# Algebra lineal R4

Juan Luis Acebal Rico

Pregunta 0. Los resultados del informe corresponden al segundo intento.

1. [10 %]

```
AP <- 0.45

fila_E <- c(0.1,0.5,0,0,0,0.4)
fila_V <- c(0,3*(1-AP)/5,AP,0,0,2*(1-AP)/5)
fila_A <- c(0,0,0.2,0.6,0,0.2)
fila_P <- c(0,0,0,0.1,0.8,0.1)
fila_F <- c(0,0,0,0,1,0)
fila_S <- c(0,0,0,0,0,1)

valores <- c(
  fila_E,
  fila_V,
  fila_A,
  fila_P,
  fila_F,
  fila_S
)
labels <- c("E", "V", "A", "P", "F", "S")
byrow=TRUE
P <- matrix(data = valores, byrow = byrow , nrow = 6, dimnames = list(labels, labels))


suma_filas <- rowSums(P)
suma_columnas <- colSums(P)

suma_filas
```

```
## E V A P F S
## 1 1 1 1 1 1
```

```
suma_columnas
```

```
##    E    V    A    P    F    S
## 0.10 0.83 0.65 0.70 1.80 1.92
```

2. [10 %]

```r
estados_absorbentes <- which(diag(P) == 1 & rowSums(P) == 1)

estados_absorbentes
```

```
## F S
## 5 6
```

3. [10 %]

```r
matriz_2_pasos <- P %*% P
probabilidad_F_2 <- matriz_2_pasos["E", "F"]

matriz_3_pasos <- matriz_2_pasos %*% P
probabilidad_F_3 <- matriz_3_pasos["E", "F"]

probabilidad_F_2
```

```
## [1] 0
```

```r
probabilidad_F_3
```

```
## [1] 0
```

4. [20 %]

```r
set.seed(123)
simulate_path <- function(P, steps) {
current_state <- 1
path <- numeric(steps)
for (i in 1: steps) {
path [i] <- current_state
current_state <- sample(1:nrow(P), size=1 , prob=P[current_state, ])
}
return (path)
}
clientes <- 100
pasos <- 10
results <- replicate(clientes, simulate_path(P, pasos))
porcentaje_finalizacion <- sum(apply(results, 2, function(x) any(x==5)))/clientes

porcentaje_finalizacion
```

```
## [1] 0.2
```

5. [10 %]

```r
valores_propios <- eigen(t(P))$values
vectores_propios <- eigen(t(P))$vectors
```

```r
pos_1 <- which(abs(valores_propios - 1) < 1e-8)
estacionarios_list <- list()

for (i in pos_1) {
  autovector <- vectores_propios[, i]
  autovector <- Re(autovector)
  autovector_normalizado <- autovector / sum(autovector)
  estacionarios_list[[ length(estacionarios_list) + 1 ]] <- autovector_normalizado
}

cat("{")
```

```
## {
```

```r
for (j in seq_along(estacionarios_list)) {
  v <- estacionarios_list[[j]]
  cat("[", paste(v, collapse=","), "]", sep="")
  if (j < length(estacionarios_list)) cat(",")
}
```

```
## [0,0,0,0,0,1],[0,0,0,0,0,1,0]
```

```r
cat("}\n")
```

```
## }
```

6. [10 %]

```r
matriz_canonica <- P[c(estados_absorbentes, setdiff(1:6, estados_absorbentes)),
                                    c(estados_absorbentes, setdiff(1:6, estados_absorbentes))]
matriz_canonica
```

```
##     F    S   E    V    A   P
## F 1.0 0.00 0.0 0.00 0.00 0.0
## S 0.0 1.00 0.0 0.00 0.00 0.0
## E 0.0 0.40 0.1 0.50 0.00 0.0
## V 0.0 0.22 0.0 0.33 0.45 0.0
## A 0.0 0.20 0.0 0.00 0.20 0.6
## P 0.8 0.10 0.0 0.00 0.00 0.1
```

Pregunta 7. [20 %]

```r
k <- length(estados_absorbentes)
D <- matriz_canonica[(k + 1):6, (k + 1):6]
A <- matriz_canonica[(k + 1):6, 1:k]
I_identidad <- diag(nrow(D))

matriz_Sr <- solve(I_identidad - D) %*% A
matriz_Sr
```

```
##           F         S
## E 0.2487562 0.7512438
## V 0.4477612 0.5522388
## A 0.6666667 0.3333333
## P 0.8888889 0.1111111
```

8. [10 %]

```
prop_compra <- matriz_Sr["E", "F"]  # Columna F
prop_no_compra <- matriz_Sr["E", "S"]  # Columna S
prop_compra
```

```
## [1] 0.2487562
```

```
prop_no_compra
```

```
## [1] 0.7512438
```