

CodigoFuenteProyecto

Salvador P

2023-11-15

```
library(markovchain)
```

```
## Package:  markovchain
## Version:   0.9.4
## Date:      2023-08-31 10:30:02 UTC
## BugReport: https://github.com/spedygiorgio/markovchain/issues
```

```
library(ggplot2)
```

##Se tomaron los datos del estado de clima de los meses de Enero y Noviembre del 2023 en Saltillo, Coahuila, siendo los datos los siguientes. 334 en total.

```
# Estados:
## 0: soleado
## 1: nublado
## 2: lluvioso
## 3: tormenta
datos = c(0,1,1,0,0,0,0,1,1,1,0,1,1,0,0,0,0,0,0,1,2,1,1,1,0,1,1,1,1,0,0,
1,1,0,0,1,0,1,1,0,0,0,2,1,0,1,0,0,1,1,0,1,1,1,1,1,0,1,1,
0,0,0,0,1,0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,0,1,0,0,0,1,1,1,1,1,
2,1,1,2,1,1,1,1,1,0,1,1,1,1,1,2,0,0,1,0,1,1,1,1,1,1,3,2,1,1,
1,1,0,1,1,2,1,2,1,2,0,3,2,2,2,2,1,3,3,1,1,1,2,2,2,1,1,3,2,3,2,
0,2,2,2,2,1,0,0,0,2,0,0,1,1,2,0,1,1,1,2,0,2,2,0,0,1,2,2,2,2,
2,1,2,2,2,2,2,2,2,0,2,2,0,2,2,3,0,2,2,2,2,2,1,1,1,2,2,3,1,0,
2,3,2,2,0,0,0,0,2,1,1,2,2,2,2,2,2,2,3,2,2,1,1,1,2,2,2,1,1,1,2,
2,3,2,1,1,1,1,1,0,1,1,2,1,2,1,2,1,1,1,1,2,2,0,2,2,1,1,2,0,2,
2,2,2,2,2,1,1,1,1,1,1,1,2,0,0,2,2,0,1,2,2,3,0,0,0,1,0,0,0,0,0,
2,0,0,1,0,0,0,0,0,1,1,1,1,1,1,0,0,1,0,0,0,0,0,0,0,2,1,1,2,1,0)
```

##Definición de la matriz de transición

$$P = \begin{bmatrix} P(\text{Soleado}|\text{Soleado}) & P(\text{Nublado}|\text{Soleado}) & P(\text{Lluvioso}|\text{Soleado}) & P(\text{Tormenta}|\text{Soleado}) \\ P(\text{Soleado}|\text{Nublado}) & P(\text{Nublado}|\text{Nublado}) & P(\text{Lluvioso}|\text{Nublado}) & P(\text{Tormenta}|\text{Nublado}) \\ P(\text{Soleado}|\text{Lluvioso}) & P(\text{Nublado}|\text{Lluvioso}) & P(\text{Lluvioso}|\text{Lluvioso}) & P(\text{Tormenta}|\text{Lluvioso}) \\ P(\text{Soleado}|\text{Tormenta}) & P(\text{Nublado}|\text{Tormenta}) & P(\text{Lluvioso}|\text{Tormenta}) & P(\text{Tormenta}|\text{Tormenta}) \end{bmatrix}$$

Donde $P(X/Y)$ representa la probabilidad de transición del estado Y al estado X.

Después de la recopilación de datos hacemos un conteo de los posibles cambios de estados partiendo de un día soleado.

```
##Se contarán las veces que siga soleado al día siguiente,  
##Las veces que de soleado cambió a nublado, de soleado a ##lluvioso y de soleado a tormenta.  
conteo0 = c(0, 0, 0, 0)  
for(i in c(11:334)){  
  if(datos[i-1]==0){  
    if(datos[i]==0) conteo0[1]=conteo0[1]+1  
    if(datos[i]==1) conteo0[2]=conteo0[2]+1  
    if(datos[i]==2) conteo0[3]=conteo0[3]+1  
    if(datos[i]==3) conteo0[4]=conteo0[4]+1  
  }  
}
```

Se hace el mismo conteo para los demás cambios de estados.

```
conteo1 = c(0, 0, 0, 0)  
for(i in c(11:334)){  
  if(datos[i-1]==1){  
    if(datos[i]==0) conteo1[1]=conteo1[1]+1  
    if(datos[i]==1) conteo1[2]=conteo1[2]+1  
    if(datos[i]==2) conteo1[3]=conteo1[3]+1  
    if(datos[i]==3) conteo1[4]=conteo1[4]+1  
  }  
}  
  
conteo2 = c(0, 0, 0, 0)  
for(i in c(11:334)){  
  if(datos[i-1]==2){  
    if(datos[i]==0) conteo2[1]=conteo2[1]+1  
    if(datos[i]==1) conteo2[2]=conteo2[2]+1  
    if(datos[i]==2) conteo2[3]=conteo2[3]+1  
    if(datos[i]==3) conteo2[4]=conteo2[4]+1  
  }  
}  
  
conteo3 = c(0, 0, 0, 0)  
for(i in c(11:334)){  
  if(datos[i-1]==3){  
    if(datos[i]==0) conteo3[1]=conteo3[1]+1  
    if(datos[i]==1) conteo3[2]=conteo3[2]+1  
    if(datos[i]==2) conteo3[3]=conteo3[3]+1  
    if(datos[i]==3) conteo3[4]=conteo3[4]+1  
  }  
}
```

##Despues normaliza los conteos de transición para cada estado climático. Línea por línea:

##n = sum(conteo0): Calcula la suma de los conteos de transición para el estado 0 (soleado). Es decir,

suma las veces que el estado 0 cambió a otros estados.

##conteo0[1]=conteo0[1]/n: Normaliza la frecuencia de transición del estado 0 al estado 0 (soleado a soleado) dividiendo el conteo de esta transición por el total de transiciones desde el estado 0.

##conteo0[2]=conteo0[2]/n: Hace lo mismo que el paso anterior pero para la transición del estado 0 al estado 1 (soleado a nublado).

##conteo0[3]=conteo0[3]/n: Normaliza la frecuencia de transición del estado 0 al estado 2 (soleado a lluvioso).

##conteo0[4]=conteo0[4]/n: Normaliza la frecuencia de transición del estado 0 al estado 3 (soleado a tormenta).

```
n = sum(conteo0)
conteo0[1]=conteo0[1]/n
conteo0[2]=conteo0[2]/n
conteo0[3]=conteo0[3]/n
conteo0[4]=conteo0[4]/n
```

```
n = sum(conteo1)
conteo1[1]=conteo1[1]/n
conteo1[2]=conteo1[2]/n
conteo1[3]=conteo1[3]/n
conteo1[4]=conteo1[4]/n
```

```
n = sum(conteo2)
conteo2[1]=conteo2[1]/n
conteo2[2]=conteo2[2]/n
conteo2[3]=conteo2[3]/n
conteo2[4]=conteo2[4]/n
```

```
n = sum(conteo3)
conteo3[1]=conteo3[1]/n
conteo3[2]=conteo3[2]/n
conteo3[3]=conteo3[3]/n
conteo3[4]=conteo3[4]/n
```

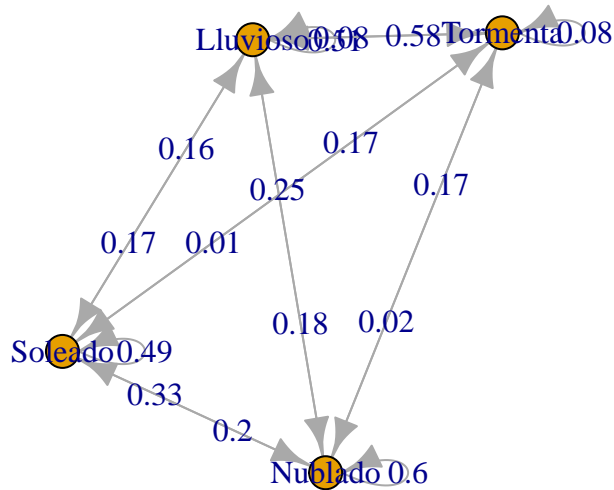
##Sacamos los porcentajes dada la siguiente formula: $p(x) = \text{conteos}/n$

```
statesNames = c("Soleado", "Nublado", "Lluvioso", "Tormenta")
mc_p = new("markovchain", transitionMatrix = matrix(c(conteo0, conteo1, conteo2, conteo3),byrow=TRUE,
nrow=4, dimnames=list(statesNames, statesNames)))
```

```
print(mc_p)
```

```
##           Soleado  Nublado  Lluvioso  Tormenta
## Soleado  0.4880952 0.3333333 0.1666667 0.01190476
## Nublado  0.1985294 0.6029412 0.1764706 0.02205882
## Lluvioso 0.1630435 0.2500000 0.5108696 0.07608696
## Tormenta 0.1666667 0.1666667 0.5833333 0.08333333
```

```
plot(mc_p)
```



#Esta matriz muestra las probabilidades de transición entre los estados climáticos Soleado, Nublado, Lluvioso y Tormenta. Cada fila representa el estado inicial y cada columna representa la probabilidad de pasar a otro estado.

$$P = \begin{bmatrix} 0.488 & 0.333 & 0.167 & 0.012 \\ 0.199 & 0.603 & 0.176 & 0.022 \\ 0.163 & 0.250 & 0.511 & 0.076 \\ 0.076 & 0.167 & 0.583 & 0.083 \end{bmatrix}$$

#Hemos creado una función para aleatoriamente con las probabilidades cargadas, se estima el clima del siguiente día.

```

estimarClima <- function(climaAnterior){
  aleatorio = runif(1)
  climaAnterior = climaAnterior+1
  if(aleatorio<sum(mc_p[climaAnterior][1:1])){
    return(0)
  }
  else if(aleatorio<sum(mc_p[climaAnterior][1:2])){
    return(1)
  }
  else if(aleatorio<sum(mc_p[climaAnterior][1:3])){
    return(2)
  }
  else if(aleatorio<sum(mc_p[climaAnterior][1:4])){
    return(3)
  }
}

```

```
}  
}
```

##Se supondra del estado que empieze, calcular aleatoriamente con las probabilidades de la cadena de markov, (Este no siempre será el mismo).

##Recuerda que:

```
## 0: Soleado  
## 1: Nublado  
## 2: Lluvioso  
## 3: Tormenta  
print(estimarClima(0))
```

```
## [1] 1  
print(estimarClima(1))
```

```
## [1] 0  
print(estimarClima(2))
```

```
## [1] 2  
print(estimarClima(3))
```

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

*# Crear una función para simular la cadena de Markov por un número
#específico de días usando temporalidad.*

```
simularClima <- function(dias, climaInicial = 0) {  
  climaActual <- climaInicial  
  historial <- c(climaActual)  
  
  for (i in 1:(dias - 1)) { #Bucle para asegurar que la longitud sea 'dias'.  
    climaActual <- estimarClima(climaActual)  
    historial <- c(historial, climaActual)  
  }  
  
  return(historial)  
}
```

Utilizar la función para simular 31 días de clima basado en la cadena de Markov usando temporalidad:

```
historialSimuladoSoleado <- simularClima(31, climaInicial = 0)  
print("Suponiendo que hoy ha sido un día soleado:")
```

```
## [1] "Suponiendo que hoy ha sido un día soleado:"  
print(historialSimuladoSoleado)
```

```
## [1] 0 0 0 1 0 2 2 2 2 0 0 0 0 0 0 1 0 0 1 0 2 1 2 2 1 1 1 2 2 2
```

```
historialSimuladoNublado <- simularClima(31, climaInicial = 1)  
print("Suponiendo que hoy ha sido un día nublado:")
```

```
## [1] "Suponiendo que hoy ha sido un día nublado:"  
print(historialSimuladoNublado)
```

```

## [1] 1 0 1 1 1 0 1 1 0 1 0 2 2 1 1 0 3 3 2 2 2 2 1 1 1 1 0 2 2 1 1
historialSimuladoLluvioso <- simularClima(31, climaInicial = 2)
print("Suponiendo que hoy ha sido un día lluvioso:")

## [1] "Suponiendo que hoy ha sido un día lluvioso:"
print(historialSimuladoLluvioso)

## [1] 2 2 0 2 2 1 3 2 2 2 3 2 2 2 1 1 1 0 0 1 1 1 1 1 1 2 1 1 1 1 1
historialSimuladoTormenta <- simularClima(31, climaInicial = 3)
print("Suponiendo que hoy ha sido un día de tormenta:")

## [1] "Suponiendo que hoy ha sido un día de tormenta:"
print(historialSimuladoTormenta)

## [1] 3 1 1 0 0 1 1 0 1 3 2 2 0 1 2 2 0 1 2 2 2 2 0 2 2 2 2 3 2 0
# Función para obtener el conteo con todos los estados
obtenerConteo <- function(historial) {
  estados <- c(0, 1, 2, 3)
  conteo <- table(factor(historial, levels = estados))
  conteo[as.character(estados)] <- conteo[as.character(estados)]
  return(conteo)
}

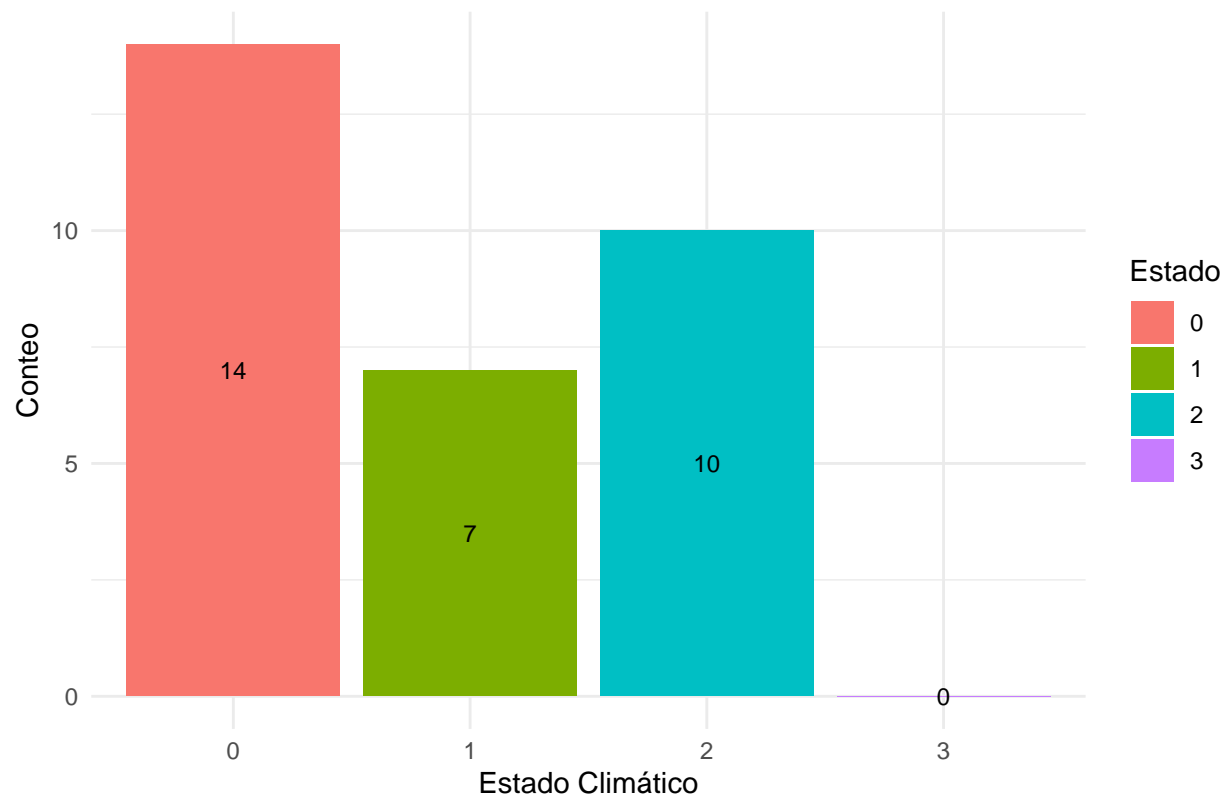
# Función para obtener el gráfico con todos los estados
obtenerConteoYPlot <- function(historial, titulo) {
  conteo <- obtenerConteo(historial)
  datosConteo <- data.frame(
    Estado = as.factor(names(conteo)),
    Conteo = as.numeric(conteo)
  )
  ggplot(datosConteo, aes(x = Estado, y = Conteo, fill = Estado)) +
    geom_bar(stat = "identity") +
    geom_text(aes(label = Conteo), position = position_stack(vjust = 0.5), size = 3) +
    labs(title = titulo, x = "Estado Climático", y = "Conteo") +
    theme_minimal()
}

# Obtener conteo y plotear para cada simulación
plotSoleado <- obtenerConteoYPlot(historialSimuladoSoleado, "Simulación de si el día empieza en soleado")
plotNublado <- obtenerConteoYPlot(historialSimuladoNublado, "Simulación de si el día empieza en nublado")
plotLluvioso <- obtenerConteoYPlot(historialSimuladoLluvioso, "Simulación de si empieza en día lluvioso")
plotTormenta <- obtenerConteoYPlot(historialSimuladoTormenta, "Simulación de si empieza en día de tormenta")

# Mostrar los gráficos
print(plotSoleado)

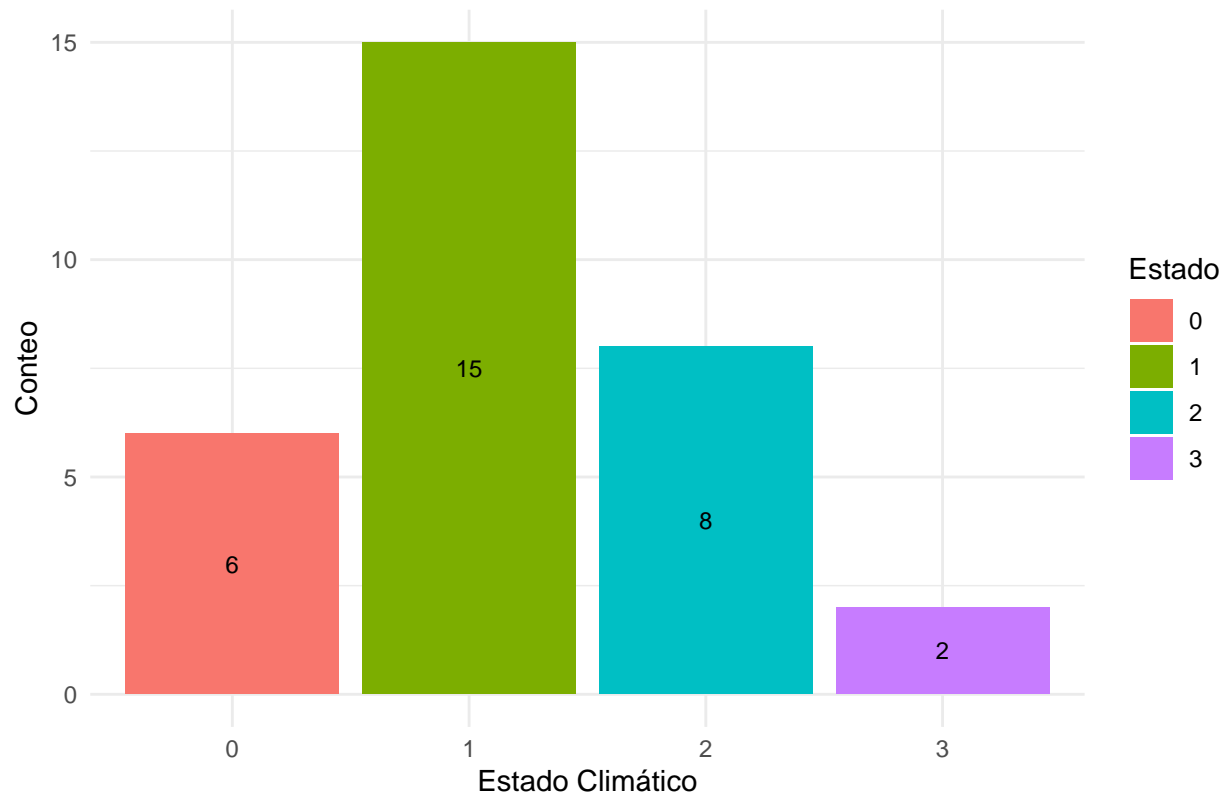
```

Simulación de si el día empieza en soleado; Estado 0



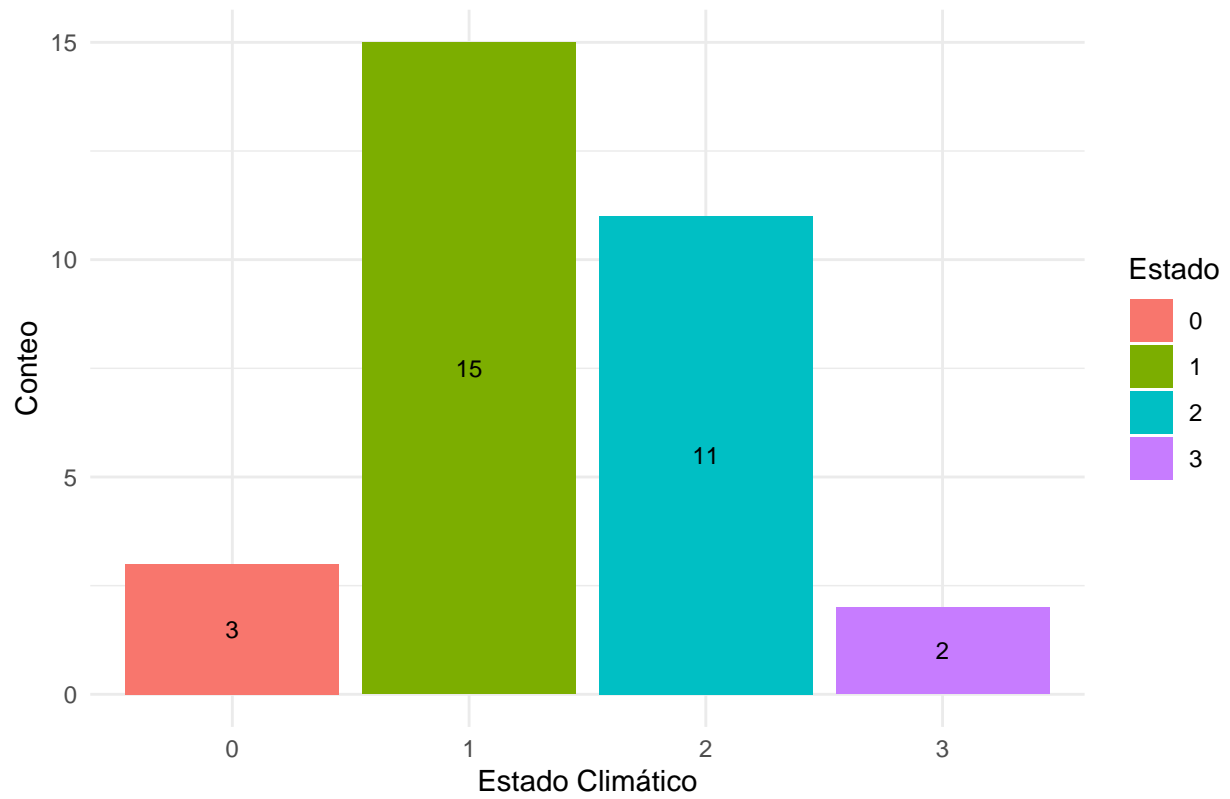
```
print(plotNublado)
```

Simulación de si el día empieza en nublado; Estado 1



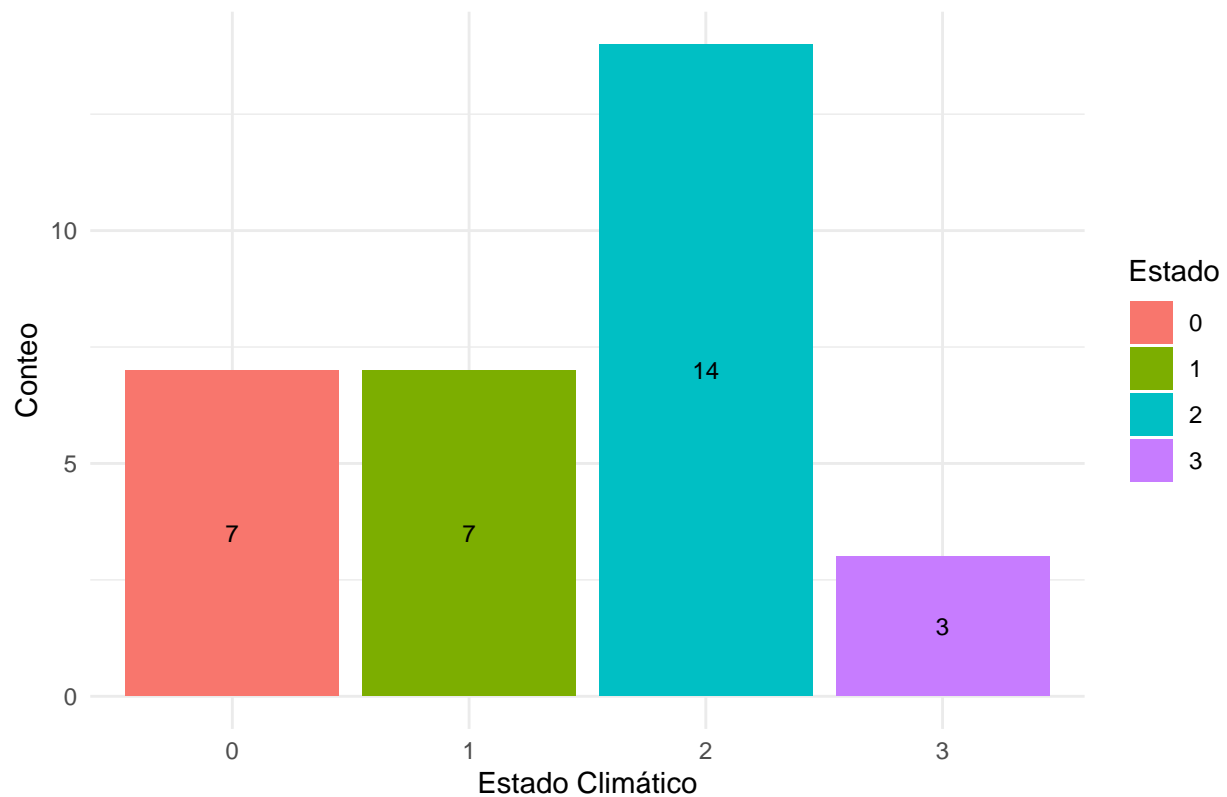
```
print(plotLluvioso)
```


Simulación de si empieza en día lluvioso; Estado 2



```
print(plotTormenta)
```

Simulación de si empieza en día de tormenta; Estado 3



```
# Función para obtener el conteo con todos los estados
obtenerConteo <- function(historial) {
  # Crear un vector con los estados posibles:
  estados <- c(0, 1, 2, 3)

  # Obtener el conteo para cada estado:
  conteo <- table(factor(historial, levels = estados))

  # Convertir a un vector con todos los estados:
  conteo[as.character(estados)] <- conteo[as.character(estados)]
  return(conteo)
}

# Función para obtener el gráfico con todos los estados:
obtenerConteoYPlot <- function(historial, titulo) {
  conteo <- obtenerConteo(historial)

  # Crear un marco de datos con todos los estados:
  datosConteo <- data.frame(
    Estado = as.factor(names(conteo)),
    Conteo = as.numeric(conteo)
  )

  # Trazar el gráfico con todos los estados
  ggplot(datosConteo, aes(x = Estado, y = Conteo, fill = Estado)) +
    geom_bar(stat = "identity") +
```

```

    geom_text(aes(label = Conteo), position = position_stack(vjust = 0.5), size = 3) +
    labs(title = titulo,
         x = "Estado Climático",
         y = "Conteo") +
    theme_minimal()
}

# Obtener el conteo para cada simulación
conteoSoleado <- obtenerConteo(historialSimuladoSoleado)
conteoNublado <- obtenerConteo(historialSimuladoNublado)
conteoLluvioso <- obtenerConteo(historialSimuladoLluvioso)
conteoTormenta <- obtenerConteo(historialSimuladoTormenta)

# Calcular el total y promedio para todos los estados
totalEstados <- conteoSoleado + conteoNublado + conteoLluvioso + conteoTormenta
promedioEstados <- totalEstados / 4

# Crear un marco de datos con todos los estados para el gráfico
datosTotales <- data.frame(
  Estado = as.factor(names(totalEstados)),
  Total = as.numeric(totalEstados),
  Promedio = as.numeric(promedioEstados)
)

# Trazar el gráfico con todos los estados
ggplot(datosTotales, aes(x = Estado)) +
  geom_bar(aes(y = Total, fill = "Total"), stat = "identity") +
  geom_bar(aes(y = Promedio, fill = "Promedio"), stat = "identity") +
  geom_text(aes(y = Total, label = Total), position = position_stack(vjust = 0.5), size = 3) +
  geom_text(aes(y = Promedio, label = round(Promedio, 2)), position = position_stack(vjust = 0.5), size = 3) +
  labs(title = "Total y Promedio de Estados en las 4 simulaciones",
       x = "Estado Climático",
       y = "Conteo") +
  theme_minimal() +
  scale_fill_manual(values = c("Total" = "blue", "Promedio" = "green"))

```

Total y Promedio de Estados en las 4 simulaciones

