Understanding and addressing missing data

Frank Edwards

Review of GLMs

The Generalized Linear Model

A linear predictor η:

$$\eta = \mathsf{X}\beta$$

A link function g

$$g(E(Y|X)) = \eta$$

A mean expectation $E(Y|X) = \mu$

$$\mu = g^{-1}(\eta)$$

The Normal model

OLS:

$$y|X \sim Normal(\mu, \sigma^2)$$

$$E(Y|X) = X\beta = \mu$$

In GLM form:

$$g(E(Y|X)) = X\beta = \mu$$

Where g is the Identity function ($f(x) = x$)

In R: `lm(y~x)`

The Logistic model

$$Y|X \sim Bernoulli(p)$$

$$logit(E(Y|X)) = X\beta = logit(p)$$

$$p = logit^{-1}(X\beta)$$

In R: `glm(y~x, family = binomial)`

The Multinomial model

$$Y|X \sim Multinomial(p_1, p_2 \cdots p_k)$$

$$log\frac{Pr(y_i = 1)}{Pr(y_i = K)} = \beta x_i$$

$$\cdots$$

$$log\frac{Pr(y_i = K-1)}{Pr(y_i = K)} = \beta x_i$$

In R: `nnet::multinom(y~x)`

The Poisson model

$$y \sim Poisson(\lambda)$$

$$E(y|x) = e^{\lambda}$$

$$log(E(y|x)) = \lambda = \beta X$$

In R: `glm(y~x, family = poisson)`

Missing data

Why should we care?

- Most statistical software will conduct “complete-case analysis” by default
- This may result in throwing away a lot of perfectly good information!
- Listwise deletion understates uncertainty, may result in bias

Three general causes of missing data: MCAR

- **Missing completely at random (MCAR)**: The probability of a value being missing is the same for all observations in the data. Missingness is determined by a coin flip/dice roll
- Potential MCAR mechanisms: survey non-response due to exogenous factors: e.g. lost mail, bad weather, software errors.
- Can be verified by comparing group means of missing and non-missing data on observables: for large N, values are equal

Three general causes of missing data: MAR

- **Missing at random (MAR)**: The probability of a value being missing is *not* completely at random.
- The probability of a value being missing is determined by other variables in the data
- After controlling for other values in the data, missingess is random
- Potential MAR mechanisms: people with high income less likely to report total wealth; places with high poverty less likely to submit voluntary administrative data; news reports unlikely to identify other characteristics of child victims of crime / violence;

Three general causes of missing data: MNAR

- **Missing not at random (MNAR)**: The probability of a value being missing depends on either *A)* some unobserved variable or *B)* the value itself (censorship)
- Examples: police departments with high crime opt-out of the UCR; police departments with high levels of use-of-force opt-out of reporting to federal arrest-related-deaths programs; persons who do not vaccinate their children opt-out of answering a survey question about vaccination; people who die (unrecorded) do not respond to a wave of a survey
- We cannot distinguish between MAR and MNAR: you must think carefully about missing data mechanisms

Mechanisms of missing data

- Missing completely at random: missingness determined by a coin flip
- Missing (conditionally) at random: missingness on variable x determined by some other variable y
- Missing not at random: missingness on variable x depends only on variable x (or some unobserved variable z)

Let’s look at some missing data

Returning to the Fatal Encounters data: Is race Missing at Random?

```
kable(fe%>%group_by(race)%>%summarise(count = n()))
```

race	count
African-American/Black	5201
Asian/Pacific Islander	350
European-American/White	7748
Hispanic/Latino	3188
Middle Eastern	40
Native American/Alaskan	246
Race unspecified	8687

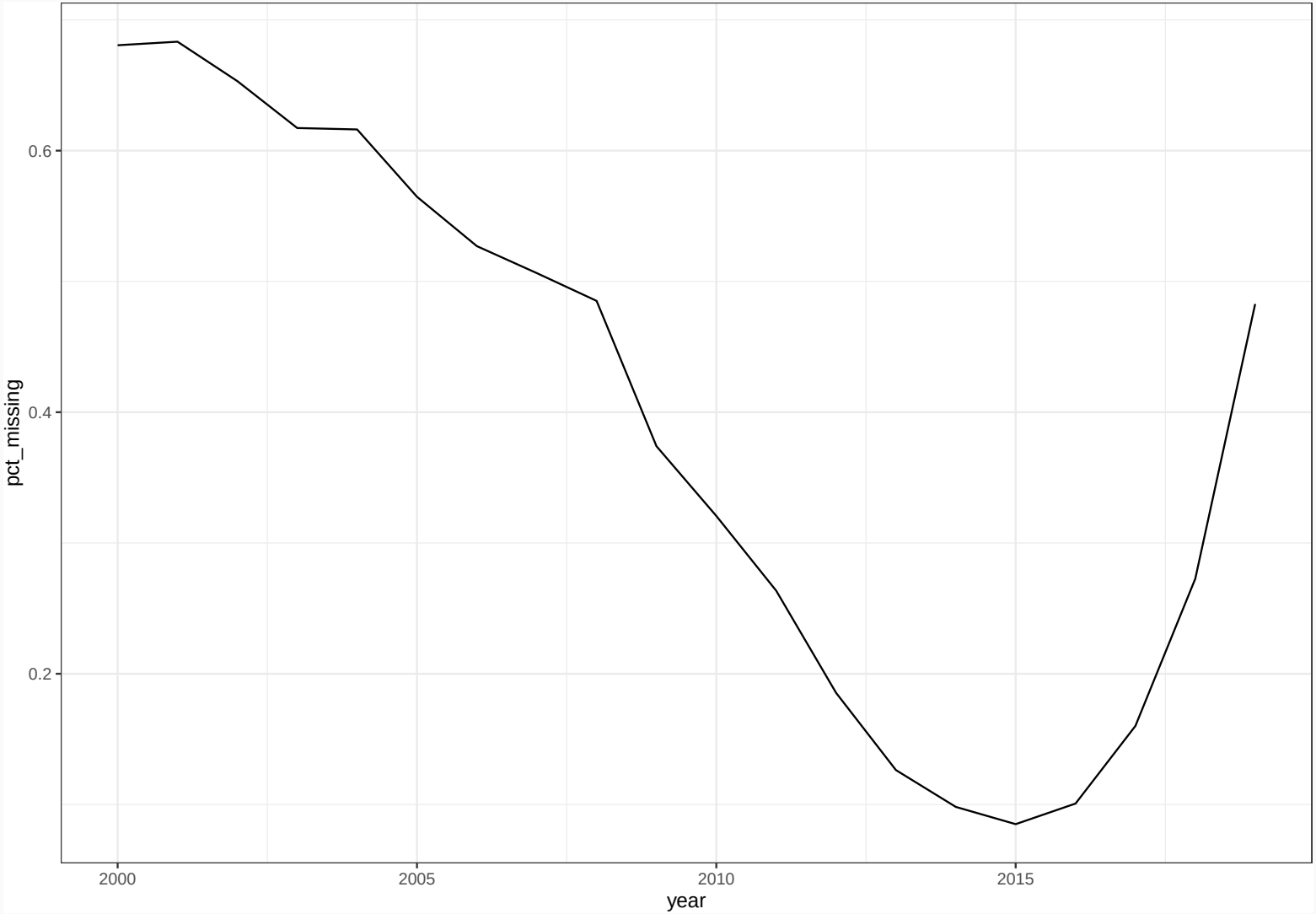
MAR check: age, sex

```
kable(fe%>%group_by(race=="Race unspecified")%>%
        summarise(mean_age = mean(age, na.rm=TRUE),
                  pct_male = sum(sex=="M", na.rm=TRUE)/n(),
                  pct_female = sum(sex=="F", na.rm=TRUE)/n()))
```

race == "Race unspecified"	mean_age	pct_male	pct_female
FALSE	34.50277	0.9022238	0.0960472
TRUE	35.93760	0.9045700	0.0887533

MAR check: year

```
ggplot(fe%>%group_by(year)%>%
        summarise(pct_missing = sum(race=="Race unspecified")/n())) +
  geom_line(aes(x = year, y = pct_missing))
```



Is race missing completely at random? Why?

Is race missing at random? Why?

Is race missing not at random? Why?

How do we handle missing data?

Basic approaches to missing data

- Listwise deletion (complete case analysis)
    - Appropriate for data with very few missing observations, and when missingness is completely at random
- Using alternative information (e.g. borrowing observation of sex from prior survey wave)
- Imputation of missing values (deterministic, stochastic)

Basic approaches to missing data, deterministic

- Missing value is generated by a fixed (non-random) procedure
- Many examples: linear interpolation, last observed, regression imputation
- This is generally a bad idea. Covariance estimates and standard errors are biased downward

Basic approaches to missing data, stochastic

- Missing value is generated through random sampling
- Many approaches, but multiple imputation has become industry standard

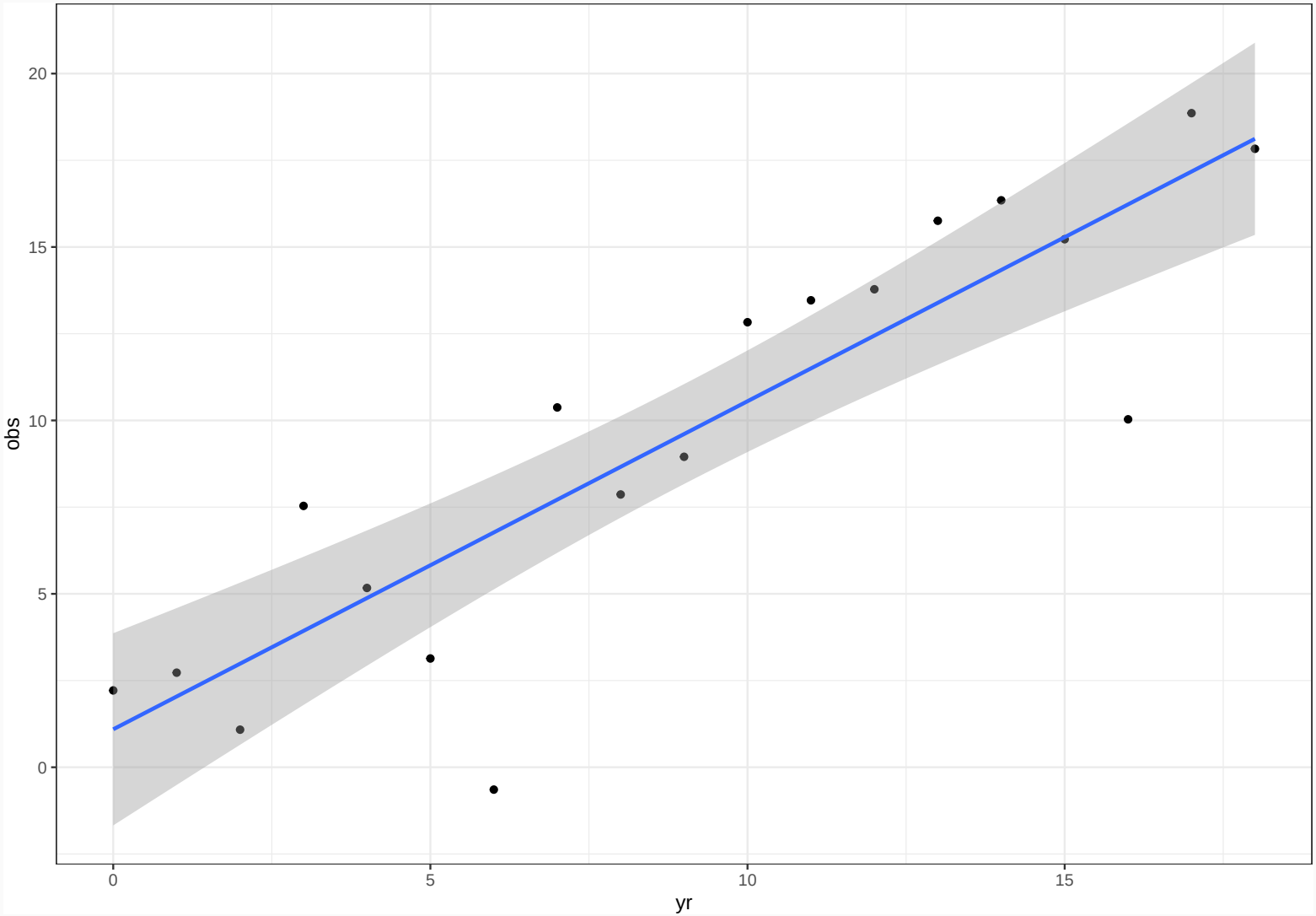
Multiple imputation

- Iterative modeling of all missing outcomes/predictors in model
- Produces series of fake datasets where missing values are predicted with from regression model (with error)
- Allows you to estimate uncertainty generated by missing data
- Does not recover “true” values
- Under missing at random assumption, generates unbiased parameter and variance estimates

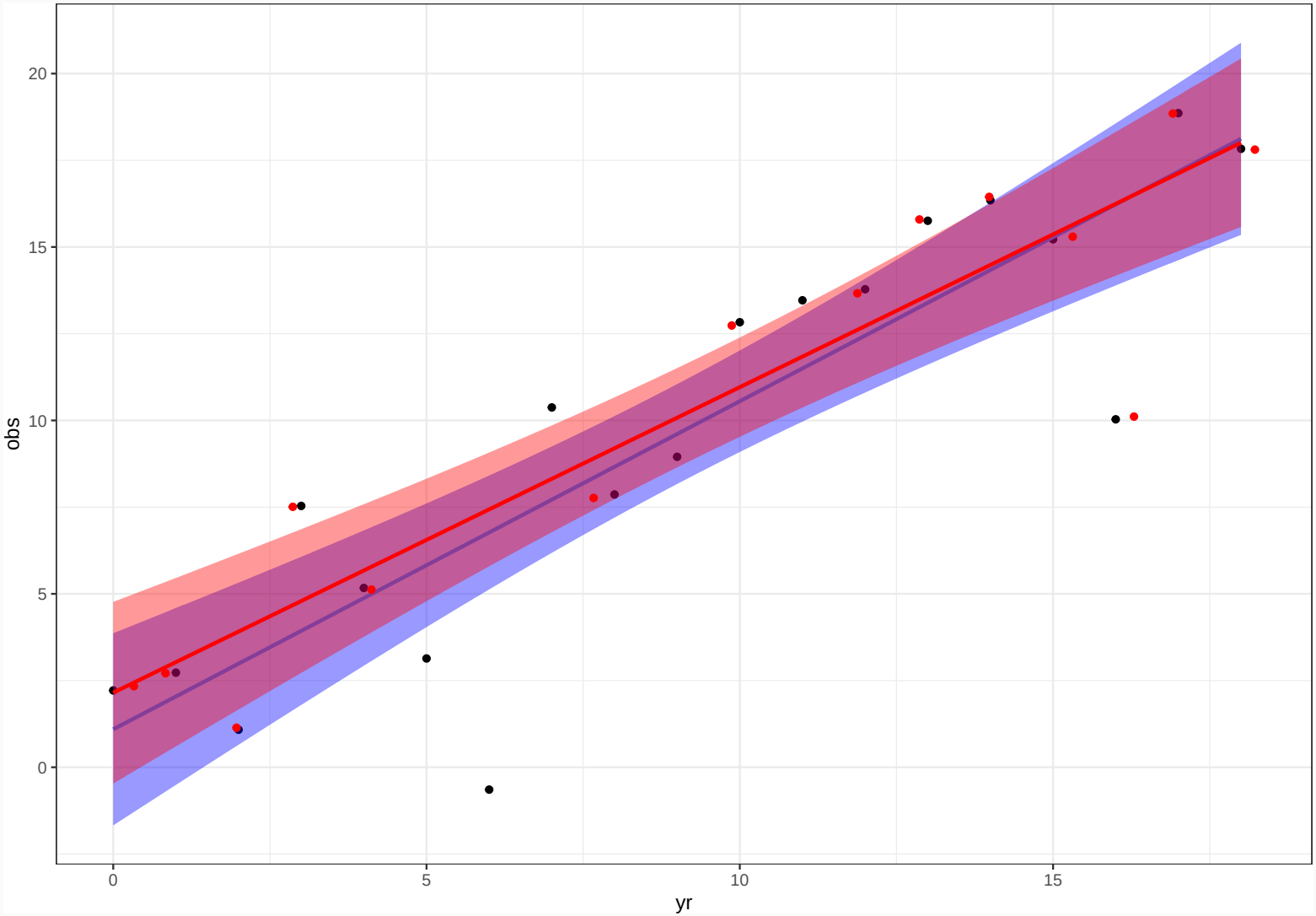
What muliple imputation does:

- Has two effects on model uncertainty
    - Increases your N because we aren't deleting data (pushes standard errors downward)
    - Adds in appropriate noise due to uncertainty around where missing values are (pushes standard errors upward)
- If missingess is associated with observables, MI can correct bias in parameter estimates

Examples of alternative approaches on a time series



Listwise deletion (remove missing values)



Listwise deletion (remove missing values)

```
tidy(lm(obs~yr, data = sim))
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic     p.value
##   <chr>          <dbl>     <dbl>     <dbl>       <dbl>
## 1 (Intercept)    1.09      1.31      0.833 0.416
## 2 yr             0.946     0.125     7.59  0.000000743
```

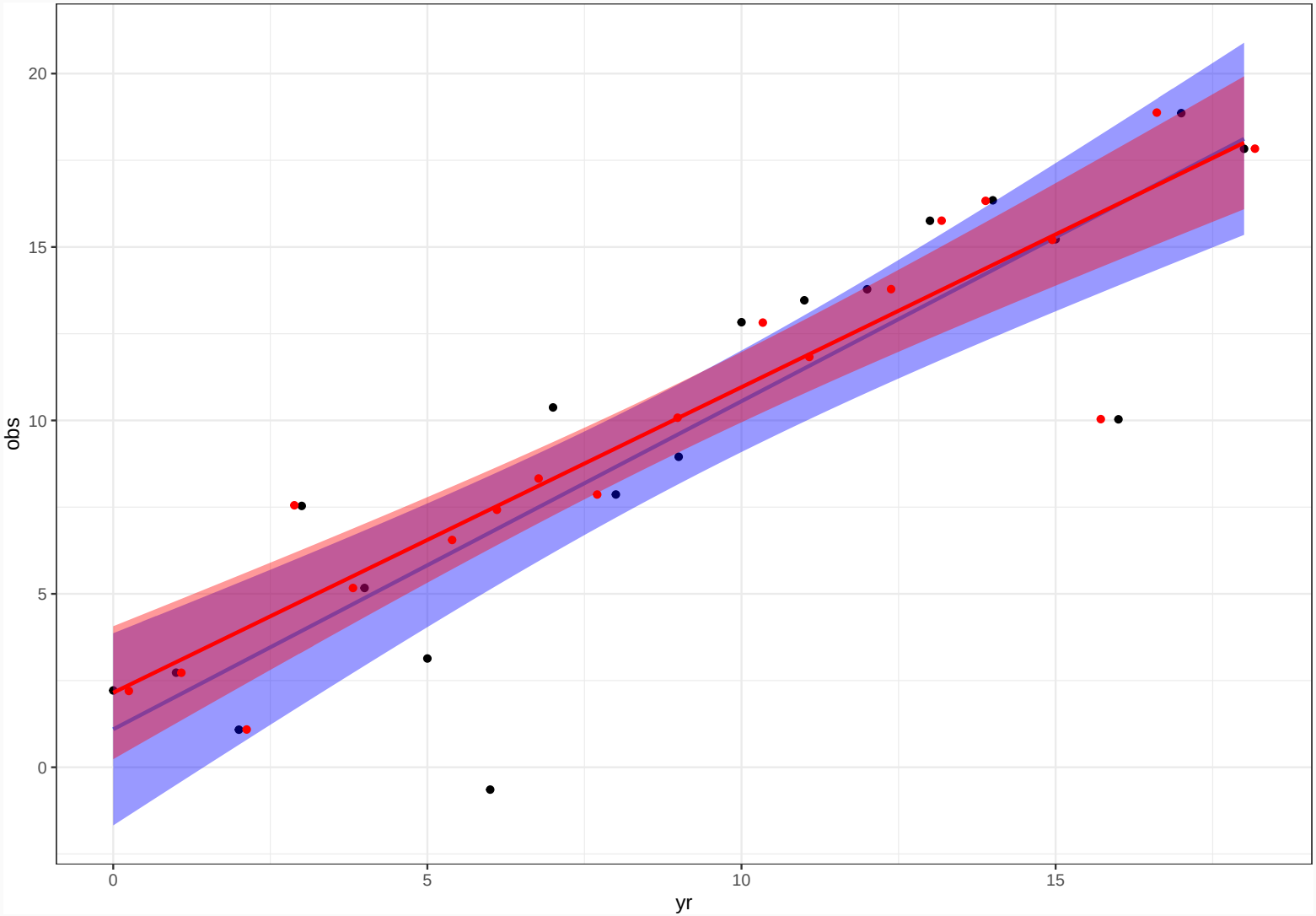
```
tidy(lm(obs~yr, data = sim_mcar))
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic    p.value
##   <chr>          <dbl>     <dbl>     <dbl>      <dbl>
## 1 (Intercept)    2.15      1.20       1.79 0.0989
## 2 yr             0.881     0.106      8.30 0.00000256
```

Single regression linear imputation

```
m1<-lm(obs~yr, data = sim_mcar)
preds<-predict(m1, newdata=sim_mcar%>%filter(is.na(obs)))
sim_reg_imp<-sim_mcar
sim_reg_imp[which(is.na(sim_reg_imp$obs)), "obs"]<-preds
```

Single regression linear imputation



Single regression linear imputation

```
tidy(lm(obs~yr, data = sim))
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic     p.value
##   <chr>          <dbl>     <dbl>     <dbl>       <dbl>
## 1 (Intercept)    1.09      1.31      0.833 0.416
## 2 yr             0.946     0.125     7.59  0.000000743
```

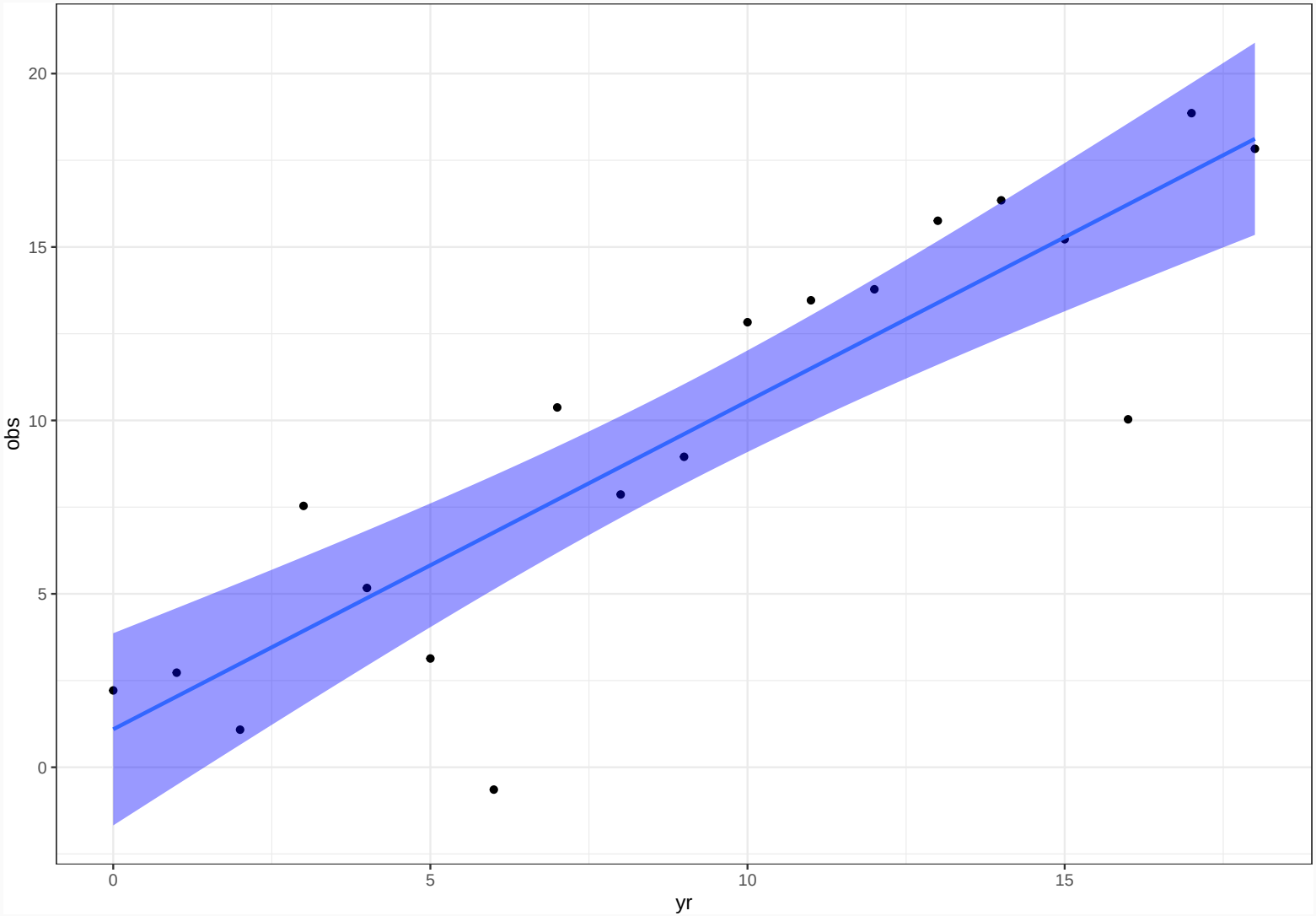
```
tidy(lm(obs~yr, data = sim_reg_imp))
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic      p.value
##   <chr>          <dbl>     <dbl>     <dbl>        <dbl>
## 1 (Intercept)    2.15     0.908       2.37 0.0301
## 2 yr             0.881    0.0862     10.2  0.0000000113
```

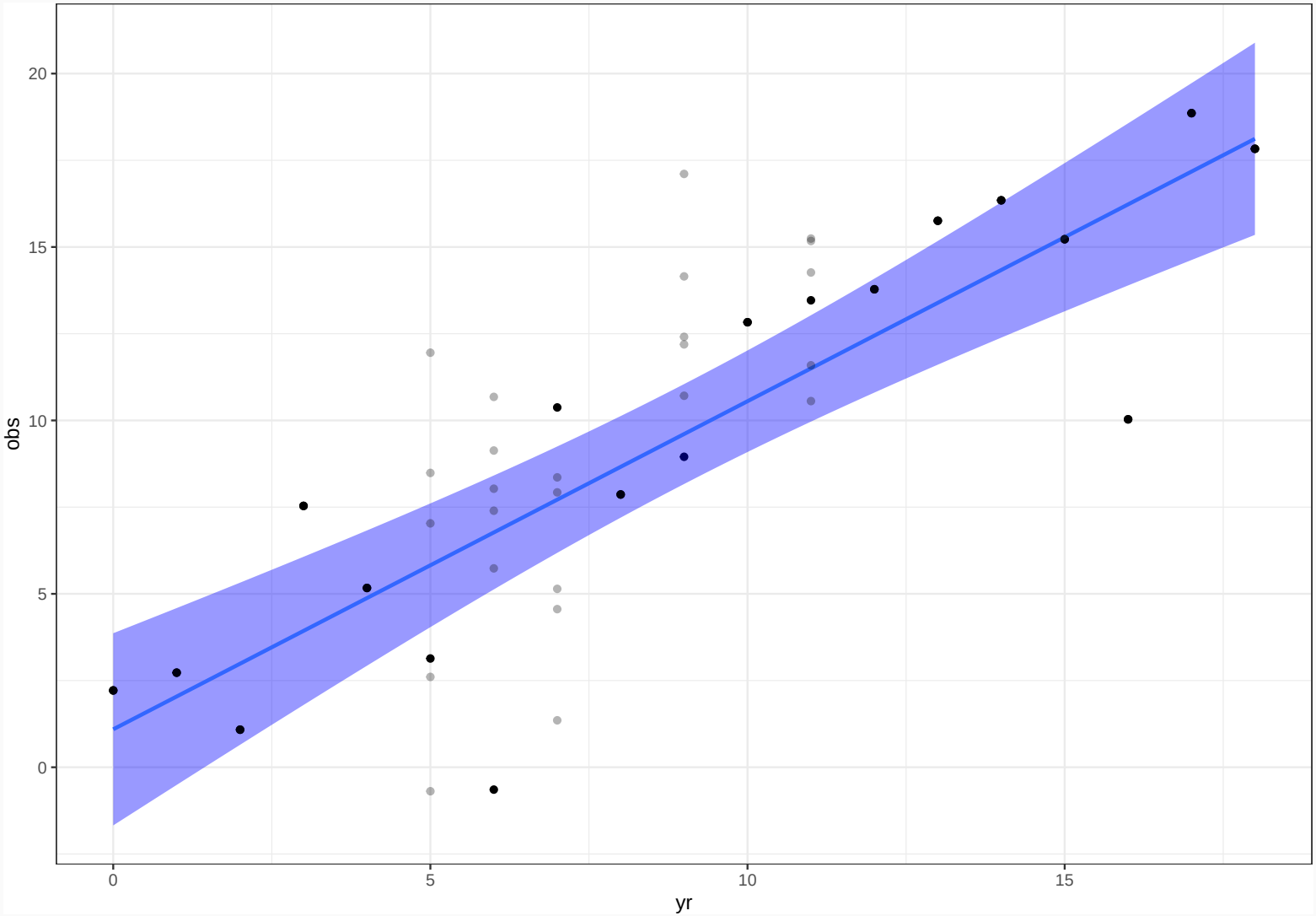
Multiple imputation

```
imps<-mice(sim_mcar, m = 5, printFlag=FALSE, method = "norm")
sims_imp<-complete(imps, include = FALSE, action = "long")
```

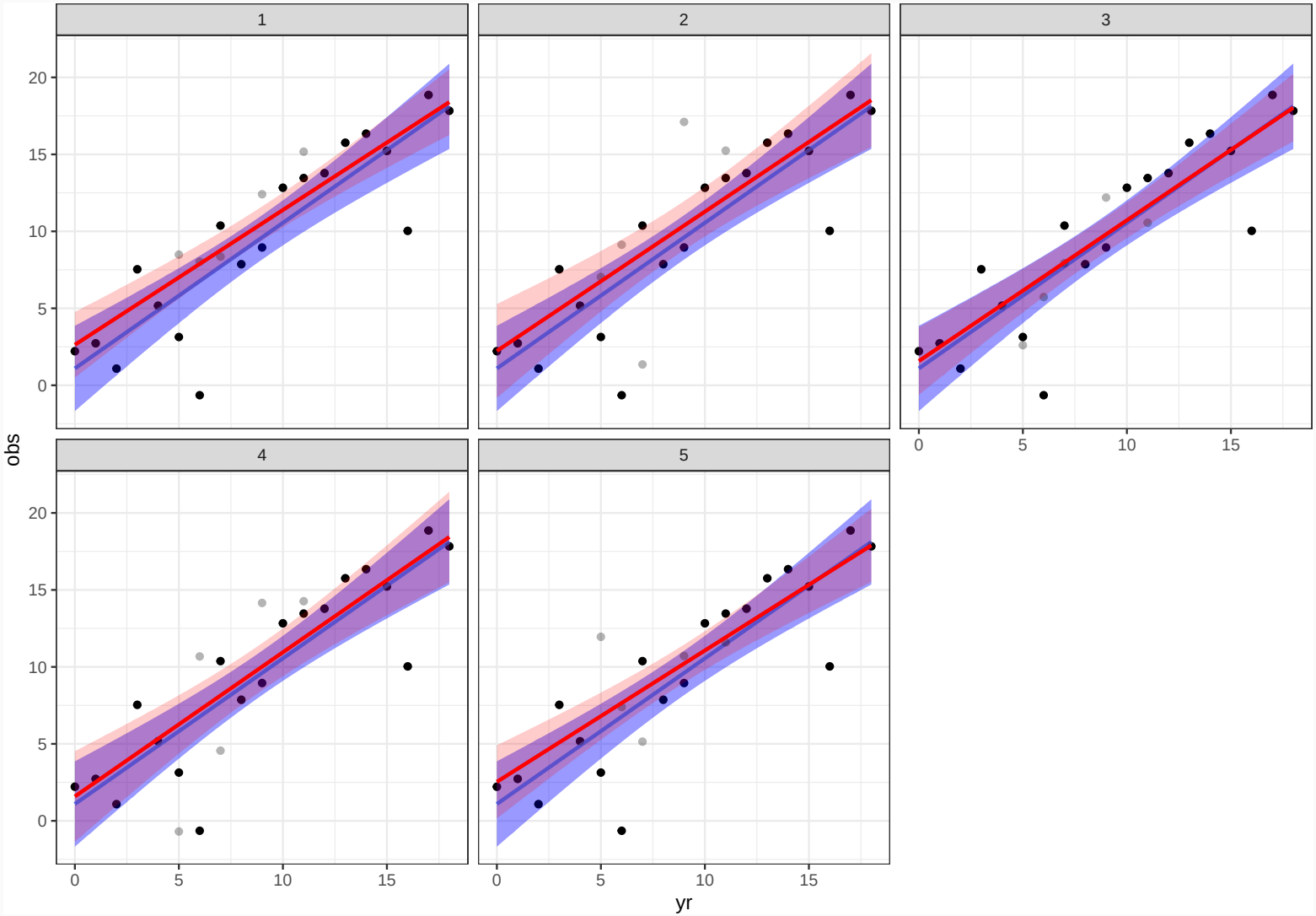
Multiple imputation



Multiple imputation



Multiple imputation



Multiple imputation: regression output

```
tidy(lm(obs~yr, data = sim))
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic     p.value
##   <chr>          <dbl>     <dbl>     <dbl>       <dbl>
## 1 (Intercept)    1.09      1.31      0.833 0.416
## 2 yr             0.946     0.125     7.59  0.000000743
```

```
summary(pool(with(data = imps, lm(obs~yr))))
```

```
##          term  estimate std.error statistic       df      p.value
## 1 (Intercept) 2.1100410 1.3381571  1.576826 11.58505 1.417343e-01
## 2          yr 0.8970177 0.1209864  7.414200 13.60059 3.897253e-06
```

Practice

Let’s first simulate some MCAR data

```
summary(mtcars$mpg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.40   15.43   19.20   20.09   22.80   33.90
```

```
## randomly delete some observations
to_delete<-sample(1:nrow(mtcars), 6, replace = F)
mtcars[to_delete, "mpg"]<-NA
###
summary(mtcars$mpg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   10.40   15.57   19.45   20.47   22.80   33.90       6
```

My preferred approach

- Understand your data!
  - Read the documentation
  - Do plenty of exploratory data analysis (cross tabs, data visuals, descriptives, look at the raw data)
  - Develop an understanding of the mechanisms of missing data in each dataset you use
  - Test your ideas for mechanisms of missing data when feasible

My preferred approach

- Use available information
    - Borrow data from other observations when possible
    - Some variables are time-stable (age) and can be borrowed from prior observations - but remember cautions against deterministic imputation and inducing bias

My preferred approach

- If MAR is a reasonable assumption (it often is), conduct multiple imputation
    - Because MAR is conditional on observables, including many variables in imputation models is often a good idea
- Apply preferred final model / analysis over each imputed dataset, combine with Rubin's rules (mice::pool), report revised estimates.

Applying missing data methods to FE: a very brief introduction

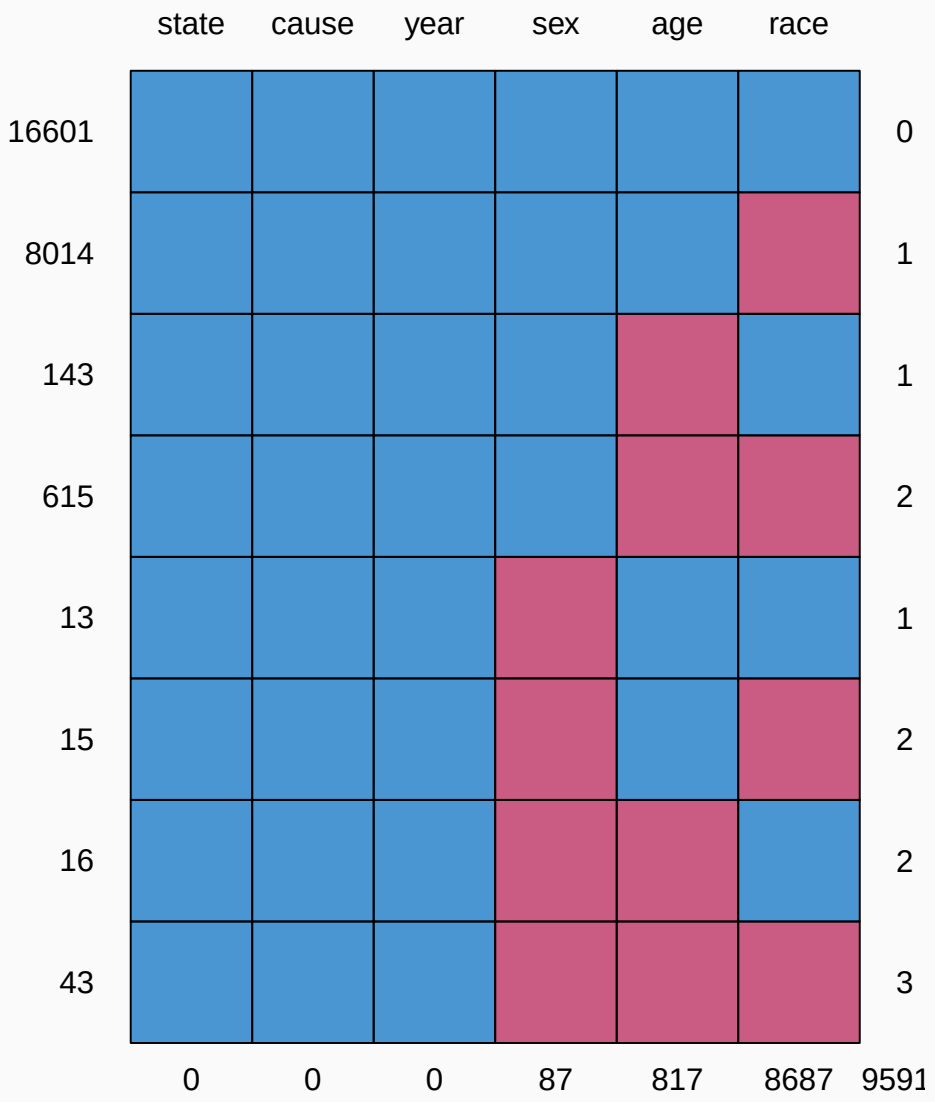
Load in packages and data

```
### load required packages
library(mice)
fe<-fe%>%
  mutate(race = ifelse(race=="Race unspecified", NA, race))
kable(fe%>%summarise_all(function(x)(sum(is.na(x)))/n()))
```

age	sex	race	state	cause	year
0.0320896	0.0034171	0.3412019	0	0	0

See missing data patters

```
md.pattern(fe)
```



Setting up our MI models

- `mice()` estimates a separate model for each variable with missing data
- In this case: age, sex, race
- What kind of model should we use to predict age? sex? race?

Setting up the models: first steps

```
## convert characters to desired type
fe<-fe%>%
  mutate(sex = factor(sex), race = factor(race), state = factor(state), cause = factor(cause))
## We can manually set the regression methods, though mice usually picks good defaults
## these go in the order of the column names, or names(fe)
## normal = OLS, logreg = Logistic, polyreg = Multinomial, pmm = Partial Mean Matching
methods<-c("norm", "logreg", "polyreg", "", "", "")
```

Setting up the models: first steps

```
## Predictor matrix
imp_init<-mice(fe, m=1, maxit=0, printFlag = FALSE)
pred_mat<-imp_init$predictorMatrix
pred_mat
```

```
##       age sex race state cause year
## age     0   1    1     1     1    1
## sex     1   0    1     1     1    1
## race    1   1    0     1     1    1
## state   1   1    1     0     1    1
## cause   1   1    1     1     0    1
## year    1   1    1     1     1    0
```

```
## 1 means that predictor (column) is included in the model for the outcome (row)
## we aren't predicting state, cause, or year. Let's set those rows to zero
pred_mat["state", ] <- 0
pred_mat["cause", ] <- 0
pred_mat["year", ] <- 0
```

Proceeding with MI

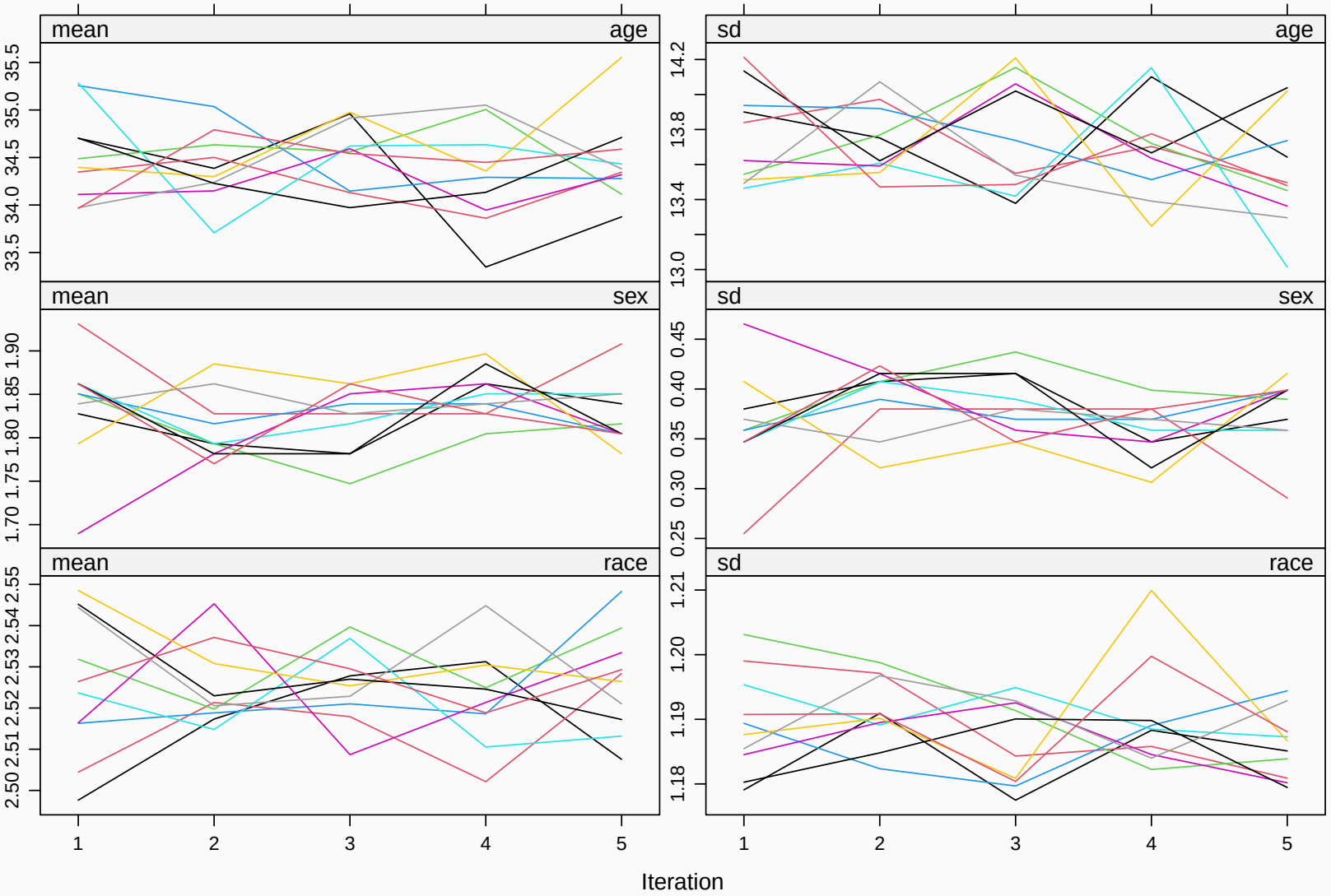
- Assumptions:
  - Age is missing at random, conditional on sex, race, state, cause, and year
  - Sex is MAR, conditional on sex, race, state, cause, and year
  - Race is MAR, conditional on sex, race, state, cause, and year
- We sequentially estimate OLS models for age, race, sex, after predicting values for other missing data in the set, then iterate. This introduces noise into the models (desirable!), and is similar to a Bayesian procedure called Markov Chain Monte Carlo estimation, where we produce estimates though random sampling.

Setting up the models

```
## rule of thumb: do 10 imputations. 5 is ok when there isn't much missing data.
## Use m>10 when there is a high volume of missing data
## set a random seed so your results will be consistent when you re-run it
set.seed(43)
imps_fe<-mice(fe, m = 10, method = methods, predictorMatrix = pred_mat)
```

```
##
##  iter imp variable
##   1   1  age  sex  race
##   1   2  age  sex  race
##   1   3  age  sex  race
##   1   4  age  sex  race
##   1   5  age  sex  race
##   1   6  age  sex  race
##   1   7  age  sex  race
##   1   8  age  sex  race
##   1   9  age  sex  race
##   1   10  age  sex  race
##   2   1  age  sex  race
##   2   2  age  sex  race
##   2   3  age  sex  race
##   2   4  age  sex  race
##   2   5  age  sex  race
##   2   6  age  sex  race
##   2   7  age  sex  race
##   2   8  age  sex  race
##   2   9  age  sex  race
##   2   10  age  sex  race
##   3   1  age  sex  race
```

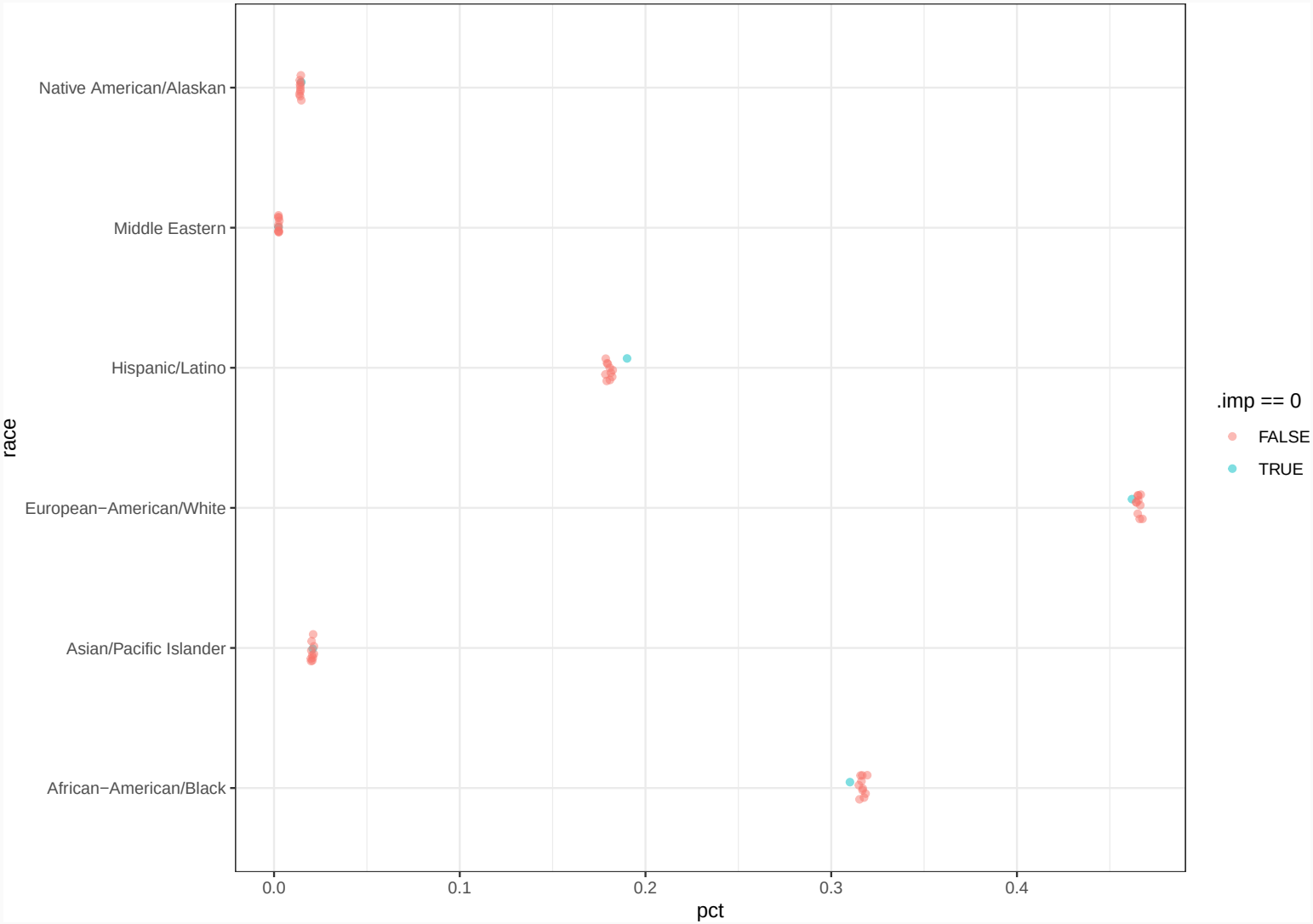
Check out convergence - We don’t want to see obvious trends here



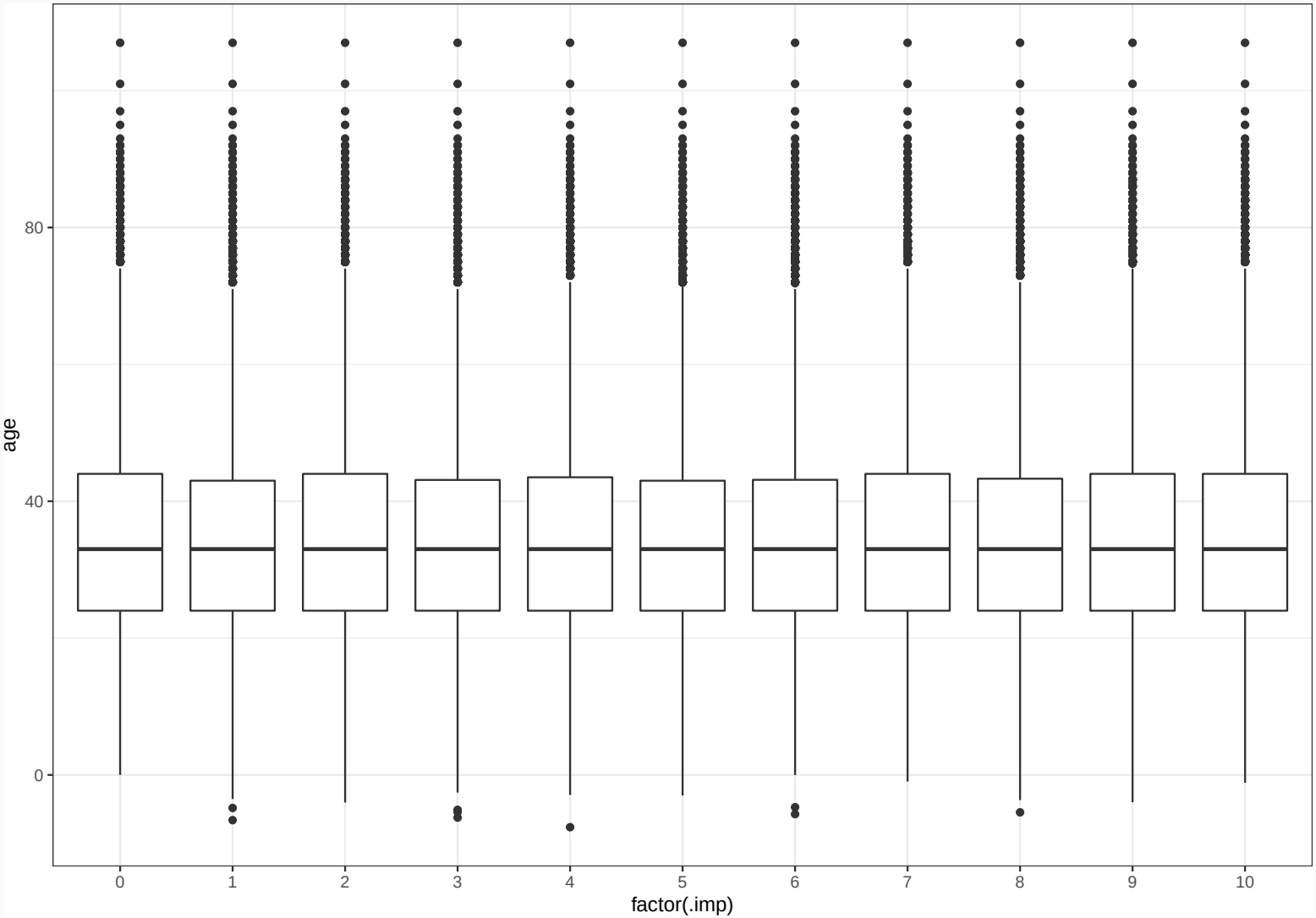
Check out effects of imputation

```
imputed<-complete(imps_fe, action = "long", include = TRUE)
imputed_race_check<-imputed%>%
  filter(!(is.na(race)))%>%
  group_by(race, .imp)%>%
  summarise(count = n())%>% ### make counts of race by imputation
  left_join(imputed%>% ### join to total case count by imputation (remember we dropped NAs from observed here)
              filter(!(is.na(race)))%>%
              group_by(.imp)%>%
              summarise(total = n()))%>%
  mutate(pct = count / total)
```

Check out effects of imputation



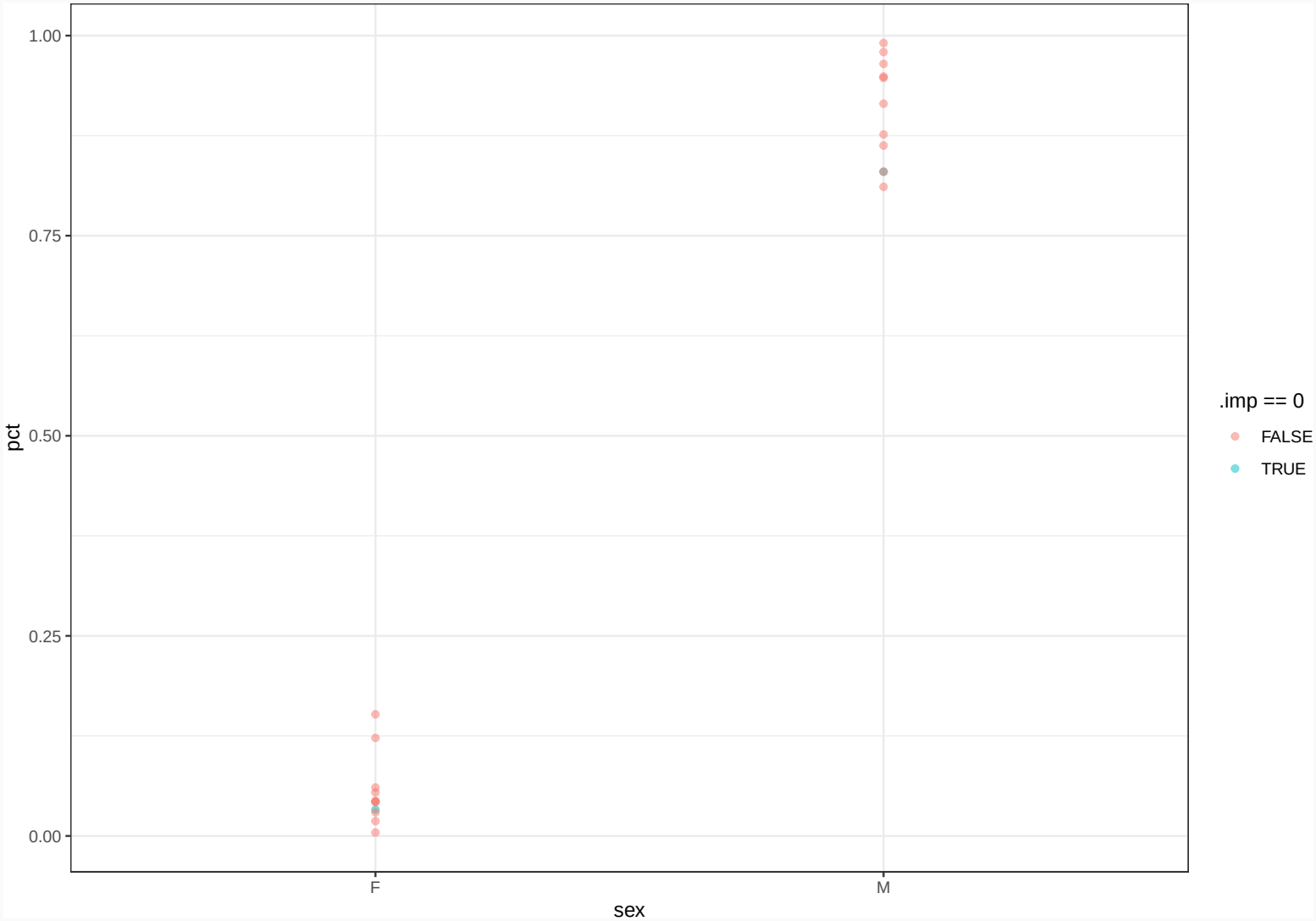
Compare imputed to original data



Compare imputed to original data

```
imputed_sex_check<-imputed%>%
  filter(!(is.na(sex)))%>%
  group_by(sex, .imp)%>%
  summarise(count = n())%>%
  left_join(imputed%>%
              filter(!(is.na(sex)))%>%
              group_by(.imp)%>%
              summarise(total= n()))%>%
  mutate(pct = count / total)
```

Compare imputed to original data



Procedure for regression modeling

- Explore your data and determine if data are MCAR, MAR, or MNAR
- Build imputation models if appropriate
- Evaluate convergence and effects of imputation models: adjust if needed
- Fit desired regression model on each imputed dataset

Using MI data for regression: estimating β by hand

Don't worry, there's an automatic way too

Rubin's rules for combination of parameter estimates

$$\bar{\beta} = \frac{1}{m}\left(\sum_{i=1}^{m} \beta_i\right)$$

or in R `mean(beta)`

Estimating standard errors by hand

This is where it gets tricky. We need to account for variance both across and within imputations.

Within imputation variance is simply the average of the variance across imputations, or `mean(SE^2)`. We'll call this var_w

Between imputation variance is a little more complex.

$$var_b = \frac{1}{m} \sum_{i=1}^{m} (\beta_i - \bar{\beta})$$

We can provide the pooled standard error as $var(\bar{\beta}) = var_w + var_b$

Conduct a pooled analysis: the easy way

```
### predict victim sex by age, race
fit_imp<-with(imps_fe, glm(sex ~ age + race ,
                           family = "binomial"))
## Pool results with Rubin's rules
pooled<-pool(fit_imp)
### just with observed data
fit_obs<-glm(sex ~ age + race ,
                           family = "binomial", data = fe)
```

Final notes

- Approaches to missing data are not one-size fits all.
- Think hard about why your data are missing
- If they are MAR conditional on observables, MI may be appropriate

Further reading

- Rubin, “Multiple Imputation for Nonresponse in Surveys”
- Gelman and Hill, “Data Analysis Using Regression and Multilevel/Hierarchical Models”
- van Buuren, “mice: Multivariate Imputation by Chained Equations in R”

Thanks for a great semester!