Correlación: teoría y práctica

Pablo Vinuesa, CCG-UNAM. http://www.ccg.unam.mx/~vinuesa/

v2, 1 de Agosto, 2018

Contents

1 Correlación: teoría

Este tema es parte del **Taller 2 - Análisis exploratorio y estadístico de datos biológicos usando R**, de la Universidad Nacional Autónoma de México, impartido entre 30 de Julio y 3 de Agosto de 2018 en el Centro de Ciencias Genómicas. Para más información consultar la página del taller en: http://congresos.nnb.unam.mx/TIB2018/t2-analisis-exploratorio-y-estadistico-de-datos-biologicos-usando-r/. .

El matrial se distribuye desde el repositiorio GitHub curso_Rstas

La parte teórica está basada en (Crawley 2015), (Everitt and Hothorn 2014) y (A. P. Field, Miles, and Field 2012).

Este documento está aún en construcción y es generado con R (R Core Team 2018), rstudio (RStudio Team 2016), knitr (Xie 2018), rmarkdown (Allaire et al. 2018), pandoc (MacFerlane 2016) y $LaTeX$.

1.1 Introducción: el concepto de correlación

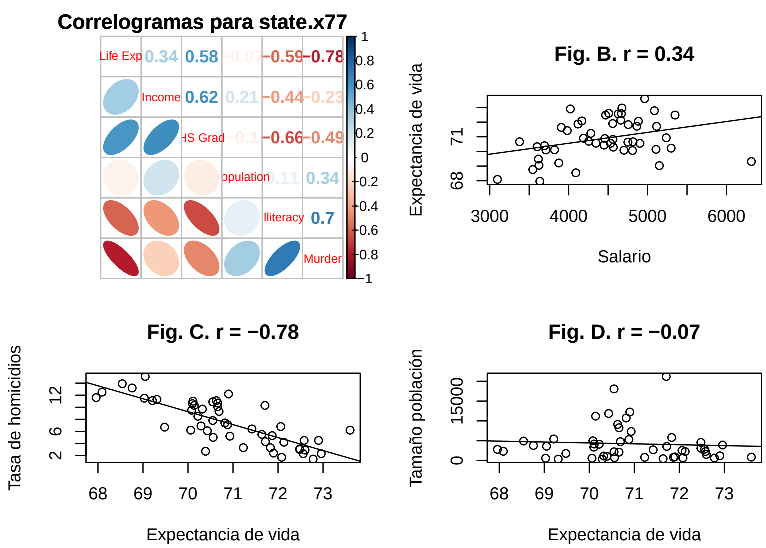
La correlación es una **medida de la relación (covariación) lineal entre dos variables cuantitativas contínuas (x, y)**. La manera más sencilla de saber si dos variables están correlacionadas es determinar si co-varían (varían conjuntamente). Es importante hacer notar que esta **covariación no implica necesariamente causalidad**, la correlación puede ser fortuita, como en el caso clásico de la correlación entre entre el número de venta de helados e incendios, debido al **efecto de una tercera variable**, la temperatura ambiental.

La correlación es en esencia una **medida normalizada de asociación o covariación lineal entre dos variables**. Esta medida o índice de correlación r puede variar entre -1 y +1, ambos extremos indicando correlaciones perfectas, negativa y positiva respectivamente. Un valor de $r = 0$ indica que no existe relación lineal entre las dos variables. Una correlación positiva indica que ambas variables varían en el mismo sentido. Una correlación negativa significa que ambas variables varían en sentidos opuestos. Lo interesante del índice de correlación es que r **es en sí mismo una medida del tamaño del efecto**, que suele interpretarse de la siguiente manera:

- correlación **despreciable**: $r < |0.1|$
- correlación **baja**: $|0.1| < r <= |0.3|$
- correlación **mediana** : $|0.3| < r <= |0.5|$
- correlación **fuerte o alta**: $r > |0.5|$

La siguiente figura muestra ejemplos de pares de variables con correlación positiva moderada, negativa fuerte, así como correlación despreciable. ¿Sabrías identificar cada caso? Estos son datos que provienen del set state.x77 del paquete *datasets* que viene con la instalación de base de R. Se trata de una matriz de 50 filas y 8 columnas con estadísticas para 50 estados de EU relativos al tamaño de la población, renta per cápita, % de analfabetismo, expectancia de vida, tasa de asesinato, % de graduados de preparatoria, número promedio de días con heladas y área del estado en millas cuadradas. Esta información la pueden ver con el comando $help("state.x77")$.

Analizemos visualmente las relaciones entre un subconjunto de las variables de state.x77 para afianzar los conceptos clave. Nótese que en las Figs. C-D, además de los **gráficos de dispersión ("scatterplots")**, se muestra la **recta de regresión** correspondiente al ajuste de un **modelo lineal** a los datos, con el fin de visualizar mejor la desviación de los puntos con respecto al modelo lineal.



1.2 Definiciones formales

La **correlación** se define en términos de la **varianza** (s^2) de las variables x e y, así como de la **covarianza** *cov* de x,y. Es por tanto una medida de la variación conjunta de ambas variables ($cov(x, y)$).

1.2.1 varianza (s^2)

La varianza de una muestra representa el promedio de la desviación de los datos con respecto a la media

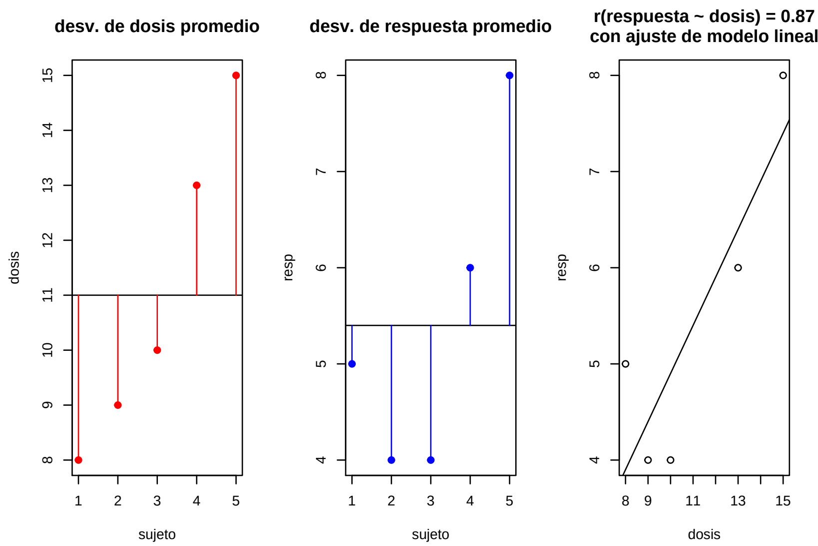
$$Varianza(s^2) = \frac{\sum(x_i - \bar{x})^2}{N-1} = \frac{\sum(x_i - \bar{x})(xi - \bar{x})}{N-1}$$

1.2.2 covarianza $cov(x, y)$

La covarianza entre dos variables x e y es una medida de la relación "promedio" éstas. Es la desviación promedio del producto cruzado entre ellas:

$$cov(x, y) = \frac{\sum(x_i - \bar{x})(yi - \bar{y})}{N - 1}$$

Veamos un ejemplo de dos variables que co-varían (respuesta ~ dosis en 5 pacientes), es decir, que están correlacionadas.
Los datos: dosis=(8,9,10,13,15); resp=(5,4,4,6,8);



1.2.2.1 Cálculo "a mano" de la covarianza de dos variables

$cov(dosis, resp) = \frac{\sum(x_i - \bar{x})(yi - \bar{y})}{N - 1} = \frac{(-0.4)(-3) + (-1.4)(-2) + (-1.4)(-1) + (0.6)(2) + (2.6)(4)}{4} = \frac{17}{4} = 4.25$

Un valor de covarianza positivo indica que ambas variables se desvían de la media en la misma dirección, mientras que uno negativo indica que las desviaciones acontecen en sentidos opuestos.

1.2.2.2 Cálculo en R de las medias y desviaciones para cada variable, así como el coeficiente de covariación

```r
# genermos los vectores dosis y resp
dosis <- c(5,4,4,6,8)
resp <- c(8,9,10,13,15)
# calculemos la dosis y respuesta medias
dosis.mean <- mean(dosis); resp.mean <- mean(resp);
cat("dosis media =", dosis.mean, "; resp.mean =", resp.mean)
```

```
## dosis media = 5.4 ; resp.mean = 11
```

```r
# cálculo de las desviaciones de cada dosis y respuesa con respecto a sus
# valores promedio
dosis.dev <- dosis - mean(dosis); resp.dev <- resp - mean(resp);
cat("dosis.dev =", dosis.dev, "; resp.dev =", resp.dev)
```

```
## dosis.dev = -0.4 -1.4 -1.4 0.6 2.6 ; resp.dev = -3 -2 -1 2 4
```

```r
# Cálculo del coef. de covariación
Covar <- sum((dosis.dev)*(resp.dev))/(length(dosis)-1); cat("cov =", Covar)
```

```
## cov = 4.25
```

```r
# cálculo de la covariación entre dosis y respuesta con cov(x,y)
cov(dosis,resp)
```

```
## [1] 4.25
```

El problema de usar la covarianza como medida de relación entre variables estriba en que depende de la escala de las medidas usadas. Es decir, **la covarianza no es una medida estandarizada**. Por tanto la covarianza no puede ser usada para comparar las relaciones entre variables medidas en diferentes unidades.

1.2.3 El coeficiente de correlación de Pearson r = coef. de covariación estandarizado

Para resolver el problema de dependencia de la escala o unidades de las mediciones (valores), necesitamos una unidad a la cual pueda convertirse cualquier medida. Esta **unidad de medida libre de escala** es la **desviación estándar (s ó σ)**. Al igual que la varianza, mide la desviación promedio de los datos con respecto a la media aritmética por no ser otra cosa que la $\sqrt{varianza}$ ó $\sqrt{s^2}$. Al dividir cualquier distancia de la media por la desviación estándar, obtendremos una distancia en unidades de desviación estándar.

Por tanto, **para normalizar la covarianza la tenemos que dividir por la desviación estándar**. Como la covarianza se calcula para dos variables cov(x,y), tenemos que calcular la desviación estándar para cada variable, multiplicándolas entre ellas, es decir:

$$\text{Coef. de correlación de Pearson}(r) = \frac{cov(x,y)}{s_x s_y} = \frac{\sum(x_i - \bar{x})(yi - \bar{y})}{(N-1)s_x s_y}$$

1.2.3.1 Cálculo del coeficiente de correlación de Pearson en R:

```r
# uso básico de la función cor()
cor(dosis,resp)
```

```
## [1] 0.8711651
```

```r
# comprobamos el resultado usando la fórmula de r indicada arriba
cov(dosis,resp)/(sd(resp)*sd(dosis))
```

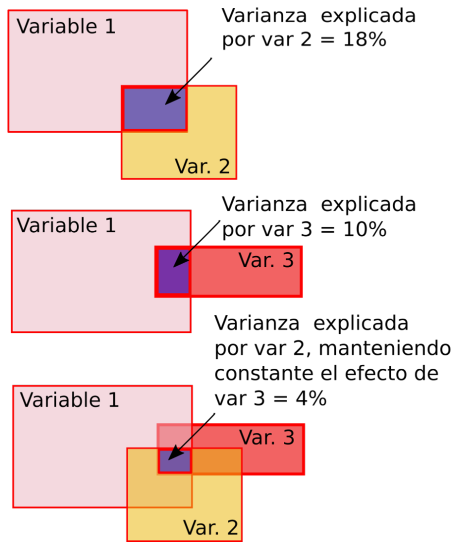
```
## [1] 0.8711651
```

1.2.4 Correlaciones parciales

Permiten evaluar la correlación entre dos variables (Var.1 y Var.2) considerando el efecto (varianza) de una tercera (Var.3) o más variables.

Eliminando la varianza compartida por las variables de interés con la o las variables auxiliares, obtenemos una medida de r que refleja los efectos de las variables de interés primario.

En R podemos hacer análisis de correlación parcial usando la función $pcor()$ del paquete ggm. Veremos en la prática el uso de las funciones $ggm :: pcor()$ y $ggm::cpor.test().



1.2.5 Supuestos hechos por el estadístico de correlación de Pearson r

- Ambas deben ser variables cuantitativas contínuas (medidas de intervalo)
- Si queremos correr tests de significancia, las variables deben estar normalmente distribuidas

1.2.6 El coeficiente de correlación no paramétrico de Kendall τ

El coeficiente de correlación τ de Kendall es no paramétrico, es decir, se puede usar cuando se viola el supesto de distribución normal de las variables a comparar. La correlación τ de Kendall es particularmente adecuada cuando tenemos un set de datos pequeño con muchos valores en el mismo rango o clase. Se puede usar por ejemplo con datos categóricos codificados binariamene (0,1). Estudios estadísticos han demostrado que el coeficiente de correlación τ de Kendall es un mejor estimador de la correlación en la población que el coeficiente de correlación no paramétrico de Spearman ρ, por lo que se recomienda usar τ para análisis de datos no paramétricos. En R se puede estimar la correlación τ o ρ cambiando el valor del argumento method="kendall" o method="spearman" de la función *cor*, en la que por defecto method="pearson". Por ejemplo: $cor(x, y, method = "kendall")$

1.2.7 El coeficiente de determinación R^2

El coeficiente de correlación elevado al cuadrado es el **coeficiente de determinación,** R^2, que mide la cantidad de variación en una variable que es compartida por otra. Vimos en el ejemplo anterior que la r para cor(dosis,resp) era de 0.8711651, y por tanto $R^2 = 0.7589286$. Por tanto podemos decir que la respuesta comparte un ~76% de la variación mostrada por la dosis. Tengan en cuenta de nuevo que compartir variabilidad no implica necesariamente causalidad.

```
# Cálculo del coeficiente de determinación, expresado como porcentaje, de la correlación
# entre dosis y respuesta.
cat("R^2(dosis,resp) =", round( cor(dosis,resp)^2 * 100, 1), "%.")
```

```
## R^2(dosis,resp) = 75.9 %.
```

1.3 Significancia del coeficiente de correlación (r)

Se trata de probar la hipótesis de que $r \neq 0$, es decir, buscamos rechazar la $H_0 : r = 0$.
Tenemos dos maneras de retar la H_0: *i*) usando $z - scores$; *ii*) mediante estadístico t.

1.3.1 Cálculo de la significancia de r usando $z - scores$

Los $z - scores$ son útiles ya que conocemos la probabilidad de ocurrencia de un valor de z determinado, siempre y cuando la distribución de la que proviene sea normal. Dado que es sabido que r tiene una distribución muestral no normal, tenemos que hacer una **transformación de Fisher** para normalizarla:

$$z_r = \frac{1}{2} log_e \left(\frac{1 + r}{1 - r} \right)$$

La z_r resultante tiene un error estándar de:

$$SE_{z_r} = \frac{1}{\sqrt{N - 3}}$$

- Para nuestro ejemplo tenemos que $r = .87$, $z_r = 1.33$ y $SE_{z_r} = .71$, como se demustra abajo aplicando las dos fórmulas arriba enunciadas:

```
# el valor de r previamente calculado
r <- round(cor(dosis,resp), 2) # .87

# Cálculo del z-score para r
Zr <- 0.5*(log((1+r)/(1-r)))
cat("z-score para r (Zr): ", Zr)
```

```
## z-score para r (Zr):  1.33308
```

```
# Cálculo del error estándar de zr
SEzr <- 1/sqrt(5-3)
cat("error estándar de zr (SEzr):", SEzr)
```

```
## error estándar de zr (SEzr): 0.7071068
```

Ahora podemos transformar esta r ajustada a un $z - score$, tal y como vimos anteriormente para los scores crudos. Recuerden que para calcular un $z - score$ que represente el tamaño de la correlación con respecto a un valor particular, simplemente calculamos el $z - score$ contra dicho valor usando el SE_{z_r} asociado. Este valor generalmente va a ser 0, ya que queremos probar que $H_1 \neq H_0$, es decir, trataremos de rechazar la $H_0 : r = 0$. Por tanto

$$z = \frac{z_r}{SE_{z_r}}$$

- **Cálculo de la significancia de r usando z**

```
z <- Zr/SEzr
cat("valor de z: ", z, "## ¿Será significativo?")
```

```
## valor de z:  1.885259 ## ¿Será significativo?
```

```
pvalue1sided = pnorm(-abs(z))
cat("valor p de una cola: ", pvalue1sided)
```

```
## valor p de una cola:  0.02969742
```

```
pvalue2sided <- 2*pnorm(-abs(z))
cat("valor p de dos colas: ", pvalue2sided, "## <<< Generalmente usamos este caso")
```

```
## valor p de dos colas:  0.05939484 ## <<< Generalmente usamos este caso
```

Pueden usar la función *convert.z.score*() para convertir $z - scores$ a $p - values$, mostrada abajo.

```r
# Vean cómo se importa código de en un archivo almacenado en el disco local
source("./code/func_convert.z.score.R")

# con este comando podemos imprimir a STDOUT (consola) el contenido del archivo guardado
# en disco
cat(readLines('./code/func_convert.z.score.R'), sep = '\n')
```

```
## # función para convertir z-scores a p-values, tomada de https://www.biostars.org/p/17227/
## convert.z.score<-function(z, one.sided=NULL) {
##   if(is.null(one.sided)) {
##     pval = pnorm(-abs(z));
##     pval = 2 * pval
##   } else if(one.sided=="-") {
##     pval = pnorm(z);
##   } else {
##     pval = pnorm(-z);
##   }
##   return(pval);
## }
```

```r
# Con estas líneas llamamos a la función que hemos importado con source
pval.2s <- convert.z.score(z)
pval.1s <- convert.z.score(z, 1)

# imprimimos el contenido de las variables que almacenan el resultado devuelto
# por la función
cat("2-sided pval: ", pval.2s, "| 1-sided pval: ", pval.1s)
```

```
## 2-sided pval:  0.05939484 | 1-sided pval:  0.02969742
```

1.3.2 Cálculo de la significancia de r mediante el estadístico-t y la funcion *cor.test*()

En R el test de significancia de r está implementado en la función *cor.test*(), basado en el estadístico-t. con $N - 2$ grados de libertad, que pueden obtenerse directamente de r:

$$t_r = \frac{r\sqrt{(N-2)}}{\sqrt{1-r^2}}$$

```r
#>>> Cálculo a mano de la significancia de r
# Cálculo de r
r <- cor(dosis,resp)

# Cálculo del estadístico-t
tr <- r*(sqrt((5-2)))/sqrt((1-r**2))
cat("valor del estadístico-t: ", tr)
```

```
## valor del estadístico-t:  3.073181
```

```r
# cálculo de la p del estadístico
ptr <- 2*pt(tr, 3, lower.tail = FALSE)
cat("valor de la p para el estadístico-t: ", ptr)
```

```
## valor de la p para el estadístico-t:  0.05442624
```

```r
#>>> Cálculo automático con la función de R cor.test()
(cor.test(dosis, resp))
```

```
##
##  Pearson's product-moment correlation
##
## data:  dosis and resp
## t = 3.0732, df = 3, p-value = 0.05443
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.0479747  0.9914236
## sample estimates:
##       cor
## 0.8711651
```

1.3.3 Análisis de potencia y significancia estadística de r

El **análisis de potencia** es un aspecto importante del diseño experimental. Nos permite determinar el tamaño muestral (la famosa n) requerido para detectar un efecto de un determinado tamaño con un grado determinado de confianza. De modo complementario, tambíen nos permite determinar la probabilidad de detectar un efecto de un tamaño determinado, dados un nivel de confiaza y tamaño de muestra predeterminados. Así por ejemplo, si el nivel de confianza o estima del tamaño del efecto no son satisfactorios para un n dada, lo único aconsejable es incrementar la n o abandonar el experimento.

Las siguientes cuatro magnitudes están íntimamente relacionadas:
tamaño muestal (número de observaciones)
tamaño del efecto (medida de la fuerza de un fenómeno; la magnitud del estadístico, r, t, chi^2 ...) wikipedia
nivel de significancia = P(error Tipo I) = probabilidad de observar un effecto inexistente (usualmente $\alpha = .05$)
potencia = 1 - P(error Tipo II) = probabilidad de observar un effecto real (un valor frecuentemente usado es 0.8)

Dadas cualesquiera tres, podemos determinar la cuarta magnitud.

Recordemos los tipos de error: - **Error de tipo I**: ocurre cuando *estimamos que existe un efecto* en la población, cuando *en realidad no hay tal*. - **Error de tipo II**: ocurre cuando *estimamos que no existe un efecto* en la población, cuando *en realidad sí lo hay*.

Los coeficientes de correlación son un ejemplo clásico de **tamaños del efecto**, por tanto podemos interpretar r sin la necesidad de *valoresp*, particularmente debido a que los *valoresp* están relacionados con el tamaño de la muestra. Hay muchos estadísticos que recomiendan no obsesionarse con los famosos *valoresp* para coeficientes de correlación, o coeficientes de regresión, ya que ambos son en sí estimadores de tamaños del efecto.

1.4 La importancia de visualizar gráficamente los datos antes de someterlos a análisis de correlación: lecciones del cuarteto de Anscombe

De lo explicado arriba debe quedar claro que el análisis de correlación sólo debe aplicarse entre pares de variables con relación lineal entre ellas. Es por tanto importante siempre graficar los datos para visualizar el tipo de relaciones entre las variables y detectar la presencia de valores atípicos o aberrantes ("outliers").

El cuarteto de Anscombe comprende cuatro conjuntos de datos que tienen las mismas propiedades estadísticas (medias y varianzas), presentando los mismos valores de correlación entre pares de variables, con el mismo ajuste lineal (recta de regresión), pero que son marcadamente distintas al inspeccionarlas gráficamente.

Cada conjunto consiste de once puntos (x, y) y fueron construidos por el estadístico F. J. Anscombe. El cuarteto es una demostración de la importancia de visualizar gráficamente un conjunto de datos antes de someterlos a análisis estadísticos.

R trae el set de datos de Anscombe en el paquete *datasets*. Exploremos sus características estadísticas, grafiquemos las relaciones entre los pares de variables x1-y1, x2-y2, x3-y3, x4-y4 y discutamos los resultados.

1.4.1 Código para generar las gráficas y estadísticas del cuarteto de Anscombe

Nota: este código puede ser un poco difícil de entender ya que contiene un bucle for y otros detalles sintácticos que no hemos visto en el curso. Muestro el código para los interesados en ejemplos para aprender estos elementos esenciales del lenguaje. Pero lo importante no es el código, sino las gráficas y análisis estadísticos resultantes, que discutiremos en la siguiente sección.

```
# visualizemos el conjunto de datos de Anscombe
# recuerda: str(anscombe) nos muestra la estructura del objeto
anscombe
```

```
##    x1 x2 x3 x4    y1   y2    y3    y4
## 1  10 10 10  8  8.04 9.14  7.46  6.58
## 2   8  8  8  8  6.95 8.14  6.77  5.76
## 3  13 13 13  8  7.58 8.74 12.74  7.71
## 4   9  9  9  8  8.81 8.77  7.11  8.84
## 5  11 11 11  8  8.33 9.26  7.81  8.47
## 6  14 14 14  8  9.96 8.10  8.84  7.04
## 7   6  6  6  8  7.24 6.13  6.08  5.25
## 8   4  4  4 19  4.26 3.10  5.39 12.50
## 9  12 12 12  8 10.84 9.13  8.15  5.56
## 10  7  7  7  8  4.82 7.26  6.42  7.91
## 11  5  5  5  8  5.68 4.74  5.73  6.89
```

```
summary(anscombe)
```

```
##        x1             x2             x3             x4
##  Min.   : 4.0   Min.   : 4.0   Min.   : 4.0   Min.   : 8
##  1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 8
##  Median : 9.0   Median : 9.0   Median : 9.0   Median : 8
##  Mean   : 9.0   Mean   : 9.0   Mean   : 9.0   Mean   : 9
##  3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.: 8
##  Max.   :14.0   Max.   :14.0   Max.   :14.0   Max.   :19
##        y1               y2              y3              y4
##  Min.   : 4.260   Min.   :3.100   Min.   : 5.39   Min.   : 5.250
##  1st Qu.: 6.315   1st Qu.:6.695   1st Qu.: 6.25   1st Qu.: 6.170
##  Median : 7.580   Median :8.140   Median : 7.11   Median : 7.040
##  Mean   : 7.501   Mean   :7.501   Mean   : 7.50   Mean   : 7.501
##  3rd Qu.: 8.570   3rd Qu.:8.950   3rd Qu.: 7.98   3rd Qu.: 8.190
##  Max.   :10.840   Max.   :9.260   Max.   :12.74   Max.   :12.500
```

```
# calculemos las medias y varianza de cada columna
# Noten el uso de la función sapply
sapply(anscombe, mean)
```

```
##       x1       x2       x3       x4       y1       y2       y3       y4
## 9.000000 9.000000 9.000000 9.000000 7.500909 7.500909 7.500000 7.500909
```

```r
sapply(anscombe, var)
```

```
##        x1        x2        x3        x4        y1        y2        y3
## 11.000000 11.000000 11.000000 11.000000  4.127269  4.127629  4.122620
##        y4
##  4.123249
```

```r
# veamos ahora las correlaciones entre las viables x1-y1, x2-y2 ...
cor(anscombe[,1:4], anscombe[,5:8])
```

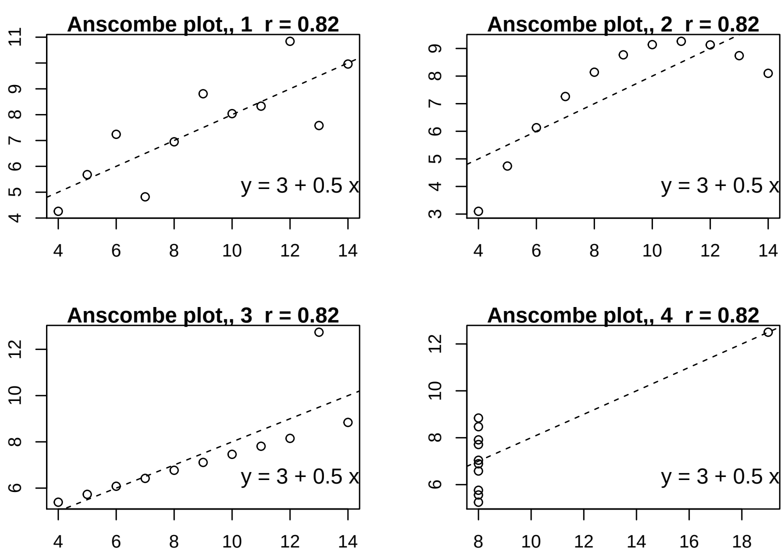
```
##            y1         y2         y3         y4
## x1  0.8164205  0.8162365  0.8162867 -0.3140467
## x2  0.8164205  0.8162365  0.8162867 -0.3140467
## x3  0.8164205  0.8162365  0.8162867 -0.3140467
## x4 -0.5290927 -0.7184365 -0.3446610  0.8165214
```

```r
# de la matriz anterior realmente necesitamos sólo su diagonal
diag(cor(anscombe[,1:4], anscombe[,5:8]))
```

```
## [1] 0.8164205 0.8162365 0.8162867 0.8165214
```

```r
# Grafiquemos las relaciones entre las virables x1-y1, x2-y2 ...
# El objetivo del código es:
# 1) mediante par(mfrow=c(2,2), mar=c(6,4,1,1)) controlamos que las
#    4 gráficas queden en un solo display, como una matriz de 2x2.
# 2) Mediante dos bucles for anidados, obtenemos los índices para
#    poder indexar el dataframe asncombe, atendiendo a los números
#    de las columnas de las variables x,y que nos interesan: x=(1,2,3,4), y=(5,6,7,8).
# 3) Lo que sigue es llamar a plot para cada par x,y
# 4) para finalmente graficar la regresión lineal (explicado en el próximo tema).

# La información relevante (r y recta de regresión se imprime como texto sobre la gráfica)
# Las líneas correspondientes están comentadas en el código
# Si les parece complicada mi solución, vean la que propone la documentación de R
# tecleando help(anscombe); tiene detalles sintácticos muy interesantes

# par() controla los parámetros gráficos. guardamos en oldpar <- los ajustes originales
# de par()
oldpar <- par()
# plotea 4 gráficas en 2 filas y 2 columnas, dando  márgenes entre ellas
par(mfrow=c(2,2), mar=c(4,4,1,1))
for(i in 1:4){ # 1,2,3,4
    j <- i+4  # 5,6,7,8
    # las siguientes variables capturan información que añadiremos al título de cada
    # plot (..., main=h)

    # función de redondeo, a 2 dígitos
    r <- round(cor(anscombe[,i], anscombe[,j]), 2)
    # pegamos diferentes elementos en una cadena
    h <- paste("Anscombe plot,,", i, " r =", r, sep = ' ')

    # ajustemos un modelo lineal lm() y capturemos sus coeficientes:
    # corte eje ordenadas (y) y pendiente. Capturamos los valores y los
    # pegamos en la fórmula general de la recta: y = a + bx.
    # lm() ajusta un modelo lineal a los datos
    fit <- lm(anscombe[,j] ~ anscombe[,i])
```

```
    # con fit$coefficients[[1]] extraemos del objeto fit (una lista),
    # el primer [[1]] y segundo [[2]]  elemento almacenados en el vector
    # llamado coefficients. Nótese el uso de [[]] para sacar elementos de una lista!!!
    regr <- paste("y =", round(fit$coefficients[[1]], 2), "+",
                  round(fit$coefficients[[2]], 1), "x")

    # estas líneas son para los "scatterplots" de los pares de variables
    plot(anscombe[,i], anscombe[,j], main = h, xlab = "", ylab = "")

    # y para graficar la recta de regresión
    abline(lm(anscombe[,j] ~ anscombe[,i]), lty=2)

    # con mtext() podemos escribir los coeficientes de la fórmula dentro del área
    # de cada gráfica especificando el lado derecho (side=1), pegado a la derecha
    # (adj = 1), y dos líneas abajo de la línea central (line = -2)
    mtext(regr, side = 1, adj = 1, line = -2)
}
```




```
# re-establecemos los parámetros gráficos originales
par(oldpar)
```

1.4.2 Discusión sobre los resultados de las gráficas y análisis estadístico del cuarteto de Anscombe

El primer gráfico muestra lo que parece una relación lineal simple, correspondiente a dos variables correlacionadas que satisfacen el supuesto de normalidad. El segundo gráfico no está distribuido normalmente, aunque se observa relación entre los datos, esta no es lineal y el coeficiente de correlación de Pearson no es relevante ya que se viola claramente el supuesto de relación lineal. En la tercera gráfica la distribución es lineal pero con una línea de regresión diferente de la que se sale el dato extremo que influye lo suficiente como para alterar la línea de regresión y disminuir el coeficiente de correlación de 1 a 0.816. Por último, la cuarta gráfica (abajo a la derecha) es un ejemplo de muestra en la que un valor atípico es suficiente para producir un coeficiente de correlación alto incluso cuando la relación entre las dos variables no es lineal.

1.5 Resumen de conceptos clave

- Al estandarizar la covarianza, r variará entre ± 1
- La **correlación es positiva** si ambas variables covarían en el mismo sentido
- La **correlación es negativa** si ambas variables covarían en sentidos opuestos
- $r = \pm 1$ implica una correlación perfecta (ajuste perfecto a modelo lineal) entre la variable de respuesta y la var. independiente
- $r = 0$ implica que no existe correlación alguna entre la variable de respuesta y la var. independiente
- Dado que r es una medida estandarizada, se usa frecuentemente para medir el **tamaño de un efecto**, que generalmente se interpretan así:
  - **efecto despreciable**: $r < |0.1|$
  - **efecto pequeño**: $|0.1| < r <= |0.3|$

  - **efecto mediano**: $|0.3| < r <= |0.5|$
  - **efecto grande**: $r > |0.5|$
- Es importante **explorar los datos visualmente mediante un gráficos de dispersión**, ya que r sólo puede aplicarse a pares de variables que covarían linealmente. Recuerden las enseñanzas derivadas del análisis del cuarteto de Anscombe.
- Podemos usar **correlaciones parciales** para obtener un valor de r más realista entre las variables x,y, al determinar la porción de la varianza propia o atribuida específicamente a ellas, al considerar y por tanto eliminar los efectos (varianza) de una o más variables de control que ejercen efecto sobre x e y.

2 Correlación - prácticas

Vamos a usar las funciones del paquete base de R para análisis de correlación: *cor* y *cor.test*

Las opciones básicas son:

- **cor(x, use= , method=)**
  Donde:
  x: Matrix or data frame.
  use: Specifies the handling of missing data. The options are **all.obs** (assumes no missing data—missing data will produce an error), **everything** (any correlation involving a case with missing values will be set to missing), **complete.obs** (listwise deletion), and **pairwise.complete.obs** (pairwise deletion).
  method: Specifies the type of correlation. The options are **pearson**, **spearman**, or **kendall**.

- **cor.test(x, y, alternative = , method =)**
  x, y: numeric vectors of data values. x and y must have the same length.
  alternative: indicates the alternative hypothesis and must be one of "two.sided", "greater" or "less". You can specify just the initial letter. "greater" corresponds to positive association, "less" to negative association.

La lista completa de cada función la puedes ver con **?cor** o **help(cor.test)**

2.1 Paquetes adicionales útiles para análisis de correlación

Usaremos los paquetes *car*, *corrplot*, *psych*, *ggm* y *pwr*, que puedes instalar, junto con sus dependencias, con el comando: install.packages(c("car", corrplot“,”psych“,”ggm“,”pwr“), dep=TRUE)

2.2 Datos

Usaremos el set de datos *states.x77* del paquete base *datasets* que R carga por defecto

Veamos cómo obtener información sobre este conjunto de datos:

```r
# veamos la lista de datos pre-cargados en el ambiente
data()
help(package="datasets")

# más detalles con help
help("state")

# exploremos los detalles de states.x77
head(state.x77)
```

```
##            Population Income Illiteracy Life Exp Murder HS Grad Frost
## Alabama          3615   3624        2.1    69.05   15.1    41.3    20
## Alaska            365   6315        1.5    69.31   11.3    66.7   152
## Arizona          2212   4530        1.8    70.55    7.8    58.1    15
## Arkansas         2110   3378        1.9    70.66   10.1    39.9    65
## California      21198   5114        1.1    71.71   10.3    62.6    20
## Colorado         2541   4884        0.7    72.06    6.8    63.9   166
##              Area
## Alabama     50708
## Alaska     566432
## Arizona    113417
## Arkansas    51945
## California 156361
## Colorado   103766
```

```r
colnames(state.x77)
```

```
## [1] "Population" "Income"     "Illiteracy" "Life Exp"   "Murder"
## [6] "HS Grad"    "Frost"      "Area"
```

```r
# recureda estos comandos muy útiles para hacer la primera exploración de un dataframe
#dim(state.x77)
#str(state.x77)
#summary(state.x77)
```

2.3 Análisis de correlación en R

2.3.1 Objetivos

1. Haremos un análisis de correlación de Pearson, calcularemos las significancias de las correlaciones con *cor.test*() del paquete *stats* de base, así como con *corr.test*() del paquete *psych* (*psych* :: *corr.test*())
2. graficaremos los resultados en forma de matrices de correlación usando *corrplot* :: *corrplot*

3. Correremos también correlaciones parciales con *pcor*() del paquete *ggm*.
4. Determinaremos la potencia estadístico de los datos usados para el ejercicio 3 mediante el uso de *pwr* :: *pwr.r.test*()
5. Haremos un análisis de correlación no paramétrico sobre variables codificadas binariamente, usando la τ de Kendall.

2.3.2 Ejercicio 1: análisis de correlación de Pearson usando *cor*(), *cor.test*(), *psych* :: *corr.test*() y *corrplot* :: *corrplot*()

```r
# 1. Carguemos de una vez las librerías
library("corrplot") # corrplot::corrplot()
library("psych")    # psych::corr.test
library("ggm")      # ggm::pcor
library("car")      # car::scatterplotMatrix

# 2. Vamos a reducir el dataframe original a las primeras 6 columnas para hacer menos
# voluminosa la salida
states<- state.x77[,1:6] # ojo: states es una matriz, no un dataframe!
class(states)
```

```
## [1] "matrix"
```

```r
head(states); tail(states)
```

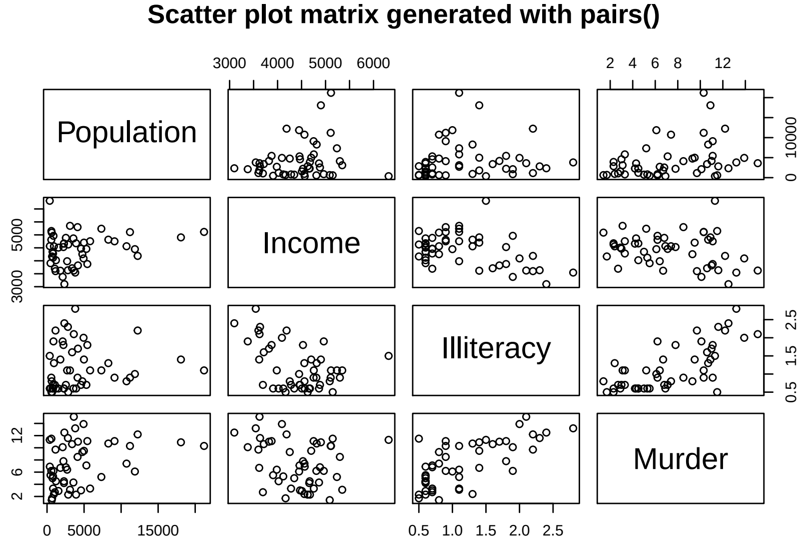
```
##            Population Income Illiteracy Life Exp Murder HS Grad
## Alabama          3615   3624        2.1    69.05   15.1    41.3
## Alaska            365   6315        1.5    69.31   11.3    66.7
## Arizona          2212   4530        1.8    70.55    7.8    58.1
## Arkansas         2110   3378        1.9    70.66   10.1    39.9
## California      21198   5114        1.1    71.71   10.3    62.6
## Colorado         2541   4884        0.7    72.06    6.8    63.9
```

```
##               Population Income Illiteracy Life Exp Murder HS Grad
## Vermont              472   3907        0.6    71.64    5.5    57.1
## Virginia            4981   4701        1.4    70.08    9.5    47.8
## Washington          3559   4864        0.6    71.72    4.3    63.5
## West Virginia       1799   3617        1.4    69.48    6.7    41.6
## Wisconsin           4589   4468        0.7    72.48    3.0    54.5
## Wyoming              376   4566        0.6    70.29    6.9    62.9
```

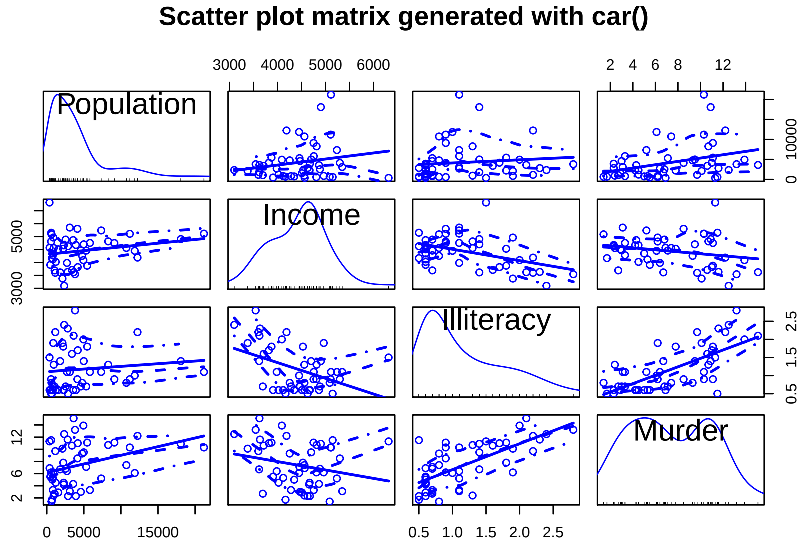
```r
# 3. Hagamos un análisis de correlación de Pearson entre todos los pares de variables
# (columnas) numéricas.
# Cuáles son las correlaciones más fuertes que encuentras?
# Te parecen lógicas o verosímiles?

# use="everything", method="pearson" son los valores por defecto  de estos parámetros.
cor(states)
```

```
##             Population     Income Illiteracy    Life Exp     Murder
## Population  1.00000000  0.2082276  0.1076224 -0.06805195  0.3436428
## Income      0.20822756  1.0000000 -0.4370752  0.34025534 -0.2300776
## Illiteracy  0.10762237 -0.4370752  1.0000000 -0.58847793  0.7029752
## Life Exp   -0.06805195  0.3402553 -0.5884779  1.00000000 -0.7808458
## Murder      0.34364275 -0.2300776  0.7029752 -0.78084575  1.0000000
## HS Grad    -0.09848975  0.6199323 -0.6571886  0.58221620 -0.4879710
##                HS Grad
## Population -0.09848975
## Income      0.61993232
## Illiteracy -0.65718861
## Life Exp    0.58221620
## Murder     -0.48797102
## HS Grad     1.00000000
```

```r
# 3.1 visualización de las correlaciones entre múltiples variables en matrices,
# usando pairs()
pairs(~Population+Income+Illiteracy+Murder, data = states,
      main ="Scatter plot matrix generated with pairs()")
```




```r
# 3.2 visualización de las correlaciones entre múltiples variables en matrices,
#     usando car()
car::scatterplotMatrix(~Population+Income+Illiteracy+Murder, data = states,
                       main ="Scatter plot matrix generated with car()",
                       spread=FALSE, smoother.args=list(lty=2))
```




```r
# 3.3 Como vieron en 3, por defecto se producen matrices cuadradas, calculando r
# para todos los pares de variables numéricas. Pero a veces queremos focalizar el
# análisis a unas variables en particular, para estudiar su variación con respecto
# a otro conjunto de variables a la luz de las cuales queremos entender la variación
# de las primeras. Por ejemplo, ¿Cómo influyen "Population", "Income", "Illiteracy"
# y "HS Grad" en las variables de interés primario "Life Exp" y "Murder"?
x <- states[,c("Population", "Income", "Illiteracy", "HS Grad")]
y <- states[,c("Life Exp", "Murder")]
```

```r
cor(x,y)
```

```
##               Life Exp     Murder
## Population -0.06805195  0.3436428
## Income      0.34025534 -0.2300776
## Illiteracy -0.58847793  0.7029752
## HS Grad     0.58221620 -0.4879710
```

```r
# 4. Cálculo de la significancia de la correlación entre Income vs. Illiteracy,
# Income vs. HS Grad. Nótese que cor.test() sólo puede tomar vectores de
# valores x,y, no dataframes o matrices multicolumna
cor.test(states[,"Illiteracy"], states[,"Murder"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  states[, "Illiteracy"] and states[, "Murder"]
## t = 6.8479, df = 48, p-value = 1.258e-08
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5279280 0.8207295
## sample estimates:
##       cor
## 0.7029752
```

```r
cor.test(states[,"Illiteracy"], states[,"HS Grad"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  states[, "Illiteracy"] and states[, "HS Grad"]
## t = -6.0408, df = 48, p-value = 2.172e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.7908657 -0.4636561
## sample estimates:
##        cor
## -0.6571886
```

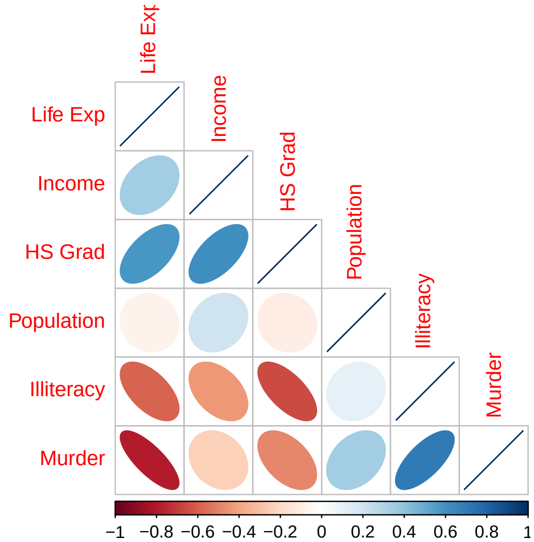
```r
# 4.1 Como vieron en 4, con la funcion cor.test() sólo podemos evaluar
# la significancia de correlaciones individuales (un par de variables cada vez).
# Si queremos evaluar múltiples pares de variables, necesitamos escribir
# una función que llame a cor.test() las veces que queramos o usar psych::corr.test()
?psych::corr.test # vean las opciones
psych::corr.test(states, use = "complete")
```

```
## Call:psych::corr.test(x = states, use = "complete")
## Correlation matrix
##            Population Income Illiteracy Life Exp Murder HS Grad
## Population       1.00   0.21       0.11    -0.07   0.34   -0.10
## Income           0.21   1.00      -0.44     0.34  -0.23    0.62
## Illiteracy       0.11  -0.44       1.00    -0.59   0.70   -0.66
## Life Exp        -0.07   0.34      -0.59     1.00  -0.78    0.58
## Murder           0.34  -0.23       0.70    -0.78   1.00   -0.49
## HS Grad         -0.10   0.62      -0.66     0.58  -0.49    1.00
## Sample Size
```

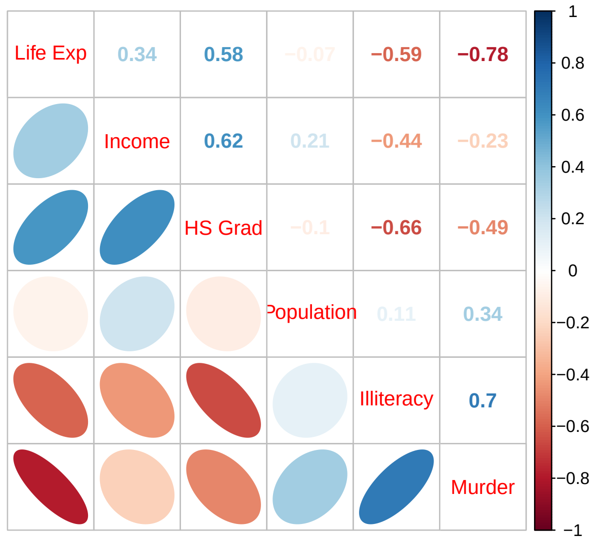
```
## [1] 50
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##            Population Income Illiteracy Life Exp Murder HS Grad
## Population       0.00   0.59       1.00      1.0   0.10       1
## Income           0.15   0.00       0.01      0.1   0.54       0
## Illiteracy       0.46   0.00       0.00      0.0   0.00       0
## Life Exp         0.64   0.02       0.00      0.0   0.00       0
## Murder           0.01   0.11       0.00      0.0   0.00       0
## HS Grad          0.50   0.00       0.00      0.0   0.00       0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

```r
# 5. visualizacion de matrices de correlación con corrplot
# Recuerden que para acceder a la ayuda del paquete pueden usar help() y vignette()
# help("corrplot")
# vignette("corrplot-intro")

# 5.1 - salida por defecto de corrplot
corrplot::corrplot(cor(states))
```

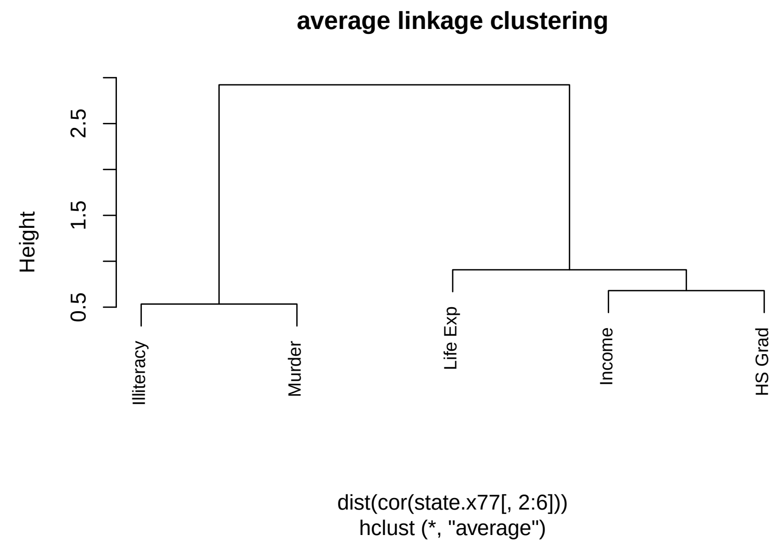

```r
# 5.2 - hagamos la salida más legible e informativa: usemos elipses para indicar
#       el grado  y signo de  la correlación,  mostrando sólo la diagonal inferior,
#       y ordenando los valores jerárquicamente
corrplot::corrplot(cor(states), method="ellipse", type="lower", order="hclust")
```




```
# 5.3 Similar como arriba, pero añadiendo una diagonal superior con los
#      valores numéricos de r
corrplot::corrplot.mixed(cor(states), lower="ellipse", upper="number", order="hclust")
```




```
# 6. Por último podemos hacer un análisis de agrupamiento jerárquico de la matriz
#    de correlación
fit.average <- hclust(dist(cor(state.x77[,2:6])), method = "average")
plot(fit.average, cex=.8, main = "average linkage clustering")
```



2.3.3 Ejercicio 2: análisis de correlación parcial con $ggm :: pcor()$, y evaluación de la significancia de r con $ggm :: pcor.test()$

La correlación parcial se calcula entre dos variables cuantitativas, controlando para el efecto de una tercera o más variables cuantitativas adicionales.
La forma básica del comando es $ggm :: pcor(u, s)$, donde u es un vector numérico de tipo *int*, con los dos primeros números correspondiendo a los índices de las variables a ser correlacionadas, los demás números representando los índices de las variables condicionantes. s es la matriz de covarianza entre todas las variables incluidas en u. Es el uso de la covarianza para la matriz total lo que permite corregir la correlación entre las dos variables focales considerando el efecto condicionante de las demás variables auxiliares.

```
# 6 Hagamos un análisis de correlación parcial, es decir, considerando el efecto de las
#    variables auxiliares. Compara e interpreta estos resultados con respecto a los
#     obtenidos en el análisis de correlación de Pearson.
library("ggm")
colnames(states)
```

```
## [1] "Population" "Income"     "Illiteracy" "Life Exp"   "Murder"
## [6] "HS Grad"
```

```
# (c(1:length(colnames(states)))) # imprime los índices numéricos
# Population ~ Murder
ggm::pcor(c(1,5,2,3,6), cov(states))
```

```
## [1] 0.3462724
```

```
# Illiteracy ~ Murder
ggm::pcor(c(3,5,2,1,6), cov(states))
```

```
## [1] 0.5989741
```

```
# 7. Finalmente evaluemos la significancia de las correlaciones parciales
#    ggm::pcor.test(r, q, n);  r: coef. de correlación parcial calculado por ggm::pcor;
#    q: número de variables condicionantes; n > 0: tamaño muestral

pcor1 <- ggm::pcor(c(1,5,2,3,6), cov(states))
pcor1
```

```
## [1] 0.3462724
```

```r
ggm::pcor.test(pcor1, 3, length(row.names(states)))
```

```
## $tval
## [1] 2.476049
##
## $df
## [1] 45
##
## $pvalue
## [1] 0.01711252
```

2.3.4 Ejercicio 3: Evaluación del efecto del tamaño de muestra sobre la significancia de r mediante análisis de potencia usando $pwr::pwr.r.test()$

Como ya se ha mencionado, **los coeficientes de correlación representan tamaños de efecto**, por tanto podemos interpretar r sin la necesidad de $valoresp$, particularmente debido a que éstos están relacionados con el tamaño de la muestra. Por tanto no hay que obsesionarse con los famosos $valoresp$, como recomiendan muchos estadísticos modernos (por ejemplo (**???**)). No obstante, puede ser intersante hacer un **test de potencia** de los datos para determinar si tenemos una muestra suficientemente grande como para que tenga poder estadístico. Podemos usar el paquete *pwr* de R para determinar la potencia estadística de nuestros datos para una amplia gama de tests estadísticos. Ilustraremos el uso de *pwr* para determinar r

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

donde:
n: es el tamaño muestal (no. de observaciones)
r: es el coeficiente de correlación lineal.
$sig.level$: es el nivel de significancia (proabilidad del error de Tipo I) # generalmente 0.05 $power$: potencia o fuerza del test (1 - probabilidad de error de Tipo II) # generalmente 0.8

Como vimos en 1.3.3, dados cualesquiera tres de estos valores, podremos calcular el cuarto, como mostraremos seguidamente.

Usamos el coef. de correlación poblacional com la medida de tamaño de efecto. Cohen sugiere que valores de r de 0.1, 0.3, y 0.5 representan tamaños de efecto pequeño. medio y grande, respectivamente.

```r
# Análisis de potencia para r
# NOTA IMPORTANTE: exactamente uno de n, r, power, y sig.level tiene que ser = NULL
library("pwr")
# análisis de potencia de r: predicción de la potencia, dado un tamaño de muestra,
# valro de r y nivel de significancia
pwr.r.test(n = length(row.names(states)), r = pcor1, sig.level = .05, power = NULL)
```

```
##
##      approximate correlation power calculation (arctangh transformation)
##
##               n = 50
##               r = 0.3462724
##       sig.level = 0.05
##           power = 0.70466
##     alternative = two.sided
```

```r
# análisis de potencia de r: predicción de p, dado un tamaño de muestra, valor de r
# y nivel de potencia  de 0.8, que es un varlo estándar.
pwr.r.test(n = length(row.names(states)), r = pcor1, sig.level = NULL, power = .8)
```

```
##
```

```
##      approximate correlation power calculation (arctangh transformation)
##
##               n = 50
##               r = 0.3462724
##       sig.level = 0.09698378
##           power = 0.8
##     alternative = two.sided
```

```
# análisis de potencia de r: predección del tamaño mínimo de n, para alcanzar
# niveles dados valores de r y potencia
pwr.r.test(n = NULL, r = pcor1, sig.level = .05, power = .8)
```

```
##
##      approximate correlation power calculation (arctangh transformation)
##
##               n = 62.32216
##               r = 0.3462724
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
```

2.3.5 Ejercicio 4 Análisis de correlación no paramétrico sobre variables categóricas codificadas binariamente, usando la τ de Kendall

```
# Análisis de correlación no paramétrico sobre variables categóricas
# codificadas binariamente, usando la $\tau$ de Kendall.
# ... TODO
```

3 Ejercicios de tarea

El ejercicio se va a hacer en base al conjunto de datos **mtcars** que distribuido en el paquete *datasets* de la instalación base de R. Responde de manera concisa y precisa a las siguientes preguntas, mostrando el código usado y la salida del mismo usando Rmarkdown y Rstudio, salvando la salida en un archivo html, que será el que subas al sistema moodle del curso, para su calificación.

1. ¿De qué publicación fueron extraídos estos datos?
2. ¿Qué estructura de datos es?
3. Indica las dimensiones de la estructura de datos e indica la clase de cada variable
4. Visualiza la correlación entre todos los pares de variables
5. ¿Qué pares de variables parecen covariar positivamente?
6. ¿Qué pares de variables parecen estar negativamente correlacionadas?

7. ¿Qué comparaciones pareadas no parecen ser adecuadas para un análisis de correlación, y porqué?
8. Haz un indexado del dataframe para extraer sólo las variables mpg, cyl, disp, np, wt, carb y calcula los coeficientes de correlación entre todas estas variables.
9. ¿Cuáles de estas correlaciones son significativas?

10. ¿Es suficientemente grande el número de muestras para la correlación mpg ~ wt para $\alpha = .05$, $potencia = .95$, $r = -0.87$? ¿Cuál es la potencia del test? ¿Qué siginifica esta última respuesta?
11. Haz un análisis de correlación parcial usando *mpg* y *wt* como las variables primarias, tomando cyl como variable auxiliar. Interpreta el resultado.

4 Bibliografía selecta

4.1 Libros

4.1.1 R y estadística

- Andy Field, Discovering statistics using R (A. P. Field, Miles, and Field 2012)
- Michael J. Crawley, Statistics - An introduction using R (Crawley 2015)
- Brian S. Everitt and Torsten Hothorn - A handbook of statistical analyses using R (Everitt and Hothorn 2014)

4.1.2 R - aprendiendo el lenguaje base

- Michael J. Crawley, The R book, 2nd edition (Crawley 2012)
- Paul Teetor, The R cookbook (Teetor and Loukides 2011)
- Richard Cotton, Learning R (Cotton 2013)
- Paul Murrell, R graphics (Murrell 2009)

4.1.3 R - manipulación, limpieza y visualización avanzada de datos con tidyr, dplyr y ggplot2

- Bradley C Boehmke, Data wrangling with R (Boehmke 2016)
- Hadley Wickham, ggplot2 - elegant graphics for data analysis (Wickham 2016)

4.1.4 R - aplicaciones en análisis de datos usando multiples paquetes

- Robert Kabacoff, R in action (Kabacoff 2015)
- Jared Lander, R for everyone (Lander 2014)

4.1.5 R - programación

- Garrett Grolemund, Hands on programming with R (Grolemund 2014)
- Norman Matloff, The art of R programming (Matloff 2011)

4.1.6 Investigación reproducible con R y RStudio

- Christopher Gandrud, Reproducible research with R and RStudio (Gandrud 2015)

5 Funciones y paquetes de R usados para este documento

5.1 Análisis de correlación

5.1.1 Funciones de paquetes base (R Core Team 2018)

- abline()
- class()
- colnames()
- cor()
- cor.test()

- cov()
- data()
- dist()
- lm()
- hclust()
- head()
- help()
- library()
- par()
- pairs()
- plot()
- str()
- subset()
- summary()
- tail()

5.1.2 Datos del paquetes base

- datasets [R-datasets]

5.1.3 Paquetes especializados

- car::scatterplotMatrix() (Fox, Weisberg, and Price 2018)
- corrplot::corrplot(); corrplot::corrplot.mixed() (Wei and Simko 2017)
- ggm::pcor(); ggm::pcor.test(); (Marchetti, Drton, and Sadeghi 2015)
- psych::corr.test() (Revelle 2018)
- pwr::pwr.r.test() (Champely 2018)

5.2 Paquetes y software para investigación reproducible y generación de documentos en múltiples formatos

- knitr (Xie 2018)
- pandoc (MacFerlane 2016)
- rmarkdown (Allaire et al. 2018)

6 Recursos en línea

6.1 The comprehensive R archive network (CRAN)

- CRAN

6.2 Cursos

- RStudio - online learning
- datacamp - learning R
- swirl - learn R, in R

6.3 Consulta

- R cookbook
- QuickR
- downloadable books o R and stats
- Use R!
- Official CRAN documentation
- r, stackoverflow

6.4 Manipulación y graficado de datos con paquetes especializados

- plotly and gglplot2 user guide
- Data wrangling with R and RStudio
- Data wrangling with R and RStudio - cheatsheet
- Data wrangling with R and RStudio - webinar
- Bradley C Boehmke - Data wrangling with R

Referencias

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, and Winston Chang. 2018. *Rmarkdown: Dynamic Documents for R*. https://CRAN.R-project.org/package=rmarkdown.

Boehmke, Bradley. 2016. *Data Wrangling with R*. 1st ed. Springer. http://www.springer.com/la/book/9783319455983.

Champely, Stephane. 2018. *Pwr: Basic Functions for Power Analysis*. https://CRAN.R-project.org/package=pwr.

Cotton, Richard. 2013. *Learning R*. 1st ed. O'Reilly.

Crawley, Michael J. 2012. *The R book*. 2nd ed. Wiley.

———. 2015. *Statistics : an introduction using R*. 2nd ed. Wiley.

Everitt, Bryan, and Torsten Hothorn. 2014. *A handbook of statistical analyses using R*. 3rd ed. Boca Raton: CRC Press. https://cran.r-project.org/web/packages/HSAUR/vignettes/Ch{_}introduction{_}to{_}R.pdf.

Field, Andy P., Jeremy Miles, and Zoe. Field. 2012. *Discovering statistics using R*. 1st ed. London: Sage.

Fox, John, Sanford Weisberg, and Brad Price. 2018. *Car: Companion to Applied Regression*. https://CRAN.R-project.org/package=car.

Gandrud, Christopher. 2015. *Reproducible research with R and RStudio*. 2nd ed. Chapman & Hall. https://github.com/christophergandrud/Rep-Res-Book.

Grolemund, Garrett. 2014. *Hands-on programming with R*. O'Reilly.

Kabacoff, Robert. 2015. *R in action : data analysis and graphics with R*. 2nd ed. Manning. https://github.com/kabacoff/RiA2 http://www.statmethods.net/.

Lander, Jared P. 2014. *R for everyone : advanced analytics and graphics*. New York, N.Y.: Addison-Wesley.

MacFerlane, John. 2016. "Pandoc - a universal document converter." http://pandoc.org/.

Marchetti, Giovanni M., Mathias Drton, and Kayvan Sadeghi. 2015. *Ggm: Functions for Graphical Markov Models*. https://CRAN.R-project.org/package=ggm.

Matloff, Norman S. 2011. *The art of R programming : tour of statistical software design*. No Starch Press.

http://heather.cs.ucdavis.edu/{~}matloff/132/NSPpart.pdf.

Murrell, Paul. 2009. “R Graphics.” In *Wiley Interdisciplinary Reviews: Computational Statistics*, 1:216–20. doi:10.1002/wics.22.

R Core Team. 2018. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Revelle, William. 2018. *Psych: Procedures for Psychological, Psychometric, and Personality Research*. https://CRAN.R-project.org/package=psych.

RStudio Team. 2016. *RStudio: Integrated Development Environment for R*. Boston, MA: RStudio, Inc. http://www.rstudio.com/.

Teetor, Paul, and Michael Kosta. Loukides. 2011. *R cookbook*. O’Reilly. http://www.cookbook-r.com/.

Wei, Taiyun, and Viliam Simko. 2017. *Corrplot: Visualization of a Correlation Matrix*. https://CRAN.R-project.org/package=corrplot.

Wickham, Hadley. 2016. *Ggplot2*. 2nd ed. Use R! Cham: Springer International Publishing. doi:10.1007/978-3-319-24277-4.

Xie, Yihui. 2018. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://CRAN.R-project.org/package=knitr.