n")
```

```
## 'foo' appears at least once in
```

```
##   arm foot lefroo bafoobar
```

```
i
```

```
## [1] 2 4
```

```
txt[i]
```

```
## [1] "foot"      "bafoobar"
```

5.2 stringr::str_extract

```
shopping_list <- c("apples x4", "bag of flour", "bag of sugar", "milk x2")  
stringr::str_extract(shopping_list, "\\d")
```

```
## [1] "4" NA NA "2"
```

```
stringr::str_extract(shopping_list, "[a-z]+")
```

```
## [1] "apples" "bag"      "bag"      "milk"
```

```
stringr::str_extract(shopping_list, "[a-z]{1,4}")
```

```
## [1] "appl" "bag"     "bag"     "milk"
```

```
stringr::str_extract(shopping_list, "\\b[a-z]{1,4}\\b")
```

```
## [1] NA      "bag"   "bag"   "milk"
```

6 Common Packages

```
data <- tibble::tribble(  
  ~ Name, ~ Description, ~ Function,  
  "base", "data", "data.frame, c, list, ...",  
  "base", "Expression", "if, else ...",  
  "base", "String", "paste0, paste, print, cat, ...",  
  "base", "Match string", "grep, grepl, sub, gsub ...",  
  "base", "Loop", "for, lapply, apply, mapply ...",  
  "data.table", "For fast read and write table", "fread, fwrite",  
  "dplyr", "Modify table", "select, filter, arrange, distinct, slice, mutate ...",  
  "ggplot2", "Visualization", "...",  
  "stringr", "Match strings", "str_extract ..."  
)  
data <- dplyr::relocate(data, Name, Function)  
pretty_flex2(data, "Common Packages and Functions",  
  weight = c(  
    Description = 1.5, Name = .7))
```

Table 1: Common Packages and Functions

Name	Function	Description
base	data.frame, c, list, ...	data
base	if, else ...	Expression
base	paste0, paste, print, cat, ...	String
base	grep, grepl, sub, gsub ...	Match string
base	for, lapply, apply, mapply ...	Loop
data.table	fread, fwrite	For fast read and write table
dplyr	select, filter, arrange, distinct, slice, mutate ...	Modify table
ggplot2	...	Visualization
stringr	str_extract ...	Match strings

7 * S4: Classes and Methods

7.1 Classes

```
library(MCnebula2)
mcn <- mcnebula()
slotNames(mcn)
```

```
## [1] "creation_time"      "ion_mode"           "melody"             "mcn_dataset"
## [5] "statistic_set"      "project_version"    "project_path"       "project_conformation"
## [9] "project_metadata"   "project_api"        "project_dataset"    "parent_nebula"
## [13] "child_nebulae"      "export_path"        "export_name"
```

```
mcn@mcn_dataset
```

```
## An object of class "mcn_dataset"
## Slot "dataset":
## list()
##
## Slot "reference":
## list()
##
## Slot "backtrack":
## list()
```



```
mcn_dataset(mcn)
```

```
## An object of class "mcn_dataset"  
## Slot "dataset":  
## list()  
##  
## Slot "reference":  
## list()  
##  
## Slot "backtrack":  
## list()
```

7.2 Methods

7.2.1 Demo (Not Run)

```
mcn <- mcn_5features  
mcn1 <- filter_structure(mcn)  
mcn1 <- create_reference(mcn1)  
mcn1 <- filter_formula(mcn1, by_reference = T)  
mcn1 <- create_stardust_classes(mcn1)  
mcn1 <- create_features_annotation(mcn1)  
mcn1 <- cross_filter_stardust(mcn1, 2, 1)  
mcn1 <- create_nebula_index(mcn1)  
mcn1 <- compute_spectral_similarity(mcn1)  
mcn1 <- create_parent_nebula(mcn1, 0.01)  
mcn1 <- create_child_nebulae(mcn1, 0.01)  
mcn1 <- create_parent_layout(mcn1)  
mcn1 <- create_child_layouts(mcn1)  
mcn1 <- activate_nebulae(mcn1)  
  
## optional Child-Nebulae  
visualize(mcn1)  
  
visualize(mcn1, "parent")  
visualize(mcn1, 1)  
visualize_all(mcn1)
```

7.2.2 Demo (Run)

```
mcn <- mcn_5features  
mcn1 <- filter_structure(mcn)
```

```

## [INFO] MCnebula2: filter_structure

## ## msframe: filter_msframe group_by: ~ .features_id
mcn1 <- create_reference(mcn1)

## [INFO] MCnebula2: create_reference
## ## create_reference: fill == T
## filling missing features with filtered formula
## [INFO] MCnebula2: filter_formula
## ## msframe: filter_msframe group_by: ~ .features_id
mcn1 <- filter_formula(mcn1, by_reference = T)

## [INFO] MCnebula2: filter_formula
## ## filter_formula: by_reference == T
## case formula, ignore `fun_filter`
mcn1 <- create_stardust_classes(mcn1)

## [INFO] MCnebula2: create_stardust_classes
## [INFO] MCnebula2: filter_ppcp
## ## filter_ppcp: by_reference == T
## ## filter_ppcp: validate annotation data .canopus >>> .f3_canopus
## ## msframe: filter_msframe group_by: ~ paste0(.features_id, "_", .candidates_id)
mcn1 <- create_features_annotation(mcn1)

## [INFO] MCnebula2: create_features_annotation
mcn1 <- cross_filter_stardust(mcn1, 2, 1)

## [INFO] MCnebula2: cross_filter_stardust
## ## cross_filter_stardust: quantity
## ## cross_filter_stardust: score
## ## cross_filter_stardust: identical
mcn1 <- create_nebula_index(mcn1)

## [INFO] MCnebula2: create_nebula_index
mcn1 <- compute_spectral_similarity(mcn1)

## [INFO] MCnebula2: compute_spectral_similarity
## ## compute_spectral_similarity: compareSpectra

```

```

mcn1 <- create_parent_nebula(mcn1, 0.01)

## [INFO] MCnebula2: create_parent_nebula
mcn1 <- create_child_nebulae(mcn1, 0.01)

## [INFO] MCnebula2: create_child_nebulae
mcn1 <- create_parent_layout(mcn1)

## [INFO] MCnebula2: create_parent_layout
mcn1 <- create_child_layouts(mcn1)

## [INFO] MCnebula2: create_child_layouts
mcn1 <- activate_nebulae(mcn1)

## [INFO] MCnebula2: activate_nebulae
## optional Child-Nebulae
visualize(mcn1)

## [INFO] MCnebula2: visualize

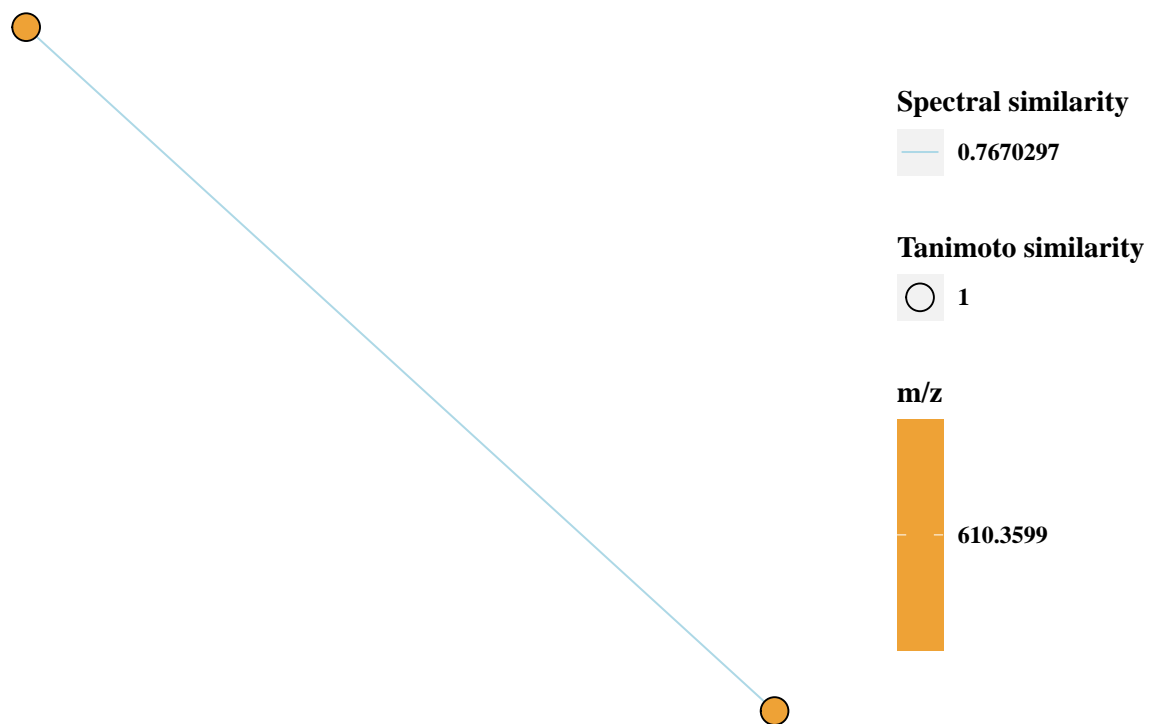
## Specify item as following to visualize:

## # A tibble: 18 x 3
##       seq hierarchy class.name
##   <int>    <dbl> <chr>
## 1     1      5 Amino acids and derivatives
## 2     2      4 Amino acids, peptides, and analogues
## 3     3      2 Benzenoids
## 4     4      4 Carbonyl compounds
## 5     5      5 Carboxylic acid amides
## 6     6      4 Carboxylic acid derivatives
## 7     7      3 Carboxylic acids and derivatives
## 8     8      3 Heteroaromatic compounds
## 9     9      4 Indoles
## 10    10      5 Ketones
## 11    11      3 Lactams
## 12    12      3 Macrolactams
## 13    13      2 Organic acids and derivatives
## 14    14      3 Organic oxides
## 15    15      5 Peptides
## 16    16      2 Phenylpropanoids and polyketides
## 17    17      3 Pyrroles
## 18    18      4 Substituted pyrroles

```

```
visualize(mcn1, "parent")
```

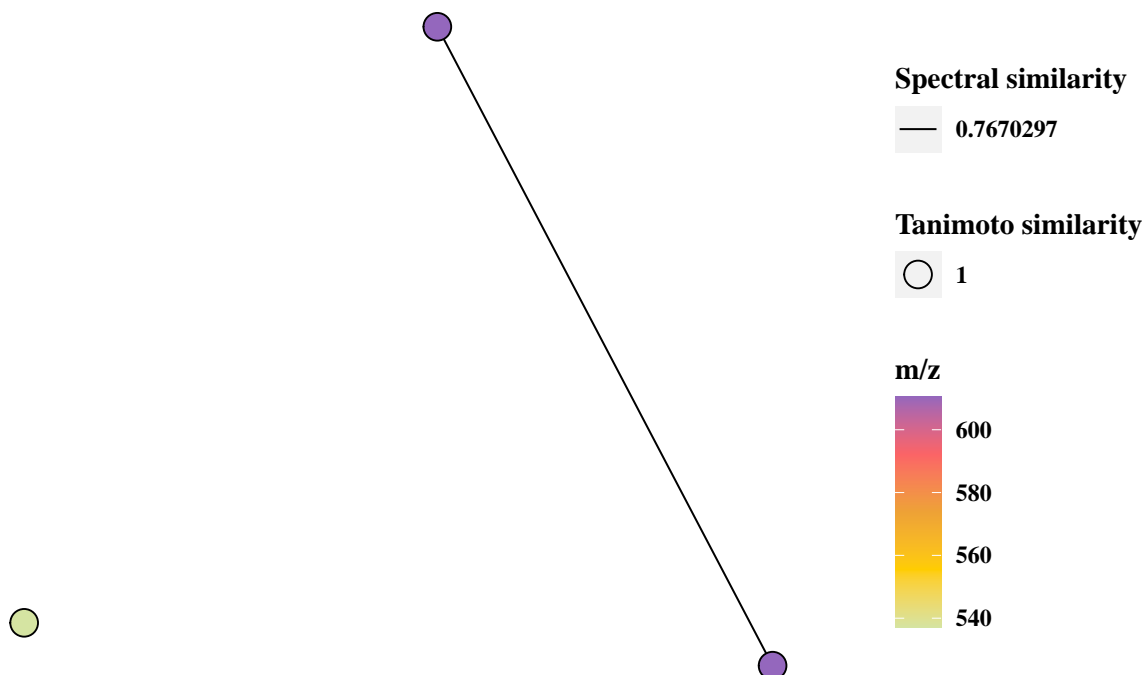
```
## [INFO] MCnebula2: visualize
```



```
visualize(mcn1, 1)
```

```
## [INFO] MCnebula2: visualize
```

Amino acids and derivatives



```
visualize_all(mcn1)
```

```
## [INFO] MCnebula2: visualize_all
## ## BEGIN: current.viewport:
##   viewport[ROOT]
## ## info: current.viewport:
##   viewport[GRID.VP.3135]
## ## info: current.viewport:
##   viewport[legend_hierarchy]
## ## info: current.viewport:
##   viewport[sub_panel]
## ## info: current.viewport:
##   viewport[ROOT]
## ## visualize: legend:
##   extract legend from `ggset(child_nebulae(x))[[1]]` (nebula names:).
##   In default, legend scales have been unified for all child-nebulae.
## ## END: current.viewport:
##   viewport[GRID.VP.3136]
```

