

Predicción de la intensidad de precipitaciones usando el modelo de las cadenas de Markov en la región Áncash

Juan Andrés Vargas Castillo
Roosevelt Jhans Ubaldo Chavez

July 13, 2019

Contents

| | | |
|----------|--|-----------|
| 1 | Introducción | 3 |
| 2 | Definición del problema | 3 |
| 3 | Marco Teórico | 3 |
| 3.1 | Autómata probabilístico | 3 |
| 3.1.1 | Autómata Finito | 3 |
| 3.1.2 | Autómatas Finitos Deterministas y no Deterministas . . . | 4 |
| 3.1.3 | Definición de un Autómata probabilístico | 4 |
| 3.2 | Cadenas de Márkov | 4 |
| 3.2.1 | Probabilidad de Transición | 4 |
| 3.2.2 | Vector de Distribución | 4 |
| 3.2.3 | Comportamiento a Largo Plazo | 5 |
| 3.3 | Ejemplo del Clima | 6 |
| 4 | Estado del Arte | 6 |
| 4.1 | Forecasting of meteorological drought using Hidden Markov Model | 6 |
| 4.2 | Markov chain analysis of the rainfall patterns of five geographical locations in the south eastern coast of Ghana | 7 |
| 4.3 | Markov Chain Model and Its Application to Annual Rainfall Distribution for Crop Production | 7 |
| 5 | Propuesta | 7 |
| 6 | Cálculo de las matrices de transición | 8 |
| 6.1 | Matrices de transición | 12 |
| 7 | Conclusiones | 23 |

1 Introducción

Los autómatas son un modelo computacional muy importante para poder representar distintas distribuciones de un determinado conjunto de símbolos. Un autómata probabilístico es un tipo de autómata en el que se puede observar cual es la probabilidad de que un estado llegue a cualquiera de los otros estados. Estos autómatas están relacionados a las cadenas de Márkov, que se basan en los estados pasados por los que ya paso un sistema para poder predecir a que estado se llegará en el futuro.

A través de las propiedades de las cadenas de Márkov se buscará predecir la intensidad de las futuras precipitaciones en la región de Áncash en el futuro, para esto se usarán las definiciones de autómatas probabilísticos y cadenas de Márkov, así como también los datos meteorológicos previos de la región.

2 Definición del problema

Las precipitaciones en las zonas altoandinas de nuestro país son de suma importancia para la siembra y/o cosecha de productos. Una siembra antes de tiempo, podría ocasionar que la planta muera por falta de agua, o en un mayor gasto en riego por parte del agricultor. Del mismo modo, cosechar fuera de tiempo podría ocasionar que se pierda por completo el cultivo, ya que se encuentra en etapa de maduración y ya no requiere agua, de hacerlo podría podrirse la raíz y por ende el fruto. Esto último puede tener un mayor impacto en el cultivo y en la vida del agricultor si se dan las inundaciones a causa de las intensas lluvias.

En el presente año, según la Cámara de Comercio de Moquegua, más de 800 hectáreas de cultivos fueron perdidas a causa del desborde del río Moquegua ocasionado por las intensas precipitaciones de los últimos meses. Casos similares a este se han venido reportando en los últimos años en las regiones como Piura, San Martín, Ucayali y Áncash. Es por dicho motivo, que modelar las precipitaciones resultan de suma importancia para la agricultura en nuestro país, así como en acciones para mitigar su impacto.

3 Marco Teórico

3.1 Autómata probabilístico

3.1.1 Autómata Finito

Se define como autómata finito a un modelo computacional, tiene como finalidad reconocer si una entrada o lenguaje pertenece a un conjunto de lenguajes regulares o no. Un autómata también puede ser representado como una 5 - tupla, formada por el conjunto de estados Q a los que puede llegar la entrada, el alfabeto Σ con los símbolos que podemos usar para crear nuestros lenguajes, el estado inicial q_0 por donde debe empezar el recorrido nuestro lenguaje, la función de transición δ

que nos indica el estado al que llegará un lenguaje al recibir un nuevo símbolo, por último el conjunto de estados finales F que indica si el lenguaje es aceptado o no.

3.1.2 Autómatas Finitos Deterministas y no Deterministas

Se tienen dos tipos de autómatas finitos, deterministas y no deterministas. En el primer caso, cuando un símbolo pasa por la función de transición, sólo puede llegar a un estado o a ninguno. Por otro lado, en los no deterministas la función de transición puede retornar más de un estado.

3.1.3 Definición de un Autómata probabilístico

Este tipo de autómata se trata de una generalización de un autómata finito no determinista. A diferencia del mencionado, la función de transición no sólo retorna el conjunto de estados a los que se llega con un símbolo, sino que devuelve la probabilidad de llegar a cualquier estado en el autómata.

3.2 Cadenas de Márkov

Las cadenas de Márkov o Sistemas de Márkov son un tipo de autómata probabilístico, en este se busca predecir un posible estado futuro de un sistema. Dicha probabilidad se basa en la historia previa del sistema, esta se encuentra resumida en el estado actual en el que se encuentra. Las cadenas de Márkov tienen diversas aplicaciones en la meteorología, modelos epidemiológicos, problemas con simulaciones, genética y más.

3.2.1 Probabilidad de Transición

Se define como probabilidad de transición P_{ij} a la probabilidad que hay de pasar desde un estado i a un estado j . El conjunto total de las probabilidades de transición se pueden representar en una matriz de transición.

La siguiente matriz representa un Sistema de Márkov con n estados.

$$\begin{bmatrix} P_{11} & P_{12} & P_{13} & \dots & P_{1n} \\ P_{21} & P_{22} & P_{23} & \dots & P_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ P_{n1} & P_{n2} & P_{n3} & \dots & P_{nn} \end{bmatrix}$$

Cada fila i representa un vector con las probabilidades tal que estamos en el estado i , por lo tanto todas las probabilidades de una fila deben sumar 1.

3.2.2 Vector de Distribución

Vector con entradas para cada probabilidad de cada estado en un determinado tiempo o número de pasos. La suma de los valores en el vector debe ser igual a 1.

El siguiente vector representa una distribución inicial con n estados.

$$\pi(0) = [X_1 \quad X_2 \quad X_3 \quad \dots \quad X_n]$$

Para hallar el siguiente vector de distribución $\pi(1)$ se debe multiplicar el vector de distribución inicial por la matriz de transición.

$$\pi(1) = \pi(0) * P$$

$$\pi(1) = [X_1 \quad X_2 \quad X_3 \quad \dots \quad X_n] * \begin{bmatrix} P_{11} & P_{12} & P_{13} & \dots & P_{1n} \\ P_{21} & P_{22} & P_{23} & \dots & P_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ P_{n1} & P_{n2} & P_{n3} & \dots & P_{nn} \end{bmatrix}$$

Para hallar $\pi(2)$ debemos multiplicar el vector anterior y multiplicarlo por la matriz de transición.

$$\pi(2) = \pi(1) * P \Rightarrow \pi(2) = \pi(0) * P^2$$

Por lo tanto si queremos hallar el vector de distribución $\pi(n)$ después de n pasos.

$$\pi(n) = \pi(0) * P^n$$

3.2.3 Comportamiento a Largo Plazo

Si para un vector de distribución se cumple que $\pi(n) = \pi(n) * P$ significa que llegamos a vector de estado de equilibrio o vector estabilizado. Por otro lado, una matriz de equilibrio o matriz estabilizada es una matriz donde se cumple $P = P * P$ donde además cada fila es igual a las demás.

Para hallar un vector de distribución a largo plazo, vamos potenciando nuestra matriz de transición hasta llegar a una potencia que representa una matriz estabilizada.

Ejemplo:

Si tenemos la siguiente matriz de transición:

$$P = \begin{bmatrix} 0.2 & 0.8 & 0 \\ 0.4 & 0 & 0.6 \\ 0.5 & 0.5 & 0 \end{bmatrix}$$

Al calcular P^{256} llegaremos a la matriz de transición estabilizada:

$$P = \begin{bmatrix} 35/99 & 40/99 & 24/99 \\ 35/99 & 40/99 & 24/99 \\ 35/99 & 40/99 & 24/99 \end{bmatrix}$$

3.3 Ejemplo del Clima

Dado un pueblo donde el clima puede cambiar de un día para otro:

- El clima sólo puede ser o húmedo o seco.
- Si el día actual es seco, la probabilidad de que mañana sea seco es de 80%.
- Si el día actual es húmedo, la probabilidad de que mañana sea seco es de 60%.

$X = EstadoDelClima.$

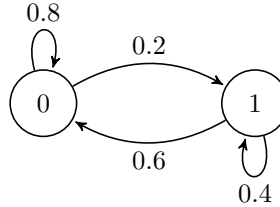
$$X = \begin{cases} 0 & seco \\ 1 & húmedo \end{cases}$$

1. Matriz de transición

| Estado Actual X | Estado Siguiente X | |
|-----------------|--------------------|----------------|
| | 0 | 1 |
| 0 | $P_{00} = 0.8$ | $P_{01} = 0.2$ |
| 1 | $P_{10} = 0.6$ | $P_{11} = 0.4$ |

Table 1: Matriz de transición

2. Diagrama de transición



4 Estado del Arte

4.1 Forecasting of meteorological drought using Hidden Markov Model

En este estudio, a través de un modelo homogéneo de Markov oculto, se busca pronosticar sequías en base a al índice de precipitación estandarizado, a corto y mediano plazo. Para medir el rendimiento de este modelo se usaron varios criterios de pronóstico. Los resultados parecen validar la fiabilidad del modelo usado para pronosticar posibles eventos con las sequías.

4.2 Markov chain analysis of the rainfall patterns of five geographical locations in the south eastern coast of Ghana

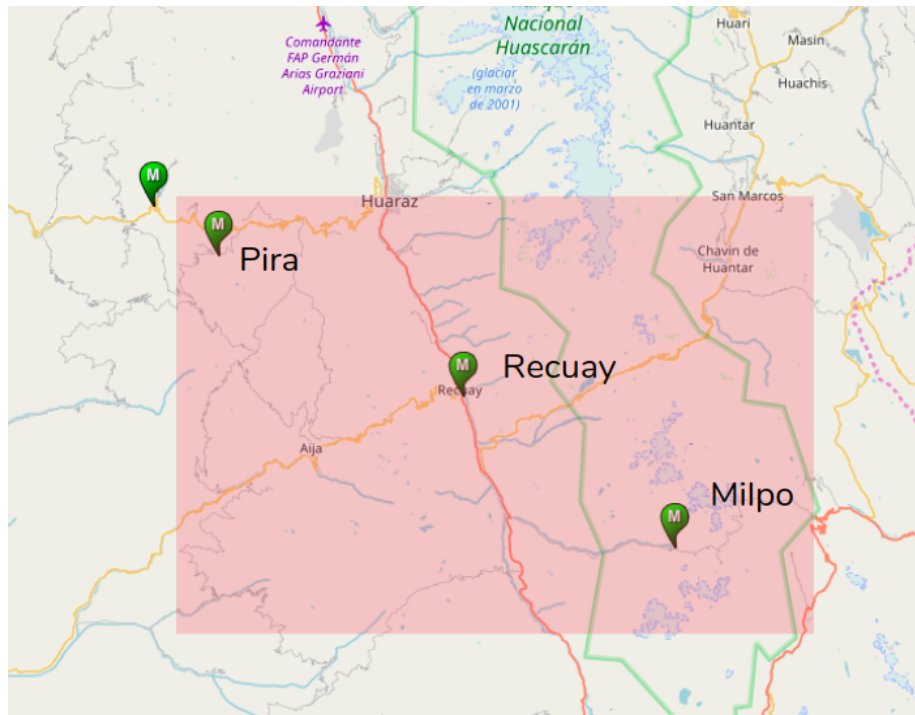
Este estudio realiza un análisis de Márkov usando datos de lluvias registradas por día entre los años 1980 y 2010 en las siguientes ciudades pertenecientes a la costa sudeste de Ghana; Costa del Cabo, Accra, Akuse, Akatsi y Keta. Se calculan las matrices de transición para cada ciudad y mes usando la probabilidad condicional de si en un día hubo o no lluvia. De los resultados se concluyó que para Accra, la temporada de lluvias era en los meses de mayo a junio y de septiembre a octubre. También determinamos que la probabilidad de lluvia generalmente tiende a aumentar de este a oeste a lo largo de la costa sureste de Ghana.

4.3 Markov Chain Model and Its Application to Annual Rainfall Distribution for Crop Production

Este estudio trabaja con distintos estados y tiempos a través del uso de cádenas de Márkov. A través de este principio se plantea un modelo de cuatro estados para la distribución anual de lluvias en Minna con respecto a la producción de cultivos. Con este modelo, podemos recibir el estado de un determinado año y en base a este poder determinar la probabilidad de hacer una transición a cualquiera de los otros estados. El modelo provee información sobre las lluvias en relación a los cultivos que puede ser usada tanto por campesinos como por el gobierno.

5 Propuesta

En el presente artículo se van a analizar las presencia de precipitaciones en la región de Áncash mediante el uso de las cadenas de Markov. La data a usar en este estudio será recopilada de 3 estaciones metereológicas del SENAHMI dispersas a lo largo de la región (Milpo, Recuay y Pira). Los meses considerados para el estudio corresponden al periodo de siembra del maíz entre los meses de marzo-abril y su respectiva cosecha entre julio-agosto del 2018. Se uso el maíz como referencia, ya que es uno de los productos con mayor volumen de producción en la región.



Se modelará la cadena de Markov mediante una autómatas probabilística de tres estados seco, húmedo y lluvioso.

6 Cálculo de las matrices de transición

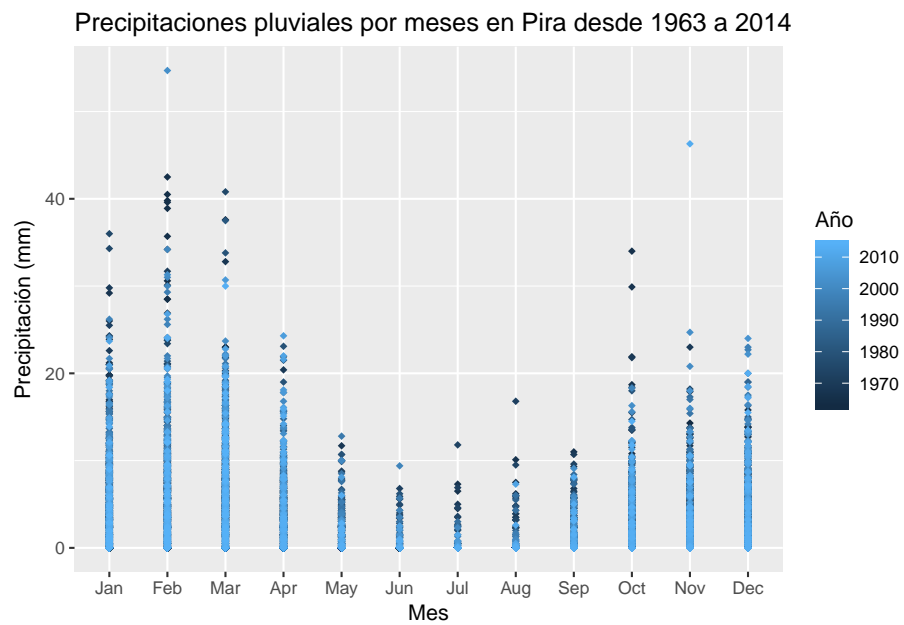
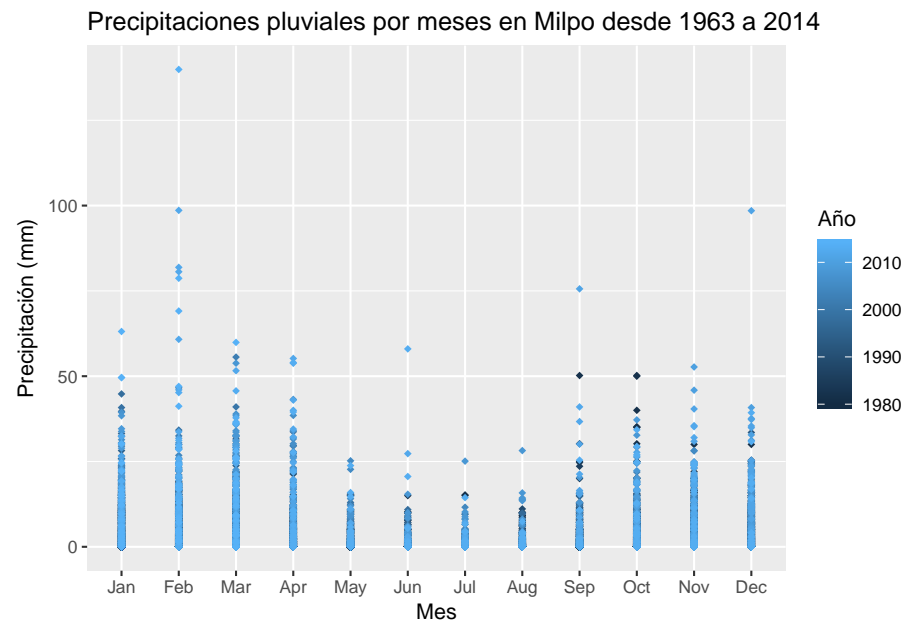
Se realizaron los cálculos de la matriz de transición para los puesto metereológicos de Milpo, Pira y Recuay. Todos ellos dentro del departamento de Ancash. El proceso se realizó haciendo uso de R y se detalla a continuación:

Los dataset fueron extraídos de la página web del Senahmi.

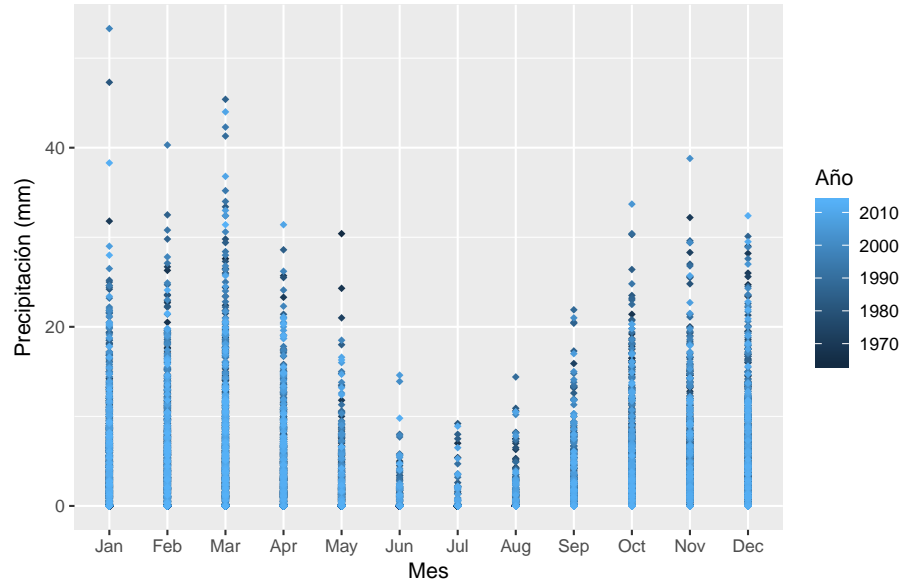
```
#Cargar los datasets
milpo<-read.table("milpo.txt", header = FALSE, sep = " ", dec = ".")
pira<-read.table("pira.txt", header = FALSE, sep = " ", dec = ".")
recuay<-read.table("recuay.txt", header = FALSE, sep = " ", dec = ".")

#Cambiar los datos faltantes por NA
milpo[milpo==--99.9]<-NA
pira[pira==--99.9]<-NA
recuay[recuay==--99.9]<-NA
```


A continuación se muestran gráficos que resumen la data que se está utilizando:

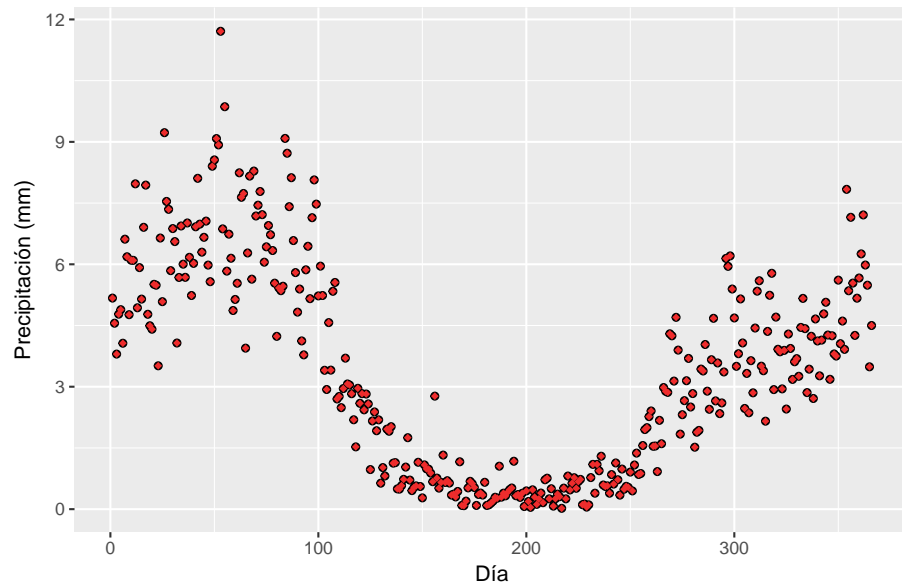


Precipitaciones pluviales por meses en Recuay desde 1963 a 2014

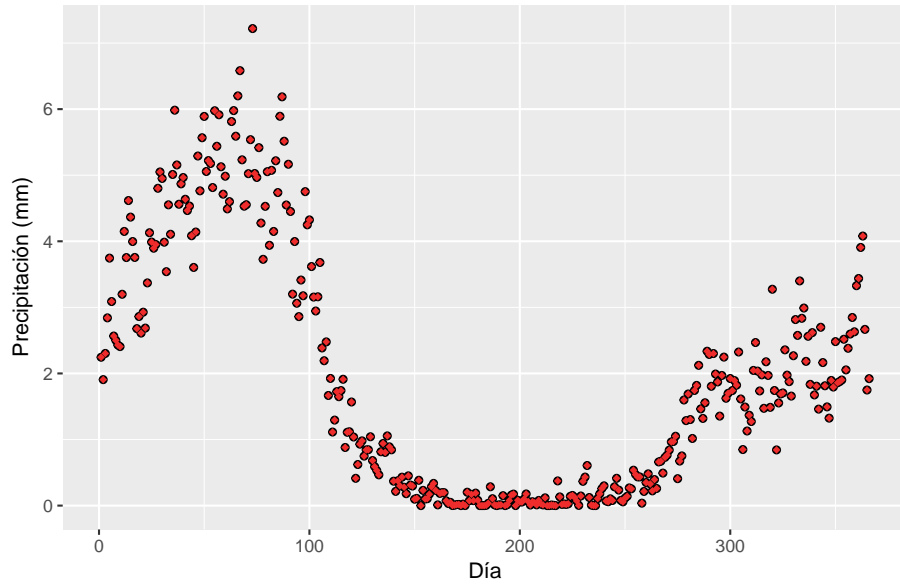


También se genera un gráfico de un año promedio para cada una de las estaciones meteorológicas consideradas en el estudio:

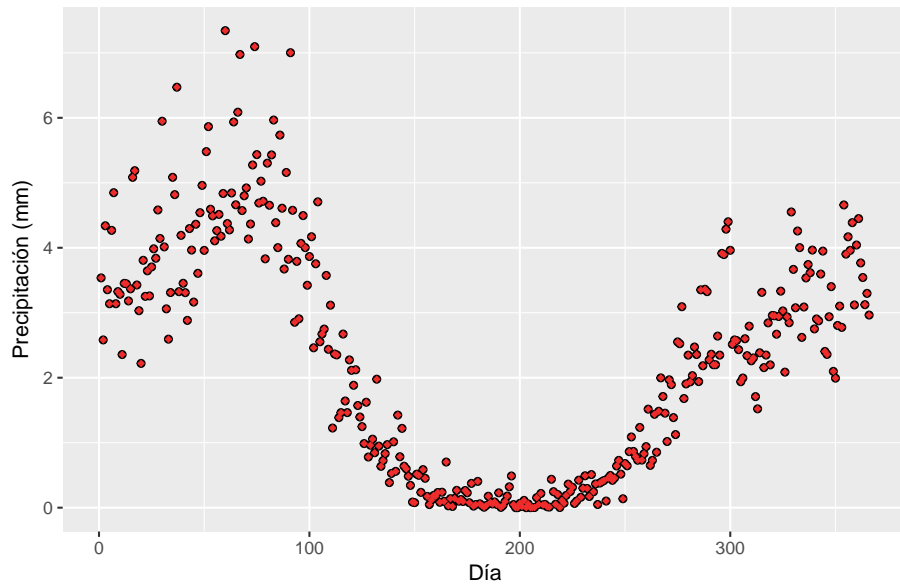
Precipitaciones pluviales anual promedio en Milpo desde 1963 a 2014



Precipitaciones pluviales anual promedio en Pira desde 1963 a 2014



Precipitaciones pluviales anual promedio en Recuay desde 1963 a 2014



6.1 Matrices de transición

Para realizar la matriz transición se consideraron tres estados: seco, húmedo y lluvioso. El criterio para clasificar un día dentro de estos estados se base en la clasificación de las lluvias según su intensidad.

Table 2: Clasificación de las precipitaciones según su intensidad

| Intensidad | mm/h |
|-------------------|-------------|
| Ligera | < 2.5 |
| Moderada | 2.5 - 7.5 |
| Fuerte | > 7.5 |

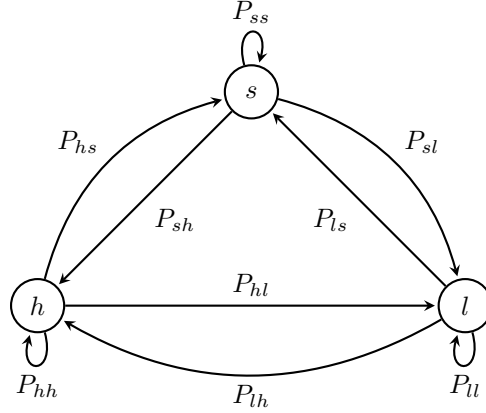
Fuente:García,J.(2004)

Table 3: Criterio de clasificación de los estados

| Intensidad | Estado | Nomenclatura |
|-------------------|---------------|---------------------|
| Ligera | Seco | <i>s</i> |
| Moderada | Húmedo | <i>h</i> |
| Fuerte | Lluvioso | <i>l</i> |

A partir de lo anterior se escogieron los siguientes estados de transición y se les asignó un número distintivo.

- Ayer fue seco y hoy fue seco P_{ss} : 1
- Ayer fue seco y hoy fue húmedo P_{sh} : 2
- Ayer fue seco y hoy fue lluvioso P_{sl} : 3
- Ayer fue húmedo y hoy fue seco P_{hs} : 4
- Ayer fue húmedo y hoy fue húmedo P_{hh} : 5
- Ayer fue húmedo y hoy fue lluvioso P_{hl} : 6
- Ayer fue lluvioso y hoy fue seco P_{ls} : 7
- Ayer fue lluvioso y hoy fue húmedo P_{lh} : 8
- Ayer fue lluvioso y hoy fue lluvioso P_{ll} : 9



Por lo tanto, la **matriz de transición**: será una matriz de 3×3 como se muestra a continuación:

$$M = \begin{pmatrix} P_{ss} & P_{sh} & P_{sl} \\ P_{hs} & P_{hh} & P_{hl} \\ P_{ls} & P_{lh} & P_{ll} \end{pmatrix}$$

La siguiente función se hizo para determinar los estados antes mencionados:

```

# Determinación de estados
conteoCasos<-function (data){
  data$V4[data$V4<2.5]<-0
  data$V4[data$V4>=2.5 & data$V4<7.5]<-1
  data$V4[data$V4>=7.5]<-2
  tot<-data %>% select(-V6) %>% filter(V4 == 0 | V4 == 1 | V4 == 2) ##obviamos NAs
  tot$V5[1]<-0 #el primer día tiene valor cero
  for (i in 2:nrow(tot)){
    if (tot$V4[i-1]==0 & tot$V4[i]==0){ #Pss
      tot$V5[i]<-1
    } else if (tot$V4[i-1]==0 & tot$V4[i]==1){ #Psh
      tot$V5[i]<-2
    } else if (tot$V4[i-1]==0 & tot$V4[i]==2){ #Psl
      tot$V5[i]<-3
    } else if (tot$V4[i-1]==1 & tot$V4[i]==0){ #Phs
      tot$V5[i]<-4
    } else if (tot$V4[i-1]==1 & tot$V4[i]==1){ #Phh
      tot$V5[i]<-5
    } else if (tot$V4[i-1]==1 & tot$V4[i]==2){ #Phl
      tot$V5[i]<-6
    } else if (tot$V4[i-1]==2 & tot$V4[i]==0){ #Pls
      tot$V5[i]<-7
    } else if (tot$V4[i-1]==2 & tot$V4[i]==1){ #Plh
      tot$V5[i]<-8
    }
  }
}

```

```

    }else if(tot$V4[i-1]==2 & tot$V4[i]==2){ #Pll
      tot$V5[i]<-9
    }

  }
  return (tot)
}

```

```

# Función que verifica que existan valores en los meses, si no lo hay los rellena con ceros
check <- function (data){
  dif <- setdiff(c(1:12),c(data$V2))
  if (length(dif)>=1){
    for(i in dif){
      data <- rbind(data, c(i, 0))
    }
  }
  data <- data[order(data$V2),]
  return(data)
}

```

#Función de conteo y probabilidad de cada estado de la matriz de transición

```

probabilidades<-function(data){
  dta <- conteoCasos(data) %>% filter(V5 !=0) #eliminamos primer día
  uno <-dta %>% group_by(V2) %>% filter(V5==1) %>% summarise(uno=n()) %>% check
  dos <-dta %>% group_by(V2) %>% filter(V5==2) %>% summarise(dos=n()) %>% check
  tres <-dta %>% group_by(V2) %>% filter(V5==3) %>% summarise(tres=n()) %>% check
  cuatro <-dta %>% group_by(V2) %>% filter(V5==4) %>% summarise(cuatro=n()) %>% check
  cinco <-dta %>% group_by(V2) %>% filter(V5==5) %>% summarise(cinco=n()) %>% check
  seis <-dta %>% group_by(V2) %>% filter(V5==6) %>% summarise(seis=n()) %>% check
  siete <-dta %>% group_by(V2) %>% filter(V5==7) %>% summarise(siete=n()) %>% check
  ocho <-dta %>% group_by(V2) %>% filter(V5==8) %>% summarise(ocho=n()) %>% check
  nueve <-dta %>% group_by(V2) %>% filter(V5==9) %>% summarise(nueve=n()) %>% check

  a<-dta %>% group_by(V2) %>% filter(V5==6)
  table(a$V2)

  resultado <- merge(merge(merge(merge(merge(merge(merge(uno, dos, "V2"), tres, "V2"),
  mutate(seco = uno + dos + tres , humedo = cuatro + cinco + seis, lluvioso = siete + ocho +
    group_by(V2) %>% summarise(pUno = uno/seco, pDos = dos/seco, pTres = tres/seco,
    pCuatro = cuatro/humedo, pCinco = cinco/humedo, pSeis = seis/humedo, pSiete = siete/lluv
  return (resultado)
}

```

```

# Formateo a listas de las matrices de transición para cada mes
propaMatriz <- function(data){
  prop <- probabilidades(data)
  start <- list(matrix(nrow = 3, ncol = 3, c(prop[1,1], prop[1,2],
                                             prop[1,3], prop[1,4],
                                             prop[1,5], prop[1,6],
                                             prop[1,7], prop[1,8],
                                             prop[1,9]), byrow = TRUE))

  for(i in 2:12){
    new <-list(matrix(nrow = 3, ncol = 3, c(prop[i,1], prop[i,2],
                                             prop[i,3], prop[i,4],
                                             prop[i,5], prop[i,6],
                                             prop[i,7], prop[i,8],
                                             prop[i,9]), byrow = TRUE))

    start <- c(start, new)
  }
  return (start)
}

```

```

#Función para estabilizar la matriz en 2^n
estabilizar <- function(data, n){
  estado <- TRUE
  contador <- 0
  matrices <- propaMatriz(data)
  while(estado){
    contador <- contador + 1
    for (i in 1:12){
      matrices[[i]] <- matrices[[i]] %*% matrices[[i]]
      if(round(matrices[[1]][1,1],n) == round(matrices[[1]][2,1],n)){
        estado <- FALSE
      }
    }
  }
  print(paste("Iteraciones: ", contador))
  return (matrices)
}

```

```

#Limiting probabilities
limiting <- function(data, n){
  matrices <- estabilizar(data, n)
  matrices[[1]] <- matrix(nrow = 1, ncol=3, c(1,0,0), byrow = TRUE) %*% matrices[[1]]
  first<-cbind(as.data.frame(month.abb[1]), as.data.frame(round(matrices[[1]],3)))
  names(first)<- c("Mes", "Seco", "Húmedo", "Lluvioso")
  for (i in 2:12){
    matrices[[i]] <- matrix(nrow = 1, ncol=3, c(1,0,0), byrow = TRUE) %*% matrices[[i]]
  }
}

```

```

        second <- cbind(as.data.frame(month.abb[i]), as.data.frame(round(matrices[[i]],3)))
        names(second) <- c("Mes", "Seco", "Húmedo", "Lluvioso")
        first <- rbind(first, second)
    }

    return(first)
}

```

A continuación se muestran las matrices estabilizadas:

```

#Matriz estabilizada de Milpo
estabilizar(milpo, 5)

## [1] "Iteraciones:  4"

## [[1]]
##           [,1]      [,2]      [,3]
## [1,] 0.4713439 0.2121192 0.3165370
## [2,] 0.4713394 0.2121197 0.3165408
## [3,] 0.4713378 0.2121199 0.3165423
##
## [[2]]
##           [,1]      [,2]      [,3]
## [1,] 0.4055748 0.2617739 0.3288425
## [2,] 0.4050286 0.2614425 0.3284434
## [3,] 0.4054660 0.2617395 0.3288285
##
## [[3]]
##           [,1]      [,2]      [,3]
## [1,] 0.4114407 0.2604170 0.3281423
## [2,] 0.4114404 0.2604170 0.3281426
## [3,] 0.4114400 0.2604171 0.3281430
##
## [[4]]
##           [,1]      [,2]      [,3]
## [1,] 0.5908909 0.2309756 0.1781336
## [2,] 0.5908771 0.2309795 0.1781434
## [3,] 0.5908671 0.2309824 0.1781505
##
## [[5]]
##           [,1]      [,2]      [,3]
## [1,] 0.8447565 0.1164156 0.04054570
## [2,] 0.8457744 0.1165559 0.04059455
## [3,] 0.8448180 0.1164241 0.04054865
##
## [[6]]

```



```

##           [,1]           [,2]           [,3]
## [1,] 0.9377285 0.04371879 0.01855267
## [2,] 0.9377284 0.04371887 0.01855271
## [3,] 0.9377283 0.04371893 0.01855275
##
## [[7]]
##           [,1]           [,2]           [,3]
## [1,] 0.9525394 0.03713284 0.01048171
## [2,] 0.9525273 0.03713236 0.01048158
## [3,] 0.9538298 0.03718314 0.01049591
##
## [[8]]
##           [,1]           [,2]           [,3]
## [1,] 0.9259193 0.05252566 0.02155505
## [2,] 0.9259122 0.05252942 0.02155835
## [3,] 0.9259016 0.05253505 0.02156330
##
## [[9]]
##           [,1]           [,2]           [,3]
## [1,] 0.7782768 0.1463957 0.07322787
## [2,] 0.7771801 0.1461908 0.07312561
## [3,] 0.7778585 0.1463192 0.07319003
##
## [[10]]
##           [,1]           [,2]           [,3]
## [1,] 0.6275643 0.2156275 0.1577380
## [2,] 0.6282087 0.2158518 0.1579037
## [3,] 0.6266553 0.2153197 0.1575155
##
## [[11]]
##           [,1]           [,2]           [,3]
## [1,] 0.6196131 0.1955618 0.1848252
## [2,] 0.6196084 0.1955631 0.1848284
## [3,] 0.6196065 0.1955637 0.1848298
##
## [[12]]
##           [,1]           [,2]           [,3]
## [1,] 0.5246001 0.2303345 0.2450654
## [2,] 0.5245862 0.2303380 0.2450758
## [3,] 0.5245803 0.2303395 0.2450802

```

```

#Matriz estabilizada de Pira
estabilizar(recuay, 5)

```

```

## [1] "Iteraciones: 4"

```

```

## [[1]]
##           [,1]      [,2]      [,3]
## [1,] 0.5957145 0.2190385 0.185247
## [2,] 0.5957145 0.2190385 0.185247
## [3,] 0.5957145 0.2190385 0.185247
##
## [[2]]
##           [,1]      [,2]      [,3]
## [1,] 0.5281705 0.2556959 0.2215087
## [2,] 0.5268264 0.2550452 0.2209450
## [3,] 0.5272660 0.2552580 0.2211294
##
## [[3]]
##           [,1]      [,2]      [,3]
## [1,] 0.4804047 0.2735160 0.2544579
## [2,] 0.4797710 0.2731552 0.2541222
## [3,] 0.4797101 0.2731205 0.2540899
##
## [[4]]
##           [,1]      [,2]      [,3]
## [1,] 0.6542532 0.217887 0.1278598
## [2,] 0.6542532 0.217887 0.1278598
## [3,] 0.6542532 0.217887 0.1278598
##
## [[5]]
##           [,1]      [,2]      [,3]
## [1,] 0.8859335 0.07750568 0.03593485
## [2,] 0.8850994 0.07743271 0.03590102
## [3,] 0.8867617 0.07757814 0.03596844
##
## [[6]]
##           [,1]      [,2]      [,3]
## [1,] 0.9756538 0.0197332 0.00461301
## [2,] 0.9756538 0.0197332 0.00461301
## [3,] 0.9756538 0.0197332 0.00461301
##
## [[7]]
##           [,1]      [,2]      [,3]
## [1,] 0.9895547 0.007696537 0.002748763
## [2,] 0.9895547 0.007696537 0.002748763
## [3,] 0.9895547 0.007696537 0.002748763
##
## [[8]]
##           [,1]      [,2]      [,3]
## [1,] 0.9625906 0.02874606 0.008663316
## [2,] 0.9625906 0.02874606 0.008663316

```

```
## [3,] 0.9625906 0.02874606 0.008663316
##
## [[9]]
##      [,1]      [,2]      [,3]
## [1,] 0.8655563 0.09069906 0.04374468
## [2,] 0.8655563 0.09069906 0.04374468
## [3,] 0.8655563 0.09069906 0.04374468
##
## [[10]]
##      [,1]      [,2]      [,3]
## [1,] 0.6785747 0.1977618 0.1236635
## [2,] 0.6785747 0.1977618 0.1236635
## [3,] 0.6785747 0.1977618 0.1236635
##
## [[11]]
##      [,1]      [,2]      [,3]
## [1,] 0.6968886 0.1693864 0.1356827
## [2,] 0.6970200 0.1694184 0.1357083
## [3,] 0.6978087 0.1696101 0.1358619
##
## [[12]]
##      [,1]      [,2]      [,3]
## [1,] 0.6302411 0.1876964 0.1793328
## [2,] 0.6294622 0.1874645 0.1791112
## [3,] 0.6300815 0.1876489 0.1792874
```

```
#Matriz estabilizada de Recuay
estabilizar(recuay, 5)
```

```
## [1] "Iteraciones: 4"

## [[1]]
##      [,1]      [,2]      [,3]
## [1,] 0.5957145 0.2190385 0.185247
## [2,] 0.5957145 0.2190385 0.185247
## [3,] 0.5957145 0.2190385 0.185247
##
## [[2]]
##      [,1]      [,2]      [,3]
## [1,] 0.5281705 0.2556959 0.2215087
## [2,] 0.5268264 0.2550452 0.2209450
## [3,] 0.5272660 0.2552580 0.2211294
##
## [[3]]
##      [,1]      [,2]      [,3]
## [1,] 0.4804047 0.2735160 0.2544579
```

```

## [2,] 0.4797710 0.2731552 0.2541222
## [3,] 0.4797101 0.2731205 0.2540899
##
## [[4]]
##           [,1]      [,2]      [,3]
## [1,] 0.6542532 0.217887 0.1278598
## [2,] 0.6542532 0.217887 0.1278598
## [3,] 0.6542532 0.217887 0.1278598
##
## [[5]]
##           [,1]      [,2]      [,3]
## [1,] 0.8859335 0.07750568 0.03593485
## [2,] 0.8850994 0.07743271 0.03590102
## [3,] 0.8867617 0.07757814 0.03596844
##
## [[6]]
##           [,1]      [,2]      [,3]
## [1,] 0.9756538 0.0197332 0.00461301
## [2,] 0.9756538 0.0197332 0.00461301
## [3,] 0.9756538 0.0197332 0.00461301
##
## [[7]]
##           [,1]      [,2]      [,3]
## [1,] 0.9895547 0.007696537 0.002748763
## [2,] 0.9895547 0.007696537 0.002748763
## [3,] 0.9895547 0.007696537 0.002748763
##
## [[8]]
##           [,1]      [,2]      [,3]
## [1,] 0.9625906 0.02874606 0.008663316
## [2,] 0.9625906 0.02874606 0.008663316
## [3,] 0.9625906 0.02874606 0.008663316
##
## [[9]]
##           [,1]      [,2]      [,3]
## [1,] 0.8655563 0.09069906 0.04374468
## [2,] 0.8655563 0.09069906 0.04374468
## [3,] 0.8655563 0.09069906 0.04374468
##
## [[10]]
##           [,1]      [,2]      [,3]
## [1,] 0.6785747 0.1977618 0.1236635
## [2,] 0.6785747 0.1977618 0.1236635
## [3,] 0.6785747 0.1977618 0.1236635
##
## [[11]]

```

```
##           [,1]      [,2]      [,3]
## [1,] 0.6968886 0.1693864 0.1356827
## [2,] 0.6970200 0.1694184 0.1357083
## [3,] 0.6978087 0.1696101 0.1358619
##
## [[12]]
##           [,1]      [,2]      [,3]
## [1,] 0.6302411 0.1876964 0.1793328
## [2,] 0.6294622 0.1874645 0.1791112
## [3,] 0.6300815 0.1876489 0.1792874
```

Finalmente, los vectores estabilizados y limitados para cada estación meteorológica se detalla a continuación:

```
#Vector estabilizada de Milpo
limiting(milpo, 5)
```

```
## [1] "Iteraciones: 4"

##      Mes   Seco   Húmedo  Lluvioso
## [1,] "Jan" "0.471" "0.212" "0.317"
## [2,] "Feb" "0.406" "0.262" "0.329"
## [3,] "Mar" "0.411" "0.260" "0.328"
## [4,] "Apr" "0.591" "0.231" "0.178"
## [5,] "May" "0.845" "0.116" "0.041"
## [6,] "Jun" "0.938" "0.044" "0.019"
## [7,] "Jul" "0.953" "0.037" "0.010"
## [8,] "Aug" "0.926" "0.053" "0.022"
## [9,] "Sep" "0.778" "0.146" "0.073"
## [10,] "Oct" "0.628" "0.216" "0.158"
## [11,] "Nov" "0.620" "0.196" "0.185"
## [12,] "Dec" "0.525" "0.230" "0.245"
```

```
#Vector estabilizada de Pira
limiting(pira, 1)
```

```
## [1] "Iteraciones: 3"

##      Mes   Seco   Húmedo  Lluvioso
## [1,] "Jan" "0.574" "0.254" "0.166"
## [2,] "Feb" "0.456" "0.294" "0.247"
## [3,] "Mar" "0.387" "0.346" "0.268"
## [4,] "Apr" "0.637" "0.273" "0.090"
## [5,] "May" "0.900" "0.087" "0.013"
## [6,] "Jun" "0.984" "0.015" "0.001"
## [7,] "Jul" "0.997" "0.010" "0.001"
```

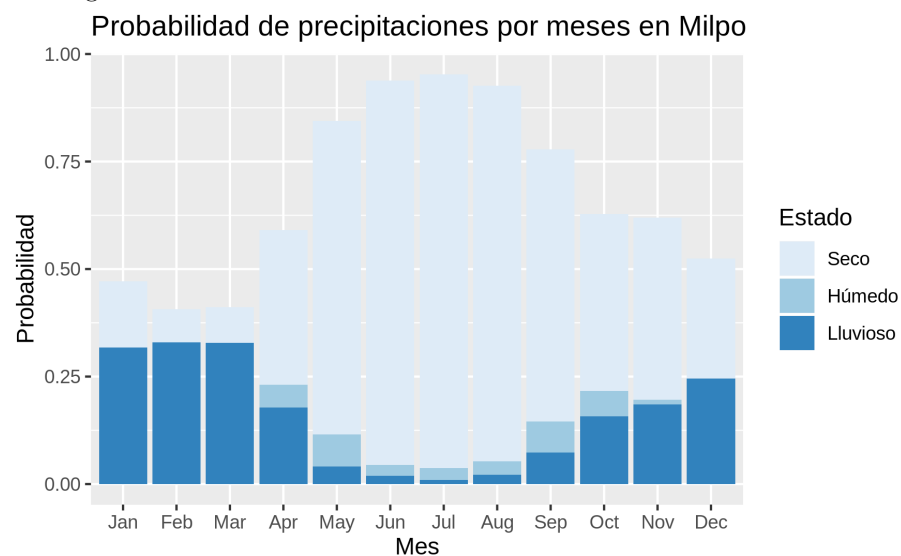
```
## [8,] "Aug" "0.974" "0.016" "0.002"
## [9,] "Sep" "0.921" "0.068" "0.011"
## [10,] "Oct" "0.754" "0.185" "0.061"
## [11,] "Nov" "0.716" "0.203" "0.082"
## [12,] "Dec" "0.685" "0.219" "0.096"
```

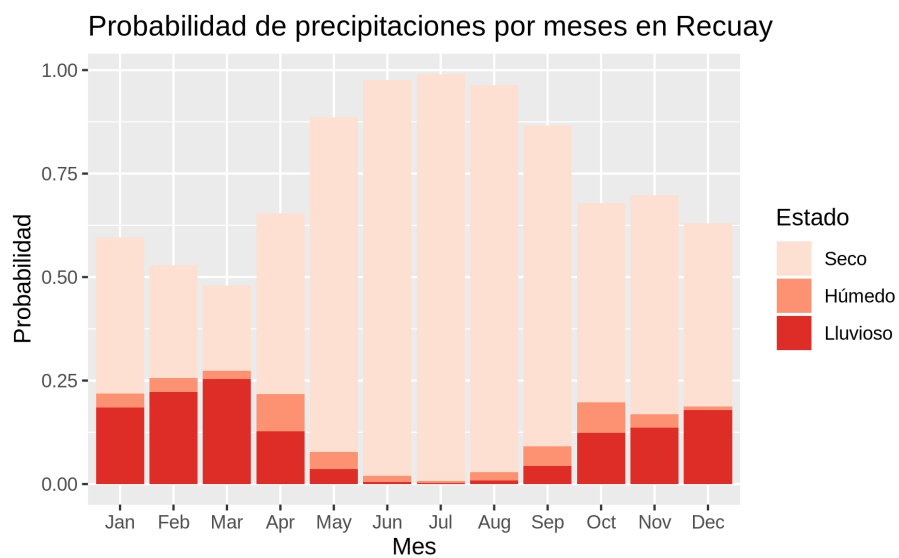
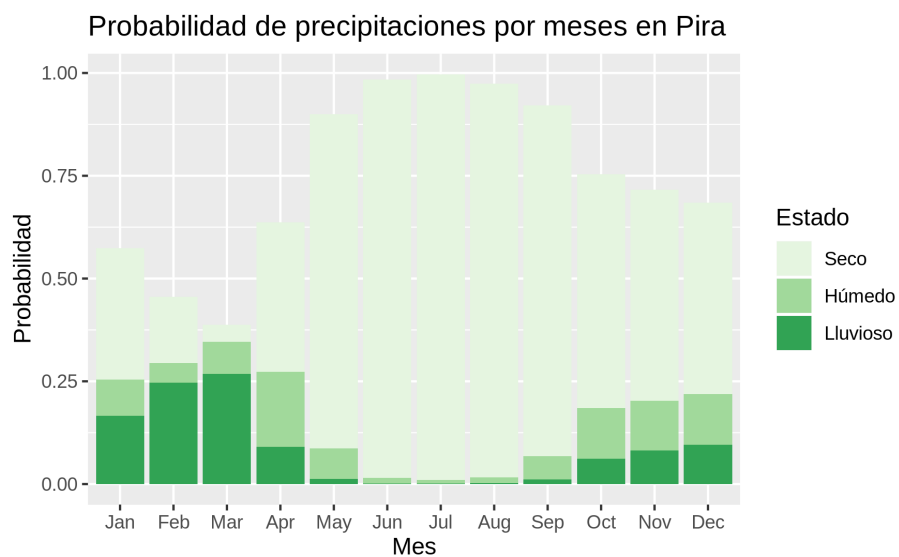
```
#Vector estabilizada de Recuay
limiting(recuay, 5)
```

```
## [1] "Iteraciones: 4"
```

```
##      Mes      Seco      Húmedo      Lluvioso
## [1,] "Jan" "0.596" "0.219" "0.185"
## [2,] "Feb" "0.528" "0.256" "0.222"
## [3,] "Mar" "0.480" "0.274" "0.254"
## [4,] "Apr" "0.654" "0.218" "0.128"
## [5,] "May" "0.886" "0.078" "0.036"
## [6,] "Jun" "0.976" "0.020" "0.005"
## [7,] "Jul" "0.990" "0.008" "0.003"
## [8,] "Aug" "0.963" "0.029" "0.009"
## [9,] "Sep" "0.866" "0.091" "0.044"
## [10,] "Oct" "0.679" "0.198" "0.124"
## [11,] "Nov" "0.697" "0.169" "0.136"
## [12,] "Dec" "0.630" "0.188" "0.179"
```

Para un mejor entendimiento de los resultados obtenidos, se realizaron las siguientes gráficas:





7 Conclusiones

- Podemos notar que Milpo es la zona donde hay mayor probabilidad de lluvias durante los meses de siembra (enero-marzo), por lo que sería conveniente sembrar en esta zona productos que requieran grandes cantidades

de recursos hídricos.

- Recuay y Pira también poseen lluvias de moderada a fuerte intensidad en los mismos meses, pero no en la misma proporción.
- Observando ambas gráficas se puede observar que los resultados obtenidos por la simulación son muy similares a los datos originales, por lo que el modelo de Márkov simuló las precipitaciones y estados futuros con éxito.

8 Referencias

- [1] Khadr, M. (2016, 03). Forecasting of meteorological drought using Hidden Markov Model (case study: The upper Blue Nile river basin, Ethiopia). *Ain Shams Engineering Journal*, 7(1), 47-56.
doi:10.1016/j.asej.2015.11.005
- [2] Tettey, M., Oduro, F. T., Adedia, D., & Abaye, D. A. (2017, 08). Markov chain analysis of the rainfall patterns of five geographical locations in the south eastern coast of Ghana. *Earth Perspectives*, 4(1).
Doi:10.1186/s40322-017-0042-6
- [3] Yoo, C., Lee, J., & Ro, Y. (2016). Markov Chain Decomposition of Monthly Rainfall into Daily Rainfall: Evaluation of Climate Change Impact. *Advances in Meteorology*, 2016, 1-10.
Doi:10.1155/2016/7957490
- [4] Yusuf, A. U. (2014). Markov Chain Model and Its Application to Annual Rainfall Distribution for Crop Production. *American Journal of Theoretical and Applied Statistics*, 3(2), 39. doi:10.11648/j.ajtas.20140302.12