Proyecto de Nivel 4
Resultados prueba Saber 11

Objetivo general

El objetivo de este proyecto es crear una aplicación para analizar los resultados a nivel nacional de las pruebas de estado Saber 11 en el año 2020. En el desarrollo de esta aplicación pondrá en práctica los conceptos del nivel 4.

Objetivos específicos

1. Implementar algoritmos para construir y recorrer matrices.
2. Utilizar las librerías pandas y matplotlib, así como consultar los sitios web oficiales donde se encuentra la documentación.
3. Descomponer un problema en subproblemas e implementar las funciones que los resuelven.

Instrucciones generales

La sección descripción de la aplicación le permitirá conocer el alcance, las funcionalidades esperadas y lo que debe realizar en este proyecto. **Tenga en cuenta que,** a lo largo de dicha sección encontrará el título "**ATENCIÓN**" con indicaciones para conseguir que el resultado de su proyecto corresponda con lo esperado. Es importante que las siga cuidadosamente. Antes de empezar, le sugerimos leer con atención todo el proyecto. Mientras lo lee, trate de reconocer los conceptos del curso que tendrá que poner en práctica. Recuerde que este proyecto debe realizarse de forma **completamente individual**.

Descripción de la aplicación

Las pruebas de estado Saber 11 son un requisito para la graduación y entrada a la educación superior en todo el territorio nacional. Tienen como objetivo evaluar los conocimientos adquiridos por los estudiantes a lo largo de su educación básica secundaria y media. Para el año 2020, las pruebas evaluaban 5 componentes distintos: matemáticas, ciencias naturales, ciencias sociales, lectura crítica e inglés. Todos con un puntaje dentro del rango [0,100]. Así mismo, el estado realiza un censo demográfico anual utilizando como muestra los estudiantes que presentan el examen cada año. Esta información es de utilidad para conocer la distribución demográfica de los estudiantes y sus familias. Dentro de los datos de la encuesta demográfica se pueden encontrar el estrato socioeconómico, el número de personas en la familia, los grados de escolaridad de los padres y si cuenta con acceso a televisión, internet, computador, lavadora, entre otras. Los resultados anonimizados son publicados por el gobierno nacional y son de libre acceso.

En este proyecto, usted trabajará con datos anonimizados del examen Saber 11 y su respectivo censo demográfico del año 2020. Estos datos se encuentran registrados en el portal de datos del gobierno colombiano (https://www.datos.gov.co/Educaci-n/Resultados-Saber-11-2018-1-Refinado/ptck-fi3s). En el archivo "ICFES.csv" usted encuentra una versión simplificada y filtrada de los datos originales: suprimimos algunas entradas incompletas y columnas que no vamos a utilizar en este proyecto. Con este conjunto de datos usted puede participar en la interesante tarea de conocer el comportamiento de los resultados por características demográficas, socioeconómicas y explorar las distribuciones de estas pruebas a lo largo del territorio nacional.

¡Le deseamos la mejor de las suertes explorando los resultados Saber 11!

Su aplicación debe tener las siguientes partes:

Parte 1: Leer la información del archivo

Requerimiento 0: Cargar datos

Lo primero que debe hacer es permitir que se carguen los datos de un archivo csv a un DataFrame. Le debe preguntar al usuario el nombre del archivo, es decir que, **la función que implemente este requerimiento debe recibir como parámetro el nombre del archivo y debe retornar un DataFrame.**

Nota: Las tildes y eñes han sido suprimidas del archivo para evitar problemas de formatos al leer los datos. Tenga en cuenta que los países, municipios, departamentos e instituciones educativas están representados con mayúsculas.

Las columnas del archivo y sus significados son las siguientes:

- TIPO_DOCUMENTO: Tipo de documento de la persona en el momento de presentar la prueba.
- NACIONALIDAD: Representa la nacionalidad de la persona que presentó la prueba Saber 11.
- GENERO: Género de la persona que realizó la prueba Saber 11. Puede ser “F” o “M”.
- DEPARTAMENTO: Nombre del departamento en el cual reside la persona que presentó la prueba.
- MUNICIPIO: Nombre del municipio en el cual reside la persona que presentó la prueba
- ETNIA: Grupo étnico al que pertenece la persona que presentó la prueba. Si la persona no pertenece a ningún Grupo étnico, el valor es “Ninguno”.
- ESTRATO: Estrato socioeconómico al cual pertenece la persona que presentó el examen.
- COLE_NOMBRE: Nombre de la institución educativa en la cual la persona está cursando su último año de educación media. El archivo .csv proporcionado solo contiene información de personas que presentaron la prueba vinculadas con alguna institución educativa.
- PUNT_LECTURA_CRITICA: Puntaje obtenido por la persona en la categoría de lectura crítica sobre 100 puntos.
- PUNT_MATEMATICAS Puntaje obtenido por la persona en la categoría de matemáticas sobre 100 puntos.
- PUNT_NATURALES: Puntaje obtenido por la persona en la categoría de ciencias naturales sobre 100 puntos.
- PUNT_SOCIALES: Puntaje obtenido por la persona en la categoría de ciencias sociales sobre 100 puntos.
- PUNT_INGLES: Puntaje obtenido por la persona en la categoría de inglés sobre 100 puntos.
- PUNT_GLOBAL: Puntaje global de la prueba obtenido por la persona correspondiente sobre 500 puntos.
- LACTEOS: Representa la frecuencia con la que la persona consume productos lácteos. Puede tomar los valores de: *1 o 2 veces por semana, 3 a 5 veces por semana, Nunca o rara vez* y *Todos o casi todos los días.*
- PROTEINAS: Representa la frecuencia con la que la persona consume proteínas como huevos y carne. Puede tomar los valores de: *1 o 2 veces por semana, 3 a 5 veces por semana, Nunca o rara vez* y *Todos o casi todos los días.*
- FRUTOS: Representa la frecuencia con la que la persona consume frutas y legumbres. Puede tomar los valores de: *1 o 2 veces por semana, 3 a 5 veces por semana, Nunca o rara vez* y *Todos o casi todos los días.*
- LECTURA_DIARIA: Representa la frecuencia con la que la persona lee por entretenimiento a la semana. Puede tomar los valores de: *30 minutos o menos, entre 30 y 60 minutos, entre 1 y 2 horas,* más de 2 horas y *No leo por entretenimiento.*

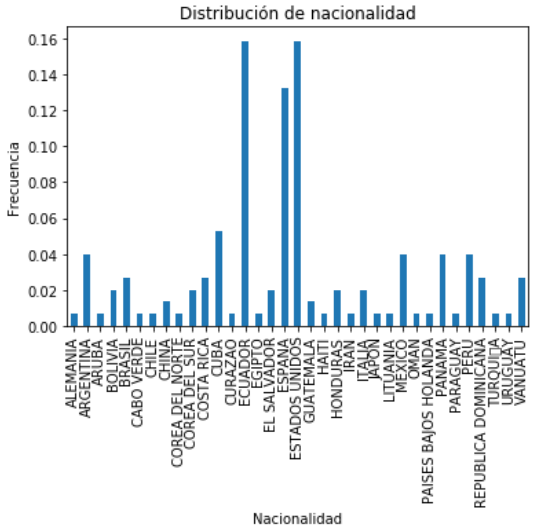
ATENCIÓN: cuando esté estudiando el problema y el archivo, recuerde que las funciones describe(), unique() y filter() pueden serle de utilidad. La función describe() aplicada sobre un DataFrame retorna información estadística de todas las columnas numéricas. La función unique(), aplicada sobre una columna, retorna una lista con los valores únicos que aparezcan en esa columna.

Parte 2: Análisis de la distribución de la prueba Saber 11

Requerimiento 1: Distribución de la población de acuerdo con la nacionalidad

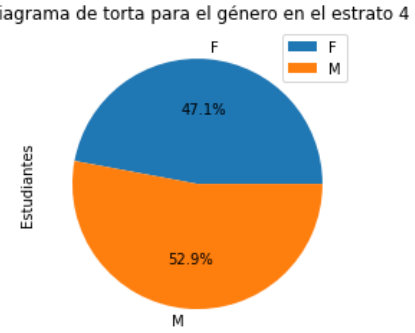
Para esto, se debe crear un nuevo DataFrame a partir del original y presentar un gráfico de barras, el cual contenga la información de frecuencia de todas las nacionalidades a lo largo de la base de datos, a excepción de las nacionalidades

“COLOMBIA” y “VENEZUELA”, las cuales no deben ser tenidas en cuenta. La siguiente figura muestra la apariencia de la gráfica esperada.



Requerimiento 2: Distribución de la población por género y estrato

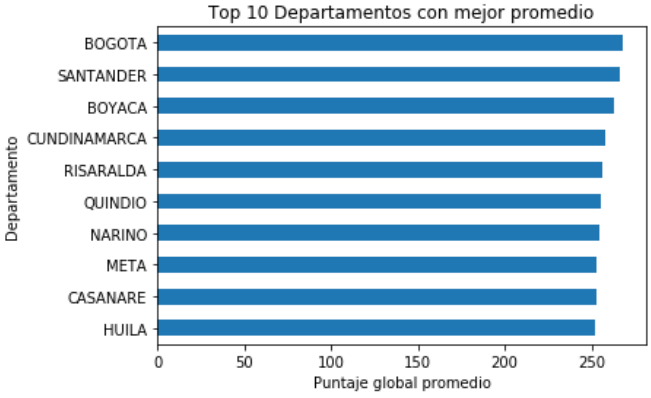
En este requerimiento se quiere conocer la distribución de los estudiantes de acuerdo con su género para un estrato ingresado por el usuario. Para esto, debe calcular la frecuencia de los estudiantes de acuerdo con si son de género masculino (M) o género femenino (F). Debe mostrar los datos en un diagrama de torta como se muestra a continuación. Tenga en cuenta que el diagrama ejemplo es para la distribución del estrato 4.



Parte 3: Análisis del desempeño en las pruebas Saber 11

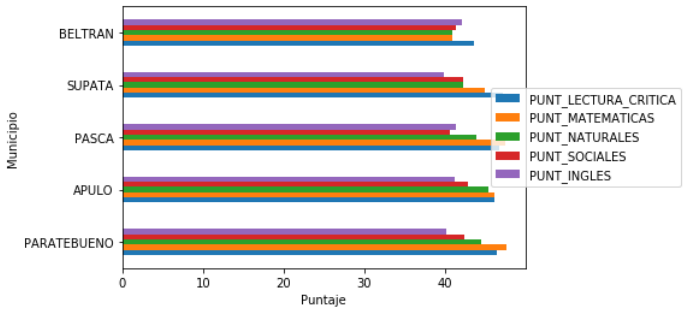
Requerimiento 3: Desempeño de los 10 mejores departamentos

En este requerimiento se quieren conocer los 10 mejores departamentos teniendo como criterio el promedio del puntaje global de todos los estudiantes provenientes de dichos departamentos. Para esto, debe generar una gráfica de barras horizontal que muestre, de mayor a menor, los 10 departamentos con los mejores promedios en la prueba Saber 11. A continuación se muestra la apariencia que debe tener la gráfica generada por su programa.



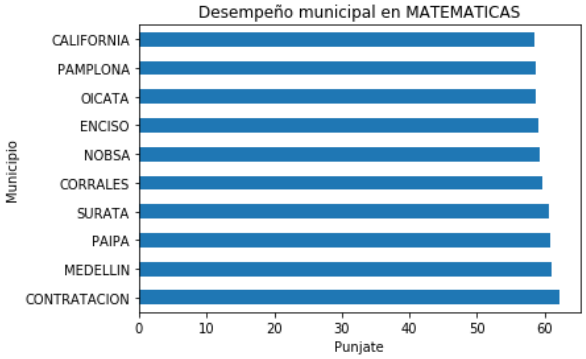
Requerimiento 4: 5 municipios de un departamento con el puntaje más bajo

En este requerimiento se quieren mostrar los 5 municipios con peor desempeño en las pruebas Saber 11 para un departamento dado. Para esto, usted debe filtrar los datos para encontrar aquellos que pertenecen únicamente al departamento dado. Posteriormente, el ordenamiento se debe hacer con el puntaje global promedio de todos los estudiantes que pertenecen a un mismo municipio. Finalmente, la figura generada debe mostrar el desempeño promedio de los peores 5 municipios en cada una de las categorías que evalúa la prueba. A continuación, se muestra la apariencia esperada de la figura para el departamento de *CUNDINAMARCA*.



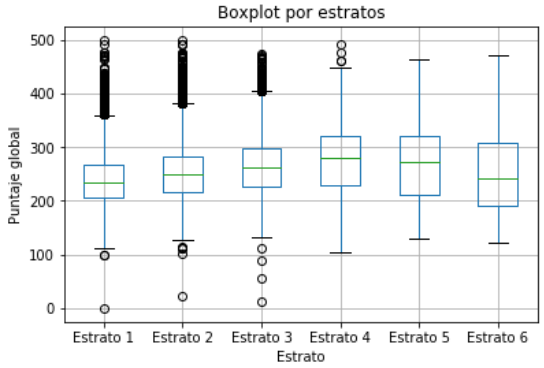
Requerimiento 5: Top 10 desempeño por Municipio dada una categoría

En este requerimiento se quiere mostrar el desempeño en las pruebas Saber 11 de los municipios para una categoría del examen ingresada por el usuario. Este requerimiento debe permitir al usuario escoger la categoría que desea evaluar a nivel municipal. El usuario podrá ingresar las categorías: *MATEMATICAS, NATURALES, SOCIALES, LECTURA* e *INGLES*. El requerimiento debe mostrar los 10 municipios que obtuvieron los mejores puntajes promedio en la categoría especificada por el usuario en orden ascendente. La siguiente figura muestra el diseño esperado de la gráfica para la categoría de *MATEMATICAS.*



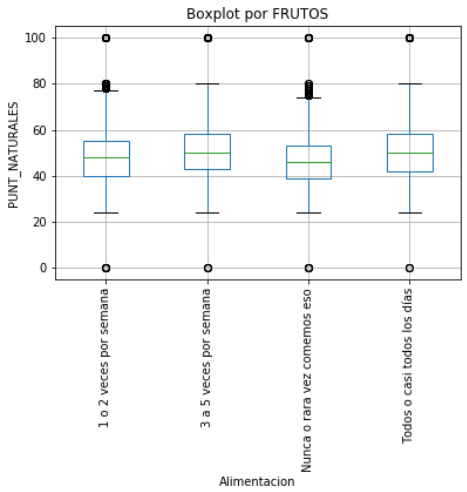
Requerimiento 6: Diagramas de caja según estratos

En este requerimiento se quiere mostrar el desempeño, es decir, el puntaje global en las pruebas Saber 11, de acuerdo con el estrato socioeconómico de las personas que presentaron el examen. Para esto, usted deberá realizar un diagrama de caja por cada estrato registrado en el DataFrame con el que está trabajando, para mostrar la variación en el puntaje global. Tenga en cuenta las recomendaciones dadas al inicio del enunciado. La siguiente figura muestra el aspecto deseado de los diagramas de caja para cada uno de los estratos:



Requerimiento 7: Diagramas de caja por nutrición y categoría.

Dentro de los datos que recoge el censo demográfico, se encuentran variables como la frecuencia con la que un estudiante consume ciertos alimentos (*PROTEINAS*, *LACTEOS* y *FRUTOS*). Se desea determinar si existe una relación entre la nutrición de los estudiantes y el desempeño en una categoría del examen. Este requerimiento consiste en realizar un gráfico de cajas del desempeño de los estudiantes en una categoría, con respecto a un grupo alimenticio dado por el usuario. A continuación, se muestra el aspecto deseado del diagrama de caja para el puntaje en ciencias naturales ("*PUNT_NATURALES*") y el consumo de alimento de tipo "*FRUTOS*".



Parte 4: Estudiar la cantidad de estudiantes por censo demográfico

En esta última parte, no se trabajará directamente sobre el DataFrame original, sino que se creará una matriz a partir de este. La matriz debe relacionar la cantidad de estudiantes que presentaron el examen según su estrato y el número horas de lectura diaria.

Requerimiento 8: Construcción de la matriz de estrato vs horas de lectura

Este requerimiento consiste en construir una matriz que cruce los estratos de los estudiantes y la cantidad de horas que los estudiantes leen por entretenimiento. La matriz tiene la siguiente estructura:

1234	45123	41254	…	12224
6464	645	8746	…	41231
…	…	…	…	…
8412	87211	57571	…	54252

NOTA: La matriz mostrada anteriormente es un ejemplo y **NO** hace referencia a los valores que usted obtendrá al construir la matriz.

Así mismo, se propone la construcción de 2 diccionarios para la referencia de las filas y las columnas. La llave de los diccionarios hace referencia al índice de la fila o columna de la matriz y el valor hace referencia a la respuesta en la encuesta demográfica. A continuación, se muestran los diccionarios para las filas y las columnas.

Llave	Valor
0	"Estrato 1"
1	"Estrato 2"
2	"Estrato 3"
3	"Estrato 4"
4	"Estrato 5"
5	"Estrato 6"

Llave	Valor
0	"30 minutos o menos"
1	"Entre 30 y 60 minutos"
2	"Entre 1 y 2 horas"
3	"Mas de 2 horas"
4	"No leo por entretenimiento"

NOTA: El código para la creación de los dos diccionarios ya se encuentra en el esqueleto proporcionado.

Cada posición (f,c) en la matriz contiene el número de estudiantes que son del estrato f y que leen diariamente un tiempo c. Por ejemplo, si "Estrato 4" está en la fila 3, y "Entre 30 y 60 minutos" en la columna 1, la casilla (3,1) representa el número de estudiantes de estrato 4 que leen diariamente entre 30 y 60 minutos. Note que la suma de una fila completa da como resultado el total de estudiantes que presentaron el examen y que pertenecen a dicho estrato. Por otra parte, note que la suma de una columna da el total de estudiantes que leen dicho número de horas diarias.
La función que usted implementará en este punto debe recibir por parámetro el DataFrame original y retornar una tupla que contenga la matriz con la estructura e información descrita anteriormente, el diccionario para las etiquetas de las filas y el diccionario para las etiquetas de las columnas, en ese orden.

ATENCIÓN:

- Recuerde que tiene a su disposición la función unique() para obtener todos los valores únicos de una columna en un DataFrame.
- Para comprobar que los valores de la matriz creada son los correctos puede utilizar filtros, agrupaciones y sumas sobre el DataFrame original y verificar que estos arrojen el mismo resultado que el almacenado en su matriz.
- Puede ser de mucha utilidad pensar en descomponer el problema en varios problemas más pequeños.

Nota importante: Dado que el tamaño del DataFrame es bastante grande, es muy probable que la construcción de la matriz tarde más tiempo del que suelen tardar otras funciones. Si el tiempo supera los 2 minutos es muy probable que haya un problema en el código, en ese caso, le recomendamos que detenga el programa y revise si existe algún error en sus funciones.

Utilizando como base la matriz anterior, usted deberá implementar los siguientes requerimientos:

Requerimiento 9: Número de estudiantes que presentaron la prueba según estrato

En este requerimiento se debe calcular el número total de estudiantes que presentaron la prueba según un estrato socioeconómico. Para esto, el usuario debe indicar el estrato para el cual desea obtener el número total de estudiantes. La función que implemente esta opción debe retornar un entero que represente el número de estudiantes que presentaron la prueba para el estrato dado.

Requerimiento 10: Número de estudiantes con una lectura diaria establecida

En este requerimiento se debe calcular el número total de estudiantes que leen un determinado número de horas al día. Para esto, el usuario debe indicar la cantidad de horas de lectura diaria para el cual desea obtener el total de estudiantes. La función que implemente esta opción debe retornar el número total de estudiantes que cumplan con el tiempo de lectura ingresado por el usuario.

Requerimiento 11: Estrato con mayor número de estudiantes

En este requerimiento se debe determinar el estrato que tuvo el mayor número de estudiantes que tomaron la prueba Saber 11. La función que implemente esta opción debe calcular el número de estudiantes para todos los estratos de la matriz y debe retornar aquel que tuvo mayor número de estudiantes.

Requerimiento 12: Estrato y tiempo de lectura con mayor cantidad de estudiantes

En este requerimiento se debe determinar el estrato y el rango de lectura diaria en el que se registró el mayor número de estudiantes. La función que implemente está opción debe calcular la fila y la columna en donde se registró el mayor número de estudiantes dentro de toda la matriz, y retornar una tupla de la forma (*e,t*), donde *e* es el estrato y *t* el tiempo de lectura para los cuales está el mayor número de estudiantes dentro de la matriz.

Actividad 1: Preparación del ambiente de trabajo

1. Cree una carpeta para trabajar, poniéndole su nombre o login.
2. Descargue de BrightSpace el archivo "ICFES.zip" que contiene el archivo ".csv" con los datos a procesar y un archivo .py que debe usar como esqueleto para realizar el proyecto. Este archivo contiene el código para crear los diccionarios de las filas y columnas de la matriz que usted debe construir.
3. Abra Spyder y cambie la carpeta de trabajo para que sea la carpeta donde descargó el archivo con los datos.

Actividad 2: Construir el módulo de funciones

4. Usando Spyder, cree en su carpeta de trabajo un nuevo archivo con el nombre "resultados_icfes.py". En este archivo usted va a construir el módulo en el que va a implementar las funciones que responden a los requerimientos de la aplicación. **Defina**, **documente** e **implemente** las funciones en su nuevo archivo. Usted puede crear cuántas funciones considere necesarias dentro de su librería o módulo. Mínimo debe haber una función por cada uno de los requerimientos del programa.

Actividad 3: Construir la interfaz de usuario basada en consola

5. Implemente la interfaz de la aplicación en un nuevo archivo llamado "consola_ICFES.py". Esta interfaz debe seguir la misma estructura de las consolas que hemos implementado en laboratorios y proyectos anteriores. Esto es: debe existir una función llamada iniciar_aplicacion() para que muestre el menú usando la función mostrar_menu() y permita al usuario seleccionar una opción. El menú que se despliegue debe permitir al usuario ejecutar todas las acciones de la descripción de la aplicación, así como salir (terminar) del programa. Por cada una de las funciones principales de su programa, debe existir una función en la consola que la ejecute, pidiendo previamente los datos necesarios al usuario (si aplica) e imprimiendo por pantalla el resultado de la función. Se sugiere nombrar estas funciones como ejecutar_XX, donde XX es la respectiva función de su módulo. Cuando haya implementado la función iniciar_aplicacion()corra su programa y verifique que se comporta adecuadamente, le permite al usuario seleccionar las opciones que quiere ejecutar, y termina el programa cuando se le indique.

Actividad 4: Probar el correcto funcionamiento de su programa

6. **Ejecute el programa** y **pruebe** cada una de las funciones para asegurarse que esté funcionando. Puede probar el correcto funcionamiento de su programa cargando la información que se encuentra en el archivo "ICFES.csv" o creando su propio archivo de prueba de menor tamaño (respetando el mismo formato) que le permita corroborar que los resultados arrojados por su programa son correctos.

Entrega

7. Comprima los dos archivos: resultados_icfes.py y consola_ICFES.py en un solo archivo .zip. El archivo comprimido debe llamarse "**N4-PROY-login.zip**", donde login es su nombre de usuario de Uniandes (omita el punto del login para evitar posibles problemas con la extensión de los archivos. Por ejemplo, si su login fuese "p.perez123", nombre el archivo como "N4-PROY-pperez123.zip").
8. Entregue el archivo comprimido a través de BrightSpace en la actividad designada como "**Proyecto N4**".