Data

and

Databases

Outline

- The types of things you find in data, and their encoding
- The representation of different objects as arrays, and their encoding
- How does one store data?

Types in Data

Tables

Data comes in many ways, but most often it comes in tables. These might be tables in relational databases, Pandas dataframes, csv files on disk, and excel spreadsheets.

Items

item index	quality	type	price
0	High	Toy	20
1	High	Book	5
2	Medium	Craft	12
3	Low	Craft	10

Customers

customer index	item index	quantity	total
0	0	5	100
1	0	3	60
2	2	4	48
3	3	10	100

Types

Ordinal

Nominal/Categorical

Quantitative

Items

item index	quality	type	price
0	High	Toy	20
1	High	Book	5
2	Medium	Craft	12
3	Low	Craft	10

Customers

"Foreign Key"

customer index	item index	quantity	total
0	0	5	100
1	0	3	60
2	2	4	48
3	3	10	100

Data has "type". It might be numbers, words, characters, true/false's. And it may be missing, which is a `None`, or a "NA" or a `NULL`.

Sometimes data refers to other tables, using a "Foreign Key", such as a reference to another sheet in a spreadsheet, or an id refering to a column in another database table.

Often you must **encode** the data in some form to be useful. For example, you might use integers to encode the ordinal column "quality" here.

Types: more detail

Think about the types here!

Table: contributors

New Record | Delete Record

	id	last_name	first_name	middle_name	street_1	street_2	city	state	zip	amount	date	candidate_id
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Agee	Steven	NULL	549 Laurel ...	NULL	Floyd	VA	24091	500	2007-06-30	16
2	5	Akin	Charles	NULL	10187 Suga...	NULL	Bentonville	AR	72712	100	2007-06-16	16
3	6	Akin	Mike	NULL	181 Baywo...	NULL	Monticello	AR	71655	1500	2007-05-18	16
4	7	Akin	Rebecca	NULL	181 Baywo...	NULL	Monticello	AR	71655	500	2007-05-18	16
5	8	Aldridge	Brittni	NULL	808 Capitol...	NULL	Washington	DC	20024	250	2007-06-06	16
6	9	Allen	John D.	NULL	1052 Cann...	NULL	North Augu...	SC	29860	1000	2007-06-11	16
7	10	Allen	John D.	NULL	1052 Cann...	NULL	North Augu...	SC	29860	1300	2007-06-29	16
8	11	Allison	John W.	NULL	P.O. Box 10...	NULL	Conway	AR	72033	1000	2007-05-18	16
9	12	Allison	Rebecca	NULL	3206 Sum...	NULL	Little Rock	AR	72227	1000	2007-04-25	16

Encoding types (example SQL)

Data type	Description
CHARACTER(n)	Character string. Fixed-length n
VARCHAR(n) or CHARACTER VARYING(n)	Character string. Variable length. Maximum length n
BINARY(n)	Binary string. Fixed-length n
BOOLEAN	Stores TRUE or FALSE values
VARBINARY(n) or BINARY VARYING(n)	Binary string. Variable length. Maximum length n
INTEGER(p)	Integer numerical (no decimal). Precision p
FLOAT(p)	Approximate numerical, mantissa precision p.
DATE	Stores year, month, and day values
TIME	Stores hour, minute, and second values
TIMESTAMP	Stores year, month, day, hour, minute, and second values
INTERVAL	Rrepresenting a period of time, depending on the type of interval

Typing our tables in SQL

```sql
DROP TABLE IF EXISTS "candidates";
DROP TABLE IF EXISTS "contributors";

CREATE TABLE "candidates" (
    "id" INTEGER PRIMARY KEY  NOT NULL ,
    "first_name" VARCHAR,
    "last_name" VARCHAR,
    "middle_name" VARCHAR,
    "party" VARCHAR NOT NULL
);
```

```sql
CREATE TABLE "contributors" (
    "id" INTEGER PRIMARY KEY  AUTOINCREMENT  NOT NULL,
    "last_name" VARCHAR,
    "first_name" VARCHAR,
    "middle_name" VARCHAR,
    "street_1" VARCHAR,
    "street_2" VARCHAR,
    "city" VARCHAR,
    "state" VARCHAR,
    "zip" VARCHAR,
    "amount" INTEGER,
    "date" DATETIME,
    "candidate_id" INTEGER NOT NULL,
    FOREIGN KEY(candidate_id) REFERENCES candidates(id)
);
```

Typing in Pandas with Encoding

```
>>>dfcand.dtypes

id                int64
first_name       object
last_name        object
middle_name      object
party            object
dtype: object
```

We can convert the party to Categoricals

```
>>>dfcand['party'] = dfcand.party.
    astype('category')
```

```
id                int64
first_name       object
last_name        object
middle_name      object
party          category
dtype: object

>>>dfcand.party.cat.categories

Index(['D', 'R'], dtype='object')

>>>dfcand.party.cat.ordered

False
```

More Control and Ordinals

62	BURKE	SUZANNE	M.	3401 EVANSTON	NaN	SEATTLE	WA	981038677	-700.0	2008-03-05	22
100	Aaronson	Rebecca	NaN	2000 Village Green Dr Apt 12	NaN	Mill Creek	WA	980125787	100.0	2008-02-08	34
106	Aaronson	Rebecca	NaN	2000 Village Green Dr Apt 12	NaN	Mill Creek	WA	980125787	100.0	2008-02-14	34
0	Agee	Steven	NaN	549 Laurel Branch Road	NaN	Floyd	VA	24091	500.0	2007-06-30	16

```
states = ["AL", "AK", "AZ", "AR", "CA", ...]
from pandas.api.types import CategoricalDtype
dfcwci['state'] = dfcwci.state.astype(
    CategoricalDtype(states))
```

In this way incomplete categorical levels are handled.

For ordinals we can do (states are not really ordinal, but think of ratings and grades):

```
dfcwci['state'] = dfcwci.state.astype(
    CategoricalDtype(states, ordered=True))
dfcwci.sort_values(by='state', ascending=False)
```

Representation of Data

	id	first_name	last_name	middle_name	party
	Filter	Filter	Filter	Filter	Filter
1	16	Mike	Huckabee		R
2	20	Barack	Obama		D
3	22	Rudolph	Giuliani		R
4	24	Mike	Gravel		D
5	26	John	Edwards		D
6	29	Bill	Richardson		D
7	30	Duncan	Hunter		R
8	31	Dennis	Kucinich		D
9	32	Ron	Paul		R

Some data is tabular

And some data is an array of numbers

Images are arrays of numbers.

So are videos.

So is text.

From One Image

(height, width, color_depth)

To Many Images

```
(sample_size, height, width, color_depth)
```

1D TENSOR/
VECTOR
5
7
4 5
1 2
- 6
3
2 2
1
6
3
- 9
2D TENSOR /
MATRIX
- 9 4 2 5 7
3 0 1 2 8 6 1
1 2 3 - 6 4 5 2
2 2 3 - 1 7 2 6
One channel of an image
a document
3D TENSOR/
CUBE
An image with multiple channels.
Many 1-channel images
multiple documents
multiple images with more than one channel
one video consisting of multiple image frames
4D TENSOR
VECTOR OF CUBES
multiple videos
5D TENSOR
MATRIX OF CUBES

Text Data

We can store text data in a 3D tensor too. Let's take a look at tweets.

Tweets are max 280 characters. Twitter uses the UTF-8 standard, which allows for millions of types of characters, but lets focus on the first 128, which are basic ASCII.

Then one tweet is a 2D array of shape (280,128). What is in there? What if a tweet is shorter?

For multiple tweets we go to a 3D array/tensor of shape:

```
(number_of_tweets_captured, 280, 128)
```

(As an array this is a wasteful encoding, but there are better formats)

@realdonaldtrump tweet: Great. Most corrupt election in history, by far. We won!!!

Vocabulary

Text of Tweet

	a	b	c	d	e	f	g	h	I	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z		,	.	!
g	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
e	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Embeddings

Convert categorical data to a compressed numerical representation. Far less wasteful and more amenable to meaning.

Graph Data

- Has vertices and edges
- Example: Social connectivity graph, flight paths between airports with edge length being distance (for fuel optimization), computer networks, code graphs, web graph
- can be converted into tables or dealt with as is

Databases and Indexes

Relational Database

- Tables (in Pandas and relational databases) are relations
- A table is merely a collection of tuples
- Storage may be row (data point) wise or column (feature) wise.
- Read all or some rows matching some condition (a column subset/projection), written in a declarative language: SQL.
- SQL is declarative, the query optimizer decides how to make the query.
- tables refer to other tables using foreign keys. This is the *Normal Form*.

Document Database

Appropriate for apps with mostly one-to-many relationships (tree-structured data) or no relationships between "documents". JSON is a highly used format.

Notice that there is lesser common structure in jobs, than education. Finding commonalities is harder across jobs from different users.

```
{
  "user_id":     251,
  "first_name":  "Bill",
  "last_name":   "Gates",
  "summary":     "Co-chair of the Bill & Melinda Gates... Active blogger.",
  "region_id":   "us:91",
  "industry_id": 131,
  "photo_url":   "/p/7/000/253/05b/308dd6e.jpg",
  "positions": [
    {"job_title": "Co-chair", "organization": "Bill & Melinda Gates Foundation"},
    {"job_title": "Co-founder, Chairman", "organization": "Microsoft"}
  ],
  "education": [
    {"school_name": "Harvard University",       "start": 1973, "end": 1975},
    {"school_name": "Lakeside School, Seattle", "start": null, "end": null}
  ],
```

Graph or Relational?

- How do we handle many-to-many relationships?
- Relational model can handle simple cases
- As connections become more complex, model directly as a graph
- We have seen examples of graph data

What kind of data access do you need?

- **relational**: pandas, SQL: Postgres, sqlite, Hbase, VoltDB
- **document oriented**: MongoDB, CouchDB
- **key-value**: Riak, Redis, Memcached
- **graph oriented**: Neo4J

OLTP vs OLAP[4]

OLTP: Online Transaction processing: look up a few records by some key, using an index. Insertion or updating based on users clicks, input. Ecommerce!

OLAP: Queries for business intelligence. Join lots of tables, get information about an entity of interest, such as a user, or a product.

Typically, ETL (Extract-Transform-Load) scripts sct on OLTP databases to produce OLAP ones.

[4] From Designing Data Intendive Applications, Martin Kleppmann, O'reilly Publishers

Indexes

Basic Idea: subdividing ranges into known-sized "pages" allows for a B-tree structure with fast retrieval. Adding entries sometimes splits pages.

OLAP Queries

Not usually fast lookups. The index is typically used for large range queries, and then tables are joined together based on common keys.

The ultimate goal of OLAP systems is to output some subset of queried data, with no foreign keys, all in one wide table. This process is called de-normalization and is at the heart of data science pipelines.

dim_product table

product_sk	sku	description	brand	category
30	OK4012	Bananas	Freshmax	Fresh fruit
31	KA9511	Fish food	Aquatech	Pet supplies
32	AB1234	Croissant	Dealicious	Bakery

dim_store table

store_sk	state	city
1	WA	Seattle
2	CA	San Francisco
3	CA	Palo Alto

fact_sales table

date_key	product_sk	store_sk	promotion_sk	customer_sk	quantity	net_price	discount_price
140102	31	3	NULL	NULL	1	2.49	2.49
140102	69	5	19	NULL	3	14.99	9.99
140102	74	3	23	191	1	4.49	3.89
140102	33	8	NULL	235	4	0.99	0.99

dim_date table

date_key	year	month	day	weekday	is_holiday
140101	2014	jan	1	wed	yes
140102	2014	jan	2	thu	no
140103	2014	jan	3	fri	no

dim_customer table

customer_sk	name	date_of_birth
190	Alice	1979-03-29
191	Bob	1961-09-02
192	Cecil	1991-12-13

dim_promotion table

promotion_sk	name	ad_type	coupon_type
18	New Year sale	Poster	NULL
19	Aquarium deal	Direct mail	Leaflet
20	Coffee & cake bundle	In-store sign	NULL