

Máster Universitario en Ciencias Actuariales y Financieras  
Curso académico 2020-2021

*Trabajo Fin de Máster*

“Estudio de la longevidad aplicando Redes  
Neuronales Artificiales”

---

**Susana Bautista Ramos**

Tutor/es

José Miguel Rodríguez Pardo del Castillo

Jesús Ramón Simón del Potro

Lugar y fecha de presentación prevista

12 de Julio de 2021

Esta tesis es propiedad del autor. No está permitida la reproducción total o parcial de este documento sin mencionar su fuente. El contenido de este documento es de exclusiva responsabilidad del autor, quien declara que no se ha incurrido en plagio y que la totalidad de referencias a otros autores han sido expresadas en el texto.

En caso de obtener una calificación igual o superior a 9.0 (Sobresaliente), autorizo la publicación de este trabajo en el centro de Documentación de la Fundación Mapfre.

Sí,  autorizo a su publicación.

No,  desestimo su publicación.

Firmado:



#### DETECCIÓN DEL PLAGIO

La Universidad utiliza el programa **Turnitin Feedback Studio** para comparar la originalidad del trabajo entregado por cada estudiante con millones de recursos electrónicos y detecta aquellas partes del texto copiadas y pegadas. Copiar o plagiar en un TFM es considerado una **Falta Grave**, y puede conllevar la expulsión definitiva de la Universidad.



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**

# Índice

<b>1. Introducción</b>	<b>1</b>
<b>Abstract</b>	<b>2</b>
<b>2. Modelos clásicos de Longevidad</b>	<b>3</b>
2.1 Modelo de Lee-Carter . . . . .	4
2.2 Modelo de Renshaw-Haberman . . . . .	6
2.3 Edad-Periodo-Cohorte . . . . .	7
2.4 Cairns-Blake-Dowd . . . . .	7
2.5 P-splines . . . . .	9
<b>3. Redes Neuronales</b>	<b>11</b>
3.1 Neurona biológica . . . . .	11
3.2 Redes Neuronales artificiales . . . . .	13
3.2.1 Estructura de una RNA . . . . .	15
3.2.1.1 Función de red . . . . .	16
3.2.1.2 Funciones de activación . . . . .	17
3.2.1.3 Arquitectura de una RNA . . . . .	18
3.2.2 Entrenamiento de RNA . . . . .	19
3.2.2.1 Algoritmos de aprendizaje . . . . .	21
<b>4. Metodología</b>	<b>23</b>
4.1. Tablas de mortalidad . . . . .	23
4.2 Redes Neuronales para el estudio de la longevidad . . . . .	24
4.2.1. Redes Neuronales Clásicas . . . . .	24
4.3. Ejercicio de Reserving . . . . .	26
<b>5. Datos y software</b>	<b>28</b>
<b>6. Resultados</b>	<b>30</b>
6.1 Ajuste y forecasting In-sample . . . . .	30
6.1.1 Resultados para el modelo Lee-Carter . . . . .	30
6.1.1.1 Proceso ARIMA . . . . .	30
6.1.1.2 Modelo de RNA . . . . .	33
6.1.2 Resultados para el modelo Renshaw-Haberman . . . . .	34
6.1.2.1 Proceso ARIMA . . . . .	35
6.1.2.2 Modelo de RNA . . . . .	37
6.1.3 Resultados para el modelo Edad-Periodo-Cohorte . . . . .	39
6.1.3.1 Proceso ARIMA . . . . .	39
6.1.3.2 Modelo de RNA . . . . .	42
6.1.4 Resultados para el modelo CBD . . . . .	43
6.1.4.1 Proceso ARIMA . . . . .	43
6.1.4.2 Modelo de RNA . . . . .	46
6.2. Ejercicio de Reserving . . . . .	48
<b>7. Conclusiones</b>	<b>51</b>
<b>Bibliografía</b>	<b>53</b>
<b>Anexo - Código R</b>	<b>56</b>

# 1.Introducción

Uno de los riesgos asociados a la vida humana que más interés despierta en el campo actuarial es el estudio del riesgo de longevidad. Este riesgo se define como la probabilidad de que las personas puedan sobrevivir más allá de lo esperado, generando una creciente preocupación en el mercado asegurador del negocio de seguros de vida debido a la posibilidad de la subestimación de las reservas, lo cual implica un riesgo de déficit de recursos financieros para cumplir las obligaciones de pago futuras. Una forma de mitigación de este tipo de riesgos es la proyección de la mortalidad de la población a nivel país, permitiendo al país o población asegurada estructurar sus planes de pensiones, o sirviendo de asistencia a las entidades aseguradoras en procesos de pricing o reserving. A lo largo del tiempo se han desarrollado diferentes técnicas y modelos orientados a la predicción de la mortalidad. Entre ellos se encuentran modelos paramétricos, como los conocidos modelos clásicos o modelos no paramétricos como el modelo de P-splines.

El desarrollo de técnicas más avanzadas, como las basadas en la Inteligencia Artificial, han permitido un estudio de la longevidad desde un nuevo paradigma, el cual podría dar lugar al nacimiento de modelos que arrojen una predicción más precisa que la aportada por los métodos hasta ahora utilizados. Una de estas técnicas es la desarrollada en el presente trabajo, basada en el estudio de la longevidad utilizando “Redes Neuronales Artificiales” , a partir de ahora RNA. Las RNA cuentan con una base matemática compleja, así como un conjunto de parametrizaciones como el número de neuronas en cada capa, tipo de aprendizaje, funciones de activación etc, que hacen que la estimación de dichos parámetros se basen en procesos de prueba y error.

La estructura del trabajo es la siguiente. En el punto 2 desarrollaremos teóricamente los modelos clásicos hasta ahora más utilizados para el estudio de la longevidad. El punto 3 está destinado a obtener un conocimiento más profundo sobre las Redes Neuronales, desde sus orígenes hasta su estructura. En el punto 4 nos centraremos en la metodología utilizada para el desarrollo del trabajo. El punto 5 se centra en el análisis de la obtención de los datos necesarios para el estudio. En el punto 6 analizaremos los resultados obtenidos de nuestro estudio y finalmente en el punto 7 expondremos las conclusiones obtenidas.

# Abstract

One of the most interesting human life associated risks in the actuarial field is the longevity risk study. This risk is defined as the probability of persons living longer than expected, increasing the insurance market's concern because of the possibility of reserve underestimates, which could lead to a risk of financial resources deficit to fulfill future liabilities. A way of mitigating these kinds of risks is to forecast the mortality rate at country level, allowing the country or population safeguarded structure in their pension scheme, or supporting insurance pricing or reserving process. Over time different techniques and models have been developed to predict mortality rates. Between them we can find parametric models as classic models or non-parametric models as P-splines. The advanced of development techniques, such as Artificial Intelligence, has allowed longevity to be studied from a new point of view, which could lead to the birth of new more precise predictions models than the more using models until now. One of these techniques is the focus of this study, based on the longevity study using "Artificial Neural Networks" (ARN). ARN has a complex mathematical base, as well as a number of parameters as the neuron's numbers in each layer, learning process, activation functions etc. which means that the estimation of these parameters are based on a trial and error process.

The project's structure is the following. In the second chapter we'll develop the more used classical models from a theoretical point of view. In the third chapter we'll earn a deeper knowledge and understanding of Neural Networks, from their origins to their structure. in the fourth chapter we're going to develop the methodology of our project. In the fifth chapter we're going to analyse the source of the needed data for our project. In the sixth chapter we're going to analyse the results of our project and finally, in the seventh chapter, we're going to present our conclusions.

## 2. Modelos clásicos de Longevidad

El incremento de la longevidad observado en el último siglo es una situación que concierne a la industria aseguradora en gran medida (Hainaut, 2018). La existencia de cambios recientes en la mortalidad han contribuido a definir un estado “cambiante”, cuyas tendencias en la edad adulta revelan dos características diferentes. En edades avanzadas las probabilidades de fallecimiento disminuyen, mientras que lo contrario ocurre en edades más tempranas, en particular si nos centramos en el tramo de edad entre los 20 y los 40 años (Rodríguez-Pardo et al.,2014).

Este estado cambiante nos indica que no podemos asumir una hipótesis de mortalidad estática, al menos para periodos largos de tiempo, por lo que deberíamos de considerar una naturaleza estocástica de la mortalidad. Así, desde comienzos de la década de los 90 del siglo anterior, se han desarrollado una serie de modelos que tratan de explicar este fenómeno. Entre ellos, modelos paramétricos entre los que se incluyen Lee-Carter (Lee y Carter, 1992) y sus extensiones posteriores u otros no paramétricos como P-splines (Currie, 2006).

Diremos que una variable aleatoria  $\mathbf{X}$  sigue un modelo paramétrico si su distribución de probabilidad pertenece a la familia de distribuciones indexadas por un parámetro de dimensión finita. En cambio, diremos que la variable aleatoria  $\mathbf{X}$  sigue un modelo no paramétrico si sobre su distribución se suponen algunas condiciones de regularidad.

Para culminar, expondremos algunas de las propiedades que deberían cumplir los modelos de longevidad para ser considerados adecuados (Cairns, et al., 2008):

- La tasa de mortalidad ha de ser positiva.
- El modelo debe ser consistente con los datos históricos.
- La dinámica a largo plazo del modelo debe ser biológicamente razonable.
- Los parámetros estimados, relativos al periodo de los datos y rango de edades empleados, deben ser robustos.
- El modelo debe ser relativamente parsimonioso.

- Para algunos países, el modelo debería incorporar el efecto cohorte.

Modelos clásicos	
Lee-Carter	$m(x, t) = \exp[\beta_x^{(1)} + \beta_x^{(2)} \kappa_t^{(2)}]$
Renshaw-Haberman	$\log m(x, t) = \beta_x^{(1)} + \beta_x^{(2)} \kappa_t^{(2)} + \beta_x^{(4)} \gamma_{t-x}^{(3)}$
Edad-Periodo-Cohorte	$\log m(x, t) = \beta_x^{(1)} + \frac{1}{n_x} \kappa_t^{(2)} + \frac{1}{n_x} \gamma_{t-x}^{(3)}$
Cairns-Blake-Dowd	$\text{logit } q(x, t) = \beta_x^{(1)} \kappa_x^{(1)} + \beta_x^{(2)} \kappa_x^{(2)} + \beta_x^{(3)} \gamma_{t-x}^{(3)}$
P-splines	$\log m(x, t) = \sum_{i,j} \theta_{ij} B_{ij}(x, t)$

Tabla 1: Modelos clásicos para el estudio de la longevidad

## 2.1 Modelo de Lee-Carter

Lee y Carter propusieron este modelo en 1992 con el fin de predecir la mortalidad de EEUU debido a la creciente preocupación por el incremento de la esperanza de vida desde los años 1900 a 1988. Su modelo se basó en datos históricos comprendidos entre los años 1933 y 1987. Además asumen que la fuerza de mortalidad tiene una estructura lineal con respecto al tiempo y que las covariables dependen de la edad (Lee y Carter, 1992).

El modelo es dado por:

$$m(x, t) = \exp[\beta_x^{(1)} + \beta_x^{(2)} \kappa_t^{(2)}] \quad (1)$$

Donde  $m(x, t)$  es la tasa central de mortalidad para una edad  $x$  en un tiempo  $t$ ,  $\beta_x^{(k)}$  son efectos de la edad  $x$  y  $\kappa_t^{(2)}$  mide la tendencia de la mortalidad en el tiempo.

La estimación de la función se realiza a través del proceso de máxima verosimilitud. Además,  $\kappa_t$ , habitualmente es modelizado como un proceso ARIMA, asumiendo sus comportamientos futuros a través de un paseo aleatorio con deriva, planteado de la siguiente forma:

$$\kappa_{t+1} = \kappa_t + \text{drift} + \epsilon_{t+1} \quad (2)$$

$$\epsilon_{t+1} \sim N(0, \sigma^2)$$

Bajo este modelo  $\beta_x^{(1)}$  representa el promedio de la tasa de mortalidad logarítmica a lo largo del tiempo a una edad  $x$ , mientras que  $\beta_x^{(2)}$  representa la velocidad con la que cambia la fuerza de mortalidad en cada edad a lo largo del tiempo (Cairns et al., 2008).

Con el fin de asegurar la identificabilidad del modelo, Lee y Carter impusieron dos restricciones dadas por:

$$\sum_{t=1}^n k_t^{(2)} = 0 \quad \sum_{x=x_1}^{x_p} \beta_x^{(2)} = 1. \quad (3)$$

El método de Lee-Carter es apropiado para extrapolar distribuciones de mortalidad con tendencias de nivel y edad históricas, pero la extrapolación no siempre es el procedimiento más adecuado. A continuación reflejaremos algunos de los problemas del modelo:

- Es un modelo de un único factor, lo que genera una correlación perfecta entre todas las edades cuando se produce una mejora de la mortalidad (Cairns et al., 2008).
- El modelo asume un cierto patrón de cambios en la distribución de la edad de mortalidad, manteniendo constante la caída de la tasa de mortalidad a lo largo del tiempo. Pero en la práctica, la velocidad relativa de caída puede variar en función de la edad (Lee, 2000).
- El modelo básico de Lee-Carter puede generar una falta de uniformidad al estimar el efecto de la edad,  $\beta_x^{(2)}$ . Podemos realizar una predicción utilizando datos out-of sample tomando los parámetros  $\hat{\beta}_x^{(1)}$  y  $\hat{\beta}_x^{(2)}$  y estimando  $\kappa_x^{(2)}$  para cada año futuro. La función para calcular los residuos estandarizados para el periodo out-of sample viene dada por:

$$\epsilon(x, t) = \frac{(D(x, t) - m(x, t)E(x, t))}{\sqrt{m(x, t)E(x, t)}}$$

Siendo  $D(x, t)$  el número de fallecimientos a una edad  $x$  en el tiempo  $t$  y  $E(x, t)$  la esperanza de vida a una edad  $x$  en el tiempo  $t$ . Si el método es correcto, entonces  $\epsilon(x, t)$  debería ser Normal, idéntica e independientemente distribuída (Niid). Experimentos preliminares han indicado que  $\hat{\beta}_x^{(2)}$  carece de uniformidad, por lo que no cumple la condición anterior (Cairns et al., 2008).

- Lee y Miller identificaron un problema de sesgo (bias) y su solución propuesta surgió de la incapacidad de realizar un ajuste correcto de los datos por parte del modelo de Lee-Carter (Cairns et al., 2008).



## 2.2 Modelo de Renshaw-Haberman

Renshaw y Haberman (2006) realizan una extensión del modelo de Lee-Carter en la que incorporan un parámetro para calcular el efecto global de la cohorte en  $t - x$ .

$$\log m(x, t) = \beta_x^{(1)} + \beta_x^{(2)} \kappa_t^{(2)} + \beta_x^{(4)} \gamma_{t-x}^{(3)} \quad (4)$$

Donde  $\kappa_t^{(2)}$  es un efecto de periodo aleatorio y  $\gamma_{t-x}^{(3)}$  es un efecto de cohorte aleatorio en función del año de nacimiento ( $t - x$ ).

El efecto cohorte busca capturar los cambios reflejados en la mortalidad de una generación a lo largo del tiempo con respecto a otras generaciones precedentes y/o posteriores. Entre otros, estos cambios pueden deberse a la sucesión de eventos catastróficos como guerras o epidemias, o por el contrario, hechos que supongan una mejora de la mortalidad en una generación determinada.

Para la estimación del modelo los autores asumen que el número de fallecimientos  $D(x, t)$  se distribuyen según una Poisson y utilizan la función log-link sobre el tanto instantáneo de mortalidad  $m(x, t)$ . Más adelante, para resolver problemas de estabilidad, los autores proponen definir  $\beta_x^{(3)} = 1$ . El predictor quedaría invariante bajo la transformación de parámetros:

$$(\beta_x^{(1)}, \beta_x^{(2)}, \kappa_t^{(2)}, \gamma_{t-x}^{(3)}) \rightarrow (\beta_x^{(1)} + c_1 \beta_x^{(2)} + c_2, \frac{1}{c_3} \beta_x^{(2)}, c_3(\kappa_t^{(2)} - c_1), \gamma_{t-x}^{(3)} - c_2) \quad (5)$$

Con  $c_1, c_2, c_3 \in \mathbb{R}$ ,  $c_1 \neq 0, c_2 \neq 0, c_3 \neq 0$ . Este modelo está sujeto a las siguientes restricciones:

$$\sum_t \kappa_t^{(2)} = \sum_{t-x} \gamma_{t-x}^{(3)} = 0 \quad \sum_t \beta_x^{(2)} = \sum_t \beta_x^{(3)} = 1 \quad (6)$$

Las cuales son obtenidas imponiendo:

$$c_1 = \frac{1}{n} \sum_t \kappa_t^{(2)}, \quad c_2 = \frac{1}{n + \kappa - 1} \sum_{c=t_1-x_k}^{t_n-x_1} \gamma_c, \quad c_3 = \sum_x \beta_x^{(2)} \quad (7)$$

Los parámetros son estimados a través de un proceso iterativo.

## 2.3 Edad-Periodo-Cohorte

El modelo Edad-Periodo-Cohorte, como su propio nombre indica, se considera un modelo para la edad, año de nacimiento y efectos de fallecimiento sobre una generación, donde dichos efectos influyen en las tasas de mortalidad de manera independiente. Este modelo ha sido considerado previamente por Currie (2006) y Renshaw y Haberman (2006), aunque la referencia clásica es atribuida a Clayton y Schifflers (1987). El modelo es dado por:

$$\log m(x, t) = \beta_x^{(1)} + \frac{1}{n_\alpha} \kappa_t^{(2)} + \frac{1}{n_\alpha} \gamma_{t-x}^{(3)} \quad (8)$$

Donde  $n_\alpha$  es el número de edades en el set de datos. Este modelo es un caso especial del modelo de Renshaw-Haberman cuando  $\beta_x^{(2)} = \frac{1}{n_\alpha}$  y  $\beta_x^{(3)} = \frac{1}{n_\alpha}$  (Cairns, et al., 2009). Los parámetros están sujetos a las siguientes restricciones:

$$\sum_t \kappa_t = 0 \quad \sum_{x,t} \gamma_{t-x} = 0 \quad (9)$$

## 2.4 Cairns-Blake-Dowd

Uno de las alternativas más populares al modelo de Lee-Carter en la literatura actuarial es el modelo CBD, propuesto por Cairns, et al. (2006). Cairns, et al. (2006b) se centran en edades avanzadas (60 a 89 años) y utilizan un patrón de mortalidad relativamente simple a estas edades para ajustar un modelo más parsimonioso basado en la transformación logística de la tasa de mortalidad (Cairns et al., 2008):

$$\text{logit } q(x, t) = \log \frac{q(x, t)}{1 - q(x, t)} = \beta_x^{(1)} \kappa_t^{(1)} + \beta_x^{(2)} \kappa_t^{(2)} \quad (10)$$

Donde  $\beta_x^{(2)} = (x - \bar{x})$ ,  $\beta_x^{(1)} = 1$  y  $(\kappa_t^{(1)}, \kappa_t^{(2)})$  se modeliza como una serie temporal bivalente. Esta especificación no presenta ningún problema de identificabilidad, por lo que no será necesario imponer restricciones.

Cairns, Blake y Dowd propusieron extensiones de su modelo en los años sucesivos:

- CBD con efecto cohorte.

Este modelo es una generalización del CBD que incluye el efecto cohorte:

$$\text{logit } q(x, t) = \beta_x^{(1)} \kappa_x^{(1)} + \beta_x^{(2)} \kappa_x^{(2)} + \beta_x^{(3)} \gamma_{t-x}^{(3)} \quad (11)$$

Para este modelo asumimos:

$$\beta_x^{(1)} = 1 \quad \beta_x^{(2)} = (x - \bar{x}) \quad \beta_x^{(3)} = 1 \quad (12)$$

Por lo que tendríamos:

$$\text{logit } q(t, x) = \kappa_x^{(1)} + (x - \bar{x}) \kappa_x^{(2)} + \gamma_{t-x}^{(3)} \quad (13)$$

Debido a los problemas de identificabilidad del modelo, podemos sustituir  $\gamma_{t-x}^{(3)}$  por  $\tilde{\gamma}_{t-x}^{(3)} = \gamma_{t-x}^{(3)} + \phi_1 + \phi_2(t - x - \bar{x})$  y con los correspondientes ajustes de  $\kappa_t^{(1)}$  y  $\kappa_t^{(2)}$  no habría impacto sobre los valores ajustados de  $q(t, x)$  (Cairns et al., 2009). Esto requiere las siguientes dos restricciones:

$$\sum_{c \in C} \gamma_c^{(3)} = 0 \quad \sum_{c \in C} c \gamma_c^{(3)} = 0 \quad (14)$$

Donde C es el set de cohortes incluidos en el análisis. Estas restricciones provocan que  $\hat{\phi}_1 = 0$  y  $\hat{\phi}_2 = 0$  con el fin de que  $\gamma_c^{(3)}$  fluctue sobre 0 y no tenga una tendencia lineal discernible (Cairns et al., 2009).

- CBD añadiendo un término cuadrático sobre el efecto cohorte.

Este modelo es una generalización del modelo anterior, incluyendo un término cuadrático sobre el efecto cohorte. La inclusión de este término se debe a la posible identificación de una curvatura en los gráficos de  $\text{logit } q(t, x)$  (Cairns et al., 2009).

El modelo viene dado por:

$$\text{logit } q(t, x) = \kappa_x^{(1)} + \kappa_x^{(2)}(x - \bar{x}) + \kappa_x^{(3)}((x - \bar{x})^2 - \hat{\sigma}_x^2) + \gamma_{t-x}^{(4)} \quad (15)$$

Donde la constante  $\hat{\sigma}_x^2 = n_x^{-1} \sum_i (x - \bar{x})^2$  es la media de  $(x - \bar{x})^2$ .

Al igual que en el modelo anterior, tenemos problemas de identificabilidad, por lo que podemos sustituir  $\gamma_{t-x}^{(4)}$  por  $\tilde{\gamma}_{t-x}^{(4)} = \gamma_{t-x}^{(4)} + \phi_1 + \phi_2(t - x - \bar{x}) + \phi_3(t - x - \bar{x})^2$  teniendo en cuenta los correspondientes ajustes de  $\kappa_t^{(1)}$ ,  $\kappa_t^{(2)}$  y  $\kappa_t^{(3)}$ . Para prevenir elecciones arbitrarias sobre  $\phi_1$ ,  $\phi_2$  y  $\phi_3$ , se impondrán las siguientes restricciones:

$$\sum_{c \in C} \gamma_c^{(4)} = 0 \quad \sum_{c \in C} c \gamma_c^{(4)} = 0 \quad \sum_{c \in C} c^2 \gamma_c^{(4)} = 0 \quad (16)$$

Para no sobrecargar este apartado, citaremos otros modelos clásicos de longevidad como la ley de Moivre (1725), la ley de Gompertz (1825), la ley de Makeham (1859) y el modelo de Kannistö (1992).

## 2.5 P-splines

Existen diferentes variantes de splines, entre los que se encuentran los splines suavizados, los splines regresivos o los B-splines, los cuales son constituidos a partir de piezas polinómicas unidas en ciertos puntos de  $x$ , conocidos como nodos. El modelo propuesto por Currie et al.(2006) consiste en la aplicación de penalizaciones a los coeficientes de la variante B-splines, dando lugar a los P-splines, la cual es una generalización del modelo de regresión (Eilers y Marx, 1996):

La fórmula generalizada de P-splines viene dada por:

$$\log m(x, t) = \sum_{i,j} \theta_{ij} B_{ij}(x, t) \quad (17)$$

Siendo  $\theta_{ij}$  el parámetro a ser estimado y  $B_{ij}$  la base para la regresión(B-splines).

El modelo P-splines se caracteriza por tener las siguientes propiedades:

- No muestra efectos de frontera, el cual supone que al extender la curva fuera del dominio de  $x$  esta caiga hacia 0.
- Ajusta de forma exacta los polinomios.
- Conserva los momentos independientemente del parámetro de suavizado.

Los modelos aditivos son una clase importante dentro de los modelos multidimensionales, ya que la hipótesis de aditividad controla posibles problemas de computación derivados de la estimación de curvas. Sin embargo esta hipótesis es demasiado restrictiva, presentando limitaciones de suavizado. Bajo esta hipótesis se han desarrollado modelos dentro del campo de P-spline como los modelos mixtos, que utilizan bases radiales. En este caso nos encontraríamos ante un modelo isotrópico, cuyo método de suavizado no es adecuado para variables explicativas medidas en diferentes unidades (Durbán, 2008). Como alternativa, Durbán et al. (2002) propuso el producto tensorial de B-splines como base para el suavizado de dos o más dimensiones, cuya característica reside en la penalización de las filas y columnas de la matriz.

### 3. Redes Neuronales

La Inteligencia Artificial, en adelante IA, ha permitido la posibilidad de instruir a máquinas con el fin de la realización, de forma trivial, de actividades rutinarias del ser humano. Dentro de la IA podemos diferenciar varias ramas computacionales, entre las que destacamos la corriente de actuación de aprendizaje automático, en inglés Machine Learning, en adelante ML.

ML permite el aprendizaje automático de algoritmos a través de la utilización de datos anteriores. Esta corriente posee la característica de ser autoescalable. Dentro del ML podemos encontrar la rama de Deep Learning, la cual permite la utilización de redes neuronales, cuyos modelos computacionales imitan características arquitecturales del sistema nervioso biológico, permitiendo que dentro de la propia estructura del sistema se encuentren redes de unidades de proceso que permitan extraer características ocultas de los datos utilizados.

#### 3.1 Neurona biológica

El tejido nervioso, compuesto por cien mil millones de neuronas interconectadas, actúa como la base de datos y centro de comunicaciones del organismo, el cual se sustenta en la presencia de receptores que reciben y transmiten información a los centros encargados de su procesamiento, con la que generan nueva información y la compara con la ya almacenada (Gartner y Hiatt, 2011).

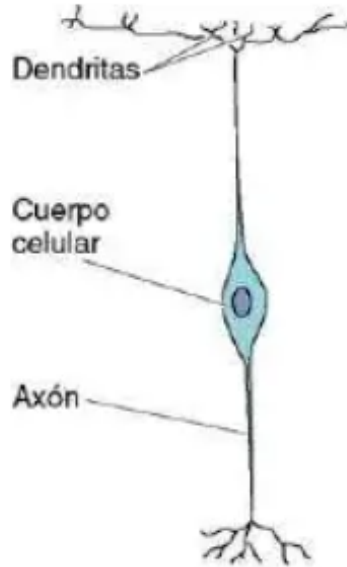
La mayor parte de las neuronas están formadas por tres partes diferenciadas (ver figura 1):

- Cuerpo celular: Base de la cual emergen una serie de prolongaciones de donde surgen ramificaciones (Crossman y Neary, 2007).
- Dendritas: Proyecciones celulares encargadas de la recepción de estímulos internos y externos que transmiten en forma de impulso hacia el cuerpo celular (Gartner y Hiatt, 2011).
- Axon: Aporta inervación motora a músculos y glándulas. Finalizan en terminales

axonales en los que se establecen sinapsis con otras células (Gartner y Hiatt, 2011).

**Figura 1**

*Estructura de una neurona.*



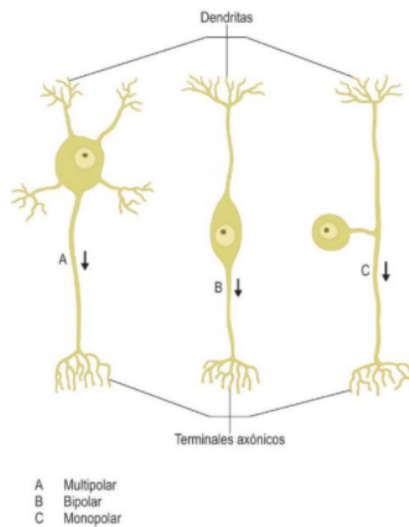
Tomado de Histología básica por Gartner LP, Hiatt JL, 2011, Elsevier.

Las neuronas pueden seguir tres patrones básicos (ver figura 2):

- Neuronas multipolares: Son las más frecuentes. Poseen una única proyección que bifurca en una rama periférica que continua hasta la célula a la que inerva (Gartner y Hiatt, 2011).
- Neuronas bipolares: Poseen dos proyecciones: una dendrita y un axón (Gartner y Hiatt, 2011).
- Neuronas monopolares: Poseen una única prolongación que parte del cuerpo celular (Crossman y Neary, 2007).

## Figura 2

Patrones básicos de las neuronas.



Tomado de Histología básica por Gartner LP, Hiatt JL, 2011, Elsevier.

Cada neurona conforma una entidad física independiente. Para que sea posible la producción del procesamiento de información en redes de neuronas, esta ha de pasar entre neuronas. Este proceso se da en la sinapsis (Crossman y Neary, 2007).

La sinapsis está formada por tres componentes:

- Elemento presináptico: Aquella neurona que emite la información.
- Elemento postsináptico: Neurona receptora de la información.
- Hendidura sináptica: Espacio por el cual se produce la transmisión de la información.

El proceso de sinapsis consiste en la liberación de un neurotransmisor desde el elemento presináptico hasta la hendidura sináptica. Este neurotransmisor se unirá a los receptores de la membrana postsináptica, la cual sufrirá cambios en su potencial de acción, pudiendo ser excitada o inhibida (Haines y Mihailoff, 2002).

## 3.2 Redes Neuronales artificiales

*“Una red neuronal es un procesador masivamente paralelo distribuido, formado por unidades de procesamiento simples, que tiene una propensión natural a almacenar*



*conocimiento experimental y hacerlo disponible para su uso. Se asemeja al cerebro en dos aspectos” (Haykin, 1999):*

- El conocimiento es adquirido por la red a través de un proceso de aprendizaje
- La fuerza de interconexión neuronal, conocida como pesos sinápticos, es utilizada para almacenar el conocimiento adquirido.

*“Una red neuronal es un conjunto de elementos de procesamiento simples interconectados, unidades o nodos, cuyo funcionamiento está basado en la neurona animal. La capacidad de procesamiento de la red neuronal es almacenada en unidades de peso, obtenidas por un procesamiento de adaptación a, o aprendidas de, un set de patrones de entrenamiento” (Gurney, 1997).*

Estas son dos de las definiciones de las Redes Neuronales Artificiales, las cuales tienen su origen en McCulloch y Pitts (1943) con la publicación de su artículo *“A logical calculus of the ideas immanent in nervous activity”*. La colaboración de un neurofisiólogo y un matemático hicieron posible la elaboración de un modelo de neurona describiendo su actividad como un proceso “todo o nada”, donde la conexión sináptica podía activar la neurona en cualquier momento independientemente de la actividad anterior.

Posteriormente Hebb (1949) supone el hecho de que la repetición persistente de una acción sobre una célula puede provocar cambios sobre esta estableciendo que el aprendizaje de una red neuronal se basa en el proceso de crecimiento o cambio en el metabolismo de dicha célula. En 1951 Marvin Minsky, conocido como el padre de la IA, creó la primera RNA, “SNARC”.

Rosenblatt (1958) desarrolló el perceptrón el cual puede interpretar patrones que no sean necesariamente geométricos. Un año después Widrow y Hoff (1960) desarrollaron el modelo ADALINE (ADAPTative LINEar Elements) la cual fue la primera red neuronal aplicada a una situación real. Minsky y Papert (1969) publicaron el libro “Perceptrons”, donde probaron que este no era capaz de resolver problemas relativamente fáciles. A partir de esto se produjo una caída de la inversión a este tipo de proyectos, dando paso al periodo conocido como “Invierno de la IA”. Después de 15 años, Rumelhart et al. (1986) redescubrieron el algoritmo de backpropagation. Gracias a esto se retomó la

investigación sobre la IA.

Las redes neuronales utilizadas en el ámbito de la longevidad buscan obtener unas predicciones más precisas. Durante los últimos años se han utilizado estas técnicas como soporte de los modelos clásicos como el de Lee-Carter, introduciendo por ejemplo Redes Neuronales Recurrentes Long Short-Term Memory, las cuales permiten obtener una predicción de la mortalidad más coherente, ya que es capaz de almacenar relaciones significativas entre datos, preservando información a lo largo del tiempo (Nigri et al., 2019).

### **3.2.1 Estructura de una RNA**

De forma similar a la estructura de las redes neuronales biológicas, las RNA definen la neurona como una unidad de procesamiento central que ejecuta una operación matemática para obtener un output (salida) a partir de un set de inputs. El output de una neurona es una función de la suma ponderada de los inputs más el sesgo. Cada neurona realiza una operación muy sencilla cuya activación tendrá lugar si la cantidad de señal recibida supera el umbral de activación (Ciaburro y Venkateswaran, 2017).

En cada RNA se encuentran tres tipos de neuronas, estructuradas en un enfoque de capas que es también seguido por las redes neuronales biológicas:

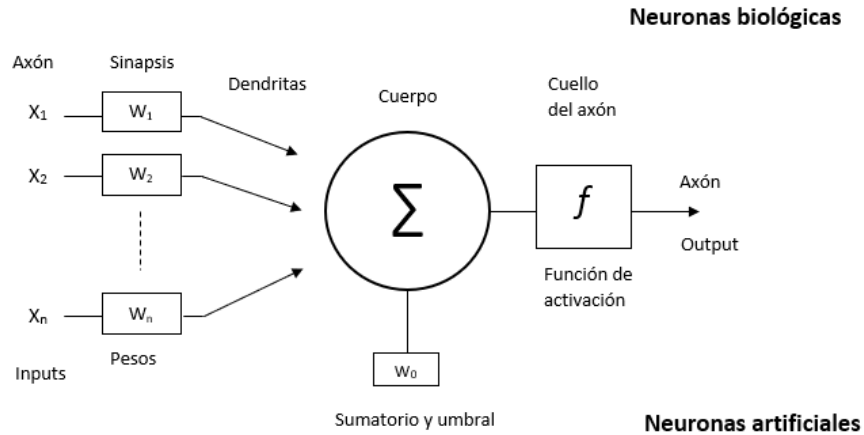
- Capa de entrada: También conocida como capa sensorial, está compuesta por neuronas que reciben información del exterior, las cuales pueden ser utilizados con propósitos de entrenamiento y predicción.
- Capa oculta: Son las capas intermedias donde se produce el proceso de aprendizaje, para ello recopila la información necesaria de la capa sensorial. Para llevar a cabo los procesos desarrollados deben interactuar los elementos de sesgo (bias) y pesos (ponderaciones).
- Capa de salida: Está compuesta por neuronas que, una vez procesada toda la información, proporcionan la respuesta de la red neuronal.

Cada unidad de proceso o neurona está compuesta por un conjunto de entradas y pesos

sinápticos, una función de red consistente en una combinación lineal entre las entradas y los pesos sinápticos, siendo habitual añadir a estos últimos un parámetro adicional denominado umbral, el cuál suele restar al potencial pos-sináptico. Una función de activación aplicada por un núcleo central de proceso y por último una función de salida limitada a reflejar el valor devuelto por la función de activación. Un proceso esquemático de lo anteriormente descrito puede verse en la figura 3.

**Figura 3**

*Estructura de neurona biológica y artificial.*



Adaptado de Principles of Artificial Networks, por Graupe.P, 2007, Scientific World.

### 3.2.1.1 Función de red

Como hemos mencionado anteriormente, la función de red es una combinación lineal entre las entradas y los pesos sinápticos, la cual es representada de la siguiente forma:

$$Net_j = \sum_{i=1}^n w_{j,i} x_i \quad (18)$$

Donde el subíndice  $j$  se refiere a la unidad de proceso  $j$ ,  $w_{ij}$  son los pesos sinápticos que une la neurona  $i$  con la  $j$  de una capa posterior y finalmente  $x_i$  es la salida de una neurona correspondiente a la capa anterior.

En el caso de incluir el parámetro sesgo, denotado como  $\theta$ , tendríamos:

$$Net_j = \sum_{i=1}^n w_{j,i} x_i + \theta \quad (19)$$

### 3.2.1.2 Funciones de activación

Las funciones de activación son el núcleo de una RNA. Esta es una función matemática que transforma una entrada global en un valor de activación. Sin las funciones de activación, la RNA generaría el mismo resultado que una función lineal (Ciaburro y Venkateswaran, 2017). Si denotamos dicha función por  $y_i$  tenemos:

$$y_i = f(Net_i) \quad (20)$$

Dentro de las funciones de activación podemos encontrar varios tipos: lineales, de umbral y no lineales. Las funciones lineales, también conocidas como identidad, implican que la salida será igual a la entrada, por lo que en caso de aplicarla en una red neuronal de varias capas, tendríamos una regresión lineal. Por otro lado, las funciones de umbral nos proporcionan una salida binaria. Es decir, el resultado asume el valor 0 en caso de valores negativo y el valor 1 en caso contrario.

$$f(x) = \begin{cases} 0, & \text{si } x < 0 \\ 1, & \text{si } x \geq 0 \end{cases} \quad (21)$$

Por último, las funciones no lineales son utilizadas por la mayoría de las RNA. Estas permiten crear complejos mapeos entre inputs y outputs, necesarios para el proceso de aprendizaje. Con respecto a las funciones lineales, las funciones no lineales, debido a su diferenciabilidad, permiten llevar a cabo el algoritmo de “backpropagation”, ya que será posible obtener una derivada que conecte con el input. Dentro de las funciones no lineales podemos destacar:

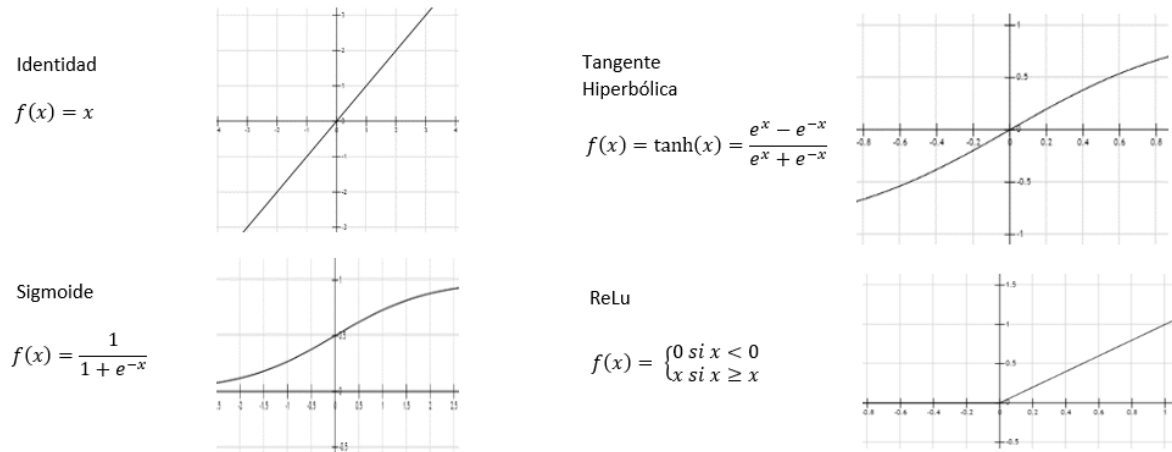
- **Función sigmoide:** Es una de las funciones de activación más importante en la construcción de redes neuronales. Es definida como una función estrictamente positiva que toma cualquier valor entre 0 y 1 (Ciaburro y Venkateswaran, 2017). Además presenta un balance entre los comportamientos lineales y no lineales. Un ejemplo de función sigmoidea es la función logística, la cuál está definida en un rango de valores de salida entre 0 y 1, los cuales pueden ser interpretados como

una probabilidad.

- Función tangente hiperbólica: Esta función es muy similar a la función sigmoide, con la diferencia en que su rango está definido entre -1 y 1.
- Función ReLu (Rectified Linear Unit): Es la función más utilizada, ya que permite un proceso de aprendizaje rápido en las RNA. El rango de salida se encontrará entre 0 e infinito. Cuando obtenemos que su función y derivada son iguales a 0, se genera una muerte de neuronas. A pesar de que este hecho puede ser negativo en algunos casos, permite la regularización “Dropout”, la cual supone un proceso de descarte probabilístico. La figura 4 muestra un ejemplo de cada tipo de función de activación.

**Figura 4**

*Funciones de activación.*



Elaboración propia.

### 3.2.1.3 Arquitectura de una RNA

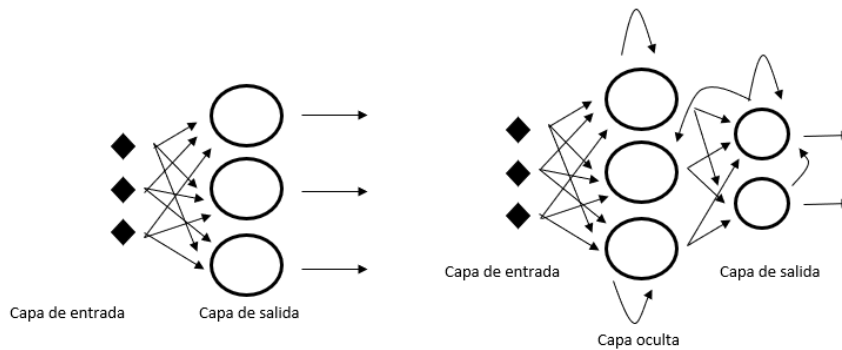
La arquitectura de una RNA comprende su topología, estructura o patrón de conexionado. Atendiendo al flujo de información podemos considerarlas redes de propagación hacia delante (feedforward) o redes recurrentes o realimentadas (feedback). Considerando su estructura distinguimos entre redes monocapas o redes multicapas.

- Redes feedforward: La información fluye desde la capa de inputs hacia la capa de outputs de forma estrictamente unidireccional.

- Redes feedback: La información puede fluir entre neuronas de un mismo nivel (incluidas ellas mismas) o de distinto nivel. La idea de estas redes es buscar la activación de una neurona durante un periodo de tiempo, para así poder estimular otras neuronas con el fin de alcanzar una activación de neuronas en cascada (Nielsen, 2015).
- Redes monocapas: Es la red neuronal más simple. Está compuesta por la capa de salida, ya que la capa de entrada, al no realizar ningún cálculo, no es considerada. Esta red simplemente proyecta la información de entrada hacia la capa de salida.
- Redes multicapas: Es una generalización de las redes monocapas, compuestas por varias capas intermedias (capas ocultas), en cuyas neuronas se realizan los procesos de cálculos. Una representación gráfica de los tipos arquitectura pueden verse en la figura 5.

**Figura 5**

*Redes feedforward monocapa y feedback multicapa de tres neuronas ocultas.*



Adaptado de Neural Networks: A comprehensive foundation, por S.Haykin, 1998, Pearson Prentice Hall.

### 3.2.2 Entrenamiento de RNA

Una RNA tiene la capacidad de aprender de su entorno a través de un proceso interactivo de ajuste aplicado a sus pesos sinápticos y bias. Normalmente la red adquiere más conocimiento de su entorno después de cada proceso iterativo de aprendizaje (Haykin, 1998).

El proceso de aprendizaje puede ser clasificado en dos categorías: Aprendizaje super-

visado y aprendizaje no supervisado.

- Aprendizaje supervisado: Se caracteriza por necesitar un agente que aporte a la red la respuesta que debería dar sobre la base de los inputs recibidos. Esto permitirá que la red pueda comparar los resultados observados con los esperados, de tal forma que pueda corregirse a sí misma hasta que ambos resultados sean lo más parecidos posibles. Dentro del aprendizaje supervisado podemos distinguir tres formas de llevarlo a cabo.
  - Aprendizaje por corrección de error: Consiste en el ajuste de los pesos sinápticos en función del error cometido, es decir, entre los valores esperados y los observados. Ejemplos pueden ser la *regla del aprendizaje del Perceptron* (Rosenblatt, 1958), la *regla delta* o regla del error mínimo cuadrado (Widrow y Hoff, 1960), que mejora el anterior, o la *regla delta generalizada* o algoritmo de retropropagación del error, en inglés, error backpropagation (Werbos, 1974), también conocido como regla Least-Mean-Square Error utilizado en redes multicapa feedforward.
  - Aprendizaje por refuerzo: Es un aprendizaje más lento que el de corrección por error, en el que no se indica la salida exacta que se desea que sea proporcionada por la red ante una entrada determinada. En este caso la función del agente se limita a indicar, a través de una señal de refuerzo (éxito = 1, fracaso = -1), si la salida resultante se ajusta a la deseada, cambiando los pesos en función de esto. Como ejemplo podemos citar *Adaptive Heuristic Critic* (Barto, et al., 1983).
  - Aprendizaje estocástico: Este aprendizaje se basa en la realización de cambios aleatorios de los pesos sinápticos para su posterior evaluación en función del error cometido.
- Aprendizaje no supervisado: Al contrario del aprendizaje supervisado, este tipo de aprendizaje no necesita un agente que regule los pesos sinápticos. Sólo es necesario aportar a la red datos de entrada con el fin de que esta encuentre características, correlaciones, regulaciones o categorías entre ellos para su posterior agrupación.

### 3.2.2.1 Algoritmos de aprendizaje

Los algoritmos de aprendizaje son un conjunto de reglas bien definidas establecidas para solucionar un problema de aprendizaje. Entre ellos destacamos los algoritmos de backpropagation, aprendizaje hebbiano y aprendizaje competitivo.

- Backpropagation: Como mencionamos en el apartado 3.2, el algoritmo de backpropagation supuso un punto de inflexión en la investigación de las redes neuronales. Este algoritmo de aprendizaje supervisado se basa en la repetición del ajuste de los pesos sinápticos en la red, con el fin de minimizar la diferencia entre los resultados esperados y observados (señal de error) (Rumelhart et al., 1986). El propósito de este algoritmo reside en minimizar una función de coste basada en una señal de error, de manera que la respuesta real generada por cada neurona de salida se aproxime más a la respuesta deseada. Para encontrar los pesos sinápticos que minimicen la función de coste utilizamos el método del descenso del gradiente.

Un gradiente es un vector, conformado por un conjunto de derivadas parciales, que nos indica la dirección en la que la pendiente desciende. En el caso del algoritmo de backpropagation, centramos nuestro interés en el gradiente de la función de coste. El objetivo del descenso del gradiente es encontrar el mínimo global de dicha función, ya que en ese punto es donde se produce un mejor ajuste de los datos, y por lo tanto, una mejor predicción.

- Aprendizaje Hebbiano: Es el más famoso y antiguo de todas las reglas de aprendizaje. Está basado en el postulado de Hebb (1949) “*The organization of behavior*”, cuya idea reside en el hecho de que si una neurona activa de forma reiterada a otra, se produciría un refuerzo de la conexión entre ambas o un cambio metabólico. Si la conexión entre ambas neuronas es activada de forma sincronizada, esta se verá reforzada. En caso contrario, se debilitará (Haykin, 1998).
- Aprendizaje competitivo: Como su propio nombre indica, las neuronas de salida de una red neuronal “compiten” entre ellas por ser activadas para llevar a cabo una determinada tarea. Sólo una neurona de salida o neurona por grupo puede



ser activada a la vez, alcanzando el valor de respuesta máximo. Esta competición se produce en todas las capas de la red neuronal, donde las conexiones feedback laterales generan inhibición de la neurona, mientras que las conexiones feedforward generan excitación (activación). Como resultado de este aprendizaje, la información parecida activará una misma neurona de salida, formando un clustering de datos (Haykin, 1998).

## 4. Metodología

En este punto desarrollaremos la metodología de las técnicas utilizadas para llevar a cabo nuestro estudio, así como el proceso de construcción de las tablas de mortalidad y su aplicación en una práctica de Reserving. Es por ello que dividiremos este apartado en tres subapartados, donde trataremos la metodología de la construcción de las tablas de mortalidad, el estudio de la población de estudio enfocándonos en la aplicación de las redes neuronales y un caso práctico donde aplicaremos las proyecciones obtenidas del modelo P-splines en un ejercicio de Reserving.

### 4.1. Tablas de mortalidad

Para el correcto estudio de la longevidad es necesario obtener una tabla de mortalidad, la cual es un instrumento de análisis demográfico que permite analizar la incidencia de la mortalidad de los individuos de diferentes poblaciones en un horizonte temporal determinado. Concretamente, la tabla de mortalidad pretende describir el comportamiento coyuntural del fenómeno sobre la población estudiada en un horizonte temporal determinado, a través de la simulación de la incidencia del mismo sobre una cohorte de individuos sometidos a un patrón de mortalidad por edad idéntico al observado sobre la población estudiada durante el periodo de análisis.

Se construye a partir de los supervivientes a cada edad. Una manera habitual de elaborar esta tabla es a partir de un valor inicial,  $l_0$ , para la variable biométrica  $l_x$ , la cual representa el número de individuos que alcanzan la edad  $x$ . Asumiendo la muerte como única contingencia de salida, esta variable disminuye a medida que transcurren las edades, debido a la incidencia de la mortalidad en el colectivo, representada por la probabilidad de que una persona de edad  $x$  fallezca antes de alcanzar la edad  $x + 1$ , denotada como  $q_x$ . También es común utilizar la variable biométrica  $m_x$ , definida como el cociente entre el número de personas que fallecen a la edad  $x$  y el número de personas expuestas al riesgo en el tiempo  $t$ .

La tasa central de mortalidad es definida como:

$$m_{x,t} = \frac{D_{x,t}}{E_{x,t}^C} \quad (22)$$

donde:

$E_{x,t}^C$  es la exposición central al riesgo, definida como el tiempo expuesto al riesgo de morir a la edad  $x$  en el intervalo  $(t : t + 1)$ . Para su cálculo son necesarios datos que referencien las fechas de nacimiento, entrada al colectivo de riesgo y salida del colectivo de riesgo. Se asumirán las hipótesis de distribución uniforme de fallecidos bajo un sistema demográfico cerrado (Lledó et al., 2019).

$D_{x,t}$ : Número de fallecimientos en un año en el intervalo de tiempo  $(t : t + 1)$  entre los individuos que cumplieron  $x$  años en su último cumpleaños.

## 4.2 Redes Neuronales para el estudio de la longevidad

Un parámetro clave de los modelos clásicos paramétricos es el parámetro kappa ( $\kappa_t$ ), común para todos ellos, el cual mide la tendencia de la mortalidad en el tiempo. En todos ellos hemos realizado la proyección de este parámetro asumiendo un proceso ARIMA, los cuales son recomendados en la literatura para la estimación de  $k_t$ .

Con el fin de hacer una comparación con las nuevas técnicas de machine learning, hemos desarrollado una extensión basada en las redes neuronales artificiales.

### 4.2.1. Redes Neuronales Clásicas

Las redes neuronales no predicen de una forma directa la mortalidad, sino que resumen la evolución de la tendencia de la mortalidad en factores latentes que son extrapolados con paseos aleatorios. Posteriormente la estructura de esta tendencia es reconstruida con una RNA (Hainaut, 2018).

Esta técnica consiste en la utilización de múltiples capas de funciones no lineales que son optimizadas para transformar los objetos de entrada de un modelo de regresión (en este caso utilizando tres retardos de nuestra muestra) en nuevas características que

permitirán una predicción óptima de  $\kappa_t$ .

Definimos nuestra red neuronal en dos pasos. El primero consiste en el tratamiento de nuestros datos para conseguir una correcta aplicación. Posteriormente procederemos a la obtención de una estimación mediante el ajuste de un modelo, utilizando hiperparámetros que permitan una obtención precisa de los resultados.

- **Pretratamiento de nuestros datos de estudio.**

Este proceso comenzará con la estimación del parámetro kappa de los diferentes modelos, mediante la utilización de diversas funciones que nos permitirán la obtención de los valores de los diferentes parámetros de cada modelo. Posteriormente procederemos a realizar un tratamiento de nuestros datos de estudio permitiendo su incorporación a la RNA.

Comenzaremos con la división de nuestros datos en conjuntos de entrenamiento, prueba y validación. Una combinación óptima recomendada en la literatura es la de 80%, 15% y 5% respectivamente. El conjunto de entrenamiento será utilizado, para entrenar nuestro modelo de forma supervisada, para posteriormente ponerlo a prueba con nuestros datos de prueba y por último proyectar utilizando nuestro conjunto de validación.

Para la correcta aplicación de nuestros datos a la red neuronal será necesaria su previa estandarización, permitiendo que su valor esté comprendido en un rango entre 0 y 1. Con esto conseguimos que nuestros datos se encuentren en una misma escala, evitando la inducción a error en la toma de decisiones por parte de nuestra red. Además, gracias a esta práctica conseguiremos mejorar la velocidad de tratamiento de nuestros datos.

- **Ajuste de nuestro modelo**

La forma propuesta de nuestra RNA es la de una red feedforward dada por las siguientes funciones:

$$\begin{aligned}Z_1 &= y_0(w_0 * \kappa_t + \theta_0) \\Z_2 &= y_1(w_1 * y_0 + \theta_1) \\ \hat{\kappa}_t &= y_2(w_2 * y_1 + \theta_2)\end{aligned}\tag{23}$$

Donde  $\kappa_t$ , es un vector que contiene las variables de estudio recogidas en el conjunto de entrenamiento,  $Z_1$  es un vector intermedio que contiene los valores obtenidos de la aplicación de la función de activación no lineal  $y_0$  en la combinación lineal de los inputs, con la matriz de pesos  $w_0$  y el sesgo  $\theta_0$ . Después de calcular otro vector intermedio,  $Z_2$ , calculamos la estimación de la variable  $\hat{\kappa}$  en el momento  $t$ , aplicando el último vector mencionado.

Una vez completada la “Fase de Entrenamiento” de nuestro modelo, aplicaremos el modelo entrenado para la “Fase de Prueba”. El objetivo de esta fase es comprobar que nuestro modelo realiza una buena estimación de nuestros datos y así comprobar su validez.

Completada la “Fase de Prueba”, utilizaremos muestras del vector estimado  $\hat{\kappa}_t$ , correspondiente con los últimos años de las  $\kappa$  recogidas en el conjunto de entrenamiento, con el fin de comprobar si el grado de ajuste es elevado. Para ello aplicaremos la metodología de Backtesting, la cual consiste en la utilización de datos históricos para probar el modelo predictivo. Cabe destacar que en el mundo de la inteligencia artificial no existe un modelo perfecto, sino un mejor modelo en función de los hiperparámetros utilizados.

### 4.3. Ejercicio de Reserving

El proceso de evaluación, supervisión y estimación de las obligaciones pendientes de pago dentro de una aseguradora o reaseguradora es conocido como el cálculo de reservas o, en inglés, Reserving. Una estimación precisa de estas obligaciones es muy importante para las empresas de seguros o reaseguro. Es por ello que hemos procedido a realizar esta práctica a partir de las proyecciones obtenidas por el método P-splines para el año 2019. La decisión de utilizar las proyecciones de este modelo para realizar este ejercicio se basa en que, a diferencia de los modelos clásicos, este modelo extiende el enfoque unidimensional de los mismos para poder suavizar de forma simultánea la fuerza de mortalidad en función de la edad y del tiempo, además este modelo ha sido utilizado por el Continuous Mortality Investigation (Departamento dependiente del Instituto y Facultad de Actuarios del Reino Unido), y es ampliamente aceptado a nivel

internacional.

Para poder construir las tablas de mortalidad a utilizar, han sido necesarias la mortalidad para ambos sexos y el cálculo del factor de mejora entre años, definido como:

$$\lambda_{x,t} = 1 - \left( \frac{\mu_{x,t}}{\mu_{x,t-1}} \right) \quad (24)$$

Siendo  $\lambda_{x,t}$  el factor de mejora entre periodos,  $\mu_{x,t}$  la tasa de mortalidad para una edad  $x$  en el año  $t$ , en nuestro caso el año 2019 y  $\mu_{x,t-1}$  la tasa de mortalidad para una edad  $x$  en el año  $t - 1$ . Utilizando estas dos variables generaremos las tablas de mortalidad haciendo uso de una hoja de cálculo creada para este propósito.

Para realizar este ejercicio ha sido necesario el:

- Cálculo de la mejor estimación o BEL, el cual es la media de los flujos de caja futuros ponderada por su probabilidad, teniendo en cuenta el valor temporal del dinero mediante la aplicación de la pertinente estructura temporal de tipos de interés sin riesgo.
- Capital de solvencia obligatorio de vida para el riesgo de longevidad, el cuál supone la disminución de un 20% la mortalidad de la población objeto de estudio.

## 5. Datos y software

Para la realización de este trabajo ha sido necesaria la utilización de datos históricos sobre los fallecimientos y expuestos al riesgo de una población. Estos datos han sido obtenidos de *Human Mortality Database*, la cual es una página web originada a partir del proyecto llevado a cabo, en el año 2000, por el Departamento demográfico de la Universidad de California, Berkeley (USA) y el Instituto de Investigación Demográfica Max Planck, MPIDR según sus siglas en inglés. Este proyecto alberga la información detallada de la mortalidad y datos poblacionales de 41 países diferentes, comprendida entre los años 1900 y 2018 (este periodo puede variar en función del país). Debido a la catástrofe demográfica ocasionada por la reciente pandemia COVID-19, *Human Mortality Database* ha decidido establecer una nueva fuente de datos llamada *Short Term Mortality Fluctuations*, la cual recogerá información sobre fallecimientos, actualizada semanalmente, de 38 países con el fin de proporcionar un soporte de información a las diferentes estrategias utilizadas para abordar epidemias.

Nuestra población objeto de estudio es la española, de la cual utilizaremos información sobre las variables biométricas de fallecimiento ( $D_{x,t}$ ) y exposición al riesgo ( $E_{x,t}$ ) por edad y género, de personas de 0 a 100 años, comprendidas entre los años 1960 y 2018.

La dinámica de la longevidad española presenta dos fenómenos relevantes. El primero se caracteriza por el continuado desplazamiento de fallecimientos a edades más avanzadas. La edad modal presenta un desplazamiento hacia la derecha, además de leves incrementos en el tiempo del número de fallecidos a dicha edad (ver tabla 2). Este fenómeno podría explicarse por un desplazamiento en la edad promedio en que inician las enfermedades crónicas. El segundo fenómeno es conocido como el efecto de la “comprensión de la mortalidad”, el cual indica que la mejora de la mortalidad se debe a (i) la desaparición de las muertes agudas, neonatales e infantiles y (ii) sustitución de la mortalidad por enfermedades agudas por la mortalidad por enfermedades crónicas, donde se se está produciendo un declive de los padecimientos crónicos.

**Tabla 2**

Edad modal de fallecimiento para ambos géneros en la población española durante el periodo entre 1960-2018. Edades entre 0 y 100 años.

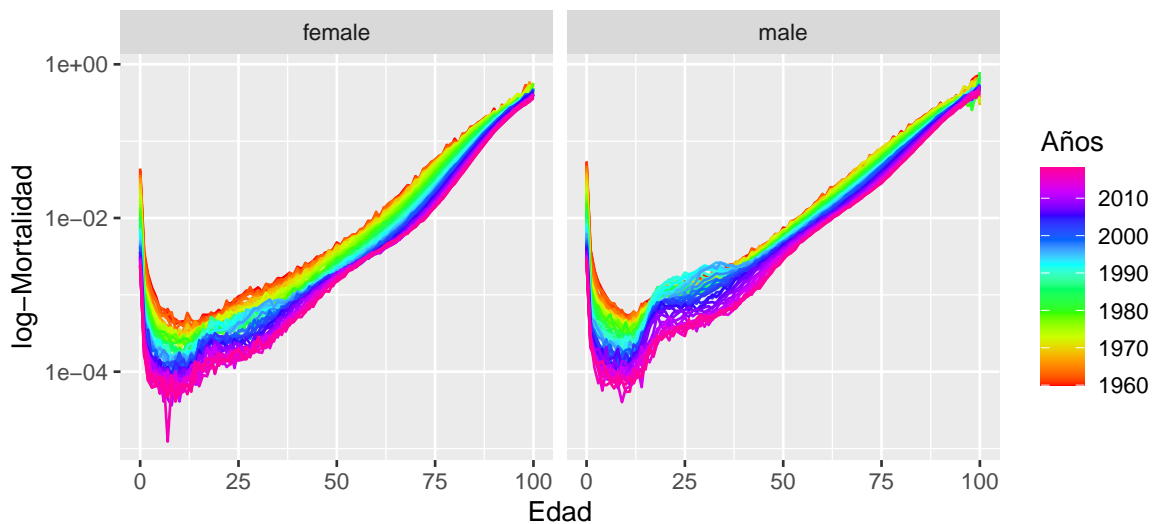
Género/Periodo	1960-1970	1971-1980	1981-1990	1991-2000	2001-2009	2010-2018
Hombres	73	75	78	82	81	85
Mujeres	79	83	83	86	87	89

Elaboración propia a partir de datos de Human Mortality Database.

Como podemos observar en la Tabla 2, la edad modal de fallecimiento se ha visto incrementada durante las últimas 6 décadas, tanto para el género masculino como para el femenino, teniendo este último una edad modal superior, lo que indicaría una mayor esperanza de vida con respecto al género masculino.

**Figura 6**

Tasa de mortalidad logarítmica para ambos géneros en la población española durante el periodo entre 1960-2018. Edades entre 0 y 100 años.



Elaboración propia a partir de datos de Mortality database.

Los valores presentados representan el tanto bruto de mortalidad en escala logarítmica. Debido a la utilización de esta escala se consigue una sucesión de puntos bastante regular y una curva sin muchos puntos de inflexión.

El proyecto ha sido realizado en su gran mayoría con el software estadístico libre R Markdown utilizando los packages “neuralnet”, “sarima” y “forecast” entre otros.



## 6. Resultados

### 6.1 Ajuste y forecasting In-sample

Los siguientes subapartados reflejarán los resultados obtenidos para el ajuste y forecasting del parámetro kappa de los modelos clásicos analizados, utilizando un proceso ARIMA y un modelo de RNA. El objetivo de este apartado será la comparación de la capacidad predictiva entre ambos métodos. Para llevar a cabo el ajuste de los modelos clásicos, hemos utilizado las funciones contenidas en el archivo “fitModels.R”.

#### 6.1.1 Resultados para el modelo Lee-Carter

El modelo de Lee-Carter es ajustado asumiendo una distribución Poisson, por lo que la función de liga es logarítmica. El parámetro kappa de Lee-Carter está representado en un horizonte temporal de 58 años, comprendidos entre 1960 y 2018, para personas de edades entre 1 y 100 años (ver figura 7).

##### 6.1.1.1 Proceso ARIMA

En primera instancia, hemos seguido un proceso ARIMA(p,d,q) para la estimación nuestro parámetro. Previamente hemos establecido una diferencia para ambos set de datos, hombres y mujeres, con el fin de eliminar la tendencia de la serie, haciendo que nuestros datos rechacen la  $H_0$  del test de Dickey-Fuller, la existencia de una raíz unitaria. Una vez eliminado el componente tendencia de nuestros datos, utilizamos la función “auto.arima”, la cuál nos proporcionará los procesos de nuestro modelo. Puede verse representado en la figura 8.

Género	ARIMA(p,d,q)
Hombres	(1,1,2)
Mujeres	(1,1,0)

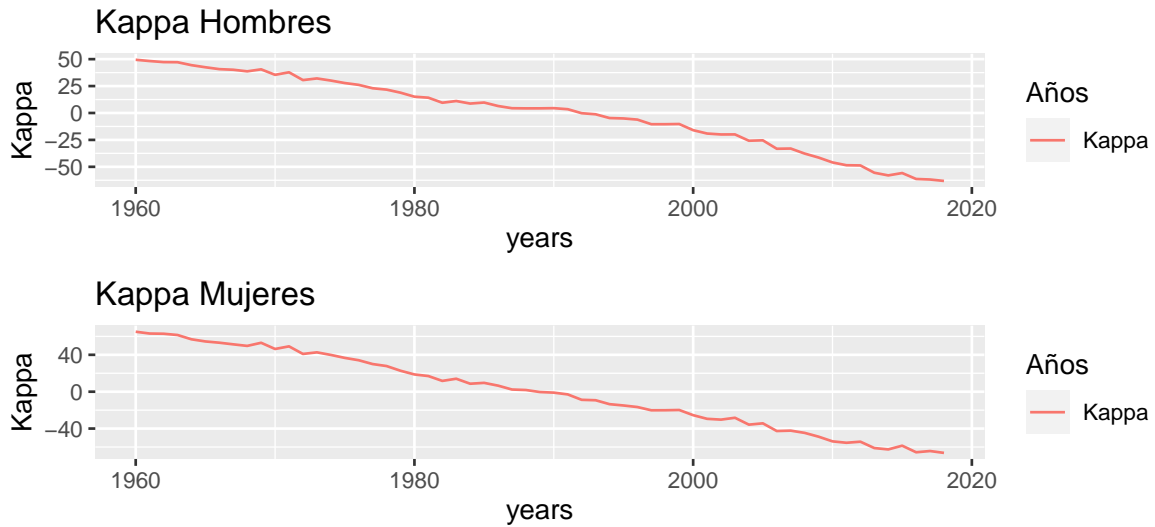
Tabla 3: Procesos ARIMA para el modelo de Lee-Carter

Para el estudio de la capacidad predictiva del proceso ARIMA hemos creído conveniente realizar predicciones a uno, cinco y diez años. Correspondiendo con los años 2010, 2014

y 2018 respectivamente. De esta forma podremos comprobar el comportamiento de nuestro modelo a corto y largo plazo.

**Figura 7**

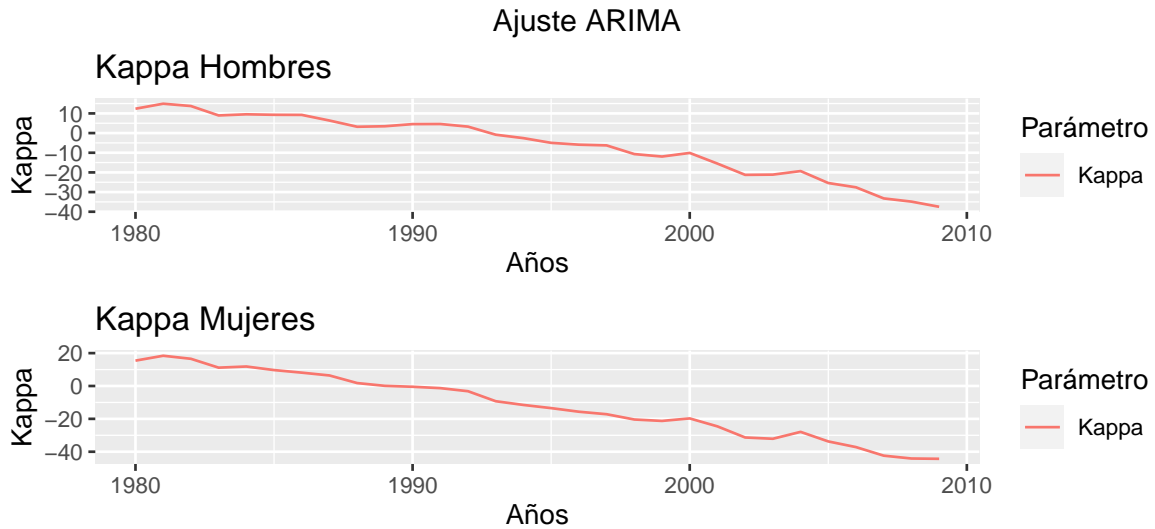
*Parámetro Kappa del modelo de Lee-Carter para los años 1960-2018.*



Elaboración propia.

**Figura 8**

*Parámetro Kappa del modelo de Lee-Carter ajustado con proceso ARIMA para los años 1980-2009.*



Elaboración propia.

Hombres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-46	-49	-49	-56	-58	-56	-61	-62	-63
Kappa ARIMA	-42	-43	-43	-43	-43	-42	-42	-42	-41

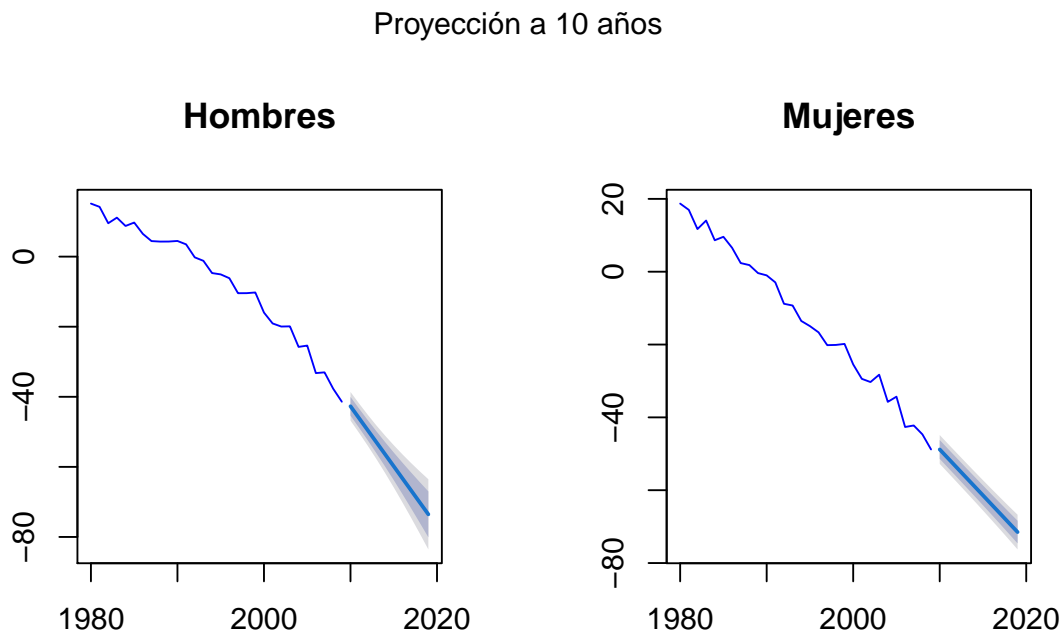
Mujeres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-54	-55	-54	-61	-63	-59	-66	-64	-66
Kappa ARIMA	-50	-50	-49	-49	-49	-48	-48	-48	-48

Tabla 4: Valores observados y proyectados del parámetro kappa aplicando el proceso ARIMA

Como podemos observar en la Tabla 4, el proceso ARIMA no predice de forma exacta el parámetro Kappa para el modelo de Lee-Carter, obteniendo unas predicciones más acertadas durante los tres primeros años de proyección, mientras que para años posteriores, este nivel va disminuyendo. Además, las proyecciones para el género femenino son más acertadas que para el género masculino. Esta predicción se puede ver reglejada en la figura 9.

**Figura 9**

*Forecasting parámetro Kappa del modelo de Lee-Carter ajustado con proceso ARIMA para los años 2009-2018.*



Elaboración propia.

### 6.1.1.2 Modelo de RNA

Nuestro modelo de RNA para la predicción del parámetro kappa se ha construido a partir de la estimación de una serie de hiperparámetros. Como se comentó en el marco teórico, la elección de estos hiperparámetros se han basado en un proceso de prueba y error. Después de la realización de una serie de simulaciones con distintos hiperparámetros, hemos llegado a la conclusión de que los más acertados serían los reflejados en la Tabla 5.

Género	Capas ocultas	Neuronas	Umbral	Algoritmo	Función de activación
Hombres	1	5	0.01	rprop+ <sup>1</sup>	Softplus <sup>2</sup>
Mujeres	1	5	0.01	rprop+	Softplus

Tabla 5: Hiperparámetros seleccionados para entrenar nuestro modelo de RNA

El umbral es utilizado como un criterio de parada, el cual impone un límite que no debe sobrepasarse antes de propagarse a otra neurona. Un umbral demasiado pequeño podría provocar un sobre ajuste de nuestros datos, mientras que al contrario, podría hacer que el entrenamiento de nuestra red no fuera suficiente.

Una vez entrenado el modelo, lo aplicaremos a nuestro set de entrenamiento, comprendido entre los años 2000 y 2009. Debido a que para testear el modelo hemos utilizado como inputs tres muestras de la propia muestra, las cuales contienen un retardo cada una, el modelo nos ha arrojado predicciones para el rango de años de 2003 a 2010. Los resultados obtenidos presentan un nivel de ajuste considerablemente aceptable (ver Tabla 6).

Años	Hombres		Mujeres	
	Real	RNA	Real	RNA
2003	-19.07	-21.14	-29.42	-29.01
2004	-19.95	-19.84	-30.29	-31.17
2005	-19.89	-18.36	-28.28	-30.81
2006	-25.76	-25.60	-35.74	-35.55
2007	-25.38	-25.55	-34.31	-35.04
2008	-33.24	-33.36	-42.64	-41.88
2009	-33.02	-33.60	-42.22	-42.14
2010	-37.69	-36.16	-44.66	-45.43

Tabla 6: Valores observados y esperados del conjunto de testeo aplicando nuestro modelo de RNA

<sup>1</sup>Retropropagación resiliente. Técnica de optimización basada en gradientes.

<sup>2</sup>Forma suavizada de la función de activación ReLu. Produce resultados entre  $(0, \infty)$ .

Una vez comprobada la eficiencia de nuestro modelo, procedemos a aplicarlo a nuestro set de datos de validación, con el que proyectaremos el parámetro Kappa, al igual que con el proceso ARIMA, a uno, cinco y diez años.

Hombres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-46	-49	-49	-56	-58	-56	-61	-62	-63
Kappa ARN	-42	-44	-44	-51	-53	-51	-57	-57	-59

Mujeres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-54	-55	-54	-61	-63	-59	-66	-64	-66
Kappa ARN	-50	-55	-57	-55	-62	-60	-67	-65	-67

Tabla 7: Valores observados y esperados de las predicciones obtenidas con el modelo de RNA

Como podemos comprobar en la Tabla 7, los resultados de la proyección de nuestro modelo de RNA se ajustan bastante bien a los datos originales del parámetro Kappa para ambos sexos. Podemos observar que en algunos casos existe un ligero sobre ajuste, algo común cuando se trabaja con este tipo de técnicas.

Años	RNA		ARIMA	
	Hombres RMSE	Mujeres RMSE	Hombres RMSE	Mujeres RMSE
2010	4.02	1.25	4	5
2014	4.21	1.19	9.81	9.01
2018	4.38	1.14	14.61	12.51

Tabla 8: Comparación de los resultados obtenidos de la aplicación de nuestro modelo de RNA y el proceso ARIMA

En la Tabla 8 podemos observar que la capacidad predictiva de la RNA con respecto al proceso ARIMA es notablemente superior. Este hecho es principalmente visible para las predicciones a largo plazo de ambos sexos. En el caso del parámetro Kappa para el género femenino, hay una visible diferencia durante todo el periodo predictivo.

### 6.1.2 Resultados para el modelo Renshaw-Haberman

El modelo de Renshaw-Haberman es una extensión del modelo de Lee-Carter por lo que de igual forma, se distribuye según una poisson y su función de liga es logarítmica.

El parámetro kappa está representado en un horizonte temporal de 58 años, comprendidos entre 1960 y 2018, para personas de edades entre 1 y 100 años (ver figura 10).

### 6.1.2.1 Proceso ARIMA

En primer lugar hemos establecido dos diferencias para ambos set de datos, hombres y mujeres, con el fin de eliminar la tendencia de la serie. Utilizamos la función “auto.arima”, la cuál nos proporcionará los procesos de nuestro modelo. Puede verse representado en la figura 11.

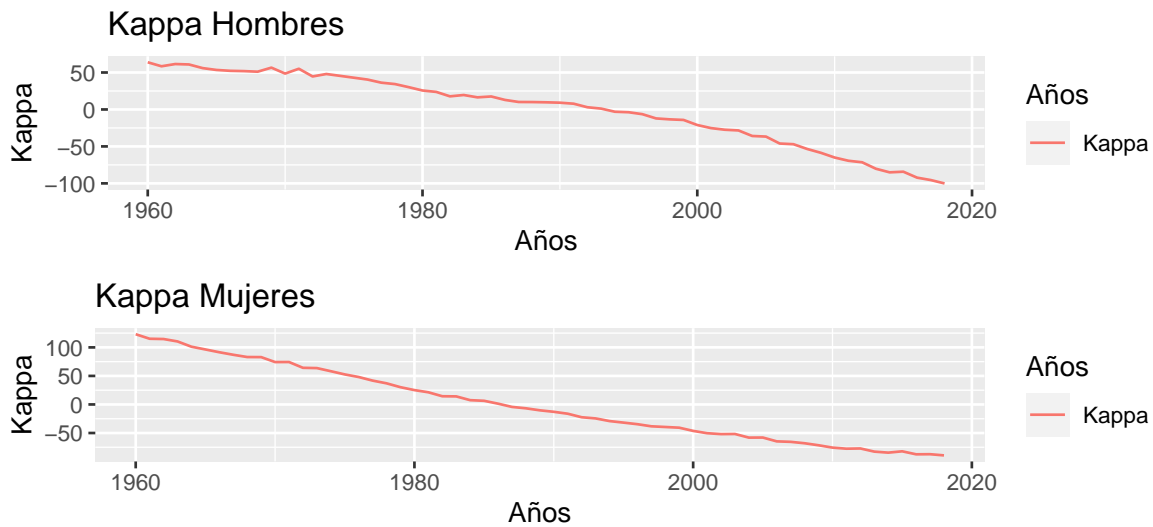
Género	ARIMA(p,d,q)
Hombres	(1,1,1)
Mujeres	(1,1,1)

Tabla 9: Procesos ARIMA para el modelo de Renshaw-Haberman

Para el estudio de la capacidad predictiva del proceso ARIMA realizaremos predicciones a uno, cinco y diez años. Correspondiendo con los años 2010, 2014 y 2018 respectivamente. De esta forma podremos comprobar el comportamiento de nuestro modelo a corto y largo plazo.

Figura 10

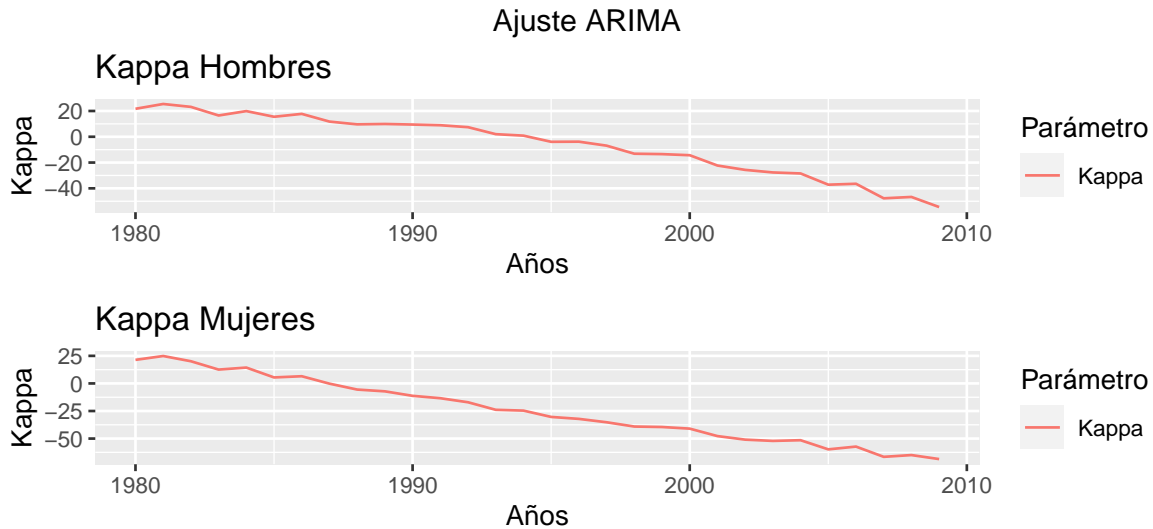
Parámetro Kappa del modelo de Renshaw-Haberman para los años 1960-2018.



Elaboración propia.

**Figura 11**

Parámetro Kappa del modelo de Renshaw-Haberman ajustado con proceso ARIMA para los años 1980-2009.



Elaboración propia.

Hombres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-65	-69	-71	-80	-85	-84	-92	-96	-100
Kappa ARIMA	-59	-59	-59	-58	-58	-58	-58	-57	-57

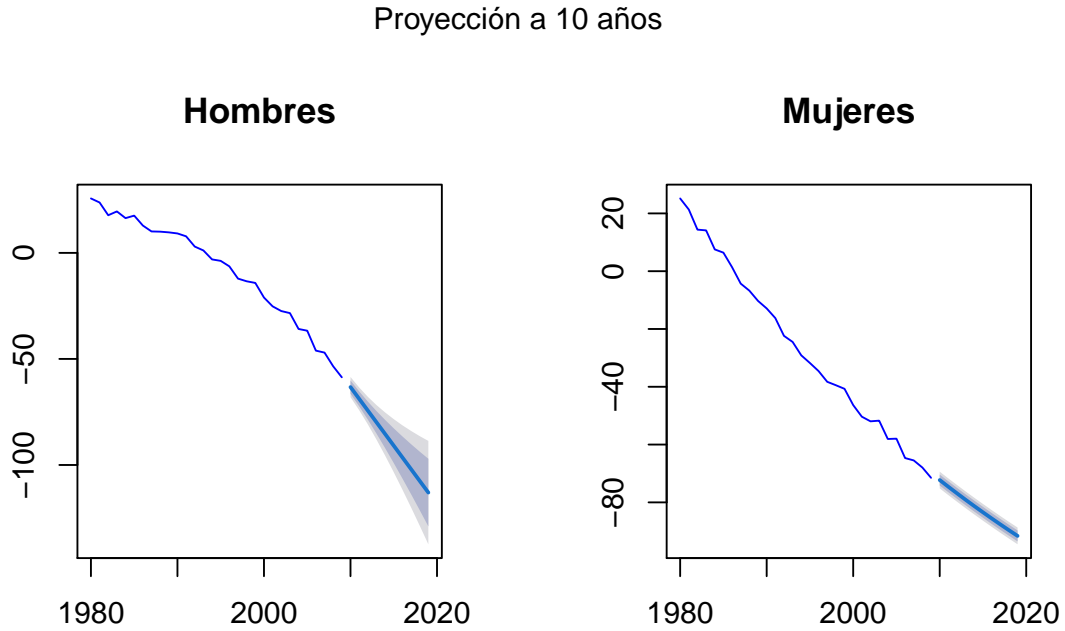
Mujeres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-76	-77	-77	-83	-84	-82	-87	-87	-89
Kappa ARIMA	-72	-72	-71	-71	-71	-71	-70	-70	-70

Tabla 10: Valores observados y proyectados del parámetro kappa aplicando el proceso ARIMA

Como podemos observar en la Tabla 10, el proceso ARIMA no predice de forma exacta el parámetro Kappa para el modelo de Renshaw-Haberman, obteniendo unas predicciones más acertadas, en el caso del género femenino, durante los tres primeros años de proyección, y en el caso del género masculino, sólo para el primer año. En general, en este caso, las proyecciones para el género femenino también son más acertadas que para el género masculino. Estas predicciones pueden verse reflejadas en la figura 12.

**Figura 12**

*Forecasting parámetro Kappa del modelo de Renshaw-Haberman ajustado con un proceso ARIMA para los años 2009-2018.*



Elaboración propia.

### 6.1.2.2 Modelo de RNA

Nuestro modelo de RNA para la predicción del parámetro kappa se ha construido a partir de la estimación de una serie de hiperparámetros. Después de la realización de una serie de simulaciones con distintos hiperparámetros, hemos llegado a la conclusión de que los más acertados serían los reflejados en la Tabla 11.

Género	Capas ocultas	Neuronas	Umbral	Algoritmo	Función de activación
Hombres	1	10	0.01	rprop+	Softplus
Mujeres	1	5	0.01	rprop+	Softplus

*Tabla 11: Hiperparámetros seleccionados para entrenar nuestro modelo de RNA*

Una vez entrenado el modelo, lo aplicaremos a nuestro set de entrenamiento, comprendido entre los años 2000 y 2009. Debido a que para testear el modelo hemos utilizado como



inputs tres muestras de la propia muestra, las cuales contienen un retardo cada una, el modelo nos ha arrojado predicciones para el rango de años de 2003 a 2010. Los resultados obtenidos presentan un nivel de ajuste considerablemente aceptable (ver Tabla 12).

Años	Hombres		Mujeres	
	Real	RNA	Real	RNA
2003	-25.20	-24.44	-50.35	-52.50
2004	-27.43	-28.33	-51.95	-52.95
2005	-28.39	-30.39	-51.72	-51.60
2006	-35.88	-36.03	-58.07	-58.44
2007	-36.68	-35.94	-57.95	-58.68
2008	-46.09	-44.91	-64.66	-64.62
2009	-47.02	-44.93	-65.48	-65.83
2010	-53.47	-52.72	-67.96	-66.78

Tabla 12: Valores observados y esperados del conjunto de testeo aplicando nuestro modelo de RNA

Una vez comprobada la eficiencia de nuestro modelo, procedemos a aplicarlo a nuestros set de datos de validación, con el que proyectaremos el parámetro Kappa, al igual que con el proceso ARIMA, a uno, cinco y diez años.

Hombres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-65	-69	-71	-80	-85	-84	-92	-96	-100
Kappa ARN	-60	-67	-71	-73	-82	-86	-93	-97	-97

Mujeres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-76	-77	-77	-83	-84	-82	-87	-87	-89
Kappa ARN	-73	-74	-74	-79	-81	-79	-83	-83	-85

Tabla 13: Valores observados y esperados de las predicciones obtenidas con el modelo de RNA

Como podemos comprobar en la Tabla 13, los resultados de la proyección de nuestro modelo de RNA se ajustan bastante bien a los datos originales del parámetro Kappa para ambos sexos.

Años	RNA		ARIMA	
	Hombres RMSE	Mujeres RMSE	Hombres RMSE	Mujeres RMSE
2010	1.71	1.02	6	4
2014	1.59	1.34	17.28	8.7
2018	1.38	1.81	27.27	12.49

Tabla 14: Comparación de los resultados obtenidos de la aplicación de nuestro modelo de RNA y el proceso ARIMA

En la Tabla 14 podemos observar que la capacidad predictiva de la RNA con respecto al proceso ARIMA es notablemente superior. Este hecho es más contrastado para las predicciones a largo plazo de ambos sexos. En el caso de las predicciones del parámetro Kappa para el género masculino, hay una diferencia importante entre ambos modelos.

### 6.1.3 Resultados para el modelo Edad-Periodo-Cohorte

El modelo Edad-Periodo-Cohorte, como su propio nombre indica, se considera un modelo para la edad, año de nacimiento y efectos de fallecimiento sobre una generación, donde dichos efectos influyen en las tasas de mortalidad de manera independiente. El parámetro kappa está representado en un horizonte temporal de 58 años, comprendidos entre 1960 y 2018, para personas de edades entre 1 y 100 años (ver figura 13).

#### 6.1.3.1 Proceso ARIMA

En primer lugar hemos establecido una diferencia para ambos set de datos, hombres y mujeres, con el fin de eliminar la tendencia de la serie. Utilizamos la función “auto.arima”, la cuál nos proporcionará los procesos de nuestro modelo. Puede verse representado en la figura 14.

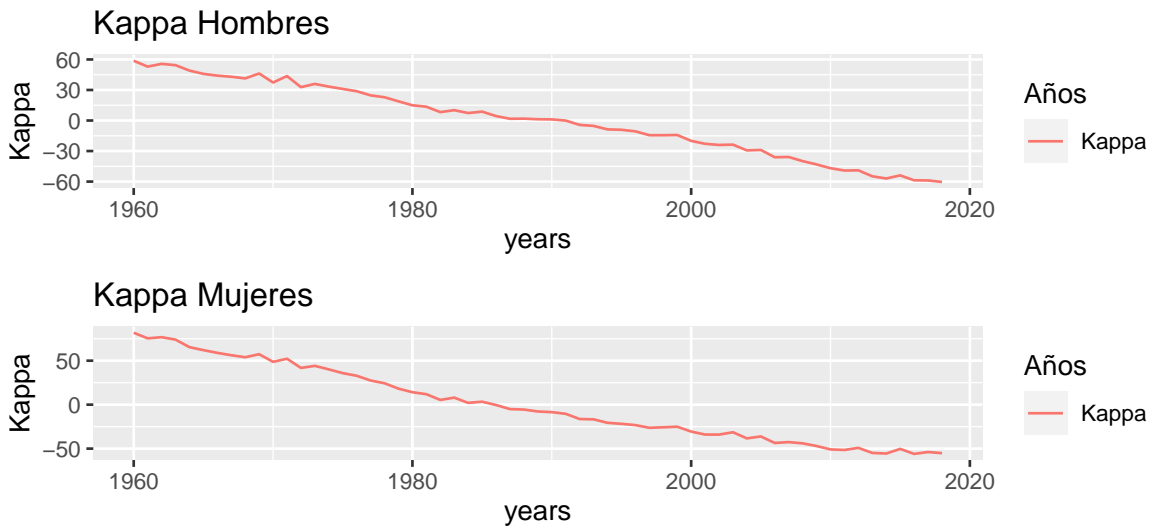
Género	ARIMA(p,d,q)
Hombres	(1,1,0)
Mujeres	(1,2,1)

Tabla 15: Procesos ARIMA para el modelo APC

Para el estudio de la capacidad predictiva del proceso ARIMA realizaremos predicciones a uno, cinco y diez años. Correspondiendo con los años 2010, 2014 y 2018 respectivamente. De esta forma podremos comprobar el comportamiento de nuestro modelo a corto y largo plazo.

**Figura 13**

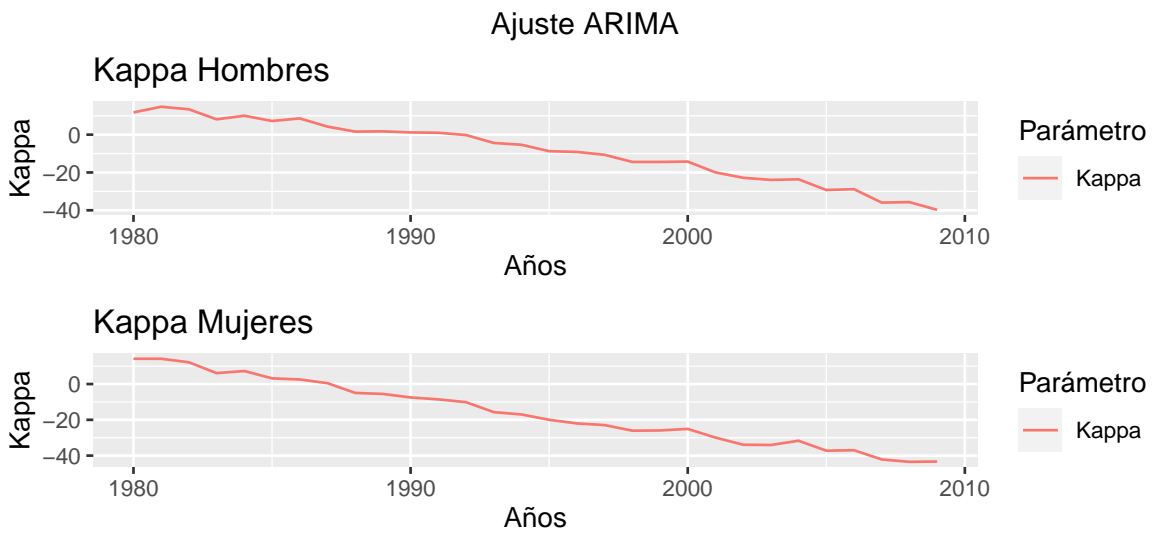
*Parámetro Kappa del modelo de APC para los años 1960-2018.*



Elaboración propia.

**Figura 14**

*Parámetro Kappa del modelo de APC ajustado con proceso ARIMA para los años 1980-2009.*



Elaboración propia.

Hombres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-47	-49	-49	-55	-57	-54	-59	-59	-60
Kappa ARIMA	-43	-43	-43	-42	-42	-42	-42	-42	-42

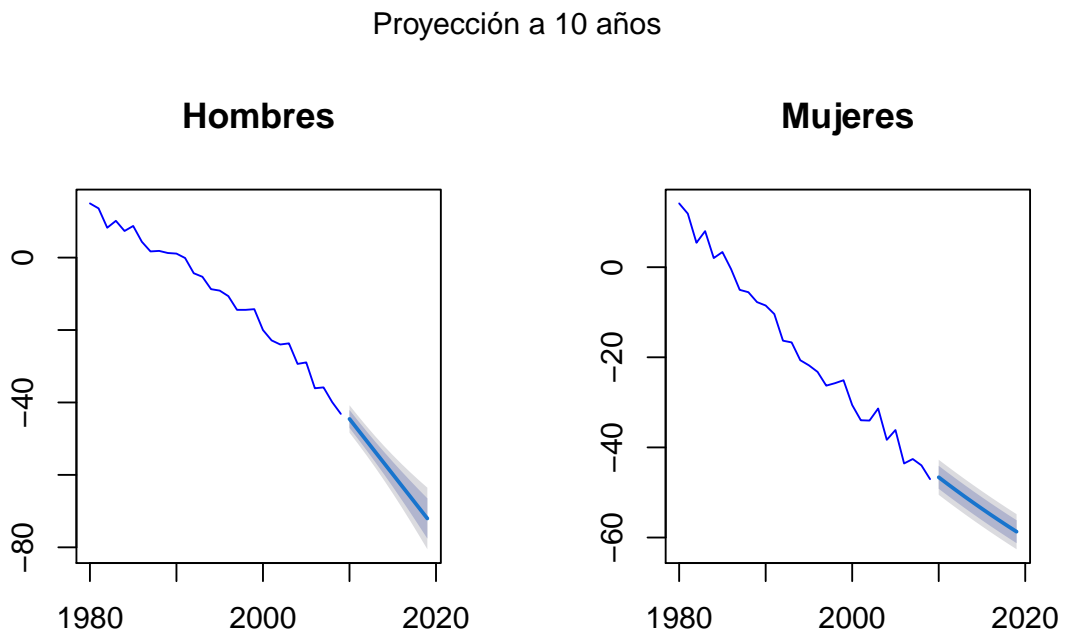
Mujeres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-51	-51	-49	-55	-56	-50	-56	-54	-55
Kappa ARIMA	-47	-47	-47	-47	-47	-47	-47	-47	-47

Tabla 16: Valores observados y proyectados del parámetro kappa aplicando el proceso ARIMA

Como podemos observar en la Tabla 16, el proceso ARIMA no predice de forma precisa el parámetro Kappa para el modelo de Edad-Periodo-Cohorte, llegando a ajustar mejor a corto plazo. Estas predicciones pueden verse reflejadas en la figura 15.

**Figura 15**

Forecasting parámetro Kappa del modelo de APC ajustado con proceso ARIMA para los años 2009-2018.



Elaboración propia.

### 6.1.3.2 Modelo de RNA

Después de la realización de una serie de simulaciones con distintos hiperparámetros, hemos llegado a la conclusión de que los más acertados para nuestro modelo serían los reflejados en la Tabla 17.

Género	Capas ocultas	Neuronas	Umbral	Algoritmo	Función de activación
Hombres	1	10	0.01	rprop+	Softplus
Mujeres	1	5	0.05	rprop+	Softplus

Tabla 17: Hiperparámetros seleccionados para entrenar nuestro modelo de RNA

Elaboración propia

Una vez entrenado el modelo, lo aplicaremos a nuestro set de testeo, comprendido entre los años 2000 y 2009. Debido a que para testear el modelo hemos utilizado como inputs tres muestras de la propia muestra, las cuales contienen un retardo cada una, el modelo nos ha arrojado predicciones para el rango de años de 2003 a 2010. No encontramos grandes diferencias entre los resultados observados y esperados (ver Tabla 18).

	Hombres		Mujeres	
Años	Real	RNA	Real	RNA
2003	-22.85	-24.10	-33.97	-32.68
2004	-23.99	-24.31	-34.04	-34.06
2005	-23.68	-23.48	-31.36	-33.55
2006	-29.35	-29.37	-38.32	-38.08
2007	-28.93	-29.75	-36.13	-35.23
2008	-36.07	-36.65	-43.55	-42.42
2009	-35.83	-37.13	-42.57	-40.45
2010	-39.91	-40.01	-43.97	-44.09

Tabla 18: Valores observados y esperados del conjunto de testeo aplicando nuestro modelo de RNA

Elaboración propia

Una vez comprobada la eficiencia de nuestro modelo, procedemos a aplicarlo a nuestros set de datos de validación, con el que proyectaremos el parámetro Kappa a uno, cinco y diez años.

Hombres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-47	-49	-49	-55	-57	-54	-59	-59	-60
Kappa RNA	-47	-49	-49	-55	-58	-55	-60	-60	-62
Mujeres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-51	-51	-49	-55	-56	-50	-56	-54	-55
Kappa RNA	-50	-51	-49	-54	-55	-50	-55	-53	-54

Tabla 19: Valores observados y esperados de las predicciones obtenidas con el modelo de RNA  
Elaboración propia

Como podemos comprobar en la Tabla 19, los resultados de la proyección de nuestro modelo de RNA se ajustan bastante bien a los datos originales del parámetro Kappa para ambos sexos. En el caso del género masculino podemos observar que en los últimos años de estudio se produce un sobre ajuste de nuestros datos.

Años	RNA		ARIMA	
	Hombres RMSE	Mujeres RMSE	Hombres RMSE	Mujeres RMSE
2010	0.42	0.41	4	4
2014	0.49	0.6	9.81	6
2018	1.16	0.78	13	6.53

Tabla 20: Comparación de los resultados obtenidos de la aplicación de nuestro modelo de RNA y el proceso ARIMA

Al igual que en los modelos estudiados anteriormente, en la Tabla 20 podemos observar que la capacidad de predicción de la RNA es superior a la del proceso ARIMA.

#### 6.1.4 Resultados para el modelo CBD

El modelo CBD es una de las alternativas más populares al modelo de Lee-Carter. Principalmente se centra en edades avanzadas, es por eso que en este caso hemos decidido reducir el rango de edad a las comprendidas entre 40 y 100 años. El horizonte temporal sobre el que se representa el parámetro kappa sigue siendo el mismo, 58 años (ver figura 16).

##### 6.1.4.1 Proceso ARIMA

En primer lugar hemos establecido una diferencia para ambos set de datos, hombres y mujeres, con el fin de eliminar la tendencia de la serie. Utilizamos la función “auto.arima”,

la cuál nos proporcionará los procesos de nuestro modelo. Puede verse representado en la figura 17.

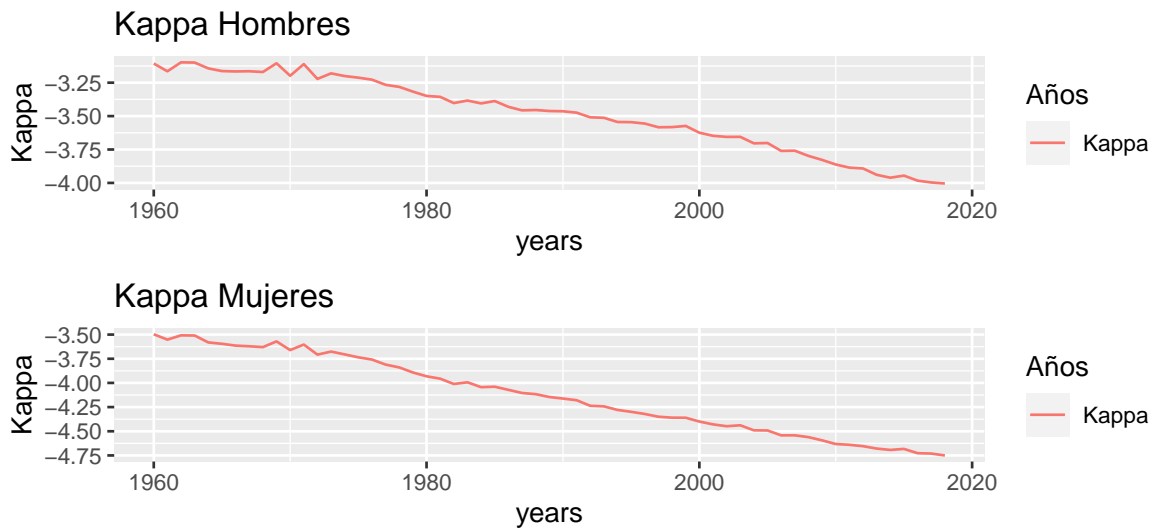
Género	ARIMA(p,d,q)
Hombres	(1,1,0)
Mujeres	(1,1,0)

Tabla 21: Procesos ARIMA para el modelo APC

Al igual que en los modelos anteriores, para el estudio de la capacidad predictiva del proceso ARIMA realizaremos predicciones a uno, cinco y diez años. Correspondiendo con los años 2010, 2014 y 2018 respectivamente. De esta forma podremos comprobar el comportamiento de nuestro modelo a corto y largo plazo.

Figura 16

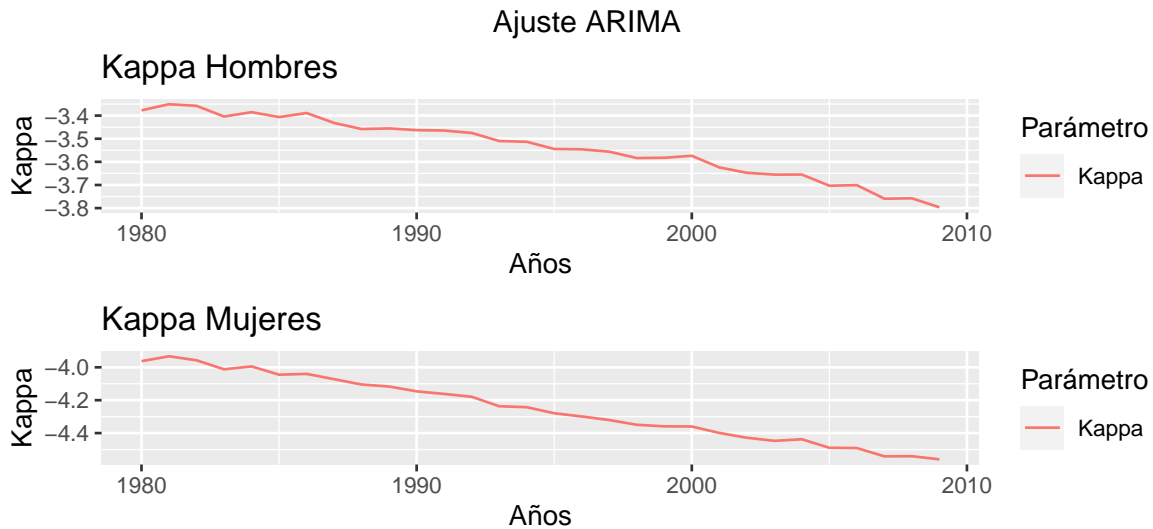
Parámetro Kappa del modelo de CBD para los años 1960-2018.



Elaboración propia.

**Figura 17**

Parámetro Kappa del modelo de CBD ajustado con proceso ARIMA para los años 1980-2009.



Elaboración propia.

Hombres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-4	-4	-4	-4	-4	-4	-4	-4	-4
Kappa ARIMA	-4	-4	-4	-4	-4	-4	-4	-4	-4

Mujeres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-5	-5	-5	-5	-5	-5	-5	-5	-5
Kappa ARIMA	-5	-5	-5	-5	-5	-5	-5	-5	-5

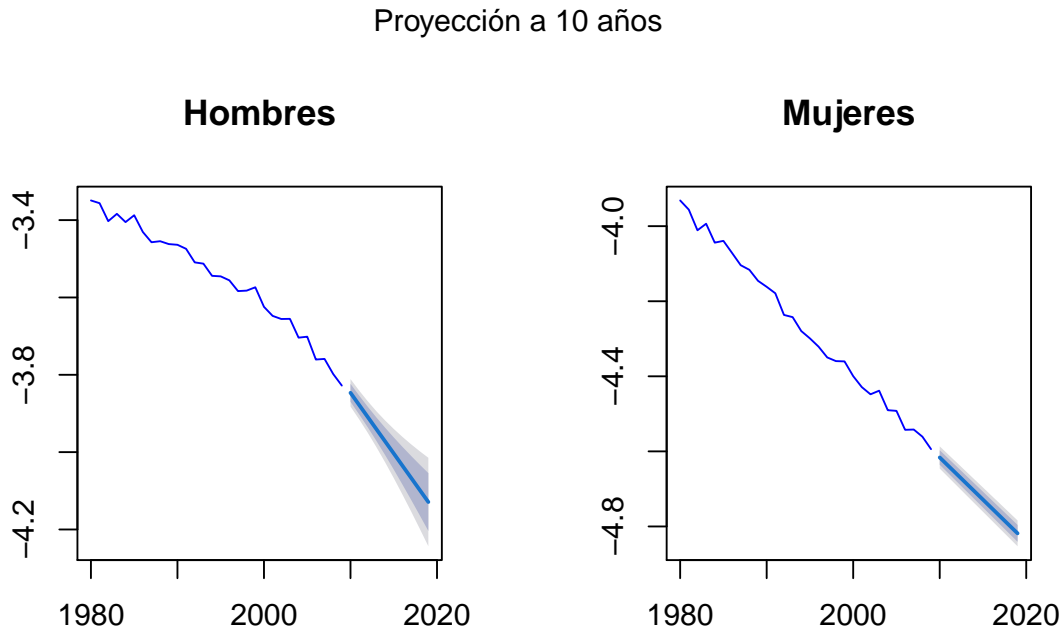
Tabla 22: Valores observados y proyectados del parámetro kappa aplicando el proceso ARIMA

Como podemos observar en la Tabla 22, la capacidad de predicción es muy buena, arrojando resultados muy similares a los esperados. Cabe destacar que los valores están redondeados al número entero superior más próximo, por lo que aunque ambos resultados, observados y esperados, son muy similares, no llegan a ser idénticos.



**Figura 18**

*Forecasting parámetro Kappa del modelo de CBD ajustado con un proceso ARIMA para los años 2009-2018.*



Elaboración propia.

#### 6.1.4.2 Modelo de RNA

Nuestro modelo de RNA para la predicción del parámetro kappa se ha construido a partir de la estimación de una serie de hiperparámetros. Después de la realización de una serie de simulaciones con distintos hiperparámetros, hemos llegado a la conclusión de que los más acertados serían los reflejados en la Tabla 23.

Género	Capas ocultas	Neuronas	Umbral	Algoritmo	Función de activación
Hombres	1	4	0.1	rprop+	Softplus
Mujeres	1	5	0.01	rprop+	Softplus

*Tabla 23: Hiperparámetros seleccionados para entrenar nuestro modelo de RNA*

Una vez entrenado el modelo, lo aplicaremos a nuestro set de testeo, comprendido entre los años 2000 y 2009. Debido a que para testear el modelo hemos utilizado como inputs

tres muestras de la propia muestra, las cuales contienen un retardo cada una, el modelo nos ha arrojado predicciones para el rango de años de 2003 a 2010. No encontramos grandes diferencias entre los resultados observados y esperados.

Años	Hombres		Mujeres	
	Real	RNA	Real	RNA
2003	-3.64	-3.62	-4.42	-4.44
2004	-3.65	-3.65	-4.44	-4.46
2005	-3.65	-3.67	-4.43	-4.45
2006	-3.70	-3.70	-4.49	-4.48
2007	-3.70	-3.69	-4.49	-4.52
2008	-3.76	-3.75	-4.54	-4.54
2009	-3.75	-3.74	-4.54	-4.57
2010	-3.79	-3.80	-4.56	-4.56

*Tabla 24: Valores observados y esperados del conjunto de testeo aplicando nuestro modelo de RNA*

Como podemos comprobar en la Tabla 24, los valores arrojados por la RNA para ambos sexos son muy similares a los valores originales, por lo que podemos llegar a la conclusión de que el modelo ha sido entrenado correctamente.

Hombres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-4	-4	-4	-4	-4	-4	-4	-4	-4
Kappa RNA	-4	-4	-4	-4	-4	-4	-4	-4	-4

Mujeres	2010	2011	2012	2013	2014	2015	2016	2017	2018
Kappa original	-5	-5	-5	-5	-5	-5	-5	-5	-5
Kappa RNA	-5	-5	-5	-5	-5	-5	-5	-5	-5

*Tabla 25: Valores observados y esperados de las predicciones obtenidas con el modelo de RNA*

Al igual que en los resultados obtenidos por el proceso ARIMA, los resultados arrojados por la RNA son muy similares a los valores esperados del parámetro kappa (ver Tabla 25). De igual modo, estos valores están redondeados al entero superior, por lo que no llegan a ser totalmente idénticos. Debido a la similitud de los resultados en ambos modelos, a continuación podremos valorar cuál de ellos tiene una mejor capacidad predictiva a partir del análisis del Error Medio Cuadrático (RMSE).

	RNA		ARIMA	
Años	Hombres RMSE	Mujeres RMSE	Hombres RMSE	Mujeres RMSE
2010	0.03	0.009	0.035	0.04
2014	0.03	0.01	0.09	0.07
2018	0.04	0.011	0.13	0.1

Tabla 26: Comparación de los resultados obtenidos de la aplicación de nuestro modelo de RNA y el proceso ARIMA

En la Tabla 26 podemos observar que la capacidad de predicción de la RNA es ligeramente superior a la del proceso ARIMA durante todo el periodo, pero este hecho es más pronunciado a largo plazo. De igual modo, podemos destacar la notable capacidad de predicción de ambos modelos para el rango de edad seleccionado.

RNA						
Modelo	Hombres			Mujeres		
	1 año	5 años	10 años	1 año	5 años	10 años
LC(RMSE)	4.02	4.21	4.38	1.25	1.19	1.14
LC(RMSPE)	10%	9.67%	28.8%	2.39%	2.16%	1.94%
RH(RMSE)	1.71	1.59	1.38	1.02	1.34	1.81
RH(RMSPE)	8.9%	8.4%	7.6%	3.7%	4%	4.5%
APC(RMSE)	0.42	0.49	1.16	0.41	0.6	0.78
APC(RMSPE)	0.98%	0.96%	1.95%	0.82	1.16%	1.47%
CBD(RMSE)	0.03	0.03	0.04	0.009	0.01	0.011
CBD(RMSPE)	0.78%	0.87%	1%	0.2%	0.22%	0.24%

ARIMA						
Modelo	Hombres			Mujeres		
	1 año	5 años	10 años	1 año	5 años	10 años
LC(RMSE)	4	9.81	14.61	5	9.01	12.51
LC(RMSPE)	9.52%	22.85	34.82	10.2%	18.35%	25.87%
RH(RMSE)	6	17.28	27.27	4	8.7	12.49
RH(RMSPE)	10%	29.6%	47.4%	5.5%	12.2%	17.7%
APC(RMSE)	4	9.81	13	4	6	6.53
APC(RMSPE)	9.3%	23.3%	31%	8.5%	12.8%	12.9%
CBD(RMSE)	0.035	0.09	0.13	0.04	0.07	0.1
CBD(RMSPE)	0.94%	2.44%	3.46%	0.86%	1.62%	2.39%

Tabla 27: RMSE y RMSPE para las proyecciones a 1, 5 y 10 años del parámetro kappa de los modelos clásicos analizados, utilizando el modelo de Red Neuronal Artificial y el proceso ARIMA

## 6.2. Ejercicio de Reserving

El proceso de evaluación, supervisión y estimación de las obligaciones pendientes de pago dentro de una aseguradora o reaseguradora es conocido como el cálculo de reservas o, en inglés, Reserving. Para llevar a cabo este ejercicio hemos decidido utilizar las

proyecciones de mortalidad, para el año 2019, obtenidas aplicando el modelo P-splines, por las razones explicadas en el apartado 4.3. del marco teórico.

En primer lugar, hemos calculado el factor de mejora para ambos géneros utilizando la tasa de mortalidad de los años 2018 y 2019, para posteriormente, junto con la mortalidad de cada género, crear unas tablas de mortalidad que nos serán necesarias para poder calcular el BEL y SCR para el riesgo de longevidad.

El ejercicio se ha realizado para una renta vitalicia, para dos personas de diferente sexo, con edades igual a 67 años, las cuales percibirán una renta anual de 10.000€. Se estiman unos gastos fijos de 12€ al año y se tiene en cuenta un factor de inflación sobre dichos gastos del 2%. Para realizar el cálculo del SCR para el riesgo de longevidad hemos tenido en cuenta el shock de longevidad, el cual implica la disminución de la mortalidad un 20%.

Hombre			Mujer		
BEL	BEL long	SCR	BEL	BEL long	SCR
190.995,87€	203.547,81€	12.551,95€	224.062,8€	232.578,75€	8515,96€

Tabla 28: BEL y SCR obtenidos de la aplicación de tablas de mortalidad proyectadas por p-splines

Hombre			Mujer		
BEL	BEL long	SCR	BEL	BEL long	SCR
191.290,63€	202.763,44€	11.472,80€	221.252,73€	229.974,35€	8.721,61€

Tabla 29: BEL y SCR obtenidos de la aplicación de tablas de mortalidad PERM/F 2020

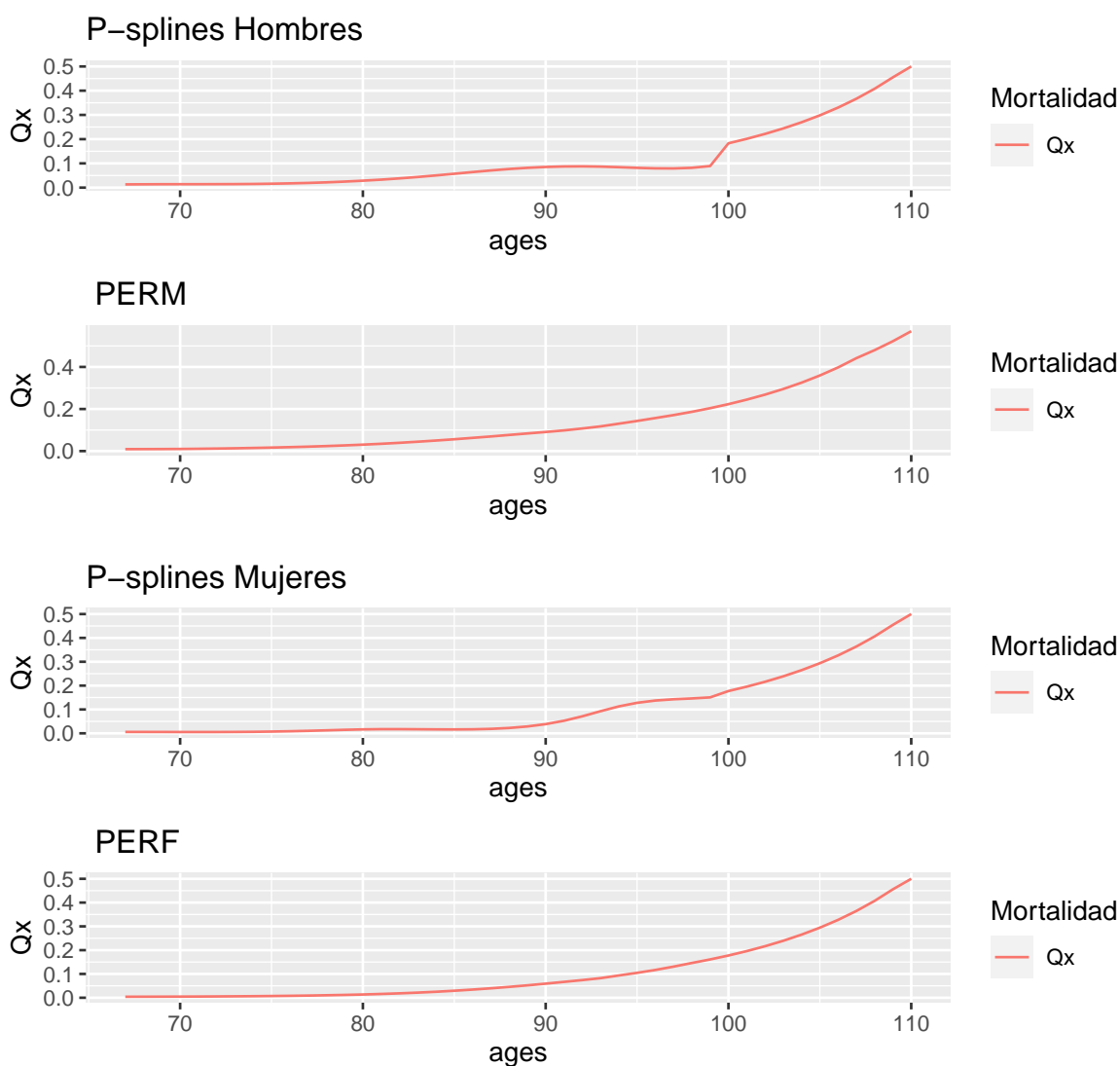
El comportamiento de la mortalidad proyectada con P-splines con respecto a la resultante de utilizar las tablas PERM/F es ligeramente distinto. Aunque a simple vista pueda ser difícil de percibir, la mortalidad resultante de la aplicación de las tablas realizadas a través de P-splines es superior a la mortalidad resultante de aplicar las PERM/F. Como podemos observar en la figura 19, para edades avanzadas, la mortalidad de P-splines presenta dificultades para capturar la tendencia de forma adecuada, a diferencia de la mortalidad de las tablas PERM/F, donde podemos observar una curva suavizada. Esto podría deberse a la escasa existencia de masa poblacional, lo que provoca un incremento de la volatilidad. A diferencia del género masculino, la curva de mortalidad para el género femenino presenta una forma más suavizada, pero de igual modo, con dificultades

para capturar la tendencia correctamente.

Este comportamiento podría explicar que, a pesar de que la mortalidad calculada a partir de las tablas PERM/F es menor que la mortalidad calculada a partir de P-splines, la diferencia entre los BEL calculados utilizando ambas tablas no sea muy pronunciada, llegando incluso a, en el caso del género femenino, ser mayor el BEL estimado a partir de P-splines que el estimado a partir de las tablas PERF (ver Tablas 28 y 29).

**Figura 19**

*Q<sub>x</sub> calculadas en el ejercicio de Reserving procedentes de la utilización de las tablas P-splines y PERM/F para ambos sexos*



Elaboración propia.

## 7. Conclusiones

El estudio del riesgo de longevidad de una determinada población es una actividad esencial dentro del negocio asegurador o reasegurador, ya que nos permitirá desarrollar una capacidad de actuación anticipada ante cualquier riesgo sobrevenido. Existen infinidad de modelos orientados a la predicción de la mortalidad. Entre ellos, los desarrollados en nuestro trabajo, los cuales han sido los más utilizados hasta el momento.

Con la llegada de nuevas técnicas orientadas a la Inteligencia Artificial, hemos podido desarrollar modelos que puedan mejorar la capacidad de predicción de la ocurrencia de diversos riesgos, entre ellos, el riesgo de longevidad. En este trabajo nos hemos centrado principalmente en el desarrollo de modelos utilizando Redes Neuronales Artificiales Clásicas. Como hemos podido observar, concretamente para el caso de la predicción del parámetro Kappa de los diferentes modelos clásicos, los resultados obtenidos han sido bastante favorables, obteniendo predicciones, tanto a largo como a corto plazo, bastante acertadas.

Podríamos decir que, en general, la mejor combinación de hiperparámetros, en este caso, se ha basado en la existencia de una capa oculta, con entre cinco y diez neuronas ocultas, un sesgo que oscilase entre 0.01 y 0.001 y una función de activación softplus. Con respecto a la comparación con otras técnicas de predicción, como se puede apreciar en la Tabla 27, para este set de datos, las Redes Neuronales Artificiales han tenido una mejor capacidad predictiva, independientemente del género analizado, destacando el hecho de que para proyecciones a largo plazo, como es el caso de las proyecciones a 10 años, la capacidad predictiva de las Redes Neuronales Artificiales ha sido notablemente superior. La fuerza predictiva de los modelos de Inteligencia Artificial podría explicarse por la aplicación del modelo en una serie temporal con un horizonte temporal extenso, en nuestro caso, este ha sido de 49 años.

Como hemos mencionado en el marco teórico, no existe un modelo perfecto, sino un mejor modelo según los hiperparámetros utilizados. Esto quiere decir que, posiblemente, existan modelos más o menos eficientes según los hiperparámetros elegidos. Esto lleva a destacar, desde nuestro punto de vista, una de las principales limitaciones de esta

técnica, la selección de hiperparámetros a través de un proceso de prueba y error, lo cual implica una inversión considerable de tiempo intentando obtener un modelo lo más óptimo posible.

Con respecto al ejercicio de Reserving realizado, hemos observado que la mortalidad obtenida a través de la proyección del modelo P-splines es mayor que la aplicada en las tablas PERM/F. En edades avanzadas, debido a la existencia de poca masa poblacional, este modelo comienza a mostrar dificultades a la hora de captar la tendencia de forma correcta debido a la gran volatilidad existente. Este comportamiento, contrastando con el hecho de que para edades avanzadas en las tablas PERM/F existe una desaceleración de la evolución de la mortalidad, podría explicar el hecho de que exista tal ligera diferencia entre el BEL obtenido por las tablas obtenidas a través de la proyección del modelo P-splines y las tablas PERM/F.

Como posibles estudios futuros, sería interesante el poder analizar otros tipos de técnicas de Machine Learning, concretamente relacionadas con las Redes Neuronales Artificiales. Según la literatura, una red que podría tener una buena capacidad predictiva para este tipo de datos podría ser las Redes Neuronales Recurrentes, debido a su capacidad de almacenamiento de información a lo largo de toda la red.

## Bibliografía

1960 IRE WESCON Convention Record, 96-104.

Anderson, J. A. (1993), *The BSB Model: A simple nonlinear autoassociative neural network*, New York, NY: Oxford U. Press.

Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). *Neuronlike adaptive elements that can solve difficult learning control problems*. IEEE Transactions on Systems, Man, and Cybernetics, SMC-13, 834–846.

Cairns, A. J. G., Blake, D., Dowd, K., Coughlan, G. D., Epstein, D., Ong, A., y Balevich, I. (2009). *A Quantitative Comparison of Stochastic Mortality Models Using Data From England and Wales and the United States*. North American Actuarial Journal, 13, 1–35.

Cairns, A. J. G., Blake, D., y Dowd, K. (2006). *A Two-Factor Model for Stochastic Mortality with Parameter Uncertainty: Theory and Calibration*. Journal of Risk & Insurance, 73, 687–718.

Cairns, A. J. G., Blake, D., y Dowd, K. (2008). *Modelling and management of mortality risk: a review*. Scandinavian Actuarial Journal, 2008, 79–113. Ciaburro, G. y Venkateswaran, B. (2017). *Neural Networks with R*, Birmingham, UK:Packt Publishing Ltd.

Clayton, D., y Schifflers, E. (1987). *Models for temporal variation in cancer rates. II: Age–period–cohort models*. Statistics in Medicine, 6, 469–481.

Crossman, A.R. y Neary, D. (2007). *Neuroanatomía. Texto y Atlas en color*, Barcelona, España: Elsevier

Currie, I. D. (2013). *Smoothing constrained generalized linear models with an application to the Lee-Carter model*. Statistical Modelling: An International Journal, 13, 69–93.

Currie, I. D., Durban, M., y Eilers, P. H. C. (2006). *Generalized linear array models with applications to multidimensional smoothing*. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 68, 259–280.

Currie, I. D., y Durban, M. (2002). *Flexible smoothing with P-splines: a unified approach*.



- Statistical Modelling: An International Journal, 2, 333–349. Durbán, M. (2008). Splines con penalizaciones (P-splines); teoría y aplicaciones, Pamplona, España: Universidad Pública de Navarra.
- Eilers, P. H. C., y Marx, B. D. (1996). *Flexible smoothing with B -splines and penalties*. Statistical Science, 11, 89–121.
- Gartner, L. y Hiatt, J. (2011). Histología básica, Barcelona, España: Elsevier Graupe,
- D. (2007). Principles of Artificial Neural Networks. USA: World Scientific.
- Grossberg, S. (1987). *Competitive learning: From interactive activation to adaptive resonance*. Cognitive Science, 11, 23–63.
- Gurney, K. (1997). An introduction to neural networks, London, UK: UCL Press.
- Hainaut, D. (2018). *A Neural-Network Analyzer for Mortality Forecast*. ASTIN Bulletin, 48, 481–508.
- Haines, D. y Mihailoff, G. (2002). Principios de la neurociencia. Aplicaciones básicas y clínicas, Barcelona, España: Elsevier.
- Haykin, S. (1998). Neural Networks: A comprehensive Foundation, Singapore: Pearson Prentice Hall.
- Hebb, D.O. (1949) Organization of behavior, New York: Science Editions.
- Lee, R. (2000). *The Lee-Carter Method for Forecasting Mortality, with Various Extensions and Applications*. North American Actuarial Journal, 4, 80–91.
- Lee, R. D., y Carter, L. R. (1992). *Modeling and Forecasting U.S. Mortality*. Journal of the American Statistical Association, 87, 659–671.
- Lledó, J., Pavía, J. M., & Morillas-Jurado, F. G. (2019). Incorporating big microdata in life table construction: A hypothesis-free estimator. Insurance: Mathematics and Economics, 88, 138–150.
- McCulloch, W. S., y Pitts, W. (1943). *A logical calculus of the ideas immanent in nervous activity*. *The Bulletin of Mathematical Biophysics*, 5, 115–133.
- Minsky, M.L. y Papert, S.A. (1969). Perceptrons, MA, Cambridge: MIT Press.

- Nielsen. M. (2015). Neural Networks and Deep Learning. Recuperado de <http://neuralnetworksanddeeplearning.com/>
- Nigri, A., Levantesi, S., Marino, M., Scognamiglio, S., & Perla, F. (2019). *A Deep Learning Integrated Lee–Carter Model*. *Risks*, 7, 33.
- Rebala, G., Ravi, A. & Churiwala, S. (2019). An introduction to machine learning. Recuperado de <https://doi.org/10.1007/978-3-030-15729-6>
- Renshaw, A. E., y Haberman, S. (2006). *A cohort-based extension to the Lee–Carter model for mortality reduction factors*. *Insurance: Mathematics and Economics*, 38, 556–570.
- Rodríguez-Pardo, J. M., Albarrán, I., Ariza, F. Cóbreces, V. M. y Durbán, M. L. (2014). El riesgo de longevidad y su aplicación práctica a Solvencia II. Modelos actuariales avanzados para su gestión, Madrid, España: Cyan, Proyectos Editoriales, S.A
- Rosenblatt, F. (1958). *The perceptron: A probabilistic model for information storage and organization in the brain*. *Psychological Review*, 65, 386–408.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). *Learning representations by back-propagating errors*. *Nature*, 323, 533-536.
- Werbos, P. J. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Unpublished doctoral dissertation , Harvard University.
- Widrow, B. y Hoff, M.E. (1960). *Adaptative sampled-data systems, a statistical theory of adaptation*.

## Anexo - Código R

```
#El proceso es el mismo para todos los modelos clásicos, con la diferencia de la  
#estimación de dicho modelo a partir de la función fitMotdels.R  
  
#Renshaw-Haberman  
  
#ajuste_M1_H=fit702(ages,years,t(Ex_men),t(Dx_men),matrix(1,length(years),length(ages)))  
  
#APC  
  
#ajuste_M1_H=fit705(ages,years,t(Ex_men),t(Dx_men),matrix(1,length(years),length(ages)))  
  
#CBD  
  
#ajuste_M1_H=fit703(ages,years,t(Ex_men),t(Dx_men),matrix(1,length(years),length(ages)))  
  
#Cargamos los datos  
  
setwd("C:\\Users\\Susana\\Desktop\\TFM\\TFM_final\\Datos")  
  
library(readxl)  
  
#Fallecimientos  
  
Dx_women<- read_excel("Datos_mortalidad.xlsx", sheet="Mujeres",col_names = FALSE)
```

```

Dx_men<- read_excel("Datos_mortalidad.xlsx", sheet="Hombres",col_names = FALSE)

Dx_Total<- read_excel("Datos_mortalidad.xlsx", sheet="Total",col_names = FALSE)

#Esperanza de vida

Ex_women<- read_excel("Datos_EsperanzaV.xlsx", sheet="Mujeres",col_names = FALSE)

Ex_men<- read_excel("Datos_EsperanzaV.xlsx", sheet="Hombres",col_names = FALSE)

Ex_Total<- read_excel("Datos_EsperanzaV.xlsx", sheet="Total",col_names = FALSE)

#Representación tasa mortalidad
ages <-1:100

years<-1960:2018 #Horizonte temporal de los datos

Dx_men<-Dx_men[1:100,1:length(years)]
Ex_men<-Ex_men[1:100,1:length(years)]

Dx_women<-Dx_women[1:100,1:length(years)]
Ex_women<-Ex_women[1:100,1:length(years)]

# Calculamos la fuerza de mortalidad para Hombres y Mujeres.

Mxt_H<-Dx_men/Ex_men
Mxt_M<-Dx_women/Ex_women

source("C:\\Users\\Susana\\Desktop\\TFM\\TFM_final\\Datos\\fitModels.R")

```

```
#LEE CARTER
```

```
ajuste_M1_H=fit701(ages,years,t(Ex_men),t(Dx_men),matrix(1,length(years),  
length(ages)))
```

```
ajuste_M1_M=fit701(ages,years,t(Ex_women),t(Dx_women),matrix(1,length(years),  
length(ages)))
```

```
kappa_times_H<-ajuste_M1_H$kappa
```

```
kappa_times_M<-ajuste_M1_M$kappa
```

```
datos_H<-ts(kappa_times_H,start=c(1960,1),end=c(2018,1), frequency=1)
```

```
datos_M<-ts(kappa_times_M,start=c(1960,1),end=c(2018,1), frequency=1)
```

```
library(fUnitRoots)
```

```
#Hombres
```

```
adfTest(datos_H)
```

```
adfTest(diff(datos_H))
```

```
#Mujeres
```

```
adfTest(datos_M)
```

```
adfTest(diff(datos_M))
```

```
newdatos_H<-diff(datos_H)
```

```
newdatos_M<-diff(datos_M)
```

```

#ETAPA 2: ESTIMACIÓN DEL MODELO

library(forecast)

#Estimación

#Hombres(1,0,2)

ARIMAfit_H <- auto.arima(newdatos_H,approximation=FALSE,trace=TRUE)
summary(ARIMAfit_H)

#Mujeres (1,0,0)

ARIMAfit_M <- auto.arima(newdatos_M,approximation=FALSE,trace=TRUE)
summary(ARIMAfit_M)

library(astsa)

#Hombres(1,0,2)

#(x_H=sarima((newdatos_H),1,0,2))

#x_H$table

#Mujeres (1,0,0)

#(x_M=sarima((newdatos_M),1,0,0))

#x_M$table

```

```

library(Metrics)
#Orig_X_ lo utilizamos para observar
#Orig2_X_ lo utilizamos para proyectar

Fdatos_H<-kappa_times_H[21:59]
Fdatos_M<-kappa_times_M[21:59]

#Hombres

#2009-2010

Orig_2010_H<-ts(Fdatos_H,frequency = 1,start = c(1979,2),end=c(2009,1))

Orig2_2010_H<-ts(Fdatos_H,frequency = 1,start = c(1979,2),end=c(2010,1))

#2009-2014

Orig_2014_H<-ts(Fdatos_H,frequency = 1,start = c(1979,2),end=c(2009,1))

Orig2_2014_H<-ts(Fdatos_H,frequency = 1,start = c(1979,2),end=c(2014,1))

#2009-2018

Orig_2018_H<-ts(Fdatos_H,frequency = 1,start = c(1979,2),end=c(2009,1))

Orig2_2018_H<-ts(Fdatos_H,frequency = 1,start = c(1979,2),end=c(2018,1))

#Mujeres

#2009-2010

```

```

Orig_2010_M<-ts(Fdatos_M,frequency = 1,start = c(1979,2),end=c(2009,1))

Orig2_2010_M<-ts(Fdatos_M,frequency = 1,start = c(1979,2),end=c(2010,1))

#2009-2014

Orig_2014_M<-ts(Fdatos_M,frequency = 1,start = c(1979,2),end=c(2009,1))

Orig2_2014_M<-ts(Fdatos_M,frequency = 1,start = c(1979,2),end=c(2014,1))

#2009-2018

Orig_2018_M<-ts(Fdatos_M,frequency = 1,start = c(1979,2),end=c(2009,1))

Orig2_2018_M<-ts(Fdatos_M,frequency = 1,start = c(1979,2),end=c(2018,1))

library(Metrics)
library(MLmetrics)

#Hombres

#2009-2010

fit_2010_H<-Arima(Orig_2010_H,order=c(1,0,2))
pred_2010_H<- predict(fit_2010_H, n.ahead = 1)
pred1_2010_H<-pred_2010_H$pred

(esperados_2010_H<-round(tail(Orig2_2010_H,1),0))

```



```
(observados_2010_H<-round(head(pred1_2010_H,1),0))
```

```
(rmse(esperados_2010_H,observados_2010_H))
```

```
(RMSPE(esperados_2010_H,observados_2010_H))
```

```
#2009-2014
```

```
fit_2014_H<-Arima(Orig_2014_H,order=c(1,0,2))
```

```
pred_2014_H<- predict(fit_2014_H, n.ahead = 5)
```

```
pred1_2014_H<-pred_2014_H$pred
```

```
(esperados_2014_H<-round(tail(Orig2_2014_H,5),0))
```

```
(observados_2014_H<-round(head(pred1_2014_H,5),0))
```

```
(rmse(esperados_2014_H,observados_2014_H))
```

```
(RMSPE(esperados_2014_H,observados_2014_H))
```

```
#2009-2018
```

```
fit_2018_H<-Arima(Orig_2018_H,order=c(1,0,2))
```

```
pred_2018_H<- predict(fit_2018_H, n.ahead = 10)
```

```
pred1_2018_H<-pred_2018_H$pred
```

```
(esperados_2018_H<-round(tail(Orig2_2018_H,10),0))
```

```
(observados_2018_H<-round(head(pred1_2018_H,9),0))
```

```

(rmse(esperados_2018_H,observados_2018_H))
(RMSPE(esperados_2018_H,observados_2018_H))

#Mujeres

#2009-2010

fit_2010_M<-Arima(Orig_2010_M,order=c(1,0,2))
pred_2010_M<- predict(fit_2010_M, n.ahead = 1)
pred1_2010_M<-pred_2010_M$pred

(esperados_2010_M<-round(tail(Orig2_2010_M,1),0))
(observados_2010_M<-round(head(pred1_2010_M,1),0))

(rmse(esperados_2010_M,observados_2010_M))
(RMSPE(esperados_2010_M,observados_2010_M))

#2009-2014

fit_2014_M<-Arima(Orig_2014_M,order=c(1,0,2))
pred_2014_M<- predict(fit_2014_M, n.ahead = 5)
pred1_2014_M<-pred_2014_M$pred

(esperados_2014_M<-round(tail(Orig2_2014_M,5),0))
(observados_2014_M<-round(head(pred1_2014_M,5),0))

```

```

(rmse(esperados_2014_M,observados_2014_M))
(RMSPE(esperados_2014_M,observados_2014_M))

#2009-2018

fit_2018_M<-Arima(Orig_2018_M,order=c(1,0,2))
pred_2018_M<- predict(fit_2018_M, n.ahead = 10)
pred1_2018_M<-pred_2018_M$pred

(esperados_2018_M<-round(tail(Orig2_2018_M,10),0))
(observados_2018_M<-round(head(pred1_2018_M,10),0))

(rmse(esperados_2018_M,observados_2018_M))
(RMSPE(esperados_2018_M,observados_2018_M))

```

#### *#5. ANN con kappa*

```

ages <-1:100
years <- 1960:2018

Dx_men<-Dx_men[1:100,1:length(years)]
Ex_men<-Ex_men[1:100,1:length(years)]

Dx_women<-Dx_women[1:100,1:length(years)]
Ex_women<-Ex_women[1:100,1:length(years)]

```

*# Calculamos la fuerza de mortalidad para Hombres y Mujeres.*

```

Mxt_H<-Dx_men/Ex_men
Mxt_M<-Dx_women/Ex_women

#Activamos las librerías necesarias
library(readxl)
library(dplyr)
library(forecast)
library(MASS)
library(neuralnet)
library(ggplot2)
library(ggfortify)

#Estandarizamos y dividimos nuestros datos

Train<-as.data.frame(kappa_times_H[1:40])

mean<-apply(Train,2,mean)
sd<-apply(Train,2,sd)

Total<-scale(kappa_times_H, center = mean, scale = sd)

Train_scale<-Total[1:40,]

Test_scale<-Total[40:50,]

Val_scale<-Total[50:59,]

#Train_scale * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')

```

```

#Test_scale * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')

#Val_scale *attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')

#Parte de entrenamiento
#Para la regresión lineal vamos a utilizar 3 variables:
  #X_1= X con un retardo, X_2= X con 2 retardos y X_3= X con 3 retardos

x<-Train_scale[4:40]
x_1<-Train_scale[3:39]
x_2<-Train_scale[2:38]
x_3<-Train_scale[1:37]

#Creamos dataframe con las variables

Ent<-data.frame(x_1,x_2,x_3,x)

softplus <- function(x) log(1+exp(x))

modelo.nn<-neuralnet(x~x_1+x_2+x_3,
                    data=Ent,
                    hidden = c(5),
                    threshold = 0.01,
                    algorithm = "rprop+",
                    err.fct = "sse",
                    linear.output=TRUE,
                    act.fct = softplus

```

```

)
#Aplicamos modelo a nuestro conjunto de testeo
x<-Test_scale[3:10]
x_1<-Test_scale[3:10]
x_2<-Test_scale[2:9]
x_3<-Test_scale[1:8]

Test<-data.frame(x_1,x_2,x_3)

Prueba.nn <- neuralnet::compute(modelo.nn, Test)
results <- data.frame(Real=x,actual = Test, prediction = Prueba.nn$net.result)

uns_test<-results * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')

#Backtesting

#2009-2010

x_1<-Val_scale[1:2]
x_2<-Val_scale[1:2]
x_3<-Val_scale[1:2]
Val<-data.frame(x_1,x_2,x_3)

Val.nn<-neuralnet::compute(modelo.nn,Val)

Valnet_uns_10<-Val.nn$net.result * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')
Val_uns_10<-Val_scale * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')

```

```

rmse(Val_uns_10[1:2],Valnet_uns_10)
RMSPE(Val_uns_10[1:2],Valnet_uns_10)

#2009-2014

x_1<-Val_scale[1:5]
x_2<-Val_scale[1:5]
x_3<-Val_scale[1:5]
Val<-data.frame(x_1,x_2,x_3)

Val.nn<-neuralnet::compute(modelo.nn,Val)

Valnet_uns_14<-Val.nn$net.result * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')
Val_uns_14<-Val_scale * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')

rmse(Val_uns_14[1:5],Valnet_uns_14)
RMSPE(Val_uns_14[1:5],Valnet_uns_14)

#2009-2018

x_1<-Val_scale[1:10]
x_2<-Val_scale[1:10]
x_3<-Val_scale[1:10]
Val<-data.frame(x_1,x_2,x_3)

Val.nn<-neuralnet::compute(modelo.nn,Val)

Valnet_uns_18<-Val.nn$net.result * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')
Val_uns_18<-Val_scale * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')

```

```

rmse(Val_uns_18,Valnet_uns_18)
RMSPE(Val_uns_18,Valnet_uns_18)

#Mujeres

#Dividimos nuestros datos en Test , entrenamiento y validación

Train<-as.data.frame(kappa_times_M[1:40])

mean<-apply(Train,2,mean)
sd<-apply(Train,2,sd)

Total<-scale(kappa_times_M, center = mean, scale = sd)

Train_scale<-Total[1:40,]

Test_scale<-Total[40:50,]

Val_scale<-Total[50:59,]

#Train_scale * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')

#Test_scale * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')

#Val_scale *attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')

```



```

#Parte de entrenamiento

#Para la regresión lineal vamos a utilizar 3 variables:
  #X_1= X con un retardo, X_2= X con 2 retardos y X_3= X con 3 retardos

x<-Train_scale[4:40]
x_1<-Train_scale[3:39]
x_2<-Train_scale[2:38]
x_3<-Train_scale[1:37]

#Creamos dataframe con las variables

Ent<-data.frame(x_1,x_2,x_3,x)

softplus <- function(x) log(1+exp(x))

modelo.nn<-neuralnet(x~x_1+x_2+x_3,
                    data=Ent,
                    hidden = c(5),
                    threshold = 0.01,
                    algorithm = "rprop+",
                    err.fct = "sse",
                    linear.output=TRUE,
                    act.fct = softplus

)

#Aplicamos modelo a conjunto de testeo

```

```

x<-Test_scale[3:10]
x_1<-Test_scale[3:10]
x_2<-Test_scale[2:9]
x_3<-Test_scale[1:8]

Test<-data.frame(x_1,x_2,x_3)

Prueba.nn <- neuralnet::compute(modelo.nn, Test)
results <- data.frame(Real=x,actual = Test, prediction = Prueba.nn$net.result)

#Unscale

uns_test<-results * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')

rmse(uns_test$Real,uns_test$prediction)

#Backtesting

#2009-2010

x_1<-Val_scale[1:2]
x_2<-Val_scale[1:2]
x_3<-Val_scale[1:2]
Val<-data.frame(x_1,x_2,x_3)

Val.nn<-neuralnet::compute(modelo.nn,Val)

Valnet_uns_10<-Val.nn$net.result * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')
Val_uns_10<-Val_scale * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')

```

```

rmse(Val_uns_10[1:2],Valnet_uns_10)
RMSPE(Val_uns_10[1:2],Valnet_uns_10)

#2009-2015

x_1<-Val_scale[1:5]
x_2<-Val_scale[1:5]
x_3<-Val_scale[1:5]
Val<-data.frame(x_1,x_2,x_3)

Val.nn<-neuralnet::compute(modelo.nn,Val)

Valnet_uns_14<-Val.nn$net.result * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')
Val_uns_14<-Val_scale * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')

rmse(Val_uns_14[1:5],Valnet_uns_14)
RMSPE(Val_uns_14[1:5],Valnet_uns_14)

#2009-2018

x_1<-Val_scale[1:10]
x_2<-Val_scale[1:10]
x_3<-Val_scale[1:10]
Val<-data.frame(x_1,x_2,x_3)

Val.nn<-neuralnet::compute(modelo.nn,Val)

Valnet_uns_18<-Val.nn$net.result * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')
Val_uns_18<-Val_scale * attr(Total, 'scaled:scale') + attr(Total, 'scaled:center')

```

```

rmse(Val_uns_18,Valnet_uns_18)
RMSPE(Val_uns_18,Valnet_uns_18)

#P-splines

library(svcM)
library(lattice)
library(splines)
library(Matrix)
library(MortalitySmooth)

#model_bic_H = c()
#model_bic_M = c()

#nodos_edad = c()
#nodos_yrs = c()

#for (j in 2:15){
#  for (i in 2:15){
#    fitBIC_H <- Mort2Dsmooth(x=ages, y= years, Z= as.matrix(Dx_men),
#                           offset = log(as.matrix(Ex_men)), ndx =c(i,j))
#    model_bic_H[length(model_bic_H)+1] = fitBIC_H$bic
#    nodos_edad[length(nodos_edad)+1] = i
#    nodos_yrs[length(nodos_yrs)+1] = j
#  }
#}

#results_H <-cbind(nodos_edad,nodos_yrs,model_bic_H)

```

```

#for (j in 2:15){
  #for (i in 2:15){
    # fitBIC_M <- Mort2Dsmooth(x=ages, y= years, Z= as.matrix(Dx_women),
    #                          offset = log(as.matrix(Ex_women)), ndx =c(i,j))
    #   model_bic_M[length(model_bic_M)+1] = fitBIC_M$bic
    #   nodos_edad[length(nodos_edad)+1] = i
    #   nodos_yrs[length(nodos_yrs)+1] = j
  # }
#}

```

```

#results_M <-cbind(nodos_edad,nodos_yrs,model_bic_H)

```

```

#results<-cbind(nodos_edad,nodos_yrs,model_bic_H)

```

```

#results<-cbind(nodos_edad,nodos_yrs,model_bic_M)

```

```

#min(model_bic_H) # Corresponde a 15 para edad y 12 para años .

```

```

#min(model_bic_M) # Corresponde a 15 para edad y 12 para años .

```

```

#fitBIC_H <- Mort2Dsmooth(x=ages,y=years,Z=as.matrix(Dx_men),
                          #offset = log(as.matrix(Ex_men)), ndx =c(15,12))

```

```

#fitBIC_M <- Mort2Dsmooth(x=ages,y=years,Z=as.matrix(Dx_women),
                          #offset = log(as.matrix(Ex_women)), ndx =c(15,12))

```

```

fitBIC_H<-list.load("C:\\Users\\Susana\\Desktop\\TFM\\TFM_final\\scripts_modelos\\P-sp

```

```

fitBIC_M<-list.load("C:\\Users\\Susana\\Desktop\\TFM\\TFM_final\\scripts_modelos\\P-sp

```

```

source("C:\\Users\\Susana\\Desktop\\TFM\\TFM_final\\Datos\\my.predict.R")

newyears <-1960:2019

newdata<-list(x=ages, y=newyears)

pred_H<-my.predict(fitBIC_H,newdata=newdata,se.fit=TRUE) #Saca intervalos
pred2_H<-predict(fitBIC_H,newdata=newdata,se.fit=TRUE)

#Mujeres

pred_M<-my.predict(fitBIC_M,newdata=newdata,se.fit=TRUE) #Saca intervalos
pred2_M<-predict(fitBIC_M,newdata=newdata,se.fit=TRUE)

#Transformamos mx a qx

#Hombres

qx_H<-exp(pred_H$fit)

qx_H<-1-exp(-qx_H)

#Mujeres

qx_M<-exp(pred_M$fit[,60])

qx_M<-1-exp(-qx_M)

#Factor de mejora para tabla de mortalidad

```

```
#Hombres
mx2019_H<-as.data.frame(exp(pred_H$fit[,60]))
mx2018_H<-as.data.frame(exp(pred_H$fit[,59]))

mxt_t1_H<-data.frame("2019"=mx2019_H, "2018"=mx2018_H, lambda=mx2019_H/mx2018_H)

lambda_H<-vector(length = 100)
for (i in 1:100){
  lambda_H[i]<-1-mx2019_H[i,1]/mx2018_H[i,1]
  if (lambda_H[i]<0){
    lambda_H[i]=0
  }
}
```