

Práctica 8

Objetivos: Practicar con recorridos de arrays

Descripción: Crear un programa para tratar los datos de un experimento, con las siguientes clases:

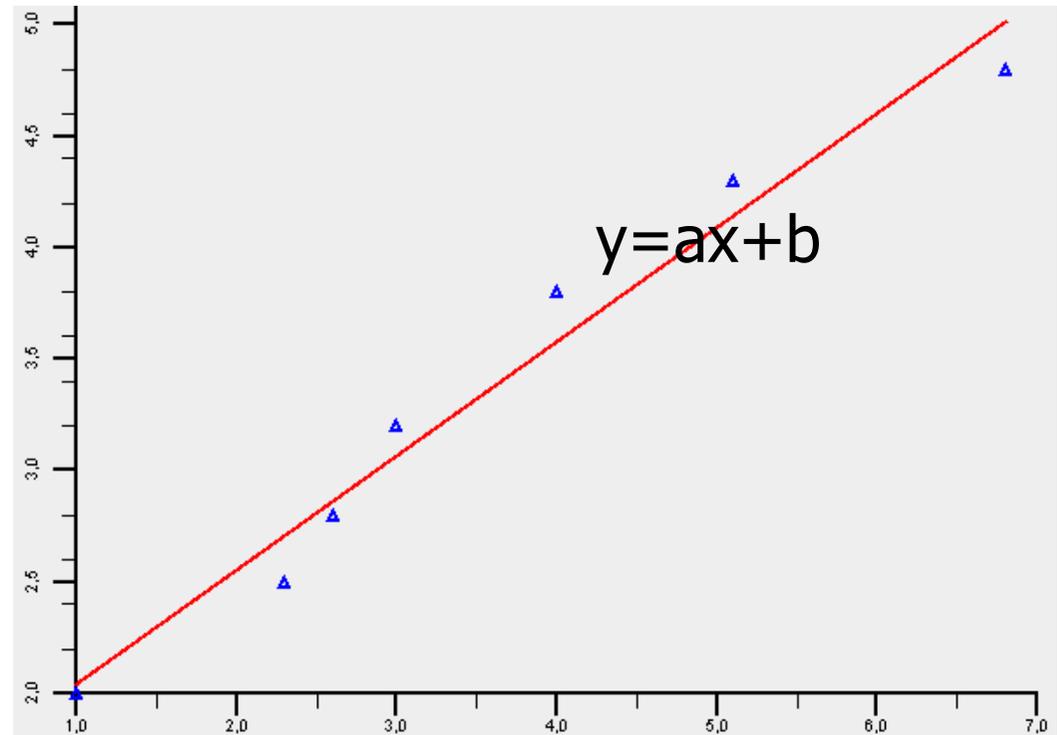
- La clase **Coeficientes** que contiene los resultados de aplicar una regresión lineal a los datos experimentales
 - esta clase se da ya hecha
- La clase **Estadística** que guarda los datos de las medidas experimentales de una función real $y=f(x)$, en forma de parejas de puntos $\{x, y\}$
- La clase **DatosExperimento** con un programa principal que pide por teclado los datos de un experimento y muestra en pantalla resultados de la estadística de estos datos

Regresiones lineales

El ajuste por regresiones lineales permite encontrar la recta que más se aproxima a la función real $y=f(x)$ para un conjunto de puntos medidos experimentalmente

La recta se determina en función de:

- coeficientes a y b , de la ecuación de la recta
- coeficiente de correlación, r , que mide la precisión de la solución
 - la solución es mejor cuanto más cerca de la unidad esté r , en valor absoluto



Clase Coeficientes

Atributos:

- los coeficientes **a** y **b** de la recta y la correlación **r**

Métodos:

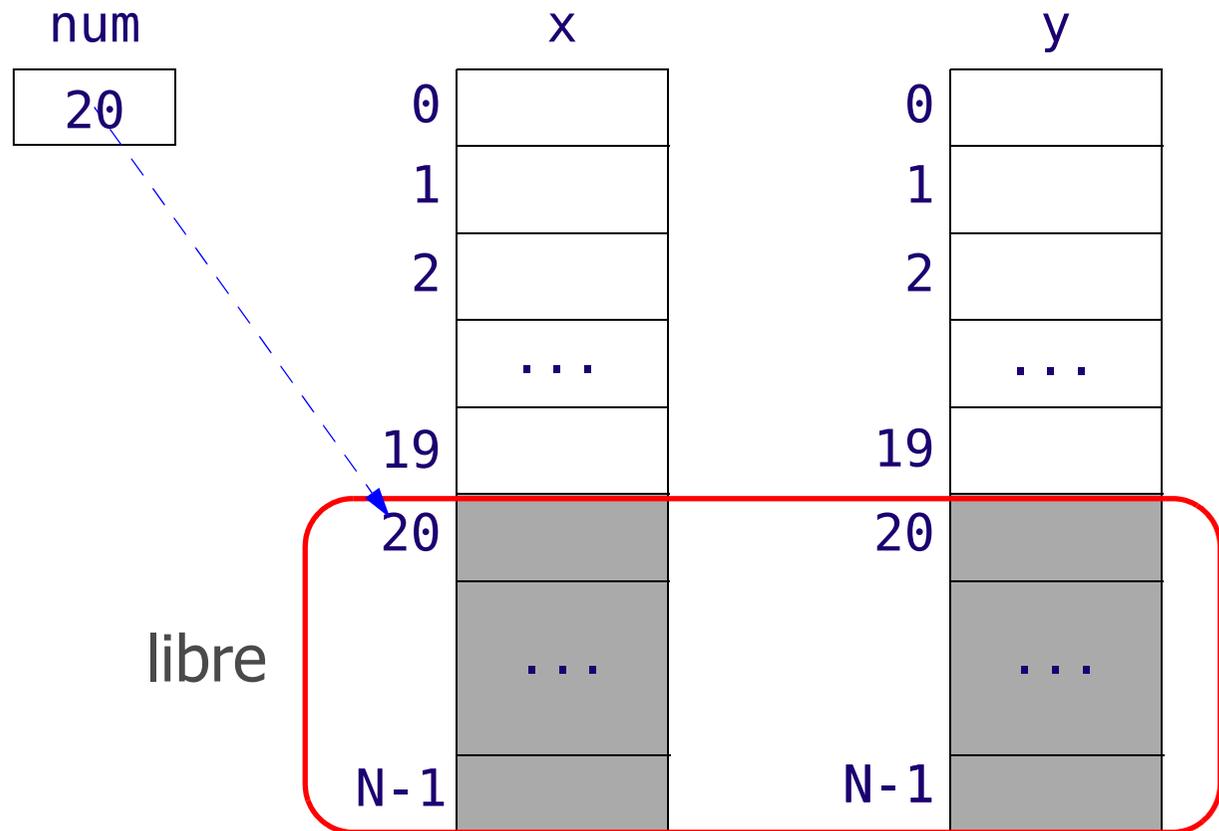
- *constructor*: al que se pasan los valores iniciales de los atributos
- *observadores* para cada atributo

Coeficientes
-double a -double b -double r
+Coeficientes(double a, double b, double r) +double coefA() +double coefB() +double correlacion()

Clase Estadística

Atributos:

- x e y : dos arrays "paralelos", del mismo tamaño, uno para los valores de las x y otro para las y
- num : una variable entera que indica cuántas casillas de los arrays x o y se encuentran ocupadas
 - las demás quedan libres, para meter valores adicionales



Ejemplo con 20 casillas ocupadas

Clase Estadística (cont.)

Métodos:

- *constructor*: crea los arrays de tamaño `maxDatos` y pone `num` a cero
- *inserta()*: si los arrays `x` e `y` aún tienen casillas libres guarda los parámetros en `x[num]` e `y[num]`, incrementa `num` en uno y retorna `true`. Si no hay casillas libres retorna `false` para indicar el error
- *numDatos()*: retorna el atributo `num`

Estadística
-double[] x -double[] y -int num
+Estadística(int maxDatos) +boolean inserta(double x, double y) +int numDatos() +double media() +pintaRegresionLineal() +Coeficientes regresionLineal()

Clase Estadística (cont.)

- `media()`: retorna la media de la función $y=f(x)$, obtenida mediante la siguiente expresión:

$$\frac{1}{(x_{num-1} - x_0)} \sum_{k=0}^{num-2} \frac{y_{k+1} + y_k}{2} (x_{k+1} - x_k)$$

siendo $x_i=x[i]$ e $y_i=y[i]$

- `pintaRegresionLineal()`: muestra una gráfica de los puntos y la regresión lineal
 - se da ya hecho
- `regresionLineal()`: Realiza el cálculo de la regresión lineal explicado más abajo y retorna los resultados en un objeto de la clase `Coeficientes`. Si `num` es 2 o menos, el cálculo no puede hacerse y se retorna `null` para indicar el error

Cálculo de la regresión lineal

Para hacer el cálculo deben obtenerse los siguientes sumatorios respecto a las parejas de valores (x_i, y_i) almacenados en los atributos x e y :

- $\text{sumX} = \sum x$
- $\text{sumY} = \sum y$
- $\text{sumX2} = \sum x^2$
- $\text{sumY2} = \sum y^2$
- $\text{sumXY} = \sum x \cdot y$

Observar que el recorrido que se haga para estos cálculos es solo para las casillas ocupadas: desde 0 hasta $\text{num}-1$

Los coeficientes de la regresión se obtienen con las siguientes expresiones

Cálculo de la regresión lineal (cont.)

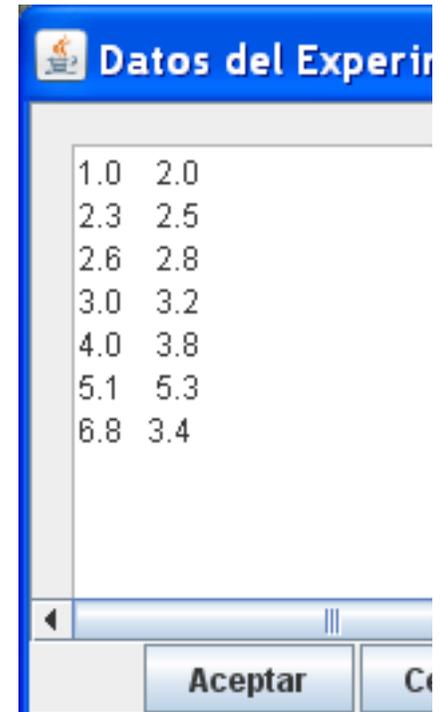
$$b = \frac{\sum y - a \sum x}{num}$$

$$a = \frac{num \sum xy - \sum x \sum y}{num \sum x^2 - (\sum x)^2}$$

$$r = \frac{num \sum xy - \sum x \sum y}{\sqrt{(num \sum x^2 - (\sum x)^2) \cdot (num \sum y^2 - (\sum y)^2)}}$$

Programa principal (DatosExperimento)

- crea un objeto de la clase Estadística
- crea un objeto de la clase CajaTexto
- lee de él los puntos x e y , que tendrán el formato que se muestra
 - los valores x están en la 1ª columna y los y en la 2ª
- inserta estos puntos en la estadística con `inserta()`
 - si se detecta que no caben, pone un mensaje de error en pantalla
- si hay menos de dos parejas (x,y) pone en pantalla un mensaje de error; en caso contrario:
 - muestra en pantalla los resultados de la regresión lineal $(a, b$ y $r)$ usando una ventana de la clase Escritura
 - invoca al método `pintaRegresionLineal()` de la estadística, para pintar la gráfica



Resumen del funcionamiento de la clase CajaTexto

Método	Descripción
<code>CajaTexto("título", alto, ancho)</code>	Crea una ventana con el título y el tamaño indicados
<code>void espera()</code>	Espera a que el usuario pueda introducir los datos
<code>boolean hayMas()</code>	Retorna un booleano que indica si hay más datos para leer o no
<code>double leeDouble(0)</code>	Lee un número real de la primera columna de la línea actual (valor <i>x</i>)
<code>double leeDouble(1)</code>	Lee un número real de la segunda columna de la línea actual (valor <i>y</i>)
<code>void avanzaLinea()</code>	Salta a la siguiente línea

Parte Avanzada

Modificar la clase `Estadistica` para añadir un método que calcule una regresión logarítmica, en lugar de lineal, obteniendo la siguiente curva de ajuste:

$$y = a \cdot \ln(x) + b$$

Añadir también a la clase un método similar a `pintaRegresionLineal()` que pinte la regresion logarítmica en una gráfica

Crear un nuevo programa principal similar al anterior que aplique una regresión logarítmica

Nota: Consultar https://es.wikipedia.org/wiki/Regresi%C3%B3n_no_lineal#Regresi%C3%B3n_logar%C3%ADtmica

Entregar

El proyecto Bluej en un archivo comprimido

Informe:

- Parte básica:
 - código de la clase `Estadistica`
 - código del programa principal realizado
 - resultados obtenidos usando los datos que aparecen en el informe:
 - en la consola Java
 - captura de pantalla de la gráfica
- Parte avanzada
 - código de los métodos añadidos a la clase `Estadistica`
 - resultados obtenidos usando los datos que aparecen en el informe:
 - en la consola Java
 - captura de pantalla de la gráfica