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 Poisson model

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

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

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

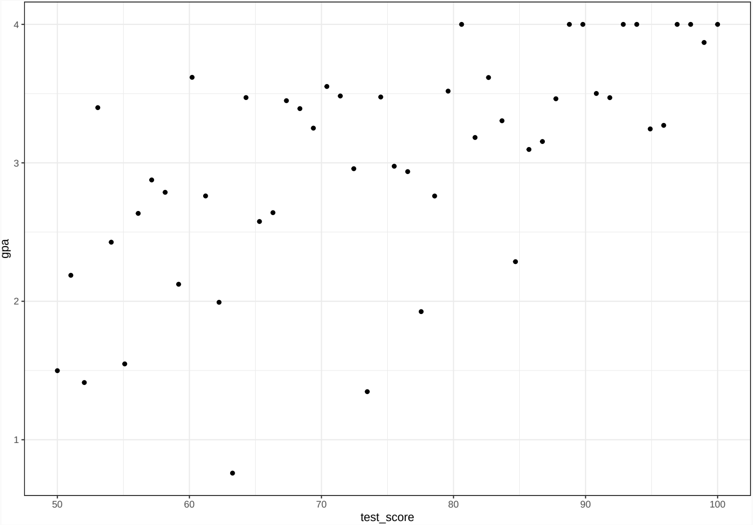
In R: `glm(y~x, family = poisson)`

Missing data

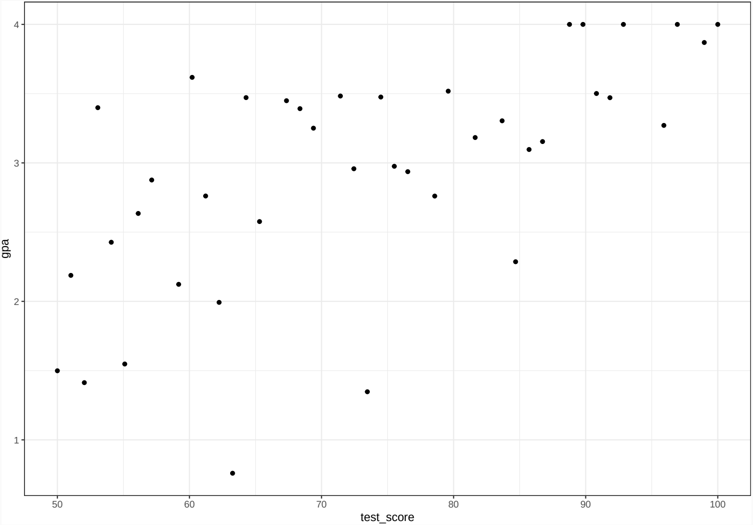
Why should we care?

- 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

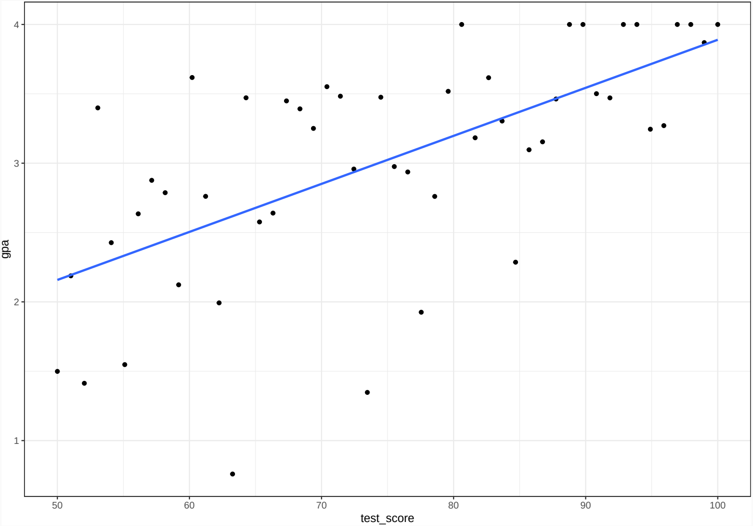
A hypothetical with missing data: predicting student GPA from a math test



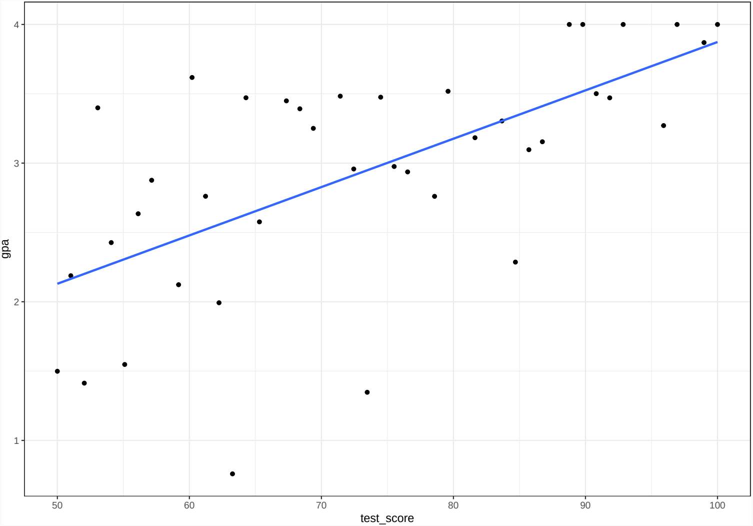
Missing observations



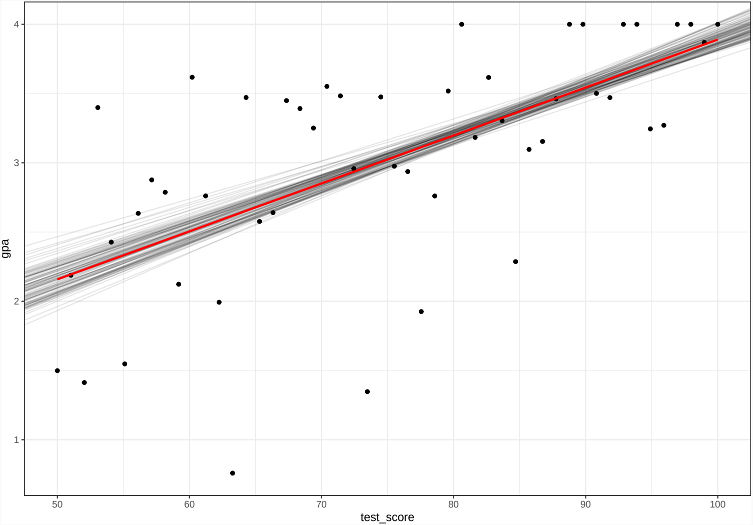
Best fit line under complete data



Best fit line under missing data



100 hypothetical lines with different sets of 10 cases missing completely at random



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

MCAR results in unbiased Beta estimates, but increases standard errors and uncertainty

```
### true values
tidy(lm(gpa ~ test_score, data = sim))
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic     p.value
##   <chr>          <dbl>     <dbl>     <dbl>       <dbl>
## 1 (Intercept)   0.427    0.462       0.924 0.360
## 2 test_score    0.0346   0.00604     5.73  0.000000642
```

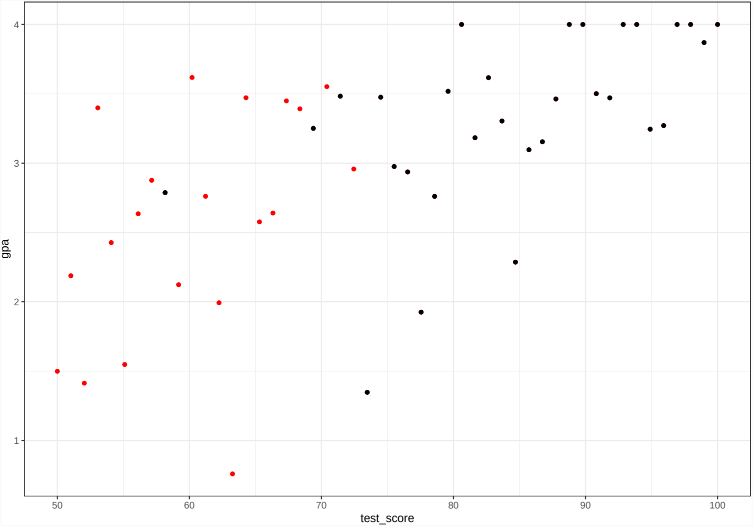
```
### average parameter estimates for 100 simulations with missing data
lines %>%
    summarize(mean_intercept = mean(intercept), mean_slope = mean(slope), sd_intercept = sd(intercept),
        sd_slope = sd(slope))
```

```
##   mean_intercept mean_slope sd_intercept    sd_slope
## 1      0.4515659 0.03437496    0.2383352 0.002782726
```

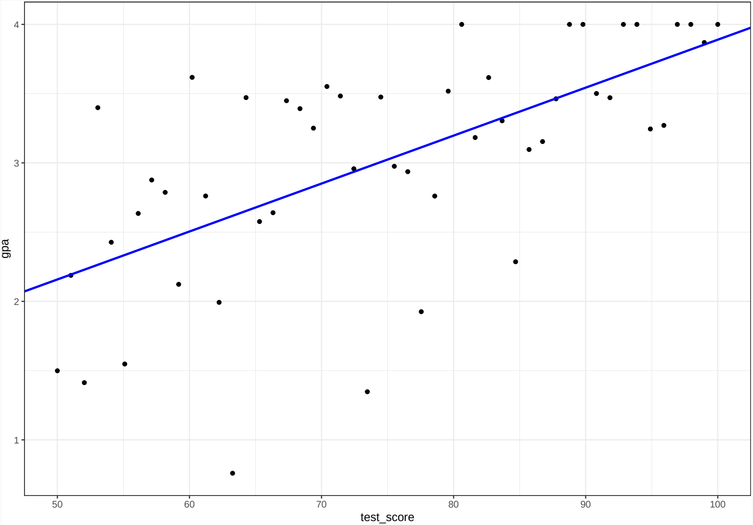
Three general causes of missing data: Missing at random

- **Missing at random (MAR)**: The probability of a value being missing is *not* completely at random (I know...) ...
- 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

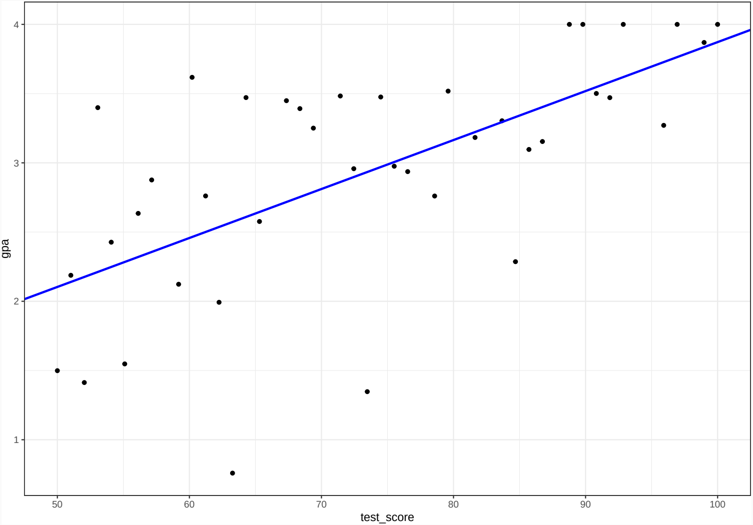
What if students with low GPAs were more likely to miss school on test day?



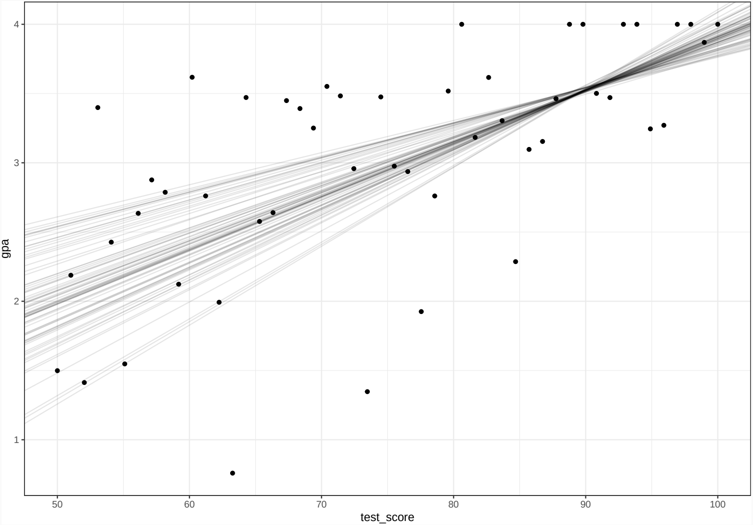
Best fit line under complete data



Best fit line under missing data

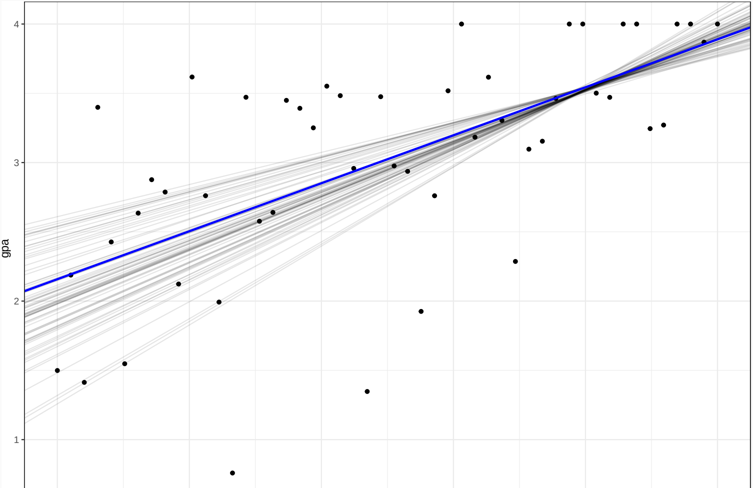


100 hypothetical lines with different sets of 10 cases missing completely at random



And with the true regression line

```
ggplot(sim, aes(x = test_score, y = gpa)) + geom_point() + geom_abl
    aes(intercept = intercept, slope = slope), alpha = 0.1) + geom_
    slope = coef(lm_temp)[2]), color = ”blue”, size = 1)
```



What’s going on?

The probability of data being missing is conditional on GPA. If we ignore the missing data, then we will systematically understate the relationship between test score and GPA. …

$$P(missing) \neq P(missing|GPA)$$

Results of regression models (spot the bias!)

```r
### true values
tidy(lm(gpa ~ test_score, data = sim))
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic     p.value
##   <chr>          <dbl>     <dbl>     <dbl>       <dbl>
## 1 (Intercept)   0.427    0.462       0.924 0.360
## 2 test_score    0.0346   0.00604     5.73  0.000000642
```

```r
### average parameter estimates for 100 simulations with missing da
lines %>%
    summarize(mean_intercept = mean(intercept), mean_slope = mean(s
        sd_slope = sd(slope))
```

```
##   mean_intercept mean_slope sd_intercept    sd_slope
## 1      0.1492993 0.03753875    0.6485505 0.007202301
```

Three general causes of missing data: non-random missingness

- **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 may opt-out of reporting their data to the federal government; police departments with high levels of use-of-force opt-out of reporting to federal arrest-related-deaths programs; people who do not vaccinate their children opt-out of answering a survey question about vaccination
- 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)

So what can we do?

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 on known or stable variables (e.g. imputing age based on information 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.

Basic approaches to missing data, stochastic

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

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 and we have enough data, MI can correct bias in parameter estimates

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.

Lab: practice with some simple data

With simple data

```r
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

Combination post-imputation

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(se
# race , family = 'binomial')) ## Pool results with Rubin's rules
# pooled<-pool(fit_imp) ### just with observed data fit_obs<-glm(se
# race , family = 'binomial', data = fe)
```