

Data Import

with `readr`, `tibble`, and `tidyR`

Cheat Sheet



R's **tidyverse** is built around **tidy data** stored in **tibbles**, an enhanced version of a data frame.

The front side of this sheet shows how to read text files into R with `readr`.

The reverse side shows how to create tibbles with `tibble` and to layout tidy data with `tidyR`.

Other types of data

Try one of the following packages to import other types of files

- `haven` - SPSS, Stata, and SAS files
- `readxl` - excel files (.xls and .xlsx)
- `DBI` - databases
- `jsonlite` - json
- `xml2` - XML
- `httr` - Web APIs
- `rvest` - HTML (Web Scraping)

Write functions

Save `x`, an R object, to `path`, a file path, with:

`write_csv(x, path, na = "NA", append = FALSE, col_names = !append)`

Tibble/df to comma delimited file.

`write_delim(x, path, delim = " ", na = "NA", append = FALSE, col_names = !append)`

Tibble/df to file with any delimiter.

`write_excel_csv(x, path, na = "NA", append = FALSE, col_names = !append)`

Tibble/df to a CSV for excel

`write_file(x, path, append = FALSE)`

String to file.

`write_lines(x, path, na = "NA", append = FALSE)`

String vector to file, one element per line.

`write_rds(x, path, compress = c("none", "gz", "bz2", "xz", ...))`

Object to RDS file.

`write_tsv(x, path, na = "NA", append = FALSE, col_names = !append)`

Tibble/df to tab delimited files.

Read functions

Read tabular data to tibbles

These functions share the common arguments:

```
read_*(file, col_names = TRUE, col_types = NULL, locale = default_locale(), na = c("", "NA"), quoted_na = TRUE, comment = "", trim_ws = TRUE, skip = 0, n_max = Inf, guess_max = min(1000, n_max), progress = interactive())
```

| |
|--------------------------|
| a,b,c 1,2,3 4,5,NA |
| |

| |
|--------|
| A B C |
| 1 2 3 |
| 4 5 NA |

`read_csv()`

Reads comma delimited files.
`read_csv("file.csv")`

| |
|--------------------------|
| a;b;c 1;2;3 4;5;NA |
| |

| |
|--------|
| A B C |
| 1 2 3 |
| 4 5 NA |

`read_csv2()`

Reads Semi-colon delimited files.
`read_csv2("file2.csv")`

| |
|--------------------------|
| a b c 1 2 3 4 5 NA |
| |

| |
|--------|
| A B C |
| 1 2 3 |
| 4 5 NA |

`read_delim()`

(delim, quote = "\\"", escape_backslash = FALSE, escape_double = TRUE) Reads files with any delimiter.
`read_delim("file.txt", delim = "|")`

| |
|--------------------------|
| a b c 1 2 3 4 5 NA |
| |

| |
|--------|
| A B C |
| 1 2 3 |
| 4 5 NA |

`read_fwf(col_positions)`

Reads fixed width files.
`read_fwf("file.fwf", col_positions = c(1, 3, 5))`

`read_tsv()`

Reads tab delimited files. Also `read_table()`.
`read_tsv("file.tsv")`

Useful arguments

| |
|--------------------------|
| a,b,c 1,2,3 4,5,NA |
| |

| |
|--------|
| 1 2 3 |
| 4 5 NA |
| |

Example file

```
write_csv(path = "file.csv",  
          x = read_csv("a,b,c\n1,2,3\n4,5,NA"))
```

| |
|--------|
| 1 2 3 |
| 4 5 NA |
| |

Skip lines

```
read_csv("file.csv",  
        skip = 1)
```

| |
|--------|
| A B C |
| 1 2 3 |
| 4 5 NA |

Read in a subset

```
read_csv("file.csv",  
        n_max = 1)
```

| |
|----------|
| A B C |
| 1 2 3 |
| NA NA NA |

Missing Values

```
read_csv("file.csv",  
        na = c("4", "5", "?"))
```

| |
|--------|
| x y z |
| A B C |
| 1 2 3 |
| 4 5 NA |

| |
|----------|
| A B C |
| 1 2 3 |
| NA NA NA |

Read non-tabular data

`read_file(file, locale = default_locale())`

Read a file into a single string.

`read_file_raw(file)`

Read a file into a raw vector.

`read_lines(file, skip = 0, n_max = -1L, locale = default_locale(), na = character(), progress = interactive())`

Read each line into its own string.

`read_lines_raw(file, skip = 0, n_max = -1L, progress = interactive())`

Read each line into a raw vector.

`read_log(file, col_names = FALSE, col_types = NULL, skip = 0, n_max = -1, progress = interactive())`

Apache style log files.

Parsing data types

readr functions guess the types of each column and convert types when appropriate (but will NOT convert strings to factors automatically).

A message shows the type of each column in the result.

```
## Parsed with column specification:  
## cols(  
##   age = col_integer(),  
##   sex = col_character(),  
##   earn = col_double()  
## )
```

age is an integer
sex is a character
earn is a double (numeric)

1. Use `problems()` to diagnose problems
`x <- read_csv("file.csv"); problems(x)`

2. Use a `col_` function to guide parsing

- `col_guess()` - the default
- `col_character()`
- `col_double()`
- `col_euro_double()`
- `col_datetime(format = "")` Also `col_date(format = "")` and `col_time(format = "")`
- `col_factor(levels, ordered = FALSE)`
- `col_integer()`
- `col_logical()`
- `col_number()`
- `col_numeric()`
- `col_skip()`

```
x <- read_csv("file.csv", col_types = cols(  
  A = col_double(),  
  B = col_logical(),  
  C = col_factor()  
)
```

3. Else, read in as character vectors then parse with a `parse_` function.

- `parse_guess(x, na = c("", "NA"), locale = default_locale())`
 - `parse_character(x, na = c("", "NA"), locale = default_locale())`
 - `parse_datetime(x, format = "", na = c("", "NA"), locale = default_locale())` Also `parse_date()` and `parse_time()`
 - `parse_double(x, na = c("", "NA"), locale = default_locale())`
 - `parse_factor(x, levels, ordered = FALSE, na = c("", "NA"), locale = default_locale())`
 - `parse_integer(x, na = c("", "NA"), locale = default_locale())`
 - `parse_logical(x, na = c("", "NA"), locale = default_locale())`
 - `parse_number(x, na = c("", "NA"), locale = default_locale())`
- `x$A <- parse_number(x$A)`

Tibbles - an enhanced data frame

The **tibble** package provides a new S3 class for storing tabular data, the tibble. Tibbles inherit the data frame class, but improve two behaviors:

- Display** - When you print a tibble, R provides a concise view of the data that fits on one screen.
- Subsetting** - always returns a new tibble, `[[` and `$` always return a vector.
- No partial matching** - You must use full column names when subsetting

| # tibble: 234 x 6 | manufacturer | model | displ | <chr> | <dbl> |
|--|--------------|------------|-------|------------|-------|
| 1 | audi | a4 | 1.8 | audi | 1.8 |
| 2 | audi | a4 | 1.8 | a4 | 1.8 |
| 3 | audi | a4 | 2.0 | a4 | 2.0 |
| 4 | audi | a4 | 2.0 | a4 | 2.0 |
| 5 | audi | a4 | 2.8 | a4 | 2.8 |
| 6 | audi | a4 | 2.8 | a4 | 2.8 |
| 7 | audi | a4 | 3.1 | a4 | 3.1 |
| 8 | audi | a4 quattro | 1.8 | a4 quattro | 1.8 |
| 9 | audi | a4 quattro | 1.8 | a4 quattro | 1.8 |
| 10 | audi | a4 quattro | 2.0 | a4 quattro | 2.0 |
| # ... with 224 more rows, and 3 variables: | | | | | |
| # more variables: year <int>, cyl <dbl>, trans <chr> | | | | | |

tibble display

| | | | |
|--|------|---|------------|
| 156 | 1999 | 6 | auto(l4) |
| 157 | 1999 | 6 | auto(l4) |
| 158 | 2008 | 6 | auto(l4) |
| 159 | 2008 | 8 | auto(s4) |
| 160 | 1999 | 4 | manual(m5) |
| 161 | 1999 | 4 | auto(l4) |
| 162 | 2008 | 4 | manual(m5) |
| 163 | 2008 | 4 | manual(m5) |
| 164 | 2008 | 4 | auto(l4) |
| 165 | 1999 | 4 | auto(l4) |
| 166 | 1999 | 4 | auto(l4) |
| [I reached getOption("max.print") - omitted 68 rows] | | | |

data frame display

- Control the default appearance with options:
`options(tibble.print_max = n, tibble.print_min = m, tibble.width = Inf)`
- View entire data set with `View(x, title)` or `glimpse(x, width = NULL, ...)`
- Revert to data frame with `as.data.frame()` (required for some older packages)

Construct a tibble in two ways

tibble(...)

Construct by columns.

```
tibble(x = 1:3,
      y = c("a", "b", "c"))
```

tribble(...)

Construct by rows.

```
tribble(
```

```
~x, ~y,
1, "a",
2, "b",
3, "c")
```

`as_tibble(x, ...)` Convert data frame to tibble.

`enframe(x, name = "name", value = "value")`

Converts named vector to a tibble with a `names` column and a `values` column.

`is_tibble(x)` Test whether x is a tibble.

Both make this tibble

drop_na(data, ...)

Drop rows containing NA's in ... columns.

| x1 | x2 | x1 | x2 |
|----|----|----|----|
| A | 1 | A | 1 |
| B | NA | D | 3 |
| C | NA | | |
| D | 3 | | |
| E | NA | | |

`drop_na(x, x2)`

fill(data, ..., direction = c("down", "up"))

Fill in NA's in ... columns with most recent non-NA values.

| x1 | x2 | x1 | x2 |
|----|----|----|----|
| A | 1 | A | 1 |
| B | NA | B | 1 |
| C | NA | C | 1 |
| D | 3 | D | 3 |
| E | NA | E | 3 |

`fill(x, x2)`

replace_na(data, ..., replace = list(...))

Replace NA's by column.

| x1 | x2 | x1 | x2 |
|----|----|----|----|
| A | 1 | A | 1 |
| B | NA | B | 2 |
| C | NA | C | 2 |
| D | 3 | D | 3 |
| E | NA | E | 2 |

`replace_na(x, list(x2 = 2), x2)`

Expand Tables - quickly create tables with combinations of values

complete(data, ..., fill = list())

Adds to the data missing combinations of the values of the variables listed in ...

`complete(mtcars, cyl, gear, carb)`

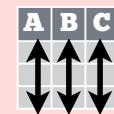
expand(data, ...)

Create new tibble with all possible combinations of the values of the variables listed in ...

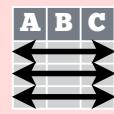
`expand(mtcars, cyl, gear, carb)`

Tidy Data with tidyR

Tidy data is a way to organize tabular data. It provides a consistent data structure across packages. A table is tidy if:



Each **variable** is in its own **column**



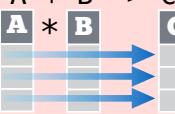
Each **observation**, or **case**, is in its own **row**

Tidy data:



Makes variables easy to access as vectors

$A * B \rightarrow C$



Preserves cases during vectorized operations

Split and Combine Cells

Use these functions to split or combine cells into individual, isolated values.

separate(data, col, into, sep = "[^[:alnum:]]+", remove = TRUE, convert = FALSE, extra = "warn", fill = "warn", ...)

Separate each cell in a column to make several columns.

table3

| country | year | rate | country | year | cases | pop |
|---------|------|----------|---------|------|-------|------|
| A | 1999 | 0.7K/19M | A | 1999 | 0.7K | 19M |
| A | 2000 | 2K/20M | A | 2000 | 2K | 20M |
| B | 1999 | 37K/172M | B | 1999 | 37K | 172M |
| B | 2000 | 80K/174M | B | 2000 | 80K | 174M |
| C | 1999 | 212K/1T | C | 1999 | 212K | 1T |
| C | 2000 | 213K/1T | C | 2000 | 213K | 1T |

separate_rows(table3, rate, into = c("cases", "pop"))

separate_rows(data, ..., sep = "[^[:alnum:]]+", convert = FALSE)

Separate each cell in a column to make several rows. Also **separate_rows_()**.

table3

| country | year | rate | country | year | rate |
|---------|------|----------|---------|------|------|
| A | 1999 | 0.7K/19M | A | 1999 | 0.7K |
| A | 2000 | 2K/20M | A | 2000 | 2K |
| B | 1999 | 37K/172M | B | 1999 | 37K |
| B | 2000 | 80K/174M | B | 2000 | 80K |
| C | 1999 | 212K/1T | C | 1999 | 212K |
| C | 2000 | 213K/1T | C | 2000 | 213K |
| | | | | | 1T |

separate_rows(table3, rate)

unite(data, col, ..., sep = "_", remove = TRUE)

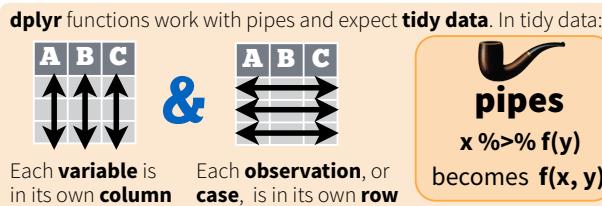
Collapse cells across several columns to make a single column.

table5

| country | century | year | country | year |
|---------|---------|------|---------|------|
| Afghan | 19 | 99 | Afghan | 1999 |
| Afghan | 20 | 0 | Afghan | 2000 |
| Brazil | 19 | 99 | Brazil | 1999 |
| Brazil | 20 | 0 | Brazil | 2000 |
| China | 19 | 99 | China | 1999 |
| China | 20 | 0 | China | 2000 |

unite(table5, century, year, col = "year", sep = "")

Data Transformation with dplyr Cheat Sheet



Summarise Cases

These apply **summary functions** to columns to create a new table. Summary functions take vectors as input and return one value (see back).



`summarise(.data, ...)`
Compute table of summaries. Also `summarise_()`.
`summarise(mtcars, avg = mean(mpg))`

`count(x, ..., wt = NULL, sort = FALSE)`
Count number of rows in each group defined by the variables in ... Also `tally()`.
`count(iris, Species)`

Variations

- `summarise_all()` - Apply funs to every column.
- `summarise_at()` - Apply funs to specific columns.
- `summarise_if()` - Apply funs to all cols of one type.

Group Cases

Use `group_by()` to create a "grouped" copy of a table. dplyr functions will manipulate each "group" separately and then combine the results.

`mtcars %>%
group_by(cyl) %>%
summarise(avg = mean(mpg))`

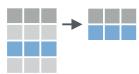
`group_by(.data, ..., add = FALSE)`
Returns copy of table grouped by ...
`g_iris <- group_by(iris, Species)`

`ungroup(x, ...)`
Returns ungrouped copy of table.
`ungroup(g_iris)`

Manipulate Cases

Extract Cases

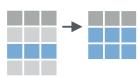
Row functions return a subset of rows as a new table. Use a variant that ends in _ for non-standard evaluation friendly code.



`filter(.data, ...)`
Extract rows that meet logical criteria. Also `filter_()`. `filter(iris, Sepal.Length > 7)`



`distinct(.data, ..., .keep_all = FALSE)`
Remove rows with duplicate values. Also `distinct_()`. `distinct(iris, Species)`



`sample_frac(tbl, size = 1, replace = FALSE, weight = NULL, .env = parent.frame())`
Randomly select fraction of rows.
`sample_frac(iris, 0.5, replace = TRUE)`



`sample_n(tbl, size, replace = FALSE, weight = NULL, .env = parent.frame())`
Randomly select size rows.
`sample_n(iris, 10, replace = TRUE)`

`slice(.data, ...)`
Select rows by position. Also `slice_()`.
`slice(iris, 10:15)`

`top_n(x, n, wt)`
Select and order top n entries (by group if grouped data). `top_n(iris, 5, Sepal.Width)`

Logical and boolean operators to use with filter()

| | | | | | |
|---|----|----------|------|---|-------|
| < | = | is.na() | %in% | | xor() |
| > | >= | !is.na() | ! | & | |

See `?base::logic` and `?Comparison` for help.

Arrange Cases



`arrange(.data, ...)`
Order rows by values of a column (low to high), use with `desc()` to order from high to low.
`arrange(mtcars, mpg)`
`arrange(mtcars, desc(mpg))`



`add_row(.data, ..., .before = NULL, .after = NULL)`
Add one or more rows to a table.
`add_row(faithful, eruptions = 1, waiting = 1)`

Manipulate Variables

Extract Variables

Column functions return a set of columns as a new table. Use a variant that ends in _ for non-standard evaluation friendly code.



`select(.data, ...)`
Extract columns by name. Also `select_if()`.
`select(iris, Sepal.Length, Species)`

Use these helpers with `select()`, e.g. `select(iris, starts_with("Sepal"))`

`contains(match)`
`ends_with(match)`
`matches(match)`

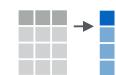
`num_range(prefix, range)` ; e.g. `mpg:cyl`
`one_of(...)` -, e.g. `-Species`
`starts_with(match)`

Make New Variables

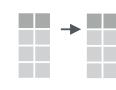
These apply **vectorized functions** to columns. Vectorized funs take vectors as input and return vectors of the same length as output (see back).



`mutate(.data, ...)`
Compute new column(s).
`mutate(mtcars, gpm = 1/mpg)`



`transmute(.data, ...)`
Compute new column(s), drop others.
`transmute(mtcars, gpm = 1/mpg)`

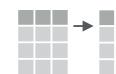


`mutate_all(.tbl, .funs, ...)`
Apply funs to every column. Use with `funs()`.
`mutate_all(faithful, funs(log(.), log2(.)))`



`mutate_at(.tbl, .cols, .funs, ...)`
Apply funs to specific columns. Use with `funs()`, `vars()` and the helper functions for `select()`.
`mutate_at(iris, vars(-Species), funs(log(.)))`

`mutate_if(.tbl, .predicate, .funs, ...)`
Apply funs to all columns of one type. Use with `funs()`.
`mutate_if(iris, is.numeric, funs(log(.)))`



`add_column(.data, ..., .before = NULL, .after = NULL)`
Add new column(s).
`add_column(mtcars, new = 1:32)`



`rename(.data, ...)`
Rename columns.
`rename(iris, Length = Sepal.Length)`

Vectorized Functions

to use with mutate()

mutate() and **transmute()** apply vectorized functions to columns to create new columns. Vectorized functions take vectors as input and return vectors of the same length as output.



Offsets

dplyr::lag() - Offset elements by 1
dplyr::lead() - Offset elements by -1

Cumulative Aggregates

dplyr::cumall() - Cumulative all()
dplyr::cumany() - Cumulative any()
cummax() - Cumulative max()
dplyr::cummean() - Cumulative mean()
cummin() - Cumulative min()
cumprod() - Cumulative prod()
cumsum() - Cumulative sum()

Rankings

dplyr::cume_dist() - Proportion of all values <= rank
dplyr::dense_rank() - rank with ties = min, no gaps
dplyr::min_rank() - rank with ties = min
dplyr::ntile() - bins into n bins
dplyr::percent_rank() - min_rank scaled to [0,1]
dplyr::row_number() - rank with ties = "first"

Math

+, -, *, /, ^, %/%, %% - arithmetic ops

log(), log2(), log10() - logs

<, <=, >, >=, !=, == - logical comparisons

Misc

dplyr::between() - x >= left & x <= right
dplyr::case_when() - multi-case if_else()
dplyr::coalesce() - first non-NA values by element across a set of vectors
dplyr::if_else() - element-wise if() + else()
dplyr::na_if() - replace specific values with NA
pmax() - element-wise max()
pmin() - element-wise min()
dplyr::recode() - Vectorized switch()
dplyr::recode_factor() - Vectorized switch() for factors

Summary Functions

to use with summarise()

summarise() applies summary functions to columns to create a new table. Summary functions take vectors as input and return single values as output.



Counts

dplyr::n() - number of values/rows
dplyr::n_distinct() - # of uniques
sum(!is.na()) - # of non-NA's

Location

mean() - mean, also mean(!is.na())
median() - median

Logicals

mean() - Proportion of TRUE's
sum() - # of TRUE's

Position/Order

dplyr::first() - first value
dplyr::last() - last value
dplyr::nth() - value in nth location of vector

Rank

quantile() - nth quantile
min() - minimum value
max() - maximum value

Spread

IQR() - Inter-Quartile Range
mad() - mean absolute deviation
sd() - standard deviation
var() - variance

Row names

Tidy data does not use rownames, which store a variable outside of the columns. To work with the rownames, first move them into a column.

| A | B | C | A | B |
|---|---|---|---|---|
| 1 | a | t | 1 | a |
| 2 | b | u | 2 | b |
| 3 | c | v | 3 | c |

rownames_to_column()

Move row names into col.
a <- rownames_to_column(irris, var = "C")

| A | B | C |
|---|---|---|
| a | t | 1 |
| b | u | 2 |
| c | v | 3 |

column_to_rownames()

Move col in row names.
column_to_rownames(a, var = "C")

Also has_rownames(), remove_rownames()

Combine Tables

Combine Variables

| x | y |
|---|---|
| A | B |
| a | t |
| b | u |
| c | v |
| 1 | 3 |

| x | y |
|---|---|
| A | B |
| a | t |
| b | u |
| c | w |
| 1 | 3 |

Use bind_cols() to paste tables beside each other as they are.

| A | B | C | A | B | D |
|---|---|---|---|---|---|
| a | t | 1 | a | t | 3 |
| b | u | 2 | b | u | 2 |
| c | v | 3 | d | w | 1 |

bind_cols(...)

Returns tables placed side by side as a single table.
BE SURE THAT ROWS ALIGN.

Use a "Mutating Join" to join one table to columns from another, matching values with the rows that they correspond to. Each join retains a different combination of values from the tables.

| A | B | C | D |
|---|---|---|----|
| a | t | 1 | 3 |
| b | u | 2 | 2 |
| c | v | 3 | NA |

left_join(x, y, by = NULL, copy = FALSE, suffix=c("x","y"),...)
Join matching values from y to x.

| A | B | C | D |
|---|---|----|---|
| a | t | 1 | 3 |
| b | u | 2 | 2 |
| d | w | NA | 1 |

right_join(x, y, by = NULL, copy = FALSE, suffix=c("x","y"),...)
Join matching values from x to y.

| A | B | C | D |
|---|---|---|---|
| a | t | 1 | 3 |
| b | u | 2 | 2 |

inner_join(x, y, by = NULL, copy = FALSE, suffix=c("x","y"),...)
Join data. Retain only rows with matches.

| A | B | C | D |
|---|---|----|----|
| a | t | 1 | 3 |
| b | u | 2 | 2 |
| c | v | 3 | NA |
| d | w | NA | 1 |

full_join(x, y, by = NULL, copy = FALSE, suffix=c("x","y"),...)
Join data. Retain all values, all rows.

| A | B | C | B | y | D |
|---|---|---|----|----|---|
| a | t | 1 | t | 3 | |
| b | u | 2 | u | 2 | |
| c | v | 3 | NA | NA | |

Use by = c("col1", "col2") to specify the column(s) to match on.

left_join(x, y, by = "A")

| A.x | B.x | C | A.y | B.y | D |
|-----|-----|---|-----|-----|---|
| a | t | 1 | t | 3 | |
| b | u | 2 | u | 2 | |
| c | v | 3 | NA | NA | |

Use a named vector, by = c("col1" = "col2"), to match on columns with different names in each data set.

left_join(x, y, by = c("C" = "D"))

| A1 | B1 | C | A2 | B2 |
|----|----|---|----|----|
| a | t | 1 | d | w |
| b | u | 2 | b | u |
| c | v | 3 | a | t |

Use suffix to specify suffix to give to duplicate column names.
left_join(x, y, by = c("C" = "D"), suffix = c("1", "2"))

Combine Cases

| A | B | C |
|---|---|---|
| a | t | 1 |
| b | u | 2 |

| A | B | C |
|---|---|---|
| c | v | 3 |
| d | w | 4 |

Use bind_rows() to paste tables below each other as they are.

| DF | A | B | C |
|----|---|---|---|
| x | a | t | 1 |
| x | b | u | 2 |
| x | c | v | 3 |
| z | c | v | 3 |
| z | d | w | 4 |

bind_rows(..., .id = NULL)

Returns tables one on top of the other as a single table. Set .id to a column name to add a column of the original table names (as pictured)

| A | B | C |
|---|---|---|
| a | t | 1 |
| b | u | 2 |

intersect(x, y, ...)

Rows that appear in both x and z.

| A | B | C |
|---|---|---|
| a | t | 1 |
| b | u | 2 |

setdiff(x, y, ...)

Rows that appear in x but not z.

| A | B | C |
|---|---|---|
| a | t | 1 |
| b | u | 2 |

union(x, y, ...)

Rows that appear in x or z. (Duplicates removed). union_all() retains duplicates.

| x | y |
|---|---|
| A | B |
| a | t |
| b | u |

setequal(x, y)

Use setequal() to test whether two data sets contain the exact same rows (in any order).

| A | B | C |
|---|---|---|
| a | t | 1 |
| b | u | 2 |

semi_join(x, y, by = NULL, ...)

Return rows of x that have a match in y. USEFUL TO SEE WHAT WILL BE JOINED.

| A | B | C |
|---|---|---|
| c | v | 3 |
| d | w | 4 |

anti_join(x, y, by = NULL, ...)

Return rows of x that do not have a match in y. USEFUL TO SEE WHAT WILL NOT BE JOINED.

Data Science in Spark

with sparklyr
Cheat Sheet



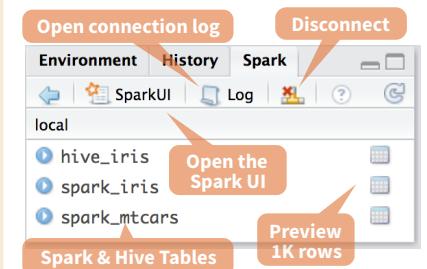
Intro

sparklyr is an R interface for Apache Spark™, it provides a complete **dplyr** backend and the option to query directly using **Spark SQL** statement. With sparklyr, you can orchestrate distributed machine learning using either **Spark's MLLib** or **H2O** Sparkling Water.



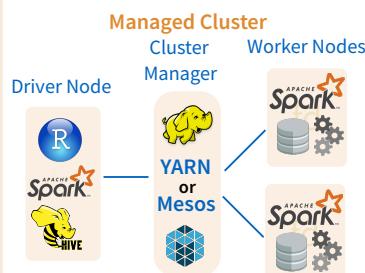
Starting with **version 1.044**, **RStudio Desktop, Server and Pro include integrated support for the sparklyr package**. You can create and manage connections to Spark clusters and local Spark instances from inside the IDE.

RStudio Integrates with sparklyr

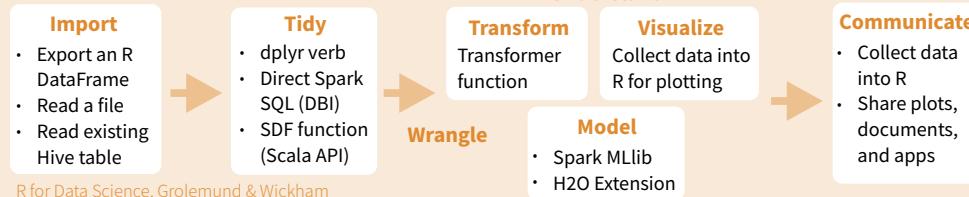


Cluster Deployment

Cluster Deployment Options



Data Science Toolchain with Spark + sparklyr



Getting started

Local Mode

Easy setup; no cluster required

1. Install a local version of Spark:
`spark_install ("2.0.1")`
2. Open a connection
`sc <- spark_connect (master = "local")`

On a Mesos Managed Cluster

1. Install RStudio Server or Pro on one of the existing nodes
2. Locate path to the cluster's Spark directory
3. Open a connection
`spark_connect(master = "[mesos URL]", version = "1.6.2", spark_home = [Cluster's Spark path])`

Using Livy (Experimental)

1. The Livy REST application should be running on the cluster
2. Connect to the cluster
`sc <- spark_connect(master = "http://host:port", method = "livy")`

On a YARN Managed Cluster

1. Install RStudio Server or RStudio Pro on one of the existing nodes, preferably an edge node
2. Locate path to the cluster's Spark Home Directory, it normally is "/usr/lib/spark"
3. Open a connection
`spark_connect(master = "yarn-client", version = "1.6.2", spark_home = [Cluster's Spark path])`

On a Spark Standalone Cluster

1. Install RStudio Server or RStudio Pro on one of the existing nodes or a server in the same LAN
2. Install a local version of Spark:
`spark_install (version = "2.0.1")`
3. Open a connection
`spark_connect(master = "spark:// host:port", version = "2.0.1", spark_home = spark_home_dir())`

Using sparklyr

A brief example of a data analysis using Apache Spark, R and sparklyr in local mode

```
library(sparklyr); library(dplyr); library(ggplot2);
library(tidyr); set.seed(100)
```

Install Spark locally

```
spark_install("2.0.1")
```

Connect to local version

```
sc <- spark_connect(master = "local")
```

Copy data to Spark memory

```
partition_iris <- sdf_partition(
  import_iris, training=0.5, testing=0.5)
```

Partition data

```
sdf_register(partition_iris,
c("spark_iris_training","spark_iris_test"))
```

Create a hive metadata for each partition

```
tidy_iris <- tbl(sc,"spark_iris_training") %>%
  select(Species, Petal_Length, Petal_Width)
```

Spark ML Decision Tree Model

```
model_iris <- tidy_iris %>%
  ml_decision_tree(response="Species",
  features=c("Petal_Length","Petal_Width"))
```

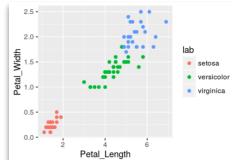
Create reference to Spark table

```
test_iris <- tbl(sc,"spark_iris_test")
```

```
pred_iris <- sdf_predict(
  model_iris, test_iris) %>%
  collect
```

Bring data back into R memory for plotting

```
pred_iris %>%
  inner_join(data.frame(prediction=0:2,
  lab=model_iris$model.parameters$labels)) %>%
  ggplot(aes(Petal_Length, Petal_Width, col=lab)) +
  geom_point()
```



```
spark_disconnect(sc)
```

Disconnect

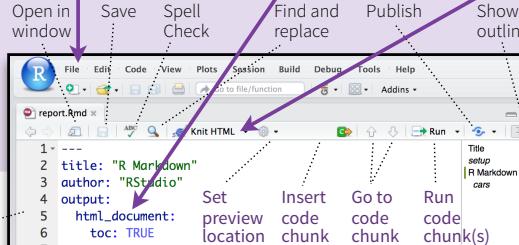
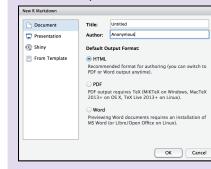
| Import | Visualize & Communicate | Model (MLlib) |
|--|--|---|
| <p>Copy a DataFrame into Spark <code>sdf_copy_to(sc, iris, "spark_iris")</code></p> <p><code>sdf_copy_to(sc, x, name, memory, repartition, overwrite)</code></p> <p>Import into Spark from a File Arguments that apply to all functions: <code>sc, name, path, options = list(), repartition = 0, memory = TRUE, overwrite = TRUE</code></p> <p>CSV <code>spark_read_csv(header = TRUE, columns = NULL, infer_schema = TRUE, delimiter = "", quote = "", escape = "\\", charset = "UTF-8", null_value = NULL)</code></p> <p>JSON <code>spark_read_json()</code></p> <p>PARQUET <code>spark_read_parquet()</code></p> | <p>Spark SQL commands <code>DBI::dbWriteTable(sc, "spark_iris", iris)</code></p> <p><code>DBI::dbWriteTable(conn, name, value)</code></p> <p>From a table in Hive <code>my_var <- tbl_cache(sc, name = "hive_iris")</code></p> <p><code>tbl_cache(sc, name, force = TRUE)</code> Loads the table into memory</p> <p><code>my_var <- dplyr::tbl(sc, name = "hive_iris")</code></p> <p><code>dplyr::tbl(sc, ...)</code> Creates a reference to the table without loading it into memory</p> | <p>Download data to R memory <code>r_table <- collect(my_table)</code> <code>plot(Petal_Width ~ Petal_Length, data = r_table)</code></p> <p><code>dplyr::collect(x)</code> Download a Spark DataFrame to an R DataFrame</p> <p><code>sdf_read_column(x, column)</code> Returns contents of a single column to R</p> |
| Wrangle | | |
| <p>Spark SQL via dplyr verbs Translates into Spark SQL statements</p> <p><code>my_table <- my_var %>% filter(Species == "setosa") %>% sample_n(10)</code></p> <p>Direct Spark SQL commands <code>my_table <- DBI::dbGetQuery(sc, "SELECT * FROM iris LIMIT 10")</code></p> <p><code>DBI::dbGetQuery(conn, statement)</code></p> <p>Scala API via SDF functions</p> <p><code>sdf_mutate(.data)</code> Works like dplyr mutate function</p> <p><code>sdf_partition(x, ..., weights = NULL, seed = sample (.Machine\$integer.max, 1))</code> <code>sdf_partition(x, training = 0.5, test = 0.5)</code></p> <p><code>sdf_register(x, name = NULL)</code> Gives a Spark DataFrame a table name</p> <p><code>sdf_sample(x, fraction = 1, replacement = TRUE, seed = NULL)</code></p> <p><code>sdf_sort(x, columns)</code> Sorts by >=1 columns in ascending order</p> <p><code>sdf_with_unique_id(x, id = "id")</code> Add unique ID column</p> <p><code>sdf_predict(object, newdata)</code> Spark DataFrame with predicted values</p> | <p>ML Transformers</p> <p><code>ft_binarizer(my_table, input.col = "Petal_Length", output.col = "petal_large", threshold = 1.2)</code></p> <p>Arguments that apply to all functions: <code>x, input.col = NULL, output.col = NULL</code></p> <p><code>ft_binarizer(threshold = 0.5)</code> Assigned values based on threshold</p> <p><code>ft_bucketizer(splits)</code> Numeric column to discretized column</p> <p><code>ft_discrete_cosine_transform(invers e = FALSE)</code> Time domain to frequency domain</p> <p><code>ft_elementwise_product(scaling.col)</code> Element-wise product between 2 columns</p> <p><code>ft_index_to_string()</code> Index labels back to label as strings</p> <p><code>ft_one_hot_encoder()</code> Continuous to binary vectors</p> <p><code>ft_quantile_discretizer(n.buckets = 5L)</code> Continuous to binned categorical values</p> <p><code>ft_sql_transformer(sql)</code></p> <p><code>ft_string_indexer(params = NULL)</code> Column of labels into a column of label indices.</p> <p><code>ft_vectorAssembler()</code> Combine vectors into a single row-vector</p> | <p>Reading & Writing from Apache Spark</p> <p><code>sdf_copy_to</code> <code>dplyr::copy_to</code> <code>DBI::dbWriteTable</code></p> <p><code>tbl_cache</code> <code>dplyr::tbl</code></p> <p><code>sdf_collect</code> <code>dplyr::collect</code></p> <p><code>sdf_read_column</code></p> <p><code>File</code> <code>System</code></p> <p><code>spark_read_<fmt></code> <code>spark_write_<fmt></code></p> <p>Extensions Create an R package that calls the full Spark API & provide interfaces to Spark packages.</p> <p>Core Types</p> <p><code>spark_connection()</code> Connection between R and the Spark shell process</p> <p><code>spark_job()</code> Instance of a remote Spark object</p> <p><code>spark_dataframe()</code> Instance of a remote Spark DataFrame object</p> <p>Call Spark from R</p> <p><code>invoke()</code> Call a method on a Java object</p> <p><code>invoke_new()</code> Create a new object by invoking a constructor</p> <p><code>invoke_static()</code> Call a static method on an object</p> <p>Machine Learning Extensions</p> <p><code>ml_create_dummy_variables()</code> <code>ml_options()</code></p> <p><code>ml_prepare_dataframe()</code> <code>ml_model()</code></p> <p><code>ml_prepare_response_features_intercept()</code></p> |
| sparklyr is an R interface for Apache Spark dplyr ML Extensions | | |

R Markdown Cheat Sheet

learn more at rmarkdown.rstudio.com



- 1 Open a new .Rmd file at File ▶ New File ▶ R Markdown. Use the wizard that opens to pre-populate the file with a template.



.Rmd structure

YAML Header

Optional section of render (e.g. pandoc) options written as keyvalue pairs (YAML).

- At start of file
- Between lines of ---

Text

Narration formatted with markdown, mixed with:

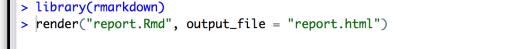
Code chunks

Chunks of embedded code. Each chunk:

- Begins with `{{`{r}
- ends with `}`}

R Markdown will run the code and append the results to the doc.

It will use the location of the .Rmd file as the working directory



Inline code

Insert with `r <code>`. Results appear as text without code.

Built with
`r getRVersion()`

→ Built with 3.2.3

Important chunk options

cache - cache results for future knits (default = FALSE)

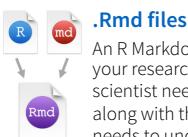
cache.path - directory to save cached results in (default = "cache")

child - file(s) to knit and then include (default = NULL)

collapse - collapse all output into single block (default = FALSE)

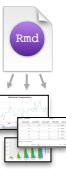
comment - prefix for each line of results (default = "#")

Options not listed above: R.options, aniopts, autoplot, background, cache.comments, cache.lazy, cache.rebuild, cache.vars, dev, dev.args, dpi, engine.opts, engine.path, fig.asp, fig.env, fig.ext, fig.keep, fig.lp, fig.path, fig.pos, fig.process, fig.retina, fig.scap, fig.show, fig.showtext, fig.subcap, interval, out.extra, out.height, out.width, prompt, purl, ref.label, render, size, split, tidy.opts



.Rmd files

An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.



Reproducible Research

At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.



Dynamic Documents

You can choose to export the finished report as a html, pdf, MS Word, ODT, RTF, or markdown document; or as a html or pdf based slide show.

Workflow

- 2 Write document by editing template



- 3 Knit document to create report Use knit button or render() to knit



- 4 Preview Output in IDE window



- 5 Publish (optional) to web or server

Synch publish button to accounts at

- rpubs.com
- shinyapps.io
- RStudio Connect

Reload document

Find in document

File path to output document

- 6 Examine build log in R Markdown console

- 7 Use output file that is saved alongside .Rmd

Interactive Documents

Turn your report into an interactive Shiny document in 4 steps



- 1 Add runtime: shiny to the YAML header.

- 2 Call Shiny input functions to embed input objects.

- 3 Call Shiny render functions to embed reactive output.

- 4 Render with rmarkdown::run or click Run Document in RStudio IDE

```
---  
output: html_document  
runtime: shiny  
---  
  
```{r, echo = FALSE}  
numericInput("n",
 "How many cars?", 5)

renderTable({
 head(cars, input$n)
})
```
```

| How many cars? | |
|----------------|-------|
| speed | dist |
| 1 | 2.00 |
| 2 | 10.00 |
| 3 | 4.00 |
| 4 | 22.00 |
| 5 | 16.00 |

Embed a complete app into your document with shiny::shinyAppDir()

* Your report will be rendered as a Shiny app, which means you must choose an html output format, like **html_document**, and serve it with an active R Session.

Embed code with knitr syntax

Code chunks

```
```{r echo=TRUE}  
getRVersion()
```
```



```
getRVersion()  
## [1] '3.2.3'
```

Global options

Set with knitr::opts_chunk\$set(), e.g.

```
```{r include=FALSE}  
knitr::opts_chunk$set(echo = TRUE)
```
```

message - display code messages in document (default = TRUE)

results (default = 'markup')

'asis' - passthrough results

'hide' - do not display results

'hold' - put all results below all code

tidy - tidy code for display (default = FALSE)

warning - display code warnings in document (default = TRUE)

Parameters

Parameterize your documents to reuse with different inputs (e.g., data sets, values, etc.)

- 1 Add parameters

Create and set parameters in the header as sub-values of **params**

```
---  
params:  
  n: 100  
  d: !r Sys.Date()  
---
```

- 2 Call parameters

Call parameter values in code as **params\$name**

```
Today's date  
is `r params$d`
```

- 3 Set parameters

Set values with **Knit with parameters** or the **params** argument of **render()**:

```
render("doc.Rmd",  
  params = list(n = 1, d = as.Date("2015-01-01")))
```

Pandoc's Markdown

Write with syntax on the left to create effect on right (after render)

Plain text
 End a line with two spaces to start a new paragraph.
`*italics*` and `**bold**`
`'verbatim code'`
`sub/superscript^2~2~`
`--strikethrough--`
`escaped: *___*`
`endash: --, emdash: ---`
`equation: $A = \pi * r^2$`
`equation block:`

$$\text{---}$$

$$E = mc^2$$

block quote

Header1
Header 2
Header 3
Header 4
Header 5
Header 6
<!--Text comment-->
`\textbf{Text ignored in HTML}`
`HTML ignored in pdfs`
`<http://www.rstudio.com>`
`[link](www.rstudio.com)`
`Jump to [Header 1] (#anchor)`
`image:`
`! [Caption] (smallorb.png)`
`* unordered list`
`+ sub-item 1`
`+ sub-item 2`
`- sub-sub-item 1`
`* item 2`
`Continued (indent 4 spaces)`
`1. ordered list`
`2. item 2`
`1) sub-item 1`
`A. sub-sub-item 1`
`1. A list whose numbering continues after`
`2. an interruption`
`Term 1`
`Definition 1`

| Right | Left | Default | Center |
|-------|------|---------|--------|
| 12 | 12 | 12 | 12 |
| 123 | 123 | 123 | 123 |
| 1 | 1 | 1 | 1 |

`- slide bullet 1`
`- slide bullet 2`
`(-> to have bullets appear on click)`
`horizontal rule/slide break:`
`***`
`A footnote [^1]`
`[^1]: Here is the footnote.`
`1. Here is the footnote. ↵`

When you render, R Markdown
 1. runs the R code, embeds results and text into .md file with knitr
 2. then converts the .md file into the finished format with pandoc



Set a document's default output format in the YAML header:

```
---
```

```
output: html_document
```

```
---
```

```
# Body
```

output value creates

| | |
|------------------------------------|----------------------------------|
| <code>html_document</code> | <code>html</code> |
| <code>pdf_document</code> | <code>pdf</code> (requires Tex) |
| <code>word_document</code> | Microsoft Word (.docx) |
| <code>odt_document</code> | OpenDocument Text |
| <code>rtf_document</code> | Rich Text Format |
| <code>md_document</code> | Markdown |
| <code>github_document</code> | Github compatible markdown |
| <code>ioslides_presentation</code> | ioslides HTML slides |
| <code>slidy_presentation</code> | slidy HTML slides |
| <code>beamer_presentation</code> | Beamer pdf slides (requires Tex) |

Customize output with sub-options (listed at right):

```
---
```

```
output: Indent 2 spaces
```

```
          Indent 4 spaces
```

```
html_document:
```

```
  code_folding: hide
```

```
  toc_float: TRUE
```

```
---
```

```
# Body
```

html tabs

Use .tabset.css class to place sub-headers into tabs

```
# Tabset {.tabset .tabset-fade .tabset-pills}
```

Tabset

Tab 1 Tab 2

text 1

End tabset

Create a Reusable template

1

Create a new package with a `inst/rmarkdown/templates` directory

2

In the directory, Place a folder that contains:

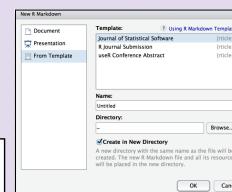
- `template.yaml` (see below)
- `skeleton.Rmd` (contents of the template)
- any supporting files

3

Install the package

4

Access template in wizard at File ► New File ► R Markdown



`template.yaml`

```
---
```

```
name: My Template
```

```
---
```

Set render options with YAML

| sub-option | description | html | pdf | word | odt | rtf | md | gitbook | ioslides | slidy | beamer |
|---|--|------|-----|------|-----|-----|----|---------|----------|-------|--------|
| <code>citation_package</code> | The LaTeX package to process citations, natbib, biblatex or none | X | | X | | | | | | | X |
| <code>code_folding</code> | Let readers to toggle the display of R code, "none", "hide", or "show" | X | | | | | | | | | X |
| <code>colortheme</code> | Beamer color theme to use | | | | | | | | | | X |
| <code>css</code> | CSS file to use to style document | X | | | X | X | | X | X | X | X |
| <code>dev</code> | Graphics device to use for figure output (e.g. "png") | X | X | | X | X | X | X | X | X | X |
| <code>duration</code> | Add a countdown timer (in minutes) to footer of slides | | | | | | | | | | X |
| <code>fig_caption</code> | Should figures be rendered with captions? | X | X | X | | | X | X | X | X | X |
| <code>fig_height, fig_width</code> | Default figure height and width (in inches) for document | X | X | X | X | X | X | X | X | X | X |
| <code>highlight</code> | Syntax highlighting: "tango", "pygments", "kate", "zenburn", "textmate" | X | X | X | | | | | | | X |
| <code>includes</code> | File of content to place in document (<code>in_header</code> , <code>before_body</code> , <code>after_body</code>) | X | X | X | X | X | X | X | X | X | X |
| <code>incremental</code> | Should bullets appear one at a time (on presenter mouse clicks)? | | | | | | | | | | X |
| <code>keep_md</code> | Save a copy of .md file that contains knitr output | X | X | X | X | | | | | | X |
| <code>keep_tex</code> | Save a copy of .tex file that contains knitr output | | | | | X | | | | | X |
| <code>latex_engine</code> | Engine to render latex, "pdflatex", "xelatex", or "lualatex" | | | | | X | | | | | X |
| <code>lib_dir</code> | Directory of dependency files to use (Bootstrap, MathJax, etc.) | X | | | | | | | | | X |
| <code>mathjax</code> | Set to <code>local</code> or a URL to use a local/URL version of MathJax to render | X | | | | | | | | | X |
| <code>md_extensions</code> | Markdown extensions to add to default definition or R Markdown | X | X | X | X | X | X | X | X | X | X |
| <code>number_sections</code> | Add section numbering to headers | X | X | | | | | | | | X |
| <code>pandoc_args</code> | Additional arguments to pass to Pandoc | X | X | X | X | X | X | X | X | X | X |
| <code>preserve_yaml</code> | Preserve YAML front matter in final document? | | | | | | | | | | X |
| <code>reference_docx</code> | docx file whose styles should be copied when producing docx output | | | | | | | | | | X |
| <code>self_contained</code> | Embed dependencies into the doc | | | | | X | | | | | X |
| <code>slide_level</code> | The lowest heading level that defines individual slides | | | | | | | | | | X |
| <code>smaller</code> | Use the smaller font size in the presentation? | | | | | | | | | | X |
| <code>smart</code> | Convert straight quotes to curly, dashes to em-dashes, ... to ellipses, etc. | X | | | | | | | | | X |
| <code>template</code> | Pandoc template to use when rendering file | X | X | X | | | | | | | X |
| <code>theme</code> | Bootswatch or Beamer theme to use for page | X | | | | | | | | | X |
| <code>toc</code> | Add a table of contents at start of document | X | X | X | | | | | | | X |
| <code>toc_depth</code> | The lowest level of headings to add to table of contents | X | X | X | | | | | | | X |
| <code>toc_float</code> | Float the table of contents to the left of the main content | | | | | | | | | | X |
| Options not listed: extra_dependencies, fig_crop, fig_retina, font_adjustment, font_theme, footer, logo, html_preview, reference_odt, transition, variant, widescreen | | | | | | | | | | | |

Table suggestions

Several functions format R data into tables

| Table with kable | eruptions.waiting | Table with stargazer |
|------------------|-------------------|----------------------|
| eruptions | eruptions.waiting | eruptions.waiting |
| waiting | 1 3.60 79.00 | 1 3.600 79 |
| | 2 1.80 54.00 | 2 1.800 54 |
| | 3 3.33 74.00 | 3 3.333 74 |
| | 4 2.28 62.00 | 4 2.283 62 |
| | | |
| | | Table with xtable |
| | | eruptions.waiting |
| | | 1 3.600 79 |
| | | 2 1.800 54 |
| | | 3 3.333 74 |
| | | 4 2.283 62 |

data <faithful[1:4,]

```
```{r results = 'asis'}
```

```
knitr::kable(data, caption = "Table with kable")
```

```
```{r results = "asis"}
```

```
print(xtable::xtable(data, caption = "Table with xtable"),
      type = "html", html.table.attributes = "border=0"))
```

```
```{r results = "asis"}
```

```
stargazer::stargazer(data, type = "html",
 title = "Table with stargazer")
```

Learn more in the `stargazer`, `xtable`, and `knitr` packages.

## Citations and Bibliographies

Create citations with .bib, .bibtex, .copac, .enl, .json, .medline, .mods, .ris, .wos, and .xml files

**1** Set bibliography file and CSL 1.0 Style file (optional) in the YAML header

```

```

```
bibliography: refs.bib
```

```
csl: style.csl
```

```

```

**2** Use citation keys in text

Smith cited [@smith04].  
 Smith cited without author [-@smith04].  
 @smith04 cited in line.

**3** Render. Bibliography will be added to end of document

Smith cited (Joe Smith 2004).  
 Smith cited without author (2004).  
 Joe Smith (2004) cited in line.

# RStudio IDE Cheat Sheet

learn more at [www.rstudio.com](http://www.rstudio.com)



The RStudio IDE is an Integrated Development Environment in R that comes in three versions



## Desktop IDE

A local version of the IDE  
for your desktop



**Open Source Server**  
for larger compute resources  
and remote access



## Professional Server

for teams that share large compute  
resources, large data, and uniform  
environments for collaboration

Download all at [www.rstudio.com](http://www.rstudio.com). Each provides the same useful interface:

### Documents and Apps

Open Shiny, R Markdown, knitr, Sweave, LaTeX, .Rd files and more in Source Pane

Check spelling, Render output, Choose output format, Choose output location, Insert code chunk

Jump to previous chunk, Jump to next chunk, Run selected lines, Publish to server, Show file outline

Access markdown guide at [Help > Markdown Quick Reference](#)

Jump to chunk, Set knitr chunk options, Run this and all previous code chunks, Run this code chunk

...{r pressure, echo=FALSE}  
plot(pressure)  
...  
(Top Level) R Markdown

RStudio recognizes that files named **app.R**, **server.R**, **ui.R**, and **global.R** belong to a shiny app

Run app, Choose location to shinyapps.io or server, Publish to shinyapps.io or server, Manage publish accounts

### Write Code

Navigate tabs, Open in new window, Save, Find and replace, Compile as notebook, Run selected code

# Good start...  
# Cursors of shared users  
# Re-run previous code  
# Source with or without Echo  
# Show file outline

Multiple cursors/column selection with **Alt + mouse drag**.

Code diagnostics that appear in the margin. Hover over diagnostic symbols for details.

get\_digit <-function() {  
 ("num" %% (10 ^ n))  
 %% (10 ^ (n - 1))  
}  
fo  
for (n in 1:10) {  
 digits[n] <- get\_digit(n)  
}  
force  
Multi-language code snippets to quickly use common blocks of code.

Jump to function in file

Syntax highlighting based on your file's extension

Tab completion to finish function names, file paths, arguments, and more.

Change file type

Working Directory

Maximize, minimize panes

Press ↑ to see command history

Drag pane boundaries

Path to displayed directory

A File browser keyed to your working directory. Click on file or directory name to open.

### R Support

Import data file with wizard

History of past commands to run/add to source

Display .RPres slideshows

**File > New File > R Presentation**

Load workspace, Save workspace, Delete all saved objects, Search inside environment

Choose environment to display from list of parent environments

Data

Iris  
Values  
a  
Functions  
foo

150 obs. of 5 variables  
1  
function (x)

View in data viewer, View function source code

Files Plots Packages Help Viewer

New Folder Upload Delete Rename More  
Create folder Upload file Delete file Rename file Go To Working Directory Change directory

Set as Working Directory

Path to displayed directory

Export plot, Delete plot, Delete all plots

### Project System

File > New Project

RStudio saves the call history, workspace, and working directory associated with a project. It reloads each when you re-open a project.

Name of current project

Share Project Active shared collaborator, Start new R Session in current project, Close R Session in project, Select R Version

R New Project... Open Project... Close Share Project... IDEcheatsheet RStudio-Essentials Essential shiny-examples Clear Project List Project Options...

R version 3.2.2 R version 3.1.3 R version 3.0.3 R version 2.15.3

GUI Package manager lists every installed package

Install Update Packages Create reproducible package library for your project

scales shiny shinydashboard

Click to load package with **library()**. Click to detach package with **detach()**

Package version installed Delete from library

### Debug Mode

Open with **debug()**, **browse()**, or a breakpoint. RStudio will open the debugger mode when it encounters a breakpoint while executing code.

Click next to line number to add/remove a breakpoint.

Highlighted line shows where execution has paused

Run commands in environment where execution has paused

Examine variables in executing environment

Select function in traceback to debug

Step through code one line at a time

Step into and out of functions to run

Resume execution

Quit debug mode

Environment History Build Git

Console ~ /IDEcheatsheet /> foo() Error in get\_digit(num, x) : Show Traceback Error!

Console ~ /IDEcheatsheet /> Next Continue Stop

### Version Control with Git or SVN

Turn on at **Tools > Project Options > Git/SVN**

Stage files: Added, Deleted, Modified, Renamed, Untracked

Show file diff, Commit, Push/Pull staged files to remote, View History

Environment History Git

Diff Commit Seged Status Path file-with-changes.R Revert... Ignore... Open shell to type commands current branch

File > New Project > New Directory > R Package

Turn project into package, Enable roxygen documentation with **Tools > Project Options > Build Tools**

Roxygen guide at [Help > Roxygen Quick Reference](#)

Viewer Pane displays HTML content, such as Shiny apps, RMarkdown reports, and interactive visualizations

Stop Shiny app, Publish to shinyapps.io, rpubs, RSConnect, ...

View(<data>) opens spreadsheet like view of data set

Environment History Build Git

Load All Clean and Rebuild Test Package CMF7 Check Package OEM Build Source Package Build Binary Package Document OMD Configure Build Tools...

Filter Sepal.Length Sepal.Width Petal.Length Petal.Width Species

All All All All All

1 5.1 3.5 1.4 0.2 setosa

2

3 Filter rows by value or value range Sort by values Search for value

|                                   |                            |                      |                           |                          |                         |                                                                                                      |                      |                     |
|-----------------------------------|----------------------------|----------------------|---------------------------|--------------------------|-------------------------|------------------------------------------------------------------------------------------------------|----------------------|---------------------|
| <b>1 LAYOUT</b>                   | <b>Windows/Linux</b>       | <b>Mac</b>           | <b>4 WRITE CODE</b>       | <b>Windows /Linux</b>    | <b>Mac</b>              | <b>5 DEBUG CODE</b>                                                                                  | <b>Windows/Linux</b> | <b>Mac</b>          |
| Move focus to Source Editor       | Ctrl+1                     | Ctrl+1               | <b>Attempt completion</b> | <b>Tab or Ctrl+Space</b> | <b>Tab or Cmd+Space</b> | Toggle Breakpoint                                                                                    | Shift+F9             | Shift+F9            |
| Move focus to Console             | Ctrl+2                     | Ctrl+2               | Navigate candidates       | ↑/↓                      | ↑/↓                     | Execute Next Line                                                                                    | F10                  | F10                 |
| Move focus to Help                | Ctrl+3                     | Ctrl+3               | Accept candidate          | Enter, Tab, or →         | Enter, Tab, or →        | Step Into Function                                                                                   | Shift+F4             | Shift+F4            |
| Show History                      | Ctrl+4                     | Ctrl+4               | Dismiss candidates        | Esc                      | Esc                     | Finish Function/Loop                                                                                 | Shift+F6             | Shift+F6            |
| Show Files                        | Ctrl+5                     | Ctrl+5               | Undo                      | Ctrl+Z                   | Cmd+Z                   | Continue                                                                                             | Shift+F5             | Shift+F5            |
| Show Plots                        | Ctrl+6                     | Ctrl+6               | Redo                      | Ctrl+Shift+Z             | Cmd+Shift+Z             | Stop Debugging                                                                                       | Shift+F8             | Shift+F8            |
| Show Packages                     | Ctrl+7                     | Ctrl+7               | Cut                       | Ctrl+X                   | Cmd+X                   |                                                                                                      |                      |                     |
| Show Environment                  | Ctrl+8                     | Ctrl+8               | Copy                      | Ctrl+C                   | Cmd+C                   |                                                                                                      |                      |                     |
| Show Git/SVN                      | Ctrl+9                     | Ctrl+9               | Paste                     | Ctrl+V                   | Cmd+V                   |                                                                                                      |                      |                     |
| Show Build                        | Ctrl+0                     | Ctrl+0               | Select All                | Ctrl+A                   | Cmd+A                   |                                                                                                      |                      |                     |
| <b>2 RUN CODE</b>                 | <b>Windows/Linux</b>       | <b>Mac</b>           | Delete Line               | Ctrl+D                   | Cmd+D                   |                                                                                                      |                      |                     |
| <b>Search command history</b>     | <b>Ctrl+↑</b>              | <b>Cmd+↑</b>         | Select                    | Shift+[Arrow]            | Shift+[Arrow]           |                                                                                                      |                      |                     |
| Navigate command history          | ↑/↓                        | ↑/↓                  | Select Word               | Ctrl+Shift+←/→           | Option+Shift+←/→        | <b>6 VERSION CONTROL</b>                                                                             | <b>Windows/Linux</b> | <b>Mac</b>          |
| Move cursor to start of line      | Home                       | Cmd+←                | Select to Line Start      | Alt+Shift+←              | Cmd+Shift+←             | Show diff                                                                                            | Ctrl+Alt+D           | Ctrl+Option+D       |
| Move cursor to end of line        | End                        | Cmd+→                | Select to Line End        | Alt+Shift+→              | Cmd+Shift+→             | Commit changes                                                                                       | Ctrl+Alt+M           | Ctrl+Option+M       |
| Change working directory          | Ctrl+Shift+H               | Ctrl+Shift+H         | Select Page Up/Down       | Shift+PageUp/Down        | Shift+PageUp/Down       | Scroll diff view                                                                                     | Ctrl+↑/↓             | Ctrl+↑/↓            |
| Interrupt current command         | Esc                        | Esc                  | Select to Start/End       | Shift+Alt+↑/↓            | Cmd+Shift+↑/↓           | Stage/Unstage (Git)                                                                                  | Spacebar             | Spacebar            |
| <b>Clear console</b>              | <b>Ctrl+L</b>              | <b>Ctrl+L</b>        | Delete Word Left          | Ctrl+Backspace           | Ctrl+Opt+Backspace      | Stage/Unstage and move to next                                                                       | Enter                | Enter               |
| Quit Session (desktop only)       | Ctrl+Q                     | Cmd+Q                | Delete Word Right         |                          | Option+Delete           |                                                                                                      |                      |                     |
| <b>Restart R Session</b>          | <b>Ctrl+Shift+F10</b>      | <b>Cmd+Shift+F10</b> | Delete to Line End        |                          | Ctrl+K                  |                                                                                                      |                      |                     |
| <b>Run current line/selection</b> | <b>Ctrl+Enter</b>          | <b>Cmd+Enter</b>     | Delete to Line Start      |                          | Option+Backspace        |                                                                                                      |                      |                     |
| Run current (retain cursor)       | Alt+Enter                  | Option+Enter         | Indent                    | Tab (at start of line)   | Tab (at start of line)  | <b>7 MAKE PACKAGES</b>                                                                               | <b>Windows/Linux</b> | <b>Mac</b>          |
| Run from current to end           | Ctrl+Alt+E                 | Cmd+Option+E         | Outdent                   | Shift+Tab                | Shift+Tab               | Build and Reload                                                                                     | Ctrl+Shift+B         | Cmd+Shift+B         |
| Run the current function          | Ctrl+Alt+F                 | Cmd+Option+F         | Yank line up to cursor    | Ctrl+U                   | Ctrl+U                  | <b>Load All (devtools)</b>                                                                           | <b>Ctrl+Shift+L</b>  | <b>Cmd+Shift+L</b>  |
| Source a file                     | Ctrl+Shift+O               | Cmd+Shift+O          | Yank line after cursor    | Ctrl+K                   | Ctrl+K                  | <b>Test Package (Desktop)</b>                                                                        | <b>Ctrl+Shift+T</b>  | <b>Cmd+Shift+T</b>  |
| <b>Source the current file</b>    | <b>Ctrl+Shift+S</b>        | <b>Cmd+Shift+S</b>   | Insert yanked text        | Ctrl+Y                   | Ctrl+Y                  | Test Package (Web)                                                                                   | Ctrl+Alt+F7          | Cmd+Alt+F7          |
| Source with echo                  | Ctrl+Shift+Enter           | Cmd+Shift+Enter      | <b>Insert &lt;-</b>       | Alt+-                    | Option+-                | Check Package                                                                                        | Ctrl+Shift+E         | Cmd+Shift+E         |
| <b>3 NAVIGATE CODE</b>            | <b>Windows /Linux</b>      | <b>Mac</b>           | <b>Insert %&gt;</b>       | Ctrl+Shift+M             | Cmd+Shift+M             | <b>Document Package</b>                                                                              | <b>Ctrl+Shift+D</b>  | <b>Cmd+Shift+D</b>  |
| <b>Goto File/Function</b>         | <b>Ctrl+.</b>              | <b>Ctrl+.</b>        | Show help for function    | F1                       | F1                      | <b>8 DOCUMENTS AND APPS</b>                                                                          | <b>Windows/Linux</b> | <b>Mac</b>          |
| Fold Selected                     | Alt+L                      | Cmd+Option+L         | Show source code          | F2                       | F2                      | Preview HTML (Markdown, etc.)                                                                        | Ctrl+Shift+K         | Cmd+Shift+K         |
| Unfold Selected                   | Shift+Alt+L                | Cmd+Shift+Option+L   | New document              | Ctrl+Shift+N             | Cmd+Shift+N             | <b>Knit Document (knitr)</b>                                                                         | <b>Ctrl+Shift+K</b>  | <b>Cmd+Shift+K</b>  |
| Fold All                          | Alt+O                      | Cmd+Option+O         | New document (Chrome)     | Ctrl+Alt+Shift+N         | Cmd+Shift+Alt+N         | Compile Notebook                                                                                     | Ctrl+Shift+K         | Cmd+Shift+K         |
| Unfold All                        | Shift+Alt+O                | Cmd+Shift+Option+O   | Open document             | Ctrl+O                   | Cmd+O                   | Compile PDF (TeX and Sweave)                                                                         | Ctrl+Shift+K         | Cmd+Shift+K         |
| Go to line                        | Shift+Alt+G                | Cmd+Shift+Option+G   | Save document             | Ctrl+S                   | Cmd+S                   | Insert chunk (Sweave and Knitr)                                                                      | Ctrl+Alt+I           | Cmd+Option+I        |
| Jump to                           | Shift+Alt+J                | Cmd+Shift+Option+J   | Close document            | Ctrl+W                   | Cmd+W                   | Insert code section                                                                                  | Ctrl+Shift+R         | Cmd+Shift+R         |
| Switch to tab                     | Ctrl+Shift+.               | Ctrl+Shift+.         | Close document (Chrome)   | Ctrl+Alt+W               | Cmd+Option+W            | Re-run previous region                                                                               | Ctrl+Shift+P         | Cmd+Shift+P         |
| Previous tab                      | Ctrl+F11                   | Ctrl+F11             | Close all documents       | Ctrl+Shift+W             | Cmd+Shift+W             | Run current document                                                                                 | Ctrl+Alt+R           | Cmd+Option+R        |
| Next tab                          | Ctrl+F12                   | Ctrl+F12             | Extract function          | Ctrl+Alt+X               | Cmd+Option+X            | <b>Run from start to current line</b>                                                                | <b>Ctrl+Alt+B</b>    | <b>Cmd+Option+B</b> |
| First tab                         | Ctrl+Shift+F11             | Ctrl+Shift+F11       | Extract variable          | Ctrl+Alt+V               | Cmd+Option+V            | <b>Run the current code section</b>                                                                  | <b>Ctrl+Alt+T</b>    | <b>Cmd+Option+T</b> |
| Last tab                          | Ctrl+Shift+F12             | Ctrl+Shift+F12       | Reindent lines            | Ctrl+I                   | Cmd+I                   | Run previous Sweave/Rmd code                                                                         | Ctrl+Alt+P           | Cmd+Option+P        |
| Navigate back                     | Ctrl+F9                    | Cmd+F9               | <b>(Un)Comment lines</b>  | Ctrl+Shift+C             | Cmd+Shift+C             | Run the current chunk                                                                                | Ctrl+Alt+C           | Cmd+Option+C        |
| Navigate forward                  | Ctrl+F10                   | Cmd+F10              | Reflow Comment            | Ctrl+Shift+/             | Cmd+Shift+/             | Run the next chunk                                                                                   | Ctrl+Alt+N           | Cmd+Option+N        |
| Jump to Brace                     | Ctrl+P                     | Ctrl+P               | Reformat Selection        | Ctrl+Shift+A             | Cmd+Shift+A             | Sync Editor & PDF Preview                                                                            | Ctrl+F8              | Cmd+F8              |
| Select within Braces              | Ctrl+Shift+Alt+E           | Ctrl+Shift+Alt+E     | Select within braces      | Ctrl+Shift+E             | Ctrl+Shift+E            | Previous plot                                                                                        | Ctrl+Alt+F11         | Cmd+Option+F11      |
| Use Selection for Find            | Ctrl+F3                    | Cmd+E                | Show Diagnostics          | Ctrl+Shift+Alt+P         | Cmd+Shift+Alt+P         | Next plot                                                                                            | Ctrl+Alt+F12         | Cmd+Option+F12      |
| Find in Files                     | Ctrl+Shift+F               | Cmd+Shift+F          | Transpose Letters         | Ctrl+T                   | Ctrl+T                  | <b>Show Keyboard Shortcuts</b>                                                                       | Alt+Shift+K          | Option+Shift+K      |
| Find Next                         | Win: F3, Linux: Ctrl+G     | Cmd+G                | Move Lines Up/Down        | Alt+↑/↓                  | Option+↑/↓              | <b>Why RStudio Server Pro?</b>                                                                       |                      |                     |
| Find Previous                     | W: Shift+F3, L: Ctrl+Shift | Cmd+Shift+G          | Copy Lines Up/Down        | Shift+Alt+↑/↓            | Cmd+Option+↑/↓          | Do everything you would do with the open source server with a commercial license, support, and more. |                      |                     |
| Jump to Word                      | Ctrl+←/→                   | Option+←/→           | Add New Cursor Above      | Ctrl+Alt+Up              | Ctrl+Alt+Up             |                                                                                                      |                      |                     |
| Jump to Start/End                 | Ctrl+↑/↓                   | Cmd+↑/↓              | Add New Cursor Below      | Ctrl+Alt+Down            | Ctrl+Alt+Down           |                                                                                                      |                      |                     |
|                                   |                            |                      | Move Active Cursor Up     | Ctrl+Alt+Shift+Up        | Ctrl+Alt+Shift+Up       |                                                                                                      |                      |                     |
|                                   |                            |                      | Move Active Cursor Down   | Ctrl+Alt+Shift+Down      | Ctrl+Alt+Shift+Down     |                                                                                                      |                      |                     |
|                                   |                            |                      | Find and Replace          | Ctrl+F                   | Cmd+F                   |                                                                                                      |                      |                     |
|                                   |                            |                      | Use Selection for Find    | Ctrl+F3                  | Cmd+E                   |                                                                                                      |                      |                     |
|                                   |                            |                      | Replace and Find          | Ctrl+Shift+J             | Cmd+Shift+J             |                                                                                                      |                      |                     |

# Interactive Web Apps with shiny Cheat Sheet

learn more at [shiny.rstudio.com](http://shiny.rstudio.com)



## Basics

A **Shiny** app is a web page (**UI**) connected to a computer running a live R session (**Server**)



Users can manipulate the UI, which will cause the server to update the UI's displays (by running R code).

### App template

Begin writing a new app with this template. Preview the app by running the code at the R command line.

```
library(shiny)
ui <- fluidPage()
server <- function(input, output){}
shinyApp(ui = ui, server = server)
```

- **ui** - nested R functions that assemble an HTML user interface for your app
- **server** - a function with instructions on how to build and rebuild the R objects displayed in the UI
- **shinyApp** - combines **ui** and **server** into a functioning app. Wrap with **runApp()** if calling from a sourced script or inside a function.

### Share your app

[shinyapps.io](http://shinyapps.io) The easiest way to share your app is to host it on shinyapps.io, a cloud based service from RStudio

1. Create a free or professional account at <http://shinyapps.io>
2. Click the **Publish** icon in the RStudio IDE ( $\geq 0.99$ ) or run:  
`rsconnect::deployApp("<path to directory>")`

### Build or purchase your own Shiny Server

at [www.rstudio.com/products/shiny-server/](http://www.rstudio.com/products/shiny-server/)

## Building an App

- Complete the template by adding arguments to `fluidPage()` and a body to the `server` function.

Add inputs to the UI with `*Input()` functions

Add outputs with `*Output()` functions

Tell server how to render outputs with R in the server function. To do this:

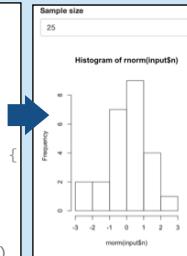
1. Refer to outputs with `output$<id>`
2. Refer to inputs with `input$<id>`
3. Wrap code in a `render*()` function before saving to output

```
library(shiny)
```

```
ui <- fluidPage(
 numericInput(inputId = "n",
 "Sample size", value = 25),
 plotOutput(outputId = "hist")
)

server <- function(input, output) {
 output$hist <- renderPlot({
 hist(rnorm(input$n))
 })
}

shinyApp(ui = ui, server = server)
```



Save your template as **app.R**. Alternatively, split your template into two files named **ui.R** and **server.R**.

```
library(shiny)
ui <- fluidPage(
 numericInput(inputId = "n",
 "Sample size", value = 25),
 plotOutput(outputId = "hist")
)

server <- function(input, output) {
 output$hist <- renderPlot({
 hist(rnorm(input$n))
 })
}

shinyApp(ui = ui, server = server)
```

```
ui.R
fluidPage(
 numericInput(inputId = "n",
 "Sample size", value = 25),
 plotOutput(outputId = "hist")
)
```

```
server.R
function(input, output) {
 output$hist <- renderPlot({
 hist(rnorm(input$n))
 })
}
```

**ui.R** contains everything you would save to ui.

**server.R** ends with the function you would save to server.

No need to call `shinyApp()`.

Save each app as a directory that contains an **app.R** file (or a **server.R** file and a **ui.R** file) plus optional extra files.

● ● ● **app-name**  
app.R  
global.R  
DESCRIPTION  
README  
<other files>  
www

- ← The directory name is the name of the app
- ← (optional) defines objects available to both ui.R and server.R
- ← (optional) used in showcase mode
- ← (optional) data, scripts, etc.
- ← (optional) directory of files to share with web browsers (images, CSS, js, etc.) Must be named "www"

Launch apps with  
`runApp(<path to directory>)`

## Outputs - render\*() and \*Output() functions work together to add R output to the UI



`DT::renderDataTable(expr, options, callback, escape, env, quoted)`



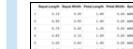
`dataTableOutput(outputId, icon, ...)`



`renderImage(expr, env, quoted, deleteFile)`



`renderPlot(expr, width, height, res, ..., env, quoted, func)`



`renderPrint(expr, env, quoted, func, width)`



`renderTable(expr, ..., env, quoted, func)`



`renderText(expr, env, quoted, func)`



`renderUI(expr, env, quoted, func)`



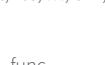
`imageOutput(outputId, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline)`



`plotOutput(outputId, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline)`



`verbatimTextOutput(outputId)`



`tableOutput(outputId)`



`textOutput(outputId, container, inline)`



`uiOutput(outputId, inline, container, ...)`



`htmlOutput(outputId, inline, container, ...)`

## Inputs - collect values from the user

Access the current value of an input object with `input $<inputId>`. Input values are **reactive**.

### Action

`actionButton(inputId, label, icon, ...)`

### Link

`actionLink(inputId, label, icon, ...)`

- Choice 1
- Choice 2
- Choice 3

`checkboxGroupInput(inputId, label, choices, selected, inline)`

- Check me

`checkboxInput(inputId, label, value)`

`dateInput(inputId, label, value, min, max, format, startview, weekstart, language)`

`dateRangeInput(inputId, label, start, end, min, max, format, startview, weekstart, language, separator)`

`fileInput(inputId, label, multiple, accept)`

`numericInput(inputId, label, value, min, max, step)`

`passwordInput(inputId, label, value)`

- Choice A
- Choice B
- Choice C

`radioButtons(inputId, label, choices, selected, inline)`

`selectInput(inputId, label, choices, selected, multiple, selectize, width, size) (also selectizeInput())`

`sliderInput(inputId, label, min, max, value, step, round, format, locale, ticks, animate, width, sep, pre, post)`

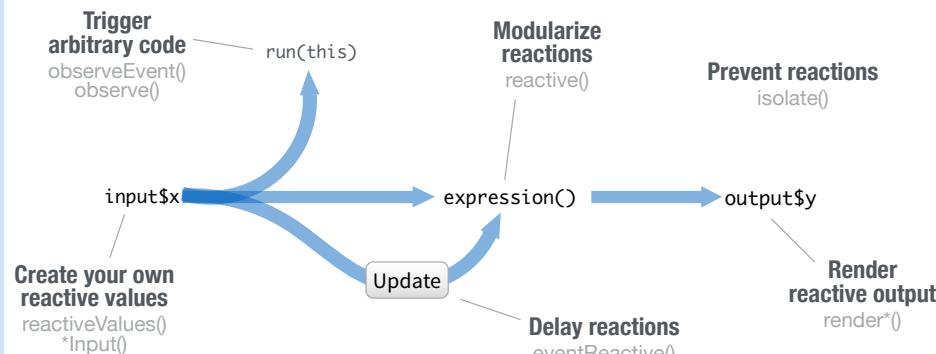
`submitButton(text, icon)`  
(Prevents reactions across entire app)

### Enter text

`textInput(inputId, label, value)`

## Reactivity

Reactive values work together with reactive functions. Call a reactive value from within the arguments of one of these functions to avoid the error [Operation not allowed without an active reactive context](#).



### Create your own reactive values

```
example snippets
ui <- fluidPage(
 textInput("a","","A"))

server <- function(input,output){
 rv <- reactiveValues()
 rv$number <- 5
}
```

**\*Input() functions**  
(see front page)

**reactiveValues(...)**

Each input function creates a reactive value stored as `input$<inputId>`

`reactiveValues()` creates a list of reactive values whose values you can set.

### Prevent reactions

```
library(shiny)
ui <- fluidPage(
 textInput("a","","A"),
 textOutput("b"))

server <- function(input,output){
 output$b <- renderText({
 isolate({input$a})
 })
}

shinyApp(ui, server)
```

### Modularize reactions

```
library(shiny)
ui <- fluidPage(
 textInput("a","","A"),
 textInput("z","","Z"),
 textOutput("b"))

server <- function(input,output){
 re <- reactive({
 paste(inputs$a,input$z)})
 output$b <- renderText({
 re()
 })
}

shinyApp(ui, server)
```

**reactive(x, env, quoted, label, domain)**

Creates a reactive expression that

- caches its value to reduce computation
- can be called by other code
- notifies its dependencies when it has been invalidated

Call the expression with function syntax, e.g. `re()`

### Render reactive output

```
library(shiny)
ui <- fluidPage(
 textInput("a","","A"),
 textOutput("b"))

server <- function(input,output){
 output$b <- renderText({
 isolate({input$a})
 })
}

shinyApp(ui, server)
```

**render\*() functions**  
(see front page)

Builds an object to display. Will rerun code in body to rebuild the object whenever a reactive value in the code changes.

Save the results to `output$<outputId>`

### Trigger arbitrary code

```
library(shiny)
ui <- fluidPage(
 textInput("a","","A"),
 actionButton("go","Go"))

server <- function(input,output){
 observeEvent(input$go,{
 print(isolate({input$a}))
 })
}

shinyApp(ui, server)
```

Runs code in 2nd argument when reactive values in 1st argument change. See `observe()` for alternative.

### Delay reactions

```
library(shiny)
ui <- fluidPage(
 textInput("a","","A"),
 actionButton("go","Go"),
 textOutput("b"))

server <- function(input,output){
 re <- eventReactive(
 isolate({input$a}),
 output$b <- renderText({
 re()
 }))
}

shinyApp(ui, server)
```

**eventReactive(eventExpr, valueExpr, event.env, event.quoted, value.env, value.quoted, label, domain, ignoreNULL)**

Creates reactive expression with code in 2nd argument that only invalidates when reactive values in 1st argument change.

## UI

An app's UI is an HTML document. Use Shiny's functions to assemble this HTML with R.

```
fluidPage(
 textInput("a","",)
)
<div class="container-fluid">
<div class="form-group shiny-input-container">
<label for="a">a</label>
<input id="a" type="text"
class="form-control" value="">
</div>
</div>
```

Returns HTML



Add static HTML elements with `tags`, a list of functions that parallel common HTML tags, e.g. `tags$a()`. Unnamed arguments will be passed into the tag; named arguments will become tag attributes.

|                               |                             |                             |                             |                             |
|-------------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| <code>tags\$a</code>          | <code>tags\$data</code>     | <code>tags\$h6</code>       | <code>tags\$nav</code>      | <code>tags\$span</code>     |
| <code>tags\$abbr</code>       | <code>tags\$list</code>     | <code>tags\$head</code>     | <code>tags\$noscript</code> | <code>tags\$strong</code>   |
| <code>tags\$address</code>    | <code>tags\$dd</code>       | <code>tags\$header</code>   | <code>tags\$object</code>   | <code>tags\$style</code>    |
| <code>tags\$area</code>       | <code>tags\$del</code>      | <code>tags\$hgroup</code>   | <code>tags\$ol</code>       | <code>tags\$sub</code>      |
| <code>tags\$article</code>    | <code>tags\$details</code>  | <code>tags\$optgroup</code> | <code>tags\$summary</code>  |                             |
| <code>tags\$aside</code>      | <code>tags\$dfn</code>      | <code>tags\$HTML</code>     |                             |                             |
| <code>tags\$audio</code>      | <code>tags\$div</code>      | <code>tags\$si</code>       | <code>tags\$output</code>   | <code>tags\$table</code>    |
| <code>tags\$b</code>          | <code>tags\$drl</code>      | <code>tags\$iframe</code>   | <code>tags\$p</code>        | <code>tags\$tbody</code>    |
| <code>tags\$base</code>       | <code>tags\$em</code>       | <code>tags\$img</code>      | <code>tags\$param</code>    | <code>tags\$thead</code>    |
| <code>tags\$bdi</code>        | <code>tags\$embed</code>    | <code>tags\$input</code>    | <code>tags\$pre</code>      | <code>tags\$textarea</code> |
| <code>tags\$blockquote</code> | <code>tags\$events</code>   | <code>tags\$ins</code>      | <code>tags\$progress</code> | <code>tags\$tfoot</code>    |
| <code>tags\$body</code>       | <code>tags\$fieldset</code> | <code>tags\$kbd</code>      | <code>tags\$q</code>        | <code>tags\$th</code>       |
| <code>tags\$br</code>         | <code>tags\$keygen</code>   | <code>tags\$keytype</code>  | <code>tags\$ruby</code>     | <code>tags\$thead</code>    |
| <code>tags\$button</code>     | <code>tags\$label</code>    | <code>tags\$label</code>    | <code>tags\$rp</code>       | <code>tags\$time</code>     |
| <code>tags\$figure</code>     | <code>tags\$legend</code>   | <code>tags\$legend</code>   | <code>tags\$rt</code>       | <code>tags\$title</code>    |
| <code>tags\$canvas</code>     | <code>tags\$li</code>       | <code>tags\$li</code>       | <code>tags\$ss</code>       | <code>tags\$track</code>    |
| <code>tags\$caption</code>    | <code>tags\$form</code>     | <code>tags\$link</code>     | <code>tags\$script</code>   | <code>tags\$u</code>        |
| <code>tags\$cite</code>       | <code>tags\$link</code>     | <code>tags\$mark</code>     | <code>tags\$section</code>  | <code>tags\$var</code>      |
| <code>tags\$code</code>       | <code>tags\$map</code>      | <code>tags\$map</code>      | <code>tags\$menu</code>     | <code>tags\$video</code>    |
| <code>tags\$h1</code>         | <code>tags\$menu</code>     | <code>tags\$meta</code>     | <code>tags\$select</code>   |                             |
| <code>tags\$h2</code>         | <code>tags\$meta</code>     | <code>tags\$small</code>    | <code>tags\$source</code>   |                             |
| <code>tags\$h3</code>         | <code>tags\$small</code>    | <code>tags\$small</code>    |                             |                             |
| <code>tags\$h4</code>         | <code>tags\$source</code>   | <code>tags\$source</code>   |                             |                             |
| <code>tags\$command</code>    | <code>tags\$h5</code>       | <code>tags\$source</code>   |                             |                             |

The most common tags have wrapper functions. You do not need to prefix their names with `tags$`

```
ui <- fluidPage(
 h1("Header 1"),
 hr(),
 br(),
 p(strong("bold")),
 p(em("italic")),
 p(code("code")),
 a(href="", "Link"),
 HTML("<p>Raw html</p>"))
```



To include a CSS file, use `includeCSS()`, or

- Place the file in the `www` subdirectory
- Link to it with

```
tags$head(tags$link(rel = "stylesheet",
 type = "text/css", href = "<file name>"))
```



To include JavaScript, use `includeScript()`

- Place the file in the `www` subdirectory
- Link to it with

```
tags$head(tags$script(src = "<file name>"))
```



To include an image

- Place the file in the `www` subdirectory
- Link to it with `img(src=<file name>)`

## Layouts

Combine multiple elements into a "single element" that has its own properties with a panel function, e.g.

```
wellPanel(
 dateInput("a", ""),
 submitButton()
)
```

`absolutePanel()` `conditionalPanel()` `fixedPanel()` `headerPanel()` `inputPanel()` `mainPanel()` `navlistPanel()` `sidebarPanel()` `tabPanel()` `tabsetPanel()` `titlePanel()` `wellPanel()`

Organize panels and elements into a layout with a layout function. Add elements as arguments of the layout functions.

### fluidRow()

```
ui <- fluidPage(
 fluidRow(column(4),
 column(2, offset = 3)),
 fluidRow(column(12)))
```

### flowLayout()

```
ui <- fluidPage(
 flowLayout(object1,
 object2,
 object3,
 object3))
```

### sidebarLayout()

```
ui <- fluidPage(
 sidebarLayout(
 sidebarPanel(),
 mainPanel()))
```

### splitLayout()

```
ui <- fluidPage(
 splitLayout(object1,
 object2))
```

### verticalLayout()

```
ui <- fluidPage(
 verticalLayout(object1,
 object2,
 object3))
```

Layer tabPanels on top of each other, and navigate between them, with:



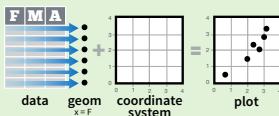
# Data Visualization with ggplot2

Cheat Sheet

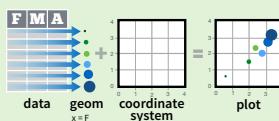


## Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data set**, a **coordinate system**, and **geoms**—visual marks that represent data points.



To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.



Complete the template below to build a graph.

```
ggplot(data = <DATA>) +
 <GEOM_FUNCTION>(
 mapping = aes(<MAPPINGS>),
 stat = <STAT>,
 position = <POSITION>
) +
 <COORDINATE_FUNCTION> +
 <FACET_FUNCTION> +
 <SCALE_FUNCTION> +
 <THEME_FUNCTION>
```

Required  
Not required, sensible defaults supplied

```
ggplot(data = mpg, aes(x = cty, y = hwy))
```

Begins a plot that you finish by adding layers to.  
Add one geom function per layer.

**aesthetic mappings**    **data**    **geom**

```
qplot(x = cty, y = hwy, data = mpg, geom = "point")
```

Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

**last\_plot()**

Returns the last plot

**ggsave("plot.png", width = 5, height = 5)**

Saves last plot as 5'x5' file named "plot.png" in working directory. Matches file type to file extension.

**Geoms** - Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

### Graphical Primitives

```
a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))
a + geom_blank()
(b + geom_curve(aes(yend = lat + 1, xend = long + 1, curvature = z)) - x, yend, alpha, angle, curvature, linetype, size)
a + geom_path(lineend = "butt", linejoin = "round", linemitre = 1)
a + geom_polygon(aes(group = group))
b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1)) - xmin, ymax, alpha, color, fill, linetype, size
a + geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900)) - ymax, ymin, alpha, color, fill, group, linetype, size
```

### Line Segments

common aesthetics: x, y, alpha, color, linetype, size

```
b + geom_abline(aes(intercept = 0, slope = 1))
b + geom_hline(aes(yintercept = lat))
b + geom_vline(aes(xintercept = long))
b + geom_segment(aes(yend = lat + 1, xend = long + 1))
b + geom_spoke(aes(angle = 1:1155, radius = 1))
```

### One Variable

**Continuous**

```
c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)
c + geom_area(stat = "bin")
c + geom_density(kernel = "gaussian")
c + geom_dotplot()
c + geom_freqpoly()
c + geom_histogram(binwidth = 5)
c2 + geom_qq(aes(sample = hwy))
```

**Discrete**

```
d <- ggplot(mpg, aes(fct))
d + geom_bar()
```

### Two Variables

#### Continuous X, Continuous Y

```
e <- ggplot(mpg, aes(cty, hwy))
e + geom_label(aes(label = cty), nudge_x = 1, check_overlap = TRUE)
e + geom_jitter(height = 2, width = 2)
e + geom_point()
e + geom_quantile()
e + geom_rug(sides = "bl")
```

```
e + geom_smooth(method = lm)
e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE)
AB
```

#### Discrete X, Continuous Y

```
f <- ggplot(mpg, aes(class, hwy))
f + geom_col()
f + geom_boxplot()
f + geom_dotplot(binaxis = "y", stackdir = "center")
f + geom_violin(scale = "area")
```

#### Discrete X, Discrete Y

```
g <- ggplot(diamonds, aes(cut, color))
g + geom_count()
```

### Three Variables

```
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))
l <- ggplot(seals, aes(long, lat))
l + geom_contour(aes(z = z))
l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE)
l + geom_tile(aes(fill = z))
```

#### Continuous Bivariate Distribution

```
h <- ggplot(diamonds, aes(carat, price))
h + geom_hex()
h + geom_bin2d(binwidth = c(0.25, 500))
h + geom_density2d()
```

```
i <- ggplot(economics, aes(date, unemploy))
i + geom_area()
i + geom_line()
i + geom_step(direction = "hv")
```

#### Visualizing error

```
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))
```

```
j + geom_crossbar(fatten = 2)
j + geom_errorbar()
j + geom_linerange()
j + geom_pointrange()
```

#### Maps

```
data <- data.frame(murder = USArrests$Murder,
state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))
k + geom_map(aes(map_id = state), map = map) +
 expand_limits(x = map$long, y = map$lat)
map_id, alpha, color, fill, linetype, size
```



# Package Development

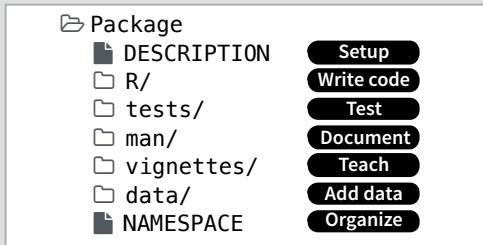
with devtools Cheat Sheet



## Package Structure

A package is a convention for organizing files into directories.

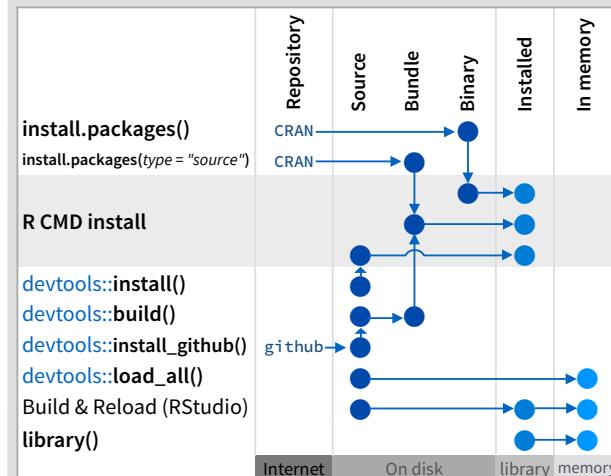
This sheet shows how to work with the 7 most common parts of an R package:



The contents of a package can be stored on disk as a:

- **source** - a directory with sub-directories (as above)
- **bundle** - a single compressed file (.tar.gz)
- **binary** - a single compressed file optimized for a specific OS

Or installed into an R library (loaded into memory during an R session) or archived online in a repository. Use the functions below to move between these states.



Adds file to .Rbuildignore, a list of files that will not be included when package is built.

## Setup (DESCRIPTION)

The `DESCRIPTION` file describes your work and sets up how your package will work with other packages.

You must have a `DESCRIPTION` file

Add the packages that yours relies on with `devtools::use_package()`

Adds a package to the Imports field (or Suggests field (if second argument is "Suggests").

**CC0**

No strings attached.

**MIT**

MIT license applies to your code if re-shared.

**GPL-2**

GPL-2 license applies to your code, and all code anyone bundles with it, if re-shared.

Package: mypackage

Title: Title of Package

Version: 0.1.0

Authors@R: person("Hadley", "Wickham", email = "hadley@me.com", role = c("aut", "cre"))

Description: What the package does (one paragraph)

Depends: R (>= 3.1.0)

License: GPL-2

LazyData: true

Imports:

dplyr (>= 0.4.0),  
ggvis (>= 0.2)

Suggests:  
knitr (>= 0.1.0)

Import packages that your package must have to work. R will install them when it installs your package.

Suggest packages that are not very essential to yours. Users can install them manually, or not, as they like.

## Write code (R/)

All of the R code in your package goes in `R/`. A package with just an `R/` directory is still a very useful package.

Create a new package project with `devtools::create("path/to/name")`

Create a template to develop into a package.

Save your code in `R/` as scripts (extension .R)

## Workflow

1. Edit your code.
2. Load your code with one of

`devtools::load_all()`

Re-loads all saved files in `R/` into memory.

**Ctrl/Cmd + Shift + L** (keyboard shortcut)

Saves all open files then calls `load_all()`.

3. Experiment in the console.
4. Repeat.

- Use consistent style with [r-pkgs.had.co.nz/r.html#style](http://r-pkgs.had.co.nz/r.html#style)
- Click on a function and press F2 to open its definition
- Search for a function with **Ctrl + .**

## Visit [r-pkgs.had.co.nz](http://r-pkgs.had.co.nz) for more

Learn more at <http://r-pkgs.had.co.nz> • devtools 1.6.1 • Updated: 1/15

RStudio® is a trademark of RStudio, Inc. • All rights reserved

[info@rstudio.com](mailto:info@rstudio.com) • 844-448-1212 • [rstudio.com](http://rstudio.com)

## Test (tests/)

Use `tests/` to store unit tests that will inform you if your code ever breaks.

Add a `tests/` directory and import `testthat` with `devtools::use_testthat()`

Sets up package to use automated tests with `testthat`

Write tests with `context()`, `test()`, and expectations

Save your tests as .R files in `tests/testthat/`

## Workflow

1. Modify your code or tests.

2. Test your code with one of

`devtools::test()`

Runs all tests saved in `tests/`.

**Ctrl/Cmd + Shift + T** (keyboard shortcut)

3. Repeat until all tests pass

`expect_equal()` is equal within small numerical tolerance?

`expect_identical()` is exactly equal?

`expect_match()` matches specified string or regular expression?

`expect_output()` prints specified output?

`expect_message()` displays specified message?

`expect_warning()` displays specified warning?

`expect_error()` throws specified error?

`expect_is()` output inherits from certain class?

`expect_false()` returns FALSE?

`expect_true()` returns TRUE?

## Example test

```
context("Arithmetic")
test_that("Math works", {
 expect_equal(1 + 1, 2)
 expect_equal(1 + 2, 3)
 expect_equal(1 + 3, 4)
})
```

## Document (📄 man/)

📄 man/ contains the documentation for your functions, the help pages in your package.

- Use roxygen comments to document each function beside its definition
- Document the name of each exported data set
- Include helpful examples for each function

### Workflow

1. Add roxygen comments in your .R files
2. Convert roxygen comments into documentation with one of

#### devtools::document()

Converts roxygen comments to .Rd files and places them in 📄 man/. Builds NAMESPACE.

#### Ctrl/Cmd + Shift + D (Keyboard Shortcut)

3. Open help pages with ? to preview documentation
4. Repeat

#### .Rd formatting tags

|                       |                                   |
|-----------------------|-----------------------------------|
| \emph{italic text}    | \email{name@foo.com}              |
| \strong{bold text}    | \href{url}{display}               |
| \code{function(args)} | \url{url}                         |
| \pkg{package}         | \link[=dest]{display}             |
| \dontrun{code}        | \linkS4class{class}               |
| \dontshow{code}       | \code{\link{function}}            |
| \donttest{code}       | \code{\link[package]{function}}   |
| \deqn{a + b (block)}  | \tabular{lcr}{                    |
| \eqn{a + b (inline)}  | left \tab centered \tab right \cr |
|                       | cell \tab cell     \tab cell \cr} |

### The roxygen package

roxygen lets you write documentation inline in your .R files with a shorthand syntax.

- Add roxygen documentation as comment lines that begin with #’.
- Place comment lines directly above the code that defines the object documented.
- Place a roxygen @ tag (right) after #’ to supply a specific section of documentation.
- Untagged lines will be used to generate a title, description, and details section (in that order)

```
#' Add together two numbers.
#'
#' @param x A number.
#' @param y A number.
#' @return The sum of \code{x} and \code{y}.
#' @examples
#' add(1, 1)
#' @export
add <- function(x, y) {
 x + y
}
```

### Common roxygen tags

|                  |                |              |
|------------------|----------------|--------------|
| @aliases         | @inheritParams | @seealso     |
| @concepts        | @keywords      | @format      |
| @describeIn      | <b>@param</b>  | @source data |
| <b>@examples</b> | @rdname        | @include     |
| <b>@export</b>   | <b>@return</b> | @slot S4     |
| @family          | @section       | @field RC    |

## Teach (📄 vignettes/)

📄 vignettes/ holds documents that teach your users how to solve real problems with your tools.

- Create a 📄 vignettes/ directory and a template vignette with devtools::use\_vignette()  
Adds template vignette as vignettes/my-vignette.Rmd.
- Append YAML headers to your vignettes (like right)
- Write the body of your vignettes in R Markdown ([rmarkdown.rstudio.com](http://rmarkdown.rstudio.com))

```

title: "Vignette Title"
author: "Vignette Author"
date: " `r Sys.Date() ` "
output: rmarkdown::html_vignette
vignette: >
 %> VignetteIndexEntry{Vignette Title}
 %> VignetteEngine{knitr::rmarkdown}
 %> usepackage[utf8]{inputenc}

```

## Add data (📄 data/)

The 📄 data/ directory allows you to include data with your package.

- Store data in one of data/, R/Sysdata.rda, inst/extdata
- Always use **LazyData: true** in your DESCRIPTION file.
- Save data as .Rdata files (suggested)

#### devtools::use\_data()

Adds a data object to data/ (R/Sysdata.rda if **internal = TRUE**)

#### devtools::use\_data\_raw()

Adds an R Script used to clean a data set to data-/raw/. Includes data.raw/ on .Rbuildignore.

#### Store data in

- **data/** to make data available to package users
- **R/sysdata.rda** to keep data internal for use by your functions.
- **inst/extdata** to make raw data available for loading and parsing examples. Access this data with **system.file()**

## Organize (📄 NAMESPACE)

The 📄 NAMESPACE file helps you make your package self-contained: it won’t interfere with other packages, and other packages won’t interfere with it.

- Export functions for users by placing **@export** in their roxygen comments
- Import objects from other packages with **package::object** (recommended) or **@import**, **@importFrom**, **@importClassesFrom**, **@importMethodsFrom** (not always recommended)

### Workflow

1. Modify your code or tests.
2. Document your package (devtools::document())
3. Check NAMESPACE
4. Repeat until NAMESPACE is correct

## Submit your package

[r-pkgs.had.co.nz/release.html](http://r-pkgs.had.co.nz/release.html)



# R Markdown Reference Guide

Learn more about R Markdown at [rmarkdown.rstudio.com](http://rmarkdown.rstudio.com)  
 Learn more about Interactive Docs at [shiny.rstudio.com/articles](http://shiny.rstudio.com/articles)

Contents:

1. **Markdown Syntax**
2. Knitr chunk options
3. Pandoc options

## Syntax

```
Plain text

End a line with two spaces
to start a new paragraph.

italics and _italics_
bold and __bold__
superscript^2^
~~strikethrough~~
[link](www.rstudio.com)

Header 1
Header 2
Header 3
Header 4
Header 5
Header 6

endash: --
emdash: ---
ellipsis: ...

inline equation: $A = \pi \times r^2$

image:

horizontal rule (or slide break):

> block quote

* unordered list
* item 2
 + sub-item 1
 + sub-item 2

1. ordered list
2. item 2
 + sub-item 1
 + sub-item 2

Table Header | Second Header
----- | -----
Table Cell | Cell 2
Cell 3 | Cell 4
```

## Becomes

```
Plain text

End a line with two spaces to start a new paragraph.

italics and italics
bold and bold
superscript2
strikethrough
link

Header 1

Header 2

Header 3

Header 4

Header 5

Header 6

endash: –

emdash: —

ellipsis: ...

inline equation: $A = \pi * r^2$

image: 

horizontal rule (or slide break):


```

### block quote

- unordered list
  - item 2
    - sub-item 1
    - sub-item 2
1. ordered list
  2. item 2
    - sub-item 1
    - sub-item 2

| Table Header | Second Header |
|--------------|---------------|
| Table Cell   | Cell 2        |
| Cell 3       | Cell 4        |



# R Markdown Reference Guide

Learn more about R Markdown at [rmarkdown.rstudio.com](http://rmarkdown.rstudio.com)

Learn more about Interactive Docs at [shiny.rstudio.com/articles](http://shiny.rstudio.com/articles)

## Contents:

1. Markdown Syntax
- 2. Knitr chunk options**
3. Pandoc options

## Syntax

Make a code chunk with three back ticks followed by an r in braces. End the chunk with three back ticks:

```
```{r}
paste("Hello", "World!")
```
```

## Becomes

Make a code chunk with three back ticks followed by an r in braces. End the chunk with three back ticks:

```
paste("Hello", "World!")
```

```
[1] "Hello World!"
```

Place code inline with a single back ticks. The first back tick must be followed by an R, like this `r paste("Hello", "World!")`.

Add chunk options within braces. For example, `echo=FALSE` will prevent source code from being displayed:

```
```{r eval=TRUE, echo=FALSE}
paste("Hello", "World!")
```
```

Place code inline with a single back ticks. The first back tick must be followed by an R, like this Hello World!.

Add chunk options within braces. For example, `echo=FALSE` will prevent source code from being displayed:

```
[1] "Hello World!"
```

Learn more about chunk options at <http://yihui.name/knitr/options>

## Chunk options

| option                 | default value | description                                                                                                                                                                                                                                                                     |
|------------------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Code evaluation</b> |               |                                                                                                                                                                                                                                                                                 |
| <b>child</b>           | NULL          | A character vector of filenames. Knitr will knit the files and place them into the main document.                                                                                                                                                                               |
| <b>code</b>            | NULL          | Set to R code. Knitr will replace the code in the chunk with the code in the code option.                                                                                                                                                                                       |
| <b>engine</b>          | 'R'           | Knitr will evaluate the chunk in the named language, e.g. <code>engine = 'python'</code> . Run <code>names(knitr::knit_engines\$get())</code> to see supported languages.                                                                                                       |
| <b>eval</b>            | TRUE          | If FALSE, knitr will not run the code in the code chunk.                                                                                                                                                                                                                        |
| <b>include</b>         | TRUE          | If FALSE, knitr will run the chunk but not include the chunk in the final document.                                                                                                                                                                                             |
| <b>purl</b>            | TRUE          | If FALSE, knitr will not include the chunk when running <code>purl()</code> to extract the source code.                                                                                                                                                                         |
| <b>Results</b>         |               |                                                                                                                                                                                                                                                                                 |
| <b>collapse</b>        | FALSE         | If TRUE, knitr will collapse all the source and output blocks created by the chunk into a single block.                                                                                                                                                                         |
| <b>echo</b>            | TRUE          | If FALSE, knitr will not display the code in the code chunk above it's results in the final document.                                                                                                                                                                           |
| <b>results</b>         | 'markup'      | If 'hide', knitr will not display the code's results in the final document. If 'hold', knitr will delay displaying all output pieces until the end of the chunk. If 'asis', knitr will pass through results without reformatting them (useful if results return raw HTML, etc.) |
| <b>error</b>           | TRUE          | If FALSE, knitr will not display any error messages generated by the code.                                                                                                                                                                                                      |
| <b>message</b>         | TRUE          | If FALSE, knitr will not display any messages generated by the code.                                                                                                                                                                                                            |
| <b>warning</b>         | TRUE          | If FALSE, knitr will not display any warning messages generated by the code.                                                                                                                                                                                                    |
| <b>Code Decoration</b> |               |                                                                                                                                                                                                                                                                                 |
| <b>comment</b>         | '##'          | A character string. Knitr will append the string to the start of each line of results in the final document.                                                                                                                                                                    |
| <b>highlight</b>       | TRUE          | If TRUE, knitr will highlight the source code in the final output.                                                                                                                                                                                                              |
| <b>prompt</b>          | FALSE         | If TRUE, knitr will add > to the start of each line of code displayed in the final document.                                                                                                                                                                                    |
| <b>strip.white</b>     | TRUE          | If TRUE, knitr will remove white spaces that appear at the beginning or end of a code chunk.                                                                                                                                                                                    |
| <b>tidy</b>            | FALSE         | If TRUE, knitr will tidy code chunks for display with the <code>tidy_source()</code> function in the <code>formatR</code> package.                                                                                                                                              |



# R Markdown Reference Guide

Learn more about R Markdown at [rmarkdown.rstudio.com](http://rmarkdown.rstudio.com)

Learn more about Interactive Docs at [shiny.rstudio.com/articles](http://shiny.rstudio.com/articles)

## Contents:

1. Markdown Syntax
- 2. Knitr chunk options**
3. Pandoc options

## Chunk options (Continued)

| option                             | default value   | description                                                                                                                                                                                                                                                                                            |
|------------------------------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Chunks</b>                      |                 |                                                                                                                                                                                                                                                                                                        |
| <b>opts.label</b>                  | NULL            | The label of options set in <code>knitr:: opts_template()</code> to use with the chunk.                                                                                                                                                                                                                |
| <b>R.options</b>                   | NULL            | Local R options to use with the chunk. Options are set with <code>options()</code> at start of chunk. Defaults are restored at end.                                                                                                                                                                    |
| <b>ref.label</b>                   | NULL            | A character vector of labels of the chunks from which the code of the current chunk is inherited.                                                                                                                                                                                                      |
| <b>Cache</b>                       |                 |                                                                                                                                                                                                                                                                                                        |
| <b>autodep</b>                     | FALSE           | If TRUE, knitr will attempt to figure out dependencies between chunks automatically by analyzing object names.                                                                                                                                                                                         |
| <b>cache</b>                       | FALSE           | If TRUE, knitr will cache the results to reuse in future knits. Knitr will reuse the results until the code chunk is altered.                                                                                                                                                                          |
| <b>cache.comments</b>              | NULL            | If FALSE, knitr will not rerun the chunk if only a code comment has changed.                                                                                                                                                                                                                           |
| <b>cache.lazy</b>                  | TRUE            | If TRUE, knitr will use <code>lazyload()</code> to load objects in chunk. If FALSE, knitr will use <code>load()</code> to load objects in chunk.                                                                                                                                                       |
| <b>cache.path</b>                  | 'cache/'        | A file path to the directory to store cached results in. Path should begin in the directory that the .Rmd file is saved in.                                                                                                                                                                            |
| <b>cache.vars</b>                  | NULL            | A character vector of object names to cache if you do not wish to cache each object in the chunk.                                                                                                                                                                                                      |
| <b>dependson</b>                   | NULL            | A character vector of chunk labels to specify which other chunks a chunk depends on. Knitr will update a cached chunk if its dependencies change.                                                                                                                                                      |
| <b>Animation</b>                   |                 |                                                                                                                                                                                                                                                                                                        |
| <b>anipots</b>                     | 'controls,loop' | Extra options for animations (see the <code>animate</code> package).                                                                                                                                                                                                                                   |
| <b>interval</b>                    | 1               | The number of seconds to pause between animation frames.                                                                                                                                                                                                                                               |
| <b>Plots</b>                       |                 |                                                                                                                                                                                                                                                                                                        |
| <b>dev</b>                         | 'png'           | The R function name that will be used as a graphical device to record plots, e.g. <code>dev='CairoPDF'</code> .                                                                                                                                                                                        |
| <b>dev.args</b>                    | NULL            | Arguments to be passed to the device, e.g. <code>dev.args=list(bg='yellow', pointsize=10)</code> .                                                                                                                                                                                                     |
| <b>dpi</b>                         | 72              | A number for knitr to use as the dots per inch (dpi) in graphics (when applicable).                                                                                                                                                                                                                    |
| <b>external</b>                    | TRUE            | If TRUE, knitr will externalize tikz graphics to save LaTex compilation time (only for the <code>tikzDevice::tikz()</code> device).                                                                                                                                                                    |
| <b>fig.align</b>                   | 'default'       | How to align graphics in the final document. One of 'left', 'right', or 'center'.                                                                                                                                                                                                                      |
| <b>fig.cap</b>                     | NULL            | A character string to be used as a figure caption in LaTex.                                                                                                                                                                                                                                            |
| <b>fig.env</b>                     | 'figure'        | The Latex environment for figures.                                                                                                                                                                                                                                                                     |
| <b>fig.ext</b>                     | NULL            | The file extension for figure output, e.g. <code>fig.ext='png'</code> .                                                                                                                                                                                                                                |
| <b>fig.height, fig.width</b>       | 7               | The width and height to use in R for plots created by the chunk (in inches).                                                                                                                                                                                                                           |
| <b>fig.keep</b>                    | 'high'          | If 'high', knitr will merge low-level changes into high level plots. If 'all', knitr will keep all plots (low-level changes may produce new plots). If 'first', knitr will keep the first plot only. If 'last', knitr will keep the last plot only. If 'none', knitr will discard all plots.           |
| <b>fig.lp</b>                      | 'fig:'          | A prefix to be used for figure labels in latex.                                                                                                                                                                                                                                                        |
| <b>fig.path</b>                    | 'figure/'       | A file path to the directory where knitr should store the graphics files created by the chunk.                                                                                                                                                                                                         |
| <b>fig.pos</b>                     | "               | A character string to be used as the figure position arrangement in LaTex.                                                                                                                                                                                                                             |
| <b>fig.process</b>                 | NULL            | A function to post-process a figure file. Should take a filename and return a filename of a new figure source.                                                                                                                                                                                         |
| <b>fig.retina</b>                  | 1               | Dpi multiplier for displaying HTML output on retina screens.                                                                                                                                                                                                                                           |
| <b>fig.scap</b>                    | NULL            | A character string to be used as a short figure caption.                                                                                                                                                                                                                                               |
| <b>fig.subcap</b>                  | NULL            | A character string to be used as captions in sub-figures in LaTex.                                                                                                                                                                                                                                     |
| <b>fig.show</b>                    | 'asis'          | If 'hide', knitr will generate the plots created in the chunk, but not include them in the final document. If 'hold', knitr will delay displaying the plots created by the chunk until the end of the chunk. If 'animate', knitr will combine all of the plots created by the chunk into an animation. |
| <b>fig.showtext</b>                | NULL            | If TRUE, knitr will call <code>showtext::showtext.begin()</code> before drawing plots.                                                                                                                                                                                                                 |
| <b>out.extra</b>                   | NULL            | A character string of extra options for figures to be passed to LaTex or HTML.                                                                                                                                                                                                                         |
| <b>out.height, out.width</b>       | NULL            | The width and height to scale plots to in the final output. Can be in units recognized by output, e.g. 8\linewidth, 50px                                                                                                                                                                               |
| <b>resize.height, resize.width</b> | NULL            | The width and height to resize tike graphics in LaTex, passed to \resizebox{}{}                                                                                                                                                                                                                        |
| <b>sanitize</b>                    | FALSE           | If TRUE, knitr will sanitize tike graphics for LaTex.                                                                                                                                                                                                                                                  |



# R Markdown Reference Guide

Learn more about R Markdown at [rmarkdown.rstudio.com](http://rmarkdown.rstudio.com)  
Learn more about Interactive Docs at [shiny.rstudio.com/articles](http://shiny.rstudio.com/articles)

## Contents:

1. Markdown Syntax
2. Knitr chunk options
3. Pandoc options

| Templates             | Basic YAML                 | Template options      | Latex options               | Interactive Docs           |
|-----------------------|----------------------------|-----------------------|-----------------------------|----------------------------|
| html_document         | ---                        | ---                   | ---                         | ---                        |
| pdf_document          | title: "A Web Doc"         | title: "Chapters"     | title: "My PDF"             | title: "Slides"            |
| word_document         | author: "John Doe"         | output:               | output: pdf_document        | output:                    |
| md_document           | date: "May 1, 2015"        | <b>html_document:</b> | <b>fontsize: 11pt</b>       | <b>slidy_presentation:</b> |
| ioslides_presentation | <b>output: md_document</b> | <b>toc: true</b>      | <b>geometry: margin=1in</b> | <b>incremental: true</b>   |
| slidy_presentation    | ---                        | <b>toc_depth: 2</b>   | ---                         | <b>runtime: shiny</b>      |
| beamer_presentation   | ---                        | ---                   | ---                         | ---                        |

## Syntax for slide formats (ioslides, slidy, beamer)

```
Dividing slides 1
Pandoc will start a new slide at each first level header
Header 2
... as well as each second level header

You can start a new slide with a horizontal rule '***' if you do not want
a header.

Bullets
Render bullets with
- a dash
- another dash

Incremental bullets
>- Use this format
>- to have bullets appear
>- one at a time (incrementally)
```

becomes

The diagram illustrates the conversion of R Markdown slide syntax into Beamer presentation slides. On the left, the R Markdown code includes sections for 'Dividing slides 1', 'Bullets', and 'Incremental bullets'. An arrow points from this code to the right, where the resulting Beamer slides are shown. The first slide, titled 'Dividing slides 1', contains the text 'Pandoc will start a new slide at each first level header ... as well as each second level header'. The second slide, titled 'Header 2', contains the text '... as well as each second level header'. The third slide, titled 'Bullets', contains the text 'Render bullets with - a dash - another dash'. The fourth slide, titled 'Incremental bullets', contains the text 'You can start a new slide with a horizontal rule \*\*\* if you do not want a header.' Below these four slides is a fifth slide with the text 'Incremental bullets' and a list: '- Use this format - to have bullets appear - one at a time (incrementally)'.

## Slide display modes

Press a key below during presentation to enter display mode. Press **esc** to exit display mode.

### ioslides

- f** - enable fullscreen mode
- w** - toggle widescreen mode
- o** - enable overview mode
- h** - enable code highlight mode
- p** - show presenter notes

### slidy

- C** - show table of contents
- F** - toggle display of the footer
- A** - toggle display of current vs all slides
- S** - make fonts smaller
- B** - make fonts bigger

## Top level options to customize LaTex (pdf) output

| option                                        | description                                                                                     |
|-----------------------------------------------|-------------------------------------------------------------------------------------------------|
| <b>lang</b>                                   | Document language code                                                                          |
| <b>fontsize</b>                               | Font size (e.g. 10pt, 11pt, 12 pt)                                                              |
| <b>documentclass</b>                          | Latex document class (e.g. article)                                                             |
| <b>classoption</b>                            | Option for document class (e.g. oneside); may be repeated                                       |
| <b>geometry</b>                               | Options for geometry class (e.g. margin=1in); may be repeated                                   |
| <b>mainfont, sansfont, monofont, mathfont</b> | Document fonts (works only with xelatex and lualatex, see the <code>latex_engine</code> option) |
| <b>linkcolor, urlcolor, citecolor</b>         | Color for internal, external, and citation links (red, green, magenta, cyan, blue, black)       |



# R Markdown Reference Guide

Learn more about R Markdown at [rmarkdown.rstudio.com](http://rmarkdown.rstudio.com)

Learn more about Interactive Docs at [shiny.rstudio.com/articles](http://shiny.rstudio.com/articles)

## Contents:

1. Markdown Syntax
2. Knitr chunk options
- 3. Pandoc options**

| option         | html | pdf | word | md | ioslides | slidy | beamer | description                                                                                                                                                                                                                                                    |
|----------------|------|-----|------|----|----------|-------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| colortheme     |      |     |      |    |          | X     |        | Beamer color theme to use (e.g., <code>colortheme: "dolphin"</code> ).                                                                                                                                                                                         |
| css            | X    |     |      |    | X        | X     |        | Filepath to CSS style to use to style document (e.g., <code>css: styles.css</code> ).                                                                                                                                                                          |
| duration       |      |     |      |    |          | X     |        | Add a countdown timer (in minutes) to footer of slides (e.g., <code>duration: 45</code> ).                                                                                                                                                                     |
| fig_caption    | X    | X   | X    |    | X        | X     | X      | Should figures be rendered with captions?                                                                                                                                                                                                                      |
| fig_crop       |      | X   |      |    |          |       | X      | Should pdfcrop utility be automatically applied to figures (when available)?                                                                                                                                                                                   |
| fig_height     | X    | X   | X    | X  | X        | X     | X      | Default figure height (in inches) for document.                                                                                                                                                                                                                |
| fig_retina     | X    |     |      | X  | X        | X     |        | Scaling to perform for retina displays (e.g., <code>fig_retina: 2</code> ).                                                                                                                                                                                    |
| fig_width      | X    | X   | X    | X  | X        | X     | X      | Default figure width (in inches) for document.                                                                                                                                                                                                                 |
| font_adjustmen |      |     |      |    |          | X     |        | Increase or decrease font size for entire presentation (e.g., <code>font_adjustment: -1</code> ).                                                                                                                                                              |
| fonttheme      |      |     |      |    |          |       | X      | Beamer font theme to use (e.g., <code>fonttheme: "structurebold"</code> ).                                                                                                                                                                                     |
| footer         |      |     |      |    |          | X     |        | Text to add to footer of each slide (e.g., <code>footer: "Copyright (c) 2014 RStudio"</code> ).                                                                                                                                                                |
| highlight      | X    | X   |      |    |          | X     | X      | Syntax highlighting style (e.g. <code>"tango"</code> , <code>"pygments"</code> , <code>"kate"</code> , <code>"zenburn"</code> , and <code>"rstudio"</code> ).                                                                                                  |
| includes       | X    | X   |      | X  | X        | X     |        | See below                                                                                                                                                                                                                                                      |
| -in_header     | X    | X   |      |    | X        | X     | X      | File of content to place in document header (e.g., <code>in_header: header.html</code> ).                                                                                                                                                                      |
| -before_body   | X    | X   |      |    | X        | X     | X      | File of content to place before document body (e.g., <code>before_body:</code>                                                                                                                                                                                 |
| -after_body    | X    | X   |      |    | X        | X     | X      | File of content to place after document body (e.g., <code>after_body: doc_suffix.html</code> ).                                                                                                                                                                |
| incremental    |      |     |      |    | X        | X     | X      | Should bullets appear one at a time (on presenter mouse clicks)?                                                                                                                                                                                               |
| keep_md        | X    |     |      |    | X        | X     |        | Save a copy of .md file that contains knitr output (in addition to the .Rmd and HTML files)?                                                                                                                                                                   |
| keep_tex       |      | X   |      |    |          | X     |        | Save a copy of .tex file that contains knitr output (in addition to the .Rmd and PDF files)?                                                                                                                                                                   |
| latex_engine   |      | X   |      |    |          |       |        | Engine to render latex. Should be one of <code>"pdflatex"</code> , <code>"xelatex"</code> , and <code>"lualatex"</code> .                                                                                                                                      |
| lib_dir        | X    |     |      |    | X        | X     |        | Directory of dependency files to use (Bootstrap, MathJax, etc.) (e.g., <code>lib_dir: libs</code> ).                                                                                                                                                           |
| logo           |      |     |      |    | X        |       |        | File path to a logo (at least 128 x 128) to add to presentation (e.g., <code>logo: logo.png</code> ).                                                                                                                                                          |
| mathjax        | X    |     |      |    | X        | X     |        | Set to <code>local</code> or a URL to use a local/URL version of MathJax to render equations                                                                                                                                                                   |
| number_section | X    | X   |      |    |          |       |        | Add section numbering to headers (e.g., <code>number_sections: true</code> ).                                                                                                                                                                                  |
| pandoc_args    | X    | X   | X    | X  | X        | X     | X      | Arguments to pass to Pandoc (e.g., <code>pandoc_args: ["--title-prefix", "Foo"]</code> ).                                                                                                                                                                      |
| preserve_yaml  |      |     |      | X  |          |       |        | Preserve YAML front matter in final document?                                                                                                                                                                                                                  |
| reference_docx |      |     | X    |    |          |       |        | A .docx file whose styles should be copied to use (e.g., <code>reference_docx:</code>                                                                                                                                                                          |
| self_contained | X    |     |      |    | X        | X     |        | Embed dependencies into the doc? Set to <code>false</code> to keep dependencies in external files.                                                                                                                                                             |
| slide_level    |      |     |      |    |          | X     |        | The lowest heading level that defines individual slides (e.g., <code>slide_level: 2</code> ).                                                                                                                                                                  |
| smaller        |      |     |      |    | X        |       |        | Use the smaller font size in the presentation?                                                                                                                                                                                                                 |
| smart          | X    |     |      |    | X        | X     |        | Convert straight quotes to curly, dashes to em-dashes, ... to ellipses, and so on?                                                                                                                                                                             |
| template       | X    | X   |      |    |          | X     | X      | Pandoc template to use when rendering file (e.g., <code>template:</code>                                                                                                                                                                                       |
| theme          |      | X   |      |    |          | X     |        | Bootswatch or Beamer theme to use for page. Valid bootswatch themes include <code>"cerulean"</code> , <code>"journal"</code> , <code>"flatly"</code> , <code>"readable"</code> , <code>"space lab"</code> , <code>"united"</code> , and <code>"cosmo"</code> . |
| toc            | X    | X   |      | X  |          | X     |        | Add a table of contents at start of document? (e.g., <code>toc: true</code> ).                                                                                                                                                                                 |
| toc_depth      | X    | X   |      | X  |          |       |        | The lowest level of headings to add to table of contents (e.g., <code>toc_depth: 2</code> ).                                                                                                                                                                   |
| transition     |      |     |      |    | X        |       |        | Speed of slide transitions should be <code>"slower"</code> , <code>"faster"</code> or a number in seconds.                                                                                                                                                     |
| variant        |      |     |      | X  |          |       |        | The flavor of markdown to use; one of <code>"markdown"</code> , <code>"markdown_strict"</code> , <code>"markdown_github"</code> , <code>"markdown_mmd"</code> , and <code>"markdown_phpextra"</code>                                                           |
| widescreen     |      |     |      | X  |          |       |        | Display presentation in widescreen format?                                                                                                                                                                                                                     |