Actividad 0 - Navegando Datos

ExactasPrograma - Datos

Invierno 2020

¿Qué es lo primero que necesita alguien para llamarse investigador en datos? ¡Datos! Vamos a utilizar las respuestas que ustedes completaron en sus formularios de inscripción al curso. En este link pueden encontrarlas.

Vamos a trabajar en un *Notebook*, un espacio de trabajo donde podemos escribir código y texto, y ejecutar fácilmente las porciones de código que elijamos. Para poder trabajar en equipo, utilizaremos una versión colaborativa, la plataforma Google Colab. Clickeando en `New Notebook`, obtendremos un notebook en blanco sobre el cual podemos empezar a escribir nuestros programas.

Uno de los fuertes de un notebook es poder escribir bloques que son de código y bloques de texto. Sugerimos utilizar un bloque (o más) de código para cada ejercicio y utilizar bloques de texto para describir las secciones, escribir conclusiones y reflexiones sobre el código que se ejecuta. Estas características son ideales para realizar y documentar análisis todo en el mismo lugar.

El objetivo de esta actividad será analizar nuestros primeros datos, usando `Python` para responder algunas preguntas de manera ordenada en un notebook.

Utilizaremos dos nuevos módulos además de los conocidos `numpy` y `matplotlib`:

- Por un lado, `pandas`[1], que nos va a servir para *almacenar y manipular nuestros datos.*
- Por otro, `seaborn`[2], que nos permite *graficar* de manera sencilla la información.

Para utilizarlos, podemos importarlos ejecutando en nuestro entorno de `Python`:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

`pandas` nos ofrece dos nuevos tipos de datos: *Series* (como las listas que conocemos, pero con algunas funcionalidades extra) y *Dataframes*, que permiten almacenar tablas de información.

Un recurso muy útil al iniciarse con un lenguaje, biblioteca o recurso nuevo son las CheatSheets (hojas de trucos, machetes). Hay muchas versiones en Internet. En el campus del curso hay algunas que pueden resultar útiles a lo largo de este curso. Igualmente, siempre se puede buscar en Internet documentación, ejemplos, y tutoriales.

Importante: En este actividad utilizaremos las respuestas que suministraron en la encuesta de inscripcion. Las preguntas que se plantean se refieren apenas a la base de datos con la que trabajamos. **No se pretende extrapolar ni inferir comportamientos en otras poblaciones.**

1. **Obtener los datos**. Vamos a levantar los datos que vienen en formato CSV (Comma-separated values). Es un formato bastante estándar y `pandas` tiene la función `read_csv()` para leerlos.

   Si escribimos `df = pd.read_csv(ORIGEN)`, `pandas` va a leer el contenido de lo que indiquemos en `ORIGEN` (puede ser un archivo que suban al Colab, o simplemente una URL); y lo guardará en un `dataframe` llamado `df`. Probarlo para los datos propuestos, y luego utilizar `display(df)` para imprimirlo.

   ¿Qué estructura tiene? ¿Qué información guarda cada fila?

[1] https://pandas.pydata.org/docs/
[2] https://seaborn.pydata.org/

2. Utilizar `df.columns` para obtener la lista de columnas. ¿Qué información guarda cada una?

3. Explorar los comandos `df.describe()` y `df.info()` y responder:

   - ¿Cuál es la altura más alta registrada? ¿Y la más baja?
   - ¿Cuál es la edad promedio?
   - ¿Hay algún dato que *falte*?

4. Obtener la columna que contiene las *alturas*. Escribir un código que calcule la altura promedio, la más alta y la más baja.

   Ayuda: `df.mean()` calcula el promedio de cada columna del *dataframe* `df`. Explorar funciones similares para calcular el mínimo y el máximo de cada columna.

5. En el caso anterior, ¿cómo maneja `pandas` los casos faltantes (*missing values* en ingles). ¿Está bien su manejo? ¿Hace "trampa" borrándolos?

6. Calcular el peso promedio.

7. Responder:

   - ¿Qué edad tiene el estudiante más joven?
   - ¿Y el más grande?
   - ¿De qué signo del horóscopo chino es cada uno?

   Ayuda: Podemos ordenar las filas de un *dataframe* ejecutando `df.sort_values('nombre_columna')`

8. Filtrar[3] el conjunto de datos para quedarte únicamente con las respuestas de estudiantes de tu misma carrera. Responder:

   - ¿Cuántos son en total?
   - ¿Qué edad tienen en promedio?
   - ¿De qué cuadro es el que tiene mayor promedio?
   - ¿Cuál es el promedio general (o sea, el promedio entre los promedios)?

9. Generar una lista de todas las carreras. Para cada una, calcular su promedio general.

   - ¿Cuál es la carrera con mayor promedio?
   - ¿Y cuál es la que tiene el más bajo?
   - ¿Alcanza esto para poder afirmar que los estudiantes de la primera carrera se esfuerzan más que los de la segunda?
   - ¿Alcanza esto para poder afirmar que los estudiantes de la primera carrera tienen mayores promedios que los de la segunda?

10. Filtrar el conjunto de datos para quedarte con las personas que son más altas **y** tienen peor promedio que vos. ¿Cuántas son?

11. Repetir el ejercicio anterior, pero ahora con las personas que son más altas **o** tienen peor promedio que vos. ¿Cuántas son?

12. Para los datos considerados, ¿es cierto que...

    - ... los hinchas de Boca son la mitad más uno?
    - ... los estudiantes de biología son la mitad más uno?
    - ... los *monos* en el horóscopo chino tienen mejor promedio que el resto?

[3] *filtrar* o *seleccionar* son expresiones usadas para referirse al subconjunto de los datos que cumple alguna condición.

- ... las *gallinas* (o gallos) en el horóscopo chino tienen mayor predilección por River?

13. Nos gustaría ahora mostrar cuántos hinchas tiene cada equipo, y una buena opción para ello es utilizar un gráfico de barras. La función de `seaborn` `sns.countplot(data=mi_dataframe, x='nombre_col')` genera un gráfico de este tipo, contando en el dataframe `mi_dataframe`, cuántas filas tienen cada valor posible en la columna llamada `nombre_col`.

    Explorar el uso de esta función para mostrar la cantidad de hinchas de cada equipo.

    Ayuda: Si las etiquetas de tu eje x se solapan, podés colocar en el mismo bloque el siguiente comando: `plt.xticks(rotation=90)`.

14. Repetir el ejercicio anterior, ahora mostrando cuánta gente es de cada signo del horóscopo chino.

15. En el ejercicio 9 calculamos los promedios generales de cada carrera. Podemos, para visualizar mejor esta información, utilizar un gráfico de barras. Para ello, podemos utilizar nuevamente `seaborn`, completando los datos correspondientes en esta función `sns.barplot(data=..., x=..., y=...)`

16. Cuando une quiere analizar la distribución de alguna característica en una población, puede utilizar los llamados **histogramas**. Este tipo de gráficos particiona todos los posibles valores en intervalos discretos, y muestra qué porcentaje de la población observada tiene dicha característica dentro de cada uno.

    Utilizar `sns.distplot(<datos>, kde=False)` para graficar un histograma de las alturas reportadas. ¿Se corresponde con lo que esperaban? ¿Qué anomalidades se observan?

17. Plantear una regla para corregir los datos mal ingresados que observaste en el ejercicio 16. Modificar la columna correspondiente del dataframe, aplicando esta regla.

18. ¿Está bien el resultado que reportaste en el ejercicio 6? Si tu respuesta es sí, realizar un histograma de los datos. Una vez que tu respuesta sea no, corregí los datos y volvé a responder la pregunta original.

19. Realizar, para cada género reportado, un histograma como el del ejercicio anterior. ¿Qué conclusiones podés obtener?

20. Repetir el ejercicio 16, ahora considerando los promedios reportados. Agregar a la función el argumento `bins=<cantidad_de_bins>` para modificar la cantidad de intervalos considerados. Mirando el gráfico que generaste, ¿qué podés decir respecto a los promedios?

    ¿Qué sucede si cambiás el valor del parámetro `kde`?

21. Realizar un `sns.scatterplot` con la altura en un eje y el promedio en el otro. ¿Se puede observar alguna relación?

22. Repetir el ejercicio anterior pero eligiendo, en lugar del promedio, una columna que creas que correlaciona con la altura. ¿El gráfico acompaña tu hipótesis?

    ¿Qué sucede si en estos dos últimos gráficos se clasifica a los puntos según 'Genero'? ¿Y según 'Avance'?

    Ayuda: Utilizar el argumento `hue` de dichos gráficos.

23. ¿Existe alguna relación entre la carrera y el promedio? ¿Y entre el avance en la carrera y alguna de ellas? Utilizar las funciones `sns.catplot`, `sns.swarmplot`, `sns.boxplot` y `sns.violinplot` para intentar responder esas preguntas.

24. En ejercicios anteriores, realizaste gráficos filtrando los datos según el valor de alguna(s) de sus columnas: el género, la carrera, el cuadro y otras. `seaborn` nos permite, a través de la función `sns.FacetGrid`, generar múltiples gráficos de manera sencilla, cada uno según una (o más) categorías.

    - Utilizar `grid = sns.FacetGrid(df, col='nombre_columna', margin_titles=True)` para crear una grilla de gráficos, uno para cada género en el dataframe. ¿Qué sucede si lo ejecutamos?

- Para cada uno de ellos, necesitamos rellenarlo con algún tipo de gráfico. Para ello, podés ejecutar `grid.map(<tipo_grafico>, "columna_datos")`, donde `<tipo_grafico>` es cualquiera de las funciones que utilizamos en esta guía (como por ejemplo `sns.distplot`)

Realizar una grilla de gráficos, en cada uno mostrando un histograma de las alturas de las personas de un género.

25. (*Optativo 1*): La idea a continuación es intentar contestar la siguiente pregunta: ¿hay alguna diferencia entre la distribución de promedios de las carreras de cada pabellón? Para ello:

    - Primero armar una lista `pabellon_1` que contenga los nombres de las carreras que se cursan en el Pabellón 1 de la Facultad, y otra análoga `pabellon_2`.
    - Luego, definir una función `dame_pabellon(carrera)` que, dado el nombre de una carrera, devuelva el número de su pabellón correspondiente, y '0' si no corresponde a ninguna carrera de la facultad.
    - Finalmente, agregar al dataframe **una nueva columna** llamada `'Pabellon'`, que para cada alumno contenga el número de su pabellón.

      Ayuda: `df['Carrera'].apply(dame_pabellon)` devuelve una serie con el resultado de aplicar la función `dame_pabellon` a cada elemento de `df['Carrera']`

    Realizar una grilla de gráficos, mostrando la distribución de promedios para cada pabellón. ¿Siguen la misma distribución? ¿Qué diferencias observa?

26. (*Optativo 2*: ¿Quiénes estiman mejor?) Cada persona realizó una estimación del peso del **toro** de la imagen.

    a) Calcular el promedio de estimaciones de les estudiantes de cada carrera. ¿Cuál resultó más cercana al valor real?

    b) Realizar una grilla de gráficos, uno por cada pabellón, con la distribución de las estimaciones. ¿Qué particularidades observa?

    c) Agregar una dimensión más a la grilla, según el género. ¿Se puede observar alguna información de interés?

    d) Realizar un único gráfico que contenga todas las estimaciones, clasificadas según el pabellón.

27. (*Optativo 3*)

    a) Escribir una función que, dado un valor de la columna "ApellidoNombre", devuelva únicamente el *apellido*.

    b) Agregar al dataframe una columna que contenga los apellidos.

    c) Imprimir la nueva columna. ¿Cómo resolvió tu función los casos donde el usuario no ingresó su nombre con el formato esperado? Si el comportamiento no es correcto, volver al ítem *a*).

    d) Responder:

    - ¿Cuántos apellidos empiezan con la letra *g*?
    - ¿Hay algún apellido repetido?
    - ¿Cuál es el número promedio de vocales por apellido?