Topic E
Data Integration & ETL Process

Data Integration & ETL Process

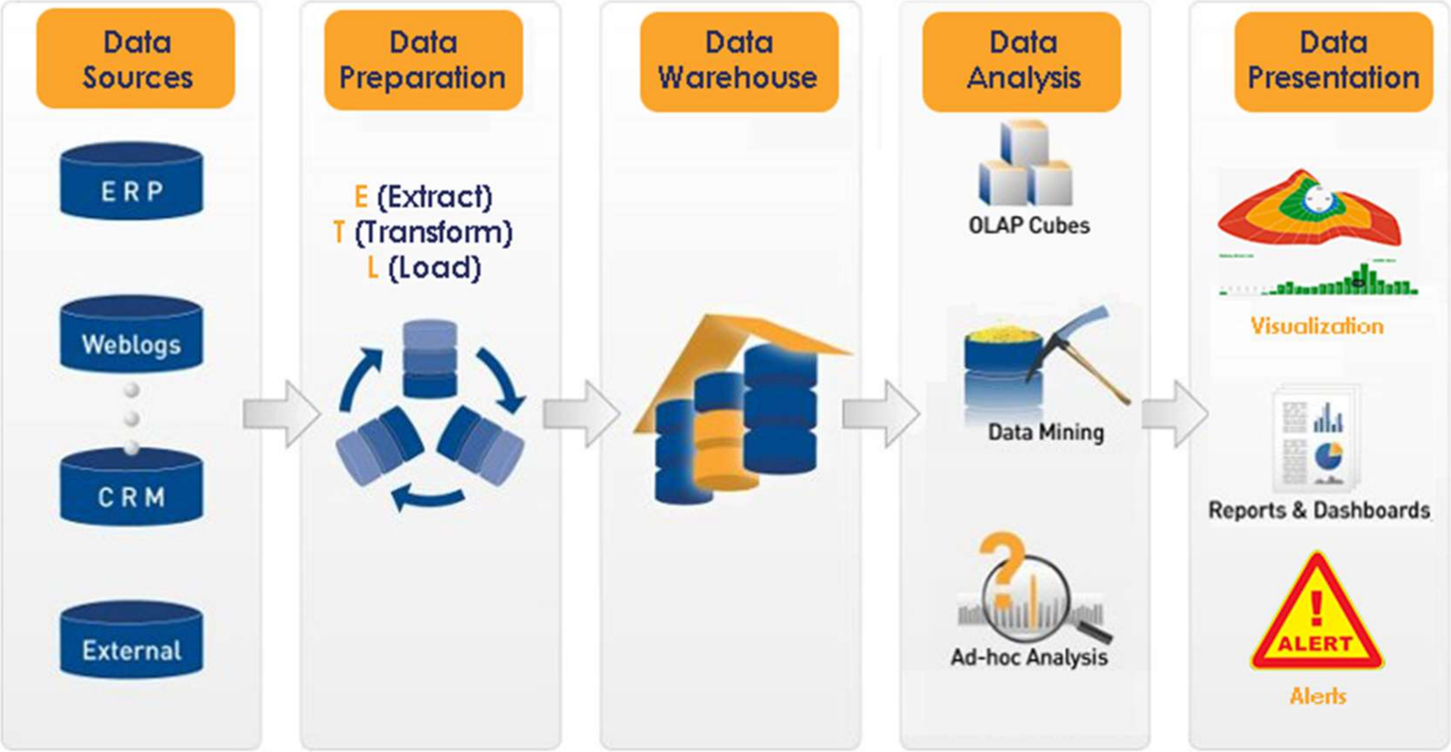
CONTENT

- Components of a BI Solution
- Describe the ETL process
- The importance of Data Quality
- Data Cleansing Methodology

Components of a BI Solution

A typical BI solution in an enterprise consists of the following components:

1. Data Sources
2. Data preparation
3. Data Warehouse
4. Data Analytics
5. Data presentation



What is ETL?

- ETL (Extract, Transform and Load) is a process in data warehousing responsible for pulling data out of the source systems and placing it into a data warehouse.
  - The ETL process is critical to a successful data warehouse
  - ETL must ensure the data loaded into the data warehouse is high-quality, accurate, relevant, useful and accessible
  - ETL is the most time-consuming phase in building a data warehouse as routines must be developed to select the required fields from numerous sources of data

What is ETL? – The tasks

- **Extract** : the process of reading data from a source database
- **Transform** : the process of converting the extracted data from its previous form into the form it needs to be in so that it can be placed into another database.
- **Load** : the process of writing the data into the target database.

The ETL process is often represented by gears icons

What is ETL? - Extraction

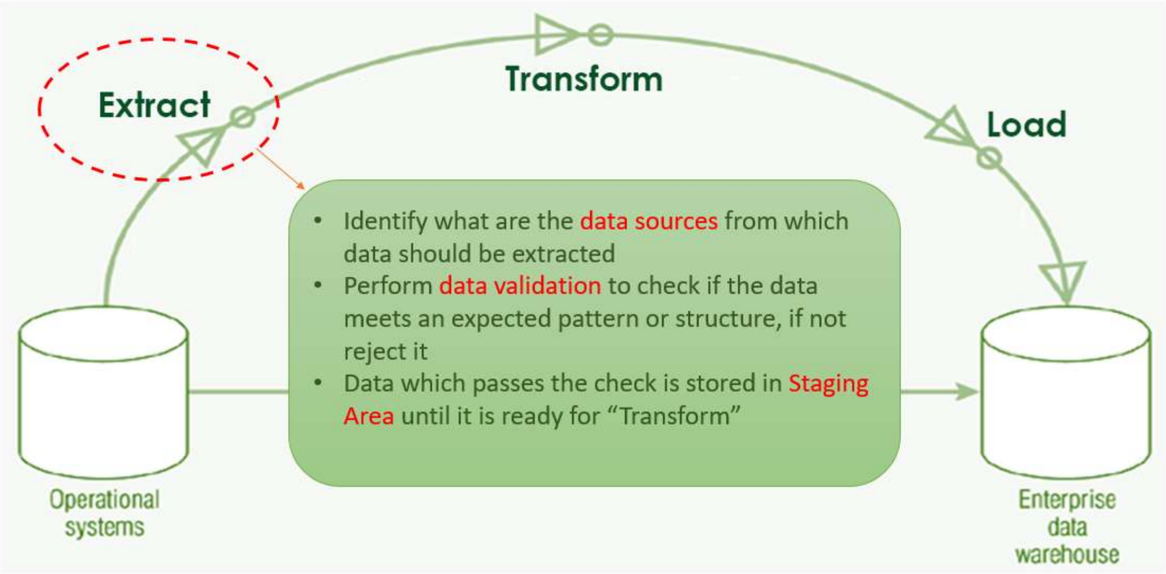
In this phase, the required data is identified and extracted from different sources

The extracted data is then held in temporary storage (Staging Area) until the "Transform" and "Load" phases can be executed

During extraction, validation rules are applied to test whether data has expected values essential to the data warehouse

Data that fails the validation is rejected and further processed to discover why it failed validation and remediate if possible.

E.g. CustomerID in Orders table that does not exist in Customers table, empty fields etc



- Identify what are the data sources from which data should be extracted
- Perform data validation to check if the data meets an expected pattern or structure, if not reject it
- Data which passes the check is stored in Staging Area until it is ready for "Transform"

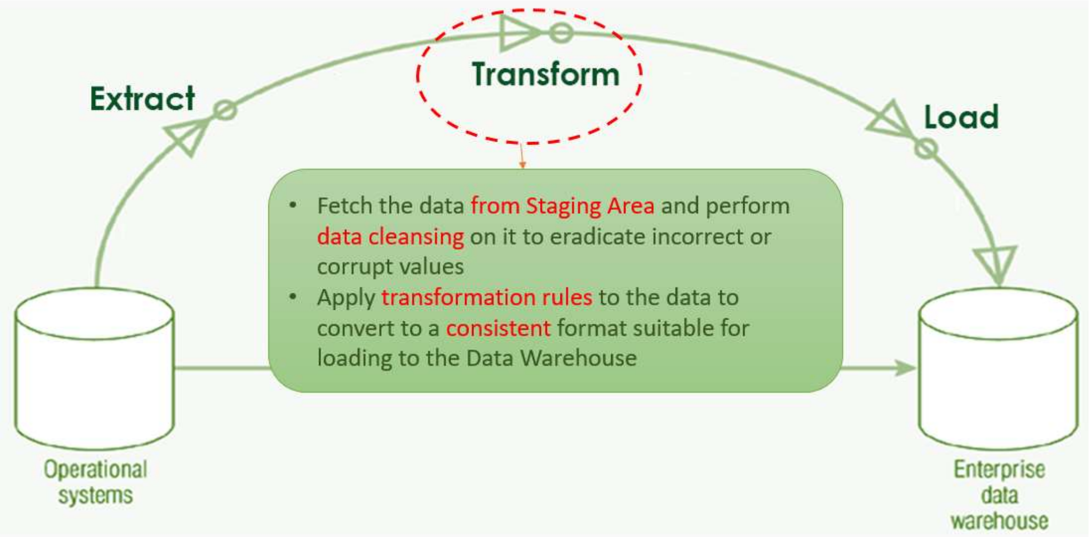
What is ETL? – Transform

In this phase, data is fetched from the staging area, “cleansed” and transformed.

Data cleansing or data scrubbing is the process of detecting and correcting (or removing) data anomalies to improve the data quality.

Examples of data anomalies or inaccurate records are:

- A delivery date is earlier than the order date
- A phone number that contains alphabetical characters
- A birthdate that is in the future
- A missing value e.g. postal code
- Order record refers to a customer id that is missing



- Fetch the data from Staging Area and perform data cleansing on it to eradicate incorrect or corrupt values
- Apply transformation rules to the data to convert to a consistent format suitable for loading to the Data Warehouse

What is ETL? – Data Quality

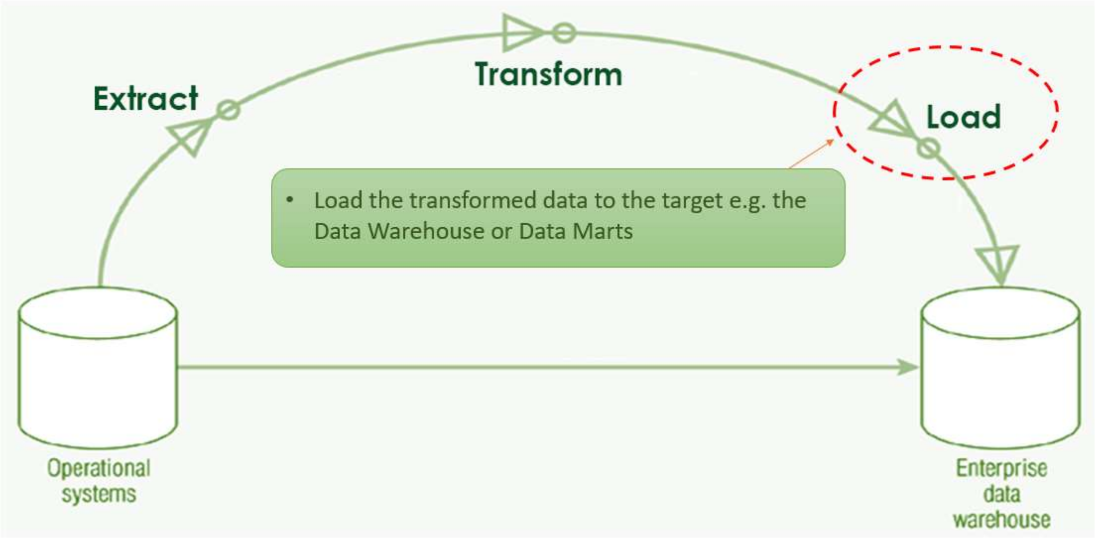
Some examples of “Dirty Data”

Incorrect Data	Data with values that are obviously incorrect, e.g. Age = 200
Inaccurate Data	Data with values that are correct but inaccurate e.g. Singapore Polytechnic, Malaysia
Business Rules Violation	Data sets that do not comply to business rules, e.g. Payment Types can only be Cash/CreditCard/Cheque but data has “Card”
Inconsistent Data	In one table, customer is called “Mary Tan”, another table, is “Tan Mary”
Incomplete Data	Marketing segmentation data missing because gender of customer was not captured at point of purchase
Non-integrated Data	Customer A with Primary Key 123 on one system and Primary Key 321 on another system

What is ETL? – Load

In this phase, data is loaded unto the end target, usually a data warehouse or data mart This process varies widely depending on the requirements of the organization.

- Some data warehouses may add new data in an historical form at regular intervals — for example, hourly.
- Some data warehouses may overwrite existing information with cumulative information
- Updating extracted data is frequently done on a daily, weekly, or monthly basis



- Load the transformed data to the target e.g. the Data Warehouse or Data Marts

Data Quality

- Data cleaning is one those things that everyone does but no one really talks about
- However, proper data cleaning can make or break your project. Professional data scientists usually spend a very large portion of their time on this step.
- Why? Because of a simple truth in machine learning:
  - **Better data beats fancier algorithms.**

Data Quality- Better Data > Fancier Algorithms

- **Remove Unwanted observations**
- The first step to data cleaning is removing unwanted observations from your dataset.
- This includes **duplicate** or **irrelevant** observations.

Duplicate observations

Duplicate observations most frequently arise during **data collection**, such as when you:

•Combine datasets from multiple places

•Scrape data

•Receive data from clients/other departments

Irrelevant observations

Irrelevant observations are those that don't actually fit the **specific problem** that you're trying to solve.

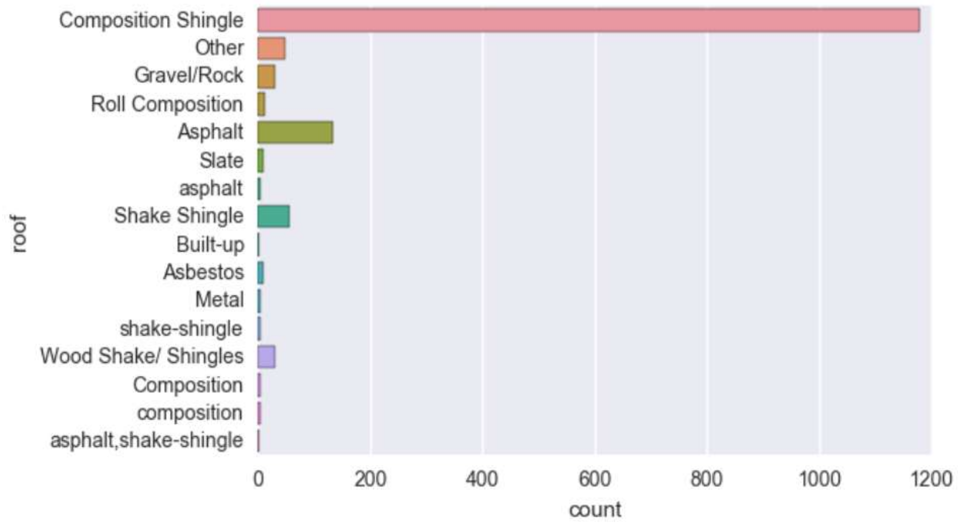
•For example, if you were building a model for Single-Family homes only, you wouldn't want observations for Apartments in there.

•This is also a great time to review your charts from Exploratory Analysis. You can look at the distribution charts for categorical features to see if there are any classes that shouldn't be there.

•Checking for irrelevant observations **before engineering features** can save you many headaches down the road.

Data Quality– Fix Structural Errors

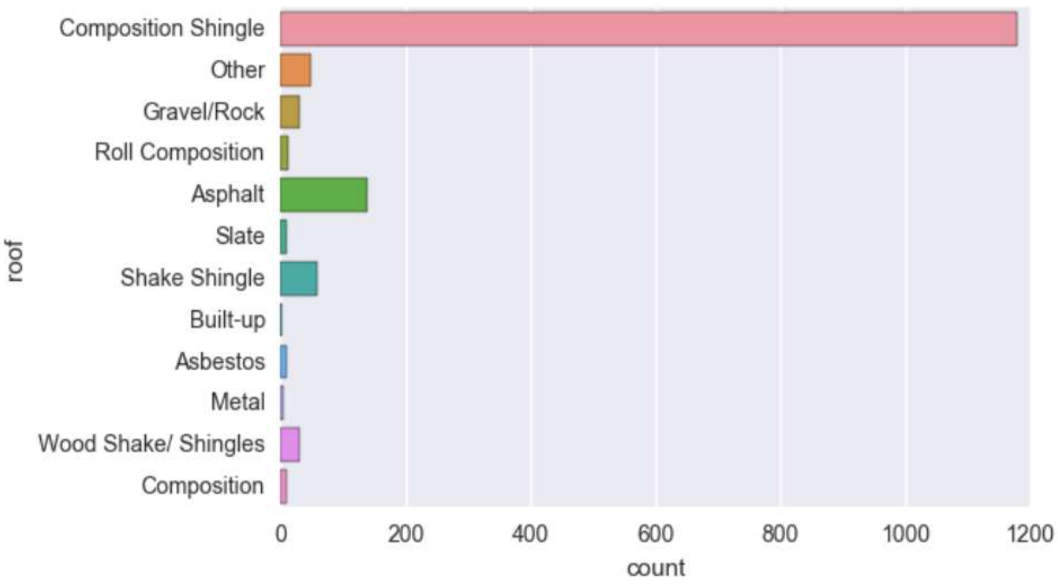
- The next bucket under data cleaning involves fixing structural errors.
- Structural errors are those that arise during measurement, data transfer, or other types of **"poor housekeeping."**
- For instance, you can check for **typos** or **inconsistent capitalization.** This is mostly a concern for categorical features, and you can look at your bar plots to check



As you can see:
•'composition' is the same as 'Composition'
•'asphalt' should be 'Asphalt'
•'shake-shingle' should be 'Shake Shingle'
•'asphalt,shake-shingle' could probably just be 'Shake Shingle' as well

Data Quality– Fix Structural Errors

- After we replace the typos and inconsistent capitalization, the class distribution becomes much cleaner
- Finally, check for **mislabeled classes**, i.e. separate classes that should really be the same.
- e.g. If ’N/A’ and ’Not Applicable’ appear as two separate classes, you should combine them.
- e.g. ’IT’ and ’information_technology’ should be a single class.



Data Quality- Filter Unwanted Outliers

- Outliers can cause problems with certain types of models. For example, linear regression models are less robust to outliers than decision tree models.
- In general, if you have a **legitimate** reason to remove an outlier, it will help your model's performance.
- However, outliers are **innocent until proven guilty.** You should never remove an outlier just because it's a "big number." That big number could be very informative for your model.
- We can't stress this enough: you must have a good reason for removing an outlier, such as suspicious measurements that are unlikely to be real data.

Data Quality- Handle Missing Data

- Missing data is a deceptively tricky issue in applied machine learning.
- First, just to be clear, **you cannot simply ignore missing values in your dataset.** You must handle them in some way for the very practical reason that most algorithms do not accept missing values.
- **"Common sense" is not sensible here**
- Unfortunately, from our experience, the 2 most commonly recommended ways of dealing with missing data actually suck.
- They are:
  - **Dropping** observations that have missing values
  - **Imputing** the missing values based on other observations

Data Quality- Handle Missing Data

- Dropping missing values is sub-optimal because when you drop observations, you **drop information.**
- The fact that the value was missing may be informative in itself.
- Plus, in the real world, you often need to make predictions on new data even if some of the features are missing!
- Imputing missing values is sub-optimal because the value was originally missing but you filled it in, which always leads to a loss in information, no matter how sophisticated your imputation method is.
- Again, **"missingness"** is almost always informative in itself, and you should **tell your algorithm** if a value was missing.
- Even if you build a model to impute your values, you're not adding any real information. You're just reinforcing the patterns already provided by other features.

Missing data is like missing a puzzle piece. If you drop it, that's like pretending the puzzle slot isn't there. If you impute it, that's like trying to squeeze in a piece from somewhere else in the puzzle.

Data Quality- Handle Missing Data

- In short, you should always tell your algorithm that a value was missing because **missingness is informative**.
- So how can you do so?

Missing categorical data

The best way to handle missing data for *categorical* features is to simply label them as 'Missing'!

•You're essentially adding a *new class* for the feature.

•This tells the algorithm that the value was missing.

•This also gets around the technical requirement for no missing values.

Missing numeric data

For missing *numeric* data, you should **flag and fill** the values.

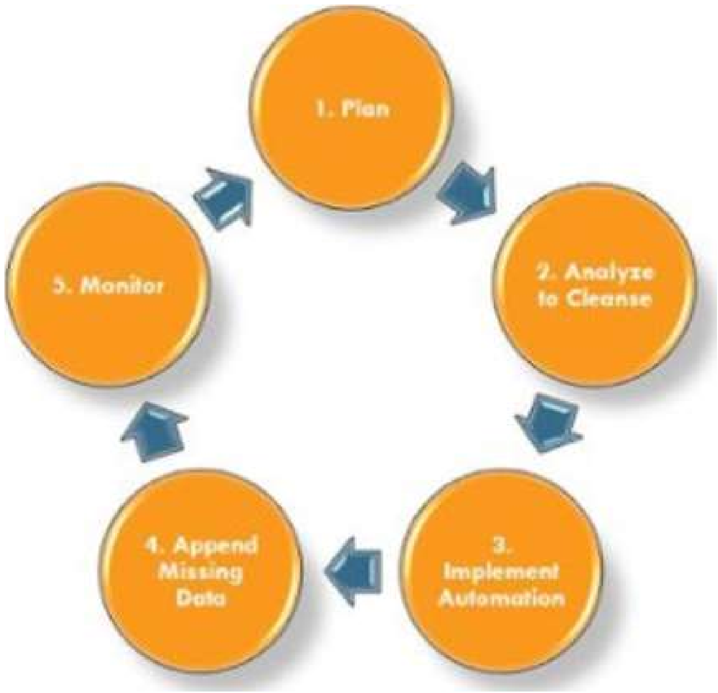
1.Flag the observation with an indicator variable of missingness.

2.Then, fill the original missing value with 0 just to meet the technical requirement of no missing values.

By using this technique of flagging and filling, you are essentially **allowing the algorithm to estimate the optimal constant for missingness**, instead of just filling it in with the mean

The key is to tell your algorithm that the value was originally missing.

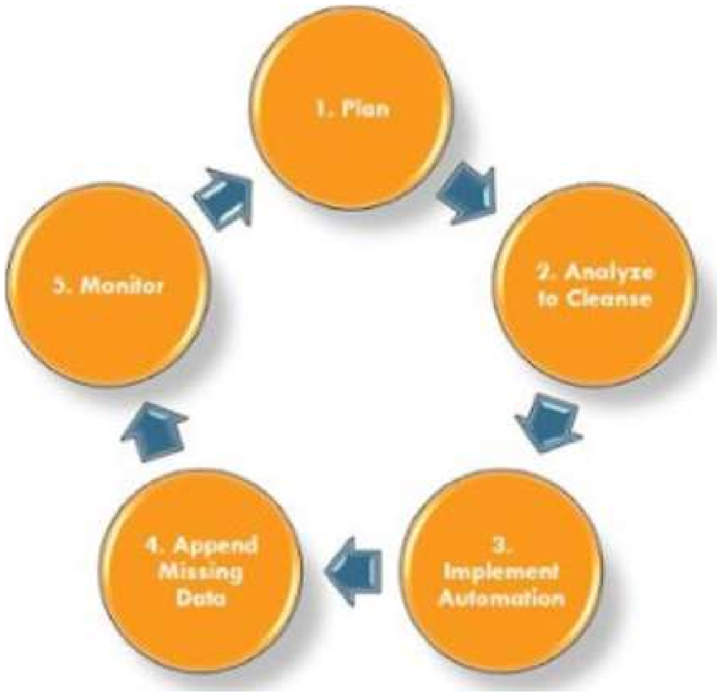
Data Cleansing Cycle (Methodology)



1 Plan

- First off, you want to identify the set of data that is critical.
- Focus on high priority data, and start small
- Create and put into place specific validation rules at this point to standardize and cleanse the existing data as well as automate this process for the future.

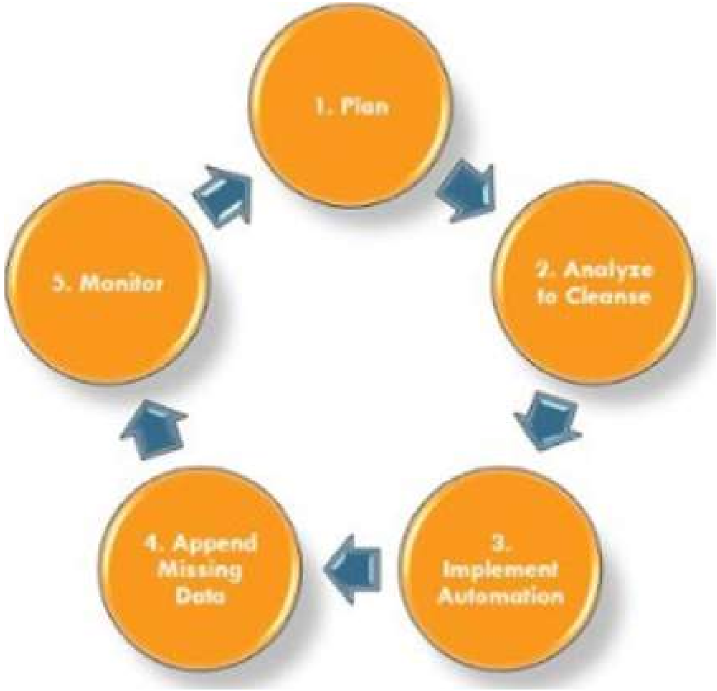
Data Cleansing Cycle (Methodology)



2 Analyze to Cleanse

- Go through the data you already have in order to see what is missing, what can be thrown out, and what, if any, are gaps between them.
- Identify a set of resources to handle and manually cleanse exceptions to your rules.
- The amount of manual intervention is directly correlated to the amount of acceptable levels of data quality

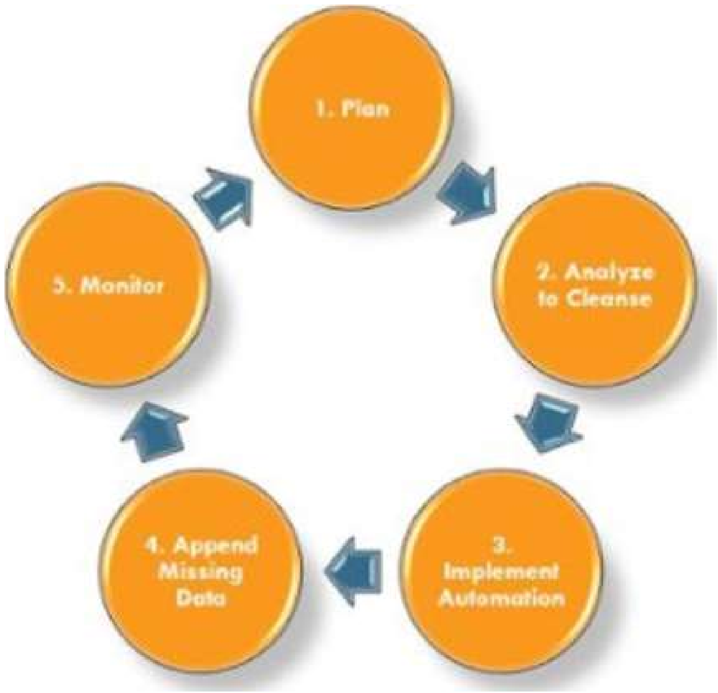
Data Cleansing Cycle (Methodology)



3 Implement Automation

- Begin to standardize and cleanse the flow of new data as it enters the system by creating scripts or workflows. These can be run in real-time or in batch (daily, weekly, monthly).
- These routines can be applied to new data, or to previously keyed-in data.

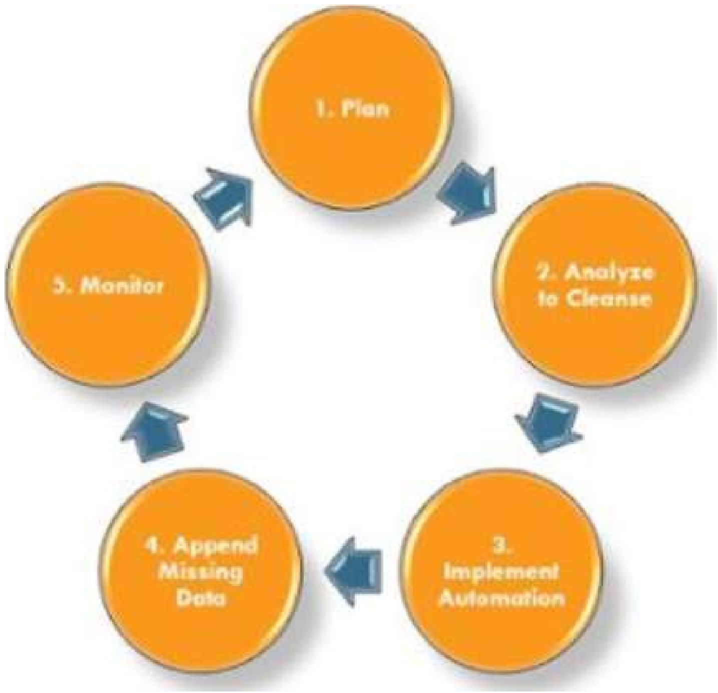
Data Cleansing Cycle (Methodology)



4 Append Missing Data

- It is important especially for records that cannot be automatically corrected. Examples of this are emails, phone numbers, industry, company size, etc.
- It's important to identify the correct way of getting a hold of the missing data,

Data Cleansing Cycle (Methodology)



5 Monitor

- Set up a periodic review so that you can monitor issues before they become a major problem.
- Bring the whole process full circle. Revisit your plans from the first step and reevaluate.
- Can priorities be changed? Do the rules you implemented still fit into your overall business strategy?
- Conduct periodic reviews to make sure that your data cleansing is running with smoothness and accuracy.