The Journal of Open Source Software

Welcome to the Tidyverse

Hadley Wickham[1], Mara Averick[1], Jennifer Bryan[1], Winston Chang[1], Lucy D'Agostino McGowan[8], Romain François[1], Garrett Grolemund[1], Alex Hayes[12], Lionel Henry[1], Jim Hester[1], Max Kuhn[1], Thomas Lin Pedersen[1], Evan Miller[13], Stephan Milton Bache[3], Kirill Müller[2], Jeroen Ooms[14], David Robinson[5], Dana Paige Seidel[10], Vitalie Spinu[4], Kohske Takahashi[9], Davis Vaughan[1], Claus Wilke[6], Kara Woo[7], and Hiroaki Yutani[11]

1 RStudio **2** cynkra **3** Redbubble **4** Erasmus University Rotterdam **5** Flatiron Health **6** Department of Integrative Biology, The University of Texas at Austin **7** Sage Bionetworks **8** Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health **9** Chukyo University, Japan **10** Department of Environmental Science, Policy, & Management, University of California, Berkeley **11** LINE Corporation **12** University of Wisconsin, Madison **13** None **14** University of California, Berkeley

DOI: 10.21105/joss.01686

Software

- Review
- Repository
- Archive

Editor: Karthik Ram

Reviewers:

- @ldecicco-USGS
- @jeffreyhanson

Submitted: 09 August 2019
Published: 21 November 2019

Summary

At a high level, the tidyverse is a language for solving data science challenges with R code. Its primary goal is to facilitate a conversation between a human and a computer about data. Less abstractly, the tidyverse is a collection of R packages that share a high-level design philosophy and low-level grammar and data structures, so that learning one package makes it easier to learn the next.

The tidyverse encompasses the repeated tasks at the heart of every data science project: data import, tidying, manipulation, visualisation, and programming. We expect that almost every project will use multiple domain-specific packages outside of the tidyverse: our goal is to provide tooling for the most common challenges; not to solve every possible problem. Notably, the tidyverse doesn't include tools for statistical modelling or communication. These toolkits are critical for data science, but are so large that they merit separate treatment. The tidyverse package allows users to install all tidyverse packages with a single command.

There are a number of projects that are similar in scope to the tidyverse. The closest is perhaps Bioconductor (Gentleman et al., 2004; Huber et al., 2015), which provides an ecosystem of packages that support the analysis of high-throughput genomic data. The tidyverse has similar goals to R itself, but any comparison to the R Project (R Core Team, 2019) is fundamentally challenging as the tidyverse is written in R, and relies on R for its infrastructure; there is no tidyverse without R! That said, the biggest difference is in priorities: base R is highly focussed on stability, whereas the tidyverse will make breaking changes in the search for better interfaces. Another closely related project is data.table (Dowle & Srinivasan, 2019), which provides tools roughly equivalent to the combination of dplyr, tidyr, tibble, and readr. data.table prioritises

concision and performance.

This paper describes the tidyverse package, the components of the tidyverse, and some of the underlying design principles. This is a lot of ground to cover in a brief paper, so we focus on a 50,000-foot view showing how all the pieces fit together with copious links to more detailed resources.

Tidyverse package

The tidyverse is a collection of packages that can easily be installed with a single "meta"-package, which is called "tidyverse". This provides a convenient way of downloading and installing all tidyverse packages with a single R command:

```
install.packages("tidyverse")
```

The core tidyverse includes the packages that you're likely to use in everyday data analyses, and these are attached when you attach the tidyverse package:

```
library(tidyverse)
#> -- Attaching packages --------------------------- tidyverse 1.2.1 --
#> v ggplot2 3.2.1     v purrr   0.3.3
#> v tibble  2.1.3     v dplyr   0.8.3
#> v tidyr   1.0.0     v stringr 1.4.0
#> v readr   1.3.1     v forcats 0.4.0
#> -- Conflicts ------------------------------ tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
```

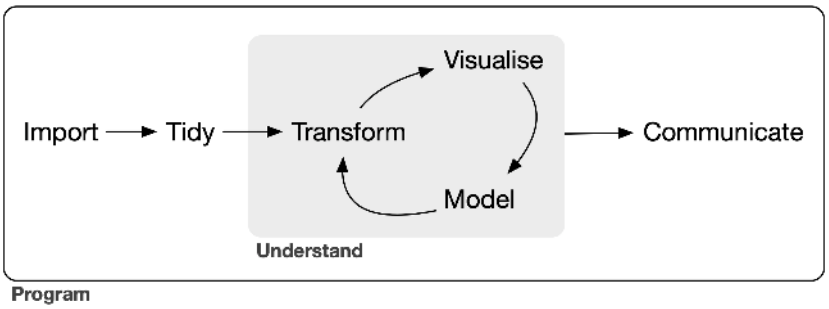
This is a convenient shortcut for attaching the core packages, produces a short report telling you which package versions you're using, and succinctly informs you of any conflicts with previously loaded packages. As of tidyverse version 1.2.0, the core packages include dplyr (Wickham et al., 2019a), forcats (Wickham, 2019a), ggplot2 (Wickham, 2016), purrr (Henry & Wickham, 2019), readr (Wickham & Hester, 2018), stringr (Wickham, 2019b), tibble (Müller & Wickham, 2018), and tidyr (Wickham & Henry, 2019).

Non-core packages are installed with `install.packages("tidyverse")`, but are not attached by `library(tidyverse)`. They play more specialised roles, so will be attached by the analyst as needed. The non-core packages are: blob (Wickham, 2018a), feather (Wickham, 2019c), jsonlite (Ooms, 2014), glue (Hester, 2018), googledrive (D'Agostino McGowan & Bryan, 2019), haven (Wickham & Miller, 2018), hms (Müller, 2018), lubridate (Spinu, Grolemund, & Wickham, 2018), magrittr (Bache & Wickham, 2014), modelr (Wickham, 2018b), readxl (Wickham & Bryan, 2019), reprex (Bryan, Hester, Robinson, & Wickham, 2019), rvest (Wickham, 2019d), and xml2 (Wickham et al., 2019b).

The tidyverse package is designed with an eye for teaching: `install.packages("tidyverse")` gets you a "batteries-included" set of 87 packages (at time of writing). This large set of dependencies means that it is not appropriate to use the tidyverse package within another package; instead, we recommend that package authors import only the specific packages that they use.

Components

How do the component packages of the tidyverse fit together? We use the model of data science tools from "R for Data Science" (Wickham & Grolemund, 2017):



Every analysis starts with data **import**: if you can't get your data into R, you can't do data science on it! Data import takes data stored in a file, database, or behind a web API, and reads it into a data frame in R. Data import is supported by the core readr (Wickham & Hester, 2018) package for tabular files (like csv, tsv, and fwf).

Additional non-core packages, such as readxl (Wickham & Bryan, 2019), haven (Wickham & Miller, 2018), googledrive (D'Agostino McGowan & Bryan, 2019), and rvest (Wickham, 2019d), make it possible to import data stored in other common formats or directly from the web.

Next, we recommend that you **tidy** your data, getting it into a consistent form that makes the rest of the analysis easier. Most functions in the tidyverse work with tidy data (Wickham, 2014), where every column is a variable, every row is an observation, and every cell contains a single value. If your data is not already in this form (almost always!), the core tidyr (Wickham & Henry, 2019) package provides tools to tidy it up.

Data **transformation** is supported by the core dplyr (Wickham et al., 2019a) package. dplyr provides verbs that work with whole data frames, such as `mutate()` to create new variables, `filter()` to find observations matching given criteria, and `left_join()` and friends to combine multiple tables. dplyr is paired with packages that provide tools for specific column types:

- stringr for strings.
- forcats for factors, R's categorical data type.
- lubridate (Spinu et al., 2018) for dates and date-times.
- hms (Müller, 2018) for clock times.

There are two main tools for understanding data: **visualisation** and **modelling**. The tidyverse provides the ggplot2 (Wickham, 2016) package for visualisation. ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics (Wilkinson, 2005).

You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details. Modelling is outside the scope of this paper, but is part of the closely affiliated tidymodels (Kuhn & Wickham, 2018) project, which shares interface design and data structures with the tidyverse.

Finally, you'll need to communicate your results to someone else. **Communication** is one of the most important parts of data science, but is not included within tidyverse. Instead, we expect people will use other R packages, like rmarkdown (Allaire et al., 2018) and shiny (Chang, Cheng, Allaire, Xie, & McPherson, 2019), which support dozens of static and dynamic output formats.

Surrounding all these tools is **programming**. Programming is a cross-cutting tool that you use in every part of a data science project. Programming tools in the tidyverse include:

- purrr (Henry & Wickham, 2019), which enhances R's functional programming toolkit.
- tibble (Müller & Wickham, 2018), which provides a modern re-imagining of the venerable data frame, keeping what time has proven to be effective, and throwing out what it has not.

- reprex (Bryan et al., 2019), which helps programmers get help when they get stuck by easing the creation of reproducible examples.
- magrittr (Bache & Wickham, 2014), which provides the pipe operator, %>%, used throughout the tidyverse. The pipe is a tool for function composition, making it easier to solve large problems by breaking them into small pieces.

Design principles

We are still working to explicitly describe the unifying principles that make the tidyverse consistent, but you can read our latest thoughts at https://design.tidyverse.org/. There is one particularly important principle that we want to call out here: the tidyverse is fundamentally **human centred**. That is, the tidyverse is designed to support the activities of a human data analyst, so to be effective tool builders, we must explicitly recognise and acknowledge the strengths and weaknesses of human cognition.

This is particularly important for R, because it's a language that's used primarily by non-programmers, and we want to make it as easy as possible for first-time and end-user programmers to learn the tidyverse. We believe deeply in the motivations that lead to the creation of S: "to turn ideas into software, quickly and faithfully" (Chambers, 1998). This means that we spend a lot of time thinking about interface design, and have recently started experimenting with surveys to help guide interface choices.

Similarly, the tidyverse is not just the collection of packages — it is also the community of people who use them. We want the tidyverse to be a diverse, inclusive, and welcoming community. We are still developing our skills in this area, but our existing approaches include active use of Twitter to solicit feedback, announce updates, and generally listen to the community. We also keep users apprised of major upcoming changes through the tidyverse blog, run developer days, and support lively discussions on RStudio community.

Acknowledgments

The tidyverse would not be possible without the immense work of the R-core team who maintain the R language and we are deeply indebted to them. We are also grateful for the financial support of RStudio, Inc.

References

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., et al. (2018). *rmarkdown: Dynamic documents for R*. Retrieved from https://rmarkdown.rstudio.com

Bache, S. M., & Wickham, H. (2014). *magrittr: A forward-pipe operator for R*. Retrieved from https://CRAN.R-project.org/package=magrittr

Bryan, J., Hester, J., Robinson, D., & Wickham, H. (2019). *reprex: Prepare reproducible example code via the clipboard*. Retrieved from https://CRAN.R-project.org/package=reprex

Chambers, J. M. (1998). *Programming with data: A guide to the S language*. Springer.

Chang, W., Cheng, J., Allaire, J., Xie, Y., & McPherson, J. (2019). *shiny: Web application framework for R*. Retrieved from https://CRAN.R-project.org/package=shiny

D'Agostino McGowan, L., & Bryan, J. (2019). *googledrive: An interface to google drive*. Retrieved from https://CRAN.R-project.org/package=googledrive

Dowle, M., & Srinivasan, A. (2019). *data.table: Extension of 'data.frame'*. Retrieved from https://CRAN.R-project.org/package=data.table

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., et al. (2004). Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, *5*(10), R80.

Henry, L., & Wickham, H. (2019). *purrr: Functional programming tools*. Retrieved from https://CRAN.R-project.org/package=purrr

Hester, J. (2018). *glue: Interpreted string literals*. Retrieved from https://CRAN.R-project.org/package=glue

Huber, W., Carey, V. J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B. S., Bravo, H. C., et al. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods*, *12*(2), 115–121. Retrieved from http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html

Kuhn, M., & Wickham, H. (2018). *tidymodels: Easily install and load the 'tidymodels' packages*. Retrieved from https://CRAN.R-project.org/package=tidymodels

Müller, K. (2018). *hms: Pretty time of day*. Retrieved from https://CRAN.R-project.org/package=hms

Müller, K., & Wickham, H. (2018). *tibble: Simple data frames*. Retrieved from https://CRAN.R-project.org/package=tibble

Ooms, J. (2014). The jsonlite package: A practical and consistent mapping between JSON data and R objects. *arXiv:1403.2805 [stat.CO]*. Retrieved from https://arxiv.org/abs/1403.2805

R Core Team. (2019). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Spinu, V., Grolemund, G., & Wickham, H. (2018). *lubridate: Make dealing with dates a little easier*. Retrieved from https://CRAN.R-project.org/package=lubridate

Wickham, H. (2014). Tidy data. *The Journal of Statistical Software*, *59*.

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. useR. Springer.

Wickham, H. (2018a). *blob: A simple s3 class for representing vectors of binary data ('blobs')*. Retrieved from https://CRAN.R-project.org/package=blob

Wickham, H. (2018b). *modelr: Modelling functions that work with the pipe*. Retrieved from https://CRAN.R-project.org/package=modelr

Wickham, H. (2019a). *forcats: Tools for working with categorical variables (factors)*. Retrieved from https://CRAN.R-project.org/package=forcats

Wickham, H. (2019b). *stringr: Simple, consistent wrappers for common string operations*. Retrieved from https://CRAN.R-project.org/package=stringr

Wickham, H. (2019c). *Feather: R bindings to the feather API*. Retrieved from https://CRAN.R-project.org/package=feather

Wickham, H. (2019d). *rvest: Easily harvest (scrape) web pages*. Retrieved from https://CRAN.R-project.org/package=rvest

Wickham, H., & Bryan, J. (2019). *readxl: Read excel files*. Retrieved from https://CRAN.R-project.org/package=readxl

Wickham, H., François, R., Henry, L., & Müller, K. (2019a). *dplyr: A grammar of data manipulation*. Retrieved from https://CRAN.R-project.org/package=dplyr

Wickham, H., & Grolemund, G. (2017). *R for data science: Import, tidy, transform, visualize, and model data*. O'Reilly Media, Inc.

Wickham, H., & Henry, L. (2019). *tidyr: Tidy messy data*. Retrieved from https://CRAN.R-project.org/package=tidyr

Wickham, H., & Hester, J. (2018). *readr: Read rectangular text data*. Retrieved from https://CRAN.R-project.org/package=readr

Wickham, H., Hester, J., & Ooms, J. (2019b). *xml2: Parse XML*. Retrieved from https://CRAN.R-project.org/package=xml2

Wickham, H., & Miller, E. (2018). *haven: Import and export SPSS, Stata, and SAS files*. Retrieved from https://CRAN.R-project.org/package=haven

Wilkinson, L. (2005). *The grammar of graphics*. Berlin, Heidelberg: Springer-Verlag. doi:10.1007/0-387-28695-0