Welcome to CS181 - Data Analytics and Visualization

Spring 2025

Prof. Gabe Hope

Warm-up discussion

- What is the goal of data visualization?

- **But first: Introductions!**
  - **Going around:** Introduce yourself with your name, year and major
  - *Optionally:* Is there something you are particularly excited to learn about in this class?

Course Logistics (website)

Why is effective visualization important?

January 26, 1986

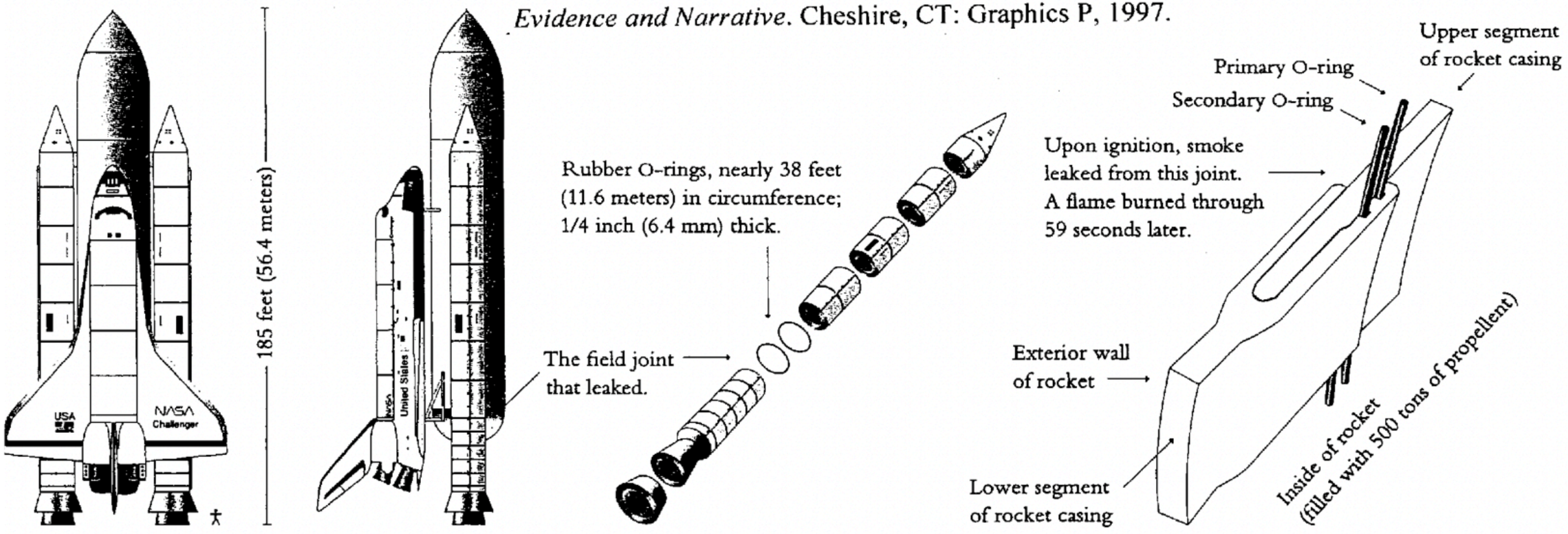
Challenger Mission

Crew of 7: Dick Scobee, Michael Smith, Ellison Onizuka, Judith Resnik, Ronald McNair, Gregory Jarvis, and Chirsta McAuliffe.

A cold launch day

Engineering concerns

From Tufte, Edward. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Cheshire, CT: Graphics P, 1997.



Last minute debate

CONCLUSIONS:

- TEMPERATURE OF O-RING IS NOT ONLY PARAMETER CONTROLLING BLOW-BY

  SRM 15 WITH BLOW-BY HAD AN O-RING TEMP AT 53°F
  SRM 22 WITH BLOW-BY HAD AN O-RING TEMP AT 75°F
  FOUR DEVELOPMENT MOTORS WITH NO BLOW-BY WERE TESTED AT O-RING TEMP OF 47° TO 52°F

  DEVELOPMENT MOTORS HAD PUTTY PACKING WHICH RESULTED IN BETTER PERFORMANCE

- AT ABOUT 50°F BLOW-BY COULD BE EXPERIENCED IN CASE JOINTS

- TEMP FOR SRM 25 ON 1-28-86 LAUNCH WILL BE 29°F 9AM
  38°F 2PM

- HAVE NO DATA THAT WOULD INDICATE SRM 25 IS DIFFERENT THAN SRM 15 OTHER THAN TEMP

RECOMMENDATIONS:

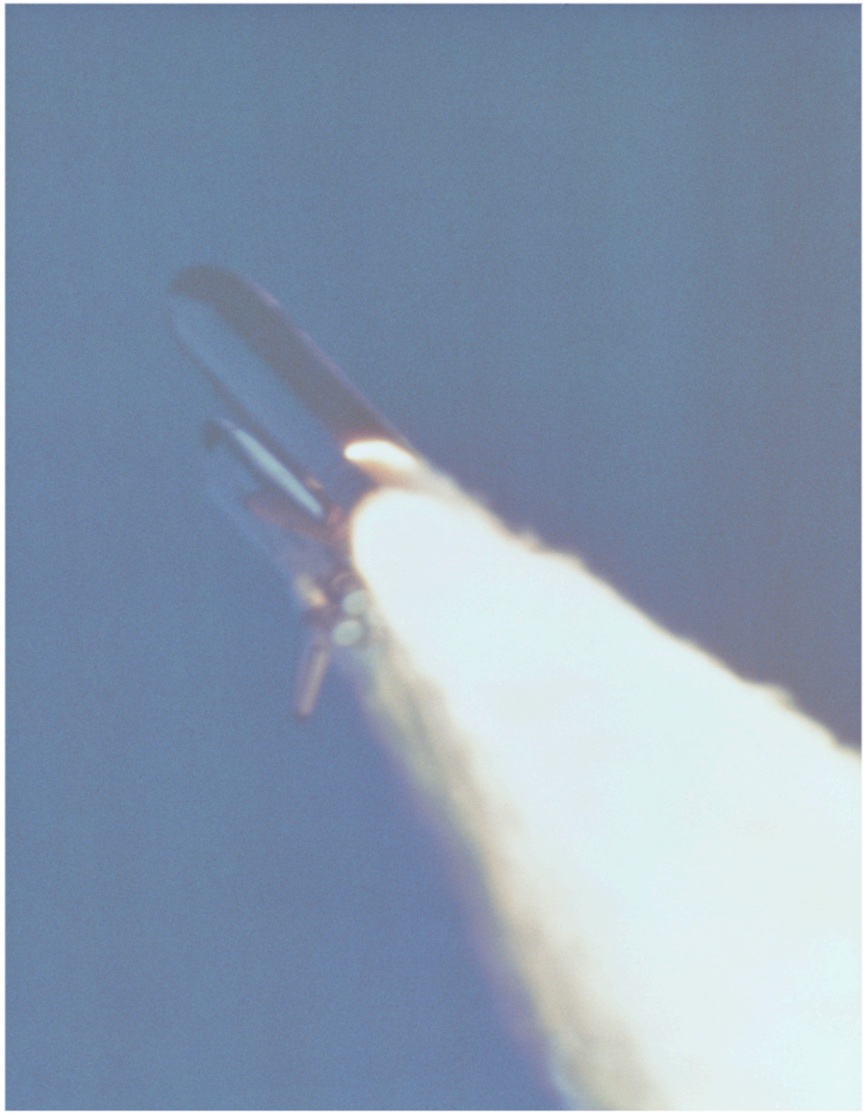
- O-RING TEMP MUST BE $\geq$ 53°F AT LAUNCH

  DEVELOPMENT MOTORS AT 47° TO 52°F WITH PUTTY PACKING HAD NO BLOW-BY
  SRM 15 (THE BEST SIMULATION) WORKED AT 53°F

- PROJECT AMBIENT CONDITIONS (TEMP & WIND) TO DETERMINE LAUNCH TIME

A predictable failure

Plume indicating an O-ring failure at launch



Tragedy

What went wrong?

Tufte's take

"…there was a clear proximate cause: an inability to assess the link between cool temperature and O-ring damage on earlier flights…"

*"…rocket engineers and managers needed a quick, smart **analysis** of evidence about the threat of cold to the O-rings, as well as an effective **presentation** of evidence in order to convince NASA offcials not to launch"*

- Edward Tufte, Visual Explanations

Morton Thiokol Charts

VISUAL EXPLANATIONS - EDWARD TUFTE

HISTORY OF O-RING DAMAGE ON SRM FIELD JOINTS

	SRM No.	Cross Sectional View: Erosion Depth (in.)	Cross Sectional View: Perimeter Affected (deg)	Cross Sectional View: Nominal Dia. (in.)	Top View: Length Of Max Erosion (in.)	Top View: Total Heat Affected Length (in.)	Clocking Location (deg)
61A LH Center Field**	22A	None	None	0.280	None	None	36°-66°
61A LH ~~CENTER~~ FIELD**	22A	NONE	NONE	0.280	NONE	NONE	338°-18°
51C LH Forward Field**	15A	0.010	154.0	0.280	4.25	5.25	163
51C RH Center Field (prim)***	15B	0.038	130.0	0.280	12.50	58.75	354
51C RH Center Field (sec)***	15B	None	45.0	0.280	None	29.50	354
41D RH Forward Field	13B	0.028	110.0	0.280	3.00	None	275
41C LH Aft Field*	11A	None	None	0.280	None	None	--
41B LH Forward Field	10A	0.040	217.0	0.280	3.00	14.50	351
STS-2 RH Aft Field	2B	0.053	116.0	0.280	--	--	90

*Hot gas path detected in putty. Indication of heat on O-ring, but no damage.
**Soot behind primary O-ring.
***Soot behind primary O-ring, heat affected secondary O-ring.

Clocking location of leak check port - 0 deg.

OTHER SRM-15 FIELD JOINTS HAD NO BLOWHOLES IN PUTTY AND NO SOOT NEAR OR BEYOND THE PRIMARY O-RING.

SRM-22 FORWARD FIELD JOINT HAD PUTTY PATH TO PRIMARY O-RING, BUT NO O-RING EROSION AND NO SOOT BLOWBY. OTHER SRM-22 FIELD JOINTS HAD NO BLOWHOLES IN PUTTY.

Blow By History

SRM-15 worst Blow-By
- 2 case joints (80°), (110°) Arc
- Much worse visually than SRM-22

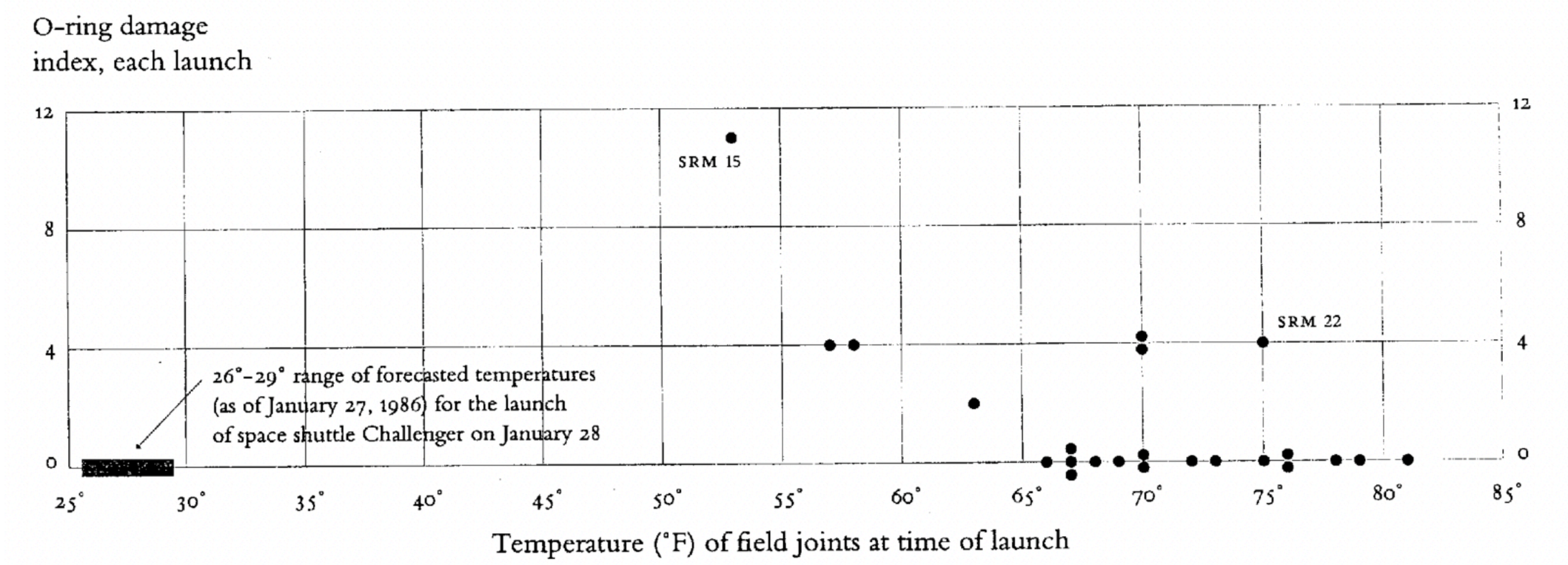
SRM 22 Blow-By
- 2 case joints (30-40°)

SRM-13A, 15, 16A, 18, 23A 24A
- Nozzle Blow-By

History of O-Ring Temperatures (Degrees-F)

Motor	MBT	Amb	O-Ring	Wind
DM-1	68	36	47	10 mph
DM-2	76	45	52	10 mph
QM-3	72.5	40	48	10 mph
QM-4	76	48	51	10 mph
SRM-15	52	64	53	10 mph
SRM-22	77	78	75	10 mph
SRM-25	55	26	29 27	10 mph 25 mph

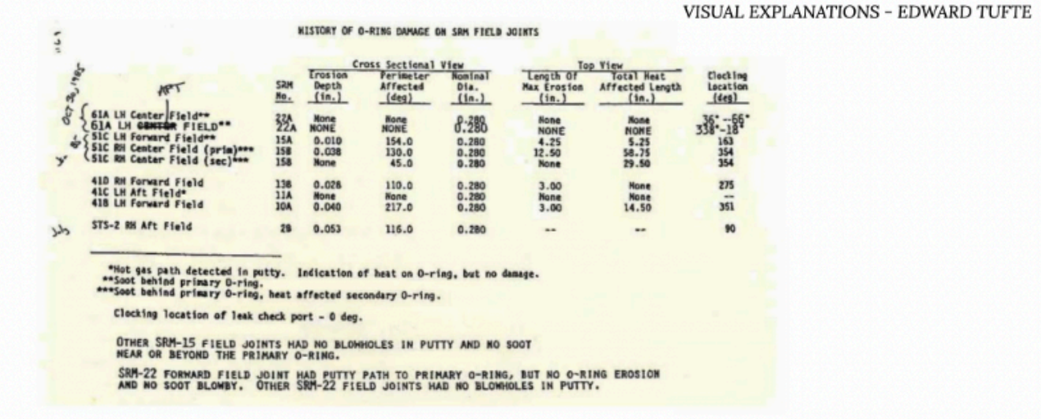
Tufte's version



Why create visualizations?

Make informed descisions

The challenger disaster shows the importance of this



VISUAL EXPLANATIONS - EDWARD TUFTE

HISTORY OF O-RING DAMAGE ON SRM FIELD JOINTS

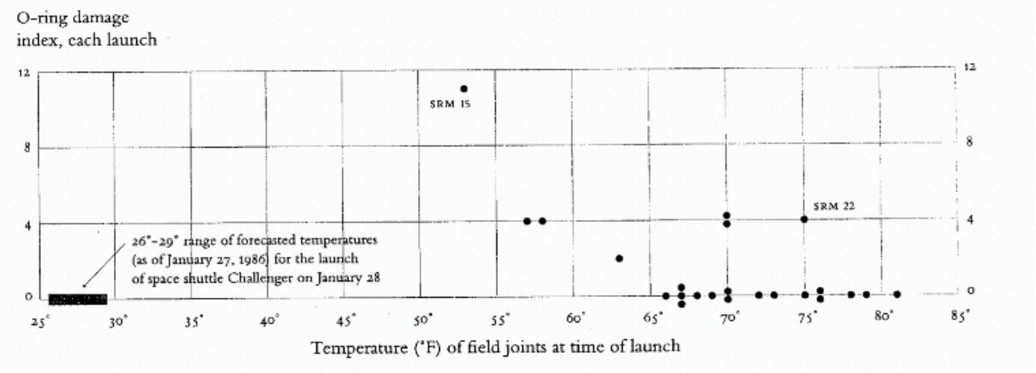
	SRM No.	Cross Sectional View: Erosion Depth (in.)	Cross Sectional View: Perimeter Affected (deg)	Cross Sectional View: Nominal Dia. (in.)	Top View: Length Of Max Erosion (in.)	Top View: Total Heat Affected Length (in.)	Clocking Location (deg)
61A LH Center Field**	22A	None	None	0.280	None	None	36°-66°
61A LH CENTER FIELD**	22A	NONE	NONE	0.280	NONE	NONE	338°-18°
51C LH Forward Field**	15A	0.010	154.0	0.280	4.25	5.25	163
51C RH Center Field (prim)***	15B	0.038	130.0	0.280	12.50	58.75	354
51C RH Center Field (sec)***	15B	None	45.0	0.280	None	29.50	354
41D RH Forward Field	13B	0.028	110.0	0.280	3.00	None	275
41C LH Aft Field*	11A	None	None	0.280	None	None	--
41B LH Forward Field	10A	0.040	217.0	0.280	3.00	14.50	351
STS-2 RH Aft Field	2B	0.053	116.0	0.280	--	--	90

*Hot gas path detected in putty. Indication of heat on O-ring, but no damage.
**Soot behind primary O-ring.
***Soot behind primary O-ring, heat affected secondary O-ring.

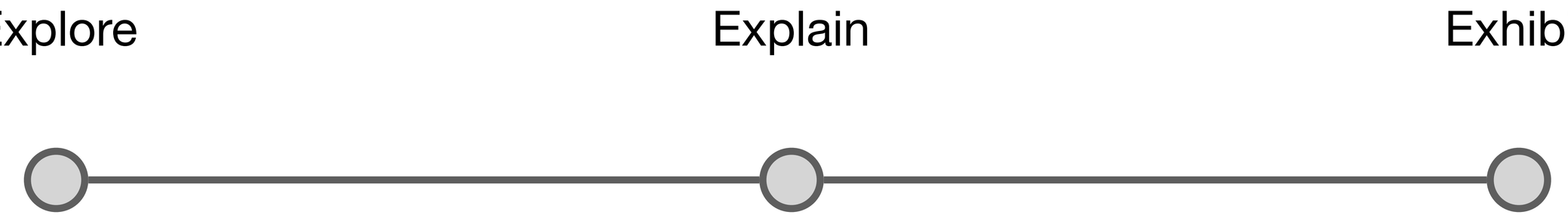
Clocking location of leak check port - 0 deg.

OTHER SRM-15 FIELD JOINTS HAD NO BLOWHOLES IN PUTTY AND NO SOOT NEAR OR BEYOND THE PRIMARY O-RING.

SRM-22 FORWARD FIELD JOINT HAD PUTTY PATH TO PRIMARY O-RING, BUT NO O-RING EROSION AND NO SOOT BLOWBY. OTHER SRM-22 FIELD JOINTS HAD NO BLOWHOLES IN PUTTY.

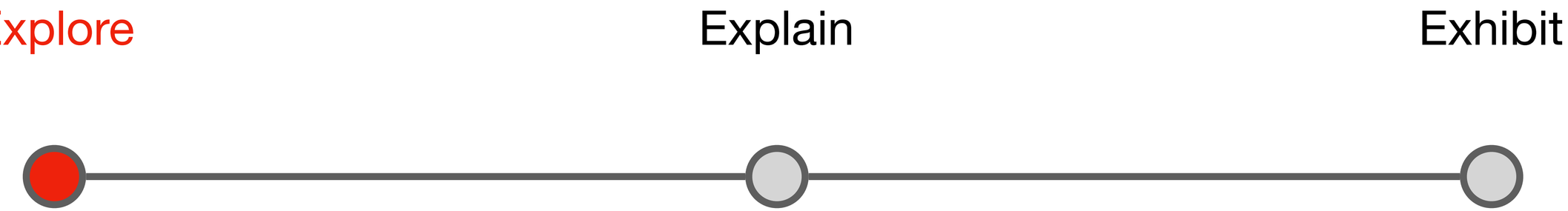


Spectrum of visualization goals



Credit: Christian Swinehart

Spectrum of visualization goals



Credit: Christian Swinehart

Exploration - Understand data

Anscombe's quartet

Dataset I		Dataset II		Dataset III		Dataset IV	
x	*y*	*x*	*y*	*x*	*y*	*x*	*y*
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

What are the differences between these 4 datasets?

Exploration - Understand data

Anscombe's quartet

Dataset I		Dataset II		Dataset III		Dataset IV	
x	*y*	*x*	*y*	*x*	*y*	*x*	*y*
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

We could try using some basic tools from statistics to answer this question. *e.g. means, variances, corrleations, regression coefficients, etc.*

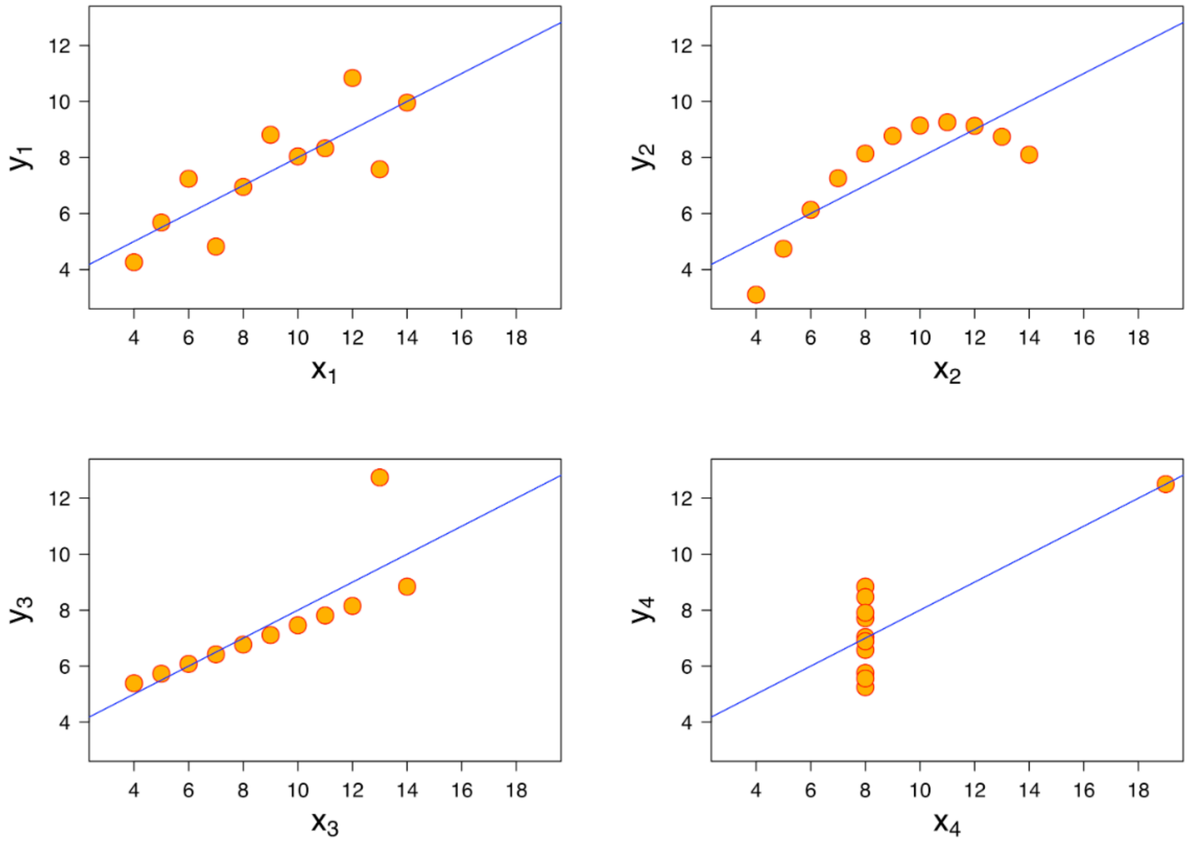
Exploration - Understand data

Property	Value	Accuracy
Mean of x	9	exact
Sample variance of x: s_x^2	11	exact
Mean of y	7.50	to 2 decimal places
Sample variance of y: s_y^2	4.125	±0.003
Correlation between x and y	0.816	to 3 decimal places
Linear regression line	$y = 3.00 + 0.500x$	to 2 and 3 decimal places, respectively
Coefficient of determination of the linear regression: R^2	0.67	to 2 decimal places

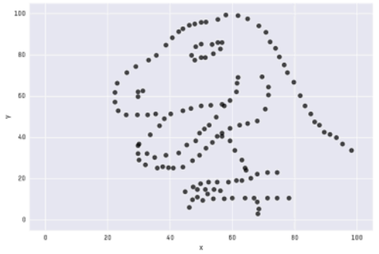
They're the same for every dataset?!?!

- Can we conclude that the datasets are all the same?

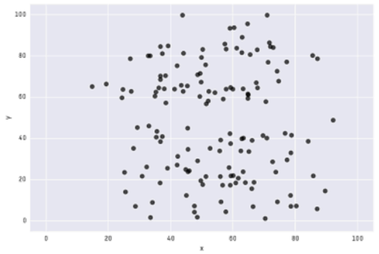
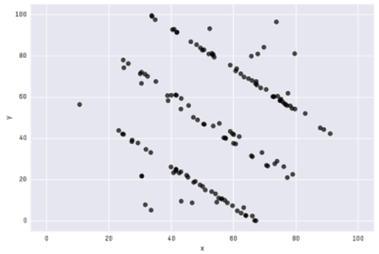
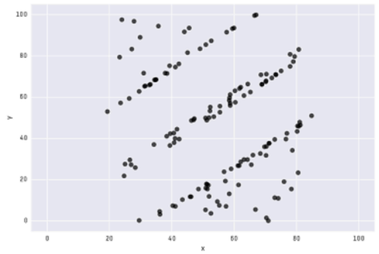
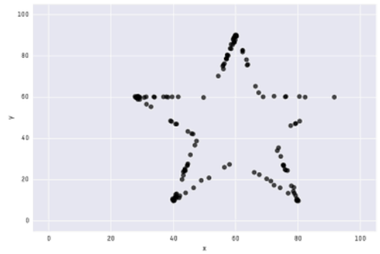
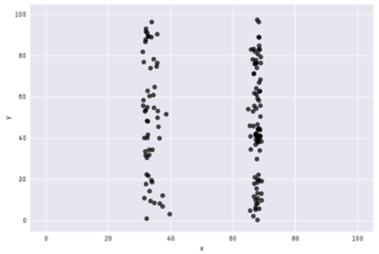
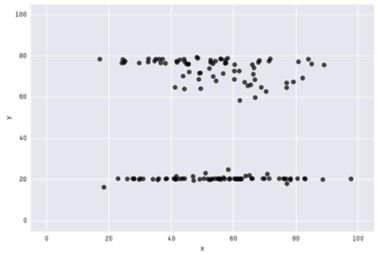
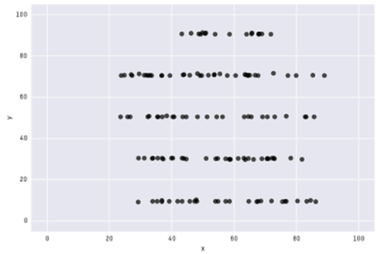
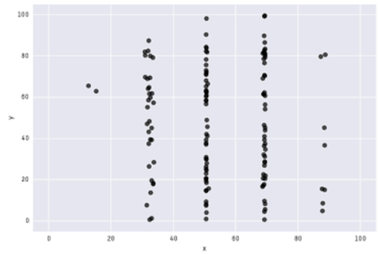
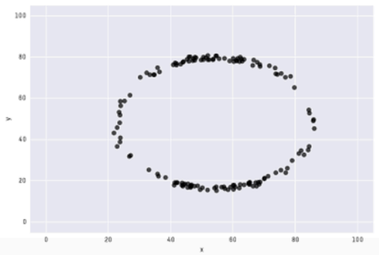
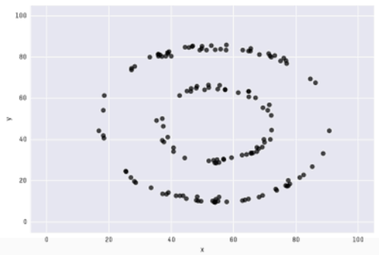
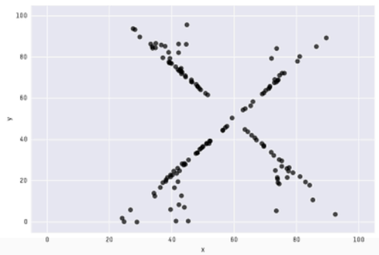
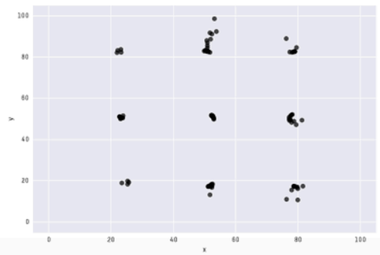
Exploration - Anscombe's quartet



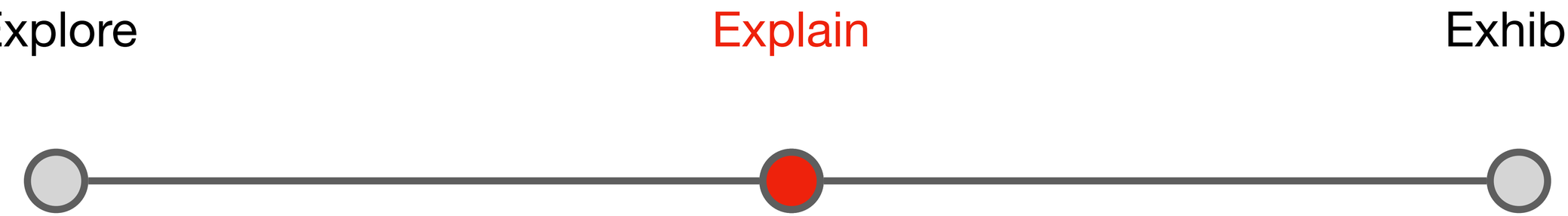
Visualization shows there's much more going on!



X Mean: 54.26
Y Mean: 47.83
X SD : 16.76
Y SD : 26.93
Corr. : -0.06

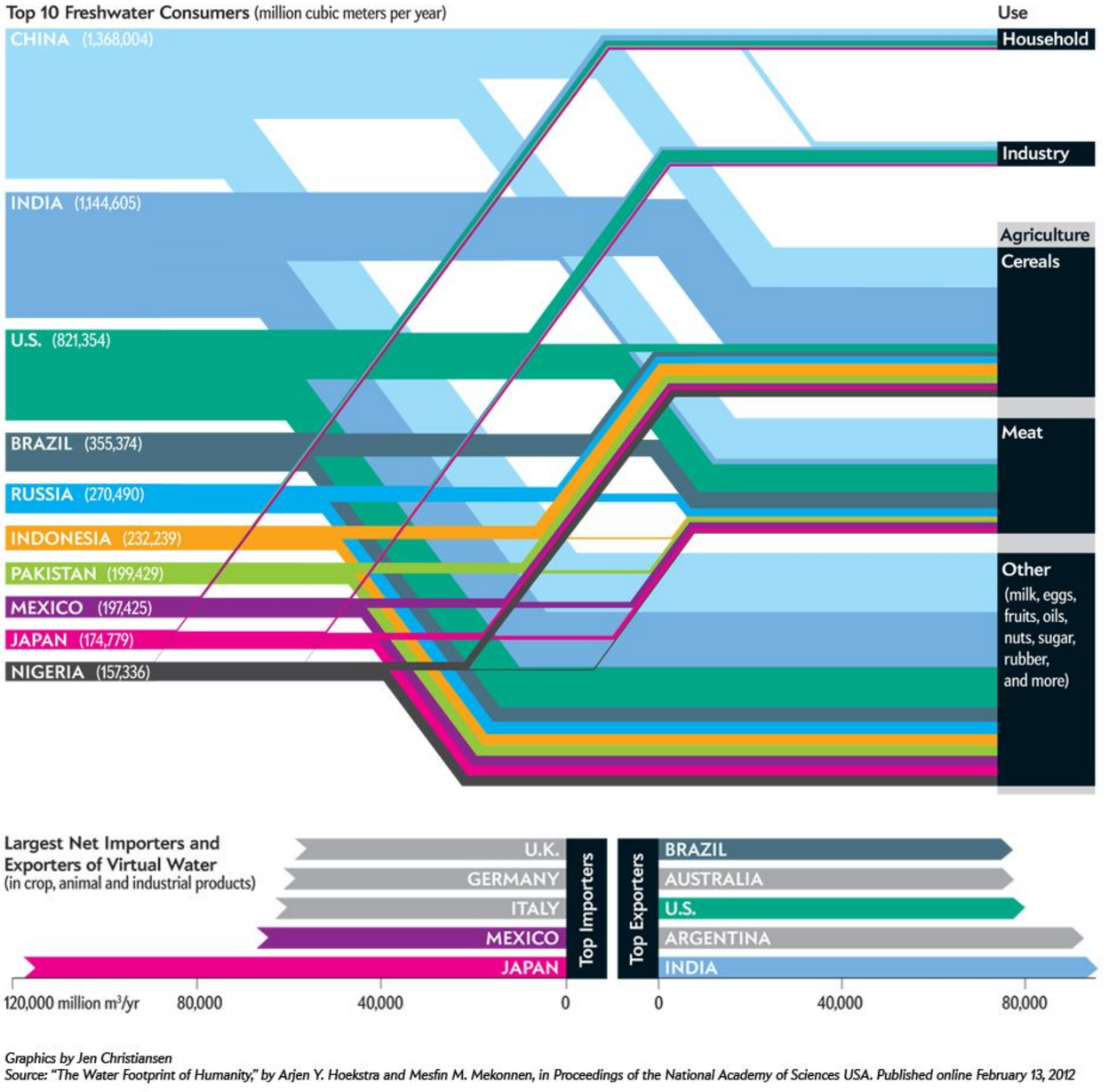













Spectrum of visualization goals



Credit: Christian Swinehart

Explain

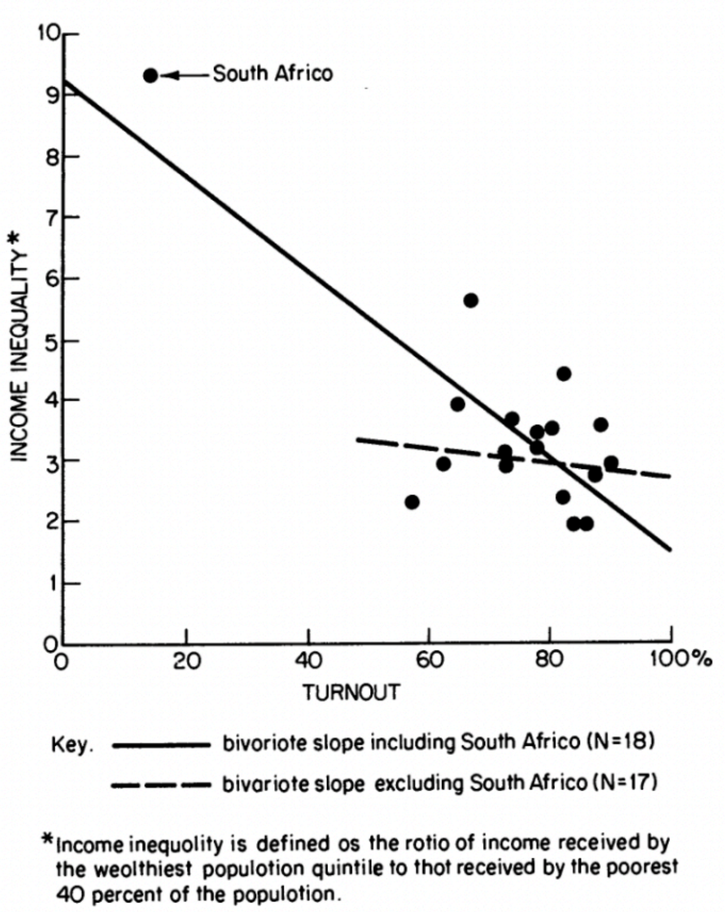


Graphics by Jen Christiansen
Source: "The Water Footprint of Humanity," by Arjen Y. Hoekstra and Mesfin M. Mekonnen, in Proceedings of the National Academy of Sciences USA. Published online February 13, 2012

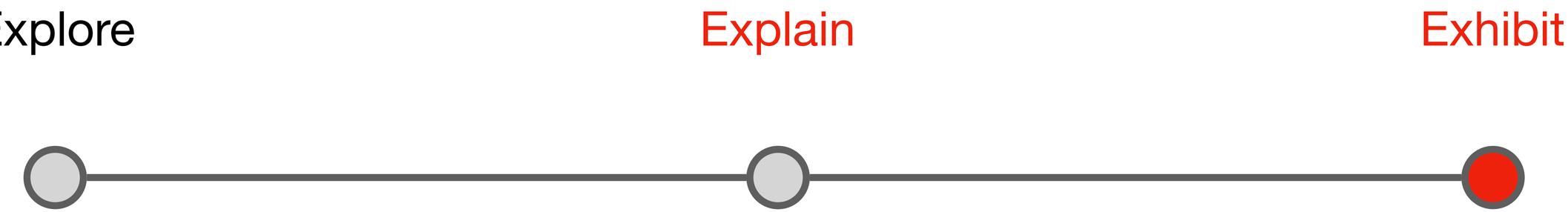
Validation

Even if we *think* we know what's going on, we should visualize to check our understanding!

- **1980:** Jackman shows the effect of earlier work is dominated by an outlier

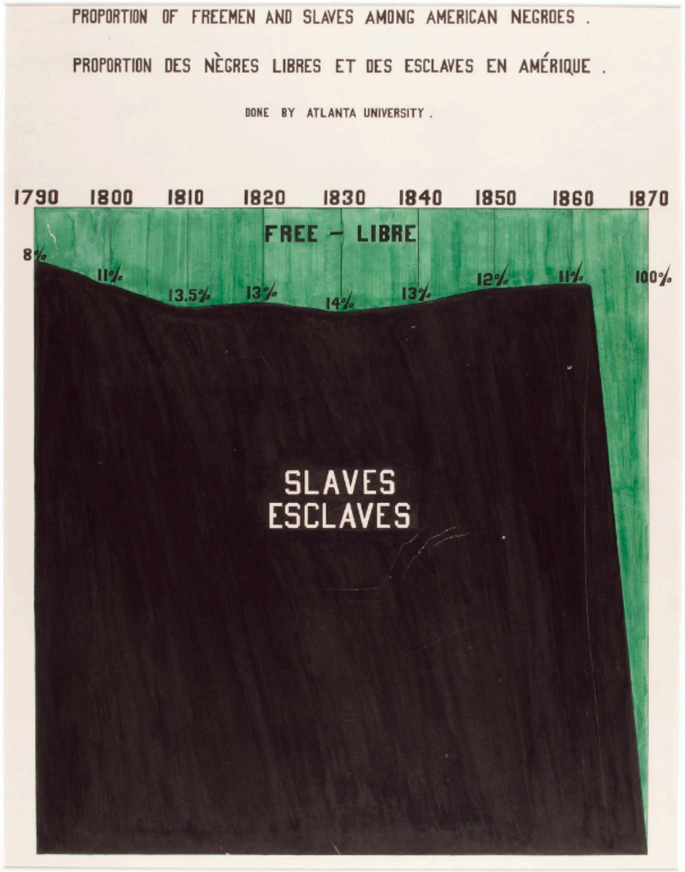


Spectrum of visualization goals



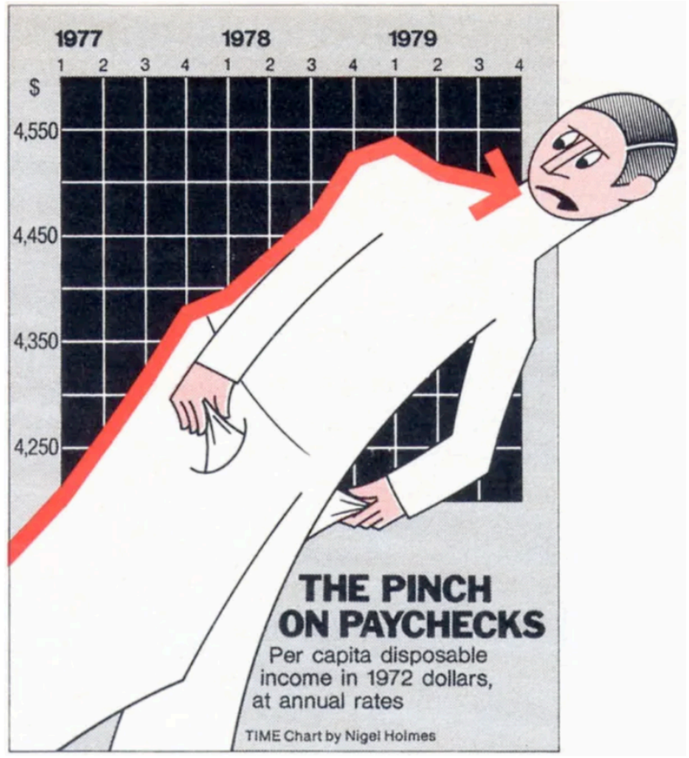
Credit: Christian Swinehart

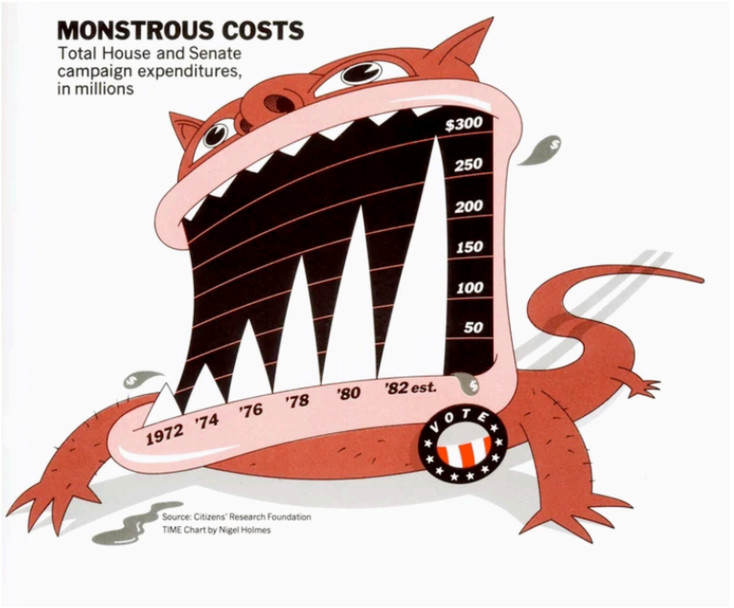
W. E. B. DuBois



Sociologist and civil rights activist W. E. B. Du Bois made many influential visualizations.

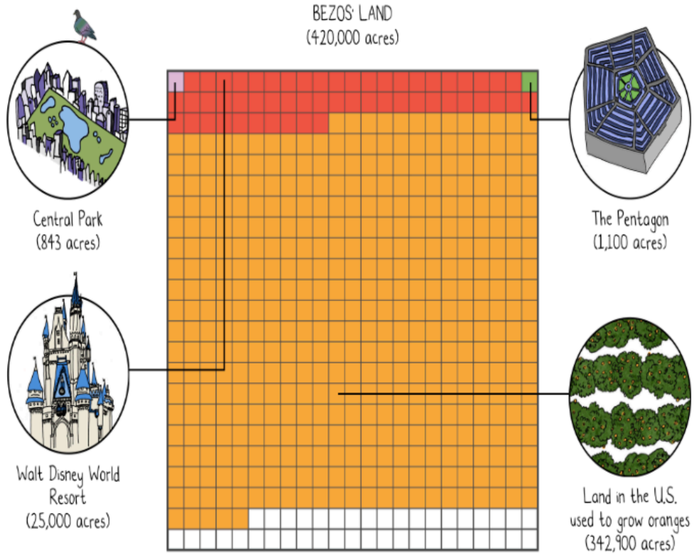
Nigel Holmes

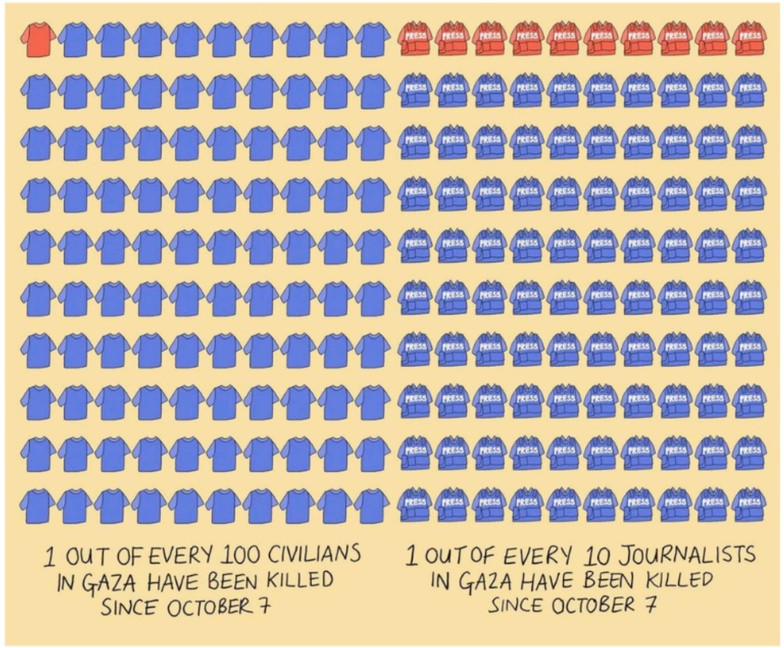




Graphic designer Nigel Holmes is known for playful and informative visualizations.

Mona Chalabi





Contemporary data journalist Mona Chalabi uses visualization to create impactful stories and inspire activism.

Two big themes for this course

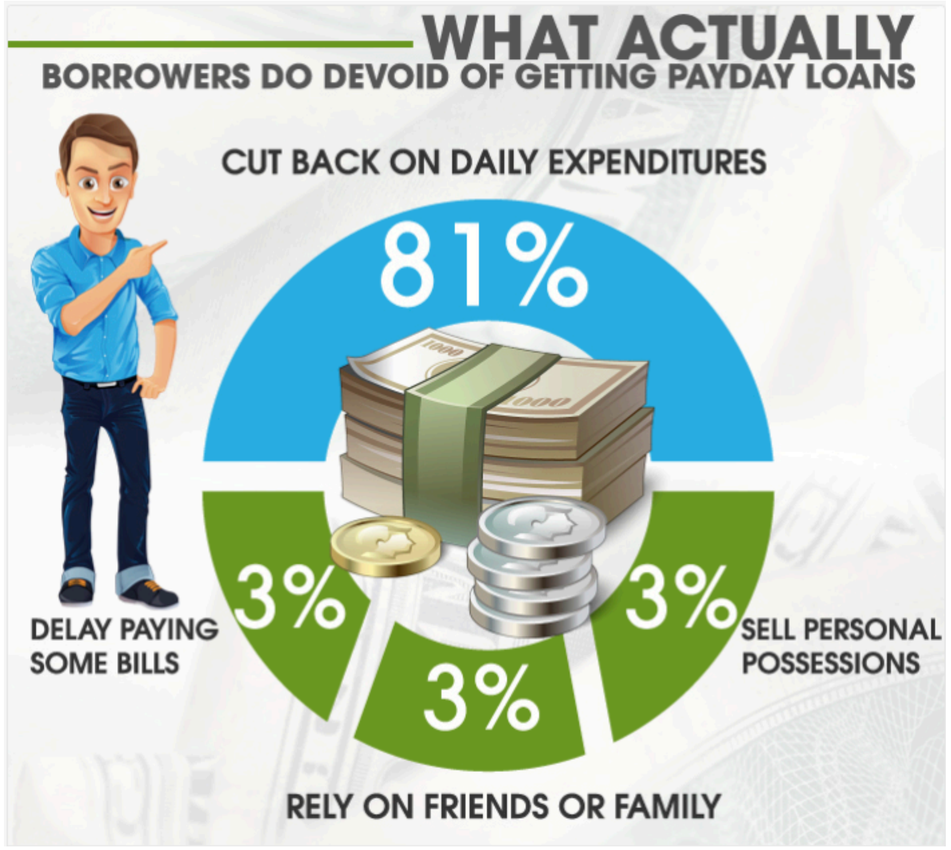
- foundations
  - –building visualizations is fundamentally about tradeoffs. foundations help us understand these tradeoffs and make informed decisions
    - principles: why should I design it in this way vs that way?
    - techniques: what kinds of designs are possible?

- tooling mechanics / programming
  - –how to build a visualization programmatically
  - –D3, JavaScript, CSS, HTML

Foundations: Respect the math in the data

- semantics matter!
- not everything you can do with data makes sense

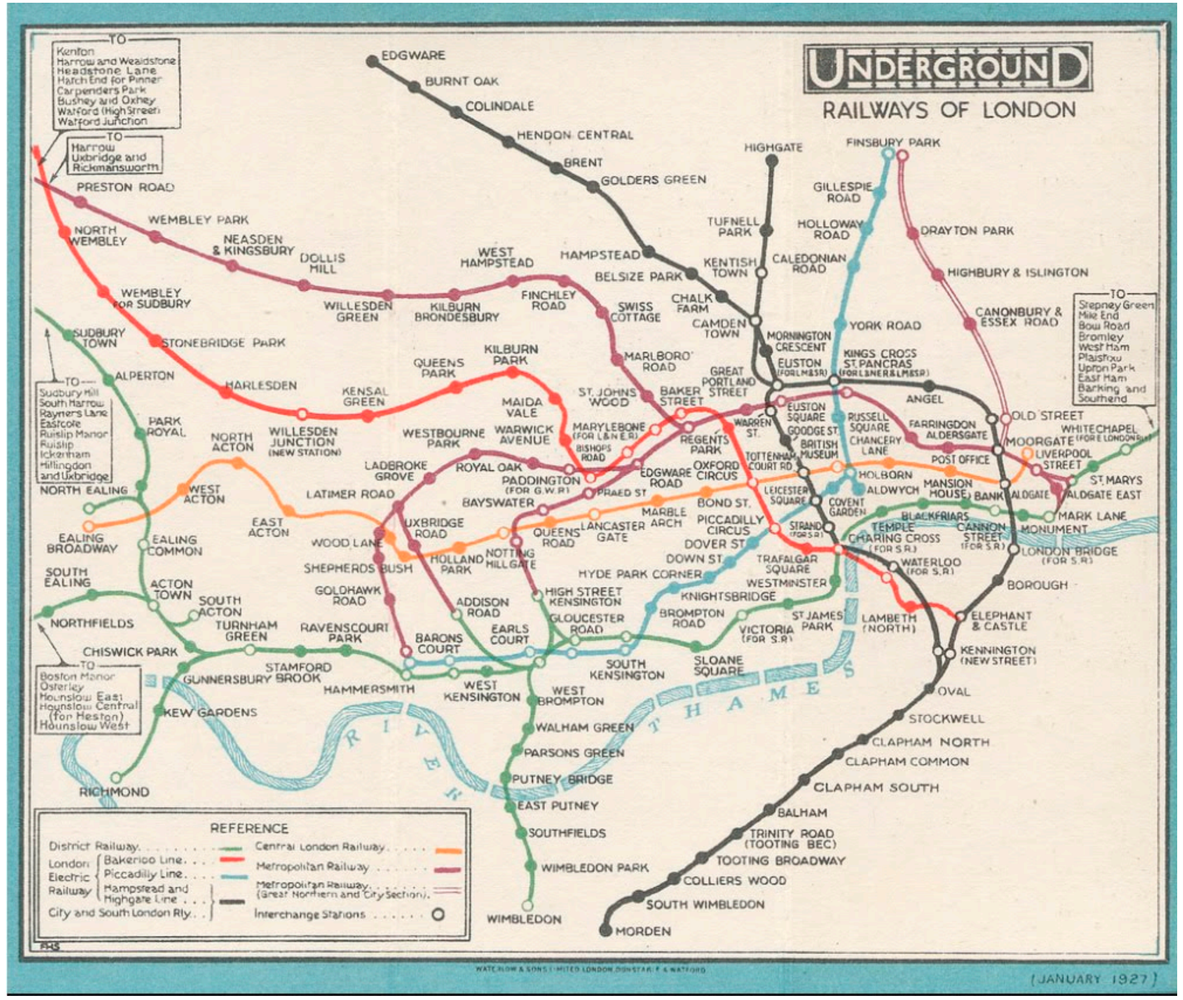
https://imgur.com/gNefvUG/



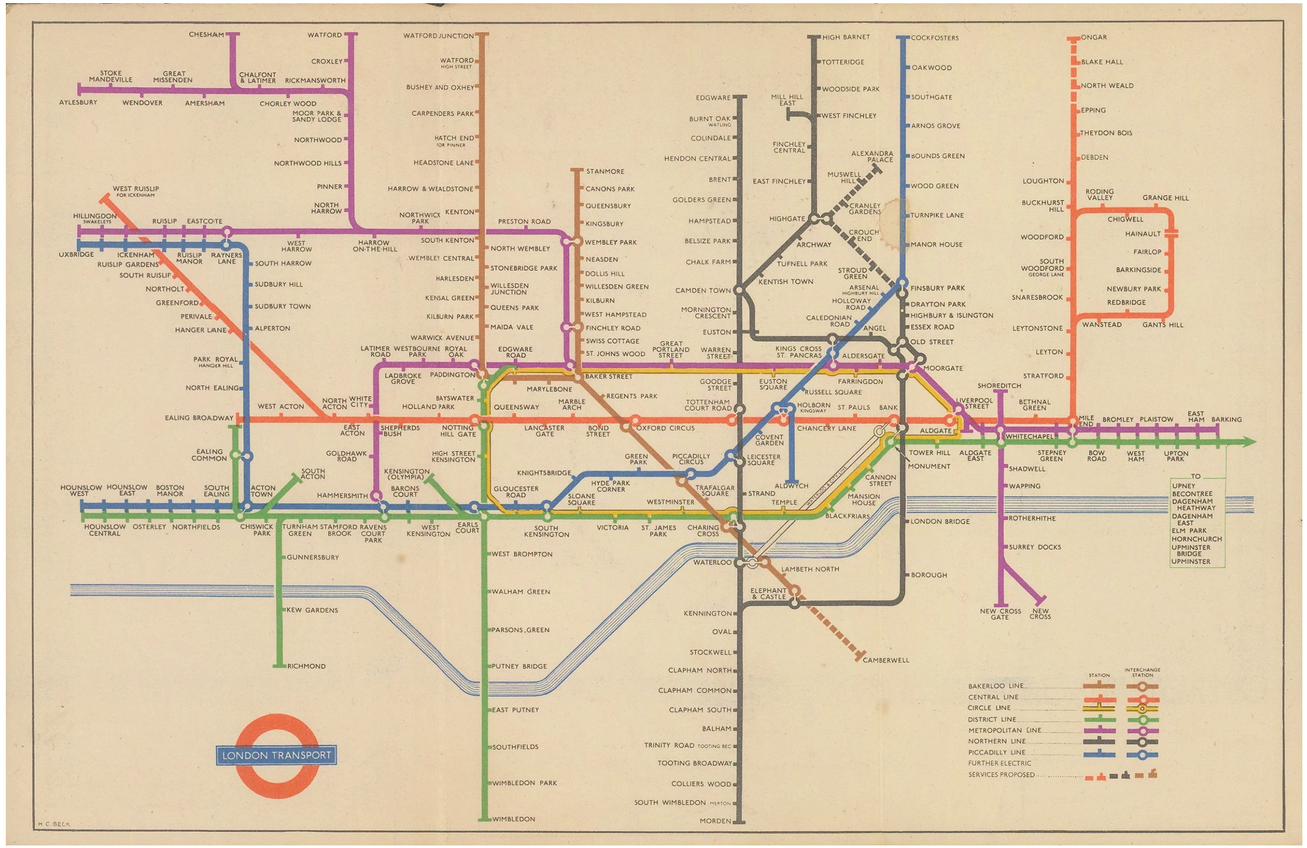
https://viz.wtf/post/107440754050/how-payday-loans-add-up

Foundations: Which subway map is better?

A: 1927



B: 1950



Tooling landscape

https://www.cs.ubc.ca/~tmm/courses/547-22/tools/

Level of Abstraction

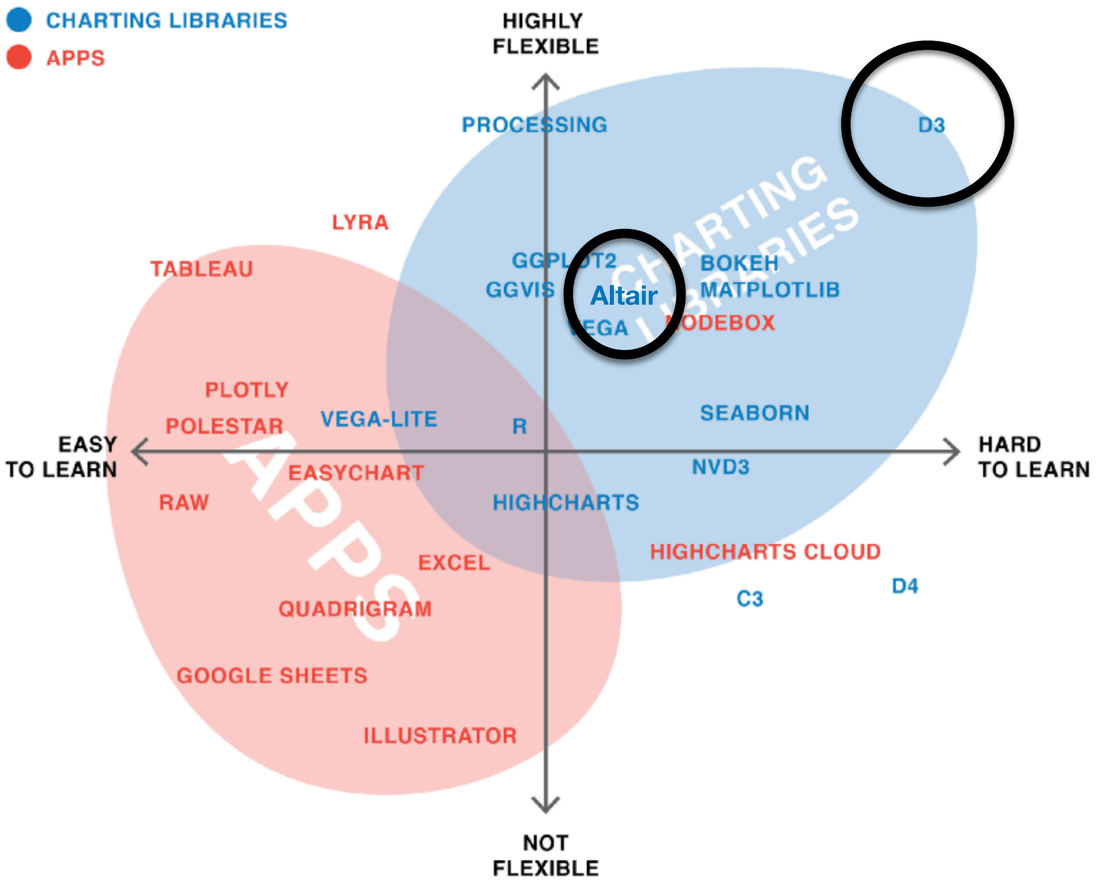
1. Graphics Libraries	2. Low-level Building Blocks	3. Visualization Grammars	4. High-level Building Blocks	5. Chart Templates

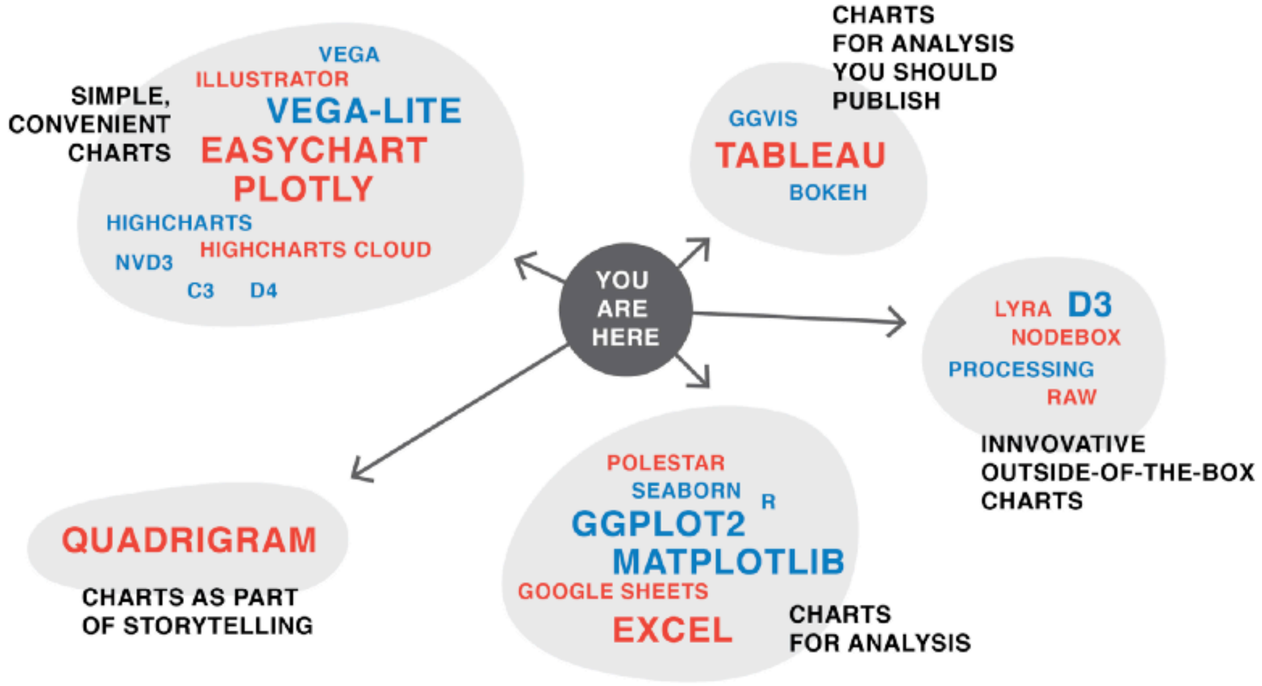
from scratch — Composable Building Blocks — ready-to-use

Source: A metaphorical representation of Levels of Abstraction by Krist Wongsuphasawat.

WebGL **D3** R/ggplot python/Altair Tableau Excel

Our tools: D3 and Altair





https://source.opennews.org/articles/what-i-learned-recreating-one-chart-using-24-tools/

Tooling: Why D3?

- why choose web stack?
  - ubiquitous
  - easy to talk to a server
  - fast
- why D3 in specific?
  - state of the art
  - interactive
  - flexible & powerful: beyond chart libraries

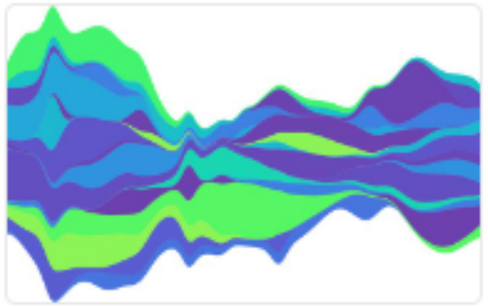
apple.com

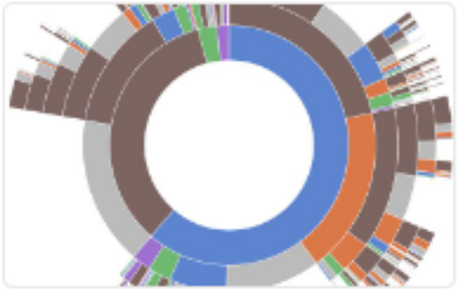
https://observablehq.com/@d3/gallery

D3: interactive & flexible

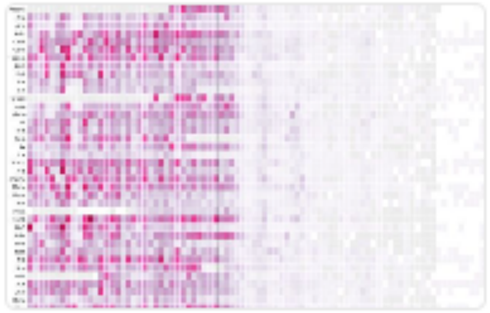
- interactivity

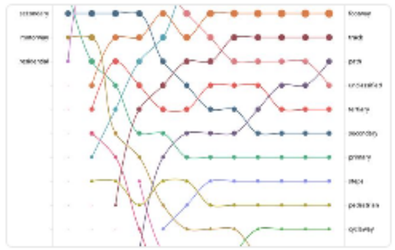
Index chart

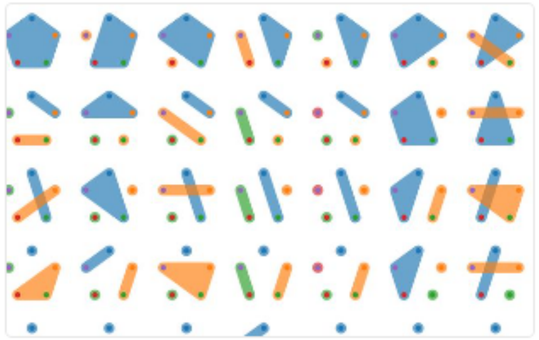

Streamgraph transitions

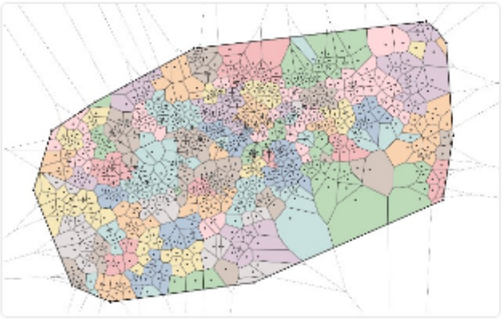

Sequences sunburst

- flexibility:
  beyond basic chart types


The impact of vaccines


Evolution of values used with the highway key in OpenStreetMap (2007-2018)

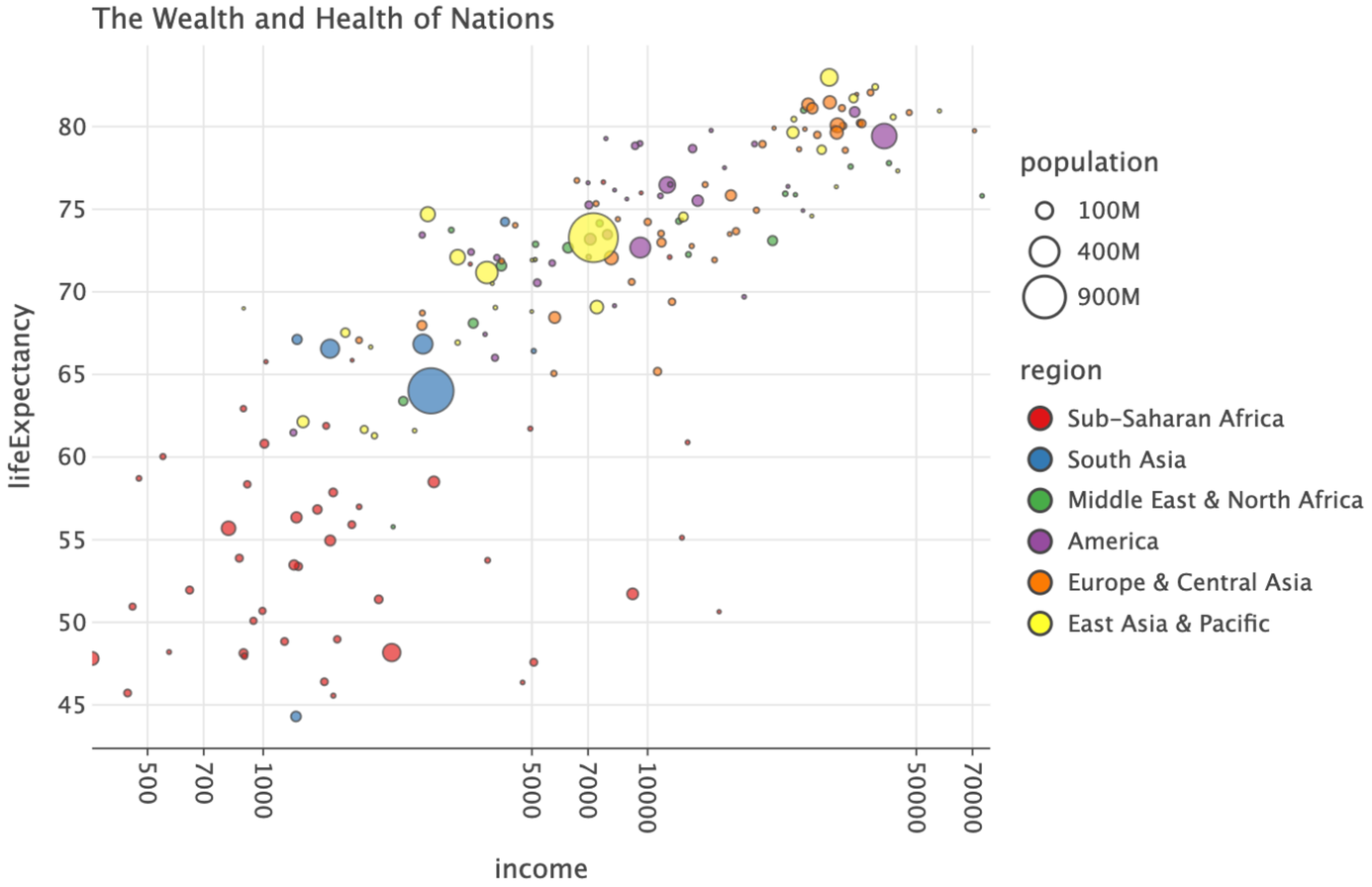

Set Partitions


Creating an area-based map from a set of points

Thinking about data

Warmup exercise

What information is shown in this plot?



Information shown

- Countries! (names)
- Populations
- Incomes
- Life expectencies
- Regions

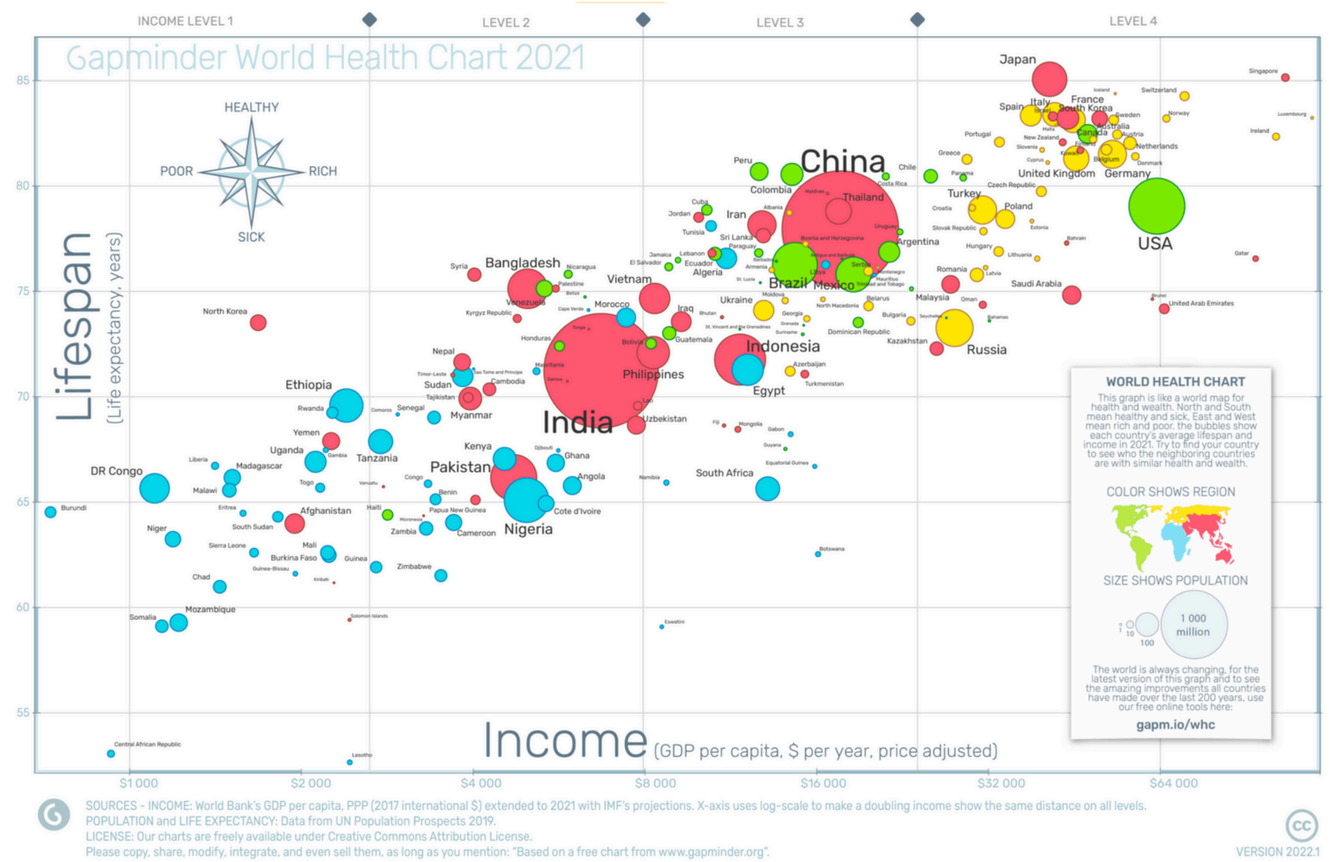
Data source: Gapminder

Gapminder is a Swedish non-profit organization focused on global development

- Founded by Hans Rosling *"Gapminder identifies systematic misconceptions about important global trends and proportions and uses reliable data to develop easy to understand teaching materials to rid people of their misconceptions."*

Dataset

A collection of **values** that capture various aspects of the world
(Typically for a specific **domain**)



What is data?

Some *rough* definitions

Variable:

A **variable** defines some *measurement* we can make about the world

Nations dataset

- A country's *name*, *population*, *income*, *life expectency*, *etc.*

Colleges dataset

- A college's *name*, *tuition*, *enrollment*, *type (public vs. private)*, *etc.*

Observation:

An **observation** is a collection of *values* corresponding to a single *entity*

- Name: *United States*
- Population: *300M*
- Life exp.: *76.3*
- Income: *47,000*

- Name: *Chile*
- Population: *20M*
- Life exp.: *78.9*
- Income: *17,000*

Observation:

An **observation** is a collection of *values* corresponding to a single *entity*

- Name: *Harvey Mudd*
- Tuition: *$68,262*
- Enrollment: *915*
- Type: *Private*

- Name: *UC Irvine*
- Tuition: *$17,105*
- Enrollment: *29,503*
- Type: *Public*

One way to think of this:

VARIABLES

OBSERVATIONS

	Tuition	Enrollment	Public vs. Private	…
Smith College	$46,288	2,563	private	
UMass Amherst	$16,115	28,635	public	
Hampshire College	$48,065	1,400	private	
Mount Holyoke College	$43,886	2,189	private	
Amherst College	$50,562	1,792	private	
⋮				

Credit: R. Jordan Crouser, Smith College

Another way to think of this:

```
class school_obs:
   def __init__(tuition, enrollment,
                pub_or_priv):
      self.tuition = tuition
      self.enrollment = enrollment
      self.pub_or_priv = pub_or_priv
```

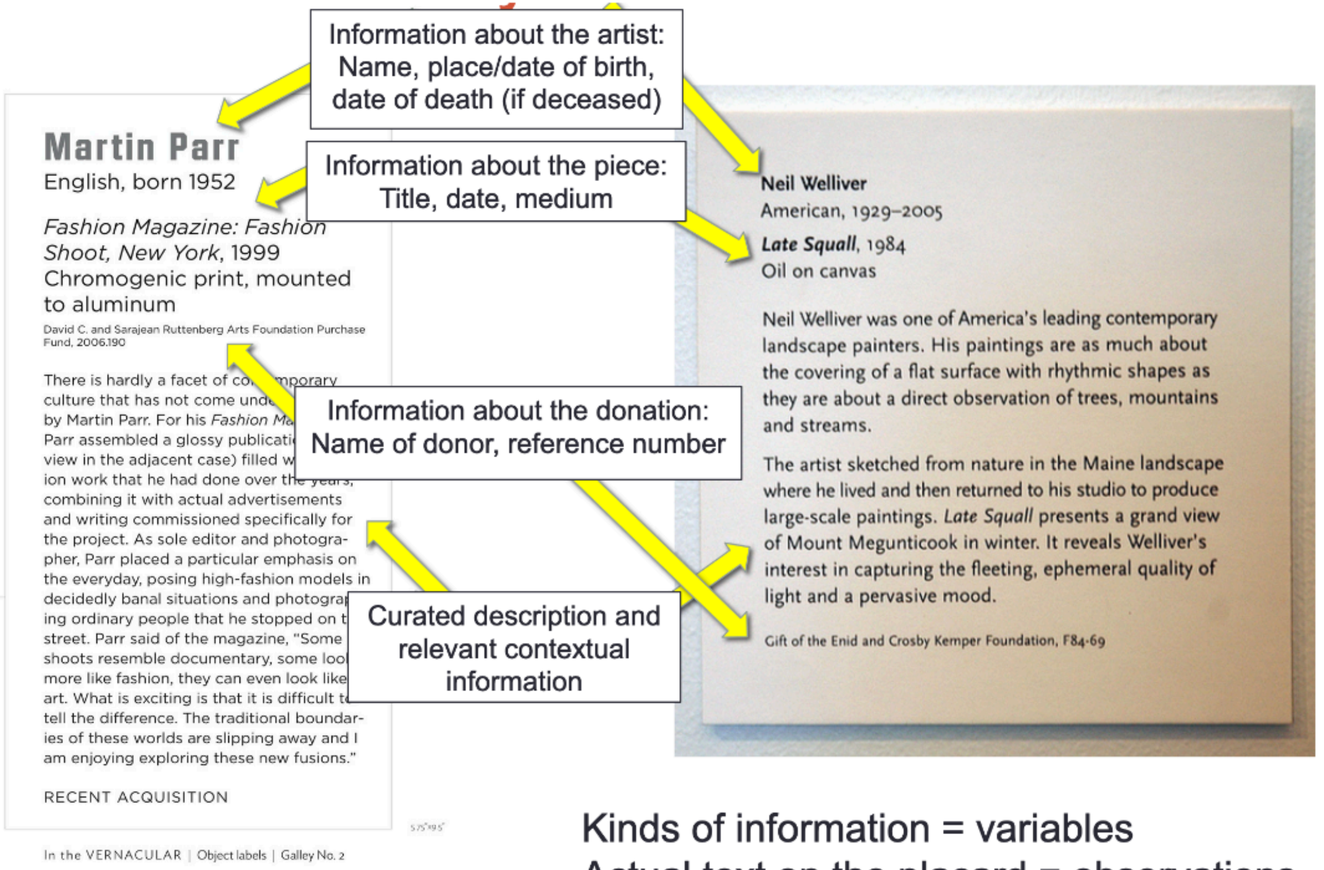
VARIABLES

OBSERVATIONS

```
smith = school_obs(46288, 2563, "private")
umass = school_obs(16115, 28635, "public")
```

Credit: R. Jordan Crouser, Smith College

Yet another way to think of this:



Kinds of information = variables
Actual text on the placard = observations

Credit: R. Jordan Crouser, Smith College

Let’s get to code!

The Pandas library

Pandas is the core library we'll use for representing datasets in code

The central data structure in Pandas is the `DataFrame`

```
1 # A DataFrame object
2 nations = pd.read_csv('data/nations.csv')
3 nations.head() # Shows first 5 rows
```

	name	region	income	population	lifeExpectancy
0	Angola	Sub-Saharan Africa	5055.59	12707546	47.58
1	Benin	Sub-Saharan Africa	1457.57	8294941	61.89
2	Botswana	Sub-Saharan Africa	12282.28	1638393	55.12
3	Burkina Faso	Sub-Saharan Africa	1234.42	14761339	53.38
4	Burundi	Sub-Saharan Africa	457.07	8691005	50.95

- A 2-dimensional "array" of values
- We can view it as a *table* with *rows* and *columns*

The Pandas library

Each column has a *label*, as shown.

- Should specify what the values in a column mean

```
# A DataFrame object
nations = pd.read_csv('data/nations.csv')
nations.head() # Shows first 5 rows
```

	name	region	income	population	lifeExpectancy
0	Angola	Sub-Saharan Africa	5055.59	12707546	47.58
1	Benin	Sub-Saharan Africa	1457.57	8294941	61.89
2	Botswana	Sub-Saharan Africa	12282.28	1638393	55.12
3	Burkina Faso	Sub-Saharan Africa	1234.42	14761339	53.38
4	Burundi	Sub-Saharan Africa	457.07	8691005	50.95

```
nations.columns
```

```
Index(['name', 'region', 'income',
'population', 'lifeExpectancy'],
dtype='object')
```

Rows can (*sort of*) have labels too, we'll get back to this!

Series

Each column in Pandas is a 1-dimensional array-like object called a **series**

```
1 nations['population']
```

```
0      12707546
1       8294941
2       1638393
3      14761339
4       8691005
         ...
175     1130120
176        1401
177      118993
178    86116559
179      215053
Name: population, Length: 180, dtype:
int64
```

```
1 nations['name']
```

```
0             Angola
1              Benin
2           Botswana
3       Burkina Faso
4            Burundi
              ...
175      Timor-Leste
176           Tokelau
177             Tonga
178           Vietnam
179           Vanuatu
Name: name, Length: 180, dtype: object
```

Tidy data

> *“Happy families are all alike; every unhappy family is unhappy in its own way.”* –– Leo Tolstoy

> *“Tidy datasets are all alike, but every messy dataset is messy in its own way.”* –– **Hadley Wickham

How would you organize this data?

A

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

B

country	year	type	count
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

C

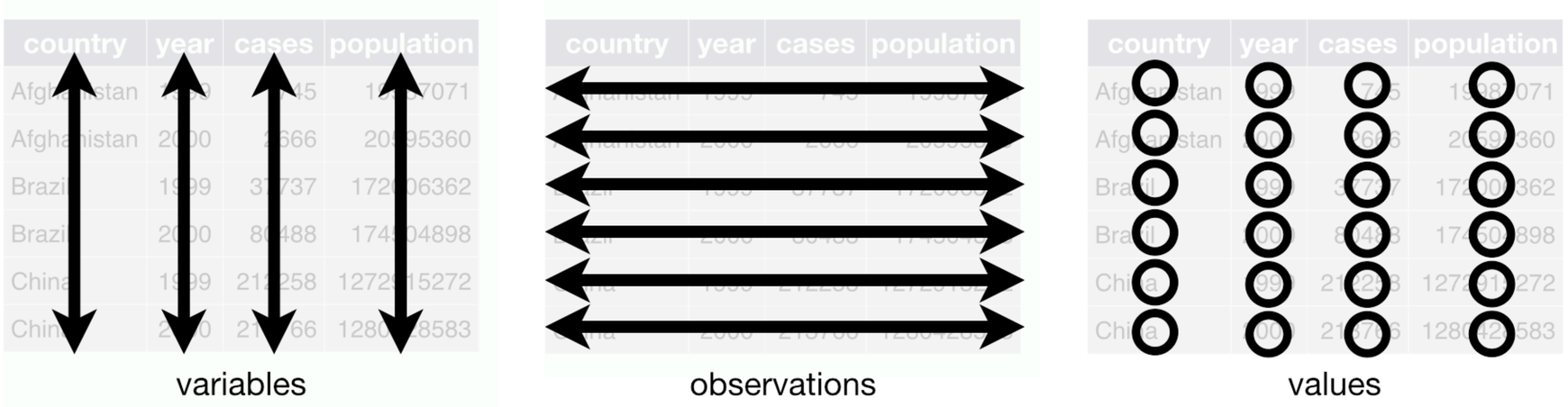
country	year	rate
Afghanistan	1999	745/19987071
Afghanistan	2000	2666/20595360
Brazil	1999	37737/172006362
Brazil	2000	80488/174504898
China	1999	212258/1272915272
China	2000	213766/1280428583

D

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

country	1999	2000
Afghanistan	19987071	20595360
Brazil	172006362	174504898
China	1272915272	1280428583

Wickham's tidy data principles



- Each variable forms a column.
- Each observation forms a row.
- Each type of observational unit forms a table (DataFrame)
- Each value has its own cell.

Why tidy data?

Consistency

- Different practitioners tools have a consistent *format* for data
- Useful for visualization in particular

Implementation

- Easier and more efficient to implement

Tidy

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

Too long

Multiple variables are stored in one column

country	year	type	count
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

Too compact

Multiple values are stored in one cell

country	year	rate
Afghanistan	1999	745/19987071
Afghanistan	2000	2666/20595360
Brazil	1999	37737/172006362
Brazil	2000	80488/174504898
China	1999	212258/1272915272
China	2000	213766/1280428583

Too wide

Variables are stored in both rows and columns. (*Wide form data*)

A single observational unit is stored in multiple tables

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

country	1999	2000
Afghanistan	19987071	20595360
Brazil	172006362	174504898
China	1272915272	1280428583

Always the right choice?

No! In some cases untidy data can be preferred!

- Often for performance/memory reasons
- We'll cover this when we get to spatial data

Space of data types

Tamara Munzer

What?

Datasets

Data Types

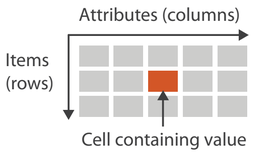
→ Items → Attributes → Links → Positions → Grids

Data and Dataset Types

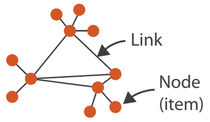
Tables	Networks & Trees	Fields	Geometry	Clusters, Sets, Lists
Items	Items (nodes)	Grids	Items	Items
Attributes	Links	Positions	Positions	
	Attributes	Attributes		

Dataset Types

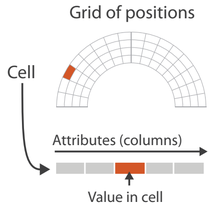
→ Tables



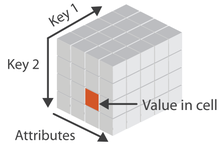
→ Networks



→ Fields (Continuous)



→ *Multidimensional Table*



→ *Trees*

→ Geometry (Spatial)

Attributes

Attribute Types

→ Categorical

→ Ordered

→ *Ordinal*

→ *Quantitative*

Ordering Direction

→ Sequential

→ Diverging

→ Cyclic

Variable types

A defining characteristic of a variable is its **type**

```
1 nations['population'].dtype
```

```
dtype('int64')
```

- This should be familar to computer scientists!

The **type** of a variable determines what kinds of values it can take and how we should interpret them

Variable types

- Nominal
  - Categorical
  - Arbitrary
- Ordinal
- Quantitative
  - Interval
  - Ratio

Nominal variables

An *unordered* set of non-numeric values, representing labels or categories

Categorical - Finite

Possible values are finite and *known*

- Colors: *{red, green, blue}*
- Regions: *{South Asia, Europe & Central Asia, …}*

Arbitrary - Infinite

Possible values are unbounded

- Addresses: *{"12 Main St. Boston MA", "45 Wall St. New York NY", …}*
- Names: *{"John Smith", "Jane Doe", …}*

Variable types

- Nominal
  - Categorical
  - Arbitrary
- **Ordinal**
- Quantitative
  - Interval
  - Ratio

Ordinal variables

An *ordered* set of (usually) non-numeric values, representing levels

- Grades: *{A, B, C, D, F}*
- Ratings: *{G, PG, PG-13, R}*

Variable types

- Nominal
  - Categorical
  - Arbitrary
- Ordinal
- **Quantitative**
  - Interval
  - Ratio

Quantitative variables

Numeric data that we can perform mathematical operations on

Interval

Location of 0 is arbitrary, only differences/intervals can be compared

- Date-Times: *Jan, 19, 2006*

Ratio

0 is well-defined and meaningful. Can compare values as *ratios*

- Height: *{5'3", 6'1", 5'9",…}*
- Population: *23M, 350k, 1.4B, …*

Operations by type

Nominal (Labels or categories)

- Concrete types: `str`, `bool`, `category`
- Operations: =, ≠

Ordinal (Ordered, non-numeric)

- Concrete types: `str`, `int`, `category`
- Operations: =, ≠, <, >

Interval (Arbitrary 0)

- Concrete types: `float`, `int`, `datetime`
- Operations: =, ≠, <, >, -

Ratio (Well-defined 0)

- Concrete types: `float`, `int`, `timedelta`
- Operations: =, ≠, <, >, -, /

Data vs conceptual model, example

- data model: floats
  - 32.52, 54.06, -14.35, ...
- conceptual model
  - temperature
- multiple possible data abstractions
  - continuous to 2 significant figures: quantitative
    - task: forecasting the weather
  - hot, warm, cold: ordinal
    - task: deciding if bath water is ready
  - above freezing, below freezing: categorical
    - task: decide if I should leave the house today