

Cheatsheets

Czyli legalne ściąganie

Czym są cheatsheety?

Przykładowe cheatsheety



rstudio::conf

Products

Resources

Pricing

About Us

Blogs



RStudio Cheat Sheets

The cheat sheets below make it easy to learn about and use some of our favorite packages. From time to time, we will add new cheat sheets to the gallery. If you'd like us to drop you an email when we do, let us know by clicking the button to the right.

SUBSCRIBE TO CHEAT SHEET UPDATES HERE

Work with Strings Cheat Sheet

The `stringr` package provides an easy to use toolkit for working with strings, i.e. character data, in R. This cheatsheet guides you through `stringr`'s functions for manipulating strings. The back page provides a concise reference to *regular expressions*, a mini-language for describing, finding, and matching patterns in strings. Updated 10/17.

DOWNLOAD

Work with strings with stringr :: CHEAT SHEET

The `stringr` package provides a set of internally consistent tools for working with character strings, i.e. sequences of characters surrounded by quotation marks.

Detect Matches

- `str_detect(string, pattern)` Detect the presence of a pattern match in a string. `str_detect_full(1, 8)` or `str_detect(1, 8)`
- `str_which(string, pattern)` Find the indices of strings that contain a pattern match. `str_which_full(1, 8)` or `str_which(1, 8)`
- `str_count(string, pattern)` Count the number of matches in a string. `str_count_full(1, 8)` or `str_count(1, 8)`
- `str_locate(string, pattern)` Locate the positions of pattern matches in a string. Also `str_locate_all(1, 8)` or `str_locate(1, 8)`

Subset Strings

- `str_sub(string, start = 1L, end = 1L)` Extract substrings from a character vector. `str_sub_full(1, 8)` or `str_sub(1, 8)`
- `str_subset(string, pattern)` Return only the strings that contain a pattern match. `str_subset_full(1, 8)` or `str_subset(1, 8)`
- `str_extract(string, pattern)` Return the first pattern match found in each string, as a vector. Also `str_extract_all(1, 8)` or `str_extract(1, 8)`
- `str_match(string, pattern)` Return the first pattern match found in each string, as a matrix with a column for each (1) group in pattern. Also `str_match_all(1, 8)` or `str_match(1, 8)`

Manage Lengths

- `str_length(string)` The width of strings (i.e. number of code points, which generally equals the number of characters). `str_length_full(1, 8)` or `str_length(1, 8)`
- `str_pad(string, width, side = "left", "right", "both", pad = " ")` Pad strings to constant width. `str_pad_full(1, 8)` or `str_pad(1, 8)`
- `str_trunc(string, width = 4L, "right", "left", "center", ellipsis = "...")` Truncate the width of strings, replacing content with ellipsis. `str_trunc_full(1, 8)` or `str_trunc(1, 8)`
- `str_trim(string, side = c("both", "left", "right"))` Trim whitespace from the start and/or end of a string. `str_trim_full(1, 8)` or `str_trim(1, 8)`

Mutate Strings

- `str_sub() <- value` Replace substrings by identifying the substrings with `str_sub()` and assigning into the results. `str_sub_full(1, 8) <- "foo"`
- `str_replace(string, pattern, replacement)` Replace the first matched pattern in each string. `str_replace_full(1, 8) <- "foo"`
- `str_replace_all(string, pattern, replacement)` Replace all matched patterns in each string. `str_replace_all_full(1, 8) <- "foo"`
- `str_to_lower(string, locale = "en")` Convert strings to lower case. `str_to_lower_full(1, 8)` or `str_to_lower(1, 8)`
- `str_to_upper(string, locale = "en")` Convert strings to upper case. `str_to_upper_full(1, 8)` or `str_to_upper(1, 8)`
- `str_to_title(string, locale = "en")` Convert strings to title case. `str_to_title_full(1, 8)` or `str_to_title(1, 8)`

Join and Split

- `str_c(..., sep = "", collapse = NULL)` Join multiple strings into a single string. `str_c_full(1, 8)` or `str_c(1, 8)`
- `str_glue(..., sep = "", collapse = NULL)` Collapse a vector of strings into a single string. `str_glue_full(1, 8)` or `str_glue(1, 8)`
- `str_split(string, pattern, n = 1L)` Split a string from strings and (expressions) to evaluate glue glue. `str_split_full(1, 8)` or `str_split(1, 8)`
- `str_split_full(string, pattern, n = 1L)` Split a vector of strings into a matrix of substrings (splitting at occurrences of a pattern match). Also `str_split(1, 8)` or `str_split(1, 8)`

Order Strings

- `str_sort(string, decreasing = FALSE, na.last = "first", locale = "en", numeric = FALSE, ...)` Sort a character vector. `str_sort_full(1, 8)` or `str_sort(1, 8)`
- `str_order(string, decreasing = FALSE, na.last = "first", locale = "en", numeric = FALSE, ...)` Sort a character vector. `str_order_full(1, 8)` or `str_order(1, 8)`

Helpers

- `str_conv(string, encoding)` Override the encoding of a string. `str_conv_full(1, 8)` or `str_conv(1, 8)`
- `str_detect(string, pattern, match = NA)` View `NA`, encoding of first regex match in each string. `str_detect_full(1, 8)` or `str_detect(1, 8)`
- `str_view(string, pattern, match = NA)` View `NA`, encoding of all regex matches. `str_view_full(1, 8)` or `str_view(1, 8)`
- `str_wrap(string, width = 80, indent = 0, exdent = 0)` Wrap strings into nicely formatted paragraphs. `str_wrap_full(1, 8)` or `str_wrap(1, 8)`

RStudio

© 2017 RStudio, Inc. All rights reserved. RStudio is a trademark of RStudio, Inc. R is a trademark of the R Foundation for Statistical Computing. stringr is a trademark of RStudio, Inc. stringr version 1.4.0 (2017-10-17)

<https://www.rstudio.com/resources/cheatsheets/>

MI

**Dlaczego warto je tworzyć, po co są
nam one potrzebne w naszej
grupie?**

Jak je zrobić?



rstudio::conf

Products

Resources

Pricing

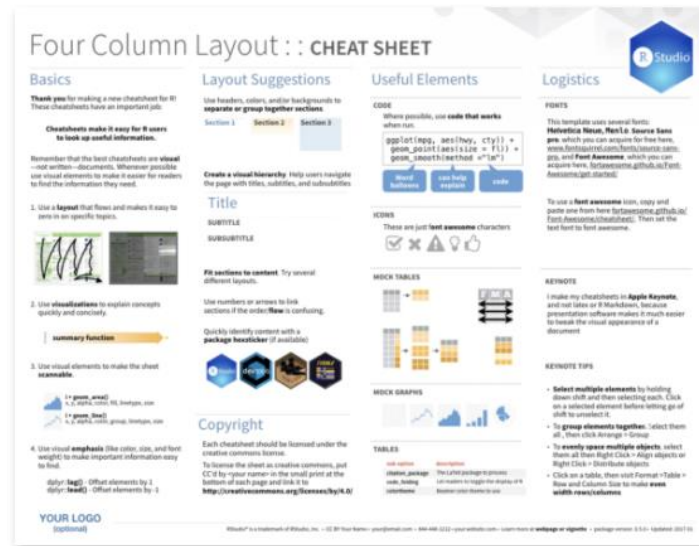
About Us

Blogs



Want to contribute a cheatsheet of your own?

We'd like to help you make and share high quality cheatsheets on R topics. The template below provides a useful starting place. It contains tips for designing a three or four column cheatsheet, as well as reusable elements to build your sheet with.



DOWNLOAD APPLE KEYNOTE ►

DOWNLOAD POWERPOINT ►

<https://www.rstudio.com/resources/cheatsheets/how-to-contribute-a-cheatsheet/>

MI

Stary szablon

Four Column layout Cheat Sheet

Your LOGO

Basics

Thank you for making a new cheatsheet for R! These cheatsheets have an important job: Cheatsheets make it easy for R users

to look up useful information.

Remember that the best cheatsheets are **visual**—not written—documents. Whenever possible use visual elements to make it easier for readers to find the information they need.

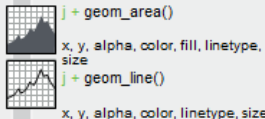
1. Use a **layout** that flows from top to bottom and left to right to make it easy to zero in on specific topics.



2. Use **visualizations** to explain concepts quickly and concisely.

summary function

3. Use **visual elements** to make the sheet scannable.



4. Use **visual emphasis** (like color, size, and weight) to make important information easy to find.

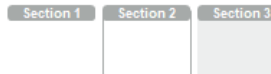
```
dplyr::bind_rows(y, z)
```

Append z to y as new rows.

Title - Group sections with titles, subtitles, and subsubtitles to create a visual hierarchy

Layout suggestions

Use headers, outlines, and/or backgrounds to **separate or group together sections**.



Use titles, subtitles, and subsubtitles to **create a visual hierarchy** that will help users navigate the page.

Title

Subtitle

Fit **sections to content**. Try several different layouts.

Use numbers or arrows to link sections if the **order/flow** is confusing.

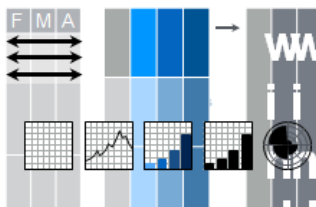
Useful elements

icons



These are just font awesome characters

Mock tables



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?

Copyright

This cheatsheet should be licensed under a Creative Commons license. The sheet as creative commons license. To use a font awesome icon, copy and paste one from here <http://fontawesome.io/> or <http://creativecommons.org/licenses/by/4.0/>

Subtitle

Example code

Where possible, use **code that works** when run.

```
dplyr::lead
Copy with values shifted by 1.
dplyr::lag
Copy with values lagged by 1.
dplyr::dense_rank
Ranks with no gaps.
dplyr::min_rank
Ranks. Ties get min rank.
dplyr::percent_rank
Ranks rescaled to [0, 1].
dplyr::row_number
Ranks. Ties got to first value.
dplyr::ntile
Bin vector into n buckets.
dplyr::between
Are values between a and b?
dplyr::cume_dist
Cumulative distribution.
```

Color Scheme

Please use the following **color scheme** when designing new cheatsheets to be distributed through <http://www.rstudio.com/resources/cheatsheets/>



Keynote

I make my cheatsheets in **Apple Keynote**, and not latex or R Markdown, because presentation software makes it much easier to tweak the visual appearance of a document

Keynote tips

- **Select multiple elements** by holding down shift and then selecting each. Click on a selected element before letting go of shift to unselect it.
- To **group elements together**, select them all, then click Arrange > Group
- To **evenly space multiple objects**, select them all then Right Click > Align objects or Right Click > Distribute objects
- Click on a table, then visit Format > Table > Row and Column Size to make **even width rows/columns**.

Code snippets

```
ggplot(mpg, aes(hwy, cty)) +
  geom_point(aes(color = cyl)) +
  geom_smooth(method = "lm") +
  coord_cartesian() +
  scale_color_gradient() +
  theme_bw()
```

Word balloons

can be useful for

explaining code

Fonts

This template uses several fonts: **Helvetica Neue**, **Menlo**, **Source Sans pro**, which you can acquire for free here, <http://www.fontsquirrel.com/fonts/source-sans-pro>, and **Font Awesome**, which you can acquire here, <http://fontawesome.io/> or <http://fontawesome.io/Font-Awesome/get-started/>. To use a **font awesome** icon, copy and paste one from here <http://fontawesome.io/> or <http://fontawesome.io/Font-Awesome/cheatsheet/>. Then set the text font to font awesome.

Składowe R-owej ściągawki

Four Column Layout : : CHEAT SHEET



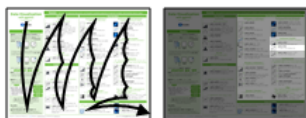
Basics

Thank you for making a new cheatsheet for R! These cheatsheets have an important job:

Cheatsheets make it easy for R users to look up useful information.

Remember that the best cheatsheets are **visual**—not written—documents. Whenever possible use visual elements to make it easier for readers to find the information they need.

1. Use a **layout** that flows and makes it easy to zero in on specific topics.



2. Use **visualizations** to explain concepts quickly and concisely.

summary function

3. Use visual elements to make the sheet **scannable**.

```
i + geom_area()
x, y, alpha, color, fill, linetype, size

i + geom_line()
x, y, alpha, color, group, linetype, size
```

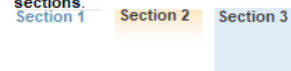
4. Use visual **emphasis** (like color, size, and font weight) to make important information easy to find.

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

YOUR LOGO
(optional)

Layout Suggestions

Use headers, colors, and/or backgrounds to **separate or group together** sections.



Create a **visual hierarchy**. Help users navigate the page with titles, subtitles, and subtitles.

Title

SUBTITLE

SUBSUBTITLE

Fit **sections to content**. Try several different layouts.

Use numbers or arrows to link sections if the **order/flow** is confusing.

Quickly identify content with a **package hexsticker** (if available)



Copyright

Each cheatsheet should be licensed under the **creative commons** license. To license the sheet as creative commons, put CC'd by <your name> in the small print at the bottom of each page and link it to <http://creativecommons.org/licenses/by/4.0/>

Useful Elements

CODE

Where possible, use **code that works** when run.

```
ggplot(mpg, aes(hwy, cty)) +
  geom_point(aes(size = fl)) +
  geom_smooth(method = "lm")
```

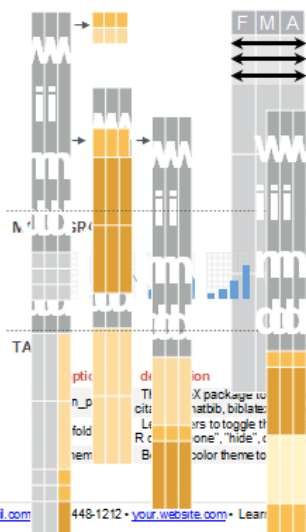
Word balloons can help explain code

ICONS

These are just **fontawesome** characters



MOCK TABLES



Logistics

FONTS

This template uses several fonts: **Helvetica Neue**, **Menlo**, **Source Sans pro**, which you can acquire for free here, www.fontsquirrel.com/fonts/source-sans-pro, and **Font Awesome**, which you can acquire here, fontawesome.github.io/Font-Awesome/get-started/

To use a **font awesome** icon, copy and paste one from here fontawesome.github.io/Font-Awesome/cheatsheet/. Then set the text font to font awesome.

KEYNOTE

I make my cheatsheets in **Apple Keynote**, and not latex or R Markdown, because presentation software makes it much easier to tweak the visual appearance of a document

KEYNOTE TIPS

- **Select multiple elements** by holding down shift and then selecting each. Click on a selected element before letting go of shift to unselect it.
- To **group elements together**, select them all, then click Arrange > Group
- To **evenly space multiple objects**, select them all then Right Click > Align objects or Right Click > Distribute objects
- Click on a table, then visit Format > Table > Row and Column Size to make **even width rows/columns**.

Praca w grupach

<https://github.com/lantanacamara/lightgbmExplainer>

<https://github.com/kapelner/ICEbox>

<https://github.com/geneticsMiNIng/cr17>