Modelos para diferentes cantones

Jimena Murillo

2022-06-06

```r
library(keras) # for deep learning
library(tidyverse) # general utility functions
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(caret) # machine learning utility functions
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(tibble)
library(readr)
library(ggplot2)
library(tensorflow)
```

```
##
## Attaching package: 'tensorflow'
```

```
## The following object is masked from 'package:caret':
##
##     train
```

```r
library(neuralnet)
```

```
##
## Attaching package: 'neuralnet'

## The following object is masked from 'package:dplyr':
##
##     compute
```

Datos

```r
load("C:/Users/usuario1/Desktop/CIMPA/Github_CIMPA/PRACTICA_CIMPA/base_cantones.RData")


basecanton = basecanton  %>%

  dplyr::select(Canton, Year,Month,Nino12SSTA, Nino3SSTA, Nino4SSTA,Nino34SSTA,Nino34SSTA1, Nino34SSTA2

  arrange(Canton,Year,Month) %>% ungroup() %>% mutate(Month=as.numeric(Month))

#Funciones

normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

denorm <- function(x,base) {
  return (x*(max(base$RR) - min(base$RR))+min(base$RR))
}

metricas <- function(tabla){
  NRMSE <- mean((tabla$y_pred-tabla$y)^2)/mean(tabla$y)
  NIS_95 <- mean((tabla$e_u_upper-tabla$e_u_lower)+
                   (2/0.05)*(tabla$e_u_lower-tabla$y)*(tabla$y<tabla$e_u_lower)+
                   (2/0.05)*(tabla$y-tabla$e_u_upper)*(tabla$y>tabla$e_u_upper))/mean(tabla$y)
  return(data.frame(NRMSE,NIS_95))
}




basecanton2 = basecanton %>% group_by(basecanton$Canton) %>%
  mutate_if(is.numeric, normalize)
```

```
## `mutate_if()` ignored the following grouping variables:
## * Column `basecanton$Canton`
```

```r
basecanton2 = basecanton2[,-35]
```

```r
#Train y test

data_train = as.data.frame(basecanton2) %>% filter(Year < 1)#PARA ENTRENAR HASTA 2018
data_test = as.data.frame(basecanton2) %>% filter(Year >= 1)

X_train = data_train[,-ncol(data_train)]
y_train = as.data.frame(data_train[,c("Canton","RR")])

X_test = as.data.frame(data_test[,-ncol(data_test)])
y_test = as.data.frame(data_test[,c("Canton","RR")])



Fecha = paste(basecanton$Year, basecanton$Month)
Fecha = Fecha[1:235]
```

Arquitectura y programación del modelo

Generar un Wrapper para el learning dropout

```r
model.gen = function(X_train, y_train, X_test, X_all, RR) {

  # R6 wrapper class, a subclass of KerasWrapper
  ConcreteDropout <- R6::R6Class("ConcreteDropout",

    inherit = KerasWrapper,

    public = list(
      weight_regularizer = NULL,
      dropout_regularizer = NULL,
      init_min = NULL,
      init_max = NULL,
      is_mc_dropout = NULL,
      supports_masking = TRUE,
      p_logit = NULL,
      p = NULL,

      initialize = function(weight_regularizer,
                            dropout_regularizer,
                            init_min,
                            init_max,
                            is_mc_dropout) {
        self$weight_regularizer <- weight_regularizer
        self$dropout_regularizer <- dropout_regularizer
        self$is_mc_dropout <- is_mc_dropout
        self$init_min <- k_log(init_min) - k_log(1 - init_min)
        self$init_max <- k_log(init_max) - k_log(1 - init_max)
      },

      build = function(input_shape) {
```

```
    super$build(input_shape)

    self$p_logit <- super$add_weight(
      name = "p_logit",
      shape = shape(1),
      initializer = initializer_random_uniform(self$init_min, self$init_max),
      trainable = TRUE
    )

    self$p <- k_sigmoid(self$p_logit)

    input_dim <- input_shape[[2]]

    weight <- private$py_wrapper$layer$kernel

    kernel_regularizer <- self$weight_regularizer *
                          k_sum(k_square(weight)) /
                          (1 - self$p)

    dropout_regularizer <- self$p * k_log(self$p)
    dropout_regularizer <- dropout_regularizer +
                           (1 - self$p) * k_log(1 - self$p)
    dropout_regularizer <- dropout_regularizer *
                           self$dropout_regularizer *
                           k_cast(input_dim, k_floatx())

    regularizer <- k_sum(kernel_regularizer + dropout_regularizer)
    super$add_loss(regularizer)
  },

  concrete_dropout = function(x) {
    eps <- k_cast_to_floatx(k_epsilon())
    temp <- 0.1

    unif_noise <- k_random_uniform(shape = k_shape(x))

    drop_prob <- k_log(self$p + eps) -
                 k_log(1 - self$p + eps) +
                 k_log(unif_noise + eps) -
                 k_log(1 - unif_noise + eps)
    drop_prob <- k_sigmoid(drop_prob / temp)

    random_tensor <- 1 - drop_prob

    retain_prob <- 1 - self$p
    x <- x * random_tensor
    x <- x / retain_prob
    x
  },

  call = function(x, mask = NULL, training = NULL) {
    if (self$is_mc_dropout) {
      super$call(self$concrete_dropout(x))
```

```r
      } else {
        k_in_train_phase(
          function()
            super$call(self$concrete_dropout(x)),
          super$call(x),
          training = training
        )
      }
    }
  )
)

# function for instantiating custom wrapper
layer_concrete_dropout <- function(object,
                                   layer,
                                   weight_regularizer = 1e-6,
                                   dropout_regularizer = 1e-5,
                                   init_min = 0.1,
                                   init_max = 0.1,
                                   is_mc_dropout = TRUE,
                                   name = NULL,
                                   trainable = TRUE) {
  create_wrapper(ConcreteDropout, object, list(
    layer = layer,
    weight_regularizer = weight_regularizer,
    dropout_regularizer = dropout_regularizer,
    init_min = init_min,
    init_max = init_max,
    is_mc_dropout = is_mc_dropout,
    name = name,
    trainable = trainable
  ))
}

# sample size (training data)
n_train <- 232
# sample size (validation data)
n_val <- 3
# prior length-scale
l <- 4e-3
# initial value for weight regularizer
wd <- l^2/232
# initial value for dropout regularizer
dd <- 2/3

# Arquitectura del modelo


input_dim <- 32
output_dim <- 1

input <- layer_input(shape = input_dim)
```

```r
output <- input %>% layer_concrete_dropout(
layer = layer_dense(units = 100, activation = "relu"),
weight_regularizer = wd,
dropout_regularizer = dd
) %>% layer_dense(units = 50, activation = "relu") %>%
layer_concrete_dropout(
layer = layer_dense(units = 50, activation = "relu"),
weight_regularizer = wd,
dropout_regularizer = dd
) %>% layer_concrete_dropout(
layer = layer_dense(units = 50, activation = "relu"),
weight_regularizer = wd,
dropout_regularizer = dd
) %>% layer_concrete_dropout(
layer = layer_dense(units = 50, activation = "relu"),
weight_regularizer = wd,
dropout_regularizer = dd
) %>% layer_dense(units = 25, activation = "relu") %>%
layer_concrete_dropout(
layer = layer_dense(units = 25, activation = "relu"),
weight_regularizer = wd,
dropout_regularizer = dd
) %>% layer_concrete_dropout(
layer = layer_dense(units = 25, activation = "relu"),
weight_regularizer = wd,
dropout_regularizer = dd
) %>% layer_concrete_dropout(
layer = layer_dense(units = 25, activation = "relu"),
weight_regularizer = wd,
dropout_regularizer = dd
) %>% layer_dense(units = 12, activation = "relu") %>%
layer_concrete_dropout(
layer = layer_dense(units = 12, activation = "relu"),
weight_regularizer = wd,
dropout_regularizer = dd
) %>% layer_concrete_dropout(
layer = layer_dense(units = 12, activation = "relu"),
weight_regularizer = wd,
dropout_regularizer = dd
) %>% layer_dense(units = 6, activation = "relu") %>%
layer_concrete_dropout(
layer = layer_dense(units = 6, activation = "relu"),
weight_regularizer = wd,
dropout_regularizer = dd
) %>% layer_concrete_dropout(
layer = layer_dense(units = 6, activation = "relu"),
weight_regularizer = wd,
dropout_regularizer = dd
)


## Loss function
heteroscedastic_loss <- function(y_true, y_pred) {
```

```r
  mean <- y_pred[, 1:output_dim]
  log_var <- y_pred[, (output_dim + 1):(output_dim * 2)]
  precision <- k_exp(-log_var)
  k_sum(precision * (y_true - mean) ^ 2 + log_var, axis = 2)
}


## Output del Modelo


mean <- output %>% layer_concrete_dropout(
  layer = layer_dense(units = output_dim),
  weight_regularizer = wd,
  dropout_regularizer = dd
)

log_var <- output %>% layer_concrete_dropout(
  layer_dense(units = output_dim),
  weight_regularizer = wd,
  dropout_regularizer = dd
)

output <- layer_concatenate(list(mean, log_var))


model <- keras_model(input, output)



model %>% compile(
optimizer = "adam",
loss = "mse",
metrics = c(custom_metric("heteroscedastic_loss", heteroscedastic_loss)))



history <- model %>% fit(
  X_train,
  y_train,
  epochs = 50,
  batch_size = 18,
  validation_split = 0.1,
  shuffle = F
)


## MonteCarlo sampling


denorm <- function(x , base) {
  return (x*(max(base$RR) - min(base$RR))+min(base$RR))
}
```

```
  num_MC_samples <- 300

  samples = list()

  MC_samples.pd <- array(0, dim = c(num_MC_samples, nrow(X_test), 2 * output_dim))
  for (k in 1:num_MC_samples) {
    MC_samples.pd[k, , ] <- denorm((model %>% predict(X_test)),base)
  }

  MC_samples.tot <- array(0, dim = c(num_MC_samples, nrow(X_all), 2 * output_dim))
  for (k in 1:num_MC_samples) {
    MC_samples.tot[k, , ] <- denorm((model %>% predict(X_all)),base)
  }


  samples[[1]] <- MC_samples.pd
  samples[[2]] <- MC_samples.tot

return (samples)

}
```

Entrenamiento de modelos para cada cantón

Entrenar al modelo y predecir

```
Cantones = unique(basecanton$Canton)
Eval.pd = matrix(NA, ncol = 2, nrow = length(Cantones))
Eval.tot = matrix(NA, ncol = 2, nrow = length(Cantones))


p1 = list()
p2 = list()

Predicciones = matrix(NA, ncol = 4, nrow = 3*length(Cantones))
Index = seq(1,3*length(Cantones), 3)

for (i in 1:length(Cantones)) {

  X_trainc = X_train %>% filter(Canton == Cantones[i])
  X_trainc = as.matrix(X_trainc[,-1])
  y_trainc = y_train %>% filter(Canton == Cantones[i])
  y_trainc = as.matrix(y_trainc[,-1])

  X_testc = X_test %>% filter(Canton == Cantones[i])
  X_testc = as.matrix(X_testc[,-1])
  y_testc = y_test %>% filter(Canton == Cantones[i])
  y_testc = as.matrix(y_testc[,-1])

  X_all = basecanton2 %>% filter(Canton == Cantones[i])
  X_all = as.matrix(X_all[,-c(1,33)])
```

```r
base = as.data.frame(basecanton %>% filter(Canton == Cantones[i]) %>% dplyr::select(RR))

samples = list()
samples = model.gen(X_trainc, y_trainc, X_testc, X_all, base)

## Generar intervalo de confianza

output_dim  = 1

MC_samples.pd = samples[[1]]


means = NULL
means <- MC_samples.pd[, , 1:output_dim]

predictive_mean <- apply(means, 2, mean)

epistemic_uncertainty <- apply(means, 2, var)

logvar = NULL
logvar <- MC_samples.pd[, , (output_dim + 1):(output_dim * 2)]
aleatoric_uncertainty <- exp(colMeans(logvar))


y_testc = denorm(y_testc, base)


df1 <- data.frame(
  x = Fecha[(236-nrow(X_testc)):235],
  y = y_testc,
  y_pred = predictive_mean,
  e_u_lower = predictive_mean - sqrt(epistemic_uncertainty),
  e_u_upper = predictive_mean + sqrt(epistemic_uncertainty),
  a_u_lower = predictive_mean - sqrt(aleatoric_uncertainty),
  a_u_upper = predictive_mean + sqrt(aleatoric_uncertainty),
  u_overall_lower = predictive_mean -
                  sqrt(epistemic_uncertainty) -
                  sqrt(aleatoric_uncertainty),
  u_overall_upper = predictive_mean +
                  sqrt(epistemic_uncertainty) +
                  sqrt(aleatoric_uncertainty)
)


Eval.pd[i,1:2] = as.numeric(metricas(df1))


p1[[i]] = ggplot(df1, aes(x = x, y = y, group = 1)) + geom_line(colour = "blue") +
geom_line( aes(x = x, y = y_pred, colour = "red"))+
theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
panel.background = element_blank(), axis.text.x = element_text(angle = 45), legend.position = "none" )
labs (x = "Fecha", y = "Riesgo Relativo") +
```

```
ggtitle(paste("Predicciones 2021 del cantón", Cantones[i], sep = " "))+
geom_ribbon(aes(ymin = e_u_lower, ymax = e_u_upper), alpha = 0.3)

Predicciones[Index[i]:(Index[i]+2),1:4] = cbind(Cantones[i], df1$e_u_lower, df1$y_pred, df1$e_u_upper

#### VALORES APROXIMADOS ####

## Generar valores ajustados


MC_samples.tot = samples[[2]]

means = NULL
means <- MC_samples.tot[, , 1:output_dim]

# average over the MC samples
predictive_mean <- apply(means, 2, mean)

epistemic_uncertainty <- apply(means, 2, var)


logvar = NULL
logvar <- MC_samples.tot[, , (output_dim + 1):(output_dim * 2)]
aleatoric_uncertainty <- exp(colMeans(logvar))


df2 <- data.frame(
  x = Fecha,
  y = base$RR,
  y_pred = predictive_mean,
  e_u_lower = predictive_mean - sqrt(epistemic_uncertainty),
  e_u_upper = predictive_mean + sqrt(epistemic_uncertainty),
  a_u_lower = predictive_mean - sqrt(aleatoric_uncertainty),
  a_u_upper = predictive_mean + sqrt(aleatoric_uncertainty),
  u_overall_lower = predictive_mean -
                  sqrt(epistemic_uncertainty) -
                  sqrt(aleatoric_uncertainty),
  u_overall_upper = predictive_mean +
                  sqrt(epistemic_uncertainty) +
                  sqrt(aleatoric_uncertainty)
)




Eval.tot[i,1:2] = as.numeric(metricas(df2))

everyother1 <- function(x) x[(seq_along(Fecha) + 5)%%12 == 6]

p2[[i]] = ggplot(df2, aes(x = x, y = y, group = 1)) + geom_line(colour = "blue") +
geom_line(aes(x = x, y = y_pred, colour = "red"))+
theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
panel.background = element_blank(), axis.text.x = element_text(angle = 45), legend.position = "none" )
```

```
  scale_x_discrete(breaks = everyother1) + labs (x = "Fecha", y = "Riesgo Relativo") +
  ggtitle(paste("Valores aproximados de training del cantón", Cantones[i], sep = " "))+
  geom_ribbon(aes(ymin = e_u_lower, ymax = e_u_upper), alpha = 0.3)


}
```

```
## Loaded Tensorflow version 2.8.0
```

Resultados de métricas

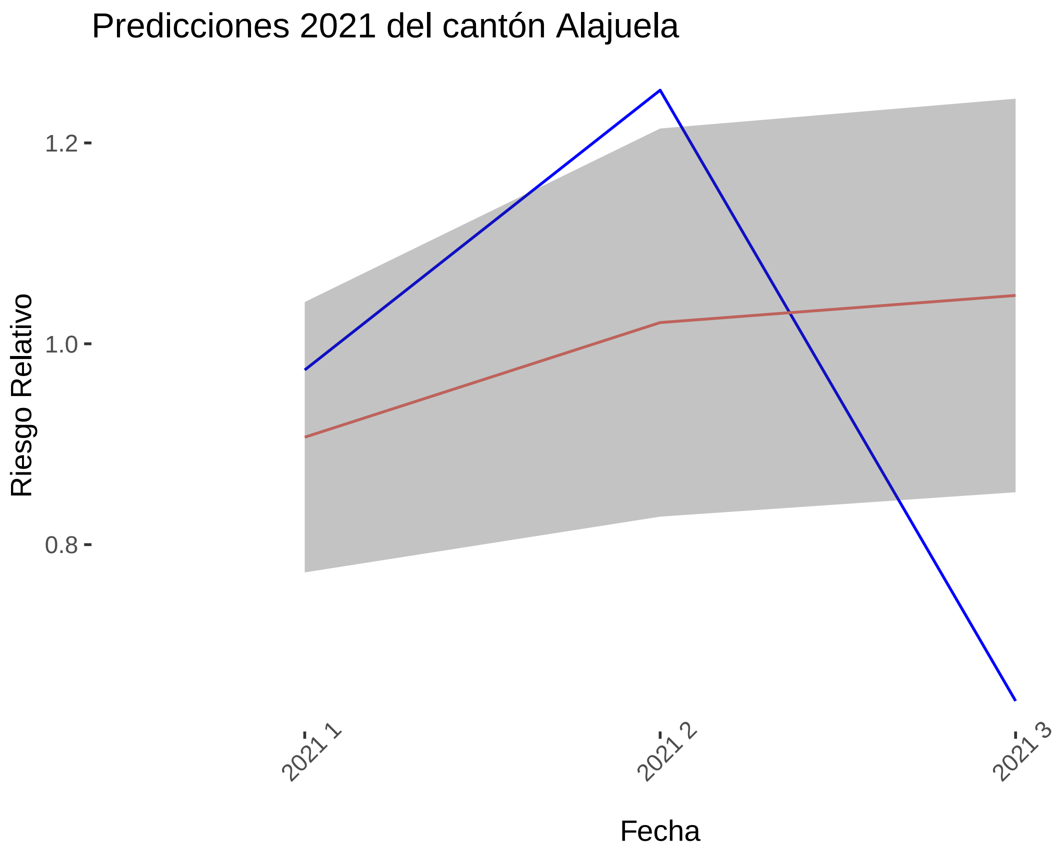
```
Metricas = cbind (Eval.pd, Eval.tot)
colnames(Metricas) = c("NMRSE 2021", "NIS 2021", "NMRSE total", "NIS total")
rownames(Metricas) = Cantones
as.data.frame(Metricas)
```

```
##                 NMRSE 2021   NIS 2021 NMRSE total NIS total
## Alajuela        0.07700838   3.792431  0.41506884 11.332765
## Alajuelita      0.37845445  19.937712  0.24291951 25.992279
## Atenas         11.15310395  78.323969  1.85833296 10.178058
## Cañas           5.65632566  98.120385  4.47416652 19.530497
## Carrillo        3.74507748  59.020308  5.07197507 31.446554
## Corredores      0.89349261  20.300746  6.25552322 38.553850
## Desamparados    0.01412810   5.375914  0.06065042 10.823176
## Esparza         1.82977690  33.742844  2.41044705 18.412752
## Garabito       15.19368190  45.308308  3.28462408  6.326925
## Golfito         2.24639458  55.572411  3.85603386 33.126683
## Guacimo         1.46293397   7.971516  1.01980503  9.465005
## La Cruz         7.30587928 227.332035  6.90897150 40.897075
## Liberia         6.39888950 154.885764  1.68363976 16.844782
## Limon           2.69236901  24.685619  1.54386799 14.797120
## Matina          1.65345428  16.701676  2.12245225  9.232764
## Montes de Oro  27.20805072 298.178014  5.55353905 31.250456
## Nicoya          5.88006476 202.749120  0.90185122 11.678795
## Orotina         1.21381829  17.408114 10.17743911 33.390120
## Osa             1.62620494  25.120153  2.27360333 20.467316
## Parrita        39.00736488 211.899209 14.90248030 25.573467
## Perez Zeledón   1.79809026  31.077671  2.21732951 38.284071
## Pococí          0.64725608  19.375353  1.10985604 12.735356
## Puntarenas      0.98455250  31.021366  0.62769581  7.921806
## Quepos         23.88118335 286.529271  5.41376551 15.018459
## San Jose        0.03786127   3.656812  0.08899379  8.791889
## Santa Ana       0.49359244  58.147737  0.85009888 30.794507
## SantaCruz      37.08166218 320.183274 12.98368026 37.897510
## Sarapiquí       1.17852829  23.392374  9.91152686 37.678511
## Siquirres       0.17424912   1.813297  1.81229064 15.199974
## Talamanca      11.42127874  27.386765  4.64633974 21.671513
## Turrialba       1.03288544  26.765707  2.70994507 36.935814
## Upala                  Inf        Inf  0.97685076 28.825972
```

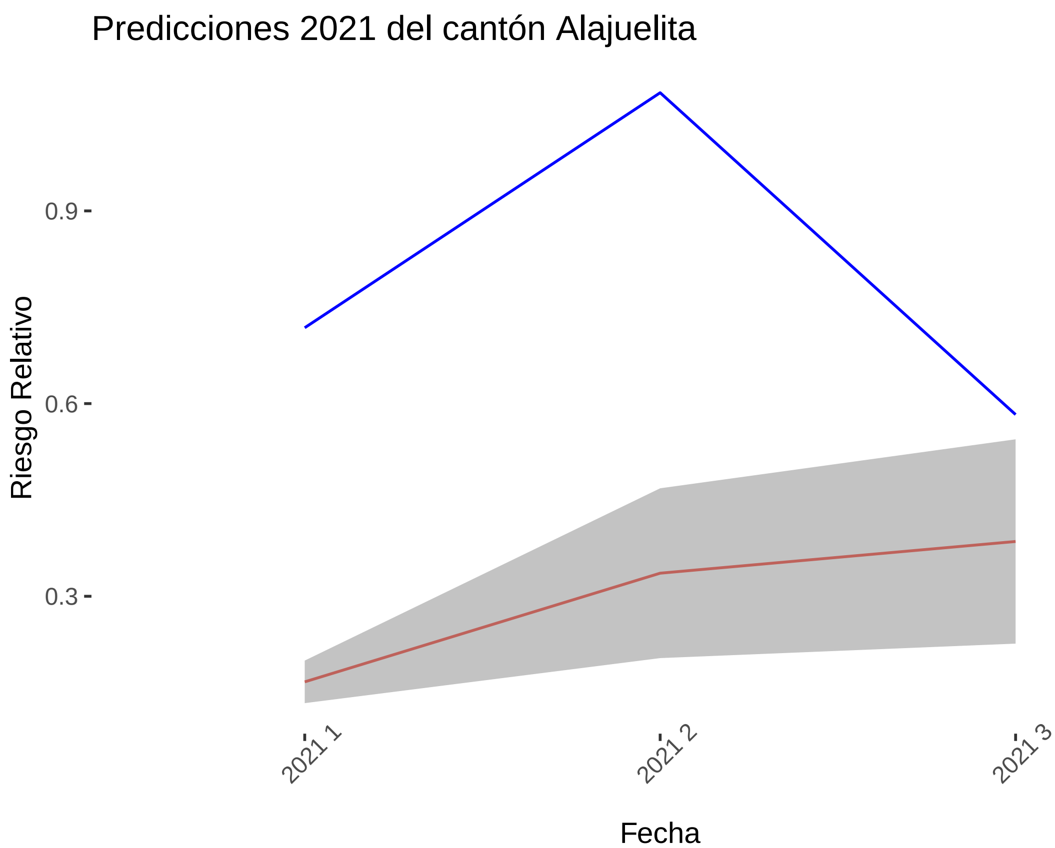
Gráficos

```
p1
```

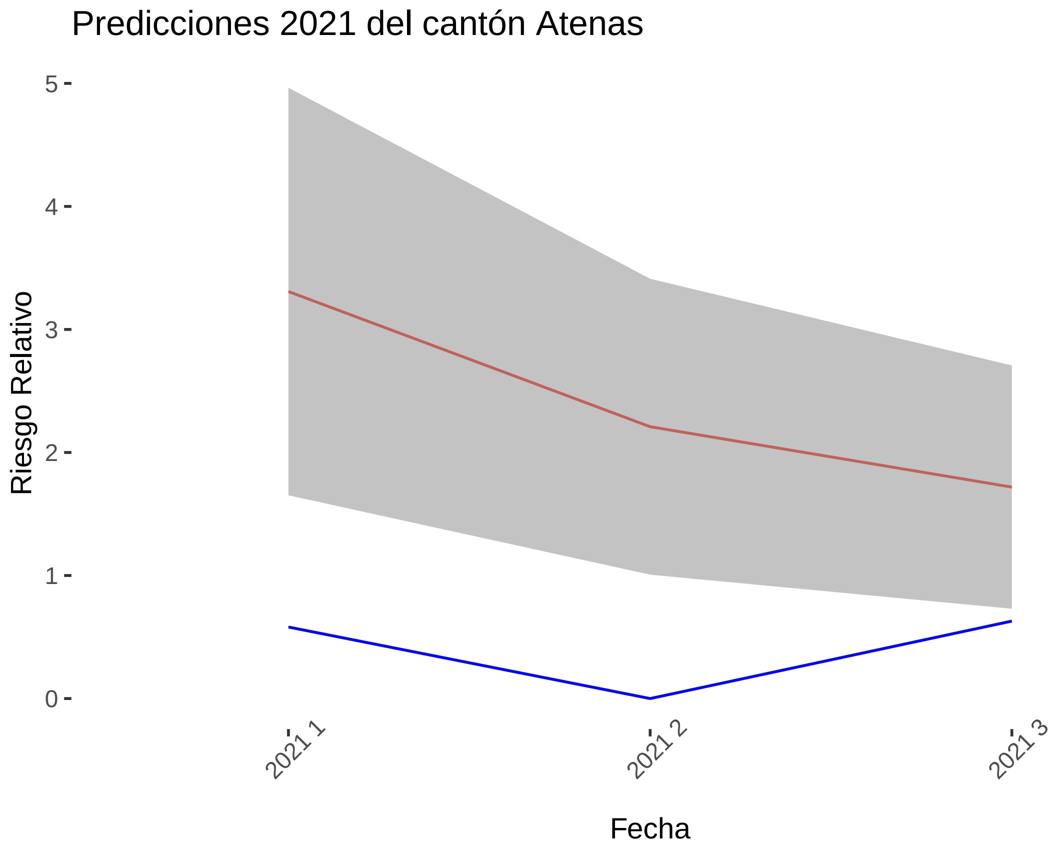
```
## [[1]]
```



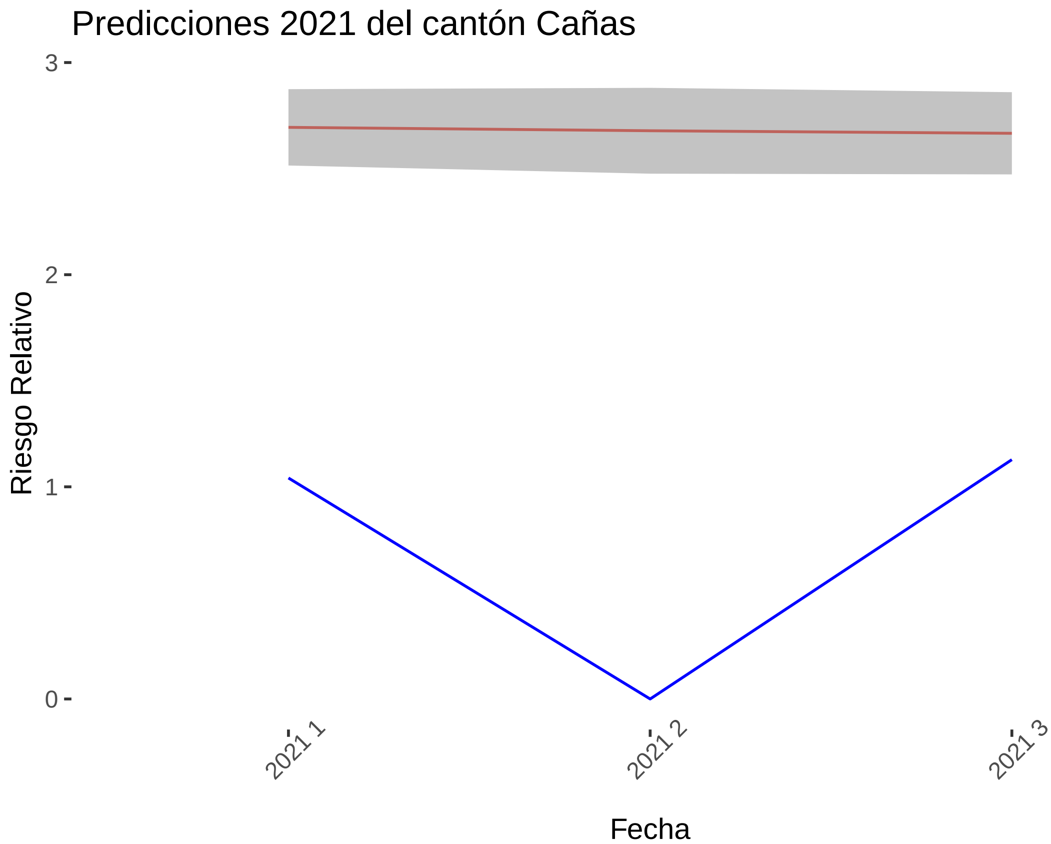

```
##
## [[2]]
```


Predicciones 2021 del cantón Alajuelita
0.9
0.6
0.3
Riesgo Relativo
2021 1
2021 2
2021 3
Fecha

```
##
## [[3]]
```

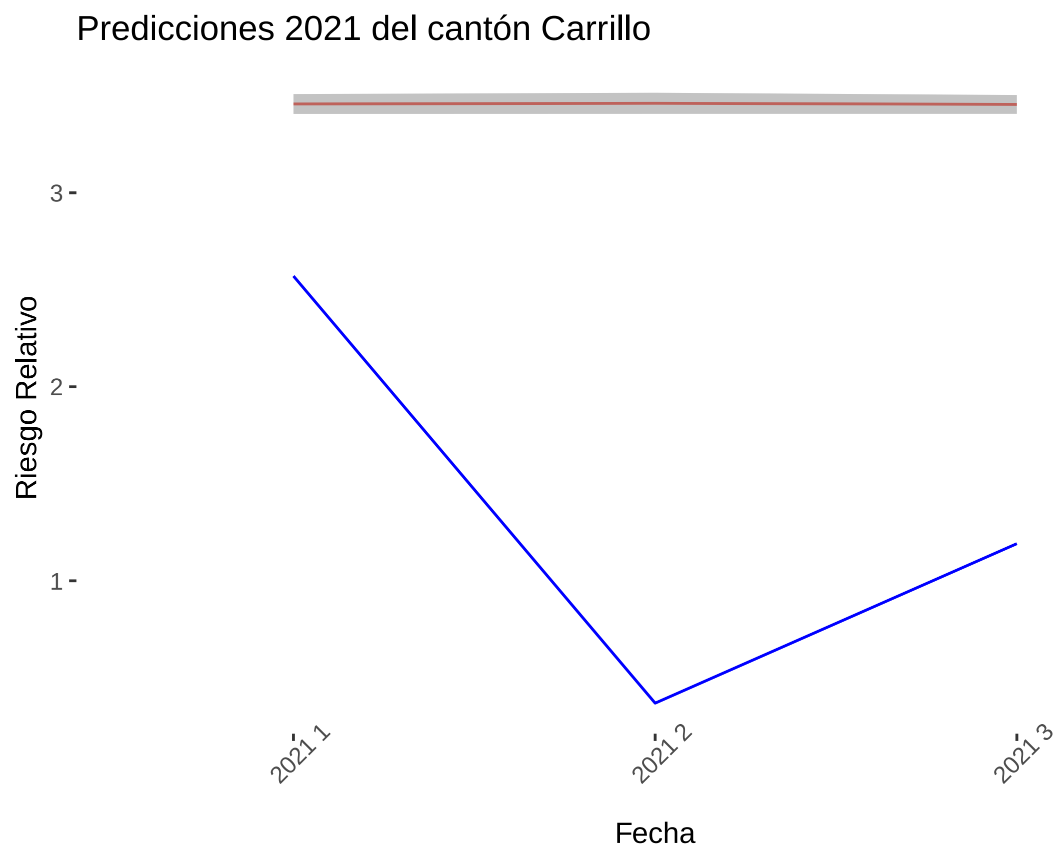

Predicciones 2021 del cantón Atenas
Riesgo Relativo
5
4
3
2
1
0
2021 1
2021 2
2021 3
Fecha

```
##
## [[4]]
```

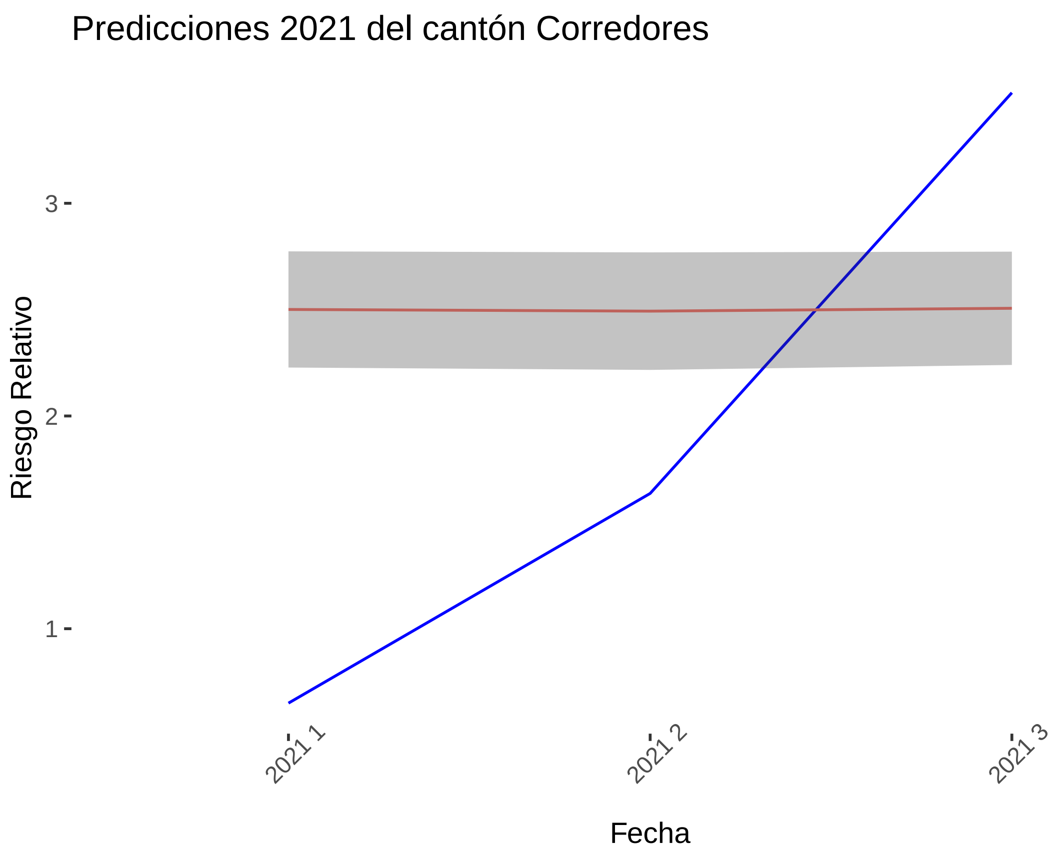


Predicciones 2021 del cantón Cañas
Riesgo Relativo
3
2
1
0
2021 1
2021 2
2021 3
Fecha

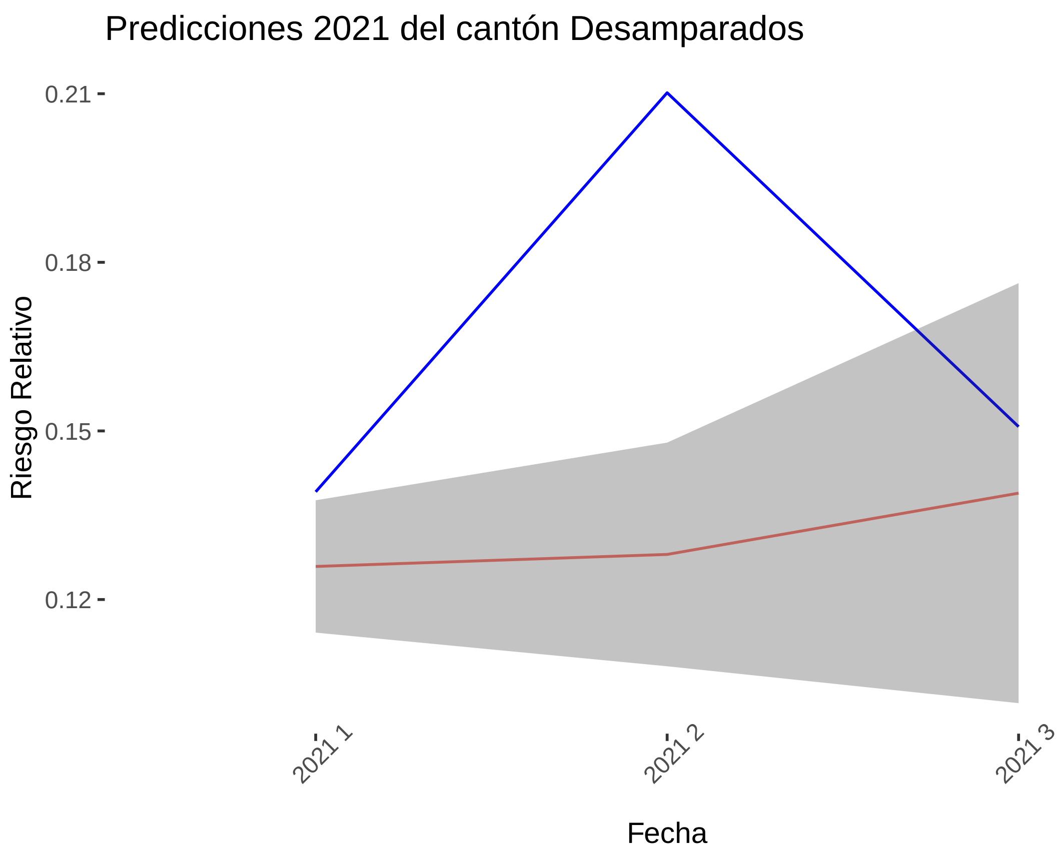

```
##
## [[5]]
```


Predicciones 2021 del cantón Carrillo
Riesgo Relativo
3
2
1
2021 1
2021 2
2021 3
Fecha

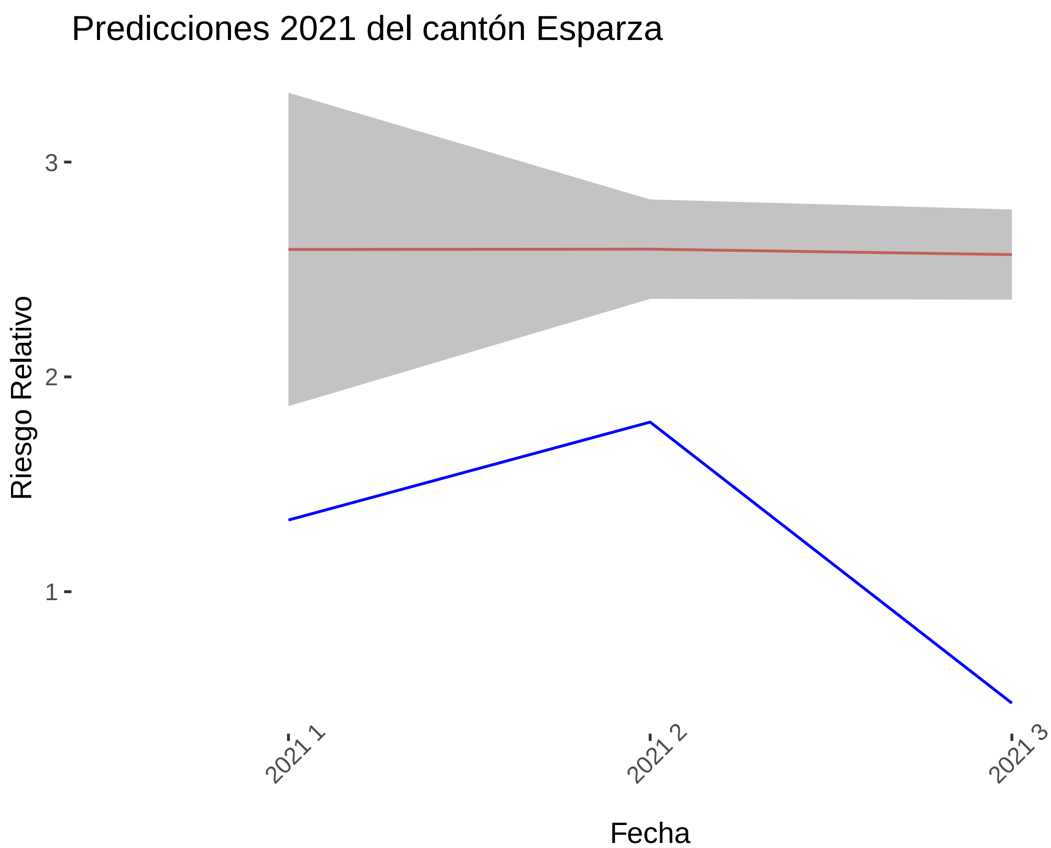
```
##
## [[6]]
```


Predicciones 2021 del cantón Corredores
Riesgo Relativo
3
2
1
2021 1
2021 2
2021 3
Fecha

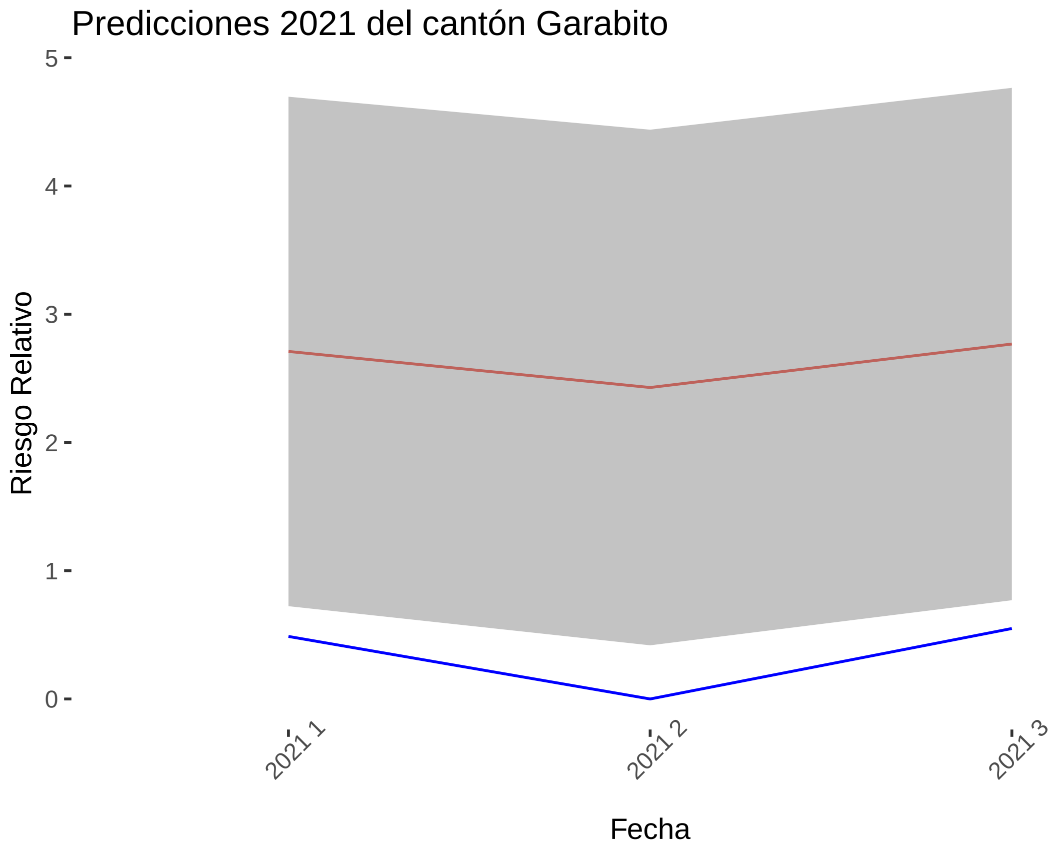
```
##
## [[7]]
```


Predicciones 2021 del cantón Desamparados
0.21
0.18
0.15
0.12
Riesgo Relativo
2021 1
2021 2
2021 3
Fecha

```
##
## [[8]]
```

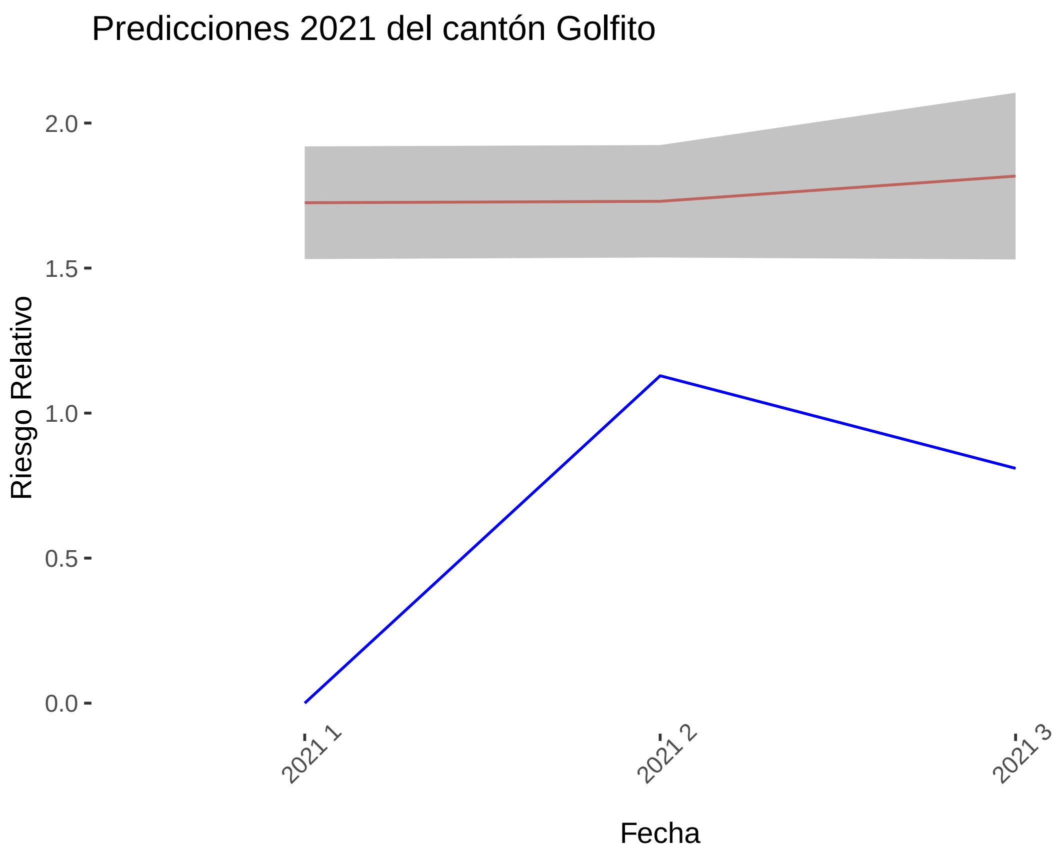

Predicciones 2021 del cantón Esparza
3
2
1
Riesgo Relativo
2021 1
2021 2
2021 3
Fecha

```
##
## [[9]]
```

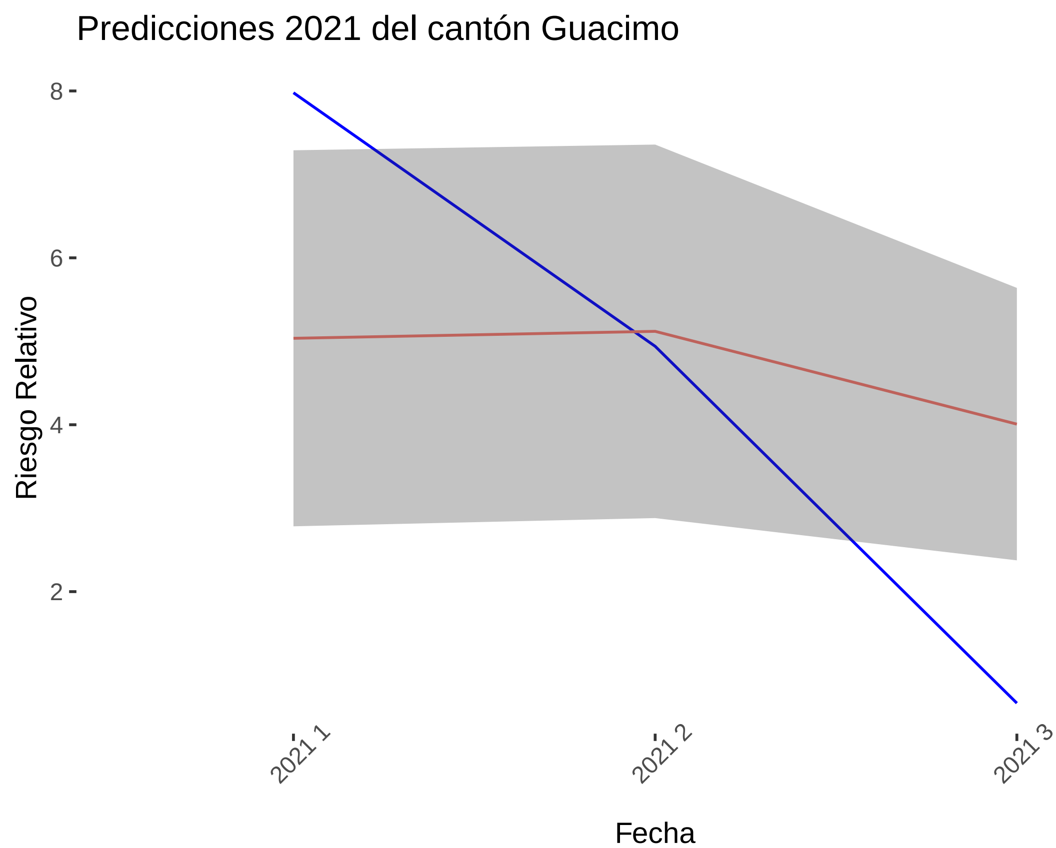


Predicciones 2021 del cantón Garabito
Riesgo Relativo
0
1
2
3
4
5
2021 1
2021 2
2021 3
Fecha

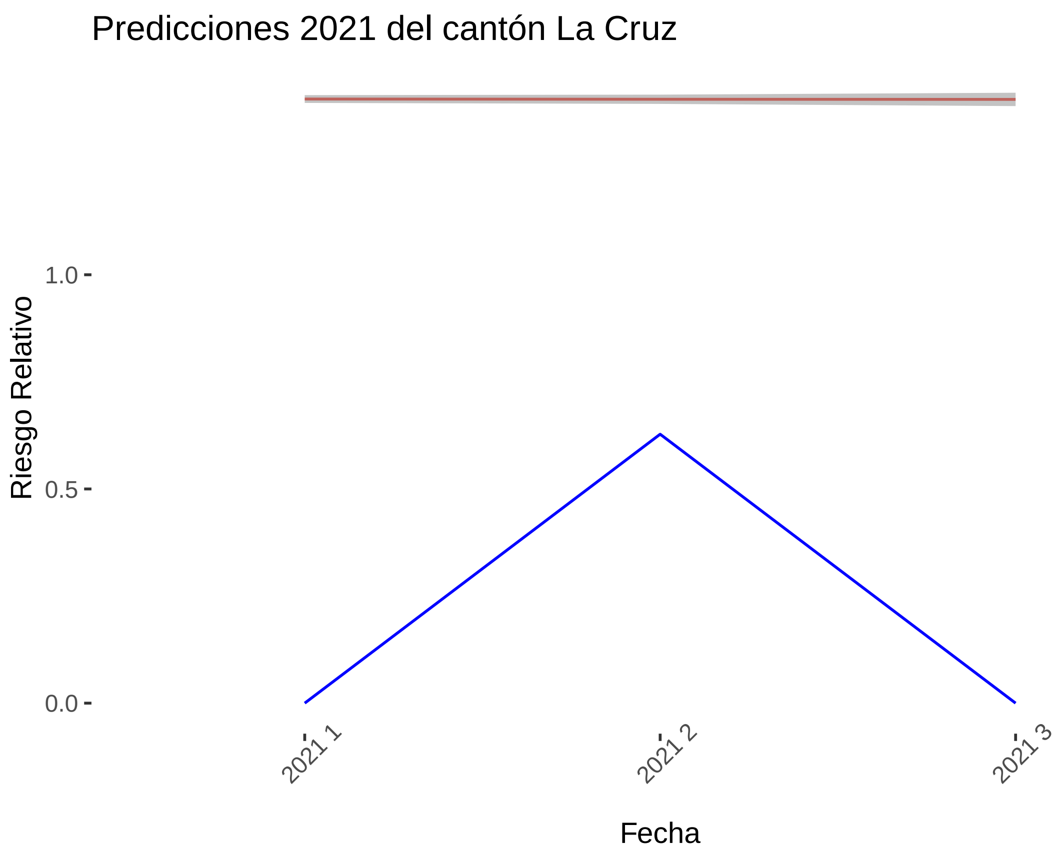

```
##
## [[10]]
```


Predicciones 2021 del cantón Golfito
2.0
1.5
1.0
0.5
0.0
Riesgo Relativo
2021 1
2021 2
2021 3
Fecha

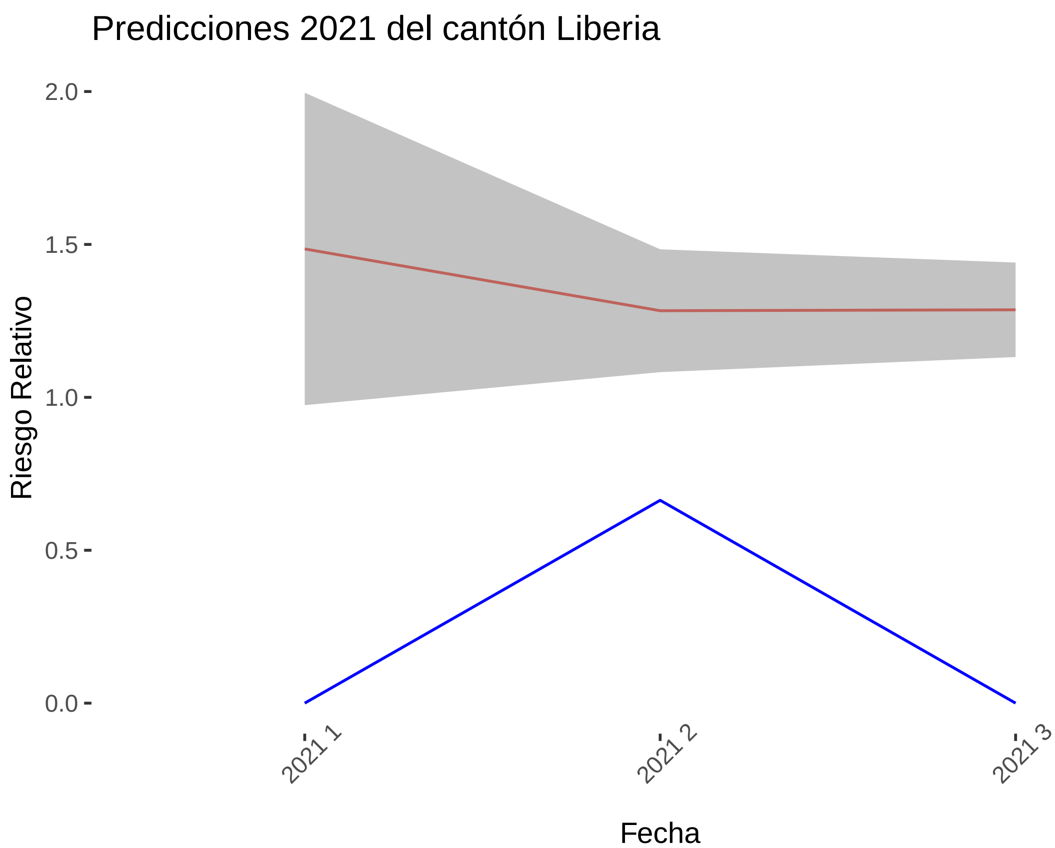
```
##
## [[11]]
```


Predicciones 2021 del cantón Guacimo
8
6
4
2
Riesgo Relativo
2021 1
2021 2
2021 3
Fecha

```
##
## [[12]]
```

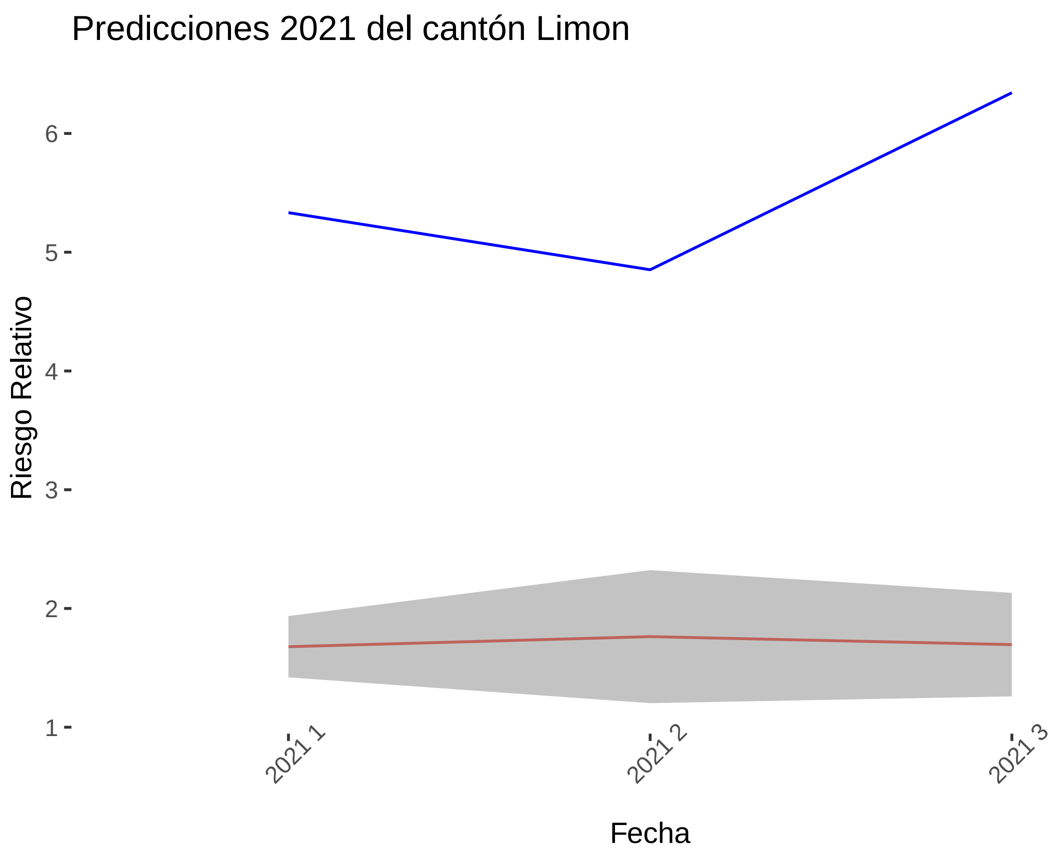

Predicciones 2021 del cantón La Cruz
1.0
0.5
0.0
Riesgo Relativo
2021 1
2021 2
2021 3
Fecha

```
##
## [[13]]
```

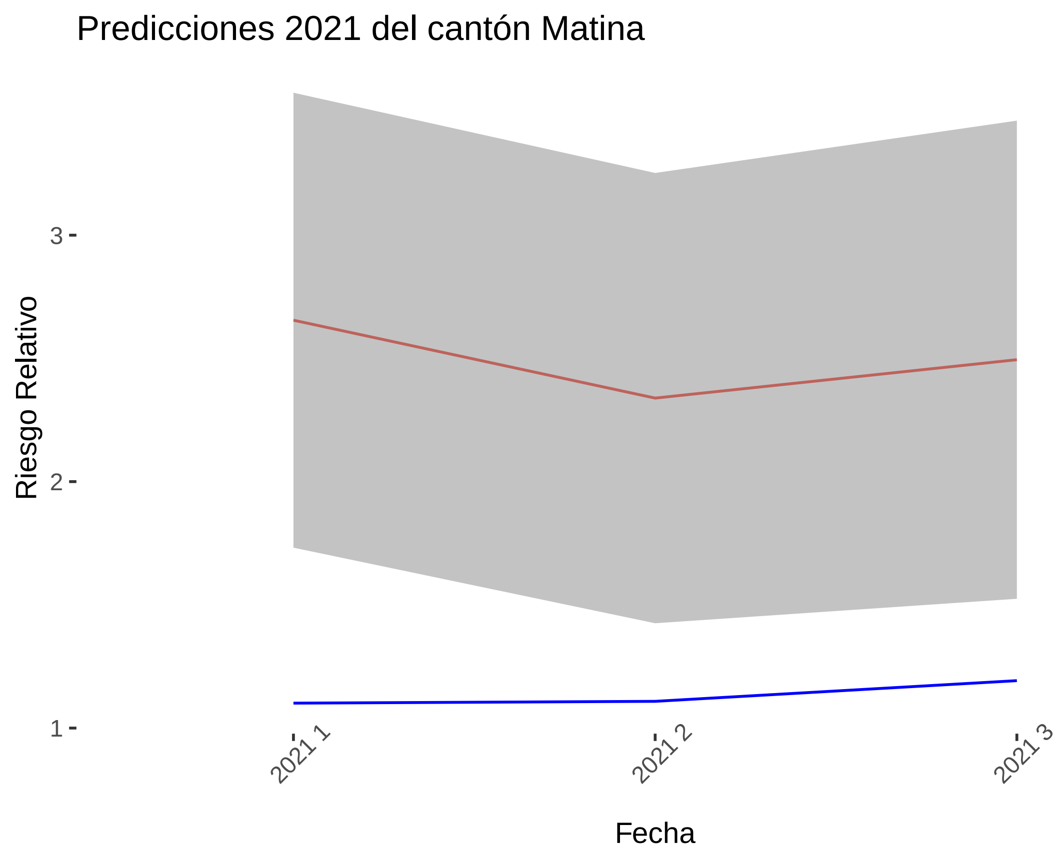


Predicciones 2021 del cantón Liberia
2.0
1.5
1.0
0.5
0.0
Riesgo Relativo
2021 1
2021 2
2021 3
Fecha


```
##
## [[14]]
```

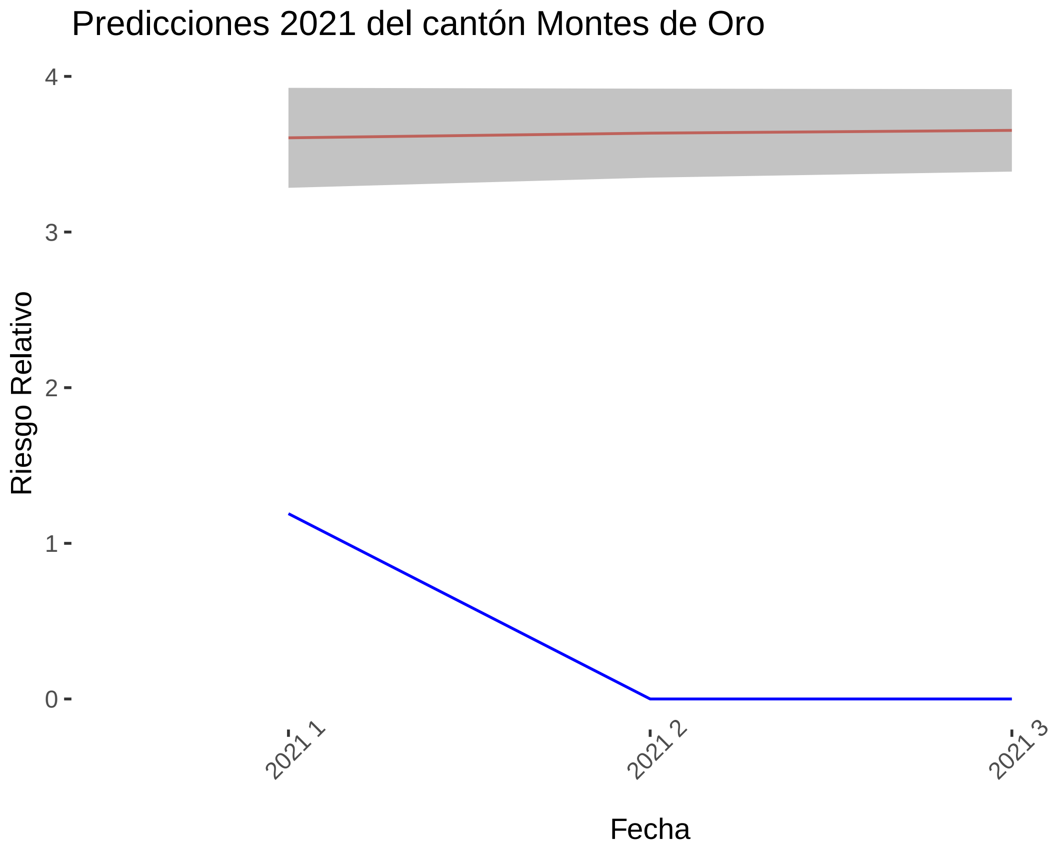

Predicciones 2021 del cantón Limon
Riesgo Relativo
6
5
4
3
2
1
2021 1
2021 2
2021 3
Fecha

```
##
## [[15]]
```



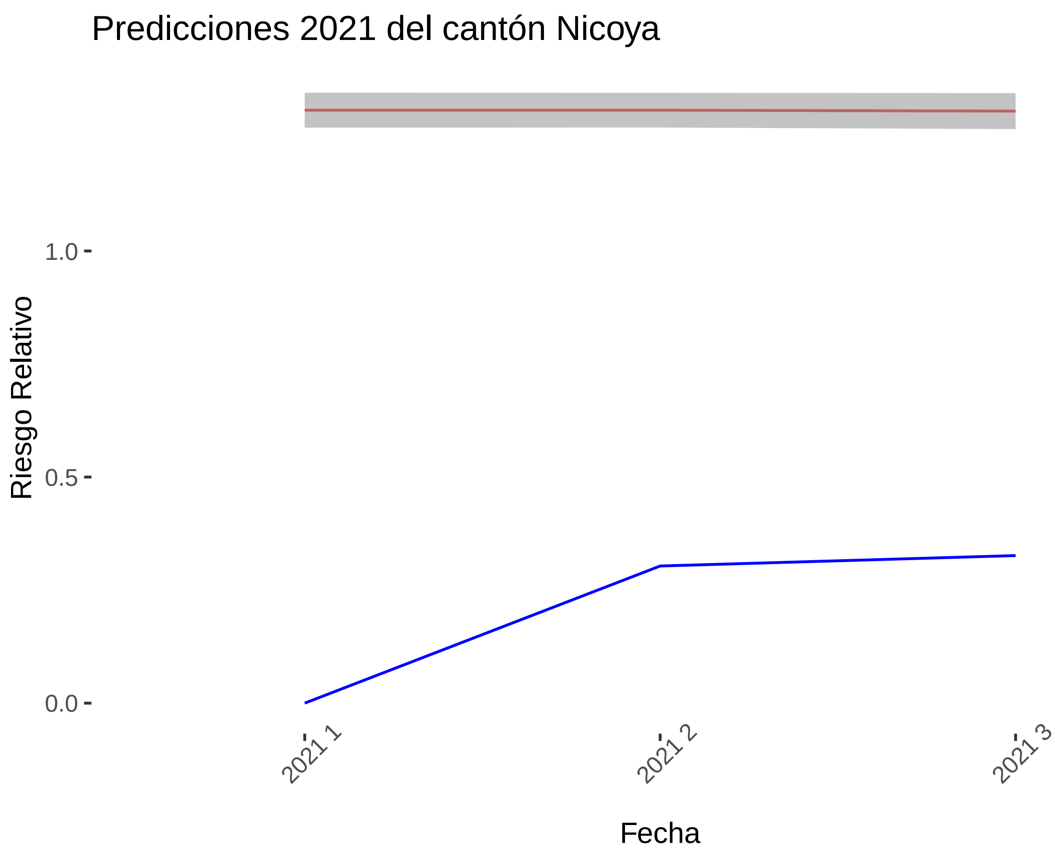
Predicciones 2021 del cantón Matina
Riesgo Relativo
3
2
1
2021 1
2021 2
2021 3
Fecha


```
##
## [[16]]
```

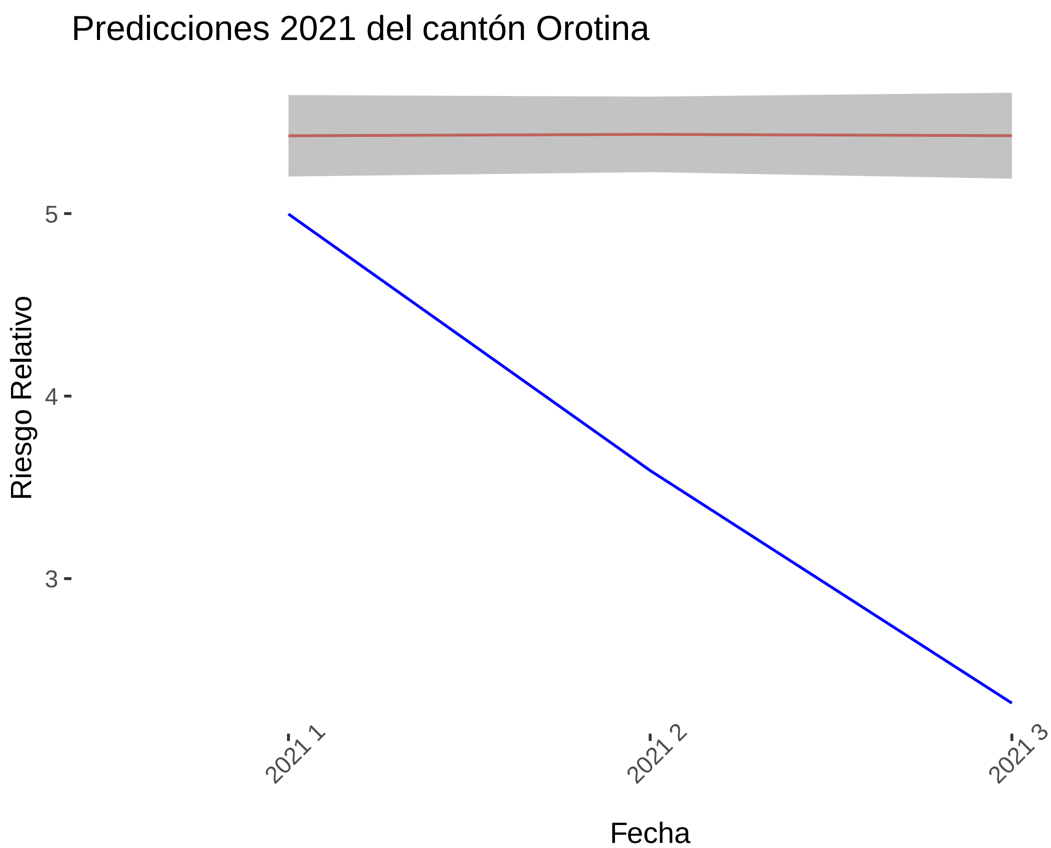


Predicciones 2021 del cantón Montes de Oro
Riesgo Relativo
4
3
2
1
0
2021 1
2021 2
2021 3
Fecha


```
##
## [[17]]
```

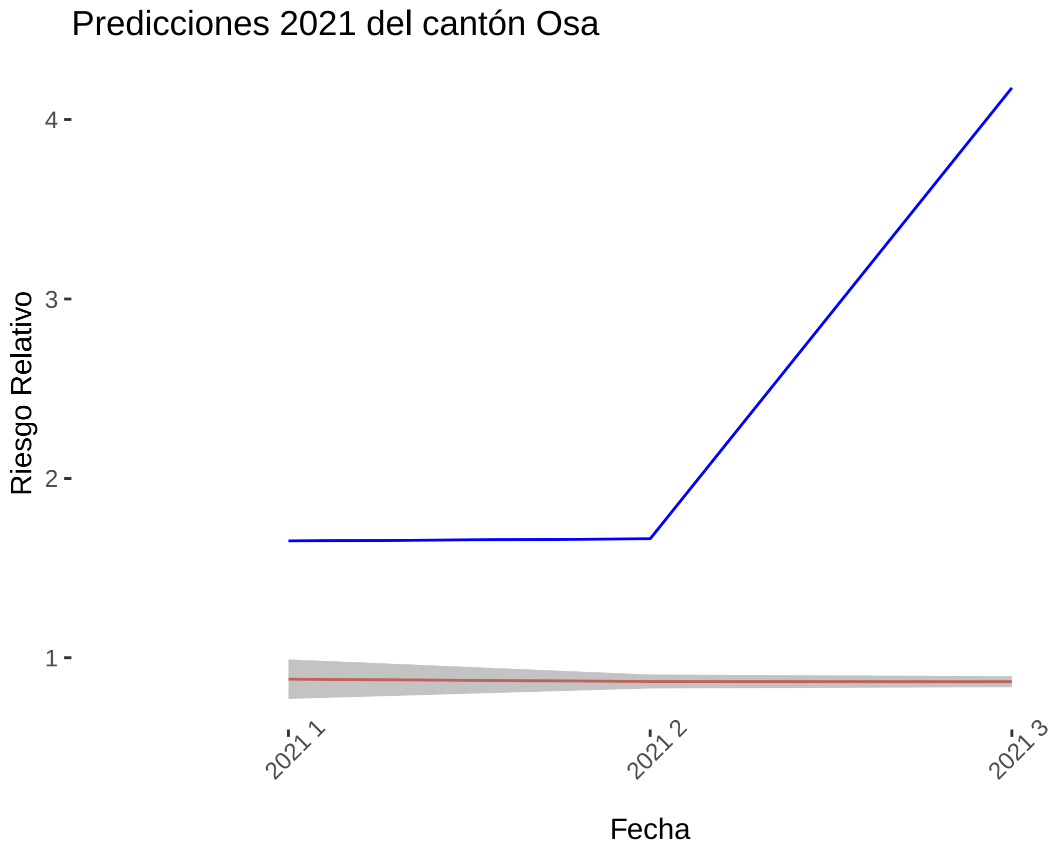

Predicciones 2021 del cantón Nicoya
Riesgo Relativo
1.0
0.5
0.0
2021 1
2021 2
2021 3
Fecha

```
##
## [[18]]
```

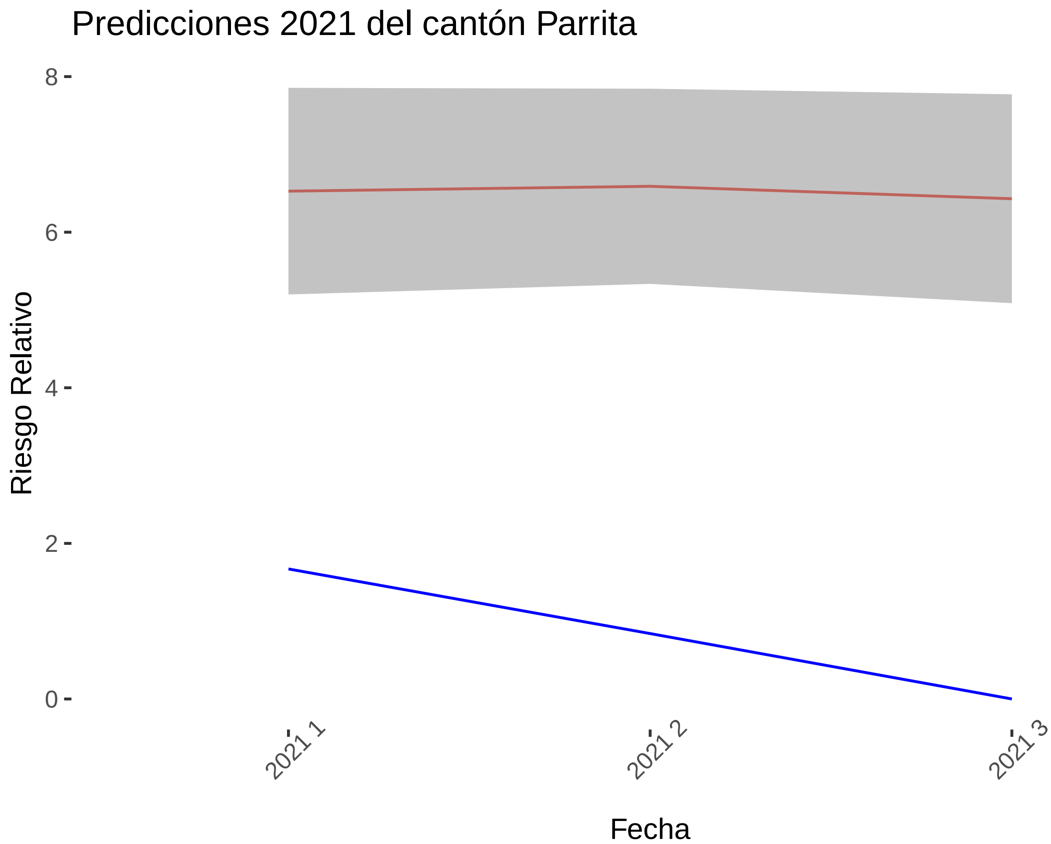


Predicciones 2021 del cantón Orotina
Riesgo Relativo
5
4
3
2021 1
2021 2
2021 3
Fecha

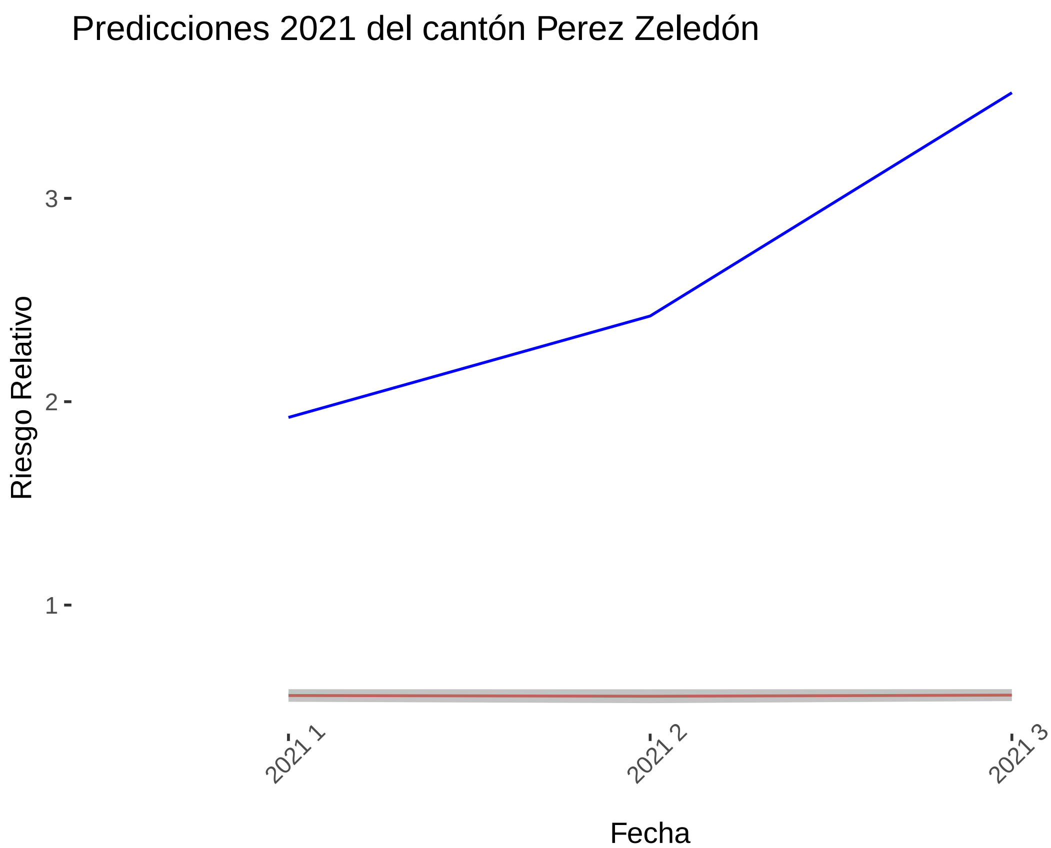

```
##
## [[19]]
```


Predicciones 2021 del cantón Osa
Riesgo Relativo
4
3
2
1
2021 1
2021 2
2021 3
Fecha

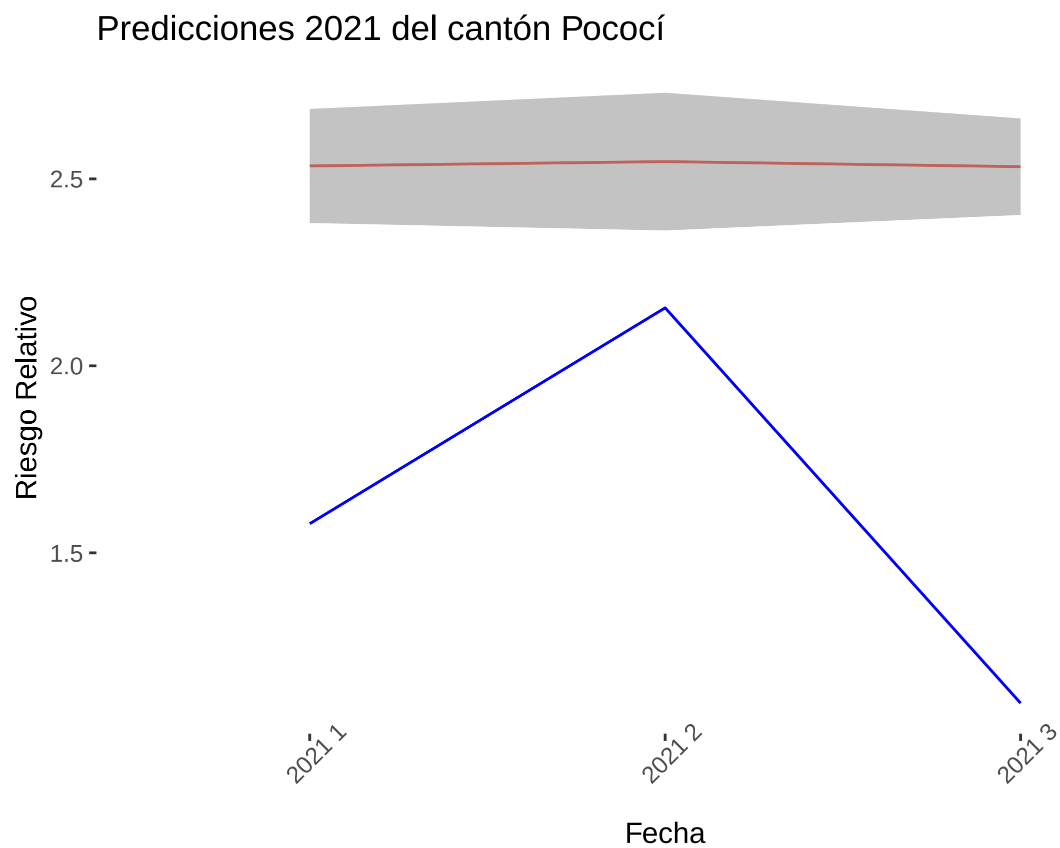
```
##
## [[20]]
```


Predicciones 2021 del cantón Parrita
8
6
4
2
0
Riesgo Relativo
2021 1
2021 2
2021 3
Fecha

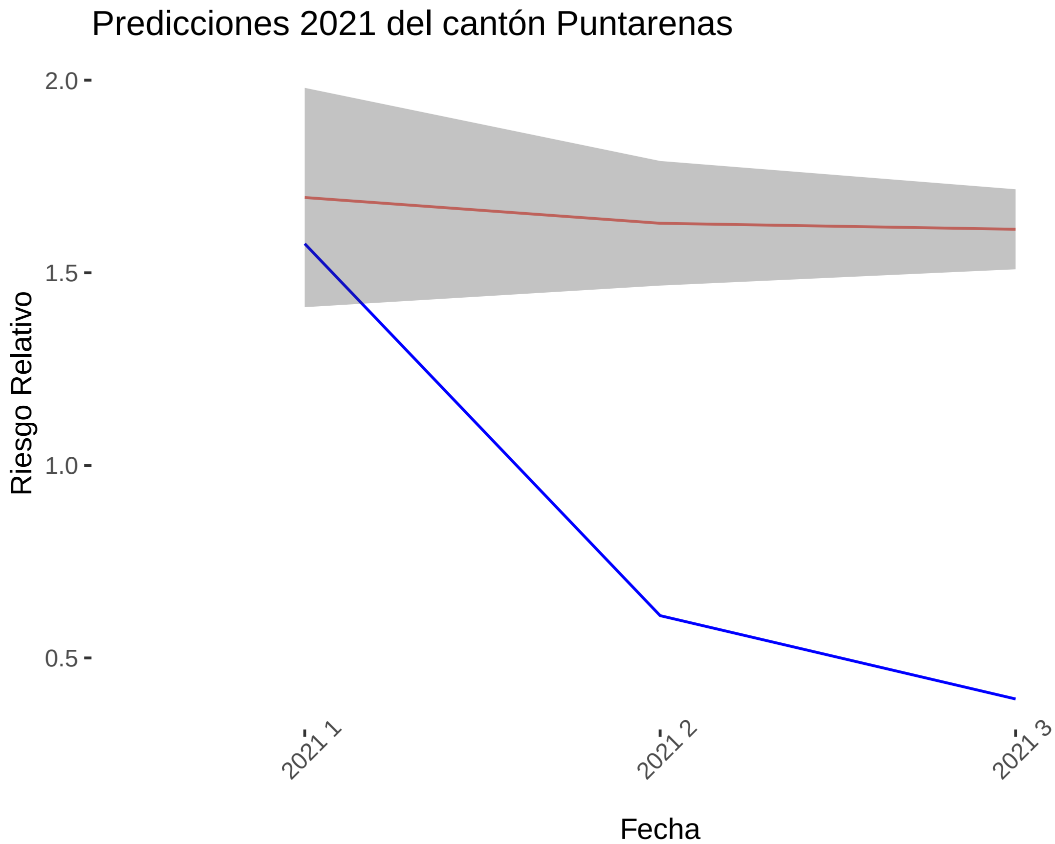
```
##
## [[21]]
```


Predicciones 2021 del cantón Perez Zeledón
Riesgo Relativo
3
2
1
2021 1
2021 2
2021 3
Fecha

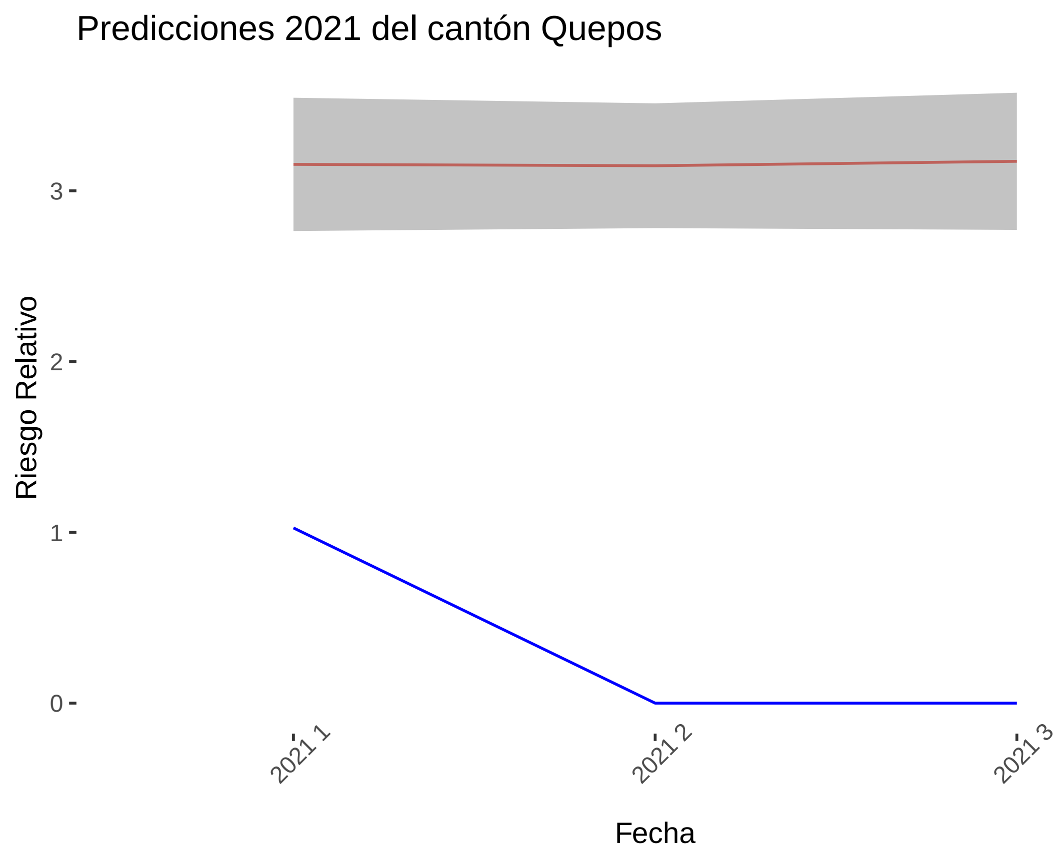
```
##
## [[22]]
```


Predicciones 2021 del cantón Pococí
Riesgo Relativo
2.5
2.0
1.5
2021 1
2021 2
2021 3
Fecha

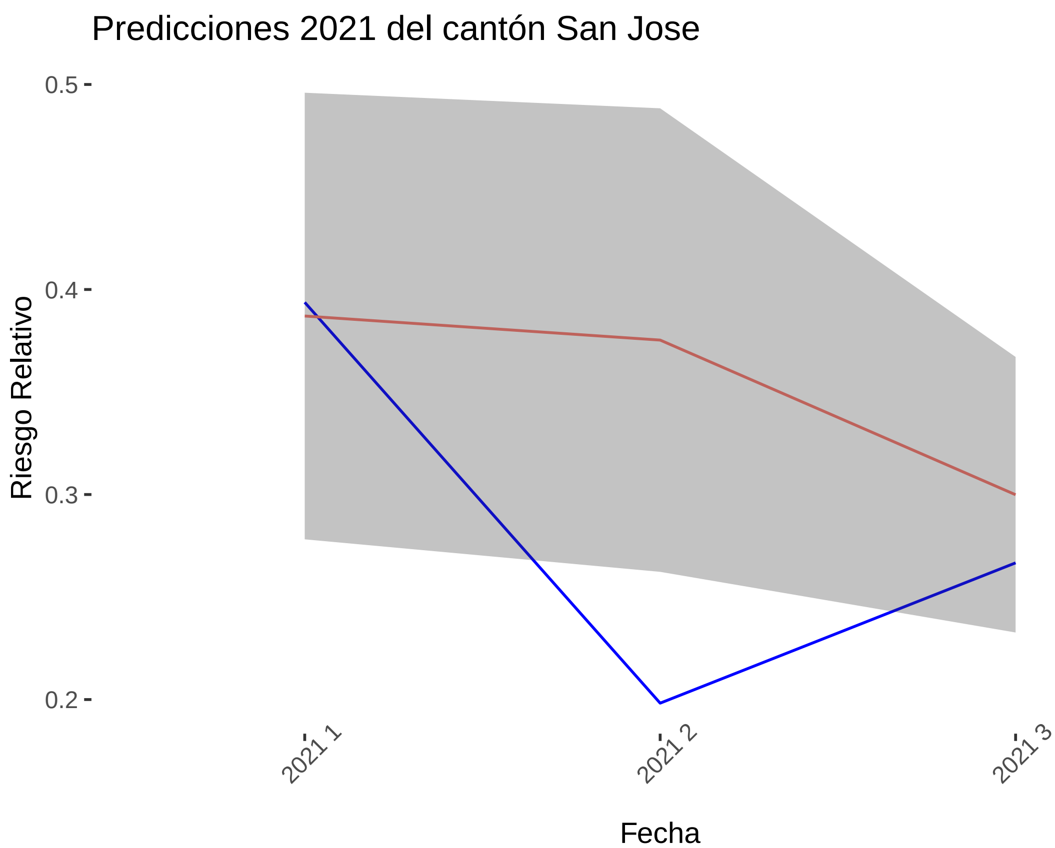
```
##
## [[23]]
```


Predicciones 2021 del cantón Puntarenas
2.0
1.5
1.0
0.5
Riesgo Relativo
2021 1
2021 2
2021 3
Fecha

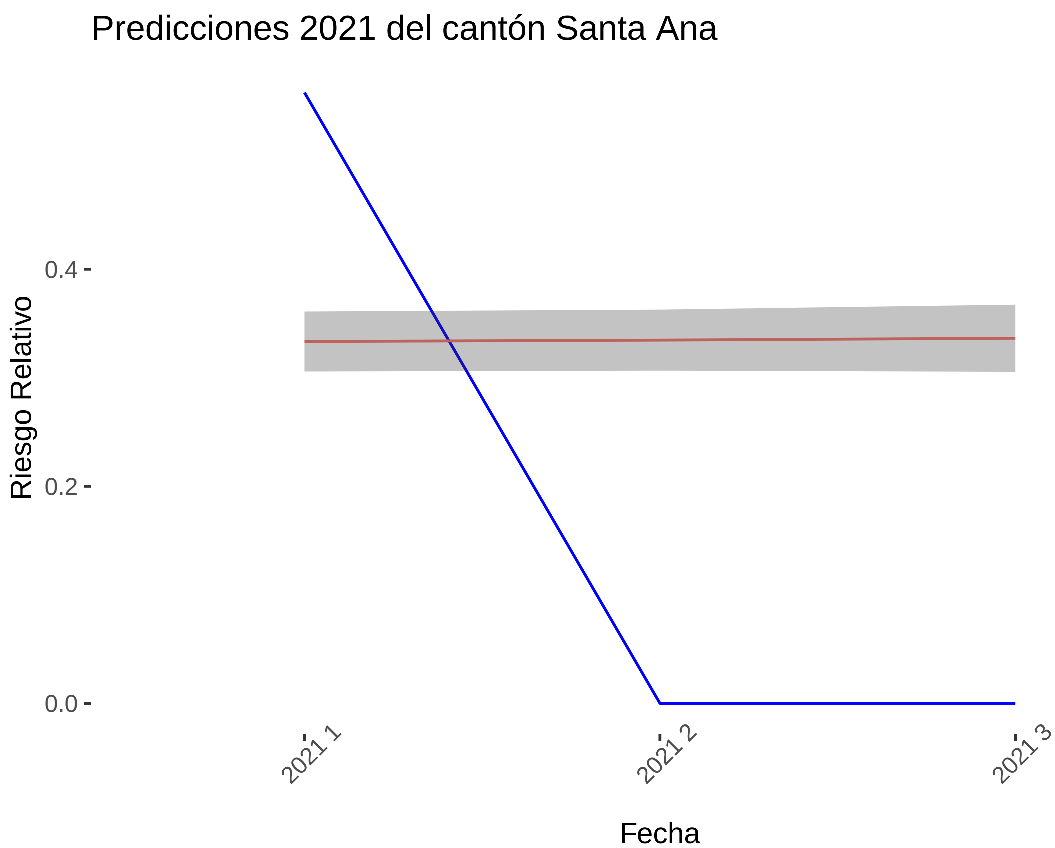
```
##
## [[24]]
```


Predicciones 2021 del cantón Quepos
Riesgo Relativo
3
2
1
0
2021 1
2021 2
2021 3
Fecha

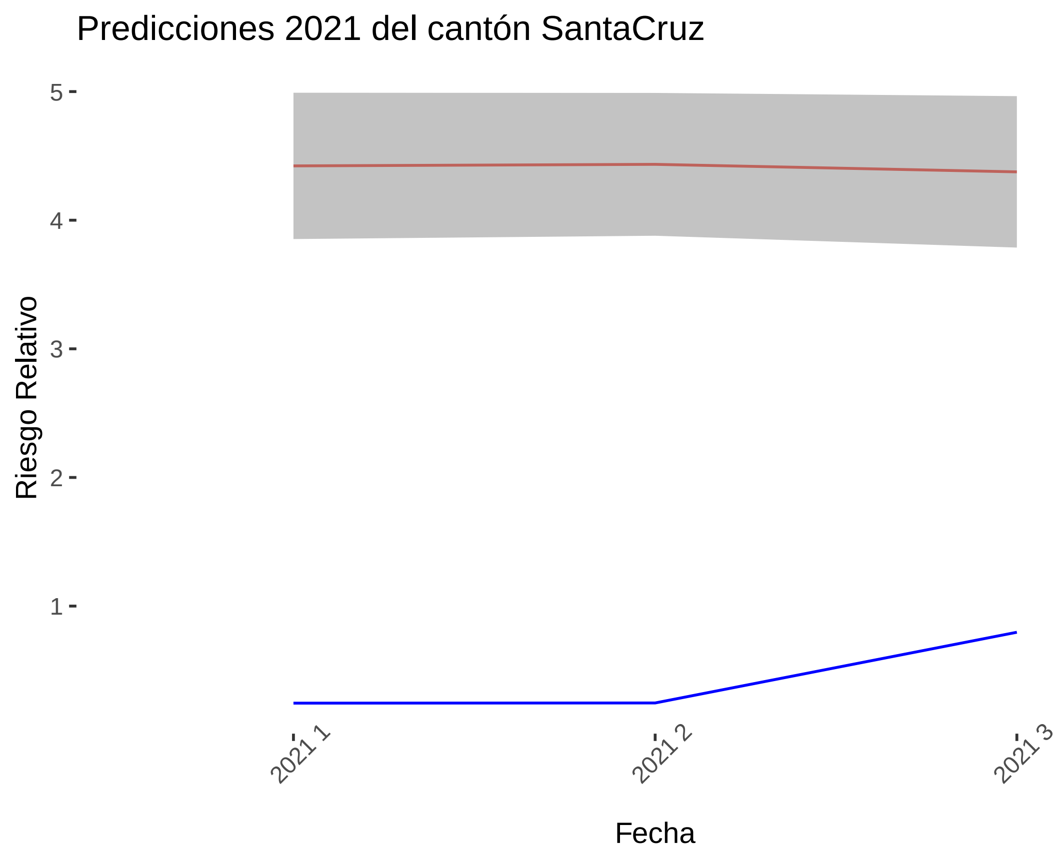
```
##
## [[25]]
```


Predicciones 2021 del cantón San Jose
0.5
0.4
0.3
0.2
Riesgo Relativo
2021 1
2021 2
2021 3
Fecha

```
##
## [[26]]
```

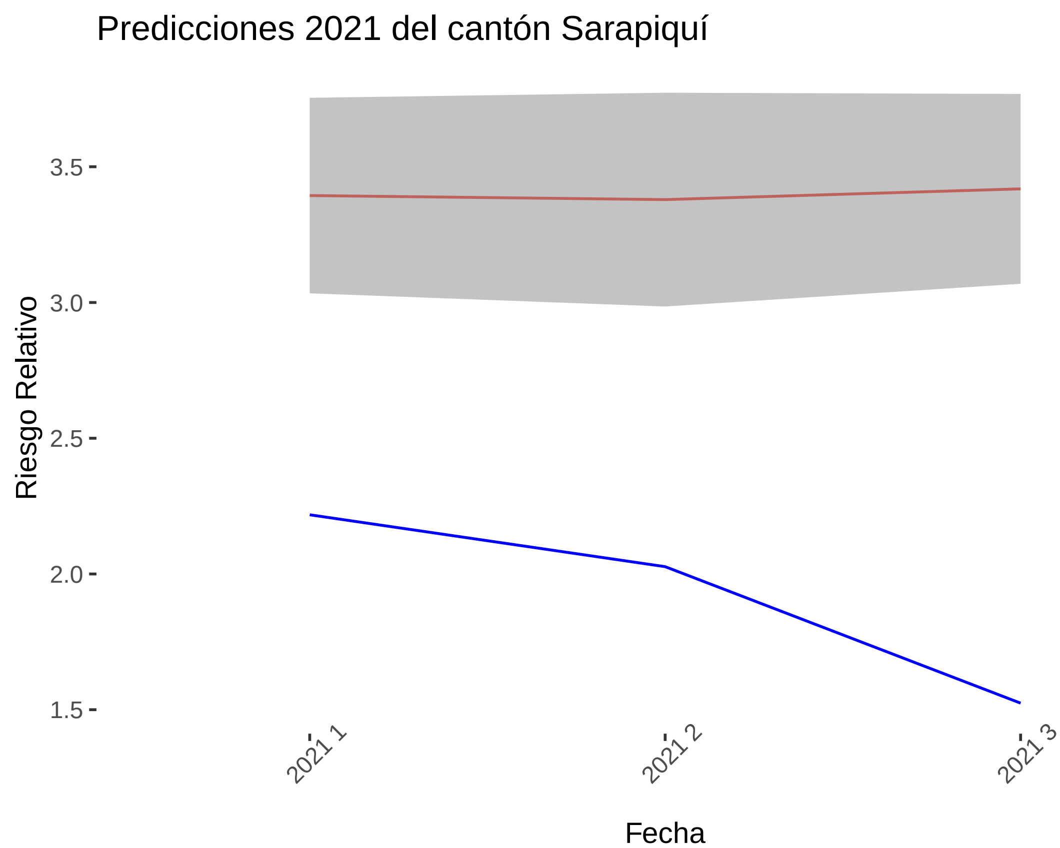

Predicciones 2021 del cantón Santa Ana
0.4
0.2
0.0
Riesgo Relativo
2021 1
2021 2
2021 3
Fecha

```
##
## [[27]]
```

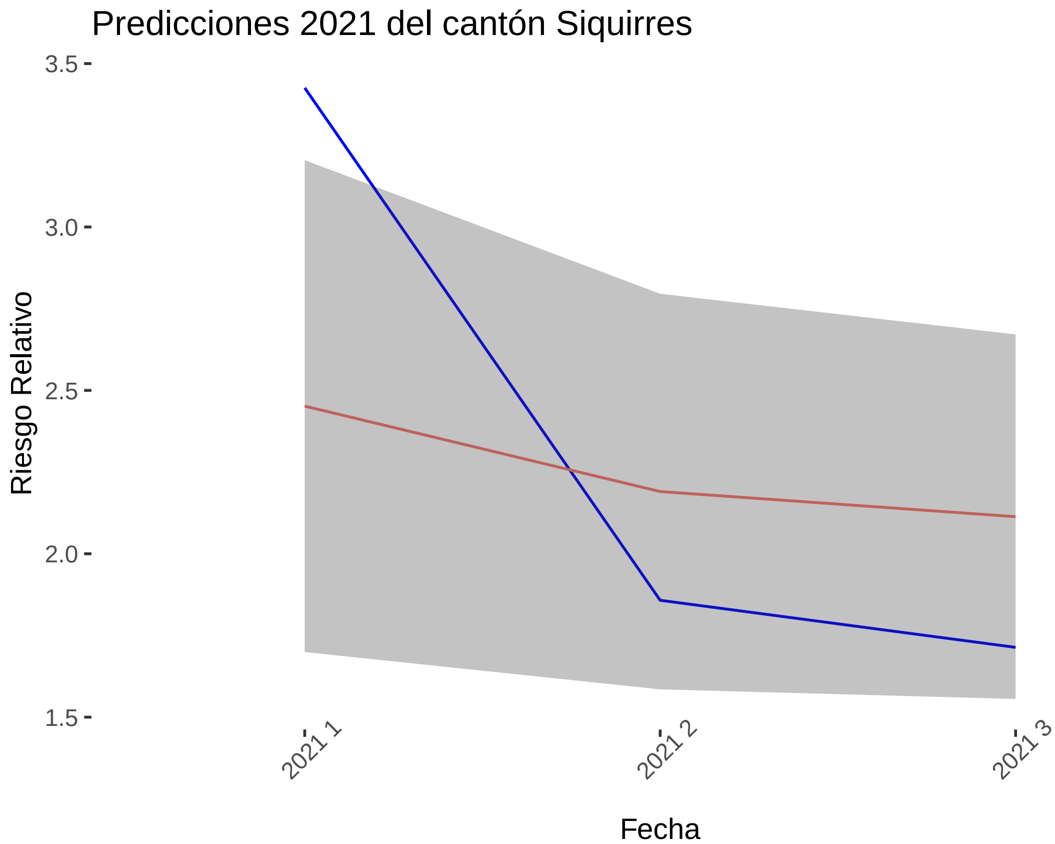


Predicciones 2021 del cantón SantaCruz
5
4
3
2
1
Riesgo Relativo
2021 1
2021 2
2021 3
Fecha

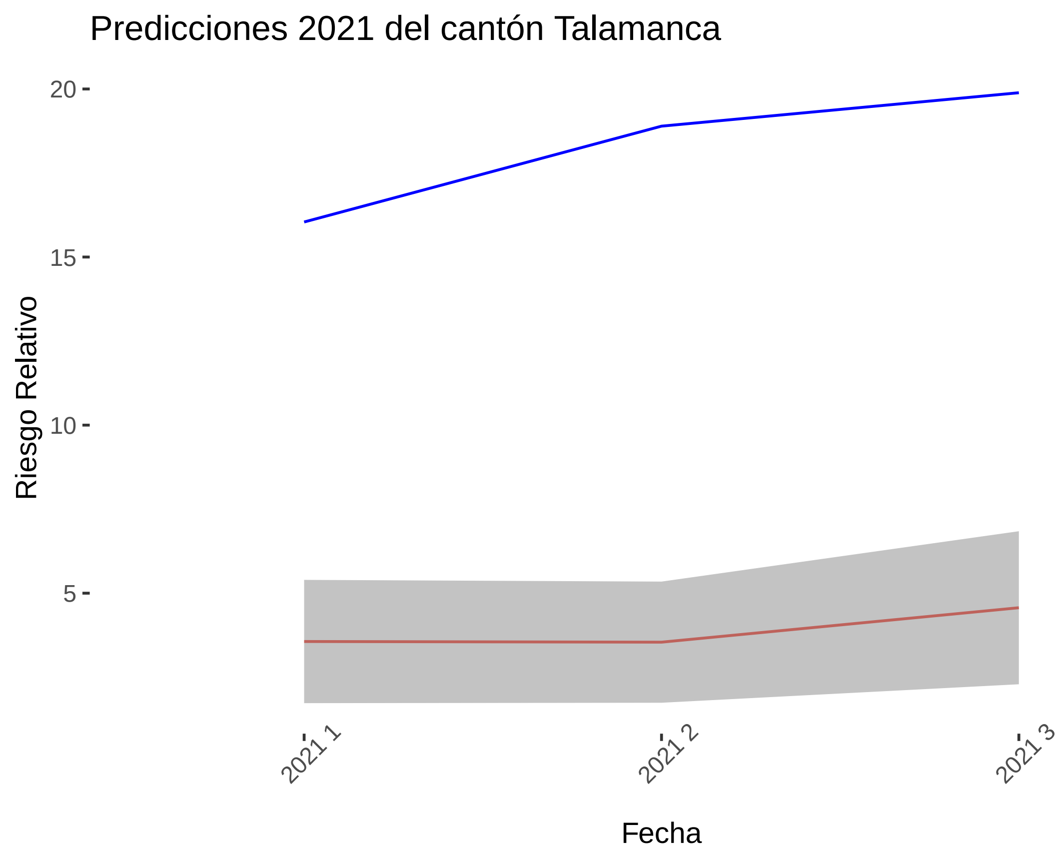

```
##
## [[28]]
```


Predicciones 2021 del cantón Sarapiquí
Riesgo Relativo
3.5
3.0
2.5
2.0
1.5
2021 1
2021 2
2021 3
Fecha

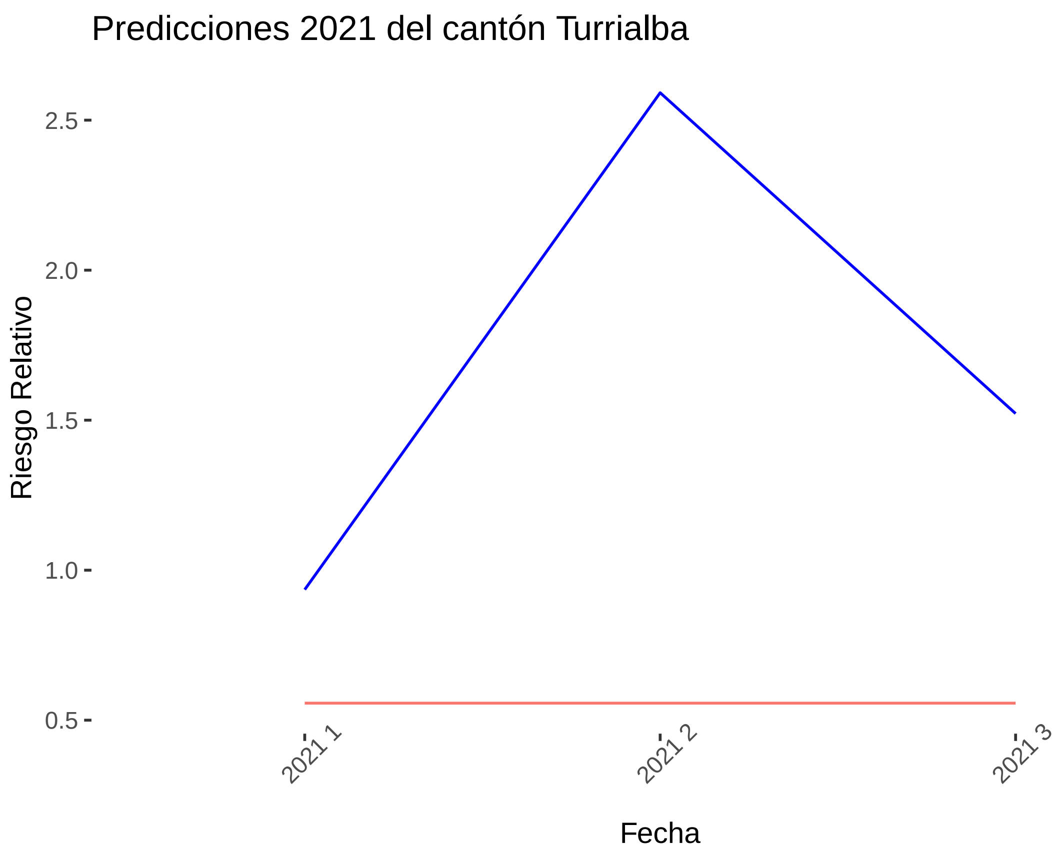
```
##
## [[29]]
```


Predicciones 2021 del cantón Siquirres
3.5
3.0
2.5
2.0
1.5
Riesgo Relativo
2021 1
2021 2
2021 3
Fecha

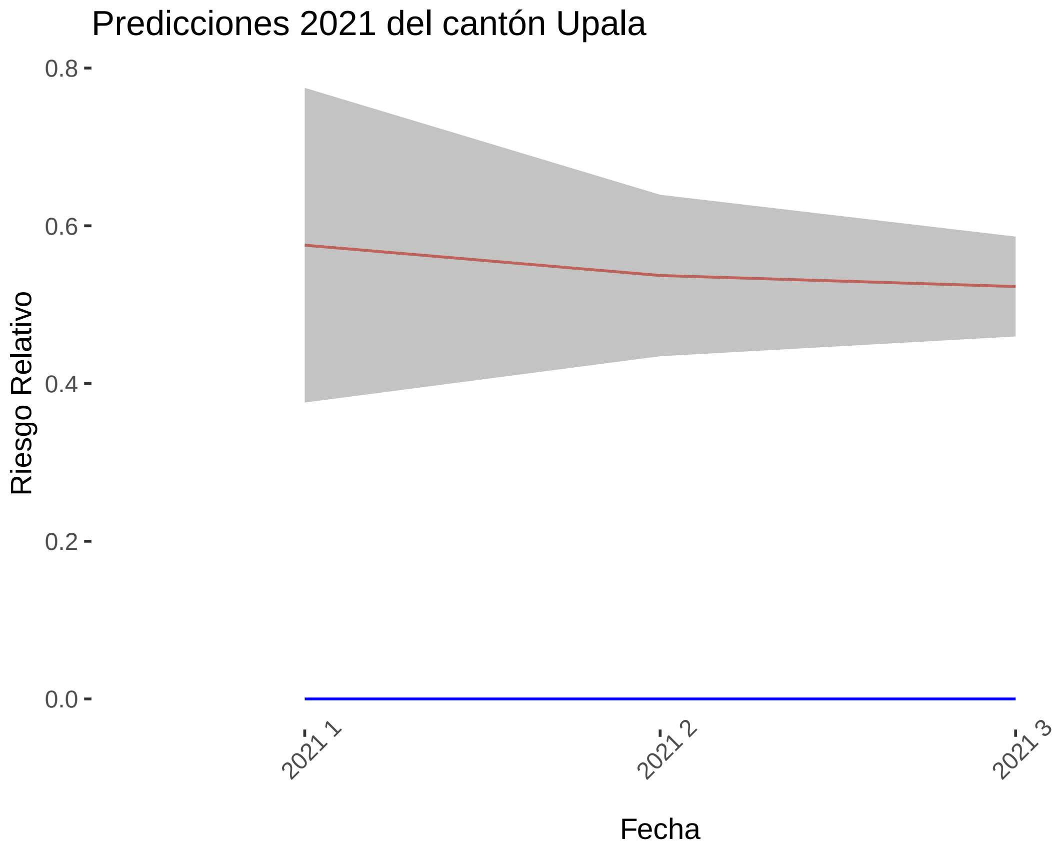
```
##
## [[30]]
```


Predicciones 2021 del cantón Talamanca
20
15
10
5
Riesgo Relativo
2021 1
2021 2
2021 3
Fecha

```
##
## [[31]]
```


Predicciones 2021 del cantón Turrialba
Riesgo Relativo
2.5
2.0
1.5
1.0
0.5
2021 1
2021 2
2021 3
Fecha

```
##
## [[32]]
```

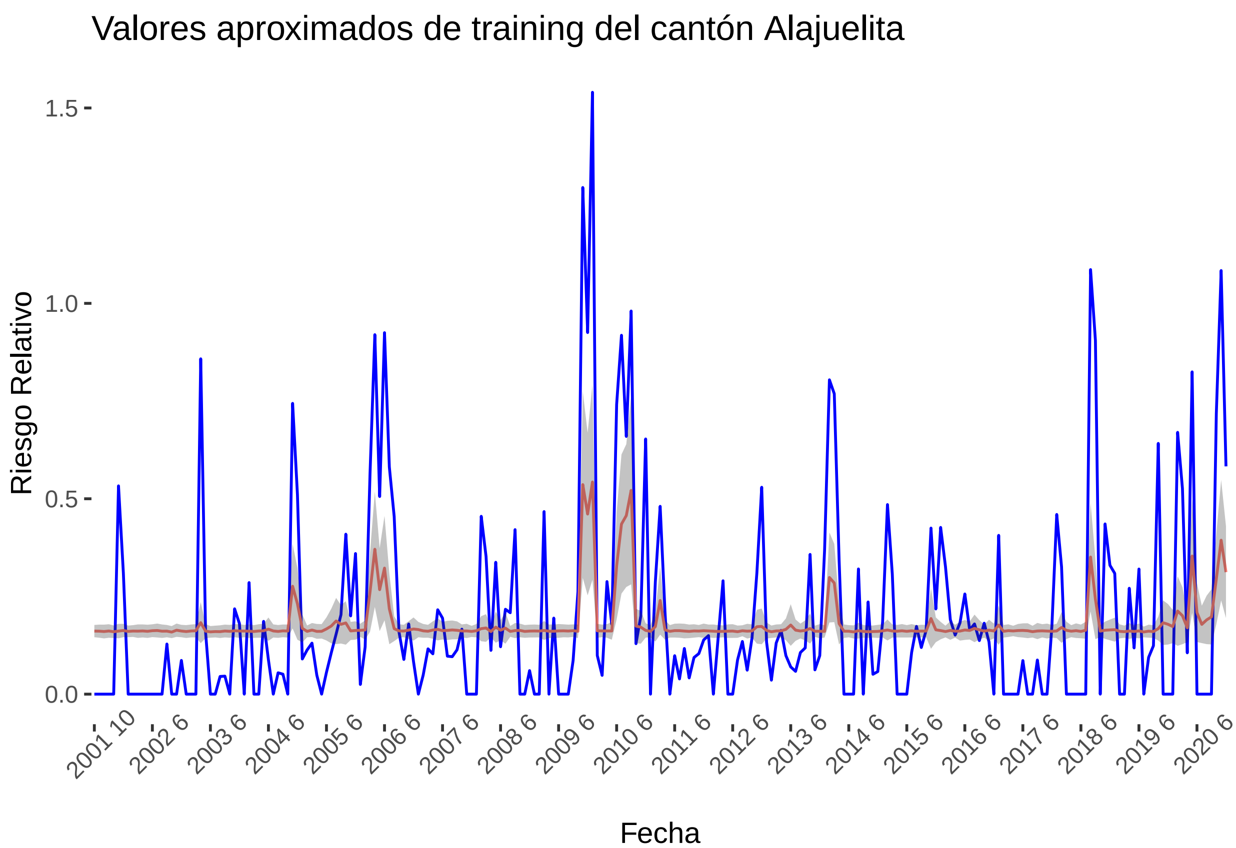

Predicciones 2021 del cantón Upala
Riesgo Relativo
0.8
0.6
0.4
0.2
0.0
2021 1
2021 2
2021 3
Fecha

```
p2
```

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

Valores aproximados de training del cantón Alajuela
Riesgo Relativo
Fecha
0
1
2
3
4
5
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6


```
##
## [[2]]
```



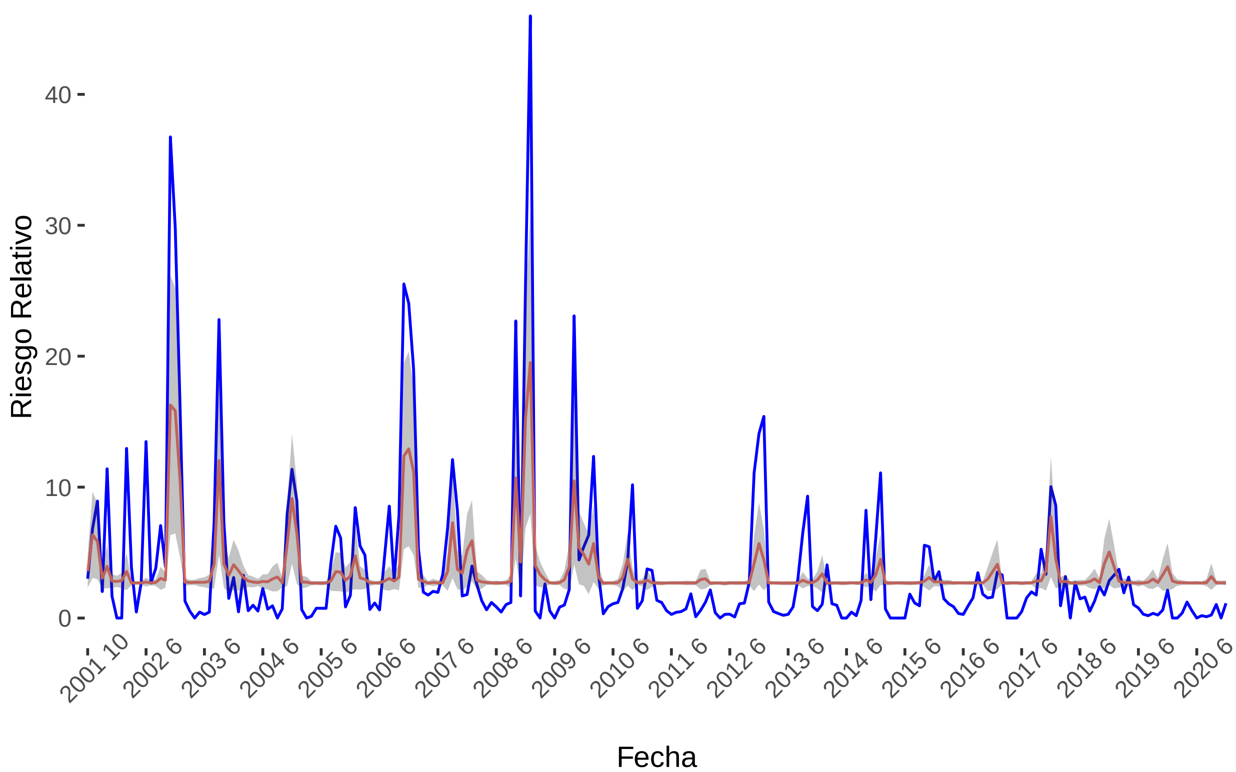
Valores aproximados de training del cantón Alajuelita
Riesgo Relativo
1.5
1.0
0.5
0.0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha


```
##
## [[3]]
```


Valores aproximados de training del cantón Atenas
Riesgo Relativo
20
15
10
5
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

```
##
## [[4]]
```

Valores aproximados de training del cantón Cañas




```
##
## [[5]]
```

Valores aproximados de training del cantón Carrillo
Riesgo Relativo
40
30
20
10
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha


```
##
## [[6]]
```

Valores aproximados de training del cantón Corredores
Riesgo Relativo
0
5
10
15
20
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha


```
##
## [[7]]
```


Valores aproximados de training del cantón Desamparados
Riesgo Relativo
1.5
1.0
0.5
0.0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

```
##
## [[8]]
```


Valores aproximados de training del cantón Esparza
Riesgo Relativo
30
20
10
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

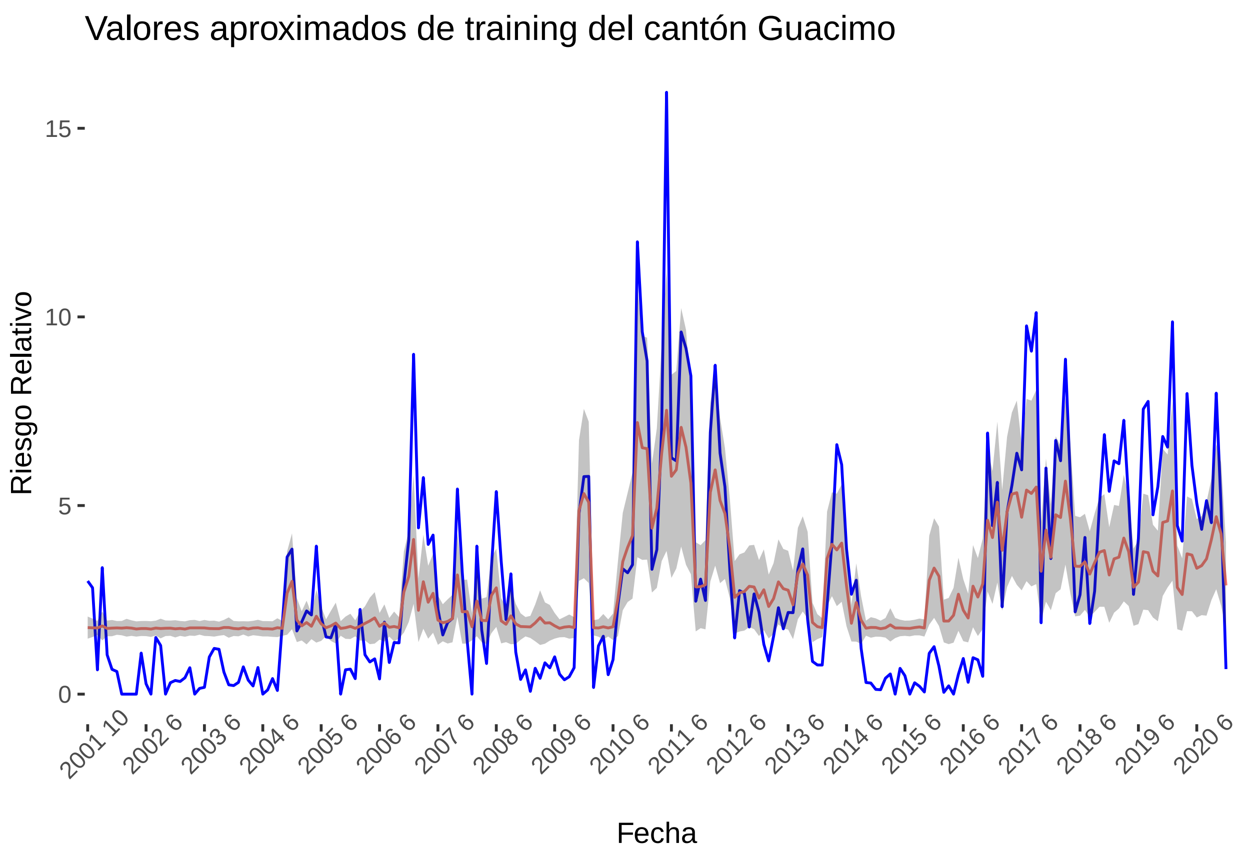
```
##
## [[9]]
```


Valores aproximados de training del cantón Garabito
Riesgo Relativo
40
30
20
10
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

```
##
## [[10]]
```


Valores aproximados de training del cantón Golfito
Riesgo Relativo
20
15
10
5
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

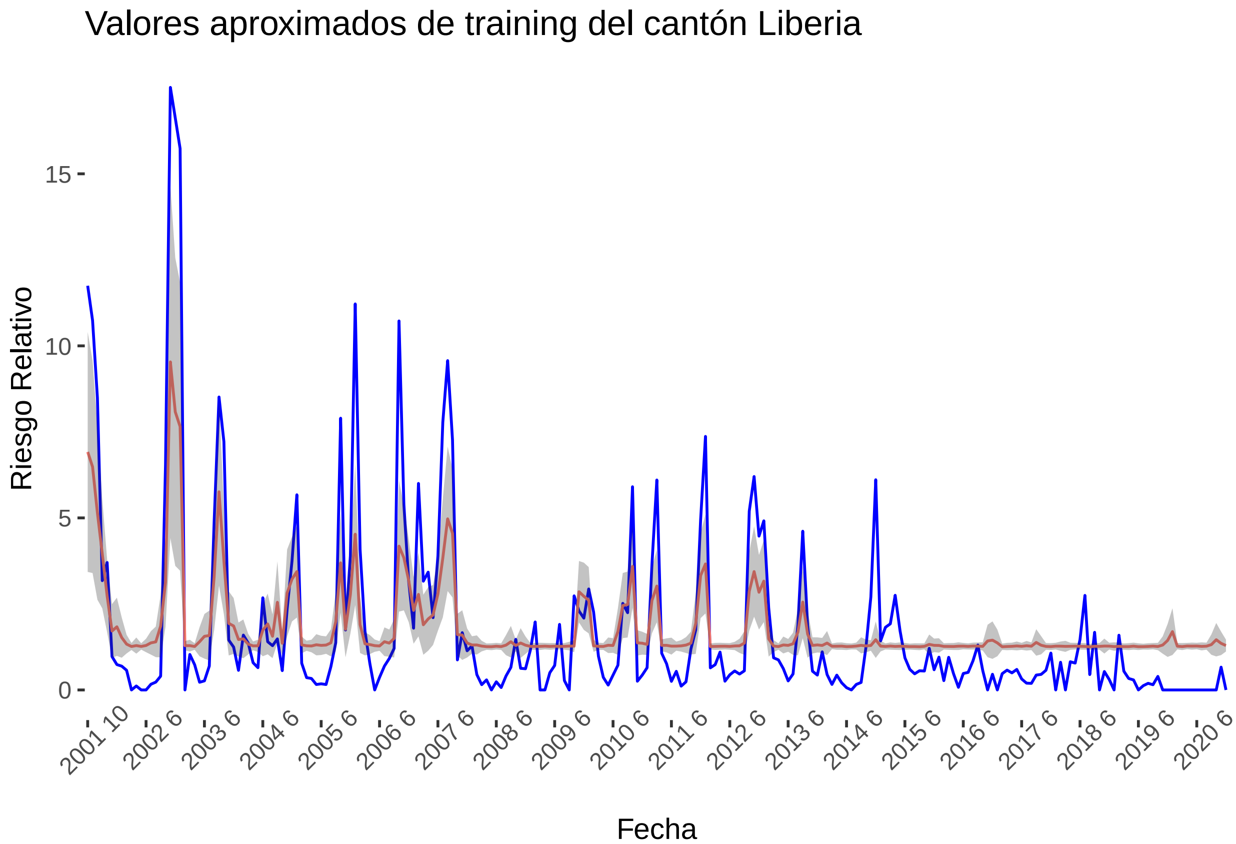
```
##
## [[11]]
```


Valores aproximados de training del cantón Guacimo
Riesgo Relativo
15
10
5
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

```
##
## [[12]]
```


Valores aproximados de training del cantón La Cruz
Riesgo Relativo
30
20
10
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

```
##
## [[13]]
```

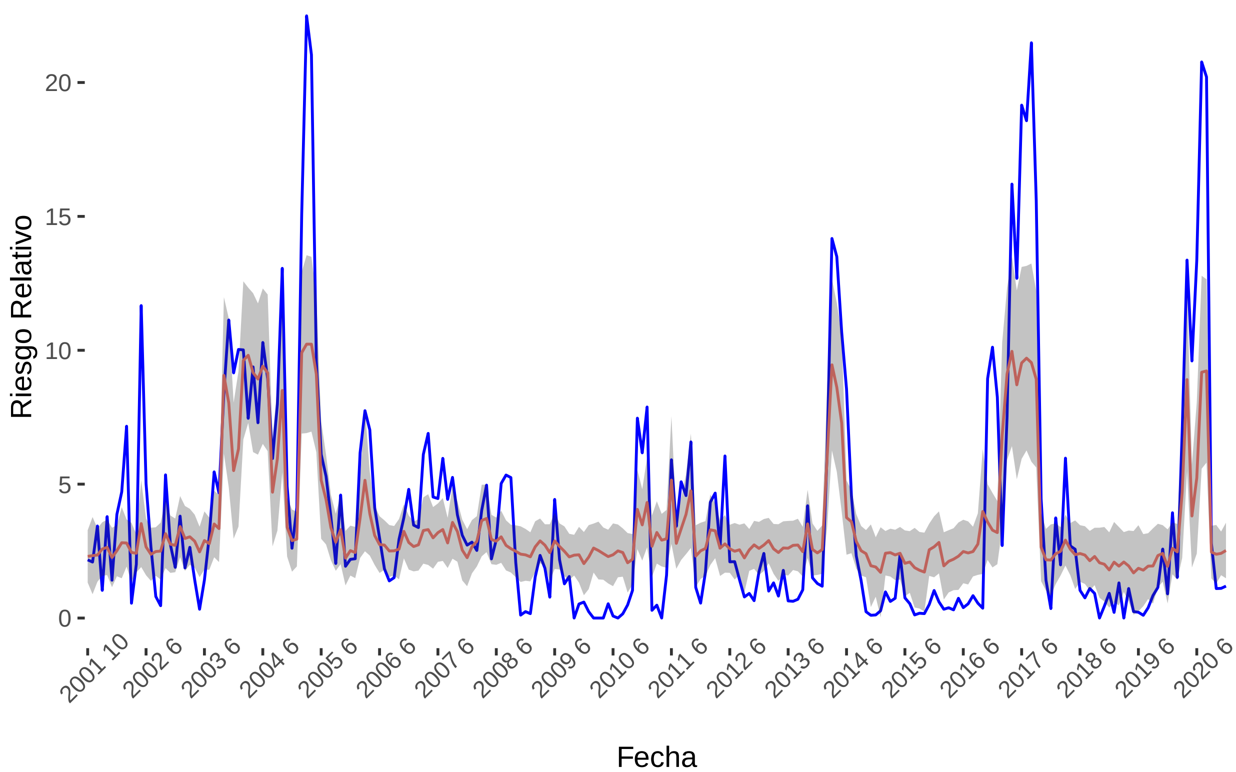

Valores aproximados de training del cantón Liberia
Riesgo Relativo
15
10
5
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

```
##
## [[14]]
```


Valores aproximados de training del cantón Limon
Riesgo Relativo
15
10
5
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

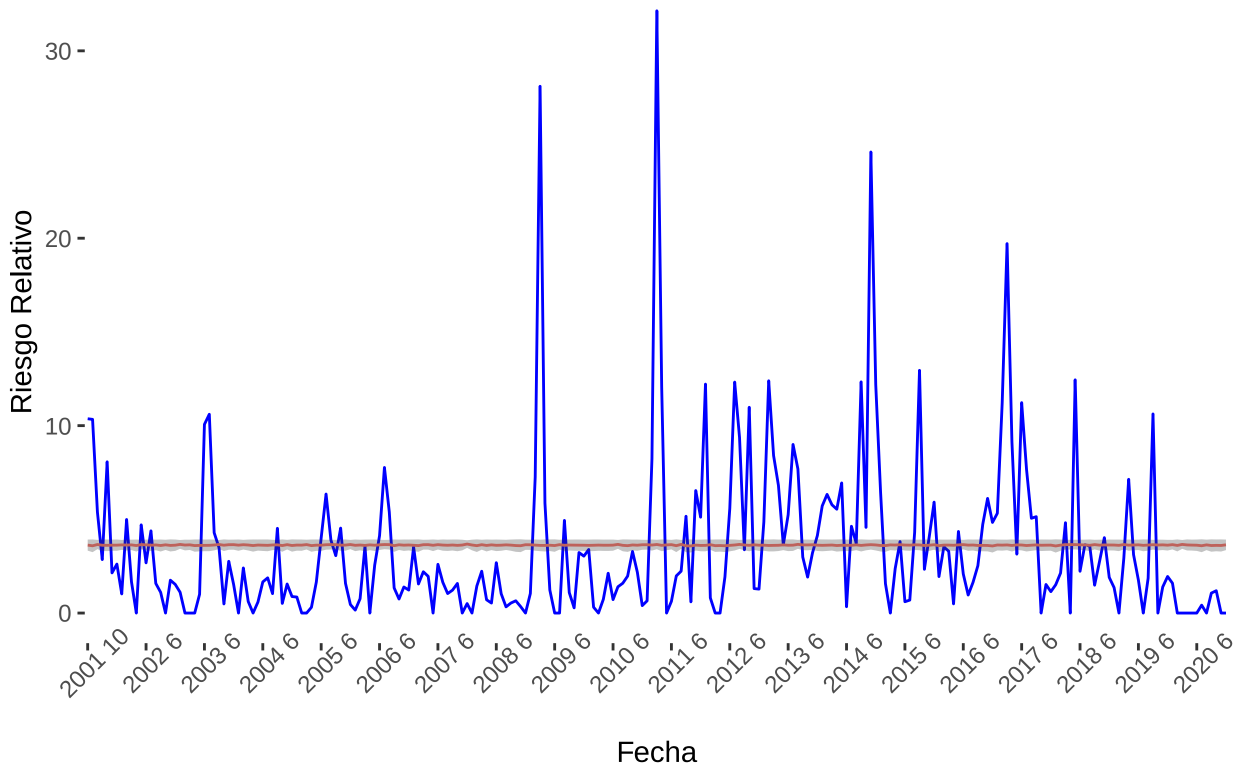
```
##
## [[15]]
```

Valores aproximados de training del cantón Matina



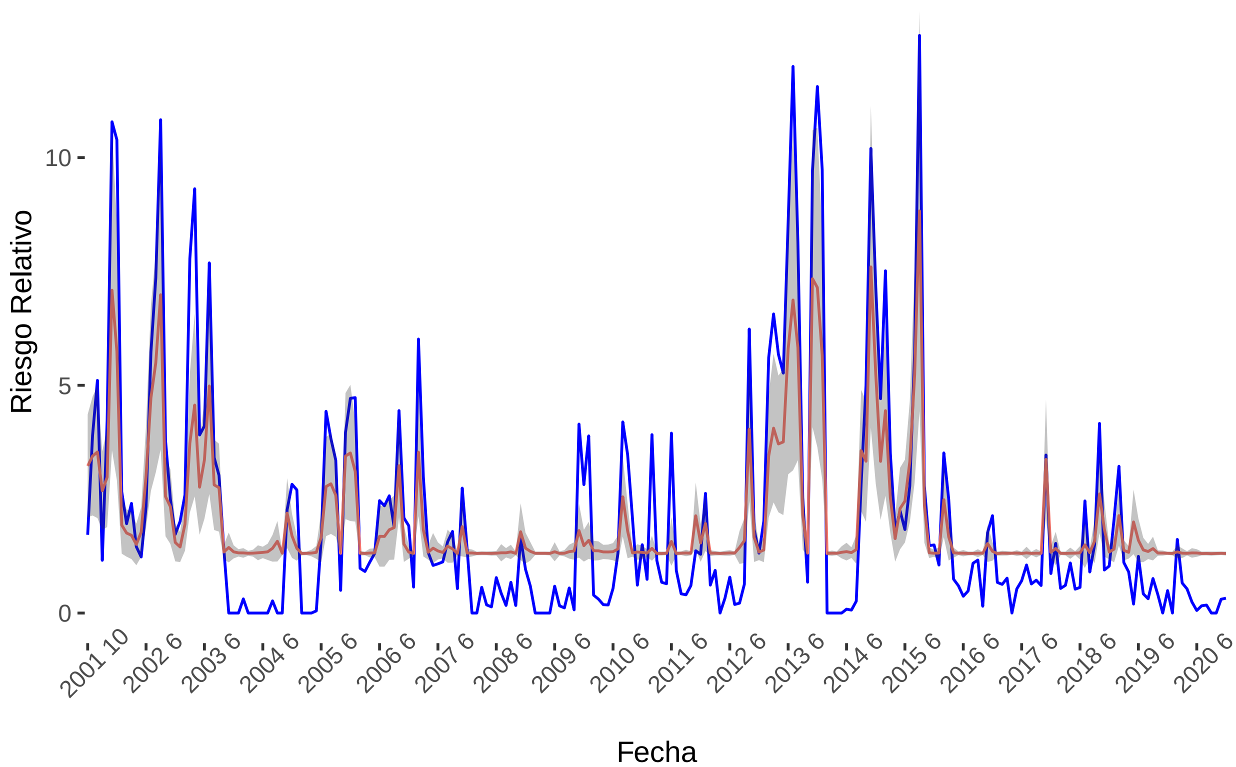

```
##
## [[16]]
```

Valores aproximados de training del cantón Montes de Oro




```
##
## [[17]]
```

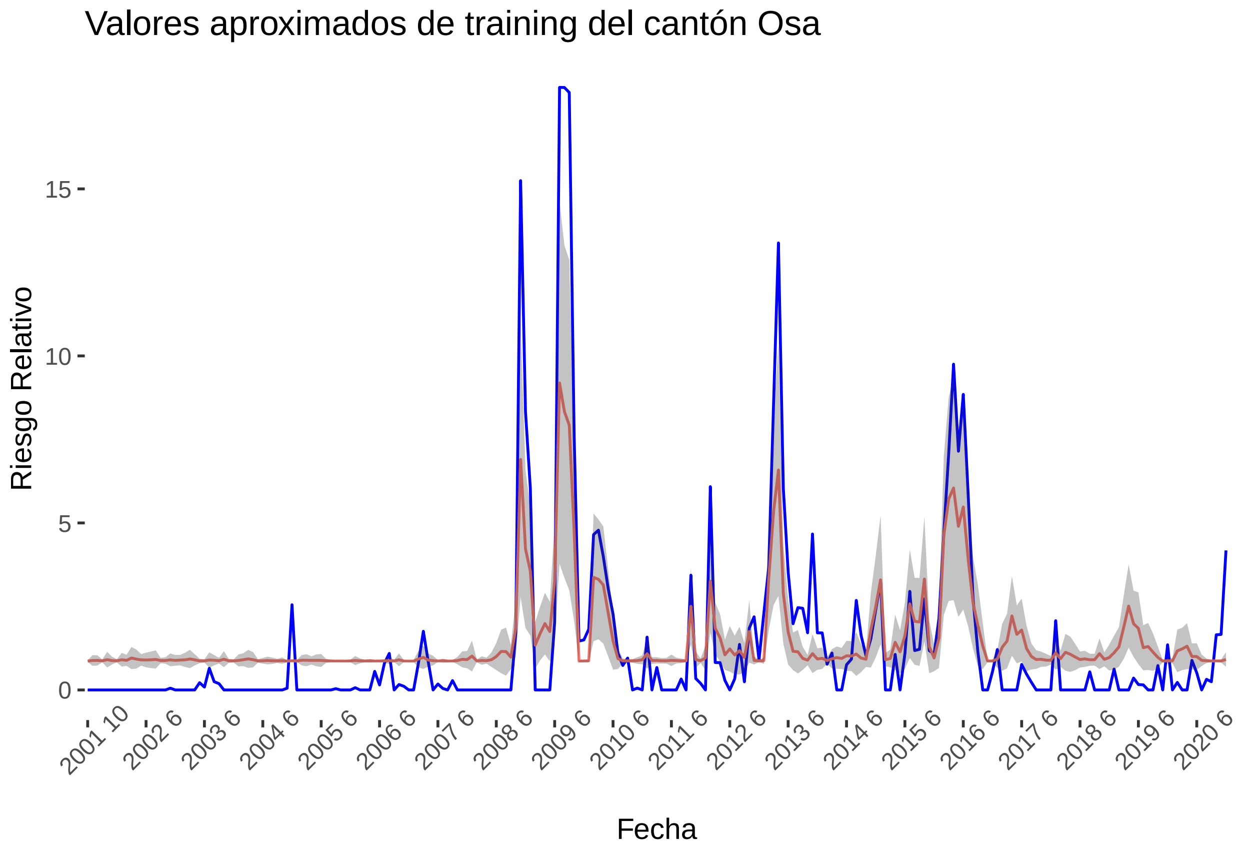
Valores aproximados de training del cantón Nicoya




```
##
## [[18]]
```


Valores aproximados de training del cantón Orotina
Riesgo Relativo
40
20
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

```
##
## [[19]]
```



Valores aproximados de training del cantón Osa
Riesgo Relativo
15
10
5
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha


```
##
## [[20]]
```

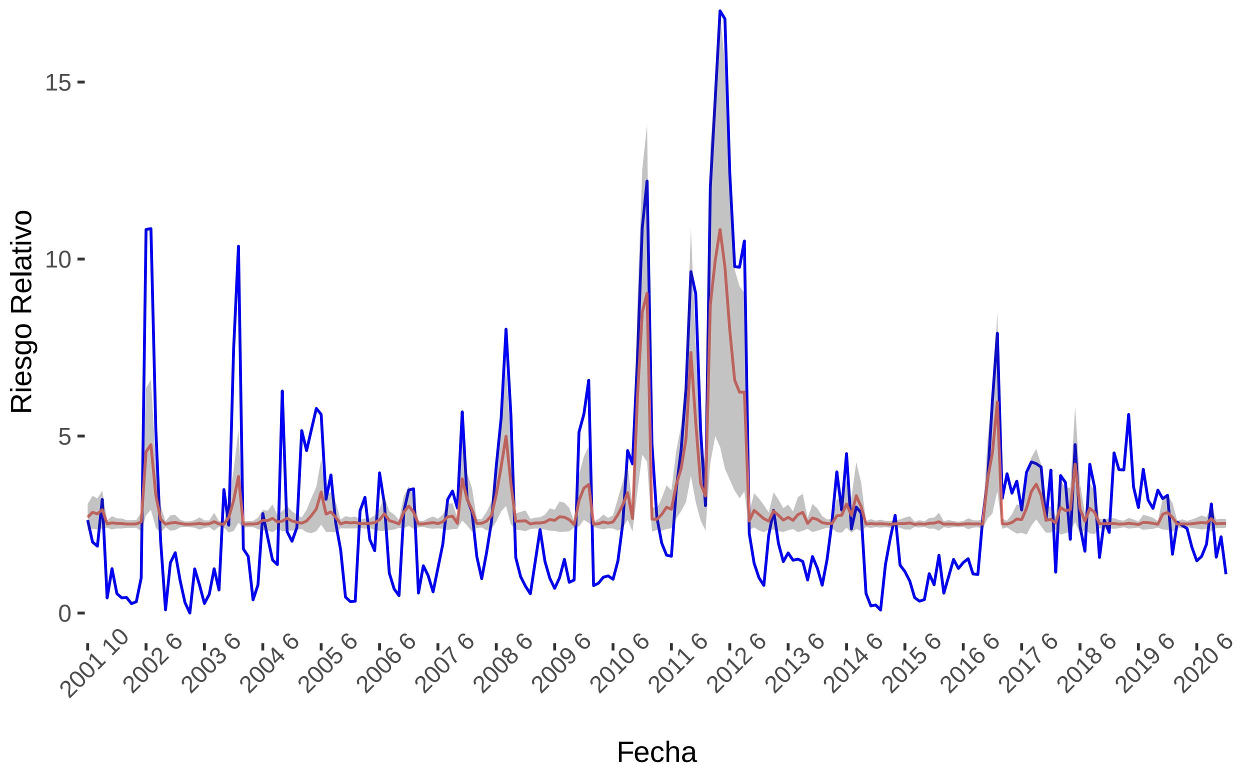

Valores aproximados de training del cantón Parrita
Riesgo Relativo
150
100
50
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

```
##
## [[21]]
```

Valores aproximados de training del cantón Perez Zeledón
Riesgo Relativo
0
2
4
6
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

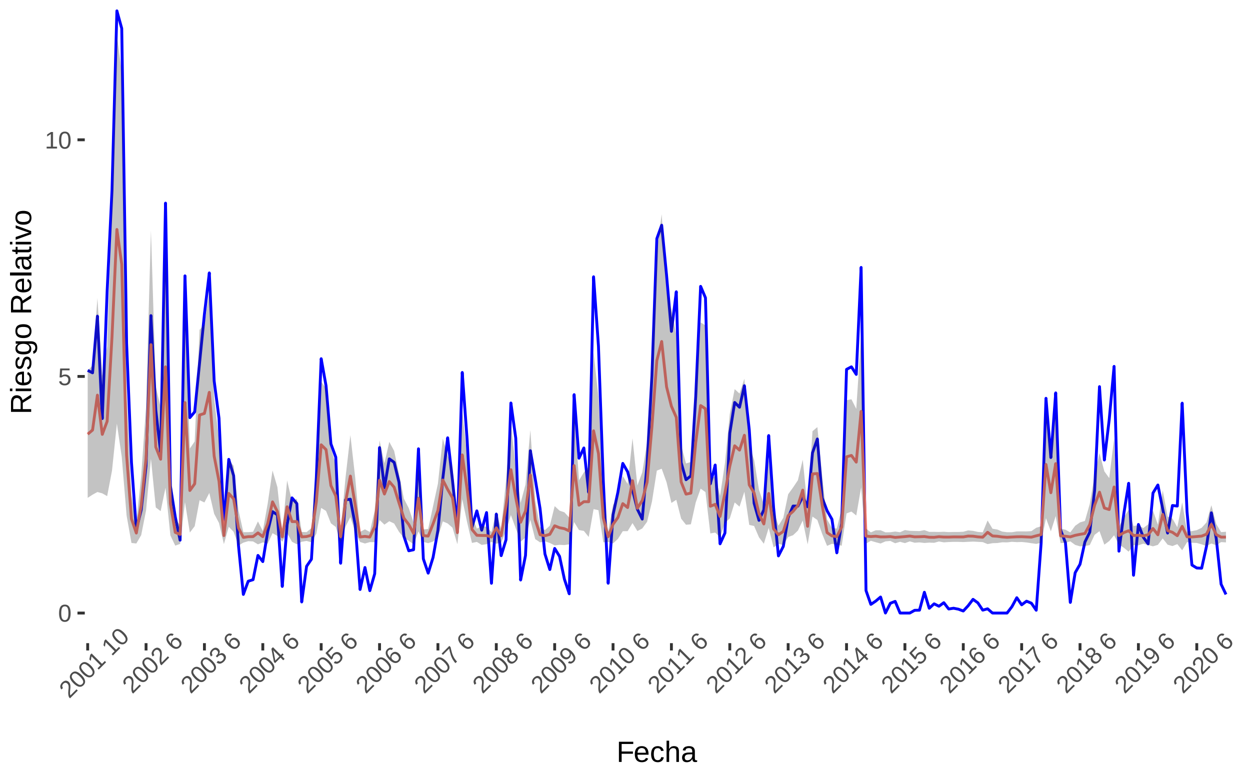

```
##
## [[22]]
```

Valores aproximados de training del cantón Pococí




```
##
## [[23]]
```

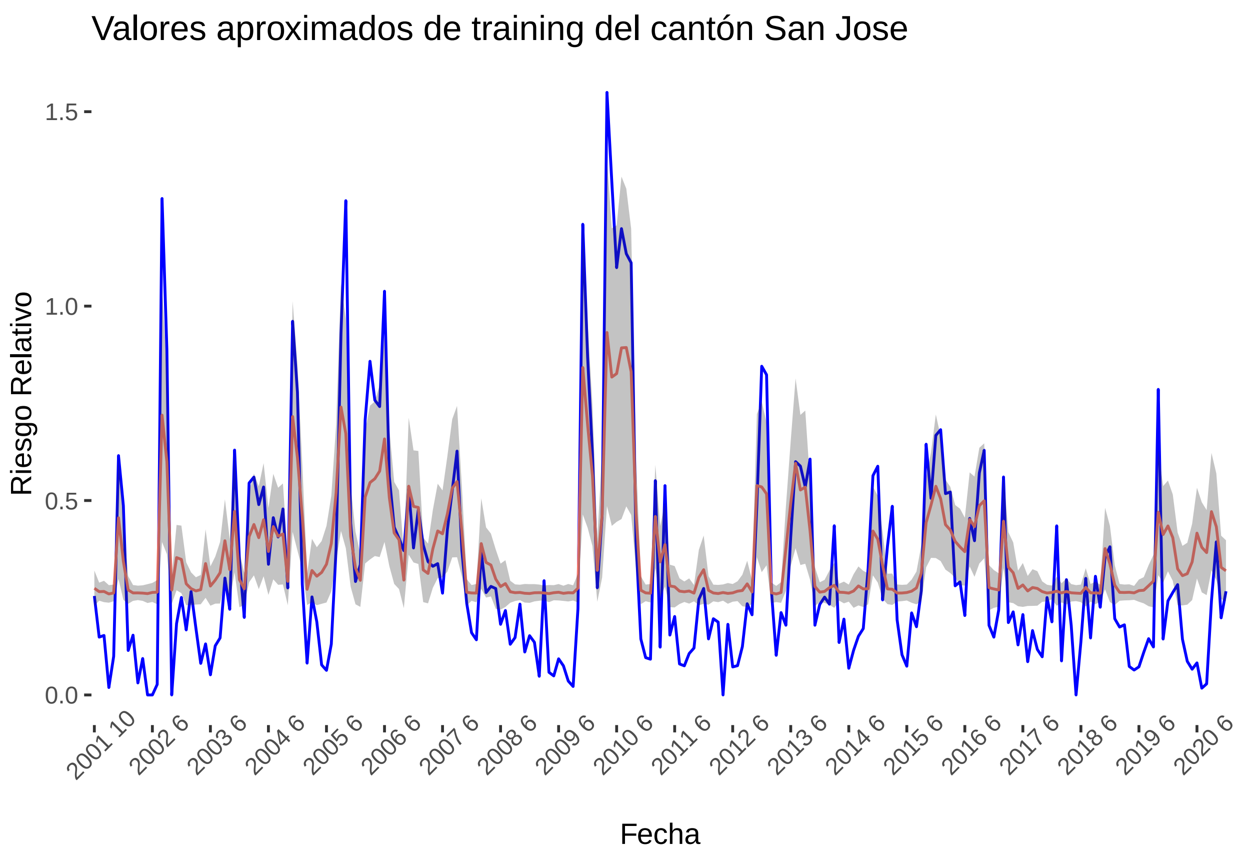
Valores aproximados de training del cantón Puntarenas



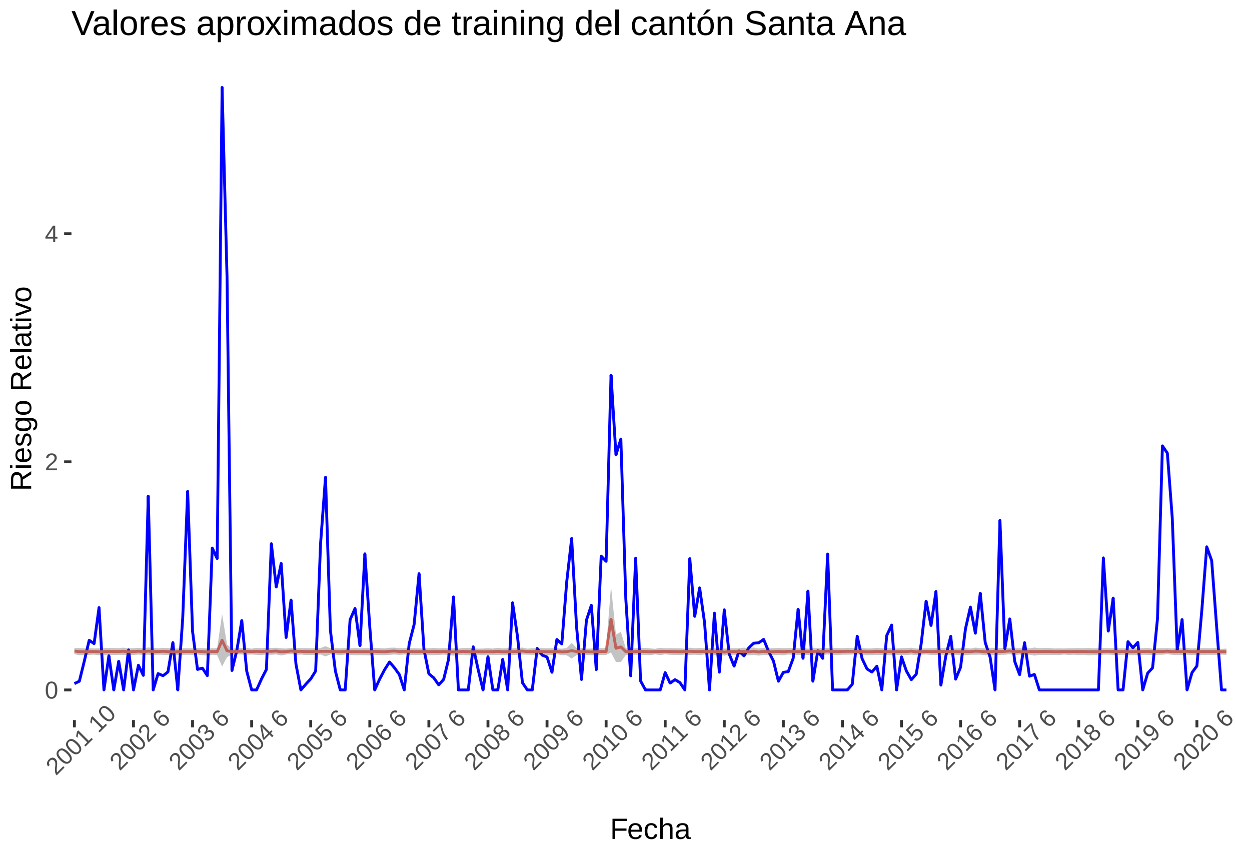

```
##
## [[24]]
```

Valores aproximados de training del cantón Quepos
Riesgo Relativo
60
40
20
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

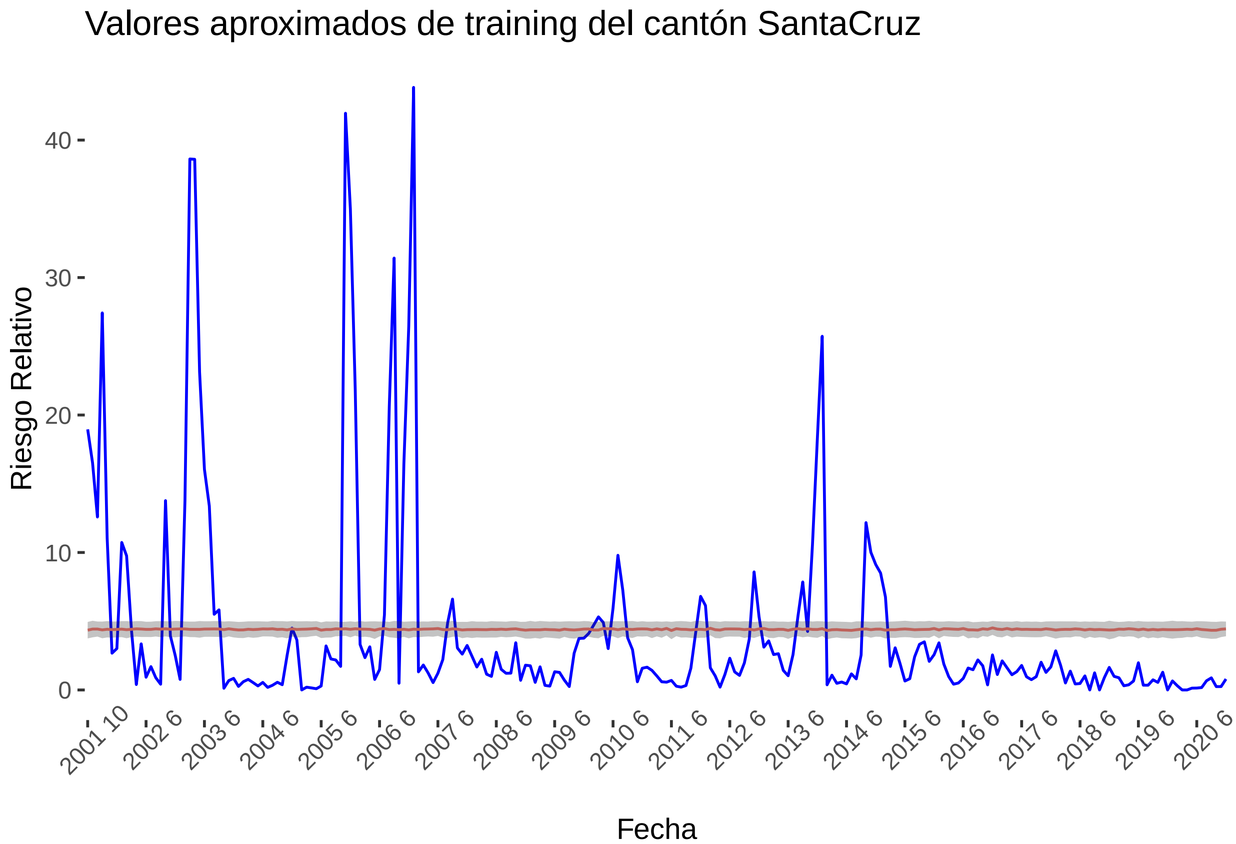

```
##
## [[25]]
```


Valores aproximados de training del cantón San Jose
Riesgo Relativo
1.5
1.0
0.5
0.0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

```
##
## [[26]]
```


Valores aproximados de training del cantón Santa Ana
Riesgo Relativo
0
2
4
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

```
##
## [[27]]
```

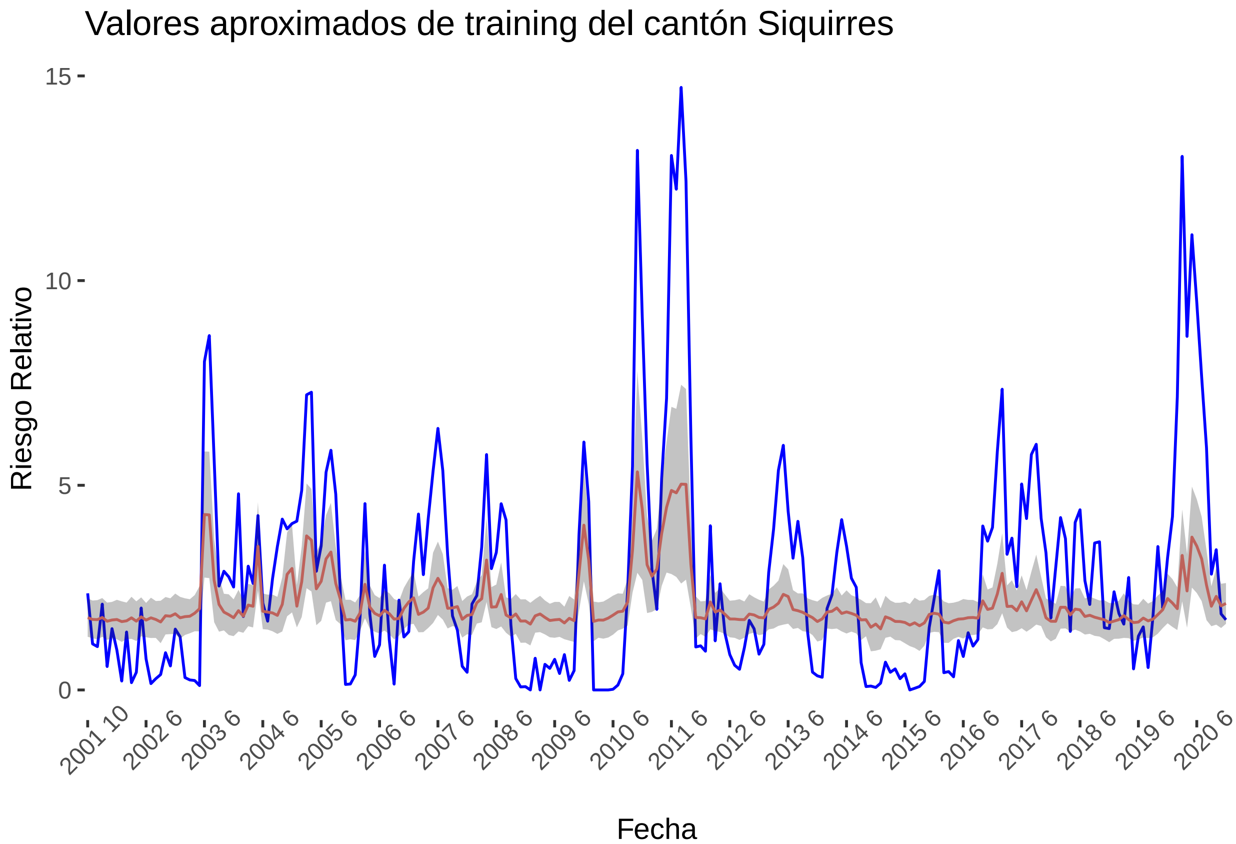


Valores aproximados de training del cantón SantaCruz
Riesgo Relativo
40
30
20
10
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

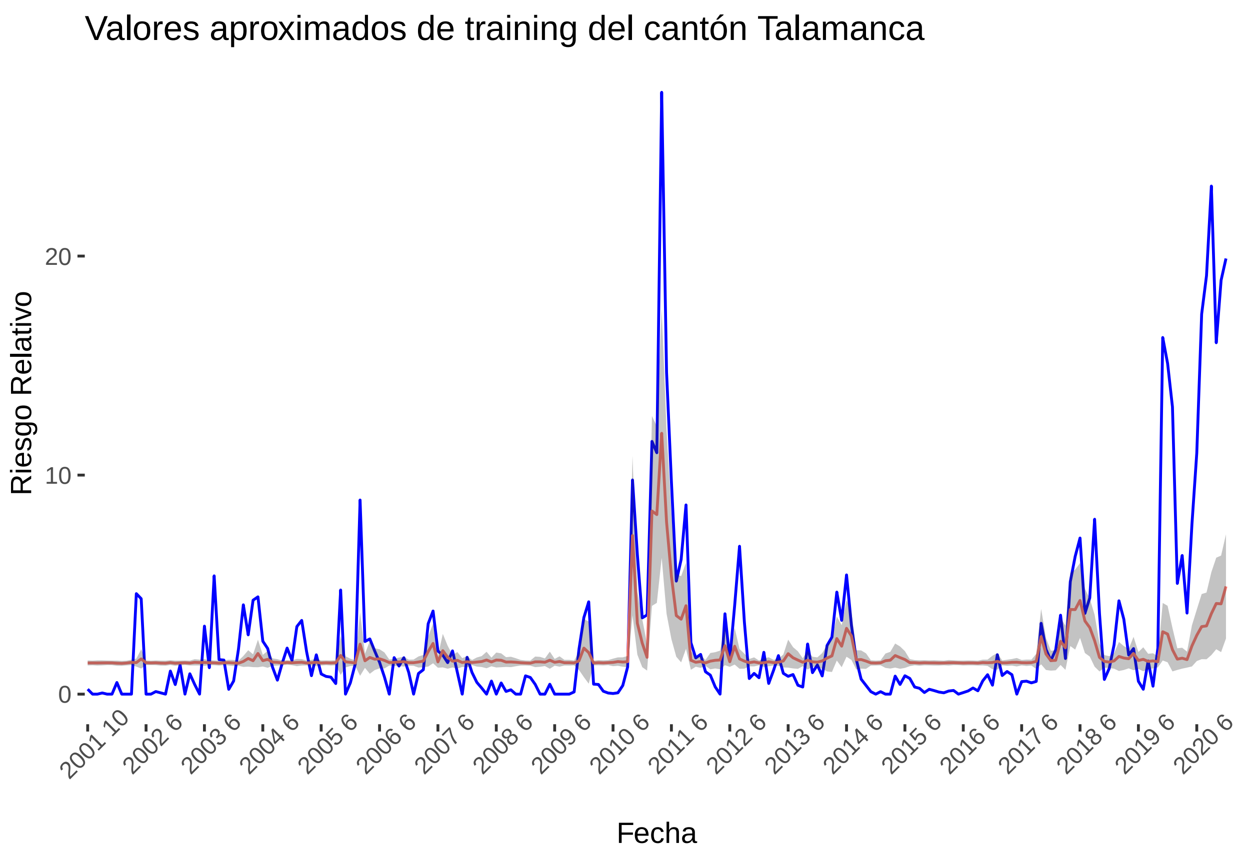

```
##
## [[28]]
```

Valores aproximados de training del cantón Sarapiquí
Riesgo Relativo
30
20
10
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha


```
##
## [[29]]
```


Valores aproximados de training del cantón Siquirres
Riesgo Relativo
15
10
5
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

```
##
## [[30]]
```

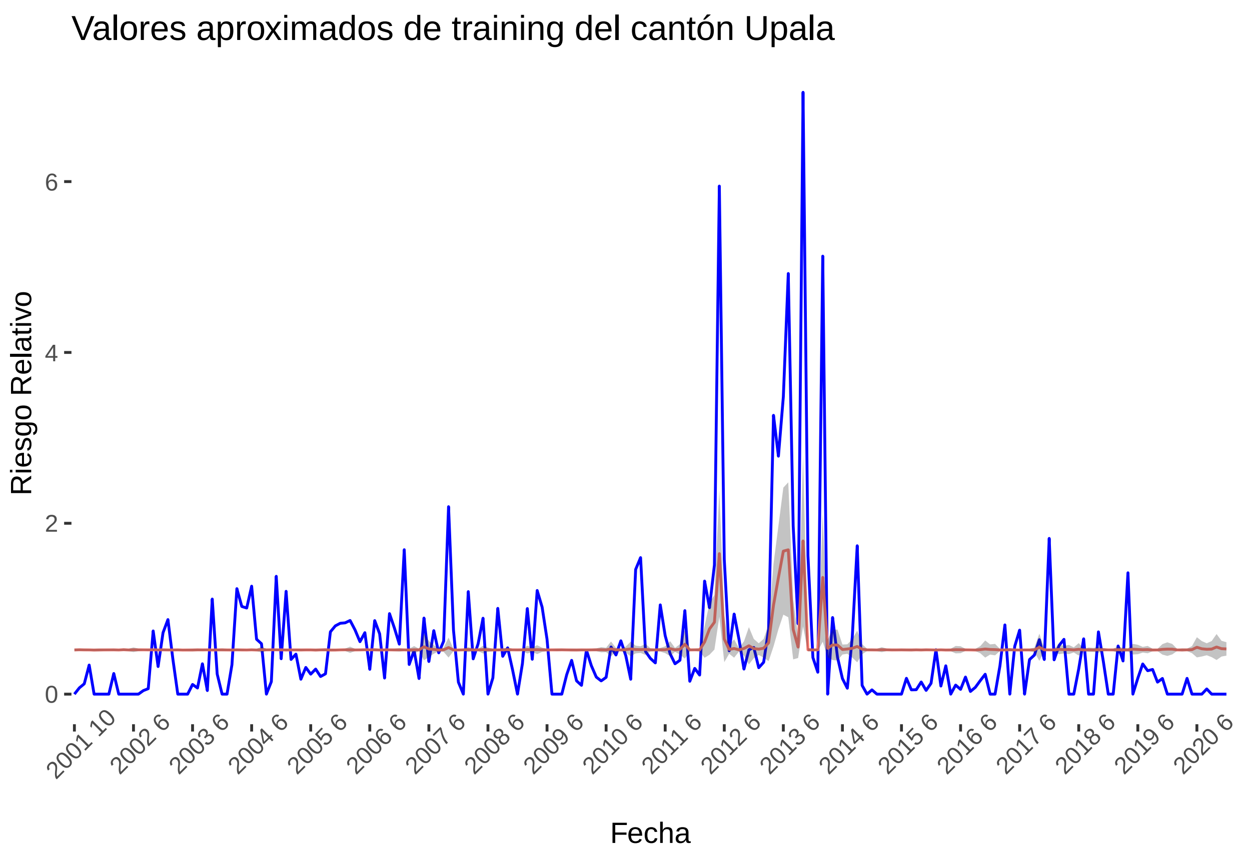


Valores aproximados de training del cantón Talamanca
Riesgo Relativo
20
10
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha


```
##
## [[31]]
```

Valores aproximados de training del cantón Turrialba
Riesgo Relativo
10
5
0
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha


```
##
## [[32]]
```



Predicciones

```
Predicciones
```

```
##        [,1]            [,2]                [,3]
##  [1,] "Alajuela"      "0.772327174083828" "0.906946022393552"
##  [2,] "Alajuela"      "0.827845321208599" "1.02110186285526"
##  [3,] "Alajuela"      "0.852146590627058" "1.04810821286744"
##  [4,] "Alajuelita"    "0.133619162514082" "0.166753695265876"
##  [5,] "Alajuelita"    "0.203720361673025" "0.335928515406815"
##  [6,] "Alajuelita"    "0.226255586542857" "0.385330695232055"
##  [7,] "Atenas"        "1.6520446574766"   "3.30816458661465"
##  [8,] "Atenas"        "1.00697379996308"  "2.20938200879165"
##  [9,] "Atenas"        "0.730226317851652" "1.71874782025966"
## [10,] "Cañas"         "2.51458316390509"  "2.69454094427034"
## [11,] "Cañas"         "2.47608660740546"  "2.67845482902407"
## [12,] "Cañas"         "2.47268921246286"  "2.66649257938477"
## [13,] "Carrillo"      "3.40696236749742"  "3.45809471464361"
## [14,] "Carrillo"      "3.40722364467774"  "3.46165253302398"
## [15,] "Carrillo"      "3.40725897768524"  "3.45580765979791"
## [16,] "Corredores"    "2.22779773597326"  "2.50088189718748"
## [17,] "Corredores"    "2.21670961691059"  "2.49299808176769"
## [18,] "Corredores"    "2.24002460540979"  "2.50632062825296"
## [19,] "Desamparados"  "0.114115593845374" "0.125888235196135"
```

```
## [20,] "Desamparados"  "0.108125182833705" "0.128022702841998"
## [21,] "Desamparados"  "0.101565042490031" "0.138927801951117"
## [22,] "Esparza"       "1.86356861237954"  "2.59322987475408"
## [23,] "Esparza"       "2.36325090353526"  "2.59460931935664"
## [24,] "Esparza"       "2.35940820015131"  "2.56927324812057"
## [25,] "Garabito"      "0.723647166906904" "2.70970864340052"
## [26,] "Garabito"      "0.418256084060738" "2.42838862593687"
## [27,] "Garabito"      "0.769916818644973" "2.76750985972391"
## [28,] "Golfito"       "1.53104732947216"  "1.72531316210221"
## [29,] "Golfito"       "1.53645384201602"  "1.73026456427445"
## [30,] "Golfito"       "1.52970237909421"  "1.81716801226845"
## [31,] "Guacimo"       "2.78292035969861"  "5.03594103532115"
## [32,] "Guacimo"       "2.88153942234201"  "5.11944675100217"
## [33,] "Guacimo"       "2.37488568428995"  "4.0073656024329"
## [34,] "La Cruz"       "1.40111831896358"  "1.41021261080375"
## [35,] "La Cruz"       "1.39883811131429"  "1.40965597612688"
## [36,] "La Cruz"       "1.39372073628224"  "1.40932740522057"
## [37,] "Liberia"       "0.974364284291267" "1.48522495448323"
## [38,] "Liberia"       "1.08241914234435"  "1.2832449465136"
## [39,] "Liberia"       "1.13177946453976"  "1.28624913205573"
## [40,] "Limon"         "1.41995729102312"  "1.67817492398546"
## [41,] "Limon"         "1.20278448935259"  "1.76287050574855"
## [42,] "Limon"         "1.25979164044433"  "1.69555830397141"
## [43,] "Matina"        "1.73201560416604"  "2.65502561312698"
## [44,] "Matina"        "1.42534064121336"  "2.33883412240112"
## [45,] "Matina"        "1.52453611775455"  "2.4946649055373"
## [46,] "Montes de Oro" "3.2840048119757"   "3.60523536778184"
## [47,] "Montes de Oro" "3.34931712308284"  "3.63528565041752"
## [48,] "Montes de Oro" "3.38821339995177"  "3.65313336731697"
## [49,] "Nicoya"        "1.27278432856846"  "1.31140771859656"
## [50,] "Nicoya"        "1.27302357514792"  "1.31141157202639"
## [51,] "Nicoya"        "1.26957805506514"  "1.30939253450058"
## [52,] "Orotina"       "5.20315934852474"  "5.42622766799323"
## [53,] "Orotina"       "5.2271764744776"   "5.43399743587774"
## [54,] "Orotina"       "5.1912361909555"   "5.42668600713702"
## [55,] "Osa"           "0.770452976594941" "0.880816492482187"
## [56,] "Osa"           "0.828899999189059" "0.867954749714411"
## [57,] "Osa"           "0.836054858780839" "0.86672726118578"
## [58,] "Parrita"       "5.1995373324729"   "6.52766603322847"
## [59,] "Parrita"       "5.33589758843673"  "6.59015362006882"
## [60,] "Parrita"       "5.0876986893662"   "6.43014085612839"
## [61,] "Perez Zeledón" "0.524335315461713" "0.55529250477923"
## [62,] "Perez Zeledón" "0.517692851325805" "0.55176594366391"
## [63,] "Perez Zeledón" "0.527771958048924" "0.557175199798467"
## [64,] "Pococí"        "2.38231788733731"  "2.53482674831598"
## [65,] "Pococí"        "2.36222252585349"  "2.54642465783853"
## [66,] "Pococí"        "2.40379498234441"  "2.5328540779386"
## [67,] "Puntarenas"    "1.41063893814758"  "1.69539118427901"
## [68,] "Puntarenas"    "1.46676327563055"  "1.62848628631669"
## [69,] "Puntarenas"    "1.50906666903332"  "1.61294936529354"
## [70,] "Quepos"        "2.76469852639702"  "3.15483646313398"
## [71,] "Quepos"        "2.78157296390363"  "3.14687814198977"
## [72,] "Quepos"        "2.7713278860824"   "3.17288777001975"
## [73,] "San Jose"      "0.278141177801272" "0.387058669618319"
```

```
## [74,] "San Jose"      "0.262280911570088" "0.375329825033688"
## [75,] "San Jose"      "0.232696607269142" "0.299895513950076"
## [76,] "Santa Ana"     "0.305722577971651" "0.333372033891959"
## [77,] "Santa Ana"     "0.306544585685018" "0.334648010103201"
## [78,] "Santa Ana"     "0.305467598913898" "0.33640198974901"
## [79,] "SantaCruz"     "3.85346696907648"  "4.4222970840821"
## [80,] "SantaCruz"     "3.87912269103195"  "4.43434192443204"
## [81,] "SantaCruz"     "3.78726707805677"  "4.37583568688024"
## [82,] "Sarapiquí"     "3.03379811901415"  "3.39391192827444"
## [83,] "Sarapiquí"     "2.98523744959292"  "3.37896084659143"
## [84,] "Sarapiquí"     "3.06882824242354"  "3.41853066590886"
## [85,] "Siquirres"     "1.69929440351679"  "2.45185405683184"
## [86,] "Siquirres"     "1.58520696658793"  "2.19041550016808"
## [87,] "Siquirres"     "1.55571664207103"  "2.11351566941658"
## [88,] "Talamanca"     "1.72910688705216"  "3.56247961241136"
## [89,] "Talamanca"     "1.74058909444021"  "3.54267970080278"
## [90,] "Talamanca"     "2.29130185197547"  "4.56732557633052"
## [91,] "Turrialba"     "0.556808996307577" "0.556808996307577"
## [92,] "Turrialba"     "0.556808996307577" "0.556808996307577"
## [93,] "Turrialba"     "0.556808996307577" "0.556808996307577"
## [94,] "Upala"         "0.375837589864489" "0.575384354798262"
## [95,] "Upala"         "0.434583303445377" "0.536959681076533"
## [96,] "Upala"         "0.4596872689148"   "0.522992228098648"
##       [,4]
##  [1,] "1.04156487070328"
##  [2,] "1.21435840450192"
##  [3,] "1.24406983510782"
##  [4,] "0.19988822801767"
##  [5,] "0.468136669140605"
##  [6,] "0.544405803921253"
##  [7,] "4.9642845157527"
##  [8,] "3.41179021762023"
##  [9,] "2.70726932266766"
## [10,] "2.87449872463559"
## [11,] "2.88082305064268"
## [12,] "2.86029594630669"
## [13,] "3.5092270617898"
## [14,] "3.51608142137023"
## [15,] "3.50435634191058"
## [16,] "2.77396605840171"
## [17,] "2.7692865466248"
## [18,] "2.77261665109612"
## [19,] "0.137660876546895"
## [20,] "0.147920222850291"
## [21,] "0.176290561412202"
## [22,] "3.32289113712862"
## [23,] "2.82596773517803"
## [24,] "2.77913829608983"
## [25,] "4.69577011989413"
## [26,] "4.43852116781301"
## [27,] "4.76510290080284"
## [28,] "1.91957899473227"
## [29,] "1.92407528653288"
## [30,] "2.10463364544268"
```

```
## [31,] "7.28896171094369"
## [32,] "7.35735407966232"
## [33,] "5.63984552057585"
## [34,] "1.41930690264393"
## [35,] "1.42047384093946"
## [36,] "1.42493407415891"
## [37,] "1.9960856246752"
## [38,] "1.48407075068285"
## [39,] "1.4407187995717"
## [40,] "1.9363925569478"
## [41,] "2.32295652214452"
## [42,] "2.13132496749848"
## [43,] "3.57803562208792"
## [44,] "3.25232760358887"
## [45,] "3.46479369332004"
## [46,] "3.92646592358799"
## [47,] "3.9212541777522"
## [48,] "3.91805333468217"
## [49,] "1.35003110862466"
## [50,] "1.34979956890486"
## [51,] "1.34920701393603"
## [52,] "5.64929598746172"
## [53,] "5.64081839727788"
## [54,] "5.66213582331854"
## [55,] "0.991180008369433"
## [56,] "0.907009500239764"
## [57,] "0.89739966359072"
## [58,] "7.85579473398403"
## [59,] "7.84440965170091"
## [60,] "7.77258302289058"
## [61,] "0.586249694096747"
## [62,] "0.585839036002015"
## [63,] "0.586578441548011"
## [64,] "2.68733560929466"
## [65,] "2.73062678982356"
## [66,] "2.6619131735328"
## [67,] "1.98014343041044"
## [68,] "1.79020929700282"
## [69,] "1.71683206155376"
## [70,] "3.54497439987095"
## [71,] "3.51218332007591"
## [72,] "3.57444765395711"
## [73,] "0.495976161435366"
## [74,] "0.488378738497288"
## [75,] "0.36709442063101"
## [76,] "0.361021489812267"
## [77,] "0.362751434521384"
## [78,] "0.367336380584121"
## [79,] "4.99112719908772"
## [80,] "4.98956115783214"
## [81,] "4.96440429570371"
## [82,] "3.75402573753472"
## [83,] "3.77268424358994"
## [84,] "3.76823308939418"
```

```
## [85,] "3.20441371014689"
## [86,] "2.79562403374823"
## [87,] "2.67131469676214"
## [88,] "5.39585233777056"
## [89,] "5.34477030716535"
## [90,] "6.84334930068558"
## [91,] "0.556808996307577"
## [92,] "0.556808996307577"
## [93,] "0.556808996307577"
## [94,] "0.774931119732034"
## [95,] "0.63933605870769"
## [96,] "0.586297187282496"
```