Problemas numéricos con corrección automática: qué se puede y qué se debe hacer

González Alastrué, *José Antonio*

jose.a.gonzalez@upc.edu

Resumen

La presente comunicación describe los objetivos, organización, proceso y discusión de un taller orientado a proporcionar a docentes del campo de la estadística y la investigación operativa (y posiblemente de otros campos) una primera noción sobre la construcción de problemas con la plataforma e-status. El taller se estructura en base a cuatro casos que van incrementando el nivel de complejidad, para ir mostrando diversas posibilidades didácticas a la hora de elaborar buenos ejercicios que hagan reflexionar al estudiante.

Palabras clave: e-status, plataformas educativas, problemas automáticos.

Clasificación AMS: 62-01, 97-01, 97-04.

Introducción

El objetivo del taller es que el asistente se haga una idea clara sobre cómo se construye un problema con la plataforma e-status. Es un taller del tipo *hands-on*, o «manos a la obra»: es decir, desde el primer momento el participante se ha de involucrar activamente, en vez de darle todo el protagonismo al ponente para que ocupe el tiempo hablando. Por supuesto, dado que es de esperar que en la audiencia haya personas que conocen poco o nada sobre e-status, es necesario dar unas referencias mínimas pero suficientes para proporcionar un contexto.

e-status es un aplicativo que funciona en web, y que se utiliza para poner a disposición de los alumnos (de asignaturas de probabilidad y estadística, generalmente) una serie de problemas que el alumno contesta mediante el navegador, obteniendo la corrección correspondiente de inmediato. Los problemas pueden repetirse tantas veces como sea necesario, puesto que tienen la capacidad de variar sus condiciones aleatoriamente. Al cambiar estas condiciones, las soluciones también son distintas.

El sistema que gobierna la lógica de los problemas es el programa/lenguaje R, bien conocido por los estadísticos. El profesor escribe un código para generar el modelo del problema y para valorar las respuestas de los alumnos. La aplicación se ocupa, entre otros aspectos, de la comunicación en el servidor entre lo que es propiamente el servicio web y R. Puede encontrarse más información en [1] y [2].

La clave de e-status radica en su especialidad. El programa se ha desarrollado desde cero, y sin plantear su integración en entornos más generales, conocidos como LMS (*Learning Management Systems*), de los cuales posiblemente sea el más popular Moodle por el hecho de ser *software* libre. Moodle dispone de algunas herramientas para elaborar cuestiones (o *quizzes*) de tipos muy variados. Sin embargo, la elaboración de preguntas numéricas es limitada y, en opinión del autor, algo farragosa. Existe un paquete para R llamado *exams* que puede integrarse en Moodle, mediante el paquete *Sweave*, y puede utilizarse el lenguaje R para programar la lógica del problema, aunque se trata fundamentalmente de la generación del modelo; la valoración de las respuestas continúa siendo relativamente pobre y compleja. e-status debe incorporar una gestión mínima de alumnos y profesores, asignaturas y grupos, calificaciones y estadísticas de resultados, pero a cambio proporciona agilidad y sencillez de uso al profesor para que este consiga hacer realidad el problema que tiene en la mente.

En las páginas que siguen se va a describir el funcionamiento del taller a partir de los contenidos empleados para el mismo, y que se pueden encontrar en la página web:

<http://www-eio.upc.es/~josean/genaeio14/problema1.html>.

Procedimiento del taller

Previamente a la celebración del taller, se solicitó a los participantes una serie de datos que permitieran efectuar el alta con el perfil de «profesor» (que permite editar problemas) en e-status. Los pocos casos que quedaran sin resolver pueden solucionarse en los minutos iniciales del taller. De esta manera, los participantes pueden entrar en la aplicación inmediatamente. El ponente muestra en el proyector la ubicación del enlace para entrar en el editor, obviando por el momento otras funcionalidades de e-status. También proporciona la dirección web que incluye los contenidos preparados para el taller: se trata de cuatro páginas (problema1.html, problema2.html, problema3.html, problema4.html) enlazadas entre sí. Cuatro problemas completos que representan sendas fases de complejidad creciente, y que cada uno de los presentes tiene el cometido de implementar y probar desde el editor.

Como se puede deducir de lo anterior, el taller se ha planteado alejándose de un paradigma muy frecuente: pedir a los asistentes que construyan su trabajo a partir de un tema de su interés particular. En vez de emplear este enfoque, hemos decidido dar un material común a todos, pormenorizadamente detallado para que el participante no tenga más que copiar y pegar una serie de textos en el lugar correspondiente. ¿Creatividad? De acuerdo, sacrificada en aras de la eficiencia (si se dispusiera de más de dos horas, tal vez sería bueno invitar al asistente a experimentar). Hay un acuerdo generalizado, que no se va a poner aquí en duda, de que la implicación personal del alumno es un factor clave para el aprovechamiento de un curso. Sin embargo, el grado de implicación suele ser muy diverso, y es necesario disponer de tiempo sobrado para conseguir resultados satisfactorios en la sesión. Por estas razones se prefiere que este taller procure llevar hasta el final al mayor número posible de asistentes, teniendo en cuenta que tampoco es necesario completar los cuatro ejercicios. Un participante puede sentirse satisfecho si completa solo tres, o dos, problemas, y el resto puede aportar un beneficio marginal, aunque obviamente se le invita a finalizar las tareas en un momento posterior.

La primera pantalla del editor es el organizador de problemas (Figura 20). El ponente explica brevemente que el profesor-autor de problemas tiene la capacidad de abrir las carpetas que necesite para gestionar sus propios problemas, y además puede acceder a los problemas de sus compañeros siempre que estos se lo permitan (la filosofía general es la compartición de materiales, pero sin renunciar al derecho a la privacidad). Lo que el profesor vería en la pantalla son dos estructuras de tipo árbol: a la izquierda puede gestionar el objeto tipo *problema*, y a la derecha el objeto tipo *carpeta* (una operación típica sería mover el problema marcado a la carpeta marcada). Obviamente, al principio no hay ningún objeto creado, solo puede ver tres contenedores genéricos llamados «Carpetas propias», «Externas subscritas» y «Externas no subscritas»; se pide al participante que comience creando una carpeta, y un problema vacío en la misma. Tras ello accede al editor propiamente dicho (Figura 21), y puede empezar a componer el primer ejercicio, que se encuentra estructurado en las Tablas 1 y 2, y que son una réplica de las que se encuentran en la página web del taller.

Figura 20. Organizador de carpetas y problemas

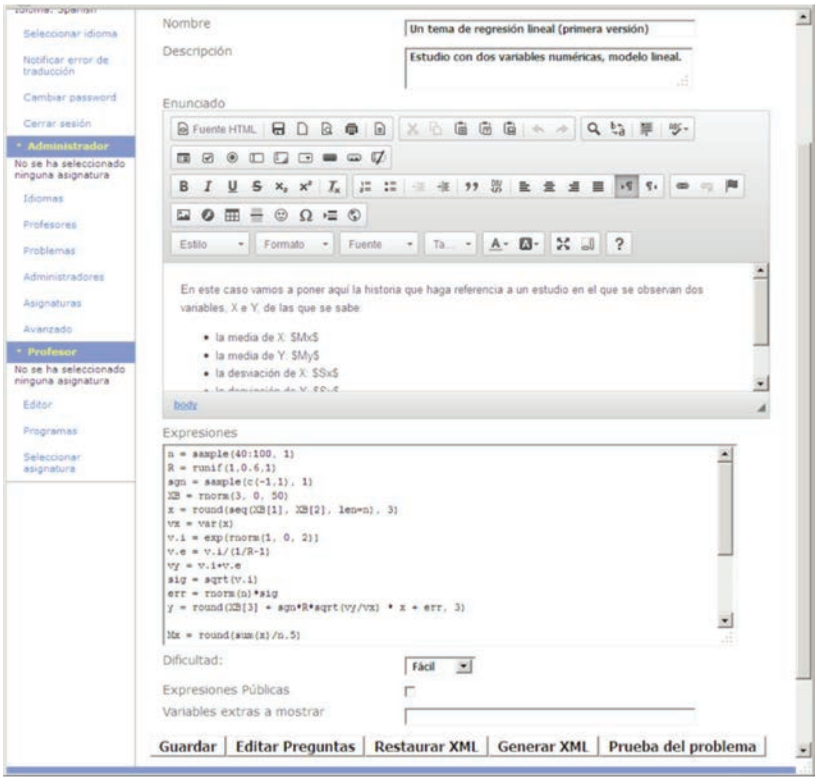

Figura 21. El editor de problemas

Un tema de regresión lineal (primera versión)	
Descripción	**Estudio con dos variables numéricas, modelo lineal.**
Enunciado	**En este caso vamos a poner aquí la historia que haga referencia a un estudio en el que se observan dos variables, X e Y, de las que se sabe:** • **la media de X: Mx** • **la media de Y: My** • **la desviación de X: Sx** • **la desviación de Y: Sy** • **la covariancia de X e Y: Vxy**
Modelo R (Expresiones)	**n = sample(40:100, 1)** **R = runif(1,0.6,1)** **sgn = sample(c(-1,1), 1)** **XB = rnorm(3, 0, 50)** **x = round(seq(XB[1], XB[2], len=n), 3)** **vx = var(x)** **v.i = exp(rnorm(1, 0, 2))** **v.e = v.i/(1/R-1)** **vy = v.i+v.e** **sig = sqrt(v.i)** **err = rnorm(n)*sig** **y = round(XB[3] + sgn*R*sqrt(vy/vx) * x + err, 3)** **Mx = round(sum(x)/n,5)** **My = round(sum(y)/n,5)** **Vxy = round(cov(x,y),5)** **Sx = round(sd(x),5)** **Sy = round(sd(y),5)** **LM = lm(y ~x)** **summ = summary(LM)** **tind = summ$coef[1,1]**
Expresiones públicas	**Para compartir el modelo con otros autores.**
Variables extras a mostrar	**(de momento, nada)**

Tabla 1. Datos principales para la primera versión del ejercicio

Una vez se ha introducido la información contenida en la Tabla 1, se pulsa sobre el botón «Guardar», y seguidamente se elige la opción «Editar preguntas». En la pantalla que aparece inicialmente tendremos una lista vacía: usaremos una opción que aparece en el menú de la izquierda para crear la primera pregunta, con lo que se accede a la página que permite rellenar los diferentes campos que componen una pregunta, cuyo contenido se encuentra en la Tabla 2.

Preguntas	
Enunciado 1	**Si queremos predecir la Y con la X, utilizando un modelo lineal, ¿cuánto vale el término independiente de la recta estimada (usando el método de mínimos cuadrados)?**
Solución a la pregunta	**Tind**
Parámetro de sintaxis	
Sintaxis de la respuesta	**Real**
Plantilla de expresiones de la respuesta	**Error relativo**
Parámetro de expresiones	**1 (un tanto por ciento)**
Expresiones de respuesta	**if (estatus.error_relativo(solucion_,respuesta_)<=parametro_)** **{** **resultado_<-'1';** **}else{** **resultado_<-'0';** **}** **(no tocamos nada)**
Peso	**1**
Ayuda contextual	**Aquí va un enlace en donde puedes leer más sobre el método:**
Link	**<http://es.wikipedia.org/wiki/Regresi%C3%B3n_lineal#Eqnref_12>**
Intentos	**1**
Mostrar solución	**(normalmente es lo más sensato)**
Mensaje correcto	**¡Bravo!**
Mensaje semicorrecto	
Mensaje incorrecto	**¡La próxima vez lo harás mejor!**

Tabla 2. Datos de la pregunta para la primera versión del ejercicio

Con esta información completada, pulsamos sobre el botón «Guardar», con lo que retornamos a la pantalla de preguntas, en las que ya figura una pregunta en la lista, que se identifica gracias al enunciado de la misma, la variable de respuesta y el peso asignado. Podemos retroceder y volver a la pantalla de edición principal con la opción «Editar problema» que aparece en el menú de la izquierda.

El aspecto del problema, tras haber actuado sobre el botón «Prueba del problema», podría ser como el de la Figura 22.

La página del tutorial contiene además algunas ideas más generales, incluidas en la siguiente caja. En primer lugar se menciona el reto de la precisión en la respuesta. Puesto que normalmente las soluciones son números reales, es necesario dar como admisible una respuesta que esté lo suficientemente cerca del valor exacto que conocemos (cuando sea el caso). En realidad, el criterio a utilizar depende en buena medida del valor que esperamos; quizá no tanto del valor numérico como del tipo. No es lo mismo una probabilidad que una medida cuyo orden de magnitud pueda ser muy variable, y tampoco es lo mismo una probabilidad cualquiera que una probabilidad que sabemos que generalmente va a ser pequeña.

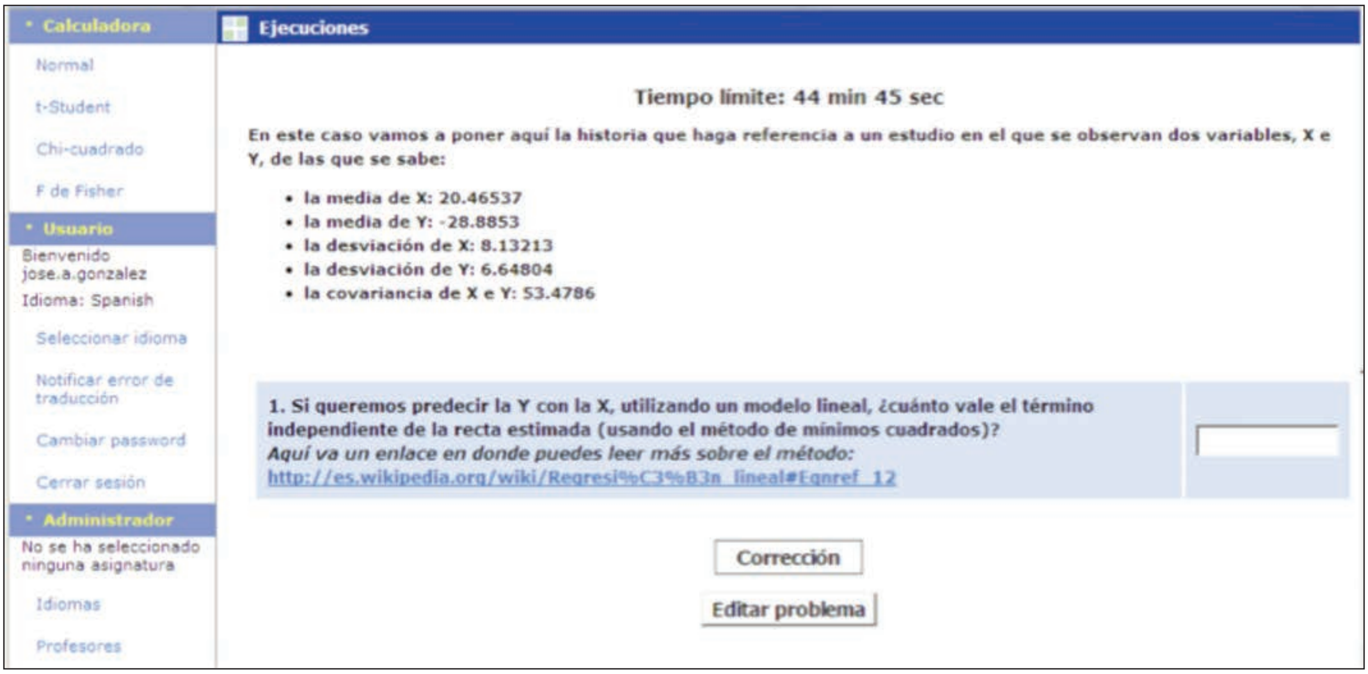

Figura 22. Aspecto de la primera versión del problema

Seguidamente figuran una serie de propuestas, que no son otra cosa más que plantear el campo de acción natural que se abre a continuación, después de crear nuestro primer y simple prototipo. La segunda versión va a abordar algunas de estas cuestiones.

Comentario

El criterio del error relativo para corregir una respuesta es muy sensible. ¿Qué pasaría si en un caso la solución estuviera muy cerca de 0? ¿Sería lógico exigirle al menos dos cifras significativas? Por otra parte, sin conocer el valor exacto, el alumno no tiene fácil saber cómo ha de redondear. Tampoco es fácil para el profesor explicar qué criterio se va a utilizar. Otros criterios son más simples, como el error absoluto, pero en cambio puede ser complicado para el profesor definir el umbral de corrección: se podría exigir demasiado o, en el otro extremo, se podrían aceptar como válidas respuestas con errores garrafales.

Propuestas

- *Probar con varios criterios de corrección*
- *Probar con diferente número de intentos*
- *Crear un mensaje orientador (esto no es el término independiente, parece que lo has confundido con la pendiente)*
- *Añadir en el enunciado un gráfico de la nube de puntos*
- *Añadir los datos*

Un enlace en la parte inferior permite acceder a la página que contiene la segunda versión del ejercicio.

Lo que se plantea es crear un problema nuevo, en vez de añadir algunos cambios a la primera versión, lo cual permitirá al alumno conservar la secuencia de versiones. Para ello recomendamos hacer las siguientes operaciones:

- En la página principal del primer problema, generar una copia que podremos guardar localmente en un fichero XML: «Generar XML».
- Desde el editor, marcar la carpeta de trabajo y elegir la opción «Crear problema».
- Antes de escribir nada sobre los campos de este nuevo problema, pulsar sobre «Restaurar XML», elegir el fichero que se ha generado previamente y aceptar: en este momento tenemos una copia idéntica del primer problema, que procederemos a modificar según las indicaciones que figuran en la Tabla 3.

El texto destacado en color amarillo indica novedades. Por ejemplo, el enunciado es ligeramente diferente. Se le ha dado formato con *tags* de html para crear una tabla, en la que en la parte izquierda figurará el texto propiamente dicho y a la derecha aparecerá un gráfico. Para poder editar explícitamente *tags* de html (o, como en este caso, para poder copiar directamente dicho código), hay que seleccionar en el editor que aparece en la página principal del problema el botón marcado como «Fuente HTML».

Un tema de regresión lineal (segunda versión)	
Descripción	**Estudio con dos variables numéricas, modelo lineal.**
Enunciado	**Incluimos el gráfico:** **<table align=”left” border=”0” cellpadding=”1” cellspacing=”4”>** **<tbody>** **<tr>** **<td style=”width:400px”> En este caso vamos a poner aquí la historia que haga referencia a un** **estudio en el que se observan dos variables, X e Y, de las que se sabe:** ** ** ** - la media de X: Mx ** ** - la media de Y: My ** ** - la desviación de X: Sx ** ** - la desviación de Y: Sy ** ** - la covariancia de X e Y: Vxy ** ** ** **</td>** **<td>$plos$</td>** **</tr>** **</tbody>** **</table>** ** **

Modelo R (Expresiones)	**n = sample(40:100, 1)
	R = runif(1,0.6,1)
	sgn = sample(c(-1,1), 1)
	XB = rnorm(3, 0, 50)
	x = round(seq(XB[1], XB[2], len=n), 3)
	vx = var(x)
	v.i = exp(rnorm(1, 0, 2))
	v.e = v.i/(1/R-1)
	vy = v.i+v.e
	sig = sqrt(v.i)
	err = rnorm(n)*sig
	y = round(XB[3] + sgn*R*sqrt(vy/vx) * x + err, 3)
	Mx = round(sum(x)/n,5)
	My = round(sum(y)/n,5)
	Vxy = round(cov(x,y),5)
	Sx = round(sd(x),5)
	Sy = round(sd(y),5)
	LM = lm(y ~x)
	summ = summary(LM)
	tind = summ$coef[1,1]
	#Atención:
	plos = ini_imagen(360, 360)
	par(mar=c(3,3,1,1))
	plot(x, y, pch=19)
	abline(LM, col='grey')
	fin_imagen()
	pend = summ$coef[2,1]**

Expresiones públicas	**Para compartir el modelo con otros autores.**
Variables extras a mostrar	**x, y**

Tabla 3. Datos principales para la segunda versión del ejercicio

Nuevamente se pulsa sobre el botón «Guardar», y seguidamente se elige la opción «Editar preguntas». Nótese que como se trata de una copia del primer problema ya existe una pregunta, que procedemos a editar y a modificar como se indica en la Tabla 4.

Preguntas	
Enunciado 1	**Si queremos predecir la Y con la X, utilizando un modelo lineal, ¿cuánto vale el término independiente de la recta estimada (usando el método de mínimos cuadrados)?**
Solución a la pregunta	**Tind**
Parámetro de sintaxis	
Sintaxis de la respuesta	**Real**
Plantilla de expresiones de la respuesta	**Número decimales correctos**
Parámetro de expresiones	**2**

Expresiones de respuesta	**Adaptamos el código base que aparece:
	#si entran parametro_ = 4 , el valor absoluto deberá ser igual o menor de 0.00005 (4 decimales correctos)
	if (estatus.decimales_correctos(solucion_ , respuesta_ , parametro_))
	{
	resultado_ <- 1
	}else{
	resultado_ <- 0
	if (estatus.decimales_correctos(pend , respuesta_ , parametro_))
	{
	mensaje_ = 'Parece que estás confundiendo "término independiente" con "pendiente".'
	}
	}**

Peso	**1**
Ayuda contextual	**Aquí va un enlace en donde puedes leer más sobre el método:**

Link	**<http://es.wikipedia.org/wiki/Regresi%C3%B3n_lineal#Eqnref_12>**
Intentos	**2**
Mostrar solución	**(normalmente es lo más sensato)**
Mensaje correcto	**¡Bravo!**
Mensaje semicorrecto	
Mensaje incorrecto	**¡La próxima vez lo harás mejor!**

Tabla 4. Datos de la pregunta para la segunda versión del ejercicio

Comentario

En este caso se incluye un control (rudimentario) para verificar que el valor proporcionado por el estudiante no sea el coeficiente de la pendiente, un error común. En caso afirmativo, aparecerá un texto destacado. También se podría crear al vuelo el texto que aparece en las cajas «correcto», «semicorrecto» o «incorrecto».

Propuestas

- *Introducir otras preguntas relacionadas con el tema, añadiendo las expresiones en R necesarias para generar la solución.*

Como se puede apreciar si se realiza la prueba del problema, se incorpora el gráfico al enunciado, en el que se representa no solo la nube de puntos sino también la recta de regresión (con lo que el estudiante dispone de una valiosa ayuda). También hemos incluido los datos, a los que se puede acceder mediante el enlace «Copiar datos para pegar en otro programa»: este enlace conduce a una pestaña del navegador que posibilita la copia de datos, con facilidades para determinar el símbolo decimal o el carácter que separa las columnas.

Sin embargo, el cambio más importante debe verse en las posibilidades que ofrece el código que acompaña la evaluación de la pregunta. Puesto que el autor del problema puede controlar cualquier aspecto que se pueda implementar en código R, las posibilidades para evaluar flexiblemente y para dar *feedback* adecuado aumentan considerablemente.

La tercera versión del problema amplía el número de preguntas. Con la que se añade se pretende comprobar si el estudiante entiende el concepto de residuos en el modelo lineal y si lo sabe aplicar en casos concretos. Con este fin, se muestran tres gráficos estándar de residuos, aunque solo uno de ellos es el correcto. El reto para el autor de problemas consiste en saber generar los gráficos falsos, que resulten creíbles pero no tanto como para que solamente un ojo experto pueda discernir el auténtico de los demás. Este no es más que un caso particular de pregunta de opción única que, como todo maestro sabe, tiene que contener siempre diversas opciones realistas (al menos una que pueda competir con la solución real), únicamente que con e-status deben crearse mediante un modelo probabilístico lo bastante robusto como para que el riesgo de que aparezca una pregunta demasiado fácil o demasiado difícil sea mínimo.

La tercera versión puede crearse a partir de una copia local generada en XML de la segunda, tal como se procedió la vez anterior; de esta manera conservamos todas las versiones.

Un tema de regresión lineal (tercera versión)	
Descripción	Estudio con dos variables numéricas, modelo lineal.
Enunciado	<table align=”left” border=”0” cellpadding=”1” cellspacing=”4”> <tbody> <tr> <td style=”width:400px”> En este caso vamos a poner aquí la historia que haga referencia a un estudio en el que se observan dos variables, X e Y, de las que se sabe: - la media de X: Mx - la media de Y: My - la desviación de X: Sx - la desviación de Y: Sy - la covariancia de X e Y: Vxy </td> <td>$plos$</td> </tr> </tbody> </table>

Modelo R (Expresiones)	n = sample(40:100, 1)
	R = runif(1,0.6,1)
	sgn = sample(c(-1,1), 1)
	XB = rnorm(3, 0, 50)
	x = round(seq(XB[1], XB[2], len=n), 3)
	vx = var(x)
	v.i = exp(rnorm(1, 0, 2))
	v.e = v.i/(1/R-1)
	vy = v.i+v.e
	sig = sqrt(v.i)
	err = rnorm(n)*sig
	y = round(XB[3] + sgn*R*sqrt(vy/vx) * x + err, 3)
	Mx = round(sum(x)/n,5)
	My = round(sum(y)/n,5)
	Vxy = round(cov(x,y),5)
	Sx = round(sd(x),5)
	Sy = round(sd(y),5)
	LM = lm(y ~x)
	summ = summary(LM)
	tind = summ$coef[1,1]
	plos = ini_imagen(360, 360)
	par(mar=c(3,3,1,1))
	plot(x, y, pch=19)
	abline(LM, col=’grey’)
	fin_imagen()
	pend = summ$coef[2,1]
	#Atención:
	u = sample(1:3, 1)
	d=0.66
	u.2 = c(d^2, d, 1, d^(-1), d^(-2))[u:(u+2)]
	Rs = array(NA, dim=c(n, 3))
	for (i in 1:3) {
	 j = u.2[i]
	 if (j==1) { Rs[,i] = summ$resid }
	 else {Rs[,i] = sample(qnorm(seq(0.5/n, 1, by=1/n)))* summ$sigma*j}
	}
	ylym=range(Rs)
	fig = ini_imagen(640,250)
	opar=par()
	par(mfrow=c(1,3),mar=c(3.5,2.5,1,0.3))
	for (i in 1:3) {
	 plot(x,Rs[,i], main=i, ylim=ylym, pch=4)
	}
	par(opar)
	fin_imagen()
	corre = 4-u

Expresiones públicas	Para compartir el modelo con otros autores.
Variables extras a mostrar	x, y

Tabla 5. Datos principales para la tercera versión del ejercicio

De nuevo guardamos y procedemos a «Editar preguntas». En esta ocasión volvemos a crear una nueva pregunta, completándola con los contenidos de la Tabla 6. La Figura 23 muestra el aspecto final de la pregunta.

Preguntas	
Enunciado 2 **Nueva pregunta**	**Una de estas tres figuras corresponde al diagrama de los residuos para los datos mostrados ajustados con el modelo lineal. Diga con un número de 1 a 3 cuál es la imagen correcta.** **fig**
Solución a la pregunta	**Corre**
Parámetro de sintaxis	
Sintaxis de la respuesta	**Natural**
Plantilla de expresiones de la respuesta	**Exacto**
Parámetro de expresiones	
Expresiones de respuesta	**if (solucion_ == respuesta_)** **{** **resultado_ <- 1** **}else{** **resultado_ <- 0** **}**
Peso	**1**
Ayuda contextual	
Link	
Intentos	**1**
Mostrar solución	
Mensaje correcto	**Excelente**
Mensaje semicorrecto	
Mensaje incorrecto	

Tabla 6. Datos de la pregunta para la tercera versión del ejercicio

Comentario

¿Qué hemos hecho? El objetivo consiste en identificar la estructura de los residuos. Se han generado tres gráficos, uno de los cuales es real y los otros no.

La gracia de la pregunta consiste en saber hacer de trilero: colocar el gráfico correcto en una posición impredecible, y que tampoco ayude la magnitud relativa de la desviación residual.

Aunque frecuentemente las preguntas que se construyan esperan una respuesta numérica abierta, en e-status es posible preparar preguntas de opción múltiple. Esto será normalmente la única posibilidad si se desea que el estudiante valore la corrección de distintas afirmaciones, para detectar cuál o cuáles son válidas, dado que la técnica para interpretar respuestas textuales abiertas es muy compleja. Por supuesto, las opciones de la pregunta pueden ser también gráficos, como en este caso, expresiones matemáticas o simplemente números.

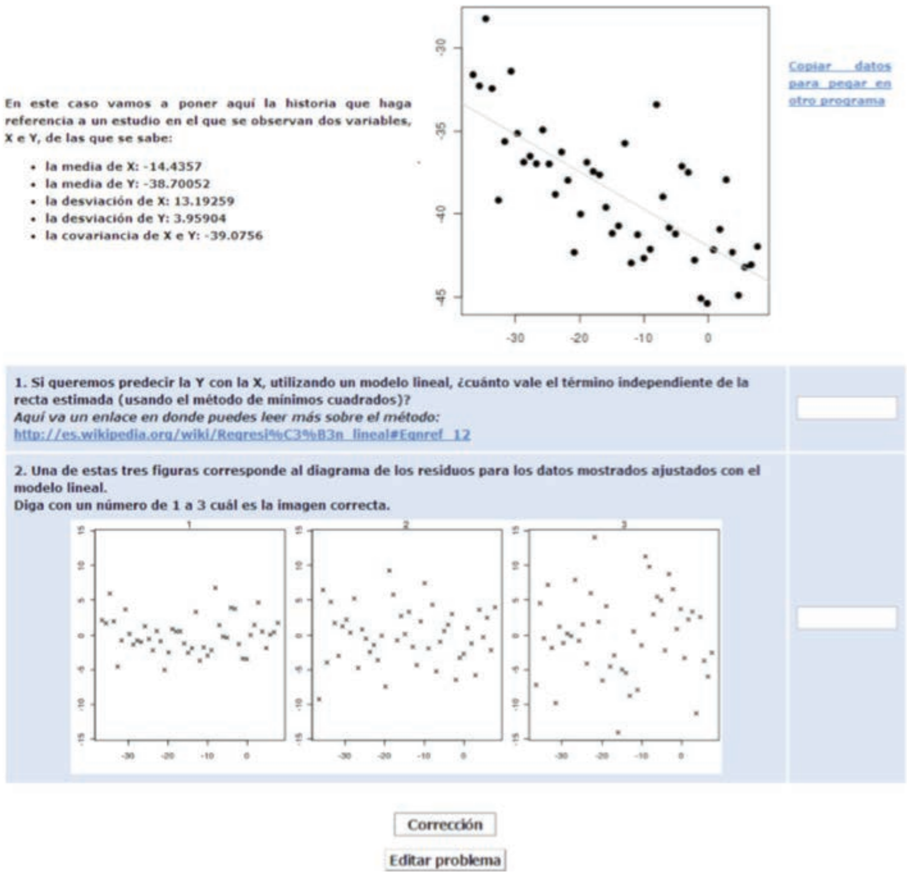

Figura 23. Aspecto de la tercera versión del problema

No hay una forma sencilla de programar preguntas de opción múltiple todavía. Generalmente, el profesor ante la pregunta que quiere plantear elige —además de la correcta— varias posibilidades que le parezcan típicos errores. Después aplica una permutación a las opciones y registra cuál de ellas es la solución. Veamos un ejemplo.

Queremos preguntar: «¿Cuál es el valor que empleamos para un intervalo de confianza del 90% para la media, con una muestra de tamaño n, y σ desconocida?». Tomamos 4 opciones:

```
Op = c(qt(0.95, n-1), qnorm(0.95), qt(0.95, n), qt(0.975, n-1))
```

Desordenamos las opciones y anotamos la correcta:

```
u = sample(1:4)
Ok = which(u==1) # ¿dónde está el 1? Es la opción correcta.
```

Preparamos una lista numerada con las opciones, que se mostrará en el enunciado a continuación de la pregunta anterior (como símbolo: $lista$):

```
lista = "<ol>"
for (i in 1:4) {
 lista = paste(lista, "<li>", round(Op[u[i]], 4), "</li>")
}
lista = paste(lista, "</ol>")
```

Para esta pregunta la respuesta correcta corresponde a la variable «Ok», y se utiliza la misma plantilla «Exacto», porque se trata de un valor entero y no caben problemas de precisión.

La cuarta versión del problema aborda una técnica que puede ser útil en algunas ocasiones. Consiste en interconectar las diversas preguntas del problema, lo cual permite que el profesor incluya cuestiones cuyo fin es pedir al estudiante que introduzca valores para ser incorporados al problema. Es como cuando un mago invita a un espectador a subir al escenario para participar en primera persona del juego (salvando las distancias).

Antes de continuar, digamos que el código de las preguntas puede utilizar ciertas variables propias de e-status que le permiten acceder a cierto tipo de información, o cambiar el estado del problema. Algunas ya las conocemos:

- solucion_: la solución correcta, si está definida.
- respuesta_: lo que el alumno ha introducido.
- resultado_: la nota de la pregunta.
- mensaje_: un texto que aparece tras hacer la corrección de la pregunta.
- respuestas_: la lista de todas las respuestas del alumno.
- resultados_: las notas para todas las preguntas.
- chances_: el número de intentos disponibles.

Además, las variables que se definan tanto en el código principal como en el de las preguntas previas son visibles en el ámbito de cada pregunta (por tanto, hay que procurar no repetir nombres para las variables que deban ser visibles en otros ámbitos). Todo ello permite elaborar este ejercicio: consiste en pedir al alumno que defina una supuesta nueva observación, entrando el valor de X y el de Y. Se le pedirá que este nuevo punto, a unir a un subconjunto determinado de los originales, no sea influyente (ni respecto las X ni respecto las Y). A continuación, se le muestra el gráfico con los puntos seleccionados y el punto del alumno (marcado en color rojo) y se le pide un cálculo para este conjunto, concretamente, un intervalo de confianza para la pendiente.

Un tema de regresión lineal (cuarta versión)	
Descripción	**Estudio con dos variables numéricas, modelo lineal.**
Enunciado	**<table align="left" border="0" cellpadding="1" cellspacing="4"> <tbody> <tr> <td style="width:400px"> En este caso vamos a poner aquí la historia que haga referencia a un estudio en el que se observan dos variables, X e Y, de las que se sabe: - la media de X: Mx - la media de Y: My - la desviación de X: Sx - la desviación de Y: Sy - la covariancia de X e Y: Vxy </td> <td>$plos$</td> </tr> </tbody> </table> **

<table>
<tr><td>Modelo R (Expresiones)</td><td><pre>
n = sample(40:100, 1)
R = runif(1,0.6,1)
sgn = sample(c(-1,1), 1)
XB = rnorm(3, 0, 50)
x = round(seq(XB[1], XB[2], len=n), 3)
vx = var(x)
v.i = exp(rnorm(1, 0, 2))
v.e = v.i/(1/R-1)
vy = v.i+v.e
sig = sqrt(v.i)
err = rnorm(n)*sig
y = round(XB[3] + sgn*R*sqrt(vy/vx) * x + err, 3)
Mx = round(sum(x)/n,5)
My = round(sum(y)/n,5)
Vxy = round(cov(x,y),5)
Sx = round(sd(x),5)
Sy = round(sd(y),5)
LM = lm(y ~x)
summ = summary(LM)
tind = summ$coef[1,1]
plos = ini_imagen(360, 360)
par(mar=c(3,3,1,1))
plot(x, y, pch=19)
abline(LM, col='grey')
fin_imagen()
pend = summ$coef[2,1]
u = sample(1:3, 1)
d=0.66
u.2 = c(d^2, d, 1, d^(-1), d^(-2))[u:(u+2)]
Rs = array(NA, dim=c(n, 3))
for (i in 1:3) {
 j = u.2[i]
 if (j==1) { Rs[,i] = summ$resid }
 else {Rs[,i] = sample(qnorm(seq(0.5/n, 1, by=1/n)))* summ$sigma*j}
}
ylym=range(Rs)
fig = ini_imagen(640,250)
opar=par()
par(mfrow=c(1,3),mar=c(3.5,2.5,1,0.3))
for (i in 1:3) {
 plot(x,Rs[,i], main=i, ylim=ylym, pch=4)
}
par(opar)
fin_imagen()
corre = 4-u
#Atención:lista = seq(from=1, len=9, by=round(n/9))
</pre></td></tr>
<tr><td>Expresiones públicas</td><td>Para compartir el modelo con otros autores.</td></tr>
<tr><td>Variables extras a mostrar</td><td>x, y</td></tr>
</table>

Tabla 7. Datos principales para la cuarta versión del ejercicio

La Tabla 7 no es muy diferente de la tabla correspondiente a la tercera versión: solo define el subconjunto de los puntos que serán utilizados para la cuarta pregunta. La Tabla 8 contiene la innovación real de este ejercicio (atención: hay que crear dos preguntas).

Preguntas	
Enunciado 3 Nueva pregunta	Escoja los pares de observaciones de los datos proporcionados correspondientes a las posiciones: $lista$. Añada un punto más (le pedimos que su punto siga más o menos la tendencia general), introduciendo primero la coordenada X y luego la Y, separadas por un blanco.
Solución a la pregunta	(no relevante)
Parámetro de sintaxis	2
Sintaxis de la respuesta	Vector
Plantilla de expresiones de la respuesta	(no relevante)
Parámetro de expresiones	

Expresiones de respuesta	Escriba todo esto:
	nvo = respuesta_
	x2 = c(x[lista], nvo[1])
	y2 = c(y[lista], nvo[2])
	lm2 = lm(y2 ~ x2)
	su2 = summary(lm2)
	sres = lm2$resid[10]/su2$sigma
	mensaje_=””
	resultado_ = 1
	if (abs((nvo[1]-Mx)/Sx) > 3) {
	 mensaje_ = “La abcisa está muy alejada: este punto sería muy influyente.”
	 resultado_ = 0
	}
	if (abs(sres) > 2) {
	 mensaje_ = paste(mensaje_, “Parece que tu punto se aleja bastante del resto.”)
	 resultado_ = 0
	} else {
	 plos2 = ini_imagen(300, 300)
	 par(mar=c(3,3,1,1))
	 plot(x2, y2, pch=19, col=c(rep(1, 9), 2))
	 fin_imagen()
	}

Peso	1
Ayuda contextual	
Link	
Intentos	2
Mostrar solución	¡No!
Mensaje correcto	Este es tu nuevo gráfico: $plos2$
Mensaje semicorrecto	
Mensaje incorrecto	
Enunciado 4 Nueva pregunta	Con los datos de la pregunta anterior (incluso si la respuesta no fue aceptable), calcule un intervalo de confianza al 95% para la pendiente de la recta $Y = \beta0 + \beta1X$ (separe los dos valores por un espacio, y proporcione por lo menos tres decimales correctos).
Solución a la pregunta	(no relevante)
Parámetro de sintaxis	2
Sintaxis de la respuesta	Vector
Plantilla de expresiones de la respuesta	(no relevante)

Parámetro de expresiones	**3 (decimales correctos)**
Expresiones de respuesta	**Escriba todo esto:** **if (!exists("x2")) {** **mensaje_ = "Debe responder antes la pregunta previa."** **} else {** **tam = length(x2)** **solu = su2$coef[2,1] + c(-1,1)*qt(0.975, tam-2)*su2$coef[2,2]** **solu = round(solu, 6)** **if (respuesta_[1] > respuesta_[2]) respuesta_ = respuesta_[2:1]** **resultado_ = 0** **len_res = length(respuesta_)** **for (i in 1:len_res) {** **if (estatus.decimales_correctos(solu[i], respuesta_[i], parametro_)) resultado_ = resultado_ + 1/ len_res** **}** **if (chances_ > 0) {** **msg = "¡Animo!"** **} else {** **msg = paste("La respuesta correcta es (", solu[1], ", ", solu[2], ")", sep='')** **}** **}**
Peso	**1**
Ayuda contextual	
Link	
Intentos	**2**
Mostrar solución	**¡No!**
Mensaje correcto	
Mensaje semicorrecto	**¡Qué rabia! Has fallado uno de los dos extremos. msg**
Mensaje incorrecto	**msg**

Tabla 8. Datos de la pregunta para la cuarta versión del ejercicio

Comentario

¿Qué hemos hecho? Las preguntas en sí puede que no sean lo más interesante: estaríamos comprobando si el alumno puede imaginar un punto que no sea flagrantemente contrario a la tendencia de la nube representada. También le pedimos un cálculo para estimar por IC la pendiente. Requerirá habilidades de manejo de una herramienta estadística, para seleccionar los valores especificados, realizar el ajuste y obtener el IC.

Una lectura diferente es ilustrar algunas posibilidades que ofrece la herramienta. Un profesor puede inventarse preguntas que no posean solución única (basta con que haya un procedimiento para valorar la corrección de la respuesta, que se pueda implementar con R).

Además, vemos que las preguntas tienen conexión entre ellas (comparten el código, y es posible saber qué se contestó a cierta pregunta y el resultado de dicha evaluación).

La consecuencia obvia del aumento de complejidad en la evaluación de las respuestas es: cuidado, aumenta el número de situaciones posibles que hay que prever, y se incrementa el riesgo de cometer errores.

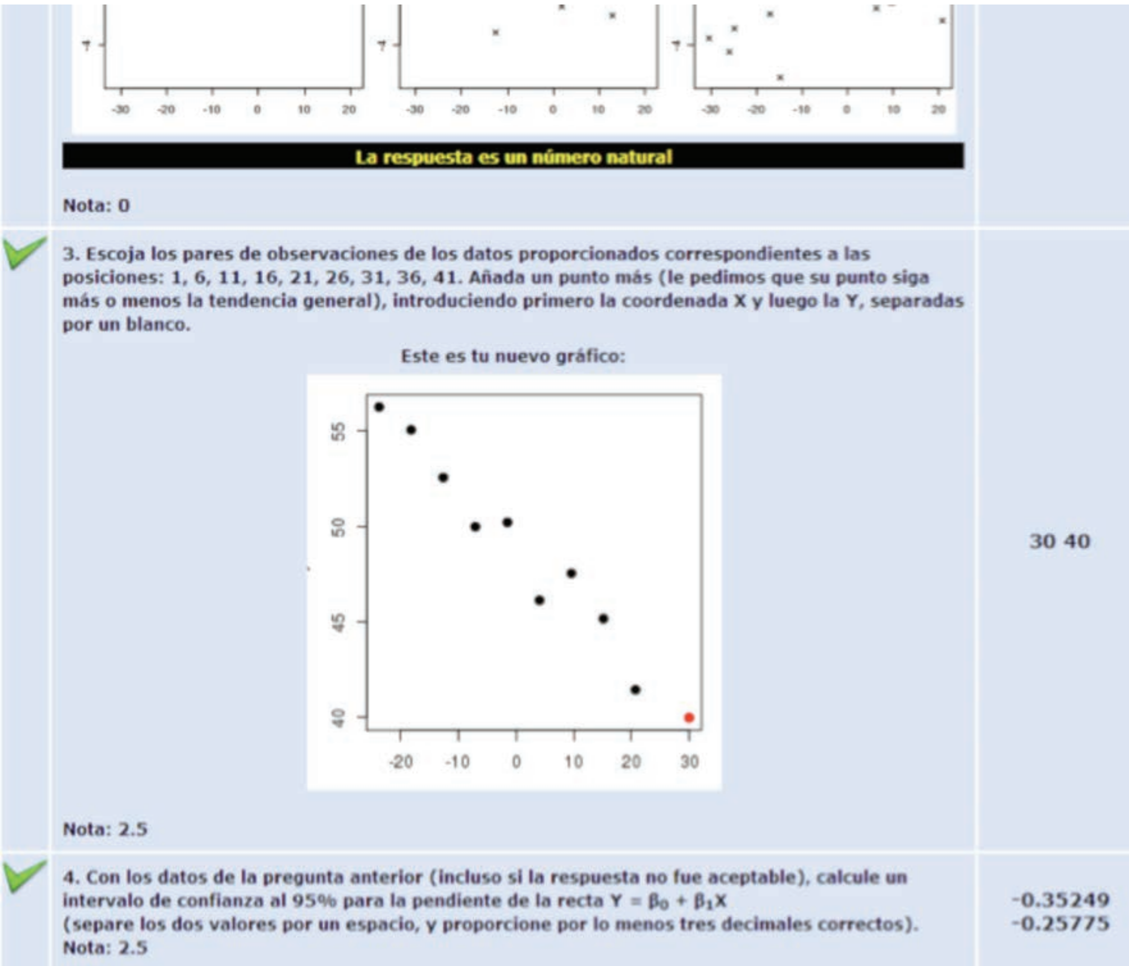

Figura 24. Aspecto de la cuarta versión del problema

La Figura 24 ilustra cómo podría ser un resultado de la solución de estas últimas dos preguntas. El punto introducido (30, 40) es una opción muy razonable —se observa en el gráfico que se muestra tras responder que dicho punto está perfectamente alineado con la tendencia de los puntos seleccionados—. El intervalo consecuente también debe encajar bien con la pendiente de la nube original, si el alumno se ha preocupado por comparar. Por otro lado, si se hubiera introducido un punto *discordante* (hasta lo permitido), el estudiante habría podido comprobar con su propio caso como el nuevo punto influye y altera la magnitud de la pendiente, lo cual también sería un buen ejemplo de *descubrimiento* capaz de asentar mejor el conocimiento.

Con este procedimiento podemos incorporar al problema preguntas que van a exigir al estudiante un esfuerzo diferente del requerido para calcular respuestas únicas, que algunas veces resultan ser mecánicas y previsibles, en cierta manera. También es cierto que se requiere un control más exigente, si se vinculan unas preguntas con otras, más aún si una pregunta depende de una respuesta introducida por el estudiante. Sencillamente, las cosas pueden ir mal, y además es muy conveniente proporcionar explicaciones al estudiante ante respuestas incoherentes.

Por ejemplo: podríamos pedir al estudiante que complete una determinada distribución de probabilidad de una variable discreta. Después le pediremos que halle la variancia de dicha variable. Si las probabilidades resultantes no son positivas o no suman 1 deberíamos notificar el error. Pero si los intentos disponibles para contestar la primera pregunta se agotan antes de responder correctamente, la segunda no se puede contestar, y ambas se puntuarán como incorrectas (aunque el alumno supiera cómo responder a la segunda pregunta).

Puede parecer innecesario decirlo, pero cualquier problema debe ser intensamente validado antes de ponerlo a disposición del alumno; más aún si es elaborado. El coste de rectificar un problema cuando ya ha sido probado por los alumnos es cuantioso, puesto que no es fácil ni recomendable eliminar las resoluciones defectuosas de la base de datos.

Discusión

El uso de las tecnologías de la información en la docencia de estadística y ramas afines es intenso pero aún no decididamente claro. No es de extrañar, puesto que el sistema actual, que concentra un gran volumen de contenidos en pocas semanas de clase, implica que el profesor sigue tomando el protagonismo porque entiende que es la única manera de cumplir con el compromiso de la guía docente correspondiente. En cambio, las TIC son herramientas para el alumno, a utilizar con el ritmo particular que cada uno sabe que le va bien. Si es que el alumno percibe que la herramienta le aporta algo: no basta con poner los utensilios a su alcance.

El profesor tiene que seleccionar los medios más apropiados para complementar sus clases, y tiene que comprobar que el alumno asocia unos temas con otros. Esta fase de seguimiento es crucial, no basta con saber que en el examen todo saldrá a la luz. No confundamos la evaluación con la calificación. La diferencia está en que tras la evaluación el profesor y los alumnos tienen aún una oportunidad de reconducir el aprendizaje.

Como es normal, los objetivos que persigue el estudiante son variados, y simplificando bastante podríamos decir que son una combinación entre aprobar la asignatura y aprender algo que le pueda ser de provecho en el futuro. Cuando la estadística es una materia relativamente marginal dentro de un plan de estudios, el balance entre estos dos objetivos suele inclinarse llamativamente hacia el primero de ellos. Posiblemente, el profesor se verá obligado a establecer una parte de la calificación en base de los ejercicios resueltos con los recursos TIC empleados, aunque sepa que estos resultados tienen una validez discutible. De lo contrario, hay pocas garantías de que los estudiantes dediquen parte de su tiempo a un esfuerzo que no les reporta un beneficio inmediato (aprender no se percibe así).

El taller que se explica en esta comunicación puede que insista mucho en «lo que se puede hacer», pero indudablemente tiene mucha importancia «lo que se debe hacer», y seguramente más aún. Las posibilidades técnicas de una herramienta al fin y al cabo son medios para llegar a un objetivo, pero lo que guía a un docente cuando diseña la colección de problemas que complementa al resto de materiales es la intención de que el alumno se excite descubriendo conocimientos que quizá él mismo no sospechaba. Dado que el *partenaire* del alumno no es el profesor sino una máquina que responde a un algoritmo, esta no le va a ofrecer una ayuda muy sofisticada o, al menos, será tan rica como la imaginación del autor de los problemas y su habilidad con R le permitan. El buen problema ha de obligar a pensar, al menos al principio (cuando se gasta y pierde la magia, se vuelve mecánico y aburrido), y en esa actividad el alumno interpreta la situación, analiza diversas posibilidades, escoge y descarta procedimientos y, en suma, interviene activamente tratando de utilizar sus recursos para construir nuevo conocimiento.

Referencias

[1] González, J. A.; Muñoz, P. (2006): «e-status: an Automatic Web-Based Problem generator - Applications to Statistics», Computer Applications In Engineering Education, V14(2), 151-159.

[2] González, J. A.; Jover, L.; Cobo, E.; Muñoz, P. (2010): «A Web-Based Learning Tool Improves Student Performance In Statistics: A Randomized Masked Trial», Computers & Education, V55(2), 704-713.