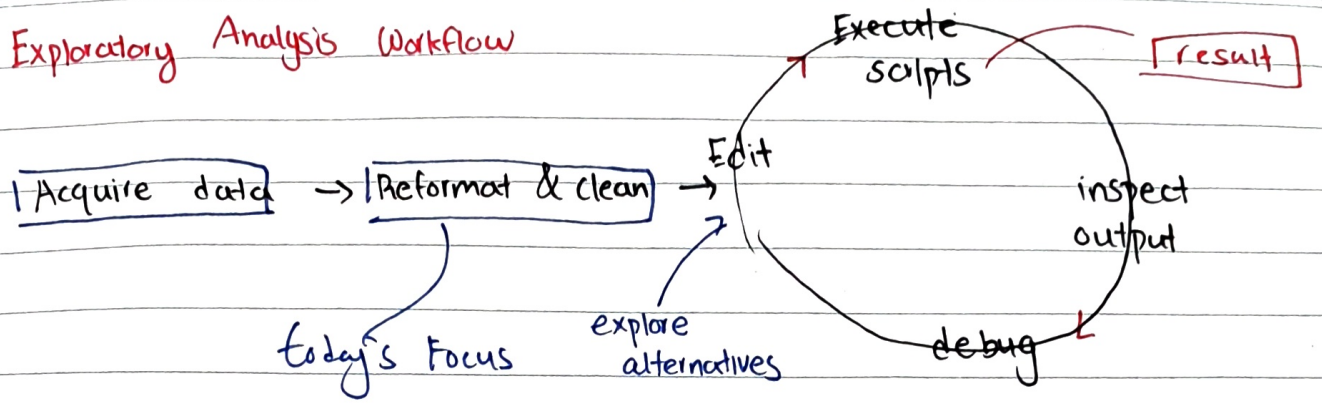
Wk 2 - Lecture

Exploratory Analysis Workflow



Level of Measurements & Types of data

- Categorical (Qualitative)
  - nominal vs ordinal (ordered; distance cannot be implied)
  - Nominal ex) gender, state
  - ordinal ex) [low, medium, high]
- Quantitative
  - interval : values encode differences; equal interval between variables
  - ratio : zero is defined
  - ✱ No True zero eg. celsius
  - eg. kelvin, experience, weight, pressure

Other

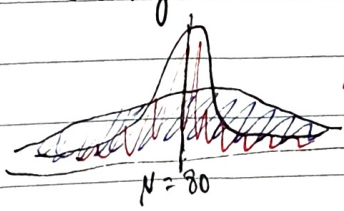
- images, text, videos, etc

	Nominal	Ordinal	Interval	Ratio
Countable, equally defined	✓	○	○	○
Order defined	✗	○	○	○
Difference defined (+, −)	✗	✗	○	○
Zero defined (×, ÷)	✗	✗	✗	○

Nominal	mode	median	mean	count / distribution	min/max, range, percentile	variance, sd
Ordinal					
Interval					
Ration					

Q variance / SD ?

- Looking at mean (only) can't tell the performance of a particular student



SD = 10 (most are on the same line, few extremes)

SD = 30 (more dispersed)

Acquiring Data & Cleaning

Acquire from

- File Access (csv, json, excel, XML)
- Programmatically (web scraping, APIs of web services like twitter)
- Database Access
- Collect data (survey)

Clean by (real data often dirty)

- type and name conversion
- filtering of missing/inconsistent data
- unifying semantic data representation
- matching of entities from different sources

) data collected from different source may have different name, formats etc

- handling of date & time
- scaling & normalization, centralise around mean (mean=0, sd=1) (advanced)

optional dimensionality reduction for visualisation often (2D or 3D) choose specific vars. (can't display all vars!)

Cleaning with Python

- look at the data first !! before any operations.
- keep the original data !! before modifications.

csv module

- csv.reader : rows → arrays (column num = idx)
- csv.DictReader : rows → dict (colum name : rows)

pprint (pretty print)

alternative: pandas

↳ open source

↳ data import and analysis functionality

↳ tabular (DF) 2D

↳ time-series (series) 1D

↳ matrix 1000 3000 2D, 3D ... ND

functions

display first/last n rows : head, tail

missing values ? : dropna() fillna(0), replace(Null) → 0

Cleaning data

- both csv / pandas will likely read-in data as strings.
- we need type conversion or else error (mean of strings?)

ex) `data[column] = data[colum].astype(int)`

fails if any entry ≠ raise Exception
(NaN ↛ !nt)

or

define our own function
to loop thru row & ignore NaN (numpy.nan)
(unrecognised values → np.NAN (Default))

Descriptive statistics with Pandas

count, sum, abs
min, max
mean, median, mode, std

`data = pd.read_csv('_____.csv')`

ex) `data['power'].min()`
`.std()`

prod(): product of values, cumsum(): cumulative sum, cumprod(): cumulative product

groupby(), apply(), applymap()

Filtering

`data.loc( data[col] == '_____', col )`

`data.loc ( data['type'] == 'wind-turbine' (&) data['power'] > 100 ]`

select

= wind generator with power > 100 mw

`data.groupby('class').size()`

group-by

column

Numpy: calculate central tendency & dispersion

`np.nanmax(data) - np.nanmin(data)`

v = row

nan values ignored automatically

`iqr = np.nanpercentile(v, 75) - np.nanpercentile(v, 25)`

Visualising with python

(matplotlib)

```
distribution = data.groupby('fueltype').count().reset_index(name = 'numstations')
```

create frequency distribution

↳ ≈ select * AS ____ (in sql)

```
plt.bar(distribution['fueltype'], distribution['numstations'], alpha = 0.5, align = 'center')
```

- title( )
- xlabel( )
- ylabel( )
- grid( )
- xticks(rotation = 40)

```
data['power'].hist(bins = 10, rwidth = 0.9)
```

```
sub = plt.subplot()
data.plot.scatter(x = 'numGen', y = 'power', c = 'Darkblue')
sub.set_xlim(0,50)
```

look for correlation with scatterplot

Correlation statistics to measure dependence

- parametric
  - pearson's r value
- non-parametric
  - spearman's rho
  - kendall's tau

} scipy