APORTACIONES DEL SOFTWARE LIBRE «R» AL PROCESO DE INVESTIGACIÓN PSICOLÓGICA

RAFAEL JÓDAR ANCHÍA [1]

Fecha de recepción: noviembre de 2010
Fecha de aceptación y versión definitiva: diciembre de 2010

RESUMEN: La intención fundamental de este artículo es presentar el software libre «R» de análisis estadístico que permite realizar, entre otras muchas cosas, dos análisis relevantes en la investigación psicológica que no son posibles en el software privado habitual (generalmente SPSS): los cálculos de poder y el cálculo de las correlaciones policóricas. Además, presentamos de modo esquemático las funciones más básicas de R en el análisis de datos en la investigación psicológica, con el fin de facilitar al lector la toma de contacto con este software.

PALABRAS CLAVE: Paquete estadístico, Correlación policórica, Poder estadístico, Software libre, Proyecto R.

Introducing «R» free software contributions to the psychological research process

ABSTRACT: The main purpose of this article is to introduce the free software for statistical analysis called R, that allows conducting among other many calculus, two important analyses non available in the common non-free software (e. g. SPSS): power analysis and polychoric correlations. In addition, the basic R commands related to the most common statistical analysis in psychology research are described.

KEY WORDS: Statistical package, Polychoric correlation, Power analysis, Free software, R project.

[1] Profesor en la Facultad de Ciencias Humanas y Sociales, Universidad Pontificia Comillas. E-mail: rafajodar@chs.upcomillas.es

1. INTRODUCCIÓN

El software R tiene un impactante presente y muy prometedor futuro otorgando recursos al investigador en el campo psicológico. R es un programa distribuido bajo licencia pública general GNU que protege la libre distribución, la modificación y el uso del software, de forma que no pueda ser apropiado por intereses privados. Últimamente se están produciendo diferentes publicaciones que resaltan las ventajas de R frente a SPSS, remarcándose que su versatilidad (puedes generar tus propias funciones), la capacidad y calidad de los gráficos, análisis muy avanzados (al comunidad científica que participa en su desarrollo es numerosa y de gran calidad) y el acceso sin coste económico (Oliden, 2009). Además, una vez descargado el programa, puede seguir siendo actualizado, de forma que las mejoras se añaden automáticamente y se puede acceder a bases de datos, algunas de ellas históricos, lo que supone un interesante recurso para la docencia.

Ciertas técnicas de análisis de datos se acaban implantando en la comunidad científica, no sólo por su conveniencia estadística o su utilidad, sino también por la facilidad con la que los investigadores puedan acceder al software que permite realizar dichas técnicas. El cálculo de la potencia estadística es uno de estos casos. Aunque la *Task Force on Statistical Inference* de la APA indica la conveniencia de calcular la potencia previamente a la recogida y análisis de datos, para estimar el número de participantes necesarios (Wilkinson, 1999), el cálculo de la potencia es habitualmente poco accesible y por tanto el cálculo del poder no es una práctica extendida en la investigación (Cohen, 1992). Otra técnica de análisis que presenta ventajas claras pero que su difusión está amenazada por la dificultad de acceso a la misma es la correlación policórica, muy útil para calcular la fiabilidad de escalas con pocas opciones de respuesta (Oliden y Zumbo, 2008) y como forma de realizar análisis factoriales confirmatorios con datos ordinales (Flora y Curran, 2004). Ambos análisis pueden ser fácilmente realizados mediante el software libre R.

2. UTILIZACIÓN DE R

2.1. Descargar R

Descargamos R de la página web http://cran.r-project.org, eligiendo el subdirectorio «base». Responderemos «Sí» a la pregunta «¿Opciones de configuración?», con el fin de poder indicar que queremos ventanas

separadas, indispensables para poder usar un interfaz gráfico. Una forma de instalar directamente R y *Rcomander* es a través del archivo de instalación de la Universidad de Cádiz (http://knuth.uca.es/R/doku.php). Al instalar R de esta forma, se abrirá automáticamente *Rcmdr* al comenzar el programa.

Al descargar y ejecutar R nos encontramos con una línea de comandos (señalada mediante el símbolo >) y un menú desplegable superior. En dicho menú, en la opción «Paquetes» podemos «instalar» y posteriormente «cargar» paquetes que poseen diferentes funciones estadísticas. Actualmente se han desarrollado más de 800 paquetes. Al instalar y cargar el paquete «*Rcmdr*» encontramos que se despliega una interfaz gráfica muy similar a SPSS, que permite realizar las operaciones necesarias para un curso introductorio de análisis de datos para la investigación social. Aunque la facilidad de utilizar *Rcmdr* es evidente, creemos que utilizar la línea de comando ofrece ventajas claras. En primer lugar, *Rcmdr* no da acceso más que algunas de las funciones de R (aunque se irán incluyendo con el paso del tiempo). Además, una vez superado la aridez inicial, la sintaxis de R es fácil de asimilar y permite que una parte muy importante de las ventajas asociadas a R sean accesibles.

2.2. Introducir datos en R

Una forma muy sencilla de comenzar es utilizar Excel para introducir los datos (escribiendo en la primera fila los nombres de las variables) y grabar en Excel como formato .csv (separados por comas).

Edad	*Peso*	*Sexo*	*Fumador*	*i1*	*i2*	*i3*
20	80	1	Sí	5	4	5
21	60	2	No	5	5	4
22	75	1	No	3	4	5
23	64	2	No	3	2	3
22	68	1	Sí	2	3	3
24	72	2	No	2	2	1
25	95	1	Sí	1	1	1
26	73	2	No	2	1	2

Luego desde R, tecleamos el comando

```
> data1 <- read.csv("c://data1.csv", header=T, sep=";")
```

Así hemos introducido en la variable data1 la matriz de datos que originalmente teníamos en Excel (es muy importante notar que R distingue entre mayúsculas y minúsculas, de forma que debemos teclear con cuidado). Es también llamativo que en vez de el signo «=», R utiliza preferentemente «<-», dado que el signo igual queda reservado para operaciones lógicas. Al teclear «> *data1*» visualizamos los datos que inicialmente teníamos en Excel en la consola de R.

Hasta este momento R sólo reconoce un objeto, data1, de manera que cuando le pedimos la media de una variable que sabemos que está contenida en data1, por ejemplo peso, nos devuelve un error. Para poder trabajar con las variables por separado es necesario el comando «*attach*» que permite adjuntar las columnas de data1 como variables vectores:

```
> attach(data1)
```

2.3. Definir tipos de variables

Mientras que en SPSS podemos indicar qué variables son propiamente de escala (métrica), cuáles ordinales y cuáles nominales, en R es crucial que indiquemos qué variables no son numéricas sino que constituyen factores (variables categóricas).

En nuestro ejemplo, tenemos una variable que figura como numérica cuando en realidad es cualitativa (sexo) que, por tanto, podemos definir como un factor y otorgar etiquetas. Para ello:

```
> data1 <- transform(data1, sexo=factor(sexo,labels=c("V","M")))
```

Al explorar nuestros datos ahora (>*data1*) veremos que la variable sexo está codificada convenientemente.

2.4. Cálculo de poder

Abordamos a continuación el cálculo de la potencia o poder, disponible en el paquete R. ¿Necesitamos más datos para aumentar nuestra capacidad de detectar las diferencias o asociaciones que hayamos hipotetizado? El poder o potencia es la probabilidad de hallar efectos significativos cuando éstos existen, mientras que su complementario consiste en la probabilidad de no detectar efectos verdaderos (error tipo II o β). La potencia o el poder

de un análisis depende del tamaño muestral, del tamaño del efecto que esperamos encontrar (que podemos apreciar al revisar la literatura) y del nivel de significación fijado (α, que suele ser 0,05).

En las funciones de cálculo de poder que se describen a continuación se puede dejar alguno de los elementos sin especificar (tamaño muestral o n, tamaño del efecto, nivel de significación y poder), de forma que R calculará el valor omitido a partir de los que han sido especificados.

Para correlaciones:

> pwr.r.test(n = , r = , sig.level = , power =)

Siendo n el tamaño de la muestra, r el valor de la correlación. Cohen (1992) sugiere que correlaciones de 0,1, 0,3 y 0,5 corresponden a valores pequeños, medios y grandes respectivamente. Por ejemplo, podemos pedir que tamaño muestral (n) necesitamos para poder captar una correlación media (0,3) con un alfa de 0,05, si queremos tener una potencia mínima de 0,7 tecleamos:

> pwr.r.test(,r=0.3,sig.level=0.05,power=0.7)

Devolviendo R los siguientes resultados:

n = 67.3403, r = 0.3, sig.level = 0.05, power = 0.7,
alternative = two.sided

Los resultados muestran que con 68 participantes alcanzaríamos una potencia de 0,7, si la correlación hipotetizada tiene una magnitud en valor absoluto de 0,3. Es muy importante que estos cálculos se realicen previamente a la recogida de información, y que el tamaño del efecto que esperamos en nuestro estudio lo extraigamos de la revisión de la literatura y nunca de los propios datos. Para evitar sospechas, es muy recomendable especificar las fuentes en las que nos basamos al anticipar el tamaño del efecto (Wilkinson, 1999).

La forma de realizar cálculos de potencia utilizando contrastes t, de tamaño idéntico en los grupos es:

> pwr.t.test(n = , d = , sig.level = , power = , type = c("two.sample",
"one.sample", "paired"))

Donde n es el tamaño muestral, del tamaño del efecto, y *type* indica a R si vamos a ejecutar un test de dos muestras independientes («two.sample»), un test de una muestra frente a un valor poblacional («one.sample») o una t para muestras relacionadas («paired»).

Si tenemos tamaños muestrales diferentes, usamos:

> pwr.t2n.test(n1 = , n2= , d = , sig.level =, power =)

Tamaños del efecto (d) de 0,2, 0,5 y 0,8 corresponden a valores pequeños, medios y grandes respectivamente (Cohen, 1992).

Los cálculos de poder para otros análisis se indican de la siguiente forma:

> *pwr.chisq.test(w =, N = , df = , sig.level =, power =)*, para chi cuadrado.

> *pwr.anova.test(k = , n = , f = , sig.level = , power =)*, para ANOVA de tamaños iguales, muestras independientes.

> *pwr.f2.test(u =, v = , f2 = , sig.level = , power =)*, para modelos lineales.

> *pwr.2p.test(h = , n = , sig.level =, power =)*, pruebas sobre proporciones.

2.5. Estudios descriptivos

Antes de proceder al contraste de hipótesis propiamente dicho es recomendable realizar una exploración y descripción de nuestros datos. Tres funciones son básicas para explorar las variables:

> *summary(data1)*

> *describe(data1)* (previamente instalar y cargar *Hmisc* para poder utilizar la función describe).

> *freq(data1)* (instalar y cargar *prettyR*).

Estas funciones devuelven cuartiles, media, mediana, varianza, frecuencias, etc., de la variable introducida.

2.5.1. *Exploración de las variables cuantitativas*

La representación gráfica permite la rápida identificación de errores en la introducción de nuestros datos, así como la observación de patrones. Para representar gráficamente variables cuantitativas disponemos de:

> *boxplot(edad)* Muy útil para ver outliers.

> *hist(sueldo)* Permite realizar histogramas.

Para realizar estudios de normalidad, el test de Shpiro Wilk y el test de Kolmogorov-Smirnov con la corrección de Lilliefors se ejecutan simplemente con los comandos:

> *shapiro.test(peso)*

> *lillie.test(peso)* (instalar y cargar *nortest*).

Además, R dispone de otros test recientes para estimar la normalidad de los datos, como el test de Anderson-Darling o el Cramer-von Mises.

2.5.2. *Explorar gráficamente variables cualitativas*

> *pie(table(sexo))* (Gráfico de sectores).

> *pie(table(fumador))*

Para variables semi-cuantitativas u ordinales:

> *barplot(table(i1)).* Gráfico de barras.

Para explorar gráficamente la covariación de variables cuantitativas, podemos realizar diagramas de dispersión mediante la instrucción:

> *plot(edad,peso)*

2.6. Estudio de relación entre variables

2.6.1. *Asociación entre variables cuantitativas*

Para estudiar la asociación entre variables cuantitativas, podemos realizar una correlación de Pearson:

> *cor(edad,peso).* Este comando devuelve el valor de la r de Pearson entre edad y peso.

> *cor.test(edad,peso,method="pearson").* Este comando añade además una prueba de significación.

2.6.2. *Asociación entre variables cualitativas*

La asociación entre variables cualitativas la podemos realizar mediante la observación de tablas de contingencia:

> *table(sexo,fumador)*

Si queremos añadir proporciones a la tabla de contingencia:

> *prop.table(table(sexo,fumador))*

Añadiendo además el test chi cuadrado:

> *chisq.test(table(sexo,fumador))*

El test realiza la corrección de Yates automáticamente si la tabla es 2x2. Además R tiene funciones que permiten estimar la asociación en caso de que la frecuencia esperada sea inferior a 5 en alguna casilla.

2.6.3. *Asociación entre variables ordinales*

Las correlaciones policóricas permiten estudiar la asociación entre variables ordinales. Para hallar una correlación policórica, empleamos la función polychor (del paquete polycor).

El código «> *polychor(i1,i2)*» nos devuelve la correlación policórica entre las puntuaciones de los ítems 1 y 2.

Teniendo en «*data*» una matriz de datos que contenga sólo las variables ordinales de las que se desea extraer la matriz de correlaciones policóricas, se ha de teclear en primer lugar el código que se adjunta en el Anexo, que añade la función *mpolycor* a R. Esta función ha sido desarrollada por el autor e ilustra cómo mediante escasas líneas de código el usuario puede adaptar el software R a sus propias necesidades. Es precisamente esta maleabilidad una de las características más importantes del software libre.

Después de introducir el código, al teclear «> *mpolycor(data*)» automáticamente se genera la matriz de correlaciones policóricas. Esta matriz es la información básica para poder realizar cálculos posteriores como el coeficiente de fiabilidad (Oliden y Zumbo, 2008) o el análisis factorial confirmatorio (Flora y Curran, 2004).

2.7. 2.7 Diferencia de medias

2.7.1. *Contraste de la media respecto a un valor*

Para realizar una comparación de la media con respecto a un valor determinado (generalmente una media poblacional), se teclea*:*

> *t.test(peso,mu=60)*

2.7.2. *Contraste de medias de grupos independientes*

Si la intención es realizar un contrate de medias de dos grupos independientes (por ejemplo, el estudio de las diferencias en peso entre hombres y mujeres) y asumimos varianzas iguales, tecleamos:

> *t.test(peso~sexo, var.equal=T*

(el símbolo «~» se consigue pulsando «Alt Gr» y «4» al mismo tiempo)

Podemos conocer si las varianzas de la variable dependiente (peso en este caso) son iguales entre hombres y mujeres, mediante el siguiente código:

> *leveneTest(peso, sexo)* (disponible en el paquete *car*).

Existen en R otras muchas funciones para estimar la diferencia de varianza y una de las ventajas de este software es que las nuevas técnicas que se propongan en la comunidad científica serán rápidamente incorporadas quedando disponibles para el usuario.

Si buscamos además explorar los datos de dos grupos gráficamente, teclearíamos:

> *boxplot(peso~sexo)*

2.7.3. *Contraste de medias de dos grupos relacionados (medidas repetidas)*

Para estudiar diferencias entre dos medidas cuantitativas (por ejemplo la medida antes y después de un tratamiento, o la valoración de un grupo de personas hacia dos productos diferentes):

> *t.test(peso,pesod, paired=TRUE)*

De esta forma compararíamos el peso antes y el peso después.

2.7.4. *Diferencia de medias entre más de dos grupos independientes*

La instrucción para realizar el ANOVA de muestras independientes es la siguiente:

> *summary(aov(peso~fumador,data1)*

«*peso*» es en este caso la variable dependiente, cuantitativa, y fumador es la variable cualitativa independiente (factor), con tres niveles (fumador, no fumador, fumador muy ocasional).

El código «> *tapply(peso,sexo,mean)*» permite ver las medias para cada grupo, mientras que «> *TukeyHSD(aov(peso~fumador,data1),"peso")*» realiza las comparaciones por pares según Tukey.

3. CONCLUSIÓN

Hemos repasado las funciones básicas de R que permiten analizar los datos en una investigación psicológica básica, subrayando dos funciones (los cálculos de poder y las correlaciones policóricas) que no suelen estar accesibles en otros paquetes estadísticos. Usando una interfaz gráfica (cargando el paquete *Rcmdr*) se accede a otras funciones con el mismo sistema y lógica que la ventana de datos de SPSS. Sin embargo hemos creído conveniente sistematizar el código de R porque la comprensión de la sintaxis de este lenguaje permite al usuario beneficiarse de todas las posibilidades del programa. El libre acceso a este software, la colaboración en la comunidad científica en su mejora y su distribución, y el amplísimo repertorio de análisis que el R posee le convierten en una valiosa herramienta para la docencia e investigación psicológica.

4. BIBLIOGRAFÍA

COHEN, J. (1992), «A power primer», *Psychological Bulletin,* 112 (1): 155-159. doi:10.1037/0033-2909.112.1.155

FLORA, D. B., y CURRAN, P. J. (2004), «An empirical evaluation of alternative methods of estimation for confirmatory factor analysis with ordinal data», *Psychological Methods*, 9: 466-491.

OLIDEN, P. E. (2009), «¿Existe vida más allá de SPSS? descubre R», *Psicothema*, 21 (4): 652-655.

OLIDEN, P. E., y ZUMBO, B. D. (2008), «Coeficientes de fiabilidad para escalas de respuesta categórica ordenada», *Psicothema*, 20 (4): 896-901.

WILKINSON, L. (1999), «Statistical methods in psychology journals: Guidelines and explanations», *American Psychologist*, 54 (8): 594-604. doi:10.1037/0003-*066X.54.8.594*

ANEXO

Se presenta a continuación el código realizado por el autor que crea una nueva función en R para generar la matriz de correlaciones policóricas de un conjunto de datos. Esta matriz es fundamental para poder trabajar desde ella en los análisis factoriales confirmatorios o en la estimación de la fiabilidad para datos ordinales.

```
mpolycor <- function(puntuaciones) {

dd <- numeric()

length (dd) <- length (puntuaciones) * length (puntuaciones)

attr(dd, «dim») <- c(length(puntuaciones), length(puntuaciones))

for(x in c(1:(length(puntuaciones)-1))) for(y in c((x+1):length
(puntuaciones))) dd[x,y]<-(polychor(puntuaciones[,x],puntuacione
s[,y]))

for (x in c(1:length(puntuaciones))) dd[x,x]<- 1

for(x in c(1:(length(puntuaciones)-1))) for(y in
c((x+1):length(puntuaciones))) dd[y,x]<-dd[x,y]

print(dd)

}
```