

# My Organization's First R package

Teach your package

`rstudio::conf(2020L)`





`open 05-teach/05_avalanchr.rproj`



`open module 05-teach`

## **Your Turn 1: Discussion**

**What do you do when you want to learn a new package?**

# Learning new packages

- 1 Examples
- 2 Vignettes/pkgdown
- 3 README
- 4 Blog posts
- 5 Books

# Learning new packages

- 1 **Examples**
- 2 **Vignettes/pkgdown**
- 3 **README**
- 4 **Blog posts**
- 5 **Books**

# Learning new packages

- 1 Examples
- 2 **Vignettes/pkgdown**
- 3 **README**
- 4 Blog posts
- 5 Books

## Guiding users

document	scope
Vignettes	User-friendly, deeper introductions and complex topics
README	Simple introduction, installation. Get users going
Documentation and examples	Granular, function-specific details and examples

## Examples revisited

If you don't want to run examples, wrap them in `dontrun{}` or `donttest{}`

```
## [other roxygen code]  
## @examples  
##  
## dontrun{  
##   get_data("daily_active_users")  
## }  
get_data <- function(x) {  
  ## code to get data  
}
```



## Examples revisited

Don't mess around with the user's directory. Use `tempfile()` or `fs` if you need to.

```
library(fs)  
dir <- file_temp()  
dir_create(dir)
```

```
create_package(path(dir, "temp.package"))
```

```
## ✓ Creating '/var/folders/03/9x7925g54mncswxx06wpkx100000gn/...
## ✓ Setting active project to '/private/var/folders/03/9x7925...
## ✓ Creating 'R/'
## ✓ Writing 'DESCRIPTION'

## Package: temp.package
## Title: What the Package Does (One Line, Title Case)
## Version: 0.0.0.9000
## Authors@R (parsed):
##   * Malcolm Barrett <malcolmbarrett@gmail.com> [aut, cre] (<https://,
## Description: What the package does (one paragraph).
## License: MIT + file LICENSE
## Encoding: UTF-8
## LazyData: true
## Roxygen: list(markdown = TRUE)

## ✓ Writing 'NAMESPACE'
## ✓ Writing 'temp.package.Rproj'
## ✓ Adding '.Rproj.user' to '.gitignore'
## ✓ Adding '^temp\\.package\\.Rproj$', '^\\.Rproj\\.user$' to...
## ✓ Opening '/var/folders/ 03/9x7925g54mncswxx06wpkx100000gn/...'
## ✓ Setting active project to '<no active project>'
```

# Vignettes

Long-form documentation, written in **R Markdown**

# Vignettes

Long-form documentation, written in R Markdown

**Great for general introductions and complex topics you don't want buried in the documentation**

# Vignettes

Long-form documentation, written in R Markdown

Great for general introductions and complex topics you don't want buried in the documentation

**Get rendered on pkgdown sites. See also `use_article()`**

# Vignettes

Long-form documentation, written in R Markdown

Great for general introductions and complex topics you don't want buried in the documentation

Get rendered on pkgdown sites. See also `use_article()`

`use_vignette()`

```
use_vignette("intro-to-shinRa")
```

```
shinRa
```

```
├── .Rbuildignore
├── .gitignore
├── DESCRIPTION
├── NAMESPACE
├── R/
│   ├── themes.R
├── man
│   ├── theme_mako.Rd
├── tests
│   ├── testthat
│   │   └── test-themes.R
│   └── testthat.R
├── vignettes
│   └── intro-to-shinRa.Rmd
└── shinRa.Rproj
```

```
use_vignette("intro-to-shinRa")
```

```
---
```

```
title: "intro-to-shinRa"
```

```
output: rmarkdown::html_vignette
```

```
vignette: >
```

```
%\VignetteIndexEntry{intro-to-shinRa}
```

```
  %\VignetteEngine{knitr::rmarkdown}
```

```
  %\VignetteEncoding{UTF-8}
```

```
---
```



```
use_vignette("intro-to-shinRa")
```

```
---
```

```
title: "intro-to-shinRa"
```

```
output: rmarkdown::html_vignette
```

```
vignette: >
```

```
%\VignetteIndexEntry{intro-to-shinRa}
```

```
  %\VignetteEngine{knitr::rmarkdown}
```

```
  %\VignetteEncoding{UTF-8}
```

```
---
```

## lightweight R Markdown output



```
---  
title: "intro-to-shinRa"  
output: rmarkdown::html_vignette  
vignette: >  
  
%\VignetteIndexEntry{intro-to-shinRa}  
%\VignetteEngine{knitr::rmarkdown}  
%\VignetteEncoding{UTF-8}  
---
```

# Need a Markdown refresher?

Interactive, 10-20 min tutorial:

<https://commonmark.org/help/tutorial/>

The R Markdown [website](#) or [book](#)

## Your Turn 2

**Open** vignettes/intro-to-avalanchr.Rmd.

**Let's add some more content before we knit this vignette. Lines 27-29 have an R Markdown code chunk. On line 28, fill in the blank with this code:** `db_con("residents_per_sector")`

**Let's also add some examples of the summarizing and plotting functions. On line 57, fill in the blank with** `count_donations()`. **On line 62, use** `plot_donations()`

**Knit the vignette. If you're having trouble finding some of your functions, try re-loading or documenting and re-building.**

# README

**A quick overview of your package**

# README

A quick overview of your package

**A good place for installation instructions**

# README

A quick overview of your package

A good place for installation instructions

**Becomes the homepage for a pkgdown site**

# README

A quick overview of your package

A good place for installation instructions

Becomes the homepage for a pkgdown site

`use_readme_md()` **or** `use_readme_rmd()`



# What should I put in the README?

- 1 **Badges** (?use\_badge()), where applicable.
- 2 **Installation instructions**
- 3 **A few examples**
- 4 **Maybe: how to contribute**  
(use\_tidy\_contributing())

# R Packages: **ggplot2's** README

README.md


ggplot2

build passing

build passing

coverage 81%

CRAN 3.2.1



## Overview

ggplot2 is a system for declaratively creating graphics, based on [The Grammar of Graphics](#). You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

## Installation

```
# The easiest way to get ggplot2 is to install the whole tidyverse:
install.packages("tidyverse")

# Alternatively, install just ggplot2:
install.packages("ggplot2")

# Or the development version from GitHub:
# install.packages("devtools")
devtools::install_github("tidyverse/ggplot2")
```

## Cheatsheet



## Usage

It's hard to succinctly describe how ggplot2 works because it embodies a deep philosophy of visualisation. However, in most cases you start with `ggplot()`, supply a dataset and aesthetic mapping (with `aes()`). You then add on layers (like `geom_point()` or `geom_histogram()`), scales (like `scale_colour_brewer()`), faceting specifications (like `facet_wrap()`) and coordinate systems (like `coord_flip()`).

# Other READMEs: Zhi Yang's TidyTuesday repo

README.md

Data Visualization Gallery

Twitter

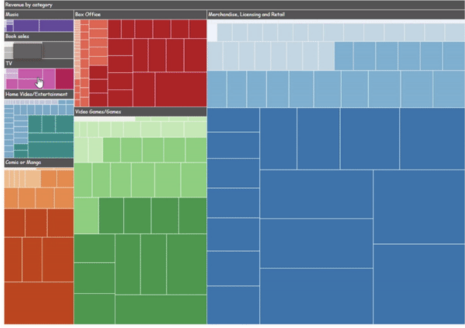
LinkedIn

GitHub

Website

► Table of Contents (click to expand)

How to make an interactive treemap using `treemap` and `d3treer`



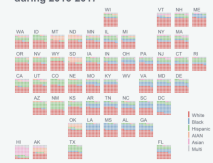
Data: introduction of #TidyTuesday media franchise data [here](#)

Code: click [here](#)

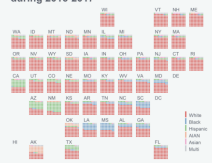
Note: to interact with the treemaps, please go to the [post](#)

How to make an geofacet waffle chart using `geofacet`

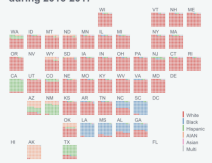
Population distribution by race/ethnicity among diverse school districts during 2016-2017



Population distribution by race/ethnicity among undiverse school districts during 2016-2017



Population distribution by race/ethnicity among extremely undiverse school districts during 2016-2017



27 / 30

# Other READMEs: Zhi Yang's TidyTuesday repo

Check out "Building a Better README"

README.md

Data Visualization Gallery

Twitter

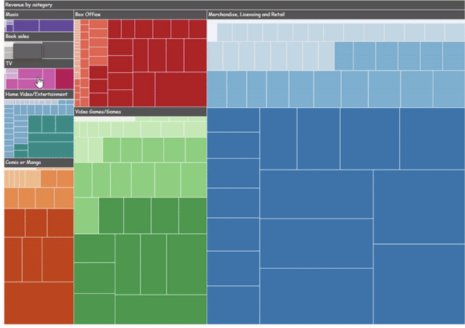
LinkedIn

GitHub

Website

► Table of Contents (click to expand)

How to make an interactive treemap using `treemap` and `d3treer`



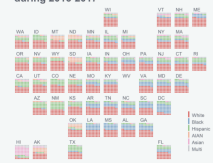
Data: introduction of #TidyTuesday media franchise data [here](#)

Code: click [here](#)

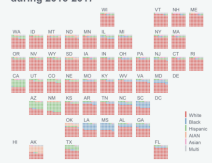
Note: to interact with the treemaps, please go to the [post](#)

How to make an geofacet waffle chart using `geofacet`

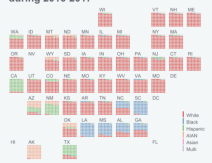
Population distribution by race/ethnicity among diverse school districts during 2016-2017



Population distribution by race/ethnicity among undiverse school districts during 2016-2017



Population distribution by race/ethnicity among extremely undiverse school districts during 2016-2017



28 / 30

## **Your Turn 3**

**Run `use_readme_rmd()` to setup a README. Knit the file and take a look.**

**Let's add some badges. Run `use_lifecycle_badge("experimental")` and `use_cran_badge()`. Then, re-knit.**

# Spell check with the **spelling** package

`use_spell_check()`