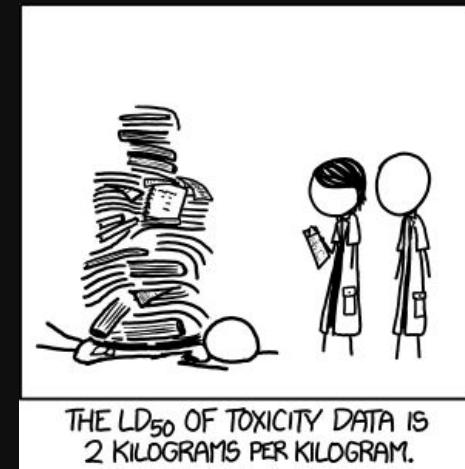


Session 1

Data at Scale: An Introduction



THE LD₅₀ OF TOXICITY DATA IS
2 KILOGRAMS PER KILOGRAM.

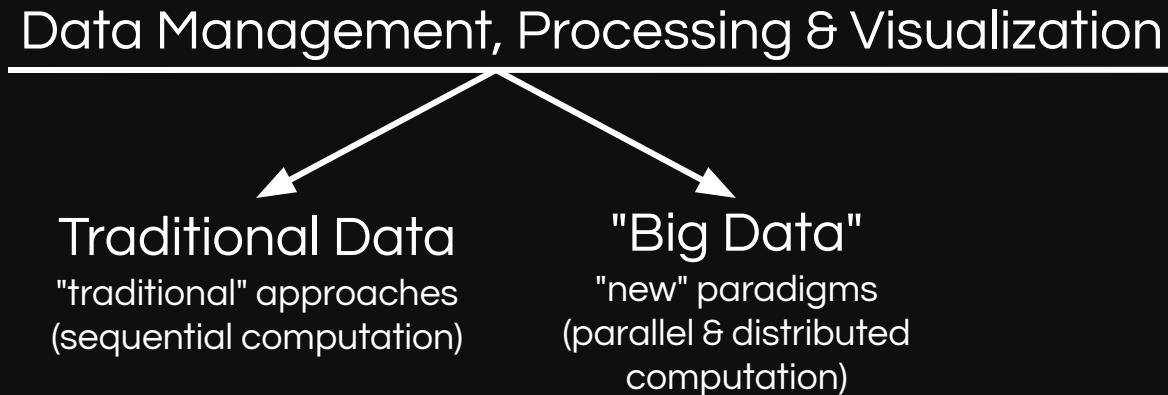


What do I mean by *Data at Scale* anyway?

- Data (*that uses computers to be managed and processed*)

So, in this module

The different options to:





But what is Big Data anyway...

Data has been around for a long time....

1975 Conference on Very Large Databases was established

Turning data into useful information has also been around for a long time....

- Data analytics
- Data mining
- Knowledge discovery
- Machine learning
- Statistical modelling...



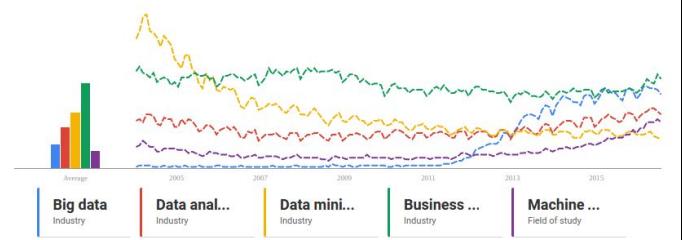
NLAB:

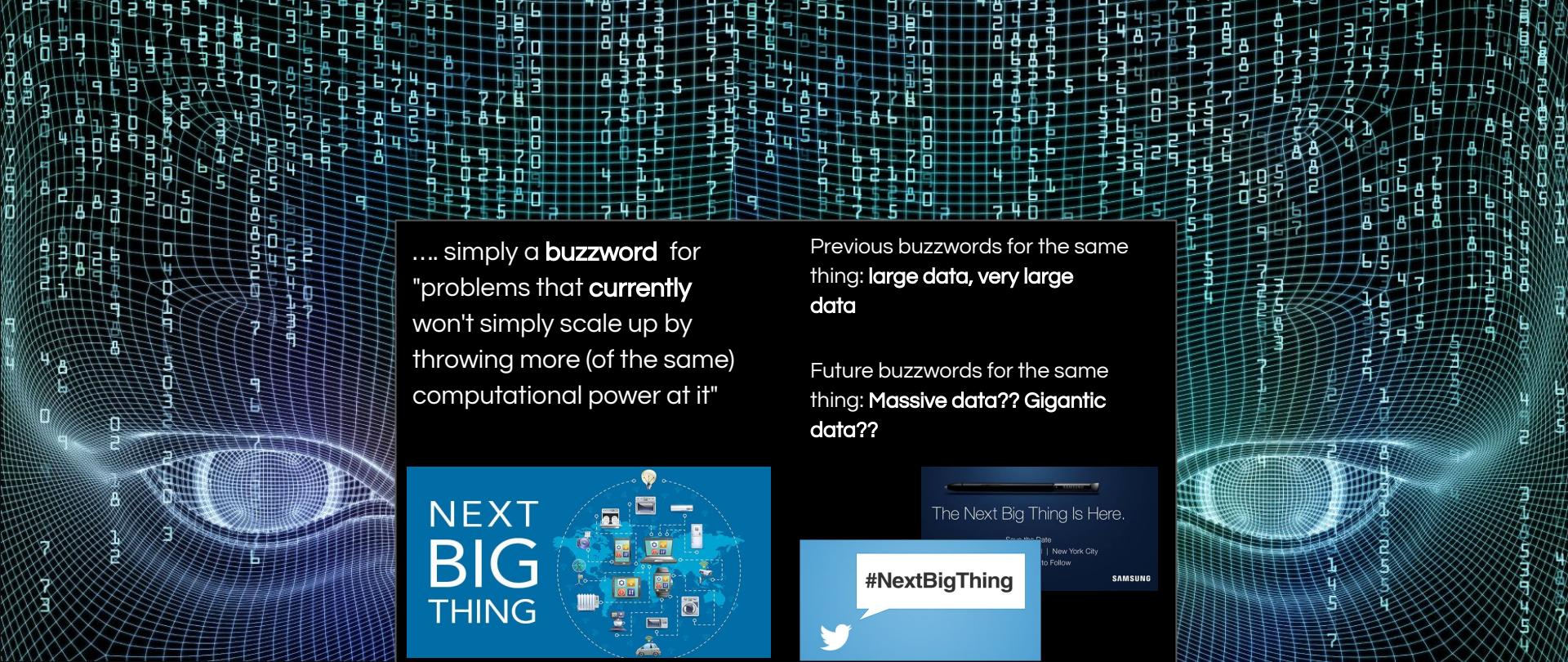
Data at Scale

Dr Evgeniya Lukinova

Why is there
a new term...

Popularity of terms over
time (Google Trends)



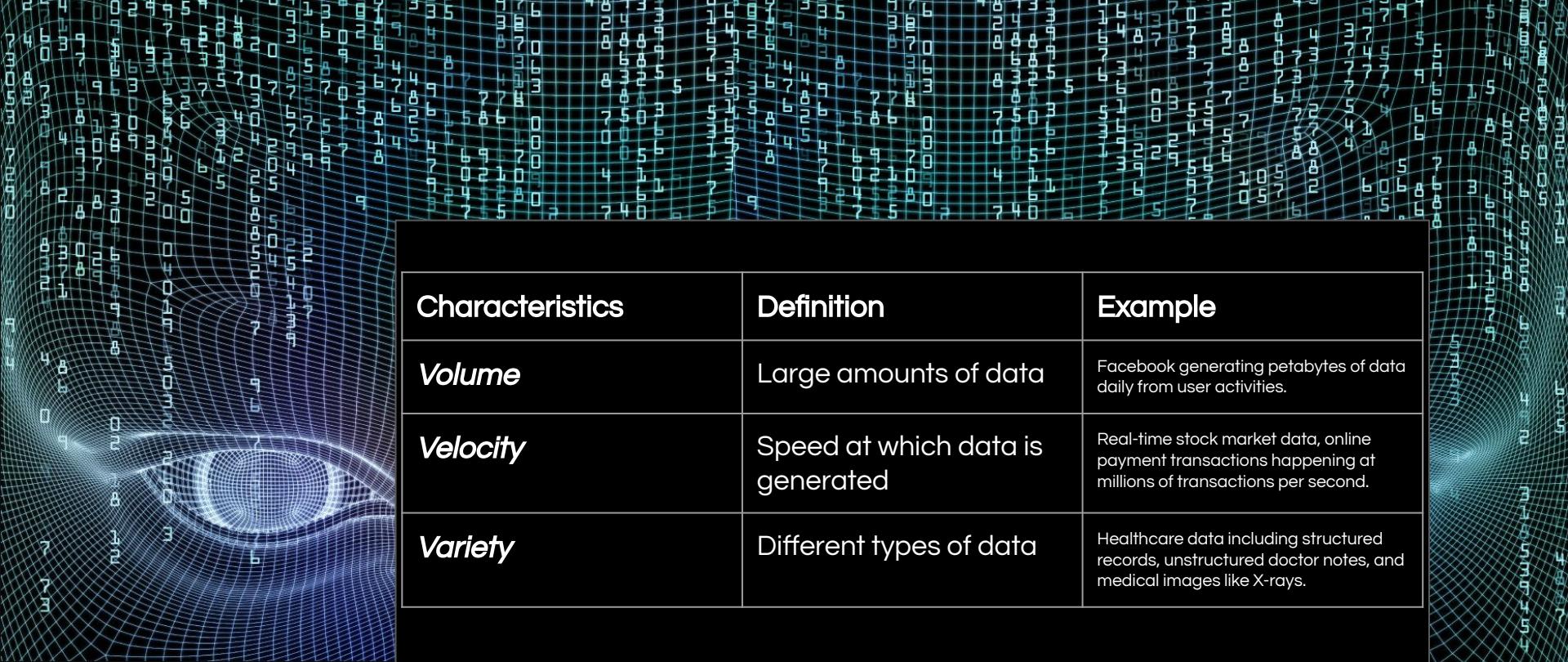


.... simply a **buzzword** for
"problems that **currently**
won't simply scale up by
throwing more (of the same)
computational power at it"

Previous buzzwords for the same
thing: **large data, very large
data**

Future buzzwords for the same
thing: **Massive data?? Gigantic
data??**





Characteristics	Definition	Example
Volume	Large amounts of data	Facebook generating petabytes of data daily from user activities.
Velocity	Speed at which data is generated	Real-time stock market data, online payment transactions happening at millions of transactions per second.
Variety	Different types of data	Healthcare data including structured records, unstructured doctor notes, and medical images like X-rays.

3 Vs model of Big Data, Doug Laney, 2001



Data at Scale

Dr Evgeniya Lukinova



So "Big Data" is not
a new field.

(just continuation of advances)

So "Big Data" is not
a single technology

(it is not hadoop)

Useful label

(Data size causing issues for
current methods)

Signals to business

- New technology required.
- New ways of thinking/training
are often required.

To sell intelligently...
Why?

to underpin
analytics...



New Product Prediction
(targeted marketing)



Smoother legs? It's all in the Swirl

Hello Seed,

It hugs every contour and curve to give you the smoothest legs ever – for half price! The new Venus Swirl Razor with Flexiball Technology is now in store and online at Boots. With its revolutionary design and five adjusting blades, it gives you six times more flexibility* - contouring over curves and leaving virtually no missed hairs, for long-lasting smoothness.

So don't miss out on flawless skin, pick up your Venus Swirl today for half price.

The Boots Advantage Card Team

[Shop now](#)

P.S. Great news! Did you know that any Venus blade can fit onto any Venus handle?



[Buy now](#)



Data at Scale

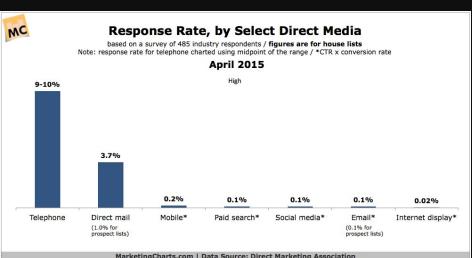
Dr Evgeniya Lukinova

To sell intelligently...
Why?

to underpin
analytics...



Some notes on the study below
Mix of B2B (52%), B2C (32%) & unknown
Study run in 2012
House = current/former customers



New Product Prediction
(targeted marketing)

Runs significant direct mailing
campaigns

In 2013 it was reported¹ they had a
redemption rate of over 70%
(uplift though?...)

Worked with dunnhumby



Smoother legs? It's all in the Swirl

Hello Seed,

It hugs every contour and curve to give you the smoothest legs ever – for half price! The new Venus Swirl Razor with Flexiball Technology is now in store and online at Boots. With its revolutionary design and five adjusting blades, it gives you six times more flexibility* - contouring over curves and leaving virtually no missed hairs, for long-lasting smoothness.

So don't miss out on flawless skin, pick up your Venus Swirl today for half price.

The Boots Advantage Card Team

[Shop now](#)

P.S. Great news! Did you know that any Venus blade can fit onto any Venus handle?



Venus with a Touch of Olay Violet Swirl Shave Gel

Why not try four times the moisture for an even smoother shave? Just pair your Swirl Razor with Satin Care with a Touch of Olay Violet Swirl Shave Gel and you'll be wowed by the difference!

[Buy now](#)

[1] <https://www.forbes.com/sites/tomgroenfeldt/2013/10/28/kroger-knows-your-shopping-patterns-better-than-you-do/#2784b20d746a>

To adjust staffing &
supply chain
strategies...
Why?

to underpin
analytics...

Sell intelligently



Predicting company's near future
prospects

A screenshot of the Not On The High Street website. At the top, there is a search bar and navigation links for "CHRISTMAS", "NEW", "GIFTS", "CARDS", "EDITS", "HOME", "PRINTS & ART", "JEWELLERY", "BABY & CHILD", "FOOD & DRINK", "WEDDINGS", and "SEE MORE". Below the navigation, there are several promotional sections: "NOT ON THE HIGH STREET choose a life less ordinary", "THOUGHTFUL GIFTS FOR EVERY OCCASION", "SHOP NOW", "BRAND NEW WAYS TO make her feel wonderful", "SHOP BEST NEW GIFTS FOR HER", "Marvellously magical CHRISTMAS INSPIRATION", "Williams' Santa Express", "Gifted.", and "SHOP THE SEPTEMBER COLLECTIONS". There are also images of a woman wearing a striped scarf and a small wooden train.



Data at Scale

Dr Evgeniya Lukinova

To adjust staffing &
supply chain
strategies...
Why?

to underpin
analytics...

Sell intelligently



Predicting company's near future
prospects



The Weather Company:

100,000 dedicated weather
sensors

~10 billion forecast points / day
(sensors + smartphones etc)

Weather can have non-trivial
impact on sales.

Also impacts energy companies
supply/demand forecasting.

To minimize

equipment & asset
failures...

Why?

to underpin
analytics...

Sell intelligently



Preventative maintenance
i.e. predict maintenance requirements



Aim: Reduce unplanned aircraft
engine maintenance.

Each engine:
~ 100 parameters per snapshot
soon ~5000 parameters

Petabytes of data
1PB= 1000 TB
~4000 of the hard drives in your laptop



To leverage
customer lifetime
value...
Why?

to underpin
analytics...

Sell intelligently



Minimize Equipment & Asset Failures



Depending on how you define
customer....

avis budget group

Segmentation
(predicted lifetime value)

Tiered incentives

Multichannel marketing campaign

Additionally : forecasting regional
demand for fleet placements &
pricing

More information:

<https://www.informationweek.com/it-leadership/big-data-6-real-life-business-cases>

Why?

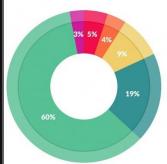
to underpin
analytics...

But is data management,
processing & visualization
really worth caring about...

DATA MANAGEMENT

Data scientists time:

- Building training sets: 3%
- Cleaning and organizing data: 60%
- Collecting data sets: 19%



CrowdFlower
via Forbes 2017

DATA PROCESSING

*72% of business and analytics leaders **aren't satisfied** with how long it takes to retrieve the insights they need from data* Alteryx

- Mining data for patterns: 9%
- Refining algorithms: 4%
- Other: 5%

DATA VISUALIZATION

"Data **visualization** is the key to **actionable insights** "

Head of Bus. Intel. & Data Analytics at AccuWeather

Actionable Insights : The Missing Link Between Data And **Business Value** " Director Data Strategy (Domo), in Forbes 2016

"74% of firms say they want to be "data-driven," **only 29%** say they are good at connecting analytics to action" Forrester 2016

DATA ANALYTICS IN BUSINESS

Through 2017, **60%** of data projects will fail to go beyond piloting and experimentation and will be abandoned Gartner

Only 27% the executives surveyed described their data initiatives as successful Capgemini



65% of CEOs think their organisation is able to **interpret only a small proportion** of the information to which they have access The Economist



Minimize Equipment & Asset Failures

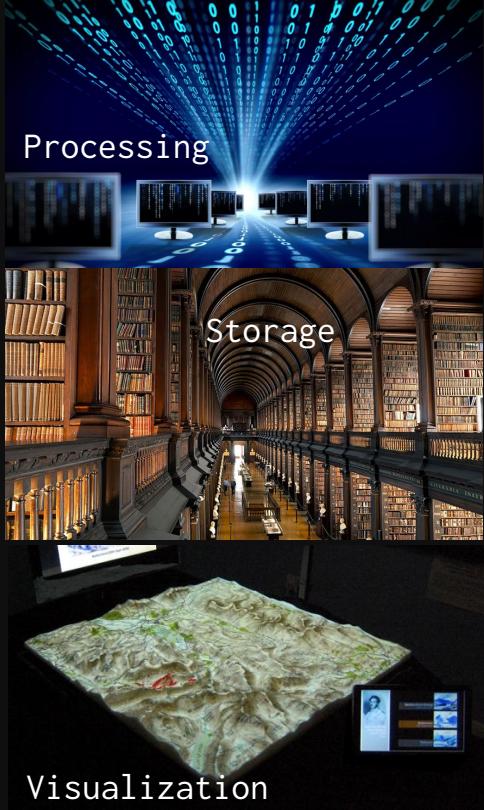


Leverage Customer Lifetime Value



How?

Many
competing
tools



How?

Many competing tools



How to describe a problem?

- To humans?
- Computers?

Best way?

- Some history

An Example

Machine Code in Hex	Assembly Code	High-Level Code
27B0B001	ldah gp, main	main()
23B0E004	ldah gp, main	
230EFFF0	lda sp, -16(sp)	int a, b, c;
A61D0018	ldq r16, 8(sp)	a = 3;
A77D0010	ldq r27, printf	b = 4;
47FF0000	mov r7, r27	c = a + b;
230E0000	stg r26, (sp)	printf("\n%d\n", c);
6B5B4000	jsr r26, printf	
27BA0001	ldah gp, main	
A75E0000	ldq r26, (sp)	
230E0000	ldah gp, main	
47FF0400	clr r26	
230E0010	lda sp, 16(sp)	
6BFAB001	ret r26	

How?

Many competing tools



How to describe a problem?

- A "paradigm"
- Expression of problem within the paradigm (hidden detail, syntax)

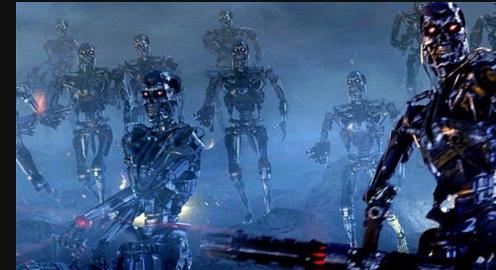
Simplification of expression

=

automatic concept translation by rules

Better for humans
vs.
better for machines

Some concepts can be automated easily resulting in efficient machine code. Some can't.



How?

Many competing tools



How to describe a problem?

For sequential processors
some standard
paradigms have "won".



Still significant variance in syntax and in the level of "simplified expression".

How?

Many competing tools



How to describe a problem?

Describing **non-sequential** processors -
hard for humans!!!

No simple & "complete" paradigm.

Paradigms make some tasks:

easy to describe , while other tasks hard/impossible .

easy to translate to efficient code , while other tasks hard/impossible .

- + syntax variation
- + level of abstraction



How?

Many competing tools

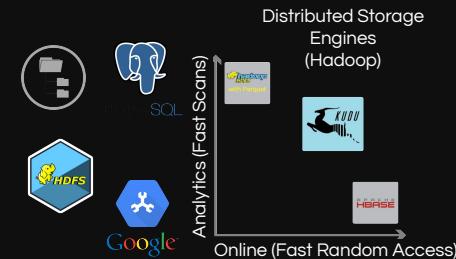


How to ~~describe~~ a problem?

...organise data in a physical space?

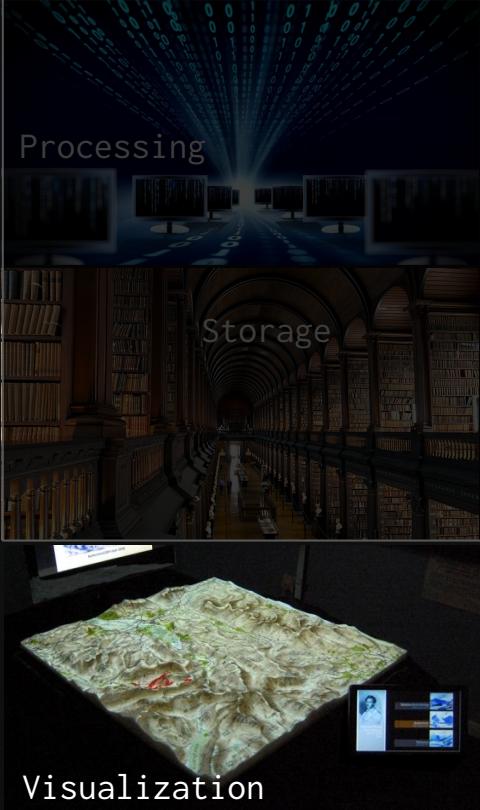
Trades speed of access
with
ease to access

→ too much of a trade-off,
not all processing viable



How?

Many competing tools

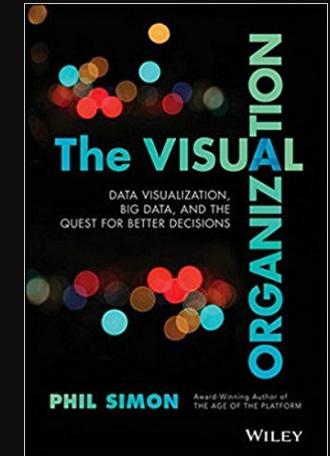


"Data visualization is the key to actionable insights "

Head of Business Intelligence and Data Analytics (AccuWeather)

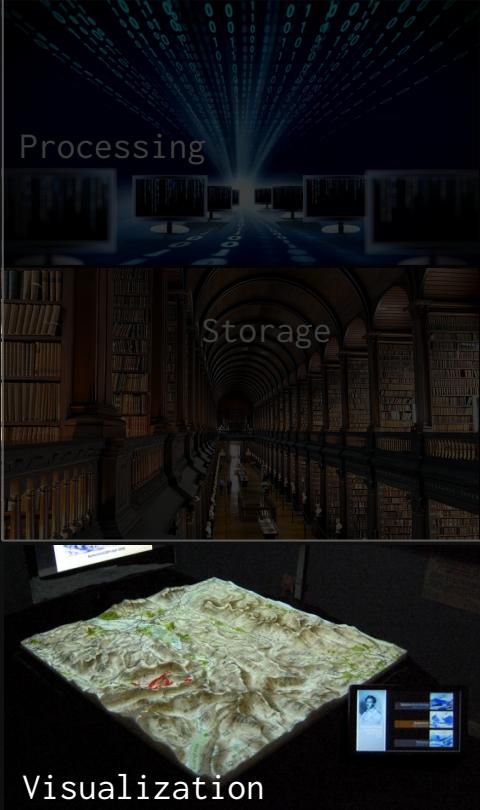
Wide range of visualizations.

Wide range of tools.



How?

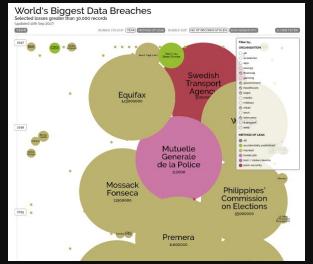
Many competing tools



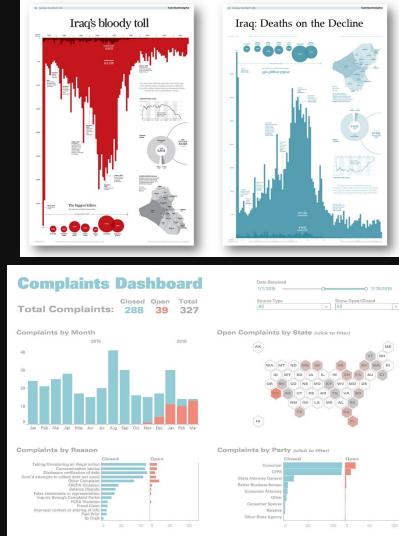
"Data visualization is the key to actionable insights "
Head of Business Intelligence and Data Analytics (AccuWeather)

Exploratory vs. Explanatory

Lots of types of visualizations



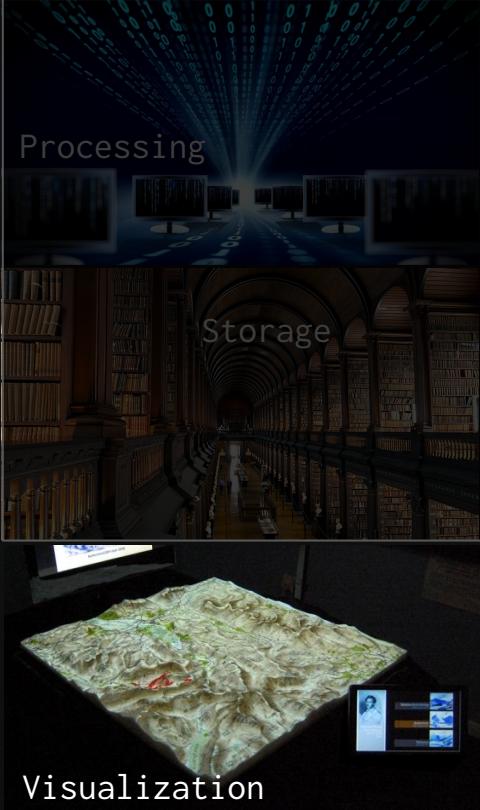
Simplicity, familiarity, interpretability, & expectations



World's Biggest Data Breaches Interactive Visualization:
<http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/>

How?

Many competing tools



"Data visualization is the key to actionable insights "
Head of Business Intelligence and Data Analytics (AccuWeather)

Exploratory vs. Explanatory

Leads to lots of tools/software

(desktop? web? mobile? print?)

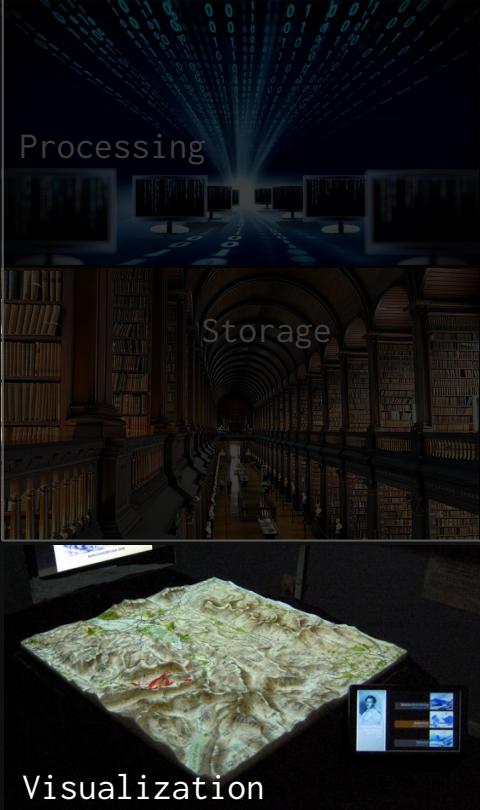


Simplicity, familiarity, interpretability, & expectations



How?

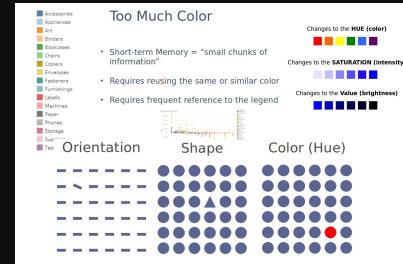
Many competing tools



"Data visualization is the key to actionable insights "
Head of Business Intelligence and Data Analytics (AccuWeather)

Exploratory vs. Explanatory

Design choices / concepts



The Duell's Rules for Actionable Visualizations

The question to answer must be identifiable

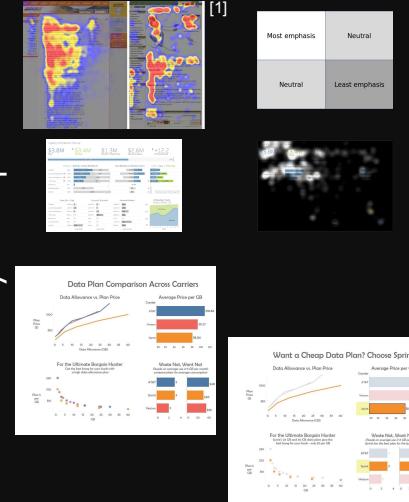
The data needed must be available

The visualization should be tailored to the person who will use the information

The story uncovered in the visualization should be evident

The action required should be clear

Simplicity, familiarity & expectations



[1] <https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/>

Data Management, Processing & Visualization*

	Normal Data	Big Data
Management (data storage)	Digital files Hierarchical databases Network databases Relational Databases     Microsoft SQL Server 	     
Processing	         	        
Visualization	  Many language specific packages/frameworks    + 	Other frameworks      

*Representative list

Module overview



NLAB: *Data at Scale*

Dr Evgeniya Lukinova

What is expected of you: Use Moodle!

Data at Scale: Management, Processing, Visualization (BUSI4369 UNUK) (AUT1 25-26)

Bulk actions 

Module Settings Participants Grades Reports More 

> General 



< Key Module Information 

Welcome to Data at Scale!

This course is a face-to-face course. Please see the important notes and resources below for more information.

Who: Dr Evgeniya Lukinova (module convener), Dr Georgiana Nica-Avram, and two additional teaching assistants.
What: See what we will be doing in this [module overview document](#).

When & Where:

Lecture & Practical sessions (2 hrs)

- Tuesday: 9am - 11am. Business School South B02, Jubilee Campus
- Thursday: 11am - 1pm. Business School South B02, Jubilee Campus

Support Sessions (sessions begin TBA)

Attendance is monitored via the QR code system. Please ensure **you scan the QR** for each session in [this document](#). The code is only active during the session. We will also take physical attendance via lists randomly during the semester.

Assessment: SQL Test (in class, computer based): 25%. Advanced SQL, Visualisation and Big data Test (in class, computer based): 25%. Coursework (1500 words + pitch deck): 50%.



Data at Scale

Dr Evgeniya Lukinova

What is expected
of you: Use Moodle!



Data at Scale: At a glance.

Note: Schedule is subject to change.

Lecturers: Dr. Evgeniya Lukinova (module convenor) and Dr Georgiana Nica-Avram

Week 1	<p>Session 1 (Lecture & Practical): The what and why of "data at scale". Introduction to the first concept of data storage and processing (<i>aka welcome to the course and the basics</i>).</p> <p>Session 2 (Lecture): Introduction to relational and object based paradigms for storing and processing data at scale (<i>aka why relational databases will be part of your future job as an analyst</i>).</p> <p>Session 2 Vis (Lecture): The theory behind making good visualisations (<i>aka how to visually manipulate people, for good.... or at least to help you get your message across</i>).</p>
Week 2	<p>Session 3 (Lecture): Graph types and dashboards, the when, where and how you should use them building on the theory behind making good visualisations previously discussed. Part 1 (<i>aka, let's make nice pictures that drive "actionable insights"</i>).</p> <p>Session 4 (Lecture & Practical): An in-depth look and how-to guide to using Tableau for visualisation (<i>aka, creating visualisations to drive your point home</i>).</p>
Week 3	<p>Session 5 (Lecture & Practical): SQL I: Foundations of relational databases in practice (<i>aka the basis of 20%+ of most data analytics jobs</i>).</p> <p>Session 6 (Lecture): The theory behind relational database design and how it affects you as an analyst even though database design is not your job. How to read the (often poor) documentation left by database designers for the database you will use (<i>aka, how to work with what you've been given in your job</i>).</p>



What is expected of you: Use Moodle

Attendance will be monitored
via QR codes...

Although the Google Doc
with QR codes is shared, we
will transition to live codes
with the new system SEAtS!

Assessments and Feedback

25% SQL Test (in-class, week 6). [open book, open web, no ChatGPT or similar, no collaboration with others]

25% Advanced SQL, Visualisation and Big Data Test (in-class, week 11). [closed book, 1 single-sided A4 page of handwritten notes allowed]

50% Coursework. Data analysis task based on a real-world based scenario and a data set (per group). 1500 word **individual** report and a **group** presentation.

Why 3 assessments? We opt for smaller, more regular assessment as continued study and feedback is the optimal way to learn the technical content (in this case SQL).

Deadline Date for Submission of Coursework

Due: 3PM, Thursday, 8th January 2026



Data at Scale

What is expected of you: Work hard.

This is a technical module in a
qualification .

You will learn and be certified to
know how to undertake data
analysis and visualization.

This takes time and practice,
**even after you understand
the higher level concepts** .



Data at Scale

What is expected of you: Work hard.

This is a technical module in
a **qualification** .

You will learn and be
certified to know how to
undertake data analysis
and visualization.

This takes time and
practice, **even after you
understand the higher
level concepts** .

Lectures

Guide you in what you need to
learn, providing material and
pointing to further resources.

Practicals

- Consolidate the technical
elements.
- Provide real-world business
use case examples.

What is expected of you: Work hard.

This is a technical module in a **qualification**.

You will learn and be certified to know how to undertake data analysis and visualization.

This takes time and practice, **even after you understand the higher level concepts**.



Lectures

Guide you in what you need to learn, providing material and pointing to further resources.

Practicals

- Consolidate the technical elements.
- Provide real-world business use case examples.

- Make sure you finish your practicals in your own time.
- Use the extra material if you are not confident with components of the course.
- Use the support session.
- Ask for help if you need it.

What is expected
of you: work hard.

Total learning time:
200 hours

Contact time:
44 - 55 hours [with support session]

Background study / coursework / revision:
145 - 156 hours

- Make sure you finish your practicals in your own time.
- Use the extra material if you are not confident with components of the course.
- Use the support session.
- Ask for help if you need it.

What is expected of you: Don't cheat.

Plagiarism means to pass off someone else's work, intentionally or unintentionally, as your own.

This might be by copying or paraphrasing someone's published or unpublished work without proper acknowledgment, or representing someone's artistic or technical work or creation as your own.



Data at Scale

University of Nottingham
UK | CHINA | MALAYSIA

Study Research Business Global About A-Z keyword(s)

[University of Nottingham](#) > [Studying effectively](#) > [Writing](#) > [Avoiding plagiarism](#)

Studying Effectively

Home
Studying at university
Types of teaching
Being organised
Reading and interpreting sources and data
Writing
Writing tasks at university
Strategies for writing
Referencing and citing
Avoiding plagiarism
Do you understand plagiarism
Preparing for assessment

Avoiding plagiarism

Plagiarism means to pass off someone else's work, intentionally or unintentionally, as your own.

This might be by copying or paraphrasing someone's published or unpublished work without proper acknowledgment, or representing someone's artistic or technical work or creation as your own.

The University's policy on plagiarism

An act of Academic Misconduct is, generally speaking, any action in which may give a student an unpermitted academic advantage; as such, it is not acceptable in a scholarly community. The most common examples of acts of Academic Misconduct are plagiarism, cheating in exams, collusion, and fabricating results or data. It can be, however, anything that gives you an unfair advantage in an assessment.

Incidences of plagiarism will first be addressed within the School, and they may apply penalties such as giving you a mark of zero for the piece of work concerned. The University's Academic Misconduct Committee has the power to apply a range of penalties for serious or repeated cases, including terminating your course.

Tips for avoiding plagiarism

- Don't just copy

Academic integrity

Quicklinks
[Test your skills](#)
▪ [Plagiarism quiz](#)

Further reading
[Studying at university](#)
▪ [Academic integrity and plagiarism](#)

Writing
▪ [Referencing and citing](#)

<https://www.nottingham.ac.uk/studyingeffectively/writing/plagiarism/index.aspx>

Dr Evgeniya Lukinova

What is expected of YOU: Don't cheat.

An act of Academic Misconduct is, generally speaking, **any action in which may give a student an unpermitted academic advantage**; as such, it is not acceptable in a scholarly community.

The most common examples of acts of Academic Misconduct are **plagiarism, cheating in exams, collusion, and fabricating results or data**. It can be, however, anything that gives you an unfair advantage in an assessment .



University of
Nottingham
UK | CHINA | MALAYSIA

Study Research Business Global About A-Z

keyword(s) 

[University of Nottingham](#) > [Studying effectively](#) > [Writing](#) > [Avoiding plagiarism](#)

Studying Effectively

- Home
- Studying at university
- Types of teaching
- Being organised
- Reading and interpreting sources and data
- Writing**
- Writing tasks at university
- Strategies for writing
- Referencing and citing
- Avoiding plagiarism**
- Do you understand plagiarism
- Preparing for assessment

Avoiding plagiarism

Plagiarism means to pass off someone else's work, intentionally or unintentionally, as your own.

This might be by copying or paraphrasing someone's published or unpublished work without proper acknowledgment, or representing someone's artistic or technical work or creation as your own.

The University's policy on plagiarism

An act of Academic Misconduct is, generally speaking, any action in which may give a student an unpermitted academic advantage; as such, it is not acceptable in a scholarly community. The most common examples of acts of Academic Misconduct are plagiarism, cheating in exams, collusion, and fabricating results or data. It can be, however, anything that gives you an unfair advantage in an assessment.

Incidences of plagiarism will first be addressed within the School, and they may apply penalties such as giving you a mark of zero for the piece of work concerned. The University's Academic Misconduct Committee has the power to apply a range of penalties for serious or repeated cases, including terminating your course.

Tips for avoiding plagiarism

- Don't just copy

Academic integrity

Quicklinks

Test your skills

- Plagiarism quiz

Further reading

Studying at university

- Academic integrity and plagiarism

Writing

- Referencing and citing

<https://www.nottingham.ac.uk/studyingeffectively/writing/plagiarism/index.aspx>



Data at Scale

Dr Evgeniya Lukinova

What is expected of YOU: Don't cheat.

DON'T DO IT.

The University takes it very seriously.
It protects the qualification you are
all trying to earn.

- Zeros. More pressure than before.
- End of course.
- Black marks on Record.
- Ruined references from us when
you apply for your job after.



University of
Nottingham
UK | CHINA | MALAYSIA

UK
China
Malaysia

Study Research Business Global About A-Z

[University of Nottingham](#) > [Studying effectively](#) > [Writing](#) > [Avoiding plagiarism](#)

Studying Effectively

Home

[Studying at university](#)

[Types of teaching](#)

[Being organised](#)

[Reading and interpreting
sources and data](#)

Writing

[Writing tasks at university](#)

[Strategies for writing](#)

[Referencing and citing](#)

[Avoiding plagiarism](#)

[Do you understand
plagiarism](#)

[Preparing for assessment](#)

Avoiding plagiarism

Plagiarism means to pass off someone else's work, intentionally or unintentionally, as your own.

This might be by copying or paraphrasing someone's published or unpublished work without proper acknowledgment, or representing someone's artistic or technical work or creation as your own.

The University's policy on plagiarism

An act of Academic Misconduct is, generally speaking, any action in which may give a student an unpermitted academic advantage; as such, it is not acceptable in a scholarly community. The most common examples of acts of Academic Misconduct are plagiarism, cheating in exams, collusion, and fabricating results or data. It can be, however, anything that gives you an unfair advantage in an assessment.

Incidences of plagiarism will first be addressed within the School, and they may apply penalties such as giving you a mark of zero for the piece of work concerned. The University's Academic Misconduct Committee has the power to apply a range of penalties for serious or repeated cases, including terminating your course.

[Tips for avoiding plagiarism](#)

- Don't just copy



Academic integrity

Quicklinks

Test your skills

- [Plagiarism quiz](#)

Further reading

Studying at university

- [Academic integrity and
plagiarism](#)

Writing

- [Referencing and citing](#)

<https://www.nottingham.ac.uk/studyingeffectively/writing/plagiarism/index.aspx>



Data at Scale

Dr Evgeniya Lukinova

What is expected of YOU: Don't cheat.

DON'T DO IT.

The University takes it very seriously.
It protects the qualification you are
all trying to earn.

- Zeros. More pressure than before.
- End of course.
- Black marks on Record.
- Ruined references from us when
you apply for your job after.



Data at Scale

University of Nottingham UK | CHINA | MALAYSIA

Study Research Business Global About A-Z keyword(s)

[University of Nottingham](#) > [Studying effectively](#) > [Writing](#) > [Avoiding plagiarism](#)

Studying Effectively

Home
Studying at university
Types of teaching
Being organised
Reading and interpreting sources and data
Writing
Writing tasks at university
Strategies for writing
Referencing and citing
Avoiding plagiarism
Do you understand plagiarism
Preparing for assessment
Tips for avoiding plagiarism

- Don't just copy

Avoiding plagiarism

Plagiarism means to pass off someone else's work, intentionally or unintentionally, as your own.

This might be by copying or paraphrasing someone's published or unpublished work without proper acknowledgment, or representing someone's artistic or technical work or creation as your own.

The University's policy on plagiarism

An act of Academic Misconduct is, generally speaking, any action in which may give a student an unpermitted academic advantage; as such, it is not acceptable in a scholarly community. The most common examples of acts of Academic Misconduct are plagiarism, cheating in exams, collusion, and fabricating results or data. It can be, however, anything that gives you an unfair advantage in an assessment.

Incidences of plagiarism will first be addressed within the School, and they may apply penalties such as giving you a mark of zero for the piece of work concerned. The University's Academic Misconduct Committee has the power to apply a range of penalties for serious or repeated cases, including terminating your course.

Academic integrity

Quicklinks

Test your skills

- Plagiarism quiz

Further reading

Studying at university

- Academic integrity and plagiarism

Writing

- Referencing and citing

<https://www.nottingham.ac.uk/studyingeffectively/writing/plagiarism/index.aspx>

Dr Evgeniya Lukinova

What is expected
of you: Don't cheat.

Quick note about AI (e.g., ChatGPT)

- The university considers the unauthorised use of AI tools false authorship.
- It will be clearly communicated to you whether you may use AI tools in assessment and how you are permitted to do so.
- **PLEASE ADHERE!**

<https://www.nottingham.ac.uk/currentstudents/news/using-ai-tools-in-your-studies>



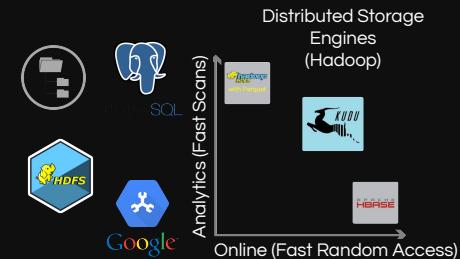
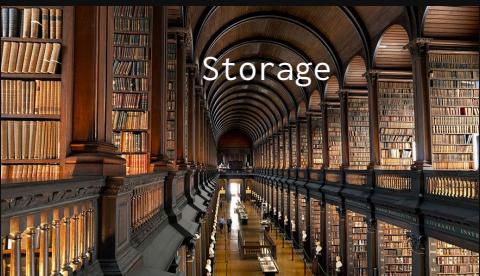
Data at Scale

Dr Evgeniya Lukinova

A gentle start...

How to organise data in physical space?

Data analytics starts
with loading data from
somewhere!

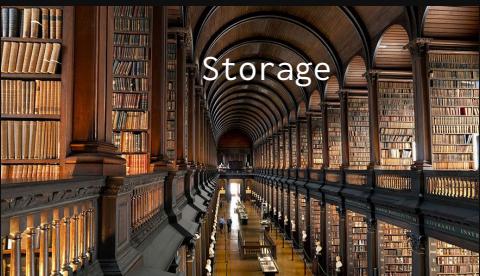


Data at Scale

Dr Evgeniya Lukinova

A gentle start...

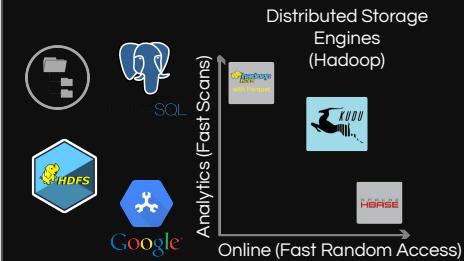
Data analytics starts
with loading data from
somewhere!



Option 1 (today):
→ Store data as
"objects" (**file**)
→ Store objects in a
hierarchy (**tree
structure**)



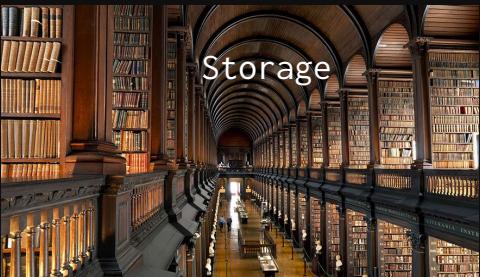
How to organise data
in physical space?



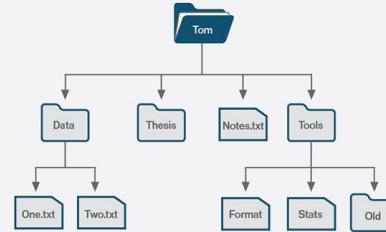
A gentle start...

How to organise data
in physical space?

Data analytics starts
with loading data from
somewhere!



Option 1 (today):
→ Filesystem of traditional
computers



A gentle start...

How to organise data
in physical space?

Data analytics starts
with loading data from
somewhere!



Option 1 (today):
→ Store data as
"objects" (**file**)
→ Store objects in a
hierarchy (**tree**
structure)



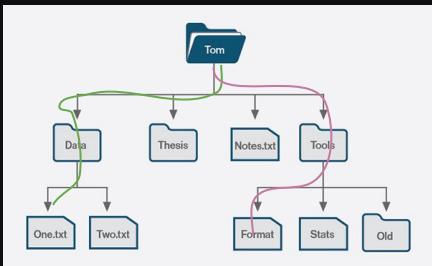
Benefits:

- + Store arbitrary things
- + Conceptually simple
- + Tree structure means single path locates all files

Negatives:

- Coming next, when we look at alternatives

Objects stored as trees.

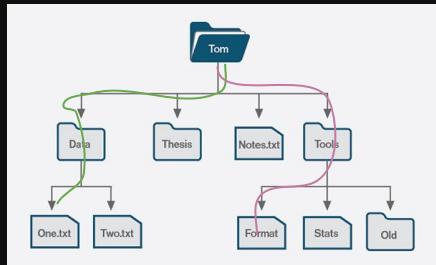


Each object's location can be defined by a path.

Tom/Data/One.txt

Tom/Tools/Format

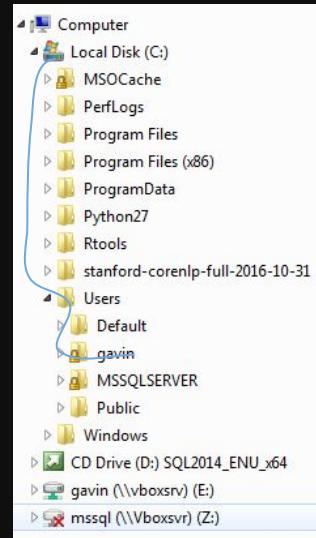
Objects stored as trees.



Each object's location can be defined by a path.

Tom/Data/One.txt

Tom/Tools/Format



C:\Users\gavin

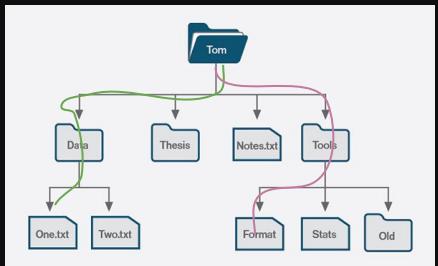


Data at Scale

Dr Evgeniya Lukinova



Objects stored as trees.

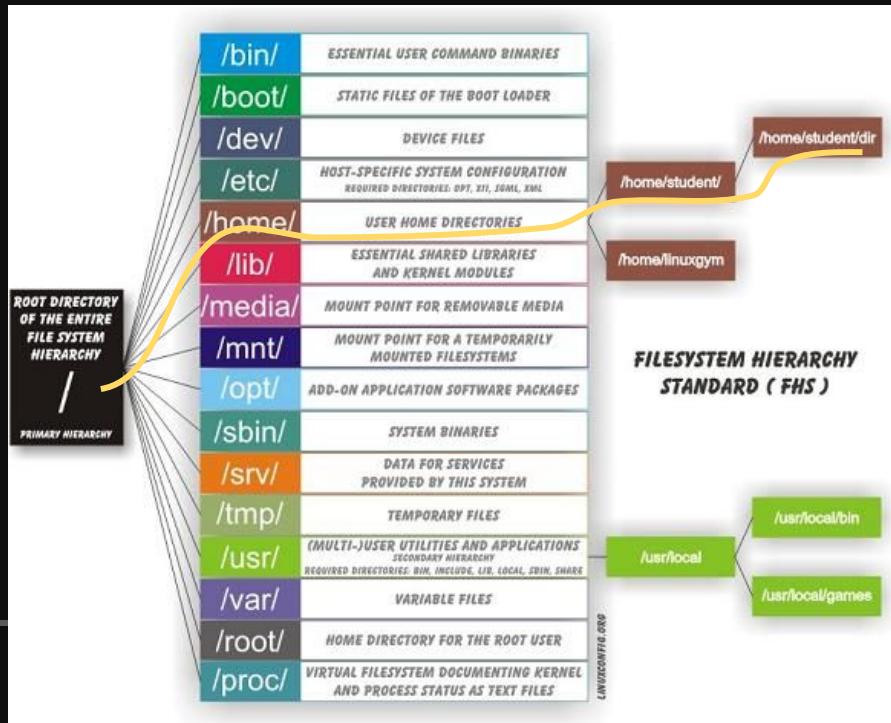


Each object's location can be defined by a path.

Tom/Data/One.txt

Tom/Tools/Format

/home/student/dir



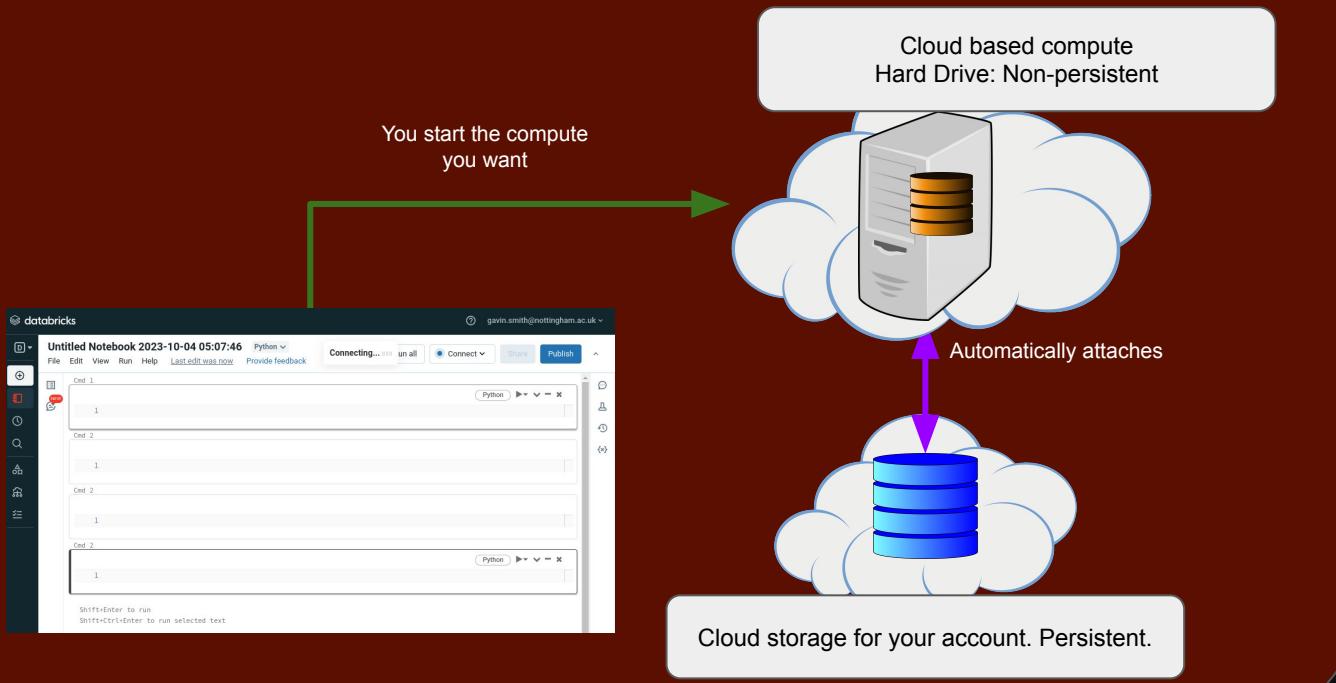
Data at Scale

Dr Evgeniya Lukinova

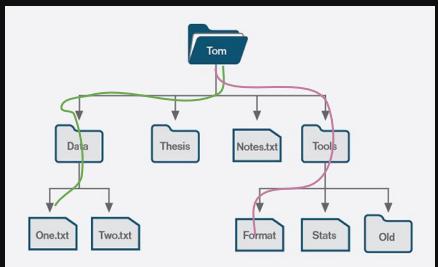


databricks

In cloud based systems we often have more than one "hierarchical object store" (filesystem).
... But Databricks Free Edition is restricted to one workspace, one metastore, one object store.



Objects stored as trees.



Temporary compute has a hard drive and runs a linux version.

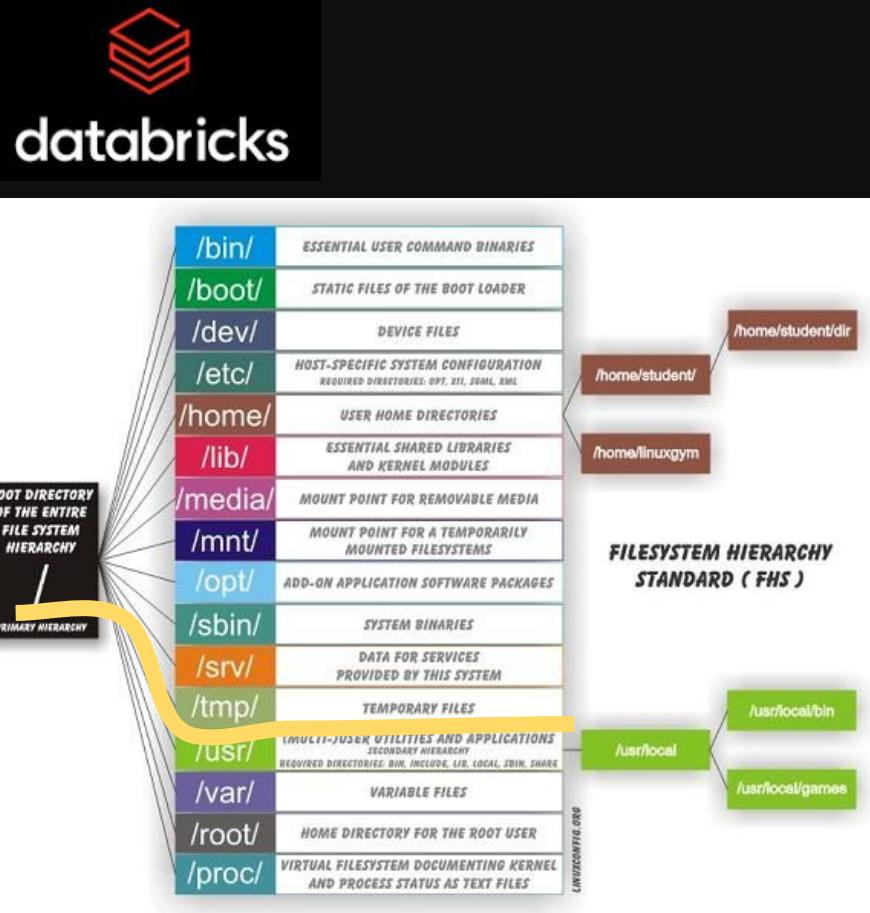
Each object's location can be defined by a path.

Can use the temporary folder if needed (**typically won't**).

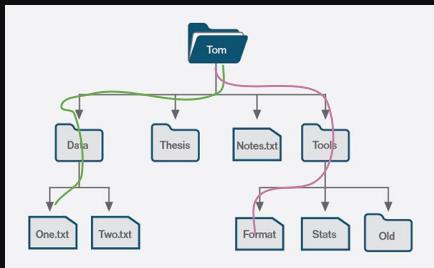
/tmp/

Tom/Data/One.txt

Tom/Tools/Format



Objects stored as trees.



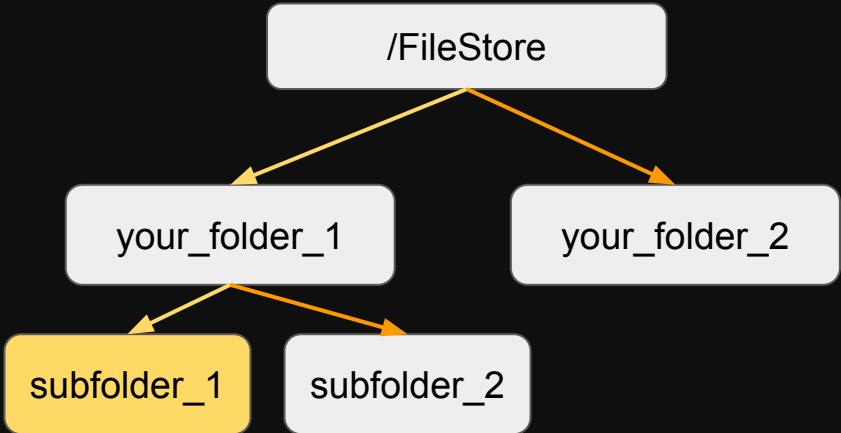
Persistent cloud filestore also stores objects as a tree.

Start with just a root node
"/Filestore"

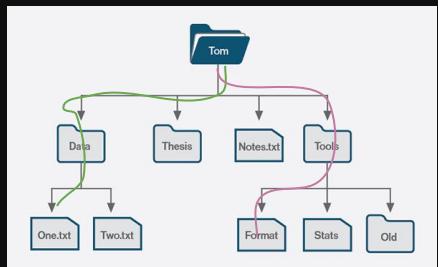
Tom/Data/One.txt

/Filestore/your_folder1/subfolder1

Tom/Tools/Format



Objects stored as trees.



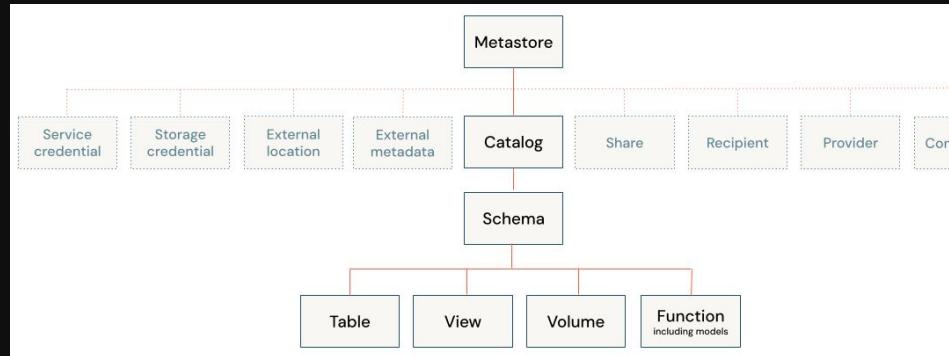
Persistent cloud storage also stores objects as a tree.

The root is the **Unity Catalog** metastore, under which you define catalogs, schemas, and tables.

Tom/Data/One.txt

catalog.schema.table

Tom/Tools/Format



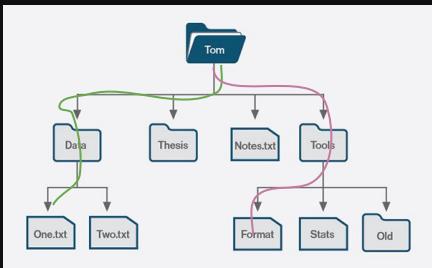
<https://docs.databricks.com/aws/en/data-governance/unity-catalog/>



Data at Scale

Dr Evgeniya Lukinova

Objects stored as trees.



Each object's location can be defined by a path.

Tom/Data/One.txt

Tom/Tools/Format

Why do I care?

To store / load data for analytics you need to know where it is.

Selecting files via a graphical interface doesn't scale.

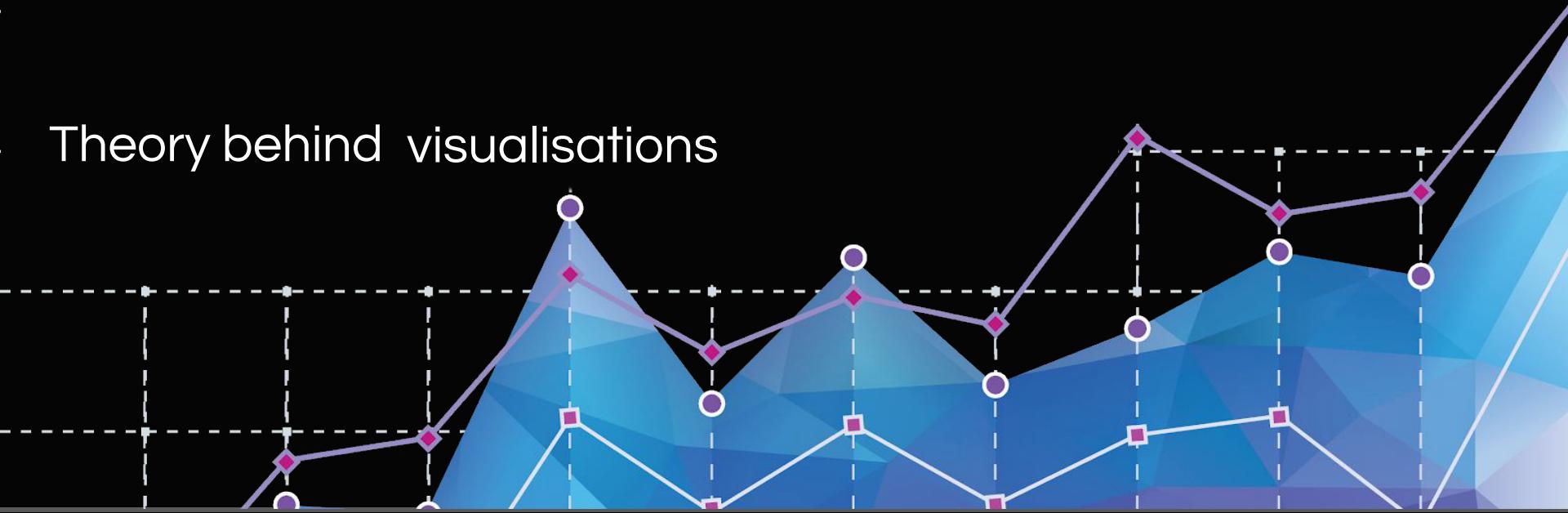
Graphical interfaces often cannot handle large data.... as we'll see in the practicals!

Try the Practical!



Session 2

Theory behind visualisations



Visualization basics & an introduction to Tableau



Data

Facts and statistics collected together for reference or analysis.



Data at Scale

Visualization basics & an introduction to Tableau



Data

Facts and statistics collected together for reference or analysis.

Information

Facts provided or learned about something or someone.



Data at Scale

Dr Evgeniya Lukinova

Visualization basics & an introduction to Tableau



Data

Facts and statistics collected together for reference or analysis.

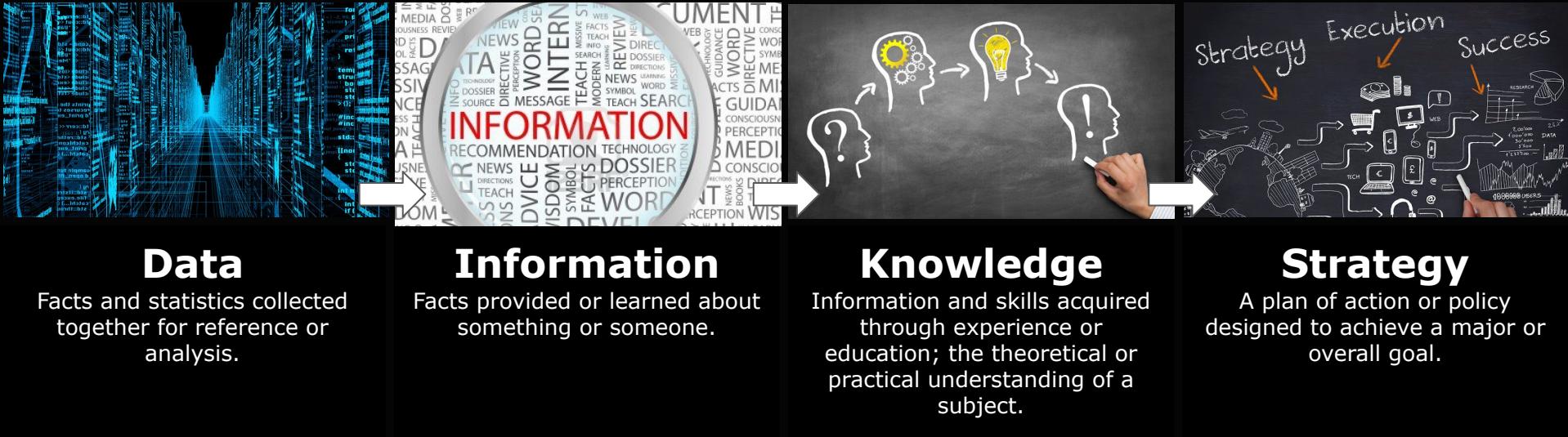
Information

Facts provided or learned about something or someone.

Knowledge

Information and skills acquired through experience or education; the theoretical or practical understanding of a subject.

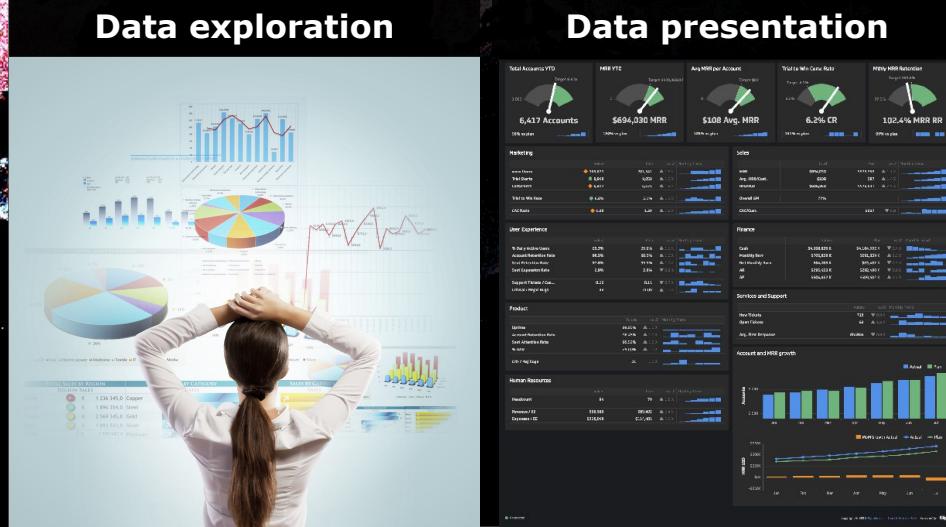
Visualization basics & an introduction to Tableau

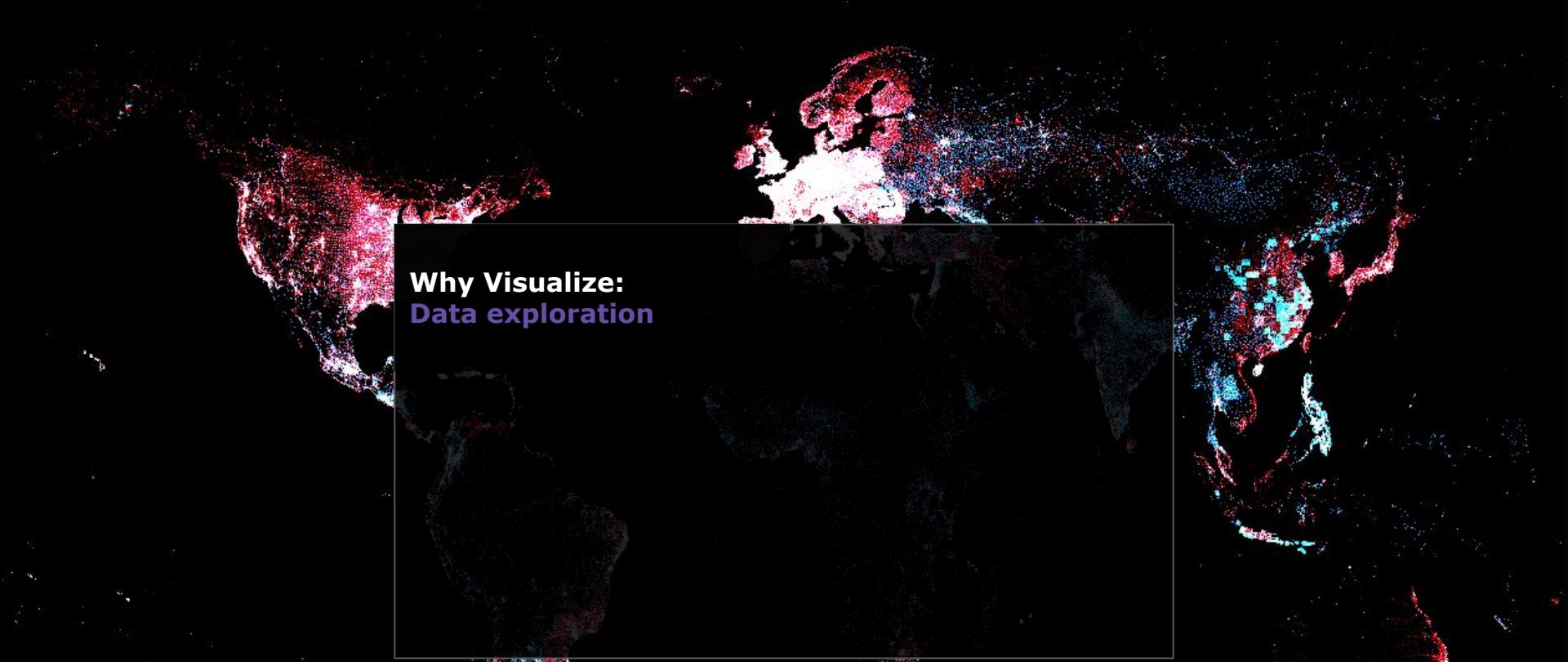


Data at Scale

Dr Evgeniya Lukinova

Visualization basics & an introduction to Tableau





**Why Visualize:
Data exploration**



Data at Scale

Dr Evgeniya Lukinova

Why Visualize: Data exploration

Let's look at some data...

Four data sets. Are they the same?

I		II		III		IV	
x	y	x	y	x	y	x	y
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.74	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.1	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.1	4	5.39	19	12.5
12	10.84	12	9.13	12	8.15	8	5.56
7	4.82	7	7.26	7	6.42	8	7.91
5	5.68	5	4.74	5	5.73	8	6.89



Data at Scale

Dr Evgeniya Lukinova

Why Visualize: Data exploration



Let's look at some data...
now with added stats!

Four data sets. Let's check!

I		II		III		IV	
x	y	x	y	x	y	x	y
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.74	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.1	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.1	4	5.39	19	12.5
12	10.84	12	9.13	12	8.15	8	5.56
7	4.82	7	7.26	7	6.42	8	7.91
5	5.68	5	4.74	5	5.73	8	6.89
9	--	9	--	9	--	9	--
11	--	11	--	11	--	11	--
--	7.5	--	7.5	--	7.5	--	7.5
--	4.122		4.122		4.122		4.122
0.816	0.816	0.816	0.816	0.816	0.816	0.816	0.816
$y = 3 + 0.5x$							

Mean of x
Variance of x
Mean of y
Variance of y
Correlation between x and y
Linear regression line



Data at Scale

Why Visualize: Data exploration

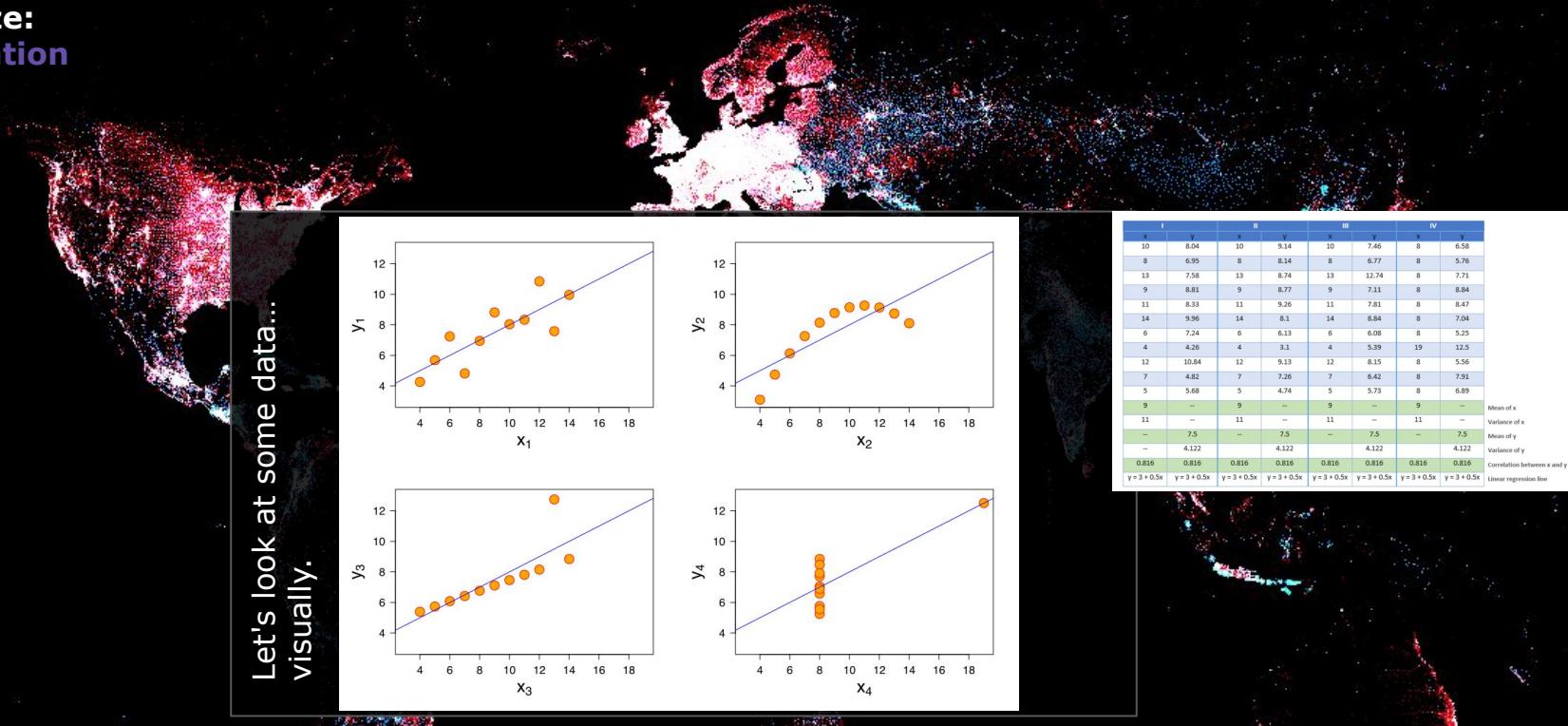


Image source Wikipedia. CC BY-SA 3.0. Anscombe's quartet.

Why Visualize: Data exploration



Data at Scale

Why Visualize: Data exploration

So

summary stats < visualization??



Data at Scale

Why Visualize: Data exploration

So

summary stats < visualization??



→ Compress data to focus
on one data property

→ for the given summary
statistic the whole data set
(high dimensional or not) is
correctly represented (by
definition)

Why Visualize: Data exploration

→ need to choose
the right summary
statistic

So
summary stats < visualization??



→ Compress data to focus
on one data property

→ for the given summary
statistic the whole data set
(high dimensional or not) is
correctly represented (by
definition)

Why Visualize: Data exploration

→ need to choose
the right summary
statistic

So
summary stats < visualization??

→ Compress data to focus
on one data property

→ for the given summary
statistic the whole data set
(high dimensional or not) is
correctly represented (by
definition)

→ Displays a range of data
dimensions

→ cannot accurately show
high-dimensional and/or
large data sets (often must
visualize summary
statistics)

Why Visualize: Data exploration

→ need to choose
the right summary
statistic

So

summary stats < visualization??

→ Compress data to focus
on one data property

→ for the given summary
statistic the whole data set
(high dimensional or not) is
correctly represented (by
definition)

→ Displays a range of data
dimensions

→ cannot accurately show
high-dimensional and/or
large data sets (often must
visualize summary
statistics)

→ need to choose the
dimensions (features/attributes)
to visualize

Why Visualize: Data exploration

→ need to choose
the right summary
statistic

So

summary stats < visualization??

→ Compress data to focus
on one data property

→ for the given summary
statistic the whole data set
(high dimensional or not) is
correctly represented (by
definition)

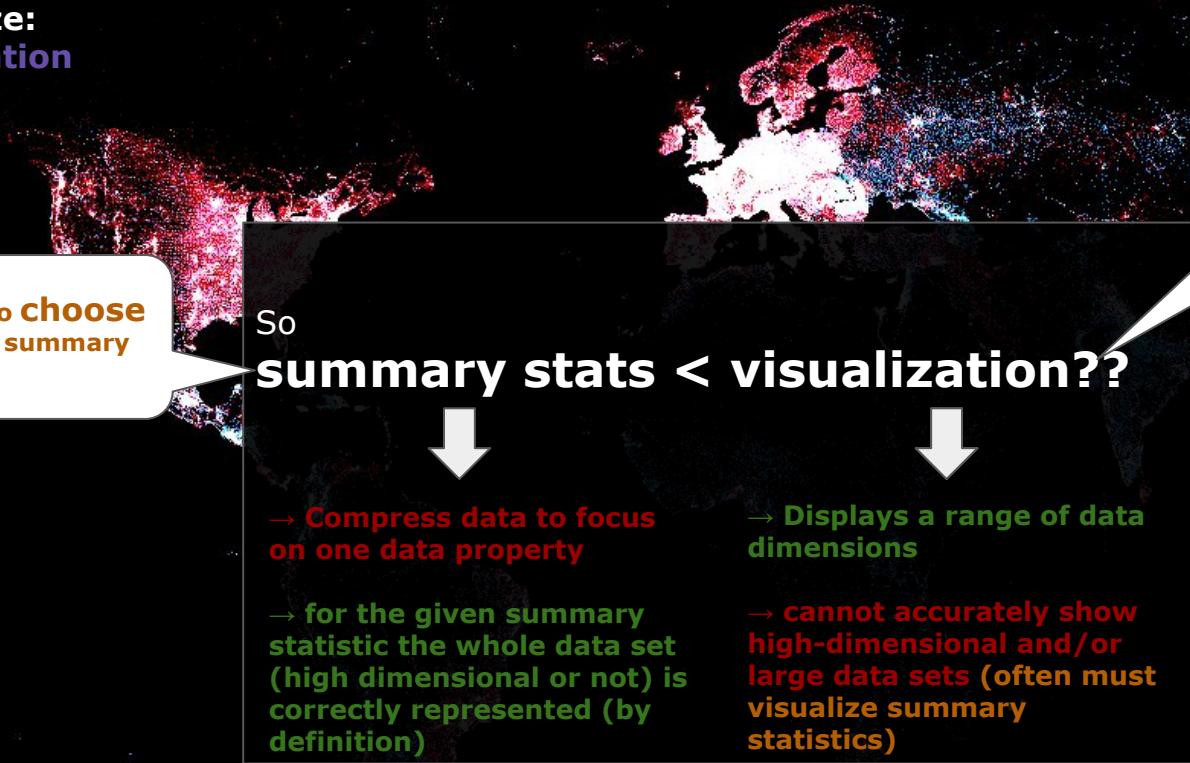
→ Displays a range of data
dimensions

→ cannot accurately show
high-dimensional and/or
large data sets (often must
visualize summary
statistics)

→ need to choose the
dimensions (features/attributes)
to visualize

→ need to choose the
right visualization
(highlights different properties of the
dataset ≈ summary stats)

Why Visualize: Data exploration



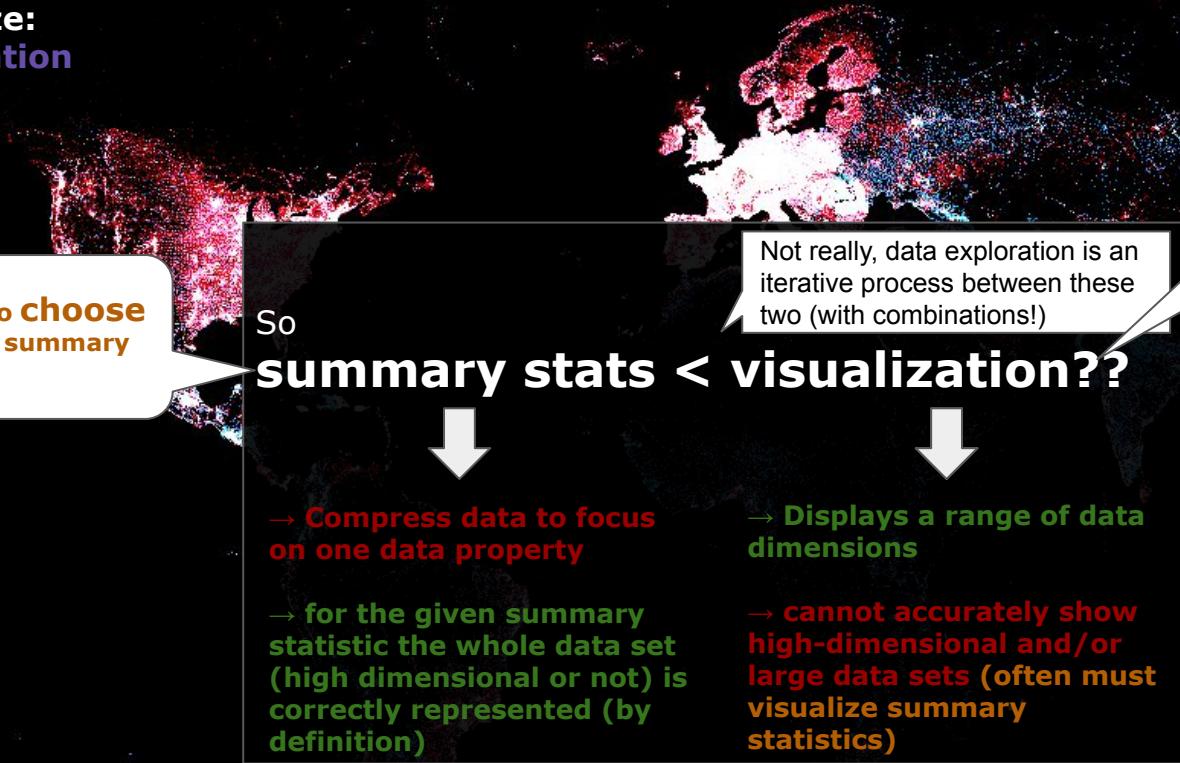
→ need to choose the dimensions (features/attributes) to visualize

→ need to choose the right visualization
(highlights different properties of the dataset ≈ summary stats)

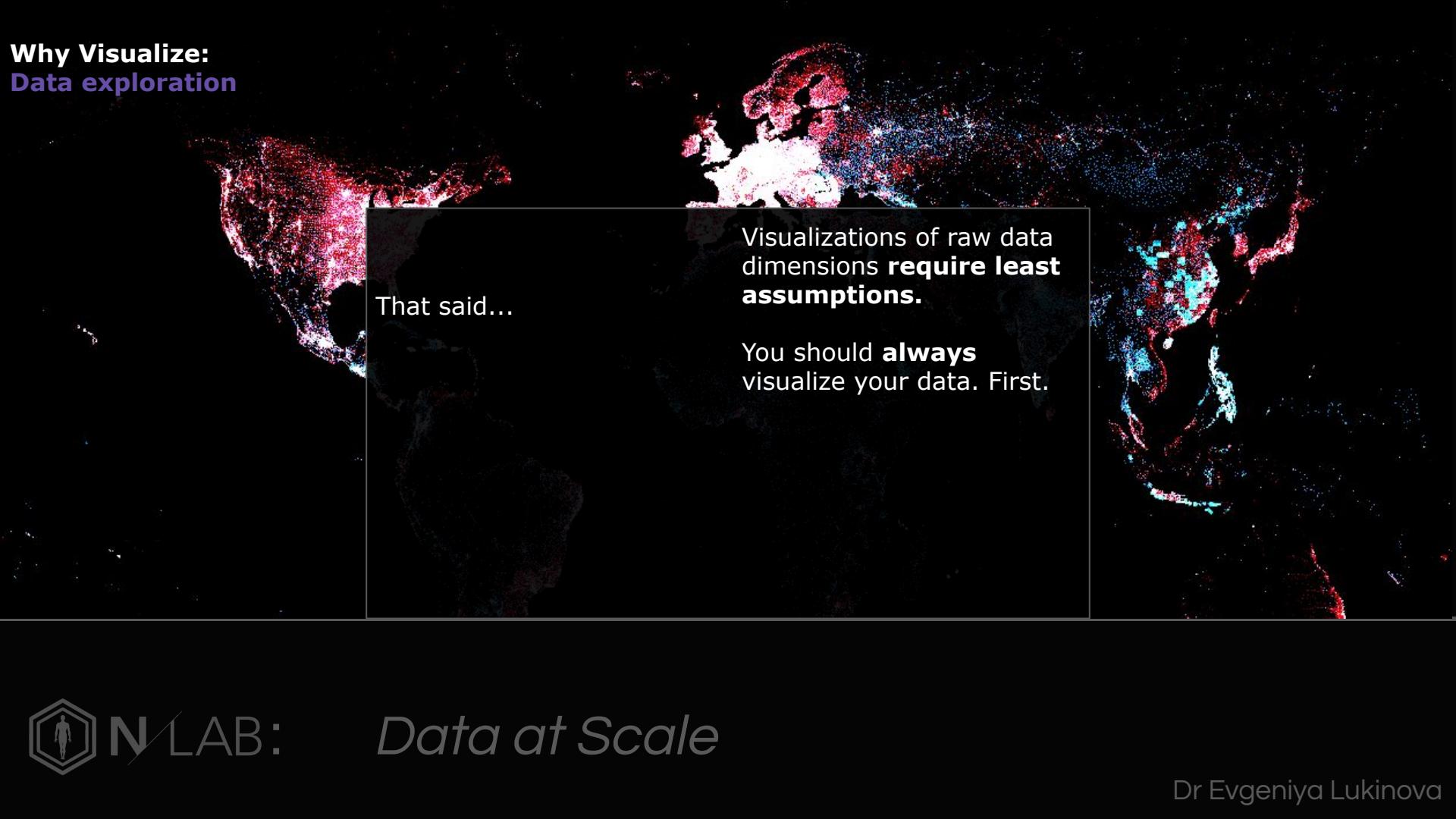
→ Can't visualize every aspect at once: both summary stats & visualizations compress the data for interpretation.

→ Eventually for presentation you'll be deciding on the "best" compression (for your story)

Why Visualize: Data exploration



Why Visualize: Data exploration



That said...

Visualizations of raw data dimensions **require least assumptions.**

You should **always** visualize your data. First.



Data at Scale

Why Visualize: Data exploration



That said...

Visualizations of raw data dimensions **require least assumptions.**

You should **always** visualize your data. First.

Visualization can play a **critical role** in helping you figure out what the **interesting questions** are.



Data at Scale

Dr Evgeniya Lukinova



Figure from: Tor Norretranders' The User Illusion. Visualisation from Stephen Few's Information Dashboard Design



Data at Scale

Dr Evgeniya Lukinova

Why Visualize: Data presentation

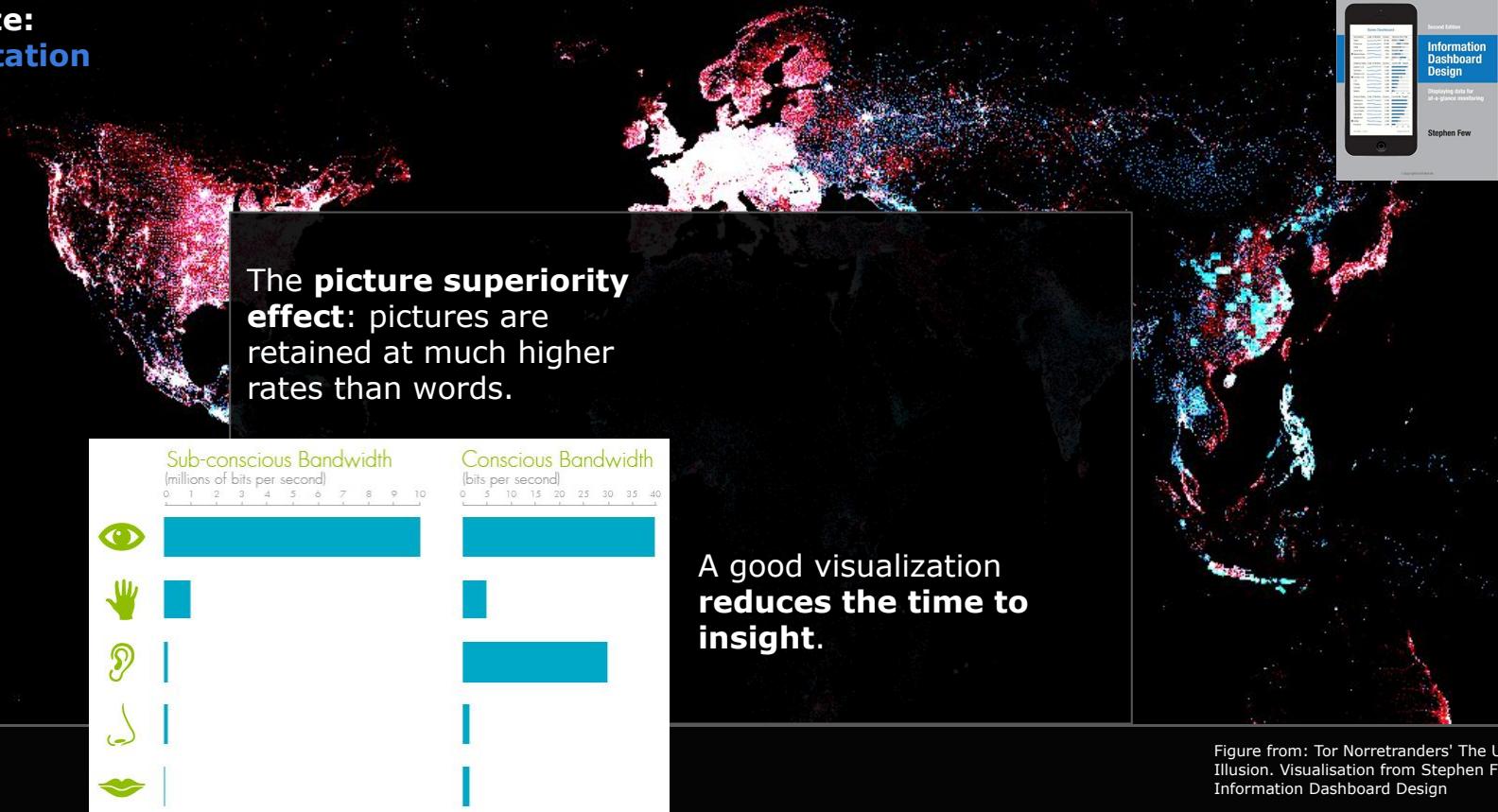


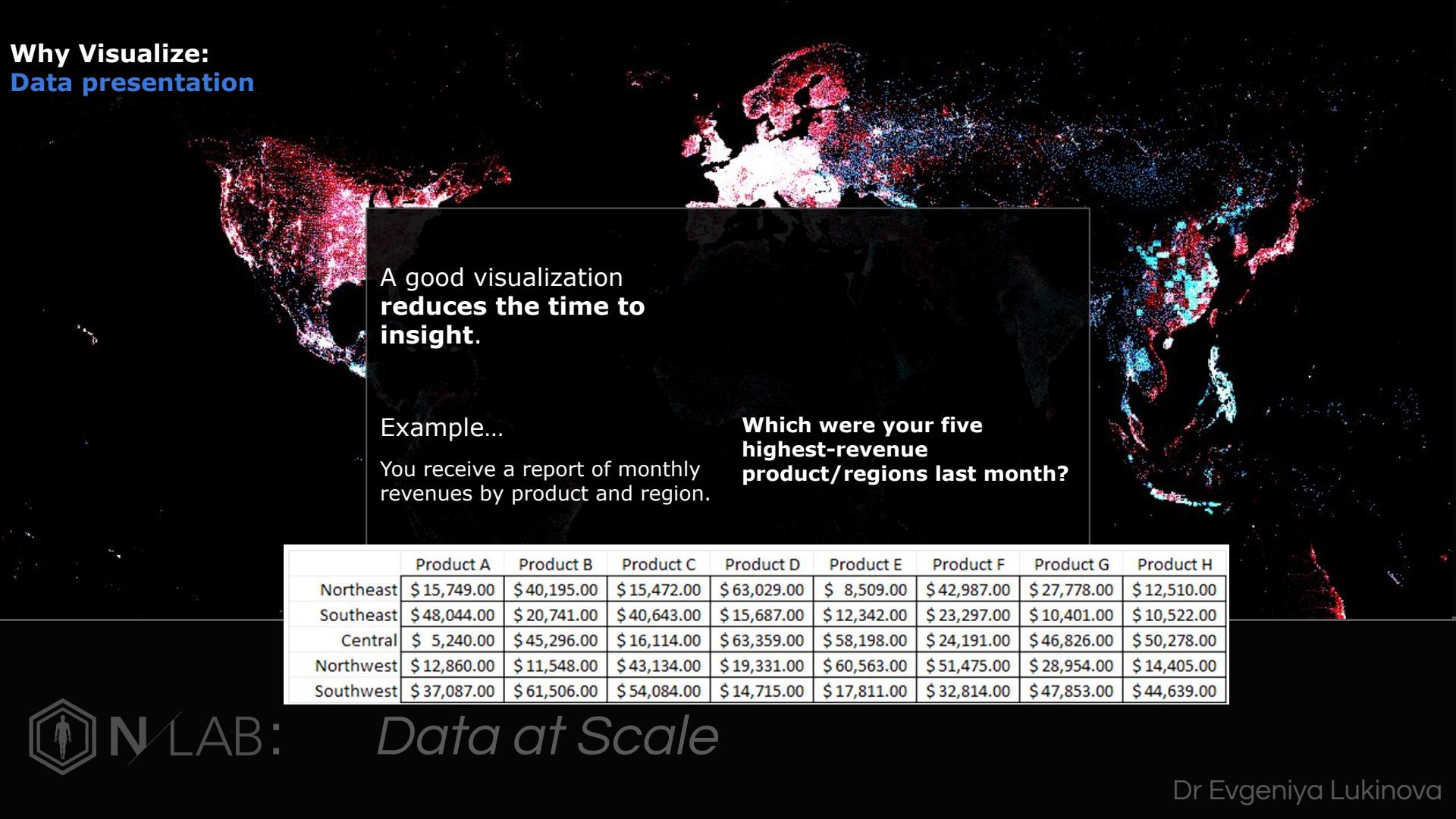
Figure from: Tor Norretranders' The User Illusion. Visualisation from Stephen Few's Information Dashboard Design



Data at Scale

Dr Evgeniya Lukinova

Why Visualize: Data presentation



A good visualization
**reduces the time to
insight.**

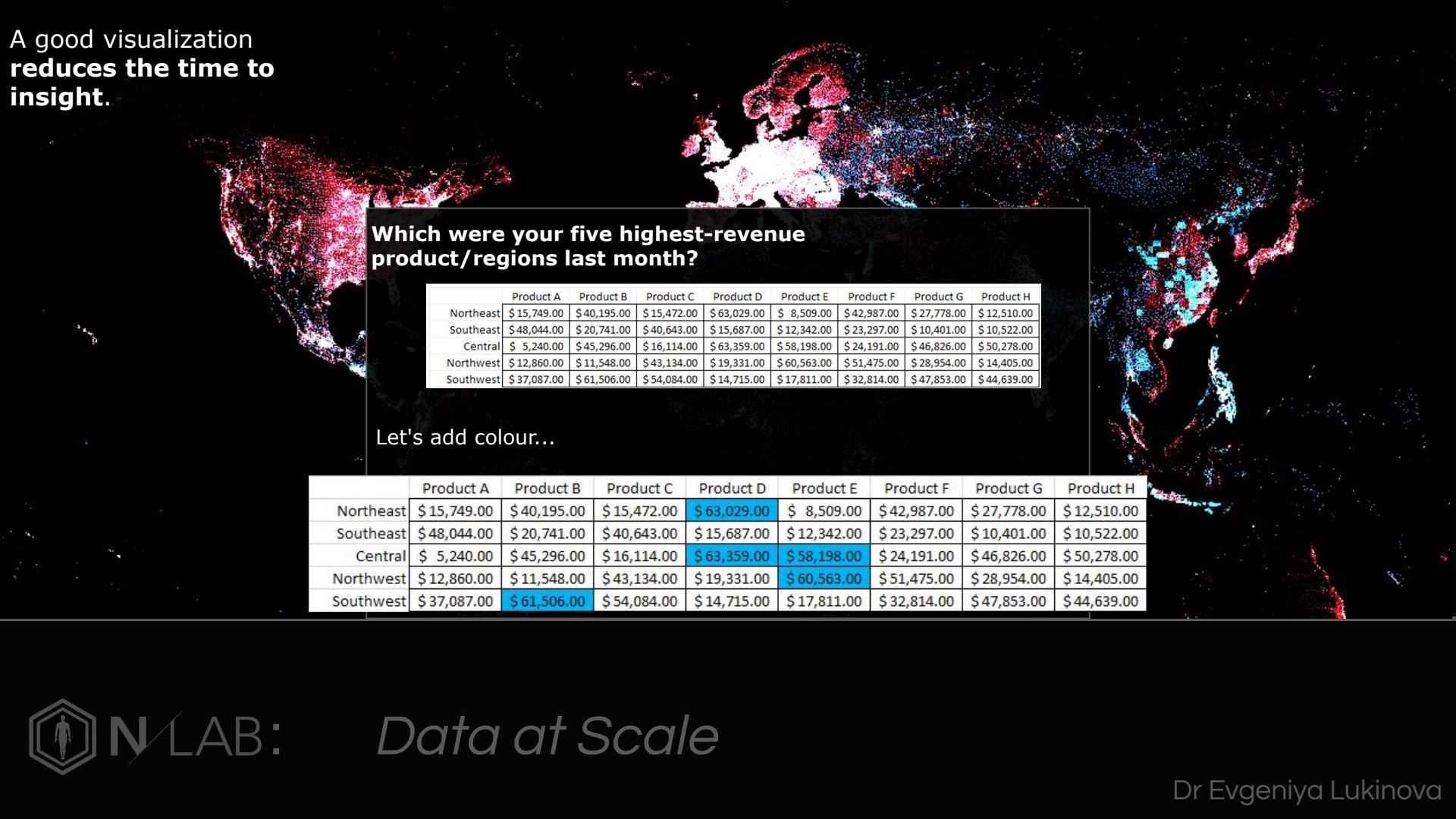
Example...

You receive a report of monthly revenues by product and region.

**Which were your five
highest-revenue
product/regions last month?**

	Product A	Product B	Product C	Product D	Product E	Product F	Product G	Product H
Northeast	\$ 15,749.00	\$ 40,195.00	\$ 15,472.00	\$ 63,029.00	\$ 8,509.00	\$ 42,987.00	\$ 27,778.00	\$ 12,510.00
Southeast	\$ 48,044.00	\$ 20,741.00	\$ 40,643.00	\$ 15,687.00	\$ 12,342.00	\$ 23,297.00	\$ 10,401.00	\$ 10,522.00
Central	\$ 5,240.00	\$ 45,296.00	\$ 16,114.00	\$ 63,359.00	\$ 58,198.00	\$ 24,191.00	\$ 46,826.00	\$ 50,278.00
Northwest	\$ 12,860.00	\$ 11,548.00	\$ 43,134.00	\$ 19,331.00	\$ 60,563.00	\$ 51,475.00	\$ 28,954.00	\$ 14,405.00
Southwest	\$ 37,087.00	\$ 61,506.00	\$ 54,084.00	\$ 14,715.00	\$ 17,811.00	\$ 32,814.00	\$ 47,853.00	\$ 44,639.00

A good visualization
reduces the time to
insight.



Which were your five highest-revenue product/regions last month?

	Product A	Product B	Product C	Product D	Product E	Product F	Product G	Product H
Northeast	\$ 15,749.00	\$ 40,195.00	\$ 15,472.00	\$ 63,029.00	\$ 8,509.00	\$ 42,987.00	\$ 27,778.00	\$ 12,510.00
Southeast	\$ 48,044.00	\$ 20,741.00	\$ 40,643.00	\$ 15,687.00	\$ 12,342.00	\$ 23,297.00	\$ 10,401.00	\$ 10,522.00
Central	\$ 5,240.00	\$ 45,296.00	\$ 16,114.00	\$ 63,359.00	\$ 58,198.00	\$ 24,191.00	\$ 46,826.00	\$ 50,278.00
Northwest	\$ 12,860.00	\$ 11,548.00	\$ 43,134.00	\$ 19,331.00	\$ 60,563.00	\$ 51,475.00	\$ 28,954.00	\$ 14,405.00
Southwest	\$ 37,087.00	\$ 61,506.00	\$ 54,084.00	\$ 14,715.00	\$ 17,811.00	\$ 32,814.00	\$ 47,853.00	\$ 44,639.00

Let's add colour...

	Product A	Product B	Product C	Product D	Product E	Product F	Product G	Product H
Northeast	\$ 15,749.00	\$ 40,195.00	\$ 15,472.00	\$ 63,029.00	\$ 8,509.00	\$ 42,987.00	\$ 27,778.00	\$ 12,510.00
Southeast	\$ 48,044.00	\$ 20,741.00	\$ 40,643.00	\$ 15,687.00	\$ 12,342.00	\$ 23,297.00	\$ 10,401.00	\$ 10,522.00
Central	\$ 5,240.00	\$ 45,296.00	\$ 16,114.00	\$ 63,359.00	\$ 58,198.00	\$ 24,191.00	\$ 46,826.00	\$ 50,278.00
Northwest	\$ 12,860.00	\$ 11,548.00	\$ 43,134.00	\$ 19,331.00	\$ 60,563.00	\$ 51,475.00	\$ 28,954.00	\$ 14,405.00
Southwest	\$ 37,087.00	\$ 61,506.00	\$ 54,084.00	\$ 14,715.00	\$ 17,811.00	\$ 32,814.00	\$ 47,853.00	\$ 44,639.00

A good visualization
reduces the time to
insight.

Which were your five highest-revenue
product/regions last month?

	Product A	Product B	Product C	Product D	Product E	Product F	Product G	Product H
Northeast	\$ 15,749.00	\$ 40,195.00	\$ 15,472.00	\$ 63,029.00	\$ 8,509.00	\$ 42,987.00	\$ 27,778.00	\$ 12,510.00
Southeast	\$ 48,044.00	\$ 20,741.00	\$ 40,643.00	\$ 15,687.00	\$ 12,342.00	\$ 23,297.00	\$ 10,401.00	\$ 10,522.00
Central	\$ 5,240.00	\$ 45,296.00	\$ 16,114.00	\$ 63,359.00	\$ 58,198.00	\$ 24,191.00	\$ 46,826.00	\$ 50,278.00
Northwest	\$ 12,860.00	\$ 11,548.00	\$ 43,134.00	\$ 19,331.00	\$ 60,563.00	\$ 51,475.00	\$ 28,954.00	\$ 14,405.00
Southwest	\$ 37,087.00	\$ 61,506.00	\$ 54,084.00	\$ 14,715.00	\$ 17,811.00	\$ 32,814.00	\$ 47,853.00	\$ 44,639.00

Let's add colour...

	Product A	Product B	Product C	Product D	Product E	Product F	Product G	Product H
Northeast	\$ 15,749.00	\$ 40,195.00	\$ 15,472.00	\$ 63,029.00	\$ 8,509.00	\$ 42,987.00	\$ 27,778.00	\$ 12,510.00
Southeast	\$ 48,044.00	\$ 20,741.00	\$ 40,643.00	\$ 15,687.00	\$ 12,342.00	\$ 23,297.00	\$ 10,401.00	\$ 10,522.00
Central	\$ 5,240.00	\$ 45,296.00	\$ 16,114.00	\$ 63,359.00	\$ 58,198.00	\$ 24,191.00	\$ 46,826.00	\$ 50,278.00
Northwest	\$ 12,860.00	\$ 11,548.00	\$ 43,134.00	\$ 19,331.00	\$ 60,563.00	\$ 51,475.00	\$ 28,954.00	\$ 14,405.00
Southwest	\$ 37,087.00	\$ 61,506.00	\$ 54,084.00	\$ 14,715.00	\$ 17,811.00	\$ 32,814.00	\$ 47,853.00	\$ 44,639.00

Color automatically
focused your brain –
no mental calculations /
comparisons needed!

A good visualization
reduces the time to
insight.

Which were your five highest-revenue
product/regions last month?

	Product A	Product B	Product C	Product D	Product E	Product F	Product G	Product H
Northeast	\$ 15,749.00	\$ 40,195.00	\$ 15,472.00	\$ 63,029.00	\$ 8,509.00	\$ 42,987.00	\$ 27,778.00	\$ 12,510.00
Southeast	\$ 48,044.00	\$ 20,741.00	\$ 40,643.00	\$ 15,687.00	\$ 12,342.00	\$ 23,297.00	\$ 10,401.00	\$ 10,522.00
Central	\$ 5,240.00	\$ 45,296.00	\$ 16,114.00	\$ 63,359.00	\$ 58,198.00	\$ 24,191.00	\$ 46,826.00	\$ 50,278.00
Northwest	\$ 12,860.00	\$ 11,548.00	\$ 43,134.00	\$ 19,331.00	\$ 60,563.00	\$ 51,475.00	\$ 28,954.00	\$ 14,405.00
Southwest	\$ 37,087.00	\$ 61,506.00	\$ 54,084.00	\$ 14,715.00	\$ 17,811.00	\$ 32,814.00	\$ 47,853.00	\$ 44,639.00

For **some applications**, eliminating all distractions
with a very simple viz can be effective:

	Product A	Product B	Product C	Product D	Product E	Product F	Product G	Product H
Northeast								
Southeast								
Central								
Northwest								
Southwest								

1) However, removing the base data removes
information for reader.

2) Now the info is gone
perhaps a list would simply
be better?

A good visualization
reduces the time to
insight.

How many nines are there...

4	7	7	5	5	2	7	4	7	1
4	9	2	5	7	7	2	6	1	7
1	7	6	9	3	4	7	5	1	2
5	1	6	3	3	8	4	8	6	6
6	5	6	4	9	3	8	9	1	9
3	8	1	5	2	2	3	6	3	9
4	6	4	5	6	3	7	7	9	1
9	1	3	3	6	1	3	3	1	8
8	1	1	8	7	5	8	1	7	4
3	6	9	2	8	9	3	7	5	7
4	4	4	2	8	2	2	9	2	8

A good visualization
reduces the time to
insight.

How many nines are there...

4	7	7	5	5	2	7	4	7	1
4	9	2	5	7	7	2	6	1	7
1	7	6	9	3	4	7	5	1	2
5	1	6	3	3	8	4	8	6	6
6	5	6	4	9	3	8	9	1	9
3	8	1	5	2	2	3	6	3	9
4	6	4	5	6	3	7	7	9	1
9	1	3	3	6	1	3	3	1	8
8	1	1	8	7	5	8	1	7	4
3	6	9	2	8	9	3	7	5	7
4	4	4	2	8	2	2	9	2	8



Data at Scale

Dr Evgeniya Lukinova

A good visualization
reduces the time to
insight.



So colour can be used to
highlight data and reduce time
to insight.

While providing the same data.
Guides interpretation.

But....

A good visualization
reduces the time to
insight.

Using colour must be done with care...

"Color used well can enhance and beautify, but color used poorly can be worse than no color at all." Maureen Stone

"...avoiding catastrophe becomes the first principle in bringing color to information: Above all, do no harm."
(Envisioning Information, Edward Tufte, Graphics Press, 1990)

A good visualization
reduces the time to
insight.

Using colour must be done with care...

- ~ 8% of men worldwide are colour blind
 - Be nice, **avoid red/green palettes**
 - Blue/orange is a good alternative

A good visualization
reduces the time to
insight.

Using colour must be done with care...

Not a natural designer? Use "known good" colour ramps.

E.g. In python, see [seaborn](#).
For printing/white backgrounds:
[colorbrewer2.org](#)

- For continuous data, **colour ramps** are effective
- For discrete data, always try to limit colours (as a guide think 8 max)

A good visualization
reduces the time to
insight.

Using colour must be done with care...

The use of too many colors makes it hard to distinguish, and also requires frequent referencing of the legend.

Limit yourself to just a few colors so your audience can actually remember what's what.
Be nice :)

- For continuous data, **colour ramps** are effective
- For discrete data, always try to limit colours (as a guide think 8 max)

Not a natural designer? Use "known good" colour ramps.

E.g. In python, see [seaborn](#).
For printing/white backgrounds:
[colorbrewer2.org](#)

A good visualization
reduces the time to
insight.

Using colour must be done with care...

The use of too many colors makes it hard to distinguish, and also requires frequent referencing of the legend.

Limit yourself to just a few colors so your audience can actually remember what's what.
Be nice :)

- For continuous data, **colour ramps** are effective
- For discrete data, always try to limit colours (as a guide think 8 max)

Not a natural designer? Use "known good" colour ramps.

E.g. In python, see [seaborn](#).
For printing/white backgrounds:
[colorbrewer2.org](#)

But the amount of colours and type is quite variable depending on context... i.e. are the size of the shapes varying?
[c.f. "In Color Perception, Size Matters" Maureen Stone 2012]

A good visualization
reduces the time to
insight.

Some more on colour

- Colour has the potential to communicate meaning:
(more recent work on the communication of emotion [1])
- Bright colours should be reserved for small highlight areas and almost never used as backgrounds



- Colour is relative, not absolute!

[1] Lyn Bartram, Abhishek Patra, and Maureen Stone. 2017. Affective Color in Visualization. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 1364-1374. DOI: <https://doi.org/10.1145/3025453.3026041>

A good visualization
reduces the time to
insight.

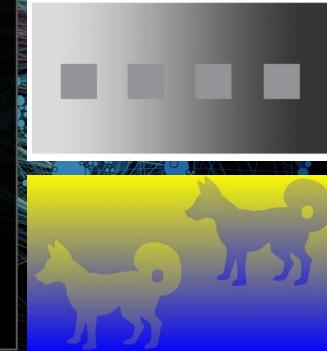
Some more on colour

- Colour has the potential to communicate meaning:
(more recent work on the communication of emotion [1])



- Bright colours should be reserved for small highlight areas and almost never used as backgrounds

- Colour is relative, not absolute!
The squares are the same shade of gray.



[1] Lyn Bartram, Abhishek Patra, and Maureen Stone. 2017. Affective Color in Visualization. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 1364-1374. DOI: <https://doi.org/10.1145/3025453.3026041>

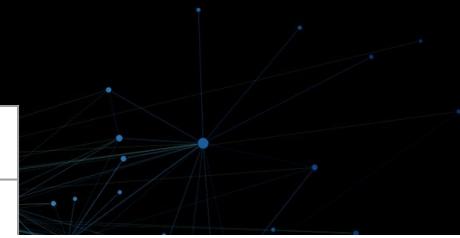
A good visualization
reduces the time to
insight.

A final bit on colour...

When using colour in graphs there are three main types of colour schemes to consider (useful w.r.t colourbrewer2.org, seaborn)...

Other standard uses for colour:
→ Highlight
→ Alert

Type	Use when
Sequential	ordering values from low to high
Divergent	ordered values + critical mid-point (e.g average or zero)
Categorical / Qualitative	Data is in distinct groups, any colour could appear next to any colour (e.g. colouring counties based on properties). No implication of magnitude differences between legend classes.



THE USE OF COLOR IN DATA VISUALIZATION

SEQUENTIAL
color is ordered from low to high



DIVERGING
two sequential colors with a neutral midpoint



CATEGORICAL
contrasting colors for individual comparison



HIGHLIGHT
color used to highlight something



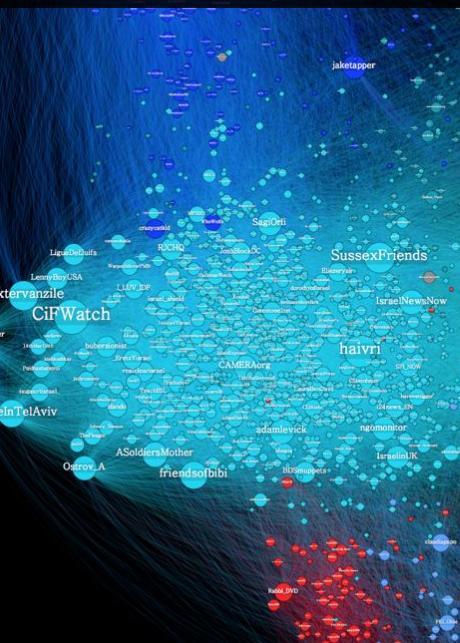
ALERT
color used to get reader's attention



[2] Cynthia A. Brewer, 1994, "Color Use Guidelines for Mapping and Visualization," Chapter 7 (pp. 123-147) in *Visualization in Modern Cartography*, edited by A.M. MacEachren and D.R.F. Taylor, Elsevier Science, Tarrytown, NY.

Image: The Big Book of Dashboards by Andy Cotgreave, Jeffrey Shaffer, and Victor Grossman (pg 15)

A good visualization
reduces the time to insight. Yes, still this.



Your brain is on autopilot...

At least a bit...

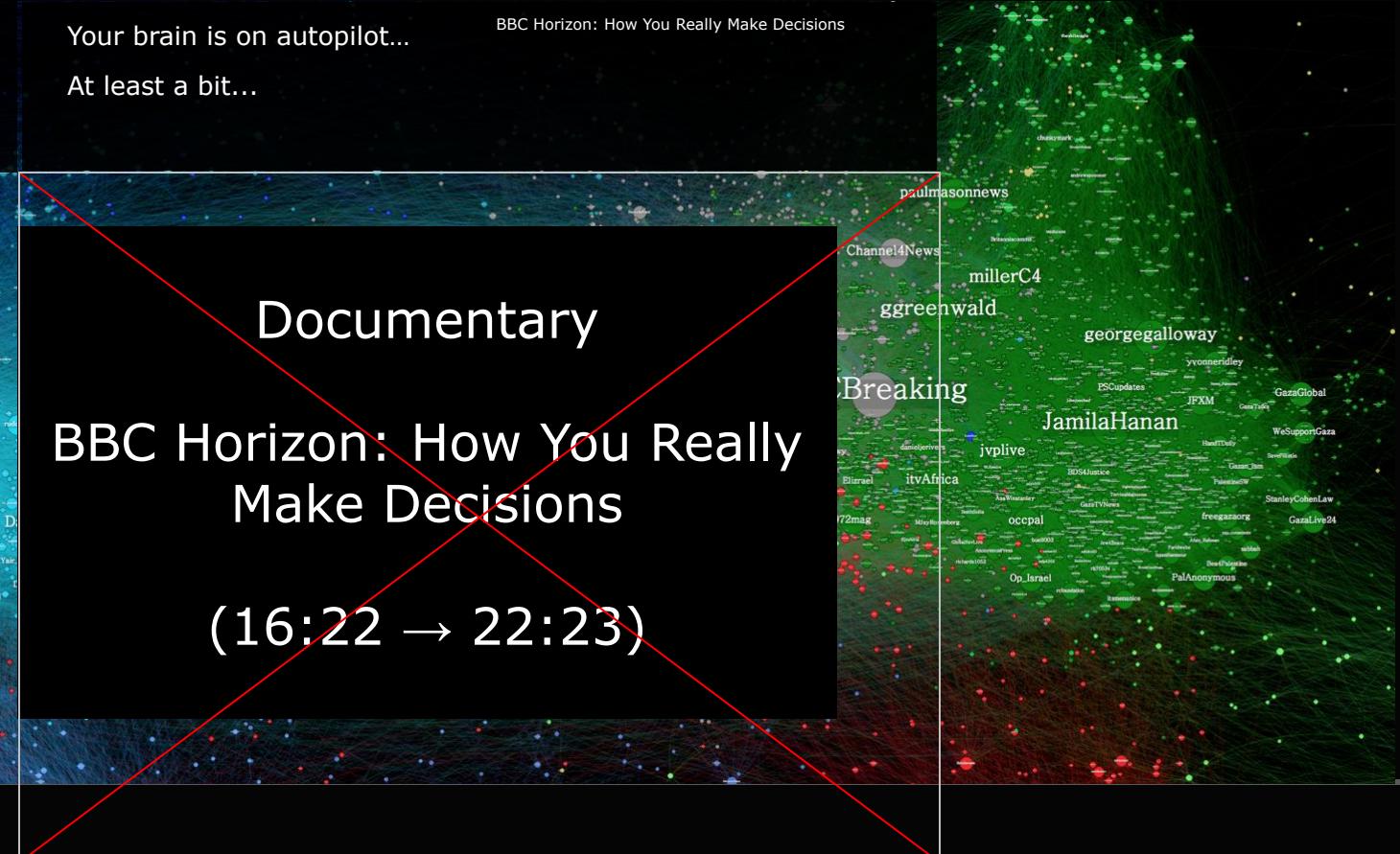
BBC Horizon: How You Really Make Decisions

Documentary

BBC Horizon: How You Really Make Decisions

(16:22 → 22:23)

Data at Scale



A good visualization
reduces the time to
insight.

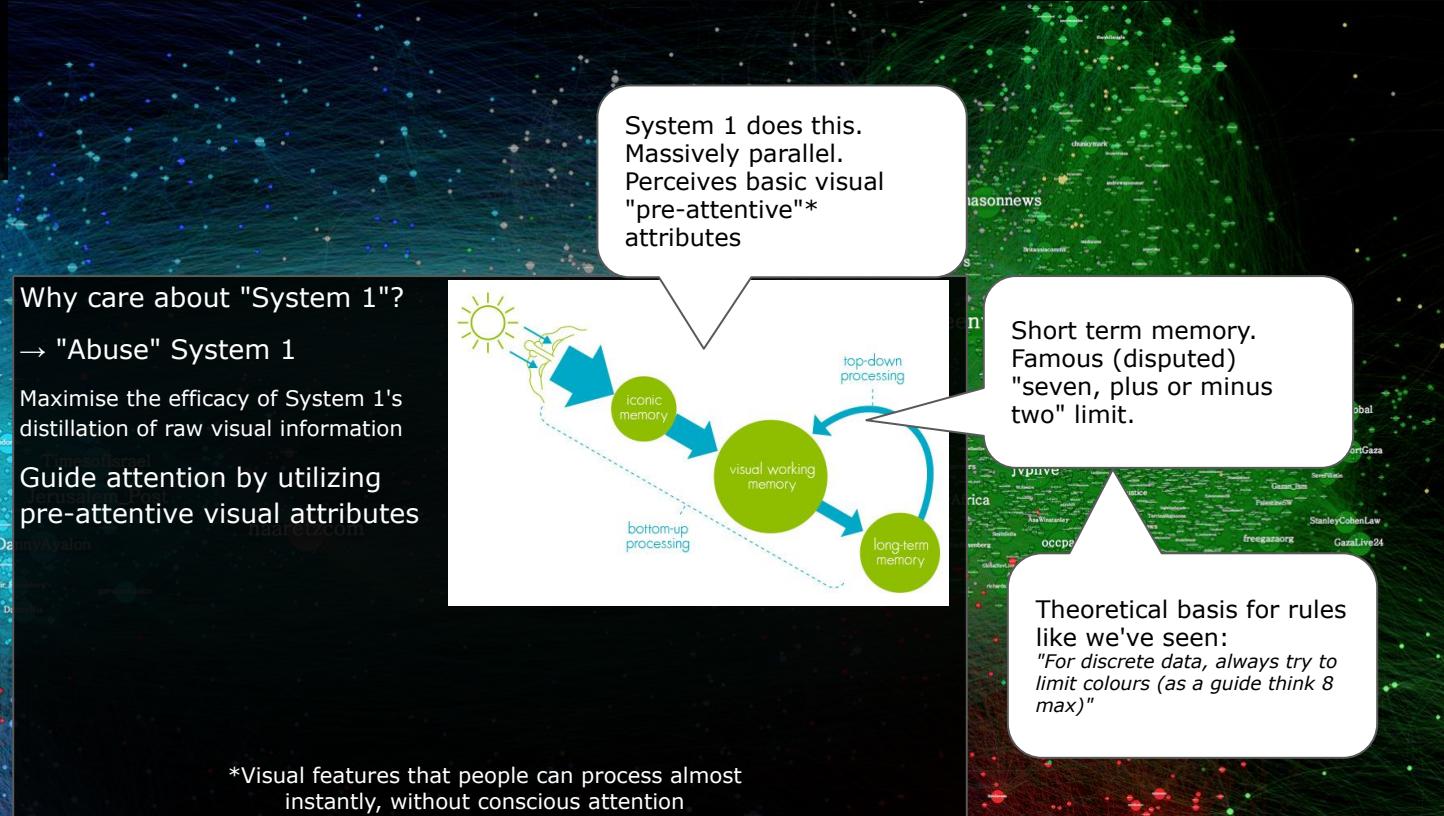
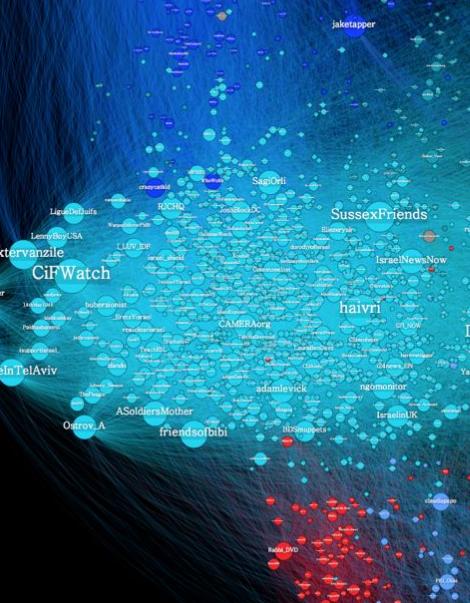


Figure: From Alberto Cairo's The Functional Art

A good visualization
reduces the time to
insight.

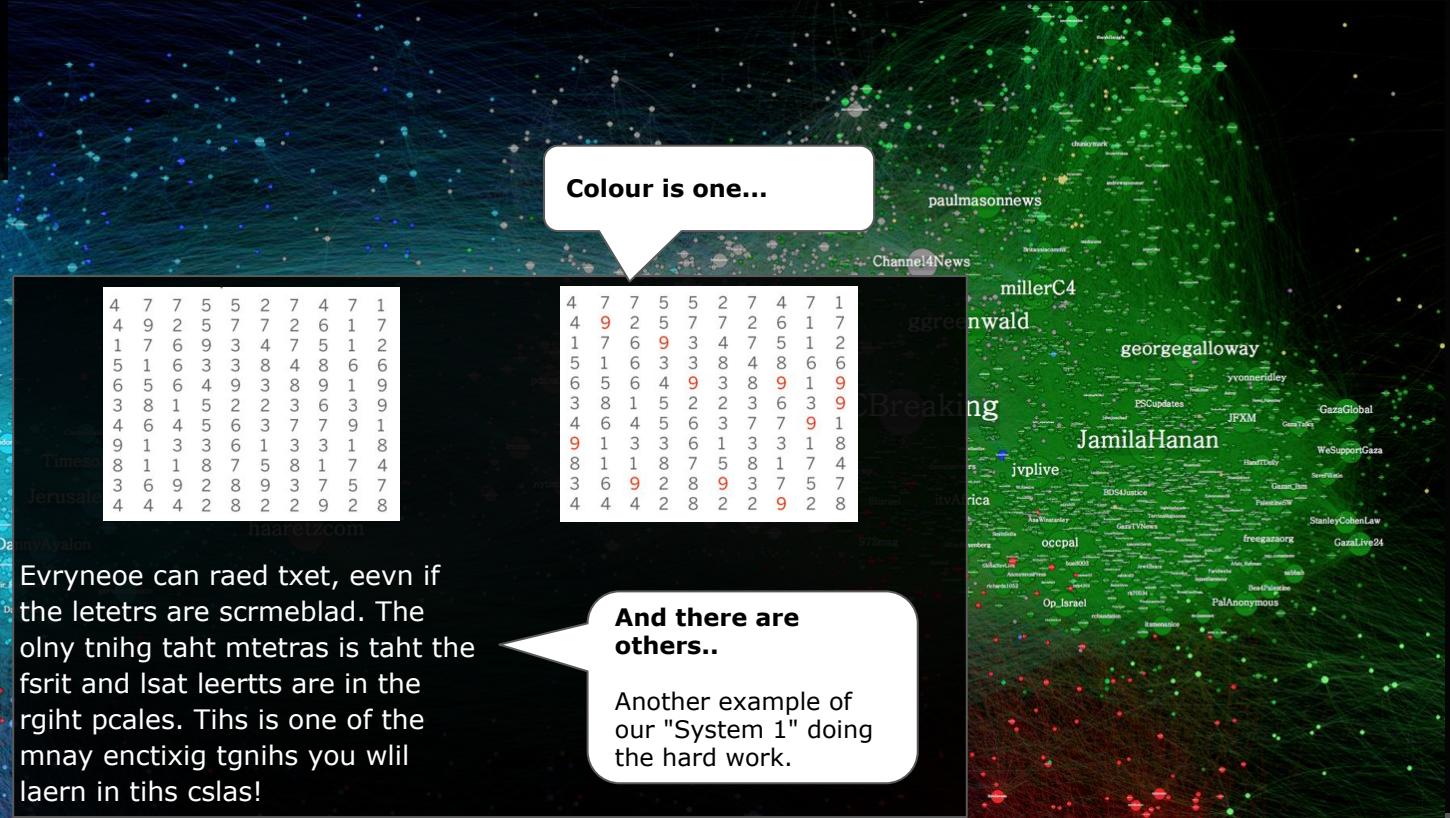
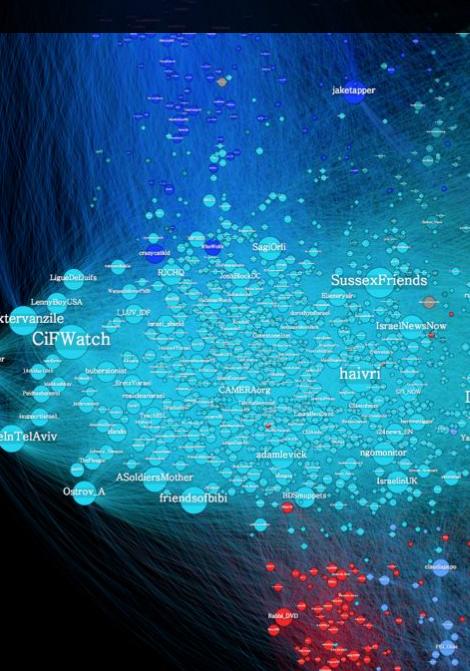
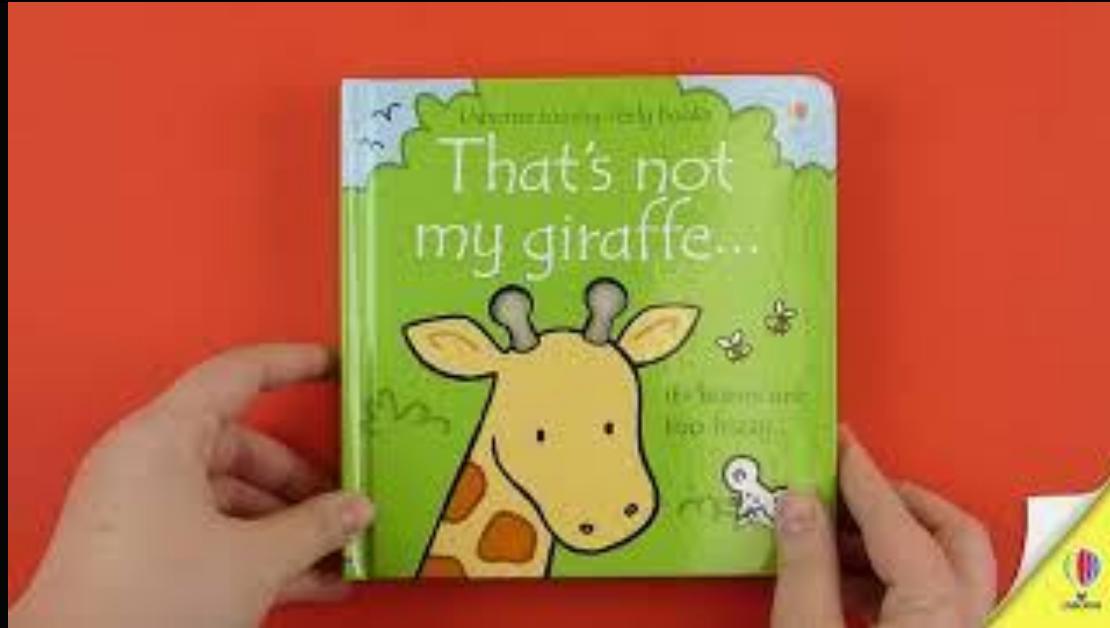
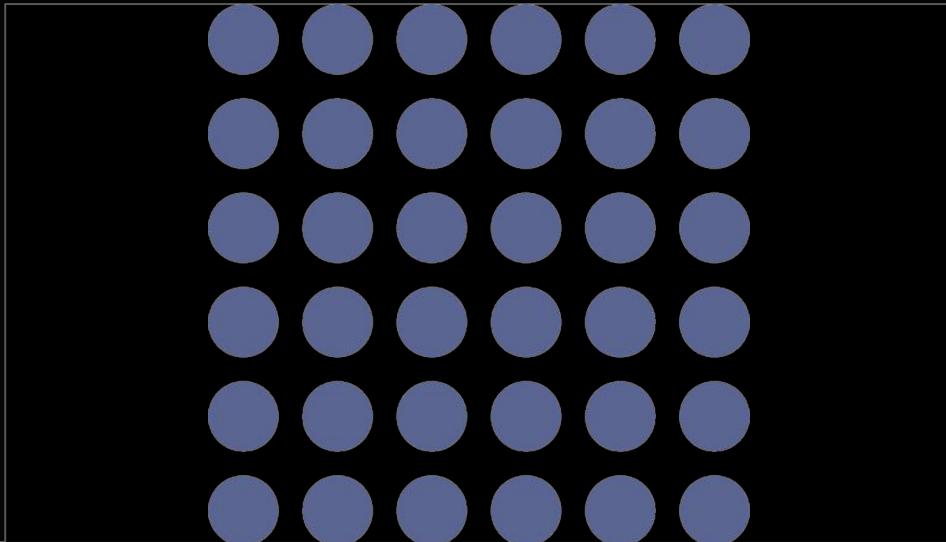


Figure: From Alberto Cairo's The Functional Art

What other pre-attentive attributes can you think of?



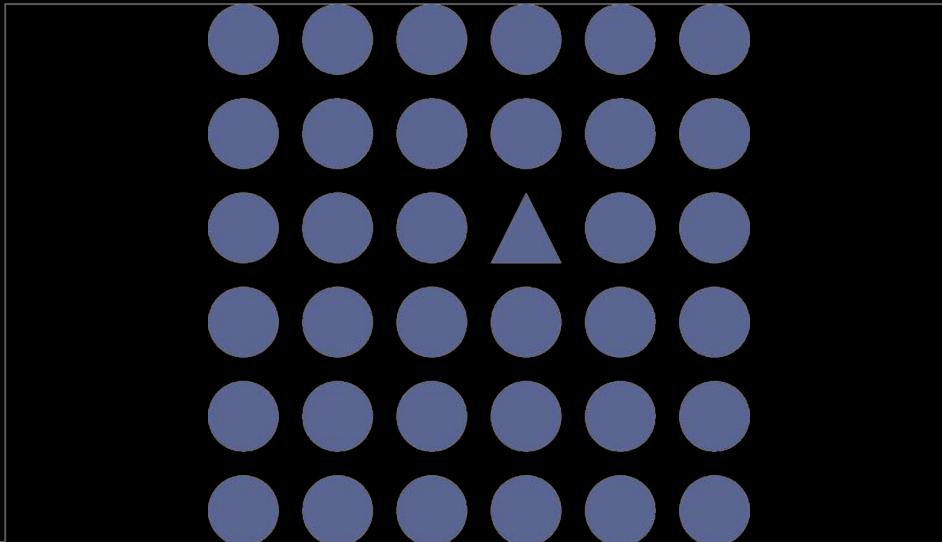
Shape



Data at Scale

Dr Evgeniya Lukinova

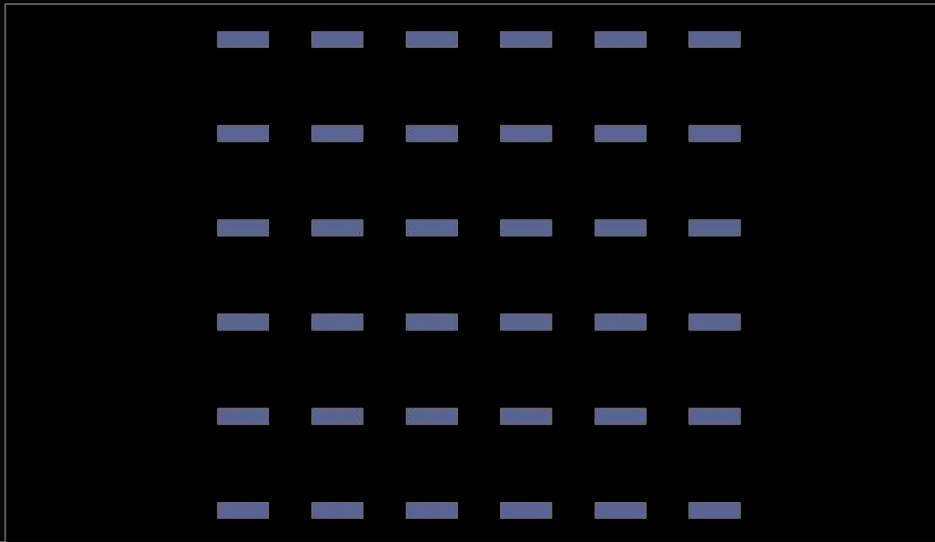
Shape



Data at Scale

Dr Evgeniya Lukinova

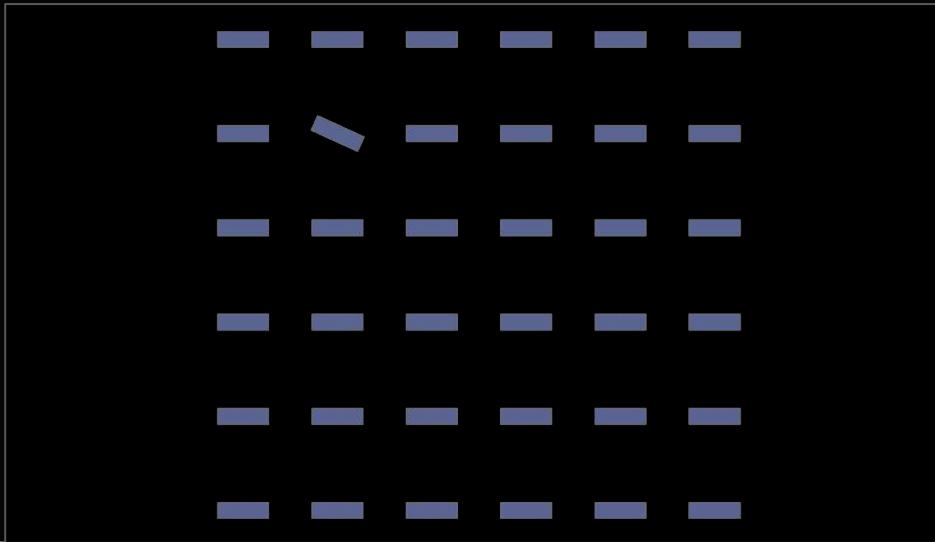
Orientation



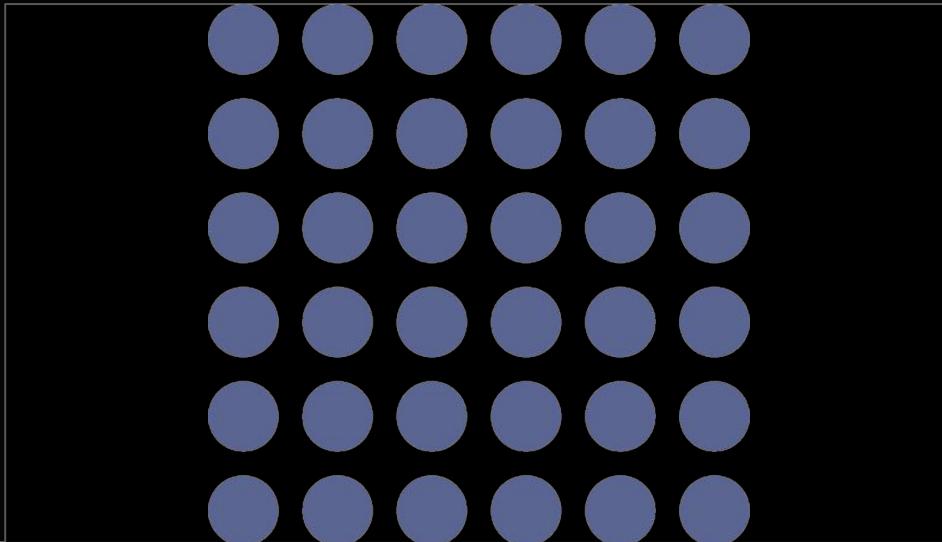
Data at Scale

Dr Evgeniya Lukinova

Orientation



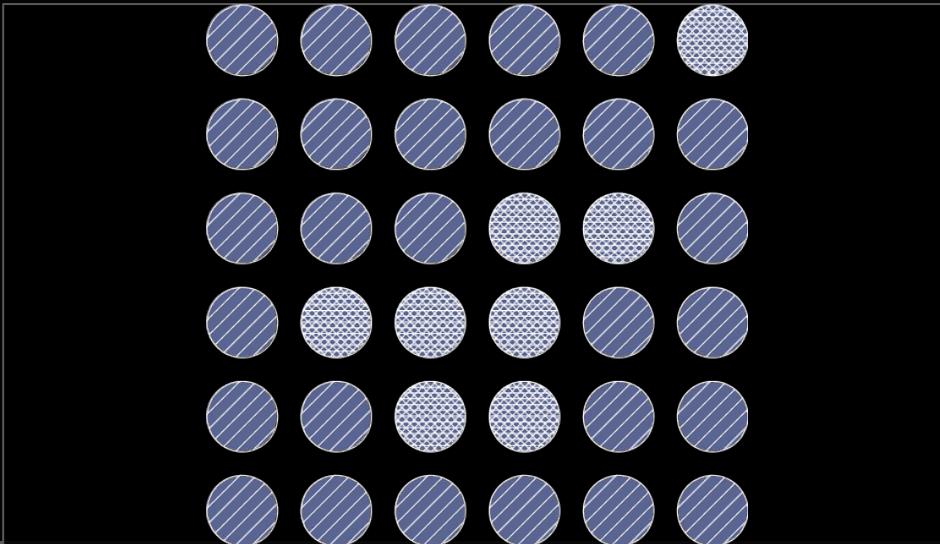
Texture



Data at Scale

Dr Evgeniya Lukinova

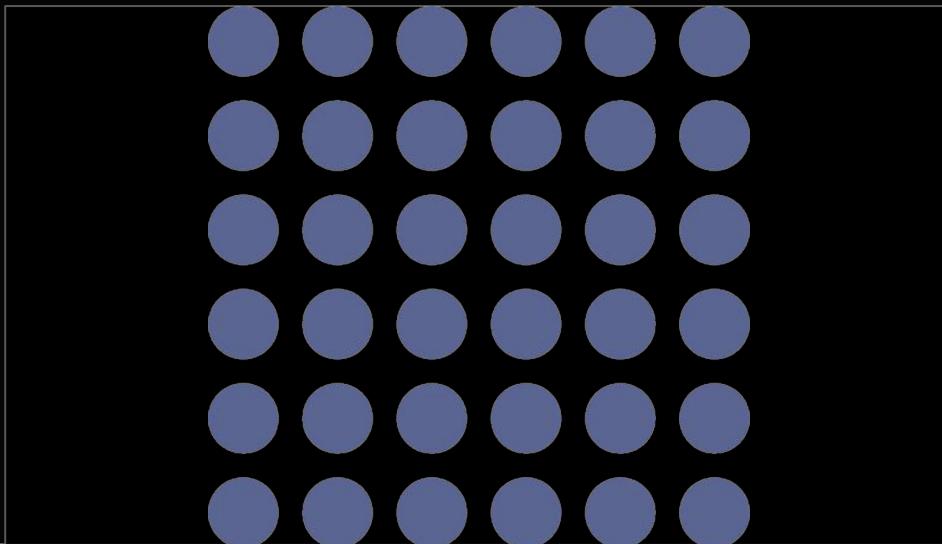
Texture



Data at Scale

Dr Evgeniya Lukinova

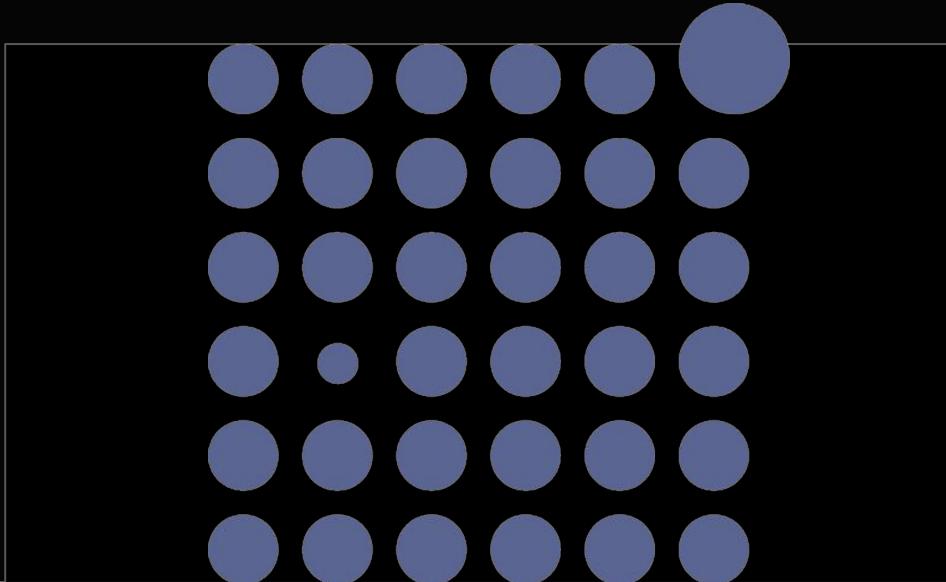
Size



Data at Scale

Dr Evgeniya Lukinova

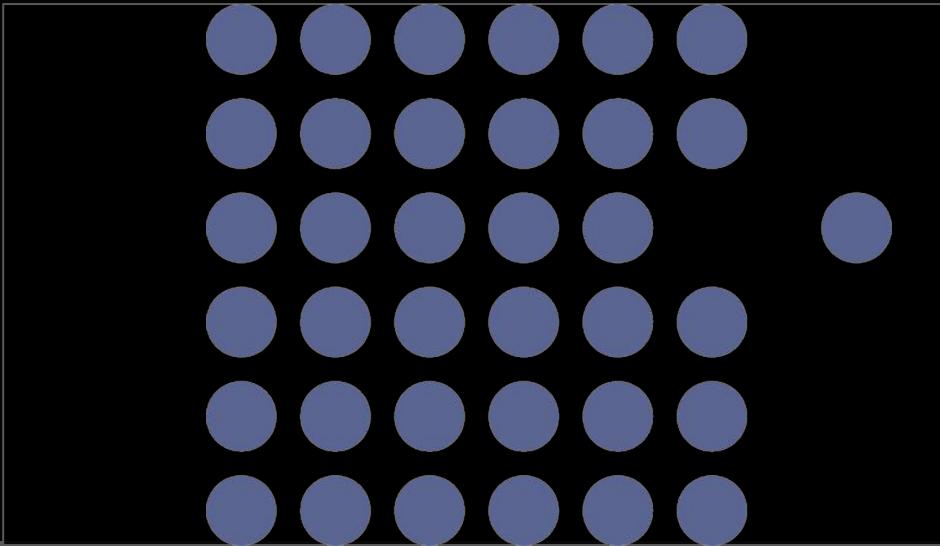
Size



Data at Scale

Dr Evgeniya Lukinova

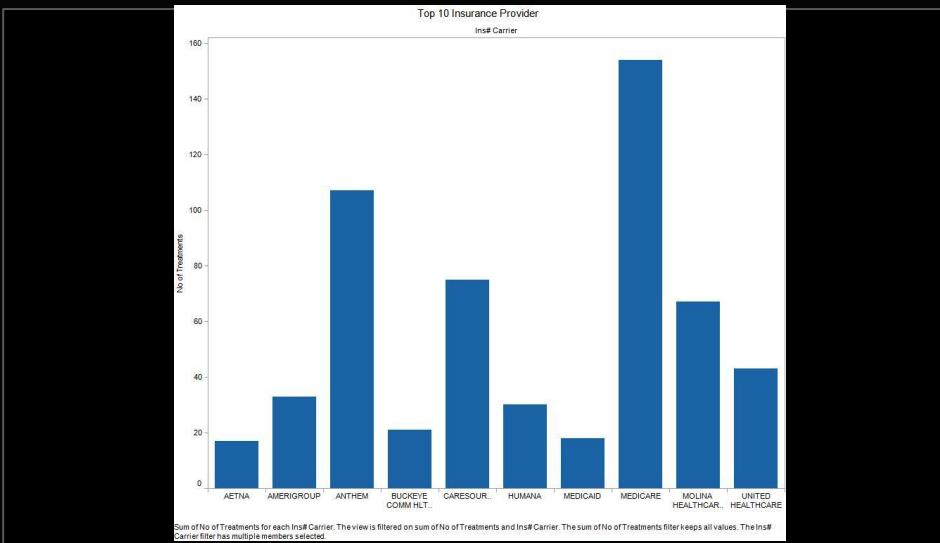
Position



Data at Scale

Dr Evgeniya Lukinova

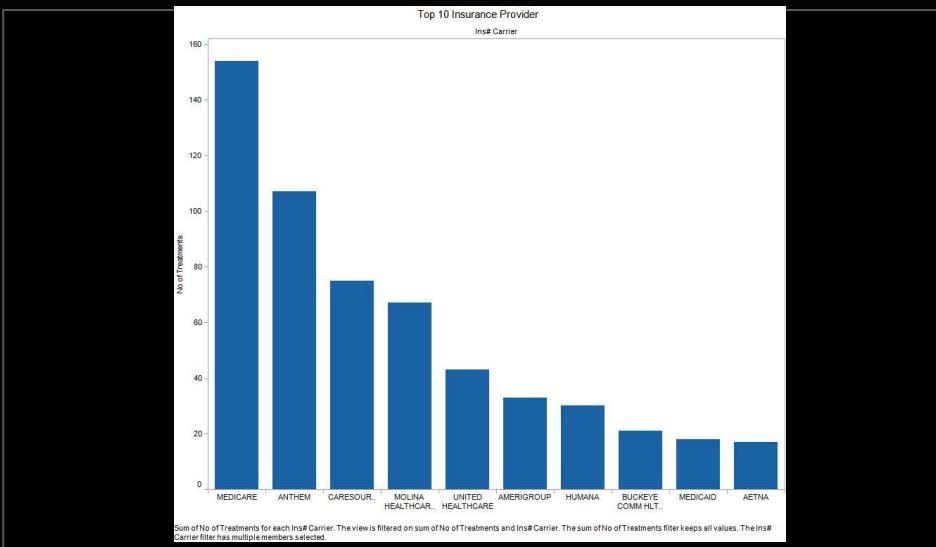
Order



Data at Scale

Dr Evgeniya Lukinova

Order



Data at Scale

Ok, so hopefully you're convinced pre-attentive attributes are a thing...

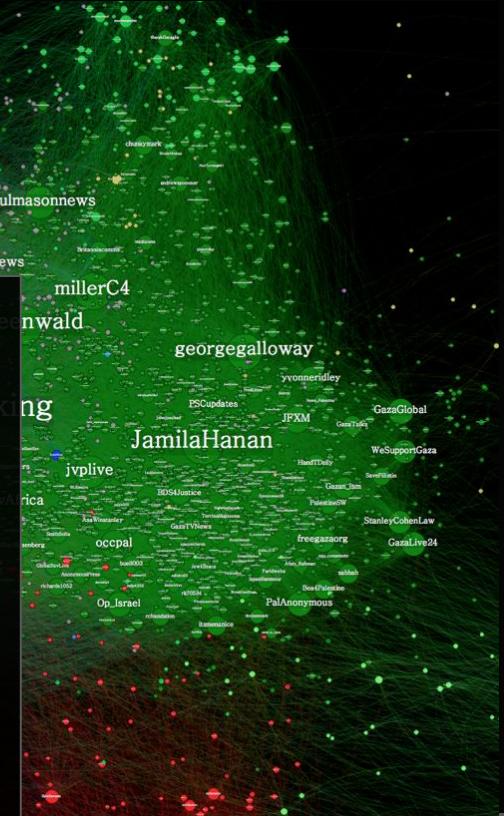


Figure: From Alberto Cairo's The Functional Art



Data at Scale

Dr Evgeniya Lukinova



A good visualization reduces the time to insight.

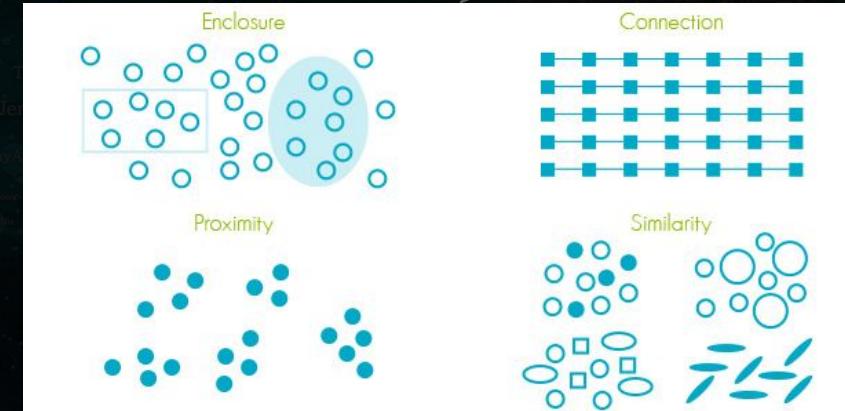


Order is another important pre-attentive attribute for highlighting relationships.

A good choice of visualisations can effortlessly and immediately highlight desired insights.

So how do we use them...

Highlighting relationships



The diagram shows four examples of highlighting relationships:

- Enclosure:** A group of circles is enclosed within a rectangular frame.
- Connection:** A series of squares connected by horizontal lines.
- Proximity:** A group of circles clustered together.
- Similarity:** A group of circles of varying sizes and shapes.

Figure 1 & 2: From Stephen Few's Information Dashboard Design

Figure 3: from Cleveland and McGill. Visualisation from Alberto Cairo's The Functional Art



Data at Scale

Dr Evgeniya Lukinova



A good visualization
reduces the time to
insight.



More analytical, higher in the chart (more standard).



Compromise between accuracy and visual interest required for the particular story.

So how do we use them...

Highlighting relationships

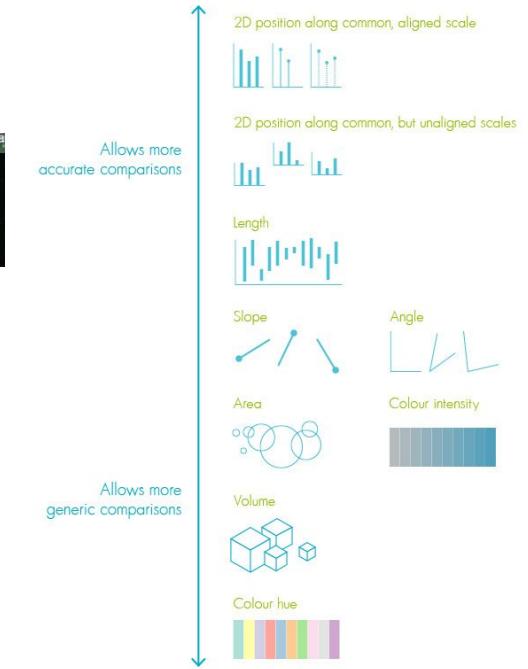
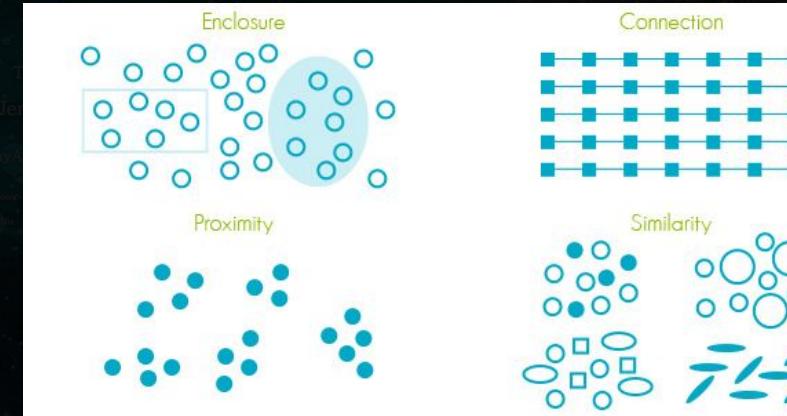
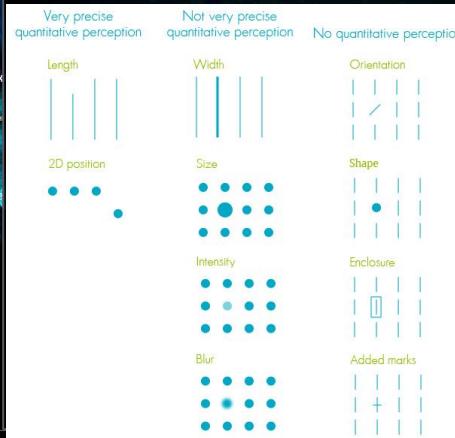


Figure 1 & 2: From Stephen Few's Information Dashboard Design

Figure 3: from Alberto Cairo's The Functional Art



Data at Scale

Dr Evgeniya Lukinova

A good visualization
reduces the time to
insight.

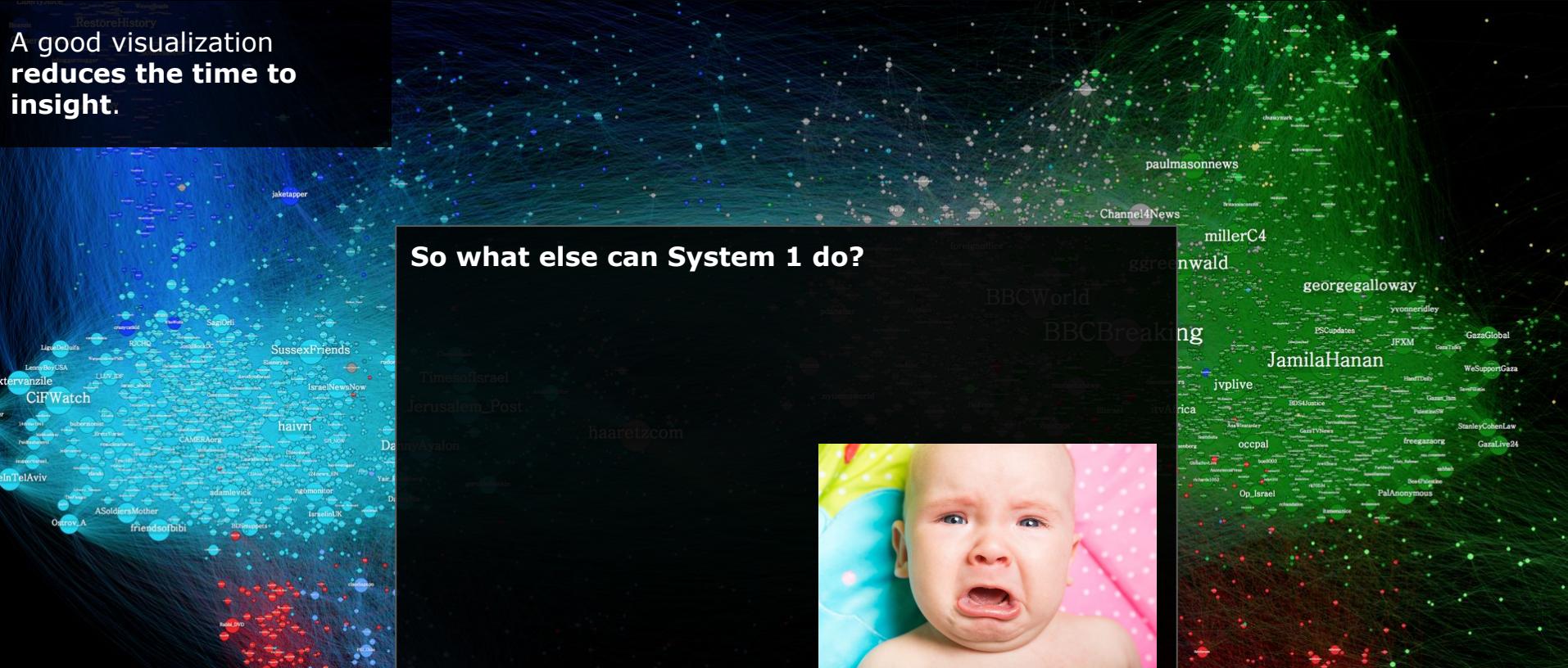


Figure: From Alberto Cairo's The Functional Art



Data at Scale

Dr Evgeniya Lukinova

A good visualization
**reduces the time to
insight.**



So what else can System 1 do?



Figure: From Alberto Cairo's The Functional Art

Labeled by System
Ways Back
RestoreHistory

A good visualization
reduces the time to
insight.

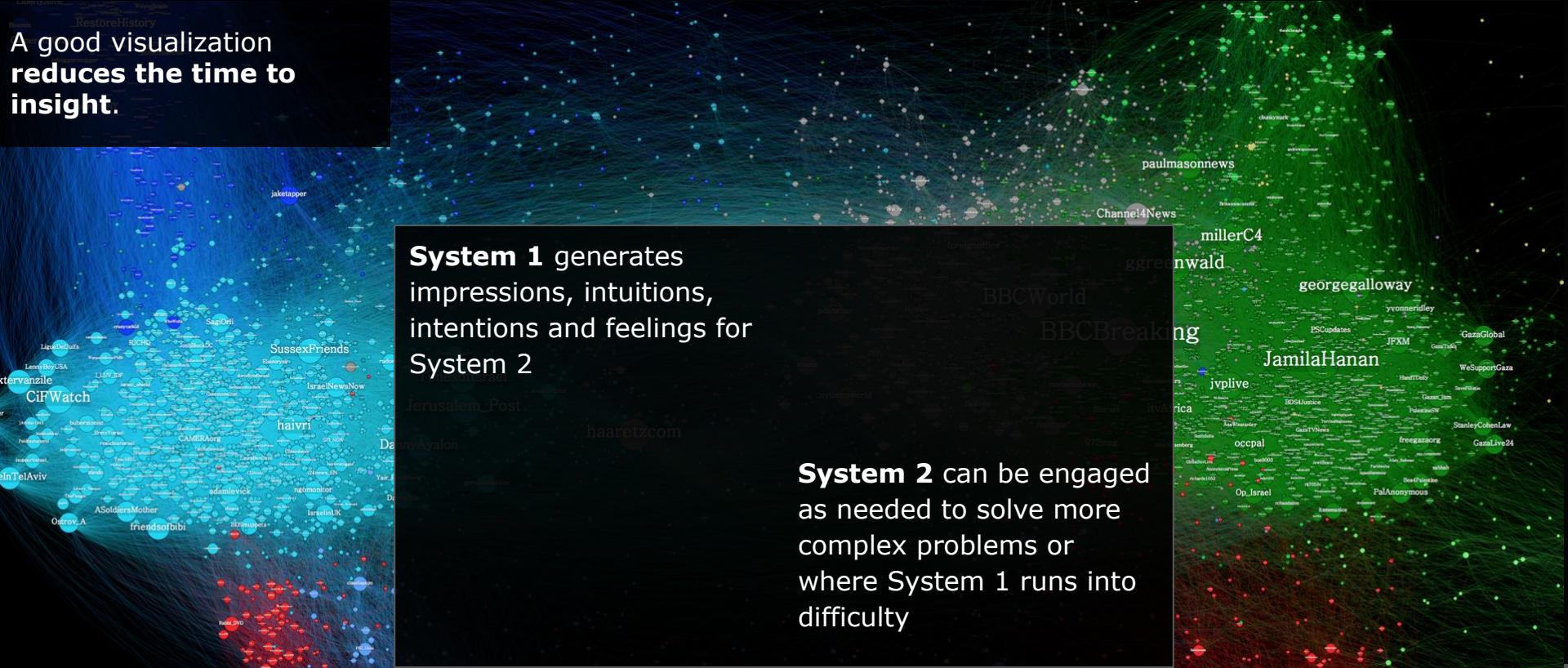


Figure: From Alberto Cairo's The Functional Art



Data at Scale

Dr Evgeniya Lukinova

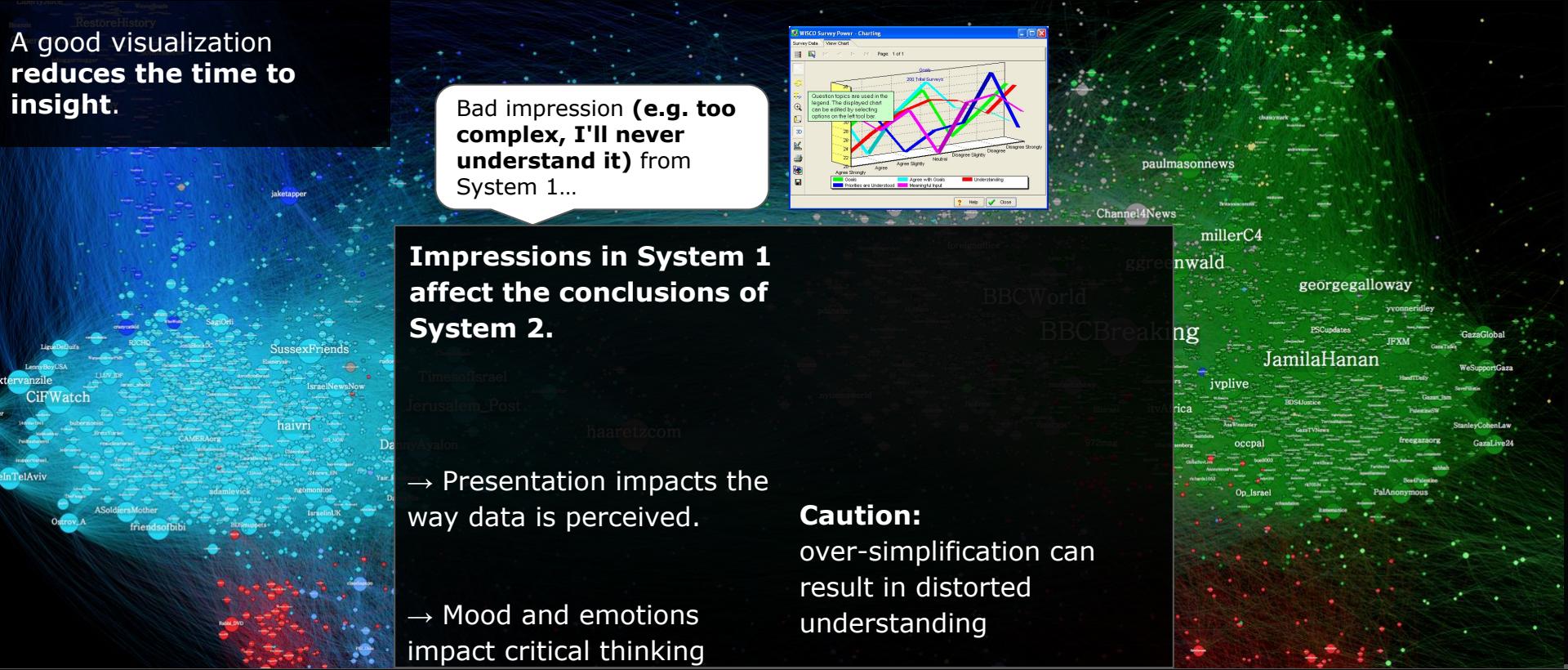


Figure: From Alberto Cairo's The Functional Art

Summary

Why now? No excuse not to make nice visualizations. Ever. Not even at the start.

Data exploration vs. Data presentation (focus).

However, never forget:
"Above all else, show the data! Graphics is *intelligence made visible*"

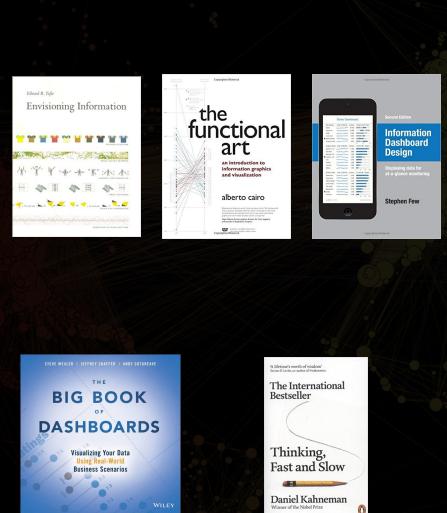
Edward Tufte

Understanding and using pre-attentive attributes

"an explanation should be as simple as possible, but no simpler" Einstein, obviously

We'll take a look at **chart types** and more general **design considerations** later

Interested in reading more?



[1] Lyn Bartram, Abhishek Patra, and Maureen Stone. 2017. Affective Color in Visualization. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17). ACM, New York, NY, USA, 1364-1374. DOI: <https://doi.org/10.1145/3025453.3026041>)

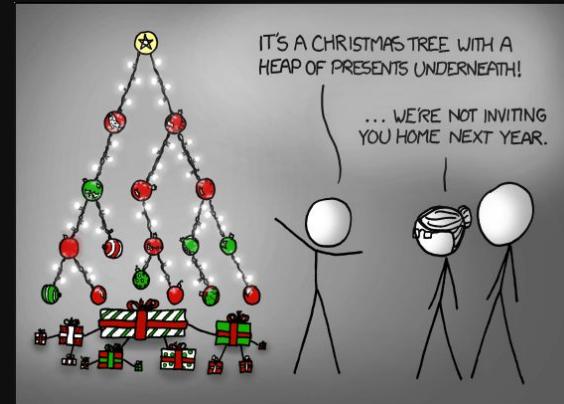
[2] Cynthia A. Brewer, 1994, "Color Use Guidelines for Mapping and Visualization," Chapter 7 (pp. 123-147) in *Visualization in Modern Cartography*, edited by A.M. MacEachren and D.R.F. Taylor, Elsevier Science, Tarrytown, NY.

[Research on colour by Maureen Stone](#)

[Good blog: The science behind data visualisation](#)

Session 2

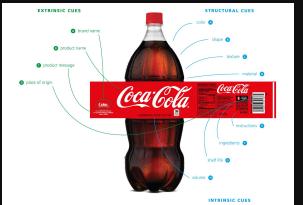
Data structures for Business



Context is everything... (so let's take a look)



Google Analytics
mouseflow
Website tracking data



Product Detail Data



Office for National Statistics
census 2021





(Formally)
Describing data
requires a paradigm

Intuitive paradigm?

Objects vs

Relational

Objects!



Lego slides from:

<http://nosqlroadshow.com/nosql-london-2012/speaker/Akmal+B.+Chaudri>
<https://www.slideshare.net/VeryFatBoy/nosql-roadshow-london-2012>



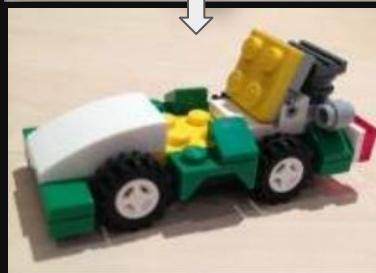
Data at Scale

Dr Evgeniya Lukinova

Best paradigm?

Not always
objects...

Alternatively I could store
my lego in pieces...



Need to know what pieces
(simple building blocks) **are**
available and to have the
assembly **instructions**



Images from:
<http://nosqlroadshow.com/nosql-london-2012/speaker/Akmal+B.+Chaudri>
<https://www.slideshare.net/VeryFatBoy/nosql-roadshow-london-2012>



Data at Scale

Dr Evgeniya Lukinova

Benefit: Flexibility



Otherwise to make cars:

- 1st understand how it is constructed
- Work out how to break it down
- Create new cars



Images from:

<http://nosqlroadshow.com/nosql-london-2012/speaker/Akmal+B.+Chaudri>
<https://www.slideshare.net/VeryFatBoy/nosql-roadshow-london-2012>

Best paradigm?

But **why not just store all three cars?**

What if I bought the wrong wheels to replicate my favorite car, and the hub caps are silver?

(**data replication**)

- hard to update
- data often conflicts



Images from:

<http://nosqlroadshow.com/nosql-london-2012/speaker/Akmal+B.+Chaudri>
<https://www.slideshare.net/VeryFatBoy/nosql-roadshow-london-2012>

Benefit: Flexibility

Cars With the Most Safety Recalls - iSeeCars Study

Rank	Model	Expected 30-Year Lifetime Recalls
1	Porsche Taycan	70.7
2	Tesla Model Y	66.9
3	Tesla Model 3	60.7
4	Porsche Panamera	43.1
5	Lucid Air	40.1
6	Tesla Model S	38.5
7	Tesla Model X	37.6
8	Lincoln Aviator	26.2
9	Genesis GV70	22.3
10	Kia Telluride	22.2



In 2023, the average car was projected to have 3.2 recalls throughout its 30-year lifespan.

The Porsche Taycan was projected to have 70.7, Tesla Model Y - 66.9, while the Mini Convertible was projected to have 0.2.

Images from:

<https://imageio.forbes.com/specials-images/imageserve/668dcda8f89592a74415bc6f/2024-Most-Recalled-Cars/960x0.png?format=png&width=1440>
<https://imageio.forbes.com/specials-images/imageserve/b68dc01d16c9ce19e00bd19/Tesla-Issues-Recall-On-2-Million-Of-Its-Vehicles-In-The-U.S-Due-To-Autopilot-Issue/960x0.jpg?format=jpg&width=1440>

Summary:

Storing one car and having to break it into pieces and rebuild things is hard.

Better just to store the pieces and build instructions.



Storing multiple copies so you don't have to break the original into pieces can duplicate data - introduces potential for error.

Better just to store the pieces and build instructions.



Storing pieces and build instructions can be slower if you just want to ask about a specific car.

Are you sure you're only ever going to want to ask about a specific car?

Is the speed gain worth it?



Images from:

<http://nosqlroadshow.com/nosql-london-2012/speaker/Akmal+B.+Chaudri>
<https://www.slideshare.net/VeryFatBoy/nosql-roadshow-london-2012>

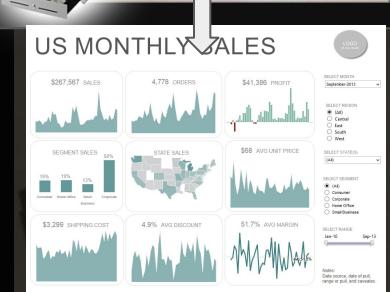
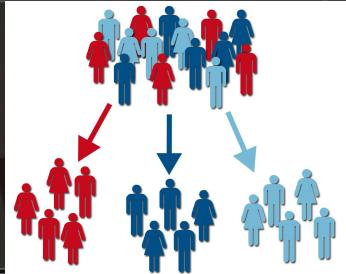


Why are we talking
about lego anyway?

Best paradigm?

Why are we talking about lego anyway?

Let's consider transactional data, i.e. a receipt.



Of course data comes originally in one format.
Cost associated with:
→ Working out best "pieces"
→ Breaking it up

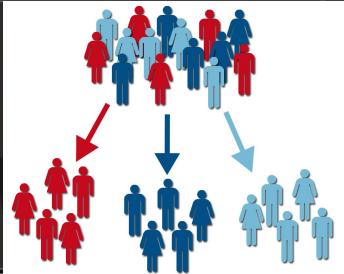
Typically not your job....



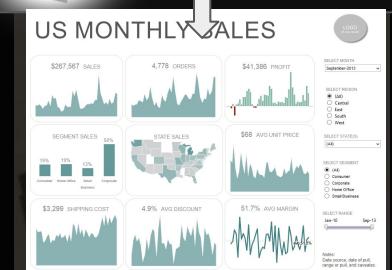
Best paradigm?

Why are we talking about lego anyway?

Let's consider transactional data, i.e. a receipt.



Original format



Of course data comes originally in one format.
Cost associated with:
→ Working out best "pieces"
→ Breaking it up

Typically not your job....

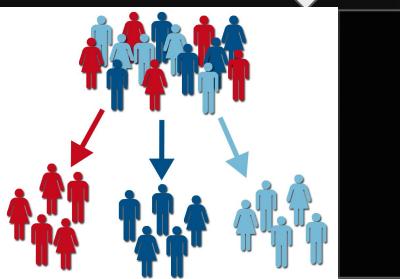
If it was your job, or if you had an **object based datastore**....

- Need to examine the structure of **each object type each time**
- Need to work out what pieces you have **for each object**
(information loss?)
(may end up in complex structures...)

Best paradigm?

Not always
objects...

If this isn't your job, and has been done once by the company
(or just stored correctly by design in the first place)...



Still need to know what pieces (simple building blocks) are available and the assembly instructions



But we start from a better position.

Really want a well defined, standard set of base building blocks from which it is proven we can construct anything.

Plus a standard (easy) language for the instructions.

Rather than different styles of building blocks dreamt up by an individual or company.

Images from:

<http://nosqlroadshow.com/nosql-london-2012/speaker/Akmal+B.+Chaudri>
<https://www.slideshare.net/VeryFatBoy/nosql-roadshow-london-2012>

Best paradigm?

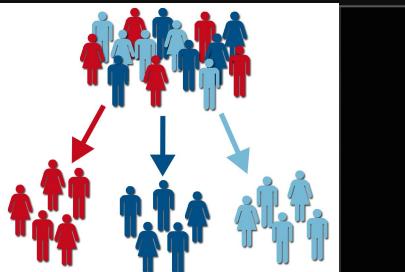
But **why not just store all three cars?**

What if I want to make a new car? (flexibility)

What if my data entry was wrong customer was female instead of male?

(data duplication)

- hard to update
- data often conflicts



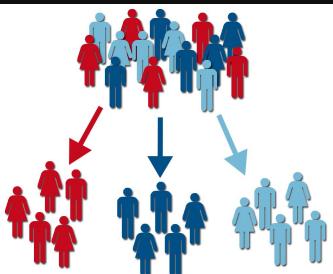
Same issue with "cars" as in the real-world, but potentially worse if incorrect data has then been used and **copied** into lots of different reports...

Images from:

<http://nosqlroadshow.com/nosql-london-2012/speaker/Akmal+B.+Chaudri>
<https://www.slideshare.net/VeryFatBoy/nosql-roadshow-london-2012>

Summary:

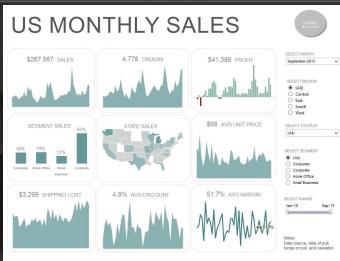
Storing complex **structured data** & having to break it into pieces & rebuild things is hard.



Better just to store the pieces and build instructions.

Storing multiple copies so you don't have to break the original into pieces can duplicate data - introduces potential for error.

Better just to store the pieces and build instructions.



Storing pieces and build instructions can be slower if you just want to ask about cars.

*Are you sure you're only ever going to want to show a single report?
Is the speed gain worth it?*



Images from:
<http://nosqlroadshow.com/nosql-london-2012/speaker/Akmal+B.+Chaudri>
<https://www.slideshare.net/VeryFatBoy/nosql-roadshow-london-2012>

A collage of various LEGO characters from the movie 'The LEGO Movie'. In the foreground, there's a large yellow LEGO man with a surprised expression wearing an orange vest over a blue shirt. To his right is a pink cat-like character with large eyes. Behind them is a blue robot, a white ghost-like character with green eyes, a black Batman-like figure, and a woman with blue hair and a pink jacket. In the background, there's a large multi-headed orange robot with a skull and crossbones on its chest, and other smaller characters like a small dog and a person with glasses.

So maybe we do
want this lego style
thing...

So what are these
lego pieces anyway?



Data is Latin for “facts”

Turns out (rather than objects)
facts make good basis
(lego) pieces to construct
things (i.e. cars, reports).

So what are these
lego pieces anyway?



Data is Latin for “facts”

Turns out (rather than objects)
facts make good basis
(lego) pieces to construct
things (i.e. cars, reports).

Why?

- Facts are either true or false.

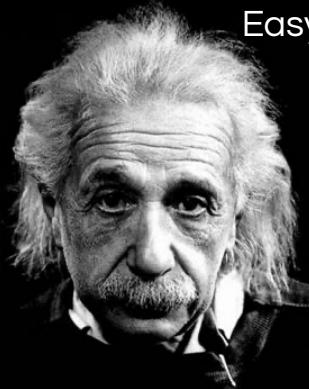


"A Dairy Milk costs £2.50"
"Evgeniya lives in Nottingham"

“Everything should be made
as simple as possible,
but not simpler.”

Albert Einstein

Binary.
Easy for humans.
Easy mathematically.
Easy for machines.



So what are these
lego pieces anyway?



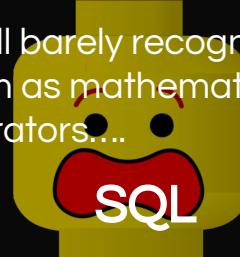
Data is Latin for “facts”

Turns out (rather than objects)
facts make good basis
(lego) pieces to construct
things (i.e. cars, reports).

~8 Mathematical operators can be used to do almost all required data manipulation



You'll barely recognise them as mathematical operators...



Interested in only a few chocolate bars?

Filter based on fact x being true...



So what are these
Simple set of operators,
lego pieces facts very well?
understood
mathematical theory



Data is Latin for “facts”

Turns out (rather than objects)
facts make good basis
(lego) pieces to construct
things (i.e. cars, reports).

Translation from
operators (easy for humans) →
**computer code is well
understood.**



Can directly use operators.

Describe what we want (ask
questions), not how to do it (list steps).

Computer will (mostly)
automatically optimise.



So what are these
lego pieces anyway?

Data is Latin for “facts”

Turns out (rather than objects)
facts make good basis
(lego) pieces to construct
things (i.e. cars, reports).

But wait, there's more...

Good base paradigm +
development since the
70's mean....

Sets of facts can be designed to
prevent data duplication.

Multiple people building reports
from a single source of up-to-date
data **ensures consistency as
data changes.**

Sets of facts can have **some
constraints** (basic business rules)
enforced on entry.



Data at Scale

Dr Evgeniya Lukinova

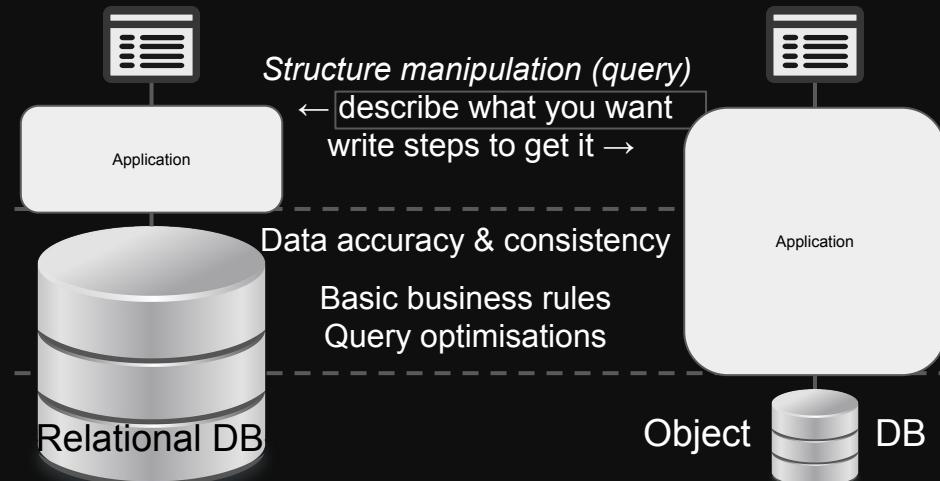
What does that mean in practice?

Reduced development time.

Less errors.

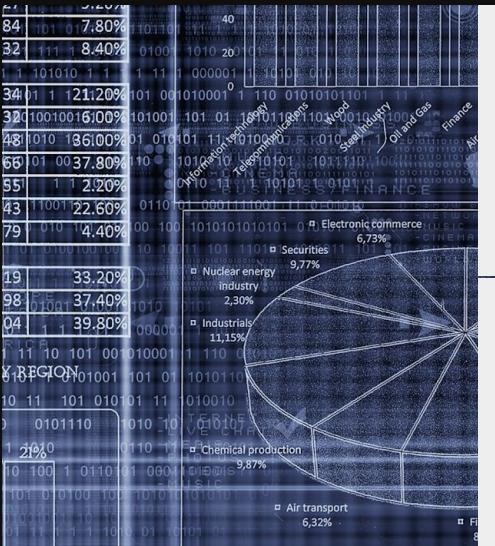
"... we find developers spend a significant fraction of their time building extremely complex and error-prone mechanisms to cope with eventual consistency and handle data that may be out of date. We think this is an unacceptable burden to place on developers and that consistency problems should be solved at the database level."

[Google] Shute, Jeff, et al. "F1: A distributed SQL database that scales." Proc. VLDB (2013): 1068-1079.

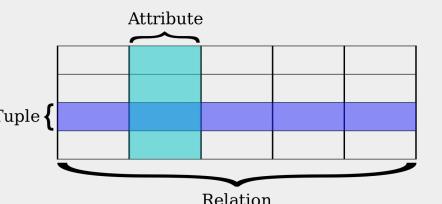


Sets of facts == relational databases!!

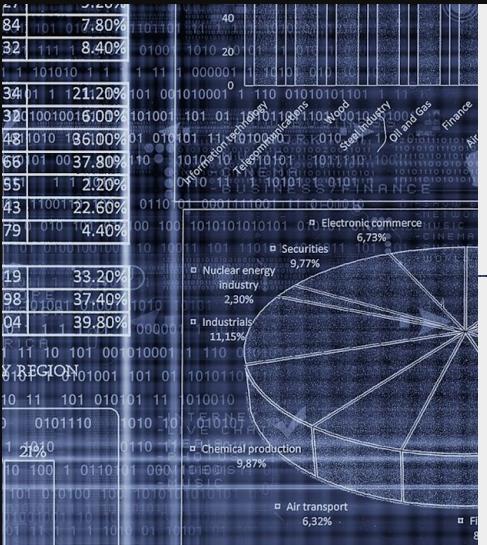
Lots more
in a
week.....



- Replaced “Navigational Databases” in the late 70's, because of E.F. Codd's work at IBM.
- Based on **Tables, columns and rows**.
- Or more accurately **relations, attributes and tuples**.



Data is Latin for “facts”

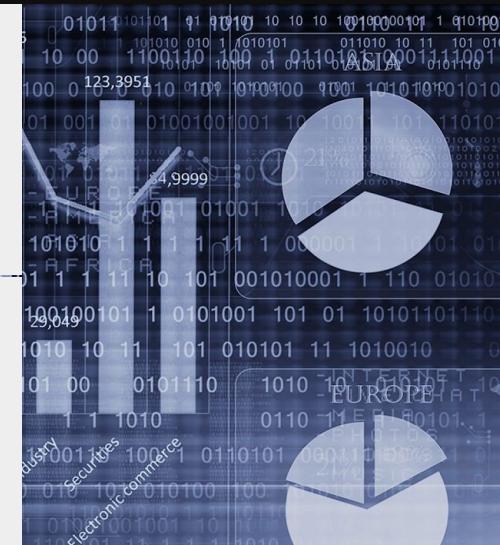


Kermit is a green Frog.
Miss Piggy is a pink pig.
Fozzy is an orange bear.

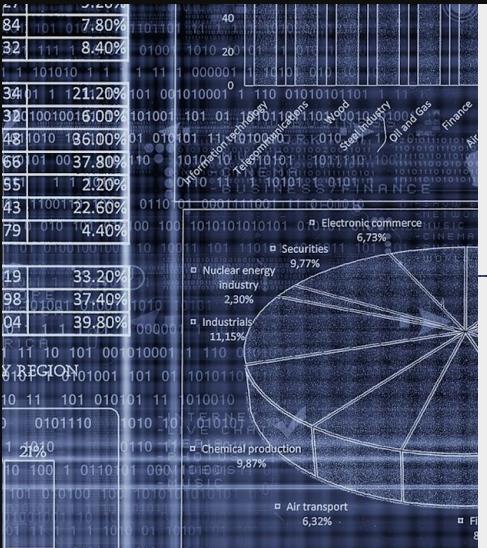
Relation header = (Name, Colour, Animal)
Relation content = { (Kermit, Green, Frog),
 (Miss Piggy, Pink, Pig),
 (Fozzy, Orange, Bear) }

Name(Kermit) & Colour(Green) & Animal(Frog)
Name(Miss Piggy) & Colour(Pink) & Animal(Pig)
Name(Fozzy) & Colour(Orange) & Animal(Bear)

Name	Colour	Animal
Kermit	green	frog
Miss piggy	pink	pig
Fozzy	orange	bear



Relations (Tables) are lists of facts about different <table name>



Name	Colour	Animal
Kermit	green	frog
Miss piggy	pink	pig
Fozzy	orange	bear

Example: Shop receipts

Receipt Table (Relation)

Receipt ID	Cust ID	Date	Shop	Total
0001	0001	1/1/17	NC1N 2HT	£1.70

Facts (about a receipt)

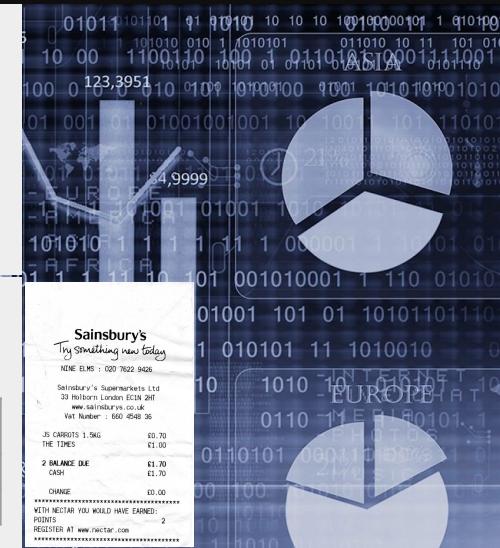
Each row (tuple) refers to facts about a single entity*

*need not be an object!

Receipt Line Table (Relation)

Receipt ID	Line no.	Desc.	Qty	Amnt
0001	0001	JS CARROTS 1.5KG	1	£0.70
0001	0002	THE TIMES	1	£1.00

Facts (about a line on a receipt)



The take-home message
(long version)....

+ve & -ve for objects
as a storage paradigm
(vs. relational databases)

Data access is simple when processing objects as stored & if objects have
identical/correct structure
& data is error free (or you don't care)



Images:

<http://nosqlroadshow.com/nosql-london-2012/speaker/Akmal+B.+Chaudhri>
<https://www.slideshare.net/VeryFatBoy/nosql-roadshow-london-2012>

Source:
[https://softwareefficiency.wordpress.com/2015/03/14/big-data-technology-a
nd-the-responsibility-shift/](https://softwareefficiency.wordpress.com/2015/03/14/big-data-technology-and-the-responsibility-shift/)

The take-home message (long version)....

+ve & -ve for objects as a storage paradigm (vs. relational databases)

Storing objects can be fast
dump objects (cars) one at a time
as they turn up

Data access is simple when

processing objects as stored

- If true...
 - Fast, no need for "rebuilding".
 - Potentially conceptually simpler
- If false...
 - Complex language and logical structure to navigate for rebuilding.
 - Less automatic optimisations due to paradigm & reduced structure

- If true...
 - no time spent on business rule checks
 - business rules can easily evolve
 - good custom code can be faster

If false...

- re-inventing code to check business rules in each application or not checking
- custom code is likely slower / prone to errors

3

if objects have
identical/correct structure
& data is error free (or you don't care)

- If true...
 - no time spent on error checks
- If false...
 - re-inventing code to check errors in each application or not checking
 - custom code is likely slower / has errors



Images:

<http://nosqlroadshow.com/nosql-london-2012/speaker/Akmal+B.+Chaudhri>
<https://www.slideshare.net/VeryFatBoy/nosql-roadshow-london-2012>

Source:

<https://softwareefficiency.wordpress.com/2015/03/14/big-data-technology-a-and-the-responsibility-shift/>

The take-home message
(short version)....

Unless...:

Normally → Use relational databases

Too much data makes checks /
writes/access too slow.

Data is always in, and processed
in, objects*.

Data structure changes
constantly.

May store as
objects

(or another logical structure)



Data at Scale

Dr Evgeniya Lukinova

The take-home message
(short version)....

Unless...:

Normally → Use relational databases

Too much data makes checks /
writes/access too slow.

Data is always in, and processed
in, objects*.

Data structure changes
constantly.

May store as
objects

(more likely another logical structure or
potentially in "best of both worlds" databases -
covered later)



Data at Scale

The take-home message
(short version)....

Normally

Use relational
databases

Otherwise

Too much data makes checks /
writes/access slow.

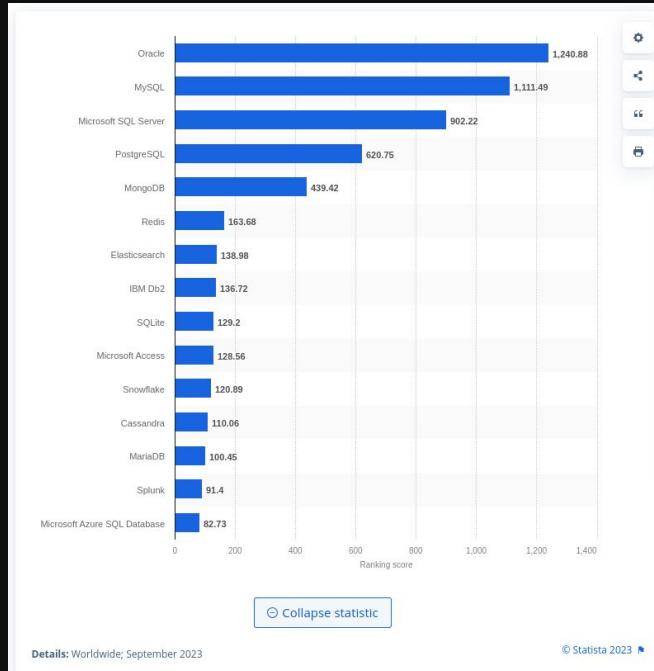
Data is always in, and processed
in, objects*.

Data structure changes
constantly.

May store as
objects

(more likely another logical structure or
potentially in "best of both worlds" databases -
covered later)

noSQL
newSQL
sparkSQL



Data at Scale

Image:
<https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/>

Dr Evgeniya Lukinova



ukinova

Context is everything... (so let's take a look)

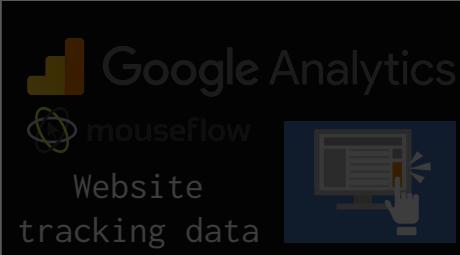
Pick the
processing & storage
paradigm* for the job.

(* we now know two!)

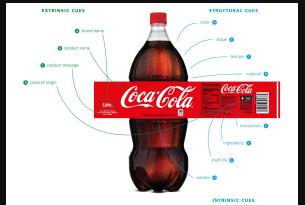
Is traditional (relational)
approach enough?



Transaction data



Security logs



Product Detail Data



Call Detail Record (CDR) Data



Survey Data



Social Media Data



Census Data

Transactional data

(Event data)



Pick the
processing & storage
paradigm for the job.

Is traditional approach
enough?

- Payment data
- Returns
- Loyalty card data
- Signups
- Subscriptions
- Reservations
- Lending
- Trades

Transactional data

(Event data)

What about refunds?



Description qty amount
Description qty amount
Description qty amount
Disc. Item qty disc_amnt
item saving qty -amount
multi buy disc. qty -amount
Description qty 0.00
Description qty -amount
...

Total, payment type, cash taken, card type, card details (number?)



NLAB:

A case study: Till transactions

What about promotions?

Description qty amount
item saving qty -amount
Description qty amount
Description qty amount
...

Total, payment type, cash taken, card type, card details (number?)

Description qty amount
Disc. Item qty disc_amnt
Description qty amount
...

Total, payment type, cash taken, card type, card details (number?)



Seem simple enough...

Description qty amount

Description qty amount

Description qty amount

...

Total, cash taken, change

What about credit cards?

Seem a little more complex...

Description qty amount

Description qty amount

Description qty amount

Total, payment type, cash taken, card type, card details (number?)

What about prepaid items?

Description qty amount
Description qty amount
Description qty amount
Disc. Item qty disc_amnt
item saving qty -amount
multi buy disc. qty -amount
Description qty 0.00
...

Total, payment type, cash taken, card type, card details (number?)

TESCO

LINCOLN 2 0845 6779423

How did we do?
Visit www.tescocomments.co.uk and tell us about your shopping trip

RED LETCS	1.75
SUMMER FRUIT'S	0.66
FRESH MILK	0.25
T / MEAL BREAD	0.49
TOLET 4 ROLL	1.33
TINNED FRUIT	0.43
LIGHTER CHEESE	0.49
CHOCOLATE	0.86
MAYONNAISE	0.48
VALUE 12 ROLLS	0.35
SPRING WATER *	0.11
CHEESE SPREAD *	1.30
CHEESE SPREAD	1.30

...
Total, payment type, cash taken, card type, card details (number?)

What about multi-buys/meal deals?

Description qty amount
Description qty amount
Description qty amount
Disc. Item qty disc_amnt
item saving qty -amount
multi buy disc. qty -amount

Total, payment type, cash taken, card type, card details (number?)



RED LETCS	1.75
SUMMER FRUIT'S	0.66
FRESH MILK	0.25
T / MEAL BREAD	0.49
TOLET 4 ROLL	1.33
TINNED FRUIT	0.43
LIGHTER CHEESE	1.84
CHOCOLATE	0.86
MAYONNAISE	0.48
VALUE 12 ROLLS	0.35
SPARKLING WATER *	0.16
CHEESE SPREAD	1.30
CHEESE SPREAD	1.30

Sub-total: £12.18
Card sales: £0.48
Total saving today: £2.79

MULTIBUY SAVINGS	
T SELECTED CHEESE	£0.59
PRUNELLA 150G	-0.60
Total savings	-1.19



In a hurry? Self-service kiosk offer
Get a card at check-out. Any Retail
Associate can show you how.
www.tesco.com/shop-on-line

Data at Scale

Dr Evgeniya Lukinova

Transactional data

(Event data)



A case study: Till transactions

```
Description qty amount
Description qty amount
Description qty amount
Description qty amount
Disc. Item qty disc_amt
item saving qty -amount
multi buy disc. qty -amount
Description qty 0.00
Description qty -amount
```

...
Total, payment type, cash taken, card type, card details
(number?)

Actually looks
like

```
Description qty amount
Description qty amount
Description qty amount
Description qty amount
Description qty -amount
Description qty 0.00
Description qty -amount
```

...
Total, payment type, cash taken, card type, card details
(number?)

Take home message: In real world analytics understanding the business practice and process is **exceptionally important**.

Standard potentially incorrect assumption:
negative values are refunds.

Garbage in, garbage out.

Transactional data

(Event data)



and more...

Refunds,
multibuy,
promos,
prepaid

Description qty amount
Description qty amount
Description qty amount
Disc. Item qty disc_amt
item saving qty -amount
multi buy disc. qty -amount
Description qty 0.00
Description qty -amount
...
Total, payment type, cash taken, card type, card details
(number?)

What about
loyalty points?

Payment by
part card, part
cash?

Payment part
by points?

CLUBCARD STATEMENT		
CLUBCARD NUMBER:	123456789012345678	21
TOTAL NUMBER OF VISIT	10	
INCLUDES :	10	
DOUBLE POINTS	1	
GROCERIES AND BAG RE-USE	6/8	
TOTAL UP TO 10/05/11	8	
TOTAL INCLUDES :		
GREEN CLUBCARD POINTS		

But wait, there's more...

Actually looks like

Description qty amount
Description qty amount
Description qty amount
Description qty amount
Description qty -amount
Description qty -amount
Description qty 0.00
Description qty -amount
...

Total, payment type, cash taken, card type, card details
(number?)

What about discounts from loyalty
card coupons?

What about recording an identifier for
the promotion?

So more
info is
recorded

Description qty amount refund_flag collection_flag
Description qty 0.00 refund_flag collection_flag
Description qty -amount refund_flag collection_flag
...

Total, payment type, cash taken, card type, card details
(number?)

*In real world analytics
understanding the business
practice and process is
exceptionally important.*

Let's take a look at some real world
examples of what is recorded....

Transactional data

(Event data)

Back to receipts!

We really saw a simplistic
version.

More realistic
(just Receipt Table)

Refunds,
multibuys,
promos,
prepaid

Description qty amount
Description qty amount
Description qty amount
Disc. item qty disc_amt
item saving qty -amount
multi buy disc. qty -amount
Description qty 0.00
Description qty -amount
...
Total, payment type, cash taken, card type, card details
(number?)

Example: Shop receipts

Receipt Table (Relation)

Receipt ID	Cust ID	Date	Shop	Total
0001	0001	1/1/17	NC1N 2HT	£1.70

Facts (about a receipt)

Field Name	Description	Example
loyalty_number	Customer number (may not exist)	783252
epos_transaction_id	Unique transaction ID?	43586329
till_transaction_type_code	Sale (0), Refund (1), Cancellation (2)....	1
receipt_date	Date of transaction	2017-04-17
receipt_time	Time of transaction	14:46:00
staff_id	Identifier of the staff member that served the	87
discount_card_number	card number where applicable - otherwise 0.	0
store_number	Identifier of the store	313
points_change	Loyalty card points earned or spent	28
number_of_deals	Number of deals in basket	0
total_inc_vat	Total spend including VAT	2.99
total_exc_vat	Total spend before VAT	2.4916
sales_units	Number of items in basket. This can be negative (due to refunds).	1
deal_savings_local	Total savings due to store level deals	0.00
deal_savings_global_promo	Total savings due to company wide promotions	0.00
savings_coupons	Total savings due to redeemed coupons	0.00
register_number	The register that the transaction was done on	32
payment_card	Payment amount done on card	2
payment_cash	Payment amount taken in cash	5
bar_code	Bar code printed on receipt	6686586581583500135

Receipt Line Table (Relation)

Receipt ID	Line no.	Desc.	Qty	Amt
0001	0001	JS CARROTS 1.5KG	1	£0.70
0001	0002	THE TIMES	1	£1.00

Facts (about a line on a receipt)

Data at Scale



NLAB:

Dr Evgeniya Lukinova

Transactional data

(Event data)

Example: Shop receipts

Receipt Table (Relation)

Receipt ID	Cust ID	Date	Shop	Total
0001	0001	1/1/17	NC1N 2HT	£1.70

Facts (about a receipt)

Field Name	Description	Example
loyalty_number	Customer number (may not exist)	783252
epos_transaction_id	Unique transaction ID?	43586329
till_transaction_type_code	Sale (0), Refund (1), Cancellation (2)....	1
receipt_date	Date of transaction	2017-04-17
receipt_time	Time of transaction	14:46:00
staff_id	Identifier of the staff member that served the	87
discount_card_number	card number where applicable - otherwise 0.	0
store_number	Identifier of the store	313
points_change	Loyalty card points earned or spent	28
number_of_deals	Number of deals in basket	0
total_inc_vat	Total spend including VAT	2.99
total_exc_vat	Total spend before VAT	2.4916
sales_units	Number of items in basket. This can be negative (due to refunds).	1
deal_savings_local	Total savings due to store level deals	0.00
deal_savings_global_promo	Total savings due to company wide promotions	0.00
savings_coupons	Total savings due to redeemed coupons	0.00
register_number	The register that the transaction was done on	32
payment_card	Payment amount done on card	2
payment_cash	Payment amount taken in cash	5
bar_code	Bar code printed on receipt	6686586581583500135

Receipt Line Table (Relation)

Receipt ID	Line no.	Desc.	Qty	Amt
0001	0001	JS CARROTS 1.5KG	1	£0.70
0001	0002	THE TIMES	1	£1.00

Facts (about a line on a receipt)

Field Name	Description	Example
loyalty_number	Customer number (may not exist)	783252
transaction_id	Unique transaction ID	43586329
item_id	Unique item identifier	741321
store_number	Unique store ID	96
receipt_date	Date of transaction	2017-04-17
receipt_time	Time of transaction	14:46:00
transaction_type	Sale (0), Refund (1), Cancellation (2)....	1
qty	Quantity of this item purchased	2
total_inc_vat	Receipt line total including VAT	4.29
total_exc_vat	Receipt line total excluding VAT	3.6508
discount	Amount the item was discounted	0.0
refund_flag	Is this being refunded Y/N	Y
price_overridden	Was the price overridden by the register staff? Y or N	N
total	Total amount for the line on the receipt	0.0000
loyalty_points	Total loyalty points (may be negative) for this receipt line	0.0000
sap_sales_at_tisp		0.0000



Data at Scale

Transactional data

(Event data)



Example: Shop receipts

Receipt Table (Relation)

Receipt ID	Cust ID	Date	Shop	Total
0001	0001	1/1/17	NCIN 2HT	£1.70

Facts (about a receipt)

Receipt Line Table (Relation)

Receipt ID	Line no.	Desc.	Qty	Amt
0001	0001	JS CARROTS 1.5KG	1	£0.70
0001	0002	THE TIMES	1	£1.00

Facts (about a line on a receipt)

Large retailer
~ 250 million
transactions per year
within the UK



Product Detail Data

Other directly related tables:

- Customer
- Item
- Store
- Payment information

Other indirectly related tables:

- Brand
- Sub Brand
- Category

Other complications:

- information change over time (changes in facts)
hard for (any) paradigm to handle efficiently
(correction vs updates?)

Context is everything... (so let's take a look)

Pick the
processing & storage
paradigm* for the job.

(* we now know two!)

Is a traditional (relational)
approach enough?



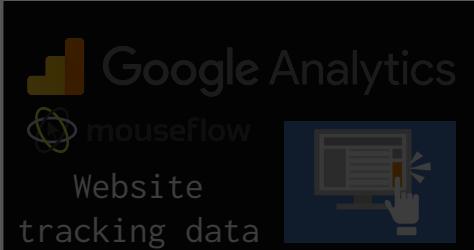
Transaction data



Product Detail Data



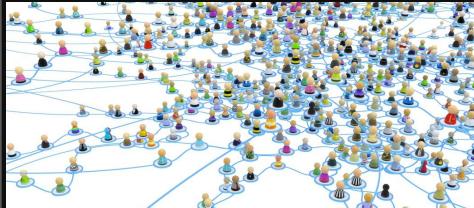
Call Detail Record (CDR) Data



Website
tracking data



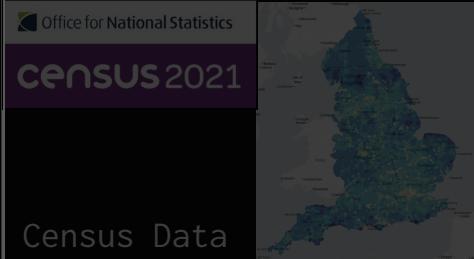
Survey Data



Social Media Data

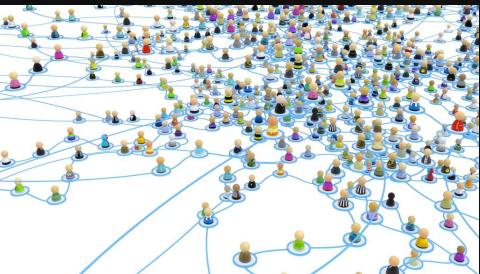


Security logs



Census Data

Twitter/X data

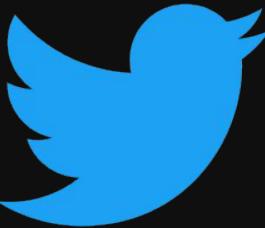


coordinates
favorited
truncated
created_at
id_str
entities

- urls
- Hashtags
- user_mentions

in_reply_to_user_id_str
contributors
text
retweet_count
place
user [...]
source

Standard Tweets



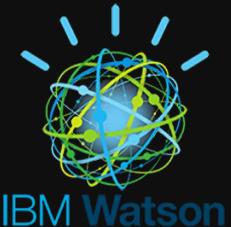
Extra information from IBM Watson Analytics

Sentiment	+ve, -vd, neutral, ambivalent, unknown
Sentiment positive signals	["great", "awesome"]
Sentiment negative signals	["poor", "terrible"]

Total Number of Tweets sent per Day:

500 million

Last updated: 03/06/23



id
created_at
recipient_id
recipient_screen_name
sender_id
sender_screen_name
text
entities [hashtags, urls, user_mentions]

recipient

- id
- created_at
- description
- followers_count
- profile_image_url
- geo_enabled
- verified
- ... (38 total)

sender

- id
- created_at
- description
- followers_count
- profile_image_url
- geo_enabled
- verified
- ... (38 total)

Context is everything... (so let's take a look)

Pick the
processing & storage
paradigm* for the job.

(* we now know two!)

Is a traditional (relational)
approach enough?

The slide features a central collage of nine images representing different data types:

- Google Analytics**: A wall of small video screens showing people interacting with products.
- mouseflow**: A logo with a yellow circle and a blue arrow.
- Website tracking data**: An icon of a computer monitor with a bar chart and a hand cursor.
- Product Detail Data**: A Coca-Cola bottle with a complex network diagram overlaid, showing internal and external components.
- Survey Data**: A hand holding a green button over a grid of smiley and frowny faces.
- Census Data**: The Office for National Statistics logo and a map of the United Kingdom colored by census data.
- Call Detail Record (CDR) Data**: Five small portraits of people looking at their phones.
- Social Media Data**: A network graph with many small human icons connected by lines.

Web tracking data

(Event data)



Website
tracking data

Server generating the page can
track what it sends.

Code running in browsers can send
logs back.

referrer_url
page_url
location
IP
Browser
Operating System
Language settings
Screen resolution

GUID (via cookie, javascript etc)

Optional, if event:

click information (what clicked, etc)
form submission (event, success code)
mouse movement (I.e. mouseflow)

Pretty well everything can be
recorded

Web tracking data

(Event data)



Website
tracking data

Server generating the page can
track what it sends.

Code running in browsers can send
logs back.

referrer_url
page_url
location
IP
Browser
Operating System
Language settings
Screen resolution

GUID (via cookie, javascript etc)

- Track sessions
 - GUID
 - User login

Challenge:
Multiple tabs, non-linear navigation flow

Optional, if event:
click information (what clicked, etc)
form submission (event, success code)
mouse movement (I.e. mouseflow)

Pretty well everything can be
recorded

Web tracking data

(Event data)



Website tracking data

Server generating the page can track what it sends.

Code running in browsers can send logs back.

referrer_url

page_url

location

IP

Browser

Operating System

Language settings

Screen resolution

GUID (via cookie, javascript etc)

Optional, if event:

click information (what clicked, etc)

form submission (event, success code)
mouse movement (i.e. mouseflow)

Pretty well everything can be recorded

Google Analytics

Track sessions

- GUID
- User login

Challenge:

Multiple tabs, non-linear navigation flow

Dwell time

Bounce rate (visit 1 page only)

Conversion rate (all vs. geo vs....)

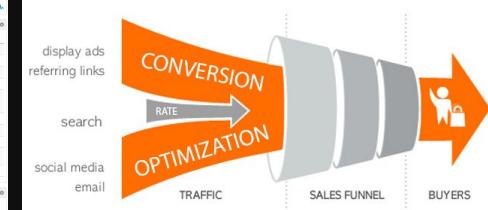
Traffic source statistics

Unique vs return visitors

Traffic flow (cart abandonment)

Search terms on site

Percentage of desktop/mobile use



Web tracking data

(Event data)



Website tracking data

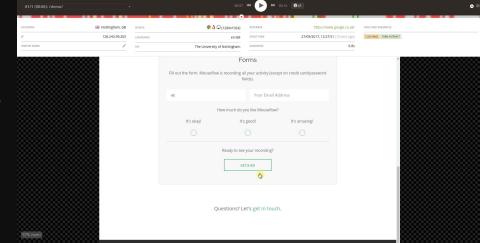
- Server generating the page can track what it sends.
- Code running in browsers can send logs back.

referrer_url
page_url
location
IP
Browser
Operating System
Language settings
Screen resolution

GUID (via cookie, javascript etc)

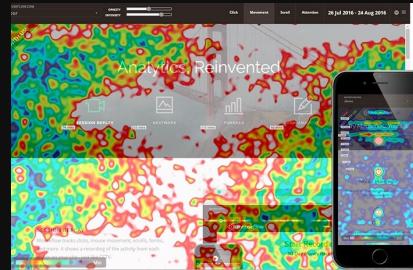
Optional, if event:
click information (what clicked, etc)
form submission (event, success code)
mouse movement (i.e. mouseflow)

Pretty well everything can be recorded



Site design

- engagement
- promotion placement
- page simplification

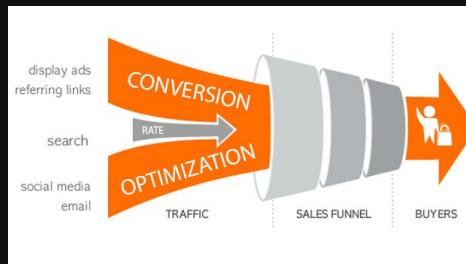


Retention analysis & redesign

Conversion analysis & redesign

Dwell time++

Ad campaign evaluation



Context is everything... (so let's take a look)

Pick the
processing & storage
paradigm* for the job.

(* we now know two!)

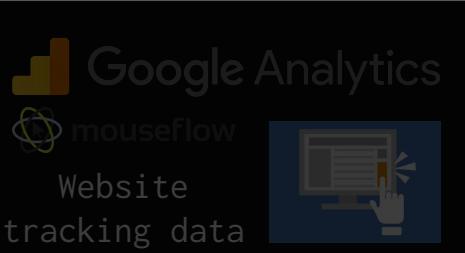
Is a traditional (relational)
approach enough?



Transaction data



Product Detail Data



Website
tracking data



Survey Data



Call Detail Record (CDR) Data



Social Media Data



Security logs



Census Data



Call Detail Record (CDR) Data

(More event data...)



Calls



start time
tower id
caller id
called id
tariff type
prepaid balance
product id
...
duration
charge
serviceflow
roaming
result code
incoming
...

Initiating subscriber ID	Unique BTS ID	Timestamp
jgsmi13227abc	12038097523	14-03-2014 00:01:12

Call Detail Record (CDR) Data

(More event data...)



Calls



start time
tower id
caller id
called id
tariff type
prepaid balance
product id
...

duration
charge
serviceflow
roaming
result code
incoming
...

Initiating subscriber ID	Unique BTS ID	Timestamp
jggsml13227abc	12038097523	14-03-2014 00:01:12

SMS



start time
tower id
caller id
called id
tariff type
prepaid balance
product id
...

charge
serviceflow
roaming
result code
incoming
sms length
...

Call Detail Record (CDR) Data

(More event data...)



Calls



start time
tower id
caller id
called id
tariff type
prepaid balance
product id
...

duration
charge
serviceflow
roaming
result code
incoming
...

Initiating subscriber ID	Unique BTS ID	Timestamp
jggsml13227abc	12038097523	14-03-2014 00:01:12

SMS



start time
tower id
caller id
called id
tariff type
prepaid balance
product id
...

charge
serviceflow
roaming
result code
incoming
sms length
...

Data



start time
tower id
caller id
tariff type
prepaid balance
product id
duration
data up
data down
...

Context is everything... (so let's take a look)

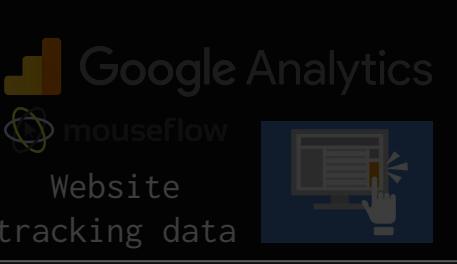
Pick the
processing & storage
paradigm* for the job.

(* we now know two!)

Is a traditional (relational)
approach enough?



Transaction data



Security logs



Product Detail Data



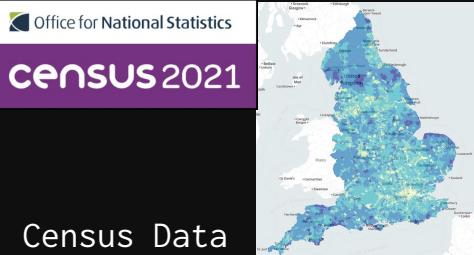
Call Detail Record (CDR) Data



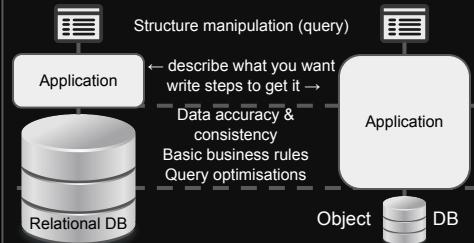
Survey Data



Social Media Data



Census Data



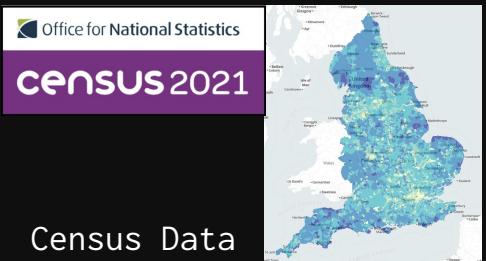
To summarize...



Survey Data



Security logs



Census Data

Pick the
processing & **storage**
paradigm* for the job.
(* we now know two!)



Facebook

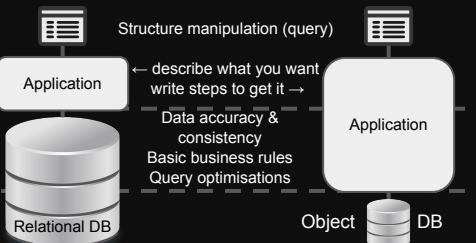


Credit Card Transactions



Plane Engine Maintenance

Some hints:
(from prior slides)



Are we going to process objects as stored & if objects have identical/correct structure & data is error free (or we do not care)?

Use relational
DBs unless...

Too much data makes checks /access too slow.

Data is always in, and processed in, objects*.

Data structure changes constantly.

Session 3

Graph types & Dashboards

The when, where & how



This week:

Graphs. When, why & how.

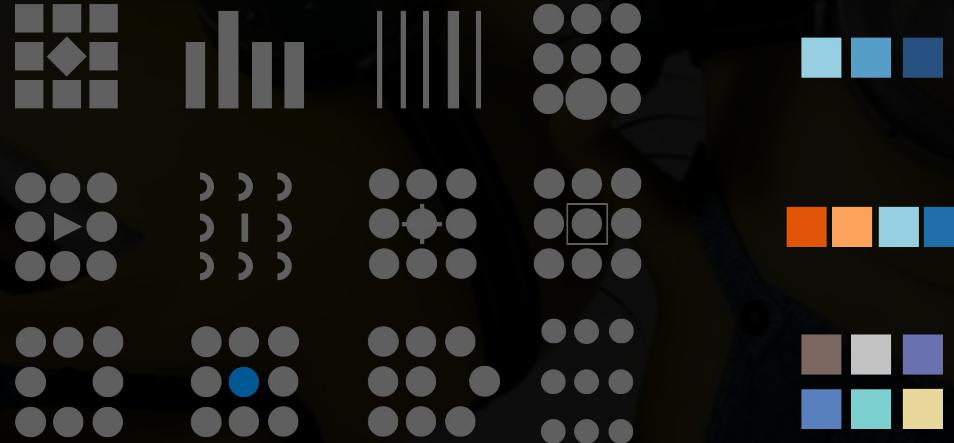
Dashboards.



Graphs - visual version of our data!

"Above all else, show the data! Graphics is *intelligence made visible*" Edward Tufte

Aim to "abuse" preattentive attributes of visual perception and use of colour.



Graphs - visual version of our data!

"Above all else, show the data! Graphics is *intelligence made visible*" Edward Tufte

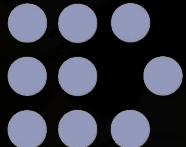
Precise Quantitative Comparisons



Length or Width

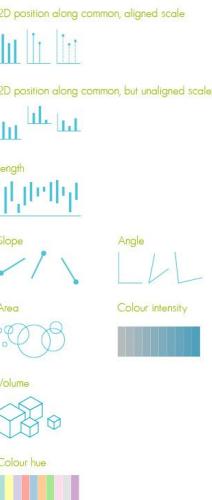
More analytical, higher in the chart (more standard).

Compromise between accuracy and visual interest required for the particular story.



2D Position

Allows more accurate comparisons

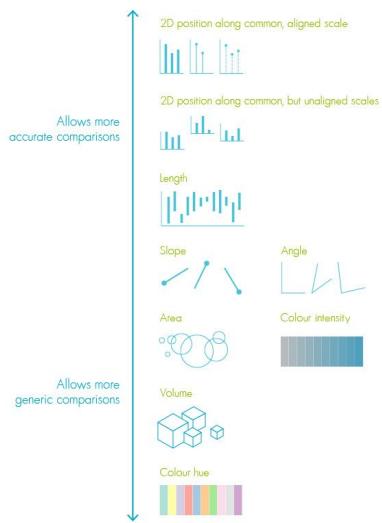
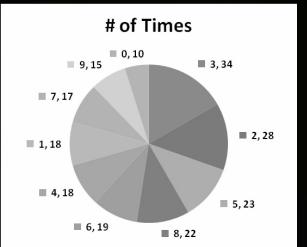


Allows more generic comparisons

Graphs - visual version of our data!

"Above all else, show the data! Graphics is *intelligence made visible*" Edward Tufte

Precise Quantitative Comparisons



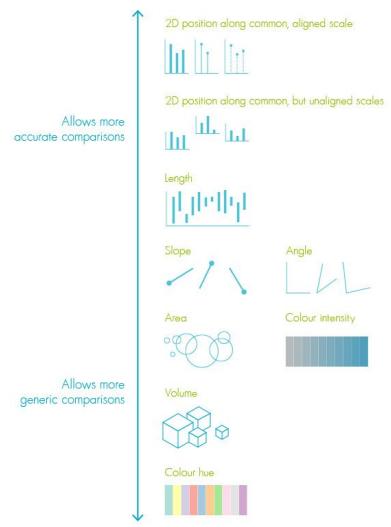
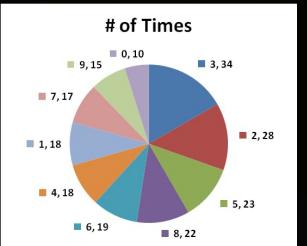
Graphs - visual version of our data!

"Above all else, show the data! Graphics is *intelligence made visible*" Edward Tufte

Precise Quantitative Comparisons



7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
5 2 8 3 6 1 9 3 6 2 5 3 4 3 8 3 6
5 8 9 6 2 1 4 4 3 9 3 6 5 2 4 9 1
0 2 5 2 8 3 6 1 6 2 9 3 8 3 8 5 8
4 2 0 3 3 5 4 1 8 2 0 1 2 5 3 6 4
3 9 1 0 8 9 5 3 4 5 3 2 5 2 8 3 6
1 6 2 9 3 8 3 8 5 8 4 2 0 3 3 5 4
1 8 2 0 1 9 6 2 1 4 4 3 9 3 6 5 2
4 9 1 0 2 5 2 8 3 6 1 6 2 9 3 8 3
8 5 4 8 2 0 3 5 4 1 8 2 0 1 2 5
3 6 4 3 9 1 0 8 9 5 3 4 5 3 2 5 6
8 3 6 1 6 2 4 6 2 5 9 1 5 2 6 3 6



Graphs - visual version of our data!

"Above all else, show the data! Graphics is *intelligence made visible*" Edward Tufte

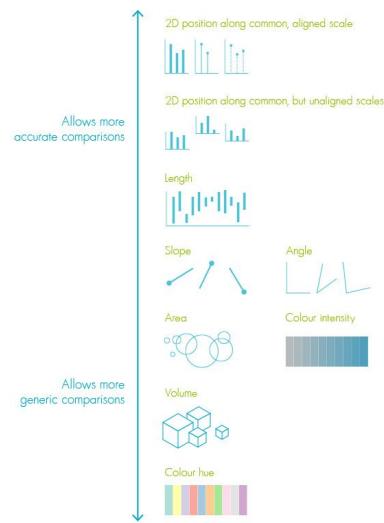
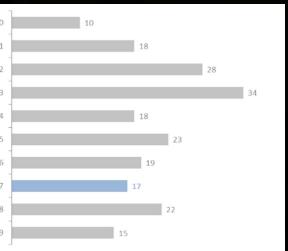
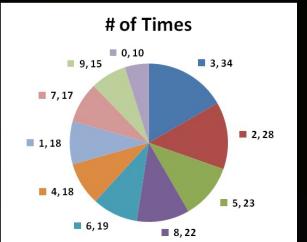
Precise Quantitative Comparisons



Length or Width



2D Position



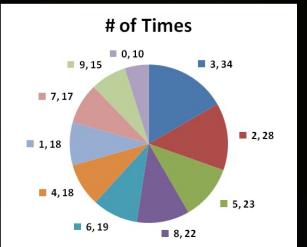
Graphs - visual version of our **data!**

"Above all else, show the data! Graphics is *intelligence made visible*" Edward Tufte

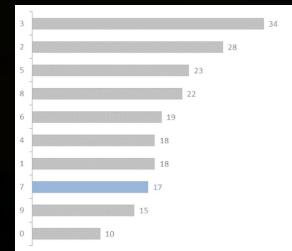
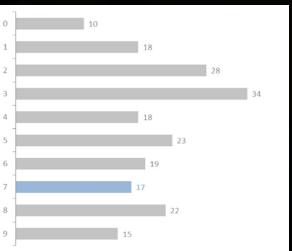
Precise Quantitative Comparisons



Length or Wid



2D Position

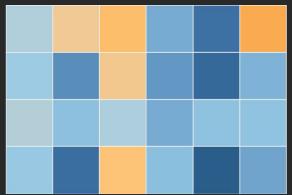
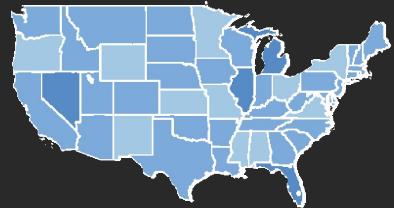
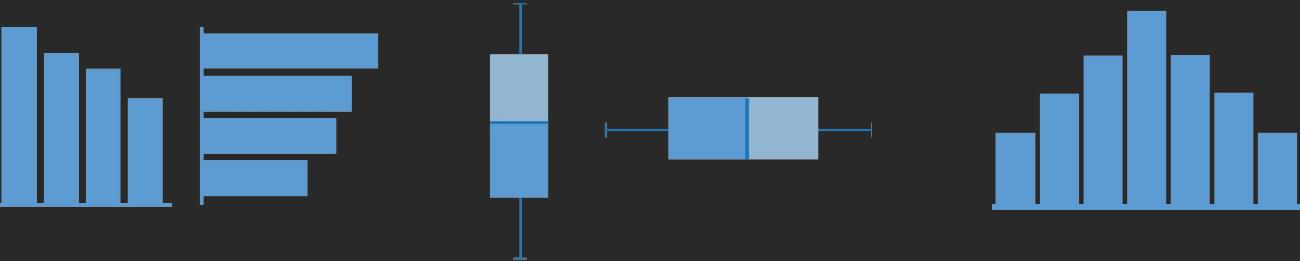


All graphs give the same information.

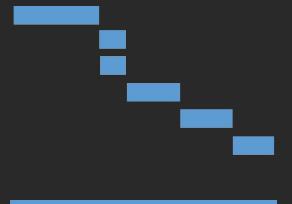
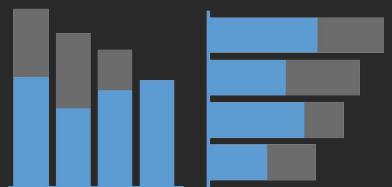
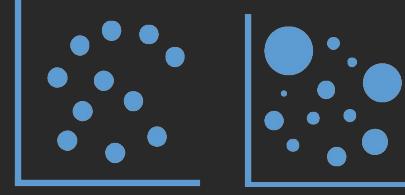
Clearly:
length + order comparison -
more precise comparison
angle, area and colour
comparison -
faster comparison

Allows more accurate comparisons

Common Types of Charts



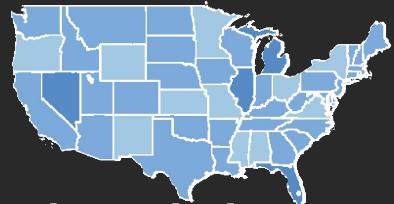
That you probably
recognise...



\$29,071	\$17,307	\$30,073
\$2,603	\$2,353	\$5,079
\$66,106	\$53,891	\$42,444
\$20,173	\$14,151	\$26,664
\$100,615	\$58,304	\$98,684
\$71,613	\$35,768	\$70,533
\$10,760	\$8,319	\$18,127
\$39,140	\$43,916	\$84,755



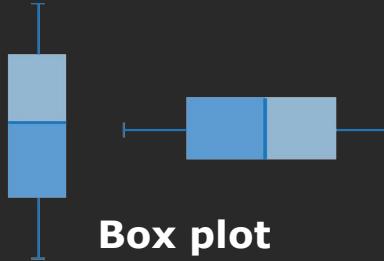
Common Types of Charts



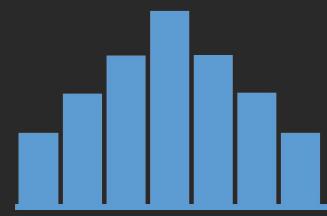
Choropleth map



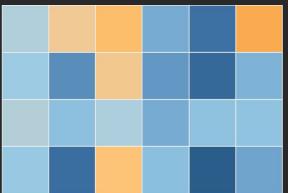
Bar chart



Box plot

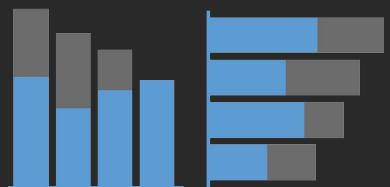


Histogram

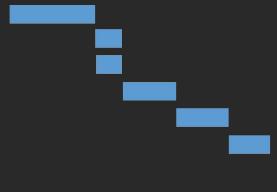


Heat map

That you probably
recognise...



Stacked Bar chart



Gantt chart

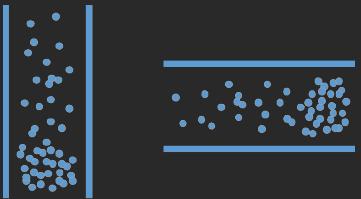
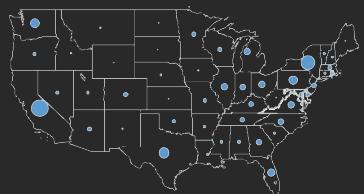
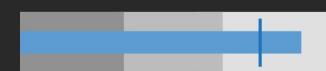
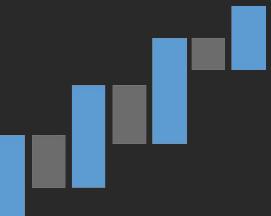
\$29,071	\$17,307	\$30,073
\$2,603	\$2,353	\$5,079
\$66,106	\$53,891	\$42,444
\$20,173	\$14,151	\$26,664
\$100,615	\$58,304	\$98,684
\$71,613	\$35,768	\$70,533
\$10,760	\$8,319	\$18,127
\$39,140	\$43,916	\$84,755

Highlight table

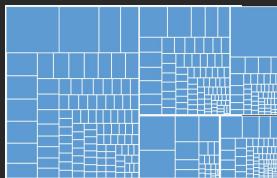
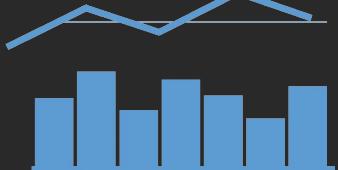
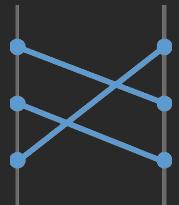
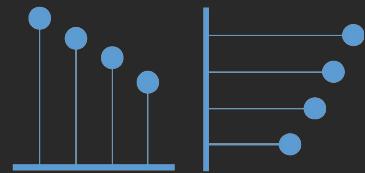


Line graph

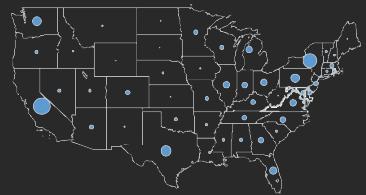
Common Types of Charts



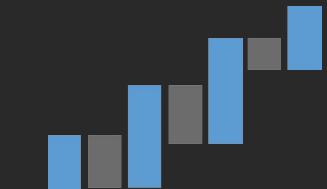
That you possibly
don't know...



Common Types of Charts



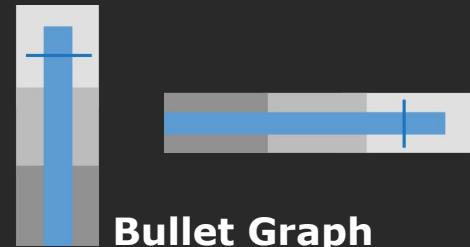
Symbol (dot) map



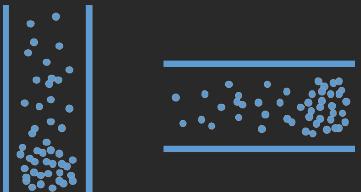
Waterfall chart



Dot plot

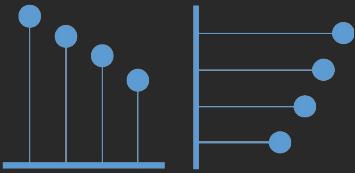


Bullet Graph

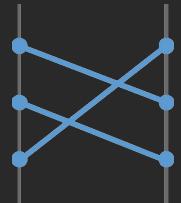


Dot plot with jitter

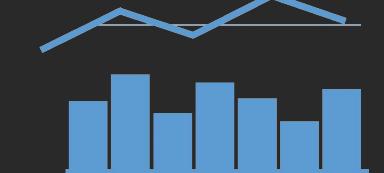
That you possibly
don't know...



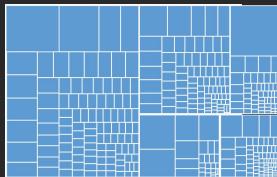
Lollipop chart



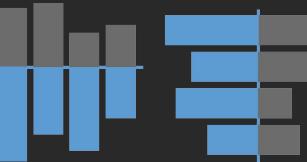
Slopegraph



Sparkline/sparkbar

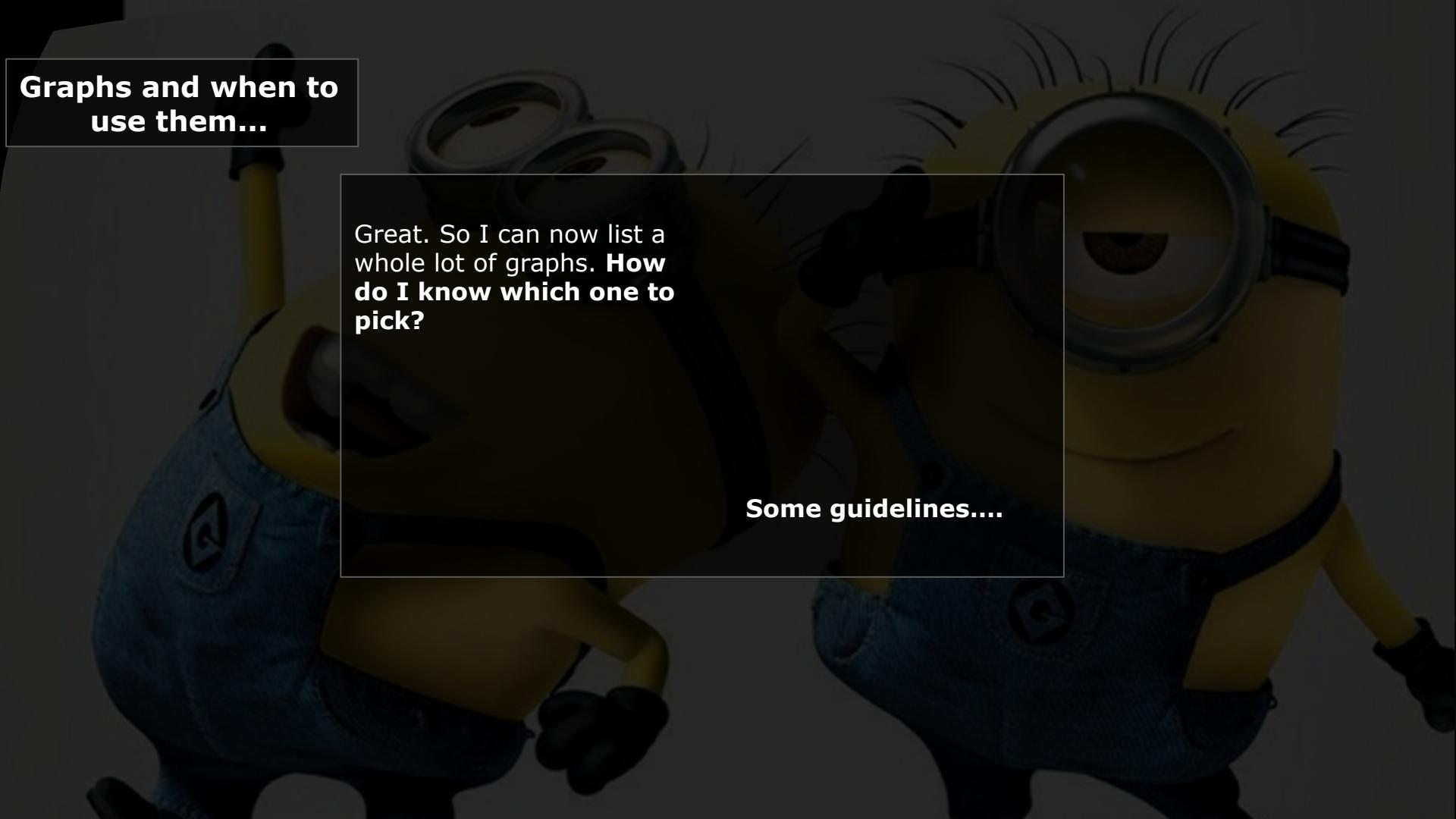


Treemap



Diverging bar chart

Graphs and when to use them...



Great. So I can now list a whole lot of graphs. **How do I know which one to pick?**

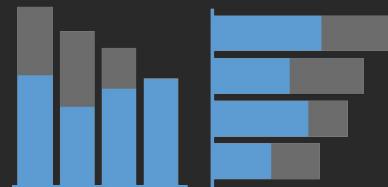
Some guidelines....

Task:

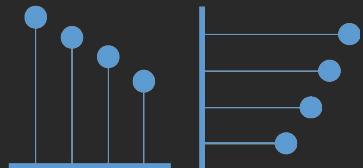
Comparing data across categories



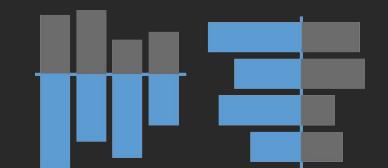
Bar chart



Stacked Bar chart



Lollipop chart



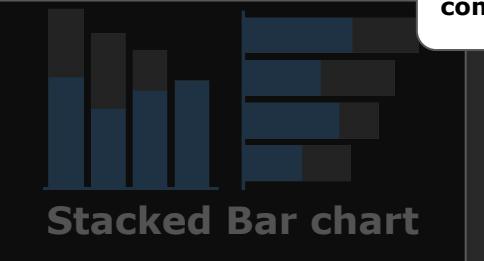
Diverging bar chart

Task:

Comparing data across categories



Bar chart



Stacked Bar chart

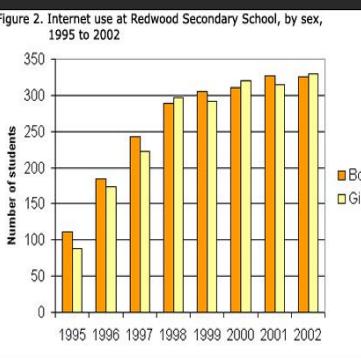
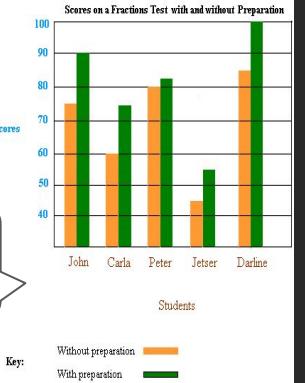


Lollipop chart



Diverging bar chart

May be
comparative.



The visual cue is **bar height**.

Bar width and spacing
do not represent
values.

Can show discrete (ordinal)
or continuous values
(with some kind of logical
binning).

x-axis can be inherently
ordered (e.g. time) or
ordered by value to aid
interpretation.

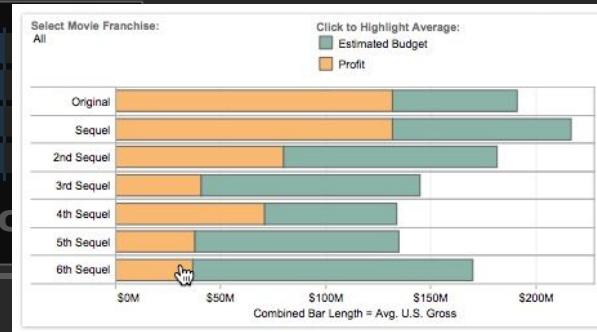
Task:

Comparing data across categories



Bars are coloured / shaded with proportions.

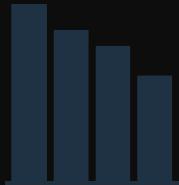
Caution: Don't slice into too many segments!



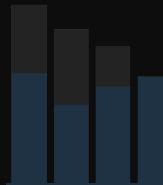
Are film sequels profitable? In this example of a bar chart, you quickly get a sense of how profitable sequels are for the box office franchises.

Task:

Comparing data across categories



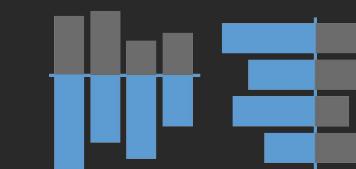
Bar chart



Stacked Bar chart

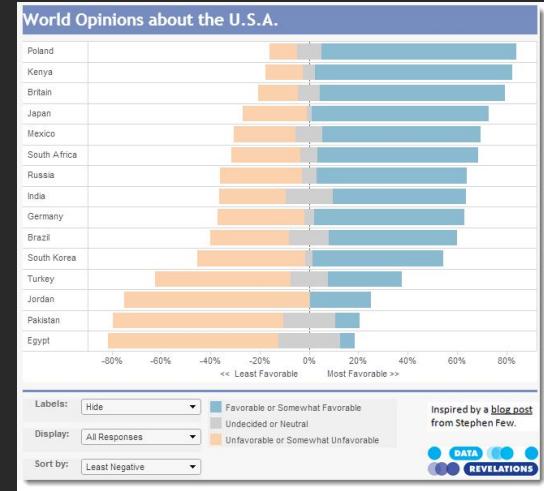


Lollipop chart



Diverging bar chart

Categorical comparisons where a mid-point is important.



Task:

Comparing data across categories



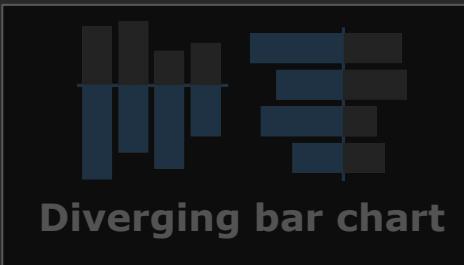
Bar chart



Stacked Bar chart



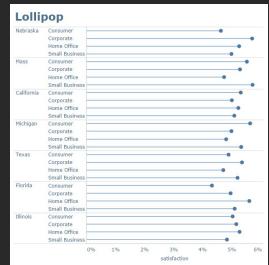
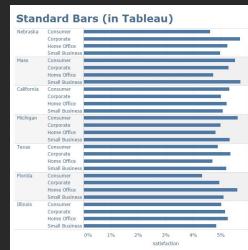
Lollipop chart



Diverging bar chart

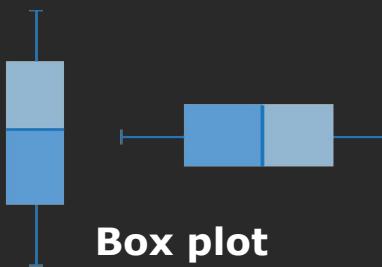
Use to get a different visual effect, particularly when there are many long bars (provides a better ink to data ratio*).

Do not use if many bars of same length - harder to compare than bar charts.



Task:

Showing / understanding
the distribution of your
data



Box plot



Histogram

Task:

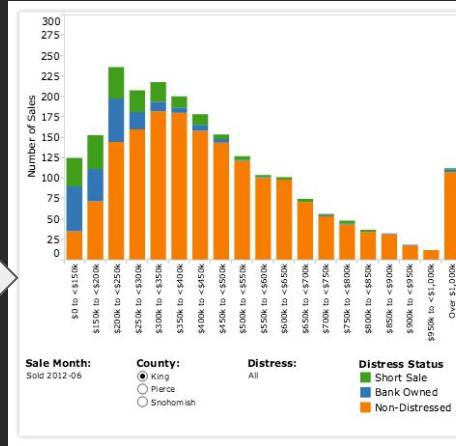
Showing / understanding
the distribution of your
data



Grouping is important!
(categorical data is easier than
continuous)

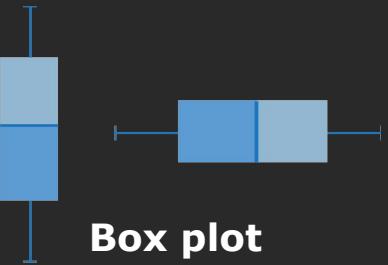
If interactive - add a filter
to enable users to drill
down into information.

Which houses are
selling? This histogram
shows which houses are
seeing the most sales in
a month.



Task:

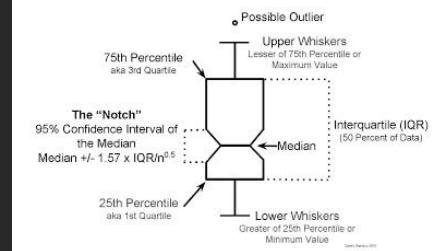
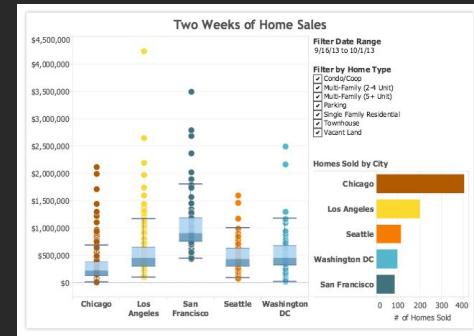
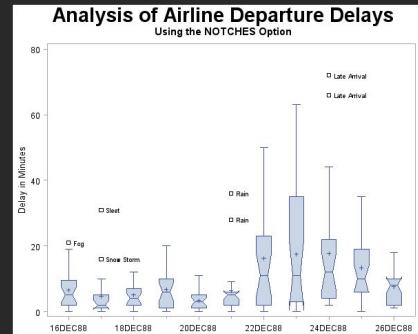
Showing / understanding
the distribution of your
data



Histogram

Quickly displays
distribution's median
(optionally mean),
quartiles, range and
outliers.

Can compare, can show
indications of statistical
significance.



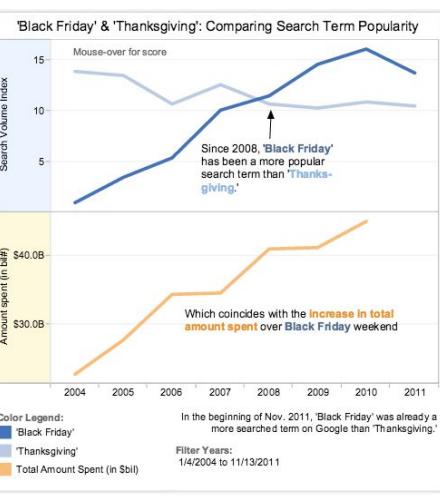
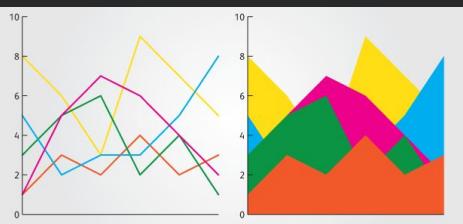
Task:

Viewing trends in data over time



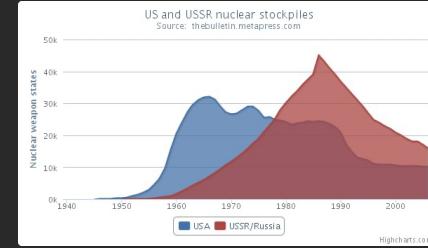
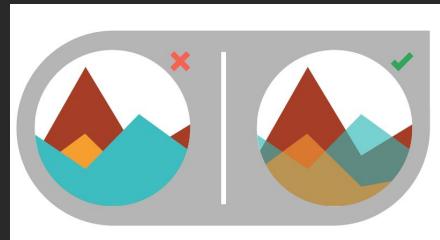
Line graph

Sometimes it might be worth filling area under the lines → **area chart**.
Maybe.



Basic lines reveal powerful insight. These two line charts illuminate the increasing popularity of "Black Friday" as an epic event in the United States.

It's quick to see that Thanksgiving lost ground to the popular shopping period in 2008.

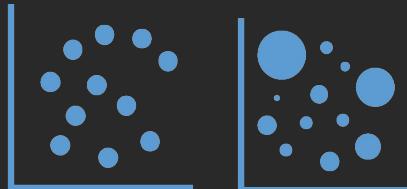


Task:

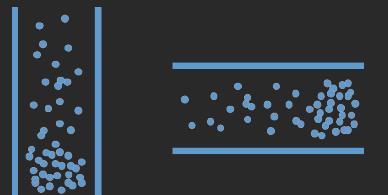
Investigating the relationship between different variables



Dot plot



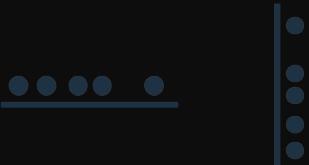
Scatter plot



Dot plot with jitter

Task:

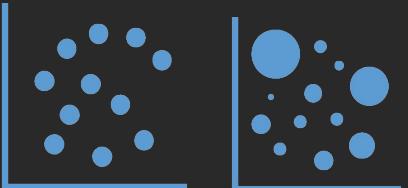
Investigating the relationship between different variables



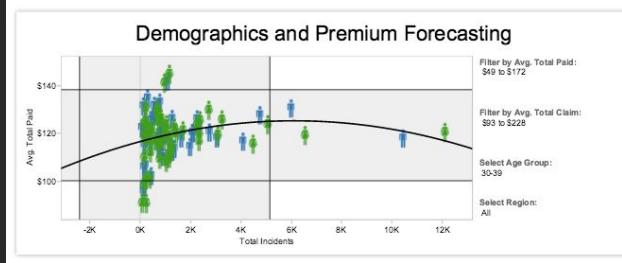
Dot plot



Dot plot with jitter



Scatter plot



Scatter plots show relationships between two variables (x-axis vs y-axis).

Can use colours and markers to give more information.

Can add trend lines etc.

If interactive can use filters.

Who is most expensive to insure? (green female, blue male: one marker per region).

Total incidents is the number of payouts.

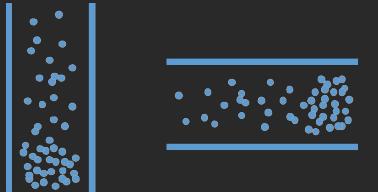
Avg paid is the premium paid.

Task:

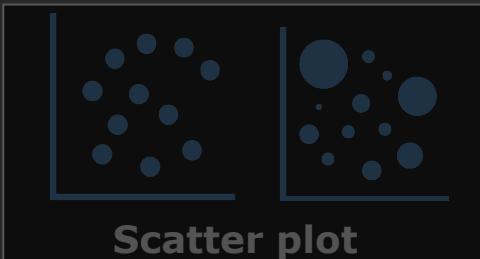
Investigating the relationship between different variables



Dot plot



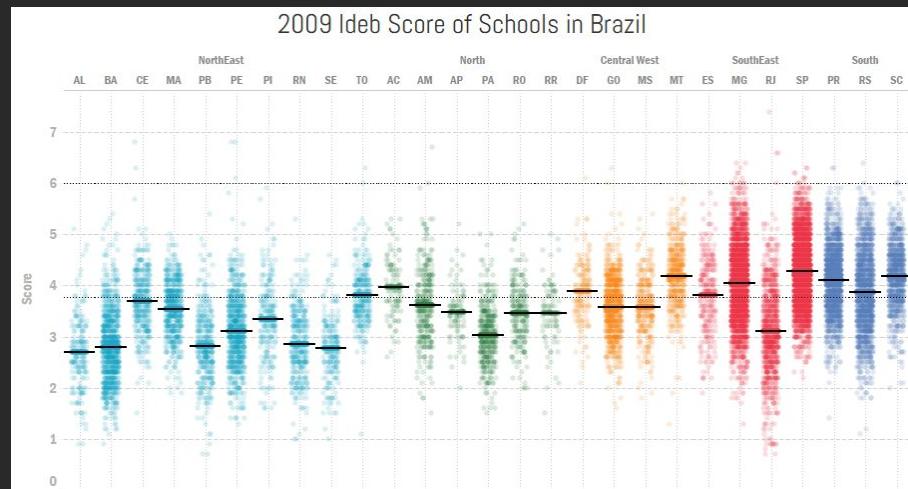
Dot plot with jitter



Scatter plot

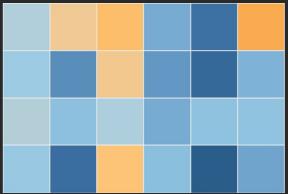
1D scatter plot.

Use jitter to highlight density and avoid data point overlap.



Task:

Showing the relationship between two factors.



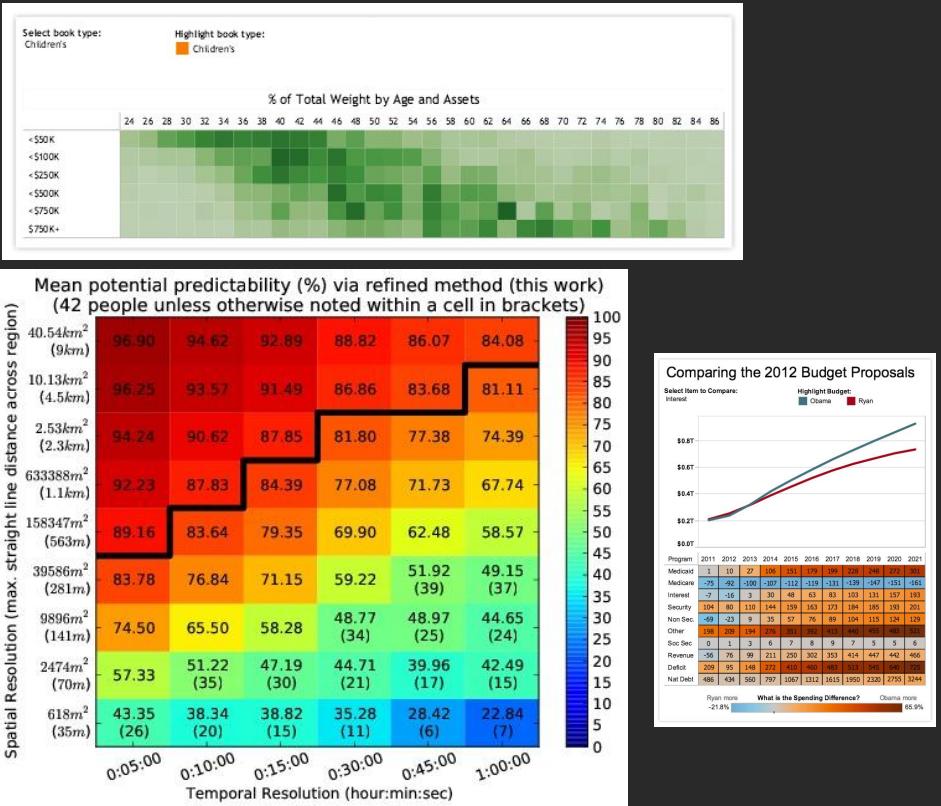
Heat map

\$29,071	\$17,307	\$30,073
\$2,603	\$2,353	\$5,079
\$66,106	\$53,891	\$42,444
\$20,173	\$14,151	\$26,664
\$100,615	\$58,304	\$98,684
\$71,613	\$35,768	\$70,533
\$10,760	\$8,319	\$18,127
\$39,140	\$43,916	\$84,755

Highlight table

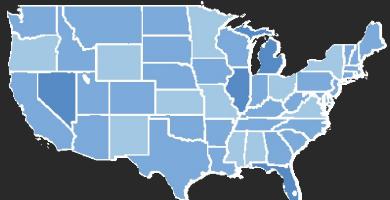
Can use the size of the square to show a second value, in addition to colour.

Consider combining with other chart types to show trends etc.

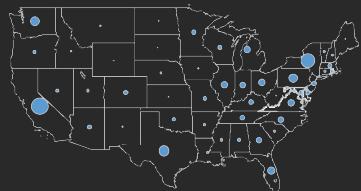


Task:

Showing geocoded
(located) data



Choropleth map

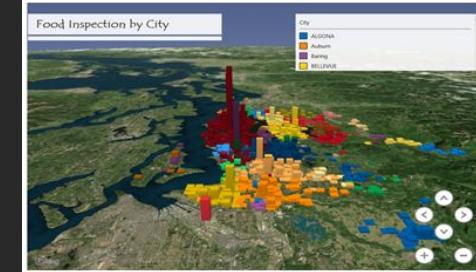
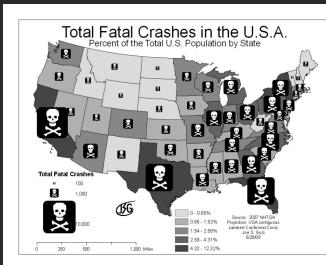
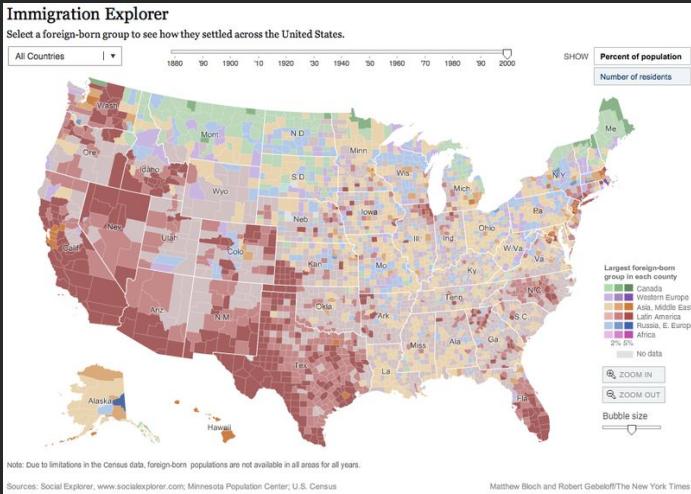


Symbol (dot) map

Use shading, colour and symbols to indicate values.

Use maps as a filter
(click to show) for other chart types.

Plot other chart types
(i.e. scatter) overtop of maps.



Task:

Displaying things in use over time.



Gantt chart

Task Name	Q1 2009			Q2 2009			Q3 2009		
	Dec '08	Jan '09	Feb '09	Mar '09	Apr '09	May '09	Jun '09	Jul '09	Aug
Planning									
Research									
Design									
Implementation									
Follow up									

E.g. Displaying a project's schedule.

In this case "things" are project parts.

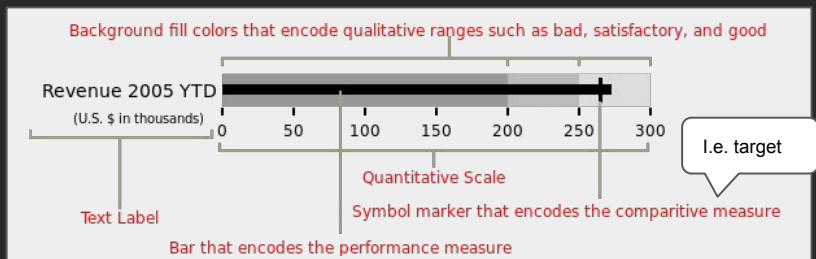


Task:

Evaluating performance of a metric against a goal



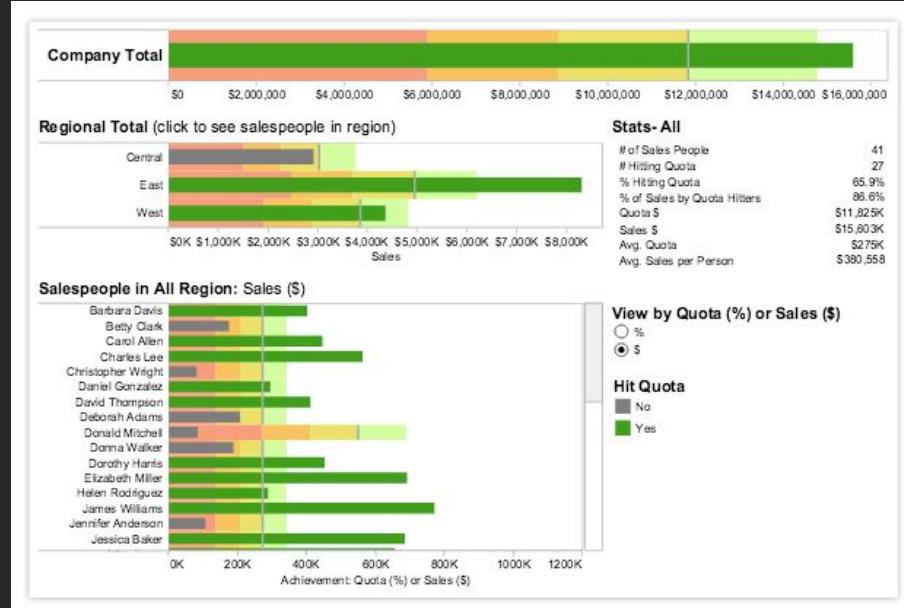
Bullet Graph



Uses colour to indicate achievement of thresholds.

Provides summary insights.

Tracking a sales team's progression to hitting its quota is a critical element to managing success.

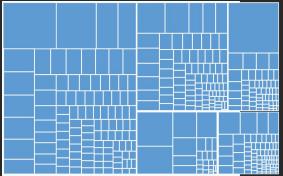


NLAB:

Data at Scale

Task:

Showing hierarchical data as a proportion of a whole



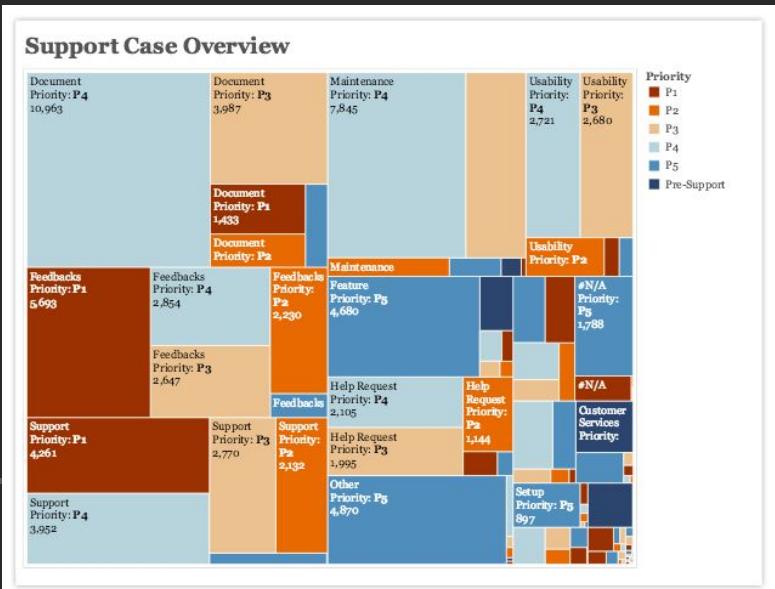
Treemap

Can also:

- Colour rectangles by category.
- Combine / embed treemaps with bar charts enabling **quantitative comparisons**.

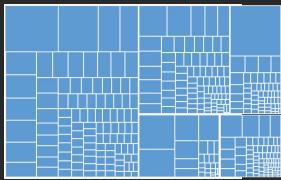
This treemap shows all of the company's support cases, broken by case type, and also priority level.

You can see that Document, Feedback, Support and Maintenance make up the lion share of support cases. However, in Feedback and Support, P1 cases make up the most number of cases, whereas most other categories are dominated by relatively mild P4 cases.



Task:

Showing hierarchical data as a proportion of a whole



Treemap

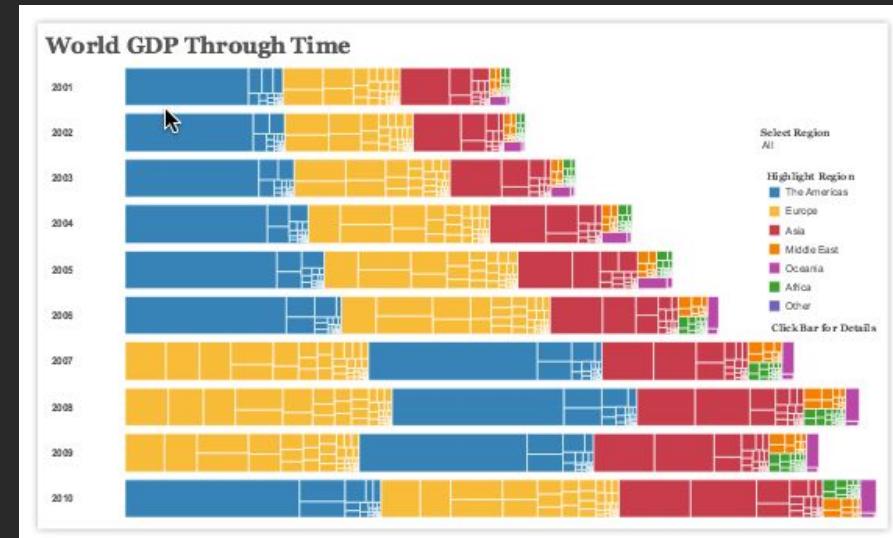
Can also:

- Colour rectangles by category.
- Combine / embed treemaps with bar charts enabling **quantitative comparisons**.

Visualizing World GDP.

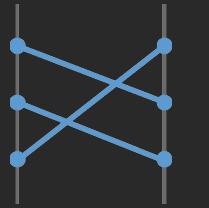
In this **treemap-bar combination chart**, we can see how overall GDP has grown over time (with the exception of 2009, when GDP fell),

Treemap shows region, then sub-region then country, etc.



Task:

Showing a comparison of rank (typically between two time periods)



Slopegraph

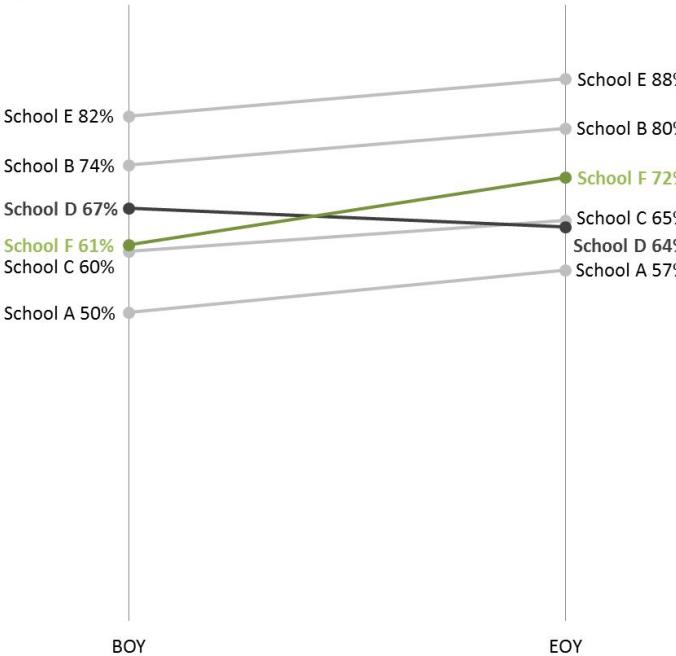
Typically shows how an entity changed rank over time.

Places focus strictly on the beginning and the end points.

Consider:

- using colour & thickness
- highlighting the most important changes with colour
- more than two time points

While most schools increased, only **School F** met growth targets. **School D** decreased.



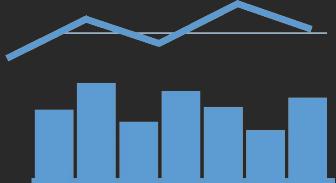
In this example, the slope graph compares school performance at the beginning of the year (BOY) to the end of the year (EOY).

Source:
Stephanie Evergreen

<http://www.betterevaluation.org/en/evaluation-options/slopegraph>

Task:

Show general trends / overall information, add extra context



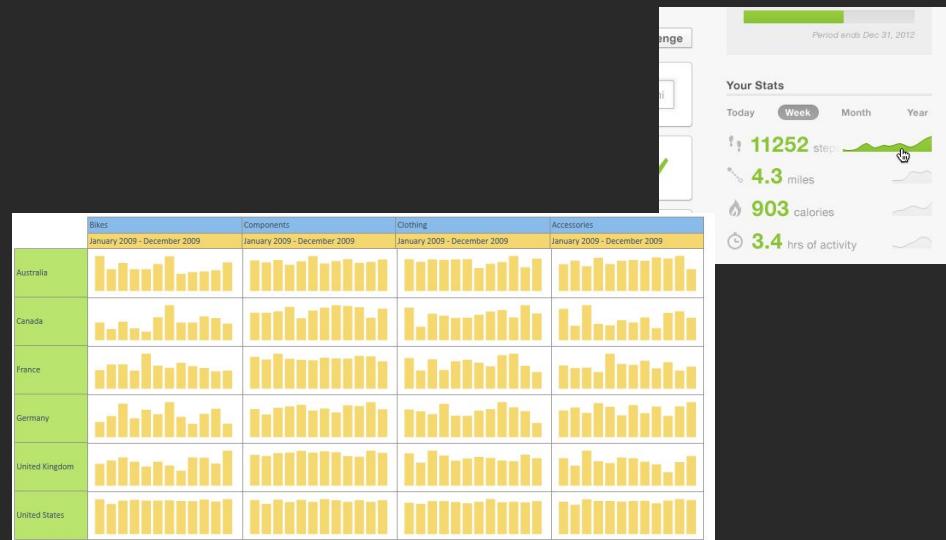
Sparkline/sparkbar

Small, word-sized graphic.

No: axis, labels etc.

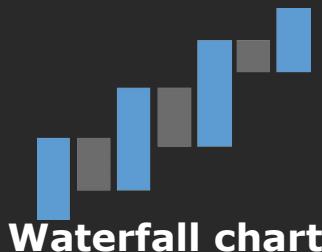
Context comes from related content.

Consider multiple sparkbars in a grid



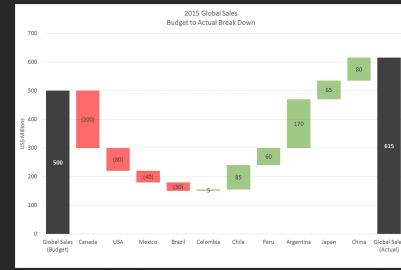
Task:

Showing the gradual transition (+/-) in the quantitative value



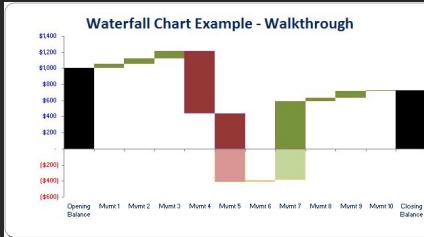
Waterfall chart

Modified bar chart showing **stepwise** increase / decrease in a value across the x-axis.



Typically the transition is across time, but could be between ordered categories.

Consider adding colour and data labels.

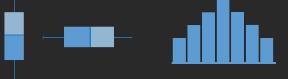




Comparing data across categories



Showing / understanding the distribution of your data



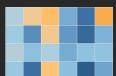
Viewing trends in data over time



Investigating the relationship between different variables



Showing the relationship between two factors



\$12,075	\$17,817	\$10,078
\$12,075	\$17,817	\$10,078
\$12,075	\$17,817	\$10,078
\$12,075	\$17,817	\$10,078
\$12,075	\$17,817	\$10,078
\$12,075	\$17,817	\$10,078
\$12,075	\$17,817	\$10,078
\$12,075	\$17,817	\$10,078
\$12,075	\$17,817	\$10,078
\$12,075	\$17,817	\$10,078

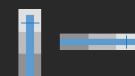
Showing geocoded (located) data



Displaying things in use over time



Evaluating performance of a metric against a goal



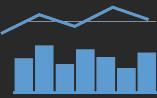
Showing hierarchical data as a proportion of a whole



Showing a comparison of rank (typically between two time periods)



Show general trends / overall information, add extra context



Showing the gradual transition (+/-) in the quantitative value



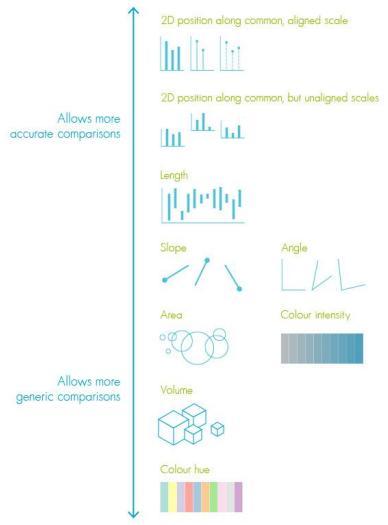
Common Types of Charts

Why? They ask the viewer to compare.

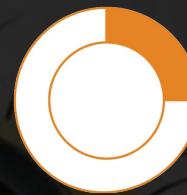
But **accurate comparisons** are hard.



Bubble chart



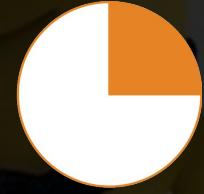
Concentric Circles



Donut Chart

That you **think twice
before using...**
(probably shouldn't use...)

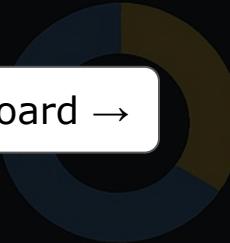
Least
Some
Most
More
Few
Word Cloud



Pie Chart

A quick introduction to: Dashboards

Dashboard →



A dashboard is a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance.

- Stephen Few (2004)

← Faceted analytical display

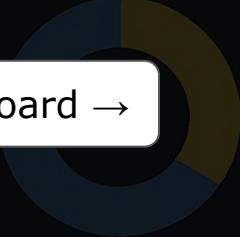


VS.



A quick introduction to: Dashboards

Dashboard →



A dashboard is a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance.

- Stephen Few (2004)



A dashboard is a visual display of data used to monitor conditions and/or facilitate understanding.

- The Big Book of Dashboards (2017)

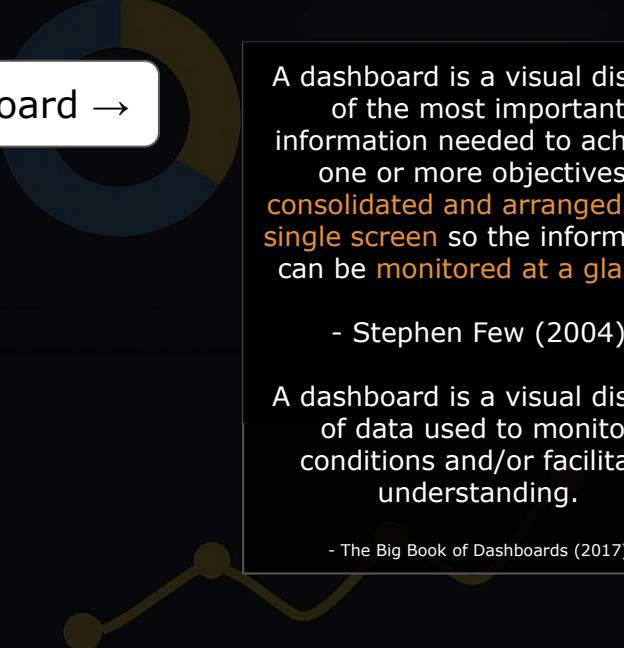


← Faceted analytical display

VS.

A quick introduction to: Dashboards

Dashboard →



A dashboard is a visual display of the most important information needed to achieve one or more objectives; **consolidated and arranged on a single screen** so the information can be **monitored at a glance**.

- Stephen Few (2004)

A dashboard is a visual display of data used to monitor conditions and/or facilitate understanding.

- The Big Book of Dashboards (2017)

A “faceted analytical display” is a set of interactive charts (primarily graphs and tables) that simultaneously reside on a single screen, each of which presents a somewhat different view of a common dataset, and is used to analyze that information.

- Stephen Few (2007)

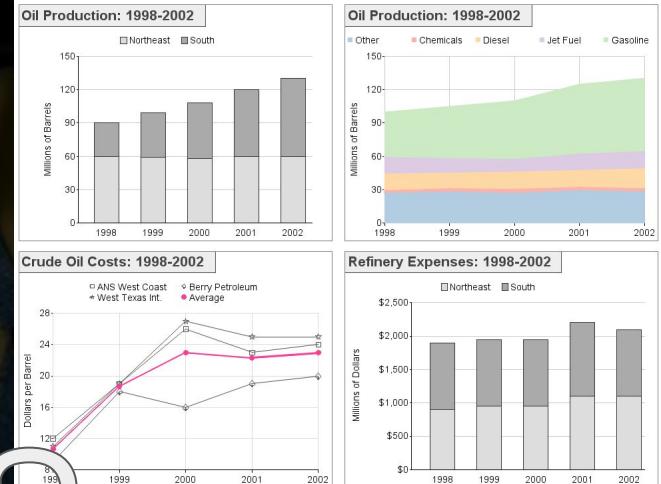
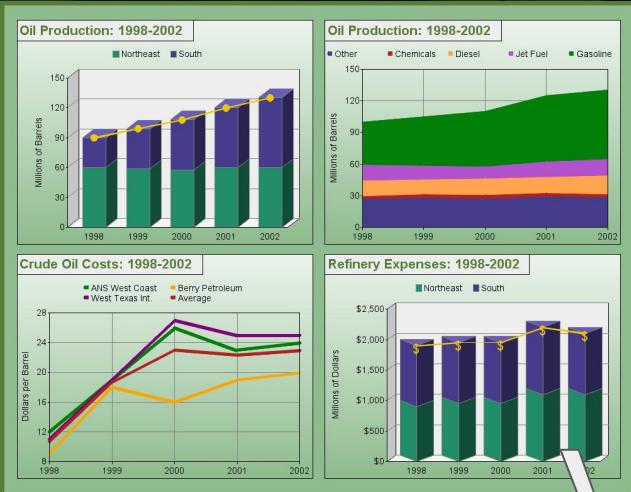
VS.

← Faceted analytical display

A quick introduction to: Dashboards

Presentation:

- clearly stated messages
- concise (data to ink ratio)
- direct
- customized to goals
- consistent layout - data changes over time, not layout

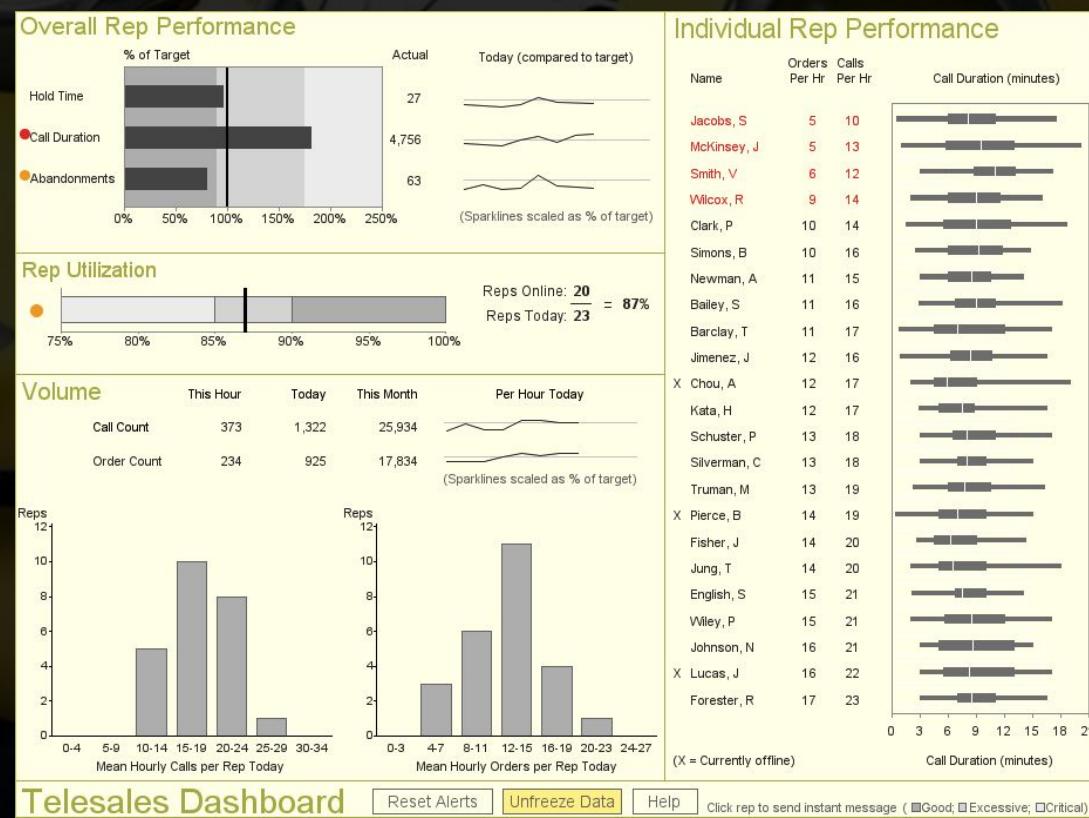


vs.

A quick introduction to: Dashboards

Presentation:

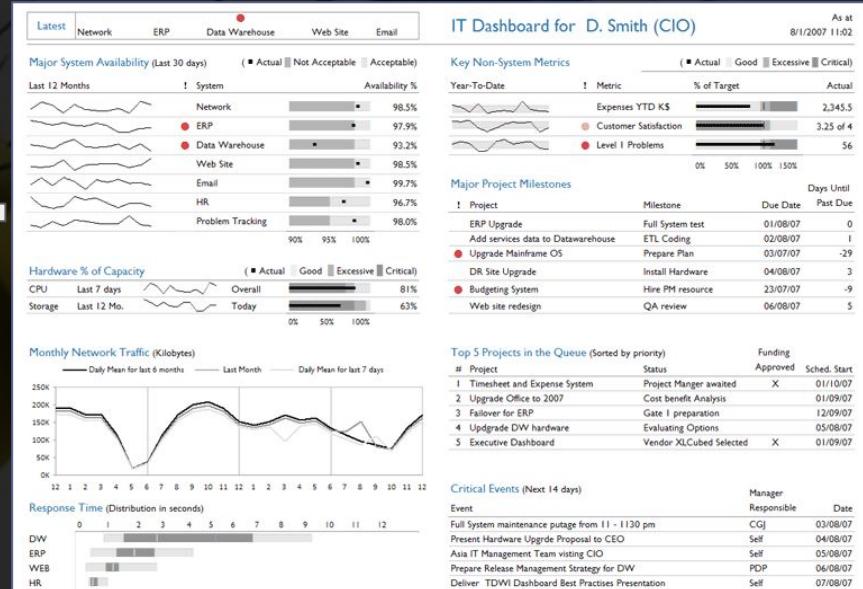
- clearly stated messages
- concise (data to ink ratio)
- direct
- customized to goals
- consistent layout - data changes over time, not layout



A quick introduction to: Dashboards

Presentation:

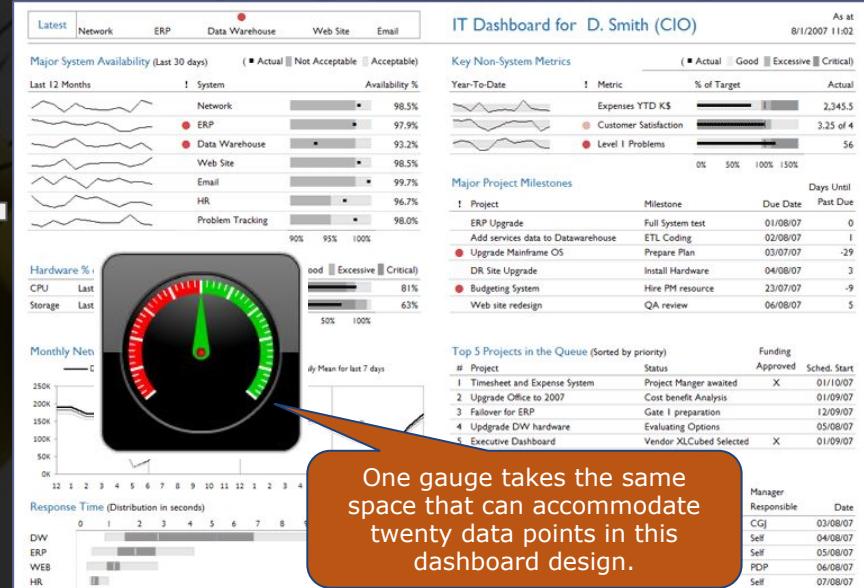
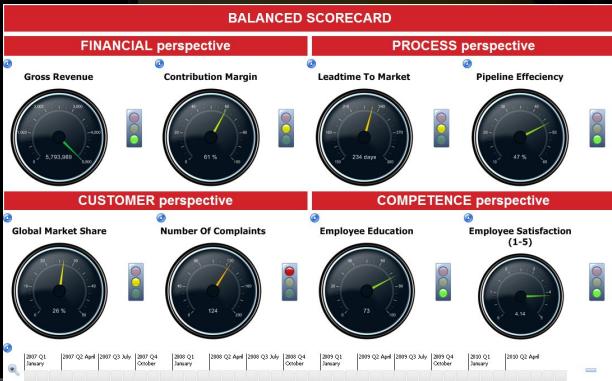
- **Avoid useless bling**



A quick introduction to: Dashboards

Presentation:

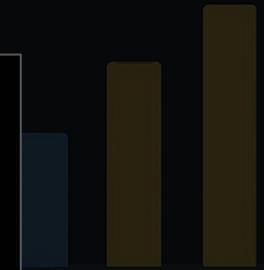
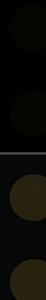
- Avoid useless bling



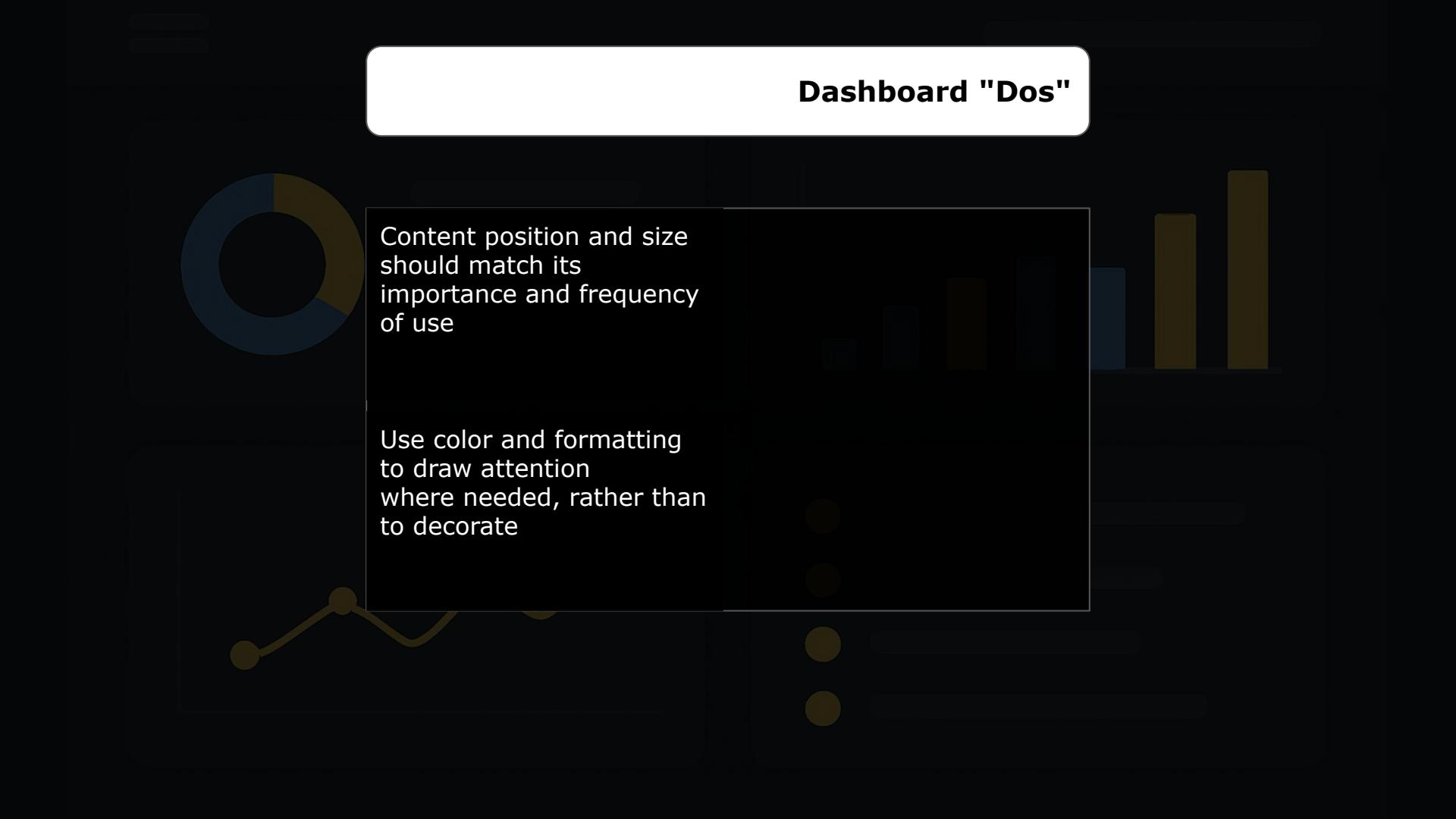
Dashboard "Dos"



Content position and size
should match its
importance and frequency
of use



Dashboard "Dos"



Content position and size should match its importance and frequency of use

Use color and formatting to draw attention where needed, rather than to decorate

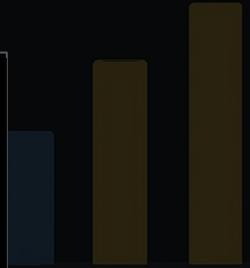
Dashboard "Dos"



Content position and size should match its importance and frequency of use

Use color and formatting to draw attention where needed, rather than to decorate

Visually associate data and content that is related



Dashboard "Dos"

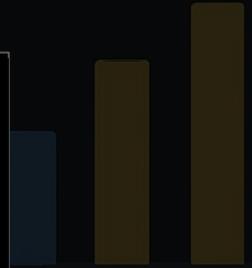


Content position and size should match its importance and frequency of use

Use color and formatting to draw attention where needed, rather than to decorate

Visually associate data and content that is related

Use the needs of the user to drive the layout, rather than forcing layout with an inflexible grid (note: this is a consideration when choosing tools)



Dashboard "Dos"



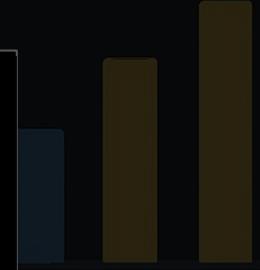
Content position and size should match its importance and frequency of use

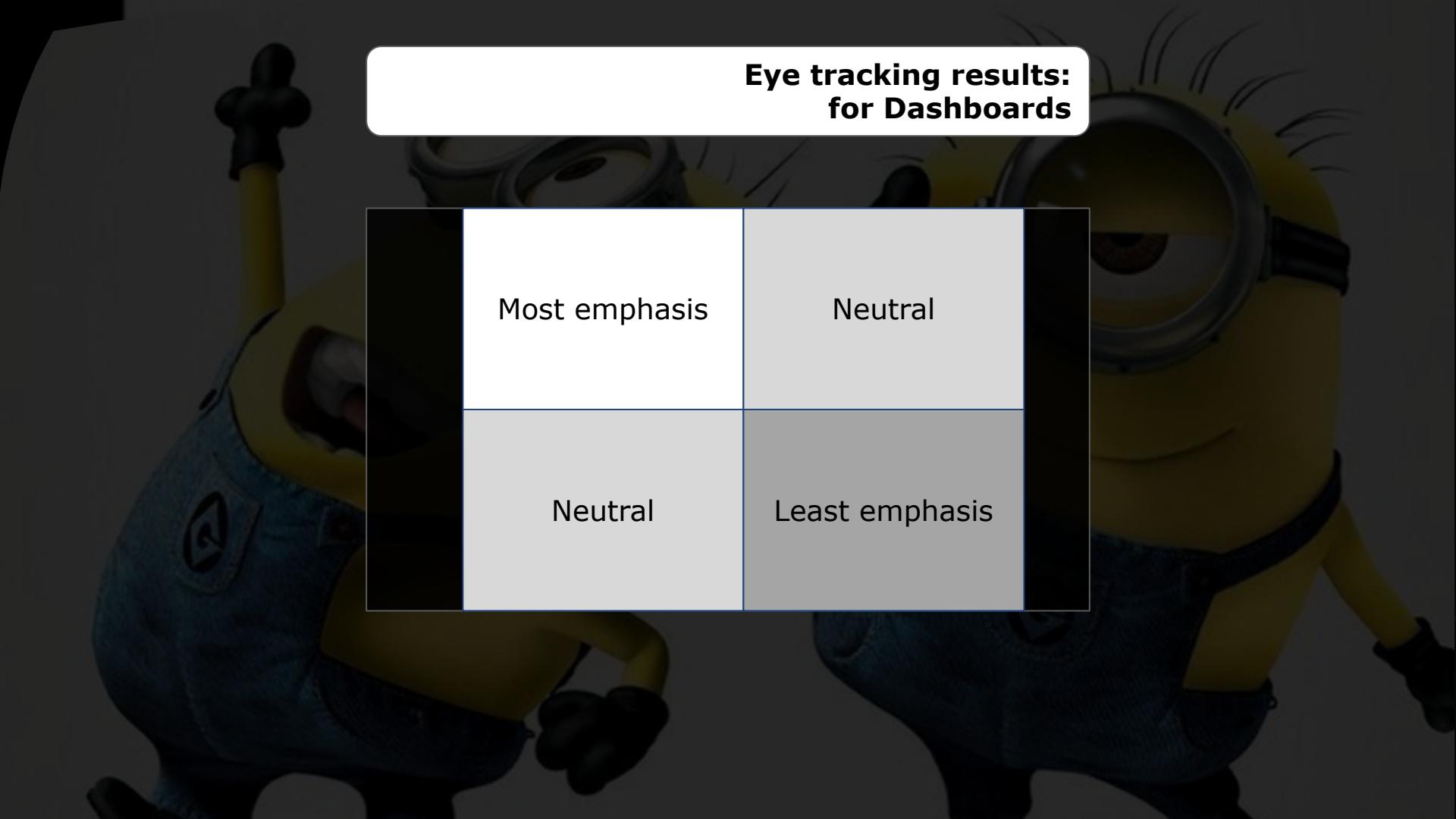
Visually associate data and content that is related

When deciding on placement, consider how the eye will scan the page...

Use color and formatting to draw attention where needed, rather than to decorate

Use the needs of the user to drive the layout, rather than forcing layout with an inflexible grid (note: this is a consideration when choosing tools)



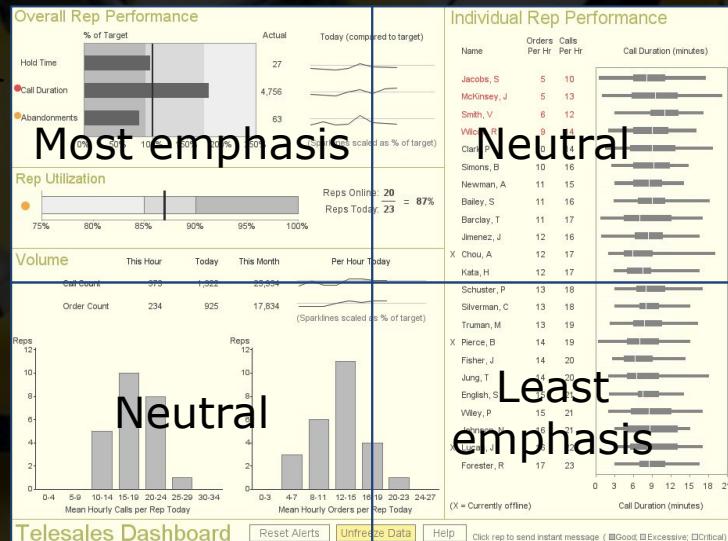


Eye tracking results: for Dashboards

	Most emphasis	Neutral	
	Neutral	Least emphasis	

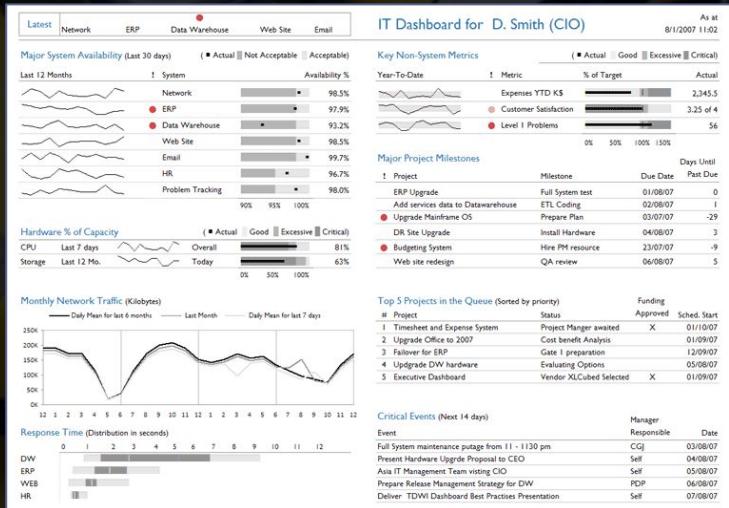
Example: Sales Team Dashboard

The most important information about overall rep performance is in the top left quadrant.



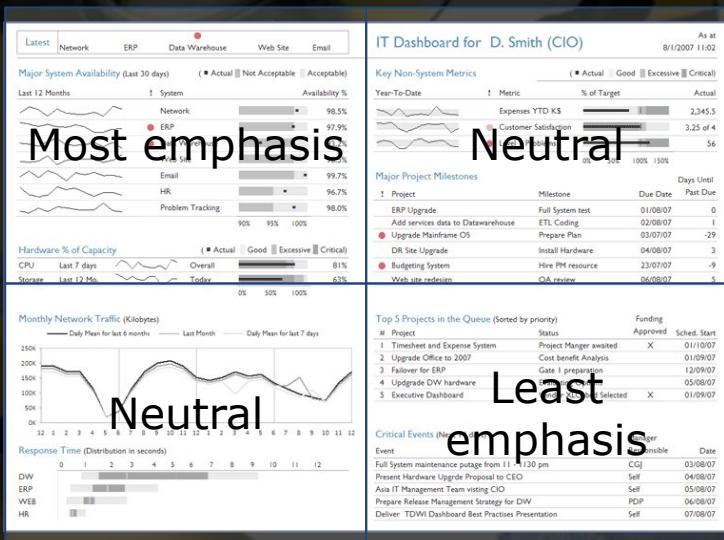
Example: Sales Team Dashboard

The most important information about overall rep performance is in the top left quadrant.



Example: CIO Dashboard

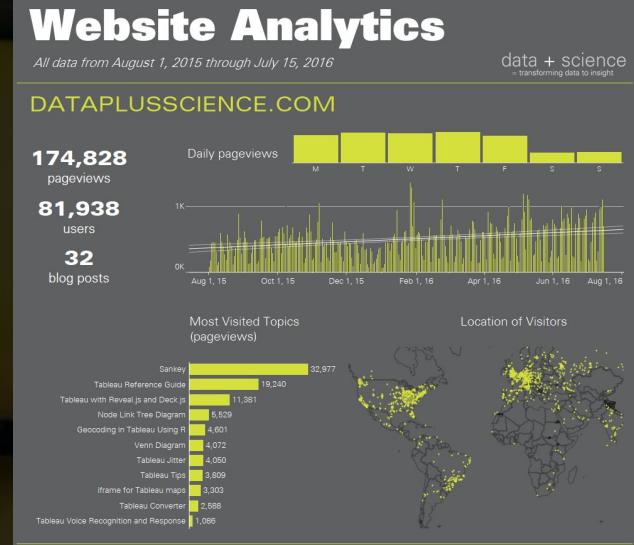
Critical information:
“System Availability”



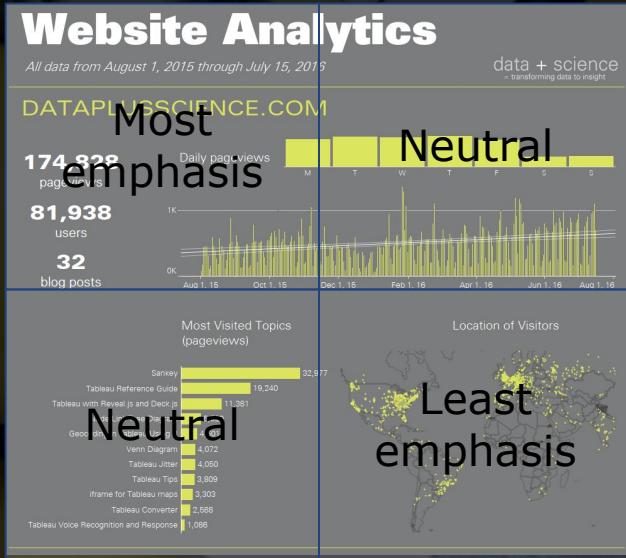
The neutral areas are filled with non-system metrics and overall monthly network traffic.

The least important information are the project status updates.

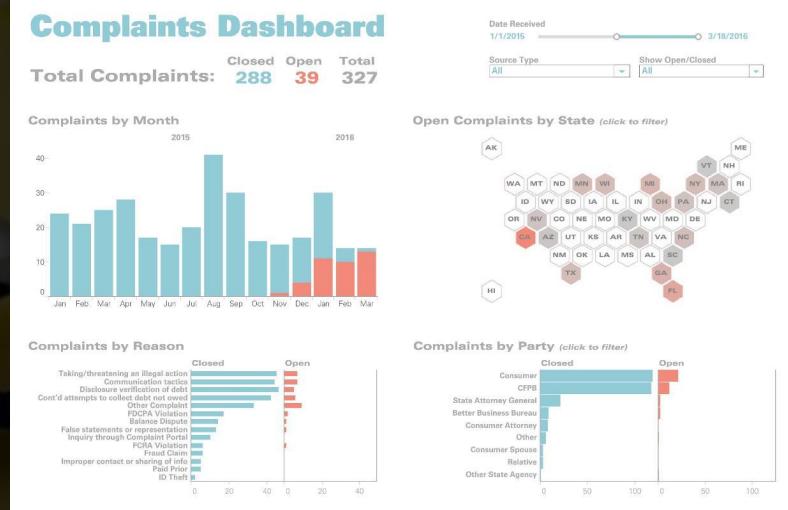
Example: Website Dashboard



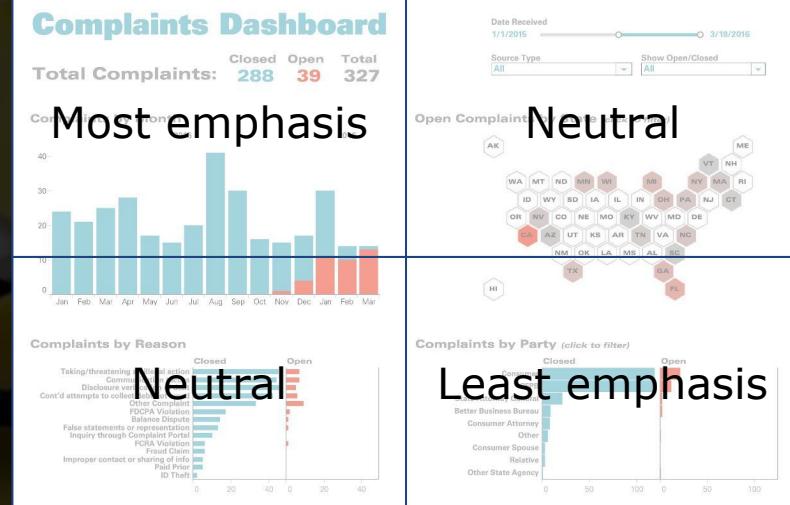
Example: Website Dashboard



Example: Complaints Dashboard



Example: Complaints Dashboard



Dashboards **MUST** show actionable insights!

Rules for Actionable Visualizations

1. The question to answer must be identifiable

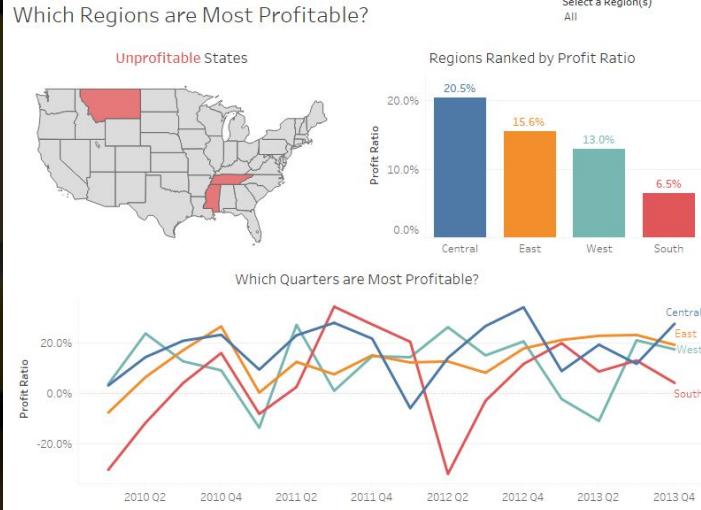
Articulate the question you wish to answer and write it out:

"I want to know who my best customers are."

"I need to be able to identify customers at risk."

"How is our sales team performing against its goals?"

The question to answer must be identifiable



The question to answer must be identifiable



The question to answer must be identifiable



Dashboards MUST show actionable insights!

Rules for Actionable Visualizations

2. The data needed must be available

In some cases, you may need to create it, based on conditions in the data you already have, or by bringing in additional data from another source.



Dashboards **MUST** show actionable insights!

Rules for Actionable Visualizations

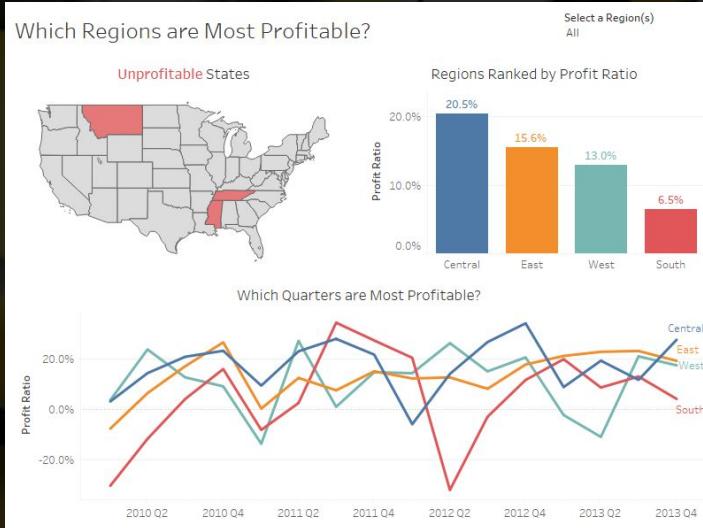
3. The visualization should be tailored to the person who will use the information

Your audience may be "the general public" but in other cases, it's the VP of Finance. You may discover you need more than one to cover all levels of the organisation!

4. The story uncovered in the visualization should be evident

The viewer should not need an advanced degree in statistics to interpret the visualization, or have to make leaps of logic to understand what the data really means.

The story uncovered in the visualization should be evident

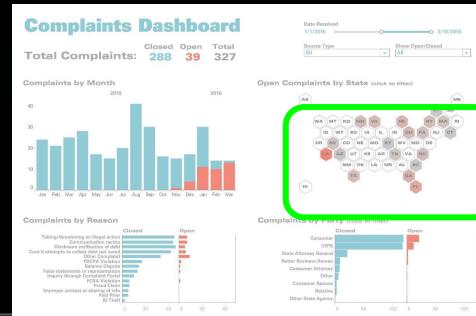


Dashboards MUST show actionable insights!

Rules for Actionable Visualizations

5. The action required should be clear

Framed by the original question, this answers "what do I need to do?" based on the findings.



Interactive provides drill down.





Session 4



Visualizations in practice



Today, Tableau
demystified.



Key Concepts of Tableau



[Official guide online](#)
[Tableau Desktop PDF](#)
[Quick reference](#)

Six main things to know!

- 1) Data sources
- 2) Sheets & Dashboards
- 3) Shelves and cards
- 4) Continuous vs. Discrete
- 5) Measures vs Dimensions
- 6) Show me popup

Data sources contain the raw data and can be **tables / query results from a database**, excel files etc.

Key Concepts of Tableau



[Official guide online](#)
[Tableau Desktop PDF](#)
[Quick reference](#)

Six main things to know!

- 1) Data sources
- 2) [**Sheets & Dashboards**](#)
- 3) Shelves and cards
- 4) Continuous vs. Discrete
- 5) Measures vs Dimensions
- 6) Show me popup

Sheets are where you create **single graphs**.

Dashboards collect and display **previously created sheets**.

Key Concepts of Tableau



[Official guide online](#)
[Tableau Desktop PDF](#)
[Quick reference](#)

Six main things to know!

- 1) Data sources
- 2) Sheets & Dashboards
- 3) [**Shelves and cards**](#)
- 4) Continuous vs. Discrete
- 5) Measures vs Dimensions
- 6) Show me popup

Shelves and cards are **where you drag attributes**.

Attributes = measures or dimensions

Key Concepts of Tableau



[Official guide online](#)
[Tableau Desktop PDF](#)
[Quick reference](#)

Six main things to know!

- 1) Data sources
- 2) Sheets & Dashboards
- 3) [**Shelves and cards**](#)
- 4) Continuous vs. Discrete
- 5) Measures vs Dimensions
- 6) Show me popup

Shelves and cards are **where you drag attributes**.

Attributes = measures or dimensions

By placing fields on shelves or cards, you can **create** the rows and columns of a **data view**, **exclude data** from the view, create pages, and **control mark properties**

Key Concepts of Tableau



[Official guide online](#)
[Tableau Desktop PDF](#)
[Quick reference](#)

Six main things to know!

- 1) Data sources
- 2) Sheets & Dashboards
- 3) [**Shelves and cards**](#)
- 4) Continuous vs. Discrete
- 5) Measures vs Dimensions
- 6) Show me popup

Shelves and cards are **where you drag attributes**.

Attributes = measures or dimensions

By placing fields on shelves or cards, you can **create** the rows and columns of a **data view**, **exclude data** from the view, create pages, and **control mark properties**

The **different placement and combinations** of measures and dimensions, and whether the values are continuous or discrete, **dictate which graph Tableau shows**.

Key Concepts of Tableau



[Official guide online](#)
[Tableau Desktop PDF](#)
[Quick reference](#)

Six main things to know!

- 1) Data sources
- 2) Sheets & Dashboards
- 3) Shelves and cards
- 4) [**Continuous vs. Discrete**](#)
- 5) Measures vs Dimensions
- 6) Show me popup

Each data source (table) contains a set of attributes (columns).

Each attribute can be either **continuous** or a **discrete**.

Continuous attributes those with values along a number line (decimal numbers).

Discrete attributes are those with values that are "individually separate and distinct."

Key Concepts of Tableau



[Official guide online](#)
[Tableau Desktop PDF](#)
[Quick reference](#)

Six main things to know!

- 1) Data sources
- 2) Sheets & Dashboards
- 3) Shelves and cards
- 4) [**Continuous vs. Discrete**](#)
- 5) Measures vs Dimensions
- 6) Show me popup

Each data source (table) contains a set of attributes (columns).

Each attribute can be either **continuous** or a **discrete**.

Continuous attributes those with values along a number line (decimal numbers).

Discrete attributes are those with values that are "individually separate and distinct."

When placed on the **columns** or **rows** shelves:

Continuous fields produce axes.

Discrete fields create headers.

Key Concepts of Tableau



[Official guide online](#)
[Tableau Desktop PDF](#)
[Quick reference](#)

Six main things to know!

- 1) Data sources
- 2) Sheets & Dashboards
- 3) Shelves and cards
- 4) Continuous vs. Discrete
- 5) [**Measures vs Dimensions**](#)
- 6) Show me popup

Each data source (table) contains a set of attributes (columns).

Each attribute is either a **measure** or a **dimension**.

Measures are numbers, they quantify the extent of something.

Dimensions are usually names, categorizing the "measure" in different kinds.

What is a measure and what is a dimension depends on the problem.

Am I measuring someone's weight, or do I care about a disease and am grouping people by weight (low/med/high) for my study?

Key Concepts of Tableau



[Official guide online](#)
[Tableau Desktop PDF](#)
[Quick reference](#)

Six main things to know!

- 1) Data sources
- 2) Sheets & Dashboards
- 3) Shelves and cards
- 4) Continuous vs. Discrete
- 5) [**Measures vs Dimensions**](#)
- 6) Show me popup

Each data source (table) contains a set of attributes (columns).

Each attribute is either a **measure** or a **dimension**.

Attributes can be **converted between** being **measures** and **dimensions**.

Initially Tableau guesses.

Measures are numbers, they quantify the extent of something.

Dimensions are usually names, categorizing the "measure" in different kinds.

Key Concepts of Tableau



[Official guide online](#)
[Tableau Desktop PDF](#)
[Quick reference](#)

Six main things to know!

- 1) Data sources
- 2) Sheets & Dashboards
- 3) Shelves and cards
- 4) Continuous vs. Discrete
- 5) [**Measures vs Dimensions**](#)
- 6) Show me popup

Each data source (table) contains a set of attributes (columns).

Each attribute is either a **measure** or a **dimension**.

Attributes can be **converted between** being **measures** and **dimensions**.

Initially Tableau guesses.

Measures are numbers, they quantify the extent of something.

Dimensions are usually names, categorizing the "measure" in different kinds.

The **different placement and combinations** of measures and dimensions, and whether the values are continuous or discrete, **dictates which graph Tableau shows**.

Key Concepts of Tableau



[Official guide online](#)
[Tableau Desktop PDF](#)
[Quick reference](#)

Six main things to know!

- 1) Data sources
- 2) Sheets & Dashboards
- 3) Shelves and cards
- 4) Continuous vs. Discrete
- 5) Measures vs Dimensions
- 6) Show me popup

The show me popup helps you
**see what graphs Tableau
can make.**

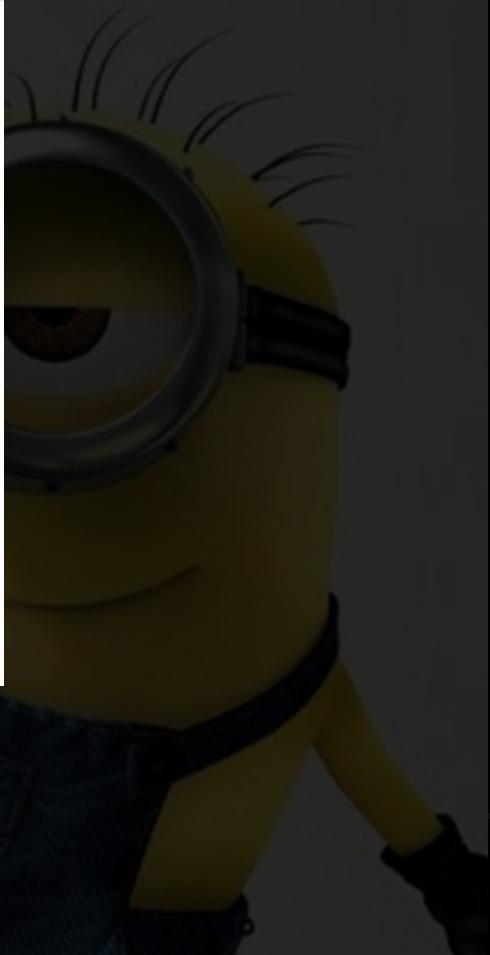
Shows you what you need
(dimensions, measures, etc) in
order for Tableau to show the
graph.

Main focus of today...



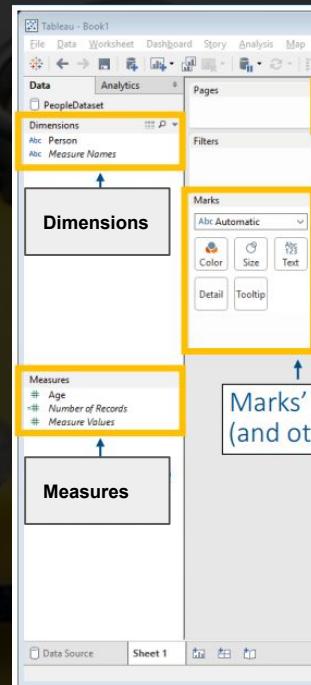
The main interface

The screenshot shows the Tableau desktop application interface. On the left, the Data pane displays a dataset named "PeopleDataset" with sections for Dimensions (containing "Abc. Person" and "Abc. Measure Names") and Measures (containing "Age", "Number of Records", and "Measure Values"). The Marks pane contains options for "Automatic" marks and visual variables like Color, Size, and Text, along with "Detail" and "Tooltip" buttons. The main workspace is divided into "Columns" and "Rows" areas, each with a "Drop field here" placeholder. A callout box labeled "Attributes we want to visualize are dropped here" points to the Columns area. Another callout box labeled "Visualizations will appear here" points to the Rows area. To the right is the "Show Me" pane, which lists various visualization types such as bar charts, line graphs, and maps, with a note: "Select or drag data. Use the Shift or Ctrl key to select multiple fields".



Recall **Dimensions** and **Measures**.

Example: People Dataset



Know your data!

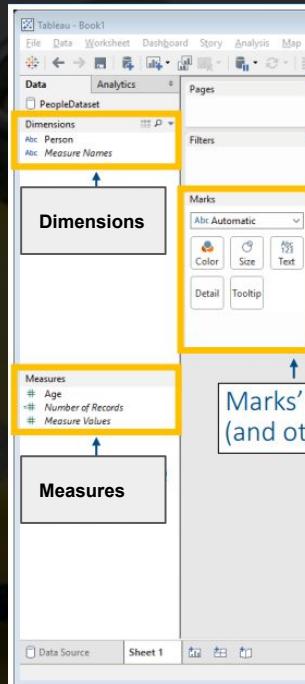
What was a measurement (**measure**)?

What was a label describing what was measured (**dimension**)?

Don't know? Why are you visualizing something you don't understand....

Recall **Dimensions** and **Measures**.

Example: People Dataset



Know your data!

What was a measurement (**measure**)?

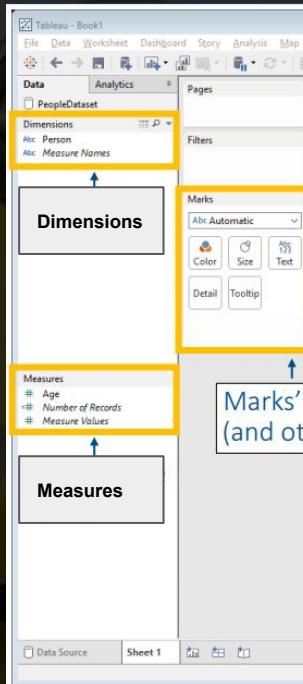
What was a label describing what was measured (**dimension**)?

Don't know? Why are you visualizing something you don't understand....

Why care? Tableau will enable options based on its understanding of the data.

Recall **Dimensions** and **Measures**.

Example: People Dataset



Know your data!

What was a measurement (**measure**)?

What was a label describing what was measured (**dimension**)?

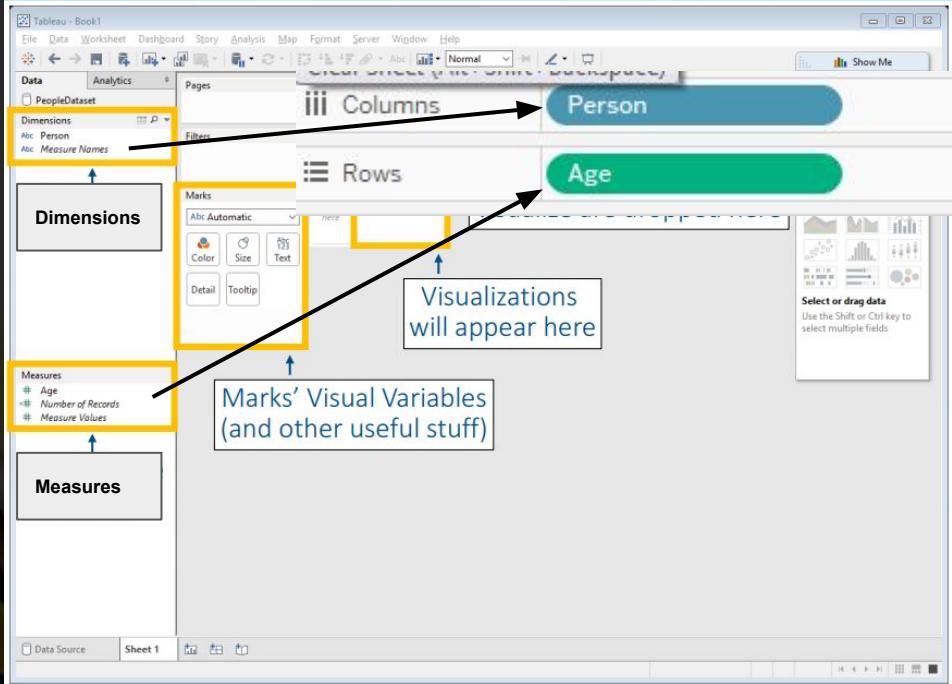
Don't know? Why are you visualizing something you don't understand....

Why care? Tableau will enable options based on its understanding of the data.

Want to predict the future of a label regarding a measurement? **Huh?!?**

Predict a measured value? **Sure!**

Creating visualizations



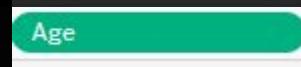
A note on "Pills"

Pill colour denote if an attribute is **discrete** or **continuous**.



Blue pill: Discrete attributes.

Green pill: Continuous attributes.



Creating visualizations

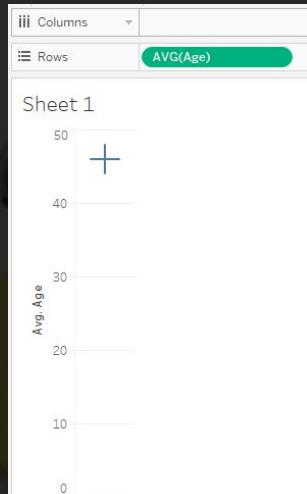
Abc	#
PeopleDataset.c...	PeopleDataset.c...
Person	Age
Emily	45
John	31
Charles	38
Claire	51
Samantha	65



Placing discrete attributes creates "headers".
Value existence is then plotted.

Creating visualizations

Abc	#
PeopleDataset.c...	PeopleDataset.c...
Person	Age
Emily	45
John	31
Charles	38
Claire	51
Samantha	65



Placing continuous attributes creates
"plotted aggregate values" **on an axis**.

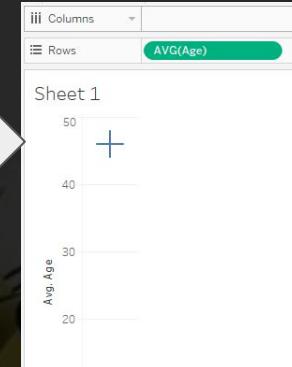


Dragging and dropping continuous attributes by default creates **single point aggregate** measures.

However: Measures can be returned to a set of values as per the original data.

Abc	#
Person	Age
Emily	45
John	31
Charles	38
Claire	51
Samantha	65

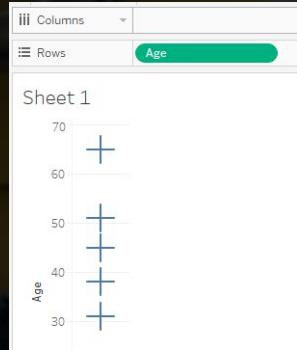
A screenshot of the Tableau desktop interface. A context menu is open over a measure named 'Avg(Age)'. The menu path 'Measures' > 'Avg(Age)' is highlighted. Other options in the 'Measures' submenu include 'Number of Records' and 'Measure Values'. The main menu bar at the top includes 'File', 'Data', 'Worksheet', 'Dashboard', 'Story', 'Analysis', 'Map', 'Format', 'Server', 'Window', and 'Help'. The bottom status bar shows '1 mark 1 row by 1 column SUM of AVG(Age): 46.00'.



Placing continuous attributes creates "plotted **aggregate values**" **on an axis**.

Or one can select to not aggregate...

A screenshot of the Tableau desktop interface. A context menu is open over a measure named 'Age'. The menu path 'Measures' > 'Age' is highlighted. Other options in the 'Measures' submenu include 'Number of Records' and 'Measure Values'. The main menu bar at the top includes 'File', 'Data', 'Worksheet', 'Dashboard', 'Story', 'Analysis', 'Map', 'Format', 'Server', 'Window', and 'Help'. The bottom status bar shows '1 mark 1 row by 1 column SUM of AVG(Age): 46.00'.



Abc	#
Person	Age
Emily	45
John	31
Charles	38
Claire	51
Samantha	65

Adding Person to columns
gave us headers.

iii Columns	Person
Rows	
Sheet 1	
Charles	Person
+	Emily
John	+
Samantha	+

By placing attributes in **rows** and **columns** we can build basic graphs.

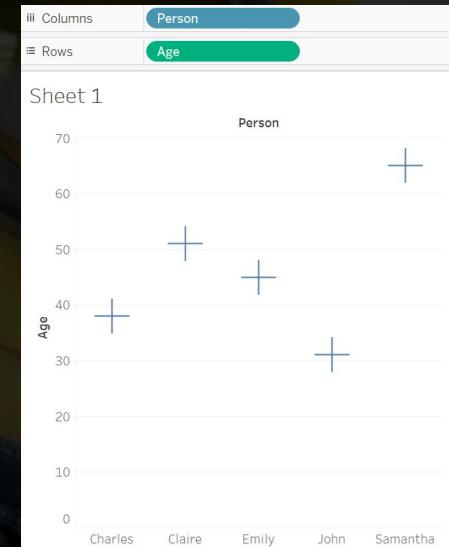
Abc	#
PeopleDataset.c...	PeopleDataset.c...
Person	Age
Emily	45
John	31
Charles	38
Claire	51
Samantha	65

Adding Person to columns gave us headers.

iii Columns	Person
iii Rows	
Sheet 1	
Charles	Person
+	Emily
Charles	John
+	Samantha
Charles	
+	

By placing attributes in **rows** and **columns** we can build basic graphs.

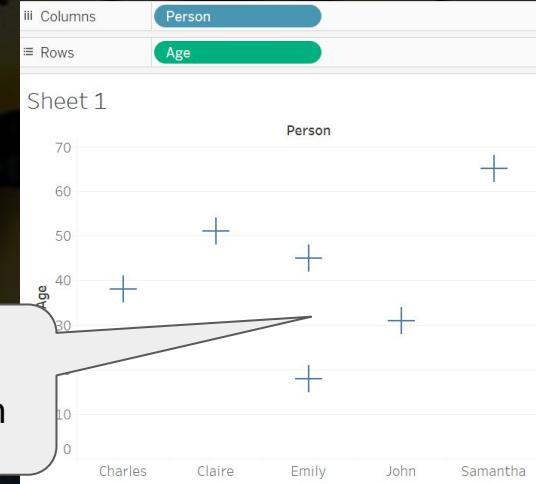
By combining 1 dimension and 1 measure we can build basic graphs.



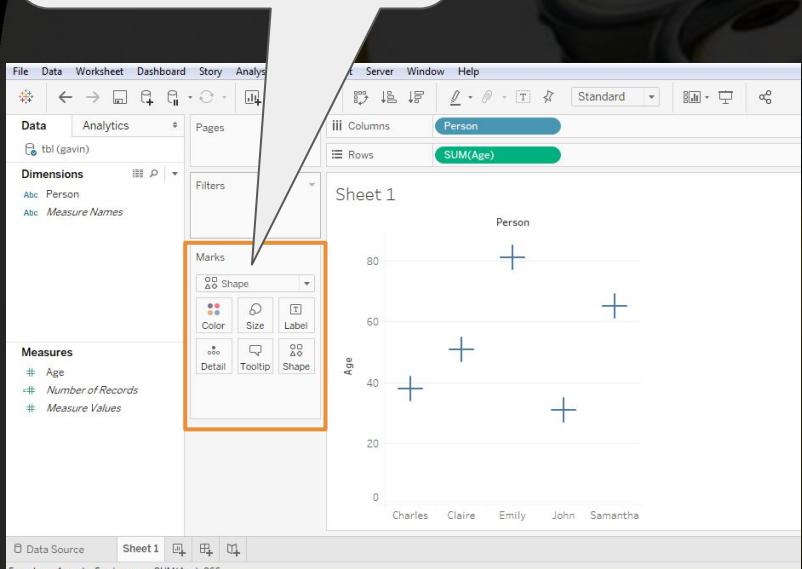
Let's update our data!

By combining 1 dimension and 1 measure we can build basic graphs.

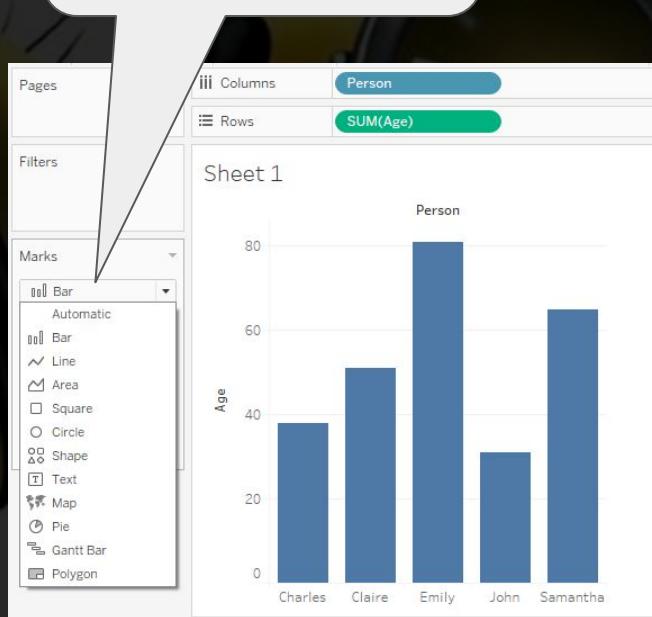
Abc	#
Person	Age
PeopleDataset_fc...	PeopleDataset_fc...
Emily	45
John	31
Charles	38
Claire	51
Samantha	65
Emily	18
Emily	18



The "mark" drawn for the graph can be selected

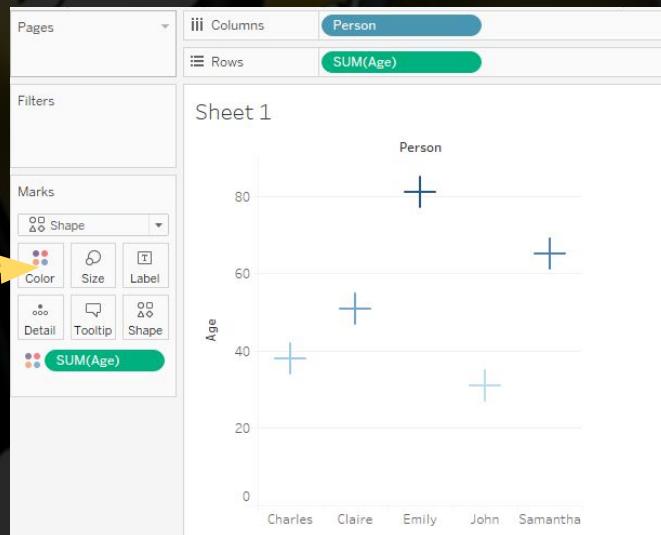
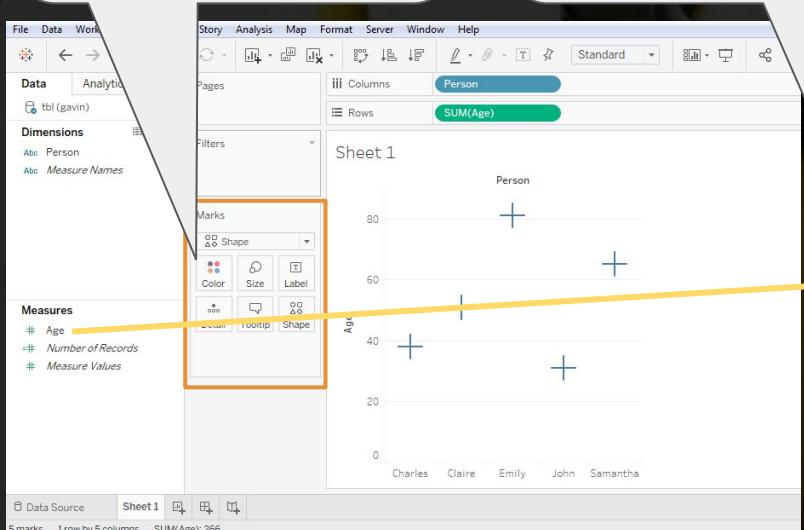


Here "bar" has now been selected.



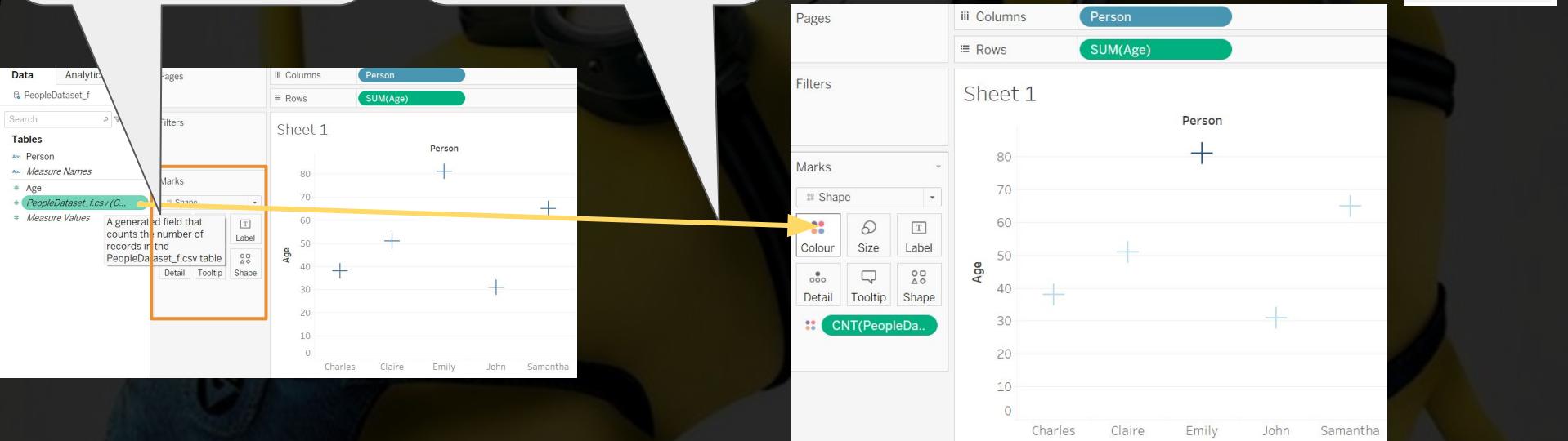
Or colour, size, label, detail, tooltip or shape changed **based on the value by a pill value** (drag & drop)

E.g. Dragging "Age" to "Colour"



Or colour, size, label, detail, tooltip or shape changed **based on the value** by a **pill value** (drag & drop)

E.g. Dragging number of records (or "Count") to "Colour"
(remember Emily appears x3)



Adding more **attributes** to the **Rows** and **Columns** shelves adds more rows, columns, and panes to the table.



Dimensions (labels of the measurements) combine ("nest") to show the measure values for all possible label combinations.

Adding more **attributes** to the **Rows** and **Columns** shelves adds more rows, columns, and panes to the table.



Dimensions (labels of the measurements) combine ("nest") to show the measure values for all possible label combinations.

Measures have created axis and plotted values.

Adding more **attributes** to the *Rows* and *Columns* shelves adds more rows, columns, and panes to the table.



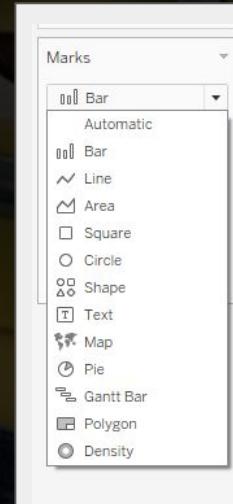
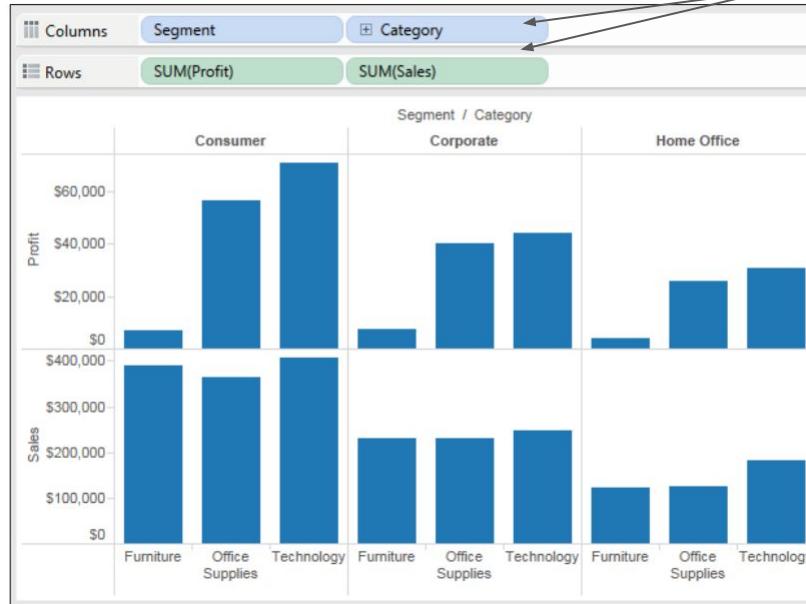
Dimensions (labels of the measurements) combine ("nest") to show the measure values for all possible label combinations.

Measures have created axis and plotted values.

Measures GO AFTER dimensions on a shelf.

Inner attributes determine an automatic mark type. Here: Bar.

Adding more **attributes** to the *Rows* and *Columns* shelves adds more rows, columns, and panes to the table.



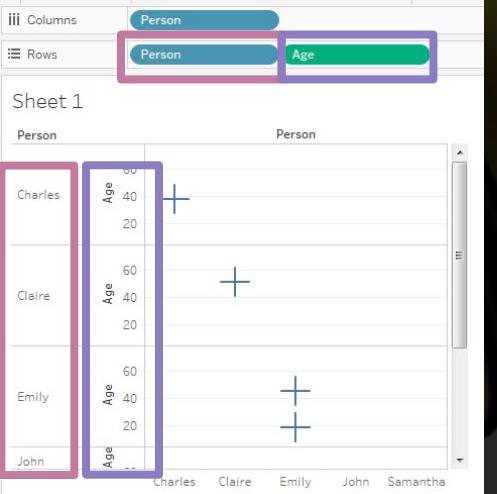
Automatic is not always what you want.

Change it as you see fit in the Marks card!

Meaningless graphs to show pill combinations

outer

inner

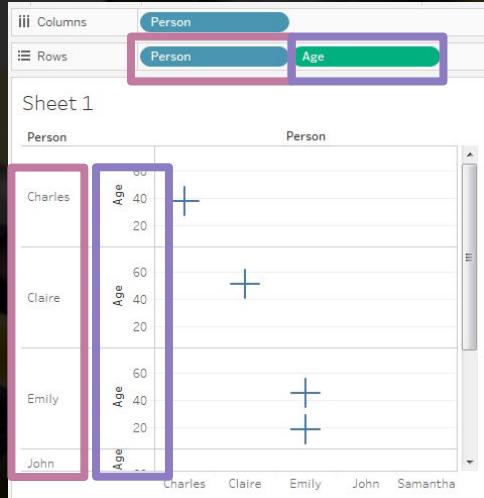


Person	Age
Emily	45
John	31
Charles	38
Claire	51
Samantha	65
Emily	18
Emily	18

Meaningless graphs to show pill combinations

outer

inner



A small problem for context.

Data: Every week, the Minions are busy across their different activities...

..., i.e. managing banana supplies, plotting evil plans, working in the lab, causing mischief, training, and cleaning up the mess afterward.

Now being a data guy, what does Gru do?



A small problem for context.

Data: Every week, the Minions are busy across their different activities...

..., i.e. managing banana supplies, plotting evil plans, working in the lab, causing mischief, training, and cleaning up the mess afterward.

Now being a data guy, what does Gru do?

Well that's simple, he counts them.
Gru keeps track of how their energy is distributed across these activities over 10 weeks.

A small problem for context.

Data: Every week, the Minions are busy across their different activities...

..., i.e. managing banana supplies, plotting evil plans, working in the lab, causing mischief, training, and cleaning up the mess afterward.

Now being a data guy, what does Gru do?

Well that's simple, he counts them.

Gru keeps track of how their energy is distributed across these activities over 10 weeks.

The Data is very simple.

Week	Banana Supply	Evil Plans	Lab Work	Mischief	Training	Cleanup
Week 1	101	67	56	18	49	35
Week 2	90	60	56	29	38	23
Week 3	86	40	39	37	31	21
Week 4	75	51	57	40	42	32
Week 5	69	65	57	51	39	18
Week 6	65	61	45	57	44	39
Week 7	50	68	44	62	22	28
Week 8	46	51	59	76	24	32
Week 9	39	64	59	82	38	23
Week 10	29	56	44	86	26	35

Tableau...
Ok. I think I get it.

[How about a demo...](#)

1. Bar chart of Minions per week.
2. Stacked bar chart of Minions per week showing proportion of counts per hour.
3. Stacked bar chart of Minions per week showing proportion of counts per time slot.
4. Plot of marks of Minions per hour per activity.
5. Box plot of Minions count per activity.
6. Spotting trends per Minions activity.

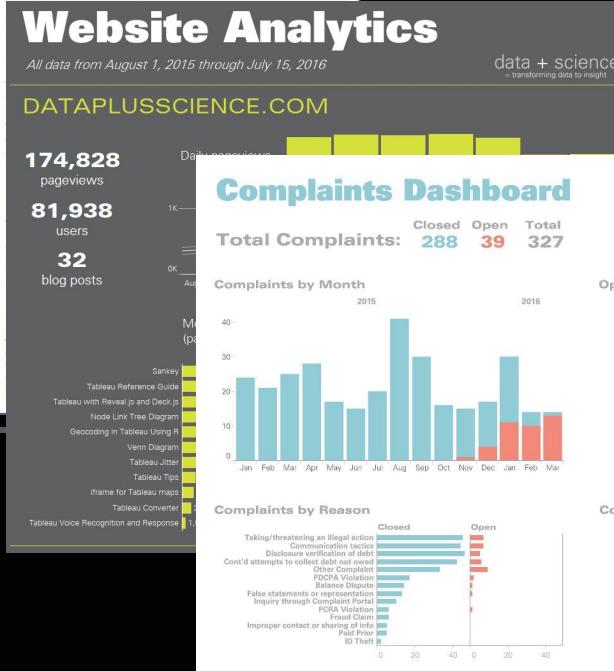
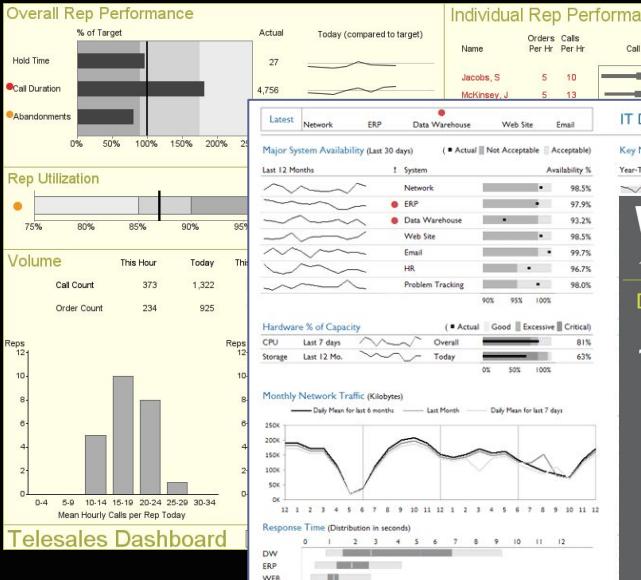
Extra. Predict the counts of Minions causing mischief for the following weeks.

Dashboards *are* part of real world data analytics

(at least until you can get someone else to do it)

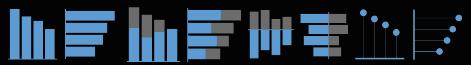
Presentation:

- clearly stated messages
- concise (data to ink ratio)
- direct
- customized to goals
- consistent layout - data changes over time, not layout





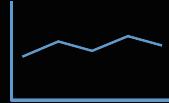
Comparing data across categories



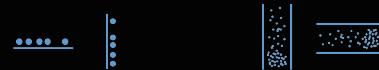
Showing / understanding the distribution of your data



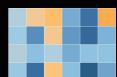
Viewing trends in data over time



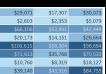
Investigating the relationship between different variables



Showing the relationship between two factors.



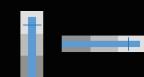
Showing geocoded (located) data



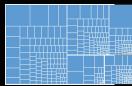
Displaying things in use over time.



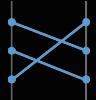
Evaluating performance of a metric against a goal



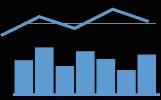
Showing hierarchical data as a proportion of a whole



Showing a comparison of rank (typically between two time periods)



Show general trends / overall information, add extra context



Showing the gradual transition (+/-) in the quantitative value.



Data at Scale

Dr. Evgeniya Lukinova

Big Data Tech for

Analytics,
Geo,
Data science,
Machine Learning,
Stuff....



Big Data Technology covers both storage and processing

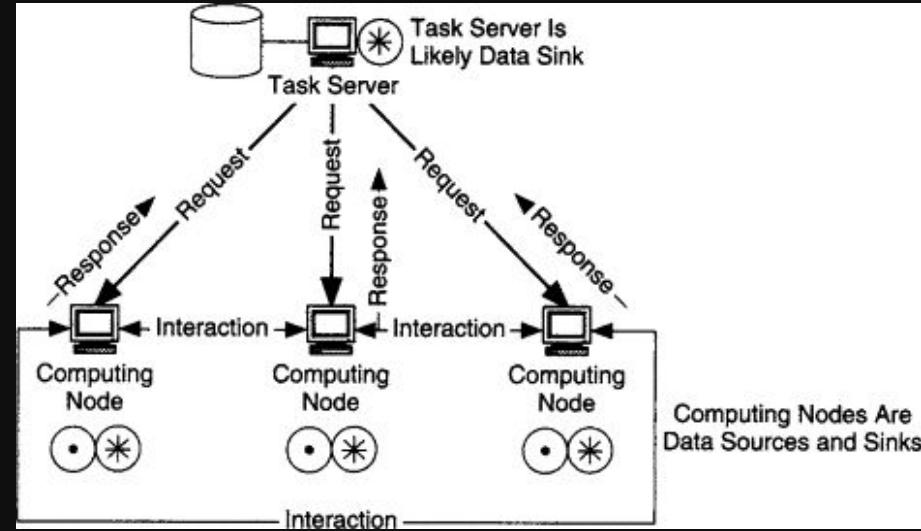
Lots of workers? Need a manager(s)
Need a communication protocol.

Need to know how to subdivide tasks
Need to send code to workers
Need to transfer data between workers
→ unless it is already correctly subdivided and located

Need to collate results of the workers

What if a worker is lazy? Quits?
Makes an error?

Faster to "just do it yourself"



Big Data Technology covers both storage and processing

Lots of workers? Need a manager(s)
Need a communication protocol.

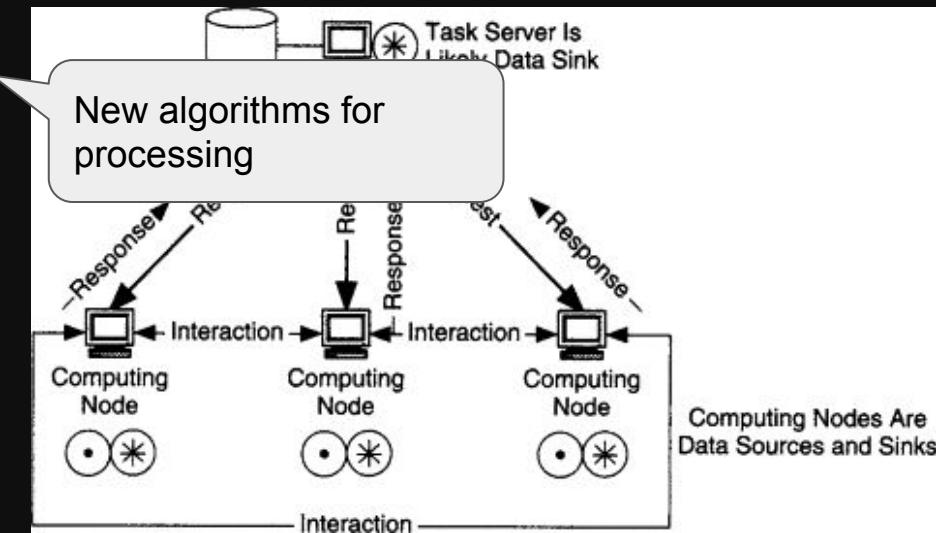
New programming paradigms to write algorithms

Need to know how to subdivide tasks
Need to send code to workers
Need to transfer data between workers
→ unless it is already correctly subdivided and located

Extra logic and complexity within the algorithms

Need to collate results of the workers
What if a worker is lazy? Quits?
Makes an error?
Faster to "just do it yourself"

It might run slower, or not that much faster (cost outweigh the benefit)



Big Data Technology covers both storage and processing

Either because we use a database that "guesses" or we write code to pre-distribute (replicated?) data or we have an algorithm that does multiple passes of data

Easy option:
→ Each table is stored on a different compute node
→ Fast single table reads
→ Slow joins....

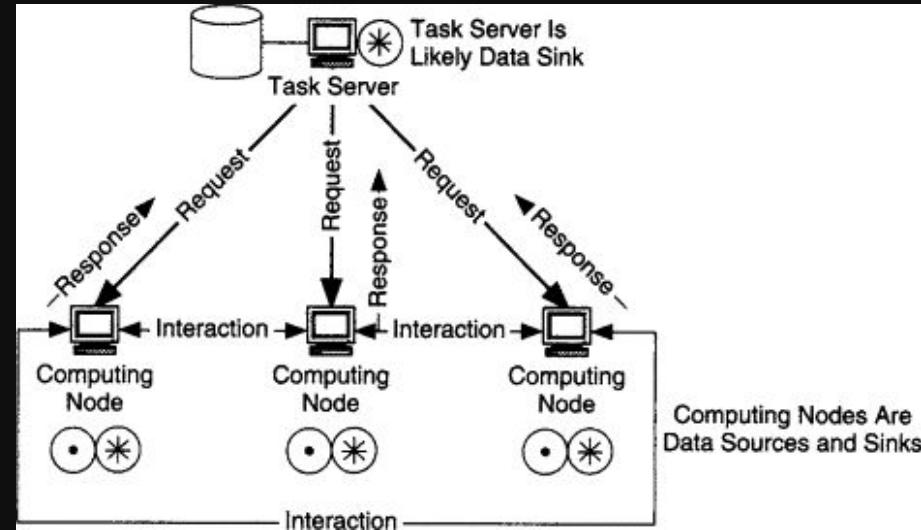
Lots of workers? Need a manager(s)
Need a communication protocol.

Need to know how to subdivide tasks
Need to send code to workers
Need to transfer data between workers
→ unless it is already correctly subdivided and located

Need to collate results of the workers

What if a worker is lazy? Quits?
Makes an error?

Faster to "just do it yourself"?



Big Data Technology covers both storage and processing

Either because we use a database that "guesses" or we write code to pre-distribute (replicated?) data or we have an algorithm that does multiple passes of data

Easy option:
→ Each table is stored on a different compute node
→ Fast single table reads
→ Slow joins....

Lots of workers? Need a manager(s)
Need a communication protocol.

Need to know how to subdivide tasks
Need to send code to workers
Need to transfer data between workers
→ unless it is already correctly subdivided and located

Need to collate results of the workers

What if a worker is lazy? Quits?
Makes an error?

Faster to "just do it yourself"?

Holy Grail of Big data tech - hide all this complexity.

Hard to use the original procedural programming - new paradigm required.
→ parfor, MapReduce, Spark

Each paradigm enables certain parallel tasks to be written easily. But not all.
Work is ongoing and highly efficient solutions remain hard.

SQL... since this doesn't contain instructions, one just lists the task... huge potential.

Also since this includes the specification of the data storage can potentially leverage benefits of jointly optimising storage and processing!

Big Data Technology covers both storage and processing

Before we talk about these more user friendly big data technology solutions (spoiler, it's SQL) let's summarize what "big data tech" buys us when we distribute storage and/or processing.

A summary of "big data" technologies use cases

Specifically, we are talking about enabling distributed storage and processing.

Centralized data and processing

- Limited central processing
- Limited data storage

Traditional systems (e.g. RDBMs)

- Simple availability
(all in the same location, either all up/down)
- Centrally controlled algorithms
(easy concurrency, ACID, etc)
- No time costs for moving data around
- No data duplication



A summary of "big data" technologies use cases

Specifically, we are talking about enabling distributed storage and processing.

Centralized data and processing

- Limited central processing
- Limited data storage

Traditional systems (e.g. RDBMs)

- Simple availability
(all in the same location, either all up/down)
- Centrally controlled algorithms
(easy concurrency, ACID, etc)
- No time costs for moving data around
- No data duplication

Distributed data, centralized processing

- Limited processing
- Unlimited storage

Analytics when processing data locally from a data lake/relational database.

- Complexity in availability
(could have partial data access due to outages, need logic to deal with this)
- Centrally controlled algorithms
- Cost (complexity and transfer time) to bring the data to you

"Big Data" technology use cases (for analytics)

Specifically, we are talking about enabling distributed storage and processing.

Centralized data and processing

- Limited central processing
- Limited data storage

Traditional systems (e.g. RDBMs)

- Simple availability
(all in the same location, either all up/down)
- Centrally controlled algorithms
(easy concurrency, ACID, etc)
- No time costs for moving data around
- No data duplication

Distributed data, centralized processing

- Limited processing.
- Unlimited storage

Analytics when processing data locally from a data lake/relational database.

- Complexity in availability
(could have partial data access due to outages, need logic to deal with this)
- Centrally controlled algorithms
- Cost (complexity and transfer time) to bring the data to you

Centralized data, distributed processing

- Unlimited processing capability
- Limited storage capability

Simple, ad-hoc distributed computing tasks.

- Complexity in availability (can easily avoid nodes not available at start, but what about failure partially through processing?)
- Need a new programming paradigm to split up computation
- Cost to send the data to the processing. **Data is still distributed, just in memory. Still need distributed data structures.**

"Big Data" technology use cases (for analytics)

Specifically, we are talking about enabling distributed storage and processing.

Centralized data and processing

- Limited processing
- Limited data storage

Traditional systems (e.g. RDBMs)

- Simple availability (all in the same location, either all up/down)
- Centrally controlled algorithms (easy concurrency, ACID, etc)
- No time costs for moving data around
- No data duplication

Distributed data, centralized processing

- Limited processing.
- Unlimited storage

Analytics when processing data locally from a data lake/relational database.

- Complexity in availability (could have partial data access due to outages, need logic to deal with this)
- Centrally controlled algorithms
- Cost (complexity and transfer time) to bring the data to you

Centralized data, distributed processing

- Unlimited processing capability
- Limited storage capability

Simple, ad-hoc distributed computing tasks.

- Complexity in availability (can easily avoid nodes not available at start, but what about failure partially through processing?)
- **Need a new programming paradigm to split up computation.**
- Cost to send the data to the processing. **Data is still distributed, just in memory. Still need distributed data structures.**

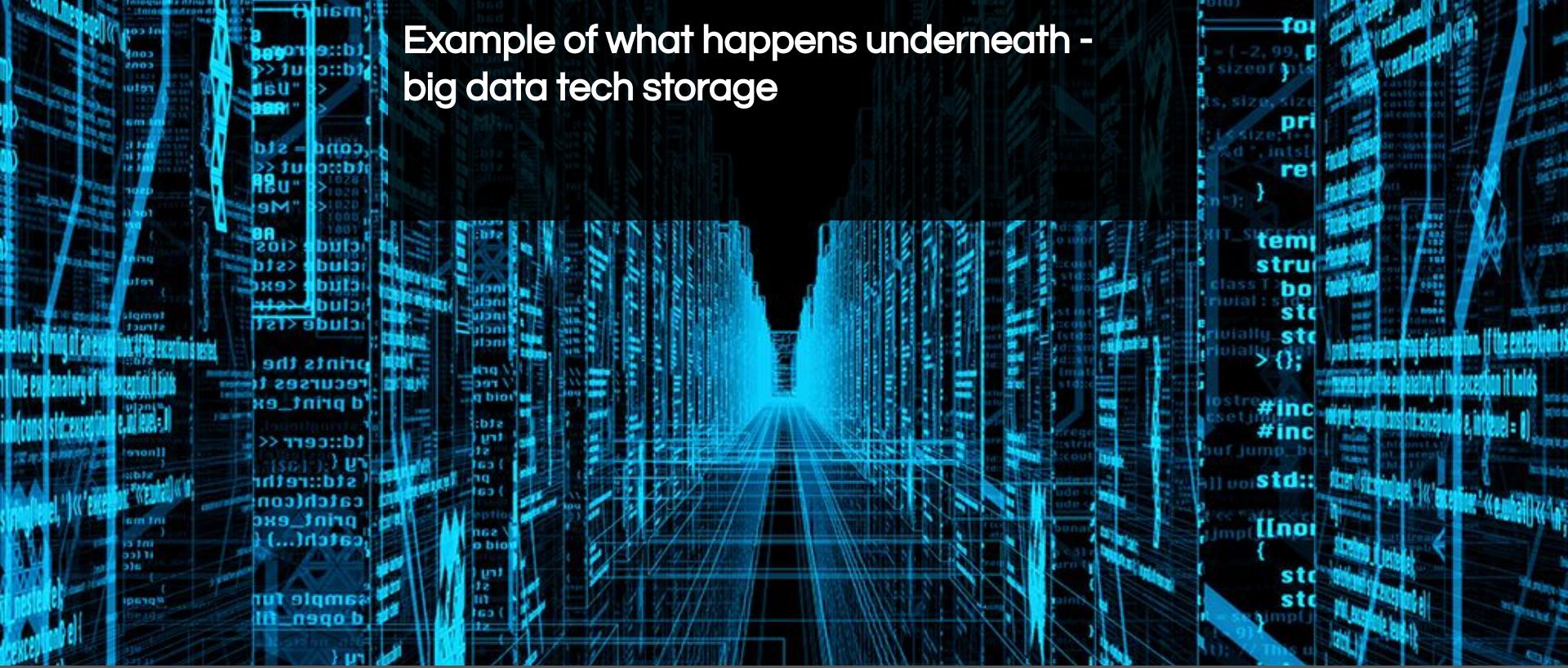
Distributed data, distributed processing

- Reduced data transfer costs
- Ability to scale to many compute nodes and arbitrary storage.

Premeditated, repeated "big data" processing.

- Both sets of complexity in availability
- **Need a new programming paradigm to split up computation**
- **Need ways of manually/auto splitting data (optimally)**
- Costs for moving data around when processing if required and back to you (may be a negligible cost).

Example of what happens underneath - big data tech storage



Data at Scale

Dr. Evgeniya Lukinova

An example of a Big Data Storage solution and why it is hard.... (so you know what you're getting into if you try it....)

NoSQL data stores / processing paradigms

Simple strategies for distribution make some access tasks hard.

- **Replication makes updates hard**
- **Partitioning data can lead to long access times**

New parallel programming paradigms (more in a minute) aim to allow easy partitioning and replication of data.

Enables programmers (or algorithms on our behalf) to easily dynamically replicate and partition.

For analytics the above is often less of an issue.

More of an issue is the extra costs: when big data technology makes things slower

Key-value stores as a data structure (rather than binary blobs - files).

- A very large dictionary.
- **Provide a way to break data into logical groups**
 - Operations can then be specified to run on these groups in parallel
 - Distribute + Computate



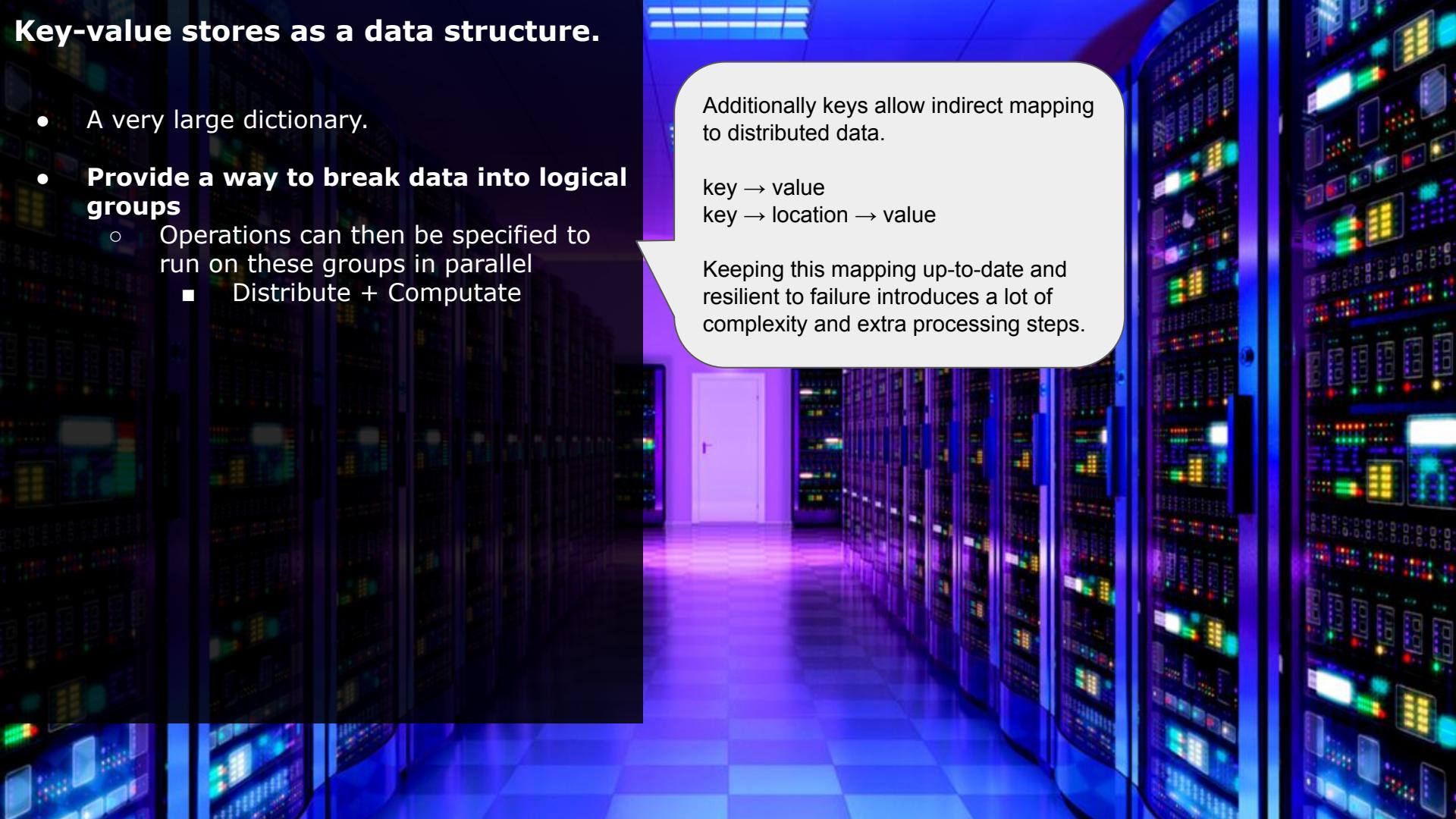
Key-value stores as a data structure.

- A very large dictionary.
- **Provide a way to break data into logical groups**
 - Operations can then be specified to run on these groups in parallel
 - Distribute + Compute

Additionally keys allow indirect mapping to distributed data.

key → value
key → location → value

Keeping this mapping up-to-date and resilient to failure introduces a lot of complexity and extra processing steps.



Key-value stores as a data structure.

- A very large dictionary.
- **Provide a way to break data into logical groups**
 - Operations can then be specified to run on these groups in parallel
 - Distribute + Compute
- **IMPORTANT:** Not all tasks can be completed by a divide and conquer (in parallel) approach! Not all problems are suitable for parallel computation.
 - Eg. the overall mean student performance computed as the average of mean module performance
 - here there are two steps that must be done sequentially (1st compute the per module mean and then take these results and compute the overall mean)



FBA: 76, 62, 52

D@S: 55, 66

AVG(FBA)=63.3

AVG(D@S)=60.5

$$(63.3+60.5)/2 = 61.9$$

vs.

$$(76+62+52+55+66)/5 = 62.2$$

Key-value stores as a data structure.

- A very large dictionary.
- **Provide a way to break data into logical groups**
 - Operations can then be specified to run on these groups in parallel
 - Distribute + Computate
- **IMPORTANT:** Not all tasks can be completed by a divide and conquer (in parallel) approach! Not all problems are suitable for parallel computation.
 - Eg. the overall mean student performance computed as the average of mean module performance
- How we think when programming distributed computing (map-reduce paradigm, spark) will be based on this data structure
- Levels of abstraction are now built on top to hide this, but understanding is still important when things break or go slowly (technology is new, happens more than we'd like)



- Tables can be logically constructed
 - Keys = columns
 - Values = data in the column
 - OR
 - Keys = row IDs
 - Values = rows
- Key value pairs can conceptually form tables.
- So you can see how we start to get SQL built on top..... however.... not all SQL tasks are easy (quick) to do in parallel.

Summary.

Key → Value pairs can be easily distributed. Basis for distributed processing paradigm coming up.

Fast access.

There is still significant complexity. It is just hidden from you.

[setup and maintenance can be harder]





Trade-off between consistency and availability.



Data at Scale

Dr. Evgeniya Lukinova



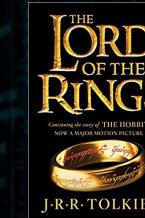
NLAB:

Data at Scale

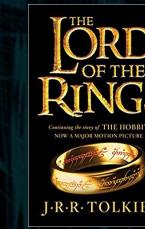
Data replicated and distributed for speed
(high availability)
and in case of network
failures
(partition tolerance)



Global stock:
1



Global stock:
1



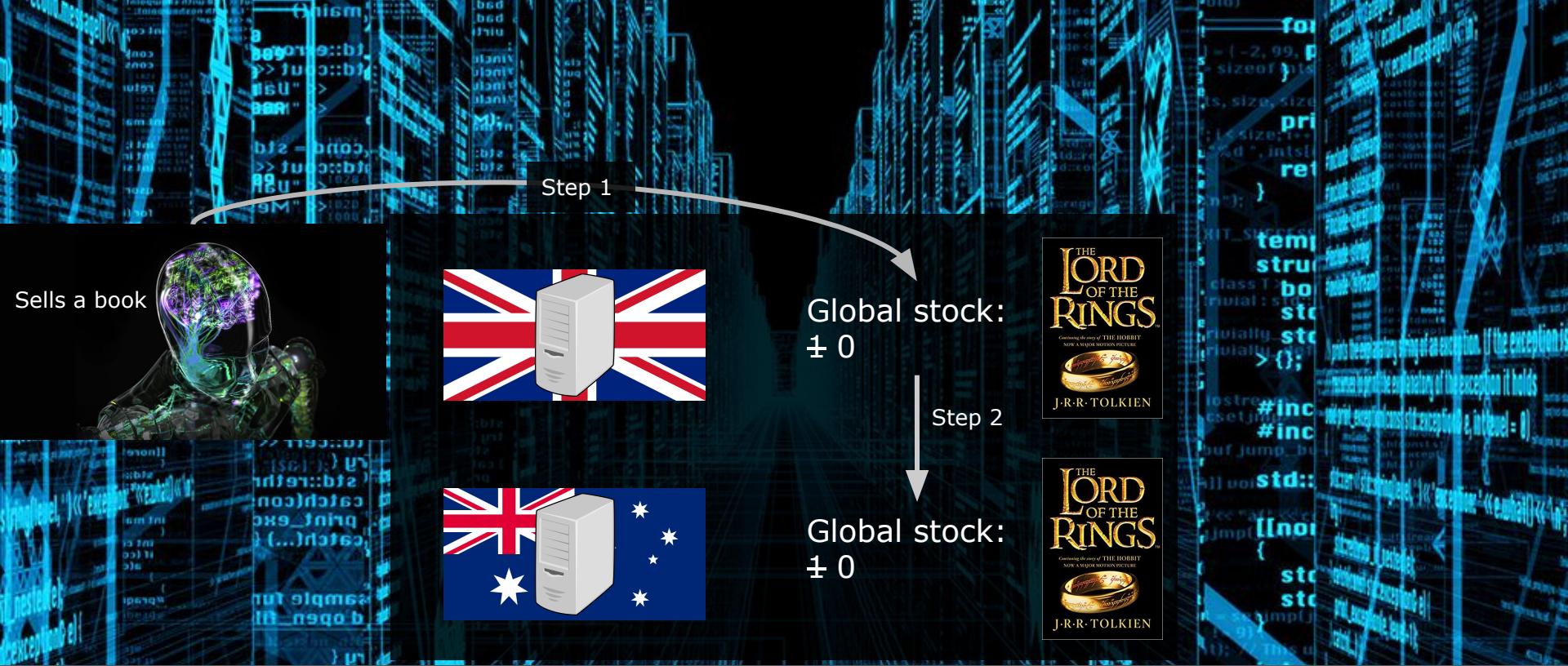
Dr. Evgeniya Lukinova



NLAB:

Data at Scale

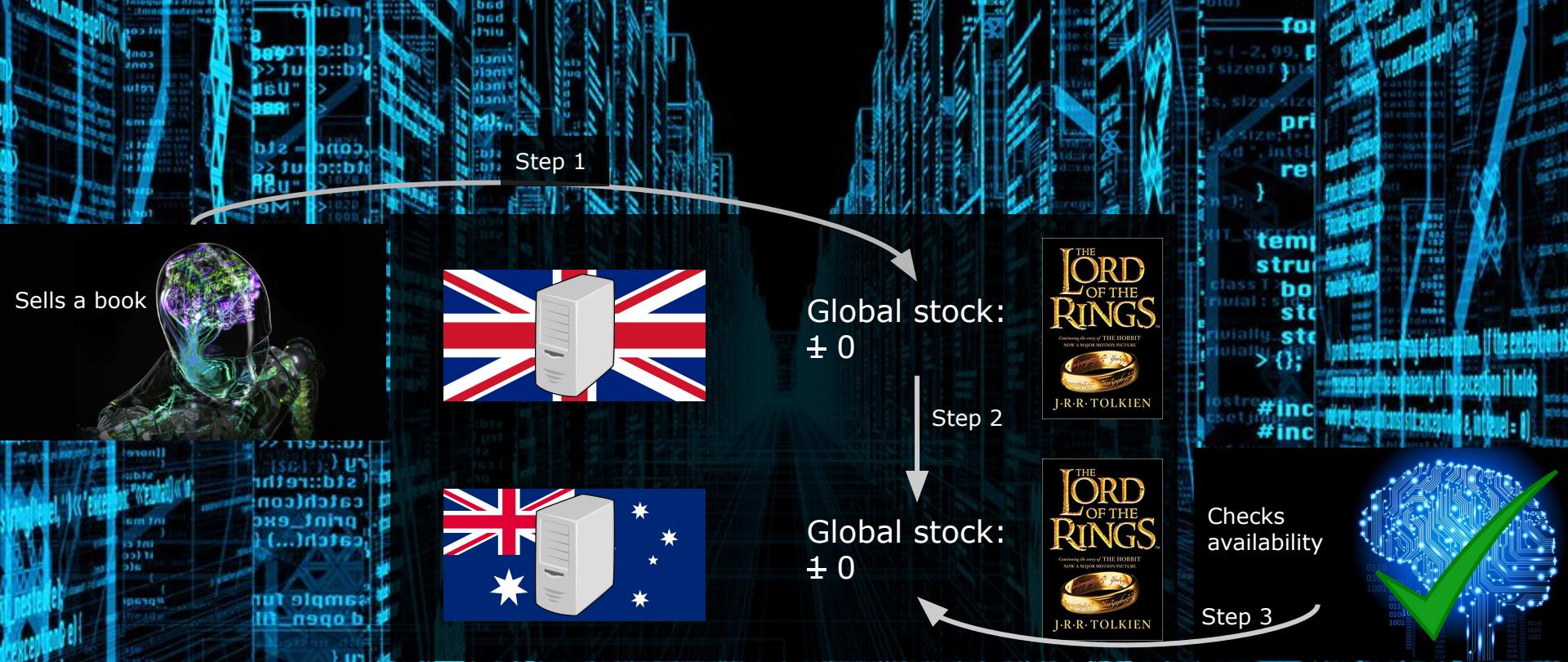
Dr. Evgeniya Lukinova



NLAB:

Data at Scale

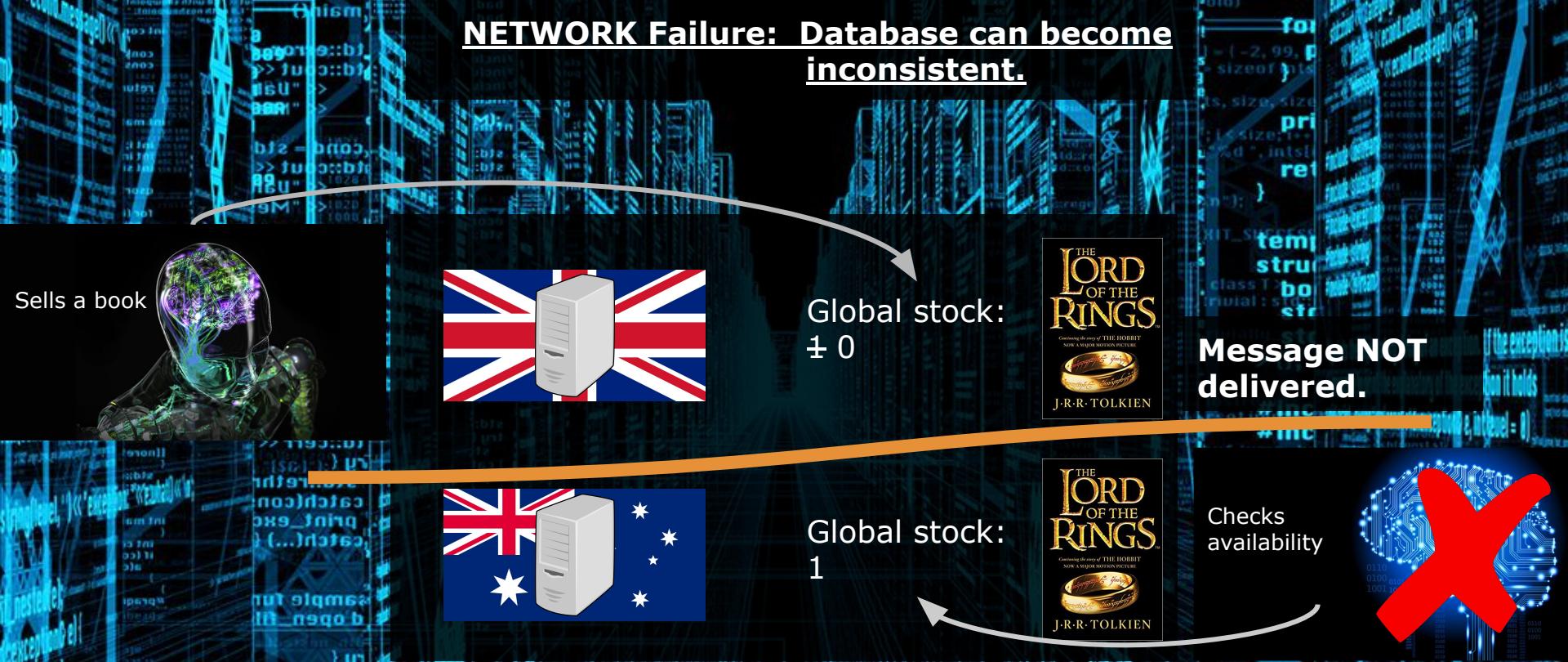
Dr. Evgeniya Lukinova



Data at Scale

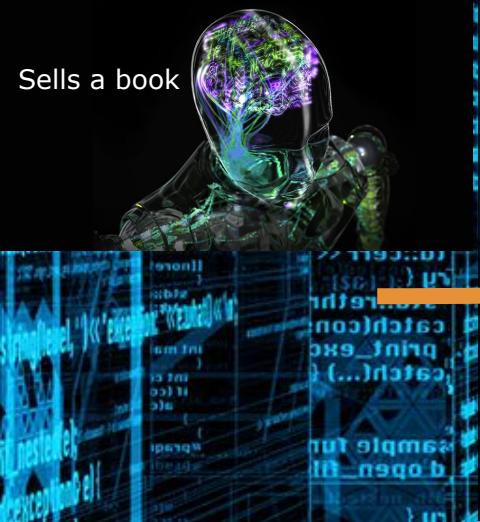
Dr. Evgeniya Lukinova

NETWORK Failure: Database can become inconsistent.

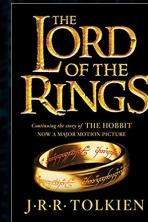


Trade-off,
consistency vs.
availability

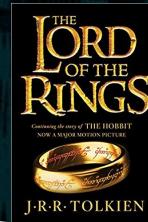
Could monitor link and messages (via receipts), if down throw error.
Don't sell. Or don't list book availability.



Global stock:
 ± 0

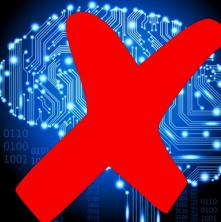


Global stock:
1

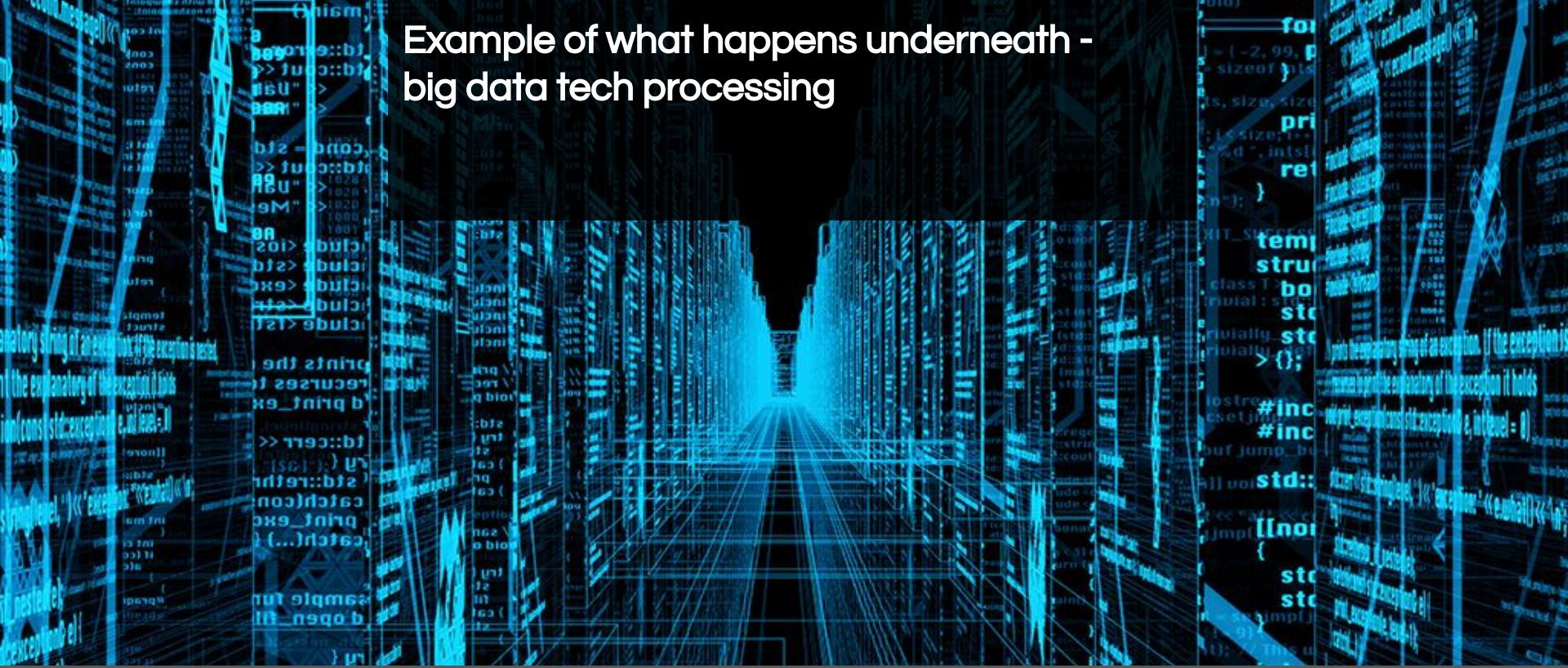


**Message NOT
delivered.**

Checks
availability



Example of what happens underneath - big data tech processing



Data at Scale

Dr. Evgeniya Lukinova

From storage to processing...

Map-Reduce

Traditional parallelism.

Data is brought to the compute.
Bottleneck?



From storage to
processing...

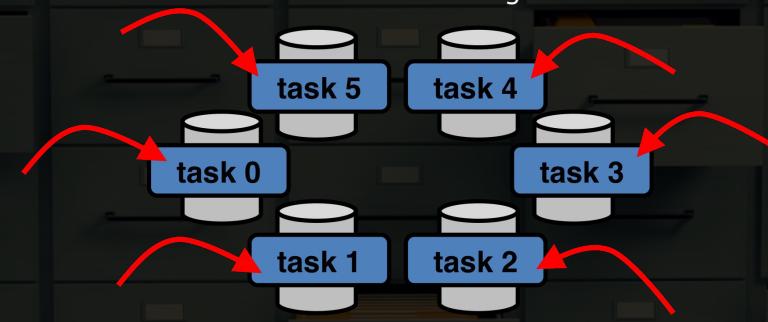
Map-Reduce

Map-Reduce parallelism.

Compute is (already)
moved to the data!

Assume data is already stored
in a distributed way in a
key-value store*.

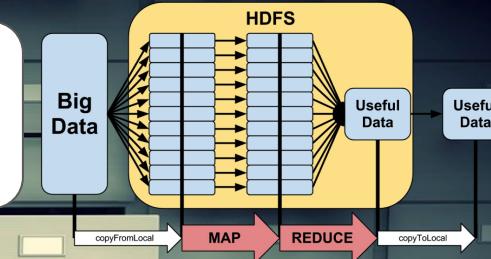
Assume compute exists on each
storage node.



Task: Find the number of unique 1 character words, 2 character words... in 50,000 blog posts.

Map-Reduce

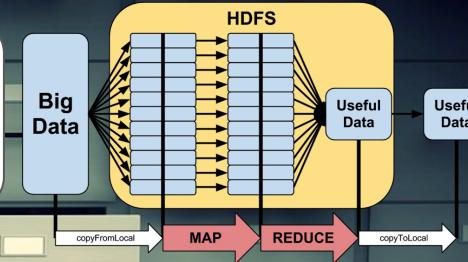
Multi-stage compute paradigm



Task: Find the number of unique 1 character words, 2 character words... in 50,000 blog posts.

Map-Reduce

Multi-stage compute paradigm



Map stage (**distributed**).

Takes data **already on the computer**, maps it into **appropriate** (key, values) pairs.

Per document (independently, in parallel)
place all words of same length in different buckets.

Each worker gets 50 blog posts
(id, blog).

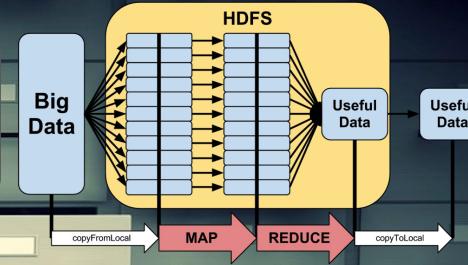
For each word in each blog list:
(char_ct, word)

m1: [(1,"a"),(5, "hello"),(2, "if"), (1, "l")]
m2: [(2, "an"),(5, "break"),(3, "the"),(1, "a")]

Task: Find the number of unique 1 character words, 2 character words... in 50,000 blog posts.

Map-Reduce

Multi-stage compute paradigm



Map stage (**distributed**).

Takes data **already on the computer**, maps it into **appropriate** key, values pairs.

Per document (independently, in parallel)
place all words of same length in different buckets.

Each worker gets 50 blog posts
(id, blog).

For each word in each blog list:
(char_ct, word)

m1: [(1,"a"),(5,"hello"),(2,"if"), (1,"l")]
m2: [(2,"an"),(5,"break"),(3,"the"),(1,"a")]

Group stage.

All values with the same key grouped and sent to the same node for compute.

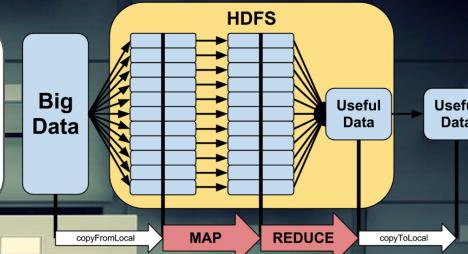
Merge all buckets with the same labels.
Redistribute the buckets amongst the workers.

r1: (1, ["a", "l", "a",.....])
r2: (2, ["if", "an",])
r3: (3, ["the",])
r4: (5, ["hello","break",])

Task: Find the number of unique 1 character words, 2 character words... in 50,000 blog posts.

Map-Reduce

Multi-stage compute paradigm



Map stage (**distributed**).

Takes data, maps it into **appropriate** key, values pairs.

Per document (independently, in parallel)
place all words of same length in different buckets.

Each worker gets 50 blog posts
(id, blog).

For each word in each blog list:
(char_ct, word)

m1: [(1,"a"),(5,"hello"),(2,"if"), (1,"l")]
m2: [(2,"an"),(5,"break"),(3,"the"),(1,"a")]

Group stage.

All values with the same key grouped and sent to the same node for compute.

Merge all buckets with the same labels.
Redistribute the buckets amongst the workers.

r1: (1, ["a", "l", "a",.....])
r2: (2, ["if", "an",])
r3: (3, ["the",])
r4: (5, ["hello", "break",])

Reduce stage.

Takes all key-value pairs with a given key. Performs some compute to get a value.

Per bucket (independently and in parallel)
count the number of distinct words.

(if there really was only this little data)

r1: (1, ["a", "l", "a"]) → 2
r2: (2, ["if", "an"]) → 2
r3: (3, ["the"]) → 1
r4: (5, ["hello", "break"]) → 2

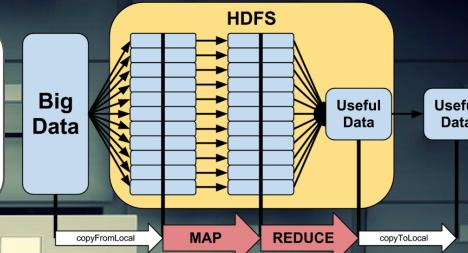
DONE!

(fetch the counts, make a list and return it)

Task: Find the number of unique 1 character words, 2 character words... in 50,000 blog posts.

Map-Reduce

Multi-stage compute paradigm



Map stage (**distributed**).

Takes data, maps it into **appropriate** key, values pairs.

Per document (independently, in parallel)
place all words of same length in different buckets.

DISTRIBUTED, say 1,000 nodes
(wherever the data is)

Independently processing input data
chunks of data that can be processed
independently.

Programmer must define.

Group stage.

All values with the same key
grouped and sent to the same
node for compute.

Merge all buckets with the same labels.
Redistribute the buckets amongst the
workers.

NOT DISTRIBUTED

Fast non-independent task (system)

Reduce stage.

Takes all key-value pairs with a
given key. Performs some
compute to get a value.

Per bucket (independently and in parallel)
count the number of distinct words.

DISTRIBUTED, say 100 nodes

Independently process pre-defined
independent tasks

Programmer must define.

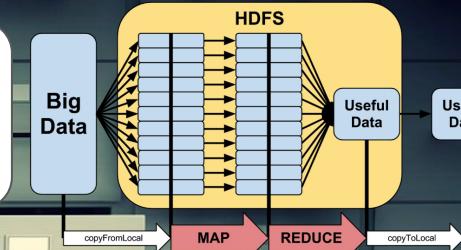
DONE!

(fetch the counts, make a list
and return it)

Task: Find the number of unique 1 character words, 2 character words... in 50,000 blog posts.

Map-Reduce

Multi-stage compute paradigm



Map stage (**distributed**).

Takes data, maps it into **appropriate** key, values pairs.

Per document (independently, in parallel)
place all words of same length in different buckets.

DISTRIBUTED, say 1,000 nodes
(wherever the data is)

Independently processing input data
chunks that can be processed
independently.

Programmer must define.

Group stage.

All values with the same key
grouped and sent to the same
node for compute.

Merge all buckets with the same labels.
Redistribute the buckets amongst the
workers.

NOT DISTRIBUTED

Fast non-independent task (system)

Reduce stage.

Takes all key-value pairs with a
given key. Performs some
compute to get a value.

Per bucket (independently and in parallel)
count the number of distinct words.

DISTRIBUTED, say 100 nodes

Independently process pre-defined
independent tasks

Programmer must define.

Paradigm forces **you** to
re-cast your problem into
chunks of **independent
computation** in a **given
structure**.

If you can do this, it can be
done fast!

Not all problems can be. **And
we're at algorithm
design....**

Map-Reduce

How we can use it...

Map-Reduce programming.

Chain of independent (Map, group, reduce) operations **on set of key-value data.**

Cannot do everything, still need to be embedded in sequential programming.

Clean data, convert data to correct format



Call someone elses mapreduce function



Call someone elses mapreduce function



Post-process & visualize the results

Map-Reduce

How we can use it...

Map-Reduce programming.

Chain of independent (Map, group, reduce) operations **on set of key-value data**.

Cannot do everything, still need to be embedded in sequential programming.

Significantly more complex than Python.

→ **Cannot do everything.**

→ **Low level interface** (other's MapReduce functions do not do the high-level tasks we might want to)

Probably not for us.

Clean data, convert data to correct format



Call someone elses mapreduce function



Call someone elses mapreduce function



Post-process & visualize the results

We'll be having an introduction to this way of programming via a Demo.

Using Spark, which is built on Map-Reduce but better!

We're not algorithm designers...

But as people build on top of this and provide better libraries and abstractions, why not!

(but since these are harder to design, techniques availability may lag)

However, there is a lot of extra complexity.

Data movement. Small data, poor implementations = worse performance!!!

Distributed linear regression? Why not.

Distributed deep learning.
Almost required.

Premature optimization is the root of all evil*.

(evil = frustration + bugs + time lost)



Builds on Map-Reduce but faster (**10-100x speedup**).



OK, so what good libraries do we have?

For Data processing:

- SparkSQL
- Cockroachdb
- Google Spanner

For Analytics:

- Spark MLib
- Spark
- MapReduce

Can be sped up by correct data storage:

- Delta lake (from Databricks - SparkSQL)
- Cockroachdb (PostgreSQL compatible)
- Google Spanner (own SQL variant)

OK, so what good libraries do we have?

For Data processing:

- SparkSQL
- Cockroachdb
- Google Spanner

Can be sped up by correct data storage:

- Delta lake (from Databricks - SparkSQL)
- Cockroachdb (PostgreSQL compatible)
- Google Spanner (own SQL variant)

For Analytics:

- Spark MLlib
- Spark
- MapReduce

The implementation of distributed SQL is getting quite good.

Distributed ML algorithms is harder.

Silver lining - often, after preprocessing / feature engineering, data is no longer "big data".



Builds on Map-Reduce but faster
(10-100x speedup).



Higher level of abstraction for
doing machine learning & data
analytics

Slightly higher level than
MapReduce. Required to be
aware of because Spark ML
doesn't quite hide everything
from us yet.

What we want to use.



In MapReduce, after each
(map, group, reduce) data is
written to disk and reloaded.

Rather than write things to disk,
**Spark provides a graph/chain
processing paradigm**. Moves
away from key-value pairs to
Resilient Distributed Datasets
(RDDs)



Builds on Map-Reduce but faster
(10-100x speedup).



Higher level of abstraction for
doing machine learning & data
analytics

Slightly higher level than
MapReduce. Required to be
aware of because Spark ML
doesn't quite hide everything
from us yet.

What we want to use.

In MapReduce, after each (map, group, reduce) data is written to disk and reloaded.

Rather than write things to disk, **Spark provides a graph/chain processing paradigm**. Moves away from key-value pairs to Resilient Distributed Datasets (RDDs)

Intuitively, programs are written as a set of consecutive (from your point of view) number of transforms on a common data structure.

Somewhat like SQL!

Chain length is arbitrary, execution order is globally optimised.

Data is kept in **local memory or redistributed** based on automatic analysis of the chain. **10-100x faster!**



Builds on Map-Reduce but faster (**10-100x speedup**).



Higher level of abstraction for doing machine learning & data analytics

Slightly higher level than MapReduce. Required to be aware of because Spark ML doesn't quite hide everything from us yet.

What we want to use.

**Ok, so we are not using
key-value pairs, what
now!**

RDDs are built on key-value
pairs.

Collections (lists) of data with
either explicit partitioning for
parallelizing
→ list is a list of key-value pairs

or implicit partitioning if not
→ if the list is something else

**More operations and low level
control of parallelization if it is
a list of key-value pairs**

Ok, so we are not using key-value pairs, what now!

RDDs are built on key-value pairs.

Collections (lists) of data with either explicit partitioning for parallelizing
→ list is a list of key-value pairs

or implicit partitioning if not
→ if the list is something else

More operations and low level control of parallelization if it is a list of key-value pairs

Resilient Distributed Datasets (RDDs):

A collection of data.

E.g:

a collection of numbers:
[1,2,3,4]

a collection of key value pairs:
[('a',7),('a',2),('b',2)]

a collection of tuples:
[(10, [0.5,0.1]), (4, [0.5,0.4])]

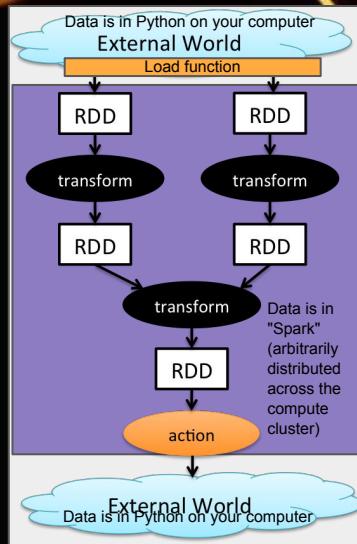
Introducing...



When you load data into spark you load it in a special format that can be automatically distributed.

RDDs (Resilient Distributed Datasets)
→ A collection numbers, tuples,
key-value pairs....

RDDs live in the spark cluster NOT
your computer.



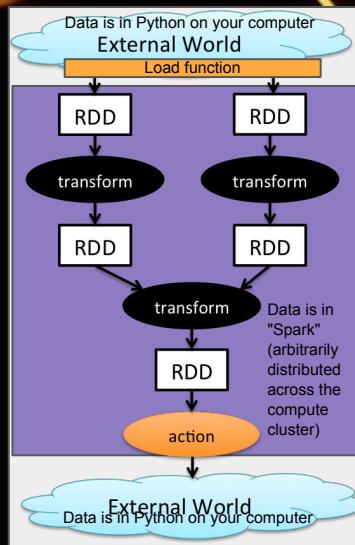
Introducing...



When you load data into spark you load it in a special format that can be automatically distributed.

RDDs (Resilient Distributed Datasets)
→ A collection numbers, tuples, key-value pairs....

RDDs live in the spark cluster NOT your computer.



We load data from Python (our computer) into the spark cluster in RDDs via special functions.

We process data via **transforms** in Spark (away from our computer).

Transforms: Actions on RDD(s) that return RDD(s).

We move the data back to our computer (Python) via an **action**.

* Most common. Some exceptions exist but these are well beyond the scope of this course.

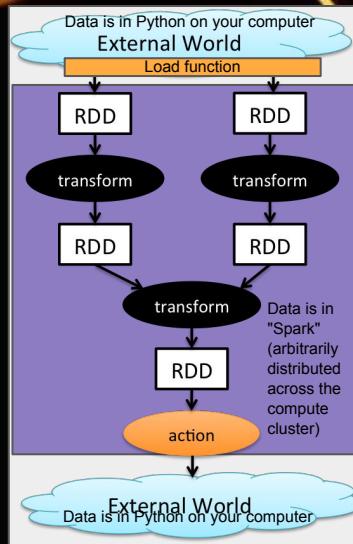
Introducing...



When you load data into spark you load it in a special format that can be automatically distributed.

RDDs (Resilient Distributed Datasets)
→ A collection numbers, tuples, key-value pairs....

RDDs live in the spark cluster NOT your computer.



We load data from Python (our computer) into the spark cluster in RDDs via special functions.

We process data via **transforms** in Spark (away from our computer).

Transforms: Actions on RDD(s) that return RDD(s).

We move the data back to our computer (Python) via an **action**.

Transforms are lazy!

They form a chain of processing that are not done until an action is requested.

Why? The computer will optimize the way it processes based on the set of actions.

Adding a new transform could drastically alter the best way of doing things.

Also: If no one needs (requests) the output, why do the processing?

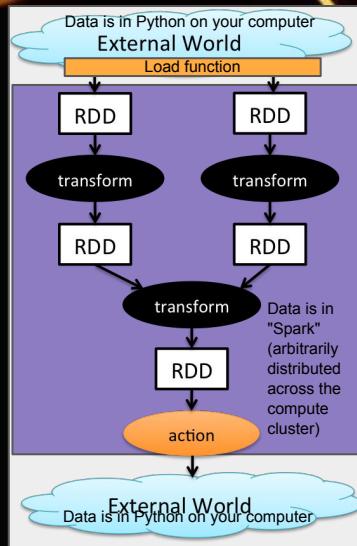
* Most common. Some exceptions exist but these are well beyond the scope of this course.

Introducing...



A common transform is `reduceByKey`

- is a method of an RDD
- takes a **compatible function** as a parameter



A **compatible function** here is one of the form:

```
def fn(a, b):  
    <some processing>  
    return value
```

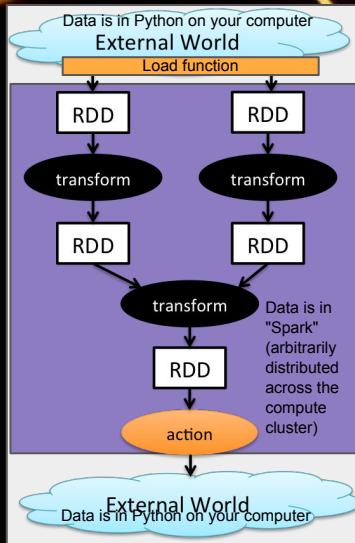
e.g.

```
def fn(a, b):  
    return a + b
```

Introducing...



- A common transform is `reduceByKey`
- is a method of an RDD
- takes a **compatible function** as a parameter
- all item pairs in the RDD are grouped (put into buckets) based on the key value
- for each bucket the function is repeatedly applied to pairs of values until a single value is computed
- returns a new RDD (one key→ value pair per bucket)



A **compatible function** here is one of the form:

```
def fn(a, b):  
    <some processing>  
    return value
```

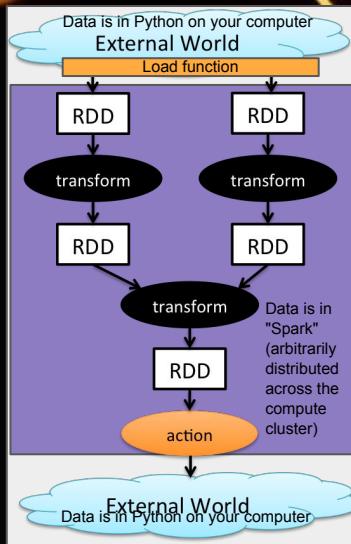
e.g.

```
def fn(a, b):  
    return a + b
```

Introducing...



- A common transform is `reduceByKey`
- is a method of an RDD
- takes a **compatible function** as a parameter
- all item pairs in the RDD are grouped (put into buckets) based on the key value
- for each bucket the function is repeatedly applied to pairs of values until a single value is computed
- returns a new RDD (one key→ value pair per bucket)



A **compatible function** here is one of the form:

```
def fn(a, b):  
    <some processing>  
    return value
```

Like SQL, a finite number of transformations exist.

The way you combine them and the functions you pass enable a wide (**but not complete**) range of processing.

Unfortunately, **not all problems can be cast in this** (map/reduce) way.

This paradigm is not complete. Also, some tasks are very hard to convert to this way of thinking.

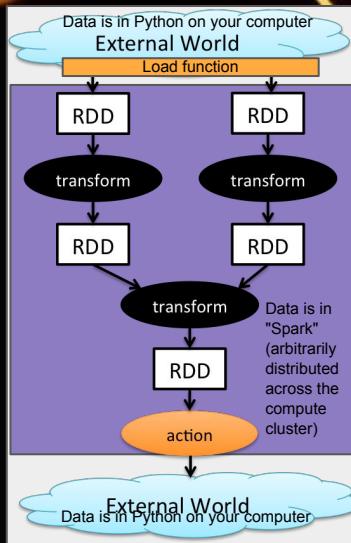
Still need languages like Python.

e.g.
def fn(a, b):
 return a + b

Introducing...



- A common transform is `reduceByKey`
- is a method of an RDD
- takes a **compatible function** as a parameter
- all item pairs in the RDD are grouped (put into buckets) based on the key value
- for each bucket the function is repeatedly applied to pairs of values until a single value is computed
- returns a new RDD (one key→ value pair per bucket)



A **compatible function** here is one of the form:

```
def fn(a, b):  
    <some processing>  
    return value
```

Like SQL, a finite number of transformations exist.

The way you combine them and the functions you pass enable a wide (**but not complete**) range of processing.

Unfortunately, **not all problems can be cast in this** (map/reduce) way.

Here we are **abstracting away** how to access and schedule across a **distributed system**.

Still, Spark is a lower level of abstraction than SQL.

This paradigm is not complete. Also, some tasks are very hard to convert to this way of thinking.

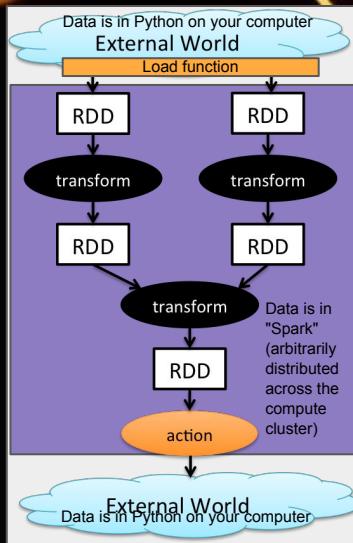
Still need languages like Python.

* Most common. Some exceptions exist but these are well beyond the scope of this course.

Introducing...



- A common transform is `reduceByKey`
- is a method of an RDD
- takes a **compatible function** as a parameter
- all item pairs in the RDD are grouped (put into buckets) based on the key value
- for each bucket the function is repeatedly applied to pairs of values until a single value is computed
- returns a new RDD (one key→ value pair per bucket)



Still saying what to do. Can't do everything. Still encapsulated in e.g. Python.

Spark **does** provide an SQL like transforms and actions! **Not complete. Not guaranteeing ACID etc. Not necessarily faster.**

More broadly this is what **NewSQL databases** are looking at! Abstraction on top of this abstraction...

This level can be good for analytics though... let's look at this more!

In SQL we are abstracting navigational access.

Here we are **abstracting away** how to access and schedule across a **distributed system**.

Still, Spark is a lower level of abstraction than SQL.

Like SQL, a finite number of transformations exist.

The way you combine them and the functions you pass enable a wide (**but not complete**) range of processing.

Unfortunately, **not all problems can be cast in this** (map/reduce) way.

This paradigm is not complete. Also, some tasks are very hard to convert to this way of thinking.

Still need languages like Python.

* Most common. Some exceptions exist but these are well beyond the scope of this course.

Introducing...

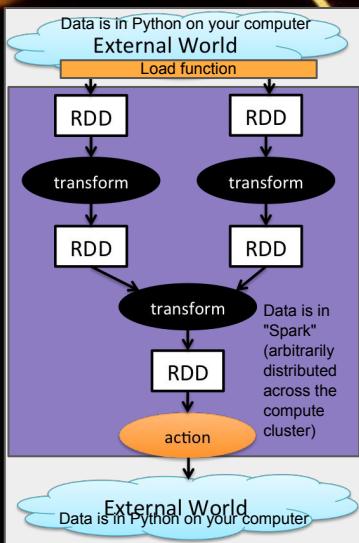


Transforming RDDs is still quite low level.

Machine learning (building supervised/unsupervised models) has a fixed common set of steps.

Someone else should work out a shorthand version of the process that parallelize automatically.

→ **Less general (suitable for ML only), but higher level of abstraction. Yes please.**

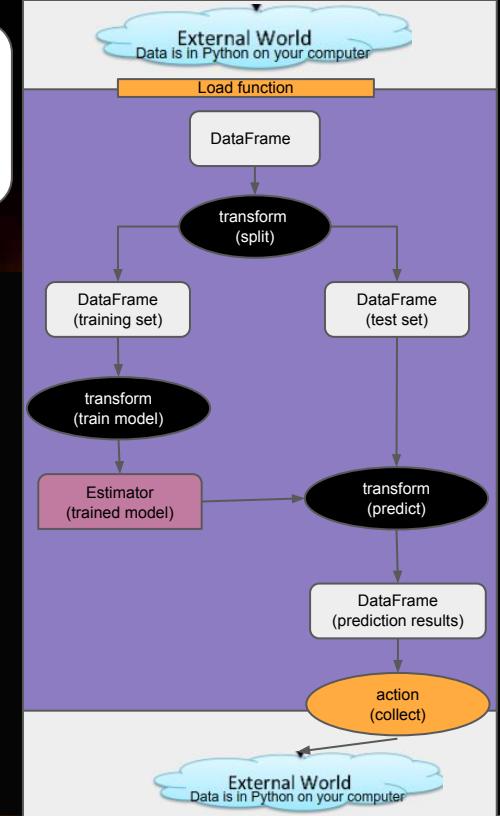


Rather than RDDs → DataFrames.

Spark DataFrames:
RDDs with extra information.

Spark ML we transform
DataFrames.

Unlike Spark, in Spark ML we also have **estimators** (models). We can fit these to end up with **trained models**.



What you need to remember from today:

- What a key-value data structure is.
- Accept that key-value pairs can be easily distributed and form the basis of most parallel processing paradigms.

Solutions part-way between

- Processing key-value pairs in parallel can not always be done: **Task and data dependent.**

Current programming languages either do not give the computer enough freedom to decide when to do things in parallel and when not to (Python)
OR

They are not smart enough yet to work it out (SQL for data manipulation, unknown for full data analytics)

- Requires us to think / learn a new programming paradigm OR use libraries so we still think in linear steps but each step involves (hidden) parallel processing.



Map-Reduce is a parallel programming paradigm.

Spark is a parallel programming framework.

Typically backed by a **distributed file system**.

+

A distributed resource manager.

Can't do everything.

Embedded in a traditional programming language.

The open version of this ecosystem is known as **Hadoop**.

Spark runs on top of Hadoop. Hadoop incorporates MapReduce.

With Hadoop

Distributed data, distributed processing

Without Hadoop

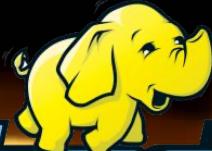
Centralized data, distributed processing



Hadoop **tries** to provide a distributed file system that **looks like a traditional file system.**

The Hadoop file system.

Arbitrary files can be stored.



hadoop



Hadoop **tries** to provide a distributed file system that **looks like a traditional file system**.

The Hadoop file system.

Arbitrary files can be stored.

However, to ensure correct automatic distribution and use by algorithms you must use specific formats.

For Spark we load into Resilient Distributed Dataset (RDD) structures or DataFrames for processing.

Save DataFrames to Columnar file format!

These are ones that Hadoop knows how to cut into key-value pairs*!!

A common format is tab-delimited text files.

Others:
Sequence files (basically key→ value pairs)
Columnar file formats (towards tables as an abstraction over key-value pairs). These become **DataFrames** in Spark!

Spark Demo
Almost.



FUNCTIONS

```
def add5( num_in ):  
    return num_in + 5
```

Functions comprise of a **name** (human readable versions of memory address where they are stored) and a **definition**.

Just like variables comprise of a **name** and a **value (definition)**.

e.g. age = 3
 my_book = 



FUNCTIONS

```
def add5( num_in ):  
    return num_in + 5
```

```
def add5( num_in ):  
    return num_in + 5  
  
def run_fn( fn, num_in ):  
    return fn( num_in )
```

```
>> run_fn( add5, 2)  
>> 7
```

Passing Functions

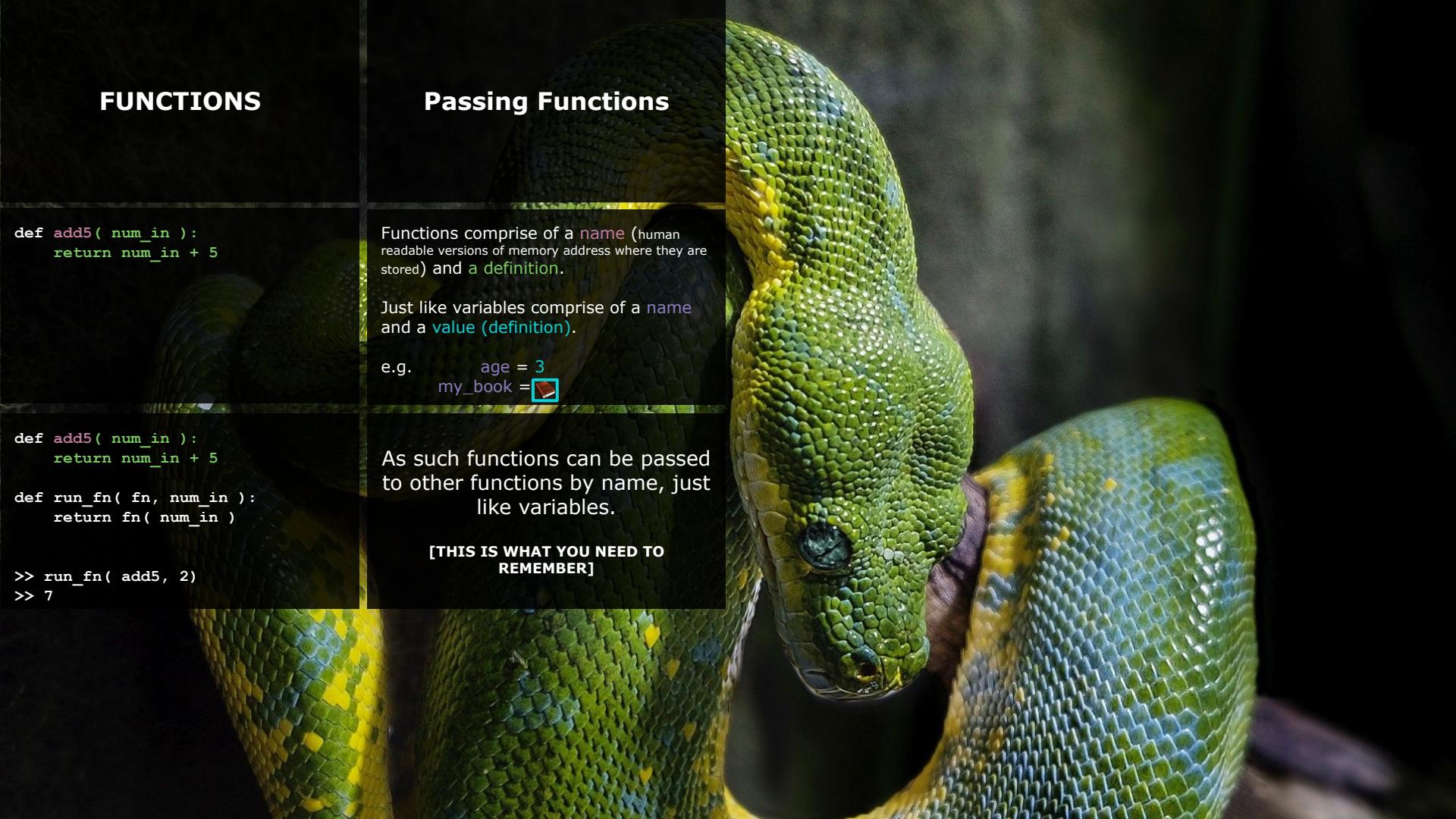
Functions comprise of a **name** (human readable versions of memory address where they are stored) and a **definition**.

Just like variables comprise of a **name** and a **value (definition)**.

e.g. age = 3
 my_book = 

As such functions can be passed to other functions by name, just like variables.

[THIS IS WHAT YOU NEED TO REMEMBER]



FUNCTIONS

```
def add5( num_in ):  
    return num_in + 5
```

```
def add5( num_in ):  
    return num_in + 5  
  
def run_fn( fn, num_in ):  
    return = fn( num_in )
```

```
>> run_fn( add5, 2)  
>> 7
```

Passing Functions

Functions comprise of a **name** (human readable versions of memory address where they are stored) and a **definition**.

Just like variables comprise of a **name** and a **value (definition)**.

e.g. age = 3
 my_book = 

As such functions can be passed to other functions by name, just like variables.

[THIS IS WHAT YOU NEED TO REMEMBER]

Anonymous Functions

Sometimes we do not declare variables but directly use them in functions:

```
print('at')
```

rather than

```
my_str = 'hi'  
print(my_str)
```

FUNCTIONS

```
def add5( num_in ):  
    return num_in + 5
```

```
def add5( num_in ):  
    return num_in + 5  
  
def run_fn( fn, num_in ):  
    return fn( num_in )  
  
>> run_fn( add5, 2)  
>> 7
```

Passing Functions

Functions comprise of a **name** (human readable versions of memory address where they are stored) and a **definition**.

Just like variables comprise of a **name** and a **value (definition)**.

e.g. age = 3
 my_book = 

As such functions can be passed to other functions by name, just like variables.

[THIS IS WHAT YOU NEED TO REMEMBER]

Anonymous Functions

```
def add5( num_in ):  
    return num_in + 5
```

Is the same as:

```
lambda num_in: num_in + 5
```

We can do the same with functions via a keyword **lambda**.

```
def run_fn( fn, num_in ):  
    return fn( num_in )  
  
>> run_fn( lambda num_in: num_in + 5, 2)  
  
>> 7
```

FUNCTIONS

Passing Functions

Anonymous Functions

```
def add5( num_in ):  
    return num_in + 5  
  
Is the same as:  
  
lambda num_in: num_in + 5
```

We can do the same with functions via a keyword **lambda**.

```
def run_fn( fn, num_in ):  
    return fn( num_in )  
  
>> run_fn( lambda num_in: num_in + 5, 2)  
    >> 7
```

Passing function definitions is important in parallel processing as the code to run needs to be distributed to each compute node as well as the data.

As such functions can be passed to other functions by name, just like variables.

[THIS IS WHAT YOU NEED TO REMEMBER]

```
def add5( num_in ):  
    return num_in + 5
```

```
def add5( num_in ):  
    return num_in + 5  
  
def run_fn( fn, num_in ):  
    return fn( num_in )  
  
>> run_fn( add5, 2)  
    >> 7
```

Spark Demo

