

Métodos Computacionales

Tarea 3 — 2018-10

La solución a este taller debe subirse por SICUA antes de las 5:00PM del lunes 19 de marzo del 2018. Si se entrega la tarea antes del lunes 12 de marzo del 2018 a las 11:59PM los ejercicios se van a calificar con el bono indicado.

(10 puntos) Los archivos del código deben subirse en un único archivo `.zip` con el nombre `NombreApellido_taller3.zip`, por ejemplo si su nombre es Vandana Shiva debería subir el zip `VandanaShiva_taller3.zip` al descomprimir el zip debe crearse la carpeta `VandanaShiva_taller3` y adentro deben estar los archivos solicitados. En la implementación principal de los algoritmos solicitados la copia y reutilización de código de cualquier fuente de internet (incluido el repositorio del curso) deja la nota en cero.

1. Un script para hacer todo

(10 puntos) Prepare un unico script llamado `NombreApellido_tarea3.sh` con los comandos para descargar los datos y ejecutar los scripts de python de los siguientes ejercicios.

2. Airbnb Berlin

(30 (35) puntos) En este ejercicio usted debe analizar los datos de Airbnb de la ciudad de Berlin que se encuentran en <https://s3.amazonaws.com/tomslee-airbnb-data-2/berlin.zip>. Para esto escriba el script de Python `berlin.py` que **para cada uno de los archivos dentro del zip**:

- Lea los datos de `overall_satisfaction`, `acommodates`, `bedrooms`, `price`, `minstay`.
- Calcule la media y desviación estándar de cada columna para renormalizar los datos para que tengan media cero y desviación estándar de uno.
- Usando su propia implementación, calcule la matriz de covarianza.
- Utilice matriz de covarianza para calcular la lista de las dos variables que están más correlacionadas y las dos más anticorrelacionadas.

Los resultados de las variables correlacionadas/anticorrelacionadas se deben escribir en el archivo de texto `variables_berlin.txt` con cinco columnas: la fecha que se encuentra el nombre del archivo procesado (por ejemplo `2015-07-04`), el nombre de las dos variables correlacionadas y el nombre de las dos variables anticorrelacionadas.

3. Educación, Etnicidad y Salarios

(30 (35) puntos) En este ejercicio usted debe analizar los datos de educación, etnicidad y salarios que se encuentran aquí:

<https://github.com/vincentarelbundock/Rdatasets/blob/master/csv/DAAG/cps1.csv>.

La descripción de estos datos se encuentra aquí:

<https://github.com/vincentarelbundock/Rdatasets/blob/master/doc/DAAG/cps1.html>

Para esto escriba el script de Python `pca_salario.py` que incluya una función

```
def predice_salario(age,educ,black,hisp,marr,nodeg)
```

que prediga el salario en 1974 dadas las entradas correspondientes. La función debe devolver un float con la predicción para el salario. La predicción se **debe** calcular usando análisis de componentes principales tal como se explica en la sección 6.3.1 del libro ISL.

4. Spline cúbico

(30 (35) puntos) El spline cúbico es un método de interpolación. La idea es que dados N pares de puntos discretos (x_0, \dots, x_{N-1}) con sus valores correspondientes (y_0, \dots, y_{N-1}) se puede encontrar un conjunto de funciones que conecta esos puntos de una manera suave. En este ejercicio debe escribir su propia implementación tomando como condición de frontera que la segunda derivada en los puntos extremos es cero.

Para esto debe crear un archivo llamado `interpolacion.py` que contenga a la función

```
def mi_spline(x_in, y_in, x_inter)
```

que toma como entrada los arrays de numpy `x_in`, `y_in` como los puntos que dan la base para crear la interpolación y `x_inter` los puntos que se desean interpolar.

Si `x_inter` es un escalar la función debe devolver un escalar con el valor de la interpolación en ese punto. Si `x_inter` es un array de numpy la función debe devolver un array de numpy con los valores correspondientes de la interpolación en esos puntos.

La función `mi_spline` **debe** encontrar los valores buscados a través de la resolución del sistema de ecuaciones lineales que definen al spline cúbico. Ver por ejemplo https://en.wikiversity.org/wiki/Cubic_Spline_Interpolation