

# Código para la creación del modelo Arima

Raúl Frugone Zaror, Angel Llanos Herrera y Felipe Neira Rojas

2024-09-12

## Contents

<b>Librerías, datos y funciones a utilizar</b>	<b>2</b>
Librerías . . . . .	2
Funciones . . . . .	2
Datos . . . . .	3
<b>Modelo con el 50% de los datos para predecir hasta el 60%</b>	<b>4</b>
Filtrar datos y escalar precipitaciones . . . . .	4
Predicción . . . . .	4
Test de hipótesis para los errores . . . . .	6
<b>Modelo con el 60% de los datos para predecir hasta el 70%</b>	<b>7</b>
Filtrar datos y escalar precipitaciones . . . . .	7
Predicción . . . . .	7
Test de hipótesis para los errores . . . . .	8
<b>Modelo con el 70% de los datos para predecir hasta el 80%</b>	<b>9</b>
Filtrar datos y escalar precipitaciones . . . . .	9
Predicción . . . . .	9
Test de hipótesis para los errores . . . . .	10
<b>Modelo con el 80% de los datos para predecir hasta el 90%</b>	<b>10</b>
Filtrar datos y escalar precipitaciones . . . . .	10
Predicción . . . . .	11
Test de hipótesis para los errores . . . . .	12
<b>Modelo con el 70% de los datos para predecir el 100%</b>	<b>12</b>
Filtrar datos y escalar precipitaciones . . . . .	12
Predicción . . . . .	13
Test de hipótesis para los errores . . . . .	14

<b>Modelo con el 80% de los datos para predecir el 100%</b>	<b>15</b>
Filtrar datos y escalar precipitaciones . . . . .	15
Predicción . . . . .	15
Test de hipotesis para los errores . . . . .	17
<b>Modelo predictivo con la reconstrucción del dique en 2026</b>	<b>17</b>
Filtrar datos y escalar precipitaciones . . . . .	17
Predicción . . . . .	18
Test de hipotesis para los errores . . . . .	20
<b>Modelo predictivo sin una reinstalación del dique</b>	<b>21</b>
Filtrar datos y escalar precipitaciones . . . . .	21
Predicción . . . . .	21
Test de hipotesis para los errores . . . . .	23

El presente documento tiene como objetivo el desarrollo de un modelo predictivo con base en el metodo ARIMA para prever variaciones en la superficie de un lago afectada por condiciones climaticas, humanas y geograficas. Utilizando datos historicos de precipitaciones, temperaturas maximas y la presencia o ausencia de un dique desde el año 1984 hasta 2022.

## Librerias, datos y funciones a utilizar

### Librerias

Las funciones utilizadas requieren de la libreria “forecast”

```
# Instalar y cargar el paquete forecast si no está instalado
if (!require(forecast)){install.packages("forecast")}
```

```
## Cargando paquete requerido: forecast
```

```
## Registered S3 method overwritten by 'quantmod':
##   method      from
##   as.zoo.data.frame zoo
```

```
library(forecast)
```

### Funciones

#### Escalar precipitaciones

se define una función `escalar_precipitaciones` que normaliza los datos de precipitaciones. Este método de normalización permite centrar los datos cercano a cero dismuyendo la dispersión.

```
# Escalar precipitaciones
escalar_precipitaciones <- function(precipitaciones){
  resultado <- (precipitaciones - mean(precipitaciones)) / sd(precipitaciones)
  return(resultado)
}
```

## Ajustar ARIMA y realizar predicción

Se define la función `ajustar_y_predecir` que ajusta un modelo ARIMA. Necesita de la serie temporal `y_entrenamiento`, las variables independientes `xreg_entrenamiento` y la predicción de estos `xreg_predicción`, el orden del modelo ARIMA(p,q,d) `orden`, y el número de periodos a predecir `h_periodos`.

```
# Función para ajustar modelo ARIMA y hacer predicción
ajustar_y_predecir <- function(y_entrenamiento,
                              xreg_entrenamiento,
                              xreg_predicción, orden,
                              h_periodos){

  modelo <- Arima(y_entrenamiento,
                 order = orden,
                 xreg = xreg_entrenamiento)

  resultado_predicción <- forecast(modelo,
                                   xreg = xreg_predicción,
                                   h = h_periodos)

  return(resultado_predicción)
}
```

## Datos

Se importan los datos mediante una base de datos de SPSS (.sav)

```
#Ingresar datos,
library(haven)
SPSS_Registros_Anuales_5_ <- read_sav("C:/Users/angel/Downloads/SPSS Registros Anuales Lago Caburgua.sav")

data <- SPSS_Registros_Anuales_5_

# Datos hasta 2022
dique_1984_2022 <- data$DIQUE[1:39]
superficie_1984_2022 <- data$Superficie_Aproximada30deDiciembre[1:39]
max_temperatura_1984_2022 <- data$TemperaturaMáximapromedioanual[1:39]
precipitaciones_1984_2022 <- data$Precipitacionesanuales[1:39]
```

Para simplificar las propiedades al usar modelos ARIMA es necesario comprobar estacionariedad en cada una de las series de tiempo.

Comprobamos la existencia de estacionariedad con Test de raíces unitarias.

Prueba de Dickey Fuller

$H_0$ : La serie temporal tiene una raíz unitaria (no estacionaria).  $H_1$ : La serie temporal no tiene raíz unitaria (estacionaria).

```
if (!require(tseries)){install.packages("tseries")}

## Cargando paquete requerido: tseries

library(tseries)

adf.test(superficie_1984_2022,k=0) # Test de Dickey Fuller para superficie
```

```
##
## Augmented Dickey-Fuller Test
##
## data: superficie_1984_2022
## Dickey-Fuller = -3.769, Lag order = 0, p-value = 0.03295
## alternative hypothesis: stationary
```

```
adf.test(max_temperatura_1984_2022,k=0) # Test de Dickey Fuller para temperatura maxima
```

```
##
## Augmented Dickey-Fuller Test
##
## data: max_temperatura_1984_2022
## Dickey-Fuller = -6.3832, Lag order = 0, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(precipitaciones_1984_2022, k=0) # Test de Dickey Fuller para precipitaciones
```

```
##
## Augmented Dickey-Fuller Test
##
## data: precipitaciones_1984_2022
## Dickey-Fuller = -5.9319, Lag order = 0, p-value = 0.01
## alternative hypothesis: stationary
```

Las tres series (superficie, temperatura maxima y precipitaciones) presentan suficiente evidencia estadística para rechazar la hipótesis nula (no estacionariedad). Por lo tanto, las series de superficie, temperatura maxima y precipitaciones son estacionarias. Las propiedades estadísticas (como media y varianza) de las series no cambian a lo largo del tiempo.

## Modelo con el 50% de los datos para predecir hasta el 60%

### Filtrar datos y escalar precipitaciones

```
# Datos hasta 2004
superficie_1984_2004 <- data$Superficie_Aproximada30deDiciembre[1:21]
max_temperatura_1984_2004 <- data$TemperaturaMáximapromedioanual[1:21]
precipitaciones_1984_2004 <- data$Precipitacionesanuales[1:21]
superficie_1984_2008<-data$Superficie_Aproximada30deDiciembre[1:25]

# Datos predictores para predicciones futuras (2004-2008)
max_temperatura_2005_2008 <- data$TemperaturaMáximapromedioanual[22:25]
precipitaciones_2005_2008 <- data$Precipitacionesanuales[22:25]

precipitaciones_1984_2004_escalado <- escalar_precipitaciones(precipitaciones_1984_2004)
precipitaciones_2005_2008_escalado <- escalar_precipitaciones(precipitaciones_2005_2008)
```

### Predicción

```

# Predicciones 2004-2008
predictores_1984_2004 <- cbind(
  precipitaciones_1984_2004_escalado,
  max_temperatura_1984_2004)

predictores_2005_2008 <- cbind(
  precipitaciones_2005_2008_escalado,
  max_temperatura_2005_2008)

predicciones_2005_2008 <- ajustar_y_predecir(
  superficie_1984_2004,
  predictores_1984_2004,
  predictores_2005_2008,
  orden = c(3,0,1),
  h_periodos = 4)

# Gráfico de predicciones
plot(
  predicciones_2005_2008,
  main = "Comparación de Predicciones y Datos Reales 2004-2008",
  flty = 1,
  lwd = 2
)

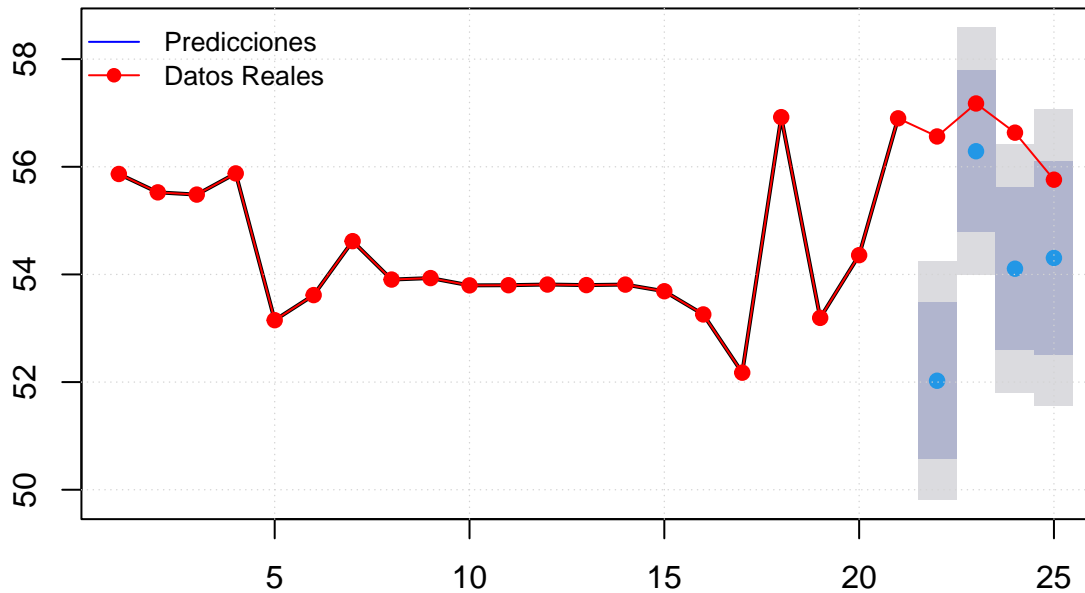
# Añadir el grid con una línea cada 1 unidad
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted", lwd = 0.7,
     equilogs = TRUE)

# Añadir los datos reales al gráfico
points(seq_along(superficie_1984_2008), superficie_1984_2008, col = "red", pch = 19)
lines(seq_along(superficie_1984_2008), superficie_1984_2008, col = "red")

# Añadir una leyenda con tamaño reducido
legend("topleft", legend = c("Predicciones", "Datos Reales"),
      col = c("blue", "red"), lty = c(1, 1), pch = c(NA, 19),
      cex = 0.87, # Tamaño del texto de la leyenda
      bty = "n") # Sin borde en la leyenda

```

## Comparación de Predicciones y Datos Reales 2004–2008



### Test de hipótesis para los errores

Se realiza el test de Ljung-Box sobre los errores de predicción del modelo ARIMA.

$H_0$ : Los residuos del modelo no tienen autocorrelación significativa; es decir, los errores son ruido blanco.

$H_1$ : Los residuos del modelo tienen autocorrelación significativa; es decir, los errores no son ruido blanco.

*#Error*

```
Box.test(predicciones_2005_2008, type = "Ljung-Box")
```

```
##  
## Box-Ljung test  
##  
## data: predicciones_2005_2008  
## X-squared = 3.5653, df = 1, p-value = 0.059
```

En este caso, el valor-p dado por la prueba de hipótesis (0.059) es mayor a nivel de significancia establecido (0.05), por lo tanto, se puede concluir que los errores del modelo no tienen autocorrelación significativa, lo que indica un buen ajuste del modelo.

En este caso, el valor p obtenido (0.059) es suficiente para concluir que los errores no tienen autocorrelación significativa, lo que indica un buen ajuste del modelo.

# Modelo con el 60% de los datos para predecir hasta el 70%

## Filtrar datos y escalar precipitaciones

```
# Datos hasta 2008
superficie_1984_2008<-data$Superficie_Aproximada30deDiciembre[1:25]
max_temperatura_1984_2008<-data$TemperaturaMáximapromedioanual[1:25]
precipitaciones_1984_2008 <- data$Precipitacionesanuales[1:25]
superficie_1984_2012<-data$Superficie_Aproximada30deDiciembre[1:29]

# Datos predictores para predicciones futuras (2008-2012)
max_temperatura_2009_2012 <- data$TemperaturaMáximapromedioanual[26:29]
precipitaciones_2009_2012 <- data$Precipitacionesanuales[26:29]

precipitaciones_1984_2008_escalado <- escalar_precipitaciones(precipitaciones_1984_2008)
precipitaciones_2009_2012_escalado <- escalar_precipitaciones(precipitaciones_2009_2012)
```

## Predicción

```
# Predicciones 2008-2012
predictores_1984_2008 <- cbind(
  precipitaciones_1984_2008_escalado,
  max_temperatura_1984_2008)

predictores_2009_2012 <- cbind(
  precipitaciones_2009_2012_escalado,
  max_temperatura_2009_2012)

predicciones_2009_2012 <- ajustar_y_predecir(
  superficie_1984_2008,
  predictores_1984_2008,
  predictores_2009_2012,
  orden = c(3,0,1),
  h_periodos = 4)

# Gráfico de predicciones
plot(
  predicciones_2009_2012,
  main = "Comparación de Predicciones y Datos Reales 2008-2012",
  flty = 1,
  lwd = 2
)

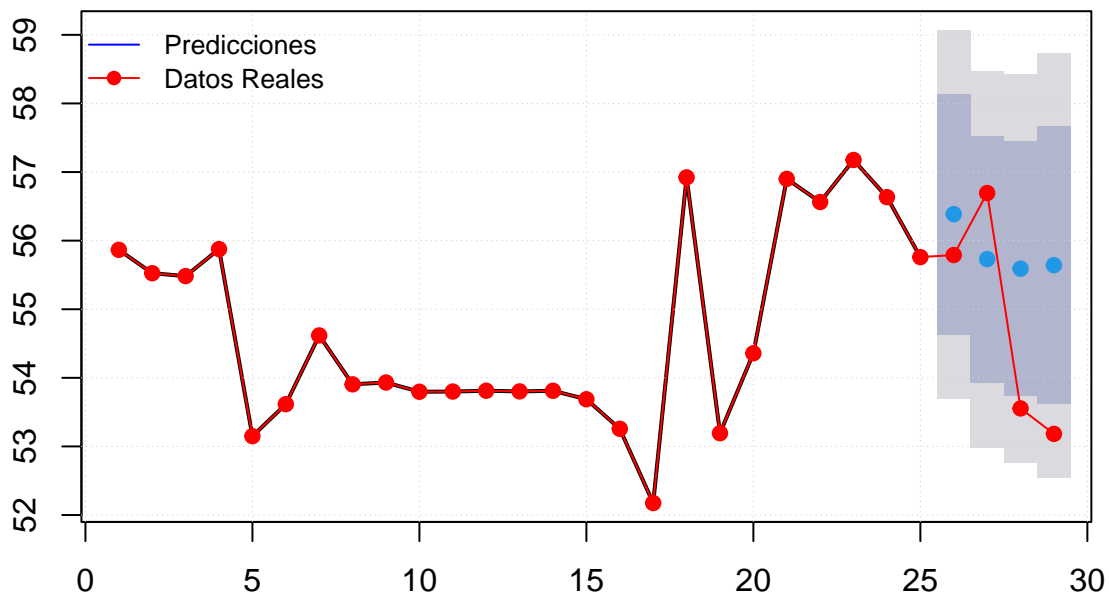
# Añadir el grid con una línea cada 1 unidad
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted", lwd = 0.6,
     equilog = TRUE)

# Añadir los datos reales al gráfico
points(seq_along(superficie_1984_2012), superficie_1984_2012, col = "red", pch = 19)
lines(seq_along(superficie_1984_2012), superficie_1984_2012, col = "red")

# Añadir una leyenda con tamaño reducido
legend("topleft", legend = c("Predicciones", "Datos Reales"),
```

```
col = c("blue", "red"), lty = c(1, 1), pch = c(NA, 19),
cex = 0.87, # Tamaño del texto de la leyenda
bty = "n") # Sin borde en la leyenda
```

## Comparación de Predicciones y Datos Reales 2008–2012



## Test de hipótesis para los errores

$H_0$ : Los residuos del modelo no tienen autocorrelación significativa; es decir, los errores son ruido blanco.

$H_1$ : Los residuos del modelo tienen autocorrelación significativa; es decir, los errores no son ruido blanco.

```
Box.test(predicciones_2009_2012, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: predicciones_2009_2012
## X-squared = 0.019594, df = 1, p-value = 0.8887
```

En este caso, el valor-p dado por la prueba de hipótesis (0.8887) es mayor a nivel de significancia establecido (0.05), por lo tanto, se puede concluir que los errores del modelo no tienen autocorrelación significativa, lo que indica un buen ajuste del modelo.



## Modelo con el 70% de los datos para predecir hasta el 80%

### Filtrar datos y escalar precipitaciones

```
# Datos hasta 2012
superficie_1984_2012<-data$Superficie_Aproximada30deDiciembre[1:29]
max_temperatura_1984_2012 <- data$TemperaturaMáximapromedioanual[1:29]
precipitaciones_1984_2012 <- data$Precipitacionesanuales[1:29]
superficie_1984_2016 <- data$Superficie_Aproximada30deDiciembre[1:33]

# Datos predictores para predicciones futuras (2012-2016)
max_temperatura_2013_2016 <- data$TemperaturaMáximapromedioanual[30:33]
precipitaciones_2013_2016 <- data$Precipitacionesanuales[30:33]

precipitaciones_1984_2012_escalado <- escalar_precipitaciones(precipitaciones_1984_2012)
precipitaciones_2013_2016_escalado <- escalar_precipitaciones(precipitaciones_2013_2016)
```

### Predicción

```
# Predicciones 2012-2016
predictores_1984_2012 <- cbind(
  precipitaciones_1984_2012_escalado,
  max_temperatura_1984_2012)

predictores_2013_2016 <- cbind(
  precipitaciones_2013_2016_escalado,
  max_temperatura_2013_2016)

predicciones_2013_2016 <- ajustar_y_predecir(
  superficie_1984_2012,
  predictores_1984_2012,
  predictores_2013_2016,
  orden = c(3,0,1),
  h_periodos = 4)

# Gráfico de predicciones
plot(
  predicciones_2013_2016,
  main = "Comparación de Predicciones y Datos Reales 2012-2016",
  flty = 1,
  lwd = 2
)

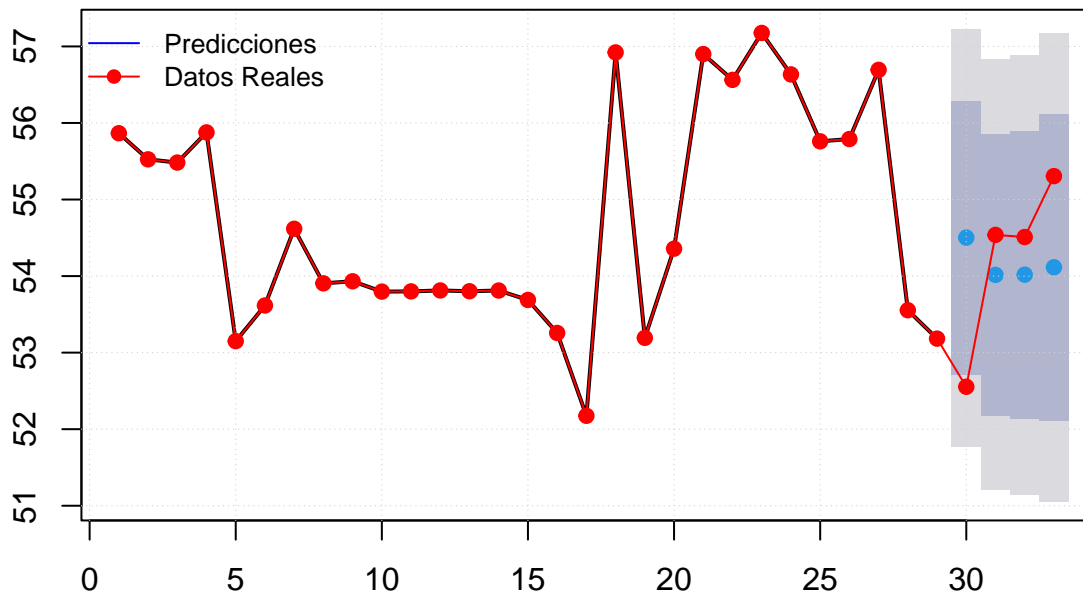
# Añadir el grid con una línea cada 1 unidad
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted", lwd = 0.6,
     equilog = TRUE)

# Añadir los datos reales al gráfico
points(seq_along(superficie_1984_2016), superficie_1984_2016, col = "red", pch = 19)
lines(seq_along(superficie_1984_2016), superficie_1984_2016, col = "red")

# Añadir una leyenda con tamaño reducido
legend("topleft", legend = c("Predicciones", "Datos Reales"),
      col = c("blue", "red"), lty = c(1, 1), pch = c(NA, 19),
```

```
cex = 0.87, # Tamaño del texto de la leyenda
bty = "n") # Sin borde en la leyenda
```

## Comparación de Predicciones y Datos Reales 2012–2016



## Test de hipótesis para los errores

```
Box.test(predicciones_2013_2016, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: predicciones_2013_2016
## X-squared = 0.24869, df = 1, p-value = 0.618
```

En este caso, el valor-p dado por la prueba de hipótesis (0.618) es mayor a nivel de significancia establecido (0.05), por lo tanto, se puede concluir que los errores del modelo no tienen autocorrelación significativa, lo que indica un buen ajuste del modelo.

## Modelo con el 80% de los datos para predecir hasta el 90%

### Filtrar datos y escalar precipitaciones

```

# Datos hasta 2016
superficie_1984_2016 <- data$Superficie_Aproximada30deDiciembre[1:33]
max_temperatura_1984_2016 <- data$TemperaturaMáximapromedioanual[1:33]
precipitaciones_1984_2016 <- data$Precipitacionesanuales[1:33]

# Datos predictores para predicciones futuras (2016-2020)
superficie_1984_2020 <- data$Superficie_Aproximada30deDiciembre[1:37]
max_temperatura_2017_2020 <- data$TemperaturaMáximapromedioanual[34:37]
precipitaciones_2017_2020 <- data$Precipitacionesanuales[34:37]

precipitaciones_1984_2016_escalado <- escalar_precipitaciones(precipitaciones_1984_2016)
precipitaciones_2017_2020_escalado <- escalar_precipitaciones(precipitaciones_2017_2020)

```

## Predicción

```

# Predicciones 2016-2020
predictores_1984_2016 <- cbind(
  precipitaciones_1984_2016_escalado,
  max_temperatura_1984_2016)

predictores_2017_2020 <- cbind(
  precipitaciones_2017_2020_escalado,
  max_temperatura_2017_2020)

predicciones_2017_2020 <- ajustar_y_predecir(
  superficie_1984_2016,
  predictores_1984_2016,
  predictores_2017_2020,
  orden = c(3,0,1),
  h_periodos = 4)

# Gráfico de predicciones
plot(
  predicciones_2017_2020,
  main = "Comparación de Predicciones y Datos Reales 2016-2020",
  flty = 1,
  lwd = 2
)

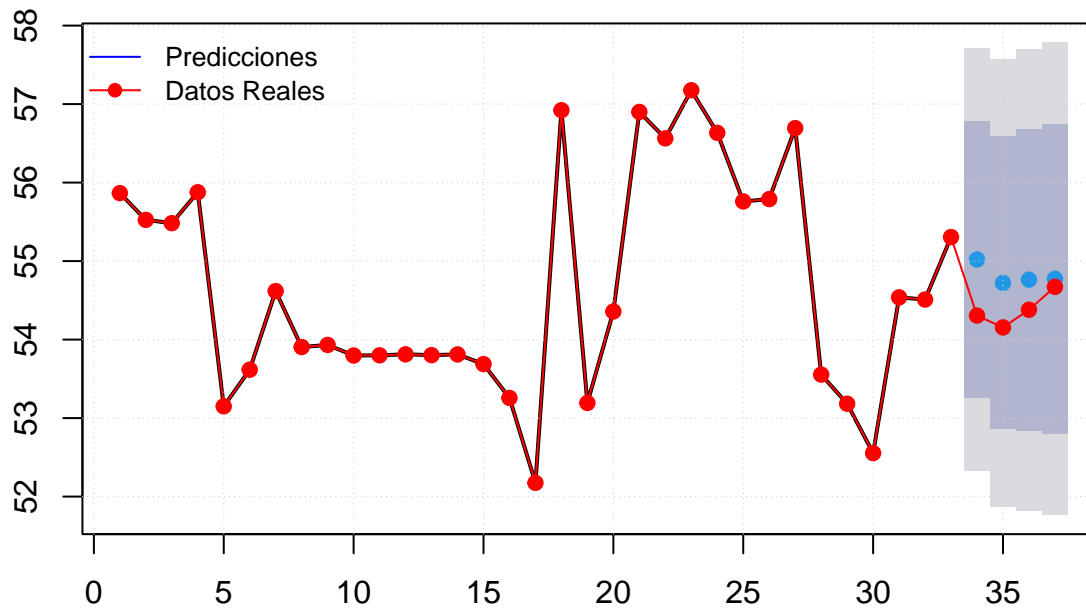
# Añadir el grid con una línea cada 1 unidad
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted", lwd = 0.6,
     equilogs = TRUE)

# Añadir los datos reales al gráfico
points(seq_along(superficie_1984_2020), superficie_1984_2020, col = "red", pch = 19)
lines(seq_along(superficie_1984_2020), superficie_1984_2020, col = "red")

# Añadir una leyenda con tamaño reducido
legend("topleft", legend = c("Predicciones", "Datos Reales"),
      col = c("blue", "red"), lty = c(1, 1), pch = c(NA, 19),
      cex = 0.87, # Tamaño del texto de la leyenda
      bty = "n") # Sin borde en la leyenda

```

## Comparación de Predicciones y Datos Reales 2016–2020



### Test de hipótesis para los errores

$H_0$ : Los residuos del modelo no tienen autocorrelación significativa; es decir, los errores son ruido blanco.

$H_1$ : Los residuos del modelo tienen autocorrelación significativa; es decir, los errores no son ruido blanco.

```
Box.test(predicciones_2017_2020, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: predicciones_2017_2020
## X-squared = 0.58746, df = 1, p-value = 0.4434
```

En este caso, el valor-p dado por la prueba de hipótesis (0.4434) es mayor a nivel de significancia establecido (0.05), por lo tanto, se puede concluir que los errores del modelo no tienen autocorrelación significativa, lo que indica un buen ajuste del modelo.

## Modelo con el 70% de los datos para predecir el 100%

### Filtrar datos y escalar precipitaciones

```

#Datos para grafico 70/30
dique_1984_2012<- data$DIQUE[1:29]
dique_2013_2022<- data$DIQUE[30:39]
precipitaciones_2013_2022<-data$Precipitacionesanuales[30:39]
max_temperatura_2013_2022<-data$TemperaturaMáximapromedioanual[30:39]
precipitaciones_1984_2012<-data$Precipitacionesanuales[1:29]

precipitaciones_1984_2012_escalado <- escalar_precipitaciones(precipitaciones_1984_2012)
precipitaciones_2013_2022_escalado<-escalar_precipitaciones(precipitaciones_2013_2022)

```

## Predicción

```

#Modelo 70/30

predictores_1984_2012 <- cbind(
  precipitaciones_1984_2012_escalado,
  max_temperatura_1984_2012,
  dique_1984_2012)

predictores_2013_2022 <- cbind(
  precipitaciones_2013_2022_escalado,
  max_temperatura_2013_2022,
  dique_2013_2022)

predicciones_2013_2022 <- ajustar_y_predecir(
  superficie_1984_2012,
  predictores_1984_2012,
  predictores_2013_2022,
  orden = c(3,0,1),
  h_periodos = 10)

#valores reales para los periodos 2013 2022
valores_reales <- c(superficie_1984_2022[30:39])
# Calcular el Error Cuadrático Medio (ECM)
ecm_training <- accuracy(predicciones_2013_2022, valores_reales)[1,2] # Extraer el ECM de la tabla de precisión
ecm_test <- accuracy(predicciones_2013_2022, valores_reales)[2,2]

# Gráfico de predicciones
plot(
  predicciones_2013_2022,
  main = "Comparación de Predicciones y Datos Reales \n 2012-2022 (70%-30%)",
  flty = 1,
  lwd = 2,
  xaxt = "n"
)

# Personalizar el eje x con los años
axis(1, at = seq(1, length(superficie_1984_2022), by = 1),
     labels = seq(1984, 2022, by = 1), las = 2, cex.axis = 0.8)

# Añadir el grid con una línea cada 1 unidad en ambos ejes

```

```

grid(nx = length(superficie_1984_2022)+8, ny = NULL, col = "lightgray",
     lty = "dotted", lwd = 0.4, equilogs = TRUE)

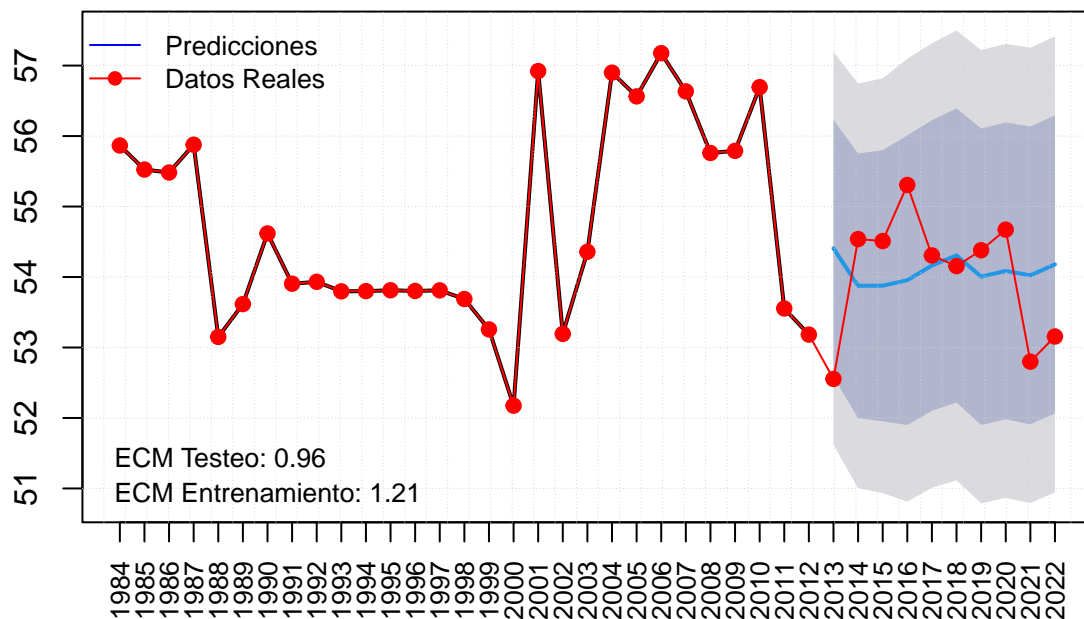
# Añadir los datos reales al gráfico
points(seq_along(superficie_1984_2022), superficie_1984_2022, col = "red", pch = 19)
lines(seq_along(superficie_1984_2022), superficie_1984_2022, col = "red")

# Añadir una leyenda con tamaño reducido
legend("topleft", legend = c("Predicciones", "Datos Reales"),
      col = c("blue", "red"), lty = c(1, 1), pch = c(NA, 19),
      cex = 0.87, # Tamaño del texto de la leyenda
      bty = "n") # Sin borde en la leyenda
# Añadir el texto para ECM en el gráfico
text(x = 0, y = 50.9,
     labels = paste("ECM Entrenamiento:", round(ecm_training, 2)),
     pos = 4, col = "black", cex = 0.8)

text(x = 0, y = 51.4,
     labels = paste("ECM Testeo:", round(ecm_test, 2)),
     pos = 4, col = "black", cex = 0.8)

```

## Comparación de Predicciones y Datos Reales 2012–2022 (70%–30%)



## Test de hipótesis para los errores

$H_0$ : Los residuos del modelo no tienen autocorrelación significativa; es decir, los errores son ruido blanco.

$H_1$ : Los residuos del modelo tienen autocorrelación significativa; es decir, los errores no son ruido blanco.

```
Box.test(predicciones_2013_2022, type = "Ljung-Box")
```

```
##  
## Box-Ljung test  
##  
## data: predicciones_2013_2022  
## X-squared = 0.026252, df = 1, p-value = 0.8713
```

En este caso, el valor-p dado por la prueba de hipótesis (0.8713) es mayor a nivel de significancia establecido (0.05), por lo tanto, se puede concluir que los errores del modelo no tienen autocorrelación significativa, lo que indica un buen ajuste del modelo.

## Modelo con el 80% de los datos para predecir el 100%

### Filtrar datos y escalar precipitaciones

```
#Datos para 80/20  
  
dique_1984_2016<- data$DIQUE[1:33]  
dique_2017_2022<- data$DIQUE[34:39]  
precipitaciones_2017_2022<- data$Precipitacionesanuales[34:39]  
max_temperatura_2017_2022<-data$TemperaturaMáximapromedioanual[34:39]  
  
precipitaciones_1984_2016_escalado <- escalar_precipitaciones(precipitaciones_1984_2016)  
precipitaciones_2017_2022_escalado<-escalar_precipitaciones(precipitaciones_2017_2022)
```

### Predicción

```
#Modelo 80-20  
  
predictores_1984_2016 <- cbind(  
  precipitaciones_1984_2016_escalado,  
  max_temperatura_1984_2016,dique_1984_2016)  
  
predictores_2017_2022 <- cbind(  
  precipitaciones_2017_2022_escalado,  
  max_temperatura_2017_2022,  
  dique_2017_2022)  
  
predicciones_2017_2022 <- ajustar_y_predecir(  
  superficie_1984_2016,  
  predictores_1984_2016,  
  predictores_2017_2022,  
  orden = c(3,0,1),  
  h_periodos = 6)  
  
#valores reales para los periodos 2017 2022  
valores_reales <- c(superficie_1984_2022[34:39])  
# Calcular el Error Cuadrático Medio (ECM)
```

```

ecm_training <- accuracy(predicciones_2017_2022, valores_reales)[1,2] # Extraer el ECM de la tabla de precisión
ecm_test <- accuracy(predicciones_2017_2022, valores_reales)[2,2]

# Gráfico de predicciones
plot(
  predicciones_2017_2022,
  main = "Comparación de Predicciones y Datos Reales \n 2016-2022 (80%-20%)",
  flty = 1,
  lwd = 2,
  xaxt = "n"
)

# Personalizar el eje x con los años
axis(1, at = seq(1, length(superficie_1984_2022), by = 1),
     labels = seq(1984, 2022, by = 1), las = 2, cex.axis = 0.8)

# Añadir el grid con una línea cada 1 unidad en ambos ejes
grid(nx = length(superficie_1984_2022), ny = NULL, col = "lightgray",
     lty = "dotted", lwd = 0.4, equilogs = TRUE)

# Añadir los datos reales al gráfico
points(seq_along(superficie_1984_2022), superficie_1984_2022, col = "red", pch = 19)
lines(seq_along(superficie_1984_2022), superficie_1984_2022, col = "red")

legend("topleft", legend = c("Predicciones", "Datos Reales"),
      col = c("blue", "red"), lty = c(1, 1), pch = c(NA, 19),
      cex = 0.87, # Tamaño del texto de la leyenda
      bty = "n") # Sin borde en la leyenda

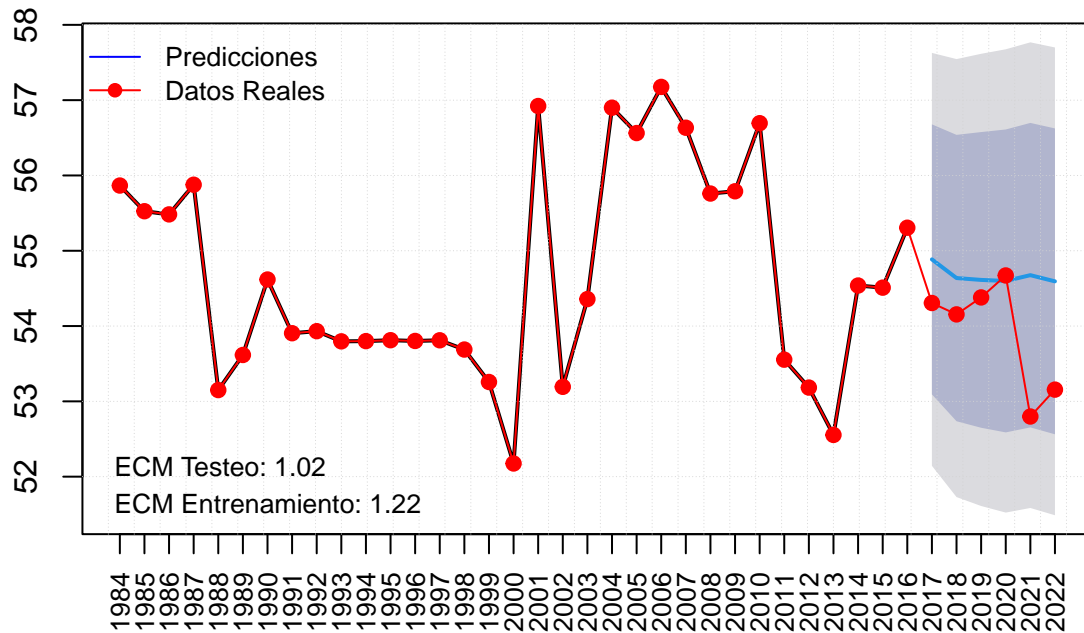
# Añadir el texto para ECM en el gráfico
text(x = 0, y = 51.6,
     labels = paste("ECM Entrenamiento:", round(ecm_training, 2)),
     pos = 4, col = "black", cex = 0.8)

text(x = 0, y = 52.1,
     labels = paste("ECM Testeo:", round(ecm_test, 2)),
     pos = 4, col = "black", cex = 0.8)

```



## Comparación de Predicciones y Datos Reales 2016–2022 (80%–20%)



### Test de hipótesis para los errores

$H_0$ : Los residuos del modelo no tienen autocorrelación significativa; es decir, los errores son ruido blanco.

$H_1$ : Los residuos del modelo tienen autocorrelación significativa; es decir, los errores no son ruido blanco.

```
Box.test(predicciones_2017_2022, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: predicciones_2017_2022
## X-squared = 0.019593, df = 1, p-value = 0.8887
```

En este caso, el valor-p dado por la prueba de hipótesis (0.8887) es mayor a nivel de significancia establecido (0.05), por lo tanto, se puede concluir que los errores del modelo no tienen autocorrelación significativa, lo que indica un buen ajuste del modelo.

## Modelo predictivo con la reconstrucción del dique en 2026

### Filtrar datos y escalar precipitaciones

```

# Datos predictores para predicciones futuras (2023-2030)
max_temperatura_2023_2030 <- data$Pronosticado_TemperaturaMáximapromedioanual_Modelo_1[40:47]

precipitaciones_2023_2030 <- data$Pronosticado_Precipitacionesanuales_Modelo_1[40:47]

dique_2023_2030_CD <- data$DIQUE[40:47]

precipitaciones_1984_2022_escalado<-escalar_precipitaciones(precipitaciones_1984_2022)
precipitaciones_2023_2030_escalado <- escalar_precipitaciones(precipitaciones_2023_2030)

```

## Predicción

```

# Predicciones 2023-2030
predictores_1984_2022_CD <- cbind(
  precipitaciones_1984_2022_escalado,
  max_temperatura_1984_2022,
  dique_1984_2022 )

predictores_2023_2030_CD<- cbind(
  precipitaciones_2023_2030_escalado,
  max_temperatura_2023_2030,
  dique_2023_2030_CD)

predicciones_2023_2030_CD <- ajustar_y_predecir(
  superficie_1984_2022,
  predictores_1984_2022_CD,
  predictores_2023_2030_CD,
  orden = c(3,0,1),
  h_periodos = 8)

summary(predicciones_2023_2030_CD)

```

```

##
## Forecast method: Regression with ARIMA(3,0,1) errors
##
## Model Information:
## Series: y_entrenamiento
## Regression with ARIMA(3,0,1) errors
##
## Coefficients:
##          ar1      ar2      ar3      ma1  intercept
##          0.0481  0.2457  0.1554  0.2920   56.4116
## s.e.      0.8952  0.3491  0.2378  0.8716    6.2014
##          precipitaciones_1984_2022_escalado  max_temperatura_1984_2022
##                                          -0.0427                -0.0694
## s.e.                                          0.2647                0.2626
##          dique_1984_2022
##          -0.5529
## s.e.          0.7935
##
## sigma^2 = 1.706: log likelihood = -61.43
## AIC=140.87   AICc=147.07   BIC=155.84
##
## Error measures:

```

```
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.02707878 1.164514 0.9293467 -0.09561055 1.701676 1.043408
##           ACF1
## Training set 0.00557976
##
## Forecasts:
##   Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## 40      54.25854 52.58463 55.93245 51.69851 56.81856
## 41      54.19739 52.42935 55.96542 51.49341 56.90136
## 42      54.33504 52.51343 56.15666 51.54913 57.12096
## 43      53.89151 52.02188 55.76114 51.03216 56.75086
## 44      53.90130 52.01919 55.78341 51.02286 56.77974
## 45      53.93447 52.04358 55.82535 51.04261 56.82633
## 46      54.00630 52.11113 55.90147 51.10789 56.90471
## 47      54.01405 52.11700 55.91110 51.11277 56.91534
```

```
# Extraer los valores predichos
```

```
prediccion_temperatura_2030_CD <- predicciones_2023_2030_CD$mean
```

```
# Gráfico de predicciones
```

```
plot(
  predicciones_2023_2030_CD,
  main = "Predicción Superficie hasta 2030 Con Dique",
  flty = 1,
  lwd = 2,
  xaxt = "n"
)
```

```
# Personalizar el eje x con los años
```

```
axis(1, at = seq(1, length(superficie_1984_2022)+8, by = 1),
     labels = seq(1984, 2030, by = 1), las = 2, cex.axis = 0.8)
```

```
# Añadir el grid con una línea cada 1 unidad en ambos ejes
```

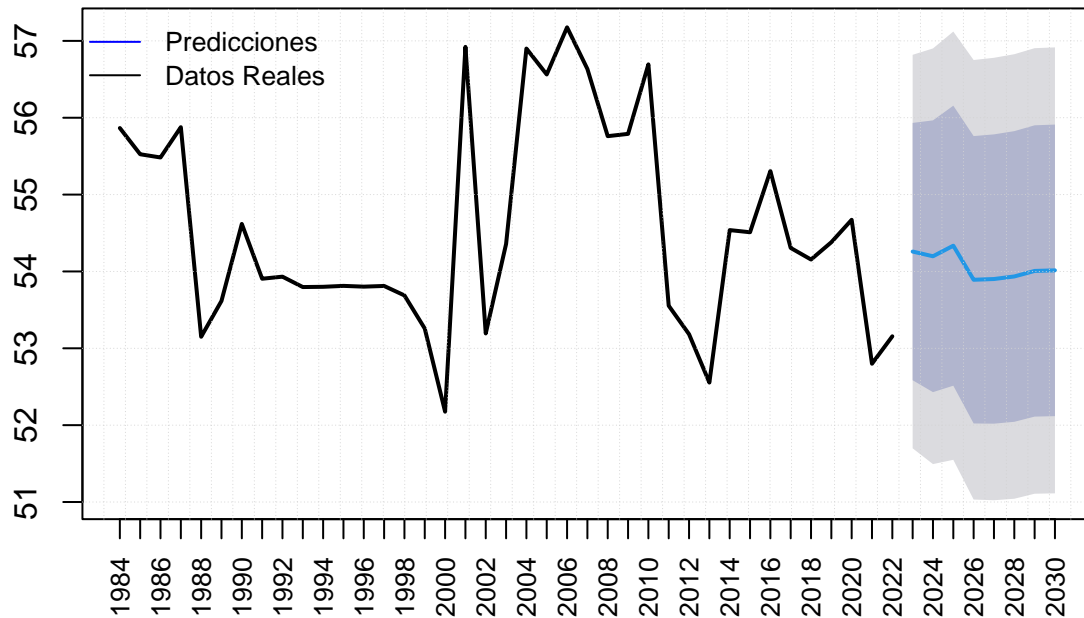
```
grid(nx = length(superficie_1984_2022)+8, ny = NULL, col = "lightgray",
     lty = "dotted", lwd = 0.4, equilog = TRUE)
```

```
lines(seq_along(superficie_1984_2022), superficie_1984_2022, col = "black")
```

```
# Añadir una leyenda con tamaño reducido
```

```
legend("topleft", legend = c("Predicciones", "Datos Reales"),
      col = c("blue", "black"), lty = c(1, 1), pch = c(NA, NA),
      cex = 0.87, # Tamaño del texto de la leyenda
      bty = "n") # Sin borde en la leyenda
```

## Predicción Superficie hasta 2030 Con Dique



```
# Imprimir los valores de la predicción
print(prediccion_temperatura_2030_CD)
```

```
## Time Series:
## Start = 40
## End = 47
## Frequency = 1
## [1] 54.25854 54.19739 54.33504 53.89151 53.90130 53.93447 54.00630 54.01405
```

## Test de hipótesis para los errores

$H_0$ : Los residuos del modelo no tienen autocorrelación significativa; es decir, los errores son ruido blanco.

$H_1$ : Los residuos del modelo tienen autocorrelación significativa; es decir, los errores no son ruido blanco.

```
Box.test(predicciones_2023_2030_CD, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: predicciones_2023_2030_CD
## X-squared = 1.7468, df = 1, p-value = 0.1863
```

En este caso, el valor-p dado por la prueba de hipótesis (0.1863) es mayor a nivel de significancia establecido (0.05), por lo tanto, se puede concluir que los errores del modelo no tienen autocorrelación significativa, lo que indica un buen ajuste del modelo.

# Modelo predictivo sin una reinstalación del dique

## Filtrar datos y escalar precipitaciones

```
# Datos predictores para predicciones futuras (2023-2030)
max_temperatura_2023_2030 <- data$Pronosticado_TemperaturaMáximapromedioanual_Modelo_1[40:47]

precipitaciones_2023_2030 <- data$Pronosticado_Precipitacionesanuales_Modelo_1[40:47]

dique_2023_2030_SD<-c(0,0,0,0,0,0,0,0)

precipitaciones_1984_2022_escalado <- escalar_precipitaciones(precipitaciones_1984_2022)
precipitaciones_2023_2030_escalado <- escalar_precipitaciones(precipitaciones_2023_2030)
```

## Predicción

```
# Predicciones 2023-2030
predictores_1984_2022_SD <- cbind(
  precipitaciones_1984_2022_escalado,
  max_temperatura_1984_2022,
  dique_1984_2022)

predictores_2023_2030_SD<- cbind(
  precipitaciones_2023_2030_escalado,
  max_temperatura_2023_2030,
  dique_2023_2030_SD)

predicciones_2023_2030_SD <- ajustar_y_predecir(
  superficie_1984_2022,
  predictores_1984_2022_SD,
  predictores_2023_2030_SD,
  orden = c(3,0,1),
  h_periodos = 8)

summary(predicciones_2023_2030_SD)
```

```
##
## Forecast method: Regression with ARIMA(3,0,1) errors
##
## Model Information:
## Series: y_entrenamiento
## Regression with ARIMA(3,0,1) errors
##
## Coefficients:
##          ar1      ar2      ar3      ma1  intercept
##          0.0481  0.2457  0.1554  0.2920   56.4116
## s.e.      0.8952  0.3491  0.2378  0.8716    6.2014
##          precipitaciones_1984_2022_escalado max_temperatura_1984_2022
##                                           -0.0427      -0.0694
## s.e.                                           0.2647      0.2626
##          dique_1984_2022
##          -0.5529
```

```
## s.e.          0.7935
##
## sigma^2 = 1.706: log likelihood = -61.43
## AIC=140.87   AICc=147.07   BIC=155.84
##
## Error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.02707878 1.164514 0.9293467 -0.09561055 1.701676 1.043408
##              ACF1
## Training set 0.00557976
##
## Forecasts:
##   Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## 40      54.25854 52.58463 55.93245 51.69851 56.81856
## 41      54.19739 52.42935 55.96542 51.49341 56.90136
## 42      54.33504 52.51343 56.15666 51.54913 57.12096
## 43      54.44439 52.57476 56.31402 51.58504 57.30374
## 44      54.45418 52.57207 56.33629 51.57574 57.33262
## 45      54.48735 52.59646 56.37823 51.59549 57.37921
## 46      54.55918 52.66402 56.45435 51.66077 57.45759
## 47      54.56693 52.66988 56.46398 51.66565 57.46822
```

```
# Extraer los valores predichos
```

```
prediccion_temperatura_2030_SD <- predicciones_2023_2030_SD$mean
```

```
# Gráfico de predicciones
```

```
plot(
  predicciones_2023_2030_SD,
  main = "Predicción Superficie hasta 2030 Sin Dique",
  flty = 1,
  lwd = 2,
  xaxt = "n"
)
```

```
# Personalizar el eje x con los años
```

```
axis(1, at = seq(1, length(superficie_1984_2022)+8, by = 1),
     labels = seq(1984, 2030, by = 1), las = 2, cex.axis = 0.8)
```

```
# Añadir el grid con una línea cada 1 unidad en ambos ejes
```

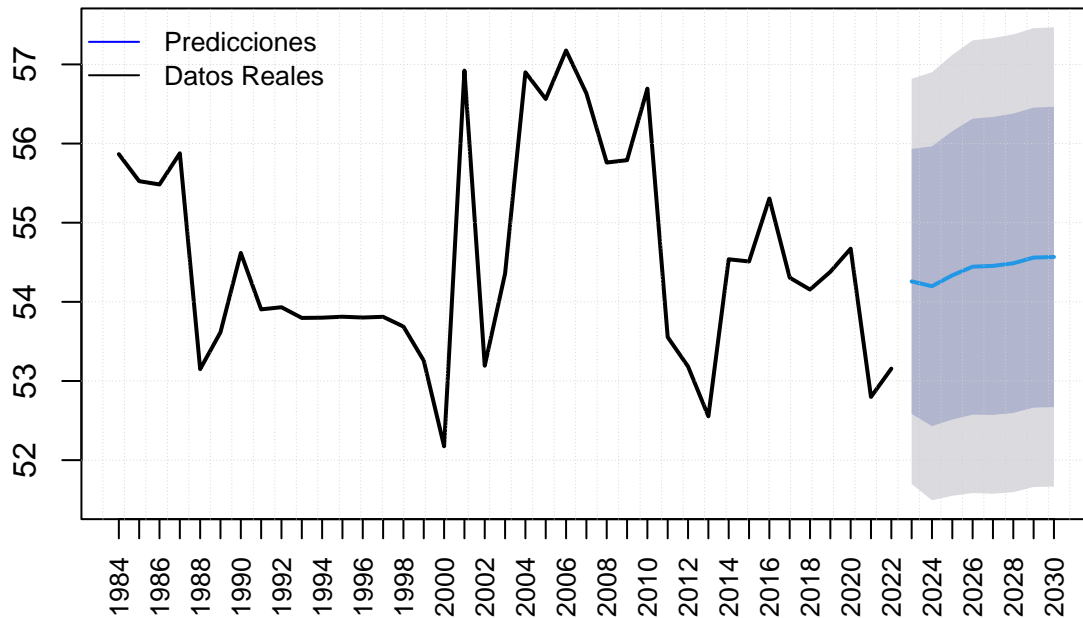
```
grid(nx = length(superficie_1984_2022)+8, ny = NULL, col = "lightgray",
     lty = "dotted", lwd = 0.4, equilogs = TRUE)
```

```
lines(seq_along(superficie_1984_2022), superficie_1984_2022, col = "black")
```

```
# Añadir una leyenda con tamaño reducido
```

```
legend("topleft", legend = c("Predicciones", "Datos Reales"),
      col = c("blue", "black"), lty = c(1, 1), pch = c(NA, NA),
      cex = 0.87, # Tamaño del texto de la leyenda
      bty = "n") # Sin borde en la leyenda
```

## Predicción Superficie hasta 2030 Sin Dique



```
# Imprimir los valores de la predicción
print(prediccion_temperatura_2030_SD)
```

```
## Time Series:
## Start = 40
## End = 47
## Frequency = 1
## [1] 54.25854 54.19739 54.33504 54.44439 54.45418 54.48735 54.55918 54.56693
```

## Test de hipotesis para los errores

$H_0$ : Los residuos del modelo no tienen autocorrelación significativa; es decir, los errores son ruido blanco.

$H_1$ : Los residuos del modelo tienen autocorrelación significativa; es decir, los errores no son ruido blanco.

```
Box.test(predicciones_2023_2030_SD, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: predicciones_2023_2030_SD
## X-squared = 5.7795, df = 1, p-value = 0.01621
```

En este caso, el valor-p dado por la prueba de hipotesis (0.01621) no es mayor a nivel de significancia establecido (0.05), por lo tanto, se puede concluir que los errores del modelo tienen autocorrelación significativa, lo que puede indicar que el modelo ARIMA no está capturando la estructura temporal de los datos, quedando dependencia en los residuos. Sin embargo, con base en las pruebas realizadas en el modelo con el 80% para predecir el 100% de los datos, podemos ver un buen ajuste (dentro de los intervalos de confianza establecidos para el 95% de confianza).