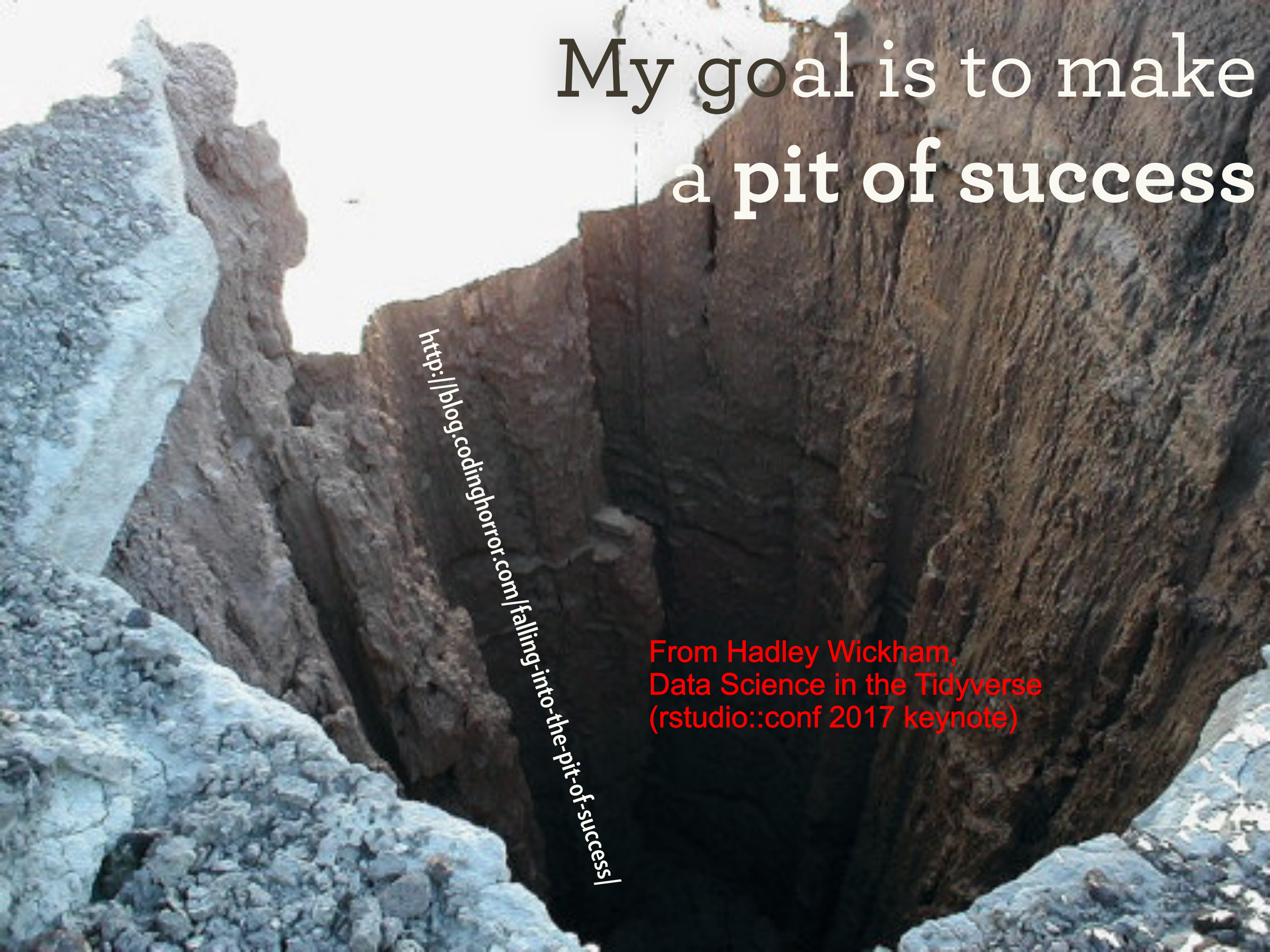
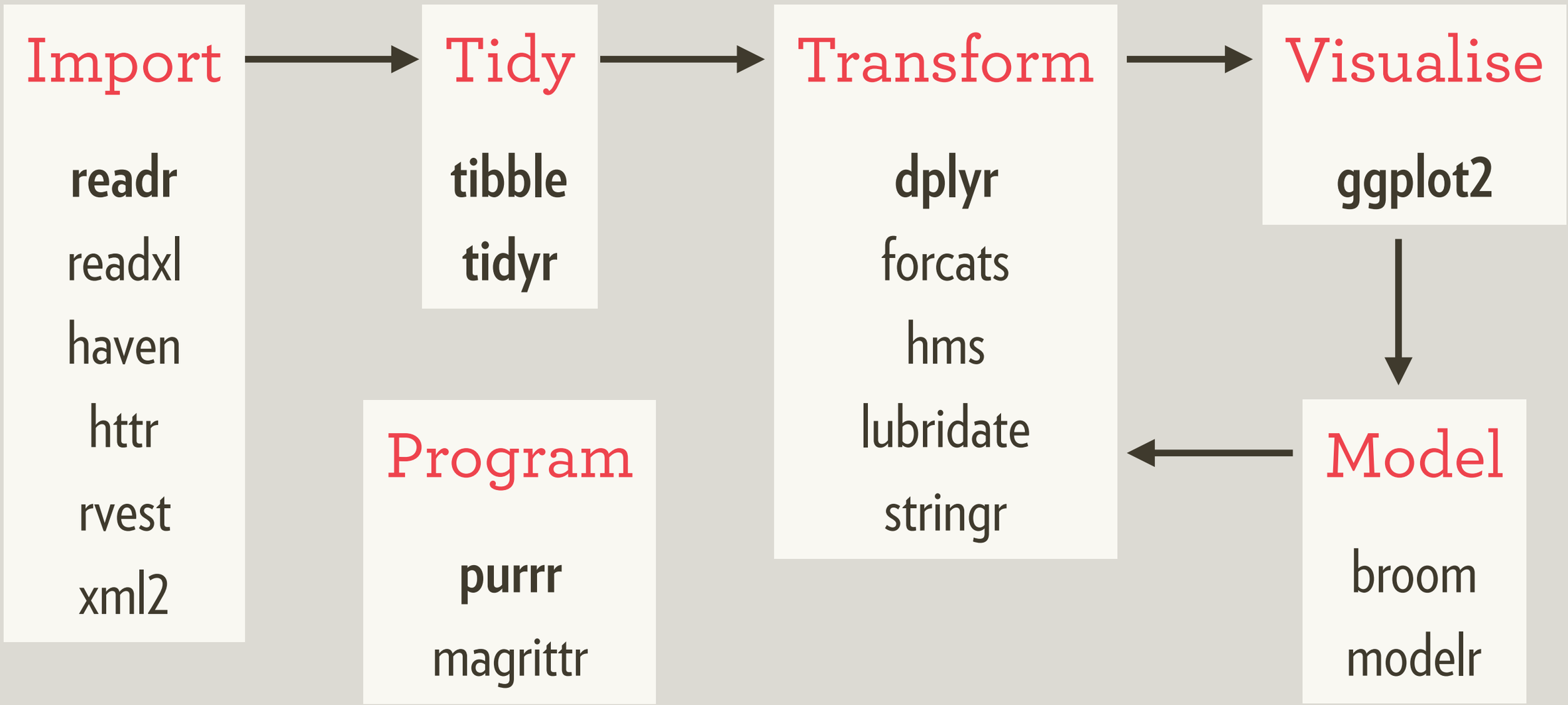

My goal is to make
a **pit of success**
http://blog.codinghorror.com/falling-into-the-pit-of-success/
From Hadley Wickham,
Data Science in the Tidyverse
(rstudio::conf 2017 keynote)


Import
readr
readxl
haven
httr
rvest
xml2
Tidy
tibble
tidyr
Transform
dplyr
forcats
hms
lubridate
stringr
Visualise
ggplot2
Model
broom
modelr
Program
purrr
magrittr

Installing tidyverse installs everything

```
install.packages("tidyverse")

# Instead of
install.packages(c(
  "broom", "dplyr", "feather",
  "forcats","ggplot2", "haven",
  "httr", "hms", "jsonlite",
  "lubridate", "magrittr",
  "modelr", "purrr", "readr",
  "readxl", "stringr", "tibble",
  "rvest", "tidyr", "xml2"
))
```

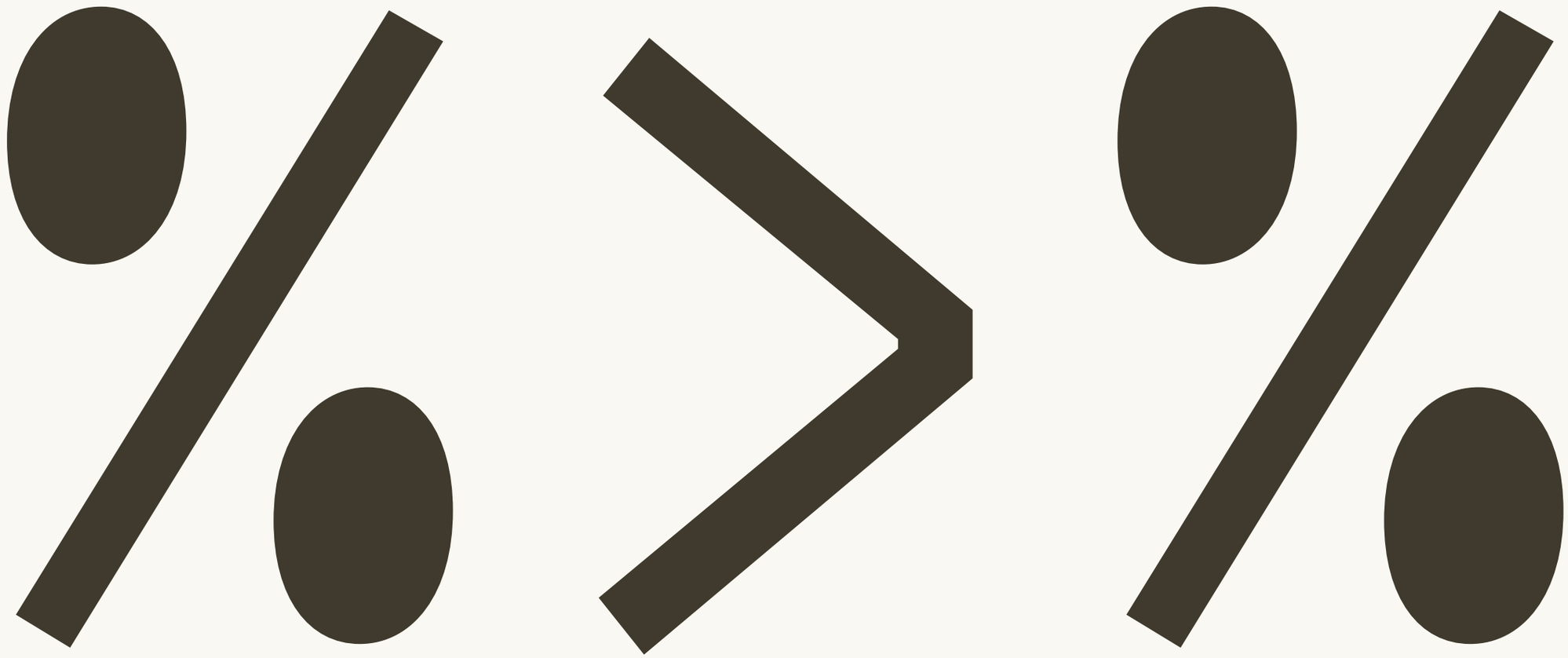
Loading it loads the **core** tidyverse

```
library(tidyverse)

# Instead of:
library(ggplot2)
library(tibble)
library(tidyr)
library(readr)
library(purrr)
library(dplyr)

# These are the packages you use in almost
# every analysis
```

magrittr::



The command-query distinction is useful for pipes

The body is made up of **queries**

Every pipe is ended by a **command**

Where is the command function?

```
flights %>%
  group_by(dest) %>%
  summarise(
    delay = mean(dep_delay, na.rm = TRUE),
    n = n()
) %>%
filter(n > 100) %>%
arrange(desc(delay))
```

In the absence of a command, R prints

```
flights %>%
  group_by(dest) %>%
  summarise(
    delay = mean(dep_delay, na.rm = TRUE),
    n = n()
) %>%
filter(n > 100) %>%
arrange(desc(delay)) %>%
print()
```

Another common command is **assign**

```
flights %>%
  group_by(dest) %>%
  summarise(
    delay = mean(dep_delay, na.rm = TRUE),
    n = n()
) %>%
filter(n > 100) %>%
arrange(desc(delay)) ->
dest_delays
```

But leading with assignment improves readability

```
dest_delays <- flights %>%
  group_by(dest) %>%
  summarise(
    delay = mean(dep_delay, na.rm = TRUE),
    n = n()
  ) %>%
  filter(n > 100) %>%
  arrange(desc(delay))
```

Functions fit best into a pipe when:

1. The first argument is the “data”
2. The data is the same type across a family of functions

Tidy data

Goal: Solve complex problems by combining simple, **uniform pieces**.

Tidy data is a consistent way of storing data

1. Each **dataset** goes in a **data frame**.

2. Each **variable** goes in a **column**.

Happy families are all alike;
every unhappy family is
unhappy in its own way

— *Leo Tolstoy*

Tidy datasets are all alike;
every messy dataset is
messy in its own way

— *Hadley Wickham*

Messy data has a varied shape

```
# A tibble: 5,769 × 22
   iso2  year   m04  m514  m014 m1524 m2534 m3544 m4554 m5564   m65    mu   f04  f514  f014 f1524
   <chr> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
1  AD    1989    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
2  AD    1990    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
3  AD    1991    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
4  AD    1992    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
5  AD    1993    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
6  AD    1994    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
7  AD    1996    NA    NA     0     0     0     4     1     0     0    NA    NA    NA     0     1
8  AD    1997    NA    NA     0     0     1     2     2     1     6    NA    NA    NA     0     1
9  AD    1998    NA    NA     0     0     0     1     0     0     0    NA    NA    NA    NA    NA
10 AD    1999    NA    NA     0     0     0     1     1     0     0    NA    NA    NA     0     0
11 AD    2000    NA    NA     0     0     1     0     0     0     0    NA    NA    NA    NA    NA
12 AD    2001    NA    NA     0    NA    NA     2     1    NA    NA    NA    NA    NA    NA    NA
13 AD    2002    NA    NA     0     0     0     1     0     0     0    NA    NA    NA     0     1
14 AD    2003    NA    NA     0     0     0     1     2     0     0    NA    NA    NA     0     1
15 AD    2004    NA    NA     0     0     0     1     1     0     0    NA    NA    NA     0     0
16 AD    2005     0     0     0     0     1     1     0     0     0     0     0     0     0     1
17 AD    2006     0     0     0     1     1     2     0     1     1     0     0     0     0     0
# ... with 5,752 more rows, and 6 more variables: f2[illegible]
#   f5564 <int>, f65 <int>, fu <int>
```

What are the variables in this dataset?

(Hint: f = female, u = unknown, 1524 = 15-24)

Tidy data has a uniform shape

```
# A tibble: 35,750 × 5
   country  year   sex   age       n
     <chr> <int> <chr> <chr> <int>
1       AD  1996     f   014       0
2       AD  1996     f  1524       1
3       AD  1996     f  2534       1
4       AD  1996     f  3544       0
5       AD  1996     f  4554       0
6       AD  1996     f  5564       1
7       AD  1996     f    65       0
8       AD  1996     m   014       0
9       AD  1996     m  1524       0
10      AD  1996     m  2534       0
# ... with 35,740 more rows
```

tidytext

by Julia Silge & David Robinson

The family of Dashwood had long been settled in Sussex. Their estate was large, and their residence was at Norland Park, in the centre of their property, where, for many generations, they had lived in so respectable a manner as to engage the general good opinion of their surrounding acquaintance.

— *Sense & Sensibility*, **Jane Austen**

tidytext provides an answer

```
# A tibble: 724,880 × 4
                  book linenumber chapter     word
                <fctr>      <int>   <int>    <chr>
1  Sense & Sensibility         10       1  chapter
2  Sense & Sensibility         10       1        1
3  Sense & Sensibility         13       1      the
4  Sense & Sensibility         13       1   family
5  Sense & Sensibility         13       1       of
6  Sense & Sensibility         13       1 dashwood
7  Sense & Sensibility         13       1      had
8  Sense & Sensibility         13       1     long
9  Sense & Sensibility         13       1     been
10 Sense & Sensibility         13       1  settled
# ... with 724,870 more rows
```