

## start\_\_2\_R\_basics

In R, an object is anything that can be assigned to a variable. This includes constants, data structures, functions, and even graphs. Objects have a mode (which describes how the object is stored) and a class (which tells generic functions like print how to handle it).

Vectors are one-dimensional arrays that can hold numeric data, character data, or logical data. The combine function `c()` is used to form the vector.

```
x <- c(1,2,4) #The "c" stands for "concatenate."
x
```

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

```
q <- c(x,x,8)
q
```

```
## [1] 1 2 4 1 2 4 8
```

```
#
a <- c(1, 2, 5, 3, 6, -2, 4)
b <- c("one", "two", "three")
c <- c(TRUE, TRUE, TRUE, FALSE, TRUE, FALSE)
```

Seen Above -- a is numeric vector, b is a character vector, and c is a logical vector. Note that the data in a vector must only be one type or mode (numeric, character, or logical). You can't mix modes in the same vector.

Indexing vectors -

```
#
a_idx_1_3 <- a[c(1:3)]
```

```
a_idx_1_3 <- a[c(1:3)]
a_idx_1_3
```

```
## [1] 1 2 5
print(a_idx_1_3)
```

```
## [1] 1 2 5
# x <- c(1,3,4) #The "c" stands for "concatenate."
# x
# q <- c(x,x,8)
# q
```

A matrix is a two-dimensional array where each element has the same mode (numeric, character, or logical). Matrices are created with the matrix function

```
#
The option byrow indicates whether the matrix should be filled in
by row ( byrow=TRUE ) or by column ( byrow=FALSE ). The default is by column.
```

```
myymatrix <- matrix(vector, nrow=number_of_rows, ncol=number_of_columns,
byrow=logical_value, dimnames=list(
char_vector_rownames, char_vector_colnames))
```

```
matrx_y <- matrix(1:20, nrow=5, ncol=4)
matrx_y
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    6   11   16
## [2,]    2    7   12   17
## [3,]    3    8   13   18
## [4,]    4    9   14   19
## [5,]    5   10   15   20
```

## Read CSV

```
library(readr)
df_iris <- read_csv("~/temp/11_22/#Rstats/RStudio_Nov22/git_up/rstats_nov22/iris.csv")
```

```
## Rows: 150 Columns: 5
## -- Column specification -----
## Delimiter: ","
## chr (1): Species
## dbl (4): Sepal.Length, Sepal.Width, Petal.Length, Petal.Width
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df_iris_1 <- read_csv("~/temp/11_22/#Rstats/RStudio_Nov22/git_up/rstats_nov22/iris_1.csv") ## Dummy IRI

## Rows: 150 Columns: 5
## -- Column specification -----
## Delimiter: ","
## chr (1): Species
## dbl (4): Sepal.Length, Sepal.Width, Petal.Length, Petal.Width
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
View(df_iris)
```

## Get DF Cols

```
print(df_iris$Species)

##      [1] "setosa"      "setosa"      "setosa"      "setosa"      "setosa"
##      [6] "setosa"      "setosa"      "setosa"      "setosa"      "setosa"
##     [11] "setosa"      "setosa"      "setosa"      "setosa"      "setosa"
##     [16] "setosa"      "setosa"      "setosa"      "setosa"      "setosa"
##     [21] "setosa"      "setosa"      "setosa"      "setosa"      "setosa"
##     [26] "setosa"      "setosa"      "setosa"      "setosa"      "setosa"
##     [31] "setosa"      "setosa"      "setosa"      "setosa"      "setosa"
##     [36] "setosa"      "setosa"      "setosa"      "setosa"      "setosa"
##     [41] "setosa"      "setosa"      "setosa"      "setosa"      "setosa"
##     [46] "setosa"      "setosa"      "setosa"      "setosa"      "setosa"
##     [51] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
##     [56] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
##     [61] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
##     [66] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
##     [71] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
##     [76] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
##     [81] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
##     [86] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
##     [91] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
##     [96] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
##    [101] "virginica"   "virginica"   "virginica"   "virginica"   "virginica"
##    [106] "virginica"   "virginica"   "virginica"   "virginica"   "virginica"
##    [111] "virginica"   "virginica"   "virginica"   "virginica"   "virginica"
##    [116] "virginica"   "virginica"   "virginica"   "virginica"   "virginica"
##    [121] "virginica"   "virginica"   "virginica"   "virginica"   "virginica"
##    [126] "virginica"   "virginica"   "virginica"   "virginica"   "virginica"
##    [131] "virginica"   "virginica"   "virginica"   "virginica"   "virginica"
##    [136] "virginica"   "virginica"   "virginica"   "virginica"   "virginica"
##    [141] "virginica"   "virginica"   "virginica"   "virginica"   "virginica"
##    [146] "virginica"   "virginica"   "virginica"   "virginica"   "virginica"

vector_species <- df_iris$Species
print(class(vector_species))
```

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

## Check Duplicates

```
print(class(vector_species))
```

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

```
duplicated(vector_species)
```

```
## [1] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [13] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [25] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [37] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [49] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [61] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [73] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [85] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [97] TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [109] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [121] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [133] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [145] TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
print(length(vector_species))#150
```

```
## [1] 150
```

```
sum(duplicated(vector_species))
```

```
## [1] 147
```

```
# 147 - as the 1st Occurences are Not considered as DUPLICATES
```

```
# print(nrow(vector_species))#NULL -- Its a VECTOR NOT a DF
```

```
# print(ncol(vector_species)) # NULL-- Its a VECTOR NOT a DF
```

## Merge 2 Data Sets

```
# df_merged_iris <- merge(df_iris,df_iris_1,by="Species",all.x=TRUE,all.y=TRUE)
```

```
# #View(df_merged_iris)
```

```
# print(nrow(df_merged_iris))## nrow
```

```
# print(ncol(df_merged_iris)) ## ncol
```

## Order Data in a DATA FRAME

```

#order_iris <- df_iris[order[df_iris$Species]]
#Error in order[df_iris$Species] :
#object of type 'closure' is not subsettable

order_iris <- df_iris[order(df_iris$Species),]
order_iris_1 <- df_iris[order(df_iris$Species , df_iris$Sepal.Length),]
#Sepal.Length,
order_iris_2_ascending <- df_iris[order(df_iris$Species , df_iris$Sepal.Width),]
order_iris_2_decending <- df_iris[order(df_iris$Species , - df_iris$Sepal.Width),]
#Sepal.Width,
#
print(class(order_iris)) #[1] "spec_tbl_df" "tbl_df"      "tbl"      "data.frame"

## [1] "tbl_df"      "tbl"      "data.frame"
print(head(order_iris))

## # A tibble: 6 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##         <dbl>      <dbl>      <dbl>      <dbl> <chr>
## 1         5.1         3.5         1.4         0.2 setosa
## 2         4.9         3         1.4         0.2 setosa
## 3         4.7         3.2         1.3         0.2 setosa
## 4         4.6         3.1         1.5         0.2 setosa
## 5          5         3.6         1.4         0.2 setosa
## 6         5.4         3.9         1.7         0.4 setosa

print(tail(order_iris))

## # A tibble: 6 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##         <dbl>      <dbl>      <dbl>      <dbl> <chr>
## 1         6.7         3.3         5.7         2.5 virginica
## 2         6.7         3         5.2         2.3 virginica
## 3         6.3         2.5         5         1.9 virginica
## 4         6.5         3         5.2         2   virginica
## 5         6.2         3.4         5.4         2.3 virginica
## 6         5.9         3         5.1         1.8 virginica

#
print(head(order_iris_1)) ###Sepal.Length,

## # A tibble: 6 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##         <dbl>      <dbl>      <dbl>      <dbl> <chr>
## 1         4.3         3         1.1         0.1 setosa
## 2         4.4         2.9         1.4         0.2 setosa
## 3         4.4         3         1.3         0.2 setosa
## 4         4.4         3.2         1.3         0.2 setosa
## 5         4.5         2.3         1.3         0.3 setosa
## 6         4.6         3.1         1.5         0.2 setosa

print(tail(order_iris_1)) ###Sepal.Length,

```

```
## # A tibble: 6 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##         <dbl>         <dbl>         <dbl>         <dbl> <chr>
## 1         7.6           3           6.6           2.1 virginica
## 2         7.7           3.8          6.7           2.2 virginica
## 3         7.7           2.6          6.9           2.3 virginica
## 4         7.7           2.8          6.7           2   virginica
## 5         7.7           3           6.1           2.3 virginica
## 6         7.9           3.8          6.4           2   virginica

#
print(head(order_iris_2_decending)) ##Sepal.Width,

## # A tibble: 6 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##         <dbl>         <dbl>         <dbl>         <dbl> <chr>
## 1         5.7           4.4           1.5           0.4 setosa
## 2         5.5           4.2           1.4           0.2 setosa
## 3         5.2           4.1           1.5           0.1 setosa
## 4         5.8           4           1.2           0.2 setosa
## 5         5.4           3.9           1.7           0.4 setosa
## 6         5.4           3.9           1.3           0.4 setosa

print(tail(order_iris_2_decending)) ##Sepal.Width,

## # A tibble: 6 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##         <dbl>         <dbl>         <dbl>         <dbl> <chr>
## 1         6.1           2.6           5.6           1.4 virginica
## 2         4.9           2.5           4.5           1.7 virginica
## 3         6.7           2.5           5.8           1.8 virginica
## 4         5.7           2.5           5           2   virginica
## 5         6.3           2.5           5           1.9 virginica
## 6         6           2.2           5           1.5 virginica

# TODO - Petal.Length, Petal.Width
# print(nrow(df_merged_iris))## nrow
# print(ncol(df_merged_iris)) ## ncol
```

## AS TIBBLE

- [https://github.com/tidyverse/tibble/blob/HEAD/R/as\\_tibble.R](https://github.com/tidyverse/tibble/blob/HEAD/R/as_tibble.R)
- [https://tibble.tidyverse.org/reference/as\\_tibble.html](https://tibble.tidyverse.org/reference/as_tibble.html)

`remotes::install_github("thinkr-open/golem")`

```
library(dplyr)
```

```
##
## 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

tibble_order_iris_2_decending <- as_tibble(order_iris_2_decending)
print(head(tibble_order_iris_2_decending))

## # A tibble: 6 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##         <dbl>      <dbl>      <dbl>      <dbl>   <chr>
## 1         5.7         4.4         1.5         0.4 setosa
## 2         5.5         4.2         1.4         0.2 setosa
## 3         5.2         4.1         1.5         0.1 setosa
## 4         5.8         4         1.2         0.2 setosa
## 5         5.4         3.9         1.7         0.4 setosa
## 6         5.4         3.9         1.3         0.4 setosa

library(dplyr)
tibble_order_iris_2_decending <- as_tibble(order_iris_2_decending)
print(head(tibble_order_iris_2_decending))

## # A tibble: 6 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##         <dbl>      <dbl>      <dbl>      <dbl>   <chr>
## 1         5.7         4.4         1.5         0.4 setosa
## 2         5.5         4.2         1.4         0.2 setosa
## 3         5.2         4.1         1.5         0.1 setosa
## 4         5.8         4         1.2         0.2 setosa
## 5         5.4         3.9         1.7         0.4 setosa
## 6         5.4         3.9         1.3         0.4 setosa

tibble_order_iris_2_ascending <- as_tibble(order_iris_2_ascending)
print(head(tibble_order_iris_2_ascending))

## # A tibble: 6 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##         <dbl>      <dbl>      <dbl>      <dbl>   <chr>
## 1         4.5         2.3         1.3         0.3 setosa
## 2         4.4         2.9         1.4         0.2 setosa
## 3         4.9         3         1.4         0.2 setosa
## 4         4.8         3         1.4         0.1 setosa
## 5         4.3         3         1.1         0.1 setosa
## 6         5         3         1.6         0.2 setosa
```

**Horizontal Join** – two data frames are joined by one or more common key variables (that is an inner join). #

**Import CSV from WWW**

```
# #https://www.rdocumentation.org/packages/utils/versions/3.6.2/topics/read.table
# www <- "http://www.massey.ac.nz/~pscower/ts/cbe.dat"
# CBE <- read.table(www, header = T)
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.