CSSS 594 / POLS 559:
Time Series and Panel Data for the Social Sciences

Multiple Imputation for Panel Data

Christopher Adolph

Department of Political Science
and
Center for Statistics and the Social Sciences
University of Washington, Seattle

Overview

Why listwise deletion can be harmful

Why "crude" imputation is no cure

What multiple imputation does

Special problems for imputing TSCS data

Sources

The methods and ideas emphasized here come from:

Gary King et al (2001) "Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation", *American Political Science Review*

James Honaker and Gary King (2010) "What to Do about Missing Values in Time-Series Cross-Section Data", *American Journal of Political Science*

while the classic source on missing data imputation is

Roderick Little and Donald Rubin (2002), *Statistical Analysis with Missing Data,* 2nd Ed., Wiley.

From a certain point of view, all inference problems are missing data problems; just treat parameters as "missing data"

For today, we will just consider missingness in the dependent variable and covariates

A contrived example

$$y_i = -1 \times x_i + 1 \times z_i + \varepsilon_i$$

$$\begin{bmatrix} x_i \\ z_i \end{bmatrix} \sim \mathcal{MVN}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix} \right)$$

$$\varepsilon \sim \mathrm{N}(0, 4)$$

We will create some data using this model, then delete some of it,
and compare the effectiveness of different methods of coping with missing data

We start with 200 cases, but

In particular, we will assume that x_i has a 80% chance of being missing
if and only if $z_i < \bar{z}_i$, and is observed otherwise

So we end up with about 120 cases fully observed

A contrived example

$$\text{Democracy}_i = -1 \times \text{Inequality}_i + 1 \times \text{GDP}_i + \varepsilon_i$$

$$\begin{bmatrix} x_i \\ z_i \end{bmatrix} \sim \mathcal{MVN}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix} \right)$$

$$\varepsilon \sim \text{N}(0, 4)$$

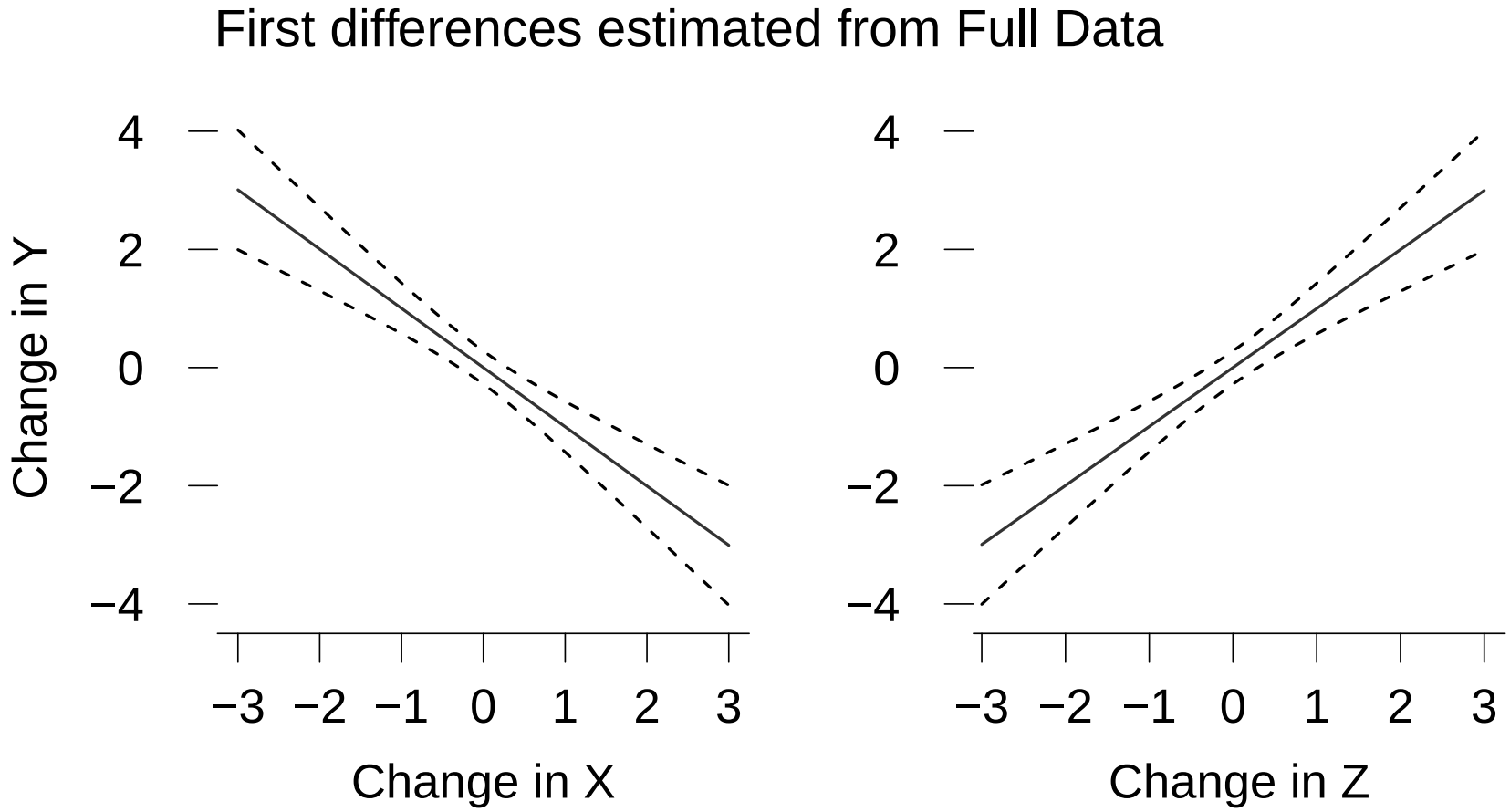
It may help to imagine some context, but remember this example is fictive:

Imagine Democracy is hampered by Inequality and aided by Development,

And Inequality tends to be lower in developed countries,

But poorer countries are less likely to gather data on levels of Inequality

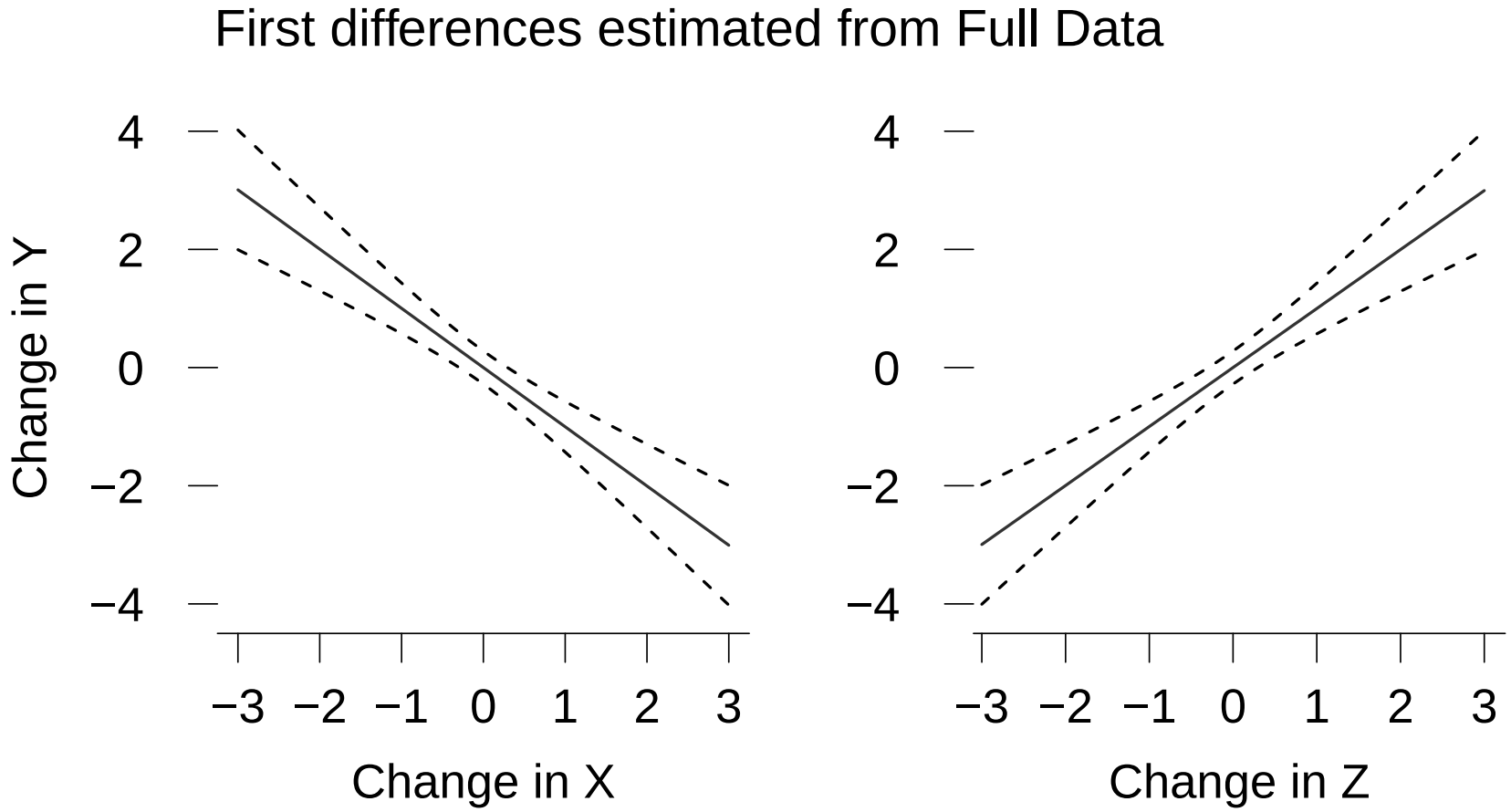
A contrived example



Above shows what we would estimate if we fully observed our 200 cases

Our goal in henceforth is to reproduce these fits and 95% CI's as closely as possible

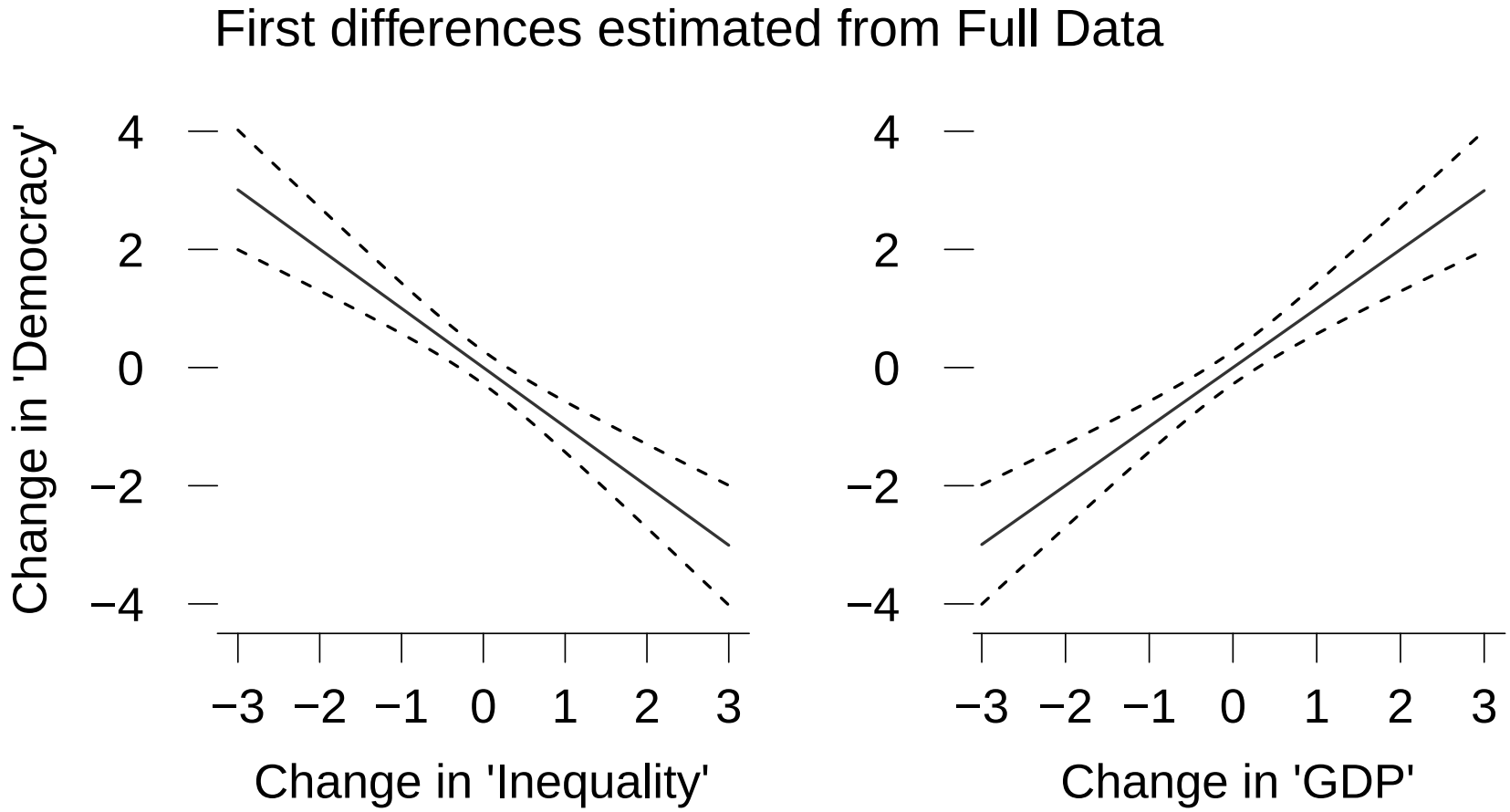
A contrived example



For all plots, I've actually averaged results after
running the whole experiment (creating a dataset, then estimating the model) 1000×

This eliminated Monte Carlo error,
and shows us what will happen on average in a case like this

A contrived example



To make the example easier to follow,
I've replaced x, y, and z with our fictive variable names

Of course, we don't have any real evidence on this hypothetical research question;
all the data are made up

Costs of listwise deletion

Our dataset contains 3 variables and 200 cases

But for about 80 of our cases, a single variable has a missing value

This means that only $80/(3 \times 200) = 13\%$ of our cells are missing

But listwise deletion will remove 40% of our cases,
increasing standard errors considerably

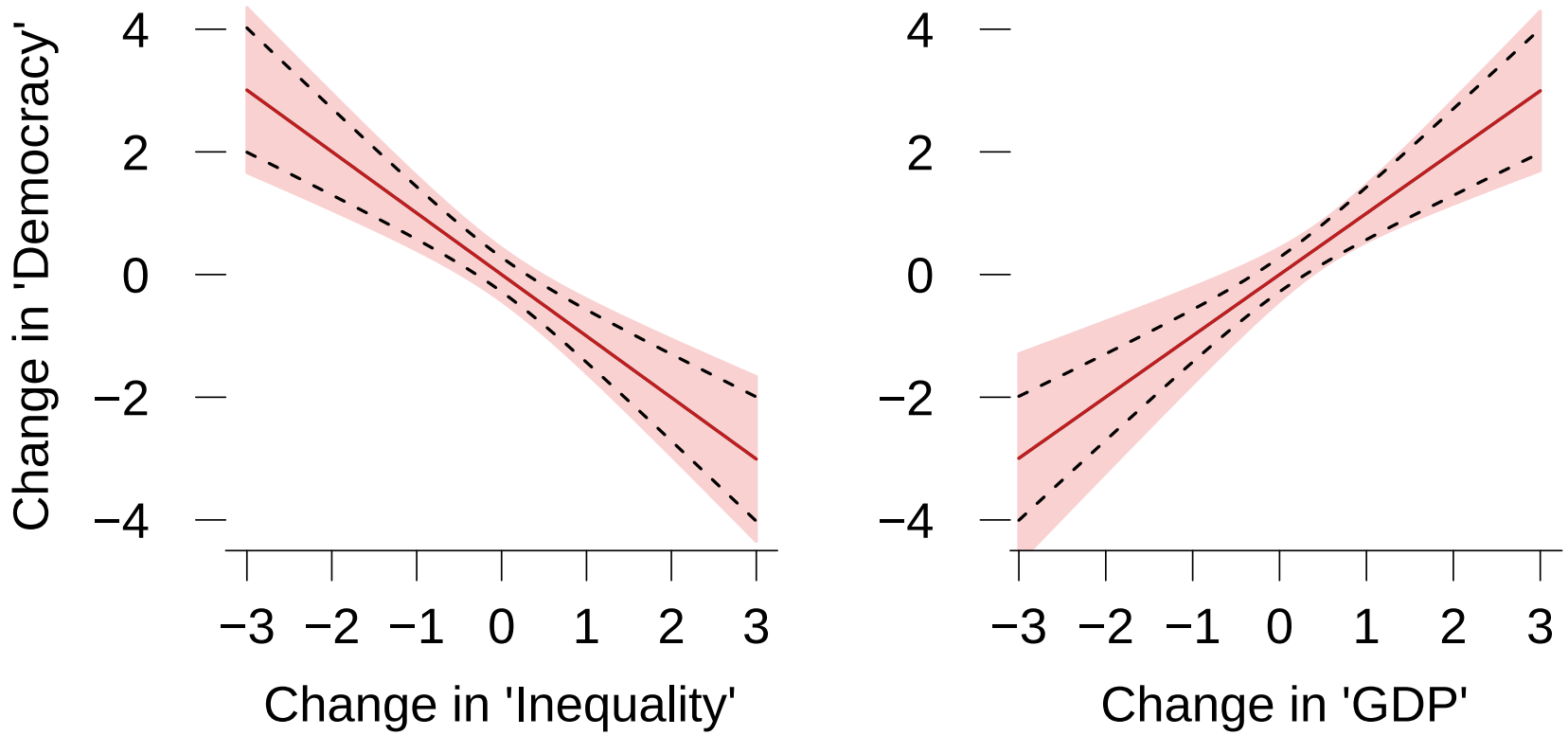
We've thrown away 160 cells containing actual data

Imagine collecting by hand 160 data points, then tossing them in the trash

But this isn't just wasted data collection effort:
it's statistically inefficient!

Listwise deletion is wasteful

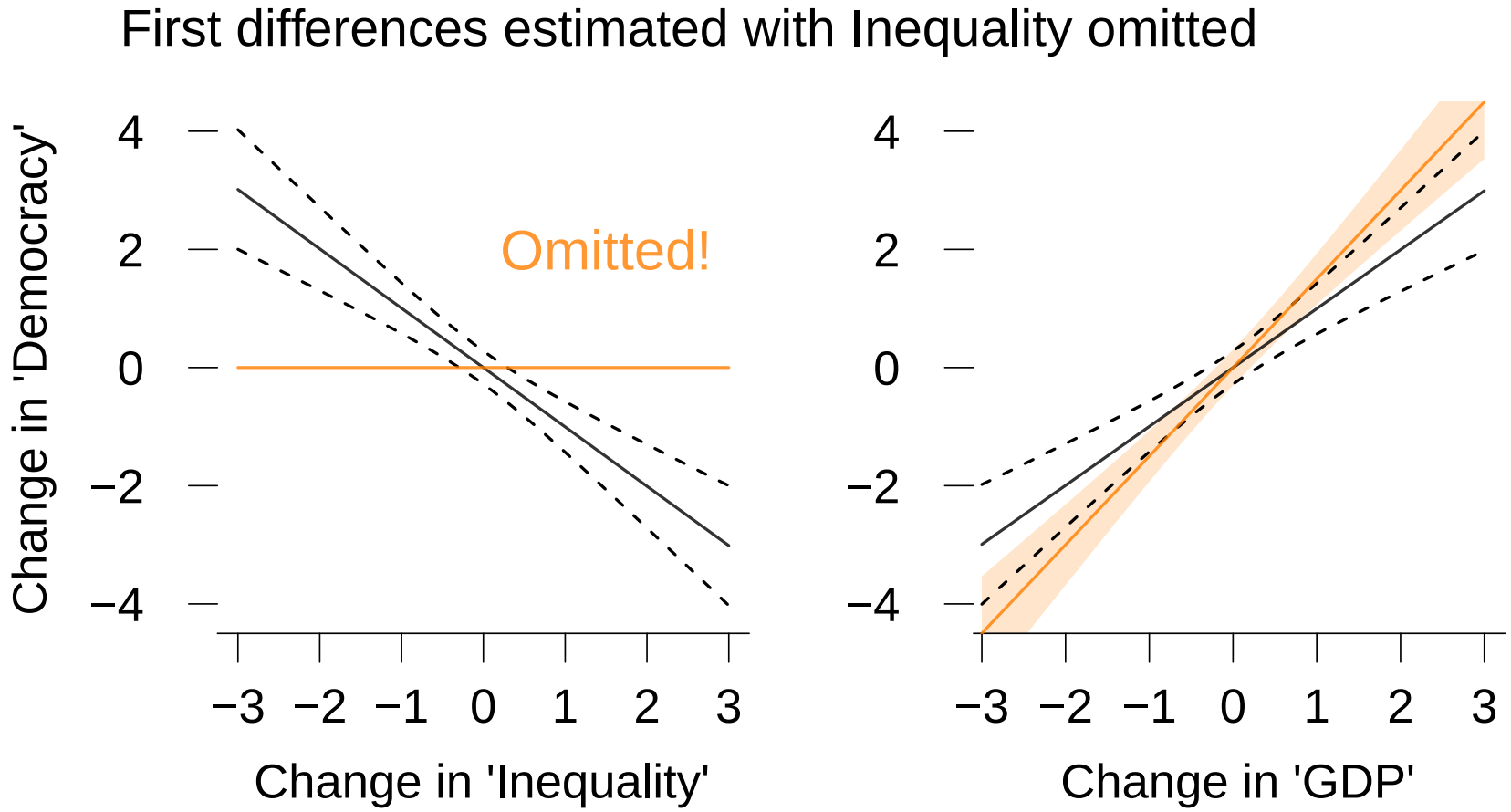
First differences estimated from Listwise Deleted Data



In our hypothetical example, listwise deletion is wasteful

Standard errors and CIs are wider than they should be,
so we might fail to detect significant relationships because of missingness

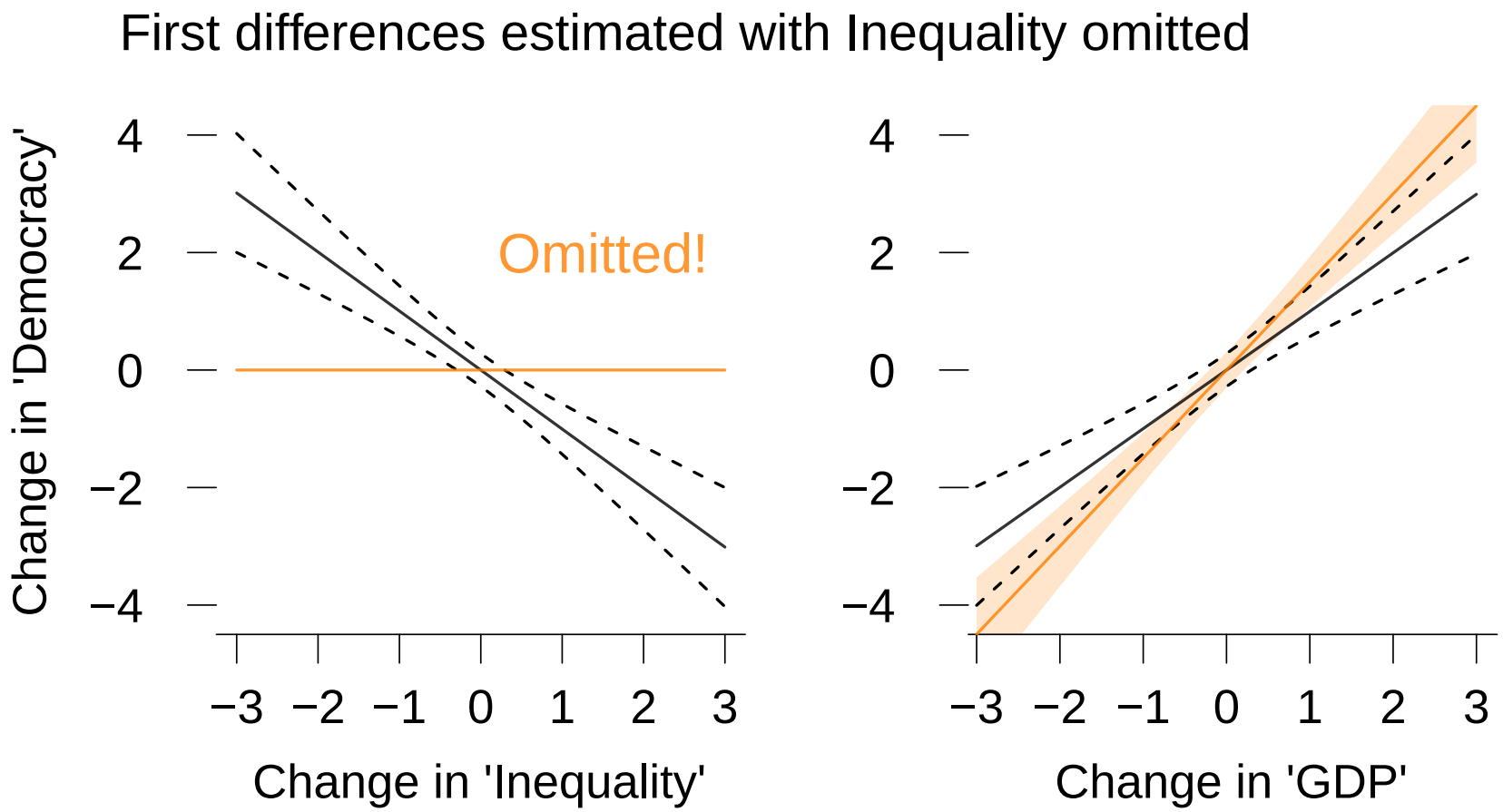
Tradeoff: listwise deletion and omitted covariates



Why did we listwise delete?

Why not drop Inequality from the model instead?

Tradeoff: listwise deletion and omitted covariates



Even if we didn't care about estimating the relationship between Inequality and GDP, we still need it in the model

Including Inequality is necessary to get unbiased estimates of the effect of GDP (which is correleted with Inequality)

Costs of listwise deletion

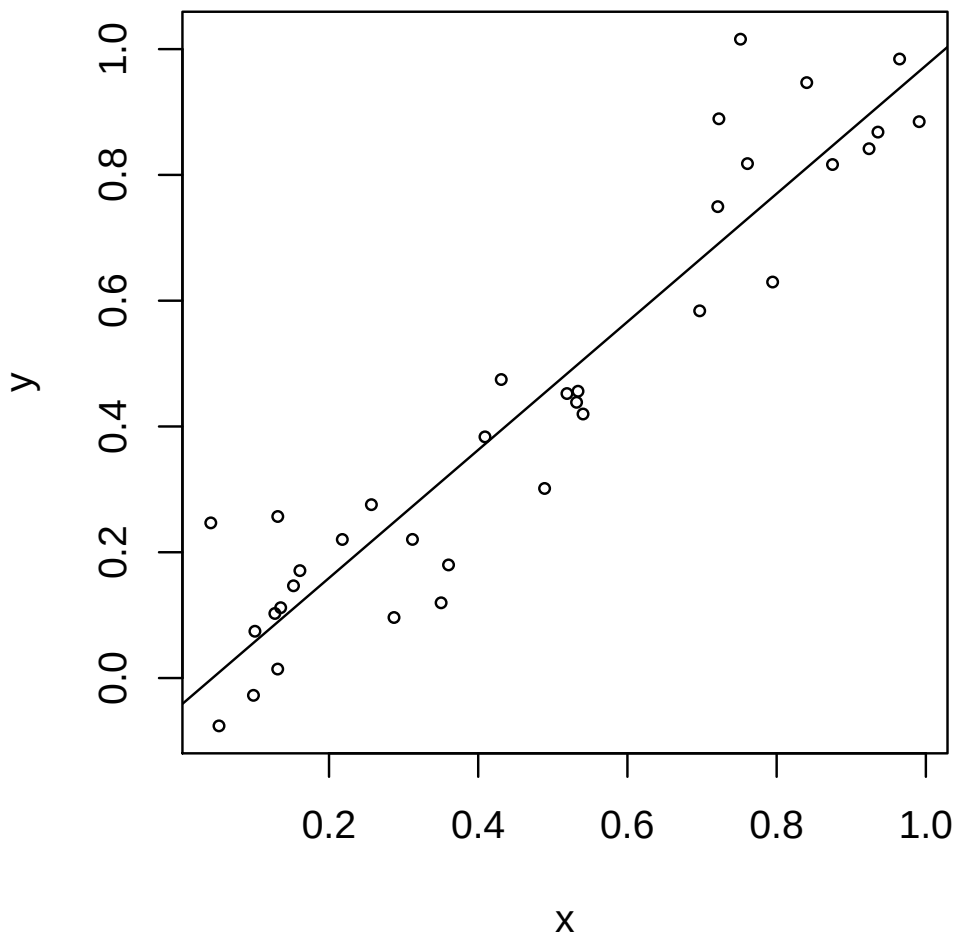
We've just traded one problem (omitted variables)
for another (listwise deletion)

Listwise deletion isn't just inefficient

In some cases it can introduce substantial bias

Recall the problem of "selection on the dependent variable"

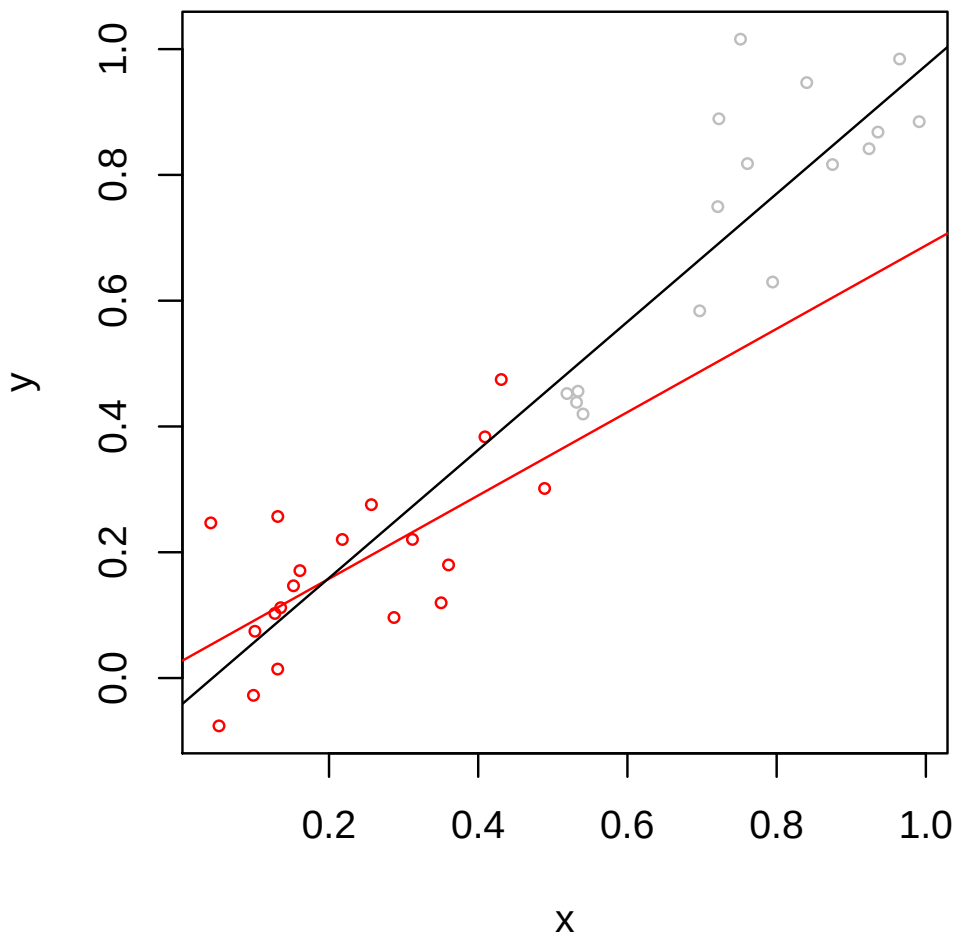
Selection bias



Suppose we conduct a survey & ask people their income (x) & conservatism (y)

With the full range of responses, we find a strong relationship

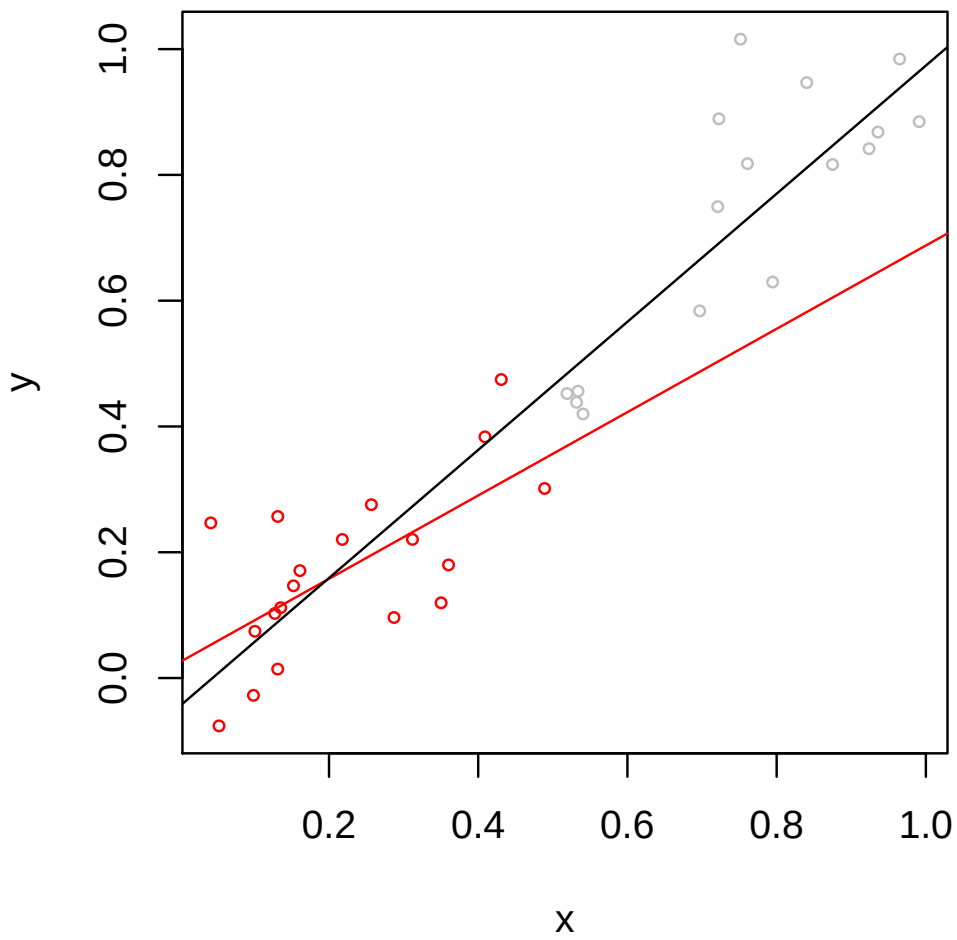
Selection bias



But suppose high income (or highly conservative) people decline to answer

Then we run a regression on the red dots only.

Selection bias



This patten of missingness biased our result biased towards 0.

This happens whether we "selected" cases intentionally,
or had them selected for us by missing data

Crude imputation methods don't help

If listwise deletion is sometimes biased and usually inefficient,
what about filling in the missing data?

If we do this right,
we can eliminate both problems of omitted variables and missing data

This approach called *imputation*, and there are obvious crude methods:

Mean imputation Filling in the missing x_i's with $\bar{x}_i$.

SIngle imputation Filling in the missing x_i's with expected values, $\mathrm{E}(x_i)$

Neither crude approach works;
in fact, both are *worse* than listwise deletion most of the time

What is mean imputation?

Democracy	Inequality	GDP
-0.195	-2.095	-0.039
0.236	0.431	0.431
-0.584	NA	-1.715
-0.310	-0.198	0.347
-3.961	NA	-0.579
1.836	NA	-0.513
...		

Above are the first six observations

Mean imputation says to replace each `NA` with the observed mean of that variable

What is mean imputation?

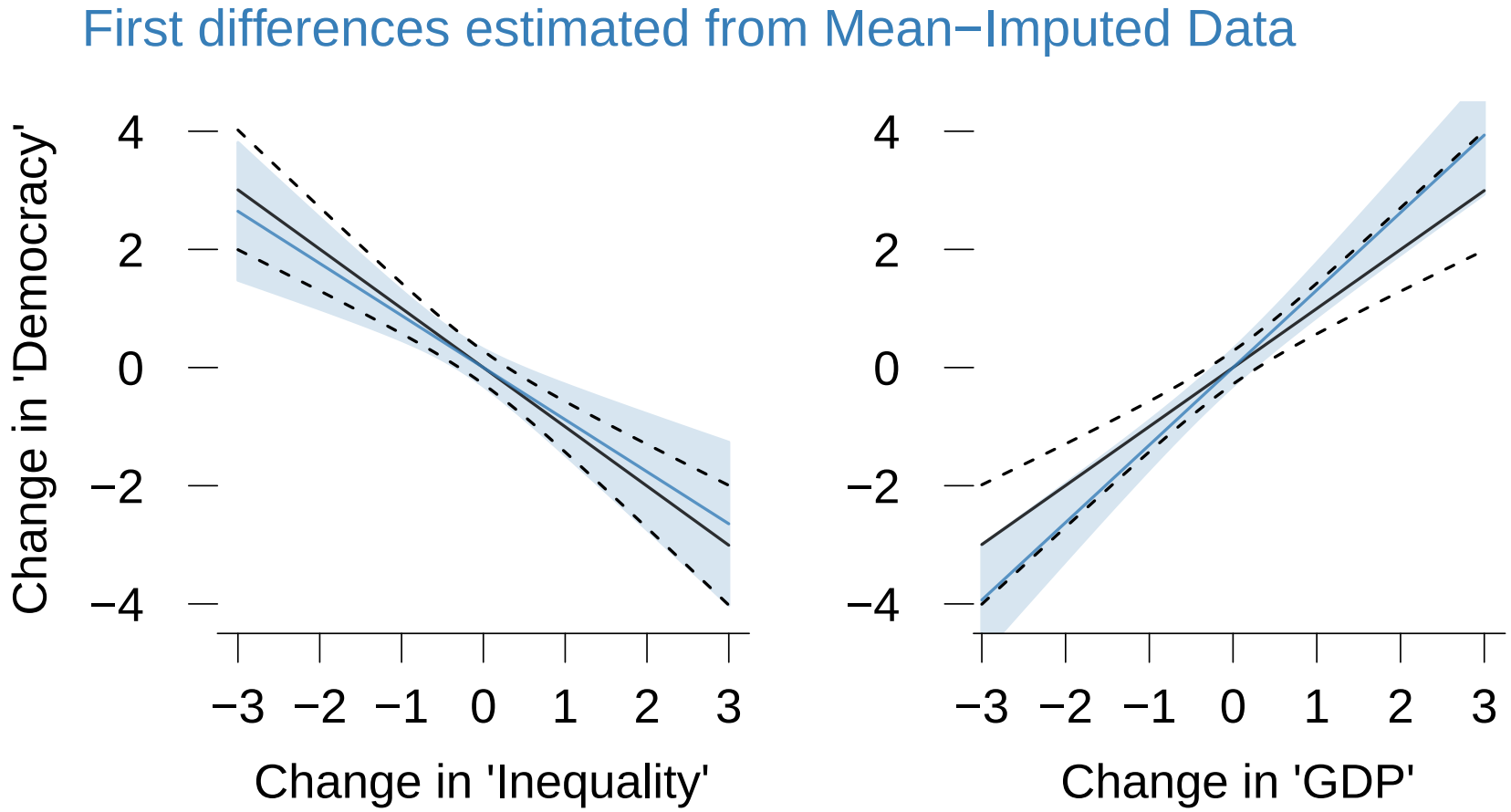
Democracy	Inequality	GDP
-0.195	-2.095	-0.039
0.236	0.431	0.431
-0.584	-0.342	-1.715
-0.310	-0.198	0.347
-3.961	-0.342	-0.579
1.836	-0.342	-0.513
...		

Above are the first six observations

Mean imputation says to replace each `NA` with the observed mean of that variable

The observed mean of Inequality is -0.342

Mean imputation doesn't work



Mean imputation biases coefficients for missing variables downwards

And biases correlated observed variables upwards!

Why mean imputation doesn't work

Several problems are evident:

1. Filling in missing data with the mean *assumes* that there is no relationship among our variables

   But our model already assumes there *is* a conditional relationship!

   For example, we to fill in the third observation, we need
   $\mathrm{E}(\mathrm{Inequality}_3|\mathrm{Democracy}_3, \mathrm{GDP}_3)$,
   not just $\mathrm{E}(\mathrm{Inequality})$

   Democracy and GDP are both low in case 3; if these are inversely correlated with Inequality, we should fill in a high value, not a low one

   Filling in the unconditional mean biases $\hat{\beta}_{\mathrm{Democracy}}$ towards zero!

2. The true mean of Inequality in the fully observed data is -0.002, not -0.342!

   The missing data has also biased our estimate of the mean of Inequality, and we've moved this bias into our imputations

What is single imputation?

Democracy	Inequality	GDP
-0.195	-2.095	-0.039
0.236	0.431	0.431
-0.584	NA	-1.715
-0.310	-0.198	0.347
-3.961	NA	-0.579
1.836	NA	-0.513
...		

Mean imputation failed because we didn't take the model into account

If our variables are correlated—and we think they are—
we need to condition on that correlation when imputing

What is single imputation?

Democracy	Inequality	GDP
-0.195	-2.095	-0.039
0.236	0.431	0.431
-0.584	NA	-1.715
-0.310	-0.198	0.347
-3.961	NA	-0.579
1.836	NA	-0.513
...		

Suppose that we fit the following model for our fully observed cases:

$$\text{Inequality}_i = \gamma_0 + \gamma_1 \text{GDP}_i + \gamma_2 \text{Democracy}_i + \nu_i$$

And then use this model to predict the missing values of Inequality j:

$$\text{E}(\text{Inequality}_j) = \hat{\gamma}_0 + \hat{\gamma}_0 \text{GDP}_j + \hat{\gamma}_2 \text{Democracy}_j$$

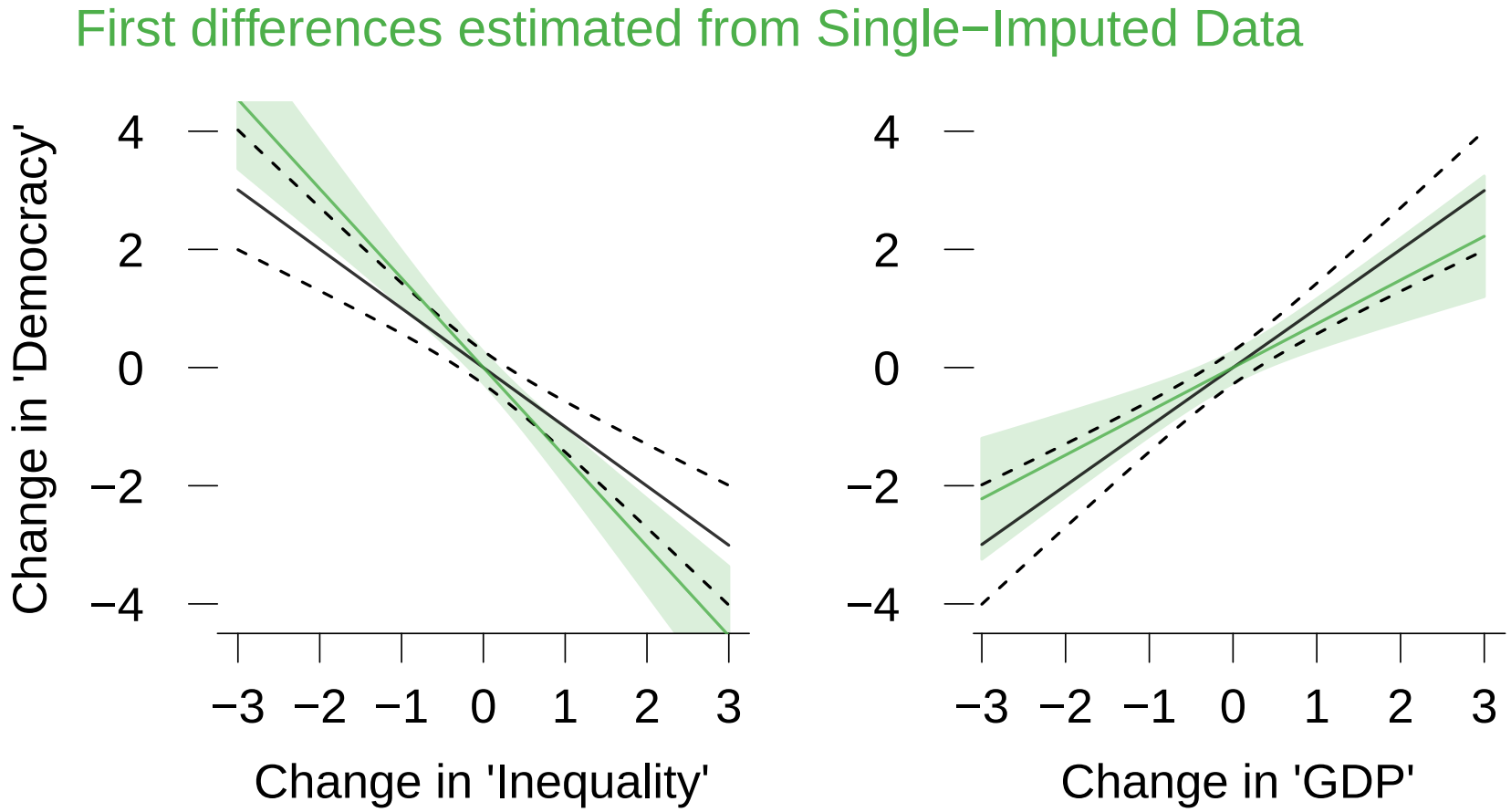
What is single imputation?

Democracy	Inequality	GDP
-0.195	-2.095	-0.039
0.236	0.431	0.431
-0.584	0.342	-1.715
-0.310	-0.198	0.347
-3.961	0.631	-0.579
1.836	-0.327	-0.513
$\ldots$		

This seems better:
the imputed Inequality values at least seem consistent with the rest of the data

But actually, what we've done is worse than before!

Why single imputation doesn't work



Single imputation biases imputed variables upwards

And biases correlated observed variables downwards!

Why single imputation doesn't work

1. We assumed any missing values are exactly equal to their expected values, with *no error*

   But randomness is fundamental to all real world variables—
   none of our other variables are precise functions of covariates

   We've assumed that the cases we didn't see are more consistent with our model than the cases we did see!

   This leads to considerable overconfidence, and biases our β's upwards!

2. And how would we implement this approach consistently across cases if different or multiple variables are missing?

3. The linear model of Inequality is still estimated using listwise deletion, so the bias from LWD passes on to our imputations!

Multiple imputation

Goal: keep all observed values in our original data, and summarize the uncertainty about our missing values implied by the observed data

Specifically, the method should

1. Impute our missing values conditional on the structure of the *full* dataset

2. Include the uncertainty in our estimation of the missings,
   as we'll never be sure we have the right imputation model

3. Includes the randomness of real world variables,
   which can't be exactly predicted even by the true model

Multiple imputation is a family of methods that achieve these goals

Generally improves on listwise deletion, except in rare cases

Here we discuss the King, Honaker et al method known as Amelia

How Amelia works

Take all of our data—dependent variables, independent variables, and even related variables not in our model—and place them in a matrix $\mathbf{D}$

Call the known elements of this matrix $\mathbf{D}_{\text{obs}}$, and the missing elements $\mathbf{D}_{\text{miss}}$

We will assume that all these variables are jointly multivariate normally distributed:

$$\text{P}(\mathbf{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{MVN}(\mathbf{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

This leads to a likelihood function for the joint distribution of the data:

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{D}) = \prod_{i=1}^{N} \mathcal{MVN}(\mathbf{d}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

where $\mathbf{d}_i$ refers to the ith observation in the dataset $\mathbf{D}$

How Amelia works

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{D}) = \prod_{i=1}^{N} \mathcal{MVN}(\mathbf{d}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

If we knew the true $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, we could use them to draw several predicted values of the missing values $\mathbf{D}_{\text{miss}}$ and fill them into several new predicted "copies" of our dataset $\tilde{\mathbf{D}}$

Each copy of the dataset would contain the known values for $\mathbf{D}_{\text{obs}}$,
but a different set of predicted draws for $\tilde{\mathbf{D}}_{\text{miss}}$

Variation across $\tilde{\mathbf{D}}_{\text{miss}}$ would summarize uncertainty about these imputations,

while the mean value of $\tilde{\mathbf{D}}_{\text{miss}}$ would capture the expected value the missing data

Amazingly, only 5 imputed datasets is usually enough to summarize uncertainty

What is multiple imputation

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{D}) = \prod_{i=1}^{N} \mathcal{MVN}(\mathbf{d}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

But we *don't* know the true $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$

If we try to estimate them from $\mathbf{D}_{\text{obs}}$ only using listwise deletion, we will have biased estimates, as in single imputation

What is multiple imputation

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{D}) = \prod_{i=1}^{N} \mathcal{MVN}(\mathbf{d}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Instead, we use a method called Expectation Maximization, or EM,
which iterates back and forth between two steps:

Expectation step Take the current estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ and predict the missing data from them

Maximization step Fill in $\mathbf{D}_{\text{miss}}$ with the current guesses of $\mathbf{D}_{\text{miss}}$, and compute new estimates of μ and $\boldsymbol{\Sigma}$

We iterate until convergence & never need to delete any rows with missing data

Naturally, there are a few extra pieces to the model
(Bayesian priors, empirical priors, etc.)

What is multiple imputation

Democracy	Inequality	GDP
-0.195	-2.095	-0.039
0.236	0.431	0.431
-0.584	NA	-1.715
-0.310	-0.198	0.347
-3.961	NA	-0.579
1.836	NA	-0.513
...		

Original Data, with missings

We end up with not one but five imputed datasets

What is multiple imputation

Democracy	Inequality	GDP
-0.195	-2.095	-0.039
0.236	0.431	0.431
-0.584	0.570	-1.715
-0.310	-0.198	0.347
-3.961	0.968	-0.579
1.836	0.701	-0.513
...		

Imputed dataset 1

We need to run all our analyses in parallel on the five datasets

What is multiple imputation

Democracy	Inequality	GDP
-0.195	-2.095	-0.039
0.236	0.431	0.431
-0.584	-0.476	-1.715
-0.310	-0.198	0.347
-3.961	0.466	-0.579
1.836	-0.437	-0.513
...		

Imputed dataset 2

Then combine the results using simulation

What is multiple imputation

Democracy	Inequality	GDP
-0.195	-2.095	-0.039
0.236	0.431	0.431
-0.584	-0.019	-1.715
-0.310	-0.198	0.347
-3.961	1.414	-0.579
1.836	-0.097	-0.513
...		

Imputed dataset 3

Specifically, just take one-fifth of your simulated $\hat{\beta}$'s from each of your five analyses, then `rbind()` them together before computing counterfactual scenarios

What is multiple imputation

Democracy	Inequality	GDP
-0.195	-2.095	-0.039
0.236	0.431	0.431
-0.584	-0.346	-1.715
-0.310	-0.198	0.347
-3.961	0.338	-0.579
1.836	0.250	-0.513
...		

Imputed dataset 4

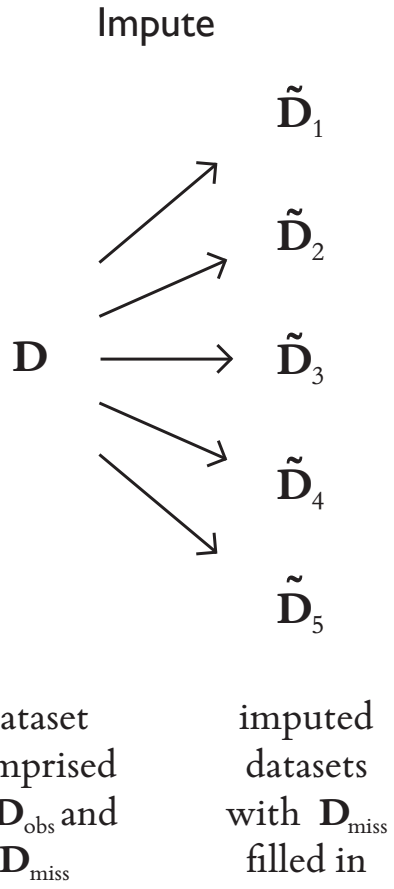
`zelig()` can automate this for you,
but only works for certain statistical models

What is multiple imputation

Democracy	Inequality	GDP
-0.195	-2.095	-0.039
0.236	0.431	0.431
-0.584	1.886	-1.715
-0.310	-0.198	0.347
-3.961	0.351	-0.579
1.836	-0.917	-0.513
...		

Imputed dataset 5

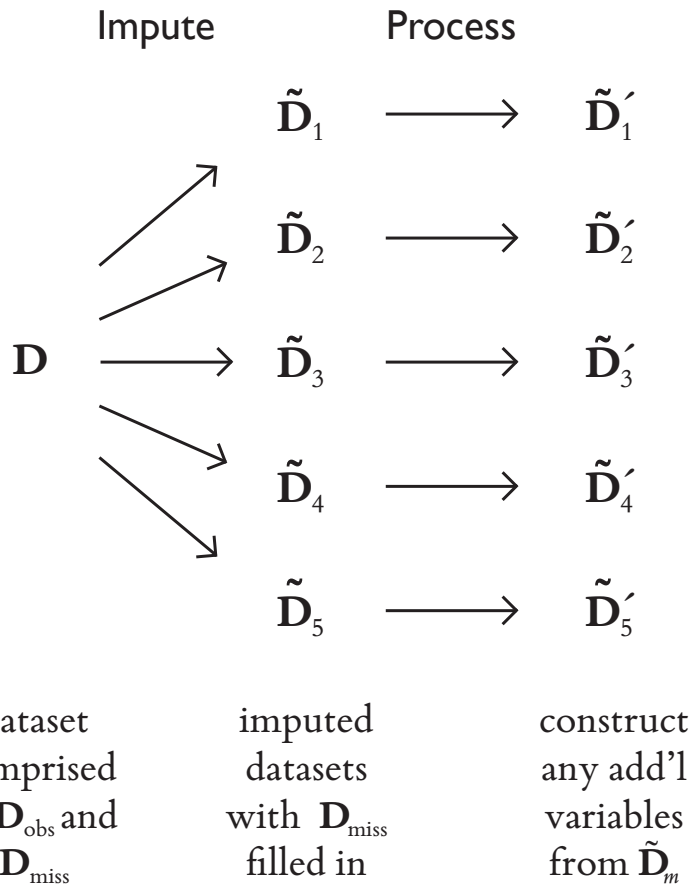
But you could also do it by hand, which is more flexible. . .



Step 1: Perform multiple imputation to create $m = 5$ or more imputation datasets

(Very time consuming, especially if run multiple times under different assumptions)

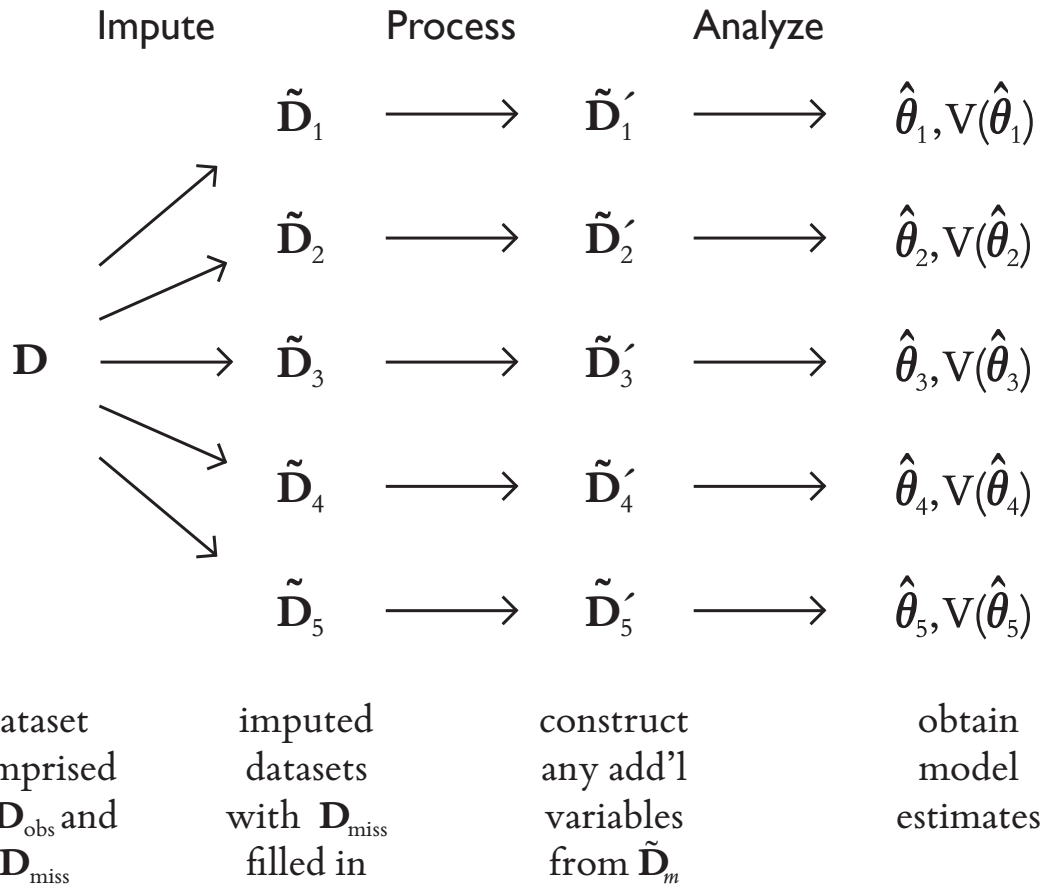
Imputing splits the analysis into M streams,
so it helps to loop over the imputed datasets for each subsequent step



Step 2: Construct additional variables from the imputed datasets

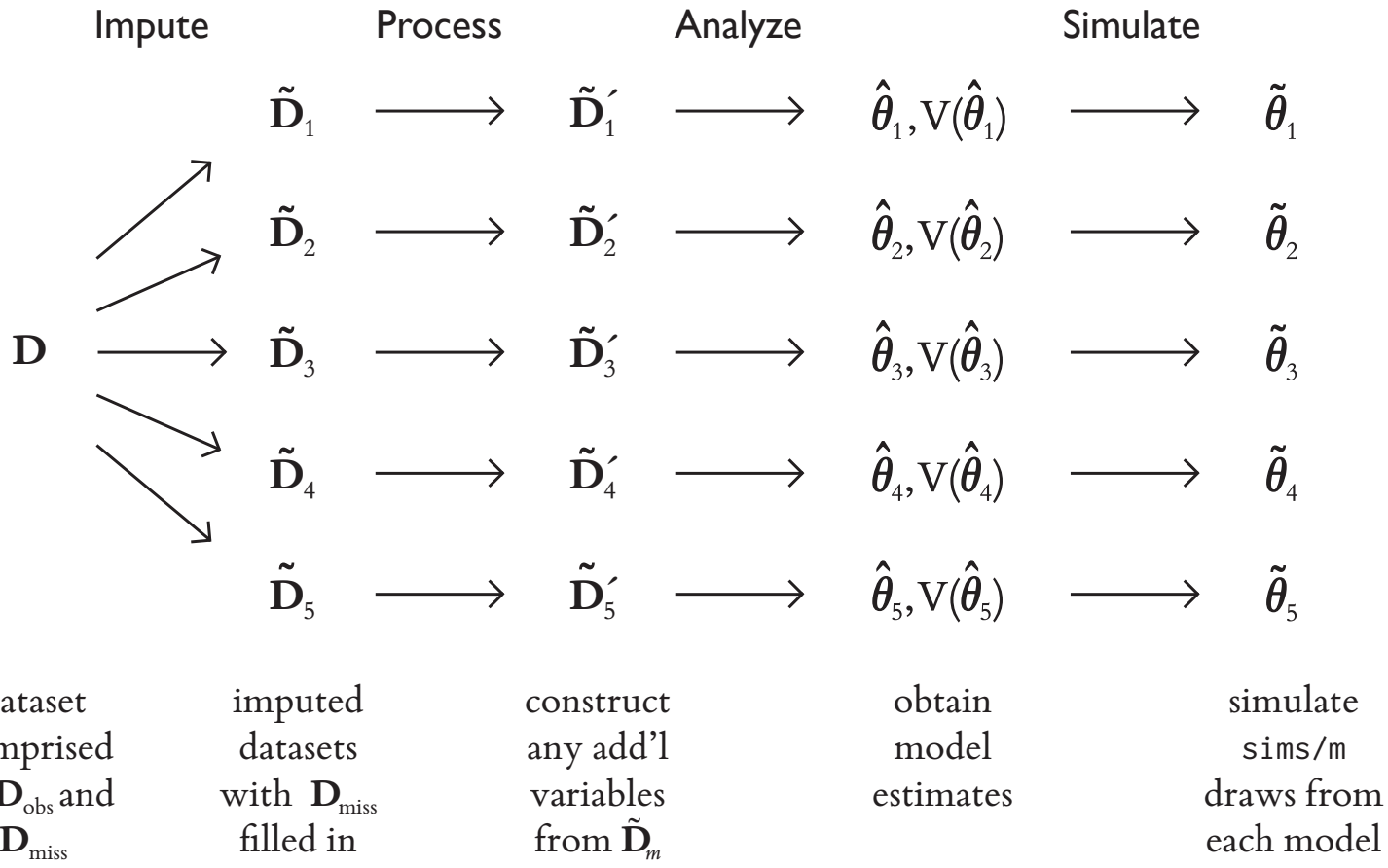
E.g., interaction terms, sums of components, or other products and sums

(e.g., if you impute GDP and popuation,
construct GDP per capita *after* all missings in either are imputed)



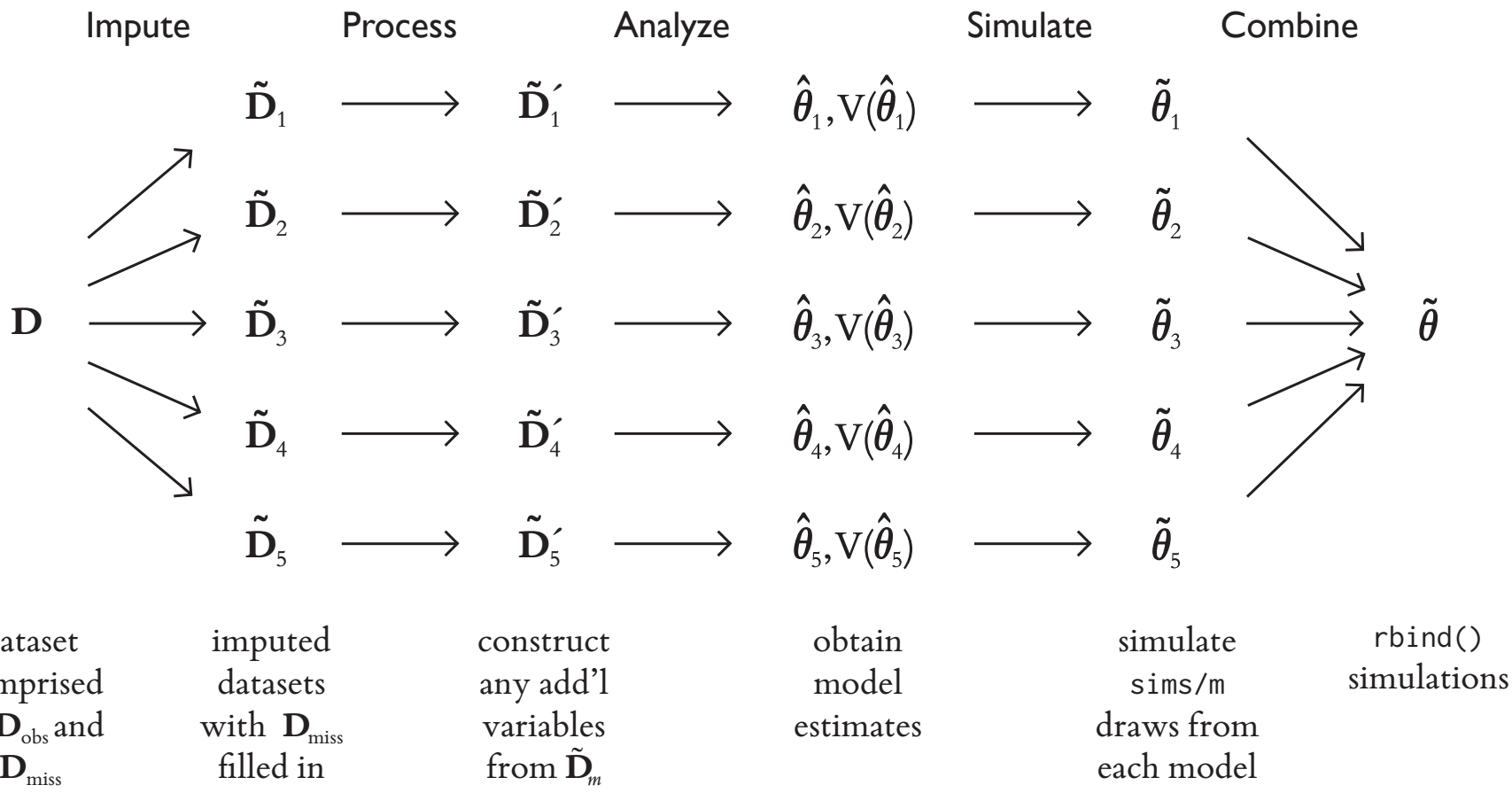
Step 3: Estimate the analysis model separately on each dataset m, and save each set of estimates $\boldsymbol{\theta}_m$ and variance-covariance matrix $\mathrm{V}(\hat{\boldsymbol{\theta}}_m)$

Each model should be the *same*, so use a loop or `lapply()`



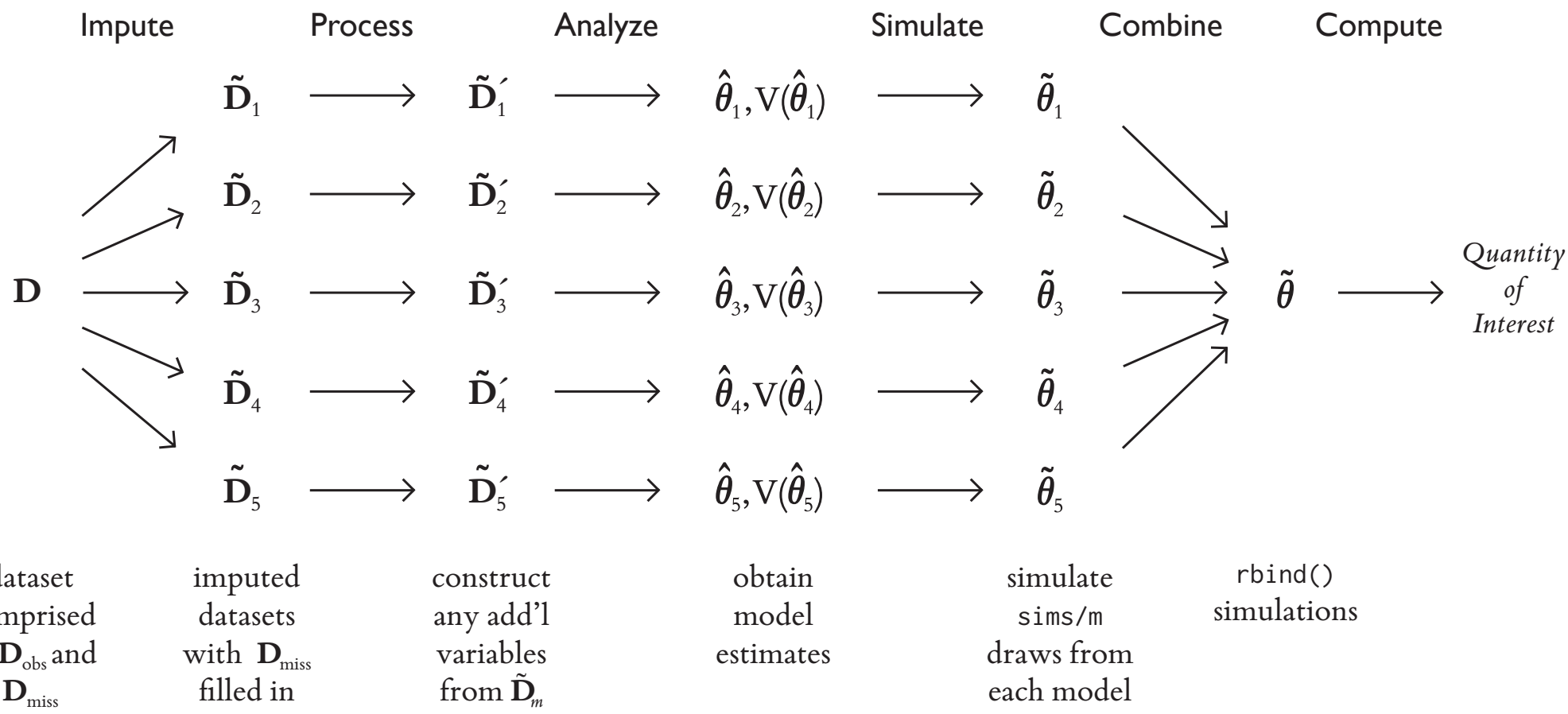
Step 4: Draw `sims/M` sets of simulated parameters from each of the M analyses

Use `mvrnorm()` as usual for this step, but in a loop over the M analysis runs



Step 5: Combine the M sets of simulated parameters into a single matrix using `rbind()`

This brings the $M = 5$ streams of the analysis back together;
after this step, we only need to do things *once* for the whole analysis

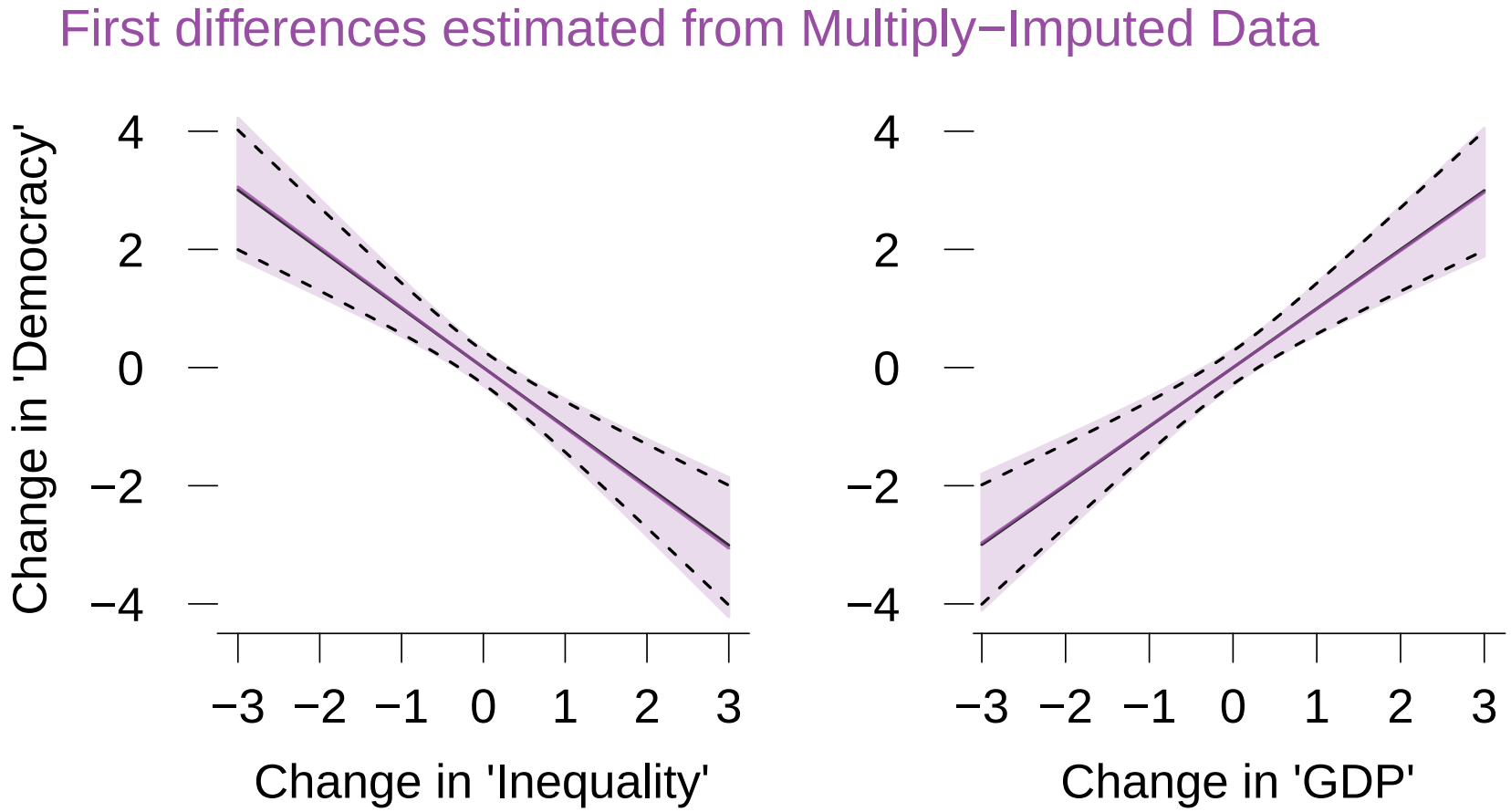


Step 6: Produce counterfactual scenarios and graphics as usual

The code for this step can be exactly the same as for a non-imputation analysis

You may wish to average the $M = 5$ datasets at this stage for computing factual and counterfactual values of the covariates

Multiple imputation usually outperforms listwise deletion



Success! We have closely matched the original full data results

We've gotten more information & precision out of our data than with LWD, and not added any bias despite imputing

Multiple imputation usually outperforms listwise deletion

Under fairly broad conditions, multiple imputation will outperform or match the performance of listwise deletion

So it's good to make it a standard part of your statistical toolkit

But, like any method, you need to be careful

Multiple imputation by Amelia usually works well if you. . .

- transform continous variables to be as close to Normal as possible,
e.g., through log, logit, or quadratic transformations

- tell your imputation model which variables are binary, ordered, or nominal

- include as many well-observed variables related to your partially observed variables as you can find.

The variables you use in the imputation model can be a much larger set of variables than in your analysis model.

- remember *every* variable in the analysis model *must* also be in the imputation model

- check available diagnostics to make sure imputation worked.

For example, you could "overimpute": use the imputation model to predict the known data one observation at a time, and check that it fits

Multiple imputation tends to fail in two cases

1. If you try to impute a dataset that has a very high percentage of missing values, or some variables which are almost never observed

   You may need to give up on some variables and exclude them from your study in this case

2. If you try to impute variables which are more likely to be missing in ways that could only be understood if we observed those missing values

   Example: Suppose people who are higher income are more likely to refuse to answer questions about their income, *and* nothing in the rest of our observed covariates fully explains this

   (Called "non-ignorable missingness")

   Then *any* imputation method will produce biased estimates of the missing values of income

   But listwise deletion in this case is likely to be just as biased, with rare exceptions

When to use an alternative to Amelia. . .

Even if multiple imputation is a good idea for your data,
Amelia is not necessarily the best option

1. Custom-build imputation is often superior

   If you can take advantage of specific features of your data to build an imputation model, it will likely outperform generic methods

2. Better options exist for data with lots of missing categorical variables

   In this case, consider `mice`, or Multiple Imputation by Chained Equations

I leave these options for self-study

Multiple imputation for cross-sectional data

In R, the amelia() function in the Amelia package does multiple imputation for cross-sectional, time series, and TSCS data

For cross-sectional data, it's usually very easy to make your imputed datasets:

```
library(Amelia)


# Run Amelia and save imputed data, and number of imputed datasets
amelia.res <- amelia(observedData)
MiData <- amelia.res$imputations
nimp <- length(MiData)
```

MiData is a list object with 5 elements, each of which is a complete dataset

Then you can run your analysis 5 times in a loop, saving each result to the list object:

```
# Run least squares on each imputed dataset,
# and save results in a list vector
mi <- vector("list",nimp)
for (i in 1:nimp) {
  mi[[i]] <- lm(y ~ x + z, data=MiData[[i]])
}
```

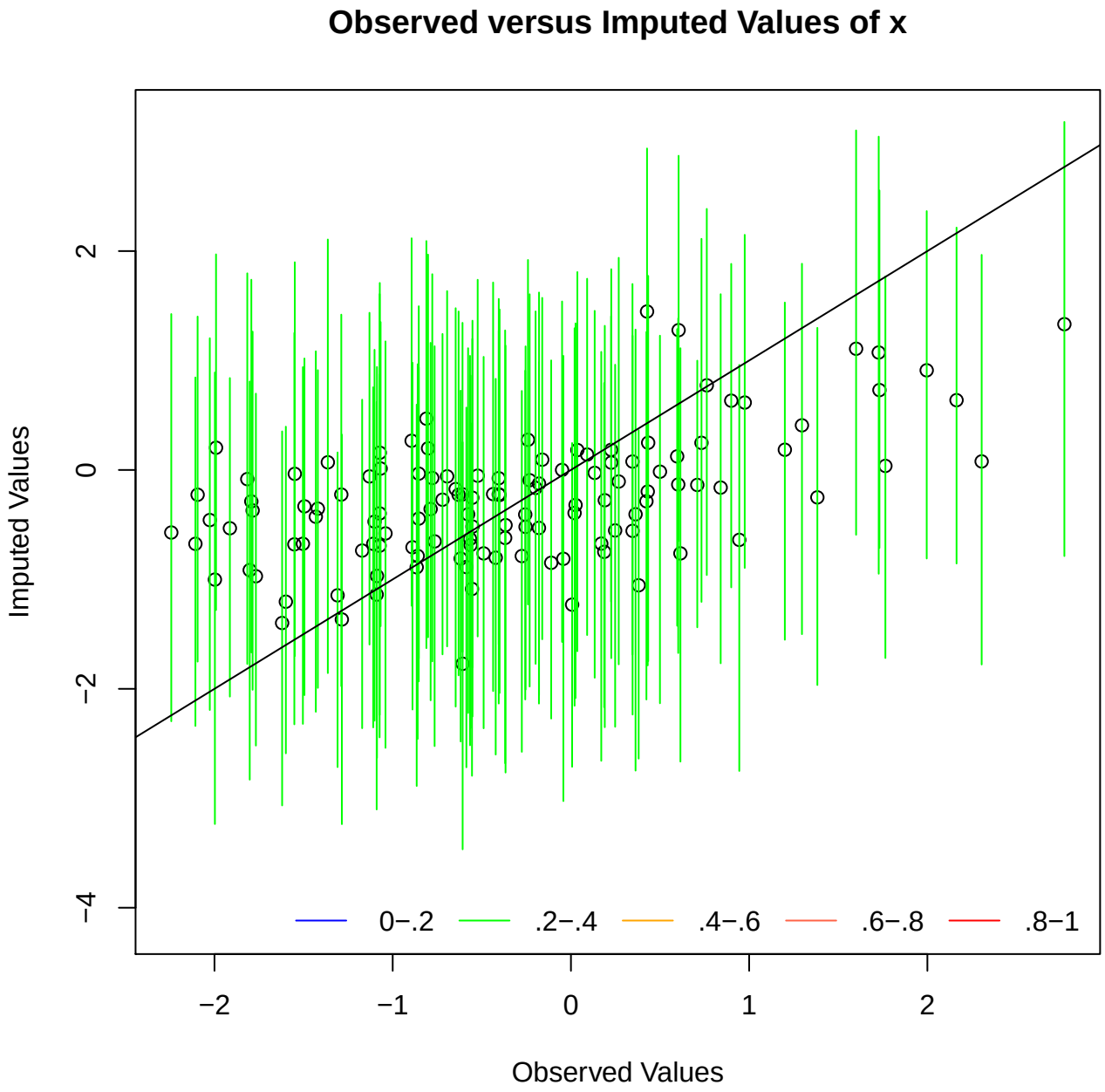
Multiple imputation for cross-sectional data

Finally, you need to combine the results by drawing one-fifth of your simulated β's from each model, like so:

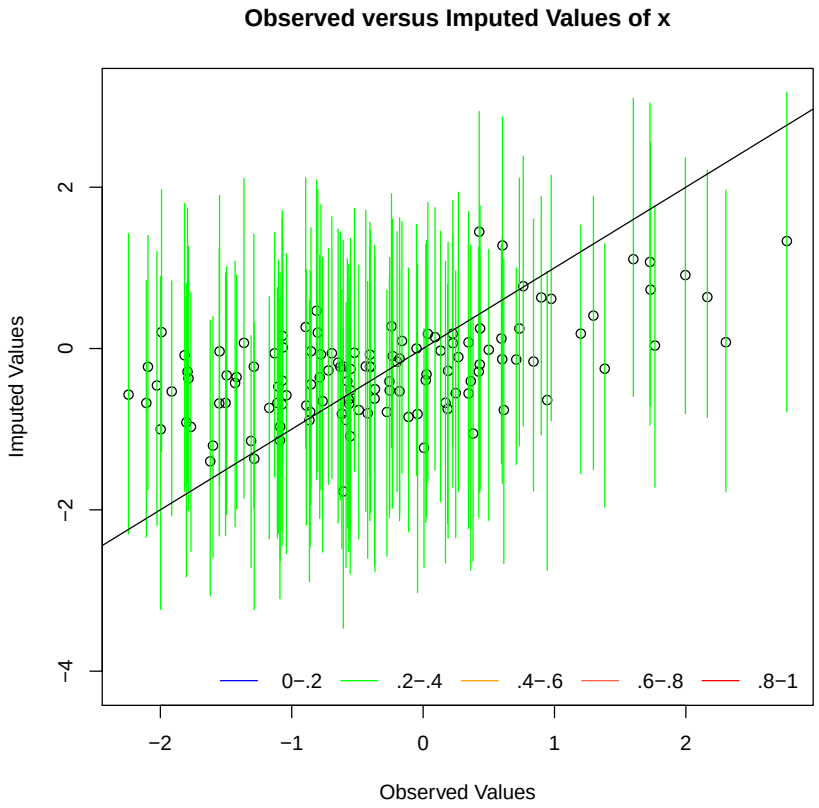
```
# Draw 1/nimp of the beta simulations from each run of least squares
sims <- 10000
simbetas <- NULL
for (i in 1:nimp) {
  simbetas <- rbind(simbetas,
                    mvrnorm(sims/nimp, coef(mi[[i]]), vcov(mi[[i]]))
                    )
}
```

From this point, you can simulate counterfactuals as normal

Of course, you could have `zelig()` automate all of this, as Zelig knows what to do with Amelia objects



Overimputation diagnostic: 90% of colored lines should cross the black line




```
pdf("overimputeX.pdf")
overimpute(amelia.res, var="x")
dev.off()
```

Multiple imputation for TSCS data

Until recently, off-the-shelf multiple imputation failed for TSCS data

For example, TSCS data are not iid, so we can't "just" treat them as MVN

King and Honaker (2010) propose a straightforward solution

Add to the EM imputation model several tricks to flexibly model time series and panel structures:

1. Allow observations from the same unit to be a smooth function of time.

2. Allow contemporaneous observations to be more or less correlated with other cross-sectional units

Multiple imputation for TSCS data

To use Amelia with panel data, you need to set a few more options, like this:

```
data.mi <- amelia(data, idvars=2, ts=3, cs=1,
                  polytime=3,intercs=FALSE,nom=56,
                  bounds=bounds,
                  log=c(4:38,40:55))
```

In this example, taken from my work, I identify for Amelia the column containing the id of each observation (2), the column with the period (3), and the country name (1)

Multiple imputation for TSCS data

To use Amelia with panel data, you need to set a few more options, like this:

```
data.mi <- amelia(data, idvars=2, ts=3, cs=1,
                  polytime=3,intercs=FALSE,nom=56,
                  bounds=bounds,
                  log=c(4:38,40:55))
```

I request time to be smoothed using a cubic polynomial (`polytime=3`)

An alternative would be to ask for a spline smoother with k knots by setting `splinetime=k`

Splines are faster, especially if k is high

Multiple imputation for TSCS data

To use Amelia with panel data, you need to set a few more options, like this:

```
data.mi <- amelia(data, idvars=2, ts=3, cs=1,
                  polytime=3,intercs=FALSE,nom=56,
                  bounds=bounds,
                  log=c(4:38,40:55))
```

Setting `intercs=TRUE` would allow each country to have a different smoother over time; I turned this off to speed up estimation

Multiple imputation for TSCS data

To use Amelia with panel data, you need to set a few more options, like this:

```
data.mi <- amelia(data, idvars=2, ts=3, cs=1,
                  polytime=3,intercs=FALSE,nom=56,
                  bounds=bounds,
                  log=c(4:38,40:55))
```

I had a nominal variable in column 56, and needed to log the variables in columns 4 to 38 and 40 to 55 to make them more approximately Normal

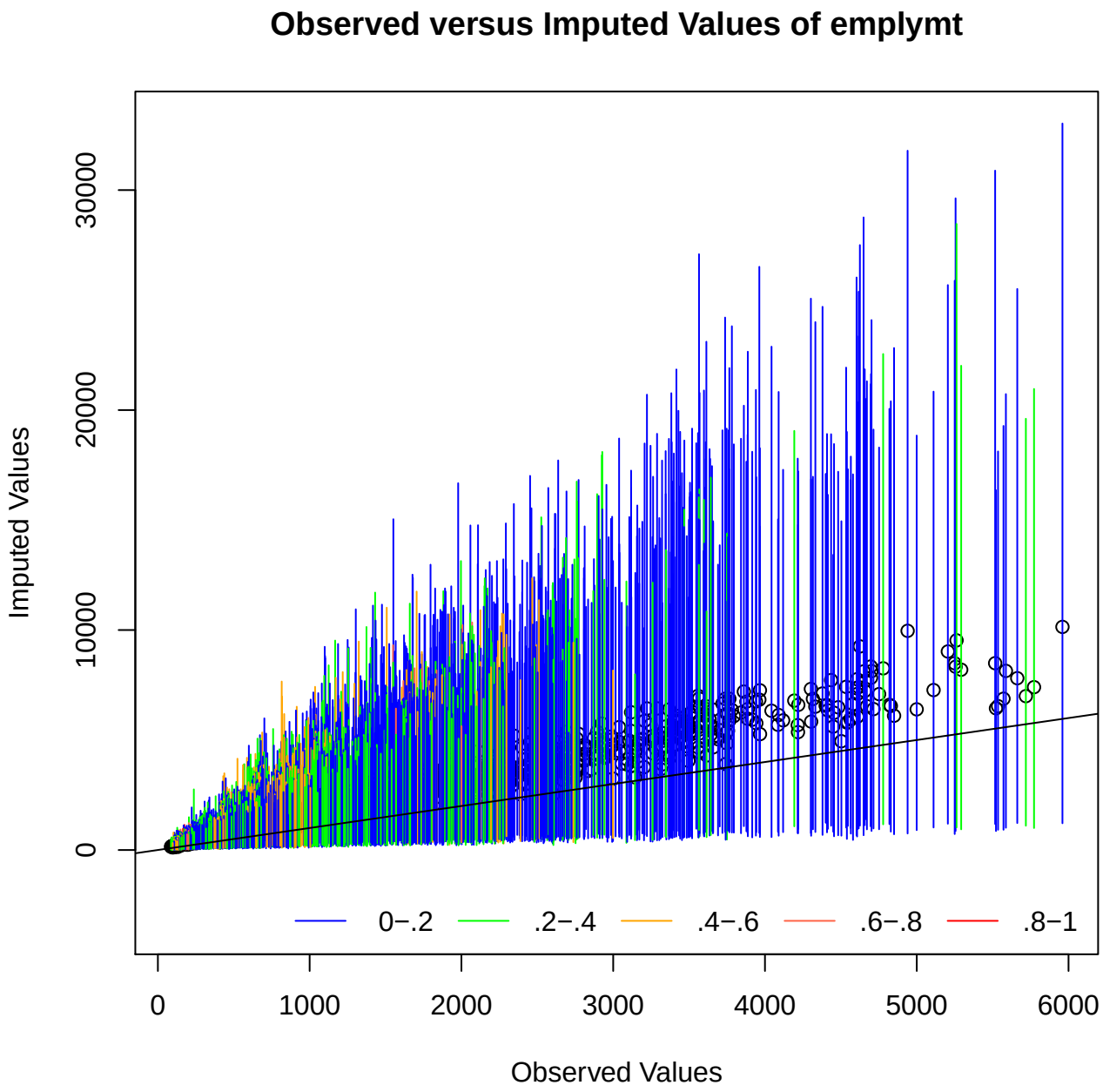
Finally, a number of my variables were bounded (e.g., from 0 to 1); setting the bounds argument to identify these bounds by variable allows Amelia handle them appropriately

Multiple imputation for TSCS data

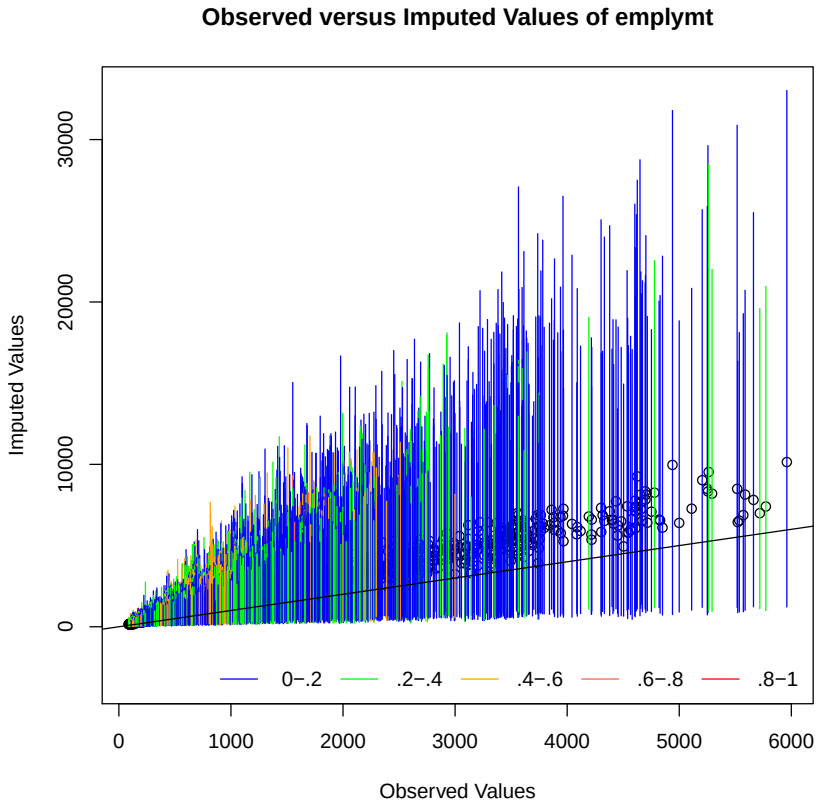
To use Amelia with panel data, you need to set a few more options, like this:

```
data.mi <- amelia(data, idvars=2, ts=3, cs=1,
                  polytime=3,intercs=FALSE,nom=56,
                  bounds=bounds,
                  log=c(4:38,40:55))
```

Other inputs I didn't use (but you might) include options for including `lags` or `leads` of variables in the imputation model, for requesting square root or logit transformations, and for adjusting the priors to help estimation in difficult cases

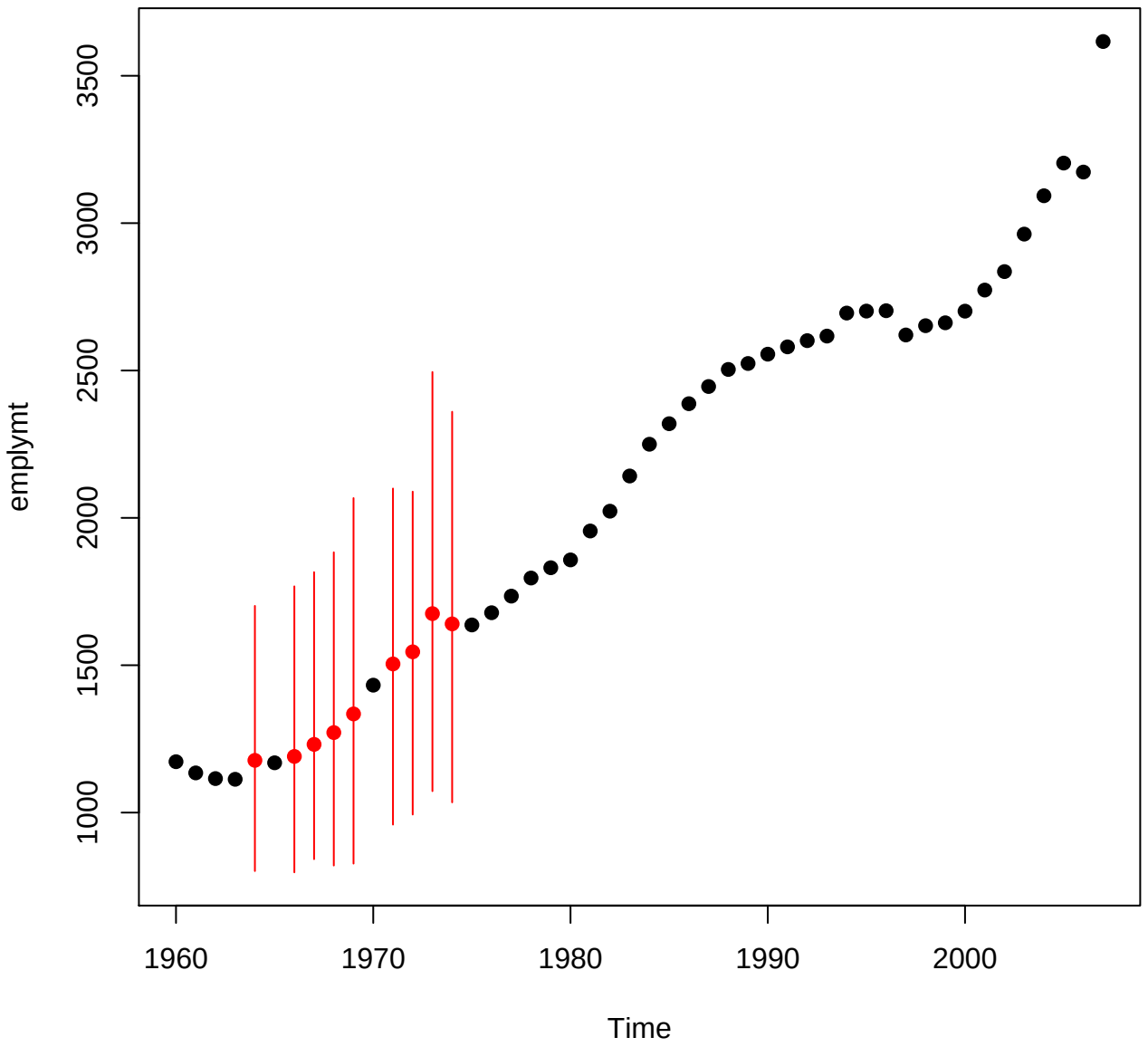


Overimputation diagnostic: 90% of colored lines should cross the black line




```
# Loop over all data variables
for (i in 4:ncol(data)) {

  # Make the overimputation plot for current variable
  pdf(paste("overimp_",names(data[i]),".pdf",sep=""))
  overimpute(data.mi, var=i)
  dev.off()
}
```



Time series diagnostic: Check for a plausible, usually smooth pattern

```
# Get the province codes & number of provinces
allprov <- unique(data$provcode)
nprov <- length(allprov)

# Loop over all data columns
for (i in 4:ncol(data)) {

  # Loop over all provinces
  for (j in 1:nprov) {

    # Make each province's time series plot, with imputations added
    pdf(paste("tscs_",names(data[i]),"_",allprov[j],".pdf",sep=""))
    tscsPlot(data.mi, var=i, cs=allprov[j])
    dev.off()
  }
}
```