

**How Do We Solve
the World's
Spreadsheet Problem?**

Alex Rasmussen

@alexras

Hi, I'm Alex!

@alexras
alexras.info

freenome
freenome.com



BITS ON DISK

bitsondisk.com



My Background

2009-2013: really fast sorting

2013-2016: data wrangling

2017-2018: cancer-fighting robots

I **think**/**worry** a lot
about spreadsheets.

Today's focus:

spreadsheet data

(for compute, feliennne.com)

This talk:

- 1. Spreadsheets are great**
- 2. Spreadsheets are a problem**
- 3. How we can fix it**

This talk:

- 1. Spreadsheets are great**
2. Spreadsheets are a problem
3. How we can fix it

What's so **great**
about spreadsheets?

Spreadsheets are

Ubiquitous

1.2 billion Office users (~16% of humans)

1.2 billion Office users (~16% of humans)

60 million Office 365 customers

1.2 billion Office users (~16% of humans)

60 million Office 365 customers

>5 million businesses use Google Apps

Spreadsheets are

Approachable

Microsoft Excel - SHEET2.XLS									
File Edit Formula Format Data Options Macro Window								Help	
D1		A	B	C	D	E	F	G	H
1	DESCRIPTION	This is a column!	Multiply						
2	I	5	17.5						
3	Like	12	42						
4	DOS	33.6	117.6						
5	Games	640	2240						
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									

Ready

NUM

Spreadsheets are

Flexible

Data grids

Data grids

Graphs

Data grids

Graphs

Anything tabular

Data grids

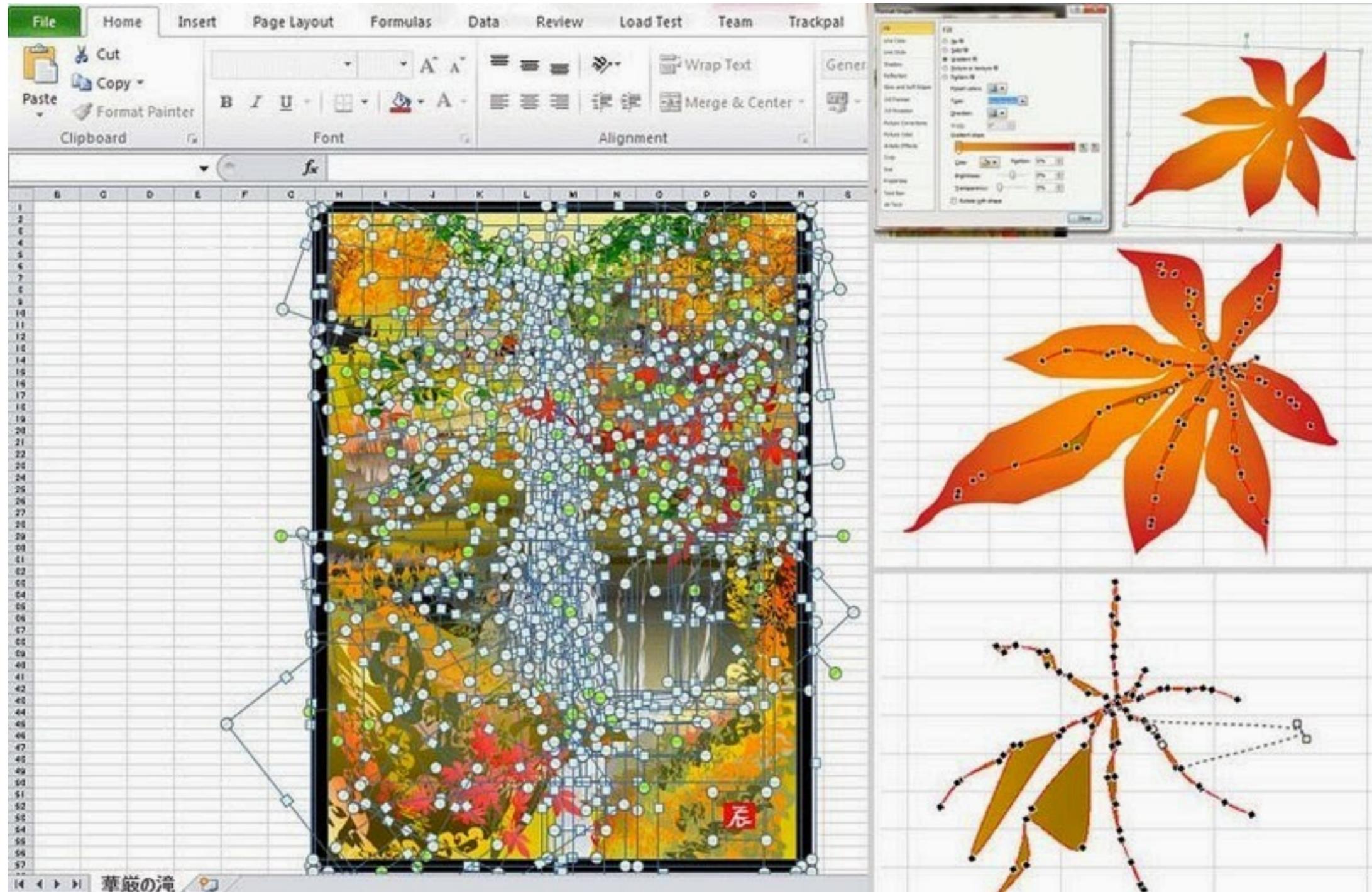
Graphs

Anything tabular

Full-scale “apps”



Tatsuo Horiuchi (b. 1940)
Kegon Falls, 2007
AutoShape on canvas



<https://pasokonga.com/>

So if spreadsheets are
ubiquitous, approachable,
and **flexible,**
what's the problem?

This talk:

1. Spreadsheets are great
- 2. Spreadsheets are a problem**
3. How we can fix it

Problem #1:

Data Types



GSV Arson Kite

@Phylan



ME: *makes typo while entering a number*

EXCEL: WAS THAT A DATE

ME: no I meant t-

EXCEL: THAT WAS A  DATE

ME: it doesn't even make sen-

EXCEL: MAY 12TH 1382. LOOK I EVEN FORMATTED IT.
IT IS THIS FOREVER

9:00 AM - Feb 1, 2018

♥ 64.8K 💬 17.5K people are talking about this



Automatic type
conversion can cause
serious problems.

DEC1

DEC1

12/1

RIKEN Identifier

2310009E13

RIKEN Identifier

2310009E13

2.31E+13

“We confirmed gene name errors in **987** supplementary files from **704** published articles (**19.6% of all articles**).”

<https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-1044-7>

False Equivalence

000123
= 00123
= 123

True if they're integers,
but **what if they're strings?**

Enumerated Types

Enumerated Types

“Prostate Cancer”

Enumerated Types

“Prostate Cancer”

“prostate cancer”

Enumerated Types

“Prostate Cancer”

“prostate cancer”

“prostatecancer”

Enumerated Types

“Prostate Cancer”

“prostate cancer”

“prostatecancer”

“PC”

Enumerated Types

“Prostate Cancer”

“prostate cancer”

“prostatecancer”

“PC”

“prostate”

Enumerated Types

“Prostate Cancer”

“prostate cancer”

“prostatecancer”

“PC”

“prostate”

“prostrate”

List Validations? Sheet Protection?

 Easy to add

 Easy to remove by accident

 Hard to enforce

Data loss!

False equivalence!

Ontological chaos!

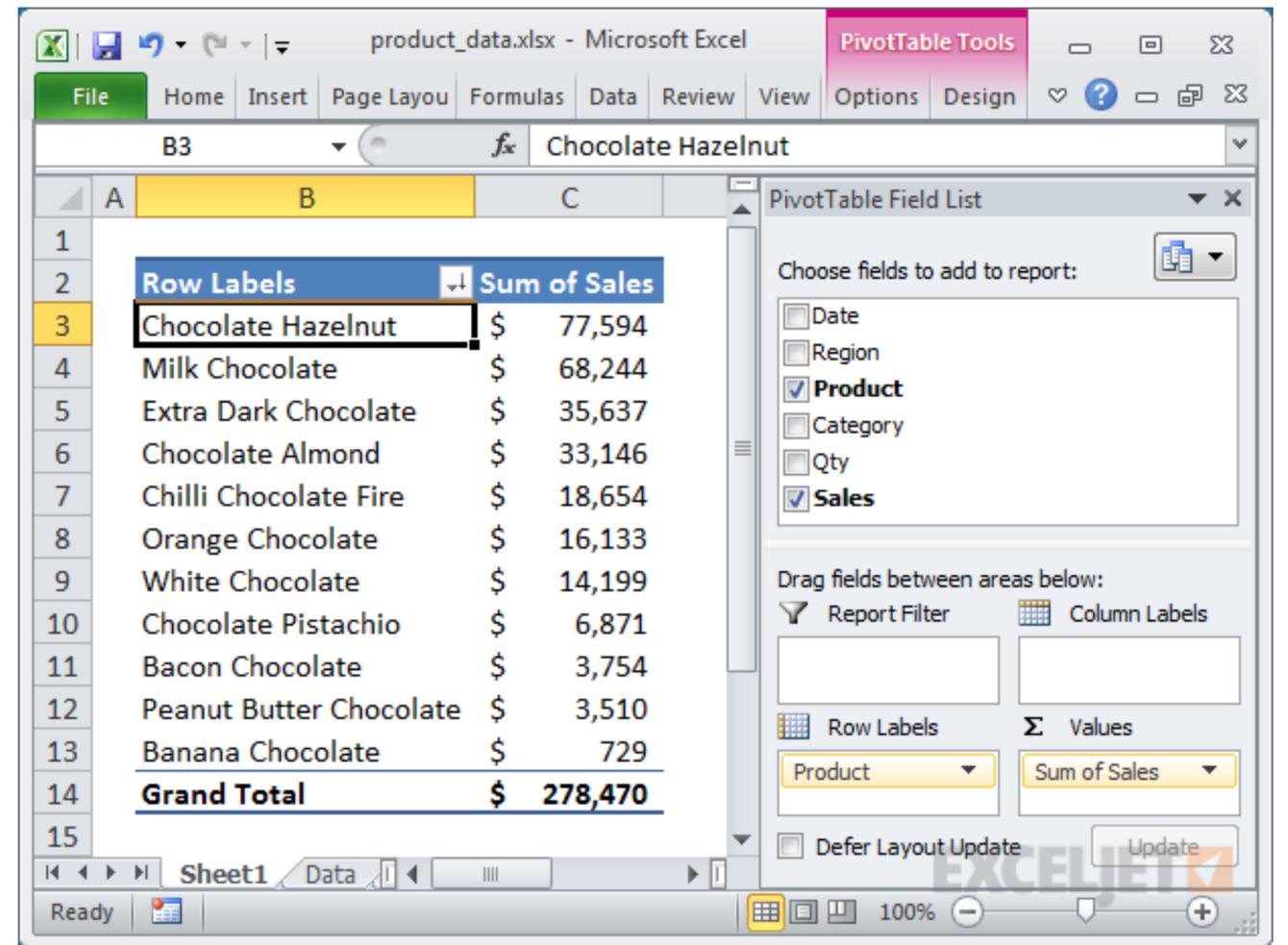
Mass hysteria!

Problem #2:

Queryability

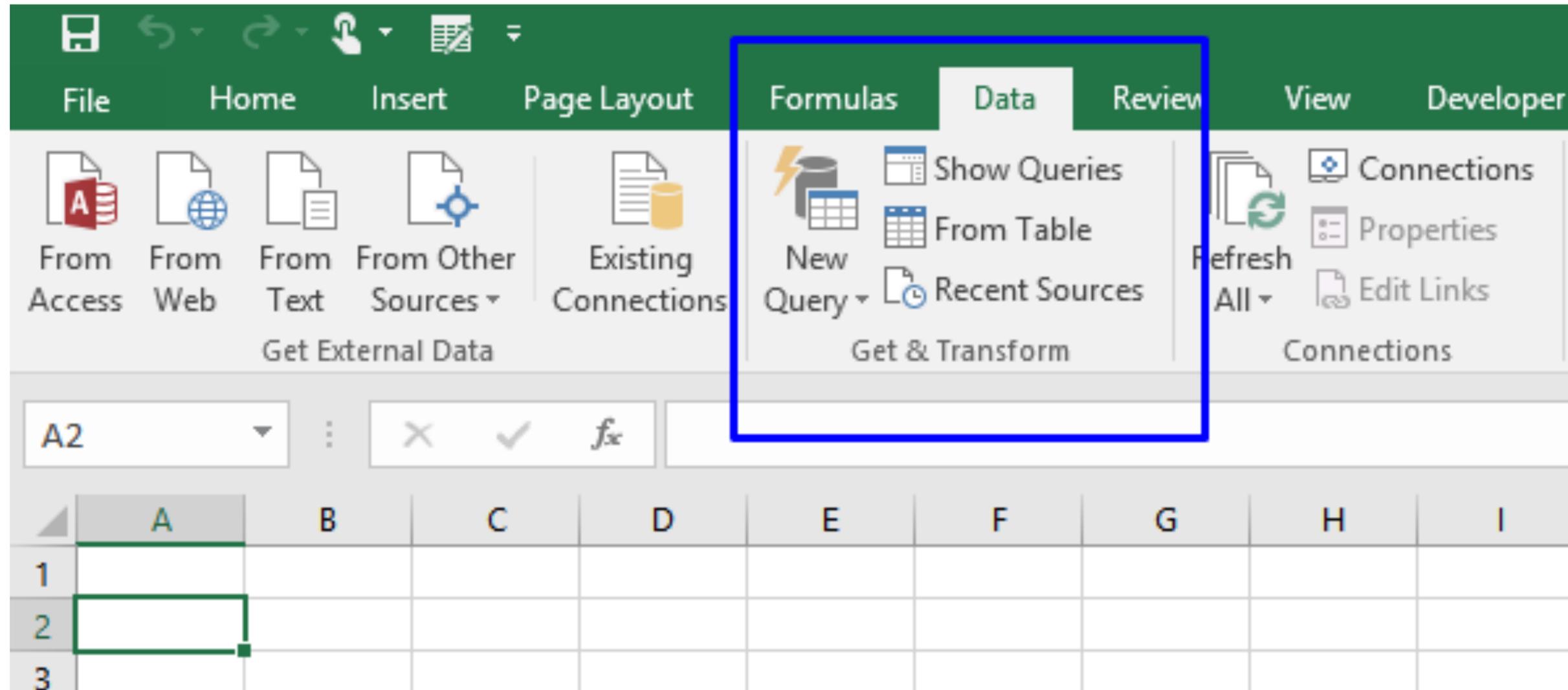
**Inside a
spreadsheet, things
are pretty good!**

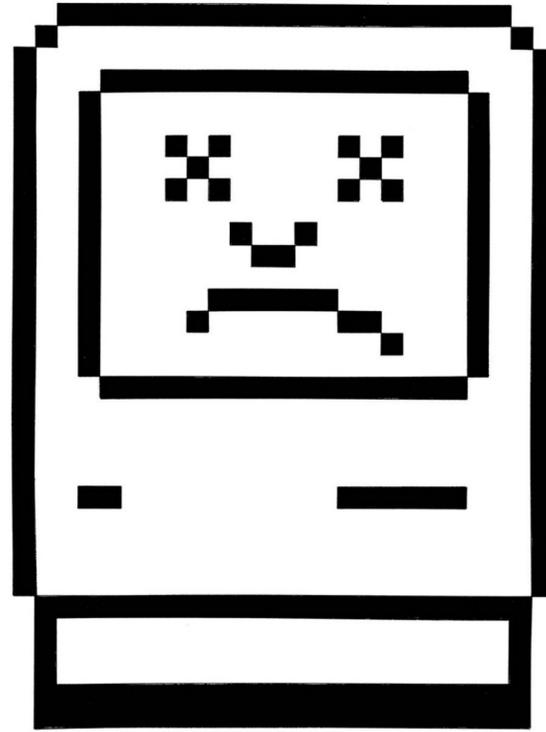
Formulas!
Pivot Tables!
Filters!



What about
querying **across**
spreadsheets?

Get and Transform





No Mac support.

Structure changes?

Type changes?

Column Renames?

Have fun re-loading.

And what
about **joins**?

There's VLOOKUP

```
=VLOOKUP ("Product 1",  
Prices!$A$2:$B$9,2,FALSE)
```

... but, like, eww.

Data **inside** a
spreadsheet is hard to
connect to data **outside**
that spreadsheet.

Summary:

Spreadsheets are
bad at types and
hard to query

This talk:

1. Spreadsheets are great
2. Spreadsheets are a problem
- 3. How we can fix it**



What about **databases**?

Databases are **great**
in ways that
spreadsheets **aren't**.

Databases are great at
data type **definition**
and **enforcement**.

So Many Types of Types!

Numeric

Enumerated

XML

Monetary

Geometric

JSON

Character

Network Address

Arrays

Binary

Bit String

Composite

Date/Time

Text Search

Range

Boolean

UUID

Pseudo-Types

Databases are
purpose-built for
queries and joins.

BUT

Databases

aren't as **approachable**

as spreadsheets.

The screenshot shows the phpMyAdmin interface for a MySQL database named 'world'. The left sidebar displays a tree view of databases, with 'world' expanded to show tables: 'City', 'Country', and 'CountryLanguage'. The main area shows the structure of three tables:

- world.City**:
 - ID : int(11) (Primary Key)
 - Name : char(35)
 - CountryCode : char(3)
 - District : char(20)
 - Population : int(11)
- world.Country**:
 - Code : char(3) (Primary Key)
 - Name : char(52)
 - Continent : enum('Asia','Europe','North America','Africa','Oceania','Antarctica','South America')
 - Region : char(26)
 - SurfaceArea : float(10,2)
 - IndepYear : smallint(6)
 - Population : int(11)
 - LifeExpectancy : float(3,1)
 - GNP : float(10,2)
 - GNPOld : float(10,2)
 - LocalName : char(45)
 - GovernmentForm : char(45)
 - HeadOfState : char(60)
 - Capital : int(11)
 - Code2 : char(2)
- world.CountryLanguage**:
 - CountryCode : char(3) (Primary Key)
 - Language : char(30)
 - IsOfficial : enum('T','F')
 - Percentage : float(4,1)

Green arrows indicate foreign key relationships: one from 'City.CountryCode' to 'Country.Code', and another from 'CountryLanguage.CountryCode' to 'Country.Code'.

```
$ psql -d postgres
psql (10.4, server 9.6.9)
Type "help" for help.
```

```
postgres=#
```

Databases
aren't as **flexible**
as spreadsheets.

Databases are good at
storing and **querying** data.

But that's it.

Spreadsheets and
databases have
complementary
skillsets.

So, what do we
do about it?

How to **Solve**

Your Spreadsheet

Problem

- 1. Identify the use case.**
- 2. Stop the spread.**
- 3. Backfill.**

1. Identify the use case.

2. Stop the spread.

3. Backfill.

Every spreadsheet
solves a **problem**.

What is that problem?

What's the **business need**?

How much data is there?

How fast does it change?

How frequent are additions?

1. Identify the use case.

2. Stop the spread.

3. Backfill.

Give new data a
structured home.



Ragic!

No custom apps.

At least at first.

Optimize for
Speed

1. Identify the use case.

2. Stop the spread.

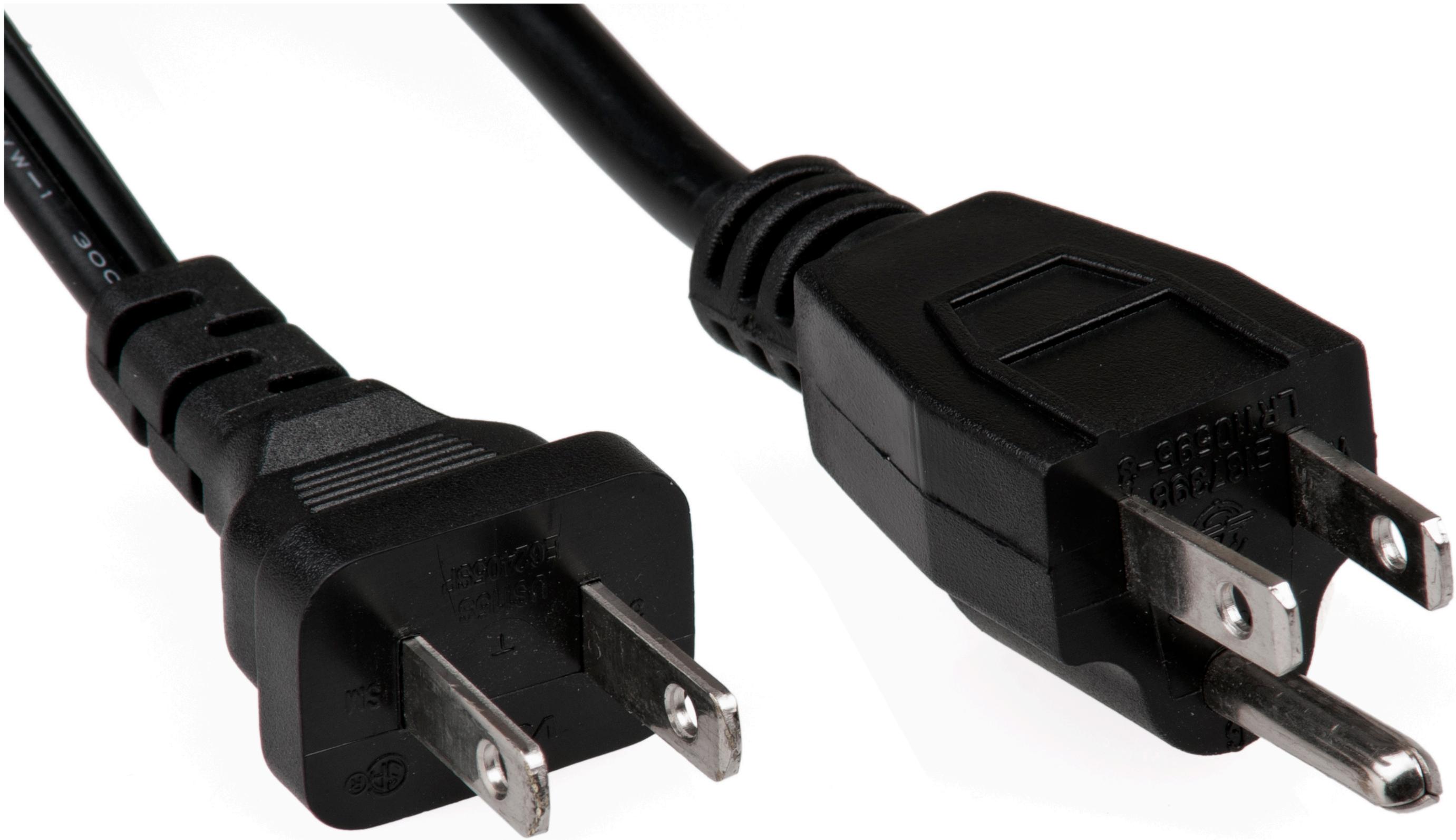
3. Backfill.

It's time for some

Data Wrangling.

(yee-haw 🤨)

Writing one-off
scripts is *sometimes*
the best option.



Sample 1 - First 500KB | 11 Columns | 2,980 Rows | 2 Data Types | Grid | Columns: All | Rows: All | Transformed - 2 Columns | Transformed - 2,979 Rows | Filter in grid

Source	Preview																				
<table border="1"> <thead> <tr> <th>ABC</th> <th>ID</th> <th>ABC</th> <th>column3</th> <th>ABC</th> <th>column4</th> <th>ABC</th> <th>column1</th> <th>ABC</th> <th>column5</th> <th>ABC</th> <th>column6</th> <th>ABC</th> </tr> </thead> </table>	ABC	ID	ABC	column3	ABC	column4	ABC	column1	ABC	column5	ABC	column6	ABC	<table border="1"> <thead> <tr> <th>ABC</th> <th>column1</th> <th>ABC</th> <th>column5</th> <th>ABC</th> <th>column6</th> <th>ABC</th> </tr> </thead> </table>	ABC	column1	ABC	column5	ABC	column6	ABC
ABC	ID	ABC	column3	ABC	column4	ABC	column1	ABC	column5	ABC	column6	ABC									
ABC	column1	ABC	column5	ABC	column6	ABC															
2,839 Categories	258 Categories	101 Categories	1 Category	2,475 Categories	2,475 Categories	2,959 Categories															
customer_id	first_name	last_name		SSN	credit_card	address															
"4abe6b808c96e647239677f2a9f247fd"	Julian	"Russell"	"	"451-59-0366"	"4516009576471550"	"2166·Cedar·Lane															
"d8983c31f0c1031ca2837f42852fbf24"	Nathan	"Davis"	"	"308-61-6226"	"4407614812304060"	"7475·Madison·St															
"f8ebc4a9d5c03b7e2f934019fc10e9d4"	Elijah	"Wright"	"	"593-19-3579"	"5584472636741872"	"48533·2nd·Stree															
"7d9c5c49ad12e8233c558dd88ed3c143"	Cole	"Thomas"	"	"177-74-6463"	"4257017589440200"	"2366·Linden·Str															
"334ae2126c83dffbe28bd8a13d4ae50b"	Andrew	"Green"	"	"557-30-0305"	"5477064333168580"	"4252·10th·ST,·I															
"af67b3a6f43ff02dfedb81ee94cd0bf5"	Adam	"Howard"	"	"076-69-2166"	"5409820014340117"	"278·Orange·Stre															
"12f35d87b7c6e54aec5593e4c19b9824"	Andrew	"Price"	"	"457-96-9416"	"4087559818775316"	"953·Prospect·ST															
"c98623c793c911e68e9c4b7502429983"	Erin	"Barnes"	"	"	"	"370·Dogwood·Dri															
"4435b2c7c3712c154bca9d76427ba72f"	Daniel	"Perez"	"	"329-36-9209"	"5290545373364620"	"6268·Fairway·Dr															
"30a359c8d57c68de61eb5be6128d8d37"	Blake	"Bell"	"	"071-17-4141"	"4477504255299526"	"7709·Holly·Driv															

SUGGESTIONS Cancel Modify Add to Script

Extract on: `""`

ABC	column4	ABC	column1
	last_name		
	"Russell"	"	
	"Davis"	"	

Affects 1 column, 2979 rows | Creates 1 column

Countpattern on: `""`

ABC	column4	#	column1
	last_name	0	
	"Russell"	2	
	"Davis"	2	

Affects 1 column, 2979 rows | Creates 1 column

Extractlist on: `{any}+` delimiter: `""`

ABC	column4	column1
	last_name	["last_name"]
	"Russell"	["", "Russell", ""]
	"Davis"	["", "Davis", ""]

Affects 1 column, all rows | Creates 1 column

<https://www.trifacta.com/start-wrangling/>

Sample 1 - First 500KB | 11 Columns | 2,980 Rows | 2 Data Types | Grid | Columns: All | Rows: All | Transformed - 2 Columns | Transformed - 2,979 Rows | Filter in grid

ABC	ID	ABC	column3	ABC	column4	ABC	column1	ABC	column5	ABC	column6	ABC
2,839 Categories		258 Categories		101 Categories		1 Category		2,475 Categories		2,475 Categories		2,959 Categories
1	customer_id											address
2	"4abe6b808c96e647239677f2a9f2"	"fd"	"Julia"	"Russell"	""	""	"451-9-03"	"451-9-03"	"451-9-03"	"451-9-03"	"451-9-03"	"2166-Cedar-Lane"
3	"d8983c31f0c1031ca2837f42852"	"24"	"Nathan"	"Davis"	""	""	"308-61-62"	"308-61-62"	"308-61-62"	"308-61-62"	"308-61-62"	"7475-Madison-St"
4	"f8ebc4a9d5c03b7e2f934019fc10"	"d4"	"Lilja"	"Wright"	""	""	"593-9-35"	"593-9-35"	"593-9-35"	"593-9-35"	"593-9-35"	"48533-2nd-Street"
5	"7d9c5c49ad12e8233c558dd88ed3c143"		Cole	"Thomas"	""	""	"177-74-6463"	"177-74-6463"	"177-74-6463"	"177-74-6463"	"177-74-6463"	"2366-Linden-Str"
6	"334ae2126c83dfb28bd8a13d4ae50b"		Andrew	"Green"	""	""	"557-30-0305"	"557-30-0305"	"557-30-0305"	"557-30-0305"	"557-30-0305"	"4252-10th-ST, I"
7	"af67b3a6f43ff02dfedb81ee94cd0bf5"		Adam	"Howard"	""	""	"076-69-2166"	"076-69-2166"	"076-69-2166"	"076-69-2166"	"076-69-2166"	"278-Orange-Stre"
8	"12f35d87b7c6e54aec5593e4c19b9824"		Andrew	"Price"	""	""	"457-96-9416"	"457-96-9416"	"457-96-9416"	"457-96-9416"	"457-96-9416"	"953-Prospect-ST"
9	"c98623c793c911e68e9c4b75024278"		Erin	"Barnes"	""	""	""	""	""	""	""	"370-Dogwood-Dri"
10	"4435b2c7c3712c154bca9d76427ba72f"		Daniel	"Pere"	""	""	"529-920"	"529-920"	"529-920"	"529-920"	"529-920"	"6268-Fairway-Dr"
11	"30a359c8d57c68de61eb5be6128d8d37"		Brian	"Bell"	""	""	"071-7-41"	"071-7-41"	"071-7-41"	"071-7-41"	"071-7-41"	"7709-Holly-Driv"

Infer wrangle "recipe" from high-level actions.

SUGGESTIONS

- Extract** on: `''`
ABC column4 | ABC column1
last_name | "Russell" | "Davis"
Affects 1 column | 2979 rows
- Countpattern** on: `''`
ABC column4 | # | column1
last_name | 0 |
"Russell" | 2 |
"Davis" | 2 |
Affects 1 column | 2979 rows
- Extractlist** on: `{any}+` delimiter: ```
ABC column4 | column1
last_name | ["last_name"]
"Russell" | ["Russell", ""]
"Davis" | ["", "Davis", ""]
Affects 1 column | all rows

<https://www.trifacta.com/start-wrangling/>

**Another Option:
Programming
By Example**

FlashRelate

	A	B	C	D	E	...	R
1		value	year	value	year		Comments
2	Albania	1,000	1950	930	1981		FRA 1
3	Austria	3,139	1951	3,177	1955		FRA 3
4	Belgium	541	1947	601	1950		
5	Bulgaria	2,964	1947	3,259	1958		FRA 1
6	Czech ...	2,416	1950	2,503	1960		NC

...
(a)

	A	B	C	D
1	Albania	1,000	1950	FRA 1
2	Albania	930	1981	FRA 1
...				
5	Austria	3,139	1951	FRA 3
6	Austria	3,177	1955	FRA 3
...				
9	Belgium	541	1947	
10	Belgium	601	1950	

...
(b)

Provide example rows,
synthesize *layout* transformations.

<https://github.com/microsoft/prose>

Foofah

	A	B	C	D	E	F	G
1	Description	9/14/2009	9/15/2009	9/16/2009	9/17/2009	9/18/2009	
2	Item 4	900	0	1800	1800		
3	Item 6					1200	
4	Item 8		1800				
5							

	A	B	C	D
1	Item 4	9/14/2009	900	
2	Item 4	9/15/2009	0	
3	Item 4	9/16/2009	1800	
4	Item 4	9/17/2009	1800	
5	Item 6	9/18/2009	1200	
6	Item 8	9/15/2009	1800	
7				

Provide input/output sample, synthesize *layout* and *syntactic* transformations.

<https://github.com/umich-dbggroup/foofah>

How to **Solve** Your Spreadsheet Problem

1. Identify the use case.
2. Stop the spread.
3. Backfill.

What about
the **future**?

Spreadsheets aren't
going anywhere,
for good reason.

Learn from the
spreadsheet.

Meet the users

where they are.

Thank you.

@alexras

Consulting Inquiries: contact@bitsondisk.com