CS 677: Big Data

Spatiotemporal Data

Lecture 16

Today’s Schedule

- Spatiotemporal Data
- Geohash Algorithm

Today’s Schedule

- **Spatiotemporal Data**
- Geohash Algorithm

Spatiotemporal Data

- One of the many sources of big data is ***spatiotemporal*** datasets
- These datasets are multidimensional:
  1. Space (geographic location, x-y coordinate, etc.)
  2. Time (could be years, days, even microseconds)
- Besides space and time, a spatiotemporal data point isn't very useful without additional ***features***:
  - Name, Age, ID
  - Speed, Weight, Direction

Spatiotemporal Data Sources

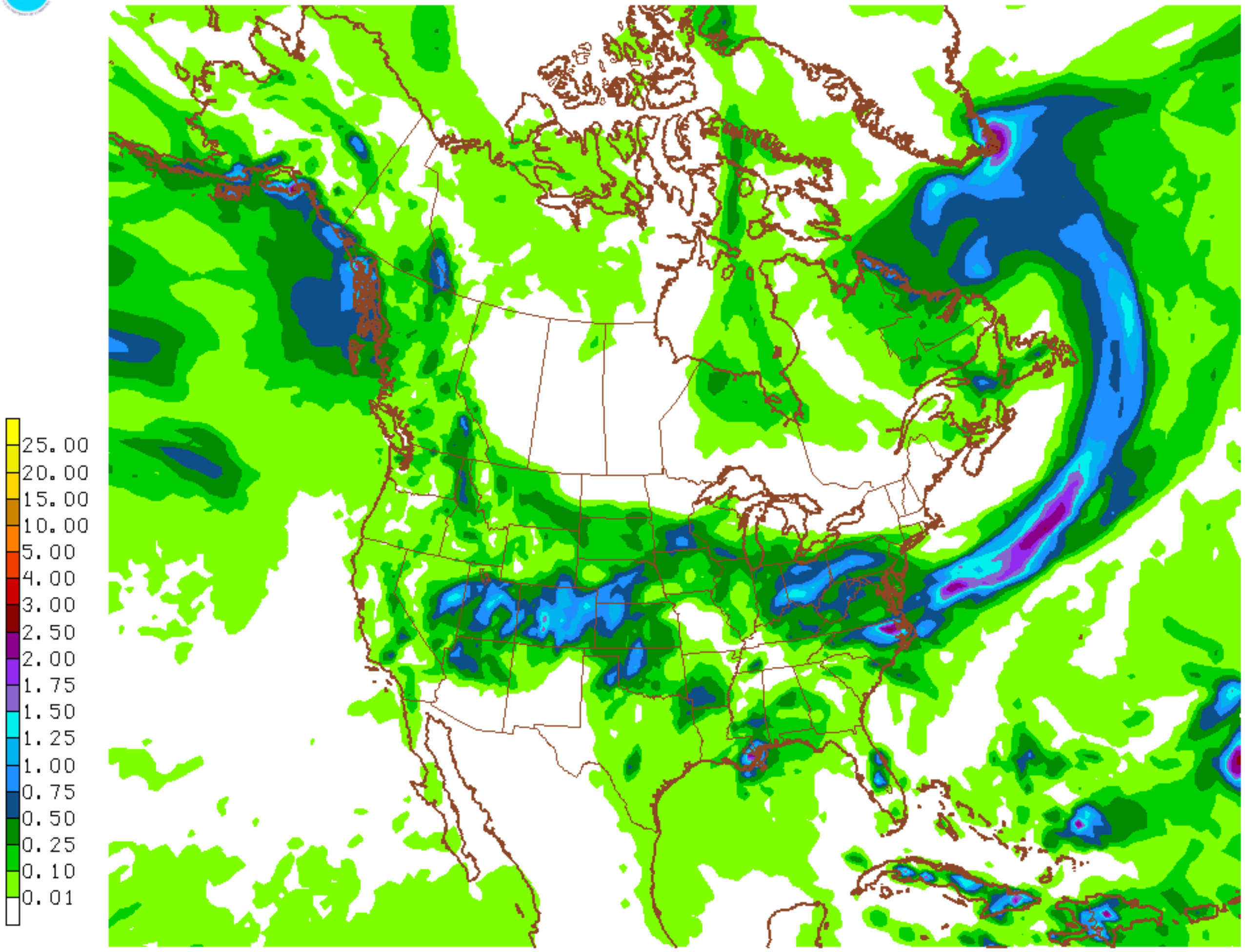
- Geographic information systems
    - Electric usage in a city over time
- Object tracking systems
    - GPS, atomic clocks, speed, direction
- Multiplayer games
    - Player location, attributes
- Networked sensors and radars
    - Temperature sensor with Wi-Fi connectivity
    - Cloud cover or reflectivity readings

P3 Motivation: NCDC Dataset

- Sourced from NOAA
- Some Dimensions/Features:
  - Geospatial: Latitude, Longitude
  - Time Series: Time stamp
  - Temperature
  - Relative Humidity
  - Wind Speed
  - Etc.

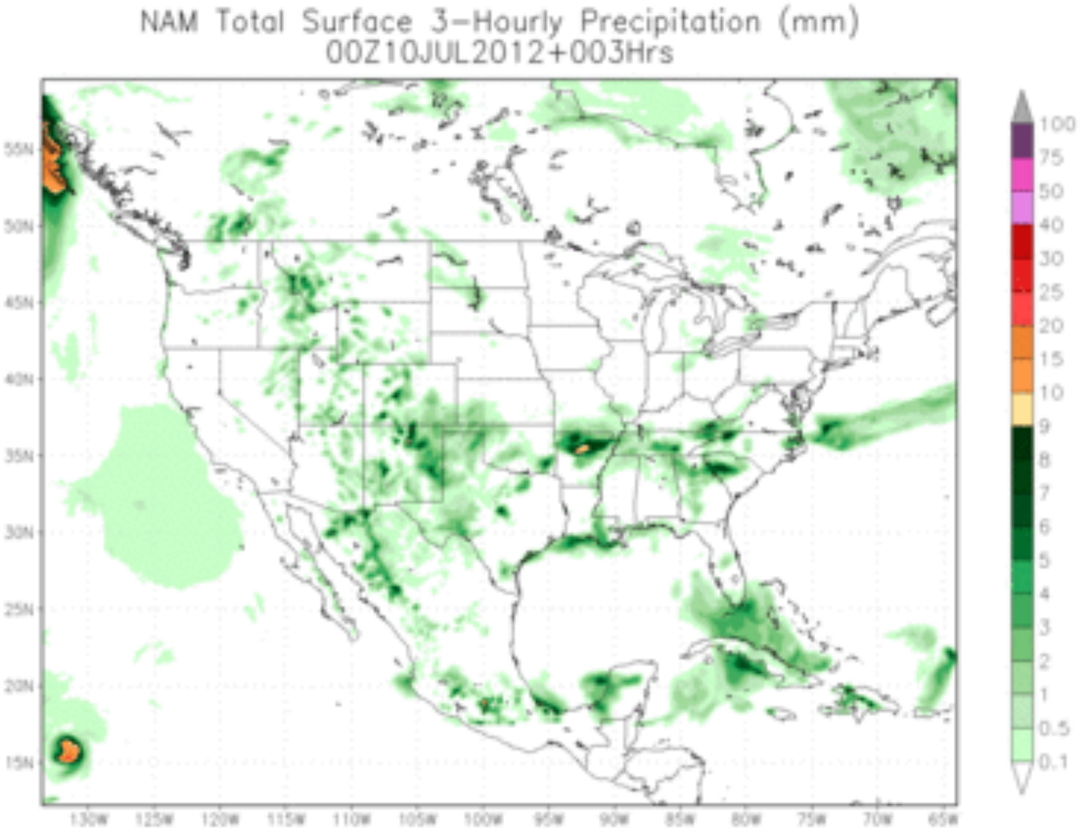
Precipitation Snapshot

04/28/16 00UTC 042HR FCST VALID Fri 04/29/16 18UTC NCEP/NWS/NOAA



160429/1800V042 NAM44 042-HR TOTAL PCPN (IN)

Animation


NAM Total Surface 3-Hourly Precipitation (mm)
00Z10JUL2012+003Hrs
55N
50N
45N
40N
35N
30N
25N
20N
15N
130W
125W
120W
115W
110W
105W
100W
95W
90W
85W
80W
75W
70W
65W
100
75
50
40
30
25
20
15
10
9
8
7
6
5
4
3
2
1
0.5
0.1

Learning More

- See:

  https://www.ncei.noaa.gov/products/weather-climate-models/north-american-mesoscale

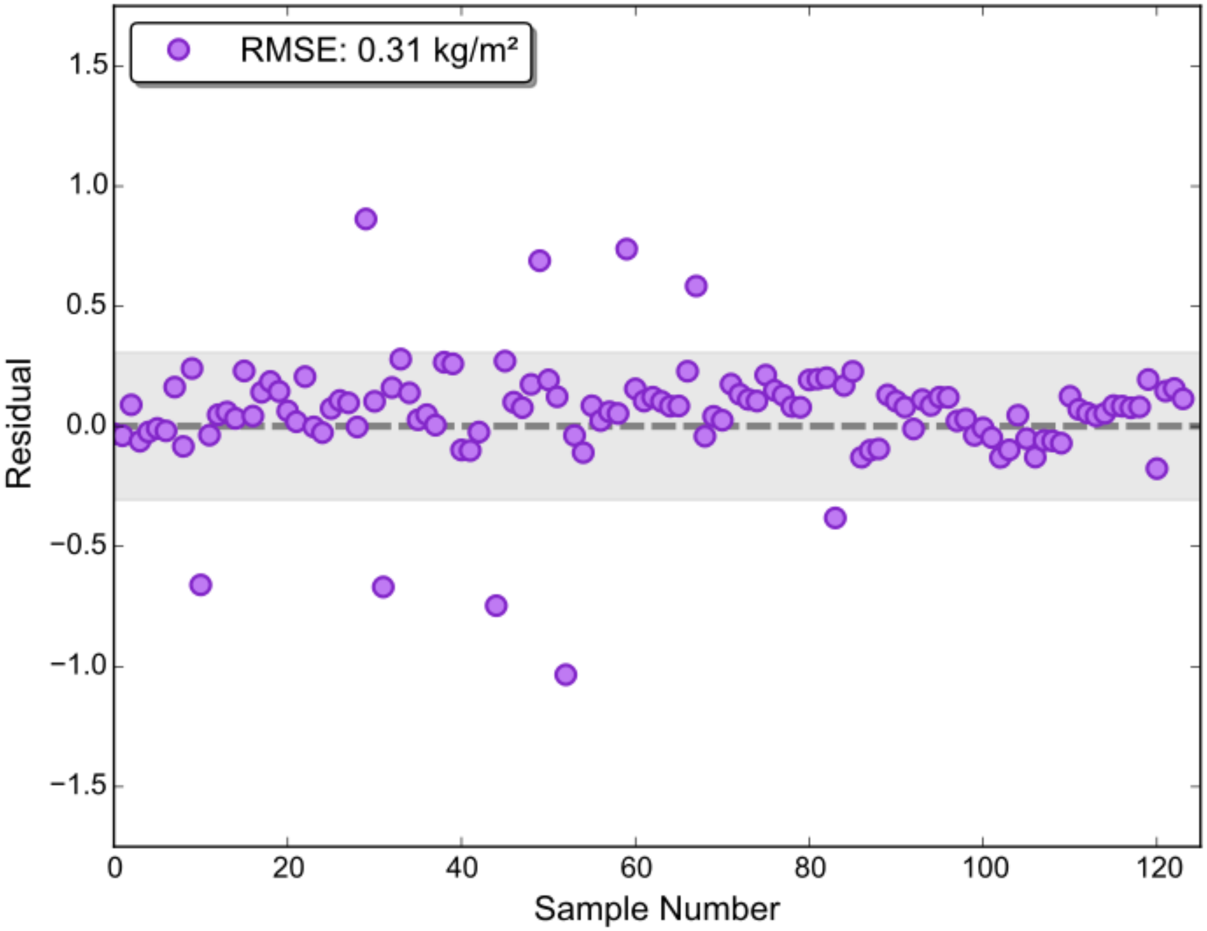
Dataset Applications

- Predicting future weather events or patterns
  - Machine learning
  - Statistics
- Summarizing Information
  - Visualizations
  - Reports
- Exploring relationships between features
  - How does the temperature influence humidity?
  - How does the location influence precipitation?

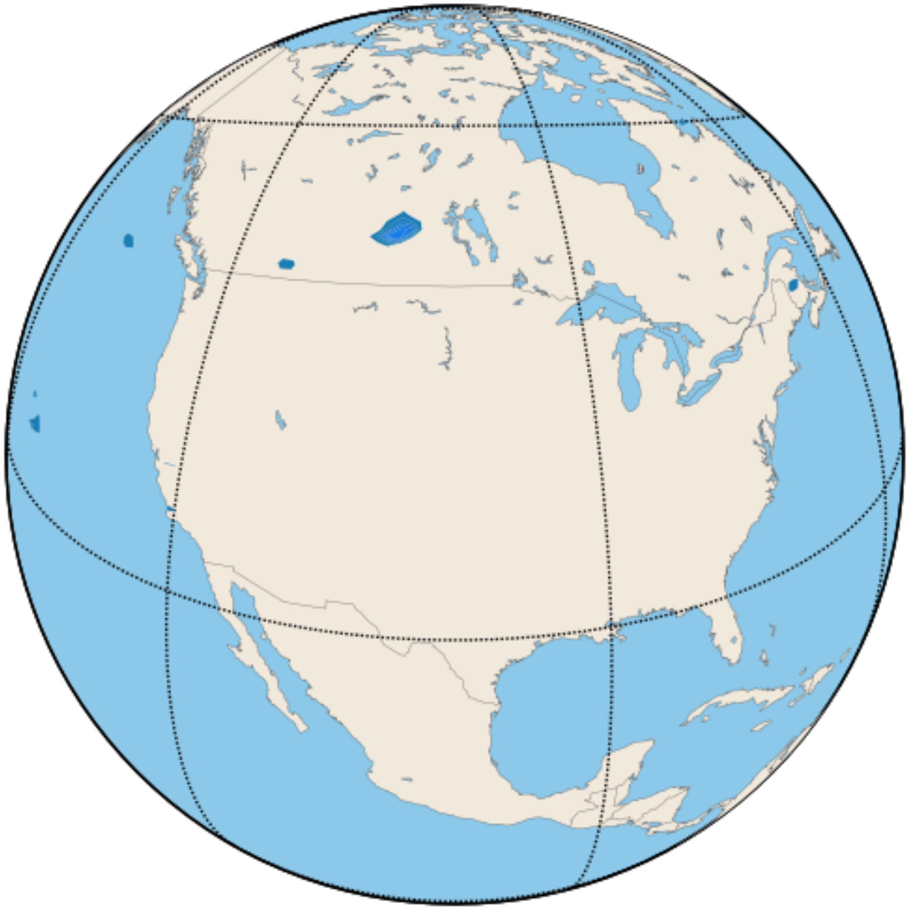
Dataset Specifics

- Each file represents a day+time
  - Contains a reading for each weather station on a grid across the entirety of North America
- Original data goes back to 2006
  - Stored in GRIB format
- I've preprocessed the dataset a bit already
  - Each day/time is represented as a .tdv file
  - Each feature is separated by tabs
  - Contains a header with feature names

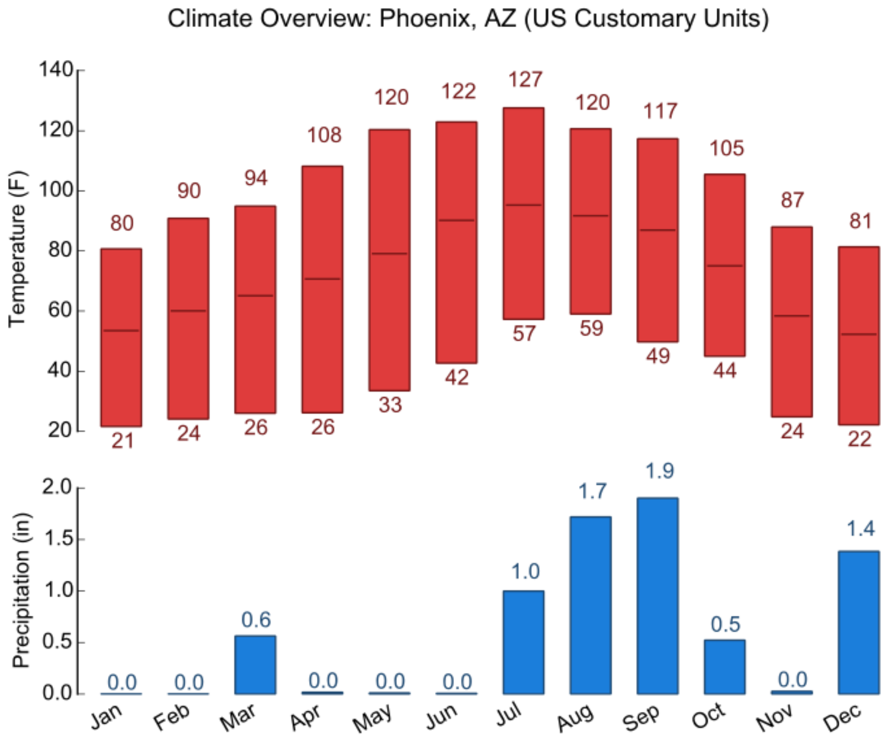
Predicting Rainfall: Wyoming

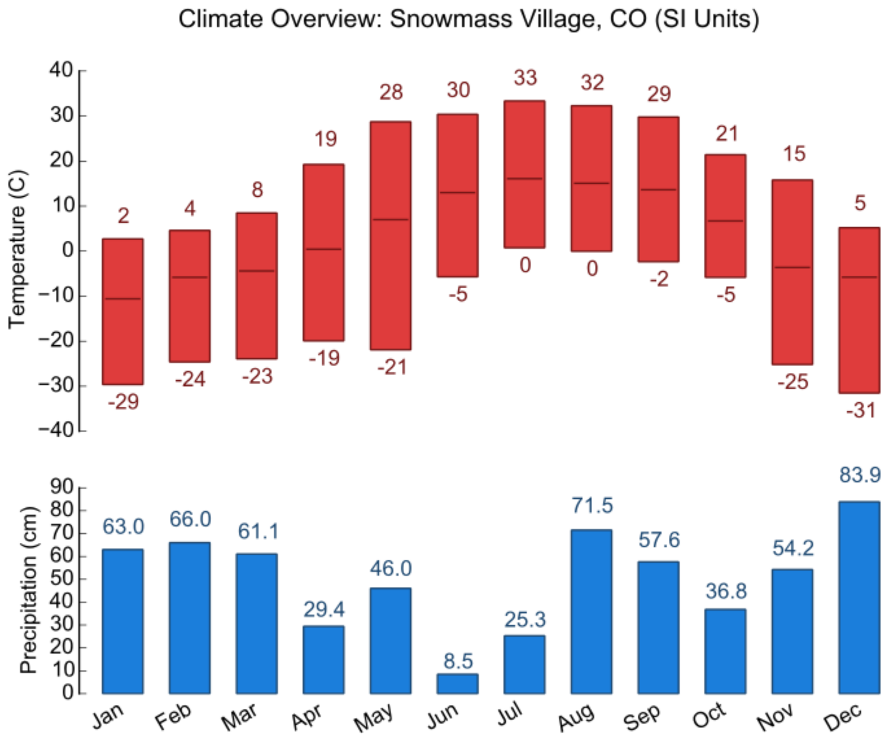


Contour Visualization

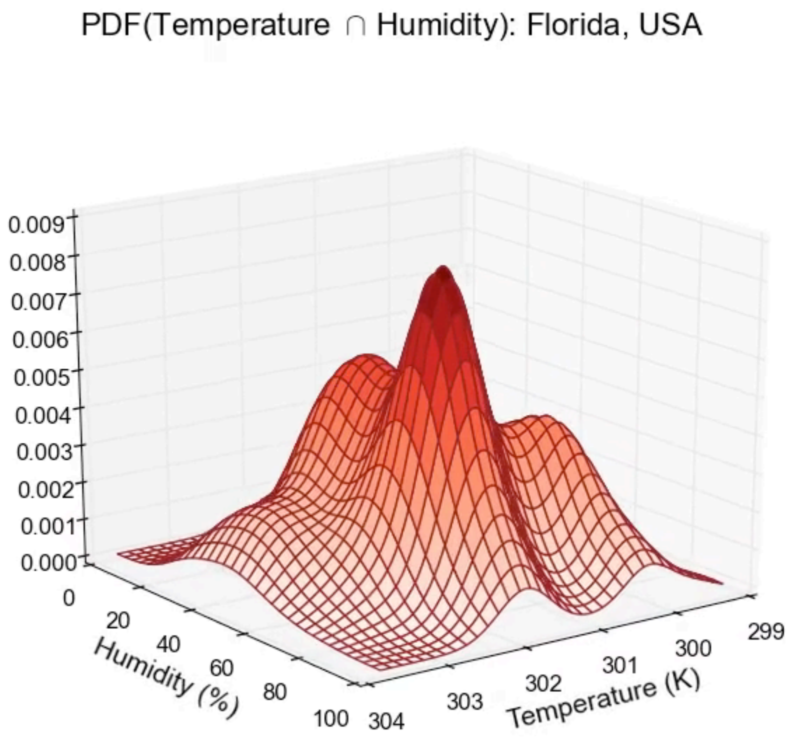


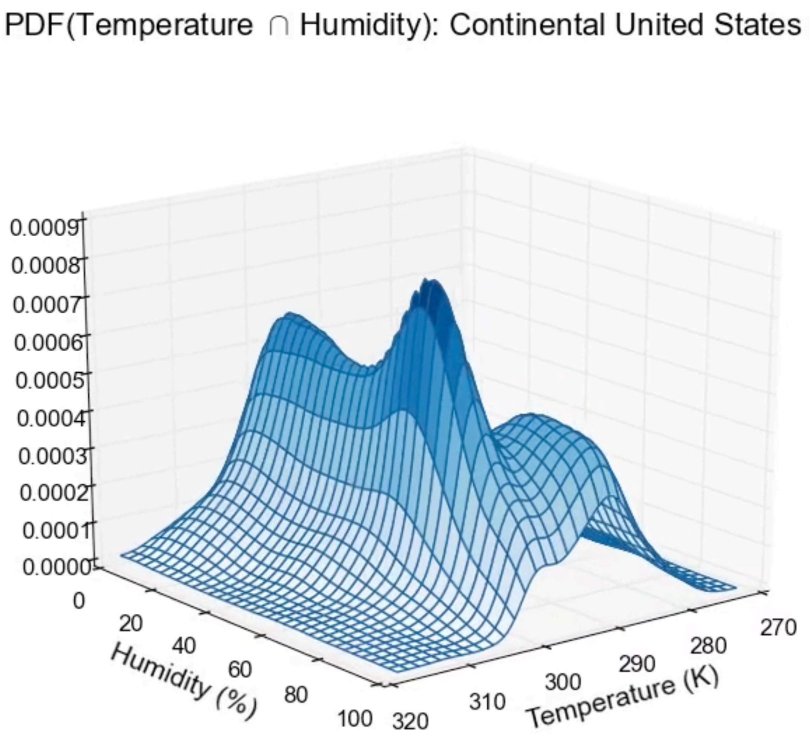
Climate Chart





Relationships: Temp & Humidity


PDF(Temperature ∩ Humidity): Florida, USA


PDF(Temperature ∩ Humidity): Continental United States

Gathering Insights

- This dataset contains a wealth of information, but extracting insights from the data is challenging
- Multiple dimensions
- Storage requirements: where do we put all of it?
- Querying the data
  - (*knowledge discovery*)

Today’s Schedule

- Spatiotemporal Data

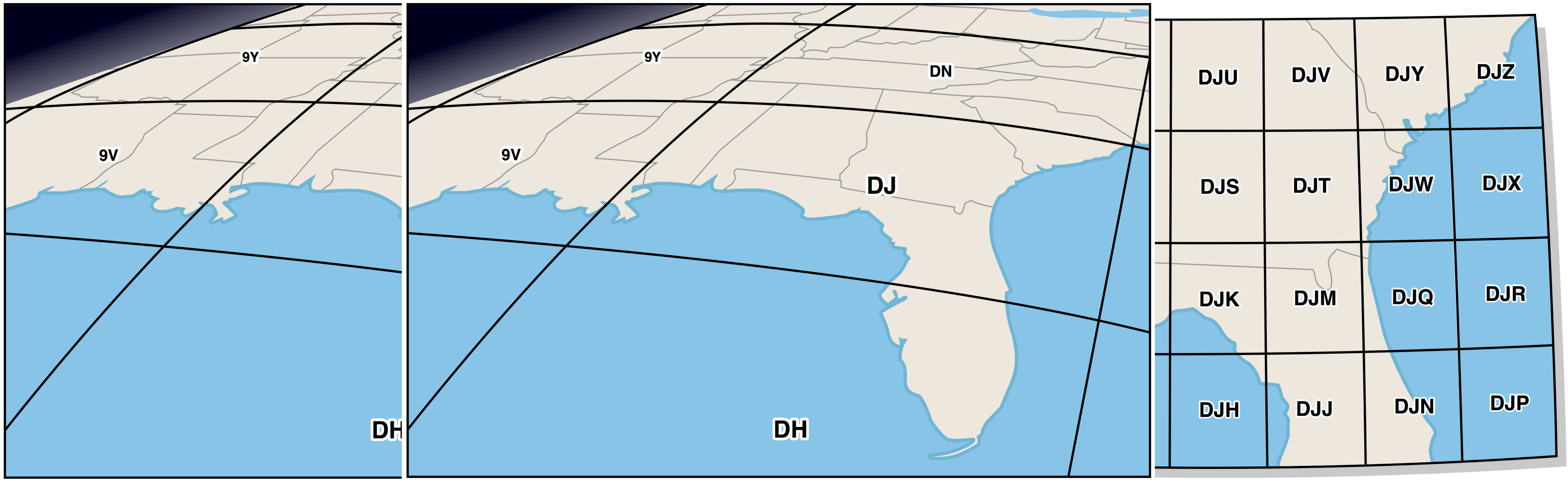
- **Geohash Algorithm**

Spatial Queries

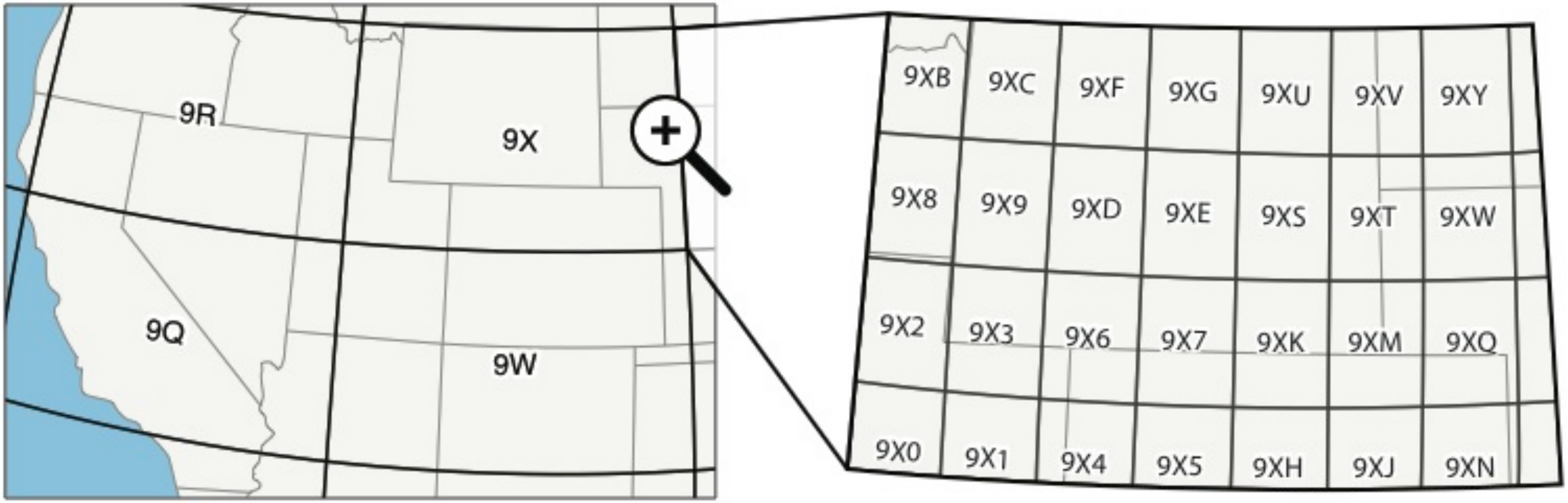
- Querying spatial data is a whole subject in itself
- If I gave you lat-lon pairs in the dataset, you could use those to perform simple spatial queries
    - If lat is >= something && lat <= something else:

      ```
                blah blah blah();
                etc();
      ```
- A **better** option is to use the Geohash algorithm
    - Maps the earth to base32 strings
    - Defines a spatial **hierarchy** we can exploit

The Geohash Algorithm (1/2)



The Geohash Algorithm (2/2)



Geohash Details

- We use the ***Geohash*** algorithm to represent the spatial location associated with our sensor readings
  - Maps 2D spatial locations to 1D strings
  - Precision is determined by string length
- 9X58VY4 ➔ Glenwood Springs, Colorado
  - Similar string prefixes refer to similar locations
- Want to support range queries? Just match more or less of the string prefix

Geohash Resolutions

- Spatiotemporal data is not always evenly distributed
    - Compare the density of New York City and Glenwood Springs, Colorado
- Hash: 9XJQBF
    - 9XJQ = 20x30 km
    - 9X = 600x1000 km

Geohash Implementation

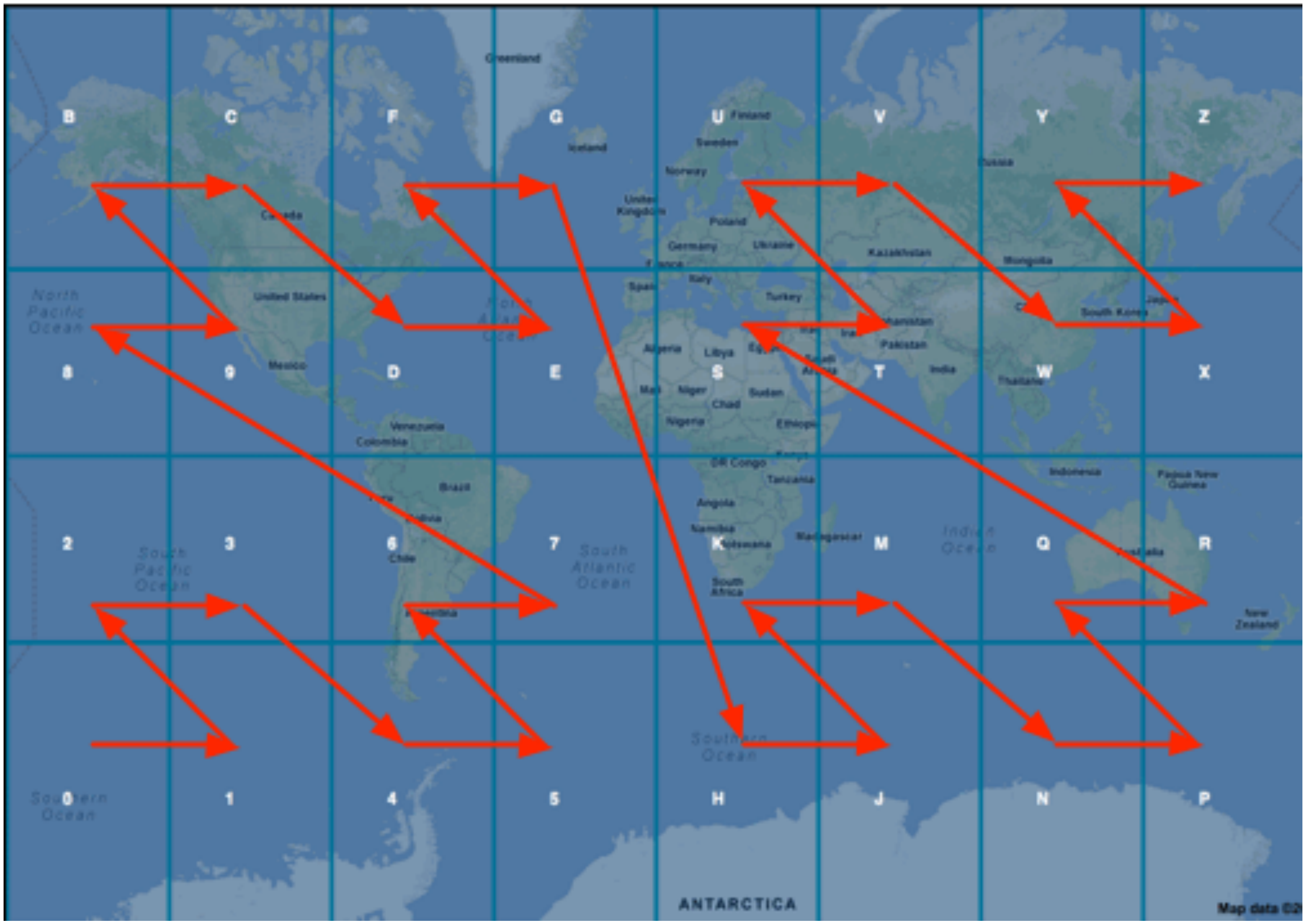
- Divides the bounding boxes in half with each binary bit added to the string
  - 1 bit = left or right half of the earth
  - 2 bits = top or bottom half of the left/right half
  - And so on…
- Uses 32 alphanumeric characters (Base 32)
  - 32 characters = 5 bits per character (5 divisions)
  - Omits some letters to avoid forming words

Encoding/Decoding

- An example Geohash-coordinate pair: 9QXY ⇔ (38, -113)
- Even bits = longitude = east-west
- Odd bits = latitude = north-south
- Each character represents 5 bits:

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Base 32	0	1	2	3	4	5	6	7	8	9	b	c	d	e	f	g
Decimal	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Base 32	h	j	k	m	n	p	q	r	s	t	u	v	w	x	y	z

Z-Order Curve



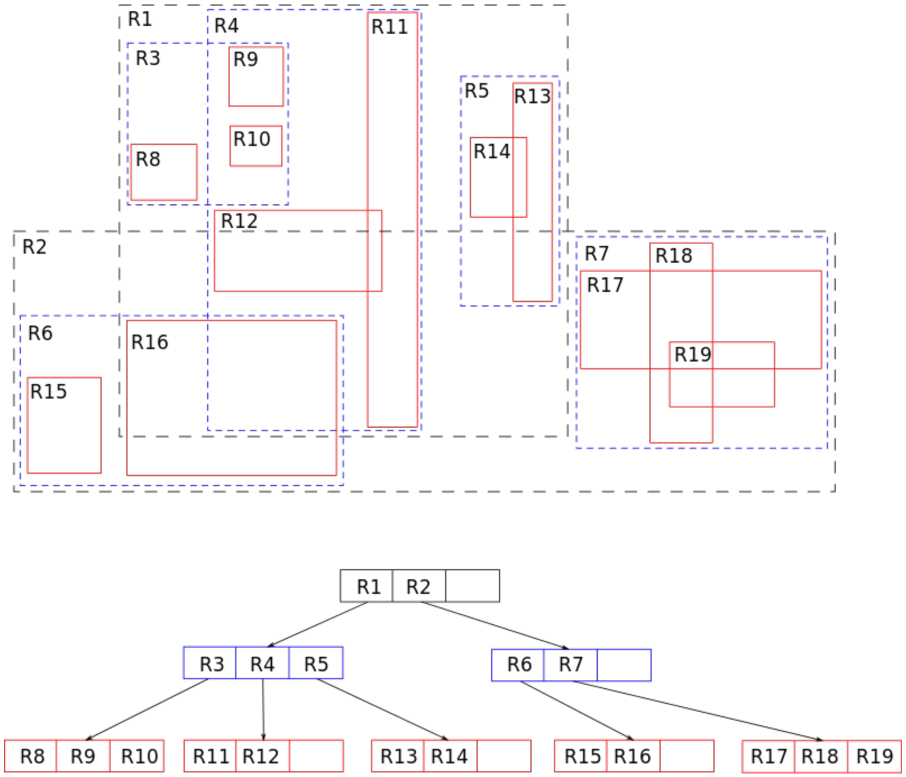
Source: http://www.bigdatamodeling.org/2013/01/intuitive-geohash.html

Geohash Fun Facts

- Originally designed to allow users to share short URLs that represent locations
- Similar implementations have been used to identify locations for businesses, government
  - Ireland's proof-of-concept ***openpostcode*** can uniquely identify all locations within the UK
- Play with it! https://geohash.softeng.co

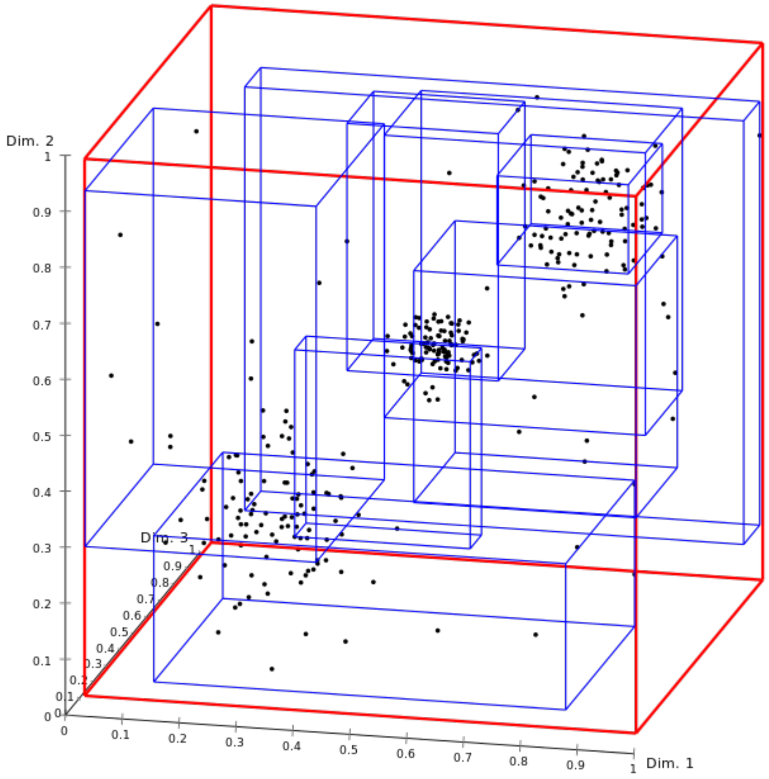
Spatial Indexing: R-Trees

- **R-Trees** are a widely-used spatial index
- Share many similarities with B-Trees, but support spatial features:
  - Multiple dimensions
  - Intersection, containment queries
  - Nearest neighbor search



R-Tree Drawbacks

- R-Trees can be overwhelmed by extremely large datasets
- Query performance decreases as the number of leaves in the tree expands
  - Too much precision



Applying this to P3…

- Let’s talk about how this helps us with P3.

Defining Regions via Geohash

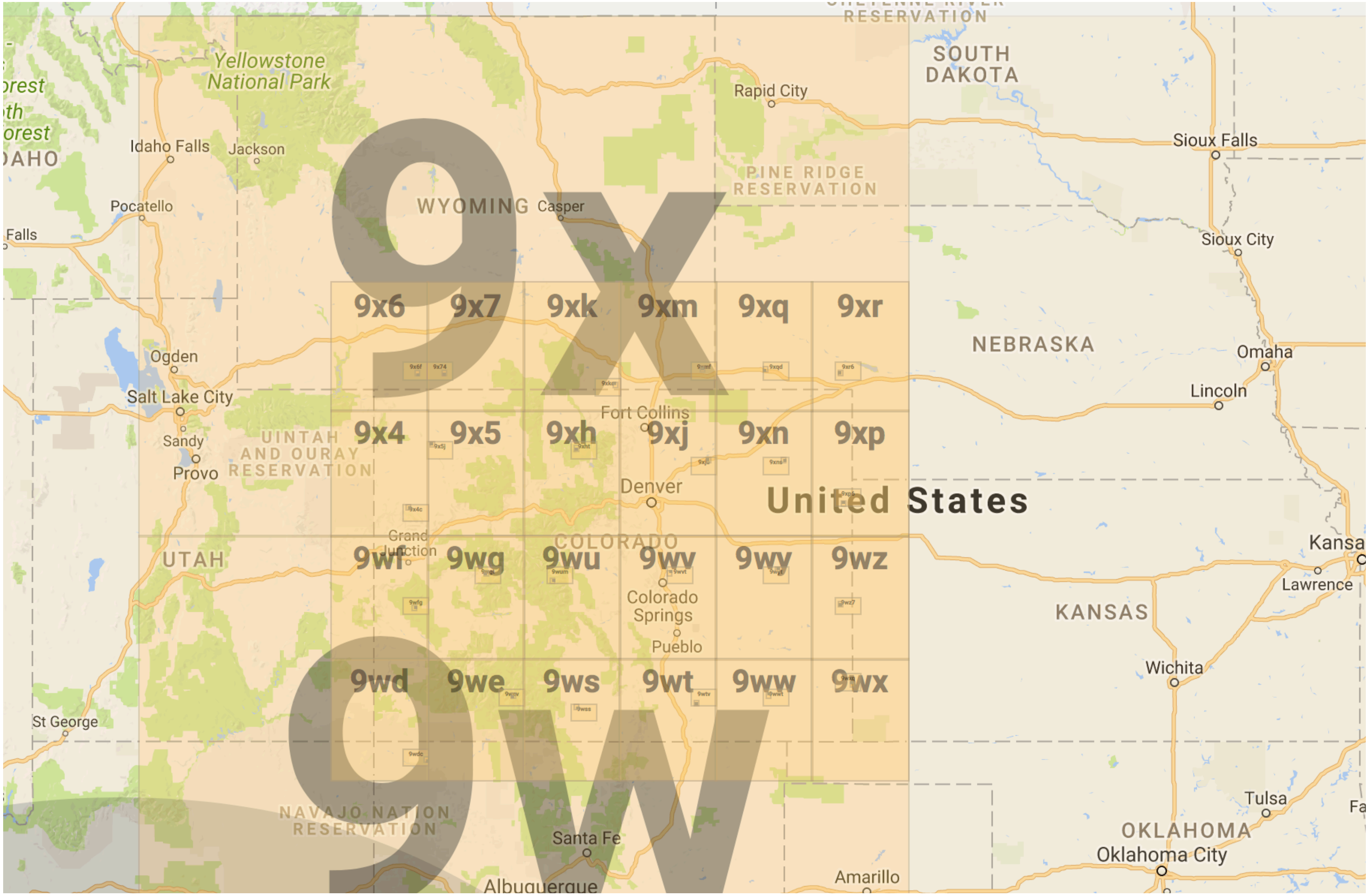
- How do we define regions via geohash? For example, the bay area.
- My recommendation:

  https://geohash.softeng.co

- Visually locate the areas you are interested and note their

  Geohashes in a list
- Then filter based on the entries in the list

Defining Colorado



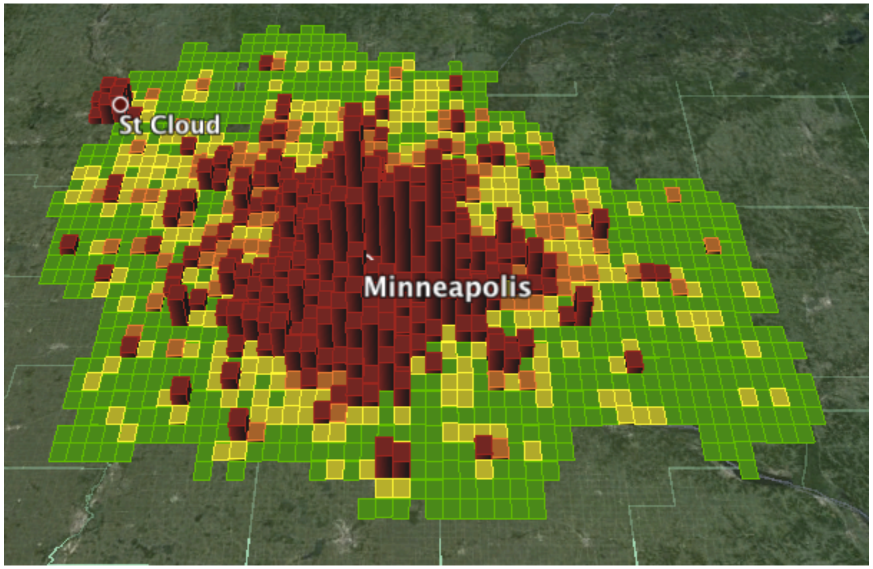
Constraining our Analysis

- For a few questions, I ask for a specific Geohash precision
  - For example, four-character Geohashes
- To do this, just chop the extra characters off the string:
  - 9xjq94b → 9xjq

Interesting: geohash2kml

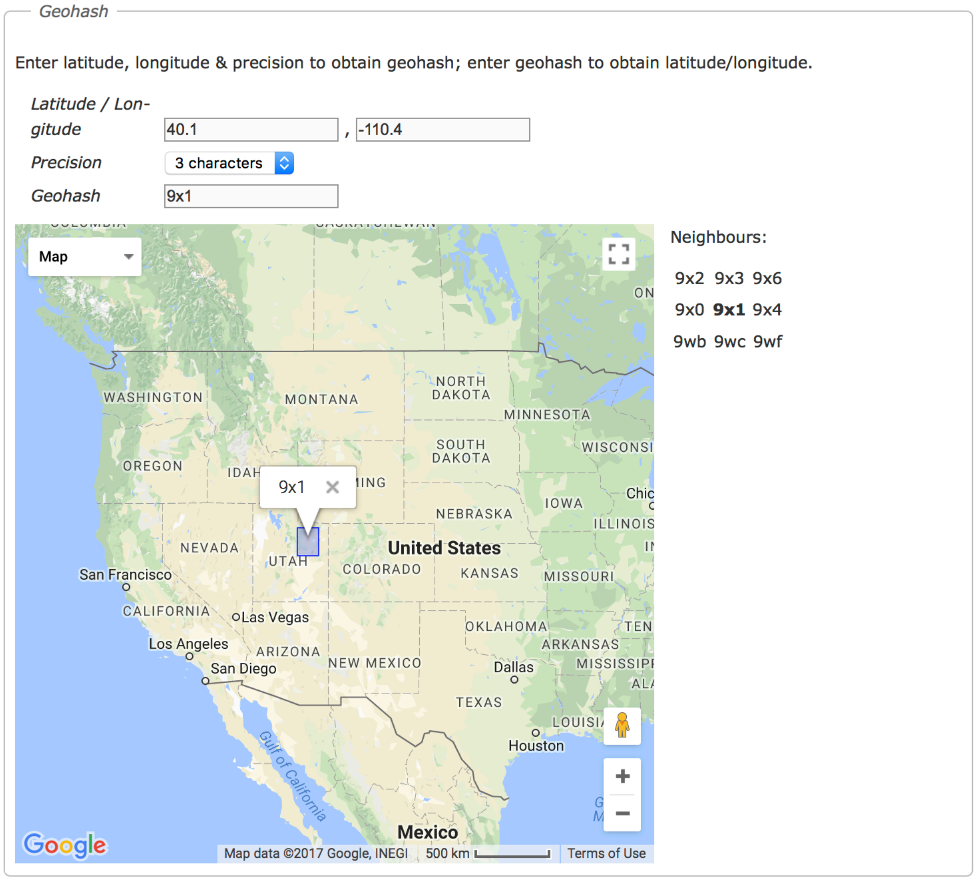
Here's a library for generating Google Earth visualizations:

https://github.com/abeusher/geohash2kml



Also Interesting: Geohash + Map

http://www.movable-type.co.uk/scripts/geohash.html



Geohash

Enter latitude, longitude & precision to obtain geohash; enter geohash to obtain latitude/longitude.

Latitude / Longitude	40.1 , -110.4
Precision	3 characters
Geohash	9x1

Neighbours:

9x2	9x3	9x6
9x0	**9x1**	9x4
9wb	9wc	9wf

Sanity Checking

- You are welcome to use other tools to learn more about the data
  - A text editor is a good way to start ☺
- Some basic python or shell scripts can confirm your Spark jobs are working properly
  - Run on a small subset of the input files, then verify with your scripts (or even visually by inspecting the source files)