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

IIC2115 – Programación como Herramienta para la Ingeniería (I/2019)

Laboratorio 10 - Tópicos avanzados

Objetivos

- Utilizar conceptos básicos de *Web Scraping* para extraer información de una página web.
- Utilizar la librería Geopandas para la visualización de mapas.

Entrega

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **L10**.
- **Primera entrega parcial: lunes 10 de junio a las 16:50 hrs.**
- **Segunda entrega parcial: lunes 17 de junio a las 16:50 hrs.**
- **Entrega final: domingo 23 de junio a las 23:59 hrs.**
- **Formato de entrega:** archivo python notebook (**.ipynb**) con la solución, ubicado en la carpeta **L10**. Suba además, en la misma carpeta, un archivo `README.md` con las instrucciones para ejecutar su tarea. No se debe subir ningún otro archivo a la carpeta. Utilice múltiples celdas de texto y código para facilitar la revisión de su tarea. **No suba los archivos CSV utilizados o va a tener problemas al subir el laboratorio**.
- **Descuentos:** se descontará 0.5 ptos. por cada hora de atraso y fracción en la entrega final. Tareas que no cumplan el formato de entrega tendrán un descuento de 0.5 ptos.
- **Entregas parciales subidas fuera de plazo no serán consideradas.**

- **Tareas con errores de sintaxis y/o que generen excepciones serán calificadas con nota 1.0.**
- Las discusiones en las *issues* del Syllabus en GitHub son parte de este enunciado.

Introducción

Geografía de servicios de Transantiago

En el desarrollo de los laboratorios 7, 8 y 9 usted ha estado trabajando con los datos de ciertos servicios de buses para determinar información específica de los GPS de Transantiago.

Lamentablemente, en las oficinas de DTPM se ha destruido uno de los servidores que almacenan la información de los trazados de los servicios de buses. Los trazados corresponden a las rutas geográficas que deben seguir los buses para operar en el sistema. Por suerte se han salvado los archivos que almacenan información de las paradas y las zonas de Transantiago, ambos en formato *shape*. Un archivo *shape* almacena las posiciones y geometrías de objetos.

Dada las pérdidas que ha sufrido DTPM, usted deberá ser capaz de administrar la información disponible para generar parte de la información perdida. Su trabajo se basará principalmente en el uso de una librería llamada **Geopandas**. Esta librería trabaja con los conocidos *dataframes* pero esta vez incluyen información geográfica.

Los datos disponibles

En este laboratorio se cuenta con la base de datos de emisiones GPS (utilizada en los laboratorios 7, 8 y 9), un archivo *shape* de las zonas de Transantiago y un archivo *shape* de los paraderos de buses del sistema. Estos archivos se encuentran en el mismo link de laboratorios pasados. Esta vez va a encontrar un total de cinco carpetas, tres de las cuales poseen la información de las emisiones GPS, y dos carpetas con los archivos *shapes* mencionados anteriormente.

Las tres carpetas conocidas poseen un archivo llamado **emisiones.csv**. La cantidad de datos es la única diferencia entro los archivos, donde la de menor tamaño es "emisiones muy pequeña", luego "1 semana" y finalmente, la más grande es "30 dias". Recuerde que los valores de la primera fila corresponden a los nombres de las columnas y el resto de las filas contienen la información asociada a emisiones GPS de buses. Cada fila corresponde a una emisión GPS.

Dentro de la carpeta "zonas" encontrará una serie de archivos. Estos deben llamarse todos de la misma forma y estar siempre bajo la misma carpeta. El archivo principal se llama **zonas.shp**. Del mismo modo, en la carpeta "paradas", encontrará el archivo **paradas.shp**.

Cada archivo, emisiones, zonas y paradas, cuentan con una tabla de información.

Emisiones

Cada columna (o atributos) que posee una emisión (de emisiones.csv) se describen a continuación:

1. **measurement_id**: identificador interno de una medición GPS. Este valor corresponde a un número hexadecimal de largo fijo.
2. **expedition_id**: identificador interno de la expedición asociada a la medición GPS. Una expedición corresponde a la realización de un recorrido por un bus. Este valor corresponde a un número hexadecimal de largo fijo.
3. **dispatch_time**: instante de tiempo en que ocurre el despacho de la expedición asociada a la medición GPS. Este valor se encuentra representado en el formato `año-mes-dia hora:minuto:segundo`, por ejemplo, "2018-04-01 14:33:21".
4. **line_id**: identificador interno de la línea del bus. La línea de un bus se refiere a servicio prestado por el sistema. Este valor corresponde a un número hexadecimal de largo fijo.
5. **line_code**: código usuario de la línea. Este corresponde al código que llevan los buses en su parte frontal. Se encuentra represando por el código, por ejemplo "C02".
6. **direction**: sentido de operación de la línea. Este campo indica si la operación es en el sentido "ida", representado por una **I** o "regreso", representado por una **R**.
7. **bus_id**: identificador interno del bus. Este valor corresponde a un número hexadecimal de largo fijo.
8. **license_plate**: placa patente del bus. Por ejemplo "BFRD-27" o "FG-3241".
9. **bus_capacity**: capacidad física máxima del bus. Representado con un número que representa la cantidad de personas que caben en el bus.
10. **gps_time**: instante de tiempo en que ocurre la medición GPS. Este valor se encuentra representado en el formato `año-mes-dia hora:minuto:segundo`, por ejemplo, "2018-04-01 14:33:21".
11. **latitude**: latitud de la medición GPS.
12. **longitude**: longitud de la medición GPS.

13. **distance_kms**: distancia recorrida por el bus desde el despacho hasta el instante de la emisión GPS en kilómetros.

14. **total_kms**: distancia total a recorrer por el bus en kilómetros.

15. **measurement_speed**: estimación de la velocidad para el instante de la medición GPS, basada en la emisiones recientes en km/h.

16. **instant_speed**: velocidad instantánea al momento de registrar la medición GPS en km/h.

Zonas

Cada columna (o atributos) que posee una zona (de zonas.shp) se describen a continuación:

1. **id**: identificador interno de una figura dentro de zonas.shp. Este valor corresponde a un número entero.

2. **AREA**: área de la figura en km^2.

3. **AREA1**: área de la figura en m^2.

4. **PERIMETER**: perímetro de la figura en m.

5. **COMUNAS_ID**: identificador interno de la comuna asociada a la figura. Este valor corresponde a un número entero.

6. **COMUNA**: nombre de la comuna asociada a la figura.

7. **ZONA**: identificador de la zona Transantiago asociada a la figura. Este valor corresponde a un número entero.

8. **ZONA_TS**: nombre de la zona Transantiago asociada a la figura.

Paradas

Cada columna (o atributos) que posee una parada (de paradas.shp) se describen a continuación:

1. **id**: identificador interno de una parada dentro de paradas.shp. Este valor corresponde a un número entero.

2. **CODINFRA**: código interno de la parada utilizado por el área infreaestructura en DTPM.

3. **SIMT**: código usuario de la parada, visible en los letreros de paradas.

4. **NOMBRE_PAR**: nombre del paradero

5. **FREPMA**: frecuencia de buses en el paradero en *buses/hora* para punta mañana.

6. **FREPTA**: frecuencia de buses en el paradero en *buses/hora* para punta tarde.

7. **NSERVICIOS**: cantidad de servicios sentido que se detienen en la parada.

8. **SERVICIOS**: lista de servicios sentido que se detienen en la parada (separados por ";").

Misiones

Para este último desafío, usted deberá completar una serie de misiones utilizando las librerías **pandas**, **geopandas**, **matplotlib**, **urllib** y **bs4**. Los desarrollos deben ser realizados en **Google Colab** aunque si usted logra instalar correctamente las librerías en su computador, puede utilizarlo. Con el fin de facilitar la corrección, programe cada misión en un celda aislada (puede usar más de una celda por misión, pero no combinar desarrollos de varias misiones en una misma celda). Indique con un comentario al inicio de cada celda la misión que se trabaja en dicha celda.

```
#Mision X
#acá va su desarrollo
#el output de la celda debe ser lo que se pida en cada misión
```

Donde "x" es el número de la misión a responder. La respuesta debe poder visualizarse directamente en Python. A continuación se describen las misiones que deberá completar:

M1. Su primera misión será trasladar los archivos necesarios hacia Google Colab. Una vez con los archivos cargados, puede verificarlos ejecutando el comando `!ls` en una celda. Su objetivo será visualizar el *dataframe* presente en el archivo "zonas.shp" y luego mostrar el contenido geográfico, todo esto mediante el uso de **geopandas**. ¿Qué hay de diferente en el *dataframe*? No es necesario que responda esta pregunta. **Output esperado:** visualización de 5 filas del *dataframe* y la vista gráfica. (0.5 pts)

M2. Realice exactamente lo mismo que la misión anterior para el archivo "paradas.shp". Luego haga una visualización conjunta de ambos archivos *shapes*. Le recuerdo que está trabajando con *dataframes*, por lo tanto, puede utilizar la función *plot* (como la ha usado antes) para llevar a cabo esta misión. **Output esperado:** visualización de 5 filas del *dataframe* de paradas, del archivo gráfico de paradas y de la visualización conjunta. (0.5 pts)

M3. ¿Ha notado que existen paradas fuera de las zonas definidas? Su misión será eliminarlas. Para ello debe utilizar el método *sjoin* de **geopandas** que le permite solucionar este problema. Busque ese método en la materia del curso o en internet y encuentre la manera de llevar a cabo esta misión. **Output esperado:** visualización gráfica conjunta de las zonas y las nuevas paradas filtradas geográficamente. (1.0 pt)

M4. Volvamos a nuestra antigua base de datos, las emisiones GPS. En esta misión solo debe realizar un filtro común. Utilizando la base de datos mediana (1 semana) de emisiones GPS, proceda a filtrar el *dataframe*, quedándose sólo con los datos de un servicio sentido. Es libre de escoger el servicio sentido que desee. **Output esperado:** *dataframe* de emisiones filtrado para un único servicio sentido. **NOTA: para su entrega parcial puede utilizar las emisiones pequeñas**. (0.5 pts)

M5. A continuación de la misión anterior, tome su nuevo *dataframe* filtrado y transfórmelo en un *geodataframe*. Recuerde que la única diferencia entre un *dataframe* y un *geodataframe* es la inclusión de una columna geográfica. Para llevar a cabo esta transformación, utilice las columnas *latitude* y *longitude* de las emisiones para crear una nueva columna de puntos geográficos. Básese en la materia del curso. Luego visualice conjuntamente las zonas y las emisiones GPS convertidas. **Output esperado:** *dataframe* de emisiones geográfico y visualización conjunta de emisiones GPS y zonas. (1.0 pt)

M6. En esta misión deberá crear una nueva columna en su nuevo *geodataframe* de emisiones GPS. Esta nueva columna debe indicar la *ZONA_TS* en la que ocurrió tal medición. Es decir, debe agregar la información de la columna *ZONA_TS* (del shape de zonas) en las emisiones que ocurrieron en dicha zona. Para ello puede utilizar funciones de **geopandas** o usar lógica geométrica con ayuda de Python. **Output esperado:** *geodataframe* de emisiones con la columna *ZONA_TS*. (1.5 pts)

M7. En esta misión, usted deberá construir el trazado utilizando las emisiones GPS determinadas en la misión anterior. Hasta ahora, hemos trabajado con polígonos y puntos, es hora de que usted logre crear un nuevo objeto de líneas. Primero ordene las emisiones GPS (sin importar cuando ocurrieron) desde la más cercana a la más lejana del lugar de despacho del bus. Luego, utilice este orden para crear una línea geográfica a partir de los puntos geográficos de las emisiones. Busque el uso de la librería *shapely* (Esta es la que usa *geopandas* para trabajar con geometrías). **Output esperado:** Objeto LineString con la concatenación de puntos GPS. (1.0 pt)

M8. Repita las misiones 4, 5, 6 y 7 para el servicio de dirección opuesta al trabajado. Con los trazados de ambas direcciones construya un *geodataframe*. **Output esperado:** Objeto LineString con la concate-

nación de puntos GPS para ambos sentidos de un servicio y creación de un *dataframe* con las columnas: "servicio", "sentido" y "geometría". (1.0 pt)

M9. Filtre los paraderos del *shape* de paradas ("paradas.shp") para visualizar solo las paradas relacionadas con el servicio (ida y regreso) del shape resultante de la misión anterior. Luego, visualícelos de forma conjunta. **Output esperado:** Filtro de paradas y visualización conjunta de *shapes* de trazados y paradas asociadas. (1.0 pt)

M10. En esta nueva misión, deberá completar información presente en internet. Para ello entre en el siguiente link de Wikipedia. Dentro de la web, encontrará una tabla de la conurbación de Santiago. Basado en los nombre de comunas presentes en el *shape* de zonas y con la ayuda de las librerías **bs4** y **urllib** extraiga de la página web, la información de viviendas y población presentes en la tabla. Luego, incorpórelas al *shape* de zonas. **Output esperado:** *shape* de zonas con información de internet. (2.0 pts)

M11. Utilizando este nuevo *shape* (de la misión 10) y el *shape* completo de paradas, determine el número de paraderos por comuna (añadir a una nueva columna). Con la ayuda de los indicadores $paraderos/persona$ y $paraderos/vivienda$, ¿Es homogénea la distribución de paraderos en la ciudad? Visualice gráficamente las zonas con ambos indicadores y con la cantidad de paraderos (3 visualizaciones), esta vez, que el polígono sea mas obscuro si posee un indicador mayor y más claro si posee un indicador menor. **Output esperado:** *shape* de zonas con nuevas columnas de conteo de paraderos, paraderos_persona y paraderos_vivienda. Además, la visualización de los indicadores. (1.0 pt)

M12. Busque en internet alguna otra información comunal, léala y añádala a su tabla. luego haga una visualización interesante. **Output esperado:** *shape* de zonas con alguna nueva información interesante y su respectiva visualización. (1.0 pt)

Corrección

Para la corrección de este laboratorio, se revisarán las misiones en base a los **outputs esperados**.

Primer avance parcial (final del la clase del 10/06)

1. Responda las misiones 1, 2 y 3

Segundo avance parcial (final del la clase del 17/06)

1. Responder hasta la misión 4, 5 y 6.

Entrega final

1. Responda todas las misiones del enunciado

Política de Integridad Académica

"Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad."

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por "copia" o "plagio" se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.