learn to love the data frame

@JennyBryan

@jennybc

@STAT545

http://stat545.com

decision fatigue
aggravation
cutting corners

mastery
efficiency
safety

decision fatigue
aggravation
cutting corners

mastery
efficiency
safety

I want this for you!

Next 45 mins:

I want to convince you that

“when in doubt, stick it in a data frame”

will

increase your

mastery, efficiency, safety

and reduce misery.

Three tricky bits:

- wide range of previous R experience
- existing BioC context re: classes
- tibble variant of data frame very new

do first bit of pirates vs ninjas live coding

R objects come in a few flavours

a simple view of simple R objects that will get you pretty far

Simple view	Technically correct R view		
	mode	class	typeof
character	character	character	character
logical	logical	logical	logical
numeric	numeric	integer or numeric	integer or double
factor	numeric	factor	integer

R objects come in a few flavours

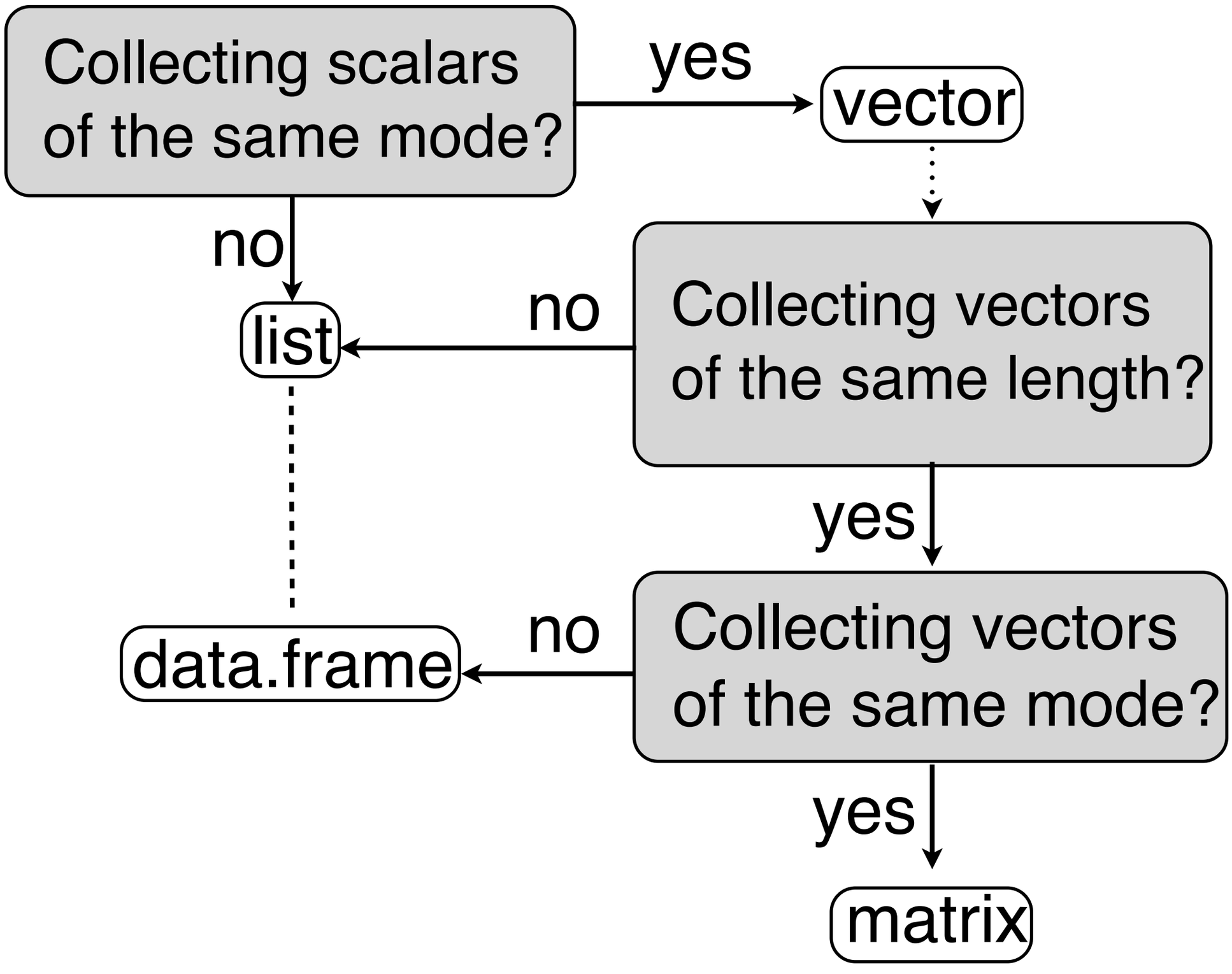
a simple view of simple R objects that will get you pretty far

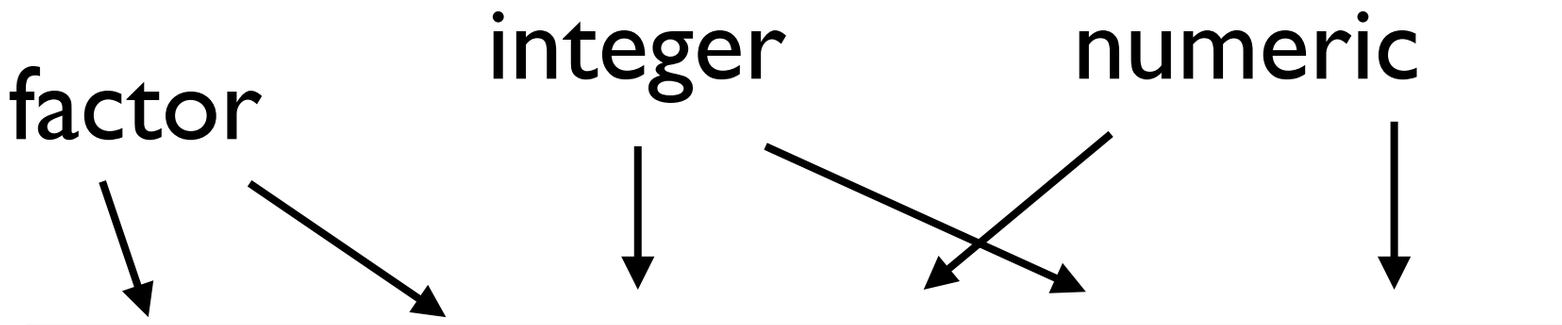
Simple view	Technically correct R view		
	mode	class	typeof
character	character	character	character
logical	logical	logical	logical
numeric	numeric	integer or numeric	integer or double
factor	numeric	factor	integer

R objects come in a few flavours

a simple view of simple R objects that will get you pretty far

Simple view	Technically correct R view		
	mode	class	typeof
character	character	character	character
logical	logical	logical	logical
numeric	numeric	integer or numeric	integer or double
factor	numeric	factor	integer


Collecting scalars of the same mode?
yes
vector
no
list
no
Collecting vectors of the same length?
yes
Collecting vectors of the same mode?
no
data.frame
yes
matrix



country	continent	year	lifeExp	pop	gdpPercap
Afghanistan	Asia	1952	28.801	8425333	779.4453
Afghanistan	Asia	1957	30.332	9240934	820.8530
Afghanistan	Asia	1962	31.997	10267083	853.1007
Afghanistan	Asia	1967	34.020	11537966	836.1971
Afghanistan	Asia	1972	36.088	13079460	739.9811
Afghanistan	Asia	1977	38.438	14880372	786.1134
Afghanistan	Asia	1982	39.854	12881816	978.0114
Afghanistan	Asia	1987	40.822	13867957	852.3959
Afghanistan	Asia	1992	41.674	16317921	649.3414
Afghanistan	Asia	1997	41.763	22227415	635.3414
Afghanistan	Asia	2002	42.129	25268405	726.7341
Afghanistan	Asia	2007	43.828	31889923	974.5803
Albania	Europe	1952	55.230	1282697	1601.0561
Albania	Europe	1957	59.280	1476505	1942.2842
Albania	Europe	1962	64.820	1728137	2312.8890

If you can put it in a data frame, DO THAT.

Operate on the data frame holistically.

Pass it to other functions, pref. intact and whole.

Learn how to limit computation to specific rows or columns. Don’t create copies or excerpts lightly.

minimize the creation of data excerpts and copies ...

... they will just confuse you later

back to pirates vs ninjas live coding

but with the tidyverse

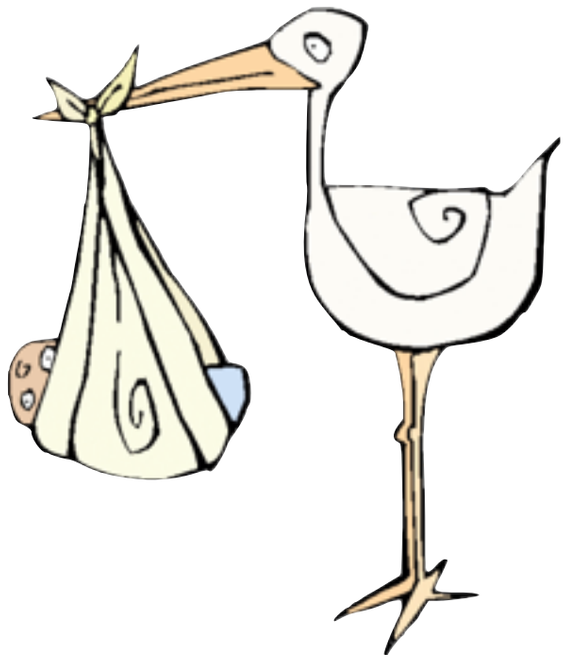
If you can put it in a data frame, DO THAT.

Develop your preferred workflow for df manipulation and use it maniacally.

Strong recommend: use the “tidyverse”

- tibble + dplyr + tidyr
- tbl_df or “tibble” is a variant of data.frame

Where do data frames come from?



Delimited file
`read.table()` and friends
`readr` package

Coercion
`as.data.frame()`
`as_tibble()`

Assembly from vector parts
`data.frame()`
`tibble::tibble(), tibble::enframe()`

Growing existing object
`transform()`
`dplyr::mutate()`

Things you need to know about tibbles:

no partial name matching with `$`
stringsAsFactors = FALSE
df[, “X1”] will be a tibble, i.e. drop = FALSE
you can print them with wild abandon
no row names
do not alter with variable names
will only recycle input of length 1

Preview of Friday "what should you do next?"

What if you need to divide the df into its rows or groups of rows and compute on them?

Putting complicated objects into list-columns, temporarily. How and why?

decision fatigue
aggravation
cutting corners

mastery
efficiency
safety