



MÁSTER EN MACHINE LEARNING CON R SOFTWARE

MODELO PREDICTIVO DE DEMANDA DE ATENCIÓN DE URGENCIA PEDIÁTRICA POR CAUSA RESPIRATORIA, EN LOS ESTABLECIMIENTOS HOSPITALARIOS DEL SERVICIO DE SALUD CHILOÉ

AUTOR: CLAUDIO CÁRDENAS MANSILLA

DIRECTOR: IGNACIO GARCÍA

FECHA: 14/02/2023

ENTIDAD COLABORADORA:



RESUMEN

Las enfermedades respiratorias en los niños/as son la causa más frecuente de consulta médica. Una situación que en invierno tiene su peak más alto, y que hoy, en un escenario de pandemia, resulta inquietante para padres que buscan una orientación médica sobre los posibles diagnósticos que pudieran tener sus hijos.

En el presente Trabajo Fin de Máster se ha procedido a la generación de modelos de predicción de Series Temporales, a través de algoritmos de *Machine Learning*, con el objetivo de seleccionar los mejores modelos en función de **métricas de precisión**¹ (RMSE, MAE y MAPE) y **análisis gráficos del ajuste de los modelos calculados**², sobre el conjunto de *datos de test (validación)*.

Para el modelado, se han utilizado los paquetes *forecast* y *nnfor*, los cuales, generan modelos de *Machine Learning* para la predicción de Series Temporales de diversos tipos, para el caso específico de este trabajo se utilizaron modelos *Autorregresivo Integrado de Media Móvil (ARIMA)*, *Autorregresión de Redes Neuronales (NNAR)*, *Red Neuronal Multilayer Perceptron (MLP)* y *Red Neuronal Extreme Learning Machines (EML)*.

Todo el ejercicio se ha realizado en *R Software*, y para el mismo se utilizó una serie temporal que contiene información acerca de *conteo de consultas de urgencia respiratoria infantil mensuales, desde enero del año 2011 y hasta noviembre del año 2022*, extraídos del DEIS (Departamento de Estadísticas e Información en Salud).

Considerando que, durante todo el *periodo vivido bajo pandemia Covid-19 en Chiloé, la demanda por consultas de urgencia fue “prácticamente nula”*, como consecuencia del pánico generalizado en la población y que los hospitales modificaron sus procesos clínicos y asistenciales significativamente, en función de la contingencia sanitaria. Se propone declarar “Nulos” todos los registros observados durante todo el periodo antes mencionado. Para luego, obtener sus estimaciones mediante el

¹ **El error del pronóstico o precision**, es la diferencia entre el valor real y el valor pronosticado para un período específico, en: <https://www.pricing.cl/conocimiento/precision-del-pronostico/>

² **El ajuste de datos** es el proceso mediante el que se ajustan modelos a datos y se analiza la precisión del ajuste. Los ingenieros y los científicos utilizan técnicas de ajuste de datos, incluidas ecuaciones matemáticas y métodos no paramétricos, para modelar datos adquiridos, en: <https://la.mathworks.com/discovery/>.

método de *imputación de Kalman* con *modelos ARIMA de espacio de estados*, ya que estos modelos en el espacio de estados pueden entre otras muchas cosas estimar observaciones perdidas.

Finalmente, el *análisis gráfico del ajuste de las predicciones de los modelos estimados* y los resultados obtenidos mediante los *indicadores de precisión, calculados ambos sobre el conjunto de test*, sugieren que *el algoritmo que presenta el mejor rendimiento*, prediciendo la demanda mensual de consultas de urgencia respiratoria pediátrica, es el *Modelo de Red Neuronal Extreme Learning Machines (EML)*. Presentando *el mejor desempeño en MAE (99.46 consultas)* y *MAPE (13.15 %)*, y el *segundo mejor en RMSE (138.92 consultas)*, por debajo del *modelo SARIMA*, siendo este último, el segundo mejor evaluado.

Palabras claves: *Machine Learning, Series Temporales, Imputación de Datos, ARIMA y Redes Neuronales.*

AGRADECIMIENTOS

A mi amada esposa María y,
A mi adorado hijo Samuel
Por su amor y comprensión.

A mis profesores Rosana y Nacho,
por su valioso apoyo en esta linda aventura.

A Dios por guiar mi camino y
no dejar que me pierda.

ÍNDICE DE CONTENIDOS

RESUMEN.....	1
AGRADECIMIENTOS	3
ÍNDICE DE CONTENIDOS.....	4
1. INTRODUCCIÓN.....	6
2. MATERIAL & MÉTODOS.	7
3. RESULTADOS.....	9
3.1 Análisis exploratorio de serie temporal.....	9
3.1.1 Importación y visualización de serie temporal.....	9
3.1.2 Evaluación de puntos de cambio en la serie temporal.	11
3.1.3 Evaluación de quiebres estructurales en la serie temporal.....	12
3.1.4 Propuesta metodológica para tratamiento de datos Outliers durante el Periodo de Pandemia Covid-19 en Chiloé.....	15
3.1.5 Verificación de supuestos estadísticos de la serie temporal corregida.....	18
3.1.6 Verificación de supuestos estadísticos de la serie temporal corregida diferenciada en componente regular y estacional.	19
3.2 Partición de muestra de datos de la serie temporal.....	22
3.3 Modelamiento de la Serie Temporal.....	23
3.3.1 Modelo ARIMA, mediante método de modelado automático.....	23
3.3.2 Modelo ARIMA, mediante análisis de correlograma ACF y PACF.	25
3.3.3 Modelo de Autorregresión de Redes Neuronales (NNAR).....	27
3.3.4 Modelo de Red Neuronal <i>Multilayer Perceptron</i> (MLP).	29
3.3.5 Modelo de Red Neuronal <i>Extreme Learning Machines</i> (EML).	31
3.4 Comparación de indicadores y pronósticos de algoritmos utilizados.	33
3.4.1 Indicadores de precisión de pronósticos de algoritmos.	33
3.4.2 Visualizaciones de pronósticos de algoritmos.....	35
4. DISCUSIÓN.....	36
5. REFERENCIAS/BIBLIOGRAFÍA.	37
6. ANEXOS.	39
6.1 ANEXO A: CÓDIGO R, PARA LA RESOLUCIÓN DEL PROCESO DE ANÁLISIS.	39
Librerías R, utilizadas en TFM.	39
Importación y visualización de serie temporal.....	40
Evaluación de puntos de cambio en la serie temporal.	41

Evaluación de quiebres estructurales en la serie temporal.....	42
Propuesta metodológica para tratamiento de datos Outliers durante el Periodo de Pandemia Covid-19 en Chiloé.	43
Verificación de supuestos estadísticos de la serie temporal corregida.....	44
Verificación de supuestos estadísticos de la serie temporal corregida diferenciada en componente regular y estacional.....	45
Partición de muestra de datos de la serie temporal.	46
Modelo ARIMA, mediante método de modelado automático.....	47
Modelo ARIMA, mediante análisis de correlograma ACF y PACF.	48
Modelo de Autorregresión de Redes Neuronales (NNAR).....	49
Modelo de Red Neuronal Multilayer Perceptron (MLP).	50
Modelo de Red Neuronal Extreme Learning Machines (EML).	52
Indicadores de precisión de pronósticos de algoritmos.	54
Visualizaciones de pronósticos de algoritmos.....	54

1. INTRODUCCIÓN.

Las aglomeraciones en los Servicios de Urgencia (SU) son un problema de salud pública a nivel mundial, que afecta tanto a pacientes como al personal de salud. Estos problemas provocan retardos en el diagnóstico, aumento en la mortalidad, errores en la atención, retardos en el inicio del tratamiento de enfermedades críticas, disminución en la capacidad de camas y disminución de la seguridad del paciente, entre otros (Contreras, S. 2018).

Según Cárdenas et., al (2014), el aumento desmesurado de la demanda en los servicios de urgencias tiene consecuencias negativas para el centro en forma de falta de espacio físico, sobre carga de los servicios de laboratorio y radiología y de los profesionales del hospital, alargamiento de la lista de espera de ingresos programados y pérdidas en productividad por las dificultades para programar el trabajo y sobre todo, se pueden producir demoras en la asistencia a pacientes que sufren situaciones de riesgo vital.

Por otra parte, las enfermedades respiratorias en los niños/as son la causa más frecuente de consulta médica. Una situación que en invierno tiene su peak más alto, y que hoy, en un escenario de pandemia, resulta inquietante para padres que buscan una orientación médica sobre los posibles diagnósticos que pudieran tener sus hijos/as.

Por todo lo argumentado anteriormente, un modelo predictivo de la demanda permitiría una mejor planificación y distribución de los recursos humanos y físicos (camas, boxes de atención, sala de procedimientos, equipamiento general, etc.) y de esta manera mejoraría el flujo de pacientes y del servicio en general. El uso de los datos históricos de consultas de urgencia disponibles para todo el país como datos abiertos por el Departamento de Estadísticas e Información en Salud (DEIS), del ministerio de Salud de la República de Chile, cada vez más accesibles para la aplicación procesos de ETL³, podría ser utilizado como herramienta para anticipar la demanda de consultas urgencia pediátricas de tipo respiratorio (y cualquier otro tipo) de los pacientes evitando aglomeraciones y sus problemas asociados, lo cual, permitiría sacar provecho de esta herramienta, estableciendo una gestión proactiva.

³ Es un tipo de integración de datos que hace referencia a los tres pasos (extraer, transformar, cargar) que se utilizan para mezclar datos de múltiples fuentes.

2. MATERIAL & MÉTODOS.

En el presente trabajo, se analizó un problema de Series Temporales para cuya resolución se procedió a aplicar los conocimientos en Machine Learning adquiridos durante el Máster, en cuanto a la importación, exploración y transformación de características o variables.

Sin embargo, a la hora de seleccionar el tipo de algoritmo, y con la finalidad de aprovechar la oportunidad para adquirir nuevos conocimientos, a parte de los Modelos Autorregresivos Integrados de Medias Móviles (ARIMA) y su versión estacional (SARIMA), se optó también, por aprender a utilizar el paquete *nnfor* (versión de desarrollo <https://github.com/trnnick/nnfor>) para R, lo cual facilitó la previsión de series temporales con los algoritmos de *Redes Neuronales Perceptron Multicapa* (MLP) y *Extreme Learning Machines* (ELM) .

En este sentido, la función *mlp* generó automáticamente conjuntos de *Redes Neuronales Perceptron Multicapa* (MLP), cuyo entrenamiento comienza con diferentes pesos iniciales aleatorios. Además, proporciono las entradas que se incluyeron en la red. Se observó que el desempeño de los diferentes operadores de combinación (Media, Mediana y Moda), resulta más eficiente cuando se utilizó la Mediana como indicador en desmedro de la Media. Por otra parte, para entrenar la *Red Neuronal Extreme Learning Machines* (EML) se usó la función *elm*, donde se observó que muchas de las entradas fueron idénticas a las observadas para función *mlp*. Con la diferencia, que, de forma predeterminada, las redes ELM comienzan con una capa oculta muy grande (100 nodos) que se recortó según fue necesario (automáticamente), para el mejor ajuste del modelo.

A su vez, mediante la librería *forecats*, se estimaron el Modelos SARIMA, primeramente, mediante el método de modelado automático *auto.arima*, y posteriormente se generó otro modelo SARIMA, mediante método analítico tradicional. Finalmente se estimó un quinto modelo utilizando La función *nnetar* que ajusta el modelo de *Autorregresión de Red Neuronal* o modelo NNAR(p,P,k), el cual utilizo los valores de retardos de las series de tiempo como entradas a la red neuronal.

Según esto, se estimaron cinco de algoritmos de *Machine Learning*, que únicamente a través del histórico de consultas mensuales, de urgencia respiratoria pediátrica, pudieron predecir la demanda

mensual de las mismas, en un horizonte de 11 meses, identificando cual de todos ellos alcanzó la mayor precisión.

El presente ejercicio, se realizó sobre una serie temporal de consultas de urgencia respiratoria pediátrica, agregadas en retardos mensuales, para la sumatoria de los cinco hospitales de la red asistencial del Servicio de Salud Chiloé. Esta serie comienza el enero del año 2011 y terminó en noviembre del año 2022, con una extensión total de 143 retardos. Estos datos fueron provistos por la Unidad de Estadística, del Subdepartamento de Inteligencia Sanitaria, perteneciente a la Dirección de Servicio de Salud Chiloé, quienes extrajeron esta información desde un link de descarga (<https://deis.minsal.cl/#datosabiertos>) en la página del Departamento de Estadísticas e Información de Salud (DEIS).

En primer lugar, se procedió con la importación de los datos y la transformación de las variables realizando posteriormente un análisis exploratorio de los datos. Luego se generaron los diferentes modelos, se compararon entre sí mediante la elaboración una tabla resumen con las principales métricas de cada uno, estimadas a partir de los datos de prueba o test, para a continuación, generar un gráfico resumen (con cuatro gráficos) organizado en cuadrícula, con los ajustes entre los datos pronosticados por el modelo vs. los datos observados en la muestra de prueba o test, de los cuatro modelos que presentaron las mejores métricas de rendimiento, en paso anterior.

Finalmente, y obtenida toda la evidencia anterior se concluyó con la identificación del modelo que puede predecir de mejor forma la demanda mensual de consultas de urgencia respiratoria pediátrica, a través de las métricas de precisión **RMSE⁴**, **MAE⁵** y **MAPE⁶**, y el mejor **ajuste gráfico en la visualización de pronóstico de los modelos**, todo esto, sobre la muestra destinada para prueba o test de los modelos estimados mediante la muestra o conjunto de entrenamiento (*train*).

⁴ Error de raíz cuadrada media (RMSE) es la desviación estándar de los valores residuales.

⁵ Error medio absoluto. Calcula la función de error absoluto medio para el pronóstico y los resultados posibles.

⁶ Error Porcentual Absoluto Medio. Mide el tamaño del error (absoluto) en términos porcentuales.

3. RESULTADOS.

3.1 Análisis exploratorio de serie temporal.

3.1.1 Importación y visualización de serie temporal.

Primeramente, se procedió a la lectura de los datos de la serie temporal consultas de urgencia respiratoria infantil observada en la red hospitalaria del Servicio de Salud Chiloé, los cuales fueron agregados de la base de datos nacional de atenciones de urgencia del Departamento de Estadísticas e Información de Salud (DEIS), del Ministerio de Salud de la República de Chile. El siguiente paso fue convertir los datos en un objeto de serie temporal en R, de modo que se lean fácilmente en las funciones de R que analizan datos de series temporales. para ello se usó la función `ts` en R. Una vez que se construyó la serie de tiempo en R, el siguiente paso fue graficar la serie de temporal con la función `autoplot` en R, según el siguiente código:

```
## Transformación de la variable "Urgencia_Pediatrica" en un objeto de serie
temporal ##
TS <- ts(BBDD_URGE_PED$Urgencia_Pediatrica,
        frequency=12,
        start=c(2011,1))

## Graficación de serie temporal ##
GF1 <- autoplot(TS, main= " Serie 2011-2022, Urgencia Respiratoria Infantil,
Hospitales SS Chiloé",
               xlab = "MESES/AÑOS",
               ylab = "N° CONSULTAS")

fit<- decompose(TS, type="additive") # Descomposicion de serie temporal ##

## Visualización de descomposición de serie temporal en estacionalidad y tend
encia ##
GF2 <- autoplot(TS, series="Serie Original", xlab = "MES/AÑO", ylab="N°CONSUL
TAS") +
  ggtitle("Descomposición Aditiva Clásica Urgencia Respiratoria Infantil
")+
  autolayer(trendcycle(fit), series="Tendencia") +
  autolayer(seasadj(fit), series="Ajuste estacional") +
```

```
scale_colour_manual(values=c("gray", "blue", "red"),
breaks=c("Serie Original", "Ajuste estacional", "Tendencia"))
```

```
## Organización de gráficos de la Serie Original y Descompuesta en una cuadrícula 2x1 ##
```

```
grid.arrange(GF1, GF2, nrow=2)
```

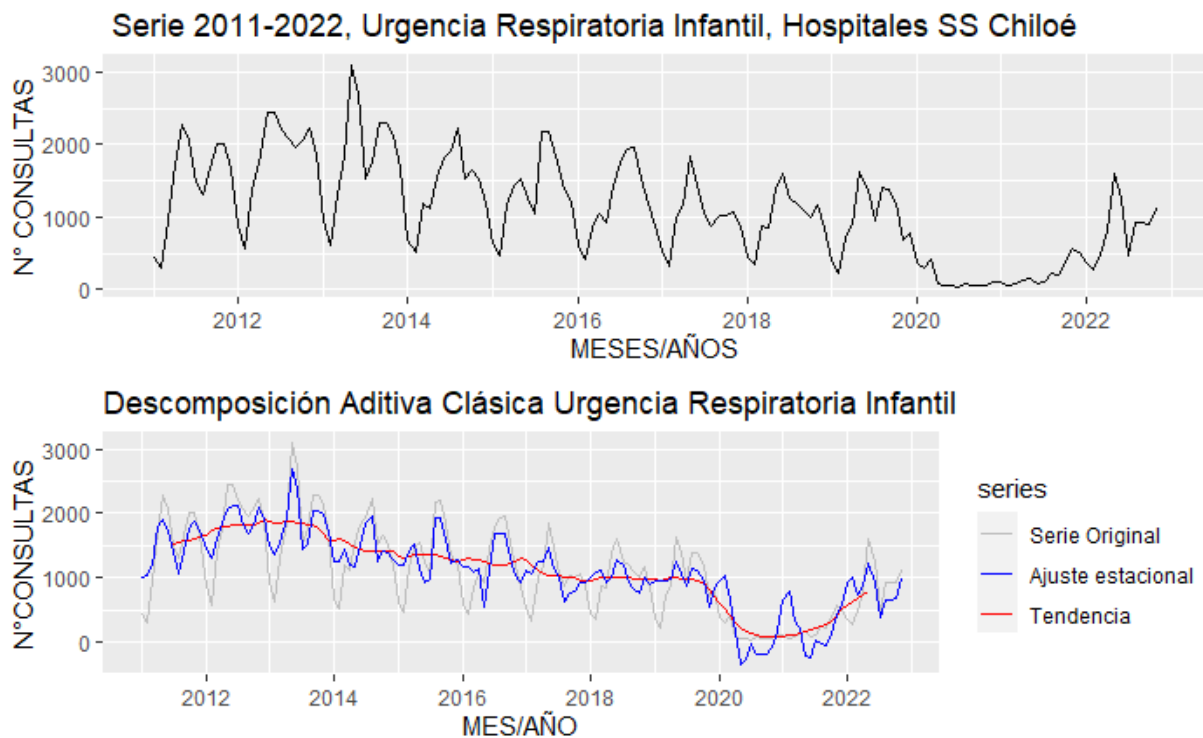


Figura 1: Descripción y descomposición aditiva clásica de la serie temporal.

Se pudo constatar en el gráfico de la serie que existe cierta tendencia, y estacionalidad (12 meses), además se observó que la serie no es estacionaria en media, ya que tiene mayores demandas de consultas antes del año 2017, y lo más importante se ve fuertemente impactada por los dos años de pandemia Covid-19 en Chiloé, periodo que se extiende aproximadamente, desde marzo 2020 hasta diciembre 2021. Además, la serie probablemente podría describirse usando un modelo aditivo, ya que las fluctuaciones aleatorias son aproximadamente constantes en el tiempo.

3.1.2 Evaluación de puntos de cambio en la serie temporal.

Se observaron dos niveles medios diferentes y el cambio entre ellos, en la serie temporal de la demanda por consultas de urgencia respiratoria pediátrica. El punto de cambio se da en el mes de octubre del año 2019, antes de esta fecha se presenta un nivel medio de 1358 consultas mensuales y posterior a esta, se reduce a 399 consultas, lo cual se evaluó, según el siguiente código:

```
## Identifica los puntos de corte (Numero de registro de la serie) ##  
pts.media <- cpt.mean(TS)  
  
## Grafica los puntos de corte (Numero de registro de la serie) ##  
plot(cpt.mean(TS),  
      penalty= "BIC",  
      method= "AMOC",  
      test.stat="Normal",  
      main="Cambios Significativos de la Media en la Serie Temporal",  
      xlab="MES/AÑO",  
      ylab="N° DE CONSULTAS")
```

Ahora este fenómeno se explica con el inicio del periodo de Pandemia Covid-19 y todo lo ocurrido en el sistema de salud, producto de esto hasta el día de hoy, tanto en el país como en la provincia de Chiloé. Por otra parte, no se observaron cambios significativos en la varianza a través de la serie temporal.

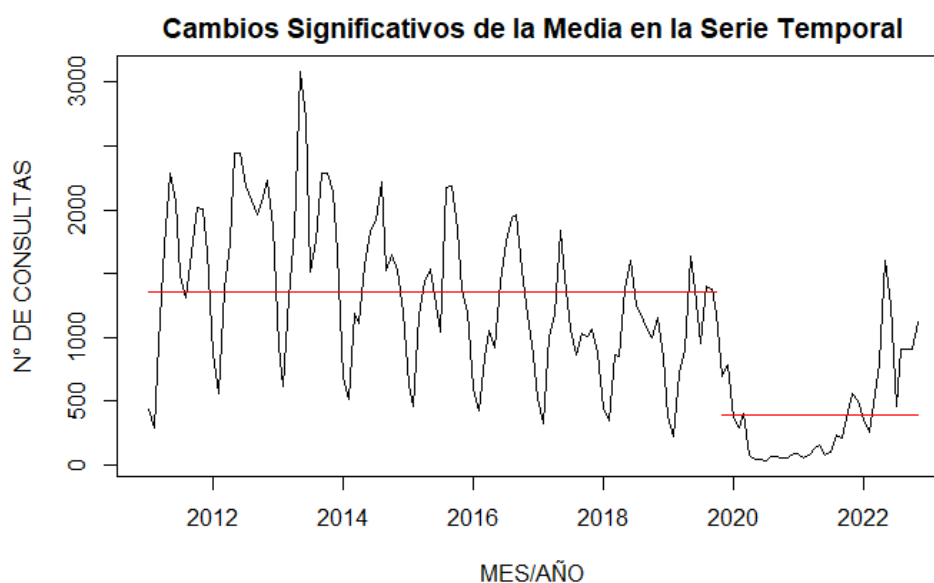


Figura 2: Detección de cambios en la media de la serie temporal

3.1.3 Evaluación de quiebres estructurales en la serie temporal.

La datación de cambios estructurales consiste en determinar si existen rupturas estructurales en el proceso de generación de datos de series de tiempo, y de ser así, sus fechas. El análisis de la intervención estima el efecto de una intervención externa o exógena en una serie de tiempo. Como en este caso de intervención, un cambio de nivel permanente, como veremos en serie temporal de consultas de urgencia respiratoria infantil observada en la red hospitalaria del Servicio de salud Chiloé, periodo enero 2011 a noviembre 2022. En este escenario, la intervención externa o exógena no se conoce de antemano, se infiere de la ruptura estructural que se pretende identificar. Para ello, se ejecutó el siguiente código:

```
## Cálculo de puntos de quiebre en relaciones de regresión. Dada una cantidad de rupturas, la función calcula los puntos de ruptura óptimos. ##  
  
(break_point <- breakpoints(TS ~ 1, hpc="none"))  
  
## Graficación de puntos de quiebre estructural ##  
plot(break_point)
```

El valor mínimo de BIC se alcanzó con $m = 2$, por ello se trazó la serie de tiempo original contra sus rupturas estructurales y sus intervalos de confianza.

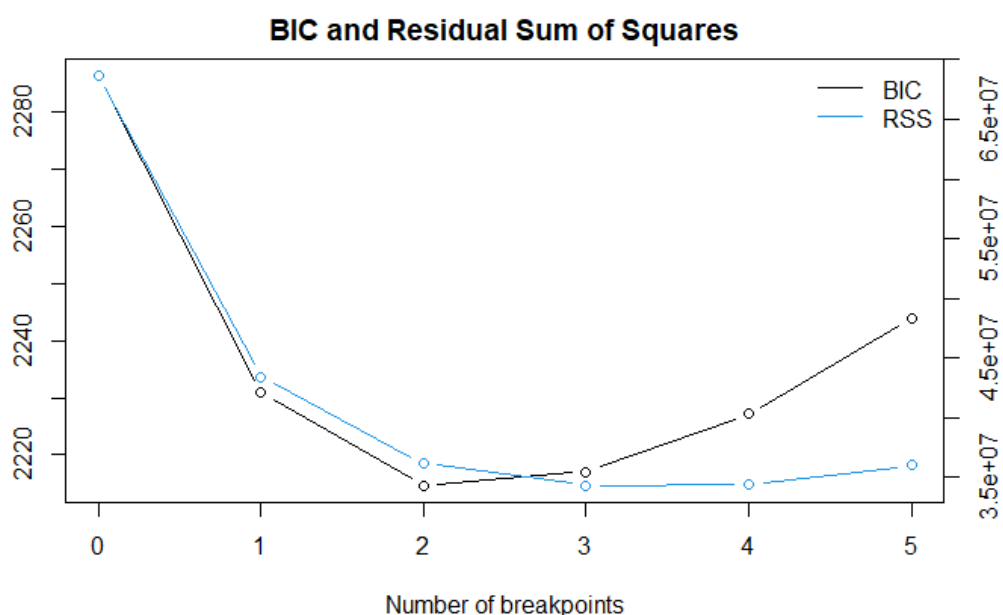


Figura 3: Identificación de quiebres estructurales en la Serie Temporal.

En relación a las rupturas estructurales, se observó dos hitos importantes en la historia de las consultas de urgencia respiratoria pediátrica, en la Urgencia Hospitalaria del Servicio de Salud Chiloé, lo cual se obtuvo, a través del código a continuación:

```
## Elaboración de gráfico de serie temporal ##
plot(TS, main="Rupturas Estructurales en la Serie Temporal",
      xlab="AÑOS/ MESES",
      ylab= "N° CONSULTAS")

## Ajuste de líneas verticales de rupturas estructurales ##
fitted.ts <- fitted(break_point, breaks =2)

## Ajuste de líneas horizontales de rupturas estructurales ##
lines(fitted.ts, col =3)

## Ajuste de líneas intervalos de confianza de rupturas estructurales ##
lines(confint(break_point, breaks =2))
```

El primero de estos hitos, se da en torno al año 2016 con el inicio de funcionamiento de los dispositivos de atención de urgencia en el nivel primario de salud, fundamentalmente en la comuna de Castro, estos dispositivos son establecimientos denominados SAR (Servicio de Atención Primaria de Urgencias de Alta Resolución), los cuales absorben gran parte de la demanda por atención de baja complejidad, impactando con ello en la demanda hospitalaria de atención de urgencia.

Ahora este fenómeno también se refuerza en otras comunas de la red asistencial, como Ancud, que al día de hoy también cuenta con este dispositivo, y en el resto del territorio rurales con los SUR (Servicio de Urgencia Rural).

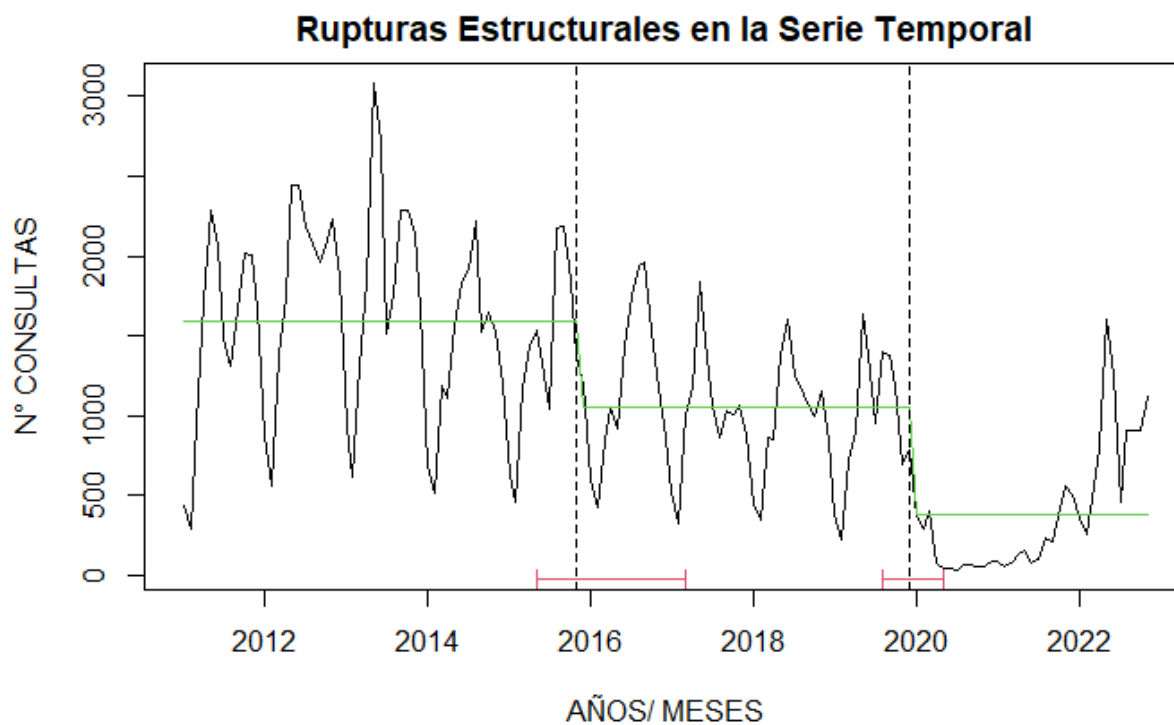


Figura 4: Rupturas estructurales y sus intervalos confianza.

El segundo hito que marca una ruptura en el patrón de atención de urgencia hospitalaria, fue el inicio del periodo de la pandemia de Covid-19, producto fundamentalmente de la adecuación de las unidades de urgencia hospitalarias y el clima de temor e incertidumbre reinante en la población, producto del desconocimiento de las consecuencias de esta nueva enfermedad.

3.1.4 Propuesta metodológica para tratamiento de datos Outliers durante el Periodo de Pandemia Covid-19 en Chiloé.

Según se describió en el análisis de exploratorio de la serie temporal, a través de la identificación de puntos de cambios de la media y rupturas estructurales de la serie temporal, el impacto de la pandemia Covid-19 sobre las demanda de consultas respiratorias infantiles en hospitales de la red asistencial, fue en la práctica catastrófico, manteniendo constante este efecto, a través de toda la duración de la misma, lo cual abarco un periodo aproximado de dos años (2020 al 2021), iniciando el 3 de marzo de 2020, con el primer caso de Covid-19 en Chile y formalizándose el 25 de marzo con primer caso en la provincia de Chiloé.

Luego, es evidente que durante todo este periodo pandemia, la demanda por consultas de urgencia fue “prácticamente nula”, lo cual es una anomalía en el patrón normal que presenta la serie temporal de urgencia. Lo anterior puede explicarse por el clima de temor e incertidumbre reinante en la población, producto del desconocimiento de las consecuencias de esta nueva enfermedad, y porque, además, los hospitales modificaron sus procesos clínicos y asistenciales significativamente, en función de la contingencia sanitaria, solicitando a los usuarios concurrir a la urgencia solo ante un evento de riesgo vital. Luego, conociendo las causas que explicaron fehacientemente este fenómeno, se propone anular todos los registros observados durante ese periodo (durante la pandemia Covid-19), en Chiloé, comprendido entre abril 2020 y diciembre 2021, y obtener sus estimaciones mediante un método de imputación, con el objetivo de generar una serie con un patrón de comportamiento lo más cercano, a lo esperado en un escenario sin pandemia.

En el caso de este trabajo se utilizará el Filtro de Kalman con modelos ARIMA de espacio de estados: todos los modelos ARIMA pueden ser reescritos (y estimados) como modelos en el espacio de estados. Estos modelos en el espacio de estados, pueden entre otras muchas cosas, estimar observaciones perdidas.

Luego los principales beneficios de representar sistemas dinámicos en forma de espacio de estados es que permite que las variables no observadas puedan ser incorporadas y estimadas junto con el modelo observable, y que, además, estos modelos en el espacio de estados pueden ser analizados

usando un algoritmo recursivo llamado filtro de Kalman, que es usado, entre otras cosas, para determinar las estimaciones óptimas de los modelos en el espacio de estados. En este caso el procedimiento de imputación de datos nulos, se ejecutara, utilizando la librería *imputeTS*, a través de la función *na_kalman*, con modelos ARIMA de espacio de estados, utilizando para ello la función *auto.arima*, en el argumento de la función, según el siguiente código:

```
## Anulación de datos observados en periodo Covid-19, para ser considerados como información faltante ##
TS[111:132]<-NA ## marzo 2020 - diciembre 2021 ##

## Imputación de datos faltantes mediante el Método de Kalman ##
Clean_TS_Kalman <- round(na_kalman(TS,
                                model="auto.arima",
                                smooth=T,
                                stationary=T),digits = 0)
```

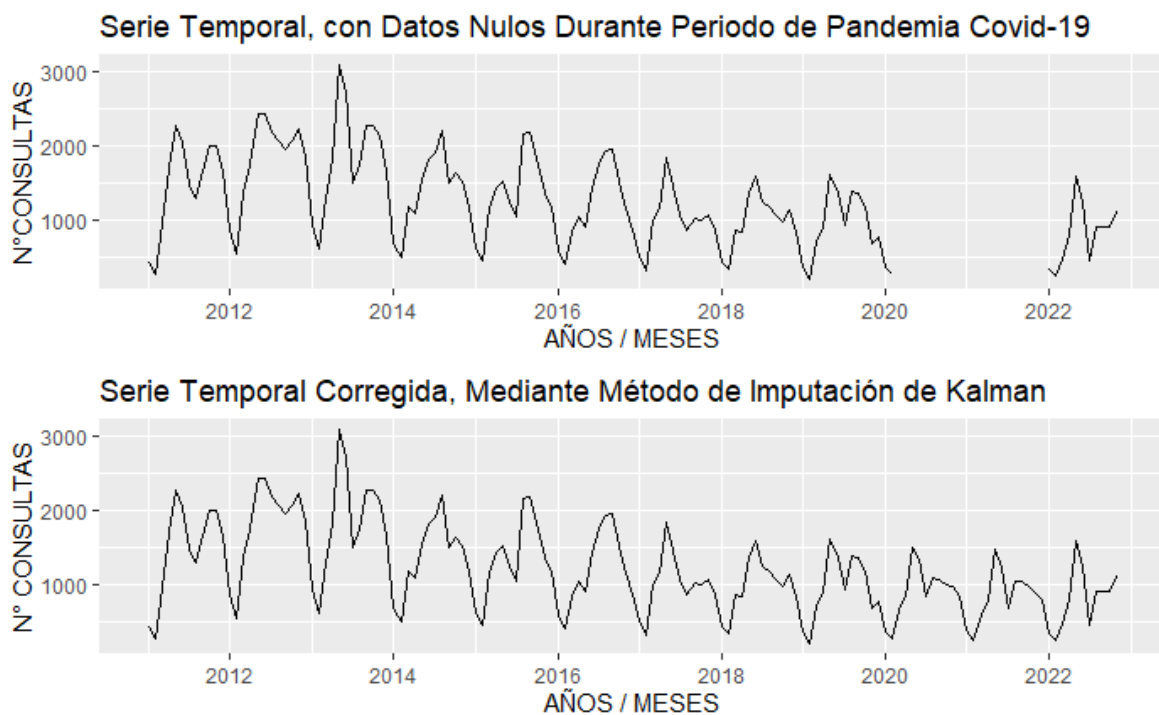


Figura 5: Serie temporal con datos "Nulos" vs. Serie temporal corregida según Filtro de Kalman.

Anteriormente, se describe la anulación de todos los datos observados durante el periodo de pandemia en la provincia de Chiloé (marzo 2020 - diciembre 2021), en contraste con la serie temporal corregida mediante el método de Imputación de Kalman, de esta forma se obtuvo una serie de datos completa y sin anomalías por causal y externa, al proceso normal de consultas de urgencia respiratoria infantil, en los hospitales de la provincia de Chiloé.

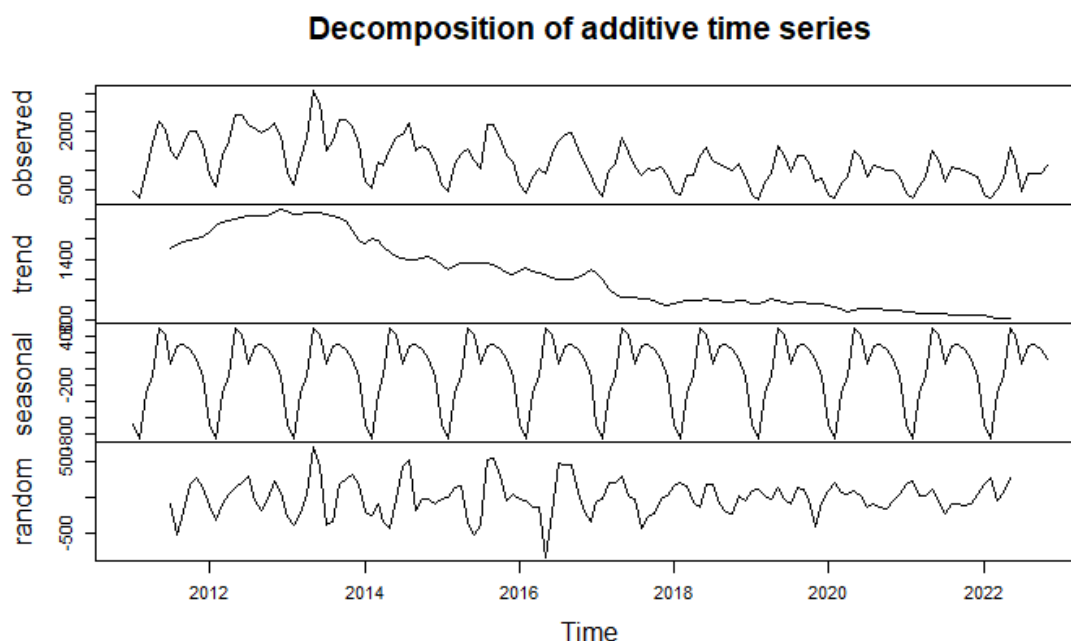


Figura 6: Descomposición aditiva de la serie temporal corregida, según Imputación de Kalman

Se observó que la serie temporal corregida (mediante imputación de Kalman), presenta tendencia y estacionalidad⁷ bastante definidas, en términos visuales, por lo tanto, se prevé la necesidad de diferenciarla, tanto, a nivel regular como estacional. Luego, es muy probable que la serie no cumpla con el supuesto estadístico de Estacionariedad⁸, dado los argumentos planteados anteriormente.

⁷ La estacionalidad es una característica de una serie de tiempo en la que los datos experimentan variaciones regulares y predecibles que se repiten cada período. Por ejemplo la estacionalidad puede ocurrir cada año, cada trimestre o cada lunes.

⁸ La estacionariedad significa que el proceso mediante el cual se crearon los datos, es constante en el tiempo. Es decir, la media y la variabilidad se mantienen constantes a lo largo del tiempo, luego, no es en función del tiempo; y además, no presenta tendencia.

3.1.5 Verificación de supuestos estadísticos de la serie temporal corregida.

Como parte del método Box & Jenkins que fue generado en el año 1970 buscando facilitar el trabajo al construir un modelo de una serie temporal⁹, para explicar su estructura y predecir la evolución de esta serie en el futuro, se verificaron los supuestos mediante las pruebas estadísticas correspondientes en cada caso, según se describe en el siguiente código:

```
ndiffs(Clean_TS_Kalman, test = "kpss")  ## Verificación diferencia regular

nsdiffs(Clean_TS_Kalman, test = "seas") ## Verificación diferencia estacional
## Dickey-Fuller Aumentado (Augmented Dickey-Fuller Test (ADF) hipótesis nula
es precisamente la presencia de una raíz unitaria en el proceso generador de
datos de la serie analizada ##
adf.test(Clean_TS_Kalman)

## La prueba de Ljung-Box La prueba Q de Ljung-Box considera la hipótesis nul
a de que las autocorrelaciones de hasta un desfases son iguales a cero ##
Box.test(Clean_TS_Kalman, type="Ljung-Box")

## Prueba no paramétrica de Cox-Stuart nos permite evaluar si existe una tend
encia (monótona) significativa en los datos ##
coxstuart(Clean_TS_Kalman, type = "trend")

## Esta función utiliza la prueba de Friedman, que proporciona los índices es
tacionales estimados ##
seas<-seasplot(Clean_TS_Kalman)
str(seas)
```

Realizadas las pruebas, se denoto que es pertinente diferenciar la serie en términos regulares y estacionales (primeras diferencias). El resultado de todos los test aplicados presento evidencia estadística suficiente para argumentar que la serie temporal no presento un comportamiento estacionario. Además, se observó que la prueba de Ljung-Box fue significativa y por tanto se presentaron autocorrelaciones de hasta un desfase. La Prueba no paramétrica de Cox-Stuart confirmo que existió una tendencia (monótona) significativa en los datos analizados. Finalmente, la prueba de Friedman, que proporciono los índices estacionales estimados, se puede decir que la serie también presento estacionalidad.

⁹ <https://www.researchgate.net/publication/274961095> Modelos ARMA y Box and Jenkins

3.1.6 Verificación de supuestos estadísticos de la serie temporal corregida diferenciada en componente regular y estacional.

Considerando toda la evidencia estadística anterior, que otorgaron los diferentes test aplicados, más la evidencia visual proporcionada por las herramientas graficas utilizadas para evaluar la serie temporal, se procede a diferenciar la serie en sus componentes regular y estacional.

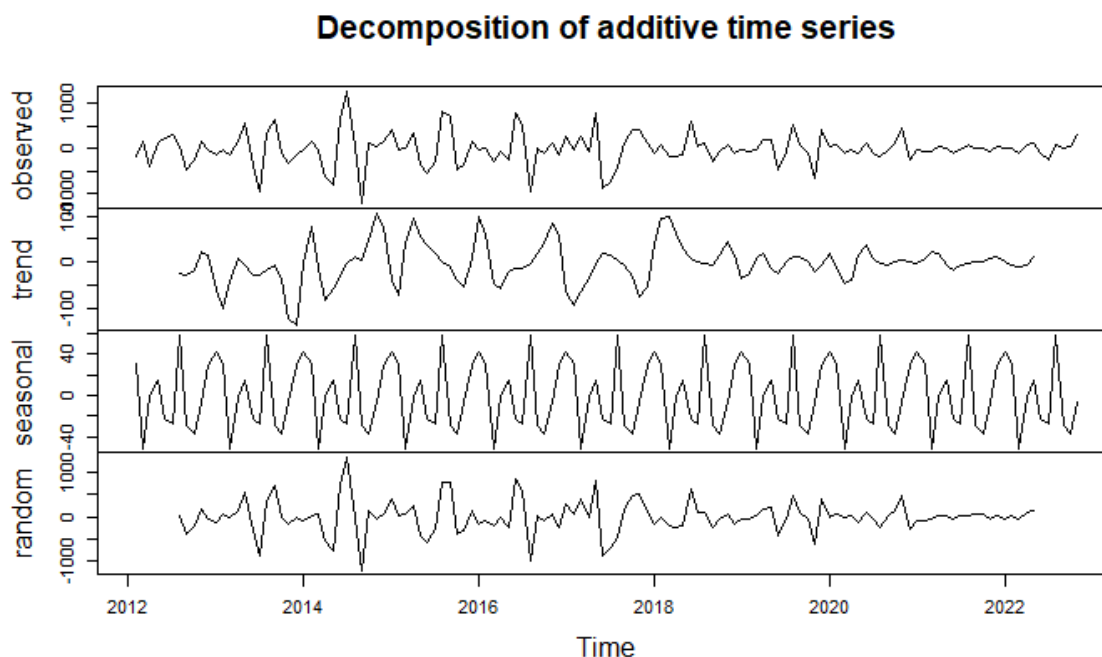


Figura 7: Descomposición aditiva de la serie temporal corregida, según Imputación de Kalman, diferenciada en sus primeras diferencias en componente regular y estacional.

La serie temporal de las primeras diferencias, regular y estacional, parece ser estacionaria en media y varianza, por lo que un modelo $ARIMA(p, 1, q)$ ($P, 1, Q$)[12] es probablemente apropiado para la serie de consultas de urgencia respiratoria pediátrica, en la provincia de Chiloé. Por otra parte, del análisis y evaluando del correlograma y el correlograma parcial de la serie de tiempo estacionaria, se observa en el correlograma (ACF) que dos retardos exceden significativamente los intervalos de confianza (el N°2 y N°10), además del retardo 12 que establece un componente autorregresivo estacional altamente significativo.

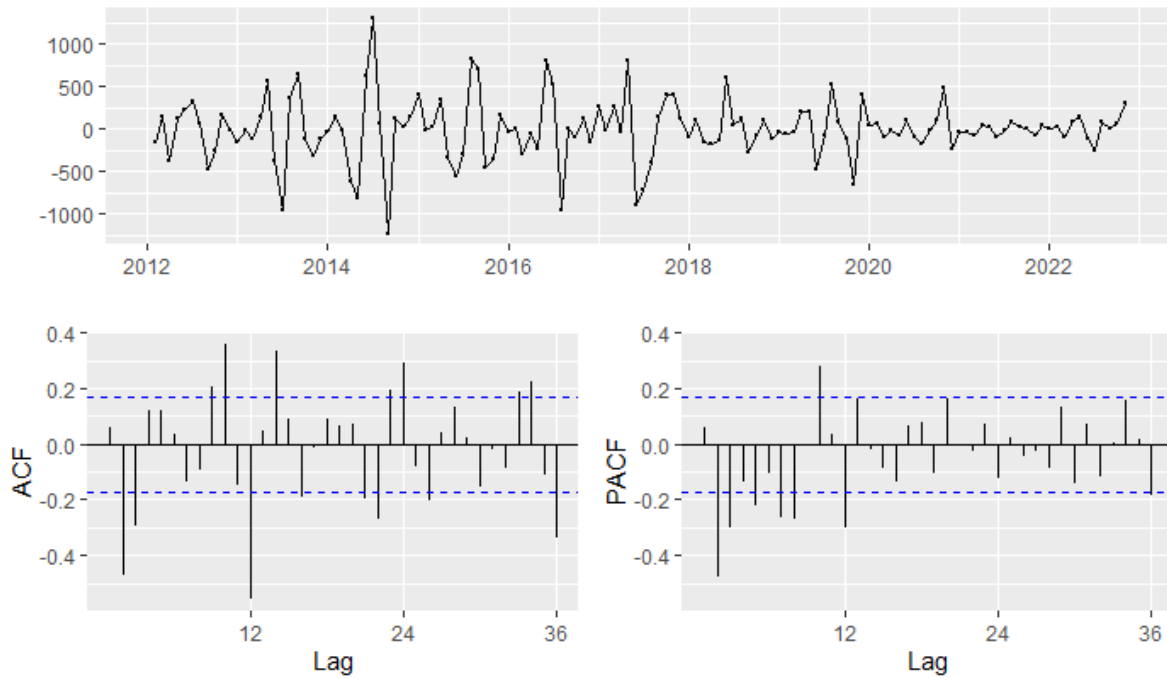


Figura 8: Serie temporal corregida, según Imputación de Kalman, diferenciada en sus primeras diferencias en componente regular y estacional vs. Gráficos de ACF y PACF.

De la misma forma, el correlograma parcial muestra un retardo que excede significativamente los intervalos de confianza (el $N^{\circ}2$) y otro en menor medida ($N^{\circ}10$), también se observa un componente de media móvil estacional de baja significación.

```
## Autocorrelations of series 'diffClean_TS_Kalman', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
## 1.000  0.057 -0.471 -0.297  0.117  0.122  0.034 -0.135 -0.095  0.205  0.357
##      11     12     13     14     15     16     17     18     19     20
## -0.148 -0.554  0.048  0.333  0.089 -0.193 -0.012  0.092  0.064  0.072
```

Figura 9: Autocorrelaciones de serie temporal corregida, según Imputación de Kalman, diferenciada en sus primeras diferencias en componente regular y estacional.

Según lo anterior se conjetura un modelo SARIMA (2, 1, 1) (1, 1, 0) [12], el cual será estimado y evaluado mediante método analítico tradicional, sin la intervención del método automático, disponible con la función *auto.arima*, siendo este último evaluado de manera individual.

```

## Partial autocorrelations of series 'diffClean_TS_Kalman', by lag
##
##      1      2      3      4      5      6      7      8      9     10     11
## 0.057 -0.475 -0.298 -0.135 -0.223 -0.103 -0.266 -0.269 -0.005 0.277 0.036
##     12     13     14     15     16     17     18     19     20
## -0.301 0.162 -0.021 -0.089 -0.139 0.065 0.075 -0.104 0.162

```

Figura 10: Autocorrelaciones Parciales de serie temporal corregida, según Imputación de Kalman, diferenciada en sus primeras diferencias en componente regular y estacional.

Para terminar esta etapa, realizadas las pruebas, ahora sobre la serie temporal corregida, diferenciada en sus primeras diferencias en componente regular y estacional, se comprobó que ya no es pertinente diferenciar la serie en términos regulares y estacionales (segundas diferencias). El resultado de todos los test aplicados presento evidencia estadística suficiente para argumentar que la serie temporal presento un comportamiento estacionario. Se observó también, que la prueba de Ljung-Box no fue significativa ($p < 0.05$) y por tanto no presenta autocorrelaciones de hasta un desfase. la Prueba no paramétrica de Cox-Stuart confirmo que ya no existe una tendencia (monótona) significativa en los datos analizados. Y finalmente, la prueba de Friedman, sugiere que la serie no presento estacionalidad.

3.2 Partición de muestra de datos de la serie temporal.

Para la elaboración del conjunto de *entrenamiento (train)*, el cual, se utilizó para entrenar los algoritmos de *Machine Learning* y descubrir posibles relaciones en los datos, y el conjunto de *prueba o test*, que se utilizó para verificar la precisión de los modelos estimados (a partir del conjunto de entrenamiento), y la fuerza de estas posibles relaciones para predecir datos futuros de la serie temporal, se consideró utilizar para el conjunto de *test*, los once meses finales de la serie temporal, es decir, desde enero hasta noviembre del año 2022, luego este conjunto representa un 7,69% del total de la serie disponible. En consecuencia, el porcentaje utilizado para entrenamiento o *train* de los algoritmos de *Machine Learning*, fue del 92,3% del total. La partición de la serie temporal se realizó según el código a continuación:

```
# Partición de muestra de datos de la Series temporal ##

train <- window(Clean_TS_Kalman,
                end = c(2021,12)) ## Muestra de entrenamiento ##

test <- window(Clean_TS_Kalman,
               start = c(2022,1)) ## Muestra de test o prueba ##
```

La extensión del periodo de *test*, es consecuencia del proceso de Anulación e Imputación de los datos observados durante el periodo de pandemia Covid-19, en la provincia de Chiloé, metodología que se explica en detalle anteriormente en este trabajo. Lo anterior fundamentado en que el *conjunto de test, no contiene datos imputados*, y comprende solo información observada post pandemia, de lo cual, se espera un proceso de *prueba (test)* de los modelos estimados, que no esté sesgado por las estimaciones realizadas en el proceso de imputación mediante el Método de Kalman.

3.3 Modelamiento de la Serie Temporal.

3.3.1 Modelo ARIMA, mediante método de modelado automático.

Para verificar la elección del modelo SARIMA (2, 1, 1) (1, 1, 0) [12], indicado anteriormente, y/o para encontrar el modelo ARIMA apropiado de manera automática, se utilizó primeramente la función *auto.arima*, según el siguiente código:

```
## Estimación de Modelo ARIMA, mediante función "auto.arima", sobre Serie de  
"train" ##  
  
fit_autoarima <-auto.arima(train,  
                           trace = T,  
                           ic = "bic")
```

Este método proporciona una opción rápida para construir pronósticos con series temporales, debido a que evalúa entre todos los posibles modelos, al mejor modelo considerando diversos criterios: estacionariedad, estacionalidad, diferencias, entre otras.

```
Best model: ARIMA(1,0,1)(0,1,1)[12] with drift  
  
z test of coefficients:  
  
      Estimate Std. Error z value Pr(>|z|)  
ar1    0.348102   0.109732   3.1723 0.001512 **  
ma1    0.458578   0.088469   5.1835 2.178e-07 ***  
sma1  -0.868182   0.165050  -5.2601 1.440e-07 ***  
drift  -7.686167   1.242541  -6.1858 6.177e-10 ***  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
      2.5 %    97.5 %  
ar1    0.133031  0.5631722  
ma1    0.285181  0.6319750  
sma1   -1.191674 -0.5446897  
drift -10.121502 -5.2508316
```

Figura 11: Estimación de parámetros modelo ARIMA (1,0,1) (0,1,1) [12].

Una vez ejecutada, la función *auto.arima* devuelve el posible mejor modelo de entre todos y realiza transformaciones de la variable de ser el caso que no sea estacionaria; adicionalmente, evalúa presencia de estacionalidad en la serie y si existe este componente, construye modelos SARIMA. En este caso determino que el mejor modelo es ARIMA(1,0,1)(0,1,1) [12], el cual difiere en varios aspectos al modelo candidato, identificado anteriormente, SARIMA (2,1,1)(1,1,0) [12], siendo una de los más importantes, la diferenciación a nivel regular de la serie temporal y el componente estacional

de Medias Móviles (AR), también es dable mencionar, que ninguno de los modelos evaluados por el método *auto.arima* se asemeja al modelo candidato, esto porque todos ellos no consideran diferenciación regular.

Los parámetros del mejor modelo son todos significativos ($p\text{-value} \leq 0.001$), Sin embargo, presenta un elevado valor de MAPE en los *datos de test, superior al 30%*, existiendo una amplia brecha con el MAPE obtenido en los *datos de train (15%)*, lo cual es un indicio que el *modelo estimado es poco robusto*¹⁰ al momento de predecir nueva información, ya que *se observa un sobreajuste (overfitting)* sobre los datos de entrenamiento (train), y una *escasa generalización*¹¹ *de lo aprendido sobre los datos de test*. Finalmente, la prueba de Ljung-Box aplicada a los residuales del modelo, resulta estadísticamente significativa ($p\text{-value} \leq 0.00109$), luego la serie de residuales en un período de tiempo específico (24 retardos) no son aleatorios e independientes. Según lo anterior, los errores no son independientes, implicando que uno puede estar correlacionado con otro *error*, k unidades de tiempo después, una relación que se denomina autocorrelación, la cual puede reducir la exactitud de un modelo predictivo basado en el tiempo, y conducir a una interpretación errónea de los datos.

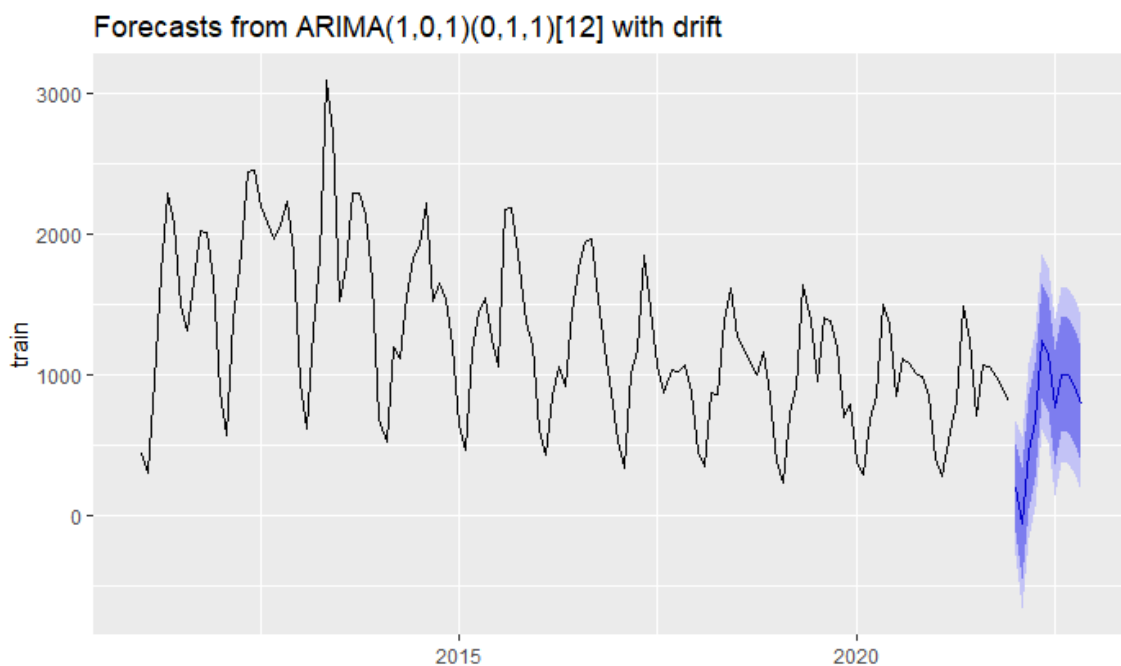


Figura 12: Predicción de 11 periodos, modelo ARIMA (1,0,1) (0,1,1) [12].

¹⁰ Es una medida de la capacidad que tiene el método analítico de permanecer inalterado por pequeñas variaciones en el procedimiento del método, en: <https://dialnet.unirioja.es/descarga/articulo/4550292.pdf>

¹¹ Generalización es la capacidad de obtener buenos resultados con datos nuevos, en: <https://www.iartificial.net/generalizacion-en-machine-learning/>

3.3.2 Modelo ARIMA, mediante análisis de correlograma ACF y PACF.

Mediante la función *arima*, se estimó el modelo candidato identificado anteriormente, SARIMA (2,1,1) (1,1,0) [12], como consecuencia del análisis de su correlograma simple (ACF) y parcial (PACF). A continuación, se presenta el código para la estimación del modelo y sus parámetros:

```
## Estimación de Modelo SARIMA (2,1,1) (1,1,0) [12], sobre Serie de "train" ##
fit_sarima <-arima(train,
                   order = c(2,1,1),
                   seasonal = list(order = c(1,1,0),
                                   period = 12))

## Estimación de parámetros del modelo y sus correspondientes intervalos de c
onfianza ##
coeftest(fit_sarima)
```

Los parámetros estimados para el modelo son todos significativos ($p\text{-value} \leq 0.000$). Presenta un valor de MAPE muy aceptable en los *datos de test*, alcanzando un **16,03%**, existiendo una mínima brecha con el MAPE obtenido en los datos de *train* (**15,19%**), lo cual es un indicio que el modelo estimado es bastante robusto al momento de predecir nueva información, ya que **no se observa un sobreajuste¹² (overfitting)**, sobre los *datos de entrenamiento (train)*, y tiene la capacidad de generalizar lo aprendido, a partir de estos últimos, sobre los *datos de test*.

```
z test of coefficients:
      Estimate Std. Error  z value  Pr(>|z|)
ar1    0.605603   0.094487   6.4094 1.461e-10 ***
ar2   -0.417732   0.087660  -4.7654 1.885e-06 ***
ma1   -0.804636   0.068790 -11.6969 < 2.2e-16 ***
sar1  -0.452632   0.079388  -5.7015 1.188e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

      2.5 %    97.5 %
ar1    0.4204119  0.7907939
ar2   -0.5895411 -0.2459221
ma1   -0.9394625 -0.6698095
sar1  -0.6082296 -0.2970337
```

Figura 13: Estimación de parámetros modelo ARIMA (2,011) (1,1,0) [12].

¹² Es un concepto en la ciencia de datos, que ocurre cuando un modelo estadístico se ajusta exactamente a sus datos de entrenamiento. Cuando esto sucede, el algoritmo desafortunadamente no puede funcionar con precisión contra datos que no conoce, frustrando su propósito, en: <https://www.ibm.com/cl-es/topics/overfitting>

Finalmente, la prueba de Ljung-Box aplicada a los residuales del modelo, no resulta estadísticamente significativa ($p\text{-value} \geq 0.05901$), luego la serie de residuales en un período de tiempo específico (24 retardos) son aleatorios e independientes. Según esto, los errores son independientes, luego no existe autocorrelación, la cual favorece la exactitud de un modelo predictivo basado en el tiempo, y conduce a una interpretación confiable de los datos.

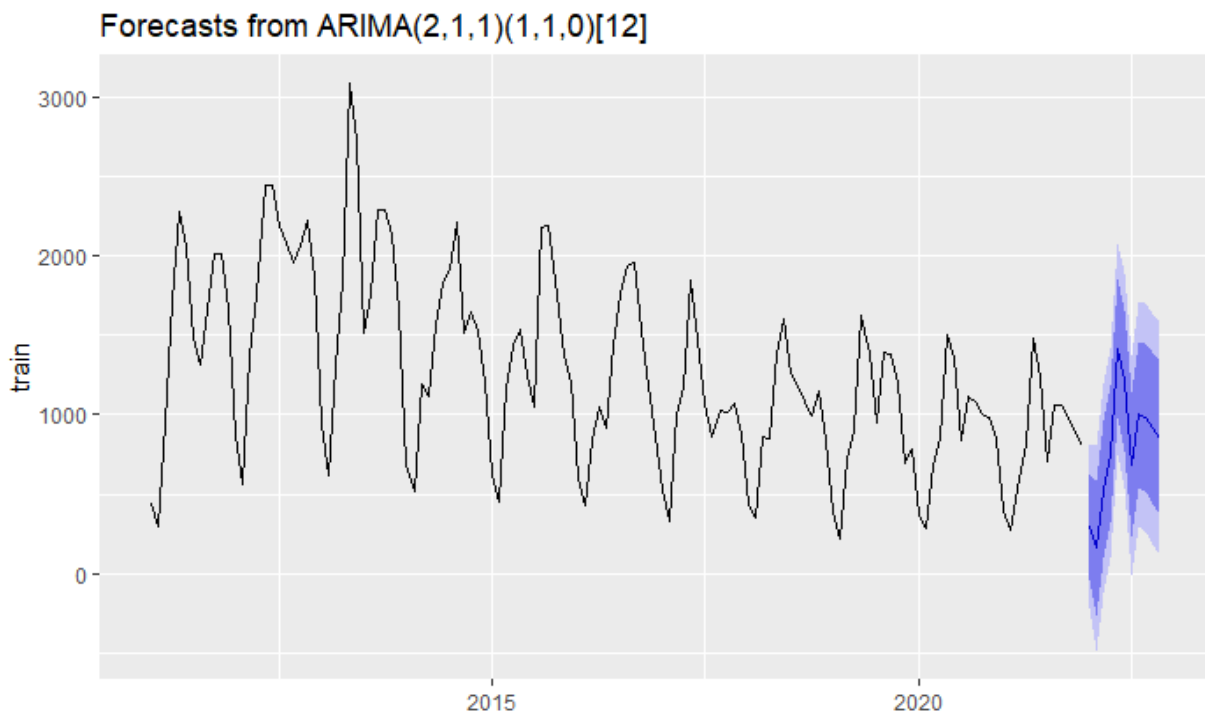


Figura 14: Predicción de 11 periodos, modelo ARIMA (2,1,1) (1,1,0) [12].

3.3.3 Modelo de Autorregresión de Redes Neuronales (NNAR).

La estructura del modelo Autorregresión de Redes Neuronales (NNAR), no se configuro de manera personalizada (es decir, numero de retardos utilizados como nodos de entrada, estacionalidad, capa oculta y número de nodos de esta), dejando que el algoritmo lo determinara automáticamente. Por otra parte, se configuro para realizar 100 repeticiones de entrenamiento y se estableció una transformación Box-Cox ¹³de lambda=0, esto último para garantizar que los pronósticos se mantengan positivos. El procedimiento anterior, se ejecutó mediante el siguiente código:

```
set.seed(123)## Semilla aleatoria ##

## Estimación de Modelo de Autorregresión de redes neuronales (NNAR) sobre S
erie de "train"##

(fit_nnetar <- nnetar(train,
                      repeats = 100,
                      lambda = 0))
```

Una vez ejecutado el modelo, este considero (automáticamente) los últimos valores observados de la misma temporada que las entradas, presentando una configuración NNAR (p,P,k)s, donde la estructura estimada fue: NNAR(8,1,5)12, es decir, una red neuronal con una primera capa que utiliza ocho retardos como nodos de entrada más el retardo N°12 como componente autorregresivo estacional, además cuenta con cinco nodos en la capa oculta y un nodo en la capa de salida.

```
Series: train
Model: NNAR(8,1,5)[12]
Call: nnetar(y = train, repeats = 100, lambda = 0)

Average of 100 networks, each of which is
a 9-5-1 network with 56 weights
options were - linear output units

sigma^2 estimated as 0.01183

```

	ME	RMSE	MAE	MPE	MAPE
MASE					
ACF1					
Theil's U					
Training set	11.64819	87.80713	58.92997	-0.2004364	4.737707
0.2231984	0.1729817	NA			
Test set	-30.81013	147.97806	117.05975	-10.3736334	16.929475
0.4433662	0.1198056	0.3487608			

Figura 15: Hiperparámetros y métricas de precisión, modelo Neuronal NNAR (8,1,5)12.

¹³ La familia de transformaciones de Box-Cox arregla problemas de normalidad y heterocedasticidad (no homogeneidad de varianzas), en: http://red.unal.edu.co/cursos/ciencias/2007315/html/un6/cont_17_78.html

Respecto de la precisión, este modelo presenta un valor de MAPE muy aceptable en los *datos de test*, alcanzando un 16,92%, existiendo, eso sí, una considerable brecha con el MAPE obtenido en los *datos de train (4,73%)*, lo cual es indicio que el modelo estimado es poco robusto al momento de predecir nueva información, ya que se observa un *sobreajuste (overfitting), sobre los datos de entrenamiento (train)*, y una escasa generalización de lo aprendido, de estos últimos, sobre los *datos de test*.

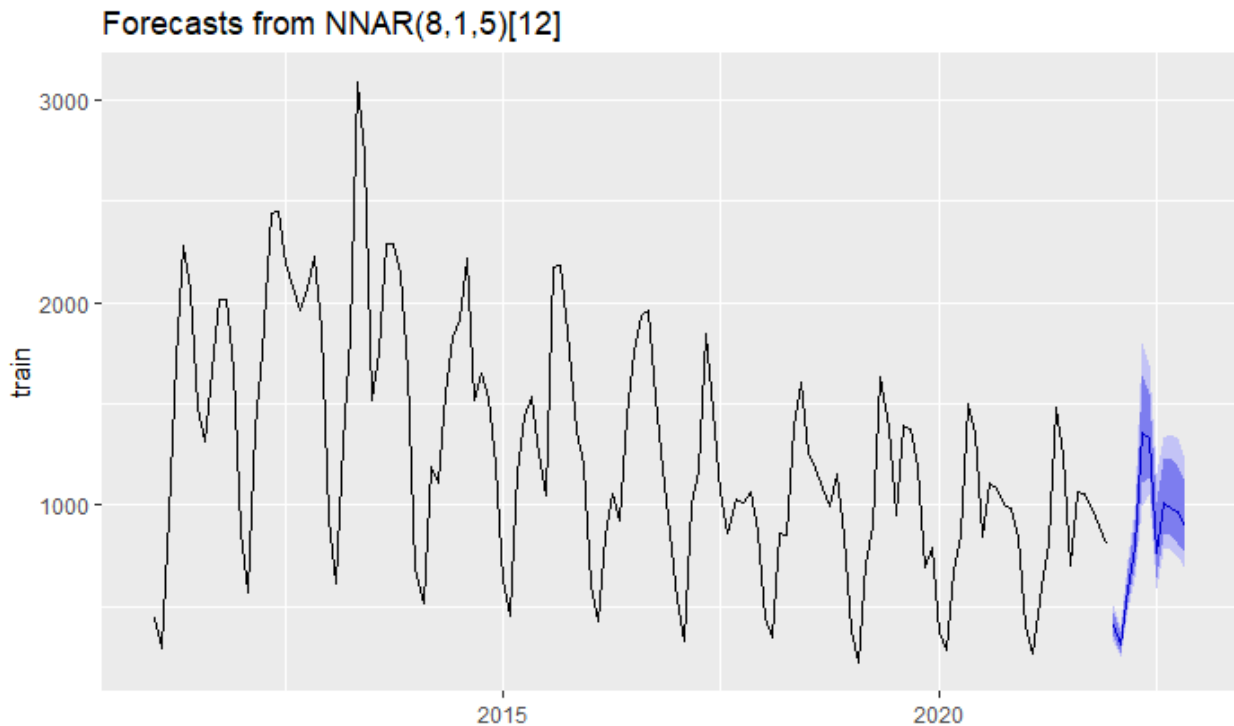


Figura 16: Predicción de 11 periodos, modelo Neuronal NNAR (8,1,5)12.

3.3.4 Modelo de Red Neuronal *Multilayer Perceptron* (MLP).

La estructura del modelo neuronal MLP no se configuro de manera personalizada (es decir, numero de retardos utilizados como nodos de entrada, capas ocultas y numero de nodos de estas), dejando que el algoritmo lo determinara automáticamente. El procedimiento anterior se realizó, según el siguiente código:

```
set.seed(123) ## semilla aleatoria

## Estimación de Modelo de Red Neuronal Multilayer Perceptron sobre Serie de
"train"##

fit_mlp <- mlp(ts(train,
                  frequency = 12,
                  end = c(2021,12)),
              reps = 100,
              difforder = c(1,12),
              comb="median")

## Grafo de estructura de la red neuronal Multilayer Perceptron ##
plot(fit_mlp)
```

Por otra parte, se configuro el argumento *reps* para realizar 100 repeticiones de entrenamiento y se estableció la forma en que se combinan las diferentes repeticiones de entrenamiento, en este caso mediante la *mediana*, utilizando el argumento *comb*.

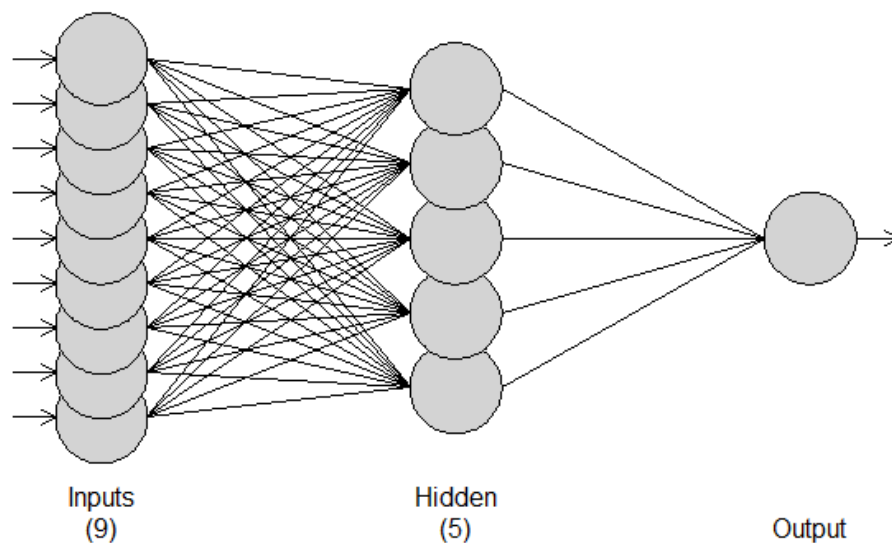


Figura 17: Hiperparámetros, modelo Neuronal *Multilayer Perceptron* (MLP).

Para terminar, dado que las redes neuronales presentan dificultades para modelar tendencias, se configuro el argumento *difforder* para realiza diferenciación en términos regulares y estacionales a la serie temporal.

	ME	RMSE	MAE	MPE	MAPE
MASE	ACF1	Theil's U			
Training set	-0.07935276	82.19531	58.85007	-0.9031676	5.760354
	0.1898647	-0.02974088	NA		
Test set	-33.36761614	148.55637	117.81008	-9.4520951	16.418571
	0.4462080	0.08804765	0.3401134		

Figura 18: Métricas de precisión, modelo Neuronal *Multilayer Perceptron* (MLP).

Una vez ejecutado, el algoritmo de manera automática, configuro la estructura neuronal con una primera capa que utilizo nueve retardos como nodos de entrada, cinco nodos en la capa oculta y un nodo en la capa de salida. Respecto de la precisión, este modelo presento un valor de MAPE muy aceptable en los datos de test, alcanzando un 16,41%, evidenciando, desafortunadamente, una considerable brecha con el MAPE obtenido en los datos de *train* (5,76%), lo cual, es indicio que el *modelo estimado es poco robusto al momento de predecir nueva información*, observándose un *sobreajuste (overfitting), sobre los datos de entrenamiento (train)*, y una escasa generalización de lo aprendido sobre los *datos de test*.

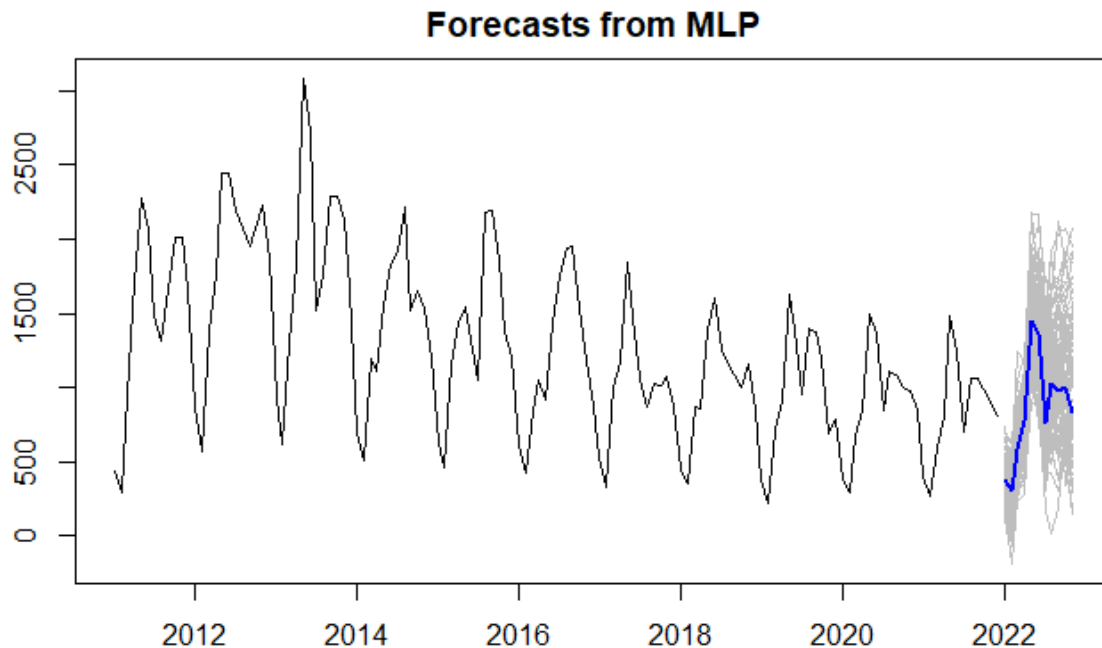


Figura 19: Predicción de 11 periodos, modelo Neuronal *Multilayer Perceptron* (MLP).

3.3.5 Modelo de Red Neuronal *Extreme Learning Machines* (EML).

La estructura del modelo neuronal EML no se configuro de manera personalizada (es decir, numero de retardos utilizados como nodos de entrada, capas ocultas y numero de nodos de estas), dejando que el algoritmo lo determinara automáticamente. Luego lo anterior se ejecutó según el código que se detalla a continuación:

```
set.seed(123) ## semilla aleatoria

## Estimación de Modelo de "Red Neuronal Extreme Learning Machines (EML), sobre Serie "train"##

fit_elm <- elm(ts(train,
                  frequency = 12,
                  end = c(2021,12)),
              type="step",
              difforder = c(1,12),
              reps = 100,
              comb = "median")

## Grafo de estructura de la Red Neuronal Extreme Learning Machines ##
plot(fit_elm)
```

Por otra parte, se configuro el argumento *reps* para realizar 100 repeticiones de entrenamiento y se estableció la forma en que se combinan las diferentes repeticiones de entrenamiento, en este caso mediante la *mediana*, utilizando el argumento *comb*.

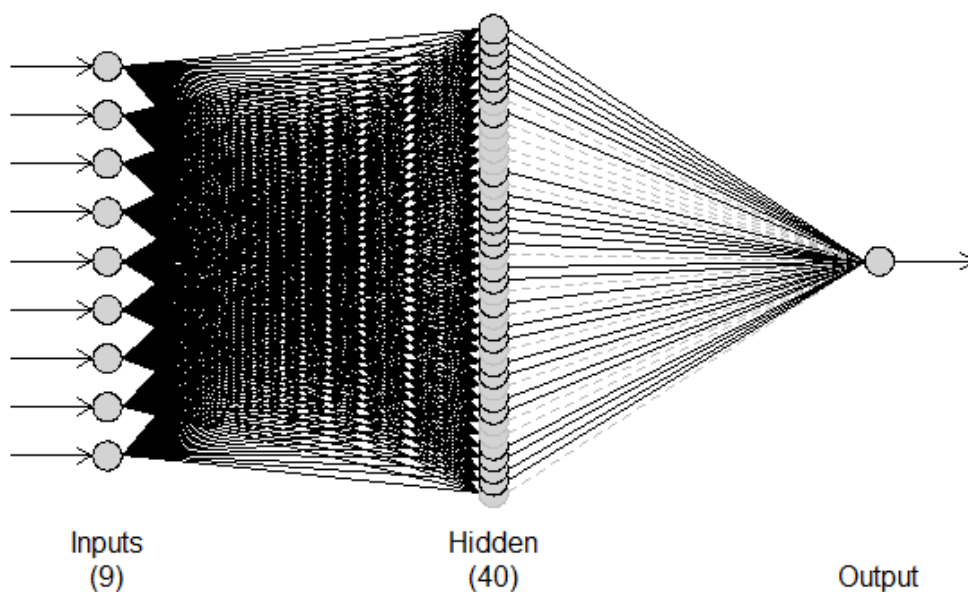


Figura 20: Hiperparámetros, modelo Neuronal *Extreme Learning Machines* (EML).

Finalmente, dado que las redes neuronales presentan dificultades para modelar tendencias, se utilizó el argumento *difforder*, para realiza diferenciación en términos regulares y estacionales a la serie temporal.

	ME	RMSE	MAE	MPE	MAPE	MASE
ACF1 Theil's U						
Training set	-0.8591725	184.2636	136.67479	-0.9604462	12.64907	0.4409463
	-0.03416117	NA				
Test set	44.2528149	138.9152	99.46038	2.0958407	13.14777	0.3767082
	0.12831353	0.3043961				

Figura 21: Métricas de precisión, modelo Neuronal *Extreme Learning Machines* (EML).

Una vez ejecutado el código, el algoritmo automáticamente, configuro la estructura de la red neuronal, con una primera capa que utilizo nueve retardos como nodos de entrada, cuarenta nodos en la capa oculta y un nodo en la capa de salida. Respecto de la precisión, este modelo presenta un valor de *MAPE muy aceptable en los datos de test, alcanzando un 13,14%*, existiendo una mínima brecha con el *MAPE* obtenido en los *datos de train (12,64%)*, lo cual es indicador que el *modelo estimado es bastante robusto al momento de predecir nueva información*, ya que no se observa un *sobreajuste (overfitting)*, sobre los *datos de entrenamiento (train)*, y generalizo de muy forma lo aprendido, de estos últimos, sobre los *datos de test*. Luego en primera instancia, evaluando la precisión de los algoritmos únicamente por el *MAPE*, este podría ser catalogado como el mejor de todos los algoritmos evaluados.

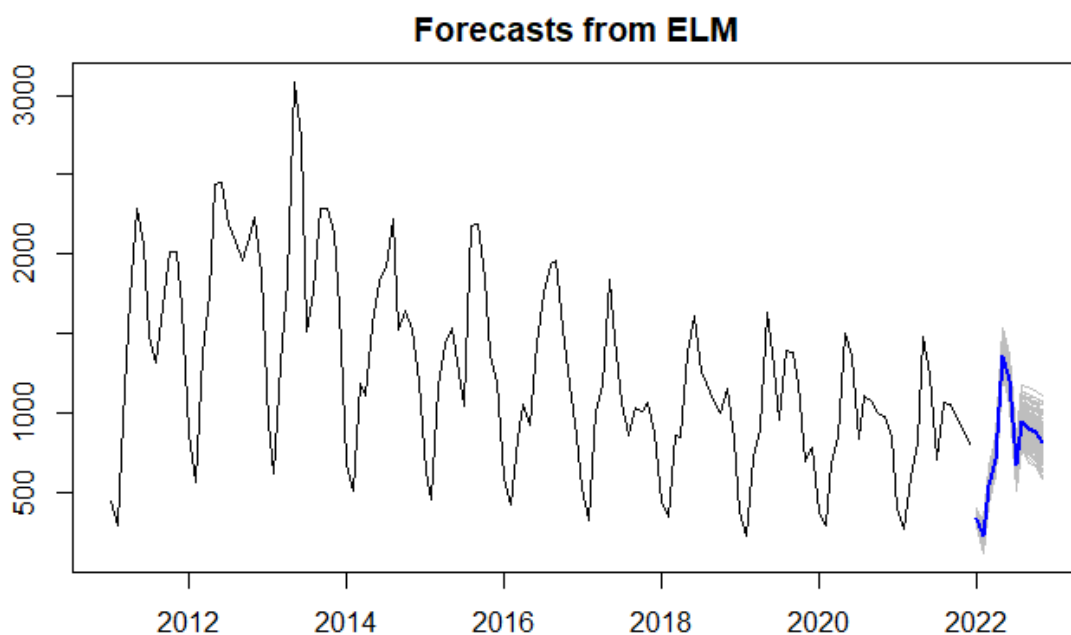


Figura 22: Predicción de 11 periodos, modelo Neuronal *Extreme Learning Machines* (EML).

3.4 Comparación de indicadores y pronósticos de algoritmos utilizados.

3.4.1 Indicadores de precisión de pronósticos de algoritmos.

Como era esperable, considerando los resultados de las visualizaciones e indicadores de precisión evaluados anteriormente, las diferencias entre los cuatro mejores modelos generados no son muy dispares y los cuatro algoritmos presentan valores bastante aceptables en cuanto al **Error cuadrático medio (RMSE)**, **Error absoluto medio (MAE)**, **Error Porcentual Absoluto Medio (MAPE)**. Para cotejar de mejor forma estas diferencias de rendimiento, se elaboró una tabla, utilizando el siguiente código:

```
## Indicadores de precisión de algoritmos utilizados sobre datos de "test" ##
tabl <- cbind.data.frame(m2,
                        m3, m4, m5)
kable(head(round(head(tabl), 2), 16),
       booktabs = TRUE,
       caption = "COMPARACION DE INDICADORES DE PRECISION DE ALGORITMOS, SOBRE
SERIE DE TEST") %>%
kable_classic()
```

	SARIMA	NNETAR	MLP	ELM
RMSE	129.79	147.98	148.56	138.92
MAE	102.79	117.06	117.81	99.46
MAPE	16.04	16.93	16.42	13.15

Tabla 1: Comparativa de indicadores de precisión de pronósticos de los modelos estimados, sobre los datos del conjunto de test.

Sin embargo, el mejor desempeño como promedio de los tres indicadores antes mencionados se da en la siguiente jerarquía:

- 1) **Red Neuronal Extreme Learning Machines (EML)**: Presenta el mejor desempeño en cuanto al Error absoluto medio (MAE) y Error Porcentual Absoluto Medio (MAPE), y el segundo mejor en Error cuadrático medio (RMSE), por debajo del modelo SARIMA, siendo este último el algoritmo que lo sigue en rendimiento de pronóstico.

- 2) **SARIMA (2,1,1) (1,1,0) [12]**: Presenta el mejor desempeño en Error cuadrático medio (RMSE), y el segundo mejor en Error absoluto medio (MAE) y Error Porcentual Absoluto Medio (MAPE), superado en ambos casos por la Red Neuronal EML.

- 3) **Autorregresión de Red Neuronal o NNAR**: Presenta el tercer mejor desempeño en Error cuadrático medio (RMSE) y en Error Absoluto Medio (MAE), luego en Error Porcentual Absoluto Medio (MAPE) presenta el peor rendimiento de los cuatro algoritmos.

- 4) **Red Neuronal Multilayer Perceptron (MLP)**: Presenta el tercer mejor desempeño en Error Porcentual Absoluto Medio (MAPE). Luego en Error Absoluto Medio (MAE) y en Error cuadrático medio (RMSE) presenta el peor rendimiento de los cuatro algoritmos.

3.4.2 Visualizaciones de pronósticos de algoritmos.

Finalmente, el análisis gráfico confirmó los resultados obtenidos en el punto anterior, según lo cual, el algoritmo que demostró un mejor rendimiento al momento de predecir la demanda mensual de consultas de urgencia respiratoria pediátrica, en un horizonte de 11 meses, cotejado con los datos del conjunto de *test*, fue el **Modelo Neuronal Extreme Learning Machines (EML)**. *La visualización simultanea del ajuste con los datos del conjunto de test*, para los cuatro modelos, se obtuvo mediante el siguiente código:

```
## Comparación de pronósticos de algoritmos, sobre datos de "test" ##  
grid.arrange(g2, g3,  
             g4, g5,  
             nrow=3)
```

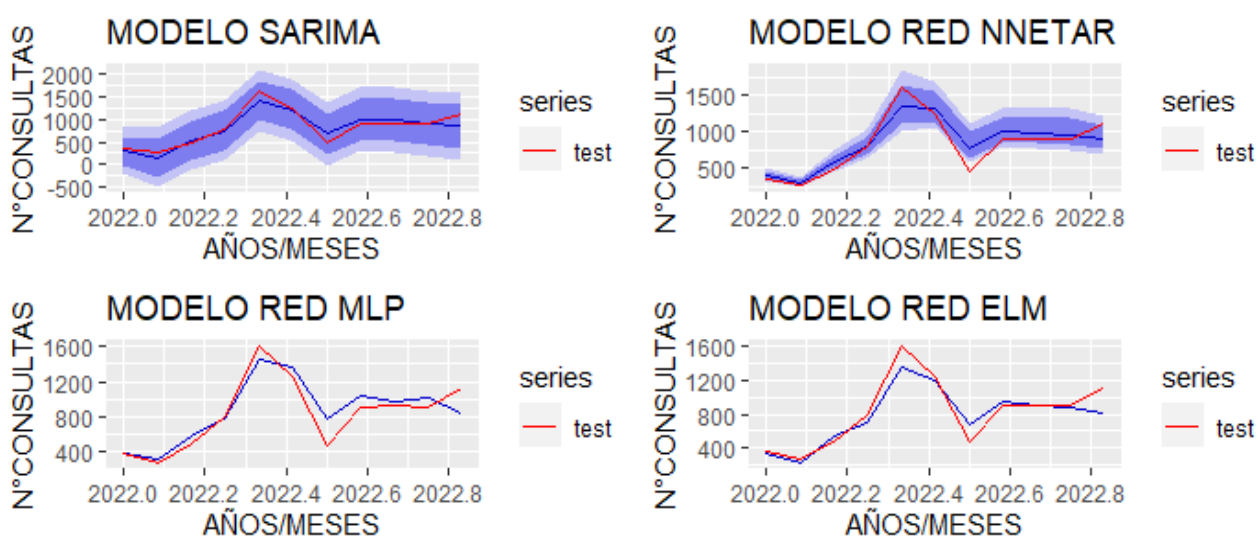


Gráfico 23: Comparativa de gráfica de precisión de pronósticos de algoritmos, sobre conjunto de test.

Luego, este algoritmo será implementado en la Dirección de Servicio de Salud Chiloé, para proporcionar estimaciones mensuales de demanda respiratoria infantil en los hospitales de la Red Asistencial.

4. DISCUSIÓN.

Los modelos predictivos elaborados en este Trabajo de Fin de Master, puede ser usados para mejorar el funcionamiento general de las Unidades de Emergencia Hospitalaria (UEH) de la red asistencial del Servicio de Salud Chiloé, ya que permite actuar proactivamente, preparando tanto recursos humanos (médicos, enfermeras, técnicos) como físicos (BOX, salas de espera, materiales o camas), para los periodos en que exista una sobredemanda en la atención.

Estos sistemas se pueden desplegar centralizadamente desde el *Data Warehouse*, del Departamento de Inteligencia Sanitaria del Servicio de Salud Chiloé, y actualizarse automáticamente, cada mes mediante proceso de ETL, con los datos de urgencia disponibles (de manera liberada) en la página del Web del DEIS - Minsal, para ser usado por los Directivos del Servicio de Salud y los Hospitales en la toma de decisiones.

Finalmente es importante dejar en claro, que tomar la decisión de Anular un periodo aproximado de 2 años de información (22 meses, equivalente a 22 registros de la serie temporal), producto de las anomalías causadas por la Pandemia de Covid-19 en la provincia de Chiloé, y luego estimar este mismo periodo, mediante el método de Imputación de Kalman con modelos ARIMA de espacio de estados (usando la función *auto.arima*), representa una innovación que vendría a dar una salida a este problema. Sin embargo, también representa un riesgo importante a la hora de utilizar esta información como histórica, y en consecuencia como soporte de predicciones, fundamentalmente usando modelos ARIMA, ya que los modelos de estados que utiliza el Filtro de Kalman en este trabajo son de la misma naturaleza.

5. REFERENCIAS/BIBLIOGRAFÍA.

- a) **LOREN, J. (2021).** Modelo de clasificación para la predicción de la rotación de clientes en una operadora telefónica. Trabajo Fin de Máster, Universidad de Nebrija, Entidad Colaboradora: Máxima Formación, Madrid, España. 32p. https://www.maximaformacion.es/wp-content/uploads/2021/12/20210726_tfm_Jose_Luis_Loren.pdf
- b) **CONTRERAS, S. (2018).** Modelo Predictivo de la Demanda de Atención de Urgencia en Chile: una Aplicación de Google Trends, Tesis para optar al Grado de Magister en Informática Médica. Facultad de Medicina, Escuela de Postgrado. Universidad de Chile. Santiago, Chile. 57p. <https://repositorio.uchile.cl/handle/2250/174096>
- c) **CÁRDENAS M, CLAUDIO, SOVIER V, CARLOS, PÉREZ R, ULDA, & GONZÁLEZ A, C. A. SERGIO. (2014).** Consultas de urgencia general y por causa respiratoria en la Red de establecimientos del Sistema Nacional de Servicios de Salud (SNSS): un modelo predictivo en el Servicio de Salud de Chiloé. Revista chilena de enfermedades respiratorias, 30(3), 133-141. <https://dx.doi.org/10.4067/S0717-73482014000300002>
- d) **MAXIMAFORMACION, (2020).** Material del Master en Machine Learning con R Software
- e) **Modelo de Autorregresión de Red Neuronal o NNAR:** <https://otexts.com/fpp2/nnetar.html>
- f) **Modelos ARIMA:** <https://otexts.com/fpp2/arima.html>
- g) **Consulta Respiratoria Pediátrica:** <https://www.clinicaalemana.cl/articulos/detalle/2020/nueva-consulta-respiratoria-pediatica>
- h) **Bases de Datos de Urgencia DEIS-Minsal:** <https://deis.minsal.cl/#datosabiertos>
- i) **Hitos Covid -19 en Chile:** <https://observatorio.medicina.uc.cl/primer-caso-de-covid-19-en/>
- j) **Gestión de datos perdidos:** <https://github.com/manuparra/seriestemporales#manejo-de-datos-perdidos-en-series-temporales>
- k) **Gestión de datos perdidos:** <https://www.datasource.ai/es/data-science-articles/todo-sobre-el-manejo-de-datos-faltantes>
- l) **Minería de Datos:** <https://www.sngular.com/es/data-science-crisp-dm-metodologia/>
- m) **Paquete nnfor:** <https://kourentzes.com/forecasting/2019/01/16/tutorial-for-the-nnfor-r-package/>
- n) **Redes Neuronales en Series Temporales:** <https://www.bruceing.ca/post/neural-network-time-series-models/>

- o) **Series Temporales:** <https://www.pricing.cl/conocimiento/estacionalidad/>
- p) **Series Temporales:** <https://finanzaszone.com/analisis-y-prediccion-de-series-temporales-con-r-ii-estacionariedad-y-raices-unitarias/>
- q) **ETL:** https://www.sas.com/es_cl/insights/data-management/.
- r) **Medidas de Precisión:** https://docs.oracle.com/cloud/help/es/pbcs_common/PFUSU/
- s) **Medidas de Precisión:** <https://support.numxl.com/hc/es/articles/215969423-MAE-Error-medio-absoluto> .
- t) **Medidas de Precisión:** <https://www.datup.ai/logistica/medidas-de-error-en-pronosticos-de-demanda/>.

6. ANEXOS.

6.1 ANEXO A: CÓDIGO R, PARA LA RESOLUCIÓN DEL PROCESO DE ANÁLISIS.

Librerías R, utilizadas en TFM.

```
## Librerías utilizadas en TFM ##  
  
library(readxl)      ## Librería para importación de datos en archivo Excel #  
#  
library(forecast)   ## Librería necesaria para modelar series temporales ##  
library(ggplot2)    ## Librería necesaria para realizar visualizaciones ##  
library(gridExtra)  ## Librería necesaria para organizar múltiples gráficos  
en cuadrícula ##  
library(lubridate)  ## Librería necesaria para trabajar con fechas en la ser  
ie temporal ##  
library(changepoint) ## Librería necesaria para detectar puntos de cambio en  
la serie temporal ##  
library(strucchange) ## Librería necesaria para identificar quiebres estructur  
ales en la serie temporal ##  
library(imputeTS)   ## Librería necesaria para imputación de datos faltantes  
(NULL) ##  
library(tseries)    ## Librería para pruebas de estacionariedad y autocorrel  
ación en series temporales.  
library(tsutils)    ## Librería para pruebas de tendencia y estacionalidad e  
n series temporales.  
library(forecast)   ## Librería necesaria para modelar series temporales ##  
library(lmttest)    ## Librería para estimar coeficiente e intervalos de con  
fianza en modelos ARIMA ##  
library(nnfor)      ## Librería de redes neuronales, para series temporales  
##  
library(kableExtra) ## Librería para crear tablas comunes, complejas y manip  
ular estilos de tablas.
```

Importación y visualización de serie temporal.

```
## Importación de datos para análisis ##
X2011_2022_CHILE_mes_CHLO <- read_excel("2011_2022_CHILE_mes_CHLO.xlsx",
  col_types = c("numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric"))
BBDD_URGE_PED <- X2011_2022_CHILE_mes_CHLO

## Transformación de la variable "Urgencia_Pediatrica" en un objeto de serie
temporal ##
TS <- ts(BBDD_URGE_PED$Urgencia_Pediatrica,
  frequency=12,
  start=c(2011,1))

## Graficación de serie temporal ##
GF1 <- autoplot(TS, main= " Serie 2011-2022, Urgencia Respiratoria Infantil,
Hospitales SS Chiloé",
  xlab = "MESES/AÑOS",
  ylab = "N° CONSULTAS")

fit<- decompose(TS, type="additive") # Descomposicion de serie temporal ##

## Visualización de descomposición de serie temporal en estacionalidad y tend
encia ##

GF2 <- autoplot(TS, series="Serie Original", xlab = "MES/AÑO", ylab="N°CONSUL
TAS") +
  ggtitle("Descomposición Aditiva Clásica Urgencia Respiratoria Infantil
")+
  autolayer(trendcycle(fit), series="Tendencia") +
  autolayer(seasadj(fit), series="Ajuste estacional") +
  scale_colour_manual(values=c("gray","blue","red"),
  breaks=c("Serie Original","Ajuste estacional","Tendencia"))

## Organización de gráficos de la Serie Original y Descompuesta en una cuadri
cula 2x1 ##
grid.arrange(GF1, GF2,nrow=2)
```

Evaluación de puntos de cambio en la serie temporal.

```
## Identifica los puntos de corte (Numero de registro de la serie) ##
pts.media <- cpt.mean(TS)

## Grafica los puntos de corte (Numero de registro de la serie) ##
plot(cpt.mean(TS),
     penalty= "BIC",
     method= "AMOC",
     test.stat="Normal",
     main="Cambios Significativos de la Media en la Serie Temporal",
     xlab="MES/AÑO",
     ylab="N° DE CONSULTAS")

## valores medios en los tramos entre los puntos de corte ##
param.est(pts.media)

## valores de las fechas de los puntos de corte en formato decimal ##
pts.media_b <- cpts.ts(pts.media)

## valores de las fechas de los puntos de corte en formato calendario ##
date_decimal(pts.media_b)

## Identifica los puntos de corte (Numero de registro de la serie) ##
cpt.var(TS, penalty="BIC",
        method="AMOC",
        test.stat="Normal",
        class=TRUE)

## Grafica los puntos de corte (Numero de registro de la serie) ##
plot(cpt.var(TS, penalty="BIC",
            method="AMOC",
            test.stat="Normal",
            class=TRUE),
     main="Cambios Significativos de la Varianza en la Serie Temporal",
     xlab="AÑOS/MESES",
     ylab="N°CONSULTAS")
```

Evaluación de quiebres estructurales en la serie temporal.

```
## Cálculo de puntos de quiebre en relaciones de regresión. Dada una cantidad
de rupturas, la función calcula los puntos de ruptura óptimos. ##
(break_point <- breakpoints(TS ~ 1, hpc="none"))

## Graficación de puntos de quiebre estructural ##
plot(break_point)

## Descripción de resultados de puntos de quiebre estructural ##
summary(break_point)

## Elaboración de gráfico de serie temporal ##
plot(TS, main="Rupturas Estructurales en la Serie Temporal",
      xlab="AÑOS/ MESES",
      ylab= "N° CONSULTAS")

## Ajuste de líneas verticales de rupturas estructurales ##
fitted.ts <- fitted(break_point, breaks =2)

## Ajuste de líneas horizontales de rupturas estructurales ##
lines(fitted.ts, col =3)

## Ajuste de líneas intervalos de confianza de rupturas estructurales ##
lines(confint(break_point, breaks =2))

## Indicadores de rupturas estructurales y sus intervalos de confianza ##
confint(break_point, breaks =2)
```

Propuesta metodológica para tratamiento de datos Outliers durante el Periodo de Pandemia Covid-19 en Chiloé.

```
## Anulación de datos observados en periodo Covid-19, para ser considerados como información faltante ##
TS[111:132]<-NA ## marzo 2020 - diciembre 2021 ##

## Visualización de datos faltantes (Anulados) durante el periodo de pandemia Covid-19 ##
IMP1 <- autoplot(TS, main="Serie Temporal, con Datos Nulos Durante Periodo de Pandemia Covid-19",
  xlab="AÑOS / MESES",
  ylab = "N°CONSULTAS")

## Imputación de datos faltantes mediante el Método de Kalman ##
Clean_TS_Kalman <- round(na_kalman(TS,
  model="auto.arima",
  smooth=T,
  stationary=T), digits = 0)

## Visualización de Serie Temporal Corregida (Clean_TS_Kalman), Mediante Imputación por Método de Kalman ##
IMP2 <- autoplot(Clean_TS_Kalman, main="Serie Temporal Corregida, Mediante Método de Imputación de Kalman",
  xlab= "AÑOS / MESES",
  ylab="N° CONSULTAS")

## Declaracion de datos Nulos durante el periodo de (pandemia Covid-19) vs. Serie corregida mediante imputación de Kalman ##
grid.arrange(IMP1,
  IMP2,
  nrow=2)

## Visualización de descomposicion aditiva de la serie temporal corregida ##
de.Clean_TS_Kalman = decompose(Clean_TS_Kalman)
plot(de.Clean_TS_Kalman)
```

Verificación de supuestos estadísticos de la serie temporal corregida.

```
ndiffs(Clean_TS_Kalman, test = "kpss")    ## Verificación diferencia regular

nsdiffs(Clean_TS_Kalman, test = "seas")  ## Verificación diferencia estaciona
l

## Dickey-Fuller Aumentado (Augmented Dickey-Fuller Test (ADF) hipótesis nula
es precisamente la presencia de una raíz unitaria en el proceso generador de
datos de la serie analizada ##

adf.test(Clean_TS_Kalman)

##Phillips-Perron (PP), tiene por hipótesis nula que los datos No son estacio
narios. Valores pequeños de  $p$  ( $< 0.05$ ) sugieren que No se requiere diferencia
ción##

pp.test(Clean_TS_Kalman, alternative="stationary")

##La prueba de Kwiatkowski-Phillips-Schmidt-Shin, tiene por hipótesis nula qu
e los datos son estacionarios. Valores pequeños de  $p$  ( $< 0.05$ ) sugieren que se
requiere diferencia. La prueba Kwiatkowski-Phillips-Schmidt-Shin (KPSS) deter
mina si una serie de tiempo es estacionaria alrededor de una tendencia media
o lineal, o si no es estacionaria debido a una raíz unitaria. Una serie tempo
ral estacionaria es aquella en la que las propiedades estadísticas, como la m
edia y la varianza, son constantes a lo largo del tiempo. ##

kpss.test(Clean_TS_Kalman)

## La prueba de Ljung-Box La prueba  $Q$  de Ljung-Box considera la hipótesis nul
a de que las autocorrelaciones de hasta un desfase son iguales a cero ##

Box.test(Clean_TS_Kalman, type="Ljung-Box")

## Prueba no paramétrica de Cox-Stuart nos permite evaluar si existe una tend
encia (monótona) significativa en los datos ##

coxstuart(Clean_TS_Kalman, type = "trend")

## Esta función utiliza la prueba de Friedman, que proporciona los índices es
tacionales estimados ##

seas<-seasplot(Clean_TS_Kalman)

str(seas)
```

Verificación de supuestos estadísticos de la serie temporal corregida diferenciada en componente regular y estacional.

```
## Descomposición aditiva de la serie temporal diferenciada regular y estacionalmente ##
de.Clean_TS_Kalman = decompose(diff(diff(Clean_TS_Kalman,lag = 12),lag = 1))
plot(de.Clean_TS_Kalman)

## Visualización de la serie temporal junto con su acf y su pacf de la serie temporal diferenciada regular y estacionalmente ##
ggtsdisplay(diff(diff(Clean_TS_Kalman,lag = 12),lag = 1))

## Serie temporal diferenciada regular y estacionalmente ##
diffClean_TS_Kalman <- diff(diff(Clean_TS_Kalman,lag = 12),lag = 1)

## Selección de un modelo ARIMA candidato ##
ggAcf(diffClean_TS_Kalman, lag.max=20, plot=FALSE) ## Función de Autocorrelación Simple.

ggPacf(diffClean_TS_Kalman, lag.max=20, plot=FALSE) ## Función de Autocorrelación Parcial.

ndiffs(diff(diff(Clean_TS_Kalman,lag = 12),lag = 1))      ## Verificación diferencia regular
nsdiffs(diff(diff(Clean_TS_Kalman,lag = 12),lag = 1))     ## Verificación diferencia regular

## Dickey-Fuller Aumentado (Augmented Dickey-Fuller Test (ADF) hipótesis nula es precisamente la presencia de una raíz unitaria en el proceso generador de datos de la serie analizada ##
adf.test(diffClean_TS_Kalman)

##Phillips-Perron (PP), tiene por hipótesis nula que los datos No son estacionarios. Valores pequeños de  $p$  ( $< 0.05$ ) sugieren que No se requiere diferenciación##
pp.test(diffClean_TS_Kalman, alternative="stationary")

##La prueba de Kwiatkowski-Phillips-Schmidt-Shin, tiene por hipótesis nula que e los datos son estacionarios. Valores pequeños de  $p$  ( $< 0.05$ ) sugieren que se requiere diferenciación##
kpss.test(diffClean_TS_Kalman)
```

```

## La prueba de Ljung-Box La prueba Q de Ljung-Box considera la hipótesis nul
a de que las autocorrelaciones de hasta un desfase d son iguales a cero ##
Box.test(diffClean_TS_Kalman, type="Ljung-Box")

## Evaluacion de tendencia en Serie Temporal Diferenciada ##
coxstuart(diffClean_TS_Kalman, type = "trend")

## Evaluación de estacionalidad en Serie Temporal Diferenciada ##
seas<-seasplot(diffClean_TS_Kalman)

```

Partición de muestra de datos de la serie temporal.

```

## Partición de muestra de datos de la Series temporal ##

train <- window(Clean_TS_Kalman,
                end = c(2021,12)) ## Muestra de entrenamiento ##

test <- window(Clean_TS_Kalman,
               start = c(2022,1)) ## Muestra de test o prueba ##

```

Modelo ARIMA, mediante método de modelado automático.

```
## Estimación de Modelo ARIMA, mediante función "auto.arima"##
fit_autoarima <-auto.arima(train,
                           trace = T,
                           ic = "bic")
## Estimación de parámetros del modelo e intervalos de confianza ##
coefstest(fit_autoarima)
confint(fit_autoarima)

##Indicadores de Precisión para SARIMA (m1), sobre Series de "train" y "test"
##
rend_autoarima <- accuracy(forecast(fit_autoarima), test)
rend_autoarima

##Extracción de Indicadores de Precisión, sobre Serie de "test", AUTOARIMA(m1)
##
m1 <- data.frame(rend_autoarima[2,c(2,3,5)])
colnames(m1) <- "AUTOARIMA"

## Visualización Serie de "test" vs. Serie pronosticada sobre la misma ##
g1=autoplot(forecast(fit_autoarima, h = 11),
            include = F,
            main="MODELO AUTOARIMA",
            xlab="AÑOS/MESES",
            ylab="N° DE CONSULTAS") +
  autolayer(test,
            series= NULL,
            colour = T) + scale_colour_manual(values="red")
g1
## Pronósticos del modelo AUTOARIMA, y sus intervalos de confianza ##
autoplot(forecast(fit_autoarima,
                 h =11))
## Diagnóstico de los residuos del modelo AUTOARIMA (1,0,1)(0,1,1)[12] ##
checkresiduals(fit_autoarima)
```

Modelo ARIMA, mediante análisis de correlograma ACF y PACF.

```
## Estimación de Modelo SARIMA (2,1,1)(1,1,0)[12], sobre Serie de "train" ##
fit_sarima <- arima(train,
                    order = c(2,1,1),
                    seasonal = list(order = c(1,1,0),
                                     period = 12))

## Estimación de parámetros del modelo y sus correspondientes intervalos de c
onfianza ##
coeftest(fit_sarima)
confint(fit_sarima)

##Indicadores de Precisión para SARIMA (m2), sobre Series de "train" y "test"
##
rend_arima <- accuracy(forecast(fit_sarima), test)
rend_arima

## Extracción de Indicadores de Precisión, sobre Serie de "test", SARIMA(m2)
##
m2 <- data.frame(rend_arima[2,c(2,3,5)])
colnames(m2) <- "SARIMA"

## Visualización Serie de "test" vs. Serie pronosticada sobre la misma ##
g2=autoplot(forecast(fit_sarima,
                    h = 11),
            include = F,
            main="MODELO SARIMA",
            xlab="AÑOS/MESES",
            ylab="N°CONSULTAS") +
  autolayer(test,
            series = "test",
            colour = T) + scale_colour_manual(values="red")
g2

## Pronósticos del modelo SARIMA, y sus intervalos de confianza ##
autoplot(forecast(fit_sarima,
                  h =11))
```

Modelo de Autorregresión de Redes Neuronales (NNAR).

```
set.seed(123)## Semilla aleatoria ##

## Estimación de Modelo de Autorregresión de redes neuronales (NNAR)##
(fit_nnetar <- nnetar(train,
                      repeats = 100,
                      lambda = 0))

## Indicadores de Precisión para Autorregresión de redes neuronales (m3), sobre Series de "train" y "test" ##
rend_nnetar <- accuracy(forecast(fit_nnetar), test)
rend_nnetar

## Extracción de Indicadores de Precisión, sobre Serie de "test", Autorregresión de redes neuronales (m3) ##
m3 <- data.frame(rend_nnetar[2,c(2,3,5)])
colnames(m3) <- "NNETAR"

## Visualización Serie de "test" vs. Serie pronosticada sobre la misma ##
g3=autoplot(forecast(fit_nnetar,
                    h = 11,
                    PI = T),
            include = F,
            main="MODELO RED NNETAR",
            xlab="AÑOS/MESES",
            ylab="N°CONSULTAS") +
  autolayer(test,
            series = "test",
            colour = T) + scale_colour_manual(values="red")
g3

## Pronósticos del modelo de Autorregresión de redes neuronales, y sus intervalos de confianza ##
autoplot(forecast(fit_nnetar,
                  PI = TRUE,
                  h =11))
```

Modelo de Red Neuronal Multilayer Perceptron (MLP).

```
set.seed(123)  ## semilla aleatoria

## Estimación de Modelo de Red Neuronal Multilayer Perceptron sobre Serie de
"train"##
fit_mlp <- mlp(ts(train,
                  frequency = 12,
                  end = c(2021,12)),
              reps = 100,
              difforder = c(1,12),
              comb="median")

## Grafo de estructura de la red neuronal Multilayer Perceptron ##
plot(fit_mlp)

## Pronósticos de cada uno de los modelos estimados, en cada repetición, y la
"mediana" de estos ##
frc <- forecast(fit_mlp, h = 11)
plot(frc)

## Indicadores de Precisión para Red Neuronal Multilayer Perceptron (m4), sob
re Series de "train" y "test" ##
rend_mlp <- accuracy(forecast(fit_mlp),
                    ts(test,
                        frequency = 12,
                        start = c(2022,1)))

rend_mlp

## Extracción de Indicadores de Precisión, sobre Serie de "test", Red Multila
yer Perceptron (m4) ##
m4 <- data.frame(rend_mlp[2,c(2,3,5)])
colnames(m4) <- "MLP"

## Visualización Serie de "test" vs. Serie pronosticada sobre la misma ##
g4 = autoplot(forecast(fit_mlp, h = 11),
              include = F,
              main="MODELO RED MLP",
```

```
xlab="AÑOS/MESES",
ylab="N°CONSULTAS") +
autolayer(ts(test, frequency = 12,
start = c(2022,1)),
series = "test",
colour = T) +
scale_colour_manual(values="red")
```

g4

Modelo de Red Neuronal Extreme Learning Machines (EML).

```
set.seed(123) ## semilla aleatoria

## Estimación de Modelo de "Red Neuronal Extreme Learning Machines (EML), sobre Serie "train"##
fit_elm <- elm(ts(train,
                 frequency = 12,
                 end = c(2021,12)),
              type="step",
              difforder = c(1,12),
              reps = 100,
              comb = "median")

## Grafo de estructura de la Red Neuronal Extreme Learning Machines ##
plot(fit_elm)

## Pronósticos de cada uno de los modelos estimados, en cada repetición, y la "mediana" de estos ##
frc <- forecast(fit_elm, h = 11 )
plot(frc)

## Indicadores de Precisión para "Red Neuronal Extreme Learning Machines (m5)", sobre las Series "train" y "test" ##
rend_elm <- accuracy(forecast(fit_elm),
                    ts(test,
                        frequency = 12,
                        start = c(2022,1)))

rend_elm

## Extracción de Indicadores de Precisión, sobre Serie de "test", Red Extreme Learning Machines (m5) ##
m5 <- data.frame(rend_elm[2,c(2,3,5)])
colnames(m5) <- "ELM"

## Visualización Serie de "test" vs. Serie pronosticada sobre la misma ##
g5 = autoplot(forecast(fit_elm, h = 11),
              include = F,
              main="MODELO RED ELM",
```

```
    xlab="AÑOS/MESES",
    ylab="N°CONSULTAS") +
autolayer(ts(test,
             frequency = 12,
             start = c(2022,1)),
          series = "test",
          colour = T) +
scale_colour_manual(values="red")
```

g5

Indicadores de precisión de pronósticos de algoritmos.

```
## Indicadores de precisión de algoritmos utilizados sobre datos de "test" ##
tabl <- cbind.data.frame(m2,
                        m3, m4, m5)

kable(head(round((head(tabl)),2), 16),
       booktabs = TRUE,
       caption = "COMPARACION DE INDICADORES DE PRECISION DE ALGORITMOS, SOBRE
SERIE DE TEST", ) %>%
kable_classic()
```

Visualizaciones de pronósticos de algoritmos.

```
## Comparación de pronósticos de algoritmos utilizados, sobre datos de "test"
##
grid.arrange(g2, g3,
            g4, g5,
            nrow=3)
```