

Ejercicio 1: crear la matriz transpuesta usando bucles for #####

Inicializa debajo la matriz aleatoria A 5x5 usando `runif(25)` para rellenarla y una matriz B cuyos componentes sean 0. También puedes asignar variables al número de filas y columnas para usarlas más adelante.

Sentencia dos bucles for: el primero que itere desde 1 hasta el número de filas de A y el otro que itere desde 1 hasta el número de columnas para barrer toda la matriz. Dentro de los bucles, piensa que `A[2,1] <- B[1,2]` o que `A[5,2] <- B [2,5]`
CUIDADO CON LAS LLAVES ({})

Comprueba el resultado de tu programa, B y la función optimizada de R para obtener la inversa: `t(matriz)` utilizando operadores lógicos (`==`)

Ejercicio 2: Función para encontrar el orden de reacciones químicas ##### conociendo los valores de concentración frente a tiempo. Para ello crearemos una función que compare el R cuadrado de concentración frente a tiempo (orden 0) $\ln(\text{concentración})$ frente a tiempo (orden 1)...

Inicializa la función `OrdenReaccion` usando el comando `function()` y, dentro de los parentesis, incluye las variables que va a utilizar la función (un vector de concentración y otro de tiempo). ABRE LA LLAVE Y NO CIERRES HASTA TERMINAR EL PROGRAMA MUY ABAJO.

Inicializa las variables que van a ser rellenadas en el programa:

1. Una matriz concentración de 0s que va a ser rellenada con diferentes cálculos al vector concentración. En la primera fila estará el vector concentración, en la segunda, el vector resultante al aplicar el logaritmo neperiano a todo el vector... Como se van a considerar 10 posibles órdenes de reacción (nunca fraccionarios), la matriz tendrá 10 filas y tantas columnas como elementos tenga el vector concentración.

2. Un vector r cuadrado que va a ser rellenado con la correlación de cada fila de la matriz con respecto al tiempo. Tiene 10 0s, uno para cada posible orden. Puedes usar bucles o la función `rep(x,y)` que repite el valor x en un vector de y componentes

Asigna el vector concentración a la primera fila de la matriz utilizando `MatrizConcentracion [1,]`. Asigna también el vector resultante de aplicar el logaritmo neperiano al vector concentración a la segunda fila. Recuerda que al aplicar funciones como `log()` o `sqrt()` a un vector, se aplica a cada elemento.

Para las 8 filas siguientes, crea un bucle que itere desde 2 hasta 10. En el interior, aplica la formula provista en el recurso. CUIDADO CON LAS LLAVES

Rellena la primera componente del vector R cuadrado empleando para ello la funcion $\text{cor}(\text{vector1}, \text{vector2})^2$ teniendo en cuenta que un vector es el tiempo y el otro es la primera fila de la matriz (`MatrizConcentracion[1,]`)

Con un bucle, itera de 2 a 10 y rellena el resto de componentes, teniendo en cuenta que el elemento i del vector R cuadrado es aplicar la funcion `cor()` arriba descrita al vector `MatrizConcentracion[i,]`. CUIDADO CON LAS LLAVES.

Una vez has relleno el vector R cuadrado, tan solo queda saber donde se encuentra el elemento maximo para hallar el orden. Como R empieza con el subindice 1 para lo que denominamos orden 0, la posicion del elemento maximo (le puedes llamar `ind`), es `Orden + 1`. (Luego: `Orden <- ind - 1`). Se presentan 2 formas de obtenerlo

1. Con un bucle. Para ello, inicializa una variable `max` a 0 (tambien tendras que inicializar `Orden` e `ind`) e itera desde 1 hasta 10 con un bucle `for`. En ese bucle, comparas (implica condicional `if{}`) cada elemento del vector R cuadrado con la variable `max`, y la condicion es que el elemento sea mayor que `max`. En caso de que la condicion se cumpla, reasigna la variable `max` a su nuevo valor y almacena su posicion (osea el valor que tome la variable del bucle `for`, que normalmente es `i`) en la variable `ind` y, restandole 1 obtienes el orden. LLAVES!!!

2. Con la funcion optimizada `which.max()`: te devuelve directamente la posicion del elemento maximo de un vector. Asignalo a `ind` y obten el orden.

Una vez tienes el orden, asigna a una variable `Reg` la regresion lineal del vector tiempo con la fila de la matriz concentracion que ha obtenido mayor R cuadrado. Piensa que ya has almacenado esa fila! La tienes en `MatrizConcentracion[ind,]`. La regresion se obtiene utilizando la funcion `lm(formula = y ~ x)`, donde `x` es el tiempo; (si no encuentras el simbolo copia pega). `Reg <- lm(...)`

Con la funcion `lm()` se puede acceder a los parametros obtenidos en la regresion, que en este caso equivalen a las constantes cineticas. Para ello, la ordenada en el origen se obtiene con `Reg$coefficients [1]` y la constante cinetica es el valor absoluto de `Reg$coefficients [2]`. Asigna una variable a cada uno (`k` y `Ordenada`)

Ya esta! Ahora solo tienes que presentar los resultados. Crea un vector `solucion` que contenga `Orden`, `k`, `Ordenada`, y el valor R cuadrado de la regresion (equivale al elemento maximo del vector R cuadrado, almacenado en la posicion `ind`). Puedes

crear un data frame incluyendo un vector con los nombres o simplemente usar la funcion: `names(Solucion) <- c("Orden", "K", "Ordenada", "R cuadrado")`. Como curiosidad, tambien puedes obtener el R cuadrado de una regresion con: `summary(Reg)$r.squared`

Por ultimo, crea una grafica usando `plot(x,y)` siendo x el tiempo e y la fila de la matriz concentracion que ha obtenido mayor R cuadrado. Ademas, incluye la grafica la recta de regresion escribiendo debajo del primer plot `abline(Reg)`

Para que el programa muestre nuestro resultado escribe `return(solucion)`. Despues termina tu programa cerrando la primera llave de todas. REPASA PARENTESIS, CORCHETES, LLAVES Y BUSCA ERRORES AL CORRER EL PROGRAMA QUE INDIQUEN FALTAS DE ORTOGRAFIA.

Comprueba tu programa con estos datos:

```
t <- c(0, 25, 50, 75, 100, 200, 250)
```

```
Con <- c(18.5, 15.7, 13.3, 11.2, 9.48, 4.93, 3.54)
```

Corre el programa para estos datos utilizando: `OrdenReaccion(t,Con)` pero ten cuidado a la hora de seguir el orden que hayas puesto al inicializar la funcion, es decir: si has puesto `OrdenReaccion <- function (tiempo, concentracion)`, el programa no funcionara si escribes `OrdenReaccion(Con,t)` por no respetar el orden.

Deberias obtener estos valores:

Orden	K	Ordenada	R cuadrado
1.000000000	0.006613361	2.916158882	0.999968439