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

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

Laboratorio 9 - Análisis y visualización de datos

Objetivos

- Aplicar los contenidos de análisis y visualización para estudiar, graficar y predecir propiedades o relaciones que se pueden observar en en conjunto de datos.

Entrega

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **L09**.
- **Primera entrega parcial: lunes 27 de mayo a las 16:50 hrs.**
- **Segunda entrega parcial: lunes 3 de junio a las 16:50 hrs.**
- **Entrega final: domingo 9 de junio a las 23:59 hrs.**
- **Formato de entrega:** archivo python notebook (**.ipynb**) con la solución, ubicado en la carpeta **L09**. 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

Estudio de datos de Transantiago

En el desarrollo de los laboratorios 7 y 8, usted ha estado trabajando con bases de datos SQLite para determinar información específica de los GPS de Transantiago.

Esta vez, usted propone a su empleador (DTPM) utilizar otra herramienta llamada **Pandas**, que le permite de igual forma que SQLite, procesar información pero con algunas ventajas. Además usted promete utilizar otras librerías de visualización y predicción de datos. DTPM acepta su oferta y quiere saber los límites de su potencial en el uso de estas herramientas. Cabe destacar que el uso de **Pandas** simplifica muchas operaciones, pero siempre es bueno tener de la mano conocimientos de base de datos, sobre todo en operaciones no cubiertas por **Pandas**.

El *set* de datos

En este laboratorio seguiremos trabajando con los mismos datos de los dos laboratorios pasados. Estos se encuentran en el link. Recuerde que va a encontrar tres carpetas. Cada una de ellas posee 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 y las columnas (o atributos) que posee una emisión 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.

Misiones

Para este nuevo desafío, usted deberá completar una serie de misiones utilizando las librerías **pandas**, **matplotlib** y **sklearn**. Al responder las misiones, puede usar funciones de estas librerías u otra librerías de análisis de datos. Recuerde que si utiliza nuevas librerías, debe dejarlas especificadas y justificadas en el archivo `README.md`. 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 importar los datos mediante la librería **pandas**. Para ello debe asegurar que su computador posee la librería pandas instalada. Aproveche el conocimiento que posee de los datos para entregar información extra a la función de importación. Es decir, si los decimales están con punto o coma; o qué símbolo se utiliza como separador de datos. En esta misión se espera que cree un *dataframe* (objeto de **pandas**). Para asegurar su importación, utilice el método *head* con el fin de mostrar las 5 primeras filas de datos. **Output esperado:** visualización de 5 filas del *dataframe*. (0.5 pts)

M2. Según la materia, existe una función que le permite analizar estadísticamente las variables numéricas de un *dataframe*. Utilícela para comentar si existen errores o inconsistencias dentro de los datos, sea breve y preciso. **Output esperado:** *dataframe* con información estadística de variables numéricas y análisis breve. (0.5 pts)

M3. ¿Cuántas emisiones GPS se realizan por servicio? Utilice una función de **pandas** que le permita responder rápidamente esta pregunta. Con la información obtenida indique el porcentaje del total de emisiones GPS para cada servicio de forma genérica. Esto último quiere decir que, su código debe funcionar incluso si cambiamos el conjunto de datos. **Output esperado:** lista de tuplas con los porcentajes de emisiones (servicio, porcentaje). (1 pt)

M4. Construya un gráfico que permita identificar los servicios (*line_code*) que poseen la mayor dispersión de la velocidad instantánea (*instant_speed*). **Output esperado:** Gráfico descrito en la misión. (1 pt)

M5. Investigue el uso de la función *loc* en pandas. Esta función le permite seleccionar un subconjunto de datos que cumplen con ciertas condiciones. Construya un nuevo *dataframe* utilizando la función *loc* solo con emisiones GPS de una expedición particular para un servicio dado. **Output esperado:** Defina una función que reciba *line_code*, *direction* y número de la expedición (número de expedición desde la primera presente en los datos para ese servicio sentido) y retorne un *dataframe* con el filtro realizado. (1 pt)

M6. Grafique la trayectoria de una expedición. Al usar la función definida en la misión anterior, tendrá un *dataframe* con todas las emisiones GPS producidas por una expedición de un servicio sentido. Considere que la fecha-hora del despacho es el **minuto 0**, utilice las columnas *distance_kms* y *gps_time* para graficar una curva x [km] vs t [min] de la trayectoria de la expedición. **Output esperado:** Defina una función que reciba *line_code*, *direction* y número de la expedición, y que muestre el gráfico de la trayectoria. **BONUS:** En vez de recibir el números de la expedición, reciba una lista de número de expediciones y grafique en un mismo objeto, todas las trayectorias de la lista de expediciones. (1 pt)

M7. Determine las variaciones de velocidad en la trayectoria de un bus. Para llevar a cabo esta misión, utilice la misión anterior para construir la trayectoria real de un bus. Ahora, construya una trayectoria para un bus ficticio (para el mismo rango de tiempo) a velocidad constante igual a la velocidad media del bus real. Al construir esta última curva, debe hacer uso de una función ***lambda***. Grafique ambas curvas en un mismo objeto (la trayectoria real y la trayectoria a velocidad constante). **Output esperado:** Defina una función que reciba *line_code*, *direction* y número de la expedición, y que muestre el gráfico con ambas curvas. (1 pt)

M8. Se requiere hacer un análisis de los despachos y tiempos de viaje. Para llevarlo a cabo, es necesario contar con un *dataframe* específico. El objetivo de esta misión será contar con un *dataframe* que posea la información relevante para estudiar los tiempos de viaje de las expediciones. Para eso, con la ayuda de la función *loc* filtre los datos para obtener solo la última emisión GPS de cada expedición. Recuerde que estas poseen el tiempo de despacho y el tiempo de la última emisión. Genere un sub *dataframe* que poseea solo las columnas *expedition_id*, *line_code*, *direction*, *license_plate*, *bus_capacity*, *total_kms*, *dispatch_time* y *gps_time*. **Output esperado:** Un nuevo *dataframe* filtrado con las columnas presentandas al final de la mision. Muestre 5 filas del *dataframe* con la función *head*. (0.5 pts)

M9. Investigue cómo incorporar nuevas columnas basadas en relaciones con las demás. Basándose en la hora de despacho (*dispatch_time*) y la última emisión GPS (*gps_time*) de cada expedición, cree las columnas *week_day*, *reception_time*, *travel_time* y *total_mean_speed* que representan: el día de la semana del despacho ("L", "M", "W", "J", "V", "S" y "D"), la hora de término de una expedición (es igual al último *gps_time* de una expedición), tiempo total de viaje (diferencia entre *reception_time* y *dispatch_time* en minutos) y la velocidad media de todas las expediciones (*total_kms* sobre *travel_time*), respectivamente. **Output esperado:** Nuevo *dataframe* con las nuevas columnas presentandas al final de la mision. Muestre 5 filas del *dataframe* con la función *head*. (1 pt)

M10. Para un servicio en específico. ¿Qué día se producen más despachos?. En rangos de dos horas para días

laborales ¿Qué horario presenta la mayor cantidad de despachos? Apóyese de gráficos que permitan apoyar su respuesta. **Output esperado:** Construya una función que reciba el servicio (*line_code*) y retorne el día con más despachos y el horario laboral (rango de dos horas) con más despachos. Además muestre las gráfica que su función utiliza para decidir. **HINT:** Agregue los días/horarios antes de calcular. (1 pt)

M11. Para un servicio en específico, en rangos de dos horas ¿Cómo distribuyen los tiempos de viaje en días laborales? ¿Cómo distribuyen los tiempos de viaje en fin de semana? Apóyese de gráficos que permitan apoyar su respuesta. **Output esperado:** Construya una función que reciba el servicio (*line_code*) y muestre las distribuciones de tiempos de viaje. **HINT:** Agregue los horarios por días antes de calcular. (1 pt)

M12. Llegó la hora de hacer algunas predicciones. Basándose en los capítulos **1.2.2.- Limpieza y depuración de los datos** y **1.2.3.- Construcción de modelos predictivos** de la materia del curso, su misión será realizar algunas predicciónes. Usando la información presente en la misión 9, genere un modelo de predicción de tiempo de viaje. Es libre de definir su modelo y variables a considerar, crear nuevas columnas, qué variables son independientes, cuáles son variables dependientes, si quiere predecir el tiempo de viaje a nivel de días, horas, rango de horas, servicio, servicio sentido, etc. **Output esperado:** Revisión de consistencia de datos a utilizar, posibles depuraciones y evaluación para completar datos faltantes. Además de la ejecución de un modelo predictivo que prediga el tiempo de viaje. (1.5 pts)

M13. Del mismo modo que la misión anterior, prediga alguna otra variable que le parezca interesante utilizando el mismo *dataframe* resultante de la misión 9. **Output esperado:** Revisión de consistencia de datos a utilizar, posibles depuraciones y evaluación para completar datos faltantes. Además de la ejecución de un modelo predictivo que prediga la variable que le pareció interesante. (1 pt)

IMPORTANTE: Para las misiones 12 y 13. Procure entregar en una celda:

```
#Mision X
# Predicción: Mencione específicamente que va a predecir.
# Desarrollo
# Ejecución del modelo
```

Corrección

Para la corrección de este laboratorio, se revisarán las misiones en base a los **outputs esperados**. Todas las misiones que impliquen el retorne de un valor, use `return` y no `print`.

Primer avance parcial (final del la clase del 27/05)

1. Responda las misiones que serán trabajadas en clases

Segundo avance parcial (final del la clase del 03/06)

1. Responder hasta la misión 10 (pensando en 8, 9, 10 de trabajo en clases)

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.