

Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación



# IIC2115 - Programación como herramienta para la ingeniería

Análisis de datos geoespaciales

**Profesor:** Hans Löbel

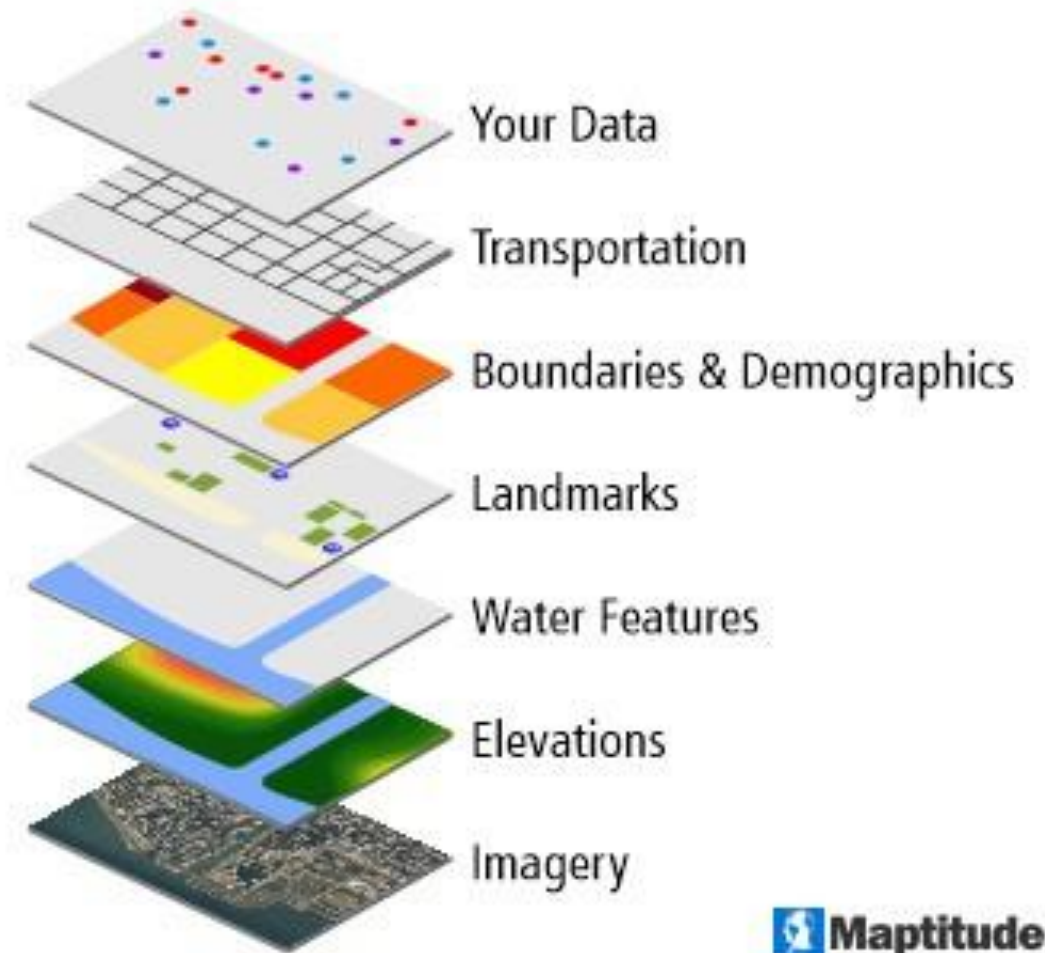
¿Cómo podríamos encontrar el mejor sector para vivir en esta ciudad?  
(manteniendo la idea del análisis exploratorio de datos)



# ¿Cómo podríamos encontrar el mejor sector para vivir en esta ciudad?

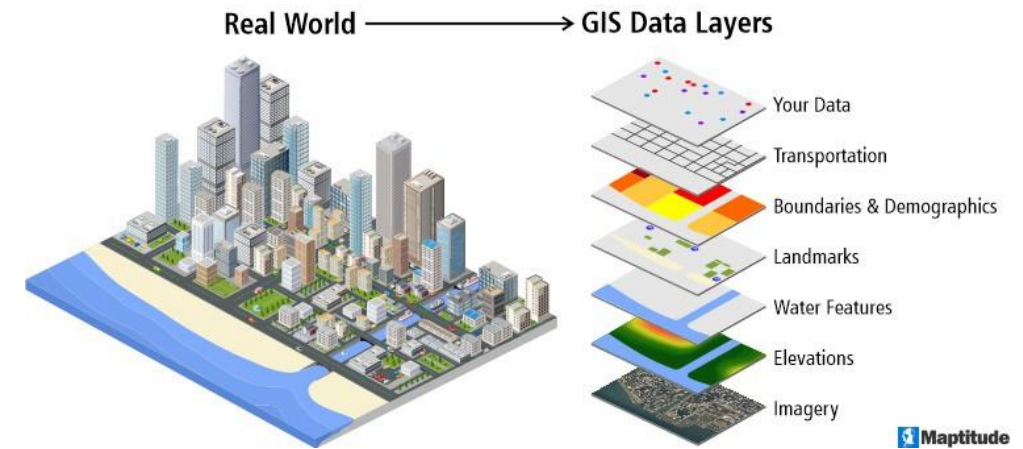
(manteniendo la idea del análisis exploratorio de datos)

**Real World** → **GIS Data Layers**



# ¿Qué es un sistema de información geográfica (SIG o GIS)?

- En simple, es la unión entre datos con **información espacial** y el conjunto de **herramientas de software** que permiten manipularla.
- Los datos se encuentran en tablas/capas y poseen **información geométrica** georreferenciable.
- De esta forma, se pueden generar cruces entre ellos, hacer resúmenes estadísticos, incorporar nuevos datos, etc.
- Visualización se realiza generalmente a través de mapas.



¿Qué GIS son utilizados regularmente?



**ArcGIS**



**pandas**

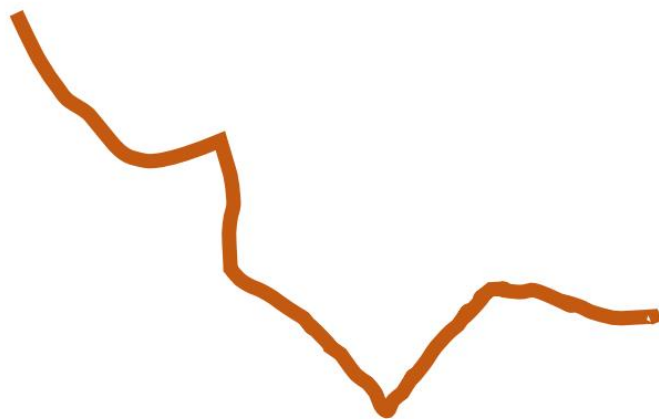


**GeoPandas**

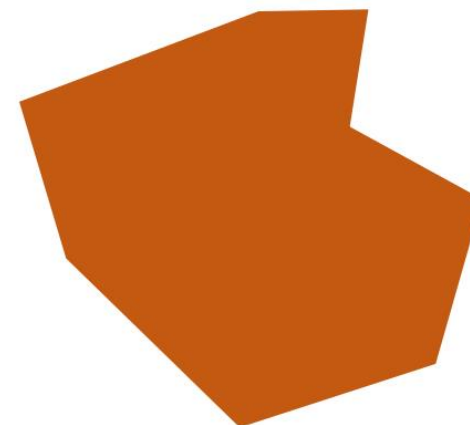
Cada **entidad** en un GIS está asociado a una **geometría**



**Punto**



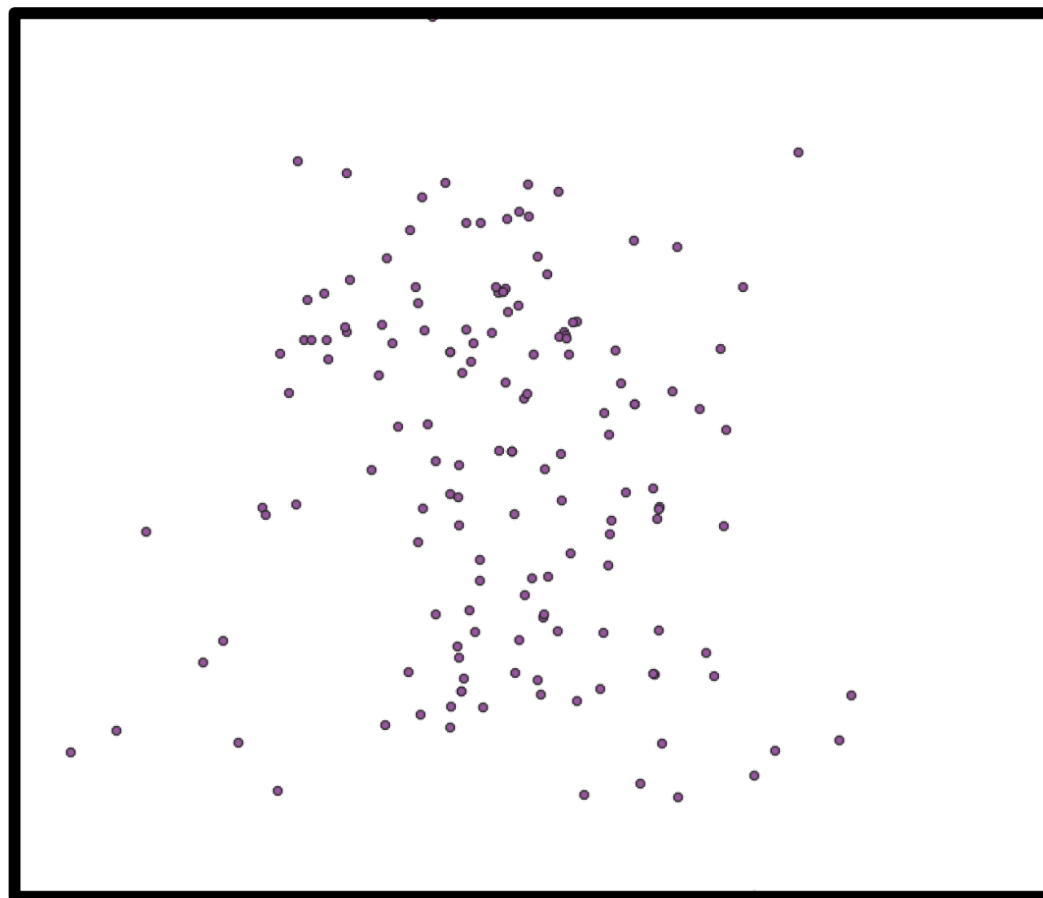
**Línea**



**Polígono**



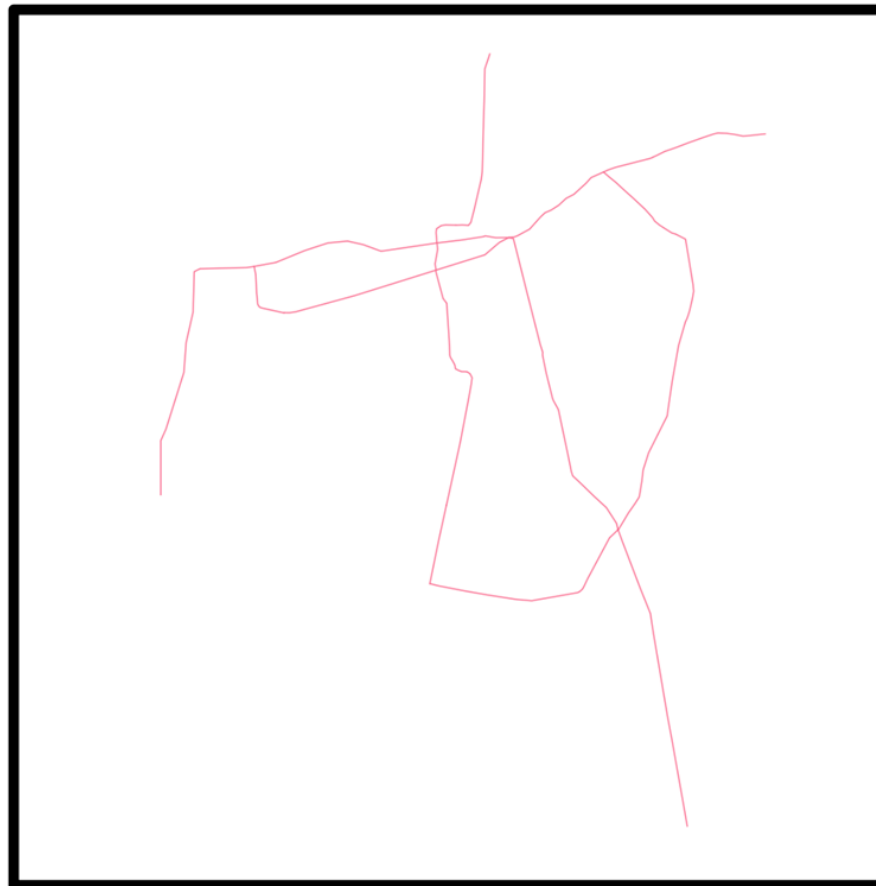
Punto



Servicios de Salud de Santiago

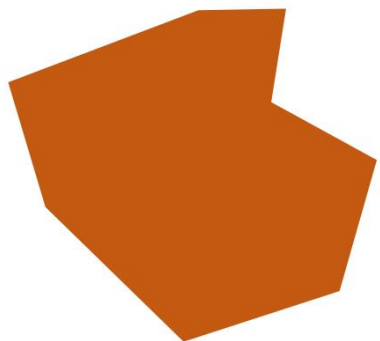


Línea

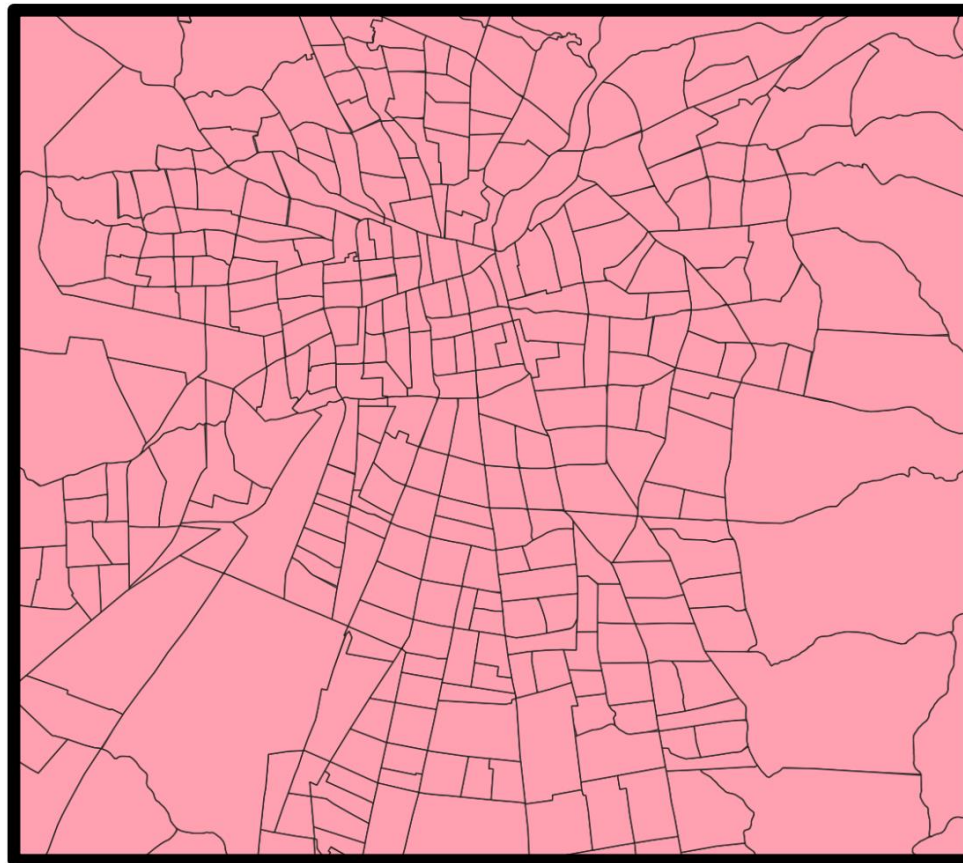


Líneas de Metro de Santiago





Polígono

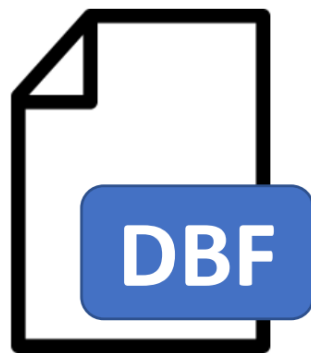


Distritos censales de Santiago

Dada su complejidad, los datos geoespaciales requieren múltiples archivos para su almacenamiento



Geometrías



Metadatos



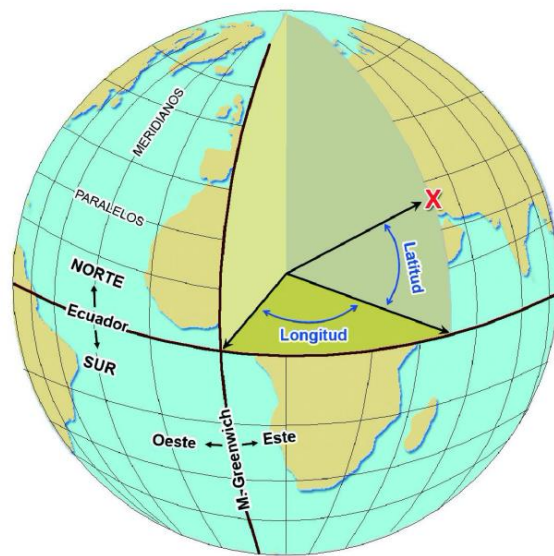
Índice  
geometría-datos



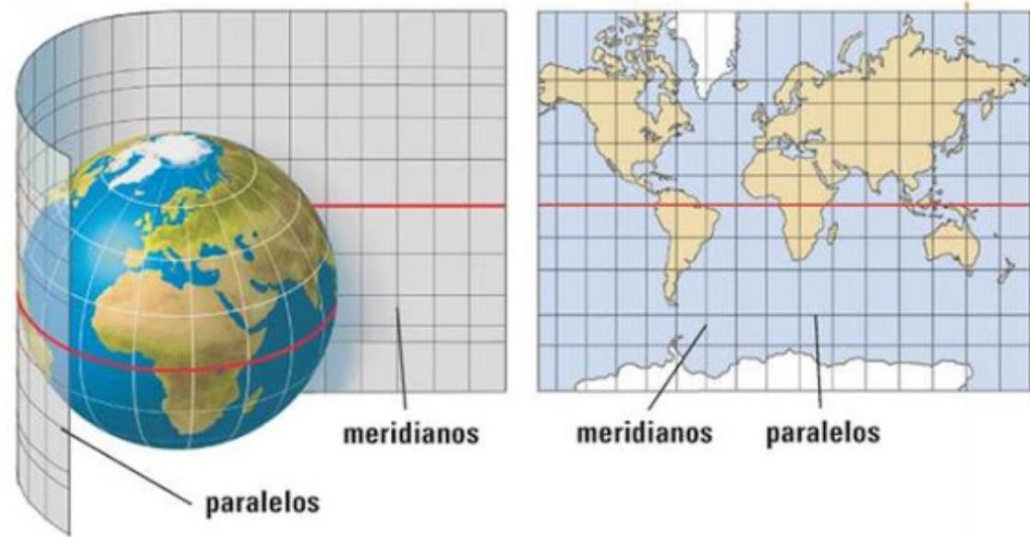
Proyección  
geográfica

Más adelante notaremos que solo “vemos” 1 archivo

Un último punto importante, las proyecciones...



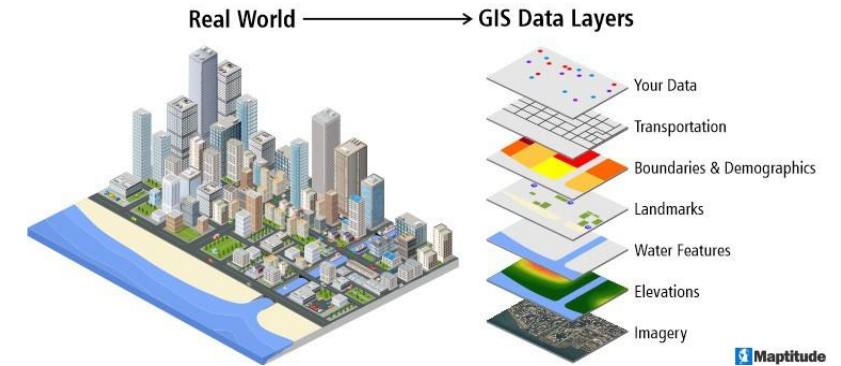
Grados



UTM

## ¿Qué tipo de ejercicios considera este capítulo?

- El flujo comienza típicamente con la carga de uno o más archivos *shape*, que contienen la información de las capas relevantes.
- Posteriormente, se transforman las capas a las proyecciones adecuadas y luego se grafican, con el fin de hacerse una idea del objeto de análisis.
- A continuación, se combina la información de las capas, ya sea a través de operaciones tabulares (*pandas*) o espaciales (*geopandas*).
- Finalmente, se grafica el resultado de las operaciones.



# Carga de datos

```
import geopandas as gpd

distritos = gpd.read_file('Data/Distritos Censales/Distritos Censales RM.shp')
lineas_metro = gpd.read_file('Data/Metro 2020/Lineas_2020/Lineas_2020.shp')
```

```
1 distritos.head(2)
```

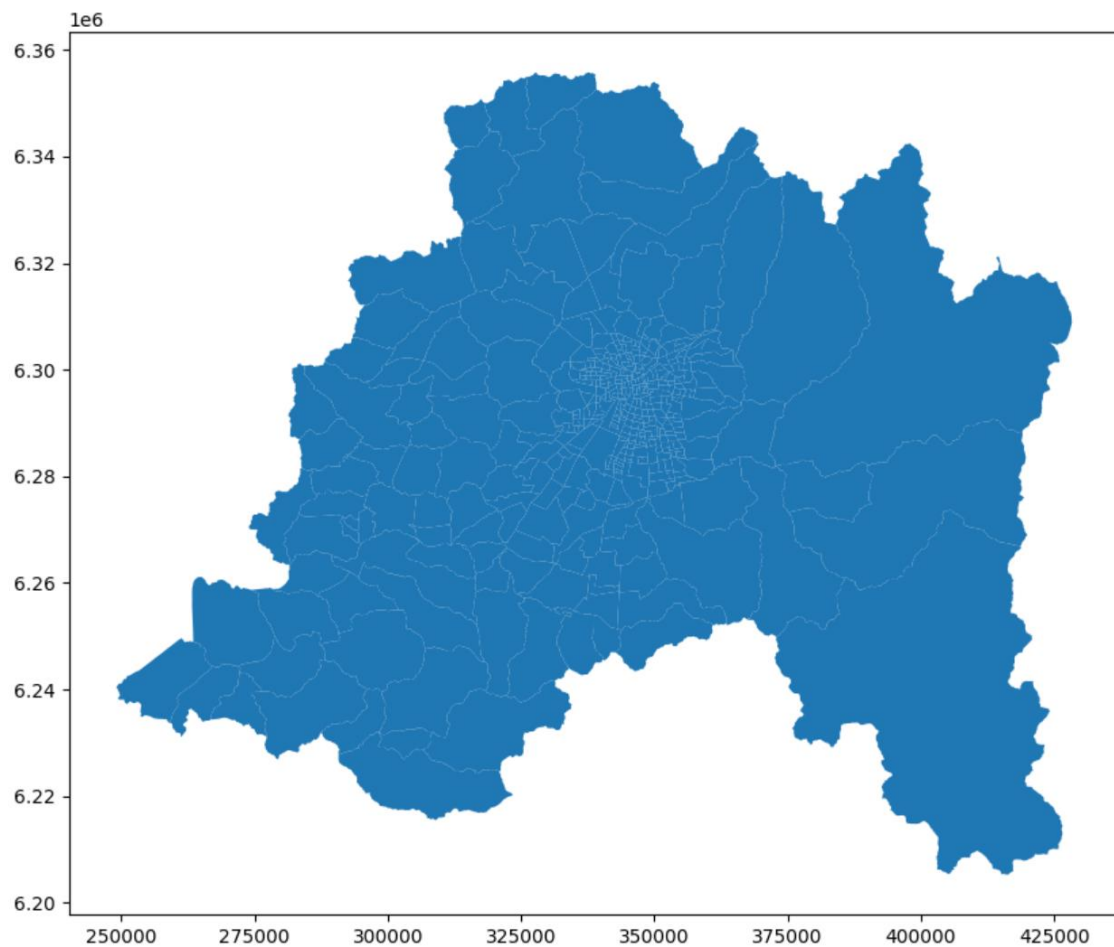
	NOM_REG	COD_PROV	NOM_PROV	COD_COM	NOM_COM	NOM_DIS	CODIGO	E	D	C3	C2	ABC1	ADIMARK_T	Area	Densidad	1_UF_M2	geometry
0	Región Metropolitana De Santiago	131	Santiago	13101	Santiago	Huelén	1310101.0	28.0	397.0	1232.0	1390.0	365.0	3412.0	704315.75	48.44	31.09	POLYGON (((346775.511 6299463.724, 346788.392 6...
1	Región Metropolitana De Santiago	131	Santiago	13101	Santiago	Moneda	1310102.0	51.0	560.0	999.0	903.0	228.0	2741.0	1166659.89	23.49	21.10	POLYGON (((345880.280 6299633.958, 345961.610 6...

```
1 lineas_metro.head(2)
```

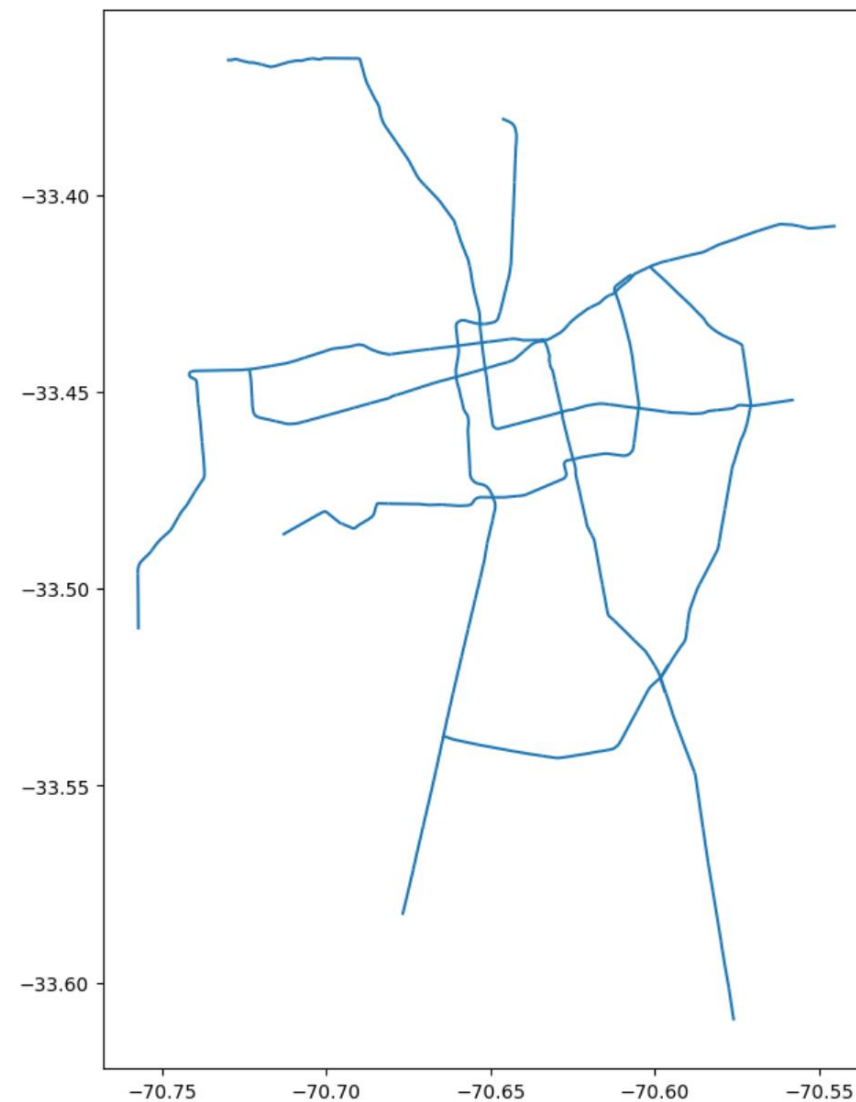
	OBJECTID	ID_LINEA	LINEA	LENGTH	Shape_Leng	Shape_Le_1	geometry
0	3	2	L2	18315.942	20087.016072	1301.975352	LINESTRING (-70.66142 -33.52651, -70.66325 -33...
1	4	2	L2	18315.942	20087.016072	1057.384032	LINESTRING (-70.65881 -33.51723, -70.66138 -33...

# Visualización y proyección

```
1 distritos.plot(figsize = (10,10));
```



```
1 lineas_metro.plot(figsize = (10,10));
```



# Visualización y proyección

## 1 distritos.crs

```
<Projected CRS: EPSG:32719>
Name: WGS 84 / UTM zone 19S
Axis Info [cartesian]:
- E[east]: Easting (metre)
- N[north]: Northing (metre)
Area of Use:
- name: Between 72°W and 66°W, southern hemisphere between 80°S and equator, onshore and offshore. Argentina. Bolivia. Brazil. Chile. Colombia. Peru.
- bounds: (-72.0, -80.0, -66.0, 0.0)
Coordinate Operation:
- name: UTM zone 19S
- method: Transverse Mercator
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

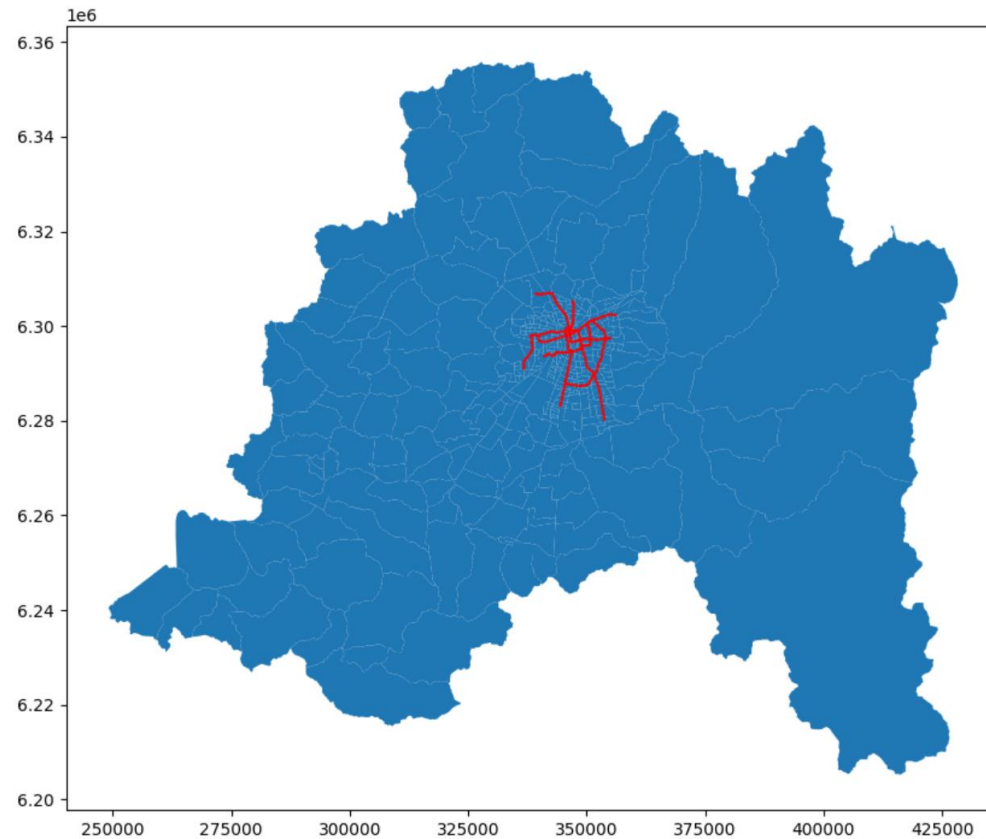
## 1 lineas\_metro.crs

```
<Geographic 2D CRS: EPSG:4326>
Name: WGS 84
Axis Info [ellipsoidal]:
- Lat[north]: Geodetic latitude (degree)
- Lon[east]: Geodetic longitude (degree)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```



# Visualización y proyección

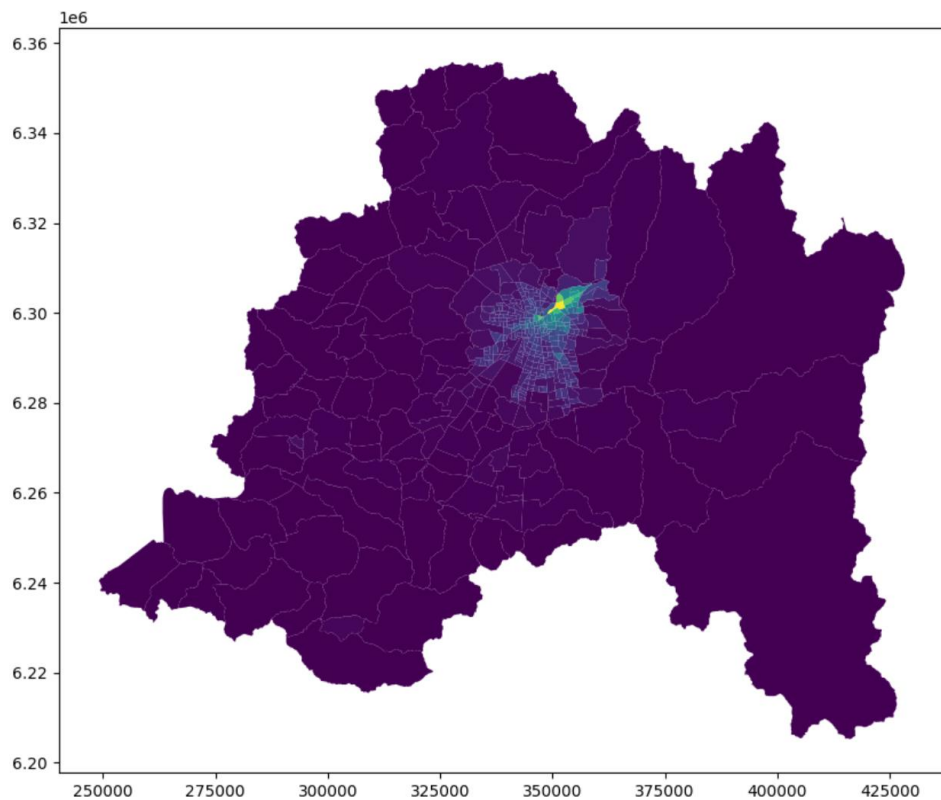
```
1 lineas_metro_utm = lineas_metro.to_crs(32719)
2
3 # Guardamos nuestro primer gráfico en una variable
4 ax = distritos.plot(figsize = (10,10));
5
6 # Usamos esa variable (que es un axes) y construimos el segundo gráfico con el anterior como base
7 # Es importante que le cambiemos el color para que se puedan visualizar sin problemas.
8 lineas_metro_utm.plot(ax = ax, figsize = (10,10), color = 'red');
```



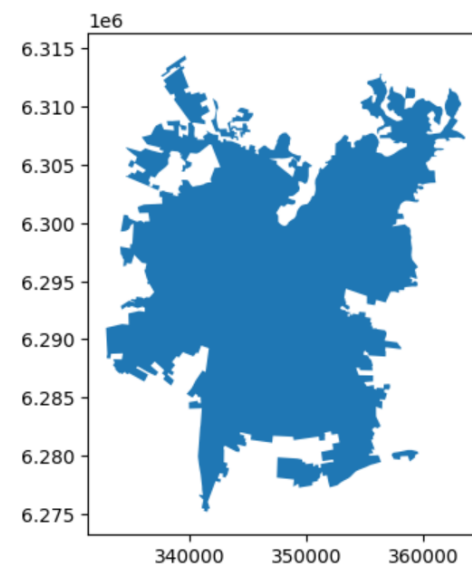


# Combinación de información tabular y espacial

```
1 distritos.plot(column = '1_UF_M2', figsize = (10,10));
```

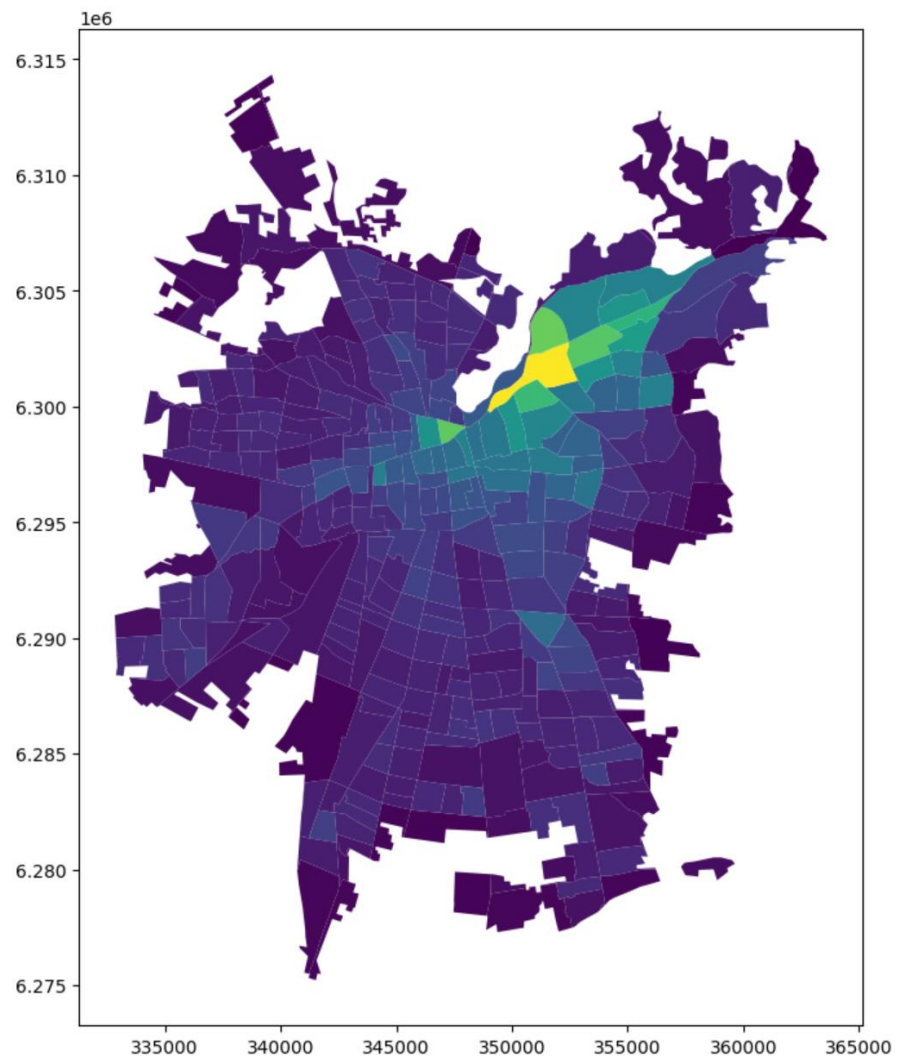


```
1 # Cargamos el shape de areas urbanas de chile
2 chile_urbano = gpd.read_file('Data/Areas Urbanas/areas_urbanas.shp')
3
4 # Filtramos solo el de Santiago y visualizamos
5 santiago_urbano = chile_urbano.loc[chile_urbano['NOMBRE']=='Santiago']
6 santiago_urbano.plot();
```



# Combinación de información tabular y espacial

```
1 distritos_urbano = gpd.overlay(distritos, santiago_urbano, how='intersection')  
2 distritos_urbano.plot(figsize = (10,10), column = '1_UF_M2');
```



Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación



# IIC2115 - Programación como herramienta para la ingeniería

Análisis de datos geoespaciales

**Profesor:** Hans Löbel