

Tarea Grande 1

Profesor Denis Parra
Anunciada: 1 de abril de 2019

Indicaciones

- Fecha de Entrega: 15 de abril de 2019.
 - Debes entregar tu tarea en tu repositorio GitHub privado asignado para esta evaluación.
 - Cada hora o fracción de atraso descuenta 0.5 puntos de la nota que obtengas.
 - La tarea es *individual*. La copia será evaluada con nota 1 en el la tarea, además de las sanciones disciplinarias correspondientes.
 - La tarea consta de dos partes las cuales poseen sus puntajes detallados en las descripciones de estas.
-

Objetivo

El objetivo de esta tarea es que aprendas a:

- Utilizar Python para manipular datos.
- Utilizar los datos para poder hacer gráficos con la librería Altair.
- Utilizar HTML+CSS+JS para construir una página web sencilla en la que se muestren las visualizaciones construidas.

Descripción de los datos

Archivos

En tu repositorio de la tarea, encontrarás un archivo denominado `data-iic1005-2019.csv`. Este fichero contiene información sobre 1559 eventos de robo de vehículos. Sus columnas son las siguientes:

1. **id**: número de identificación único para cada evento.
2. **fecha_siniestro**: fecha del robo de este vehículo. Puede estar en distintos formatos.
3. **tipo_automovil**: clasificación del vehículo que fue robado.
4. **marca**: marca del vehículo siniestrado.
5. **modelo**: modelo del vehículo siniestrado.
6. **lat**: coordenada de latitud de la ubicación del robo.
7. **lng**: coordenada de longitud de la ubicación del robo.

Instrucciones

Parte 1: manipulación y visualización de datos (3,5 puntos)

En esta parte de tu tarea trabajarás con los datos entregados para obtener lo necesario para la confección del *dashboard*. **El entregable de esta sección es un *jupyter notebook* (extensión *.ipynb*)**, donde se aprecie el **código**, **gráficos realizados** y **explicaciones** que te parezcan pertinentes. De no cumplir con el tipo de archivo optarás a **nota máxima 4** en tu tarea. Además, se deberá utilizar **obligatoriamente** la librería **Pandas** para el procesamiento de los datos, de lo contrario estarás optando a **nota máxima 4**. Cabe señalar que si no usas esta librería de forma eficiente (usando loops innecesarios) podrás llegar a tener hasta **5 décimas de descuento**. Por su parte, los gráficos deben realizarse **obligatoriamente** con la librería **Altair**. Si no utilizas la librería anteriormente mencionada, **no obtendrás puntaje en los ítems relacionados con visualización**. En particular, debes realizar lo siguiente:

1. Como se mencionó más arriba, la fecha de los siniestros está en distintos formatos. Por ejemplo, algunos siniestros tienen la fecha como DD-MM-YYYY HH:mm y otras como DD/MM/YYYY HH:mm. Deberás hacer que **todas** las fechas tengan el formato DD-MM-YYYY HH:mm **(1 punto)**.
2. Generar un **gráfico de línea** con la cantidad de robos realizados según el día de la semana (Lunes, martes, miércoles, etc.) **(1 punto)**. **Importante**: si realizas este gráfico con datos falsos, es decir, no con los procesados en el punto anterior, tendrás un **descuento de 0.5 puntos en este ítem**.
3. Confeccionar un **gráfico de barras horizontales** donde en el eje *y* se encuentren los meses y en el *x* la cantidad de robos que hubo en cada uno de ellos. Cada barra debe ser de un color diferente a las demás. **(0.5 puntos)**.

4. Construir dos **gráficos de barras verticales**: el primero con los 5 tipos de automóvil más robados y otro con las 5 marcas de automóviles más robadas. Cada barra de cada gráfico debe tener un color diferente a las demás. Para que estos gráficos resulten más agradables a la vista, deberás *capitalizar* los tipos. Esto quiere decir que si el tipo de auto es "CAMIONETA", se espera que en el gráfico aparezca como "Camioneta". Lo mismo debe cumplirse para las marcas de auto. **(0.5 puntos por gráfico)**. Adicionalmente, podrás usar `matplotlib` para realizar los gráficos de torta que estaban considerados en la primera versión de este enunciado (y no hacer los gráficos de barra verticales que se proponen en esta versión). Ojo, solo puedes usar `matplotlib` para realizar gráficos de torta. **Todos los demás gráficos deben ser realizados con Altair.**

Es importante destacar que los gráficos deben, **obligatoriamente**, contar con un título, leyenda (si amerita) y nombres de los ejes. **Cada elemento de estos que falte representará un descuento de 0.05 puntos.**

Habrás descuentos si es que en lugar de usar una función de pandas haces *loops* y funciones que hacen el código menos eficiente. Un ejemplo: En pandas puedes automáticamente sacar la media de una columna de datos de un `DataFrame` con la función `mean()`. Si en lugar de usar `mean()` haces un *loop* en Python, habrá descuento.

Para poder utilizar los gráficos en la Parte 2, estos deben ser exportados a un archivo de imagen.

Parte 2: *dashboard* (2.5 puntos)

Con las visualizaciones ya generadas, debes construir una página web que contenga cada uno de los gráficos mencionados en el enunciado. Como la confección de los gráficos será evaluada en la parte 1, en esta parte se espera que cumplas con lo siguiente:

1. Mapa con pines: seguramente notaste que el archivo `data-iic1005-2019.csv` contiene información sobre la latitud y longitud de la ubicación de ocurrencia de los siniestros. Con el objetivo aprovechar esta información para determinar los lugares donde existen más robos, se pide que, utilizando la librería de JavaScript `Leaflet` o la API de Google Maps, dispongas en un mapa cada uno de los delitos reportados en el archivo `.csv` que se entregó junto a este enunciado. Al hacer click en un *pin* en el mapa, deben mostrarse la marca y el modelo del vehículo involucrado, por ejemplo: SUZUKI GRAND VITARA 1.6 GLX AC **(1,5 puntos)**.

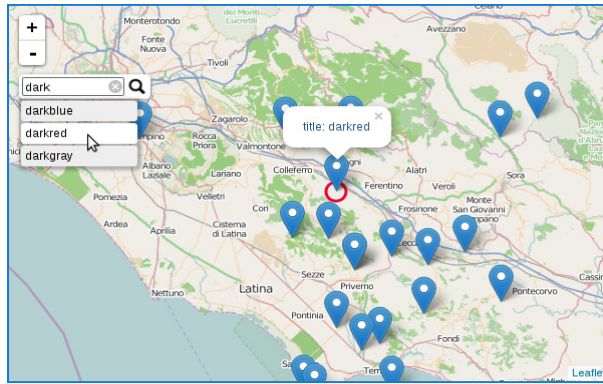


Figura 1: Mapa de referencia con pines según coordenadas en Leaflet (el menú mostrado en la imagen es totalmente opcional).

2. *Clusters* de pines: En un nuevo mapa deberás hacer que los pines aparezcan inicialmente agrupados en pequeñas secciones, que se vayan expandiendo conforme el usuario hace *zoom*. Un ejemplo de esto puede apreciarse en la siguiente imagen (**1 punto**):

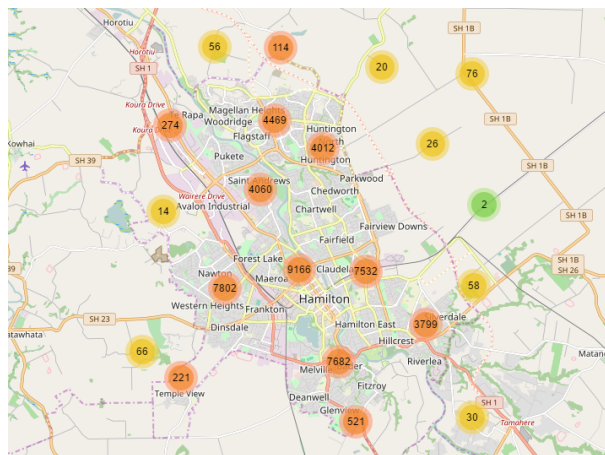


Figura 2: Ejemplo de pines agrupados en *clusters*.

Es importante notar que se esperan dos mapas en esta sección: uno con los pines sin agrupar y otro con los marcadores agrupados en *clusters*.

¡Bonus! (0,5 puntos)

Si la tarea te resultó especialmente motivadora y quieres seguir aprendiendo, te proponemos los siguientes desafíos, que permitirán que puedas obtener como **nota máxima un 7.5**:

1. Página *responsive* (**0.25 puntos extra**): al cambiar el tamaño de la ventana del navegador, tu página web debe alterar la alineación de su contenido, para adaptarse al tamaño de la ventana. Eres libre de utilizar librerías de CSS que faciliten esta tarea.
2. Página ordenada (**0.25 puntos extra**): los gráficos y mapa que obtuviste de las partes anteriores deberán ir con una breve descripción y con la siguiente distribución:

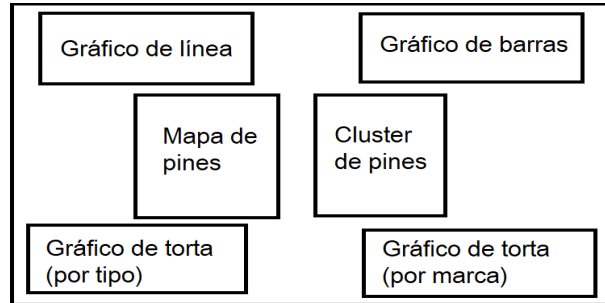


Figura 3: Distribución de los gráficos

Formato de entrega

Para evaluar con facilidad la parte 1 de esta tarea, deberás realizarla en un *jupyter notebook* y subir este archivo de extensión `.ipynb` a tu repositorio asignado. En cuanto a la parte 2, debes incluir **todos** los archivos que permitan visualizar tu página correctamente. En el repositorio base de la tarea, encontrarás una estructura propuesta que te ayudará a cumplir con el formato esperado.