

Introducción a Ciencias de la Computación

Semestre 2020-1

Práctica 14

Profesor: Amparo López Gaona
Ayud. Lab.: Marco Iván Olea Olvera

Jueves 07 de Noviembre de 2019

1 Objetivos

- Practicar el uso de excepciones y bloques `try/catch`.
- Familiarizarse con la lectura y escritura de archivos en Java.

2 Introducción

2.1 CSV

Un archivo CSV (*Comma-Separated Values*) es un archivo de texto usado para almacenar tablas en texto plano. Cada línea del archivo representa un registro, y cada registro está compuesto por uno o más campos separados por comas.

Por ejemplo:

```
#Cuenta, Carrera, Generacion
312553854, C. Computacion, 2015
313219735, Actuarial, 2016
308385091, Fisica, 2011
313009124, C. Computacion, 2016
```

2.2 Regresión lineal simple

Una regresión lineal simple es un modelo que busca aproximar la relación entre una variable independiente y una variable dependiente, típicamente representados con el eje x y el eje y en el plano Cartesiano.

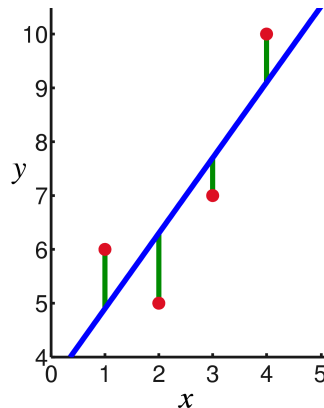
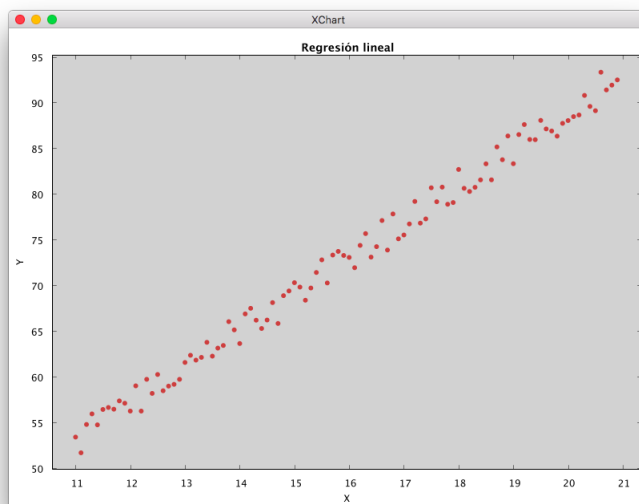


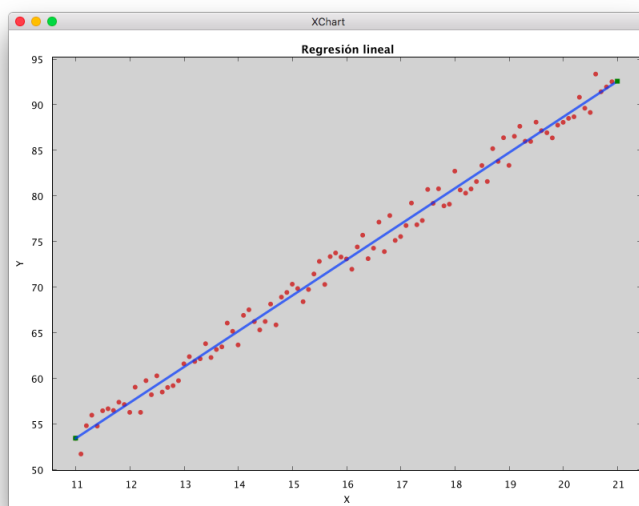
Figura 1: En una regresión lineal, las *observaciones* (rojo) son el resultado de *desviaciones* (verde) de una *relación* (azul) entre una variable dependiente (y) y una variable independiente (x).

3 Descripción

La práctica consiste en escribir un método para leer archivos CSV, y usar esa implementación para extraer los datos de un archivo CSV cuyos registros representan puntos en el plano Cartesiano. Posteriormente se calculará la función lineal que minimice las desviaciones de dichos puntos a la función calculada. Es decir, dados los siguientes puntos rojos:



Debes reconstruir la línea azul:



Para una regresión lineal simple dados n puntos $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, esto se hace de la siguiente manera:

$$m = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}$$
$$b = \frac{\sum_{i=1}^n y_i - m \sum_{i=1}^n x_i}{n}$$

Y esto nos genera la ecuación $y = mx + b$.

4 Desarrollo

1. Crea la clase `practica14.LectorCSV` que contenga el siguiente método:

```
public static List<String[]> leer(String ruta) throws IOException {}
```
2. Este método recibe un `String` que representa la ruta a un archivo CSV, y devuelve una lista donde cada elemento es un registro de la tabla del CSV, representado como un `String[]`.
Por ejemplo, dado un archivo CSV que contiene la información de la sección 2.1, este método regresaría una lista equivalente a la siguiente:

```
List.of(new String[]{"#Cuenta", "Carrera", "Generacion"},
        new String[]{"312553854", "C. Computacion", "2015"},
        new String[]{"313219735", "Actuaria", "2016"},
        new String[]{"308385091", "Fisica", "2011"},
        new String[]{"313009124", "C. Computacion", "2016"})
```

3. Usa el programa `GeneraCSV.class` y tu número de cuenta para generar puntos aleatorios, de la siguiente manera:

```
java GeneraCSV 318002389
```


Esto creará el archivo `src/main/resources/318002389.csv`.
4. En la clase `practica14.Graficacion` utiliza el método `LectorCSV.leer(String)` y el CSV generado para crear dos arreglos, uno de todas las coordenadas x y otro de todas las coordenadas y .
5. Calcula la pendiente y la intersección con el eje y .
6. Utiliza estos resultados para mostrar una gráfica como la que se encuentra en la sección 3.

Recuerda documentar tu código y seguir las convenciones y reglas de estilo de Java.

5 Entrega

<https://github.com/usuario123/ICC-2020-1>

