Biodiversity data analysis workshop – Day 2

Dominic Henry & Lizanne Roxburgh

Conservation Science Unit

dominich@ewt.org.za

lizanner@ewt.org.za

ENDANGERED
WILDLIFE TRUST
Protecting forever, together.

science
& technology
Department:
Science and Technology
REPUBLIC OF SOUTH AFRICA

NRF
National
Research
Foundation

SANBI
Biodiversity for Life
South African National Biodiversity Institute

31st October 2019

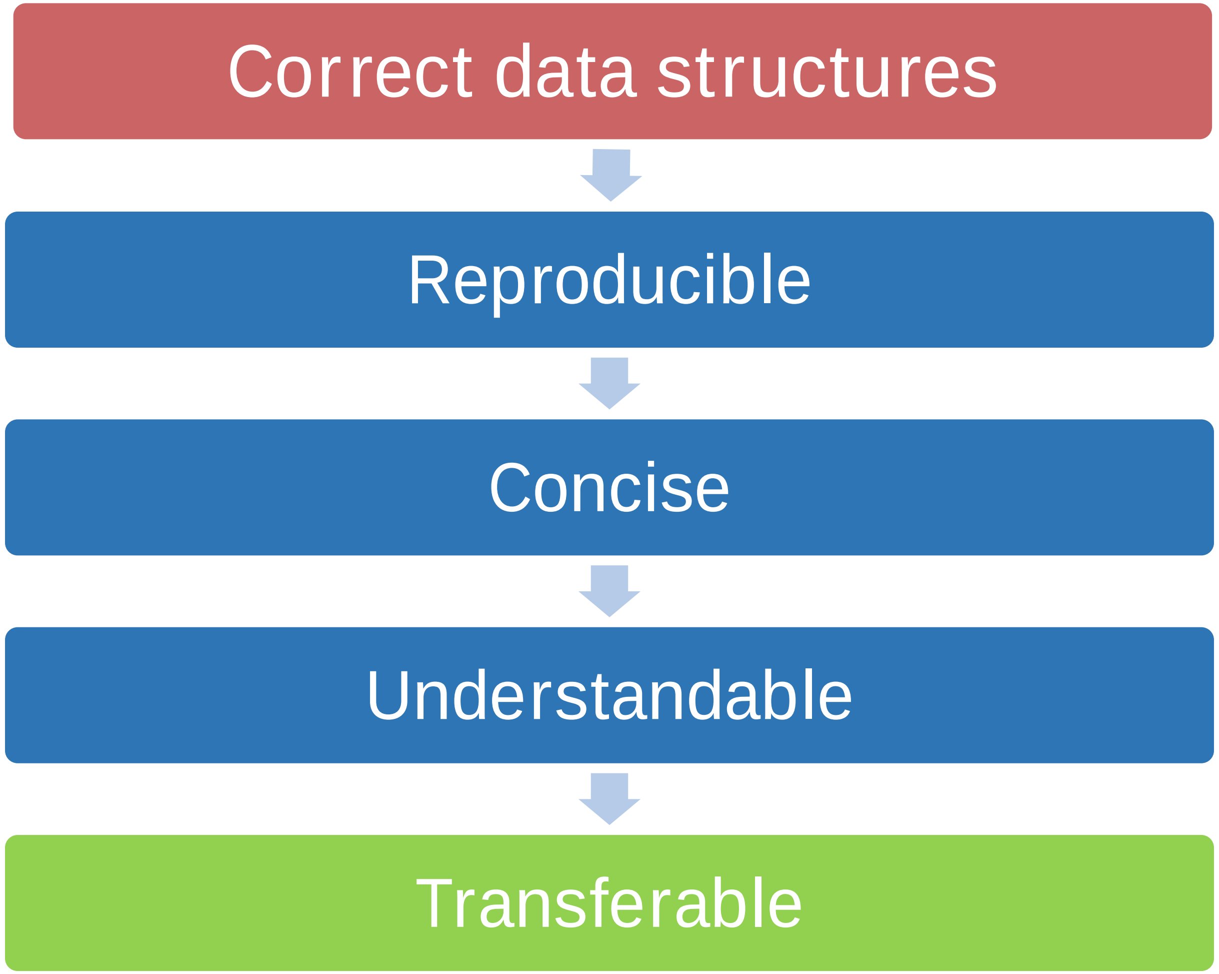
Programme – Day 2

ENDANGERED
WILDLIFE TRUST

08h30 – 10h00	Session 1 – Introduction to R, RStudio, basics of programming
10h00 – 10h30	Tea break
10h30 – 12h15	Session 2 – Data wrangling with the tidyverse
12h15 – 13h30	Lunch
13h30 – 15h00	Session 3 – Data visualisation using ggplot2
15h00 – 15h30	Tea break
15h30 – 17h00	Session 4 – Handling spatial data in R

Effective data workflow

ENDANGERED WILDLIFE TRUST



Correct data structures

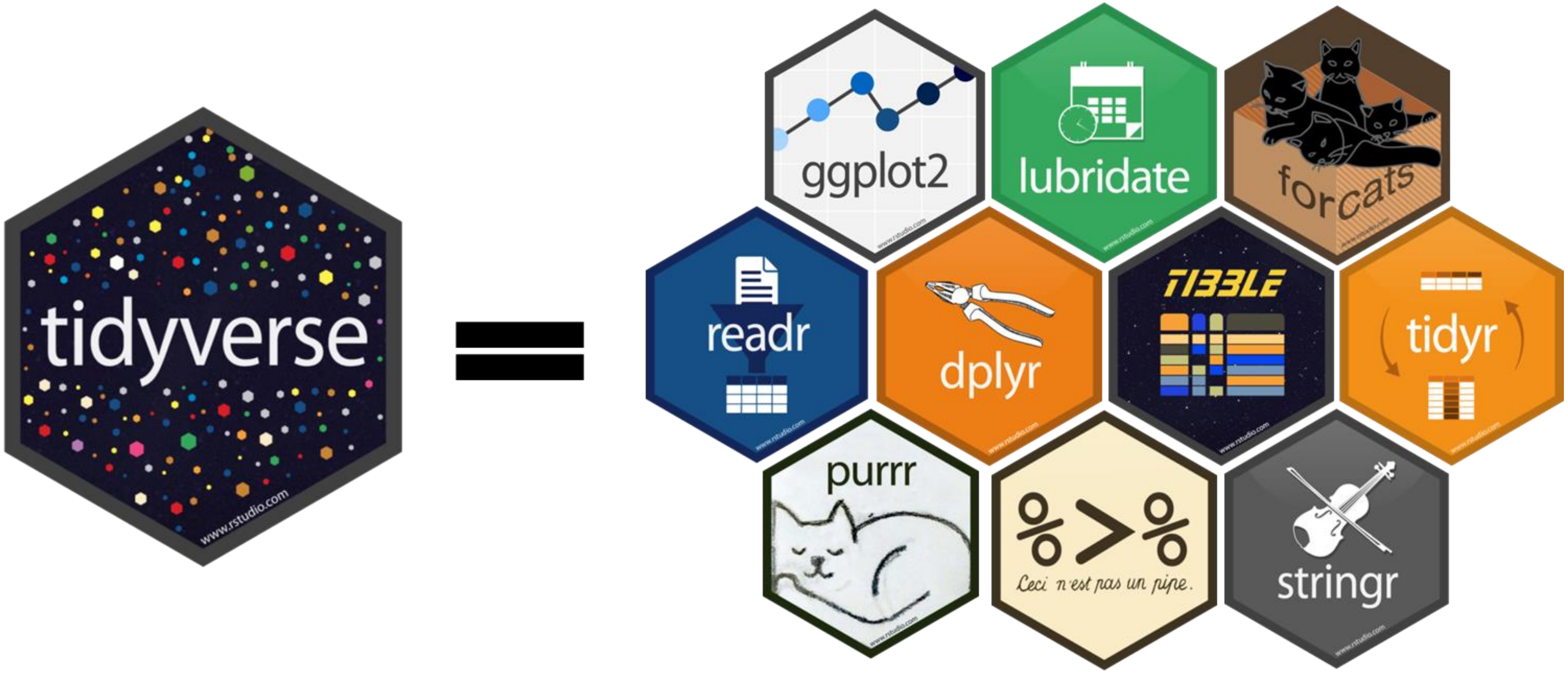
Reproducible

Concise

Understandable

Transferable

What is the tidyverse?

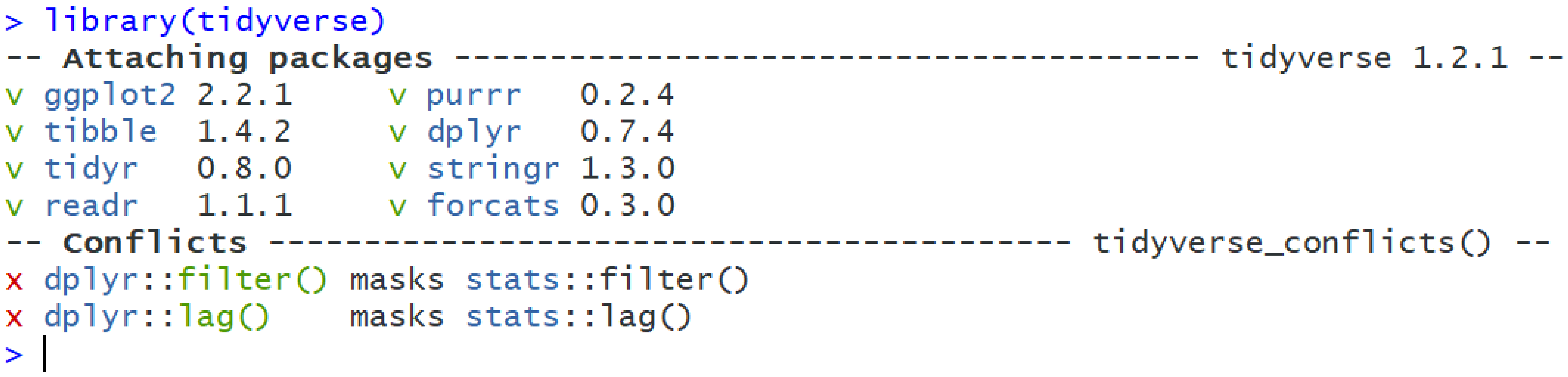


Tidyverse

- Design philosophy
- Grammar
- Data structures & representations

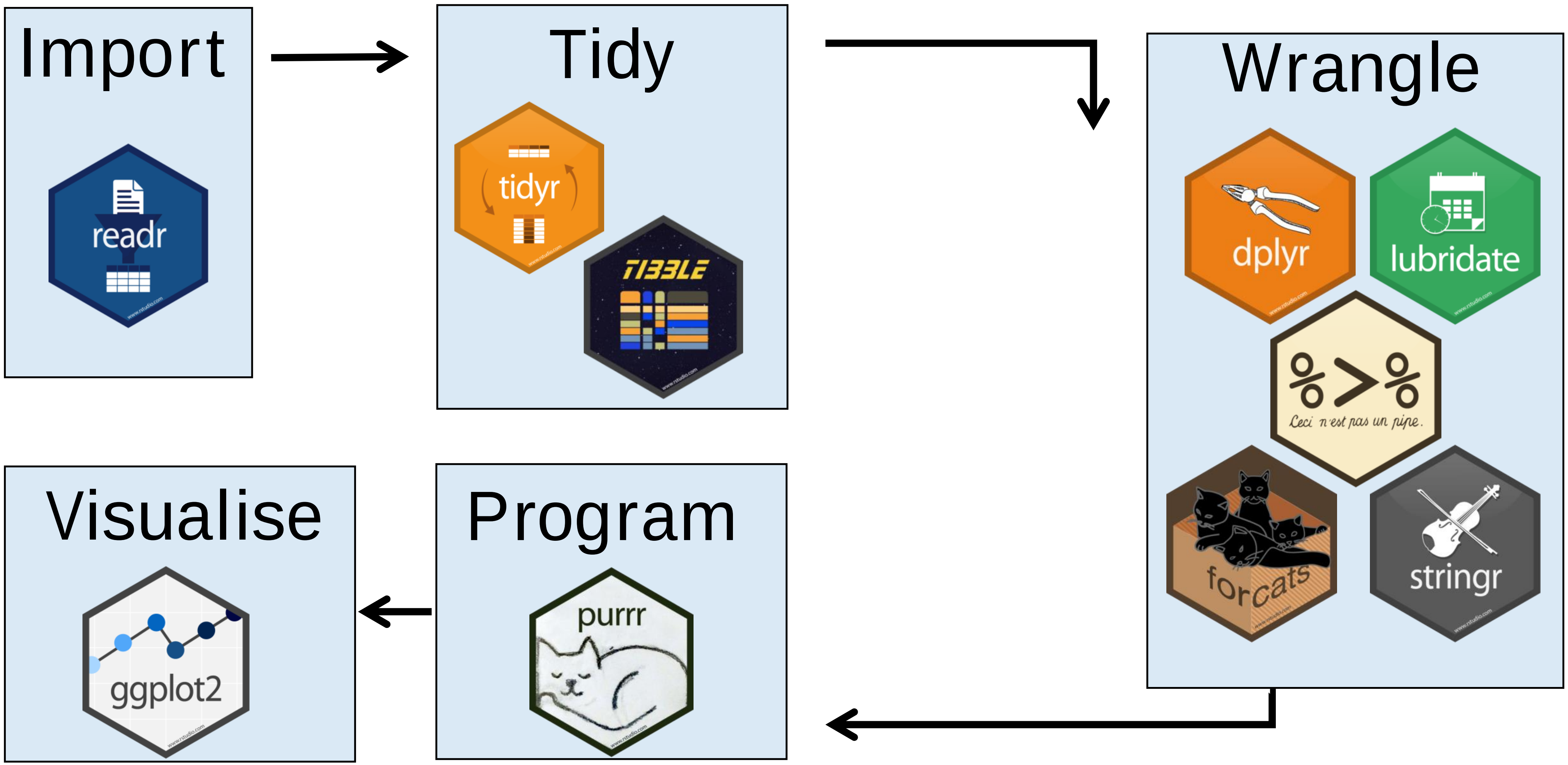
Hadley Wickham

Install and load



```
> library(tidyverse)
-- Attaching packages --------------------------------------- tidyverse 1.2.1 --
v ggplot2 2.2.1     v purrr   0.2.4
v tibble  1.4.2     v dplyr   0.7.4
v tidyr   0.8.0     v stringr 1.3.0
v readr   1.1.1     v forcats 0.3.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
> |
```

Workflow



Import

Tidy

Wrangle

Program

Visualise

Import

Behaviour:

Discards row names

Retains non-conventional column names

Ability to detect dates and times

Factors be damned! (characters remain characters)

Pros:

Fast (progress bar)

Sneak peek into column types

Creates a tibble object

`read_csv()`

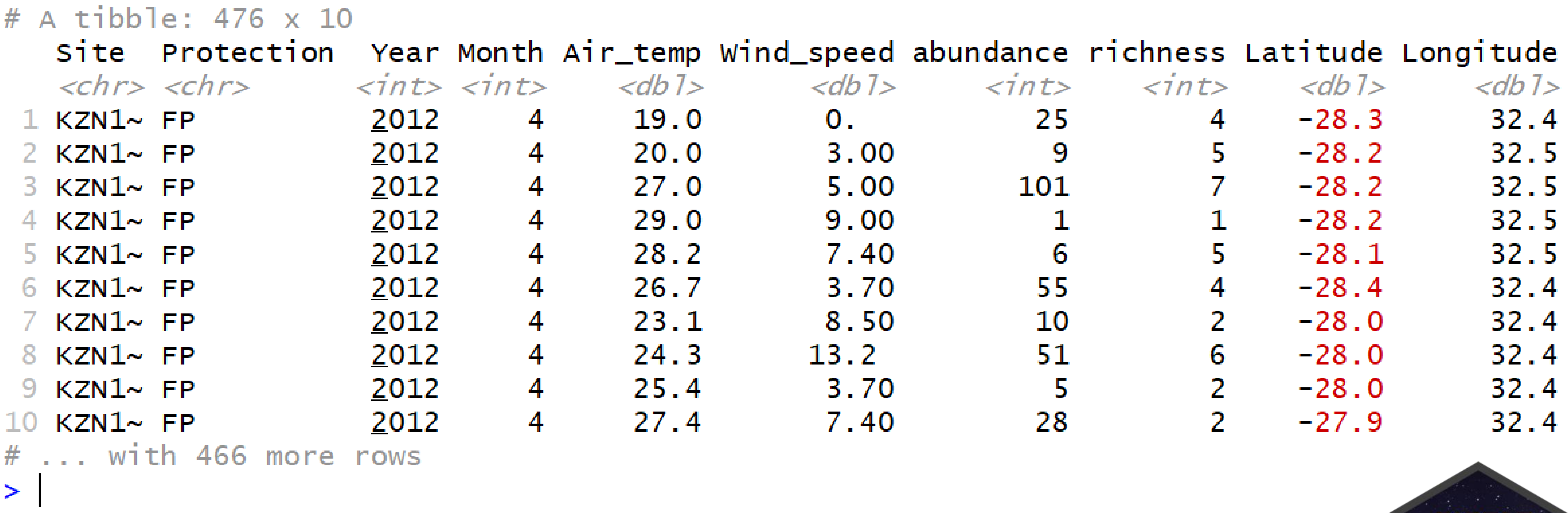
Tibble

- Data frame with modern features

- Improvements over data.frame objects:
  - Aesthetics and ease of reading
  - Column type information
  - Fit to console
  - Store lists as columns!

`tbl_df`

Tibble



```
# A tibble: 476 x 10
   Site   Protection   Year Month Air_temp Wind_speed abundance richness Latitude Longitude
   <chr>  <chr>       <int> <int>    <dbl>      <dbl>     <int>    <int>    <dbl>     <dbl>
 1 KZN1~ FP           2012     4     19.0       0.          25        4    -28.3      32.4
 2 KZN1~ FP           2012     4     20.0       3.00         9        5    -28.2      32.5
 3 KZN1~ FP           2012     4     27.0       5.00       101        7    -28.2      32.5
 4 KZN1~ FP           2012     4     29.0       9.00         1        1    -28.2      32.5
 5 KZN1~ FP           2012     4     28.2       7.40         6        5    -28.1      32.5
 6 KZN1~ FP           2012     4     26.7       3.70        55        4    -28.4      32.4
 7 KZN1~ FP           2012     4     23.1       8.50        10        2    -28.0      32.4
 8 KZN1~ FP           2012     4     24.3      13.2         51        6    -28.0      32.4
 9 KZN1~ FP           2012     4     25.4       3.70         5        2    -28.0      32.4
10 KZN1~ FP           2012     4     27.4       7.40        28        2    -27.9      32.4
# ... with 466 more rows
> |
```

Data structure (tidy data)

Species	Jan	Feb	Mar
Pied Kingfisher	5	2	7
African Jacana	20	0	23
Cape Teal	0	9	55

Species	Month	Count
Pied Kingfisher	Jan	5
Pied Kingfisher	Feb	2
Pied Kingfisher	March	7
African Jacana	Jan	20
African Jacana	Feb	0
African Jacana	Mar	23
Cape Teal	Jan	0
Cape Teal	Feb	9
Cape Teal	Mar	55

```
pivot_longer()
pivot_wider()
```

Wrangle

Typical tasks

- Explore structure
- Validate observations
- Create variables
- Select observations
- Summarise data
- Prepare input for models & visualisations

Wrangle

dplyr
www.rstudio.com
lubridate
www.rstudio.com
%>%
Ceci n'est pas un pipe.
forcats
www.rstudio.com
stringr
www.rstudio.com

The pipe operator

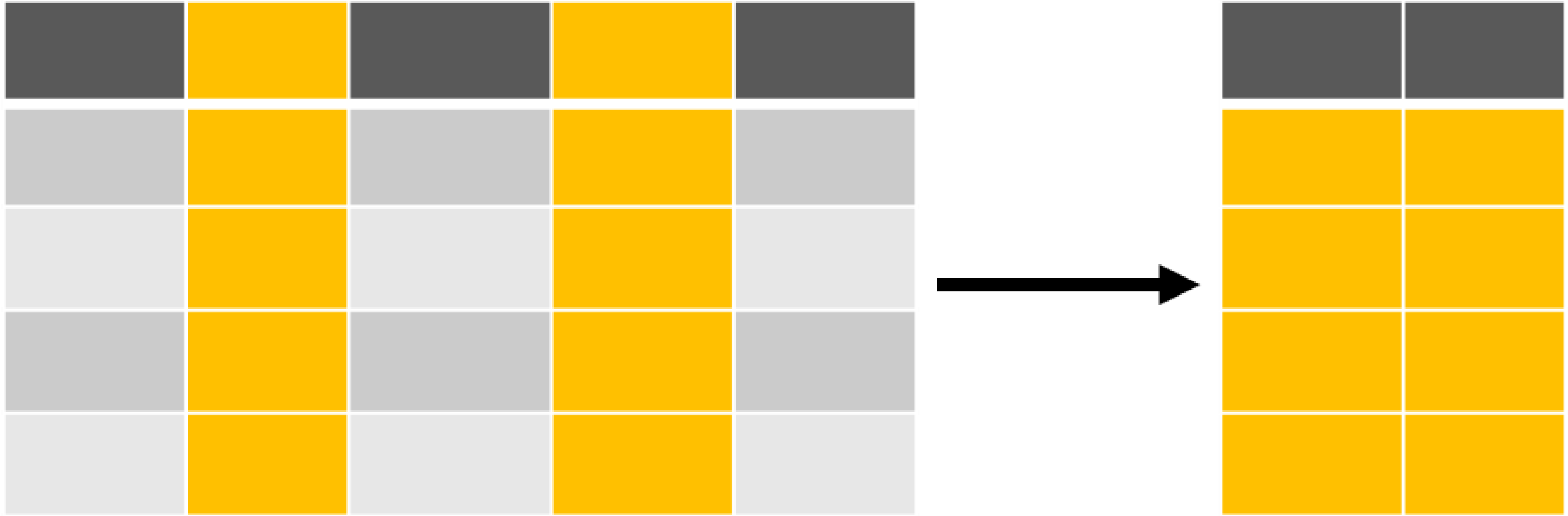
```
data %>% f1() %>% f2() %>% f3()
```

vs.

```
f3(f2(f1(data)))
```

dplyr verbs

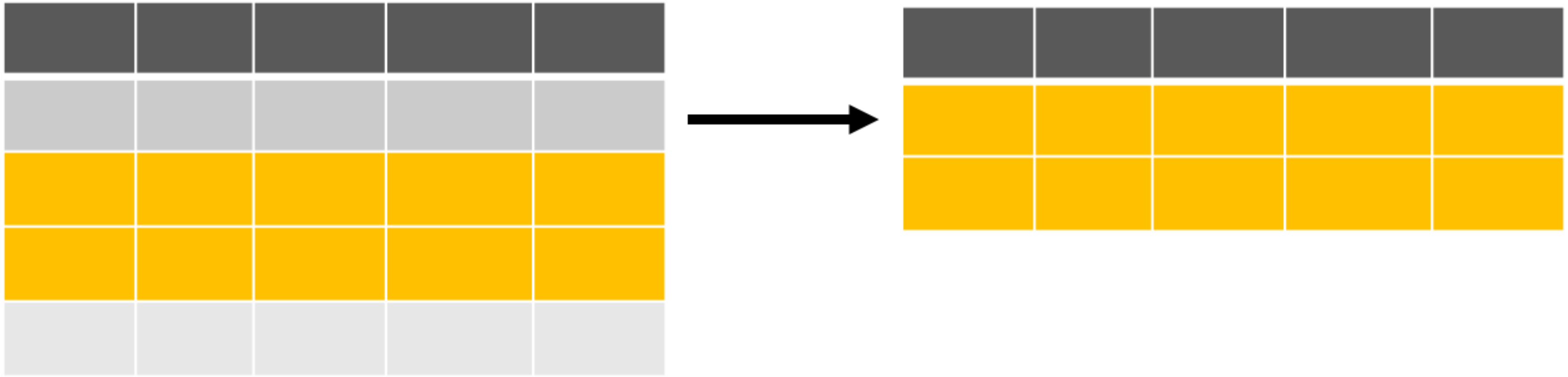
select():choose variables (cols) by name



dplyr verbs

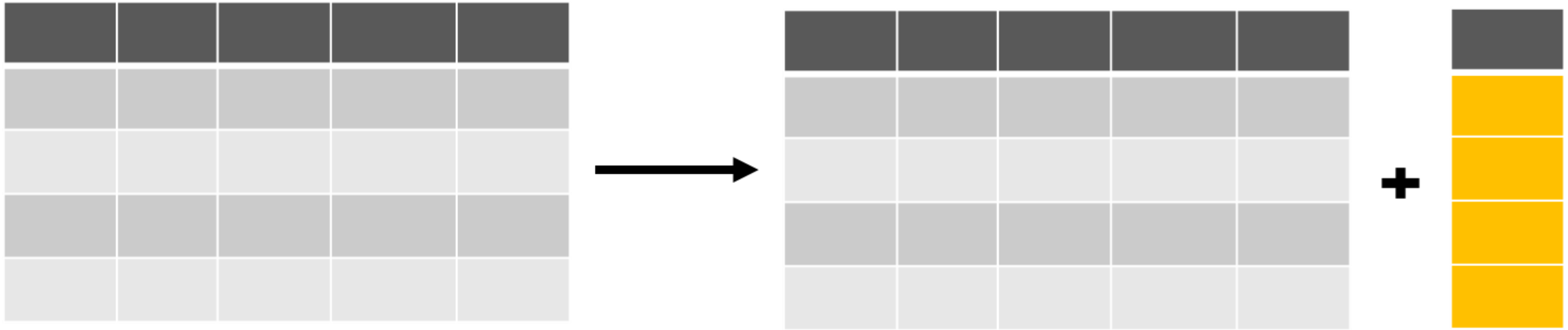
ENDANGERED
WILDLIFE TRUST

`filter()`: filter observations (rows) based on their value



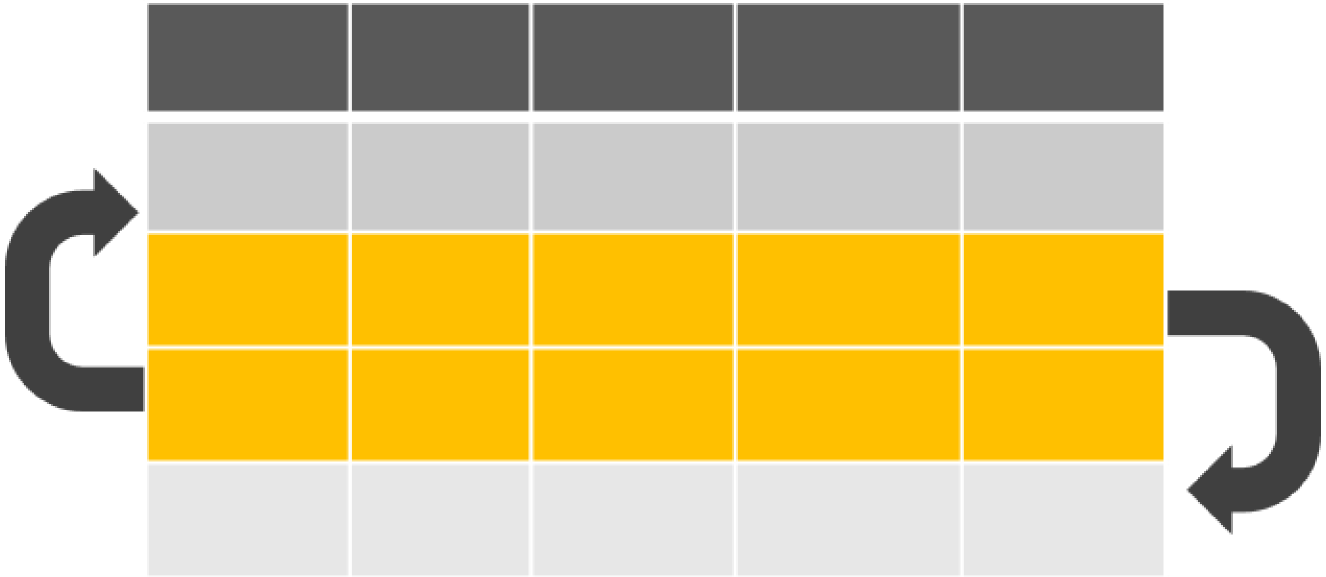
dplyr verbs

mutate():create new variables from existing ones



dplyr verbs

`arrange():change the order of observations`

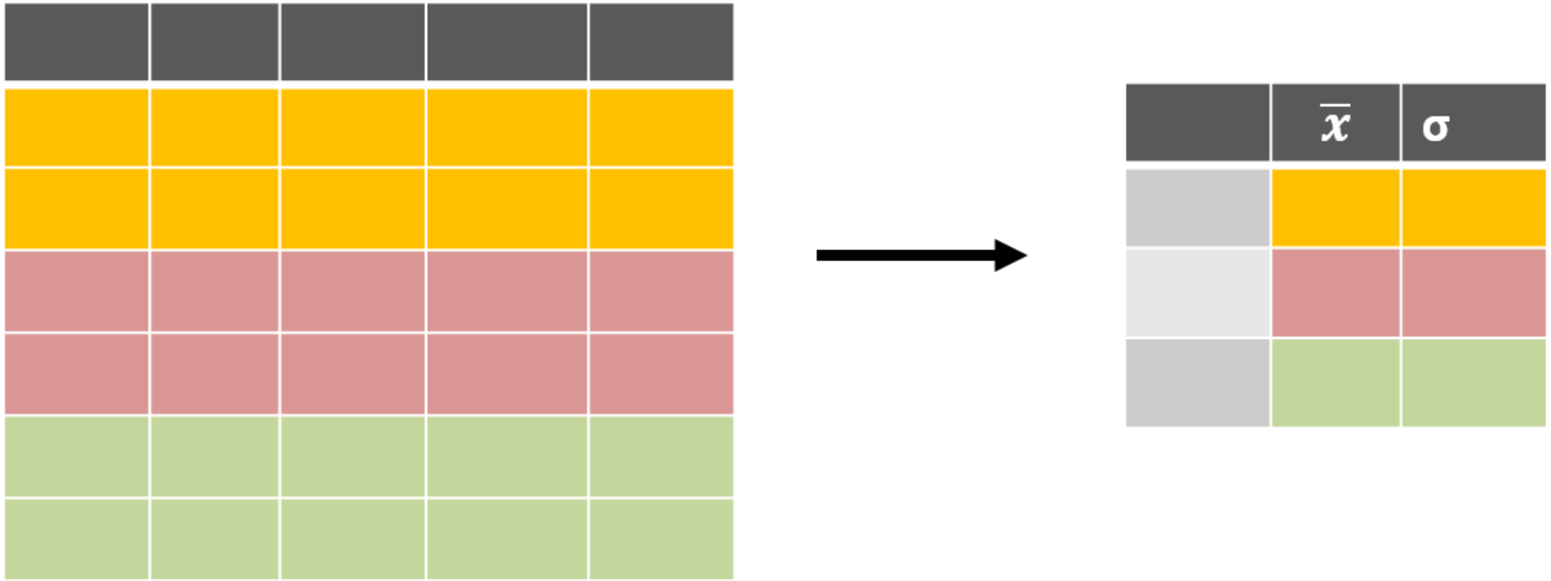


dplyr verbs

`group_by():select a factor by which to group observations`

dplyr verbs

summarise():reduce observations into a single value



dplyr + pipe

Option 1

```
data1 <- select(data, ...)
data2 <- filter(data1, ...)
data3 <- mutate(data2, ...)
```

Option 2

```
data %>% select(...) %>% filter(...) %>% mutate(...)
```

dplyr + pipe

Advantages

- Improved understanding - reads like a sentence
- Remove unnecessary intermediate steps
- Reduce creative effort (naming things sensibly is hard!)
- Focus on the final desired output

Dates and times

- Very often need to deal with dates and times
- Base R is confusing and frustrating
- lubridate makes things easy!

`dmy_hms()`

Base R

```
select(data, length)
```

```
data$length
data[[“length”]]
data[, 1]
```

Base R

```
mutate(data, length = length + 10)

data$length <- data$length + 10
```

Base R

```
filter(data, length > 10)
```

```
data[which(data$length > 10),]
```

Data wrangling

Link to Rmd file