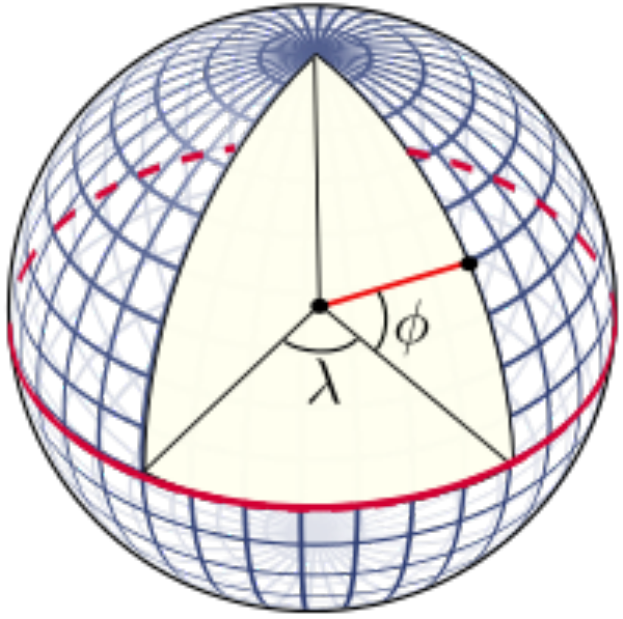
Geodata

What is geodata?

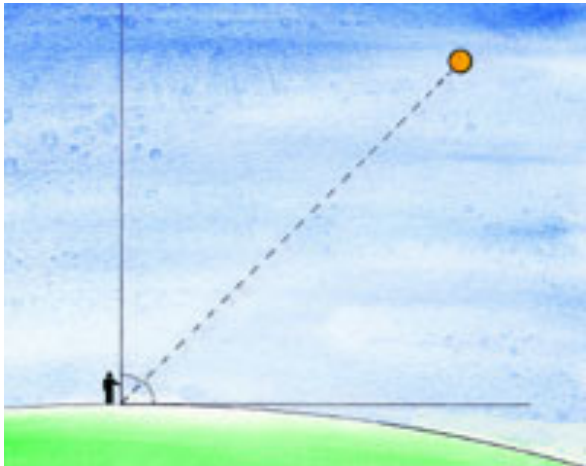
- spherical coordinate system using latitude, longitude, and elevation
- map coordinates projected onto the plane
- Hipparchus (2nd century BCE)
  - assumed a spherical earth, and divided it into 360°
  - latitude from stellar measurements
  - longitude by timings of lunar eclipses
  - prime meridian passed through Alexandria
- Latitude is around the axis of rotation
- Longitude is arbitrary
  - 1884: Greenwich Meridian international standard
  - Today: International Reference Meridian IERS (335’ E)
  - bisects center of mass
  - weighted average of hundreds of ground stations



Latitude (ϕ)

- angle between the equatorial plane and the straight line that passes through that point and through the center of the Earth
- 90N = North Pole, 90S = South Pole, 0 = Equator
- ellipsoid model leads to a variation of the nautical mile but for practical purposes one minute of latitude equals one nautical mile
- relatively easy to measure: angle of the sun at noon

https://www.open.edu/openlearn/society/politics-policy-people/geography/diy-measuring-latitude-and-longitude



Longitude (λ)

- angle E or W of a reference meridian to another that passes through that point
  - 0° at the prime meridian to +180° eastward and −180° westward
- 24 hours in a day and 360 degrees in a circle
- the sun moves across the sky at a rate of 15 degrees per hour (360° ÷ 24 hours = 15° per hour)
- compare local time to an absolute measure of time
- distance of a degree of longitude depends on the radius of a circle of latitude

φ	Δ^1_{lat}	Δ^1_{long}
0°	110.574 km	111.320 km
15°	110.649 km	107.551 km
30°	110.852 km	96.486 km
45°	111.133 km	78.847 km
60°	111.412 km	55.800 km
75°	111.618 km	28.902 km
90°	111.694 km	0.000 km

Longitude without a clock

Christopher Columbus used lunar eclipses to calculate longitude

- astronomical tables for reference
- Saona Island 1494 (second voyage)
  - off by 13°
  - San Jose - western Colorado
- Jamaica 1504 (fourth voyage)
  - off by 38°
  - San Jose - Cincinnati

Longitude and the clock problem

1530 - Gemma Frisius proposed calculating longitude using a clock: set clock on departure, compare with the local time on arrival. But accurate clocks weren't available for another 230 years.

1567 - Philip II of Spain offered a prize; similar challenge by Philip III in 1598

1667 - Italian astronomer Cassini timed eclipses of Jupiter's moons in Paris using a pendulum clock. Repeat in 1681 in the West Indies. Absolute time from the eclipses compared to local time = solved longitude on land!

1714 - the English Parliament offered a prize of £20,000 for longitude at sea to within a half a degree

Longitude and the clock problem

Pendulum clocks are unreliable on a ship.

Requirements

- not affected by variations in temperature, pressure or humidity
- accurate over long time intervals
- resist corrosion in salt air
- function on board a constantly-moving ship

John Harrison’s marine chronometer

- self-educated English carpenter and clockmaker
- took 6 years to create H4
- 1761: Portsmouth to Jamaica by sea
  - 81 days and 5 hours
  - on arrival: 5 seconds slow
  - error in longitude of 1.25 minutes
- Longitude Board called it luck and not practical
- Eventually awarded £8,750 from Parliament, but official award never given
- Revolutionized navigation! Safer long-distance sea travel!

PostGIS

- Postgres extension for Geometry data (not just geodata)
- Point but also other shapes: Polygon, Linestring
- Binary format with functions to work with other formats

```
gridium=# select city, state, ST_AsText(coordinates), ST_GeoHash(coordinates),
ST_AsGeoJSON(coordinates) from building limit 1;
-[ RECORD 1 ]+-----------------------------------------------------
city             | Mountain View
state            | CA
st_astext        | POINT(-122.5009522 37.8677705)
st_geohash    | 9q8zkmtebbj9c6hz42sc
st_asgeojson | {"type":"Point","coordinates":[-122.5009522,37.8677705]}

update building set coordinates=ST_GeomFromText('POINT(-87,63 41.88)', 4326),
city='Chicago', state='IL' where oid=3178643228654;
```

Storing geodata

- Work with geometry data in SQL
- Distance, bounding box, etc.

```
gridium=# select ST_AsText(w.coordinates)
gridium-# from wsi_axis w
gridium-# order by ST_Distance(w.coordinates,
ST_GeomFromText('POINT(-121.81544547168721 37.39132630396878)', 4326))
gridium-# limit 3;
         st_astext
---------------------------
 POINT(-121.88627 37.348541)
 POINT(-122.4019 37.791728)
 POINT(-122.54134 37.898058)
```

SRID

```
coordinates | geometry(Point,4326)
```

Coordinate + SRID define a location on the globe

SRID = spatial reference identifier

```
gridium=# select * from spatial_ref_sys where srid=4326;
srid      | 4326
auth_name | EPSG
auth_srid | 4326
srtext    | GEOGCS["WGS 84",DATUM["WGS_1984",SPHEROID["WGS
84",6378137,298.257223563,AUTHORITY["EPSG","7030"]],AUTHORITY["EPSG","6326"]],PRIMEM["Greenwich",0,
AUTHORITY["EPSG","8901"]],UNIT["degree",0.0174532925199433,AUTHORITY["EPSG","9122"]],AUTHORITY["EPS
G","4326"]]
proj4text | +proj=longlat +datum=WGS84 +no_defs
```

longitude/latitude on the WGS84 spheroid: latest revision of the World Geodetic System standard)

SRID on fields must be the same to use in functions!

Geocoding

Humans use addresses, not coordinates.

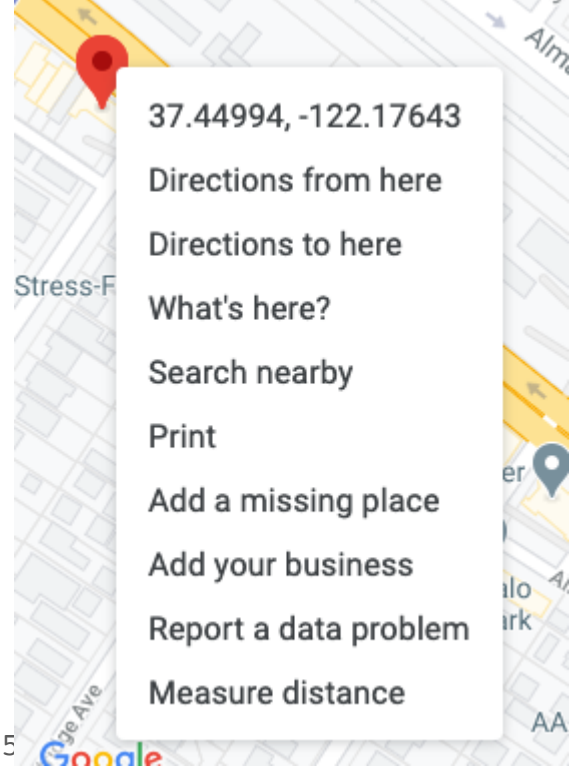
Geocoding = get coordinates from address

Manual: right click a point in Google Maps

Code: Google Maps API

```
response = requests.get(
          "https://maps.googleapis.com/maps/api/geocode/json",
          params=dict(
              sensor=False,
              address="405 El Camino Real #227 Menlo Park, CA 94025
              key=config.GOOGLE_GEOCODE_API_KEY))

response.json()["results"][0]["geometry"]["location"]
{'lat': 37.4497613, 'lng': -122.1765008}
```



Geodata formats: GeoJSON

JSON format

- organize features into `FeatureCollection`
- add arbitrary properties

```
{
   "type": "Feature",
   "geometry": {
     "type": "Point",
     "coordinates": [-122.5009522,37.8677705]
   },
   "properties": {
     "prop0": "value0"
   }
 },
```

Geodata KML

Keyhole Markup Language

- XML
- used by Google Earth, now an open standard

```
<Placemark>
  <name>River Oaks access</name>
  <styleUrl>#icon-1615-0288D1-nodesc</styleUrl>
  <Point>
    <coordinates>
  -121.9418559,37.4005623,0
    </coordinates>
  </Point>
</Placemark>
```

Geodata formats: GPX

GPS Exchange format

- waypoints, tracks, and routes

```
<?xml version="1.0" encoding="UTF-8"?>
<gpx
  version="1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.topografix.com/GPX/1/1"
  xsi:schemaLocation="http://www.topografix.com/GPX/1/1 http://www.topografix.com/GPX/1/1/gpx.xsd"
  xmlns:gpxtpx="http://www.garmin.com/xmlschemas/TrackPointExtension/v1">
<trk>
  <name><![CDATA[Other 3/28/21 2:13 pm]]></name>
  <time>2021-03-28T21:13:40Z</time>
<trkseg>
<trkpt lat="37.462247000" lon="-121.935544000"><ele>0.0</ele><time>2021-03-28T21:13:40Z</time></trkpt>
<trkpt lat="37.462158000" lon="-121.935496000"><ele>-0.1</ele><time>2021-03-28T21:18:03Z</time></trkpt>
<trkpt lat="37.462078000" lon="-121.935550000"><ele>-0.1</ele><time>2021-03-28T21:18:16Z</time></trkpt>
```

Geodata examples

Google My Maps

- Guadalupe River Trail

Strava global heat map

NY Times stories

- They Stormed the Capitol. Their Apps Tracked Them.
- Your Apps Know Where You Were Last Night, and They’re Not Keeping It Secret