



Universidad de Santiago de Chile.
Facultad de Ciencias
Departamento de Matemática y Ciencias de la Computación
Ingeniería Estadística

Implementación de Series de Tiempo Continuas para la Modelación de Datos Irregularmente Espaciados

Rodrigo Barraza Cartagena
Profesor Guía: Felipe Elorrieta López

Trabajo de titulación presentado en la Facultad de Ciencias
en cumplimiento parcial de los requisitos exigidos para
optar al título de Ingeniero Estadístico.

Santiago, 29 de Abril de 2019

Agradecimientos

Esté trabajo, está dedicado en especial, a todas las personas que fueron un pilar fundamental en mi vida y en mi proceso como estudiante, ya que sin ustedes, nada hubiese sido lo mismo y sería la persona que soy hoy.

En primer lugar, quiero agradecer a mis padres, Ricardo Barraza y Bernarda Cartagena, quienes son el pilar fundamental en mi vida, sin ellos, no estaría en estos momentos redactando mi tesis. Siempre han estado a mi lado y me han apoyado en las decisiones que he tomado a lo largo de mi vida, tanto estudiantil como personal. Gracias a ellos soy la persona que soy, con mis virtudes y defectos. Todo este trabajo está dedicado a ellos.

Quiero agradecer a mis hermanos Ricardo Barraza y Maria Fernanda Barraza que también han sido parte importante dentro de mi formación académica.

A mis mejores amigos, mis segundos hermanos, a los que siempre han estado conmigo en las buenas y en las malas. Diego, Mauricio, Brito, , Valentino, Jorge, Yerko. Gracias por ser parte de mi vida, tanto universitaria como amigos de la vida, han sido una parte importante dentro de todo este proceso llamado universidad. Gracias por las risas, bailes, borracheras, problemas, viajes, y bellos momentos que hemos disfrutado a lo largo de estos años. Se les llevará muy adentro en el corazón por siempre.

También a mi profesor guía por la paciencia, apoyo y tiempo invertido en este pequeño ser humano; sin usted, este proyecto no podría haberse llevado a cabo.

Gracias a todos los profesores que formaron parte de mi vida estudiantil. A los que aun siguen enseñando día a día para hacer de nosotros personas competentes e íntegras. Y también a los que no siguen con nosotros, ya que siempre serán una parte fundamental para los que tuvimos el agrado de llegarlos a conocer.

Don't worry, be happy

Resumen

Este trabajo ha sido realizado con el fin de otorgar el conocimiento necesario para poder implementar los modelos o procesos continuos: autoregresivo (CAR), autoregresivos de medias móviles (CARMA) y autoregresivos fraccionalmente integrado de medias móviles (CARFIMA), además de hacer una comparación con los modelos a tiempo discreto AR, ARMA y ARFIMA respectivamente; y así poder decidir y concluir qué modelos se ajustan mejor a qué tipo de datos.

En este trabajo se describirá toda la teoría necesaria que hay detrás de cada uno de ellos junto con sus propiedades y métodos de estimación, luego se trabajará con los paquetes `cts`, `yuima` y `carfima` del software R, los cuales son necesarios para ajustar los procesos CAR, CARMA y CARFIMA respectivamente. Cada paquete cuenta con las funciones necesarias para ajustar datos (simulados o reales) los que serán mencionados y descritos extensamente. Además, se describirá una reparametrización que es usada para la estimación del modelo CAR.

Por ultimo, la parte práctica que será realizada en el software R, se constatará que el modelo CARMA produce estimaciones más precisas que el modelo ARMA, en cuanto a MSE y MAPE se refiere; al igual que el modelo CARFIMA que produce estimaciones más precisas que el modelo ARFIMA. Además, se confirmará que tanto el modelo CAR(1) como AR(1), igualmente espaciado, producen resultados idénticos debido a la discretización del modelo CAR(1). Esto cambiará cuando se ocupen tiempos irregularmente espaciados (datos reales) ya que en este punto se observa que el proceso continuo CAR produce resultados más ajustados que el modelo AR, hablando en términos de MSE.

Palabras clave: Series de tiempo, Proceso continuos CAR, Procesos continuos CARMA, Procesos continuos CARFIMA, `cts`, `yuima`, `carfima`, Tiempos irregularmente espaciados, AR, ARMA, ARFIMA.

Índice general

Agradecimientos	3
Resumen	4
Indice de Gráficos	10
Indice de Tablas	12
1. Introducción	13
1.1. Planteamiento del Problema	15
1.2. Objetivos	16
1.2.1. Objetivo General	16
1.2.2. Objetivos Específicos	16
1.3. Metodología	17
2. Preliminares	18
2.1. Series de tiempo	18
2.2. Procesos estocásticos	18
2.3. Procesos estacionarios	19
2.4. Autocovarianza	19
2.5. Autocorrelación	19
2.6. Ruido Blanco discreto	20
2.7. Modelo AR	20
2.8. Modelo ARMA	21
2.9. Modelo ARFIMA	22
2.10. Modelos estado-espacio	23
2.11. Filtro de Kalman	24
3. Marco Teórico	26
3.1. Ruido Blanco continuo	26
3.2. Movimiento Browniano	26
3.3. Modelos CAR(p)	28
3.3.1. CAR(1)	28
3.3.2. Reparametrización CAR	29
3.3.3. Estimación vía filtro de Kalman	31
3.3.4. Selección del modelo	33
3.4. Modelos CARMA(p,q)	33

3.4.1. CARMA(2,1)	35
3.4.2. Estimación del modelo	35
3.5. Modelos CARFIMA(p,H,q)	38
3.5.1. Estimación del modelo	40
4. Resultados	41
4.1. Simulación	43
4.1.1. AR y CAR	43
4.1.1.1. AR(1) y CAR(1)	44
4.1.1.2. AR(2) y CAR(2)	53
4.1.2. ARMA y CARMA	56
4.1.2.1. ARMA(2,1) y CARMA(2,1)	59
4.1.2.2. ARMA(3,1) y CARMA(3,1)	60
4.1.3. ARFIMA y CARFIMA	62
4.2. Datos Reales	68
4.2.1. Aplicación Geofísica	68
4.2.2. Aplicación Médica	71
5. Conclusiones	75
Referencias	77
Apéndice	79
Función MAPE	79
Comparación autocorrelacion AR(1) y CAR(1)	79
Gráfico modelo AR(1): datos vs predichos con datos igualmente espaciado	79
Gráfico modelo CAR(1): datos vs predichos con datos igualmente espaciado	80
MSE y MAPE para 1000 simulaciones para los modelos AR(1) y CAR(1) con datos igualmente espaciado	80
Gráfico modelo AR(1): datos vs predichos con datos irregularmente espaciado	81
Gráfico modelo CAR(1): datos vs predichos con datos irregularmentee espaciado	82
MSE y MAPE para 1000 simulaciones para los modelos AR(1) y CAR(1) con datos irregularmente espaciado	82
Gráfico modelo AR(2): datos vs predichos	83
Gráfico modelo CAR(2): datos vs predichos	84
MSE y MAPE para 1000 simulaciones para los modelos AR(2) y CAR(2)	84
Gráfico modelo CARMA(2,1): datos vs predichos	85
Gráfico modelo ARMA(2,1): datos vs predichos	85
Gráfico modelo CARMA(3,1): datos vs predichos	86
Gráfico modelo ARMA(3,1): datos vs predichos	86
Gráfico modelo CARFIMA(1,H,0): datos vs predichos con datos irregularmente espaciado	87
Gráfico modelo ARFIMA(1,d,0): datos vs predichos con datos irregularmente espaciado	87
MSE y MAPE para 100 simulaciones para los modelos CARFIMA(1,H,0) y ARFIMA(1,d,0)	88
Aplicación Geofísica: Gráficos y tabla de los modelos CAR(1), CAR(3) y AR(1)	89

Aplicación Médica: Gráficos y tabla de los modelos CAR(1), CAR(2) y AR(1) 90

Índice de Gráficos

4.1. Gráfico de datos simulados versus datos predichos para un modelo AR(1) con datos regularmente espaciados (ver apéndice (5))	46
4.2. Gráfico de datos simulados versus datos predichos para un modelo CAR(1) con datos regularmente espaciados (ver apéndice (5))	47
4.3. Gráfico de datos simulados versus datos predichos para un modelo CAR(1) irregularmente espaciado (ver apéndice (5))	50
4.4. Gráfico de datos simulados versus datos predichos para un modelo AR(1) (ver apéndice (5))	50
4.5. Gráfico de datos simulados versus datos predichos para un modelo AR(2) (ver apéndice (5))	53
4.6. Gráfico de datos simulados versus datos predichos para un modelo CAR(2) (ver apéndice (5))	53
4.7. Gráfico de datos simulados versus datos predichos para un modelo CAR-MA(2,1) (ver apéndice (5))	59
4.8. Gráfico de datos simulados versus datos predichos para un modelo AR-MA(2,1) (ver apéndice (5))	60
4.9. Gráfico de datos simulados versus datos predichos para un modelo CAR-MA(3,1) (ver apéndice (5))	61
4.10. Gráfico de datos simulados versus datos predichos para un modelo AR-MA(3,1) (ver apéndice (5))	61
4.11. Gráfico de datos simulados versus datos predichos para un modelo CAR-FIMA(1,H,0), con $H = 0,7$ (ver apéndice (5))	65
4.12. Gráfico de datos simulados versus datos predichos para un modelo AR-FIMA(1,d,0), con $d = 0,1$ (ver apéndice (5))	66
4.13. Gráfico de las mediciones de abundancia relativa de un isótopo de oxígeno $\delta^{18}\text{O}$	68
4.14. Gráfico de las mediciones de abundancia relativa de un isótopo de oxígeno versus los predichos para un modelo CAR(1) (ver apéndice (5))	69
4.15. Gráfico de las mediciones de abundancia relativa de un isótopo de oxígeno versus los predichos para un modelo CAR(3) (ver apéndice (5))	69
4.16. Gráfico de las mediciones de abundancia relativa de un isótopo de oxígeno versus los predichos para un modelo AR(1) (ver apéndice (5))	70
4.17. Gráfico de las mediciones de la función pulmonar de un paciente asmático	71
4.18. Gráfico de mediciones de la función pulmonar de un paciente asmático versus los predichos para un modelo CAR(1) (ver apéndice (5))	72

4.19. Gráfico de mediciones de la función pulmonar de un paciente asmático versus los predichos para un modelo CAR(2) (ver apéndice (5))	72
4.20. Gráfico de mediciones de la función pulmonar de un paciente asmático versus los predichos para un modelo AR(1) (ver apéndice (5))	73

Índice de Tablas

4.1. Autocorrelacion con $lag = 1$ para 1000 simulaciones ajustando los modelos AR(1) y CAR(1) con datos igualmente espaciados (ver apéndice (5))	44
4.2. MSE y MAPE de las estimaciones y predicciones de los datos con los modelos AR(1) y CAR(1) con datos igualmente espaciados	47
4.3. MSE y MAPE para datos estimados con los modelos AR(1) y CAR(1) con el 90 % de los primeros datos	48
4.4. MSE y MAPE para las predicciones de los datos con los modelos AR(1) y CAR(1) con el 10 % de datos restantes (últimos datos)	48
4.5. MSE y MAPE de los modelos AR(1) y CAR(1) tanto para la base de entrenamiento como de predicción de las 1000 simulaciones, con tiempos regularmente espaciados (ver apéndice (5))	49
4.6. MSE y MAPE para datos estimados con los modelos AR(1) y CAR(1) irregularmente espaciado	51
4.7. MSE y MAPE para datos estimados con los modelos AR(1) y CAR(1) irregularmente espaciado, con el 90 % de los primeros datos	51
4.8. MSE y MAPE para datos estimados con los modelos AR(1) y CAR(1) irregularmente espaciado, con el 10 % de los últimos datos	52
4.9. MSE y MAPE de los modelos AR(1) y CAR(1) tanto para la base de entrenamiento como de predicción de las 1000 simulaciones, con tiempos desigualmente espaciados (ver apéndice (5))	52
4.10. MSE y MAPE de las estimaciones y predicciones de los datos con los modelos AR(2) y CAR(2)	54
4.11. MSE y MAPE para datos estimados con los modelos AR(2) y CAR(2) con el 90 % de los primeros datos	55
4.12. MSE y MAPE para las predicciones de los datos con los modelos AR(2) y CAR(2) con el 10 % de datos restantes (últimos datos)	55
4.13. MSE y MAPE de los modelos AR(2) y CAR(2) tanto para la base de entrenamiento como de predicción de las 1000 simulaciones, con tiempos regularmente espaciados (ver apéndice (5))	55
4.14. MSE y MAPE para datos estimados con los modelos ARMA(2,1) y CARMA(2,1) (ver apéndice (5) y (5))	60
4.15. MSE y MAPE para datos estimados con los modelos ARMA(3,1) y CARMA(3,1) (ver apéndice (5) y (5))	62
4.16. MSE y MAPE para datos estimados con los modelos ARFIMA(1,d,0) y CARFIMA(1,H,0)	66

4.17. MSE y MAPE para las estimaciones de los datos con los modelos AR-FIMA(1,0.1,0) y CARFIMA(1,0.7,0)	67
4.18. MSE y MAPE de los modelos ARFIMA(1,d,0) y CARFIMA(1,H,0) de las 1000 simulaciones, con tiempos desigualmente espaciados y parámetro $d = 0,1$ y $H = 0,7$ (ver apéndice (5) y (5))	67
4.19. MSE y MAPE para datos estimados con los modelos AR(1), CAR(1) y CAR(3), tomando base de entrenamiento y predicción (ver apéndice (5))	70
4.20. MSE y MAPE para datos estimados con los modelos AR(1), CAR(1) y CAR(2), tomando base de entrenamiento y predicción (ver apéndice (5))	73

Capítulo 1

Introducción

Actualmente, todo tipo de institución, ya sea empresa, organismos gubernamentales, negocios familiares, etc., necesitan hacer planes para el futuro si ha de sobrevivir y progresar, para lo cual, requieren conocer el comportamiento futuro de ciertos fenómenos con el fin de planificar, prever y/o prevenir. Estos comportamientos que se desean conocer a futuro suelen estar basados en lo que ha ocurrido en el pasado, por lo que es importante estudiar y analizar lo acontecido en dicho pasado; tal y como dijo el gran filósofo chino Kong Qiu [551 a.C. - 479 a.C.] creador del Confucianismo: *“Estudia el pasado si quieres pronosticar el futuro”*.

Es aquí donde se encuentra una de las técnicas más importantes para hacer inferencias sobre el futuro con base en lo ocurrido en el pasado, el análisis de series de tiempo. Existen diversos tipos de modelos relacionados con el análisis temporal de una secuencia de observaciones, como por ejemplo los modelos $ARMA(p,q)$ o $ARIMA(p,d,q)$, entre otros. Estos modelos están destinados para datos que se recopilan, observan o registran en intervalos de tiempo regulares (diario, mensual, semestral, anual, etc.) como por ejemplo: temperatura máxima diaria, ventas mensuales de una empresa, valor trimestral del PIB, etc.

Si bien existen diversos fenómenos que pueden ser medidos en intervalos de tiempo regulares, existen otros que no son de naturaleza discreta, sino más bien de tiempo continuo, ejemplo de esto son: mediciones de la función pulmonar de un paciente con asma [27], observaciones de manchas solares [21], ciclos en la temperatura de la tierra [21], análisis de datos de transacciones de IBM [21], mediciones de abundancia relativa de un isótopo de oxígeno [27], etc. Para estudiar fenómenos que posean este tipo de característica, se requieren procedimientos alternativos.

En la literatura, se han visto que los modelos de series de tiempo regularmente espaciados (ARMA, SARIMA, ARFIMA, entre otros) son una buena herramienta para modelar datos donde el tiempo entre cada observación es constante; sin embargo, estos modelos no son adecuados para datos donde los intervalos de tiempo son irregulares.

Por otro lado, no existe abundante literatura que se encargue del análisis de este tipo de datos, por lo que surge la necesidad de interiorizarse en métodos que nos permitan

modelar datos de esta naturaleza.

Por esta razón, se utilizarán los modelos de series de tiempo continuo, que son los adecuados para este tipo de datos. Entre los modelos más conocidos están: el modelo CAR(p) (Continuous Time Autoregressive Models, por sus siglas en inglés) [27], CARMA(p,q) [11, 24] (Continuous Autoregressive Moving Average) y CARFIMA(p,H,q) [23] (continuous time autoregressive fractionally integrated moving average). Estos modelos operan bajo la misma lógica que las series de tiempo discreta, tomando datos pasados, con la diferencia que no necesitan estar regularmente espaciados; es aquí la gran ventaja de estos.

La idea es ocupar estos procesos continuos de series de tiempo para modelar diversos fenómenos donde el tiempo entre observaciones no es constante.

1.1. Planteamiento del Problema

Para poder hacer cualquier tipo de estudio, es necesario contar con datos asociados al fenómeno que se desea analizar, estimar, modelar, predecir. De aquí, el tiempo entre cada observación o dato puede ser discreto (tiempo constante) o continuo (irregularmente espaciado). Por ejemplo, para los datos recopilados sobre las manchas solares, se sabe que el número de ocurrencias no es constante, sino que varía con el tiempo. Se ha observado que cada 11 años aproximadamente tiene lugar un período con mayor cantidad de manchas. Estas manchas solares son un efecto visible del aumento de la actividad magnética en la superficie solar, la que a su vez, forman una gran cantidad de viento solar, y estas pueden provocar fallos en los sistemas electrónicos y afectar a las telecomunicaciones, especialmente vía satélite, es por eso que es importante la modelación y predicciones de este fenómeno. De igual manera, en múltiples disciplinas, como Ingeniería, Astronomía, Biología, Economía [26], entre otras; podemos encontrar datos que son relevantes e interesantes para modelar/predecir.

No existe mucho conocimiento sobre el modelamiento de estos fenómenos en la literatura habitual, por lo que es necesario profundizar en modelos de series de tiempo donde el periodo transcurrido entre observaciones no es constante.

Dentro de los métodos que abordaremos para modelar datos de serie de tiempo continuo, se encuentra el modelo CAR, CARMA y CARFIMA, cada uno con sus paquetes en RStudio para la utilización de estos.

Basado en lo anterior, desde el punto de vista estadístico, se necesita estimar estos modelos para el análisis de datos irregularmente espaciados. Es por esto que se necesita implementar las series mencionadas en el software RStudio.

Cada uno de estos modelos (CAR, CARMA, CARFIMA) poseen sus propias librerías: “cts” [27] para CAR, el cual se basa en el uso del filtro de Kalman para realizar la estimación de los parámetros; “yuima” [11] para CARMA, el cual estima los parámetros usando Máxima Verosimilitud o Mínimos Cuadrados Ordinarios; y “carfima” [23] para CARFIMA, el cual estima los parámetros usando Máxima Verosimilitud o Bayesiana.

Además, cada una de estas implementaciones serán utilizadas en datos astronómicos y financieros, donde es común encontrar datos irregularmente espaciados, y poder discernir en que casos y para qué datos es mejor cada modelo, además de ocupar todos los modelos en cada una de las bases de datos; y compararlos con las series de tiempo discretas.

1.2. Objetivos

1.2.1. Objetivo General

- Implementar modelos de series de tiempo continuas en software R para modelar fenómenos irregularmente muestreados.

1.2.2. Objetivos Específicos

- Proponer y justificar modelos de series de tiempo continuo para abordar la problemática planteada.
- Implementar modelos de tiempo continuo (CAR, CARMA, CARFIMA) en software R.
- Estudiar el desempeño de los modelos de serie de tiempo continuo propuestos en datos simulados.
- Aplicar los modelos propuesto a casos reales.
- Comparar modelos propuestos con los modelos de series de tiempo regulares.

1.3. Metodología

En primera instancia, se realizará una extensa revisión bibliográfica y se estudiará la literatura pertinente sobre modelos de serie de tiempo para datos observados continuamente. Esto es necesario para obtener el conocimiento de la parte teórica necesaria para poder entender cómo se construyen este tipo de modelos junto con todos los supuestos que además debe de cumplir para poderse implementar.

Posteriormente, se indagará en las librerías que posee RStudio para la implementación de modelos de serie de tiempo para datos continuos, tales como “cts” [27], “yuima” [11] y “carfima” [23], las cuales tienen las funciones requeridas para implementar y estimar estos modelos.

Ya con el conocimiento necesario sobre la teoría e implementación en Rstudio de los modelos de series de tiempo continuo, se probará con datos simulados de tiempo continuo para comprobar el desempeño de estos.

Una vez probado estos modelos con datos simulados, se procederá a implementar en datos reales: observaciones astronómicas (brillo de las estrellas) [21] y en datos económicos (transacciones) [21] dado que estos eventos ocurren de forma irregularmente espaciadas.

Finalmente se probarán los distintos modelos de series de tiempo continuo [22] y discreto, los cuales serán analizados a fondo para poder determinar el mejor modelo posible para cada uno de las bases de datos escogidas; además de hacer una comparación entre los modelos de series de tiempo tanto continuo como discreto [25, 6] basadas en la bondad de ajuste y su poder predictivo, y así obtener resultados concluyentes de cual tipo de modelos se ajustan mejor a los datos recopilados.

Capítulo 2

Preliminares

En este capítulo se definirán conceptos claves para el entendimiento de la presente tesis, las cuales estarán basadas de diversos textos.

2.1. Series de tiempo

Una Series de Tiempo es un conjunto de observaciones sobre valores que toma una variable (cuantitativa) en diferentes momentos del tiempo. El tiempo donde los datos han sido observados pueden ser regulares (misma frecuencia entre observaciones) o irregulares. En palabras simples, “las series de tiempo son una colección de observaciones tomadas secuencialmente en el tiempo” [14].

Los datos se pueden comportar de diferentes formas a través del tiempo, puede que se presente una tendencia, un ciclo; no tener una forma definida o aleatoria, variaciones estacionales (anual, semestral, etc).

Matemáticamente, una serie de tiempo $\{Y_t\}$ corresponde a un proceso estocástico, que a su vez, se compone de variables aleatorias observadas a través del tiempo.

2.2. Procesos estocásticos

Un proceso estocástico es un conjunto de variables aleatorias $\{Y_t\}$ donde el índice t toma valores en un cierto conjunto C . En nuestro caso, este conjunto es ordenado y corresponde a los instantes temporales (días, meses, años, etc.). Para cada valor t del conjunto C (para cada instante temporal) esta definida una variable aleatoria, Y_t , y los valores observados de las variables aleatorias en distintos instantes forman una serie temporal. Es decir, una serie de T datos, $(Y_1, \dots, Y_t, \dots, Y_T)$, es una muestra de tamaño uno del vector de T variables aleatorias ordenadas en el tiempo correspondientes a los momentos $t = 1, \dots, T$ y la serie observada se considera una realización o trayectoria del proceso estocástico.

2.3. Procesos estacionarios

La base del análisis de Series de tiempo es la estacionariedad. Diremos que un proceso estocástico (serie temporal) es estacionario en sentido estricto si la distribución conjunta de $(Y_{t_1}, \dots, Y_{t_k})$ es idéntico al de $(Y_{t_1+t}, \dots, Y_{t_k+t})$ para todo t , donde k es un número entero positivo arbitrario y (t_1, \dots, t_k) es una colección de k números enteros positivos. En otras palabras, la estacionariedad estricta requiere que la distribución conjunta de $(Y_{t_1}, \dots, Y_{t_k})$ sea invariante en el cambio de tiempo. La estacionariedad estricta es una condición muy fuerte, ya que para contrastarla es necesario disponer de las distribuciones conjuntas para cualquier selección de variables del proceso.

Una propiedad más débil, pero más fácil de contrastar en la práctica, es la estacionariedad en sentido débil, que implica la estabilidad de la media, la varianza y la estructura de covarianzas a lo largo del tiempo.

Un proceso $\{Y_t\}$ es estacionario en sentido débil o de segundo orden, si para todo t :

- (a) Tiene media constante, es decir, $E(Y_t) = \mu$.
- (b) Tiene finito y constante segundo momento, es decir, $Var(Y_t) = \sigma^2$.
- (c) Existe una función $\gamma(\cdot)$ tal que $\gamma(k) = Cov(Y_t, Y_{t\pm k})$ para algún t, k [14], es decir, el valor de la covarianza entre dos periodos depende solamente de la distancia o rezago entre estos dos periodos de tiempo y no del tiempo en el cual se ha calculado la covarianza.

Por lo que se ocupará la estacionariedad débil para los procesos que definiremos más adelante.

2.4. Autocovarianza

Sea $\{Y_t\}$ una serie temporal estacionaria. La función de autocovarianza de $\{Y_t\}$ se define como:

$$\gamma(k) = Cov(Y_t, Y_{t\pm k}).$$

donde k es el retraso (lag) de la series, es decir, es el periodo que separa dos observaciones.

2.5. Autocorrelación

Sea $\{Y_t\}$ una serie temporal estacionaria. La función de autocorrelación (ACF) de $\{Y_t\}$ se define como:

$$\rho(k) = \frac{\gamma(k)}{\gamma(0)}.$$

donde k es el retraso (lag) de la series, es decir, es el periodo que separa dos observaciones.

2.6. Ruido Blanco discreto

Tanto las series de tiempo discretas como continuas está presente el ruido blanco, sólo que para las de tipo continuo, se presentan de una forma alternativa. En esta sección solo se considerará el conocido proceso de ruido blanco en tiempo discreto.

Un proceso estocástico de tiempo discreto $\{\epsilon_n, n \in \mathbb{N}_0\}$ se denomina proceso de ruido blanco (RB) discreto si tiene las siguientes propiedades:

- (i) $E(\epsilon_n) = \mu, n \in \mathbb{N}_0$
- (ii) $Var(\epsilon_n) = \sigma^2$
- iii) $Cov(\epsilon_n, \epsilon_{n+k}) = 0$

Usualmente se considera el caso de $E(\epsilon) = 0$, que se conoce como ruido blanco centrado. La notación que se ocupará es $\epsilon_n \sim RB(0, \sigma^2)$.

2.7. Modelo AR

El modelo autoregresivo de tiempo discreto del orden p , el $AR(p)$, es una herramienta ampliamente utilizada para modelar datos de series de tiempo igualmente espaciados.

Sea un proceso estocástico Y_t . Se dice autoregresivo de orden p , $AR(p)$ si

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \epsilon_t, \quad t \in \mathbb{N}, p \geq 1 \quad (2.1)$$

Donde $\epsilon_t \sim RB(0, \sigma_\epsilon^2)$, $Cov(\epsilon_t, Y_t) = 0$ para todo $j > 0$ y coeficientes ϕ_1, \dots, ϕ_p fijos. Es posible reescribir la ecuación de manera más compacta, de la forma:

$$\Phi(B)Y_t = \epsilon_t$$

donde B es definido como el operador de rezago, $BY_t = Y_{t-1}$ y $\Phi(\cdot)$ es el polinomio característico defino como

$$\Phi(z) = 1 - \phi_1 z - \phi_2 z^2 - \dots - \phi_p z^p$$

Las propiedades de este modelo pueden apreciarse para un modelo $AR(1)$. Sea Y_t un proceso autoregresivo de orden 1, $AR(1)$, es decir,

$$Y_t = \phi Y_{t-1} + \epsilon_t$$

entonces, se tiene que bajo estacionariedad:

1. $E(Y_t) = 0$.
2. $Var(Y_t) = \frac{\sigma_\epsilon^2}{1-\phi^2}$.
3. $\gamma(k) = \phi^k \gamma(0)$, con $\gamma(0) = \frac{\sigma_\epsilon^2}{1-\phi}$
4. $\rho(k) = \phi^k$

De (2) se desprende que la condición para que el proceso AR(1) sea estacionario es que $|\phi| < 1$.

2.8. Modelo ARMA

El modelo autoregresivo de medias móviles de tiempo discreto de orden p y q , ARMA(p, q), es un modelo mixto que tiene tanto componentes autoregresivas como de medias móviles.

Sea Y_t un proceso estocástico. Se dice que sigue un proceso ARMA(p, q), si

$$Y_t - \phi_1 Y_{t-1} - \dots - \phi_p Y_{t-p} = \epsilon_t - \theta_1 \epsilon_{t-1} - \dots - \theta_q \epsilon_{t-q}, \quad t \in \mathbb{N}, \quad p, q \geq 1 \quad (2.2)$$

con $\epsilon_t \sim RB(0, \sigma_\epsilon^2)$, con coeficientes $\phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q$ fijos. De igual forma, es posible reescribir en una forma más compacta

$$\Phi(B)Y_t = \Theta(B)\epsilon_t$$

donde B es el operador de rezago y $\Phi(\cdot)$ y $\Theta(\cdot)$ son los polinomios característicos definidos como

$$\begin{aligned} \Phi(z) &= 1 - \phi_1 z - \phi_2 z^2 - \dots - \phi_p z^p \\ \Theta(z) &= 1 - \theta_1 z - \theta_2 z^2 - \dots - \theta_q z^q \end{aligned}$$

De igual manera, las propiedades de este modelo es posible observarlas para el proceso ARMA(1,1). Sea Y_t un proceso ARMA(1,1), es decir,

$$Y_t - \phi Y_{t-1} = \epsilon_t - \theta \epsilon_{t-1}$$

entonces, bajo estacionariedad se tiene que:

1. $E(Y_t) = 0$.
2. $Var(Y_t) = \frac{(1-2\phi\theta+\theta^2)}{(1-\phi^2)} \sigma_\epsilon^2$
3. $\gamma(k) = \phi^{k-1} \gamma(1)$ donde $\gamma(1) = \frac{\sigma_\epsilon^2(\phi+\theta)(1+\phi\theta)}{1-\phi^2}$
4. $\rho(k) = \phi^{k-1} \rho(1)$ donde $\rho(1) = \frac{(\phi+\theta)(1+\phi\theta)}{1+2\phi\theta+\theta^2}$

Al igual que en el modelo AR(1), la condición de estacionariedad es que $\phi < 1$.

2.9. Modelo ARFIMA

Cuando se realiza un estudio de serie de tiempo es normal que se refleje la propiedad consistente en que la autocorrelación decrece exponencialmente a cero, como en los modelos ARMA o AR vistos anteriormente, ya que habitualmente cuando dos observaciones lejanas en el tiempo tienden a ser independientes. A estos modelos se le denomina de memoria corta.

Una serie de tiempo se dice que es de memoria larga cuando no presenta decaimiento exponencial en su autocorrelación, esto se distinguen porque la autocorrelación entre dos observaciones en periodos de tiempo muy grandes es alta. En otras palabras, el valor de una observación puede estar siendo influenciada por algunas observaciones pasadas bastante alejadas en el tiempo.

Otra característica no abordada hasta el momento, es que no todas las series de tiempo son precisamente procesos estacionario, existen también series no estacionarias, es decir, series que presentan una tendencia (alzas o bajas) durante todo el periodo observado. Es posible pasar de una serie con tendencia a una serie estacionaria aplicando el operador de diferencia ∇^d definido como $\nabla^d = (1 - B)^d$, para algún $d \in \mathbb{N}$, donde B es el operador de rezago.

Se dice que un proceso estocástico $\{Y_t\}$ sigue un proceso autoregresivo fraccionalmente integrado de medias móviles, ARFIMA(p, d, q) si es una solución a la ecuación:

$$\Phi(B)(1 - B)^d Y_t = \Theta(B)\epsilon_t \quad (2.3)$$

donde $\epsilon_t \sim RB(0, \sigma_\epsilon^2)$, $\Phi(B)$ y $\Theta(B)$ son los polinomios definidos en la sección (2.8). En este caso, d es un numero real no entero llamado parámetro de diferenciación fraccional. Además, es posible definir el operador de diferencias por la serie binomial ([10])

$$(1 - B)^d = \sum_{k=0}^{\infty} \frac{\Gamma(k - d)}{\Gamma(-d)\Gamma(k + 1)} B^k$$

Se dice que un proceso ARFIMA(p, d, q) es estacionario si $d \in (-0,5, 0,5)$. Es este parámetro d que determina la memoria del modelo, de forma tal que:

- Si $d \in (0, 0,5)$, la función de autocorrelación es proporcional a k^{2d-1} y decae hiperbólicamente a cero, afirmándose que el modelo exhibe una memoria de largo plazo.
- Si $d \in (-0,5, 0)$, el proceso es de memoria intermedia (antipersistencia) o dependencia negativa de largo plazo, es decir, el proceso no es de larga memoria, pero no tiene el mismo comportamiento que los procesos ARMA.
- Si $d = 0$, la memoria es de corto plazo y corresponde a un proceso ARMA(p, q).

Para observar las propiedades del proceso ARFIMA, se tomará el modelo mas sencillo, en este caso ARFIMA(0,d,0). Bajo estacionariedad, es decir, $d < 0,5$ se verifica lo siguiente:

1. $E(Y_t) = 0$
2. $Var(Y_t) = \frac{\Gamma(1-2d)}{\Gamma(1-d)^2} \sigma_\epsilon^2$, donde $\Gamma(\cdot)$ es la función gamma.
3. $\gamma(k) = \frac{\Gamma(1-2d)\Gamma(k+d)}{\Gamma(d)\Gamma(1-d)\Gamma(k+1-d)} \sigma_\epsilon^2$
4. $\rho(k) = \frac{\Gamma(1-d)\Gamma(k+d)}{\Gamma(d)\Gamma(k+1-d)} = \prod_{i=1}^k \frac{(i+d-1)}{(i-d)}$

En este caso, es posible aproximas la función de autocorrelación:

$$\rho(k) \sim \frac{\Gamma(1-d)}{\Gamma(d)} k^{2d-1} \quad (2.4)$$

cuando $k \rightarrow \infty$.

2.10. Modelos estado-espacio

Un modelo estado-espacio para una serie temporal $\{\mathbf{Y}_t\}$ consiste en dos ecuaciones. La primera, conocida como la ecuación de observación, expresa la observación \mathbf{Y}_t de w dimensiones como una función lineal de una variable de estado \mathbf{X}_t de v dimensiones más ruido. Así:

$$\mathbf{Y}_t = G_t \mathbf{X}_t + \mathbf{Z}_t, \quad t = 1, 2, \dots \quad (2.5)$$

donde $\{\mathbf{Z}_t\} \sim RB(0, R_t)$ y G_t es una matriz de $w \times v$ dimensiones. La segunda ecuación, llamada ecuación de estado, determina el estado \mathbf{X}_{t+1} en el tiempo $t + 1$ en términos del estado anterior \mathbf{X}_t y un término de ruido [5]. La ecuación de estado es:

$$\mathbf{X}_{t+1} = F_t \mathbf{X}_t + \mathbf{V}_t \quad t = 1, 2, \dots \quad (2.6)$$

donde F_t es una matriz de $v \times v$, $\{\mathbf{V}_t\} \sim RB(0, Q_t)$ y $\{\mathbf{V}_t\}$ es no correlacionado con $\{\mathbf{Z}_t\}$. Para completar la especificación, se supone que el estado inicial \mathbf{X}_1 no está correlacionado con todos los términos de ruido $\{\mathbf{V}_t\}$ y $\{\mathbf{Z}_t\}$.

Para una mayor formalidad, se tiene la siguiente definición de una estado-espacio.

Definición 2.1. Una serie temporal $\{\mathbf{Y}_t\}$ tiene una representación estado-espacio si existe un modelo estado-espacio para $\{\mathbf{Y}_t\}$ según lo especificado por las ecuaciones (2.5) y (2.6)

Es posible encontrar una representación estado-espacio para una gran número de modelos de series temporales. También se desprende claramente de la definición que ni $\{\mathbf{X}_t\}$ ni $\{\mathbf{Y}_t\}$ es necesariamente estacionario.

Para ilustrar la definición anterior, tomaremos como ejemplo el modelo autoregresivo de orden p . El proceso $AR(p)$ definido como

$$Y_{t+1} = \phi_1 Y_t + \dots + \phi_p Y_{t-p+1} + E_t \quad t = 0, 1, 2, \dots \quad (2.7)$$

Para expresar $\{\mathbf{Y}_t\}$ en forma de espacio de estado simplemente introducimos el vector de estado

$$\mathbf{X}_t = \begin{bmatrix} Y_{t-p+1} \\ Y_{t-p+2} \\ \vdots \\ Y_t \end{bmatrix}, \quad t = 0, 1, 2, \dots \quad (2.8)$$

A partir de (2.7) y (2.8) la ecuación de observación es:

$$Y_t = [0 \ 0 \ \dots \ 1] \mathbf{X}_t, \quad t = 0, 1, 2, \dots \quad (2.9)$$

mientras que la ecuación de estado viene dada por:

$$\mathbf{X}_{t+1} = \begin{bmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 1 \\ \phi_p & \phi_{p-1} & \dots & \phi_1 \end{bmatrix} \mathbf{X}_t + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} E_{t+1}, \quad t = 0, 1, 2, \dots \quad (2.10)$$

Estas ecuaciones tienen las formas requeridas (2.5) y (2.6) con $\mathbf{Z}_t = \mathbf{0}$ y $\mathbf{V}_t = (0, 0, \dots, E_{t+1})'$.

2.11. Filtro de Kalman

Las recursiones de Kalman, juegan un papel clave en el análisis de los modelos estado-espacio. Estas recursiones permiten un enfoque unificado para la predicción y estimación de todos los procesos a los que se les puede dar una representación estado-espacio [5]. La idea es poder representar los procesos continuos como un modelo estado-espacio para luego hacer una estimación usando el filtro de Kalman.

Sea las ecuaciones (2.5) y (2.6) descritas en la sección anterior:

$$\begin{aligned} \mathbf{Y}_t &= G_t \mathbf{X}_t + \mathbf{Z}_t, & \{\mathbf{Z}_t\} &\sim RB(0, R_t) \\ \mathbf{X}_{t+1} &= F_t \mathbf{X}_t + \mathbf{V}_t, & \{\mathbf{V}_t\} &\sim RB(0, Q_t) \end{aligned}$$

donde G_t y F_t son matrices de dimensiones $w \times v$ y $v \times v$, respectivamente.

Para poder definir las predicciones y las correcciones que utiliza este algoritmo, es necesario definir un par de conceptos. Sea $S_t = \{Y_1, \dots, Y_t\}$ la información disponible en el tiempo t , donde el modelo es conocido. Definimos $X_{t|j} = E(X_t | S_j)$ y $\Sigma_{t|j} = Var(X_t | S_j)$ como la media condicional y varianza condicional de X_t dado S_j ,

respectivamente. Además, sea $v_t = Y_t - Y_{t|t-1}$ y $V_t = \text{Var}(v_t|S_{t-1})$ el error de predicción a un paso y la varianza de este error de predicción dado S_{t-1} , respectivamente [26].

Ahora, dado los valores iniciales $X_{1|0}$ y $\Sigma_{1|0}$ se tiene el famoso filtro de Kalman para el modelo estado-espacio para las ecuaciones (2.5) y (2.6), el cuál, es el siguiente:

$$\begin{aligned} v_t &= \mathbf{Y}_t - G_t \mathbf{X}_{t|t-1} \\ V_t &= G_t \Sigma_{t|t-1} G_t' + R_t \\ K_t &= F_t \Sigma_{t|t-1} G_t' V_t^{-1} \\ L_t &= F_t - K_t G_t \\ \mathbf{X}_{t+1|t} &= F_t \mathbf{X}_{t|t-1} + K_t v_t \\ \Sigma_{t+1|t} &= F_t \Sigma_{t|t-1} L_t' + Q_t \end{aligned}$$

También es posible obtener las predicciones a más de un paso, la cual esta dada por:

$$\begin{aligned} \mathbf{X}_{t+h+1|t} &= F_{t+h} \mathbf{X}_{t+h|t} \\ \Sigma_{t+h+1|t} &= F_{t+h} \Sigma_{t+h|t} F_{t+h}' + Q_{t+h} \end{aligned} \tag{2.11}$$

Esta es la recursión del filtro de Kalman de la ecuación para $j + 1$ con $j = 1, \dots, h$, donde $v_{t+j} = 0$ y $K_{t+j} = 0$. El desarrollo de toda esta teoría se encuentra de manera más extensa en [26].

La idea de ocupar el filtro Kalman, aparte de las predicciones, es que nos permite poder obtener la estimación de los parámetros de un modelo estado espacio a través de la máxima verosimilitud, la cual dependerá del tipo de proceso o modelo que estemos ocupando. Se verá en la siguiente sección.

Capítulo 3

Marco Teórico

3.1. Ruido Blanco continuo

Como mencionamos anteriormente, en las series de tiempo con tiempo continuo también se encuentra presente el concepto de ruido blanco, pero en este caso, se presenta de forma continua; por lo que es necesario definir el ruido blanco continuo.

Un proceso de ruido blanco en tiempo continuo $\{\epsilon_t, t \in [0, \infty)\}$ se define como el proceso estocástico en tiempo continuo que satisface [17]:

- 1) $E(\epsilon_t) = 0$
- 2) $Cov(\epsilon_t, \epsilon_s) = \delta(t - s)\sigma_\epsilon^2$ para todo $t, s \in \mathbb{R}^+$, donde δ es la función de masa de Dirac (ver en [15])

Es necesario definir este proceso, pues está relacionado con el movimiento Browniano.

3.2. Movimiento Browniano

Es de suma importancia definir este concepto, ya que para los modelos a estudiar en esta tesis, están basados en este tipo de procesos; además de que se hará una relación entre el movimiento Browniano y el ruido blanco continuo. Se utilizará el movimiento Browniano unidimensional de valor real.

Un proceso estocástico en tiempo continuo de valor real $\{W(t), t \in [0, \infty)\}$ se conoce como movimiento Browniano si satisface lo siguiente [17]

- 1) $W(0) = 0$.
- 2) Existe una constante $\sigma_W^2 \in [0, \infty)$, llamada varianza de $\{W(t), t \in [0, \infty)\}$, de tal manera que

$$W(t) \sim N(0, t\sigma_W^2)$$

es decir, $W(t)$ tiene una distribución normal con una media 0 y varianza $t\sigma_W^2$. Cuando $\sigma^2 = 1$ decimos que el movimiento Browniano es estándar.

- 3) El proceso $\{W(t), t \in [0, \infty)\}$ tiene incrementos estacionarios, es decir, para todos $s > 0$ y $t > 0$, la distribución de $W(t+s) - W(t)$ es independiente de t .
- 4) El proceso $\{W(t), t \in [0, \infty)\}$ tiene incrementos independientes, es decir, $W(t_2) - W(t_1)$ y $W(t_3) - W(t_2)$ son independientes para cualquier número real t_1, t_2 y t_3 ; los que satisfacen

$$0 \leq t_1 \leq t_2 \leq t_3.$$

La existencia y unicidad del movimiento Browniano ha sido probada de manera concluyente (ver, [19] y [16]). Además, está relacionado con la clase de ruido blanco de tiempo continuo.

También es necesario definir el concepto de ruido blanco Gaussiano: Un ruido blanco continuo, es llamado Gaussiano si $\{\epsilon_t, t \in [0, \infty)\}$ es una familia de variables aleatorias independientes con ley común (o idénticamente distribuidas) $N(0, \sigma^2)$.

Todo lo definido anteriormente, es para relacionar el movimiento Browniano con el ruido blanco de tiempo continuo, el cual, queda de manifiesto en el siguiente teorema.

Teorema 3.2.1. *Suponga que $\{\epsilon(t)\}$ es un proceso de ruido blanco continuo. El proceso estocástico de tiempo continuo $\{W(t)\}$ está definido por*

$$W(t) = \int_0^t \epsilon(u) du, \quad t \in [0, \infty) \quad (3.1)$$

donde $\{W(t)\}$ es un movimiento Browniano.

La integral (3.1) no es una integral de Riemann (o Riemann integrable). Esta integral esta definida mediante un limite en media cuadrática. Este tipo de integral en particular, se le conoce como una integral de Itô. Una integral de Itô se define de la siguiente manera:

$$\int_0^t Z(u) dW(u) = \lim_{n \rightarrow \infty, \Delta t \rightarrow 0} Z(k\Delta t) [W((k+1)\Delta t) - W(k\Delta t)] \quad (3.2)$$

La relación que existe entre el movimiento Browniano y el ruido blanco gaussiano es fundamental para poder definir los modelos continuos: autoregresivos (CAR), autoregresivos de medias móviles (CARMA) y autoregresivos fraccionalmente integrados de medias móviles CARFIMA; que serán presentados en este capítulo. En [17] es posible ver que efectivamente se cumplen los 4 puntos mencionados para que $\{W(t)\}$ sea un movimiento Browniano.

Como se ha señalado, este trabajo constará en describir, analizar y concluir sobre los modelos de series de tiempo continuos. Toda la información descrita ha sido recopilada de variados textos que abordaban ampliamente estos procesos.

3.3. Modelos CAR(p)

El proceso continuo autoregresivo CAR posee una estructura bastante similar a su homólogo discreto AR, con la diferencia que se introduce la derivada estocástica.

Sean p un número entero no negativo. El proceso CAR(p) es definido como la solución de la ecuación diferencial estocástica:

$$Y^{(p)}(t) + \alpha_1 Y^{(p-1)}(t) + \dots + \alpha_{p-1} Y^{(1)}(t) + \alpha_p Y(t) = \epsilon(t) \quad (3.3)$$

Donde $Y^{(i)}(t)$ es la i -ésima derivada de $Y(t)$ y $\epsilon(t)$ es la derivada formal de un proceso Browniano $W(t)$.

Podemos reescribir la ecuación (3.3) bajo la notación $\alpha(D)Y(t) = \epsilon(t)$, donde

$$\alpha(D) = D^p + \alpha_1 D^{p-1} + \dots + \alpha_{p-1} D + \alpha_p \quad (3.4)$$

Donde D es el operador de derivada. La ecuación característica correspondiente viene dada por:

$$\alpha(s) = s^p + \alpha_1 s^{p-1} + \dots + \alpha_{p-1} s + \alpha_p = 0 \quad (3.5)$$

Más adelante, se presentará la condición necesaria para que el proceso sea estacionario.

3.3.1. CAR(1)

A medida que orden de un proceso autoregresivo continuo aumenta, los cálculos y propiedades de este se vuelven difíciles y complejos. Es por esto, que para apreciar las propiedades del proceso CAR nos centraremos en el modelo autoregresivo continuo de orden 1, AR(1), que se define como una solución estacionaria de la ecuación diferencial estocástica de primer orden:

$$dY(t) + \alpha Y(t) = \sigma dW(t) \quad (3.6)$$

Luego, la solución para (3.6) está dada por:

$$Y(t) = e^{-\alpha t} Y(0) + \sigma \int_0^t e^{-\alpha(t-u)} dW(u) \quad (3.7)$$

o equivalentemente

$$Y(t) = e^{-\alpha t} Y(0) + e^{-\alpha t} I(t) \quad (3.8)$$

donde $I(t)$ es conocida como la integral de Itô, definida como $I(t) = \sigma \int_0^t e^{\alpha u} dW(u)$. Para que el proceso CAR(1) sea estacionario, basta que $\alpha > 0$. Si el proceso CAR(1) es estacionario, se cumple que:

1. $E(Y(t)) = 0$
2. $Var(Y(t)) = \frac{\sigma^2}{2\alpha}$
3. $\gamma(h) = \frac{\sigma^2}{2\alpha} e^{-\alpha|h|}$
4. $\rho(h) = e^{-\alpha|h|}$

Otra de las propiedades que cumple un proceso CAR es que es posible discretizar las observaciones para poder ajustarlo como un modelo AR, en otras palabras, un proceso CAR de muestreo regular es también un proceso AR. Como ejemplo, tomemos el proceso CAR(1) definido en (3.6). Si se observa regularmente en intervalos $(t, t + \Delta, t + 2\Delta, \dots)$ se obtendrá el siguiente proceso AR(1) [5] :

$$Y_t = e^{-\alpha} Y_{t-1} + \epsilon_t \quad \epsilon_t \sim RB \left(0, \frac{\sigma^2}{2\alpha} (1 - e^{-2\alpha}) \right)$$

es decir, $\phi = \exp(-\alpha)$. Obviamente, de aquí se deduce que $\phi > 0$. Por lo tanto, un proceso AR(1) con un negativo ϕ no puede ser un proceso CAR(1). De aquí es donde se desprende que un proceso CAR observado en tiempo regulares puede ser escrito como un proceso AR. Por otra parte lo contrario no es válido, es decir, existen procesos AR que no son una versión discreta de un CAR en particular.

3.3.2. Reparametrización CAR

Belcher, Hampton y Tunnicliffe Wilson [2] modificaron la parametrización y la estructura del modelo CAR(p) de tal manera que se restableció la capacidad general de aproximación, requiriendo solo una modificación relativamente menor del procedimiento de estimación de Jones [12].

Definimos una forma de escala como

$$s = -k \frac{1 - z}{1 + z} \quad (3.9)$$

con transformación inversa

$$z = \frac{1 + s/k}{1 - s/k} \quad (3.10)$$

El primer paso para definir una nueva parametrización es mostrar cómo se pueden construir los antiguos parámetros α_i a partir de los nuevos parámetros ϕ_i . Considere el polinomio en z^{-1} , $\phi(z^{-1}) = 1 + \phi_1 z^{-1} + \dots + \phi_p z^{-p}$. Al sustituir z por s se obtiene una relación de polinomios:

$$\phi(z^{-1}) = \frac{\beta(s)}{(1 + s/k)^p} \quad (3.11)$$

donde

$$\beta(s) = \beta_0 s^p + \beta_1 s^{p-1} + \dots + \beta_{p-1} s + \beta_p = \sum_{i=0}^p \phi_i (1 - s/k)^i (1 + s/k)^{p-i} \quad (3.12)$$

y tomamos $\phi_0 = 1$. Así, las β_i son combinaciones lineales de las ϕ_i . Ahora, se tiene que $\alpha(s) = \beta(s)/\beta_0$ y $\alpha_i = \beta_i/\beta_0$. Esto completa la definición.

En base a esta parametrización, se propone que el modelo (3.3) se modifique de la siguiente manera [2]:

$$\phi(D)Y(t) = \left(1 + \frac{D}{k}\right)^{p-1} \epsilon(t) \quad (3.13)$$

con el parámetro de escalamiento $k > 0$. Esta parametrización es necesaria, puesto que el paquete a utilizar en R para hacer la estimación de los parámetros es en base al modelo (3.13)

En la práctica se ha encontrado el modelo (3.13) para ajustar los datos bastante bien sin la necesidad de un término de error de observación (ver [2]).

Para el caso $p = 1$, es decir, un CAR(1), tendremos el modelo modelo sin reparametrizar (3.14) y el modelo reparametrizado (3.15), definidos como

$$Y'(t) + \alpha_1 Y(t) = \epsilon(t) \quad (3.14)$$

$$Y'(t) + \phi_1 Y(t) = \epsilon(t) \quad (3.15)$$

El paquete cts (el cual modela datos CAR) estima el parámetro ϕ_i , que corresponde al modelo reparametrizado. Para conocer el coeficiente del modelo no reparametrizado (en este caso α_1) se debe conocer la relación entre ϕ_1 y α_1 . Como tomamos $p = 1$, se tiene lo siguiente:

$$\begin{aligned} \phi(z^{-1}) &= 1 + \phi_1 z^{-1} = 1 + \phi_1 \left(\frac{1 - s/k}{1 + s/k}\right) \\ \beta(s) &= \beta_0 s + \beta_1 \end{aligned} \quad (3.16)$$

Luego, se tiene que

$$\begin{aligned} 1 + \phi_1 \left(\frac{1 - s/k}{1 + s/k}\right) &= \frac{\beta_0 s + \beta_1}{(1 + s/k)} \\ 1 + \frac{s}{k} + \phi_1 \left(1 - \frac{s}{k}\right) &= \beta_0 s + \beta_1 \\ \left(\frac{1}{k} - \frac{\phi_1}{k}\right) s + 1 + \phi_1 &= \beta_0 s + \beta_1 \end{aligned}$$

Comparando términos, se obtiene

$$\begin{aligned} \beta_0 &= \frac{1}{k}(1 - \phi_1) \\ \beta_1 &= 1 + \phi_1 \end{aligned} \quad (3.17)$$

Por lo tanto, se llega a que

$$\alpha_1 = \beta_1/\beta_0 = k \left(\frac{1 + \phi_1}{1 - \phi_1}\right) \quad (3.18)$$

Esta es la relación entre α_1 y ϕ_1 para $p = 1$.

3.3.3. Estimación vía filtro de Kalman

Ya definido el modelo de series de tiempo continuo autoregresivo CAR, es necesario estimar los parámetros del modelo (3.13). Para esto, se aplicará el filtro Kalman, siguiendo a Jones [12] y Belcher [2]. Para aplicar el filtro Kalman, se requiere reescribir el modelo (3.13) a una forma estado-espacial, que puede encontrarse en Wiberg [28]. Sea el vector de estado $\mathbf{X}(t) = (X(t), X'(t), X''(t), \dots, X^{(p-1)}(t))^T$ y $\mathbf{X}'(t)$ la primera derivada de $\mathbf{X}(t)$. La ecuación de estado viene dada por

$$\mathbf{X}'(t) = A\mathbf{X}(t) + R\epsilon \quad (3.19)$$

donde

$$A = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 1 \\ -\alpha_p & -\alpha_{p-1} & \cdots & -\alpha_1 \end{bmatrix} \quad (3.20)$$

y

$$R^T = [0 \quad 0 \quad \cdots \quad 1] \quad (3.21)$$

El proceso Y_t es estacionario si, y sólo si todos los valores propios de A tienen partes reales negativas. Bajo lo definido anteriormente, es posible escribir:

$$Y_k = H\mathbf{X}(t_k) \quad (3.22)$$

donde los elementos del vector $H_{1 \times p}$ vienen dados por

$$H_i = \binom{p-1}{i-1} / k^{i-1} \quad i = 1, \dots, p \quad (3.23)$$

Ahora, supongamos que A puede ser diagonalizado por $A = UDU^{-1}$, donde

$$U = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ r_1 & r_2 & \cdots & r_p \\ r_1^2 & r_2^2 & \cdots & r_p^2 \\ \vdots & \vdots & \ddots & \vdots \\ r_1^{p-1} & r_2^{p-1} & \cdots & r_p^{p-1} \end{bmatrix} \quad (3.24)$$

donde r_1, \dots, r_p son las raíces de $\alpha(s)$, y \mathbf{D} es una matriz diagonal con estas raíces como sus elementos diagonales. En este caso, sea $\mathbf{X} = U\psi$, la ecuación de estado se convierte en:

$$\psi' = \mathbf{D}\psi + J\epsilon \quad (3.25)$$

con $J = U^{-1}R$. En consecuencia, la ecuación (3.22) se convierte en:

$$Y_k = G\psi(t_k) \quad (3.26)$$

donde $G = HU$. La condición necesaria y suficiente para la diagonalización de A es que A debe tener valores propios distintos.

Para iniciar las recursiones del filtro Kalman, se requieren condiciones iniciales. Para un modelo estacionario, la matriz de covarianza no condicional del vector de estado $\mathbf{X}(t)$ es conocida [7] y utilizada en Jones [12] y Harvey [9]. El estado inicial tanto para la versión estándar como para la versión diagonalizada se puede establecer como $\mathbf{X}(0) = 0$ y $\psi_0 = 0$, respectivamente. La matriz de covarianza estacionaria Q satisface

$$Q = \sigma^2 \int_0^\infty e^{As} R R^T e^{A^T s} ds \quad (3.27)$$

Si A puede ser diagonalizado, se tendrá la covarianza

$$Q_{\psi_{i,j}} = -\sigma^2 \frac{J_i \bar{J}_j}{r_i + \bar{r}_j} \quad (3.28)$$

donde \bar{J}_j y \bar{r}_j son los conjugados complejos de J_j y r_j , respectivamente.

El parámetro de escala k puede ser elegido aproximadamente como el recíproco del tiempo medio entre observaciones. El algoritmo de filtro Kalman para la forma diagonal se presenta a continuación. Comenzando con un vector de estado estacionario inicial $\psi_0 = \psi(0|0) = 0$ y la matriz de covarianza estacionaria de estado inicial Q_ψ (3.28), la recursión procede de la siguiente manera:

1. Predecir el estado. Sea

$$T_{\psi,k} = e^{\mathbf{D}\delta_k}$$

una matriz diagonal, entonces

$$\psi(t_k|t_{k-1}) = T_{\psi,k} \psi(t_{k-1}|t_{k-1}).$$

2. Calcular la matriz de covarianzas de esta predicción:

$$P_\psi(t_k|t_{k-1}) = T_{\psi,k} [P_\psi(t_{k-1}|t_{k-1}) - Q_\psi] \bar{T}_{\psi,k} + Q_\psi.$$

3. Predecir la observación en el tiempo t_k :

$$Y_\psi(t_k|t_{k-1}) = G\psi(t_k|t_{k-1})$$

4. Calcular la innovación:

$$v_\psi(t_k) = Y_\psi(t_k) - Y_\psi(t_k|t_{k-1})$$

y varianza

$$F_\psi(t_k) = G P_\psi(t_k|t_{k-1}) \bar{G}^T + V$$

5. Actualizar la estimación del vector de estado:

$$\psi(t_k|t_k) = \psi(t_k|t_{k-1}) + P_\psi(t_k|t_{k-1}) \bar{G}^T F_\psi^{-1}(t_k) v_\psi(t_k)$$

6. Actualizar la matriz de covarianzas:

$$P_\psi(t_k|t_k) = P_\psi(t_k|t_{k-1}) - P_\psi(t_k|t_{k-1}) \bar{G}^T F_\psi^{-1}(t_k) G P_\psi^T(t_k|t_{k-1})$$

7. El factor de escala desconocido σ^2 puede ser compensado dejando $\sigma^2 = 1$ temporalmente. La log-verosimilitud es calculada por:

$$\log L_{\psi, k} = \sum_{t=1}^n \log F_{\psi}(t_k) + n \log \sum_{t=1}^n \frac{v_{\psi}^2(t_k)}{F_{\psi}(t_k)}$$

La función logarítmica de la verosimilitud (25) puede ser evaluada mediante una aplicación recursiva del método filtro Kalman, y se utiliza una rutina de optimización numérica no lineal para determinar la estimación de parámetros. El factor de escala desconocido puede entonces estimarse mediante

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{t=1}^n \frac{v_{\psi}^2(t_k)}{F_{\psi}(t_k)}.$$

3.3.4. Selección del modelo

Para identificar el orden del modelo, Belcher [2] propuso comenzar con un modelo de orden grande, para luego construir la variable secuencia $AIC_d = \sum_{i=1}^d t_{\phi, i}^2 + 2d$ para $d = 1, \dots, p$. El mínimo valor del AIC_d sugiere un mejor modelo. Sin embargo, Belcher [2] sugiere que la mejor práctica es calcular el AIC o BIC clásico (Akaike [1]; Schwarz [20]).

En nuestro caso, para escoger el mejor modelo, nos guiaremos según el error cuadrático medio (MSE) y error porcentual absoluto medio (MAPE), los cuales serán definidos y detallados más adelante.

3.4. Modelos CARMA(p, q)

Un proceso continuo autoregresivo de medias móviles (CARMA) sigue básicamente la misma estructura que un modelo ARMA, salvo que se presentan las derivadas de orden superior del movimiento Browniano. Sean p, q números enteros no negativos tales que $p > q \geq 0$. El proceso CARMA(p, q) es definido como la solución de la ecuación diferencial estocástica:

$$\alpha(D)Y(t) = \beta(D)DW(t), \quad t \geq 0, \quad (3.29)$$

donde D denota la diferenciación con respecto a t , $\{W(t)\}$ es el movimiento Browniano estándar, con

$$\begin{aligned} \alpha(z) &= z^p + \alpha_1 z^{p-1} + \dots + \alpha_p \\ \beta(z) &= \beta_0 + \beta_1 z + \dots + \beta_p z^p \end{aligned} \quad (3.30)$$

y los coeficientes β_j satisfacen $\beta_q \neq 0$ y $\beta_j = 0$ para $q < j \leq p$. La condición de estacionariedad del ARMA requiere que las raíces del polinomio $\Phi(z) = 1 - \phi_1 z - \dots - \phi_p z^p$ estén fuera del círculo de la unidad. La condición de estacionariedad del CARMA requiere que las raíces del polinomio $\alpha(z) = z^p + \alpha_1 z^{p-1} + \dots + \alpha_p$ tengan partes reales

negativas y que $p > q$. Dado que las derivadas $D^j W(t)$ no existen en el contexto habitual, interpretamos (3.29) como el equivalente a las ecuaciones de observación y de estado

$$Y(t) = \mathbf{b}'\mathbf{X}(t) \quad (3.31)$$

y

$$d\mathbf{X}(t) = A\mathbf{X}(t)dt + \mathbf{e}dW(t) \quad (3.32)$$

donde

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\alpha_p & -\alpha_{p-1} & -\alpha_{p-2} & \cdots & -\alpha_1 \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-2} \\ \beta_{p-1} \end{bmatrix}$$

y asumimos que $\mathbf{X}(0)$ es un vector aleatorio normal tal que

$$\mathbf{X}(0) \text{ es independiente de } \{W(t), t \geq 0\} \quad (3.33)$$

La ecuación estado (3.32) es una Itô ecuación diferencial para $\mathbf{X}(t)$. Si $p = 1$, A es definido como $-\alpha_1$. Debido a la linealidad de (3.32), su solución tienen la forma simple

$$\mathbf{X}(t) = e^{At}\mathbf{X}(0) + \int_0^t e^{A(t-\mu)}\mathbf{e}dW(\mu) \quad (3.34)$$

El proceso $\{\mathbf{X}(\mu), \mu \geq 0\}$ también satisface la relación

$$\mathbf{X}(t) = e^{A(t-s)}\mathbf{X}(s) + \int_s^t e^{A(t-\mu)}\mathbf{e}dW(\mu) \quad (3.35)$$

donde la matriz exponencial e^A es definida como una serie de potencias

$$e^A = \sum_{h=0}^{+\infty} \frac{1}{h!} A^h \quad (3.36)$$

Se sabe (ver [3]) que la ecuación (3.35) tiene una solución débilmente estacionaria si, y sólo si los valores propios r_1, \dots, r_p de A (que son los ceros del polinomio $\alpha(z) = z^p + \alpha_1 z^{p-1} + \dots + \alpha_p$) tienen todos parte real negativas, es decir, si y sólo si

$$\Re(r_i) < 0, \quad i = 1, \dots, p \quad (3.37)$$

A partir de (3.31) la media y la función de autocovarianza de un proceso $\{Y(t)\}$ son las siguientes:

$$E[Y(t)] = 0, \quad t \geq 0$$

$$\gamma(h) = E[Y(t+h)Y(t)] = \mathbf{b}'e^{A|h|}\Sigma\mathbf{b}$$

donde Σ está dado por [4]:

$$\Sigma = E[\mathbf{X}(0)\mathbf{X}'(0)] = \int_0^\infty e^{Ay} \mathbf{e}\mathbf{e}' e^{A'y} dy \quad (3.38)$$

Por lo tanto, la función de autocorrelación está dada por [4]:

$$\rho(h) = \frac{\mathbf{b}' e^{A|h|} \Sigma \mathbf{b}}{\mathbf{b}' \Sigma \mathbf{b}} \quad (3.39)$$

Un proceso CARMA(p, q) de muestreo regular es también un proceso ARMA, tal como se mostró en la sección que un proceso CAR(p) puede ser representado como un AR(p). De la misma forma, no todo proceso ARMA puede ser representado como un CARMA.

3.4.1. CARMA(2,1)

Para poder ver las propiedades de este modelo, nos centraremos en el proceso continuo autoregresivo de medias móviles con parámetros $p = 2$ y $q = 1$ que es el modelo más indicado, ya que $p > q$. El modelo CAR(2,1) se define como la solución de la ecuación diferencial estocástica

$$Y^{(2)}(t) + \alpha_1 Y^{(1)}(t) + \alpha_2 Y(t) = \beta_0 W^{(1)}(t) + \beta_1 W^{(2)}(t) \quad t \leq 0 \quad (3.40)$$

Para que tal solución exista es necesario que las raíces de la ecuación $z^2 + \alpha_1 z + \alpha_2 = 0$ tengan parte real negativa. Bajo esta condición, el proceso $\{Y_t\}$ también es estacionario. Cumpliéndose la condición de estacionariedad y además de que las raíces de la ecuación son distintas, se tiene las siguientes propiedades:

1. $E(Y(t)) = 0$
2. $Var(Y(t)) = \frac{-(\beta_0^2 + \alpha_1 \alpha_2)}{2\alpha_1 \alpha_2 (\alpha_1 + \alpha_2)} \sigma_\epsilon^2$
3. $\gamma(h) = \frac{[\alpha_2(\beta_0^2 - \alpha_1^2)e^{\alpha_1|h|} - \alpha_1(\beta_0^2 - \alpha_2^2)e^{\alpha_2|h|}]}{2\alpha_1 \alpha_2 (\alpha_1^2 - \alpha_2^2)} \sigma_\epsilon^2$
4. $\rho(h) = \frac{[\alpha_1(\beta_0^2 - \alpha_2^2)e^{\alpha_2|h|} - [\alpha_2(\beta_0^2 - \alpha_1^2)e^{\alpha_1|h|}]}{(\alpha_1 - \alpha_2)(\beta_0^2 + \alpha_1 \alpha_2)}$

3.4.2. Estimación del modelo

Para esta sección se describirá el procedimiento de estimación de un modelo CARMA, implementado en el paquete *yuima* de Rstudio. Todo el resultado que veremos a continuación se puede ver más detalladamente en [11]. De ahora en adelante, asumimos que la condición de que los valores propios son distintos y negativos para una matriz está cumplida.

A continuación, asumimos que las observaciones $Y_1, \dots, Y_n, \dots, Y_T$ se recopilan en instantes igualmente espaciados de tiempo $0, h, 2h, \dots, Nh$ donde N es el número de observaciones y h es la longitud del paso donde los datos han sido tomados. En este

contexto, el horizonte temporal T donde han sido observados es igual a Nh .

Luego, definimos un proceso de media corregida Y_n como:

$$Y_n^* = Y_n - E(Y_n). \quad (3.41)$$

Finalmente, el proceso Y_n^* tiene una nueva representación estado-espacio:

$$\begin{aligned} Y_n^* &= \mathbf{b}^T \mathbf{X}_n^* \\ \mathbf{X}_n^* &= e^{Ah} \mathbf{X}_n^* + U_n \end{aligned} \quad (3.42)$$

donde \mathbf{X}_n^* es una media muestral corregida del proceso vectorial de estado \mathbf{X}_t en el sistema (3.32), e^A es la matriz exponencial de A y U_n es una secuencia de vectores aleatorios i.i.d. con media cero y matriz de varianzas-covarianza:

$$Q = \int_0^h e^{Au} \mathbf{e} \mathbf{e}^T e^{A^T u} du \quad (3.43)$$

Antes de introducir el algoritmo del Filtro Kalman, necesitamos calcular la matriz Q en (3.43). Comenzamos evaluando la matriz de varianzas-covarianzas estacionaria no condicionada Q_∞ que satisface el sistema de ecuaciones (más detalle ver en [24]):

$$AQ_\infty + Q_\infty A^T = -\sigma^2 \mathbf{e} \mathbf{e}^T \quad (3.44)$$

done la matriz Q es obtenida usando la siguiente formula:

$$Q = Q_\infty - e^{Au} Q A^T u \quad (3.45)$$

Usando este resultado, el algoritmo del Filtro Kalman nos da una manera simple y analítica de calcular la función de probabilidad. El procedimiento de estimación basado en el Filtro Kalman puede resumirse en cuatro pasos: inicialización, predicción, corrección y construcción de la función logarítmica de la verosimilitud.

Antes de explicar los pasos en el algoritmo del Filtro Kalman necesitamos aclarar las notaciones utilizadas. Definimos $\mathbf{X}_{n|n-1}^*$ y $\Sigma_{n|n-1}$ las estimaciones previas del proceso de las variables de estado y la matriz de varianzas-covarianzas del término de error U_n , es decir

$$\begin{aligned} \mathbf{X}_{n|n-1}^* &= E[\mathbf{X}_n^* | I_{n-1}] \\ \Sigma_{n|n-1} &= Var[U_n | I_{n-1}] \end{aligned}$$

donde el σ -álgebra I_{n-1} (ver en [29]) es generado por las observaciones del proceso Y_n^* y por las estimaciones de las variables del espacio de estado hasta el tiempo n :

$$I_{n-1} = \sigma(Y_{n-1}^*, \dots, Y_0^*, \mathbf{X}_{n-1|n-1}^*, \dots, \mathbf{X}_{0|0}^*).$$

En este caso, las estimaciones se obtienen de acuerdo con el aumento del σ -álgebra I_n^* :

$$I_n^* = \sigma(\{I_{n-1}\} \cup \{Y_n^*, \mathbf{X}_{n|n-1}^*\}).$$

- **Inicialización**

Inicializamos la variable de estado \mathbf{X}_n^* en cero, ya que bajo el supuesto de que los valores propios de la Matriz A son distintos con parte real negativa, la media no condicionada $E[\mathbf{X}_n^*]$ es igual a cero, mientras que la matriz de varianza-covarianza se inicializa en la matriz de varianza-covarianza no codificada Q_∞ . Finalmente, nos queda que:

$$\begin{aligned}\mathbf{X}_{0|0}^* &= 0 = \mathbf{X}_{n-1|n-1}^* \\ \Sigma_{0|0} &= Q_\infty = \Sigma_{n-1|n-1}\end{aligned}$$

- **Predicción**

Empezamos por predecir el proceso inobservable $\mathbf{X}_{n|n-1}^*$ y la matriz de varianza-covarianza $\Sigma_{n|n-1}$

$$\begin{aligned}\mathbf{X}_{n|n-1}^* &= e^{Ah} \mathbf{X}_{n-1|n-1}^* \\ \Sigma_{n|n-1} &= e^{Ah} \Sigma_{n-1|n-1} e^{ATh} + Q\end{aligned}\tag{3.46}$$

entonces podemos predecir el proceso observable Y_n :

$$Y_{n|n-1}^* = \mathbf{b}^T \mathbf{X}_{n|n-1}^*.$$

Definimos el termino error u_n como:

$$u_n = Y_n^* - \mathbf{b}^T \mathbf{X}_{n|n-1}^*$$

donde Y_n^* es la media corregida observada del proceso definido en (3.41). Así, el término de error u_n es normalmente distribuido

$$u_n \sim N(0, \mathbf{b}^T \Sigma_{n|n-1} \mathbf{b})$$

y usamos este resultado para construir la función de log-verosimilitud más adelante.

- **Corrección**

Necesitamos actualizar la variable de estado \mathbf{X}^* y la matriz de varianza-covarianza Σ ya que observamos la realización del proceso Y_n^* .

$$\begin{aligned}\mathbf{X}_{n|n}^* &= \mathbf{X}_{n|n-1}^* + K_n (Y_n^* - Y_{n|n-1}^*) \\ \Sigma_{n|n} &= \Sigma_{n|n-1} - K_n \mathbf{b}^T \Sigma_{n|n-1}\end{aligned}$$

donde K_n es la Matriz de Aumento de Kalman y se define como:

$$K_n = \Sigma_{n|n-1} \mathbf{b} (\mathbf{b}^T \Sigma_{n|n-1} \mathbf{b})^{-1}.$$

Ahora, utilizamos la variable de estado actualizada $\mathbf{X}_{n|n}$ y la matriz de varianza-covarianza $\Sigma_{n|n}$ como entradas en (3.46) y repetimos los pasos 1, 2 y 3 hasta $n = N$.

▪ **Construcción de la función log-verosimilitud**

Una vez que todos los términos $\{u_n, n = 1, \dots, N\}$ son obtenidos, se calcula la función logarítmica de la verosimilitud:

$$\mathcal{L}(\mathbf{a}, \mathbf{b}) = -\frac{1}{2} \sum_{n=1}^N \ln(2\pi \mathbf{b}^T \Sigma_{n|n-1} \mathbf{b}) - \frac{1}{2} \sum_{n=1}^N \frac{u_n^2}{\mathbf{b}^T \Sigma_{n|n-1} \mathbf{b}} \quad (3.47)$$

y obtener las estimaciones para los vectores \mathbf{a} y \mathbf{b} . En el paquete *yuima* también está disponible la optimización forzada.

3.5. Modelos CARFIMA(p,H,q)

Toda la información presentada a continuación estará basada del texto [23] y [25].

Recientemente, Tsai y Chan [25] propusieron el modelo de tiempo continuo autoregresivo fraccionadamente integrado de media móvil (CARFIMA), que se define como la solución estacionaria de una ecuación diferencial estocástica impulsada por el movimiento Browniano fraccionario estándar, con parámetro de Hurst H y $t \geq 0$:

$$Y_t^{(p)} - \alpha_p Y_t^{(p-1)} - \dots - \alpha_1 Y_t = \sigma \{B_{t,H}^{(1)} + \beta_1 B_{t,H}^{(2)} + \dots + \beta_q B_{t,H}^{(q+1)}\} \quad (3.48)$$

donde $\{B_{t,H} = B_t^H, t \geq 0\}$ es el movimiento Browniano fraccional estándar con parámetro de Hurst $0 < H < 1$, $Y_t^{(i)}$ es la i -ésima derivada del proceso $\{Y_t\}$ con respecto a t y $p > q \geq 0$. Asumimos que $\sigma > 0$, $\alpha_1 \neq 0$ y $\beta_q \neq 0$. Para $H = 1/2$, el proceso CARFIMA se convierte en un proceso autoregresivo de media móvil de tiempo continuo (CARMA).

Sea $0 < H < 1$ un número fijo. Es posible mostrar [8] que existe un proceso estocástico Gaussiano que satisface las tres siguientes propiedades:

1. Su estado inicial es $B_0^H = 0$.
2. $E(B_t^H) = 0$ para todo $t \geq 0$
3. Su covarianza se define como

$$Cov(B_t^H, B_s^H) = E(B_t^H B_s^H) = \frac{1}{2} (|t|^{2H} + |s|^{2H} - |t - s|^{2H})$$

El proceso Gaussiano $\{B_t^H\}$ es llamado movimiento Browniano fraccionario estándar con parámetro de Hurst H . Para $H > 1/2$, los incrementos del movimiento Browniano fraccionario estándar, $\{B_t^H - B_{t-1}^H\}$, muestran dependencia de largo alcance, por lo que de ahora en adelante el parámetro Hurst H es fijo y $1/2 < H < 1$ ya que es esta la condición que nos interesa si queremos modelar datos de larga memoria.

Como el movimiento Browniano fraccional no es diferenciable [13], por lo que la ecuación estocástica (3.48) tiene que ser escrita de manera análoga al proceso CARMA. La ecuación (3.48) puede ser equivalente en términos de la ecuación observada y estado:

$$\begin{aligned} Y_t &= \mathbf{b}'\mathbf{X}_t \quad t \geq 0 \\ d\mathbf{X}_t &= A\mathbf{X}_t dt + \sigma\delta_p dB_t^H \end{aligned} \quad (3.49)$$

donde

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_p \end{bmatrix}, \quad \mathbf{X}_t = \begin{bmatrix} X_t^{(0)} \\ X_t^{(1)} \\ \vdots \\ X_t^{(p-2)} \\ X_t^{(p-1)} \end{bmatrix}, \quad \delta_p = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ \beta_1 \\ \vdots \\ \beta_{p-2} \\ \beta_{p-1} \end{bmatrix}$$

y $\beta_j = 0$ para $j > q$.

Se dice que el proceso $\{Y_t, t \leq 0\}$ es un proceso CARFIMA(p, H, q) si $Y_t = \mathbf{b}'\mathbf{X}_t$, donde \mathbf{X}_t es la solución de (3.52) para la condición inicial \mathbf{X}_0 . La solución para (3.52) está dada por:

$$\mathbf{X}_t = e^{At}\mathbf{X}_0 + \sigma \int_0^t e^{A(t-u)}\delta_p dB_u^H \quad (3.50)$$

donde e^A es la matriz exponencial definida en (3.36). Para una condición inicial aleatoria \mathbf{X}_0 , el vector medio de $\{\mathbf{X}_t\}$ denotado por $\mu_{X,t}$, satisface:

$$\mu_{X,t} = e^{At}\mu_{X,0} \quad (3.51)$$

De la ecuación (3.48) con una condición inicial, admite una solución asintóticamente estacionaria si y sólo si todos los valores propios de A tienen partes reales negativas.

Si todos los valores propios de A tienen partes reales negativas, entonces de las ecuaciones (3.52) y (3.50) se desprende que, cuando $t \rightarrow \infty$, Y_t converge en distribución a una variable aleatoria normal con media 0 y varianza V_Y . La varianza V_Y puede sacarse de la función de autocovarianza $\gamma_Y(h)$ definida como:

$$\begin{aligned} \gamma_Y(h) &= C_H \mathbf{b}' e^{Ah} \left(\int_0^h e^{-Au} u^{2H-2} du \right) V^* \mathbf{b} \\ &+ C_H \mathbf{b}' e^{-Ah} \left(\int_h^\infty e^{Au} u^{2H-2} du \right) V^* \mathbf{b} \\ &+ C_H \mathbf{b}' e^{Ah} \left(\int_0^\infty e^{Au} u^{2H-2} du \right) V^* \mathbf{b} \end{aligned} \quad (3.52)$$

para $H \in [(0, 1/2) \cup (1/2, 1)]$, donde $C_H = H(2H - 1)$ y

$$V^* = \sigma^2 \int_0^\infty e^{Au} \delta_p \delta_p' e^{A'u} du.$$

De la autocovarianza, es posible desprender que la varianza de $Y(t)$ esta dada por:

$$\text{Var}(Y(t)) = 2C_H \mathbf{b}' \left(\int_0^\infty e^{Au} u^{2H-2} du \right) V^* \mathbf{b} \quad (3.53)$$

Si $H \neq 1/2$, cuando $h \rightarrow \infty$, tenemos la expresión asintótica

$$\gamma_Y(h) \sim \sigma^2 H(2H-1) \frac{\beta^2(0)}{\alpha^2(0)} h^{2H-2} \quad (3.54)$$

3.5.1. Estimación del modelo

Sea Y_t una serie de tiempo con intervalos de tiempo desiguales. Bajo el supuesto de estacionariedad, la función de log-verosimilitud de Y_t bajo los parámetros $(\theta, \sigma) = (\alpha_1, \dots, \alpha_p, \beta_1, \dots, \beta_q, H, \sigma)$ es:

$$\mathcal{L}_Y(\theta, \sigma^2) = -\frac{1}{2} \sum_{j=1}^N \frac{(Y_{t_j} - \hat{Y}_{t_j})^2}{v_{j-1}} - \frac{1}{2} \sum_{j=1}^N \log(v_{j-1}) - \frac{N}{2} \log(2\pi) \quad (3.55)$$

donde $\hat{Y}_{t_j} = E(Y_{t_j} | Y_{t_i})$ con $i \leq j-1$, $\tilde{Y}_j = Y_{t_j} - \hat{Y}_{t_j}$ y $v_{j-1} = \text{Var}(\tilde{Y}_j | Y_{t-i})$ con $i \leq j-1$.

$$\int_0^t Z(u) dW(u) = \lim_{n \rightarrow \infty, \Delta t \rightarrow 0} \sum_{k=0}^{n-1} Z(k\Delta t) [W((k+1)\Delta t) - W(k\Delta t)]$$

Capítulo 4

Resultados

En este capítulo se relatarán los resultados obtenidos de la estimación para cada modelo, y además, se evaluará el desempeño de cada uno de éstos.

Para probar la eficiencia de la estimación según los distintos métodos, se trabajó con simulaciones del proceso, con la finalidad de tener control de la función regente de los parámetros, y así, poder verificar que ésta estimación sea certera, comprobando que se acerca al valor “real” preestablecido.

Para mostrar de manera mas clara las virtudes y defectos adyacentes que tienen los métodos al momento de estimar, se trabajará de la siguiente manera:

- En primer lugar, se simularán y ajustarán datos para los modelos de series de tiempo autoregresivos (AR), autoregresivos de medias móviles (ARMA) y autoregresivos fraccionalmente integrado de medias móviles (ARFIMA). Aquí los intervalos de tiempo entre dos observaciones t y $t + 1$ son iguales.
- Luego se ajustará y comparará cada modelo con su homologo (CAR para un modelo AR, CARMA para un modelo ARMA y CARFIMA para un modelo ARFIMA) tomando en cuenta los mismos tiempos simulados, esto con el fin de ver las virtudes y defectos que tiene cada modelo al estimar con tiempos regularmente espaciados. Las medidas que se ocuparán para evaluar el desempeño de las estimaciones en cada modelo se definirán más adelante.
- De la misma manera, se repetirá la metodología simulando y ajustando procesos continuos (CAR, CARMA, CARFIMA) pero con tiempos entre observaciones iguales y así poder comparar con su homologo en series de tiempo.
- Posteriormente, se volverá a simular y ajustar los procesos continuos CAR, CARMA y CARFIMA pero ahora con tiempos desigualmente espaciados. Ahora se volverán ajustar los datos con su modelo homologo tomando en cuenta como si los datos estuvieran en tiempos irregularmente espaciados. Esto con el fin de ver si es posible obtener resultados mejores que un proceso continuo.
- Finalmente, se ocupará la misma metodología del punto anterior pero ahora con datos reales.

Es importante mencionar, que debido a la precariedad de los paquetes en los cuales nos encontramos trabajando, en algunos modelos no será posible realizar la totalidad de los puntos mencionados con anterioridad, ya que no se encuentran las funciones necesarias para la implementación de éstas.

Como gran parte de nuestro análisis es la comparación entre modelos, necesitamos tener una medida cuantitativa que nos ayude determinar cual de los modelos contrastados produce estimaciones mas certeras. Para esto, hemos utilizado las siguientes medidas definidas a continuación.

- **Error Cuadrático Medio**

El error cuadrático medio (MSE por sus siglas en ingles) es una forma de evaluar la diferencia entre un estimador y el valor real de la cantidad que se quiere calcular. El MSE mide el promedio del cuadrado del “error”, siendo el error el valor en la que el estimador difiere de la cantidad a ser estimada. Sea \hat{Y}_t un vector de n predicciones e Y_t el vector de los verdaderos valores para un tiempo $t = (t_1, \dots, t_n)$. El error cuadrático medio (MSE) viene dado por:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Tener un error cuadrático medio de cero (0) es ideal pero no es posible en la mayoría de las situaciones. Un MSE de 0 significa que el estimador predice las observaciones con una precisión perfecta, por lo tanto, mientras más cercano sea este valor a cero, mejores serán las predicciones.

- **Error Porcentual Absoluto Medio**

El error porcentual absoluto medio (MAPE) es un indicador del desempeño del pronostico de los datos que mide el tamaño del error (absoluto) en términos porcentuales. El hecho que se estime una magnitud del error porcentual lo hace un indicador frecuentemente utilizado por los encargados de elaborar pronósticos debido a su fácil interpretación. Sea \hat{Y}_t un vector de n predicciones e Y_t el vector de los verdaderos valores para un tiempo $t = (t_1, \dots, t_n)$. El error porcentual absoluto medio (MAPE) viene dado por:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right|$$

Es posible que algunas veces se observe un valor de MAPE muy elevado aunque el modelo parezca ajustarse a los datos adecuadamente. Esto se debe a que los valores de los datos se aproximan a cero. Debido a que MAPE divide el error absoluto entre los datos reales, los valores que se aproximan a 0 pueden aumentar significativamente el MAPE. De la misma forma que el MSE, mientras más cercano sea este valor a cero, mejores serán las predicciones.

4.1. Simulación

Como hemos mencionado con anterioridad, en primera instancia se simularán datos con el fin de ajustar los distintos tipos de modelos y comparar su poder predictivo.

Una vez simulados la base de datos, se procederá a dividir ésta en dos:

- La primera parte de la base estará formada por el 90 % de los datos (primeros valores). Luego, con esta base se ajustará el modelo y además de utilizar las medidas de desempeño MSE como MAPE para ver el poder de estimación.
- La segunda parte de la base estará formada por el 10 % de datos restantes (últimos valores), los cuales se utilizarán para comparar con los datos predichos a futuro por el modelo. También se ocupara las medidas MSE y MAPE para visualizar que tan precisas son las predicciones en cada modelo.

4.1.1. AR y CAR

Como hemos descrito en la sección (3.3) un proceso continuo autoregresivo de orden p , CAR(p), tiene la siguiente estructura:

$$Y^{(p)}(t) + \alpha_1 Y^{(p-1)}(t) + \dots + \alpha_{p-1} Y^{(1)}(t) + \alpha_p Y(t) = \epsilon(t) \quad (4.1)$$

En particular, si $p = 1$ (sección 3.3.1) nos queda el modelo

$$dY(t) + \alpha Y(t) = \sigma dW(t) \quad (4.2)$$

Ahora, por lo descrito en (2.7), un AR(1) tiene la forma:

$$Y_t = \phi Y_{t-1} + \epsilon_t \quad (4.3)$$

Es posible ver una similitud entre ambos modelos en cuanto a su estructura. Ahora por mencionado en (3.3.1) si tomamos tiempos igualmente espaciados, es posible probar la igualdad de que $\phi = e^{-\alpha}$. En efecto, esta igualdad hace referencia a la autocorrelación de orden 1 para cada modelo: ϕ^k para $k = \mathbb{N}$ en un AR(1) y $e^{-\alpha h}$ para $h = \mathbb{R}^+$.

Para reflejar lo anterior, se simularon valores igualmente espaciados que se rigen por un modelo AR(1), posteriormente se ajustó un modelo AR(1) y CAR(1) para comparar esta igualdad teórica. Luego, este proceso se realizo 1000 veces.

En la siguiente tabla, se evidencia parte de las simulaciones hechas:

ϕ Real	Autocorrelación		Diferencia con ϕ	
	AR(1)	CAR(1)	AR(1)	CAR(1)
0.432	0.4205	0.4188	0.0115	0.0132
0.656	0.7109	0.7094	0.0549	0.0534
0.588	0.6238	0.6246	0.0358	0.0366
0.518	0.5426	0.5421	0.0246	0.0241
0.474	0.3780	0.3769	0.0960	0.0971
0.833	0.8441	0.8427	0.0111	0.0097
0.713	0.6710	0.6686	0.0420	0.0444
0.730	0.7128	0.7134	0.0172	0.0166
0.888	0.8527	0.8566	0.0353	0.0314
0.432	0.4272	0.4361	0.0048	0.0041
⋮	⋮	⋮	⋮	⋮
		Media =	0.0096	0.0096
		SD =	0.0017	0.0017

Tabla 4.1: Autocorrelacion con $lag = 1$ para 1000 simulaciones ajustando los modelos AR(1) y CAR(1) con datos igualmente espaciados (ver apéndice (5))

Según la tabla (4.1) es evidente que se cumple la igualdad teórica. Al ver solo las 10 primeras simulaciones es posible darse cuenta que las autocorrelaciones con $lag = 1$ tanto para el AR(1) como para el CAR(1) prácticamente es la misma. Es posible confirmar esta afirmación al observar el promedio de la diferencia de las 1000 simulaciones, la cual es casi un cero numérico. Cabe mencionar que para que esto sea posible, el parámetro estimado ϕ debe ser mayor a cero, de otra manera no es posible representar un modelo CAR(1) para un $\phi < 0$.

4.1.1.1. AR(1) y CAR(1)

Continuando con el análisis, nos centraremos en las simulaciones para los modelos AR(1) y CAR(1). En primera instancia, las simulaciones y modelos ajustados se harán con tiempos igualmente espaciados.

Para simular la base de datos con la que trabajaremos en ambos modelos, ocuparemos la función *arima.sim*.

```
arima.sim(model, n, rand.gen = rnorm, innov = rand.gen(n, ...),
n.start = NA, start.innov = rand.gen(n.start, ...), ...)
```

Argumentos

model: Una lista con los componentes ar y/o ma, con los coeficientes AR y MA respectivamente.

n: Longitud de la serie de salida, antes de la desdiferenciación. Un entero estrictamente positivo.

`rand.gen:` opcional: una función para generar las innovaciones.
`innov:` una serie de innovaciones opcionales. Si no se proporciona, se utiliza `rand.gen`.
`n.start:` duración del período de "burn-it".
`start.innov:` una serie temporal opcional de innovaciones que se utilizarán para el período de burn-it. Si se proporciona, debe haber al menos valores `n.start` (y `n.start` se calcula de forma predeterminada dentro de la función).

Para la simulación de los datos, solo ocuparemos las dos primeras especificaciones. En este caso, se simularán una base de datos que siga una estructura AR(1). Una vez simulados, se ajustará un modelo modelo AR(1) a los datos con la función *arma*.

```
arma(x, order = c(1, 1), lag = NULL, coef = NULL, include.intercept = TRUE,
series = NULL, qr.tol = 1e-07, ...)
```

Argumentos

`x:` un vector numérico o una serie de tiempo.
`order:` un vector entero bidimensional que da las órdenes del modelo a ajustar. el orden[1] corresponde a la parte AR y el orden[2] a la parte MA.
`lag:` una lista con los componentes ar y ma. Cada componente es un vector entero, especificando los intervalos AR y MA que se incluyen en el modelo. Si se dan ambos, orden y retraso, sólo se utiliza la especificación de retraso.
`coef:` Si se da este vector numérico se utiliza como estimación inicial de los coeficientes ARMA.
`include.intercept:` ¿Debería el modelo contener una intercepción?
`serie:` nombre para la serie.
`qr.tol:` el argumento tol para qr cuando se calculan los errores estándar asintóticos de coef.

Teniendo ya los datos simulados y ajustados a un modelo AR(1), es posible ver a través de la gráfica como se comportan los datos estimados.

Gráfico modelo AR(1): datos vs predichos

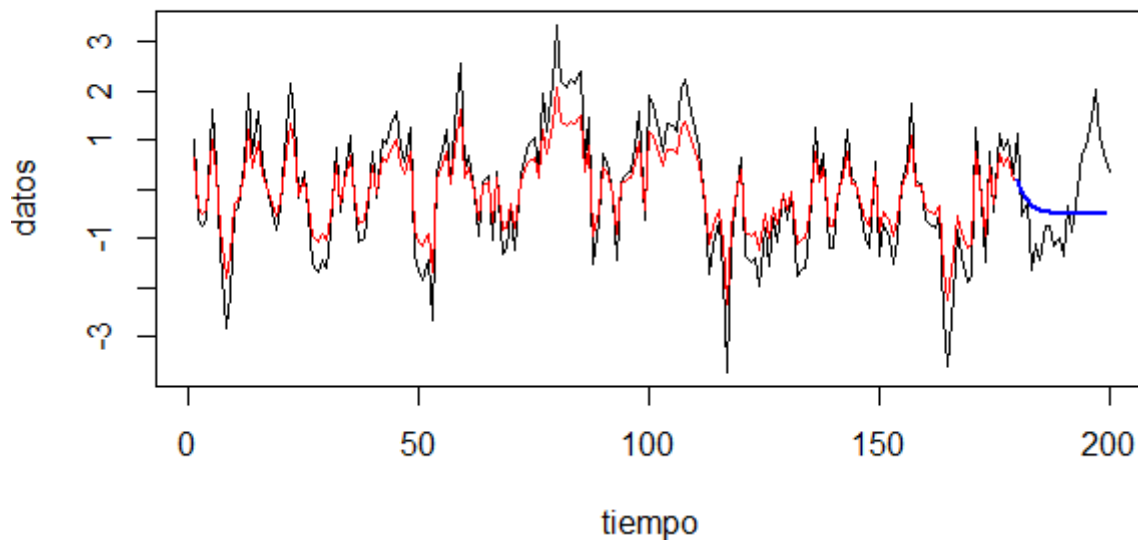


Gráfico 4.1: Gráfico de datos simulados versus datos predichos para un modelo AR(1) con datos regularmente espaciados (ver apéndice (5))

Del gráfico (4.1) se aprecia en color negro los datos simulados, en rojo los datos estimados y en azul las predicciones a futuro por el modelo AR(1).

Ahora, con los mismos datos simulados, se procederá a ajustar un modelo CAR(1) con la función *car* del paquete “cts”.

```
car(x, y=NULL, scale = 1, order = 1, ctrl=car_control())
```

Argumentos

- x: una matriz de datos de dos columnas, siendo la primera columna el tiempo de muestreo y la segunda las observaciones de la primera columna; de lo contrario, x es un vector numérico del tiempo de muestreo.
- y: No se utiliza si x tiene dos columnas; de lo contrario, y es un vector numérico de observaciones en el momento del muestreo x.
- scale: El valor kappa a que se refiere es el parámetro de escalamiento k. Se recomienda elegir el recíproco del tiempo medio entre las observaciones.
- order: Orden de autoregresión.
- ctrl: Parámetros de control utilizados en la predicción y optimización numérica.

Para el modelo CAR, la especificación *scale* es el mismo definido en la sección (3.3.2), el parámetro de escalamiento *k*.

Grafico modelo CAR(1): datos vs predichos

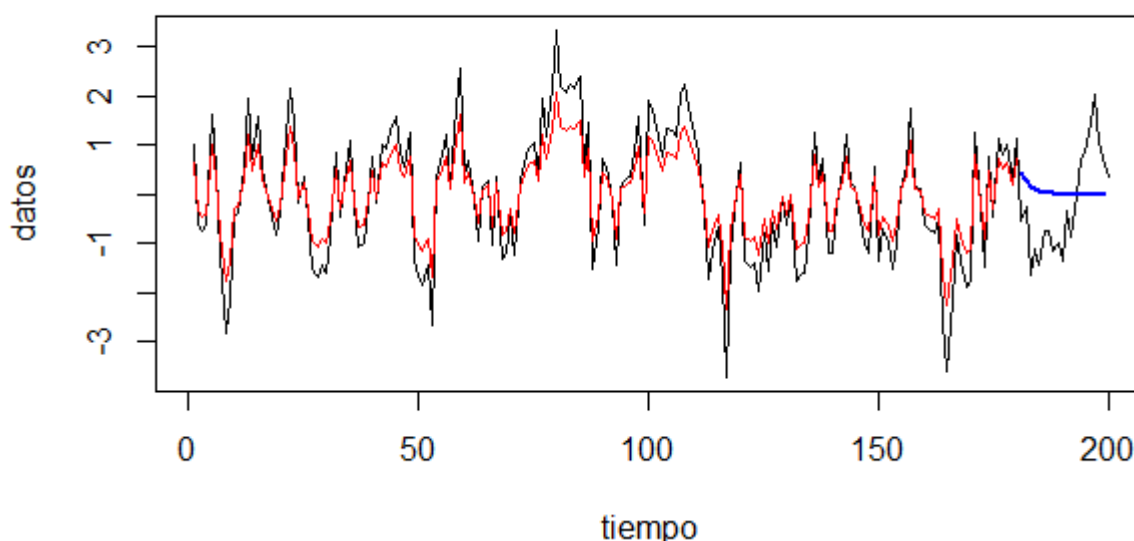


Gráfico 4.2: Gráfico de datos simulados versus datos predichos para un modelo CAR(1) con datos regularmente espaciados (ver apéndice (5))

En el gráfico (4.2), en negro se aprecia los datos simulados, en rojo los datos predichos y en azul las predicciones a futuro por el modelo CAR(1). A simple vista, los gráficos (4.1) y (4.2) son prácticamente iguales, no se ve una diferencia muy marcada en los datos estimados para ambos modelos, por lo que es necesario obtener el MSE y MAPE de cada modelo.

		Modelos	
		AR(1)	CAR(1)
90 % datos	MSE	0.2258266	0.2273369
	MAPE	0.3726526	0.3723088
10 % datos	MSE	1.179227	1.179224
	MAPE	1.040292	1.040352

Tabla 4.2: MSE y MAPE de las estimaciones y predicciones de los datos con los modelos AR(1) y CAR(1) con datos igualmente espaciados

De la tabla (4.2), el cual nos muestra el error cuadrático medio (MSE) y el error porcentual absoluto medio (MAPE) para las estimaciones de los modelos AR(1) y CAR(1), tanto el MSE como el MAPE son casi idénticos. En este caso, se utilizó el 100 % de la base para hacer ajustar el modelo a modo de ejemplo para ver diferencias en cuanto a predicción.

Posteriormente, se realizaron nuevamente 1000 simulaciones y comparamos los MSE y MAPE de cada modelo ajustado en cada simulación.

Como mencionamos al principio de la sección, los datos simulados se dividieron en dos, en donde se obtuvo el MSE y MAPE tanto para los datos ajustados como para las predicciones a futuro.

En las siguientes tablas, muestra el resultado de las simulaciones efectuadas.

Modelo: 90 % datos				
Simulación	AR(1)		CAR(1)	
	MSE	MAPE	MSE	MAPE
1	0.20715966	0.4029792	0.21203393	0.4043638
2	0.27944663	1.2639969	0.28010519	1.3102656
3	0.15130732	1.5280167	0.15092190	1.8518453
4	0.24946717	0.6457565	0.24891911	0.5744090
5	0.60015579	1.1500780	0.60229395	1.1514969
6	0.71802335	1.3728255	0.71883204	1.4151087
7	0.30421871	0.7093371	0.29932074	0.8002965
8	0.18816997	0.5144672	0.19211940	0.5266337
9	0.09897637	0.5211417	0.09703543	0.4653978
10	0.16261654	0.3841809	0.13162842	0.4058917
⋮	⋮	⋮	⋮	⋮

Tabla 4.3: MSE y MAPE para datos estimados con los modelos AR(1) y CAR(1) con el 90 % de los primeros datos

Modelo: 10 % datos				
Simulación	AR(1)		CAR(1)	
	MSE	MAPE	MSE	MAPE
1	1.3693378	1.2502524	1.3693388	1.2502512
2	0.9265832	0.9921615	0.9265788	0.9921729
3	1.0153053	1.4067387	1.0153218	1.4066176
4	0.9859558	0.9941145	0.9859593	0.9941234
5	1.1265246	1.0217936	1.1265243	1.0217939
6	1.1048712	0.9454334	1.1048711	0.9454334
7	1.3887539	1.2037081	1.3887523	1.2036992
8	1.2505967	2.1525861	1.2505809	2.1525198
9	9.1817065	2.3193861	9.1824044	2.3195998
10	2.8361495	0.8275280	2.8369222	0.8271864
⋮	⋮	⋮	⋮	⋮

Tabla 4.4: MSE y MAPE para las predicciones de los datos con los modelos AR(1) y CAR(1) con el 10 % de datos restantes (últimos datos)

		Modelos			
		AR(1)		CAR(1)	
		MSE	MAPE	MSE	MAPE
90 % datos	Media	0.3244676	0.724202	0.3241186	0.7074179
	SD	0.0063630	0.0139995	0.0063439	0.0131111
10 % datos	Media	1.811444	1.737767	1.81145	1.737757
	SD	0.0450019	0.0821531	0.0450032	0.0821543

Tabla 4.5: MSE y MAPE de los modelos AR(1) y CAR(1) tanto para la base de entrenamiento como de predicción de las 1000 simulaciones, con tiempos regularmente espaciados (ver apéndice (5))

Las tablas (4.3) y (4.4) nos muestran el resultado de cada simulación, en este caso, solo las 10 primeras simulaciones con sus respectivos MSE y MAPE. La tabla (4.5) muestra la media y desviación estándar de las 1000 simulaciones del MSE y MAPE de los modelos AR(1) y CAR(1). De aquí es fácil ver que ambos modelos producen prácticamente los mismos resultados tanto al estimar como pronosticar, arrojando diferencia casi nulas. Aquí cabe mencionar que los valores del MAPE tanto para los valores estimados como para las predicciones son muy elevados. Esto se explica ya que el orden de los datos simulados son muy cercanos a cero, por lo que el MAPE puede verse afectado, ya que esté queda dividido por el valor real, el cual es próximo a cero y desvirtúa un tanto esta medida.

Como mencionamos al principio de la sección, los resultados obtenidos hasta aquí solo han sido con datos donde el tiempo entre observaciones son iguales, obteniendo resultados muy parejos si comparamos ambos modelos.

Dado lo anterior, se procederá a efectuar el mismo análisis, pero simulando datos con tiempos irregulares.

En nuestro caso, se simularon 1000 observaciones, de las cuales, se extrajeron solo 200 de ellas de manera aleatoria, quedando así, observaciones con tiempos desigualmente espaciados.

Como el paquete “cts” no posee ninguna función que nos permita simular datos con tiempos irregularmente espaciados, debemos encontrar alguna forma de generar estos valores. Para lograr esto, se simularan datos con la función *arima.sim* y luego se extraerá una muestra aleatoria de estos valores, conservando el tiempo en que fueron simulado. De esta forma, obtendremos valores con tiempos no regulares.

Una vez hecha la simulación y extracción de los datos con un muestreo aleatorio, se procede a ajustar un modelo CAR(1) para estos valores.

Grafico modelo CAR(1): datos vs predichos

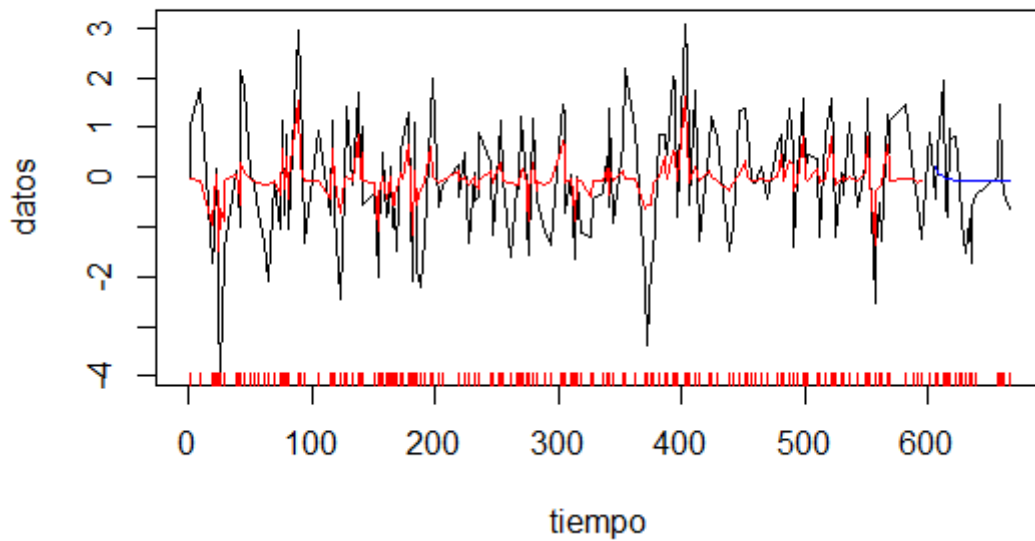


Gráfico 4.3: Gráfico de datos simulados versus datos predichos para un modelo CAR(1) irregularmente espaciado (ver apéndice (5))

De la figura (4.3) se presenta en negro los datos simulados y en rojo las estimaciones del modelo y en azul las predicciones del modelo. Las líneas roja presentes en el eje de las abscisas, representa el tiempo de cada dato. Observando los datos estimados, vemos que no se ajustan de la misma manera como en las simulaciones anteriores. De la misma forma, se ajusta el modelo AR(1).

Grafico modelo AR(1): datos vs predichos

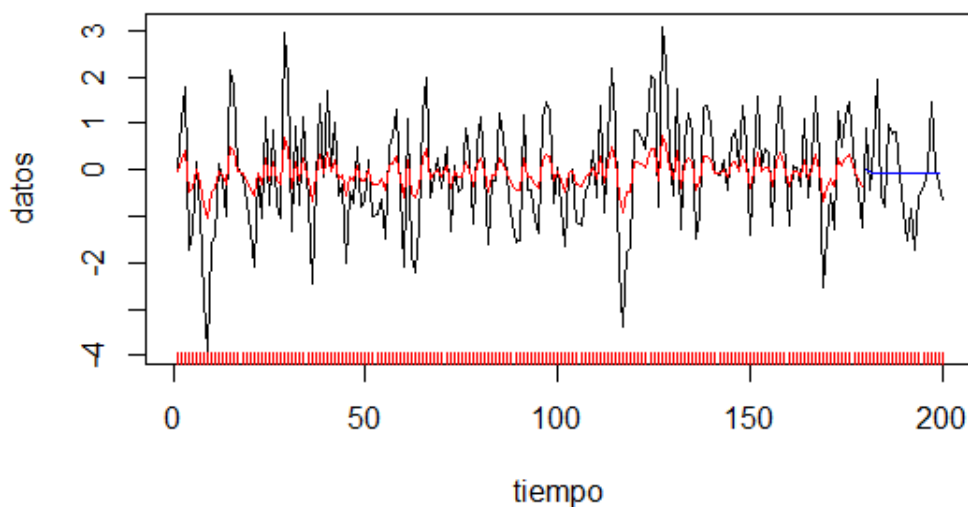


Gráfico 4.4: Gráfico de datos simulados versus datos predichos para un modelo AR(1) (ver apéndice (5))

La representaciones en colores de la figura (4.4) es la misma que en la figura (4.3). Es posible darse cuenta que las estimaciones de este modelo no son tan precisas si se mira los datos simulados con los estimados.

Para reflejar mejor que modelo produce mejores estimaciones, se calcularon los MSE y MAPE tanto para los datos ajustados como para las predicciones en ambos modelos.

		Modelos	
		AR(1)	CAR(1)
90 % datos	MSE	0.8254459	0.8537639
	MAPE	0.8574913	0.8506609
10 % datos	MSE	0.8484394	0.8422878
	MAPE	1.321558	1.275971

Tabla 4.6: MSE y MAPE para datos estimados con los modelos AR(1) y CAR(1) irregularmente espaciado

De la tabla (4.6) no es posible llegar a un consenso de de que modelo se ajusta mejor a los datos, ya que no existen una diferencia muy marcada.

Con el fin de ver que modelo estima de manera más eficiente, se realizaron 1000 simulaciones siguiendo el mismo procedimiento y calculando en cada caso, el MSE y MAPE de cada modelo.

Simulación	Modelo: 90 % datos			
	AR(1)		CAR(1)	
	MSE	MAPE	MSE	MAPE
1	0.8050111	0.8746112	0.8897915	0.9208631
2	0.9154365	1.0886716	0.8940430	1.0214196
3	0.6617218	0.4429579	0.8379437	0.4443871
4	0.9624216	0.9301629	0.9551285	0.8704734
5	0.7078004	0.8327752	0.7586205	0.8918692
6	0.6731435	0.9203958	0.8103497	0.8670060
7	0.7445618	0.7714969	0.7612737	0.7731560
8	0.5490150	0.8859363	0.5686324	0.8663391
9	0.9043814	0.9366197	0.9483024	0.9575334
10	0.4359390	0.5455519	0.5771768	0.5446796
⋮	⋮	⋮	⋮	⋮

Tabla 4.7: MSE y MAPE para datos estimados con los modelos AR(1) y CAR(1) irregularmente espaciado, con el 90 % de los primeros datos

Modelo: 10 % datos				
Simulación	AR(1)		CAR(1)	
	MSE	MAPE	MSE	MAPE
1	1.6277187	1.1100338	1.6217602	1.1007490
2	1.0073489	1.0230891	1.0064107	1.0208869
3	0.9934286	1.0620526	0.9985776	1.0545913
4	1.7537341	2.4878439	1.7529864	2.4149333
5	1.6122532	1.0546058	1.5907113	1.0302007
6	0.6536475	1.1446914	0.6669533	1.0495973
7	2.1105304	0.9887584	2.1063020	0.9950524
8	1.6803157	1.1800751	1.7252894	1.1502722
9	0.7962697	0.9807850	0.8006529	0.9417345
10	1.7962672	1.0786239	1.7200444	1.0877490
⋮	⋮	⋮	⋮	⋮

Tabla 4.8: MSE y MAPE para datos estimados con los modelos AR(1) y CAR(1) irregularmente espaciado, con el 10 % de los últimos datos

Modelos					
		AR(1)		CAR(1)	
		MSE	MAPE	MSE	MAPE
90 % datos	Media	0.7979159	1.017245	0.8479254	0.9789241
	SD	0.0052148	0.0145806	0.0038495	0.0131472
10 % datos	Media	1.607922	1.271132	1.618261	1.348505
	SD	0.0249713	0.0310699	0.0261120	0.0416009

Tabla 4.9: MSE y MAPE de los modelos AR(1) y CAR(1) tanto para la base de entrenamiento como de predicción de las 1000 simulaciones, con tiempos desigualmente espaciados (ver apéndice (5))

La tabla (4.7) y (4.8) nos muestra los resultados de de los MSE y MAPE para cada simulación (10 primeras simulaciones) con datos simulados y tiempos desigualmente espaciados. La tabla (4.9) nos muestra las medidas de desempeño de cada modelo tanto para la base de entrenamiento como para la de predicción. Comparando los resultados obtenido en ambos modelos con las tablas se logra ver que el modelo AR(1) es sutilmente mejor que el modelo CAR(1), ya que el MSE es levemente menor. En este caso, como los valores de la simulación son cercanos a cero, el valor del MAPE se ve afectado. Dado esto, el MSE nos entrega una mejor manera de decidir que modelo es más efectivo al momento de estimar y pronosticar.

4.1.1.2. AR(2) y CAR(2)

Al simular datos con una estructura autoregresiva de orden 1, los resultados obtenidos eran prácticamente iguales al comparar ambos modelos (AR(1) y CAR(1)), por lo que se decidió simular datos con tiempos regulares donde sigan una estructura autoregresiva de orden 2.

Ocupando las mismas funciones descritas en (4.1.1.1), se procederá a simular y ajustar los modelos AR(2) y CAR(2).

Grafico modelo AR(2): datos vs predichos

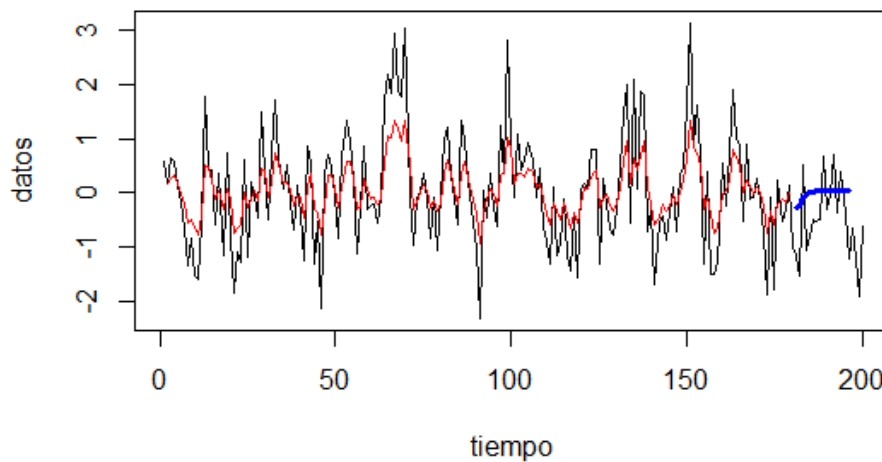


Gráfico 4.5: Gráfico de datos simulados versus datos predichos para un modelo AR(2) (ver apéndice (5))

Grafico modelo CAR(2): datos vs predichos

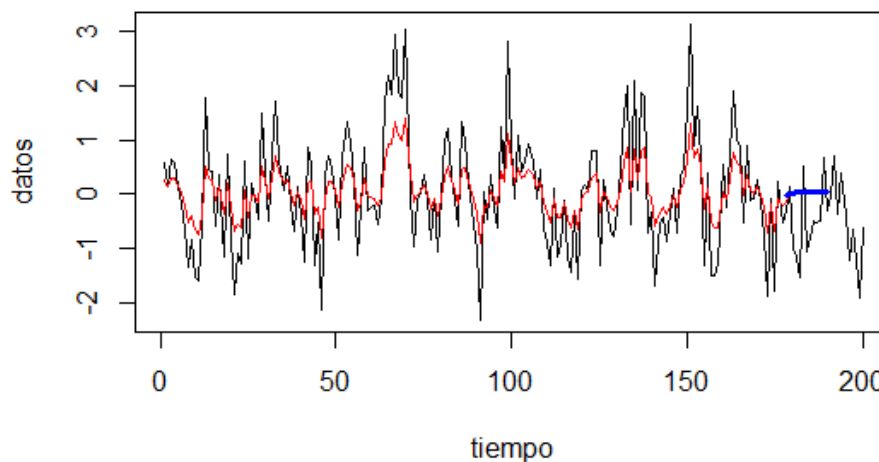


Gráfico 4.6: Gráfico de datos simulados versus datos predichos para un modelo CAR(2) (ver apéndice (5))

En los gráficos (4.5) y (4.6) nos muestran en negro los datos simulados, en rojo los datos predichos por el modelos AR(2) y CAR(2) respectivamente, y en azul las predicciones a futuros comparadas con los datos reales para los modelos AR(2) y CAR(2) respectivamente. Nuevamente, se observa que las estimaciones en ambos modelos son bastante semejantes, por lo que a simple vista no se podría decidir que modelo se ajusta mejor a los datos.

Para poder cuantificar la comparación, nuevamente obtendremos el MSE y MAPE tanto para el 90 % de los primeros datos y del 10 % restante.

		Modelos	
		AR(2)	CAR(2)
90 % datos	MSE	0.4030033	0.3870124
	MAPE	0.8660288	0.7734696
10 % datos	MSE	0.6544153	0.6538447
	MAPE	1.048118	1.041885

Tabla 4.10: MSE y MAPE de las estimaciones y predicciones de los datos con los modelos AR(2) y CAR(2)

La tabla (4.10) nos muestra el MSE y MAPE de las estimaciones y predicciones de los modelos AR(2) y CAR(2) tanto para el 90 % y 10 % de los datos simulados. A diferencia de los modelos de orden $p = 1$, es posible ver leves diferencias en cuanto a los MSE y MAPE con los datos que se ajustaron a ambos modelos, dando valores más pequeños para el modelo CAR(2). En cuanto a las predicciones, en ambos modelos los resultados obtenidos del modelo AR(2) y CAR(2) fueron prácticamente iguales.

Cabe mencionar que los observaciones ajustadas, esta igualmente espaciadas, aún así, el modelo continuo CAR nos proporcionar estimaciones más cercanas por así decirlo, tomando en cuenta el MSE y MAPE.

Para ver si efectivamente el modelo CAR(2) es mejor que el modelo AR(2) se implementan 1000 simulaciones con el fin de obtener los valores de MSE y MAPE para cada uno de ellos. Los resultados son lo que se ven a continuación en las siguientes 3 tablas.

Simulación	Modelo: 90 % datos			
	AR(2)		CAR(2)	
	MSE	MAPE	MSE	MAPE
1	0.8513905	6.2968712	0.5966041	2.6515449
2	1.2384919	2.9347165	0.7196154	2.1103723
3	0.3911785	1.4401793	0.3291385	1.3125445
4	0.2811091	0.5069512	0.2876088	0.5658280
5	0.4213097	0.8124444	0.4230213	0.7981784
6	0.5586894	1.2547085	0.5516864	1.3210624
7	0.2813026	0.9239278	0.2537772	0.7188579
8	0.6473146	5.5194458	0.5585408	2.2569864
9	0.5731389	1.0748418	0.5359272	0.9166109
10	0.5639875	2.8844589	0.5384961	2.7733797
⋮	⋮	⋮	⋮	⋮

Tabla 4.11: MSE y MAPE para datos estimados con los modelos AR(2) y CAR(2) con el 90 % de los primeros datos

Simulación	Modelo: 10 % datos			
	AR(2)		CAR(2)	
	MSE	MAPE	MSE	MAPE
1	2.1882886	0.9952117	2.2101660	1.0066845
2	1.0191711	1.5713107	1.0287301	1.3186690
3	4.9084020	1.2750800	5.1531297	1.4429121
4	2.0536956	1.0119818	2.0513025	1.0179687
5	0.9387033	1.1310487	0.9300635	1.0769214
6	0.5696265	1.6142135	0.5699725	1.6153919
7	2.3343949	1.3639969	2.2776269	1.3410689
8	1.1632846	1.0265555	1.1585936	1.0271528
9	1.3121814	0.9671662	1.3087306	0.9667002
10	1.0795537	0.8638313	1.0329865	0.8280629
⋮	⋮	⋮	⋮	⋮

Tabla 4.12: MSE y MAPE para las predicciones de los datos con los modelos AR(2) y CAR(2) con el 10 % de datos restantes (últimos datos)

		Modelos			
		AR(2)		CAR(2)	
		MSE	MAPE	MSE	MAPE
90 % datos	Media	0.5502614	1.998534	0.4641718	1.412065
	SD	0.0066365	0.0929010	0.0053018	0.0415876
10 % datos	Media	1.545685	1.461124	1.518925	1.483731
	SD	0.0384240	0.0469957	0.0375343	0.0586984

Tabla 4.13: MSE y MAPE de los modelos AR(2) y CAR(2) tanto para la base de entrenamiento como de predicción de las 1000 simulaciones, con tiempos regularmente espaciados (ver apéndice (5))

De las tablas (4.11), (4.12) y (4.13) es posible darse cuenta que el modelo CAR(2) se ajusta de mejor manera que el modelo AR(2), debido a que tanto el MSE como el MAPE es más bajo considerando las estimaciones con el 80 % de los datos. Para las predicciones (20 % de los datos finales) el MSE del modelo CAR(2) es levemente más bajo que el modelo AR(2), pero el MAPE del CAR(2) supera por unos decimales al del modelo AR(2).

4.1.2. ARMA y CARMA

Siguiendo con las simulaciones tanto para modelos discretos y continuos, en esta sección se simularán los modelos ARMA y CARMA.

Al igual que en los modelos anteriores (AR y CAR), existen funciones para la simulación de datos que siguen la estructura de estos procesos.

Para la simulación de valores que siguen un modelo ARMA, la función a utilizar es la misma que en el modelo AR, es decir, ocuparemos la función *arima.sim*. De igual forma, para ajustar el modelo se empleará la misma función que en el modelo AR, es decir, la función *arma*.

Para las simulaciones del proceso CARMA, se necesitará emplear más de una función. Todas las funciones que mencionaremos a continuación provienen del paquete “cts”. En primer lugar, es necesario fijar el orden del modelo a simular, para esto, se ocupa la función *setCarma*.

```
setCarma(p,q,loc.par=NULL,scale.par=NULL,ar.par="a",ma.par="b",
lin.par=NULL,Carma.var="v",Latent.var="x",XinExpr=FALSE, Cogarch=FALSE, ...)
```

Argumentos

- p: un número entero no negativo que indica el número de los coeficientes autorregresivos.
- q: un número entero no negativo que indica el número de los coeficientes de la media móvil.
- loc.par: coeficiente de localización. El valor por defecto loc.par=NULLL implica que $c_0=0$.
- scale.par: coeficiente de escala. El valor por defecto scale.par=NULL implica que $\sigma=1$.
- ar.par: una cadena de caracteres que es la etiqueta de los coeficientes autorregresivos. El valor por defecto es ar.par="a".
- ma.par: una cadena de caracteres que es la etiqueta de los coeficientes de la media móvil. El valor por defecto es ma.par="b".
- Carma.var: una cadena de caracteres que es la etiqueta del proceso observado. Por defecto es "v".
- Latent.var: una cadena de caracteres que es la etiqueta del proceso no observado. Por defecto es "x".
- lin.par: una cadena de caracteres que es la etiqueta de los coeficientes

lineales. Si `lin.par=NULLL`, el valor por defecto, el `'setCarma'` construye el modelo `CARMA(p,q)`.

`XinExpr`: una variable lógica. El valor por defecto `XinExpr=FALSE` implica que la condición de inicio para `Latent.var` es cero. Si `XinExpr=TRUE`, cada componente de `Latent.var` tiene un parámetro como valor inicial.

`Cogarch`: una variable lógica. El valor por defecto `Cogarch=FALSE`.

Luego de definir el orden del modelo, es necesario crear el objeto "muestreo" con la función `setSampling`, la información del muestreo a emplear.

```
setSampling(Initial = 0, Terminal = 1, n = 100, delta, grid, random = FALSE,
sdelta=as.numeric(NULL), sgrid=as.numeric(NULL), interpolation="pt" )
```

Argumentos

`Initial`: Tiempo inicial de la 'grid'.

`Terminal`: Tiempo terminal de la 'grid'.

`n`: número de intervalos de tiempo.

`delta`: el tamaño de la red en el caso de tiempo regular regular.

`grid`: una red de tiempos para la simulación, posiblemente vacía.

`random`: especifique si se trata de un muestreo aleatorio.

`sdelta`: el tamaño de la red en el caso de espacio regular.

`sgrid`: una red en el espacio para la simulación, posiblemente vacía.

`interpolation`: una regla de interpolación en caso de submuestreo. Por defecto, la interpolación de tictac anterior.

Una vez hecho esto, es posible hacer la simulación de valores que siguen un modelo `CARMA` a través de la función `simulate`.

```
simulate(object, nsim=1, seed=NULL, xinit, true.parameter, space.discretized =
FALSE, increment.W = NULL, increment.L = NULL, method = "euler", hurst,
methodfGn = "WoodChan", sampling=sampling, subsampling=subsampling, ...)
```

Argumentos

`object`: un objeto de clase `yuima`, `yuima.model-class` o `yuima.carma-class`.

`xinit`: vector del valor inicial de las variables de estado.

`true.parameter`: lista de parámetros con nombre.

`space.discretized`: para cambiar al método Euler Maruyama, discreto desde el punto de vista espacial.

increment.W:	para especificar el incremento de Wiener para cada tic de tiempo por adelantado.
increment.L:	para especificar el incremento de gravamen para cada tic de tiempo por adelantado.
method:	cadena Variable para el esquema de simulación. El valor por defecto method=euler utiliza la discretización de euler para la simulación de una ruta de muestra.
nsim:	Aún no se ha usado. Incluido sólo para que coincida con el genenirc estándar en las estadísticas del paquete.
seed:	Aún no se ha usado. Incluido sólo para que coincida con el genenirc estándar en las estadísticas del paquete.
Hurst:	valor del parámetro Hurst para la simulación del fGn. Anula la ranura especificada de Hurst.
methodfGn:	métodos de simulación de ruido Gaussiano fraccionado.
sampling:	Un objeto de clase yuima.
subsampling:	Un objeto de clase yuima.

Una vez creado los datos, es posible ajustar el modelo CARMA a través de la función *qmle*.

```
qmle(yuima, start, method = "BFGS", fixed = list(), print = FALSE, lower,
upper, joint = FALSE, Est.Incr = "NoIncr", aggregation = TRUE, threshold = NULL,
rcpp = FALSE, ...)
```

Argumentos

yuima:	un objeto yuima.
print:	puede ver el progreso de la estimación cuando la impresión es VERDADERA.
lower:	una lista con nombre para especificar los límites inferiores de los parámetros
upper:	una lista con nombre para especificar los límites superiores de los parámetros
start:	valores iniciales para ser pasados al optimizador.
fixed:	para la estimación de (cuasi) máxima verosimilitud condicional.
joint:	realizar una estimación conjunta o una estimación en dos etapas.
Est.Incr:	Si el modelo yuima es un objeto de clase yuima.carma o yuima.cogarch, el qmle devuelve un objeto de clase yuima.carma.qmle, cogarch.est.incr-class, cogarch.est-class u objeto de clase mle. Por defecto Est.Incr="NoIncr", los valores alternativos son IncrPar e Incr.
agregation:	Si aggregation=TRUE, antes de la estimación de los parámetros del gravamen se agregan los incrementos.
threshold:	Si el modelo tiene saltos tipo Compound Poisson, el umbral se utiliza para realizar el umbral de los incrementos.

Utilizando estas funciones combinadas, es posible simular y ajustar datos para el proceso CARMA.

Es importante mencionar que si bien el paquete “yuima” posee diversas funciones, no se encuentra disponible ninguna que nos permita hacer predicciones a futuro, por lo que en esta parte, se ajustará la totalidad de los datos al modelo.

Otro punto importante a mencionar, es que dado a lo complicado de crear objetos de clase yuima para datos externos a los simulados por la función *simulate*, solo se simularán datos con ésta función y no con la función *arima.sim*.

4.1.2.1. ARMA(2,1) y CARMA(2,1)

Ocupando las funciones antes mencionadas, se hizo la simulación de valores que siguen una estructura de un proceso CARMA(2,1) con tiempos entre observaciones regulares. En seguida, se ajustó al modelo antes mencionado.

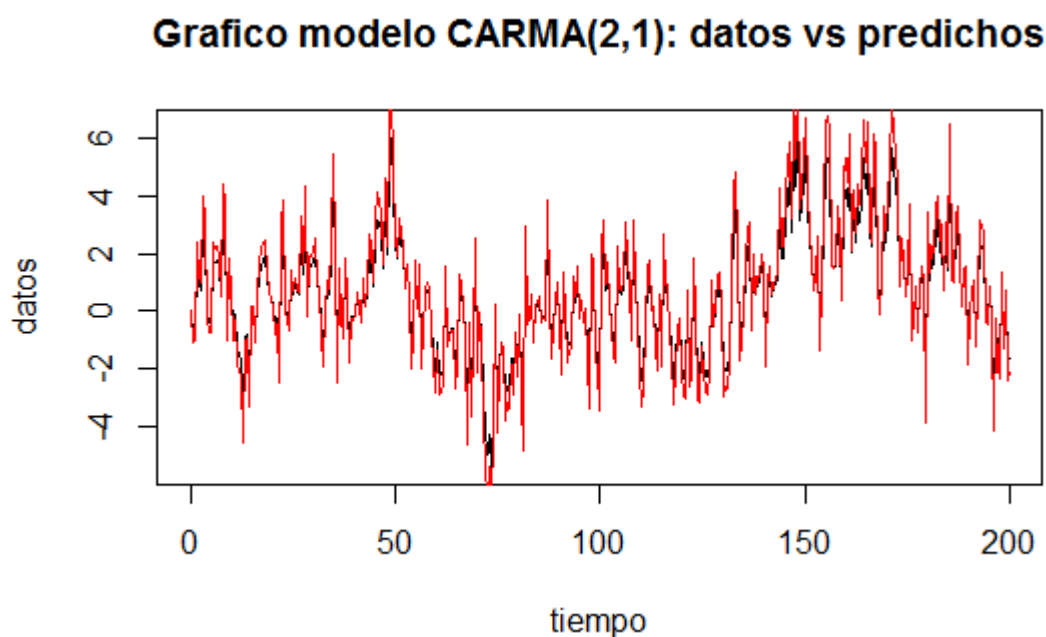


Gráfico 4.7: Gráfico de datos simulados versus datos predichos para un modelo CARMA(2,1) (ver apéndice (5))

En la gráfica (4.7), en color negro se encuentran los datos simulados, mientras que en color rojo los valores estimados por el modelo CARMA(2,1).

Luego, con los mismos datos se ajustó el modelo ARMA(2,1).

Grafico modelo ARMA(2,1): datos vs predichos

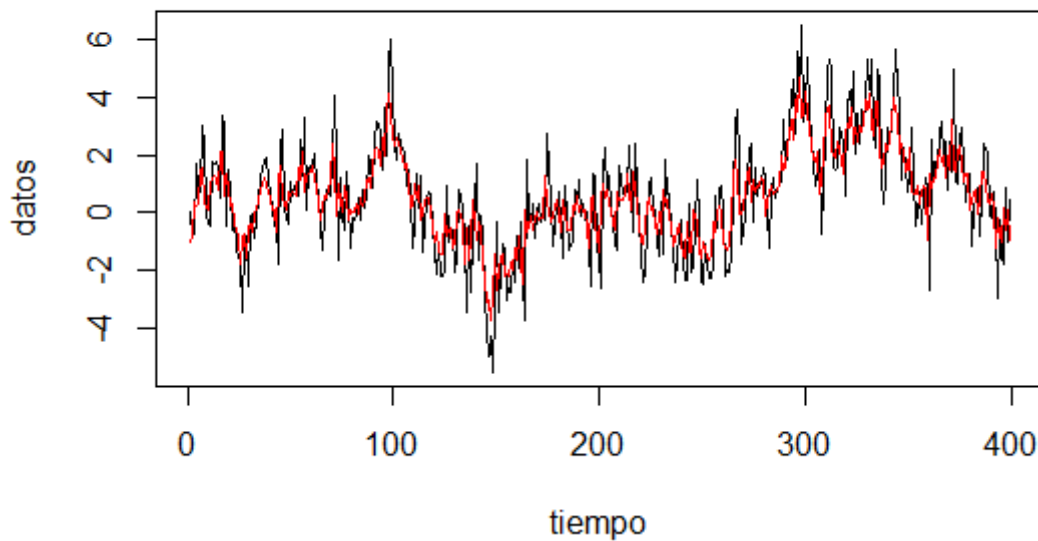


Gráfico 4.8: Gráfico de datos simulados versus datos predichos para un modelo ARMA(2,1) (ver apéndice (5))

De la figura (4.8), la cuál presenta los datos estimados por el modelo ARMA(2,1) siguiendo la misma regla de color que la gráfica (4.7), se aprecia una leve diferencia en los datos estimados en ambos modelos.

	Modelos	
	ARMA(2,1)	CARMA(2,1)
MSE	0.5435512	0.3441701
MAPE	1.464033	0.7000714

Tabla 4.14: MSE y MAPE para datos estimados con los modelos ARMA(2,1) y CARMA(2,1) (ver apéndice (5) y (5))

De la tabla (4.14), la cual nos muestra el MSE y MAPE calculado para cada modelo, claramente las estimaciones producidas son mucho mejores para el modelo CARMA, teniendo diferencias mucho más marcadas que en los modelos AR y CAR de la sección (4.1.1), dando a entender que en este caso se producen resultados más ajustados en los modelos CARMA.

4.1.2.2. ARMA(3,1) y CARMA(3,1)

De la misma manera, se simularon los datos que siguen la estructura CARMA(3,1) donde los tiempos son igualmente espaciados. Posteriormente, se ajustaron los datos tanto al modelo CARMA(3,1) y ARMA(3,1).

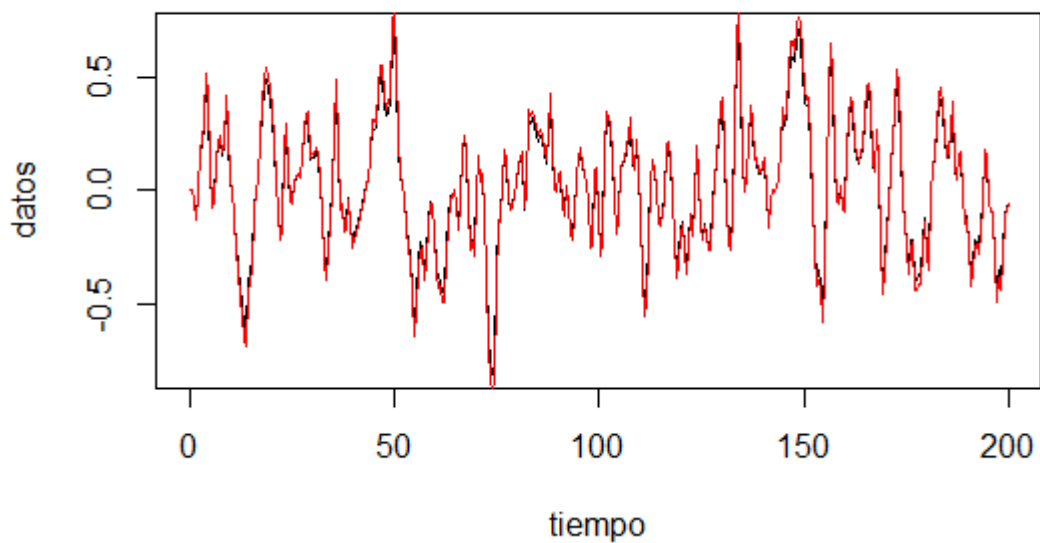
Gráfico modelo CARMA(3,1): datos vs predichos

Gráfico 4.9: Gráfico de datos simulados versus datos predichos para un modelo CARMA(3,1) (ver apéndice (5))

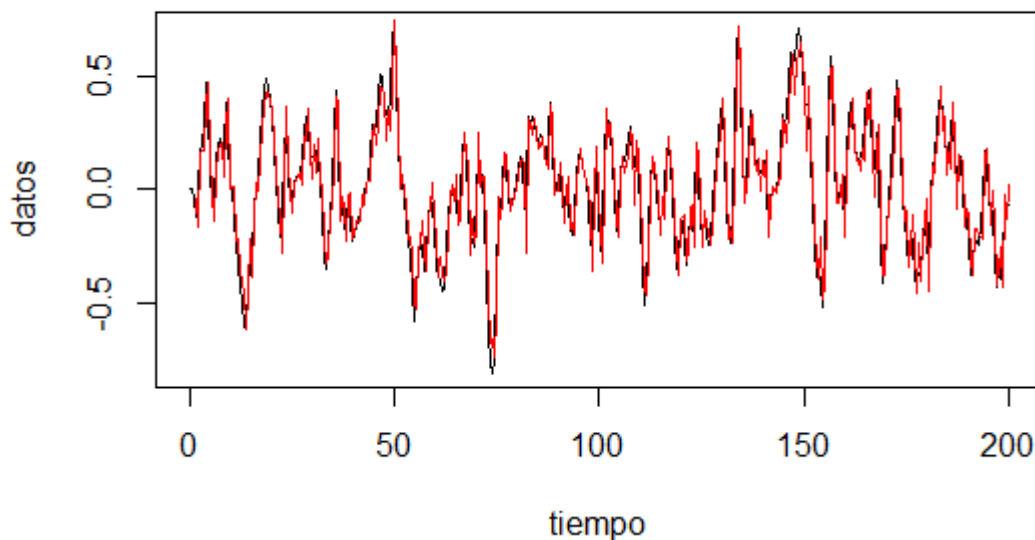
Gráfico modelo ARMA(3,1): datos vs predichos

Gráfico 4.10: Gráfico de datos simulados versus datos predichos para un modelo ARMA(3,1) (ver apéndice (5))

Tanto en las gráficas (4.9) y (4.10) el color negro representa los datos simulados y en color rojo los datos estimados. Al igual que en la sección (4.1.2.1) es posible deducir por las figuras que el modelo CARMA(3,1) produce estimaciones más precisas que el modelo ARMA(3,1).

	Modelos	
	ARMA(3,1)	CARMA(3,1)
MSE	0.009800531	0.0009797614
MAPE	1.237141	0.2822685

Tabla 4.15: MSE y MAPE para datos estimados con los modelos ARMA(3,1) y CARMA(3,1) (ver apéndice (5) y (5))

Observando la tabla (4.15), la cual nos presenta los valores MSE y MAPE calculados en ambos modelos, podemos confirmar que efectivamente el modelo CARMA(3,1) produce mejores estimaciones que los modelos ARMA(3,1).

4.1.3. ARFIMA y CARFIMA

Para realizar el análisis de los modelos correccionalmente integrados, bastará con solo una simulación tanto regular como desigualmente espaciada. Es aquí donde nos centraremos en comparar los modelos ARFIMA(1,d,0) y CARFIMA(1,H,0).

Para simular la base de datos regularmente espaciada con la que trabajaremos en ambos modelos, ocuparemos la función *arfima.sim* de paquete “arfima”.

```
arfima.sim(n, model = list(phi = numeric(0), theta = numeric(0), dint = 0,
dfrac = numeric(0), H = numeric(0), alpha = numeric(0), seasonal = list(phi =
numeric(0), theta = numeric(0), dint = 0, period = numeric(0), dfrac =
numeric(0), H = numeric(0), alpha = numeric(0))), useC = 3, sigma2 = 1,
rand.gen = rnorm, muHat = 0, zinit = NULL, innov = NULL, ...)
```

Argumentos

n: El número de puntos a generar.

model: El modelo a simular. Los argumentos phi y theta deben ser vectores con los valores de los parámetros AR y MA. Tenga en cuenta que la notación Box-Jenkins se utiliza para los parámetros MA. El argumento dint indica cuánta diferenciación debe requerirse para que el proceso sea estacionario. Los argumentos dfrac, H y alpha son valores FDWN, FGN y PLA respectivamente; tenga en cuenta que sólo uno (o ninguno) de ellos puede tener un valor, o se devuelve un error. El argumento estacional es una lista, con los mismos parámetros y un punto, que el argumento modelo. Nótese que con un modelo estacional, podemos tener una mezcla de ruido FDWN/FGN/HD: una en la parte no estacional y la otra en la parte estacional.

useC: Cuánto código C interconectado utilizar: un número entero entre 0 y 3. Se recomienda encarecidamente el valor 3.

sigma2: La variación deseada para las innovaciones de la serie.

rand.gen: La distribución de las innovaciones.

muHat: La media teórica de la serie antes de la integración (si se

realiza la integración de números enteros)

zinit: Usado para la predicción; no está destinado a ser usado directamente. Esto permite especificar el inicio de una serie temporal antes de aplicar la diferenciación inversa (integración).

innov: Usado para la predicción; no está destinado a ser usado directamente. Esto permite el uso de innovaciones dadas en lugar de las proporcionadas por `rand.gen`.

Una vez realizada la simulación de datos igualmente espaciados, se procede a ajustar dichos datos a un modelo arfima. Para esto ocuparemos la función *arfima* del paquete “arfima”.

```
arfima(z, order = c(0, 0, 0), numeach = c(1, 1), dmean = TRUE,
itmean = FALSE, fixed = list(phi = NA, theta = NA, frac = NA, seasonal =
list(phi = NA, theta = NA, frac = NA), reg = NA),
lmodel = c("d", "g", "h", "n"), ...)
```

Argumentos

z: El conjunto de datos (serie temporal)

order: El orden del modelo ARIMA a ser ajustado: $c(p, d, q)$. Tenemos que p es el número de parámetros AR (ϕ), d es la cantidad de diferenciación de enteros y q es el número de parámetros MA (θ).

numeach: El número de arranques para encajar para cada parámetro. El primer argumento en el vector es el número de inicios para cada parámetro AR / MA, mientras que el segundo es el número de inicios para el parámetro fraccionario. Cuando se establece en 0, no se ajusta el ruido fraccionario.

Dmean: Si la media debe ajustarse dinámicamente con el optimizador.

itmean: Esta opción está bajo investigación y se establecerá en FALSO automáticamente hasta que se decida qué hacer. Si la media debe ajustarse iterativamente usando la función `TrenchMean`.

fixed: Una lista de parámetros para ser arreglados. Si vamos a arreglar ciertos elementos del proceso AR, por ejemplo, ϕ fijo debe tener una longitud igual a p . Cualquier valor numérico fijará el parámetro en ese valor.

lmodel: El modelo de memoria larga (tipo de ruido) que se utilizará: "d" para FDWN, "g" para FGN, "h" para PLA y "n" para ninguno (es decir, modelos de memoria corta ARMA). El valor predeterminado es "d".

seasonal: Los componentes estacionales del modelo que deseamos ajustar, con los mismos componentes que el anterior. El período debe

ser suministrado.

Y para poder ajustar estos mismos datos al modelo CARFIMA, se utilizará la función *carfima* del paquete “*carfima*”.

```
carfima(Y, time, ar.p, ma.q, method = "mle", bayes.param.ini,
bayes.param.scale, bayes.n.warm, bayes.n.sample)
```

Argumentos

Y: Un vector de longitud k para los datos observados.
time: Un vector de longitud k para los tiempos de observación.
ar.p: Un entero positivo para el orden del modelo AR. ar.p debe ser mayor que ma.q. Si el p.ar. es mayor que 2, pueden producirse errores numéricos.
ma.q: Un número entero no negativo para el orden del modelo MA. ma.q debe ser menor que ar.p.
method: 0 "mle" o "bayes". El método "mle" realiza la inferencia basada en MLE, produciendo MLEs e incertidumbres asintóticas de los parámetros del modelo. El método "bayes" extrae muestras posteriores de los parámetros del modelo.

Para simular la base de datos desigualmente espaciada con la que trabajaremos en ambos modelos, ocuparemos la función *carfima.sim* de paquete “*carfima*”.

```
carfima.sim(parameter, time, ar.p, ma.q)
```

Argumentos

parameter: Un vector de longitud $p + q + 2$ para los valores generativos de los parámetros del modelo.
time: Un vector para los k tiempos de observación, espaciados regular o irregularmente.
ar.p: Un escalar para el orden del modelo AR.
ma.q: Un escalar para el orden del modelo MA.

Es importante aclarar que para ajustar cada modelo, no se dividirá la base de datos, esto debido a que el paquete “*carfima*” no posee ninguna función o método para poder hacer predicciones, por lo que al ajustar los datos, se contemplará la totalidad de los datos.

Cabe mencionar que, en la función carfima, es necesario especificar los tiempos de las observaciones. En este caso, los tiempos desigualmente espaciados los generaremos a partir de una variable exponencial. El código con el cual se simularon estos tiempos (desiguales) están dentro del script de la generación de un CARFIMA en la sección Anexos.

Con los tiempos ya simulados, se procede a simular los datos con intervalos de tiempos desiguales para luego ajustar un proceso CARFIMA(1,H,0).

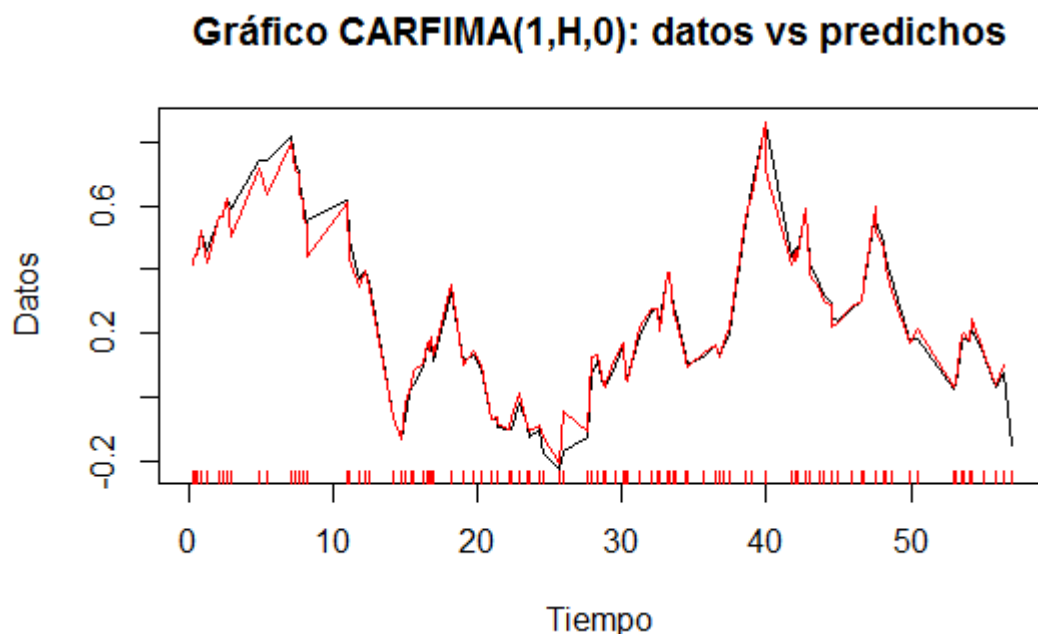


Gráfico 4.11: Gráfico de datos simulados versus datos predichos para un modelo CARFIMA(1,H,0), con $H = 0,7$ (ver apéndice (5))

En el gráfico (4.11), se aprecia en negro los datos simulados y en rojo los datos estimados por el modelo. Las líneas rojas que se aprecian en el eje X son los tiempos generados para cada observación. De este gráfica es posible observar que este modelo se ajusta de buena forma a los datos con solo mirarlo. Para comprobar que efectivamente este modelo produce una buena estimación es necesario ver el valor del MSE y MAPE, los cuales se presentarán más adelante.

En seguida, se ajustara los mismos datos a un modelo ARFIMA tomando los tiempos como si estuvieran igualmente espaciados.

Gráfico ARFIMA(1,d,0): datos vs predichos

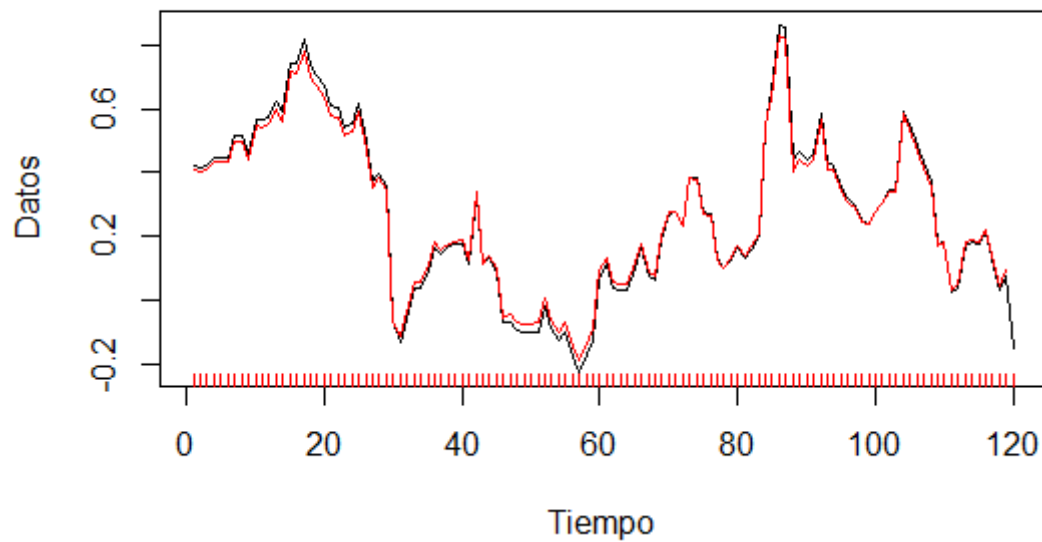


Gráfico 4.12: Gráfico de datos simulados versus datos predichos para un modelo ARFIMA(1,d,0), con $d = 0,1$ (ver apéndice (5))

Si comparamos los gráficos (4.11) y (4.12) es posible apreciar que no tienen la misma forma, esto debido a que el gráfico (4.12) se tomaron los datos como si estuvieran regularmente espaciados.

Para las estimaciones del modelo CARFIMA (gráfico 4.11) se puede observar que en ciertos puntos la estimación se aleja del valor real simulado, no así como en las estimaciones del modelo ARFIMA que a simple vista parecen ser mucho más cercanas al valor real.

Para poder comparar ambos modelos, se calculan el MSE y MAPE de cada modelo.

	Modelos	
	ARFIMA(1,d,0)	CARFIMA(1,H,0)
MSE	0.009929156	0.0009328954
MAPE	0.4753698	0.1143293

Tabla 4.16: MSE y MAPE para datos estimados con los modelos ARFIMA(1,d,0) y CARFIMA(1,H,0)

En la tabla (4.16) es posible deducir todo lo contrario con lo visto en los gráfico de las estimaciones, ya que el modelo CARMFIMA posee mejores medidas que el modelo ARFIMA, tanto para el MSE y MAPE los cuales son bastante más pequeños para el modelo CARFIMA.

Al igual que en los otros modelos vistos con anterioridad, se procederá a hacer 100 simulaciones (debido al tiempo que demora en ajustarse el proceso CARFIMA) para comparar los MSE y MAPE de ambos modelos.

Simulación	Modelos			
	ARFIMA(1,d,0)		CARFIMA(1,H,0)	
	MSE	MAPE	MSE	MAPE
1	0,00923530	1,43536724	0,00096099	0,36916699
2	0,01028523	0,83345075	0,00330535	1,83475459
3	0,00868225	1,33046278	0,00212645	1,21396457
4	0,01077062	2,35217126	0,00171261	0,69057962
5	0,01197520	6,39297669	0,00187012	5,40465603
6	0,00759796	2,37742511	0,00163390	0,82280123
7	0,01203120	0,59977899	0,00269804	0,26424272
8	0,01516959	0,76290806	0,00179917	0,22885821
9	0,00886936	0,66910507	0,00144898	0,23463046
10	0,01174076	7,42066780	0,00268646	1,69500983
⋮	⋮	⋮	⋮	⋮

Tabla 4.17: MSE y MAPE para las estimaciones de los datos con los modelos ARFIMA(1,0.1,0) y CARFIMA(1,0.7,0)

	Modelos			
	AR(1)		CAR(1)	
	MSE	MAPE	MSE	MAPE
Media	0.010558	1.914146	0.001804	0.86098
SD	0.0030942	0.3075572	0.0001083	0.1852035

Tabla 4.18: MSE y MAPE de los modelos ARFIMA(1,d,0) y CARFIMA(1,H,0) de las 1000 simulaciones, con tiempos desigualmente espaciados y parámetro $d = 0,1$ y $H = 0,7$ (ver apéndice (5) y (5))

De la tabla (4.17) y (4.18) nos confirma lo mencionado anteriormente, pues tanto el MSE Y MAPE en promedio es menor cuando se ajusta con los datos con tiempos irregulares.

Con la información recogida en (4.17) es posible concluir que al tomar datos que se encuentran con tiempos irregulares y ajustar estos valores como si fueran igualmente espaciado producen estimaciones que si bien son buenas, no son mejores que el modelo continuo.

4.2. Datos Reales

Todo el análisis anterior se hizo en base a datos simulados, dándonos una noción de como se comportan y el poder de estimación de los modelos tanto discretos como continuos. A continuación, se trabajará con datos reales obtenidos en el paquete `cts` de `R`, el cual cuenta con datos relacionados con la geofísica y la medicina.

La idea es ver qué modelo produce mejores resultados en cuanto a la estimación y predicción de las observaciones. Para esto, se escogerá los de mejores modelos autoregresivos continuos (CAR) y se comparará con el mejor modelo AR. La forma en la cuál se elegirán éstos es en base MSE y MAPE de cada modelo.

Dentro de este análisis queda al margen el modelo CARMA, ya que las funciones del paquete `yuima` de `R`, no nos permite ajustar un modelo en base a datos no simulados.

4.2.1. Aplicación Geofísica

Nuestra primera base de datos surge de una serie de 164 mediciones de abundancia relativa de un isótopo de oxígeno ($\delta^{18}\text{O}$) en un núcleo oceánico. Estas mediciones se realizan a intervalos regulares de 10 cm, pero debido a las diferentes tasas de sedimentación, corresponden a puntos de tiempo espaciados de forma desigual en el pasado, con una separación media de 2.000 años. Las medidas se creen reflejan las condiciones climáticas en el momento de la sedimentación, en particular el almacenamiento de hielo en la tierra (glaciación). En [18] se ha propuesto una escala de tiempo para estos datos sobre la base de la teoría de Milankovitch, que las variaciones en los datos siguen ciclos en la oblicuidad de la órbita de la Tierra (período 41000 años), precesión de los equinoccios (período 21000 años) y la excentricidad de la órbita de la Tierra (período 100000 años). Estos períodos son aproximados.

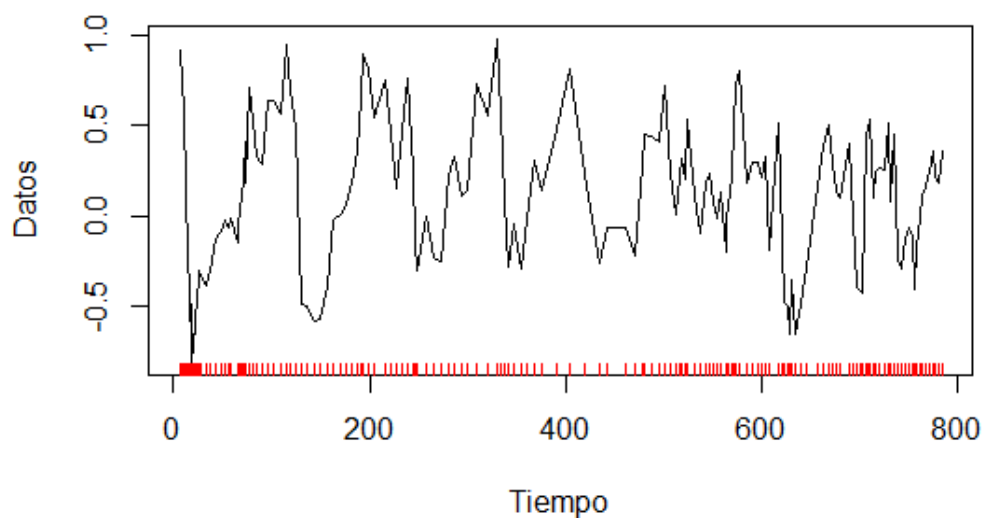


Gráfico 4.13: Gráfico de las mediciones de abundancia relativa de un isótopo de oxígeno $\delta^{18}\text{O}$

En el gráfico 4.13 nos muestra las mediciones de la abundancia relativa de un isótopo de oxígeno $\delta^{18}\text{O}$ en color negro, y en rojo los tiempo de cada observación.

Como mencionamos, hemos elegido los dos mejores modelos CAR que se ajustan a estos datos. Para esto, se ajusto un modelo para valores de $p = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ y se calculo tanto el MSE y MAPE de cada uno; de esto, se llego a que el MSE y MAPE más pequeño fue para $p = 1$ y $p = 3$, es decir, los modelo CAR(1) y CAR(3).

Las estimaciones para cada uno de estos modelos en relación a los datos reales se presentan a continuación.

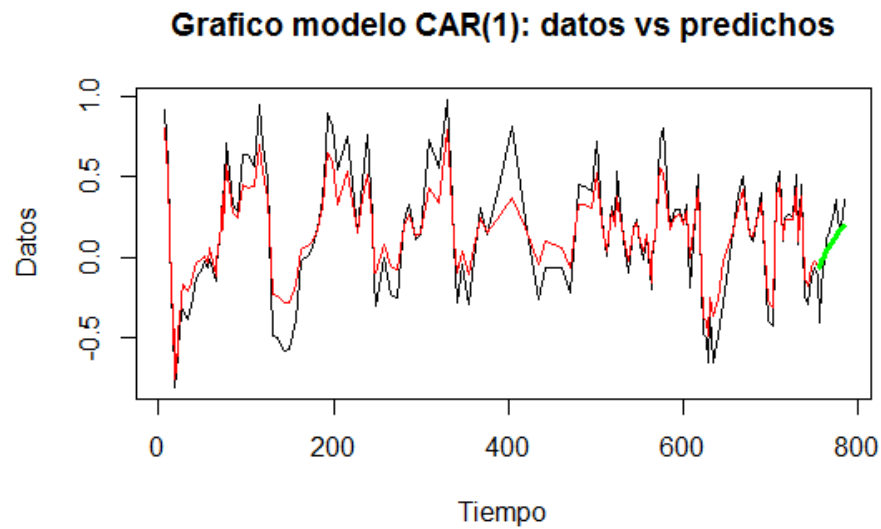


Gráfico 4.14: Gráfico de las mediciones de abundancia relativa de un isótopo de oxígeno versus los predichos para un modelo CAR(1) (ver apéndice (5))

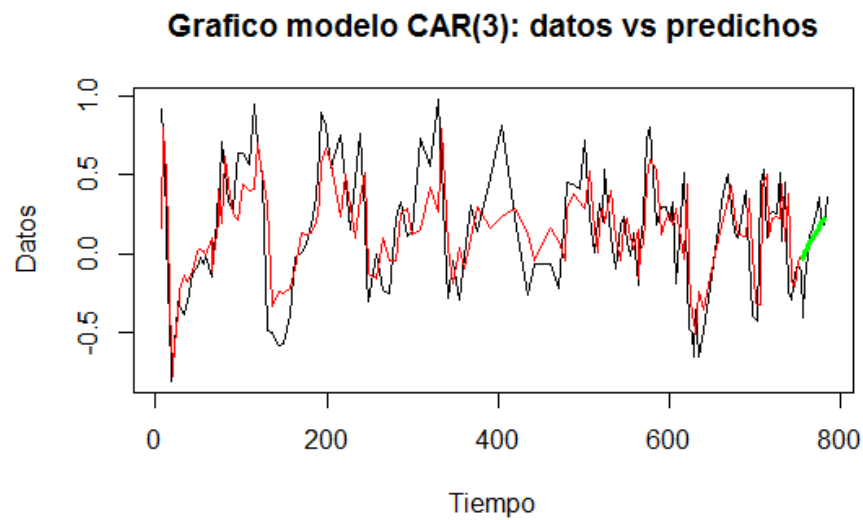


Gráfico 4.15: Gráfico de las mediciones de abundancia relativa de un isótopo de oxígeno versus los predichos para un modelo CAR(3) (ver apéndice (5))

Como se aprecia en el gráfico (4.14) y (4.15), ambos modelos producen estimaciones muy similares a simple vista. El MSE y MAPE de cada modelo será presentado más adelante. Continuando el análisis, se escogerá el mejor modelo autoregresivo (AR) para valores de $p = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$, resultando el menor MSE y MAPE para el valor $p = 1$, es decir, AR(1). Los datos estimados y predichos se ven a continuación.

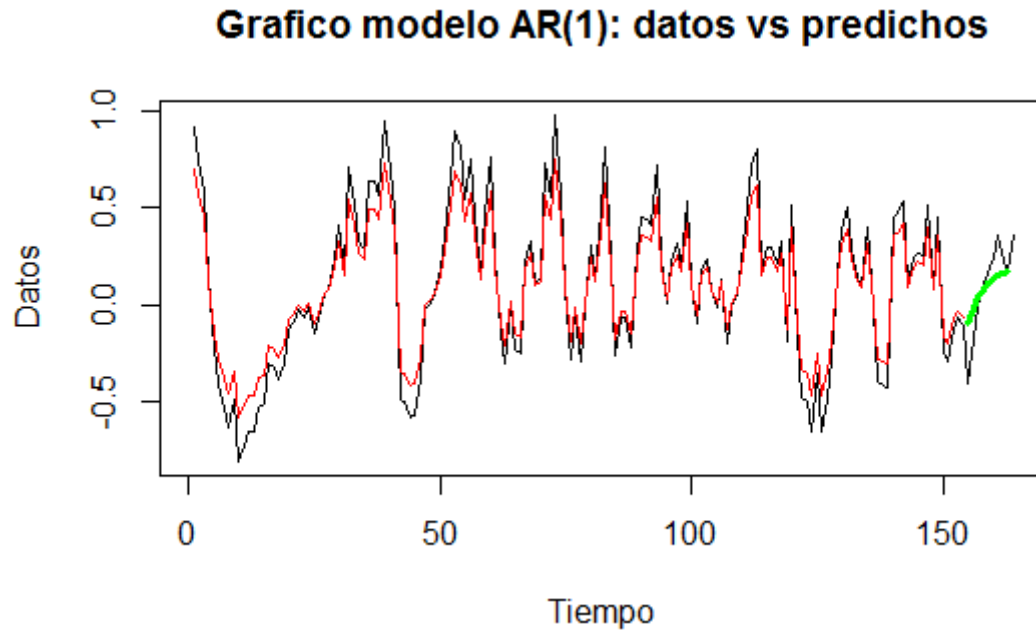


Gráfico 4.16: Gráfico de las mediciones de abundancia relativa de un isótopo de oxígeno versus los predichos para un modelo AR(1) (ver apéndice (5))

Ya teniendo los mejores modelos, se presenta la siguiente tabla con los valores MSE y MAPE de cada modelo ajustado.

		Modelos		
		AR(1)	CAR(1)	CAR(3)
90 % datos	MSE	0.072210	0.01596	0.01899
	MAPE	0.32546	0.53551	0.63358
10 % datos	MSE	589067.85	589355.08	589344.65
	MAPE	0.99999	0.99999	0.99999

Tabla 4.19: MSE y MAPE para datos estimados con los modelos AR(1), CAR(1) y CAR(3), tomando base de entrenamiento y predicción (ver apéndice (5))

Como se observa en la tabla (4.19), considerando la base de entrenamiento, el MSE más bajo se produce para el modelo CAR(1) seguido por el modelo CAR(3). Si solo nos guiamos por el MSE, los modelos continuos producen estimaciones más cercanas que el modelo de tiempo regular. Considerando como medida de ajuste el MAPE, el modelo

AR(1) es el que pose menor MAPE en comparación con los otros 2 modelos (CAR(1) y CAR(3)). Considerando la base para la predicción, los 3 modelos poseen MSE muy parejos, y el MAPE es igual en los 3 modelos, por lo que no es posible concluir con que existe mucha diferencia entre estos modelos.

De lo anterior es posible deducir que el modelo CAR puede ser mejor o peor en cuanto a las estimaciones dependiendo de la medida que se establezca.

4.2.2. Aplicación Médica

Nuestra segunda base de datos es un conjunto de 209 mediciones de la función pulmonar de un paciente asmático. Estos son realizados por los pacientes en su mayoría en intervalos de tiempo de 2 horas, pero con lagunas irregulares en este registro. Estos datos también varían según los modelos de tiempo continuo.

Las observaciones de muestran en el siguiente gráfico.

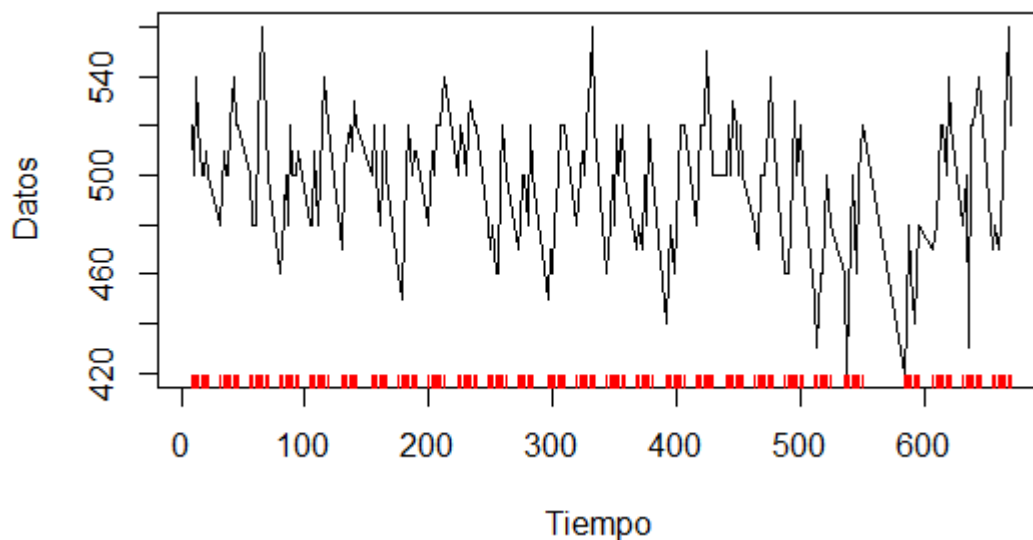


Gráfico 4.17: Gráfico de las mediciones de la función pulmonar de un paciente asmático

En el gráfico (4.17) se observan, en negro, las mediciones de la función pulmonar de un paciente asmático, y en rojo los tiempos en que fueron realizadas estas observaciones.

Nuevamente, escogeremos los 2 mejores modelos de tiempo continuo para valores de $p = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ según cual tenga el MSE y MAPE más cercano a cero. Ajustando cada modelo para los distintos valores de p se llega a que los mejores modelos resultaron ser para $p = 1$ y $p = 2$, en cuanto al MSE y MAPE. Las estimaciones de cada modelo se presentan a continuación en los gráficos (4.18) y (4.19).

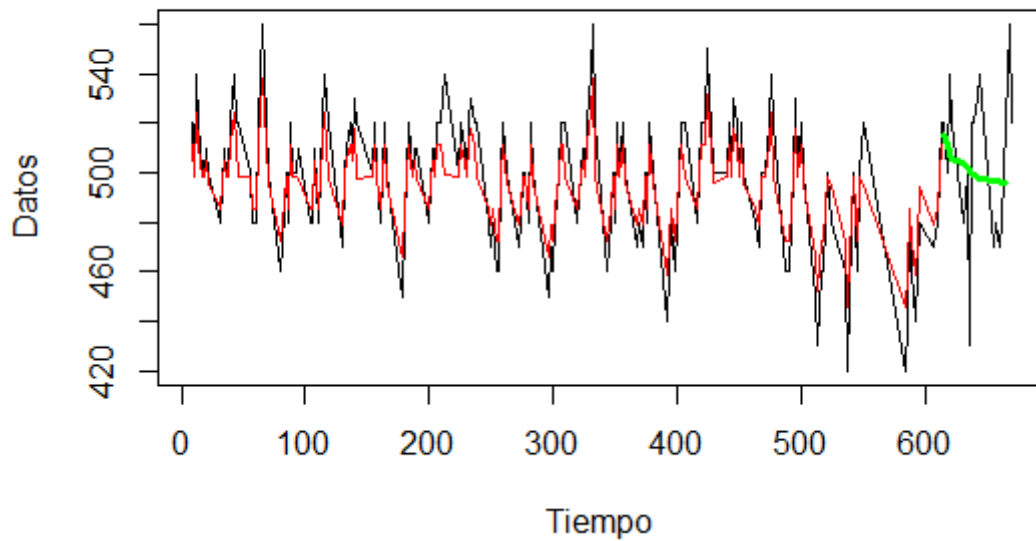
Gráfico modelo CAR(1): datos vs predichos

Gráfico 4.18: Gráfico de mediciones de la función pulmonar de un paciente asmático versus los predichos para un modelo CAR(1) (ver apéndice (5))

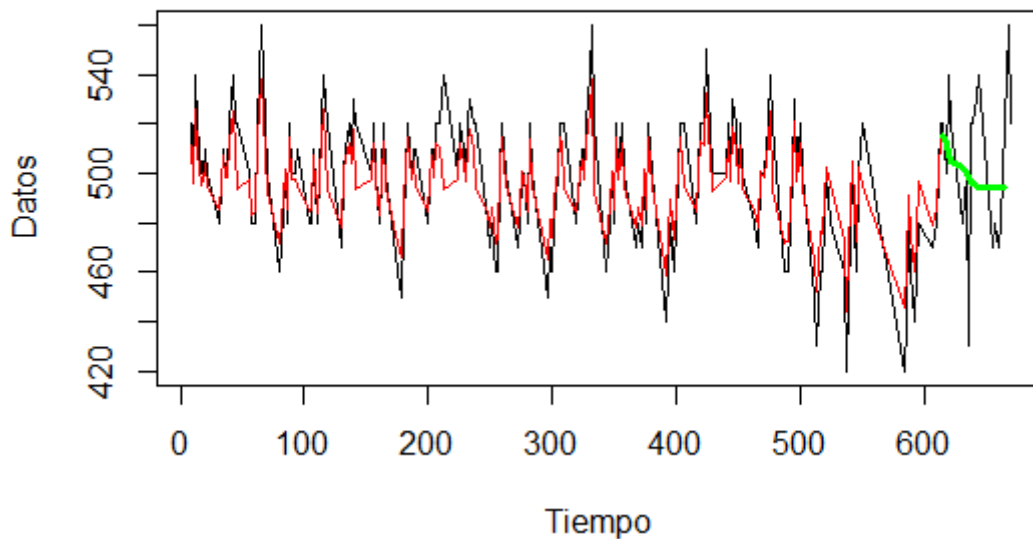
Gráfico modelo CAR(2): datos vs predichos

Gráfico 4.19: Gráfico de mediciones de la función pulmonar de un paciente asmático versus los predichos para un modelo CAR(2) (ver apéndice (5))

De los gráficos es posible observar que ambos modelos producen resultados prácticamente idénticos a simple vista. Para ver que modelo se ajusta mejor es necesario calcular el MSE y MAPE, los cuales se presentarán mas adelante.

Igualmente, se obtendrá el mejor modelo autoregresivo $AR(p)$ para valores de $p = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$. Obteniendo el MSE y MAPE para cada valor de p , se llega a que el mejor modelo, en este caso, es para $p = 1$, es decir, un $AR(1)$. Las estimaciones del modelo se presentan a continuación.

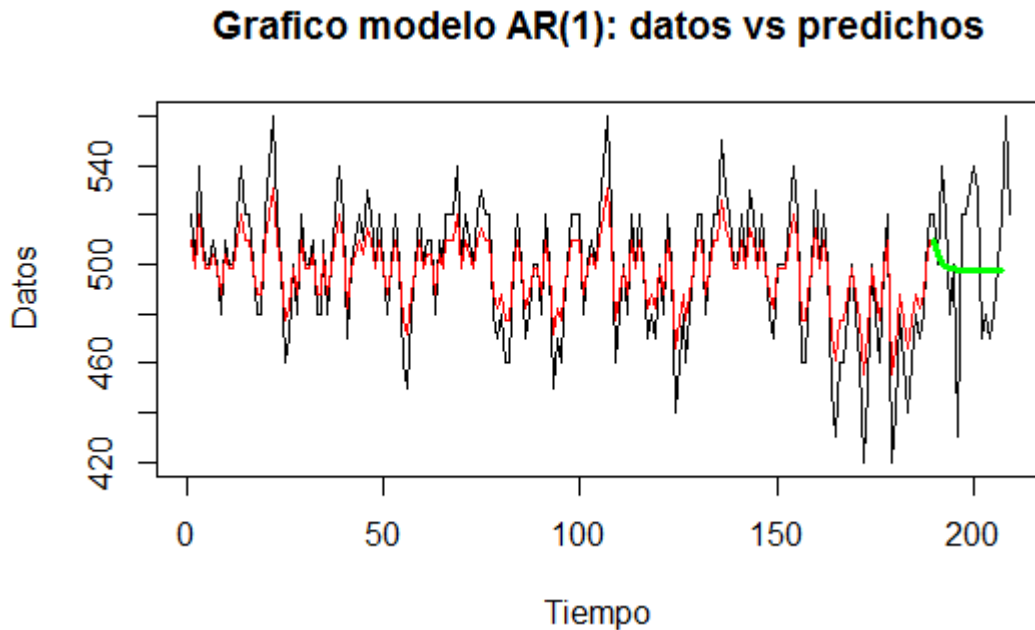


Gráfico 4.20: Gráfico de mediciones de la función pulmonar de un paciente asmático versus los predichos para un modelo $AR(1)$ (ver apéndice (5))

Del gráfico (4.20) no es posible concluir que modelo se ajusta mejor, considerando (4.18) y (4.19). A continuación se presenta el MSE y MAPE de cada modelo para la base de entrenamiento y predicción.

		Modelos		
		AR(1)	CAR(1)	CAR(2)
90 % datos	MSE	476.58	111.10	129.22
	MAPE	0.01932	0.01690	0.01803
10 % datos	MSE	16821.07	15772.18	16305.31
	MAPE	0.20848	0.20848	0.20848

Tabla 4.20: MSE y MAPE para datos estimados con los modelos $AR(1)$, $CAR(1)$ y $CAR(2)$, tomando base de entrenamiento y predicción (ver apéndice (5))

La tabla ((4.20)), considerando la base de entrenamiento (80 % de los datos), claramente, los modelos continuos $CAR(1)$ y $CAR(2)$ se ajustan de mejor manera que el modelo $AR(1)$ tanto para el MSE y MAPE, ya que los valores del MSE y MAPE

son mas pequeños que en el modelo AR(1) dando una clara señal que para datos con tiempos que no son igualmente espaciados es mejor tratarlos con este tipo de modelos (CAR) que con los modelos de series de tiempo clásicos.

De lo anterior es posible concluir que efectivamente los modelos de tiempo continuo producen mejores resultados que los modelos de tiempo discreto.

Capítulo 5

Conclusiones

Existe escasa información sobre los modelos continuos abortados en este trabajo (CAR, CARMA, CARFIMA), pero a pesar de este inconveniente es posible variadas conclusiones de cada uno de estos modelos.

En primer lugar, para el modelo CAR se ocupó el paquete `cts`. Considerando el caso igualmente espaciado para datos simulados, ambos modelos (CAR y AR) producen resultados idénticos tomando como orden $p = 1$, esto debido a que el modelo CAR(1) es posible discretizarlo y llevarlo a la forma AR(1), donde la relación que existe entre los parámetros es precisamente la autocorrelación de orden 1 en ambos modelos ($\phi = e^\alpha$), la cual, se cumplió como es de esperarse. Los resultados cambian cuando aumentamos el orden del modelo, en este caso, $p = 2$ donde el modelo que mejor produce resultados es el proceso CAR. Para el caso desigualmente espaciado, se puede decir que el modelo con mejores resultados es el AR ya que posee MSE y MAPE levemente más bajo que el CAR. Aquí cabe mencionar que se intentó simular datos CAR mediante la simulación de datos AR. Esto es una posible causa de porqué los resultados obtenidos fueron mejores en un AR. Además, podemos observar que el modelo CAR(1) detecta de muy buena manera la autocorrelación de la serie al igual que el modelo AR(1), ya que ambos arrojan autocorrelaciones prácticamente iguales, y muy cercanas a la autocorrelación real.

Para el modelo CARMA, se ocupó el paquete `yuima`, del que no fue posible simular datos con tiempos irregularmente espaciados debido a lo complicado de manipular las funciones de este. En el caso igualmente espaciado los resultados obtenidos por el modelo CARMA fueron ampliamente mejores que en el modelo ARMA, ya que el MSE y MAPE fueron mucho más bajos que el modelo discreto ARMA. A pesar de ser solo dos casos, los resultados fueron muy concluyentes si comparamos el modelo continuo con el discreto. Por esto, es posible deducir que para datos que presentan tiempos irregulares es más beneficioso ajustarlos según un modelo CARMA que un modelo ARMA.

En el modelo CARFIMA, donde se ocupó el paquete `carfima`, no nos fue posible modelar datos con tiempo regularmente espaciados debido a problemas con la función que permite modelar un CARFIMA. Para los datos simulados con tiempos desigualmente espaciados, es posible concluir que con el modelo CARFIMA se obtienen resultados

más precisos que el modelo ARFIMA considerando las medidas MSE y MAPE, ya que estas medidas fueron más cercanas a cero en el modelo CARFIMA que en el modelo ARFIMA.

Finalmente, para los datos reales, los cuales fueron extraídos directamente del paquete *cts*, el modelo continuo CAR produjo estimaciones más certeras o cercanas que el modelo AR, ya que en ambos casos vistos, el MSE fue bastante más bajo que en el modelo AR, quedando de manifiesto que para datos con tiempos irregulares, es más beneficioso modelarlos sin cambiar la naturaleza de los tiempos, es decir, ajustarlos según los modelos continuos. Se propuso trabajar con este modelo (CAR) ya que era el más adecuado en comparación al resto (CARMA y CARFIMA).

De todo lo anterior, podemos llegar a la conclusión que para datos con tiempos desigualmente espaciados, se obtienen mejores resultados conservando la naturaleza misma de sus tiempos y no discretizando estos. El único caso donde el modelo discreto fue mejor que el continuo fue en el caso $AR(1)$, pero esto puede ser debido a la simulación de los datos, ya que el paquete *cts* para modelar el proceso CAR no cuenta con una función que permita la simulación.

Mencionar que tanto el paquete *yuima* y *carfima* (que poseen funciones para modelar los procesos CARMA y CARFIMA respectivamente) no poseen ninguna función el cual permita hacer predicciones a futuro, además de que en el paquete *yuima* es algo difícil de trabajar y no está enfocado en los modelos continuos CARMA, más bien, está enfocado a la simulación de ecuaciones diferenciales estocásticas (el proceso CARMA es solo un caso particular de una ecuación diferencial estocástica).

Por último, se espera que en futuros estudios se puedan crear paquetes más amigables para modelar el proceso CARMA y que estén enfocados únicamente a este modelo, y que se puedan crear funciones donde se permita hacer predicciones a futuro de este modelo junto con el modelo CARFIMA; además de continuar en la misma línea de investigación, profundizando en la teoría de cada modelo y aportando resultados más interesantes sobre estos datos con tiempos observados de forma irregular.

Referencias

- [1] Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6), 716-723.
- [2] Belcher, J., Hampton, J. S., & Wilson, G. T. (1994). Parameterization of continuous time autoregressive models for irregularly sampled time series data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 141-155.
- [3] Brockwell, P. J. (2001). Continuous-time ARMA processes. *Handbook of statistics*, 19, 249-276.
- [4] Brockwell, P. J. (2001). Lévy-driven CARMA processes. *Annals of the Institute of Statistical Mathematics*, 53(1), 113-124.
- [5] Brockwell, P. J., Davis, R. A., & Calder, M. V. (2002). *Introduction to time series and forecasting* (Vol. 2). New York: springer.
- [6] Brockwell, P. J., & Marquardt, T. (2005). Lévy-driven and fractionally integrated ARMA processes with continuous time parameter. *Statistica Sinica*, 477-494.
- [7] Doob, J. L. (1953). *Stochastic processes* (Vol. 7, No. 2). New York: Wiley.
- [8] Duncan, T. E., Hu, Y., & Pasik-Duncan, B. (2000). Stochastic calculus for fractional Brownian motion I. Theory. *SIAM Journal on Control and Optimization*, 38(2), 582-612.
- [9] Harvey, A. C. (1990). *Forecasting, structural time series models and the Kalman filter*. Cambridge university press.
- [10] Hosking, J. R. (1981). Fractional differencing. *Biometrika*, 68(1), 165-176.
- [11] Iacus, S. M., & Mercuri, L. (2015). Implementation of Lévy CARMA model in Yuima package. *Computational Statistics*, 30(4), 1111-1141.
- [12] Jones, R. H. (1981). Fitting a continuous time autoregression to discrete data. In *Applied time series analysis II* (pp. 651-682).
- [13] Mandelbrot, B. B., & Van Ness, J. W. (1968). Fractional Brownian motions, fractional noises and applications. *SIAM review*, 10(4), 422-437.
- [14] Palma, W. (2016). *Time series analysis* (Wiley Series in Probability and Statistics, Ed.). John Wiley & Sons, Hoboken, NJ.

- [15] Pandit, S. M., & Wu, S. M. (1983). *Time series and system analysis with applications* (Vol. 3). New York: Wiley.
- [16] Rogers, L. C. G. & Williams, D. (1994a), *Diffusions, Markov processes and martingales*, Vol. 1: Foundations, 2nd edn, Cambridge University Press.
- [17] Roux, A. (2002). *Some properties of continuous-time autoregressive moving average time series models* (Doctoral dissertation, University of Pretoria Pretoria).
- [18] Shackleton, N. J. (1984). The orbital theory of Pleistocene climate: support from a revised chronology of the marine $\delta^{18}O$ record. In *Milankovitch and climate* (eds J. Imbrie, J. Hays, G. Kukla and B. Saltzman). part 1, pp. 269-305. Dordrecht: Reidel.
- [19] Shreve, S. E., & Karatzas, I. (1991). Brownian motion and stochastic calculus. *Graduate Texts in Mathematics*, 113.
- [20] Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2), 461-464.
- [21] Tómasson, H. (2015). Some computational aspects of Gaussian CARMA modelling. *Statistics and Computing*, 25(2), 375-387.
- [22] Tsai, H. (2006). Quasi-maximum likelihood estimation of long-memory limiting aggregate processes. *Statistica Sinica*, 213-226.
- [23] Tsai, H. (2009). On continuous-time autoregressive fractionally integrated moving average processes. *Bernoulli*, 15(1), 178-194.
- [24] Tsai, H., & Chan, K. S. (2000). A note on the covariance structure of a continuous-time ARMA process. *Statistica Sinica*, 989-998.
- [25] Tsai, H., & Chan, K. S. (2005). Maximum likelihood estimation of linear continuous time long memory processes with discrete time data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(5), 703-716.
- [26] Tsay, R. S. (2005). *Analysis of financial time series* (Vol. 543). John Wiley & Sons.
- [27] Wang, Z. (2013). cts: an R package for continuous time autoregressive models via Kalman filter. *Journal of Statistical Software*, 53(5), 1-19.
- [28] Wiberg, D. M. (1971). *Schaum's outline of theory and problems of state space and linear systems* (p. 117). McGraw-Hill.
- [29] Yohai, V. J. (2003). *Notas de Probabilidades y Estadística*.

Apéndice

Función MAPE

```
> MAPE<-function(real, predichos){
+   ntim <- length(real)
+   sum<-matrix(seq(1:ntim))
+   for(i in 1:ntim){
+     sum[i]<-abs(real[i]-predichos[i])/abs(real[i])
+   }
+   mape<-mean(sum[-1])
+   print(mape)
+}
```

Comparación autocorrelacion AR(1) y CAR(1)

```
>igualdad<-matrix(c(seq(1:1000),seq(1:1000),seq(1:1000)),ncol = 3)
>for(i in 1:1000) {
+   a<-runif(1,min=0.15, max=0.89)
+   datos<-arima.sim(list(order = c(1,0,0), ar=c(a)), n = 250)
+   modar<-arma(datos,order = c(1, 0))
+   igualdad[i,1]<-modar$coef[1]
+   basecar<-matrix(c(seq(1:250),datos),ncol = 2)
+   modcar <- car(basecar,scale=1,order=1)
+   alpha<-modcar$phi
+   igualdad[i,2]<-exp(-(1+alpha)/(1-alpha))
+   igualdad[i,3]<-igualdad[i,1]-igualdad[i,2]
+ }
>mean(igualdad[,3])
```

Gráfico modelo AR(1): datos vs predichos con datos igualmente espaciado

```
>library(tseries)
>library(TSA)
>datos<-arima.sim(list(order = c(1,0,0), ar=c(0.6)), n = 200)
>modar<-arma(datos,order = c(1, 0))
```

```

>plot(datos, type="l", xlab="tiempo", ylab="datos")
>title(main="Grafico modelo AR(1): datos vs predichos")
>lines(modar$fitted.values[-1],col="red")
>ree<-datos[-200]-modar$fitted.values[-1]
>mean(ree**2)
>MAPE(datos[-200],modar$fitted.values[-1])

```

Gráfico modelo CAR(1): datos vs predichos con datos igualmente espaciado

```

>library(cts)
>times<-seq(1:200)
>basecar<-matrix(c(times,datos),ncol = 2)
>modcar <- car(basecar,scale=1,order=1,ctrl=car_control(trace=TRUE))
>ntim <- dim(basecar)[1]
>plot(basecar[,1], basecar[,2], type="l", xlab="tiempo", ylab="datos")
>lines(basecar[-200,1], modcar$pre2[2:ntim], col="red")
>title(main="Grafico modelo CAR(1): datos vs predichos")
>err<-basecar[-200,2]-modcar$pre2[2:ntim]
>mean(err**2)
>MAPE(datos[-200],modcar$pre2[2:ntim])

```

MSE y MAPE para 1000 simulaciones para los modelos AR(1) y CAR(1) con datos igualmente espaciado

```

>igualdad<-matrix(c(seq(1:1000),seq(1:1000),seq(1:1000),seq(1:1000),seq(1:1000),
+ seq(1:1000),seq(1:1000),seq(1:1000),seq(1:1000),seq(1:1000)),ncol = 10)
>for(i in 1:1000) {
+ a<-runif(1,min=0.15, max=0.9)
+ datos<-arima.sim(list(order = c(1,0,0), ar=c(a)), n = 200)
+ t<-length(datos)*0.90
+ tt<-length(datos)-t
+ datos = ts(datos, start = 1)
+ modar<-arima(datos[1:t],order = c(1,0,0))
+ igualdad[i,1]<-modar$coef[1]
+ basecar<-matrix(c(seq(1:200),datos),ncol = 2)
+ modcar <- car(basecar[1:t,],scale=1,order=1,ctrl=car_control(n.ahead=tt+1))
+ alpha<-modcar$phi
+ igualdad[i,2]<-exp(-(1+alpha)/(1-alpha))
+ modar2<-arma(datos[1:t],order = c(1, 0))
+ res.ar<-datos[1:(t-1)]-modar2$fitted.values[-1]
+ res.car<-basecar[1:(t-1),2]-modcar$pre2[2:t]
+ igualdad[i,3]<-mean(res.ar**2)
+ igualdad[i,4]<-mean(res.car**2)

```

```

+ igualdad[i,5]<-MAPE(datos[1:(t-1)],modar2$fitted.values[-1])
+ igualdad[i,6]<-MAPE(datos[1:(t-1)],modcar$pre2[2:t])
+ predic<-predict(modar,n.ahead = (tt+1))
+ igualdad[i,7]<-mean((datos[-(1:t)]-predic$pred[-1])**2)
+ igualdad[i,8]<-mean((datos[-(1:t)]-modcar$predict[-1])**2)
+ igualdad[i,9]<-MAPE(datos[-(1:t)],predic$pred[-1])
+ igualdad[i,10]<-MAPE(datos[-(1:t)],modcar$predict[-1])
+}
>mean(igualdad[,3])
>mean(igualdad[,4])
>mean(igualdad[,5])
>mean(igualdad[,6])
>mean(igualdad[,7])
>mean(igualdad[,8])
>mean(igualdad[,9])
>mean(igualdad[,10])

```

Gráfico modelo AR(1): datos vs predichos con datos irregularmente espaciado

```

>a<-runif(1,min=0.1, max=0.85)
>datos<-arima.sim(list(order = c(1,0,0), ar=c(a)), n = 666)
>datos<-as.matrix(datos)
>muestra<-sample(1:nrow(datos),size=200,replace=FALSE)
>muestra<-sort(muestra, decreasing=FALSE)
>basear<-datos[muestra, ]
>t<-length(basear)*0.90
>tt<-length(basear)-t
>modar<-arima(basear[1:t],order = c(1,0, 0))
>modar2<-arma(basear[1:t],order = c(1, 0))
>res.ar<-basear[1:(t-1)]-modar2$fitted.values[-1]
>ts.plot(basear, type="l", xlab="tiempo", ylab="datos")
>rug(1:200, col="red")
>lines(1:179,modar2$fitted.values[-1], col="red")
>lines(180:199,predic$pred[-1], col="blue")
>title(main="Grafico modelo AR(1): datos vs predichos")
>mean(res.ar**2)
>MAPE(basear[1:(t-1)],modar2$fitted.values[-1])
>predic<-predict(modar,n.ahead = (tt+1))
>mean((basear[-(1:t)]-predic$pred[-1])**2)
>MAPE(basear[-(1:t)],predic$pred[-1])

```

Gráfico modelo CAR(1): datos vs predichos con datos irregularmente espaciado

```

>a<-runif(1,min=0.1, max=0.85)
>datos<-arima.sim(list(order = c(1,0,0), ar=c(a)), n = 666)
>datos<-as.matrix(datos)
>muestra<-sample(1:nrow(datos),size=200,replace=FALSE)
>muestra<-sort(muestra, decreasing=FALSE)
>basecar<-cbind(muestra,datos[muestra, ])
>k<-length(basecar[,1])/basecar[length(basecar[,1]),1]
>t<-length(basecar)*0.90
>tt<-length(basecar)-t
>modcar <- car(basecar[1:t,],scale=k,order=1,ctrl = >car_control(n.ahead = tt+1))
>res.car<-basecar[1:(t-1),2]-modcar$pre2[2:t]
>plot(basecar[,1], basecar[,2], type="l", xlab="tiempo", >ylob="datos")
>rug(basecar[,1], col="red")
>lines(basecar[1:(t-1),1],modcar$pre2[2:t], col="red")
>lines(basecar[-(1:t),1],modcar$predict[-1], col="blue")
>title(main="Grafico modelo CAR(1): datos vs predichos")
>mean(res.car**2)
>MAPE(basecar[1:(t-1),2],modcar$pre2[2:t])
>mean((basecar[-(1:t),2]-modcar$predict[-1])**2)
>MAPE(basecar[-(1:t),2],modcar$predict[-1])

```

MSE y MAPE para 1000 simulaciones para los modelos AR(1) y CAR(1) con datos irregularmente espaciado

```

>igualdad<->matrix(c(seq(1:1000),seq(1:1000),seq(1:1000),seq(1:1000),seq(1:1000),
+seq(1:1000),seq(1:1000),seq(1:1000)),ncol = 8)
>for(i in 1:1000) {
+ a<-runif(1,min=0.1, max=0.85)
+ datos<-arima.sim(list(order = c(1,0,0), ar=c(a)), n = 666)
+ datos<-as.matrix(datos)
+ muestra<-sample(1:nrow(datos),size=200,replace=FALSE)
+ muestra<-sort(muestra, decreasing=FALSE)
+ basecar<-cbind(muestra,datos[muestra, ])
+ k<-length(basecar[,1])/basecar[length(basecar[,1]),1]
+ basecar<-datos[muestra, ]
+ t<-length(basecar)*0.90
+ tt<-length(basecar)-t
+ modar<-arima(basecar[1:t],order = c(1,0, 0))
+ modcar <- car(basecar[1:t,],scale=k,order=1,ctrl = car_control(n.ahead = tt+1))
+ modar2<-arma(basecar[1:t],order = c(1, 0))

```

```

+ res.ar<-basear[1:(t-1)]-modar2$fitted.values[-1]
+ res.car<-basecar[1:(t-1),2]-modcar$pre2[2:t]
+ igualdad[i,1]<-mean(res.ar**2)
+ igualdad[i,2]<-mean(res.car**2)
+ igualdad[i,3]<-MAPE(basear[1:(t-1)],modar2$fitted.values[-1])
+ igualdad[i,4]<-MAPE(basecar[1:(t-1),2],modcar$pre2[2:t])
+ predic<-predict(modar,n.ahead = (tt+1))
+ igualdad[i,5]<-mean((basear[-(1:t)]-predic$pred[-1])**2)
+ igualdad[i,6]<-mean((basecar[-(1:t),2]-modcar$predict[-1])**2)
+ igualdad[i,7]<-MAPE(basear[-(1:t)],predic$pred[-1])
+ igualdad[i,8]<-MAPE(basecar[-(1:t),2],modcar$predict[-1])
+}
>mean(igualdad[,1])
>mean(igualdad[,2])
>mean(igualdad[,3])
>mean(igualdad[,4])
>mean(igualdad[,5])
>mean(igualdad[,6])
>mean(igualdad[,7])
>mean(igualdad[,8])

```

Gráfico modelo AR(2): datos vs predichos

```

>a1<-runif(1,min=0.1, max=0.5)
>a2<-runif(1,min=0.1, max=0.5)
>datos<-arima.sim(list(order = c(2,0,0), ar=c(a1,a2)), n = 200)
>t<-length(datos)*0.90
>tt<-length(datos)-t
>datos = ts(datos, start = 1)
>modar<-arima(datos[1:t],order = c(2,0,0))
>modar2<-arma(datos[1:t],order = c(2, 0))
>dat1<-datos[-c(1,2,200)]
>res.ar<-dat1[1:177]-modar2$fitted.values[-(1:3)]
>predic<-predict(modar,n.ahead = 16)
>plot(datos, type="l", xlab="tiempo", ylab="datos")
>title(main="Grafico modelo AR(2): datos vs predichos")
>lines(modar2$fitted.values[-1],col="red") #
>lines(181:196, predic$pred,lwd=3, col="blue")
>mean(res.ar**2)
>MAPE(dat1[1:177],modar2$fitted.values[-(1:3)])
>mean((datos[178:190]-predic$pred[-(1:3)])**2)
>MAPE(datos[178:190],predic$pred[-(1:3)])

```

Gráfico modelo CAR(2): datos vs predichos

```

>a1<-runif(1,min=0.1, max=0.5)
>a2<-runif(1,min=0.1, max=0.5)
>datos<-arima.sim(list(order = c(2,0,0), ar=c(a1,a2)), n = 200)
>t<-length(datos)*0.90
>tt<-length(datos)-t
>datos = ts(datos, start = 1)
>basecar<-matrix(c(seq(1:200),datos),ncol = 2)
>modcar <- car(basecar[1:t,],scale=1,order=2,ctrl=car_control(n.ahead=17))
>dat2<-basecar[-c(1,2,200),2]
>res.car<-dat2[1:177]-modcar$pre2[4:t]
>plot(basecar[,1], basecar[,2], type="l", xlab="tiempo", >ylob="datos")
>lines(basecar[1:179,1], modcar$pre2[2:t], col="red")
>title(main="Grafico modelo CAR(2): datos vs predichos")
>lines(178:190,modcar$predict[-(1:4)],lwd=3, col="blue")
>mean(res.car**2)
>MAPE(dat2[1:177],modcar$pre2[4:t])
>mean((datos[178:190]-modcar$predict[-(1:4)])**2)
>MAPE(datos[178:190],modcar$predict[-(1:4)])

```

MSE y MAPE para 1000 simulaciones para los modelos AR(2) y CAR(2)

```

>igualdad<-matrix(c(seq(1:1000),seq(1:1000),seq(1:1000),seq(1:1000),seq(1:1000),
+seq(1:1000),seq(1:1000),seq(1:1000)),ncol = 8)
>for(i in 1:1000) {
+ a1<-runif(1,min=0.1, max=0.5)
+ a2<-runif(1,min=0.1, max=0.5)
+ datos<-arima.sim(list(order = c(2,0,0), ar=c(a1,a2)), n = 200)
+ t<-length(datos)*0.90
+ tt<-length(datos)-t
+ datos = ts(datos, start = 1)
+ modar<-arima(datos[1:t],order = c(2,0,0))
+ basecar<-matrix(c(seq(1:200),datos),ncol = 2)
+ modcar <- car(basecar[1:t,],scale=1,order=2,ctrl=car_control(n.ahead=17))
+ modar2<-arma(datos[1:t],order = c(2, 0))
+ dat1<-datos[-c(1,2,200)]
+ dat2<-basecar[-c(1,2,200),2]
+ res.ar<-dat1[1:177]-modar2$fitted.values[-(1:3)]
+ res.car<-dat2[1:177]-modcar$pre2[4:t]
+ igualdad[i,1]<-mean(res.ar**2)
+ igualdad[i,2]<-mean(res.car**2)
+ igualdad[i,3]<-MAPE(dat1[1:177],modar2$fitted.values[-(1:3)])
+ igualdad[i,4]<-MAPE(dat2[1:177],modcar$pre2[4:t])

```

```

+ predic<-predict(modar,n.ahead = 16)
+ igualdad[i,5]<-mean((datos[178:190]-predic$pred[-(1:3)])**2)
+ igualdad[i,6]<-mean((datos[178:190]-modcar$predict[-(1:4)])**2)
+ igualdad[i,7]<-MAPE(datos[178:190],predic$pred[-(1:3)])
+ igualdad[i,8]<-MAPE(datos[178:190],modcar$predict[-(1:4)])
+}
>mean(igualdad[,1])
>mean(igualdad[,2])
>mean(igualdad[,3])
>mean(igualdad[,4])
>mean(igualdad[,5])
>mean(igualdad[,6])
>mean(igualdad[,7])
>mean(igualdad[,8])

```

Gráfico modelo CARMA(2,1): datos vs predichos

```

>mod<-setCarma(p=2,q=1,Carma.var="datos")
>>true.parm <-list(a1=1.39631,a2=0.05029,b0=1,b1=2)
>samp.L<-setSampling(Terminal=200, n=400)
>str(samp.L)
>set.seed(123)
>sim<-simulate(mod,true.parameter=true.parm,sampling=samp.L)
>str(sim)
>carmaopt <- qmle(sim, start=true.parm )
>b0<-carmaopt@coef[2]
>b1<-carmaopt@coef[1]
>a2<-carmaopt@coef[3]
>a1<-carmaopt@coef[4]
>y.fitted<-b0*sim@data@original.data[,2] + >b1*sim@data@original.data[,3]
>plot(sim@data@original.data[,1], type="l", xlab="tiempo", >ylab="datos")
>lines(y.fitted,col="red")
>title(main="Grafico modelo CARMA(2,1): datos vs predichos")
>car.resid<-sim@data@original.data[,1]-y.fitted
>mean(car.resid**2)
>MAPE(sim@data@original.data[-1,1],y.fitted[-1])

```

Gráfico modelo ARMA(2,1): datos vs predichos

```

>mod<-setCarma(p=2,q=1,Carma.var="datos")
>>true.parm <-list(a1=1.39631,a2=0.05029,b0=1,b1=2)
>samp.L<-setSampling(Terminal=200, n=400)
>set.seed(123)
>sim<-simulate(mod,true.parameter=true.parm,sampling=samp.L)
>mod1<-arma(sim@data@original.data[,1],order = c(2, 1))

```

```

>plot(sim@data@original.data[-(400:401),1], type="l", xlab="tiempo", ylab="datos")
>lines(mod1$fitted.values[-(1:2)],lwd=1,col="red")
>title(main="Grafico modelo ARMA(2,1): datos vs predichos")
>ar.re<-sim@data@original.data[-c(1,401),1]-mod1$fitted.values[-(1:2)]
>mean(ar.re**2)
>MAPE(sim@data@original.data[-c(1,401),1],mod1$fitted.values[-(1:2)])

```

Gráfico modelo CARMA(3,1): datos vs predichos

```

>Carma_brown_mod<-setCarma(p=3,q=1,Carma.var="y",Latent.var="X")
>par.Carma_brown_mod<-list(a1=4,a2=4.75,a3=1.5,b0=1,b1=0.23)
>samp<-setSampling(Terminal=200, n=400)
>set.seed(123)
>sim.Carma_brown_mod<-simulate(Carma_brown_mod,true.parameter=par.Carma_brown_mod,
+sampling=samp)
>qmle.Carma_brown_mod <- qmle(sim.Carma_brown_mod,start=par.Carma_brown_mod)
>b0<-qmle.Carma_brown_mod@coef[1]
>b1<-qmle.Carma_brown_mod@coef[2]
>a3<-qmle.Carma_brown_mod@coef[3]
>a2<-qmle.Carma_brown_mod@coef[4]
>a1<-qmle.Carma_brown_mod@coef[5]
>y.fitt<-b0*sim.Carma_brown_mod@data@original.data[,1]+
+b1*sim.Carma_brown_mod@data@original.data[,2]
>plot(sim.Carma_brown_mod@data@original.data[,1], lwd=1, type="l", xlab="tiempo",
+ylab="datos")
>lines(y.fitt, col="red")
>rug(samp@grid[[1]], col="red")
>title(main="Grafico modelo CARMA(3,1): datos vs predichos")
>mean((sim.Carma_brown_mod@data@original.data[,1]-y.fitt)**2)
>MAPE(sim.Carma_brown_mod@data@original.data[-(1:2),1],y.fitt[-(1:2)])

```

Gráfico modelo ARMA(3,1): datos vs predichos

```

>Carma_brown_mod<-setCarma(p=3,q=1,Carma.var="y",Latent.var="X")
>par.Carma_brown_mod<-list(a1=4,a2=4.75,a3=1.5,b0=1,b1=0.23)
>samp<-setSampling(Terminal=200, n=400)
>set.seed(123)
>sim.Carma_brown_mod<-simulate(Carma_brown_mod,true.parameter=par.Carma_brown_mod,
+sampling=samp)
>mod1<-arma(sim.Carma_brown_mod@data@original.data[,1],order = c(3, 1))
>plot(sim.Carma_brown_mod@data@original.data[,1], type="l", xlab="tiempo",
+ylab="datos")
>lines(mod1$fitted.values,lwd=1,col="red")
>title(main="Grafico modelo ARMA(3,1): datos vs predichos")
>mean((sim.Carma_brown_mod@data@original.data[-c(1,2,401),1]-mod1$fitted.values

```

```
+[-(1:3)]**2)
>MAPE(sim.Carma_brown_mod@data@original.data[-c(1,2,401),1],mod1$fitted.values
+[-(1:3)])
```

Gráfico modelo CARFIMA(1,H,0): datos vs predichos con datos irregularmente espaciado

```
>library(carfima)
>tiempo<-function(large){
+ length.time <- large
+ time.temp <- rexp(length.time, rate = 2)
+ time <- rep(NA, length.time + 1)
+ time[1] <- 0
+ for (i in 2 : (length.time + 1)) {
+   time[i] <- time[i - 1] + time.temp[i - 1]
+ }
+ time <- time[-1]
+}
>tt<-tiempo(120)
>parameter <- c(-0.4, 0.75, 0.2)
>simcarfima <- carfima.sim(parameter = parameter, time = tt, ar.p = 1, ma.q = 0)
>carfi1 <- carfima(Y = simcarfima, time = tt, method = "mle", ar.p = 1, ma.q = 0)
>plot(tt, simcarfima, type="l", xlab = "Tiempo", ylab = "Datos")
>zx<-carfi1$fitted.values[-1]+round(mean(simcarfima[-120]-
+carfi1$fitted.values[-1]),2)
>lines(tt[-120], zx, col="red")
>rug(serie[, 1], col = "red")
>title(main="Gráfico CARFIMA(1,H,0): datos vs predichos")
>mean((simcarfima[-120]-zx)**2)
>MAPE(simcarfima[-120],zx)
```

Gráfico modelo ARFIMA(1,d,0): datos vs predichos con datos irregularmente espaciado

```
>library(arfima)
>tiempo<-function(large){
+ length.time <- large
+ time.temp <- rexp(length.time, rate = 2)
+ time <- rep(NA, length.time + 1)
+ time[1] <- 0
+ for (i in 2 : (length.time + 1)) {
+   time[i] <- time[i - 1] + time.temp[i - 1]
+ }
+ time <- time[-1]
```

```

+}
>tt<-tiempo(120)
>parameter <- c(-0.4, 0.75, 0.2)
>simcarfima <- carfima.sim(parameter = parameter, time = tt, ar.p = 1, ma.q = 0)
>arfi1<-arfima(simcarfima, order = c(1,0,0))
>plot(simcarfima, type = "l", xlab = "Tiempo", ylab = "Datos")
>predd<-simcarfima-arfi1$modes[[1]]$residuals
>lines(predd[-1], col="red")
>rug(1:120, col = "red")
>title(main="Gráfico ARFIMA(1,d,0): datos vs predichos")
>mean((arfi1$modes[[1]]$residuals)**2)
>MAPE(simcarfima[-120],predd)

```

MSE y MAPE para 100 simulaciones para los modelos CARFIMA(1,H,0) y ARFIMA(1,d,0)

```

library(carfima)
library(arfima)
>igualdad<-matrix(c(seq(1:100),seq(1:100),seq(1:100),seq(1:100),seq(1:100)
+,seq(1:100)),ncol = 6)
>for(i in 1:100) {
+ tiempo<-function(large){
+ length.time <- large
+ time.temp <- rexp(length.time, rate = 2)
+ time <- rep(NA, length.time + 1)
+ time[1] <- 0
+ for (i in 2 : (length.time + 1)) {
+   time[i] <- time[i - 1] + time.temp[i - 1]
+ }
+ time <- time[-1]
+ }
+ tt<-tiempo(120)
+ parameter <- c(-0.4, 0.75, 0.2)
+ simcarfima <- carfima.sim(parameter = parameter, time = tt, ar.p = 1, ma.q = 0)
+ carfi1 <- carfima(Y = simcarfima, time = tt, method = "mle", ar.p = 1, ma.q = 0)
+ zx<-carfi1$fitted.values[-1]+round(mean(simcarfima[-120]-
+carfi1$fitted.values[-1]),1)
+ arfi1<-arfima(simcarfima, order = c(1,0,0))
+ predd<-simcarfima-arfi1$modes[[1]]$residuals
+ igualdad[i,3]<-mean((simcarfima[-120]-zx)**2)
+ igualdad[i,4]<-mean((arfi1$modes[[1]]$residuals)**2)
+ igualdad[i,5]<-MAPE(simcarfima[-120],zx)
+ igualdad[i,6]<-MAPE(simcarfima,predd)
+}

```

Aplicación Geofísica: Gráficos y tabla de los modelos CAR(1), CAR(3) y AR(1)

```

>a<-length(V22174[,1])/V22174[length(V22174[,1]),1]
>FP<-c("CAR(3)", "CAR(1)", "AR(1)")
>MSE.70<-c(1,1,1)
>MAPE.70<-c(1,1,1)
>MSE.30<-c(1,1,1)
>MAPE.30<-c(1,1,1)
>Autocorrelacion<-c("NA",1,1)
>tabla2<-data.frame (cbind(FP,MSE.70,MAPE.70,MSE.30,MAPE.30))
>library(cts)
>data("asth")
>car7<-car(V22174[1:155,],scale=a,order=3,ctrl=car_control(n.ahead=20))
>car1<-car(V22174[1:155,],scale=a,order=1,ctrl=car_control(n.ahead=20))
>alpha<-car1$phi
>Autocorrelacion[2]<-exp(-a*((1+alpha)/(1-alpha)))
>ar2<-arima(V22174[1:155,2],order = c(1,0,0))
>ar1<-arma(V22174[1:155,2],order = c(1,0))
>Autocorrelacion[3]<-ar1$coef[1]
>residcar7<-V22174[1:154,2]-car7$pre2[2:155]
>MSE.70[1]<-mean(residcar7**2)
>residcar1<-V22174[1:154,2]-car1$pre2[2:155]
>MSE.70[2]<-mean(residcar1**2)
>residar1<-ar1$residuals[-1]
>MSE.70[3]<-mean(residar1**2)
>predich4<-car7$pre2[2:155]
>MAPE.70[1]<-MAPE(V22174[c(1:63,65:154),2],car7$pre2[c(2:64,66:155)])
>predich1<-car1$pre2[2:155]
>MAPE.70[2]<-MAPE(V22174[c(1:63,65:154),2],car1$pre2[c(2:64,66:155)])
>MAPE.70[3]<-MAPE(V22174[c(1:63,65:154),2],ar1$fitted.values[-c(1,65)])
>predar2<-predict(ar2,n.ahead=20)
>MSE.30[1]<-mean((V22174[155:163]-car7$predict[2:10])**2)
>MSE.30[2]<-mean((V22174[155:163]-car1$predict[2:10])**2)
>MSE.30[3]<-mean((V22174[155:163]-predar2$pred[2:10])**2)
>MAPE.30<-MAPE(V22174[155:163],car7$predict[2:10])
>MAPE.30<-MAPE(V22174[155:163],car1$predict[2:10])
>MAPE.30<-MAPE(V22174[155:163],predar2$pred[2:10])
>tabla2<-data.frame (cbind(FP,MSE.70,MAPE.70,MSE.30,MAPE.30))

>plot(V22174[,1],V22174[,2], type = "l",xlab = "Tiempo", ylab = "Datos")
>rug(V22174[,1], col = "red")
>lines(V22174[1:154,1],car1$pre2[2:155], col="red")
>lines(V22174[155:164],car1$predict[1:10]+0.3, lwd=3, col="green")
>title(main="Grafico modelo CAR(1): datos vs predichos")

```

```

>plot(V22174[,1],V22174[,2], type = "l",xlab = "Tiempo", ylab = "Datos")
>lines(V22174[1:155,1],car7$pre2[1:155], col="red")
>lines(V22174[155:163],car7$predict[2:10]+0.3, lwd=3,col="green")
>title(main="Grafico modelo CAR(3): datos vs predichos")
>plot(V22174[,2], type = "l",xlab = "Tiempo", ylab = "Datos")
>lines(ar1$fitted.values[-1], col="red")
>lines(155:163, predar2$pred[2:10]+0.1, lwd=3, col="green")
>title(main="Grafico modelo AR(1): datos vs predichos")

```

Aplicación Médica: Gráficos y tabla de los modelos CAR(1), CAR(2) y AR(1)

```

>b<-length(asth[,1])/asth[length(asth[,1]),1]
>FP<-c("CAR(2)","CAR(1)","AR(1)")
>MSE.70<-c(1,1,1)
>MAPE.70<-c(1,1,1)
>MSE.30<-c(1,1,1)
>MAPE.30<-c(1,1,1)
>Autocorrelacion<-c("NA",1,1)
>tabla2<-data.frame (cbind(FP,MSE.70,MAPE.70,MSE.30,MAPE.30))
>library(cts)
>data("asth")
>car4<-car(asth[1:190,],scale=b,order=2,ctrl=car_control(n.ahead=20))
>car1<-car(asth[1:190,],scale=b,order=1,ctrl=car_control(n.ahead=20))
>alpha<-car1$phi
>Autocorrelacion[2]<-exp(-b*((1+alpha)/(1-alpha)))
>ar2<-arima(asth[1:190,2],order = c(1,0,0))
>ar1<-arma(asth[1:190,2],order = c(1,0))
>Autocorrelacion[3]<-ar1$coef[1]
>residcar4<-asth[1:189,2]-car4$pre2[2:190]
>MSE.70[1]<-mean(residcar4**2)
>residcar1<-asth[1:189,2]-car1$pre2[2:190]
>MSE.70[2]<-mean(residcar1**2)
>residar1<-ar1$residuals[-1]
>MSE.70[3]<-mean(residar1**2)
>predich4<-car4$pre2[2:190]
>MAPE.70[1]<-MAPE(asth[1:189,2],predich4)
>predich1<-car1$pre2[2:190]
>MAPE.70[2]<-MAPE(asth[1:189,2],predich1)
>MAPE.70[3]<-MAPE(asth[1:189,2],ar1$fitted.values[-1])
>predar2<-predict(ar2,n.ahead=20)
>MSE.30[1]<-mean((asth[190:198]-car4$predict[2:10])**2)
>MSE.30[2]<-mean((asth[190:198]-car1$predict[2:10])**2)
>MSE.30[3]<-mean((asth[190:198]-predar2$pred[2:10])**2)
>MAPE.30<-MAPE(asth[190:198],car4$predict[2:10])

```

```
>MAPE.30<-MAPE(asth[190:198],car1$predict[2:10])
>MAPE.30<-MAPE(asth[190:198],predar2$pred[2:10])
>tabla2<-data.frame (cbind(FP,MSE.70,MAPE.70,MSE.30,MAPE.30))

>plot(asth[,1],asth[,2], type = "l",xlab = "Tiempo", ylab = "Datos")
>rug(asth[,1], col = "red")
>lines(asth[1:189,1],car1$pre2[2:190], col="red")
>lines(asth[190:207],car1$predict[1:18], lwd=3, col="green")
>title(main="Grafico modelo CAR(1): datos vs predichos")
>plot(asth[,1],asth[,2], type = "l",xlab = "Tiempo", ylab = "Datos")
>lines(asth[1:189,1],car4$pre2[2:190], col="red")
>lines(asth[190:207],car4$predict[1:18], lwd=3,col="green")
>title(main="Grafico modelo CAR(2): datos vs predichos")
>plot(asth[,2], type = "l",xlab = "Tiempo", ylab = "Datos")
>lines(ar1$fitted.values[-1], col="red")
>lines(190:207, predar2$pred[1:18], lwd=3, col="green")
>title(main="Grafico modelo AR(1): datos vs predichos")
```