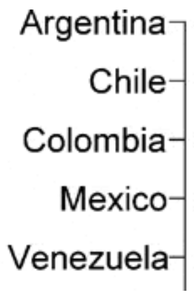
Information Visualization

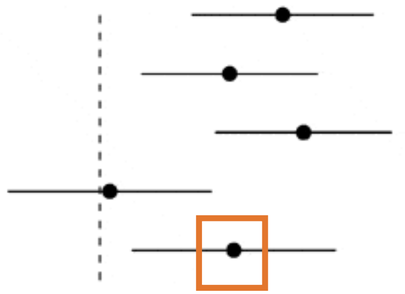
Uncertainty Visualization

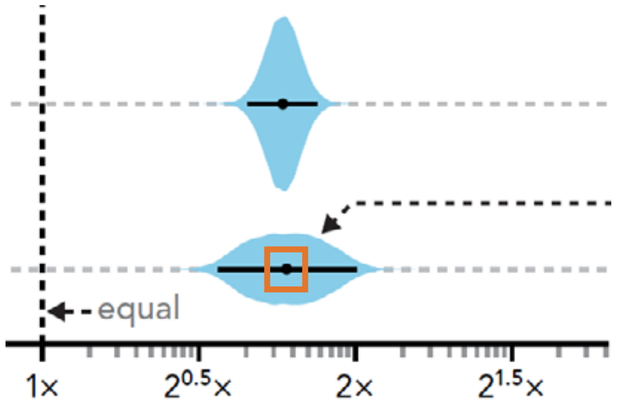
Dr. David Koop

People Ignore Uncertainty

Variable	Coefficient (Standard Error)
Constant	.41 (.93)
Countries	
Argentina	1.31 (.33)**B,M
Chile	.93 (.32)**B,M
Colombia	1.46 (.32)**B,M
Mexico	.07 (.32)A,CH,CO,V
Venezuela	.96 (.37)**B,M

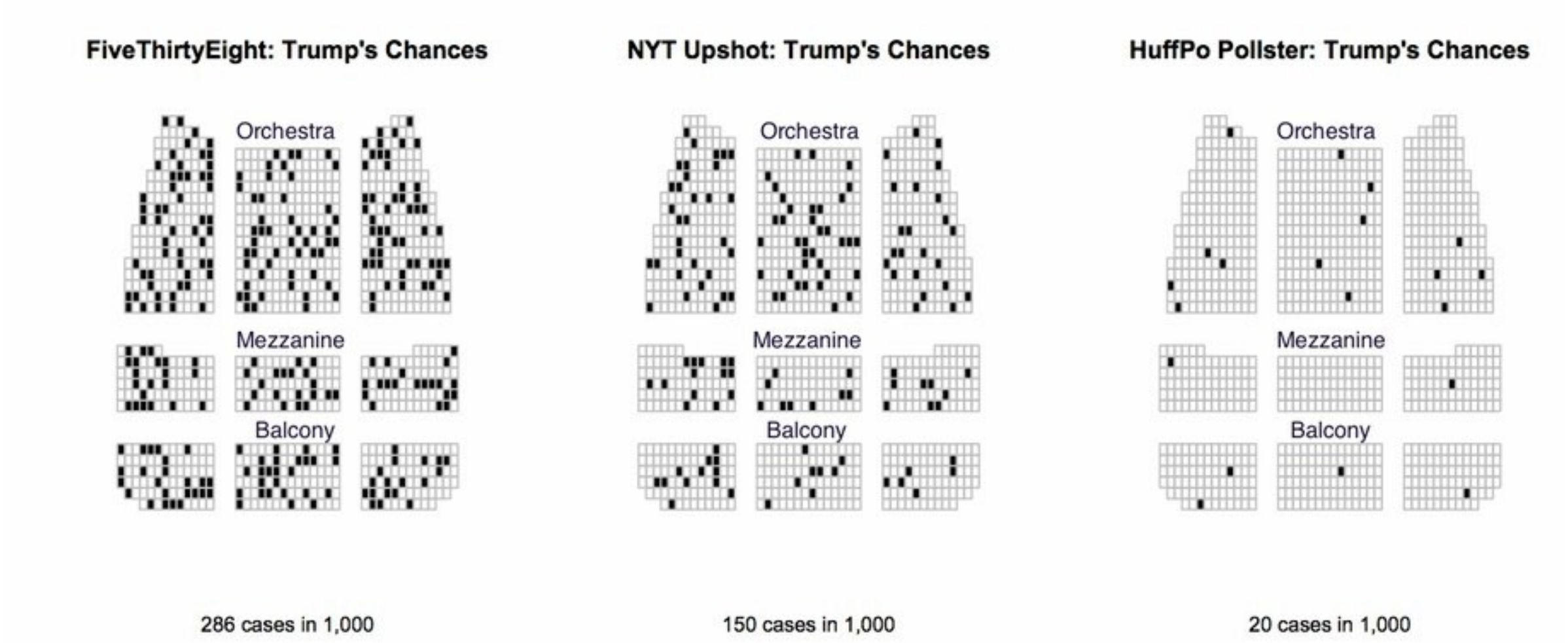






[M. Kay]

Better Ways to Present Uncertainty



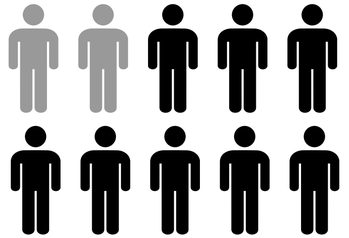
[J. H. Gross, Washington Post, 2016]

Graphical Annotations of Distributional Properties

Intervals and Ratios

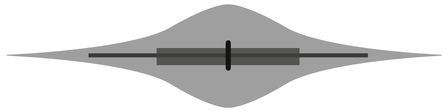
error bars

box plot

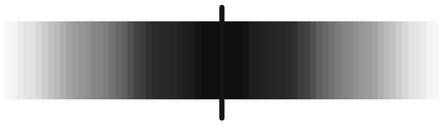


icon array

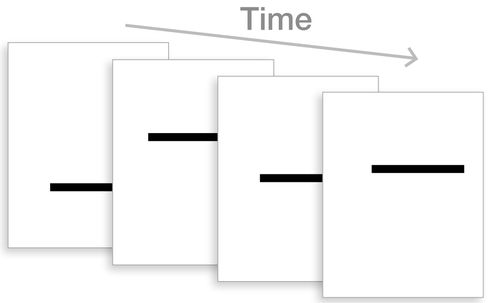
Distributions



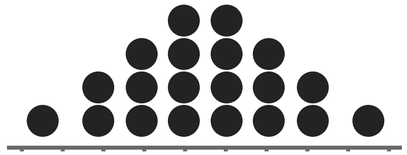
violin plot



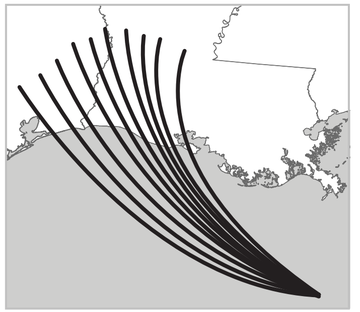
gradient plot



hypothetical outcome plot



quantile dot plot



ensemble plot

[L. Padilla et al.]

Paper

When (ish) is My Bus? User-centered Visualizations of Uncertainty in Everyday, Mobile Predictive Systems

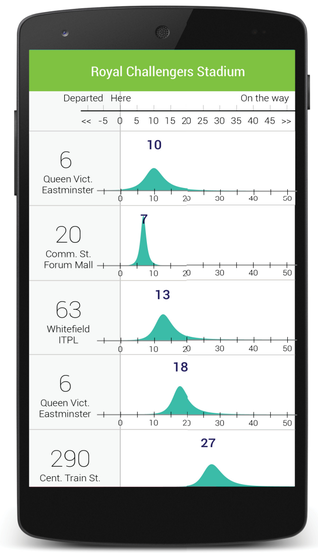
Matthew Kay
CSE | dub
University of Washington
mjskay@uw.edu

Tara Kola
Computer Science
Tufts University
tara.kola@tufts.edu

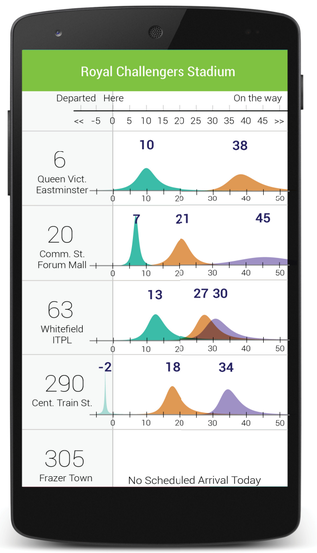
Jessica R. Hullman
iSchool | dub
University of Washington
jhullman@uw.edu

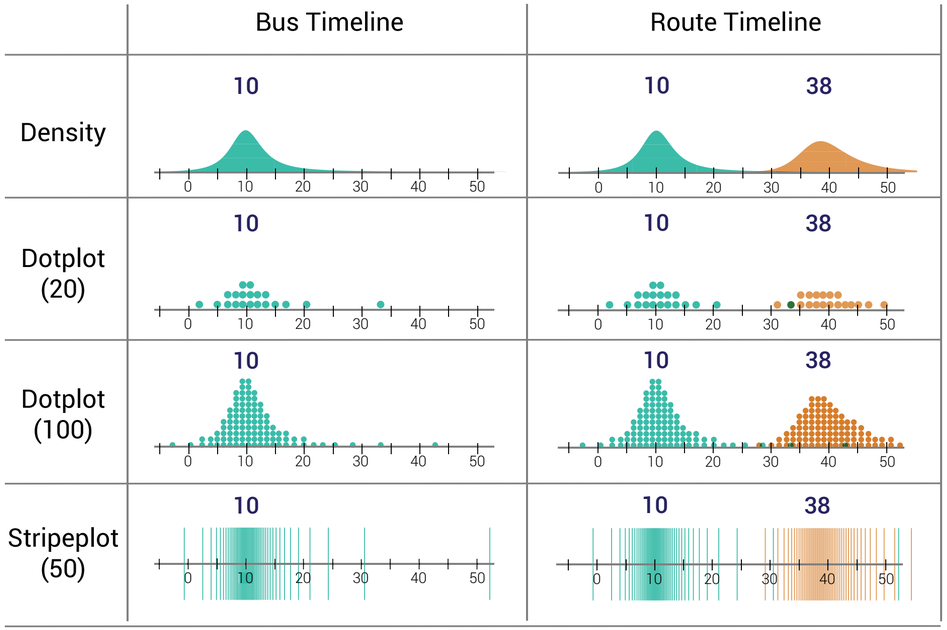
Sean A. Munson
HCDE | dub
University of Washington
smunson@uw.edu

Bus Timeline



Route Timeline



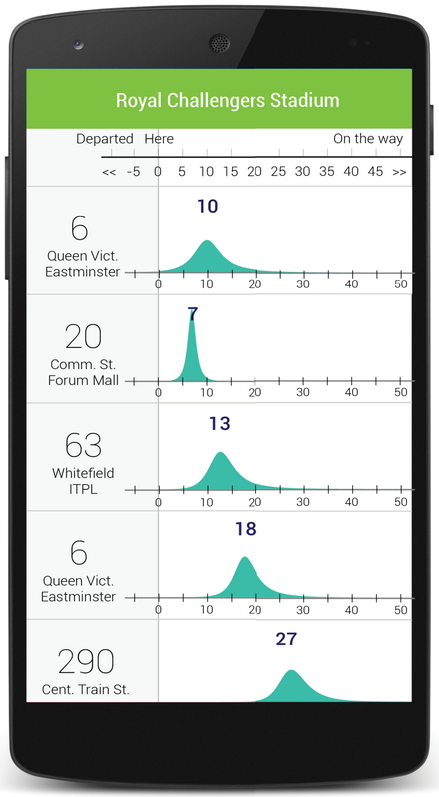


Address Questions for Bus Riders

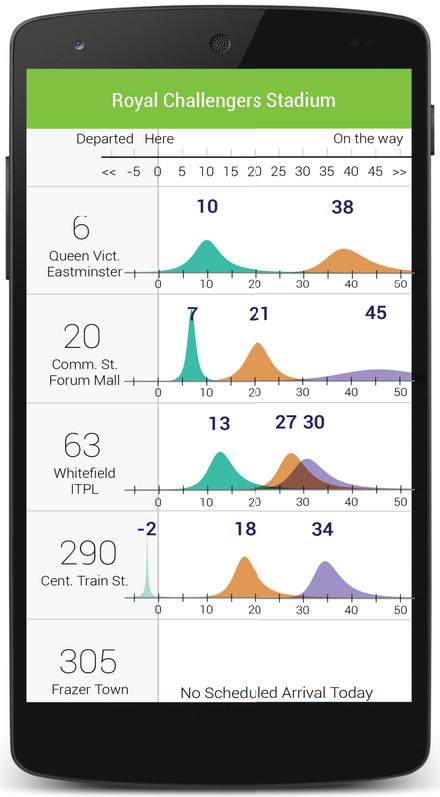
- When is the next bus?
- Has it already departed?
- Variance of predictions
- Schedule frequency

- Two views:
  - Bus Timeline: one vis per bus
  - Route Timeline: one vis per route

Bus Timeline



Route Timeline

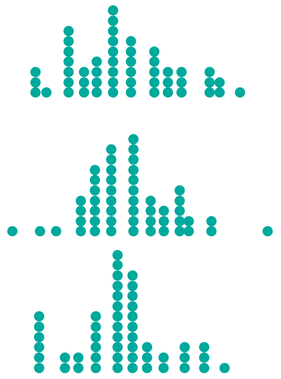


[Kay et al., 2016]

Random Draws vs. Quantiles

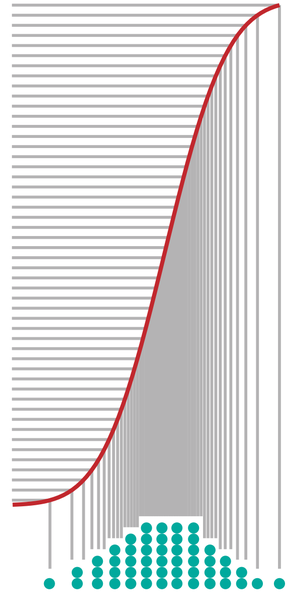
Probability density of Normal distribution

To generate a discrete plot of this distribution, we could try taking **random draws** from it. However, **this approach is noisy**: it may be very different from one instance to the next.



Instead, we use the **quantile function (inverse CDF)** of the distribution to generate "draws" from evenly-spaced quantiles.

We plot the quantile "draws" using a Wilkinsonian dotplot, yielding what we call a **quantile dotplot**: a consistent discrete representation of a probability distribution.

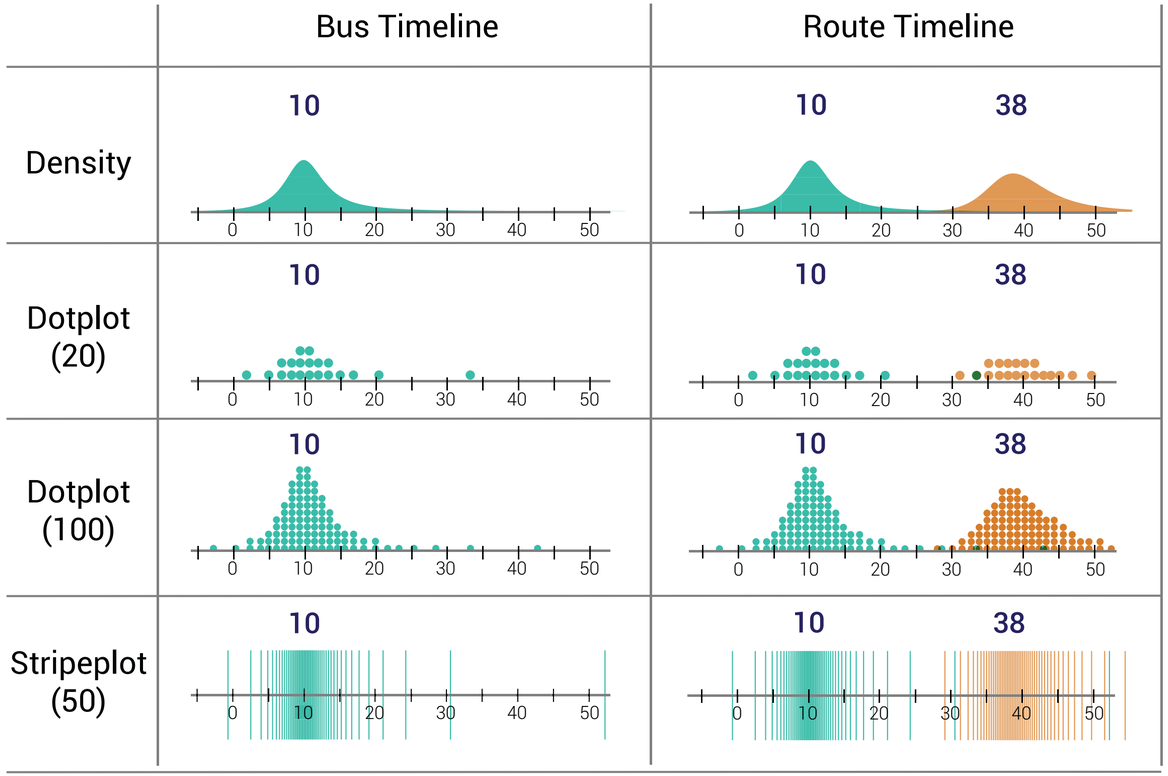


By using quantiles we facilitate interval estimation from frequencies: e.g., knowing there are 50 dots here, if we are willing to miss our bus **3/50** times, we can count **3 dots** from the left to get a one-sided **94% (1 – 3/50) prediction interval** corresponding to that risk tolerance.

[Kay et al., 2016]

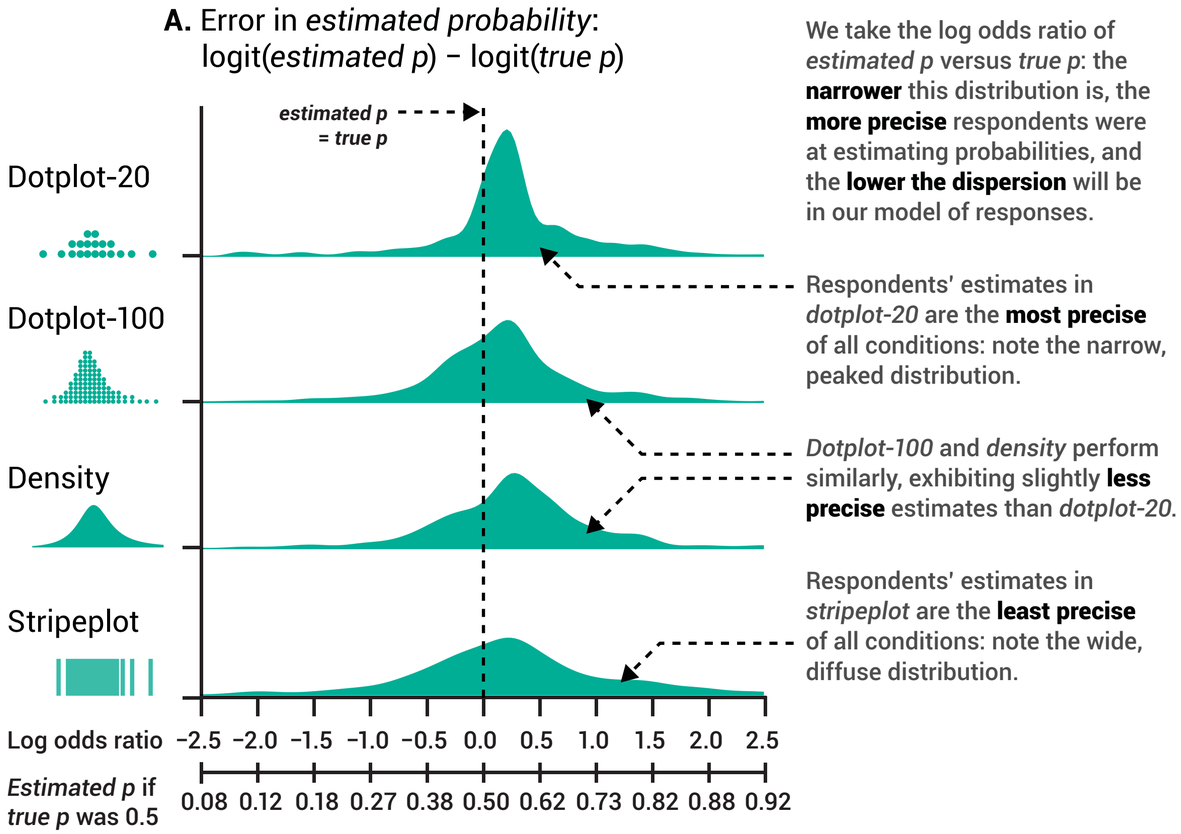
Different Encodings

	Density	Stripeplot	Density+ Stripeplot	Dotplot(20)	Dotplot(50)	Dotplot(100)
shows discrete, countable events		●	●	●	●	●
fast counting in tails		●		●	●	●
fast counting in body				●		
directly estimate density	●	●	●		●	●
directly estimate quantlies		●	●	●	●	●
tight densities drawn consistently		●		●		
project to axis		●				
easily assess range (min/max)	●	●			●	●
easily assess mode	●		●	●	●	●



[Kay et al., 2016]

Evaluation Results

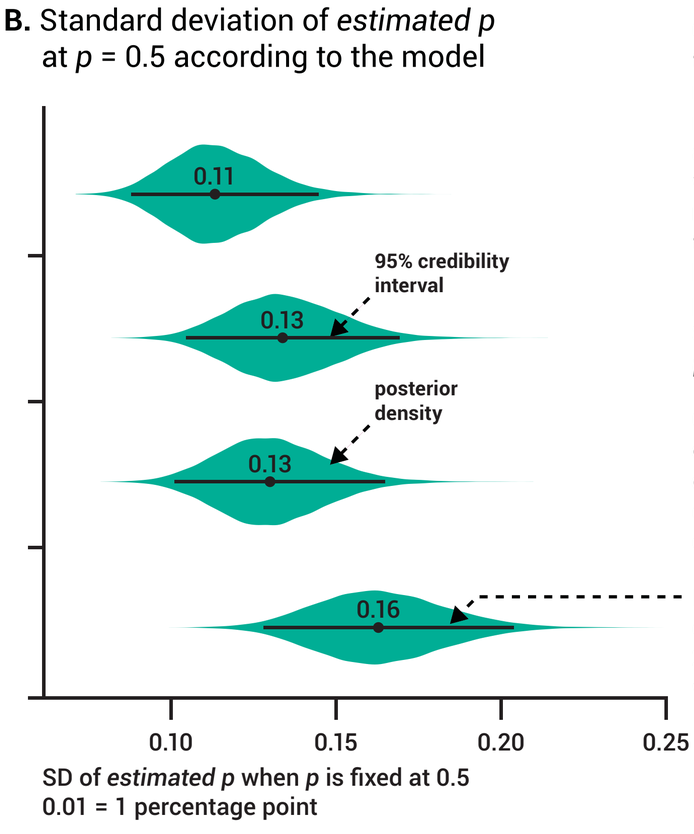


We take the log odds ratio of *estimated p* versus *true p*: the **narrower** this distribution is, the **more precise** respondents were at estimating probabilities, and the **lower the dispersion** will be in our model of responses.

Respondents' estimates in *dotplot-20* are the **most precise** of all conditions: note the narrow, peaked distribution.

Dotplot-100 and *density* perform similarly, exhibiting slightly **less precise** estimates than *dotplot-20*.

Respondents' estimates in *stripeplot* are the **least precise** of all conditions: note the wide, diffuse distribution.



Our fitted model can estimate the **standard deviation (SD)** of respondents' *estimated p*. This measures how precise people are at estimating probability intervals with each visualization: **lower is more precise**. An SD of 0.01 means the average respondent's estimates have a standard deviation of 1 percentage point when $p = 0.5$. SD depends on p, so we fix p at 0.5 for these estimates.

Our Bayesian model gives a **95% credibility interval** and a **posterior density** for each SD, capturing the estimated SD and the precision of our estimate of it.

For example, we estimate the SD of *estimated p* to be **between 13 and 20 percentage points** in *stripeplot* when $p = 0.5$.

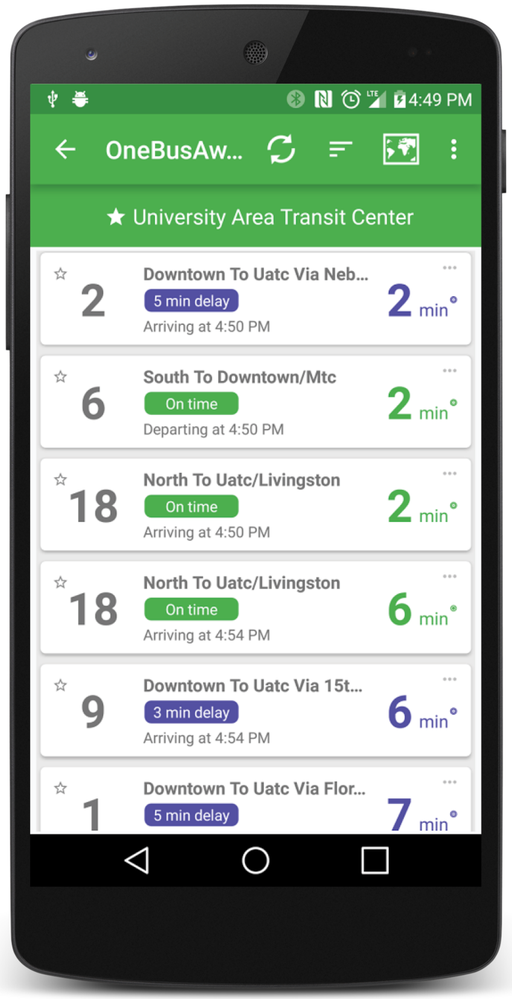
[Kay et al., 2016]

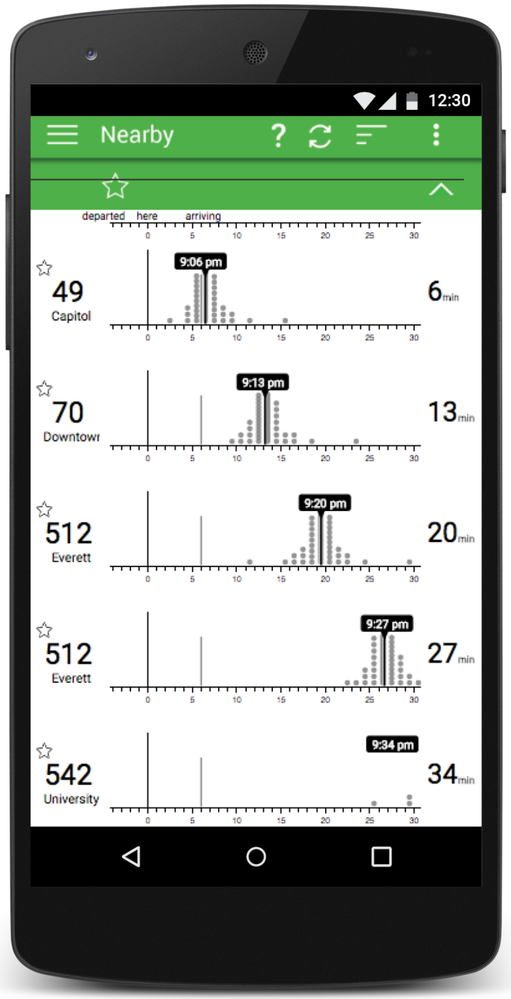
Study Results

- Use smaller numbers with discrete outcome displays (dotplot-20 instead of dotplot-100)
- Precision must be balanced with glanceability
- Visual appeal questions?
- Some people didn't like the uncertainty

[Kay et al., 2016]

Expanded Study



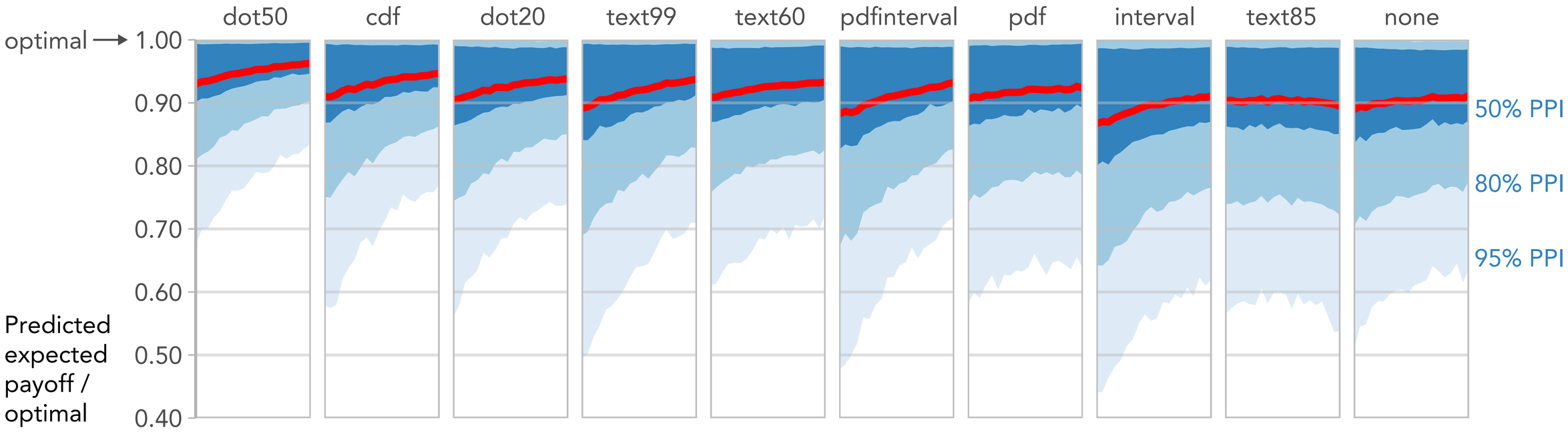


- Compare
  - no uncertainty
  - textual uncertainty
  - uncertainty visualization

[Fernandes et al., 2018]

Study Results: Visualizations best

1. **Posterior predictive intervals** and predicted **mean** for performance in each condition. These intervals are what we would predict 50%, 80%, or 95% of new observations of performance to fall into. Performance improves with additional trials, especially for dot50, cdf, and dot20. Meanwhile, performance in text depends on the risk threshold, with text85 performing similarly to no uncertainty.



[Fernandes et al., 2018]

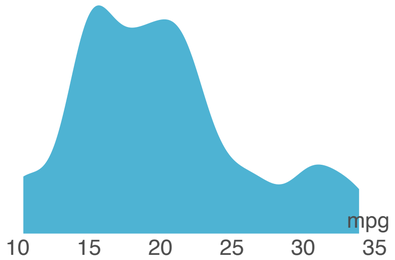
Schedule

- Today: Progress Reports & Uncertainty
- Next Tuesday: Surveys Due & Presentations
  - Focus on Overview and Categorization/Organization
- Tuesday, Oct. 26: No Class
- Thursday, Oct. 28: High-Dimensional Data Critique Due

Progress Reports

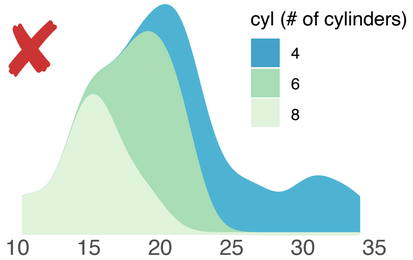
Need for better probabilistic graphics

a) P(mpg) in ggplot




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

b) Naive ggplot



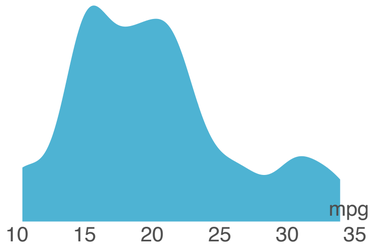

```
ggplot(mtcars) +
geom_density(
   aes(x = mpg,
       y = stat(density),
    fill = cyl),
position = "stack")
```

- A *probabilistic visualization* is **correct** if the proportions of visual elements (such as counts or areas) and their spatial placement reflect the underlying probability distribution, including any conditional probabilities or part-to-whole relationships

[Pu and Kay, 2020]

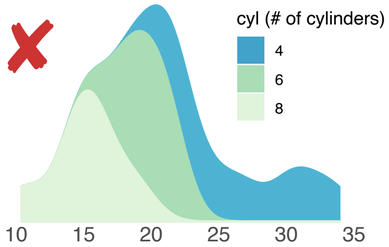
Need for better probabilistic graphics

a) P(mpg) in ggplot




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

b) Naive ggplot




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

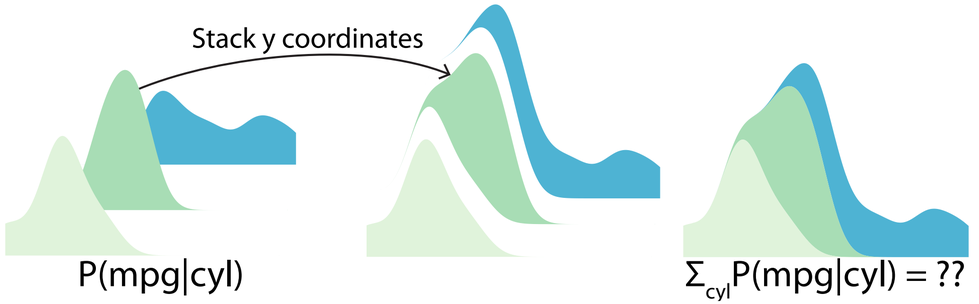
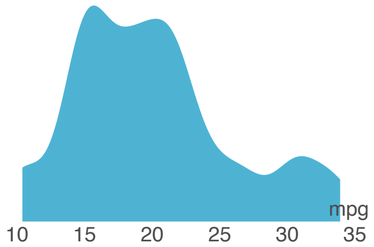


Figure 4. How `ggplot2` constructs Figure 2.b. Left: the system first computes three density estimates, each of unit area, that represent $P(mpg|cyl)$ (cyl = 4, 6 and 8). Right: `ggplot2` stacks the densities naively (`position = stack`), creating an incorrect figure: the viewer might interpret there to be roughly 1/3 8-cylinder cars, while in the data, this proportion is 43%. In terms of probability notations, stacking creates a part-whole relationship and thus implies summation, but $\sum_{cyl} P(mpg|cyl)$ does not simplify to $P(mpg)$.

[Pu and Kay, 2020]

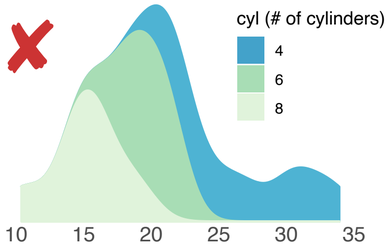
Need for better probabilistic graphics

a) P(mpg) in ggplot



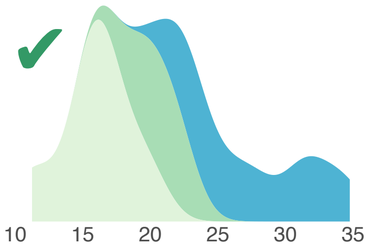

```
ggplot(mtcars) +
geom_density(
 aes(x = mpg,
     y = stat(density)
))
```

b) Naive ggplot



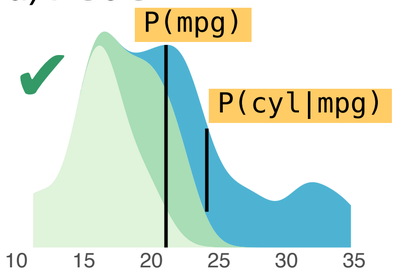

```
ggplot(mtcars) +
geom_density(
   aes(x = mpg,
       y = stat(density),
    fill = cyl),
position = "stack")
```

c) ggplot: corrected




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

d) PGoG




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

[Pu and Kay, 2020]

Probabilistic Grammar of Graphics

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

- Adds probabilistic aesthetics
- Discrete variables: probability mass function (pmf)
- Continuous variables: probability density function (pdf)
- Use other aesthetics (e.g. transparency, blur, animation)
- Can be used to better study effect of frequency formats

[Pu and Kay, 2020]