

Ejemplo comparativo de eficiencia educativa en Física, usando la prueba t de *student* con R

EDVCATIO PHYSICORVM



ISSN 1870-9095

Rubén Sánchez Sánchez, César Mora

Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada del Instituto Politécnico Nacional, CP 15,100, Ciudad de México, México.

E-mail: rsanchezs@ipn.mx

(Recibido el 2 de febrero de 2023, aceptado el 28 de febrero)

Resumen

La prueba t de *student* puede ser utilizada en educación cuando se comparan los resultados de evaluación para dos grupos e inferir si los datos son lo suficientemente diferentes. Por ejemplo, si tenemos dos grupos uno de prueba (Grupo A) que siga la enseñanza tradicional y otro que tenga en su proceso de aprendizaje una técnica de aprendizaje con metodología didáctica (Grupo B). Podemos comparar los resultados de evaluación de ambos grupos y tratar de ver si los resultados obtenidos en B son relativamente diferentes y mejores que los obtenidos por el grupo A de prueba. Suponiendo que la distribución de la evaluación en ambos grupos es aproximadamente normal, entonces la prueba t de *student*, puede darnos un criterio cuantitativo, para comprobar si hubo algún cambio significativo en el aprendizaje promedio de ambos grupos. Aquí mostramos la técnica usando un ejemplo muestral ficticio sencillo, usando el software estadístico gratuito R , y otras utilidades de software gratuito que no representarán un gasto importante para un investigador en la educación. Esperamos que el sencillo ejemplo sea de utilidad a la comunidad de profesores, en especial los profesores de Física y Ciencias afines.

Palabras clave: Software Educativo, Software R , Educación, Pruebas estadísticas.

Abstract

The *student's t* test can be used in education when comparing assessment results for two groups and inferring whether the data are different enough. For example, if we have two groups, one is a test group (Group A) that follows traditional teaching and another that has a learning technique with didactic methodology in its learning process (Group B). We can compare the evaluation results of both groups and try to see if the results obtained in B are relatively different and better than those obtained by the group A. Assuming that the distribution of the evaluation in both groups is approximately normal, then the *student's t* test can give us a quantitative criterion to check if there was any significant change in the average learning of both groups. Here we show the technique using a simple fictitious sample example, using the free statistical software R , and other free software utilities that will not represent a significant expense for an educational researcher. We hope that this simple example will be useful to the community of teachers, especially in Physics and related Sciences.

Keywords: Educational Software, R Software, Education, Statistical proofs.

I. INTRODUCCIÓN

En el campo de la educación, en particular de la educación de la Física o de las Ciencias, generalmente el docente quiere implementar una metodología didáctica que le sirva para que los estudiantes aprendan mejor los conceptos de esta área de la Ciencia que tiene bastantes detalles, que pueden presentar cierta dificultad para aprenderlas. En este trabajo, mostramos una técnica que puede ser de utilidad para dilucidar si la metodología bajo estudio está funcionando, para alcanzar un mejor nivel de aprendizaje. De aquí nace la necesidad de aplicar técnicas cuantitativas que nos permitan apreciar si una metodología educativa es conveniente, para la enseñanza, en este caso de conceptos Físicos.

En este trabajo mostramos como se puede aplicar la prueba t de *student*, para apreciar si existen ventajas en la metodología educativa aplicada. Lo que queremos es mostrar la técnica con un ejemplo hipotético muestral de dos grupos

cuyas evaluaciones siguen aproximadamente una distribución normal. Si el promedio en evaluación de un grupo A es diferente de la de un grupo B, en más de un cierto valor determinado con la t de *student*. Entonces tendremos elementos cuantitativos para esperar que la metodología didáctica tiene ventajas, sobre la enseñanza tradicional. Aquí mostramos como emplear ciertos recursos gratuitos para realizar la prueba t de *student*, como un apoyo cuantitativo, a la hora de realizar una investigación en el campo educativo.

II. EL SOFTWARE R

El software estadístico R , se inició en 1991 con Ross Ihaka y Robert Gentleman en el Departamento de Estadística de la Universidad de Auckland en Nueva Zelanda [1]. Como consecuencia de que S , otra herramienta de software que tiene el mismo fin, cambiaba de dueño para una empresa de

software. La filosofía de *R*, es diferente a la de *S*, que sigue el servicio y aplicación de la estadística para el sector privado. Generalmente *R* es utilizado en las Universidades y para apoyar al sector educativo y científico, en la estadística. Actualmente *R* tiene un sitio de internet popular [2], de donde se distribuye para el beneficio del sector público, y para apoyar la investigación en las Universidades. Como forma de ocupar *R* para el siguiente ejemplo, se recomienda revisar un manual de *R*, sobre distribuciones [3]. Como preludeo a este ejemplo, necesitamos instalar *R* en el computador, existen varias formas de instalar *R*, en un sistema operativo. Por ejemplo, para instalarlo en una Mac, se puede utilizar el manejador de paquetes *Homebrew* [4], y este método también según la página de *Homebrew*, está disponible para Linux. Otro método para instalar *R*, es bajando una distribución particular de la misma página web de *R*.

A manera de ejemplo, la forma de instalar *R*, utilizando *Homebrew* en una Mac [5], sería abriendo una terminal en el computador y dando las siguientes órdenes para instalarlo:

```
> brew tap homebrew/science
> brew install --cask r
```

Después de lo cual el computador Mac tendrá una copia de *R*, en el folder de Aplicaciones. Si se quieren revisar detalles de la instalación, se le puede pedir a *Homebrew* un reporte de ella, con la orden de línea de comando siguiente:

```
> brew info r
```

En las siguientes secciones, describiremos como utilizar *R*, y software de respaldo, para poder utilizar el estadístico de la *t* de *student*, aplicada a la investigación educativa.

III. SOFTWARE DE EDICIÓN

Además del software principal, que en este caso es el lenguaje de programación *R*, podemos usar otro software de apoyo, para la edición de programas en *R*, sin tener que escribirlo directamente en *R*. La ventaja de utilizar un software aparte de edición, es la comodidad para guardar el archivo fuente. Sin un software de apoyo de edición, normalmente podemos escribir el código fuente en un block de notas normal, sin apoyo de una interfaz para *R*. En cambio, al utilizar un software de edición de código, con apoyo de ejecución a través de *R*, tenemos la ventaja de que podemos guardar y corregir el código *R*, en cualquier momento, y además tenemos la opción de probar el código, corriéndolo de manera parcial, mientras estamos preparándolo, escribiéndolo y corrigiéndolo.

Hay varios softwares de este tipo, entre ellos están *RStudio* [6], *Notepad++* [7], *Emacs* [8] y *Neovim* [9, 10]. Cada uno de ellos, tienen ventajas, que permiten el uso de *R* de una manera interactiva y cómoda. En este trabajo describiremos brevemente como utilizar *Emacs* con *R*. Para esto, será necesario, indicar como instalar y configurar el editor de texto, para obtener un ambiente de trabajo cómodo y conveniente. De tal manera que el código fuente del

programa en *R*, pueda ser manejado y mantenido con facilidad.

IV. EDICIÓN DE *R* CON *EMACS* Y *ESS*

Emacs empezó en el Laboratorio de Inteligencia Artificial en el Massachusetts Institute of Technology (MIT) [11, 12], en 1974 como un proyecto desarrollado por el Ingeniero y Físico Richard Matthew Stallman [13, 14], que ha impulsado el uso del *software libre*.

Obviamente, primero hay que instalar *Emacs* en la computadora. Para este escrito usaremos la versión de *Emacs* para Mac, llamado *emacs-mac* [15] que usando *Homebrew* se instala con las sencillas orden de comando [16]

```
> brew tap railwaycat/emacs-macport
> brew install --cask emacs-mac
```

Después y para preparar *Emacs* en la edición y ejecución de *R*, se necesita, configurarlo. Vamos a suponer que estamos en un sistema operativo tipo Mac, la instalación en Windows debería ser muy similar. Si denotamos por el símbolo \sim , el directorio hogar del computador. Para un ejemplo sencillo, utilizando la convención o especificación XDG [17-19], para los archivos de configuración. Tenemos que escribir un archivo de configuración para *Emacs* en el directorio

```
~/config/emacs
```

El archivo se llama (por convención estándar) “init.el”, en él se recomienda escribir lo siguiente en lenguaje *elisp* [20]:

```
(require 'package)
;; Any add to list for package-archives (to add marmalade or
;; melpa) goes here
(add-to-list 'package-archives
  ("MELPA"
   "http://melpa.org/packages/")
  t)
(package-initialize)
```

Para instalar el sistema de paquetes estándar MELPA [21], en *Emacs*. Con este apoyo, y un procedimiento estándar para instalar paquetes, debemos de instalar el paquete *Emacs Speaks Statistics (ESS)* [22], que es la interfaz de *Emacs* para manejar e interaccionar con *R*. El procedimiento para instalar *ESS*, es primero correr el programa *Emacs*, luego acceder al minibuffer de *Emacs* oprimiendo Command y x (M-x) al mismo tiempo, y colocando la instrucción “package-list-packages”, buscando el paquete “ess” y luego instalarlo de la lista de paquetes. Hay muchos tutoriales en internet para instalar paquetes en *Emacs* [23], para realizar esta instalación.

Una vez instalado *ESS*, se recomienda copiar y configurar *Emacs* de acuerdo, al archivo de configuración de Paul Johnson [24]. Se accede a la página de este código, y se copia en el archivo “init.el”, para configurar nuestro *Emacs* con la interfaz *ESS* para *R*.

Otra opción para instalar *ESS* en *Emacs*, puede ser utilizando plataformas o marcos de trabajo (frameworks)

especiales que manejan paquetes para Emacs, una de ellas es DOOM Emacs de H. Lissner [25], y otra opción popular es SpaceMacs [26].

V. DATOS PROPUESTOS

Supongamos que tenemos dos grupos A y B. Donde el grupo A es nuestro grupo de control y le enseñamos un tema de Física utilizando la enseñanza tradicional, donde el profesor se dedica a darles el material y los estudiantes toman nota.

El grupo B, será el grupo donde implementamos alguna metodología didáctica activa.

Ahora supongamos que después de la clase de Física, el profesor realiza una prueba o test, cuyas calificaciones, recoge en una sencilla tabla. Esto se realiza para ambos grupos. Los resultados para ambos grupos se anotan en una tabla.

TABLA 1. Dos listas de calificaciones obtenidas para dos grupos no relacionados.

No.	Grupo A	Grupo B
1	6	7
2	5	6
3	6	7
4	7	9
5	5	5
6	7	8
7	6	7
8	4	6
9	3	7
10	6	9
11	6	8
12	5	7
13	7	8
14	8	7
15	6	6
16	5	8
17	8	7
18	4	6
19	6	
20	6	

Este ejemplo es bueno para calcular la *t* de student, ya que los datos pasan las pruebas de normalidad, y los datos son menores a 30 datos, para cada muestra. No discutiremos esto aquí, pero el lector puede consultar varias referencias en internet si está interesado en este punto [27, 28].

La hipótesis nula H_0 , es cuando los datos del grupo B, NO difieren mucho de los datos del grupo A.

La hipótesis alternativa H_1 , corresponde al criterio de que los datos del grupo B, SI difieren bastante de los datos del grupo A.

Lo que se quiere es rechazar la hipótesis nula, así los alumnos del grupo B tienen calificaciones diferentes (y presumiblemente mejores), que las calificaciones del grupo de prueba, o grupo A. Y esto es generalmente lo que se quiere lograr en educación, para mejorar la calidad educativa.

VI. PROGRAMA EN R PARA LA *t* DE STUDENT

Ya teniendo el editor para R, podemos escribir un programa para realizar una gráfica *t* de student de tipo “Welch” en R, utilizando la prueba de la *t* de student de R. Para otro tipo de prueba en Excel, el lector puede consultar una referencia también en línea [29].

Para registrar los datos de la tabla 1, en R, usamos dos vectores de R, con la función c()

```
a <- c(6,5,6,7,5,7,6,4,3,6,6,5,7,8,6,5,8,4,6,6)
b <- c(7,6,7,9,5,8,7,6,7,9,8,7,8,7,6,8,7,6)
```

Así, para este ejemplo ficticio: la variable “a”, guarda las calificaciones del grupo de control A, y la variable “b”, guarda las calificaciones obtenidas por los alumnos del grupo experimental B

Después, se usa la función de R llamada t.test(), para calcular la *t* de student tipo Welch de dos muestras con la siguiente orden

```
tt.al.ba <- t.test(x=b, y=a, paired=FALSE,
  alternative="two.sided", conf.level=0.95)
tt.al.ba
```

El resultado se guarda en la variable tt.al.ba y luego se despliega el resultado:

Welch Two Sample t-test

```
data: b and a
t = 3.4227, df = 35.855, p-value = 0.001566
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.5341108 2.0881114
sample estimates:
mean of x mean of y
 7.111111  5.800000
```

Según este resultado y como tenemos un *valor-p* menor al valor de referencia o umbral de α de 0.05 (correspondiente a un nivel de confiabilidad estándar de 0.95 “conf.level”), se debe de rechazar la hipótesis nula H_0 . Además el promedio del grupo B es mayor que el promedio alcanzado por el grupo A, ya que

```
7.111 > 5.8
Grupo B, Grupo A
x, y
b, a
```

Por lo tanto, aquí el objetivo de la investigación educativa, se ha cumplido, según el criterio estadístico de la t de *student* tipo Welch.

El objetivo aquí es mostrar como se toma este criterio haciendo una gráfica de la distribución t de *student*, para este caso y tomar la misma decisión pero en forma gráfica.

En la siguiente sección mostramos como calcular el llamado valor crítico de una prueba t de *student*, y mostramos como localizarlo gráficamente, usando herramientas nativas en R .

A. Valor crítico

Para calcular el valor crítico de nuestro ejemplo, usamos la función cuartil de t [30], correspondiente a un nivel de confianza $\alpha = 0.05$, y $df=35.855$ grados de libertad según el criterio de Welch. Como la prueba es “a dos colas”, tomamos la mitad del nivel de confianza $\alpha/2 = 0.025$, y calculamos el valor crítico de t , con la función “qt()” de R .

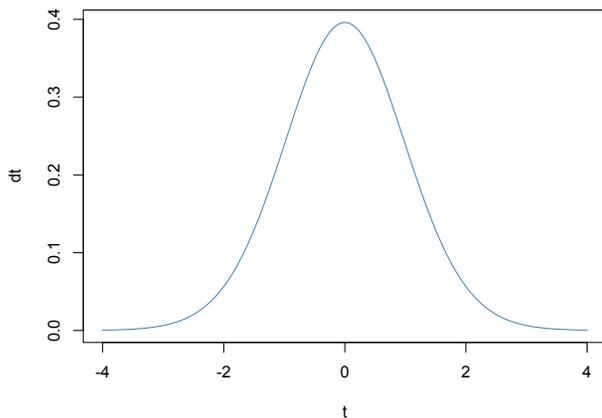


FIGURA 1. Se muestra la distribución dt o t de *student*, con la función “dt()” de R , para las dos muestras de la tabla 1.

```
qt(p=0.025, df=35.855)
-2.028379
```

Así el valor crítico es $t(\alpha/2)=-2.03$, para este ejemplo. Y el valor de t calculada es 3.42, luego se cumple la desigualdad

$$t > |t(\alpha/2)| = t_{\text{critica}}$$

Es decir t esta en la zona que podemos llamar de “rechazo de la hipótesis nula”.

Para ilustrar esto de forma gráfica, se necesita hacer uso de funciones gráficas de R [31-33], el código sería en este caso, usando la función “dt” de R :

```
z <- function(x) dt(x, df=35.855)
```

La variable “z” guardará esta función para nuestro caso que tiene $df=35.855$ grados de libertad.

Y graficamos esta función de -4.0 a 4.0. La figura 1, muestra la distribución t de *student* (dt) centrada en el origen. Para eso usamos la función “curve()” de R .

```
curve(z, from=-4, to=4, xlab="t", ylab="dt",
      col="steelblue")
```

Donde “xlab” y “ylab” son las etiquetas de los ejes x y y , y “col” es el color de la curva.

Lo que le hace falta a la figura 1, es indicar el valor de t para este caso, y el valor crítico de t_{critica} correspondiente al nivel de confianza de α de $0.05=5\%$ (que es el área bajo la curva de la distribución dt para los valores de t desde $t=-\infty$, hasta la t crítica = 2.028).

Podemos hacer uso de utilerías gráficas de bajo nivel de R , como podemos ver en un ejemplo en línea [34]. La técnica consiste en construir un polígono, usando varios puntos de la distribución dt , y después con una orden, construir un polígono coloreado, de cierto color, que pueda ser visible en la gráfica. También señalamos en la gráfica, el valor de t calculado con una línea vertical. Si la línea vertical cruza, una de las zonas sombreadas, con la técnica gráfica, entonces se encuentra en la zona, donde podemos rechazar la hipótesis nula o H_0 , y aceptar como mejor opción la hipótesis alternativa o H_1 .

Si la línea cruza una zona, donde no haya sombra, entonces estará en la región donde, la desviación estándar de el segundo conjunto de valores, no es muy diferente de la desviación estándar del primer conjunto de valores. Es decir, estaremos en la zona, donde no tenemos demasiada evidencia, para rechazar la hipótesis nula H_0 , y por lo tanto deberíamos de aceptarla.

B. Decoraciones gráficas

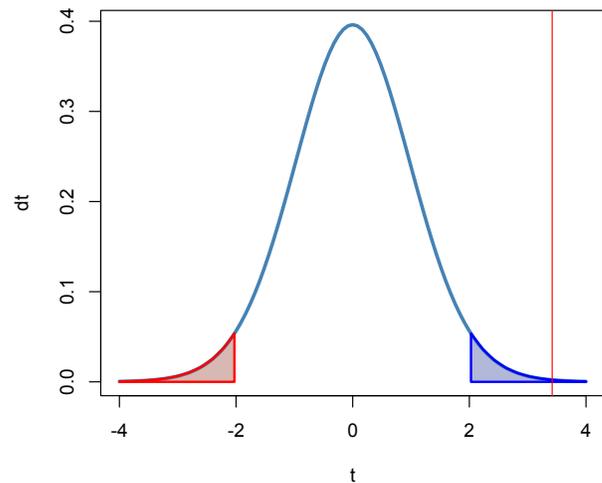


FIGURA 2. Se muestra la distribución dt o t de *student*, con la función “dt()” de R . Se añade un sombreado para ubicar las zonas críticas, cuya área de probabilidad es de 0.05, que es la tolerancia de esta prueba. Como es prueba de dos colas, el área de cada cola es de la mitad de nivel de confianza, esto es cada región sombreada tiene un área de 0.025. Se muestra el valor calculado de t con una línea vertical roja, en este caso, corta la región sombreada de la derecha.

Si adoptamos una idea modificada de la referencia [34], y señalamos la t de *student* calculada con el procedimiento de

Welch, ya escrita en *R*, obtendremos una bonita representación gráfica de nuestro problema. La podemos ver en la figura 2, donde hemos engrosado algunas líneas, para que se vean mejor.

En la figura 2, vemos que la línea roja que indica el valor de *t*, para este caso, corta la región sombreada de la derecha. Según el criterio estándar, tenemos suficientes pruebas estadísticas, para preferir rechazar la hipótesis nula. Claro que esta decisión tiene aproximadamente un margen de error tan pequeño, como el valor de confianza, que hemos tomado de unos 0.05, o en otras palabras del 5%.

El código en *R*, para sombrear las regiones de la izquierda y la derecha en la figura 2, y para colocar con una línea vertical, el valor de *t* calculado, sería el siguiente.

$$t=3.42$$

Para construir los polígonos *p1* y *p2* con una secuencia en *R*, tenemos el siguiente código

```
xx1 <- seq(2.028,4,0.2)
zz1 <- z(xx1)
p1 <- cbind(c(2.028,xx1,4), c(0,zz1,0))
```

```
xx2 <- c(seq(-4,-2.028,0.2),-2.028)
zz2 <- z(xx2)
p2 <- cbind(c(-4,xx2,-2.028), c(0,zz2,0))
```

donde *p1* es la región sombreada de rojo en la gráfica de la figura 2, y *p2* es la región poligonal sombreada de azul (a la derecha) en la misma gráfica.

Después de sombrear ambas regiones con la función “*polygon()*” y el código siguiente

```
polygon(p1, col = rgb(.25,0.31,0.61, alpha = 0.4), border =
"blue", lwd = 2)
polygon(p2, col = rgb(.61,0.31,0.25, alpha = 0.4), border =
"red", lwd = 2)
```

La línea roja vertical que indica el valor obtenido de *t*, se dibuja con la función “*abline()*” y la siguiente orden

```
abline(v=3.42, lwd=1, col="red")
```

Con esto obtenemos, la gráfica de la figura 2, la cual es bastante más ilustrativa, y es una mejora de la gráfica en la figura 1.

Nótese, que todos los valores numéricos anotados en el código, son ajustados de acuerdo a los valores obtenidos en este ejemplo en particular. Si queremos repetir lo mismo para otras muestras, hay que estar ajustando valores numéricos, en cada caso, que queramos estudiar.

Sin embargo, este ejemplo, es bastante útil para aplicar el estadístico de la *t* de student, a otros datos.

VII. ANÁLISIS DE RESULTADOS

En este escrito, mostramos un ejemplo hipotético de como aplicar el criterio de validación de hipótesis, con un conjunto

de datos ficticios, mostrados aquí, a manera de ejemplo. La *t* de student, para datos no apareados, nos puede servir para aplicar un método numérico, y tener un criterio estadístico para validar una metodología didáctica que apliquemos en el salón de clase.

En este ejemplo en particular, hemos decidido rechazar la hipótesis nula, con cierto margen de error, establecido por el nivel de confiabilidad de la prueba estadística al 5%. El “valor-*p*” es de 0.0016, o en otras palabras, en el caso de que la hipótesis nula fuera verdadera, es decir, cuando nuestra metodología didáctica no representa, una mejora en el aprendizaje del grupo experimental B, tendríamos una probabilidad de error de rechazo o de falso positivo del 0.16% por ciento, que es lo que representa el “valor-*p*”, en este caso. Como esta probabilidad es comparativamente muy baja, podemos decir que en este caso ficticio, nuestro grupo experimental B, si tuvo una diferencia en su aprendizaje, con respecto a los alumnos del grupo de control o grupo A. Y esta diferencia es positiva, favoreciendo así al grupo B, y prefiriendo la metodología didáctica aplicada, a la enseñanza tradicional. Claro que este tipo de resultados, es lo que generalmente se desea lograr en el área educativa.

Para una mejor interpretación del *valor-p*, de una prueba estadística, el lector puede consultar la referencia de Arango [35].

Así en este ejemplo ficticio, logramos nuestro objetivo inicial de mejorar el nivel educativo de nuestros alumnos hipotéticos.

VIII. ARCHIVO DE PROGRAMA

Resumiendo, podemos decir, que para este ejemplo necesitamos escribir un poco de código en *R*, en un archivo de texto, y podemos luego interactuar con este archivo enviando los comando uno a uno a *R* a través de la interface ESS de Emacs. Supongamos que tenemos este archivo nombrado por nosotros “*prueba-t-de-student.R*”, como observamos el nombre del archivo debe de terminar en “.*R*”, para que a la hora de abrirlo con Emacs, éste programa lo reconozca como archivo de entrada o como programa para *R*.

A continuación resumimos y anotamos los comandos más importantes que deberá contener este archivo preparado para el lenguaje *R*.

```
a <- c(6,5,6,7,5,7,6,4,3,6,6,5,7,8,6,5,8,4,6,6)
b <- c(7,6,7,9,5,8,7,6,7,9,8,7,8,7,6,8,7,6)
```

```
tt.al.ba <- t.test(x=b, y=a, paired=FALSE,
alternative="two.sided", conf.level=0.95)
tt.al.ba
```

```
qt(p=0.025, df=35.855)
```

```
z <- function(x) dt(x, df=35.855)
```

```
curve(z, from=-4, to=4, xlab="t", ylab="dt",
col="steelblue", lwd=3)
```

```
xx1 <- seq(2.028,4,0.2)
```

```
zz1 <- z(xx1)
p1 <- cbind(c(2.028,xx1,4), c(0,zz1,0))

xx2 <- c(seq(-4,-2.028,0.2),-2.028)
zz2 <- z(xx2)
p2 <- cbind(c(-4,xx2,-2.028), c(0,zz2,0))

polygon(p1, col = rgb(.25,0.31,0.61, alpha = 0.4),
border = "blue", lwd = 2)
polygon(p2, col = rgb(.61,0.31,0.25, alpha = 0.4),
border = "red", lwd = 2)

abline(v=3.42, lwd=1, col="red")
```

Al abrirlo con Emacs, se mostrará en el menú un icono para abrir otra ventana, donde se iniciará el programa *R*. Después en el mismo menú se puede oprimir un icono con forma de flecha que apunta hacia la derecha, para ir ejecutando el archivo línea a línea. Cuando llegue a la instrucción donde esta la función “curve()”, *R* abrirá una ventana donde empezará a desplejar los resultados de la gráfica mostrada en la figura 2, y poco a poco la irá completando, hasta finalizar con el ejemplo. Al final el gráfico se puede guardar utilizando el menú de la interfaz gráfica *XQuartz* [36], para la Mac. O una interfaz equivalente, dependiendo del sistema operativo que se esté usando. Para explorar más detalles del lenguaje *R*, existen varios libros en la literatura, que el lector puede consultar, como referencias [37-41].

Como nota final, este archivo esta preparado suponiendo que conocemos de antemano ciertos valores numéricos como los grados de libertad que arroja la función “t.test()”, y el valor crítico de *t* para el nivel de confianza $\alpha = 0.05$, por lo que se necesitará ajustar estos valores y otros parámetros dependientes, de acuerdo a los datos de las muestras iniciales. Sin embargo, el empleo de un editor que puede interactuar con *R*, como Emacs con su interfaz ESS, hace que este trabajo sea más sencillo en cada caso.

IX. CONCLUSIONES

Hemos mostrado como, utilizar el estadístico *t* de *student*, con el objeto de apoyar estadísticamente, la búsqueda de una metodología didáctica efectiva, en la que se obtengan mejores resultados de aprendizaje, que con la enseñanza tradicional. El ejemplo se aplica para dos muestras no relacionadas, esto es, para dos grupos de estudiantes diferentes. Desde luego, aunque el ejemplo es ficticio, nos muestra como podemos utilizar esta herramienta, para fomentar la investigación en el área de la educación.

AGRADECIMIENTOS

El autor quiere agradecer el apoyo recibido por el Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) de México, por el apoyo recibido en la realización de este trabajo. Asimismo, quiere expresar su agradecimiento al proyecto de la Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional, con número de

registro 20231591 y de nombre *Aplicación de la Filosofía STEM en la enseñanza de la Física*, por otorgar facilidades y apoyo en la realización del presente trabajo.

REFERENCIAS

- [1] RPubS, *Estadística con R, Blog sobre el uso de R en la estadística y creación de gráficos, Historia de R*, <<https://rstadistica.blogspot.com/2015/10/historiaR.html>>, Consultado el 13 de diciembre de 2022. (2015).
- [2] Home, *The R Project for Statistical Computing*, <<https://www.r-project.org>>, Consultado el 20 de diciembre de 2022.
- [3] Manual de R, s.f., *Introduction to R. I. Using R for Statistical Tables and Plotting Distributions*, I: R as Statical Calculator, 1-6 <<https://www.math.arizona.edu/~jwatkins/R-01.pdf>>, Consultado el 10 de diciembre de 2022.
- [4] Homebrew, *The Missing Package Manager for macOS (or Linux)*, <<https://brew.sh>>, Consultado el 8 de diciembre de 2022.
- [5] Dani, *dani blog, macOS setup, Install R and RStudio with Homebrew in MacOS*, <<https://dani.gg/en/install-r-and-rstudio-with-homebrew-on-macos>>, Consultado el 8 de diciembre de 2022.
- [6] Posit, *Download RStudio Desktop*, <<https://posit.co/download/rstudio-desktop/>>, Consultado el 9 de diciembre de 2022.
- [7] Notepad++, *Downloads*, <<https://notepad-plus-plus.org/downloads/>>, Consultado el 8 de diciembre de 2022.
- [8] GNU Emacs, *An extensible, customizable, free/libre text editor – and more*, <<https://www.gnu.org/software/emacs/>>, Consultado el 8 de diciembre de 2022.
- [9] Neovim, *hyperextendible Vim-based text editor*, <<https://neovim.io/>>, Consultado el 8 de diciembre de 2022.
- [10] LunarVim, *An IDE layer for Neovim with sane defaults*, <<https://www.lunarvim.org/es/>>, Consultado el 8 de diciembre de 2022.
- [11] Emacs Wiki, *Historia de Emacs*, <<https://www.emacswiki.org/emacs/HistoriaDeEmacs>>, Consultado el 8 de diciembre de 2022.
- [12] Massachusetts Institute of Technology, *MIT*, <<https://www.mit.edu>>, Consultado el 8 de diciembre de 2022.
- [13] Biografía de Richard Matthew Stallman, *Creador del Software libre*, <<https://docs.google.com/document/d/1hGjtEZE9gRd48d3y7jkwcbOIdxcKegsCwTvUxOIDWEI/edit>>, Consultado el 8 de diciembre de 2022.
- [14] Richard Stallman, *Richard Stallman’s Personal Site*, <<https://stallman.org>>, Consultado el 8 de diciembre de 2022.
- [15] Scohen, Railwaycat, *GitHub, Emacs mac port*, <<https://github.com/railwaycat/homebrew-emacsmacport>>, Consultado el 2 de diciembre de 2022.
- [16] macOS Setup Guide, *Emacs installation, Emacs Mac port*, <<https://sourabhbjaj.com/mac-setup/Emacs/>>, Consultado el 2 de diciembre de 2022.
- [17] Bastian, W., Karlitskaya, A., Poettering, L., Lothberg, J., *XDG Base Directory Specification*,

- <<https://specifications.freedesktop.org/basedir-spec/basedir-spec-latest.html>>, Consultado el 8 de diciembre de 2022.
- [18] Strubing, M., Farbenmeer GmbH, *The Power of the XDG Base Directory Specification*, <<https://farbenmeer.de/blog/the-power-of-the-xdg-base-directory-specification>>, Consultado el 8 de diciembre de 2022.
- [19] StackExchange, superuser, *What are the step to move all your dotfiles into XDG directories?*, <<https://superuser.com/questions/874901/what-are-the-step-to-move-all-your-dotfiles-into-xdg-directories>>, Consultado el 8 de diciembre de 2022.
- [20] Emacs Wiki, Learn emacs lisp, <<https://www.emacswiki.org/emacs/LearnEmacsLisp>>, Consultado el 4 de diciembre de 2022.
- [21] MELPA, (*milkypostman's Emacs Lisp package Archive*), <<https://melpa.org/#/getting-started>>, Consultado el 9 de diciembre de 2022.
- [22] ESS, *Emacs Speaks Statistics*, <<https://ess-r-project.org>>, Consultado el 8 de diciembre de 2022.
- [23] emacx, Emacs – instalar paquetes / packages, <https://www.youtube.com/watch?v=_GZA_GwMybo>, Consultado el 10 de diciembre de 2022.
- [24] Paul Johnson, archivo de configuración para Emacs, <<https://pj.freefaculty.org/Software/favoriteEmacsFiles/init.el>>, Consultado el 4 de diciembre de 2022.
- [25] Lissner, H., *DOOM Emacs, Almacenamiento GitHub*, <<https://github.com/doomemacs/doomemacs>>, Consultado el 3 de diciembre de 2022.
- [26] Spacemacs, *A community-driven Emacs distribution*, <<https://www.spacemacs.org>>, Consultado el 3 de diciembre de 2022.
- [27] Santana C., A. D., *Rpubs by RStudio*, https://rpubs.com/SantanaAlec/AI5UC1_6, Consultado el 10 de diciembre de 2022. (2021).
- [28] Sánchez, TRA, *t-Student. Usos y abusos*, Revista mexicana de cardiología **26** 1, 59-61 (2015), <<https://www.medigraphic.com/cgi-bin/new/resumen.cgi?IDARTICULO=56921>>, Consultado el 11 de diciembre de 2022.
- [29] El tio estadístico, *Prueba t de Student (para muestras independientes)*, <<https://www.youtube.com/watch?v=3uZ7M97JRO8>>, Consultado el 3 de diciembre de 2022.
- [30] Scribb, *How do I find the critical value of t in R*, <<https://www.scribbr.com/frequently-asked-questions/critical-value-of-t-in-r/>>, Consultado el 2 de diciembre de 2022.
- [31] Stack Overflow, *plot t distribution from t test*, <<https://stackoverflow.com/questions/41470162/plot-t-distribution-from-t-test>>, Consultado el 3 de diciembre de 2022.
- [32] Statology, *How to plot a t distribution in R*, <<https://www.statology.org/plot-t-distribution-r/>>, Consultado el 3 de diciembre de 2022.
- [33] ee Geeks for Geeks, *Plot t distribution in R*, <<https://www.geeksforgeeks.org/plot-t-distribution-in-r/>>, Consultado el 3 de diciembre de 2022.
- [34] Stack Overflow, *Colorear área bajo la curva*, <<https://es.stackoverflow.com/questions/495925/colorear-%C3%A1rea-bajo-la-curva>>, Consultado el 3 de diciembre de 2022.
- [35] Arango, J. J., Entendiendo de una vez por todas el valor de p, <<https://www.youtube.com/watch?v=PId16C1gT8k>>, Consultado el 4 de diciembre de 2022.
- [36] XQuartz, página web, <<https://www.xquartz.org/>>, Consultado el 5 de diciembre de 2022.
- [37] Chang, W., *R Graphics Cookbook*, (O'Reilly, Sebastopol, California, USA, 2018).
- [38] Murrell, P., *R Graphics*, (Chapman & Hall/CRC Computer Science & Data Analysis, Third edition, Inglaterra, 2018).
- [39] Kabacoff, R. I., *R in Action*, (Manning, 3rd Ed., Shelter Island, NY, USA, 2022).
- [40] Davies, T. M., *The book of R*, (No Starch Press, San Francisco, California, USA, 2016).
- [41] Matloff, N., *The Art of R Programming*, (No Starch Press, San Francisco, California, USA, 2011).