

0. PROGRAMMING WITH R
1. ANALYTIC QUESTIONS
2. EXPLORATORY DATA ANALYSIS
3. VECTORS
4. DATA FRAMES
5. LOAD DATA
6. TIDY DATA
7. STRINGS
8. TRANSFORM DATA
9. UNSTRUCTURED DATA
10. VISUALIZE DATA
11. COMMUNICATE FINDINGS
12. PYTHON

PROGRAMMING WITH R

variables

control structures

loops

functions

libraries

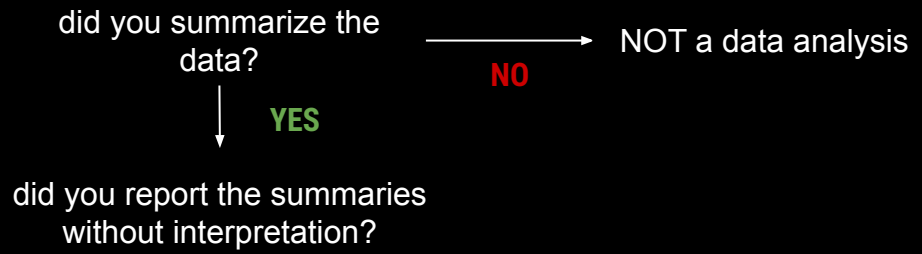
ANALYTIC QUESTIONS

did you summarize the
data?

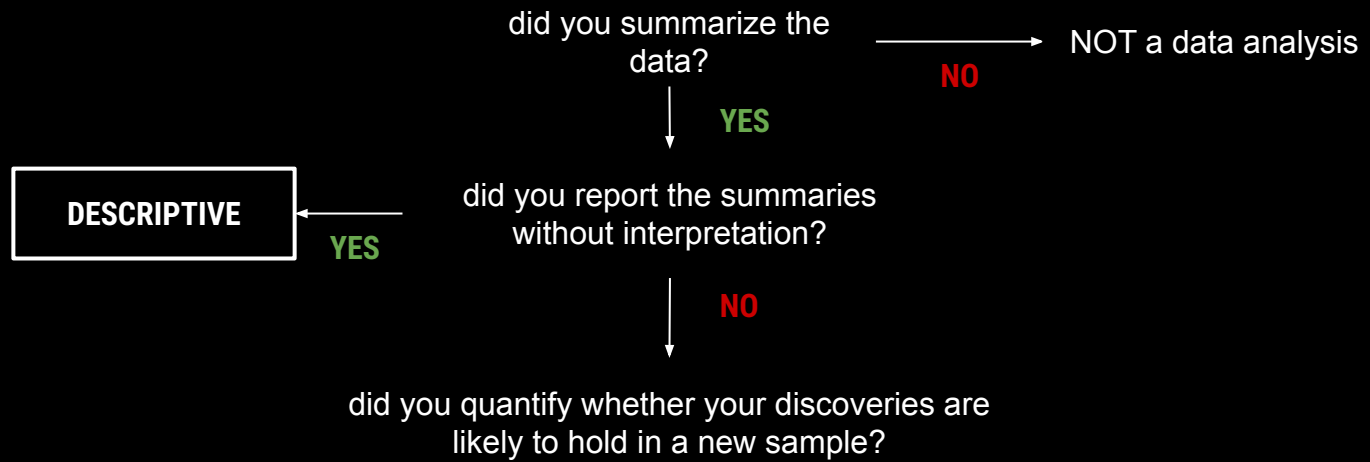
did you summarize the
data?

NO

NOT a data analysis

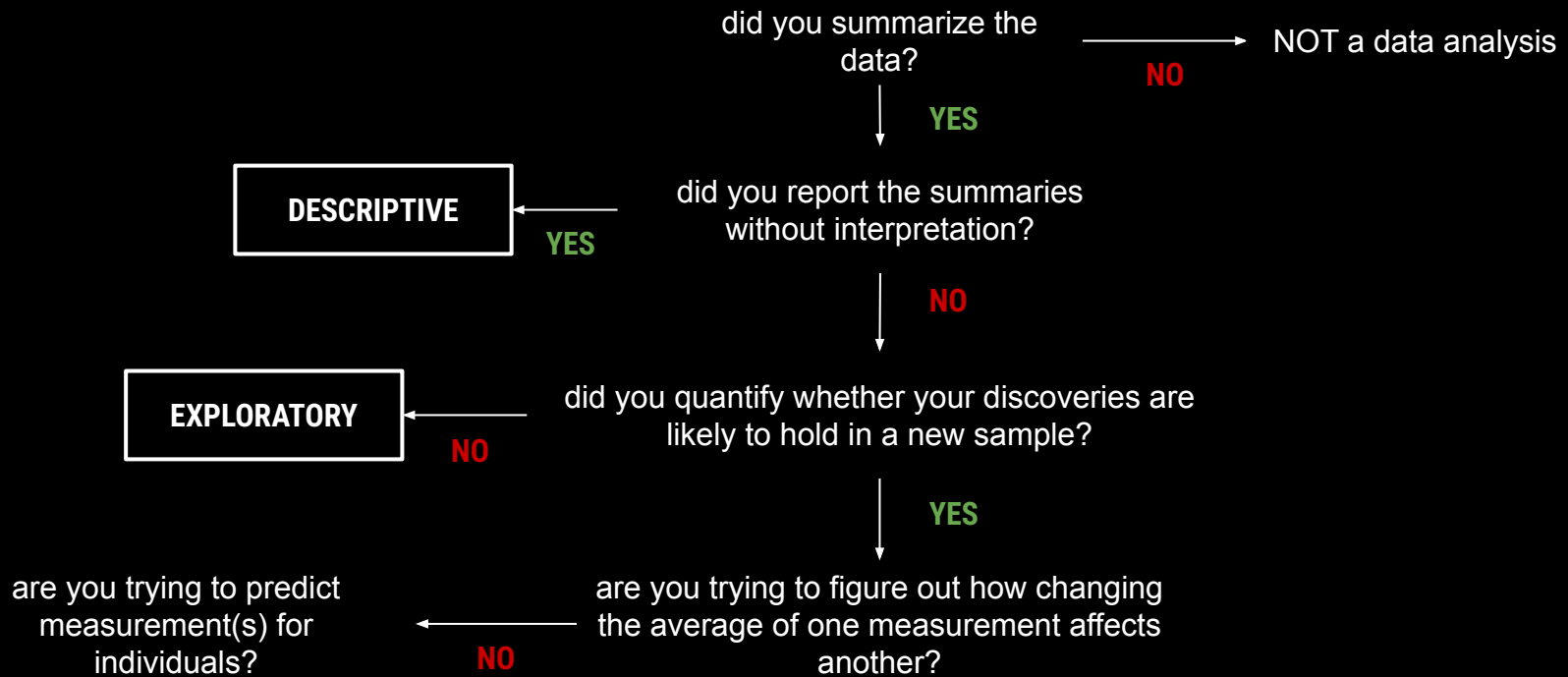














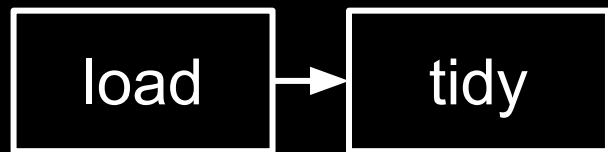


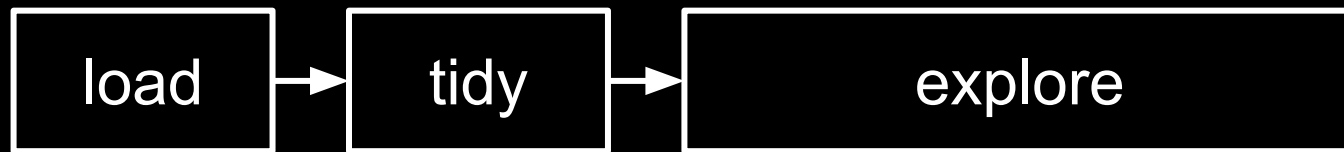




EXPLORATORY DATA ANALYSIS

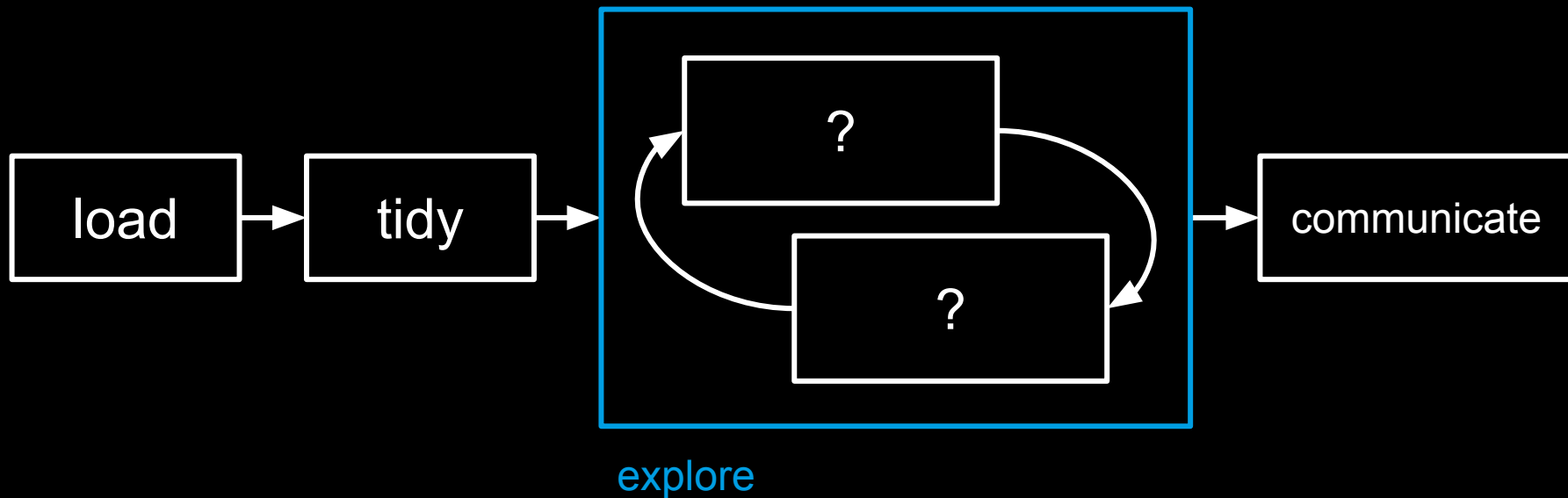
load

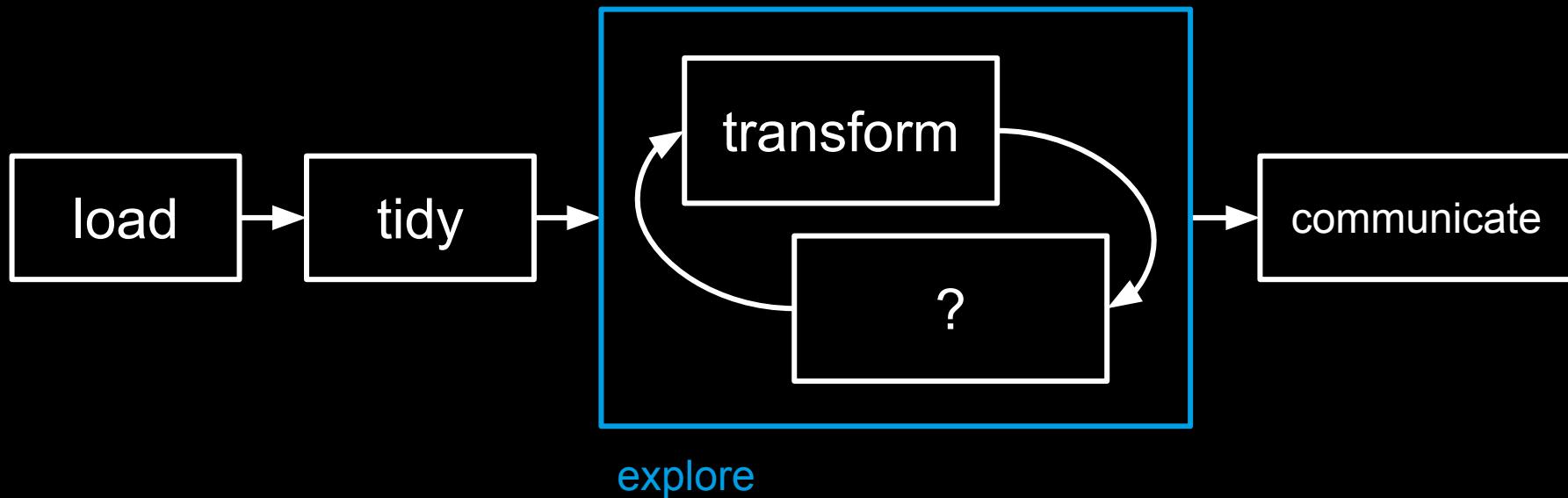


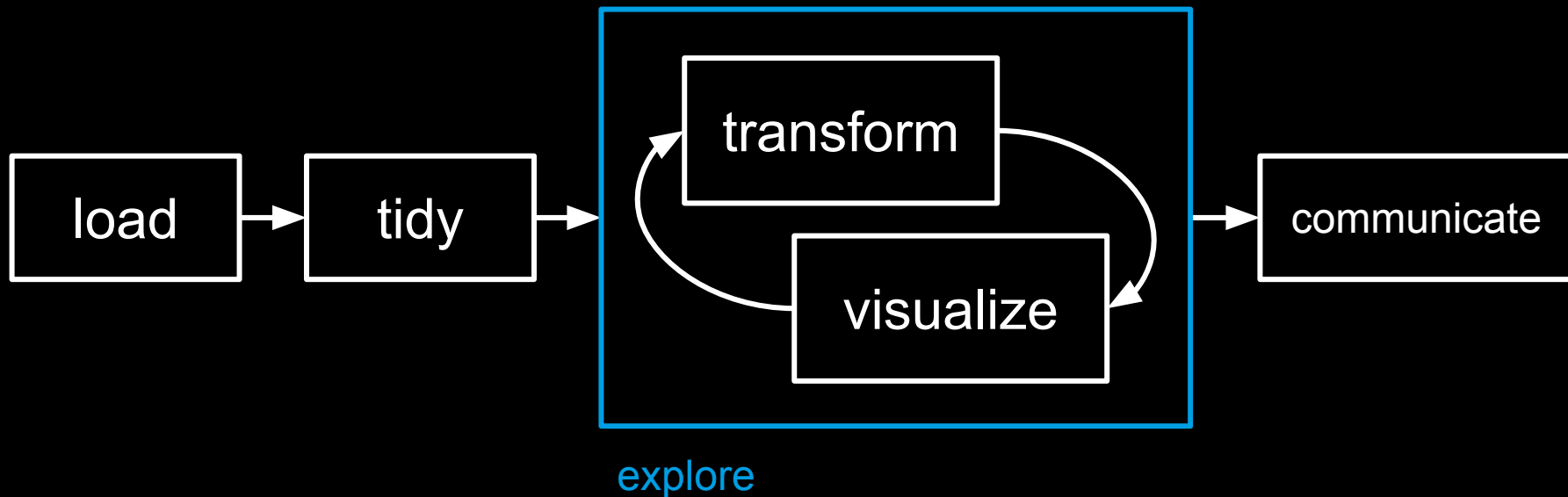


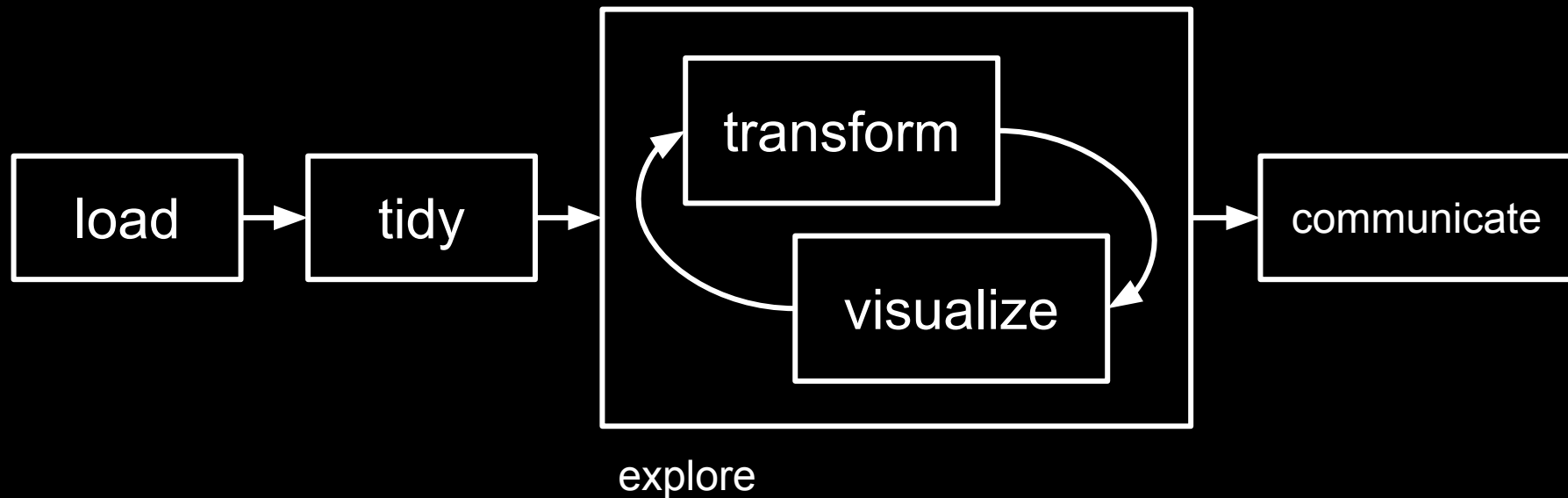


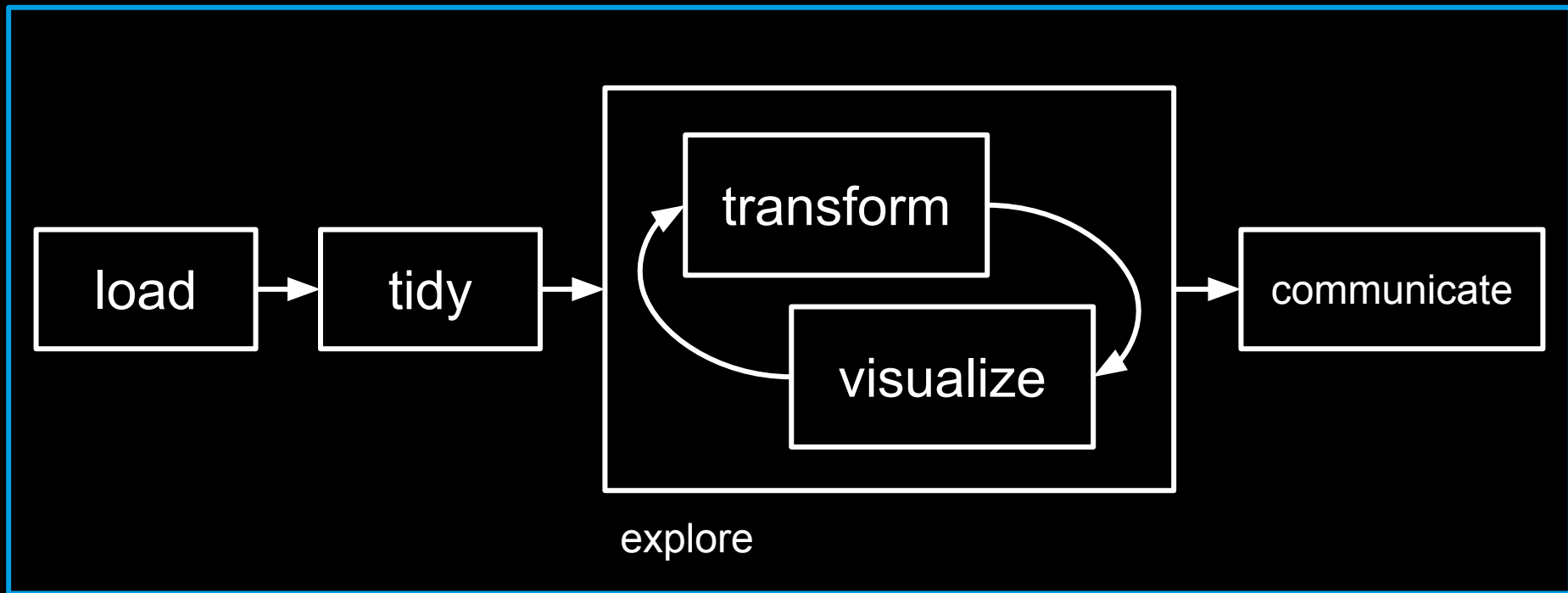




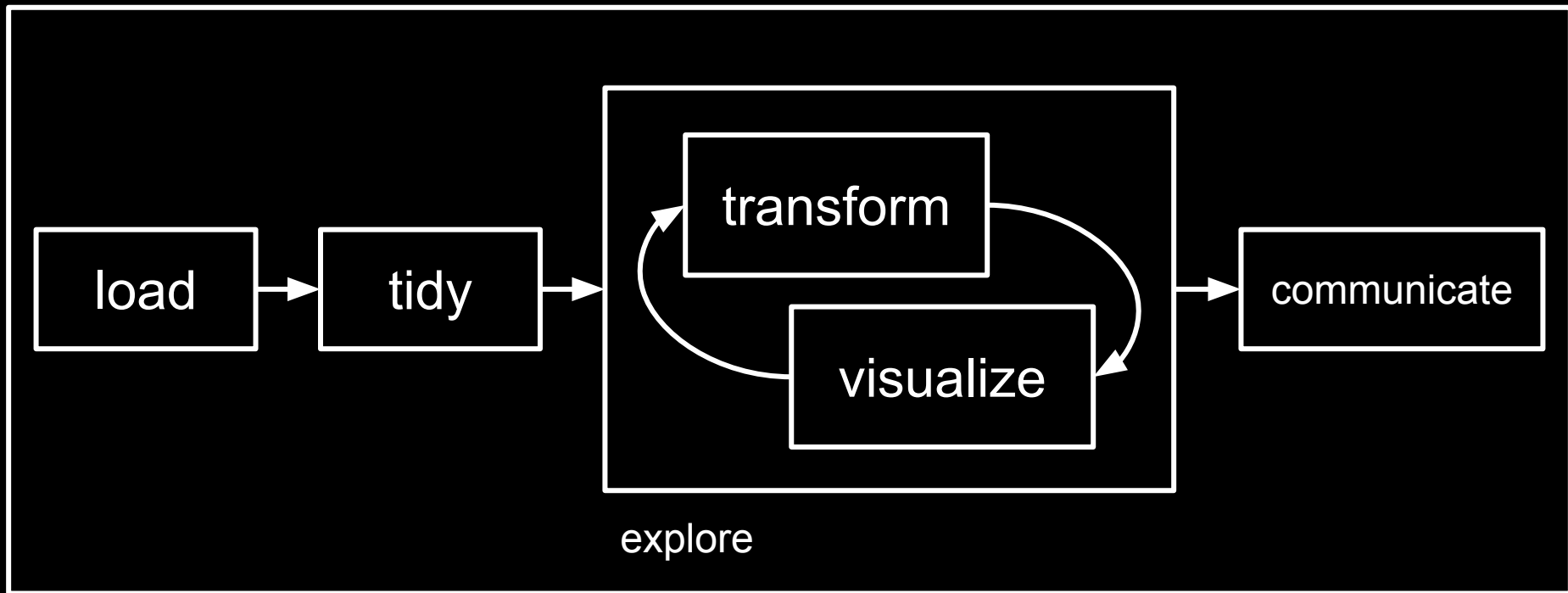








program



program

VECTORS

$c(2, 3, 5, 7, 11, 13, 17)$

DATA FRAMES

{{ tibble }}

LOAD DATA

{{ readr }}

```
read_csv()  
read_delim()
```

{{ readxl }}

`read_excel()`

TIDY DATA

tidy data

each variable is a column;
each column is a variable.

each observation is a row;
each row is an observation.

each value is a cell;
each cell is a single value.

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898

variables

country	year	cases	population
Afghanistan	1999	745	1997071
Afghanistan	2000	2666	2095360
Brazil	1999	37737	17296362
Brazil	2000	60488	17494898

observations

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898

values

country	year	type	count
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898

longer



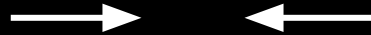
country	year	type	count
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898

wider



country	cases_1999	cases_2000	pop_1999	pop_2000
Afghanistan	745	2666	19987071	20595360
Brazil	37737	172006362	80488	174504898

compressed



country	year	rate
Afghanistan	1999	745 / 19987071
Afghanistan	2000	2666 / 20595360
Brazil	1999	37737 / 172006362
Brazil	2000	80488 / 174504898

tidy

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898

tidy

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898

vector

{{ tidyrr }}

```
pivot_wider()
```

`pivot_longer()`

STRINGS

{{ stringr }}

`str_trim()`
`str_squish()`

str_starts()
str_ends()
str_detect()

“Annabel Miller”

“Annabel Miller”

```
str_starts(txt, "Anna")
```

“Annabel Miller”

```
str_ends(txt, "Miller")
```

“Annabel Miller”

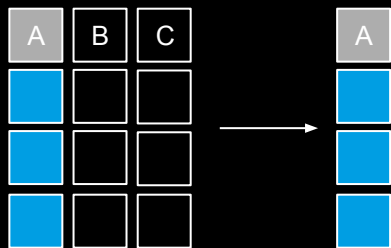
```
str_detect(txt, "Mill")
```


TRANSFORM DATA

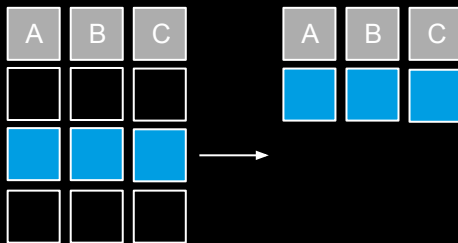
{{ dplyr }}

types of transformations

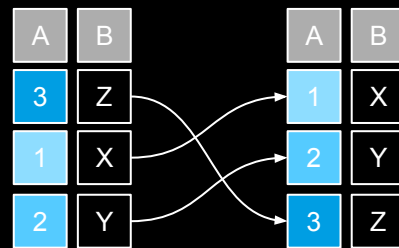
`select()`



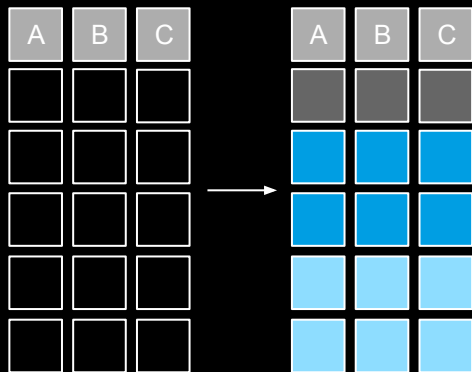
`filter()`



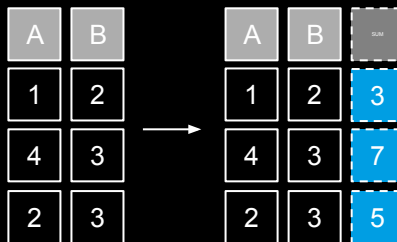
`arrange()`



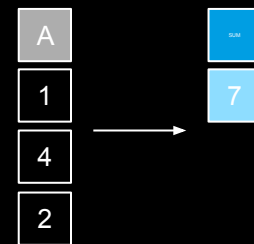
`group_by()`



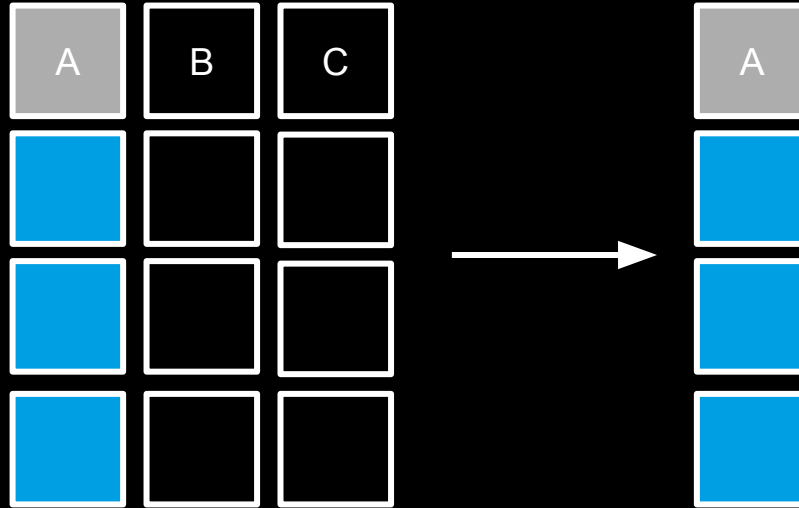
`mutate()`



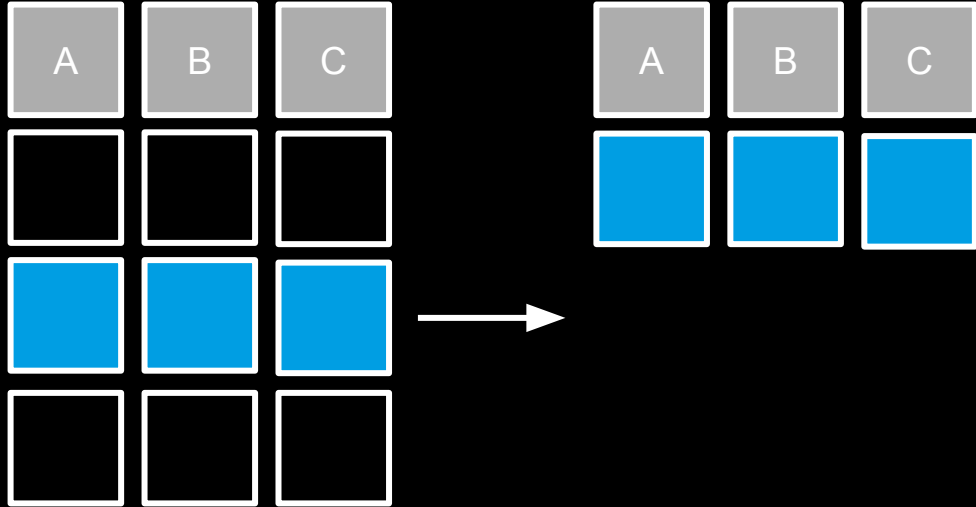
`summarize()`



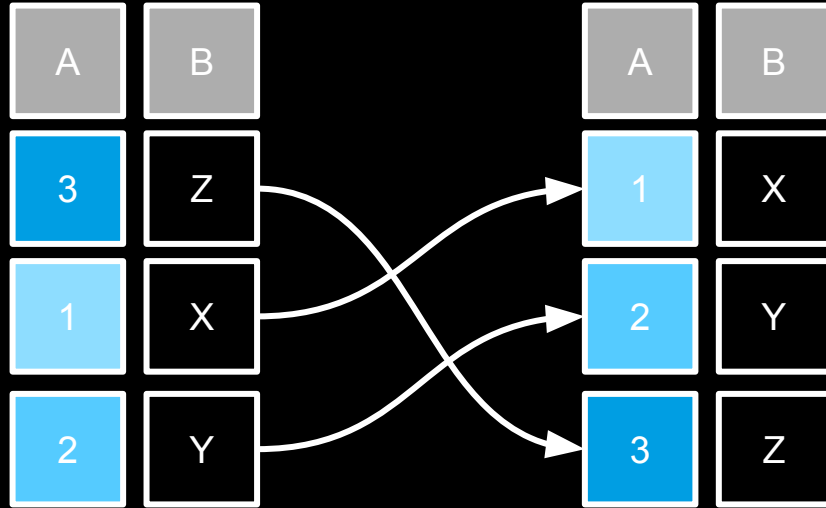
`select()`



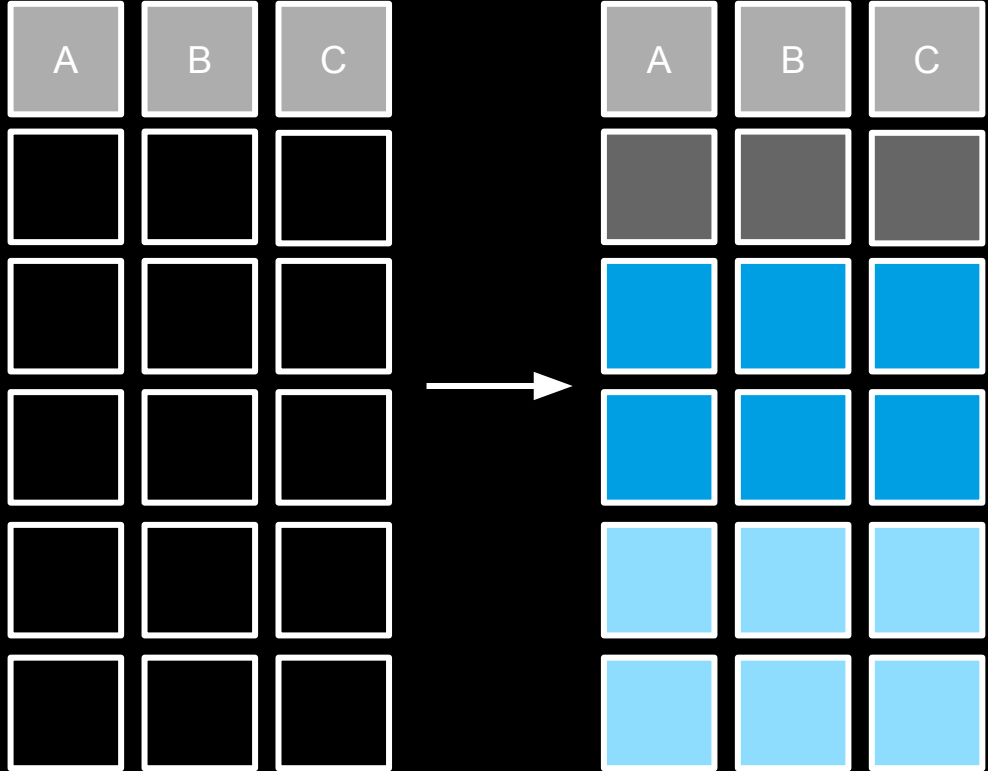
`filter()`



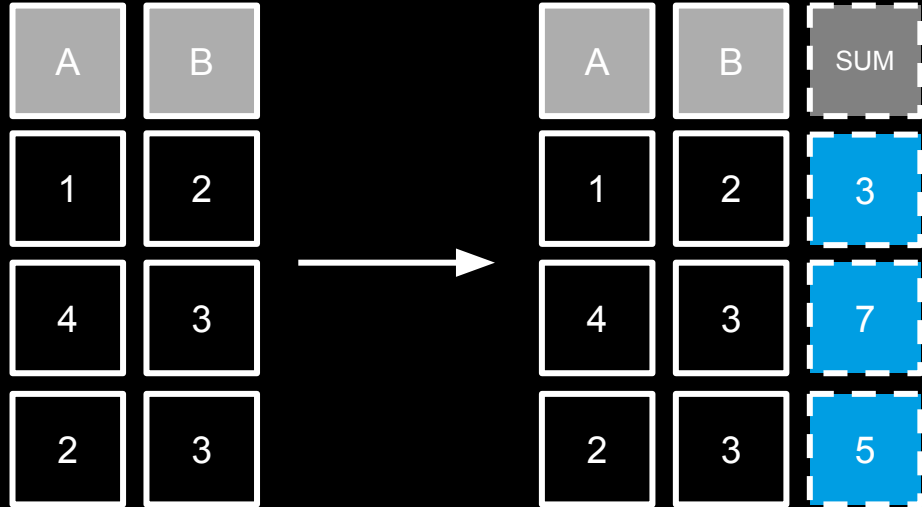
arrange()



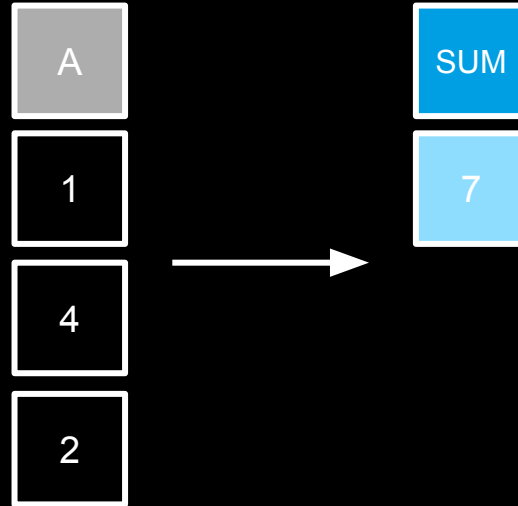
`group_by()`



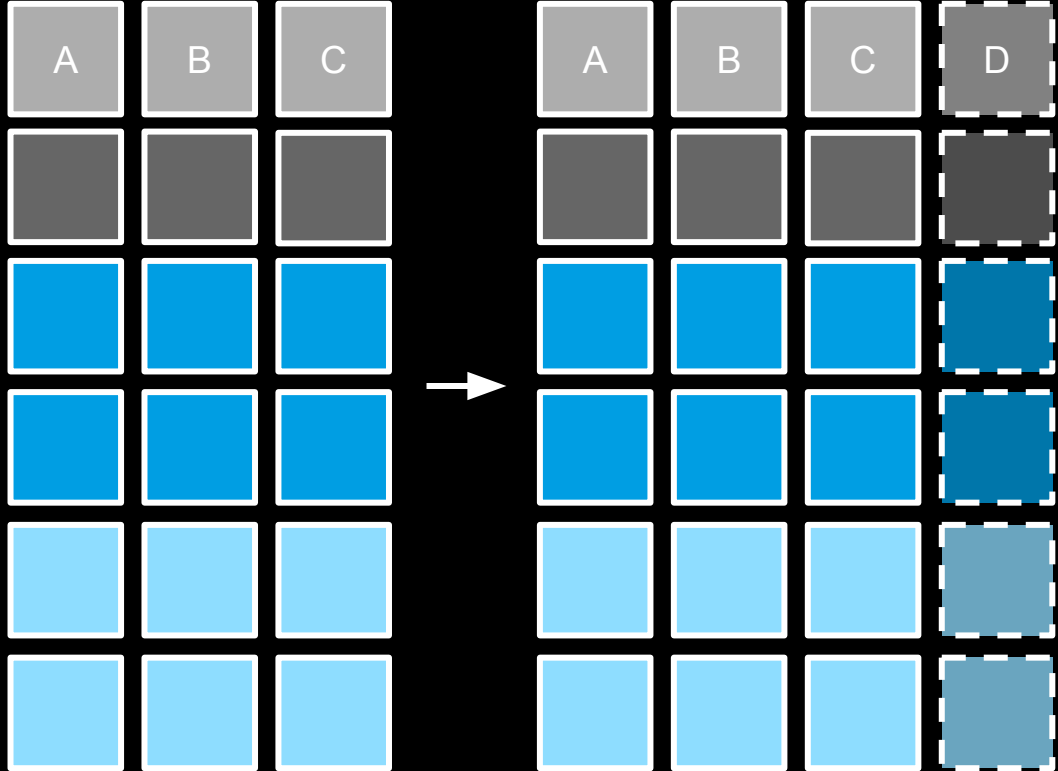
mutate()



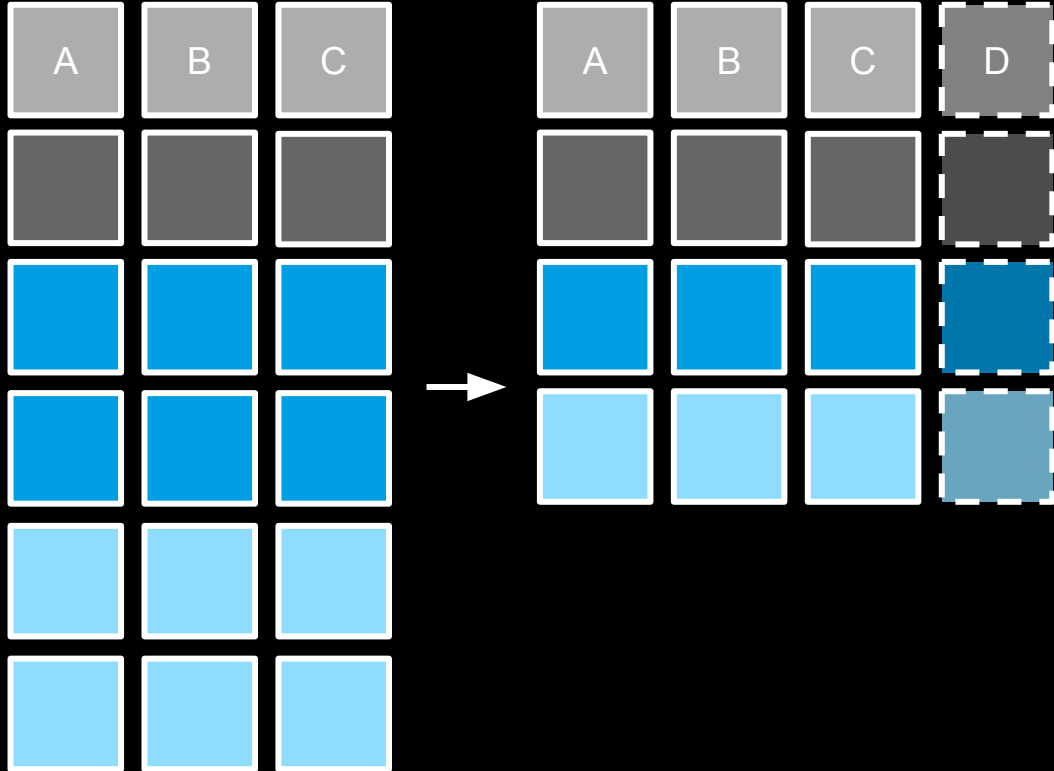
`summarize()`



`group_by()`
+
`mutate()`

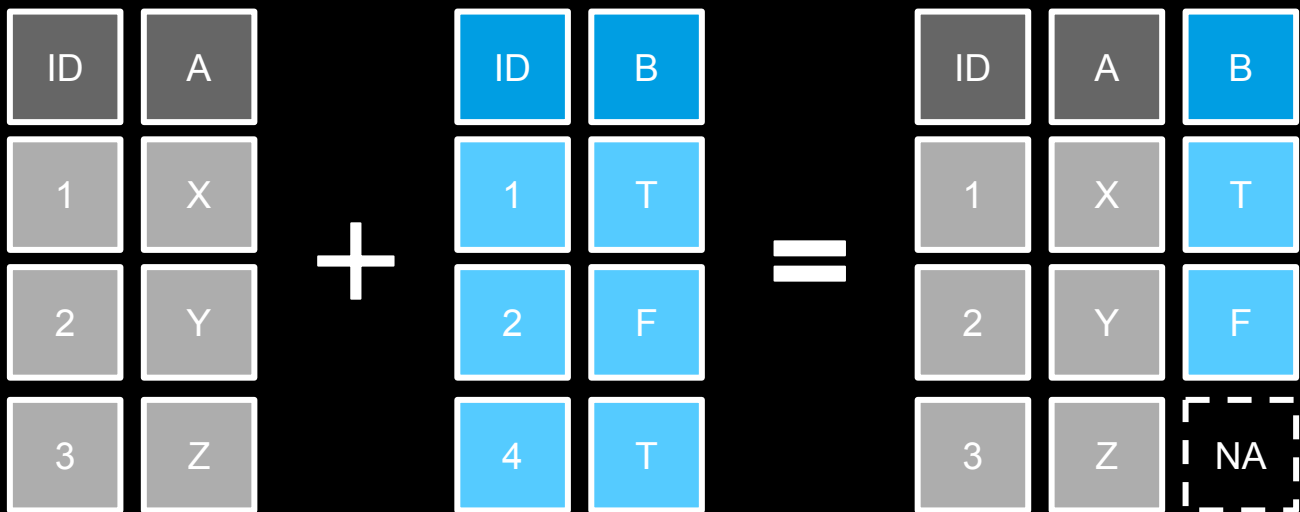


`group_by()`
+
`summarize()`

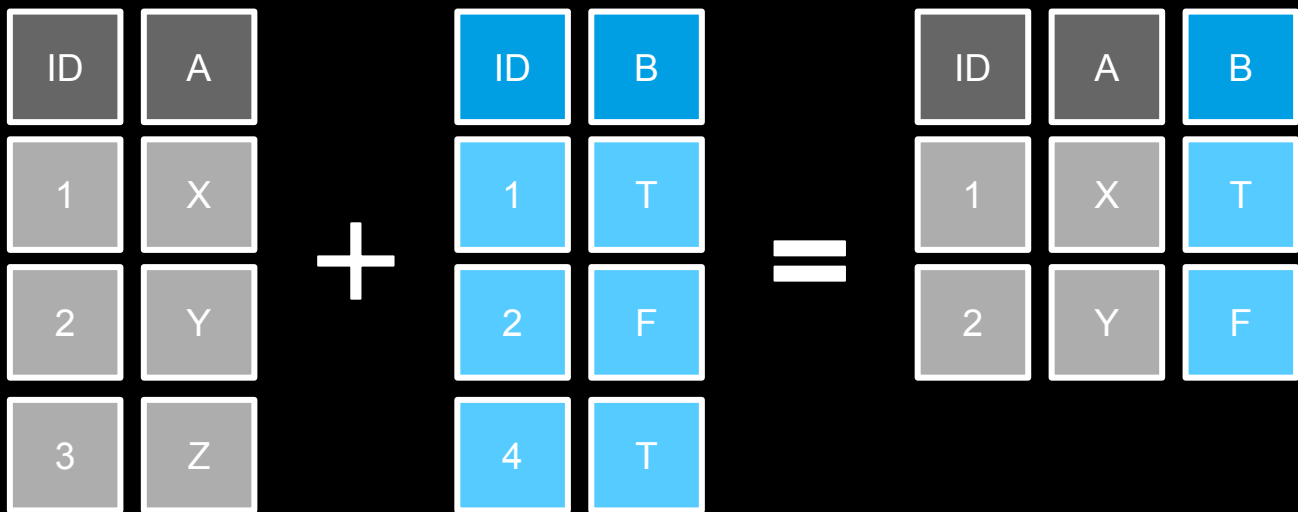


joining data

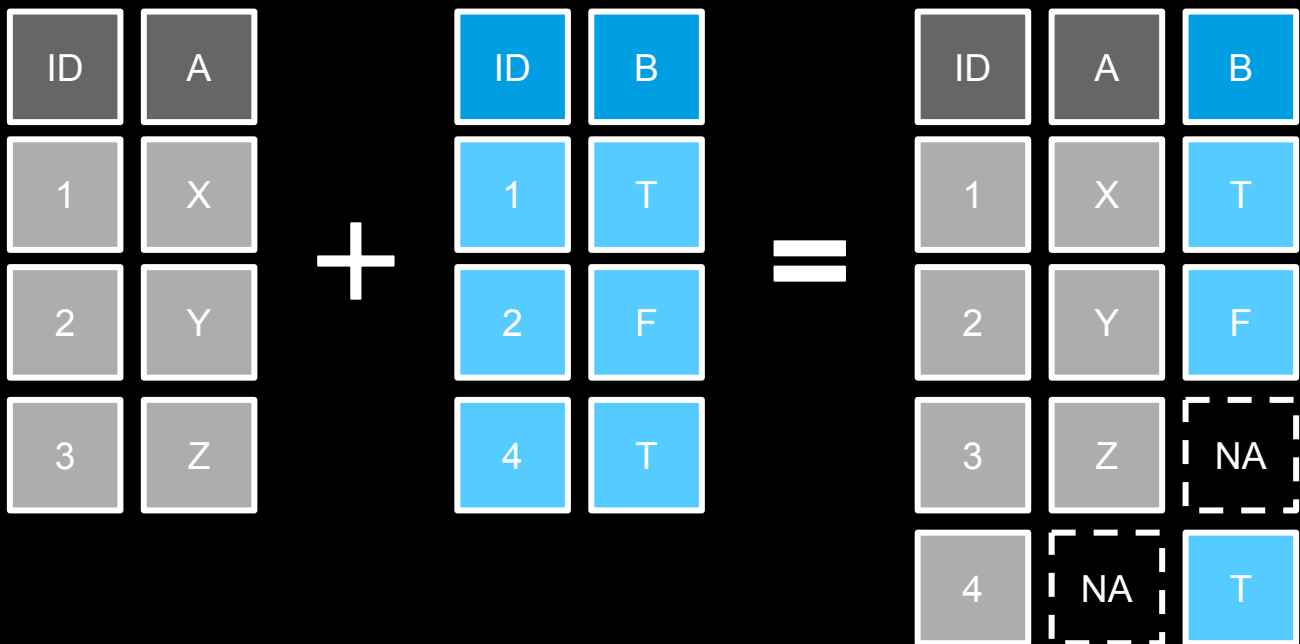
`left_join()`



`inner_join()`



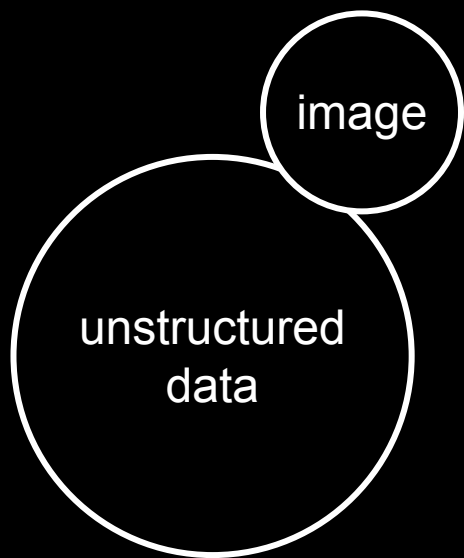
```
full_join()
```

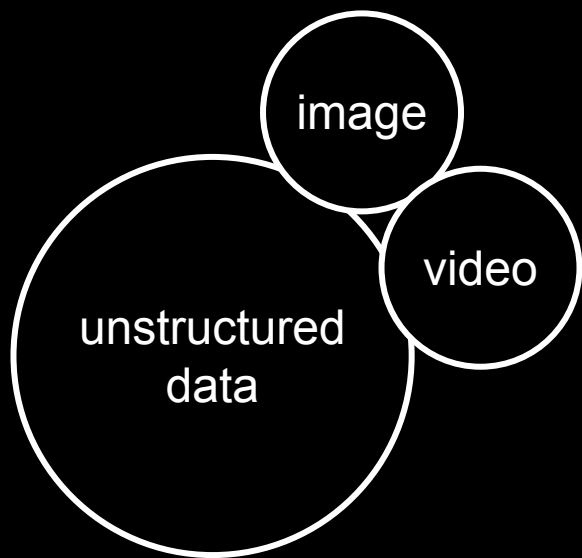


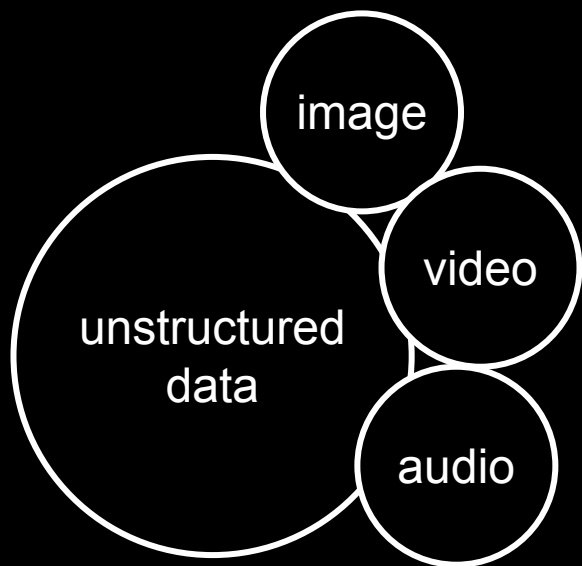
UNSTRUCTURED DATA

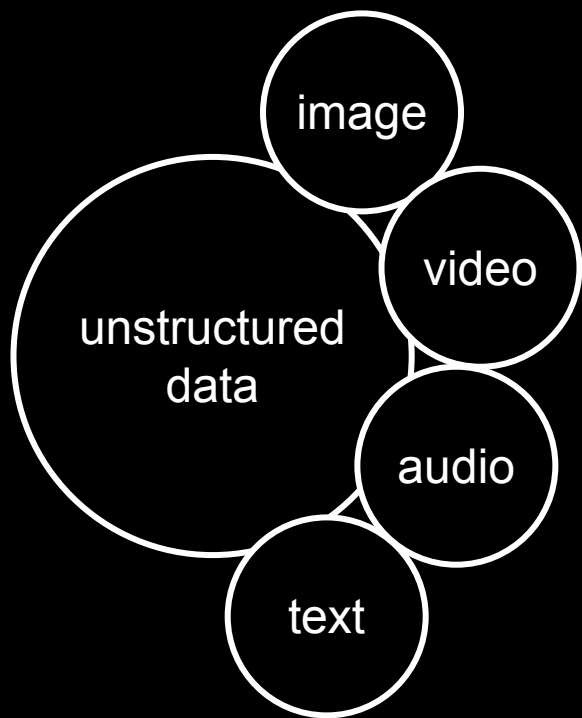


unstructured
data

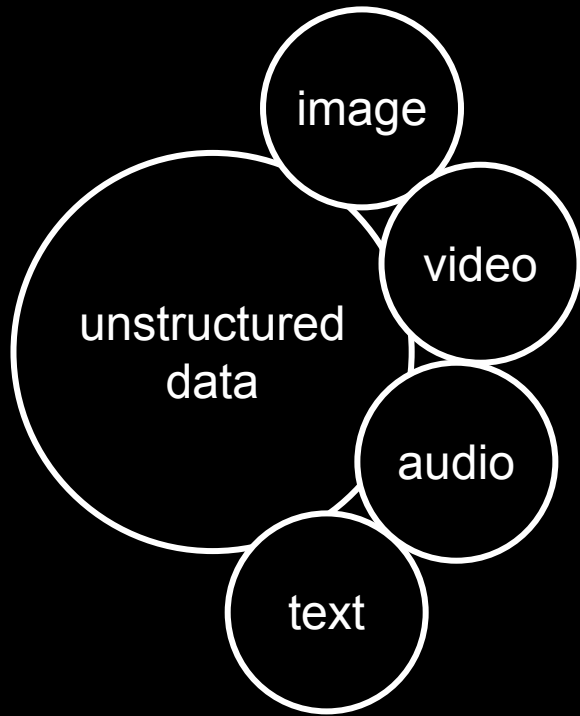




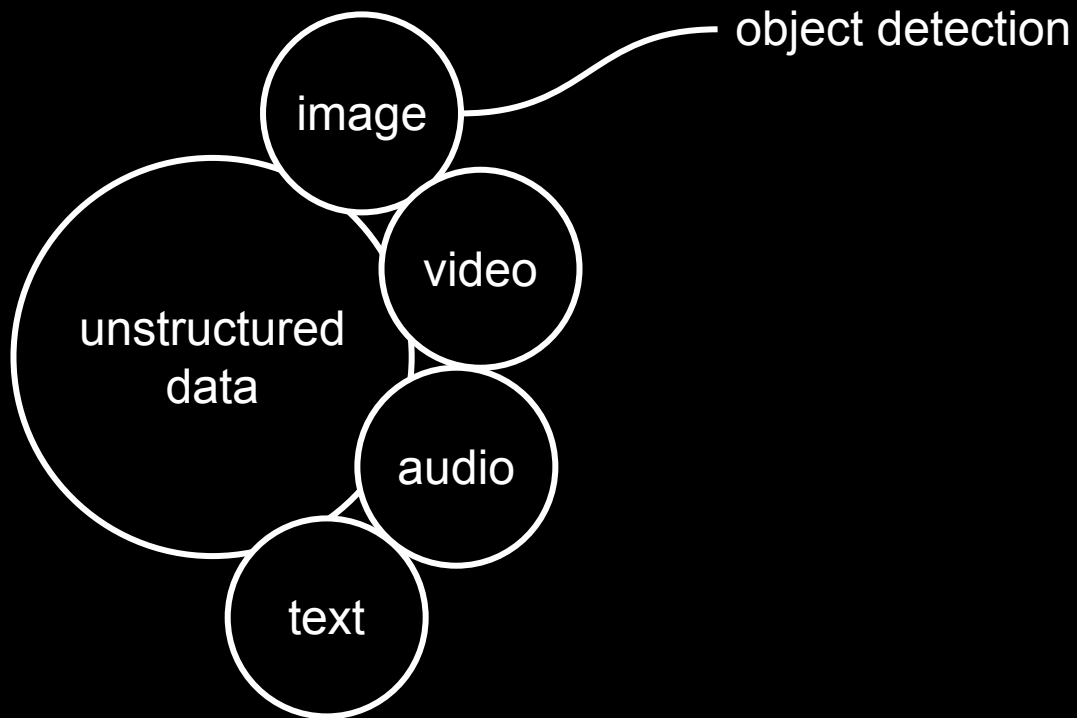




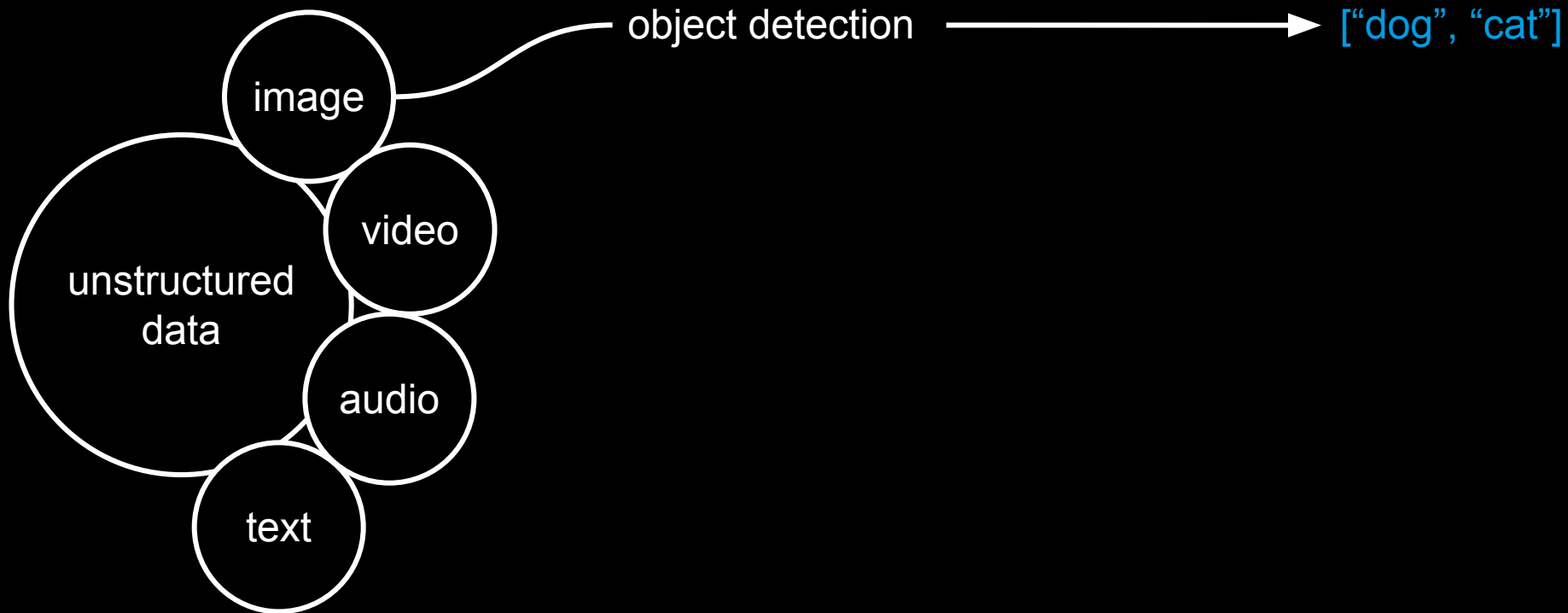
no handles to grab



no handles to grab



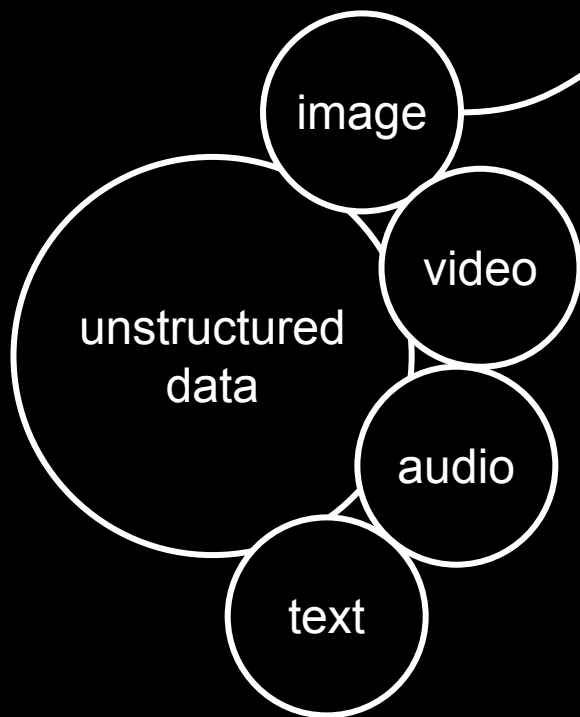
no handles to grab



no handles to grab

algorithm

extracted, structured information



object detection

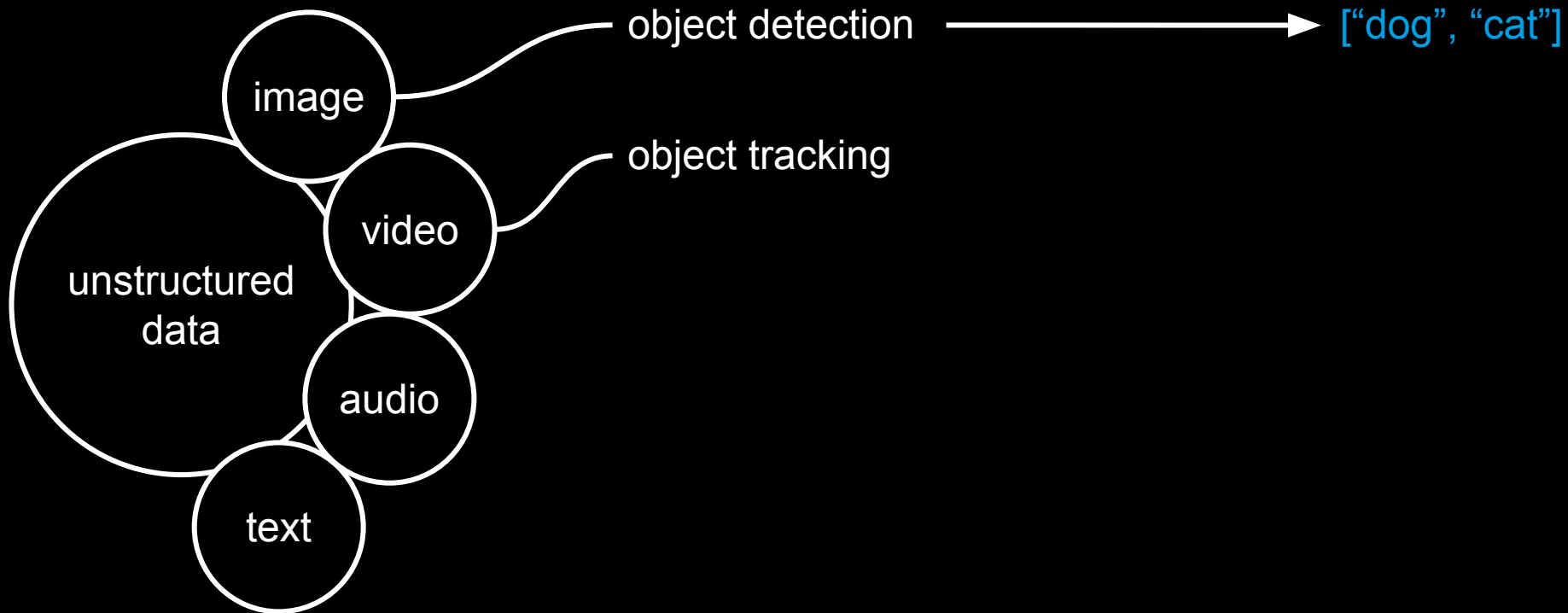


`["dog", "cat"]`

no handles to grab

algorithm

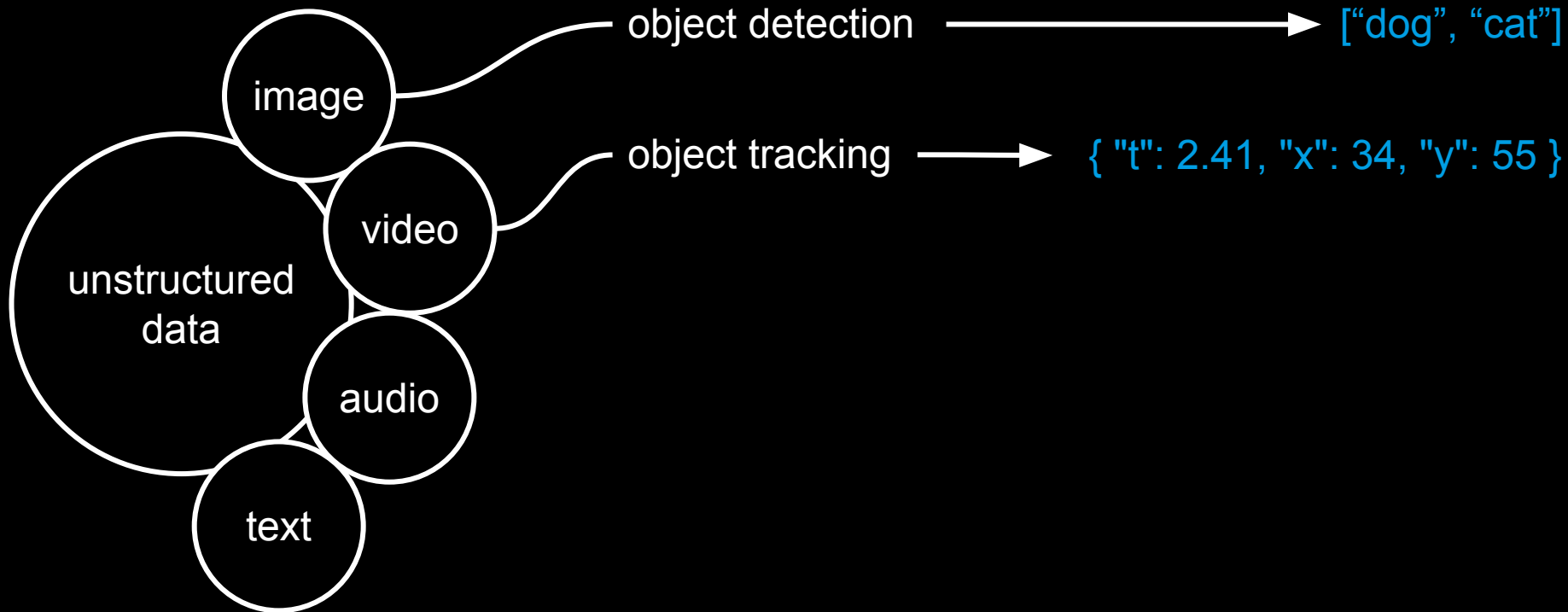
extracted, structured information



no handles to grab

algorithm

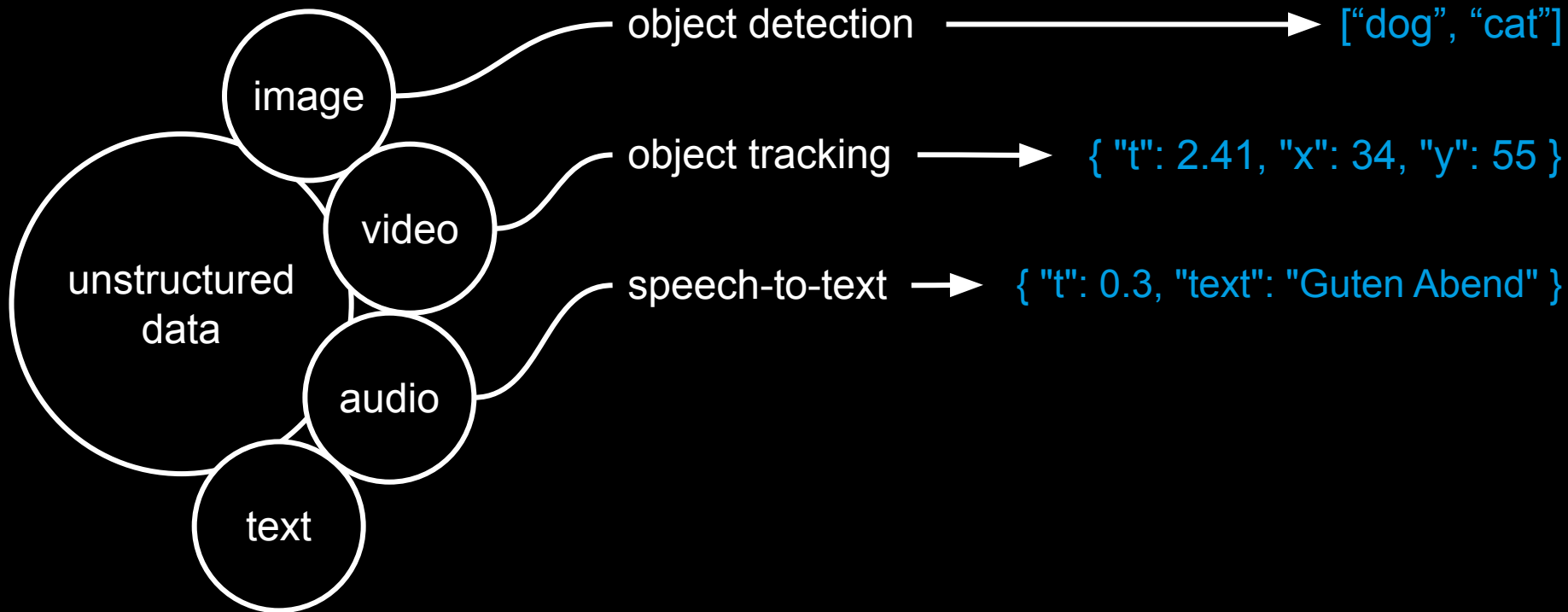
extracted, structured information



no handles to grab

algorithm

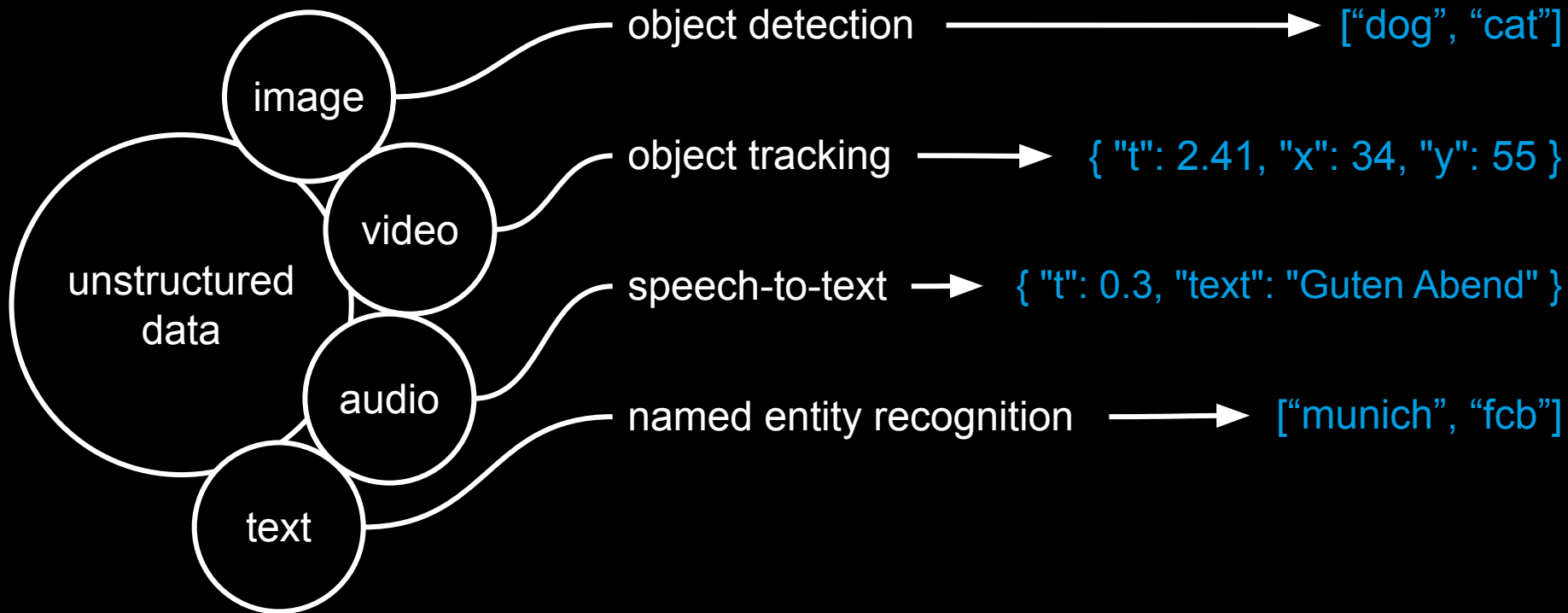
extracted, structured information



no handles to grab

algorithm

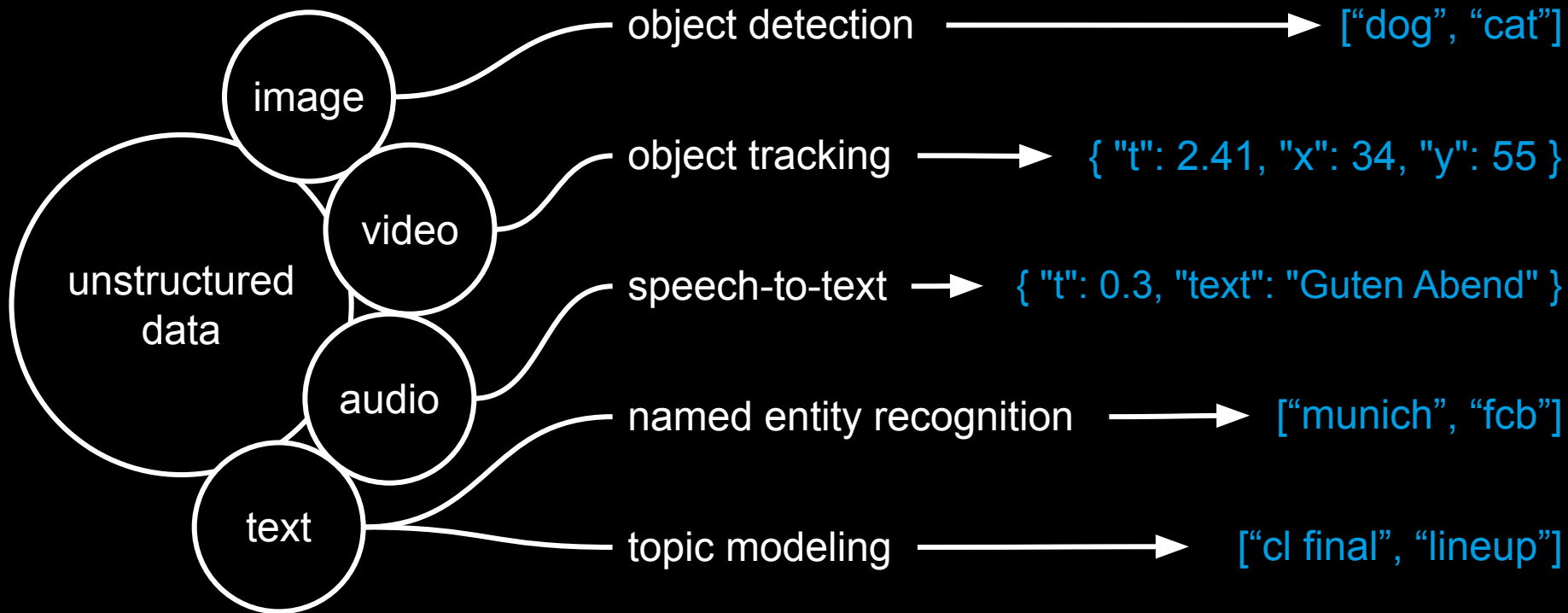
extracted, structured information



no handles to grab

algorithm

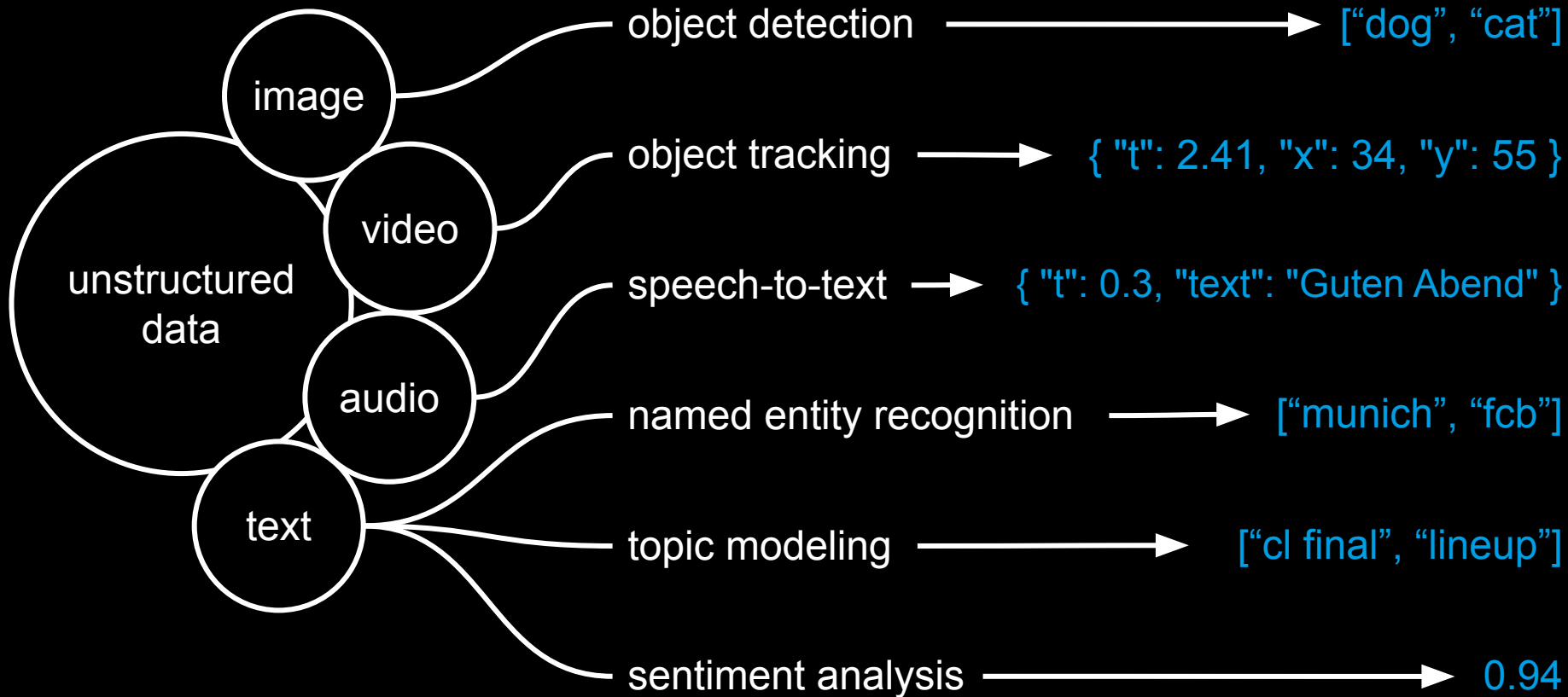
extracted, structured information



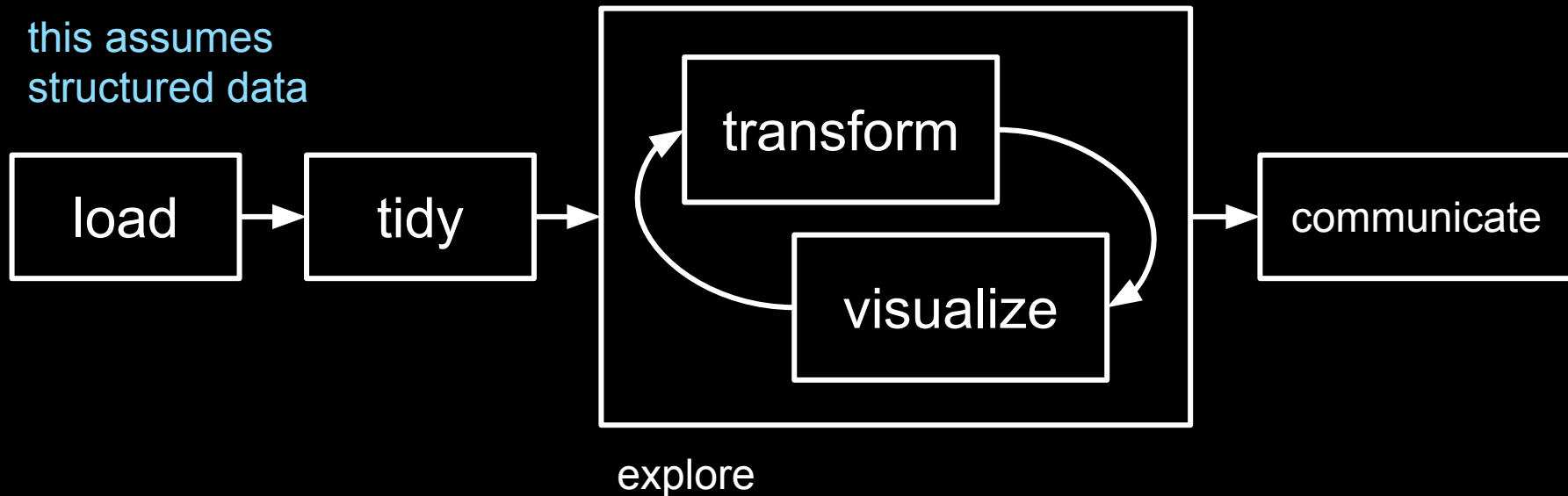
no handles to grab

algorithm

extracted, structured information

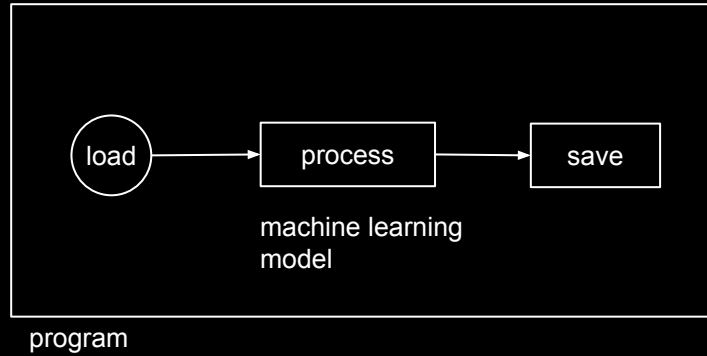


this assumes
structured data

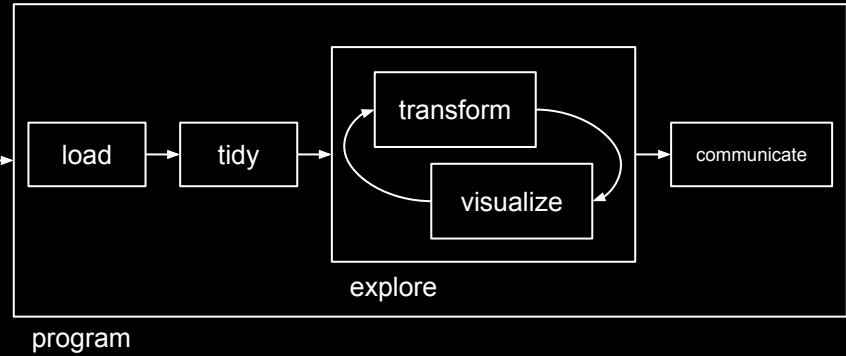


program

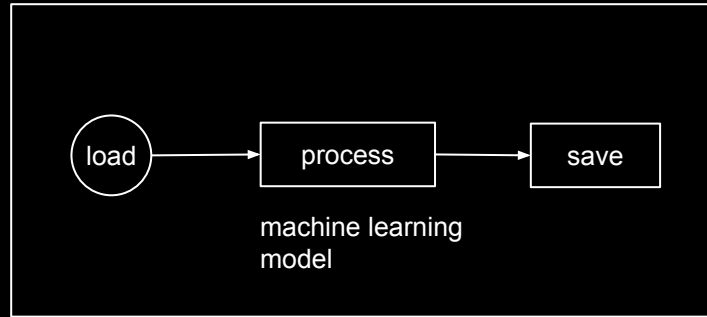
pre-process
unstructured data



exploratory data
analysis



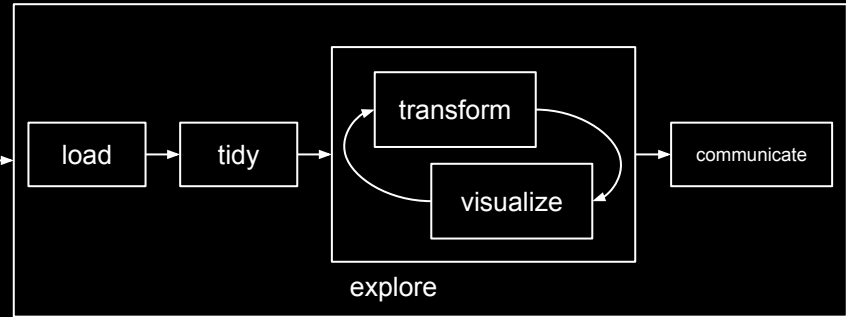
pre-process
unstructured data



program



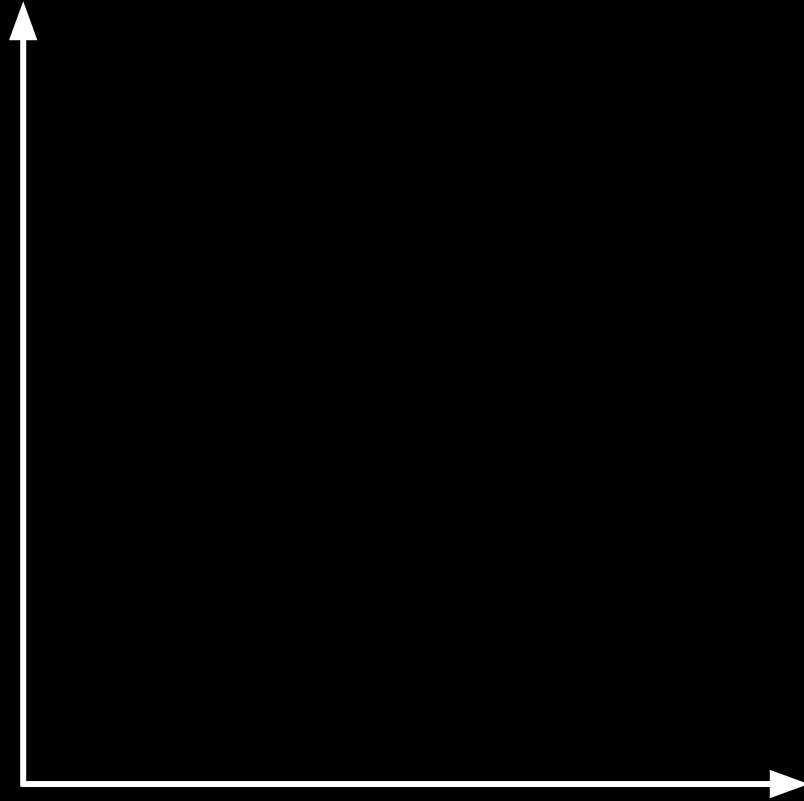
exploratory data
analysis

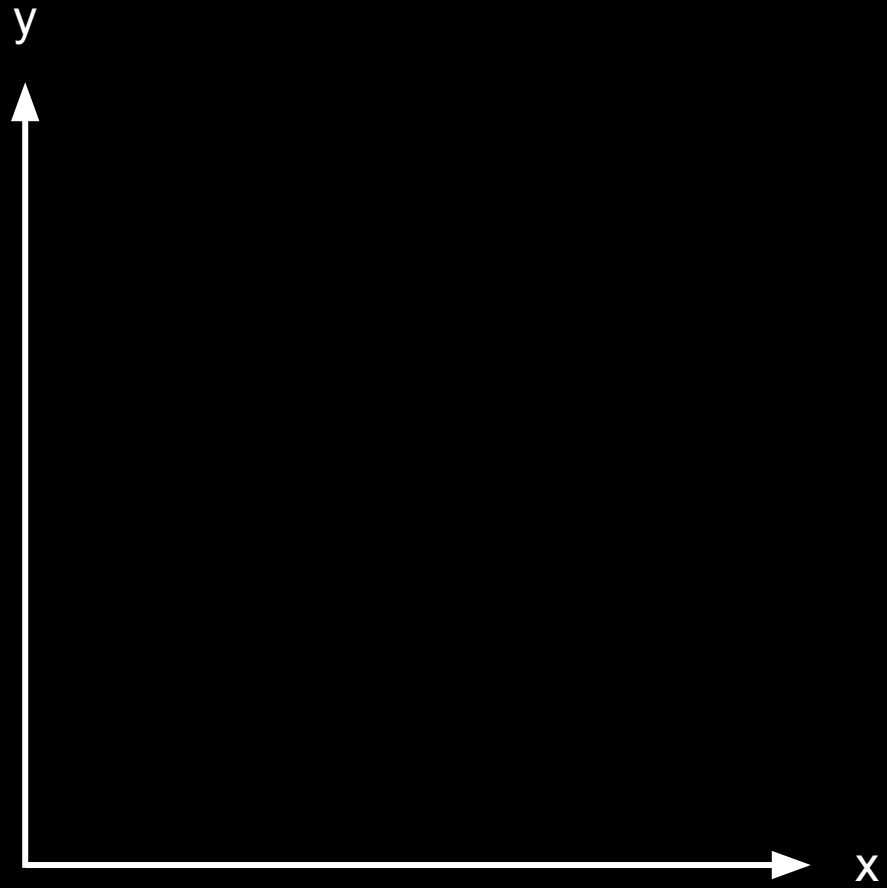


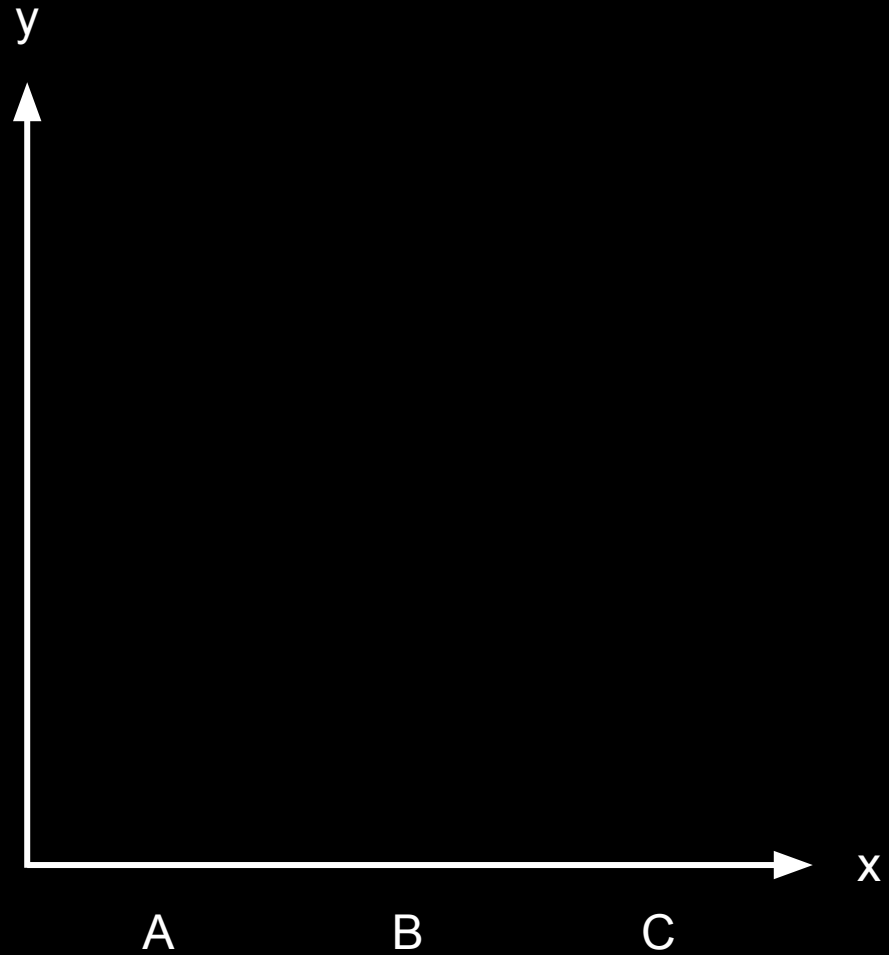
VISUALIZE DATA

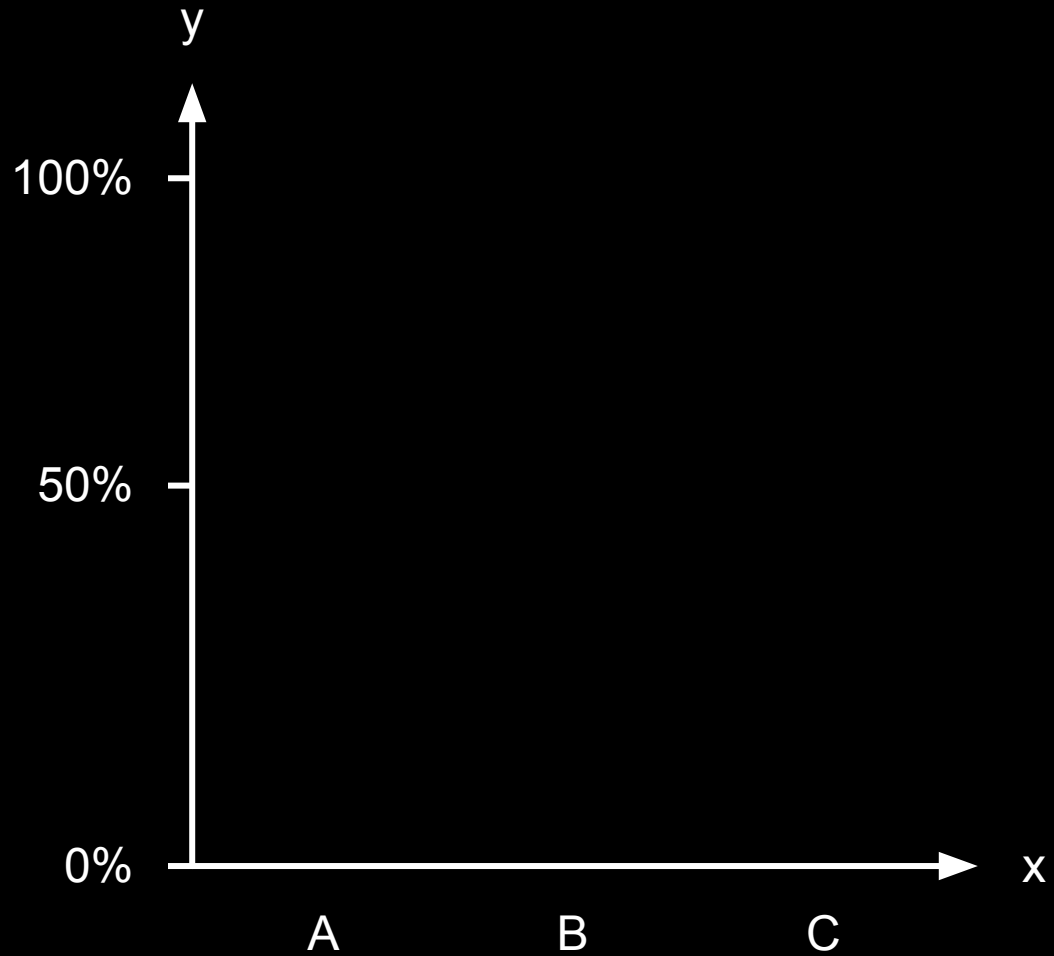
data

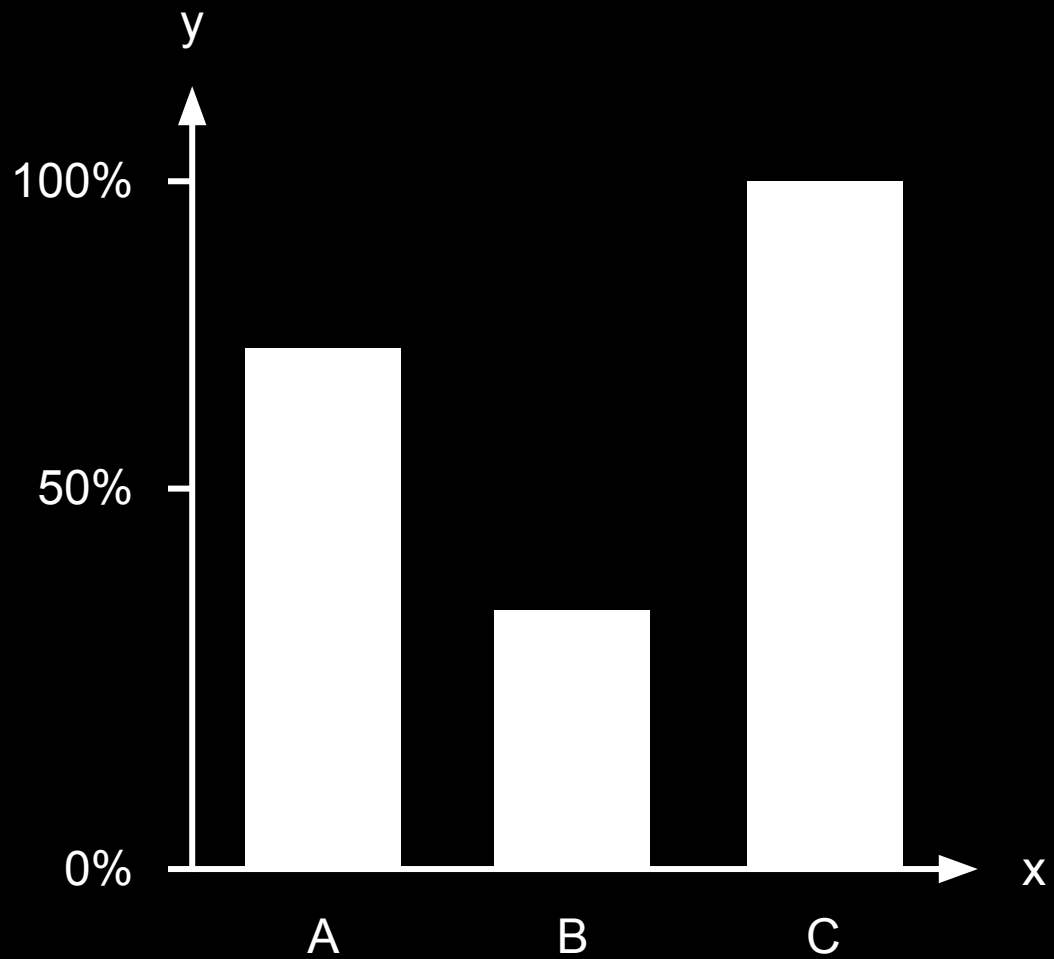
category	pct
A	75
B	33
C	100



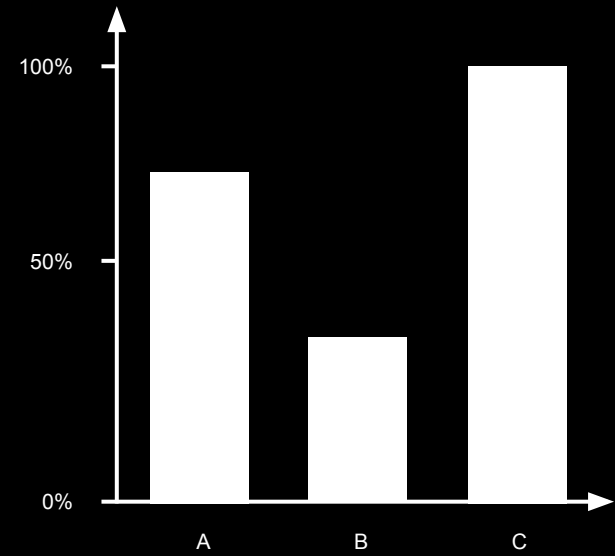








category	pct
A	75
B	33
C	100



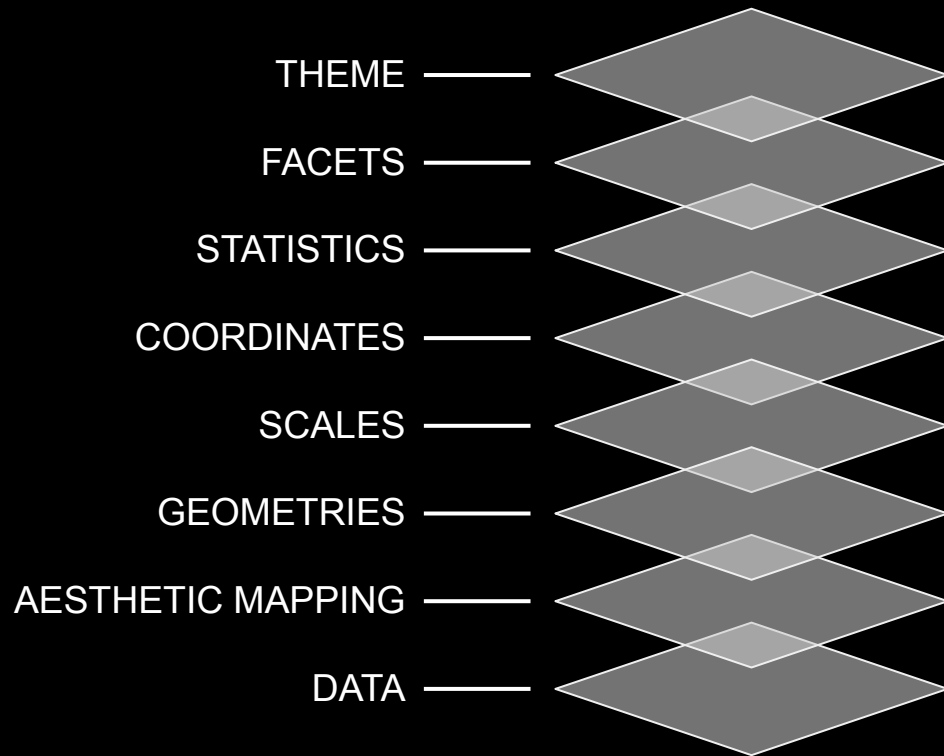
{{ ggplot2 }}

why visualize?

{{ ggplot2 }}

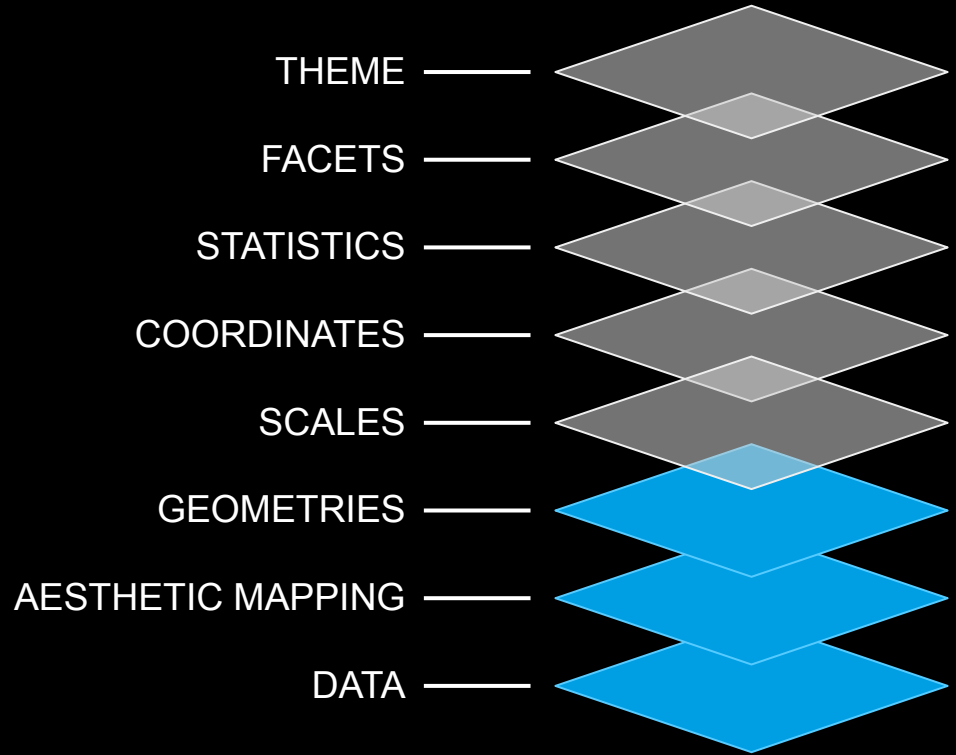
grammar of graphics

any
data
visualization



has useful defaults

mandatory

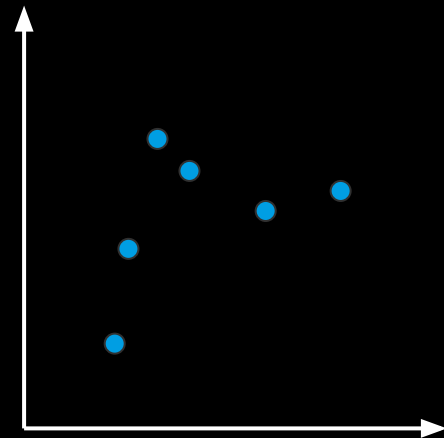


`ggplot()`

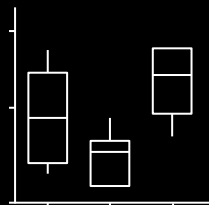
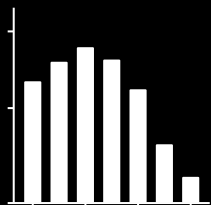
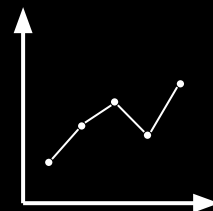
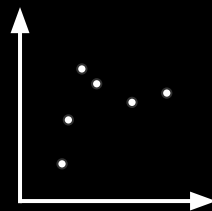
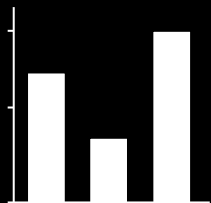
```
ggplot() +  
  aes()
```

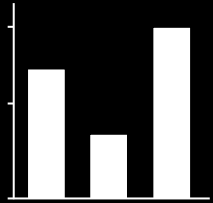
```
ggplot() +  
  aes() +  
  geom_point()
```

```
ggplot() +  
  aes() +  
  geom_point()
```

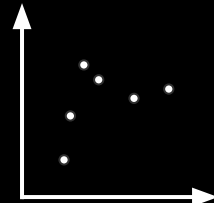


basic plots

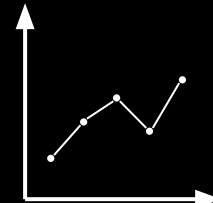




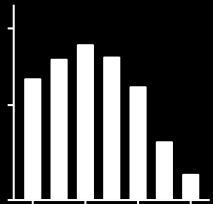
bar chart



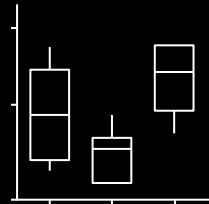
scatter plot



line chart

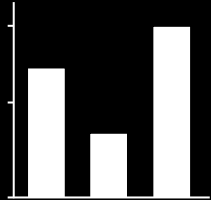


histogram



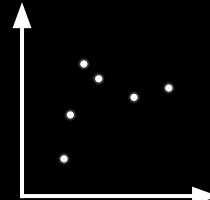
box plot

amounts
proportions
distributions (discrete)



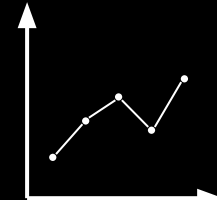
bar chart

associations
patterns



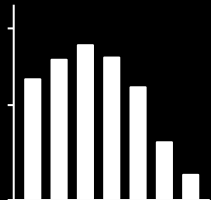
scatter plot

trends
developments



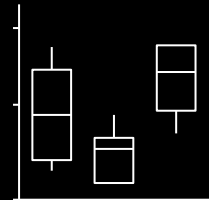
line chart

distributions (continuous)



histogram

compare distributions (continuous)



box plot

COMMUNICATE FINDINGS

Quarto

PYTHON

{{ reticulate }}