Analisis Overfitting sin IC

Jimena Murillo

2022-06-16

Paquetes

```
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()
```

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

```
## Loading required package: lattice

##
## Attaching package: 'caret'

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

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

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

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

```
library(neuralnet)
```

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

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

Datos

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


Alajuela <- basecanton %>% filter(Canton == "Alajuela") %>%

  dplyr::select(Year,Month,Nino12SSTA, Nino3SSTA, Nino4SSTA,Nino34SSTA,Nino34SSTA1, Nino34SSTA2, Nino34S

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


if(anyNA(Alajuela)){
  Alajuela <- na.omit(Alajuela)
}

#Escala


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

Alajuela2 <- apply(Alajuela, 2, normalize)
```

```
#Train y test

Fechas = c(1, 0.95, 0.90, 0.85, 0.80, 0.75, 0.70, 0.65, 0.60, 0.55, 0.50)
Eval = NULL
Evalc = NULL
Eval3 = NULL

p1 = list()
p2 = list()
p3 = list()

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

data_train = as.data.frame(Alajuela2) %>% filter(Year < Fechas[i])#PARA ENTRENAR HASTA 2018
data_test = as.data.frame(Alajuela2) %>% filter(Year >= Fechas[i])
```

```
X_train = as.matrix(data_train[,-ncol(data_train)])
y_train = as.matrix(data_train[,ncol(data_train)])

X_test = as.matrix(data_test[,-ncol(data_test)])
y_test = as.matrix(data_test[,ncol(data_test)])

## Modelo

set.seed(123)
model <- keras_model_sequential()

model %>%
  layer_simple_rnn(units = 100, input_shape = c(ncol(X_train),1), activation='tanh',
                   kernel_initializer= initializer_constant(0.5),
                   bias_initializer=initializer_zeros()) %>%
  layer_dense(units = 50, activation = "relu")%>%
  layer_dense(units = 50, activation = "relu")%>%
  layer_dense(units = 50, activation = "relu")%>%
  layer_dropout(rate = 0.1)%>%
  layer_dense(units = 25, activation = "relu")%>%
  layer_dense(units = 25, activation = "relu")%>%
  layer_dense(units = 25, activation = "relu")%>%
  layer_dropout(rate = 0.1)%>%
  layer_dense(units = 12, activation = "relu")%>%
  layer_dense(units = 12, activation = "relu")%>%
  layer_dropout(rate = 0.1)%>%
  layer_dense(units = 6, activation = "relu")%>%
  layer_dense(units = 6, activation = "relu")%>%
  layer_dense(units = 1, activation = "sigmoid")

## Entrenar al modelo


model %>% compile(
  optimizer = "adam",
  loss = "mse",
  metrics = "mae")

history <- model %>% fit(
  X_train,
  y_train,
  epochs = 100,
  batch_size = 18,
  validation_split = 0.1,
  shuffle = F
)

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

pred = denorm(model %>% predict(Alajuela2[,-33]))
results = denorm(model %>% predict(X_test))
```

```
results3 = denorm(model %>% predict(Alajuela2[233:235, -33]))

#Grafico

data1 = as.data.frame(cbind(pred, Alajuela$RR))
names(data1) = c("fit", "RR")
data2 = as.data.frame(cbind(results, Alajuela$RR[(236-length(results)):235]))
names(data2) = c("fit", "RR")
data3 = as.data.frame (cbind(results3, Alajuela$RR[233:235]))
names(data3) = c("fit", "RR")

Fecha = paste(Alajuela$Year, Alajuela$Month)

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

p1[[i]] <- ggplot(data1, aes(x = Fecha, y = RR, group = 1)) + geom_line(colour = "blue") +
  geom_line( aes(x = Fecha, y = fit, 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("Predicción desde", Fechas[i], sep = ": "))

p3[[i]] <- ggplot(data3, aes(x = Fecha[233:235], y = RR, group = 1)) + geom_line(colour = "blue") +
  geom_line( aes(x = Fecha[233:235], y = fit, 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("Predicción 3 meses training hasta", Fechas



metricas <- function(tabla){
  NRMSE <- mean((tabla$fit-tabla$RR)^2)/mean(tabla$RR)
  return(data.frame(NRMSE))
}

Eval[i] = as.numeric(metricas(data1))

Evalc[i] = as.numeric(metricas(data2))

Eval3[[i]] = as.numeric(metricas(data3))

k_clear_session()
}
```

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

Resultados

```
NRMSE = cbind (as.numeric(Eval), as.numeric(Evalc), as.numeric(Eval3))
colnames(NRMSE) = c("Total", "Test", "Solo 2021")
```

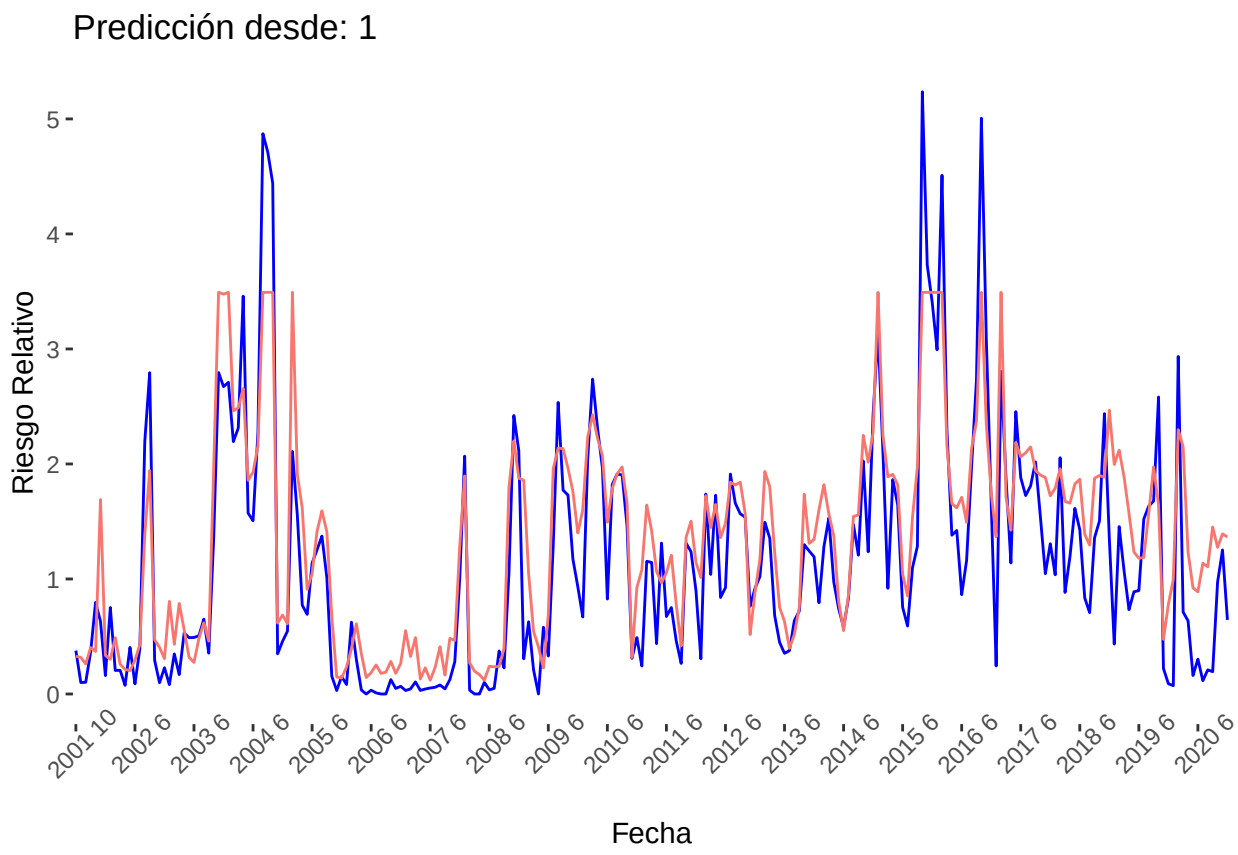
```
rownames(NRMSE) = c("2021", "2020 +", "2019 +", "2018 +", "2017+", "2016+", "2015+", "2014+", "2013+",
as.data.frame(NRMSE)
```

```
##            Total      Test Solo 2021
## 2021   0.2352648 0.2194257 0.2194257
## 2020 + 0.1651299 0.9868863 0.2449367
## 2019 + 0.3470640 0.6459621 0.4032850
## 2018 + 1.2117933 0.8252234 0.6526252
## 2017+  0.3945307 0.6604056 0.6829186
## 2016+  0.4419578 0.7750207 0.4461088
## 2015+  0.4594315 0.6016971 0.6422070
## 2014+  0.4267118 0.6738415 0.4233081
## 2013+  0.3921874 0.5140568 0.2723520
## 2012+  0.5000053 0.6555225 0.2895097
## 2011+  0.4709411 0.5149570 0.4264260
```

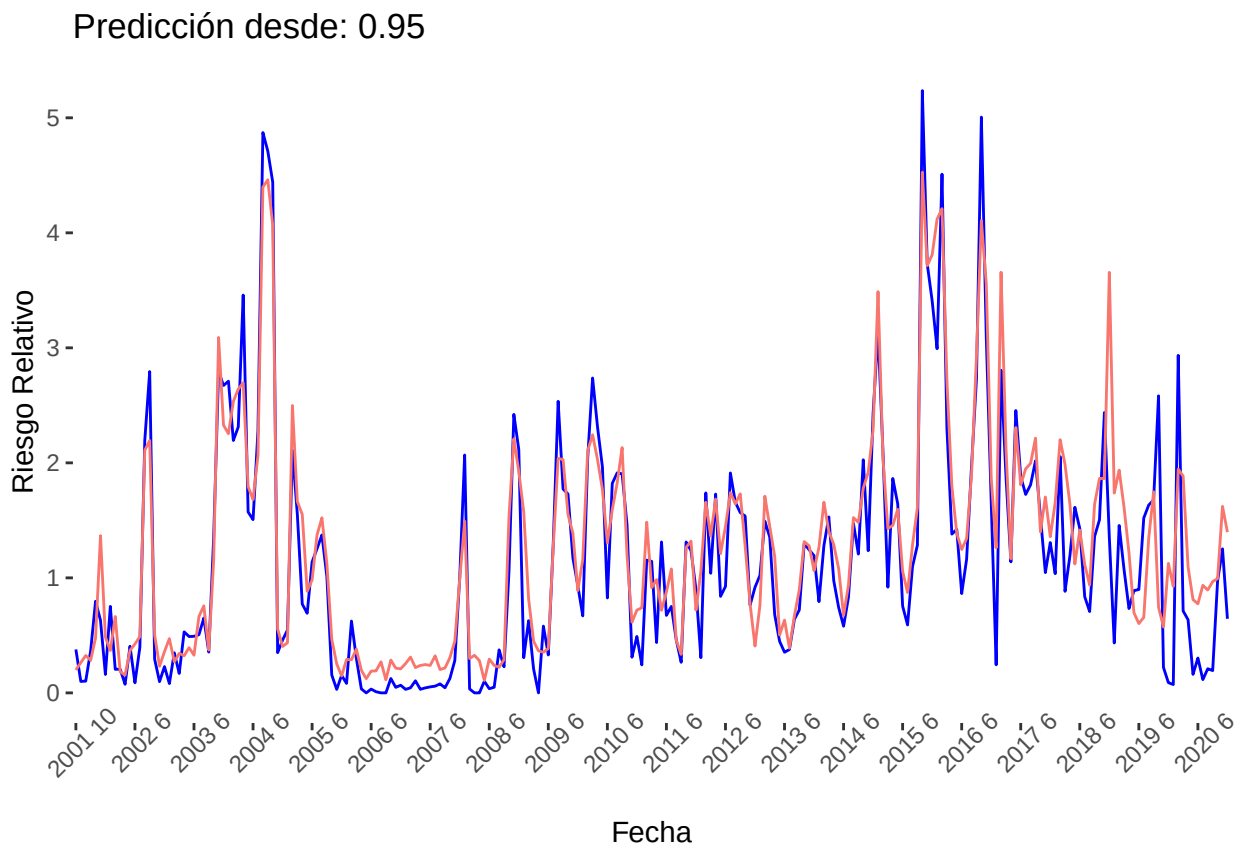
Gráficos

```
p1
```

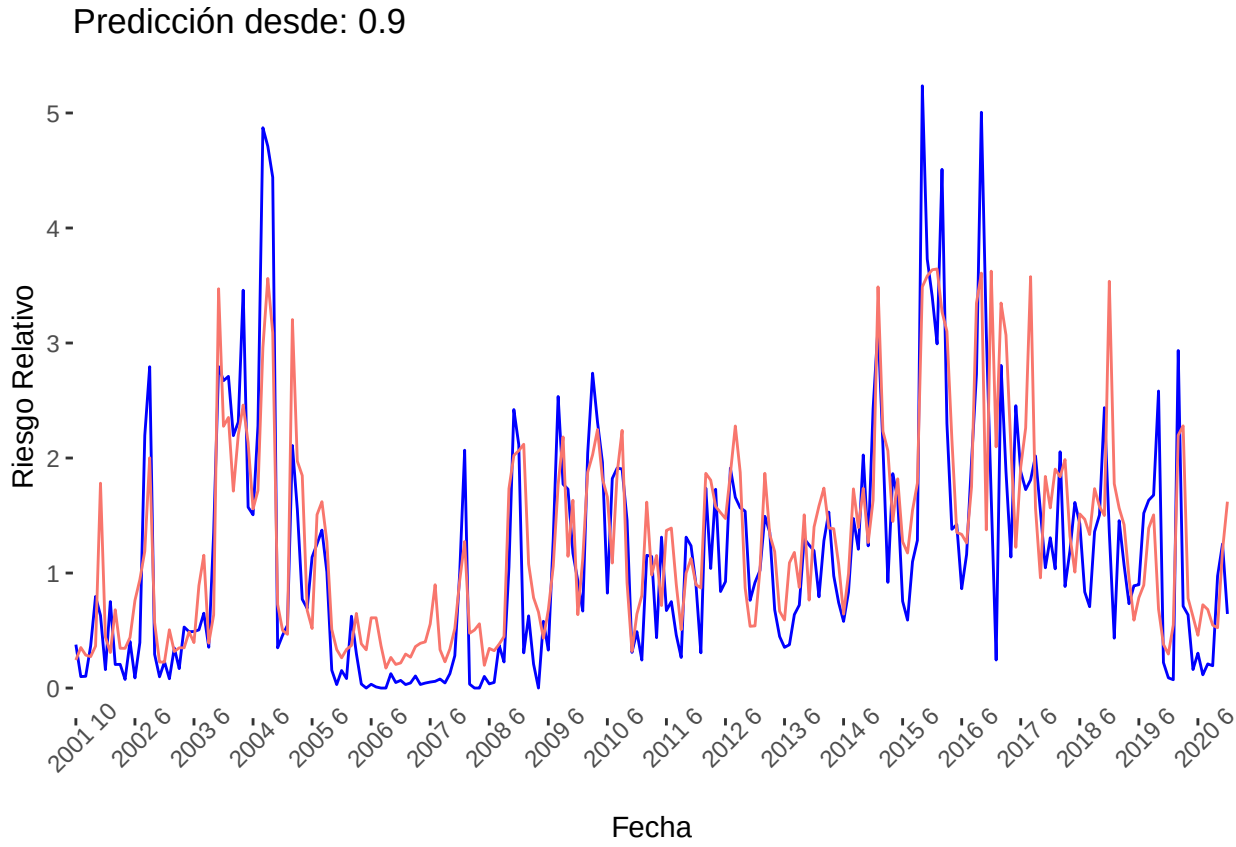
```
## [[1]]
```



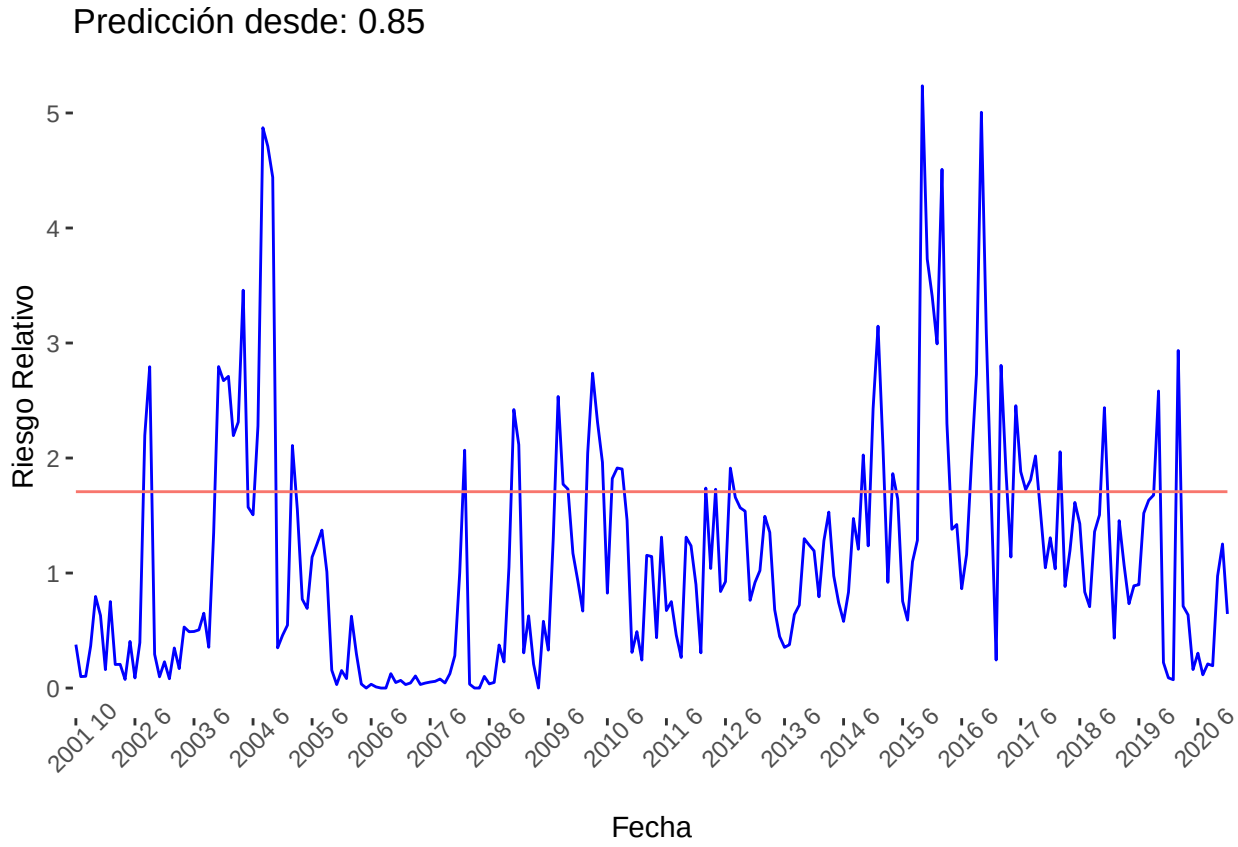
```
##
## [[2]]
```


Predicción desde: 0.95
Riesgo Relativo
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
Fecha

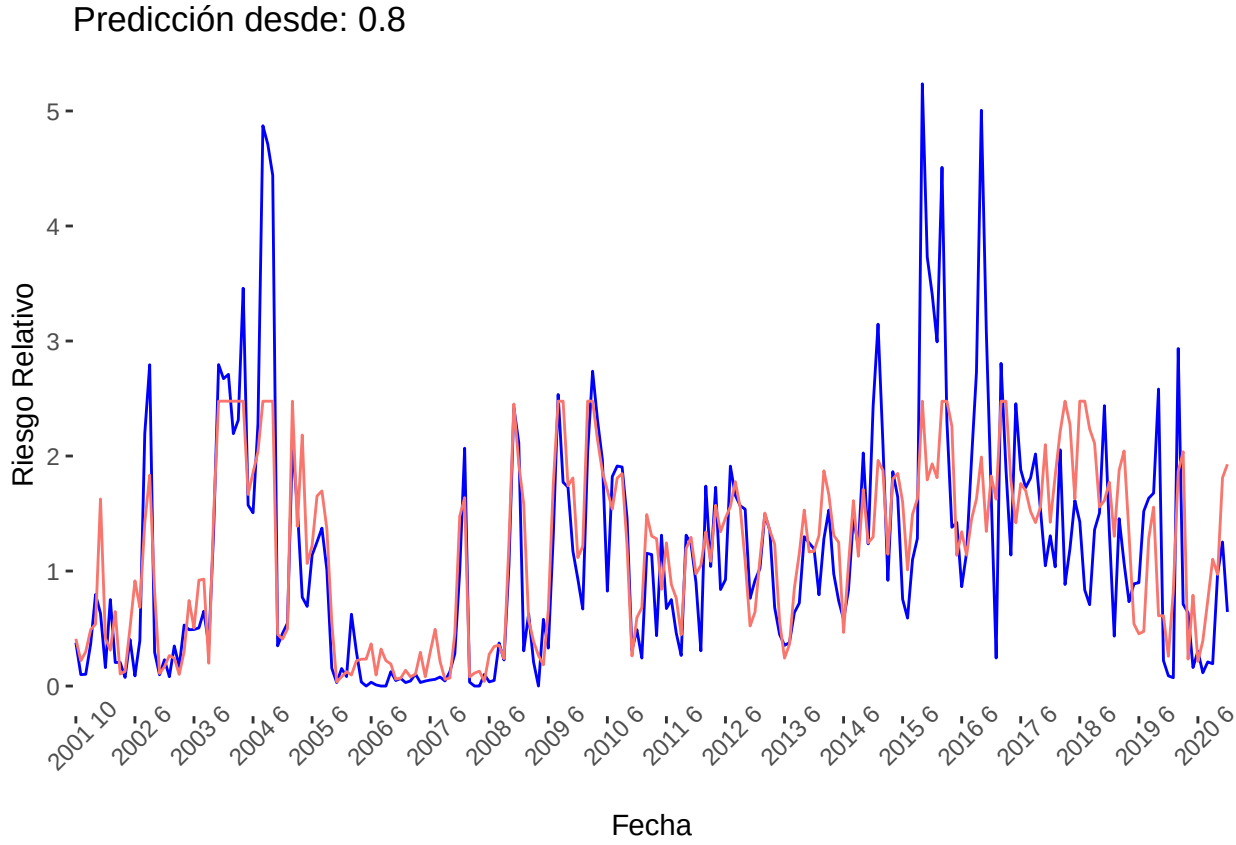
```
##
## [[3]]
```


Predicción desde: 0.9
Riesgo Relativo
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
Fecha

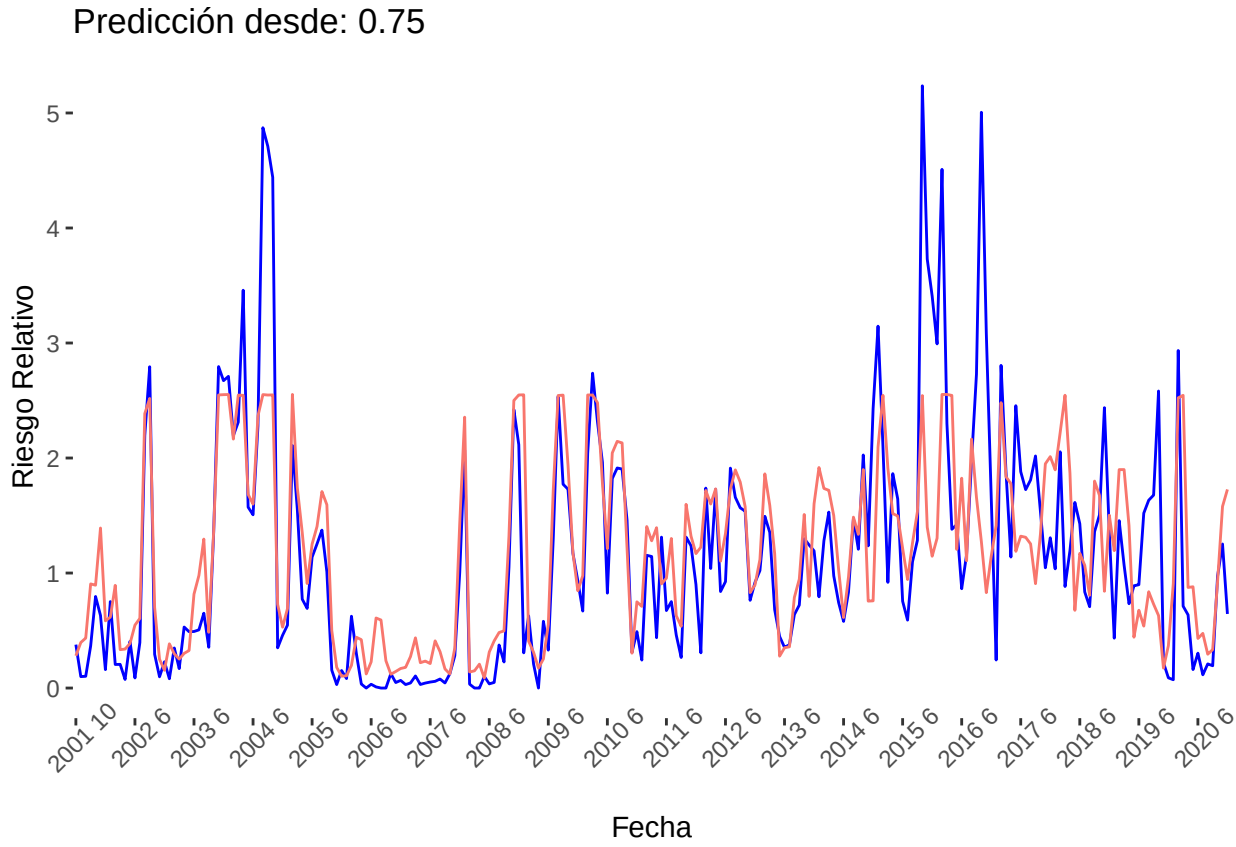
```
##
## [[4]]
```


Predicción desde: 0.85
Riesgo Relativo
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
Fecha

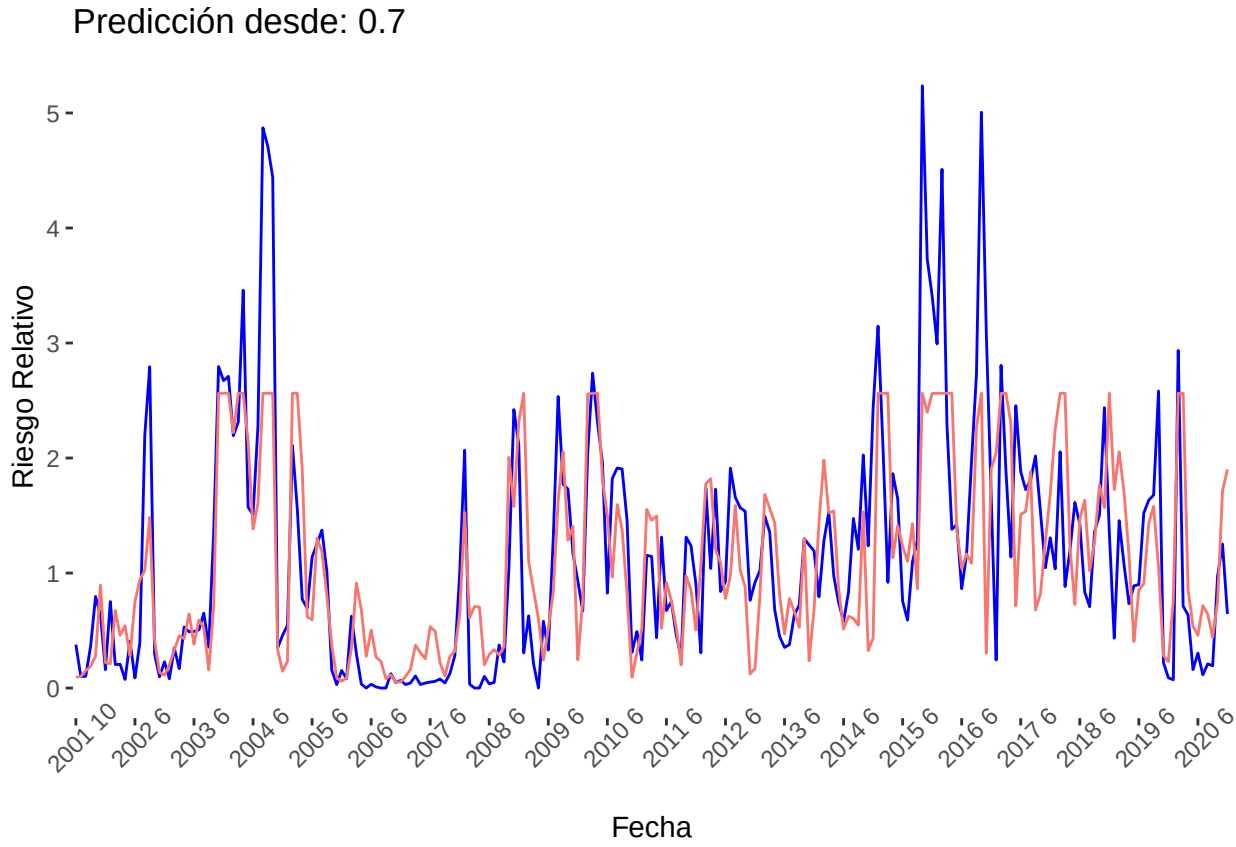
```
##
## [[5]]
```


Predicción desde: 0.8
Riesgo Relativo
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
Fecha

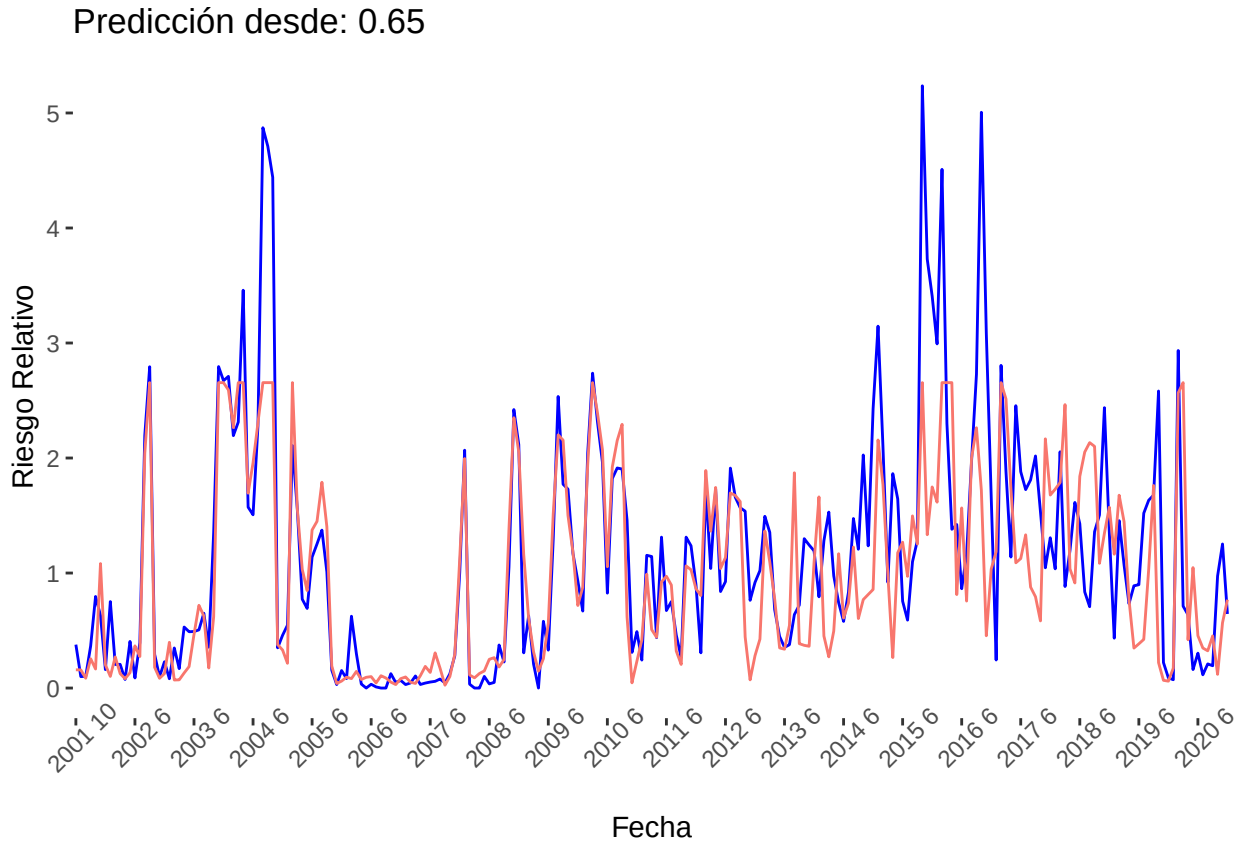
```
##
## [[6]]
```


Predicción desde: 0.75
Riesgo Relativo
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
Fecha

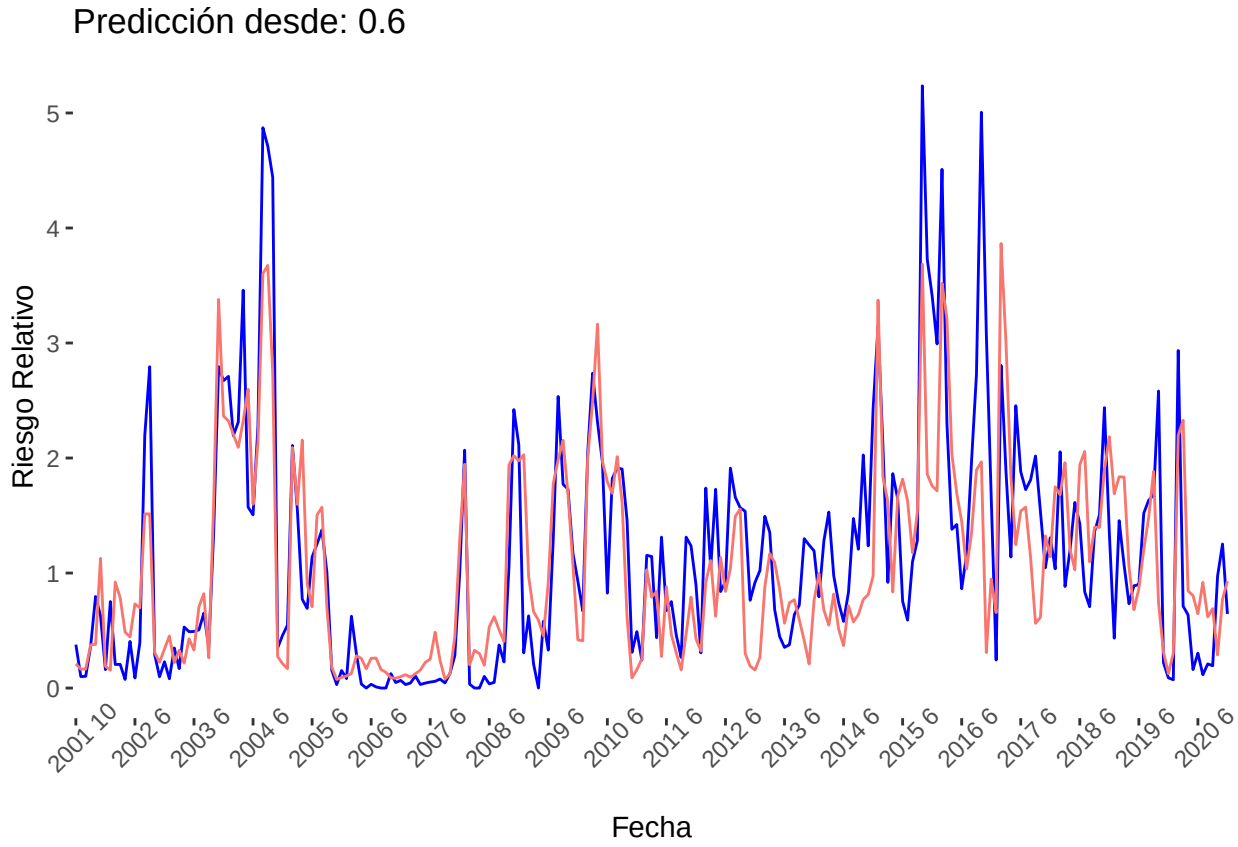
```
##
## [[7]]
```


Predicción desde: 0.7
Riesgo Relativo
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
Fecha

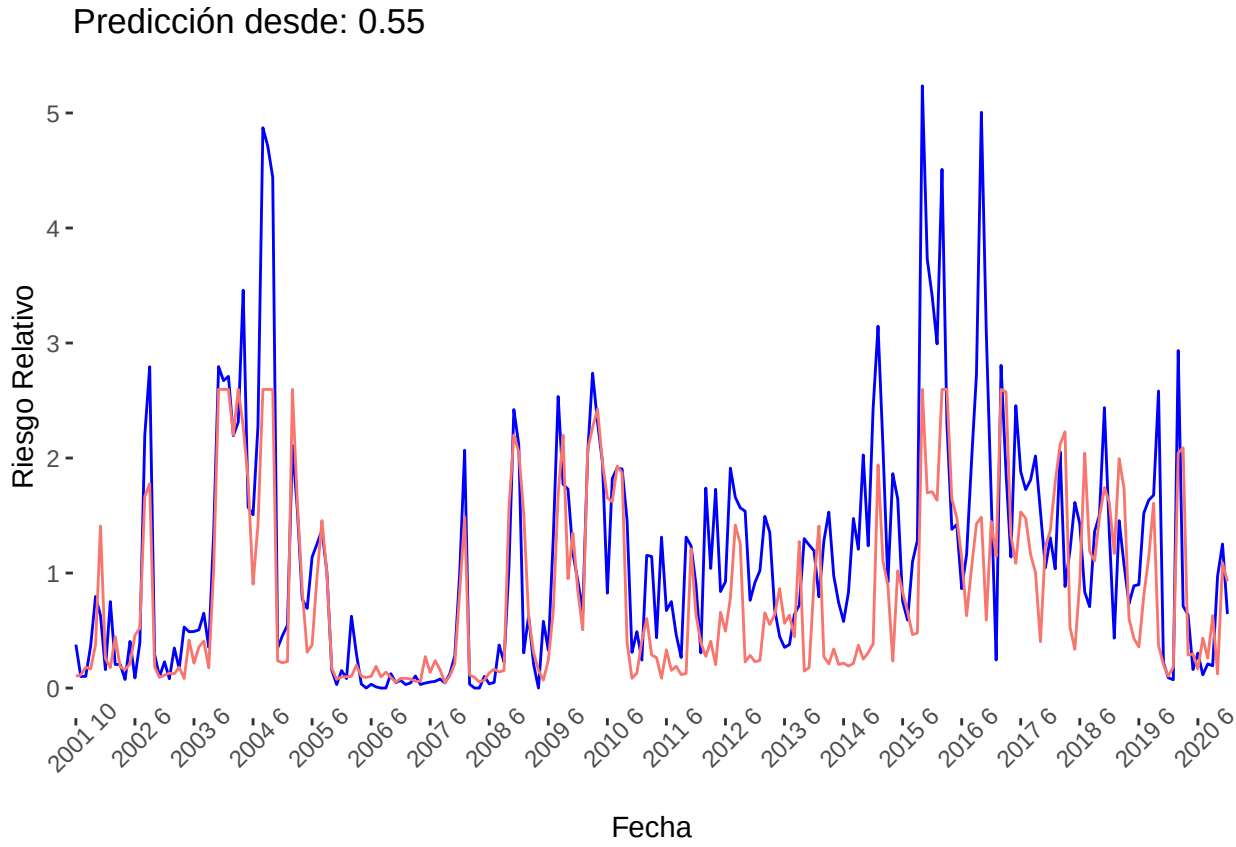
```
##
## [[8]]
```


Predicción desde: 0.65
Riesgo Relativo
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
Fecha

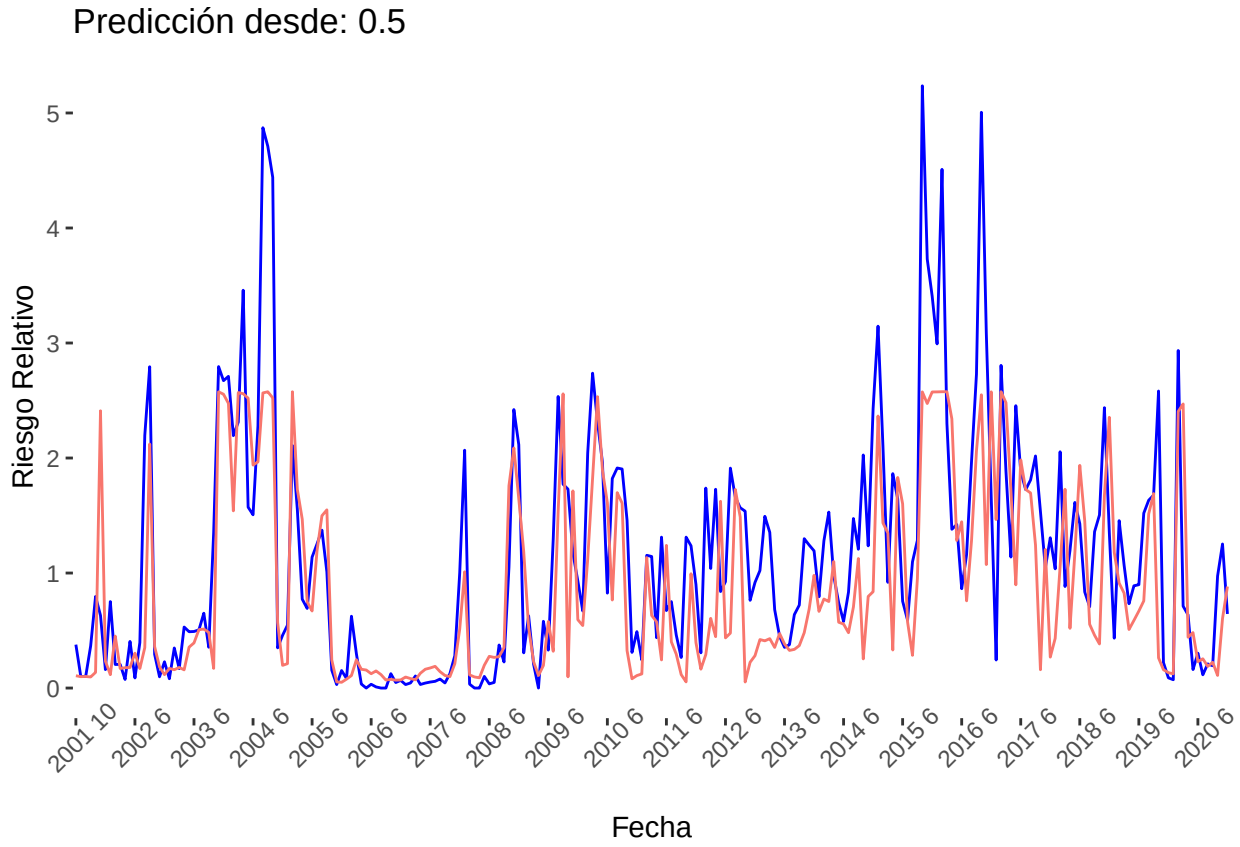
```
##
## [[9]]
```


Predicción desde: 0.6
Riesgo Relativo
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
Fecha

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

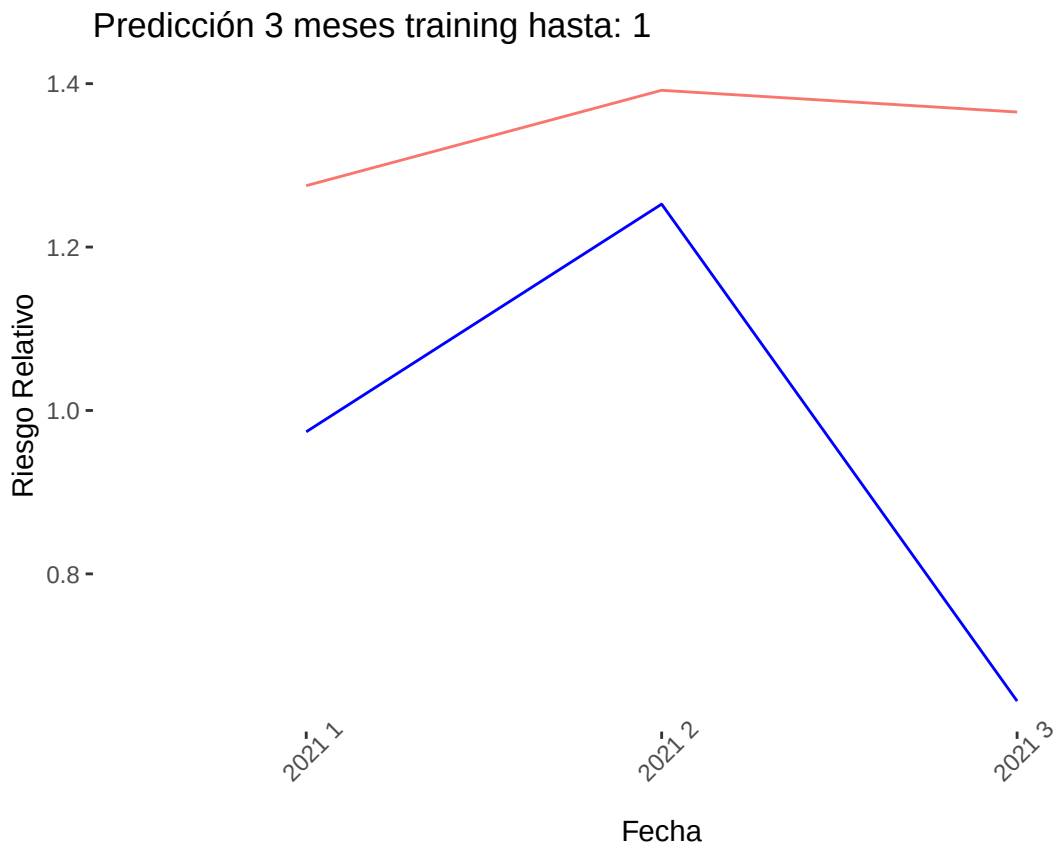

Predicción desde: 0.55
Riesgo Relativo
0
1
2
3
4
5
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

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

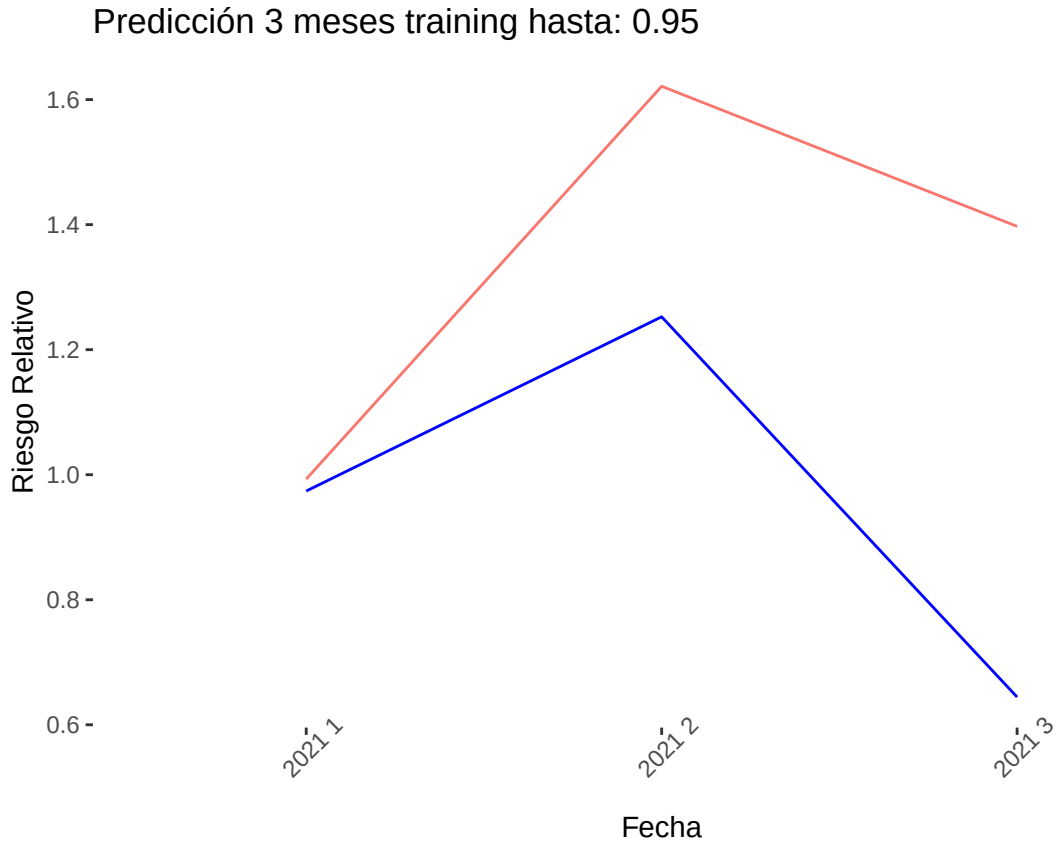

Predicción desde: 0.5
Riesgo Relativo
0
1
2
3
4
5
2001 10
2002 6
2003 6
2004 6
2005 6
2006 6
2007 6
2008 6
2009 6
2010 6
2011 6
2012 6
2013 6
2014 6
2015 6
2016 6
2017 6
2018 6
2019 6
2020 6
Fecha

```
p3
```

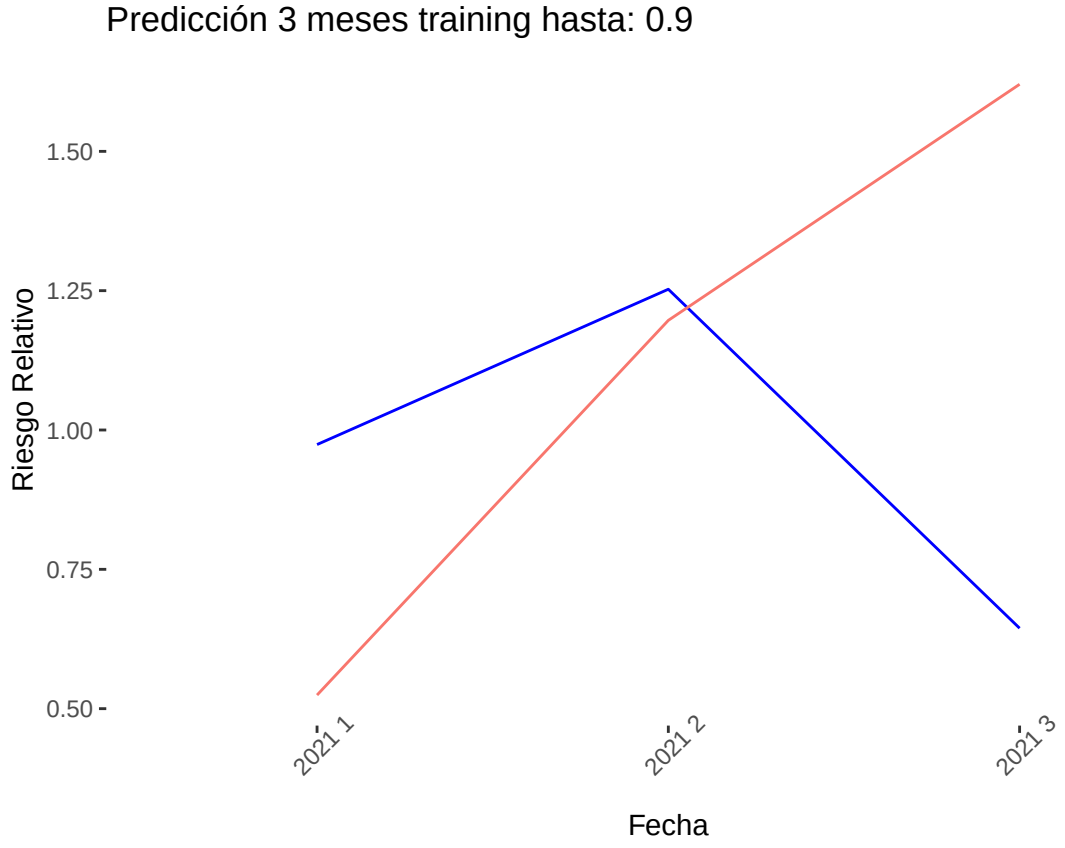
```
## [[1]]
```


Predicción 3 meses training hasta: 1
1.4
1.2
1.0
0.8
Riesgo Relativo
2021-1
2021-2
2021-3
Fecha

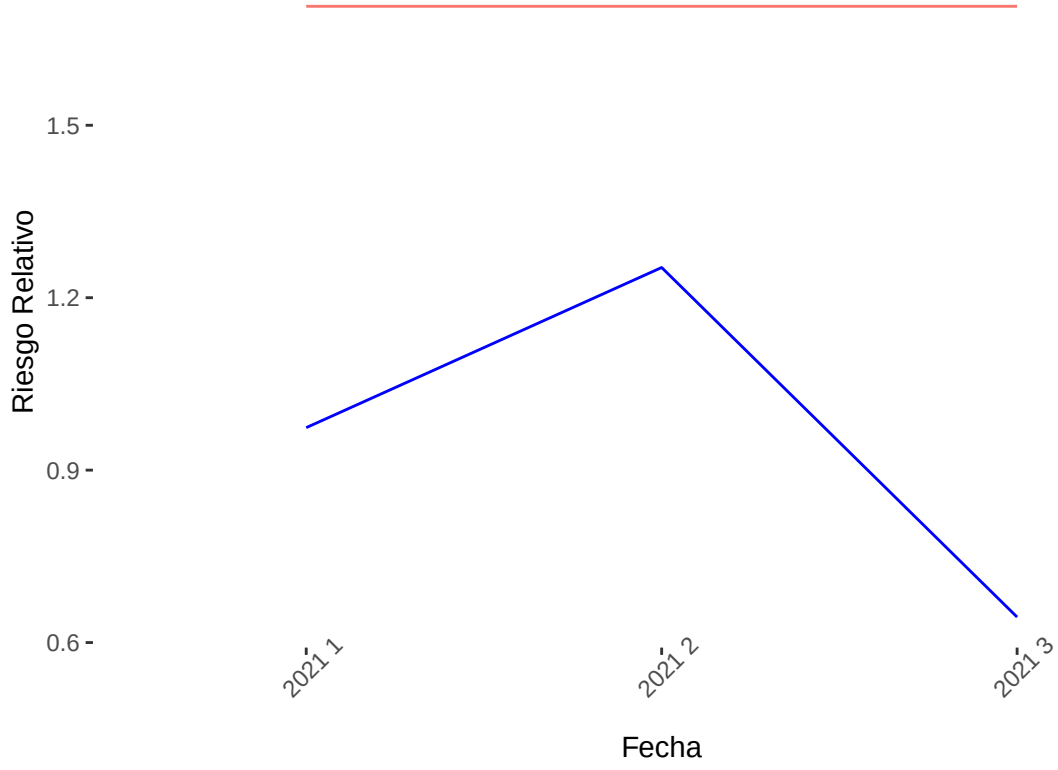
```
##
## [[2]]
```


Predicción 3 meses training hasta: 0.95
Riesgo Relativo
1.6
1.4
1.2
1.0
0.8
0.6
2021-1
2021-2
2021-3
Fecha

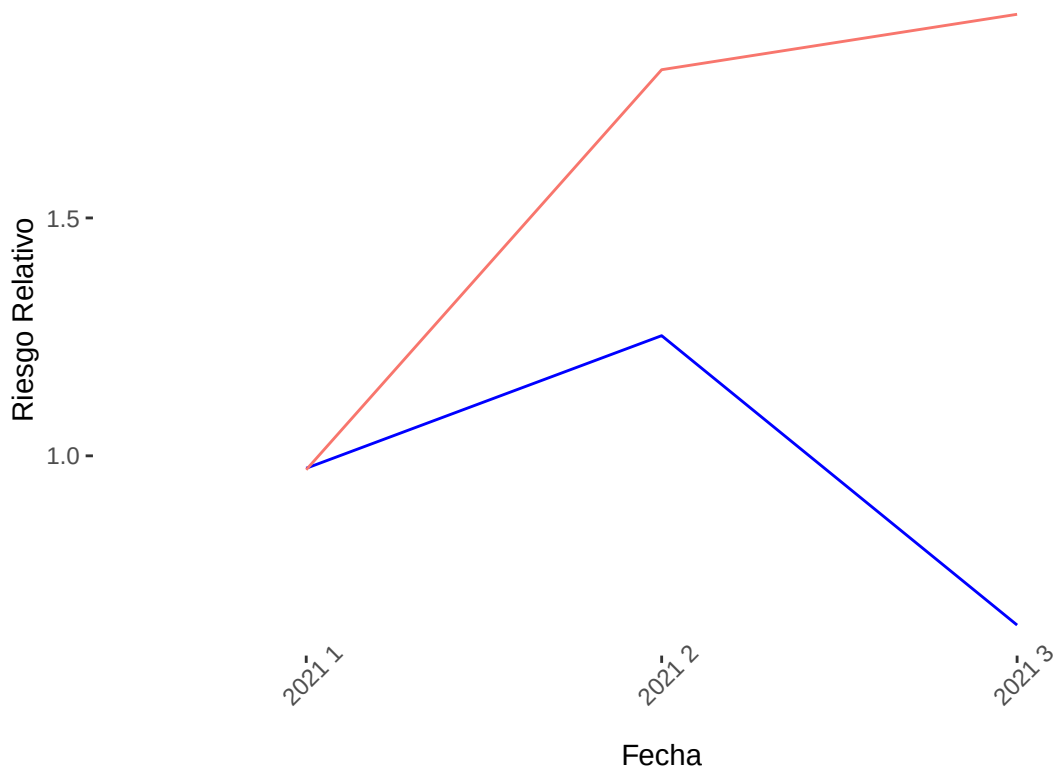
```
##
## [[3]]
```


Predicción 3 meses training hasta: 0.9
Riesgo Relativo
1.50
1.25
1.00
0.75
0.50
2021-1
2021-2
2021-3
Fecha

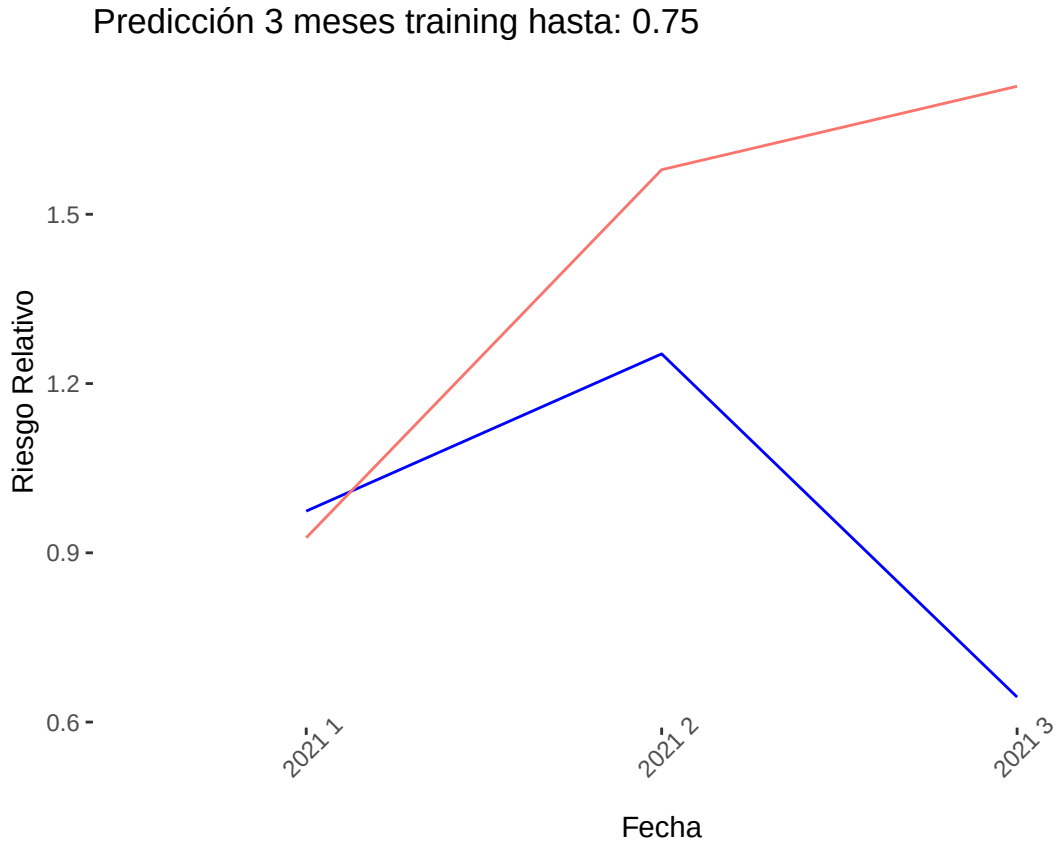
```
##
## [[4]]
```


Predicción 3 meses training hasta: 0.85
1.5
1.2
0.9
0.6
Riesgo Relativo
2021-1
2021-2
2021-3
Fecha

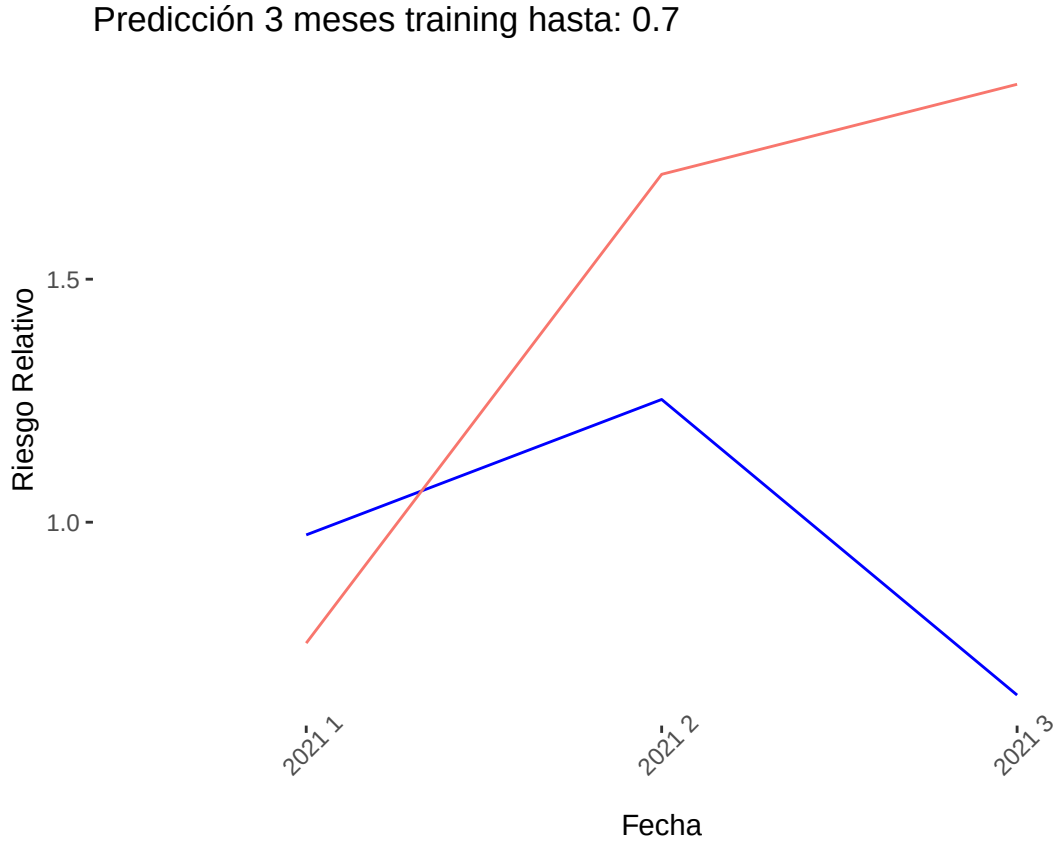
```
##
## [[5]]
```


Predicción 3 meses training hasta: 0.8
Riesgo Relativo
1.5
1.0
2021-1
2021-2
2021-3
Fecha

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


Predicción 3 meses training hasta: 0.75
Riesgo Relativo
1.5
1.2
0.9
0.6
2021-1
2021-2
2021-3
Fecha

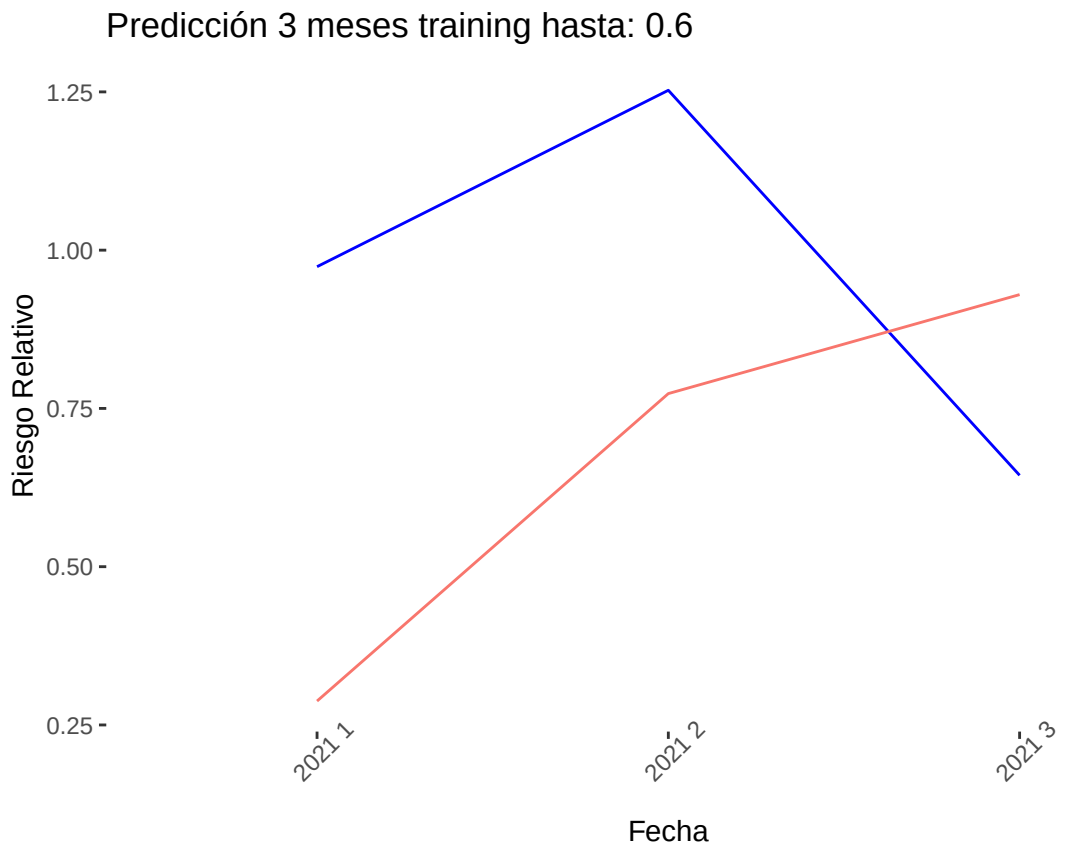
```
##
## [[7]]
```


Predicción 3 meses training hasta: 0.7
Riesgo Relativo
1.5
1.0
2021-1
2021-2
2021-3
Fecha

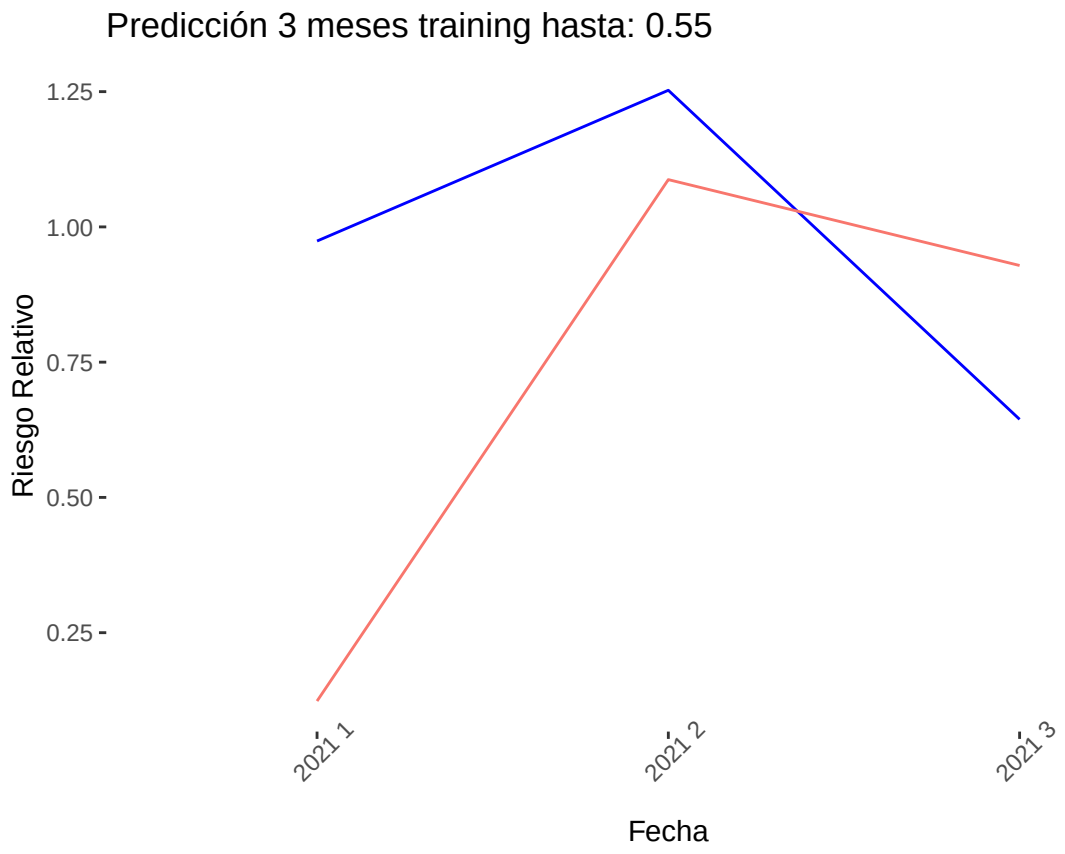
```
##
## [[8]]
```


Predicción 3 meses training hasta: 0.65
1.25
1.00
0.75
0.50
0.25
Riesgo Relativo
2021-1
2021-2
2021-3
Fecha

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


Predicción 3 meses training hasta: 0.6
1.25
1.00
0.75
0.50
0.25
Riesgo Relativo
2021-1
2021-2
2021-3
Fecha

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


Predicción 3 meses training hasta: 0.55
1.25
1.00
0.75
0.50
0.25
Riesgo Relativo
2021-1
2021-2
2021-3
Fecha

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


Predicción 3 meses training hasta: 0.5
1.25
1.00
0.75
0.50
0.25
Riesgo Relativo
2021-1
2021-2
2021-3
Fecha