R-Courses Training: Day 02

Introduction to tidyverse

Presented by Abbas Rizvi

Outline of today

1. Introduction to tidyverse
2. magrittr and the pipe!
3. Wrangling data with dplyr
4. Reshaping data with tidyr
5. Importing data with readr
6. Hands-on tutorial
    a. Interactively walk through RMarkdown (.Rmd) on RStudio Cloud

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

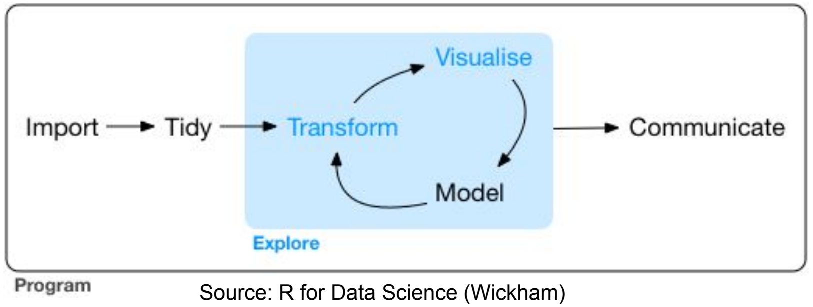
tidyverse

Abstractly: conversation between human and computer about data

Less abstractly: collection of R packages

- High-level design philosophy
- Low-level grammar and data structures

Application programming interface (API) that is at heart of every data science project.



Source: R for Data Science (Wickham)

What comes with the tidyverse?

dplyr, tidyr, readr, purrr, tibble

```
install.packages("tidyverse")

# is equivalent to

install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyr")
install.packages("readr")
install.packages("purrr")
install.packages("tibble")
```

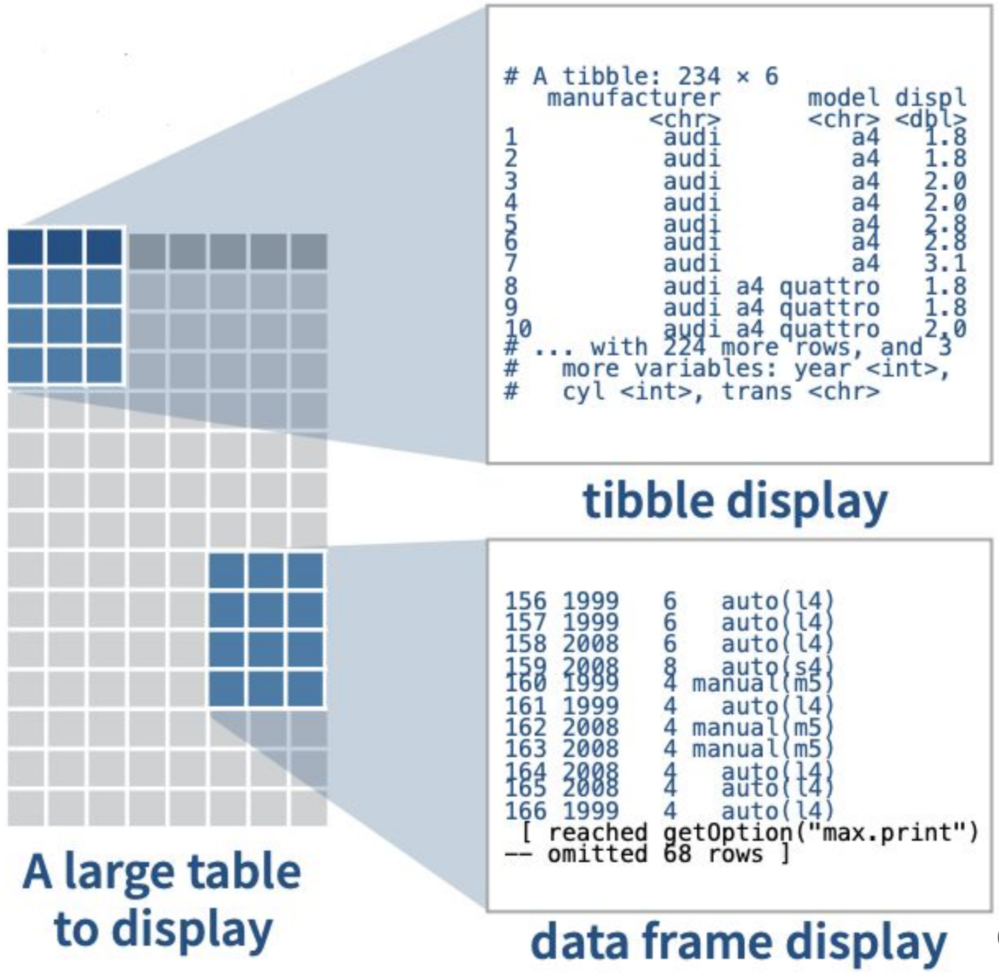
```
library(tidyverse)

# is equivalent to

library(ggplot2)
library(dplyr)
library(tidyr)
library(readr)
library(purrr)
library(tibble)
```

tibbles

The **tibble** R package that provides an “enhanced” data frame for storing and printing tabular data.

View full dataset

- **View()**
- **glimpse()**



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

```
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 2008   4    auto(l4)
166 1999   4    auto(l4)
 [ reached getOption("max.print")
-- omitted 68 rows ]
```

Create sample data with tibble

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)
convert named vector to a tibble

is_tibble(x)
test whether x is a tibble

Example dataset

Datasets

Two datasets:

1. `data(who)` - World Health Organization Tuberculosis (TB) Report 1960-2013
   a. data.frame with 7240 rows and columns
      i. country - country name
      ii. iso2, iso3 - ISO country codes, 2 letter and 3 letter abbreviation
      iii. year - year
      iv. new_sp_m014-new_relf65 - counts of new TB cases recorded by group
         1. column names encode three variables that describe the group
            a. Details available here
2. `data(populations)` - Accompanying Global Populations
   a. country - country name
   b. year - year
   c. population - nominal value of population

%>%
Ceci n'est pas un pipe.
www.tidyverse.org

magrittr and the pipe!

The pipe operator `%>%` offers to make your code more readable

- structure data operations from left-to-right (as opposed to inside-out)
- avoid nested function calls
- minimize need for local variables and functions
- easy to add/remove steps in sequence of operations

1929 Painting by René Magritte

Pipe basic usage

Basic usage:
`x %>% f` is equivalent to `f(x)`
`x %>% f(y)` is equivalent to `f(x, y)`
`x %>% f %>% g %>% h` is equivalent to `h(g(f(x)))`

“Equivalent” is technically not exact, evaluation is non-standard, and left-hand side is evaluated before right-hand side of expression.

The argument placeholder
`x %>% f(y, .)` is equivalent to `f(y,x)`
`x %>% f(y, z = .)` is equivalent to `f(y, z = x)`

Pipes example

tidyverse syntax with magrittr pipes

```
> iris %>%                              ← data
    filter(Species=="virginica") %>%    ← function(argument)
    head()                              ← function(no argument)
  Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
1          6.3         3.3          6.0         2.5 virginica
2          5.8         2.7          5.1         1.9 virginica
3          7.1         3.0          5.9         2.1 virginica
4          6.3         2.9          5.6         1.8 virginica
5          6.5         3.0          5.8         2.2 virginica
6          7.6         3.0          6.6         2.1 virginica
```

nested base R syntax

```
> head(filter(iris, Species=="virginica"))
```

```r
install.packages("dplyr")
library(dplyr)
```

Data transformation with dplyr

dplyr (https://dplyr.tidyverse.org/)

Grammar for data wrangling; consistent set of verbs to help solve most common problems

Key functions:

- Isolate/extract/manipulate data
- Group and summarize cases
- Combine tables (joins)

```r
install.packages("tidyr")
library(tidyr)
```

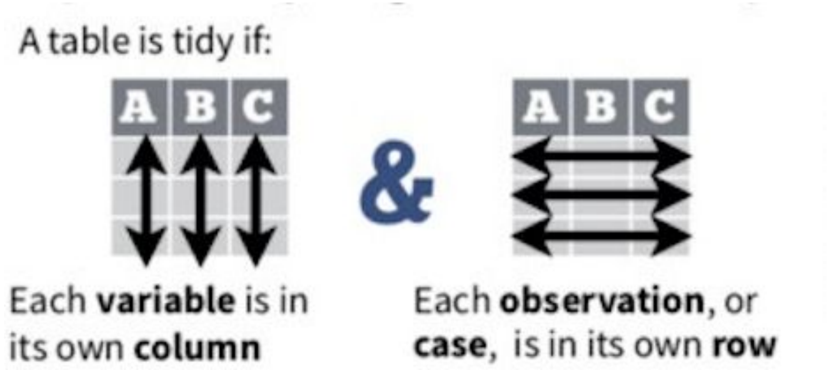
Tidy data with tidyr

tidyr (tidyr.tidyverse.org)

1. Every column is variable.
2. Every row is an observation.
3. Every cell is a single value

Key package features:

- Reshape data
- Split cells
- Handle missing values
- Expand tables


A table is tidy if:
A B C
&
A B C
Each **variable** is in its own **column**
Each **observation**, or **case**, is in its own **row**

readr

www.rstudio.com

```r
install.packages("readr")
library(readr)
```

Reading tabular and non-tabular data

The goal of readr is provide a fast and friendly way to read rectangular data. It was designed to be flexible and parse many types of data found in the wild

Key features:

- It has seven different `read_` functions
- Can declare explicit column data types.
- Has ways to read files line by line or chunks

Other types of data

Try one of the following packages:

- haven - SAS, SPSS, Stata files
- readxl - MS excel files (.xlsx and .xls)
- DBI - databases (SQL, Postgres)
- jsonlite - json files
- xml2 - XML files
- httr - connect with web API and curl commands
- rvest - for scraping HTML (Web scraping)

Now time for
hands on tutorial!