Some Reminders for a Seamless Online Class…

- Please turn on your video
- Mute yourself (press and hold spacebar when you'd like to talk)
- Don't do anything you wouldn't do in an in-person class
- I will occasionally check the chat for messages if you'd like to share there instead
- Please say your name before you speak

Logistics

- Midterm report due tonight midnight
  - One person per team should submit the report, mentioning names of others
- Programming homework due tonight midnight
  - If you discussed with/worked together with others, make sure you mention it in your submission
- Programming homework 2 will be out soon
- We are a bit behind (which is OK!), so we may push our last lecture back by a week, and/or have you record your presentations
- Other considerations
  - Grading will be lenient. It will not be on a curve. We're all going through a lot. I don't want to cause you any additional stress.
  - Class participation has gotten a lot harder to quantify, plus many of you are not in ideal setups for participation (poor internet, travel to home etc.) So I will award everyone complete class participation points, while providing a bonus (if it becomes helpful) for those who have participated a lot thus far.

Recap

- Data-savviness is the future!
- "Classical" relational databases
  - Notion of a DBMS
  - The relational data model and algebra: bags and sets
  - SQL Queries, Modifications, DDL
  - Database Design
  - Views, constraints, triggers, and indexes
  - Query processing & optimization
  - Transactions
- Non-classical data systems
  - Semi-structured data and document stores
  - Unstructured data and search engines
  - Next: Cell-structured data and spreadsheets

So far…

- After relational/structured data, we've studied
  - Unstructured data, which is essentially text-based, with no schema at all
  - Semi-structured data, where the schema is nested, flexible, and non-atomic…
- We're now going to look at yet another way to relax requirements from a database

Classical Database Assumption II

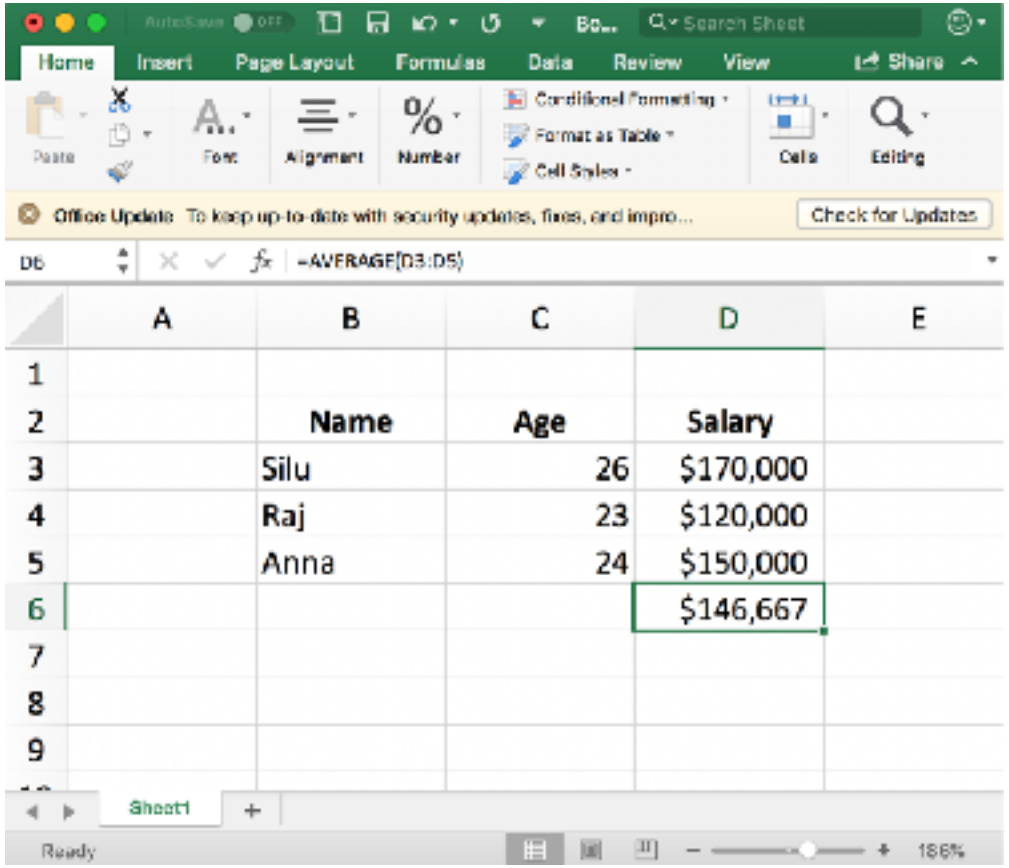
- *Data systems should manage data in relations that can only be accessed through queries, that are unordered, and have a well-defined schema, with queries that operate on relations as a whole and kept separate from the data.*

- We'll consider cell-structured data — the basis of spreadsheets, where these assumptions are relaxed.

- Let's introduced spreadsheets and cell-structured data management first before we revisit these assumptions.

Q: Who has used spreadsheets?

- And what for?

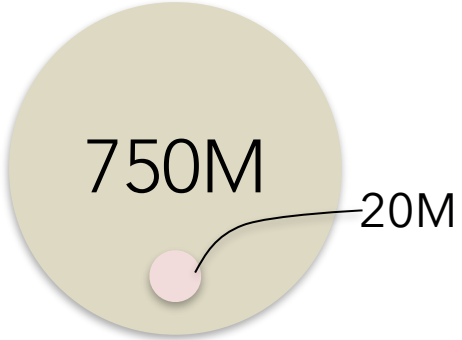


Basic Demo of Spreadsheets

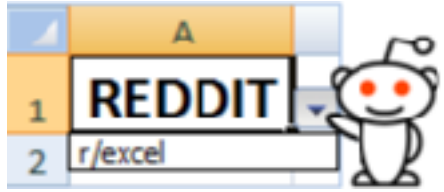
- Features:
  - Cells contain data or formulae
  - Rows and columns are equivalent
  - Cells are referenced by position
  - Formulae can reference data or other formulae
  - “Dragging down” to create a collection of related formulae
  - Direct manipulation

The most popular data management tool!

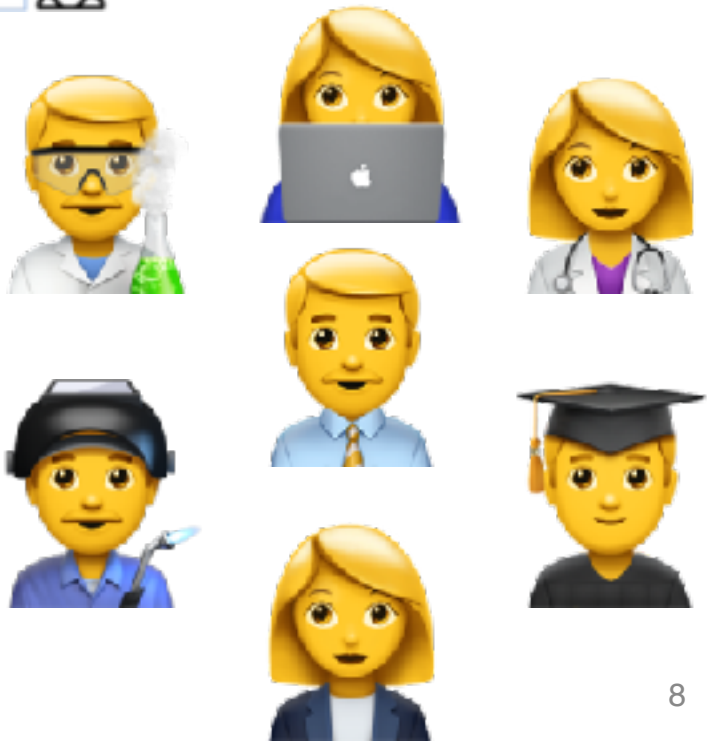
- 10% of the word uses spreadsheets (750 M)
  - Programmers a small fraction (20M)



- Use cases from /r/Excel [Mack, …, P., CHI'18]

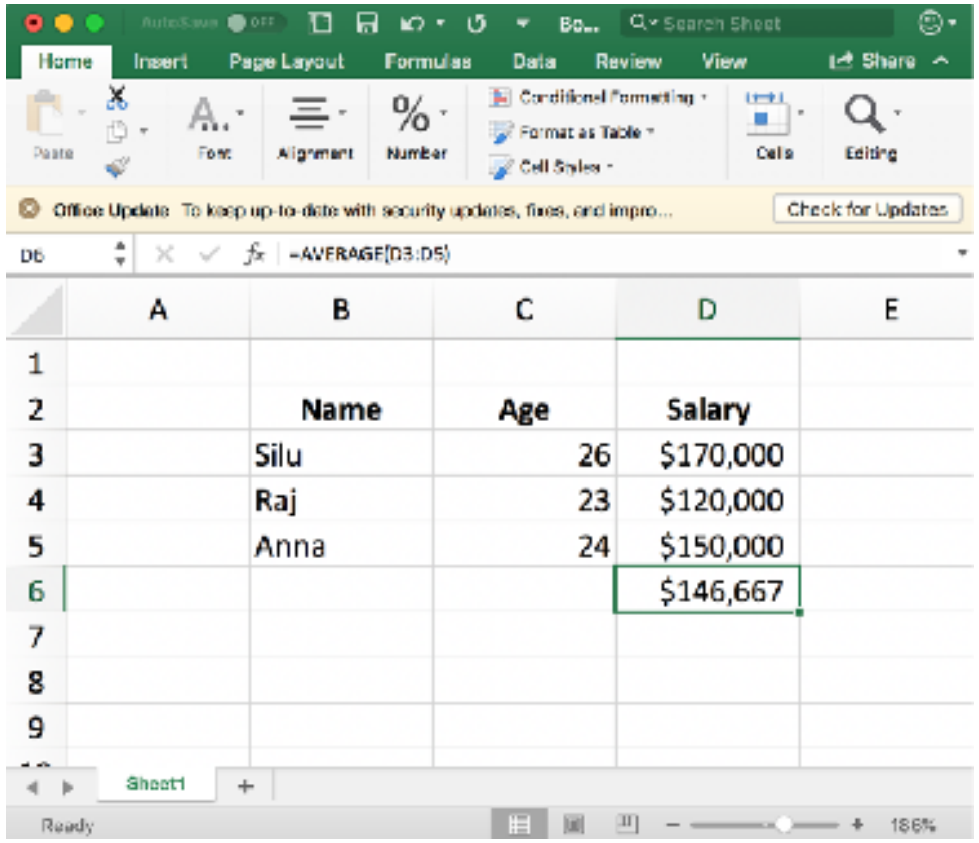


  - Professional:
    - stock tracking
    - finance data
    - inventory tracking
    - real-estate & manuf.
    - scientific exp. data
    - accounting info
    - patient info
  - Personal:
    - health & quantified self
    - sports
    - personal finance
    - …



Q: Why are spreadsheets so popular?

- Easy-to-use and flexible
- Can get started immediately
- Easy to see what is going on and get feedback
- Comes bundled together with most office software
- "Export to excel"



HCI Literature on Spreadsheet Popularity

- useful due to a "table-oriented layout" & "computation without programming" [Nardi & Miller HPL '90]
- "facilitates collaborative work", allowing sharing of knowledge [Nardi & Miller CSCW '90]
- "direct manipulation" capabilities for edits, with immediate feedback [Shneiderman '83]
  - In fact, spreadsheets are heralded as an example of a direct manipulation interface…
  - Q: what is direct manipulation?

Direct Manipulation [Shneiderman'97]

- Direct manipulation interfaces have three properties:
  - Continuous representations of the objects and actions of interest;
  - Physical actions instead of complex syntax;
  - Rapid, incremental, reversible operations whose effect on the object of interest is immediately visible
- Many examples of direct manipulation in daily life!
  - Driving a car via a steering wheel
  - Q: others?
    - Document/word editing
    - Smartphone screens

Benefits of Direct Manipulation

- Shneiderman claims the following benefits of direct manipulation:
  - *Novices can learn basic functionality quickly, especially after demonstration*
  - *Experts can work rapidly to carry out a wide range of activities, including defining new features and functions*
  - *Knowledgeable intermediate users can retain operational concepts*
  - *Users can see the effects of their actions, and can change them if needed*
  - *Users experience less anxiety because the system is comprehensible and actions can be reversed*
  - *Users gain confidence and mastery because they are initiators of actions, they feel in control, and system responses are predictable*

Direct Manipulation [Shneiderman'97]

- Direct manipulation interfaces have three properties:
  - Continuous representations of the objects and actions of interest;
  - Physical actions instead of complex syntax;
  - Rapid, incremental, reversible operations whose effect on the object of interest is immediately visible
- Many examples of direct manipulation in daily life!
- Q: Why are relational databases NOT direct manipulation interfaces?
- Q: What aspects of spreadsheets make them direct manipulation interfaces? What aspects don't?

Downsides of Direct Manipulation

- Not all is great with direct manipulation, however…
- The main downside comes from the size of data being manipulated
  - If you want to add a new column for 1000s of rows
  - Scrolling through 1000s of rows to find what you want
- One could argue that it may be easier to write code (or a SQL query) to do what you want

Spreadsheet Concepts

- A Spreadsheet Workbook comprises many sheets
- Each sheet has *cells* — thus, a spreadsheet is structured around cells
  - Cells contain
    - *values,* e.g., numbers, strings, date/time; or
    - *formulae*, indicated by a "=<Expression>"
      - formula expressions can involve arithmetic +/-
        - e.g., =A1+B1
      - or special *functions*
        - e.g., =AVERAGE(B1, D1)

Spreadsheets: Ad-hoc data layouts, from dense to sparse

	A	B	C	D	E	F
1	snp	chromosome	position	minor	major	
2	rs1208247	1	740857	T	C	
3	rs3094315	1	752566	G	A	
4	rs3131972	1	752721	A	G	
5	rs3115860	1	753406	C	A	
6	rs3131969	1	754182	A	G	
7	rs1048488	1	760912	G	A	
8	rs3115850	1	761147	A	G	
9	rs2286139	1	761732	C	T	
10	rs1256203	1	768448	A	G	
11	rs1212481	1	776546	G	A	
12	rs1080310	1	777122	A	T	
13	rs4040617	1	779322	G	A	
14	rs1080300	1	785089	A	G	
15	rs1124077	1	798959	A	G	
16	rs4970383	1	838555	A	C	
17	rs4475691	1	846808	A	G	
18	rs2860985	1	851190	A	G	
19	rs1806509	1	853954	C	A	
20	rs7537756	1	854250	G	A	
21	rs1330298	1	861808	A	G	
22	rs4040604	1	863124	C	A	
23	rs2340587	1	864938	G	A	
24	rs2857669	1	870645	G	A	
25	rs1110052	1	873558	C	A	
26	rs7523549	1	879317	A	G	
27	rs3748592	1	880238	A	G	
28	rs3748593	1	880890	A	C	
29	rs2272756	1	882033	A	G	
30	rs2340582	1	882803	A	G	
31	rs4246503	1	884815	A	G	
32	rs3748594	1	886384	A	G	
33	rs3748595	1	887560	A	C	
34	rs3748597	1	888659	T	C	
35	rs1330810	1	891945	A	G	
36	rs1330801	1	894573	G	A	

	A	B	C	D	E	F	G	H
1	bob							
2								
3		sally						steven
4				james				
5							jennifer	
6			charles					
7					dan			
8								
9						alice		
10								
11								
12					rick			
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								
33								
34								
35								
36								

Formula Functions

- The arguments to formulae are other cells, which can contain formulae or values.
  - Can be tedious when referring to lots of cells, e.g., cell A1 to A1000
- Shortcuts:
  - rectangular ranges of cells
    - e.g., B1:C3 = B1, B2, B3, C1, C2, C3
  - entire column
    - e.g., F:F = F1, F2, ….
- Standard statistical functions
  - AVERAGE (B1:C3), SUM (F:F), MIN (A1:A100)
  - Relational mapping: Like the aggregation functions in a group by query

Conditional statistical functions

- COUNTIF, AVERAGEIF, SUMIF

- Two arguments: list of cells, followed by a condition
  - e.g., = COUNTIF (F:F, "*HURRICANE*")
    - counts number of values in text field that contain HURRICANE

- Demo: three variants of COUNTIF

- Q: What does this map to from a relational database perspective?
  - Relational Mapping: Essentially extending grouping with a WHERE

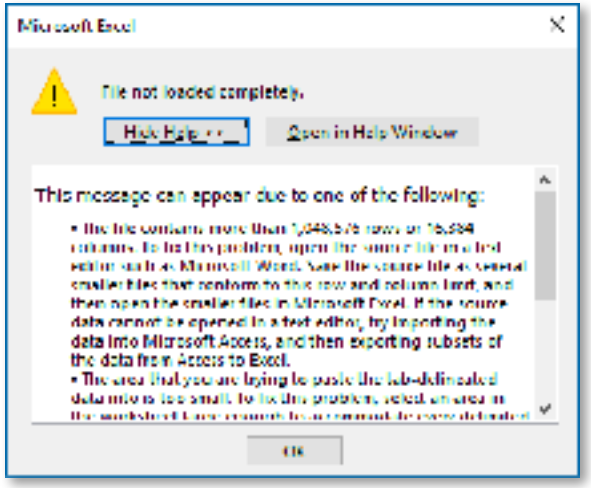
Lookups

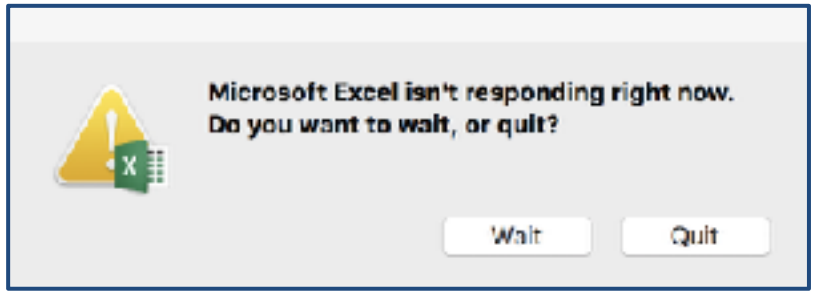
- VLOOKUP or Value Lookup
  - VLOOKUP (value v, tabular range R, col index i, approximate = FALSE)
  - Look for value v in first column of R, if matched, fetch the value in the ith column of R on the same row, and return it

- Demo: VLOOKUP of states

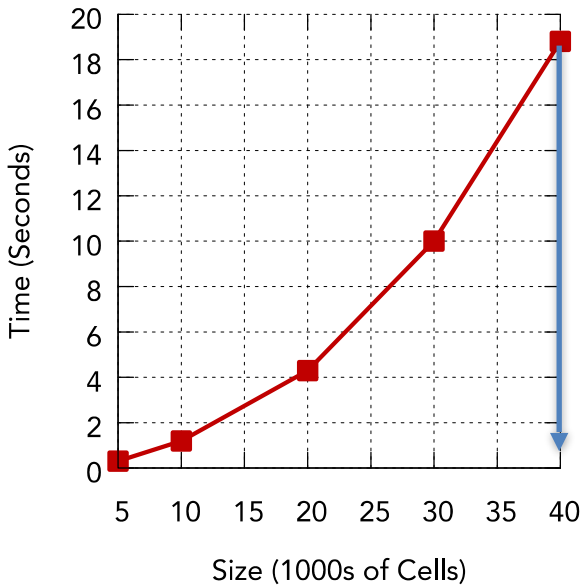
- Q: What does this remind you of from a relational perspective?
  - Relational Mapping: Like a foreign key lookup as part of join
    - Can't do full joins!

Aside: Spreadsheets are slow!





- Scalability: Can't handle large datasets > 1M rows
- Interactivity: Changes cause delays, crashes
  - More at [Rahman, …, P., SIGMOD'20]



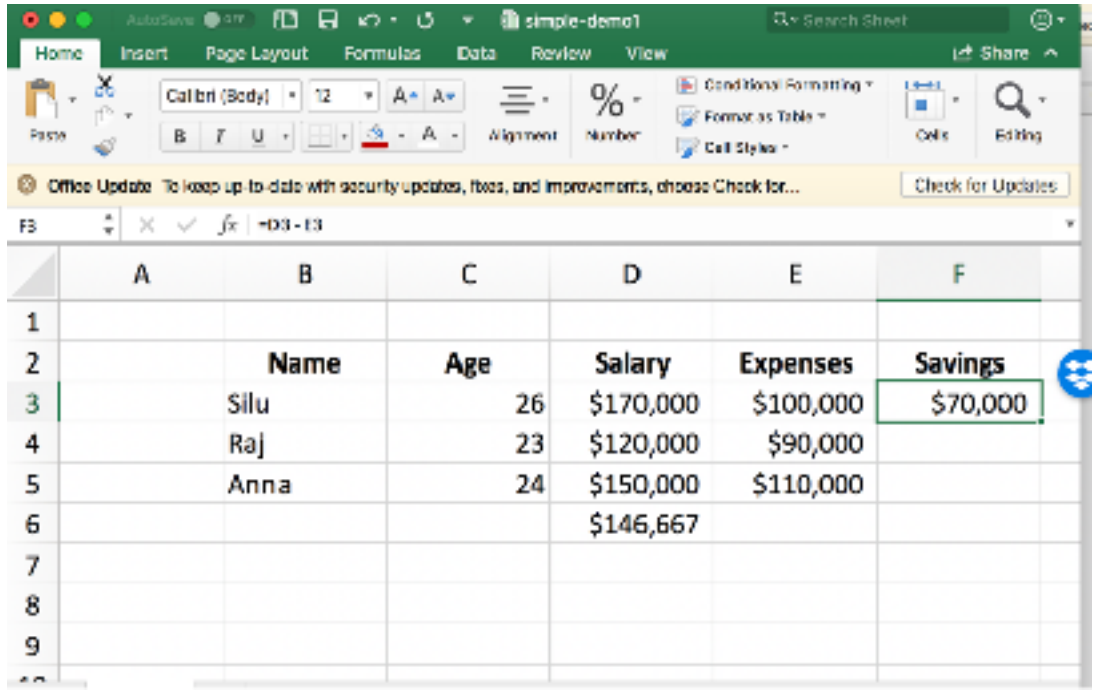
VLOOKUP Times:
Takes minutes on 100k!

Aside: Why are Spreadsheets Slow? [Rahman, …, P., SIGMOD'20]

- They perform limited query optimization, preferring to execute each formula as written
- Their optimization doesn't share computation:
  - Each VLOOKUP is executed separately, rather than treating it together as a "join"
- They don't use any indexes, even for find-and-replace
- They recompute all formulae from scratch even if there are small changes

Aside 2: Relative and Absolute Referencing

- By default, spreadsheets do relative referencing
  - So if F3 has a formula "= D3 - E3", then this means:
    - subtract the cell one step to left from the cell two steps to the left, and place the value in the current cell
  - So, when we "drag" F3 down to F4, … , we are implicitly copying the same formula
- Sometimes, we may want to keep referencing the same absolute cell
- Suppose we want to create a new column G, with savings after 1 year, where the savings after 1 year is simply the current savings (in F) multiplied by a growth rate in cell H1, then we say:
  - G3 = F3 * H1



	A	B	C	D	E	F
1						
2		Name	Age	Salary	Expenses	Savings
3		Silu	26	$170,000	$100,000	$70,000
4		Raj	23	$120,000	$90,000	
5		Anna	24	$150,000	$110,000	
6				$146,667		
7						
8						
9						

So far, we've looked at cell at a time functions…

- That is, these functions take in a collection of cells and other arguments, and return **a single value**
  - Which is why you can't do joins easily in a spreadsheet — since most joins are "multiplicative", you can't anticipate how large a join result will be.

- We'll now see some other functionality that isn't cell-at-a-time

Sorting and Filtering

- Relational mapping:
  - Sorting = ORDER BY in a relational context;
  - Filtering = WHERE in a relational context

- Sorting orders the spreadsheet based on some criteria
- Filtering retains rows that match certain criteria

- Demo of filtering

Pivot Tables

- Pivot tables are one of the most interesting & unique capabilities that spreadsheets support

- It's essentially a more group-by style aggregation, but with the ability to move data to the schema (hence the term pivot)

- Demo: Let's consider some examples…
  - Sum of disasters of various types across states
    - Like a group-by aggregation
  - Sum of disasters by year and state
    - Year is moved to the schema

Conditional Formatting

- Conditional formatting allows you to specify the formatting for cells depending on conditions
  - e.g., color the cells red if they are greater than a certain value, or among the top 10% of the values in that column

- Demo: Color the "Sum of disasters" column based on whether it is equal to 2

Summary of Spreadsheet Functionality

- Cells & Formulae, with shortcuts to reference rectangular areas
- Statistical and conditional statistical functions
- Lookup functions
- Sorting and filtering
- Pivot tables
- Conditional formatting
- Other functionality:
  - subtotals, charting capabilities, macros (a full-fledged programming language)

Downsides of Spreadsheets

- We've already covered one in Aside 1: Spreadsheets are very slow
- Q: Any other downsides?
- The second major downside essentially from the positional aspects of spreadsheets, which encourages **mistakes**, for several reasons
    - Remember Aside 2? Formulae can be confusing
    - Copy pasting of formulae often leads to spreadsheet mistakes percolating
    - Partly because of the reliance on position (which is brittle) rather than underlying intent
    - Formulae are hidden away
    - Users of spreadsheets end up being sloppy because it is so easy to edit
    - Apparently, more than half of spreadsheets contain at least one error!
- Spreadsheets have limited functionality: can't support joins, cell-at-a-time
- Spreadsheets make it hard to manipulate large volumes of data directly

Returning to the Spreadsheet-Database Comparison

- Q: How do spreadsheets and databases differ?
  - Talk about data model, modifications/updates, queries

	Aspect	Databases	Spreadsheets
I. Organization	structure	rigid	flexible
	presentation	unordered, uniform	ordered, ad-hoc
II. Manipulation	modality	relation at a time	cells, using position
	granularity	predicate-based	direct edits + add/delete row/ columns
III. Computation	modality	external	in-situ, with data
	granularity	queries on relations	formulae on cells

29

When should you use Spreadsheets?

- A spreadsheet is not just a data management system
  - Unlike databases, spreadsheets are embedded with hundreds or thousands of materialized views in the form of formulae, plus charts as well
- Thus, it contains the data plus the analysis/presentation of the data
  - Emphasized by formatting (e.g., conditional)

- If your dataset is small and ad-hoc ~1000-10000, and the kinds of analysis you want to do to your data is modest and quick-and-dirty (e.g., simple statistics, pivot tables), spreadsheets are a perfectly fine tool
  - Also easy to share your analysis with others and present it all in one tool
  - No need to install and set up your database, set a schema, …
- Another setting would be if you are actively collecting small volumes of data by editing
  - Much easier to edit than a traditional database

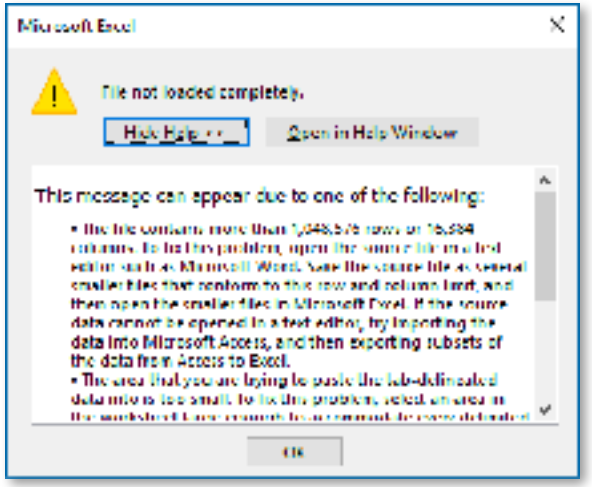
Classical Database Assumption II

- *Data systems should manage data in relations that can only be accessed through queries, that are unordered, and have a well-defined schema, with queries that operate on relations as a whole and kept separate from the data.*

- Spreadsheets violate many of these assumptions
  - Data is ad-hoc and cell-structured, not relational
  - Data can be directly manipulated
  - Data is ordered, and position is central
  - Data doesn't need to have a schema
  - Queries = formulae, operate on collections of cells at a time
  - Queries are embedded as materialized views along with data

Research Plug

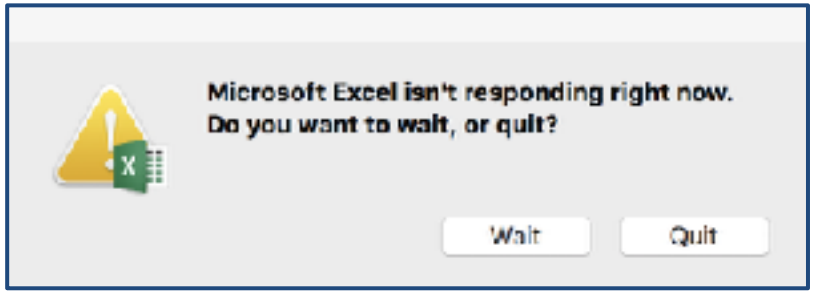
- We're working on a spreadsheet-database hybrid called DataSpread
- More at dataspread.github.io

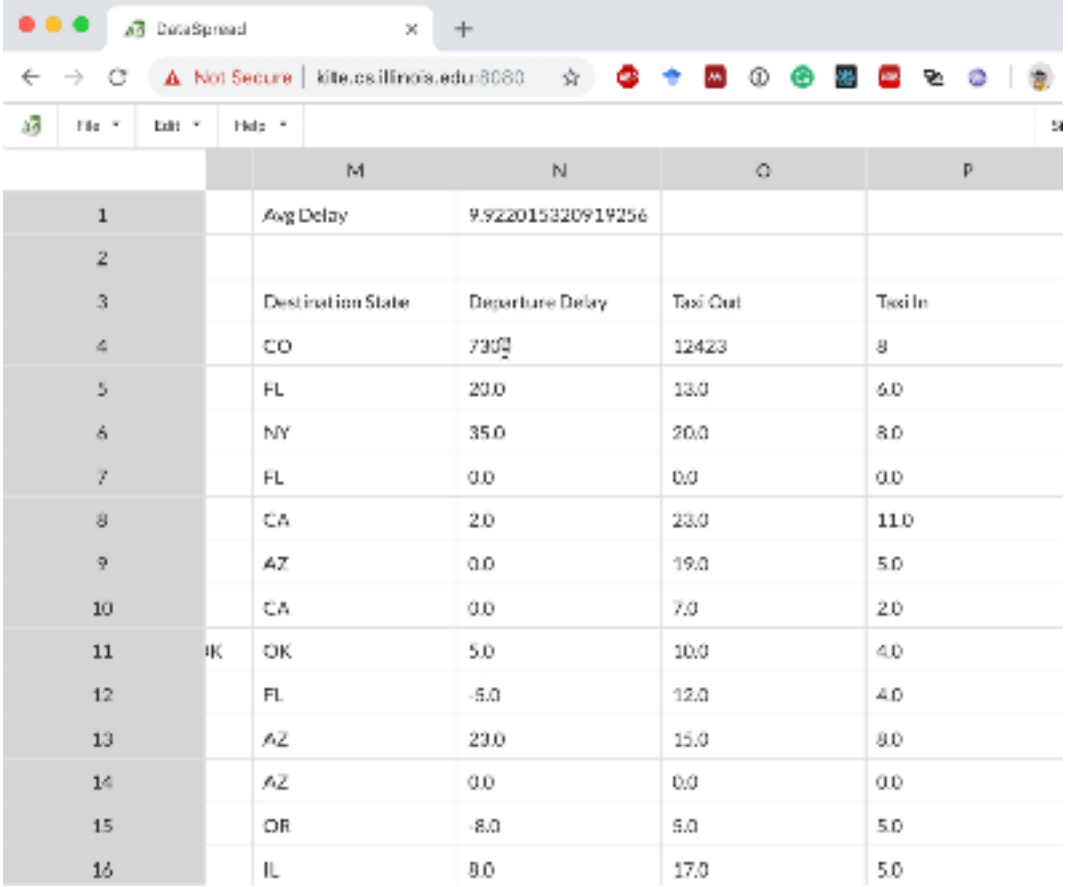


	J	K	L	M	N
1	Origin State	Destination Airport	Destination City	Destination State	Departure Delay
2	MO	LIT	Little Rock- AR	AR	30.0
3	NH	EWR	Newark- NJ	NJ	-4.0
4	NV	DEN	Denver- CO	CO	73.0
5	FL	TPA	Tampa- FLrt	FL	20.0
6	NC	LGA	New York- NY	NY	35.0
7	CA	MIA	Miami- FL	FL	0.0
8	TX	LAX	Los Angeles- CA	CA	2.0
9	UT	PHX	Phoenix- AZ	AZ	0.0
10	CA	SMF	Sacramento- CA	CA	0.0
11	NM	OKC	Oklahoma City- OK	OK	5.0
12	TX	MIA	Miami- FL	FL	-5.0
13	TX	PHX	Phoenix- AZ	AZ	23.0
14	AZ	FLG	Flagstaff- AZ	AZ	0.0
15	OR	PDX	Portland- OR	OR	-8.0
16	NC	ORD	Chicago- IL	IL	8.0
17	TX	DFW	Dallas/Fort- TX	TX	29.0
18	MN	GTF	Great Falls- MT	MT	-3.0
19	MI	ALB	Albany- NY	NY	3.0
20	GA	EWR	Newark- NJ	NJ	3.0
21	FL	ATL	Atlanta- GA	GA	0.0

Research Plug (contd.)

- Asynchronous formula computation





	M	N	O	P
1	Avg Delay	9.922015320919256		
2				
3	Destination State	Departure Delay	Taxi Out	Taxi In
4	CO	730	12423	8
5	FL	20.0	13.0	6.0
6	NY	35.0	20.0	8.0
7	FL	0.0	0.0	0.0
8	CA	2.0	23.0	11.0
9	AZ	0.0	19.0	5.0
10	CA	0.0	7.0	2.0
11	OK	5.0	10.0	4.0
12	FL	-5.0	12.0	4.0
13	AZ	23.0	15.0	8.0
14	AZ	0.0	0.0	0.0
15	OR	-8.0	5.0	5.0
16	IL	8.0	17.0	5.0

Research Plug (Contd.)

Navigation via zooming in and out

- "Scrolling and windowing introduce a discontinuity between information displayed at different times & places" [Cockburn et al. '08]

- Spreadsheet users use crutches, such as pen and paper and landmarks to help navigate. [Watts et al. '99]

- Users have difficulty comparing data across screens, requiring copying data over [Nardi & Miller '90]

https://youtu.be/MAK36CB14YI?t=73