



Zen
programming

Situations

You have a deadline tomorrow and prepare to leave at 18h30.
At 18h29, you add a label to a plot and get :

0x1016e8f20. This is a serious error. This application, or a library it uses, is using an invalid context and is thereby contributing to an overall degradation of system stability and reliability

Situations

You have a deadline tomorrow and prepare to leave at 18h30.
At 18h29, you add a label to a plot and get :

0x1016e8f20. This is a serious error. This application, or a library it uses, is using an invalid context and is thereby contributing to an overall degradation of system stability and reliability

You are going to a workshop and planning to show your new, shiny R package you developed with *Bighard Revolution pro-R™* and the R package *coolstuffr* from last week.

Nobody manages to install it.

Situations

You don't care about all those fancy new software, you're an ecologist after all.
You installed R some years ago and do stuff you consider basic with it.

- The rightmost break for the "months", "quarters" and "years" cases of `hist.POSIXlt()` has been increased by a day. (Inter alia, fixes [PR#15717](#).)
- The handling of `DF[i,] <- a` where `i` is of length 0 is improved. (Inter alia, fixes [PR#15718](#).)
- `hclust()` gains a new method "ward.D2" which implements Ward's method correctly. The previous "ward" method is "ward.D" now, with the old name still working. Thanks to research and proposals by Pierre Legendre.
- The `sunspot.month` dataset has been amended and updated from the official source, whereas the `sunspots` and `sunspot.year` datasets will remain immutable. The documentation and source links have been updated correspondingly.

April 2014

Situations

You manage a small group of people doing weather forecasting. Although people work together in discussions and meetings, the code modelling rain has five different implementations.

You have a look and cannot understand any of them.

Situations

You manage a small group of people doing weather forecasting. Although people work together in discussions and meetings, the code modelling rain has five different implementations.

You have a look and cannot understand any of them.

You manage yourself.

You have a look at your code and cannot understand it.

Problems ?

It's not only about you !

Problems ?

Change one thing: everything breaks

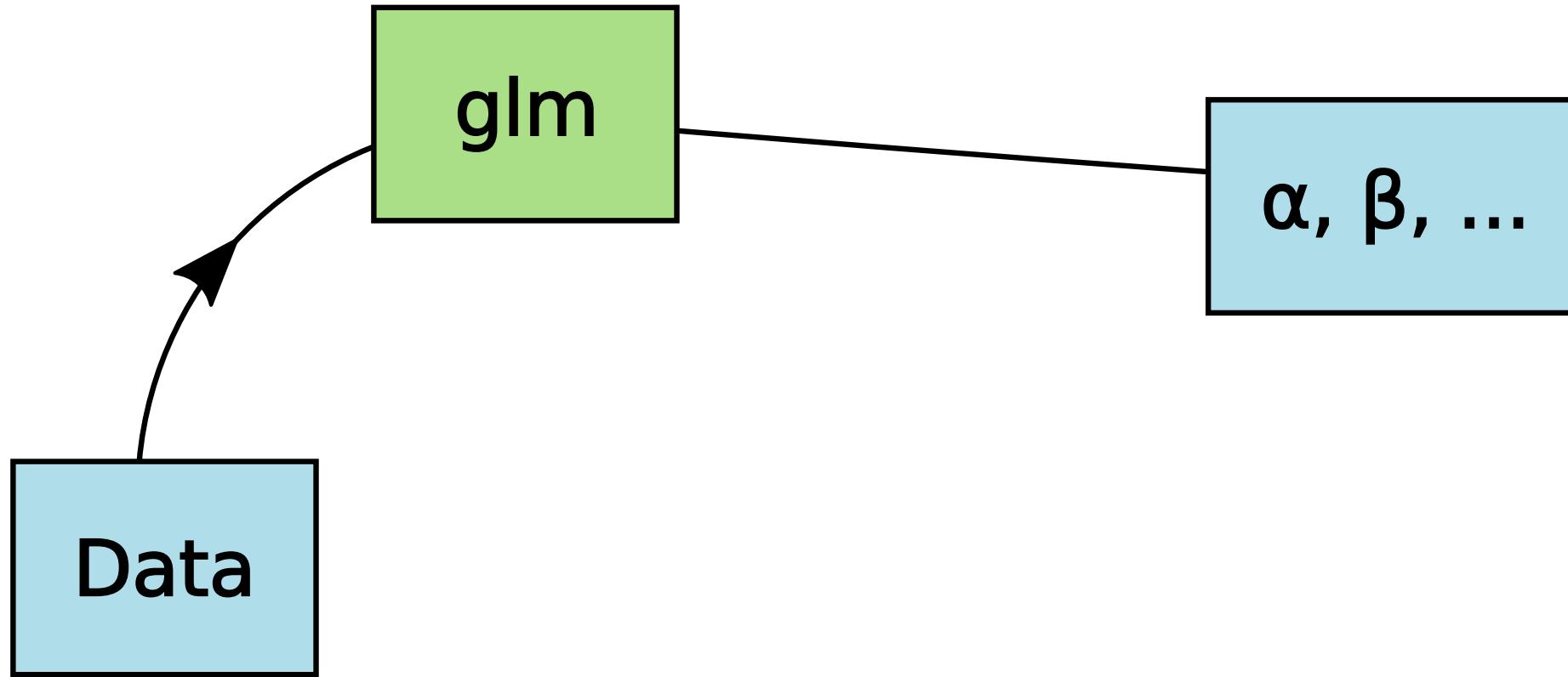
Something stops working at the last minute:
no way to reverse changes

Cryptic, untractable errors

Things never work on others' computers

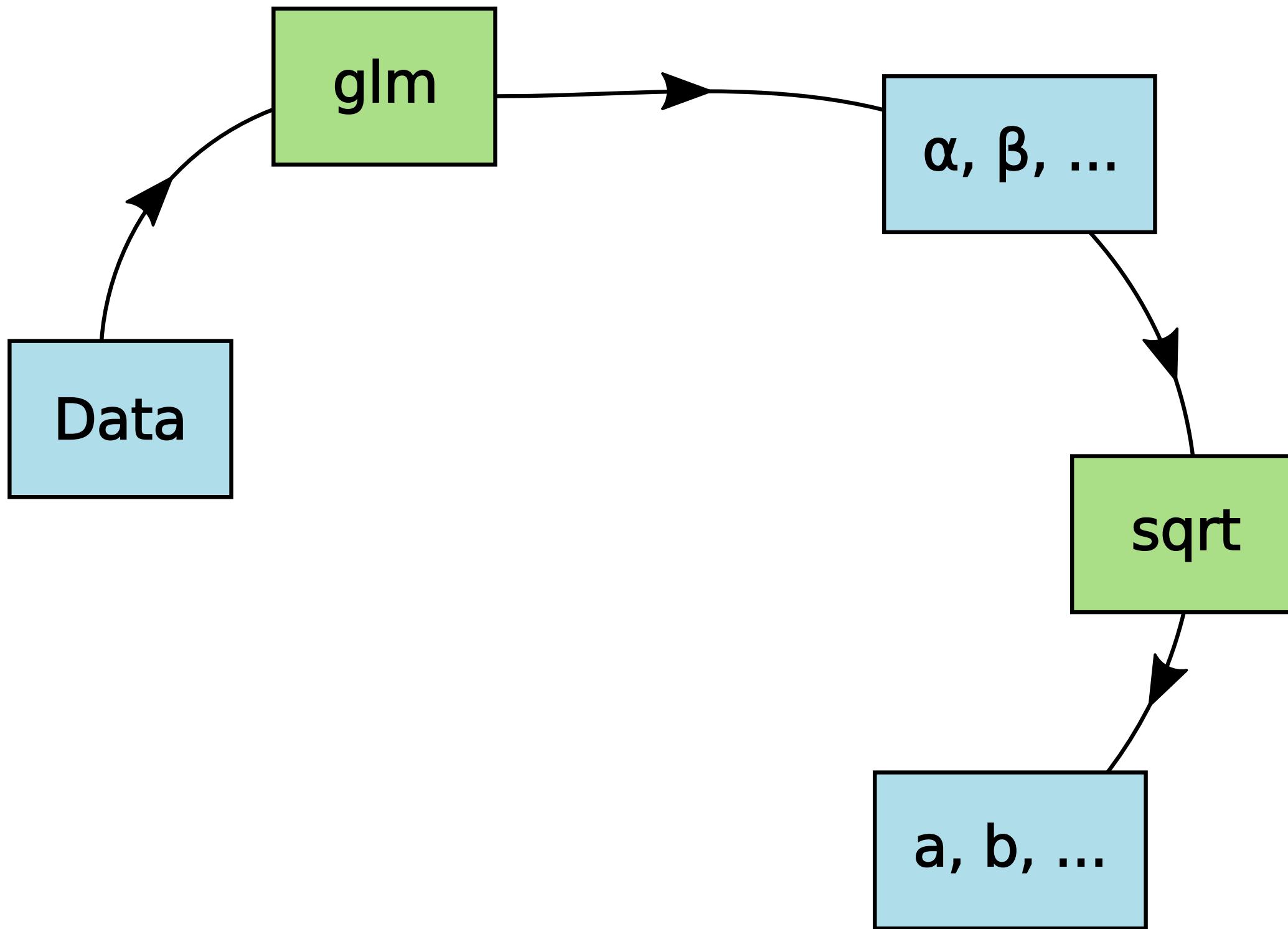
Unsure about what your program does

Why do we program ?

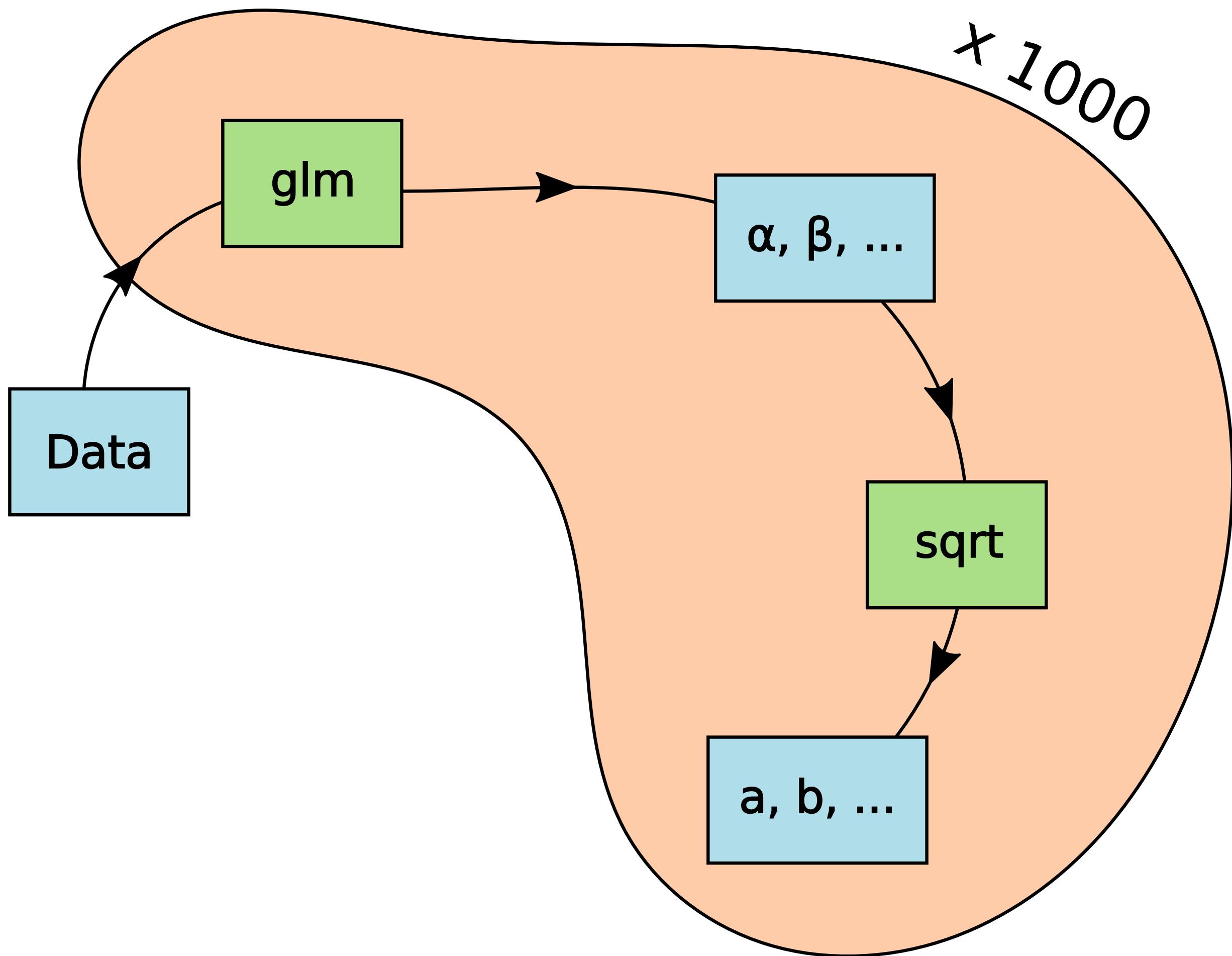


Why do we program ?

Why do we program ?

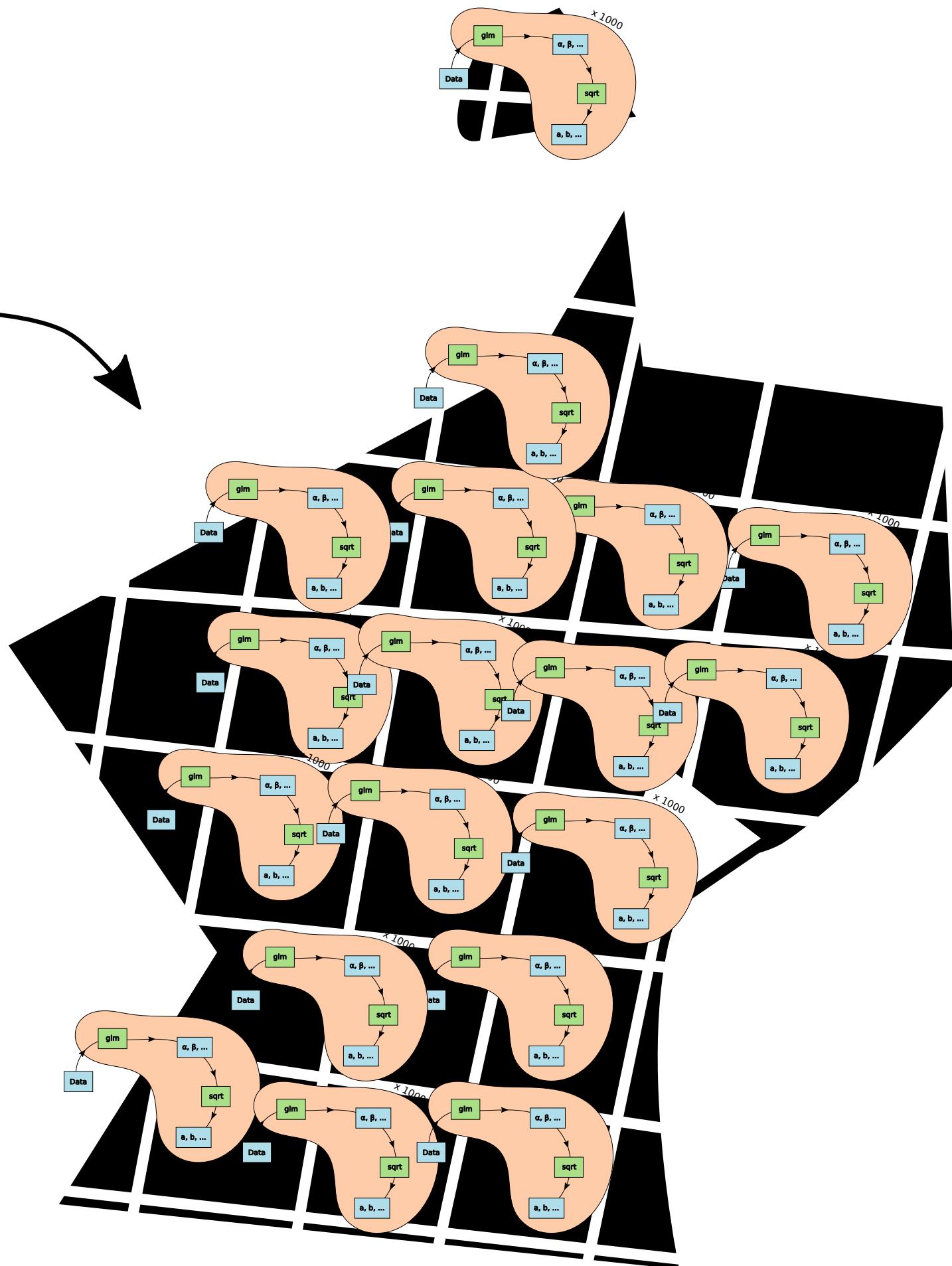


Why do we program ?



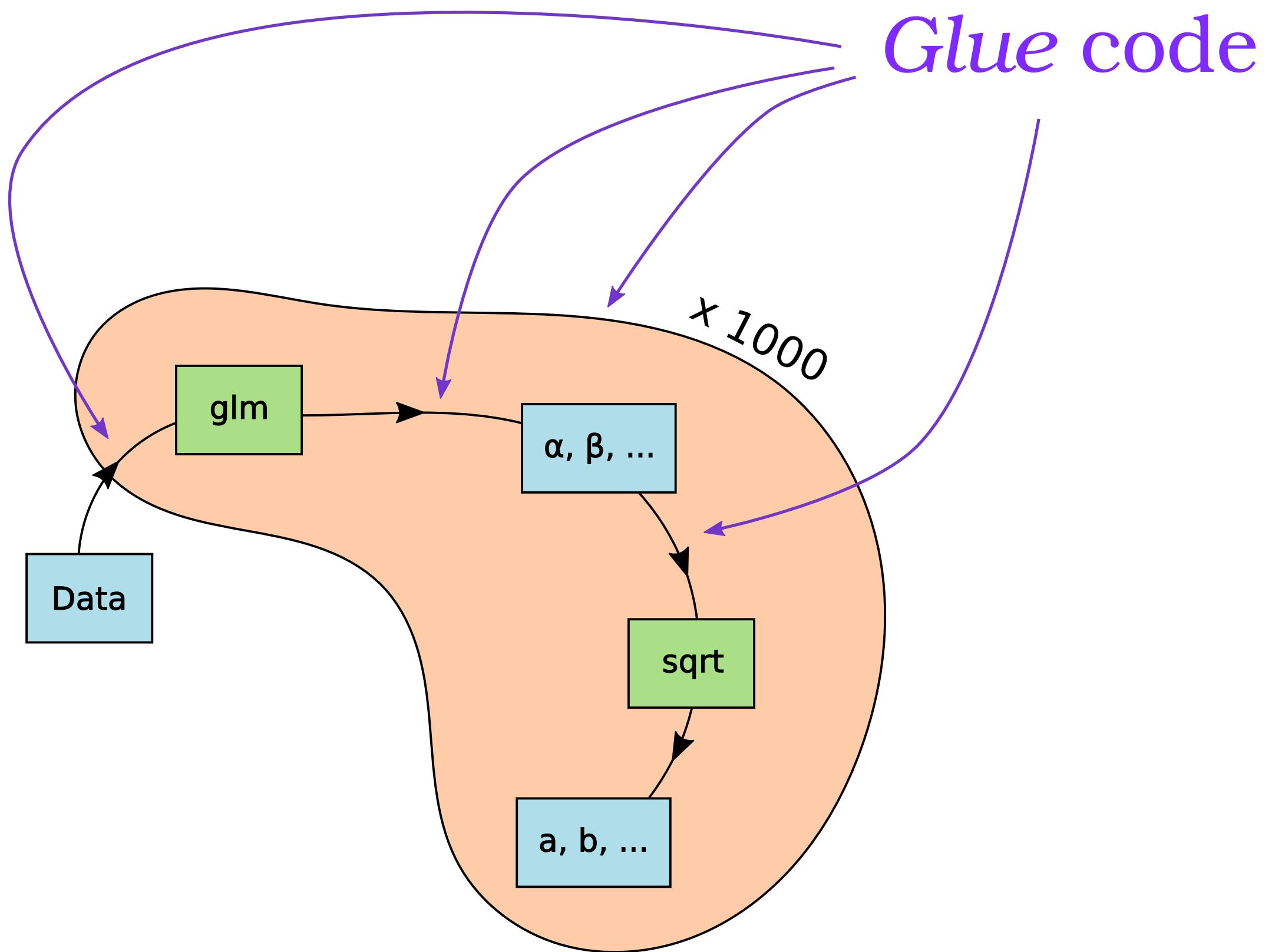
Why do we program ?

your favorite country



Why do we program ?

for *combining* tools



Why R ?

< this page is intentionnaly left blank >

Five ways to zen programming



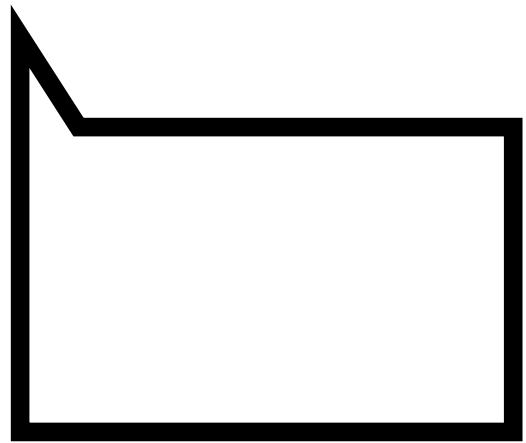
- Organize your workspace
- Choose the right data structures
- Write readable code
- Recognize programming patterns
- Use the right paradigm

Organize your files !

One project - one folder - one good name

Organize your files !

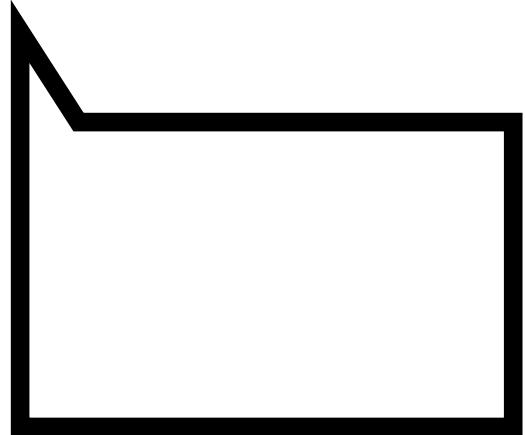
One project - one folder - one good name



Projet: "œil de tigre"

Organize your files !

One project - one folder - one good name



meet_nerdtalk

C:\Documents and settings\alex\Mes Documents\work\2016\meet_nerdtalk

68 characters

/home/alex/work/2016/meet_nerdtalk

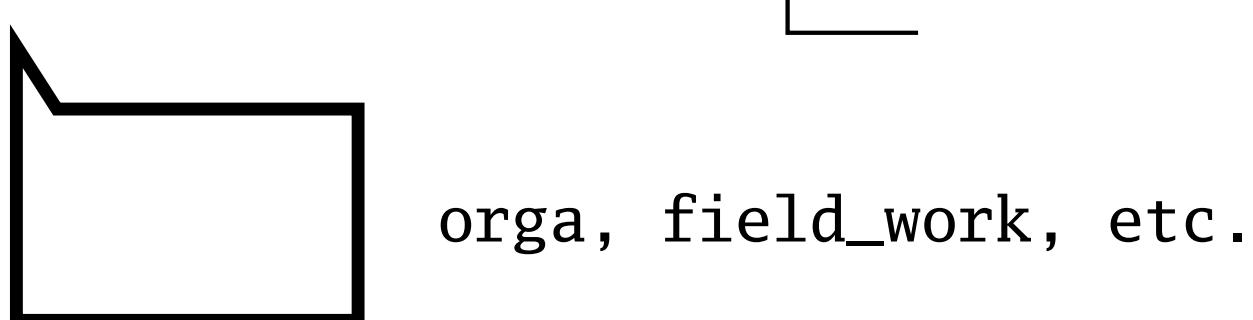
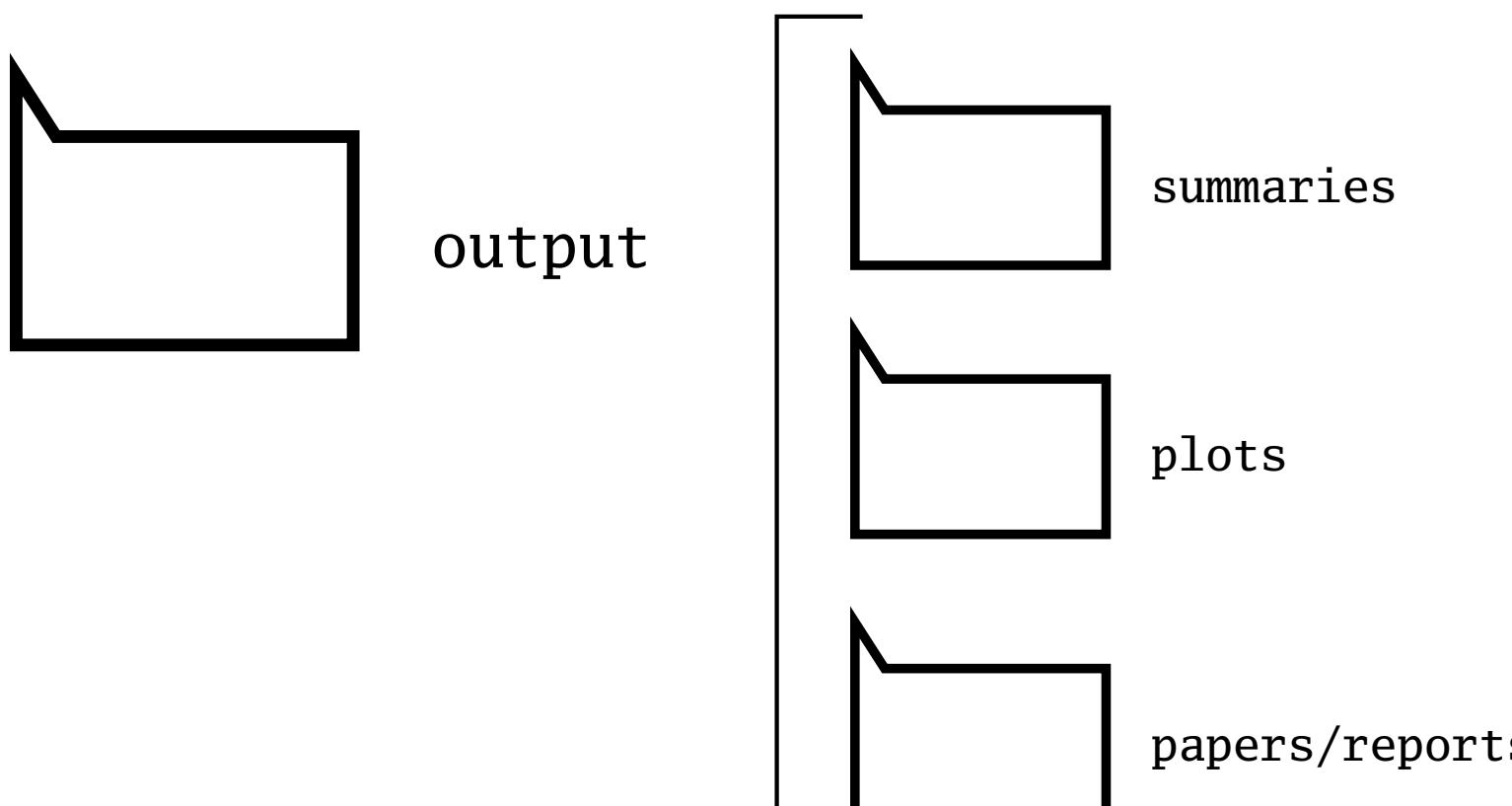
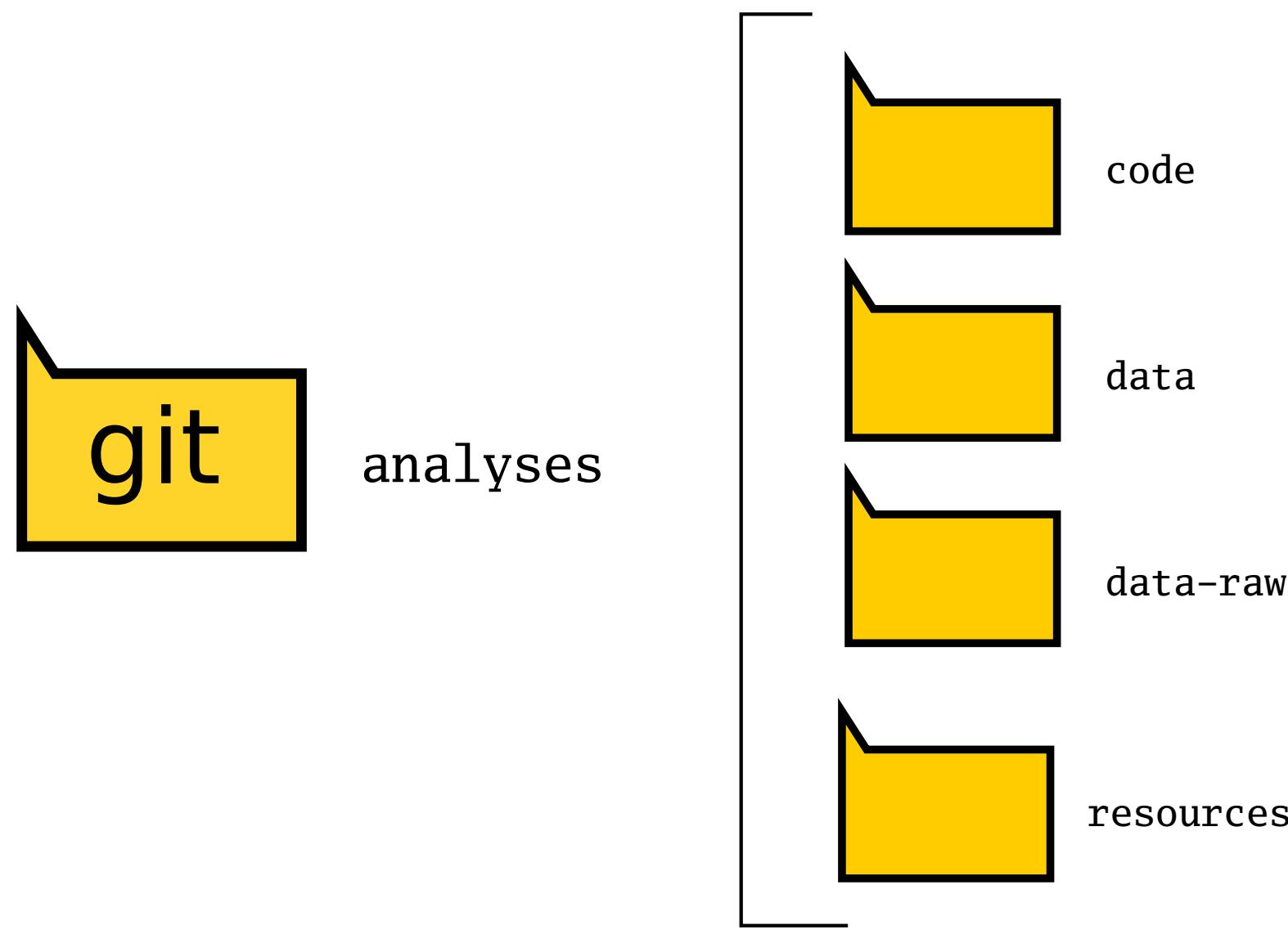
34 characters

Organize your files !

USASCII code chart

b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	Column →	0	0	0	0	1	1	0	1	0	1	0	1	1	0	1	1	1																																																																																							
Row ↓								0	1	2	3	4	5	6	7	SP	0	@	P	'	p	!	1	A	Q	a	q	"	2	B	R	b	r	#	3	C	S	c	s	\$	4	D	T	d	t	%	5	E	U	e	u	8	6	F	V	f	v	'	7	G	W	g	w	(8	H	X	h	x)	9	I	Y	i	y	*	:	J	Z	j	z	+	:	K	[k	{	,	<	L	\	l	l	-	=	M]	m	}	.	>	N	^	n	~	/	?	O	-	o	DEL
0	0	0	0	0	0	0		0	0	0	0	1	1	0	1	0	1	0	1	1	0	1	1	1																																																																																							
0	0	0	0	1	1	1		0	0	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1																																																																																						
0	0	1	0	0	1	0		0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0																																																																																						
0	0	1	1	1	1	0		0	0	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1																																																																																						
0	1	0	0	0	0	0		0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0																																																																																						
0	1	0	1	0	1	0		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1																																																																																						
0	1	0	1	1	0	1		0	1	0	1	1	0	1	0	1	1	0	1	1	0	1	0	1	1																																																																																						
0	1	1	0	0	0	0		0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0																																																																																						
0	1	1	1	1	1	1		0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1																																																																																						
1	0	0	0	0	0	0		1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0																																																																																						
1	0	0	0	1	1	1		1	0	0	0	1	1	1	1	0	1	1	1	1	1	1	0	1	1																																																																																						
1	0	1	0	0	0	0		1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0																																																																																						
1	0	1	1	0	1	1		1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1																																																																																						
1	1	0	0	0	0	0		1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0																																																																																						
1	1	0	0	1	1	0		1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1	1	0																																																																																						
1	1	1	0	0	0	0		1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0																																																																																						
1	1	1	1	0	1	0		1	1	1	1	0	1	0	0	1	1	1	0	1	0	0	1	1	0																																																																																						
1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1																																																																																						

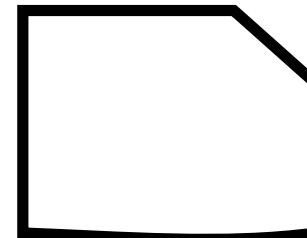
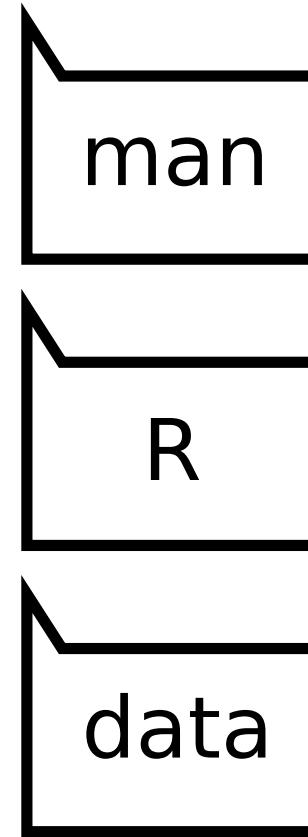
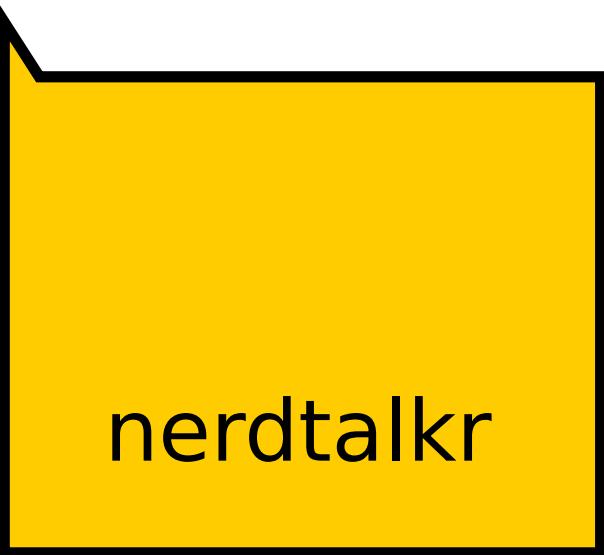
Organize your files !



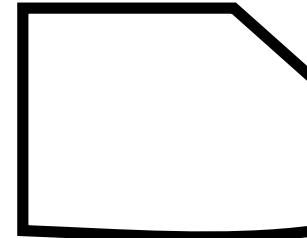
git report_comite_these

git draft_yosemite

Organize your files !



DESCRIPTION



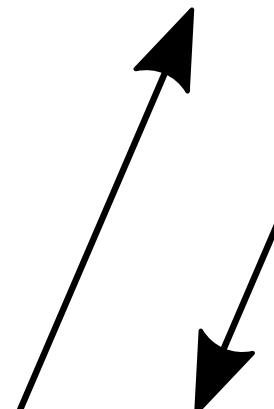
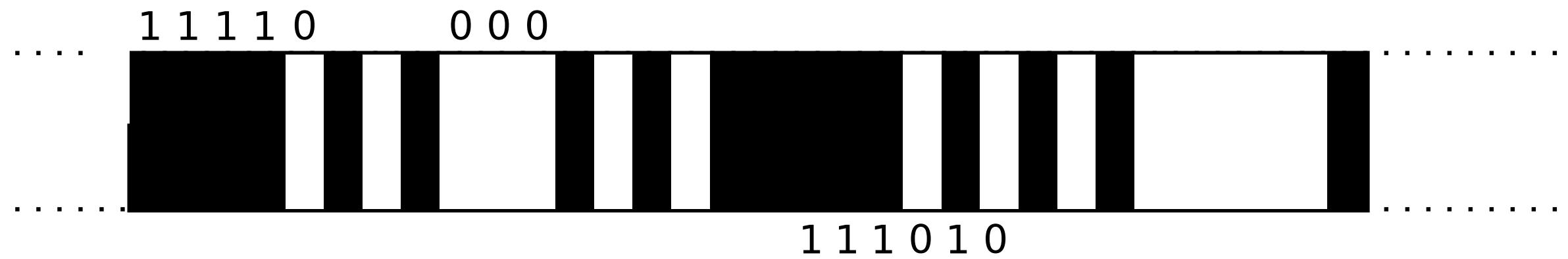
NAMESPACE

Benefits: Portability checks, unit tests, compiled code,
easy installation on other computers

Data types

A what ?

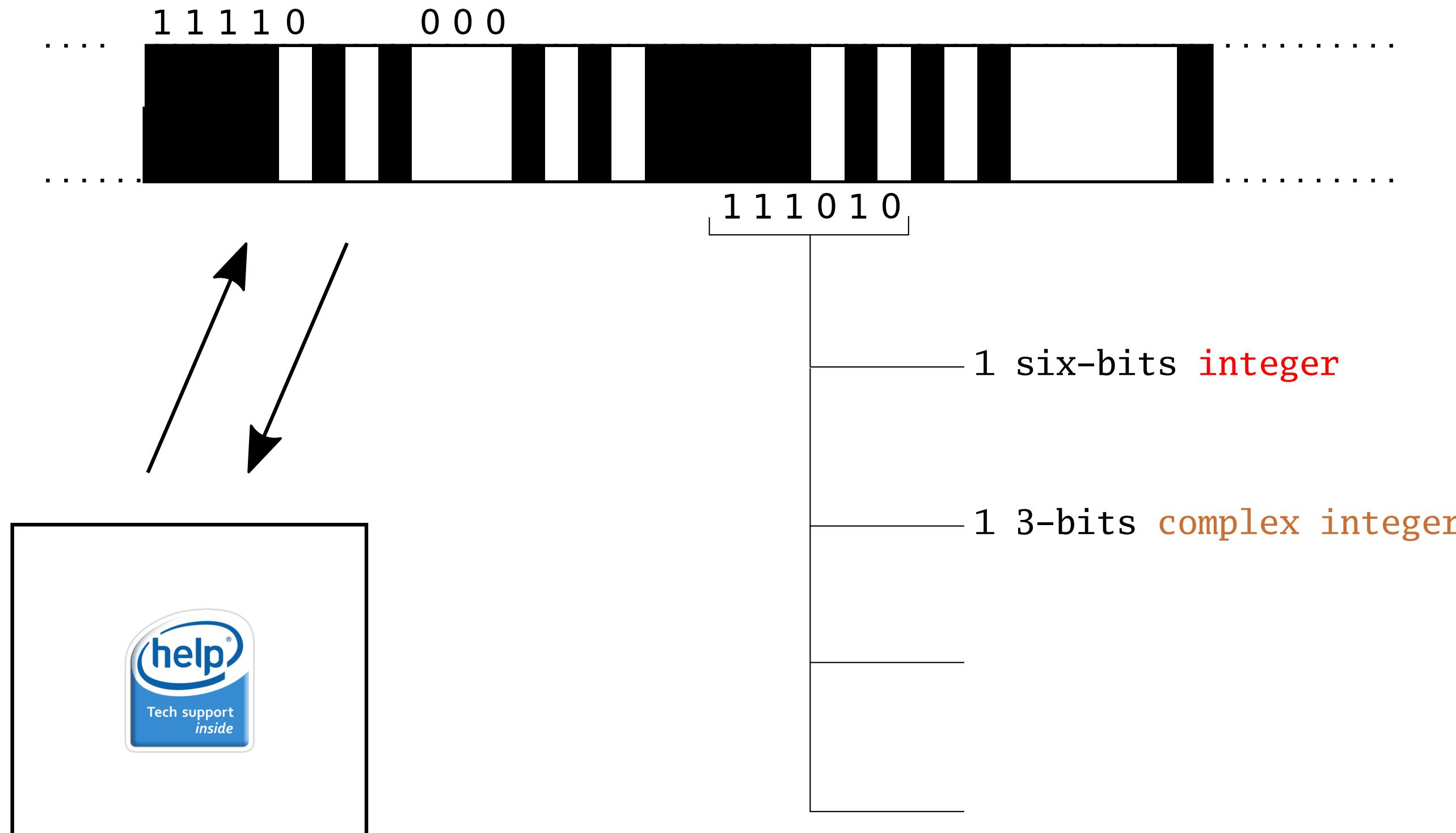
« A data type in a programming language is a set of data with values having predefined characteristics. »
techtarget.com



Data types

A what ?

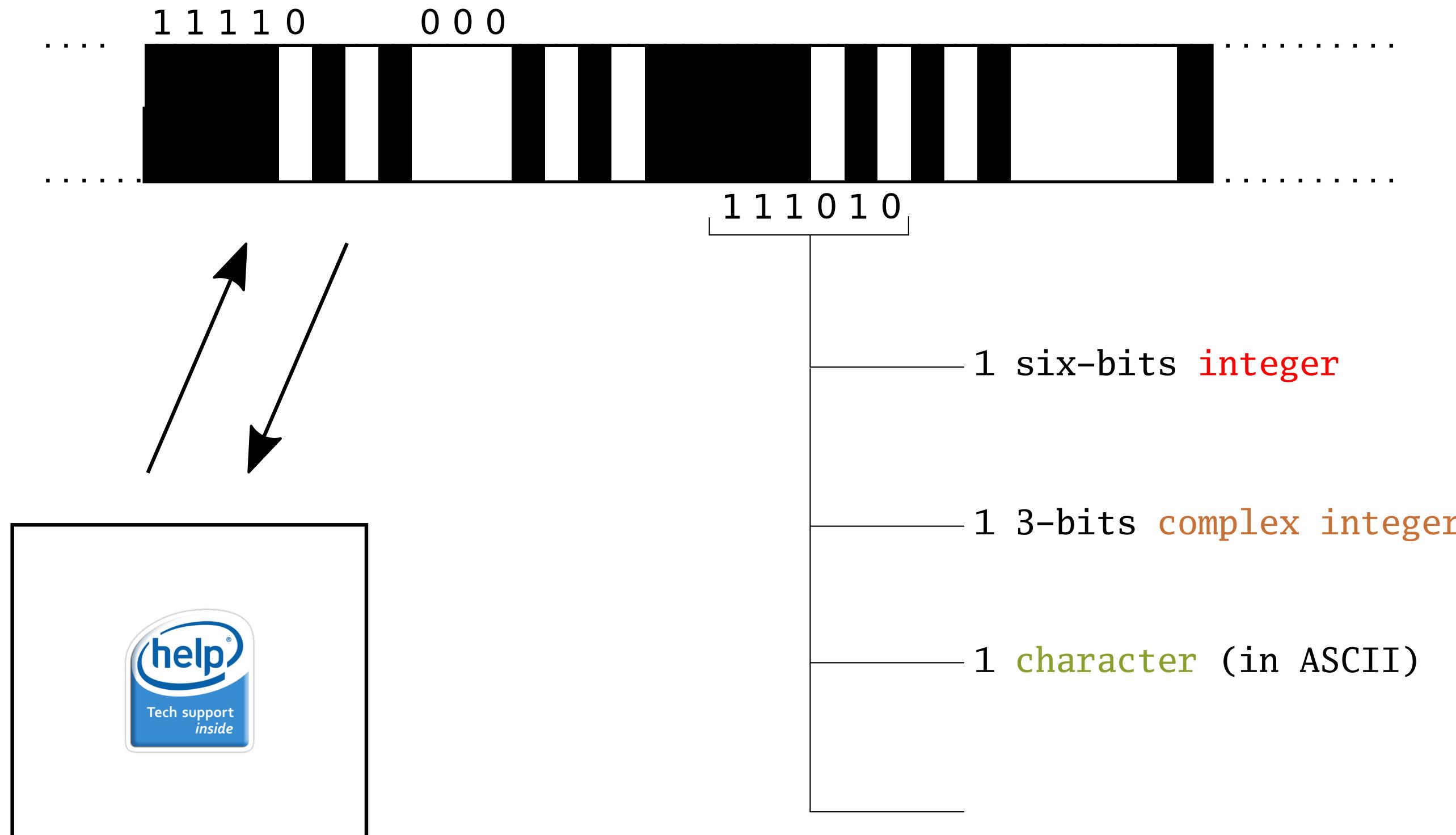
« A data type in a programming language is a set of data with values having predefined characteristics. »
techtarget.com



Data types

A what ?

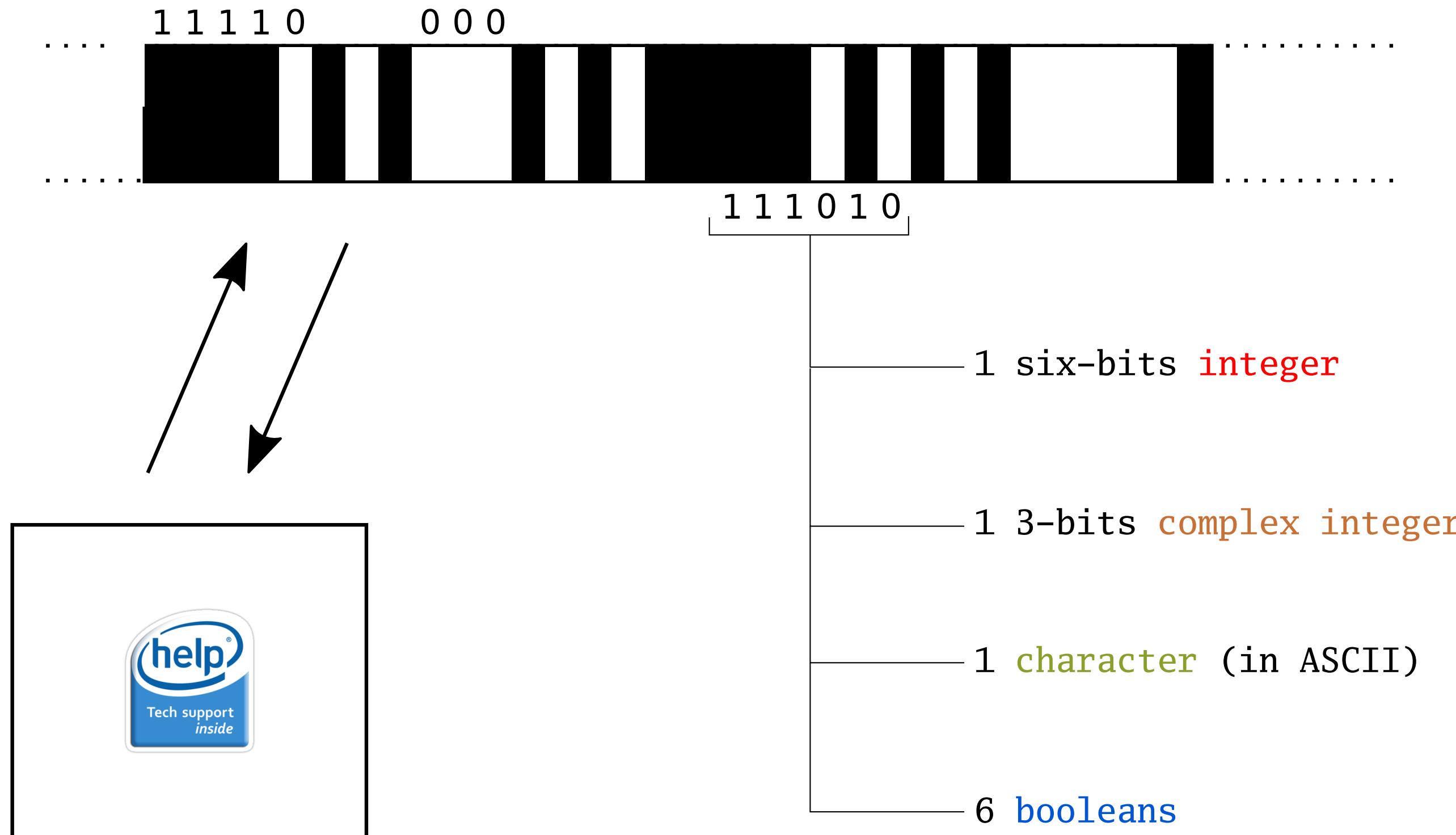
« A data type in a programming language is a set of data with values having predefined characteristics. »
techtarget.com



Data types

A what ?

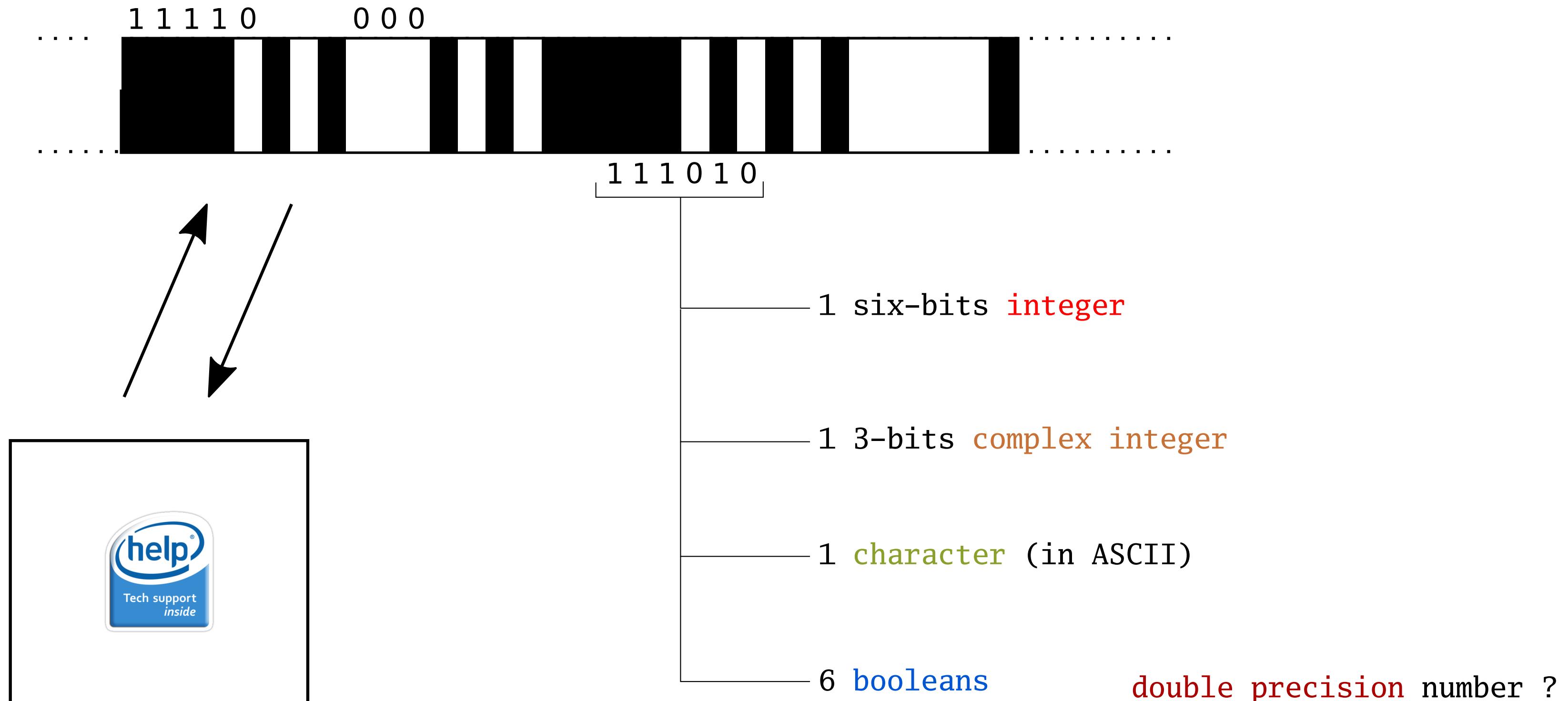
« A data type in a programming language is a set of data with values having predefined characteristics. »
techtarget.com



Data types

A what ?

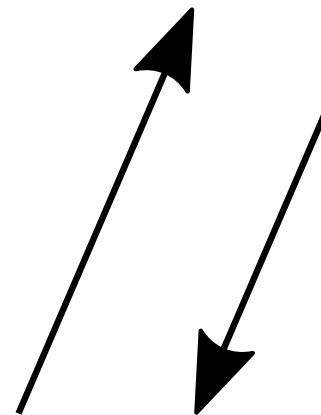
« A data type in a programming language is a set of data with values having predefined characteristics. »
techtarget.com



Data structures

A what ?

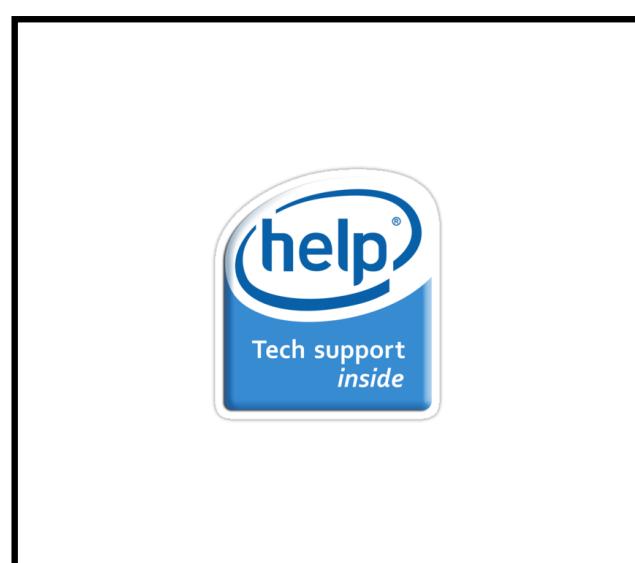
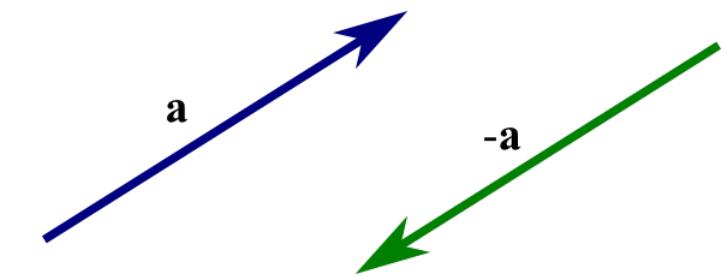
« A data structure is a particular way of organizing data in a computer so that it can be used efficiently »
Wikipedia.



Data structures

A what ?

« A data structure is a particular way of organizing data in a computer so that it can be used efficiently »
Wikipedia.



(1,2)

"the brown fox
jumps over the lazy dog"



$$V = \begin{bmatrix} 12 & 7 & 21 & 31 & 11 \\ 45 & -2 & 14 & 27 & 19 \\ -3 & 15 & 36 & 71 & 26 \\ 4 & -13 & 55 & 34 & 15 \end{bmatrix}$$

Data structures

	homogeneous	heterogeneous
1d	(atomic) vector	list
2d	matrix	data.frame
nd	array	

The coordinates on a surface

The telephone numbers of the lab members

The vegetation surveys in la Crau, France

The vegetation surveys in la Crau, France with a species trait database

	homogeneous	heterogeneous
1d	<small>(atomic)</small> vector	list
2d	matrix	data.frame
nd	array	

The time-series of an ODE model of 100 species, for 1000 simulations

	homogeneous	heterogeneous
1d	<small>(atomic) </small> vector	list
2d	matrix	data.frame
nd	array	

The phylogenetic tree of alpine plants

	homogeneous	heterogeneous
1d	<small>(atomic)</small> vector	list
2d	matrix	data.frame
nd	array	

Data structures

	homogeneous	heterogeneous
1d	(atomic) vector	list
2d	matrix	data.frame
nd	array	

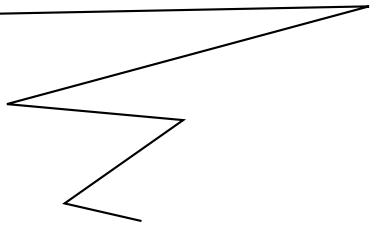
Data structures

	homogeneous	heterogeneous
1d	(atomic) vector []	list [[]] \$ []
2d	matrix [,]	data.frame [,] \$ [[]]
nd	array [, ,]	

not recommended

The diagram consists of three curved arrows originating from the text "not recommended" located at the bottom right of the slide. One arrow points from "not recommended" to the double brackets "[[]]" in the "data.frame" row of the table. Another arrow points from "not recommended" to the dollar sign "\$" in the same row. A third arrow points from "not recommended" to the triple brackets "[[[]]]" in the same row.

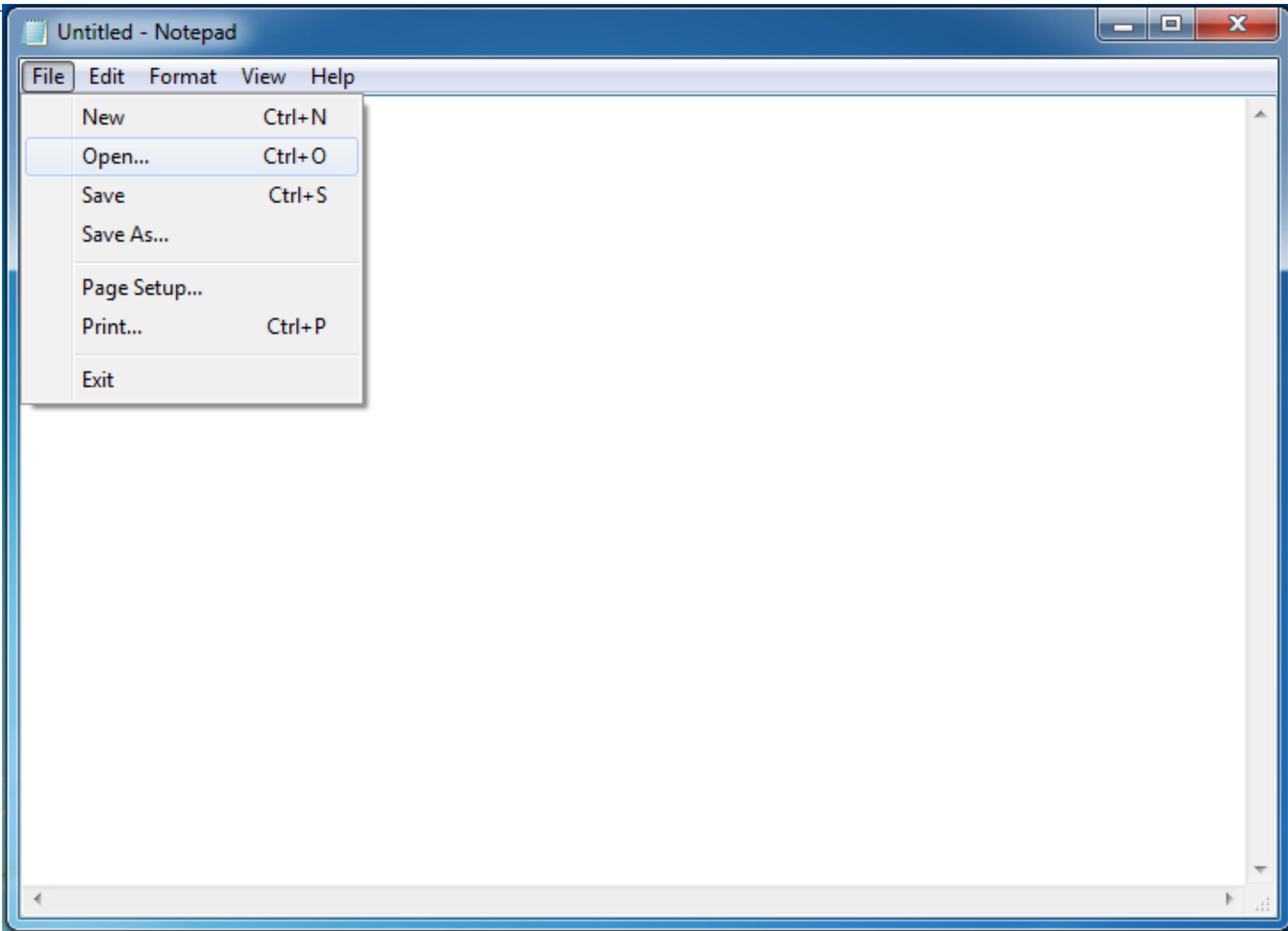
Write readable code



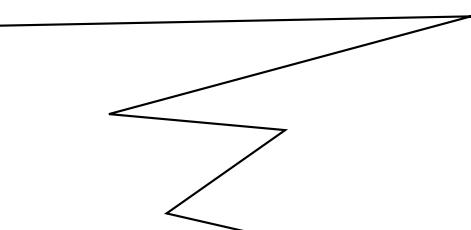
A good keyboard for programming

A good text editor

Write readable code



Write readable code



```
F:\Project Invincible\Project Invincible\core\OnRender.cpp - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
OnInit.cpp OnRender.cpp sdl_user_interface.cpp
1 #include "sdl_user_interface.h"
2
3 void markSquare(int x,
4     int y,
5     GLfloat red,
6     GLfloat green,
7     GLfloat blue,
8     GLfloat alpha,
9     GLfloat zIndex)
10 {
11     glBegin(GL_QUADS);
12         glColor4f(red, green, blue, alpha);
13         glNormal3i(0, 1, 0);
14         x = 2*x;
15         int z = -2*y;
16         glVertex3f((GLfloat)x-1, zIndex, (GLfloat)z+1);
17         glVertex3f((GLfloat)x+1, zIndex, (GLfloat)z+1);
18         glVertex3f((GLfloat)x+1, zIndex, (GLfloat)z-1);
19         glVertex3f((GLfloat)x-1, zIndex, (GLfloat)z-1);
    ...
}
C++ source file length: 5272 lines: 237 Ln:1 Col:1 Sel:0 Dos\Windows ANSI INS
```

Welcome to readable code

FOLDERS

marasys-cscart-beta

- .idea
- addons
- controllers
- core
- images
- install
- js
- lib
- payments
- schemas
- shippings
- ▼ skins
 - ▼ basic
 - admin
 - ▼ customer
 - addons
 - blocks
 - buttons
 - common_templates
 - css
 - images
 - node_modules
 - pickers
 - views
 - base.css
 - base.styl
 - demo_skin_selector.tpl
 - design_mode.css
 - dropdown.css
 - dropdown.styl
 - exception.tpl
 - index.tpl
 - meta.tpl
 - print.css
 - styles.css
 - styles.styl
 - styles_ie.css
 - top_quick_links.tpl
 - variables.css
 - variables.styl

```
1 @import 'css/nib'
2 @import 'variables'
3
4 ::selection
5   background lighten(marablue, 80%)
6   color darken(color-text, 80%)
7
8 // FONTS
9
10 font-yanone = 'Yanone Kaffeesatz', 'sans-serif'
11 font-cabin = 'Cabin Condensed', 'sans-serif'
12 font-ropa = 'Ropa Sans', 'sans-serif'
13 font-dosis = 'Dosis', 'sans-serif'
14
15 // general styles
16
17 body, h1, h2, h3, h4, h5, h6
18   font normal 13px Arial,Tahoma,Helvetica,sans-serif
19
20 body, div, p
21   color #333
22
23 .input-text, .input-text-auto, .input-text-large, .input-text-medium, .input-text-short, .input-text-100, .input-text-200, .input-text-300, .input-text-400, .input-text-500, .input-text-600, .input-text-700, .input-text-800, .input-text-900, .input-text-1000, .input-text-1100, .input-text-1200, .input-text-1300, .input-text-1400, .input-text-1500, .input-text-1600, .input-text-1700, .input-text-1800, .input-text-1900, .input-text-2000, .input-text-2100, .input-text-2200, .input-text-2300, .input-text-2400, .input-text-2500, .input-text-2600, .input-text-2700, .input-text-2800, .input-text-2900, .input-text-3000, .input-text-3100, .input-text-3200, .input-text-3300, .input-text-3400, .input-text-3500, .input-text-3600, .input-text-3700, .input-text-3800, .input-text-3900, .input-text-4000, .input-text-4100, .input-text-4200, .input-text-4300, .input-text-4400, .input-text-4500, .input-text-4600, .input-text-4700, .input-text-4800, .input-text-4900, .input-text-5000, .input-text-5100, .input-text-5200, .input-text-5300, .input-text-5400, .input-text-5500, .input-text-5600, .input-text-5700, .input-text-5800, .input-text-5900, .input-text-6000, .input-text-6100, .input-text-6200, .input-text-6300, .input-text-6400, .input-text-6500, .input-text-6600, .input-text-6700, .input-text-6800, .input-text-6900, .input-text-7000, .input-text-7100, .input-text-7200, .input-text-7300, .input-text-7400, .input-text-7500, .input-text-7600, .input-text-7700, .input-text-7800, .input-text-7900, .input-text-8000, .input-text-8100, .input-text-8200, .input-text-8300, .input-text-8400, .input-text-8500, .input-text-8600, .input-text-8700, .input-text-8800, .input-text-8900, .input-text-9000, .input-text-9100, .input-text-9200, .input-text-9300, .input-text-9400, .input-text-9500, .input-text-9600, .input-text-9700, .input-text-9800, .input-text-9900, .input-text-10000, .input-text-11000, .input-text-12000, .input-text-13000, .input-text-14000, .input-text-15000, .input-text-16000, .input-text-17000, .input-text-18000, .input-text-19000, .input-text-20000, .input-text-21000, .input-text-22000, .input-text-23000, .input-text-24000, .input-text-25000, .input-text-26000, .input-text-27000, .input-text-28000, .input-text-29000, .input-text-30000, .input-text-31000, .input-text-32000, .input-text-33000, .input-text-34000, .input-text-35000, .input-text-36000, .input-text-37000, .input-text-38000, .input-text-39000, .input-text-40000, .input-text-41000, .input-text-42000, .input-text-43000, .input-text-44000, .input-text-45000, .input-text-46000, .input-text-47000, .input-text-48000, .input-text-49000, .input-text-50000, .input-text-51000, .input-text-52000, .input-text-53000, .input-text-54000, .input-text-55000, .input-text-56000, .input-text-57000, .input-text-58000, .input-text-59000, .input-text-60000, .input-text-61000, .input-text-62000, .input-text-63000, .input-text-64000, .input-text-65000, .input-text-66000, .input-text-67000, .input-text-68000, .input-text-69000, .input-text-70000, .input-text-71000, .input-text-72000, .input-text-73000, .input-text-74000, .input-text-75000, .input-text-76000, .input-text-77000, .input-text-78000, .input-text-79000, .input-text-80000, .input-text-81000, .input-text-82000, .input-text-83000, .input-text-84000, .input-text-85000, .input-text-86000, .input-text-87000, .input-text-88000, .input-text-89000, .input-text-90000, .input-text-91000, .input-text-92000, .input-text-93000, .input-text-94000, .input-text-95000, .input-text-96000, .input-text-97000, .input-text-98000, .input-text-99000, .input-text-100000, .input-text-110000, .input-text-120000, .input-text-130000, .input-text-140000, .input-text-150000, .input-text-160000, .input-text-170000, .input-text-180000, .input-text-190000, .input-text-200000, .input-text-210000, .input-text-220000, .input-text-230000, .input-text-240000, .input-text-250000, .input-text-260000, .input-text-270000, .input-text-280000, .input-text-290000, .input-text-300000, .input-text-310000, .input-text-320000, .input-text-330000, .input-text-340000, .input-text-350000, .input-text-360000, .input-text-370000, .input-text-380000, .input-text-390000, .input-text-400000, .input-text-410000, .input-text-420000, .input-text-430000, .input-text-440000, .input-text-450000, .input-text-460000, .input-text-470000, .input-text-480000, .input-text-490000, .input-text-500000, .input-text-510000, .input-text-520000, .input-text-530000, .input-text-540000, .input-text-550000, .input-text-560000, .input-text-570000, .input-text-580000, .input-text-590000, .input-text-600000, .input-text-610000, .input-text-620000, .input-text-630000, .input-text-640000, .input-text-650000, .input-text-660000, .input-text-670000, .input-text-680000, .input-text-690000, .input-text-700000, .input-text-710000, .input-text-720000, .input-text-730000, .input-text-740000, .input-text-750000, .input-text-760000, .input-text-770000, .input-text-780000, .input-text-790000, .input-text-800000, .input-text-810000, .input-text-820000, .input-text-830000, .input-text-840000, .input-text-850000, .input-text-860000, .input-text-870000, .input-text-880000, .input-text-890000, .input-text-900000, .input-text-910000, .input-text-920000, .input-text-930000, .input-text-940000, .input-text-950000, .input-text-960000, .input-text-970000, .input-text-980000, .input-text-990000, .input-text-1000000, .input-text-1100000, .input-text-1200000, .input-text-1300000, .input-text-1400000, .input-text-1500000, .input-text-1600000, .input-text-1700000, .input-text-1800000, .input-text-1900000, .input-text-2000000, .input-text-2100000, .input-text-2200000, .input-text-2300000, .input-text-2400000, .input-text-2500000, .input-text-2600000, .input-text-2700000, .input-text-2800000, .input-text-2900000, .input-text-3000000, .input-text-3100000, .input-text-3200000, .input-text-3300000, .input-text-3400000, .input-text-3500000, .input-text-3600000, .input-text-3700000, .input-text-3800000, .input-text-3900000, .input-text-4000000, .input-text-4100000, .input-text-4200000, .input-text-4300000, .input-text-4400000, .input-text-4500000, .input-text-4600000, .input-text-4700000, .input-text-4800000, .input-text-4900000, .input-text-5000000, .input-text-5100000, .input-text-5200000, .input-text-5300000, .input-text-5400000, .input-text-5500000, .input-text-5600000, .input-text-5700000, .input-text-5800000, .input-text-5900000, .input-text-6000000, .input-text-6100000, .input-text-6200000, .input-text-6300000, .input-text-6400000, .input-text-6500000, .input-text-6600000, .input-text-6700000, .input-text-6800000, .input-text-6900000, .input-text-7000000, .input-text-7100000, .input-text-7200000, .input-text-7300000, .input-text-7400000, .input-text-7500000, .input-text-7600000, .input-text-7700000, .input-text-7800000, .input-text-7900000, .input-text-8000000, .input-text-8100000, .input-text-8200000, .input-text-8300000, .input-text-8400000, .input-text-8500000, .input-text-8600000, .input-text-8700000, .input-text-8800000, .input-text-8900000, .input-text-9000000, .input-text-9100000, .input-text-9200000, .input-text-9300000, .input-text-9400000, .input-text-9500000, .input-text-9600000, .input-text-9700000, .input-text-9800000, .input-text-9900000, .input-text-10000000, .input-text-11000000, .input-text-12000000, .input-text-13000000, .input-text-14000000, .input-text-15000000, .input-text-16000000, .input-text-17000000, .input-text-18000000, .input-text-19000000, .input-text-20000000, .input-text-21000000, .input-text-22000000, .input-text-23000000, .input-text-24000000, .input-text-25000000, .input-text-26000000, .input-text-27000000, .input-text-28000000, .input-text-29000000, .input-text-30000000, .input-text-31000000, .input-text-32000000, .input-text-33000000, .input-text-34000000, .input-text-35000000, .input-text-36000000, .input-text-37000000, .input-text-38000000, .input-text-39000000, .input-text-40000000, .input-text-41000000, .input-text-42000000, .input-text-43000000, .input-text-44000000, .input-text-45000000, .input-text-46000000, .input-text-47000000, .input-text-48000000, .input-text-49000000, .input-text-50000000, .input-text-51000000, .input-text-52000000, .input-text-53000000, .input-text-54000000, .input-text-55000000, .input-text-56000000, .input-text-57000000, .input-text-58000000, .input-text-59000000, .input-text-60000000, .input-text-61000000, .input-text-62000000, .input-text-63000000, .input-text-64000000, .input-text-65000000, .input-text-66000000, .input-text-67000000, .input-text-68000000, .input-text-69000000, .input-text-70000000, .input-text-71000000, .input-text-72000000, .input-text-73000000, .input-text-74000000, .input-text-75000000, .input-text-76000000, .input-text-77000000, .input-text-78000000, .input-text-79000000, .input-text-80000000, .input-text-81000000, .input-text-82000000, .input-text-83000000, .input-text-84000000, .input-text-85000000, .input-text-86000000, .input-text-87000000, .input-text-88000000, .input-text-89000000, .input-text-90000000, .input-text-91000000, .input-text-92000000, .input-text-93000000, .input-text-94000000, .input-text-95000000, .input-text-96000000, .input-text-97000000, .input-text-98000000, .input-text-99000000, .input-text-100000000, .input-text-110000000, .input-text-120000000, .input-text-130000000, .input-text-140000000, .input-text-150000000, .input-text-160000000, .input-text-170000000, .input-text-180000000, .input-text-190000000, .input-text-200000000, .input-text-210000000, .input-text-220000000, .input-text-230000000, .input-text-240000000, .input-text-250000000, .input-text-260000000, .input-text-270000000, .input-text-280000000, .input-text-290000000, .input-text-300000000, .input-text-310000000, .input-text-320000000, .input-text-330000000, .input-text-340000000, .input-text-350000000, .input-text-360000000, .input-text-370000000, .input-text-380000000, .input-text-390000000, .input-text-400000000, .input-text-410000000, .input-text-420000000, .input-text-430000000, .input-text-440000000, .input-text-450000000, .input-text-460000000, .input-text-470000000, .input-text-480000000, .input-text-490000000, .input-text-500000000, .input-text-510000000, .input-text-520000000, .input-text-530000000, .input-text-540000000, .input-text-550000000, .input-text-560000000, .input-text-570000000, .input-text-580000000, .input-text-590000000, .input-text-600000000, .input-text-610000000, .input-text-620000000, .input-text-630000000, .input-text-640000000, .input-text-650000000, .input-text-660000000, .input-text-670000000, .input-text-680000000, .input-text-690000000, .input-text-700000000, .input-text-710000000, .input-text-720000000, .input-text-730000000, .input-text-740000000, .input-text-750000000, .input-text-760000000, .input-text-770000000, .input-text-780000000, .input-text-790000000, .input-text-800000000, .input-text-810000000, .input-text-820000000, .input-text-830000000, .input-text-840000000, .input-text-850000000, .input-text-860000000, .input-text-870000000, .input-text-880000000, .input-text-890000000, .input-text-900000000, .input-text-910000000, .input-text-920000000, .input-text-930000000, .input-text-940000000, .input-text-950000000, .input-text-960000000, .input-text-970000000, .input-text-980000000, .input-text-990000000, .input-text-1000000000, .input-text-1100000000, .input-text-1200000000, .input-text-1300000000, .input-text-1400000000, .input-text-1500000000, .input-text-1600000000, .input-text-1700000000, .input-text-1800000000, .input-text-1900000000, .input-text-2000000000, .input-text-2100000000, .input-text-2200000000, .input-text-2300000000, .input-text-2400000000, .input-text-2500000000, .input-text-2600000000, .input-text-2700000000, .input-text-2800000000, .input-text-2900000000, .input-text-3000000000, .input-text-3100000000, .input-text-3200000000, .input-text-3300000000, .input-text-3400000000, .input-text-3500000000, .input-text-3600000000, .input-text-3700000000, .input-text-3800000000, .input-text-3900000000, .input-text-4000000000, .input-text-4100000000, .input-text-4200000000, .input-text-4300000000, .input-text-4400000000, .input-text-4500000000, .input-text-4600000000, .input-text-4700000000, .input-text-4800000000, .input-text-4900000000, .input-text-5000000000, .input-text-5100000000, .input-text-5200000000, .input-text-5300000000, .input-text-5400000000, .input-text-5500000000, .input-text-5600000000, .input-text-5700000000, .input-text-5800000000, .input-text-5900000000, .input-text-6000000000, .input-text-6100000000, .input-text-6200000000, .input-text-63000
```

```

..  

+ cache/  

+ compile/  

+ interface/  

+ ivy/  

+ launch/  

+ licenses/  

+ main/  

+ project/  

+ run/  

+ sbt/  

+ scripted/  

+ src/  

+ target/  

+ tasks/  

+ testing/  

+ util/  

CONTRIBUTING.md  

LICENSE  

NOTICE  

README.md  

api.specification  

scripted.specification

```

```

62 > def groupBy[K](discriminator: (File => K)): Map[K, Analysis]  

63  

64 > override lazy val toString = Analysis.summary(this)  

65 }  

66  

67 object Analysis  

68 {  

69 > lazy val Empty: Analysis = new MAnalysis(Stamps.empty, APIs.empty, Relations.empty, SourceInfo  

s.empty, Compilations.empty)  

70 > private[sbt] def empty(nameHashing: Boolean): Analysis = new MAnalysis(Stamps.empty, APIs.empt  

y,  

71 > > Relations.empty(nameHashing = nameHashing), SourceInfos.empty, Compilations.empty)  

72  

73 > /** Merge multiple analysis objects into one. Deps will be internalized as needed. */  

74 > def merge(analyses: Traversable[Analysis]): Analysis = {  

75 > > if (analyses.exists(_.relations.nameHashing))  

76 > > > throw new IllegalArgumentException("Merging of Analyses that have" +  

77 > > > " `relations.memberRefAndInheritanceDeps` set to `true` is not supported.")  

78  

79 > > // Merge the Relations, internalizing deps as needed.  

80 > > val mergedSrcProd = Relation.merge(analyses map { _.relations.srcProd })  

81 > > val mergedBinaryDep = Relation.merge(analyses map { _.relations.binaryDep })  

82 > > val mergedClasses = Relation.merge(analyses map { _.relations.classes })  

83  

84 > > val stillInternal = Relation.merge(analyses map { _.relations.direct.internal })  

85 > > val (internalized, stillExternal) = Relation.merge(analyses map { _.relations.direct.external  

l }) partition { case (a, b) => mergedClasses._2s.contains(b) }  

86 > > val internalizedFiles = Relation.reconstruct(internalized.forwardMap mapValues { _ flatMap m  

ergedClasses.reverse })  

87 > > val mergedInternal = stillInternal ++ internalizedFiles  

88  

89 > > val stillInternalPI = Relation.merge(analyses map { _.relations.publicInherited.internal })  

90 > > val (internalizedPI, stillExternalPI) = Relation.merge(analyses map { _.relations.publicInhe  

rited.external }) partition { case (a, b) => mergedClasses._2s.contains(b) }  

91 > > val internalizedFilesPI = Relation.reconstruct(internalizedPI.forwardMap mapValues { _ flatM  

ap mergedClasses.reverse })  

92 > > val mergedInternalPI = stillInternalPI ++ internalizedFilesPI  

93  

94 > > val mergedRelations = Relations.make(  

95 > > > mergedSrcProd,  

96 > > > mergedBinaryDep,  

97 > > > Relations.makeSource(mergedInternal, stillExternal),  

98 > > > Relations.makeSource(mergedInternalPI, stillExternalPI),  

99 > > > mergedClasses

```

```

887 [info] Resolving o  

888 [info] Resolving o  

889 [info] Resolving o  

890 [info] Resolving o  

891 [info] Resolving o  

892 [info] Resolving o  

893 [info] Resolving o  

894 [info] Resolving o  

895 [info] Resolving o  

896 [info] Resolving o  

897 [info] Resolving o  

898 [info] Resolving o  

899 [info] Resolving o  

900 [info] Resolving o  

901 [info] Resolving o  

902 [info] Resolving o  

903 [info] Resolving o  

904 [info] Resolving o  

905 [info] Resolving o  

906 [info] Resolving o  

907 [info] Resolving o  

908 [info] Resolving o  

909 [info] Resolving o  

910 [info] Resolving o  

911 [info] Resolving o  

912 [info] Resolving o  

913 [info] Resolving o  

914 [info] Resolving o  

915 [info] Resolving o  

916 [info] Resolving o  

917 [info] Resolving o  

918 [info] Resolving o  

919 [info] Done updatin  

920 [success] Total ti  

root> ~compile  

922 [success] Total ti  

923 1. Waiting for sou  

924  

925 root> project main  

926 [info] Set current  

927 Main> ~compile  

928 [success] Total ti  

929 1. Waiting for sou  

930

```

Write readable code

The screenshot shows the RStudio interface with the following components:

- Top Bar:** RStudio
- File Bar:** diamondPricing.R, diamonds
- Code Editor:** A script editor containing R code for loading ggplot2, summarizing diamonds, calculating average size, defining clarity levels, and creating a qplot. The code is well-formatted with proper indentation.
- Console:** Displays summary statistics for diamonds and the resulting qplot command.
- Environment:** Shows the diamonds dataset has 53940 observations and 10 variables, with avgSize at 0.7979 and clarity levels I1, SI2, SI1, VS2, VS1, VVS2, VVS1, and IF.
- Plots:** A scatter plot titled "Diamond Pricing" showing Price vs. Carat. The points are colored by clarity: I1 (red), SI2 (orange), SI1 (green), VS2 (cyan), VS1 (blue), VVS2 (yellow), VVS1 (purple), and IF (pink).
- Text Overlay:** The word "IDE" is written vertically in large black letters on the right side of the plot area.

Write readable code

The screenshot shows a code editor with two panes. The left pane displays R code for the `build_grid_identical` function, which generates a regular grid of points. The right pane displays the corresponding C++ implementation in `inhull_cpp.cpp`. Below the editor, a terminal window shows the results of running the code, including error messages and test results.

```
#' @title Build a regular grid
#'
#' @description Create a regular grid with the same number of points on each
#' dimension.
#'
#' @param coords A matrix or data frame of coordinates as columns
#' @param npts The approximate total number of points of the output grid
#' @param pad Padding on each dimension (a positive number makes a grid
#'           that is larger than the ranges of the coordinates).
#' @param ... other arguments are silently ignored
#'
#' @return The coordinates of a grid of points as a data frame with
#'         approximately \code{npts} rows and \code{ncol(coords)} columns. Names
#'         are transferred from the \code{coords} data frame.
#'
#' @details This function creates a grid that covers a set of points. The
#'          number of points of the output grid is the same on all dimensions.
#'          This is probably the only useful option for 1D moving-window
#'          computations.
#'
#' @family grid building functions
#'
#' @export
#'

build_grid_identical <- function(coords, npts, pad = 0, ...) {
  build_grid_check_vars(coords, npts)

  coords.ranges <- apply(coords, 2, range)
  ndims <- ncol(coords)

  # The input npts is the *total* number of points wanted so we need ^(1/ndims)
  # so we use the correct, identical amount of points on each dimension.
  length.onedim <- floor(npts^(1/ndims))
}

Error: No tests found for .
In addition: Warning message:
In library(package, lib.loc = lib.loc, character.only = TRUE, logical.return = TRUE, :
  there is no package called '.'
> library(devtools)
> test()
Loading rollply
Testing rollply
Handling of bad arguments : .....
Grid building: grid_identical : ...
Testing inahull_cpp : ...
Common uses of rollply : ...

DONE
> 
```

```
//
// This function is an alternative to alphahull::inhull designed
// for speed.
//

#include <Rcpp.h>
#include <stdlib.h>

// Define column names of $complement in hull object
#define COMP_C1 0 //
#define COMP_C2 1 //
#define COMP_R 2 // third column in complement matrix (with indexing difference)

// Relaxed floating-point equality
#define feq(x,y) ( std::fabs(x - y) < 1e-10 ? true : false)

using namespace Rcpp;

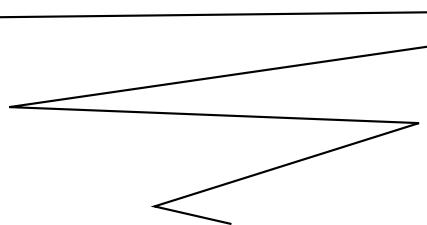
// This is a c++ implementation of alphahull::inhull
// [[Rcpp::export]]
LogicalVector inahull_cpp_multiple(List ahull_obj,
                                    NumericVector X,
                                    NumericVector Y) {

  NumericMatrix comp = as<NumericMatrix>(ahull_obj["complement"]);
  LogicalVector inhull = LogicalVector(X.length());

  // Get list of balls/halfplanes
  NumericVector halfpl = NumericVector(comp.nrow());
  NumericVector ball   = NumericVector(comp.nrow());
  int n_halfpl = 0;
  int n_ball = 0;
  for (int i=0; i<comp.nrow(); i++) {
```

Search and Replace Terminal Current Project

Write readable code



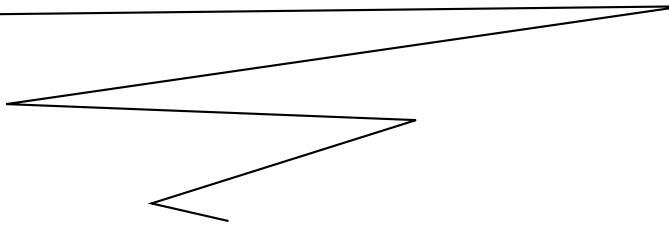
Write for people first ! (*you included*)

Write with *style* !



Google's style guide
Hadley's style guide

Write readable code



Write for people first ! (*you included*)

Write with *style* !

Do not write your whole program in a single line that takes most of my fuck

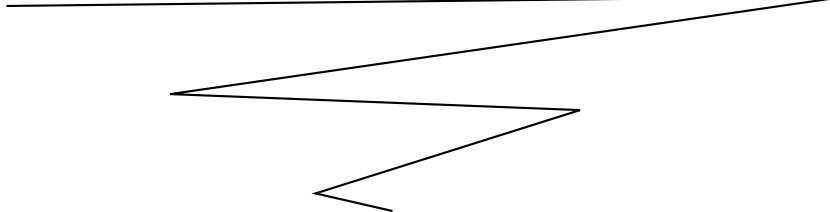
The screenshot shows the RStudio interface with the following details:

- Title Bar:** RStudio
- Toolbar:** Includes icons for file operations (New, Open, Save, Print), Go to file/function, Run, Source, and Environment.
- Project Explorer:** Shows files diamondPricing.R and diamonds.
- Code Editor:** Displays the following R code:

```
1 library(ggplot2)
2
3 View(diamonds)
4 summary(diamonds)
5
6 summary(diamonds$price)
7 avgSize <- round(mean(diamonds$carat), 4)
8 clarity <- levels(diamonds$clarity)
9
10 q <- qplot(carat, price,
11             data=diamonds, color=clarity,
12             xlab="Carat", ylab="Price", main="Diamond Pricing")
```
- Environment Tab:** Shows the global environment with objects: Data (diamond), Values (avgSize, clarity), and Global Environment (q).
- Right Panel:** Shows the Data tab with the diamond dataset.

A hand-drawn arrow points from the text "Do not write your whole program in a single line that takes most of my fuck" towards the code editor area.

Write readable code



Write for people first ! (*you included*)

Write with *style* !

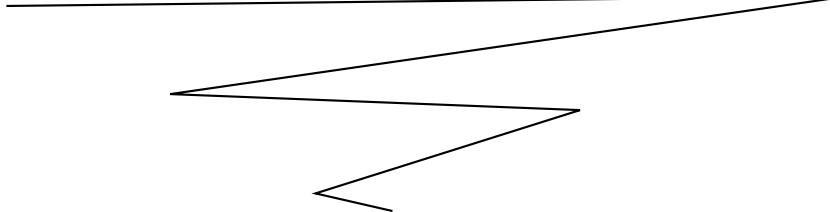
Do not write your whole program in a single line that takes most of my fuck

Comments

```
i <- i + 1 # add one to i
```

```
tmp <- head(iris) # store the first few lines of iris (a df)
```

Write readable code



Write for people first ! (*you included*)

Write with *style* !

Do not write your whole program in a single line that takes most of my fuck

Comments

```
i <- i + 1 # add one to i
```

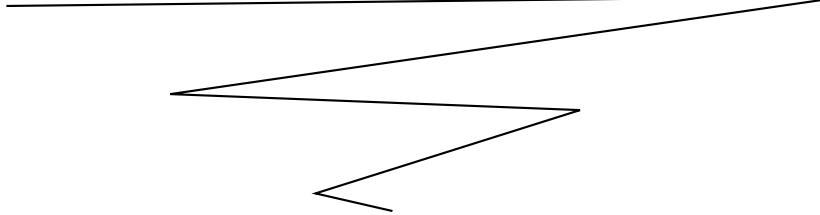
```
tmp <- head(iris) # store the first few lines of iris (a df)
```

Adapt to your public: *who is reading your code* ?

when teaching

when working with colleagues

Write readable code



```
ggplot()
```

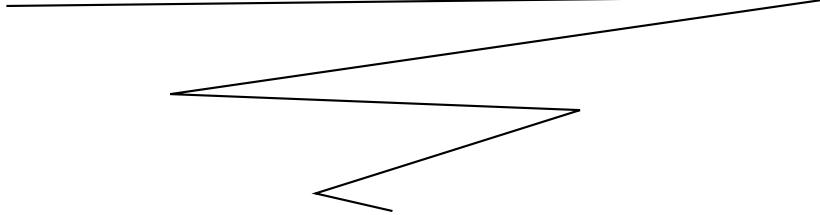
```
ode() # from pkg deSolve
```

```
%>% # from pkg magrittr
```

```
# from unknown package
f(1) <- 1
f(n) <- f(n) + f(n-1)
```

```
sim() # from package simecol
```

Write readable code



Write for people first ! (*you included*)

Write with *style* !

Do not write your whole program in a single line that takes most of my fuck

Comments

```
i <- i + 1 # add one to i
```

```
tmp <- head(iris) # store the first few lines of iris (a df)
```

Adapt to your public: *who is reading your code* ?

when teaching

when working with colleagues

Be expressive

Write readable code

Good names

Variables

One letter

i, j, k, etc.

x, y

A few letters

tmp

dat, mat

df

Functions

*Think of verbs !
Functions are actions*

add(1, to = 3)

simu_set_tstop(time = 3000)
stop_simu(at = 3000)

Many letters

plot_with_x_axis_off

df_without_site_21240

Write readable code

Good names

Variables

One letter

i, j, k, etc.

x, y

A few letters

tmp

dat, mat

df

Functions

*Think of verbs !
Functions are actions*

add(1, to = 3)

simu_set_tstop(time = 3000)
stop_simu(at = 3000)

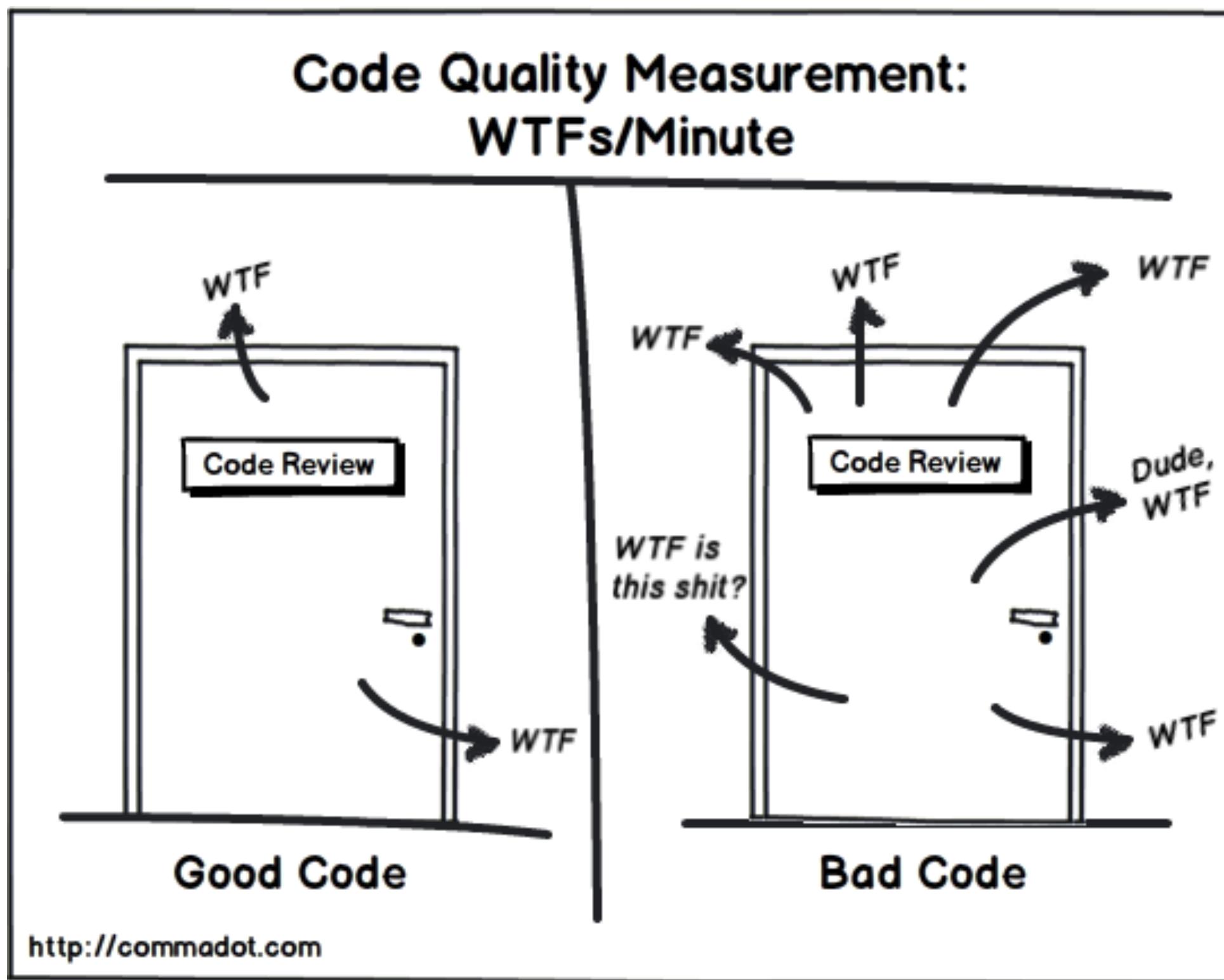
Many letters

plot_with_x_axis_off

df_without_site_21240

Write readable code

Your turn: code review



Checkpoint

- 
- Organize your workspace
 - Choose the right data structures
 - Write readable code
 - Recognize analysis patterns
 - Use the right programming paradigm

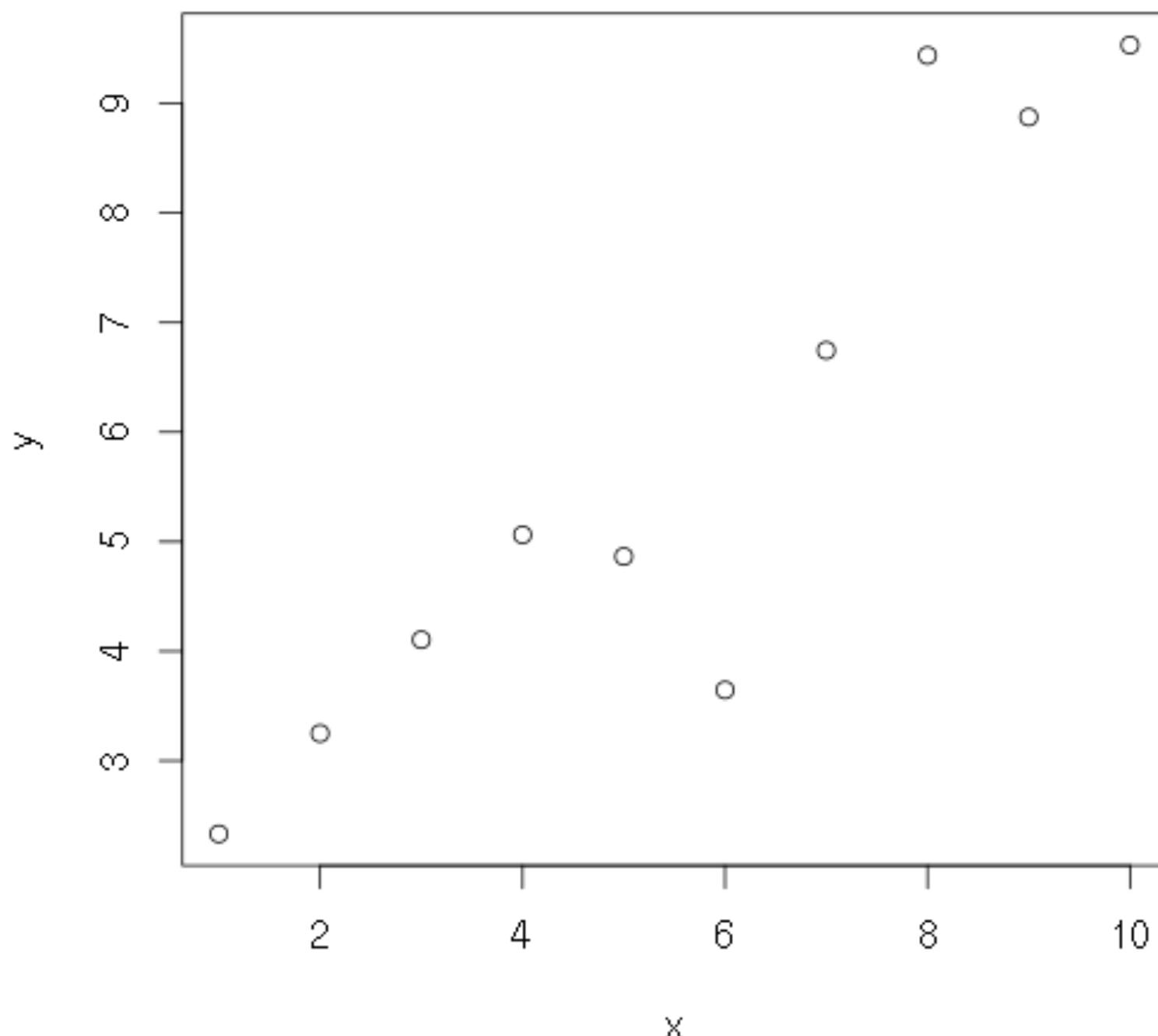
Patterns in analyses

Break it down into pieces: FUNCTIONS

Why ? DRY. Separation of concerns. Testing.

```
# Get the closest value in a vector
▼ closest_to <- function(val, x, quiet = FALSE) {
  index <- which( abs(val - x) == min(abs(val - x)) )
  new_value <- x[min(index)]
  if (!quiet) {
    cat('Picked value ', new_value, ' (error: ')
  }
  return(new_value)
}
```

```
> x <- seq.int(10)
> y <- x + rnorm(10)
> plot(x,y)
> closest_to(6, y)
Picked value 6.745802 (error: 0.745802)
[1] 6.745802
> |
```



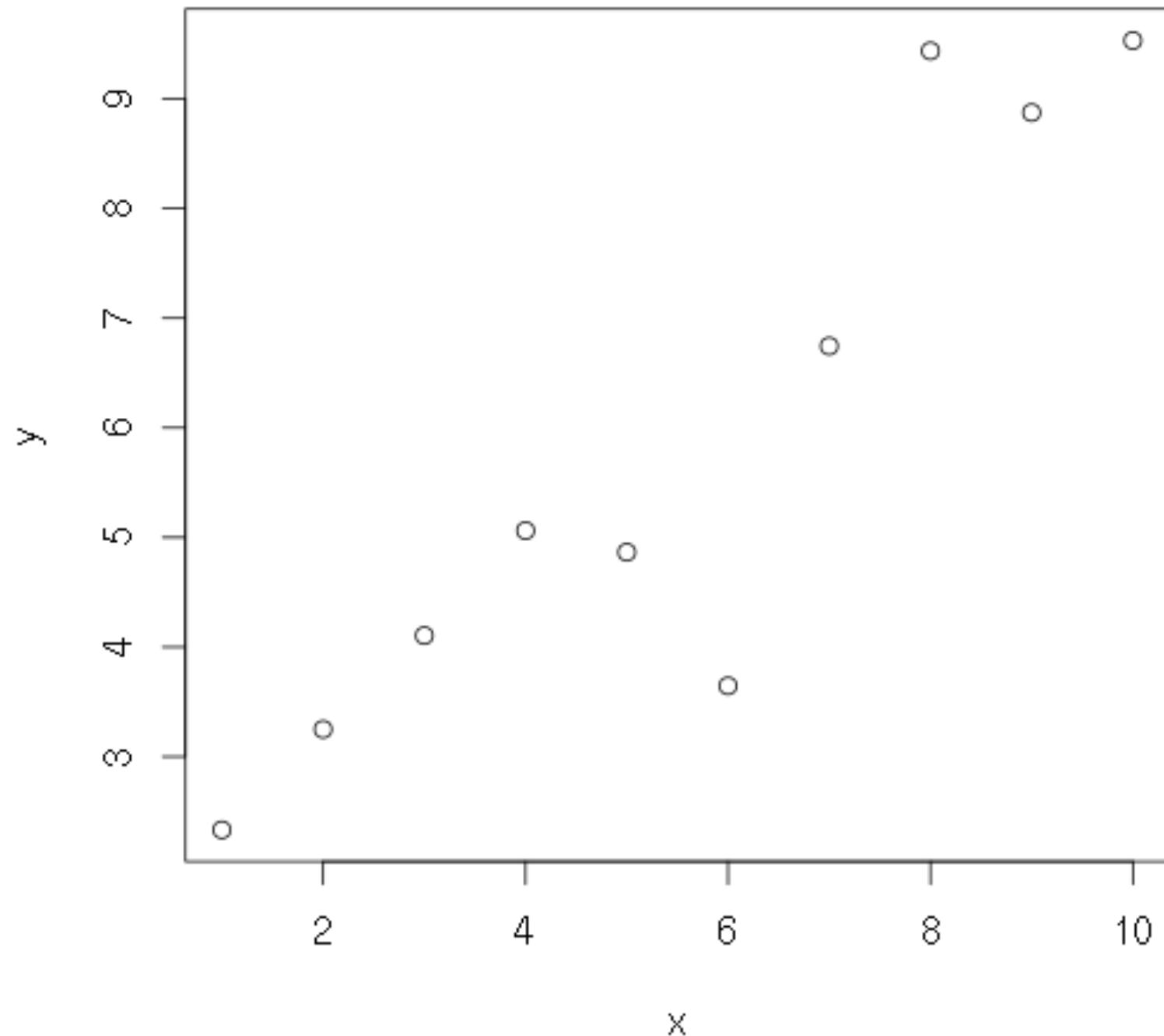
Patterns in analyses

Break it down into pieces: FUNCTIONS

Why ? DRY. Separation of concerns. T

```
# Get the closest value in a vector
closest_to <- function(val, x, quiet = FALSE) {
  index <- which( abs(val - x) == min(abs(val - x)))
  new_value <- x[min(index)]
  if (!quiet) {
    cat('Picked value ', new_value, ' (error: ', abs(val - new_value), ')')
  }
  return(new_value)
}
```

```
> x <- seq.int(10)
> y <- x + rnorm(10)
> plot(x,y)
> closest_to(6, y)
Picked value 6.745802 (error: 0.745802)
[1] 6.745802
> |
```



Patterns in analyses

```
# Print the error between an expected value and what is observed
print_error <- function(expected, observed) {
  print0('Picked value ', observed, ' (error: ', observed - expected, ')')
}

# Are the values the minima of a vector ?.
# e.g. for c(1, 2, 1), this function returns c(TRUE, FALSE, TRUE)
is_minimum <- function(vec) {
  vec == min(vec)
}

# Get the closest value in a vector
closest_to <- function(val, x, quiet = FALSE) {
  index <- which( is_minimum(abs(x-val)) )
  new_value <- x[min(index)]
  .
  if ( !quiet ) {
    print_error(val, new_value)
  }
  .
  return(new_value)
}
```

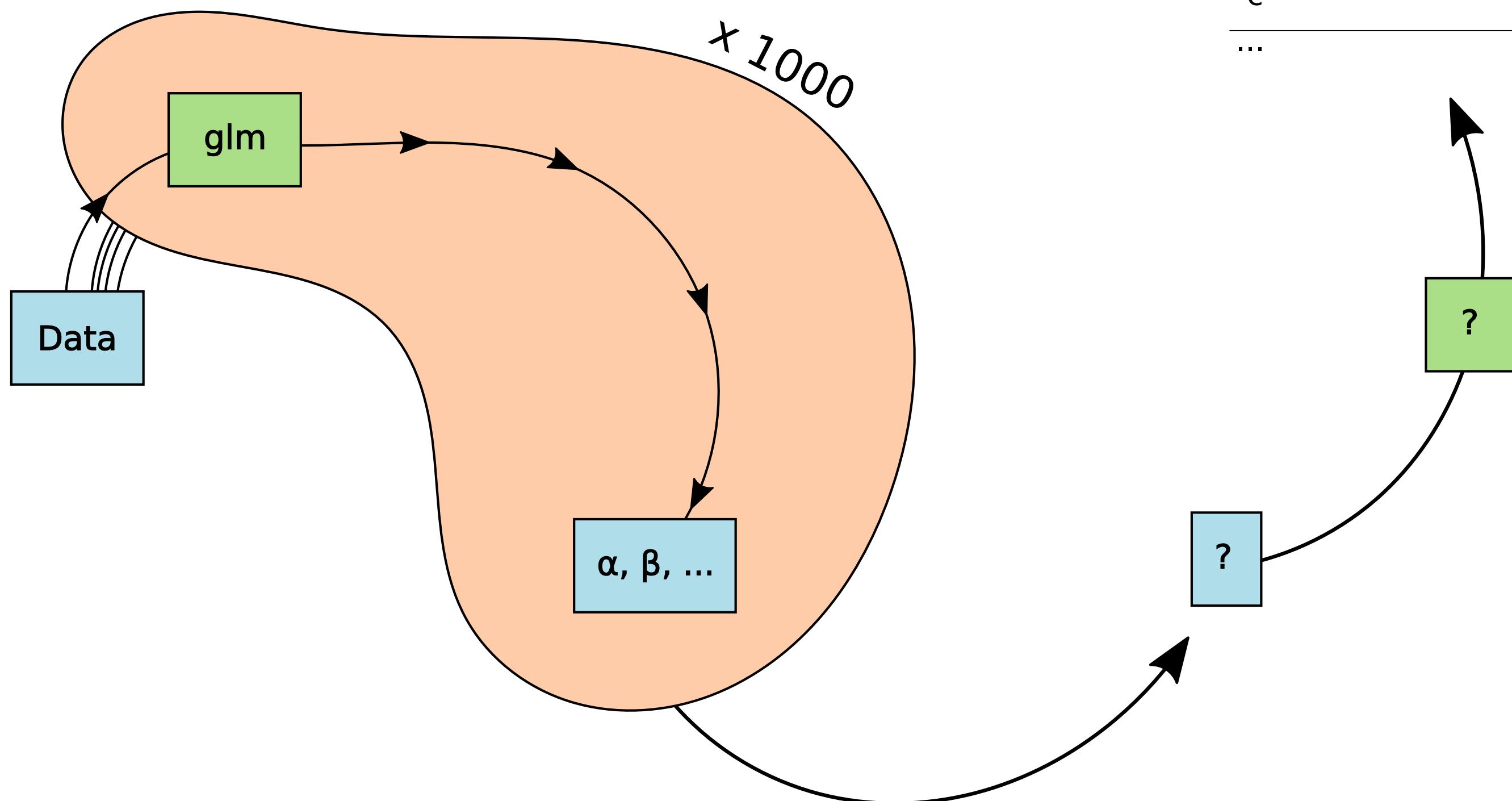
More expressive

Describe the what, not the how.
Dirty job is under the rug

Patterns in analyses

Patterns in analyses

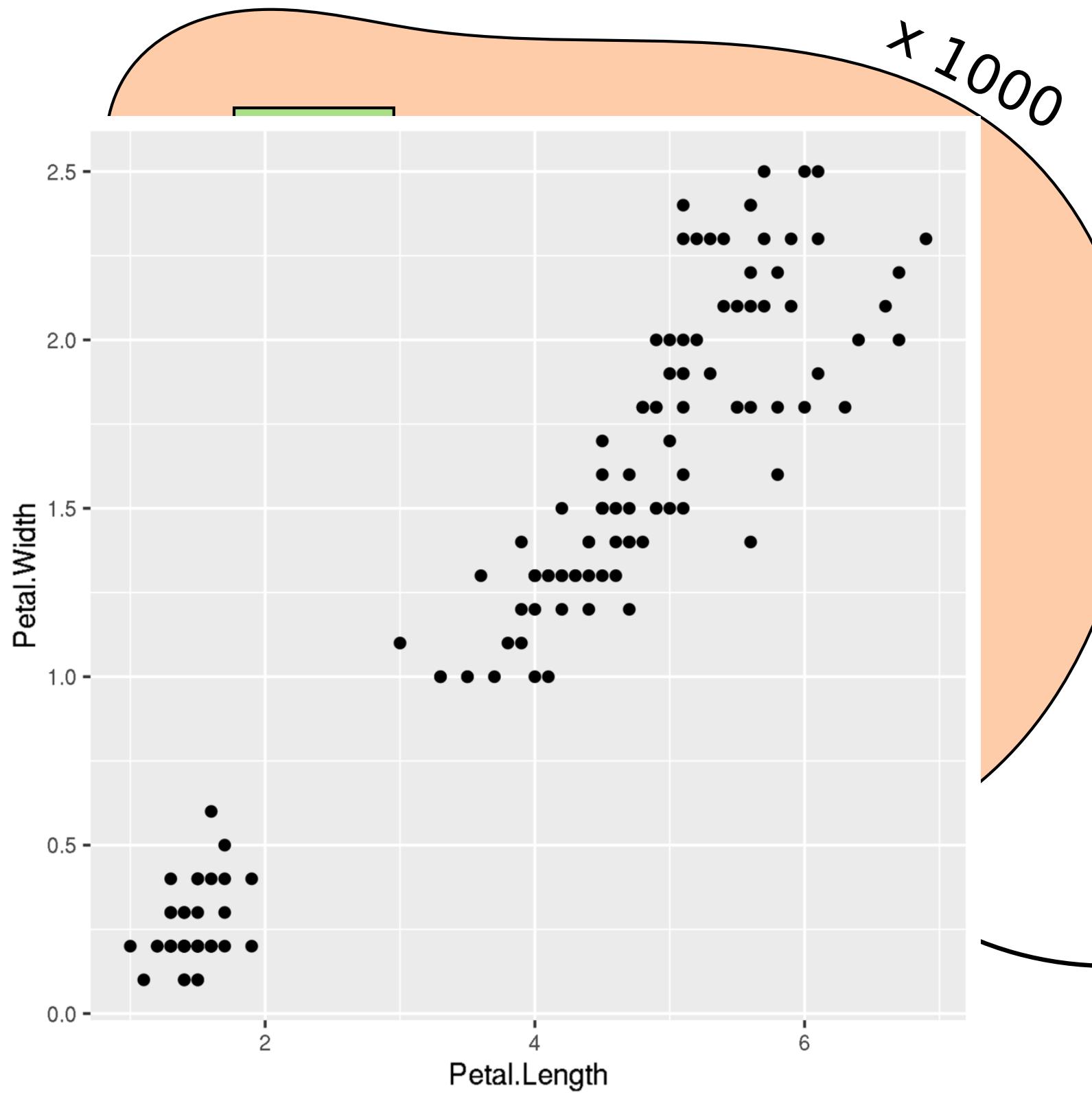
The **Repeat-combine** pattern



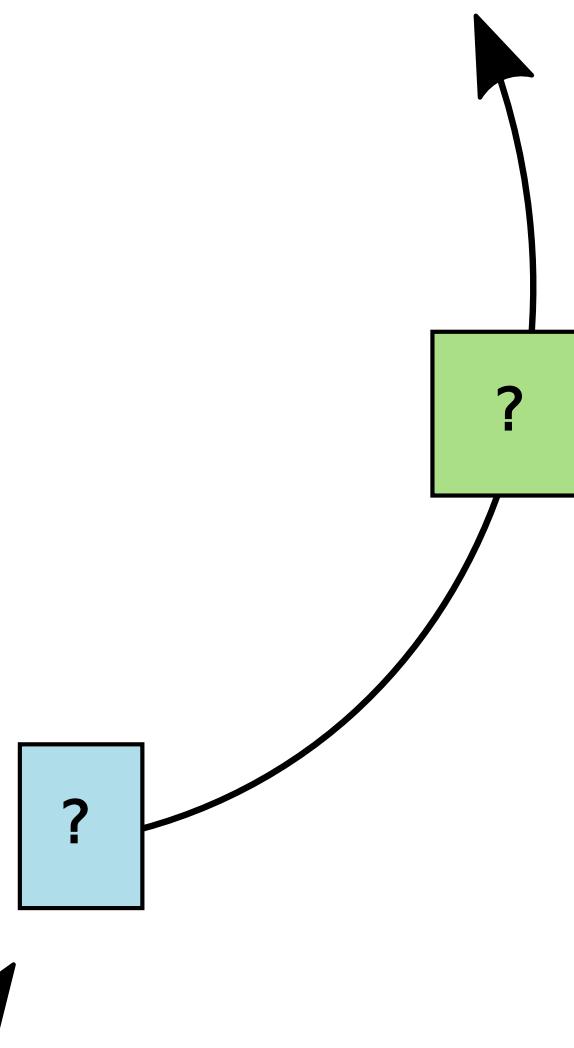
Mean	SD
a	
b	
c	
...	

Patterns in analyses

The **Repeat-combine** pattern

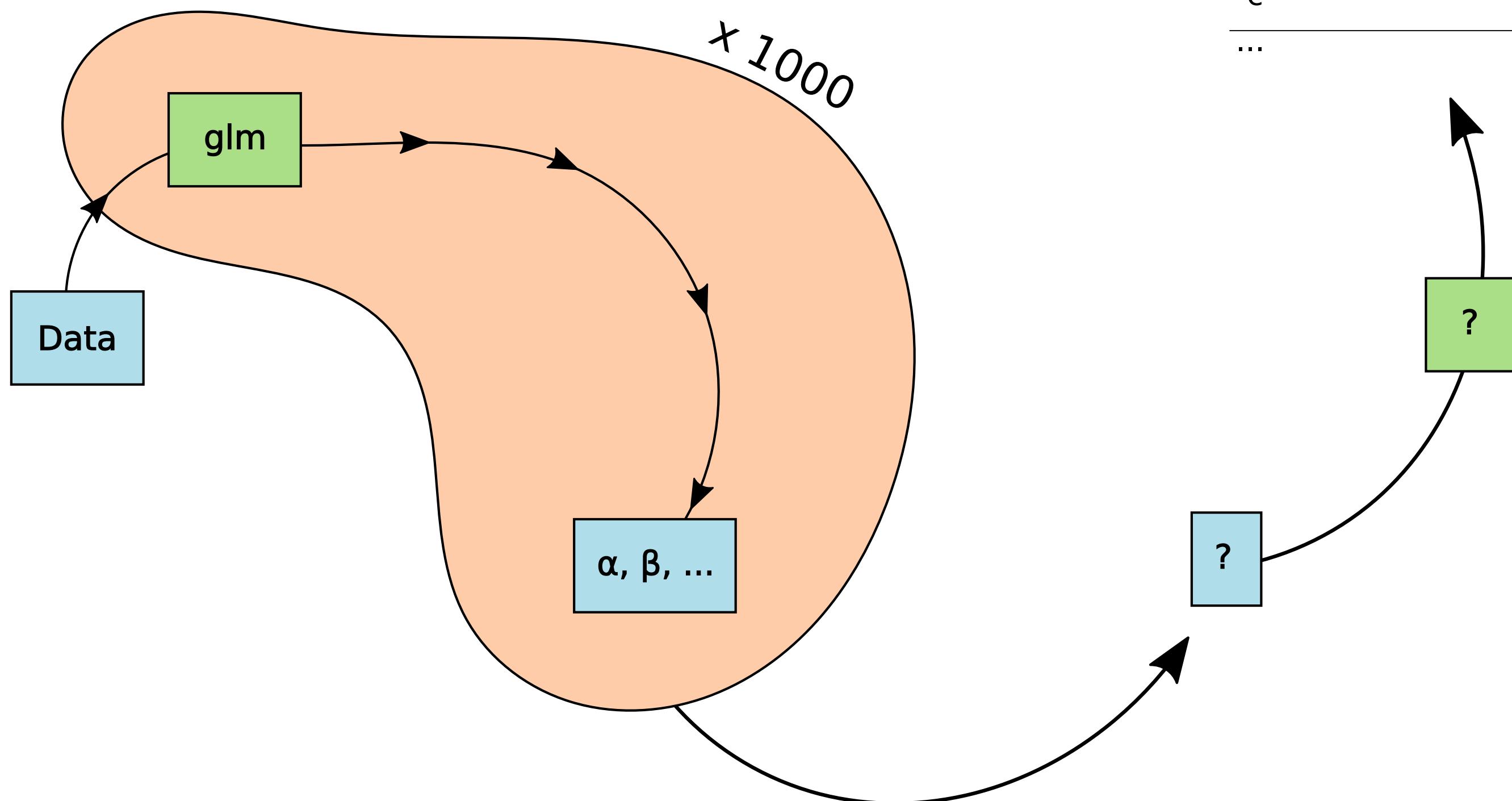


	Mean	SD
a		
b		
c		
...		



Patterns in analyses

The **Repeat-combine** pattern



Mean	SD
a	
b	
c	
...	

Patterns in analyses

The **Split - apply - combine** pattern

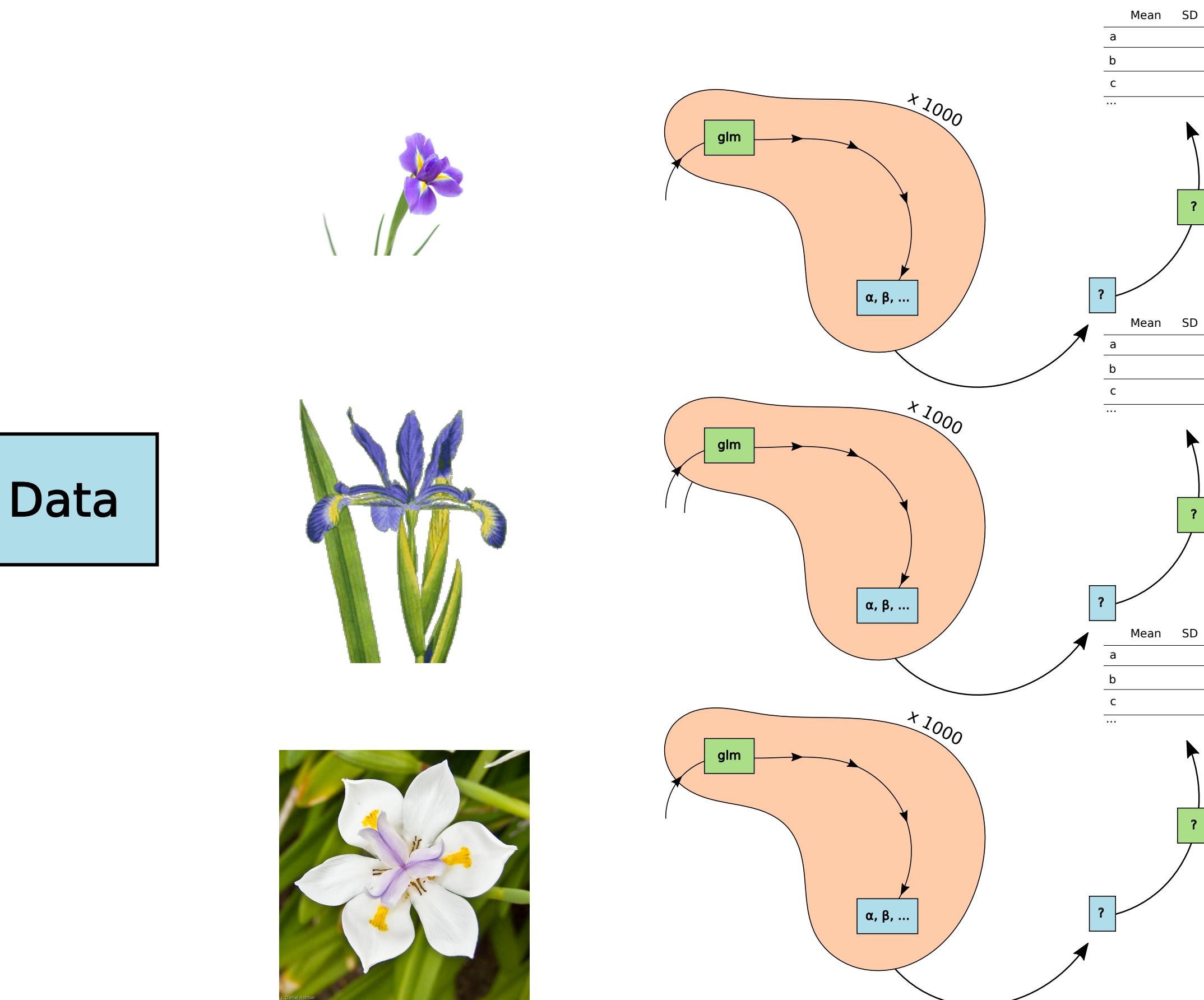


Data



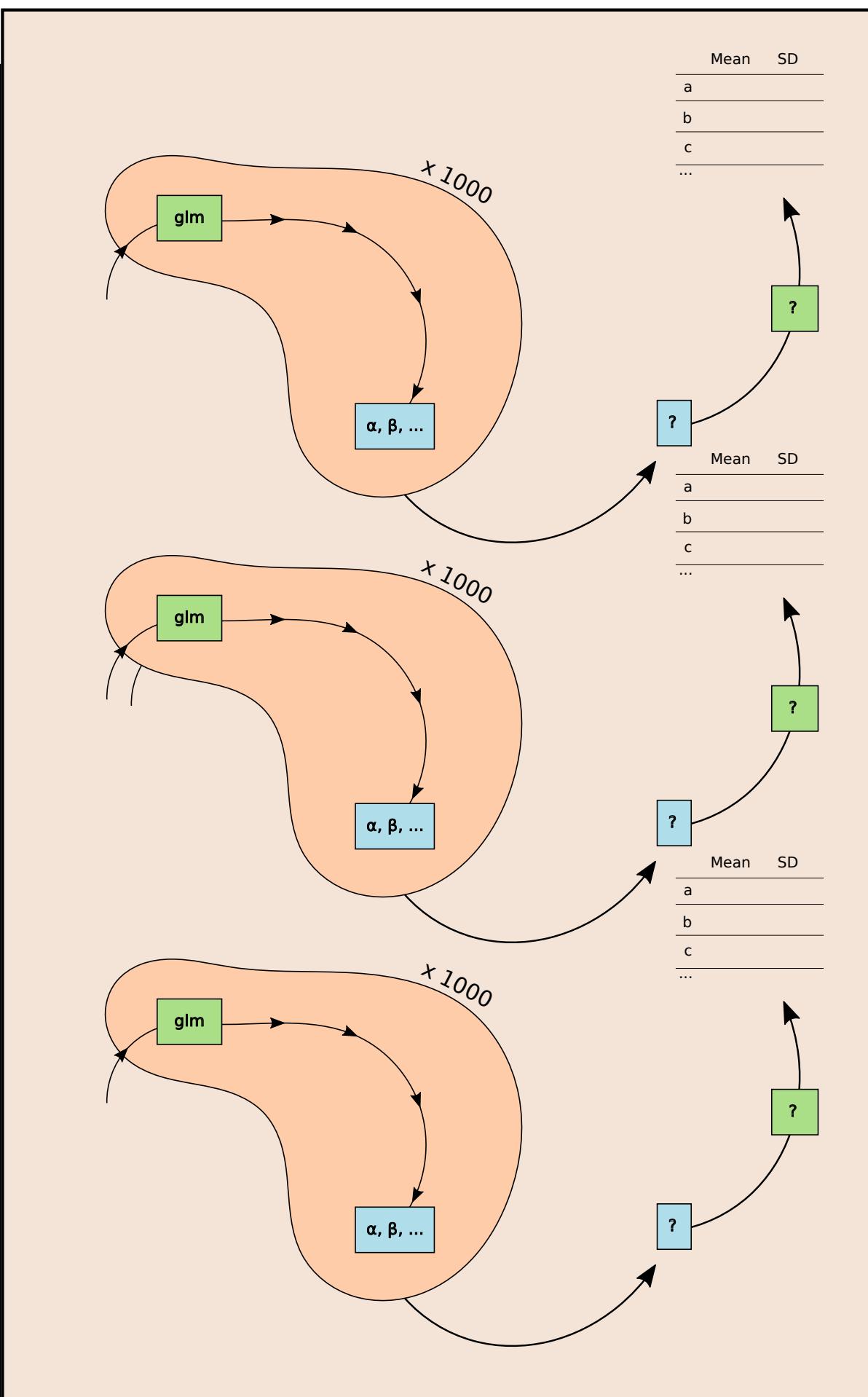
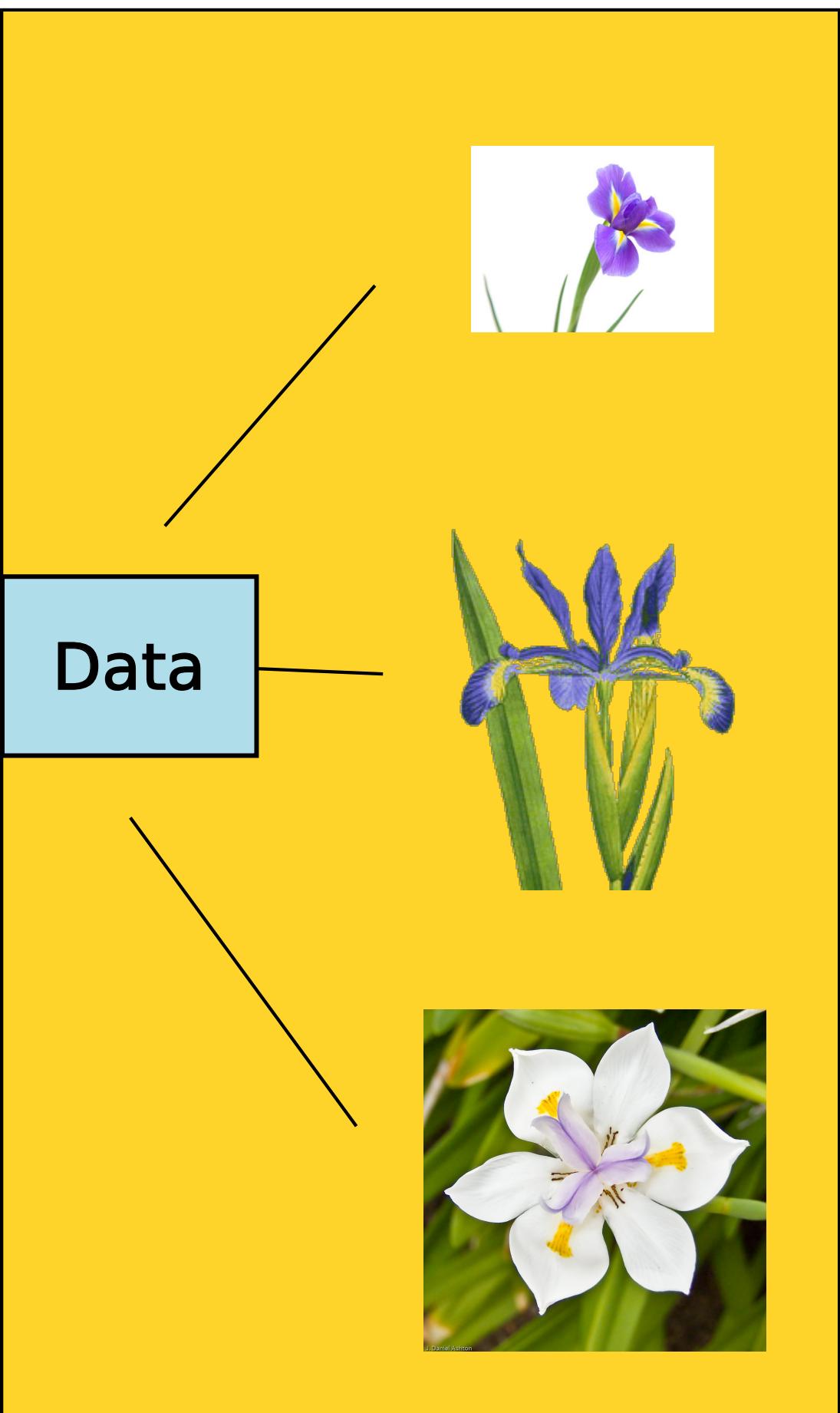
Patterns in analyses

The **Split - apply - combine** pattern



Patterns in analyses

The **Split - apply - combine** pattern



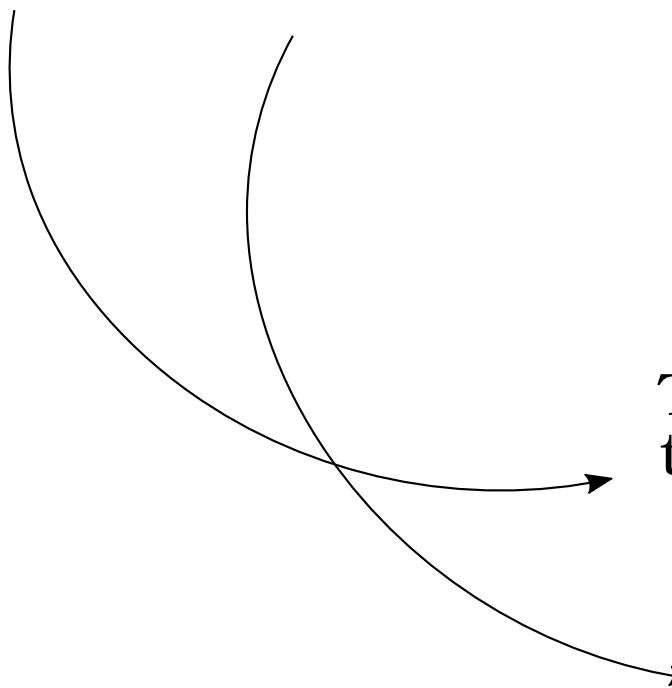
A pink background panel showing the resulting summary table. A curved arrow points from the bottom right of the diagram to the table. The table has columns "Species", "Mean", and "SD". It contains three rows: "a setosa", "b setosa", and "c virginica". Ellipses at the bottom indicate more rows.

Species	Mean	SD
a	setosa	
b	setosa	
c	virginica	
...		

Patterns in analyses

The **split-apply-combine** pattern

`apply/lapply/sapply/mapply/rapply/etc.`



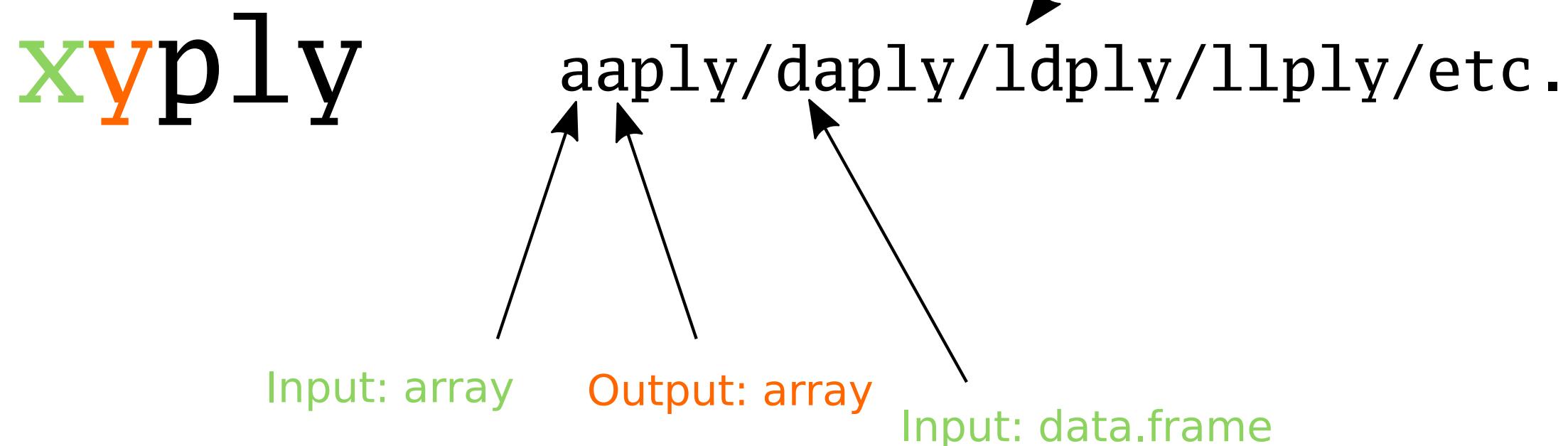
Takes a matrix, applies the function to each rows/cols and return the results as a ?

Takes a list, applies the function on each of its elements, then return the results as a ?

Patterns in analyses

The **split-apply-combine** pattern

~~apply/lapply/sapply/... plyr~~



Patterns in analyses

The **split-apply-combine** pattern

```
ddply(iris, ~ Species,  
      function(df) {  
        data.frame(Sepal.Length.mean = mean(df[ , 'Sepal.Length']))  
      })
```

```
ddply(iris, ~ Species,  
      summarize, Sepal.Length.mean = mean(Sepal.Length)))
```

Patterns in analyses

The **split-apply-combine** pattern

```
ddply(iris, ~ Species,  
      function(df) {  
        data.frame(Sepal.Length.mean = mean(df[ , 'Sepal.Length']))  
      })
```

```
ddply(iris, ~ Species,  
      summarize, Sepal.Length.mean = mean(Sepal.Length)))
```

```
dlply(iris, ~ Species,  
      function(split_df) {  
        glm(Petal.Length ~ Petal.Width, data = dsplit_df)  
      })
```

Patterns in analyses

The **split-apply-combine** pattern

Compute the petal mean and SD of each iris species (dataset: iris)

Compute the mean price of diamonds by cut and color (dataset: diamonds)

Compute the bootstraped slope/intercept estimate for each iris species

Patterns in analyses

The **split-apply-combine** pattern

```
ddply(iris, ~ Species,  
      function(df) {  
        data.frame(Sepal.Length.mean = mean(df[ , 'Sepal.Length']))  
      })
```

Patterns in analyses

Other analyses patterns ?

Checkpoint

- 
- Organize your workspace
 - Choose the right data structures
 - Write readable code
 - Recognize analysis patterns
 - Use the right programming paradigm

OOP basics: class, methods

A set of points ?

```
a <- c(06, 30, 15, 67, 12, 43)
```

```
print(a) ?
```

OOP basics: class, methods

A set of points ?

```
a <- c(06, 30, 15, 67, 12, 43)
```

```
print(a) ?
```

```
a <- c(06, 30, 15, 67, 12, 43)
```

telephone_number

```
print.telephone_number <- function(tel) {  
  paste(c('Phone number:', a), collapse = ' ')  
}
```

```
print(a) ?
```

OOP basics: class, methods

A set of points ?

```
a <- c(06, 30, 15, 67, 12, 43)
```

```
print(a) ?
```

```
a <- c(06, 30, 15, 67, 12, 43)
```

telephone_number

```
print.telephone_number <- function(tel) {  
  paste(c('Phone number:', a), collapse = ' ')  
}
```

```
print(a) ?
```

polymorphism!

method

OOP basics: class, methods

Polymorphism of the plot function

```
plot(iris)
```

```
plot(Sepal.Length ~ Sepal.Width, data = iris)
```

```
plot(list(x = seq.int(10),  
          y = seq.int(10) + rnorm(10)))
```

OOP basics: class, methods

Polymorphism of the plot function

```
plot(iris)
```

```
plot(Sepal.Length ~ Sepal.Width, data = iris)
```

```
plot(list(x = seq.int(10),  
          y = seq.int(10) + rnorm(10)))
```

Other functions ?

```
print.data.frame <- function(df) {  
  print( summary(df) )  
  cat(paste0('... ', nrow(iris), ' rows)\n'))  
}
```

functional programming

OOP: insist on *what objects are*

functional programming: insists on *how we transform objects*