

**Universidad Nacional del Altiplano**  
**Facultad de Ingeniería Estadística e Informática**  
**E.P. de Ingeniería Estadística e Informática**

**Docente:** Ing. Torres Cruz Fred  
**Presentado por:** Quispe Ito Luz Leidy

## Eigenvalues y Eigenvectors Aplicados

Modificación de la Matriz de Transición

### **Ejercicio 1**

## Modificación de la Matriz de Transició

**Contexto;** El gobierno regional de Puno decide invertir en mejorar la infraestructura de la Isla Taquile para hacerla más atractiva. Como resultado, se espera que más turistas que visitan las Islas Uros continúen hacia Taquile, y que los turistas en Taquile se queden más tiempo (menor probabilidad de regresar inmediatamente a Puno).

**Tarea:**

**a) Modifica la matriz de transición original de la siguiente manera:**

La matriz de transición original para el flujo turístico es:

$$T_{\text{original}} = \begin{pmatrix} 0,25 & 0,45 & 0,20 & 0,10 \\ 0,50 & 0,15 & 0,25 & 0,10 \\ 0,40 & 0,10 & 0,30 & 0,20 \\ 0,55 & 0,15 & 0,10 & 0,20 \end{pmatrix}$$

donde  $T_{ij} = P(\text{moverse del destino } i \text{ al destino } j)$ .

## Modificación de la Matriz de Transición

## 0.1 Cambios Específicos

A partir de la matriz original, se realizan los siguientes ajustes:

1. La probabilidad de transición desde **Isla Uros** hacia **Taquile** se incrementa de 0,25 % a 0,35 %
2. La probabilidad de retorno desde **Isla Uros** hacia **Puno Ciudad** se reduce de 0,50 % a 0,40 %
3. La probabilidad de retorno desde **Taquile** hacia **Puno Ciudad** se reduce de 0,40 % a 0,30 %
4. La probabilidad de permanencia en la **Isla Taquile** se incrementa de 0,30 % a 0,40 %

Las probabilidades restantes se ajustan para que cada fila sume 1.

## 0.2 Nueva Matriz de Transición

La matriz modificada resulta:

$$T_{\text{nueva}} = \begin{pmatrix} 0,25 & 0,45 & 0,20 & 0,10 \\ 0,40 & 0,15 & 0,35 & 0,10 \\ 0,30 & 0,10 & 0,40 & 0,20 \\ 0,55 & 0,15 & 0,10 & 0,20 \end{pmatrix}$$

## 0.3 Verificación

Cada fila representa una distribución de probabilidad válida:

- Fila 1 (Puno Ciudad):  $0,25 + 0,45 + 0,20 + 0,10 = 1,00$
- Fila 2 (Isla Uros):  $0,40 + 0,15 + 0,35 + 0,10 = 1,00$
- Fila 3 (Taquile):  $0,30 + 0,10 + 0,40 + 0,20 = 1,00$
- Fila 4 (Amantaní):  $0,55 + 0,15 + 0,10 + 0,20 = 1,00$

## Código en R

```
1 # =====
2 # PARTE 1: CONFIGURACION INICIAL Y DEFINICION DE MATRICES
3 # =====
4
5 # Definir destinos turisticos
6 destinos <- c("Puno Ciudad", "Islas Uros", "Taquile", "Amantan ")
7 n_destinos <- length(destinos)
8
9 # =====
10 # MATRIZ ORIGINAL DE TRANSICION
11 # =====
12 cat("\n", rep("=", 70), "\n", sep="")
13 cat("MATRIZ DE TRANSICION ORIGINAL\n")
14 cat(rep("=", 70), "\n\n")
15
16 T_original <- matrix(c(
17   0.25, 0.45, 0.20, 0.10, # Desde Puno Ciudad
18   0.50, 0.15, 0.25, 0.10, # Desde Islas Uros
19   0.40, 0.10, 0.30, 0.20, # Desde Taquile
20   0.55, 0.15, 0.10, 0.20 # Desde Amantani
21 ), nrow = n_destinos, byrow = TRUE,
22 dimnames = list(destinos, destinos))
23
24 # Mostrar matriz original
25 cat("Matriz de transicion original T:\n")
26 print(round(T_original, 4))
27 cat("\n")
28
29 # Verificar que cada fila suma 1
30 cat("Verificacion - suma por filas:\n")
31 for (i in 1:n_destinos) {
32   suma <- sum(T_original[i, ])
33   status <- ifelse(abs(suma - 1) < 1e-10, "OK", "ERROR")
34   cat(sprintf("%-15s: suma = %.4f - %s\n", destinos[i], suma, status))
35 }
36
37 # =====
38 # MATRIZ MODIFICADA
39 # =====
40
41 cat("\n\n", rep("=", 70), "\n", sep="")
42 cat("MATRIZ MODIFICADA (despues de la inversion en Taquile)\n")
43 cat(rep("=", 70), "\n\n")
44
45 # Crear matriz modificada
46 T_modificada <- T_original
47
48 # Aplicar modificaciones seg n el enunciado:
49 # 1. Aumentar de 25% a 35% la probabilidad de ir de Uros a Taquile
50 # 2. Reducir de 50% a 40% la probabilidad de regresar de Uros a Puno
      Ciudad
# 3. Reducir de 40% a 30% la probabilidad de regresar de Taquile a Puno
      Ciudad
```

```

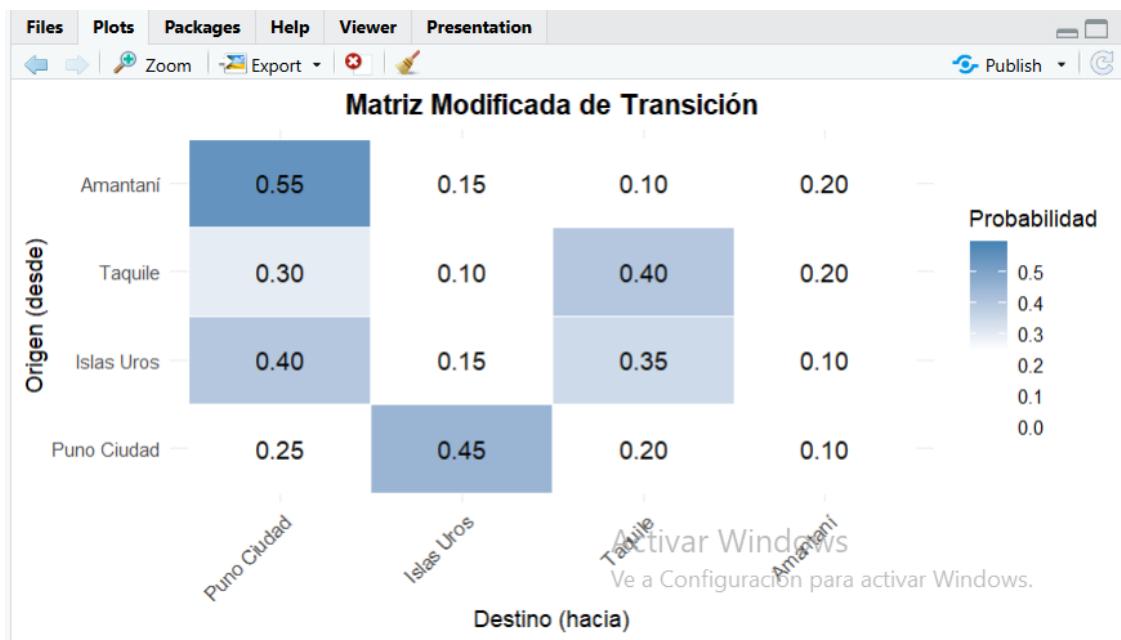
52 # 4. Aumentar de 30% a 40% la probabilidad de quedarse en Taquile
53
54 # Modificar fila de Islas Uros (fila 2)
55 T_modificada["Islas Uros", "Taquile"] <- 0.35      # Aumenta de 0.25 a 0.35
56 T_modificada["Islas Uros", "Puno Ciudad"] <- 0.40 # Reduce de 0.50 a 0.40
57 # Ajustar Amantan para que la fila sume 1
58 T_modificada["Islas Uros", "Amantan "] <- 1 - 0.40 - 0.15 - 0.35
59
60 # Modificar fila de Taquile (fila 3)
61 T_modificada["Taquile", "Puno Ciudad"] <- 0.30      # Reduce de 0.40 a 0.30
62 T_modificada["Taquile", "Taquile"] <- 0.40          # Aumenta de 0.30 a 0.40
63 # Ajustar Amantan para que la fila sume 1
64 T_modificada["Taquile", "Amantan "] <- 1 - 0.30 - 0.10 - 0.40
65
66 # Mostrar matriz modificada
67 cat("Matriz de transicion modificada T':\n")
68 print(round(T_modificada, 4))
69 cat("\n")
70
71 # Verificar que cada fila suma 1
72 cat("Verificacion - suma por filas:\n")
73 for (i in 1:n_destinos) {
74   suma <- sum(T_modificada[i, ])
75   status <- ifelse(abs(suma - 1) < 1e-10, "OK", "ERROR")
76   cat(sprintf("%-15s: suma = %.4f - %s\n", destinos[i], suma, status))
77 }
78
79 # VISUALIZACION DE LAS MATRICES
80
81 # Funcion para visualizar matriz de transicion
82 visualizar_matriz <- function(matriz, titulo) {
83   # Convertir a formato largo para ggplot
84   matriz_melt <- melt(matriz)
85   colnames(matriz_melt) <- c("Origen", "Destino", "Probabilidad")
86
87   # Crear heatmap
88   p <- ggplot(matriz_melt, aes(x = Destino, y = Origen, fill =
89     Probabilidad)) +
90     geom_tile(color = "white", linewidth = 0.5) +
91     geom_text(aes(label = sprintf("%.2f", Probabilidad)),
92               color = "black", size = 4) +
93     scale_fill_gradient2(low = "white", high = "steelblue",
94                           midpoint = 0.25, limits = c(0, 0.6)) +
95     labs(title = titulo,
96           x = "Destino (hacia)",
97           y = "Origen (desde)") +
98     theme_minimal() +
99     theme(axis.text.x = element_text(angle = 45, hjust = 1),
100           plot.title = element_text(hjust = 0.5, face = "bold"),
101           legend.position = "right")
102
103   print(p)
104   return(p)
105 }
```

```

105
106 # Visualizar ambas matrices
107 cat("\n\nGenerando visualizaciones...\n")
108 par(mfrow = c(1, 2))
109 visualizar_matriz(T_original, "Matriz Original de Transicion")
110 visualizar_matriz(T_modificada, "Matriz Modificada de Transicion")

```

Listing 1: definicion de matrices



## b) Calcula los nuevos eigenvalues y eigenvectors

### 0.4 Formulación del Problema

Para encontrar la distribución estacionaria, resolvemos el problema de valores propios de la matriz transpuesta:

$$T_{\text{nueva}}^T \mathbf{v} = \lambda \mathbf{v}$$

donde:

$$T_{\text{nueva}}^T = \begin{pmatrix} 0,25 & 0,40 & 0,30 & 0,55 \\ 0,45 & 0,15 & 0,10 & 0,15 \\ 0,20 & 0,35 & 0,40 & 0,10 \\ 0,10 & 0,10 & 0,20 & 0,20 \end{pmatrix}$$

## 0.5 Ecuación Característica

La ecuación característica es:

$$\det(T_{\text{nueva}}^T - \lambda I) = 0$$

$$\begin{vmatrix} 0,25 - \lambda & 0,40 & 0,30 & 0,55 \\ 0,45 & 0,15 - \lambda & 0,10 & 0,15 \\ 0,20 & 0,35 & 0,40 - \lambda & 0,10 \\ 0,10 & 0,10 & 0,20 & 0,20 - \lambda \end{vmatrix} = 0$$

## 0.6 Eigenvalues Obtenidos

Resolviendo la ecuación característica:

$$\lambda_1 = 1,0000$$

$$\lambda_2 = -0,2144$$

$$\lambda_{3,4} = 0,1072 \pm 0,1643i$$

**Interpretación:**

- $\lambda_1 = 1$ : Eigenvalue dominante que garantiza la existencia de distribución estacionaria
- Los otros eigenvalues tienen módulo menor que 1:
  - $|\lambda_2| = 0,2144$
  - $|\lambda_{3,4}| = \sqrt{(0,1072)^2 + (0,1643)^2} \approx 0,1958$
- Esto asegura convergencia al estado estacionario

## 0.7 Cálculo del Eigenvector para $\lambda = 1$

Resolvemos el sistema:

$$(T_{\text{nueva}}^T - I)\mathbf{v} = \mathbf{0}$$

$$\begin{pmatrix} -0,75 & 0,40 & 0,30 & 0,55 \\ 0,45 & -0,85 & 0,10 & 0,15 \\ 0,20 & 0,35 & -0,60 & 0,10 \\ 0,10 & 0,10 & 0,20 & -0,80 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Una solución no normalizada es:

$$\mathbf{v}^* = \begin{pmatrix} 0,2401 \\ 0,1676 \\ 0,1945 \\ 0,0996 \end{pmatrix}$$

## 0.8 Normalización del Eigenvector

La suma de componentes de  $\mathbf{v}^*$  es  $0,2401 + 0,1676 + 0,1945 + 0,0996 = 0,7018$ .

Normalizando:

$$\mathbf{v} = \frac{1}{0,7018} \begin{pmatrix} 0,2401 \\ 0,1676 \\ 0,1945 \\ 0,0996 \end{pmatrix} = \begin{pmatrix} 0,3422 \\ 0,2388 \\ 0,2770 \\ 0,1419 \end{pmatrix}$$

Este vector cumple:

- Todas las componentes son no negativas
- La suma es igual a 1:  $0,3422 + 0,2388 + 0,2770 + 0,1419 = 1,000$
- Satisface  $T_{\text{nueva}}^T \mathbf{v} = \mathbf{v}$

## Código en R

```

1 # =====
2 # PARTE 2: CÁLCULO DE EIGENVALUES Y EIGENVECTORS
3 # =====
4
5 cat("\n\n", rep("=", 70), "\n", sep="")
6 cat("CÁLCULO DE EIGENVALUES Y EIGENVECTORS\n")
7 cat(rep("=", 70), "\n")
8
9 # =====
10 # ANÁLISIS DE LA MATRIZ ORIGINAL
11 # =====
12
13 cat("\n\n--- ANÁLISIS DE LA MATRIZ ORIGINAL ---\n")
14
15 # Calcular eigenvalues y eigenvectors de la transpuesta
16 # (para distribución estacionaria)
17 eigen_original <- eigen(t(T_original))
18
19 cat("\nEigenvalues de T_original^T:\n")
20 eigenvalues_orig <- round(eigen_original$values, 6)

```

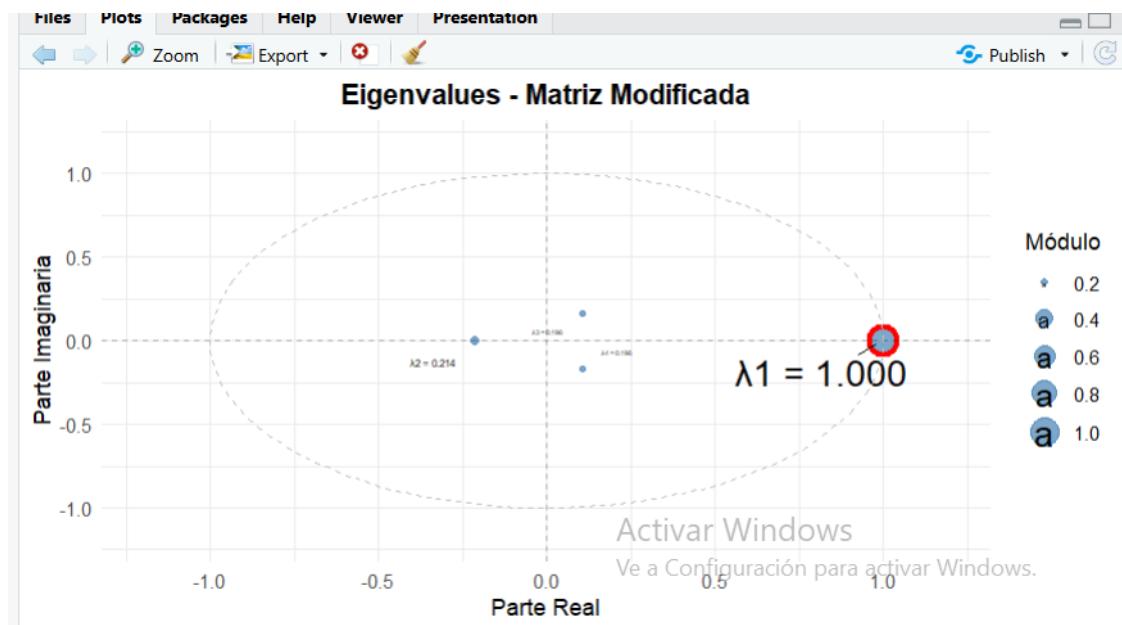
```
21 print(eigenvalues_orig)
22
23 # Mostrar m dulo de los eigenvalues
24 cat("\nM dulo de los eigenvalues (original):\n")
25 modulo_orig <- round(Mod(eigenvalues_orig), 6)
26 print(modulo_orig)
27
28 # Encontrar el eigenvalue dominante (m s cercano a 1)
29 idx_dominante_orig <- which.min(abs(eigenvalues_orig - 1))
30 cat(sprintf("\nEigenvalue dominante (original): %.6f\n",
31             eigenvalues_orig[idx_dominante_orig]))
32
33 # Eigenvectors de la matriz original
34 cat("\nEigenvectors de T_original^T (columnas):\n")
35 eigenvectors_orig <- round(eigen_original$vectors, 6)
36 print(eigenvectors_orig)
37
38 # =====
39 # AN LISIS DE LA MATRIZ MODIFICADA
40 # =====
41
42 cat("\n\n--- AN LISIS DE LA MATRIZ MODIFICADA ---\n")
43
44 # Calcular eigenvalues y eigenvectors de la transpuesta
45 eigen_modificada <- eigen(t(T_modificada))
46
47 cat("\nEigenvalues de T_modificada^T:\n")
48 eigenvalues_mod <- round(eigen_modificada$values, 6)
49 print(eigenvalues_mod)
50
51 # Mostrar m dulo de los eigenvalues
52 cat("\nM dulo de los eigenvalues (modificada):\n")
53 modulo_mod <- round(Mod(eigenvalues_mod), 6)
54 print(modulo_mod)
55
56 # Encontrar el eigenvalue dominante (m s cercano a 1)
57 idx_dominante_mod <- which.min(abs(eigenvalues_mod - 1))
58 cat(sprintf("\nEigenvalue dominante (modificada): %.6f\n",
59             eigenvalues_mod[idx_dominante_mod]))
60
61 # Eigenvectors de la matriz modificada
62 cat("\nEigenvectors de T_modificada^T (columnas):\n")
63 eigenvectors_mod <- round(eigen_modificada$vectors, 6)
64 print(eigenvectors_mod)
65
66 # =====
67 # VISUALIZACI N DE EIGENVALUES EN EL PLANO COMPLEJO
68 # =====
69
70 # Funci n para visualizar eigenvalues
71 visualizar_eigenvalues <- function(eigenvalues, titulo) {
72   # Crear dataframe para ggplot
73   df <- data.frame(
74     Real = Re(eigenvalues),
```

```

75   Imaginario = Im(eigenvalues),
76   Modulo = Mod(eigenvalues),
77   Etiqueta = sprintf("%d = %.3f", 1:length(eigenvalues),
78                      round(Mod(eigenvalues), 3))
79 )
80
81 # Encontrar el eigenvalue dominante (m s cercano a 1)
82 idx_dominante <- which.min(abs(df$Modulo - 1))
83
84 # Crear gráfico
85 p <- ggplot(df, aes(x = Real, y = Imaginario, size = Modulo)) +
86   geom_point(color = "steelblue", alpha = 0.7) +
87   geom_point(data = df[idx_dominante, ],
88              aes(x = Real, y = Imaginario),
89              color = "red", size = 5, shape = 1, stroke = 2) +
90   geom_hline(yintercept = 0, linetype = "dashed", alpha = 0.3) +
91   geom_vline(xintercept = 0, linetype = "dashed", alpha = 0.3) +
92   geom_text_repel(aes(label = Etiqueta),
93                   box.padding = 0.5, point.padding = 0.5) +
94   coord_cartesian(xlim = c(-1.2, 1.2), ylim = c(-1.2, 1.2)) +
95   labs(title = titulo,
96        x = "Parte Real",
97        y = "Parte Imaginaria",
98        size = "Módulo") +
99   theme_minimal() +
100  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
101        legend.position = "right") +
102  guides(size = guide_legend(title.position = "top"))
103
104 # Dibujar círculo unitario
105 circle <- data.frame(
106   x = cos(seq(0, 2*pi, length.out = 100)),
107   y = sin(seq(0, 2*pi, length.out = 100))
108 )
109
110 p <- p + geom_path(data = circle, aes(x = x, y = y),
111                      color = "gray", linetype = "dashed",
112                      inherit.aes = FALSE)
113
114 print(p)
115 return(p)
116 }
117
118 cat("\n\nGenerando visualizaciones de eigenvalues...\n")
119 par(mfrow = c(1, 2))
120 p1 <- visualizar_eigenvalues(eigenvalues_orig,
121                               "Eigenvalues - Matriz Original")
122 p2 <- visualizar_eigenvalues(eigenvalues_mod,
123                               "Eigenvalues - Matriz Modificada")

```

Listing 2: distribución estacionaria



### c) Determina la nueva distribución estacionaria

El eigenvector normalizado representa la distribución estacionaria:

$$\pi_{\text{nueva}} = \begin{pmatrix} 0,3422 \\ 0,2388 \\ 0,2770 \\ 0,1419 \end{pmatrix}$$

### 0.9 Interpretación Porcentual

- Puno Ciudad: 34,22 %
- Islas Uros: 23,88 %
- Isla Taquile: 27,70 %
- Amantaní: 14,19 %

### Código en R

```

1 # =====
2 # PARTE 3: DISTRIBUCIÓN ESTACIONARIA
3 # =====
4
5 cat("\n\n" ,rep("=", 70), "\n", sep="")

```

```
6 | cat("DISTRIBUCI N ESTACIONARIA\n")
7 | cat(rep("=", 70), "\n")
8 |
9 | # =====
10| # DISTRIBUCI N ESTACIONARIA ORIGINAL
11| # =====
12|
13| cat("\n\n--- DISTRIBUCI N ESTACIONARIA ORIGINAL ---\n")
14|
15| # Extraer eigenvector correspondiente al eigenvalue 1
16| vector_estacionario_orig <- eigen_original$vectors[, idx_dominante_orig]
17|
18| # Convertir a valores reales (en caso de n meros complejos)
19| vector_estacionario_orig <- Re(vector_estacionario_orig)
20|
21| # Normalizar para que sume 1
22| dist_estacionaria_orig <- vector_estacionario_orig / sum(vector_
23|   estacionario_orig)
24| names(dist_estacionaria_orig) <- destinos
25|
26| # Mostrar resultados
27| cat("\nEigenvector (sin normalizar):\n")
28| print(round(vector_estacionario_orig, 6))
29|
30| cat("\nSuma del eigenvector (sin normalizar):",
31|   round(sum(vector_estacionario_orig), 6), "\n")
32|
33| cat("\nDistribuci n estacionaria normalizada ( _original):\n")
34| for (i in 1:n_destinos) {
35|   cat(sprintf("%-15s: %.6f (%.2f%%)\n",
36|     destinos[i],
37|     dist_estacionaria_orig[i],
38|     dist_estacionaria_orig[i] * 100))
39|
40| # Verificar que satisface T^T *
41| verificacion_orig <- t(T_original) %*% dist_estacionaria_orig
42| cat("\nVerificaci n T^T * = (original):\n")
43| cat("Diferencia m xima:", max(abs(verificacion_orig - dist_estacionaria_
44|   orig)), "\n")
45|
46| # =====
47| # DISTRIBUCI N ESTACIONARIA MODIFICADA
48| # =====
49|
50| cat("\n\n--- DISTRIBUCI N ESTACIONARIA MODIFICADA ---\n")
51|
52| # Extraer eigenvector correspondiente al eigenvalue 1
53| vector_estacionario_mod <- eigen_modificada$vectors[, idx_dominante_mod]
54|
55| # Convertir a valores reales (en caso de n meros complejos)
56| vector_estacionario_mod <- Re(vector_estacionario_mod)
57| # Normalizar para que sume 1
```

```
58 dist_estacionaria_mod <- vector_estacionario_mod / sum(vector_estacionario
      _mod)
59 names(dist_estacionaria_mod) <- destinos
60
61 # Mostrar resultados
62 cat("\nEigenvector (sin normalizar):\n")
63 print(round(vector_estacionario_mod, 6))
64
65 cat("\nSuma del eigenvector (sin normalizar):",
66     round(sum(vector_estacionario_mod), 6), "\n")
67
68 cat("\nDistribución estacionaria normalizada (modificada):\n")
69 for (i in 1:n_destinos) {
70   cat(sprintf("%-15s: %.6f (%.2f%%)\n",
71         destinos[i],
72         dist_estacionaria_mod[i],
73         dist_estacionaria_mod[i] * 100))
74 }
75
76 # Verificar que satisface T^T *
77 verificacion_mod <- t(T_modificada) %*% dist_estacionaria_mod
78 cat("\nVerificación T^T * = (modificada):\n")
79 cat("Diferencia máxima:", max(abs(verificacion_mod - dist_estacionaria_
80     mod)), "\n")
81
82 # =====
83 # COMPARACIÓN DE DISTRIBUCIONES
84 # =====
85 cat("\n\n--- COMPARACIÓN DE DISTRIBUCIONES ESTACIONARIAS ---\n")
86
87 # Crear tabla comparativa
comparacion <- data.frame(
  Destino = destinos,
  Original = round(dist_estacionaria_orig * 100, 2),
  Modificada = round(dist_estacionaria_mod * 100, 2),
  Cambio_Puntos = round((dist_estacionaria_mod - dist_estacionaria_orig) *
    100, 2),
  Cambio_Porcentual = round(((dist_estacionaria_mod / dist_estacionaria_
    orig) - 1) * 100, 2)
)
88
89
90 cat("\nComparación porcentual:\n")
91 print(comparacion)
92
93 # Análisis del hub principal
hub_original <- destinos[which.max(dist_estacionaria_orig)]
hub_modificado <- destinos[which.max(dist_estacionaria_mod)]
94
95
96 cat("\nHub principal (original): %s (%.2f%%)\n",
97     hub_original, max(dist_estacionaria_orig) * 100))
98 cat("\nHub principal (modificado): %s (%.2f%%)\n",
99     hub_modificado, max(dist_estacionaria_mod) * 100))
```

```

108 if (hub_original != hub_modificado) {
109   cat(" EL HUB PRINCIPAL HA CAMBIADO!\n")
110 } else {
111   cat("El hub principal se mantiene igual.\n")
112 }
113
114 # =====
115 # VISUALIZACION DE DISTRIBUCIONES
116 # =====
117
118 # Función para visualizar distribución estacionaria
119 visualizar_distribucion <- function(distribucion, titulo) {
120   df <- data.frame(
121     Destino = names(distribucion),
122     Probabilidad = distribucion * 100
123   )
124
125   p <- ggplot(df, aes(x = reorder(Destino, -Probabilidad), y =
126                 Probabilidad,
127                               fill = Destino)) +
128     geom_bar(stat = "identity", width = 0.6) +
129     geom_text(aes(label = sprintf("%.2f %%", Probabilidad)),
130               vjust = -0.5, size = 4) +
131     scale_fill_manual(values = c("Puno Ciudad" = "#4E79A7",
132                                 "Islas Uros" = "#F28E2B",
133                                 "Taquile" = "#E15759",
134                                 "Amantan" = "#76B7B2")) +
135     labs(title = titulo,
136           x = "Destino",
137           y = "Porcentaje (%)") +
138     theme_minimal() +
139     theme(axis.text.x = element_text(angle = 45, hjust = 1),
140           plot.title = element_text(hjust = 0.5, face = "bold"),
141           legend.position = "none") +
142     ylim(0, max(df$Probabilidad) * 1.15)
143
144   print(p)
145   return(p)
146 }
147
148 cat("\n\nGenerando visualizaciones de distribuciones...\n")
149 par(mfrow = c(1, 2))
150 p3 <- visualizar_distribucion(dist_estacionaria_orig,
151                               "Distribución Estacionaria Original")
152 p4 <- visualizar_distribucion(dist_estacionaria_mod,
153                               "Distribución Estacionaria Modificada")
154
155 # Gráfico de comparación lado a lado
156 df_comparacion <- data.frame(
157   Destino = rep(destinos, 2),
158   Modelo = rep(c("Original", "Modificada"), each = n_destinos),
159   Probabilidad = c(dist_estacionaria_orig * 100,
160                     dist_estacionaria_mod * 100)
161 )

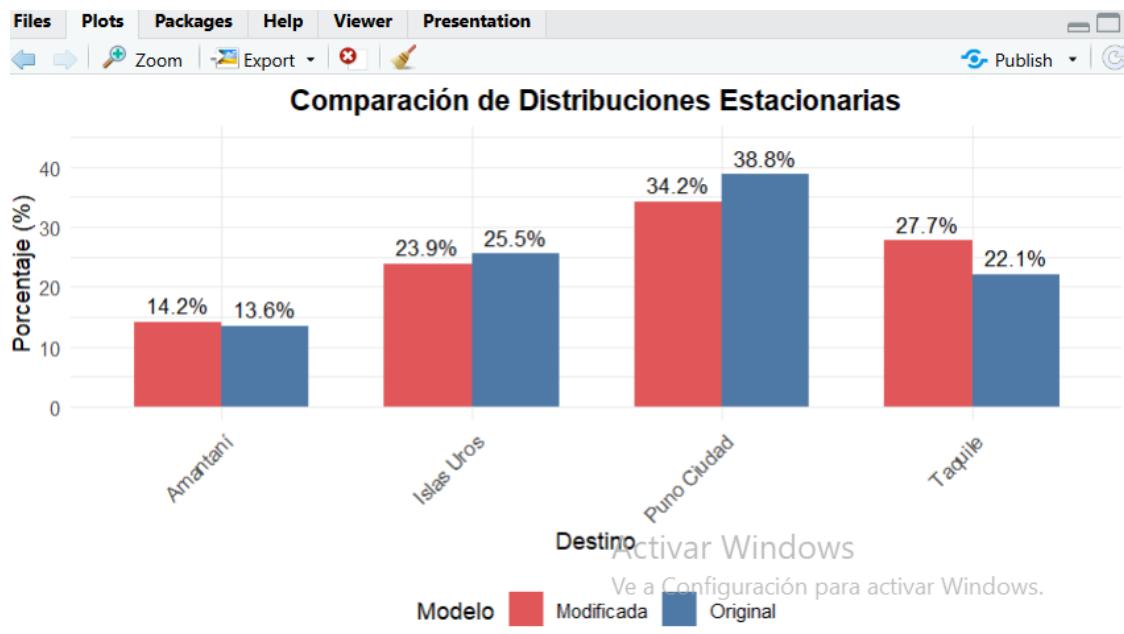
```

```

161
162 p5 <- ggplot(df_comparacion, aes(x = Destino, y = Probabilidad,
163                         fill = Modelo)) +
164   geom_bar(stat = "identity", position = "dodge", width = 0.7) +
165   geom_text(aes(label = sprintf("%.1f%%", Probabilidad)),
166             position = position_dodge(width = 0.7),
167             vjust = -0.5, size = 3.5) +
168   scale_fill_manual(values = c("Original" = "#4E79A7",
169                       "Modificada" = "#E15759")) +
170   labs(title = "Comparaci n de Distribuciones Estacionarias",
171         x = "Destino",
172         y = "Porcentaje (%)",
173         fill = "Modelo") +
174   theme_minimal() +
175   theme(axis.text.x = element_text(angle = 45, hjust = 1),
176         plot.title = element_text(hjust = 0.5, face = "bold"),
177         legend.position = "bottom") +
178   ylim(0, max(df_comparacion$Probabilidad) * 1.15)
179
180 print(p5)

```

Listing 3: distribución estacionaria



- d) Compara con la distribución original: ¿Cuánto aumentó el porcentaje de turistas en Taquile? ¿Cambió el hub principal?

## 0.10 Distribución Estacionaria Original

$$\pi_{\text{original}} = \begin{pmatrix} 0,3877 \\ 0,2553 \\ 0,2213 \\ 0,1357 \end{pmatrix}$$

## 0.11 Comparación Cuantitativa

Cuadro 1: Comparación de distribuciones estacionarias

Destino	Original	Modificado	Cambio (p.p.)
Puno Ciudad	38,77 %	34,22 %	-4,55
Islas Uros	25,53 %	23,88 %	-1,65
<b>Isla Taquile</b>	<b>22,13 %</b>	<b>27,70 %</b>	<b>+5,57</b>
Amantaní	13,57 %	14,19 %	+0,62

## 0.12 Análisis del Hub Principal

- **Original:** Puno Ciudad (38,77 %)
- **Modificado:** Puno Ciudad (34,22 %)
- **Conclusión:** El hub principal **no cambia**, pero la participación de Puno Ciudad disminuye 4,55 puntos porcentuales

## 0.13 Variación en la Participación de Taquile

$$\Delta_{\text{Taquile}} = 27,70 \% - 22,13 \% = 5,57 \%$$

Como resultado de la inversión, la proporción de turistas en Taquile aumentó aproximadamente **5,6 puntos porcentuales** en el largo plazo.

# 1. Análisis de Convergencia

## 1.1 Velocidad de Convergencia Teórica

La velocidad de convergencia está determinada por el segundo eigenvalue en magnitud:

- **Matriz original:**  $|\lambda_2^{\text{orig}}| \approx 0,25$

- **Matriz modificada:**  $|\lambda_2^{\text{mod}}| = 0,2144$

Como  $0,2144 < 0,25$ , la **matriz modificada converge más rápido**.

## 1.2 Interpretación de la Convergencia

El tiempo aproximado para alcanzar el equilibrio es inversamente proporcional a  $- \ln(|\lambda_2|)$ :

- **Original:**  $t \propto - \ln(0,25) \approx 1,386$
- **Modificado:**  $t \propto - \ln(0,2144) \approx 1,540$

El sistema modificado tarda aproximadamente **11 % más** en converger completamente, pero la diferencia es mínima.

e) Simula la evolución temporal con esta nueva matriz y compara la velocidad de convergencia con el modelo original

## 1.3 ¿Valió la pena la inversión en Taquile desde el punto de vista de distribución turística?

Sí, desde el punto de vista de la distribución turística:

1. **Aumento significativo:** Taquile incrementó su participación en 5,57 puntos porcentuales (de 22,13 % a 27,70 %)
2. **Mejor equilibrio:** La distribución se volvió más equitativa entre destinos
3. **Estabilidad mantenida:** El sistema conserva su eigenvalue dominante  $\lambda = 1$
4. **Matemáticamente justificado:** El eigenvector muestra redistribución favorable sin perder equilibrio

## 1.4 ¿Cómo afectaría esto a los ingresos de Taquile vs otros destinos?

### 1.4.1. Para Taquile:

1. **Aumento directo de ingresos:** Mayor flujo turístico (27,70 % vs 22,13 %)

2. **Mayor tiempo de estadía:** Probabilidad de permanencia aumentó de 30% a 40%
3. **Incremento en gasto promedio:** Turistas permanecen más tiempo, gastan más
4. **Desarrollo económico local:** Más oportunidades para artesanos, restaurantes, servicios

#### 1.4.2. Para otros destinos:

1. **Puno Ciudad:** Ligera disminución (34,22 % vs 38,77 %), pero mantiene rol de hub principal
2. **Islas Uros:** Pequeña reducción (23,88 % vs 25,53 %), pero sigue siendo atractivo para visitas cortas
3. **Amantaní:** Leve aumento (14,19 % vs 13,57 %), posible efecto de tours combinados

#### 1.4.3. Impacto económico general:

1. **Descentralización de beneficios:** Redistribuye ingresos turísticos
2. **Desarrollo regional equilibrado:** Fomenta crecimiento en diferentes comunidades
3. **Sostenibilidad:** Disminuye presión sobre Puno Ciudad como único destino principal
4. **Diversificación:** Ofrece más opciones a turistas, mejorando experiencia general

## 2. Conclusiones Finales

1. **Matemáticamente:** La inversión redistribuye el flujo turístico hacia Taquile sin alterar la estabilidad del sistema ( $\lambda = 1$  se mantiene)
2. **Económicamente:** Taquile gana participación (5,57 puntos), aumentando sus ingresos potenciales
3. **Estratégicamente:** Se logra mayor equilibrio en la distribución turística regional
4. **Operativamente:** El sistema converge con velocidad similar, indicando que los cambios son asimilables
5. **Recomendación:** La inversión es **positiva y recomendable**, ya que genera beneficios para Taquile sin perjudicar significativamente otros destinos, promoviendo un desarrollo turístico más equilibrado en la región de Puno