POLSCI 9592

Lecture 10: Missing Data and Multiple Imputation

Dave Armstrong

Goals for This Session

1. Why are missing data problematic?
2. What methods can we use to deal with missing data and what are their implications?
3. How do we know if our imputation "worked"?
4. Example

Why Care about Missing Data?

- Can cause bias, if missingness is systematic.
- Can reduce sample size to a point where reliable inferences are difficult to make, even if missingness is not systematic.
- Systematic missingness can truncate the sample calling into question the generalizability of results.

Consequences for Data Analysis

The effects on data analysis are the ones most commonly acknowledged by the literature on missing data.

- Missing data, at a minimum, can pose problems for statistical power (i.e., sample size).
- Statistical procedures also make assumptions about distributions (i.e., error distributions). Missing data can make some of these assumptions less likely to hold, especially if the missingness is not random.
- Missing data can also, as previously stated, reduce reliability, which reduces effect size, which in turn reduces statistical power.

Consequences for Validity

Internal validity can be defined as - the extent to which a researcher can reasonably claim that a particular factor, usually an intervention of some sort, is responsible for the observed outcome. Confounders and alternative explanations are threats to internal validity. *Selection bias* is an example:

- There can be systematic differences between responders and non-responders in experiments/surveys. These differences could be responsible for the outcome rather than the variable of interest.

Missing data can lead to a more homogeneous sample that is not representative of the population which can cause all sorts of problems for generalizability.

- The same can be true of missingness on particular variables.

Rubin's Classification Scheme

Rubin suggested three types of missing data - missing completely at random (MCAR), missing at random (MAR) and missing not at random (MNAR). This classification scheme is based on:

- The variables with the missing data,
- Associated variables (i.e., covariates), and
- a hypothetical mechanism underlying the missingness.

Background on Rubin's Classification

- Refer to $\mathbf{R}$ as the matrix of dummy variables that mirrors the data matrix where 1 indicates missing and 0 indicates non-missing.
- Refer to $\mathbf{Y}$ as the data matrix - the matrix of variables for all observations. Where $\mathbf{Y}_{obs}$ refers to the observed (i.e., non-missing) values and $\mathbf{Y}_{miss}$ as the missing values.
- ϕ is the relationship of the observed variable matrices $\mathbf{Y}_{obs}$ and $\mathbf{Y}_{miss}$ to the dummy variable matrix $\mathbf{R}$. ϕ is probabilistic (i.e., theoretical) here because we don't know the values of $\mathbf{Y}_{miss}$ and thus can never calculate ϕ.
- ϕ is the operative piece of information in Rubin's classification scheme.

Classification of Missing Data Mechanisms

- If $\phi = \mathbf{0}$ (i.e., there is no relationship between $\mathbf{R}$ and $\mathbf{Y}_{obs}$, and no relationship between $\mathbf{R}$ and $\mathbf{Y}_{miss}$), then the data are MCAR. Here, randomness is the mechanism that generated the missing data.
- Data are MAR if there is a relationship between $\mathbf{R}$ and $\mathbf{Y}_{obs}$, but there is no relationship between $\mathbf{R}$ and $\mathbf{Y}_{miss}$.
- Data are NMAR if there is a relationship between $\mathbf{R}$ and $\mathbf{Y}_{miss}$. The relationship between $\mathbf{R}$ and $\mathbf{Y}_{obs}$ is irrelevant here - it may or may not exist. Note, that this is an impossible distinction for us to make with data.

"Ignorability" is a property of MAR and MCAR data. Here, the mechanism is ignorable if we can reasonably recover that information from other observable data.

Dealing with Missing Data

Two ways of dealing with missing data:

- Listwise Deletion
- Imputation
  - Mean imputation
  - Regression Imputation
  - Hot-decking
  - Multiple Imputation

Listwise Deletion

Listwise deletion involves deleting all observations that have at least one missing data point. Conventional view is:

- In the best-case scenario, listwise deletion causes inefficiency.
- In the worst-case scenario, listwise deletion causes bias and inefficiency.

Often times, omitting variables with missing data can be preferable to listwise deletion in MSE terms (though other methods talked about later are better)

Mean Imputation

Both mean and regression imputation are trying to impute a "best guess" for missing data. Mean imputation imputes the unconditional mean of the variable for every missing observation.

- Mean imputation reduces variability in the offending X variable.
- As a result, coefficient estimates will be biased (generally toward 0).
- The variance of the coefficients will also be underestimated.

Regression Imputation

In regression imputation, the complete cases are used to estimate a regression model and predictions from that model are used as imputations.

That is, we are imputing each observation with it's *conditional mean*.

- Impractical/impossible with complicated patterns of missing data.
- Does a reasonably good job of recovering unbiased estimates of parameters in a wide range of situations.
- Drastically underestimates variability in the parameters leading to overconfidence.

Multiple Imputation

MI has a similar flavor to regression imputation, but has more appealing properties. Multiple imputation proceeds as follows:

- Fill in starting values for all observations (random, mean, etc...)
- Predict X_1 using all other variables in estimation model (including y) and fill in the missing observations on X_1 with a draw from the posterior (i.e., sampling distribution) of $\hat{X}_1$.
- Using the previously predicted values for X_1, move to X_2 and predict its values using all other variables. Fill in the missing values on X_2 with a draw from the posterior of $\hat{X}_2$.
- Move through all the variables similarly and then start over again using the conditionally imputed values from before as starting values.
- Repeat until convergence.

More MI

- With multiple imputation, we impute $m \geq 2$ and usually between 5 and 50 complete datasets.
- We combine the estimates using the following set of equations:

$$\bar{Q} = \frac{1}{m} \sum_{t=1}^{m} \hat{Q}^{(t)}$$

$$\bar{U} = \frac{1}{m} \sum_{t=1}^{m} U^{(t)}$$

$$B = \frac{1}{m-1} \sum_{t=1}^{m} \left(\hat{Q}^{(t)} - \bar{Q} \right) \left(\hat{Q}^{(t)} - \bar{Q} \right)'$$

$$T = \bar{U} + \left(1 + \frac{1}{m} \right) B$$

Sampling Distribution of Q-bar

The distribution of the test statistic is harder than you might think to derive. If we assume that proportion of missing information for every variable is the same, then, we can say:

$$\tilde{T} = (1 - r_1)\,\bar{U}$$

$$r_1 = \left(1 + \frac{1}{m}\right)\frac{tr\left(B\bar{U}^{-1}\right)}{k}$$

Sampling Distribution of Q-bar (2)

The test-statistic can be calculated against the null Q_0 as:

$$D_1 = \frac{\left(\bar{Q} - Q_0\right)' \tilde{T}^{-1} \left(\bar{Q} - Q_0\right)}{k}$$

with a p-value of $P\left(F_{k,\nu_1} \geq D_1\right)$ where:

$$\nu_1 = 4 + (t-4)\left[1 + \left(1 - t2^{-1}\right) r_1^{-1}\right]^2$$

Development of MI Algorithms

- Initially, MI software used a multivariate normal approximation to impute the missing values.
  - This has been shown to work relatively well even if normality is the wrong theoretical model (e.g., when trying to impute dummy variables).
  - If using software that does this, you should transform variables to be theoretically unbounded and recode ordinal variables to include approximately cardinal information.
  - Gary King and others' Amelia II uses this approximation.
- Other methods (e.g., MICE) respect the level of measurement of the missing variables and use different regression techniques to impute the values.
  - Transformations are still appropriate here because the underlying model for continuous data is still linear.
  - No need to recode ordinal data.
  - `mice` in R uses this technique.

Some other Advice

- All variables (including y) that are in your analysis model should be in the imputation model (otherwise, potential bias results).
  - In fact, Pepinsky as well as Arel-Bundock and Pelc suggest that imputation helps most when y has some missingness.
  - Extra variables can be included in the imputation model if they are relevant but recent results show this doesn't increase efficiency as much as we might expect.
  - Also needs to include non-linear trends if they exist (e.g., polynomials).
- The conventional advice is that somewhere in the neighborhood of 5-10 complete datasets are sufficient to generate the result.
  - You might be better off with somewhere around 100.

New Work on LWD

Arel-Bundock and Pelc (2018, *PA*) argue that:

- LWD does not cause bias
  - if data are MCAR **or**
  - if missingness is unrelated to the DV **or**
  - the missingness is related to observable covariates.
- MI only reduces bias when
  - data are MAR **and**
  - the MI model assumptions hold **and**
  - the missingness is on the DV.
- In other cases, data are NI and neither LWD nor MI are guaranteed to reduce bias.

Pepinsky

Pepinsky (2018, *PA*) shows:

Missingness	Listwise Deletion	Multiple Imputation
MCAR	Unbiased	Unbiased
MAR (Missing in X)	Unbiased	Unbiased
MAR (Missing in Y, X)	Biased	Unbiased
MNAR/NI (Missing in X)	Unbiased	?
MNAR/NI (Missing in Y, X)	Biased	Biased

Best Practices (A-B & P)

- Define the population of interest.
- Report the share of missing values for each variable and descriptive statistics for both complete and incomplete cases. Do fully observed units differ systematically from partially observed ones?
- Theorize the missingness mechanism. Is the pattern of missingness driven by (a) pure chance, (b) factors unrelated to the variables of interest, (c) values of the independentvariables, (d) values of the dependent variable, or (e) unobservable factors?
  - Under (a), (b),and (c), LWD can be used without fear that it will introduce bias in regression estimates.
  - Under (d), MI can sometimes reduce bias, but it only offers guarantees if data are MAR and the imputation model's assumptions are satisfied.
  - Under (e) data are NMAR and neither LWD nor MI promise unbiased estimates.

Best Practices 2 (A-B & P)

- Check for divergence between LWD and MI results. If estimates do diverge, which "new" observations have a strong influence on the results? Are these observations theoretically distinct?
- Robustness checks. Do alternative imputation procedures or tuning parameters produce different results? Does the imputation model have good predictive power? Does it fill in reasonable values for missing observations?

Methods for imputation in MICE

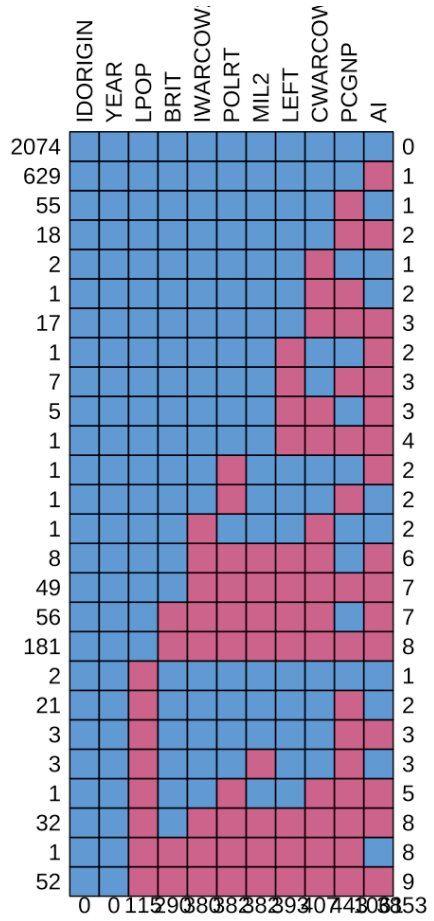
There are many different ways of imputing missing data.

- The default method for numerical data is Predictive Mean Matching (`pmm`).
- There are a suite of GLM methods (`norm`, `logreg`, `polr`, `polyreg`)
- There are machine learning approaches, `cart` and `rf`.
- The `{ImputeRobust}` package in R has a number of `gamlss` imputation models that can be plugged into `mice`.

MD Pattern

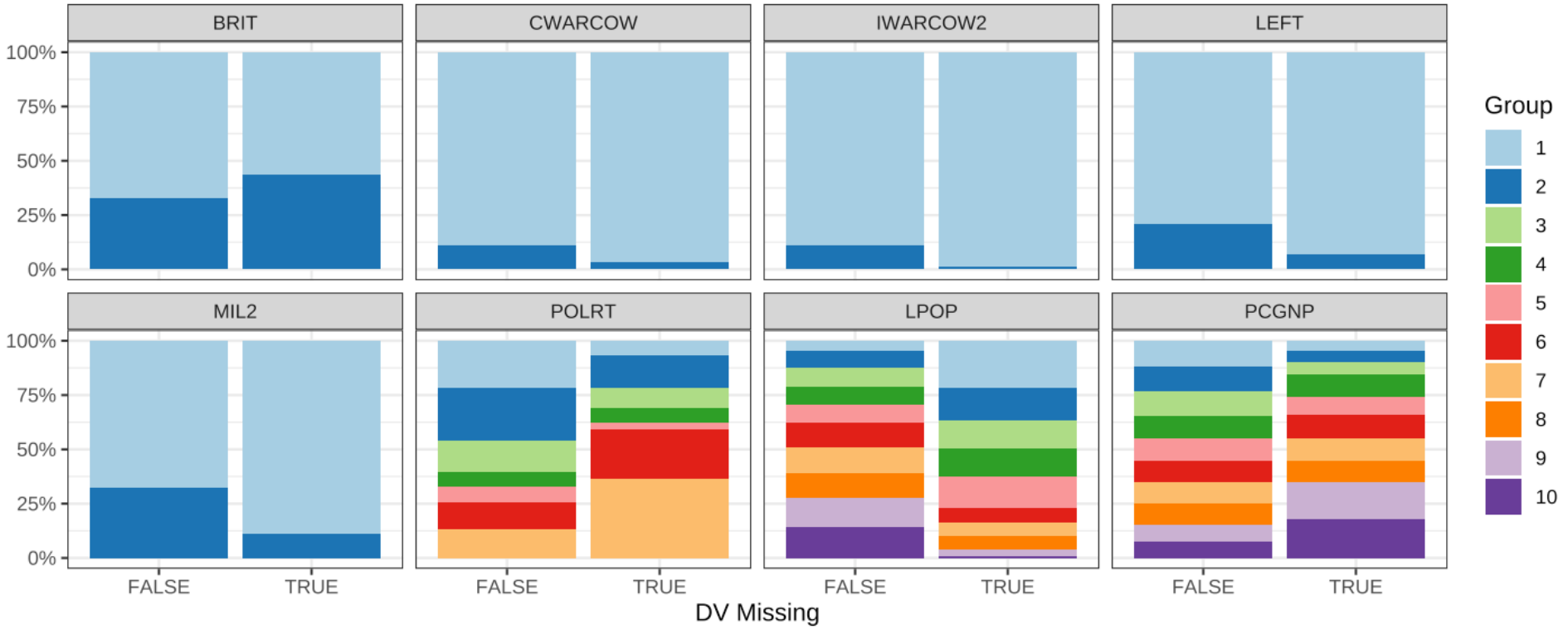
```
library(rio)
library(mice)
library(dplyr)
poetate <- rio::import("data/poetate.dta")
poetate <- poetate %>%
  select(IDORIGIN,
         YEAR, AI, POLRT, LPOP, PCGNP, LEFT,
          MIL2, BRIT, CWARCOW, IWARCOW2) %>%
  mutate(LEFT = as.factor(LEFT),
         MIL2 = as.factor(MIL2),
         BRIT = as.factor(BRIT),
         CWARCOW = as.factor(CWARCOW),
         IWARCOW2 = as.factor(IWARCOW2))

md.pattern(poetate, rotate.names=TRUE)
```



Missing DV Related to Observed Stuff?

```
mi_diag_plot(AI ~ POLRT + LPOP + PCGNP + LEFT + MIL2 + BRIT + CWARCOW + IWARCOW2, data=poetate, nrow=2)
```



Some Useful Diagnostics

Inbound Statistic - Proportion of usable cases for imputing Y_j from Y_k:

$$I_{jk} = \frac{\sum_{i=1}^{N}(1 - r_{ij})r_{ik}}{\sum_{i=1}^{n} 1 - r_{ij}}$$

where $r_{ij} = 1$ if observation i has an observed value on variable Y_j and 0 otherwise.

Outbound Statistic - Evaluate whether Y_j could be a potential predictor of Y_k

$$O_{jk} = \frac{\sum_{i=1}^{N} r_{ij}(1 - r_{ik})}{\sum_{i=1}^{n} r_{ij}}$$

In R

```
pats <- md.pairs(poetate)
inbound <- pats$mr/(pats$mr+pats$mm)
round(na.omit(inbound), 3)
```

```
##          IDORIGIN YEAR    AI POLRT  LPOP PCGNP  LEFT  MIL2  BRIT CWARCOW
## AI              1    1 0.000 0.642 0.917 0.660 0.631 0.644 0.728   0.621
## POLRT           1    1 0.005 0.000 0.775 0.170 0.008 0.008 0.241   0.005
## LPOP            1    1 0.235 0.252 0.000 0.017 0.261 0.235 0.539   0.252
## PCGNP           1    1 0.185 0.284 0.745 0.000 0.271 0.282 0.472   0.244
## LEFT            1    1 0.003 0.036 0.784 0.178 0.000 0.036 0.262   0.020
## MIL2            1    1 0.010 0.008 0.770 0.168 0.008 0.000 0.241   0.008
## BRIT            1    1 0.003 0.000 0.817 0.193 0.000 0.000 0.000   0.000
## CWARCOW         1    1 0.012 0.066 0.789 0.177 0.054 0.069 0.287   0.000
## IWARCOW2        1    1 0.005 0.003 0.776 0.171 0.003 0.003 0.237   0.000
##          IWARCOW2
## AI          0.644
## POLRT       0.008
## LPOP        0.261
## PCGNP       0.289
## LEFT        0.036
## MIL2        0.008
## BRIT        0.000
## CWARCOW     0.066
## IWARCOW2    0.000
## attr(,"na.action")
## IDORIGIN     YEAR
##        1        2
## attr(,"class")
## [1] "omit"
```

Outbound

```
outbound <- pats$rm/(pats$rm+pats$rr)
round(outbound, 3)
```

```
##          IDORIGIN YEAR    AI POLRT  LPOP PCGNP  LEFT  MIL2  BRIT CWARCOW
## IDORIGIN        0    0 0.329 0.119 0.036 0.137 0.122 0.119 0.090   0.126
## YEAR            0    0 0.329 0.119 0.036 0.137 0.122 0.119 0.090   0.126
## AI              0    0 0.000 0.001 0.012 0.038 0.000 0.002 0.000   0.002
## POLRT           0    0 0.240 0.000 0.010 0.044 0.005 0.001 0.000   0.010
## LPOP            0    0 0.313 0.095 0.000 0.106 0.099 0.095 0.076   0.103
## PCGNP           0    0 0.252 0.023 0.001 0.000 0.025 0.023 0.020   0.026
## LEFT            0    0 0.236 0.001 0.011 0.042 0.000 0.001 0.000   0.008
## MIL2            0    0 0.240 0.001 0.010 0.044 0.005 0.000 0.000   0.010
## BRIT            0    0 0.263 0.031 0.021 0.071 0.035 0.031 0.000   0.040
## CWARCOW         0    0 0.234 0.001 0.010 0.038 0.003 0.001 0.000   0.000
## IWARCOW2        0    0 0.240 0.001 0.011 0.045 0.005 0.001 0.000   0.010
##          IWARCOW2
## IDORIGIN    0.118
## YEAR        0.118
## AI          0.001
## POLRT       0.000
## LPOP        0.095
## PCGNP       0.023
## LEFT        0.000
## MIL2        0.000
## BRIT        0.031
## CWARCOW     0.000
## IWARCOW2    0.000
```

More Statistics

Influx - similar to the inbound statistic, but summed over all variables.

$$I_j = \frac{\sum_{j=1}^{p} \sum_{k=1}^{p} \sum_{i=1}^{n} (1 - r_{ij}) r_{ik}}{\sum_{k=1}^{p} \sum_{i=1}^{n} r_{ik}}$$

Outflux - potential usefulness for imputing other variables.

$$O_j = \frac{\sum_{j=1}^{p} \sum_{k=1}^{p} \sum_{i=1}^{n} r_{ij} (1 - r_{ik})}{\sum_{k=1}^{p} \sum_{i=1}^{n} 1 - r_{ij}}$$

More Statistics

FICO - Potential gain in efficiency from using MI over complete case analysis.

$$F_j = \frac{\sum_{i=1}^{n} r_{ij} c_i}{\sum_{i=1}^{n} r_{ij}}$$

where c_i is 1 if observation i has complete data an 0 otherwise.

```
round(flux(poetate), 3)
```

```
##           pobs influx outflux  ainb  aout  fico
## IDORIGIN 1.000  0.000   1.000 0.000 0.120 0.356
## YEAR     1.000  0.000   1.000 0.000 0.120 0.356
## AI       0.671  0.251   0.032 0.749 0.006 0.040
## POLRT    0.881  0.039   0.229 0.322 0.031 0.270
## LPOP     0.964  0.015   0.793 0.405 0.098 0.332
## PCGNP    0.863  0.067   0.284 0.477 0.039 0.254
## LEFT     0.878  0.042   0.220 0.335 0.030 0.267
## MIL2     0.881  0.039   0.229 0.322 0.031 0.270
## BRIT     0.910  0.028   0.399 0.301 0.052 0.293
## CWARCOW  0.874  0.045   0.210 0.352 0.029 0.263
## IWARCOW2 0.882  0.038   0.230 0.320 0.031 0.270
```

Imputation

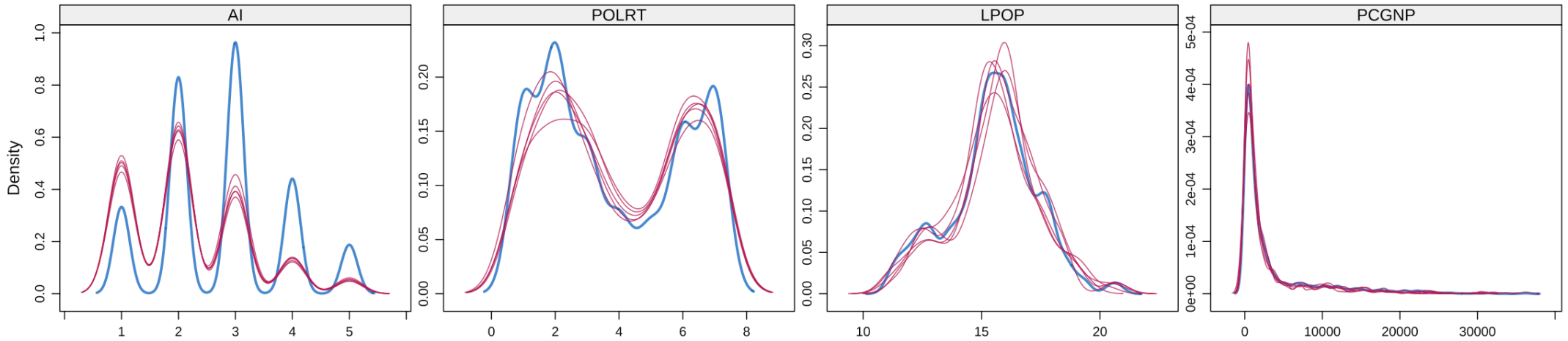
```
pt.mice <- mice(poetate, printFlag=F, m=5, maxit=20)
```

```
summary(pt.mice)
```

```
## Class: mids
## Number of multiple imputations:  5
## Imputation methods:
## IDORIGIN     YEAR       AI    POLRT     LPOP    PCGNP     LEFT     MIL2
##       ""       ""    "pmm"    "pmm"    "pmm"    "pmm" "logreg" "logreg"
##     BRIT  CWARCOW IWARCOW2
## "logreg" "logreg" "logreg"
## PredictorMatrix:
##          IDORIGIN YEAR AI POLRT LPOP PCGNP LEFT MIL2 BRIT CWARCOW IWARCOW2
## IDORIGIN        0    1  1     1    1     1    1    1    1       1        1
## YEAR            1    0  1     1    1     1    1    1    1       1        1
## AI              1    1  0     1    1     1    1    1    1       1        1
## POLRT           1    1  1     0    1     1    1    1    1       1        1
## LPOP            1    1  1     1    0     1    1    1    1       1        1
## PCGNP           1    1  1     1    1     0    1    1    1       1        1
```

Diagnostic - Density

```
densityplot(pt.mice, layout = c(4,1))
```



When the distributions don't match, could be because:

- imputation model is a bad fit
- missing data mechanism is not MCAR
- both

Attempt 2

We could try logging `PCGNP` and greating, for the purposes of imputation, `AI` as a factor.

```
poetate <- poetate %>%
  mutate(loggnp = log(PCGNP),
         AI = as.factor(AI))
```

Before we impute, we need to recognize that `loggnp` and `PCGNP` are deterministically related. So we have to

- turn off the relationships between `PCGNP` and the other variables in the imputation model.
- use passive imputation for `PCGNP`

Mice Control

Turn off the relationships between `PCGNP` and the other variables in the imputation model.

```
pm <- make.predictorMatrix(poetate)
pm["PCGNP", ] <- pm[,"PCGNP"] <- 0
```

Use passive imputation for `PCGNP`

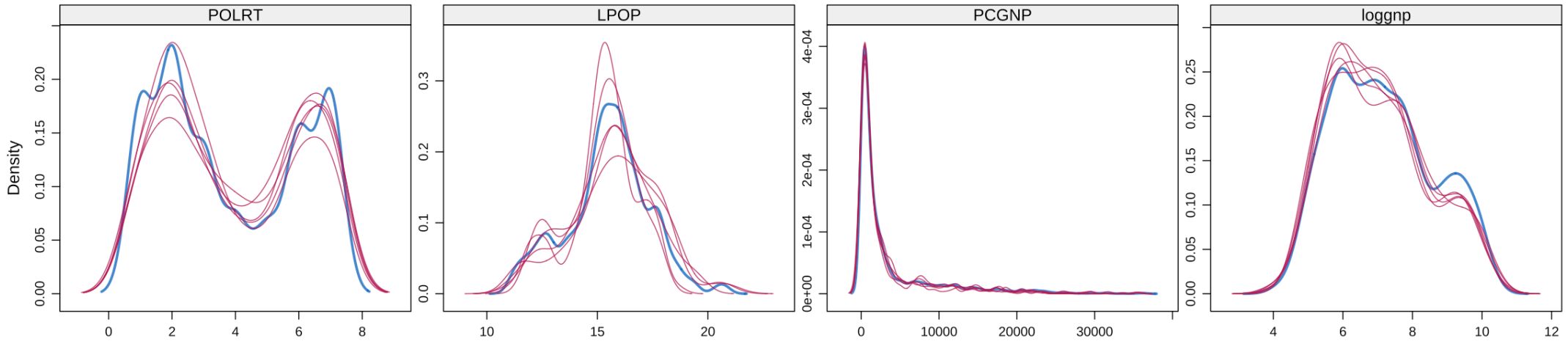
```
meth <- make.method(poetate)
meth["PCGNP"] <- "~I(exp(loggnp))"
```

Generate the imputations again using the updated `meth` and `pm` values.

```
pt.mice2 <- mice(poetate, printFlag=F, m=5, maxit=20,
                 meth=meth, pred=pm)
```

Diagnostic - Density (again)

```
densityplot(pt.mice2, layout = c(4,1))
```



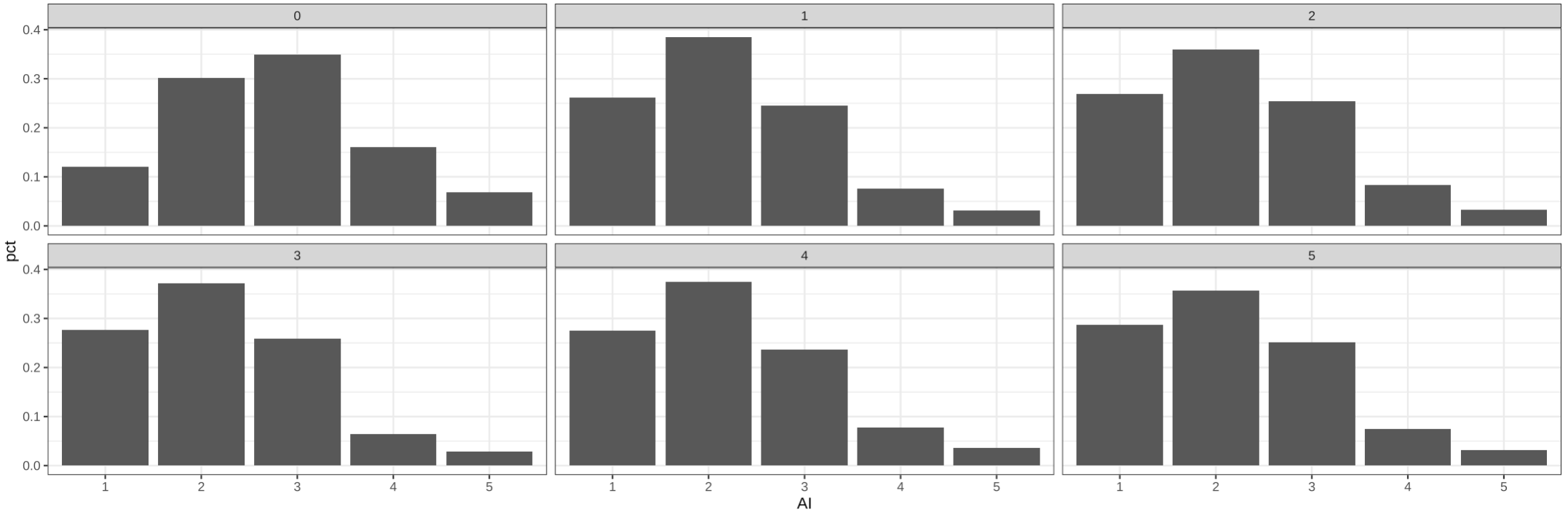
We are pretty confident about the GNP imputation now.

Factors

```
comps <- lapply(1:5, function(x)complete(pt.mice2, x))
comps <- do.call(rbind, comps)
comps <- rbind(poetate %>% mutate(loggnp = log(PCGNP)), comps)
comps$draw <- rep(0:5, each=nrow(poetate))
obsai <- as.numeric(!is.na(poetate$AI))
obsai <- factor(obsai, levels=c(0,1),
                labels=c("Imputed", "Observed"))
comps$obs <- rep(obsai, 6)
comps <- comps %>%
  filter((draw == 0 & obsai == "Observed") |
         (draw > 0 & obsai == "Imputed"))
comps <- comps %>%
  group_by(draw, obs, AI) %>%
  summarise(n = n()) %>%
  ungroup %>%
  group_by(draw, obs) %>%
  mutate(pct = n/sum(n))

ggplot(comps, aes(x=AI, y=pct)) +
  geom_bar(stat="identity") +
  facet_wrap(~draw) +
  theme_bw()
```

Plot



Imputations by Pr(incomplete)

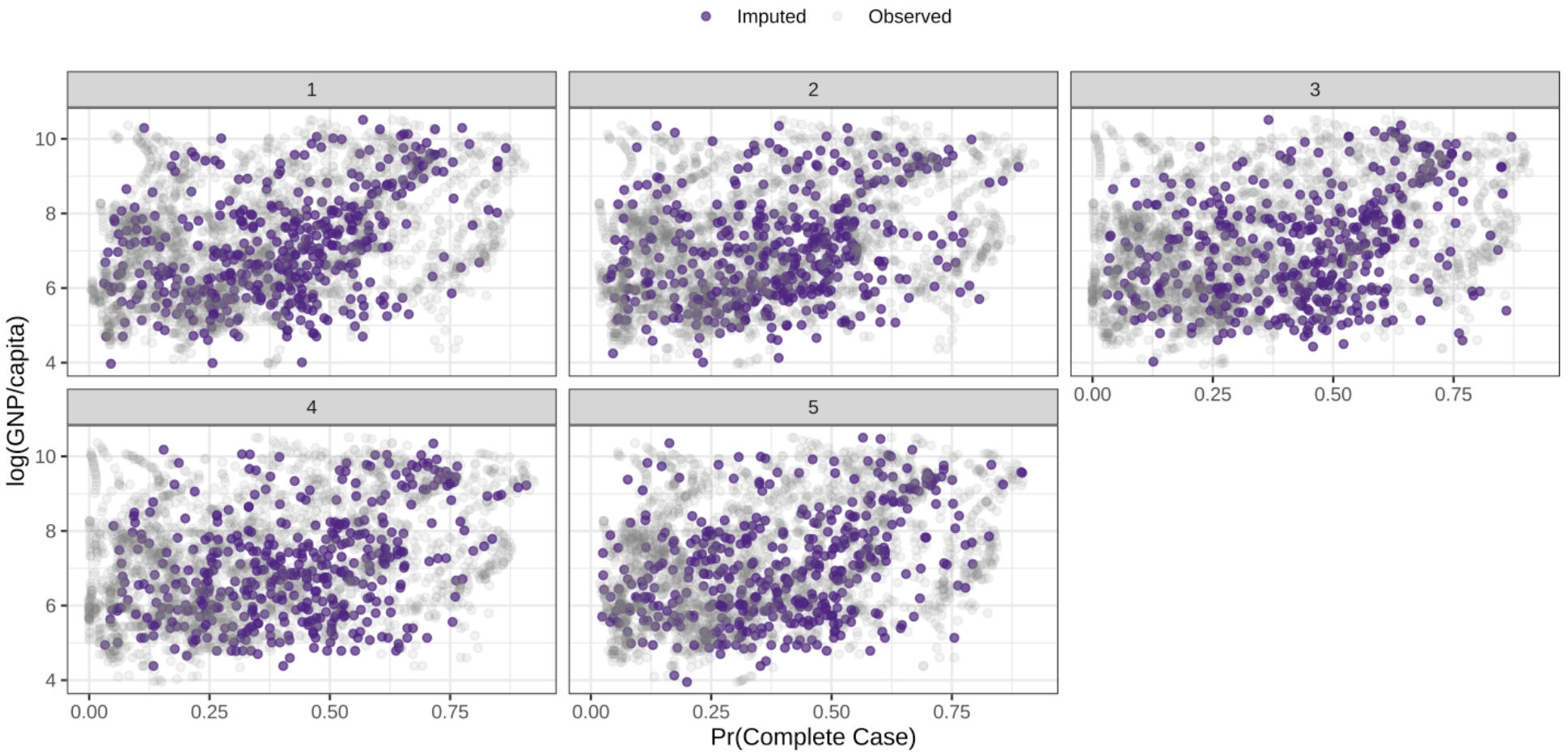
```
library(gamlss)
comps <- lapply(1:5, function(x)complete(pt.mice2, x))
```

```
mods <- lapply(comps, function(D){
  D$comp <- ici(pt.mice2)
  gamlss(comp ~ pb(loggnp) +
                pb(LPOP) + POLRT +
                AI + LEFT + MIL2 +
                IWARCOW2 + CWARCOW,
              data = D, family=BI)})
```

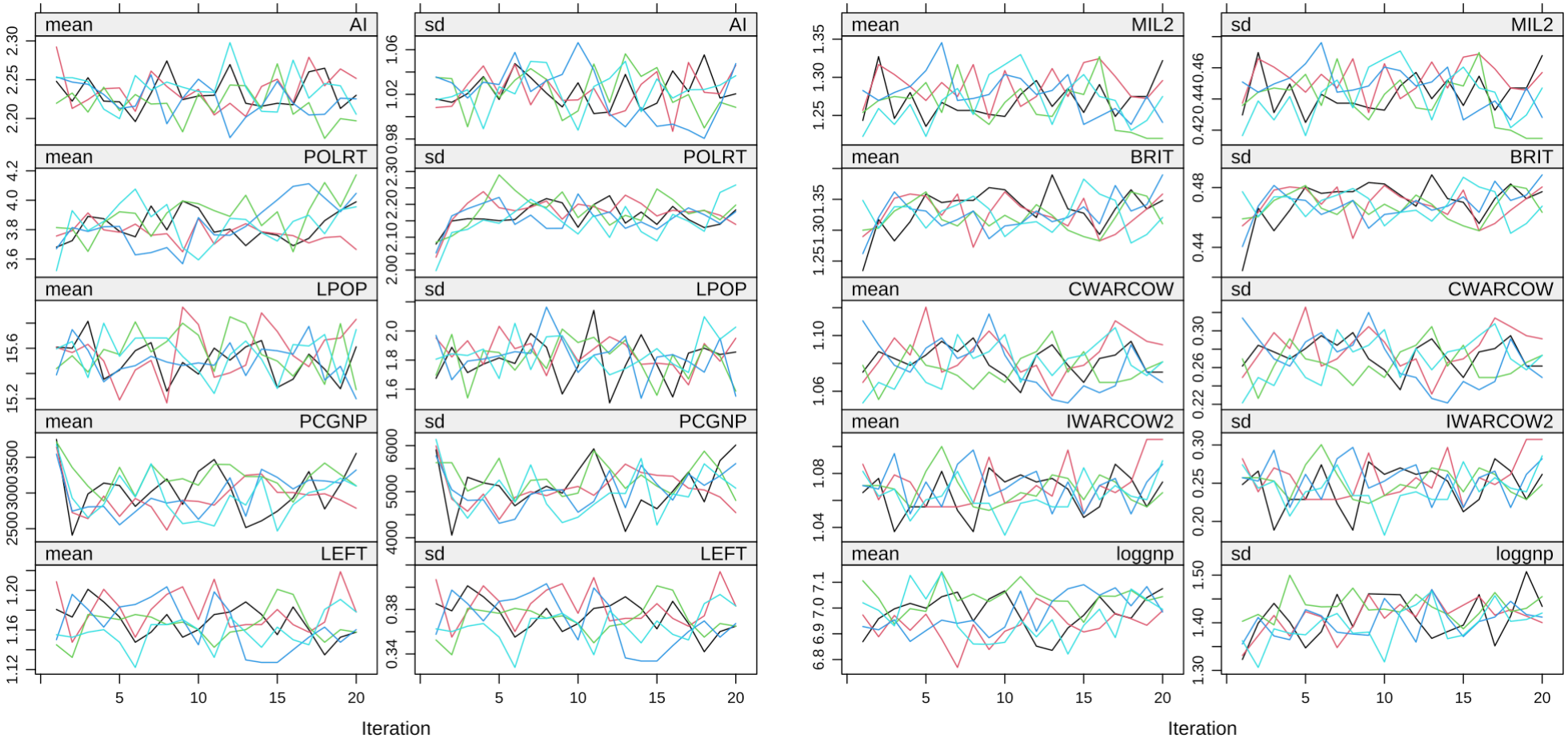
```
fits <- lapply(mods, predict, type="response")
for(i in 1:5)comps[[i]]$pcomp <- fits[[i]]
comps <- do.call(rbind, comps)
comps$gnpobs <- as.numeric(!is.na(poetate$loggnp))
comps$gnpobs <- factor(comps$gnpobs,
                       labels=c("Imputed", "Observed"))
comps$draw <- rep(1:5, each=nrow(poetate))
```

```
pal2 <- c("#4F2683","#807F83")
ggplot(comps, aes(x=pcomp, y=loggnp,
                  colour=gnpobs, fill=gnpobs,
                  alpha=gnpobs)) +
  geom_point() +
  facet_wrap(~draw) +
  theme_bw() +
  scale_colour_manual(values=pal2) +
  scale_alpha_manual(values=c(.75,.1)) +
  theme(legend.position="top") +
  labs(x="Pr(Complete Case)", y="log(GNP/capita)",
       colour="", fill="", alpha="")
```

Plot


Imputed
Observed
1
2
3
4
5
log(GNP/capita)
10
8
6
4
0.00
0.25
0.50
0.75
Pr(Complete Case)

Convergence of Imputation Models



Missing TSCS Data

Honaker and King (2010) show that MICE-type procedures tend to systematically miss trends in the data.

- Amelia II has procedures to deal with TSCS missingness.
- Essentially amounts to putting time-trends in the imputation models, either through polynomials or cubic-splines in time.
- Can allow for unit-specific time-trends as well by interacting time polynomials with categorical indicator of group membership.
- Still assumes a MAR mechanism.

PTK Model

```
comps <- lapply(1:5, function(x)complete(pt.mice2, x))
library(plm)
pcomps <- lapply(comps, function(x)pdata.frame(x, index=c("IDORIGIN", "YEAR")))
for(i in 1:length(pcomps)){
  pcomps[[i]]$AI <- as.numeric(pcomps[[i]]$AI)
  pcomps[[i]]$lagAI <- lag(pcomps[[i]]$AI)
}

mice.mods <- lapply(pcomps, function(x)
    lm(AI ~ lagAI + POLRT + LPOP +  I(PCGNP/10000) +
    LEFT + MIL2 + BRIT + CWARCOW + IWARCOW2, data=as.data.frame(x)))
library(mitools)
mice.pool <- MIcombine(mice.mods)
```

Summarize Models

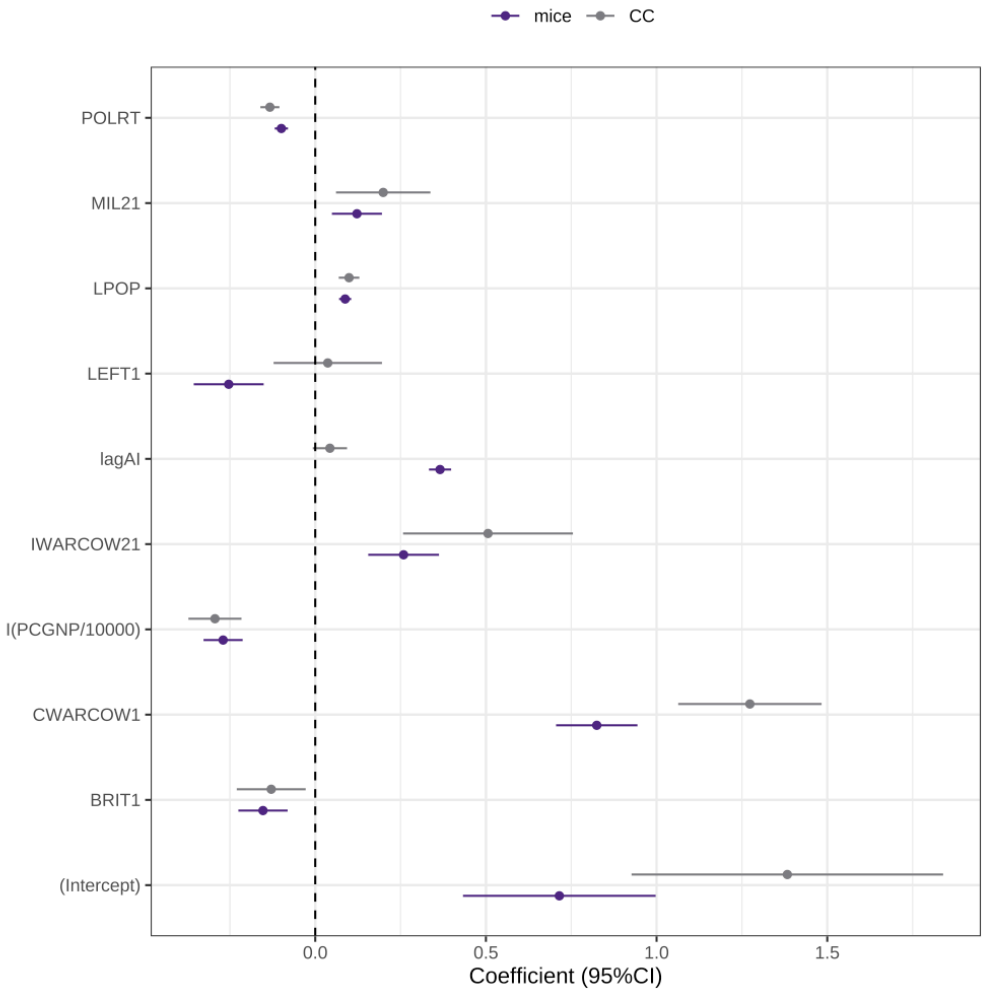
```
cis1 <- as.matrix(summary(mice.pool)[,c(1,3,4)])
```

```
## Multiple imputation results:
##       MIcombine.default(mice.mods)
##                     results          se      (lower      upper) missInfo
## (Intercept)      0.71506971 0.141630536  0.43280491  0.99733450     25 %
## lagAI            0.36558684 0.016496179  0.33308158  0.39809210     14 %
## POLRT           -0.09931811 0.009721855 -0.11879560 -0.07984062     29 %
## LPOP             0.08759185 0.008969079  0.06975737  0.10542632     24 %
## I(PCGNP/10000) -0.26991491 0.029092209 -0.32726781 -0.21256202     15 %
## LEFT1           -0.25352237 0.050594387 -0.35599564 -0.15104911     36 %
## MIL21            0.12210403 0.037123331  0.04879007  0.19541800     17 %
## BRIT1           -0.15310725 0.035713908 -0.22533347 -0.08088103     35 %
## CWARCOW1         0.82470136 0.059119653  0.70541131  0.94399141     34 %
## IWARCOW21        0.25880739 0.052730116  0.15513283  0.36248194     11 %
```

```
cmod <- lm(AI ~ lagAI + POLRT + LPOP +  I(PCGNP/10000) +
    LEFT + MIL2 + BRIT + CWARCOW + IWARCOW2,
    data=subset(as.data.frame(pcomps[[1]]), ici(pt.mice2)))
cis0 <- cbind(coef(cmod), confint(cmod))
x <- rbind(cis1, cis0)
x <- as.data.frame(x)
x$parm <- rownames(cis1)
rownames(x) <- NULL
x$parm <- as.factor(x$parm)
x$mod <- factor(rep(c(1,2), each=nrow(cis1)), levels=c(1,2),
    labels=c("mice", "CC"))
names(x)[1:3]<- c("coef", "lower", "upper")
```

Graph

```
ggplot(x, aes(x=coef,
              y=parm,
              colour=mod)) +
  geom_point(
    position=position_dodge(width=.5)) +
  geom_errorbarh(
    aes(xmin=lower, xmax=upper),
    position = position_dodge(width=.5),
    height=0) +
  geom_vline(
    xintercept=0,
    col="black",
    lty=2) +
  scale_colour_manual(
    values=pal2) +
  theme_bw() +
  theme(
    legend.position="top") +
  labs(
    x="Coefficient (95%CI)",
    y="",
    colour="")
```



Posterior Predictive Checks

Burgette and Reiter (2010, *American Journal of Epidemiology*) suggest that posterior predictive checks can help diagnose problems with inadequate imputations.

- Create 500 imputations from the relevant imputation model
- Create 500 imputations where each variable is missing completely and filled in by the imputation model
- Estimate some quantity of analytical interest (e.g., a set of regression coefficients) on both steps 1 and 2 above,
- Test to see if there is any difference between the two.

Creating the Full Data Impuatations

```
ppdat <- make_pp(AI ~ POLRT + LPOP + PCGNP + LEFT + MIL2 + BRIT + CWARCOW + IWARCOW2, data=poetate)

pp.mice <- mice(ppdat, printFlag=F, m=25, maxit=5)
res <- extract_pp(pp.mice, ppdat, poetate)
orig.imp <- mice(poetate, printFlag=F, m=25, maxit=5,
                 meth=meth, pred=pm)
```

Estimate Regression and p-values

```
library(plm)
pt.comp <- lapply(1:orig.imp$m, \(i)complete(orig.imp, i))
l1 <- lapply(pt.comp, function(x){
  lm(as.numeric(AI) ~ POLRT + LPOP +  I(PCGNP/10000) +
    LEFT + MIL2 + BRIT + CWARCOW + IWARCOW2, data=x)})
l2 <- lapply(res, function(x){
    lm(as.numeric(AI) ~ POLRT + LPOP +  I(PCGNP/10000) +
      LEFT + MIL2 + BRIT + CWARCOW + IWARCOW2, data=x)})
b1 <- sapply(l1, coef)
b2 <- sapply(l2, coef)
pval <- apply(b1>b2, 1, mean)
pval <- 2*ifelse(pval > .5, 1-pval, pval)
pval
```

```
##    (Intercept)          POLRT           LPOP I(PCGNP/10000)          LEFT1
##              0              0              0              0              0
##          MIL21          BRIT1       CWARCOW1      IWARCOW21
##              0              0              0              0
```

Omnibus Test for Regression Differences

```
library(car)
coef <- rowMeans(b1-b2)
v <- var(t(b1-b2))
linearHypothesis(model=list(df.residual=NULL),
  hypothesis.matrix=diag(length(coef)),
  vcov.=v, coef.=coef, rhs=rep(0, length(coef)))
```

```
##
## Linear hypothesis test:
##
##
## Model 1: restricted model
## Model 2: list(df.residual = NULL)
##
## Note: Coefficient covariance matrix supplied.
##
##   Df Chisq Pr(>Chisq)
## 1
## 2  9 624.5  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Sensitivity to MAR Assumption.

The MAR assumption assumes that there is no systematic difference between responders and non-responders in terms of the imputed values.

To test the sensitivity of our analysis to this assumption, we can systematically change the imputations to operationalize a particular violation of MAR.

- For numeric variables, add δ, a value representing the difference between responders and non-responders on the variable of interest.
- For categorical variables, calculate the distribution of the imputed values for each observation. Then, change the probabilities in a particular way (e.g., add δ do some and subtract δ from others). Then, resample from the imputed values with the new probabilities.

Sensitivity in R.

```
pt.mice2 <- mice(poetate, printFlag=T, m=100, maxit=5,
                 meth=meth, pred=pm)
```

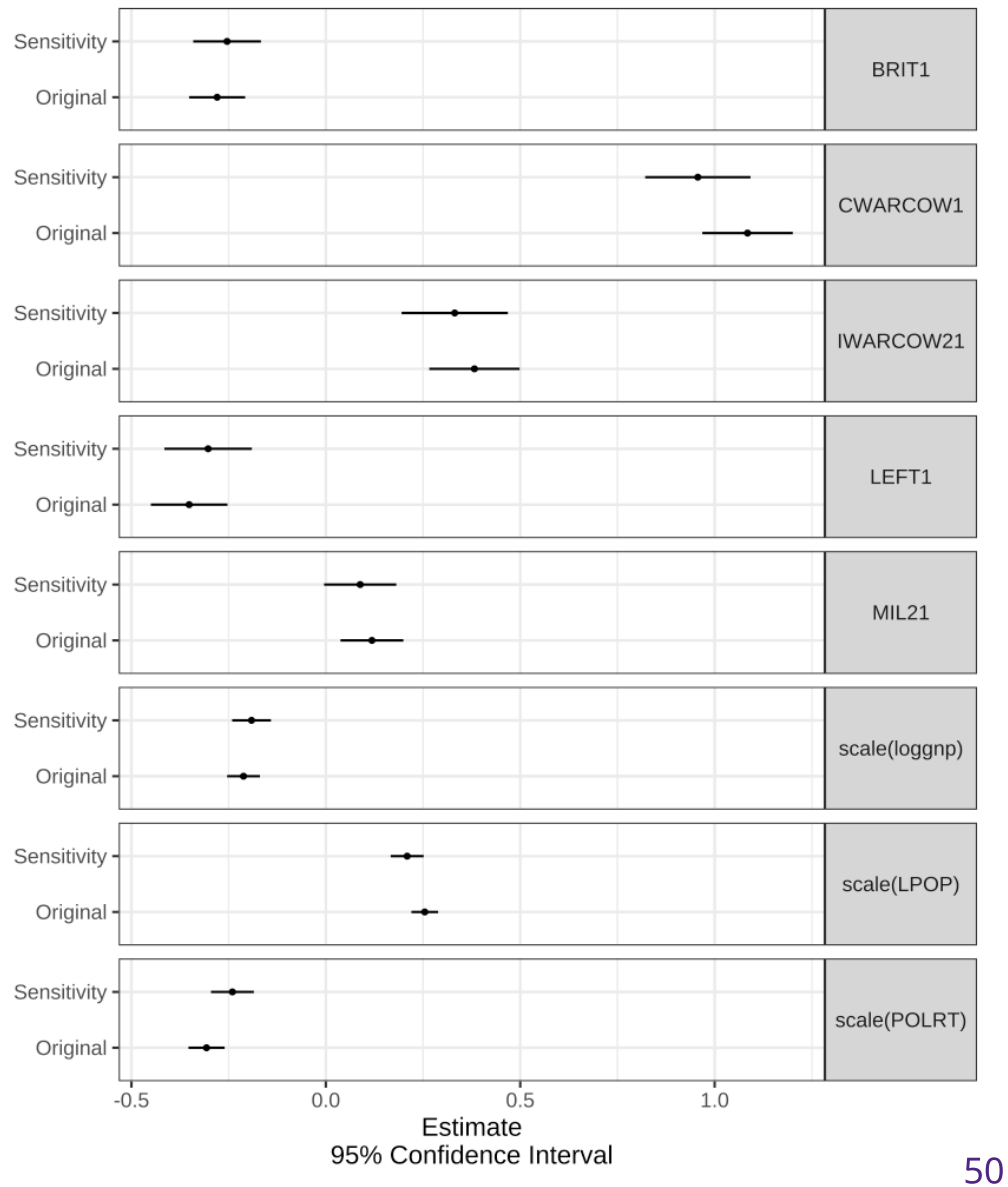
```
sense_ai <- sens_impute("AI", poetate, pt.mice2,
                        type="cat", delta=.1)

comps2 <- lapply(1:pt.mice2$m, \(i)complete(pt.mice2, i))
compss <- lapply(1:sense_ai$m, \(i)complete(sense_ai, i))

omod <- lapply(comps2, \(d)lm(as.numeric(AI) ~ scale(POLRT) +
    scale(LPOP) + scale(loggnp) + LEFT + MIL2 + BRIT +
    CWARCOW + IWARCOW2, data=d))
smod <- lapply(compss, \(d)lm(as.numeric(AI) ~ scale(POLRT) +
    scale(LPOP) + scale(loggnp) + LEFT + MIL2 + BRIT +
    CWARCOW + IWARCOW2, data=d))

omod_sum <- MIcombine(omod)
smod_sum <- MIcombine(smod)

plot.dat <- as_tibble(summary(omod_sum), rownames="param") %>%
  setNames(c("param", "estimate", "se", "lwr", "upr", "mi")) %>%
  mutate(model = "Original")
plot.dat <- plot.dat %>%
  bind_rows(as_tibble(summary(smod_sum), rownames="param") %>%
    setNames(c("param", "estimate", "se", "lwr", "upr", "mi")) %>
    mutate(model = "Sensitivity"))
```



Review

1. Why are missing data problematic?
2. What methods can we use to deal with missing data and what are their implications?
3. How do we know if our imputation "worked"?
4. Example

Exercise - Good APSR Replication

```
load("data/good_df.rda")


form1 <- GeWom ~ FemDel_P, data = df_original
form2 <- GeWom ~ FemDel_P + UNSCR + ImUN + ImOth + NAP
form3 <- GeWom ~ FemDel_P + GDI + SEP_Fem + TeenPreg
form4 <- GeWom ~ FemDel_P + PolInt_Cmb + WomParl
form5 <- GeWom ~ FemDel_P + state_prev_avg + female_combatants_exs
form6 <- GeWom ~ FemDel_P + GDI + WomParl + SEP_Fem + TeenPreg + NYT_p + UNSCR +
  Press_UNSC + ImUN + ImOth + NAP + PolInt_Cmb + JobEql_Cmb + LeadPol_Cmb +
  state_prev_avg + female_combatants_exs
```