A Probabilistic Grammar of Graphics

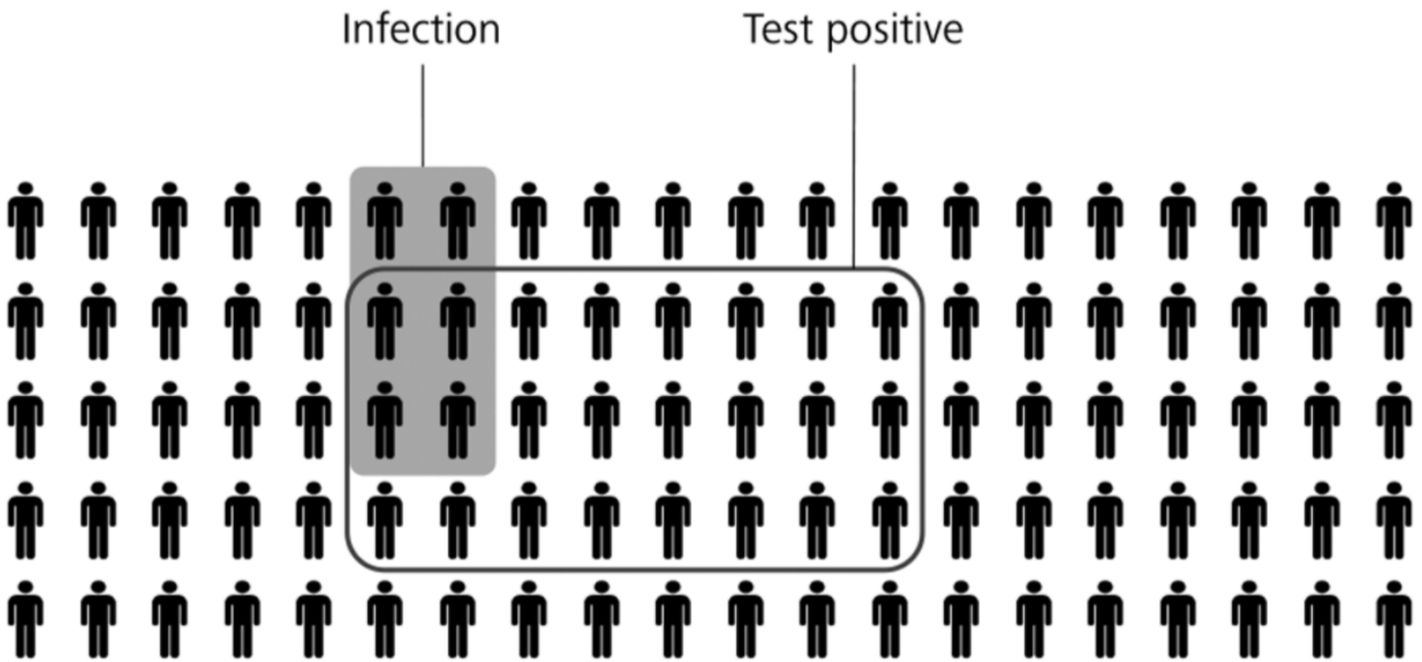
Xiaoying Pu
Prelim presentation

So I got tested for tuberculosis and it came back positive

Bayesian reasoning in
context of TB skin test

- P(test positive)?
- P(infected)?
- P(test positive AND infected)?
- P(infected GIVEN test positive)?
- arrrg, P(B|A) = P(B,A) / P(A)???!!!

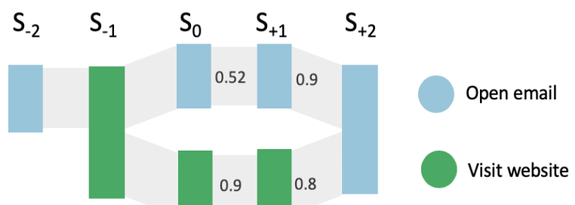
This visualization can help a stress-ridden patient



(Binder, Krauss, and Bruckmaier 2015)

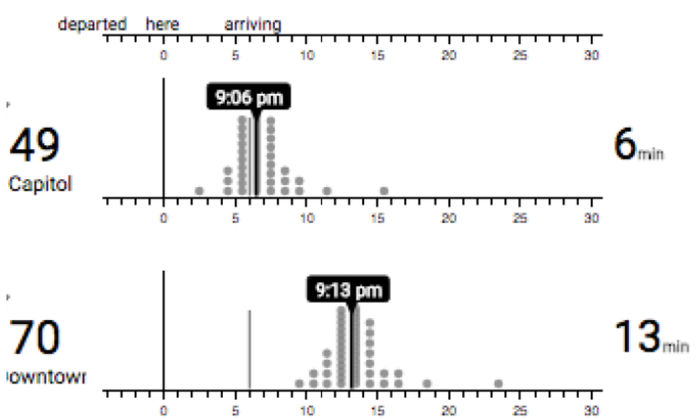
- P(test positive) = 24%
- P(infected) = 6%
- P(test positive AND infected) = 4%
- P(infected | test positive) = 17%

Machine learning



Guo et al. 2019

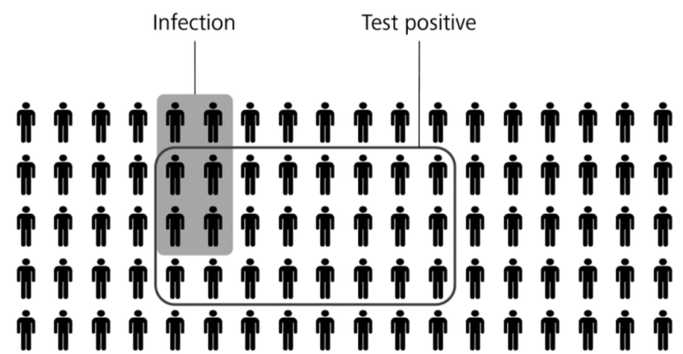
Bus arrival time



(Fernandes et al. 2018)

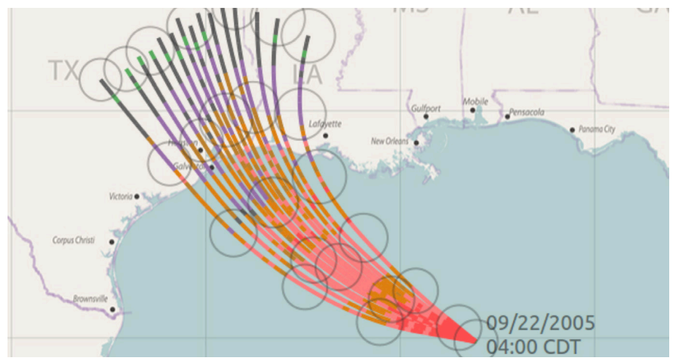
Probabilistic visualizations: same substrate, many benefits ... but difficult to spec right

Medical risk communication



(Binder, Krauss, and Bruckmaier 2015)

Hurricane forecast



(Liu et al., 2018)

Election forecast

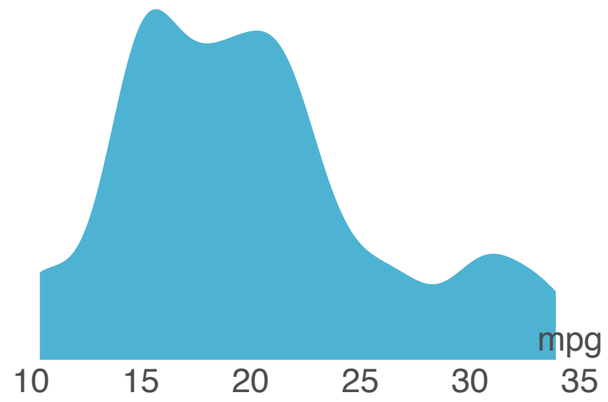
(New York Times, 2016)

Motivating example: what could possibly go wrong?

	mpg	cyl
Mazda RX4	21.0	6
Mazda RX4 Wag	21.0	6
Datsun 710	22.8	4
Hornet 4 Drive	21.4	6

A user's mental process during data exploration

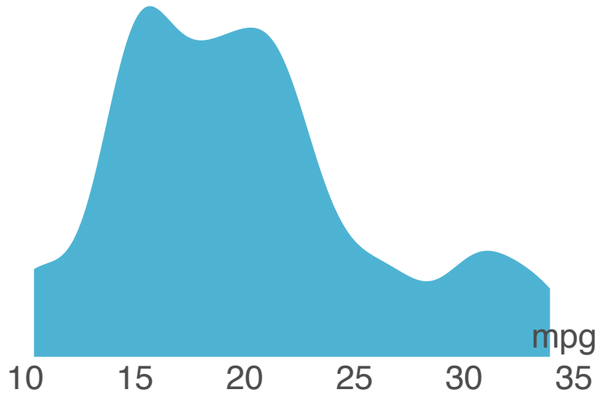
I care about emissions.
What's the mileage data like?



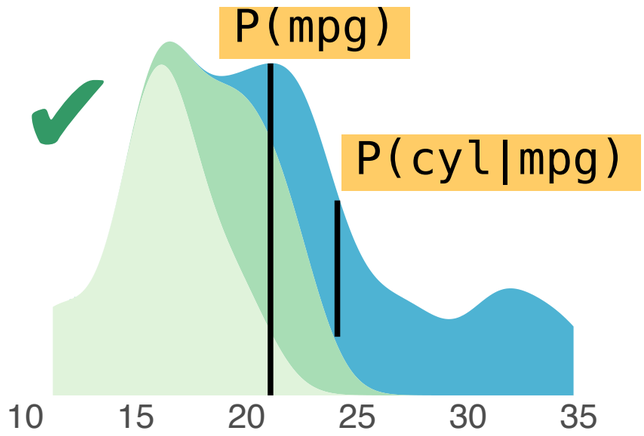
What could possibly go wrong?

A user's mental process during data exploration

What's the mileage data like?



I want to see both mileage and cylinder counts... maybe there's a pattern

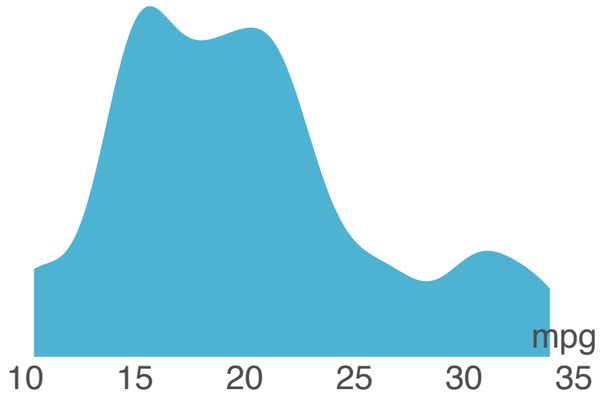


what user wants (correct)

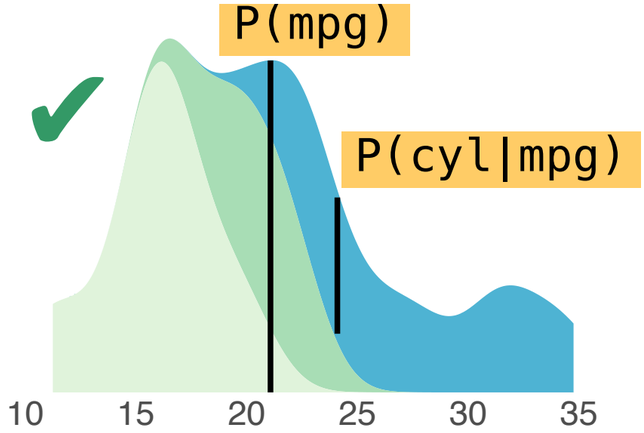
What could possibly go wrong?

A user's mental process during data exploration

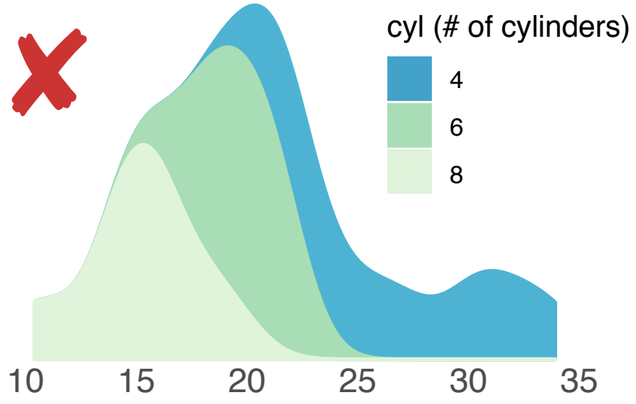
What's the mileage data like?



I want to see both mileage and cylinder counts... maybe there's a pattern

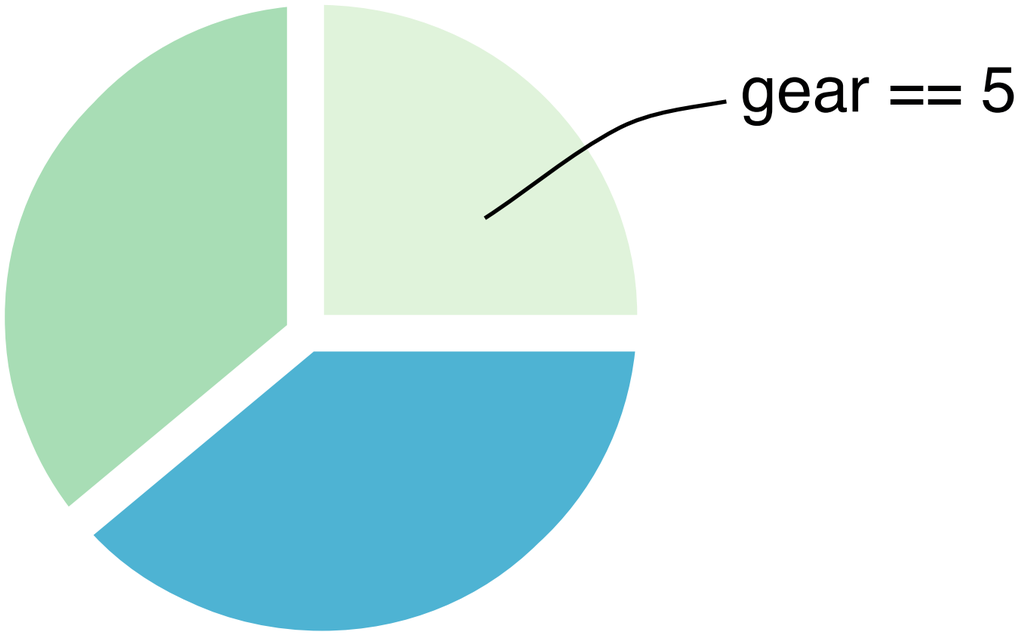


what user wants (correct)

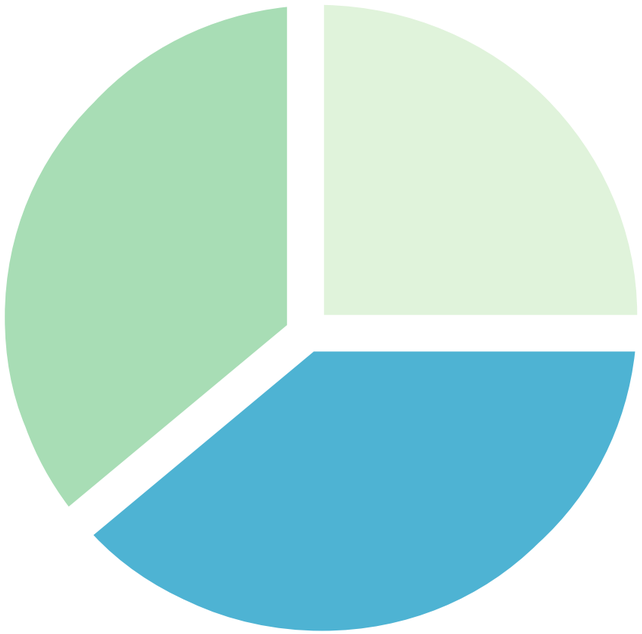


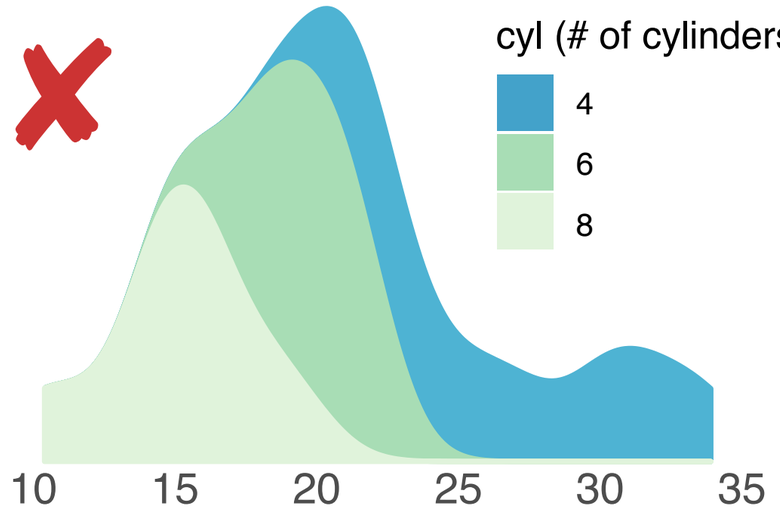
what user gets (incorrect)

Problem 1: vis shows incorrect probability distribution



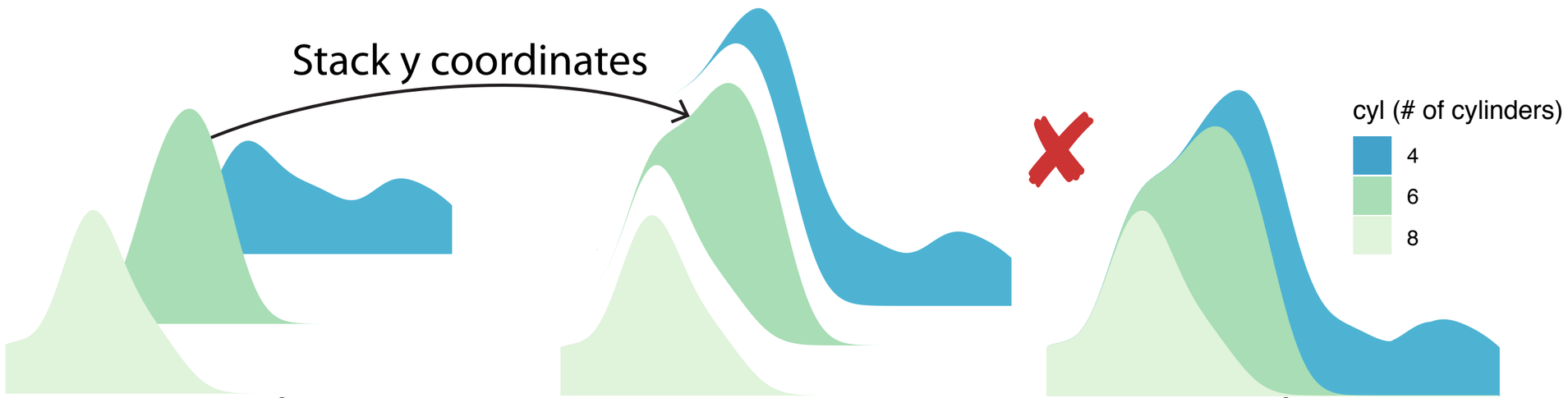
Problem 1: vis shows incorrect probability distribution





what user gets
(incorrect)

Problem 1: vis shows incorrect probability distribution



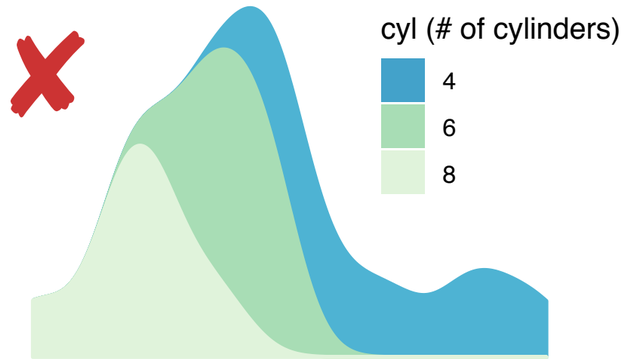
P(mpg|cyl)

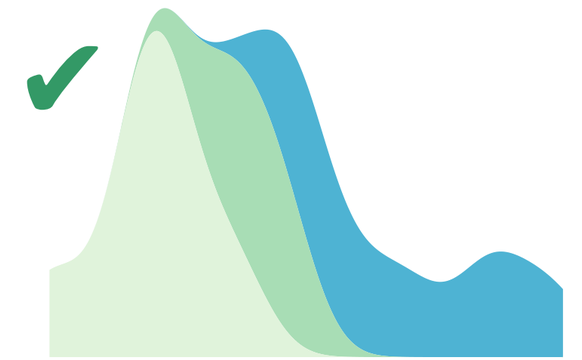
```
ggplot(mtcars) +
geom_density(aes(
        x = mpg,
     fill = cyl),
```

$\Sigma_{cyl} P(mpg|cyl) = ??$

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

Problem 1: vis shows incorrect probability distribution

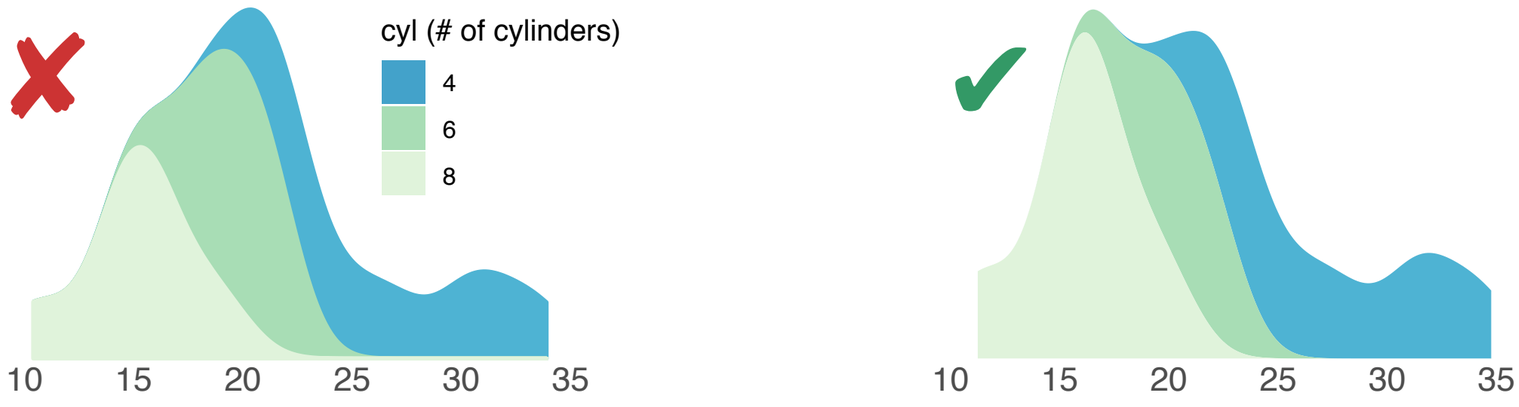




Two incorrect inferences

1. Wrong distribution of # of cylinders **P(cyl)**
2. Wrong overall distribution of mileage **P(mpg)**

Wait we can fix this density plot

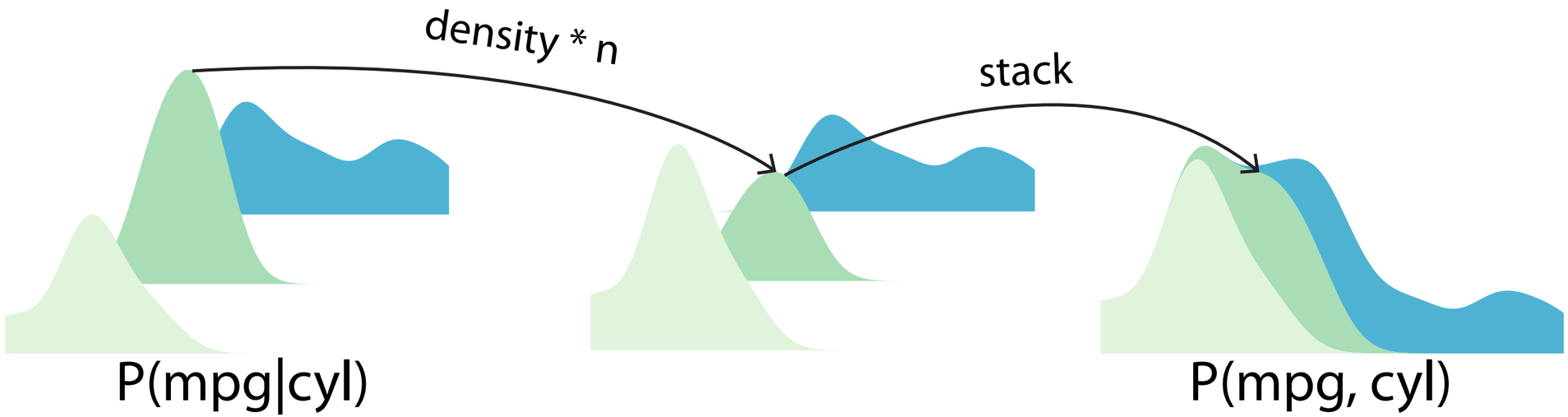



```
ggplot(mtcars) +
geom_density(
   aes(x = mpg,
       y = stat(density),
    fill = cyl),
position = "stack")
```



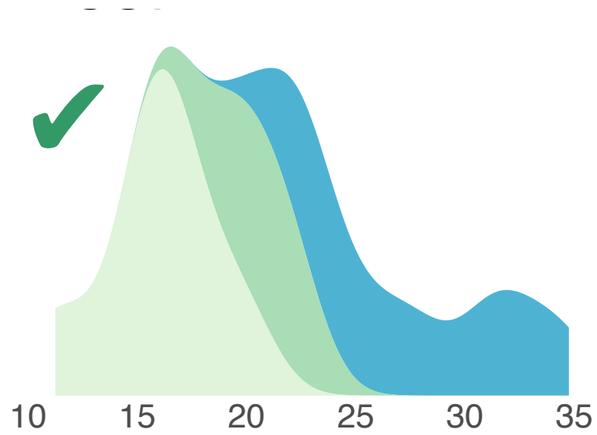
```
ggplot(mtcars)+
geom_density(
   aes(x = mpg,
       y = stat(density*n),
    fill = cyl)),
position = "stack")
```

Problem 2: specifying probability distributions is convoluted

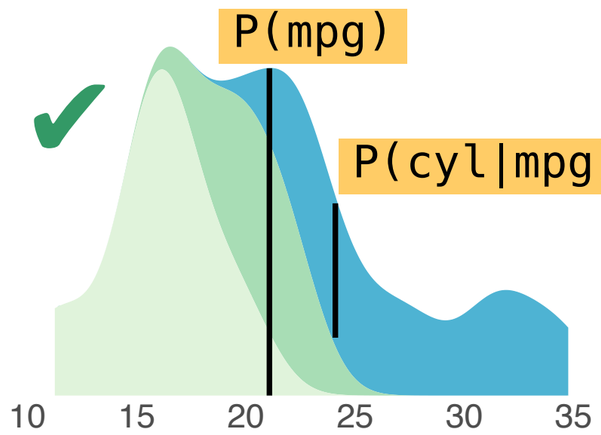


But how I am supposed to know `stat(density*n)` and `position`?

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

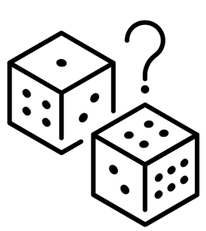
PGoG implementation

Evaluation in terms of
- Expressiveness
- Generativeness
- Cognitive ergonomics

Future: quantitative uncertainty communication!

Related work: motivations and theories behind PGoG

Probabilistic visualizations are important but difficult/error-prone to specify and explore...
What should a grammar for probabilistic visualizations look like?

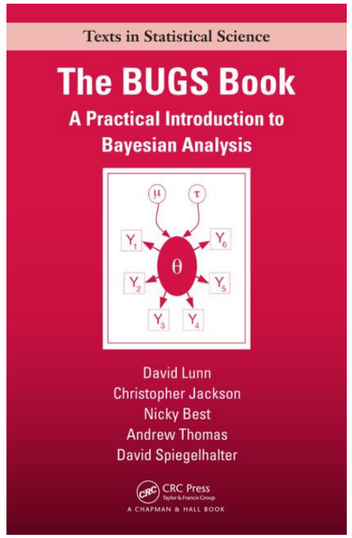


- Probability first: benefits of having a “probabilistic” grammar

- Usability and correctness: current visualization specs are inadequate for probabilistic visualizations

- Theoretical grounding: how uncertainty communication can inform visualization design

Related: probabilistic programming



BUGS

Stan

$$\theta \sim beta(1, 1)$$
$$y \sim bernoulli(\theta)$$

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

Widely cited and applied to many domains

Sticks with what users know (probabilities)
Avoids implementation details (`stat(density*n)`)

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

(Wickham 2010)

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

Data	+	Aesthetics	+
`example_df`		`x <- A` `y <- B`	

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

Tidy format, 3 columns

x	y	C
1	2	a
2	1	a
3	4	b
4	2	b

A is “mapped onto” x

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

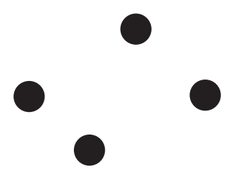
Data	+	Aesthetics	+	Geometry	+
`example_df`		`x <- A` `y <- B`		`geom_point`	

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

Tidy format, 3 columns

x	y	C
1	2	a
2	1	a
3	4	b
4	2	b

A is “mapped onto” x



Uses x, y aesthetics

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

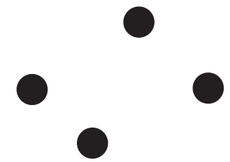
Data	+	Aesthetics	+	Geometry	+ ... =	A plot
`example_df`		`x <- A` `y <- B`		`geom_point`		

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

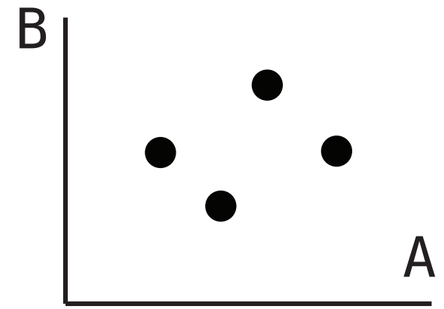
Tidy format, 3 columns

x	y	C
1	2	a
2	1	a
3	4	b
4	2	b

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

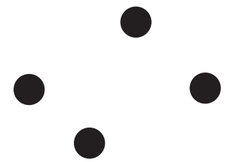
`example_df`		`x <- A`
		`y <- B`
		`color <- C`
				`geom_point`		

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

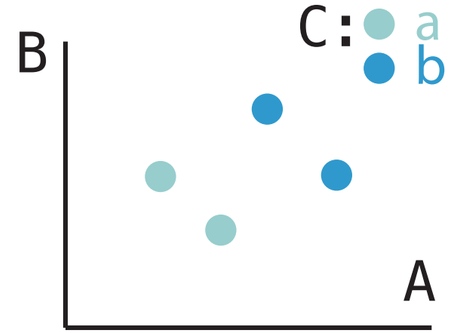
Tidy format, 3 columns

x	y	color
1	2	a
2	1	a
3	4	b
4	2	b

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)

```
{
 "data": "data/titanic.csv",
 "width": 320, "height": 240,  "padding": {..},
 "layouts": [{
  "name": "layout1",
  "type": "gridxy",
  "subgroup": {  "type": "groupby",  "key": "Class" },
  "aspect_ratio": "fillX",
  "size": { "type": "uniform",  "isShared": false },
  "direction": "LRBT",  "align": "LB",
  "margin": { ... },  "padding": {... }
 }, {
  "type": "gridxy",
  "subgroup": { "type": "groupby", "key": "Gender", "isShared": true },
  "aspect_ratio": "fillY",
  "size": { "type": "uniform", "isShared": true },
 }, {
  "subgroup": { "type": "flatten" },
  "aspect_ratio": "square",
  "size": { "type": "uniform", "isShared": true },
  "sort": { "key": "Survived"}
 }],
 "mark": {
  "shape": "circle",
  "color": { "key": "Survived", "type": "categorical" },
  "size": { "type": "max",  "isShared": false },
  "isColorScaleShared": true
 }
}
```

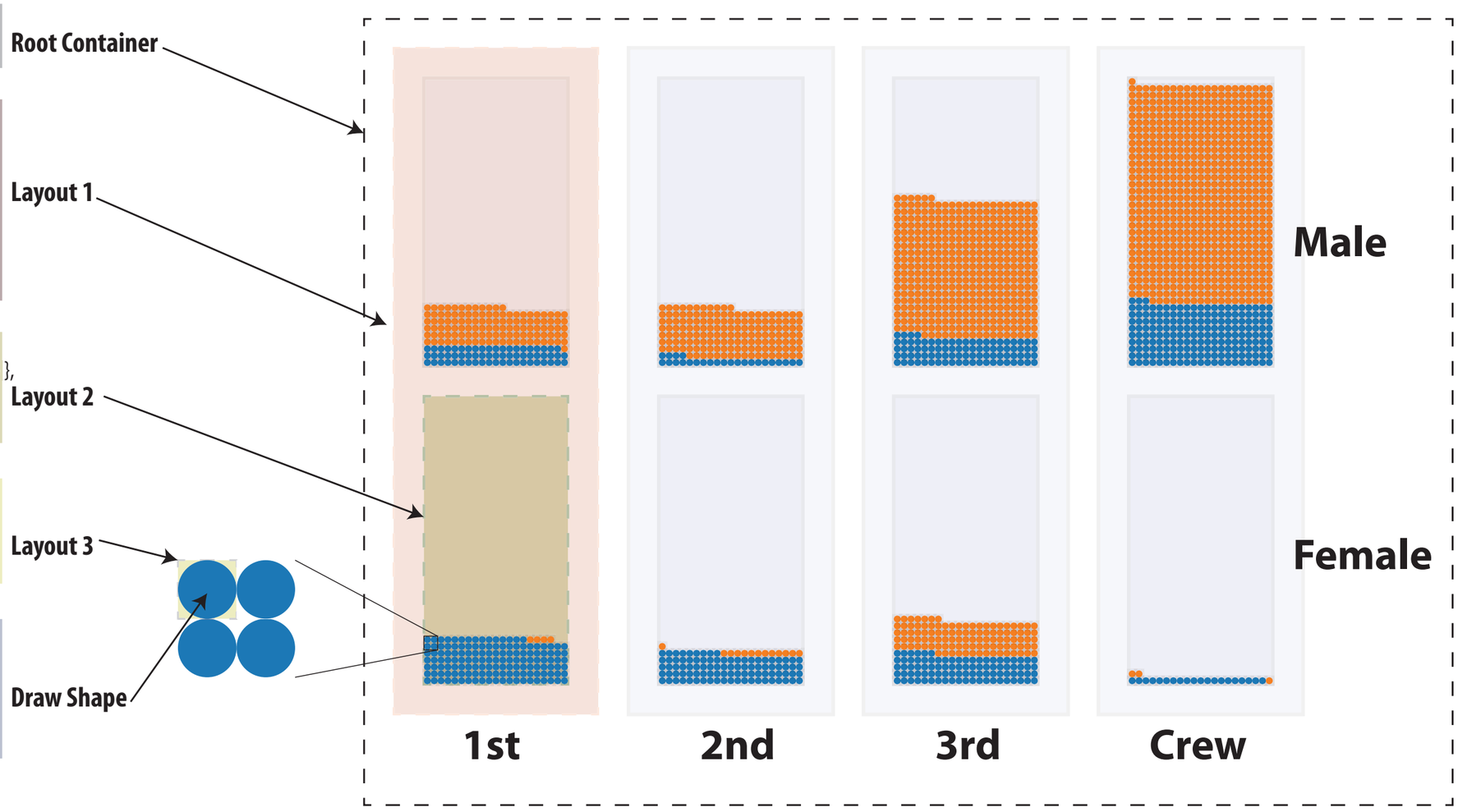
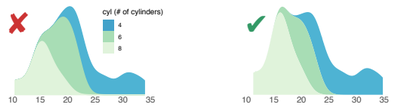


Fig. 6. Example grammar to generate a unit column chart for survivors of the Titanic by passenger class. *(Park et al. 2017)*

Related: how to specify a visualization in general

Grammar of Graphics



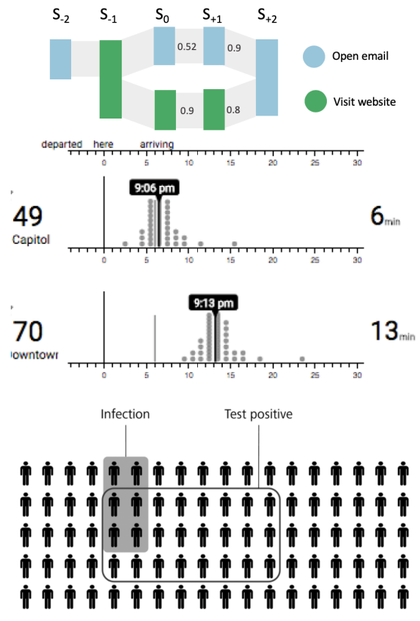
Layout-based

Constraints-based

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

(Moritz et al. 2019)

Correct?
Easy to use/ explore designs?

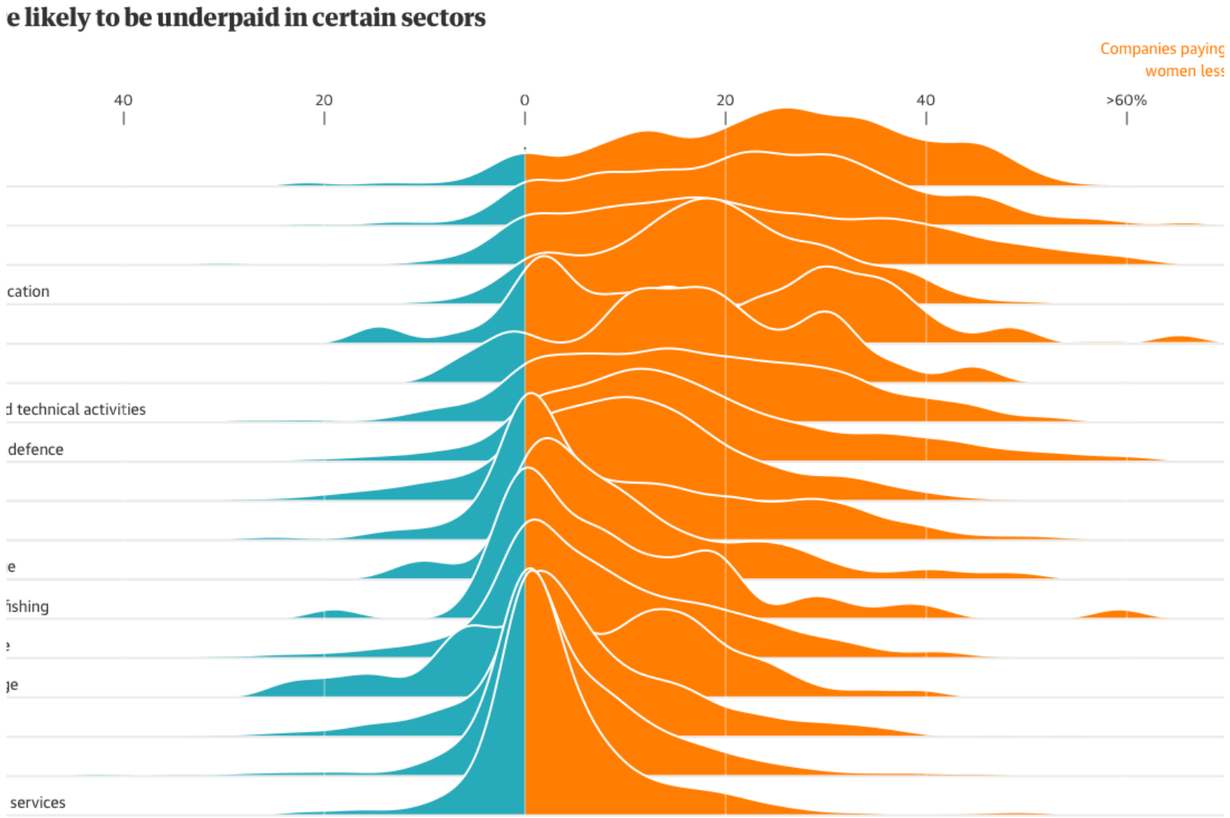


Need a closer integration between statistics and visualization

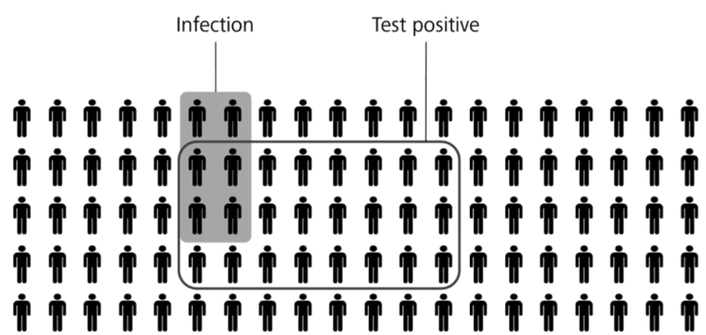
(Heer and Shneiderman 2012)

Related: communicating/visualizing uncertainty

Probabilistic visualizations
are often used to communicate uncertainty data



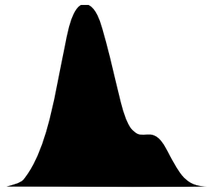
https://www.theguardian.com/news/ng-interactive/2018/apr/05/women-are-paid-less-than-men-heres-how-to-fix-it

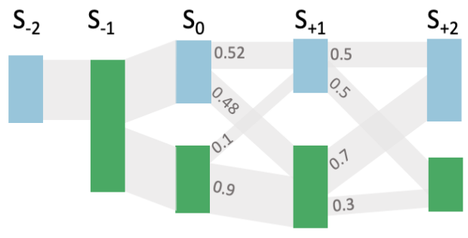


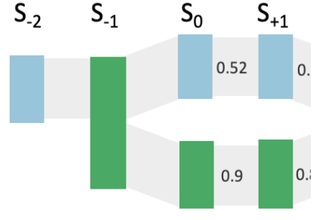
Related: communicating/visualizing uncertainty

Probabilistic visualizations
are often used to communicate uncertainty data

Probability format x%



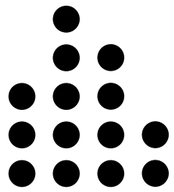


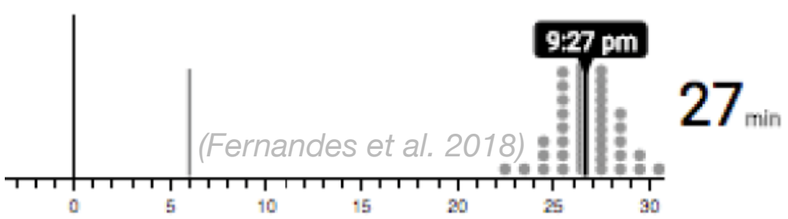


(Guo et al. 2019)

(Gigerenzer and Hoffrage 1995)

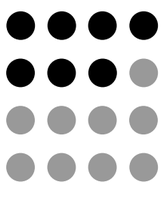
Frequency format x-in-100

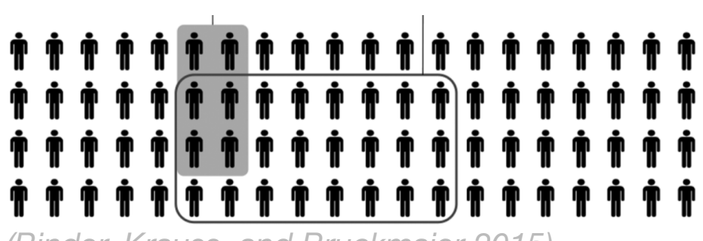




(Fernandes et al. 2018)

27 min





(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

Design Requirements for a probabilistic Grammar of Graphics

1. Guaranteeing correctness of distributions expressed in visualization

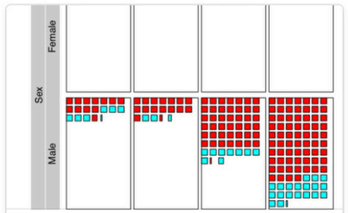
2. Enabling specification close to probability expressions, such as P(AIB), which target users know

A need to support probability and frequency formats

Design Requirements for a probabilistic Grammar of Graphics

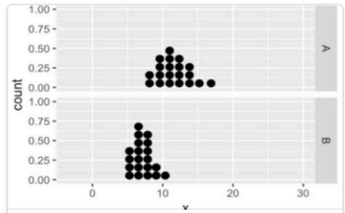
3 Facilitating exploration *with coherent and reusable grammar components*

4 (and automation in the future)



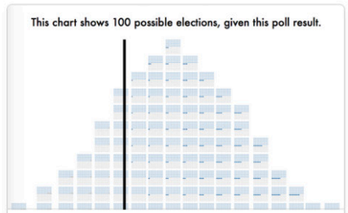
Faceted icon array

icon uncertainty



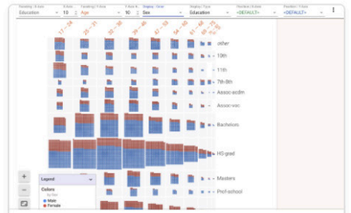
Quantile dotplots

uncertainty icon



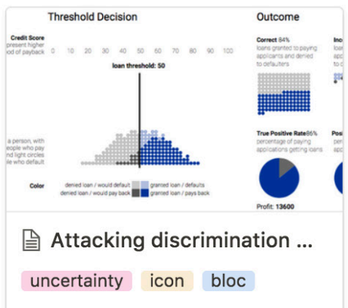
Possible worlds visualizati...

composite icon dotplot



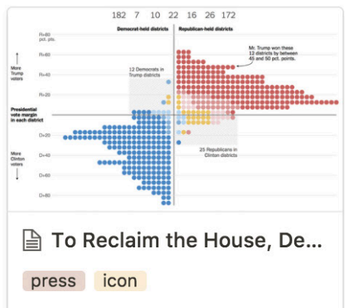
Facets DIVE

icon



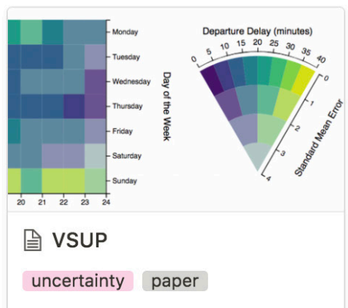
Attacking discrimination ...

uncertainty icon bloc



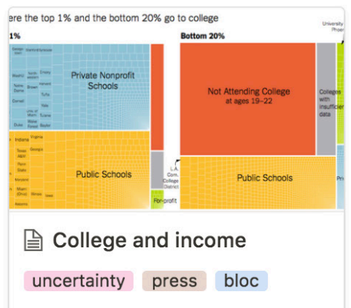
To Reclaim the House, De...

press icon



VSUP

uncertainty paper

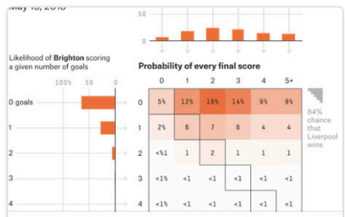


College and income

uncertainty press bloc

Confusing ggmosaic

uncertainty bloc

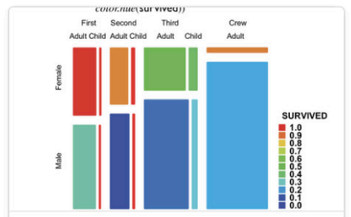


How Our Club Soccer Pre...

press sports bloc

Product plots

paper uncertainty bloc



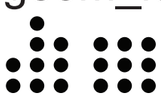
Mosaic plot of Titanic sur...

The design process

Grammar	ggplot2
Defaults	
Data	A
Aesthetics	x ← A
Layer	
Data	
Aesthetics	
Geom	geom_bar
Stat	
Position	geom_density
Scale	
	geom_points
Coord	
Facet	geom_rect
	
	geom_...

(Wickham 2010)

What is the Probabilistic Grammar of Graphics?

Grammar	ggplot2	PGoG
Defaults		
Data ------>	A	P(A|B,...)
Aesthetics -->	x ← A	height ← P(A|B,...)
Layer		
Data		
Aesthetics		
Geom ------>	geom_bar	geom_bloc
Stat		
Position	geom_density	geom_icon
Scale		
	geom_points	
Coord		
Facet	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
```

Simple variable

`mpg`

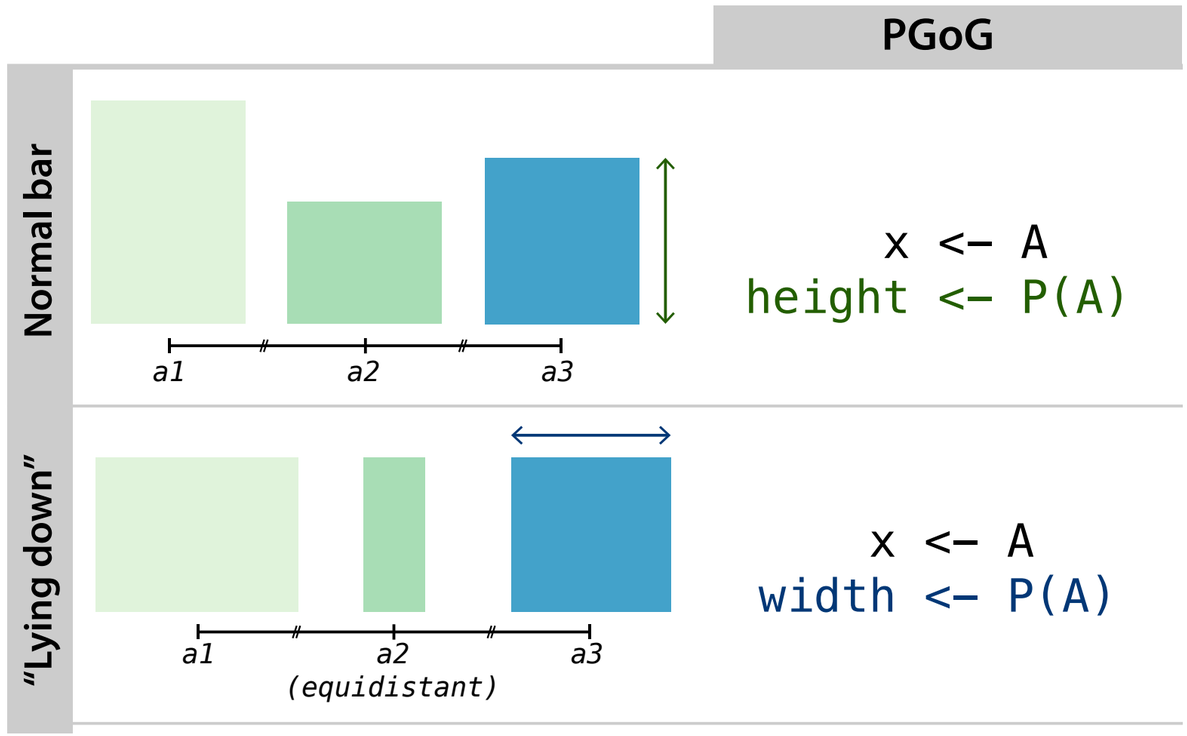
A column in tidy dataset

Probabilistic variable

`P(mpg|cyl)`

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

PGoG Grammar/*aesthetics 1/3*

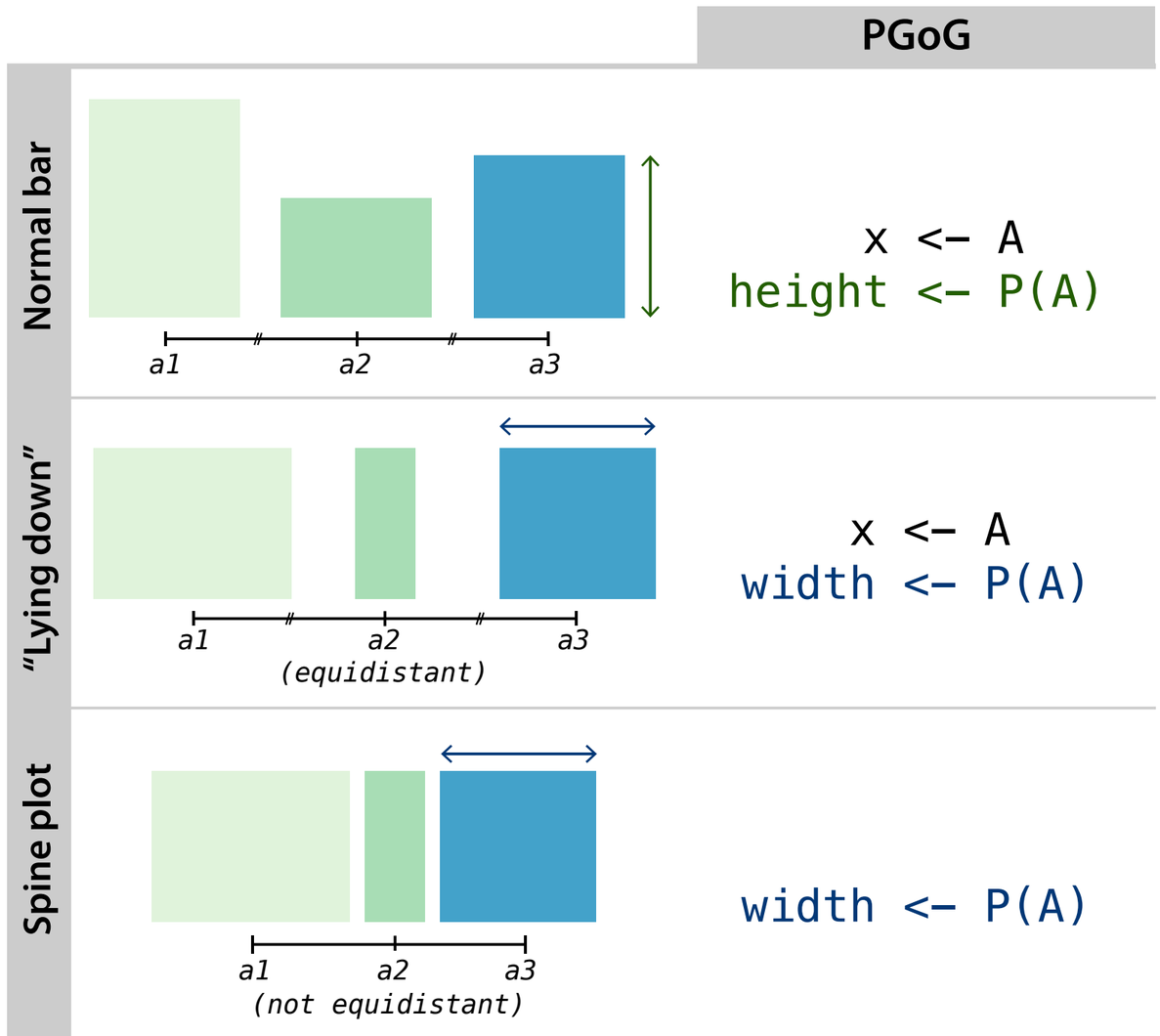


Probabilistic aesthetics

`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

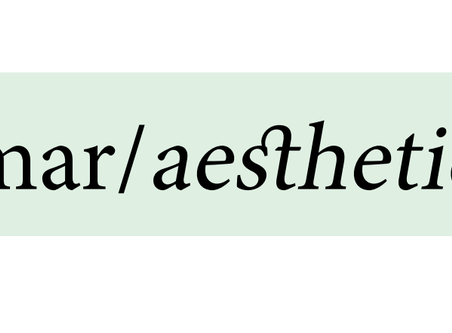
PGoG Grammar/*aesthetics 3/3*

Probabilistic aesthetics

Coordinate aesthetics

Visual aesthetics

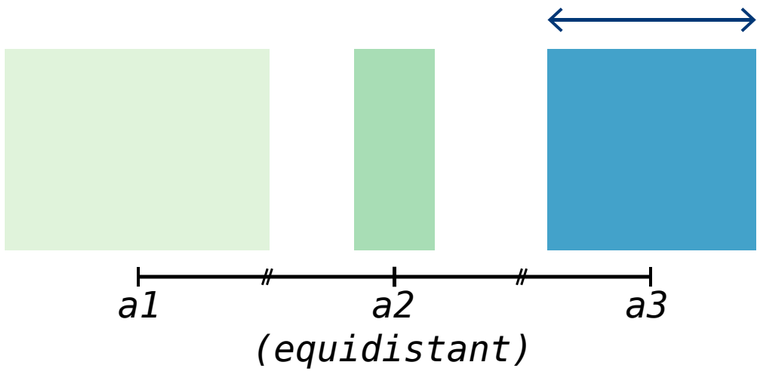
Uses x, y aesthetics



A scatter plot

```
fill,color,alpha,...
```

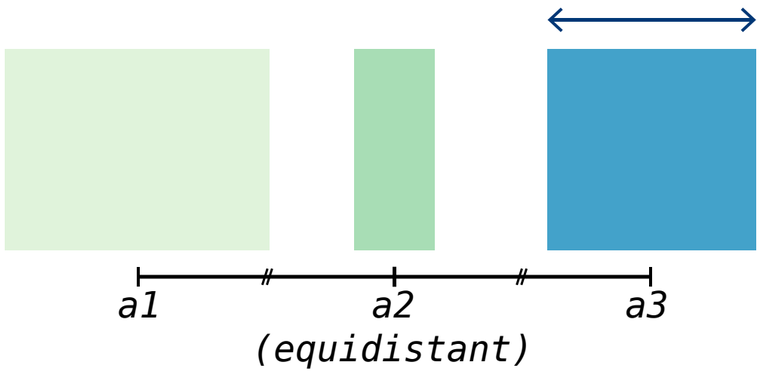
PGoG Grammar/*Example for conditional*




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

```
x <- race

width <- P(race|generation)
```

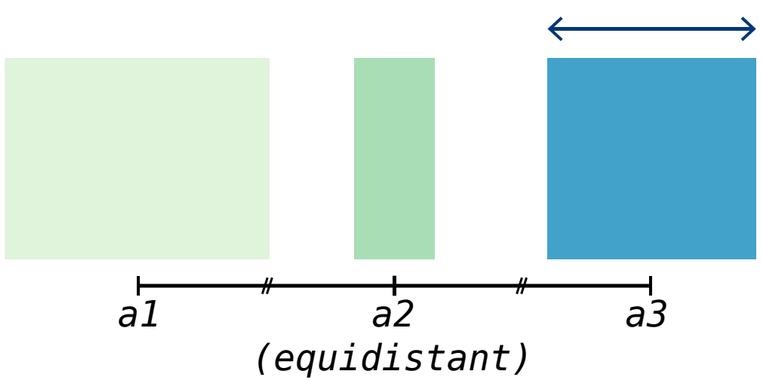
PGoG Grammar/*Example for conditional*




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

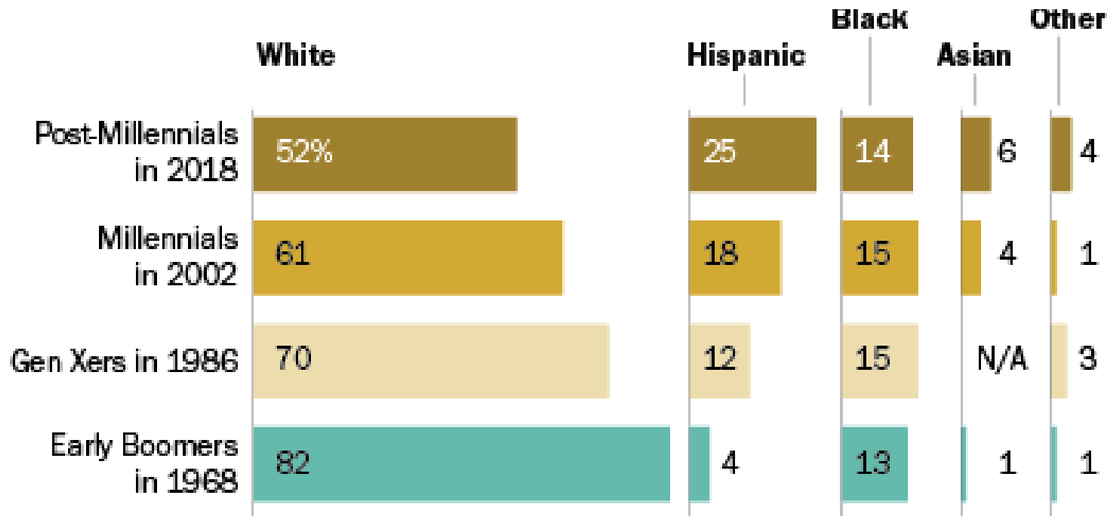

```
x <- race

width <- P(race|generation)
```

PGoG Grammar/*Example for conditional*




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

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



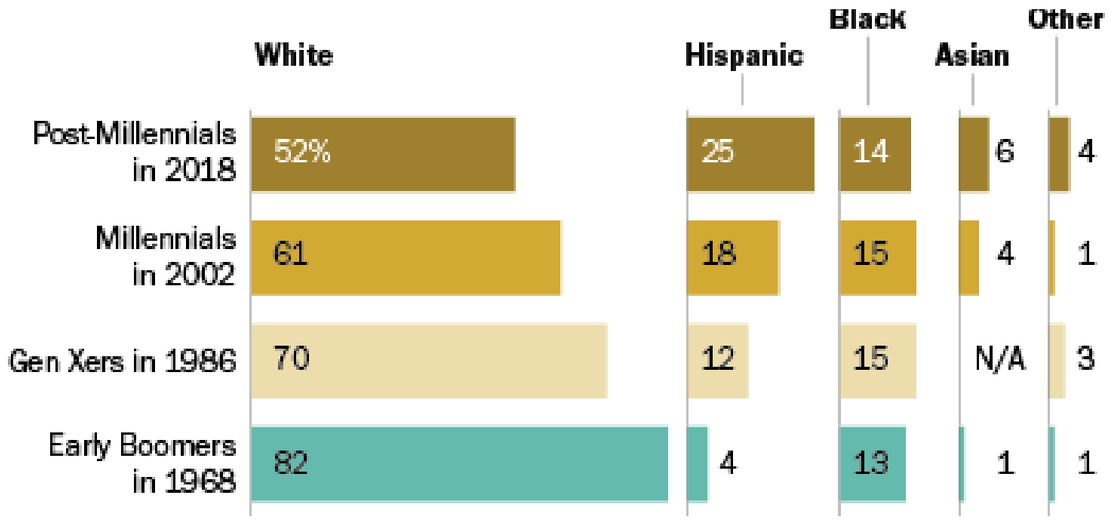

```
x <- race
y <- generation
width <- P(race|generation)
```

PGoG Grammar/*Example for conditional*

```
population %>%
  group_by(generation,race) %>%
  summarize(count.gen = n()) %>%
  mutate( prop.cyl = count.gen/
    sum(count.gen)) %>%
  ggplot(aes(x = factor(generation), y =
prop.gen)) +
  geom_col() +
  coord_flip() +
  facet_grid(.~race)
```

Base ggplot

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




```
x <- race
y <- generation
width <- P(race|generation)
```

PGoG

PGoG Grammar/*Example for joint*

Math `P(cyl|mpg) P(mpg) = P(mpg,cyl)`

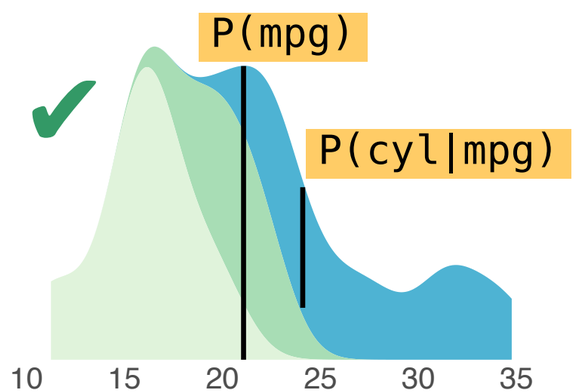
PGoG Grammar/*Example for joint*

Math	`P(cyl|mpg) P(mpg) = P(mpg,cyl)`
Coord aes	`x <- mpg`

PGoG Grammar/*Example for joint*

Math	`P(cyl|mpg) P(mpg) = P(mpg,cyl)`
Coord aes	`x <- mpg`
Prob aes	`height <- P(mpg) P(cyl|mpg)`

PGoG Grammar/*Example for joint*




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

Math	`P(cyl|mpg) P(mpg) = P(mpg,cyl)`
Coord aes	`x <- mpg`
Prob aes	`height <- P(mpg) P(cyl|mpg)`
Visual aes	`fill <- cyl`

PGoG Grammar/*checking correctness 1/2*

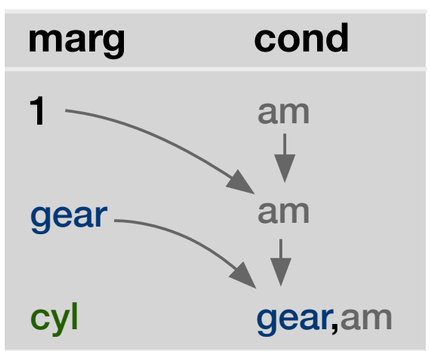
One of the **rules**: the probabilistic variables need to be valid factors of a probability function

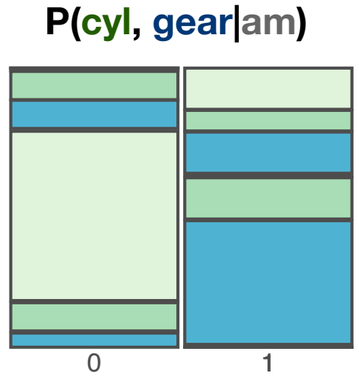
```
     x <- gear
height <- P(gear|am)
          P(cyl|gear,am)
```

PGoG Grammar/*checking correctness*

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

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





A “chain” data structure used for checking probabilistic variables

Grammar	ggplot2	PGoG
Defaults		
Data	A	P(A|B,...)
Aesthetics	x ← A	height ← P(A|B,...)
Layer		
Data		
Aesthetics		
Geom	geom_bar	geom_bloc
Stat		
Position		
	geom_density	geom_icon
Scale		
	geom_points	
Coord		
	geom_rect	
Facet		
	geom_...	

PGoG Grammar/ *geometries 1/2*

ggplot2	PGoG
geom_bar 	**geom_bloc**
geom_mosaic* 	
geom_density 	
geom_violin 	
geom_density_ridges* 	
geom_waffle* 	**geom_icon**
geom_dotplot 	

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

Also, probability and frequency formats

* `ggplot2` extensions

PGoG Grammar/*geometries*

`geom_bloc`: recursive sub-partition to support many probabilistic variables

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

`geom_icon` needs a new way to pack icons

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

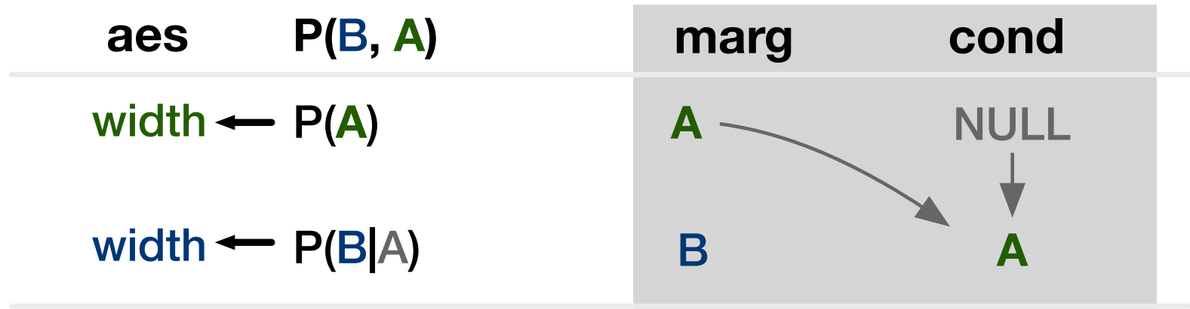
PGoG Grammar/*geometries*

`geom_bloc`: recursive sub-partition to support many probabilistic variables

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

`geom_icon` needs a new way to pack icons

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



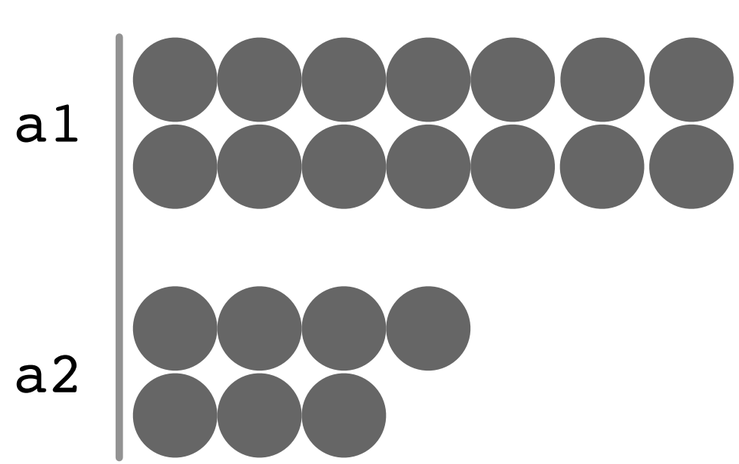
Probability structure (the “chain”) determines the visualization structure

PGoG Grammar/*geometries*

`geom_icon` needs a new way to pack icons

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

aes	P(B, A)	marg	cond
width ←	P(A)	A	NULL
width ←	P(B|A)	B	A



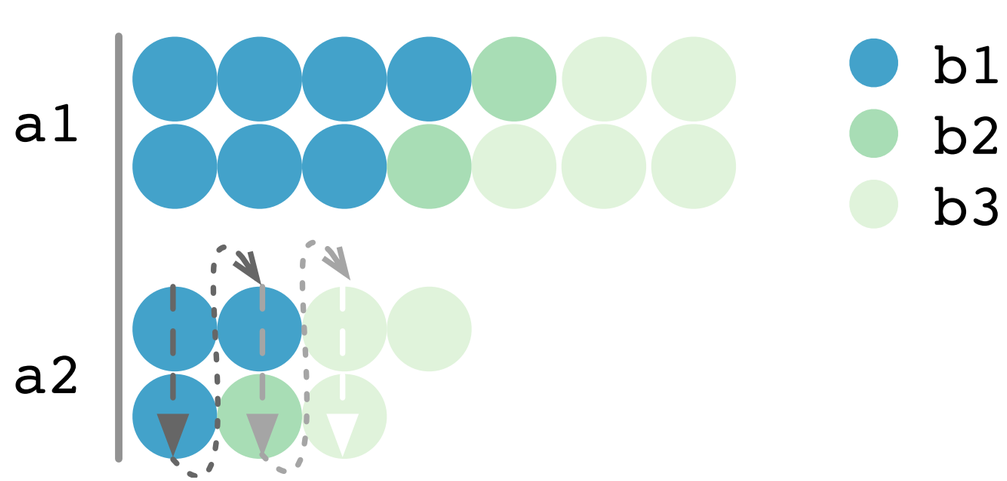
Probability structure (the “chain”) determines the visualization structure

PGoG Grammar/*geometries*

`geom_icon` needs a new way to pack icons

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

aes	P(B, A)	marg	cond
width ←	P(A)	A	NULL
width ←	P(B|A)	B	A



Probability structure (the “chain”) determines the visualization structure

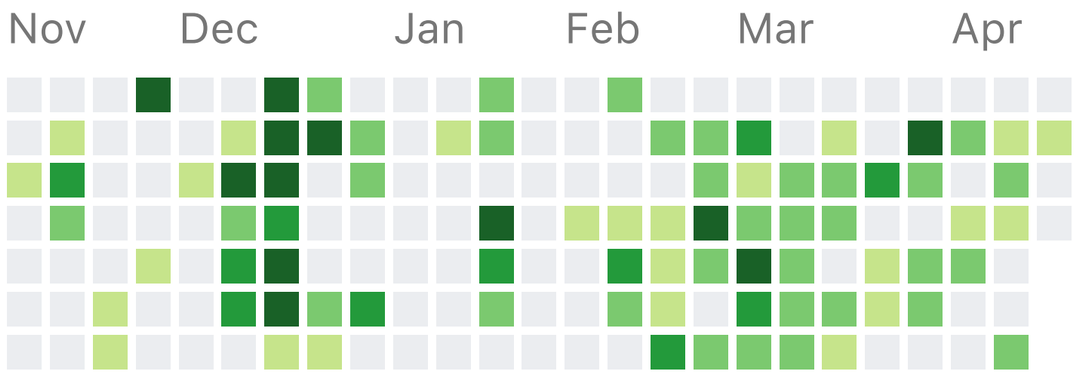
Implementing the Grammar

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

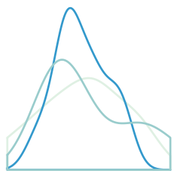
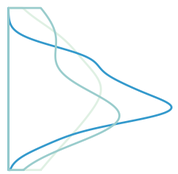
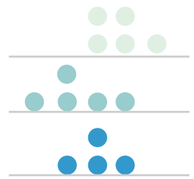
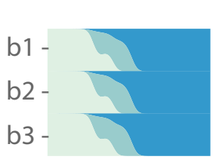
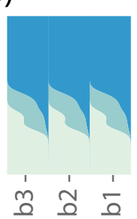
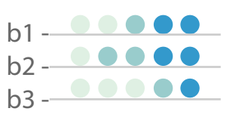
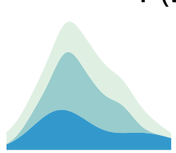
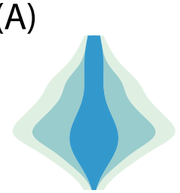
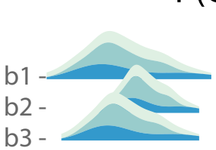
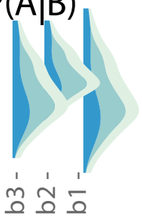
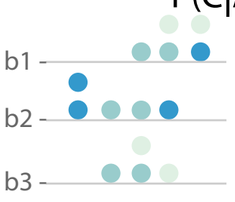
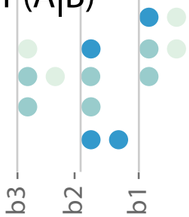
Usefulness: is it a good visualization grammar

- Expressive?
- Generative?

Usability: is it easy to express probabilities as visualizations

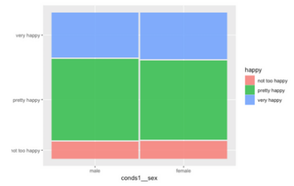
- Cognitively ergonomic?

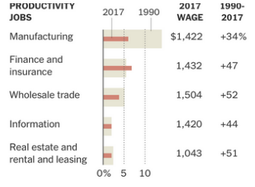
Expressiveness of the grammar

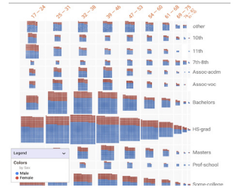
	geom_bloc				geom_icon			
	x ← A h ← ...	y ← A w ← ...	x ← A y ← B h ← ...	y ← A x ← B w ← ...	x ← A h ← ...	y ← A w ← ...	x ← A y ← B h ← ...	y ← A x ← B w ← ...
P(A)	 density plot		NA	NA	 dotplot		NA	NA
P(A|B)			 ridge plot	 violin plot				
Conditional on a continuous variable A	 P(B|A)		 P(C|A, B)		 P(B|A)		 P(C|A, B)	
Joint	 P(B|A) P(A)	 onion plot*	 P(C|A, B) P(A|B)		 P(B|A)		 P(C|A, B) P(A|B)	

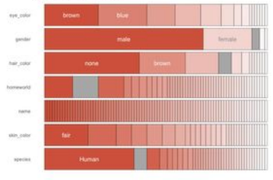
Expressiveness of the grammar, in the wild

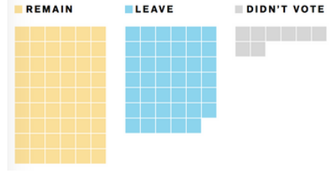
Can reasonably reproduce

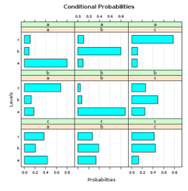


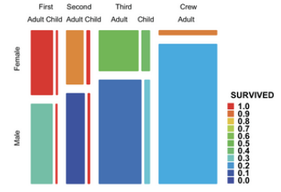


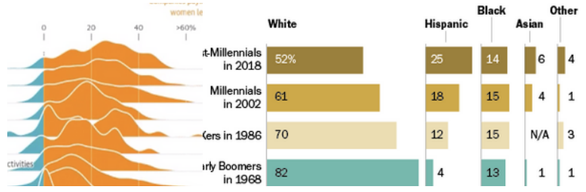




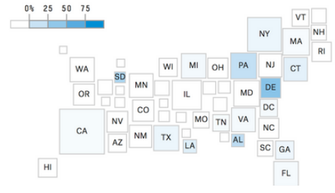


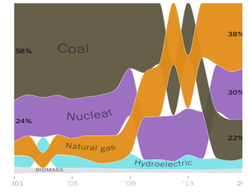


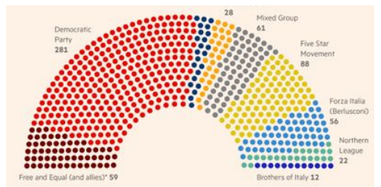


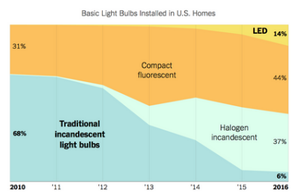


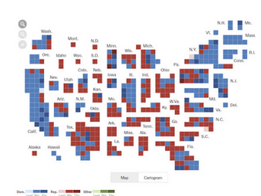
Time and space



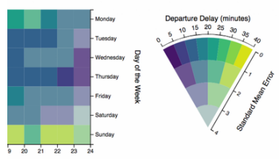


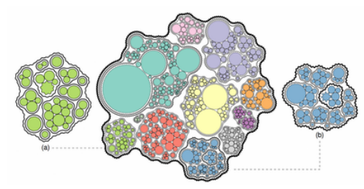


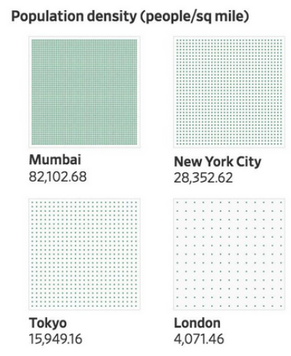




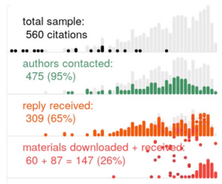
Additional channels

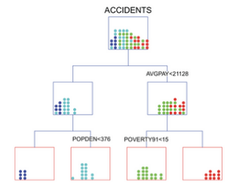


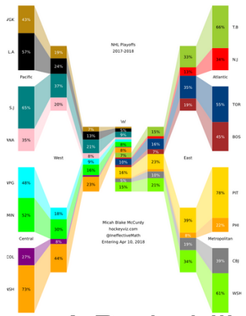


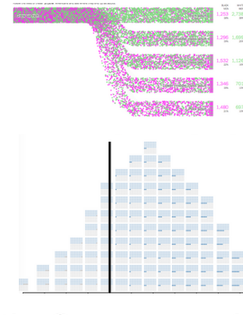


Hierarchy/structure









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)*

- ... concerns the usability of notational systems
- Evaluated with *Cognitive Dimensions of Notations*

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

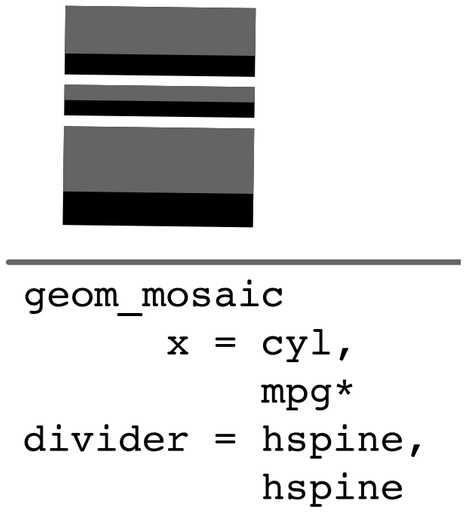
Pro: Short edit distances, close to probability expressions

- *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

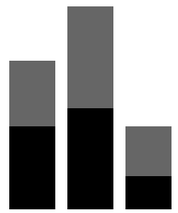
- *Low viscosity*
- *No premature commitment*

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

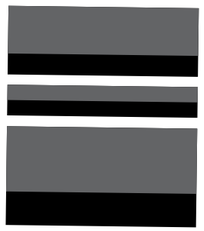
Pro: Short edit distances, close to probability expressions

- *Low viscosity*
- *No premature commitment*

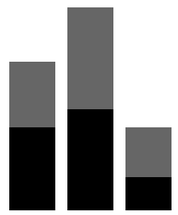
Existing ggplot2 packages

Changes

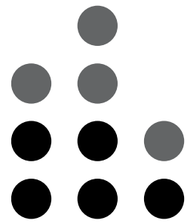
Syntax



```
geom:mosaic→bar
+fill +y
-divider
```



```
geom:bar→dotplot
+stackgroup +method
-y
```



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

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

```
geom_dotplot
        x = mpg*
     fill = cyl
stackgrp = TRUE
  method = histodot
```

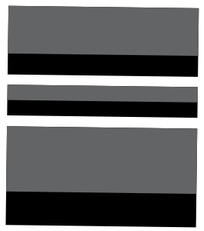
Cognitive ergonomics

Pro: Short edit distances, close to probability expressions

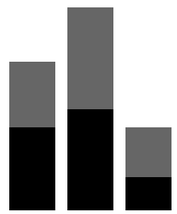
- *Low viscosity*
- *No premature committment*

Existing ggplot2 packages

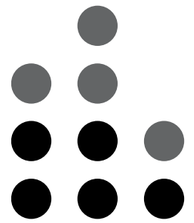
Changes



```
geom:mosaic→bar
+fill +y
-divider
```



```
geom:bar→dotplot
+stackgroup +method
-y
```



Syntax

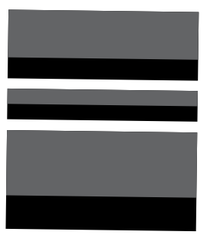
```
geom_mosaic
      x = cyl,
          mpg*
divider = hspine,
          hspine
```

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

```
geom_dotplot
        x = mpg*
     fill = cyl
stackgrp = TRUE
  method = histodot
```

Probabilistic Grammar of Graphics (PGoG)

Changes



Syntax

```
geom_bloc
  h <- P(mpg*)
       P(cyl|mpg*)
fill<- cyl
```

`mpg*`: discretized miles per gallon

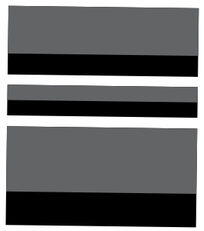
Cognitive ergonomics

Pro: Short edit distances, close to probability expressions

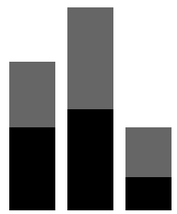
- *Low viscosity*
- *No premature committment*

Existing ggplot2 packages

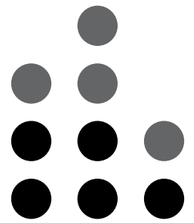
Changes



```
geom:mosaic→bar
+fill +y
-divider
```



```
geom:bar→dotplot
+stackgroup +method
-y
```



Syntax

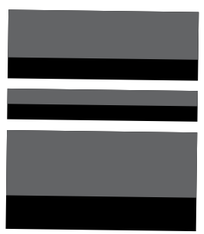
```
geom_mosaic
      x = cyl,
          mpg*
divider = hspine,
          hspine
```

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

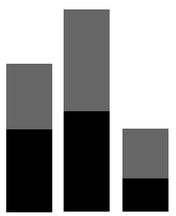
```
geom_dotplot
        x = mpg*
     fill = cyl
stackgrp = TRUE
  method = histodot
```

Probabilistic Grammar of Graphics (PGoG)

Changes



```
+x
```



Syntax

```
geom_bloc
  h <- P(mpg*)
       P(cyl|mpg*)
fill<- cyl
```

```
geom_bloc
  x <- mpg*
  h <- P(mpg*)
       P(cyl|mpg*)
fill<- cyl
```

`mpg*`: discretized miles per gallon

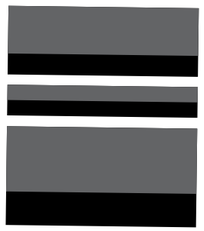
Cognitive ergonomics

Pro: Short edit distances, close to probability expressions

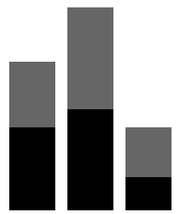
- *Low viscosity*
- *No premature commitment*

Existing ggplot2 packages

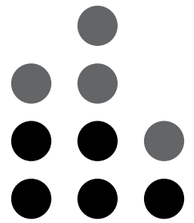
Changes



```
geom:mosaic→bar
+fill +y
-divider
```



```
geom:bar→dotplot
+stackgroup +method
-y
```



Syntax

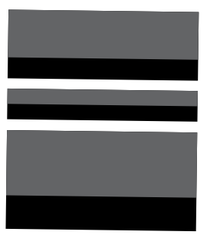
```
geom_mosaic
      x = cyl,
          mpg*
divider = hspine,
          hspine
```

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

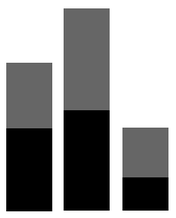
```
geom_dotplot
        x = mpg*
     fill = cyl
stackgrp = TRUE
  method = histodot
```

Probabilistic Grammar of Graphics (PGoG)

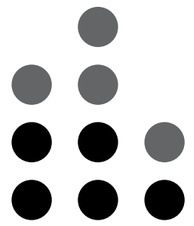
Changes



```
+x
```



```
geom:bloc→icon
```



Syntax

```
geom_bloc
  h <- P(mpg*)
       P(cyl|mpg*)
fill<- cyl
```

`mpg*`: discretized miles per gallon

```
geom_bloc
  x <- mpg*
  h <- P(mpg*)
       P(cyl|mpg*)
fill<- cyl
```

```
geom_icon
  x <- mpg
  h <- P(mpg*)
       P(cyl|mpg*)
fill<- cyl
```

Cognitive ergonomics

Con: specifying probability expressions can be difficult

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

P(mpg|cyl) P(mpg)?

Math **P(cyl|mpg) P(mpg)** = P(mpg,cyl)

Cognitive ergonomics

Con: specifying probability expressions can be difficult

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

P(mpg|cyl) P(mpg)?

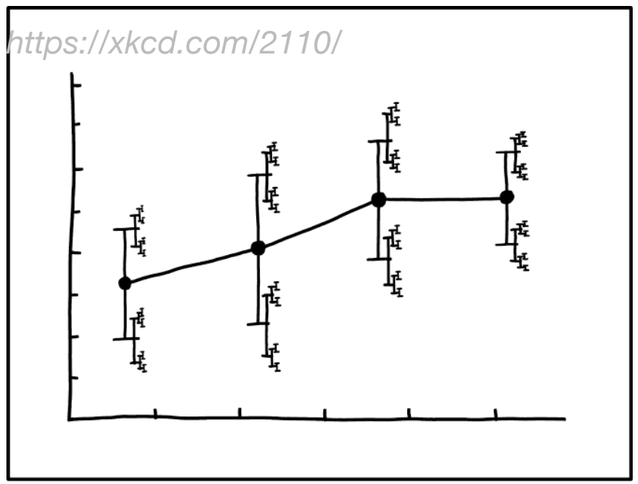
```
Math          P(cyl|mpg) P(mpg) = P(mpg,cyl)
Coord aes           x <- mpg
Prob aes    height <- P(cyl|mpg) P(mpg)
Visual aes     fill <- cyl
```

Further questions

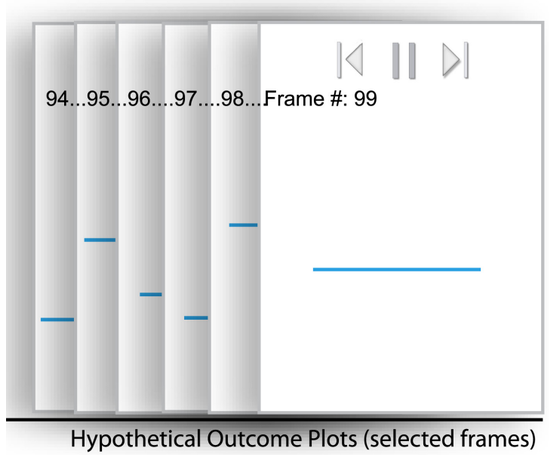
Based on *the Cognitive Dimensions of Notations*:

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

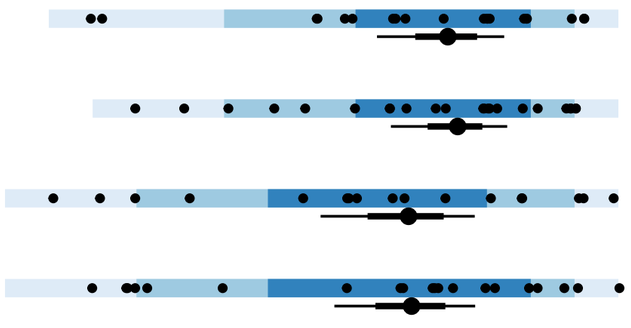
Future work: more uncertainty vizes -> more concepts



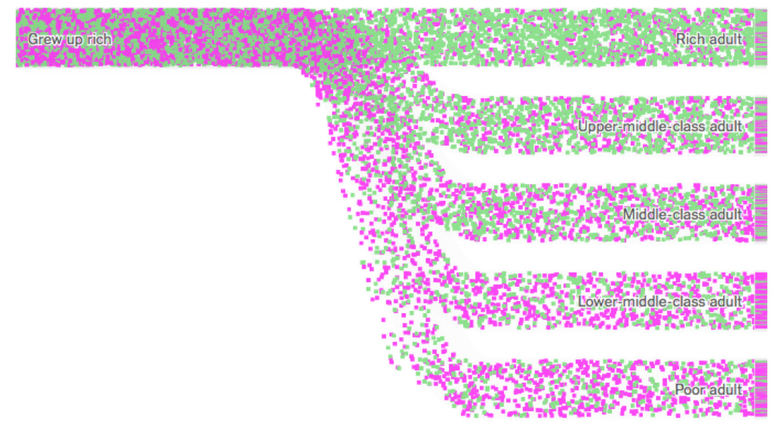
- Uncertainty sources: aleatory or epistemic
- Data structure: hierarchical, sequential, etc.
- Summary statistics, confidence intervals, etc.
- Visualization techniques such as linking



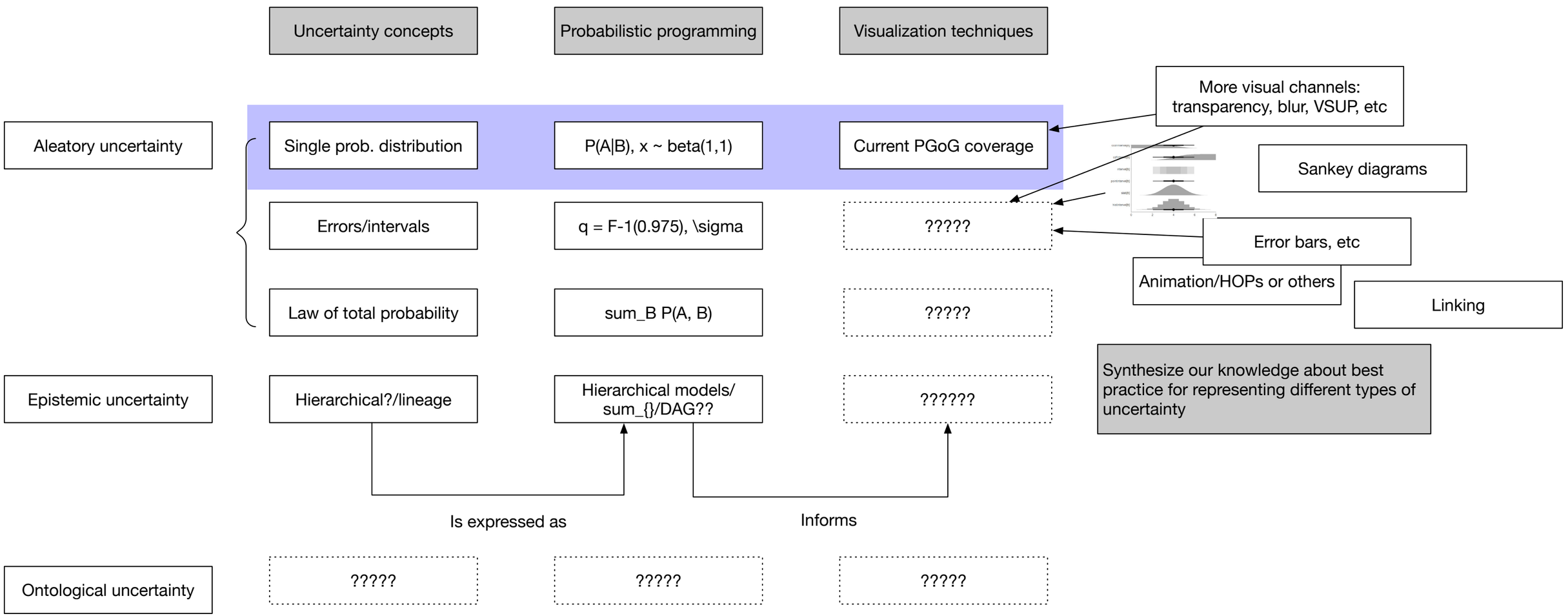
(Hullman, Resnick, and Adar 2015)



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

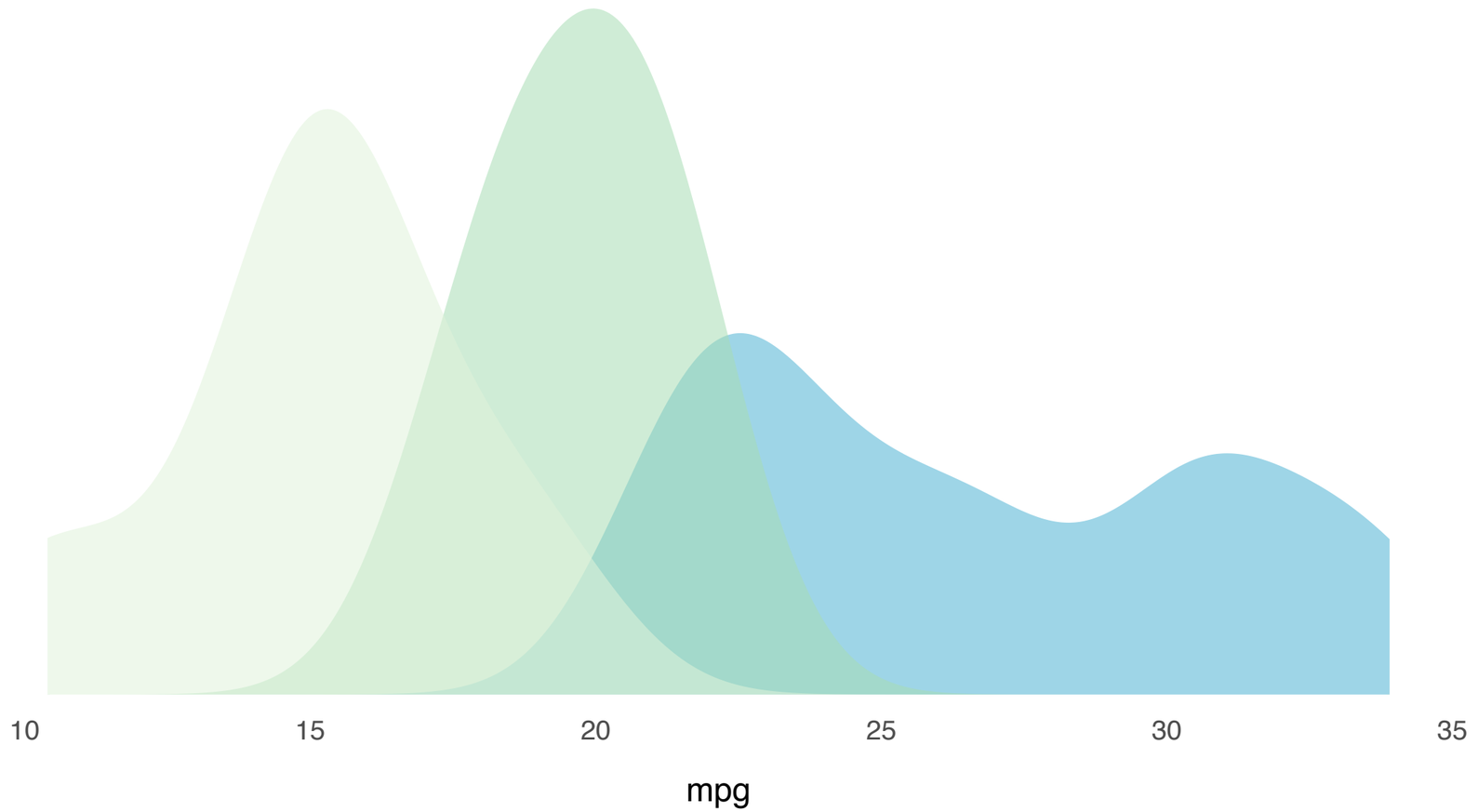


Future work: an overview inspired by uncertainty taxonomies



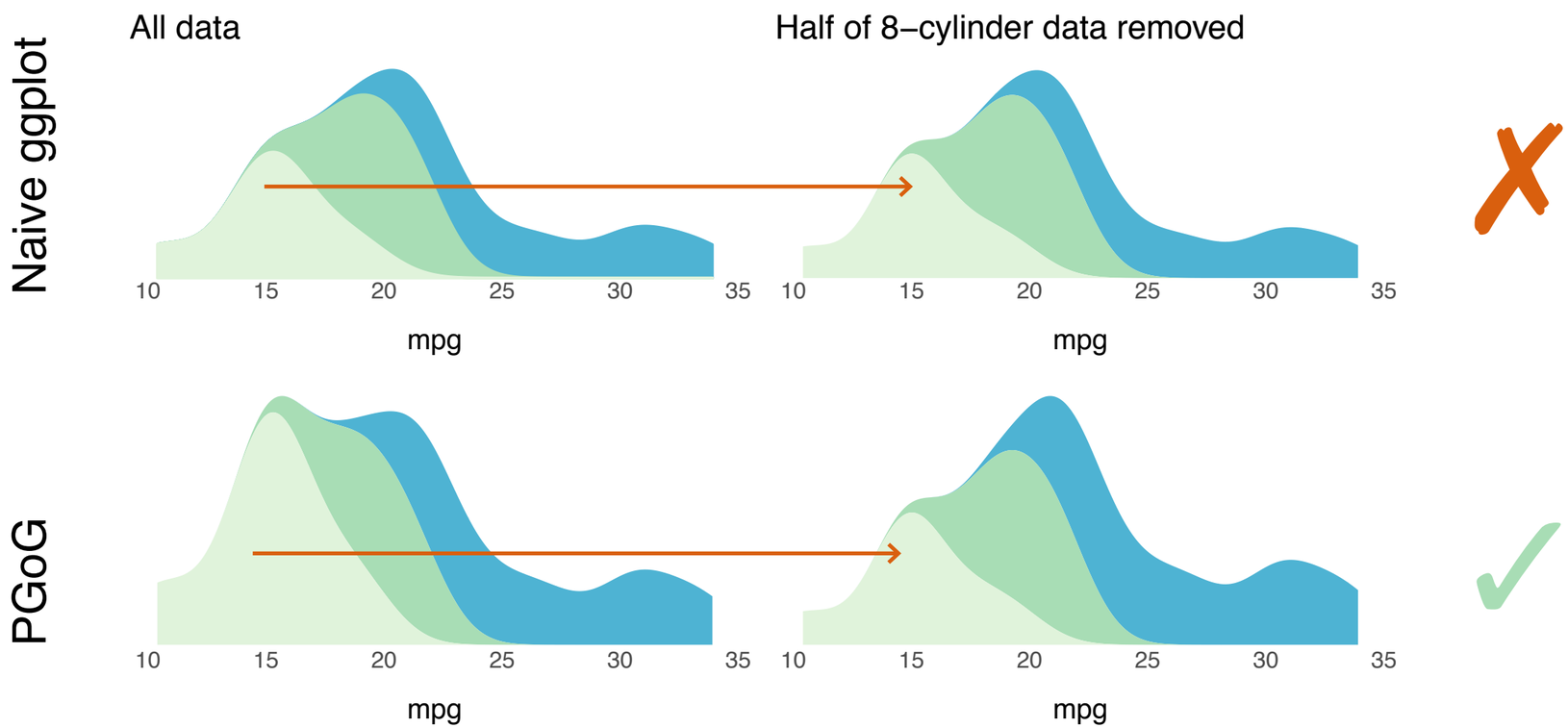
Conclusions

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(mpg|cyl)

Using an algebraic process for visualization design [kindlmann_algebraic_2014]



α: data symmetry ~ reduced 7/32 data points
ω: visualization symmetry ~ does the visualization change accordingly?
α !~= ω: violation of the principle of data-visualization correspondence

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?

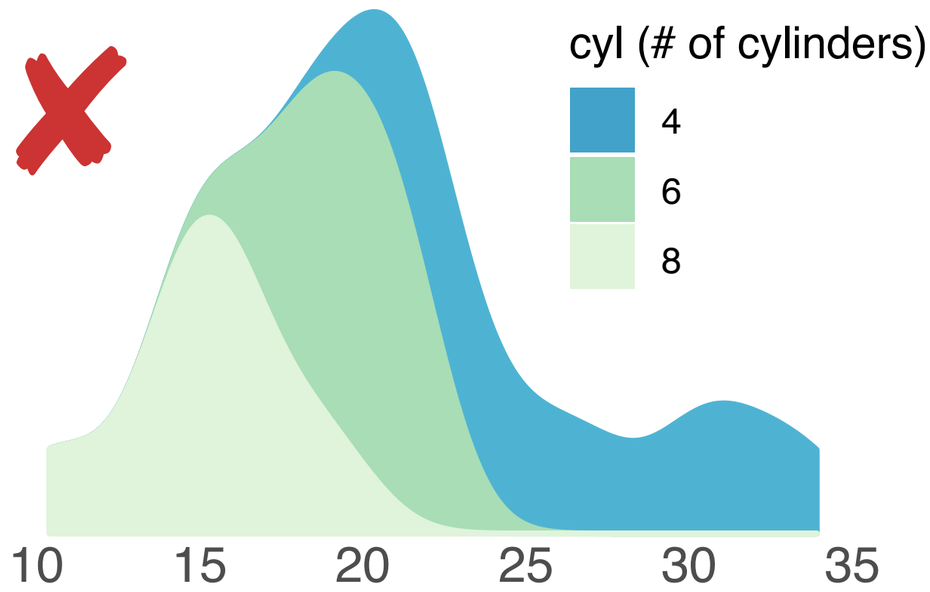
```
                   mpg cyl am
Mazda RX4         21.0   6  1
Mazda RX4 Wag     21.0   6  1
Datsun 710        22.8   4  1
Hornet 4 Drive    21.4   6  0
Hornet Sportabout 18.7   8  0
Valiant           18.1   6  0
```

```
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”)
```



Experimental evaluation challenges

- Recruiting participants with appropriate level of expertise: if they are experts in an existing system and learning the new PGoG, we might underestimate the expressiveness or usability of PGoG. Some measures such as correctness of visualizations produced might be less sensitive to this problem.
- Recruiting enough particpants: a power analysis will be conducted based on pilot studies. If for example N = 200 is needed, recruitment might be difficult, assuming in-person experiments.

Expert evaluation of the Grammar

- PGoG is new
- PGoG is a formalism/abstraction
- Using the PGoG prototype needs knowledge of ggplot

-> Expert evaluation is the appropriate first step

How we collected probabilistic visualizations in the wild

Methods of collection (informal)
- Google keyword searches: e.g., “icon array, faceted“
- Reputable news outlets
- “Best of” visualization awards/collections

Criteria for inclusion
- Does it describe a probability distribution (counts also OK)
- Is it systematically generated (as opposed to hand-drawn or with inconsistent data -> visual mapping)

How much does PGoG cover in the end?
- A lot, if judging by how much the PGoG aes/geoms apply
- A few, if “coverage” means strict replication of the original

- Viscosity: Resistance to Change
  - In layout-based grammar, need to rewrite layers, but for GoG, aesthetics mapping is easier to change
- Visibility: Ability to View Components Easily
  - Probabilities as first-class citizens!
- Premature Commitment: constraints on the Order of Doing Things
  - In PGoG, data, geoms, and aesthetics are quite independent
- Hidden dependencies
  - Probability rules that the user should know, like factoring a joint
- Role-expressiveness: The Purpose of an Entity is Readily Inferred
  - In addition to the probabilistic variables, PGoG has intuitively named aesthetics such as width and height, assuming that users understand the basic concepts of grammar of graphics
- Error-proneness: The Notation Invites Mistakes and the System Gives Little Protection
  - Foreseeable errors in probabilistic variables and implied mappings
- Abstraction: Types and Availability of Abstraction Mechanisms
  - None? PGoG implementation does not have a template-type functionality
- Secondary Notation: Extra Information in Means Other Than Formal Syntax
  - Yes, comments are supported in R implementation
- Closeness of Mapping: Closeness of Representation to Domain
  - P(A | B) yay
- Consistency: Similar Semantics Are Expressed in Similar Syntactic Forms
  - Yes? Every visualization is specified with similar syntax
- Diffuseness: Verbosity of Language
  - Yes? Only template-based visualization specs are more succinct, but they are not flexible
- Hard Mental Operations: High Demand on Cognitive Resources
  - For users familiar with probabilities, PGoG removes mental efforts in data wrangling and lower level vis specs.
- Provisionality: Degree of Commitment to Actions or Marks
  - Similar to premature commitment
- Progressive Evaluation: Work-to-Date Can be Checked at Any Time
  - Difficult, because a visualization must have data, geoms, and aesthetics specified. R implementation does provide parsed probability “chain”, however.