Handling Missing Data

① Remove all missing rows

② (or) Replace with generic substitute values (mean / median / S.D.) / most frequent if value is categorical

③ Imputation :- Estimate a probability model for the missing variable & replace the missing value with one or more samples from probability model.

→ Types of Missing Data

Data may be missing for a variety of reasons.
→ corrupt during its transfer or storage

① MCAR – Missing Completely At Random
– prob of an observation being missing does not depend on observed or unobserved measurements.

Ex:- movie rating from users, since some movies are more popular than others, some movies may not have ratings = $\boxed{\text{NOT MCAR}}$

② MAR → given the observed data, the probability that data is missing does not depend on unobserved data

Ex:-

Y	Gender	Race	Income
	M	Asian	880
	F	Italian	
	M	American	

If missing data depends on Gender/Race then it is MAR

If not its MCAR.

Missing Data & R

→ Dealing with Outliers → Trimming - Remove outliers
→ Winsorization - Shrink outliers
→ Robustness - Keep the outliers & analyze data using a robust procedure.

Detecting Outliers → (1) values below the alpha percentile (α) & $100 - \alpha$ percentile
→ (2) values more than c times std. deviation from the mean.
→ follows 1st technique (when we assume data is gaussian)

→ Issue: outliers can affect mean and other calculations.
→ So to avoid this remove most Extreme
(or) just use percentiles (more robust)

(Lecture-8) DVA

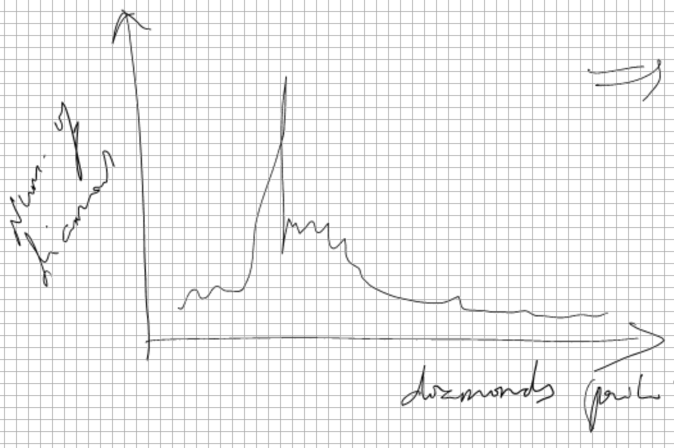
→ Data Transformations: Skewness & Power Transformations

Data is generally drawn from a highly skewed distribution and that is not well described by a common distribution.
→ A simple transformation may map the data to a form that is well described by common distribution.
↳ Once transformed a suitable model can then be fitted to data.

→ Power Transformation family (Read from video lecture)

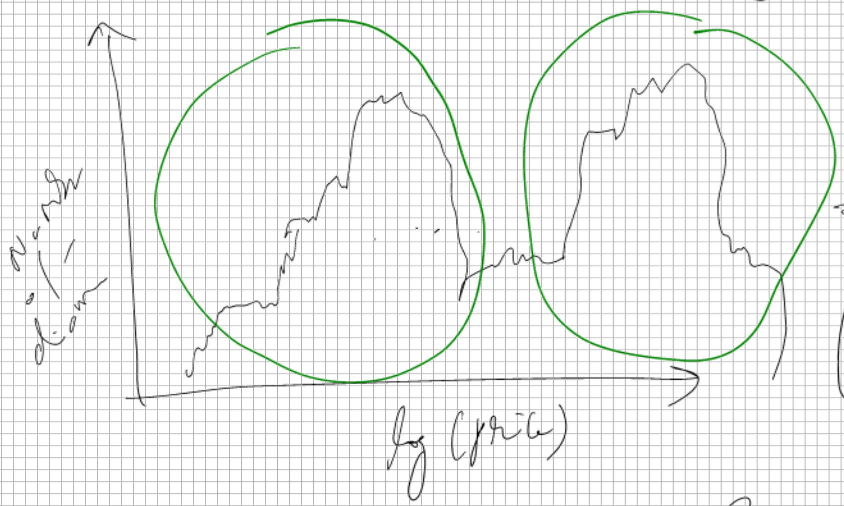
$$f_\lambda(x) = \begin{cases} (x^\lambda - 1)/\lambda & \lambda > 0 \\ \log x & \lambda = 0 \\ -(x^\lambda - 1)/\lambda & \lambda < 0 \end{cases} \qquad x > 0$$

Ex → data transformation helps to form **Insights**



⇒ As price ↑ no. of diamond ↓

Transform data



⇒ Transformed data looks like it is $\boxed{\text{Bi-modal}}$ in nature.

→ **Data Transformation** :- Binning

Data → Numerical
→ Ordinal (order – ✓, Numerical variables are a special case of ordinal)
→ Categorical

Ex:- → Seasons (have an order but can't subtract winter from spring)
↳ Customer satisfaction rating (bad, good, Awesome)

→ **Data Transformation** → **Indicator Variables**

→ Indicator variables are used in conjunction with Binning.

Bin 1-5 ⇒ 0 ; 5-16 ⇒ 1 , 16→ ⇒ 2

Ex:-

data	Bin	Indicator
1	0	000
4	0	000
15	1	010
20	2	100

MCAR Vs MAR

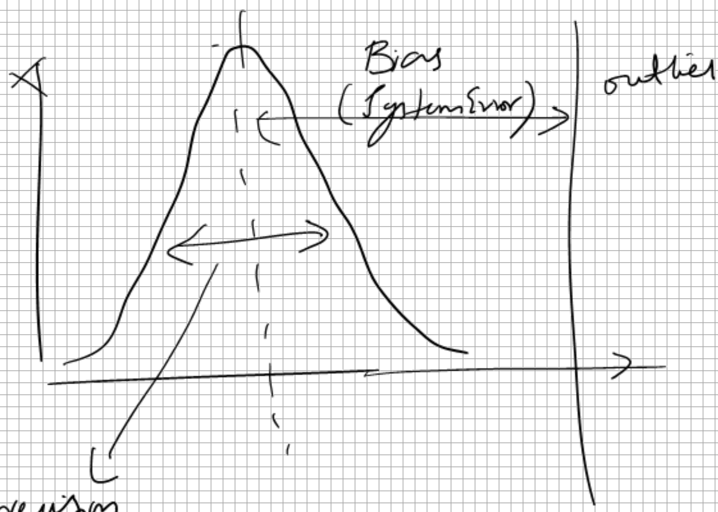
↳ Any of the 3 techniques used for handling missing data are reasonable, though some are better.

For MAR :- The 1st two techniques may introduce some bias, the 3rd technique is maybe reasonable depending on how well the probability model is built.

Outliers Types
- → Corrupted Values (human Err)
- → Unlikely Values
  ↳ are substantially unlikely values given our modelling assumptions.

Robustness :- lack of sensitivity of data analysis procedures to outliers.

"mean" gets affected by outlier, median doesn't.

→ 

Precision (uncertainty) also referred to as random Error.

→ **Handling outliers**
- → Truncating
- → Winsorization - replace it with acceptable Extreme values
- → Robustness :- Keep outliers in data, but using robust procedure

Ex:- if we want summary of value, we may use median rather than mean.

i.e median is more robust to outliers.

```
# winsorize ( df $ x1[1:100] )
  ↳ library (robustHD)
```

Tall Data : one or more columns are Keys
Atleast one column is value

↳ is convenient for adding new records incrementally & for removing old records.
i.e simply adding additional rows.

→ melt() : converts wide data to tall data.

↳ Not Easy for summarizing.

Wide Data : Represents in multiple columns the information that tall data holds in multiple rows.

↳ simpler to analyze.
- no need to pass over all data, jump to specific location.

↳ to add/remove Entries.

Ex:

2015/01/01	apples	200
2015/01/01	oranges	150
2015/01/02	apples	220
2015/01/02	oranges	130

Tall data

⇒

Date	apples	oranges
2015/01/01	200	150
2015/01/02	220	130

Wide Data.

→ melt - wide → tall (reshape2 package)

→ acast / dcast inverse of melt

↳ returns an array ↳ returns a dataframe.

↳ : tall data to wide data.