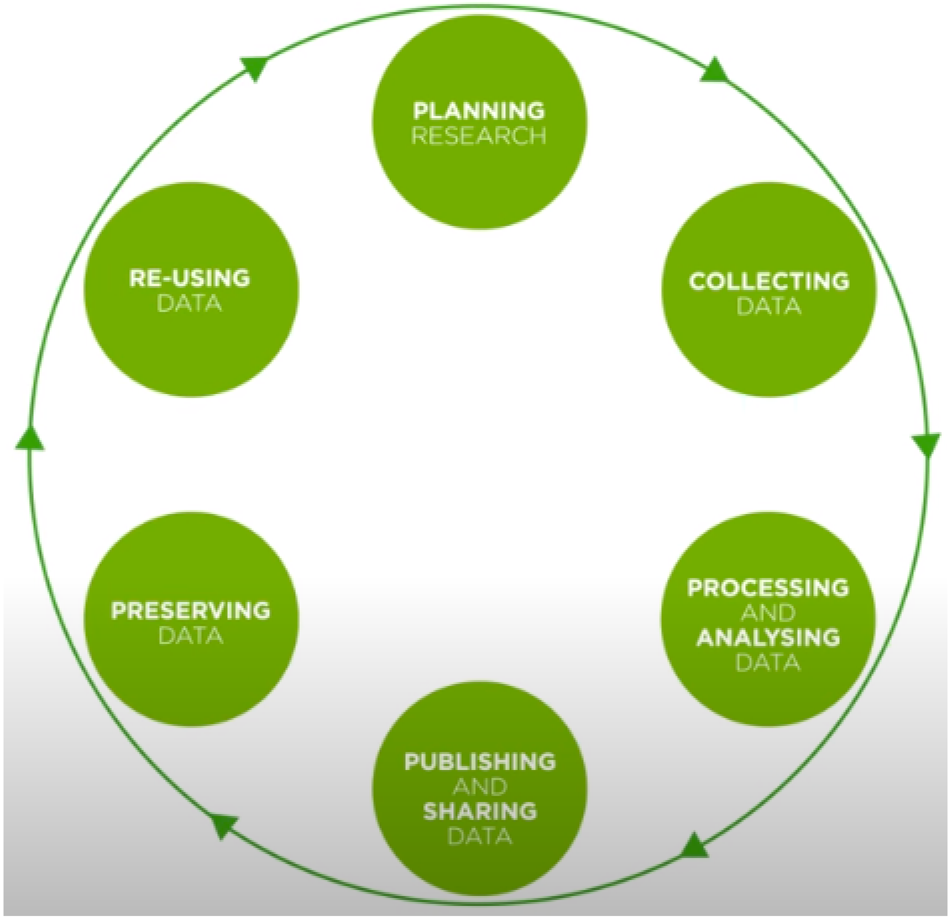
wk1:

data management

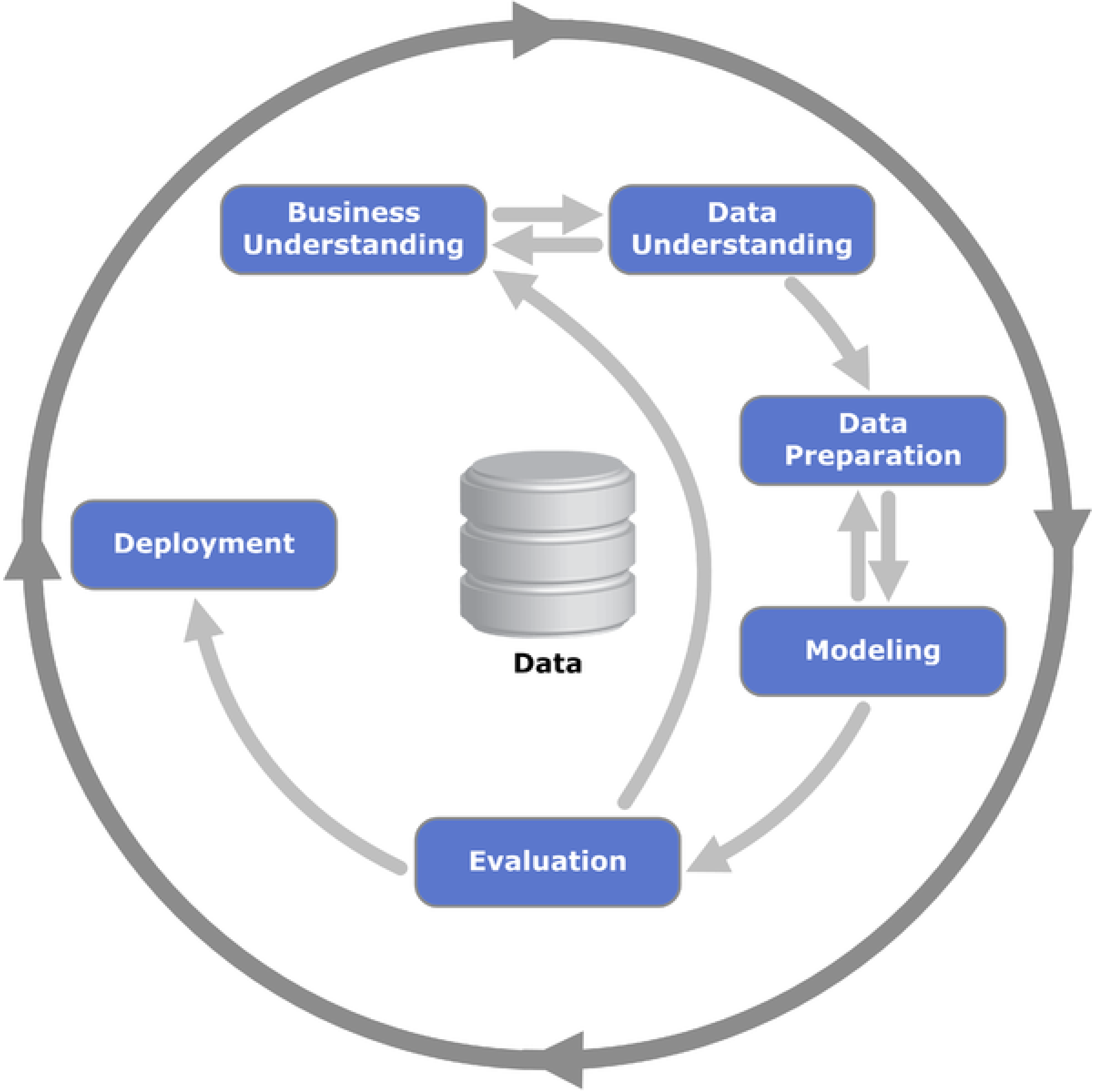
A research data management lifecycle:



other Data managament cycles:

- Domains can have specific data requirements. Check out the data lifecycle from the US Geological Survey.
- Enterprise Data Management
- CRISP-DM (data mining)

CRISP-DM:



JUPYTER NOTEBOOKS...

W1 quiz: https://www.futurelearn.com/courses/data-management-and-visualisation-data-cleaning-and-data-quality/3/steps/1600139/quiz/introduction

---

w2

Data Quality

- High-quality data is free from both errors and artefacts
- An error is data that is missing or lost due to the capture process, cant be recovered
- An artefact is something that has been introduced into the dataset during the gathering, processing, integration or cleaning activities.
- poor data is due to individual or collective issues (Piegorsch, 2015)
- issues get amplified with big data (the 4 V's)

1. Variety: **diversity** of data sources brings abundant data types and complex data structures and increases the difficulty of data **integration**.

2. Volume: significant time is required to fully **explore** and understand the data.
3. Velocity: The data **changes quickly** and the "timeliness" of data is very short so judgements on consistency need to happen rapidly.
4. Veracity is a **measure of data quality** and the **other 'Vs' combine to decrease consistency and accuracy** of very large, complex data sets.

types of errors

- formatting issue (ie.., extra whitespace)
- content issues (i.e., duplicate entries, outliers)

set of core questions and actions:

- check source file and understand format
- review availbe documentation
- Build up your set of core questions. For example, I'm always suspicious of datetime values; currency; NULL values; strings/integers/floats and matching the count of entries between files or tables.
- Implement sanity checks

Do you have or can you find an example of harm or loss caused by data errors?
https://en.wikipedia.org/wiki/Mars_Climate_Orbiter

Measuring data quality: (metrics)

Other practical means to judge data quality include:

- Inventory (expensive)
- Using a proxy measure such as tracking customer complaints
- Applying formal measures of accessibility
- Using test cases with known results and checking for glitches or errors in analysis
- Successfully completing an end-to-end process (e.g., data ingestion, processing, indexing, querying and summarisation)

Causes of data errors:

Data errors in the Gathering Phase:

Gathering:
• Capture
• Import
• Survey

- Recall that errors are generally missing data that cannot be recovered.

1. Data delivery issues such as transmission problems that may result in loss of network connectivity
2. Manual entry errors or artefacts caused by typos,
3. Poor survey or interface design: e.g. what colour is your toothbrush?

4. No standards for format or controlled vocabulary for fields, for example: DCU or Dublin City University

Solutions:

There are two types of solutions for issues in the data-gathering phase.

1) **Preemptive**: build in **integrity checks** and entry constraints.

This is where you try to imagine what could go wrong and implement either constraints in your system or put process management actions in place to enforce and/or reward accurate data entry.

For example, having critical data entry checked by other parties or using data type constraints in your entry form.

> A key part of preemptive strategies is to avoid data loss during transmission

2) **Retrospective**: focus on **cleaning through**,

for example, duplicate removal, merge/purge, name & address matching or field value standardization.

This approach has a diagnostic focus to try and **automate the detection and repair of artefacts, errors** and glitches

Data errors in the Preserving Phase

- Format conversion errors that can happen silently when importing or storing data such as string or float or date (1918 or 2018?). Format conversion errors may also include rounding or approximation such as a database field limited to integers.
- No metadata or schema information recorded, e.g., What does the field mean?
- Poor process and methodologies: for example, there is no filename convention or version control

Solutions

1. Document and publish – remember, good metadata!
2. Data exploration and retrospective checking. Does anything in the data look funny, odd or unexpected?
3. Assume that the worst and have contingency plans.

Data errors in the Processing Phase

- processing phase includes integrating data from multiple sources

Example errors or artefacts

- **Heterogeneous data**: no common key, different field formats, approximate or exact string matching.
- **Different definitions**: What is a customer, an account, a sale, a book, an address etc.
- **Time synchronization**: Does the data relate to the same time periods? Are the time windows compatible? Do you understand the time format (American month-day-year?).
- **Legacy data in multiple formats**: database systems, spreadsheets, ad-hoc files, binary data, and many other formats.
- **Sociological factors**: knowledge is power so the person you are requesting the data from may be reluctant to share or help.

Solutions:

1. **Commercial Tools (e.g., from IBM, Oracle, SAP, etc.)**: These are the result of a significant body of research in data integration. Many **tools are available** that perform convenient and powerful tasks such as address matching or schema mapping.
2. **Data browsing and exploration**: You can use many other tools to browse the metadata – either provided or extracted by you. By viewing the **before and after results** (number of entries, expected data types, etc.) you can check if the integration happened as expected.

Data errors in the Analysing Phase

What are you planning to do with all this data anyway?

This will help you to understand the level of precision required, the types of features, the importance of time-series granularity (data from every second or every minute) and what the needs are of the analytics model you are hoping to create.

examples:

- Scale and performance: how much data are you processing? Will you run into memory or processing issues?
- Insufficient domain expertise: the analyst may not know a specific fact about the real world that explains a trend or outlier.

Solutions

Engage the smell test! This is the most important skill you can develop; a finely-honed sense of potential problems and an awareness of what your data is telling you.

Generally, you want to build the following tasks into your data processing and analytics workflow.

1. **Data exploration**: determine which models and techniques are appropriate, find data bugs and develop domain expertise.
2. **Continuous analysis**: Are the results stable? How do they change?
3. **Accountability**: Make the analysis part of the feedback loop.

wk2 quiz: https://www.futurelearn.com/courses/data-management-and-visualisation-data-cleaning-and-data-quality/3/steps/1600152/quiz/introduction

---

Week 3: Practical Data Cleaning

Practical Data Cleaning

Methods for data cleaning can be categorised in three ways:

1. Implement process mandates (fix human prob)
   - schema/rule to emforce format
2. Custom tools ('hack a script')
   - cleaning that only happens once off
3. Off-the-shelf tools such as Spreadsheets,OpenRefine

Spreadsheets:

- pro: widley avail
- con: memory limits

OpenRefine:

- pro: easier for non-programmers
- con: it is standalone and less widely available AND difficult to perform editing actions like splitting rows or inserting new rows

https://libjohn.github.io/openrefine/preamble.html

Python:

- pro: powerful and flexible, the cleaning process with Python can be deployed to other parts of the workflow
- pro: regular expressions using Python are extremely powerful

- con: not so useful for non-programmers AND regular expressions require skill and practise to use well

cLEANING tIPS:

- MAKE A COPY BEFORE CHANGES
- be transparent about cleaning
- consult domain experts

spreadhseet cleaning https://www.futurelearn.com/courses/data-management-and-visualisation-data-cleaning-and-data-quality/3/steps/1600160

ReFine: https://www.futurelearn.com/courses/data-management-and-visualisation-data-cleaning-and-data-quality/3/steps/1600161

- merge duplicate cols, edit col
- cluster groups of groups merge

- fix inconsitencies
- tracks changes
- log scale distributions

turtorial: https://youtu.be/B70J_H_zAWM

Python with Jupyter:

Colab on cleaning data :
https://github.com/suzannelittle/ca682i/blob/master/notebooks/2_3_7_cleaning.ipynb

- Open and check CSV data.
- Convert a time/date string into the special datetime64 data type and use this to index the dataset.
- Detect and remove duplicate entries.
- Update entries.
- Detect missing values.
- Check place names against an original reference list.
- Detect and repair leading and trailing whitespace.
- Look for errors by examining the set of unique values.

View it here as well:
http://localhost:8890/notebooks/Documents/GitHub/ca682i/notebooks/2_3_7_cleaning.ipynb

Addressing data quality

EU Case study: https://ugc.futurelearn.com/uploads/files/1c/21/1c219bac-6d17-424b-b589-d5f420643f95/Extract_from_The_Data_Journalism_Handbook_2012.pdf

wk3 quiz: https://www.futurelearn.com/courses/data-management-and-visualisation-data-cleaning-and-data-quality/3/steps/1600155/quiz/introduction

---

Week 4: Open Data & Gathering

Open Data

Open data is data that anyone can access, use and share.

Read this case study from the Open Data Institute on how open data has been used to promote physical activity in the UK: https://theodi.org/article/openactive-addressing-inactivity-in-the-uk-case-study/

Using APIs and REST

REST APIs can be very useful ways to get the latest data from web sites or social media. The advantage of using APIs like this is that the call for data can be dynamic and embedded into a programme or data pipeline using common languages such as python, javascript or R.

You can use programming libraries like `curl` or `requests` in Python to make REST calls by creating the appropriate URL string

Colab on this: "C:\Users\Cian\Documents\GitHub\ca682i\notebooks\2_4_6_Task_APIs.ipynb"

or

https://github.com/suzannelittle/ca682i/blob/master/notebooks/2_4_6_Task_APIs.ipynb

sopace station location API

`print(response.status_code)`

get location

`iss_location = response.content`
`print(iss_location)`

convert JSON to python obj

`iss_object = json.loads(iss_location)`

which gets us:

```
{'iss_position': {'latitude': '30.6869', 'longitude': '-160.8633'},
 'timestamp': 1666620348,
 'message': 'success'}
```

which we can use as:

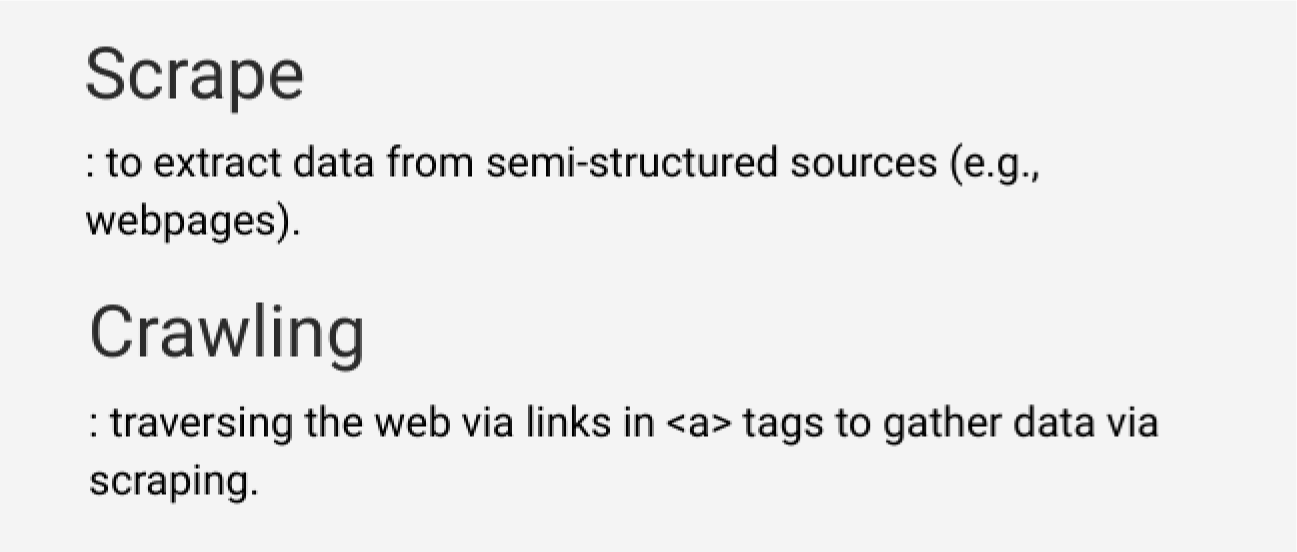
```
print("The location of the ISS at %s was %s latitude and %s longitude." %
(time.ctime(iss_object['timestamp']),iss_object['iss_position']['latitude'],
iss_object['iss_position']['longitude']))
```

Let's put all of that together in one cell so you can see that the data updates each time you call the API.

```
response = requests.get(iss)
if response.status_code != 200:
  print("Error: %d" %(response.status_code))
else:
  iss_location = response.content
  iss_object = json.loads(iss_location)
  print("The location of the ISS at %s was %s latitude and %s longitude."
(time.ctime(iss_object['timestamp']),iss_object['iss_position']['latitude'],
iss_object['iss_position']['longitude']))
```

Can you repeat the steps above to get 10 random trivia questions from the Open Trivia Database? The URL to use is https://opentdb.com/api.php?amount=10

Scraping data from websites:



Scrape

: to extract data from semi-structured sources (e.g., webpages).

Crawling

: traversing the web via links in <a> tags to gather data via scraping.

You will almost certainly need to clean the data as scraping can be very prone to introducing errors and artefacts

Python provides some handy libraries to help with scraping including:

- `requests` (which you've seen for accessing REST APIs) - downloading the page
- `BeautifulSoup` - parsing the HTML into an object to search and manipulate

These libraries are fine for once off tasks or exploring scraping but for more stable, longer term projects check out Scrapy. There are other tools for web scraping given in this ranked list.

- **Scapy**: https://doc.scrapy.org/en/latest/intro/tutorial.html
- **ranked list**: http://www.aioptify.com/top-web-scraping-frameworks-and-librares.php

check out the colab 2_4_8_Web_Scraping

http://localhost:8890/notebooks/Documents/GitHub/ca682i/notebooks/2_4_8_Web_Scraping.ipynb

Beautiful Soup: Build a Web Scraper With Python:

https://realpython.com/beautiful-soup-web-scraper-python/

A challenge for you: Who can be the first to post code showing how to get a list of all the All Ireland Senior Football Champions (just the county name) from Wikipedia? https://www.futurelearn.com/courses/data-management-and-visualisation-data-cleaning-and-data-quality/3/steps/1600171

---

wk4 quiz: https://www.futurelearn.com/courses/data-management-and-visualisation-data-cleaning-and-data-quality/3/steps/1600172/quiz/introduction

In [ ]:

```
```