# Chapter 35: tidyverse

## Section 35.1: tidyverse: an overview

**What is `tidyverse`?**

`tidyverse` is the fast and elegant way to turn basic `R` into an enhanced tool, redesigned by Hadley/Rstudio. The development of all packages included in `tidyverse` follow the principle rules of The tidy tools manifesto. But first, let the authors describe their masterpiece:

> The tidyverse is a set of packages that work in harmony because they share common data representations and API design. The tidyverse package is designed to make it easy to install and load core packages from the tidyverse in a single command.
>
> The best place to learn about all the packages in the tidyverse and how they fit together is R for Data Science. Expect to hear more about the tidyverse in the coming months as I work on improved package websites, making citation easier, and providing a common home for discussions about data analysis with the tidyverse.
>
> (source))

**How to use it?**

Just with the ordinary `R` packages, you need to install and load the package.

```
install.package("tidyverse")
library("tidyverse")
```

The difference is, on a single command a couple of dozens of packages are installed/loaded. As a bonus, one may rest assured that all the installed/loaded packages are of compatible versions.

**What are those packages?**

The commonly known and widely used packages:

- ggplot2: advanced data visualisation SO_doc
- dplyr: fast (Rcpp) and coherent approach to data manipulation SO_doc
- tidyr: tools for data tidying SO_doc
- readr: for data import.
- purrr: makes your pure functions purr by completing R's functional programming tools with important features from other languages, in the style of the JS packages underscore.js, lodash and lazy.js.
- tibble: a modern re-imagining of data frames.
- magrittr: piping to make code more readable SO_doc

Packages for manipulating specific data formats:

- hms: easily read times
- stringr: provide a cohesive set of functions designed to make working with strings as easy as posssible
- lubridate: advanced date/times manipulations SO_doc
- forcats: advanced work with factors.

Data import:

- DBI: defines a common interface between the R and database management systems (DBMS)
- haven: easily import SPSS, SAS and Stata files SO_doc
- httr: the aim of httr is to provide a wrapper for the curl package, customised to the demands of modern web APIs
- jsonlite: a fast JSON parser and generator optimized for statistical data and the web
- readxl: read.xls and .xlsx files without need for dependency packages SO_doc
- rvest: rvest helps you scrape information from web pages SO_doc
- xml2: for XML

And modelling:

- modelr: provides functions that help you create elegant pipelines when modelling
- broom: easily extract the models into tidy data

Finally, `tidyverse` suggest the use of:

- knitr: the amazing general-purpose literate programming engine, with lightweight API's designed to give users full control of the output without heavy coding work. SO_docs: one, two
- rmarkdown: Rstudio's package for reproducible programming. SO_docs: one, two, three, four

# Section 35.2: Creating tbl_df's

A tbl_df (pronounced *tibble diff*) is a variation of a data frame that is often used in tidyverse packages. It is implemented in the tibble package.

Use the `as_data_frame` function to turn a data frame into a tbl_df:

```
library(tibble)
mtcars_tbl <- as_data_frame(mtcars)
```

One of the most notable differences between data.frames and tbl_dfs is how they print:

```
# A tibble: 32 x 11
     mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
*  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1   21.0     6 160.0   110  3.90 2.620 16.46     0     1     4     4
2   21.0     6 160.0   110  3.90 2.875 17.02     0     1     4     4
3   22.8     4 108.0    93  3.85 2.320 18.61     1     1     4     1
4   21.4     6 258.0   110  3.08 3.215 19.44     1     0     3     1
5   18.7     8 360.0   175  3.15 3.440 17.02     0     0     3     2
6   18.1     6 225.0   105  2.76 3.460 20.22     1     0     3     1
7   14.3     8 360.0   245  3.21 3.570 15.84     0     0     3     4
8   24.4     4 146.7    62  3.69 3.190 20.00     1     0     4     2
9   22.8     4 140.8    95  3.92 3.150 22.90     1     0     4     2
10  19.2     6 167.6   123  3.92 3.440 18.30     1     0     4     4
# ... with 22 more rows
```

- The printed output includes a summary of the dimensions of the table (`32 x 11`)
- It includes the type of each column (`dbl`)
- It prints a limited number of rows. (To change this use `options(tibble.print_max = [number])`).

Many functions in the dplyr package work naturally with tbl_dfs, such as `group_by()`.