

```
> workshop(  
  topic      = "R Introduction for Data Science",  
  trainer    = "Muhammad Aswan Syahputra",  
  when       = "2019-05-20",  
  where      = "Telkom University, Bandung"  
)  
> ...
```

- Sensory Scientist @ [Sensolution.ID](#)
- Using R for 4+ years, keen on Data Carpentry
- Initiator of [Komunitas R Indonesia](#)
- Pkgs: sensehubr, nusandata, bandungjuara, prakiraan, etc
- Shinyapps: sensehub, thermostats, aquastats, bcrp, bandungjuara, etc

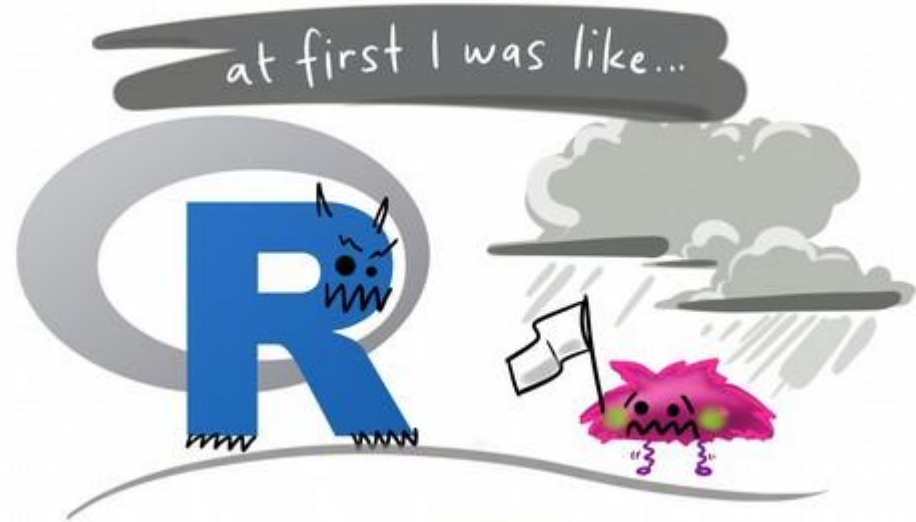


aswansyahputra

# Know your neighbour!

- Who are you?
- What you do with data?
- How would you describe your experience with R?

# Our goal



...but now it's like...



Artwork by @allison\_horst

**HELLO**

My name is

**ANDIKA**

**HELLO**

My name is

**ALYA**

**HELLO**

My name is

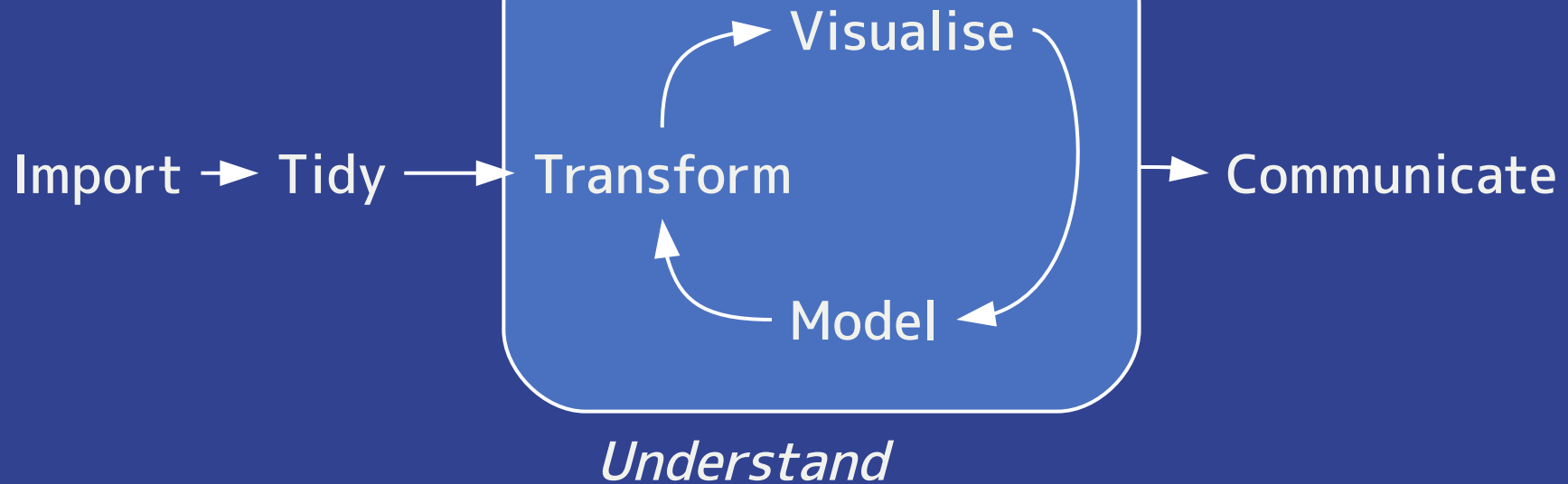
**WILLIAM**

**HELLO**

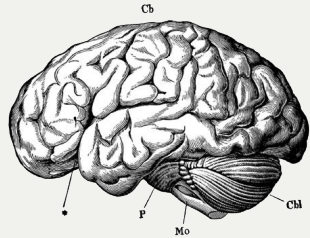
My name is

**ISHLA**

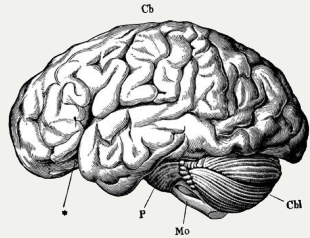




*Program*



Human thought ■—————▶ Machine Language

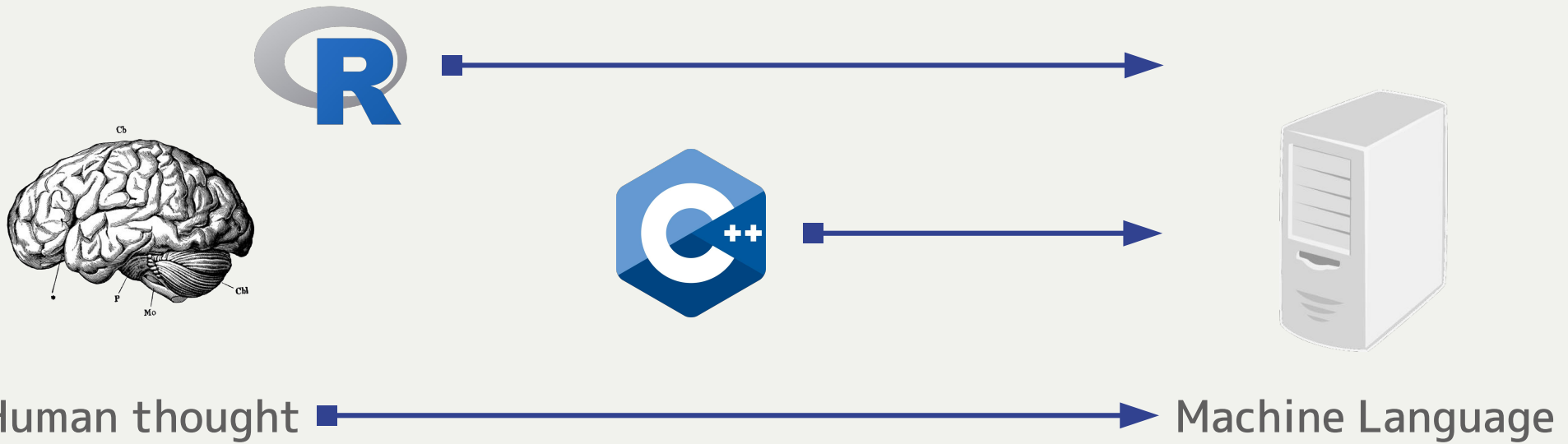


Human thought

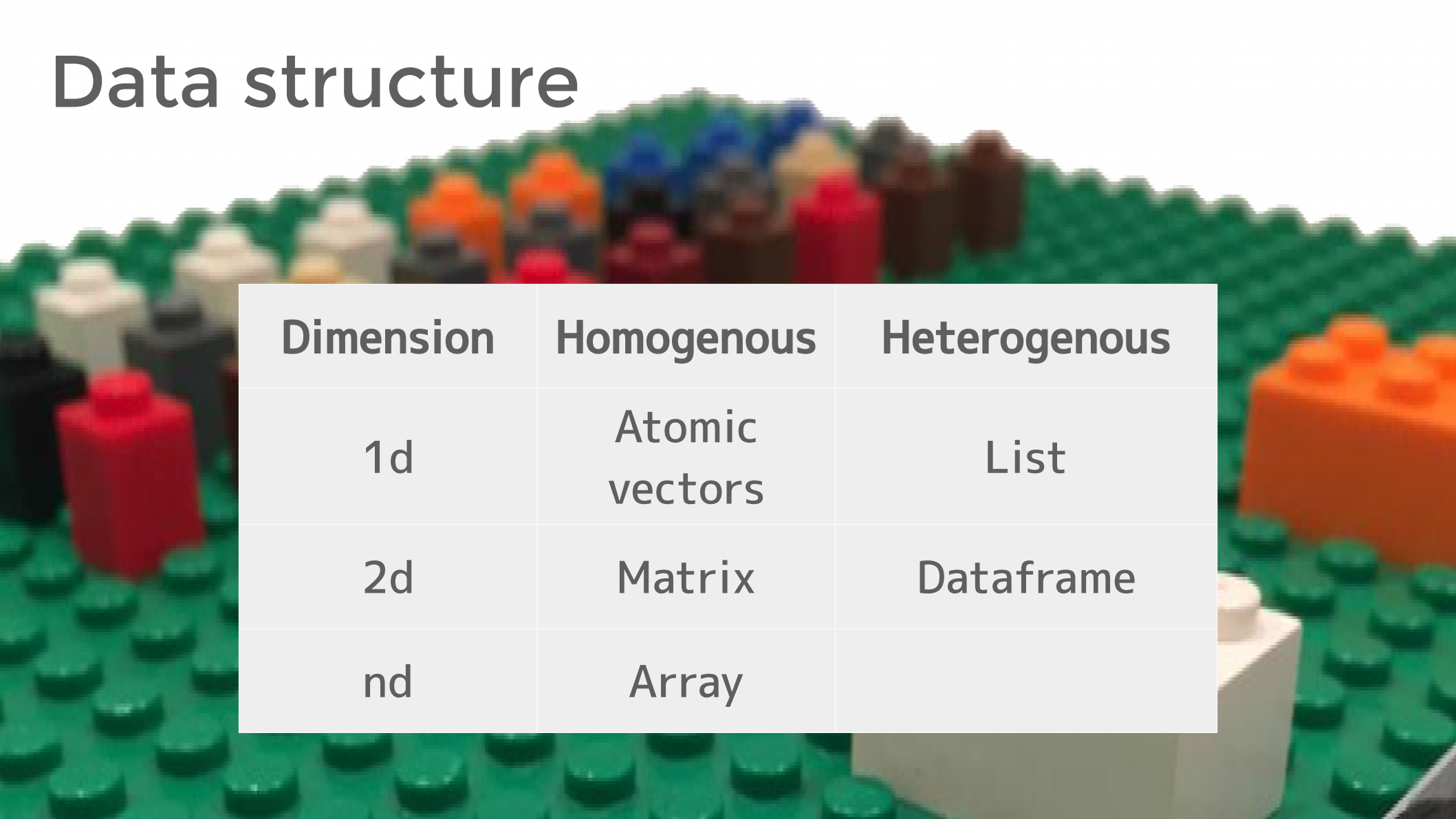


Machine Language





# Data structure



Dimension	Homogenous	Heterogenous
1d	Atomic vectors	List
2d	Matrix	Dataframe
nd	Array	

# Data structure

Dimension	Homogenous	Heterogenous
1d	Atomic vectors	List
2d	Matrix	Dataframe
nd	Array	



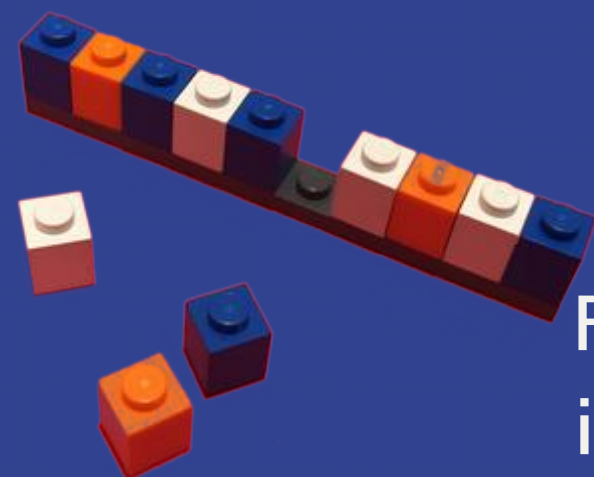
Integer, Double, Character

This image shows three long LEGO bricks representing data types. The top brick is brown with a red segment at the end. The middle brick is green with a yellow segment at the end. The bottom brick is blue with a red segment at the end. There are also several small individual bricks of these colors scattered around.



Logical

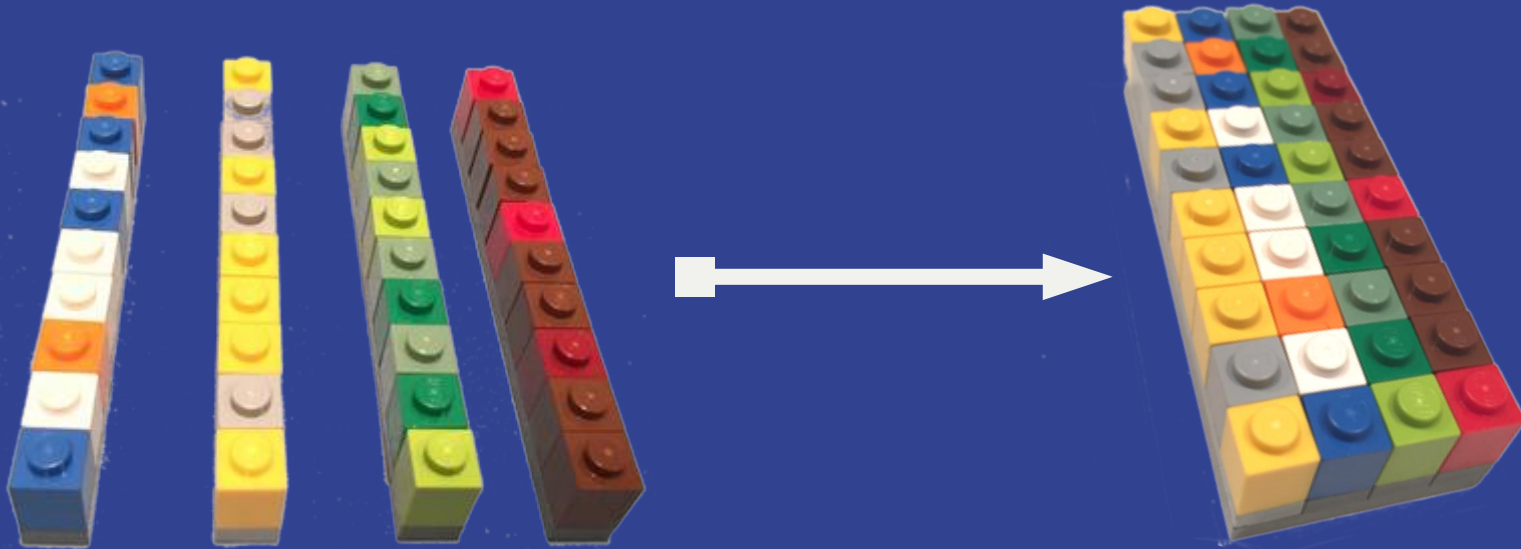
This image shows a long LEGO brick representing the Logical data type. It is yellow with a grey segment in the middle. There are also two small individual bricks, one grey and one yellow, scattered around.



Factor (basically  
integer with class)

This image shows a long LEGO brick representing the Factor data type. It is blue with a red segment in the middle. There are also several small individual bricks of these colors scattered around.



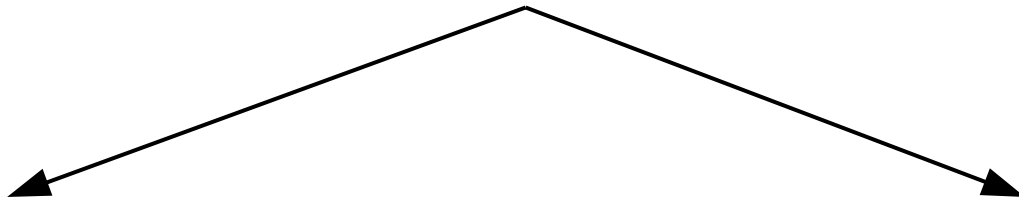


Vectors of same length

Dataframe

How do we process  
the data?

`function(arg1, arg2, arg3, ...)`

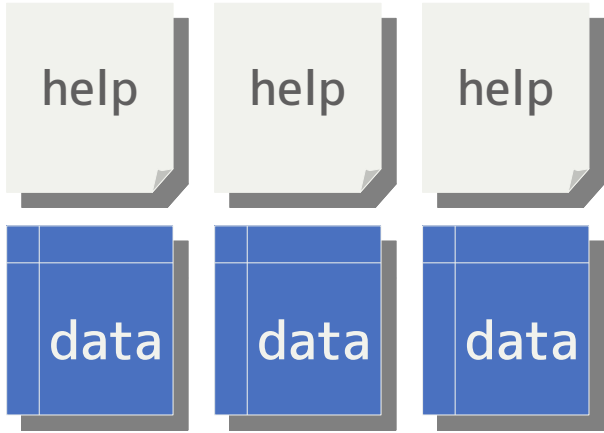


`change_the_env(...)`

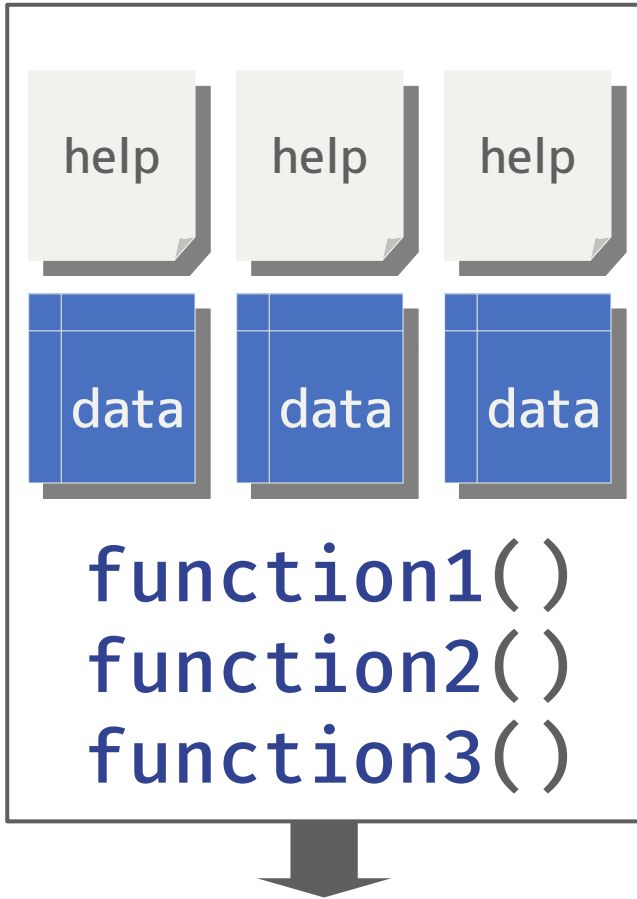
`calculate_value(...)`

assign. `<-`, `=`, `->`

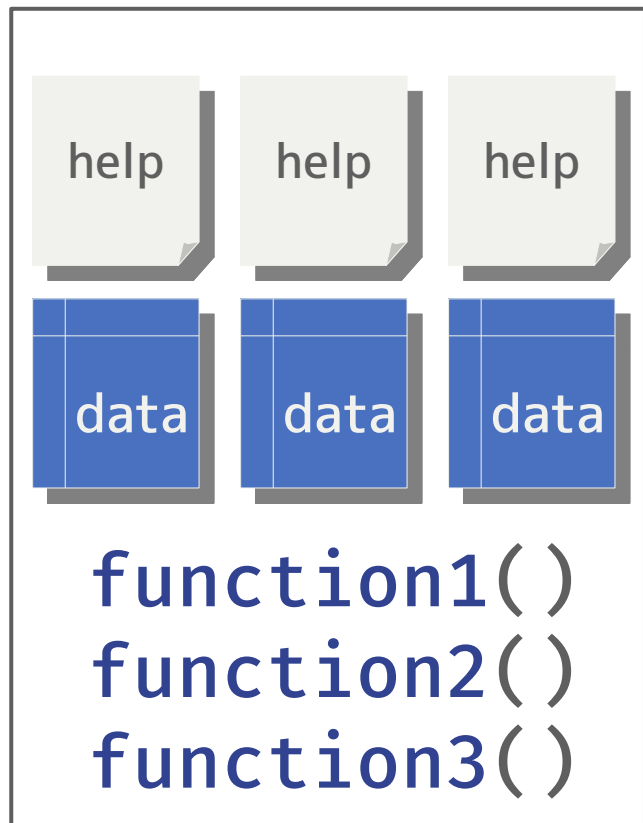
- arguments are contexts of a function
- arguments are matched by name, or
- arguments are matched by position, **be careful!**



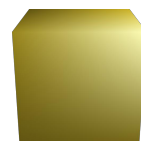
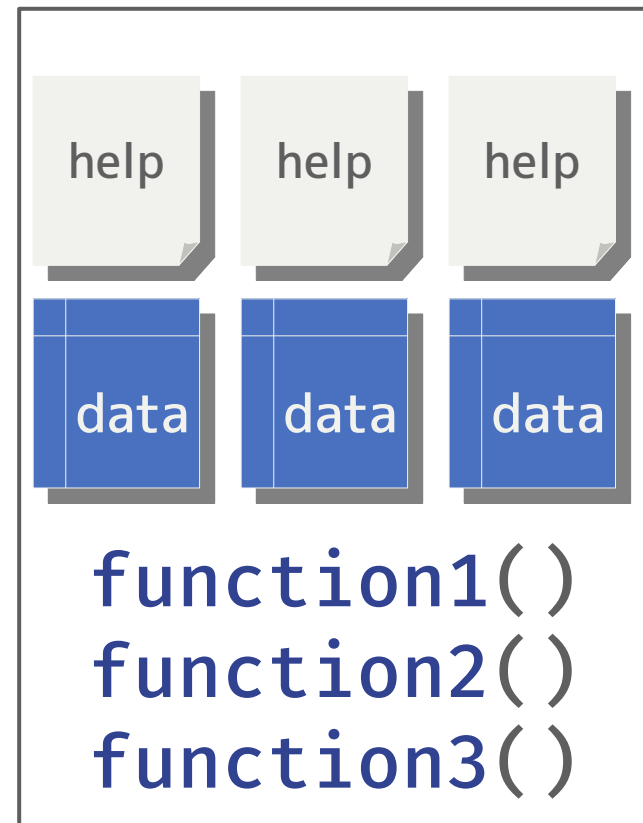
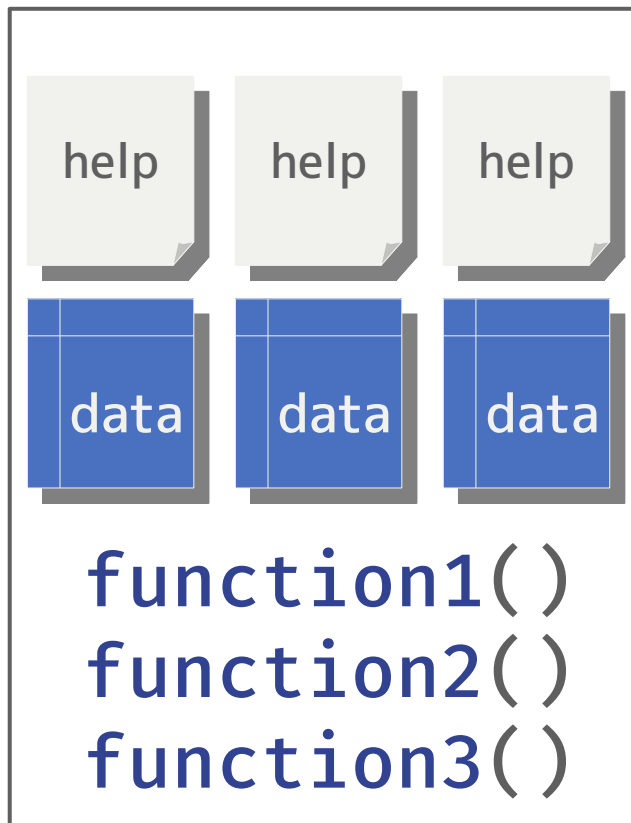
```
function1()  
function2()  
function3()
```



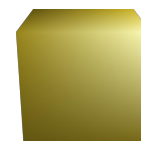
stats, graphics, grDevices, utils,  
datasets, methods, base



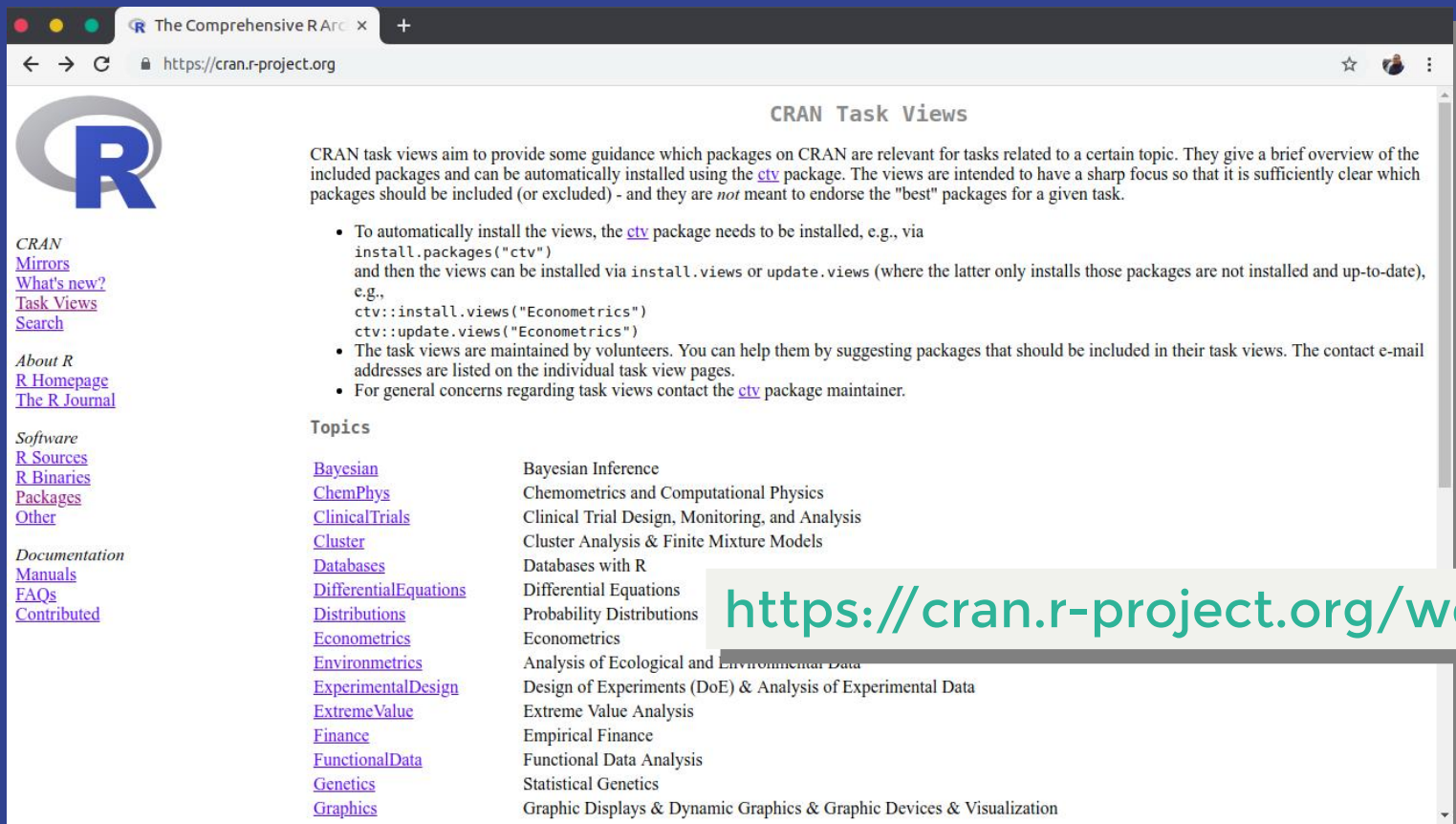
stats, graphics, grDevices, utils,  
datasets, methods, base



R packages



# CRAN Task Views



The screenshot shows a web browser window with the address bar displaying `https://cran.r-project.org`. The page title is "CRAN Task Views". On the left side, there is a navigation menu with links: "CRAN", "Mirrors", "What's new?", "Task Views", "Search", "About R", "R Homepage", "The R Journal", "Software", "R Sources", "R Binaries", "Packages", "Other", "Documentation", "Manuals", "FAQs", and "Contributed". The main content area features the R logo and a paragraph explaining that CRAN task views provide guidance on relevant packages for specific tasks, aiming for a sharp focus. Below this, a bulleted list provides instructions on how to install and update task views using the `ctv` package, and mentions that the views are maintained by volunteers. A "Topics" section follows, listing various statistical and computational topics such as Bayesian Inference, Chemometrics, Clinical Trial Design, Cluster Analysis, Databases with R, Differential Equations, Probability Distributions, Econometrics, Analysis of Ecological and Environmental Data, Design of Experiments (DoE), Extreme Value Analysis, Empirical Finance, Functional Data Analysis, Statistical Genetics, and Graphic Displays. A large green URL `https://cran.r-project.org/web/views/` is overlaid on the bottom right of the page.

## CRAN Task Views

CRAN task views aim to provide some guidance which packages on CRAN are relevant for tasks related to a certain topic. They give a brief overview of the included packages and can be automatically installed using the [ctv](#) package. The views are intended to have a sharp focus so that it is sufficiently clear which packages should be included (or excluded) - and they are *not* meant to endorse the "best" packages for a given task.

- To automatically install the views, the [ctv](#) package needs to be installed, e.g., via `install.packages("ctv")` and then the views can be installed via `install.views` or `update.views` (where the latter only installs those packages are not installed and up-to-date), e.g.,  
`ctv::install.views("Econometrics")`  
`ctv::update.views("Econometrics")`
- The task views are maintained by volunteers. You can help them by suggesting packages that should be included in their task views. The contact e-mail addresses are listed on the individual task view pages.
- For general concerns regarding task views contact the [ctv](#) package maintainer.

### Topics

<a href="#">Bayesian</a>	Bayesian Inference
<a href="#">ChemPhys</a>	Chemometrics and Computational Physics
<a href="#">ClinicalTrials</a>	Clinical Trial Design, Monitoring, and Analysis
<a href="#">Cluster</a>	Cluster Analysis & Finite Mixture Models
<a href="#">Databases</a>	Databases with R
<a href="#">DifferentialEquations</a>	Differential Equations
<a href="#">Distributions</a>	Probability Distributions
<a href="#">Econometrics</a>	Econometrics
<a href="#">Environmetrics</a>	Analysis of Ecological and Environmental Data
<a href="#">ExperimentalDesign</a>	Design of Experiments (DoE) & Analysis of Experimental Data
<a href="#">ExtremeValue</a>	Extreme Value Analysis
<a href="#">Finance</a>	Empirical Finance
<a href="#">FunctionalData</a>	Functional Data Analysis
<a href="#">Genetics</a>	Statistical Genetics
<a href="#">Graphics</a>	Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization

<https://cran.r-project.org/web/views/>

# GitHub

The screenshot shows the GitHub search interface with the query 'r'. The left sidebar contains navigation links for Repositories (5M+), Code (?), Commits (54M+), Issues (2M), Marketplace (9), Topics (1), Wikis (232K), and Users (33K). Below this is a 'Languages' section with a list of programming languages and their repository counts. The main content area displays the search results for 'r', showing the R programming language repository at the top. Below the R repository, it indicates 'Showing 5,189,217 available repository results' and lists two repositories: 'dmpe/R' and 'shifteight/R'. The 'dmpe/R' repository is highlighted with a blue dot and shows 330 stars. The 'shifteight/R' repository is shown below it with 50 stars. The URL 'https://github.com/search?q=r' is overlaid on the right side of the image.

Search · r · GitHub

GitHub, Inc. [US] | <https://github.com/search?q=r>

Why GitHub? Enterprise Explore Marketplace Pricing

r Sign in Sign up

**Repositories** 5M+

- Code ?
- Commits 54M+
- Issues 2M
- Marketplace 9
- Topics 1
- Wikis 232K
- Users 33K

**Languages**

JavaScript	967,420
Java	468,100
Ruby	346,761
HTML	333,048
Python	306,423
C#	135,565
CSS	131,039
PHP	100,750

**R** Star

R is a free programming language and software environment for statistical computing and graphics.

[See topic](#)

Showing 5,189,217 available repository results ? Sort: Best match ▾

**dmpe/R** R 330

Exercises (incl. analyses) with R language (math+statistics)

r data-analysis exercise statistics course

MIT license Updated on Oct 28, 2018

**shifteight/R** HTML 50

R exercises and examples

Updated on Dec 8, 2018

<https://github.com/search?q=r>



## Installing package, only once

```
install.packages("pkg") # from CRAN/local  
remotes::install_github("user/pkg") # from GitHub  
remotes::install_bioc("repo") # from Bioconductor
```

## Loading package, once in every session

```
library(pkg)  
pacman::p_load(pkg) # load or install if not  
available
```

A lot of R packages  
to use! :D

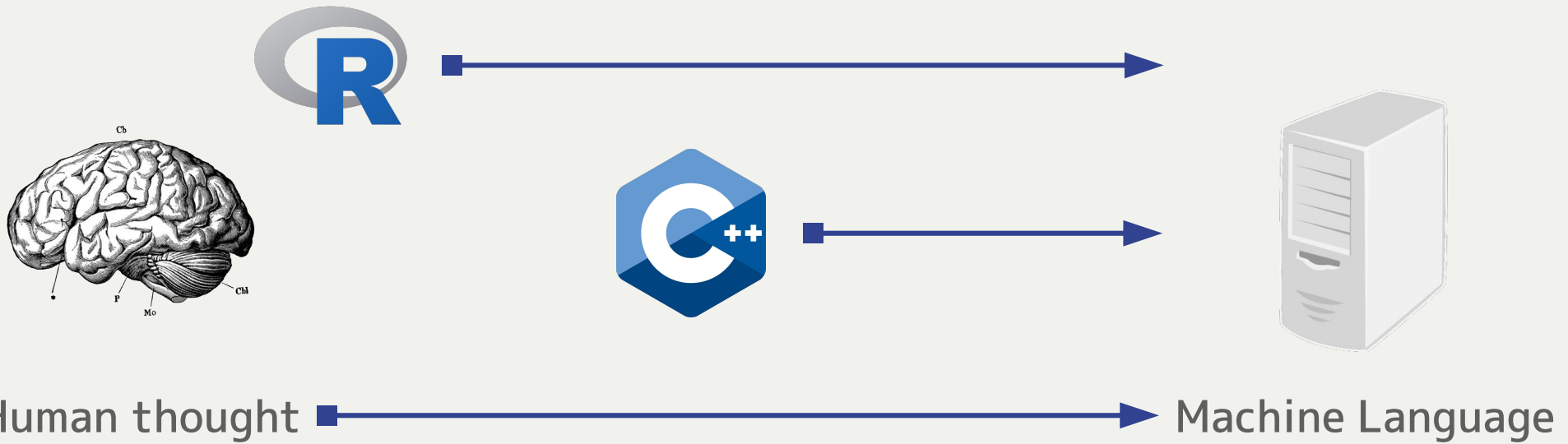
A lot of R packages  
to use! :(

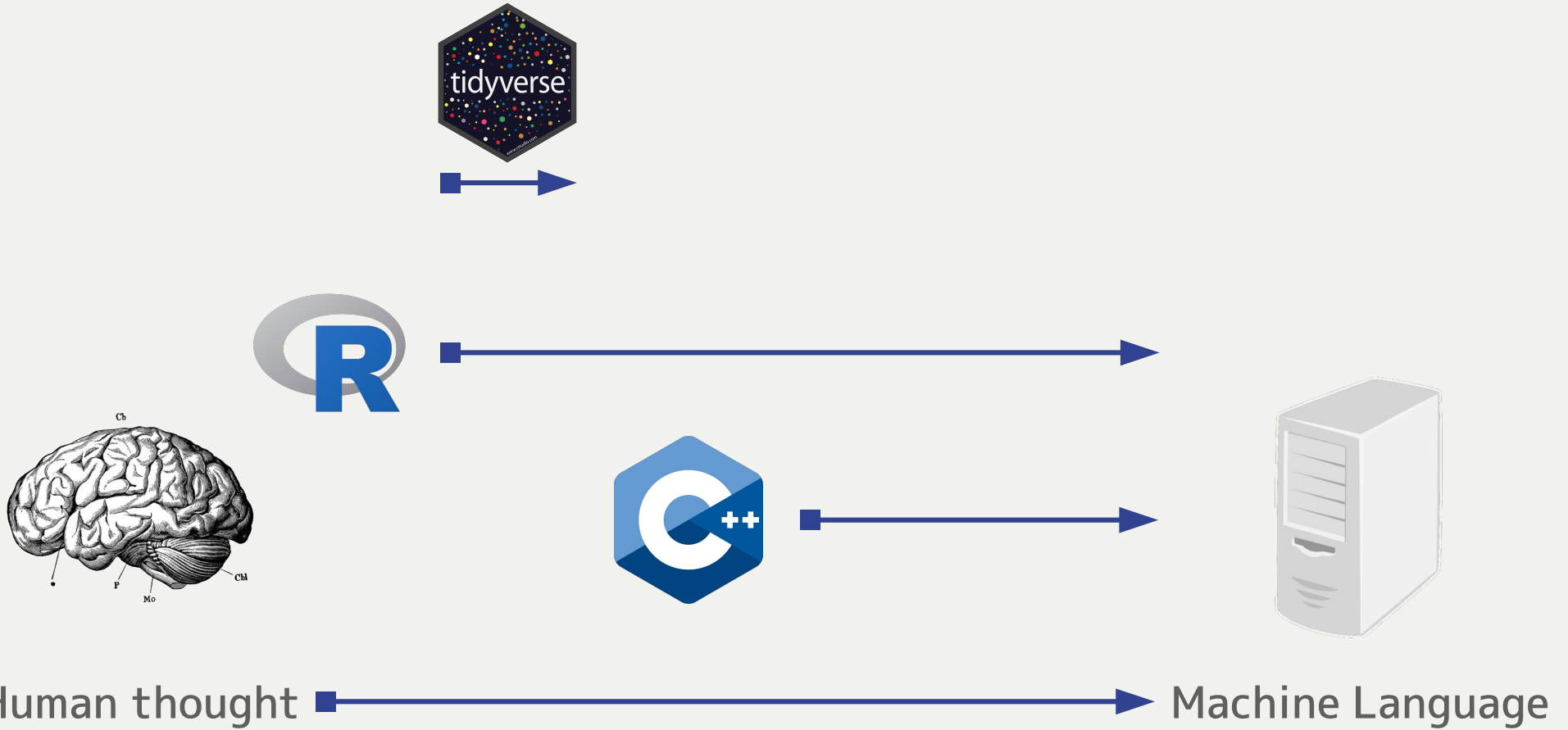




The tidyverse is an opinionated **collection of R packages** designed for **data science**. All packages **share** an underlying design philosophy, grammar, and data structures.

Artwork by @allison\_horst





# R Syntax Comparison

## Dollar sign

```
goal(data$x, data$y)
```

- A.k.a base syntax
- Subsetting data by using '['

## Formula

```
goal(y~x, data=data)
```

Mostly used in  
modeling and  
statistical test

## Tidyverse

```
data %>% goal(x, y)
```

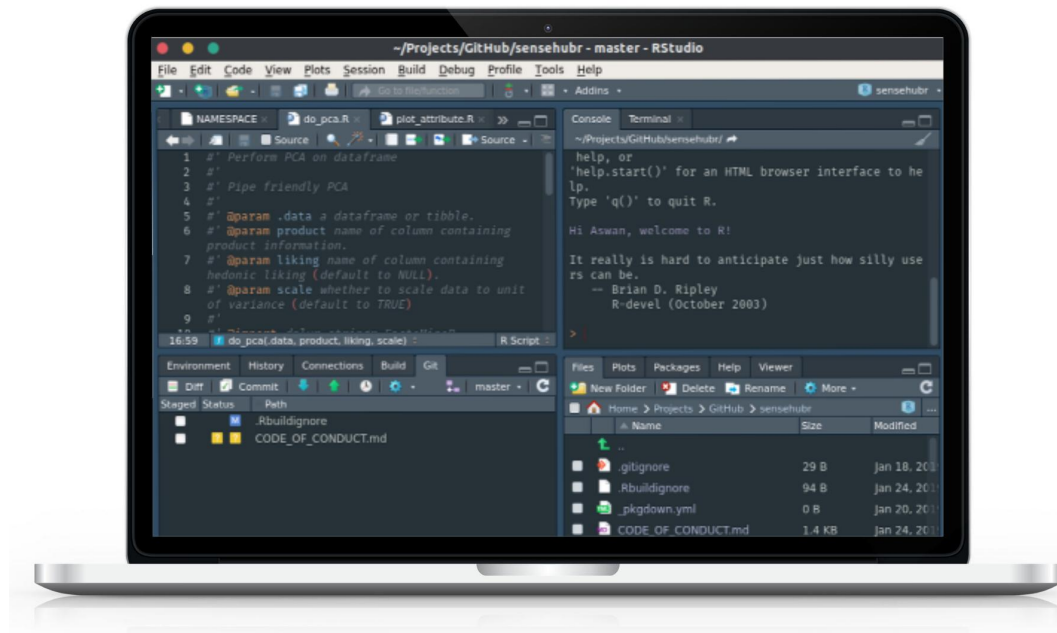
- Expecting data as the first argumen
- Plotting using '+' flavour



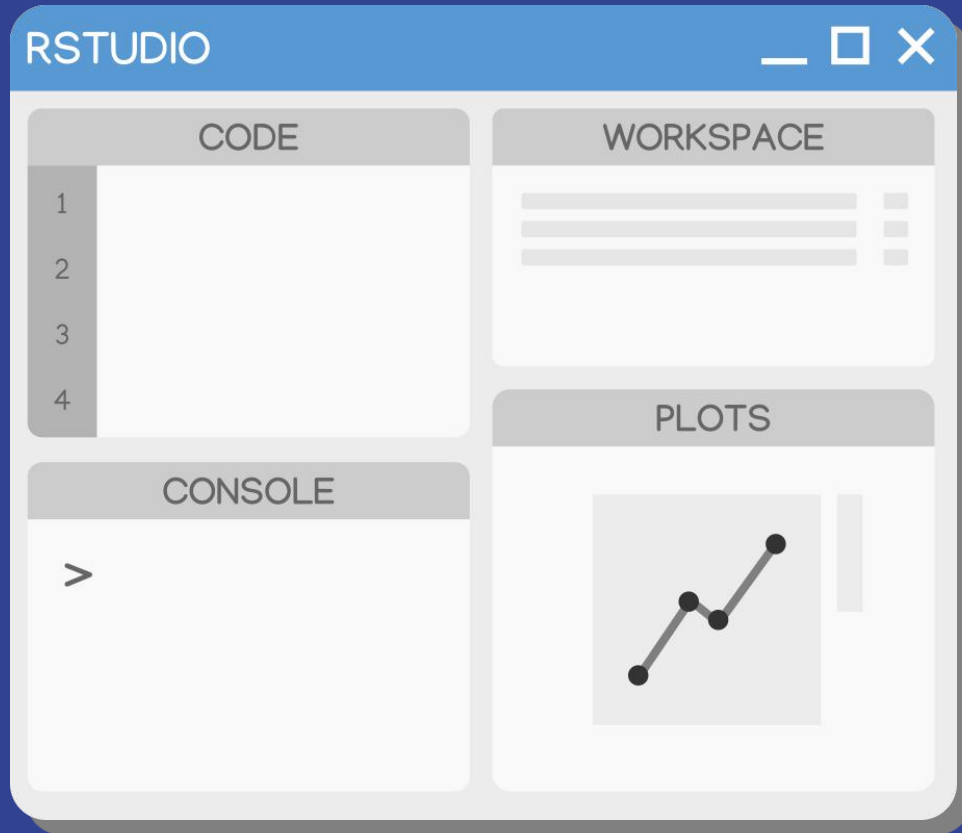
# R Studio®

## Main features:

- Console
- Syntax-highlighting editor
- Tools for plotting, history, debugging and workspace management



[rstudio.com/products/rstudio/download/](https://rstudio.com/products/rstudio/download/)

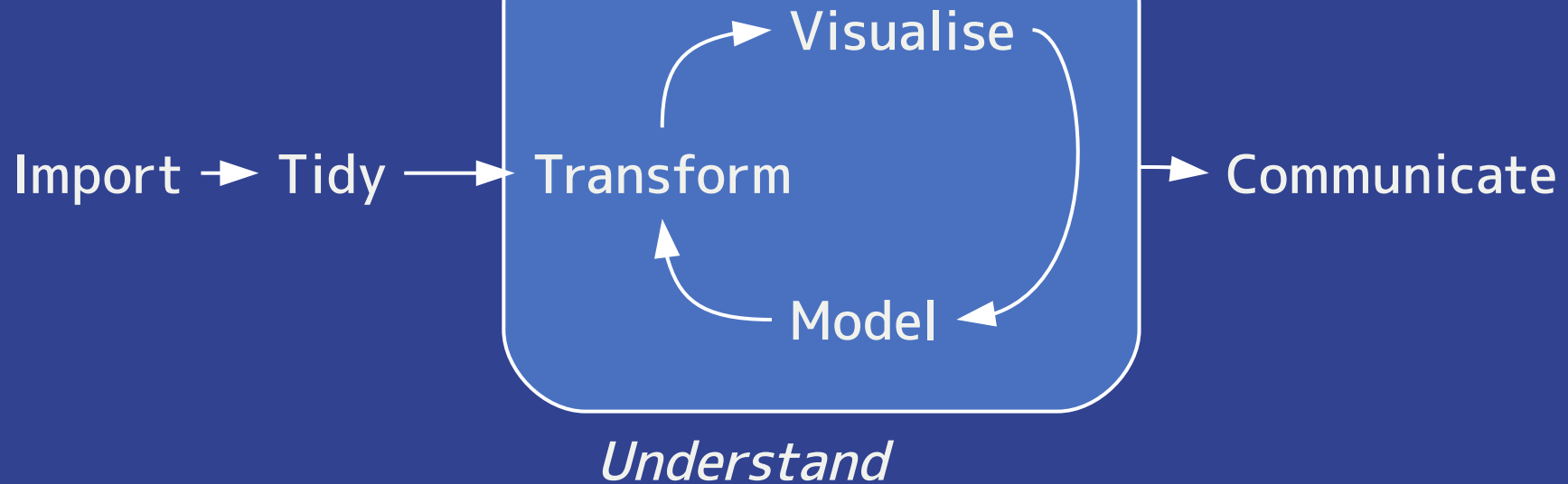


- **Tab**, autocompletion & path navigation
- **Alt + -**, for assignment operator <-
- **Ctrl + Shift + M**, for pipe operator %>%
- **Ctrl + Enter**, run current line  
code/example on help page
- **Ctrl + Up**, search for code history on  
console or editor pane
- **Alt + Up/Down**, move code to above or  
below
- **Alt + Shift + Up/Down**, copy code to  
above or below
- **Ctrl + D**, delete current line
- **Ctrl + Shift + F10**, restart R session
- **Ctrl + Alt + B**, run code up to current  
line

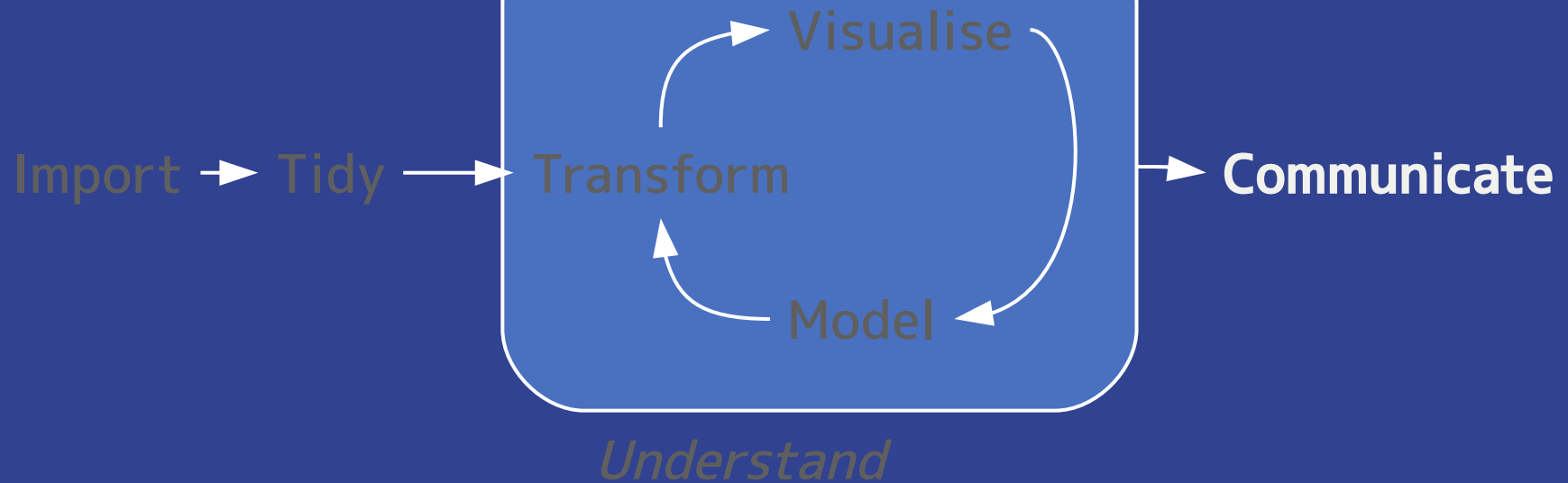
Cheatsheet: <https://github.com/rstudio/cheatsheets/raw/master/rstudio-ide.pdf>

# Let's get them!

- Download R at [cran.r-project.org](https://cran.r-project.org)
- Download RStudio at [rstudio.com](https://rstudio.com)
- Download R packages: **tidyverse**



*Program*



*Program*

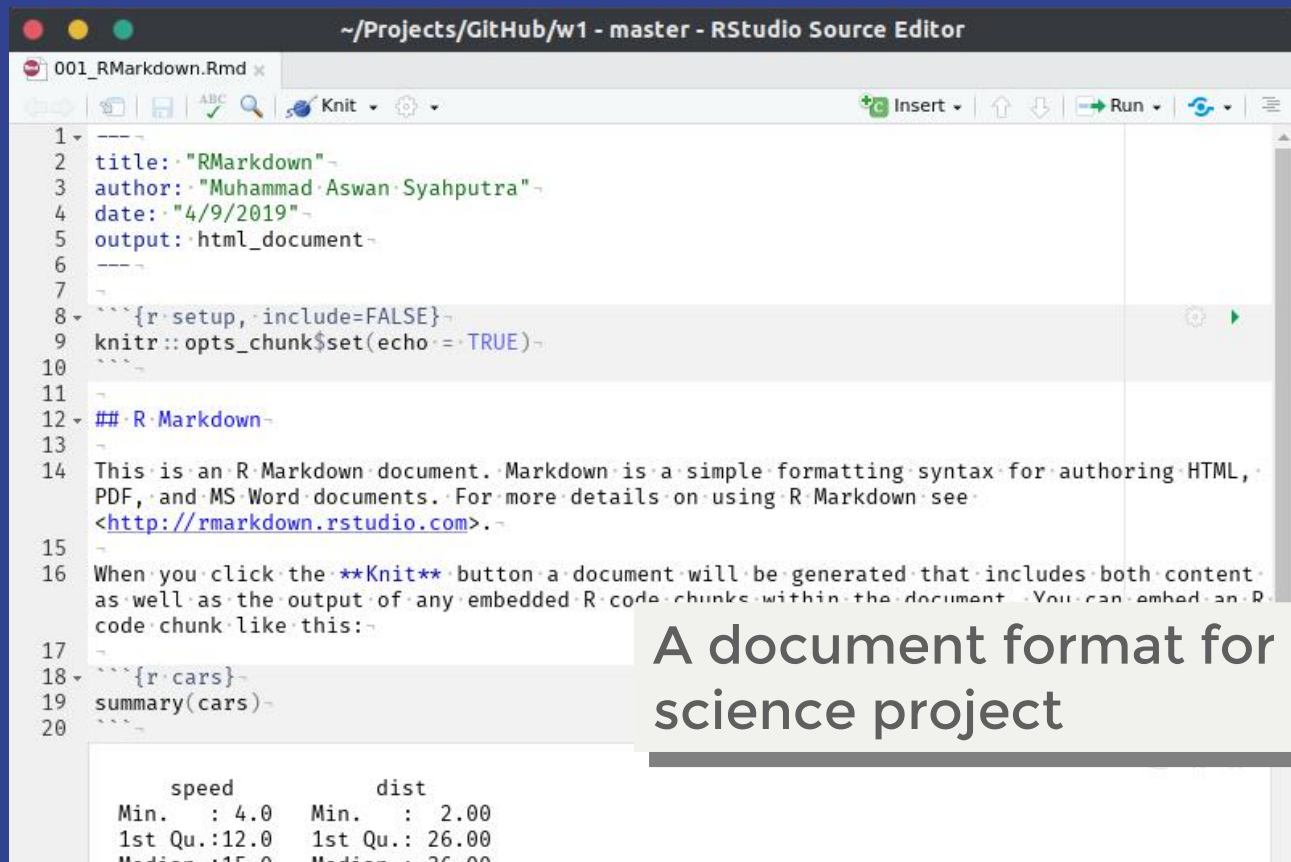
# Rmarkdown

TEXT. CODE. OUTPUT.  
(GET IT TOGETHER, PEOPLE.)



Artwork by @allison\_horst

# R Markdown



```
1 ---
2 title: "RMarkdown"
3 author: "Muhammad Aswan Syahputra"
4 date: "4/9/2019"
5 output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML,
15 PDF, and MS Word documents. For more details on using R Markdown see
16 <http://rmarkdown.rstudio.com>.
17
18 When you click the Knit button a document will be generated that includes both content
19 as well as the output of any embedded R code chunks within the document. You can embed an R
20 code chunk like this:
21
22 ```{r cars}
23 summary(cars)
24 ```
```

speed	dist
Min. : 4.0	Min. : 2.00
1st Qu.: 12.0	1st Qu.: 26.00
Median : 15.0	Median : 36.00
Mean : 15.28	Mean : 35.46
3rd Qu.: 19.0	3rd Qu.: 46.00
Max. : 25.0	Max. : 52.00

A document format for authoring data science project

- Use script (R Markdown or R Script), try to avoid console
- Use Projects, not `setwd( ... )` in script
- Set `stringsAsFactor = FALSE`, but not in the .Rprofile
- Ctrl+Shift+F10 and Ctrl+Alt+B to clean up, not `rm(list=ls())`
- Learn the handy shortcuts
- Do not save and load .Rdata
- Use version control system: git!

Reading: [happygitwithr.com](http://happygitwithr.com)



git

Download: [git-scm.com](http://git-scm.com)





With great codes,  
comes great bugs!  
- *(not) Uncle Ben*

Store and share! Why sharing your work? Motivation here.

- `git clone https://github/user/repo`
- Do some works!
- `git add file.R` or `git add .`
- `git commit -m "what you have done"`
- `git push origin master`
- Repeat: work, git add, git commit, git push

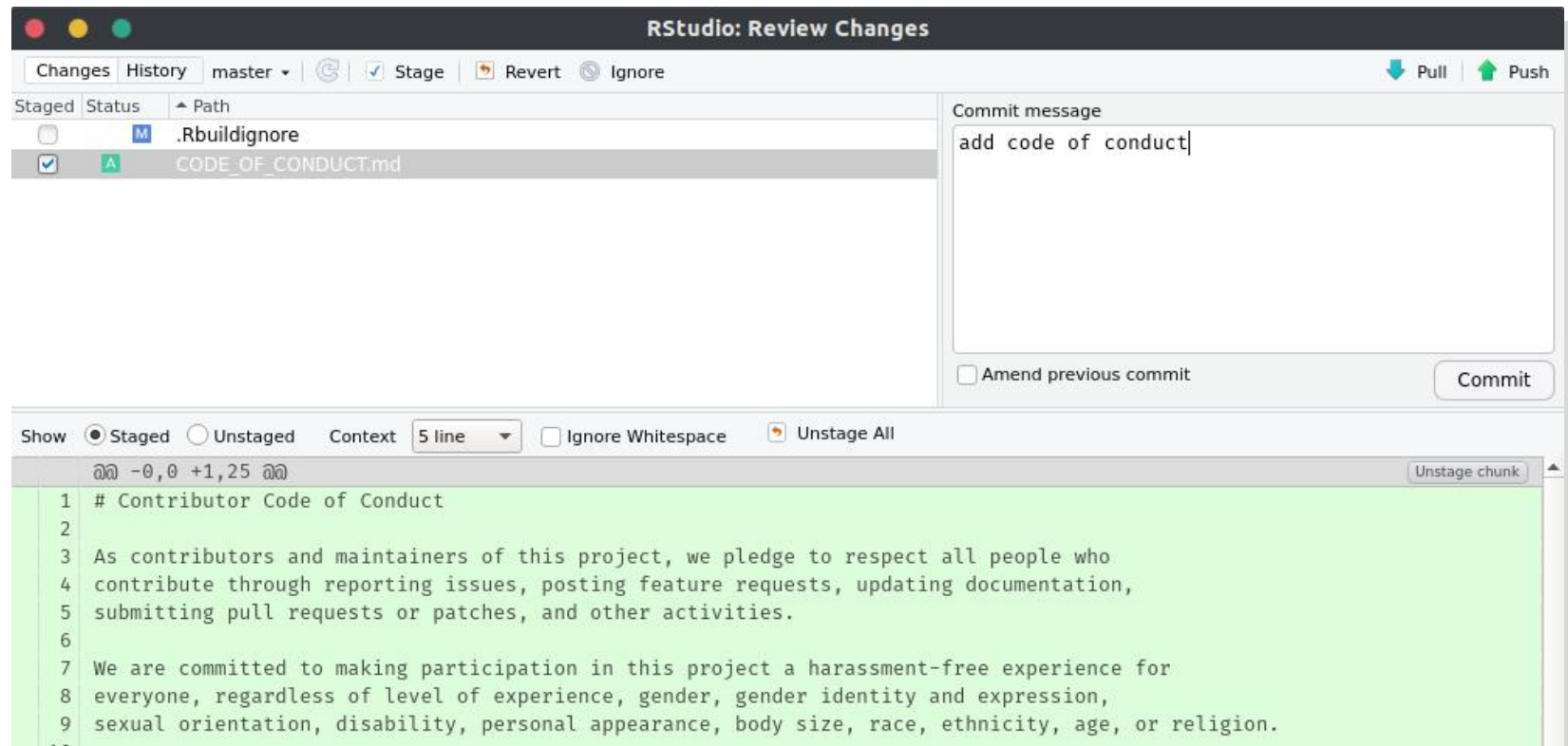


- `git init`
- `git remote add origin https://github/user/repo`
- Do some works!
- `git add file.R` or `git add .`
- `git commit -m "what you have done"`
- `git push -u origin master` #use -u only once
- Repeat: work, git add, git commit, git push



# git

# It is available in RStudio!



# How to git(Hub)?

- Create account at **github.com**
- Download and install git from **git-scm.com**
- In Rstudio, **Tools – Global Options**
  - **Git/SVN** – Browse **git executable:**
    - Windows:  
**C:/Program Files/Git/bin/git.exe**
    - UNIX/UNIX-like:  
**/usr/bin/git**

# How to set git?

- Open git bash/shell/terminal
- Run:

```
git config --global  
user.email  
"email@domain.com"
```

```
git config --global  
user.name "Your Name"
```

# Let's get started!

- Go to [github.com/r-academy/intro](https://github.com/r-academy/intro)
- Hit 'Fork' button
- Click 'Clone or Download', copy the URL
- In RStudio, File – New Project – Version Control – Git. Paste URL
- In File pane (bottom-right), click vignettes-'001\_pendahuluan.Rmd' to open it
- You have **10 minutes** to play with it!

# Working directory

```
> fs::dir_tree()
```

```
.
├── 003_kamisdata_Debat-Pilpres1-2019.Rproj
├── Dockerfile
├── R
│   ├── cari.R
│   └── impor.R
├── README.md
├── data
│   └── debat-pilpres1-2019.rda
├── data-raw
│   └── debat_pilpres1_2019.R
├── install.R
├── vignettes
│   ├── aswansyahputra-frekuensidansentimen.Rmd
│   ├── aswansyahputra-frekuensidansentimen.html
│   └── aswansyahputra-frekuensidansentimen_files
```

```
> |
```

## 3 principles for naming files:

- Machine readable
- Human readable
- Default ordering

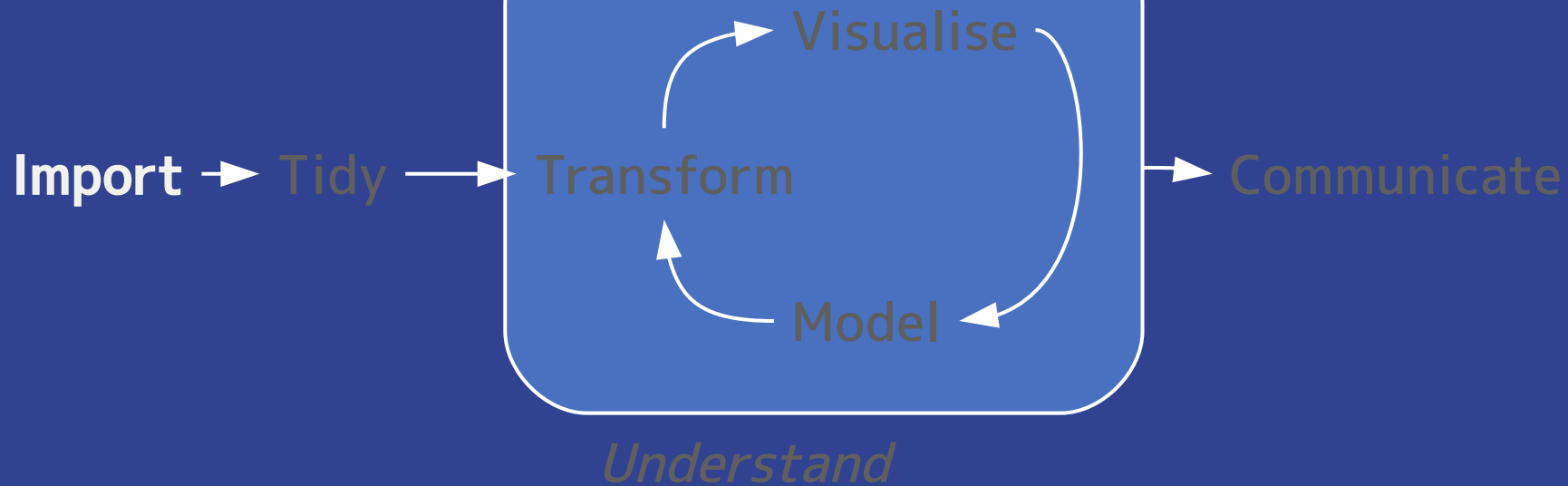
More info:

[speakerdeck.com/jennybc/how-to-name-files](https://speakerdeck.com/jennybc/how-to-name-files)

# How to use git?

- In **Environment Pane**, hit '**Git**' tab
- Click **commit**, a window pane will appear
- Select all files (Ctrl + A), click '**Stage**'
- Fill **commit message**, then click '**Commit**'
- Hit '**Push**' Button, done!
- You may check your GitHub now!





*Program*

Remote File

Local Files

API

Database

Clipboard

Remote File

The diagram consists of five rounded rectangular boxes arranged in a circular pattern. Each box has a central horizontal band of a different color, with the text centered on this band. The boxes are: 'Local Files' (yellow band), 'Remote File' (dark gray band), 'API' (dark gray band), 'Clipboard' (yellow band), and 'Database' (dark gray band). All boxes have a subtle drop shadow.

Local Files

API

Database

Clipboard



[www.rstudio.com](http://www.rstudio.com)



- **read\_csv()**: comma separated (CSV) files
- **read\_tsv()**: tab separated files
- **read\_delim()**: general delimited files
- **read\_fwf()**: fixed width files
- **read\_table()**: tabular files where columns are separated by white-space.
- **read\_log()**: web log files

# readr



**Brisbane area**

Partly cloudy. Light winds.

3:30 pm, UV Index predicted to reach 11 [Extreme]

**Brisbane area**












Partly cloudy. Medium (50%) chance of showers, most likely in the late morning and afternoon. Light winds becoming easterly 15 to 20 km/h in the late afternoon then becoming light in the evening.

3:30 pm, UV Index predicted to reach 11 [Extreme]

**Brisbane area**

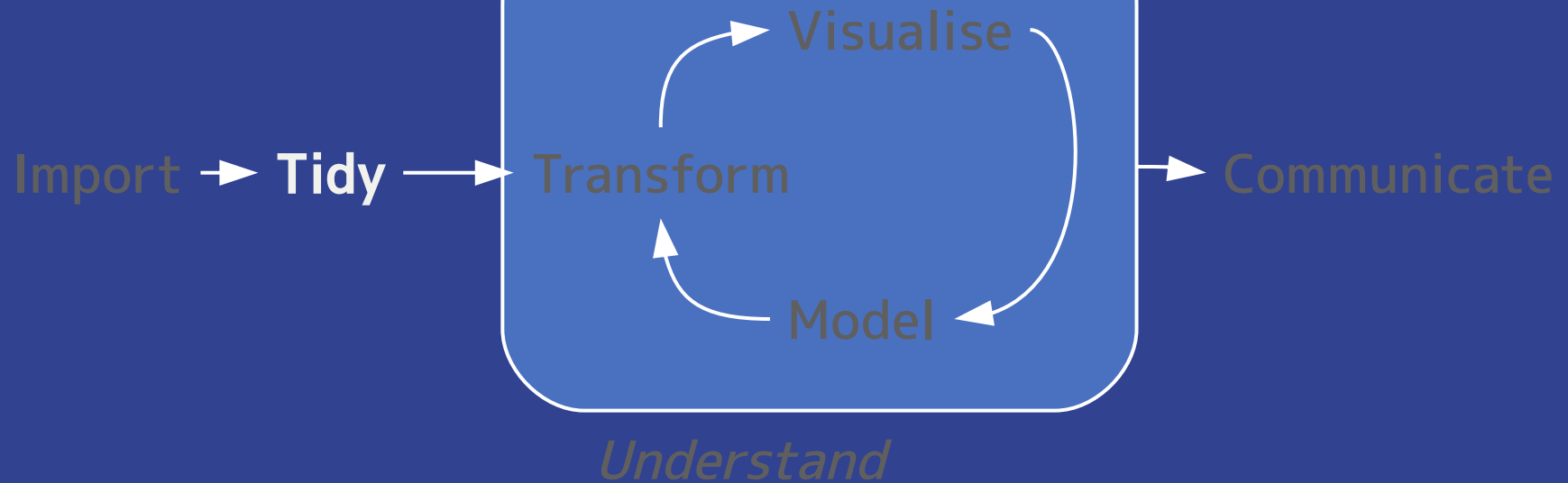
Partly cloudy. Light winds.

**7 day Town Forecasts**

	Location	Min	Max
	<a href="#">Brisbane</a>	23	30
	<a href="#">Brisbane Airport</a>	22	29
	<a href="#">Beaudesert</a>	21	30
	<a href="#">Chermside</a>	22	30
	<a href="#">Galton</a>	21	30
	<a href="#">Ipswich</a>	21	31
	<a href="#">Logan Central</a>	22	30
	<a href="#">Manly</a>	23	28
	<a href="#">Mount Gravatt</a>	22	29
	<a href="#">Oxley</a>	22	31
	<a href="#">Redcliffe</a>	23	28

# datapasta





*Program*

Tidy datasets are all alike,  
but every messy dataset  
is messy in its own way!

*- Hadley Wickham*



# A Tidy dataset

	Name	Gender	Age
1	Phil	Male	54
2	May	Female	46
3	Mack	NA	31

# A variable has its own column

	Var. 1	Var. 2	Var. 3
Obs. 1	A	B	C
Obs. 2	D	E	F
Obs. 3	G	H	I

# An observation has its own row

	Var. 1	Var. 2	Var. 3
Obs. 1	A	B	C
Obs. 2	D	E	F
Obs. 3	G	H	I

# An value has its own cell

	Var. 1	Var. 2	Var. 3
Obs. 1	A	B	C
Obs. 2	D	E	F
Obs. 3	G	H	I

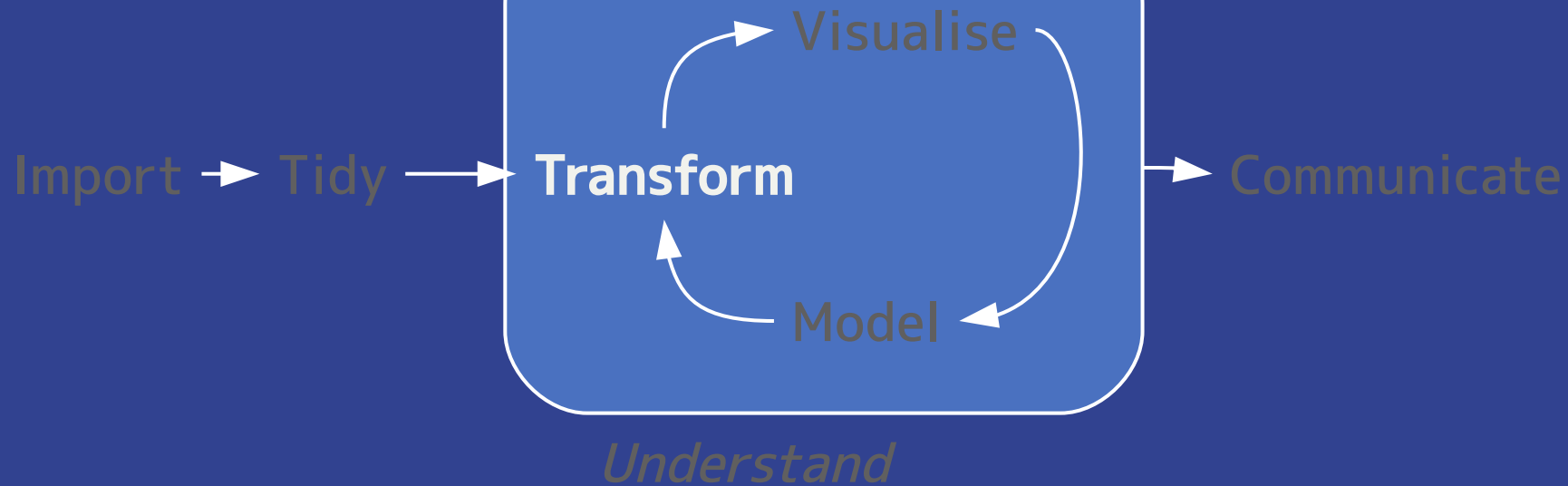


# Not only tidy, but also tame

- Use syntactical names for column names
- Use consistent case: snake\_case, camelCase, caterpillar.case
- Cast column type accordingly: <chr>, <dbl>, <lgl>, <date>, etc
- Treat <fct> carefully!
- Preferably turn implicit missing observation into explicit NA value

# Let's do practice!

- Open '002\_import-tidy-data.Rmd'
- You have **15 minutes** to play with it
- Do not forget to push your works into GitHub!



*Program*



dplyr : go wrangling



Artwork by @allison\_horst

## dplyr basic functions:

- **filter()** selects rows based on their values
- **mutate()** creates new variables
- **select()** picks columns by name
- **summarise()** calculates summary statistics
- **arrange()** sorts the rows

## tidyr basic functions:

- **gather()** wide-format >> long-format
- **spread()** long-format >> wide-format
- **fill()** fills value based on previous entry
- **complete()** turns implicit missing values into explicit


## Operators:

- **!** (not)
- **|** (or)
- **&** (and)
- **==, !=**
- **<, <=, >, >=**
- **%in%**
- **is.na()**

How can I  
chain?



- 
1. diputar
  2. dijilat
  3. dicelupin
  4. dimakan :D

- 
1. `putar(apa)`
  2. `jilat(apa, berapa_kali)`
  3. `celup(apa, ke)`
  4. `makan(apa, output)`

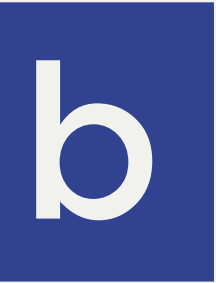
a

```
> oreo_putar ← putar(apa = "oreo")
> oreo_jilat ← jilat(apa = oreo_putar,
                     berapa_kali = 2)
> oreo_celup ← celup(apa = oreo_jilat,
                     ke = "susu")
> makan(apa = oreo_celup,
        output = "kenyang.perut")
```

a

```
> oreo_putar <- putar(apa = "oreo")
> oreo_jilat <- jilat(apa = oreo_putar,
                      berapa_kali = 2)
> oreo_celup <- celup(apa = oreo_jilat,
                      ke = "susu")
> makan(apa = oreo_celup,
        output = "kenyang.perut")
```





```
> makan(  
  celup(  
    jilat(  
      putar(apa = "oreo"),  
      berapa_kali = 2  
    ),  
    ke = "susu"  
  ),  
  output = "kenyang.perut"  
)
```

```
function(arg1, arg2, arg3, ... )
```

```
arg1 %>%  
  function(arg2, arg3, ... )
```

---

```
function(arg1, arg2, arg3, ... )
```

```
arg2 %>%  
  function(arg1, arg2=., arg3, ... )
```

# magrittr

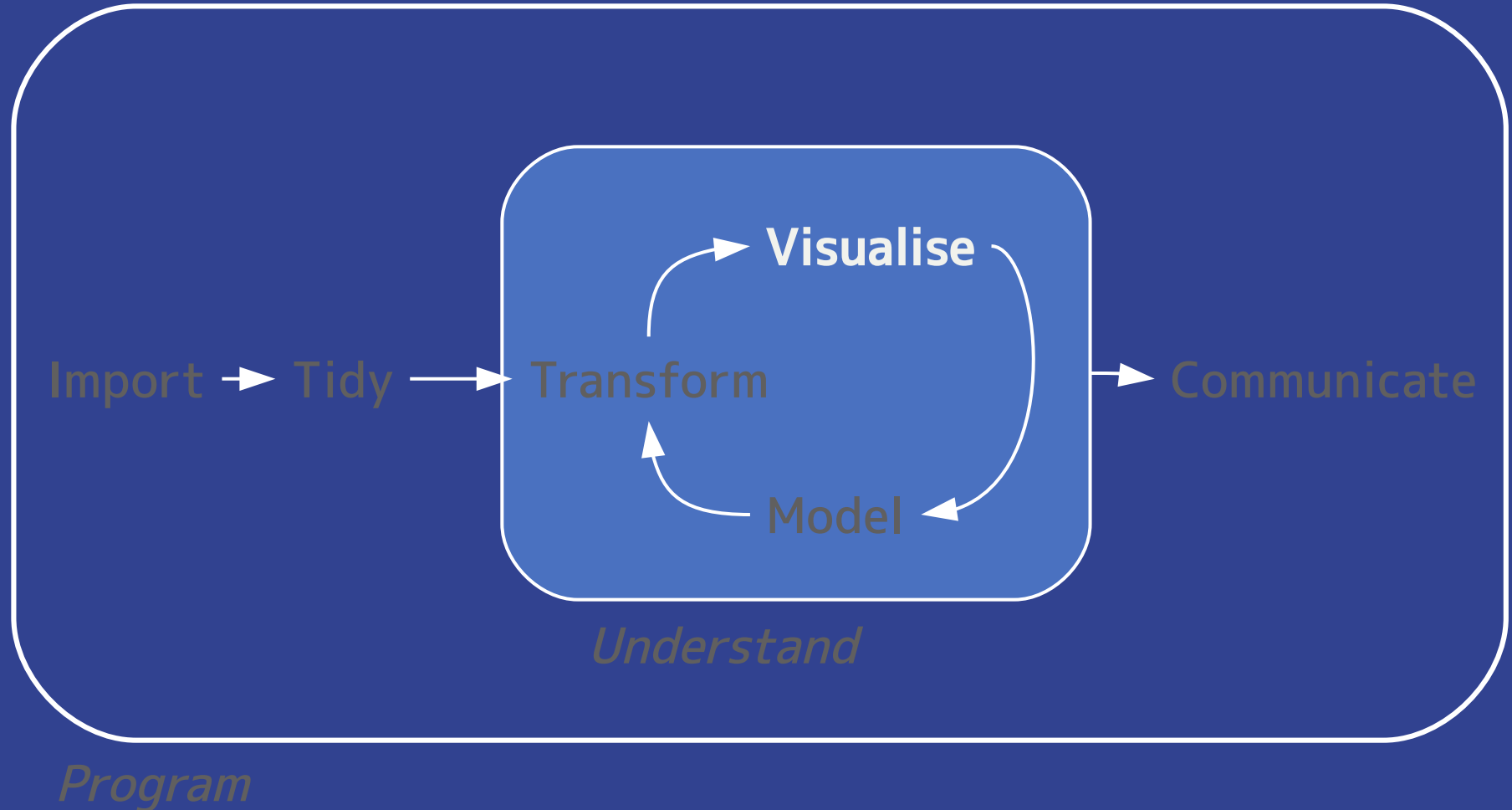




```
> putar(apa = "oreo") %>%  
  jilat(berapa_kali = 2) %>%  
  celup(ke = "susu") %>%  
  makan(output = "kenyang.perut")
```

# Let's do practice!

- Open '003\_transformasi.Rmd'
- You have **30 minutes** to play with it
- Do not forget to push your works into GitHub!





Artwork by @allison\_horst

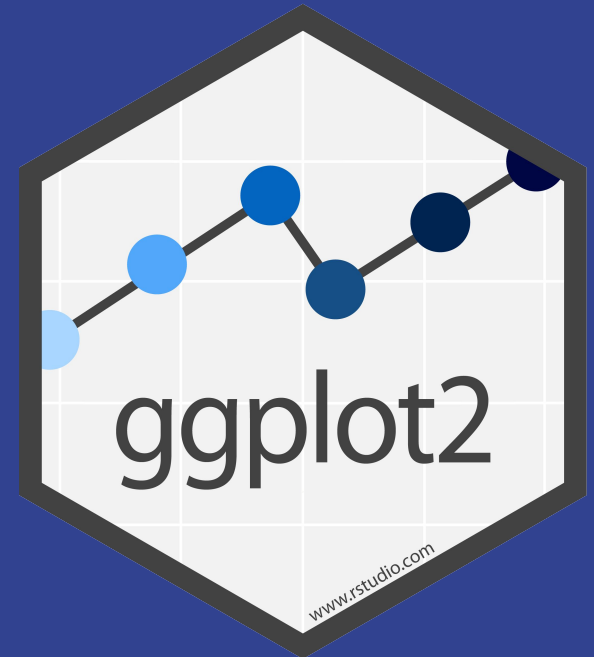
```
ggplot(data) +  
  geom_X(mapping=aes(...)) +  
  ...
```

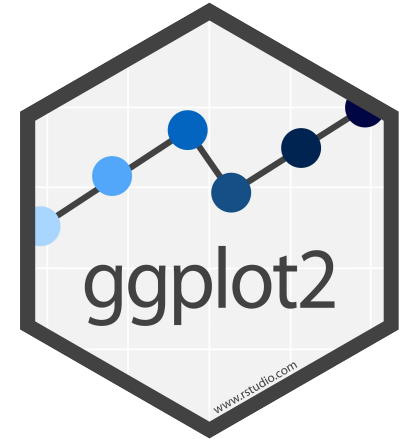
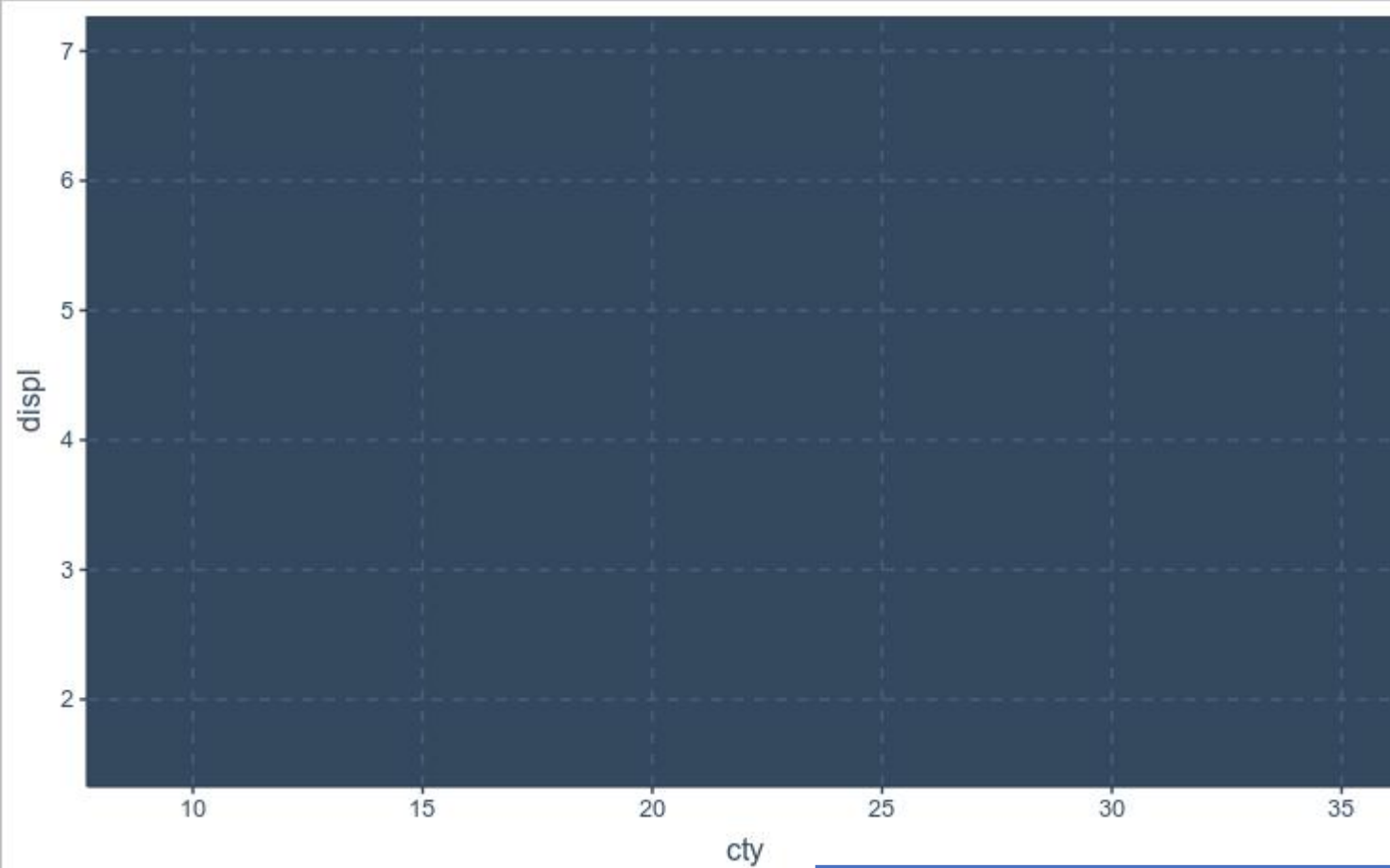
```
ggplot(data, mapping=aes( ... )) +  
  geom_X() +  
  ...
```

---

```
data %>%  
  ggplot(mapping=aes( ... )) +  
  geom_X() +  
  ...
```

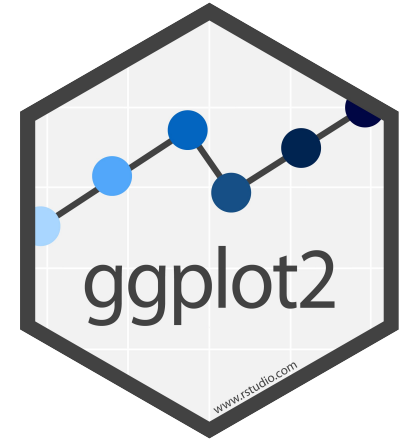
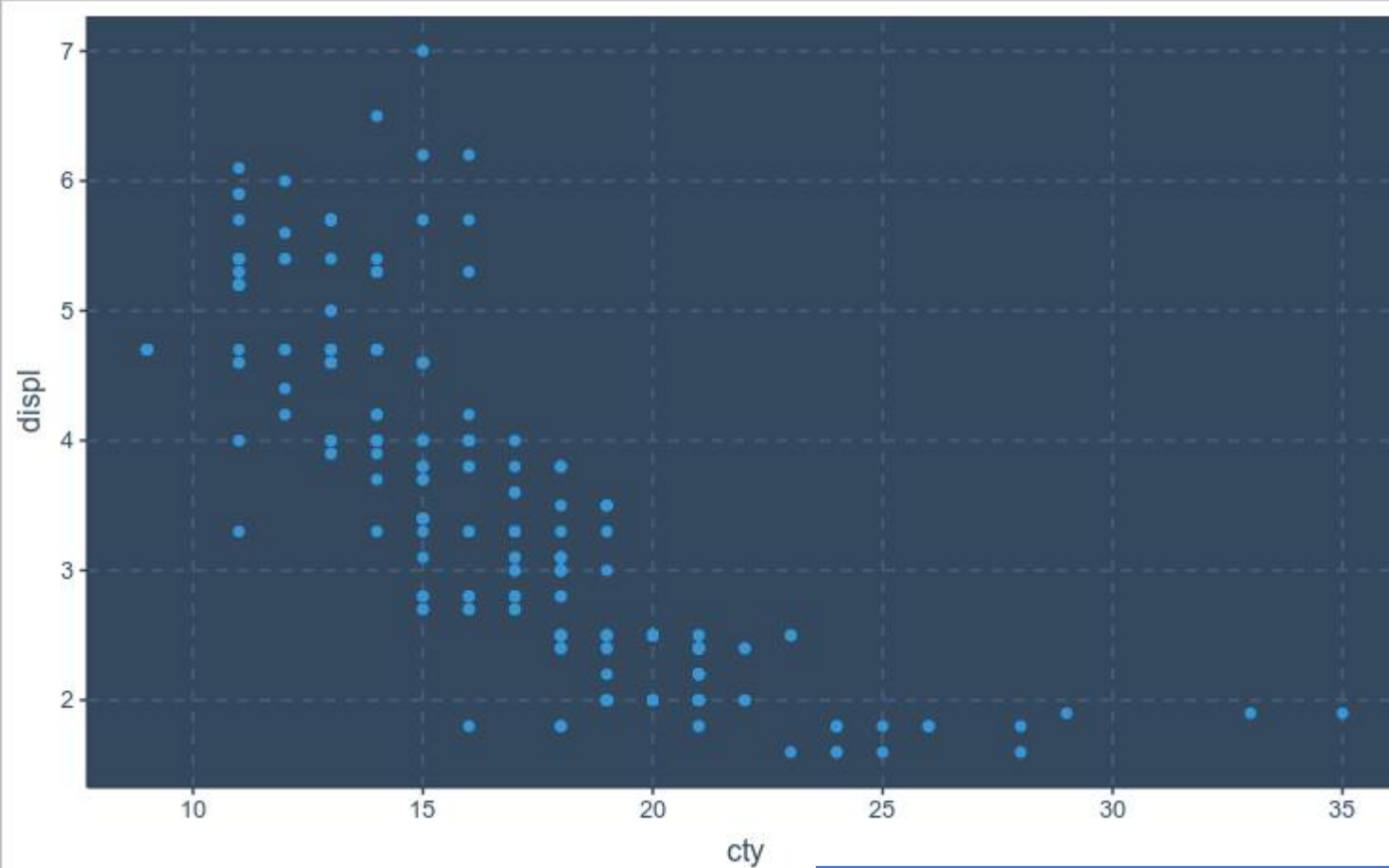
# ggplot2



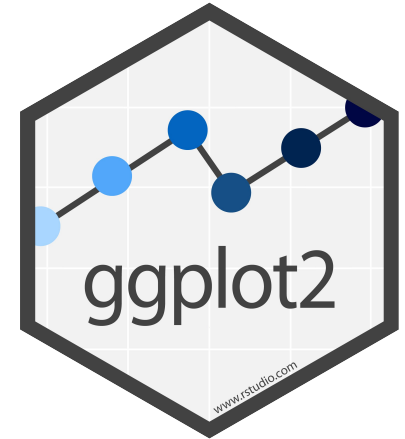
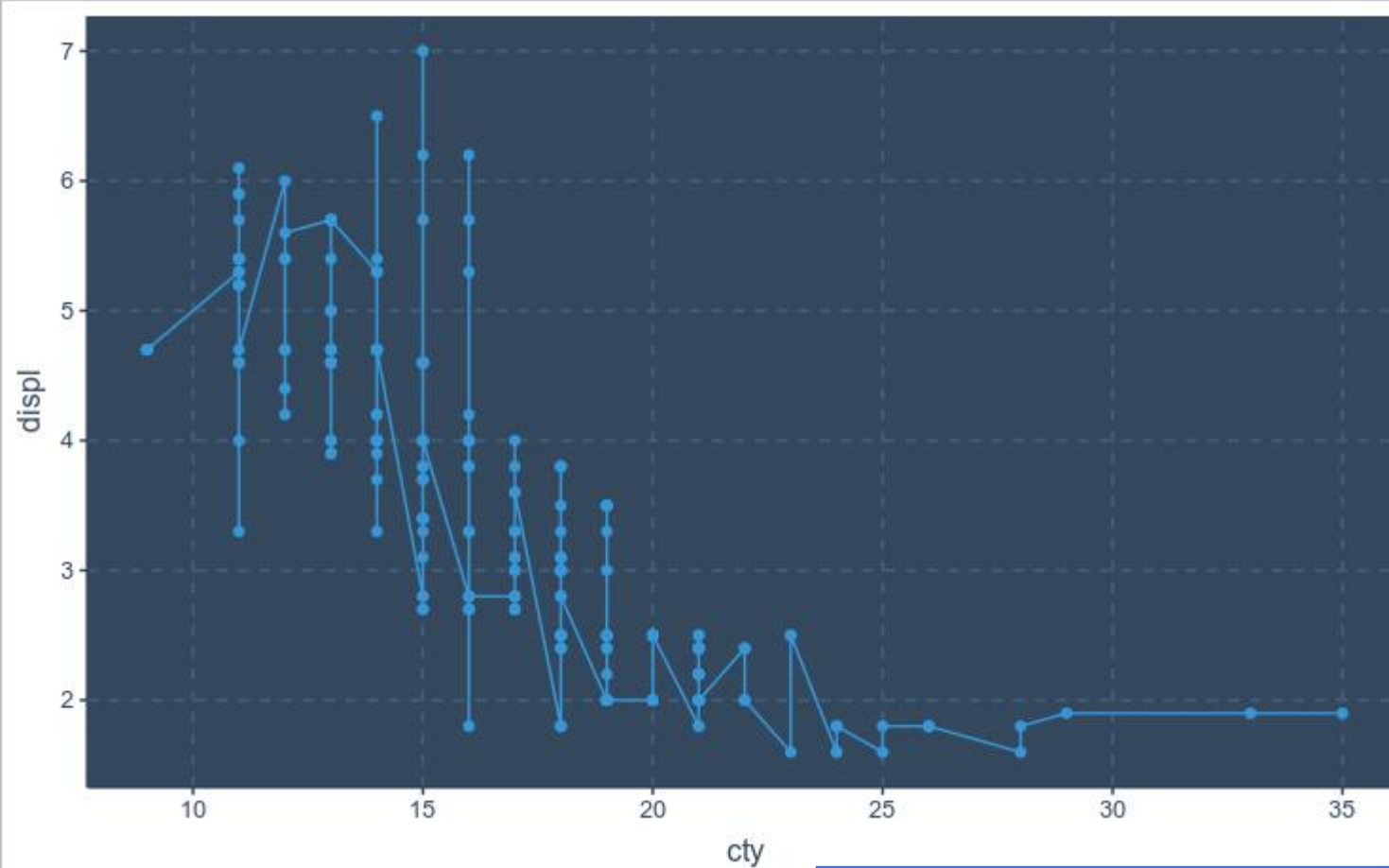


```
ggplot(mpg, aes(x = cty, y = displ))
```

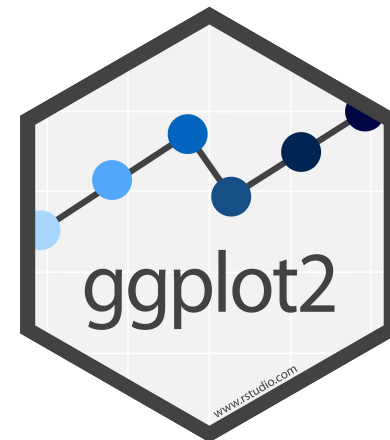
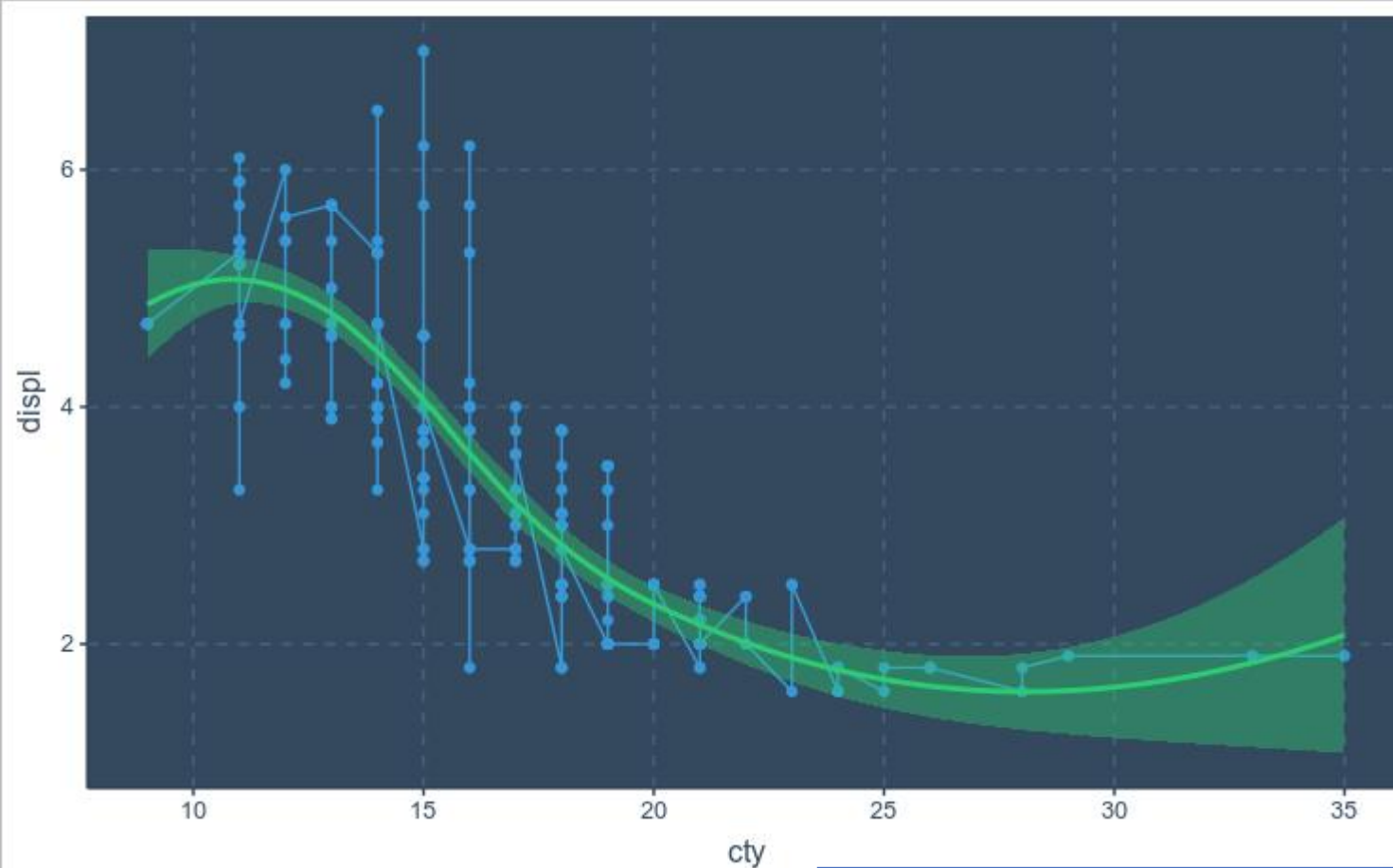




```
ggplot(mpg, aes(x = cty, y = displ)) +  
  geom_point()
```



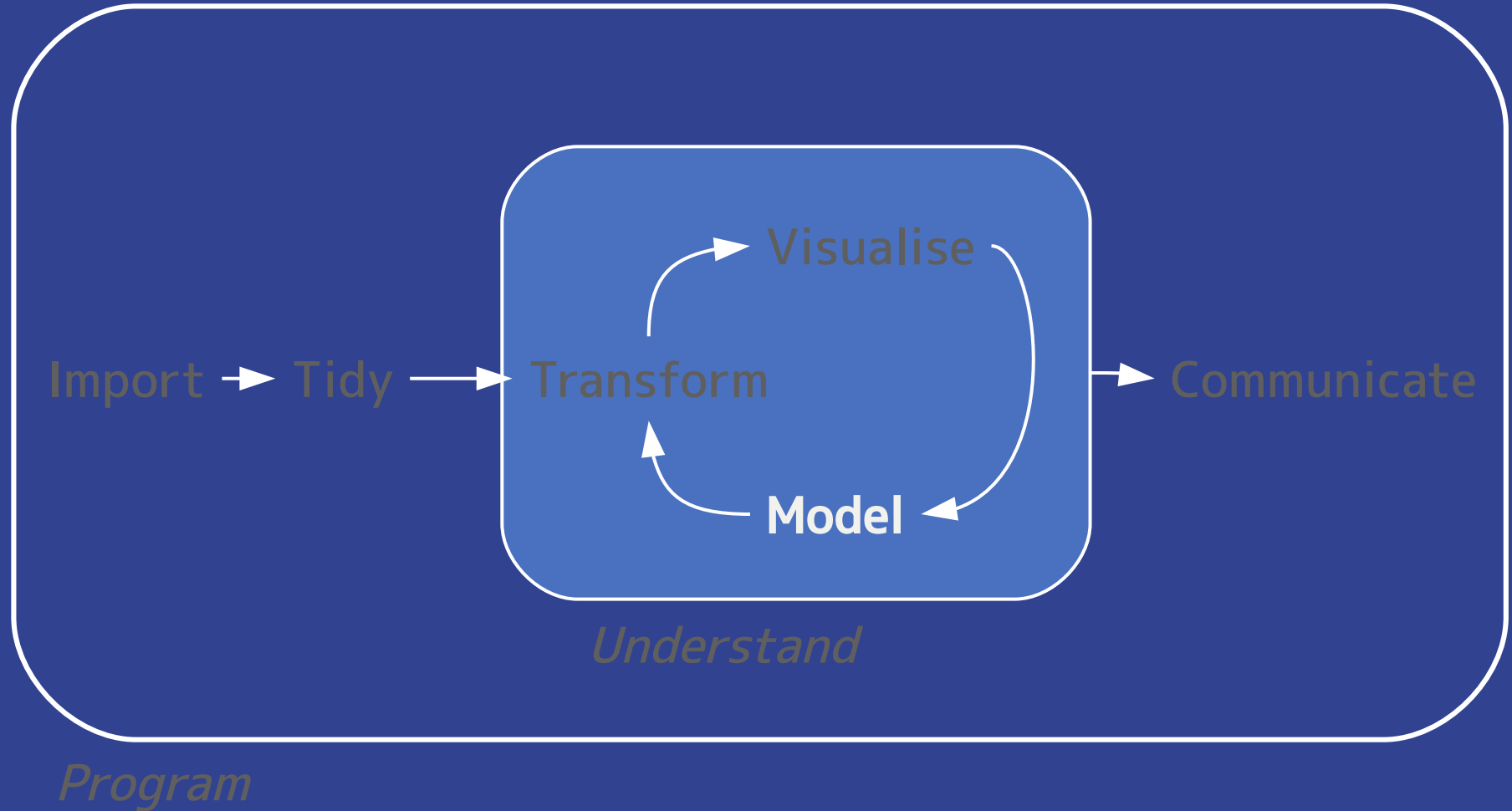
```
ggplot(mpg, aes(x = cty, y = displ)) +  
  geom_point() +  
  geom_line()
```

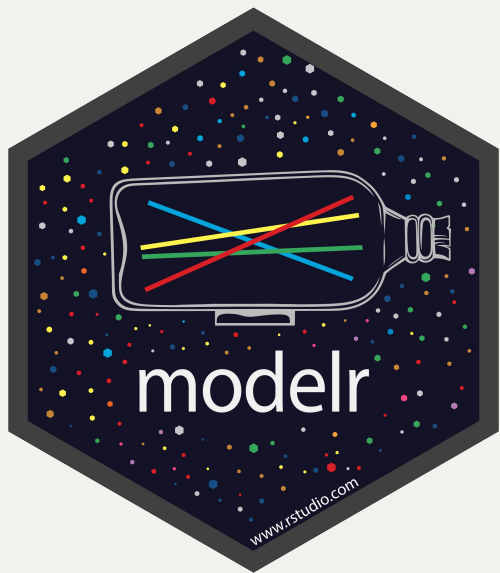


```
ggplot(mpg, aes(x = cty, y = displ)) +  
  geom_point() +  
  geom_line() +  
  geom_smooth()
```

# Let's do practice!

- Open '004\_visualisasi.Rmd'
- You have **15 minutes** to play with it
- Do not forget to push your works into GitHub!





A low dimensional  
description of a  
higher dimensional  
data set

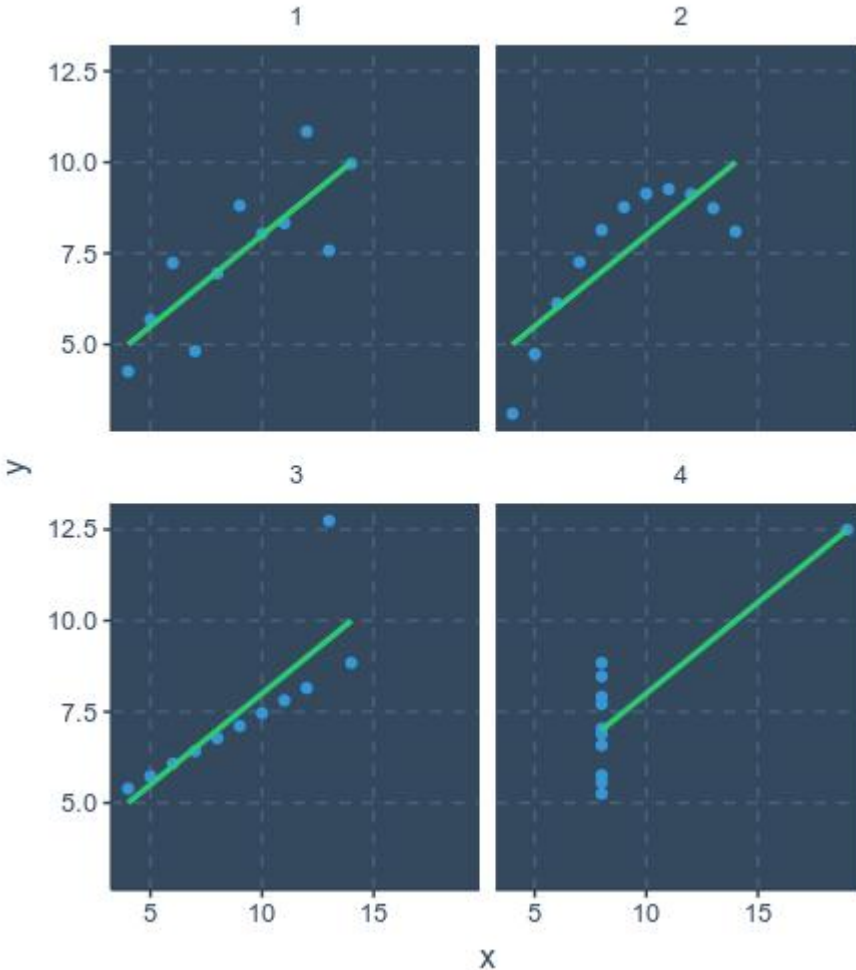


Outcome ~ Predictor/Explanatory

To predict

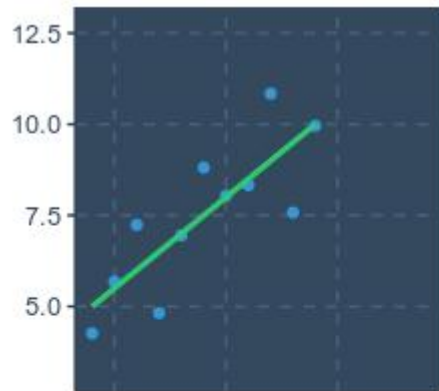
To explain

All models are  
wrong, but some are  
useful – George Box

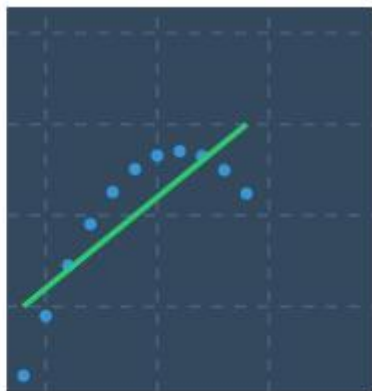


# Anscombe's Quartet

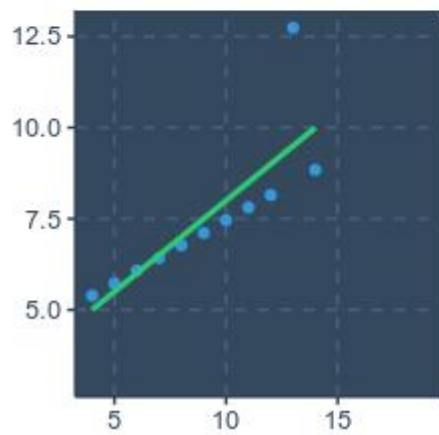
1



2



3



4



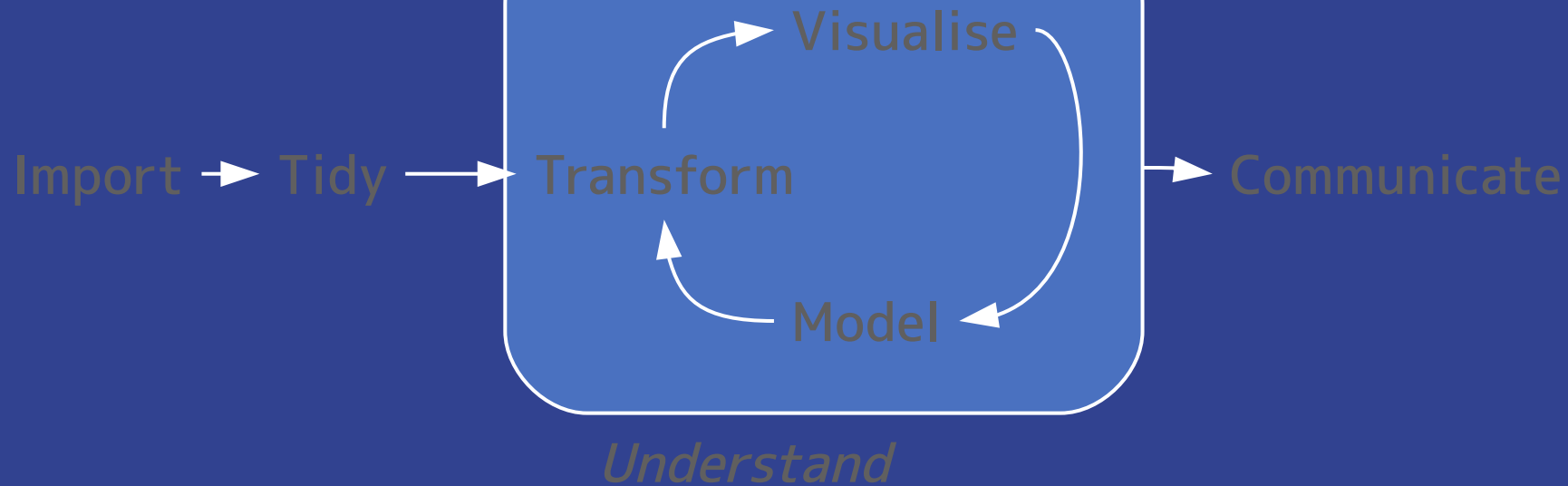
x

Mean of x	9	exact
Sample variance of x	11	exact
Mean of y	7.50	to 2 decimal places
Sample variance of y	4.125	$\pm 0.003$
Correlation between x and y	0.816	to 3 decimal places
Linear regression line	$y = 3.00 + 0.500x$	to 2 and 3 decimal places, respectively
$R^2$	0.67	to 3 decimal places



# Let's do practice!

- Open '005\_model.Rmd'
- You have **40 minutes** to play with it
- Do not forget to push your works into GitHub!



*Program*

```
x ← something  
for (i in seq_along(x)) {  
  function(x[[i]])  
}
```

```
x ← something  
*apply(x, function)
```

```
x ← something  
map_*(x, function)
```

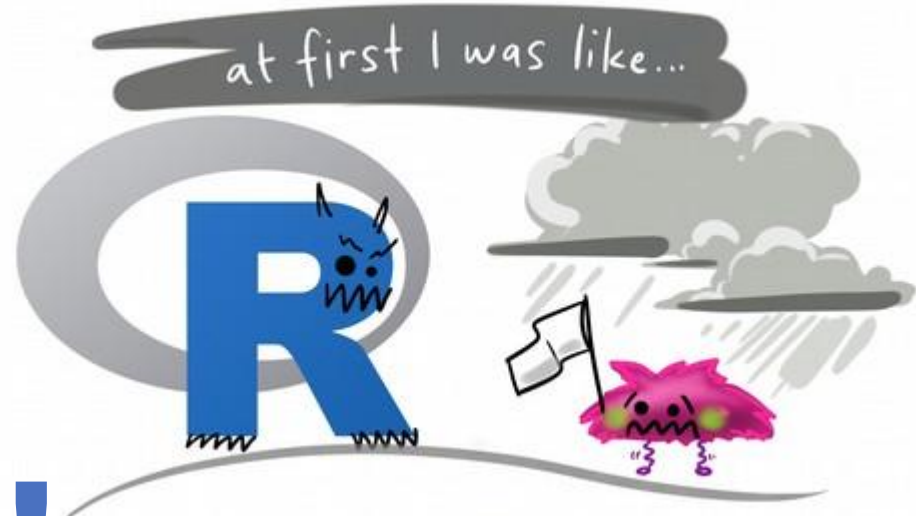
# purrr



# Let's do practice!

- Open '006\_iterasi.Rmd'
- You have **20 minutes** to play with it
- Do not forget to push your works into GitHub!

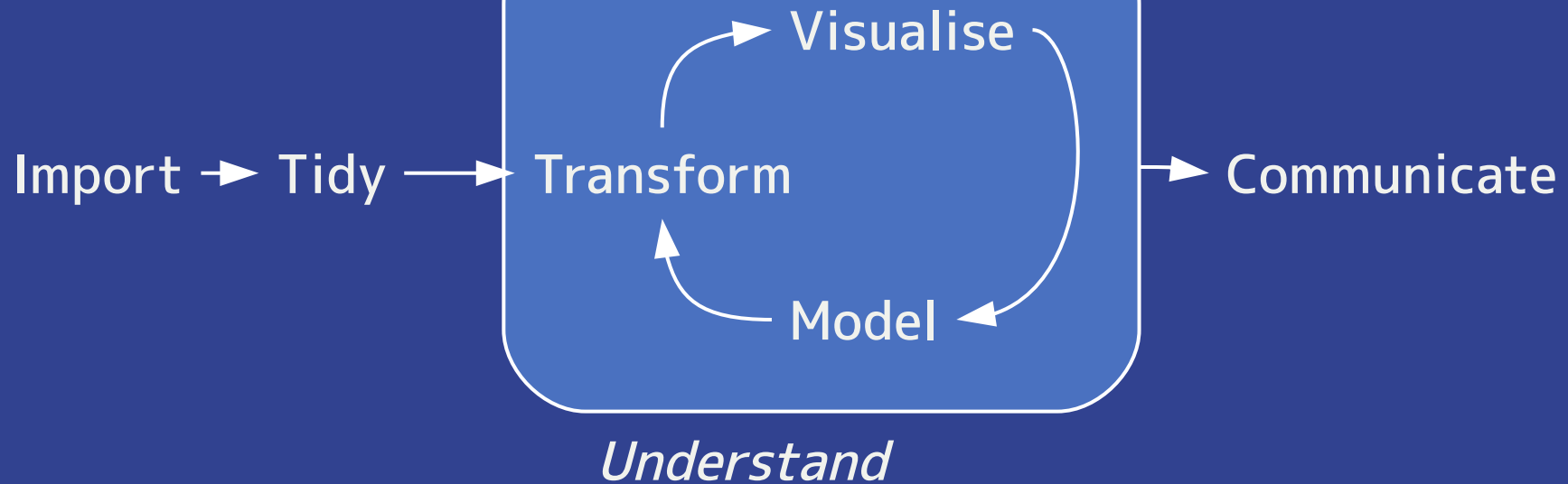
# Congrats!



...but now it's like...



Artwork by @allison\_horst



*Program*

```
> contact_me(  
    name      = "Muhammad Aswan Syahputra",  
    email     = aswansyahputra@sensolution.id  
    phone     = +62 822 3465 3816  
    twitter   = @aswansyahputra_  
)  
> ...
```