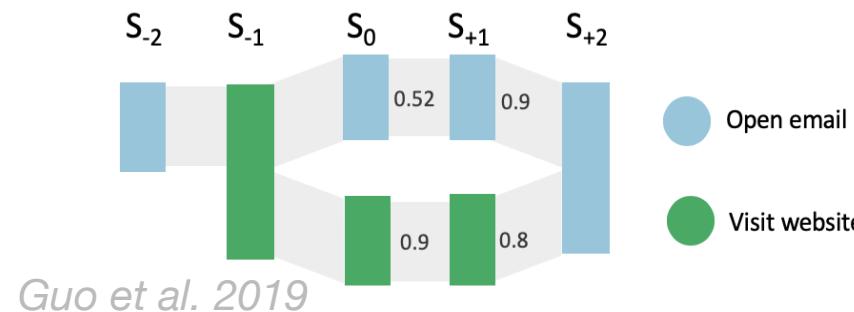


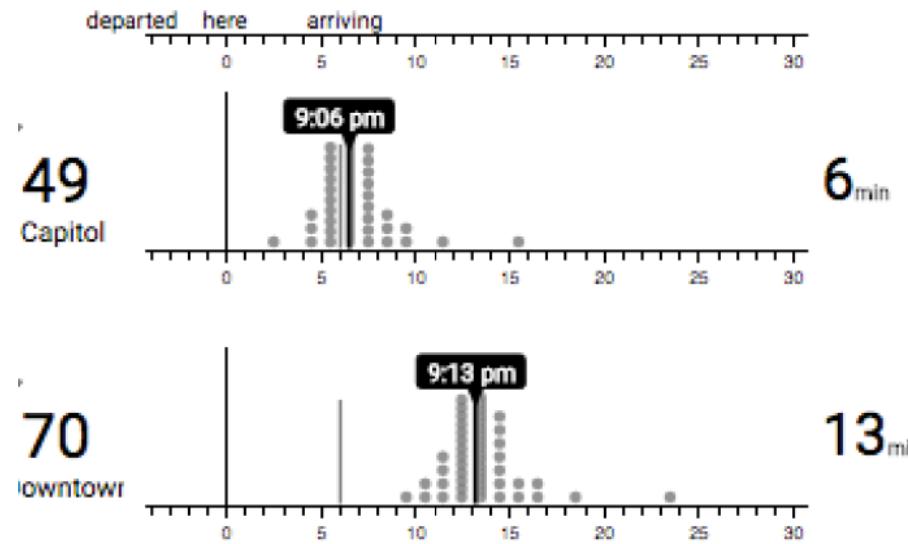
A Probabilistic Grammar of Graphics

Xiaoying Pu
Prelim presentation

Machine learning

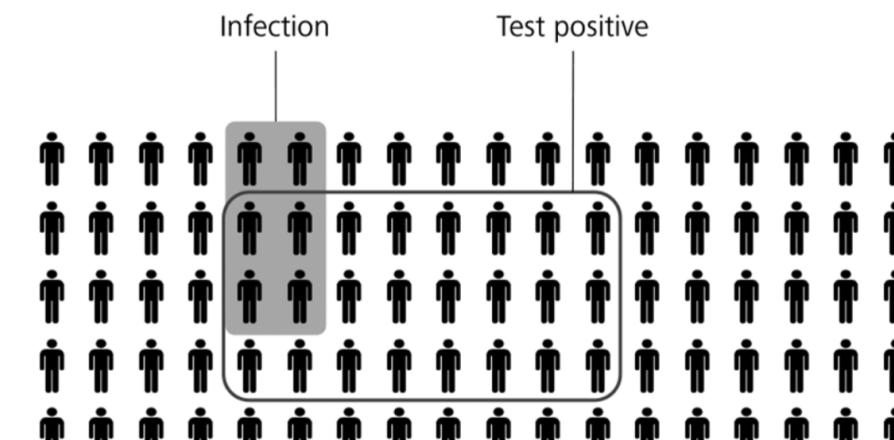


Bus arrival time



(Fernandes et al. 2018)

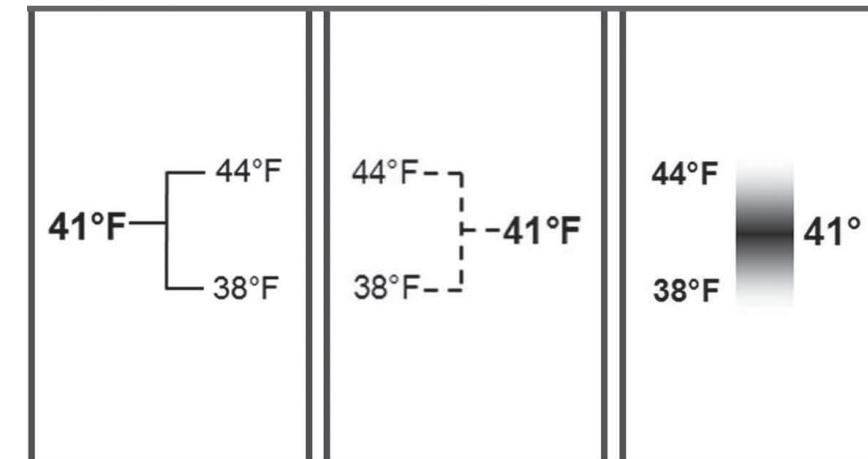
Medical risk communication



(Binder, Krauss, and Bruckmaier 2015)

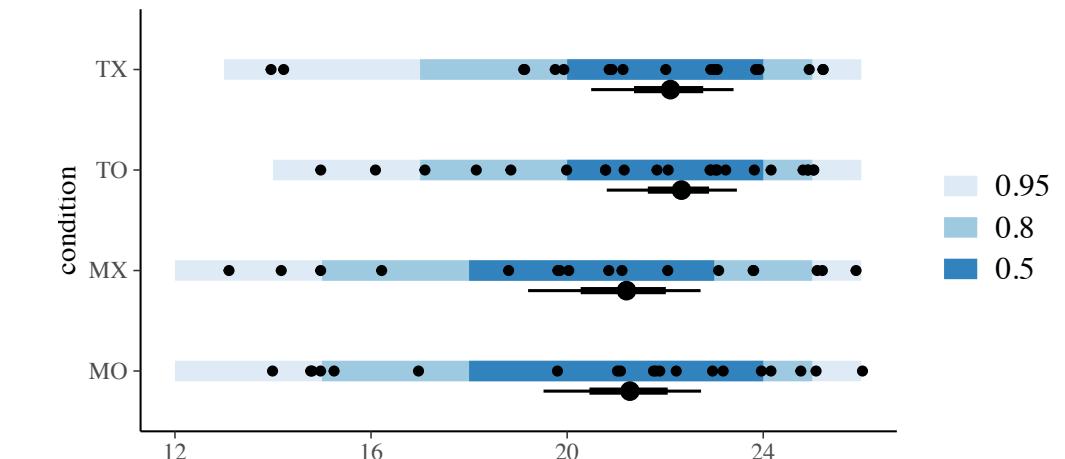
Probabilistic visualizations: same substrate, many domains

Weather forecast

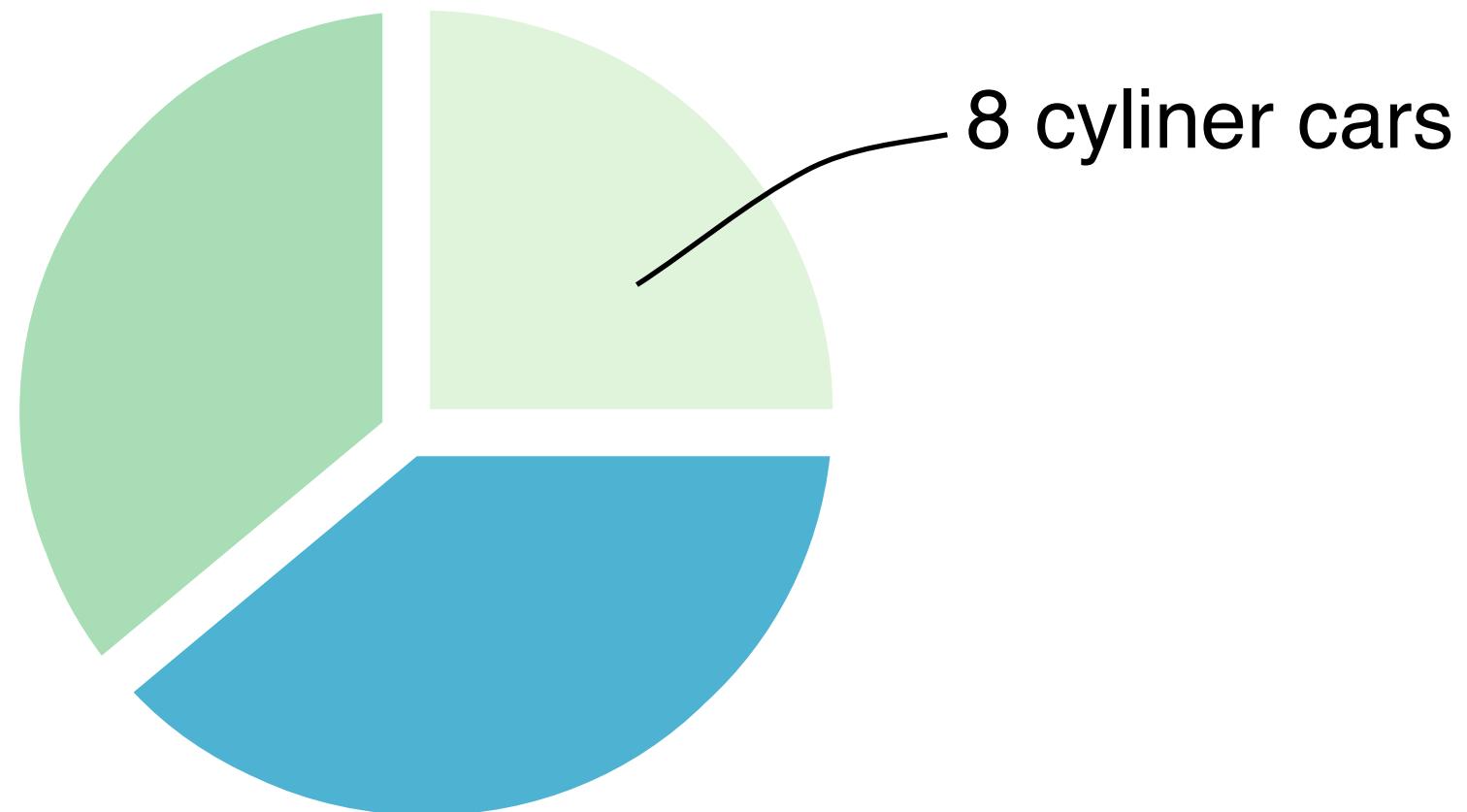


(Joslyn and LeClerc 2013)

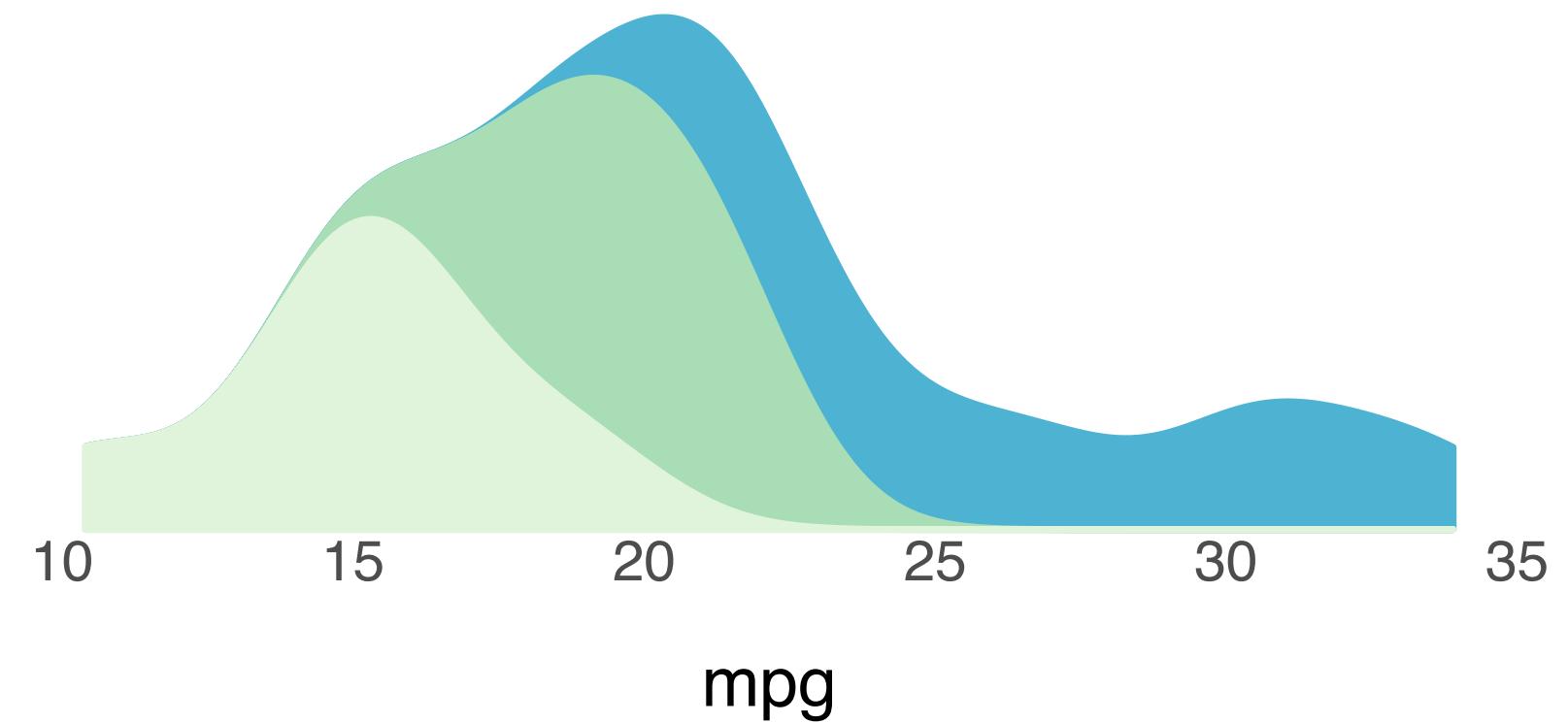
Statistical modeling



What could possibly go wrong?



What could possibly go wrong?



What could possibly go wrong?

	mpg	cyl	am
Mazda RX4	21.0	6	1
Mazda RX4 Wag	21.0	6	1
Datsun 710	22.8	4	1
Hornet 4 Drive	21.4	6	0
Hornet Sportabout	18.7	8	0
Valiant	18.1	6	0

*What's the distribution of mileage
and cylinders?*

--- a user's mental process

What could possibly go wrong?

	mpg	cyl	am
Mazda RX4	21.0	6	1
Mazda RX4 Wag	21.0	6	1
Datsun 710	22.8	4	1
Hornet 4 Drive	21.4	6	0
Hornet Sportabout	18.7	8	0
Valiant	18.1	6	0

```
ggplot(mtcars) +  
  geom_density(aes(  
    x = mpg,
```

What could possibly go wrong?

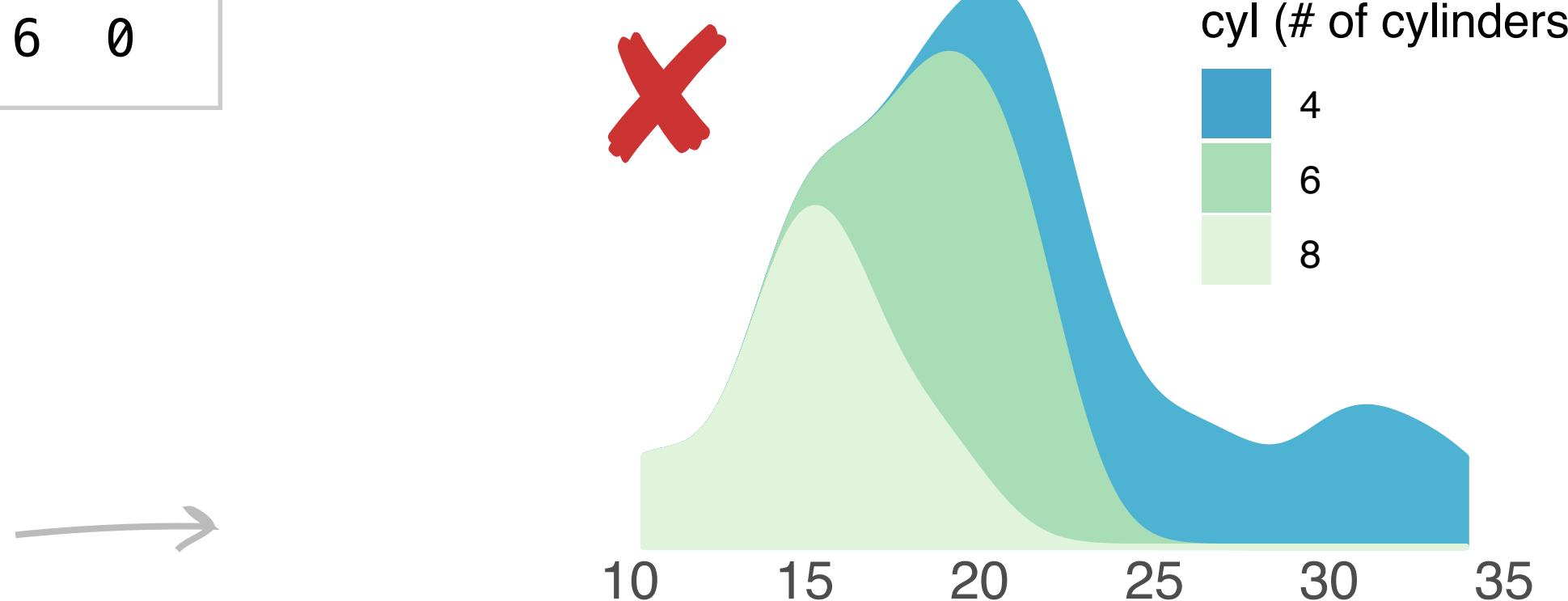
	mpg	cyl	am
Mazda RX4	21.0	6	1
Mazda RX4 Wag	21.0	6	1
Datsun 710	22.8	4	1
Hornet 4 Drive	21.4	6	0
Hornet Sportabout	18.7	8	0
Valiant	18.1	6	0

```
ggplot(mtcars) +  
  geom_density(aes(  
    x = mpg,  
    fill = cyl),  
  position = "stack")
```

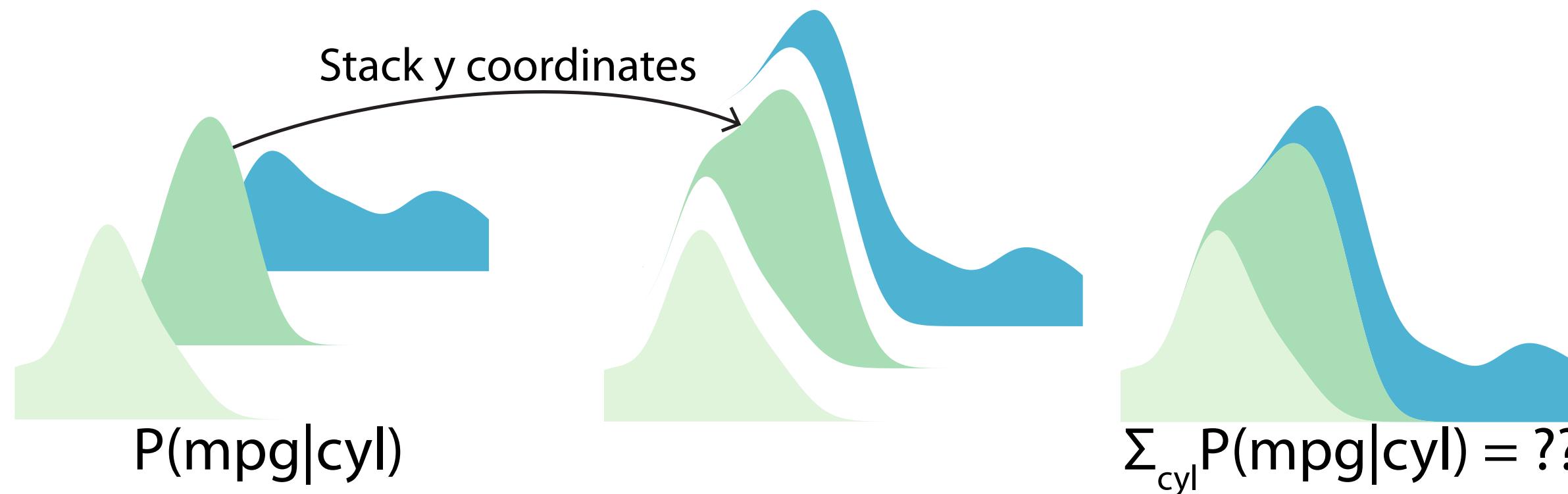
What could possibly go wrong?

	mpg	cyl	am
Mazda RX4	21.0	6	1
Mazda RX4 Wag	21.0	6	1
Datsun 710	22.8	4	1
Hornet 4 Drive	21.4	6	0
Hornet Sportabout	18.7	8	0
Valiant	18.1	6	0

```
ggplot(mtcars) +  
  geom_density(aes(  
    x = mpg,  
    fill = cyl),  
  position = "stack")
```



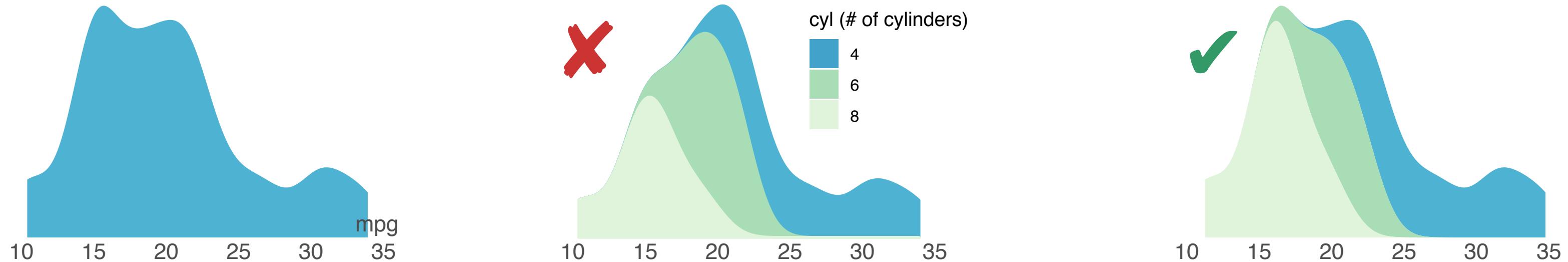
Problem 1: vis shows incorrect probability distribution



```
ggplot(mtcars) +  
  geom_density(aes(  
    x = mpg,  
    fill = cyl),
```

```
ggplot(mtcars) +  
  geom_density(aes(  
    x = mpg,  
    fill = cyl),  
  position = "stack")
```

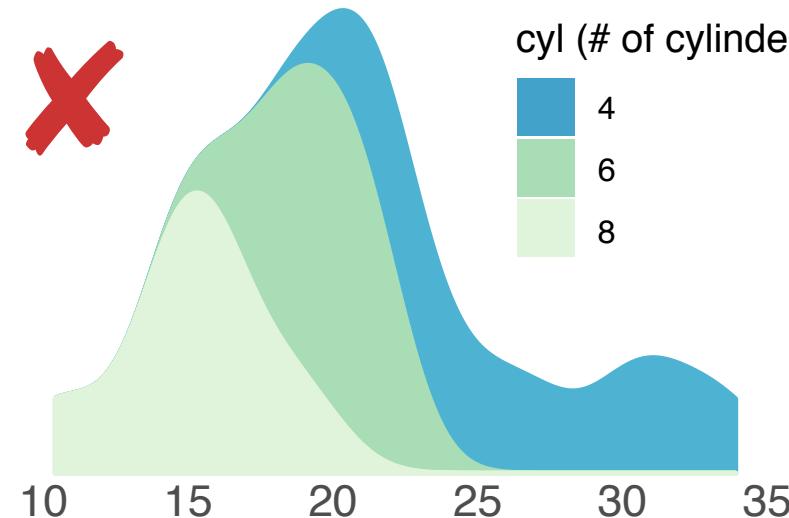
Problem 1: vis shows incorrect probability distribution



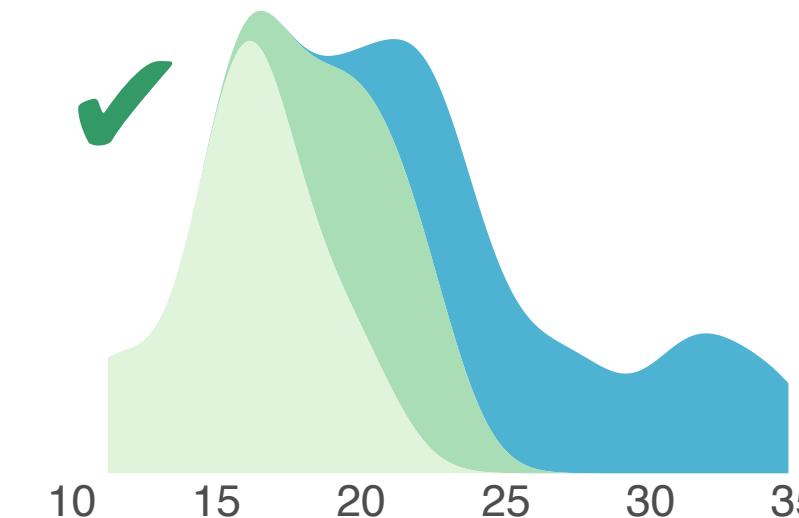
Two incorrect inferences

1. Wrong distribution of # of cylinders $P(\text{cyl})$
2. Wrong overall distribution of mileage $P(\text{mpg})$

Wait we can fix this density plot

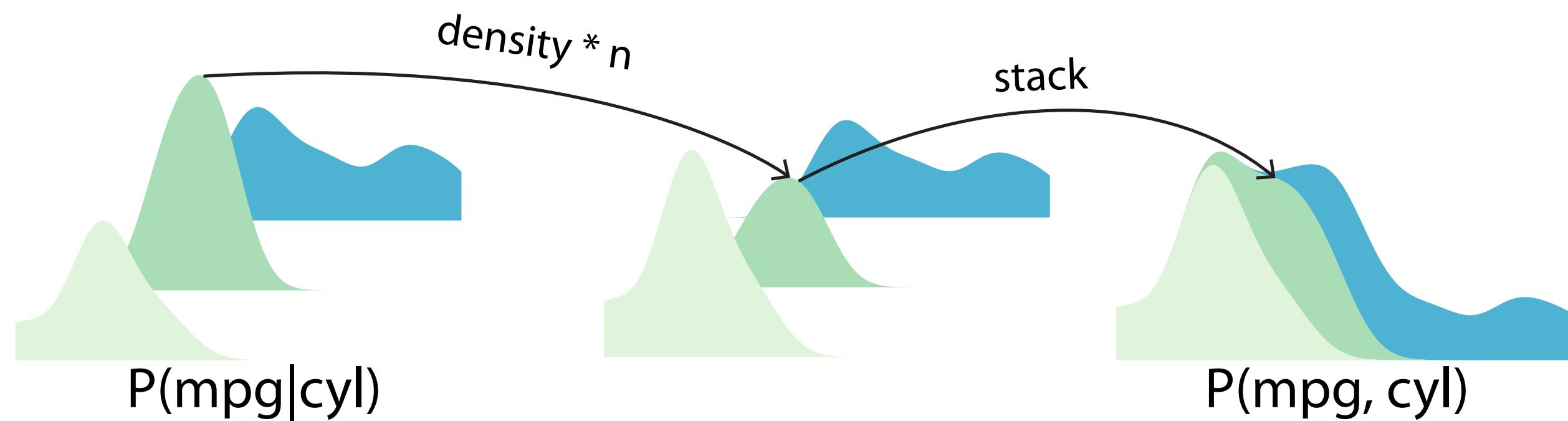


```
ggplot(mtcars) +  
  geom_density(  
    aes(x = mpg,  
        y = stat(density),  
        fill = cyl),  
    position = "stack")
```



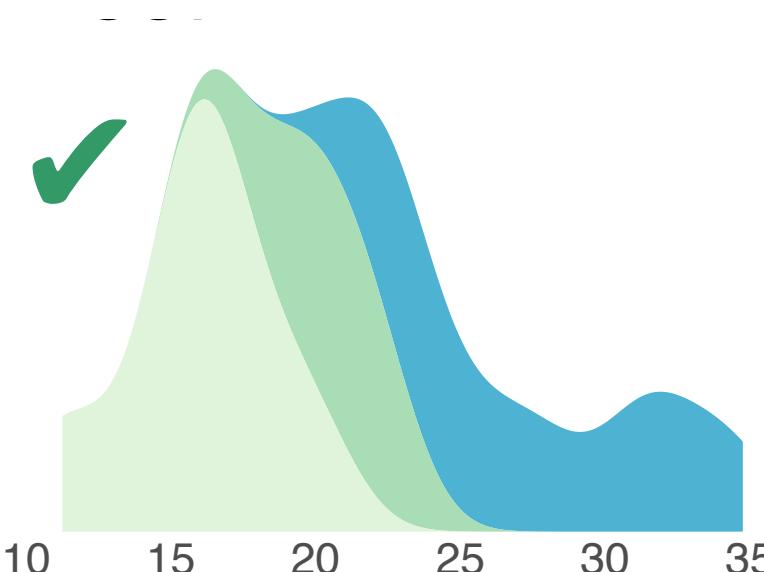
```
ggplot(mtcars) +  
  geom_density(  
    aes(x = mpg,  
        y = stat(density*n),  
        fill = cyl)),  
    position = "stack")
```

Problem 2: specifying probability distributions is convoluted

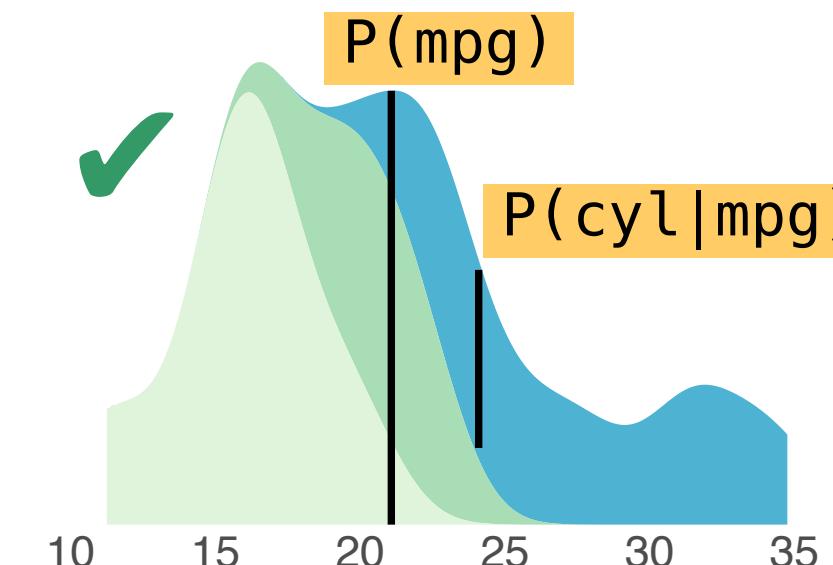


But how I am supposed to know `stat(density*n)` and `position`?

Problem 2 can be solved with ...



```
ggplot(mtcars)+  
  geom_density(  
    aes(x = mpg,  
        y = stat(density)*n),  
    fill = cyl),  
    position = "stack")
```



```
ggplot(mtcars) +  
  geom_bloc(  
    aes(x = mpg,  
        height = P(cyl|mpg) P(mpg),  
        fill = cyl))
```

Details later

PGoG

Given

1. The need to visualize *probability distributions*
2. Specifying probability distributions is convoluted and error-prone

A Probabilistic Grammar of Graphics

- A visualization grammar that makes probability distributions first-class citizens
- Unifies a meaningful set of probabilistic visualizations
- Cognitively ergonomic and guaranteed to be correct

Outline

PGoG in context of

- visualization specification grammar/languages
- formats for communicating probability distributions

Design Requirements
for PGoG

PGoG abstract grammar

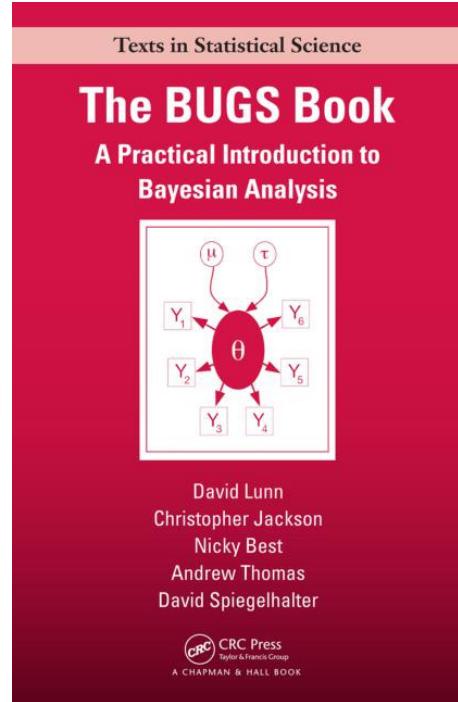
PGoG implementation

Evaluation in terms of

- Expressiveness
- Generativeness
- Cognitive ergonomics

Future: quantitative uncertainty communication!

Related: probabilistic programming



BUGS



Stan

Widely cited and applied
to many domains

```
model {  
    theta ~ beta(1, 1); //prior  
    y ~ bernoulli(theta); //likelihood  
}
```

Sticks with what users know
Avoids implementation details

Related: how to specify a visualization (Grammar of Graphics)

Data +

example_df

A	B	C
1	2	a
2	1	a
3	4	b
4	2	b

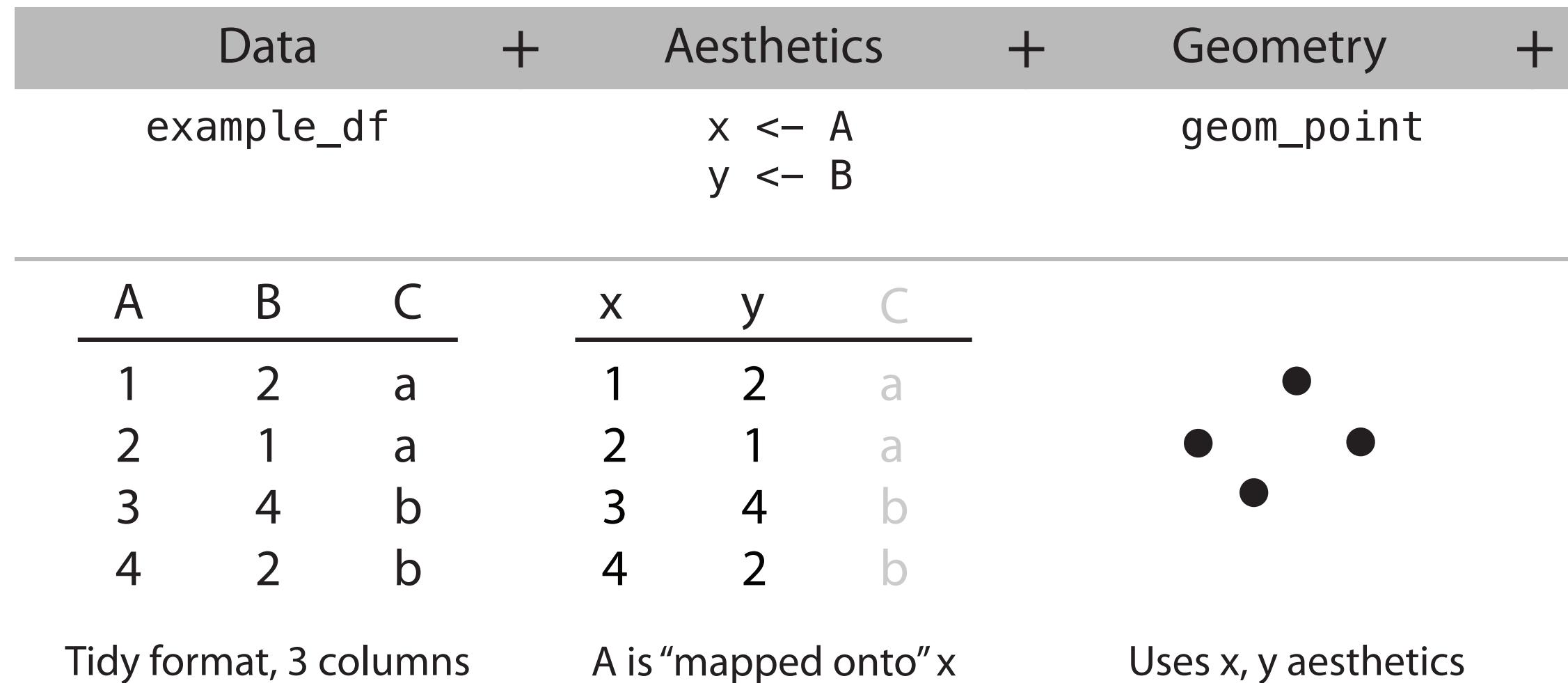
Tidy format, 3 columns

Related: how to specify a visualization (Grammar of Graphics)

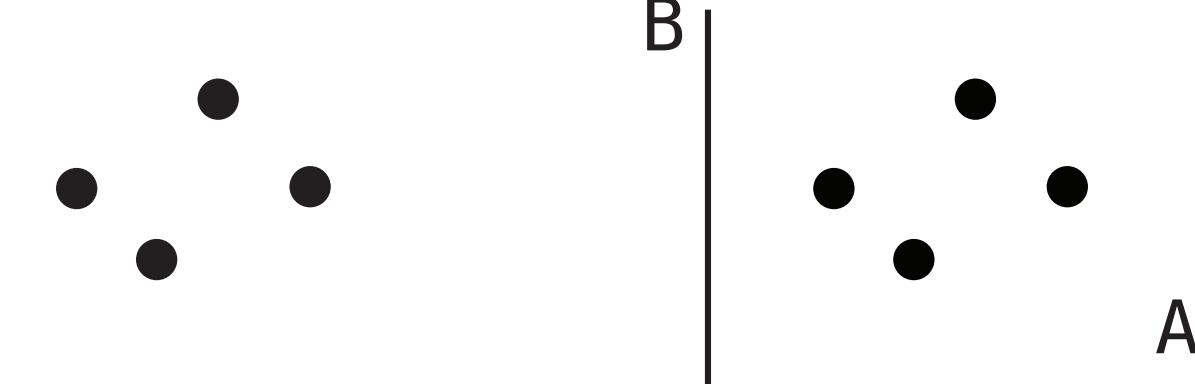
Data	+	Aesthetics	+
example_df		x <- A y <- B	
<hr/>			
A	B	C	x
1	2	a	1
2	1	a	2
3	4	b	3
4	2	b	4
			y
			2
			1
			4
			2
		C	
		a	
		a	
		b	
		b	

Tidy format, 3 columns A is “mapped onto” x

Related: how to specify a visualization (Grammar of Graphics)



Related: how to specify a visualization (Grammar of Graphics)

Data	+	Aesthetics	+	Geometry	+ ... = A plot																													
example_df		x <- A y <- B		geom_point																														
<hr/>																																		
<table><thead><tr><th>A</th><th>B</th><th>C</th></tr></thead><tbody><tr><td>1</td><td>2</td><td>a</td></tr><tr><td>2</td><td>1</td><td>a</td></tr><tr><td>3</td><td>4</td><td>b</td></tr><tr><td>4</td><td>2</td><td>b</td></tr></tbody></table>	A	B	C	1	2	a	2	1	a	3	4	b	4	2	b	<table><thead><tr><th>x</th><th>y</th><th>C</th></tr></thead><tbody><tr><td>1</td><td>2</td><td>a</td></tr><tr><td>2</td><td>1</td><td>a</td></tr><tr><td>3</td><td>4</td><td>b</td></tr><tr><td>4</td><td>2</td><td>b</td></tr></tbody></table>	x	y	C	1	2	a	2	1	a	3	4	b	4	2	b		<hr/>	
A	B	C																																
1	2	a																																
2	1	a																																
3	4	b																																
4	2	b																																
x	y	C																																
1	2	a																																
2	1	a																																
3	4	b																																
4	2	b																																
Tidy format, 3 columns	A is "mapped onto" x	Uses x, y aesthetics	A scatter plot																															

Related: how to specify a visualization (Grammar of Graphics)

Data	+	Aesthetics	+	Geometry	+ ... =	A plot																														
example_df		x <- A y <- B color <- C		geom_point																																
<table><thead><tr><th>A</th><th>B</th><th>C</th></tr></thead><tbody><tr><td>1</td><td>2</td><td>a</td></tr><tr><td>2</td><td>1</td><td>a</td></tr><tr><td>3</td><td>4</td><td>b</td></tr><tr><td>4</td><td>2</td><td>b</td></tr></tbody></table>	A	B	C	1	2	a	2	1	a	3	4	b	4	2	b		<table><thead><tr><th>x</th><th>y</th><th>color</th></tr></thead><tbody><tr><td>1</td><td>2</td><td>a</td></tr><tr><td>2</td><td>1</td><td>a</td></tr><tr><td>3</td><td>4</td><td>b</td></tr><tr><td>4</td><td>2</td><td>b</td></tr></tbody></table>	x	y	color	1	2	a	2	1	a	3	4	b	4	2	b				
A	B	C																																		
1	2	a																																		
2	1	a																																		
3	4	b																																		
4	2	b																																		
x	y	color																																		
1	2	a																																		
2	1	a																																		
3	4	b																																		
4	2	b																																		
Tidy format, 3 columns		A is "mapped onto" x		Uses x, y aesthetics		A scatter plot																														

Related: how to specify a visualization (layout-based is viscous)

(Blackwell et al. 2001)

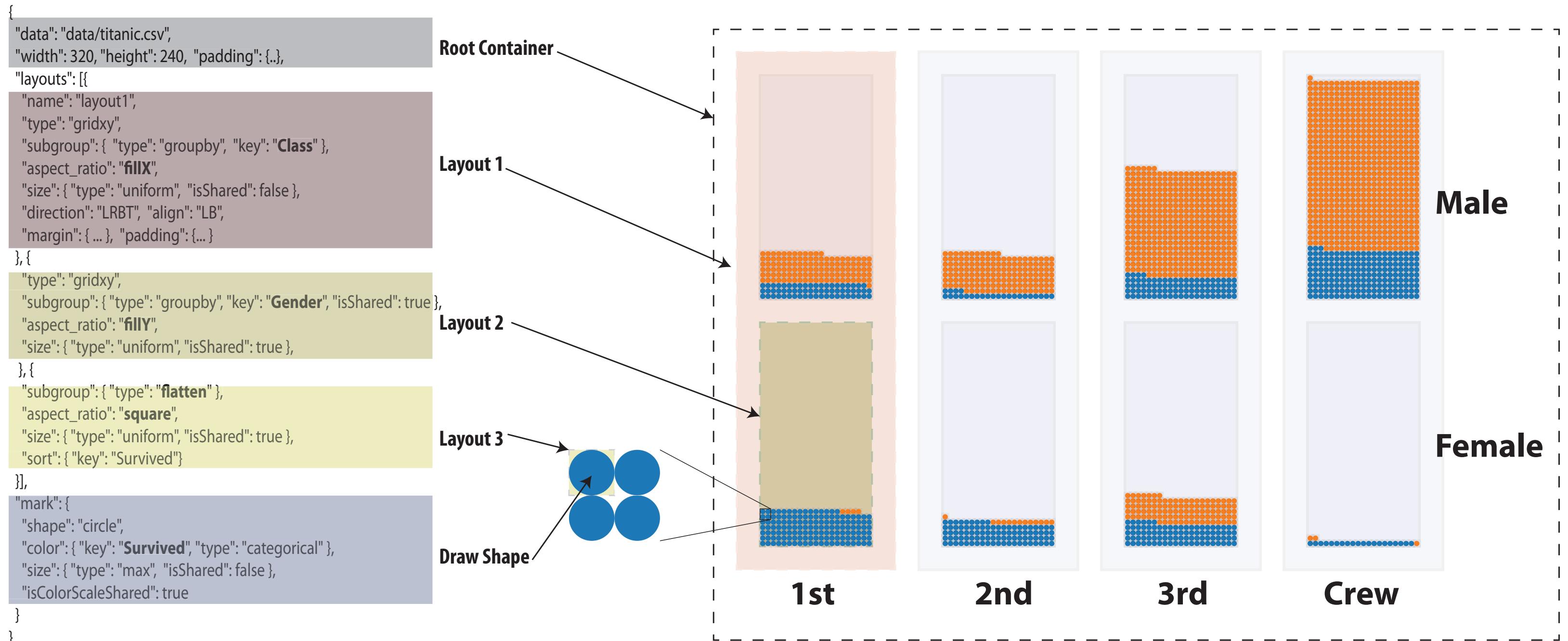
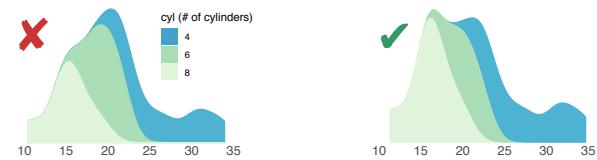
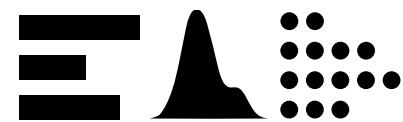


Fig. 6. Example grammar to generate a unit column chart for survivors of the Titanic by passenger class. (Park et al. 2017)

Related: how to specify a visualization in general

Grammar of Graphics



Layout-based



Correct?
Ergonomic?

Constraints-based

```
encoding(e1).  
:- not channel(e1,x).  
:- not field(e1,horsepower).  
:- not bin(e1,_).
```

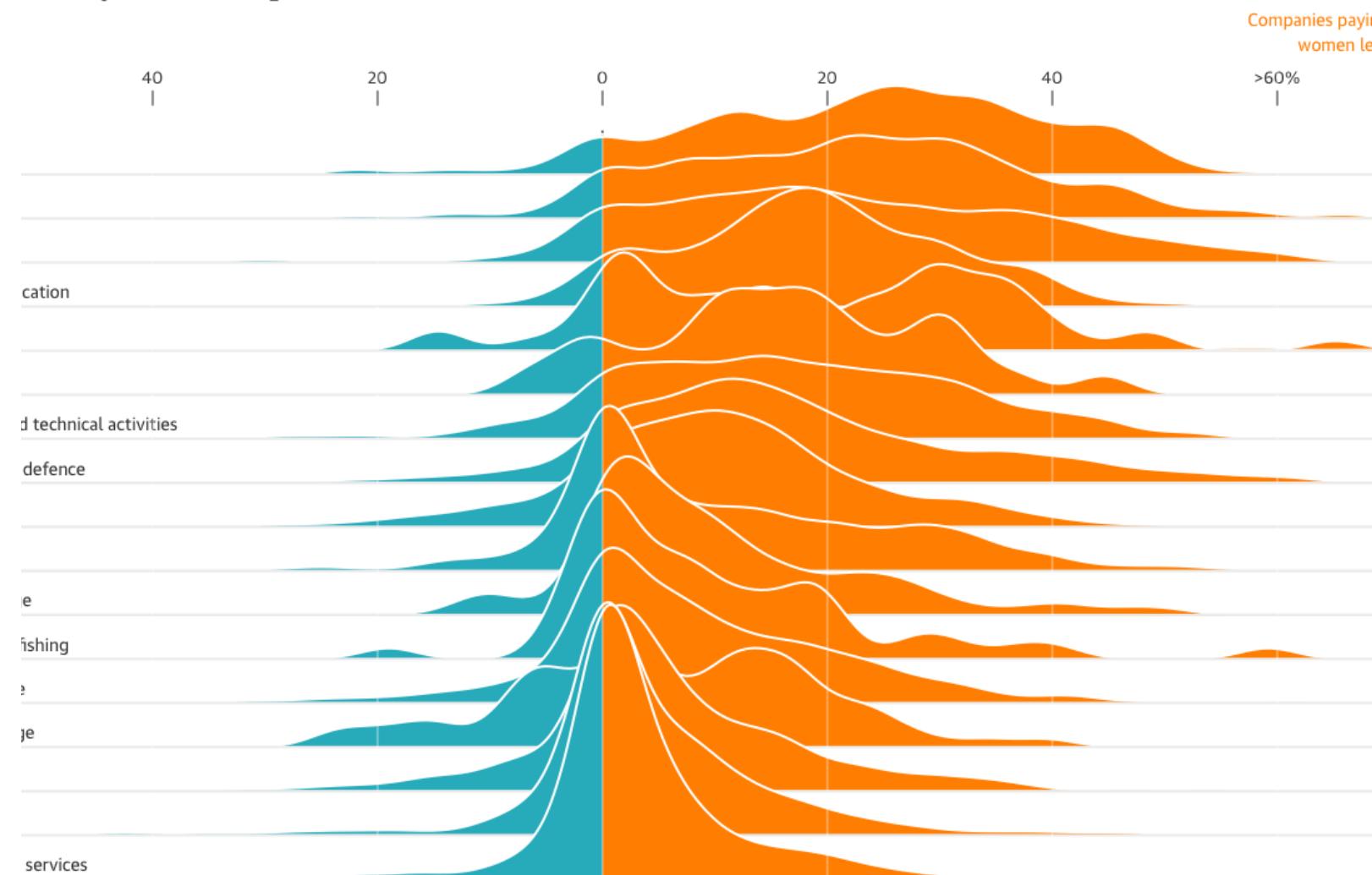
(Moritz et al. 2019)

Need a closer integration
between statistics and
visualization (Heer and Shneiderman 2012)

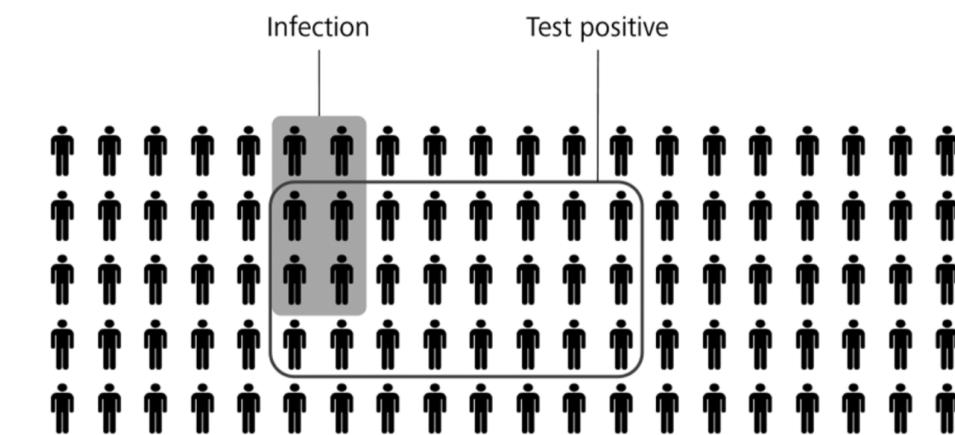
Related: communicating/visualizing uncertainty

Probabilistic visualizations
are often used to communicate uncertainty data

Women are likely to be underpaid in certain sectors



<https://www.theguardian.com/news/ng-interactive/2018/apr/05/women-are-paid-less-than-men-heres-how-to-fix-it>

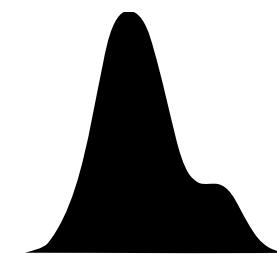


Related: communicating/visualizing uncertainty

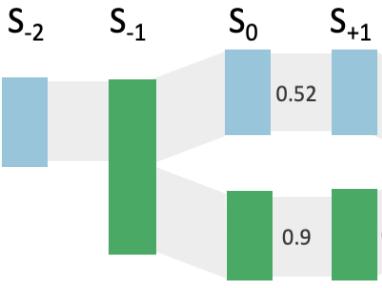
Probabilistic visualizations

are often used to communicate uncertainty data

Probability format X%

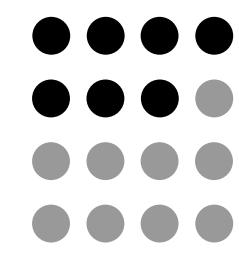
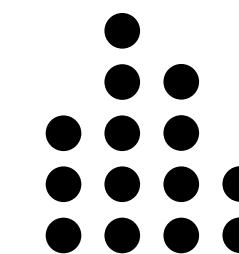


(Gigerenzer and Hoffrage 1995)

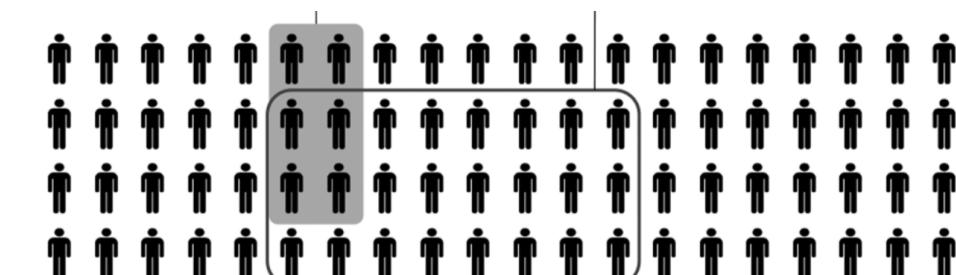


(Guo et al. 2019)

Frequency format X-in-100



(Fernandes et al. 2018)

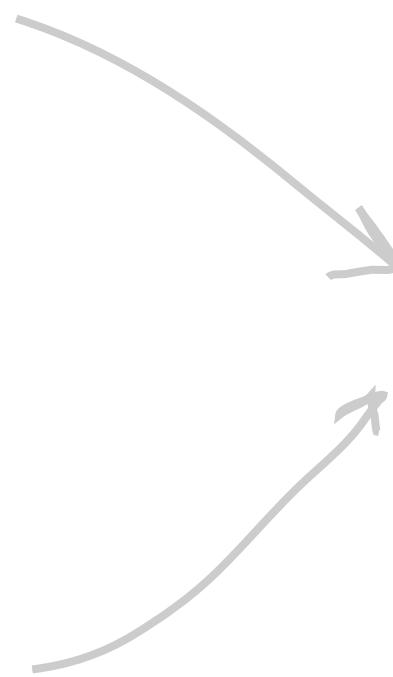


(Binder, Krauss, and Bruckmaier 2015)

Which one to choose?

A closer integration
between statistics and
visualization

A need to support
probability and
frequency formats



Design Requirements for a probabilistic Grammar of Graphics

A closer integration
between statistics and
visualization

- 
- 1 Guaranteeing correctness
of distributions expressed in
visualization
 - 2 Enabling specification close to
probability expressions, such
as $P(A|B)$, which some users
know

Design Requirements for a probabilistic Grammar of Graphics

A need to support
probability and
frequency formats

3 Facilitating exploration *with
coherent and reusable grammar
components*

4 (and automation in the future)

Design Requirements for a probabilistic Grammar of Graphics



The design process

Defaults

Data $\dashrightarrow A$

Aesthetics $\rightarrow x \leftarrow A$

Layer

Data

Aesthetics

Geom $\dashrightarrow \text{geom_bar}$

Stat

Position

Scale



geom_density



geom_points



geom_rect



geom_...

(Wickham 2010)

What is the Probabilistic Grammar of Graphics?

Grammar	ggplot2	PGoG
Defaults		
Data	$\text{Data} \dashrightarrow A$	$P(A B, \dots)$
Aesthetics	$\text{Aesthetics} \dashrightarrow x \leftarrow A$	$\text{height} \leftarrow P(A B, \dots)$
Layer		
Data		
Aesthetics		
Geom	$\text{Geom} \dashrightarrow \text{geom_bar}$	geom_bloc
Stat		
Position		geom_icon
Scale		
Coord	geom_density	
Facet		
	geom_points	
	geom_rect	
	geom_...	

(Wickham 2010)

What is the Probabilistic Grammar of Graphics?

1. The PGoG **grammar** is an extension to *Grammar of Graphics*
2. Probability distributions are first class citizens (data) and other grammar components (aesthetics and geometries) are theoretically informed.

PGoG Grammar/*data*

	mpg	cyl	am
Mazda RX4	21.0	6	1
Mazda RX4 Wag	21.0	6	1
Datsun 710	22.8	4	1
Hornet 4 Drive	21.4	6	0
Hornet Sportabout	18.7	8	0
Valiant	18.1	6	0

Column variable

mpg

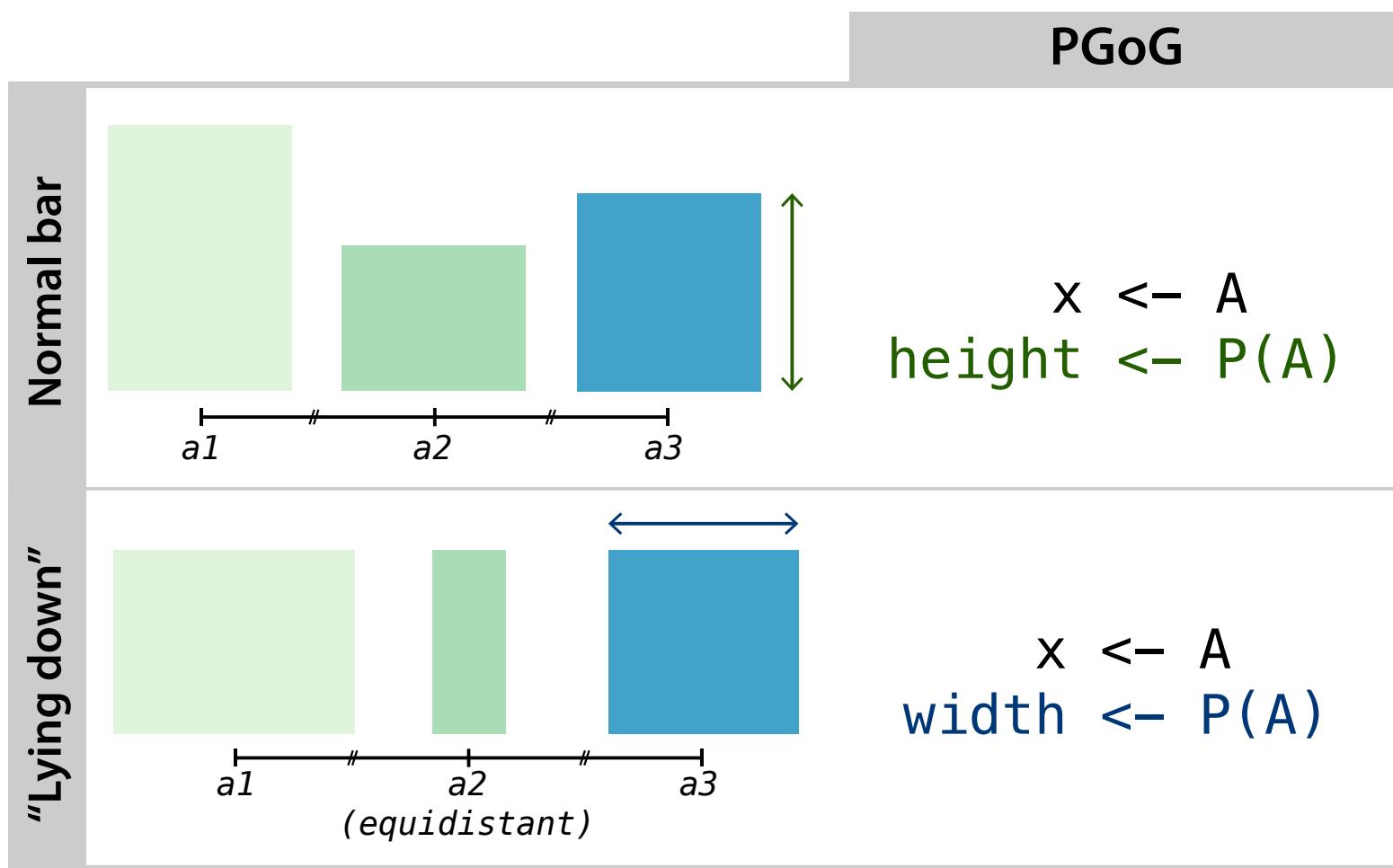
A column in tidy dataset

Probabilistic variable

$P(\text{mpg} | \text{cyl})$

In the form of $P(A...|B...)$, where A, B and ... are variables in columns

PGoG Grammar/aesthetics 1/3

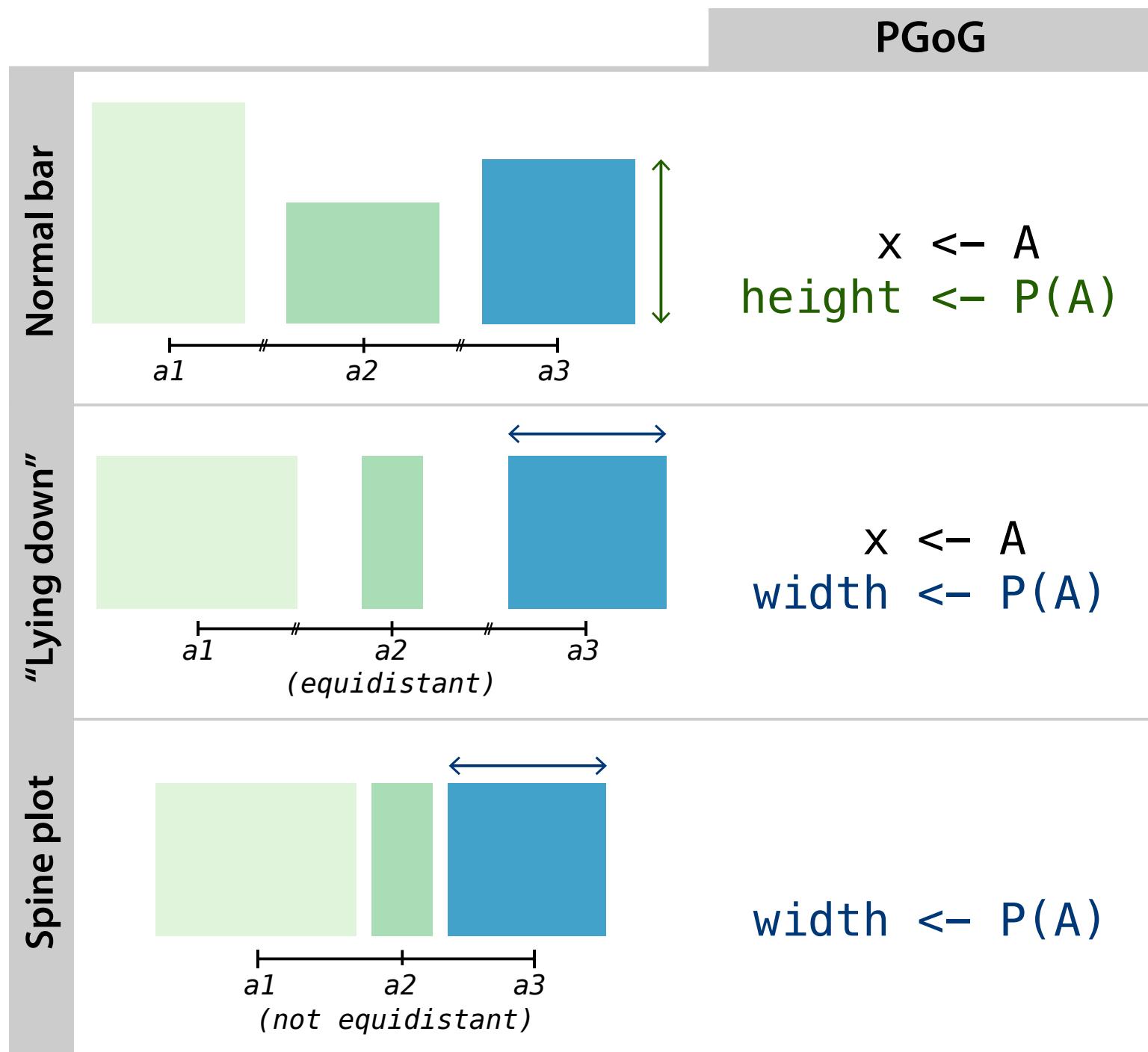


Probabilistic aesthetics

width, height

- Works with probabilistic variables only
- Expresses the probability value by length

PGoG Grammar/aesthetics 2/3



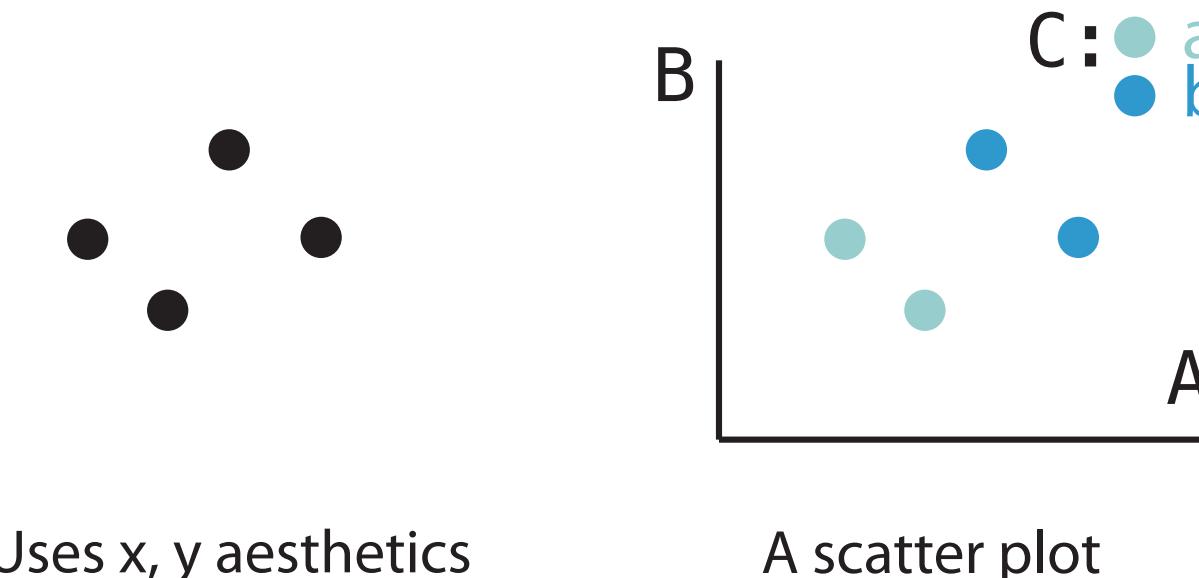
Probabilistic aesthetics

Coordinate aesthetics

X, Y

- For discrete vars: equidistant partitions
- For continuous vars: as one would expect

PGoG Grammar/aesthetics 3/3



Uses x, y aesthetics

A scatter plot

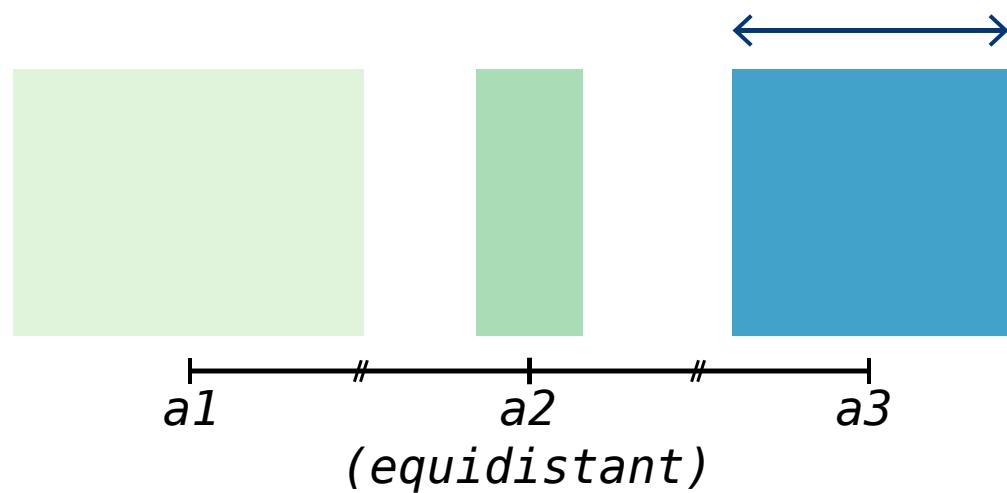
Probabilistic aesthetics

Coordinate aesthetics

Visual aesthetics

fill, color, alpha, ...

PGoG Grammar/Example for conditional

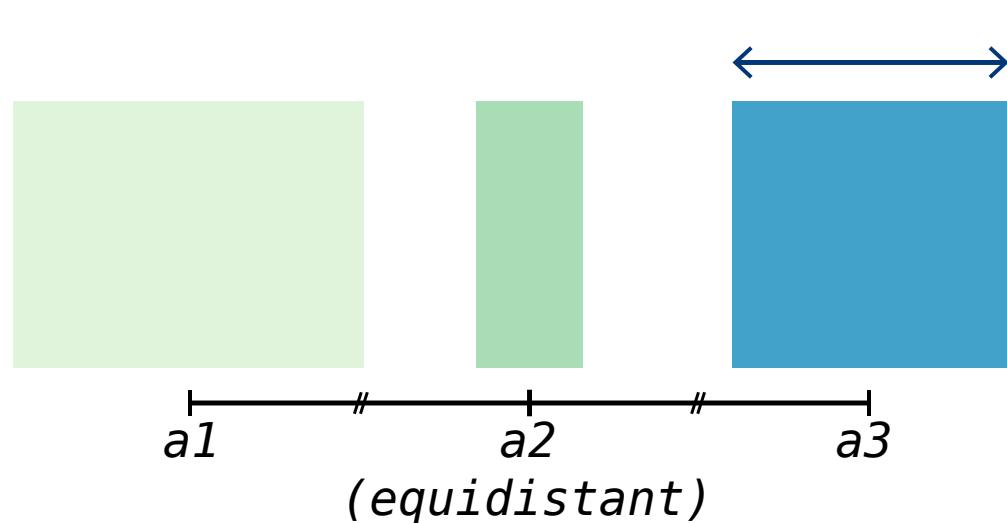


$x \leftarrow A$
 $\text{width} \leftarrow P(A)$

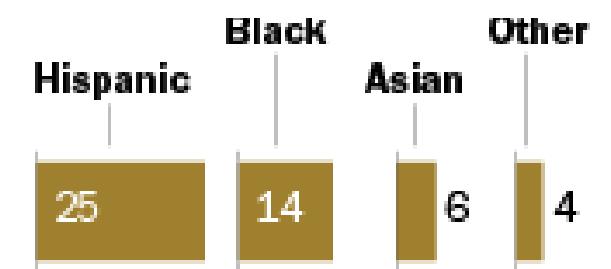
$x \leftarrow \text{race}$

$\text{width} \leftarrow P(\text{race} | \text{generation})$

PGoG Grammar/Example for conditional



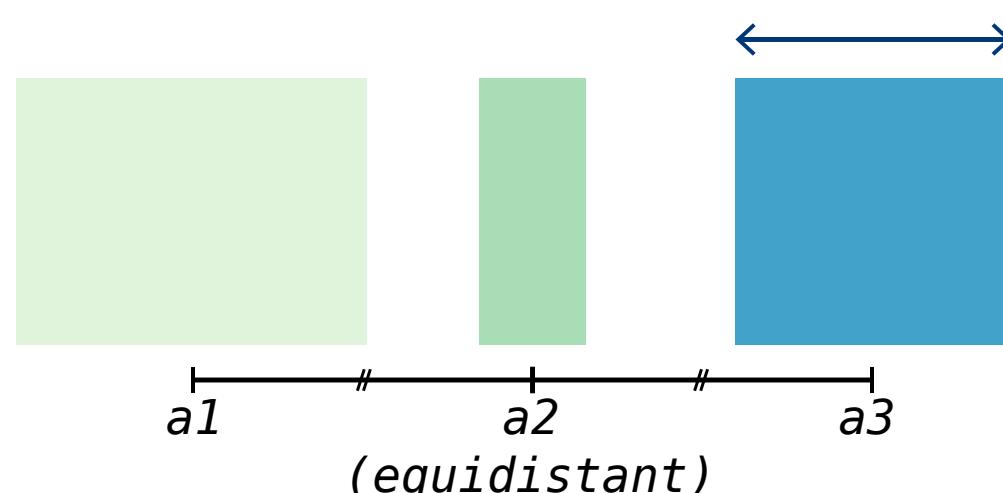
```
x <- A  
width <- P(A)
```



$x <- \text{race}$

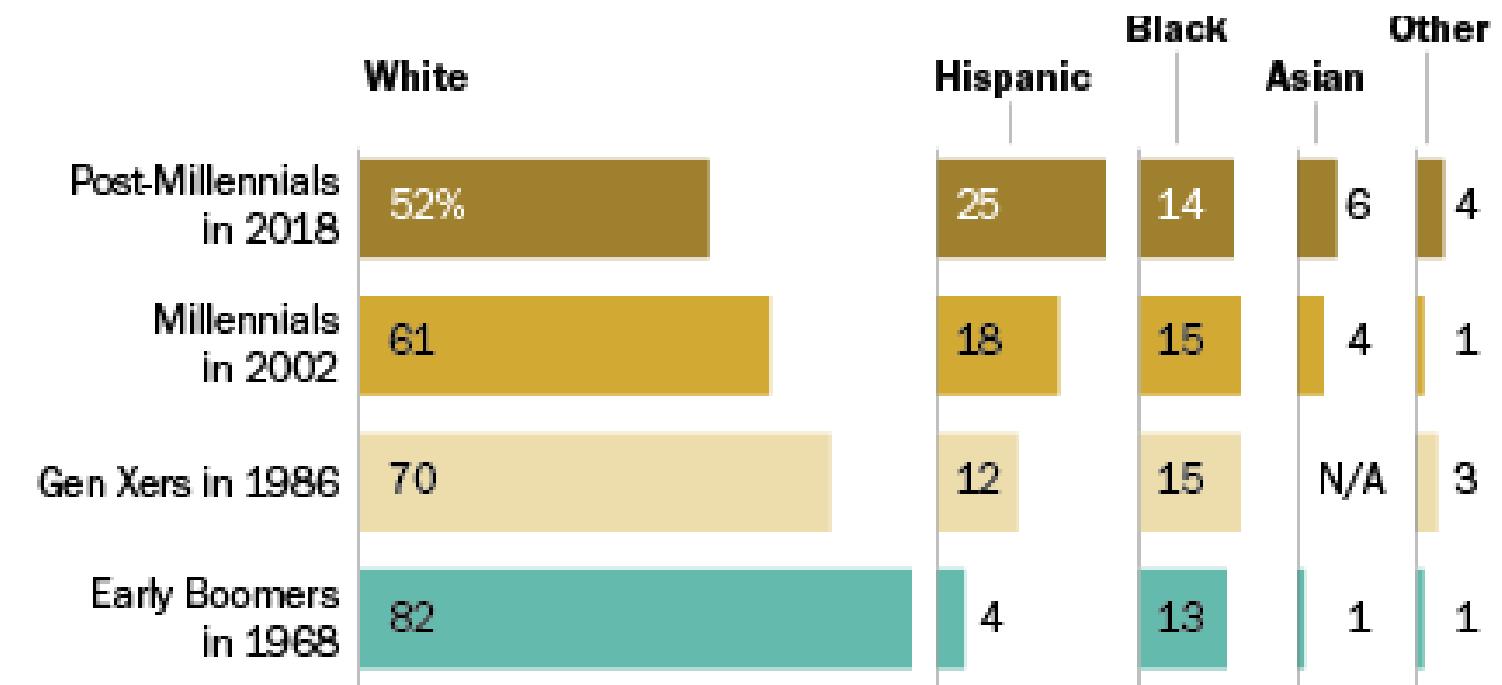
$\text{width} <- \text{P}(\text{race}|\text{generation})$

PGoG Grammar/Example for conditional



$x \leftarrow A$
 $\text{width} \leftarrow P(A)$

<http://www.pewresearch.org/fact-tank/2018/12/13/18-striking-findings-from-2018/>



$x \leftarrow \text{race}$
 $y \leftarrow \text{generation}$
 $\text{width} \leftarrow P(\text{race}|\text{generation})$

PGoG Grammar/*Example for joint*

Math

$$P(\text{mpg} | \text{cyl}) \ P(\text{cyl}) = P(\text{mpg}, \text{cyl})$$

PGoG Grammar/*Example for joint*

Math

$$P(\text{mpg} | \text{cyl}) \ P(\text{cyl}) = P(\text{mpg}, \text{cyl})$$

Coord aes

```
x <- mpg
```

PGoG Grammar/*Example for joint*

Math

$$P(\text{mpg} | \text{cyl}) \ P(\text{cyl}) = P(\text{mpg}, \text{cyl})$$

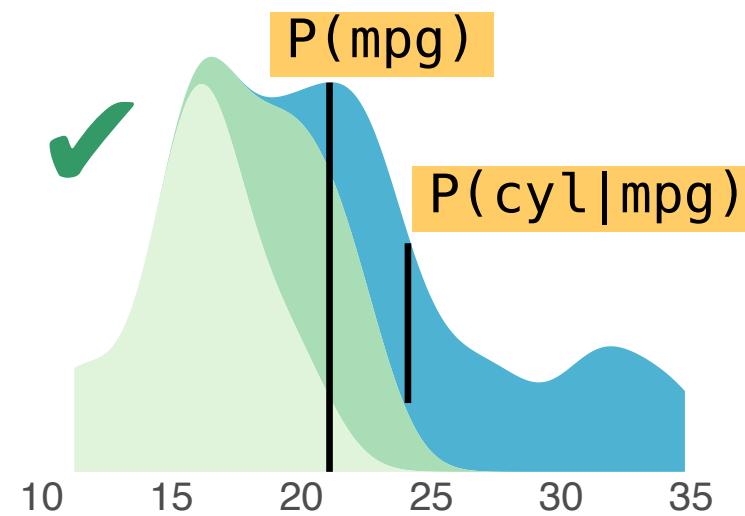
Coord aes

```
x <- mpg
```

Prob aes

```
height <- P(cyl) P(mpg | cyl)
```

PGoG Grammar/Example for joint



```
ggplot(mtcars) +  
  geom_bloc(  
    aes(x = mpg,  
        height = P(cyl|mpg) P(mpg),  
        fill = cyl))
```

Math

Coord aes
Prob aes
Visual aes

$$P(\text{cyl}|\text{mpg}) P(\text{mpg}) = P(\text{mpg}, \text{cyl})$$

`x <- mpg`

`height <- P(mpg) P(cyl|mpg)`

`fill <- cyl`

PGoG Grammar/*checking correctness* 1/2

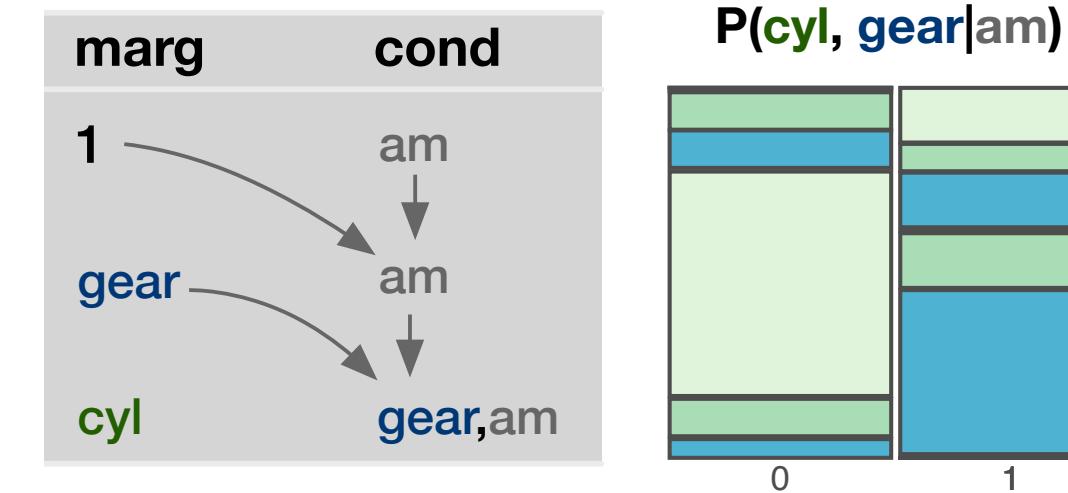
One of the **rules**: the probabilistic variables need to be valid factors of a **probability function**

```
x <- gear  
height <- P(gear|am)  
          P(cyl|gear, am)
```

PGoG Grammar/*checking correctness*

One of the rules: the probabilistic variables need to be valid factors of a **probability function**

$x \leftarrow \text{gear}$
 $\text{height} \leftarrow \begin{cases} P(\text{gear}|\text{am}) \\ P(\text{cyl}|\text{gear}, \text{am}) \end{cases}$



A simple algorithm that sorts and checks probabilistic variables

Grammar

ggplot2

PGoG

Defaults

Data $\dashrightarrow A$

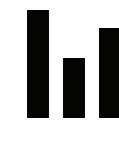
Aesthetics $\dashrightarrow x \leftarrow A$

Layer

Data

Aesthetics

Geom $\dashrightarrow \text{geom_bar}$



Stat
Position

Scale



Coord

Facet

geom_points



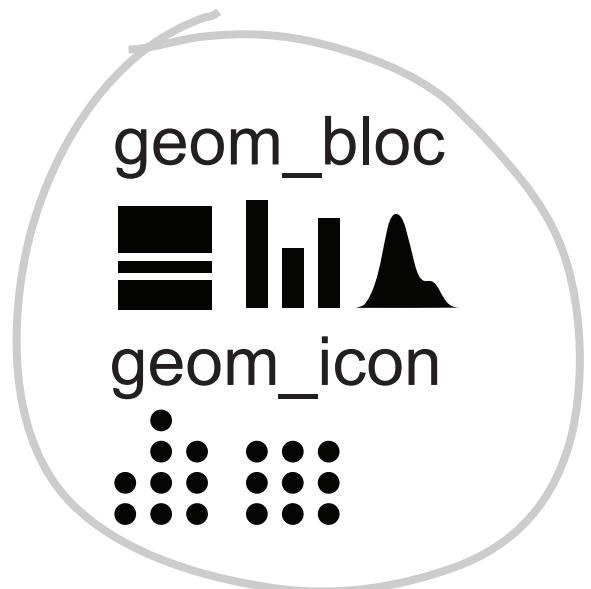
geom_rect



geom_...

$P(A|B, \dots)$

height $\leftarrow P(A|B, \dots)$



PGoG Grammar/ *geometries 1/2*

ggplot2

geom_bar



geom_mosaic*



geom_density



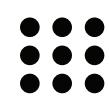
geom_violin



geom_density_ridges*



geom_waffle*



PGoG

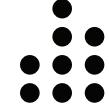
geom_bloc

Look at all those geometries
in **ggplot2** we have
replaced

Also, probability and
frequency formats

geom_icon

geom_dotplot



* **ggplot2** extensions

PGoG Grammar/*geometries*

`geom_bloc`: recursive
subpartition to support
many probabilistic vari-
ables



`geom_icon` needs a
new way to pack icons

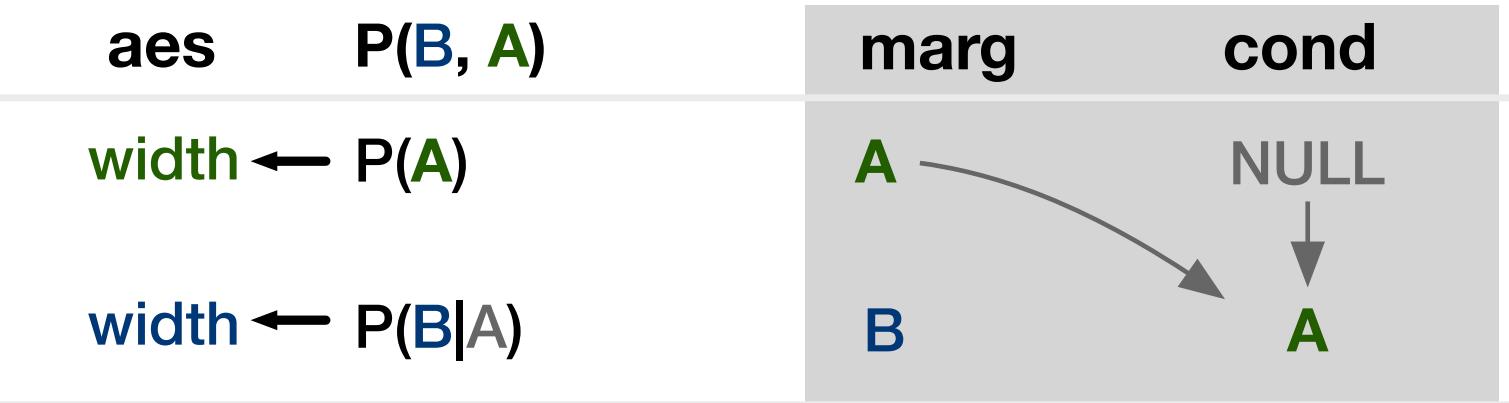
```
y <- A  
width <- P(A) P(B|A)  
fill <- B
```

PGoG Grammar/*geometries*

`geom_bloc`: recursive
subpartition to support
many probabilistic vari-
ables

`geom_icon` needs a
new way to pack icons

```
y <- A  
width <- P(A) P(B|A)  
fill <- B
```

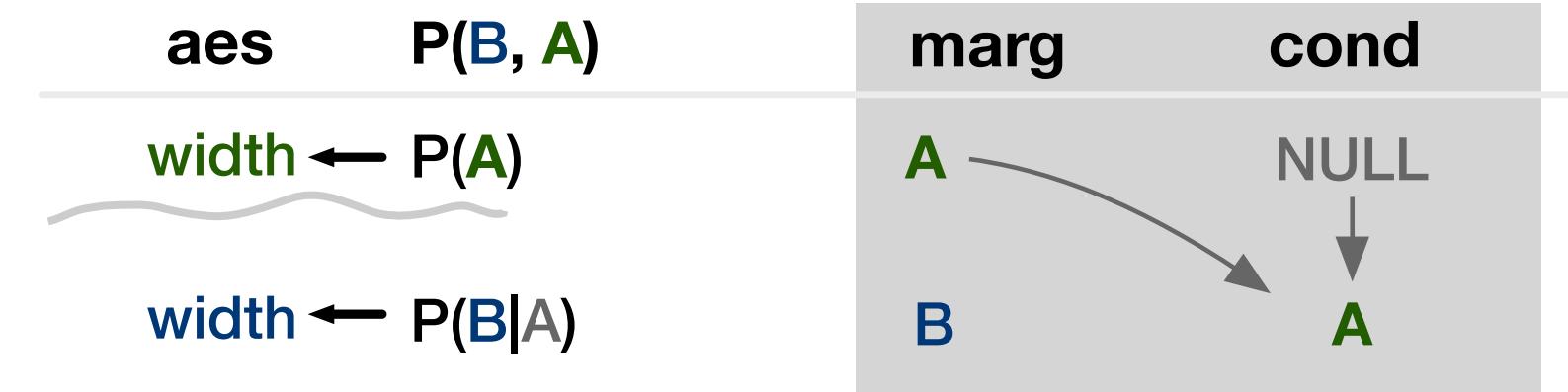
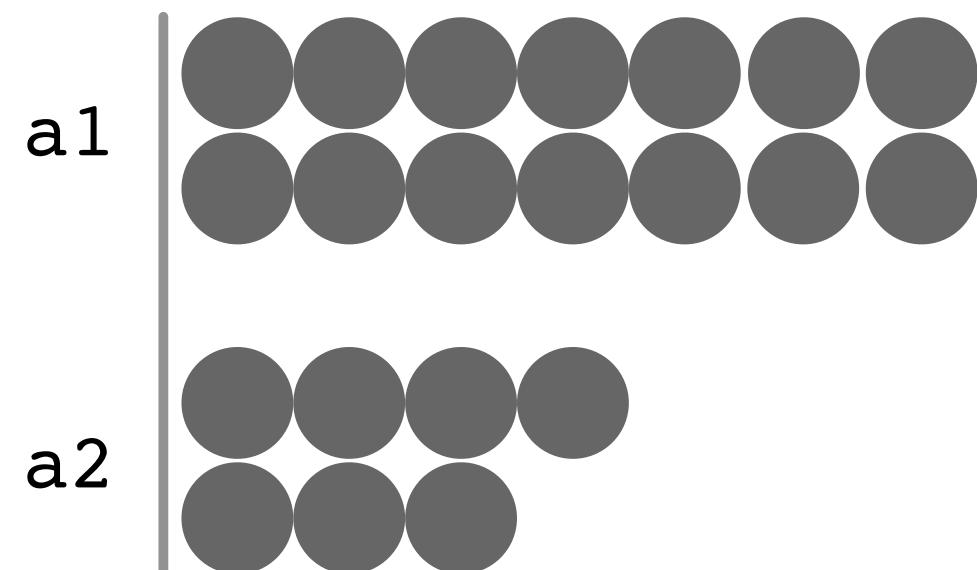


Probability structure (the “chain”) deter-
mines the visualization structure

PGoG Grammar/*geometries*

`geom_icon` needs a new way to pack icons

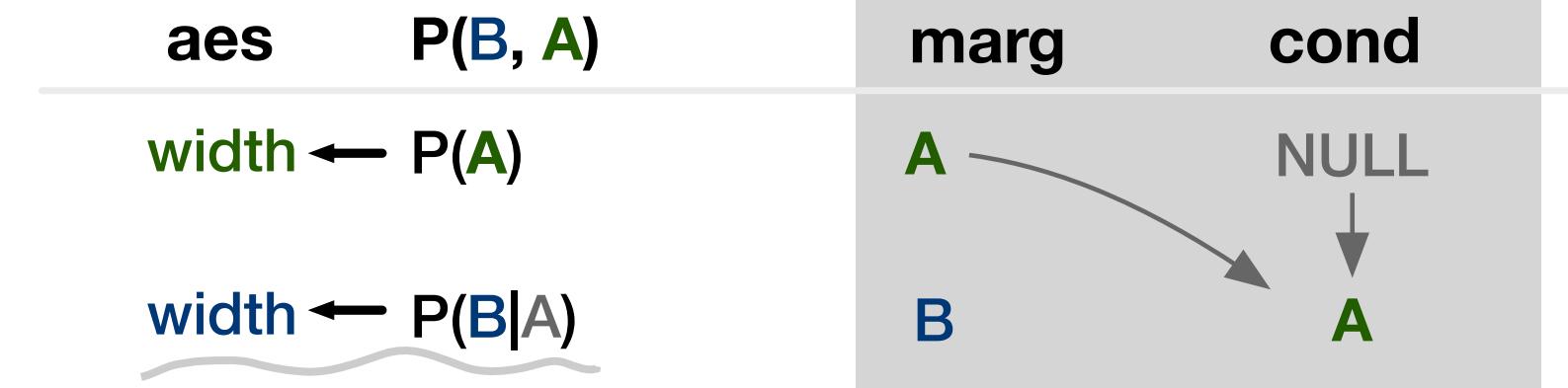
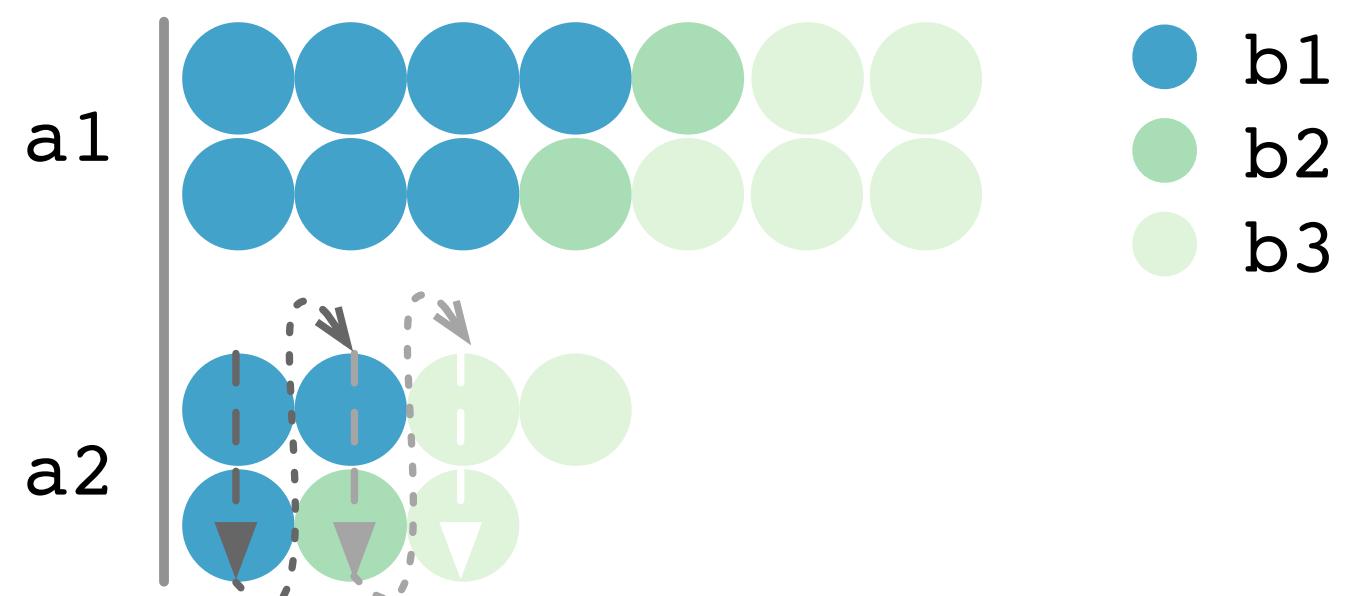
```
y <- A  
width <- P(A) P(B|A)  
fill <- B
```



PGoG Grammar/*geometries*

`geom_icon` needs a new way to pack icons

```
y <- A  
width <- P(A) P(B|A)  
fill <- B
```



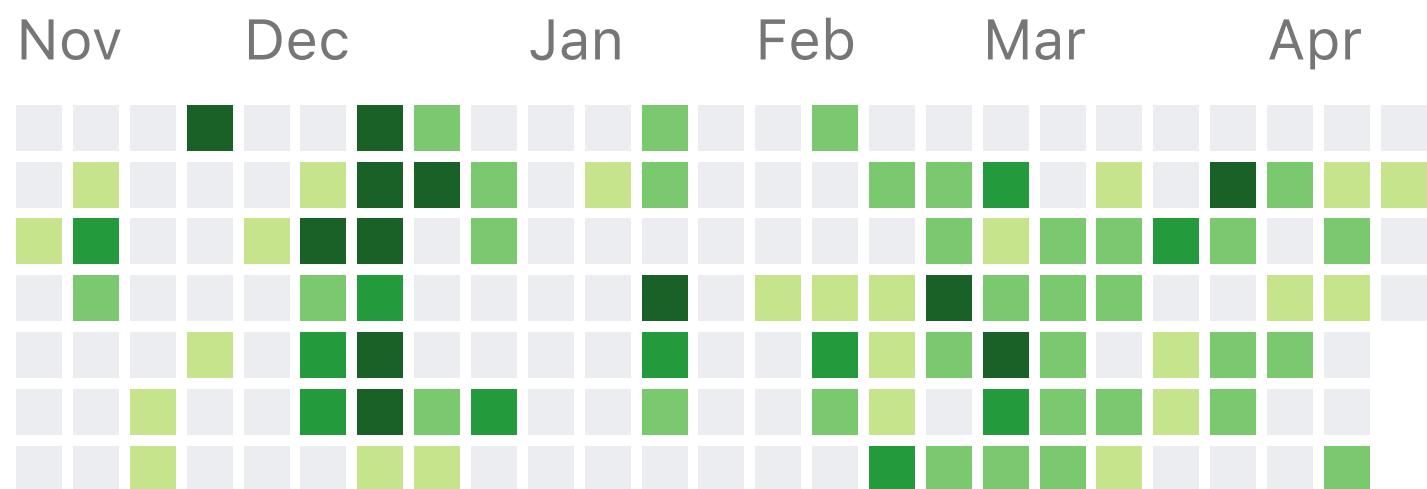
Implementing the Grammar

Why in R

- Grammar of graphics is implemented in ggplot2
- Metaprogramming features in R helps parsing
- PGoG grammar is transferable

Current progress

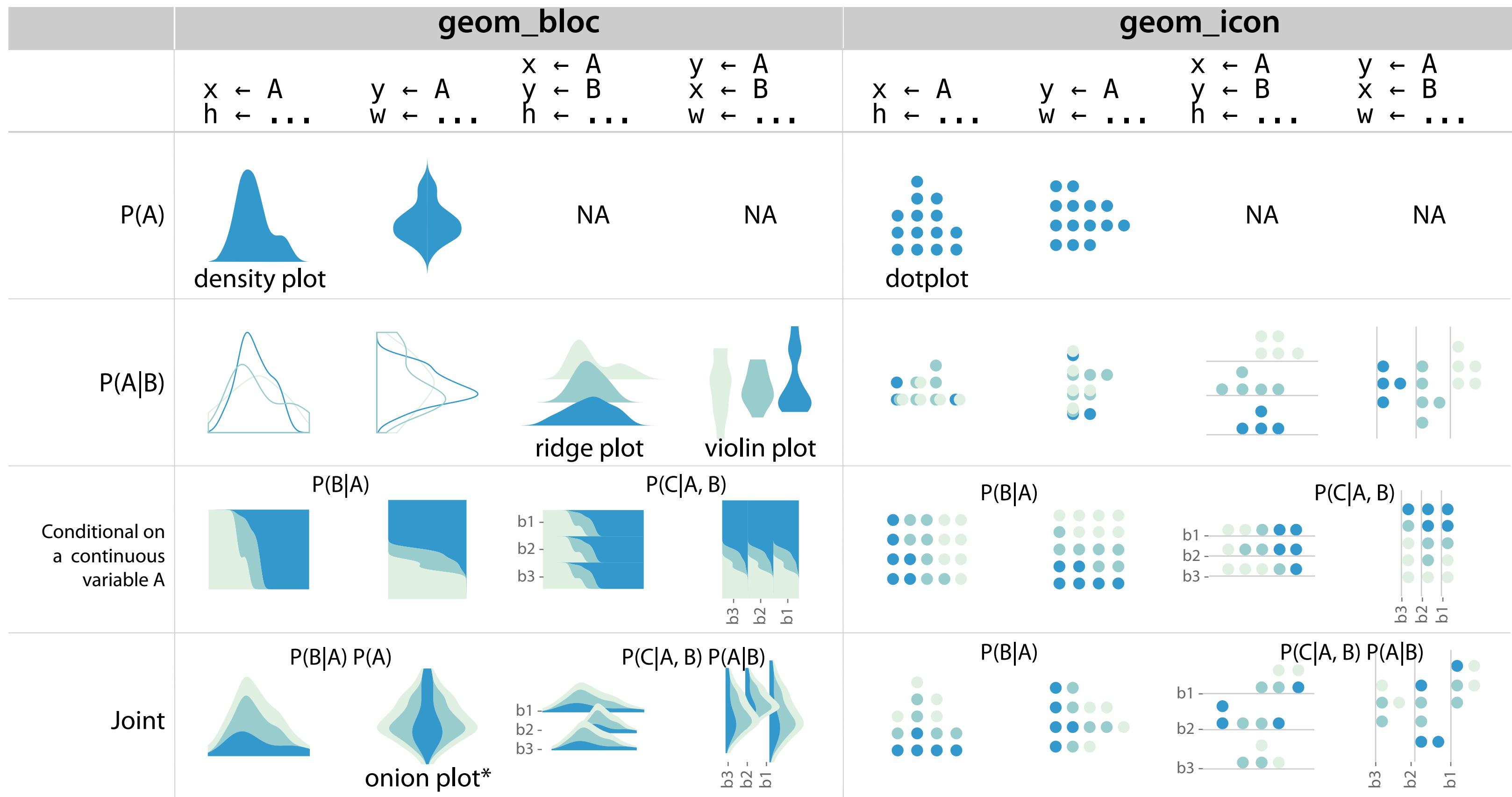
- geom_bloc with discrete variables
- geom_bloc with up to two continuous variables
- geom_icon with up to two variables
- TODO: coloring, extra parameters



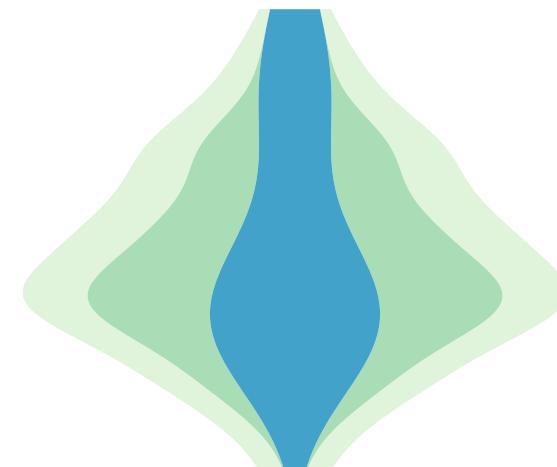
Evaluation of the Grammar

- Expressive?
- Generative?
- Cognitively ergonomic?

Expressiveness of the grammar



Generativeness from the combination of aesthetics



Onion plot

`geom_bloc:`

`y ← mpg`

`width ← P(mpg) P(cyl|mpg)`

`direction ← both`

Cognitive ergonomics

(Blackwell et al. 2001)

Pro:

Short edit distances (Kim et al. 2017)

- *Low viscosity*
- *No premature commitment*

Close to probability expressions

Con:

Specifying probability expressions can be difficult

- *Hidden dependencies*
- *Error prone-ness*

Cognitive ergonomics

Pro: Short edit distances, close to probability expressions

- *Low viscosity*
- *No premature commitment*

Existing ggplot2 packages

Changes

Syntax



```
geom_mosaic
  x = cyl,
  mpg*
divider = hspine,
  hspine
```

Cognitive ergonomics

Pro: Short edit distances, close to probability expressions

- *Low viscosity*
- *No premature commitment*

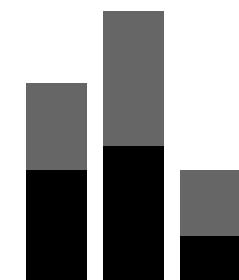
Existing ggplot2 packages

Changes

Syntax



geom:mosaic→bar
+fill +y
-divider



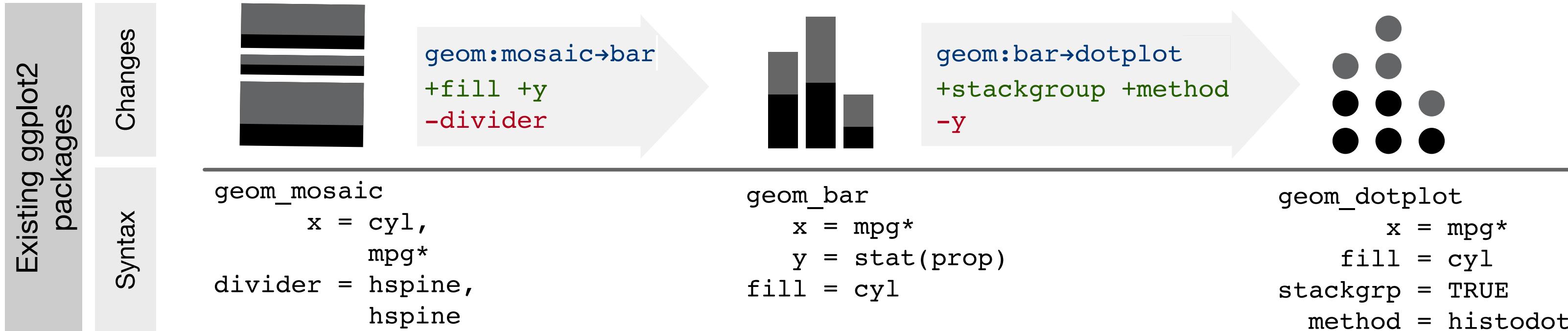
```
geom_mosaic
  x = cyl,
  mpg*
  divider = hspine,
  hspine
```

```
geom_bar
  x = mpg*
  y = stat(prop)
  fill = cyl
```

Cognitive ergonomics

Pro: Short edit distances, close to probability expressions

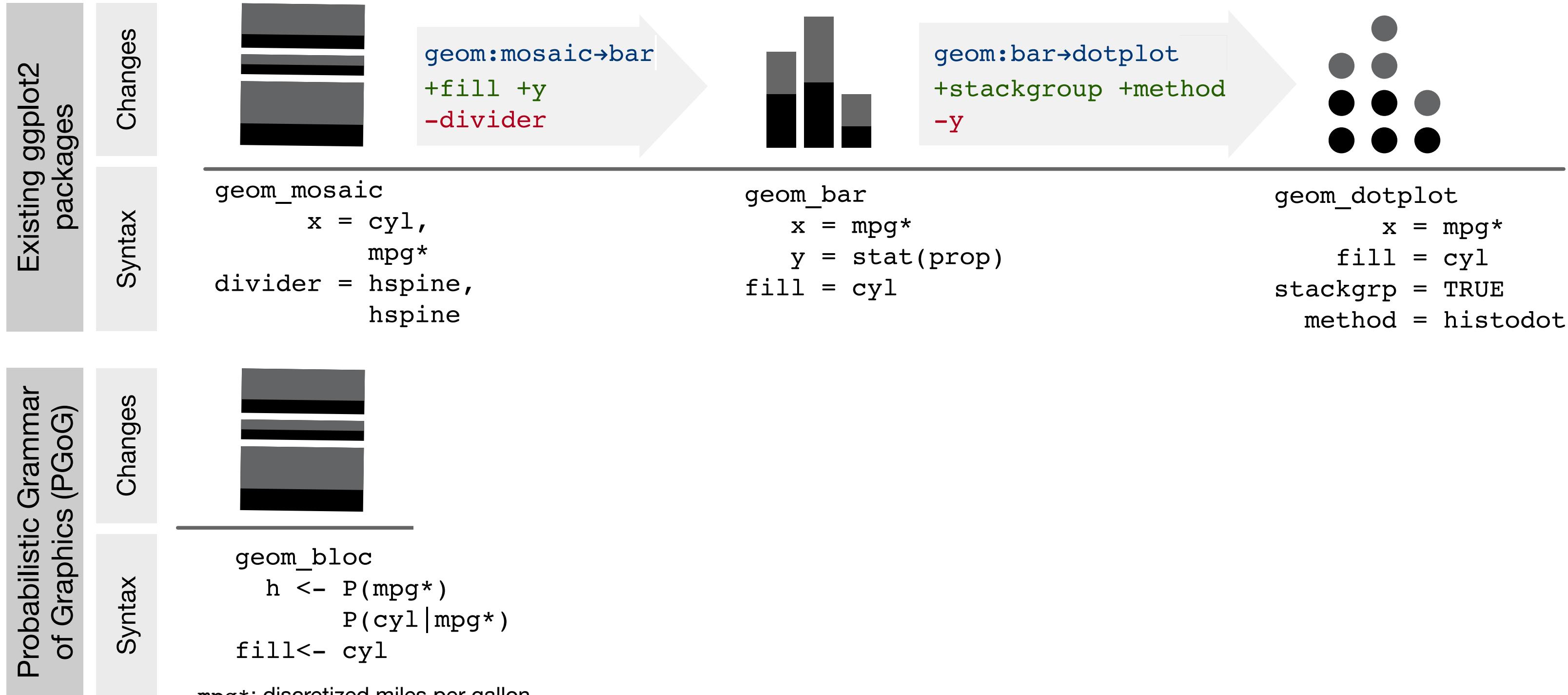
- *Low viscosity*
- *No premature commitment*



Cognitive ergonomics

Pro: Short edit distances, close to probability expressions

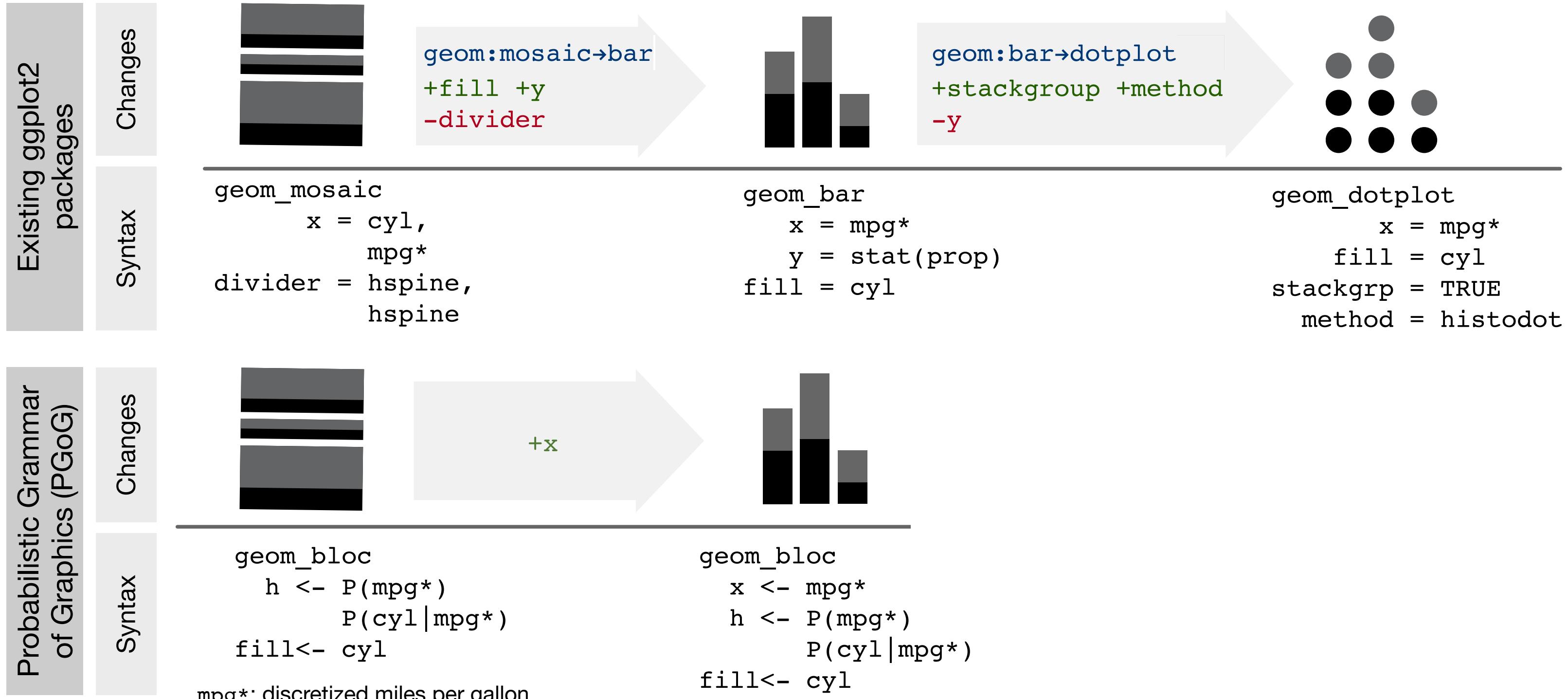
- *Low viscosity*
- *No premature commitment*



Cognitive ergonomics

Pro: Short edit distances, close to probability expressions

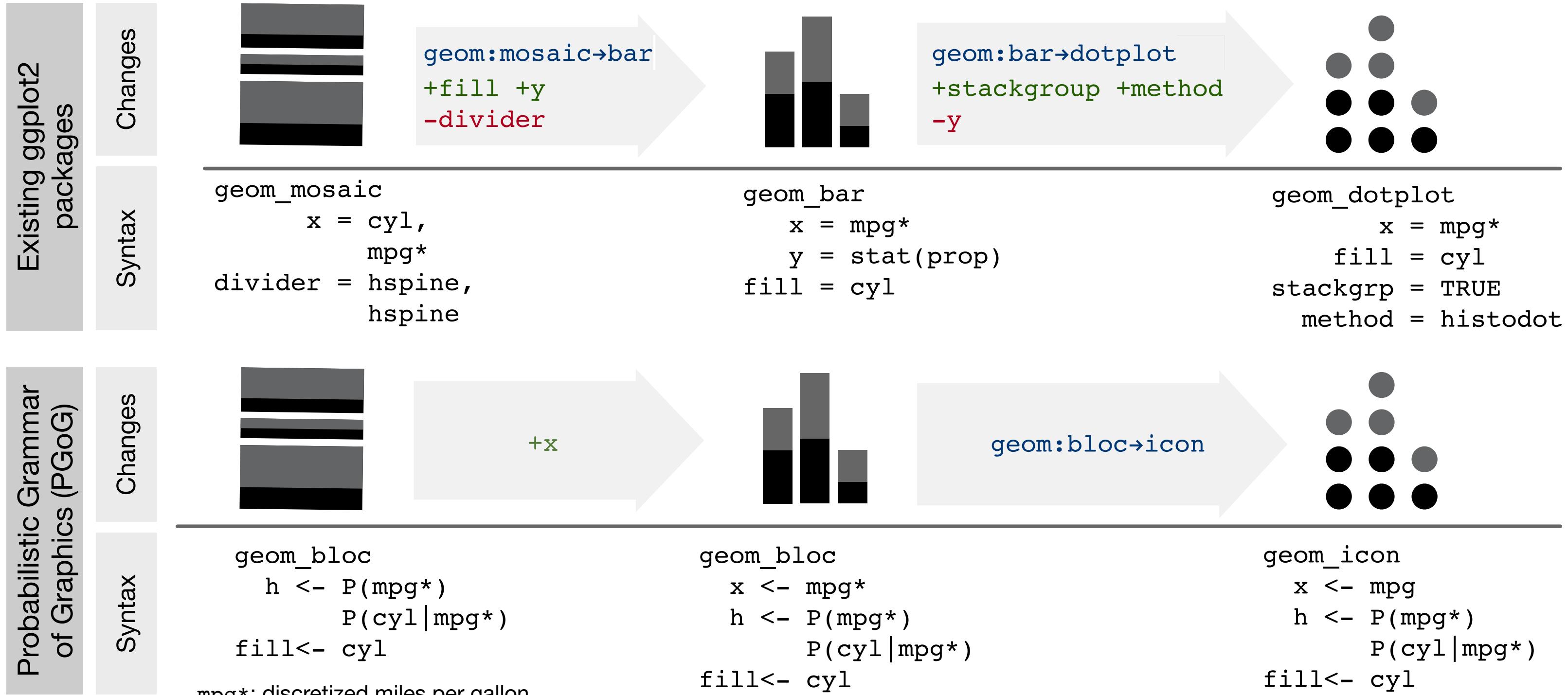
- *Low viscosity*
- *No premature commitment*



Cognitive ergonomics

Pro: Short edit distances, close to probability expressions

- *Low viscosity*
- *No premature commitment*



Cognitive ergonomics

Con: specifying probability expressions can be difficult

- *Error prone-ness*
- *Hidden dependencies*

Math

$$P(\text{mpg} | \text{cyl}) \quad P(\text{mpg}) ?$$



$$\mathbf{P}(\text{mpg} | \text{cyl}) \quad \mathbf{P}(\text{cyl}) = \mathbf{P}(\text{mpg}, \text{cyl})$$

Cognitive ergonomics

Con: specifying probability expressions can be difficult

- *Error prone-ness*
- *Hidden dependencies*

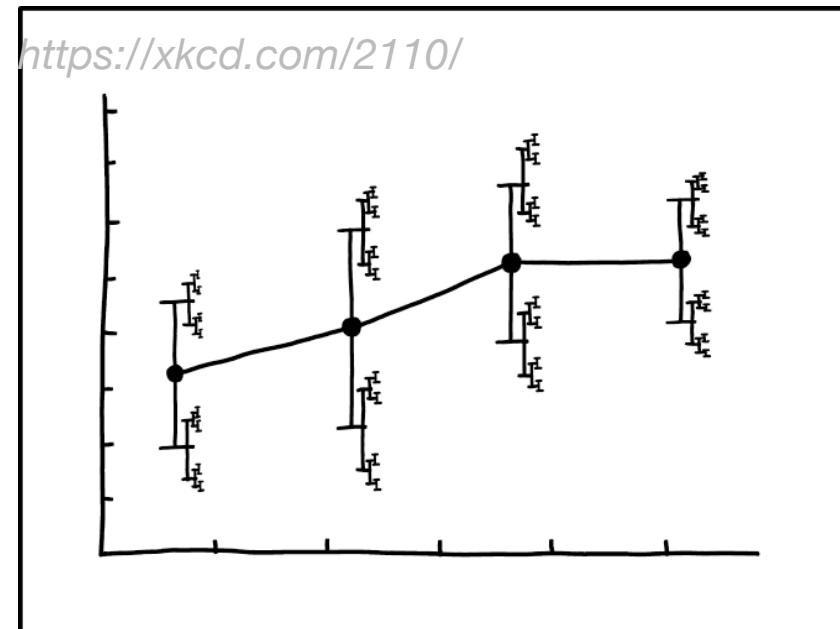
	$P(\text{mpg} \text{cyl}) \ P(\text{mpg}) ?$
Math	$P(\text{mpg} \text{cyl}) \ P(\text{cyl}) = P(\text{mpg}, \text{cyl})$
Coord aes	$x \leftarrow \text{mpg}$
Prob aes	$\text{height} \leftarrow P(\text{cyl}) \ P(\text{mpg} \text{cyl})$
Visual aes	$\text{fill} \leftarrow \text{cyl}$

Further questions

Based on *the Cognitive Dimensions of Notations*:

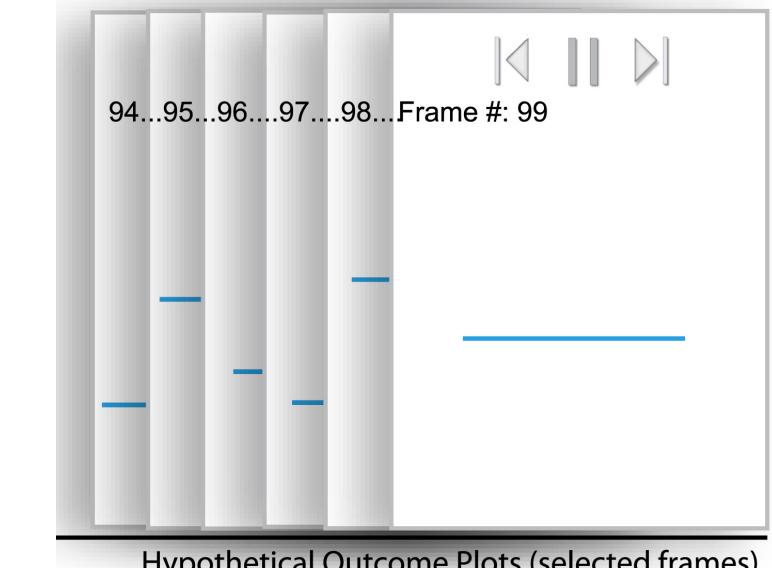
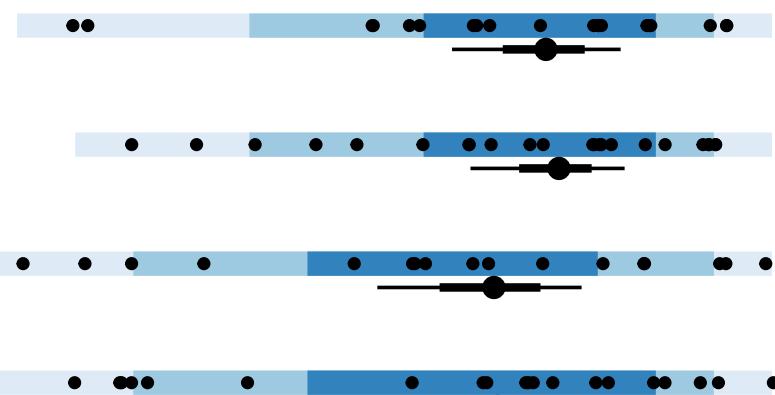
<i>error-proneness/ closeness of mapping:</i>	“Are users able to factorize and supply probability distributions?”	Rate correctness
<i>premature commitment:</i>	“Do users explore more visualization designs with PGoG than a baseline system”	Count visualizations explored
<i>hidden dependencies:</i>	“Can users learn to use the PGoG components to replicate existing probabilistic visualizations”	Record task completion time, rate completion

Future work: more uncertainty vises & systemization



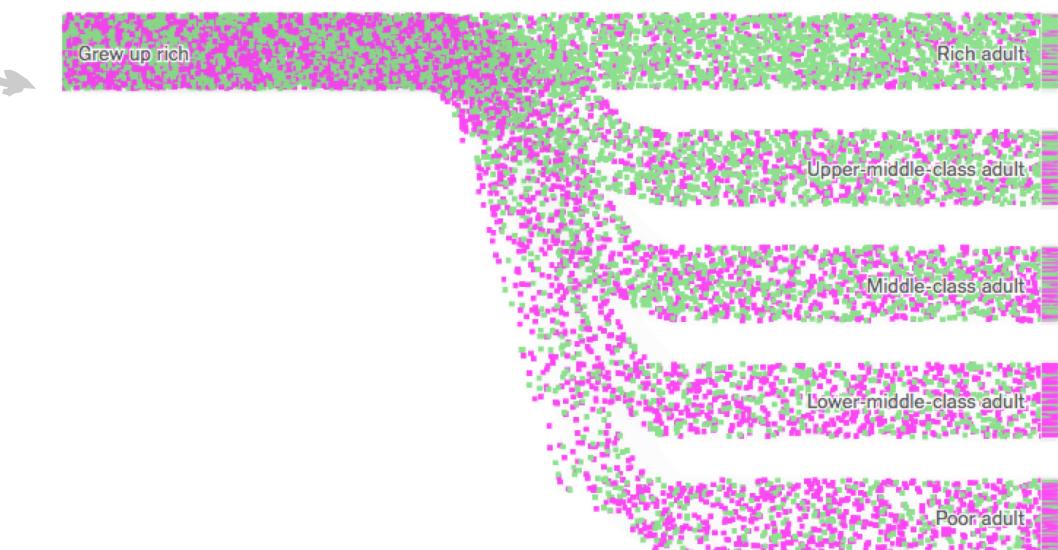
I DON'T KNOW HOW TO PROPAGATE
ERROR CORRECTLY, SO I JUST PUT
ERROR BARS ON ALL MY ERROR BARS.

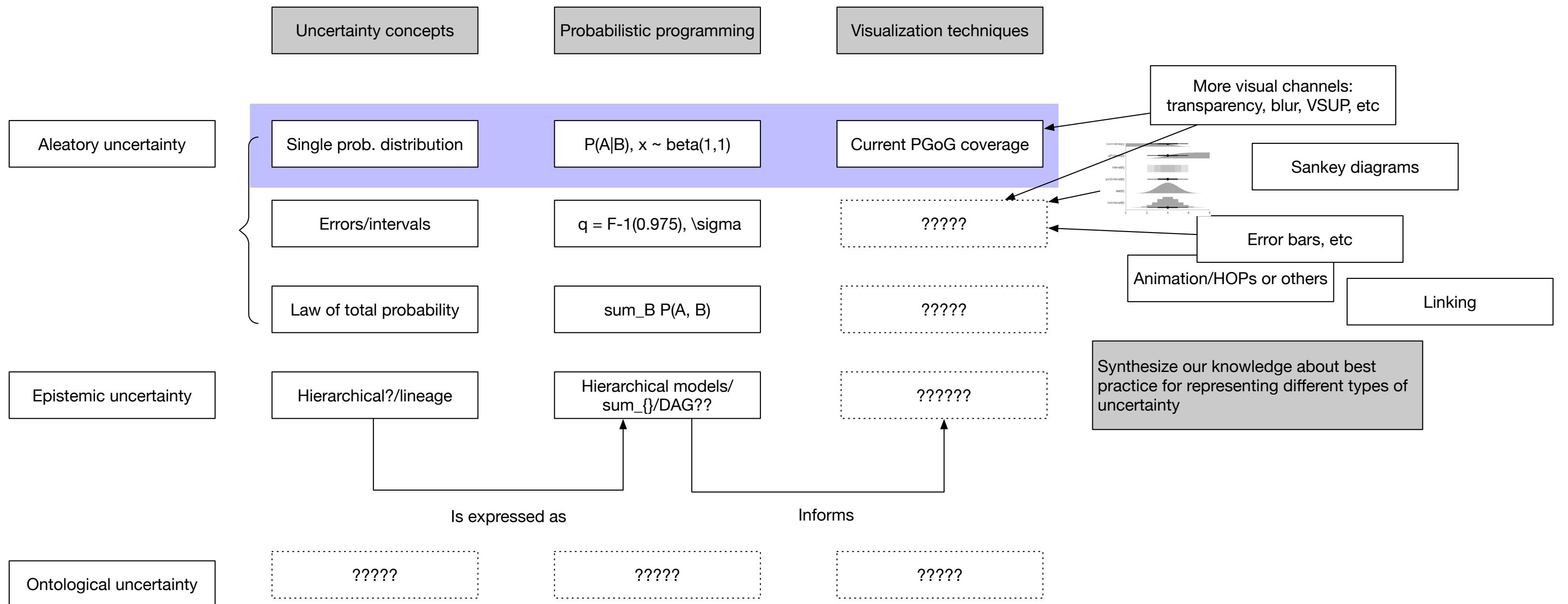
- Uncertainty sources: **aleatory** or epistemic
- Data structure: **hierarchical**, sequential, etc.
- **Summary statistics**, confidence intervals, etc.
- Visualization techniques such as **linking**



(Hullman, Resnick, and Adar 2015)

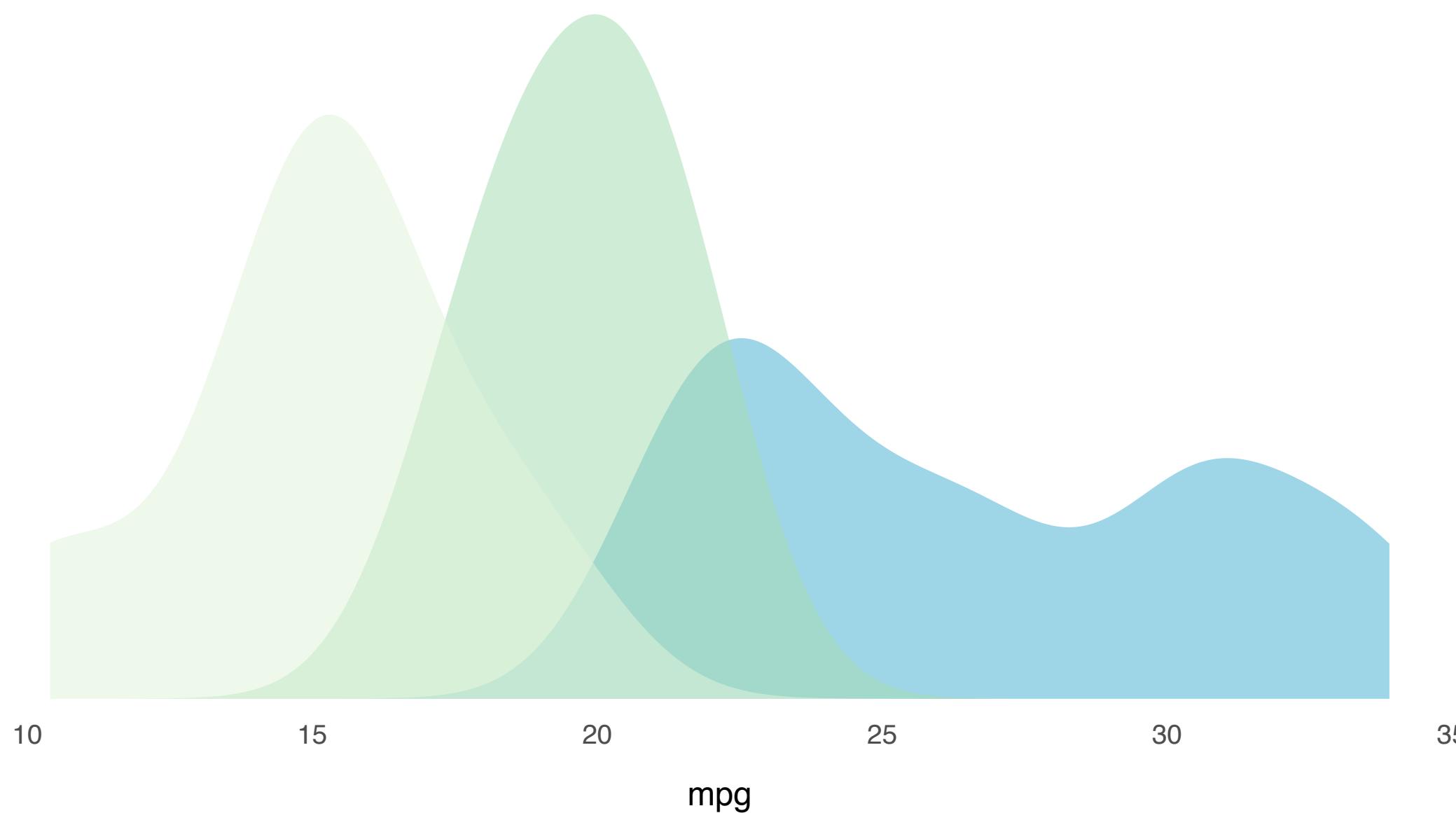
<https://www.nytimes.com/interactive/2018/03/27/upshot/make-your-own-mobility-animation.html>





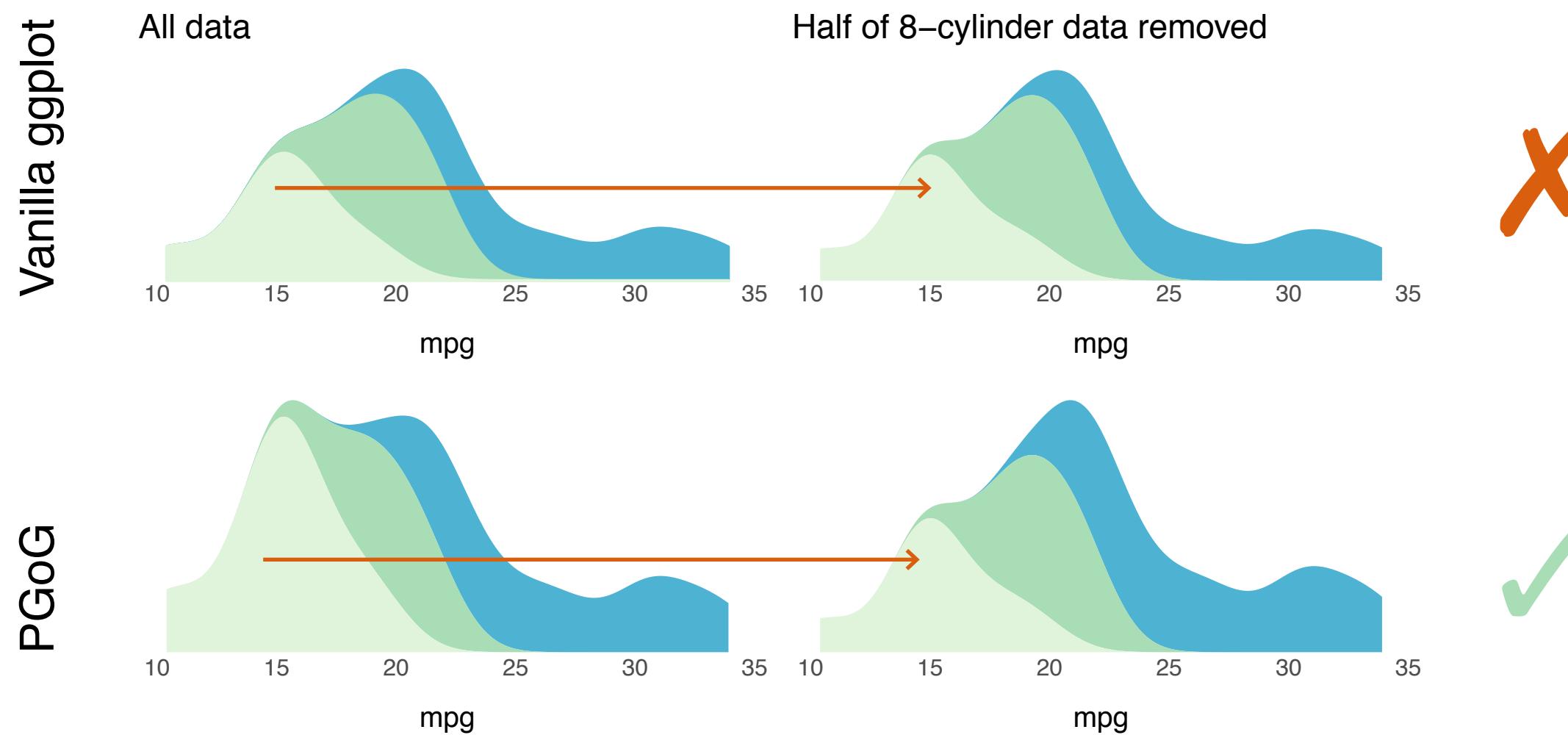
Conclusions

Probabilistic Grammar of Graphics is a visualization grammar that treats probability distributions as first-class citizens. It **shifts our thinking** about specifications for **probabilistic visualizations** and could facilitate **uncertainty** communication in the future.



The non-ambiguous way of presenting $P(\text{mpg}|\text{cyl})$

Using an algebraic process for visualization design



α : data symmetry \sim reduced 7/32 data points

ω : visualization symmetry \sim does the visualization change accordingly?

$\alpha \neq \omega$: violation of the principle of data-visualization correspondence