FOAM
Lecture 4
Introduction to Machine Learning and Data Handling

Mike O'Dea
(based upon material created by Simos Gerasimou)

Check In code: 242141

This Lecture

- Data Science Lifecycle
- Exploratory Data Analysis
- Handling Missing Data
- Identifying and Handling Outliers

Unlocking Value From Data

[Data science applications and examples]

Unlocking Value From Data

Example: **Predict Neonatal Infection**

Problem: Children born prematurely are at high risk of developing Infections, many of which are not detected until after the baby is sick

Goal: Detect subtle patterns in the data that predicts infection before it occurs

Data: 16 vital signs (e.g., heart rate, respiration rate, blood pressure)

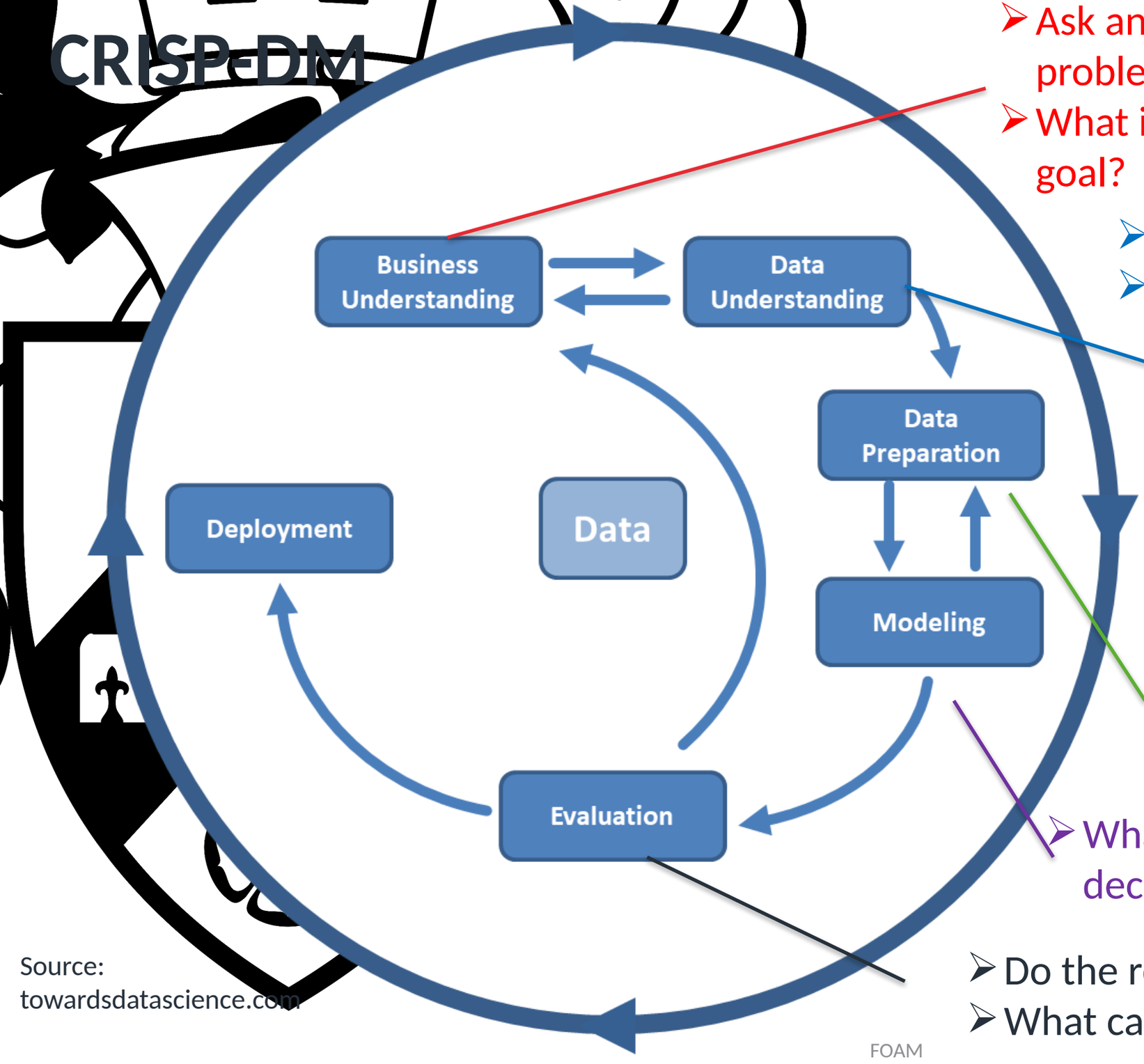
Task: Develop a predictive model using existing data and use the data from new children to predict the infection

Impact: Predict the start of infection 24 hours before traditional symptoms of infection appear

[Predicting neonatal infections by evaluation of the gastric aspirate: a study in two hundred and seven patients]

Lifecycle of a Data Science Project

CRISP-DM



Source: towardsdatascience.com

- Ask an interesting question/find a problem to tackle
- What is the business objective/scientific goal?
  - How to collect data?
  - Which data is relevant/important for my problem?
    - What each data item (e.g., column) represents?
    - Are there missing values or corrupted records?
  - Are there any anomalies and patterns?
  - How should I plot the data?
- What models should be built to enable decision-making or support?
- Do the results make sense?
- What can we learn? Do we have a story?

Data Pipeline

UNIVERSITY of York

#GCPSketchnote

@PVERGADIA
THECLOUDGIRL.DEV
8.13.2020

How to build a scalable DATA ANALYTICS PIPELINE

CAPTURE — Data ingestion at any scale

PROCESS — Reliable streaming data pipeline

STORE — Data lake and data warehousing

ANALYZE — Data warehousing

USE — Advanced analytics

CLOUD PUB/SUB
Scaled messaging platform

DATA TRANSFER SERVICE
Fast data migration from saas apps

STORAGE TRANSFER SERVICE
Data migration from other cloud or on-prem

CLOUD IoT CORE
Stream events from IoT devices

CLOUD DATAFLOW
Stream and batch processing

CLOUD DATAPROC
Managed Hadoop & Spark platform

CLOUD DATAPREP
Data prep using visual tool

CLOUD STORAGE
Use as your data lake for structured and unstructured data

BIGQUERY STORAGE
Cloud-native, highly scalable serverless data warehouse

BIGQUERY
Analysis engine

CLOUD AI PLATFORM
For machine learning

TENSORFLOW
For machine learning

LOOKER
For your analysis

SHEETS
For your analysis

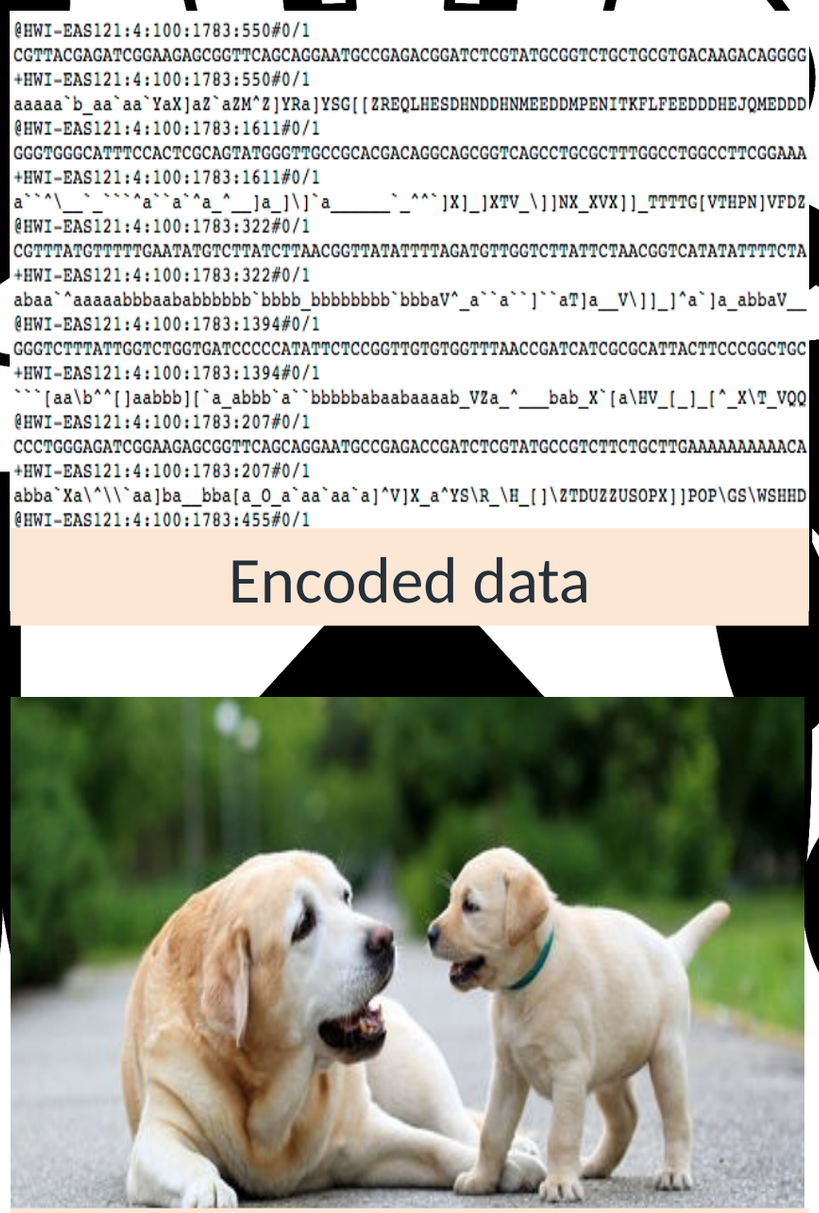
CLOUD DATA FUSION
Fully managed, code-free data integration service to manage ETL/ELT pipelines and also track lineage of that data.

DATA CATALOG
Unified view of all datasets

CLOUD COMPOSER
Fully managed, workflow orchestration built on Apache Airflow

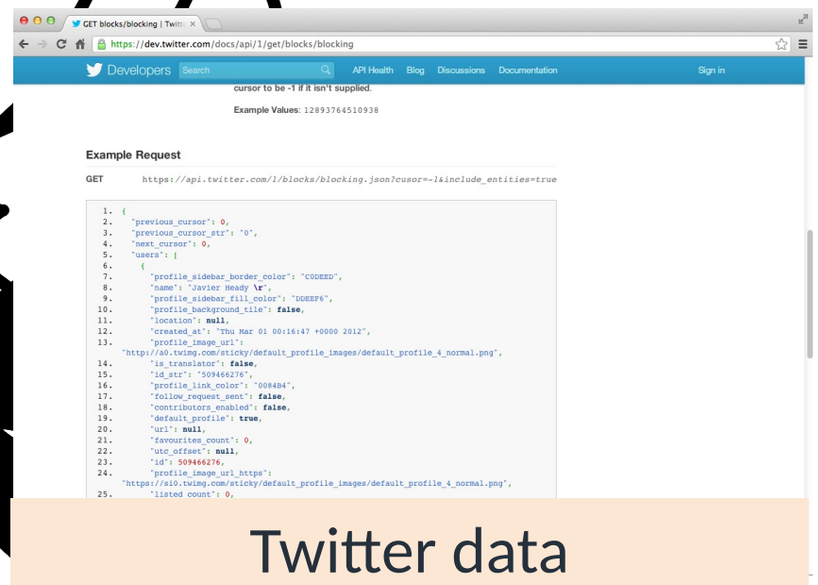
Source: peerdh.com

What do data look like?



```
@HWI-EAS121:4:100:1783:550#0/1
CGTTACGAGATCGGAAGAGCGGTTCAGCAGGAATGCCGAGACGGATCTCGTATGCGGTCTGCTGCGTGACAAGACAGGGG
+HWI-EAS121:4:100:1783:550#0/1
aaaaa`b_aa`aa`YaX]aZ`aZM^Z]YRa]YSG[[ZREQLHESDHNDDHNMEEDDMPENITKFLFEEDDDHEJQMEDDD
@HWI-EAS121:4:100:1783:1611#0/1
GGGTGGGCATTTCCACTCGCAGTATGGGTTGCCGCACGACAGGCAGCGGTCAGCCTGCGCTTTGGCCTGGCCTTCGGAAA
+HWI-EAS121:4:100:1783:1611#0/1
a```^\__`_```^a``a`^a_^__]a_]\]`a_____`_^^`]X]_]XTV_\]]NX_XVX]]_TTTTG[VTHPN]VFDZ
@HWI-EAS121:4:100:1783:322#0/1
CGTTTATGTTTTTGAATATGTCTTATCTTAACGGTTATATTTTAGATGTTGGTCTTATTCTAACGGTCATATATTTTCTA
+HWI-EAS121:4:100:1783:322#0/1
abaa`^aaaaabbbaabaabbbbbb`bbbb_bbbbbbbb`bbbaV^_a``a``]``aT]a__V\]]_]^a`]a_abbaV__
@HWI-EAS121:4:100:1783:1394#0/1
GGGTCTTTATTGGTCTGGTGATCCCCCATATTCTCCGGTTGTGTGGTTTAACCGATCATCGCGCATTACTTCCCGGCTGC
+HWI-EAS121:4:100:1783:1394#0/1
```[aa\b^^[]aabbb][`a_abbb`a``bbbbbabaabaaaab_VZa_^___bab_X`[a\HV_[_][^_X\T_VQQ
@HWI-EAS121:4:100:1783:207#0/1
CCCTGGGAGATCGGAAGAGCGGTTCAGCAGGAATGCCGAGACCGATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAACA
+HWI-EAS121:4:100:1783:207#0/1
abba`Xa\^\\`aa]ba__bba[a_O_a`aa`aa`a]^V]X_a^YS\R_\H_[]\ZTDUZZUSOPX]]POP\GS\WSHHD
@HWI-EAS121:4:100:1783:455#0/1
```

Encoded data

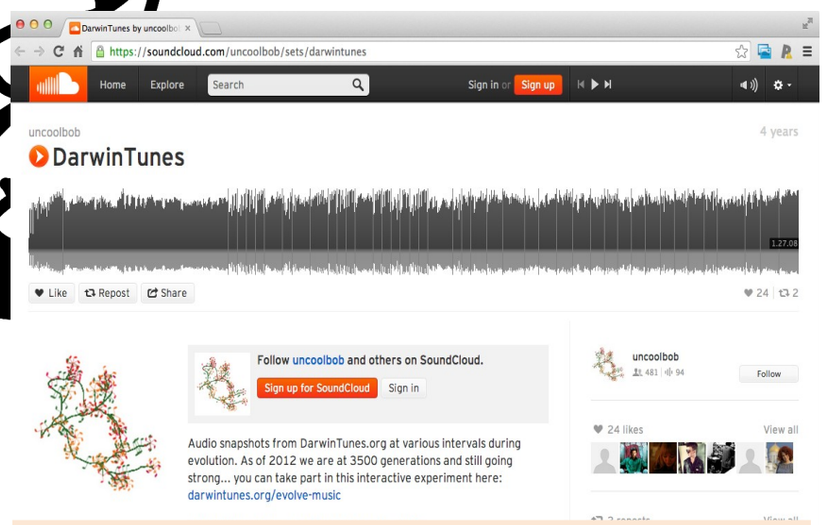



Twitter data

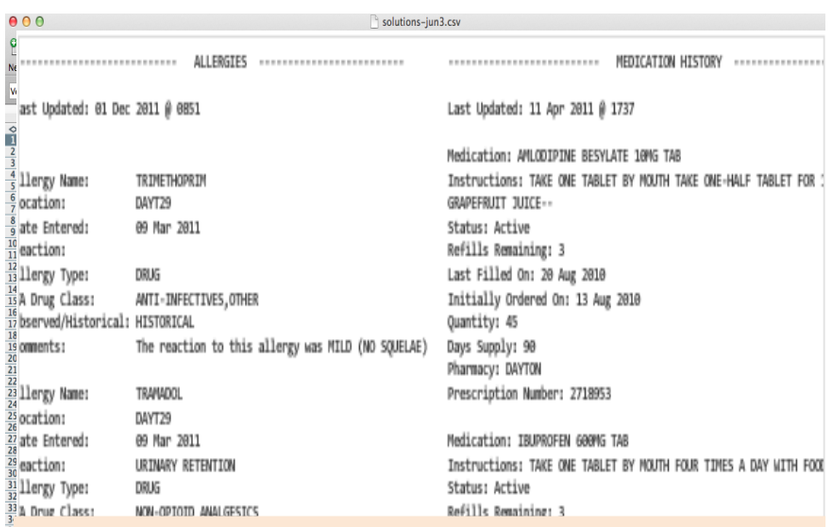

Patient records



Images

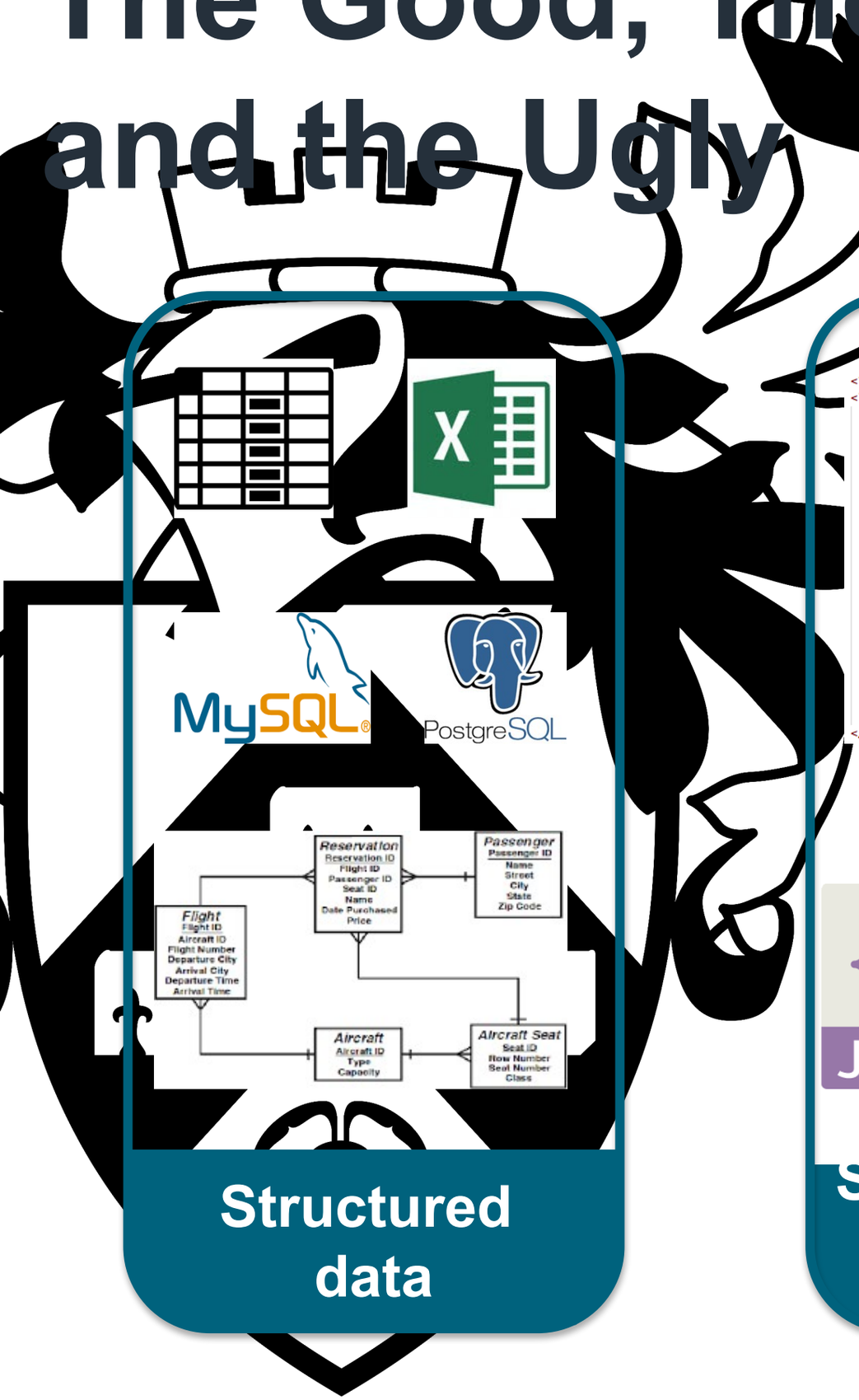



Sound




Excel data
```

The Good, The Not-So-Bad and the Ugly



Structured data

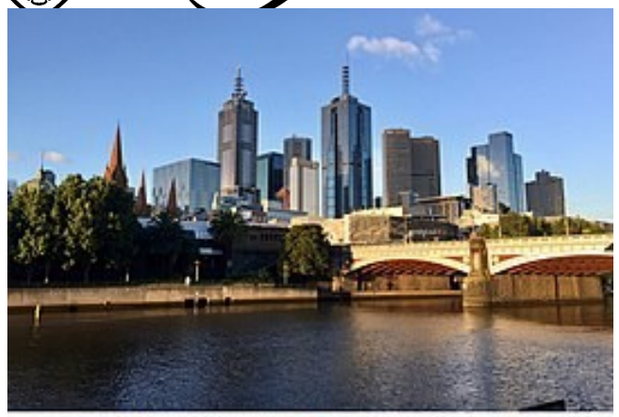
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<lectures>
  <lecture>
    <name>Lecture 1</name>
    <topic>Data Science Introduction</topic>
    <status>Done</status>
  </lecture>
  <lecture>
    <name>Lecture 2</name>
    <topic>Descriptive Statistics</topic>
    <status>Done</status>
  </lecture>
  <lecture>
    <name>Lecture 3</name>
    <topic>Data Science Process</
    <status>Done</status>
  </lecture>
  <lecture>
    <name>Lecture 4</name>
    <topic>Data Processing</topic
  </lecture>
</lectures>
```

```
{
  "DAT1 Lectures":[
    {
      "name":"Lecture 1",
      "topic":"Data Science Introduction",
      "status": "Done"
    },
    {
      "name":"Lecture 2",
      "topic":"Descriptive Statistics",
      "status": "Done"
    },
    {
      "name":"Lecture 3",
      "topic":"Data Science Process",
      "status": "Done"
    },
    {
      "name":"Lecture 4",
      "topic":"Data Processing"
    }
  ]
}
```

Semi-structured data

Unstructured data

Melbourne Housing Market

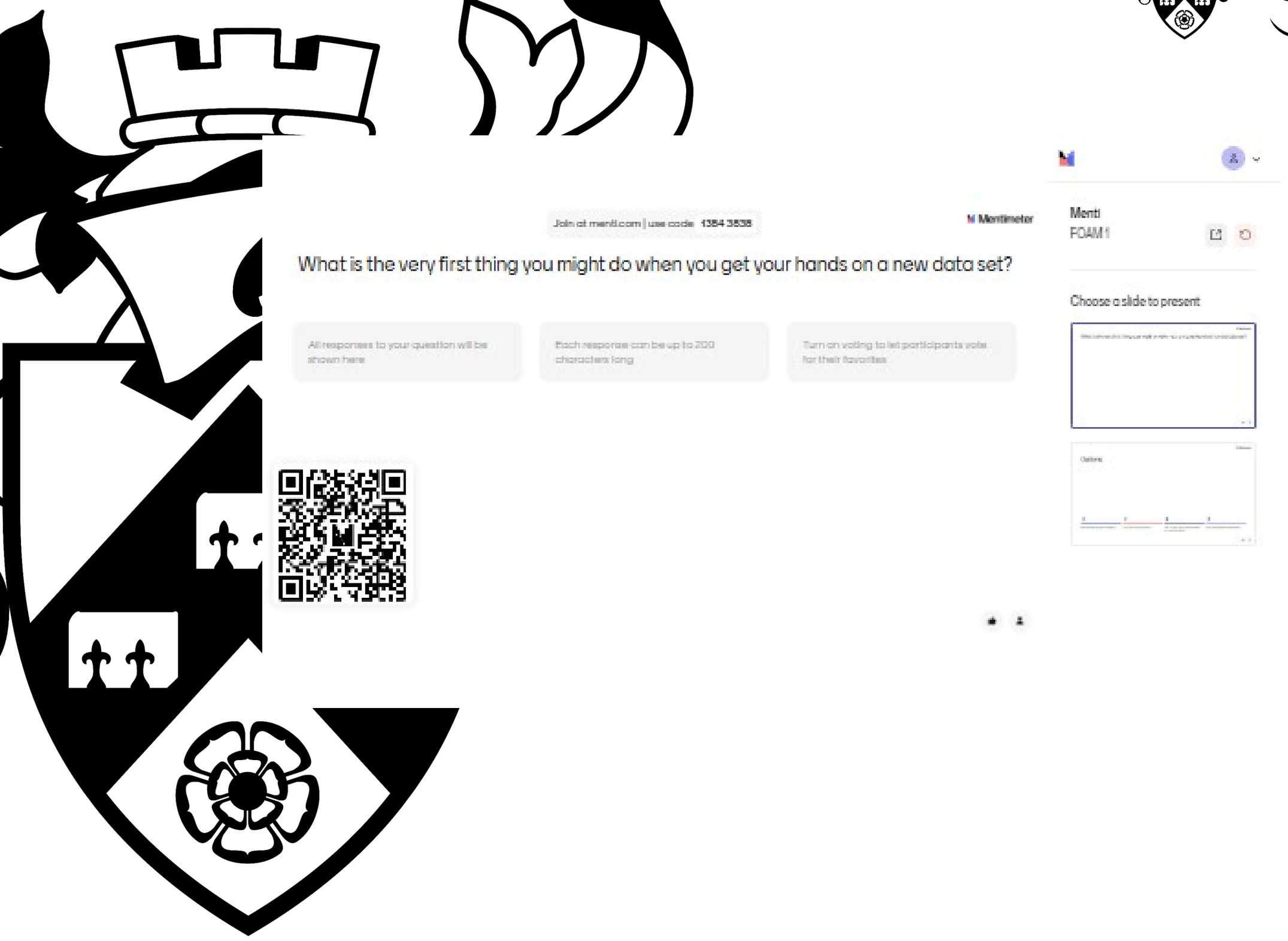


https://en.wikipedia.org/wiki/Melbourne

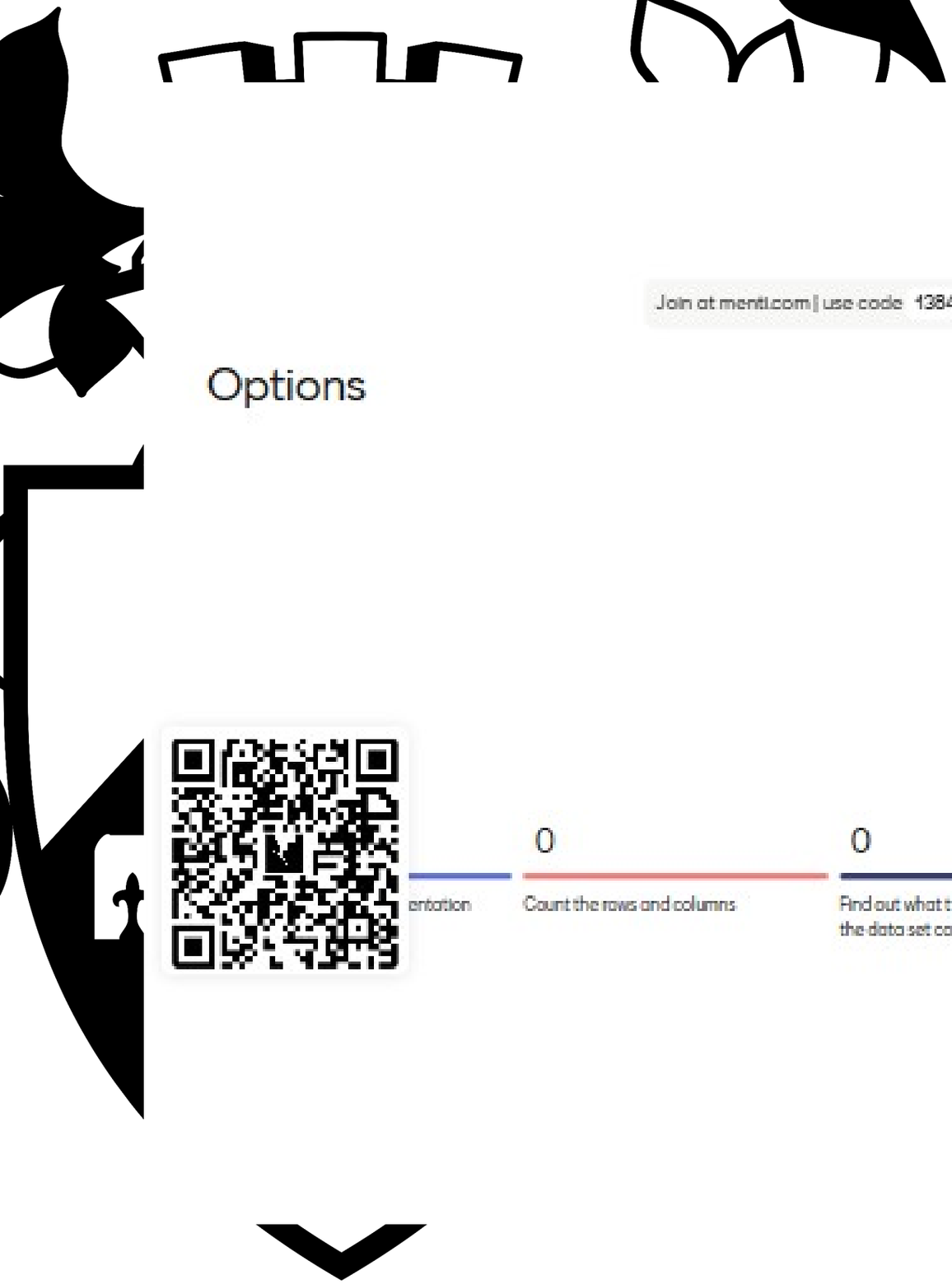
- **Context:** John and Jane Doe from York (UK) have been offered a great job as Senior Data Scientists at a big financial organisation in Melbourne (AU).
- **Problem:** Before committing to such a life changing decision, the couple wants to examine the cost of living in Melbourne, given the continually increasing house prices.
- Real data extracted by scraping the AU property site Domain.com.au
- Detailed information for more than **34K entries - 21 variables**: address, #rooms, price etc

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Postcode	Regionname	Propertycount	Distance	CouncilArea
0	Abbotsford	49 Lithgow St	3	h	1490000.0	S	Jellis	1/04/2017	3067	Northern Metropolitan	4019	3.0	Yarra City Council
1	Abbotsford	59A Turner St	3	h	1220000.0	S	Marshall	1/04/2017	3067	Northern Metropolitan	4019	3.0	Yarra City Council
2	Abbotsford	119B Yarra St	3	h	1420000.0	S	Nelson	1/04/2017	3067	Northern Metropolitan	4019	3.0	Yarra City Council
3	Aberfeldie	68 Vida St	3	h	1515000.0	S	Barry	1/04/2017	3040	Western Metropolitan	1543	7.5	Moonee Valley City Council
4	Airport West	92 Clydesdale Rd	2	h	670000.0	S	Nelson	1/04/2017	3042	Western Metropolitan	3464	10.4	Moonee Valley City Council

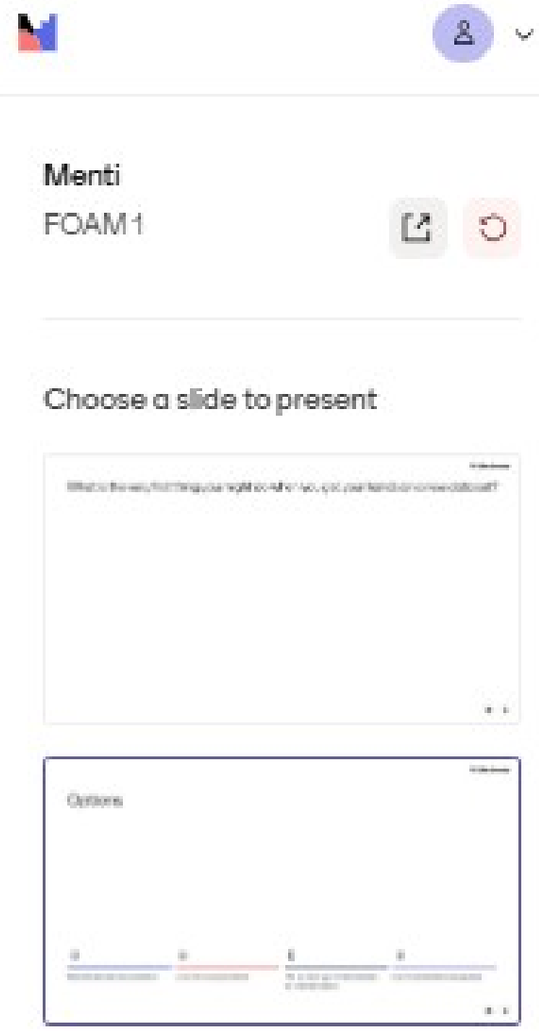
We have data. Now what?



We have data. Now what?



UNIVERSITY of York



Exploratory Data Analysis

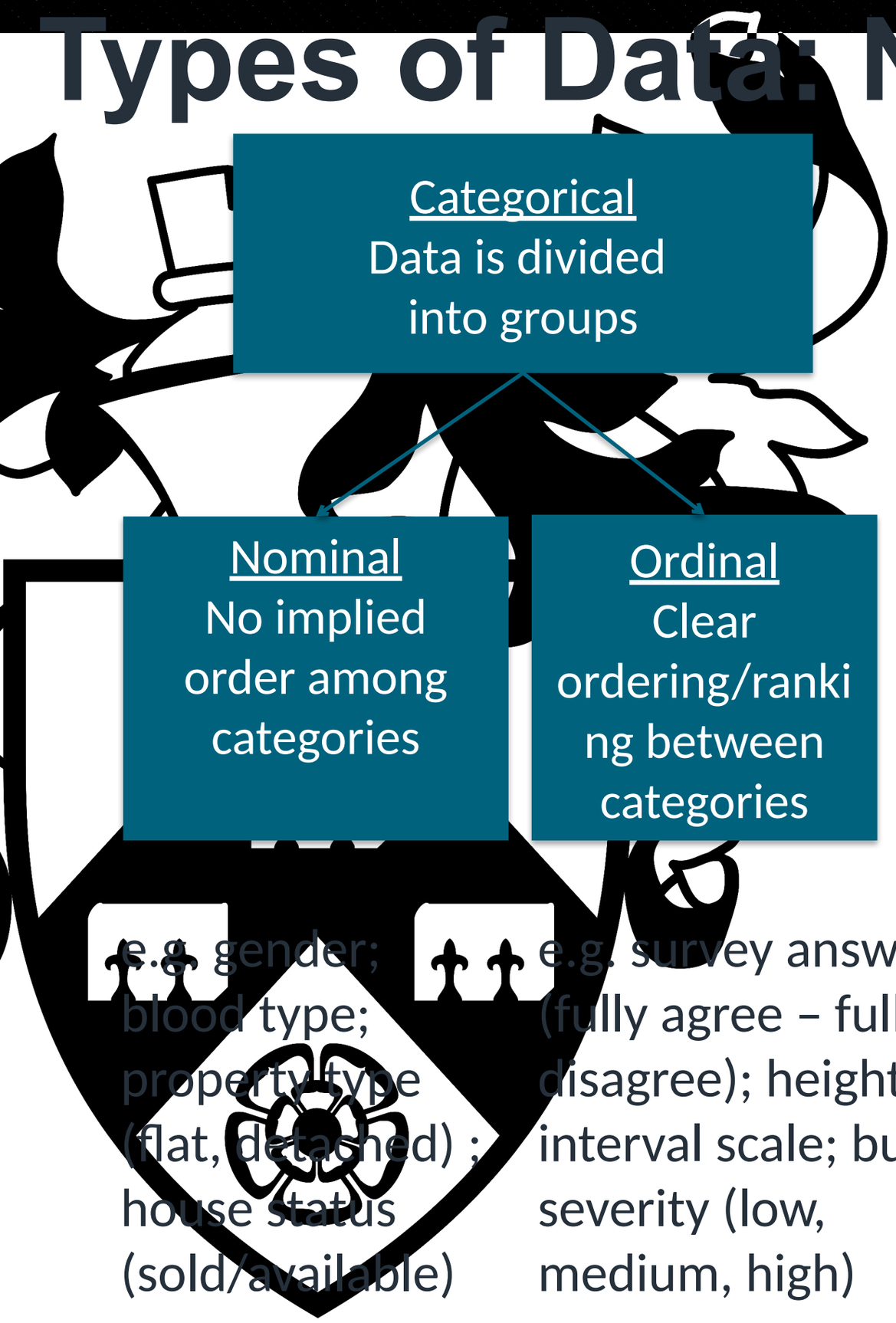
- Before making inferences from data…
  - it is essential to **examine the variables of our dataset**

Why?

- To see **patterns** in the data
- To **identify and resolve mistakes**
- To find **violations of statistical assumptions**
- To find **how many** observations/entries exist
- To find the **data types** of my dependent/independent variables
- To assess whether **the data is relevant to the problem** solved

- …and because if you don’t, you will have trouble later!

Types of Data: NOIR



Categorical
Data is divided into groups

Numerical
Data represents numbers

Nominal
No implied order among categories

e.g. gender; blood type; propertytype (flat, detached) ; house status (sold/available)

Ordinal
Clear ordering/ranking between categories

e.g. survey answers (fully agree – fully disagree); height interval scale; bug severity (low, medium, high)

Interval
Can be counted in a finite manner

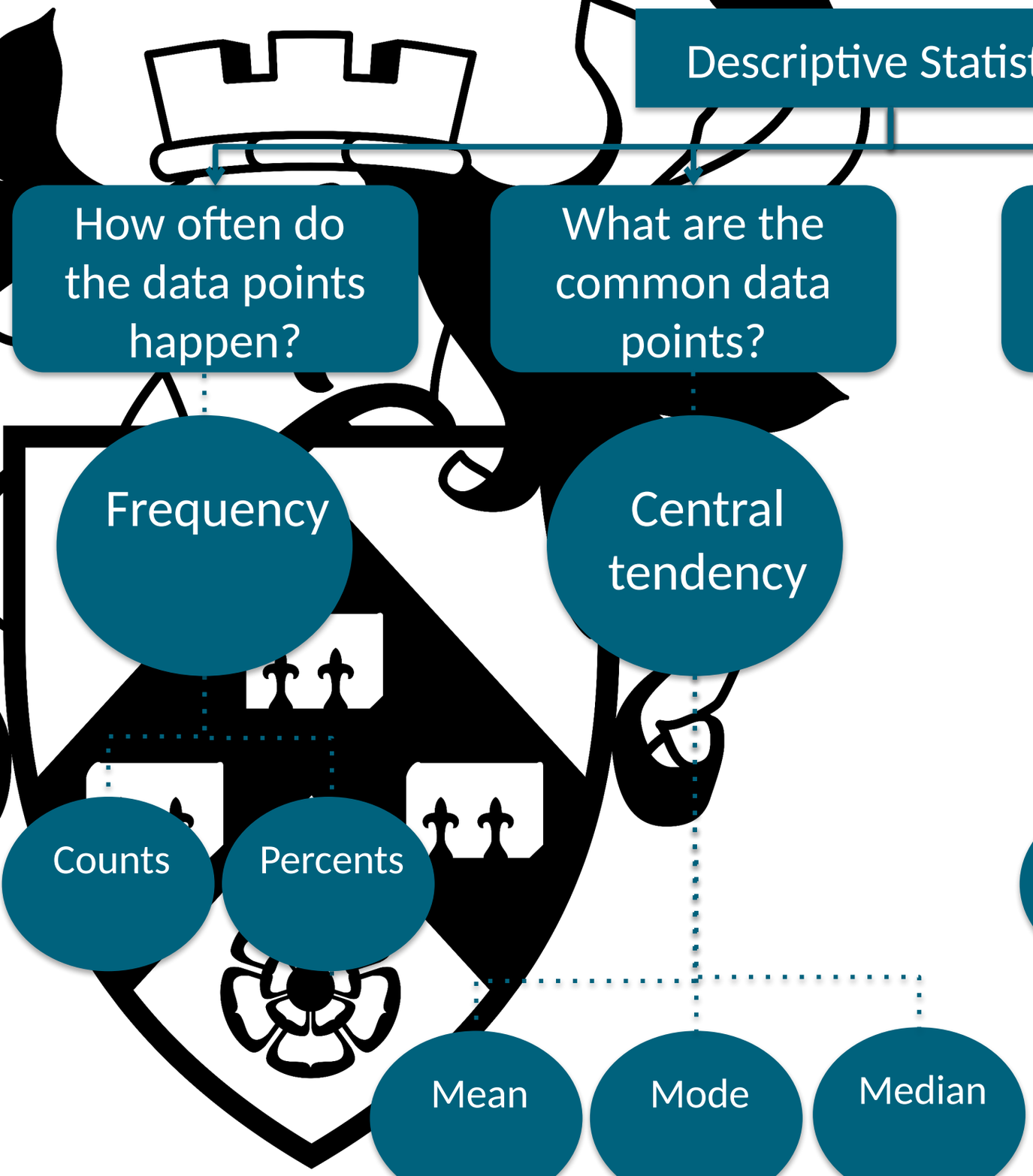
e.g. number of heads in 100 coin flips; temperature in Celsuis (Centigrade) or Fahrenheit

Ratio
Can be measured not counted; infinite number of possibilities

e.g., dog height; house price; house distance from York centre; exact amount of gas purchased from petrol station

Descriptive Statistics

UNIVERSITY of York



- Descriptive Statistics
  - How often do the data points happen?
    - Frequency
      - Counts
      - Percents
  - What are the common data points?
    - Central tendency
      - Mean
      - Mode
      - Median
  - Where are the data points (wrt to each other)?
    - Position
      - Quartile rank
      - Percentile rank
  - How does the spread of scores look like?
    - Variation/ Dispersion
      - Variance
      - Standard deviation
      - Range

Frequency

- Used with countable data, i.e., there are separate categories in which the subject can be and the number of entries in each category can be counted
- **Count**: how many times an event has happened or happened within a given time frame
- **Percent**: The percentage of a particular category over the sample size

```
#Count of houses of each type
uniqueT, countsT = np.unique(data['Type'], return_counts=True)
print("Number of houses of different types:")
print(np.asarray((uniqueT, countsT)))
```

```
Number of houses of different types:
[['h' 't' 'u']
 ['34161' '4980' '9292']]
```

```
#Percentage of house of each type
print("Number of houses of different types:")
print(np.asarray((uniqueT, countsT/len(data['Type'])*100)))
```

```
Number of houses of different types:
[['h' 't' 'u']
 ['70.532488179547' '10.282245576363223' '19.185266244089775']]
```

Mean

- Captures the centre of the distribution
- Help us understand our data
- If the data sample is drawn from a population, the mean of the sample is an unbiased estimate of the population mean

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

```
#Mean House Prices
mean = np.mean(data['Price'])
print("Houses mean price:", mean)

isH = data['Type']=='h'
priceH = data[isH]
meanH = np.mean(priceH['Price'])
print("Houses Type H mean price:", meanH)

isT = data['Type']=='t'
priceT = data[isT]
meanT = np.mean(priceT['Price'])
print("Houses Type T mean price:", meanT)
```

```
Houses mean price: 997898.2
Houses Type H mean price: 1110586.8
Houses Type T mean price: 911148.0
```

- For example, given the house values (in thousands) [70,60,80,85,92]
  - μ = (70+60+80+85+92)/5 = 77.4

Median

- The **middle score in an ordered set of data**

$$\text{Median} = (N+1)/2$$

- **Odd number of observations**
  - ➔ find the middle value
  - e.g., [30,45,67,87,94,102,124]
  - Median = 87
- **Even number of observations**
  - ➔ find the middle two values and average them
  - e.g., [30,45,67,87,94,102,124,155]
  - Median = (87+94)/2=90.5

```
#Median House Prices
mean = np.median(data['Price'])
print("Houses mean price:", mean)

isH = data['Type']=='h'
priceH = data[isH]
meanH = np.median(priceH['Price'])
print("Houses Type H mean price:", meanH)

isT = data['Type']=='t'
priceT = data[isT]
meanT = np.median(priceT['Price'])
print("Houses Type T mean price:", meanT)
```

```
Houses mean price: 830000.0
Houses Type H mean price: 935000.0
Houses Type T mean price: 830000.0
```

- The median minimises the sum of the absolute deviations

Mode

- The **most commonly occurring entry** (with the highest frequency)
- Represents the largest number of data with the same value in the sample
- If a value appears repeatedly in the data, it will influence the average towards the modal value
- Remember the frequency of type h?
  - 70.38% (more than half of the available houses)

```
#Mode - most commonly occurring house type
uniqueT, countsT = np.unique(data['Type'], return_counts=True)
modeIndex = np.argmax(countsT)
modeType  = uniqueT[modeIndex]
modeCount = countsT[modeIndex]
print("Most common type is ", modeType,
      " with ", modeCount, " houses")
```

```
Most common type is  h  with  34161  houses
```

Range

- Distance from the lowest to the highest value of the data
- Suffers from total reliance to extreme values or outliers (unusually extreme values).
- The price of 3 bedroom type H houses in Melbourne ranges from $235K to $9M!

```
#Range
max = np.max(data['Price'])
min = np.min(data['Price'])
range = max - min
print("House prices range:[",min,',',max,']')

isHwith3R = (data['Type']=='h')&(data['Rooms']==3)
housesHwith3R = data[isHwith3R]
priceHwith3R  = housesHwith3R['Price']
maxH3R        = np.max(priceHwith3R)
minH3R        = np.min(priceHwith3R)
print("Type H 3 Bedroom house prices range:[",
      minH3R,',',maxH3R,']')
```

```
House prices range:[ 85000.0 , 11200000.0 ]
Type H 3 Bedroom house prices range:[ 235000.0 , 9000000.0 ]
```

Variance

- Calculates the deviation of the group from the mean.
- The average sum of squared errors between the mean and the data (the observations)
- Sample variance

$$\frac{\sum_{i=1}^{n}(x_i-\overline{x})^2}{n-1}$$

- Problem?

The result is in units squared (e.g., £^{2}) because we squared the error

e.g. the average error in our Melbourne house data is £352 squared!

No meaning!!

```
#Variance
var = np.var(data['Price'])
print("Houses price variance:", var)
```

```
Houses price variance: 352233700000.0
```

Standard Deviation

- The square root of the variance.
- The measure of average error is in the same units as the original measure
- Sample standard deviation

$$= \frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}$$

```
#Standard deviation
std = np.std(data['Price'])
print("Houses price std:", std)
```

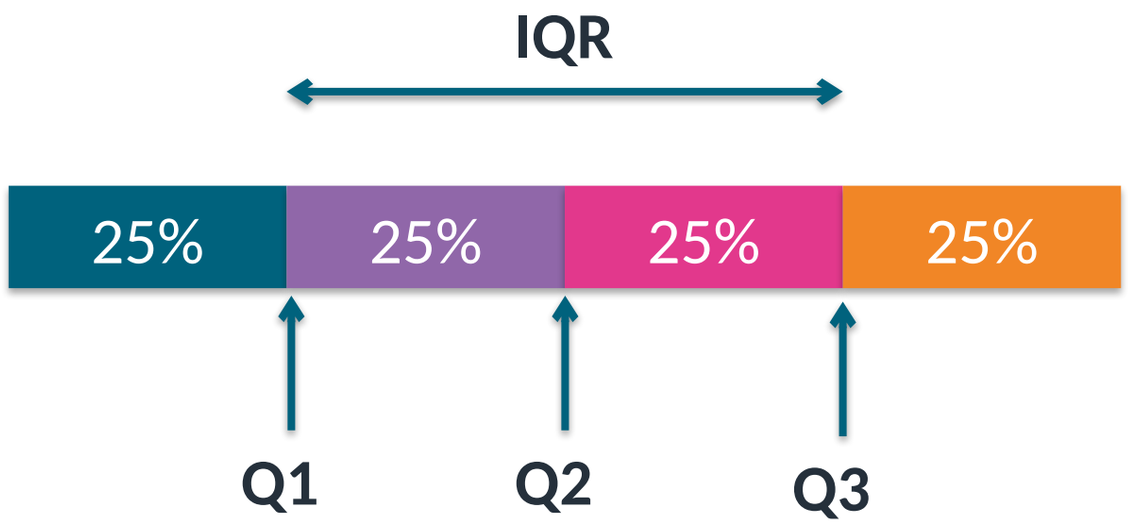
```
Houses price std: 593492.8
```

Quartiles

- Quartiles divide data into four equal regions
  - **Lower quartile Q1** is the median of the first half
  - **Middle quartile Q2** is the median
  - **Upper quartile Q3** is the median of the second half

- Consider the house values (in 100K) [60,70,80,85,92,101,125,150]
  - Q1: 75, Q2:88.5, Q3: 113

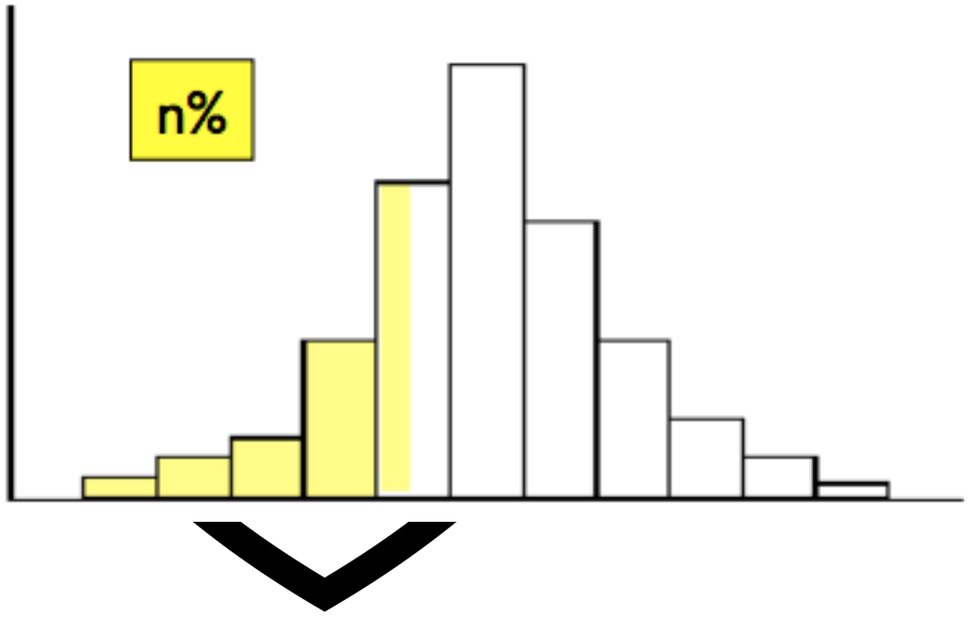
- **Interquartile range: Q3 – Q1**
  - About 50% of data falls within the IQR
  - IQR is not affected by every value in the dataset
  - IQR is not affected by outliers



Percentiles (aka Quantiles)

- The n^{th} percentile is a value such that n% of the data fall at or below or it
  - In which percentile does the house that the Does will buy fall at?

- Q1: 25^{th} percentile = 0.25 quantile
- Median: 50^{th} percentile = 0.50 quantile
- Q3: 75^{th} percentile=0.75 quantile




```
#Percentiles
np.percentile(data['Price'], [25, 50, 75])
```

```
array([ 620000.,  830000., 1220000.])
```

```
#Quantiles
np.quantile(data['Price'], [0.25, 0.50, 0.75])
```

```
array([ 620000.,  830000., 1220000.])
```

Note: Both NumPy methods have an optional interpolation parameter specifying the interpolation method to use {'linear', 'lower', 'higher', 'midpoint', 'nearest'} when the desired quantile lies between two data points.

Do we really need Data Integrity?

A lot of us might have heard about the urban myth that if you are a data analyst/data scientist, data cleaning (or known as data munging as well) forms 80% of the job tasks with the other 20% being made up of machine learning and analysis.

To be honest, I think the ratio is understated, meaning it is rather 90% data cleaning and 10% machine learning and analysis.

Missing Data: Is it serious?

Well, it depends on:

1. The **amount** of missing data
   - Given a variable (column), how much data is missing over the entire dataset?

2. The **pattern(s)** of missing data
   - Does the missing data exhibit any common characteristics?

3. The **reasons** why it is missing
   - Can I identify the reasons that caused the missing data?

Missing data in Melbourne housing market dataset

Variable	Description
Price	Property price (in AUS Dollars)
Distance	Distance from city centre in KMs
Postcode	(self-explanatory)
Bedroom2	Number of bedrooms
Bathroom	Number of bathrooms
Car	Number of car parking spaces
Landsize	Land size in metres
BuldingArea	Building size in metres
YearBuilt	Year the house was built
CouncilArea	Governing Council for the property
Lattitude	(self-explanatory)
Longitude	(self-explanatory)
Regionname	General region of the property
Propertcount	Number of properties in the suburb

UNIVERSITY of York

Missing data in Melbourne housing market dataset

```
df.isna().sum()
```

Suburb	0
Address	0
Rooms	0
Type	0
Price	7610
Method	0
SellerG	0
Date	0
Distance	1
Postcode	1
Bedroom2	8217
Bathroom	8226
Car	8728
Landsize	11810
BuildingArea	21115
YearBuilt	19306
CouncilArea	3
Lattitude	7976
Longtitude	7976
Regionname	3
Propertycount	3

Missing Data: Possible Reasons

- **Solving the problem of missing data is possible only by finding the reasons for the appearance of missing data**

- A variable (question) might **not apply to a specific record**
  e.g. questions only asked to married respondents

- Data source does **not have this information**
  e.g. independence days of countries around the world

- A respondent **fails to answer a question** in a survey
  - The question is too difficult or complicated
  - Ignored by accident
  - They don't want you to know

Missing Data: Types of Missing Data

- **Missing Completely at Random (MCAR)**
  Missingness is purely random.
  The probability of a data point being missing is ***unrelated to both the value of the variable itself and the value of any other variable*** in the dataset.
- **Missing at Random (MAR)**
  You can predict why a value is missing using the data you *do* have.
  Is the most common and often the most manageable type of missing data.
  The probability of a value being missing ***depends on other observed variables*** *in the dataset, but* ***not on the missing value itself.***
- **Missing Not at Random (MNAR)**
  Is the most difficult type. The probability of a value being missing ***depends on the value of the missing variable itself or on an external factor not in the dataset***.

Missing Data: Why Not To Ignore?

- Missing data can lead to **dubious results**
  e.g., find the mean price of properties in Melbourne (with missing values);
  what is 1.48m + NA?

- Many machine learning algorithms **do not work with variables** (features) having **missing values**
  e.g. try running a Deep Neural Network with missing data

- May introduce **bias**
  e.g. linear regression model might result in discriminative behaviour

- More **laborious processing**

Handling Missing Data

- **Information not available for a subject about whom other information exists**
- A Data Science project is only as good as its data
  - The datasets you used so far were cleaned!
- Missing data termed **NaN, Null or NA**
- **General process for handling missing data**
  1. **Understand** data, **identify** reasons for missing data
  2. **Understand distribution**/patterns of missing data
  3. **Decide** on best **recovery method**

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt
0	Abbotsford	68 Studley St	2	h	NaN	SS	Jellis	3/09/2016	2.5	3067.0	2.0	1.0	1.0	126.0	NaN	NaN
1	Abbotsford	85 Turner St	2	h	1480000.0	S	Biggin	3/12/2016	2.5	3067.0	2.0	1.0	1.0	202.0	NaN	NaN
3	Abbotsford	18/659 Victoria St	3	u	NaN	VB	Rounds	4/02/2016	2.5	3067.0	3.0	2.0	1.0	0.0	NaN	NaN
5	Abbotsford	40 Federation La	3	h	850000.0	PI	Biggin	4/03/2017	2.5	3067.0	3.0	2.0	1.0	94.0	NaN	NaN

When Should A Variable Be Excluded?

- **No strict rule about how much missing data for a variable is too much!**
- **Rule of thumb:** Missing data under 20% for an individual variable (observation) can generally be handled by **sophisticated imputation strategies**
- Except when the missing data **occur in a specific non-random manner**
  e.g. Respondents completing the survey on Friday afternoon do not answer the last two questions
- The number of records with no missing data must be sufficient for the selected analysis technique if no recovery method is applied to missing data

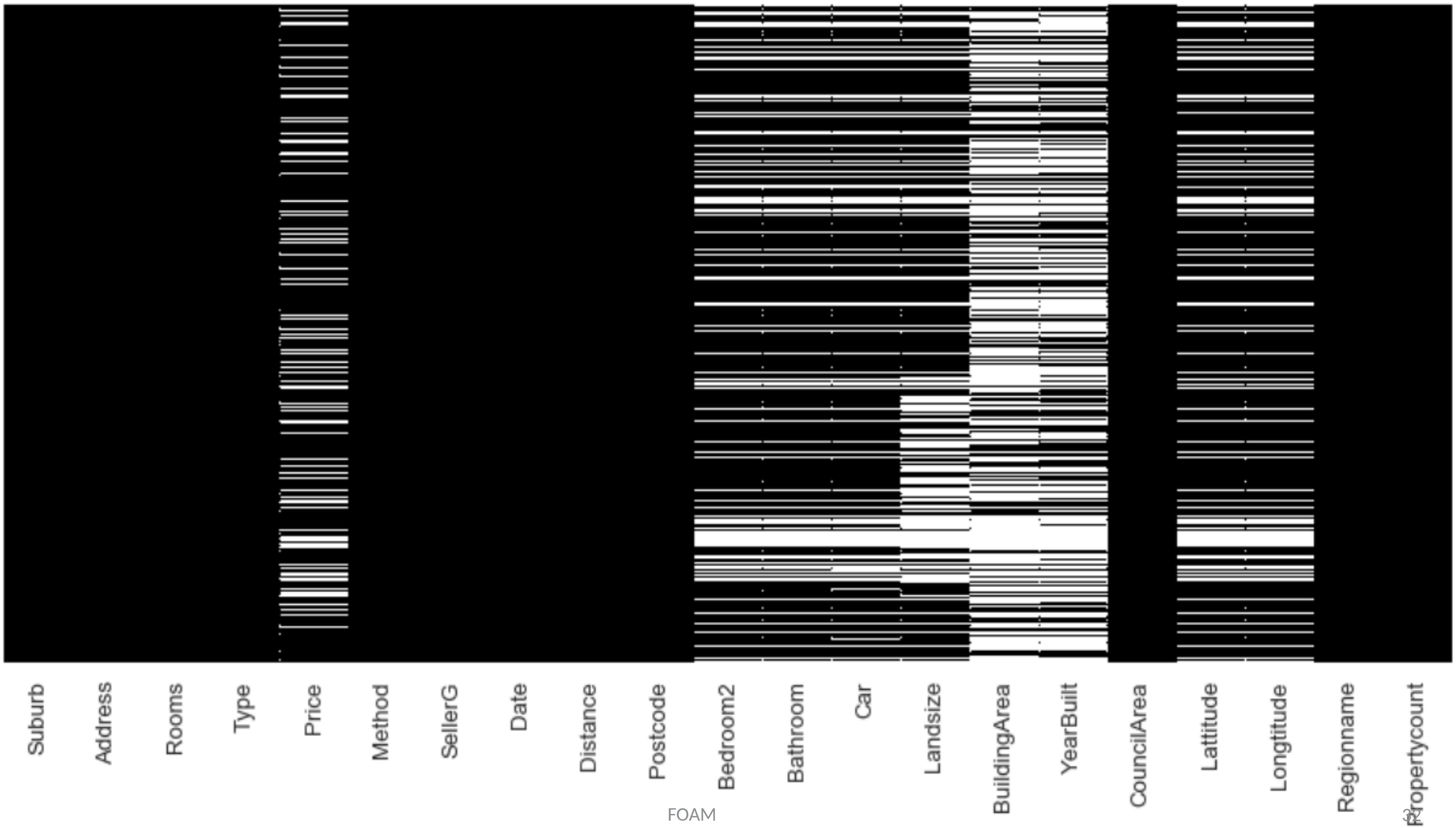
Missing data in Melbourne housing market dataset

```
df.isna().sum()
```

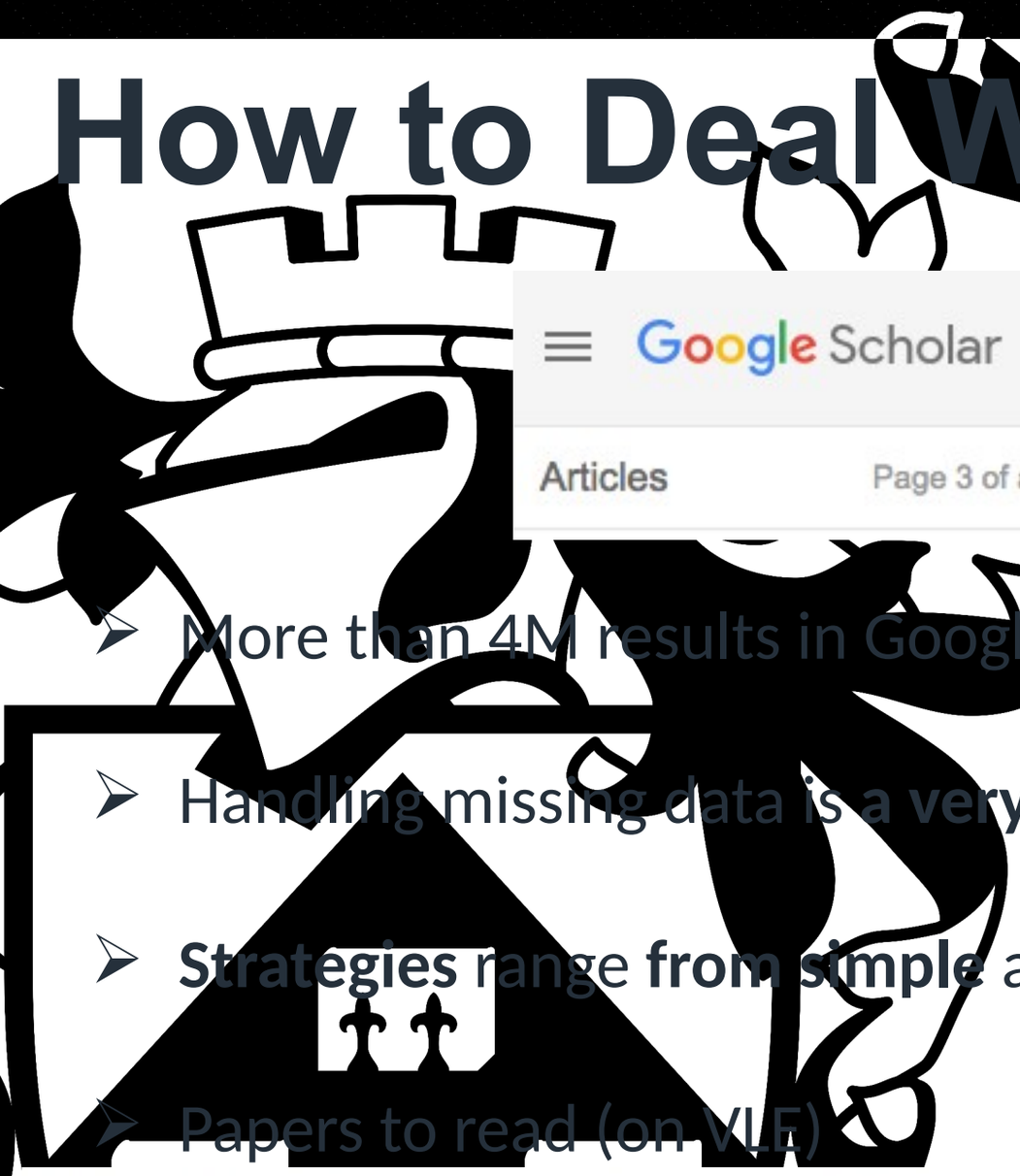
Suburb	0
Address	0
Rooms	0
Type	0
Price	7610
Method	0
SellerG	0
Date	0
Distance	1
Postcode	1
Bedroom2	8217
Bathroom	8226
Car	8728
Landsize	11810
BuildingArea	21115
YearBuilt	19306
CouncilArea	3
Lattitude	7976
Longtitude	7976
Regionname	3
Propertycount	3

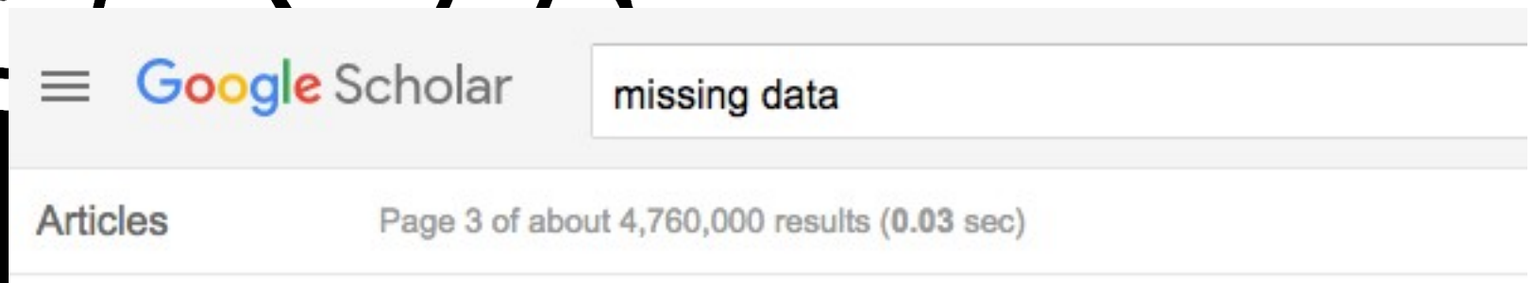
When Should A Variable Be Excluded?

- Visualised missing data for the Melbourne housing market dataset
  - White stripe ➔ missing data item



How to Deal With Missing Data?





- More than 4M results in Google scholar (link)
- Handling missing data is a **very complicated subject**
- **Strategies** range **from simple** and straightforward **to** sophisticated and **complex**
- Papers to read (on VLE)

Missing Data: Five Practical Guidelines

Daniel A. Newman

First Published September 26, 2014 | Research Article

https://doi.org/10.1177/1094428114548590

Article information

ELSEVIER

Journal of School Psychology

Volume 48, Issue 1, February 2010, Pages 5-37

An introduction to modern missing data analyses

Amanda N. Baraldi, Craig K. Enders

List-wise Deletion

Drop records with at least one missing data item

- **Only analyse records with data available for all variables**
  - Records containing missing variables are deleted
  - Rule of thumb – only an option if comprise less than 5% of overall records and data re MCAR.
- Advantages
  - Simplicity
  - Comparability across analysis
- Disadvantages
  - Reduces statistical power (fewer records)
    - Melbourne dataset has 9K entries (instead of the original 34K)
  - Doesn't use all information
  - Results may be biased if missing data is not MCAR!

```
dfMissingDropped = df.dropna(axis='rows', how='any')
dfMissingDropped.shape
```

```
(8887, 21)
```

Variable Elimination

Drop variables (columns) with many data missing

- If too many data is missing for a variable we can remove it from the dataset
- No strict rule when this should happen, proper data analysis is needed
- Advantages
  - Simplicity, easy to implement
  - Effect is easily understandable
- Disadvantages
  - If done blindly, it may affect the results of analysis

```
dfClean = df.drop(["Landsize","Lattitude", "Longtitude",
                   'Bedroom2', 'Bathroom', 'YearBuilt',
                   'BuildingArea', 'Car'], axis=1)
dfClean.isna().sum()
```

```
Suburb           0
Address          0
Rooms            0
Type             0
Price         7610
Method           0
SellerG          0
Date             0
Distance         1
Postcode         1
CouncilArea      3
Regionname       3
Propertycount    3
dtype: int64
```

Variable Elimination

Drop variables (columns) for other reasons

- If it is a predictor (independent) variable and it is highly correlated with another variable,
- If the variable is nearly constant i.e. almost all of the values are the same (~99%)

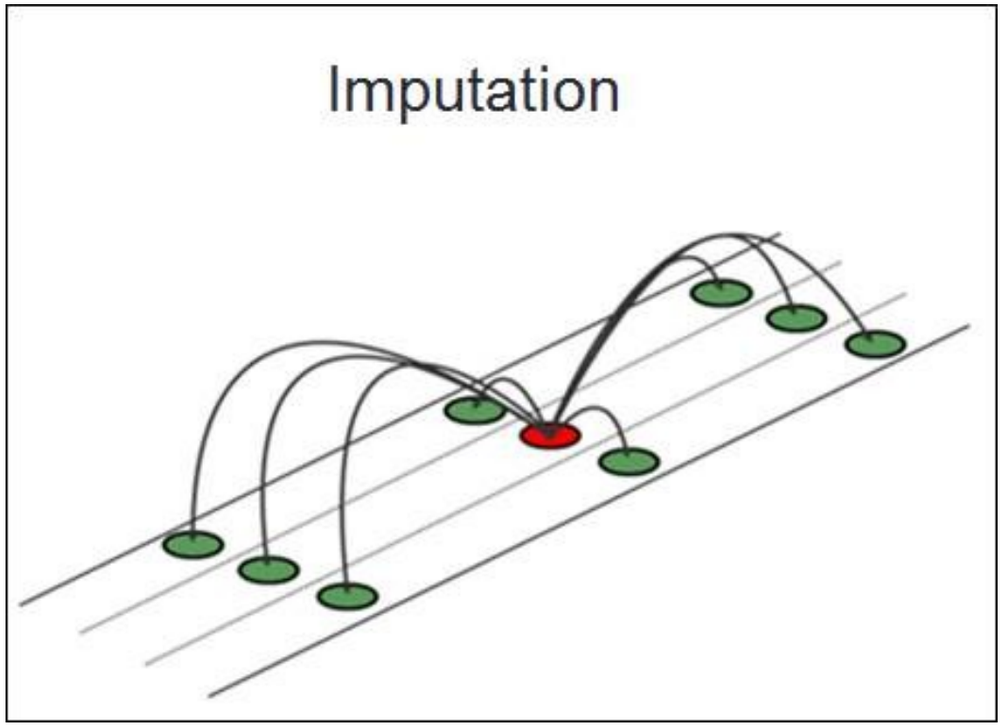
Imputation Strategies

Imputation: Insert a value into data in a "fabricated manner"

- Handle missing values by substituting them with an approximation inferred from the value of the products to which it contributes.

- **Single Imputation Strategies**
  - Mean/Median/Mode imputation
  - Dummy variable control
  - Hot/Cold deck imputation
  - and many more (see papers)

- **Model-Based Imputation Strategies**
  - Regression
  - Multiple imputation
  - and many more (see papers)



https://en.wikibooks.org/wiki/Travel_Time_Reliability_Reference_Manual/Imputing_Missing_Speed_Data

Mean/Median/Mode Imputation

- Calculate mean/median/mode of the non-missing values in a column
- Replace the missing values within each column separately and independently from the others
- Run analysis as if all complete cases
- Advantages
  - Easy and fast
  - Works well with small numerical datasets
- Disadvantages
  - Introduces too much bias and artificially lowers variability of data
  - Ignores relationship between variables ➔ weakens correlations between variables (works only on the column level)

Mean/Median/Mode Imputation

- **Example:** Mean imputation

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

mean()

	col1	col2	col3	col4	col5
0	2.0	5.0	3.0	6.0	7.0
1	9.0	11.0	9.0	0.0	7.0
2	19.0	17.0	6.0	9.0	7.0

- Pandas command: Numeric (top) & Categorical (bottom)

```
#Apply mean imputation to the missing data of the Distance variable
dfClean['Distance'].fillna(dfClean['Distance'].mean(), inplace=True)

#Apply mode imputation to the missing data of the Regionname variable
dfClean['Regionname'].fillna(dfClean['Regionname'].mode()[0], inplace=True)
```

Dummy Variable Adjustment

- Create an **indicator variable** for missing values (per variable or in total)
  - 1=value is missing; 0=value is observed for the record
- Impute missing values to a constant (e.g. mean, median)
- Advantage
  - Uses all available information about a missing record
- Disadvantage
  - Can result in biased estimates

Dummy Variable Adjustment

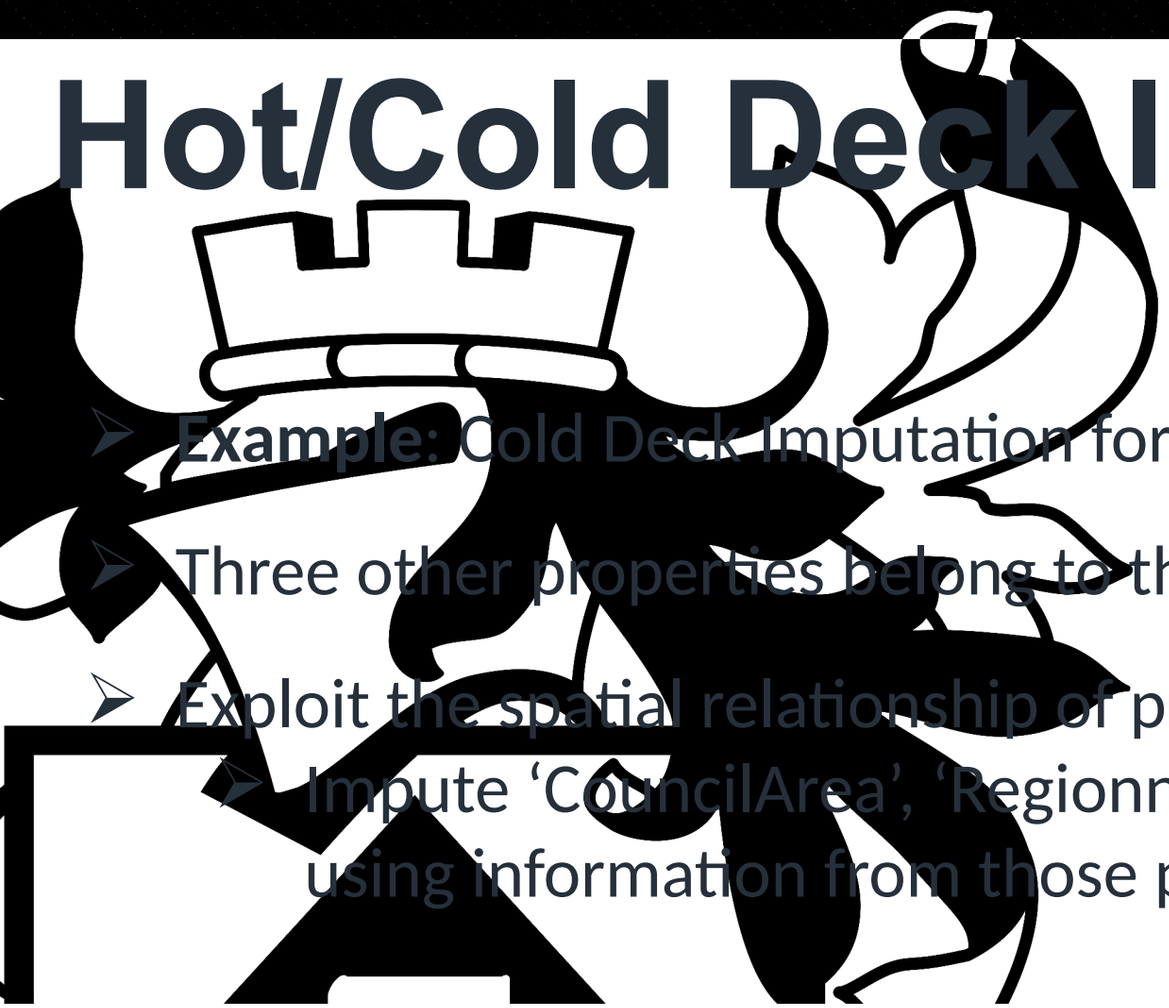
- Some properties have missing BuildingArea values. If we just delete these rows, we lose valuable data. If we put in zero, it would be wrong (no building has zero area).
- Use the **median** BuildingArea to fill missing values.
- **Why the median?**
  - The median is robust to outliers (unlike the mean)
  - If we have a few mansions with huge areas, they won't skew our imputation
  - It represents the "typical" building size in our dataset
- **Create Dummy Variables**
  - **For Property Type (Type):**
    We have 3 categories: h (house), u (unit), t (townhouse)
  - **We create 2 dummy variables:**
    Type_u = 1 if unit, 0 otherwise
    Type_t = 1 if townhouse, 0 otherwise
    (Type_h) Houses become our **reference category**

Price = $\beta_0 + \beta_1 \times$BuildingArea + $\beta_2 \times$Type_u + $\beta_3 \times$Type_t

Hot/Cold Deck Imputation

- **Replace missing data with actual values from the most similar record or best known value**
- **Hot Deck Imputation**
  - Find all the sample **records** which are **similar to other variables**, then **randomly choose one** of their values to fill in the missing data
- **Cold Deck Imputation**
  - **Systematically** choose the value from a record with **similar values to other variables** (e.g. the third item of each collection)
- Advantage
  - Constrained by pre-existing values
- Disadvantage
  - Missing data process indicates variables to use for similarity extraction
  - Must define appropriate best known values

Hot/Cold Deck Imputation



- **Example:** Cold Deck Imputation for property at 2/3 Kingsley St (yellow shaded)
- Three other properties belong to the **same suburb** (orange shaded)
- Exploit the spatial relationship of properties in this dataset
  - Impute ‘CouncilArea’, ‘Regionname’ and ‘PropertyCount’ using information from those properties

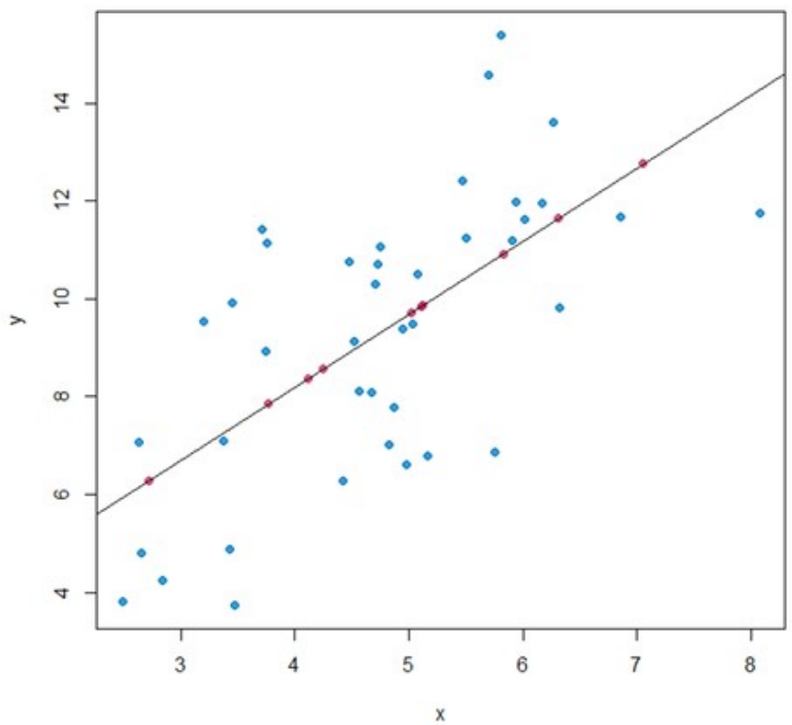
Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	CouncilArea	Regionname	Propertycount
Camberwell	6 Kingsley St	4	h	3550000.0	S	Marshall	13/05/2017	7.8	3124.0	Boroondara City Council	Southern Metropolitan	8920.0
Camberwell	22 Kingsley St	5	h	3190000.0	S	Marshall	12/08/2017	7.7	3124.0	Boroondara City Council	Southern Metropolitan	8920.0
Camberwell	13 Kingsley St	4	h	NaN	S	Jellis	21/10/2017	7.7	3124.0	Boroondara City Council	Southern Metropolitan	8920.0
Camberwell	2/3 Kingsley St	2	h	825000.0	VB	Jellis	11/11/2017	7.7	3124.0	NaN	NaN	NaN
Elwood	4/25 Kingsley St	1	u	451000.0	S	Chisholm	19/11/2016	7.7	3184.0	Port Phillip City Council	Southern Metropolitan	8989.0
Elwood	7/25 Kingsley St	2	u	582500.0	SP	Chisholm	26/07/2016	7.7	3184.0	Port Phillip City Council	Southern Metropolitan	8989.0
Elwood	2/25 Kingsley St	2	u	660000.0	S	Chisholm	27/05/2017	7.2	3184.0	Port Phillip City Council	Southern Metropolitan	8989.0
Elwood	10 Kingsley St	4	h	NaN	SN	Chisholm	3/03/2018	7.2	3184.0	Port Phillip City Council	Southern Metropolitan	8989.0
Ivanhoe	8 Kingsley St	2	h	NaN	SP	Miles	3/06/2017	7.8	3079.0	Banyule City Council	Eastern Metropolitan	5549.0
Ivanhoe	7 Kingsley St	3	h	NaN	SN	Miles	18/11/2017	7.8	3079.0	Banyule City Council	Eastern Metropolitan	5549.0

Regression Imputation

- **The imputed value is predicted from a regression equation based on other dataset variables** (regression will be covered next week)

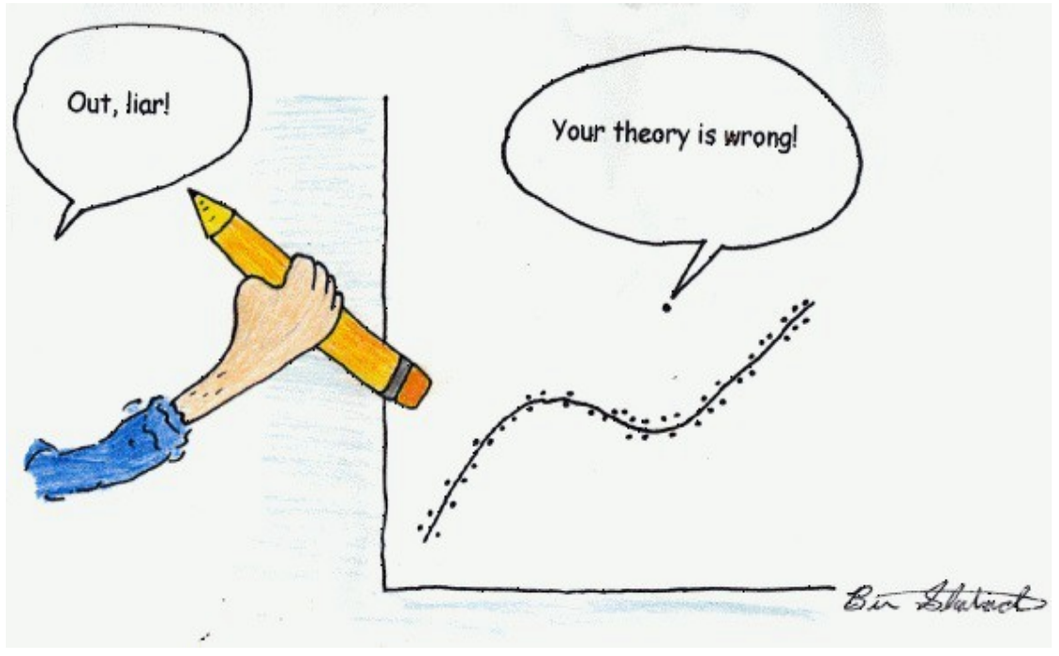
- Advantages
  - Uses information from observed data
  - Preserves relationships among variables in the imputation model

- Disadvantage
  - Limited variability around predicted values
  - Overestimates model fit and correlation estimate
  - Requires sufficient relationships among variables to generate valid predicted values



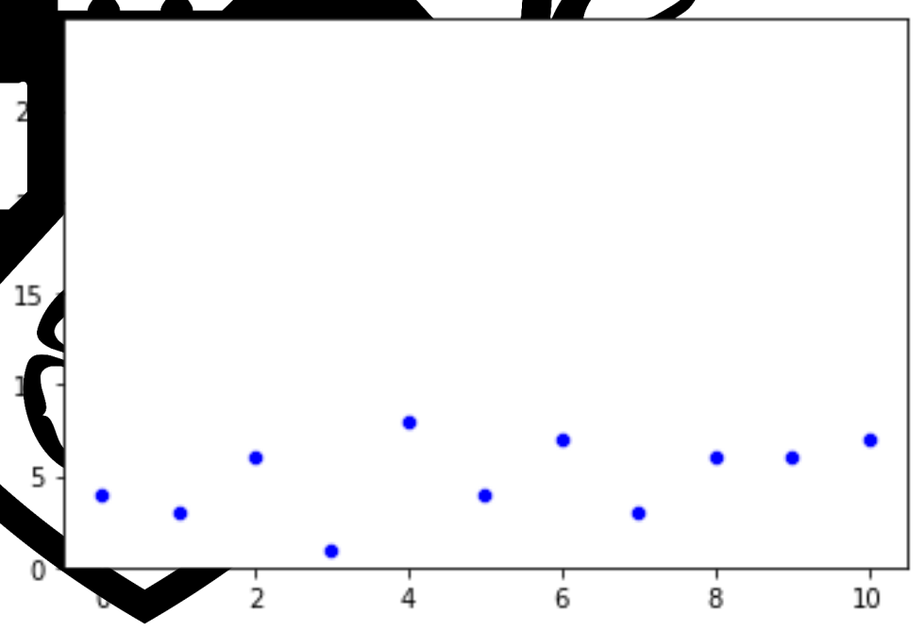
Outliers: What are they?

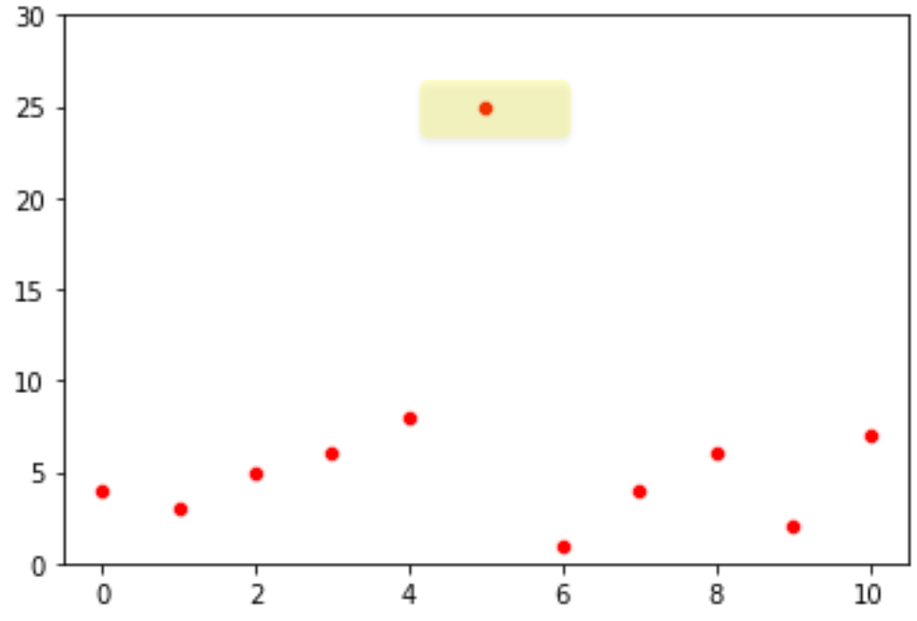
- Data (records) **significantly different** from the **majority of data within the dataset** (or population)
- They are **rare**, or **distinct**, or **do not fit** in some way with the dataset
- Examples:
  - A student's average mark is over 95% while the rest of the class is around 70%
  - Within the bank transactions of a customer there is an entry 10 times larger than most of their transactions
  - The house price of 3 houses in York is over £5m while the average price is around £500K



Outliers: What are they?

	Without Outlier	With Outlier
Values	[4, 3, 6, 1, 8, 4, 7, 3, 6, 6, 7]	[4, 3, 6, 1, 8, **26**, 7, 3, 6, 6, 7]
Mean	55/11 = 5	77/11 = **7**
Median	6	6
Mode	6	6
Std	2.045	**6.34**





Outliers: Where do they come from?

- **Data entry errors:** Human errors caused during data collection or recording
  e.g. While trying to pay your utility bills online you do not add the decimal place and pay £5000 for your water bill instead of £50,00

- **Measurement errors:** The measurement instrument is faulty
  e.g. a temperature sensor on Raspberry Pi produces wrong values

- **Intentional:** Unwillingness to disclose information, hence inputting false information in forms
  e.g. Teens might under report the amount of alcohol they consume

- **Data Processing Error**: Data extraction or manipulation (while doing data science) may lead to outliers in the dataset.

- **Natural Outlier:** A genuine outlier
  e.g. Michael Jordan (basketball), Usain Bolt (record breaking sprints)

Outliers: What is their impact?

- Outliers can have **significant impact** on the results of our data science project (data analysis and statistical modeling)
- They can **introduce bias or influence estimates** of significant interest
- Machine learning algorithms are **sensitive to provided data** (the range and distribution of data)
  - Outliers can adversely affect the training process
  - They can increase significantly training times, produce less accurate models and poorer results
- **Outliers ARE NOT ALWAYS BAD**
  - Anomalies in heartbeat data can help in predicting heart diseases
  - Anomalies in traffic patterns can help in predicting accidents

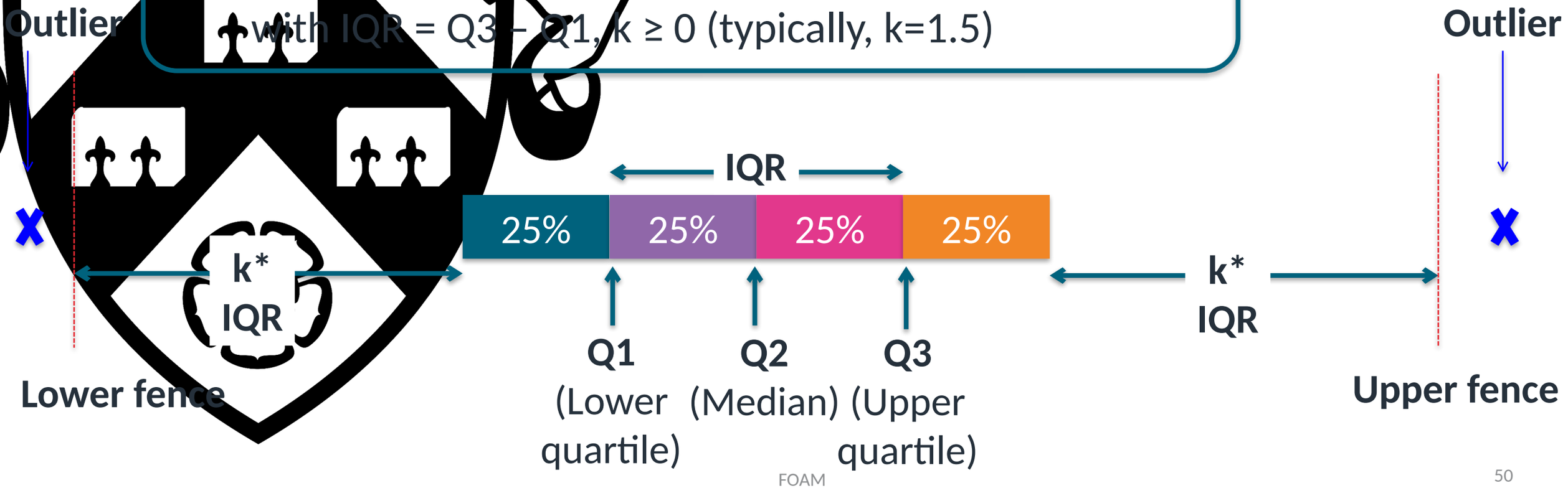
Hence, it is important to **detect and understand the origin of outliers!**

Outliers: How to detect them?

- **No precise way to identify outliers; they are specific to each dataset!**
- Data scientists must interpret the raw data and decide whether an outlier exists or not
- **Help?**
  - We can use statistical methods to identify records that appear to be unlikely given the data available
- This does not mean that the values identified are outliers and should be removed
- **These are rare events that deserve/require a closer second look!**

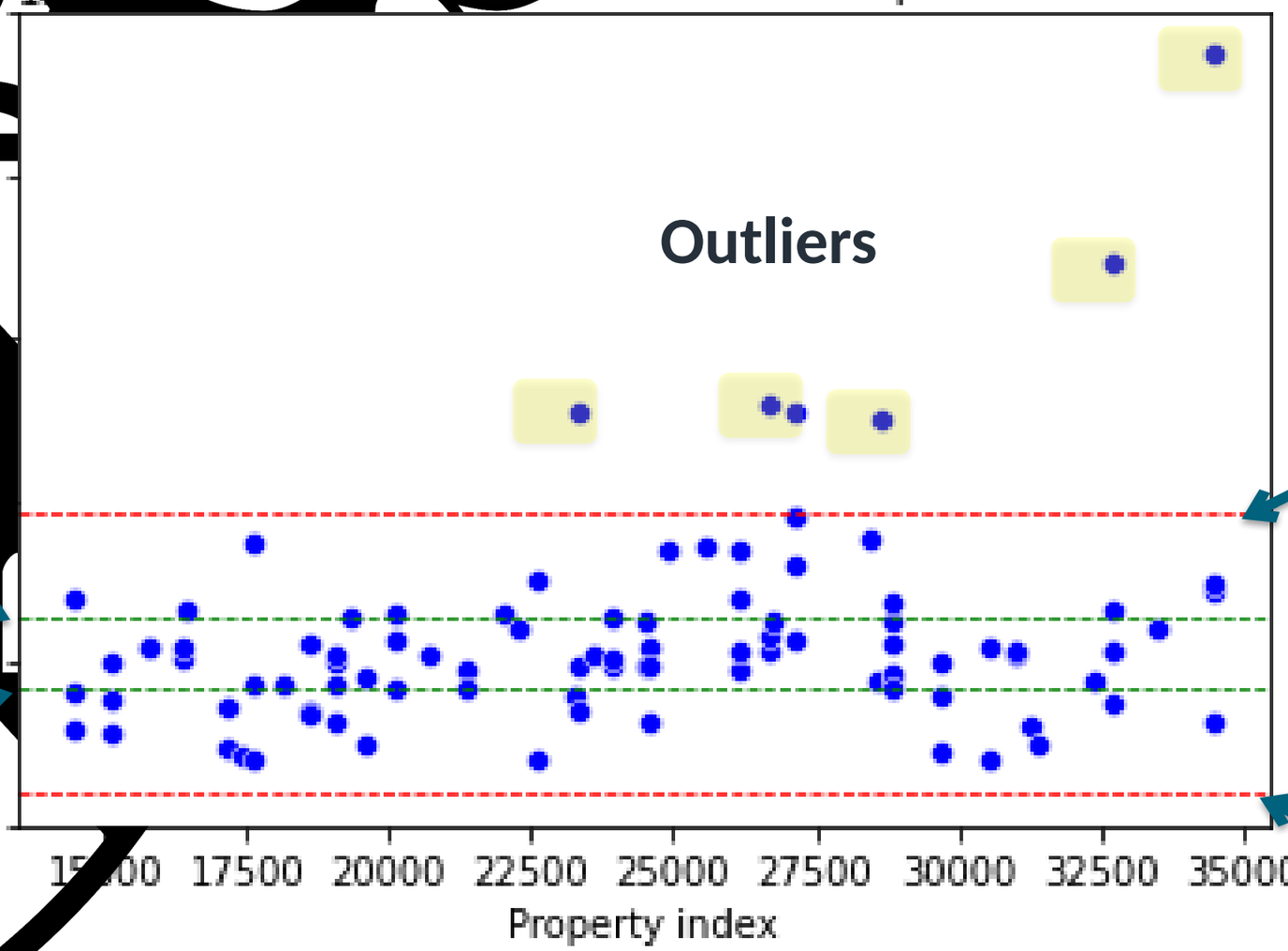
Outlier Detection Using IQR

- **IQR**
  - Measures the variability of data considering the range covered by the MIDDLE 50% of the data
- A **record (x) is a suspected outlier if**:
  - **x > upperFence**, where upperFence = Q3 + k* (IQR)
  - **x < lowerFence**, where lowerFence = Q1 - k *(IQR)
  - with IQR = Q3 – Q1, k ≥ 0 (typically, k=1.5)



Outlier Detection Using IQR

- The couple decides to investigate the properties in the Western Victoria region



Upper fence
Q1 + 1.5*IQR
587500

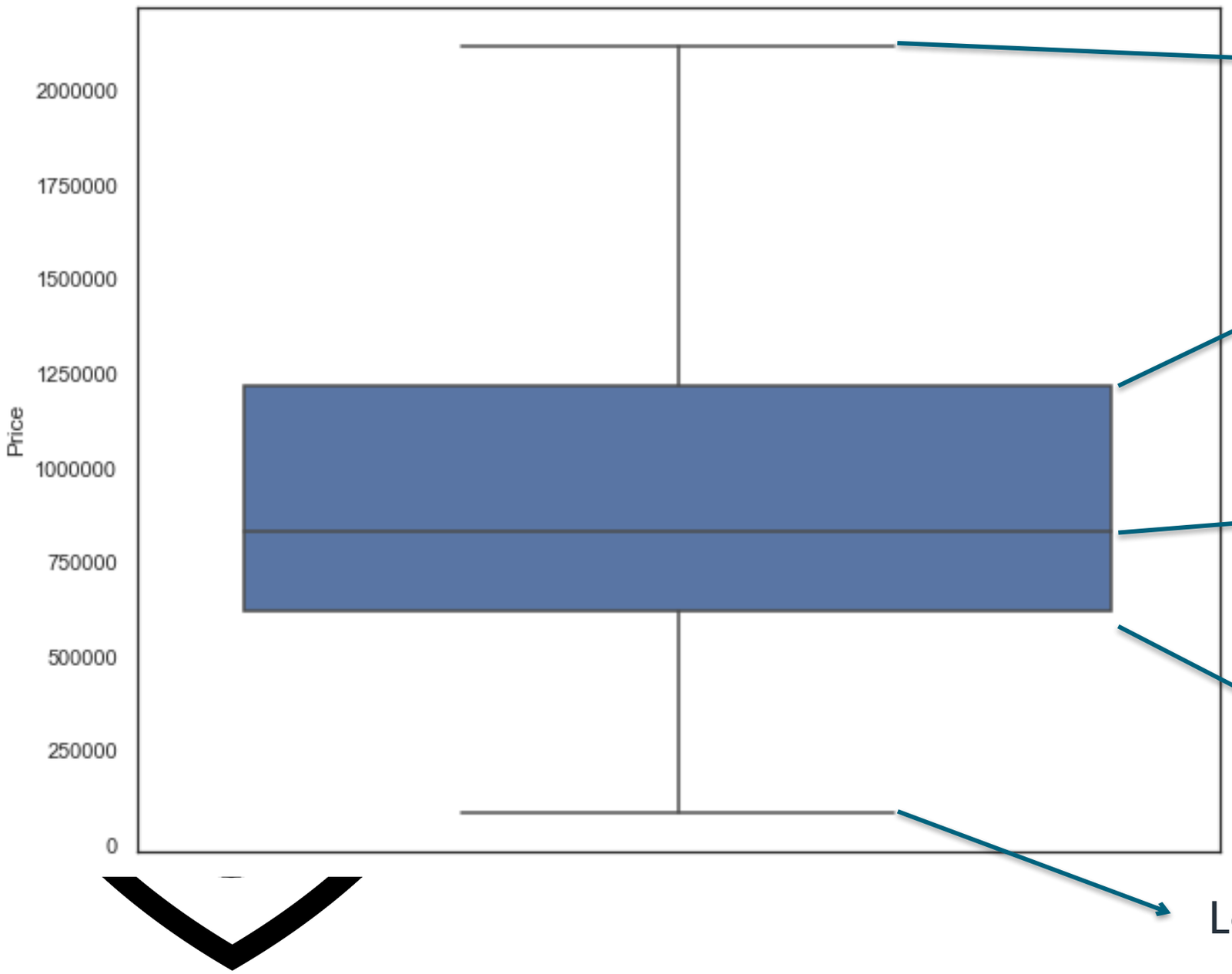
Q3
457000

Q1
370000

Lower fence
Q1 – 1.5*IQR
239500

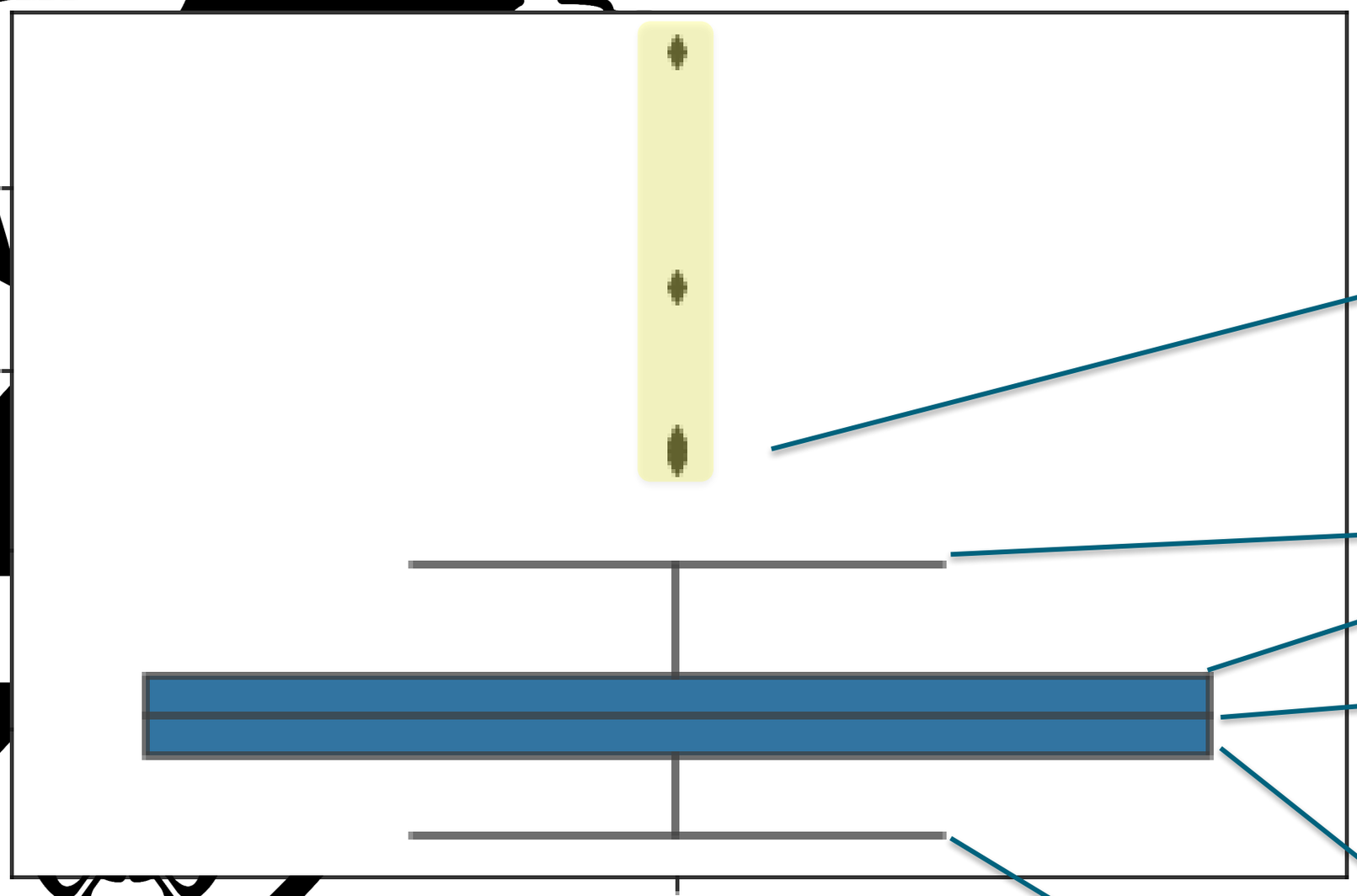
Outlier Detection Using Boxplots

- Melbourne housing market (without outliers)



Outlier Detection Using Boxplots

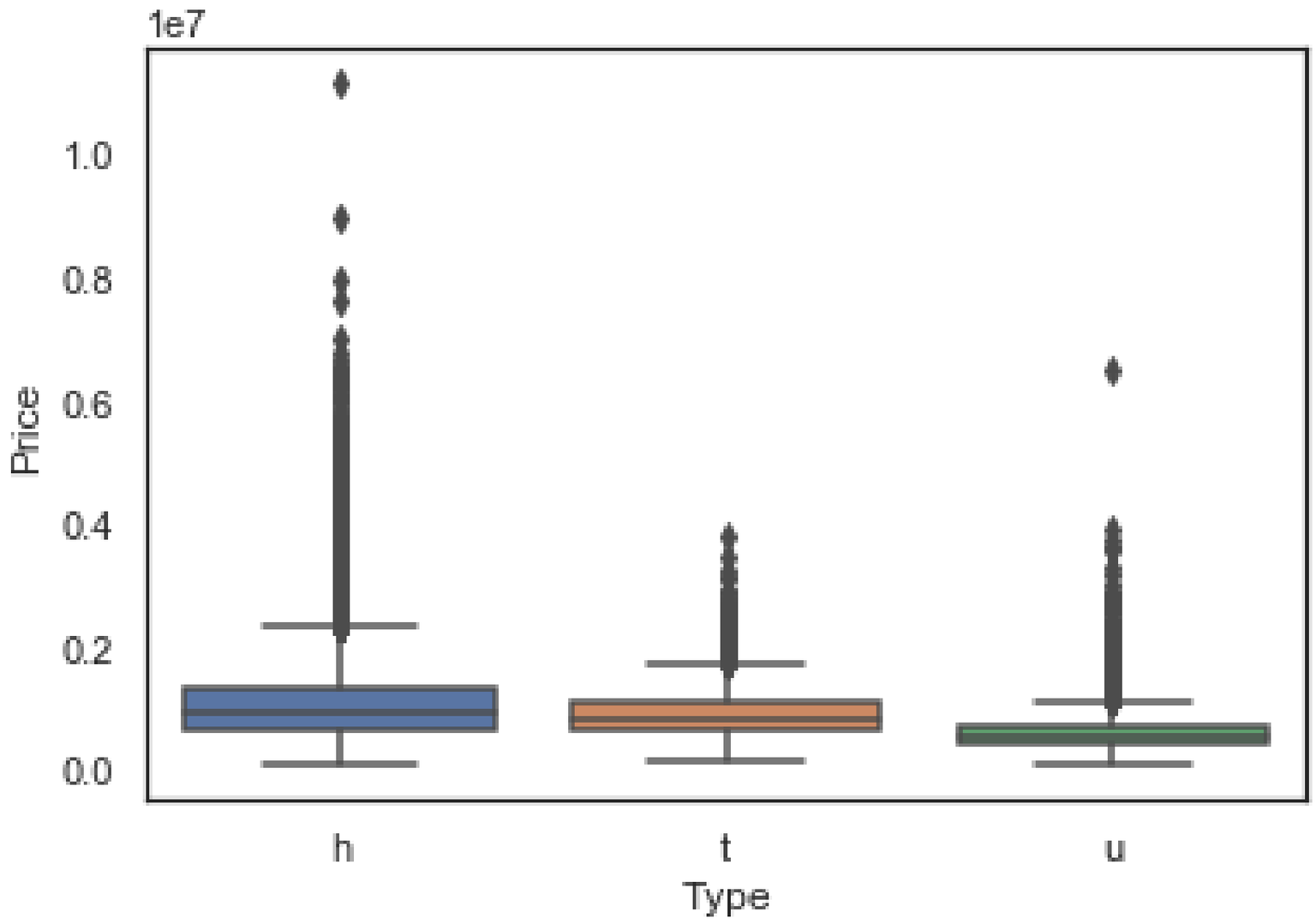
- Melbourne housing market (with outliers)



- What information do they convey? (symmetric data, variability)

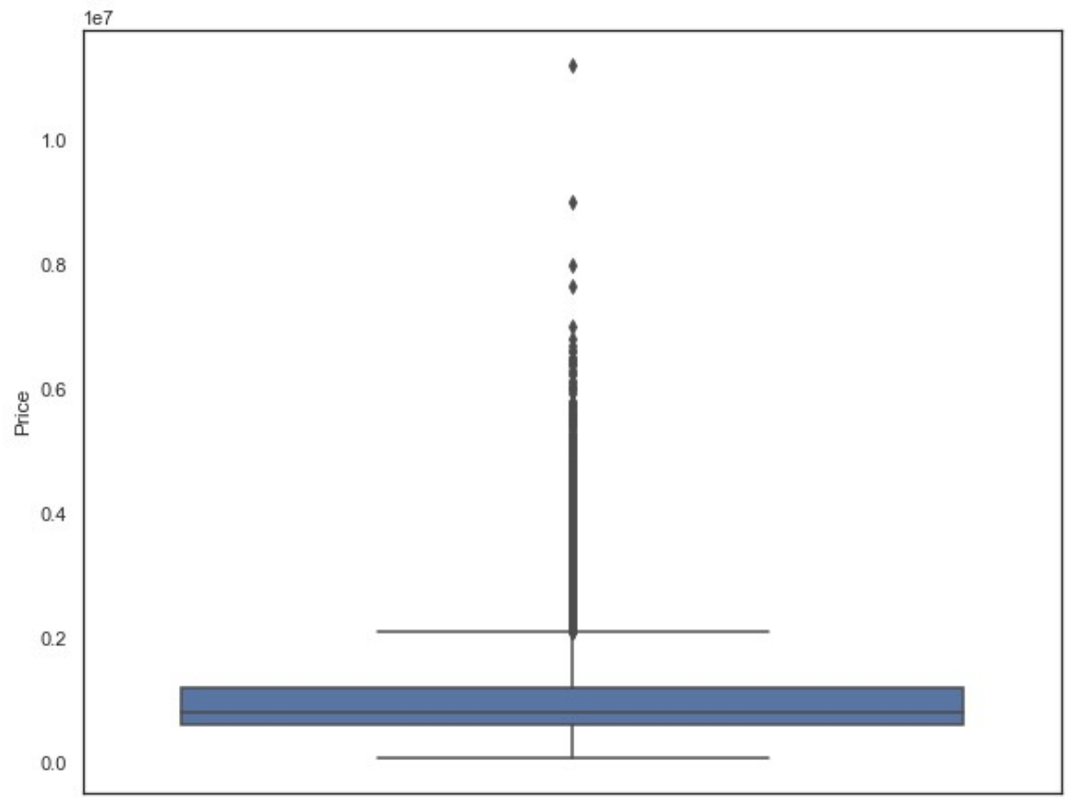
Outlier Detection Using Boxplots

- Melbourne housing market per property type



Boxplot Construction

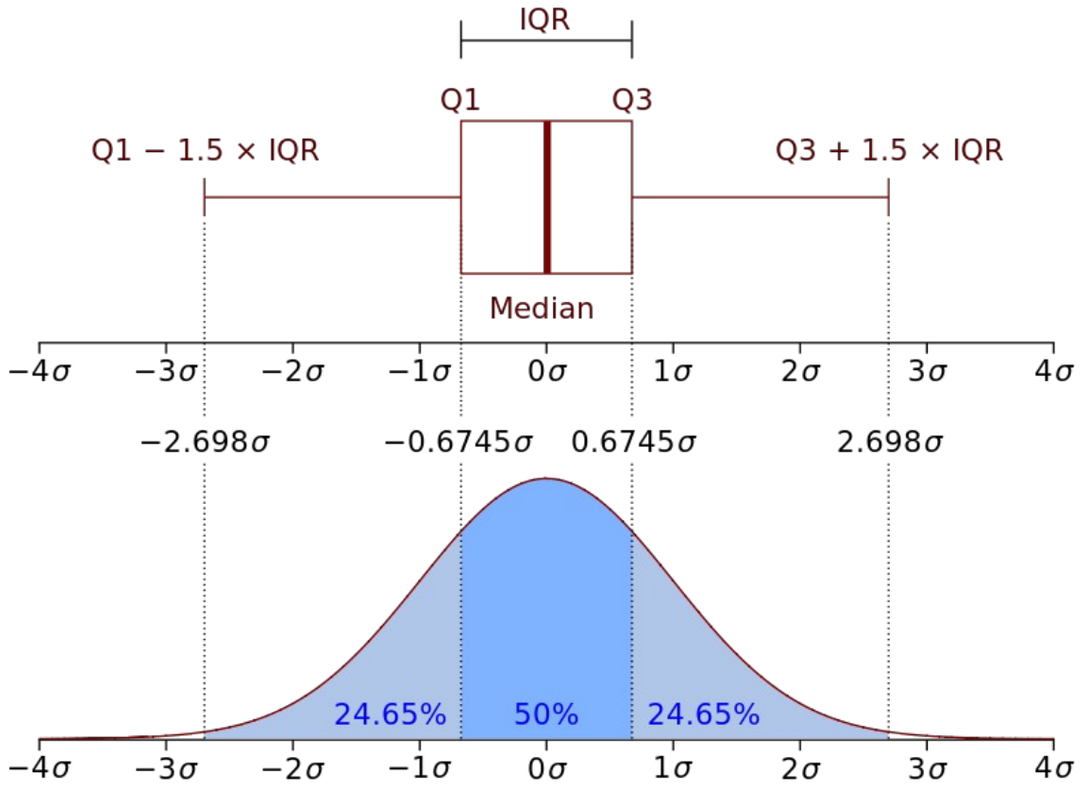
- **Recipe**
  1. Given a dataset, find the median, Q1 and Q3
  2. Draw a horizontal line for the median (Q2)
  3. Draw a horizontal line for the upper and lower quartiles (Q1, Q3)
  4. Connect these quartiles into a box
  5. Find the upper and lower fences (e.g., using the IQR method)
  6. Draw a horizontal line for the lowest and largest values within the fences
  7. Connect them to the box with whiskers
  8. Add the outliers using another shape



Outlier Detection Using Std

- If the **data distribution** in the dataset is **Gaussian-like**, we can use the **standard deviation of the sample** as a cut-off for identifying outliers

- 1 Standard Deviation from the Mean: 68%
- 2 Standard Deviations from the Mean: 95%
- 3 Standard Deviations from the Mean: 99.7%



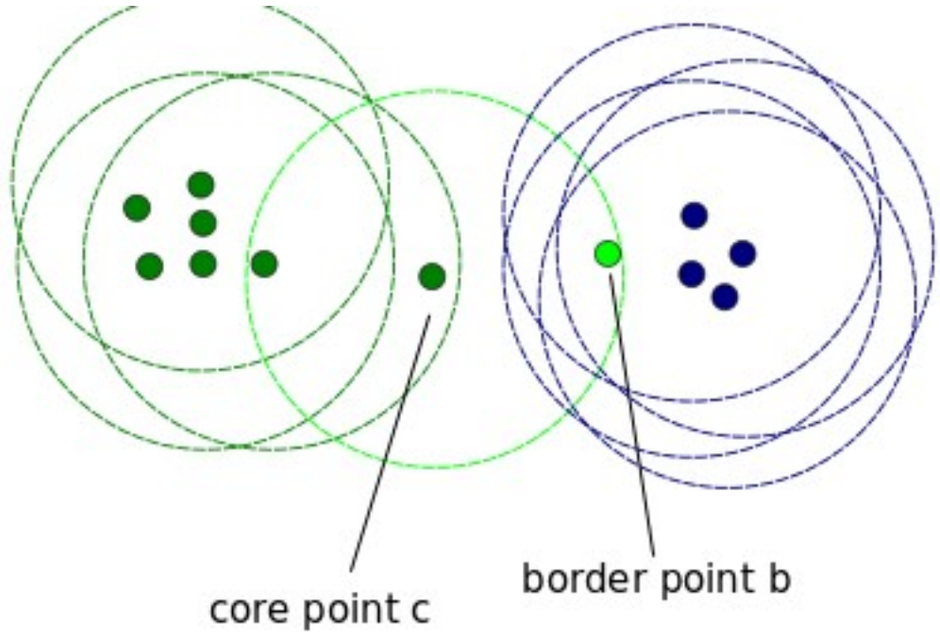
- A value that falls **outside of 3 standard deviations**, is an **unlikely or rare** event

- Use this **rule of thumb** as a cut-off for identifying outliers

- For our case study, this detection strategy returns the two most expensive properties as outliers

Other Methods For Outlier Detection

- **Clustering Algorithms**
  - Cluster data into groups
  - Can be also used for anomaly detection with single or multi-dimensional data
  - Depends on the required number of neighbours, the distance and the selected distance measure between clusters



- **Isolation Forest**
  - Data partitioning done using a set of trees
  - An anomaly score is given depending on how isolated the point is in the structure
  - The score is used to identify outliers from normal observations

- and many others!

https://scikit-learn.org/stable/auto_examples/plot_anomaly_comparison.html

Outliers: How to treat them?

- **Difficult decision:** should you remove or correct outliers?

- **Option 1: Drop outliers**
  - If they occur due to data entry error, data processing error
  - If outliers are very small in numbers

- **Option 2: Quantile-based Flooring and Capping**
  - Assign the lower values to the 10th percentile (floor)
  - Assign the higher values to the 90th percentile (cap)

```
#Find outliers based on std strategy
outliersProp = wvProp[wvProp['Price']>meanPrice + 3*stdPrice ]

#Find the 10th and 90th percentile
p10 = wvProp['Price'].quantile(0.10)
p90 = wvProp['Price'].quantile(0.90)

#Find outliers above the 10th and 90th percentile
outliersProp10 = outliersProp[outliersProp['Price']<p10]
outliersProp90 = outliersProp[outliersProp['Price']>p90]

#Assign the values to outliers
wvProp.loc[outliersProp90.index,'Price']=p90
wvProp.loc[outliersProp10.index,'Price']=p10
```

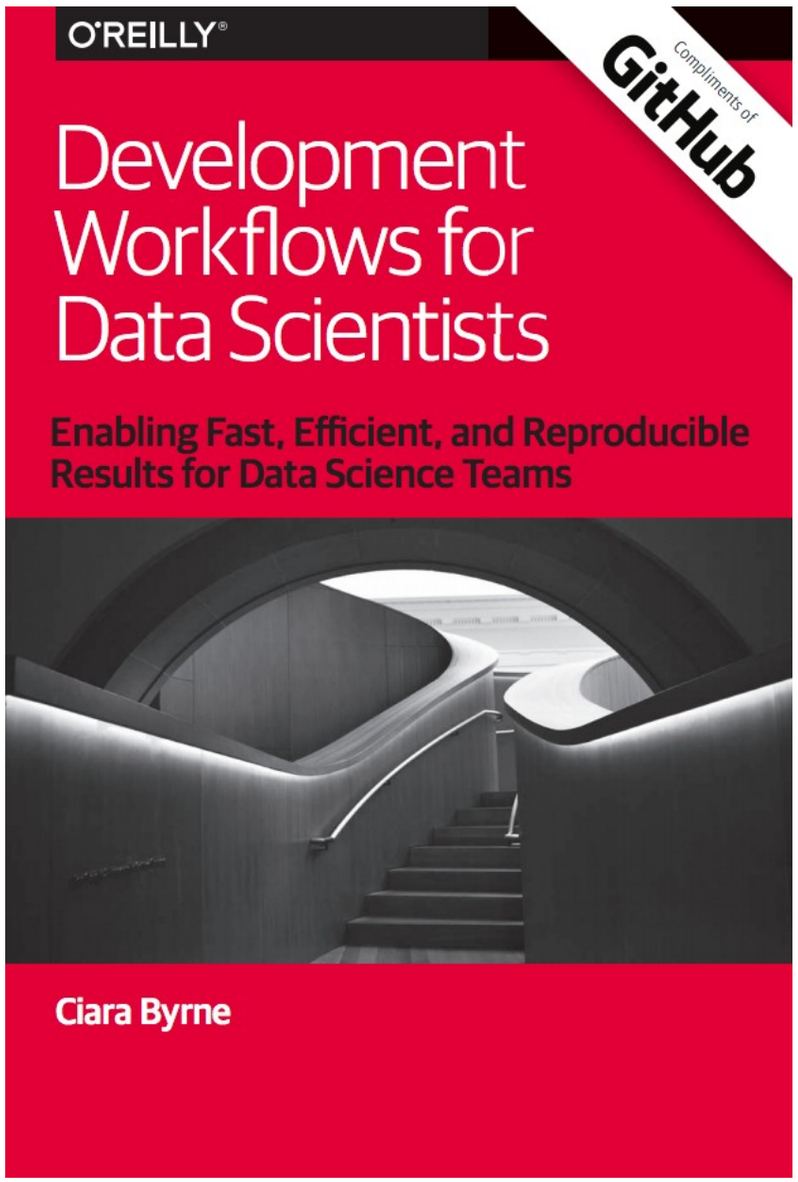
Outliers: How to treat them?

- **Option 3**: Apply imputation methods (Similarly to missing data)
  - Median, Hot/Cold deck imputation, Regression

- **Option 4**: Log transformation
  - Transformation reduces the variation caused by extreme values
  - Logarithmic (most common), square root, or square transformations

Further reading

- Development Workflows for Data Scientists
  - Enabling Fast, Efficient, and Reproducible Results for Data Science Teams
  - Available online at: https://resources.github.com/downloads/development-workflows-data-scientists.pdf



- OSEMN Process
  - http://www.dataists.com/2010/09/a-taxonomy-of-data-science

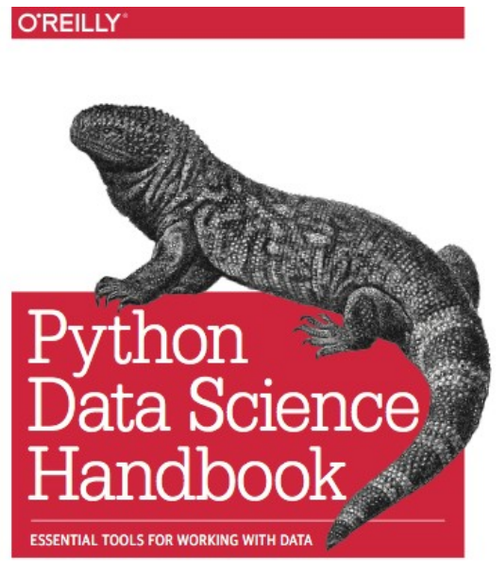
- The Team Data Science Process by Microsoft
  - https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview

- The Data Science Workflow at BinaryEdge
  - https://blog.binaryedge.io/2015/09/08/the-data-science-workflow

Practical Essentials

- Python Data Science Handbook (Reading list)
  - Available online
    https://jakevdp.github.io/PythonDataScienceHandbook

- Official Seaborn Tutorial
  https://seaborn.pydata.org/tutorial.html



2. Introduction to NumPy

- Understanding Data Types in Python
- The Basics of NumPy Arrays
- Computation on NumPy Arrays: Universal Functions
- Aggregations: Min, Max, and Everything In Between
- Computation on Arrays: Broadcasting
- Comparisons, Masks, and Boolean Logic
- Fancy Indexing
- Sorting Arrays
- Structured Data: NumPy's Structured Arrays

3. Data Manipulation with Pandas

- Introducing Pandas Objects
- Data Indexing and Selection
- Operating on Data in Pandas
- Handling Missing Data
- Hierarchical Indexing
- Combining Datasets: Concat and Append
- Combining Datasets: Merge and Join
- Aggregation and Grouping
- Pivot Tables

4. Visualization with Matplotlib

- Simple Line Plots
- Simple Scatter Plots
- Visualizing Errors
- Density and Contour Plots
- Histograms, Binnings, and Density
- Customizing Plot Legends
- Customizing Colorbars
- Multiple Subplots
- Text and Annotation
- Customizing Ticks

Jupyter Notebook

- https://jupyter-notebook.readthedocs.io/en/stable/

- A powerful tool for interactively developing and presenting data science projects

- **A notebook is a single document where you can**
  - write and run code, view the output
  - add text, explanations, formulas, charts, etc
  - explain your data science results and share them

- We will be using it for writing Python and Markdown (an easy to learn markup language for formatting plain text)

- On VLE, there is a tutorial on how to install Jupyter on your machine

Summary

- Data Science Lifecycle
- Exploratory Data Analysis
- Handling Missing Data
- Identifying and Handling Outliers