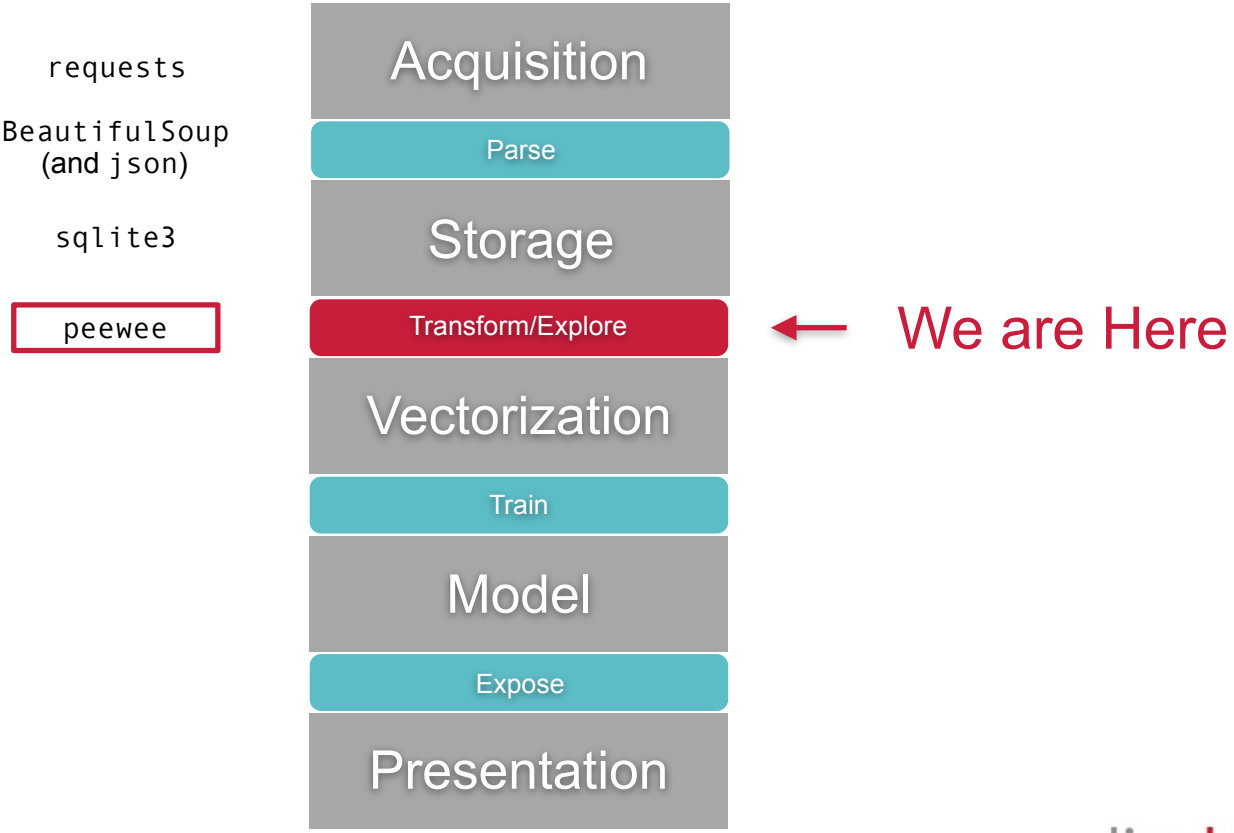
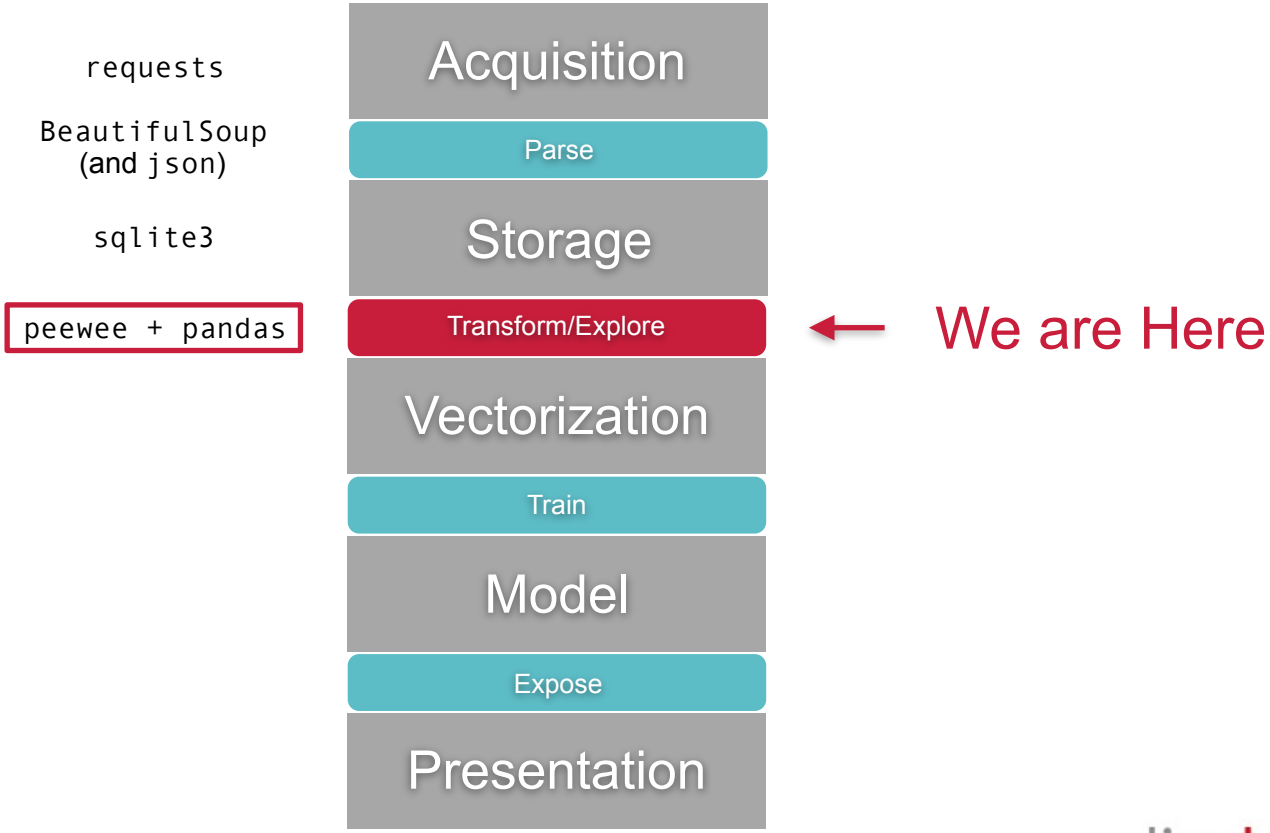
livelessons
video instruction from technology experts

Lesson 6: Validating Data: Provenance and Quality Control

Process + Tools



Process + Tools



Cheaper Beds, **Better** Breakfasts

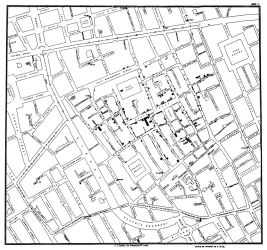
Aid users in their discovery of relevant listings

For now, we are interested not in listings themselves but the context around the listing

Which neighborhoods are the most explorable?

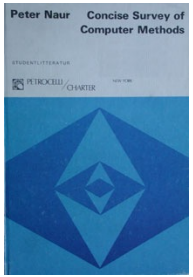
Which listings have the most lively and active POI around them?

A Brief Historical Diversion



Broad Street Cholera Outbreak (1854)

John Tukey (1962)

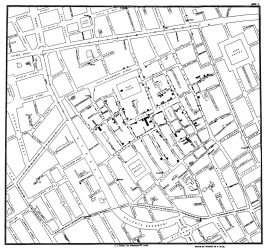


Data Science Coined (1974)

First Conference (1996)

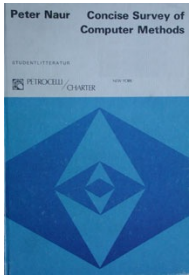
Building Data Science Teams (2008)

A Brief Historical Diversion



Broad Street Cholera Outbreak (1854)

John Tukey (1962)



Data Science Coined (1974)

First Conference (1996)

Building Data Science Teams (2008)

What Is Exploratory Data Analysis?

- Developed at Bell Labs in the 1960's by John Tukey
- Techniques used to visualize and summarize data
    - Five-number summary: `describe()`
    - Distributions: box plots, stem and leaf, histogram, scatterplot

What is Exploratory Data Analysis?

But as much as EDA is a set of tools, it's also a mindset. And that mindset is about your relationship with the data… EDA happens between you and the data and isn't about proving anything to anyone else yet.

- Cathy O'Neil (Doing Data Science)

What is Exploratory Data Analysis?

But as much as EDA is a set of tools, it's also a mindset. And that mindset is about your relationship with the data… EDA happens between you and the data and isn't about proving anything to anyone else yet.

- Cathy O'Neil (Doing Data Science)

What is Exploratory Data Analysis?

> *But as much as EDA is a set of tools, it's also a mindset. And that mindset is about your relationship with the data… EDA happens between you and the data and isn't about proving anything to anyone else yet.*

- Cathy O'Neil (Doing Data Science)

What is Exploratory Data Analysis?

> *But as much as EDA is a set of tools, it's also a mindset. And that mindset is about your relationship with the data… EDA happens between you and the data and isn't about proving anything to anyone else yet.*

- Cathy O'Neil (Doing Data Science)

Goals of Exploratory Data Analysis

- Gain greater insight
- Validate our data (consistency and completeness)
- Make comparisons between distributions
- Find outliers
- Treat missing data
- Summarize data (a statistic -> one number that represents many #'s)

Common Questions

- How many records in total are there?
- How many missing (or null) values are there?
- How many unique values does each column contain?
- What are the most common values for each column?
- For numeric columns, what are the summary statistics?
- How are the values of each column distributed?

Harder Questions

- How are the records of one table related to another?
- Deduplication and Entity Resolution
- Are there outliers?
- What real world process does each column represent?
- How was the data collected? And what bias has this introduced?

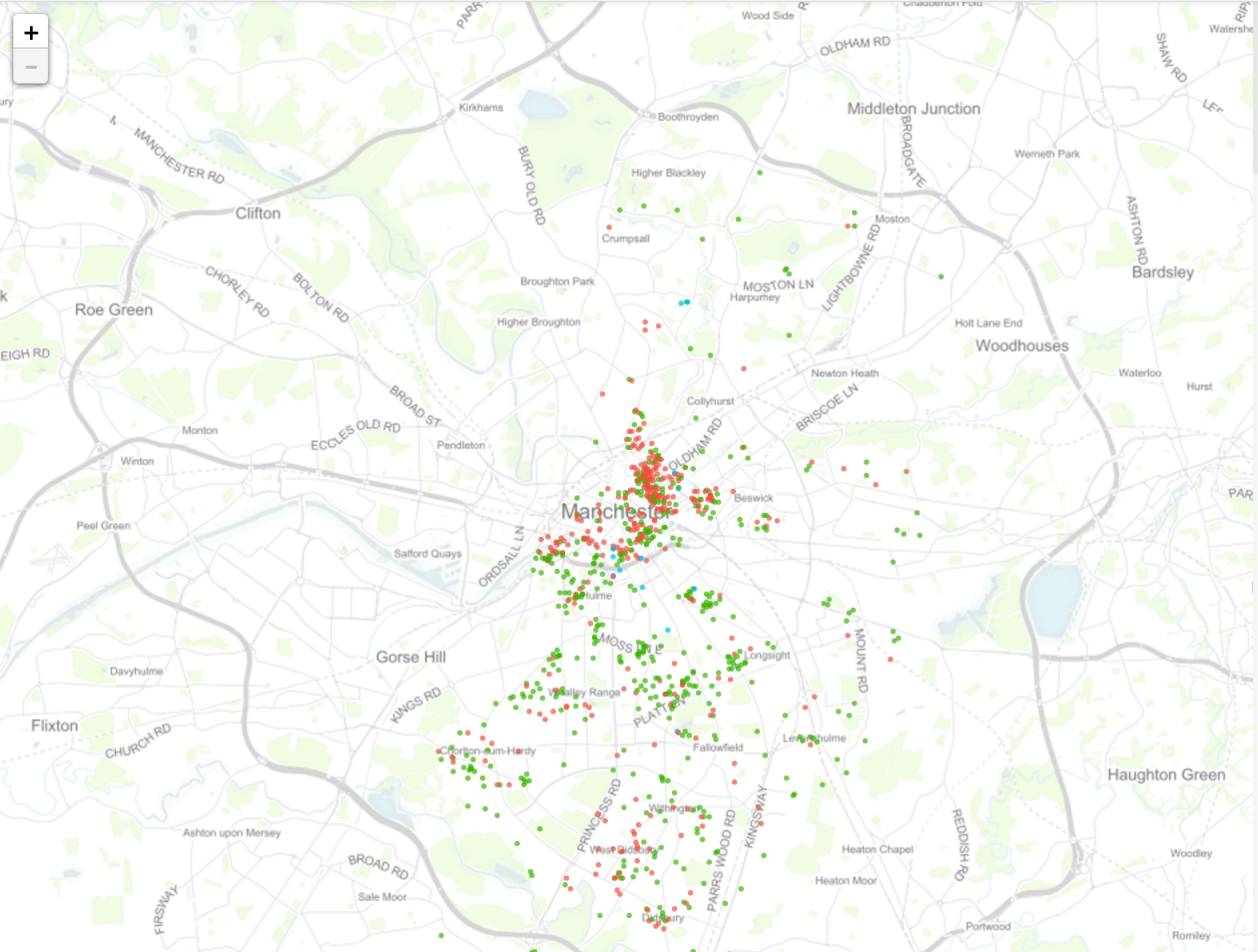
Harder Questions

- How are the records of one table related to another?
- Deduplication and Entity Resolution
- Are there outliers?
- What real world process does each column represent?
- How was the data collected? And what bias has this introduced?

Hang tight, we'll get back to our terminal soon enough…

First we should exhaust our other (faster) options

Remember… iteration



Manchester

Filter by:

Manchester

865
out of **865** listings (100%)

About Airbnb in Manchester

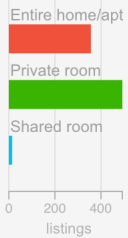
How is Airbnb really being used in and affecting your neighbourhoods?

Room Type

☐ Only entire homes/apartments

Airbnb hosts can list entire homes/apartments, private or shared rooms.

Depending on the room type, availability, and activity, an airbnb listing could be more like a hotel, disruptive for neighbours, taking away housing, and illegal.



41.3%
entire homes/apartments

£76
price/night

357 (41.3%)
entire home/apartments

494 (57.1%)
private rooms

14 (1.6%)
shared rooms

Activity

☐ Only **recent** and **frequently** booked

Airbnb guests may leave a review after their stay, and these can be used as an indicator of airbnb activity.

The minimum stay, price and number of reviews have been used to estimate the **occupancy rate**,

103
estimated nights/year

1.7
reviews/listing/month

Manchester, England, United Kingdom

See Manchester data visually here.

Date Compiled	City	File Name	Description
10 April, 2016	Manchester	listings.csv.gz	Detailed Listings data for Manchester
10 April, 2016	Manchester	calendar.csv.gz	Detailed Calendar Data for listings in Manchester
10 April, 2016	Manchester	reviews.csv.gz	Detailed Review Data for listings in Manchester
10 April, 2016	Manchester	listings.csv	Summary information and metrics for listings in Manchester (good for visualisations).
10 April, 2016	Manchester	reviews.csv	Summary Review data and Listing ID (to facilitate time based analytics and visualisations linked to a listing).
N/A	Manchester	neighbourhoods.csv	Neighbourhood list for geo filter. Sourced from city or open source GIS files.
N/A	Manchester	neighbourhoods.geojson	GeoJSON file of neighbourhoods of the city.

Melbourne, Victoria, Australia

https://www.databasic.io/en/wtfcsv/results/5792498a789166000a54729c

data BASIC.io

WTF CSV

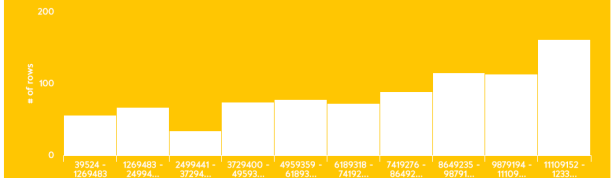
WTFcsv: manchester_04_10_2016.csv

865 rows of data grouped into 15 columns.

Here's some metadata about each column.
These results will expire in 60 days

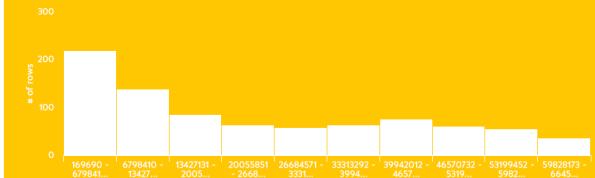
What do I do next?

id



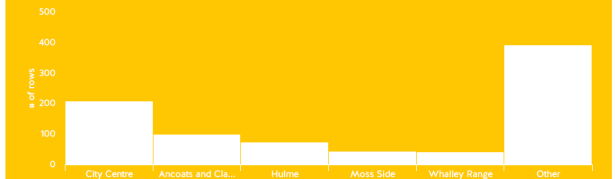
name

host_id

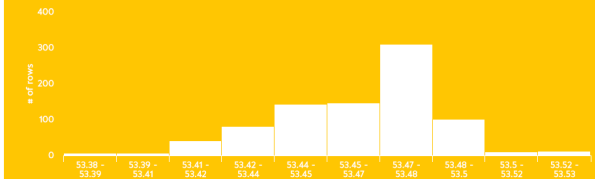


host_name

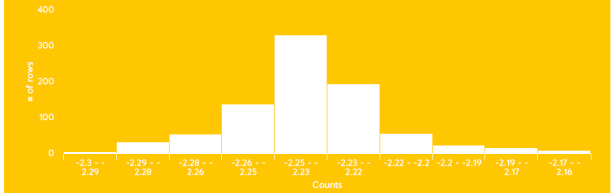
neighbourhood



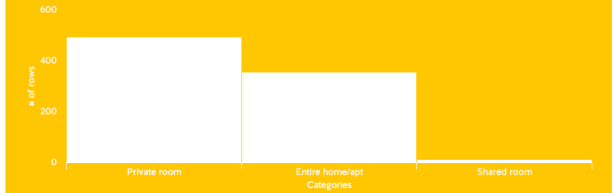
latitude



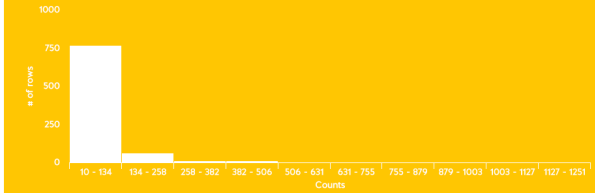
longitude



room_type



price



Manchester Listings (10 April, 2016)

jondinu@gmail.com

File Edit View Insert Format Data Tools Add-ons Help Last edit was 2 minutes ago

Comments Share

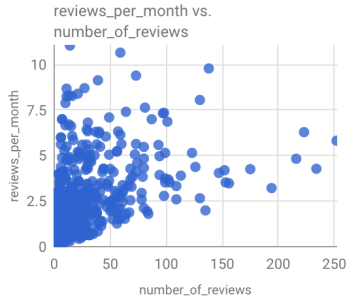
Arial 10

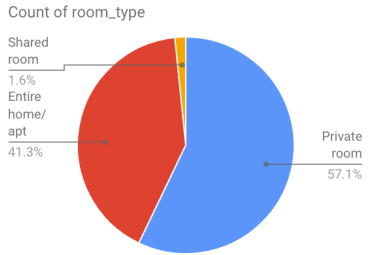
fx id

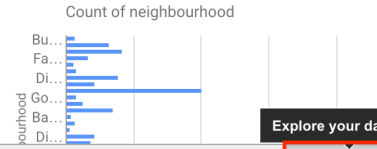
	A	B	C	D	E	F	G	H	I	J	K
1	id	name	host_id	host_name	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_
2	2613909	Comfy family hor	3962880	Soraya & Shahin	Burnage	53.4354773	-2.198658585	Private room	30	1	
3	753374	Comfy Family Hc	3962880	Soraya & Shahin	Burnage	53.43564167	-2.198358977	Private room	20	1	
4	8693211	Very comfortable	45641916	Leah Olwen	Burnage	53.42621941	-2.206232793	Private room	34	1	
5	2926014	Nice Double Ens	14940107	Ellie	Burnage	53.42115937	-2.212175841	Private room	30	1	
6	10379829	3 bedroom hous	16697099	Mo	Burnage	53.43120164	-2.206739811	Entire home/apt	351	1	
7	5270940	Beautiful propert	27284229	Abdul	Burnage	53.43369498	-2.20753078	Entire home/apt	400	1	
8	10606049	Large double be	48077465	Gaz	Burnage	53.41889579	-2.216246766	Private room	26	1	
9	7494589	Immaculate flat c	39211790	Andrew	Chorlton	53.44220345	-2.290031423	Entire home/apt	69	1	
10	6052900	Double room in c	2686669	Stuart	Chorlton	53.44007272	-2.276661646	Private room	25	1	
11	9425292	Lovely, modern 3	48706766	Jeorge	Chorlton	53.44436009	-2.278479147	Entire home/apt	105	1	
12	917498	Attic single in Wh	4917842	Gwyn	Chorlton	53.43940274	-2.282049331	Private room	36	3	
13	917497	Single in WhiteR	4917842	Gwyn	Chorlton	53.44010144	-2.2818676	Private room	36	3	
14	6419837	Double room in V	33492379	Michael	Chorlton	53.45039889	-2.271582299	Private room	20	2	
15	6574274	Chorlton-Cum-H	34380772	Jeff	Chorlton	53.43912789	-2.280734549	Private room	30	2	
16	1428087	Spacious double	7680218	Carolyn	Chorlton	53.44251005	-2.277413581	Private room	35	1	
17	11469427	Large Flat in Tre	60341399	Umar	Chorlton	53.44149627	-2.286462733	Entire home/apt	55	1	
18	6768991	Spacious resider	11056694	Paul	Chorlton	53.43724263	-2.274793183	Entire home/apt	119	3	
19	3246596	The Dog House (	12831230	Joey	Chorlton	53.43811675	-2.278654272	Private room	29	1	
20	5072185	Friendly Chorltor	26193398	Jonathon	Chorlton	53.44129309	-2.286314291	Private room	35	4	
21	10303617	Big Comfy Room	53013774	Frances	Chorlton	53.44506414	-2.284327336	Private room	39	1	
22	11349534	Lovely convertec	26193398	Jonathon	Chorlton	53.44111954	-2.289150805	Private room	32	1	
23	5449233	Stunning large g	4949917	My-Places	Chorlton	53.443413	-2.276128797	Entire home/apt	800	1	
24	1636107	Warm Chorlton fl	8689531	Elli	Chorlton	53.44172072	-2.28278488	Private room	25	1	
25	4707259	Trafford Double I	24276634	Imani	Chorlton	53.4497104	-2.27521545	Private room	50	1	
26	12251056	Large 1 Bed Bas	60341399	Umar	Chorlton	53.44071149	-2.286312608	Entire home/apt	50	1	
27	5121973	Peaceful single c	4917842	Gwyn	Chorlton	53.4398122	-2.283437887	Private room	36	3	
28	10544803	Spacious, cosy h	11915151	David	Chorlton	53.44522339	-2.282501859	Entire home/apt	100	2	
29	9365920	Nice double roor	48597897	Teresa	Chorlton	53.43396972	-2.282522113	Private room	35	1	
30	7001980	Ideally located C	26193398	Jonathon	Chorlton	53.44289952	-2.28648632	Private room	68	1	

manchester_04_10_2016

Explore







Explore your data

Automated Charts → Explore

Ranking EDA

- Most correlated columns
- Graph of columns with least missing values
- Least dispersion (variance) in histogram
- Best chart given data types (categorical, continuous vs. continuous, discrete vs. continuous, etc.)

Minimize TTU

Minimize Time To Understanding

Use the right tool for the job

EDA for Validation

- How many records in total are there?
- How many missing (or null) values are there?
- How many unique values does each column contain?
- What are the most common values for each column?
- For numeric columns, what are the summary statistics?
- How are the values of each column distributed?

Data Quality Checks

- How many records in total are there?
- How many missing (or null) values are there?
- How many unique values does each column contain?
- What are the most common values for each column?
- For numeric columns, what are the summary statistics?
- How are the values of each column distributed?

Defensive Data Analysis

"Always assume your data is hostile…"

- Jonathan Dinu

Defensive Data Analysis

"Always assume your data is hostile…"

- Jonathan Dinu (talking to himse

Incomplete Data

	Remove Row	Remove Column	Default Fill	Infer Fill (Interpolation)
Uniform Shape? (consistent Schema)	Yes	No	Yes	Yes
Preserve Record Count?	No	Yes	Yes	Yes
Consistent Values?	Yes	Yes	No	Depends

No Silver Bullet (i.e. no column is all green), have to consider tradeoffs

When to Fix

On Read

- Less storage overhead
- More flexible
- Uses more time/computation
- Less consistent/persistent
- Ex: Consume API as input to algorithm, fixing on the fly

On Write

- Always makes a copy
- Uses more storage
- Need to remember lineage
- Ex: Parse data during ETL to store in a data warehouse

When Types Matter

- Statically typed programming language (e.g C++, Java, etc.)
- Storage classes or types (e.g PostgreSQL, MySQL, MongoDB, etc.)
- Library types (e.g. numpy, pandas, etc.)
- Statistical analyses and ML (e.g. hypothesis tests, regression vs. classification, etc.)

Data Types

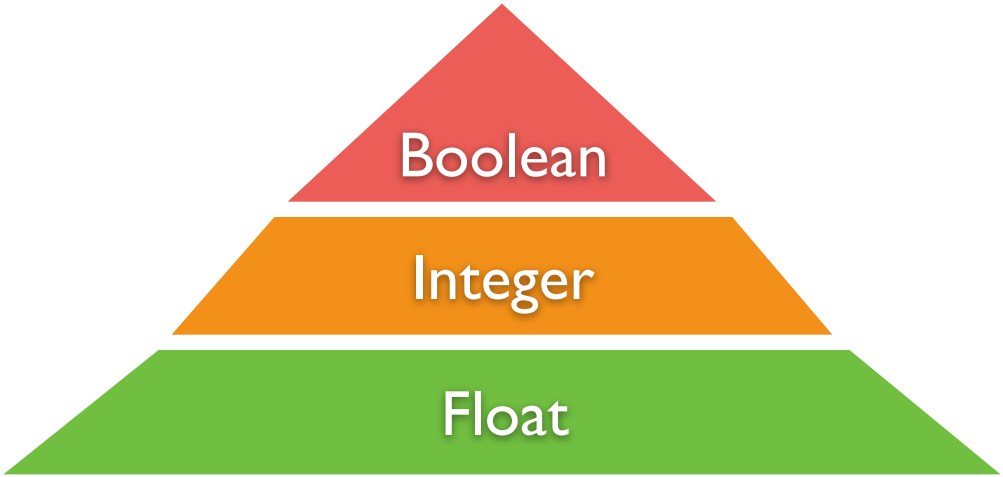
Discrete

- integer
- date
- varchar
- text
- uuid
- boolean
- NULL*

Continuous

- real
- float
- decimal/numeric
- timestamp/time

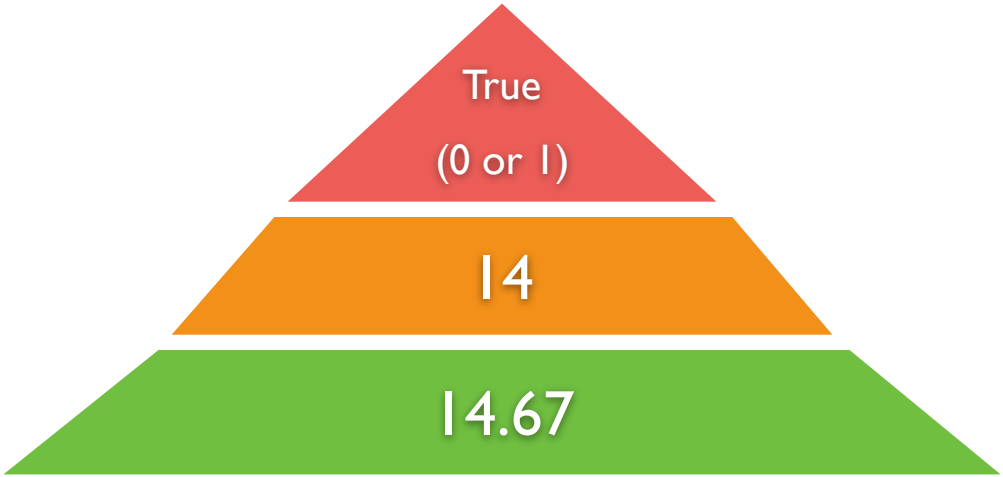
Type Hierarchy



Less precise

More precise

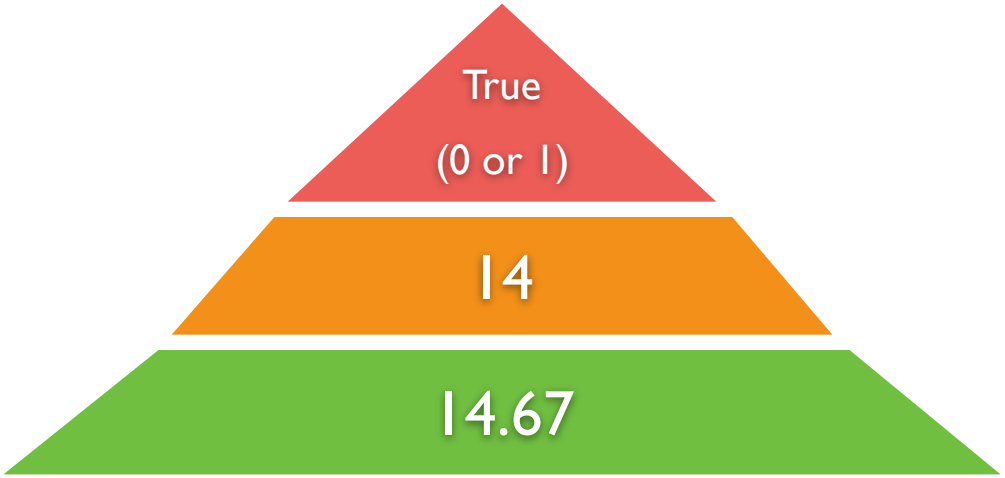
Type Hierarchy



Less precise

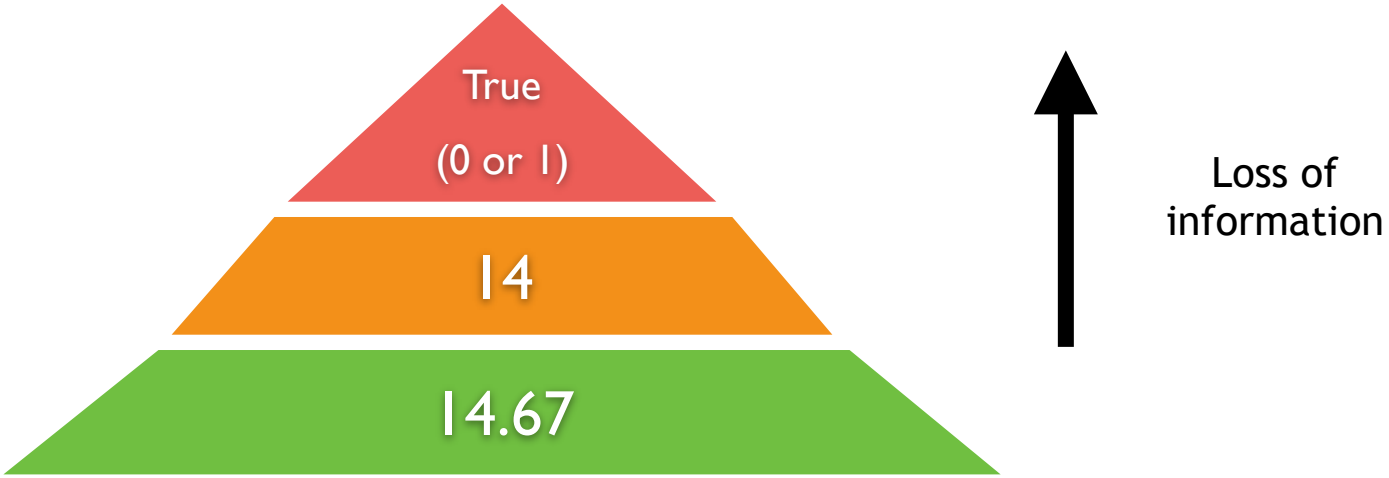
More precise

Type Casting and Coercion



"Perfect"
Conversion

Type Casting and Coercion



Type Casting and Coercion

```
float_var = 14.67

int_var = int(float_var)
# => 14

boolean_var = bool(int_var)
# => True

going_down = float(int_var)
# => 14.0

bool_to_int = int(boolean_var)
# => 1
```

Invalid Types

```
string_var = "hello"

In [20]: bool(string_var)
# => True

int(string_var)
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-21-52a2af8f25ef> in <module>
----> 1 int(string_var)

ValueError: invalid literal for int() with base 10: 'hello'
```

EDA for Insight

- Sort by column
- What are the summary statistics (min/max, mean, median, standard deviation)?
- How many unique values does each column contain?
- What are the most common values for each column?

EDA for Insight

Continuous

Discrete

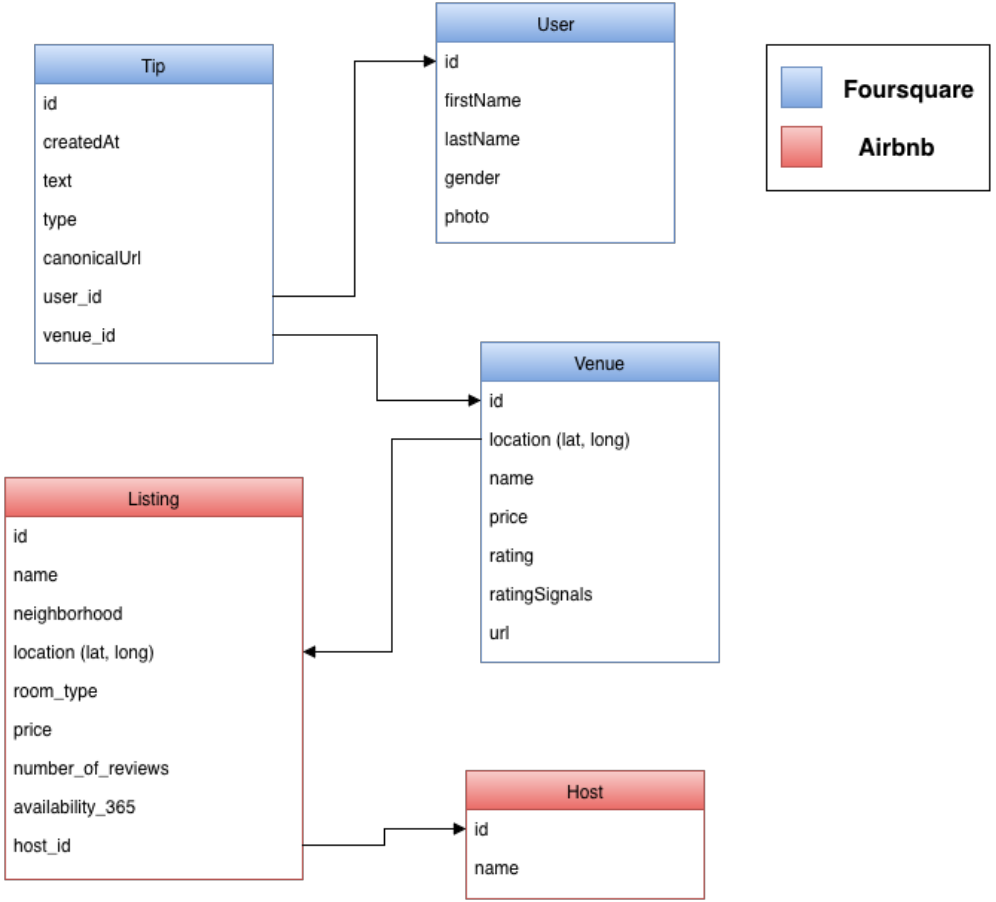
- Sort by column
- What are the summary statistics (min/max, mean, median, standard deviation)?
- How many unique values does each column contain?
- What are the most common values for each column?

For now, we are interested not in listings themselves but the context around the listing

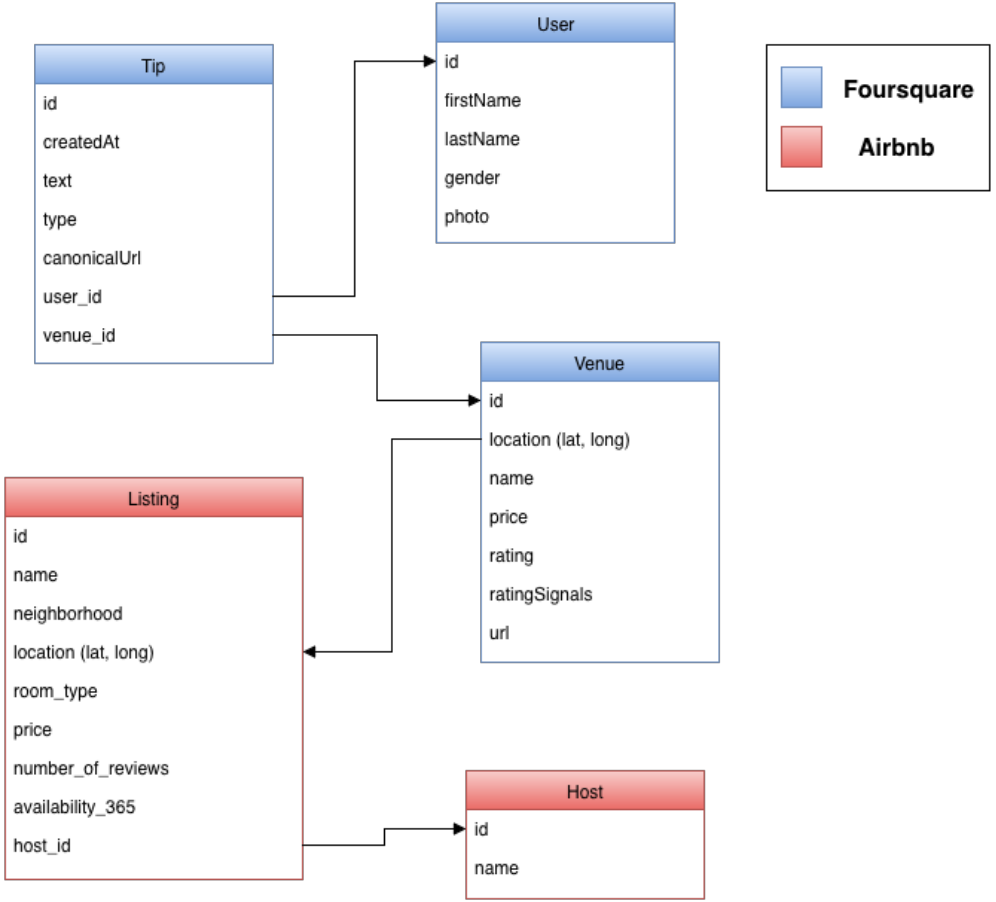
Which neighborhoods are the most explorable?

Which listings have the most lively and active POI around them?

Relations



Linking Data



And to implement this for the blue boxes…

Joins

Tip Table

id	text	type	venue_id	user_id
1	...	...	1	1

Venue Table

id	name	price	rating	rating Signals	url	lat	long
1	cafe	...	...	...	...	...	...
2	bar	...	...	...	...	...	...

User Table

id	firstName	lastName	gender	photo
1	Jessica	...	...	...

Joins

Tip Table

id	text	type	venue_id	user_id
1	...	...	1	1

Venue Table

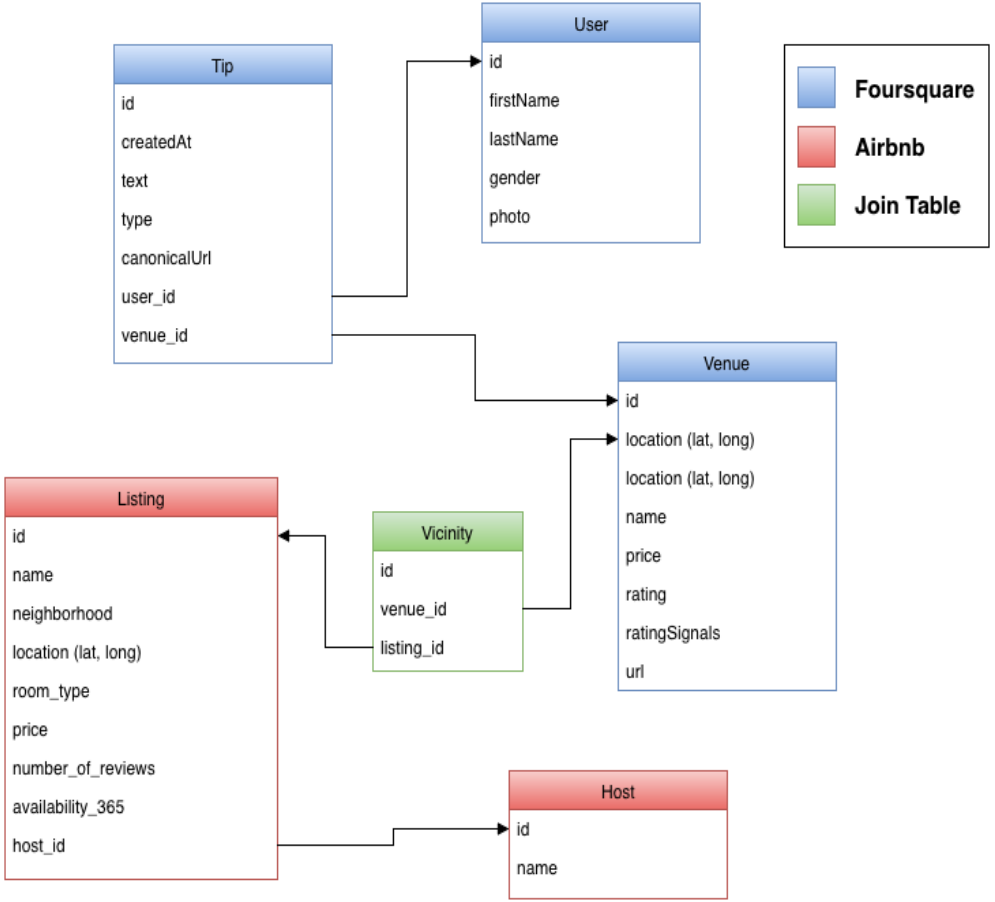
id	name	price	rating	rating Signals	url	lat	long
1	cafe	...	...	...	...	...	...
2	bar	...	...	...	...	...	...

User Table

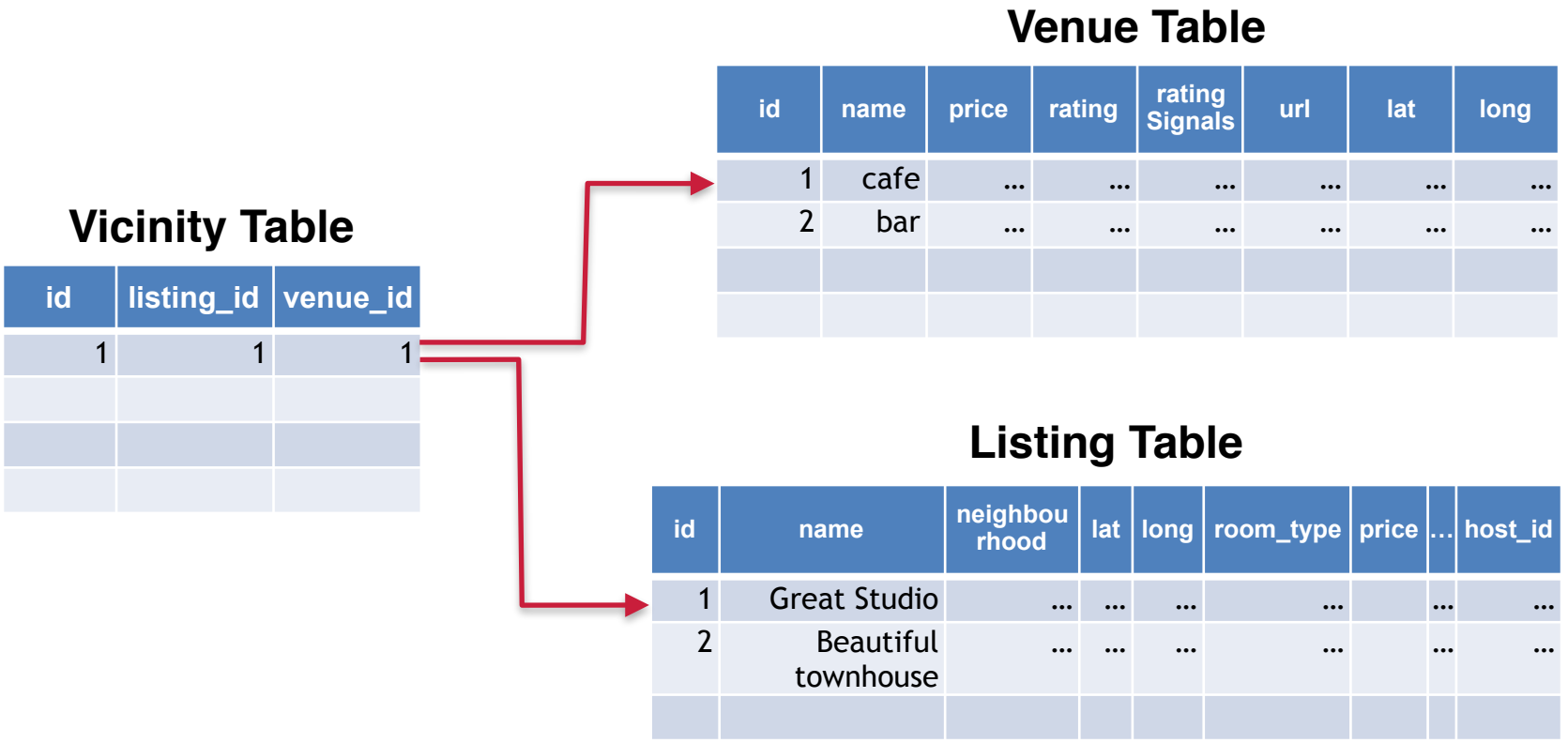
id	firstName	lastName	gender	photo
1	Jessica	...	...	...

But to link Listing and Venue we need a little extra…

Join Table



Join Tables



We have covered both:

- General concepts (ideas) that apply to ALL relational databases
- Specific syntaxes (implementation) of interfacing and querying a database with Python (and `peewee`)

Translating to SQL

peewee

```
Listing.select().limit(5)
```

SQLite3

```
SELECT * FROM listing LIMIT 5;
```

Translating to SQL

peewee

```
Listing.select(Listing.name)
       .limit(5)
```

SQLite3

```
SELECT listing.name FROM listing
  LIMIT 5;
```

```
SELECT name FROM listing
  LIMIT 5;
```

Translating to SQL

peewee

```
Listing.select()
       .order_by(fn.Random())
       .limit(5)
```

SQLite3

```
SELECT * FROM listing
  ORDER BY RANDOM()
  LIMIT 5;
```

Translating to SQL

peewee

```
Listing.select().count()
```

SQLite3

```
SELECT COUNT(*) FROM listing;
```

SQL Order of Operations

```
SELECT DISTINCT column, AGG_FUNC(column_or_expression), …
FROM mytable
    JOIN another_table
      ON mytable.column = another_table.column
    WHERE constraint_expression
    GROUP BY column
    HAVING constraint_expression
    ORDER BY column ASC/DESC
    LIMIT count OFFSET COUNT;
```

source: http://sqlbolt.com/lesson/select_queries_order_of_execution

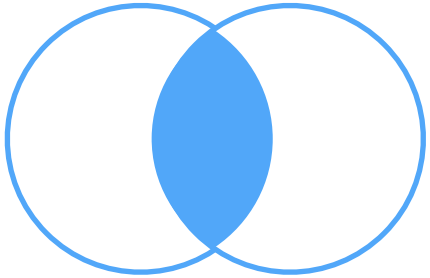
Join Operations

Like Set operations:

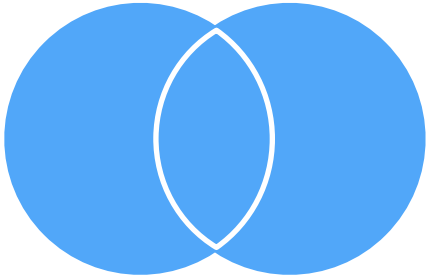
- Union
- Intersection
- Difference (or complements)
- Cartesian Product

Set Operations

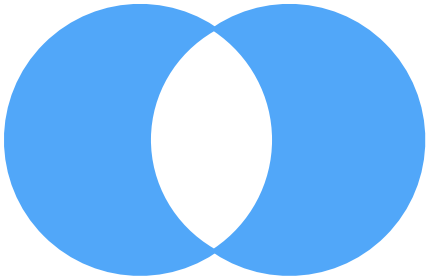
Intersection



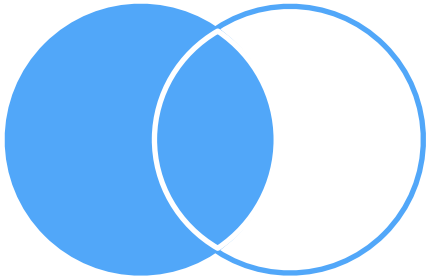
Union



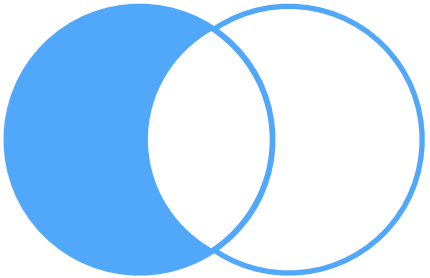
Symmetric Difference



Intersecting Complement



Difference



Set Operations (in Python)

```
a = set([1,2,3])
b = set([3,4,5])
```

Intersection

```
>>> a & b
{3}
```

Union

```
>>> a | b
{1, 2, 3, 4, 5}
```

Symmetric Difference

```
>>> a ^ b
{1, 2, 4, 5}
```

Intersecting Complement

```
>>> a | (a & b)
{1, 2, 3}
```

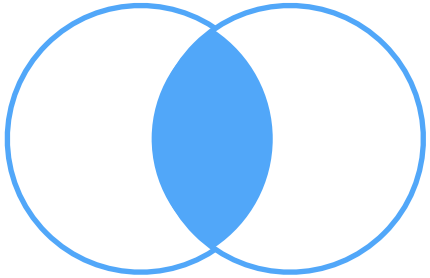
Difference

```
>>> a - b
{1, 2}
```

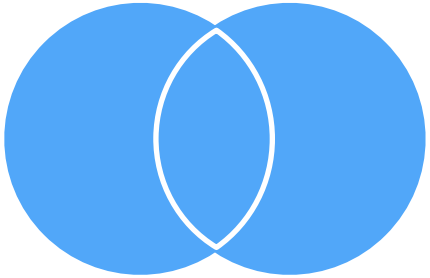
https://docs.python.org/3/library/stdtypes.html#set

Join Operations

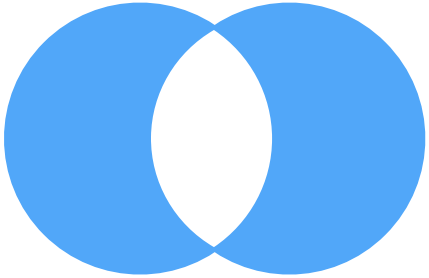
Inner Join



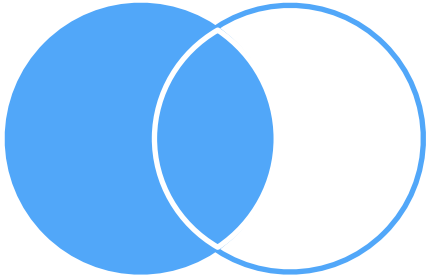
Full Outer Join



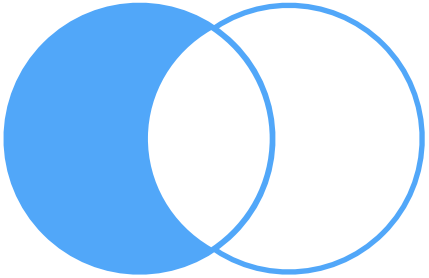
Double Exclusion



Left Outer Join



Single Exclusion



Join Operations (with peewee)

Inner Join

```
A.select(A, B) \
 .join(B, \
   on=A.value == B.value)

>>> [(3, 3, 1, 3)]
```

a

id	value
1	1
2	2
3	3

b

id	value
1	3
2	4
3	5

Full Outer Join
(not in sqlite3)

```
A.select(A, B) \
 .join(B, JOIN.FULL, \
       on=A.value == B.value)

>>> [(1, 1, None, None),
       (2, 2, None, None),
       (3, 3, 1, 3),
       (None, None, 2, 4),
       (None, None, 3, 5)]
```

Double Exclusion
(not in sqlite3)

```
A.select(A, B) \
 .join(B, JOIN.FULL, \
       on=A.value == B.value) \
 .where((A.id.is_null(True)) \
        | (B.id.is_null(True)))

>>> [(1, 1, None, None),
     (2, 2, None, None),
     (None, None, 2, 4),
     (None, None, 3, 5)]
```

Left Outer Join

```
A.select(A, B) \
 .join(B, JOIN.LEFT_OUTER, \
       on=A.value == B.value)

>>> [(1, 1, None, None),
       (2, 2, None, None),
       (3, 3, 1, 3)]
```

Single Exclusion

```
A.select(A, B)
 .join(B, JOIN.LEFT_OUTER,
       on=A.value == B.value)
 .where(B.id.is_null(True))

>>> [(1, 1, None, None),
       (2, 2, None, None)]
```

Join Operations (in SQL)

a

id	value
1	1
2	2
3	3

b

id	value
1	3
2	4
3	5

Inner Join

```
SELECT * FROM a
  JOIN b
    ON a.value = b.value;
```

id	value	id	value
3	3	1	3

Full Outer Join (not in sqlite3)

```
SELECT * FROM a
FULL OUTER JOIN b
ON a.value = b.value;
```

id	value	id	value
1	1	null	null
2	2	null	null
3	3	1	3
null	null	2	4
null	null	3	5

Double Exclusion (not in sqlite3)

```
SELECT * FROM a
FULL OUTER JOIN b
ON a.value = b.value
WHERE a.id IS null
OR b.id IS null;
```

id	value	id	value
1	1	null	null
2	2	null	null
null	null	2	4
null	null	3	5

Left Outer Join

```
SELECT * FROM a
LEFT OUTER JOIN b
ON a.value = b.value;
```

id	value	id	value
1	1	null	null
2	2	null	null
3	3	1	3

Single Exclusion

```
SELECT * FROM a
LEFT OUTER JOIN b
ON a.value = b.value
WHERE b.id IS null;
```

id	value	id	value
1	1	null	null
2	2	null	null

Learning the syntax of SQL is a matter of practice (and patience)

SQLBolt
Learn SQL with simple, interactive exercises.

Introduction to SQL

Welcome to SQLBolt, a series of interactive lessons and exercises designed to help you quickly learn SQL right in your browser.

What is SQL?

SQL, or Structured Query Language, is a language designed to allow both technical and non-technical users query, manipulate, and transform data from a relational database. And due to its simplicity, SQL databases provide safe and scalable storage for millions of websites and mobile applications.

> Did you know?
>
> There are many popular SQL databases including SQLite, MySQL, Postgres, Oracle and Microsoft SQL Server. All of them support the common SQL language standard, which is what this site will be teaching, but each implementation can differ in the additional features and storage types it supports.

Relational databases

Before learning the SQL syntax, it's important to have a model for what a relational database actually is. A relational database represents a collection of related (two-dimensional) tables. Each of the tables are similar to an Excel spreadsheet, with a fixed number of named columns (the attributes or properties of the table) and any number of rows of data.

For example, if the Department of Motor Vehicles had a database, you might find a table containing all the known vehicles that people in the state are driving. This table might need to store the model name, type, number of wheels, and number of doors of each vehicle for example.

SQL Resources

- Learn SQL the Hard Way (online book)
- SQLZOO (interactive tutorial)
- Khan Academy: SQL (interactive tutorial)
- Codecademy: SQL (interactive tutorial)
- Schemaverse (interactive game)
- GalaXQL (interactive game)
- SQL Island (interactive game)
- Udacity: Intro to Relational Databases (online course)