GOOD, FAST, CHEAP: How to do Data Science with Missing Data

https://github.com/josephofiowa/missing-data-workshop-odsc19

Note

- I recommend cloning the repo to your computer, then opening the **`interactive-plot`** and **`item-nonresponse-demo`** Jupyter notebooks.

- **Run the first cell in both notebooks.** This will ensure that all of your imports are good to go.
  - You may want to uncomment the first line in the **`interactive-plot`** notebook to ensure your packages are up to date.

- If you're just following along without a laptop or without Python, that's OK! You should still be able to follow along.

Joseph Nelson @josephofiowa

🤖 Cofounder, Roboflow.ai

Build computer vision models easily. Collect, organize, and preprocess your image data.

📈 Managing Partner, BetaVector.com

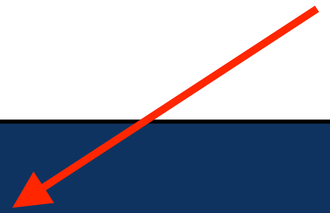
Advise, educate, and build all things data. Founded this consultancy as rocauc.com.

✏️ Distinguished Faculty, GeneralAssemb.ly

Empower others to use data science. Taught over 2000+ hours.

Previously: cofounded and sold Represently.com, enabling the US Congress to respond 90% faster to constituents writing them. Data products at Facebook. PR work at agencies and the US Senate.

An important thanks: Matt Brems, BetaVector partner; GA instructor.



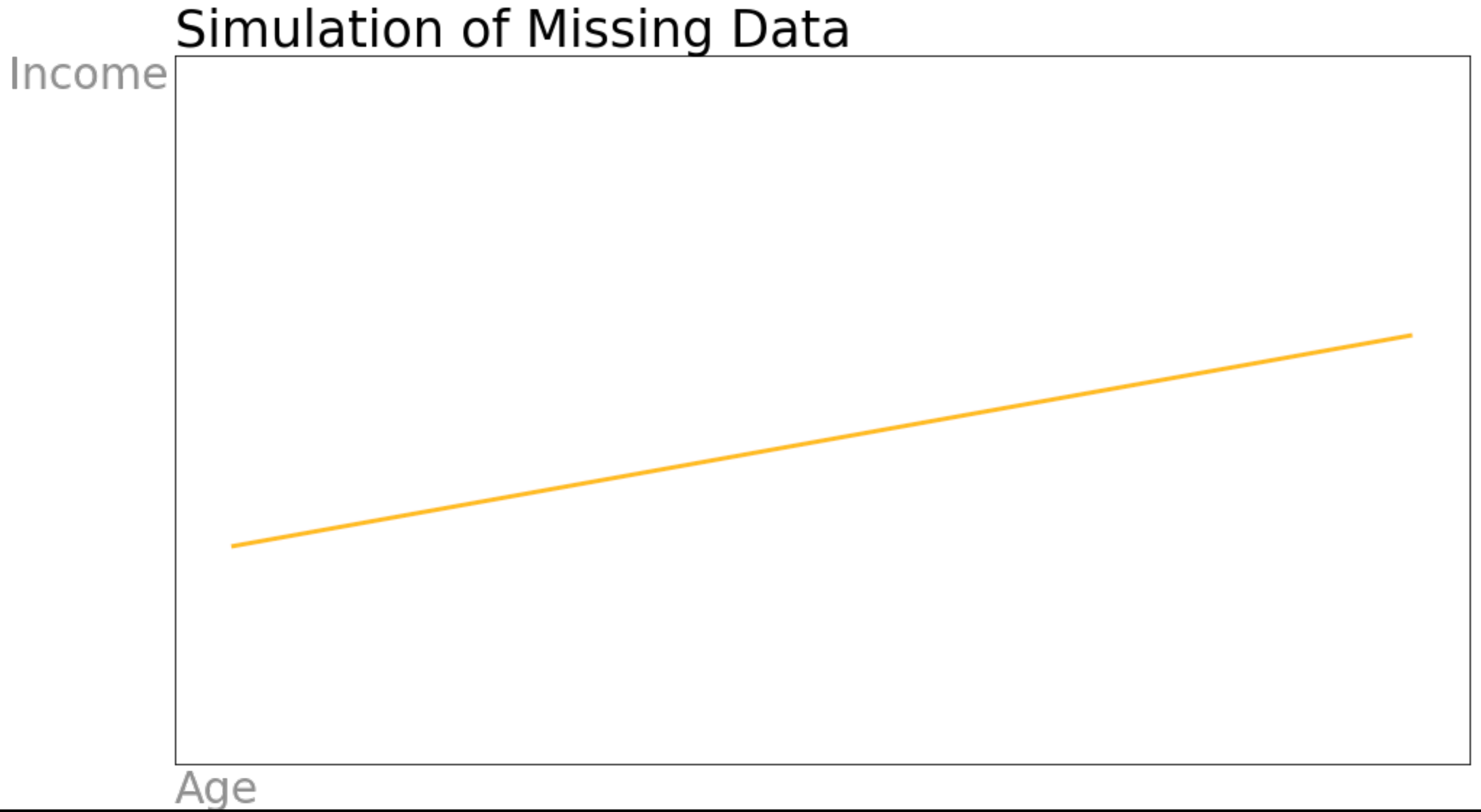
Agenda

1. **How big of a problem is missing data?**
2. The three types of missingness.
3. Three strategies for tackling missing data.
4. Practical recommendations.

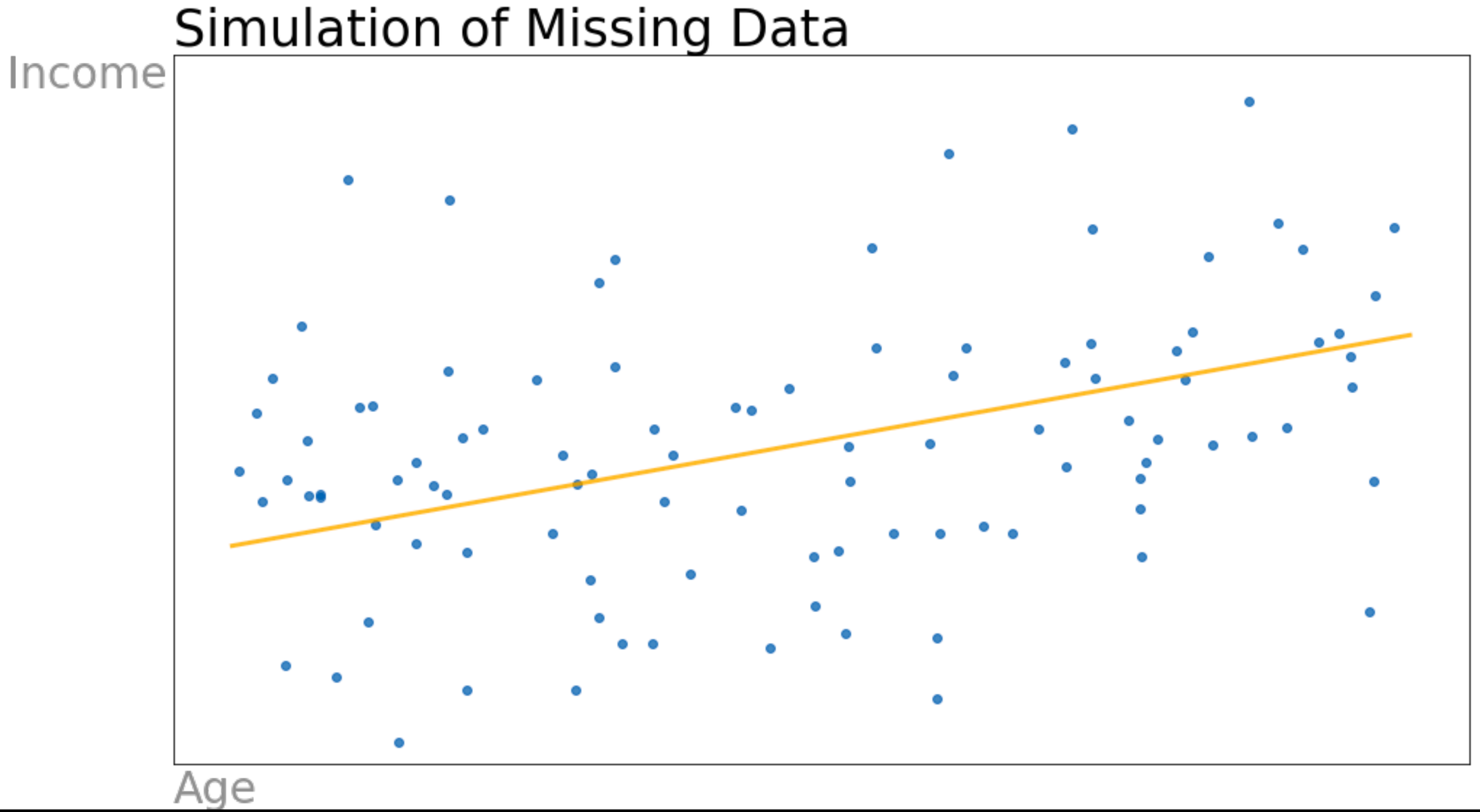
How big of a problem is missing data?

- This is a difficult question to answer.
- Practically, we can only see what we observe.
- We generally won't definitively know the unobserved values.
- **We can use simulated data to help answer this question.**

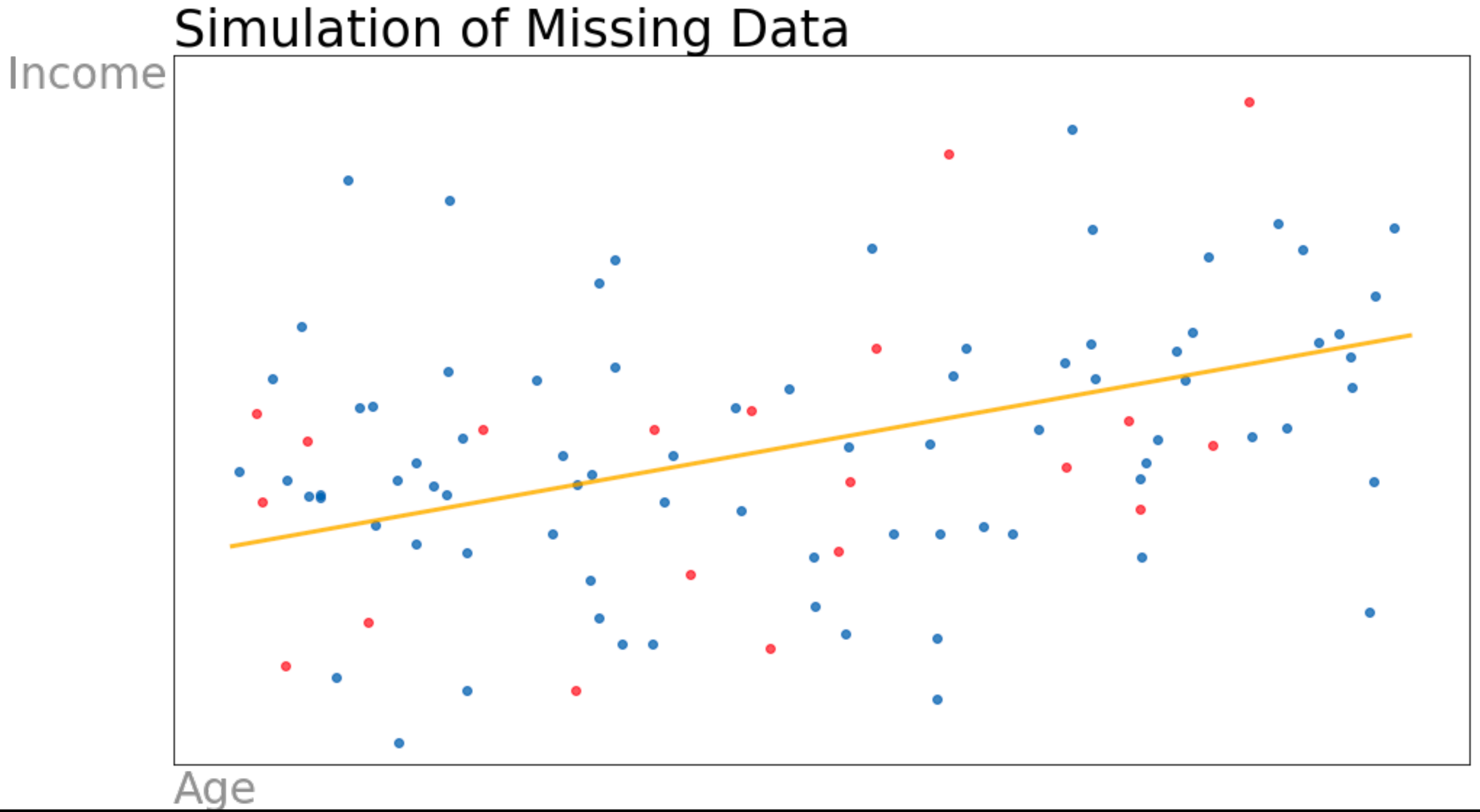
How big of a problem is missing data?



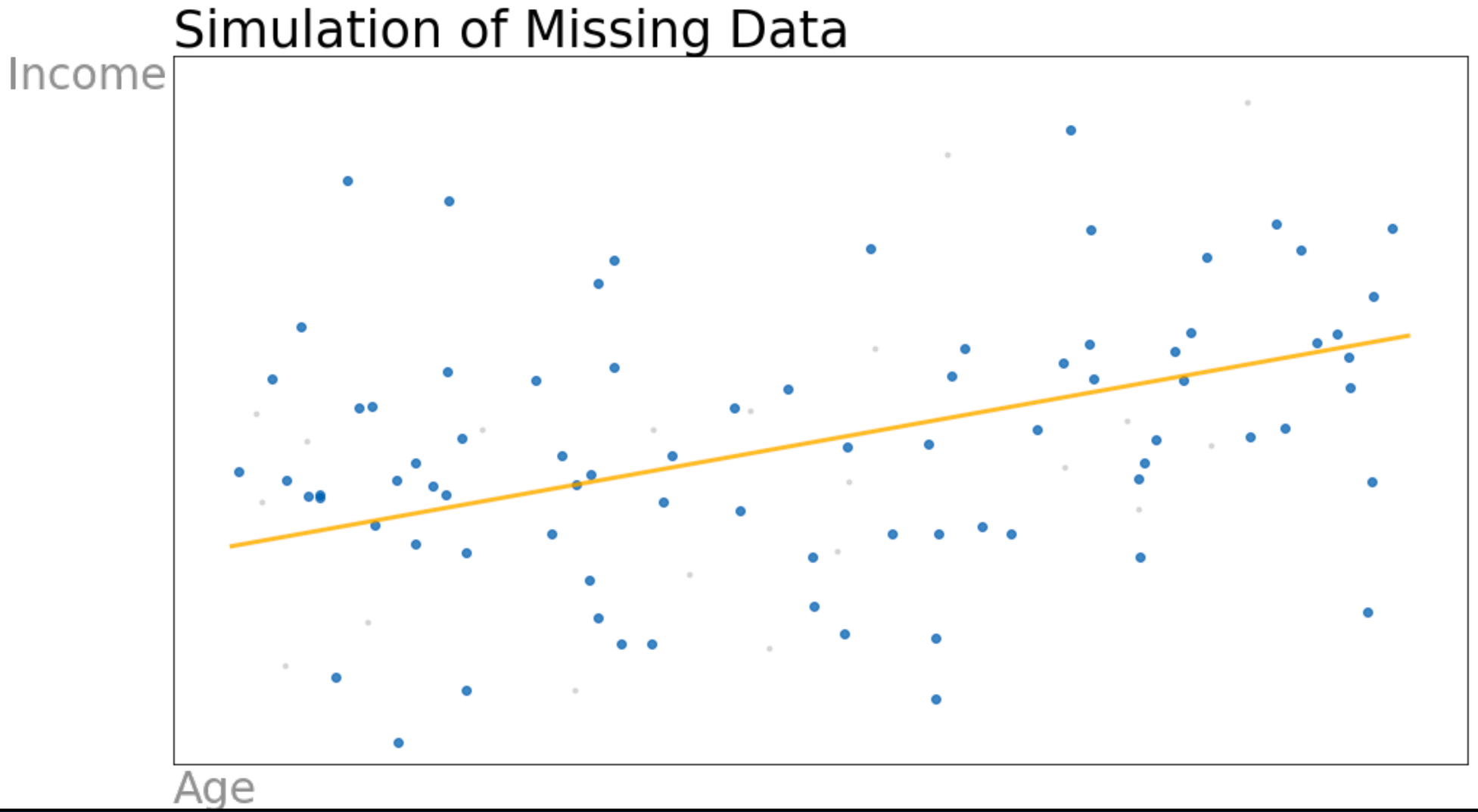
How big of a problem is missing data?



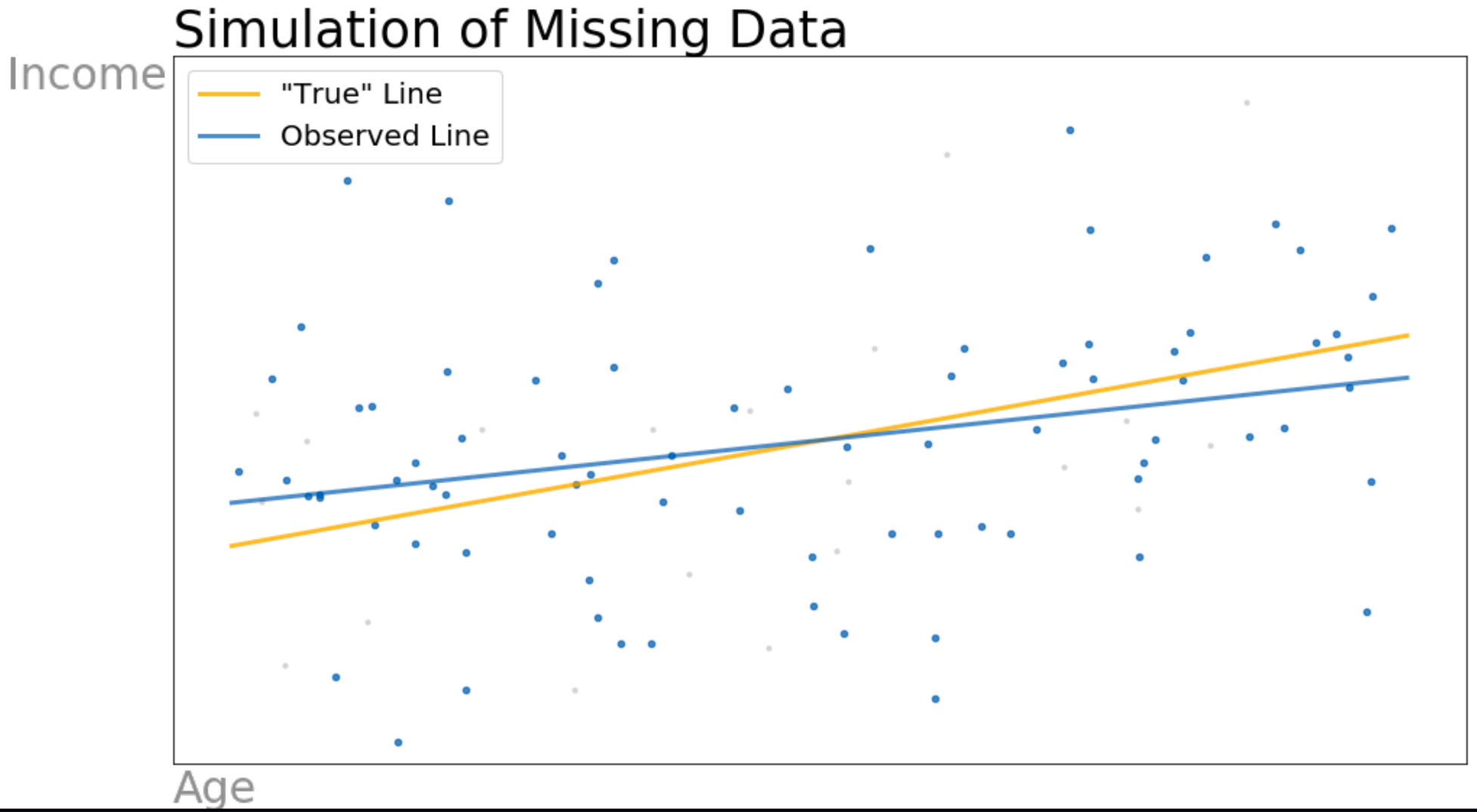
How big of a problem is missing data?



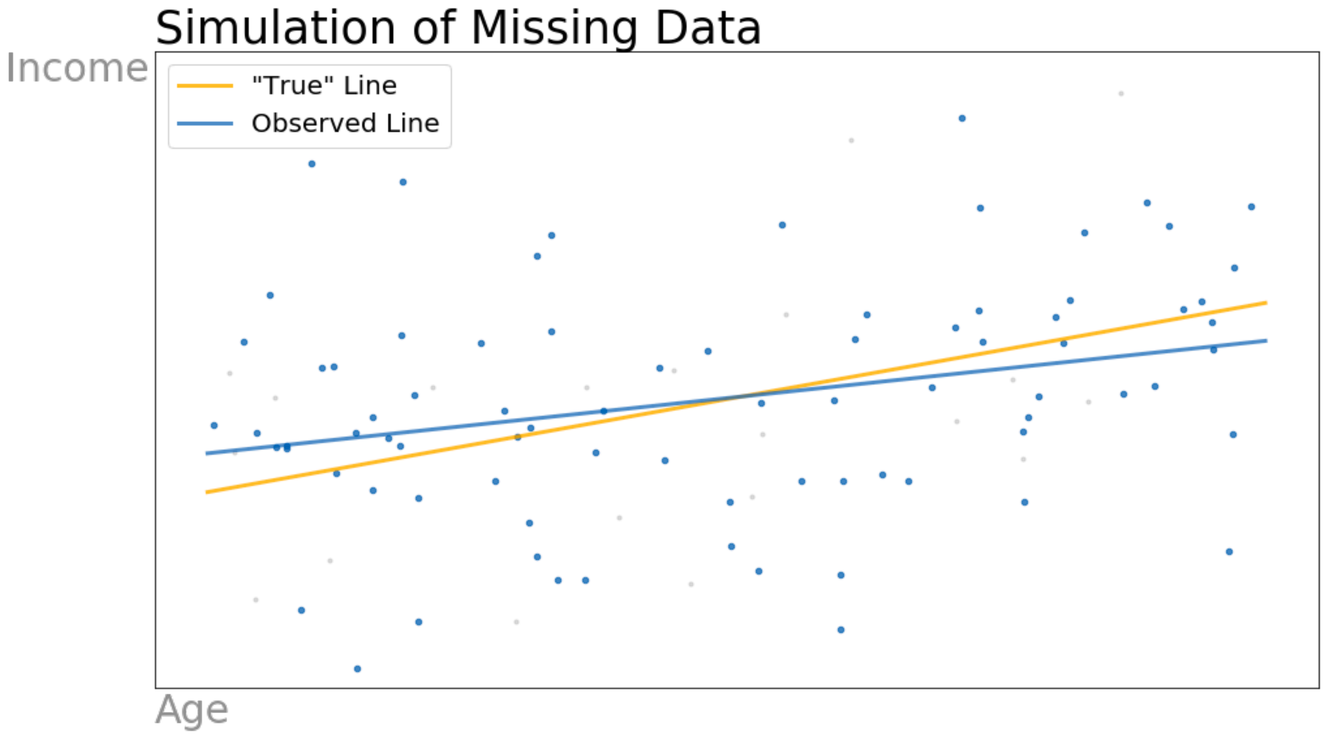
How big of a problem is missing data?



How big of a problem is missing data?



How big of a problem is missing data?



- 20% of observations are missing completely at random.

- True slope: 750
- Observed slope: 445
- Percent change: 40.7%

- True y-intercept*: $15{,}000$
- Observed y-intercept*: $27{,}229$
- Percent change: 81.5%

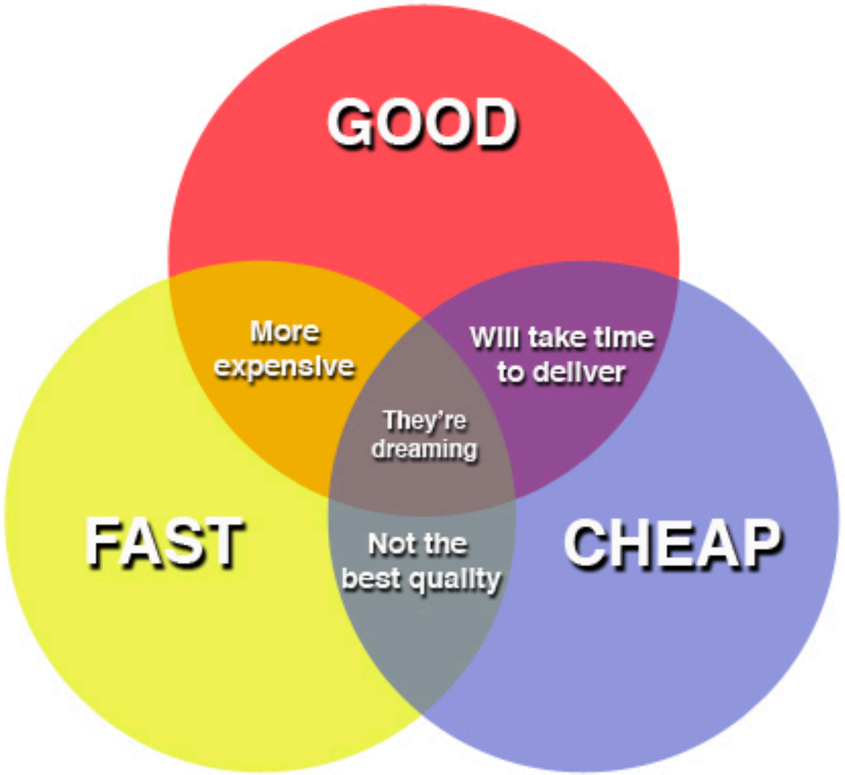
Catastrophe.

How big of a problem is missing data?

- In the last example, “reality” was that income is linearly related to age and nothing else. **Reality is usually much more complex!**
- In the last example, we assumed no measurement error. **Reality doesn’t always follow this.**
- In the last example, we could quantify difference between the “truth” and what we observe. **In reality, we cannot do this.**
- **Missing data can substantially undermine our analyses and inferences!**

Good, Fast, Cheap

Good, Fast, Cheap



- **Fast and Cheap Analysis:** Drop all missing values or do rudimentary imputation.
- **Good and Cheap Analysis:** Proper imputation.
- **Good and Fast Analysis:** Gather data in a complete manner.

Agenda

1. How big of a problem is missing data?
2. **The three types of missingness.**
3. Three strategies for tackling missing data.
4. Practical recommendations.

The three types of missingness

- Whenever you see an “NA” or a blank in some data, that generally implies that you have missing data.

- Although they look the same, **there are three different “types” of missing data**.
  - Certain tools will only be valid for certain types of missingness!

Scenario 1: MCAR

- I am a sleepy graduate student working in a lab. While pipetting, I reach over to grab my pen but I accidentally knock three Petri dishes off of the desk. From these Petri dishes, I lose all of the data that I otherwise would have collected.

- This type of missingness is called **missing completely at random**.
  - The data of interest is not systematically different between respondents and nonrespondents.

Scenario 2: MAR

- I work in a lab that contains remote sensors. One sensor broke and thus did not gather information from 6:00 a.m. until 10:00 a.m.

- This type of missingness is called **missing at random** (conditional on time).
  - **Conditional on data we have observed**, the data of interest is not systematically different between respondents and nonrespondents.
  - In this case, accounting for time can account for the missingness!

Scenario 3: NMAR

- I administer a survey that includes a question about income. Those who have lower incomes are less likely to respond to the question about income.

- This type of missingness is called **not missing at random**.
  - The data of interest are systematically different for respondents and nonrespondents.
  - Whether or not an observation is missing depends on the value of the unobserved data itself!

Agenda

1. How big of a problem is missing data?
2. The three types of missingness.
3. **Three strategies for tackling missing data.**
4. Practical recommendations.

Before “tackling” missing data, let’s shift our mindset.

- There is this naïve belief that we can just plug in the gaps in our data.
  - This is a method known as **imputation**.
  - We have to do this in a very specific way… otherwise we’re just making up data.

- Rather than using the phrase “missing data analysis” or “fixing” missing data, it’s more correct to use the phrase “data science with missing data.”

- **In most cases, we aren’t “fixing” the missing data. We’re just learning how to cope with it!**

Methods of Working with Missing Data

1. We can **avoid** it.

2. We can **ignore** it.

3. We can **account** for it.

Why talk about avoiding missingness?

- Avoiding missing data altogether allows us to reduce our uncertainty.
- It's usually cheaper for us to spend time avoiding missing data than to make guesses about the best way to "fill" it in.
- If we can avoid missing data, we make what comes afterward so much easier.

Unit Nonresponse vs. Item Nonresponse

- **Unit nonresponse** is where no values from an observation are observed.
  - Index 3

- **Item nonresponse** is where some, but not all, values from an observation are observed.
  - Indices 1, 2, and 10,000

Index	Age	Sex	Income
1	NA	M	NA
2	39	NA	75000
3	NA	NA	NA
4	28	F	50000
...	...	...	...
10000	18	F	NA

Unit Nonresponse vs. Item Nonresponse

- **Unit Nonresponse**
  - Avoid it.
  - Ignore it.
  - Account for it.

- Item Nonresponse
  - Avoid it.
  - Ignore it.
  - Account for it.

How do we avoid unit nonresponse?

- Decrease the burden on your respondent.
- Change your method of data collection.
- Improve accessibility.
- Consider changing the timing of your survey.

How do we ignore unit nonresponse?

- This is straightforward (**and probably what most of us do anyway**).

- We assume that our respondents is approximately the same as the respondents and nonrespondents.

- You may feel comfortable making this assumption if the percent of nonrespondents is low… **but be careful about what this implies**!
  - Is nonresponse tied to something that can influence what you're studying? For example, did you call people at home between 9 a.m. and 5 p.m.?

How do we account for unit nonresponse?

-
- The most common method of accounting for unit nonresponse is to do weight class adjustments.
- Take the full sample (respondents and nonrespondents) and break them into strata based on characteristics we know.
  - Age, sex, department, etc.
  - Reweight all respondents by $\frac{true\ proportion}{proportion\ of\ responses}$.

Weight Class Adjustments

- I'm estimating job satisfaction among two departments: accounting and finance.
  - These two departments are the same size.
  - 25% of my responses came from finance and 75% from accounting.
    - $w_{finance} = \frac{true\ proportion}{proportion\ of\ responses} = \frac{0.50}{0.25} = 2$
    - $w_{accounting} = \frac{true\ proportion}{proportion\ of\ responses} = \frac{0.50}{0.75} = \frac{2}{3}$
- Python: Most `sklearn` methods have a `weights` parameter to be used when fitting machine learning models.

Caution: Weight Class Adjustments

- In order to use these weight class adjustments, we need to know what the true population distribution is.

- Depending on the type of information you have, this can be unrealistic.
  - What percentage of voters in the 2018 election are 18-34, 35-54, and 55+?

- The variance of our estimates may rise dramatically when doing weight class adjustments.

Unit Nonresponse vs. Item Nonresponse

- Unit Nonresponse
  - Avoid it.
  - Ignore it.
  - Account for it.

- **Item Nonresponse**
  - Avoid it.
  - Ignore it.
  - Account for it.

How do we avoid item nonresponse?

- Design the questionnaire with the respondent in mind.
- Minimize the length of the questionnaire.
- Consider the content of your survey.

How do we ignore item nonresponse?

- **Complete-Case Analysis**
  - Drops any observation with **any** missing value.
    - Pros: Results will be well-behaved, simplest, often software default.
    - Cons: Drops some collected data, loses “information” and precision.

- **Available-Case Analysis**
  - Drops no observations and calculates results based on available data.
    - Pros: Uses all data available.
    - Cons: Can get “not well-behaved results.” (Correlations above 1.)

How do we account for item nonresponse?

- Deductive Imputation
- Inferential Imputation
  - Mean/Median/Mode Imputation
  - Regression Imputation
  - Stochastic Regression Imputation
  - Hot-Deck Imputation
  - Proper Imputation
- Pattern Submodel Approach

Deductive Imputation

- We use logical relationships to fill in missing values.
  - Respondent says they were not the victim of a crime, but left “victim of a violent crime” blank.
  - If someone has 2 children in year 1, NA children in year 2, and 2 children in year 3, we can **probably** impute that they have 2 children in year 2.

- **Pros**: Requires minimal “inference,” valid method.
- **Cons**: Requires specific coding, can be time consuming.

Mean/Median/Mode Imputation

- For any “NA” value in a given column, replace ”NA” with the mean (or median or mode).

- **Pros**: Very easy to implement and comprehend.
- **Cons**: Significantly distorts histogram, underestimates variance, invalid method.

Regression Imputation

- For any “NA” value in a given column, replace ”NA” with a value predicted from a regression line.
  - Suppose I have an observation where income is missing, but age and sex are not.
  - Build a linear regression model predicting income from age and sex.
  - Predict the missing income value using $\text{income} = \hat{\beta}_0 + \hat{\beta}_1 \text{age} + \hat{\beta}_2 \text{sex}$.

- **Pros**: An improvement over mean imputation, easy to comprehend.
- **Cons**: Distorts histogram and underestimates variance, invalid method.

Regression Imputation

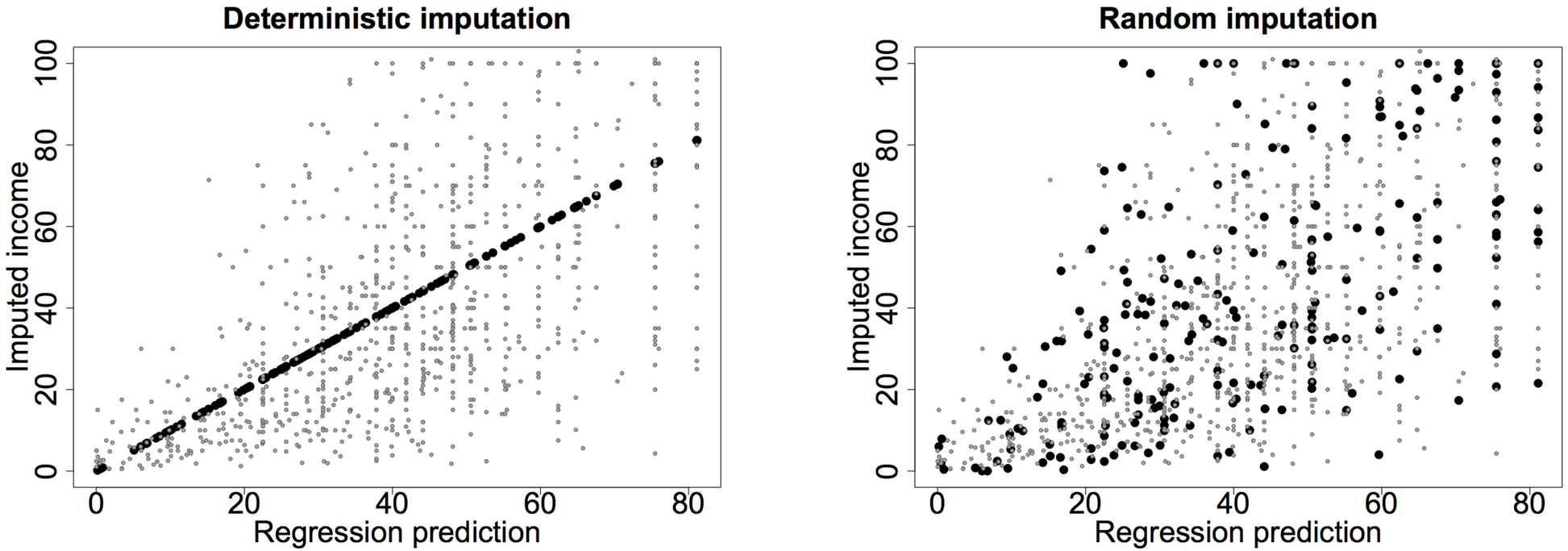


Figure 25.2 *Deterministic and random imputations for the 241 missing values of earnings in the Social Indicators Survey. The deterministic imputations are exactly at the regression predictions and ignore predictive uncertainty. In contrast, the random imputations are more variable and better capture the range of earnings in the data. See also Figure 25.1.*

Source: Andrew Gelman & Jennifer Hill, Data Analysis using Regression and Multi-level/Hierarchical *Models*: http://www.stat.columbia.edu/~gelman/arm/missing.pdf

Stochastic Regression Imputation

- For any “NA” value in a given column, replace ”NA” with a value predicted from a regression line.
  - Predict the missing income value using $\text{income} = \hat{\beta}_0 + \hat{\beta}_1 \text{age} + \hat{\beta}_2 \text{sex} + \varepsilon_i$, where the error is estimated from the data.

- **Pros**: An improvement over regression imputation, allows for better estimation of true variance.
- **Cons**: Still underestimates variance, invalid method.

Hot-Deck Imputation

- Divide sample units into classes (i.e. based on age and sex). For any “NA” value in a given class, replace ”NA” with a random observed value from that class.
  - Among 18-34 year old women, there are 20 observed values and 3 missing values. For each missing value, replace it with an observed value at random (with replacement).

- **Pros**: You’re using existing data.
- **Cons**: If you impute columns separately, multivariate relationships will be distorted. Invalid method.

Intuition of Imputation

Index	Age	Sex	Income
1	42	M	60000
2	39	M	75000
3	56	NA	100000
4	28	F	50000
...	...	...	...
10000	18	F	30000

- 80% of my observed values are female.
- 20% of my observations are male.
- What should I do?

Intuition of Imputation

Index	Age	Sex	Income
1	42	M	60000
2	39	M	75000
3	56	NA	100000
4	28	F	50000
...	...	...	...
10000	18	F	30000

- Usually the first guess is to “fill in female.”
  - But then we’re making up data.
- The next guess might be “wait… do we fill in male?”
  - Then we’re making up data **and** are picking a less likely value.
- The follow up is “use income and age to predict sex, but fill in with the likeliest value.”
  - This is still making up data!

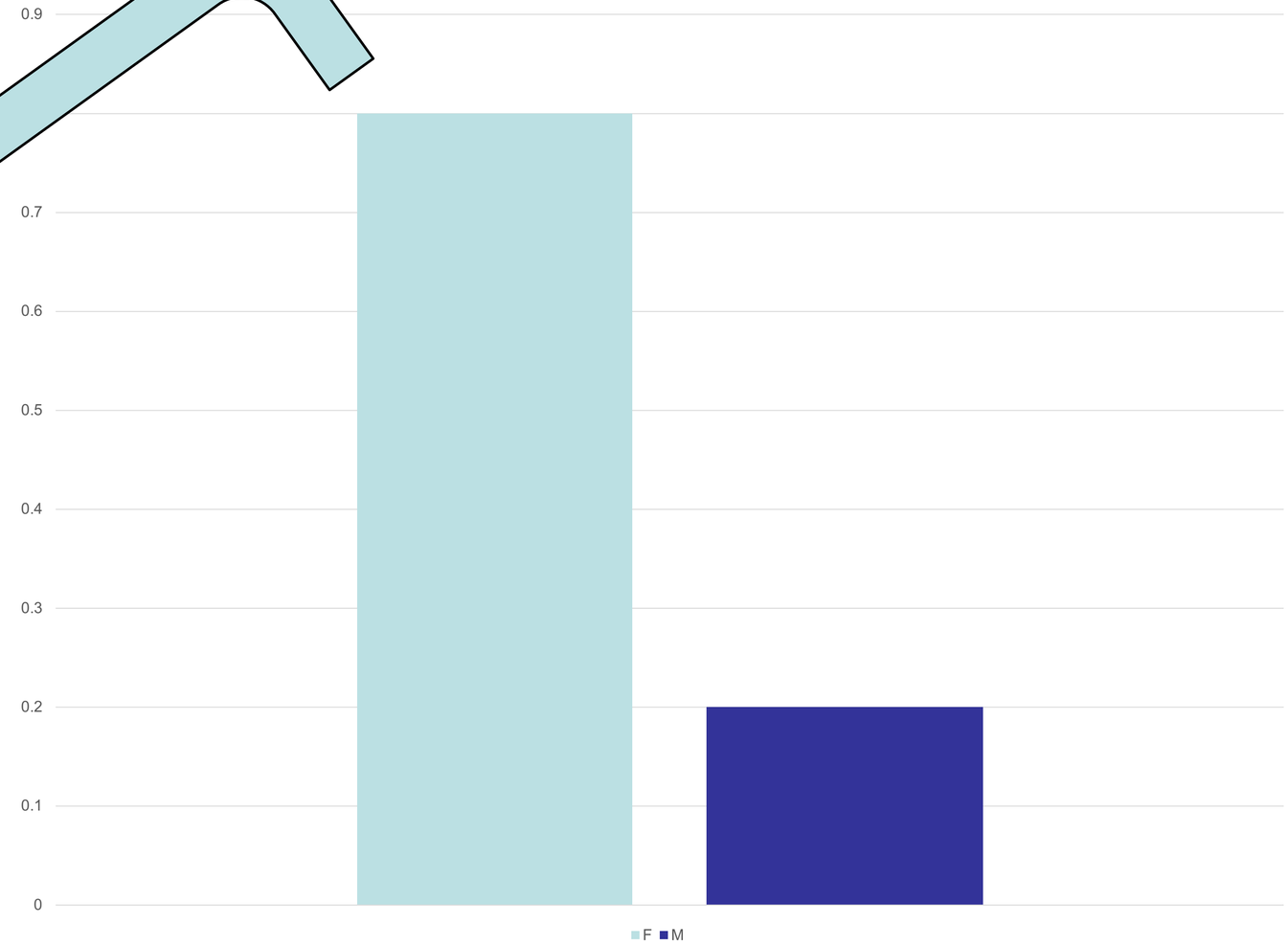
Intuition of Imputation

Index	Age	Sex	Income
1	42	M	60000
2	39	M	75000
3	56	NA	100000
4	28	F	50000
...	...	...	...
10000	18	F	30000

- We can't really do this, but what we should do is replace "NA" with a distribution.
- This distribution should be the set of all values that we think "Sex" can take on there, plus the frequency of those values.
- This distribution can totally summarize our uncertainty.

Intuition of Imputation

Index	Age	Sex	Income
1	42	M	60000
2	39	M	75[illegible]0
3	56	NA	100000
4	28	F	50000
…	…	…	…
10000	18	F	30000

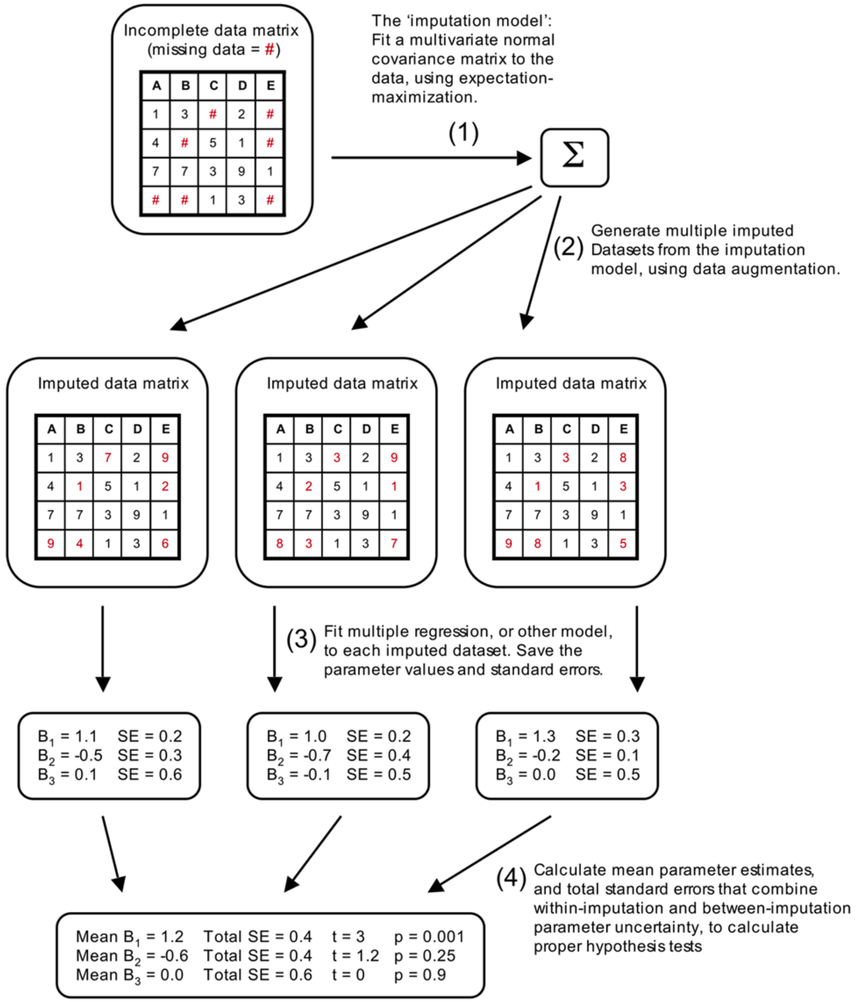


Intuition of Imputation

Index	Age	Sex	Income
1	42	M	60000
2	39	M	75000
3	56	NA	100000
4	28	F	50000
…	…	…	…
10000	18	F	30000

- Realistically, though, we're going to:
  - Make lots of copies of our dataset. (Let's say 10.)
  - Randomly simulate 10 values from that distribution replacing NA.
  - Fill each copy in with one of those 10 values.
  - Build our "final analysis" or "final model" on each of the 10 datasets.
  - Combine all of those results together at the end.

Intuition of Imputation



- Realistically, though, we're going to:
  - Make lots of copies of our dataset. (Let's say 10.)
  - Randomly simulate 10 values from that distribution replacing NA.
  - Fill each copy in with one of those 10 values.
  - Build our "final analysis" or "final model" on each of the 10 datasets.
  - Combine all of those results together at the end.

Proper Multiple Stochastic Regression Imputation

- For each dataset: Generate a set of coefficients for your regression model.
  - For each missing value in each data set: replace “NA” with a value predicted from a regression line.
  - Do your ”final analysis” or generate your “final predictions.”
- Aggregate your analysis/predictions across all datasets so you have one complete analysis or set of predictions.
  - These have been created by properly estimating the variance in your data.
- **Pros**: Very good version, **valid method**.
- **Cons**: Usually takes more effort to implement. (Python: `fancyimpute`, R: `mice`) **Assumes MAR data!**

Two comments about imputation!

- Assuming that we used a valid method of imputation, **we did not make up data**!
  - We did our analysis by properly estimating our variance and properly expressing our uncertainty in the results.
  - We never replaced an NA with a specific value - we put in multiple values that reflected our understanding of the uncertainty in that value!

- If your goal is just to have a “complete” data set for further analysis, **be careful**.
  - After you construct this dataset, nobody will know the difference between observed and imputed data.

If your goal is inference…

- Suppose that you are studying a model $Y = \beta_0 + \beta_1 X_1$ and your goal is to understand the effect of X_1 on Y.

- If you conduct proper imputation and have fit many different models, you can combine your values of $\hat{\beta}_1$ (and any parameters!) using **Rubin's rules**.
  - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2727536/
  - https://www.ibm.com/support/knowledgecenter/de/SSLVMB_22.0.0/com.ibm.spss.statistics.algorithms/alg_mi-pooling_rubin_combine.htm
  - R Code: https://stefvanbuuren.name/mice/reference/pool.html

Pattern Submodel Approach

- This is a recently published method. (September 2018!)

- Big picture: we will break our dataset into subsets based on missingness pattern. We will fit one model on each subset, resulting in many different models.

Pattern Submodel Missingness Patterns

Y	X_1	X_2
obs	obs	obs
obs	obs	obs
obs	obs	NA
obs	obs	NA
obs	NA	obs
obs	NA	obs
obs	NA	NA

Pattern Submodel Missingness Patterns

	Y	X_1	X_2
pattern 1	obs	obs	obs
	obs	obs	obs
	obs	obs	NA
	obs	obs	NA
	obs	NA	obs
	obs	NA	obs
	obs	NA	NA

Pattern Submodel Missingness Patterns

	Y	X_1	X_2
pattern 1	obs	obs	obs
	obs	obs	obs
pattern 2	obs	obs	NA
	obs	obs	NA
	obs	NA	obs
	obs	NA	obs
	obs	NA	NA

Pattern Submodel Missingness Patterns

	Y	X_1	X_2
pattern 1	obs	obs	obs
	obs	obs	obs
pattern 2	obs	obs	NA
	obs	obs	NA
pattern 3	obs	NA	obs
	obs	NA	obs
pattern 4	obs	NA	NA

Pattern Submodel Missingness Patterns

	Y	X_1	X_2	pattern
pattern 1	obs	obs	obs	1
	obs	obs	obs	1
pattern 2	obs	obs	NA	2
	obs	obs	NA	2
pattern 3	obs	NA	obs	3
	obs	NA	obs	3
pattern 4	obs	NA	NA	4

Pattern Submodel Missingness Patterns

	Y	X_1	X_2	pattern
pattern 1	obs	obs	obs	1
	obs	obs	obs	1
pattern 2	obs	obs	NA	2
	obs	obs	NA	2
pattern 3	obs	NA	obs	3
	obs	NA	obs	3
pattern 4	obs	NA	NA	4

- Big picture: we will break our dataset into subsets based on missingness pattern. We will fit one model on each subset.

Pattern Submodel Pseudocode/Python

```
for i in patterns:

    # instantiate whatever model you want
    model = Model()

    # fit that model on only one pattern at a time
    model.fit(df_train[df_train['pattern'] == i][['list_of_X_cols']],
              df_train[df_train['pattern'] == i][['y_col']])

    # generate predictions for test values on one pattern at a time
    predictions.append(model.predict(df_test[df_test['pattern'] == i][['list_of_X_cols']]))

    # store the score for each submodel for later combination *maybe*
    scores.append(model.score(df))
```

Pattern Submodel Approach

- Pros:
  - Outperforms imputation methods for NMAR data and performs on par with imputation methods for MAR and MCAR data when generating predictions;
  - allows you to generate predictions for test observations with missing values;
  - computationally efficient;
  - intuitive method;
  - does not require missingness assumptions (!).

- Cons: Is not a well-understood method for inference; new method.

Pattern Submodel Practical Considerations

- If you have p predictors, it is possible for you to have up to 2^p different missingness patterns, requiring 2^p models to be built.
  - In practice, your number of missingness patterns is likely to be far lower.

- If you have a missingness pattern with a small number of observations, your submodel on that pattern may suffer from high error due to variance (overfitting).
  - Consider collapsing small categories into larger categories, if possible.

Pattern Submodel Practical Considerations

- Suppose you have variables Y, X_1, X_2, and X_3.

- If you have 2 observations with X_1 and X_2 observed but 40 observations with only X_1 observed, then consider grouping the 2 observations in with the 40.
  - Basically, we are dropping X_2 for those 2 observations so we have one missingness pattern for all 42 observations.
    - This does make an implicit MCAR assumption, which is usually not true.
    - We're also discarding information (in this case, we're discarding 2 values of X_2).
    - However, if this is done sparingly, it *shouldn't* cause your model to perform substantially worse.

Agenda

1. How big of a problem is missing data?
2. The three types of missingness.
3. Three strategies for tackling missing data.
4. **Practical recommendations.**

What type of missingness – MCAR, MAR, or NMAR?

1. **Little's Test for MCAR**
   - Hypothesis test available in most software packages, similar to a t-test.
   - $H_0: MCAR$ vs. $H_A: MAR$
   - There is **no empirical test to evaluate NMAR**!

2. **Partition your data into "observed" and "unobserved**," then compare them.

3. Think about the missing data process. **Can you come up with a reasonable answer** based on how missing data came about?

If we have MCAR…

- We can use any of these methods we previously discussed with their respective caveats.
  - Recommendations:
    - Deductive Imputation
    - Pattern Submodel Method
    - Proper Imputation
    - Stochastic Regression Imputation
    - Hot-Deck Imputation
    - Complete-Case Analysis

If we have MAR…

- Complete-case analysis will be biased.
- We can use any of these methods we previously discussed with their respective caveats.
  - Recommendations:
    - Deductive Imputation
    - Pattern Submodel Method
    - Proper Imputation
    - Stochastic Regression Imputation
  - **This assumes that we include the MAR variables in our modeling!**

If we have NMAR…

- We can use any of these methods we previously discussed with their respective caveats:
  - Deductive Imputation
  - Pattern Submodel Method
- We cannot (should not) use these methods:
  - Complete-case analysis
    - Proper Imputation
    - Stochastic Regression Imputation
    - Hot-Deck Imputation

What is my workflow?

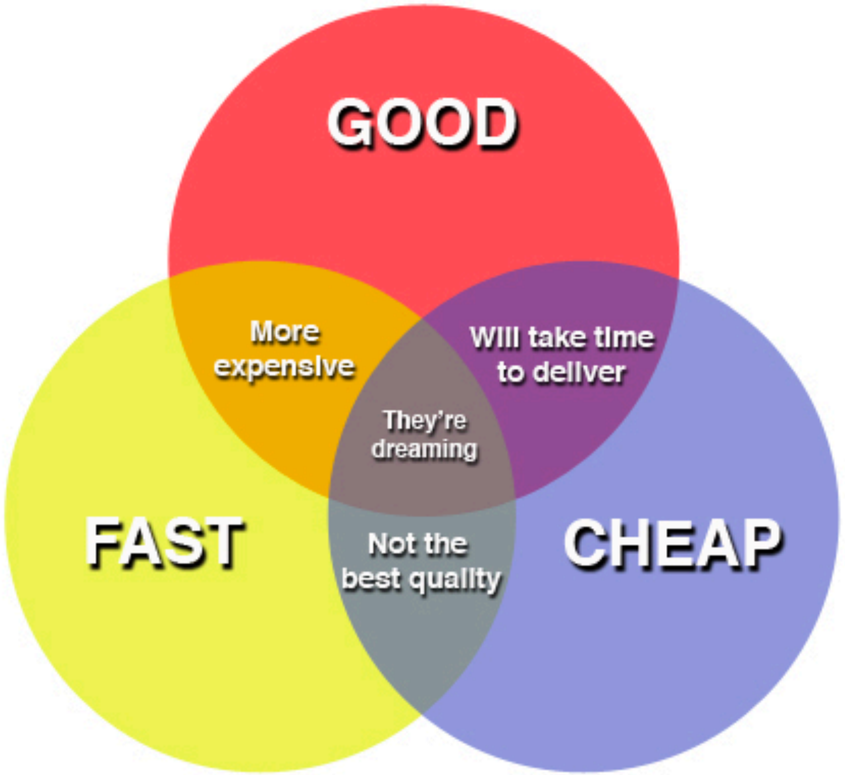
1. I evaluate how much missing data I have during EDA. **Is it worth my time** to try and address it?

2. Is it reasonable to attempt deductive imputation?

3. If my goal is to generate predictions, then **use the pattern submodel approach.**

4. If my goal is to conduct inference, then:
   - **for each variable, estimate what type of missingness I have.**
   - **use the best "accounting for missingness" method given my constraints.**
   - **if doing proper imputation, combine results using Rubin's rules.**

Good, Fast, Cheap



- **Fast and Cheap Analysis:** Drop all missing values or do rudimentary imputation.
- **Good and Cheap Analysis:** Proper imputation.
- **Good and Fast Analysis:** Gather data in a complete manner.

Thank you!

- Matt Brems
  - Global Lead Data Science Instructor, General Assembly
  - Managing Partner, ROC AUC, LLC
  - Project & Client Management Chair, Statistics Without Borders

  - LinkedIn: Matthew Brems
  - Github: matthewbrems
  - Twitter: @matthewbrems
  - Medium: @matthew.w.brems