

Grado en Estadística

Título: Análisis de la Resistencia Bacteriana:
Implementación de una Plataforma Shiny para la
Valoración del Uso de Medicamentos

Autor: Oscar Arroyo Luque

Director: Josep Anton Sànchez Espigares

Departamento: Dept. Estadística e Investigación Operativa

Convocatoria: 2023 - 2024



UNIVERSITAT DE
BARCELONA



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat de Matemàtiques i Estadística

Abstract

Resumen

Actualmente nos enfrentamos a uno de los mayores desafíos de la medicina moderna: la creciente resistencia bacteriana provocada por el uso de antibióticos.

En este trabajo abordamos esta problemática mediante el desarrollo de una interfaz interactiva Shiny, que permite a un experto en estadística realizar un estudio sobre las resistencias bacterianas mediante el ajuste de modelos MARS.

Asimismo, a través del uso de la aplicación, hemos llevado a cabo un estudio sobre la bacteria *Escherichia coli*. En este análisis, hemos determinado que el uso de ampicilina promueve el fenómeno de las resistencias.

Palabras clave: Shiny, Interfaz Interactiva, Resistencias Bacterianas, Antibiótico, Series Temporales, Modelos MARS.

Abstract

Currently we are facing one of the greatest challenges of modern medicine: the growing bacterial resistance caused by the use of antibiotics.

In this project, we address this issue through the development of an interactive Shiny interface, which allows a statistics expert to conduct a study on bacterial resistances by fitting MARS models.

Additionally, through the use of the application, we have conducted a study on the bacterium *Escherichia coli*. In this analysis, we have determined that the use of ampicillin promotes the phenomenon of resistances.

Keywords: Shiny, Interactive Interface, Bacterial Resistances, Antibiotic, Time Series, MARS Models.

Clasificación AMS

- **68U35** Computing methodologies for information systems (hypertext navigation, interfaces, decision support, etc.)
- **62M10** Time series, auto-correlation, regression, etc.
- **62G08** Nonparametric regression and quantile regression
- **62P10** Applications of statistics to biology and medical sciences; meta analysis

Agradecimientos

Agradezco al profesorado de la carrera, y en especial al director de este trabajo, por sus valiosas enseñanzas y por transmitirme la pasión por la estadística.

También quiero expresar mi profunda gratitud a mis amigos y familia por su apoyo y amor incondicional a lo largo de este viaje.

Índice general

Abstract	I
Clasificación AMS	II
Agradecimientos	III
Listado de Figuras	VI
1. Introducción	1
2. Metodología	4
2.1. Análisis de series temporales	4
2.1.1. Las series temporales	4
2.1.2. Los procesos estocásticos	5
2.1.3. Funciones de autocorrelación y Correlogramas	5
2.2. Modelización	7
2.2.1. Modelo ARMAX	7
2.2.2. Modelo MARS	8
2.2.3. Construcción de modelos	9
2.3. Estadísticos y gráficos	9
2.3.1. Estadísticas de resumen de modelos lineales	9
2.3.2. Gráficos de validación	10
2.4. Recursos informáticos	13
2.4.1. Desarrollo y programación de Shiny en RStudio	13
2.4.2. Desarrollo y programación de modelos en RStudio	15
2.4.3. Otras librerías	15
2.5. Definiciones relacionadas con el fenómeno de las resistencias bacterianas	15
2.5.1. Fenómeno de resistencia bacteriana	16
2.5.2. Definición de variables AMC y AMR	16

ÍNDICE GENERAL	v
3. La base de datos	17
4. Shiny APP: M.R.B.A.	18
5. Estudio estadístico y demostraciones de uso	20
5.1. Modelización de las resistencias Escherichia coli ante la Amoxicilina	20
5.1.1. Carga de ficheros	20
5.1.2. Exploración descriptiva de los datos	21
5.1.3. Selección de variables	25
5.1.4. Exploración preliminar de modelos	26
5.1.5. Selección de variables de AMC	32
5.1.6. Ajuste del modelo ARMAX	33
5.1.7. Modelo MARS	36
5.1.8. Conclusiones modelo MARS	41
6. Conclusiones	43
Bibliografía	45
A. Guía de Uso	47
B. Gráficos de los modelos ajustados	60
C. Código R: Shiny APP M.R.B.A.	63

Índice de figuras

2.1. Gráficos de validación de modelos: Residuos vs Valores Ajustados	11
2.2. Gráficos de validación de modelos: Normal Q-Q	11
2.3. Gráficos de validación de modelos: <i>Scale-Location</i>	12
2.4. Gráficos de validación de modelos: Residuos vs Apalancamiento	12
2.5. Gráficos de validación de modelos: ACF y PACF	13
5.1. Demostración de uso: Carga de ficheros	21
5.2. Demostración de uso: Tabla de descriptivos	21
5.3. Demostración de uso: Gráfico de barras apiladas	23
5.4. Demostración de uso: Gráfico de serie de tiempo	23
5.5. Demostración de uso: Gráfico de serie de tiempo	24
5.6. Demostración de uso: Gráfico de serie de tiempo: %Resistencias	24
5.7. Demostración de uso: Gráfico de serie de tiempo: TDI	25
5.8. Demostración de uso: Selector de variables	25
5.9. Demostración de uso: Gráfico de series temporales del modelo	26
5.10. Demostración de uso: Selector de retardos a partir de cifras	26
5.11. Demostración de uso: Ajuste del modelo lineal completo en la exploración preliminar	27
5.12. Demostración de uso: Estadísticos de ajuste del modelo lineal completo exploración preliminar	28
5.13. Demostración de uso: Gráficos de validación ajuste del modelo lineal completo exploración preliminar	29
5.14. Demostración de uso: Gráfico ACF & PACF del ajuste del modelo lineal completo exploración preliminar	29
5.15. Demostración de uso: Ajuste del modelo lineal simplificado exploración preliminar	30
5.16. Demostración de uso: Gráficos de validación ajuste del modelo lineal simplificado exploración preliminar	31

5.17. Demostración de uso: Gráficos ACF & PACF del modelo lineal simplificado exploración preliminar	32
5.18. Demostración de uso: Selección de variables AMC	33
5.19. Demostración de uso: Ajuste del modelo ARMAX	33
5.20. Demostración de uso: Ajuste modelo lineal ARMAX	35
5.21. Demostración de uso: Ajuste modelo MARS: Los datos	36
5.22. Demostración de uso: Ajuste modelo MARS: <i>Forward Pass</i>	36
5.23. Demostración de uso: Ajuste modelo MARS: Podado	37
5.24. Demostración de uso: Gráfico de residuos modelo MARS	38
5.25. Demostración de uso: <i>Normal Q-Q Plot</i> de los residuos del modelo MARS . . .	39
5.26. Demostración de uso: Gráficos ACF & PACF del modelo MARS	39
5.27. Demostración de uso: Gráfico de recreación de la serie modelo MARS	40
5.28. Demostración de uso: Gráfico del efecto de las variables explicativas sobre la respuesta	41
A.1. Guía de uso: Carga de ficheros	48
A.2. Guía de uso: Seleccionadores	49
A.3. Guía de uso: Summary datos iniciales	49
A.4. Guía de uso: Gráficos interactivos	50
A.5. Guía de uso: Seleccionadores	50
A.6. Guía de uso: Gráficos interactivos	51
A.7. Guía de uso: Selector de variables manual	52
A.8. Guía de uso: Seleccionador a partir de fichero	53
A.9. Guía de uso: Gráfico de series temporales	53
A.10. Guía de uso: Selector de retardos a partir de cifras	54
A.11. Guía de uso: Selector de retardos manual	54
A.12. Guía de uso: Ajuste del Modelo Lineal completo preliminar	55
A.13. Guía de uso: Ajuste del Modelo Lineal simplificado preliminar	55
A.14. Guía de uso: Validación modelos preliminares	56
A.15. Guía de uso: Seleccionador de variables AMC	56
A.16. Guía de uso: Selector de parámetros y ajuste del modelo ARMAX	57
A.17. Guía de uso: Ajuste del modelo lineal completo y simplificado	57
A.18. Guía de uso: Validación modelos ARMAX	58
A.19. Guía de uso: Output del modelo MARS	58
A.20. Guía de uso: Validación e Interpretación modelo MARS	59
A.21. Guía de uso: Botón de descarga	59

B.1. Demostración de uso: Ajuste del modelo lineal simple ARMAX	60
B.2. Demostración de uso: Gráficos de validación del modelo lineal completo ARMAX	61
B.3. Demostración de uso: Gráficos ACF & PACF del modelo lineal completo ARMAX	61
B.4. Demostración de uso: Gráficos de validación del modelo lineal simple ARMAX	61
B.5. Demostración de uso: Gráficos ACF & PACF del modelo lineal simplificado ARMAX	62

Capítulo 1

Introducción

Actualmente nos enfrentamos a uno de los mayores retos de la medicina moderna: la creciente resistencia bacteriana causada por el uso de antibióticos.

Utilizar agentes antibacterianos como tratamiento ante infecciones, particularmente en entornos sanitarios y análogos, ocasiona un impacto en el ecosistema microbiano presente en estos lugares, resultando en el desarrollo de cepas bacterianas que muestran características genéticas capaces de obviar los tratamientos médicos ([Organización Mundial de la Salud, 2019](#)).

Este ecosistema se ve perturbado dado que las bacterias que logran sobrevivir y luego consiguen propagarse dan origen a progenies con características genéticas que les permiten evadir estos tratamientos antimicrobianos ([Organización Mundial de la Salud, 2019](#)).

Debido a la evolución y la recurrencia de este fenómeno, se plantea un futuro en el que el uso de medicamentos para la intervención terapéutica en infecciones bacterianas o semejantes podría volverse ineficaz, lo que desencadena un conflicto entre la evolución natural bacteriana y las nuevas generaciones de medicamentos. Este escenario pone en riesgo la salud de la población, tanto de las personas bajo cuidado hospitalario, como de cualquier otra persona expuesta al contagiado ([Laxminarayan et al., 2013](#)).

El Informe del Sistema Mundial de Vigilancia de la Resistencia y el Uso de los Antimicrobianos (GLASS) de la [Organización Mundial de la Salud \(2019\)](#), señala una preocupación crítica. Según este informe, más del 50 % de las bacterias responsables de infecciones de alta gravedad en centros médicos presentan niveles alarmantes de resistencia a los fármacos. Un caso estudiado es el de la bacteria *Klebsiella pneumoniae*, agente causal de infecciones urinarias, neumonías, sepsis, infecciones de tejidos blandos y heridas quirúrgicas. En este estudio se observa que el microorganismo ha desarrollado resistencia a un grupo de agentes bactericidas

considerados como último recurso, lo que dificulta su tratamiento y puede derivar en desenlaces fatales.

En el mismo estudio encontramos que, en relación con los consumos de fármacos antibacterianos, solo el 65 % de los países participantes en el estudio lograron el objetivo de la OMS, el cual propone que al menos el 60 % de los antibióticos suministrados sean del grupo “Acceso”, es decir, que sean eficaces y aporten un menor riesgo de generar inmunotolerancias bacterianas.

Ante tal panorama, la OMS enfatiza la necesidad urgente de mejorar la vigilancia y la calidad de los datos sobre las tolerancias a los antimicrobianos para así ser capaces de formular políticas efectivas y abordar esta creciente amenaza a la salud pública mundial.

Este trabajo de fin de grado nació del interés de un hospital por monitorizar este fenómeno, y así ejercer un uso responsable y sostenible sobre la administración de medicamentos. La historia del proyecto comenzó cuando el centro sanitario contactó con el tutor de este trabajo para realizar un seguimiento de las resistencias microbacterianas, mediante la modelización de las tolerancias y consumos observados a lo largo del tiempo.

El análisis realizado por el tutor de este trabajo empezó con una exploración preliminar de variables, valiéndose de retardos tanto de resistencias como de consumos para el ajuste de modelos. Tras investigar el comportamiento de estos componentes, se seleccionaron los más relevantes para después ser examinados en un modelo ARMAX. Mediante este procedimiento, se puede determinar el número de retardos a aplicar de la variable de respuesta. Posteriormente, conociendo todas las variables de interés, se introdujeron y ejecutaron en un modelo MARS. Tras la validación del modelo, es posible graficar cada variable explicativa en relación con la variable respuesta de forma independiente. Así, a través de la interpretación gráfica, se puede determinar cómo impacta el consumo de agentes antibacterianos en las inmunotolerancias observadas.

El centro de atención médica destacó la necesidad de reproducir este estudio de forma periódica, para así observar la evolución de las bacterias presentes en su ecosistema y poder tomar nuevas decisiones en lo que respecta a la administración de medicamentos. Sin embargo, para replicar el estudio es necesario contar con conocimientos de estadística y programación. Con el propósito de facilitar la obtención de resultados, se planteó la creación de una herramienta capaz de agilizar el proceso, lo que resultó en la propuesta de este proyecto.

Por lo tanto, en el presente trabajo nos hemos planteado como **objetivo principal** la creación de una interfaz interactiva en el entorno Rstudio, que facilite a un experto en estadística realizar

el análisis previamente mencionado. Esto implica que vamos a proporcionar una herramienta flexible que permitirá la construcción, ajuste y validación de una batería de modelos estadísticos según las necesidades del usuario, lo que ayudará en la obtención y visualización rápida de resultados.

Además, como **objetivo adicional** nos proponemos dar respuesta a la problemática planteada por el centro sanitario. Para llevar a cabo esta tarea, hemos realizado un estudio exhaustivo utilizando la aplicación Shiny que hemos desarrollado, con el propósito de ofrecer un ejemplo de los posibles usos del aplicativo y demostrar los conocimientos adquiridos a lo largo del grado universitario.

Entre los beneficios de la realización de este proyecto destacamos el desarrollo de modelos capaces de identificar qué antibióticos están contribuyendo a las resistencias bacterianas, proporcionando una solución directa al fenómeno estudiado. Además, la creación de una interfaz interactiva, la cual permite a los usuarios realizar estudios basados en estos modelos, facilita la obtención rápida de resultados. Por otro lado, también garantiza que los análisis puedan replicarse con consistencia y precisión. Esta capacidad de repetición meticulosa es crucial para validar nuevos modelos, así como para actualizar las estrategias utilizadas en el entorno hospitalario a base de nuevos datos obtenidos.

Capítulo 2

Metodología

A continuación, abordamos toda la metodología relacionada con las series temporales, incluyendo los modelos presentes en el estudio y su validación. Además, examinamos todo lo que respecta al conjunto de librerías Shiny, otros recursos informáticos y definiciones sobre el fenómeno de las resistencias bacterianas.

2.1. Análisis de series temporales

En esta primera sección de la metodología expondremos la terminología y conceptos relacionados con el análisis de series temporales. Esto asentará las bases para poder profundizar en la naturaleza de los datos y también introducir los modelos ARMAX y MARS.

2.1.1. Las series temporales

Definición: las series temporales se definen como la secuencia de observaciones recogidas, en orden cronológico, de una misma variable. Se definen por la siguiente expresión matemática:

$$\{X_t\}_{t=1,\dots,T} = \{X_1, X_2, \dots, X_T\}$$

Consideraremos series temporales con valores equiespaciados en el tiempo y medidos en la escala real. Donde t se considera una variable aleatoria, otorgando una realización de la variable X en cada instante de tiempo.

En un análisis descriptivo podemos descomponer la variación de una serie temporal en varias componentes:

$$X_t = T_t + S_t + C_t + Z_t$$

- **Componente de tendencia** T_t : dirección general, ascendente o descendente de los datos durante un largo período de tiempo. No necesariamente ha de crecer o decrecer, puede mantenerse estable.
- **Componente estacional** S_t : oscilaciones regulares y periódicas en un marco temporal menor a un año.
- **Componente cíclico** C_t : variaciones de la serie que ocurren en un lapso de tiempo superior a un año.
- **Componente irregular** Z_t : fluctuaciones de la variable estudiada que no son regulares y son puramente aleatorias o irregulares.

Debido la naturaleza de nuestros datos no se considerará la presencia de componentes estacionales ni cíclicos.

2.1.2. Los procesos estocásticos

Definición: los procesos estocásticos son aquellos formados por una colección de variables aleatorias $\{X_t\}$ con índice temporal t . Las realizaciones de estas variables construyen la previamente definida serie temporal. Lo cual implica que podemos obtener varias realizaciones de series temporales a partir del mismo proceso.

$$F_{(X_{t_1}, \dots, X_{t_2})}(x_{t_1}, \dots, x_{t_2}) = F_{(X_{t_1+h}, \dots, X_{t_2+h})}(x_{t_1+h}, \dots, x_{t_2+h}) \quad \forall t_1, t_2, h$$

Definición: consideraremos un proceso estocástico estacionario, en sentido estricto, cuando las distribuciones de probabilidad conjuntas de cualquier subconjunto de observaciones $(x_{t_1}, \dots, x_{t_2})$ son la misma a lo largo de todo el proceso.

Definición: un proceso estocástico es estacionario, en sentido débil, cuando la media y covarianzas del proceso no dependen del origen de la serie. Expresado de otro modo, podemos valorar un proceso como estacionario si mantiene media y varianza constantes.

2.1.3. Funciones de autocorrelación y Correlogramas

A continuación expondremos las propiedades de los procesos estocásticos. Estas nos serán útiles en la validación de los modelos.

La **función de auto-covariancia** se define de la siguiente manera:

$$\gamma(h) = \text{Cov}(X_t, X_{t+h}) = \mathbb{E}[(X_t - \mu)(X_{t+h} - \mu)]$$

Con propiedades:

- $\gamma(0) = \sigma^2 > 0$
- $\gamma(k) = \gamma(-k) \quad \forall k$
- $|\gamma(k)| \leq \gamma(0)$

La **función de autocorrelación simple** (FAS), o en inglés *Autocorrelation Function* (ACF), tiene la siguiente expresión:

$$\rho(h) = \rho(X_t, X_{t+h}) = \frac{E((X_t - \mu)(X_{t+h} - \mu))}{\sqrt{E((X_t - \mu)^2)E((X_{t+h} - \mu)^2)}} = \frac{\gamma(h)}{\gamma(0)}$$

Con propiedades:

- $\rho(0) = 1$
- $\rho(k) = \rho(-k) \quad \forall k$
- $|\rho(k)| \leq 1 \quad \forall k$

La **función de autocorrelación parcial** (también representada como FAP), o en inglés *Partial Autocorrelation Function* (PACF), es:

$$\text{corr}(X_{t+h}, X_t \mid X_{t+h-1}, \dots, X_{t+1})$$

También es necesario mencionar el caso de *ruido blanco*.

Definición: el ruido blanco es un caso particular de proceso estacionario. Donde las variables que lo componen son incorrelacionadas ($\rho(h) = 0 \forall h$) y el proceso está centrado en cero ($E(X_t) = 0$)

Las funciones de autocorrelación simple y parcial representadas en un **correlograma** son lo que llamamos gráficos ACF y PACF. Estos nos serán útiles a la hora de validar el ajuste de nuestros modelos.

Un buen ajuste del modelo se verifica cuando los gráficos ACF y PACF de los residuos no muestran autocorrelaciones significativas, es decir, todos los valores caen dentro de los intervalos de confianza (generalmente $\pm \frac{1.96}{\sqrt{N}}$, donde N es el tamaño de la muestra). Esto indica que los residuos se comportan como ruido blanco, lo que sugiere que el modelo captura adecuadamente la estructura de la serie temporal y no queda información estructural significativa en los residuos.

Por otro lado, estos gráficos nos permiten identificar si la serie es estacionaria. Debido a la naturaleza de los datos, no fue necesario verificar la estacionariedad de la serie temporal. Los datos utilizados ya cumplen con los requisitos para ser modelizados.

2.2. Modelización

Después de familiarizarnos con toda la terminología relevante sobre el análisis de series temporales, podemos definir los modelos presentes en nuestro estudio.

2.2.1. Modelo ARMAX

Los modelos ARMAX se articulan a partir de los modelos ARMA, los cuales se definen de la siguiente manera:

Definición: un proceso estocástico $\{X_t; t = 0, \pm 1, \pm 2, \dots\}$ es un **Modelo Autorregresivo y de Media Móvil**, ARMA(p,q), si es estacionario y

$$X_t = \phi_1 X_{t-1} + \dots + \phi_p X_{t-p} + Z_t + \theta_1 Z_{t-1} + \dots + \theta_q Z_{t-q}$$

donde: $\phi_p \neq 0$, $\theta_q \neq 0$, $\sigma_z^2 > 0$

Los parámetros p y q denominan los órdenes de la parte autorregresiva y de media móvil, respectivamente.

Definición: un modelo ARMAX combina los elementos de modelos autorregresivos (AR) y de medias móviles (MA) con variables exógenas (X). La sigla ARMAX proviene de *AutoRegressive Moving Average model with exogenous inputs*.

A modo de ejemplo, con el proceso estocástico $\{X_t; t = 0, \pm 1, \pm 2, \dots\}$ y una variable exógena x_t , tenemos la siguiente expresión del modelo:

$$X_t = \beta x_t + \phi_1 X_{t-1} + \dots + \phi_p X_{t-p} + Z_t + \theta_1 Z_{t-1} + \dots + \theta_q Z_{t-q}$$

Por lo tanto, la estructura de un modelo ARMAX se describe como ARMAX(p, q, r). Donde los parámetros p, q y r denominan los órdenes de las partes AR, MA y X, respectivamente.

2.2.2. Modelo MARS

Definición: la Regresión *Spline* Adaptativa Multivariante, conocida en inglés como *Multivariate Adaptive Regression Splines* (MARS; Friedman, 1991), es un método no paramétrico de regresión que combina elementos de la regresión lineal por pasos y los árboles de decisión CART. En MARS, el modelo se construye iterativamente mediante la adición de términos *spline* lineales adaptativos, llamados funciones bisagra, que se forman a partir de los predictores originales. Estas funciones se seleccionan automáticamente durante el proceso de construcción del modelo, utilizando puntos de corte observados en los predictores (Friedman, 1991).

Expresión del modelo MARS:

$$m(x) = \beta_0 + \sum_{m=1}^M \beta_m h_m(x)$$

Definición de las funciones bisagra:

$$h(x) = \begin{cases} x & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$$

Los pasos del proceso de construcción de un modelo MARS son los siguientes:

- Primero, se inicializa el modelo con un término constante: $\hat{m}(x) = \hat{\beta}_0$
- Luego, se consideran todos los puntos de corte posibles para cada predictor, construyendo pares de funciones bisagra: $h_1(x) = h(x_j - x_{ji})$, $h_2(x) = h(x_{ji} - x_j)$. Estas se incorporan al modelo mediante una regresión lineal estándar, donde los parámetros se estiman minimizando el error residual cuadrático: $m(x) = \hat{\beta}_0 + \hat{\beta}_1 h_1(x) + \hat{\beta}_2 h_2(x)$
- El proceso continúa iterativamente, añadiendo nuevos pares de funciones bisagra en cada paso, hasta alcanzar un tamaño máximo especificado por el usuario.
- Una vez construido el modelo, se utiliza un proceso de poda para eliminar términos que no contribuyen significativamente al rendimiento del modelo. La selección del modelo óptimo se realiza mediante métodos como la validación cruzada generalizada (GCV), que evalúa la capacidad predictiva ajustada penalizando el número de parámetros efectivos del modelo. $GCV(\lambda) = \frac{RSS}{\left(1 - \frac{M(\lambda)}{n}\right)^2}$

MARS tiene ventajas significativas como la selección automática de variables predictoras, la capacidad de manejar grandes dimensiones de datos y la interpretación directa del modelo resultante. Es adecuado tanto para datos numéricos como para variables categóricas tras aplicar transformaciones estándar.

2.2.3. Construcción de modelos

Una vez conocida la definición de los modelos ARMAX y MARS podemos introducir la metodología que hemos implementado para desarrollar el estudio.

En primer lugar, llevamos a cabo una exploración preliminar de los datos. Esta exploración implica observar cómo los retardos de las variables explicativas y de la variable respuesta se ajustan en un modelo lineal. Posteriormente, aplicamos un método de selección de variables a este modelo para simplificarlo.

A continuación, con el propósito de determinar cuántos retardos de la variable respuesta nos interesan, ajustamos un modelo ARMAX.

Una vez que conocemos las variables de interés, las ajustamos en el modelo MARS. El procedimiento de este realizará su propia selección de variables, tratando los retardos de la variable respuesta como predictores lineales sin aplicar ninguna función bisagra, mientras que a las demás variables sí se les aplicará. De esta manera, el modelo incluirá aquellas variables que expliquen la variable respuesta.

2.3. Estadísticos y gráficos

En esta sección metodológica definimos los gráficos de validación y los estadísticos presentes en nuestro proyecto. Estos elementos son clave para evaluar el ajuste de los modelos.

2.3.1. Estadísticas de resumen de modelos lineales

El siguiente listado presenta los diferentes estadísticos relacionados con el ajuste de modelos lineales en nuestro estudio.

Residual Standard Error: mide la dispersión de los residuos respecto la recta de regresión.

$$\text{Residual SE} = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{n - p}}$$

R cuadrado: indica la cantidad proporcional de variación en la variable de respuesta y, explicada según las variables independientes X en el modelo.

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

R cuadrado ajustado: similar a R cuadrado, pero ajusta el valor basándose en el número de predictores del modelo y el tamaño de la muestra. Este permite comparar modelos.

$$\text{Adjusted } R^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}$$

Estadístico F: prueba la significancia global del modelo de regresión, comparando la varianza explicada por el modelo con la varianza no explicada (residual).

$$F = \frac{\frac{\sum (\hat{y}_i - \bar{y})^2}{p}}{\frac{\sum (y_i - \hat{y}_i)^2}{n - p}}$$

Usamos el estadístico F para probar la significancia global del modelo de regresión. Planteamos el siguiente test de hipótesis:

- Hipótesis nula H_0 : Todos los coeficientes de regresión (excepto el intercepto) son iguales a cero.

Esto implica que las variables independientes no tienen un efecto significativo en la variable dependiente.

- Hipótesis alternativa H_1 : Al menos uno de los coeficientes de regresión es diferente de cero.

Significa que al menos una de las variables independientes tiene un efecto significativo en la variable dependiente.

Con un nivel de confianza del 95 % y un p valor asociado al estadístico F menor a 0.05 existe suficiente evidencia estadística como para rechazar la hipótesis nula.

2.3.2. Gráficos de validación

A continuación, recorreremos distintas herramientas gráficas de validación de residuos.

Residuos vs Valores Ajustados:

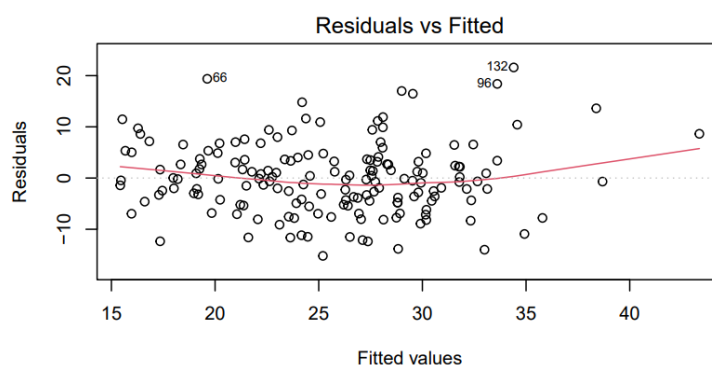


Figura 2.1: Gráficos de validación de modelos: Residuos vs Valores Ajustados

En este gráfico podemos buscar patrones en los residuos. Un patrón aleatorio y disperso sugiere que el modelo es apropiado. Patrones sistemáticos podrían indicar problemas como no linealidad o heteroscedasticidad.

Normal Q-Q:

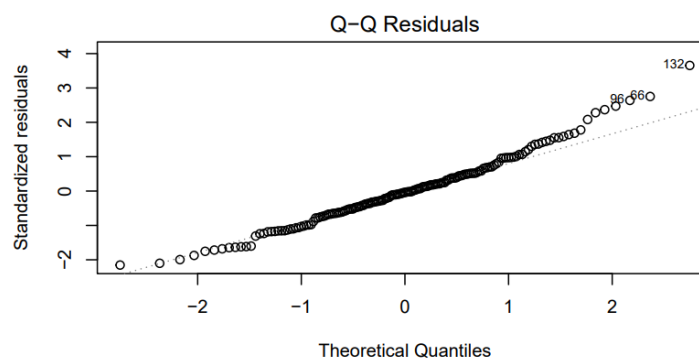


Figura 2.2: Gráficos de validación de modelos: Normal Q-Q

Este es de gran utilidad para verificar la hipótesis de normalidad en los residuos. Si los puntos siguen aproximadamente la línea diagonal, indica que los residuos se distribuyen normalmente, lo que es una suposición clave en muchos modelos estadísticos.

Scale-Location:

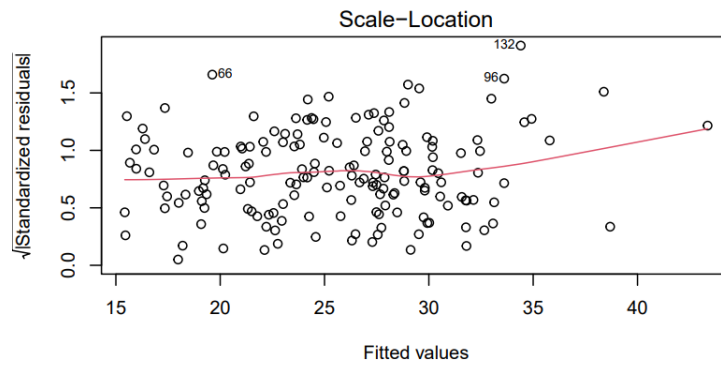


Figura 2.3: Gráficos de validación de modelos: *Scale-Location*

En este grafo se busca homogeneidad en la dispersión de los residuos a lo largo de los valores ajustados. Una dispersión constante sugiere que la varianza de los errores es constante, es decir, se cumple el supuesto de homoscedasticidad.

Residuos vs Apalancamiento:

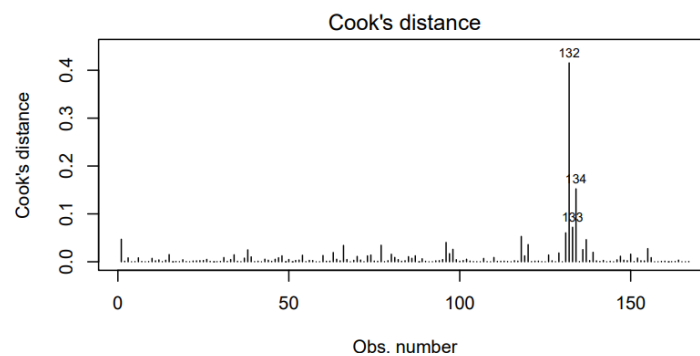


Figura 2.4: Gráficos de validación de modelos: Residuos vs Apalancamiento

Este nos ayuda a identificar observaciones con alto *leverage* o influencia en el modelo. Puntos extremos en este gráfico pueden indicar observaciones que tienen un impacto desproporcionado en el ajuste del modelo.

ACF y PACF:

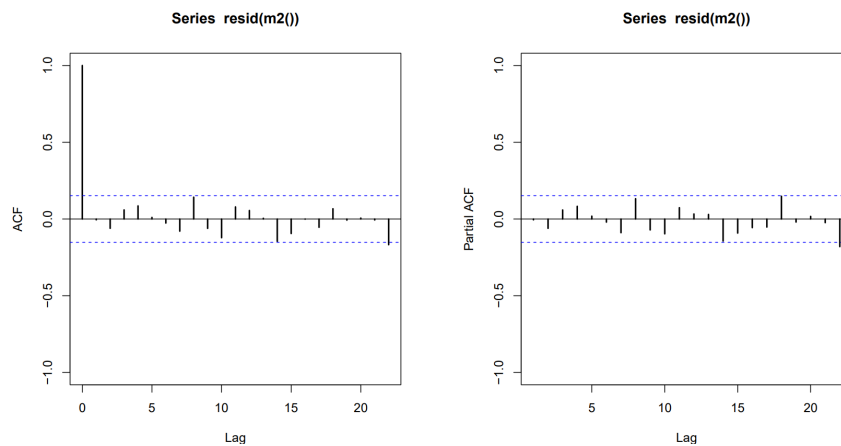


Figura 2.5: Gráficos de validación de modelos: ACF y PACF

Estos gráficos nos serán útiles para determinar la independencia de los residuos. Un patrón de ruido blanco en los gráficos, como hemos explicado previamente, nos verifica la independencia de los residuos.

2.4. Recursos informáticos

Hemos trabajado con RStudio, un entorno de desarrollo integrado (IDE) para el lenguaje de programación R, el cual es ampliamente reconocido por sus capacidades avanzadas en análisis y visualización de datos. RStudio es una herramienta potente y versátil que ofrece gran variedad de librerías para realizar estudios estadísticos de manera eficiente y efectiva.

En los siguientes apartados exploraremos las bibliotecas Shiny, que han posibilitado la creación de la interfaz interactiva. Después, detallaremos todas las funciones y paquetes utilizados en la elaboración de los modelos del estudio. Por último presentaremos varias bibliotecas que han sido útiles en el tratamiento de los datos.

2.4.1. Desarrollo y programación de Shiny en RStudio

En nuestro estudio se han utilizado las siguientes bibliotecas del entorno Shiny: *shiny*, *shinydashboard*, *shinydashboardPlus* y *shinyWidgets*.

Sin necesidad de conocimientos en programación web, estas bibliotecas facilitan la creación de una interfaz interactiva web en R. Shiny simplifica el proceso de desarrollo de aplicaciones proporcionando un marco en el que se pueden combinar de manera efectiva el código de R y la

interfaz de usuario. Esto es posible gracias a la estructura de Shiny, que se basa en dos archivos principales:

- **UI.r:** es el archivo en el cual se define la estructura de la web app, se configuran los inputs con los cuales el usuario puede interactuar y se establecen los lugares de salida de los outputs.
- **Server.r:** en este archivo se realizan todos los cálculos y evaluaciones de la aplicación, es decir, se generan los outputs.

En nuestro proyecto, se ha utilizado *shinydashboard* para crear un tablero de control con múltiples pestañas. Además, hemos decidido emplear *shinydashboardPlus*, una biblioteca recientemente publicada, que ofrece mejores opciones de visualización, al igual que *shinyWidgets*.

Con el objetivo de crear una interfaz flexible a cualquier tipo de datos subidos por el usuario, hemos maximizado las capacidades de este lenguaje de programación. Gracias al uso de objetos reactivos y otros componentes, se ha conseguido que todas las listas de selección de variables y listas de marcado no muestren un listado predeterminado, sino que reflejen todas las variables disponibles en la base de datos cargada.

Para hacer esto posible seguimos los siguientes pasos:

- Primero, implementamos una función de carga de archivos, la cual permite subir datos en formato Excel al usuario.
- Una vez cargado el archivo, extraemos los nombres de las columnas disponibles.
- Posteriormente, utilizamos estos nombres para generar los selectores dinámicos *selectInput* y *checkboxInput* en el servidor, que posteriormente insertamos en la interfaz mediante la función *renderUI*.

De no haber ideado esta solución, no habría sido posible subir datos con columnas diferentes a las nuestras.

Asimismo, hemos querido proporcionar una buena experiencia de usuario. Para evitar la visualización de errores en las salidas, se ha programado un conjunto de mensajes en los outputs cuando no existe un conjunto de datos disponible en el servidor u otros casos.

2.4.2. Desarrollo y programación de modelos en RStudio

Para ajustar los modelos de nuestro estudio en el entorno de programación RStudio hemos usado las siguientes funciones:

- *lm()* de la librería *stats*, que nos permite ajustar modelos lineales.
- *step()* del mismo paquete, la cual nos permite simplificar los modelos lineales mediante el algoritmo *stepwise[]*.
- *auto.arima()* de *forecast*, la cual ajusta de manera automática la parte autorregresiva de nuestro modelo ARMAX.
- *mars()* de la librería *earth*, esta genera el modelo MARS a partir de nuestros datos.

2.4.3. Otras librerías

Para crear gráficos interactivos se ha usado la biblioteca *plotly*, la cual permite generar representaciones visuales, avanzadas y dinámicas de los datos seleccionados.

En cuanto a los análisis estadísticos, se ha recurrido a la biblioteca *psych* para generar resúmenes descriptivos mediante la función *describe*.

Para mejorar la manipulación y el cálculo de los datos, se han utilizado las bibliotecas *tidyverse* y *dplyr*. Estas herramientas permiten realizar operaciones complejas de manipulación de datos de manera sencilla y eficiente, facilitando así la preparación de los datos para su análisis.

Finalmente, para la gestión de archivos Excel con múltiples hojas, se ha utilizado la biblioteca *openxlsx*. Esta biblioteca permite la lectura y escritura de archivos Excel de manera directa, lo que facilita la carga y manipulación de grandes conjuntos de datos estructurados en varias hojas de cálculo.

2.5. Definiciones relacionadas con el fenómeno de las resistencias bacterianas

Por último trataremos con toda la terminología y conceptos relacionados con el fenómeno de las resistencias microbacterianas.

2.5.1. Fenómeno de resistencia bacteriana

La resistencia bacteriana es un fenómeno en el cual las bacterias desarrollan la capacidad de sobrevivir y proliferar en presencia de uno o más antibióticos que anteriormente eran efectivos para eliminarlas o inhibir su crecimiento. Este proceso puede ocurrir de manera natural a través de mutaciones genéticas aleatorias, pero se ve acelerado significativamente por el uso inadecuado o excesivo de antibióticos en la medicina humana, la veterinaria y la agricultura. Las bacterias resistentes pueden transferir sus genes de resistencia a otras bacterias, propagando el problema y creando cepas bacterianas que son cada vez más difíciles de tratar.

2.5.2. Definición de variables AMC y AMR

Definimos las variables AMC y AMR que están presentes en nuestro estudio.

Definición AMC (AntiMicrobial Consumption): esta variable se refiere al consumo de antimicrobianos, es decir, la cantidad de antibióticos utilizados en un determinado ámbito (como un hospital, una comunidad o un país) y durante un período específico. El consumo de antimicrobianos se puede medir en términos de dosis diarias definidas (DDD) por cada 1,000 habitantes o pacientes, entre otros indicadores.

Definición AMR (AntiMicrobial Resistance): esta variable se refiere a la resistencia a los antimicrobianos, es decir, la proporción de bacterias que han desarrollado mecanismos de resistencia frente a los antibióticos. La resistencia se mide a través de pruebas de susceptibilidad antibiótica, que determinan qué proporción de cepas bacterianas son resistentes a ciertos antibióticos en un ámbito y período específicos.

Capítulo 3

La base de datos

En el siguiente capítulo examinamos la base de datos utilizada en este estudio.

Esta base de datos fue proporcionada por el hospital que propuso el estudio al director de este trabajo.

A través de la realización periódica de estudios y pruebas, el centro recopiló datos sobre resistencias microbianas, consumo de antibióticos y número de hospitalizaciones. Por lo tanto, los datos tienen dimensionalidad temporal, es decir, son series temporales. Estas contienen observaciones desde 2008 hasta finales de 2022.

La base de datos incluye las siguientes tablas:

- **Tabla de Ocupación:** contiene el nivel de ocupación del hospital cada mes, es decir, la cantidad de pacientes ingresados.
- **Tabla de Consumos:** incluye el consumo diario definido por cada 1000 estancias de cada medicamento suministrado en el centro.
- **Tabla de Bacterias Resistentes:** por cada tipo de bacteria y medicamento, contiene la cantidad de microorganismos resistentes al tratamiento dado.
- **Tabla de Bacterias Sensibles:** similar a la anterior, pero registra las observaciones de bacterias sensibles.

Además, se incluyen dos diccionarios:

- Diccionario de códigos y definiciones de antibióticos.
- Diccionario de códigos y definiciones de bacterias.

Capítulo 4

Shiny APP: M.R.B.A.

Os presentamos M.R.B.A. (Modelización de Resistencias Bacterianas ante tratamientos Antibióticos), una aplicación Shiny diseñada para realizar estudios estadísticos sobre las resistencias bacterianas. Con esta podremos analizar el impacto de los antibióticos a través de descripciones gráficas y el desarrollo de varios modelos estadísticos.

A continuación, describimos sus principales características y funcionalidades.

M.R.B.A. permite a los usuarios subir sus propios archivos de datos, asegurando flexibilidad y operatividad incluso si las columnas difieren de las esperadas. Esta capacidad de adaptación es crucial para trabajar con diversos tipos de datos sin necesidad de ajustes previos.

La aplicación utiliza la librería Plotly para generar gráficos interactivos. Esto ofrece a los usuarios una amplia gama de funcionalidades gráficas, permitiendo visualizar los datos de manera clara y dinámica.

M.R.B.A. incluye múltiples opciones de modelización, permitiendo ajustar y personalizar modelos con variadas técnicas estadísticas. Las herramientas analíticas integradas son esenciales para el estudio de resistencias bacterianas, facilitando análisis profundos y precisos.

Una vez realizados los análisis y modelizaciones, M.R.B.A. permite descargar todos los outputs generados. Esta funcionalidad es fundamental para documentar y compartir los resultados obtenidos.

Para asegurar que la herramienta sea accesible incluso a los usuarios menos familiarizados, M.R.B.A. cuenta con una guía de uso detallada. Además, se incluyen varios puntos de información a lo largo de la aplicación, proporcionando asistencia adicional en todo momento.

La aplicación y otros recursos se encuentran en el siguiente repositorio:

<https://github.com/oscararlu/Shiny-APP-MRBA>

En este encontraréis el código que permite ejecutar la aplicación, la guía de uso detallada y ejemplos de estructuras de datos.

Capítulo 5

Estudio estadístico y demostraciones de uso

Con el fin de dar respuesta al problema planteado por el centro sanitario y ejemplificar el uso de la aplicación M.R.B.A., desarrollaremos un estudio para modelizar resistencias bacterianas utilizando la interfaz interactiva.

En este apartado mostraremos paso a paso cómo utilizar la aplicación, detallando las decisiones tomadas en cada sección. Explicaremos e interpretaremos cada salida generada, proporcionando interpretaciones estadísticas claras. Además, presentaremos el modelo final desarrollado y las conclusiones obtenidas, explicando cómo este modelo aborda el problema de las resistencias bacterianas.

5.1. Modelización de las resistencias *Escherichia coli* ante la Amoxicilina

En la siguiente sección modelizaremos las resistencias bacterianas de *Escherichia coli* (grupo) ante el tratamiento Amoxicilina/Ácido clavulánico (AMC) a través de los consumos observados en el centro sanitario de Amoxicilina/Ácido clavulánico (AMC), Amoxicilina (AMX) y Ampicilina (AMP).

5.1.1. Carga de ficheros

Para iniciar el estudio primero cargamos los datos al sistema:

	mes	ecoAMK	ecoAMC	ecoCTX	ecoCAZ	ecoCXM	ecoCIP	ecoF
1	2008-01-01	2	8	9	9	9	17	
2	2008-02-01	0	6	8	8	9	16	
3	2008-03-01	2	6	4	4	4	18	
4	2008-04-01	1	12	12	13	14	24	
5	2008-05-01	0	4	3	3	7	29	

Figura 5.1: Demostración de uso: Carga de ficheros

Nos aseguramos de que el archivo esté en el formato correcto y revisamos todas las tablas para verificar que la información se haya cargado correctamente. Luego, para subir los datos al sistema, pulsamos el botón *Usar datos*.

5.1.2. Exploración descriptiva de los datos

Para conocer el comportamiento de *ecoAMC*, nuestra variable respuesta, acudimos a las pestañas de *Datos iniciales* y *Datos transformados*.

Seleccionamos la bacteria y antibiótico asociados a la variable y obtenemos la siguiente tabla de descriptivos:

	vars	n	mean	sd	median	trimmed	mad	min	max	range
ecoAMC_res	1	177	25.15	9.26	24.00	24.74	7.41	4	56.00	52.00
ecoAMC_sen	2	177	67.38	24.17	62.00	65.83	22.24	10	127.00	117.00
total	3	177	92.53	28.39	87.00	91.45	28.17	20	158.00	138.00
Ocupacion	4	177	22813.31	2354.53	22796.00	22804.86	2114.19	16303	29304.00	13001.00
AMC	5	177	165.16	50.32	168.51	168.08	42.18	0	271.81	271.81
			skew	kurtosis	se					
ecoAMC_res			0.51	0.72	0.70					
ecoAMC_sen			0.55	-0.37	1.82					
total			0.33	-0.57	2.13					
Ocupacion			0.01	0.16	176.98					
AMC			-0.84	1.50	3.78					

Figura 5.2: Demostración de uso: Tabla de descriptivos

En la tabla vemos el comportamiento de las siguientes variables:

- **ecoAMC_res** (resistencias observadas): esta variable tiene una media de 25.15 con una desviación estándar de 9.26, lo que indica una dispersión moderada de los valores alrededor de la media respecto las otras variables. La mediana es 24.00, y el rango intercuartílico, medido por MAD (Desviación Absoluta Mediana), es de 7.41. Los valores oscilan entre 4 y 56, con un rango total de 52. La distribución presenta una ligera asimetría positiva ($\text{skew} = 0.51$) y una kurtosis de 0.72, sugiriendo que la distribución es levemente más alargada que una normal.
- **ecoAMC_sen** (sensibilidades observadas): esta variable tiene una media de 67.38 y una desviación estándar de 24.17, indicando una mayor variabilidad en comparación con ecoAMC_res. La mediana se sitúa en 62.00, con un MAD de 22.24. Los valores van desde 10 hasta 127, lo que da un rango total de 117. La distribución también muestra una asimetría positiva ($\text{skew} = 0.55$) y una kurtosis de -0.37, lo que indica una distribución ligeramente más plana que una normal.
- **Ocupación**: esta variable tiene una media de 22813.31 y una desviación estándar de 2354.53, lo que muestra una variabilidad considerable. La mediana es 22796.00 y el MAD es 2114.19. Los valores fluctúan entre 16303 y 29304, resultando en un rango total de 13001. La distribución es casi simétrica ($\text{skew} = 0.01$) y presenta una kurtosis de 0.16, lo que indica una distribución cercana a la normal.
- **AMC**: con una media de 165.16 y una desviación estándar de 50.32, esta variable muestra una variabilidad significativa. La mediana es 168.51 y el MAD es 42.18. Los valores van desde 0 hasta 271.81, con un rango total de 271.81. La distribución presenta una asimetría negativa ($\text{skew} = -0.84$) y una kurtosis de 1.50, lo que sugiere una distribución más alargada y con colas más pesadas.

En resumen, las variables presentan diferentes grados de dispersión y asimetría. La variable Ocupación es la que muestra una distribución más normal, mientras que AMC tiene una distribución con mayor asimetría y curtosis. Las otras variables presentan características intermedias con algunas desviaciones de la normalidad.

A continuación estudiamos mediante gráficos la variable respuesta.

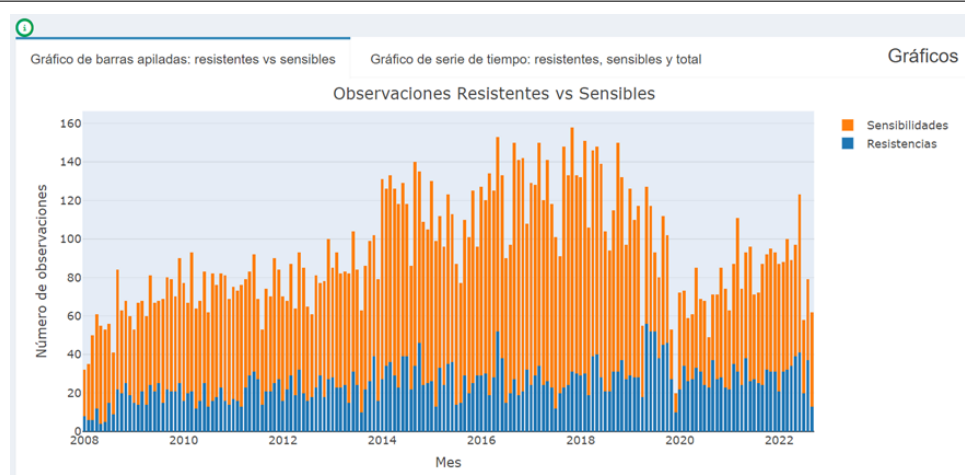


Figura 5.3: Demostración de uso: Gráfico de barras apiladas

En el gráfico de barras apiladas vemos como la cantidad de pruebas disminuye a partir de 2020, pero la cantidad de observaciones resistentes se mantiene, la cual cosa nos indica una mayor resistencia bacteriana en los últimos años.

Este comportamiento se puede observar mejor en el siguiente gráfico, donde verificamos que el total y las sensibilidades observadas disminuyen, pero las resistencias no.

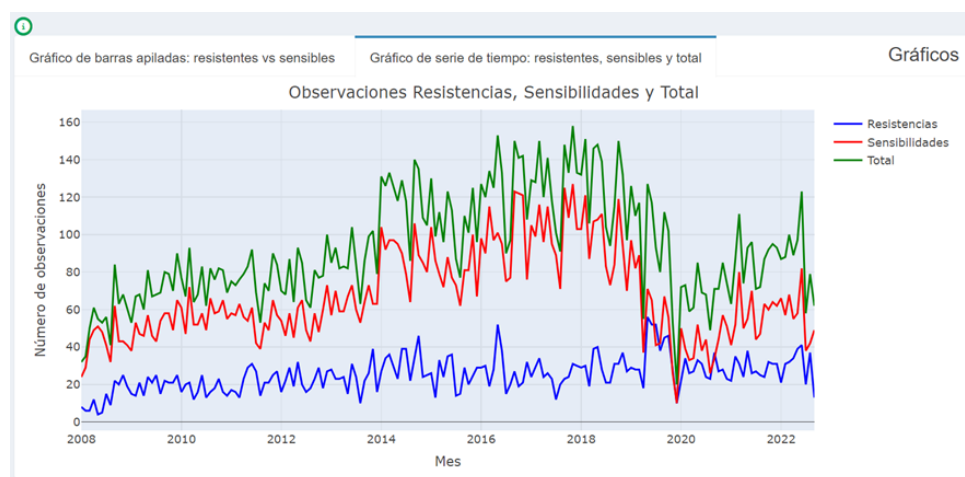


Figura 5.4: Demostración de uso: Gráfico de serie de tiempo

Por otro lado podemos ver la evolución de los consumos. Vemos una clara tendencia negativa en el comportamiento de estos.

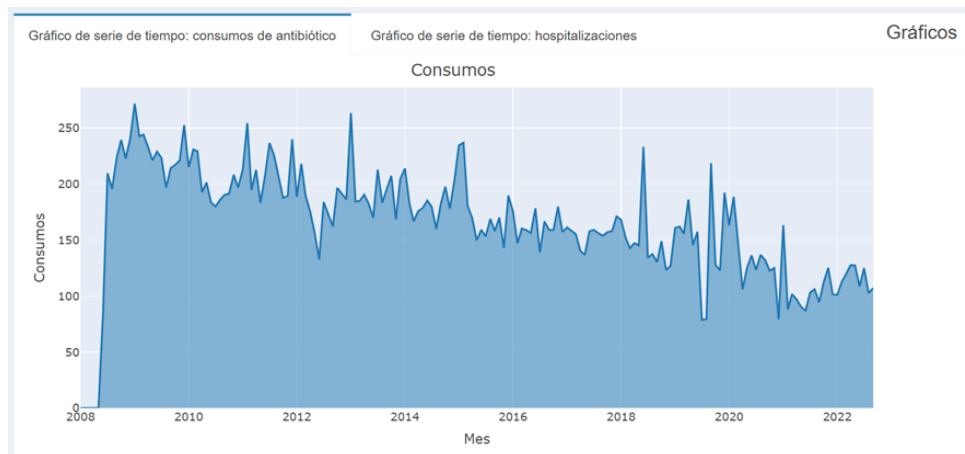


Figura 5.5: Demostración de uso: Gráfico de serie de tiempo

Para seguir explorando y comprendiendo los datos vamos a ver unos indicadores elaborados.

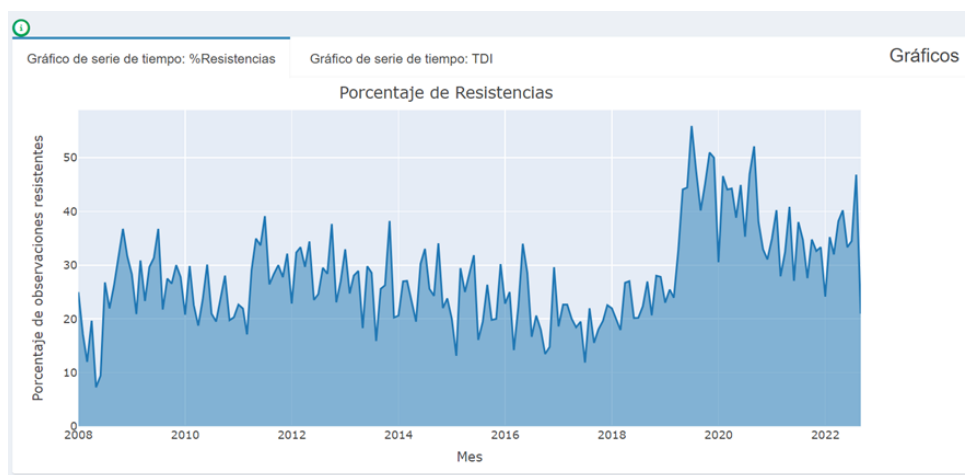


Figura 5.6: Demostración de uso: Gráfico de serie de tiempo: %Resistencias

En el porcentaje de resistencias vemos como efectivamente las resistencias han aumentado proporcionalmente, tal y como hemos interpretado previamente.

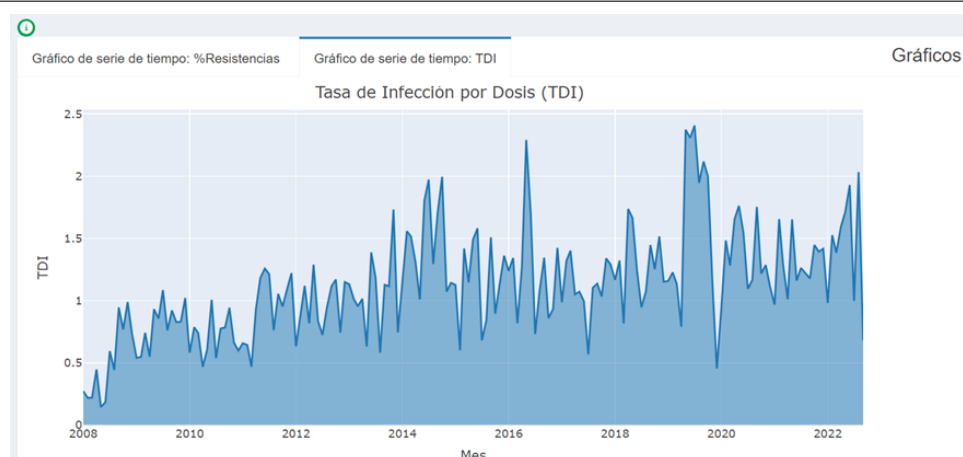


Figura 5.7: Demostración de uso: Gráfico de serie de tiempo: TDI

Por otro lado, la tasa de infección por dosis ha aumentado en los últimos años. Este aumento nos indica que cada vez más pacientes están contrayendo la infección en relación con la cantidad de dosis administradas. Esto puede deberse a una disminución en la eficacia del tratamiento y la aparición de cepas más resistentes del microorganismo. Este fenómeno sugiere la necesidad de revisar y ajustar la estrategia de suministración de antibióticos.

Modelización

En el siguiente apartado utilizaremos la metodología que ofrece la aplicación para ajustar un modelo MARS que aborde la problemática planteada por el hospital.

5.1.3. Selección de variables

Primero será necesario seleccionar las variables a modelizar:

Selector de variables manual

Suba archivo:

Selector a partir de fichero

ecoAMC

Usar modelo

Showing 1 to 5 of 172 entries

	ecoAMC	AMC	AMX	AMP
6	5	85.18	17.52	1.73
7	15	209.69	34.24	0.83
8	9	195.78	35.67	2.51
9	22	223.42	29.93	2.88
10	20	239.52	53.46	2.42

Previous 1 2 3 4 5 ... 35 Next

Figura 5.8: Demostración de uso: Selector de variables

Para seleccionar las variables decidimos utilizar un archivo proporcionado por el centro sanitario. Este archivo identifica, para cada variable de respuesta disponible en su base de datos, los consumos que pueden describir el comportamiento de las tolerancias bacterianas. Como mencionamos previamente, en el caso de *Escherichia coli* (grupo) ante el tratamiento con Amoxicilina/Ácido clavulánico (AMC), las variables Amoxicilina/Ácido clavulánico (AMC), Amoxicilina (AMX) y Ampicilina (AMP) nos serán útiles.

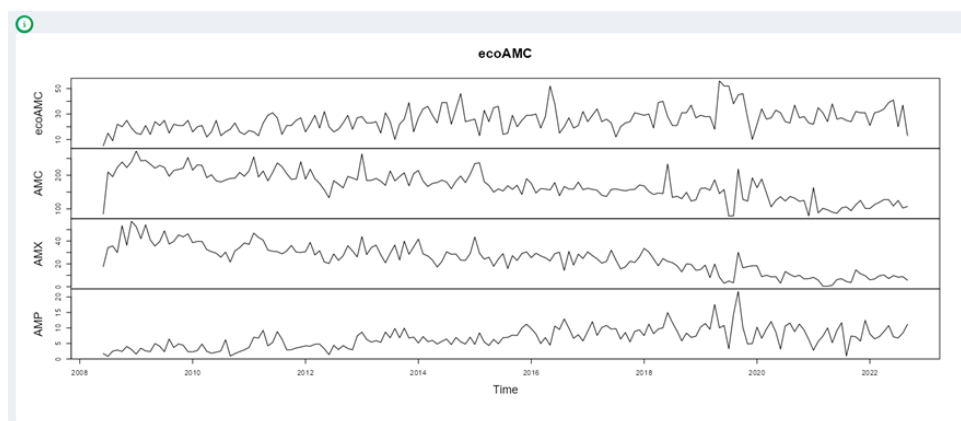


Figura 5.9: Demostración de uso: Gráfico de series temporales del modelo

5.1.4. Exploración preliminar de modelos

A continuación desarrollamos una exploración preliminar de modelos. Para ello seleccionamos la cantidad de retardos de cada variable a calcular. En nuestro caso decidimos calcular uno para la variable respuesta y ocho para las variables explicativas.

Figura 5.10: Demostración de uso: Selector de retardos a partir de cifras

Después de pulsar *Ajustar modelo* podemos visualizar el ajuste de un modelo completo y otro simplificado.

Modelo lineal completo

El ajuste del modelo lineal completo es el siguiente:

```

Call:
lm(formula = form0, data = na.omit(df))

Residuals:
    Min       1Q   Median       3Q      Max
-14.9701  -5.1320  -0.5873   4.4537  21.9782

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.1749136   7.9613651    1.404  0.16267
ecoAMC_1      0.2179262   0.0844795    2.580  0.01094 *
AMC_1        -0.0548180   0.0290352   -1.888  0.06113 .
AMC_2        -0.0375844   0.0296375   -1.268  0.20688
AMC_3         0.0333055   0.0300066    1.110  0.26896
AMC_4         0.0792700   0.0303061    2.616  0.00990 **
AMC_5        -0.0137526   0.0308065   -0.446  0.65599
AMC_6        -0.0002733   0.0299961   -0.009  0.99274
AMC_7         0.0608540   0.0296409    2.053  0.04196 *
AMC_8         0.0233095   0.0306583    0.760  0.44837
AMX_1         0.0957409   0.1137701    0.842  0.40151
AMX_2        -0.0756662   0.1154023   -0.656  0.51313
AMX_3        -0.0025641   0.1185980   -0.022  0.98278
AMX_4         0.0106391   0.1199280    0.089  0.92944
AMX_5        -0.1559515   0.1219186   -1.279  0.20299
AMX_6        -0.0241701   0.1233377   -0.196  0.84492
AMX_7        -0.2902584   0.1214913   -2.389  0.01824 *
AMX_8        -0.0168334   0.1209450   -0.139  0.88951
AMP_1         0.3606441   0.2473979    1.458  0.14718
AMP_2         0.6581059   0.2511386    2.620  0.00976 **
AMP_3        -0.0392639   0.2581964   -0.152  0.87935
AMP_4        -0.1237663   0.2507870   -0.494  0.62244
AMP_5        -0.0407290   0.2505103   -0.163  0.87108
AMP_6        -0.0597234   0.2514420   -0.238  0.81260
AMP_7         0.0901073   0.2485634    0.363  0.71752
AMP_8        -0.1456540   0.2421145   -0.602  0.54843
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Figura 5.11: Demostración de uso: Ajuste del modelo lineal completo en la exploración preliminar

Coeficientes y significancia estadística

En la estimación de los coeficientes vemos como varios son estadísticamente significativos, lo que nos indica que tienen un impacto considerable en la variable dependiente.

A continuación interpretamos los efectos de los coeficientes significativos:

- **ecoAMC_1**: este coeficiente nos indica que un aumento en una unidad de ecoAMC_1 está asociado con un aumento promedio de 0.218 en la variable dependiente. Esto sugiere que ecoAMC_1 es una variable explicativa importante en el modelo, lo cual nos informa de la necesidad de una parte autoregresiva en nuestro modelo.
- **AMC_4 y AMC_7**: ambos coeficientes son positivos y significativos, lo que nos indica que incrementos en estas variables están asociados con incrementos en la variable dependiente.

Esto sugiere que ciertos aspectos de la Amoxicilina tienen un efecto positivo sobre la variable dependiente.

- **AMX_7**: este coeficiente es negativo y significativo, lo que sugiere que un aumento en AMX_7 está asociado con una disminución en la variable dependiente. Esto indica una relación inversa importante entre AMX_7 y la variable dependiente.
- **AMP_2**: con un coeficiente positivo y significativo, AMP_2 muestra una fuerte asociación positiva con la variable dependiente, sugiriendo que aumentos en AMP_2 resultan en aumentos sustanciales en la variable dependiente.

La mayoría de los otros coeficientes no son estadísticamente significativos bajo un nivel de confianza del 95 %, lo que implica que no hay suficiente evidencia para afirmar que tienen un efecto distinto de cero sobre la variable dependiente. La falta de significancia en estos coeficientes sugiere que estas variables explicativas no tienen un impacto fuerte o consistente en la variable dependiente y por lo tanto no aportan información al modelo.

Evaluación del modelo

```
Residual standard error: 7.329 on 138 degrees of freedom  
Multiple R-squared:  0.4077,    Adjusted R-squared:  0.3004  
F-statistic: 3.799 on 25 and 138 DF,  p-value: 2.391e-07
```

Figura 5.12: Demostración de uso: Estadísticos de ajuste del modelo lineal completo exploración preliminar

En la evaluación del ajuste del modelo, observamos que el error estándar residual es de 7.329. El coeficiente de determinación R^2 es 0.4077, indicando que aproximadamente el 40.77 % de la variabilidad en la variable dependiente se explica por las variables incluidas en el modelo. El R^2 ajustado es 0.3004, que a comparación con el R^2 , sugiere que algunas variables no contribuyen significativamente al modelo.

La estadística F del modelo es 3.799 con 25 y 138 grados de libertad, con un valor p asociado extremadamente bajo. Esto nos indica que el modelo es estadísticamente significativo y que al menos uno de los coeficientes de las variables explicativas es diferente de cero. Esto demuestra que las variables explicativas mejoran significativamente la predicción de la variable dependiente en comparación con un modelo sin predictores.

Validación

Si validamos el modelo mediante herramientas gráficas vemos como:

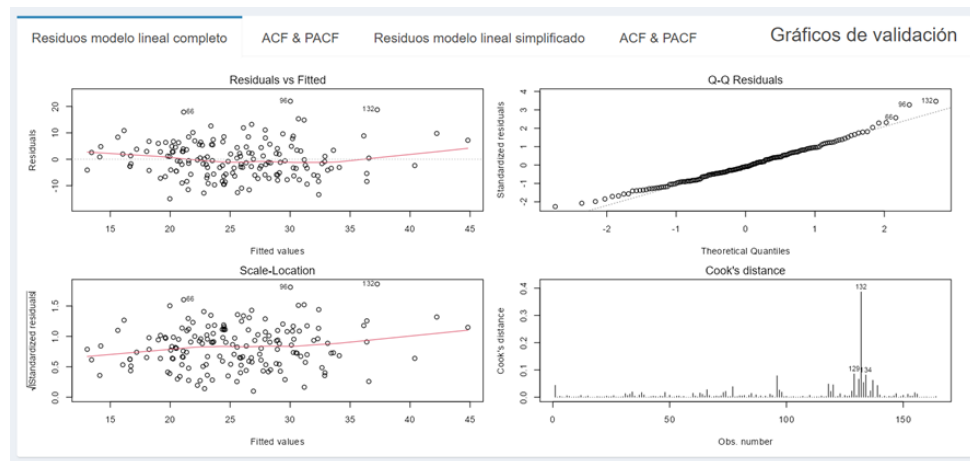


Figura 5.13: Demostración de uso: Gráficos de validación ajuste del modelo lineal completo exploración preliminar

En el primer gráfico de residuos versus valores ajustados, no observamos patrones claros en los residuos, lo que sugiere que la suposición de linealidad es razonable. En el segundo gráfico, el Normal Q-Q, a pesar de que las colas se desvían ligeramente de lo esperado en una distribución normal, consideramos a los residuos aproximadamente normales. En el tercer gráfico, Scale-Location, no vemos problemas significativos de homocedasticidad, ya que no hay una tendencia clara en la dispersión de los residuos. Finalmente, en el gráfico de apalancamiento, aunque existe algún punto influyente, lo consideraremos aleatorio.

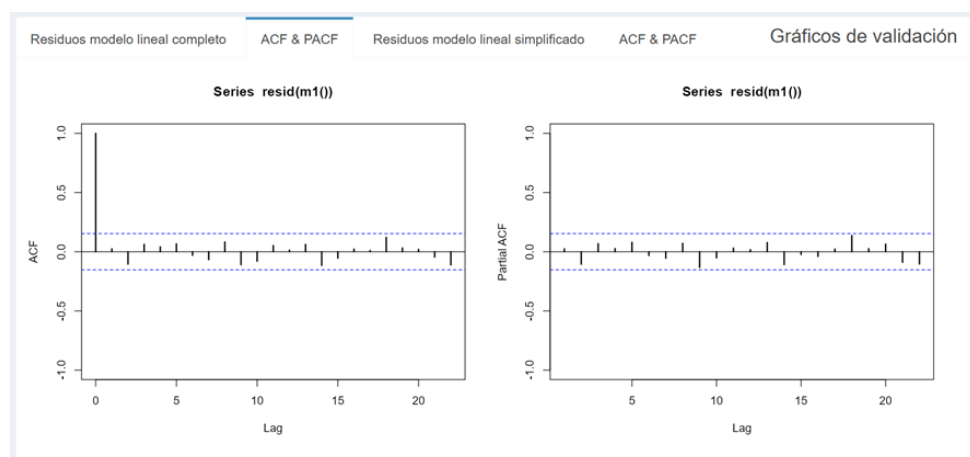


Figura 5.14: Demostración de uso: Gráfico ACF & PACF del ajuste del modelo lineal completo exploración preliminar

Por otro lado, vemos como los residuos no muestran signos de auto correlación, por lo que los consideramos independientes.

En definitiva, aún siendo un modelo validado y teniendo varios coeficientes estadísticamente significativos, el ajuste de este es mejorable. Consideramos la simplificación del modelo para mejorar su ajuste y capacidad explicativa.

Modelo lineal simplificado

A continuación estudiamos el ajuste del modelo lineal simplificado.



Figura 5.15: Demostración de uso: Ajuste del modelo lineal simplificado exploración preliminar

Evaluación del modelo

El modelo simplificado obtenido ha reducido el número de variables explicativas a siete: ecoAMC_1, AMC_2, AMC_4, AMC_7, AMX_5, AMX_7 y AMP_2. Vemos como la gran mayoría de estas son significativas en el modelo anterior.

Los residuos del modelo simplificado tienen un rango de -15.0670 a 21.2816, con una

mediana cercana a cero y un error estándar residual de 7.07, indicando una ligera mejora en la precisión del modelo respecto al original. El R^2 múltiple es 0.3768 y el R^2 ajustado es 0.3488, mostrando que el 37.68 % de la variabilidad en ecoAMC se explica por las variables del modelo, mejorando en esta métrica al anterior modelo.

El modelo simplificado incluye varios coeficientes significativos. El intercepto, con un valor p menor a 0.05, indica que el valor medio de ecoAMC es aproximadamente 10.85 cuando todas las variables explicativas son cero.

Prácticamente todas las variables son altamente significativas. Por ejemplo, ecoAMC_1 tiene un coeficiente positivo de 0.21192, sugiriendo que un aumento en ecoAMC_1 está asociado con un incremento promedio de 0.212 en ecoAMC, mientras que AMC_2 y AMX_7 tienen coeficientes negativos, indicando una disminución en ecoAMC con sus respectivos aumentos.

La estadística F del modelo es 13.47 con 7 y 156 grados de libertad, y un valor p extremadamente bajo, lo que sugiere que el modelo en su conjunto es altamente significativo, mejorando al anterior.

Validación

Veamos los gráficos de validación.

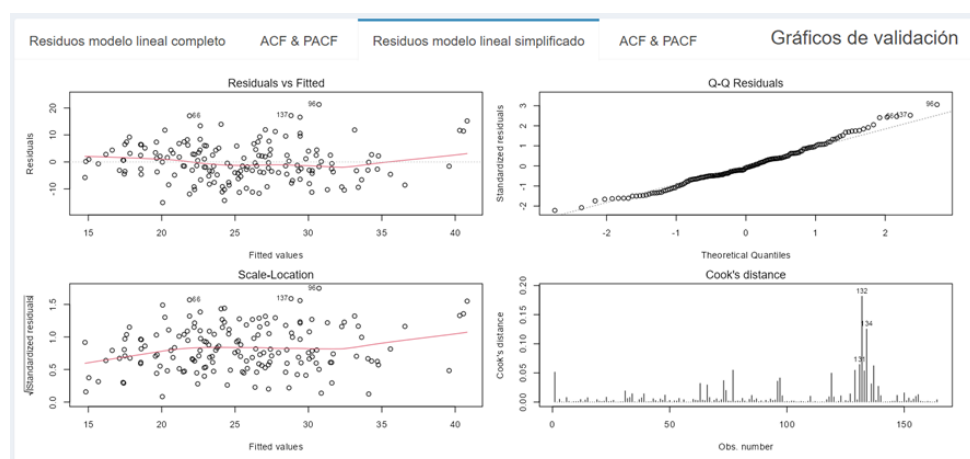


Figura 5.16: Demostración de uso: Gráficos de validación ajuste del modelo lineal simplificado exploración preliminar

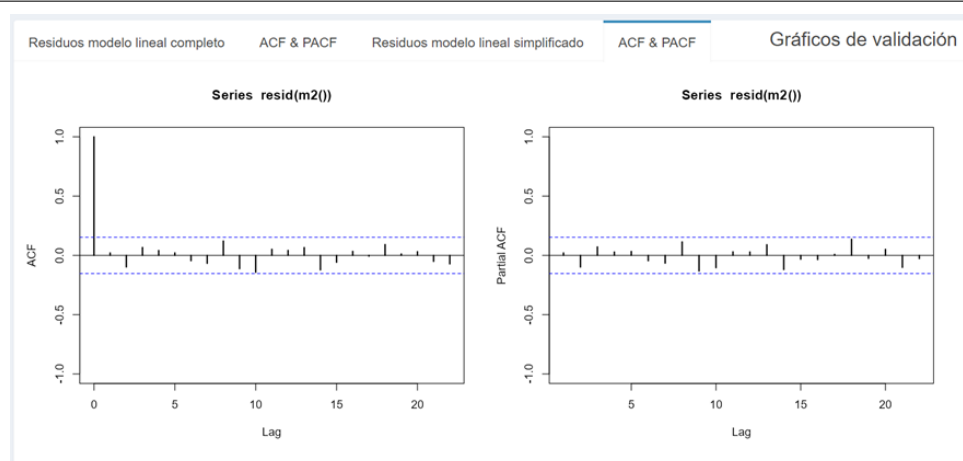


Figura 5.17: Demostración de uso: Gráficos ACF & PACF del modelo lineal simplificado exploración preliminar

En la validación gráfica del modelo, observamos que este cumple con todas las suposiciones relativas a los residuos. En el gráfico de residuos versus valores ajustados, no se detectan patrones claros, lo que sugiere que la relación entre las variables es lineal. Además, el gráfico Normal Q-Q muestra que los residuos se distribuyen aproximadamente de forma normal, aunque con algunas desviaciones menores en las colas.

El gráfico Scale-Location indica que la dispersión de los residuos no presenta tendencias significativas, lo que sugiere que no hay problemas de homocedasticidad; es decir, la variabilidad de los residuos es constante a lo largo de los valores ajustados. Por otro lado, en el gráfico de apalancamiento, aunque se identifican algunos puntos influyentes, estos no son lo suficientemente significativos como para afectar el modelo. Por último en los gráficos ACF & PACF no vemos problemas en la autocorrelación de los residuos por lo que los consideramos independientes.

En resumen, el modelo simplificado ha logrado una mejora en la precisión y la significancia estadística, destacando `ecoAMC_1`, `AMC_2`, `AMC_4`, `AMC_7`, `AMX_7` y `AMP_2` como variables explicativas clave que afectan significativamente a `ecoAMC`.

5.1.5. Selección de variables de AMC

Después de estudiar los modelos hemos optado por seleccionar los retrasos del modelo simplificado para incorporarlos en el modelo MARS. Esta elección garantiza el uso de variables significativas y simplifica el procedimiento MARS.



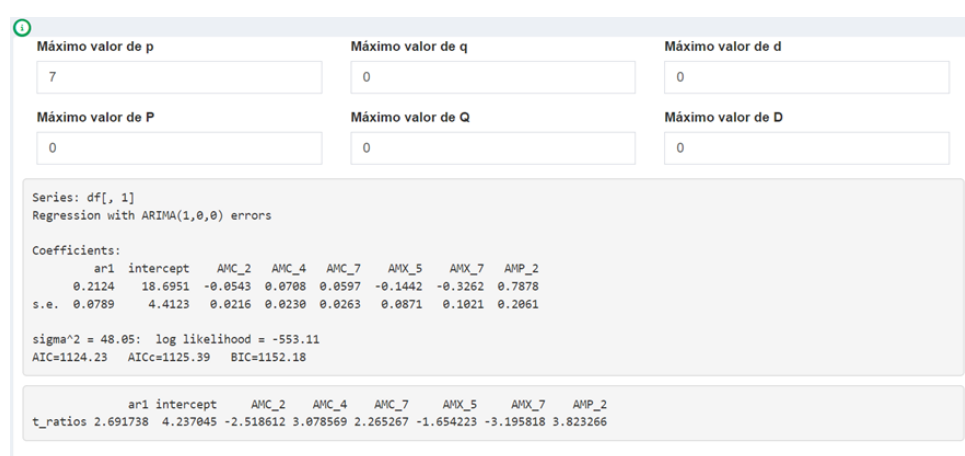
Variables AMC del modelo:

Simplificado

Figura 5.18: Demostración de uso: Selección de variables AMC

5.1.6. Ajuste del modelo ARMAX

Ahora procedemos con el ajuste del modelo ARMAX, el cual nos permitirá determinar el número de retardos para la parte autoregresiva de nuestro modelo MARS.



Máximo valor de p: 7

Máximo valor de q: 0

Máximo valor de d: 0

Máximo valor de P: 0

Máximo valor de Q: 0

Máximo valor de D: 0

Series: df[, 1]
Regression with ARIMA(1,0,0) errors

Coefficients:

	ar1	intercept	AMC_2	AMC_4	AMC_7	AMX_5	AMX_7	AMP_2
	0.2124	18.6951	-0.0543	0.0708	0.0597	-0.1442	-0.3262	0.7878
s.e.	0.0789	4.4123	0.0216	0.0230	0.0263	0.0871	0.1021	0.2061

sigma^2 = 48.05: log likelihood = -553.11
AIC=1124.23 AICc=1125.39 BIC=1152.18

	ar1	intercept	AMC_2	AMC_4	AMC_7	AMX_5	AMX_7	AMP_2
t_ratios	2.691738	4.237045	-2.518612	3.078569	2.265267	-1.654223	-3.195818	3.823266

Figura 5.19: Demostración de uso: Ajuste del modelo ARMAX

Vemos como el modelo ARMAX ajustado a la serie incluye un término autorregresivo de orden 1 (AR(1)) y varias variables regresoras (AMC_2, AMC_4, AMC_7, AMX_5, AMX_7 y AMP_2). A continuación se presentan los aspectos más destacados del ajuste del modelo ARMAX:

Coefficientes y significancia estadística

El coeficiente AR(1) es significativo con un t ratio mayor a 2, lo que nos indica que el valor de la variable dependiente en el tiempo t está influenciado por su valor en el tiempo $t - 1$. Esto sugiere una dependencia temporal en la serie de tiempo, la cual somos capaces de capturar gracias al propio coeficiente AR(1).

El intercepto es altamente significativo (valor $t > 2$), sugiriendo que, en ausencia de los efectos de las variables regresoras, la variable dependiente tiene un valor base promedio de aproximadamente 18.6951.

Que el intercepto sea significativo en el modelo nos proporciona una base sólida para confiar en las estimaciones del modelo, asegurando que se capture adecuadamente la media condicional de la serie temporal y que se interpreten correctamente los efectos de las variables explicativas y los términos autorregresivos.

AMC_2 y AMX_7 tienen un coeficiente negativo y significativo (valor $t < -2$), indicando que un aumento en estos está asociado con una disminución en la variable dependiente.

AMC_4, AMC_7 y AMP_2 son positivos y significativos (valor $t > 2$), sugiriendo que un aumento en estos está asociado con un aumento en la variable dependiente.

Aunque AMX_5 tiene un coeficiente negativo, no es estadísticamente significativo al nivel convencional (valor $t < -2$), lo que sugiere que su efecto en la variable dependiente no es concluyente.

Evaluación del modelo

El modelo tiene una varianza residual σ^2 de 48.05, lo que nos proporciona una medida de la variabilidad de los errores no explicados por el modelo. El logaritmo de verosimilitud es -553.11, y los criterios de información AIC, AICc y BIC son 1124.23, 1125.39 y 1152.18, respectivamente. Estos valores pueden usarse para comparar este modelo con otros modelos similares para determinar el mejor ajuste.

En resumen, el modelo ARMAX con AR(1) y las variables exógenas seleccionadas nos muestran que ecoAMC tiene una dependencia temporal significativa y está influenciada por las variables AMC_2, AMC_4, AMC_7, AMX_7 y AMP_2, con efectos tanto positivos como negativos. Las significancias de los coeficientes, como se reflejan en los t ratios, sugieren que estas variables tienen un impacto notable en la variable dependiente, con AMC_4, AMC_7 y AMP_2 mostrando efectos positivos significativos y AMC_2 y AMX_7 mostrando efectos negativos significativos.

A continuación vemos el ajuste del modelo lineal completo:

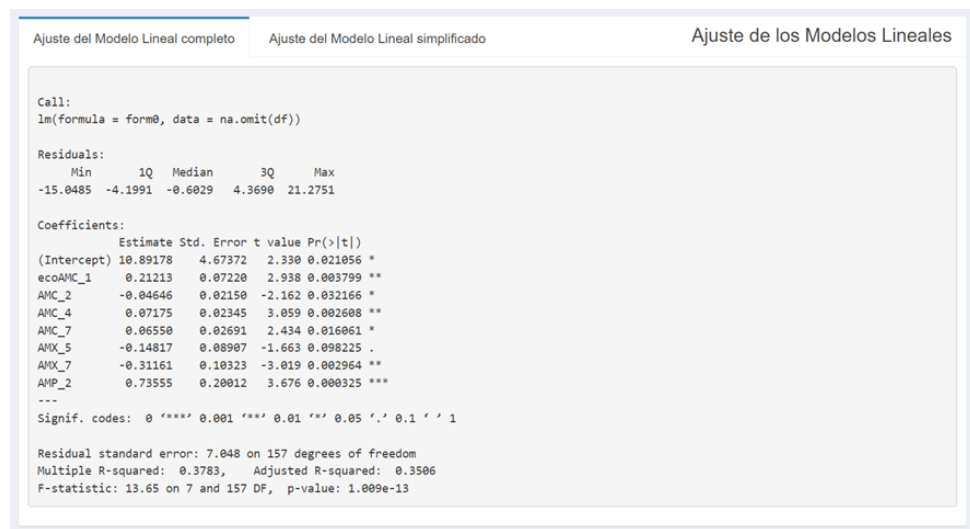


Figura 5.20: Demostración de uso: Ajuste modelo lineal ARMAX

Observamos que tanto el modelo lineal completo con retardos ARMAX como el modelo simple son idénticos al modelo simple de la exploración preliminar. Por lo tanto, como hemos discutido en la sección anterior, consideramos a ambos modelos válidos.

Puedes ver el resto de las figuras del ajuste y validación en el apéndice B

5.1.7. Modelo MARS

En esta última sección ajustamos el modelo MARS, explicamos el proceso que ha desarrollado el método y lo interpretaremos con el fin de dar solución a las problemáticas planteadas por el centro sanitario.

```
Call: earth(formula=form3, data=na.omit(df), pmethod="exh", trace=4, nprune=np, nfold=1,
            nk=ng, minspan=ms, linpreds=paste0(AMR,"_",1:p))

x[165,7]:
  ecoAMC_1  AMC_2  AMC_4  AMC_7  AMX_5  AMX_7  AMP_2
1      15 195.7800 239.520 271.810 36.210 52.620000 2.510000
2       9 223.4200 222.850 242.710 57.310 42.070000 2.880000
3      22 239.5200 240.390 244.290 52.620 54.230000 2.420000
...     20 222.8500 271.810 232.960 42.070 40.900000 4.030000
165     32 127.5952 108.603 107.232  8.318  5.662732 8.870572

y[165,1]:
  ecoAMC
1       5
2      15
3       9
...     22
165     31
```

Figura 5.21: Demostración de uso: Ajuste modelo MARS: Los datos

Vemos como los datos introducidos incluyen 165 observaciones con las variables ecoAMC_1, AMC_2, AMC_4, AMC_7, AMX_5, AMX_7 y AMP_2 y la variable de respuesta ecoAMC.

```
Linear predictors 1=ecoAMC_1
Forward pass: minspan 50 endspan 10  x[165,7] 9.02 kB  bx[165,10] 12.9 kB

      GRSq  RSq      DeltaRSq Pred      PredName      Cut  Terms  Par Deg
1      0.0000 0.0000              5      (Intercept)              2   3      1
2      0.1806 0.2201      0.2201  5      AMX_5      11.369  2   3      1
4      0.2250 0.2987      0.07869  7      AMP_2       9.66  4   5      1
6      0.2561 0.3440      0.04527  1      ecoAMC_1       9<  6      1
8      0.2518 0.3741      0.03004  3      AMC_4      169.87  7   8      1 final (reached nk 10)

Reached nk 10
After forward pass GRSq 0.252 RSq 0.374
Forward pass complete: 9 terms, 8 terms used
```

Figura 5.22: Demostración de uso: Ajuste modelo MARS: *Forward Pass*

En el *Forward Pass* se seleccionan términos basados en la reducción del error cuadrático medio y el aumento del R^2 . El modelo inicia con un intercepto y añade progresivamente variables explicativas. Al final del procedimiento el modelo alcanzó un $GRSq$ (Generalized R^2) de 0.252 y un R^2 de 0.374 con 9 términos.

```

Exhaustive pruning: number of subsets 246   bx sing val ratio 0.0053
Subset size      GRSq      RSq   DeltaGRSq nPreds  Terms (col nbr in bx)
      1          0.0000   0.0000    0.0000      0    1
      2          0.1737   0.1937    0.1737      1    1 2
      3          0.2389   0.2755    0.0652      2    1 2 4
      4          0.2770   0.3289    0.0381      3    1 2 6 4
chosen 5          0.3050   0.3711    0.0280      4    1 2 8 4 6
      6          0.2897   0.3737   -0.0153      4    1 2 5 6 8 4

Prune exhaustive penalty 2 nprune 6: selected 5 of 8 terms, and 4 of 7 preds
After pruning pass GRSq 0.305 RSq 0.371
Selected 5 of 8 terms, and 4 of 7 predictors (pmethod="exhaustive") (nprune=6)
Termination condition: Reached nk 10
Importance: AMX_5, AMP_2, ecoAMC_1, AMC_4, AMC_2-unused, AMC_7-unused, AMX_7-unused
Number of terms at each degree of interaction: 1 4 (additive model)
GCV 53.48882   RSS 7889.115   GRSq 0.3049715   RSq 0.3711253

```

Figura 5.23: Demostración de uso: Ajuste modelo MARS: Podado

En el podado exhaustivo se evalúan 246 subconjuntos de términos para seleccionar el mejor modelo. El modelo seleccionado después de este proceso incluye 5 de 8 posibles términos y 4 de 7 posibles explicativas. Los términos seleccionados fueron basados en su importancia y contribución al modelo.

Coeficientes y términos seleccionados

Por orden de importancia, tenemos las siguientes variables AMC seleccionadas para el modelo final:

- AMX_5: este término fue el primero en ser seleccionado, indicando su alta importancia en el modelo.
- AMP_2: también mostró alta importancia y fue uno de los primeros en ser seleccionado.
- ecoAMC_1: seleccionado posteriormente, pero mostrando una contribución significativa.
- AMC_4: incluido en el modelo final por su contribución adicional.

AMC_2, AMC_7 y AMX_7, aunque fueron considerados, no se incluyen en el modelo final.

Desempeño del modelo

Hemos obtenido un R^2 Generalizado (GRSq) de 0.305. Esto nos indica que aproximadamente el 30.5 % de la variabilidad de los datos puede ser explicada por el modelo ajustado.

Por otro lado tenemos un R^2 de 0.371. Esto sugiere que el modelo explica el 37.1 % de la variabilidad en la variable de respuesta.

En el Criterio de Validación Generalizado (GCV) obtenemos 53.48882. El GCV es una medida de ajuste en la cual se penaliza al modelo por la complejidad del mismo, esto nos permite comparar varios modelos con diferente número de términos.

En la Suma Residual de Cuadrados (RSS) tenemos un valor de 7889.115. Esta es la suma de los cuadrados de los residuos, una medida de la variabilidad no explicada por el modelo.

Así, aunque el modelo final explica una parte de la variabilidad en la variable de respuesta, todavía queda una cantidad considerable de variabilidad sin explicar. Esto sugiere que podría ser necesario considerar variables adicionales o realizar transformaciones de los datos para mejorar el ajuste del modelo.

Validación

Veamos la validación del modelo:

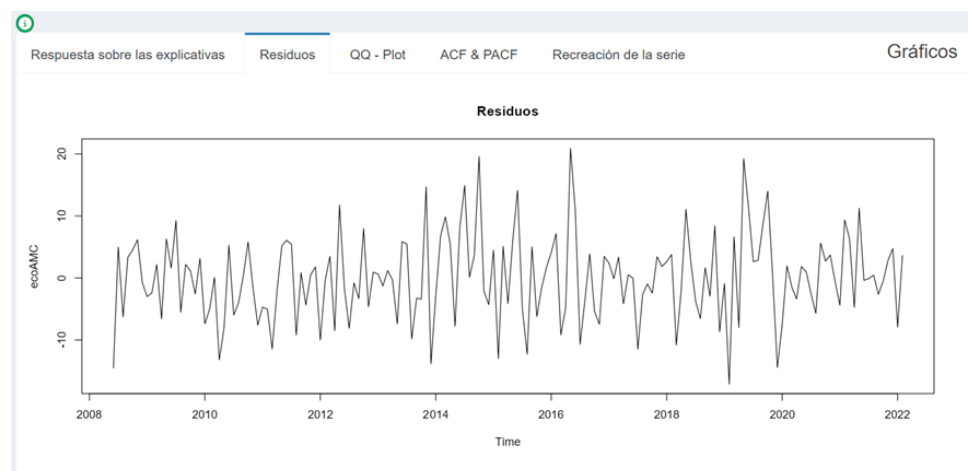


Figura 5.24: Demostración de uso: Gráfico de residuos modelo MARS

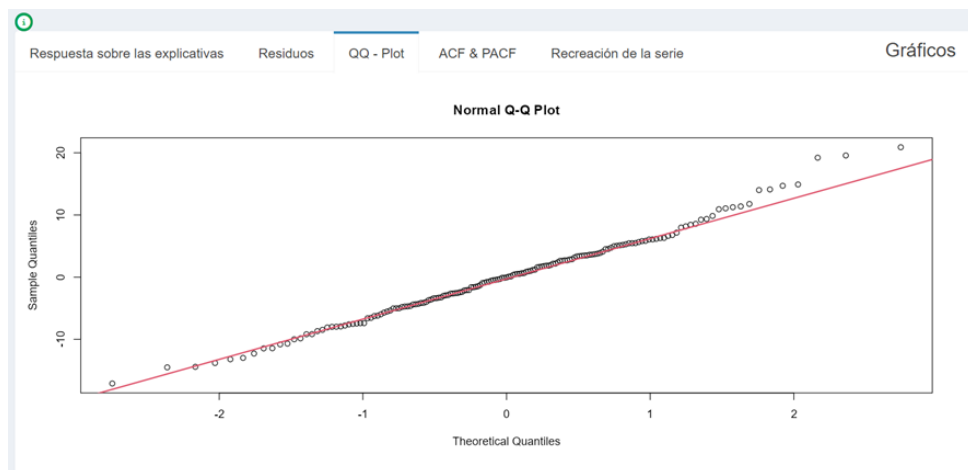


Figura 5.25: Demostración de uso: *Normal Q-Q Plot* de los residuos del modelo MARS

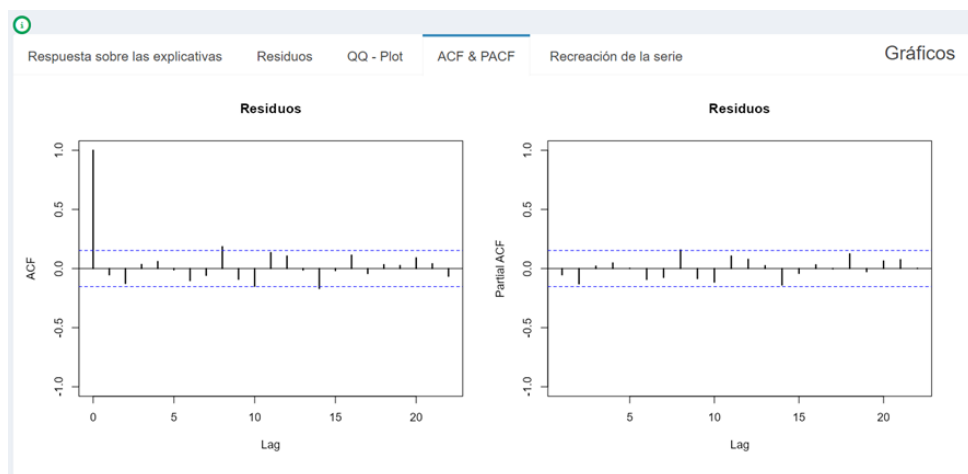


Figura 5.26: Demostración de uso: Gráficos ACF & PACF del modelo MARS

En términos de linealidad y normalidad, consideramos que el modelo es válido, ya que los residuos cumplen con estas presunciones. Por otro lado, al evaluar la independencia mediante los gráficos ACF y PACF, observamos algunos retardos que sobresalen de los intervalos de confianza. Sin embargo, dado que estos retardos se encuentran lejanos, los atribuimos a la aleatoriedad y concluimos que los residuos son independientes.

Recreación de la serie

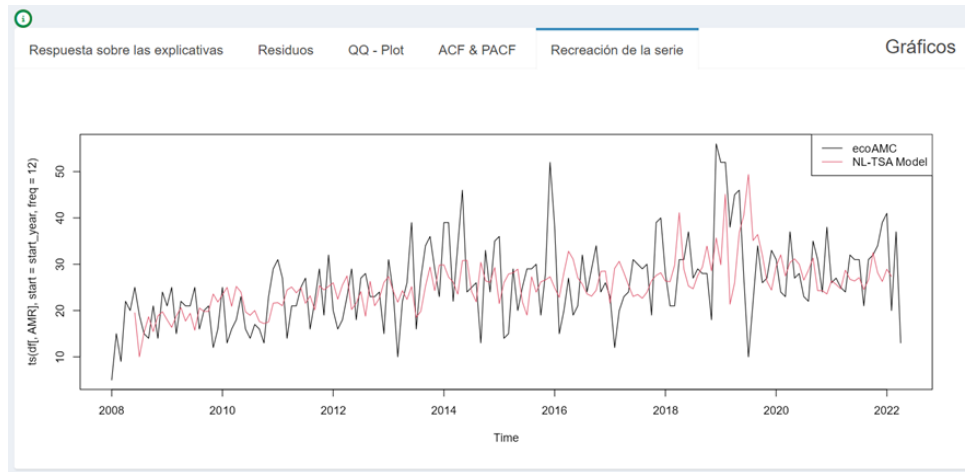


Figura 5.27: Demostración de uso: Gráfico de recreación de la serie modelo MARS

En este gráfico podemos observar las estimaciones del modelo y la serie real, lo cual nos permite visualizar cómo el modelo se ajusta a los datos y capta la variabilidad y tendencia presentes. Aunque este no logra captar completamente toda la variabilidad, sigue adecuadamente la tendencia. Concluimos que es un modelo capaz pero mejorable.

Interpretación de los cortes generados

En un modelo MARS, los cortes son puntos específicos donde las variables explicativas se dividen para crear splines, permitiendo capturar relaciones no lineales en los datos.

En nuestro caso, la variable AMX_5 se corta en 11.369, AMP_2 en 9.66, ecoAMC_1 en 9 y AMC_4 en 169.87. Estos cortes segmentan los datos, permitiendo que el modelo ajuste diferentes pendientes a cada segmento, mejorando así su capacidad explicativa al capturar patrones complejos y no lineales.

A continuación podemos visualizar gráficamente estos segmentos para cada variable AMC, lo que nos permitirá relacionar el comportamiento de las resistencias con los retardos de los consumos introducidos en el modelo.

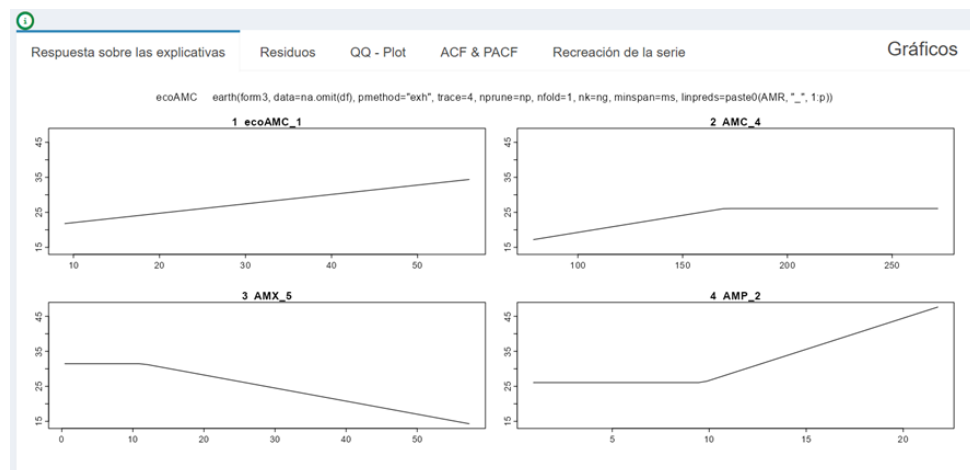


Figura 5.28: Demostración de uso: Gráfico del efecto de las variables explicativas sobre la respuesta

En los gráficos interpretamos los siguientes efectos sobre la variable dependiente para cada variable:

- **AMC_4**: muestra un efecto creciente hasta el punto de corte y se estabiliza posteriormente. En términos de resistencias bacterianas, esto sugiere que la administración de este medicamento aumenta las resistencias hasta cierto punto, después del cual se estancan. A este comportamiento lo llamamos efecto techo.
- **AMX_5**: indica que, a partir del punto de corte, la tendencia previamente estable comienza a disminuir. Esto puede interpretarse como que el uso de este antibiótico reduce las resistencias bacterianas después del corte, aunque nuestro verdadero interés es identificar qué medicamento desacelera las resistencias. A este comportamiento lo llamamos efecto escalonado decreciente
- **AMP_3**: muestra que, a partir del punto de corte, se rompe la estabilidad y comienza un efecto creciente sobre la respuesta, lo que significa que las resistencias se incrementan desde este punto. A este comportamiento lo llamamos efecto umbral.
- **ecoAMC_1**: muestra un efecto lineal, lo cual es esperado dado que esta es la parte autorregresiva del modelo ajustado y así se le ha indicado en el procedimiento.

5.1.8. Conclusiones modelo MARS

Hemos observado que el modelo MARS ajustado es válido; sin embargo, en términos de ajuste, podría capturar mejor la variabilidad de los datos.

En la interpretación de los puntos de corte generados por el modelo, destacamos el efecto umbral de la ampicilina (AMP_3) sobre la variable respuesta. Este efecto indica que el uso de este medicamento provoca un aumento en las resistencias observadas en *Escherichia coli*. Por lo tanto, recomendamos evitar el uso de ampicilina en la mayoría de los tratamientos y ajustar nuevas estrategias de administración de medicamentos en el centro sanitario para mitigar el incremento de resistencias bacterianas.

Capítulo 6

Conclusiones

En el transcurso de este proyecto hemos logrado con éxito los objetivos propuestos. Hemos desarrollado una aplicación Shiny, la cual permite a un experto estadístico llevar a cabo un análisis detallado sobre las resistencias bacterianas observadas en un centro sanitario. Además, hemos reproducido este estudio mediante el uso de la interfaz interactiva.

A continuación presentamos las características esenciales de nuestra aplicación. Estas características demuestran cómo hemos logrado nuestro objetivo principal:

- **Interfaz Interactiva, Amigable y Guiada:** la aplicación cuenta con una interfaz intuitiva y visualmente atractiva, lo cual facilita la navegación y el uso por parte del experto estadístico. Hemos puesto especial énfasis en la facilidad de uso, asegurando que todas las funciones sean accesibles de forma sencilla. Además, hemos insertado puntos de información en toda la interfaz para guiar al usuario a través del proceso, asegurando que se comprenda cada paso y se puedan aprovechar al máximo las funcionalidades ofrecidas.
- **Opciones de Subida de Archivos:** la interfaz permite al usuario subir sus propios archivos de datos, lo que proporciona flexibilidad y la posibilidad de trabajar con datos reales. La aplicación está adaptada para que, en caso de que las columnas sean distintas a las esperadas, esta siga siendo funcional.
- **Herramientas Gráficas Interactivas:** nuestra aplicación web utiliza *Plotly* para la generación de gráficos. Esta es una librería que permite la creación de gráficos interactivos, así, proporcionamos una amplia variedad de funcionalidades al usuario.
- **Modelización:** en la interfaz se han incluido numerosas opciones de modelización, las cuales permiten al usuario ajustar y personalizar los modelos según sus necesidades. Esto incluye una amplia variedad de técnicas estadísticas y herramientas analíticas que son esenciales para el estudio de las resistencias bacterianas.

- **Opción de Descarga:** a modo de complemento hemos hecho posible la descarga de todos los *outputs* del apartado de modelización.
- **Guía de Uso Extensa:** hemos desarrollado una guía de uso detallada, la cual proporciona instrucciones paso a paso sobre cómo utilizar todas las funciones disponibles. Esta guía asegura que incluso aquellos usuarios menos familiarizados con la herramienta puedan utilizarla de manera efectiva.

En resumen, la aplicación creada no solo cumple con el objetivo principal, sino que también ofrece una plataforma robusta y flexible que puede adaptarse a diversas necesidades analíticas. La combinación de una interfaz interactiva, opciones de modelización avanzadas, asistencia visual y una guía de uso completa garantiza que la herramienta sea útil y efectiva para el trabajo de un experto.

Por otro lado, en el marco de nuestro trabajo, también hemos llevado a cabo un estudio sobre resistencias bacterianas utilizando la aplicación. Esto nos ha permitido poner en práctica todos los conocimientos adquiridos a lo largo del grado, además de mostrar un ejemplo del uso del aplicativo, demostrando cómo un experto en estadística puede aprovechar al máximo las posibilidades que ofrece la aplicación.

Los resultados obtenidos en el estudio nos han permitido interpretar los efectos del consumo de agentes antibacterianos sobre las resistencias observadas. En particular, en el caso de estudio sobre *Escherichia coli* hemos identificado que el uso de ampicilina promueve las resistencias observadas en esta bacteria. Por lo tanto, recomendamos al centro médico actualizar la estrategia de suministro de medicamentos, sugiriendo que se disminuya el uso de este medicamento para mitigar las resistencias en la bacteria.

Con la mirada puesta en el futuro, planteamos la creación de una nueva versión de la aplicación. Esta iría dirigida a los médicos e investigadores de un centro sanitario, incluso para aquellos sin conocimientos estadísticos avanzados. El objetivo sería permitirles monitorear las resistencias bacterianas y, en base a los resultados obtenidos, tomar decisiones informadas sobre qué antibióticos utilizar o evitar. Concretamente, planteamos un diseño de aplicación que, a través de preguntas e información proporcionada, permita a los médicos validar modelos predefinidos.

Personalmente, la elaboración de este proyecto ha supuesto un gran reto. El uso de las bibliotecas Shiny y los modelos ARMAX y MARS me ha llevado a descubrir nuevos conceptos y aplicaciones en el ámbito de la estadística.

He tenido que recurrir a mi capacidad de aprendizaje autónomo, apoyándome en la sólida base proporcionada por la formación académica del grado. Esta combinación de autoaprendizaje y conocimientos me ha permitido superar con éxito los objetivos planteados y todos los obstáculos surgidos durante el desarrollo del proyecto.

Esta experiencia ha implicado un enriquecimiento significativo para mi formación profesional, reforzando mi motivación para continuar explorando estas áreas de estudio y desarrollo en el futuro.

Ha sido una valiosa oportunidad para crecer y aprender. Me siento profundamente agradecido por la experiencia y emocionado por las oportunidades que el futuro me puede ofrecer como profesional en el campo de la estadística.

Bibliografía

Brockwell, P. J. (2010). *Introduction to time series and forecasting*. Springer, New York.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67.

Laxminarayan, R., Duse, A., Wattal, C., Zaidi, A. K. M., Wertheim, H. F. L., Sumpradit, N., Vlieghe, E., Hara, G. L., Gould, I. M., Goossens, H., Greko, C., So, A. D., Bigdeli, M., Tomson, G., Woodhouse, W., Ombaka, E., Peralta, A. Q., Qamar, F. N., Mir, F., Kariuki, S., Bhutta, Z. A., Coates, A., Bergstrom, R., Wright, G. D., Brown, E. D., and Cars, O. (2013). Antibiotic resistance—the need for global solutions. *The Lancet Infectious Diseases*, 13(12):1057–1098.

Organización Mundial de la Salud (2019). Sistema mundial de vigilancia de la resistencia a los antimicrobianos: protocolo de implementación temprana para la inclusión de candida spp. Technical documents.

Posit (s. f.). Shiny widget gallery. Disp. en <https://shiny.posit.co/r/gallery/widgets/widget-gallery/>. Cons. 2024-05-21.

R Interface (s. f.). shinydashboardplus. <https://rinterface.github.io/shinydashboardPlus/articles/shinydashboardPlus.html>. Cons. 2024-03-30.

RDocumentation (s. f.). mars.to.earth. <https://www.rdocumentation.org/packages/earth/versions/2.0-4/topics/mars.to.earth>. Cons. 2024-06-05.

RStudio (s. f.a). Shiny. <https://www.rstudio.com/products/shiny/>. Cons. 2024-03-02.

RStudio (s. f.b). shinydashboard. <https://rstudio.github.io/shinydashboard/>. Cons. 2024-03-05.

Stack Overflow (s. f.). Stack overflow. <https://stackoverflow.com/>. Cons. 2024-05-21.

World Health Organization (2023). Antimicrobial resistance. <https://www.who.int/news-room/fact-sheets/detail/antimicrobial-resistance>. Cons. 2024-06-07.

Apéndice A

Guía de Uso

La siguiente guía de uso abarca todos los apartados de la aplicación y describe sus funcionalidades, las cuales permiten realizar un estudio sobre las resistencias microbacterianas.

A continuación recorreremos todas las pestañas presentes en la aplicación.

Carga de ficheros

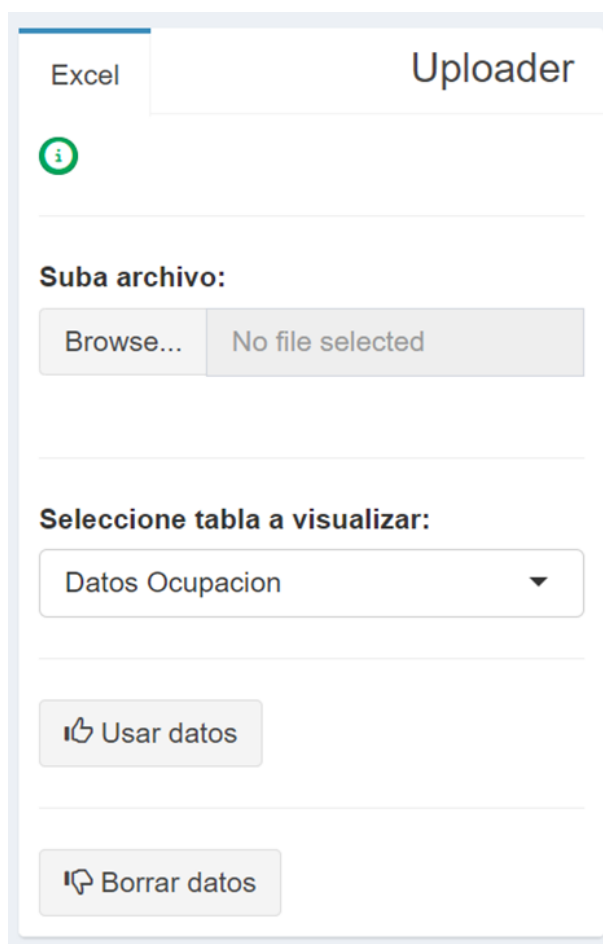
Para iniciar el estudio, primero, es necesario cargar los datos al sistema. Debes subir un archivo Excel y este debe contener las siguientes hojas:

- **Ocupacion:** contiene el nivel de ocupación del hospital cada mes, es decir, la cantidad de pacientes ingresados.
- **ConsumosDDDpor1000Estancias:** incluye el consumo diario definido por cada 1000 estancias de cada medicamento suministrado en el centro.
- **ResultadosSensibles:** por cada tipo de bacteria y medicamento, contiene la cantidad de microorganismos resistentes al tratamiento dado.
- **ResultadosResistentes:** similar a la anterior, pero registra las observaciones de bacterias sensibles.
- **DiccionarioConsumos:** codificaciones de todos los antibióticos.
- **DiccionarioResistencias:** codificaciones de todas las bacterias.

En el repositorio donde se encuentra la aplicación encontrarás un ejemplo de estructura del archivo.

Una vez has subido el archivo puedes seleccionar qué tabla quieres visualizar.

Para finalizar este apartado, si los datos son los deseados, pulsa el botón *Usar datos*. Si después deseas eliminarlos, pulsa el botón *Borrar datos*.



The image shows a web interface for uploading Excel files. At the top, there are two tabs: 'Excel' (active) and 'Uploader'. Below the tabs is a green circular icon with an 'i'. The main section is titled 'Suba archivo:' and contains a 'Browse...' button and a 'No file selected' status. Below this is a section titled 'Seleccione tabla a visualizar:' with a dropdown menu showing 'Datos Ocupacion'. At the bottom, there are two buttons: 'Usar datos' and 'Borrar datos'.

Figura A.1: Guía de uso: Carga de ficheros

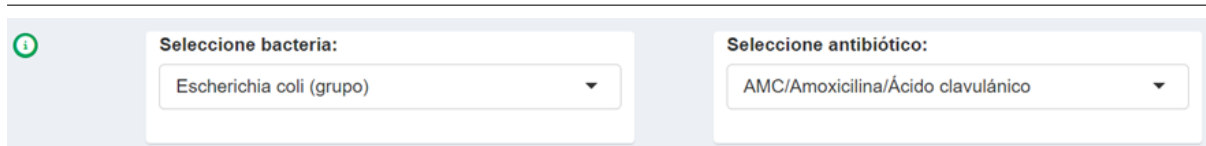
Datos iniciales

En el siguiente apartado se muestran varios descriptivos y gráficos de los datos.

Puedes seleccionar qué observaciones ver según bacteria y antibiótico.

Los listados de selección se generan a partir de los datos introducidos, por lo que estas viendo todos los datos disponibles en tu tabla.

En el caso que no existan observaciones para la combinación seleccionada, estas no se mostrarán en la tabla de descriptivos* y se indicará con un mensaje en los gráficos.



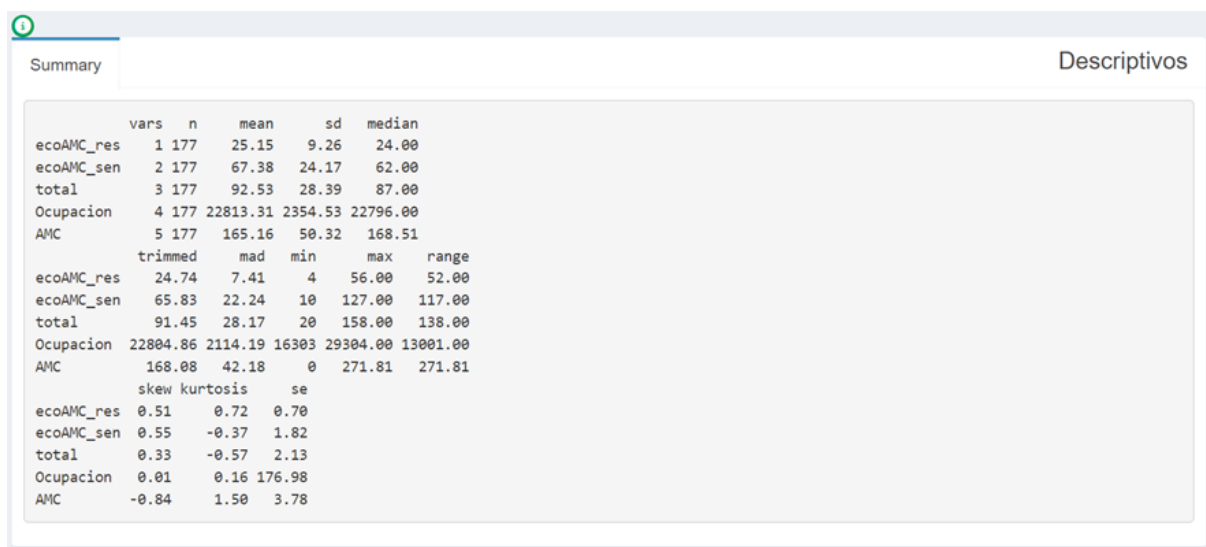
Seleccione bacteria: Escherichia coli (grupo)

Seleccione antibiótico: AMC/Amoxicilina/Ácido clavulánico

Figura A.2: Guía de uso: Seleccionadores

En la tabla de descriptivos visualizarás un resumen de las siguientes variables:

- Cantidad de bacterias resistentes al antibiótico.*
- Cantidad de bacterias sensibles al antibiótico.*
- Cantidad de bacterias en la prueba con ese antibiótico.*
- Consumos DDD por 1000 estancias del antibiótico.
- Ocupación del hospital.



Summary Descriptivos

	vars	n	mean	sd	median
ecoAMC_res	1	177	25.15	9.26	24.00
ecoAMC_sen	2	177	67.38	24.17	62.00
total	3	177	92.53	28.39	87.00
Ocupacion	4	177	22813.31	2354.53	22796.00
AMC	5	177	165.16	50.32	168.51
	trimmed	mad	min	max	range
ecoAMC_res	24.74	7.41	4	56.00	52.00
ecoAMC_sen	65.83	22.24	10	127.00	117.00
total	91.45	28.17	20	158.00	138.00
Ocupacion	22804.86	2114.19	16303	29304.00	13001.00
AMC	168.08	42.18	0	271.81	271.81
	skew	kurtosis	se		
ecoAMC_res	0.51	0.72	0.70		
ecoAMC_sen	0.55	-0.37	1.82		
total	0.33	-0.57	2.13		
Ocupacion	0.01	0.16	176.98		
AMC	-0.84	1.50	3.78		

Figura A.3: Guía de uso: Summary datos iniciales

Después puedes visualizar varios gráficos. Estos son interactivos y ofrecen múltiples opciones, como hacer zoom, seleccionar variables e incluso descargar la figura en formato PNG.

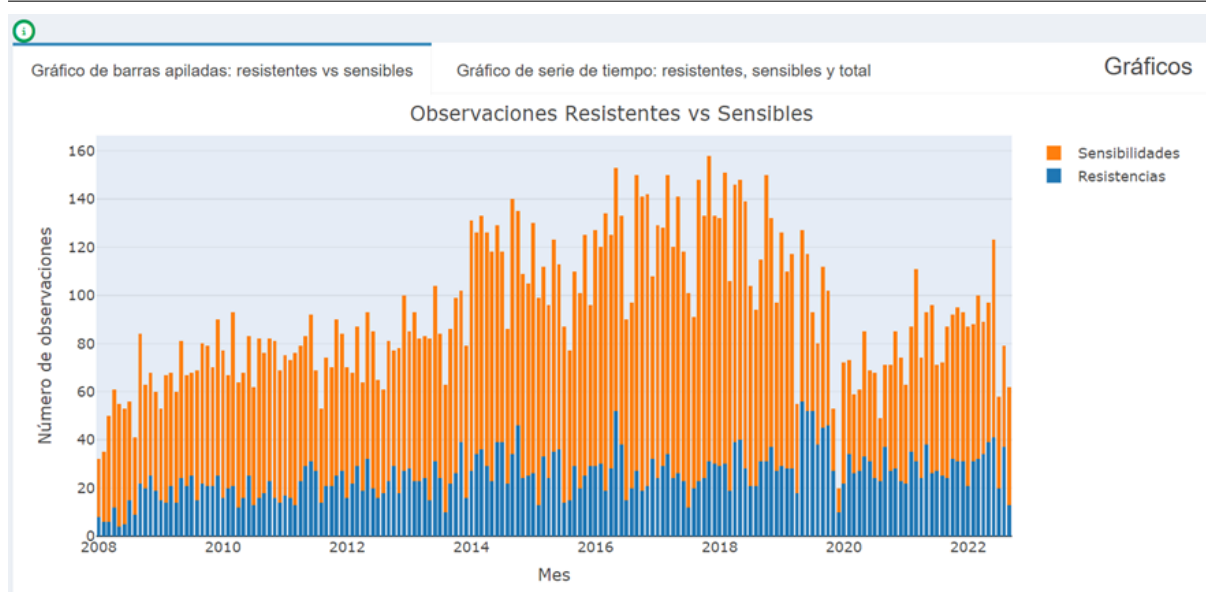


Figura A.4: Guía de uso: Gráficos interactivos

Datos elaborados

En la siguiente pestaña se muestran varios gráficos de variables elaboradas. Estos se crearán a partir de la selección de bacteria y antibiótico que tienes disponible.

Figura A.5: Guía de uso: Seleccionadores

En cuanto a los gráficos mostrados, primero encontramos el porcentaje de bacterias resistentes de la prueba respecto al total y segundo encontramos la tasa de infección por dosis.

Al igual que en la página anterior, puedes interaccionar con los gráficos.

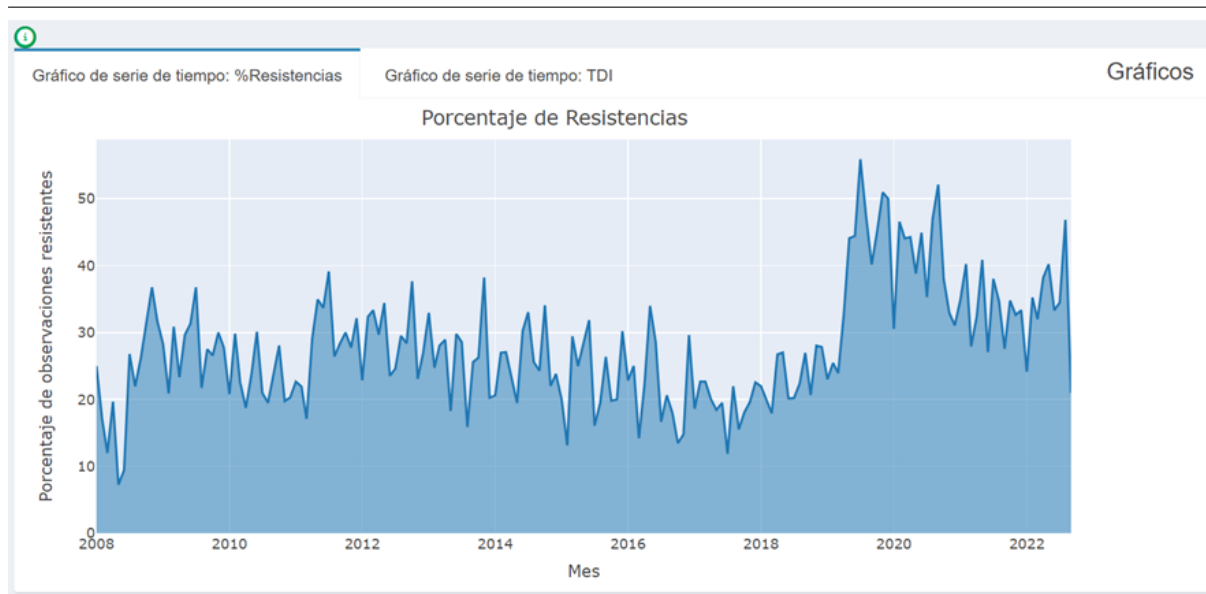


Figura A.6: Guía de uso: Gráficos interactivos

Modelización

En este apartado se elaborarán varios modelos con el objetivo de obtener un modelo MARS válido. El proceso de exploración y modelización consta de las siguientes etapas:

Selección de variables a modelizar

Primero es necesario seleccionar las variables para usar en el modelo. Ten en cuenta que la variable respuesta será la utilizada a lo largo de toda la modelización. Después, se trabajará con los retardos deseados de cada variable para la parte explicativa.

Las variables se pueden seleccionar de dos maneras.

Por un lado, se pueden seleccionar a partir de las columnas presentes en las tablas de consumos y resistencias. Una vez seleccionado el modelo, haz clic en el botón *Usar modelo*.

Selector de variables manual

Selector a partir de fichero

1

Seleccione variable respuesta / AMR:

ecoAMK

Usar modelo

Seleccione variables explicativas / AMC:

☒ AMC
 ☐ AMK
 ☐ AMP
 ☐ AMX
 ☐ AZM
 ☐ AZT
 ☐ CAZ
 ☐ CFM
 ☐ CFR
 ☐ CIP
 ☐ CLI
 ☐ CLO
 ☐ CLR
 ☐ COL

☐ CRO
 ☐ CTR
 ☐ CTX
 ☐ CXM
 ☐ CZO
 ☐ DAP
 ☐ DOX
 ☐ ERY
 ☐ ETP
 ☐ FEP
 ☐ FLU
 ☐ FOS
 ☐ GEN
 ☐ IMP

☐ ITR
 ☐ LNZ
 ☐ LVX
 ☐ MEM
 ☐ MFX
 ☐ MIC
 ☐ MNO
 ☐ MTR
 ☐ NIT
 ☐ NOR
 ☐ PEN
 ☐ POS
 ☐ SFD
 ☐ TEC

☐ TGC
 ☐ TOB
 ☐ TZP
 ☐ VAN
 ☐ VOR
 ☐ AMIN
 ☐ BLIN
 ☐ CARB
 ☐ CEF1
 ☐ CEF2
 ☐ CEF3
 ☐ GLIC
 ☐ QUIN

☐ MACR

Show 5 entries

Search:

	ecoAMK	AMC	AMK	AMP
6	0	85.18	1.99	1.73
7	0	209.69	6.3	0.83
8	1	195.78	2.76	2.51
9	1	223.42	3.52	2.88
10	0	239.52	2.77	2.42

Showing 1 to 5 of 172 entries

Previous

1

2

3

4

5

...

35

Next

Figura A.7: Guía de uso: Selector de variables manual

Por el otro lado, puedes utilizar las variables en modelos predefinidos. Para ello debes cargar un archivo en formato CSV. En este archivo, la primera fila debe contener la variable de respuesta y las filas siguientes deben incluir las variables explicativas, con cada columna representando un modelo. Si el archivo está correctamente formulado y las variables están presentes en la tabla cargada, el sistema lo procesará y mostrará los modelos disponibles. Una vez seleccionado el modelo, haz clic en el botón *Usar modelo*.

De este archivo también hay una muestra en el repositorio.

Selector de variables manual Selector a partir de fichero

Suba archivo:

Browse... Modelos.csv

Upload complete

Seleccione modelo:

ecoAMC

Usar modelo

Show 5 entries Search:

	ecoAMC	AMC	AMX	AMP
6	5	85.18	17.52	1.73
7	15	209.69	34.24	0.83
8	9	195.78	35.67	2.51
9	22	223.42	29.93	2.88
10	20	239.52	53.46	2.42

Showing 1 to 5 of 172 entries Previous 1 2 3 4 5 ... 35 Next

Figura A.8: Guía de uso: Seleccionador a partir de fichero

A continuación podrás ver el gráfico de las series temporales:

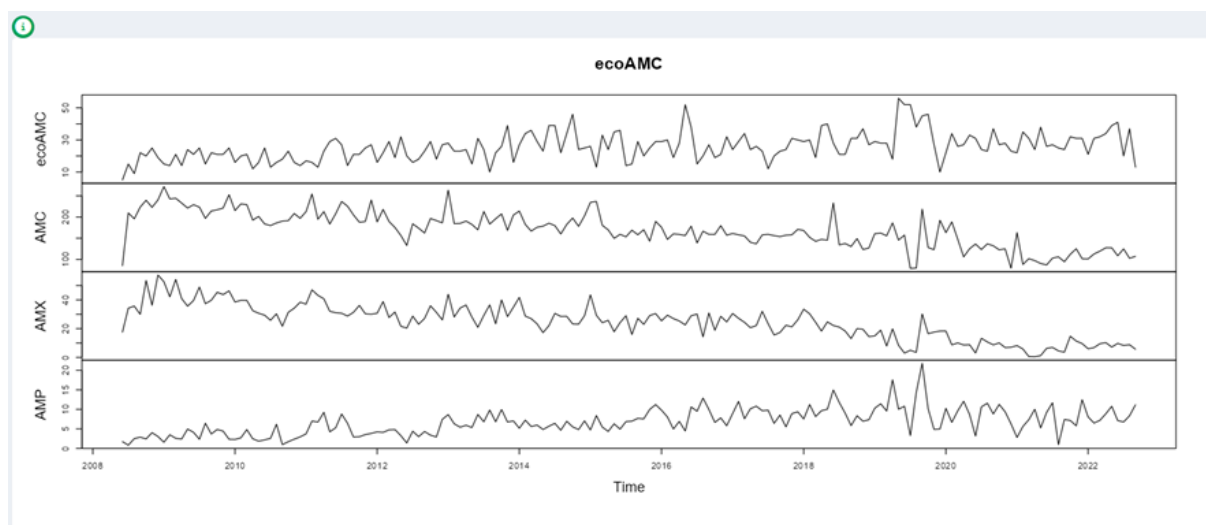


Figura A.9: Guía de uso: Gráfico de series temporales

Exploración preliminar

Así, una vez seleccionadas las variables puedes iniciar la fase de exploración preliminar de modelos.

En esta etapa debes seleccionar la cantidad de retardos AMR y AMC a ajustar en modelos lineales. Ten en cuenta que los retardos seleccionados de las variables AMC serán los utilizados

en el resto del estudio.

Puedes indicar el número de retardos a calcular para cada variable y ajustar los modelos pulsando *Ajustar modelo*. También tienes la opción de quitar y recuperar el intercept del modelo pulsando el botón *intercept*.

The interface is titled 'Generador de modelos a partir de retardos'. It has two tabs: 'Selector de retardos a partir de cifras' (active) and 'Selector de retardos manual'. Under the active tab, there are two input fields. The first is labeled 'Número de retardos para la variable respuesta / AMR:' and contains the number '1'. The second is labeled 'Número de retardos para las variables explicativas / AMC:' and contains the number '5'. To the right of these fields are two buttons: 'Intercept' and 'Ajustar modelo'.

Figura A.10: Guía de uso: Selector de retardos a partir de cifras

Por otro lado, también puedes seleccionar los retardos específicos a calcular y pulsar el botón *Ajustar modelo* cuando tengas la selección deseada. También tienes la opción de eliminar y recuperar el intercept del modelo pulsado el botón *intercept*.

The interface is titled 'Generador de modelos a partir de retardos'. It has two tabs: 'Selector de retardos a partir de cifras' and 'Selector de retardos manual' (active). Under the active tab, there is a section titled 'Retardos disponibles:' followed by a grid of checkboxes. The first row contains ecoAMC_1 through ecoAMC_7. The second row contains ecoAMC_8, AMC_1 through AMC_8. The third row contains AMX_1 through AMX_8. The fourth row contains AMP_2 through AMP_8. To the right of the checkboxes are two buttons: 'Intercept' and 'Ajustar modelo'.

Figura A.11: Guía de uso: Selector de retardos manual

A continuación se ajustan modelos lineales a partir de los retardos seleccionados.

Primero se muestra el modelo completo y segundo se muestra un modelo simplificado.

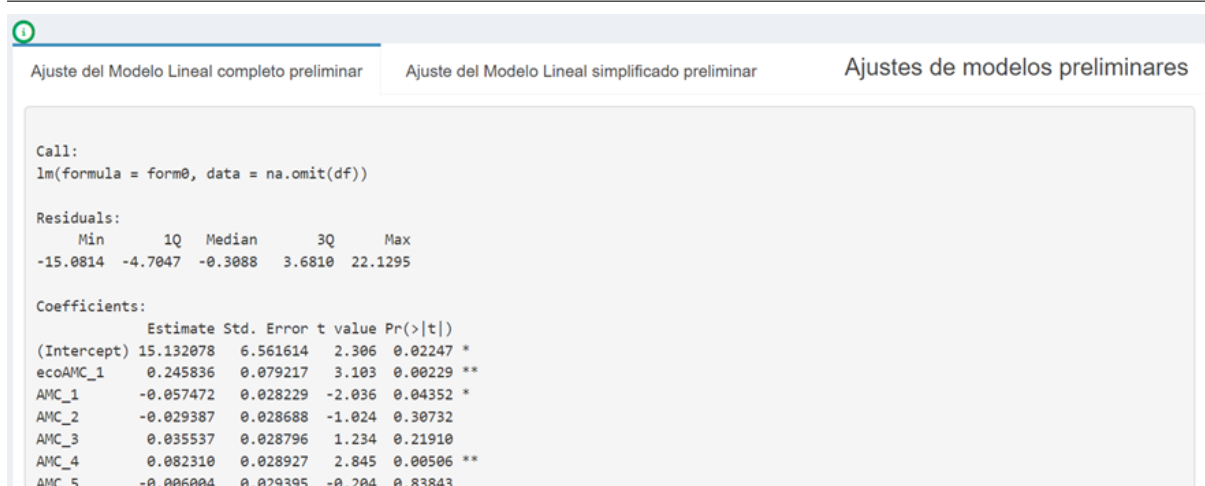


Figura A.12: Guía de uso: Ajuste del Modelo Lineal completo preliminar

El modelo simplificado esta construido mediante el método *stepwise* y puedes escoger el criterio de selección que este usará.

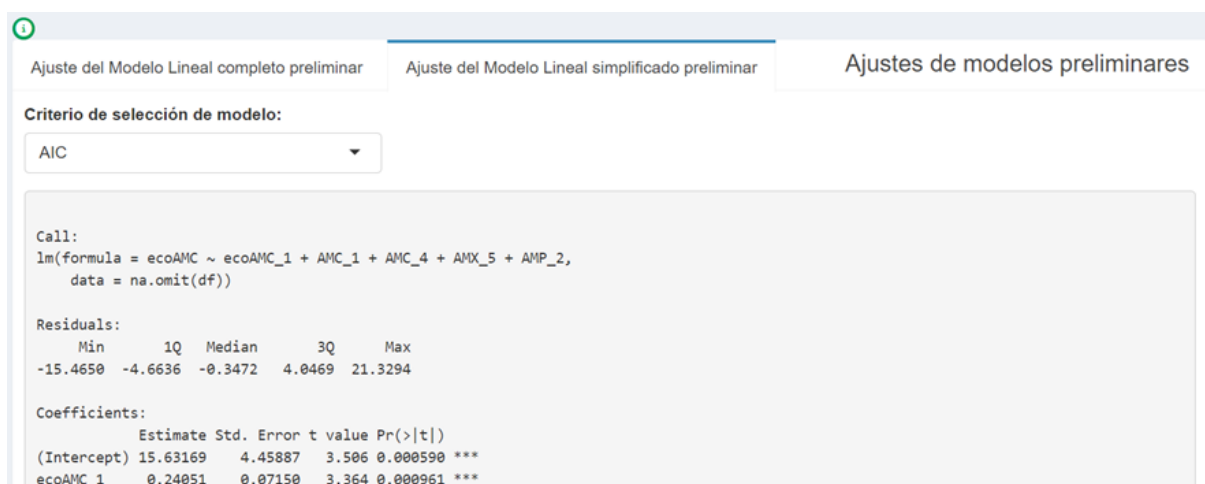


Figura A.13: Guía de uso: Ajuste del Modelo Lineal simplificado preliminar

Por último, puedes validar los modelos mediante los siguientes gráficos:

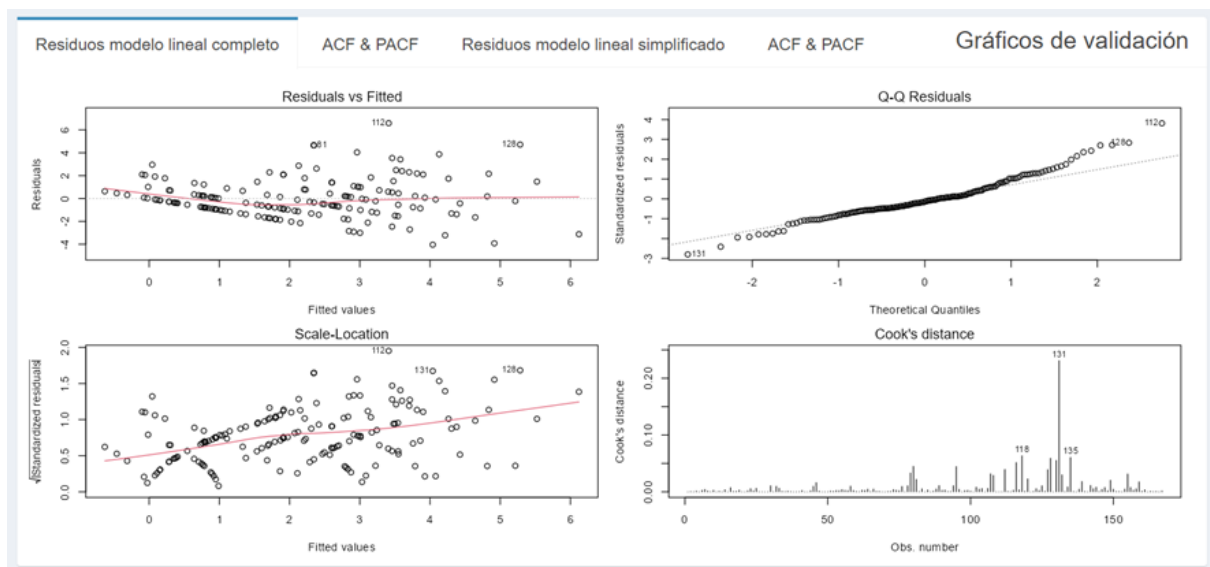


Figura A.14: Guía de uso: Validación modelos preliminares

Selección retardos AMC

Una vez exploradas las variables y sus retardos, a partir de los modelos previamente ajustados, selecciona que retardos quieres usar para el resto del estudio. Puedes seleccionar los del modelo completo o los resultantes del método *stepwise*. Ten en cuenta que estos retardos serán los introducidos en los modelos ARMAX y MARS.

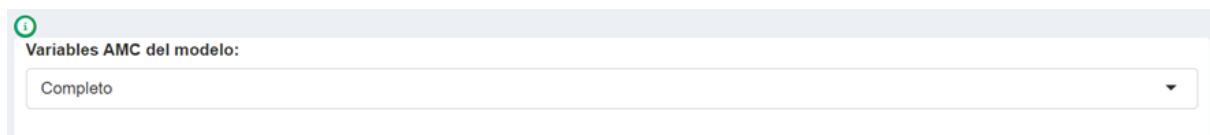


Figura A.15: Guía de uso: Seleccionador de variables AMC

Ajuste del modelo ARMAX

Con el objetivo de conocer que retardos AMR usar se ajusta un modelo ARMAX. Este se construye a partir de la función *auto.arima()*. Por lo tanto, puedes indicar al procedimiento los valores máximos de cada parámetro. Después de esta selección se muestran el modelo ajustado y los t-ratios.

Máximo valor de p

7

Máximo valor de q

0

Máximo valor de d

0

Máximo valor de P

0

Máximo valor de Q

0

Máximo valor de D

0

Series: df[, 1]
Regression with ARIMA(1,0,0) errors

Coefficients:

ar1	AMC_1	AMC_2	AMC_3	AMC_4	AMC_5	AMK_1	AMK_2	AMK_3	AMK_4	AMK_5	AMP_1
0.3048	0.0046	-0.0051	0.0003	-0.0030	0.0010	-0.0659	0.1309	-0.0579	-0.2718	0.0656	0.0892
s.e.	0.0764	0.0056	0.0056	0.0053	0.0055	0.0056	0.1369	0.1444	0.1482	0.1480	0.0513
AMP_2	AMP_3	AMP_4	AMP_5								
0.0283	0.1704	0.1029	0.0287								
s.e.	0.0505	0.0499	0.0500	0.0501							

sigma^2 = 3.055: log likelihood = -324.56
AIC=683.12 AICc=687.23 BIC=736.13

ar1	AMC_1	AMC_2	AMC_3	AMC_4	AMC_5	AMK_1	AMK_2	AMK_3	AMK_4	
t_ratios	3.988605	0.8186489	-0.9022222	0.05082174	-0.5357393	0.1756149	-0.481105	0.9061399	-0.4012194	-1.834223
AMK_5	AMP_1	AMP_2	AMP_3	AMP_4	AMP_5					
t_ratios	0.4432322	1.740471	0.5598767	3.418341	2.060264	0.5720596				

Figura A.16: Guía de uso: Selector de parámetros y ajuste del modelo ARMAX

A continuación se ajustan los modelos lineales a partir de las variables presentes en el modelo ARMAX.

Ajuste del Modelo Lineal completo

Ajuste del Modelo Lineal simplificado

Ajuste de los Modelos Lineales

Call:
lm(formula = form0, data = na.omit(df))

Residuals:

Min	1Q	Median	3Q	Max
-4.0391	-0.9494	-0.2240	0.7470	6.5887

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-3.654e-01	1.253e+00	-0.292	0.77090
ecoAMK_1	3.151e-01	7.679e-02	4.103	6.68e-05 ***
AMC_1	4.768e-03	6.084e-03	0.784	0.43450
AMC_2	-5.996e-03	6.315e-03	-0.949	0.34391
AMC_3	2.400e-03	6.181e-03	0.388	0.69837

Figura A.17: Guía de uso: Ajuste del modelo lineal completo y simplificado

Después puedes validarlos gráficamente:

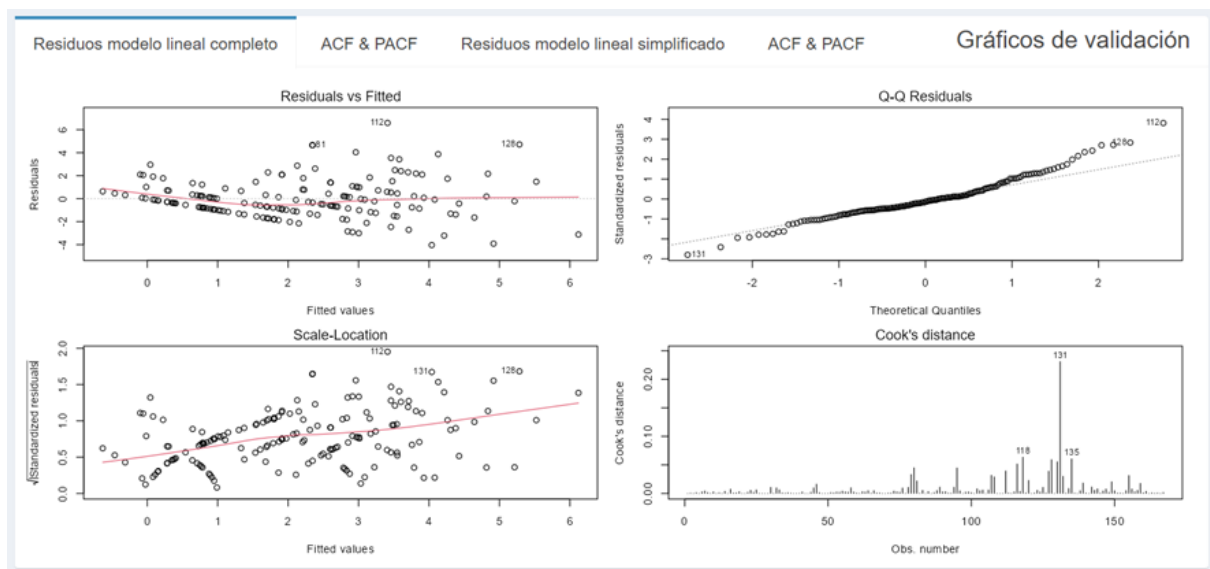


Figura A.18: Guía de uso: Validación modelos ARMAX

Ajuste del modelo MARS

Una vez seleccionadas todas las variables para la parte explicativa se modeliza un MARS.

Primero se muestra el output del procedimiento:

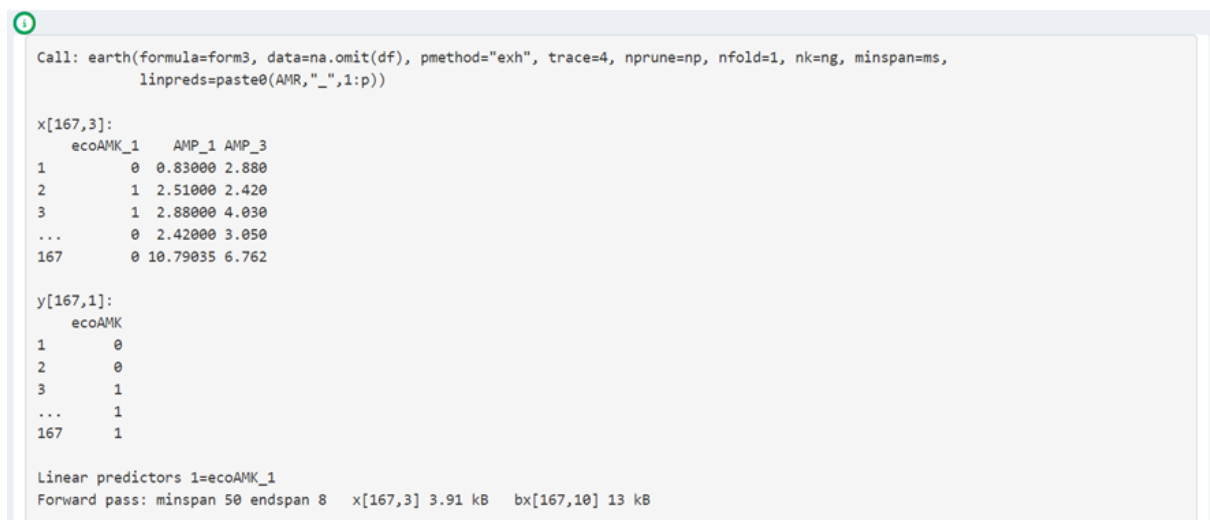


Figura A.19: Guía de uso: Output del modelo MARS

Y después se muestran varios gráficos de utilidad para validar e interpretar el modelo.

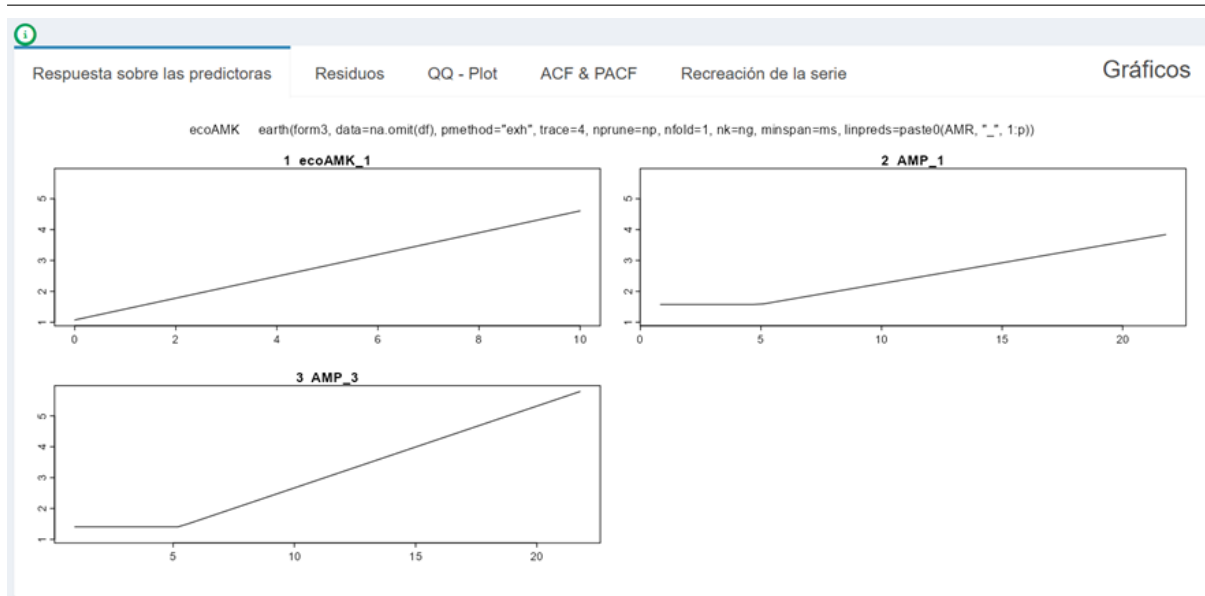


Figura A.20: Guía de uso: Validación e Interpretación modelo MARS

Descarga de outputs

Por último, pulsando el siguiente botón puedes descargar todos los outputs en formato PDF.

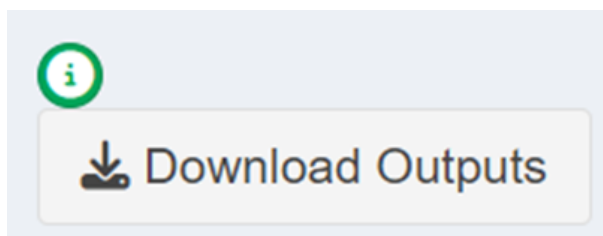


Figura A.21: Guía de uso: Botón de descarga

Con esto ya habríamos finalizado el ajuste y desarrollo de modelos para el estudio.

Apéndice B

Gráficos de los modelos ajustados

A continuación se presentan diversas figuras que ilustran el ajuste y validación de los modelos empleados en el estudio que no aparecen en el cuerpo del trabajo.

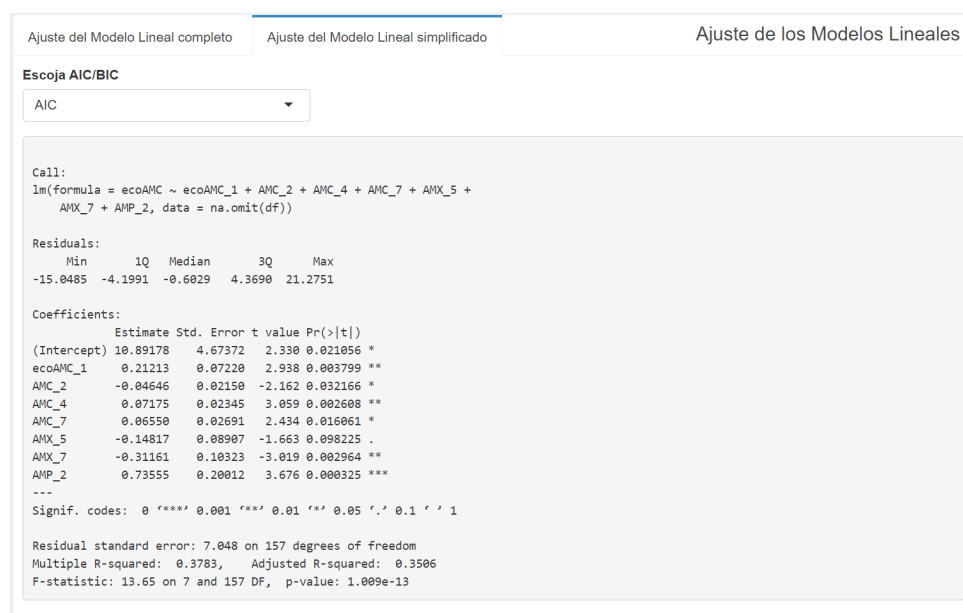


Figura B.1: Demostración de uso: Ajuste del modelo lineal simple ARMAX

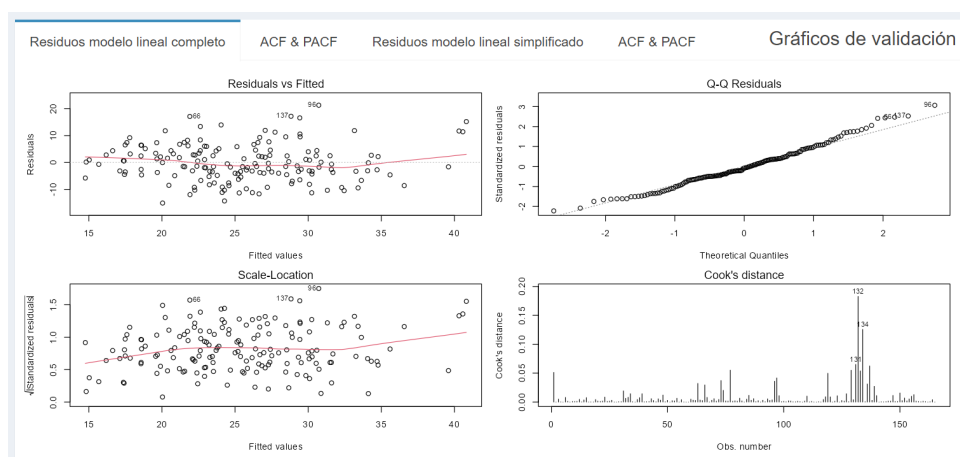


Figura B.2: Demostración de uso: Gráficos de validación del modelo lineal completo ARMAX

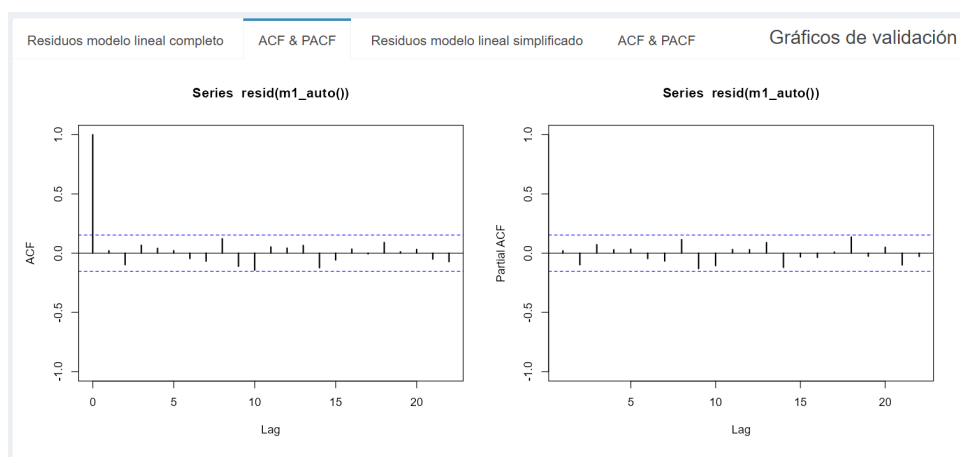


Figura B.3: Demostración de uso: Gráficos ACF & PACF del modelo lineal completo ARMAX

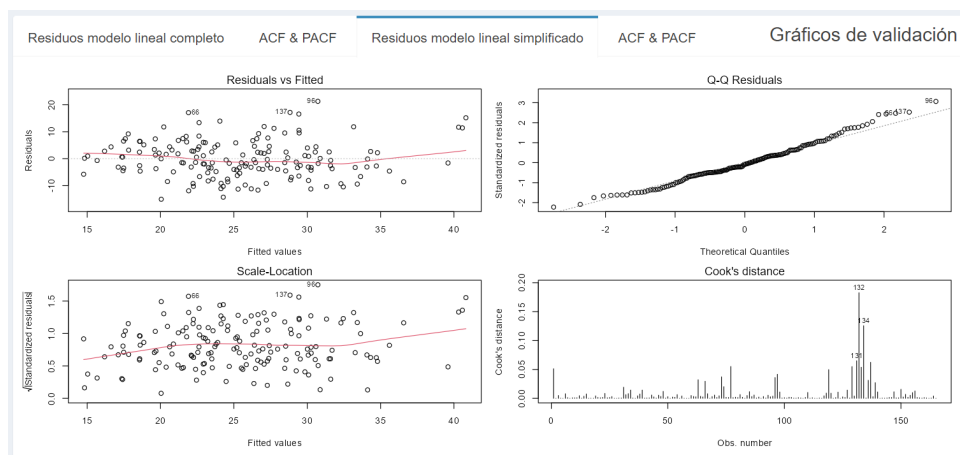


Figura B.4: Demostración de uso: Gráficos de validación del modelo lineal simple ARMAX

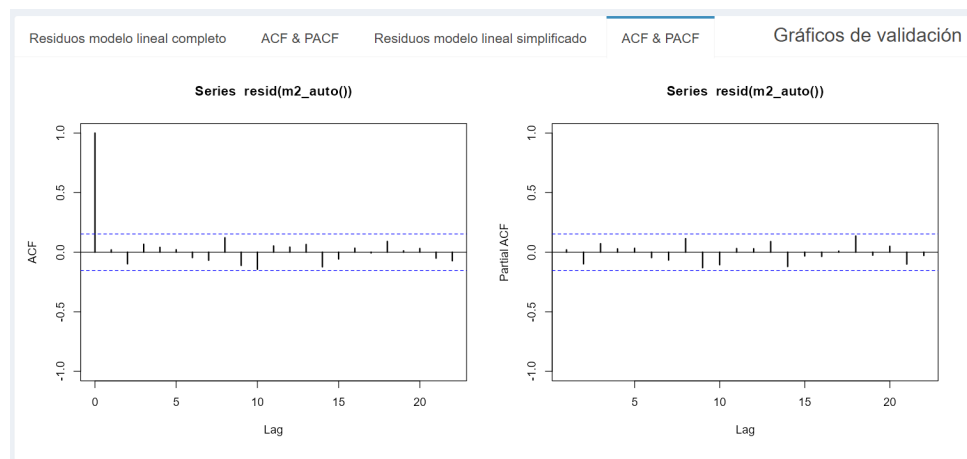


Figura B.5: Demostración de uso: Gráficos ACF & PACF del modelo lineal simplificado AR-MAX

Apéndice C

Código R: Shiny APP M.R.B.A.

```
1 ## ui.R ##
2 library(shiny)
3 library(shinydashboard)
4 library(shinydashboardPlus)
5 library(plotly)
6 library(tidyquant)
7 library(openxlsx)
8 library(tidyverse)
9 library(DT)
10 library(readxl)
11 library(forecast)
12 library(dplyr)
13 library(mgcv)
14 library(car)
15 library(urca)
16 library(earth)
17 library(shinyWidgets)
18 library(rmarkdown)
19 library(grid)
20 library(gridExtra)
21
22
23
24 dashboardPage(
25   skin = "green",
26   # md=TRUE,
```

```

27 dashboardHeader(title = "Shiny App: M.R.B.A."),
28 dashboardSidebar(
29   sidebarMenu(
30     menuItem("Inicio", tabName = "tab_inicio", icon = icon("home",
31       lib = "glyphicon")),
32     menuItem("Carga de Ficheros", tabName = "tab_uploader", icon =
33       icon("upload")),
34     menuItem("Datos Iniciales", tabName = "tab_datos_iniciales",
35       icon = icon("th-large")),
36     menuItem("Datos Elaborados", tabName = "tab_datos_elaborados",
37       icon = icon("th")),
38     menuItem("Modelización", tabName = "tab_modelos", icon = icon("
39       signal"))
40   )
41 ),
42 dashboardBody(
43   tags$head(
44     tags$style(HTML("
45     .box {
46       border-top: 10px !important; /* Remove the top border */
47       margin-bottom: 10px; /* Adjust margin for spacing */
48     }
49     .box-header {
50       display: none; /* Remove the header bar */
51     }
52     .box-body {
53       padding: 10px 10px !important; /* Reduce the padding */
54     }
55     .box.box-solid .box-body {
56       padding-top: 0px !important; /* Reduce top padding for
57         solid header boxes */
58     }
59     .nav-tabs-custom > .tab-content {
60       padding-top: 5px !important; /* Reduce top padding for
61         tabBox content */
62     }
63     .nav-tabs-custom .tab-pane {
64       padding-top: 5px !important; /* Reduce top padding inside
65         each tab panel */

```

```

58     }
59     .nav-tabs-custom .box-body {
60         padding-top: 5px !important; /* Additional adjustment for
           box-body within tabBox */
61     }
62     .multicol {
63         display: flex;
64         flex-wrap: wrap;
65     }
66     .multicol .checkbox {
67         flex: 1 1 30%;
68         margin-bottom: 5px;
69     }
70     .pretty.p-default { margin-right: 15px; }
71
72 "
73     ))
74 ),
75 tabItems(
76
77     tabItem(
78         tabName = "tab_inicio",
79         fluidRow(h2(". BIENVENIDO"),
80             tabBox(
81                 title="",
82                 id="tab_box_inicio",
83                 width=12,
84                 tabPanel(title="Inicio",
85                     fluidRow(
86                         column(
87                             12,
88                             p( "Bienvenido a la plataforma interactiva ", em("M
                               .R.B.A."),
89                             " (Modelizador de Resistencias Bacterianas ante
                               tratamientos Antibióticos).",
90                             style = "font-size:16px; text-align:justify"),
91                             p( "En esta podrás elaborar un estudio estadístico
                               sobre el comportamiento de las resistencias
                               bacterianas a través del ajuste de modelos.",

```

```

92       style = "font-size:16px; text-align:justify" ),
93     p("Esta es una aplicación flexible y adaptable a
        tus datos, además de permitir un sinfín de
        opciones en el ajuste de modelos.",
94       style = "font-size:16px; text-align:justify" ),
95     p("Para conocer más sobre el uso de esta dirígete a
        la pestaña ", em("Guía de uso."),
96       style = "font-size:16px; text-align:justify" ))))
97     ,
98   tabPanel(title="Guía de uso",
99     fluidRow(
100       column(12,
101         p("A lo largo de todas las pestañas
            encontrarás puntos de información,
102         los cuales te guiarán en la realizaci
            ón de tu estudio mediante el uso
            de
103         la aplicación",
104         style="font-size:16px; text-align:
            justify"),
105         p("Otras funcionalidades a tener en
            cuenta son:",
106         style="font-size:16px; text-align:
            justify"),
107         p("En la primera pestaña", em("Carga de
            ficheros"), "es necesario que los
108         datos cargados estén en formato Excel
            . El documento debe contener las
            siguientes
109         hojas: Ocupación,
            ConsumosDDDpor1000Estancias,
            ResultadosSensibles,
110         ResultadosResistentes,
            DiccionarioConsumos,
            DiccionarioResistencias.
111         Esto es necesario ya que el servidor
            de la aplicación se alimenta de la

```



```
style="font-size:16px; text-align:
  justify"),
p("Mi objetivo personal en la elaboraci
  ón del trabajo final de grado
  siempre ha sido aportar un proyecto ú
  til y de valor.",
  style="font-size:16px; text-align:
  justify"),
p("En la elaboración de esta aplicación
  me he dado cuenta del gran
  abanico de posibilidades y
  oportunidades que la estadística
  me
  brinda, viendo cómo mediante la
  aplicación de mis estudios y el
  autoaprendizaje soy capaz de crear
  una herramienta que facilita el
  estudio estadístico a profesionales."
  ,
  style="font-size:16px; text-align:
  justify"),
p("Durante la realización del estudio
  sobre resistencias bacterianas,
  pude retarme y demostrar mis dotes
  como futuro profesional en el
  ámbito de la estadística. Ver cómo,
  gracias a la estadística, somos
  capaces de comprender mejor nuestro
  entorno y solucionar problemas
  me enamora más de mis estudios.",
  style="font-size:16px; text-align:
  justify"),
p("Me siento totalmente agradecido de
  poder dar mi pequeña aportación
  desde el mundo de la estadística.",
  style="font-size:16px; text-align:
  justify")
```

```
)))
```

```

152         )
153     )
154 ),
155 ##### SUBIDA DE FICHEROS #####
156
157 tabItem(
158     tabName = "tab_uploader",
159     fluidRow(
160         column(width=4,
161             tabBox(
162                 title = "Uploader",
163                 id = "tab_box_uploader",
164                 width = 12,
165                 tabPanel(
166                     title = "Excel",
167                     dropdownButton(
168                         p("A continuación puedes subir el fichero con
169                             la estructura y formatos
170                             indicados en la guía de uso.",br(), "Después
171                                 , de las tablas cargadas, puedes
172                                 seleccionar cuál quieres visualizar.",br(), "
173                                     Para finalizar, si los datos son los
174                                     deseados, pulsa el botón
175                                     ",em("Usar datos."),br(),"Si deseas
176                                         eliminar los datos subidos al sistema,
177                                         pulsa el botón",em("Borrar datos."),
178                                         style="text-align: left"),
179                     circle = TRUE,
180                     status = "success",
181                     size = "xs",
182                     icon = icon("circle-info"),
183                     label = NULL,
184                     tooltip = FALSE,
185                     right = F,
186                     up = FALSE,
187                     width = NULL,
188                     margin = "10px",
189                     inline = TRUE,
190                     inputId = NULL

```

```

186     ),
187     hr(),
188     fileInput(
189       inputId = "file_input_excel",
190       label = "Suba archivo:",
191       accept = c('.xlsx', 'application/vnd.
           openxmlformats-officedocument.spreadsheetml
           .sheet')
192     ),
193     hr(),
194     selectInput("select_input_table_visual", "
           Seleccione tabla a visualizar:",
195               choices = c("Datos Ocupacion"="tabla
                           _ocupacion", "Datos Consumos"="
                           tabla_consumos", "Datos
                           Resistentes"="tabla_resistentes",
                           "Datos Sensibles"="tabla_
                           sensibles", "Diccionario Consumos
                           "="tabla_dicc_cons", "Diccionario
                           Resistencias" = "tabla_dicc_res")
                           )
196     ,hr(),
197     actionButton(inputId = 'action_button_excel_
           ready', label = 'Usar datos', icon = icon('
           thumbs-up', lib = 'glyphicon')),
198     uiOutput('check_excel'),
199     hr(),
200     actionButton(inputId = "action_button_excel_not_
           ready", label = "Borrar datos", icon = icon('
           thumbs-down', lib = 'glyphicon'))
201   )
202   )),
203   column(width=8, DT::dataTableOutput('tabla_output_visual'))
204 )
205 )
206 ,
207
208 ### PLOTS NORMALES ####
209 tabItem(

```

```

210     tabName = "tab_datos_iniciales",
211     fluidRow(
212         column(width=1, dropdownButton(
213             p("A continuación se muestran varios descriptivos y grá
                ficos.", br(),
214                 "Puedes seleccionar qué observaciones ver según
                bacteria y antibiótico.", br()
215                 , "Si no hay observaciones para la combinación
                seleccionada, estas no
216                 se mostrarán en la tabla de descriptivos* y se indicará
                con un mensaje en los gráficos.",
217                 style="text-align: left"),
218             circle = TRUE,
219             status = "success",
220             size = "xs",
221             icon = icon("circle-info"),
222             label = NULL,
223             tooltip = FALSE,
224             right = F,
225             up = FALSE,
226             width = NULL,
227             margin = "10px",
228             inline = TRUE,
229             inputId = NULL
230         )),
231         column(width=5, box(width = 12,
232             uiOutput("output_select_input_micro_data_inicial")
233         )),
234
235         column(width=5, box(width=12,
236             uiOutput("output_select_input_med_data_inicial")
237         )),
238         column(width=12, dropdownButton(
239             p("En la próxima tabla de descriptivos encontrarás las
                siguientes variables:", br(),
240                 "1: Cantidad de bacterias resistentes al antibiótico.*"
                , br(),
241                 "2: Cantidad de bacterias sensibles al antibiótico.*",
                br(),

```

```

242     "3: Cantidad de bacterias en la prueba con ese antibió
        tico.*",br(),
243     "4: Consumos DDD por 1000 estancias del antibiótico.",
        br(),
244     "5: Ocupación del hospital.",br(),
245     style="text-align: left"),
246 circle = TRUE,
247 status = "success",
248 size = "xs",
249 icon = icon("circle-info"),
250 label = NULL,
251 tooltip = FALSE,
252 right = F,
253 up = FALSE,
254 width = NULL,
255 margin = "10px",
256 inline = TRUE,
257 inputId = NULL
258 ),),
259 tabBox(title="Descriptivos",id="tab_box_data_ini_sum",width
        =12,
260         tabPanel("Summary",
261
262
263
264             verbatimTextOutput("summary_data_ini"))),
265
266 column(width=12,dropdownButton(
267     p("Puedes interaccionar con los gráficos a continuación."
        ,
268         style="text-align: left"),
269     circle = TRUE,
270     status = "success",
271     size = "xs",
272     icon = icon("circle-info"),
273     label = NULL,
274     tooltip = FALSE,
275     right = F,
276     up = FALSE,

```

```

277     width = NULL,
278     margin = "10px",
279     inline = TRUE,
280     inputId = NULL
281   )),
282
283   tabBox(title="Gráficos", id="tab_box_data_inicial_1",width
284         = 12,
285         tabPanel("Gráfico de barras apiladas: resistentes vs
286                   sensibles",plotlyOutput("plot_barras_apiladas"))
287         ,
288         tabPanel("Gráfico de serie de tiempo: resistentes,
289                   sensibles y total",plotlyOutput("plot_line_totals
290                   "))
291       )
292   ,
293   tabBox(title="Gráficos", id="tab_box_plots_inicial_2",width
294         = 12,
295         tabPanel("Gráfico de serie de tiempo: consumos de
296                   antibiótico",plotlyOutput("plot_serie_consumos"))
297         ,
298         tabPanel("Gráfico de serie de tiempo:
299                   hospitalizaciones",plotlyOutput("plot_
300                   hospitalizaciones"))
301       )
302   )
303   ,
304   tabItem(
305     tabName = "tab_datos_elaborados",
306     fluidRow(
307
308       column(width=1,dropdownButton(
309         p("A continuación se muestran varios gráficos de
310           variables elaboradas.",br(),
311           "Estos se crearán a partir de la selección de bacteria
312           y antibiótico que tienes disponible.",br(),

```

```

304     "Primero encontramos el porcentaje de bacterias
305         resistentes de la prueba respecto el total.",br(),
306     "Segundo encontramos la tasa de infección por dosis.",
307     style="text-align: left"),
308     circle = TRUE,
309     status = "success",
310     size = "xs",
311     icon = icon("circle-info"),
312     label = NULL,
313     tooltip = FALSE,
314     right = F,
315     up = FALSE,
316     width = NULL,
317     margin = "10px",
318     inline = TRUE,
319     inputId = NULL
320 ),
321 column(width=5,box(width = 12,
322     uiOutput("output_select_input_micro_data_
323         _elaborada")
324 ),
325     column(width=5,box(width=12,
326         uiOutput("output_select_input_med_data_
327             elaborada")
328     ),
329     column(width=12,dropdownButton(
330         p("Puedes interaccionar con los gráficos a continuación."
331         ,
332         style="text-align: left"),
333         circle = TRUE,
334         status = "success",
335         size = "xs",
336         icon = icon("circle-info"),
337         label = NULL,
338         tooltip = FALSE,
339         right = F,
340         up = FALSE,

```

```

339         width = NULL,
340         margin = "10px",
341         inline = TRUE,
342         inputId = NULL
343     )),
344
345
346     tabBox(title="Gráficos", id="tab_box_plots_elaborados",
347           width = 12,
348             tabPanel("Gráfico de serie de tiempo: %Resistencias"
349                   ,plotlyOutput("plot_porcentaje_resistencias")),
350             tabPanel("Gráfico de serie de tiempo: TDI",
351                   plotlyOutput("plot_TDI"))
352         )
353
354
355
356 ),
357 tabItem(
358     tabName="tab_modelos",
359     fluidRow(
360         h2(" · MODELIZACIÓN"),
361         h3(" · Selector de variables"),
362         column(width=12,dropdownButton(
363             p("Primero selecciona las variables de interés para tu
364               modelo.",br(),"Ten en cuenta que la variable
365               respuesta será la utilizada a lo largo de toda la
366               modelización. Después iremos trabajando con
367               los retardos deseados de cada variable para la parte
368               explicativa",
369               style="text-align: left"),
370             circle = TRUE,
371             status = "success",
372             size = "xs",
373             icon = icon("circle-info"),
374             label = NULL,

```

```

372     tooltip = FALSE,
373     right = F,
374     up = FALSE,
375     width = NULL,
376     margin = "10px",
377     inline = TRUE,
378     inputId = NULL
379  )),
380
381   tabBox(width=12,id="tab_box_selector_modelo",
382         tabPanel(title="Selector de variables manual",
383                 fluidRow(
384                   column(width=12,dropdownButton(
385                     p("A partir de las columnas presentes en
386                       las tablas de consumos y resistencias,
387                       puedes seleccionar tus variables de inter
388                         és para modelizar.",br(),
389                       "Una vez seleccionadas pulsa el botón",
390                       em("Usar modelo."),
391                       style="text-align: left"),
392                     circle = TRUE,
393                     status = "success",
394                     size = "xs",
395                     icon = icon("circle-info"),
396                     label = NULL,
397                     tooltip = FALSE,
398                     right = F,
399                     up = FALSE,
400                     width = NULL,
401                     margin = "10px",
402                     inline = TRUE,
403                     inputId = NULL
404                   )),
405                   column(width=6,uiOutput("output_select_
406                     input_modelo_2"))
407                 ),
408                 column(width=6,actionButton(inputId = 'action_
409                   _button_models_2_ready', label = 'Usar
410                   modelo', icon = icon('thumbs-up', lib = '

```

```

        glyphicon'))),
        uiOutput('check_models_2')),
    hr(),
    column(width=12, uiOutput("check_box_output_
        modelos")),
    hr(),
    box(width=12, DT::dataTableOutput('tabla_
        prueba_2'))))
),
tabPanel(title="Selector a partir de fichero", fluidRow
(
    column(width=12, dropdownButton(
p("En caso de querer usar modelos ya predefinidos
    puedes usar este apartado.", br(),
    "Sube un archivo con la estructura y formatos
        indicados en la guía de uso, el
    sistema se alimentará de este y te mostrará los
        modelos disponibles.", br(),
    "Una vez seleccionado el modelo deseado pulsa el
        botón", em("Usar modelo."),
    style="text-align: left"),
    circle = TRUE,
    status = "success",
    size = "xs",
    icon = icon("circle-info"),
    label = NULL,
    tooltip = FALSE,
    right = F,
    up = FALSE,
    width = NULL,
    margin = "10px",
    inline = TRUE,
    inputId = NULL
    )),
    column(width=4,
        fileInput(
            inputId = "file_input_models",
            label = "Suba archivo:",
            accept = '.csv'

```

```

436         ),
437         uiOutput("output_select_input_modelo_1"),
438         # hr(),
439         actionButton(inputId = 'action_button_models_1
         _ready', label = 'Usar modelo', icon = icon
         ('thumbs-up', lib = 'glyphicon')),
440
441         uiOutput('check_models_1')
442         ),
443         column(width=8,
444             DT::dataTableOutput('tabla_prueba_1'))
445     )),
446
447     h3(" · Serie temporal"),
448     column(width=12, dropdownButton(
449         p("Se muestra el gráfico de serie temporal de las variables
         seleccionadas.",
450             style="text-align: left"),
451         circle = TRUE,
452         status = "success",
453         size = "xs",
454         icon = icon("circle-info"),
455         label = NULL,
456         tooltip = FALSE,
457         right = F,
458         up = FALSE,
459         width = NULL,
460         margin = "10px",
461         inline = TRUE,
462         inputId = NULL
463     )),
464     box(width=12, plotOutput("plot_ts")),
465     h3(" · Exploración preliminar de modelos"),
466     column(width=12, dropdownButton(
467         p("A continuación selecciona la cantidad de retardos AMR y
         AMC a ajustar en modelos lineales.",br(),
468             "Ten en cuenta que los retardos seleccionados de las
         variables AMC serán los utilizados en el resto del
         estudio.",br(),

```

```

469         "Puedes indicar la cantidad de retardos a calcular, o
          bien hacer una
470         selección específica de los deseados.",
471         style="text-align: left"),
472     circle = TRUE,
473     status = "success",
474     size = "xs",
475     icon = icon("circle-info"),
476     label = NULL,
477     tooltip = FALSE,
478     right = F,
479     up = FALSE,
480     width = NULL,
481     margin = "10px",
482     inline = TRUE,
483     inputId = NULL
484 ),
485     tabBox(title="Generador de modelos a partir de retardos",
486           width=12, id="tab_box_seleccion_retardos",
487           tabPanel("Selector de retardos a partir de cifras",
488                 fluidRow(
489                     column(width=12, dropdownButton(
490                         p("Indica el número de retardos a
491                           calcular para cada variable y ajusta
492                           los modelos pulsando", em("Ajustar
493                             modelo."), br(),
494                           "Tienes la opción de quitar y recuperar
495                           el intercept del modelo pulsando el
496                           botón", em("intercept."),
497                           style="text-align: left"),
498                         circle = TRUE,
499                         status = "success",
500                         size = "xs",
                    icon = icon("circle-info"),
                    label = NULL,
                    tooltip = FALSE,
                    right = F,
                    up = FALSE,
                    width = NULL,

```

```

501         margin = "10px",
502         inline = TRUE,
503         inputId = NULL
504     )),
505     column(width=5, numericInput(inputId="
        numeric_input_AMR", label="Número de
        retardos para la variable respuesta /
        AMR:", value=1, min=0, max=10)),
506     column(width=5, numericInput(inputId="
        numeric_input_AMC", label="Número de
        retardos para las variables
        explicativas / AMC:", value=5, min=0,
        max=10)),
507     column(width=2,
508
509         actionButton(inputId = 'action_
        button_intercept_no_1', label =
        'Intercept'),
510
511         hr(),
512         actionButton(inputId="usar_
        retardos_1", label = 'Ajustar
        modelo')
512     )
513 )),
514 tabPanel("Selector de retardos manual", fluidRow(
    column(width=10, column(width=12, dropdownButton(
515        p("Selecciona los retardos específicos a calcular.
        ", br(),
516        "Pulsa el botón", em("Ajustar modelo"), "cuando
        tengas la selección deseada.", br(),
517        "También tienes la opcion de eliminar y
        recuperar el intercept del modelo pulsado el
        botón", em("intercept."),
518        style="text-align: left"),
519        circle = TRUE,
520        status = "success",
521        size = "xs",
522        icon = icon("circle-info"),
523        label = NULL,

```

```

524         tooltip = FALSE,
525         right = F,
526         up = FALSE,
527         width = NULL,
528         margin = "10px",
529         inline = TRUE,
530         inputId = NULL
531     )),
532     uiOutput("output_checkbox_retardos")),column(width
        =2,actionButton(inputId = 'action_button_
            intercept_no_2', label = 'Intercept'),hr(),
            actionButton(inputId="usar_retardos_2",label = '
                Ajustar modelo')))))
533     ),
534     column(width=12,dropdownButton(
535         p("Se ajustan modelos lineales a partir de los retardos
            seleccionados.",br(),
536             "Primero se muestra el modelo completo y segundo se
            muestra un modelo simplificado.",br(),
537             "El modelo simplificado esta construido mediante el mé
            todo stepwise y puedes escoger el criterio de selecció
            n
            que este usará",
            style="text-align: left"),
538         circle = TRUE,
539         status = "success",
540         size = "xs",
541         icon = icon("circle-info"),
542         label = NULL,
543         tooltip = FALSE,
544         right = F,
545         up = FALSE,
546         width = NULL,
547         margin = "10px",
548         inline = TRUE,
549         inputId = NULL
550     )),
551     tabBox(title="Ajustes de modelos preliminares",width=12, id
        ="tab_box_summarys",

```

```

554     tabPanel("Ajuste del Modelo Lineal completo preliminar",
              verbatimTextOutput("summary_m1")),
555     tabPanel("Ajuste del Modelo Lineal simplificado
              preliminar", selectInput(inputId="select_input_AIC_1",
              label="Criterio de selección de modelo:", choices=c("
              AIC"=2,"BIC"=0), selected = "AIC"), verbatimTextOutput(
              "summary_m2"))
556   ),
557   h3(" · Validación de los modelos preliminares"),
558   tabBox(title="Gráficos de validación", width=12, id="tab_box_
              plots_modelos",
559         tabPanel("Residuos modelo lineal completo",
              plotOutput("plot_m1")),
560         tabPanel("ACF & PACF", plotOutput("plot_acf_m1")),
561         tabPanel("Residuos modelo lineal simplificado",
              plotOutput("plot_m2")),
562         tabPanel("ACF & PACF", plotOutput("plot_acf_m2"))
563   ),
564   h3(" · Selección de variables explicativas / AMC"),
565   column(width=12, dropdownButton(
566     p("A partir de los modelos previamente ajustados,
              selecciona
567     que retardos (de las variables AMC) quieres usar.", br(),
568     "Puedes seleccionar los del modelo completo o los
              resultantes del método stepwise.", br(),
569     "Ten en cuenta que estos retardos serán los introducidos
              en los modelos ARMAX y MARS.",
570     style="text-align: left"),
571     circle = TRUE,
572     status = "success",
573     size = "xs",
574     icon = icon("circle-info"),
575     label = NULL,
576     tooltip = FALSE,
577     right = F,
578     up = FALSE,
579     width = NULL,
580     margin = "10px",
581     inline = TRUE,

```

```

582     inputId = NULL
583  )),
584
585   box(width=12, selectInput(inputId="select_input_modelo_usar_
    ARMAX",label="Variables AMC del modelo:",choices=c("
    Completo"="m1","Simplificado"="m2"), selected = "Completo"
    )),
586
587   h3(" · Modelo ARMAX"),
588   column(width=12,dropdownButton(
589     p("A continuación ajustamos un modelo ARMAX.",
590       "Este se construye a partir de la función",em("auto.arima
        ()."),
591       "Por lo tanto, puedes indicar al procedimiento los
        valores máximos de cada parámetro.",br(),
592       "Después, se muestran el modelo ajustado y los t-ratios."
        ,br(),
593       "Por último, se ajustan modelos lineales a partir de las
        variables presentes en el modelo ARMAX.",
594       style="text-align: left"),
595     circle = TRUE,
596     status = "success",
597     size = "xs",
598     icon = icon("circle-info"),
599     label = NULL,
600     tooltip = FALSE,
601     right = F,
602     up = FALSE,
603     width = NULL,
604     margin = "10px",
605     inline = TRUE,
606     inputId = NULL
607   )),
608   box(width=12, column(width=4,numericInput(inputId="numeric_
    input_p",label="Máximo valor de p", value=7, min=0,max=7),
    numericInput(inputId="numeric_input_P",label="Máximo valor
    de P", value=0, min=0,max=0)),
609     column(width=4,numericInput(inputId="numeric_
        input_q",label="Máximo valor de q", value=0,

```

```

        min=0,max=7),numericInput(inputId="numeric_
        input_Q",label="Máximo valor de Q", value=0,
        min=0,max=7)),
610      column(width=4,numericInput(inputId="numeric_
        input_d",label="Máximo valor de d", value=0,
        min=0,max=1),numericInput(inputId="numeric_
        input_D",label="Máximo valor de D", value=0,
        min=0,max=1)),
611      verbatimTextOutput("autoarima_m3"),verbatimTextOutput("t_
        ratios")
612    ),
613
614    tabBox(title="Ajuste de los Modelos Lineales",width=12, id=
        "tab_box_summarys_auto",
615      tabPanel("Ajuste del Modelo Lineal completo",
        verbatimTextOutput("summary_m1_auto")),
616      tabPanel("Ajuste del Modelo Lineal simplificado",
        selectInput(inputId="select_input_AIC_2",label="
        Escoja AIC/BIC",choices=c("AIC"=2,"BIC"=0),
        selected = "AIC"),verbatimTextOutput("summary_m2_
        auto"))
617    ),
618    tabBox(title="Gráficos de validación",width=12,id="tab_box_
        plots_modelos_arima",
619      tabPanel("Residuos modelo lineal completo",
        plotOutput("plot_m1_arima")),
620      tabPanel("ACF & PACF", plotOutput("plot_acf_m1_arima
        ")),
621      tabPanel("Residuos modelo lineal simplificado",
        plotOutput("plot_m2_arima")),
622      tabPanel("ACF & PACF", plotOutput("plot_acf_m2_arima
        "))
623    ),
624    h3(" · Ajuste del Modelo MARS"),
625    column(width=12,dropdownButton(
626      p("A partir de las variables explicativas y los retardos
        resultantes
627      del procedimiento anterior, ajustamos el modelo", em("
        MARS.")),

```

```

628         style="text-align: left"),
629     circle = TRUE,
630     status = "success",
631     size = "xs",
632     icon = icon("circle-info"),
633     label = NULL,
634     tooltip = FALSE,
635     right = F,
636     up = FALSE,
637     width = NULL,
638     margin = "10px",
639     inline = TRUE,
640     inputId = NULL
641 ),
642     box(width=12,verbatimTextOutput("summary_mars")),
643 column(width=12,dropdownButton(
644     p("Se muestran varios gráficos de utilidad para validar e
645       interpretar el
646         modelo", em("MARS"),
647         style="text-align: left"),
648     circle = TRUE,
649     status = "success",
650     size = "xs",
651     icon = icon("circle-info"),
652     label = NULL,
653     tooltip = FALSE,
654     right = F,
655     up = FALSE,
656     width = NULL,
657     margin = "10px",
658     inline = TRUE,
659     inputId = NULL
660 ),
661     tabBox(title="Gráficos",width=12,id="tab_box_plots_modelos_
662       mars",
663         tabPanel("Respuesta sobre las explicativas",
664           plotOutput("plotmo_mars")),
665         tabPanel("Residuos",plotOutput("plot_res_mars")),
666         tabPanel("QQ - Plot", plotOutput("plot_qq_mars")),

```

```

664         tabPanel("ACF & PACF", plotOutput("plot_acf_mars")),
665         tabPanel("Recreación de la serie", plotOutput("plot_
        final_mars"))
666
667     ),
668     column(width=12, dropdownButton(
669         p("Pulsando el siguiente botón puedes descargar todos los
        outputs en formato PDF.",
670         style="text-align: left"),
671         circle = TRUE,
672         status = "success",
673         size = "xs",
674         icon = icon("circle-info"),
675         label = NULL,
676         tooltip = FALSE,
677         right = F,
678         up = FALSE,
679         width = NULL,
680         margin = "10px",
681         inline = TRUE,
682         inputId = NULL
683     )),
684     column(width=12, downloadButton("export", "Download Outputs")
        )
685
686     )
687
688 )
689 )
690 )
691 ,
692 dashboardControlbar(collapsed = TRUE, skinSelector())
693 )

```

Listing C.1: UI.r

```

1
2 ## server.R ##
3 library(shiny)
4 library(shinydashboard)

```

```
5 library(shinydashboardPlus)
6 library(shinyWidgets)
7 library(DT)
8
9 library(plotly)
10
11 library(openxlsx)
12
13 library(tidyquant)
14 library(tidyverse)
15 library(dplyr)
16
17 library(forecast)
18
19 # library(mgcv)
20 library(car)
21 # library(urca)
22 library(earth)
23
24 # library(rmarkdown)
25 # library(devtools)
26 library(psych)
27
28
29
30
31 function(input, output) {
32
33   ###
34   ###      TAB : CARGA DE FICHEROS
35   ###
36
37   ## CARGA FICHEROS DE ENTRADA EN TABLAS VISUALIZABLES / USABLES ##
38
39   ### TABLA MERGE BY MES DE LOS DATOS ###
40
41   uploaded_file <- reactive({
42
43     file <- input$file_input_excel
```

```

44
45   if(is.null(file)){
46     return(NULL)
47   }
48
49   path <- file$datapath
50
51   df <- path %>%
52     excel_sheets() %>%
53     set_names() %>%
54     map(~ read_excel(path, .x, col_names = TRUE))
55
56   all_sheets <- excel_sheets(file$datapath)
57
58   Ocupacion <- df[[which(all_sheets == "Ocupacion")]]
59   ConsumosDDDpor1000Estancias <- df[[which(all_sheets == "
60     ConsumosDDDpor1000Estancias")]]
61   ResultadosSensibles <- df[[which(all_sheets == "
62     ResultadosSensibles")]]
63   ResultadosResistentes <- df[[which(all_sheets == "
64     ResultadosResistentes")]]
65
66   colnames(ResultadosSensibles)[2:ncol(ResultadosSensibles)] <-
67     paste0(colnames(ResultadosSensibles)[2:ncol(
68       ResultadosSensibles)], "_sen")
69   colnames(ResultadosResistentes)[2:ncol(ResultadosResistentes)] <-
70     paste0(colnames(ResultadosResistentes)[2:ncol(
71       ResultadosResistentes)], "_res")
72
73   data <- Ocupacion %>%
74     left_join(ConsumosDDDpor1000Estancias, by = "mes") %>%
75     left_join(ResultadosSensibles, by = "mes") %>%
76     left_join(ResultadosResistentes, by = "mes")
77
78   return(data)
79 })
80
81 ### TABLA OCUPACION EN EL CENTRO ###

```

```

76
77 upload_ocupacion <- reactive({
78
79   file <- input$file_input_excel
80
81   if(is.null(file)){
82     return(NULL)
83   }
84
85   path <- file$datapath
86   df <- path %>%
87     excel_sheets() %>%
88     set_names() %>%
89     map(~ read_excel(path, .x, col_names = TRUE))
90
91   all_sheets <- excel_sheets(file$datapath)
92
93   Ocupacion <- df[[which(all_sheets == "Ocupacion")]]
94   Ocupacion <- Ocupacion %>%
95     mutate(mes = as_date(mes))
96
97   return(Ocupacion)
98 })
99
100
101 ### TABLA CONSUMOS ###
102
103 upload_consumos <- reactive({
104
105   file <- input$file_input_excel
106
107   if(is.null(file)){
108     return(NULL)
109   }
110
111   path <- file$datapath
112   df <- path %>%
113     excel_sheets() %>%
114     set_names() %>%

```

```

115     map(~ read_excel(path, .x, col_names = TRUE))
116
117     all_sheets <- excel_sheets(file$datapath)
118
119     ConsumosDDDpor1000Estancias <- df[[which(all_sheets == "
120         ConsumosDDDpor1000Estancias")]]
121     ConsumosDDDpor1000Estancias <- ConsumosDDDpor1000Estancias %>%
122         mutate(mes = as_date(mes))
123
124     return(ConsumosDDDpor1000Estancias)
125 })
126
127 ### TABLA OBSERVACIONES SENSIBLES ###
128
129 upload_obs_sensibles <- reactive({
130
131     file <- input$file_input_excel
132
133     if(is.null(file)){
134         return(NULL)
135     }
136
137     path <- file$datapath
138     df <- path %>%
139         excel_sheets() %>%
140         set_names() %>%
141         map(~ read_excel(path, .x, col_names = TRUE))
142
143     all_sheets <- excel_sheets(file$datapath)
144
145
146     ResultadosSensibles <- df[[which(all_sheets == "
147         ResultadosSensibles")]]
148     ResultadosSensibles <- ResultadosSensibles %>%
149         mutate(mes = as_date(mes))
150
151     return(ResultadosSensibles)
152 })

```

```

152
153
154 ### TABLA OBSERVACIONES RESISTENTES ###
155
156 upload_obs_resistentes <- reactive({
157
158   file <- input$file_input_excel
159
160   if(is.null(file)){
161     return(NULL)
162   }
163
164   path <- file$datapath
165   df <- path %>%
166     excel_sheets() %>%
167     set_names() %>%
168     map(~ read_excel(path, .x, col_names = TRUE))
169
170   all_sheets <- excel_sheets(file$datapath)
171
172
173   ResultadosResistentes <- df[[which(all_sheets == "
174     ResultadosResistentes")]]
175   ResultadosResistentes <- ResultadosResistentes %>%
176     mutate(mes = as_date(mes))
177
178   return(ResultadosResistentes)
179 })
180
181 ### DICCIONARIO CONSUMOS (MEDICAMENTO/ANTIBIOTICO) ###
182
183 ##### PARA MOSTRAR #####
184
185 upload_diccionario_meds <- reactive({
186
187   file <- input$file_input_excel
188
189   if(is.null(file)){

```

```
190     return(c("Sin diccionario"="NULL"))
191   } else {
192
193     path <- file$datapath
194
195     df <- path %>%
196       excel_sheets() %>%
197       set_names() %>%
198       map(~ read_excel(path, .x))
199
200     all_sheets <- excel_sheets(file$datapath)
201
202     DiccionarioConsumos <- df[[which(all_sheets == "
203       DiccionarioConsumos")]]
204     colnames(DiccionarioConsumos) <- c("Code", "Name")
205
206     return(DiccionarioConsumos)
207   }
208 })
209
210 ##### PARA LOS SELECCIONADORES #####
211
212 upload_diccionario_meds_selec <- reactive({
213
214   file <- input$file_input_excel
215
216   if(is.null(file)){
217     return(c("Sin diccionario"="NULL"))
218   } else {
219
220     path <- file$datapath
221
222     df <- path %>%
223       excel_sheets() %>%
224       set_names() %>%
225       map(~ read_excel(path, .x))
226
227     all_sheets <- excel_sheets(file$datapath)
```

```

228   DiccionarioConsumos <- df[[which(all_sheets == "
      DiccionarioConsumos")]]
229   colnames(DiccionarioConsumos) <- c("Code", "Name")
230
231   p <- setNames(DiccionarioConsumos$Code, DiccionarioConsumos$Name)
232
233   return(p)
234 }
235 })
236
237 ### DICCIONARIO RESISTENCIAS (BACTERIA/MICROORGANISMO) ###
238
239 ##### PARA MOSTRAR #####
240 upload_diccionario_micro <- reactive({
241
242   file <- input$file_input_excel
243
244   if(is.null(file)){
245     return(c("Sin diccionario"="NULL"))
246   } else {
247
248     path <- file$datapath
249
250     df <- path %>%
251       excel_sheets() %>%
252       set_names() %>%
253       map(~ read_excel(path, .x))
254
255     all_sheets <- excel_sheets(file$datapath)
256
257     DiccionarioResistencias <- df[[which(all_sheets == "
      DiccionarioResistencias")]]
258     DiccionarioResistencias <- DiccionarioResistencias[,c(1,2)]
259     colnames(DiccionarioResistencias) <- c("Code", "Name")
260     DiccionarioResistencias$Code <- substr(DiccionarioResistencias$
      Code, 1, 3)
261     DiccionarioResistencias <- DiccionarioResistencias[!duplicated(
      DiccionarioResistencias$Code), ]
262

```

```

263     return(DiccionarioResistencias)
264   }
265 })
266
267
268
269 ##### PARA LOS SELECCIONADORES #####
270 upload_diccionario_micro_selec <- reactive({
271
272   file <- input$file_input_excel
273
274   if(is.null(file)){
275     return(c("Sin diccionario"="NULL"))
276   } else {
277
278     path <- file$datapath
279
280     df <- path %>%
281       excel_sheets() %>%
282       set_names() %>%
283       map(~ read_excel(path, .x))
284
285     all_sheets <- excel_sheets(file$datapath)
286
287     DiccionarioResistencias <- df[[which(all_sheets == "
288       DiccionarioResistencias")]]
289     DiccionarioResistencias <- DiccionarioResistencias[,c(1,2)]
290     colnames(DiccionarioResistencias) <- c("Code", "Name")
291     DiccionarioResistencias$Code <- substr(DiccionarioResistencias$
292       Code, 1, 3)
293     DiccionarioResistencias <- DiccionarioResistencias[!duplicated(
294       DiccionarioResistencias$Code), ]
295
296     p <- setNames(DiccionarioResistencias$Code,
297       DiccionarioResistencias$Name)
298
299     return(p)
300   }
301 })

```

```
## BOTONES PARA GUARDAR O BORRAR LA TABLA EN EL SISTEMA ##
```

```
### OBJETO PARA GUARDAR LA TABLA ###
```

```
data <- reactiveValues(table_obs = NULL, table_cons=NULL, table_res=
  NULL, table_micro = c("Sin diccionario"="NULL"), table_meds = c("
  Sin diccionario"="NULL"))
```

```
### GUARDAR TABLA ###
```

```
observeEvent(input$action_button_excel_ready, {
  data$table_obs <- uploaded_file()
  data$table_micro <- upload_diccionario_micro_selec()
  data$table_meds <- upload_diccionario_meds_selec()
  data$table_res <- upload_obs_resistentes()
  data$table_cons <- upload_consumos()
  output$check_excel <- renderUI(icon('check'))
})
```

```
### BORRAR TABLA ###
```

```
observeEvent(input$action_button_excel_not_ready, {
  data$table_obs <- NULL
  data$table_micro <- c("Sin diccionario"="NULL")
  data$table_meds <- c("Sin diccionario"="NULL")
  data$table_res <- NULL
  data$table_cons <- NULL
  output$check_excel <- renderUI(NULL)
})
```

```
## VISUALIZADOR DE TABLAS ##
```

```
output$tabla_output_visual <- renderDT({
  if(input$select_input_table_visual == "tabla_ocupacion"){
```

```

335     datatable(upload_ocupacion(),
336               options = list(
337                 lengthMenu = list(c(5, 15, -1), c('5', '15', 'All'))
338                 ),
339                 pageLength = 5,
340                 paging = TRUE
341               ))
342 } else if(input$select_input_table_visual == "tabla_dicc_res"){
343     datatable(upload_diccionario_micro(),
344               options = list(
345                 lengthMenu = list(c(5, 15, -1), c('5', '15', 'All'))
346                 ),
347                 pageLength = 5,
348                 paging = TRUE
349               ))
350 } else if(input$select_input_table_visual == "tabla_consumos"){
351     datatable(upload_consumos(),
352               options = list(
353                 lengthMenu = list(c(5, 15, -1), c('5', '15', 'All'))
354                 ),
355                 pageLength = 5,
356                 paging = TRUE
357               ))
358 } else if(input$select_input_table_visual == "tabla_sensibles"){
359     datatable(upload_obs_sensibles(),
360               options = list(
361                 lengthMenu = list(c(5, 15, -1), c('5', '15', 'All'))
362                 ),
363                 pageLength = 5,
364                 paging = TRUE
365               ))
366 } else if(input$select_input_table_visual == "tabla_resistentes")
367 {
368     datatable(upload_obs_resistentes(),
369               options = list(
370                 lengthMenu = list(c(5, 15, -1), c('5', '15', 'All'))
371                 ),
372                 pageLength = 5,
373                 paging = TRUE
374               ))

```

```

368         ))
369     } else if(input$select_input_table_visual == "tabla_dicc_cons"){
370         datatable(upload_diccionario_meds(),
371             options = list(
372                 lengthMenu = list(c(5, 15, -1), c('5', '15', 'All'))
373                 ),
374                 pageLength = 5,
375                 paging = TRUE
376             ))
377     }else{
378         return(NULL)
379     }
380 })
381
382
383
384 ###
385 ###          TAB: DATA INICIAL
386 ###
387
388 ## SELECCIONADORES DE GRAFICOS ALIMENTADOS POR EL FICHERO SUBIDO ##
389
390 ### GRAFICOS DATA INICIAL ###
391
392 output$output_select_input_med_data_inicial <- renderUI({
393     selectInput("select_input_med_data_inicial","Seleccione
394         antibiótico:",data$table_meds)
395 })
396
397 output$output_select_input_micro_data_inicial <- renderUI({
398     selectInput("select_input_micro_data_inicial","Seleccione
399         bacteria:",data$table_micro )
400 })
401
402 output$summary_data_ini <- renderPrint({
403     df <- data$table_obs

```

```

404   if (!is.null(df)) {
405     res_col <- paste(input$select_input_micro_data_inicial, input$
      select_input_med_data_inicial, "_res", sep = "")
406     sen_col <- paste(input$select_input_micro_data_inicial, input$
      select_input_med_data_inicial, "_sen", sep = "")
407
408     if (all(c(res_col, sen_col) %in% colnames(df))) {
409
410       df <- df %>%
411         select(mes, !!sym(res_col), !!sym(sen_col), Ocupacion, !!
          sym(input$select_input_med_data_inicial)) %>%
412         mutate(total = !!sym(res_col) + !!sym(sen_col)) %>%
413         select(mes, !!sym(res_col), !!sym(sen_col), total,
          Ocupacion, !!sym(input$select_input_med_data_inicial))
          %>%
414         as.data.frame()
415
416
417
418         return(describe(df[, -1]))
419     } else {
420       res_col <- paste(input$select_input_micro_data_inicial, input
        $select_input_med_data_inicial, "_res", sep = "")
421       sen_col <- paste(input$select_input_micro_data_inicial, input
        $select_input_med_data_inicial, "_sen", sep = "")
422       df <- df %>%
423         select(mes, Ocupacion, !!sym(input$select_input_med_data_
          inicial)) %>%
424         as.data.frame()
425
426
427         return(describe(df[, -1]))
428     }
429   } else {
430     return(print("Sin datos"))
431   }
432 })
433
434

```

```
## GENERADORES DE GRAFICOS ##
```

```
### GRAFICO DE BARRAS APILADAS ###
```

```
output$plot_barras_apiladas <- renderPlotly({
  df <- data$table_obs
  if (!is.null(df)) {
    res_col <- paste(input$select_input_micro_data_inicial, input$
      select_input_med_data_inicial, "_res", sep = "")
    sen_col <- paste(input$select_input_micro_data_inicial, input$
      select_input_med_data_inicial, "_sen", sep = "")
    if (all(c(res_col, sen_col) %in% colnames(df))) {
      df <- df %>%
        select(mes, !!sym(res_col), !!sym(sen_col)) %>%
        as.data.frame()

      plot_ly(df, x = ~mes) %>%
        add_trace(y = ~get(res_col), name = "Resistencias", type =
          "bar") %>%
        add_trace(y = ~get(sen_col), name = "Sensibilidades", type =
          "bar") %>%
        layout(
          title = 'Observaciones Resistentes vs Sensibles',
          xaxis = list(title = "Mes"),
          yaxis = list(title = "Número de observaciones"),
          barmode = 'stack',
          plot_bgcolor = '#e5ecf6'
        )
    } else {
      plot_ly() %>%
        layout(
          title = "La combinación seleccionada no existe",
          xaxis = list(title = "Mes"),
          yaxis = list(title = "Número de observaciones")
        )
    }
  } else {
```

```

470   plot_ly() %>%
471     layout(
472       title = "Sin datos disponibles",
473       xaxis = list(title = "Mes"),
474       yaxis = list(title = "Número de observaciones")
475     )
476   }
477 })
478
479 ### GRAFICO DE LINEAS TOTALES ###
480
481 output$plot_line_totals <- renderPlotly({
482   df <- data$table_obs
483   if (!is.null(df)) {
484     res_col <- paste(input$select_input_micro_data_inicial, input$
485       select_input_med_data_inicial, "_res", sep = "")
486     sen_col <- paste(input$select_input_micro_data_inicial, input$
487       select_input_med_data_inicial, "_sen", sep = "")
488     if (all(c(res_col, sen_col) %in% colnames(df))) {
489       df <- df %>%
490         select(mes, !!sym(res_col), !!sym(sen_col)) %>%
491         mutate(total = !!sym(res_col) + !!sym(sen_col)) %>%
492         as.data.frame()
493
494       plot_ly(df, x = ~mes) %>%
495         add_trace(y = ~get(res_col), name = "Resistencias", type =
496           "scatter", mode = "lines", line = list(color = "blue"))
497         %>%
498         add_trace(y = ~get(sen_col), name = "Sensibilidades", type
499           = "scatter", mode = "lines", line = list(color = "red"))
500         %>%
501         add_trace(y = ~total, name = "Total", type = "scatter",
502           mode = "lines", line = list(color = "green")) %>%
503         layout(
504           title = 'Observaciones Resistencias, Sensibilidades y
505             Total',
506           xaxis = list(title = "Mes"),
507           yaxis = list(title = "Número de observaciones"),
508           plot_bgcolor = '#e5ecf6'

```

```

501     )
502   } else {
503     plot_ly() %>%
504       layout(
505         title = "La combinación seleccionada no existe",
506         xaxis = list(title = "Mes"),
507         yaxis = list(title = "Número de observaciones")
508       )
509   }
510 } else {
511   plot_ly() %>%
512     layout(
513       title = "Sin datos disponibles",
514       xaxis = list(title = "Mes"),
515       yaxis = list(title = "Número de observaciones")
516     )
517 }
518 })
519
520
521
522 ### GRAFICO LINE PLOT OCUPACIONES ###
523
524 output$plot_hospitalizaciones <- renderPlotly({
525   df <- data$table_obs
526   if (!is.null(df)) {
527     if ("Ocupacion" %in% colnames(df)) {
528       df <- df %>%
529         select(mes, Ocupacion) %>%
530         as.data.frame()
531
532       plot_ly() %>%
533         add_trace(data = df, x = ~mes, y = ~Ocupacion, type = '
534           scatter', mode = 'lines', fill = 'tozeroy', name = '
535             Ocupacion') %>%
536         layout(
537           title = 'Hospitalizaciones',
538           showlegend = F,
539           yaxis = list(title = "Hospitalizaciones", zerolinecolor =

```

```

      '#ffff', zerolinewidth = 2, gridcolor = '#ffff'),
538   xaxis = list(title = "Mes", zerolinecolor = '#ffff',
      zerolinewidth = 2, gridcolor = '#ffff'),
539   plot_bgcolor = '#e5ecf6',
540   width = 900
541   )
542   } else {
543     plot_ly() %>%
544     layout(
545       title = "La columna 'Ocupacion' no existe",
546       xaxis = list(title = "Mes"),
547       yaxis = list(title = "Hospitalizaciones")
548     )
549   }
550   } else {
551     plot_ly() %>%
552     layout(
553       title = "Sin datos disponibles",
554       xaxis = list(title = "Mes"),
555       yaxis = list(title = "Hospitalizaciones")
556     )
557   }
558   })
559
560
561   ### GRAFICO SERIE CONSUMOS ###
562
563   output$plot_serie_consumos <- renderPlotly({
564     df <- data$table_obs
565     if (!is.null(df)) {
566       if (input$select_input_med_data_inicial %in% colnames(df)) {
567         df <- df %>%
568           select(mes, !!sym(input$select_input_med_data_inicial)) %>%
569           as.data.frame()
570
571         plot_ly() %>%
572         add_trace(data = df, x = ~mes, y = ~get(input$select_input_
           med_data_inicial), type = 'scatter', mode = 'lines',
           fill = 'tozeroy', name = 'Antibiótico') %>%

```

```

573     layout(
574         title = 'Consumos',
575         showlegend = F,
576         yaxis = list(title = "Consumos", zerolinecolor = '#ffff',
577                     zerolinewidth = 2, gridcolor = '#ffff'),
578         xaxis = list(title = "Mes", zerolinecolor = '#ffff',
579                     zerolinewidth = 2, gridcolor = '#ffff'),
580         plot_bgcolor = '#e5ecf6',
581         width = 900
582     )
583 } else {
584     plot_ly() %>%
585     layout(
586         title = "La columna seleccionada no existe",
587         xaxis = list(title = "Mes"),
588         yaxis = list(title = "Consumos")
589     )
590 }
591 } else {
592     plot_ly() %>%
593     layout(
594         title = "Sin datos disponibles",
595         xaxis = list(title = "Mes"),
596         yaxis = list(title = "Consumos")
597     )
598 }
599 })
600
601
602
603
604 ###
605 ###          TAB: DATA ELABORADA
606 ###
607
608 ## SELECCIONADORES DE GRAFICOS ALIMENTADOS POR EL FICHERO SUBIDO ##
609

```

```

610 output$output_select_input_med_data_elaborada <- renderUI({
611   selectInput("select_input_med_data_elaborada", "Seleccione antibió
        tico:", data$table_meds )
612
613 })
614
615 output$output_select_input_micro_data_elaborada <- renderUI({
616   selectInput("select_input_micro_data_elaborada", "Seleccione
        bacteria:", data$table_micro )
617
618 })
619
620 ## GENERADORES DE GRAFICOS ##
621
622 ### GRAFICO % RESISTENCIAS ###
623
624 output$plot_porcentaje_resistencias <- renderPlotly({
625   df <- data$table_obs
626   if (!is.null(df)) {
627     res_col <- paste(input$select_input_micro_data_elaborada, input
        $select_input_med_data_elaborada, "_res", sep = "")
628     sen_col <- paste(input$select_input_micro_data_elaborada, input
        $select_input_med_data_elaborada, "_sen", sep = "")
629     if (any(colnames(df) %in% c(res_col, sen_col))) {
630       df <- df %>%
631         select(mes, !!sym(res_col), !!sym(sen_col))
632
633       df <- df %>%
634         mutate(
635           porc_res = (!!sym(res_col) / (!!sym(res_col) + !!sym(sen_
        col))) * 100
636         )
637       df <- as.data.frame(df)
638
639       plot_ly() %>%
640         add_trace(data = df[, c("mes", "porc_res")], type = '
        scatter', mode = 'lines', fill = 'tozeroy', x = ~df[, "
        mes"], y = ~df[, "porc_res"], name = 'Resistencias') %>%
641       layout(

```

```

642     title = 'Porcentaje de Resistencias',
643     showlegend = F,
644     yaxis = list(
645         title = "Porcentaje de observaciones resistentes",
646         zerolinecolor = '#ffff',
647         zerolinewidth = 2,
648         gridcolor = 'ffff'
649     ),
650     xaxis = list(
651         title = "Mes",
652         zerolinecolor = '#ffff',
653         zerolinewidth = 2,
654         gridcolor = 'ffff'
655     ),
656     plot_bgcolor = '#e5ecf6',
657     width = 900
658 )
659 } else {
660     plot_ly() %>%
661     layout(
662         title = "La combinación seleccionada no existe",
663         xaxis = list(title = "Mes"),
664         yaxis = list(title = "% Resistencias")
665     )
666 }
667 } else {
668     plot_ly() %>%
669     layout(
670         title = "Sin datos disponibles",
671         xaxis = list(title = "Mes"),
672         yaxis = list(title = "% Resistencias")
673     )
674 }
675 })
676
677
678 ### GRAFICO TDI ###
679
680 output$plot_TDI <- renderPlotly({

```

```

681 df <- data$table_obs
682 if (!is.null(df)) {
683   res_col <- paste(input$select_input_micro_data_elaborada, input
684     $select_input_med_data_elaborada, "_res", sep = "")
685   if (res_col %in% colnames(df)) {
686     df <- df %>%
687       select(mes, Ocupacion, !!sym(res_col)) %>%
688       mutate(TDI = (!!sym(res_col) * 1000) / Ocupacion) %>%
689       as.data.frame()
690
691   plot_ly() %>%
692     add_trace(data = df, x = ~mes, y = ~TDI, type = 'scatter',
693       mode = 'lines', fill = 'tozero', name = 'TDI') %>%
694     layout(
695       title = 'Tasa de Infección por Dosis (TDI)',
696       showlegend = F,
697       yaxis = list(title = "TDI", zerolinecolor = '#ffff',
698         zerolinewidth = 2, gridcolor = '#ffff'),
699       xaxis = list(title = "Mes", zerolinecolor = '#ffff',
700         zerolinewidth = 2, gridcolor = '#ffff'),
701       plot_bgcolor = '#e5ecf6',
702       width = 900
703     )
704   } else {
705     plot_ly() %>%
706     layout(
707       title = "La combinación seleccionada no existe",
708       xaxis = list(title = "Mes"),
709       yaxis = list(title = "TDI")
710     )
711   }
712 } else {
713   plot_ly() %>%
714   layout(
715     title = "Sin datos disponibles",
716     xaxis = list(title = "Mes"),
717     yaxis = list(title = "TDI")
718   )
719 }

```

```

716 })
717
718
719
720
721 ###
722 ###      TAB : MODELOS
723 ###
724
725 ### OBJETO REACTIVO ###
726
727 dt <- reactiveValues(AMC = NULL, AMR = NULL, data = NULL)
728
729 ### GUARDAR MODELOS ###
730
731 observeEvent(input$action_button_models_1_ready, {
732   dt$AMC <- AMC_pre()
733   dt$AMR <- AMR_pre()
734   dt$data <- dat_1()
735   output$check_models_1 <- renderUI(icon('check'))
736 })
737
738 observeEvent(input$action_button_models_2_ready, {
739   dt$AMC <- AMC_manual()
740   dt$AMR <- AMR_manual()
741   dt$data <- dat_2()
742   output$check_models_2 <- renderUI(icon('check'))
743 })
744
745
746
747
748
749 ### MODELOS A PARTIR DE FICHERO PREDETERMINADO ###
750
751
752
753 upload_mod <- reactive({
754   file <- input$file_input_models

```

```
755
756   if (is.null(file)) {
757     return(NULL)
758   }
759
760   path <- file$datapath # Cambiado de file$path a file$datapath
761   mod <- read.csv2(path, stringsAsFactors = FALSE) # Asegurarse de
       que los factores no se conviertan automáticamente
762   mod <- as.list(mod)
763   modelos <- names(mod)
764   modelos <- substr(modelos, 1, 6)
765   mod <- mod[!modelos %in% c("ecoMEM")]
766   return(mod)
767 })
768
769 upload_modelos <- reactive({
770   file <- input$file_input_models
771
772   if (is.null(file)) {
773     return(NULL)
774   }
775
776   path <- file$datapath # Cambiado de file$path a file$datapath
777   mod <- read.csv2(path, stringsAsFactors = FALSE) # Asegurarse de
       que los factores no se conviertan automáticamente
778   mod <- as.list(mod)
779   modelos <- names(mod)
780   modelos <- substr(modelos, 1, 6)
781   modelos <- modelos[!modelos %in% c("ecoMEM")]
782   return(modelos)
783 })
784
785
786 output$output_select_input_modelo_1 <- renderUI({
787   selectInput("select_input_modelo_1", "Seleccione modelo:",upload_
       modelos())
788 })
789
790
```

```

791 dat_1 <- reactive({
792   lista_consumos <- upload_mod()[[input$select_input_modelo_1]]
793   lista_consumos <- lista_consumos[lista_consumos != ""]
794
795   if (length(lista_consumos) == 0) {
796     return(NULL)
797   }
798
799   cons <- data$table_cons[, c("mes", lista_consumos)]
800   res <- data$table_res[, c("mes", input$select_input_modelo_1)]
801
802   df <- merge(res, cons, by = "mes")
803   df<- df[-c(1:5),-1]
804   return(df)
805 })
806
807
808
809
810 output$tabla_prueba_1 <- renderDT(datatable(dat_1(),
811                                           options = list(
812                                             lengthMenu = list(c(
813                                               (5, 15, -1), c('5',
814                                                 , '15', 'All')),
815                                             paging = TRUE
816                                             )))
817
818 AMC_pre <-reactive({
819   lista_consumos <- upload_mod()[[input$select_input_modelo_1]]
820   lista_consumos <- lista_consumos[lista_consumos != ""]
821
822   if (length(lista_consumos) == 0) {
823     return(NULL)
824   }
825   return(lista_consumos)
826 })
827 AMR_pre <- reactive({
828   return(input$select_input_modelo_1)

```

```

828 })
829
830 ###  MODELOS A PARTIR DE SELECCIÓN  ###
831
832
833 output$output_select_input_modelo_2 <- renderUI({
834   selectInput("select_input_modelo_2", "Seleccione variable
835             respuesta / AMR:", names(data$table_res)[-c(1)])
836 })
837
838
839 output$check_box_output_modelos <- renderUI({
840   prettyCheckboxGroup(
841     "check_box_input_modelos",
842     "Seleccione variables explicativas / AMC:",
843     choices = colnames(data$table_cons)[-c(1)],
844     selected = NULL,
845     shape = c("curve"),
846     outline = TRUE,
847     fill = TRUE,
848     inline = TRUE
849   )
850 })
851
852
853
854 dat_2 <- reactive({
855
856   if(length(input$check_box_input_modelos)==0){
857     res <- data$table_res[, c("mes", input$select_input_modelo_2)]
858     df <- res
859     df<- df[-c(1:5), -1]
860     return(df)
861   } else {
862     cons <- data$table_cons[, c("mes", input$check_box_input_modelos)
863                               ]
864     res <- data$table_res[, c("mes", input$select_input_modelo_2)]

```

```

865     df <- merge(res, cons, by = "mes")
866     df<- df[-c(1:5),-1]
867     return(df)}
868 })
869
870
871 output$tabla_prueba_2 <- renderDT(datatable(dat_2(),
872                                           options = list(
873                                             lengthMenu = list(c(
874                                               (5, 15, -1), c('5'
875                                                 , '15', 'All')),
876                                             pageLength = 5,
877                                             paging = TRUE
878                                           )))
879
880 AMR_manual <-reactive({
881   return(input$select_input_modelo_2)
882 })
883
884 AMC_manual <- reactive({
885   return(input$check_box_input_modelos)
886 })
887
888 # EXPLORACION
889
890
891
892
893
894 serie <- reactive({
895
896   cons <- data$table_cons[, c("mes", dt$AMC)]
897   res <- data$table_res[, c("mes", dt$AMR)]
898
899   df <- merge(res, cons, by = "mes")
900
901   df <- df[-c(1:5),]

```

```

902
903   start_date <- as.Date(df$mes[1])
904   end_date <- as.Date(df$mes[nrow(df)])
905
906   start_year <- as.numeric(format(start_date, "%Y"))
907   start_month <- as.numeric(format(start_date, "%m"))
908
909   end_year <- as.numeric(format(end_date, "%Y"))
910   end_month <- as.numeric(format(end_date, "%m"))
911
912   df <- df[, -which(names(df) == "mes")]
913
914   df_ts <- ts(df, start = c(start_year, start_month), end = c(end_
     year, end_month), freq = 12)
915
916   return(df_ts)
917 })
918
919 output$plot_ts <- renderPlot({
920   if(is.null(dt$AMC) || is.null(dt$AMR) ){
921     return(NULL)
922   } else {
923     df_ts <- serie()
924     a <- dt$AMR
925     plot(df_ts, main = a)
926   }
927 })
928
929
930
931 ### EXPLORACION ENTRE MODELOS
932
933
934
935 output$output_checkbox_retardos <- renderUI({
936
937
938
939

```

```

940     AMR <- dt$AMR
941     AMC <- dt$AMC
942
943     if(is.null(AMC)){
944         return("")
945     } else {
946         dat <- serie()
947         la=1:8
948         nam <- c()
949         for (v in AMR){
950             for (i in 1:length(la)) {
951                 nam<-c(nam,paste(v,i,sep="_"))
952             }
953         }
954
955         for (v in AMC){
956             for (i in 1:length(la)) {
957                 nam<-c(nam,paste(v,i,sep="_"))
958             }
959         }
960         return(
961         prettyCheckboxGroup(
962             "check_box_input_retardos",
963             "Retardos disponibles:",
964             choices = nam,
965             selected = NULL,
966             shape = c("curve"),
967             outline = FALSE,
968             fill = TRUE,
969             thick = FALSE,
970             bigger = FALSE,
971             inline = TRUE))
972     }
973
974
975 })
976
977
978

```

```
979
980 forms <- reactiveValues(data=NULL)
981
982
983 observeEvent(input$usar_retardos_1, {
984   forms$data <- data_retardos_1()
985
986 })
987
988 observeEvent(input$usar_retardos_2, {
989   forms$data <- data_retardos_2()
990
991 })
992
993
994
995
996 data_retardos_1 <- reactive({
997   dat <- serie()
998
999   n <- input$numeric_input_AMR
1000   m <- input$numeric_input_AMC
1001   AMR <- c(dt$AMR)
1002   AMC <- c(dt$AMC)
1003
1004   df <- dat[,1]
1005   nam <- c(AMR)
1006
1007   for(v in AMR){
1008     for (i in 1:n) {
1009       df <- cbind(df,stats::lag(dat[,v], i))
1010       nam <- c(nam,paste(v, i, sep="_"))
1011     }
1012   }
1013
1014
1015   for(v in AMC){
1016     for (i in 1:m) {
1017       df <- cbind(df,stats::lag(dat[,v], i))
```

```

1018     nam <- c(nam,paste(v, i, sep="_"))
1019   }
1020 }
1021
1022   colnames(df) <- nam
1023   return(df)
1024 })
1025
1026
1027
1028 data_retardos_2 <- reactive({
1029   dat <- serie()
1030
1031   vector <- c(input$check_box_input_retardos)
1032
1033   variables <- substr(vector, 1, nchar(vector) - 2)
1034   retardos<- substr(vector, nchar(vector), nchar(vector))
1035   retardos <- as.numeric(retardos)
1036
1037   AMR <- c(dt$AMR)
1038   AMC <- c(dt$AMC)
1039
1040   df <- dat[,1]
1041   nam <- c(AMR)
1042
1043   for(ind in 1:length(variables)){
1044     i <- retardos[ind]
1045     v <- variables[ind]
1046     df <- cbind(df,stats::lag(dat[,v], i))
1047     nam <- c(nam,paste(v, i, sep="_"))
1048   }
1049
1050
1051   colnames(df) <- nam
1052   return(df)
1053
1054 })
1055
1056 m1 <- reactive({

```

```

1057
1058   df <- forms$data
1059   AMR <- dt$AMR
1060   form0 <- as.formula(paste0(AMR, "~ ."))
1061   if(input$action_button_intercept_no_1 == TRUE || input$action_
1062       button_intercept_no_2==TRUE){
1063       form0 <- as.formula(paste0(AMR, "~ . - 1"))
1064   } else {
1065       form0 <- as.formula(paste0(AMR, "~ ."))
1066   }
1067
1068   if(is.null(df)){
1069       m1 <- NULL
1070   } else {
1071       m1 <- lm(form0, na.omit(df))
1072   }
1073   return(m1)
1074 })
1075
1076
1077 m2 <- reactive({
1078
1079   df <- forms$data
1080   AMR <- dt$AMR
1081
1082   form1=as.formula(paste0(AMR, "~1"))
1083   if(is.null(df)){
1084       m2 <- NULL
1085   }else {
1086       if(input$select_input_AIC_1 == 2){
1087           a <- 2
1088       } else {
1089           a <- log(nrow(df))
1090       }
1091       m2<-step(m1(), scope=list(lower=form1), trace=F, k=a)
1092   }
1093   return(m2)
1094

```

```
1095 })
1096
1097
1098
1099
1100
1101
1102 output$summary_m1 <- renderPrint({
1103   if(is.null(m1())){
1104     return("NULL")
1105   } else {
1106     return(summary(m1()))
1107   }
1108 })
1109
1110
1111 output$summary_m2 <- renderPrint({
1112   if(is.null(m2())){
1113     return("NULL")
1114   } else {
1115     return(summary(m2()))
1116   }
1117 })
1118
1119 output$plot_m1 <- renderPlot({
1120
1121   if(is.null(forms$data)){
1122     return(NULL)
1123   } else {
1124     par(mfrow = c(2, 2), mar = c(4, 4, 2, 1))
1125     plot(m1(), which=1, ask=F)
1126     plot(m1(), which=2, ask=F)
1127     plot(m1(), which=3, ask=F)
1128     plot(m1(), which=4, ask=F)
1129   }
1130 })
1131
1132
1133 output$plot_m2 <- renderPlot({
```

```

1134
1135   if(is.null(forms$data)){
1136     return(NULL)
1137   } else {
1138     par(mfrow = c(2, 2), mar = c(4, 4, 2, 1))
1139     plot(m2(), which=1, ask=F)
1140     plot(m2(), which=2, ask=F)
1141     plot(m2(), which=3, ask=F)
1142     plot(m2(), which=4, ask=F)
1143   }
1144 })
1145
1146 output$plot_acf_m1 <- renderPlot({
1147
1148   if(is.null(forms$data)){
1149     return(NULL)
1150   } else {
1151     par(mfrow=c(1,2))
1152     acf(resid(m1()), ylim=c(-1,1), lwd=2)
1153     pacf(resid(m1()), ylim=c(-1,1), lwd=2)
1154   }
1155 })
1156
1157 output$plot_acf_m2 <- renderPlot({
1158
1159   if(is.null(forms$data)){
1160     return(NULL)
1161   } else {
1162     par(mfrow=c(1,2))
1163     acf(resid(m2()), ylim=c(-1,1), lwd=2)
1164     pacf(resid(m2()), ylim=c(-1,1), lwd=2)
1165   }
1166 })
1167
1168
1169 ### ARMAX
1170
1171 modelo_a_usar <- reactive({
1172   m <- m1()

```

```

1173   if(input$select_input_modelo_usar_ARMAX == "m2"){
1174     m <- m2()
1175   } else {
1176     m <- m1()
1177   }
1178   return(m)
1179
1180 })
1181
1182
1183 m3 <- reactive({
1184   df <- forms$data
1185
1186   AMC <- c(dt$AMC)
1187
1188   b <- c(names(coef(modelo_a_usar()))))
1189   c <- c()
1190   for(i in 1:length(b)){
1191     if(substr(b[i],1,nchar(b[i])-2) %in% AMC){
1192       c <- c(c,b[i])
1193     }
1194   }
1195
1196   # b <- b[nchar(b)==5]
1197   a<- df[,c]
1198
1199
1200   if(is.null(df)){
1201     m3 <- NULL
1202   } else {
1203     if (length(names(coef(modelo_a_usar())))) > 2) {
1204       m3 <- auto.arima(df[,1], max.p=input$numeric_input_p, max.q=
         input$numeric_input_q, max.d=input$numeric_input_d, max.P=
         input$numeric_input_P, max.D=input$numeric_input_D, max.Q=
         input$numeric_input_Q, xreg=a)
1205     } else {
1206       m3 <- auto.arima(df[,1], max.p=input$numeric_input_p, max.q=
         input$numeric_input_q, max.P=input$numeric_input_P, max.Q=
         input$numeric_input_Q)

```

```
1207   }
1208 }
1209
1210   return(m3)
1211 })
1212
1213 output$autoarima_m3 <- renderPrint({
1214   if(is.null(forms$data)){
1215     return("NULL")
1216   } else {
1217     print(m3())
1218   }
1219 })
1220
1221
1222 output$t_ratios <- renderPrint({
1223   if(is.null(forms$data)){
1224     return("NULL")
1225   } else {
1226     d <- sqrt(diag(m3()$var.coef))
1227     c <- m3()$coef
1228     t_ratios <- c/d
1229     a <- data.frame(t_ratios)
1230     a <- t(a)
1231     colnames(a) <- names(coef(m3()))
1232     print(a)
1233   }
1234 })
1235
1236
1237 m1_auto <- reactive({
1238   df <- forms$data
1239
1240   if (is.null(df)) {
1241     m1 <- NULL
1242   } else {
1243     dat <- serie()
1244     AMR <- dt$AMR
1245     AMC <- dt$AMC
```

```

1246 df <- dat[, 1]
1247 nam <- c(AMR)
1248 p <- length(m3()$model$phi)
1249 b <- c(names(coef(modelo_a_usar()))))
1250
1251 if (p > 0) {
1252   for (i in 1:p) {
1253     df <- cbind(df, stats::lag(dat[, 1], i))
1254     nam <- c(nam, paste(AMR, i, sep = "_"))
1255   }
1256 }
1257
1258 amc_added <- FALSE
1259 for (v in AMC) {
1260   for (i in 1:8) {
1261     if (paste(v, i, sep = "_") %in% b) {
1262       df <- cbind(df, stats::lag(dat[, v], i))
1263       nam <- c(nam, paste(v, i, sep = "_"))
1264       amc_added <- TRUE
1265     }
1266   }
1267 }
1268
1269 if (p == 0 && amc_added == FALSE) {
1270   df <- dat[, 1, drop = FALSE]
1271   nam <- c(AMR)
1272 }
1273
1274 colnames(df) <- nam
1275 if (!any(nam %in% paste(AMC, 1:8, sep = "_")) && p == 0) {
1276   form0 <- as.formula(paste0(AMR, " ~ 1"))
1277 } else {
1278   form0 <- as.formula(paste0(AMR, "~ ."))
1279   if (input$action_button_intercept_no_1 == TRUE || input$action_
1280       button_intercept_no_2 == TRUE) {
1281     form0 <- as.formula(paste0(AMR, "~ . - 1"))
1282   }
1283 }

```

```
1284   m1 <- lm(form0, na.omit(df))
1285 }
1286 return(m1)
1287 })
1288
1289 m2_auto <- reactive({
1290   df <- forms$data
1291
1292   if (is.null(df)) {
1293     m2 <- NULL
1294   } else {
1295     dat <- serie()
1296     AMR <- dt$AMR
1297     AMC <- dt$AMC
1298     df <- dat[, 1]
1299     nam <- c(AMR)
1300     p <- length(m3()$model$phi)
1301     b <- c(names(coef(modelo_a_usar()))))
1302
1303     amc_added <- FALSE
1304
1305     if (p > 0) {
1306       for (i in 1:p) {
1307         df <- cbind(df, stats::lag(dat[, 1], i))
1308         nam <- c(nam, paste(AMR, i, sep = "_"))
1309       }
1310     }
1311
1312     for (v in AMC) {
1313       for (i in 1:8) {
1314         if (paste(v, i, sep = "_") %in% b) {
1315           df <- cbind(df, stats::lag(dat[, v], i))
1316           nam <- c(nam, paste(v, i, sep = "_"))
1317           amc_added <- TRUE
1318         }
1319       }
1320     }
1321
1322     if (p == 0 && amc_added == FALSE) {
```

```

1323     df <- dat[, 1, drop = FALSE]
1324     nam <- c(AMR)
1325   }
1326
1327   colnames(df) <- nam
1328
1329   if (!any(nam %in% paste(AMC, 1:8, sep = "_")) && p == 0) {
1330     form0 <- as.formula(paste0(AMR, " ~ 1"))
1331     form1 <- as.formula(paste0(AMR, " ~ 1"))
1332   } else if (p > 0) {
1333     form0 <- as.formula(paste0(AMR, " ~ ."))
1334     form1 <- as.formula(paste0(AMR, " ~ ", paste0(AMR, "_", 1:p,
1335       collapse = "+")))
1336   } else {
1337     form0 <- as.formula(paste0(AMR, " ~ ."))
1338     form1 <- as.formula(paste0(AMR, " ~ 1"))
1339   }
1340
1341   m1 <- lm(form0, data = na.omit(df))
1342
1343   if (input$select_input_AIC_2 == 2) {
1344     a <- 2
1345   } else {
1346     a <- log(nrow(df))
1347   }
1348
1349   m2 <- step(m1, scope = list(lower = form1), trace = FALSE, k = a)
1350 }
1351 return(m2)
1352 })
1353
1354 output$summary_m1_auto <- renderPrint({
1355   if(is.null(m1_auto())){
1356     return("NULL")
1357   } else {
1358     return(summary(m1_auto()))
1359   }
1360 })

```

```
1361
1362
1363 output$summary_m2_auto <- renderPrint({
1364   if(is.null(m2_auto())){
1365     return("NULL")
1366   } else {
1367     return(summary(m2_auto()))
1368   }
1369 })
1370
1371
1372 output$plot_m1_arima <- renderPlot({
1373
1374
1375   if(is.null(forms$data)){
1376     return(NULL)
1377   } else {
1378     par(mfrow = c(2, 2), mar = c(4, 4, 2, 1))
1379     plot(m1_auto(), which=1, ask=F)
1380     plot(m1_auto(), which=2, ask=F)
1381     plot(m1_auto(), which=3, ask=F)
1382     plot(m1_auto(), which=4, ask=F)
1383   }
1384 })
1385
1386
1387 output$plot_m2_arima <- renderPlot({
1388   if(is.null(forms$data)){
1389     return(NULL)
1390   } else {
1391     par(mfrow = c(2, 2), mar = c(4, 4, 2, 1))
1392     plot(m2_auto(), which=1, ask=F)
1393     plot(m2_auto(), which=2, ask=F)
1394     plot(m2_auto(), which=3, ask=F)
1395     plot(m2_auto(), which=4, ask=F)}
1396 })
1397
1398
1399 output$plot_acf_m1_arima <- renderPlot({
```

```

1400   if(is.null(forms$data)){
1401     return(NULL)
1402   } else {
1403     par(mfrow=c(1,2))
1404     acf(resid(m1_auto()),ylim=c(-1,1),lwd=2)
1405     pacf(resid(m1_auto()),ylim=c(-1,1),lwd=2)}
1406
1407   })
1408
1409   output$plot_acf_m2_arima <- renderPlot({
1410     if(is.null(forms$data)){
1411       return(NULL)
1412     } else {
1413       par(mfrow=c(1,2))
1414       acf(resid(m2_auto()),ylim=c(-1,1),lwd=2)
1415       pacf(resid(m2_auto()),ylim=c(-1,1),lwd=2)}
1416
1417     })
1418
1419   # MARS
1420
1421   m5 <- reactive({
1422     if (length(coef(m2_auto())) > 2) {
1423       df <- forms$data
1424
1425       if (is.null(df)) {
1426         m5 <- NULL
1427       } else {
1428         p <- length(m3()$model$phi)
1429         pp <- p + 1
1430         vars <- c(names(coef(m2_auto()))[-c(1:c(pp))])
1431         np = 6
1432         ng = 10
1433         ms = 50
1434
1435         dat <- serie()
1436         AMR <- dt$AMR
1437         AMC <- c(dt$AMC)
1438         df <- dat[, 1]

```

```

1439   nam <- c(AMR)
1440   b <- c(names(coef(modelo_a_usar()))))
1441
1442   amc_added <- FALSE
1443   if (p > 0) {
1444     for (i in 1:p) {
1445       df <- cbind(df, stats::lag(dat[, 1], i))
1446       nam <- c(nam, paste(AMR, i, sep = "_"))
1447     }
1448     for (v in AMC) {
1449       for (i in 1:8) {
1450         if (paste(v, i, sep = "_") %in% b) {
1451           df <- cbind(df, stats::lag(dat[, v], i))
1452           nam <- c(nam, paste(v, i, sep = "_"))
1453           amc_added <- TRUE
1454         }
1455       }
1456     }
1457   } else {
1458     for (v in AMC) {
1459       for (i in 1:8) {
1460         if (paste(v, i, sep = "_") %in% b) {
1461           df <- cbind(df, stats::lag(dat[, v], i))
1462           nam <- c(nam, paste(v, i, sep = "_"))
1463           amc_added <- TRUE
1464         }
1465       }
1466     }
1467   }
1468
1469   if (p == 0 && amc_added == FALSE) {
1470     df <- dat[, 1, drop = FALSE]
1471     nam <- c(AMR)
1472   }
1473
1474   colnames(df) <- nam
1475
1476   if (p > 0 && amc_added) {
1477     form3 <- as.formula(paste0(AMR, "~", paste0(AMR, "_", 1:p,

```

```

collapse = "+"), "+", paste0(vars, collapse = "+"))))
1478 } else if (p == 0 && amc_added) {
1479   form3 <- as.formula(paste0(AMR, "~", paste0(vars, collapse =
collapse = "+"))))
1480 } else if (p > 0) {
1481   form3 <- as.formula(paste0(AMR, "~", paste0(AMR, "_", 1:p,
collapse = "+"))))
1482 } else {
1483   form3 <- as.formula(paste0(AMR, "~ 1"))
1484 }
1485
1486 m5 <- earth(form3, data = na.omit(df), nfold = 1, nk = ng,
minsplan = ms,
1487             nprune = np, linpreds = paste0(AMR, "_", 1:p),
trace = 4, pmethod = "exh")
1488 }
1489 return(m5)
1490 }
1491 })
1492
1493
1494 output$summary_mars <- renderPrint({
1495   if(is.null(forms$data)){
1496     return("NULL")
1497   } else {
1498     a <- m5()
1499     print(a)}
1500 })
1501
1502 output$plotmo_mars <- renderPlot({
1503   if(is.null(forms$data)){
1504     return("NULL")
1505   } else {
1506     a <- m5()
1507     plotmo(a, ask=FALSE)}
1508 })
1509
1510 output$plot_res_mars <- renderPlot({
1511   if(is.null(forms$data)){

```

```

1512     return("NULL")
1513   } else {
1514     cons <- data$table_cons[, c("mes", dt$AMC)]
1515     res <- data$table_res[, c("mes", dt$AMR)]
1516
1517     df <- merge(res, cons, by = "mes")
1518
1519     df <- df[-c(1:5),]
1520
1521     start_date <- as.Date(df$mes[1])
1522
1523     start_year <- as.numeric(format(start_date, "%Y"))
1524     start_month <- as.numeric(format(start_date, "%m"))
1525
1526     plot(ts(resid(m5()), start=c(start_year, start_month), freq=12), main="
1527           Residuos")
1528   })
1529
1530
1531 output$plot_qq_mars <- renderPlot({
1532   if(is.null(forms$data)){
1533     return("NULL")
1534   } else {
1535     qqnorm(resid(m5()))
1536     qqline(resid(m5()), col=2, lwd=2)}
1537 })
1538
1539
1540 output$plot_acf_mars <- renderPlot({
1541   if(is.null(forms$data)){
1542     return("NULL")
1543   } else {
1544     par(mfrow=c(1,2))
1545     acf(resid(m5()), ylim=c(-1,1), lwd=2, main="Residuos")
1546     pacf(resid(m5()), ylim=c(-1,1), lwd=2, main="Residuos")}
1547 })
1548
1549

```

```

1550
1551 output$plot_final_mars <- renderPlot({
1552   if(is.null(forms$data)){
1553     return("NULL")
1554   } else {
1555     AMR <- dt$AMR
1556     cons <- data$table_cons[, c("mes", dt$AMC)]
1557     res <- data$table_res[, c("mes", dt$AMR)]
1558
1559     df <- merge(res, cons, by = "mes")
1560
1561     df <- df[-c(1:5),]
1562
1563     start_date <- as.Date(df$mes[1])
1564
1565     start_year <- as.numeric(format(start_date, "%Y"))
1566     start_month <- as.numeric(format(start_date, "%m"))
1567
1568     par(mfrow=c(1,1))
1569     plot(ts(df[,AMR], start=start_year, freq=12))
1570     lines(ts(predict(m5()), start=c(start_year, start_month), freq=12),
1571           col=2)
1572     legend("topright", legend=c(AMR, "NL-TSA Model"), col=c(1,2), lty=1)
1573   }
1574 })
1575
1576 output$export <- downloadHandler(
1577   filename = function() {"Plot_output.pdf"},
1578   content = function(file) {
1579     pdf(file, width=12)
1580
1581     par(mfrow=c(1,1))
1582     df_ts <- serie()
1583     a <- dt$AMR
1584     plot(df_ts, main = a)
1585
1586     par(mfrow=c(1,1))

```

```

1587 model_summary <- summary(m1())
1588 plot.new()
1589 title("Ajuste del Modelo Lineal completo preliminar:")
1590 text(0.5, 0.5, paste(capture.output(model_summary), collapse="\n"
    ), cex=0.9) # Adjust cex for font size
1591
1592 par(mfrow=c(1,1))
1593 model_summary <- summary(m2())
1594 plot.new()
1595 title("Ajuste del Modelo Lineal simplificado preliminar:")
1596 text(0.5, 0.5, paste(capture.output(model_summary), collapse="\n"
    ), cex=0.9) # Adjust cex for font size
1597
1598
1599 par(mfrow = c(2, 2), mar = c(4, 4, 4, 4), oma = c(0, 0, 5, 0))
1600 plot(m1(), which = 1, ask = FALSE)
1601 plot(m1(), which = 2, ask = FALSE)
1602 plot(m1(), which = 3, ask = FALSE)
1603 plot(m1(), which = 4, ask = FALSE)
1604 mtext("Validacion del Modelo Lineal completo preliminar:", outer
    = TRUE, cex = 1.5, line = 1)
1605
1606 par(mfrow = c(1, 2), mar = c(4, 4, 4, 4), oma = c(0, 0, 5, 0))
1607 acf(resid(m1()),ylim=c(-1,1),lwd=2)
1608 pacf(resid(m1()),ylim=c(-1,1),lwd=2)
1609 mtext("Estructura de autocorrelacion de los residuos del Modelo
    Lineal completo preliminar:", outer = TRUE, cex = 1.5, line =
    1)
1610
1611
1612 par(mfrow = c(2, 2), mar = c(4, 4, 4, 4), oma = c(0, 0, 5, 0)
    )
1613 plot(m2(),which=1,ask=F)
1614 plot(m2(),which=2,ask=F)
1615 plot(m2(),which=3,ask=F)
1616 plot(m2(),which=4,ask=F)
1617 mtext("Validacion Modelo Lineal simplificado preliminar:",
    outer = TRUE, cex = 1.5, line = 1)
1618

```

```

1619     par(mfrow = c(1, 2), mar = c(4, 4, 4, 4), oma = c(0, 0, 5, 0)
1620         )
1621     acf(resid(m2()),ylim=c(-1,1),lwd=2)
1622     pacf(resid(m2()),ylim=c(-1,1),lwd=2)
1623     mtext("Estructura de autocorrelacion de los residuos del
1624           Modelo Lineal simplificado preliminar:", outer = TRUE, cex
1625           = 1.5, line = 1)
1626
1627     par(mfrow=c(1,1))
1628     a <- print(m3())
1629     model_output <- capture.output(print(a))
1630     plot.new()
1631     title("Ajuste del Modelo Autoarima:")
1632     text(0.5, 0.5, paste(model_output, collapse="\n"), cex=0.9)
1633
1634     d <- sqrt(diag(m3()$var.coef))
1635     c <- m3()$coef
1636     t_ratios <- c/d
1637     a <- data.frame(t_ratios)
1638     a <- t(a)
1639     colnames(a) <- names(coef(m3()))
1640     model_output <- capture.output(print(a))
1641     plot.new()
1642     title("T-ratios del Modelo Autoarima:")
1643     text(0.5, 0.5, paste(model_output, collapse="\n"), cex=0.9)
1644
1645
1646     par(mfrow=c(1,1))
1647     model_summary <- summary(m1_auto())
1648     plot.new()
1649     title("Ajuste del Modelo Lineal total con estructura AR(p):")
1650     text(0.5, 0.5, paste(capture.output(model_summary), collapse="\n"
1651                           ), cex=0.9) # Adjust cex for font size
1652
1653     par(mfrow=c(1,1))
1654     model_summary <- summary(m2_auto())

```

```

1654 plot.new()
1655 title("Ajuste del Modelo Lineal simplificado con estructura AR(p)
      :")
1656 text(0.5, 0.5, paste(capture.output(model_summary), collapse="\n"
      ), cex=0.9) # Adjust cex for font size
1657
1658
1659 par(mfrow = c(2, 2), mar = c(4, 4, 4, 4), oma = c(0, 0, 5, 0))
1660 plot(m1_auto(), which=1, ask=F)
1661 plot(m1_auto(), which=2, ask=F)
1662 plot(m1_auto(), which=3, ask=F)
1663 plot(m1_auto(), which=4, ask=F)
1664 mtext("Validacion del Modelo Lineal completo:", outer = TRUE,
      cex = 1.5, line = 1)
1665
1666 par(mfrow = c(1, 2), mar = c(4, 4, 4, 4), oma = c(0, 0, 5, 0))
1667 acf(resid(m1_auto()), ylim=c(-1,1), lwd=2)
1668 pacf(resid(m1_auto()), ylim=c(-1,1), lwd=2)
1669 mtext("Estructura de autocorrelacion de los residuos del Modelo
      Lineal completo:", outer = TRUE, cex = 1.5, line = 1)
1670
1671
1672 par(mfrow = c(2, 2), mar = c(4, 4, 4, 4), oma = c(0, 0, 5, 0))
1673 plot(m2_auto(), which=1, ask=F)
1674 plot(m2_auto(), which=2, ask=F)
1675 plot(m2_auto(), which=3, ask=F)
1676 plot(m2_auto(), which=4, ask=F)
1677 mtext("Validacion del Modelo Lineal simplificado:", outer =
      TRUE, cex = 1.5, line = 1)
1678
1679 par(mfrow = c(1, 2), mar = c(4, 4, 4, 4), oma = c(0, 0, 5, 0))
1680 acf(resid(m2_auto()), ylim=c(-1,1), lwd=2)
1681 pacf(resid(m2_auto()), ylim=c(-1,1), lwd=2)
1682 mtext("Estructura de autocorrelacion de los residuos del Modelo
      Lineal simplificado:", outer = TRUE, cex = 1.5, line = 1)
1683
1684
1685 par(mfrow=c(1,1))
1686 a <- print(m5())

```

```

1687 model_output <- capture.output(print(a))
1688 plot.new()
1689 title("Ajuste del Modelo MARS:")
1690 text(0.5, 0.5, paste(model_output, collapse="\n"), cex=0.9)
1691
1692 par(mfrow=c(1,1), mar = c(4, 4, 4, 4), oma = c(0, 0, 5, 0))
1693 a <- m5()
1694 plotmo(a, ask=FALSE)
1695 mtext("Plotmo:", outer = TRUE, cex = 1.5, line = 0)
1696
1697
1698 cons <- data$table_cons[, c("mes", dt$AMC)]
1699 res <- data$table_res[, c("mes", dt$AMR)]
1700
1701 df <- merge(res, cons, by = "mes")
1702
1703 df <- df[-c(1:5),]
1704
1705 start_date <- as.Date(df$mes[1])
1706
1707 start_year <- as.numeric(format(start_date, "%Y"))
1708 start_month <- as.numeric(format(start_date, "%m"))
1709 par(mfrow=c(1,1))
1710 plot(ts(resid(m5()), start=c(start_year, start_month), freq=12), main
1711      ="Residuos")
1712 mtext("Residuos del Modelo MARS:", outer = TRUE, cex = 1.5, line
1713      = 0)
1714
1715
1716 par(mfrow=c(1,1))
1717 qqnorm(resid(m5()))
1718 qqline(resid(m5()), col=2, lwd=2)
1719 mtext("Residuos en papel probabilistico normal:", outer = TRUE,
1720      cex = 1.5, line = 0)
1721
1722
1723 par(mfrow = c(1, 2), mar = c(4, 4, 4, 4), oma = c(0, 0, 5, 0))
1724 acf(resid(m5()), ylim=c(-1,1), lwd=2, main="Residuos")
1725 pacf(resid(m5()), ylim=c(-1,1), lwd=2, main="Residuos")

```

```

1723   mtext("Estructura de autocorrelación de los residuos del Modelo
1724         MARS:", outer = TRUE, cex = 1.5, line = 1)
1725
1726   AMR <- dt$AMR
1727   cons <- data$table_cons[, c("mes", dt$AMC)]
1728   res <- data$table_res[, c("mes", dt$AMR)]
1729
1730   df <- merge(res, cons, by = "mes")
1731
1732   df <- df[-c(1:5),]
1733
1734   start_date <- as.Date(df$mes[1])
1735
1736   start_year <- as.numeric(format(start_date, "%Y"))
1737   start_month <- as.numeric(format(start_date, "%m"))
1738
1739   par(mfrow=c(1,1))
1740   plot(ts(df[,AMR], start=start_year, freq=12))
1741   lines(ts(predict(m5()), start=c(start_year, start_month), freq
1742         =12), col=2)
1742   legend("topright", legend=c(AMR,"NL-TSA Model"), col=c(1,2), lty
1743         =1)
1743   mtext("Reconstrucción de la serie mediante el modelo:", outer =
1744         TRUE, cex = 1.5, line = 0)
1744
1745   dev.off()
1746 }
1747 )
1748 }

```

Listing C.2: Server.r