**Unit 4 Lecturecast – Cleaning (and reading Kazil, J & Jarmul, K. (2016) Data Wrangling with Python. O'Reilly. Media Inc.)**

**Chapters 6, 7**

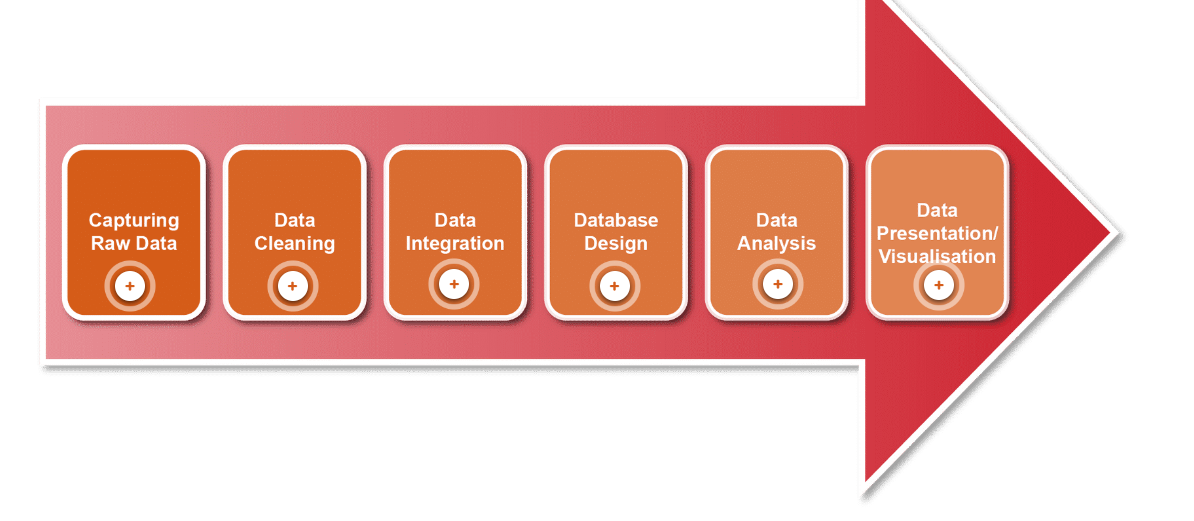
This lecturecast will help you to:

* understand and perform the basics and factors of data cleaning;
* evaluate the critical outcomes for the data design and automation;
* compile and document Python scripts;
* compile data sets and convert data into different formats.

Cleaning is quantitative checking of to check data accuracy. Involves examining, coding, inputting and handling missing or incomplete. Data scraping and screening can be used for the incomplete data.

Huxley et al (2020)

**Data Management Pipe Line**

****

**EMC, 2015)**

**Capturing raw data** – extraction from sources (which could be files, pages or software) – integrity of source must be checked.

**Data cleaning –** checking data, duplicate and redundancy removal plus mapping data headers for relevance to business logic.

**Data Integration -** Drawing data together for the user.Could be text summarisation or document relevant techniques

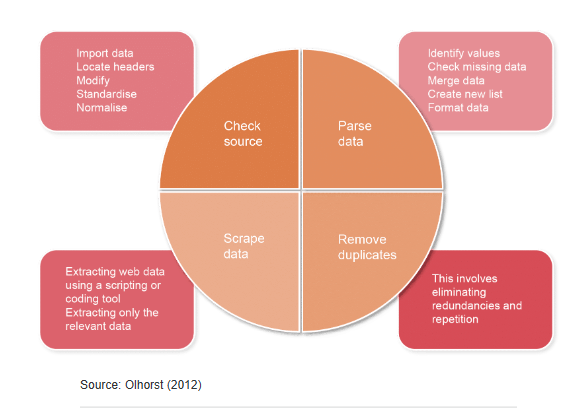
**Database Design –** Modelling data using DMS (database management system) for storage, manipulation and retrieval.

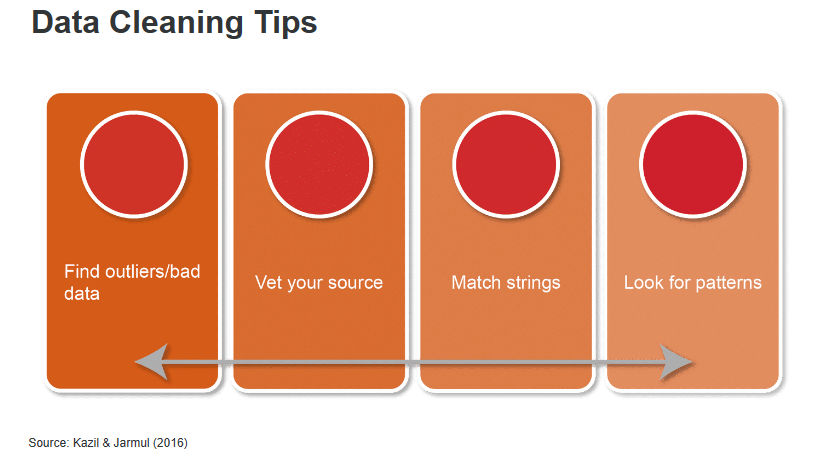
**Data Analysis -** Deciphering data to devise new meaning relevant to the organisation.

**Data presentation / viz –** Simplifying data (pictorial) to make it easily understood

**Data Clean Up Basics and Tips**

The data clean up basics are captured in four key stages involving checking the source of data, parsing data, removing duplicates and scraping the data.





**Exercise 1 – Data cleaning with Python - reflections and reading (**Kazil, J & Jarmul, K. (2016) Data Wrangling with Python. O'Reilly. Media Inc.)

The activity works through replacing coded / short format text headers with human readable ones from a second CSV file. Using DictReader library in python then a list generator function.

This shows an example of writing a clean python script to automate data cleaning:

**“scripts are best to be explicit, clean, and as practical as possible. This is the core of the Zen of Python.”**

The process matches most of the data but two headers remain unmatched – this demonstrates a decision point in data cleaning – are these headers or data needed – If so - what can we do to clean or match them? Otherwise, we could drop if they are not essential for the analysis. Is the effort worth the output, can you make cleaning reputably to save time in the future.

**Formatting data**

One of most common cleanup methods, could be to turn from machine readable to human readable for reporting or so that data is in particular format for machine reading or API to work. Many methods of formatting in Python.

Date formatting can be done easily in Python, a very common process. There are many applications.

**Outliers and bad data**

When considering outliers and bad data, the aim is to clean data not manipulate or change it. Time should be taken to explore and consider options, with removal for clear reason / intention to normalise it.

Some easy ways to check data:

* First clue is the source – how was data collected, by who, has it already been cleaned or processed? Source can determine bias in dataset which could affect conclusions.
* Potential errors when looking whether any data points do not fit. Are values missing, do data types not make sense, are there many not applicable responses which would distort a sample.

Ask, can we account for the answers, such as NA or missing values. For example is it a female only question and survey is for everyone so half would be missing.

Checking for variation in data types in rows and columns can help us understand where there could be errors and devise a cleaning strategy, are the questions important? What are the errors, are they missing values, poorly inputted or something else. Once we understand the data can make a plan to fix it.

**Duplicates**

Many methods of duplicate removal, identifying a unique key in data can be important. This could be a reference number but when no obvious unique key due to messy data will have to determine one, example given includes birth date and address.

**Fuzzy matching**

Allows determination of whether two strings are the same, not as in-depth as natural language processing or or machine learning it can help relate data items with similar meaning. Various python libraires support this. Could be unclean data that has been sloppily input such as sloppily input or deviations in synaptic or format. Knowledge of data and complexities can help with assessing the quality of the fuzzy matches.

**RegEx Matching**

RegEx means Regular Expression. Allow computer to match, find or eliminate patterns in data defined in the code. For example, identify contact details from a string of text. Regex can become complex and hard to read but they are useful.

Regex invaluable for parsing really messy data like from webscraping, can create patterns then identify data with them.

**Duplicate records**

Sometimes may wish to combine duplicate records if they contain different information. If the dataset has exact duplicate rows, likely to just remove. An example could be combining individuals interviewed from one household into household data.

**Chapter 8: Data Cleanup: Standardizing and Scripting**

**Normlaising and Standardising data**

Cleanup of data can be automated; data can be standardized and normalised. This means calculating new values using the values you currently have, or it might mean applying standardization across data.

Statistically, normalisation often means calculating new values from a dataset to a standard scale. Foe example scores across a season given a value of between 0 – 1. Normalize can help see percentiles across different groups (or cohorts). Can help with identification of normal range and outliers.

Standarization is basically what is the normal range and what is outside. Could be standard deviation.

Sometimes these both require the removal of outliers. Helps to see patterns and distribution of data. Sometimes known as trimming of outliers.

**Determining what data cleanup is right for your project**

Depending on nature of data and how often it will be analysed, you might choose a different path for your cleanup. Full scripting may not be possible if data is haphazard or infrequently used, scripting may take more time than worth.

If cleanup onerous, may want to create a repository of helper scripts, so even if can’t fully script can use these functions to process faster. For example, import or duplicate cleanup that are reusable.

**Scripting your cleanup**

If cleanup has a determined pattern and is unlikely to change, could scripty entire cleanup process.

Key part of this is making sure code is clear and readable in terms of function naming, comments – make it clear to you and others (Zen of Python rules! – explicit, clear and practicable as possible)

Documentation of code is hugely important for self and team – what makes sense to you or now, may not in the future. Also, auditable.

Good practice to describe step by step what code is doing in a readme.md

In simple terms, a script will group many lines of data into one script.

**Automation and Scaling**

Automating all parts of data pipeline from collection, analysis and presentation is important. This ensures key data details are captured, data parameters can be input via analysis tools such as IBM Watson and Cognos for kicker processing. Charts can be produced directly from this process.

Automation must be clear and focused – document the steps with outcome and inputs identified. Process should be designed to deal with most likely errors and issues with the process.

**Considerations when cleaning data for organisations:**

* Understand data and system limitations in the context of the organisation (Social-technical and organisational context\*)
* Implement different forms of data models and architecture
* Consider required and available technical resources
* Consider impact, limitations and opportunities for the project / data
* Security implications, GDPR etc

**\*** Social-technical and organisational context = culture and environment, traditions, norms, policies that are a barrier to innovation, change management issues, compatibility with service and data architecture

**Data Models –** representation of data relationships in an organisation, capture high level view of how data processed and stored

Vs **Database Architecture** – data design and how data is stored and deployed on a platform.Attributes, operation and functions performed on the data.