Goals

Be able to mark areas on a map and:

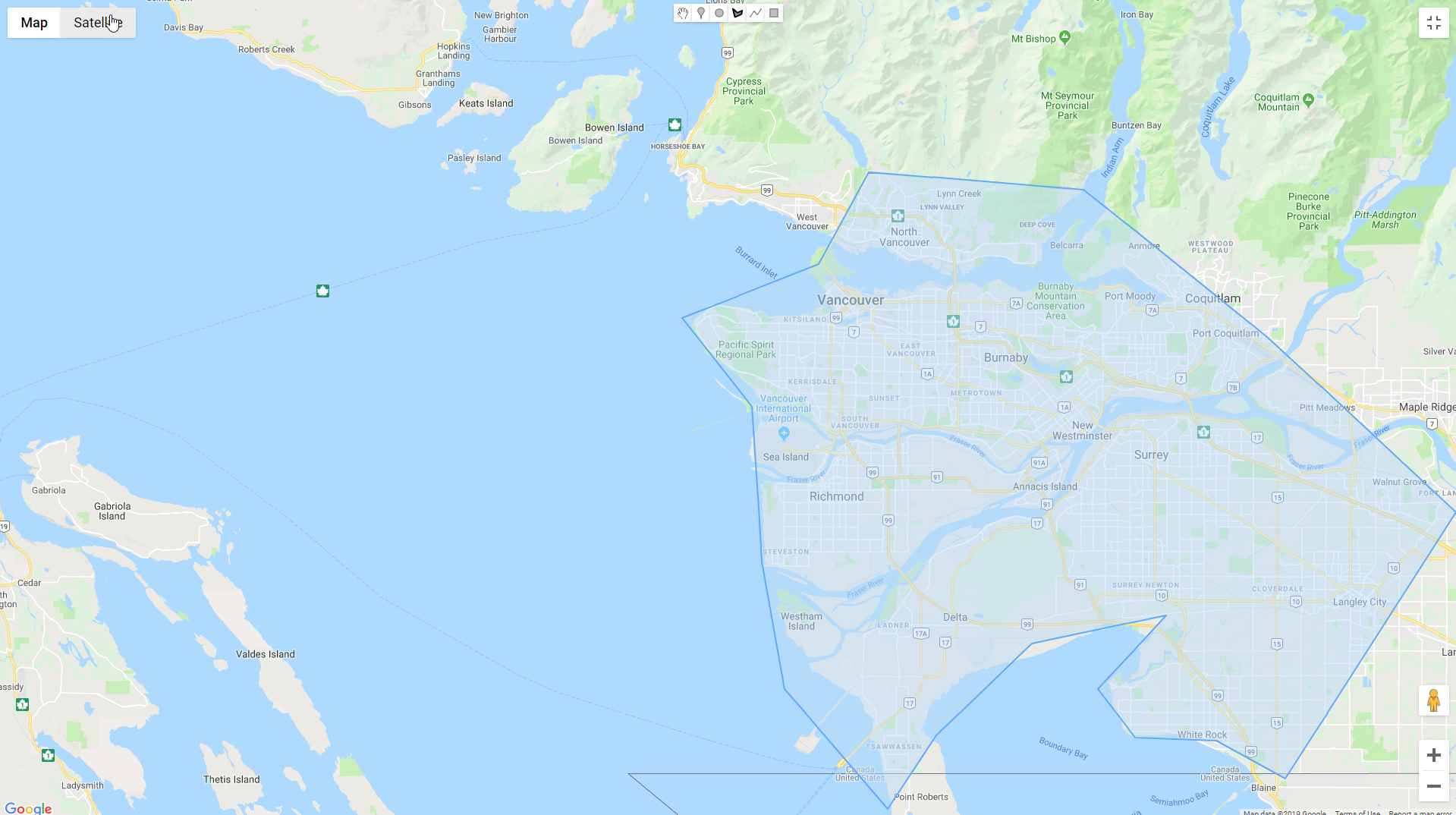
* Determine which areas are there (by city, community, postal code, region etc)
* Determine which records we have in LINKS are in those areas
* Store polygons in db, and/or shapefile

Identify which tools/technology may help us with this project

* ArcGIS? QGIS?
* Google Maps API? Bing Maps API?
* Python (with shapely? Geopandas? )

Ideas

<https://codepen.io/chrisN/pen/RRBvRB>



**polygon points:**49.357846,-123.104213;49.303701,-123.149532;49.271902,-123.273128;49.219457,-123.209957;49.12788,-123.20103;49.052336,-123.180431;48.981184,-123.087047;49.024878,-123.043102;49.079329,-122.956584;49.095967,-122.835048;49.052336,-122.896846;49.023527,-122.863201;49.021726,-122.78973;48.999207,-122.727245;49.157077,-122.57275;49.260253,-122.743038;49.347559,-122.909893;

See stack overflow:

<https://stackoverflow.com/questions/43892459/check-if-geo-point-is-inside-or-outside-of-polygon>

from shapely.geometry import Point

from shapely.geometry.polygon import Polygon

lons\_lats\_vect = np.column\_stack((lons\_vect, lats\_vect)) # Reshape coordinates

polygon = Polygon(lons\_lats\_vect) # create polygon

point = Point(y,x) # create point

print(polygon.contains(point)) # check if polygon contains point

print(point.within(polygon)) # check if a point is in the polygon

See <https://automating-gis-processes.github.io/2016/Lesson3-point-in-polygon.html>

**How to check if point is inside a polygon?**

Computationally, detecting if a point is inside a polygon is most commonly done using a specific formula called [Ray Casting algorithm](https://en.wikipedia.org/wiki/Point_in_polygon#Ray_casting_algorithm). Luckily, we do not need to create such a function ourselves for conducting the Point in Polygon (PIP) query. Instead, we can take advantage of [Shapely’s binary predicates](http://toblerity.org/shapely/manual.html" \l "binary-predicates) that can evaluate the topolocical relationships between geographical objects, such as the PIP as we’re interested here.

There are basically two ways of conducting PIP in Shapely:

1. using a function called [.within()](http://toblerity.org/shapely/manual.html#object.within) that checks if a point is within a polygon
2. using a function called [.contains()](http://toblerity.org/shapely/manual.html#object.contains) that checks if a polygon contains a point

Notice: even though we are talking here about **Point** in Polygon operation, it is also possible to check if a LineString or Polygon is inside another Polygon.

<https://kodu.ut.ee/~kmoch/geopython2018/lessons/L3/point-in-polygon.html#point-in-polygon-using-geopandas>

**In [16]: import** **geopandas** **as** **gpd**

**In [17]: import** **matplotlib.pyplot** **as** **plt**

**In [18]:** gpd.io.file.fiona.drvsupport.supported\_drivers['KML'] = 'rw'

*# Filepath to KML file*

fp = r"Data\PKS\_suuralue.kml"

**In [28]: import** **shapely.speedups**

**In [29]:** shapely.speedups.enable()

**In [30]:** pip\_mask = data.within(southern.loc[0, 'geometry'])

Google Maps

<https://stackoverflow.com/questions/41203225/check-if-point-is-in-polygon-with-google-maps-api-in-android>

You can use the PolyUtil.containsLocation method from the [Google Maps Android API Utility Library](https://developers.google.com/maps/documentation/android-api/utility/). From [the documentation](http://googlemaps.github.io/android-maps-utils/javadoc/com/google/maps/android/PolyUtil.html#containsLocation-LatLng-java.util.List-boolean-):

**public static boolean containsLocation(LatLng point, java.util.List polygon, boolean geodesic)**

Computes whether the given point lies inside the specified polygon. The polygon is always considered closed, regardless of whether the last point equals the first or not. Inside is defined as not containing the South Pole -- the South Pole is always outside. The polygon is formed of great circle segments if geodesic is true, and of rhumb (loxodromic) segments otherwise.