Tufte Without Tears

Flexible tools for visual exploration and presentation of statistical models

Christopher Adolph

Department of Political Science

and

Center for Statistics and the Social Sciences

University of Washington, Seattle

Edward Tufte & Scientific Visuals

In several beautifully illustrated books Edward Tufte gives the following advice:

1. **Assume** the reader's interest and **intelligence**

Edward Tufte & Scientific Visuals

In several beautifully illustrated books Edward Tufte gives the following advice:

1. **Assume** the reader's interest and **intelligence**
2. Maximize information, minimize ink & space: the **data-ink ratio**

Edward Tufte & Scientific Visuals

In several beautifully illustrated books Edward Tufte gives the following advice:

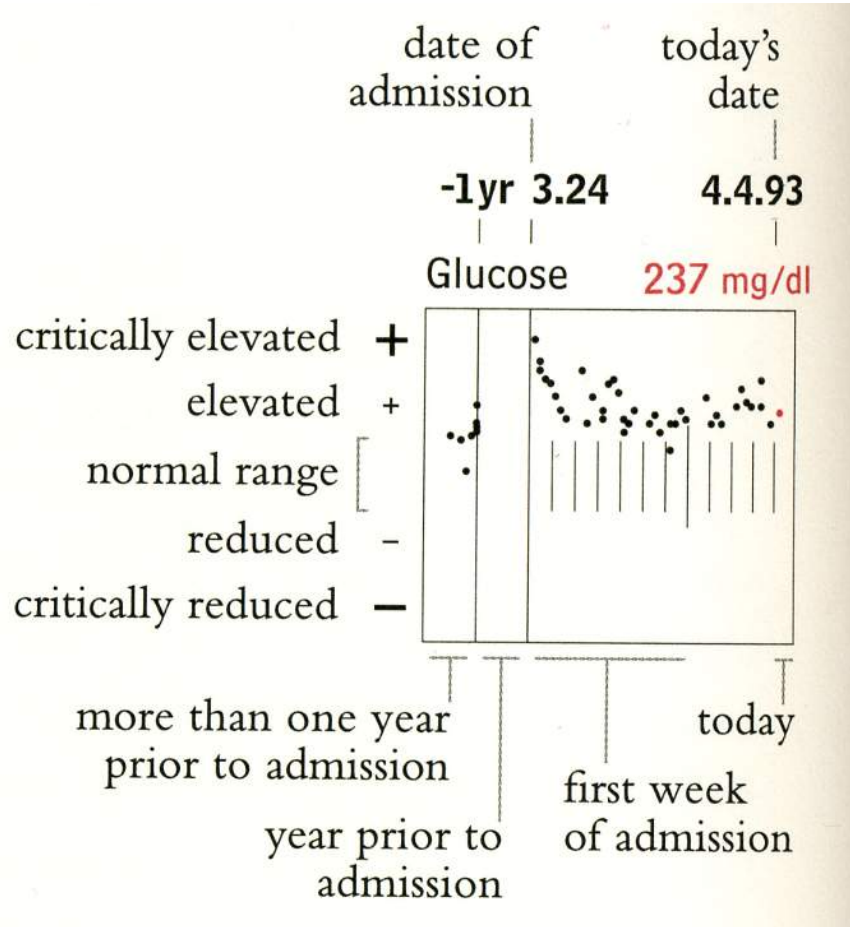
1. **Assume** the reader's interest and **intelligence**
2. Maximize information, minimize ink & space: the **data-ink ratio**
3. Show the underlying data and **facilitate comparisons**

Edward Tufte & Scientific Visuals

In several beautifully illustrated books Edward Tufte gives the following advice:

1. **Assume** the reader's interest and **intelligence**
2. Maximize information, minimize ink & space: the **data-ink ratio**
3. Show the underlying data and **facilitate comparisons**
4. Use **small multiples**: repetitions of a basic design

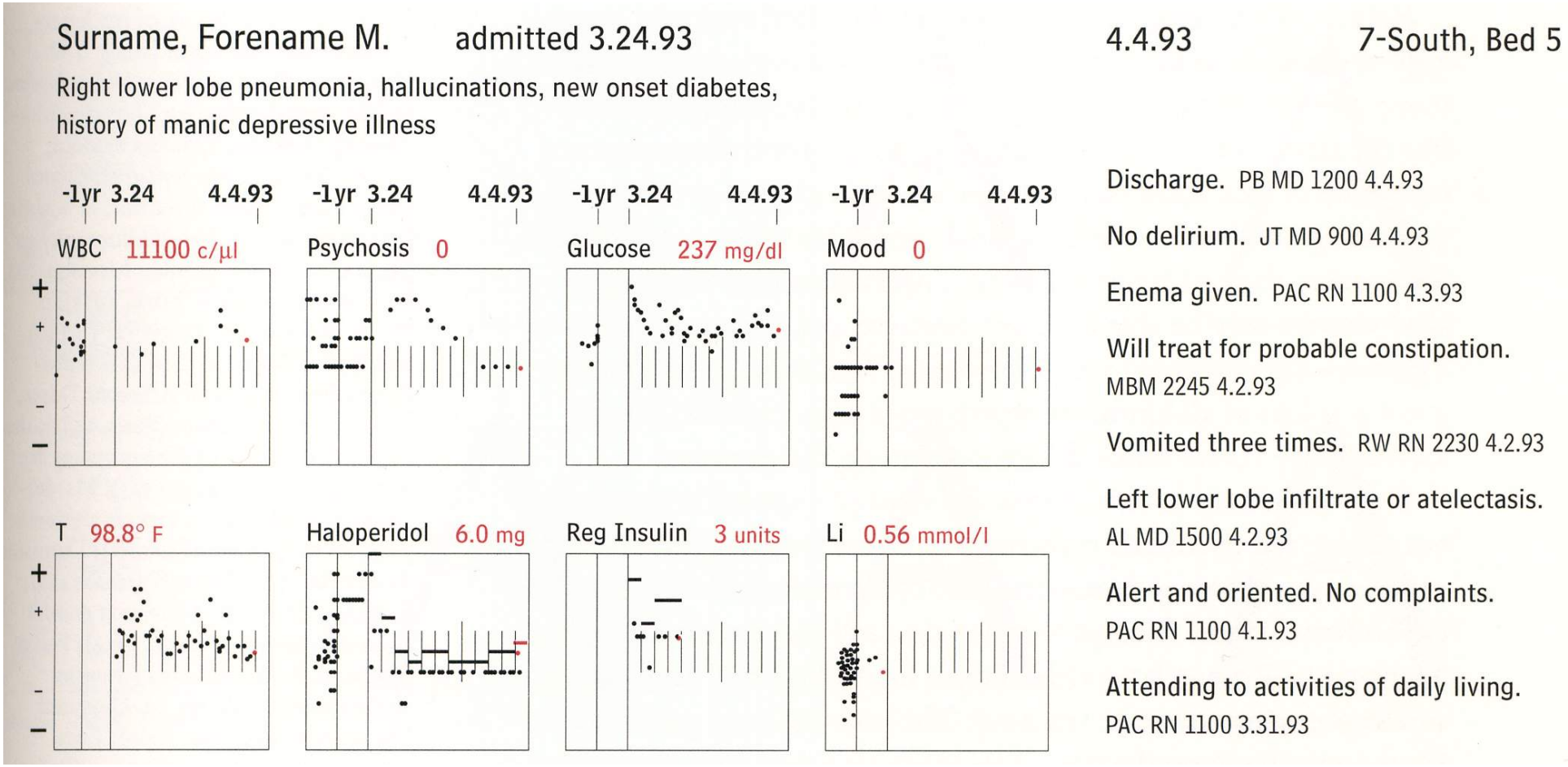
Tufte's recommended medical chart illustrates many of these ideas:



This chart is annotated for pedagogical purposes

Lots of information; little distracting scaffolding

A model that can be repeated once learned. . .



A complete layout using small multiples to convey lots of info

Elegant, information-rich. . . and hard to make

What most discussions of statistical graphics leave out

Tufte's books have had a huge impact on information visualization

However, they have two important limits:

Modeling Most examples are either exploratory or very simple models;

Social scientists want cutting edge applications

Tools Need to translate aesthetic guidelines into software

Social scientists are unlikely to do this on their own—
and shouldn't have to!

Key problem Ready-to-use techniques to visually present model results:

- for many variables

Key problem Ready-to-use techniques to visually present model results:

- for many variables
- for many robustness checks

Key problem Ready-to-use techniques to visually present model results:

- for many variables
- for many robustness checks
- showing uncertainty

Key problem Ready-to-use techniques to visually present model results:

- for many variables
- for many robustness checks
- showing uncertainty
- without accidental extrapolation

Key problem Ready-to-use techniques to visually present model results:

- for many variables
- for many robustness checks
- showing uncertainty
- without accidental extrapolation
- for an audience without deep statistical knowledge

Not covered here The theory behind effective visual display of data

Visual displays for data (not model) exploration

For these and other topics, and a reading list, see my course at `faculty.washington.edu/cadolph/vis`

Lots of examples... Too many if we need to discuss methods in detail

Who votes in American elections?

Source: King, Tomz, and Wittenberg

Method: Logistic regression

What do alligators eat?

Source: Agresti

Method: Multinomial logit

How do Chinese leaders gain power?

Source: Shih, Adolph, and Liu

Method: Bayesian model of partially observed ranks

When do governments choose liberal or conservative central bankers?

Source: Adolph

Method: Zero-inflated compositional data model

What explains the tier of European governments controlling health policies?

Source: Adolph, Greer, and Fonseca

Method: Multilevel multinomial logit

Presenting Estimated Models in Social Science

Most empirical work in social science is regression model-driven, with a focus on conditional expectation

Our regression models are

- full of covariates
- often non-linear
- usually involve interactions and transformations

If there is anything we need to visualize well, it is our models

Yet we often just print off tables of parameter estimates

Limits readers' *and analysts'* understanding of the results

Coefficients are not enough

Some limits of typical presentations of statistical results:

- Everything written in terms of arcane intermediate quantites
  (for most people, this includes logit coefficients)
- Little effort to transform results to the scale of the quantities of interest
  $\longrightarrow$ really want the conditional expectation, $\mathrm{E}(y|x)$
- Little effort to make informative statements about estimation uncertainty
  $\longrightarrow$ really want to know how uncertain is $\mathrm{E}(y|x)$
- Little visualization at all, or graphs with low data-ink ratios

Voting Example (Logit Model)

We will explore a simple dataset using a simple model of voting

People either vote ($\text{Vote}_i = 1$), or they don't ($\text{Vote}_i = 0$)

Many factors could influence turn-out; we focus on age and education

Data from National Election Survey in 2000. "Did you vote in 2000 election?"

```
      vote00 age hsdeg coldeg
 [1,]      1  49     1      0
 [2,]      0  35     1      0
 [3,]      1  57     1      0
 [4,]      1  63     1      0
 [5,]      1  40     1      0
 [6,]      1  77     0      0
 [7,]      0  43     1      0
 [8,]      1  47     1      1
 [9,]      1  26     1      1
[10,]      1  48     1      0
 ...
```

Logit of Decision to Vote, 2000 Presidential NES

	est.	s.e.	*p*-value
Age	0.074	0.017	0.000
Age^2	−0.0004	0.0002	0.009
High School Grad	1.168	0.178	0.000
College Grad	1.085	0.131	0.000
Constant	−3.05	0.418	0.000

Age enters as a quadratic to allow the probability of voting to first rise and eventually fall over the life course

Results look sensible, but what do they mean?

Which has the bigger effect, age or education?

What is the probability a specific person will vote?

An alternative to printing eye-glazing tables

1. Run your model as normal. Treat the output as an intermediate step.

An alternative to printing eye-glazing tables

1. Run your model as normal. Treat the output as an intermediate step.
2. Translate your model results back into the scale of the response variable
   - Modeling war? Show the change in probability of war associated with X
   - Modeling counts of crimes committed? Show how those counts vary with X
   - Unemployment rate time series? Show how a change in X shifts the unemployment rate over the following t years

An alternative to printing eye-glazing tables

1. Run your model as normal. Treat the output as an intermediate step.
2. Translate your model results back into the scale of the response variable
   - Modeling war? Show the change in probability of war associated with X
   - Modeling counts of crimes committed? Show how those counts vary with X
   - Unemployment rate time series? Show how a change in X shifts the unemployment rate over the following t years
3. Calculate or simulate the uncertainty in these final quantities of interest

An alternative to printing eye-glazing tables

1. Run your model as normal. Treat the output as an intermediate step.
2. Translate your model results back into the scale of the response variable
   - Modeling war? Show the change in probability of war associated with X
   - Modeling counts of crimes committed? Show how those counts vary with X
   - Unemployment rate time series? Show how a change in X shifts the unemployment rate over the following t years
3. Calculate or simulate the uncertainty in these final quantities of interest
4. Present visually as many scenarios calculated from the model as needed

A bit more formally...

We want to know the behavior of $\mathrm{E}(y|x)$ as we vary x.

In non-linear models with multiple regressors, this gets tricky.

The effect of x_1 depends on all the other x's and $\hat{\beta}$'s

Generally, we will need to make a set of "counterfactual" assumptions:
$x_1 = a, \qquad x_2 = b, \qquad x_3 = c, \quad \ldots$

- Choose $a, b, c, \ldots$ to match a particular counterfactual case of interest *or*
- Hold all but one of the x's at their mean values (or other reference baseline), then systematically vary the remaining x.

The same trick works if we are after differences in y related to changes in x, such as $\mathrm{E}(y_{\mathrm{scen1}} - y_{\mathrm{scen2}}|x_{\mathrm{scen1}}, x_{\mathrm{scen2}})$

Calculating quantities of interest

Our goal to obtain "quantities of interest," like

- Expected Values: $\mathrm{E}(Y|X_c)$
- Differences: $\mathrm{E}(Y|X_{c2}) - \mathrm{E}(Y|X_{c1})$
- Risk Ratios: $\mathrm{E}(Y|X_{c2})/\mathrm{E}(Y|X_{c1})$
- or any other function of the above

for some counterfactual X_c's.

For our Voting example, that's easy—just plug X_c into

$$\mathrm{E}(Y|X_c) = \frac{1}{1 + \exp(-X_c\beta)}$$

Getting confidence intervals is harder, but there are several options:

- For maximum likelihood models,
  simulate the response conditional on the regressors

  See King, Tomz, and Wittenberg, 2000, *American Journal of Political Science*, and the `Zelig` package for `R` or `Clarify` for Stata.

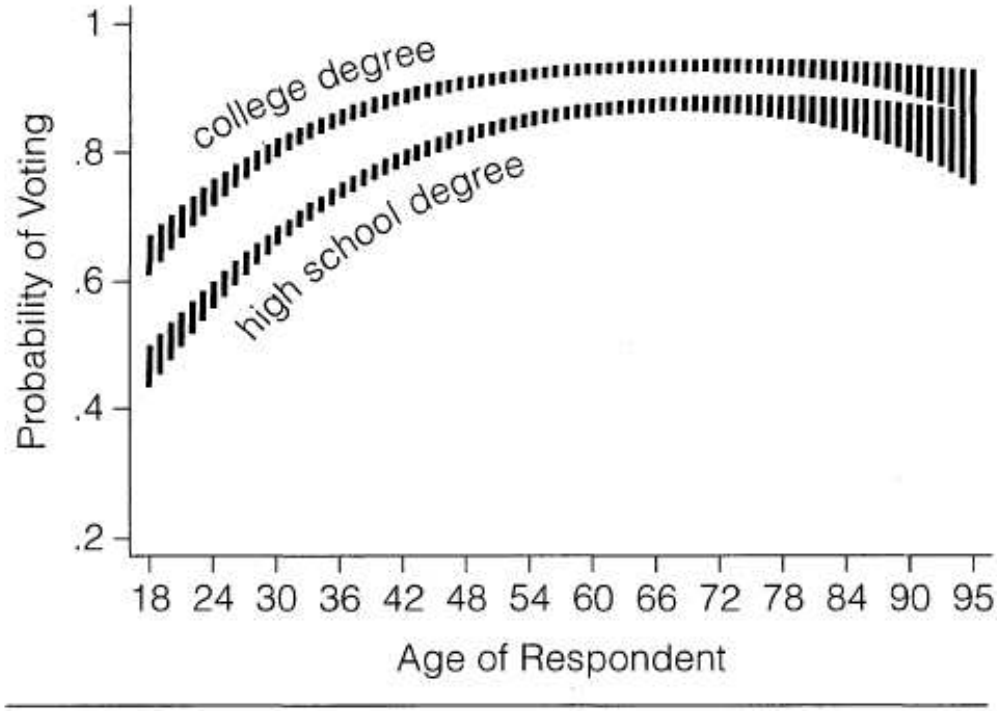
  These simulations can easily be summarized as CIs: sort them and take percentiles

- For Bayesian models, usual model output *is* a set of posterior draws

  See Andrew Gelman and Jennifer Hill, 2006, *Data Analysis Using Hierarchical/ Multilevel Models*, Cambridge UP.

Once we have the quantities of interest and confidence intervals, we're ready to make some graphs. . . but how?

FIGURE 1 Probability of Voting by Age

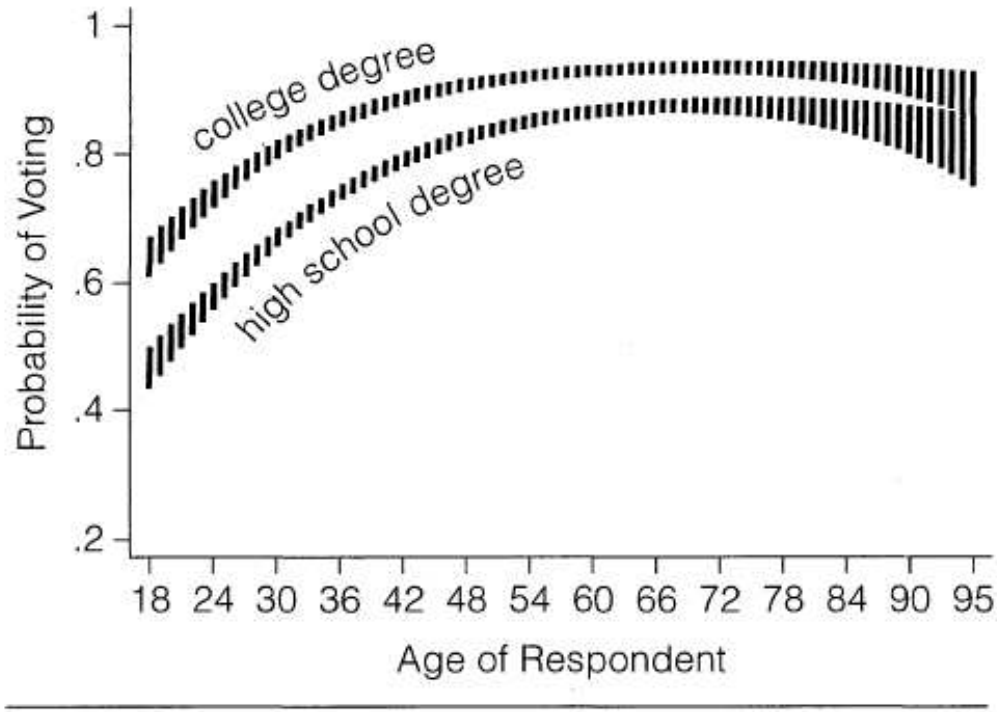


Vertical bars indicate 99-percent confidence intervals

Here is the graph that King, Tomz, and Wittenberg created for this model

How would we make this?

FIGURE 1 Probability of Voting by Age



Vertical bars indicate 99-percent confidence intervals

We could use the default graphics in `Zelig` or `Clarify` (limiting, not as nice as the above)

Or we could do it by hand (hard)

Wanted: an easy-to-use `R` package that

1. takes as input the *output* of estimated statistical models
2. makes a variety of plots for model interpretation
3. plots "triples" (lower, estimate, upper) from estimated models well
4. lays out these plot in a tiled arrangement (small multiples)
5. takes care of axes, titles, and other fussy details

Wanted: an easy-to-use `R` package that

1. takes as input the *output* of estimated statistical models
2. makes a variety of plots for model interpretation
3. plots "triples" (lower, estimate, upper) from estimated models well
4. lays out these plot in a tiled arrangement (small multiples)
5. takes care of axes, titles, and other fussy details

With considerable work, one could

- coerce `R`'s basic graphics to do this badly
- or get `lattice` to do this fairly well for a specific case

But an easy-to-use, general solution is lacking

The `tile` package

My answer is the `tile` package, written using `R`'s `grid` graphics

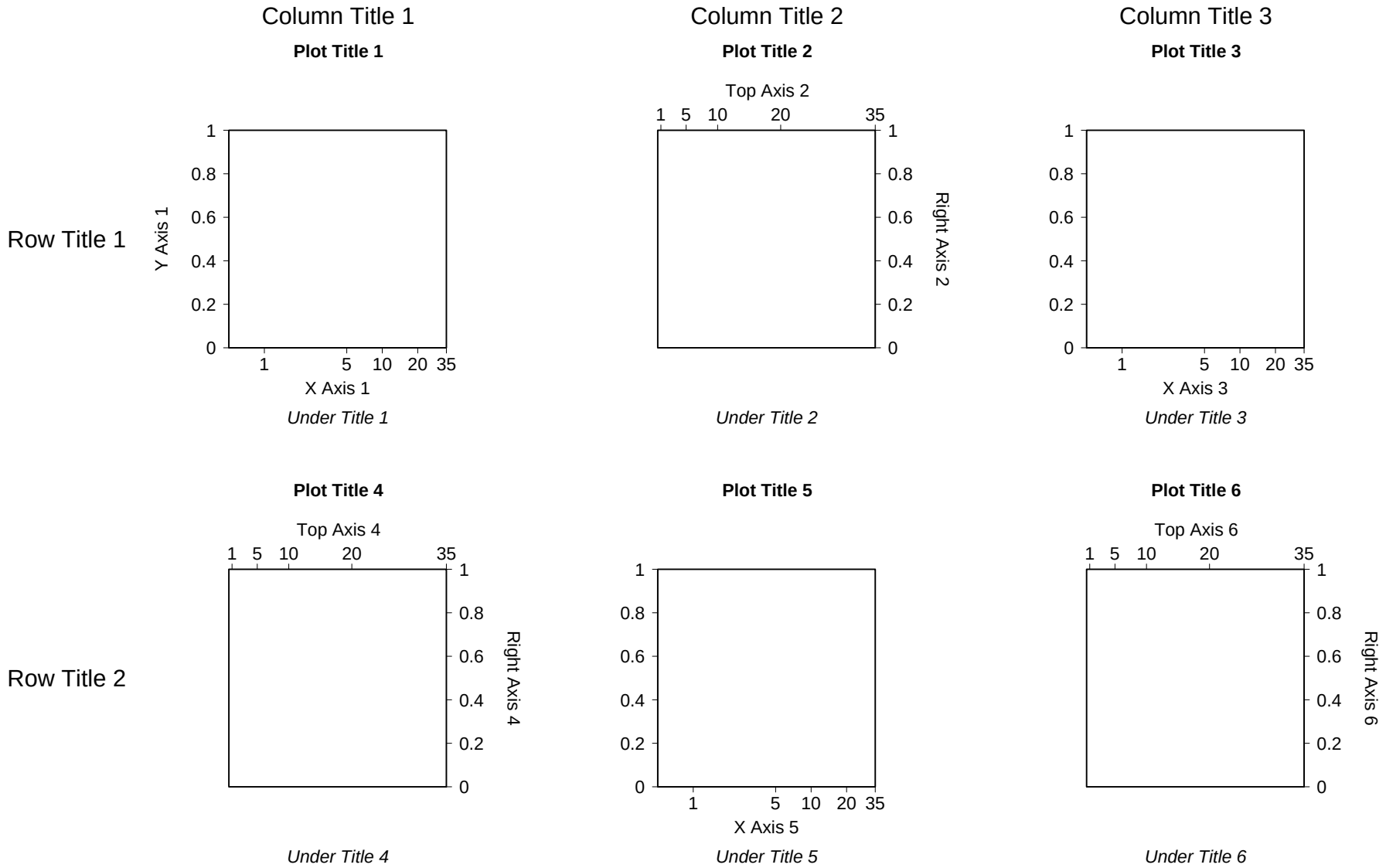
Some basic tile graphic types:

scatter	Scatterplots with fits, CIs, and extrapolation checking
lineplot	Line plots with fits, CIs, and extrapolation checking
ropeladder	Dot plots with CIs and extrapolation checking

Each can take as input draws from the posterior of a regression model

A call to a tile function makes a multiplot layout:

ideal for small multiples of model parameters


Column Title 1
Column Title 2
Column Title 3
Row Title 1
Row Title 2
Plot Title 1
Plot Title 2
Plot Title 3
Plot Title 4
Plot Title 5
Plot Title 6
Top Axis 2
Top Axis 4
Top Axis 6
1 5 10 20 35
Y Axis 1
X Axis 1
X Axis 3
X Axis 5
Right Axis 2
Right Axis 4
Right Axis 6
0 0.2 0.4 0.6 0.8 1
Under Title 1
Under Title 2
Under Title 3
Under Title 4
Under Title 5
Under Title 6

Plot simulations of QoI

Generally, we want to plot triples: lower, estimate, upper
We could do this for specific **discrete scenarios**, e.g.

Pr(Voting) given five distinct sets of *x*'s

Recommended plot: **Dotplot** with confidence interval lines

Plot simulations of QoI

Generally, we want to plot triples: lower, estimate, upper
We could do this for specific **discrete scenarios**, e.g.

Pr(Voting) given five distinct sets of *x*'s

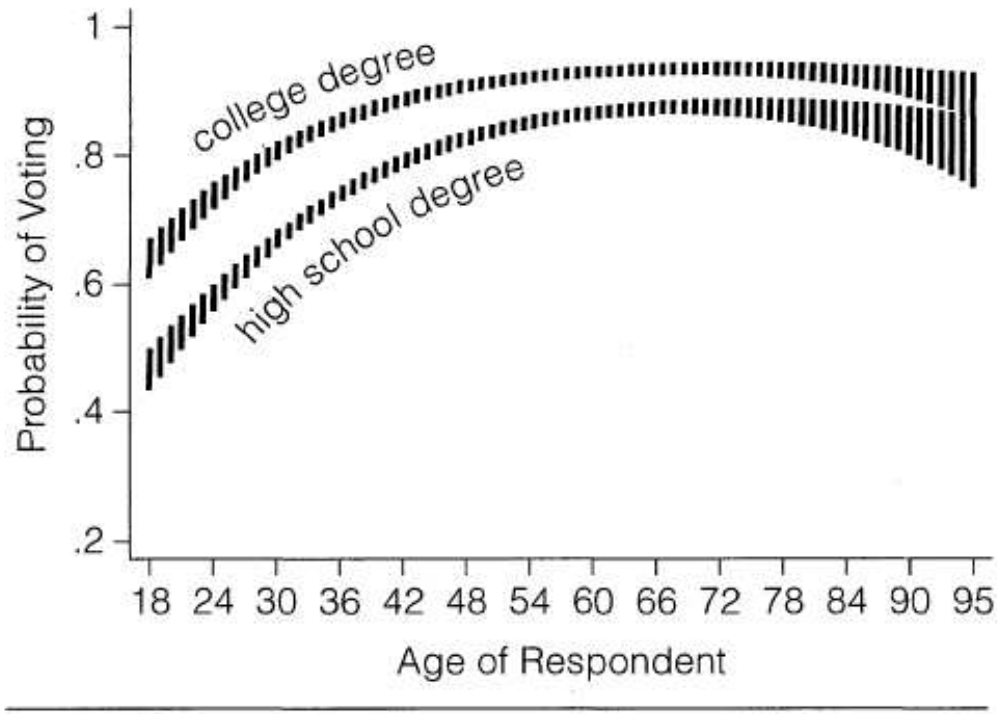
Recommended plot: **Dotplot** with confidence interval lines

Or for a continuous **stream of scenarios**, e.g.,

Hold all but Age constant, then calculate Pr(Voting) at every level of Age

Recommended plot: **Lineplot** with shaded confidence intervals

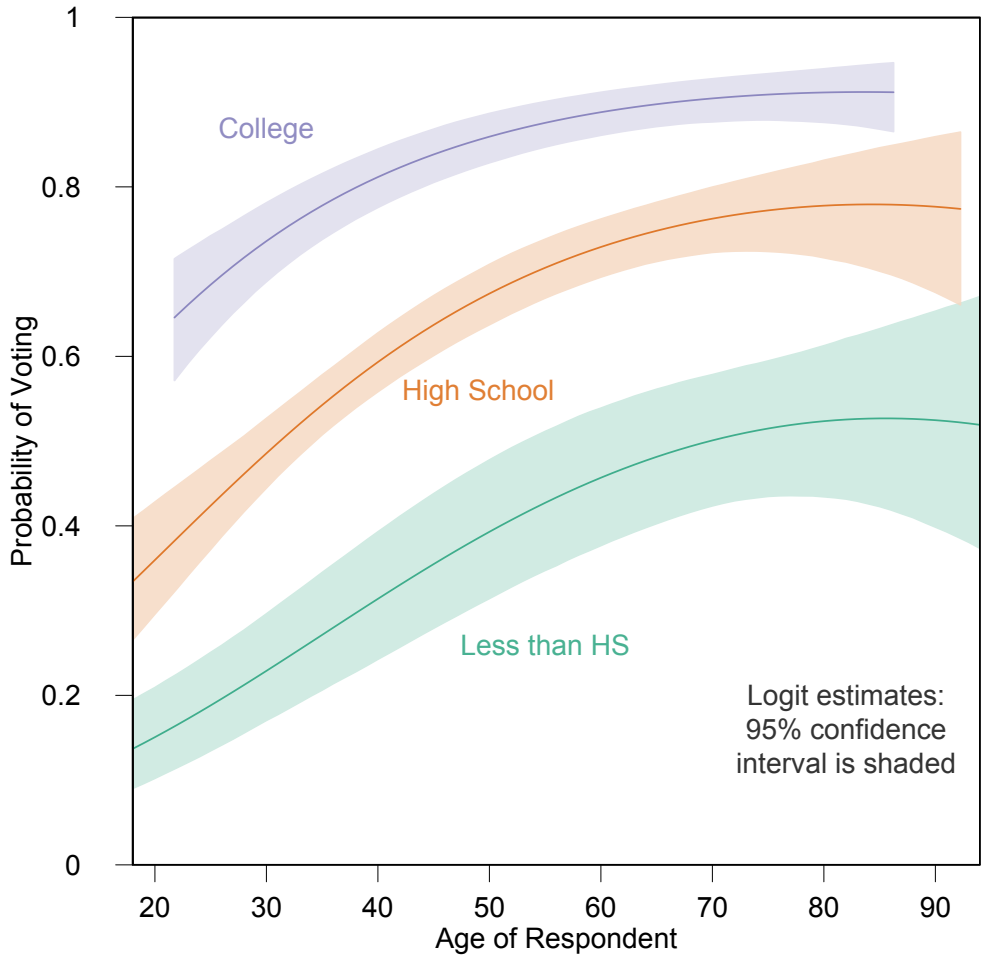
FIGURE 1 Probability of Voting by Age



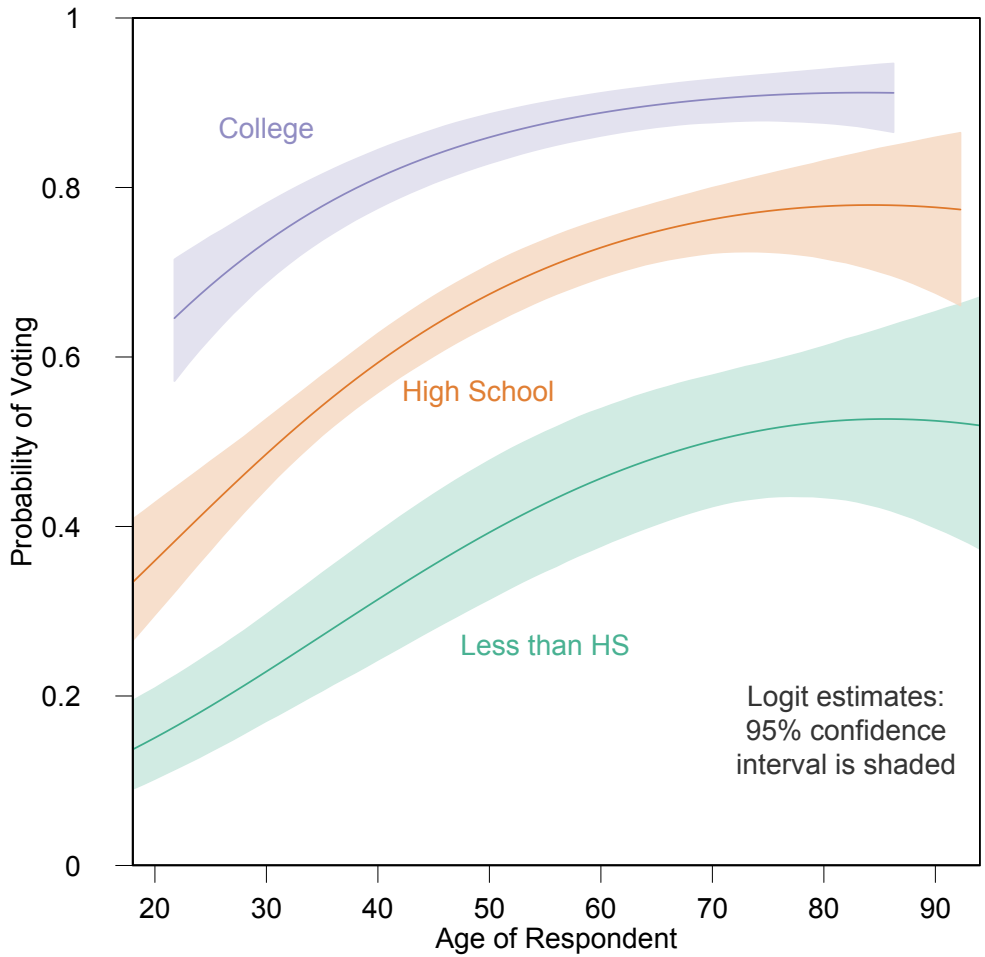
Vertical bars indicate 99-percent confidence intervals

This example is obviously superior to the table of logit coefficients

But is there anything wrong or missing here?



18 year old college grads?! And what about high school dropouts?



`tile` helps us systematize plotting model results,
and helps avoid unwanted extrapolation by limiting results to the convex hull

Three steps to make tile plots

1. **Create data traces.** Each trace contains the data and graphical parameters needed to plot a single set of graphical elements to one or more plots.

Three steps to make tile plots

1. **Create data traces.** Each trace contains the data and graphical parameters needed to plot a single set of graphical elements to one or more plots.
   - Could be a set of points, *or* text labels, *or* lines, *or* a polygon

Three steps to make tile plots

1. **Create data traces.** Each trace contains the data and graphical parameters needed to plot a single set of graphical elements to one or more plots.
   - Could be a set of points, *or* text labels, *or* lines, *or* a polygon
   - Could be a set of points *and* symbols, colors, labels, fit line, CIs, and/or extrapolation limits

Three steps to make tile plots

1. **Create data traces.** Each trace contains the data and graphical parameters needed to plot a single set of graphical elements to one or more plots.
   - Could be a set of points, *or* text labels, *or* lines, *or* a polygon
   - Could be a set of points *and* symbols, colors, labels, fit line, CIs, and/or extrapolation limits
   - Could be the data for a dotchart, with labels for each line

Three steps to make tile plots

1. **Create data traces.** Each trace contains the data and graphical parameters needed to plot a single set of graphical elements to one or more plots.
   - Could be a set of points, *or* text labels, *or* lines, *or* a polygon
   - Could be a set of points *and* symbols, colors, labels, fit line, CIs, and/or extrapolation limits
   - Could be the data for a dotchart, with labels for each line
   - Could be the marginal data for a rug
   - **All** annotation must happen in this step

Three steps to make tile plots

1. **Create data traces.** Each trace contains the data and graphical parameters needed to plot a single set of graphical elements to one or more plots.
   - Could be a set of points, *or* text labels, *or* lines, *or* a polygon
   - Could be a set of points *and* symbols, colors, labels, fit line, CIs, and/or extrapolation limits
   - Could be the data for a dotchart, with labels for each line
   - Could be the marginal data for a rug
   - **All** annotation must happen in this step
   - Basic traces: `linesTile()`, `pointsile()`, `polygonTile()`, `polylinesTile()`, and `textTile()`
   - Complex traces: `lineplot()`, `scatter()`, `ropeladder()`, and `rugTile()`

Trace functions in tile

Primitive trace functions:

`linesTile`	Plot a set of connected line segments
`pointsTile`	Plot a set of points
`polygonTile`	Plot a shaded region
`polylinesTile`	Plot a set of unconnected line segments
`textTile`	Plot text labels

Complex traces for model or data exploration:

`lineplot`	Plot lines with confidence intervals, extrapolation warnings
`ropeladder`	Plot dotplots with confidence intervals, extrapolation warnings, and shaded ranges
`rugTile`	Plot marginal data rugs to axes of plots
`scatter`	Plot scatterplots with text and symbol markers, fit lines, and confidence intervals

Three steps to make tile plots

1. **Create data traces.** Each trace contains the data and graphical parameters needed to plot a single set of graphical elements to one or more plots.

2. **Plot the data traces.** Using the `tile()` function, simultaneously plot all traces to all plots.

Three steps to make tile plots

1. **Create data traces.** Each trace contains the data and graphical parameters needed to plot a single set of graphical elements to one or more plots.

2. **Plot the data traces.** Using the `tile()` function, simultaneously plot all traces to all plots.
   - This is the step where the scaffolding gets made: axes and titles
   - Set up the rows and columns of plots
   - Titles of plots, axes, rows of plots, columns of plots, etc.
   - Set up axis limits, ticks, tick labels, logging of axes

Three steps to make tile plots

1. **Create data traces.** Each trace contains the data and graphical parameters needed to plot a single set of graphical elements to one or more plots.

2. **Plot the data traces.** Using the `tile()` function, simultaneously plot all traces to all plots.

3. **Examine output and revise.** Look at the graph made in step 2, and tweak the input parameters for steps 1 and 2 to make a better graph.

R Syntax for Lineplot of Voting Logit

```
# Using simcf and tile to explore an estimated logistic regression
# Voting example using 2000 NES data after King, Tomz, and Wittenberg
# Chris Adolph

# Load libraries
library(RColorBrewer)
library(MASS)
library(simcf)          # Download simcf and tile packages
library(tile)           # from faculty.washington.edu/cadolph

# Load data (available from faculty.washington.edu/cadolph)
file <- "nes00.csv"
data <- read.csv(file, header=TRUE)
```

R Syntax for Lineplot of Voting Logit

```
# Set up model formula and model specific data frame
model <- vote00 ~ age + I(age^2) + hsdeg + coldeg
mdata <- extractdata(model, data, na.rm=TRUE)

# Run logit & extract results
logit.result <- glm(model, family=binomial, data=mdata)
pe <- logit.result$coefficients  # point estimates
vc <- vcov(logit.result)         # var-cov matrix

# Simulate parameter distributions
sims <- 10000
simbetas <- mvrnorm(sims, pe, vc)
```

R Syntax for Lineplot of Voting Logit

```
# Set up counterfactuals:  all ages, each of three educations
xhyp <- seq(18,97,1)
nscen <- length(xhyp)
nohsScen <- hsScen <- collScen <- cfMake(model, mdata, nscen)
for (i in 1:nscen) {
  # No High school scenarios (loop over each age)
  nohsScen <- cfChange(nohsScen, "age", x = xhyp[i], scen = i)
  nohsScen <- cfChange(nohsScen, "hsdeg", x = 0, scen = i)
  nohsScen <- cfChange(nohsScen, "coldeg", x = 0, scen = i)

  # HS grad scenarios (loop over each age)
  hsScen <- cfChange(hsScen, "age", x = xhyp[i], scen = i)
  hsScen <- cfChange(hsScen, "hsdeg", x = 1, scen = i)
  hsScen <- cfChange(hsScen, "coldeg", x = 0, scen = i)

  # College grad scenarios (loop over each age)
  collScen <- cfChange(collScen, "age", x = xhyp[i], scen = i)
  collScen <- cfChange(collScen, "hsdeg", x = 1, scen = i)
  collScen <- cfChange(collScen, "coldeg", x = 1, scen = i)
}
```

R Syntax for Lineplot of Voting Logit

```
# Simulate expected probabilities for all scenarios
nohsSims <- logitsimev(nohsScen, simbetas, ci=0.95)
hsSims <- logitsimev(hsScen, simbetas, ci=0.95)
collSims <- logitsimev(collScen, simbetas, ci=0.95)

# Get 3 nice colors for traces
col <- brewer.pal(3,"Dark2")

# Set up lineplot traces of expected probabilities
nohsTrace <- lineplot(x=xhyp,
                      y=nohsSims$pe,
                      lower=nohsSims$lower,
                      upper=nohsSims$upper,
                      col=col[1],
                      extrapolate=list(data=mdata[,2:ncol(mdata)],
                                       cfact=nohsScen$x[,2:ncol(hsScen$x)],
                                           omit.extrapolated=TRUE),
                      plot=1)
```

R Syntax for Lineplot of Voting Logit

```
hsTrace <- lineplot(x=xhyp,
                    y=hsSims$pe,
                    lower=hsSims$lower,
                    upper=hsSims$upper,
                    col=col[2],
                    extrapolate=list(data=mdata[,2:ncol(mdata)],
                                     cfact=hsScen$x[,2:ncol(hsScen$x)],
                                         omit.extrapolated=TRUE),
                    plot=1)

collTrace <- lineplot(x=xhyp,
                      y=collSims$pe,
                      lower=collSims$lower,
                      upper=collSims$upper,
                      col=col[3],
                      extrapolate=list(data=mdata[,2:ncol(mdata)],
                                       cfact=collScen$x[,2:ncol(hsScen$x)],
                                           omit.extrapolated=TRUE),
                      plot=1)
```

R Syntax for Lineplot of Voting Logit

```
# Set up traces with labels and legend
labelTrace <- textTile(labels=c("Less than HS", "High School",
                                "College"),
                       x=c( 55,    49,     30),
                       y=c( 0.26,  0.56,   0.87),
                       col=col,
                       plot=1)

legendTrace <- textTile(labels=c("Logit estimates:", "95% confidence",
                                 "interval is shaded"),
                        x=c(82, 82, 82),
                        y=c(0.2, 0.16, 0.12),
                        plot=1)
```

R Syntax for Lineplot of Voting Logit

```
# Plot traces using tile
tile(nohsTrace,
     hsTrace,
     collTrace,
     labelTrace,
     legendTrace,
     width=list(null=5),
     limits=c(18,94,0,1),
     xaxis=list(at=c(20,30,40,50,60,70,80,90)),
     xaxistitle=list(labels="Age of Respondent"),
     yaxistitle=list(labels="Probability of Voting"),
     frame=TRUE
     )
```

Chomp! Coefficient tables can hide the punchline

Agresti offers the following example of a multinominal data analysis

Alligators in a certain Florida lake were studied, and the following data collected:

Principal Food	1 = Invertebrates, 2 = Fish, 3= “Other” (!!! Floridians?)
Size of alligator	in meters
Sex of alligator	male or female

The question is how alligator size influences food choice.

We fit the model in `R` using multinomial logit and get . . .

Chomp! Coefficient tables can hide the punchline

Multinomial Logit of Alligator's Primary Food Source

	$\ln(\pi_{\text{Invertebrates}}/\pi_{\text{Fish}})$	$\ln(\pi_{\text{Other}}/\pi_{\text{Fish}})$
Intercept	4.90	−1.95
	(1.71)	(1.53)
Size	−2.53	0.13
	(0.85)	(0.52)
Female	−0.79	0.38
	(0.71)	(0.91)

Direct interpretation?

Chomp! Coefficient tables can hide the punchline

Multinomial Logit of Alligator's Primary Food Source

	$\ln(\pi_{\text{Invertebrates}}/\pi_{\text{Fish}})$	$\ln(\pi_{\text{Other}}/\pi_{\text{Fish}})$
Intercept	4.90	−1.95
	(1.71)	(1.53)
Size	−2.53	0.13
	(0.85)	(0.52)
Female	−0.79	0.38
	(0.71)	(0.91)

Direct interpretation?

We could do it, using odds ratios and a calculator.

Invertebrates vs Fish: A 1 meter increase in length makes the *odds* that an alligator will eat invertebrates rather than fish $\exp(-2.53 - 0) = 0.08$ times smaller.

Chomp! Coefficient tables can hide the punchline

Multinomial Logit of Alligator's Primary Food Source

	$\ln(\pi_{\text{Invertebrates}}/\pi_{\text{Fish}})$	$\ln(\pi_{\text{Other}}/\pi_{\text{Fish}})$
Intercept	4.90	−1.95
	(1.71)	(1.53)
Size	−2.53	0.13
	(0.85)	(0.52)
Female	−0.79	0.38
	(0.71)	(0.91)

Invertebrates vs Other: A 1 meter increase in length makes the *odds* that an alligator will eat invertebrates rather than "other" food
$\exp(-2.53 - 0.13) = 0.07$ times smaller.

Chomp! Coefficient tables can hide the punchline

Multinomial Logit of Alligator's Primary Food Source

	$\ln(\pi_{\text{Invertebrates}}/\pi_{\text{Fish}})$	$\ln(\pi_{\text{Other}}/\pi_{\text{Fish}})$
Intercept	4.90	−1.95
	(1.71)	(1.53)
Size	−2.53	0.13
	(0.85)	(0.52)
Female	−0.79	0.38
	(0.71)	(0.91)

Invertebrates vs Other: A 1 meter increase in length makes the *odds* that an alligator will eat invertebrates rather than "other" food
$\exp(-2.53 - 0.13) = 0.07$ times smaller.

No reader is going to do this

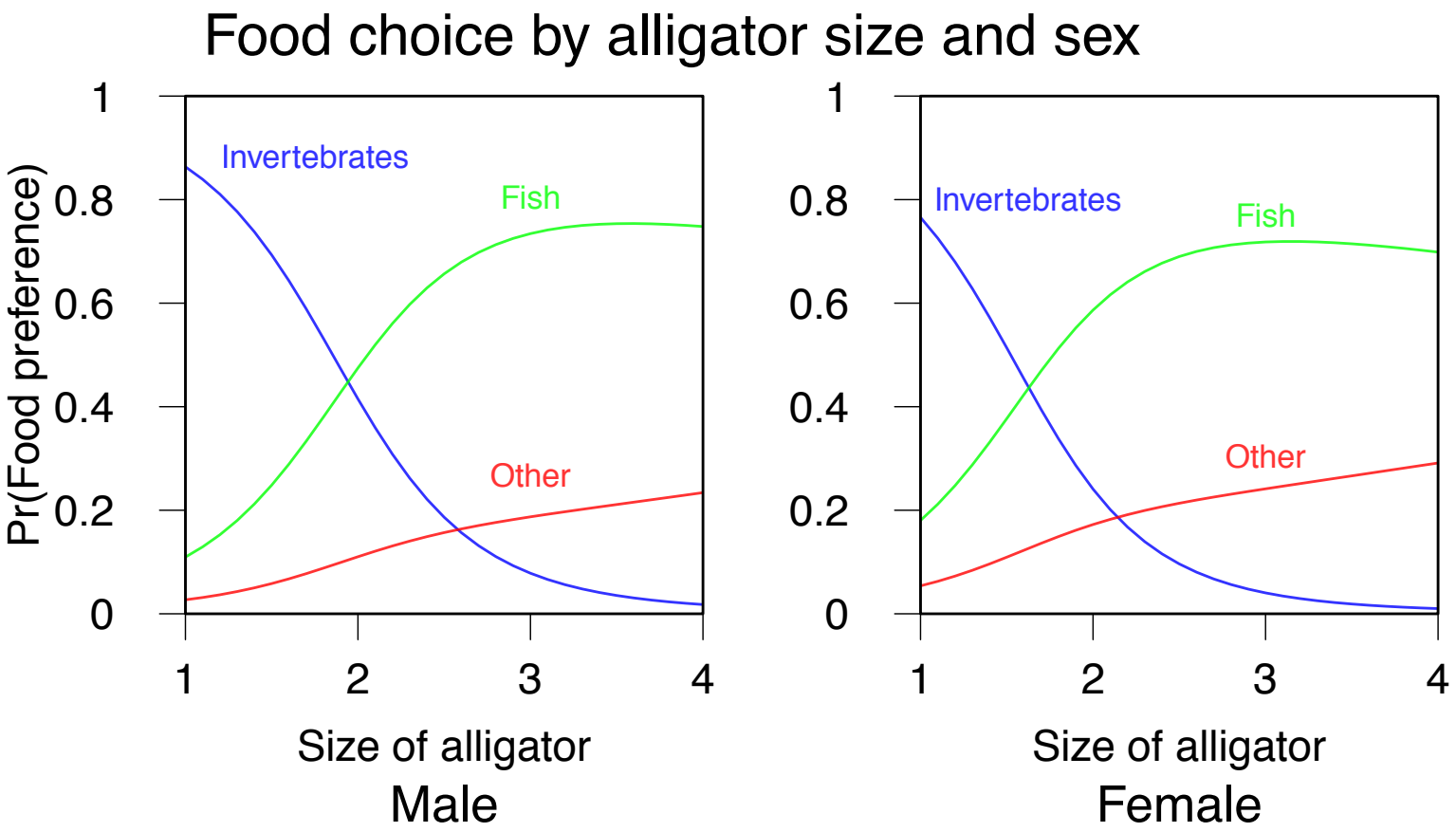
Chomp! Coefficient tables can hide the punchline

Multinomial Logit of Alligator's Primary Food Source

	$\ln(\pi_{\text{Invertebrates}}/\pi_{\text{Fish}})$	$\ln(\pi_{\text{Other}}/\pi_{\text{Fish}})$
Intercept	4.90	−1.95
	(1.71)	(1.53)
Size	−2.53	0.13
	(0.85)	(0.52)
Female	−0.79	0.38
	(0.71)	(0.91)

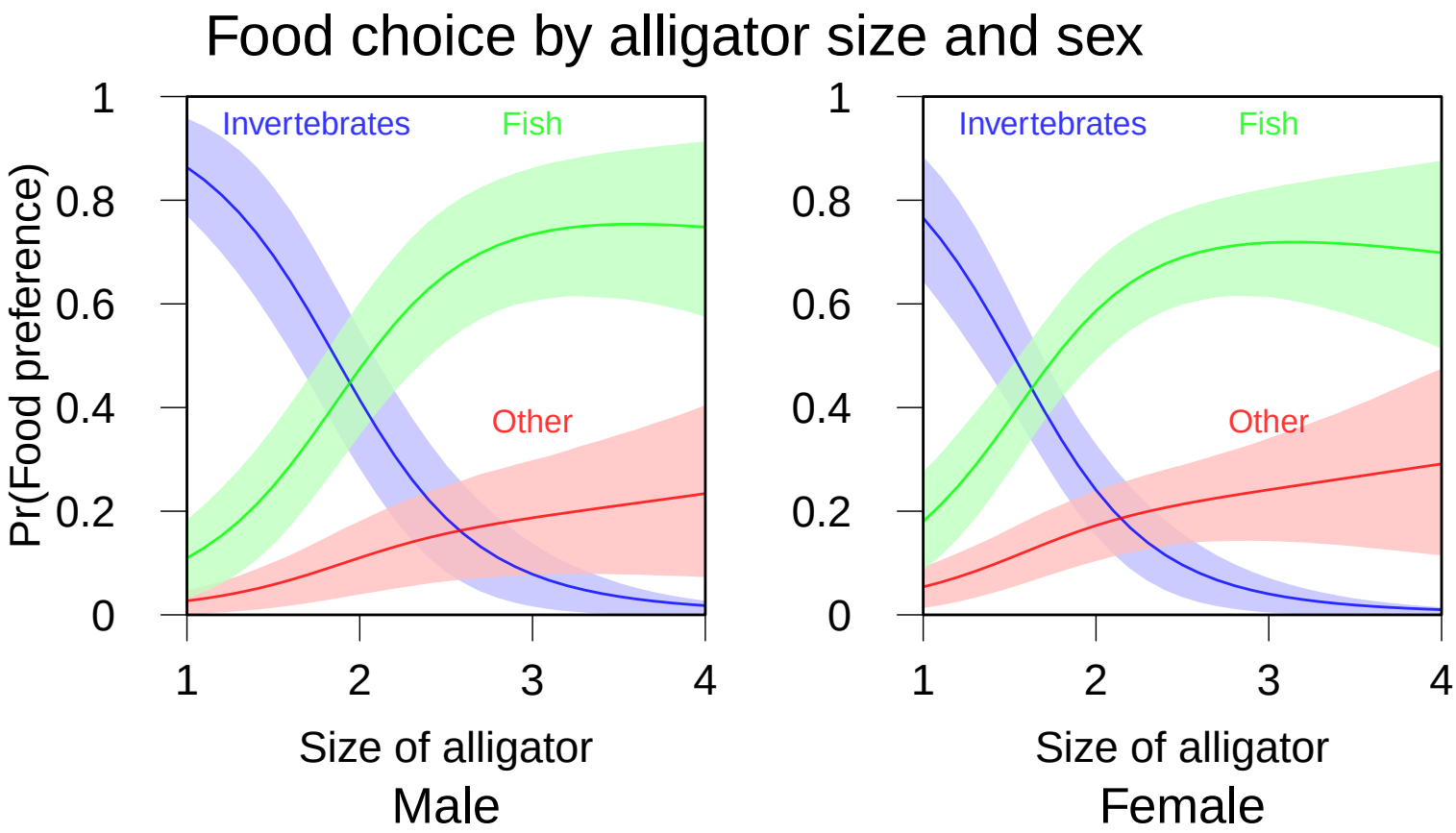
Fish vs Other: A 1 meter increase in length makes the *odds* that an alligator will eat fish rather than "other" food $\exp(0 - 0.13) = 0.87$ times smaller.

There has to be a better way



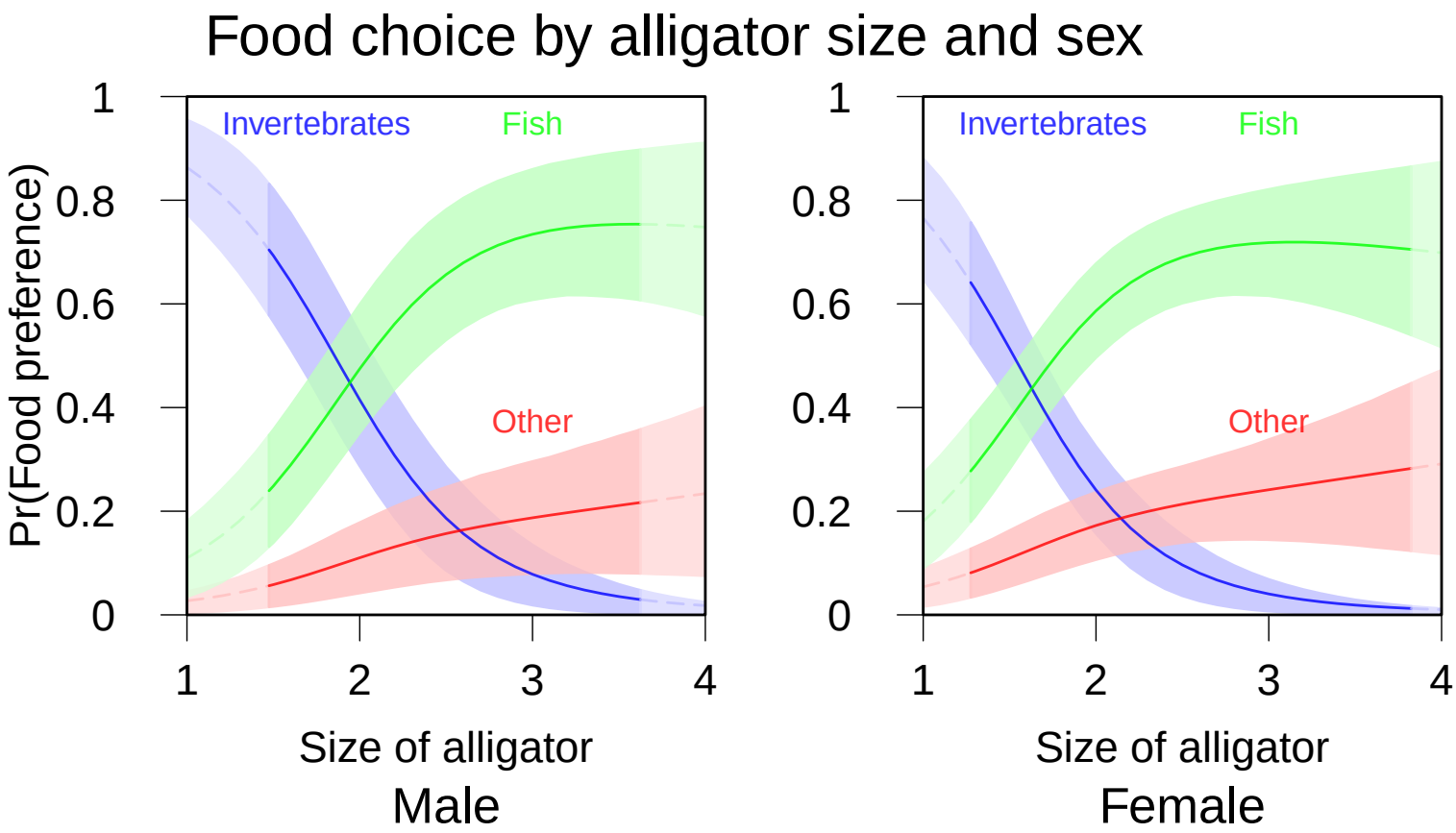
A simple lineplot made with `tile` shows the whole covariate space

Much more dramatic than the table



We can also add easy to interpret measures of uncertainty

Above are standard error regions



Easy to highlight where the model is interpolating into the observed data

. . . and where it's extrapolating into Hollywood territory

When simulation is the only option: Chinese leadership

Shih, Adolph, and Liu investigate the advancement of elite Chinese leaders in the Reform Period (1982–2002)

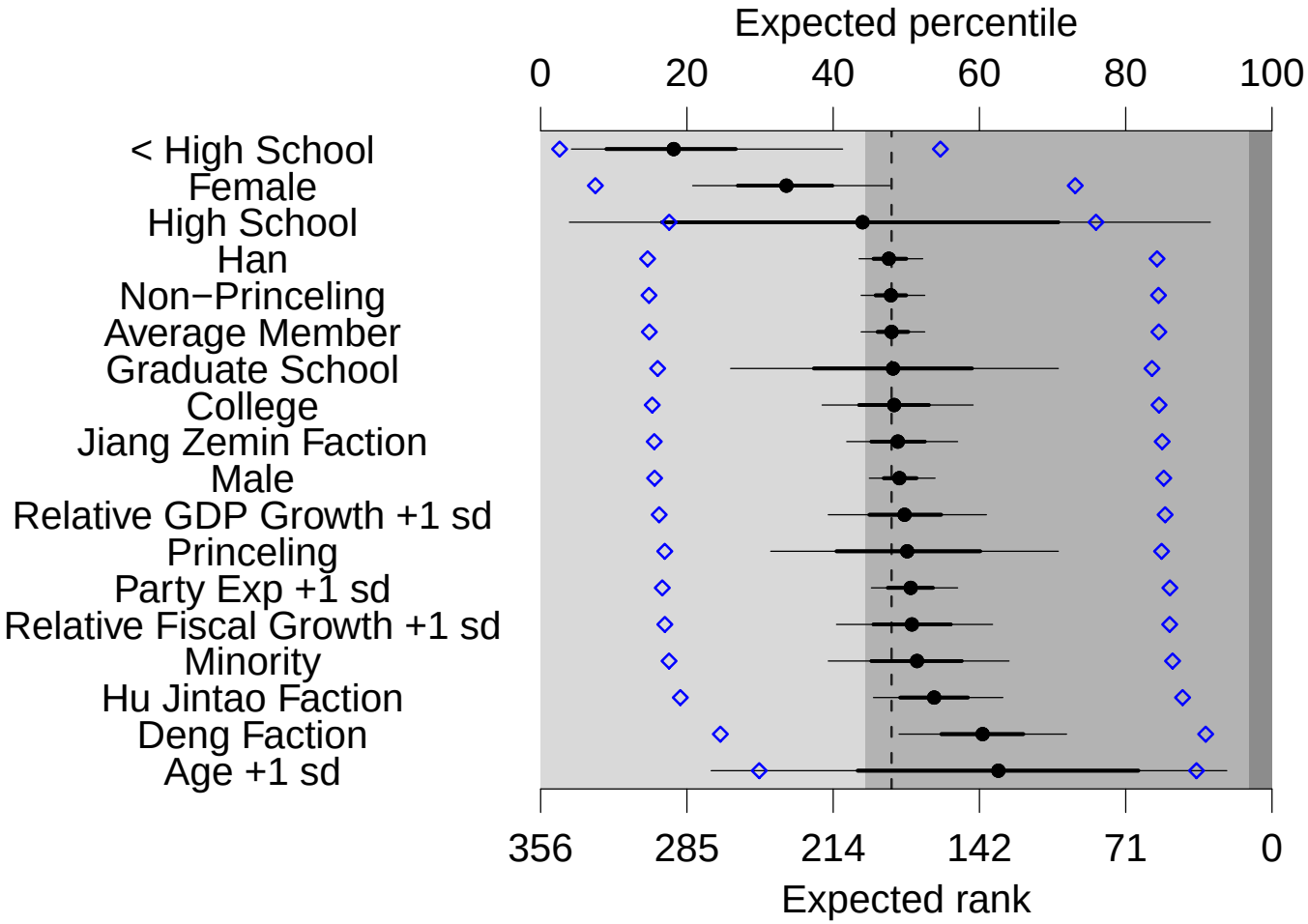
Explain (partially observed) ranks of the top 300 to 500 Chinese Communist Party leaders as a function of:

Demographics	*age, sex, ethnicity*
Education	*level of degree*
Performance	*provincial growth, revenue*
Faction	*birth, school, career, and family ties to top leaders*

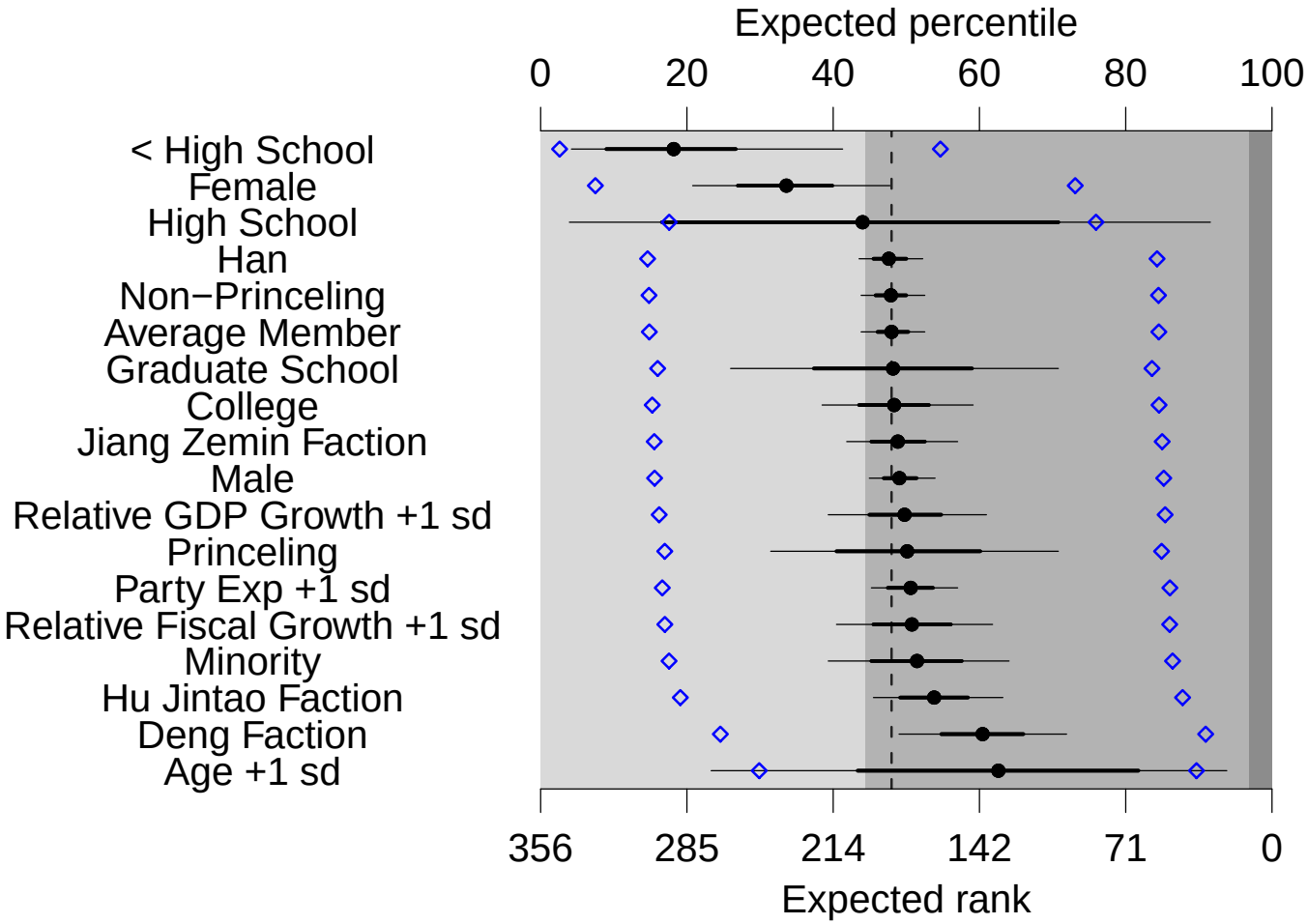
Bayesian model of partially observed ranks of CCP officials

Model parameters difficult to interpret: on a latent scale
and individual effects are conditioned on all other ranked members

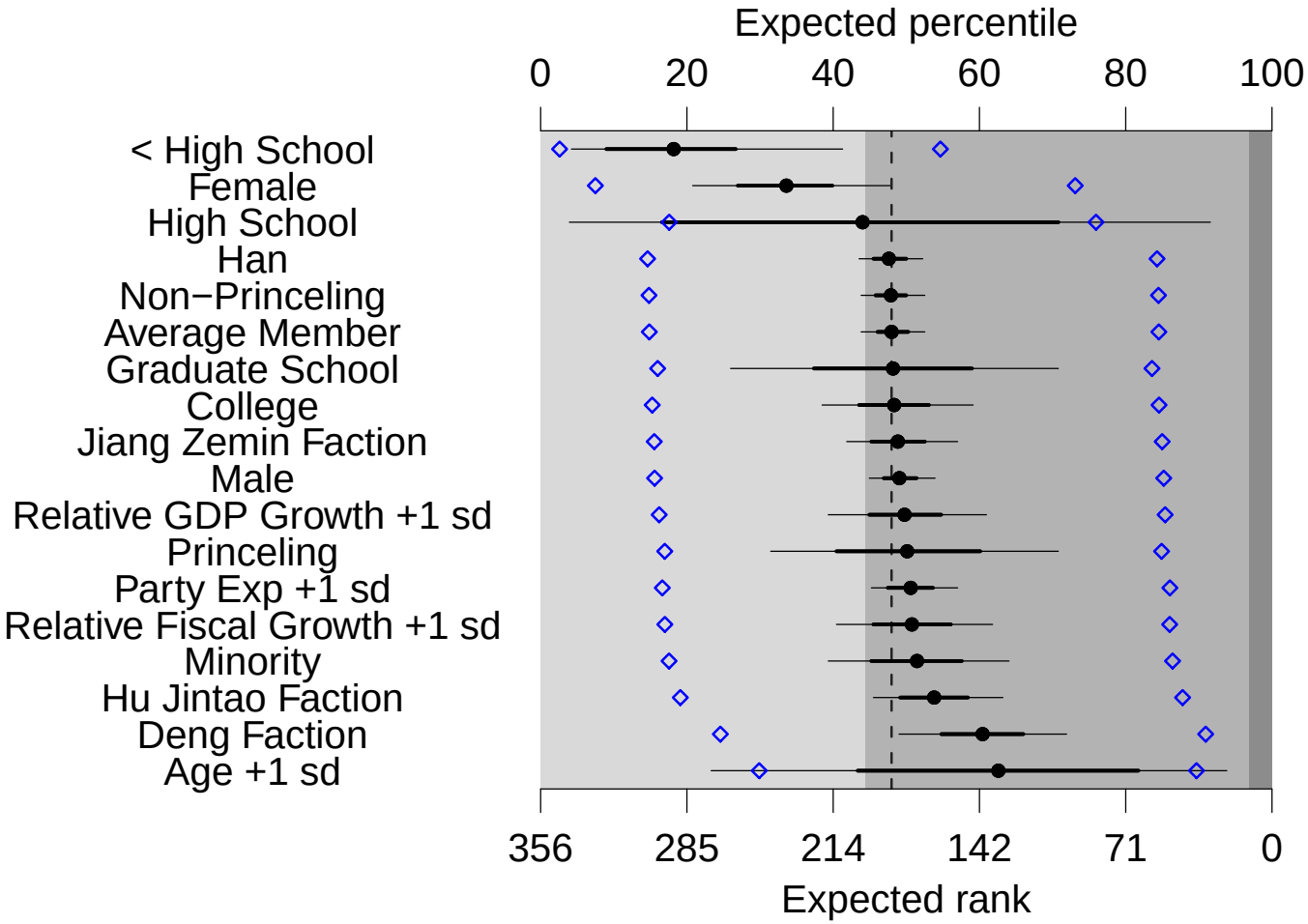
Only solution:
Simulate ranks of hypothetical officials as if placed in the observed hierarchy



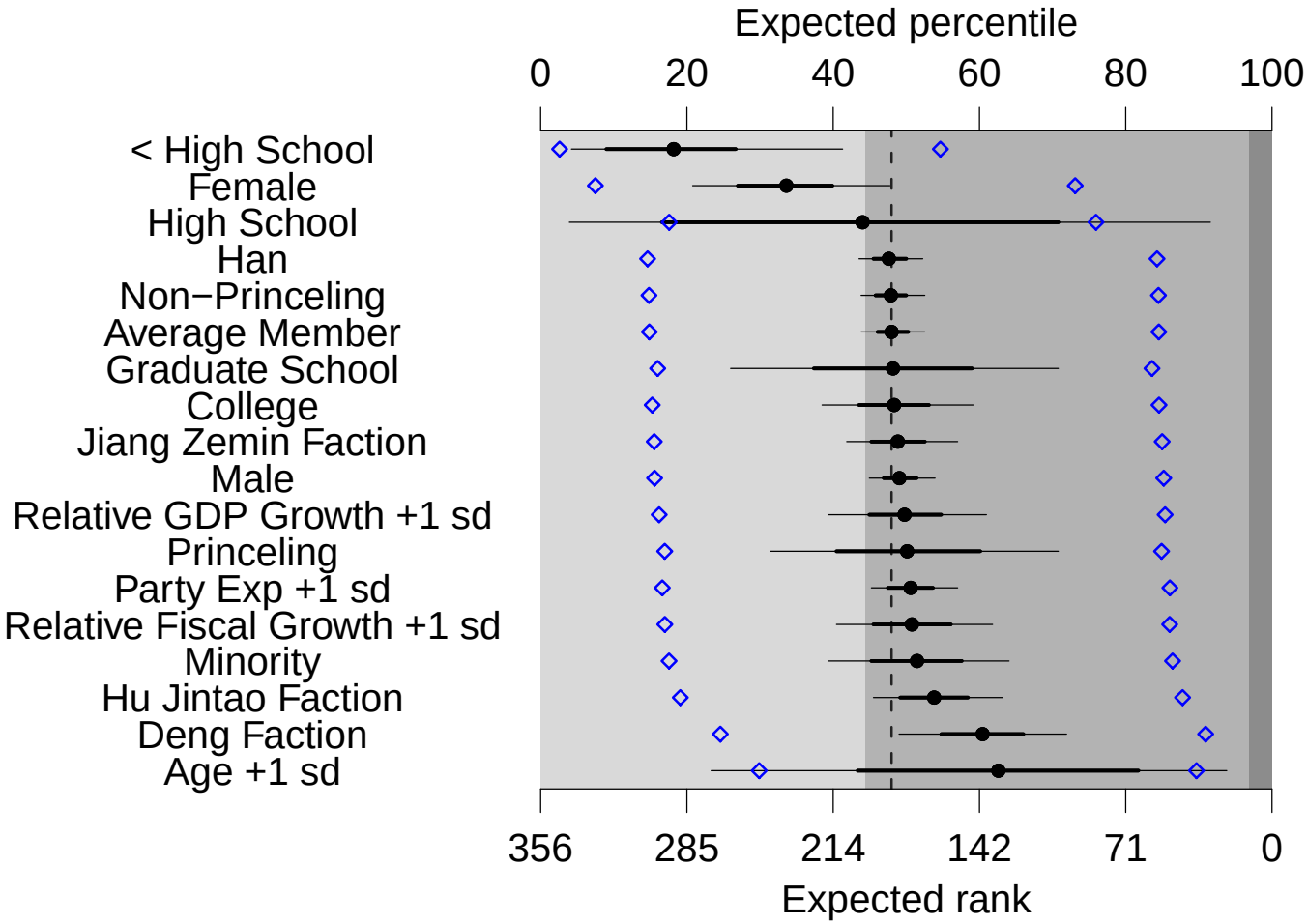
Black circles show expected ranks for otherwise average Chinese officials with the characteristic listed at left



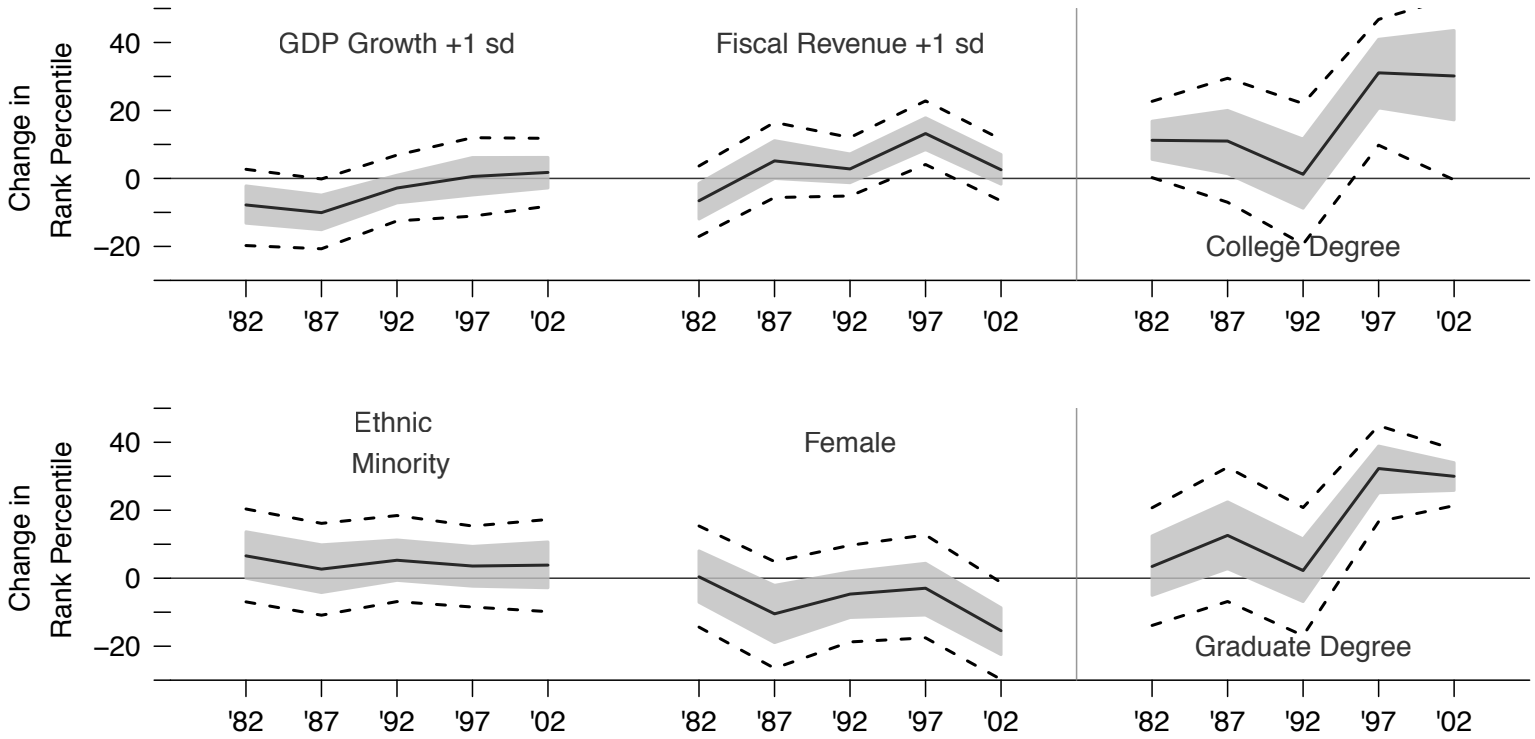
Thick black horizontal lines are 1 std error bars, and thin lines are 95% CIs



Blue triangles are officials with random effects at ± 1 sd; how much unmeasured factors matter

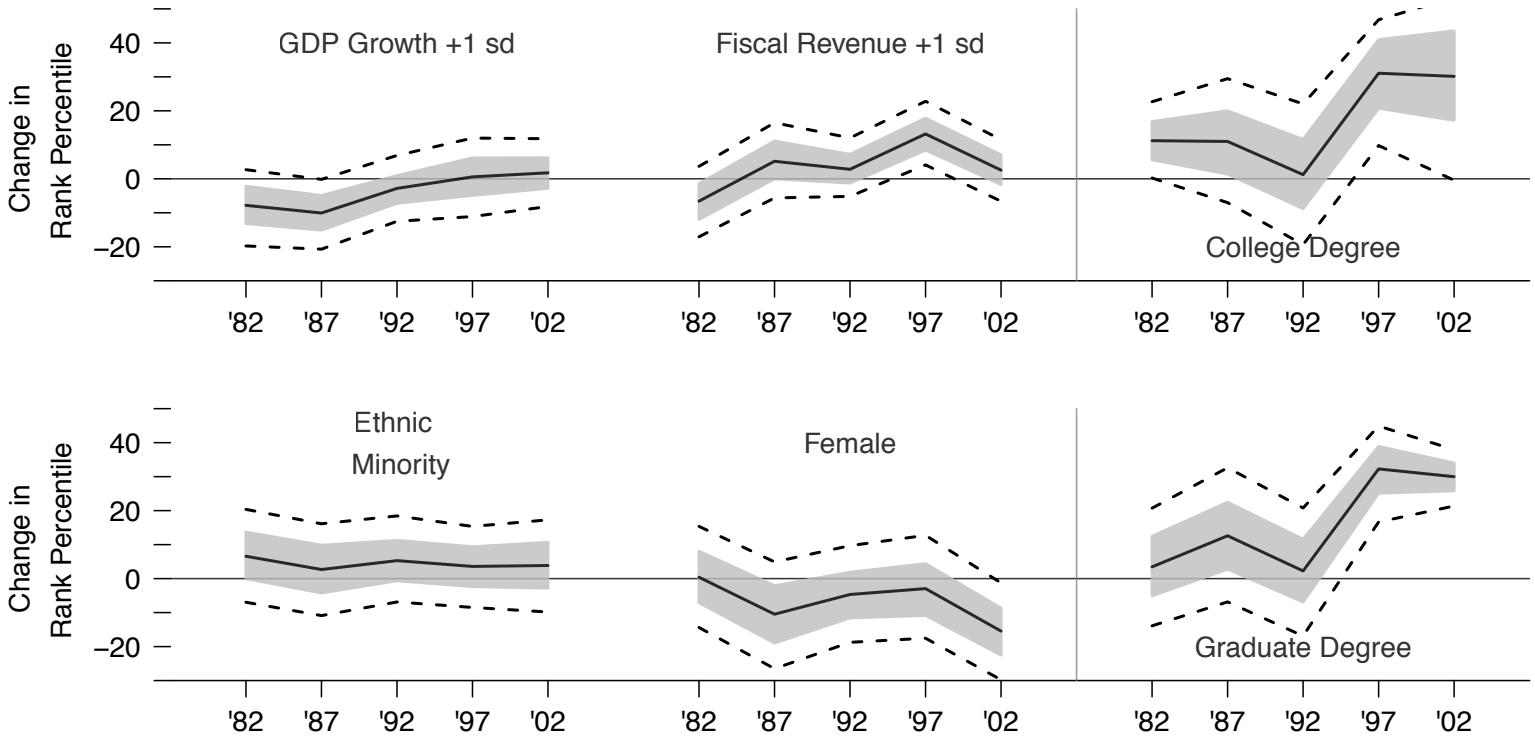


Helps to sort rows of the plot from smallest to largest effect



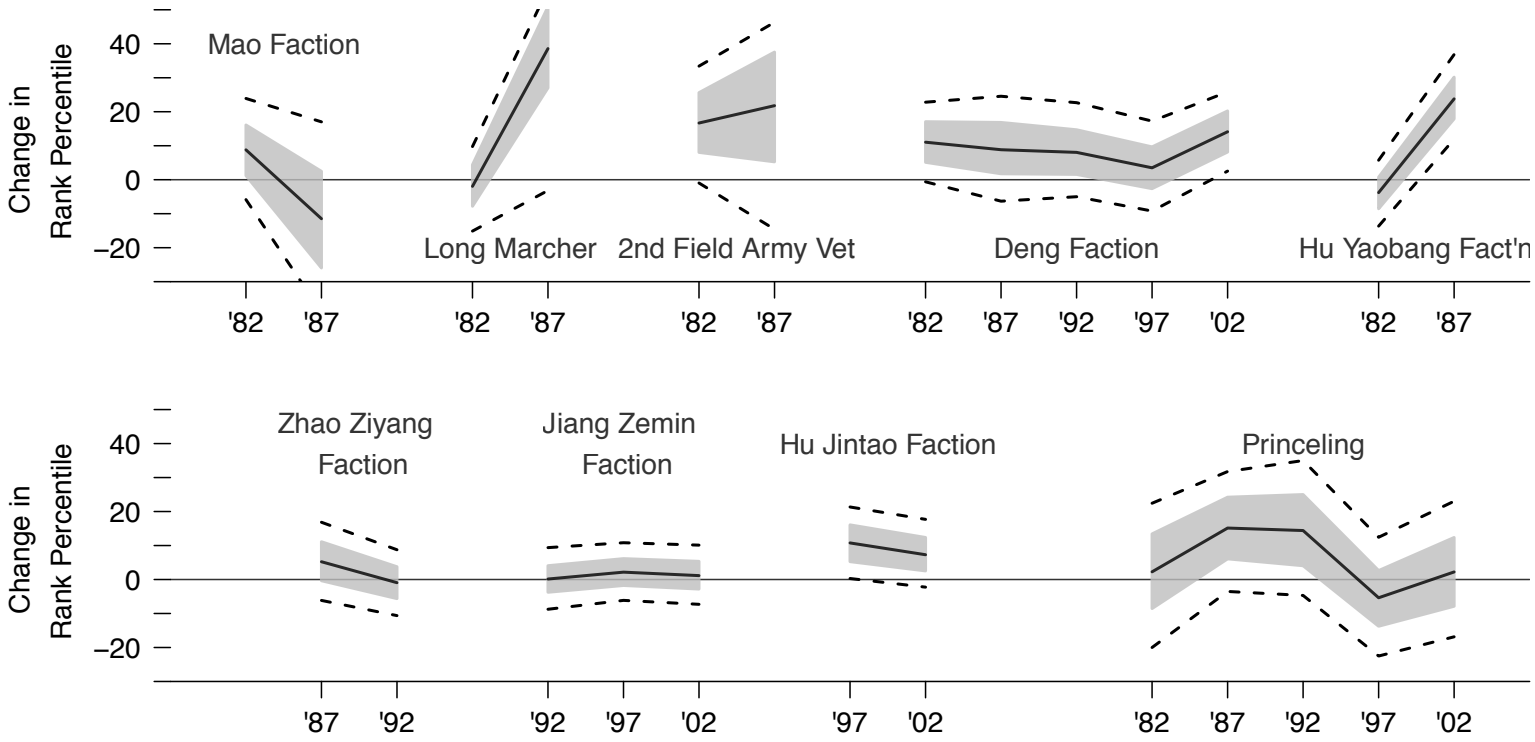
Shih, Adolph, and Liu re-estimate the model for each year, leading to a large number of results

A complex lineplot helps organize them and facilitate comparisons



Note that these results are now *first differences*:

the expected percentile change in rank for an otherwise average official who gains the characteristic noted



Over time, officials' economic performance never matters, but factions often do

Runs counter to the conventional wisdom that meritocratic selection of officials lies behind Chinese economic success

Robustness Checks

So far, we've presenting conditional expectations & differences from regressions

But are we confident that these were the "right" estimates?

The language of inference usually assumes we

- correctly specified our model
- correctly measured our variables
- chose the right probability model
- don't have influential outliers, etc.

Robustness Checks

So far, we've presenting conditional expectations & differences from regressions

But are we confident that these were the "right" estimates?

The language of inference usually assumes we

- correctly specified our model
- correctly measured our variables
- chose the right probability model
- don't have influential outliers, etc.

We're never completely sure these assumptions hold.

Most people present one model, and argue it was the best choice

Sometimes, a few alternatives are displayed

The race of the variables

	Model 1	Model 2	Model 3	Model 4	Model 5
My variable of interest, X_1	X.XX (X.XX)	X.XX (X.XX)	X.XX (X.XX)	X.XX (X.XX)	
A control I "need"	X.XX (X.XX)	X.XX (X.XX)	X.XX (X.XX)	X.XX (X.XX)	X.XX (X.XX)
A control I "need"	X.XX (X.XX)	X.XX (X.XX)	X.XX (X.XX)	X.XX (X.XX)	X.XX (X.XX)
A candidate control		X.XX (X.XX)		X.XX (X.XX)	
A candidate control			X.XX (X.XX)	X.XX (X.XX)	
Alternate measure of X_1					X.XX (X.XX)

Robustness Checks

Problems with the approach above?

1. Lots of space to show a few permutations of the model

   Most space wasted or devoted to ancillary info

Robustness Checks

Problems with the approach above?

1. Lots of space to show a few permutations of the model

   Most space wasted or devoted to ancillary info

2. What if we're really interested in $\mathrm{E}(Y|X)$, not $\hat{\beta}$?

   E.g., because of nonlinearities, interactions, scale differences, etc.

Robustness Checks

Problems with the approach above?

1. Lots of space to show a few permutations of the model

   Most space wasted or devoted to ancillary info

2. What if we're really interested in $\mathrm{E}(Y|X)$, not $\hat{\beta}$?

   E.g., because of nonlinearities, interactions, scale differences, etc.

3. The selection of permutations is *ad hoc*.

Robustness Checks

Problems with the approach above?

1. Lots of space to show a few permutations of the model

   Most space wasted or devoted to ancillary info

2. What if we're really interested in $\mathrm{E}(Y|X)$, not $\hat{\beta}$?

   E.g., because of nonlinearities, interactions, scale differences, etc.

3. The selection of permutations is *ad hoc*.

We'll try to fix 1 & 2.

Objection 3 is harder, but worth thinking about.

Robustness Checks: An algorithm

1. Identify a relation of interest between a concept $\mathcal{X}$ and a concept $\mathcal{Y}$

Robustness Checks: An algorithm

1. Identify a relation of interest between a concept $\mathcal{X}$ and a concept $\mathcal{Y}$
2. Choose:
   - a measure of $\mathcal{X}$, denoted X,

Robustness Checks: An algorithm

1. Identify a relation of interest between a concept $\mathcal{X}$ and a concept $\mathcal{Y}$
2. Choose:
   - a measure of $\mathcal{X}$, denoted X,
   - a measure of $\mathcal{Y}$, denoted Y,

Robustness Checks: An algorithm

1. Identify a relation of interest between a concept $\mathcal{X}$ and a concept $\mathcal{Y}$
2. Choose:
   - a measure of $\mathcal{X}$, denoted X,
   - a measure of $\mathcal{Y}$, denoted Y,
   - a set of confounders, Z,

Robustness Checks: An algorithm

1. Identify a relation of interest between a concept $\mathcal{X}$ and a concept $\mathcal{Y}$
2. Choose:
   - a measure of $\mathcal{X}$, denoted X,
   - a measure of $\mathcal{Y}$, denoted Y,
   - a set of confounders, Z,
   - a functional form, $g(\cdot)$

Robustness Checks: An algorithm

1. Identify a relation of interest between a concept $\mathcal{X}$ and a concept $\mathcal{Y}$
2. Choose:
   - a measure of $\mathcal{X}$, denoted X,
   - a measure of $\mathcal{Y}$, denoted Y,
   - a set of confounders, Z,
   - a functional form, $g(\cdot)$
   - a probability model of Y, $f(\cdot)$
3. Estimate the probability model $Y \sim f(\mu, \alpha)$, $\mu = g(\text{vec}(X, Z), \beta)$.

Robustness Checks: An algorithm

1. Identify a relation of interest between a concept $\mathcal{X}$ and a concept $\mathcal{Y}$
2. Choose:
   - a measure of $\mathcal{X}$, denoted X,
   - a measure of $\mathcal{Y}$, denoted Y,
   - a set of confounders, Z,
   - a functional form, $g(\cdot)$
   - a probability model of Y, $f(\cdot)$
3. Estimate the probability model $Y \sim f(\mu, \alpha)$, $\mu = g(\text{vec}(X, Z), \beta)$.
4. Simulate the quantity of interest, e.g., $\text{E}(Y|X)$ or $\text{E}(Y_2 - Y_1|X_2, X_1)$, to obtain a point estimate and confidence interval.

Robustness Checks: An algorithm

1. Identify a relation of interest between a concept $\mathcal{X}$ and a concept $\mathcal{Y}$
2. Choose:
   - a measure of $\mathcal{X}$, denoted X,
   - a measure of $\mathcal{Y}$, denoted Y,
   - a set of confounders, Z,
   - a functional form, $g(\cdot)$
   - a probability model of Y, $f(\cdot)$
3. Estimate the probability model $Y \sim f(\mu, \alpha)$, $\mu = g(\text{vec}(X, Z), \beta)$.
4. Simulate the quantity of interest, e.g., $\text{E}(Y|X)$ or $\text{E}(Y_2 - Y_1|X_2, X_1)$, to obtain a point estimate and confidence interval.
5. Repeat 2–4, changing at each iteration one of the choices in step 2.

Robustness Checks: An algorithm

1. Identify a relation of interest between a concept $\mathcal{X}$ and a concept $\mathcal{Y}$
2. Choose:
   - a measure of $\mathcal{X}$, denoted X,
   - a measure of $\mathcal{Y}$, denoted Y,
   - a set of confounders, Z,
   - a functional form, $g(\cdot)$
   - a probability model of Y, $f(\cdot)$
3. Estimate the probability model $Y \sim f(\mu, \alpha)$, $\mu = g(\text{vec}(X, Z), \beta)$.
4. Simulate the quantity of interest, e.g., $\text{E}(Y|X)$ or $\text{E}(Y_2 - Y_1|X_2, X_1)$, to obtain a point estimate and confidence interval.
5. Repeat 2–4, changing at each iteration one of the choices in step 2.
6. Compile the results in a variant of the dot plot called a *ropeladder*.

Robustness Checks: A central banks example

In *The Dilemma of Discretion*, I argue central bankers' career backgrounds explain their monetary policy choices

Central bankers with financial sector backgrounds choose more conservative policies, leading to lower inflation but potentially higher unemployment

I argue more conservative governments should prefer to appoint more conservative central bankers, e.g., those with financial sector backgrounds

Robustness Checks: A central banks example

I test this claim using data on central bankers setting monetary policy over 20 countries and 30 years

Conservatism of central bankers is measured by percentage share in "conservative" careers, and summarized by an index CBCC

Partisanship of government is measured by PCoG:
higher values = more conservative Partisan"Center of Gravity"

I estimate a zero-inflated compositional regression on career shares

and find the expected relationship:
conservative goverments pick conservative central bankers

Robustness Checks: A central banks example

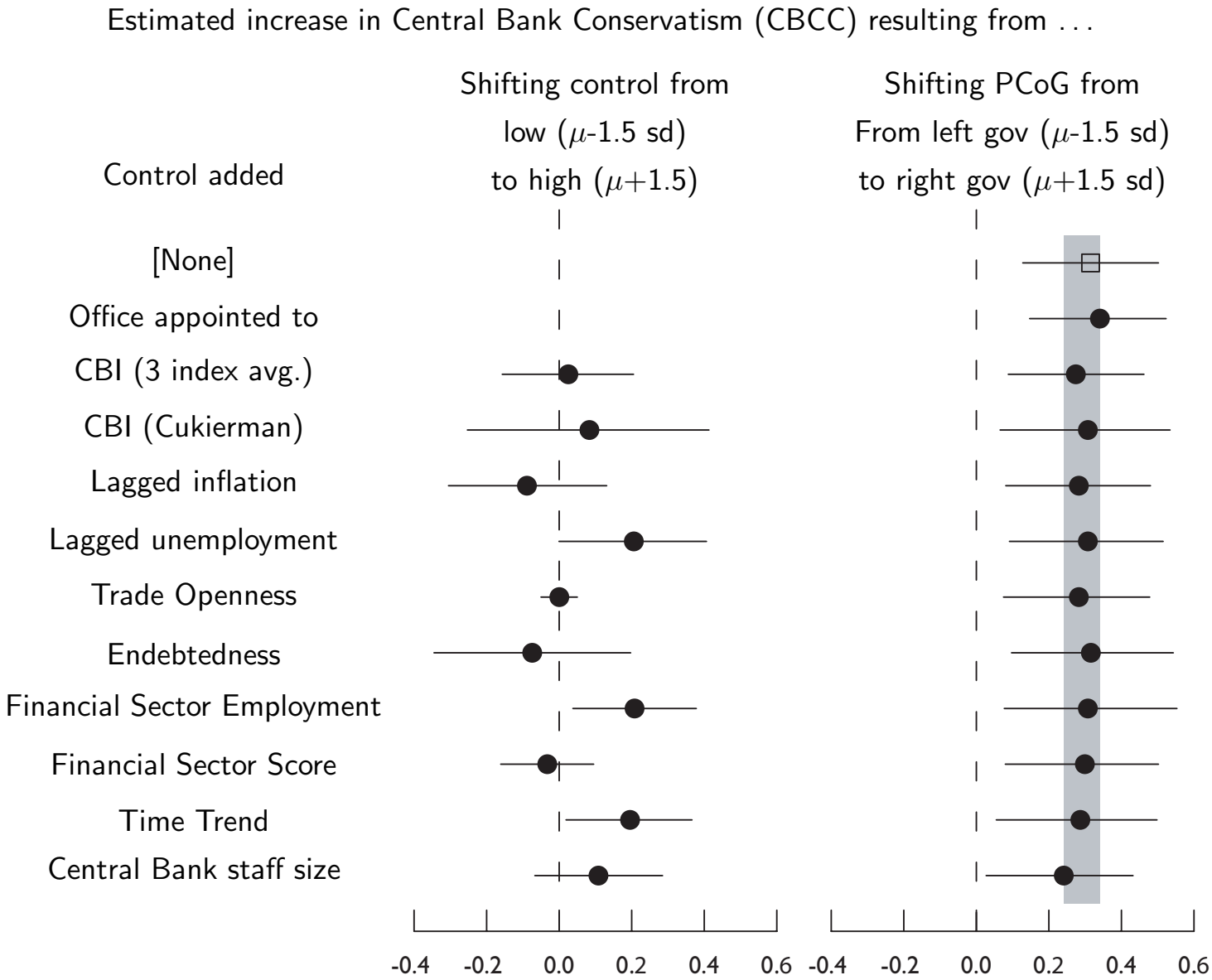
Conservative goverments pick conservative central bankers

Is this finding robust to specification assumptions?

In my case, the statistical model is so demanding it's hard to include many regressors at once.

So try one at a time, and show a long "ropeladder" plot. . .

Robustness Ropeladder: Partisan central banker appointment



Anatomy of a ropeladder plot

I call this a **ropeladder** plot.

The column of dots shows the relationship between Y and a specific X under different model assumptions

Each entry corresponds to a different assumption about the specification, or the measures, or the estimation method, etc.

Anatomy of a ropeladder plot

I call this a **ropeladder** plot.

The column of dots shows the relationship between Y and a specific X under different model assumptions

Each entry corresponds to a different assumption about the specification, or the measures, or the estimation method, etc.

If all the dots line up, with narrow, similar CIs, we say the finding is robust, and reflects the data under a range of reasonable assumptions

If the ropeladder is "blowing in the wind", we may be skeptical of the finding. It depends on model assumptions that may be controversial

Anatomy of a ropeladder plot

I call this a **ropeladder** plot.

The column of dots shows the relationship between Y and a specific X under different model assumptions

Each entry corresponds to a different assumption about the specification, or the measures, or the estimation method, etc.

If all the dots line up, with narrow, similar CIs, we say the finding is robust, and reflects the data under a range of reasonable assumptions

If the ropeladder is "blowing in the wind", we may be skeptical of the finding. It depends on model assumptions that may be controversial

The shaded gray box shows the full range of the point estimates for the QoI.

Narrow is better.

Why ropeladders?

1. Anticipate objections on model assumptions, and have concrete answers.

   Avoid: “I ran it that other way, and it came out the ‘same’.”

   Instead: “I ran it that other way, and look —it made no *substantive* or *statistical* difference worth speaking of.”

   Or: “. . . it makes *this* much difference.”

Why ropeladders?

1. Anticipate objections on model assumptions, and have concrete answers.

   Avoid: “I ran it that other way, and it came out the ‘same’.”

   Instead: “I ran it that other way, and look —it made no *substantive* or *statistical* difference worth speaking of.”

   Or: “. . . it makes *this* much difference.”

2. Investigate robustness more thoroughly.

   Traditional tabular presentation would have run to 7 pages, making comparison hard and discouraging a thorough search

Why ropeladders?

1. Anticipate objections on model assumptions, and have concrete answers.

   Avoid: “I ran it that other way, and it came out the ‘same’.”

   Instead: “I ran it that other way, and look —it made no *substantive* or *statistical* difference worth speaking of.”

   Or: “. . . it makes *this* much difference.”

2. Investigate robustness more thoroughly.

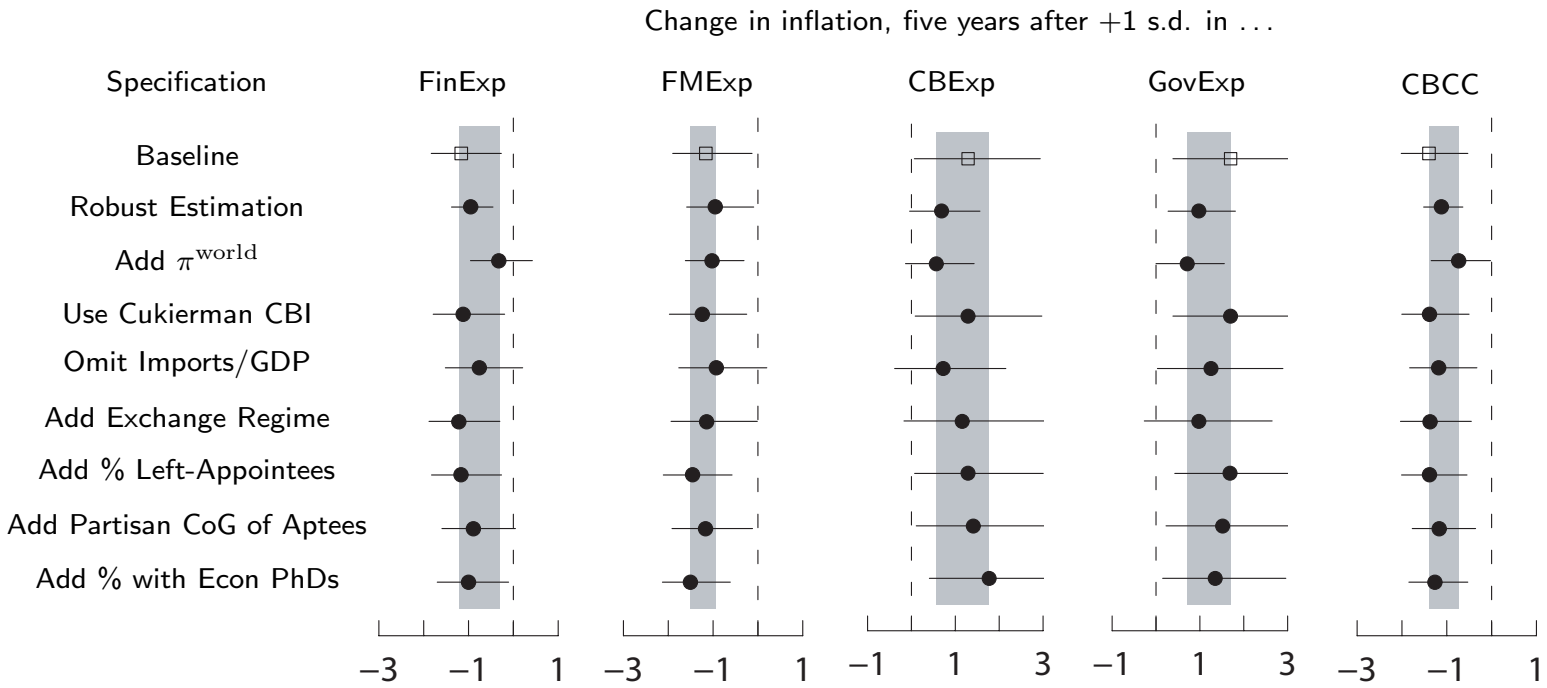
   Traditional tabular presentation would have run to 7 pages, making comparison hard and discouraging a thorough search

3. Find patterns of model sensitivity.

   Two seemingly unrelated changes in specification had the same effect. (Unemployment and Financial Sector Size)

   Turned out to be a missing third covariate. (Time trend)

Robustness for several QoIs at once



Each ropeladder, or column,
shows the effect of a different variable on the response

That is, reading *across* shows the results from a single model

Reading down shows the results for a single question across different models

Chinese Officials Redux

Earlier, I showed the relationship between many different covariates and the expected political rank of Chinese officials

I also showed how these relationships changed over time, making for a complex plot

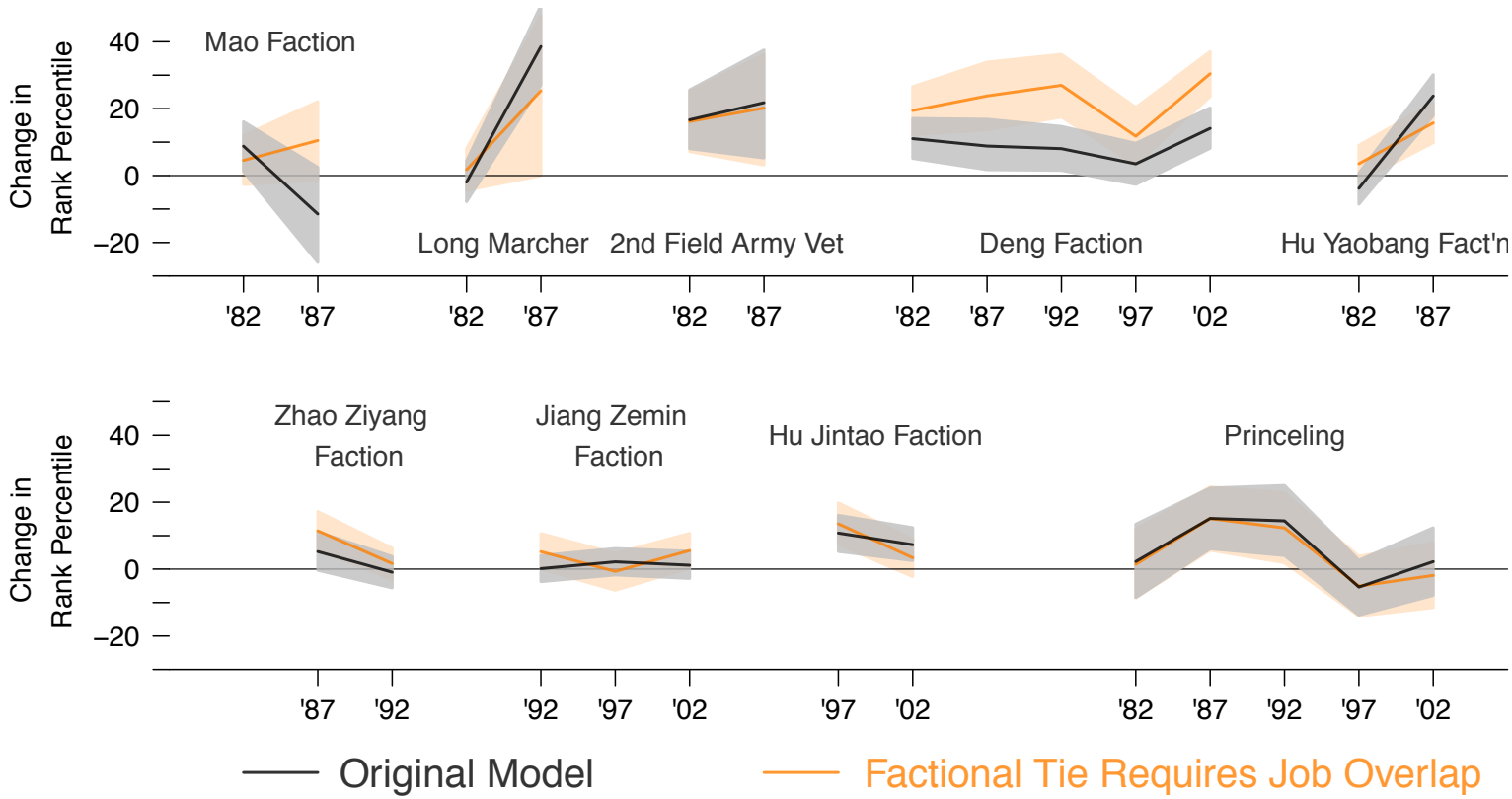
But our findings were controversial: countered the widely accepted belief that Chinese officials are rewarded for economic performance

Critics asked for lots of alternative specifications to probe our results

I can use `tile` to show how exactly what difference these robustness checks made using overlapping lineplots

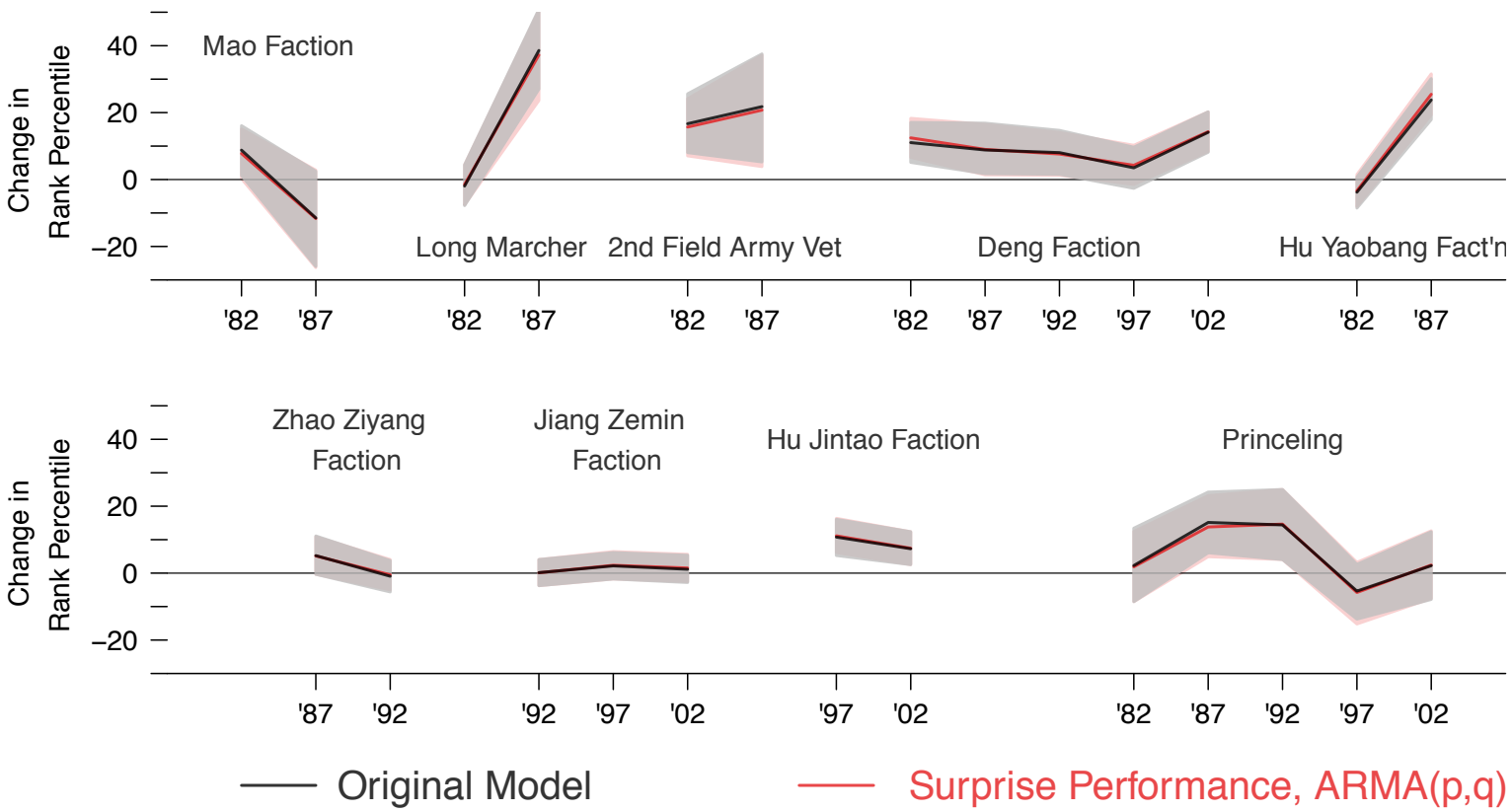
Some critics worried that our measures of faction were too sensitive, so we considered a more specific alternative

This didn't salvage the conventional wisdom on growth. . .



But did (unsuprisingly) strengthen our factional results

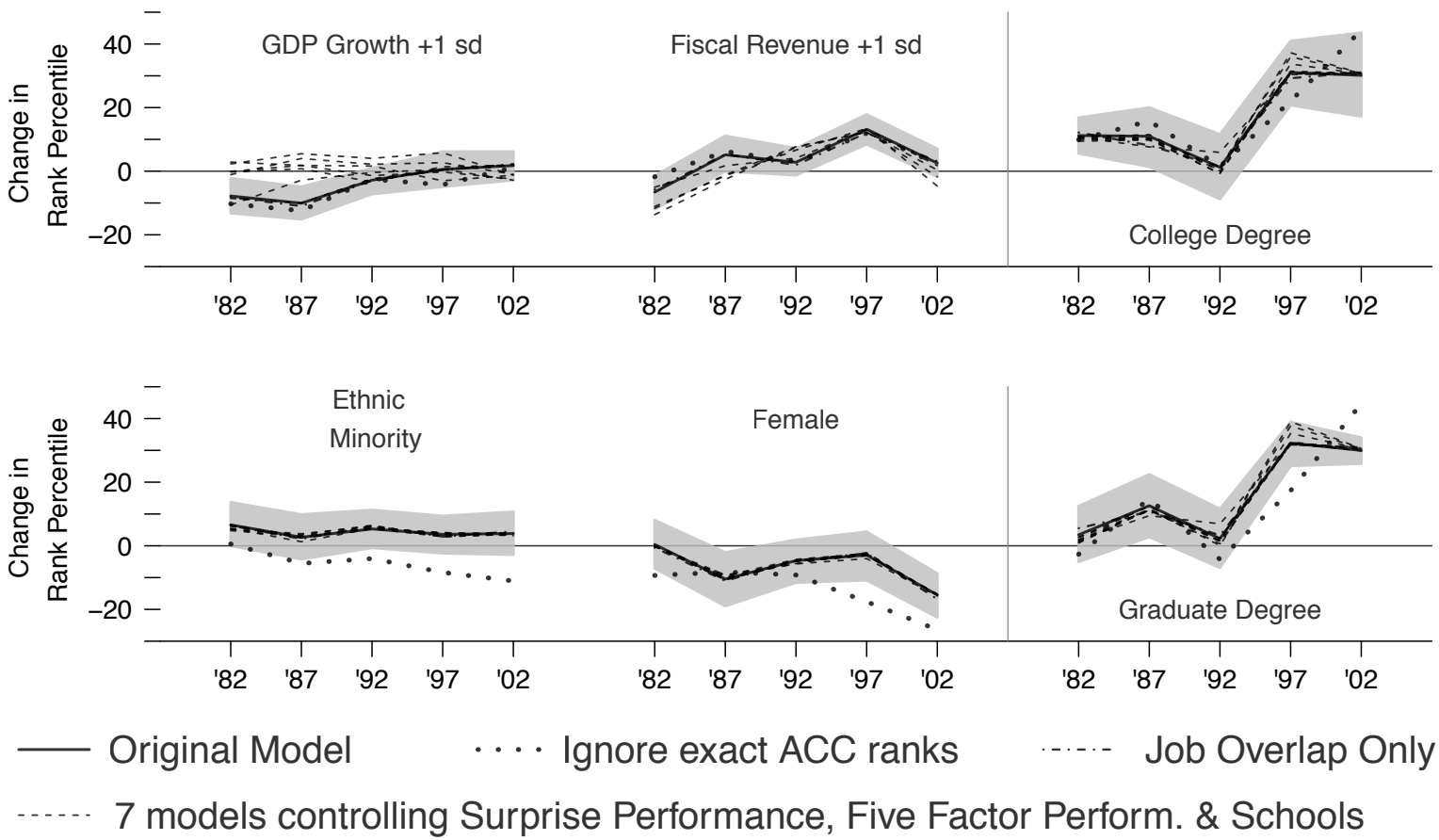
(Specific measures pick up the strongest ties)



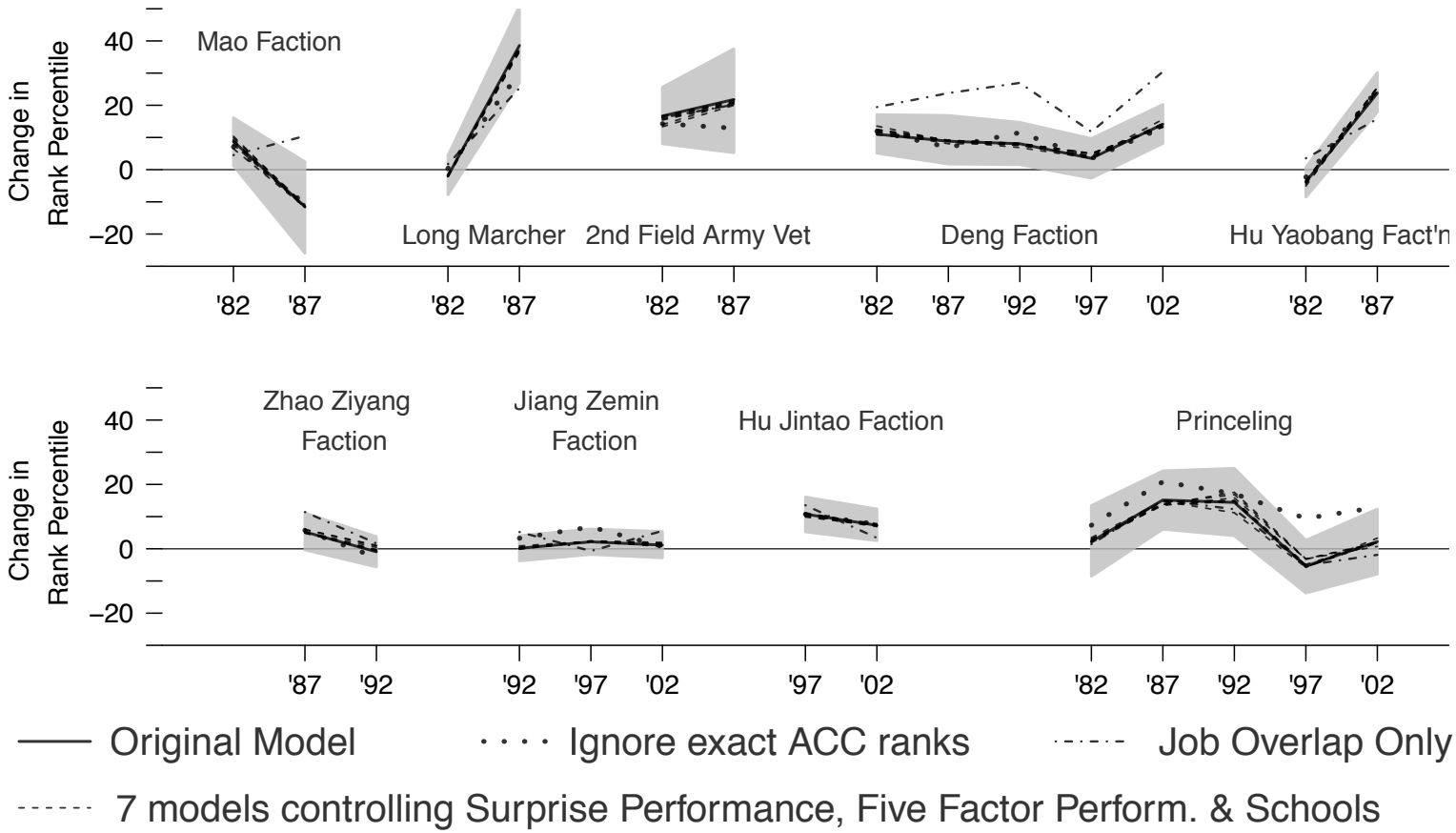
Other critics worried about endogeneity or selection effects flowing from political power to economic performance

We used measures of unexpected growth to zero in on an official's own performance in office—which still nets zero political benefit

But is there a more efficient way to show that our results stay essentially the same?



In our printed article, we show only this plot, which overlaps the full array of robustness checks



Conveys hundreds of separate findings in a compact, readable form

No knowledge of Bayesian methods or partial rank coefficients required!

Bonus Example: Allocation of Authority for Health Policy

Adolph, Greer, and Fonseca consider the problem of explaining whether local, regional, or national European government have power over specific health policy areas and instruments

Areas: Pharamceuticals, Secondary/Tertiary, Primary Care, Public Health

Instruments: Frameworks, Finance, Implementation, Provision

Each combination for each country is a case

Bonus Example: Allocation of Authority for Health Policy

Adolph, Greer, and Fonseca consider the problem of explaining whether local, regional, or national European government have power over specific health policy areas and instruments

Areas: Pharamceuticals, Secondary/Tertiary, Primary Care, Public Health

Instruments: Frameworks, Finance, Implementation, Provision

Each combination for each country is a case

Fiscal federalism suggests lower levels for information-intensive policies and higher levels for policies with spillovers or public goods

Also control for country characteristics and country random effects

With 3 nominal outcomes for each case, need a multilevel multinomial logit

Bonus Example: Allocation of Authority for Health Policy

Covariates:

Policy area	Nominal
Policy instrument	Nominal
Regions old or new	Binary
Country size	Continuous
Number of regions	Continuous
Mountains	Countinuous
Ethnic heterogeneity	Continuous

Tricky part to the model:
some cases have structural zeros for regions (when they don't exist!)

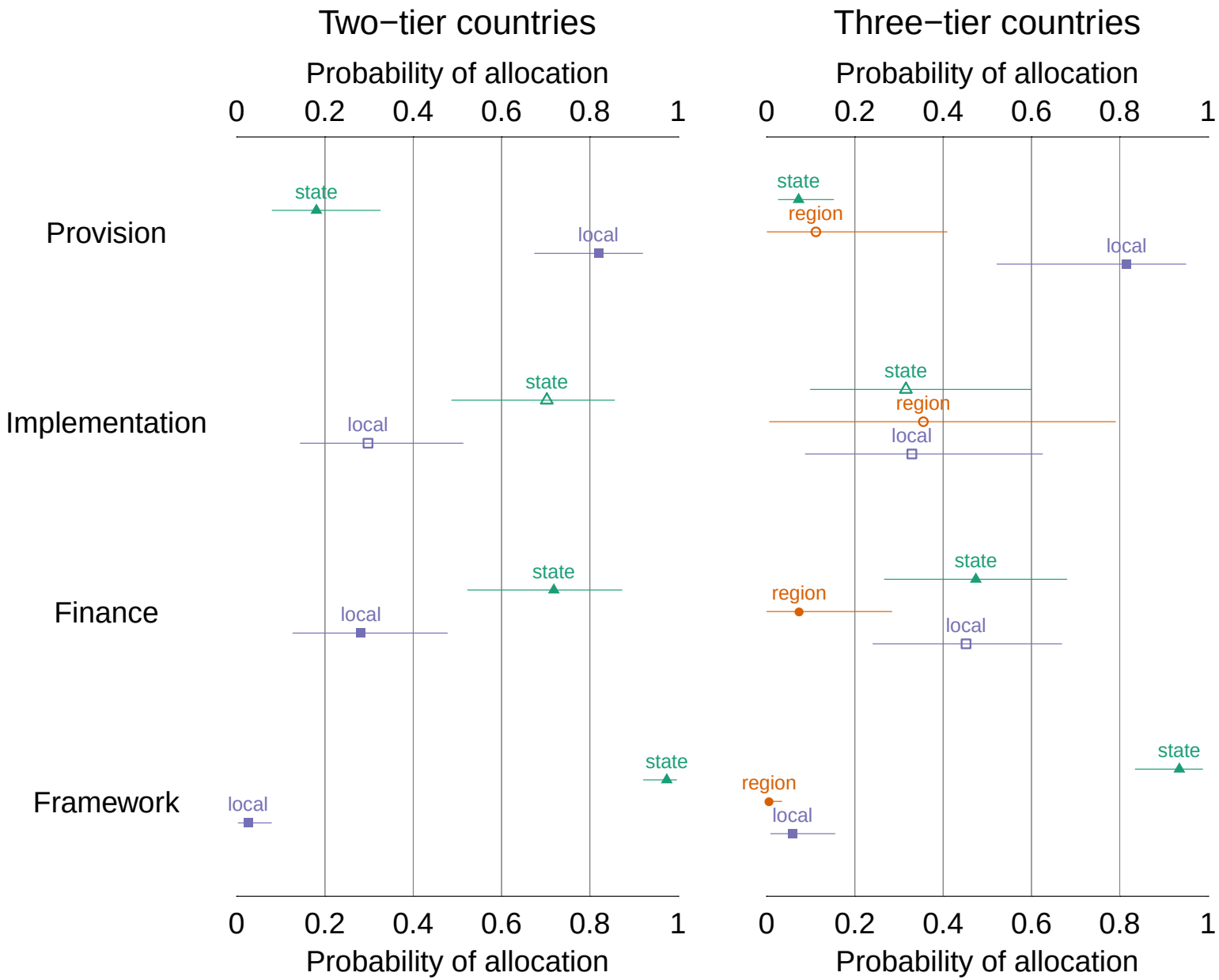
How to set up counterfactuals?

We could set all but one covariate to the mean, then predict the probability of each level of authority given varied levels of the remaining covariate

We should do this separately for countries with and without regional governments

Let's fix everthing but policy instrument to the mean values, then simulate the probability of authority at each level for each instrument

We show the results using a "nested" dot plot, made using `ropeladder()` in the `tile` package


Two−tier countries
Three−tier countries
Probability of allocation
0
0.2
0.4
0.6
0.8
1
Provision
Implementation
Finance
Framework
state
region
local

Special plots for compositional data

Probabilities have a special property: they sum to one

Variables that sum to a constraint are compositional

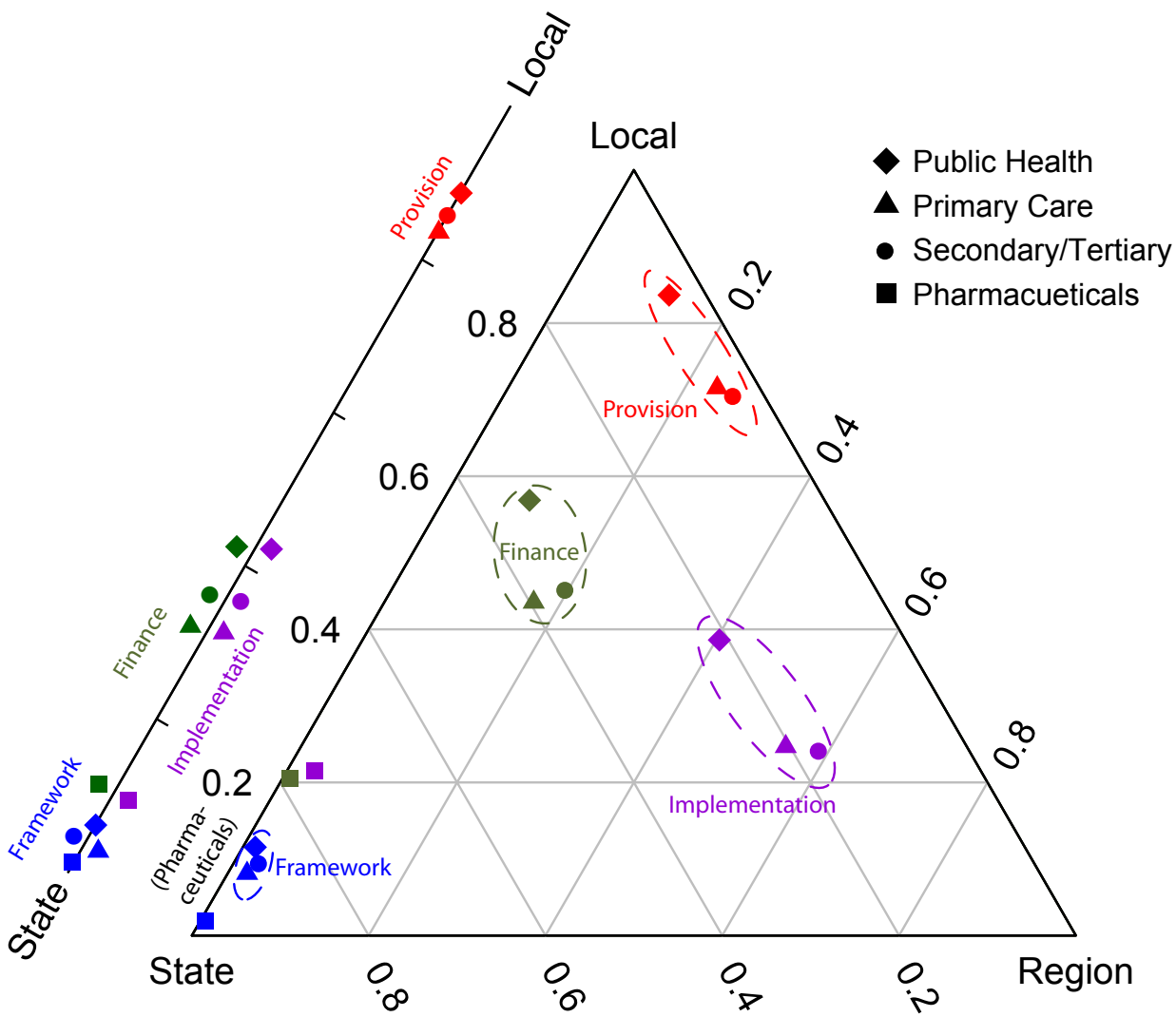
We can plot a two-part composition on a line,

and a three-part composition on a triangular plot

This makes it easier to show more complex counterfactuals, such as every combination of policy area and instrument

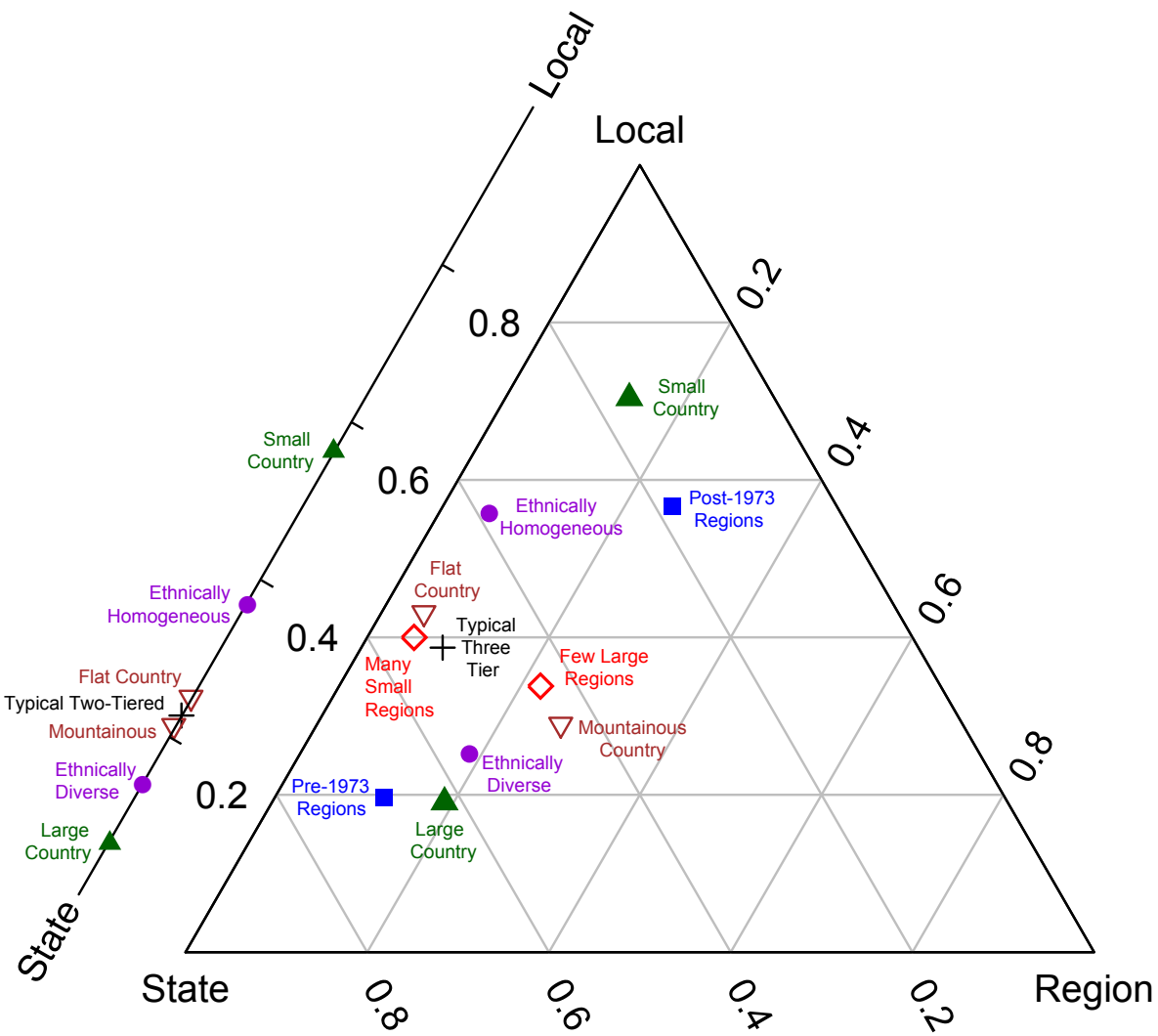
But we also need to work harder to explain these plots

Probability of allocation of authority by policy type



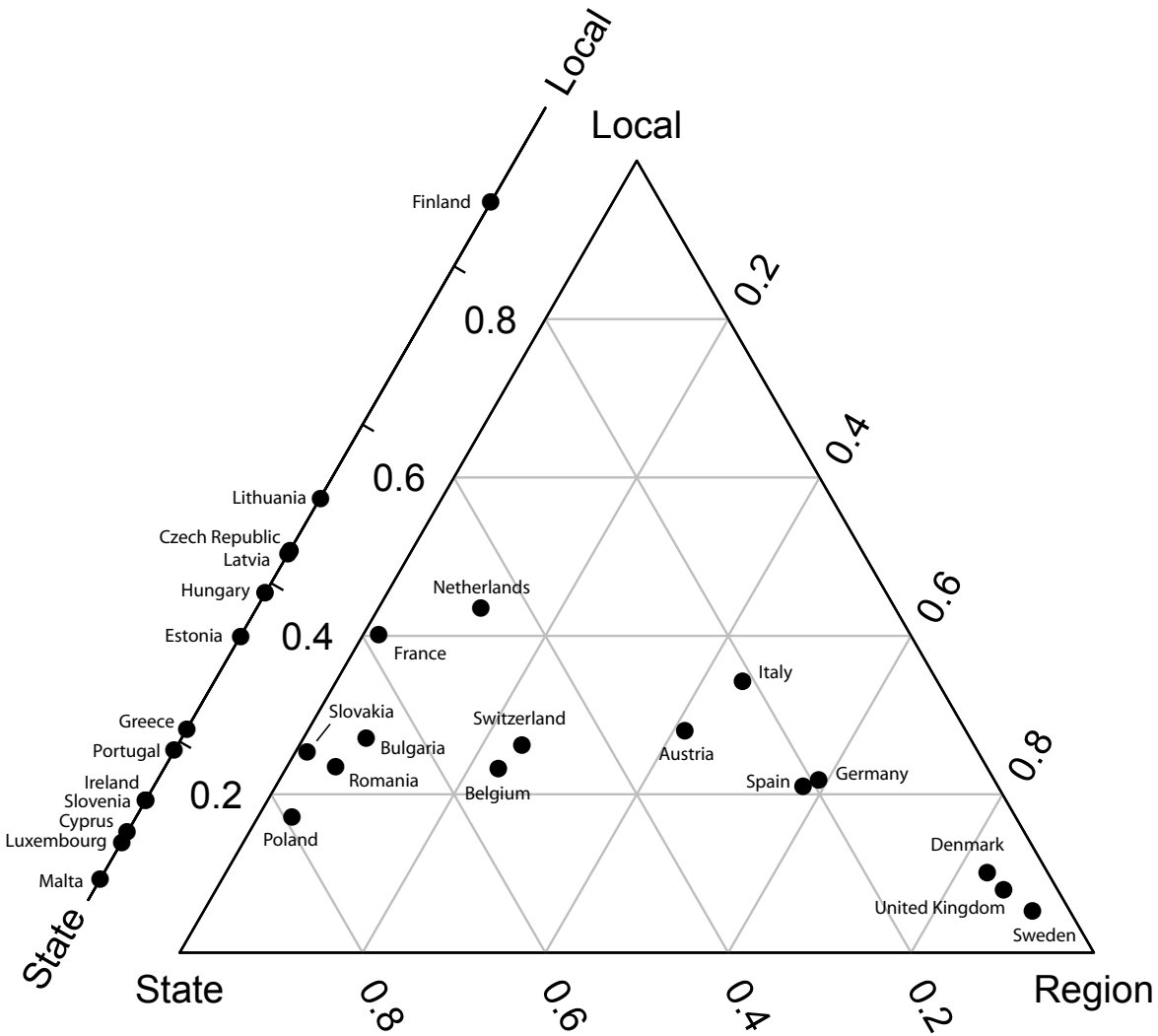
Above holds country characteristics at their means

Probability of allocation of authority by country type



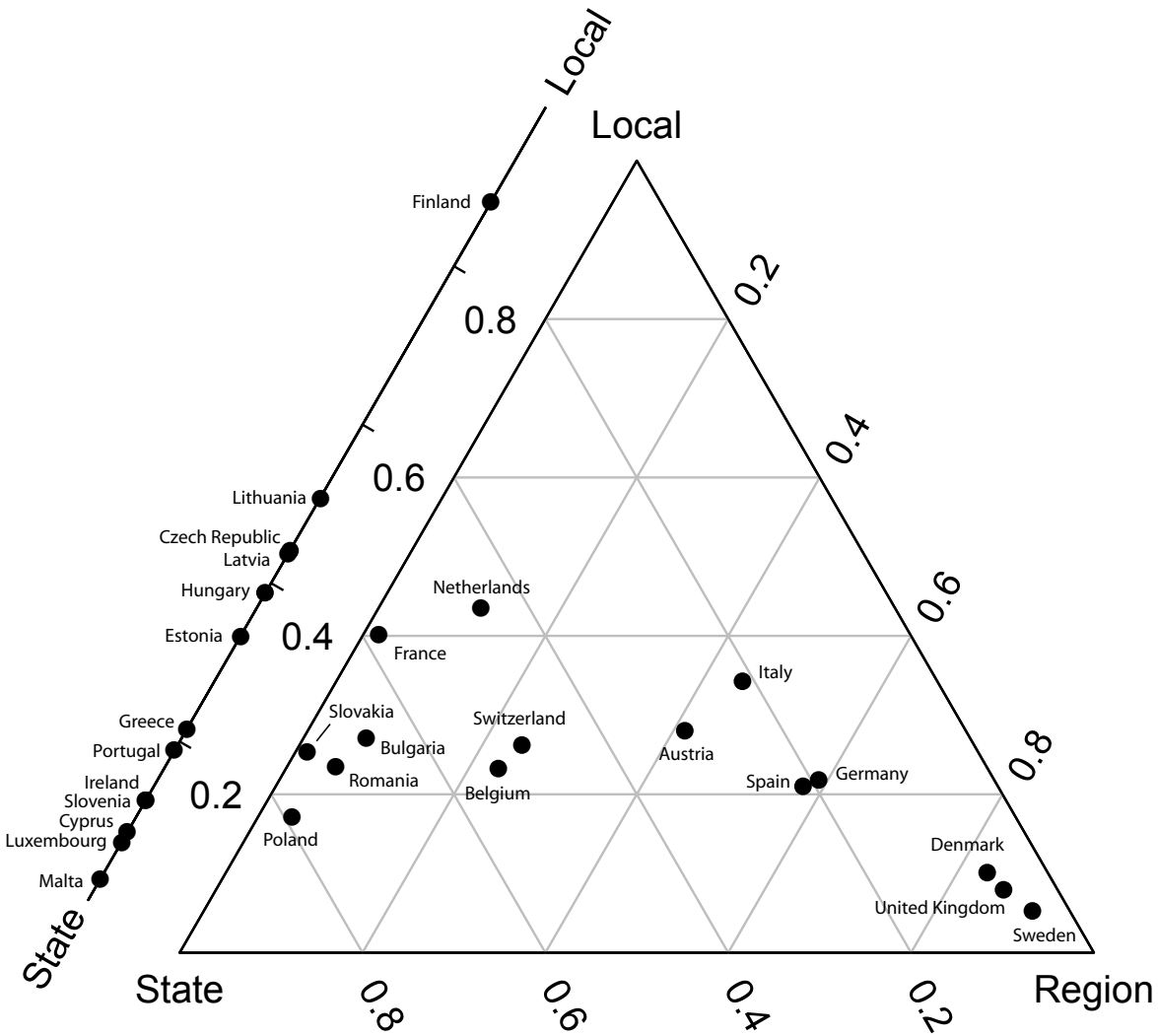
Above holds policy area and instrument “at their means”

Residual country effects



Looking at the country random effects might suggest omitted variables

Residual country effects



On a previous iteration, mountainous countries clustered as high Pr(Regions)

Lessons for practice of data analysis

Simulation + Graphics can summarize complex models for a broad audience

You might even find something you missed as an analyst

And even for fancy or complex models, we can and should show uncertainty

Payoff to programming: this is hard the first few times, but gets easier

Code is re-usable, and encourages more ambitious modeling

Teaching with `tile`

`tile` helps clarify data and models in research

Also helps in teaching statistical models

I incorporate this software throughout our graduate statistics sequence

Greatly aids intuitive understanding of models

Find out much more, and download the software, from:

`faculty.washington.edu/cadolph`

Building a scatterplot: Supplementary exploratory example

In my graphics class, I have students build a scatterplot “from scratch”

This helps us see the many choices to make, and implications for:

1. perception of the data
2. exploration of relationships
3. assessment of fit

A good warm up for `tile` before the main event (application to models)

See how `tile` helps follow Tufte recommendations

Building a scatterplot: Redistribution example

Data on

political party systems

and

redistributive effort from various industrial countries

Source of data & basic plot:

Torben Iversen & David Soskice, 2002,
“Why do some democracies redistribute more than others?” Harvard University.

Building a scatterplot: Redistribution example

Concepts for this example (electoral systems and the welfare state):

Effective number of parties:

Building a scatterplot: Redistribution example

Concepts for this example (electoral systems and the welfare state):

Effective number of parties:

- # of parties varies across countries
- electoral rules determine #
  - Winner take all (US) $\rightarrow\sim$ 2 parties.
  - Proportional representation $\rightarrow$ more parties
- To see this, need to discount trivial parties.

Building a scatterplot: Redistribution example

Concepts for this example (electoral systems and the welfare state):

Effective number of parties:

- # of parties varies across countries
- electoral rules determine #
  - Winner take all (US) $\rightarrow\sim$ 2 parties.
  - Proportional representation $\rightarrow$ more parties
- To see this, need to discount trivial parties.

Poverty reduction:

Building a scatterplot: Redistribution example

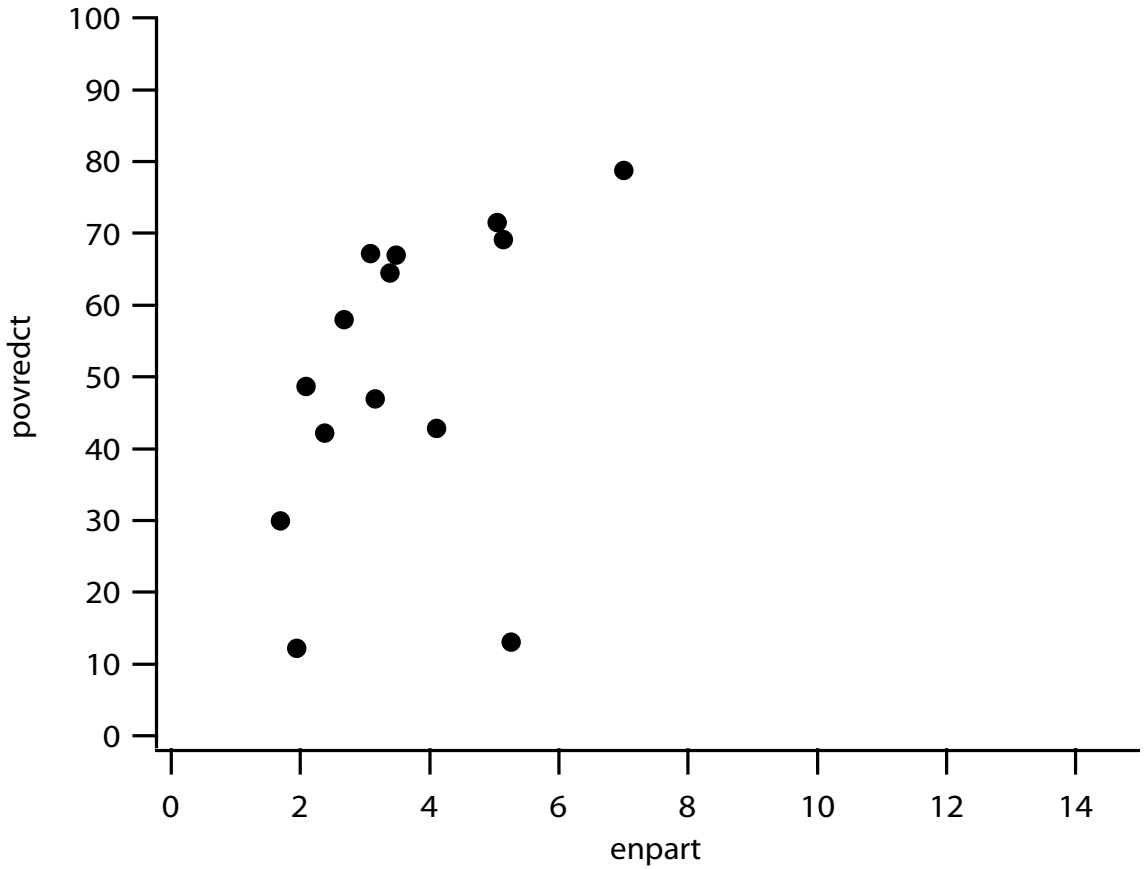
Concepts for this example (electoral systems and the welfare state):

Effective number of parties:

- # of parties varies across countries
- electoral rules determine #
  - Winner take all (US) $\rightarrow\sim$ 2 parties.
  - Proportional representation $\rightarrow$ more parties
- To see this, need to discount trivial parties.

Poverty reduction:

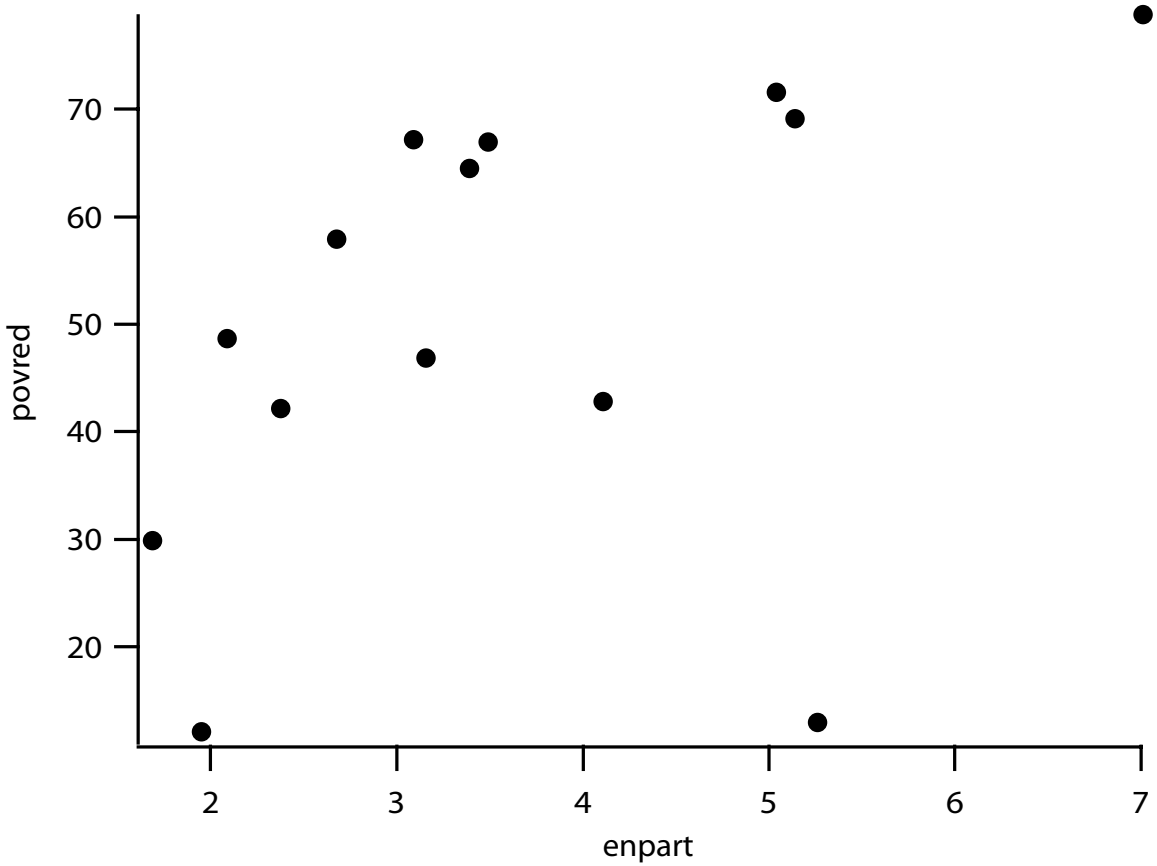
- Percent lifted out of poverty by taxes and transfers.
- Poverty = an income below 50% of mean income.



Initial plotting area is often oddly shaped (I've exaggerated)

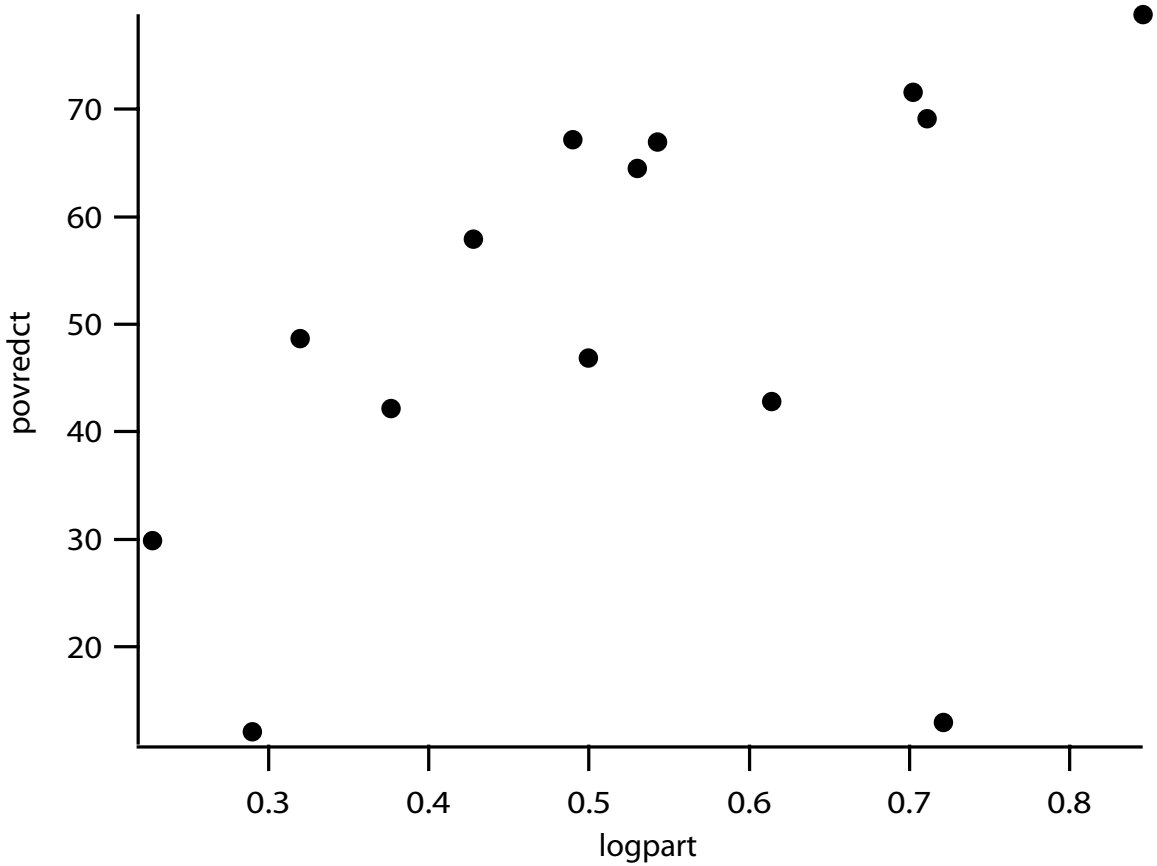
Plotting area hiding relationship here. Sometimes can even exclude data!

Filled circles: okay for a little data; open is better when data overlap



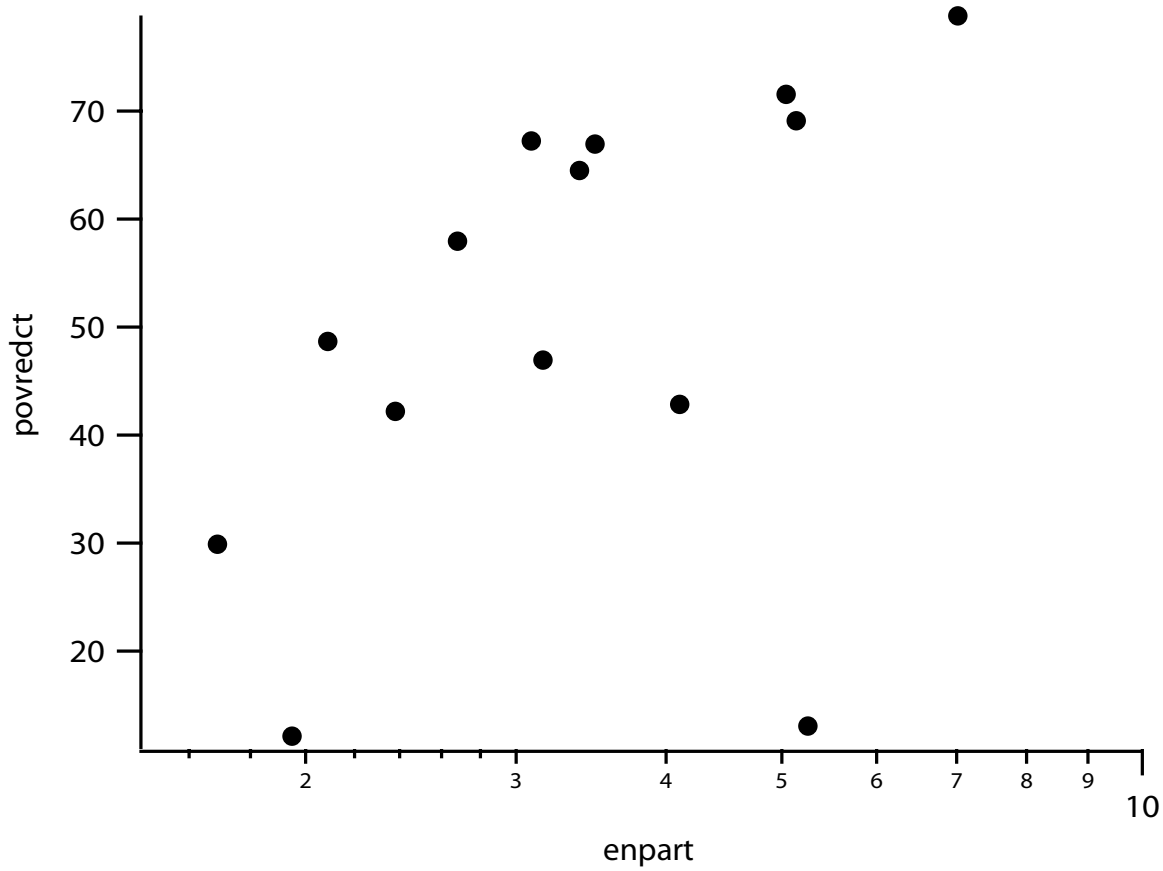
Sensible, data based plot limits

Appears to be a curvilinear relationship. Can bring that out with. . .



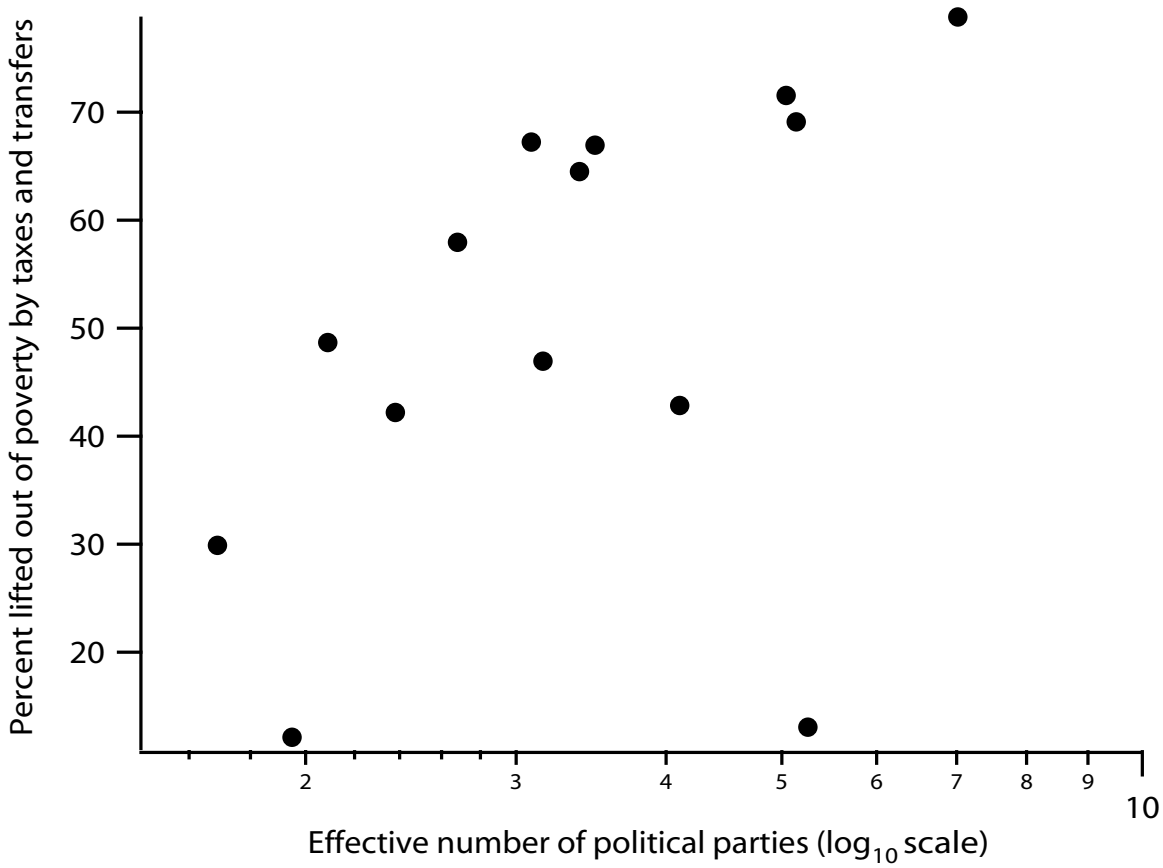
Log scaling.

But what logarthmic base? And why print the exponents?

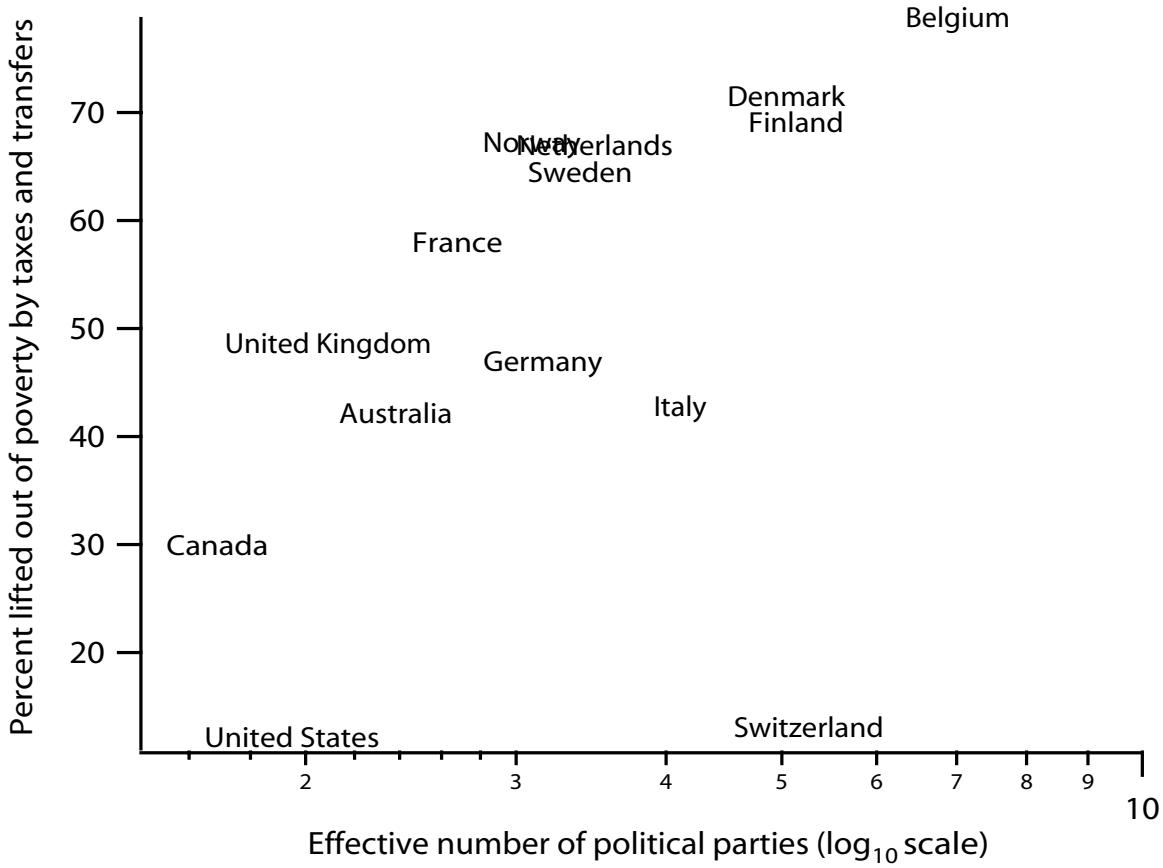


Combine a log scale with linear labels. Now everyone can read. . .

Next: Axis labels people can understand

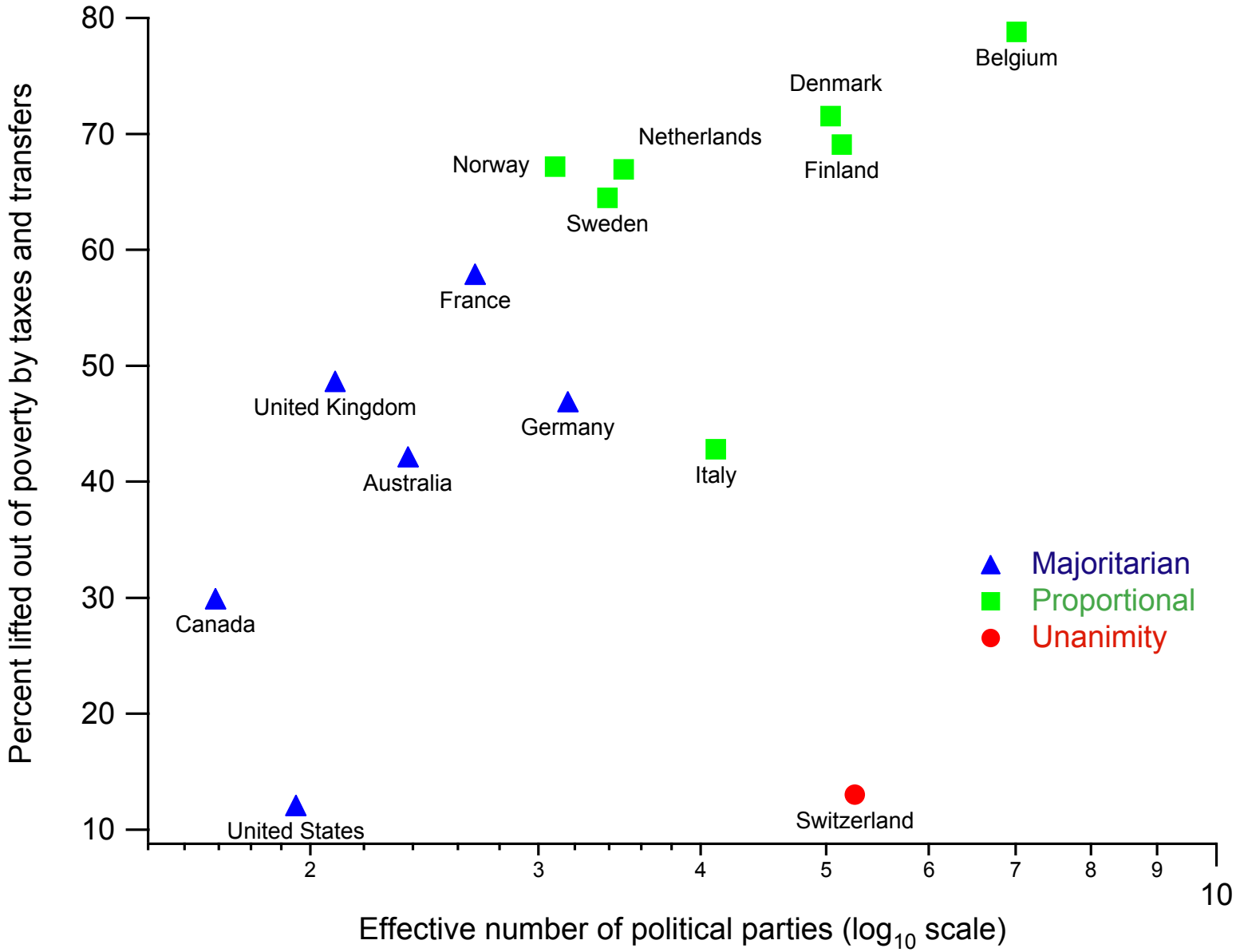


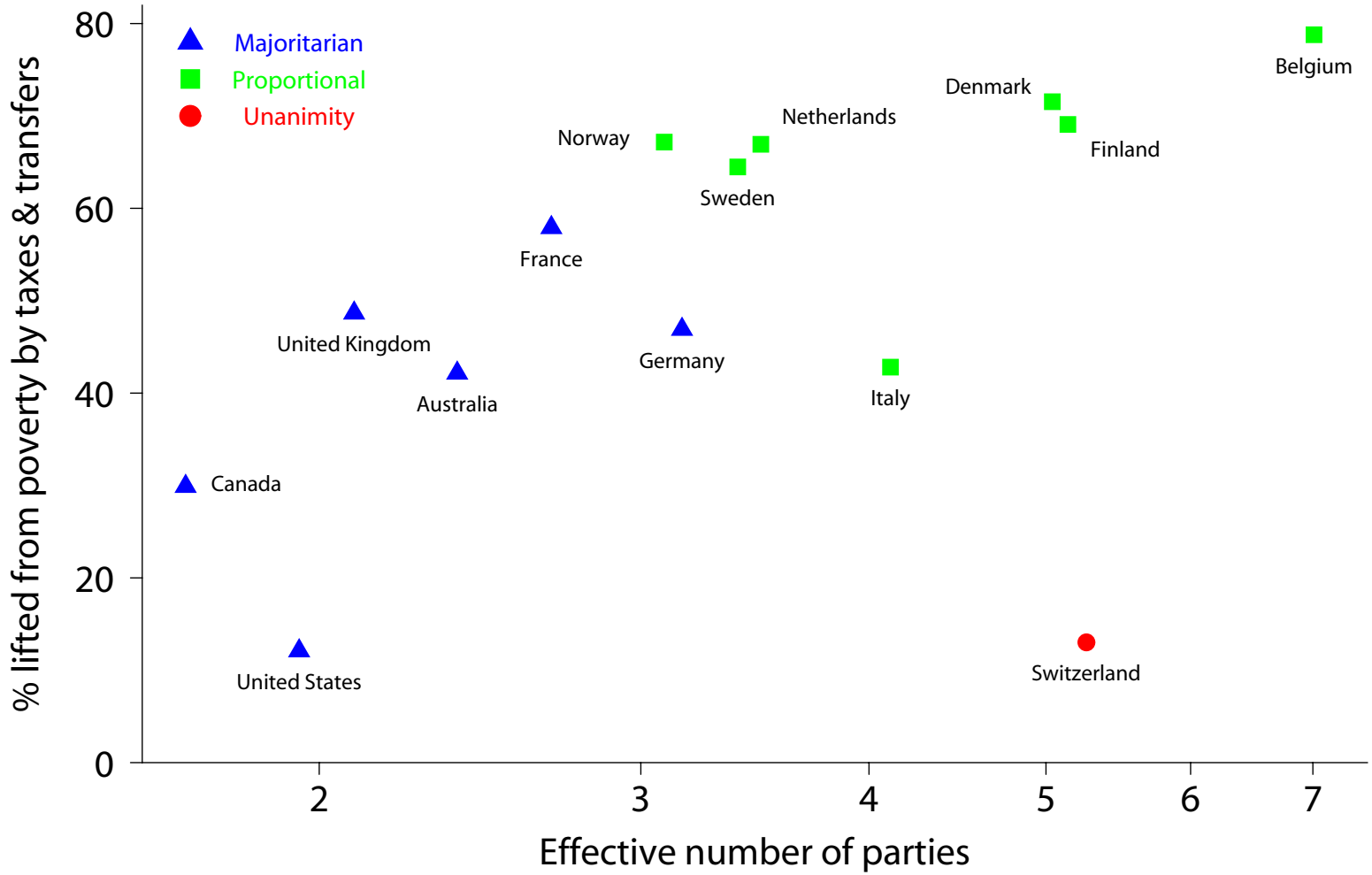
So what's going on the data? What are those outliers?



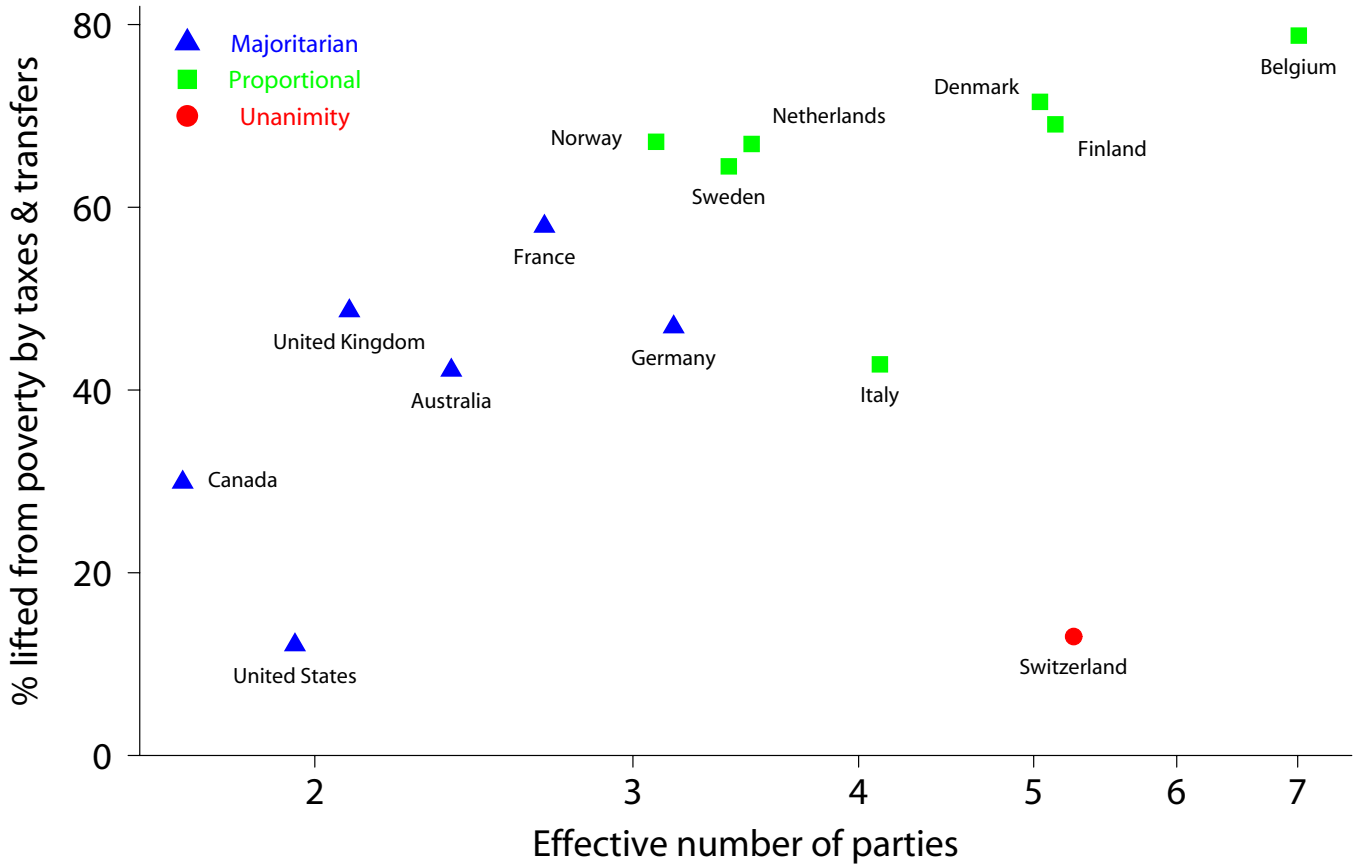
With little data & big outliers, show the name of each case

Now we can try to figure out what makes the US and Switzerland so different


Percent lifted out of poverty by taxes and transfers
80
70
60
50
40
30
20
10
Belgium
Denmark
Netherlands
Finland
Norway
Sweden
France
United Kingdom
Germany
Italy
Australia
Canada
United States
Switzerland
Majoritarian
Proportional
Unanimity
2
3
4
5
6
7
8
9
10
Effective number of political parties (log10 scale)

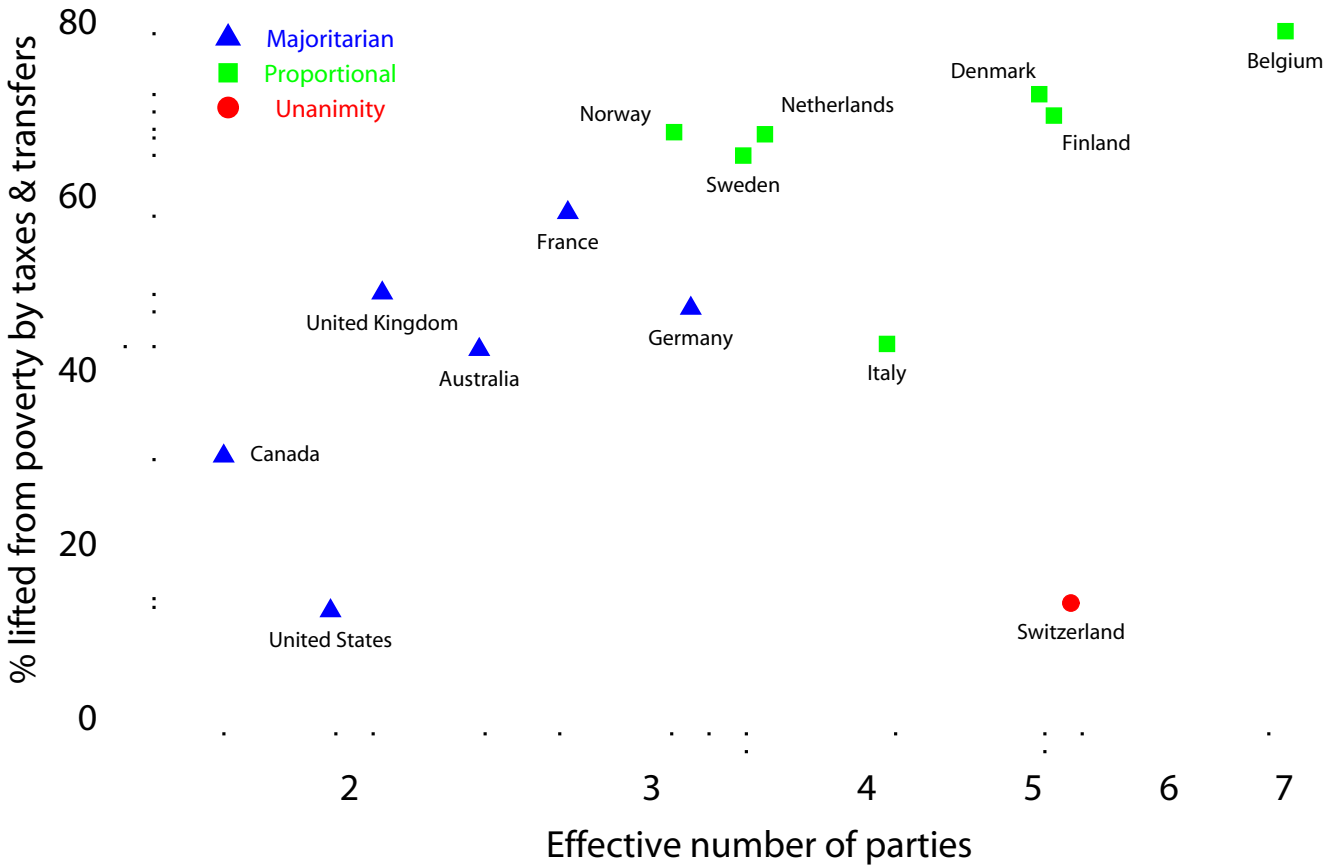


Plot redone using `scatter` (`tile` package in `R`)



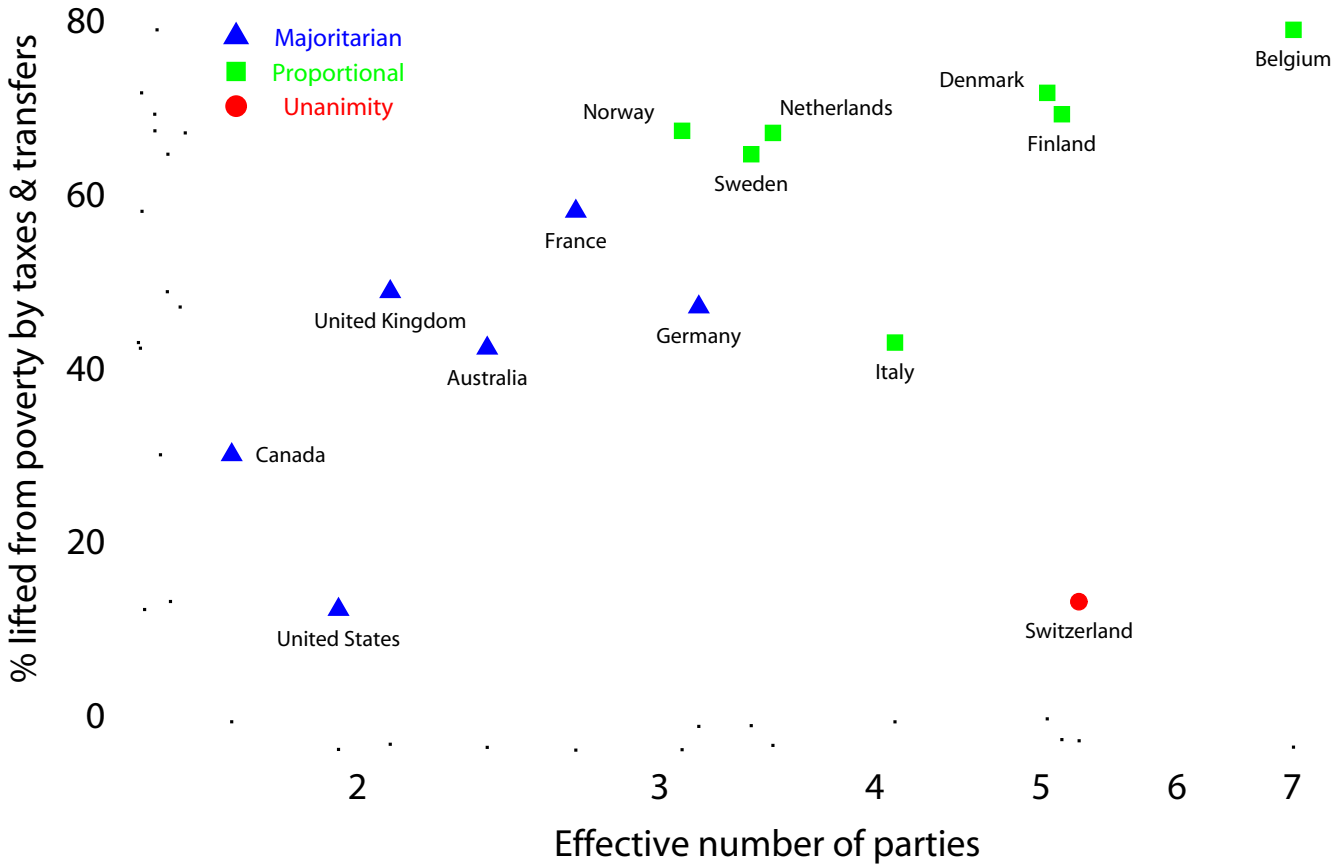
Scatterplots relate two distributions.

Why not make those marginal distributions explicit?

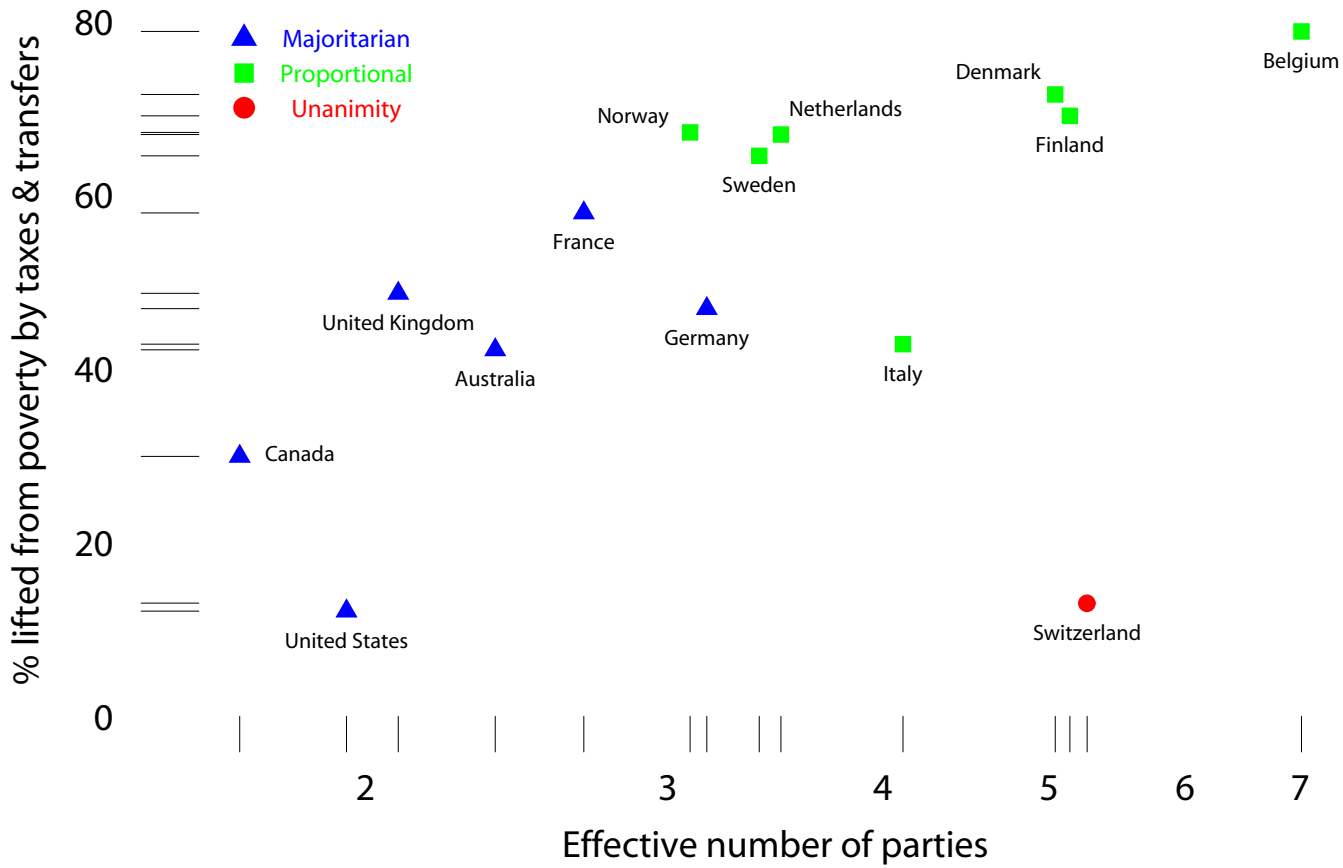


`Rugs` accomplish this by replacing the axis lines with the plots

We could choose any plotting style: from the histogram-like **dots**. . .

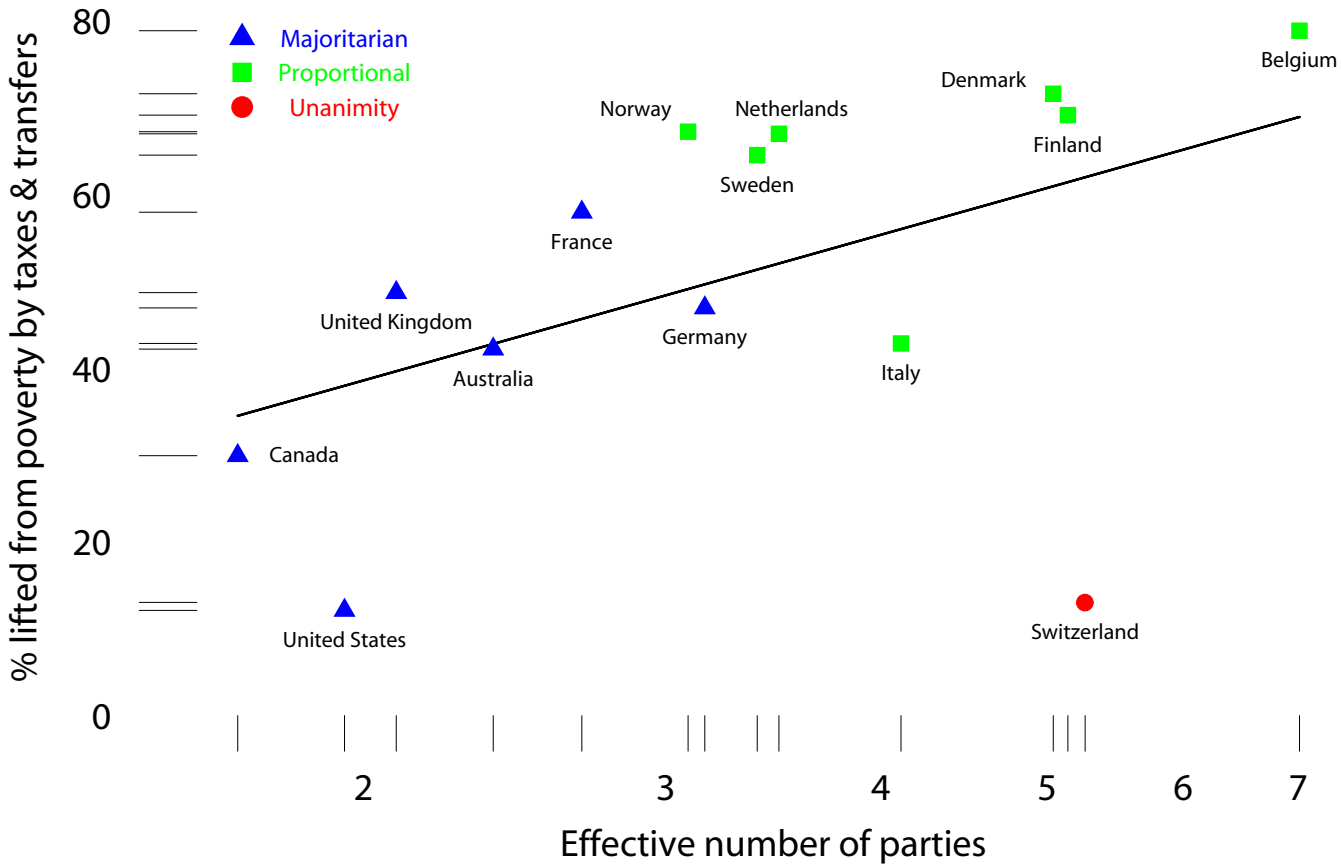


. . . to a strip of `jitter`ed data. . .



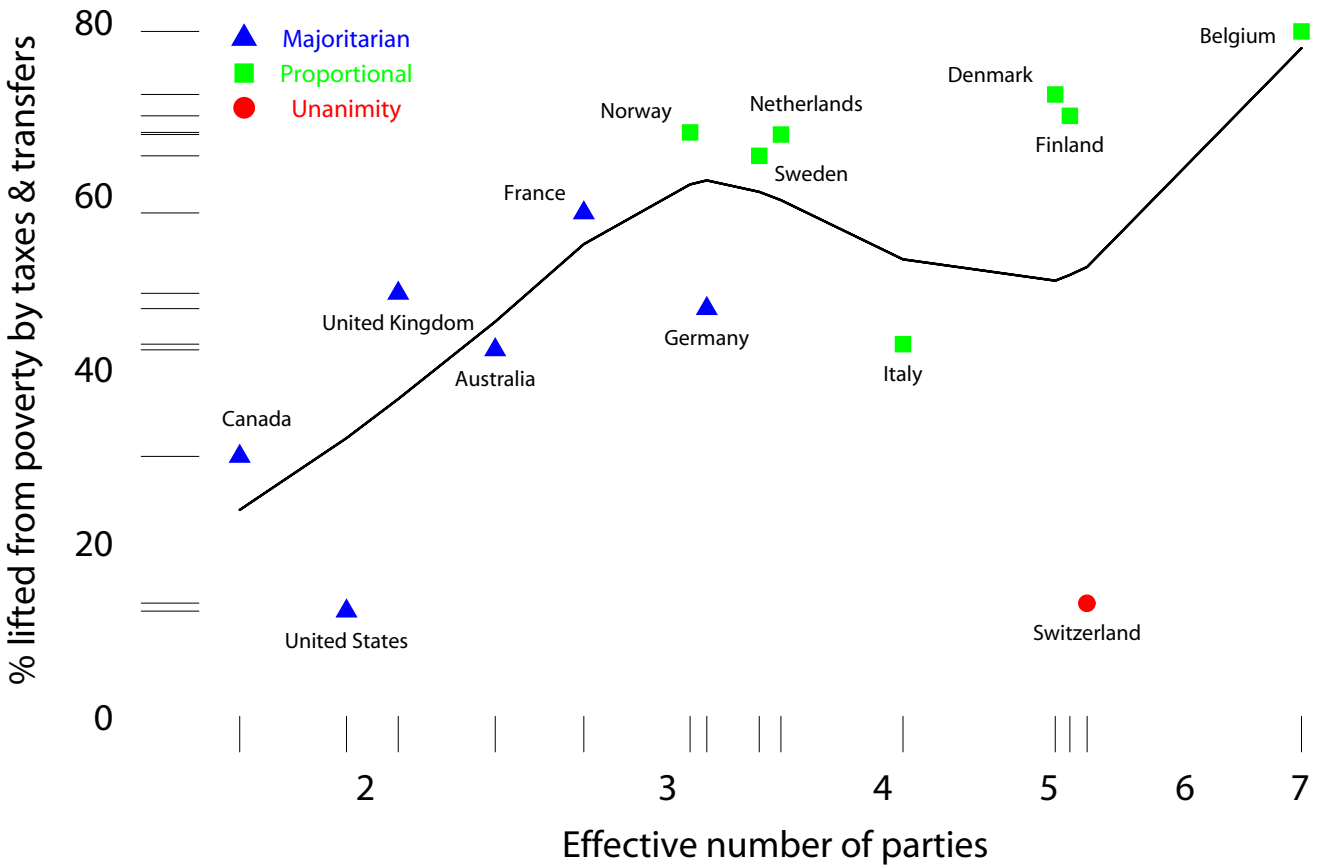
. . . to a set of very thin `lines` marking each observation

Because we have so few cases, thin lines work best for this example



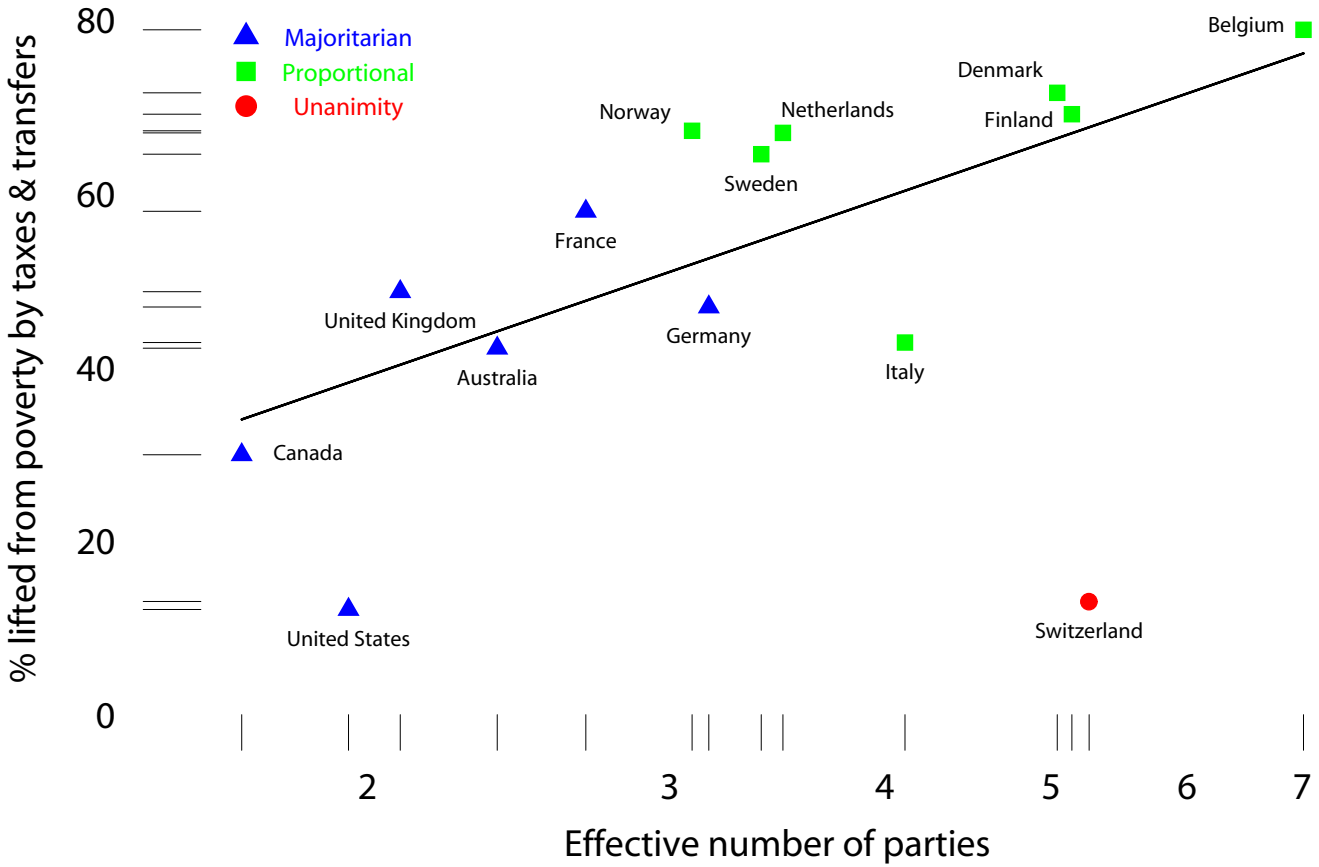
Let's add a parametric model of the data: a least squares fit line

`tile` can do this for us

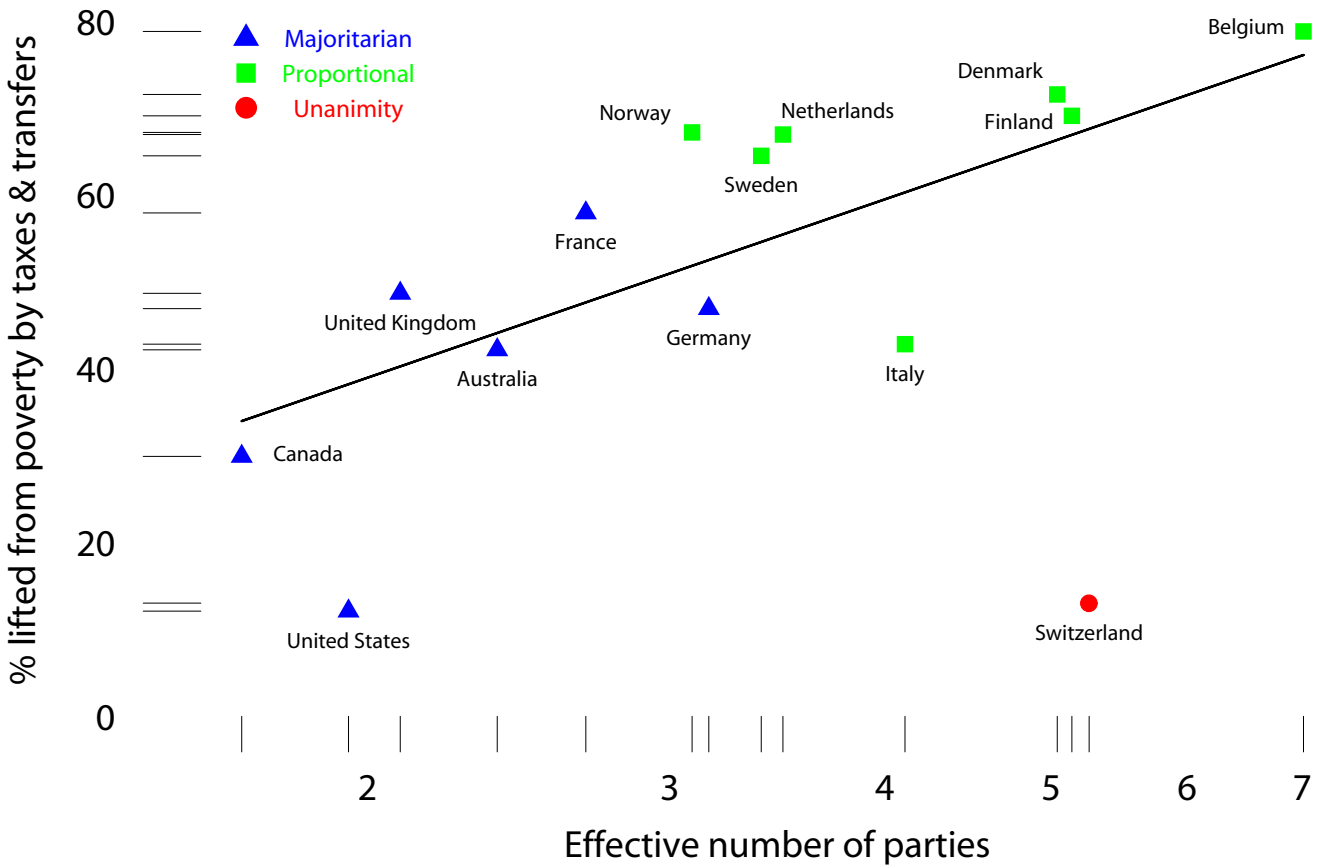


But we don't have to be parametric

A local smoother, like loess, often helps show non-linear relationships

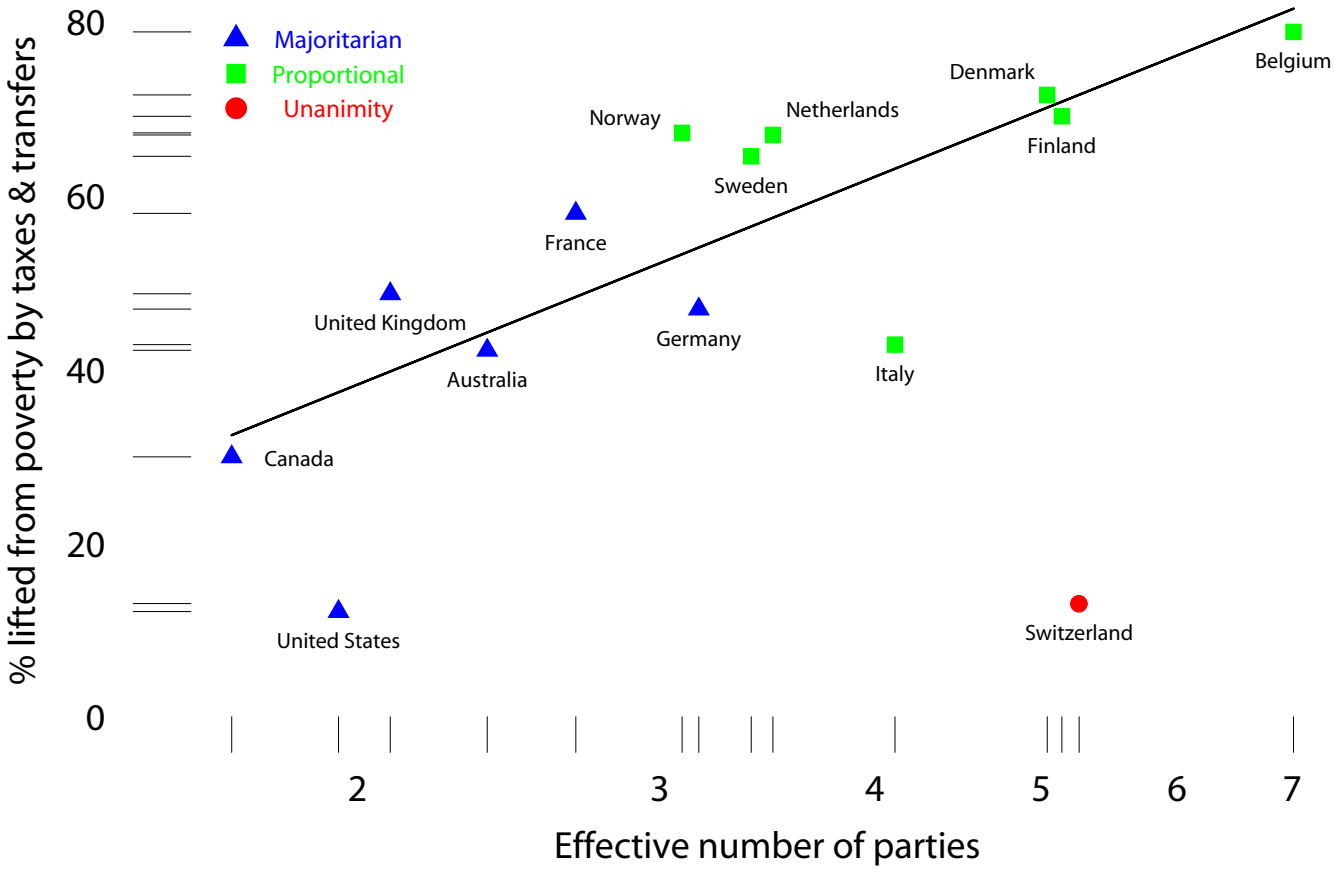


M-estimators weight observations by an influence function to minimize the influence of outliers



Even with an M-estimator, every outlier has some influence

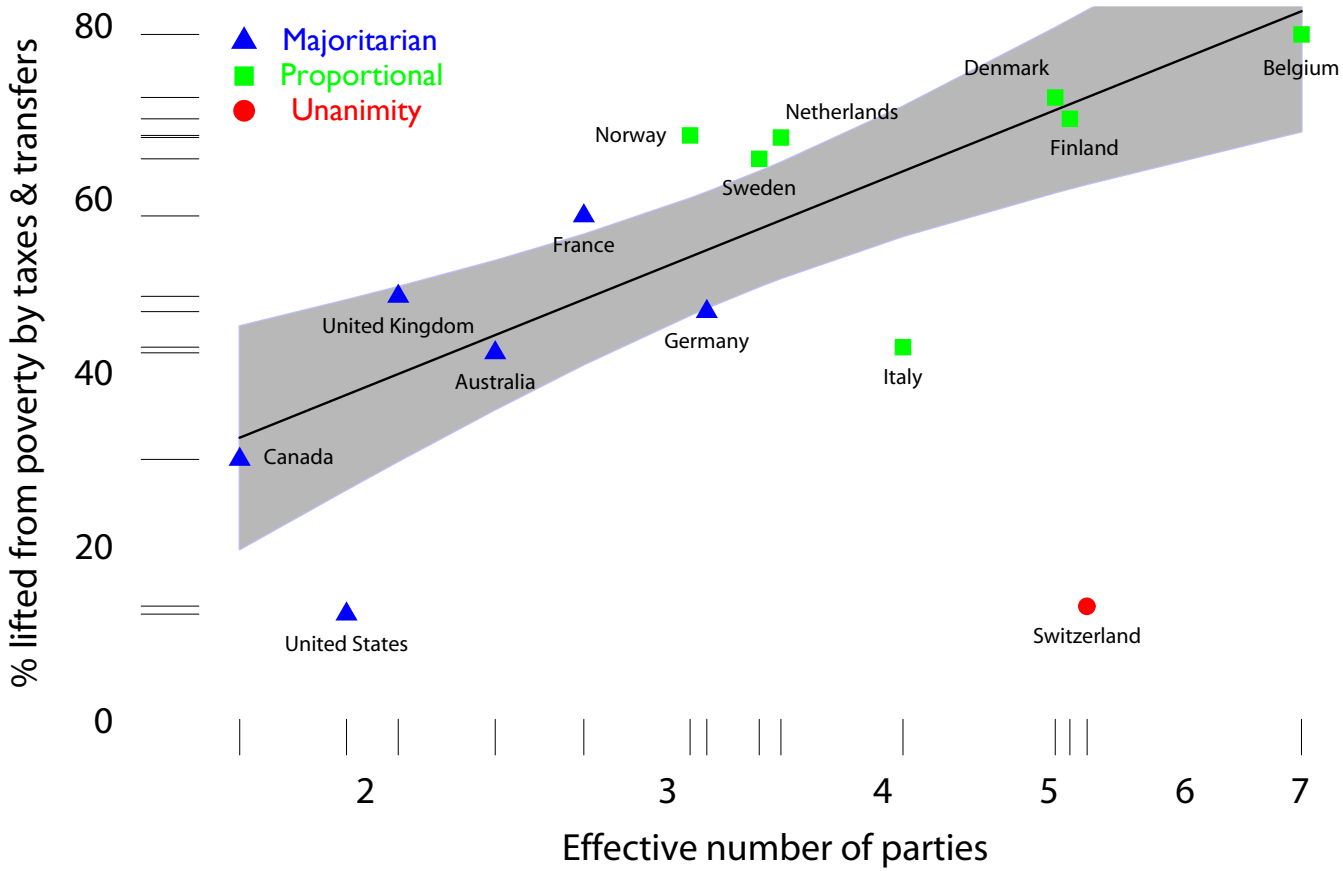
Thus any one distant outlier can bias the result



A robust and resistant MM-estimator, shown above, largely avoids this problem

Only a (non-outlying) fraction of the data influence this fit.

`rlm(method="MM")`



In our final plot, we add 95 percent confidence intervals for the MM-estimator

A measure of uncertainty is essential to reader confidence in the result

Syntax for redistribution scatter

```
library(tile)

# Load data
data <- read.csv("iver.csv", header=TRUE)
attach(data)
```

Syntax for redistribution scatter

```
# First, collect all the data inputs into a series of "traces"

# The actual scattered points
trace1 <- scatter(x = enp, # X coordinate of the data
                  y = povred, # Y coordinate of the data
                  labels = cty, # Labels for each point

                  # Plot symbol for each point
                  pch = recode(system,"1=17;2=15;3=16"),

                  # Color for each point
     col = recode(system,"1='blue';2='darkgreen';3='red'"),

                  # Offset text labels
                  labelyoffset = -0.03,  # on npc scale

                  # Fontsize
                  fontsize = 9,
```

Syntax for redistribution scatter

```
                # Marker size
                size = 1,  # could be vector for bubble plot

                # Add a robust fit line and CI
                fit = list(method = "mmest", ci = 0.95),

                # Which plot(s) to plot to
                plot = 1
                )


# The rugs with marginal distributions
rugX1 <- rugTile(x=enp, type="lines", plot = 1)

rugY1 <- rugTile(y=povred, type="lines", plot = 1)
```

Syntax for redistribution scatter

```
# A legend
legendSymbols1 <- pointsTile(x= c(1.8,    1.8,          1.8),
                             y=  c(78,    74,           70),
                             pch=c(17,    15,           16),
                             col=c("blue", "darkgreen", "red"),
                             fontsize = 9,
                             size=1,
                             plot=1
                             )

legendLabels1 <- textTile(labels=c("Majoritarian",
                                   "Proportional",
                                   "Unanimity"),
                          x=  c(2.05,   2.05,         2.05),
                          y=  c(78,     74,           70),
                          pch=c(17,     15,           16),
                          col=c("blue", "darkgreen", "red"),
                          fontsize = 9,
                          plot=1
                          )
```

Syntax for redistribution scatter

```
# Now, send that trace to be plotted with tile
tile(trace1,                    # Could list as many
     rugX1,                     # traces here as we want
     rugY1,                     # in any order
     legendSymbols1,
     legendLabels1,

     # Some generic options for tile
     RxC = c(1,1),
     #frame = TRUE,
     #output = list(file="iverson1", width=5, type="pdf"),
     height = list(plot="golden"),

     # Limits of plotting region
     limits=c(1.6, 7.5, 0, 82),
```

Syntax for redistribution scatter

```
# x-axis controls
xaxis=list(log = TRUE,
  at = c(2,3,4,5,6,7)
  ),

# x-axis title controls
xaxistitle=list(labels=c("Effective number of parties")),

# y-axis title controls
yaxistitle=list(labels=c("% lifted from poverty by taxes & transfers"

# Plot titles
plottitle=list(labels=("Party Systems and Redistribution"))
)
```