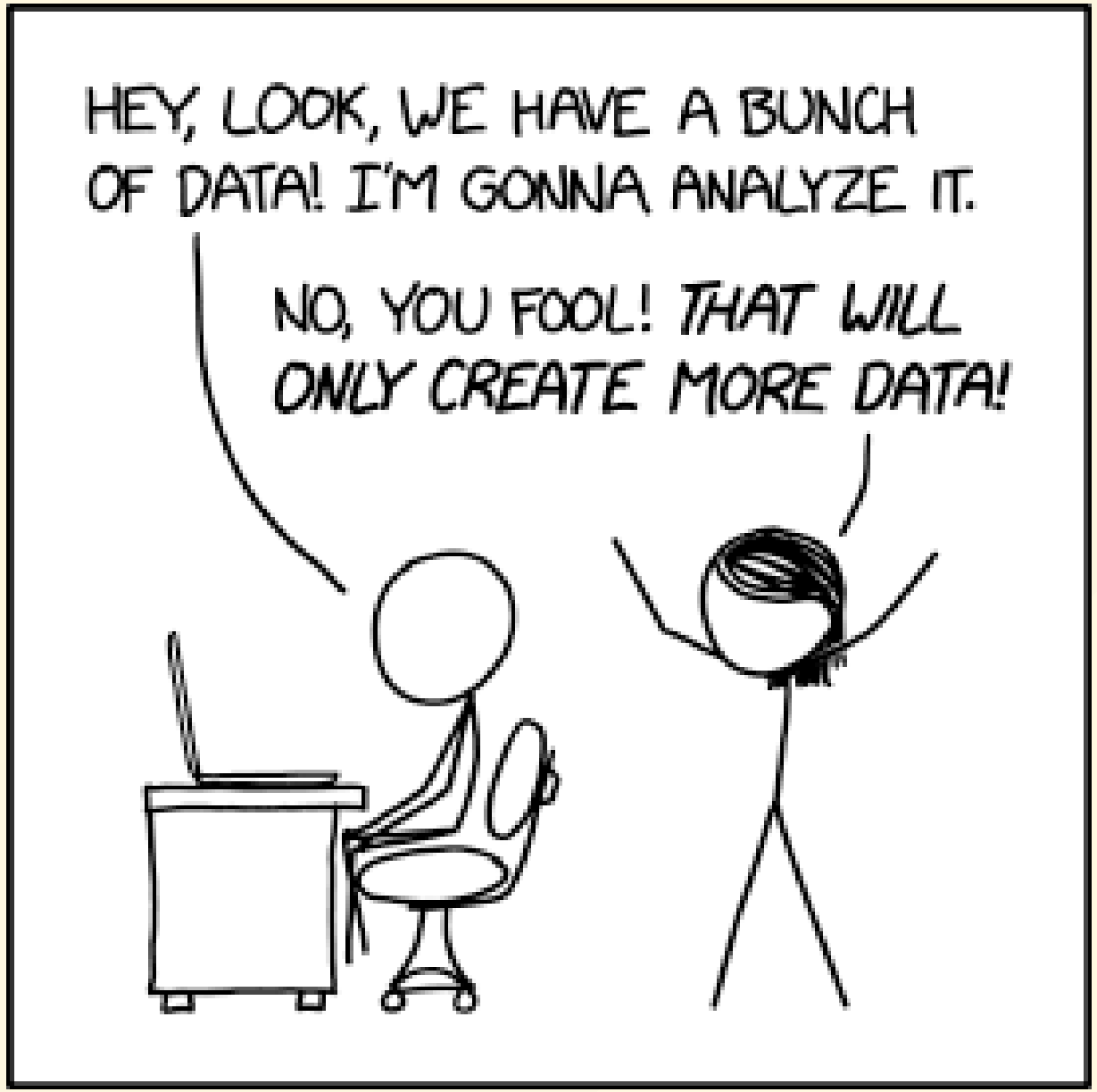
XKCD Comic Analysis

Preston Knepper and Kevin McCall

May 7, 2024

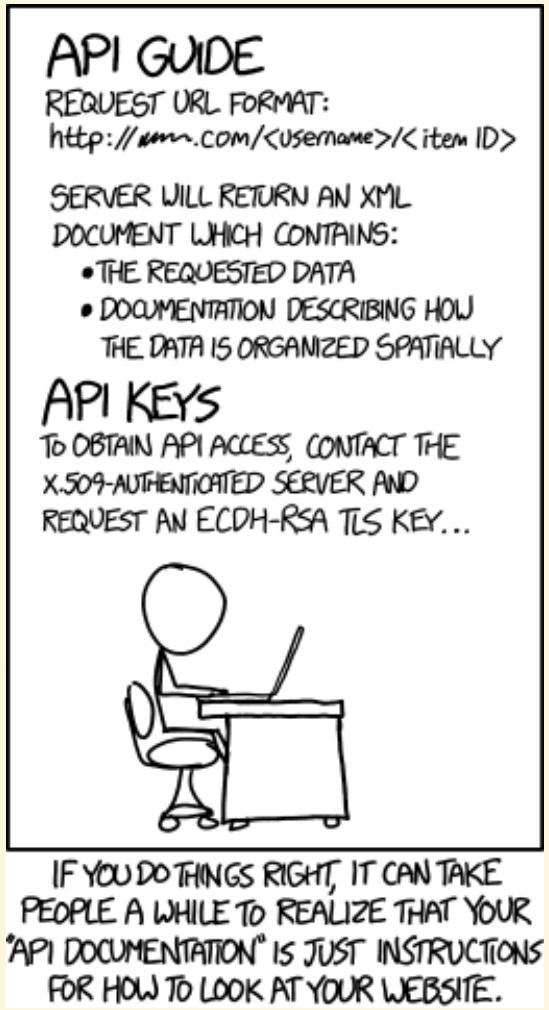


Introduction

In this project, we take a look at xkcd comics across the life of the webcomic to find interesting paternand abnormalities, as well as challenge our abilities to parse and clean data which was never meant to be studied.

Background

XKCD refers to itself as "A web-comic of romance, sarcasm, math, and language", taking its roots in nerdy humor typically about technology, math, physics and science. XKCD is popular in the world covered by those topics. The comic i posted regularly on every Monday, Wednesday, and Saturday.

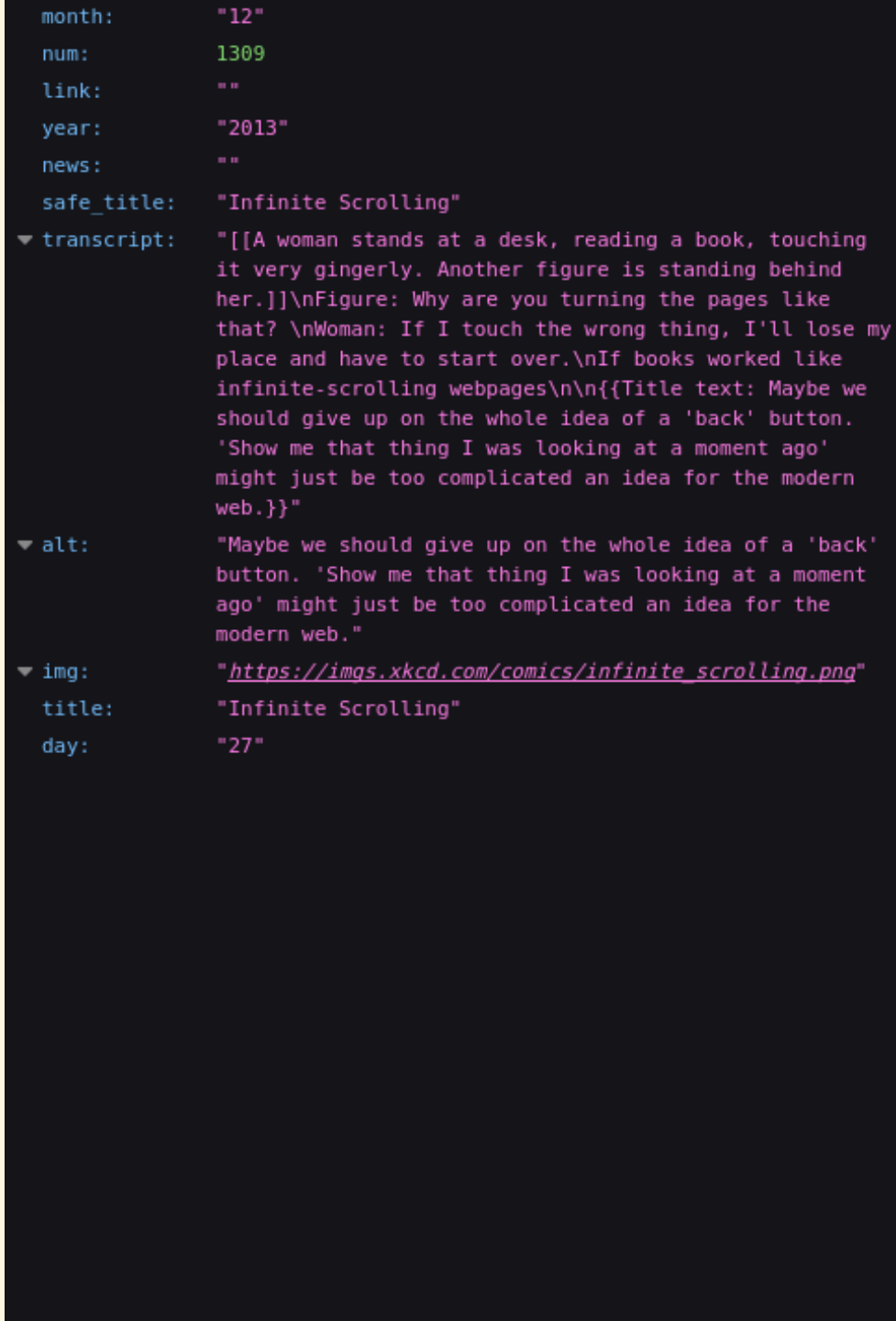


Gathering the Data

- Monroe offers a JSON API for his comic.
- It should be easy to parse comics into a .csv.
- In spite of this, the data was pretty tricky to form into a .csv.

A look at the JSON

- month: The month of the the year this comic was posted (as a number).
- num: The comic number.
- link: An external link on image click.
- year: The year the comic was posted.
- news: Comic news to display to the reader.
- safe_titl: A version of the title safe for all browsers.
- transcript: The bane of this project.
- alt: The alt text (mouseover) of the comic.
- img: A link to the comic image.
- title: The title of the comic.
- day: The day of the month the comic was posted.



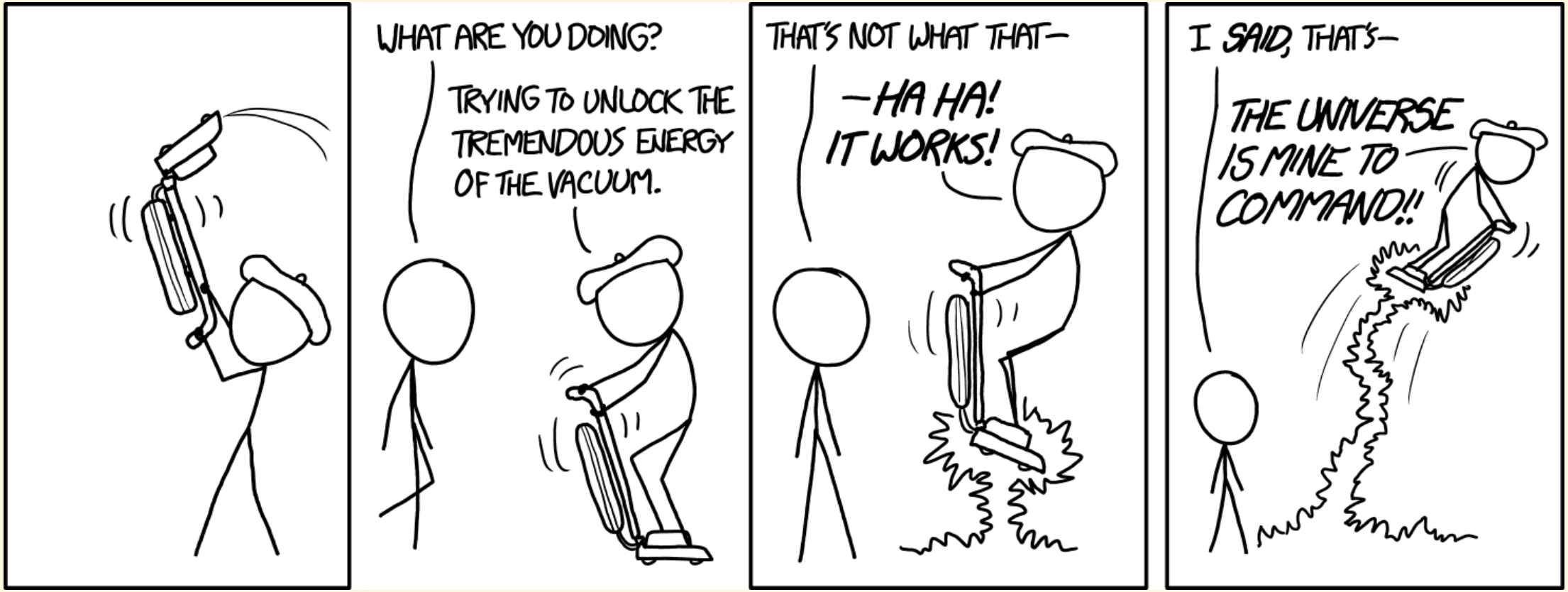
The transcript and why it's terrible

- The transcript is not parse friendly, it includes many commas, quotes, and newlines.
  - When R reads a csv (through both read*csv and read.csv), it will parse *_Every** comma as a new column.
  - When R reads a quote, it will try to end it's current value, throwing an error.
  - When R reads a newline '\n' character, it will immediatly create a new entry in the dataframe.

Parsing the Data

Analyzing the problem

Here is an example comic to parse text from:



[[Beret Guy is heaving a vacuum cleaner overhead]]\n\nCueball: What are you doing?\n[[Beret guy sets the vacuum cleaner on the ground as one would normally use it, but is standing atop the engine and desperately manhandling the grip.]]\nBeret: Trying to unlock the tremendous energy of the vacuum.\n\n[[Beret guy rises off the ground, hovering on the vacuum cleaner]]\nCueball: That's not what that-\nBeret: Ha ha! It works!\n<<BWAROUUGUMHGHHGMMM>>\n\nCueball: I said, that's-\nBeret: The univere is mine to command!\n<<GLHDFKUOUAHUUUUGUUUAAAUUAUUUUUUUGGGGGH>>\n[[Beret guy rockets away on plume of Clean Energy]]\n\n{{Title text: Do you think you could actually clean the living room at some point, though?}}

xkcd comics transcripts are in theatrical format. Each dialog is on its own line and is preceeded by who is saying it. Our strategy to parse this data will be to search for text on a new like and split the character and their speech by a colon.

The easiest way to parse data in this format is to break it down into simpler stages and tackle those one at a time.

First Stage - Cleaning text

First, lets remove the data we do not need

The {{Title Text}} is unecessary since our web scraping script has access to that field already, so we may remove it.

[[Beret Guy is heaving a vacuum cleaner overhead]]\n\nCueball: What are you doing?\n[[Beret guy sets the vacuum cleaner on the ground as one would normally use it, but is standing atop the engine and desperately manhandling the grip.]]\nBeret: Trying to unlock the tremendous energy of the vacuum.\n\n[[Beret guy rises off the ground, hovering on the vacuum cleaner]\nCueball: That's not what that-\nBeret: Ha ha! It works!\n<<BWAROUUGUMHGHHGMMM>>\n\nCueball: I said, that's-\nBeret: The univere is mine to command!\n<<GLHDFKUOUAHUUUUGUUUAAAUUAUUUUUUUGGGGGH>>\n[[Beret guy rockets away on plume of Clean Energy]]\n\n

Second Stage - Extracting Scene information

```
\[\[([^\]]*)\]\]
```

- we extract the

\n\nCueball: What are you doing?\n\nBeret: Trying to unlock the tremendous energy of the vacuum.\n\n\nCueball: That's not what that-\nBeret: Ha ha! It works!\n<<BWAROUUGUMHGHHGMMM>>\n\nCueball: I said, that's-\nBeret: The univere is mine to command!\n<<GLHDFKUOUAHUUUUGUUUAAAUUAUUUUUUUGGGGGH>>\n\n\n

Clean up from Stage 2

Many \n's remain. Furthermore, characters' dialog can span multiple lines.
We may flatten these into a single pair \n by making a call to `str_replace`, replacing `\n+` with a single `\n`.

However, since newlines can be anywhere in a character's dialog, we can't use a simple regex to parse text.

Solution: A complicated Regex

```
(?:\\n)(?!(?:.(?!\\n))+:)
```

- This regex uses *lookaheads*, a powerful regex feature that determines matches without capturing the input.
- Lookaheads are used to match a select a \n where there is another \n in between it and the next : character
- These are removed with a call to `str_remove_all`

\nCueball: What are you doing?\nBeret: Trying to unlock the tremendous energy of the vacuum.\nCueball: That's not what that-\nBeret: Ha ha! It works! <<BWAROUUGUMHGHHGMMM>>\nCueball: I said, that's-\nBeret: The univere is mine to command!\n<<GLHDFKUOUAHUUUUGUUUAAAUUAUUUUUUUGGGGGH>>

Stage 3: Parsing the dialog

From our previous stages, we have reduced the burden on ourselves to parse the regex. We can finally extract the remaining 2 parts with this last regex: `(?:\\n)([^:]+):`

- By using this regex, we get the text in between a newline and a colon. This is the name of the speaker
- To gather the dialog, we can delimit the remaining the text by the names, leaving us with what they said!

Cueball

What are you doing?

Beret

Trying to unlock the tremendous energy of the vacuum

Cueball

That's not what that-

Beret

Ha ha! It works!<<BWAROUUGUMHGHHGMMM>>

Cueball

I said, that's-

Beret

The univere is mine to command!

<<GLHDFKUOUAHUUUUGUUUAAAUUAUUUUUUUUGGGGGH>>

Results

Successes

Issues

Inconsistent delimiters

- [] instead of [[]]
- no starting \n
- colons in the text of the comic
- Same character with multiple names

Since the way we parse data is dependent on the ":" symbol, we are at the mercy of Monroe to provide to be consistent in his theatrical format. However, there are exceptions. For example:

HACKERS RECENTLY LEAKED **153 MILLION** ADOBE USER EMAILS, ENCRYPTED PASSWORDS, AND PASSWORD HINTS.

ADOBE ENCRYPTED THE PASSWORDS IMPROPERLY, MISUSING BLOCK-MODE 3DES. THE RESULT IS SOMETHING WONDERFUL:

USER PASSWORD		HINT
4e18acc1ab27b2d6		WEATHER VANE SWORD
4e18acc1ab27b2d6		
4e18acc1ab27b2d6	a0a2876cb1ea1fca	NAME1
8babb6299e06eb6d		DUH
8babb6299e06eb6d	a0a2876cb1ea1fca	
8babb6299e06eb6d	85e9da81a8a78adc	57
4e18acc1ab27b2d6		FAVORITE OF 12 APOSTLES
1ab29ae86da6e5ca	7a2d6a0a2876eb1e	WITH YOUR OWN HAND YOU HAVE DONE ALL THIS
a1f9b2b6299e7a2b	eadec1e6ab797397	SEXY EARLOBES
a1f9b2b6299e7a2b	617ab0277727ad85	BEST TOS EPISODE
397387a7adb0b8ad7	617ab0277727ad85	SUGARLAND
1ab29ae86da6e5ca		NAME + JERSEY #
877ab7889d3862b1		ALPHA
877ab7889d3862b1		
877ab7889d3862b1		
877ab7889d3862b1		OBVIOUS
877ab7889d3862b1		MICHAEL JACKSON
38a7c9279cadeb44	9dca1d79d4dec6d5	
38a7c9279cadeb44	9dca1d79d4dec6d5	HE DID THE MASH, HE DID THE
38a7c9279cadeb44		PURLOINED
a8ae5745a2b7af7a	9dca1d79d4dec6d5	FAV WATER-3 POKEMON

THE GREATEST CROSSWORD PUZZLE IN THE HISTORY OF THE WORLD

Comics with Missing Transcripts