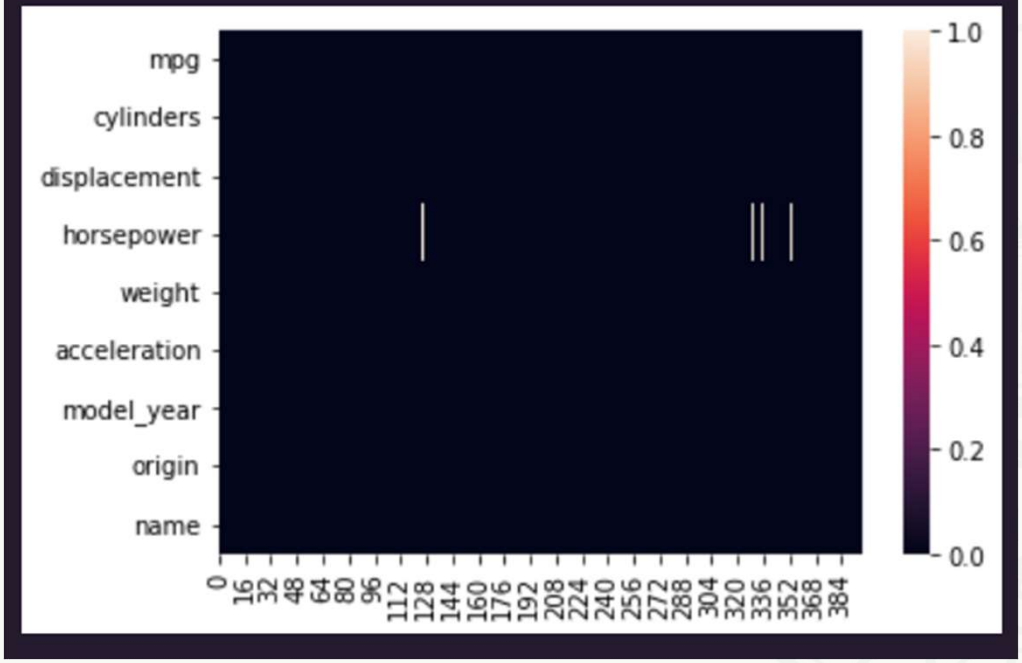
Imputation

Prof. Murillo
Computational Mathematics, Science and Engineering
Michigan State University

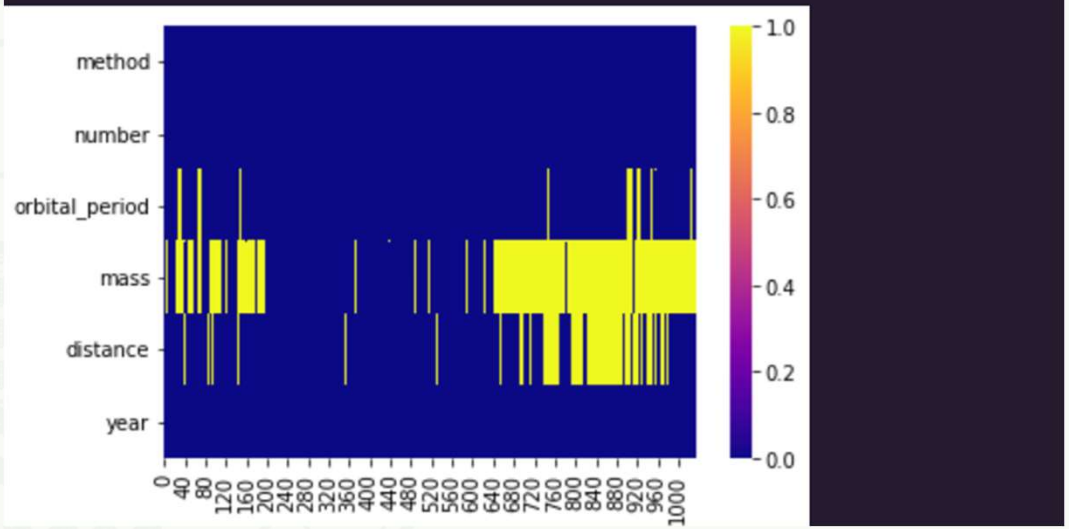
Visualize Missing Data: Part of the EDA Process

```
1 sns.heatmap(df_m.isna().transpose());
```



Seaborn: mpg

```
1 sns.heatmap(df_pl.isna().transpose(),cmap="plasma");
```



Seaborn: planets

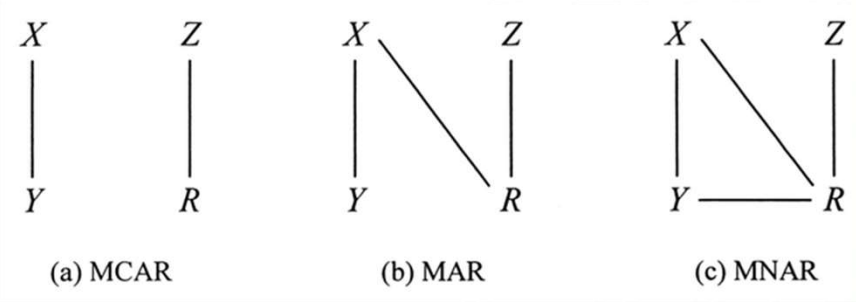
pandas.isna

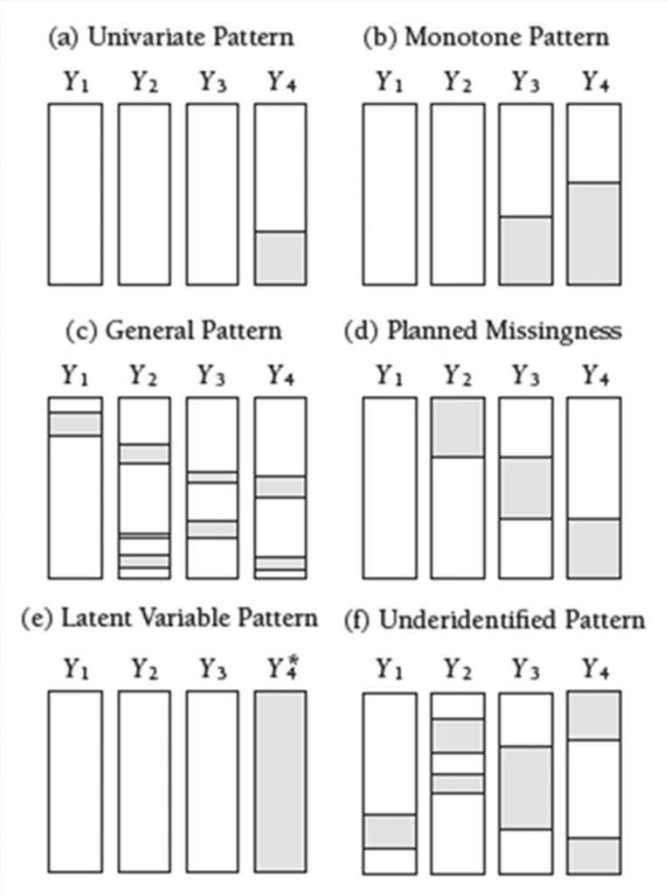
pandas.isna(*obj*) [source]

Detect missing values for an array-like object.

This function takes a scalar or array-like object and indicates whether values are missing (`NaN` in numeric arrays, `None` or `NaN` in object arrays, `NaT` in datetimelike).

Missingness Mechanisms and Patterns





MNAR Example

1. Income Surveys

- **Scenario**: In a national income survey, high-income individuals are less likely to report their income.
- **MNAR Mechanism**: The probability of missing data increases with income level.
- **Bias from Listwise Deletion**: Removing all cases with missing income data would underestimate the average income and misrepresent the income distribution.

MNAR Example

2. Mental Health Studies

- **Scenario**: In a study on depression, severely depressed individuals are less likely to complete follow-up assessments.
- **MNAR Mechanism**: The likelihood of missing data is directly related to the severity of depression.
- **Bias from Listwise Deletion**: Excluding incomplete cases would underestimate the prevalence and severity of depression in the population.

MNAR Example

3. Employee Performance Reviews

- **Scenario**: In a company's performance review database, low-performing employees are more likely to have missing evaluation scores.
- **MNAR Mechanism**: The probability of missing performance scores is higher for lower-performing employees.
- **Bias from Listwise Deletion**: Removing employees with missing scores would artificially inflate the average performance score.

MNAR Example

4. Drug Trial Dropouts

- **Scenario**: In a clinical trial for a new medication, patients experiencing severe side effects are more likely to drop out before the study's completion.
- **MNAR Mechanism**: The likelihood of missing data increases with the severity of side effects.
- **Bias from Listwise Deletion**: Excluding dropouts would underestimate the drug's side effects and overestimate its efficacy.

MNAR Example

5. Customer Satisfaction Surveys

- **Scenario**: In a product satisfaction survey, customers who are very dissatisfied are less likely to complete the survey.
- **MNAR Mechanism**: The probability of missing responses is higher for dissatisfied customers.
- **Bias from Listwise Deletion**: Removing incomplete surveys would overestimate overall customer satisfaction.

MNAR Example

6. Academic Performance Tracking

- **Scenario**: In a longitudinal study of student performance, students with poor grades are more likely to drop out of the study.
- **MNAR Mechanism**: The likelihood of missing data in later years is related to poor academic performance.
- **Bias from Listwise Deletion**: Excluding students with incomplete data would overestimate average academic performance over time.

MNAR Example

7. Sensor Data in IoT Devices

- **Scenario:** In a network of IoT devices, sensors are more likely to fail (and thus produce missing data) under extreme conditions.
- **MNAR Mechanism:** The probability of missing sensor readings increases with extreme temperature or pressure conditions.
- **Bias from Listwise Deletion:** Removing timepoints with missing sensor data would underestimate the frequency and severity of extreme conditions.

Imputation

imputation: assignment of a missing value

When we are not MCAR, we need to replace the missing values with something *reasonable*.

With **imputation**, we do not drop anything, we replace the `NaN`s with something.

Are the missing values random?

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
32	25.0	4	98.0	NaN	2046	19.0	71	usa	ford pinto
126	21.0	6	200.0	NaN	2875	17.0	74	usa	ford maverick
330	40.9	4	85.0	NaN	1835	17.3	80	europe	renault lecar deluxe
336	23.6	4	140.0	NaN	2905	14.3	80	usa	ford mustang cobra
354	34.5	4	100.0	NaN	2320	15.8	81	europe	renault 18i
374	23.0	4	151.0	NaN	3035	20.5	82	usa	amc concord dl

None of the missing values are from Japan.

If we drop these rows we are biasing the dataset toward Japan.

And, we are losing useful data on the other features.

Do you have MCAR and only a few missing values?

Use a **deletion** method.

This is the “opposite” of imputation.

Listwise Deletion

List wise deletion

Gender	Manpower	Sales
M	25	343
F	.	280
M	33	332
M	.	272
F	25	.
M	29	326
	26	259
M	32	297

Missing data is minimal and MCAR.

Removes any row with one (or more) missing values.

Easy to Code, or Pandas

pandas.DataFrame.dropna

`DataFrame.dropna(*, axis=0, how=<no_default>, thresh=<no_default>, subset=None, inplace=False, ignore_index=False)` [source]

Remove missing values.

See the User Guide for more on which values are considered missing, and how to work with missing data.

Parameters:

axis : *{0 or 'index', 1 or 'columns'}, default 0*

Determine if rows or columns which contain missing values are removed.

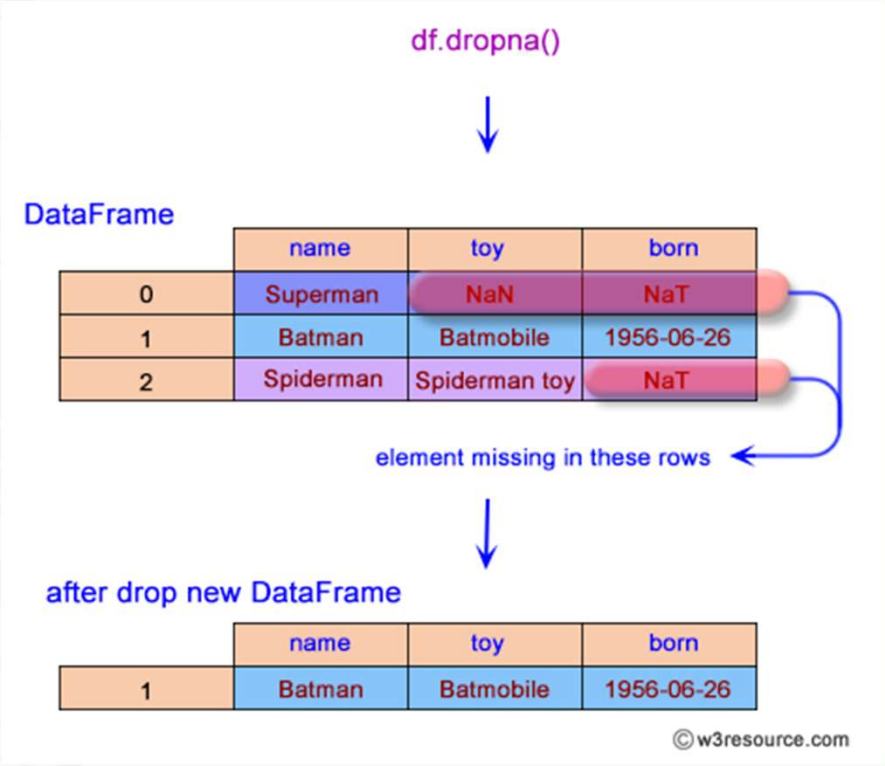
- 0, or 'index' : Drop rows which contain missing values.
- 1, or 'columns' : Drop columns which contain missing value.

Only a single axis is allowed.

how : *{'any', 'all'}, default 'any'*

Determine if row or column is removed from DataFrame, when we have at least one NA or all NA.

- 'any' : If any NA values are present, drop that row or column.
- 'all' : If all values are NA, drop that row or column.



Listwise and **Pairwise** Deletion

List wise deletion

Gender	Manpower	Sales
M	25	343
F	.	280
M	33	332
M	.	272
F	25	.
M	29	326
	26	259
M	32	297

Pair wise deletion

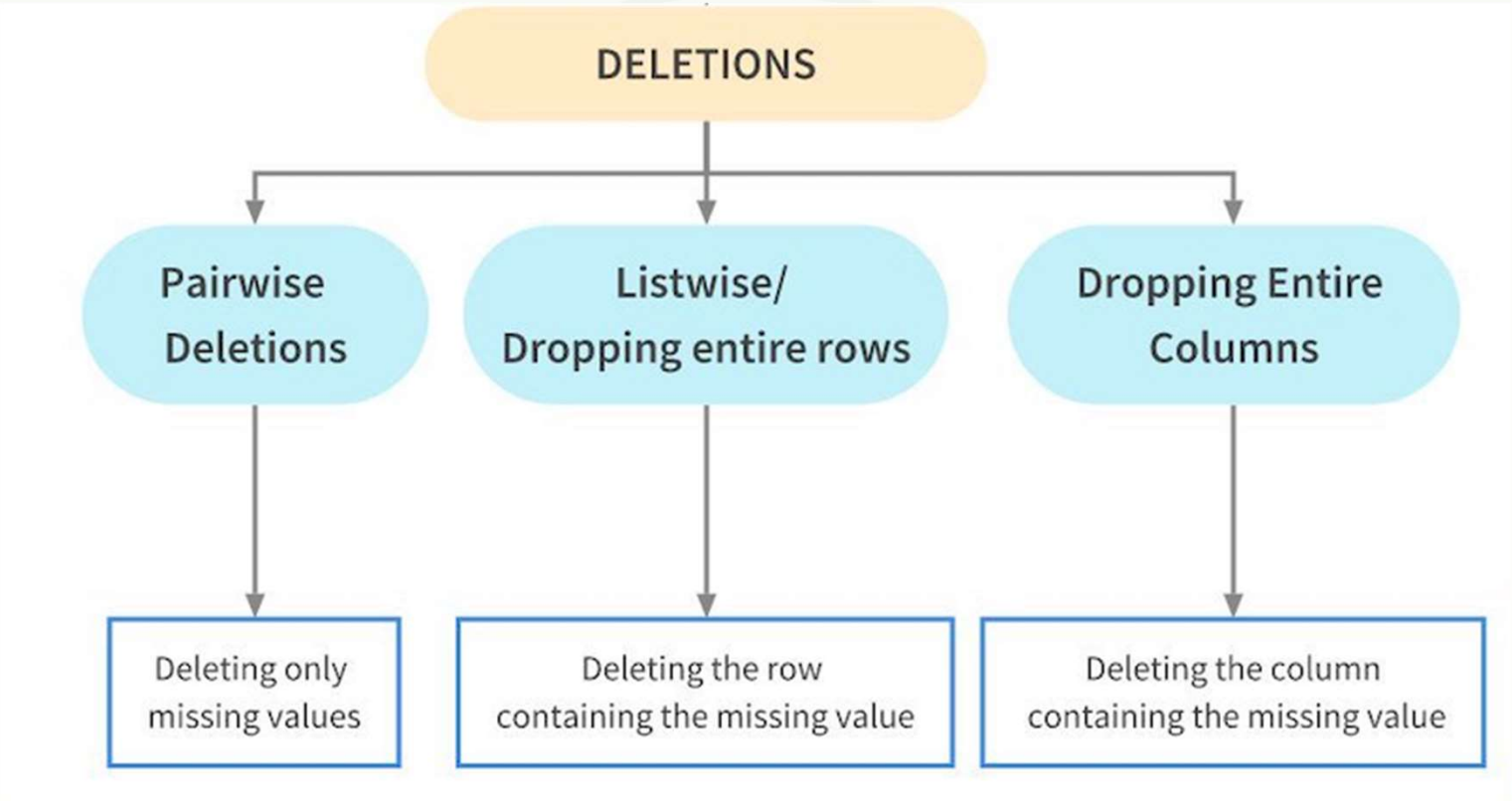
Gender	Manpower	Sales
M	25	343
F	.	280
M	33	332
M	.	272
F	25	.
M	29	326
	26	259
M	32	297

There is no reason to delete useful data in a row if you are not using the column with the NaN.

Each column now has a different number of rows, and care must be taken when computing statistics.

Don't blindly use listwise deletion before you have a workflow designed.

Really Bad? Drop the Column



Easy to Code, or Pandas

pandas.DataFrame.dropna

`DataFrame.dropna(*, axis=0, how=<no_default>, thresh=<no_default>, subset=None, inplace=False, ignore_index=False)` [source]

Remove missing values.

See the User Guide for more on which values are considered missing, and how to work with missing data.

Parameters:

axis : *{0 or 'index', 1 or 'columns'}, default 0*

Determine if rows or columns which contain missing values are removed.

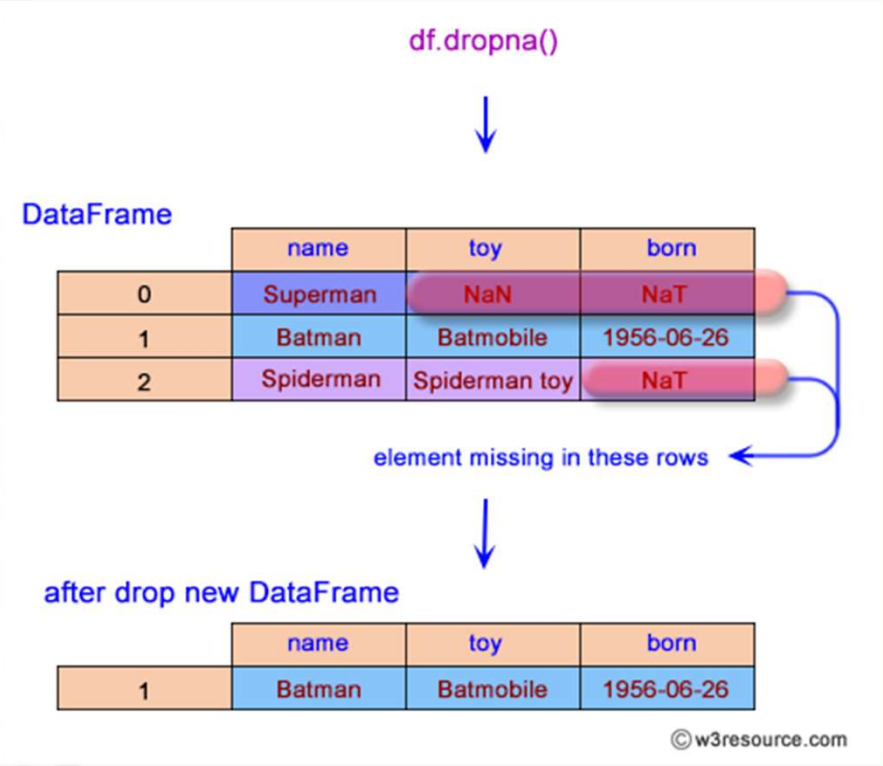
- 0, or 'index' : Drop rows which contain missing values.
- 1, or 'columns' : Drop columns which contain missing value.

Only a single axis is allowed.

how : *{'any', 'all'}, default 'any'*

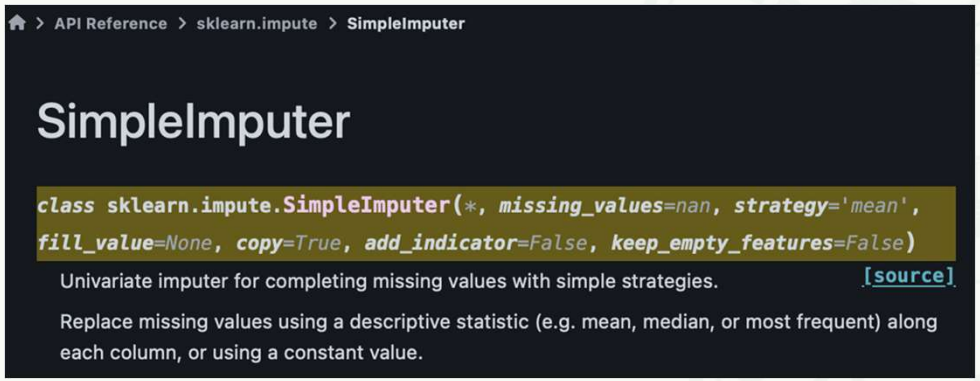
Determine if row or column is removed from DataFrame, when we have at least one NA or all NA.

- 'any' : If any NA values are present, drop that row or column.
- 'all' : If all values are NA, drop that row or column.



Handling Missing Values: Simple Approach

Suppose you don't care *why* you have missing values. Use *scikit-learn*'s `SimpleImputer`.



strategy : *str or Callable, default='mean'*

The imputation strategy.

- If "mean", then replace missing values using the mean along each column. Can only be used with numeric data.
- If "median", then replace missing values using the median along each column. Can only be used with numeric data.
- If "most_frequent", then replace missing using the most frequent value along each column. Can be used with strings or numeric data. If there is more than one such value, only the smallest is returned.
- If "constant", then replace missing values with fill_value. Can be used with strings or numeric data.
- If an instance of Callable, then replace missing values using the scalar statistic returned by running the callable over a dense 1d array containing non-missing values of each column.

Mean Substitution

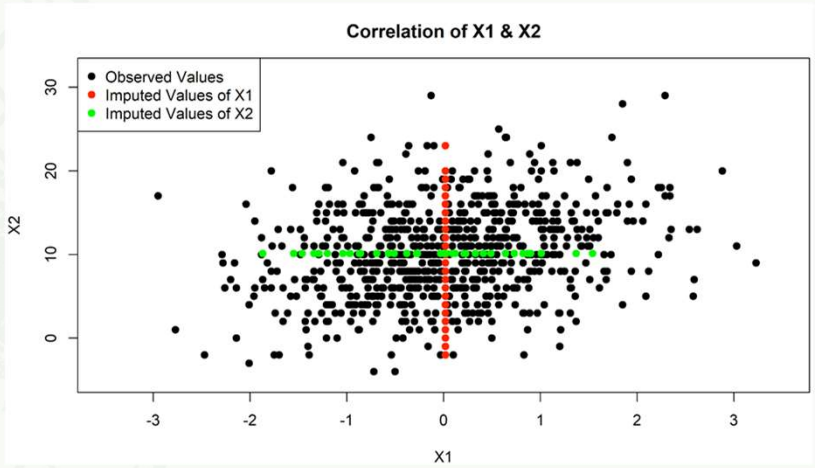
Most obvious idea: replace missing values with the average (mean/median/etc) of the values you do have.

Easy!

Price
100
90
50
40
20
100
60
120
200

Mean = 86.66

Median = 90

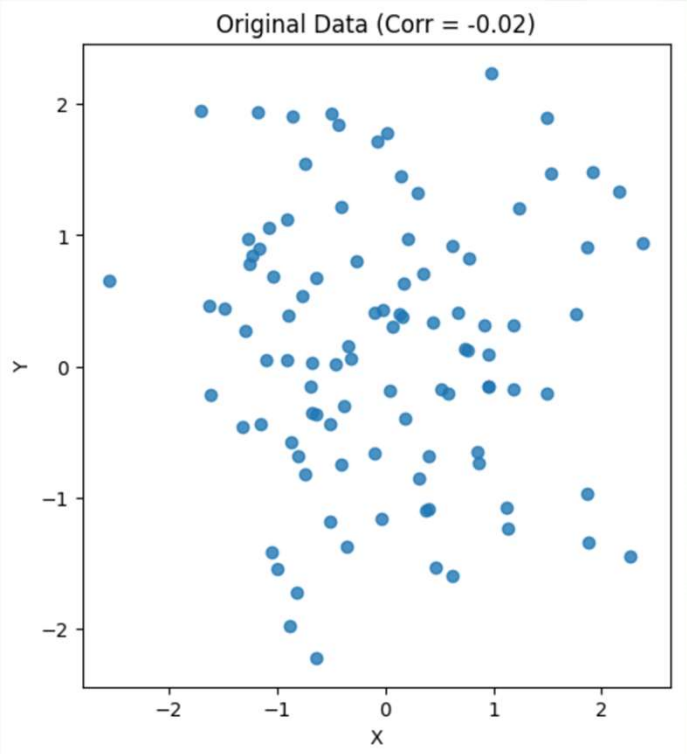


Price
100
90
50
40
20
100
86.66
60
120
86.66
200

beware...

Loss of Correlations from Mean Imputation: Examples

missing rate = 0.8

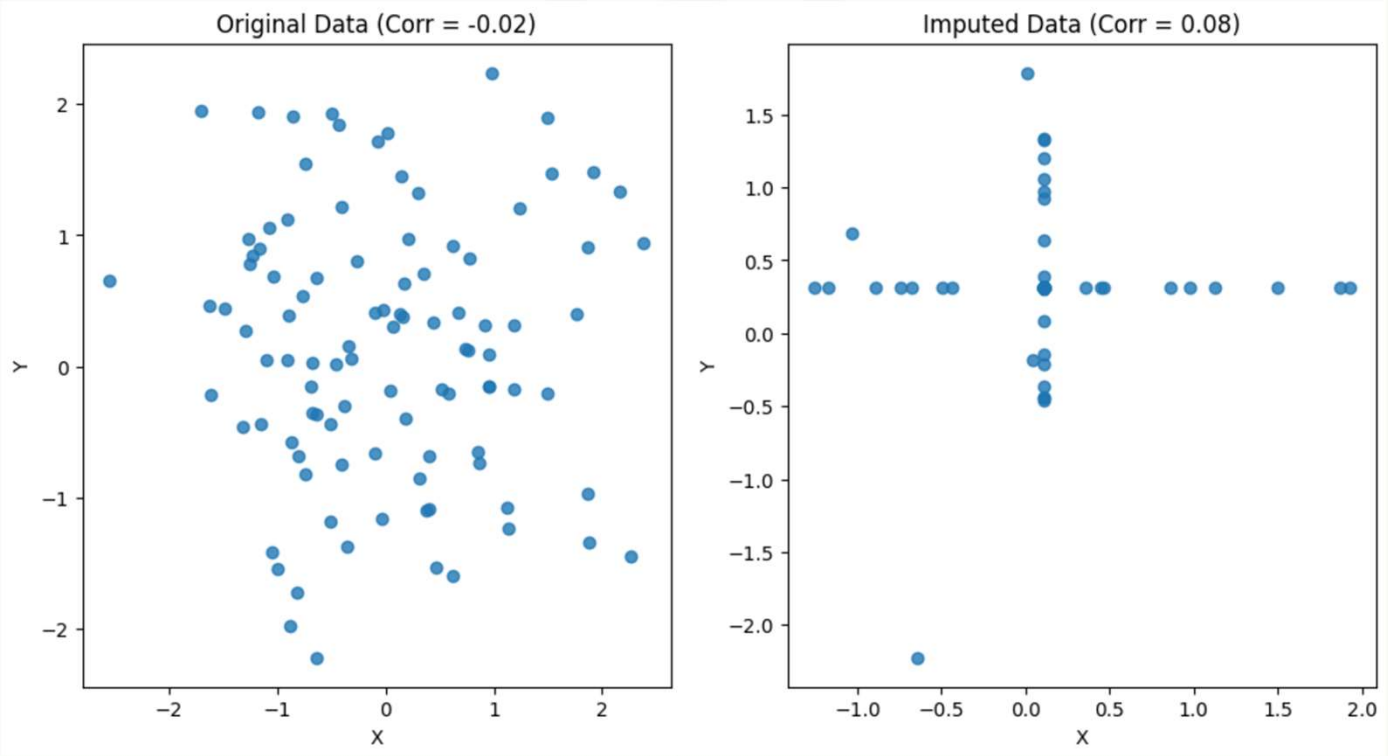


correlations are quantities that have the form:

$\langle XY \rangle$

Loss of Correlations from Mean Imputation: Examples

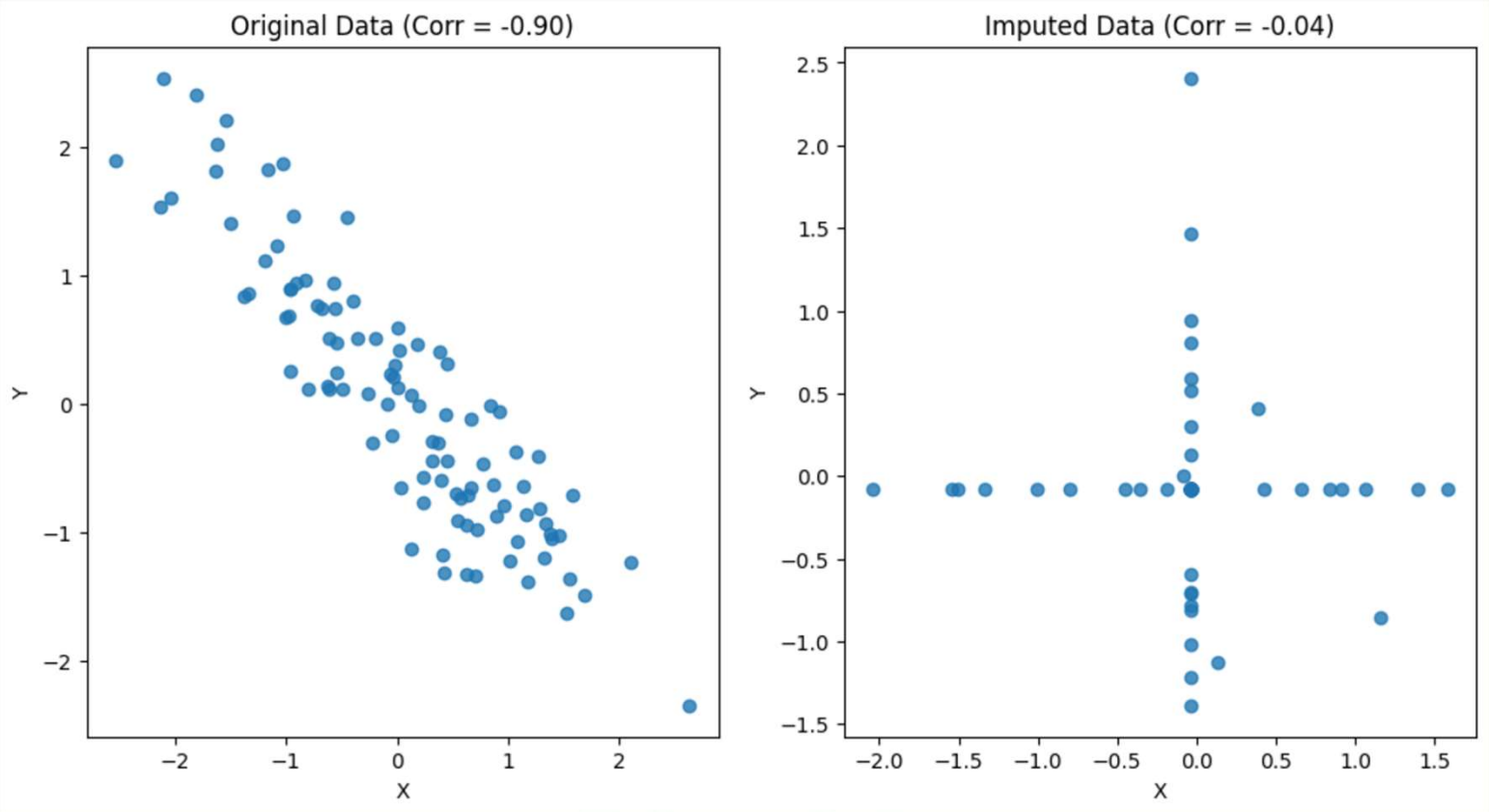
missing rate = 0.8



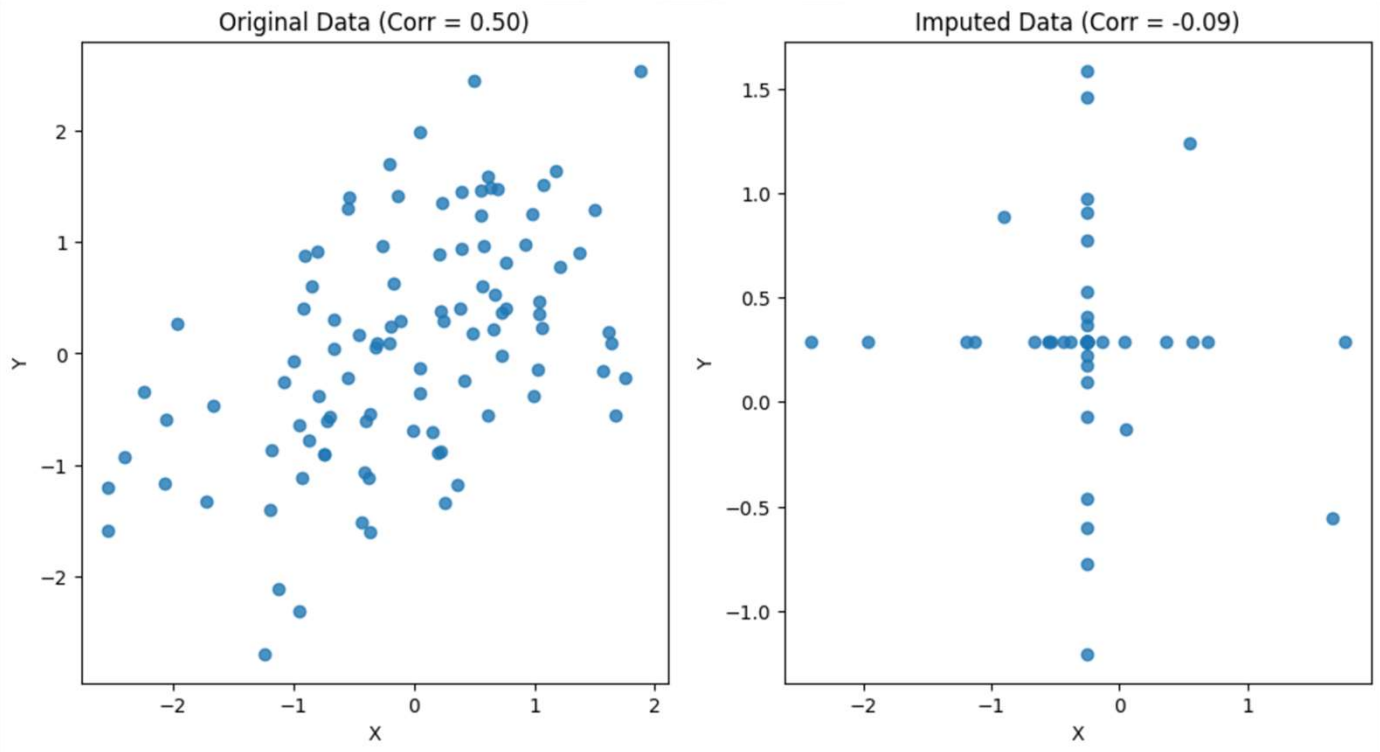
correlations are quantities that have the form:

$\langle XY \rangle$

Loss of Correlations from Mean Imputation: Examples



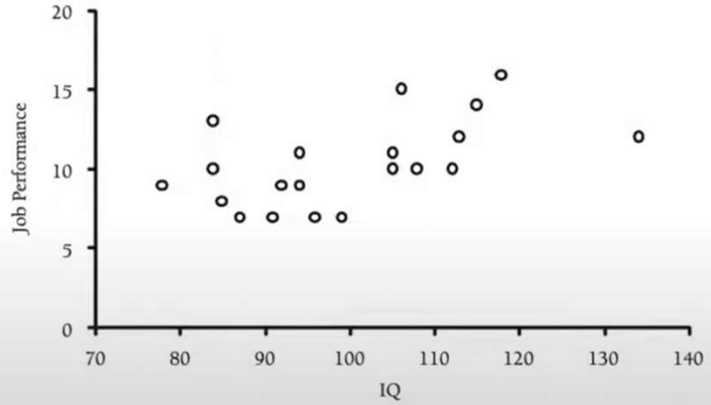
Loss of Correlations from Mean Imputation: Examples



Stochastic Regression: What is the Goal?

Here is the raw data:

Example dataset



2.1. Complete-data scatterplot of the IQ and job performance scores from Table 2.1.

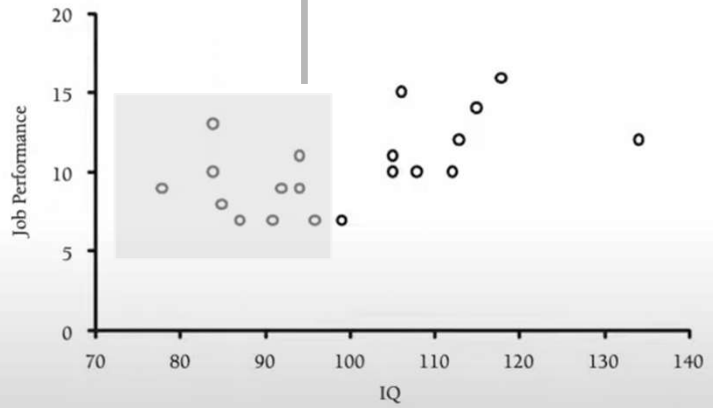
TABLE 2.1. Employee Selection Data Set

Complete data		Missing data
IQ	Job performance	Job Performance
78	9	—
84	13	—
84	10	—
85	8	—
87	7	—
91	7	—
92	9	—
94	9	—
94	11	—
96	7	—
99	7	7
105	10	10
105	11	11
106	15	15
108	10	10
112	10	10
113	12	12
115	14	14
118	16	16
134	12	12

Stochastic Regression: What is the Goal?

Here is the synthetic data:

Example dataset

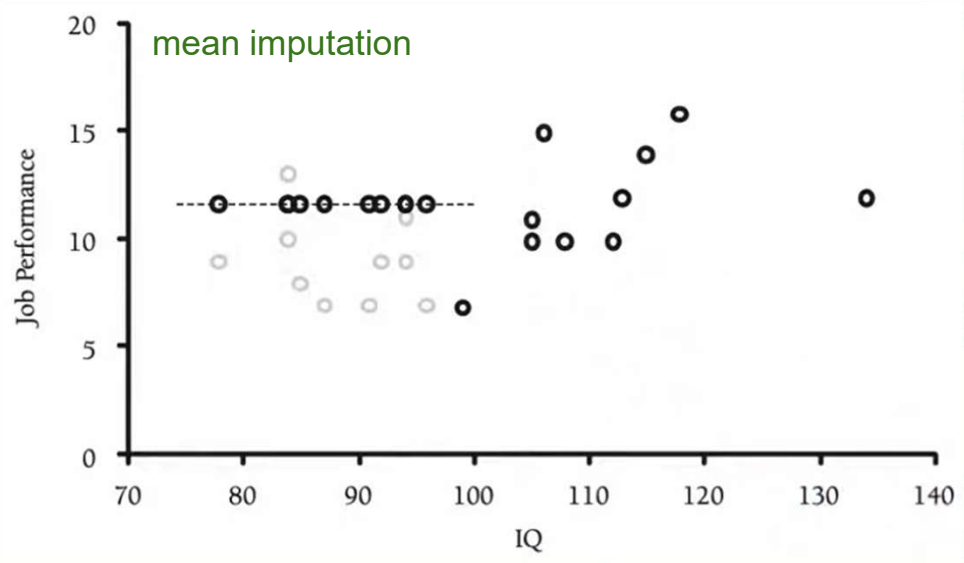


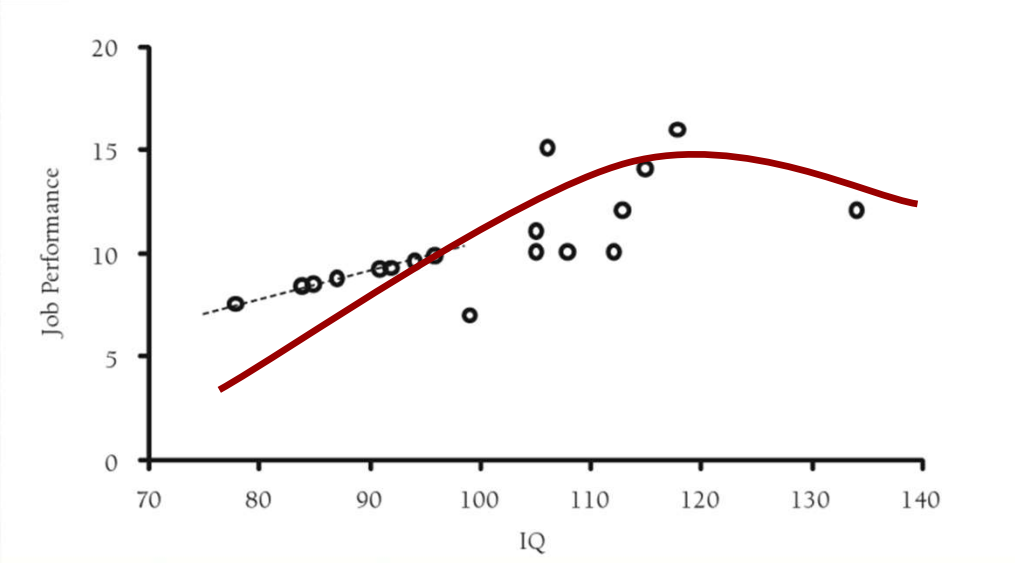
2.1. Complete-data scatterplot of the IQ and job performance scores from Table 2.1.

TABLE 2.1. Employee Selection Data Set

Complete data		Missing data
IQ	Job performance	Job Performance
78	9	—
84	13	—
84	10	—
85	8	—
87	7	—
91	7	—
92	9	—
94	9	—
94	11	—
96	7	—
99	7	7
105	10	10
105	11	11
106	15	15
108	10	10
112	10	10
113	12	12
115	14	14
118	16	16
134	12	12

Fit Data and Extrapolate



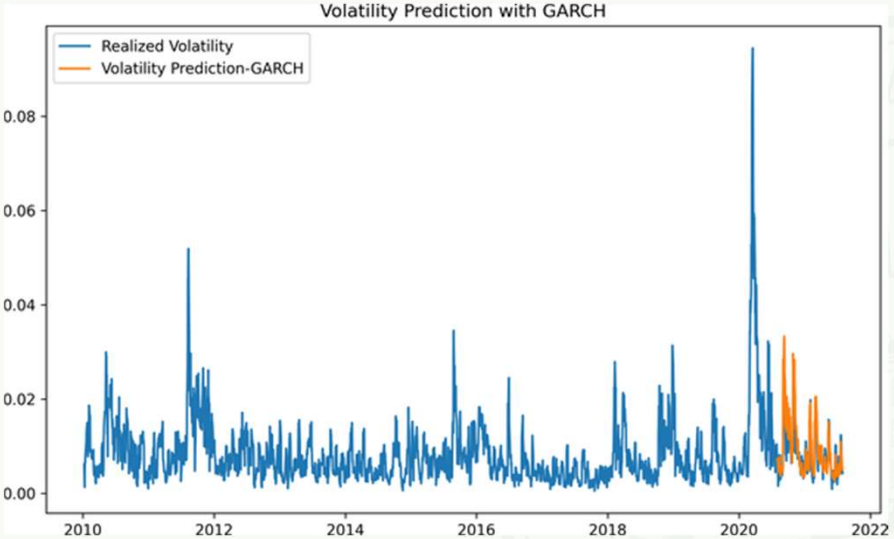


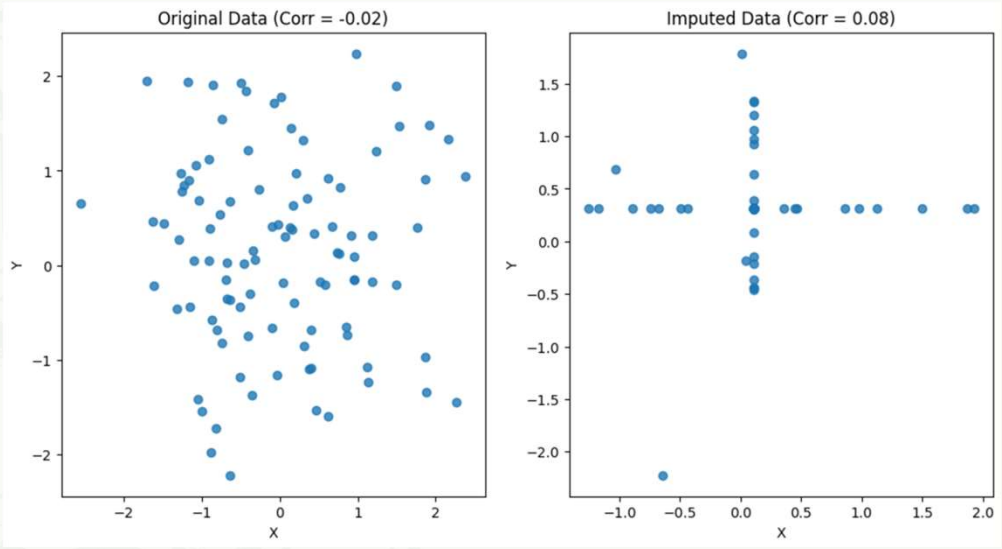
Point #1: mean imputation should “never” be done

Point #2: fitting is better, and perhaps good enough?

Point #3: don’t need to use a line (first-order polynomial)

Variance/Volatility



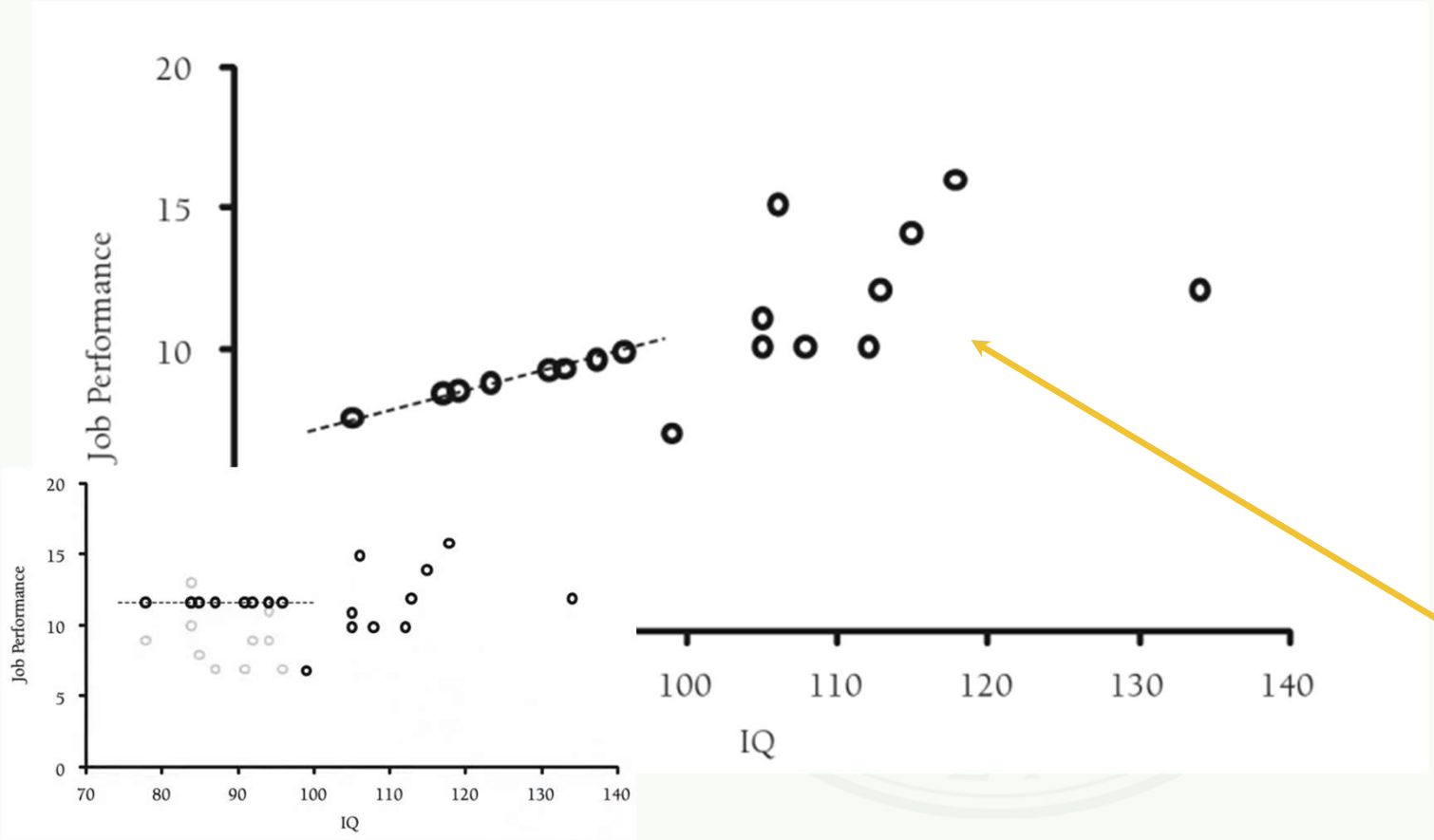


Very often we want to extrapolate the "volatility" in our data.

Sometimes, we only have "volatility".

Point #4: when we impute, we want to preserve the mean, trend and variance

Regression (Simple)



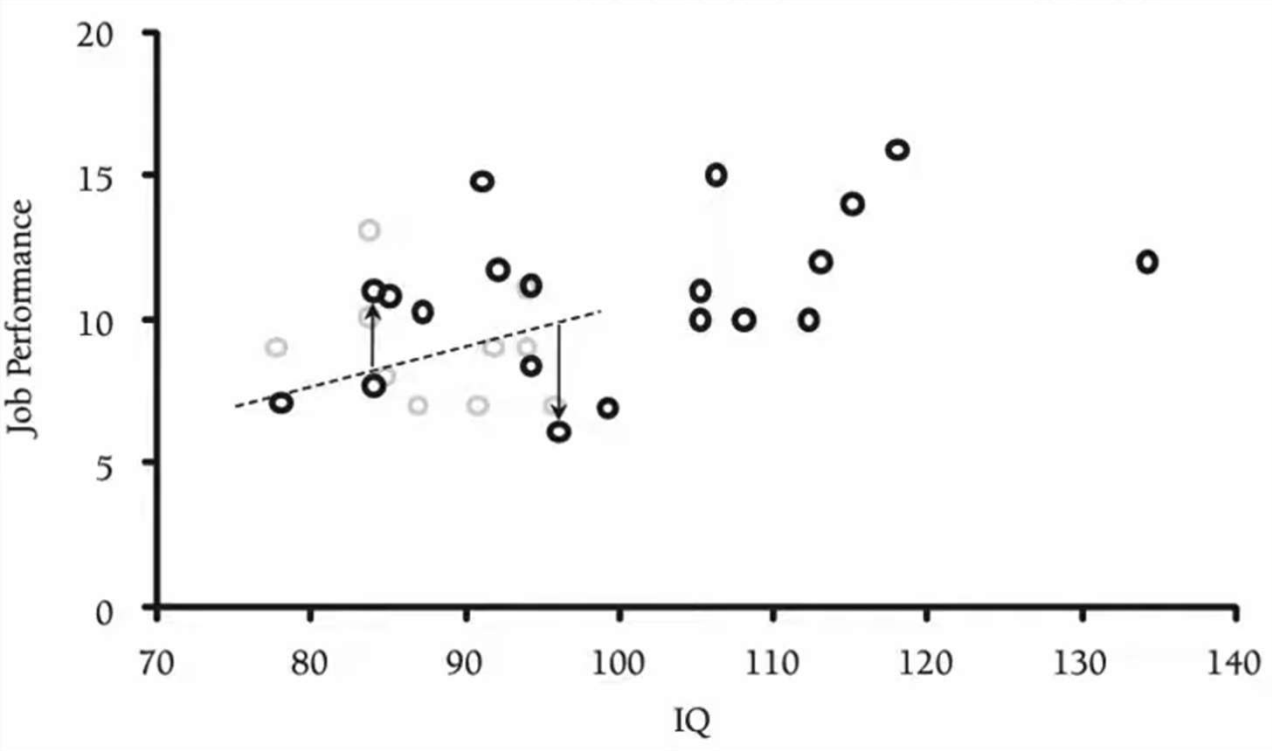
Using regression captures the trend that is lost when we use mean substitution.

However, an important message is lost!

Yes, there is a slight upward trend, but mostly there is no correlation.

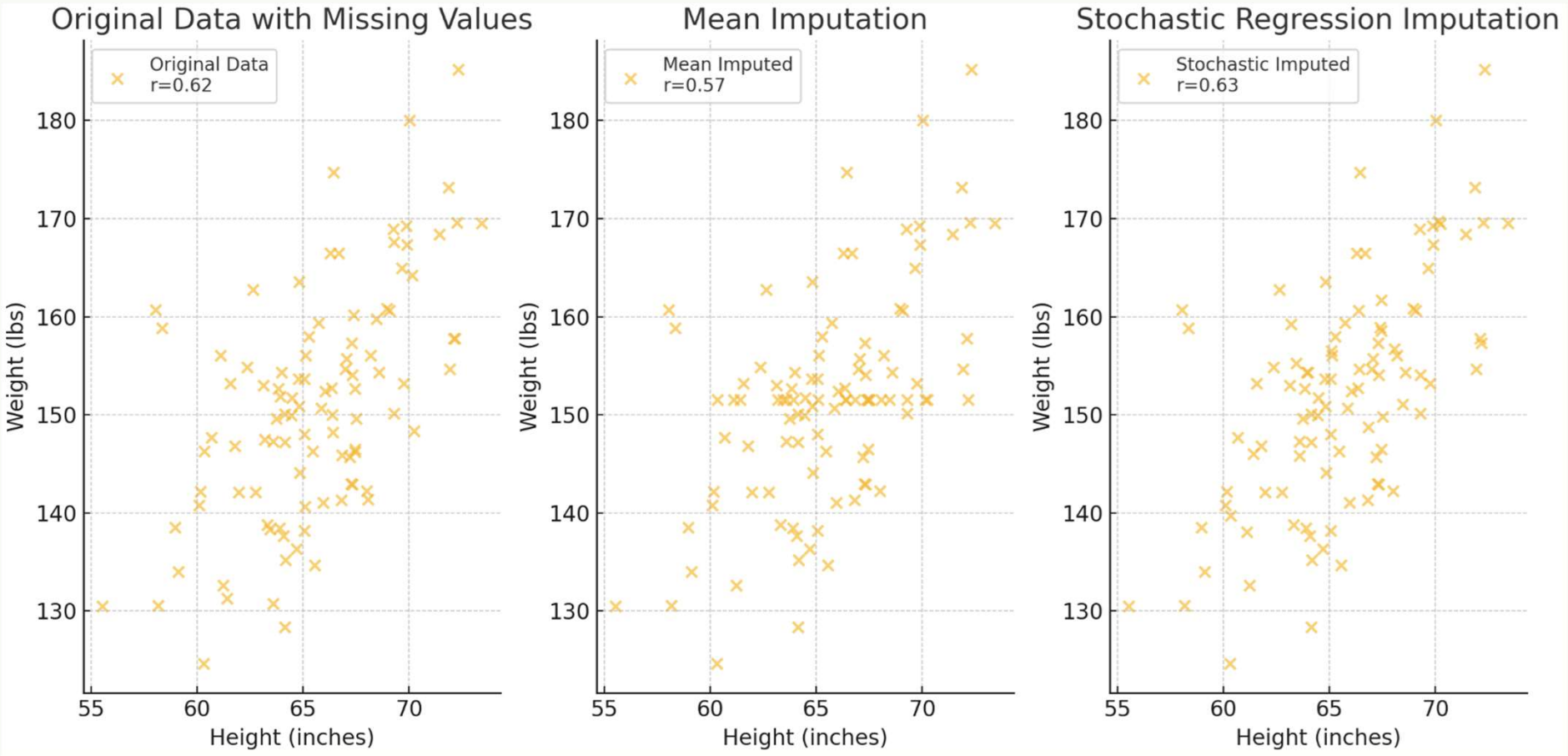
Stochastic Regression

In addition to the regression line, we can estimate the variance.
from the variance, we can add noise.



This method repairs most of the problems with previously-discussed methods.

Example: Height-Weight Dataset

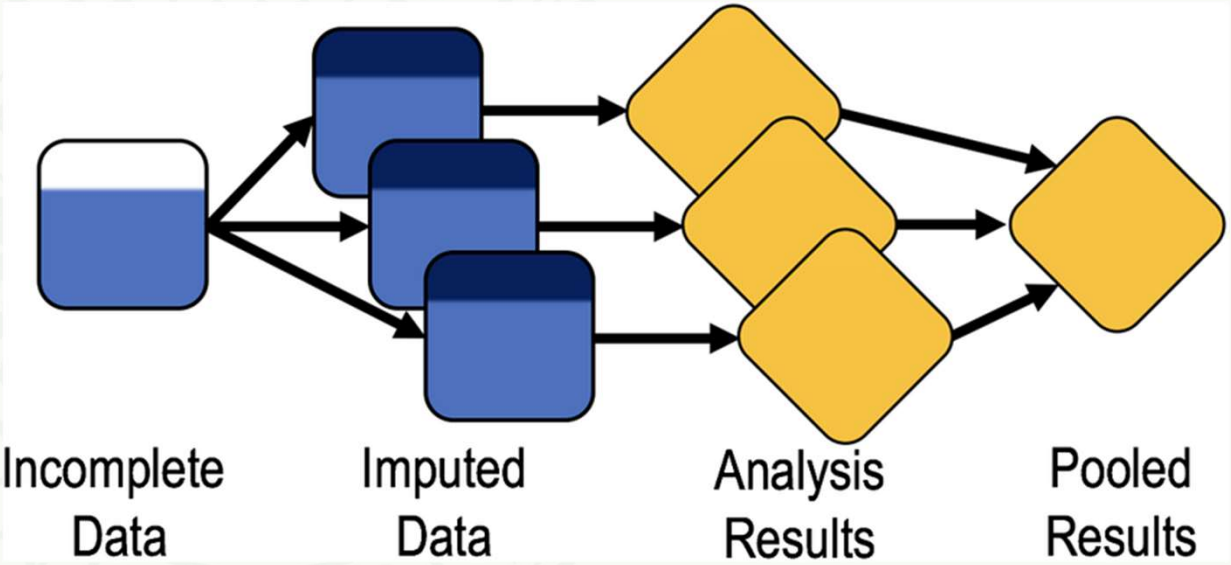


Multiple Imputation

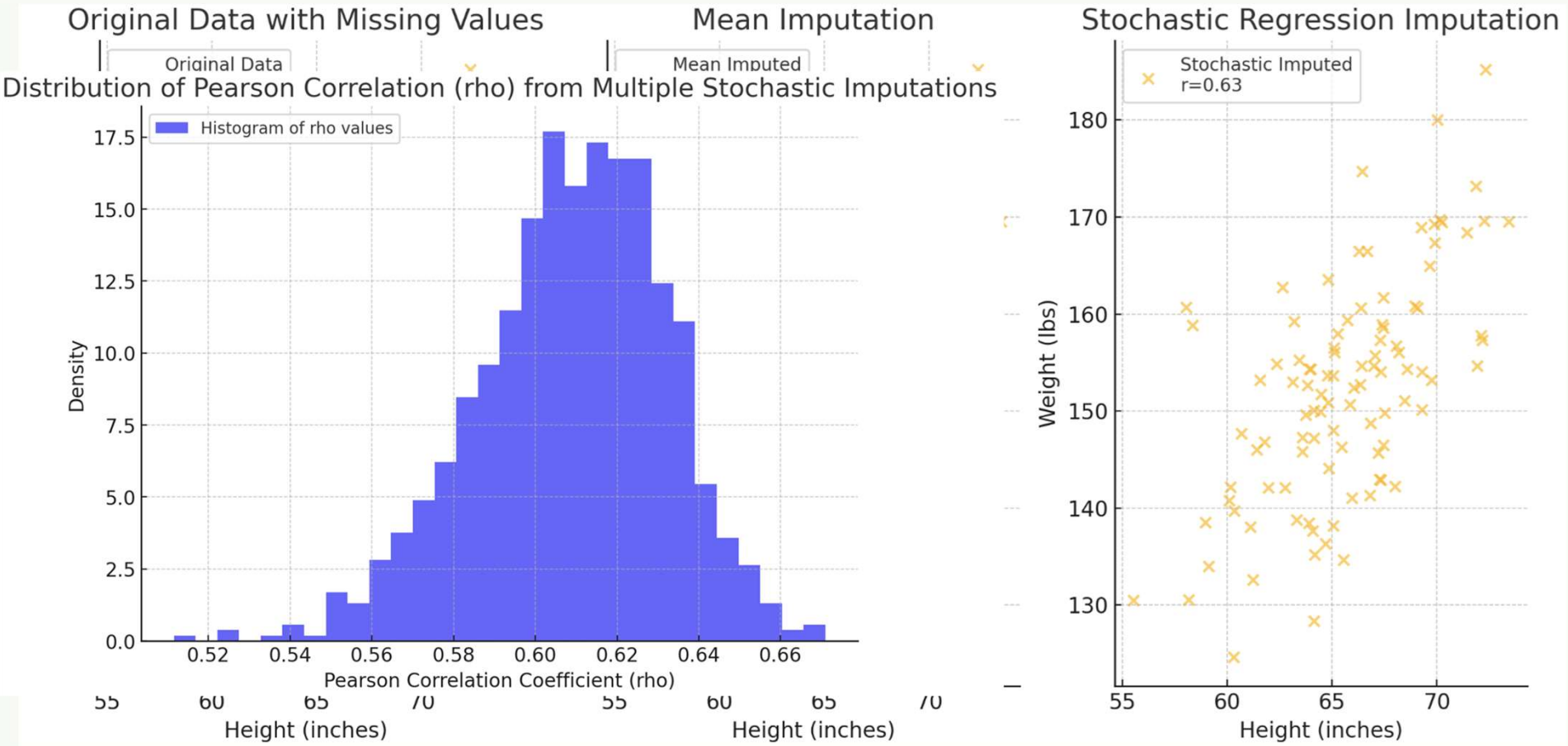
Note: the regression can be generated many times!

This is another advantage of using a stochastic method.

This means one can obtain many predictions, giving a confidence interval.



Example: Height-Weight Dataset

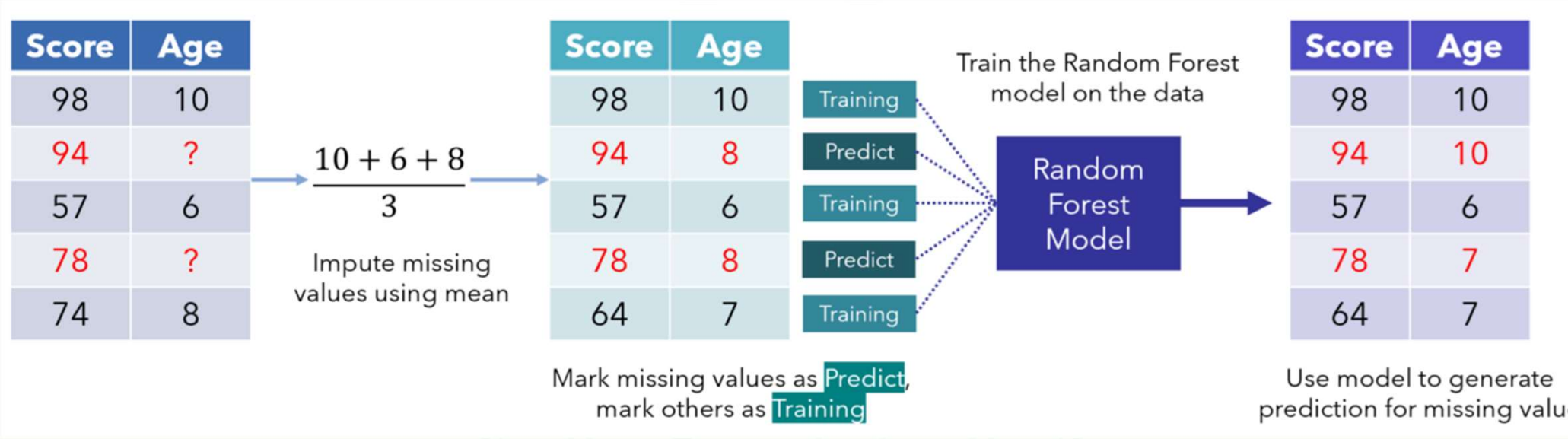


Modern Approaches: KNN

Let's examine KNN (K nearest neighbors) methods (there are *many* variants) .

28	9.0	8	304.0	193.0	4732	18.5	70	usa	hi 1200d
29	27.0	4	97.0	88.0	2130	14.5	71	japan	datsun pl510
30	28.0	4	140.0	90.0	2264	15.5	71	usa	chevrolet vega 2300
31	25.0	4	113.0	95.0	2228	14.0	71	japan	toyota corona
32	25.0	4	98.0	NaN	2046	19.0	71	usa	ford pinto
33	19.0	6	232.0	100.0	2634	13.0	71	usa	amc gremlin
34	16.0	6	225.0	105.0	3439	15.5	71	usa	plymouth satellite custom

Modern Approaches: Alternate ML Approach



Improvements:

- don't use mean imputation
- iterate
- add stochastic/multiple imputation

Repairing Missing Values: Many Techniques

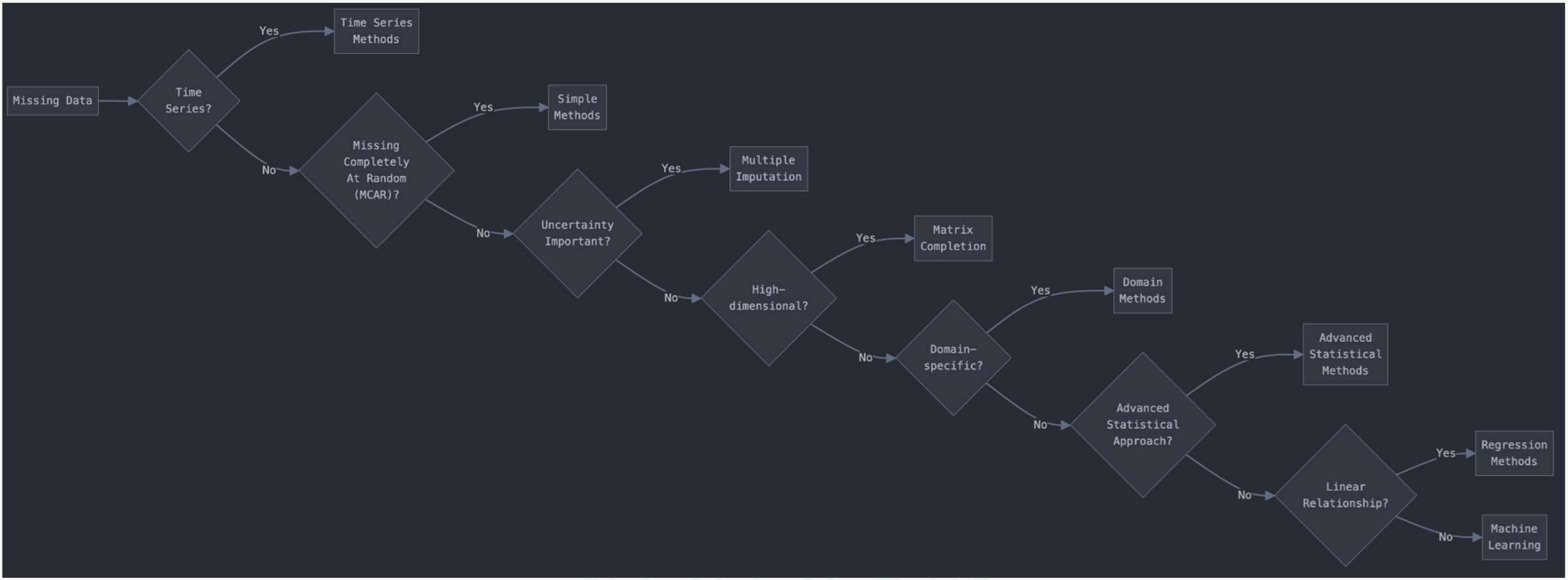
Category	Key Characteristics	Typical Use Cases	Complexity
Simple Imputation Methods	Basic statistical approaches (mean, median, mode, constant)	Quick analysis, MCAR data	Low
Regression-based Methods	Predict missing values based on other variables	When relationships between variables are important	Medium
Machine Learning Methods	Utilize advanced algorithms to impute data	Complex datasets, non-linear relationships	Medium to High
Multiple Imputation Methods	Create multiple plausible imputed datasets	When uncertainty in imputations is crucial	High
Time Series-specific Methods	Account for temporal dependencies	Time series data with missing values	Medium
Matrix Completion Methods	Leverage matrix structure of data	High-dimensional data, recommender systems	Medium to High
Advanced Statistical Methods	Based on statistical theory (e.g., EM algorithm)	When distributional assumptions can be made	High
Domain-specific Methods	Tailored to particular fields or data types	Specialized datasets (e.g., healthcare, finance)	Varies

This is *not* an exhaustive list, nor are these categories unique.

Understand your data (EDA) and your goal, then choose the most appropriate category.

Let's see how we might approach imputation....

Repairing Missing Values: Decision Tree of Strategies



Time Series

Subcategory	Pros	Cons	Use Cases	Complexity	Assumptions
Last Observation Carried Forward (LOCF)	- Simple to implement - Preserves trends	- Can introduce bias - Assumes no change over time	- Short gaps in time series - When recent values are good predictors	Low	Data changes slowly over time
Next Observation Carried Backward (NOCB)	- Simple to implement - Useful for backcasting	- Can introduce bias - Assumes no change over time	- Filling in historical data - When future values are known	Low	Data changes slowly over time
Linear Interpolation	- Preserves overall trend - Works well for short gaps	- Assumes linear change - Can smooth out fluctuations	- Regular time series with short gaps - When trend is approximately linear	Low to Medium	Linear trend between known points
Seasonal Decomposition	- Accounts for seasonality - Preserves cyclical patterns	- Requires sufficient historical data - Can be complex for irregular series	- Seasonal time series - Long-term forecasting	Medium to High	Clear seasonal patterns exist

Simple Imputation

Subcategory	Pros	Cons	Use Cases	Complexity	Assumptions
Mean Imputation	- Easy to implement - Preserves mean of variable	- Reduces variability - Ignores relationships between variables	- Quick exploratory analysis - Large datasets with few missing values	Low	Data is MCAR Mean is representative
Median Imputation	- Robust to outliers - Preserves median of variable	- Reduces variability - Ignores relationships between variables	- Skewed distributions - Presence of outliers	Low	Data is MCAR Median is representative
Mode Imputation	- Suitable for categorical data - Preserves most common category	- May overrepresent majority class - Not suitable for numerical data	- Categorical variables - When preserving proportions is important	Low	Data is MCAR Mode is representative
Constant Value Imputation	- Simple to implement - Can represent 'Unknown' state	- Introduces bias - May distort relationships	- When a logical constant exists - For creating 'Missing' categories	Low	The constant value is meaningful

Multiple Imputation

Subcategory	Pros	Cons	Use Cases	Complexity	Assumptions
Multiple Imputation by Chained Equations (MICE)	- Handles different variable types - Accounts for uncertainty in imputations	- Can be computationally intensive - Assumes MAR	- Mixed data types - When uncertainty quantification is crucial	High	Missing At Random (MAR)
Joint Modeling Multiple Imputation	- Preserves relationships between variables - Theoretically sound	- Can be complex to implement - May struggle with non-normal distributions	- When strong theoretical model exists - Multivariate normal data	High	Multivariate normality
Fully Conditional Specification	- Flexible for various data types - Can handle complex dependence structures	- Can be computationally demanding - Convergence not guaranteed	- Complex missing data patterns - Mixed data types	Very High	Conditional relationships can be specified

Matrix Completion

Subcategory	Pros	Cons	Use Cases	Complexity	Assumptions
Singular Value Decomposition (SVD)	- Captures latent structure - Works well for high-dimensional data	- Assumes linear relationships - Can be sensitive to outliers	- Recommender systems - Image reconstruction	High	Low-rank structure in the data
Nuclear Norm Minimization	- Convex optimization - Theoretical guarantees	- Can be computationally intensive - Sensitive to noise	- Collaborative filtering - When theoretical guarantees are needed	Very High	Low-rank structure with some noise
Alternating Least Squares	- Scalable to large datasets - Can incorporate regularization	- Local optima issues - Sensitive to initialization	- Large-scale recommender systems - When scalability is crucial	Medium to High	Underlying low-rank factorization

Domain Specific

Domain	Method	Description	Pros	Cons	Use Cases
Healthcare	Last Observation Carried Forward (LOCF)	Uses the last known value for future time points	- Preserves individual patient trends - Simple to implement	- Can introduce bias in long-term studies - Assumes no change over time	- Clinical trials with dropouts - Longitudinal health studies
Environmental Science	Spatial Interpolation	Estimates missing values based on nearby measurements	- Accounts for geographical patterns - Useful for spatially correlated data	- Requires good spatial coverage - Can be computationally intensive	- Weather data imputation - Soil property mapping
Finance	Multiple Imputation with Chained Equations (MICE) adapted for time series	Imputes missing financial data considering temporal dependencies	- Preserves time series characteristics - Accounts for uncertainty	- Complex to implement - Computationally intensive	- Stock market data - Economic indicators
Social Sciences	Hot Deck Imputation	Replaces missing values with values from similar respondents	- Preserves distribution of data - Works well for categorical variables	- Requires careful definition of "similar" - Can be biased if not properly implemented	- Survey data with non-response - Census data
Engineering	Physics-based Models	Uses known physical relationships to estimate missing values	- Incorporates domain knowledge - Can be highly accurate	- Requires detailed understanding of the system - May not generalize well	- Sensor data in manufacturing - Structural health monitoring

Advanced Statistical

Subcategory	Pros	Cons	Use Cases	Complexity	Assumptions
Expectation-Maximization (EM) Algorithm	- Statistically efficient - Handles complex missing data patterns	- Can be slow to converge - Sensitive to initial values	- When distributional assumptions hold - Maximum likelihood estimation	High	Specific probability distribution
Maximum Likelihood Estimation	- Unbiased under MAR - Efficient use of available information	- Computationally intensive - Requires large samples	- When efficiency is crucial - Well-defined statistical models	Very High	Missing At Random (MAR)
Bayesian Methods	- Incorporates prior knowledge - Provides uncertainty quantification	- Can be computationally demanding - Sensitive to prior specifications	- When prior information is available - Small sample sizes	Very High	Ability to specify priors

Regression-Based

Subcategory	Pros	Cons	Use Cases	Complexity	Assumptions
Simple Regression Imputation	- Accounts for relationships between variables - Easy to interpret	- Assumes linear relationships - Can underestimate variability	- When clear linear relationships exist - For continuous variables	Medium	Linear relationships between variables
Stochastic Regression Imputation	- Adds appropriate variability - More realistic imputations	- More complex than simple regression - Can be computationally intensive	- When preserving variability is important - Larger datasets	Medium to High	Linear relationships with added noise
Multiple Regression Imputation	- Can handle complex relationships - Uses information from multiple predictors	- Risk of overfitting - Sensitive to multicollinearity	- When multiple predictors are relevant - Complex datasets	Medium to High	Linear relationships with multiple predictors

Machine Learning

Subcategory	Pros	Cons	Use Cases	Complexity	Assumptions
K-Nearest Neighbors (KNN)	- Can capture non-linear relationships - Works for both categorical and continuous data	- Sensitive to the choice of K - Can be slow for large datasets	- Mixed data types - When local patterns are important	Medium	Similar observations have similar values
Decision Tree-based Methods	- Can handle non-linear relationships - Work well with mixed data types	- Risk of overfitting - Can be unstable	- Complex relationships - Hierarchical data structures	Medium to High	Hierarchical structure in data
Random Forest Imputation	- Handles complex relationships - Robust to outliers	- Computationally intensive - Less interpretable	- Large datasets - When accuracy is critical	High	Complex, potentially non-linear relationships
Neural Network Imputation	- Can capture very complex patterns - Flexible for various data types	- Requires large amounts of data - Black box nature	- Big data scenarios - When underlying patterns are unknown	Very High	Complex patterns exist in the data

Alternate Organization

Deletion Methods: Techniques that remove cases or variables with missing data from the analysis.

Imputation Methods: Approaches that fill in missing values with estimated data before analysis.

Model-Based Methods: Techniques that incorporate missing data handling directly into the analysis model, often using maximum likelihood or Bayesian approaches.