TÍTULO

CORPORACIÓN FAVORITA GROCERY SALES FORECASTING KAGGLE COMPETITION

AUTOR

Augusto Steves Mendoza Calero

Esta edición electrónica ha sido realizada en 2018

Director/Tutor	Dr. José Manuel Bravo Caro
Instituciones	Universidad Internacional de Andalucía ; Universidad de Huelva
Curso	*Máster Oficial en Economía, Finanzas y Computación*
ISBN	978-84-7993-502-3
©	Augusto Steves Mendoza Calero
©	De esta edición: Universidad Internacional de Andalucía
Fecha documento	2018

Corporación Favorita Grocery Sales Forecasting Kaggle Competition

By

Augusto Steves Mendoza Calero

A thesis submitted in conformity with the requirements for the MSc in Economics, Finance and Computer Science.

University of Huelva & International University of Andalucía

November 2018

Corporación Favorita Grocery Sales Forecasting Kaggle Competition

Augusto Steves Mendoza Calero

Master en Economía, Finanzas y Computación

Supervisado por:

Dr. José Manuel Bravo Caro

Universidad de Huelva y Universidad Internacional de Andalucía

2018

ABSTRACT

Currently, data analysis is vitally important in all sectors and industries, the process of digital transformation that is being generated worldwide makes the different organizations make important investments in building databases to later analyze them and improve the process of decision making supported by data.

This MSc. final project contributes with a solution to a problem presented by the Ecuadorian company Corporación Favorita, in the Kaggle.com platform. It's a problem of time series, which is to be performed forecasting sales of the company.

JEL Classification: C01, C02, C19, C22, C32, C53, M39.

Keywords: Forecasting, Time Series, R, Arima, Ets.

Resumen

En la actualidad el análisis de datos tiene importancia vital en todos los sectores e industrias, el proceso de transformación digital que se está generando a nivel mundial hace que las diferentes organizaciones realicen importantes inversiones en construir bases de datos para posteriormente analizarlas y mejorar el proceso de toma de decisiones apoyado en datos.

Este proyecto final de master contribuye con una solución a un problema presentado por la empresa ecuatoriana Corporación Favorita en la plataforma Kaggle.com. Es un problema de series temporales, en el cual se tiene que realizar la predicción de ventas de la empresa.

Clasificación JEL: C01, C02, C19, C22, C32, C53, M39.

Palabras clave: Predicción, Series Temporales, R, Arima, Ets.

Agradecimientos

En primer lugar, agradezco a mi familia, especialmente a mi padre y mi madre; por el apoyo constante durante toda mi formación profesional.

Al Dr. José Manuel Bravo Caro, supervisor del presente trabajo, por su ayuda y colaboración.

A la Universidad Internacional de Andalucía (UNIA), por las becas otorgadas para cubrir mi estadía en el campus universitario durante la celebración del master.

A la Asociación Universitaria Iberoamericana de Postgrado (AUIP), por las becas de posgrado otorgadas para poder realizar la movilidad entre Ecuador y España.

A todo el personal académico de la Universidad de Huelva y Universidad Internacional de Andalucía, por ayudarme a expandir el conocimiento en el extraordinario mundo del análisis de datos.

Índice de contenidos

Índice de tablas

Índice de gráficos

1 Introducción

El análisis de datos es uno de los campos en donde todo tipo de organizaciones a nivel mundial están invirtiendo mucho tiempo y esfuerzo. Según la Organización de las Naciones Unidad para el año 2015 ya se habían producido más datos que en toda la historia de la humanidad. Dado a la importancia de los datos, la ONU ha creado su propia división de análisis de datos denominada Global Pulse y a inicios del 2017 se realizó el Primer Foro Mundial de Datos, que reunió a entes público, privado, académico y de la sociedad civil.

Para las empresas, los datos se han convertido en unos de los recursos más valiosos, luego de realizar un buen proceso de tratamiento y posterior análisis puede significar en un recurso fundamental a la hora de realizar la toma de decisiones. Las empresas más poderosas del mundo como Google, Amazon, Facebook, Apple, Microsoft, al estar en el sector tecnológico tienen la capacidad necesaria para poder recopilar, almacenar, procesar y analizar cantidades descomunales de datos, para respaldar su toma de decisiones en un riguroso análisis de datos; esto es uno de los factores claves por los cuales obtienen ganancias multimillonarias.

En un mercado en donde las empresas buscan ser cada vez más competitivas, el aprovechamiento de las tecnologías de la información para generar y almacenar grandes volúmenes de datos requiere a su vez profesionales capaces de generar conocimiento realizando el análisis de datos. Por este motivo, el análisis de datos es unos de los campos con mayor demanda de profesionales altamente capacitados.

En este sentido, la mayoría de las empresas de todos los tamaños han podido realizar el proceso de recolección y almacenamiento de datos, sin embargo, pocas han podido obtener un valor agregado porque no cuentan con el recurso humano necesario para lograrlo.

Una alternativa que han encontrado las empresas es la contratación de servicios de consultoría, que se encarguen del complejo proceso de generación de conocimiento a partir de los datos, mientras estas se ven involucradas en un proceso de creación de un área especializada en el análisis de datos que pueda generar beneficios.

En el Ecuador, un país de renta media que se encuentra en la etapa inicial del proceso de transformación digital, es pequeño el número de empresas que realiza la toma de decisiones tomando como pilar fundamental el análisis de datos. De acuerdo a la Superintendencia de Compañías del Ecuador en el año 2017 solo el 10.9% de las empresas obtuvieron ingresos anuales por encima de 1 millón de dólares.

Corporación la Favorita se encuentra en el top 5 de empresas con más ingresos en el Ecuador de acuerdo al número de ingresos anuales y es la primera en convocar a una competencia de análisis de datos con la finalidad de encontrar el mejor modelo que realice el pronóstico de sus ventas.

1.1 Elección del problema

El trabajo final de master tiene como finalidad demostrar la capacidad de aplicar el conocimiento adquirido durante en el proceso de formación académica.

El master tiene como enfoque principal el análisis de datos, visto desde diferentes perspectivas como la de investigación académica o la aplicación de una solución de casos relacionados con la empresa. De tal manera que se trabaja en el desarrollo de competencias congruentes al análisis económico, métodos cuantitativos, programación informática, algoritmos, etc.

Se ha elegido resolver un problema real de análisis de datos propuesto por una de las empresas más importantes del Ecuador, es un desafío por la complejidad que se puede presentar durante todo el proceso de solución; lo cual contribuirá al desarrollo de competencias y experiencia, para en el futuro hacer frente a las diferentes problemáticas que se puedan presentar en el campo del análisis de datos.

En virtud de lo expuesto, el problema elegido tiene la particularidad de ser una competencia en donde participan analistas de datos a nivel mundial en busca del mejor pronóstico de ventas. La competición ya ha finalizado y ha encontrado un ganador, por lo cual se evita realizar el trabajo bajo los tiempos establecidos por los organizadores de la competición.

En este sentido, se busca brindar solución a la problemática planteada, pero su estructura general tendrá cierto grado de diferenciación respecto a las ya presentadas en la competencia.

1.2 Sobre Corporación la Favorita

Corporación Favorita, es una empresa de origen ecuatoriana con presencia en Ecuador, Costa Rica, Colombia, Perú, Paraguay y Chile, su sede principal se encuentra en la ciudad de Quito, Ecuador.

Nació en 1952, como una bodega que comercializaba artículos para el hogar, de origen nacional e importados. En 1957 abrió Supermercados La Favorita, el primer autoservicio del país; en 1976 fue la primera empresa en abrir su capital al público. Debido al crecimiento y diversificación sus actividades en líneas, formatos y tipos de locales, es una de las más pioneras del país.

Se encuentra entre las cinco empresas más grandes del país de acuerdo al nivel de ingresos anuales. El giro del negocio se basa principalmente en tiendas de autoservicio en las que se ofrecen alimentos, productos de primera necesidad, entre otros productos.

Debido a la diversidad de servicios que ofrece con altos volúmenes de ventas, la estructura de la empresa está dividida en cuatro áreas: Comercial, Industrial, Inmobiliaria y Responsabilidad Social.

1.3 Sobre Kaggle

Kaggle es la comunidad de científicos de datos más grande del mundo. En su plataforma se realizan principalmente competiciones de análisis de datos, aunque la plataforma también ofrece un banco de trabajo, datos públicos y educación en línea para la inteligencia artificial. Para junio del 2017, Kaggle había superado la cifra de 1.000.000 de usuarios en todo el mundo, abarcando 194 países, siendo la más diversa a nivel mundial.

Las competiciones que se han realizado en Kaggle tienen un enorme impacto, dando lugar a la publicación de artículos académicos sobre nuevos hallazgos científicos encontrados durante la competencia, así como el aporte a todo tipo de proyectos como el mejoramiento de reconocimiento de gestos de Microsoft Kinect.

El tipo de organizaciones es muy diverso, sobre todo las empresas que están en el proceso de trasformación digital han invertido en esta plataforma para que organicen competencias que ayuden a resolver problemas específicos relacionados con el campo del análisis de datos.

Las competiciones funcionan de la siguiente manera:

1. El anfitrión de la competencia prepara los datos y una descripción del problema.
2. Los participantes experimentan con diferentes técnicas y compiten entre sí para producir los mejores modelos.
3. El trabajo se envía a través de la plataforma para obtener una calificación respecto a los requerimientos de la competencia.
4. Terminado el plazo de la competición, el promotor realiza el pago del premio y obtiene a perpetuidad el modelo ganador.

Las competiciones de Kaggle no ofrecen siempre un premio económico, muchas competiciones se realizan con carácter público para alentar a la comunidad a ganar experiencia y prestigio a través de un ranking de los mejores análisis de datos.

Kaggle también organiza competencias de reclutamiento en las que los científicos de datos compiten por la posibilidad de entrevistarse en compañías líderes en ciencia de datos como Facebook, IBM y Google.

1.4 Sobre la competición Corporación Favorita Grocery Sales Forecasting

Los establecimientos comerciales siempre se encuentran con un gran problema a tener una gran cantidad de inventarios, en especial cuando se trata de productos perecederos.

Por lo cual, es importante poder predecir mejor las ventas, debido a que se puede anticipar los productos que se agotaran más rápidamente y así tener el numero óptimo de existencias.

Tener en stock los productos preferidos de los consumidores puede marcar la diferencia entre un cliente satisfecho realizando compras recurrentes y uno insatisfecho yéndose a la competencia. El problema se vuelve más complejo a medida que se agregan nuevos productos, gustos estacionales en transición y mercadeo de productos impredecible.

Corporación Favorita al operar cientos de supermercados, con más de 200,000 productos diferentes en sus estantes sabe lo complicado que puede ser realizar un buen pronóstico de ventas.

En la competición de Kaggle, Corporación la Favorita buscaba modelos que fueran capaces de pronosticar con mayor precisión las ventas, ya que sus métodos de predicción se basaban en previsión subjetiva con muy pocos datos para respaldarlos y muy poca automatización para ejecutar planes.

En este sentido, estaban muy entusiasmados en ver como el aprendizaje automático les podría garantizar que sus clientes se encuentren satisfechos al tener justo lo suficiente de los productos adecuados en el momento correcto.

La competición se realizó durante el mes de enero del 2018. Los premios ofrecidos por Corporación la Favorita ascendían a un total de $30.000 dólares, los cuales fueron repartidos de la siguiente manera:

- Primer lugar- $15,000
- Segundo lugar - $10,000
- Tercer lugar - $5,000

Además, Corporación Favorita consideraría a los mejores 3 mejores competidores individuales de nacionalidad ecuatoriana para realizar una entrevista en la empresa. En caso de que fueran contratados, los candidatos eran elegibles para un bono de bienvenida de $ 5,000 dólares en su primer aniversario de empleo.

1.5 Sobre la base de datos

La base de datos enviada por Corporación la Favorita a la competición tiene 126 millones de registros, teniendo un peso de 4.81 GB una vez que se descomprime. La cual contiene los datos de entrenamiento, los datos de test, además de archivos con metadatos sobre los artículos, las tiendas, promociones, inclusive el precio del petróleo del Ecuador.

Los datos de entrenamiento son los que se usarán para determinar las posibles relaciones existentes entre las diferentes variables para luego entrenar los modelos con los que posteriormente se realizarán las predicciones.

Mientras tanto, los datos de test son los destinados a realizar la evaluación de las predicciones, para determinar cuáles son los modelos con la mejor calidad de predicción. Una particularidad que tienen los datos de test es que contienen una pequeña cantidad de elementos que no están en los datos de entrenamiento, ya que parte de estos incluyen la predicción de un nuevo artículo basado en productos similares.

Además, se incluye un archivo de ejemplo de cómo se tienen que presentar las predicciones en la plataforma de Kaggle. Esto tiene una importancia muy grande, debido a que si no se sube en el formato establecido no se recibe ningún tipo de puntuación.

Finalmente, resultará muy interesante elegir la proporción a utilizar de los datos de entrenamiento dado la cantidad de computación necesaria para procesar tanta información.

1.6 Objetivos

El objetivo principal del presente trabajo es resolver el problema planteado por Corporación la Favorita, demostrando la capacidad de trabajar con problemas reales de análisis de datos.

Los objetivos específicos son los siguientes:

- Encontrar el mejor modelo de predicción de ventas logrando una buena puntuación en la competencia.
- Realizar un Análisis Exploratorio de Datos (EDA) de alto nivel.
- Mejorar el nivel de conocimiento y habilidades de programación en R Studio.

2 Modelos predictivos y método de evaluación

En esta sección del trabajo se desarrollará la parte teórica de los modelos de predicción que se utilizan para el tipo de problema planteado por Corporación La Favorita, así como la métrica de evaluación que se utiliza para puntuar el mejor modelo.

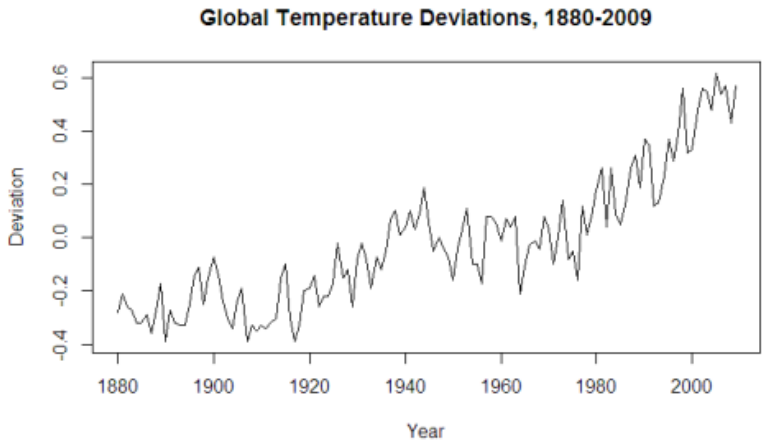
2.1 Series temporales

Una serie temporal es una colección de observaciones de una variable tomadas de forma secuencial y ordenada en el tiempo. Las series pueden tener una periodicidad anual, semestral, trimestral, mensual, etc., según los periodos de tiempo en los que están recogidos los datos que la componen.

El objetivo de una serie temporal consiste en estudiar los cambios en esa variable con respeto al tiempo (descripción), y en predecir sus valores futuros (predicción). Es decir, la serie tiene como variable independiente el tiempo ($\boldsymbol{t}$) y una variable dependiente del objetivo ($\boldsymbol{y_t}$)**.** La predicción se representa como ($\boldsymbol{\hat{y}_t}$)**.**

Las series temporales están presentes en muchos campos como: la economía con datos del PIB, tasa inflación, tasa desempleo; la demografía con las tasas de nacimientos, hasta la meteorología con datos de las temperaturas. Como se muestra en la figura 1.

Figura 1. Ejemplo de serie temporal



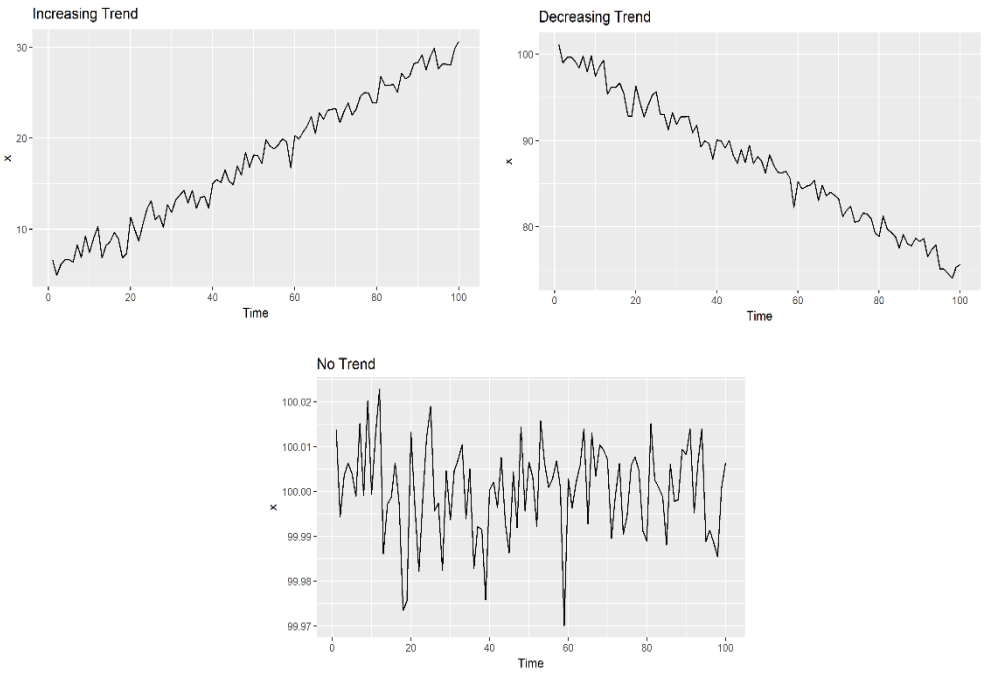
***Fuente**: Forecasting: Principles and Practice*

2.1.1 Componentes de una serie temporal

Con bastante frecuencia, las series temporales presentan una o varias características, denominadas componentes. El estudio descriptivo de series temporales se basa en la idea de descomponer la variación de una serie en varias componentes básicas que se detallan a continuación:

Tendencia: existe una tendencia cuando una serie aumenta, disminuye o permanece en un nivel constante con respecto al tiempo. Por lo tanto, el tiempo se toma como una característica. La figura 2 muestra varios ejemplos.

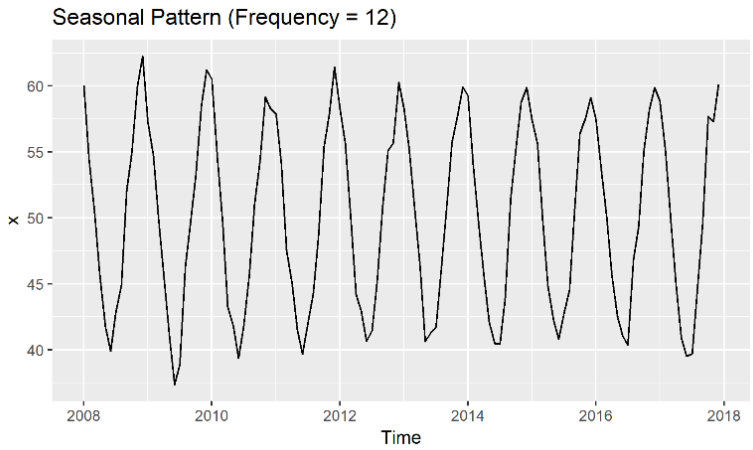
Figura 2. Componente de tendencia



***Fuente**: Forecasting: Principles and Practice*

Estacionalidad: Se refiere a las fluctuaciones en los datos relacionados con los ciclos del calendario, estos patrones periódicos se repiten a una frecuencia constante ($\boldsymbol{m}$)**.** La figura 3 muestra un componente estacional con ($\boldsymbol{m = 12}$), es decir, el patrón periódico se repite cada doce meses.

Figura 3.Componente Estacional

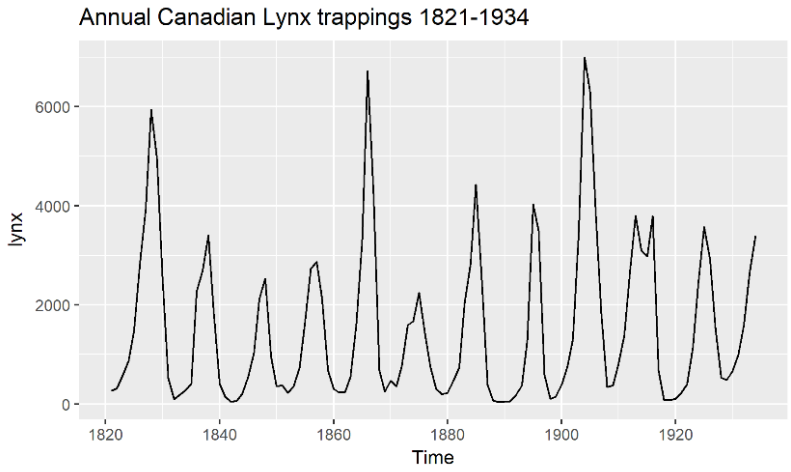


***Fuente**: Forecasting: Principles and Practice*

Ciclos: Los ciclos son estaciones que no ocurren a una tasa fija. Normalmente, los componentes de tendencia y ciclo se agrupan.

En la serie temporal que se mostrará en la figura 4, se muestran patrones estacionales y cíclicos. Estos no se repiten a intervalos de tiempo regulares y pueden ocurrir incluso si la frecuencia es 1 ($\boldsymbol{m = 1}$).

Figura 4. Componente cíclico



***Fuente**: Forecasting: Principles and Practice*

Finalmente, parte de la serie que no se puede atribuir a los componentes estacionales, de ciclo o de tendencia se conoce como **residuo o error.**

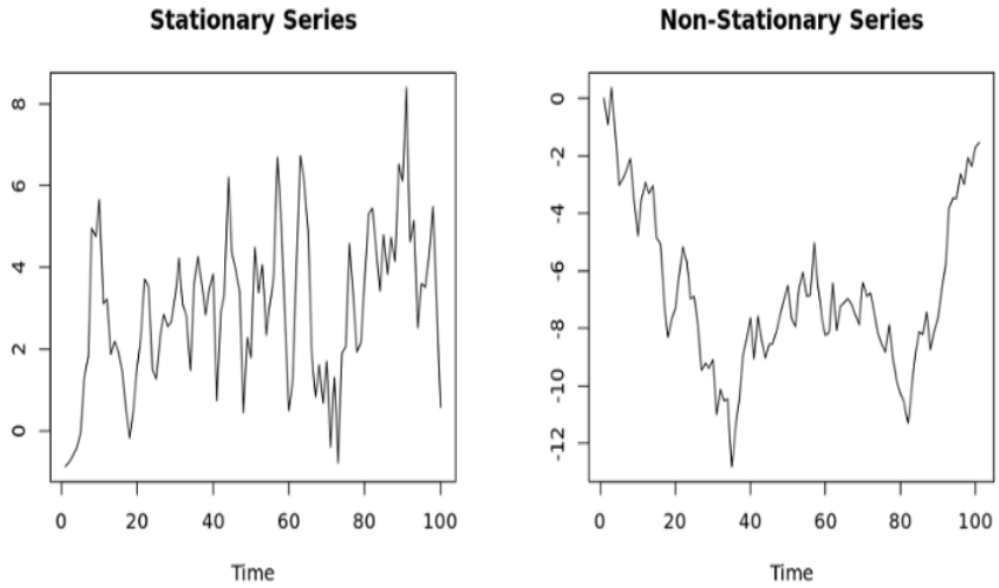
2.1.2 Tipos de series temporales

Las series temporales son de varios tipos, las cuales se detallan a continuación:

- **Estacionarias**: Se dice que una serie es estacionaria cuando su media, varianza y autocovarianza son invariantes en el tiempo.
- **No estacionarias:** Por el contrario, al concepto anterior este tipo de series son aquellas en las que las propiedades estadísticas de la serie sí varían con el tiempo. Estas series pueden mostrar cambio de varianza, tendencia o efectos estacionales a lo largo del tiempo.

En la figura 5, se muestra una serie estacionaria y otra no estacionaria.

Figura 5. Tipos de series temporales



***Fuente**: Forecasting: Principles and Practice*

Además, las series temporales también se pueden dividir según su variabilidad:

- **Univariante**: la serie temporal es un conjunto de observaciones sobre una única característica o variable.
- **Multivariante**: (o vectorial): la serie temporal es un conjunto de observaciones de varias variables.

2.2 Métodos simples

Algunos métodos de pronóstico son extremadamente simples y sorprendentemente efectivos, por lo cual es importante utilizarlos antes de realizar métodos más complejos.

2.2.1 Media Histórica

Este es el método más básico y sencillo que se puede usar para realizar una predicción en series temporales.

Aquí, las predicciones de todos los valores futuros son iguales a la media de los datos históricos. Si dejamos que los datos históricos sean denotados por $\boldsymbol{y_1, \dots, y_T}$,entonces podemos escribir los pronósticos como:

$$\boldsymbol{\hat{y}_{T+h|T} = \hat{y} = (y_1 + \cdots + y_t)/T.}$$

La notación $\boldsymbol{\hat{y}_{T+h|T}}$ es una abreviatura para la estimación de $\boldsymbol{\hat{y}_{T+h}}$ basada en los datos de $\boldsymbol{y_1, \dots, y_T}$.

Mediante este método todos los datos tienen el mismo peso.

2.2.2 Predictor ingenuo

Para el predictor ingenuo simplemente establecemos todos los pronósticos como el valor de la última observación. Es decir, $\boldsymbol{\hat{y}_{T+h|T} = y_T}$

Este método resulta muy eficiente para las series económicas y financieras.

Un método similar es útil para datos altamente estacionales. En este caso, establecemos que cada pronóstico sea igual al último valor observado de la misma temporada del año (por ejemplo, el mismo mes del año anterior). Formalmente, el pronóstico para el tiempo $\boldsymbol{T + h}$ se escribe como:

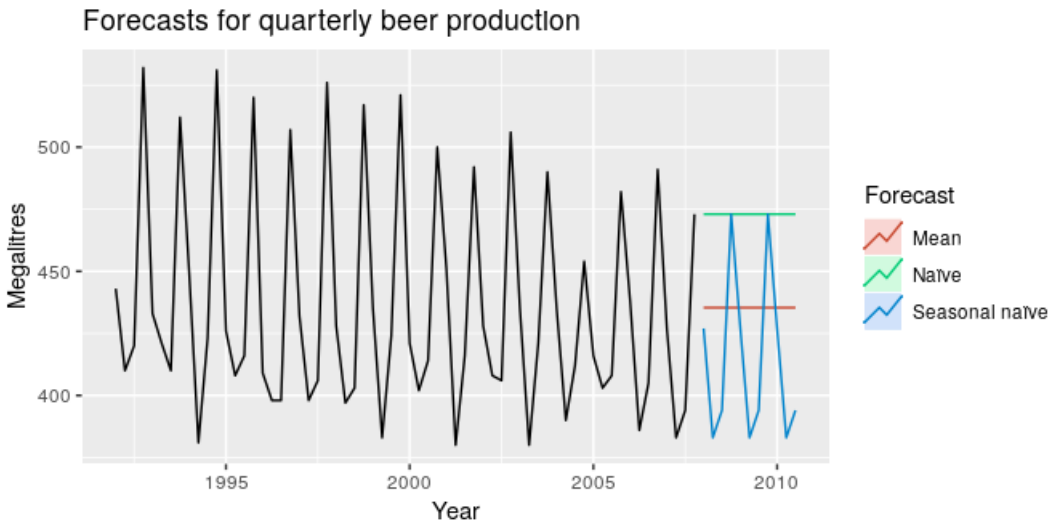
$$\boldsymbol{\hat{y}_{T+h|T} = y_{T+h-m(k+1)}}$$

donde $\boldsymbol{m}$ es el período estacional, y $\boldsymbol{k}$ es la parte entera de $\boldsymbol{(h - 1) / m}$ (es decir, el número de años completos en el período de pronóstico anterior al tiempo $\boldsymbol{T + h}$).

Por ejemplo, con datos mensuales, el pronóstico para todos los valores futuros de febrero es igual al último valor observado de febrero. Reglas similares aplican para otros meses y trimestres, y para otros períodos estacionales.

La figura 6 muestra cómo actúan los 3 métodos aplicados a series con comportamiento estacional.

Figura 6.Ejemplo de series temporales con los 3 métodos simples



***Fuente**: Forecasting: Principles and Practice*

Comenzar utilizando un método simple puede resultar muy útil para tener como punto de partida si es factible utilizar métodos más complejos.

2.3 Métodos de Suavizado Exponencial

El suavizado exponencial se propuso a fines de la década de 1950 (Brown, 1959; Holt, 1957; Winters, 1960), en este método cuanto más reciente sea la observación, mayor será el peso asociado.

Estos métodos obtienen predicciones confiables una amplia gama de series de tiempo, lo que es una gran ventaja y de gran importancia para las aplicaciones en la industria.

2.3.1 Método de Suavizado Exponencial Simple (SES)

El más simple de los métodos de suavizado exponencial se llama naturalmente suavizado exponencial simple (SES).

En este método los pronósticos se calculan utilizando promedios ponderados, donde los pesos disminuyen exponencialmente a medida que las observaciones provienen de otras más en el pasado; los pesos más pequeños se asocian con las observaciones más antiguas:

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots,$$

donde $0 \leq \alpha \leq 1$ es el parámetro de suavizado.

2.3.2 Método de tendencia lineal de Holt

Holt (1957) extendió el suavizado exponencial simple para permitir el pronóstico de datos con una tendencia. Este método implica una ecuación de pronóstico y dos ecuaciones de suavizado:

Predicción	$\hat{y}_{t+h\|t} = \ell_t + hb_t$
Ecuación de nivel	$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$
Ecuación de tendencia	$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$.

donde ℓ_t denota una estimación del nivel de la serie temporal t ,b_t denota una estimación de la tendencia t , α es el parámetro de suavizado para el nivel $0 \leq \alpha \leq 1$, y β^* es el parámetro de suavizado de tendencia, $0 \leq \beta^* \leq 1$

La evidencia empírica indica que este método tiende a sobreestimar, sobre todo para un horizonte de largo plazo. En este sentido, Gardner y McKenzie (1985) introdujeron un parámetro que "amortigua" la tendencia a una línea plana en el futuro, lo que se conoce como “Método de tendencia amortiguada”.

2.3.3 Método de tendencia amortiguada

En conjunto con los parámetros de suavizado α y β (Holt), se incluye un parámetro de amortiguación $0 < \phi < 1$:

$$\hat{y}_{t+h|t} = \ell_t + (\phi + \phi^2 + \cdots + \phi^h)b_t$$

$$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$$

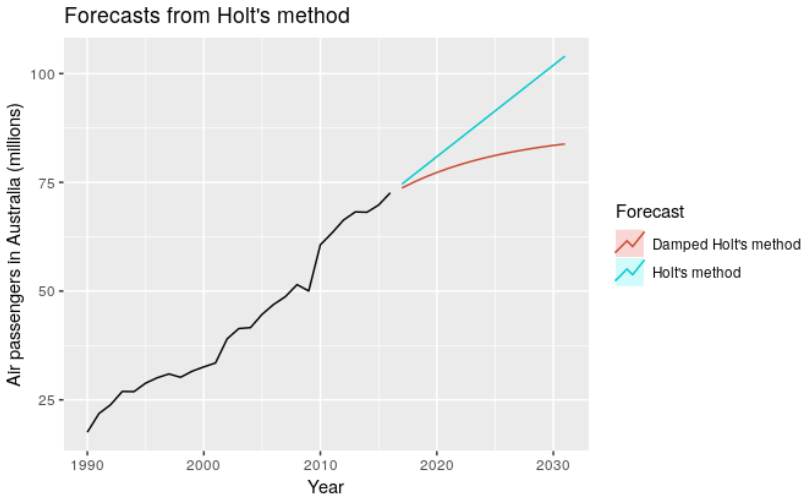
$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}.$$

Si $\phi = 1$,es idéntico al método de Holt, para valores entre 0 y 1, ϕ amortigua la tendencia para que se aproxime a una constante en el futuro. De esta manera, el pronóstico converge en $\ell_T + \phi b_T/(1-\phi)$ como $h \to \infty$ para cualquier valor $0 < \phi < 1$.

Esto significa que las previsiones a corto plazo tienen una tendencia, mientras que las previsiones a largo plazo son constantes.

En la figura 7 se muestra una comparación del método tendencia lineal versus el método de tendencia amortiguada.

Figura 7.Comparación método tendencia lineal vs tendencia amortiguada



***Fuente**: Forecasting: Principles and Practice*

2.3.4 Método estacional de Holt-Winters

El método estacional de Holt-Winters comprende la ecuación de pronóstico y tres ecuaciones de suavizado: una para el nivel $\boldsymbol{\ell_t}$, una para la tendencia $\boldsymbol{b_t}$ y una para el componente estacional $\boldsymbol{S_t}$, con el suavizador de parámetros $\boldsymbol{\alpha}$, $\boldsymbol{\beta^*}$ y $\boldsymbol{\gamma}$. Aquí se utiliza $\boldsymbol{m}$ para indicar la frecuencia de la estacionalidad.

Existen dos variaciones a este método: **El método aditivo** cuando las variaciones estacionales son aproximadamente constantes a lo largo de la serie, con la siguiente forma del componente:

$$\hat{y}_{t+h|t} = \ell_t + hb_t + s_{t+h-m(k+1)}$$

$$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$$
$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$$
$$s_t = \gamma(y_t - \ell_{t-1} + b_{t-1}) + (1-\gamma)S_{t-m},$$

El método multiplicativo cuando las variaciones estacionales cambian proporcionalmente al nivel de la serie, con la siguiente forma del componente:

$$\hat{y}_{t+h|t} = (\ell_t + hb_t)s_{t+h-m(k+1)}$$
$$\ell_t = \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1})$$
$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$$
$$s_t = \gamma\frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1-\gamma)S_{t-m}$$

2.3.5 Método amortiguado de Holt-Winters

La amortiguación es posible con los métodos de Holt-Winters aditivos y multiplicativos. Un método que a menudo proporciona pronósticos precisos y robustos para datos estacionales es el método de Holt-Winters con una tendencia amortiguada y estacionalidad multiplicativa:

$$\hat{y}_{t+h|t} = [\ell_t + (\phi + \phi^2 + \cdots + \phi^h)b_t]s_{t+h-m(k+1)}$$

$$\ell_t = \alpha(\frac{y_t}{s_{t-m}}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$$

$$s_t = \gamma \frac{y_t}{(\ell_{t-1} + \phi b_{t-1})} + (1-\gamma)S_{t-m}$$

2.3.6 Predicciones en modelos de Suavizado Exponencial (ETS)

Existen más variaciones respecto a los métodos de suavizado exponencial descritos en los apartados anteriores, por las variaciones en las combinaciones de la tendencia y los componentes estacionales, son posibles nueve métodos.

Tabla 1.Clasificación de métodos de suavizado exponencial

Trend Component	Seasonal Component		
	N (None)	A (Additive)	M (Multiplicative)
N (None)	(N,N)	(N,A)	(N,M)
A (Additive)	(A,N)	(A,A)	(A,M)
Add (Additive damped)	(Add,N)	(Add,A)	(Add,M)

***Fuente**: Elaboración propia*

Como se puede observar en la Tabla 1, cada método está etiquetado por un par de letras (T, S) que definen el tipo de componentes "Tendencia" y "Estacional".

Para cada método existen dos modelos: uno con errores aditivos y otro con errores multiplicativos. Es así, que para distinguir entre un modelo con errores aditivos y uno con errores multiplicativos (y también para distinguir los modelos de los métodos), agregamos una tercera letra (E) a la clasificación de la Tabla 1.

Etiquetamos cada modelo como ETS (·,·,··,·,·) para (Error, Tendencia, Estacionalidad). Usando la misma notación que en la Tabla 1, las posibilidades para cada componente son Error = $\{A, M\}$, Tendencia = $\{N, A, A_d\}$ y Estacionalidad = $\{N, A, M\}$.

La tabla 2 presenta las ecuaciones para todos los modelos en el marco de ETS.

Tabla 2. Ecuaciones para cada uno de los modelos ETS.

ADDITIVE ERROR MODELS

Trend	Seasonal		
	N	A	M
N	$y_t = \ell_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$	$y_t = \ell_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = \ell_{t-1}s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/\ell_{t-1}$
A	$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + b_{t-1})$
A_d	$y_t = \ell_{t-1} + \phi b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = \phi b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + \phi b_{t-1})$

MULTIPLICATIVE ERROR MODELS

Trend	Seasonal		
	N	A	M
N	$y_t = \ell_{t-1}(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$	$y_t = (\ell_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \alpha(\ell_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + s_{t-m})\varepsilon_t$	$y_t = \ell_{t-1}s_{t-m}(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$
A	$y_t = (\ell_{t-1} + b_{t-1})(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m}(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$
A_d	$y_t = (\ell_{t-1} + \phi b_{t-1})(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m}(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$

***Fuente**: Forecasting: Principles and Practice*

En este sentido 4 posibles modelos serían los siguientes:

- ETS (A, N, N): suavizado exponencial simple con errores aditivos
- ETS (M, N, N): suavizado exponencial simple con errores multiplicativos
- ETS (A, A, N): método lineal de Holt con errores aditivos
- ETS (M, A, N): método lineal de Holt con errores multiplicativos

2.4 Modelos ARIMA

Box y Jenkins (1970) desarrollaron modelos estadísticos para series temporales que pretenden describir las autocorrelaciones en los datos. Estos modelos son llamados ARIMA, que deriva de sus tres componentes Autorregresivo, Integrado y Medias Móviles.

2.4.1 Modelos Autorregresivos AR(p)

En un modelo Autorregresivo AR ($\boldsymbol{p}$) pronosticamos la variable de interés utilizando una combinación lineal de valores pasados de la variable más un término de error, se puede escribir como:

$$\boldsymbol{y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t,}$$

donde $\boldsymbol{\varepsilon_t}$ es un ruido blanco. Normalmente restringimos los modelos autorregresivos a datos estacionarios, con algunas restricciones en los valores de los parámetros:

- Para un modelo AR (1): $\boldsymbol{-1 < \phi_1 < 1}$.
- Para un modelo AR (2): $\boldsymbol{-1 < \phi_2 < 1, \phi_1 + \phi_2 < 1, \phi_2 - \phi_1 < 1}$.
- Cuando $\boldsymbol{p \geq 3}$, las restricciones son mucho más complicadas

2.4.2 Modelos de Medias Móviles MA(q)

Un modelo de Media Móvil MA(q) es aquel que usa errores de pronóstico pasados en un modo similar a una regresión, expresado como:

$$\boldsymbol{y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q,}}$$

No todos los MA(q) son invertibles, por lo cual se tienen las siguientes restricciones:

- Para un modelo MA (1): $\boldsymbol{-1 < \phi_1 < 1}$.
- Para un modelo MA (2): $\boldsymbol{-1 < \phi_2 < 1, \phi_2 + \phi_1 < -1, \phi_1 - \phi_2 < 1}$.
- Cuando $\boldsymbol{q \geq 3}$, las restricciones son mucho más complicadas

2.4.3 Modelos no estacionales ARIMA

Un modelo ARIMA $\boldsymbol{(p, d, q)}$ se puede escribir como:

$$\boldsymbol{Y_t = c + \phi_1 y_{d\ t-1} + \cdots + \phi_p y_{d\ t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q,} + \varepsilon_t,}$$

donde $\boldsymbol{y_d}$ es $\boldsymbol{Y}$ diferenciada $\boldsymbol{d}$ veces y $\boldsymbol{c}$ es una constante.

El modelo se especifica bajo los siguientes parámetros de orden:

p = orden de la parte autorregresiva;

d = grado de diferenciación en el componente integrado.

q = orden de la parte media móvil.

Las mismas condiciones de estacionariedad e invertibilidad que se utilizan para los modelos de media móvil y autorregresivos también se aplican a un modelo ARIMA.

2.4.4 Modelos estacionales ARIMA

Un modelo estacional ARIMA $(\boldsymbol{p,d,q})\ (\boldsymbol{P,D,Q})_{\boldsymbol{m}}$ se forma al incluir términos estacionales adicionales en los modelos ARIMA que hemos visto hasta ahora. Está escrito de la siguiente manera:

$$\text{ARIMA} \qquad \underbrace{(\boldsymbol{p,d,q})}_{\text{Parte no estacional del modelo}} \qquad \underbrace{(\boldsymbol{P,D,Q})_{\boldsymbol{m}}}_{\text{Parte estacional del modelo}}$$

donde **m** es el número de observaciones por año. Por ejemplo, un ARIMA (1,1,1) $(1,1,1)_4$, sin una constante es para datos trimestrales ($\boldsymbol{m = 4}$) y puede ser escrito como:

$$(\mathbf{1} - \boldsymbol{\phi_1 B})\ (\mathbf{1} - \boldsymbol{\Phi_1 B^4})(\mathbf{1} - \boldsymbol{B})(\mathbf{1} - \boldsymbol{B^4})\boldsymbol{y_t} = (\mathbf{1} + \boldsymbol{\theta_1 B})\ (\mathbf{1} + \boldsymbol{\Theta_1 B^4})\boldsymbol{\varepsilon_t}$$

Los términos estacionales adicionales simplemente se multiplican por los términos no estacionales.

2.4.5 Predicción en modelos ARIMA

Para realizar predicciones mediante modelos ARIMA $(\boldsymbol{p,d,q})$ es necesario seguir los siguientes pasos generales:

- **Examinar los datos de la serie temporal. -** Detectar patrones irregulares o valores atípicos para su limpieza, tomar logaritmos para ayuda a estabilizar la serie, etc.
- **Descomponer la serie. -** Aquí se extraen sus componentes: estacional, tendencia y ciclo. También se puede tratar de eliminar el componente estacional de una serie.
- **Estacionariedad. -** Un modelo ARIMA requiere que la serie sea estacionaria**,** se puede realizar a prueba aumentada de Dickey-Fuller (ADF) para saberlo. En caso de no serlo se realizar la diferenciación.
- **Parámetros óptimos. -** Los parámetros p, d, q se pueden encontrar usando gráficos ACF y PACF.
- **Ajuste de modelos.** – Con los parámetros ya establecidos se construye los posibles modelos. Se utilizan los criterios de información de Akaike (AIC) y los criterios de información de Baysian (BIC) para elegir los mejores modelos.

- **Realizar predicciones.** - Una vez encontrados los mejores modelos se pueden realizar las predicciones.

Para realizar predicciones con modelos ARIMA $(\boldsymbol{p, d, q})$ $(\boldsymbol{P, D, Q})_{\boldsymbol{m}}$ el procedimiento es el mismo, solo basta con incluir el componente estacional en el modelo.

2.5 Métodos de evaluación

Existen varios tipos de métricas de evaluación para series temporales, las 2 más utilizadas son:

- **Error absoluto medio (MAE):** Que mide la magnitud promedio de los errores en un conjunto de predicciones, sin considerar su dirección. Es el promedio sobre la muestra de prueba de las diferencias absolutas entre la predicción y la observación real donde todas las diferencias individuales tienen el mismo peso.
- **Error cuadrático medio (RMSE):** Que es una regla de puntuación cuadrática que también mide la magnitud promedio del error. Es la raíz cuadrada del promedio de las diferencias cuadradas entre la predicción y la observación real.

Sin embargo, para el problema de series temporales planteado por Corporación La Favorita la métrica de evaluación se hizo a través del **Error logarítmico cuadrático medio ponderado normalizado (NWRMSLE),** que se calcula así:

$$\boldsymbol{NWRMSLE} = \sqrt{\frac{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{w_i} \left(\boldsymbol{ln}(\hat{\boldsymbol{y}}_{\boldsymbol{i}} + 1) - \boldsymbol{ln}(\boldsymbol{y_i} + 1)\right)^2}{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{w_i}}}$$

Donde para cada i, $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ es la predicción de un elemento predicho, mientras que $\boldsymbol{y_i}$ es el elemento actual. $\boldsymbol{w_i}$ son los pesos, en este problema se tienen 2 tipos de pesos: 1.25 para artículos perecederos y 1.00 para no perecederos.

Esta métrica evita penalizar grandes diferencias en la predicción cuando tanto el número predicho como el verdadero son grandes: por ejemplo cuando $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$=5 y el valor $\boldsymbol{y_i}$=50 se penaliza más que cuando $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$=500 y el valor $\boldsymbol{y_i}$=545.

Los resultados se suben a la página del concurso en Kaggle y automáticamente se calcula el NWRMSLE. Se reciben 2 puntuaciones (basadas en el error), una dominada como **publica** que utilizada el 31% de los datos de test y la otra denominada **privada,** que utiliza el 69% de esos datos restantes, siendo esta última la determinante de la puntuación final.

3 Análisis del problema

Estamos ante un problema de series temporales complejo debido a la cantidad de artículos que se tienen que predecir. Como se vio en el apartado anterior hay varios métodos que podríamos aplicar para tratar de solucionar el problema.

Sin embargo, antes de comenzar a elaborar modelos es importante realizar un análisis previo de todos los datos disponibles.

3.1 Origen de los datos

En esta competencia, se estará prediciendo las ventas de unidades por miles de artículos vendidos en diferentes tiendas de La Favorita ubicadas en Ecuador. Los datos están divididos en varios conjuntos de datos. Las descripciones de archivos e información es la siguiente:

- **train.csv -** Datos de entrenamiento, tienen la estructura de la tabla 3.

Tabla 3.Estructura datos entrenamiento

	id	date	store_nbr	item_nbr	unit_sales	OnPromotion
1	86672216	2016-08-01	1	103520	3.000	FALSE
2	86672217	2016-08-01	1	103665	2.000	FALSE
3	86672218	2016-08-01	1	105574	7.000	FALSE
4	86672219	2016-08-01	1	105575	13.000	FALSE
5	86672220	2016-08-01	1	105577	2.000	FALSE
6	86672221	2016-08-01	1	105693	1.000	FALSE
7	86672222	2016-08-01	1	105737	7.000	FALSE
8	86672223	2016-08-01	1	105857	5.000	FALSE

Showing 1 to 8 of 38,824,824 entries

***Fuente**: Elaboración propia*

id. - Es un identificador único para etiquetar los datos.

date. – Contiene la fecha en que se realiza cada transacción por tienda y articulo.

store_nbr. – Contiene la información de la tienda en que se realiza una transacción en una fecha específica, son 54 tiendas en total.

item_nbr. – Contiene la información del artículo vendido en una fecha específica, son 4036 artículos en total que se tienen en el conjunto de datos.

unit_sales. – Contiene la información de las ventas unitarias por cada combinación articulo tienda. Los valores negativos de ventas representan devoluciones. Son 170909 combinaciones por cada artículo/ tienda vendido.

onpromotion. – Contiene la información sobre las promociones, el valor TRUE quiere decir que hubo una promoción para ese artículo en una tienda y fecha determinada. Aproximadamente el 16% de los datos de promociones son valores perdidos.

Los datos de entrenamiento no incluyen filas para los artículos que tenían cero ventas por unidad para una combinación de tienda / fecha. No hay información sobre si el artículo estaba o no disponible para la tienda en la fecha.

Aunque se tienen datos desde el año 2013, solo se han tomado en cuenta los datos del último año antes de la predicción. Es decir, datos desde el 01-08-2016 hasta el 15-08-2017.

- **test.csv -** Datos de prueba, tiene la estructura de la tabla 4.

Tabla 4. Estructura de los datos de test

	id	date	store_nbr	item_nbr	onpromotion
1	125497040	2017-08-16	1	96995	FALSE
2	125497041	2017-08-16	1	99197	FALSE
3	125497042	2017-08-16	1	103501	FALSE
4	125497043	2017-08-16	1	103520	FALSE
5	125497044	2017-08-16	1	103665	FALSE
6	125497045	2017-08-16	1	105574	FALSE
7	125497046	2017-08-16	1	105575	FALSE
8	125497047	2017-08-16	1	105576	FALSE
9	125497048	2017-08-16	1	105577	FALSE

Showing 1 to 9 of 3,370,464 entries

***Fuente**: Elaboración propia*

id. - Es un identificador único para etiquetar los datos.

date. – Contiene la fecha en que se tiene que predecir cada venta por tienda y articulo.

store_nbr. – Contiene la información de la tienda donde se tiene que realizar la venta.

item_nbr. – Contiene la información del artículo vendido en las fechas de predicción.

onpromotion. – Contiene la información sobre las promociones, para cada artículo- tienda en la fecha determinada de predicción.

La estructura de los datos de test es la misma que los datos de entrenamiento, con la diferencia que hace falta la variable **unit_sales** que es la variable a predecir. Las predicciones van desde el 16-08-2017 hasta el 31-08-2017. Es decir, un periodo de predicción de 16 días.

Los datos de prueba tienen una pequeña cantidad de elementos que no están contenidos en los datos de entrenamiento. Parte del ejercicio será predecir las ventas de un nuevo artículo en base a productos similares.

- **sample_submission.csv -** Un archivo de envío de muestra en el formato correcto, para subir en la plataforma de Kaggle.
- **stores.csv –** Contiene los metadatos, incluyendo ciudad, provincia y clúster. Clúster es una agrupación de tiendas similares.
- **items.csv** – Contiene metadatos de elementos, incluidos familia, clase y perecibles. Los artículos marcados como perecible tienen un peso de 1,25; de lo contrario, el peso es 1.0.
- **transactions.csv** - El recuento de transacciones de venta para combinación fecha y tienda. Solo incluido para el marco de tiempo de los datos de entrenamiento.
- **oil.csv -** Precio diario del petróleo. Incluye valores durante el período de tiempo entrenamiento y test de datos. Ecuador es un país dependiente del petróleo y su salud económica es altamente vulnerable a los choques en los precios del petróleo.
- **holidays_events.csv -** Vacaciones y eventos, con metadatos. En ecuador existe una ley de feriados, por lo cual no se tienen libre el día de la fecha importante, sino que es trasferido al fin de semana.

3.2 Análisis Exploratorio de Datos

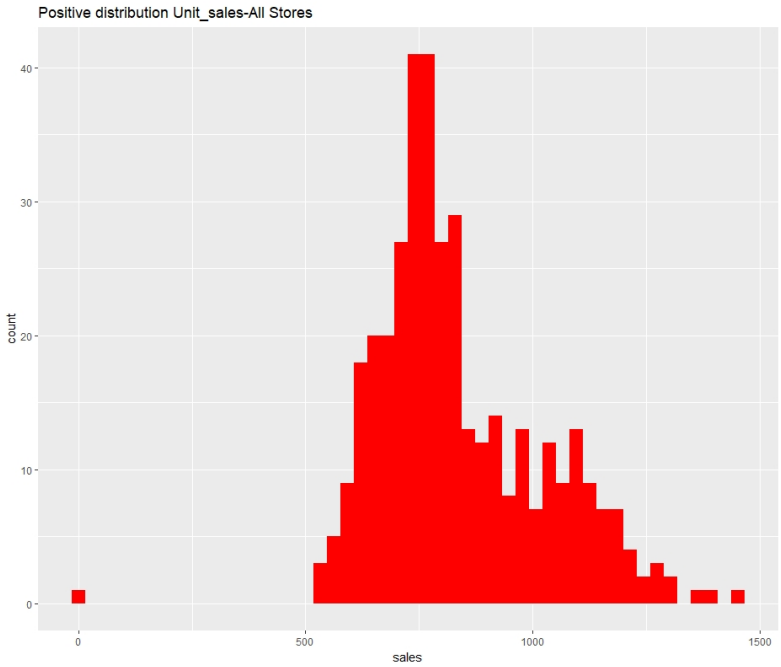
El Análisis Exploratorio de Datos, en adelante EDA; engloba un conjunto de técnicas que permiten comprender de mejor manera la naturaleza de una colección de datos.

El objetivo del EDA es resumir y visualizar datos de manera que se facilite la identificación de patrones o tendencias que subyacen ocultos en una colección de datos.

Este problema de series temporales contiene 126 millones de registros con datos diarios desde el 01 de enero del 2013 hasta el 15 de agosto del 2017, por lo cual se trabajarán con los datos del último año. En este sentido resulta aún más importante visualizar y analizar los datos antes de comenzar a elaborar los modelos que realicen los pronósticos de ventas.

El primer lugar, se analiza la composición de los datos de entrenamiento haciendo énfasis en las ventas, que es la variable que más interesa.

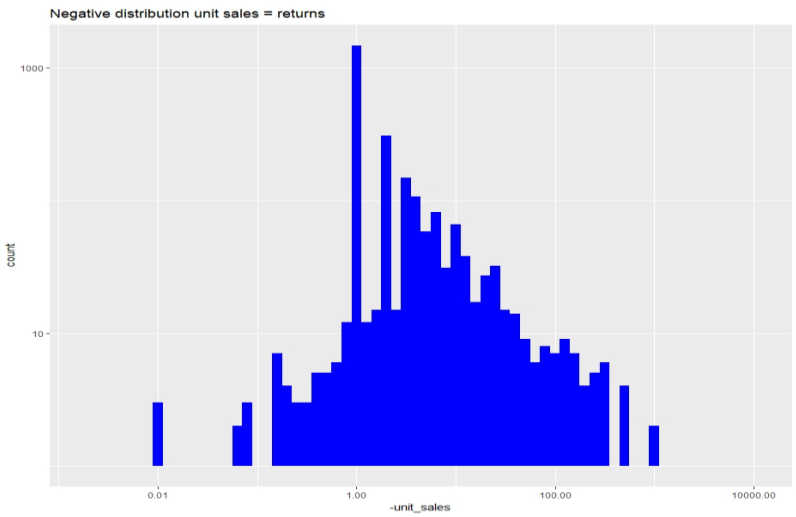
Figura 8. Distribución del total de ventas



***Fuente**: Elaboración propia*

En la figura 8, las ventas diarias muestran una distribución multimodal con un pico máximo de unas 700.000 unidades, este comportamiento de las ventas puede deberse a que los datos tienen números enteros y flotantes de acuerdo a cada tipo de producto.

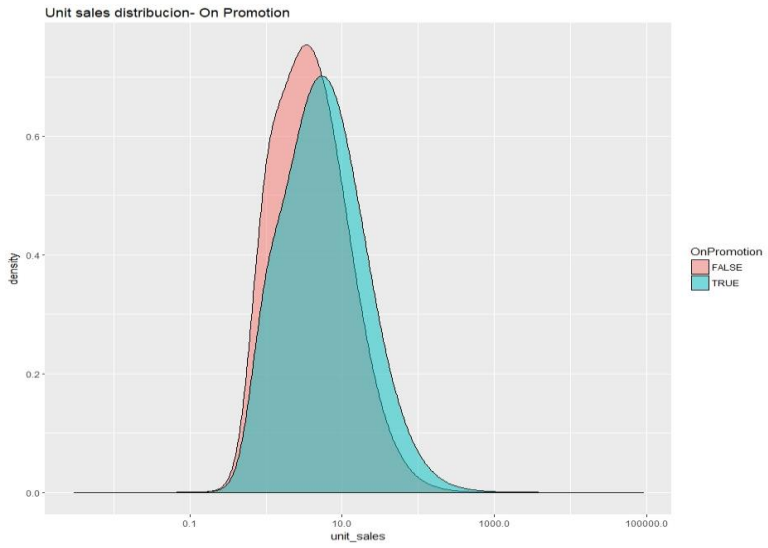
Figura 9. Devoluciones en ventas



***Fuente**: Elaboración propia*

Las devoluciones que se muestran en la Figura 9 son de pocas unidades, lo máximo son aproximadamente unas 100 unidades. Este aspecto resulta positivo, debido a que se comercializan miles de productos diferentes y se tienen un porcentaje de devoluciones mínimo.

Figura 10. Distribución de ventas en relación a las promociones

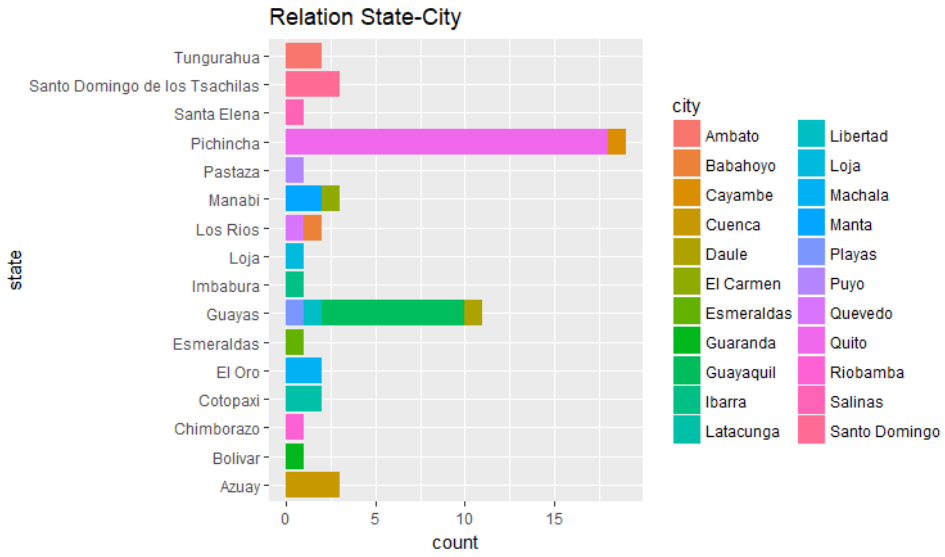


***Fuente**: Elaboración propia*

La figura 10 nos muestra que cuando hay productos en promoción las ventas son mayores que cuando no existe ningún tipo de promoción. Sin embargo, la diferencia es muy pequeña y podría no ser significativa.

Así mismo Corporacion La Favorita tiene tiendas en todo el pais. En la figura 11 se analiza la manera en que las tiendas están distribuidas.

Figura 11. Ubicación de las tiendas por Provincia y ciudad

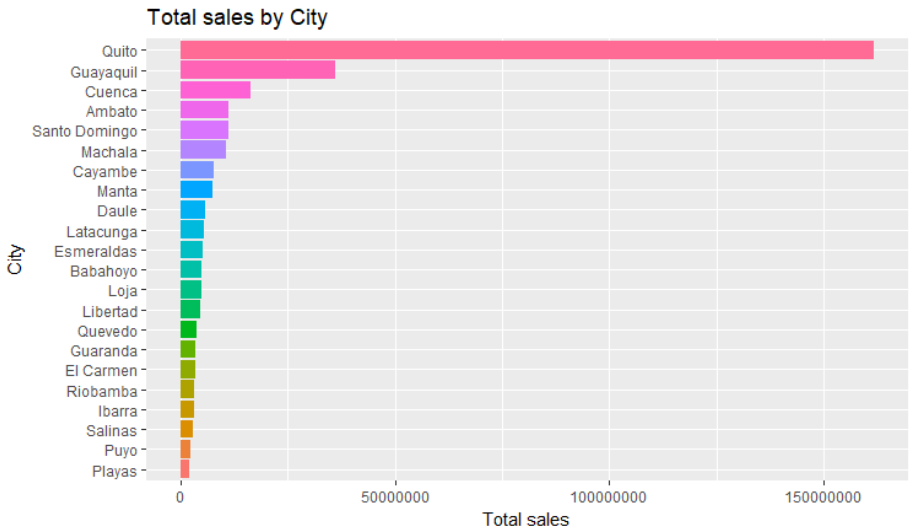


***Fuente**: Elaboración propia*

En Quito que es la capital del Ecuador es donde se concentra el mayor número de tiendas con un total de 18 tiendas, seguido por Guayaquil la segunda ciudad más importante del país con 8 tiendas.

El alcance que tienen las tiendas a nivel nacional es muy grande, puesto que están en 16 provincias con sus respectivas capitales. Cabe mencionar que el ecuador tiene 24 provincias.

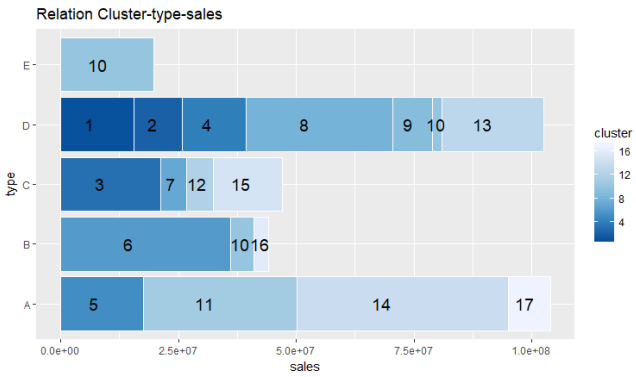
Figura 12. Ranking de ventas por ciudades



***Fuente**: Elaboración propia*

La figura 12 muestra la evidente diferencia de ventas que existe en Quito respecto al resto del país.

Figura 13. Relación de ventas, tipo de tienda y clúster



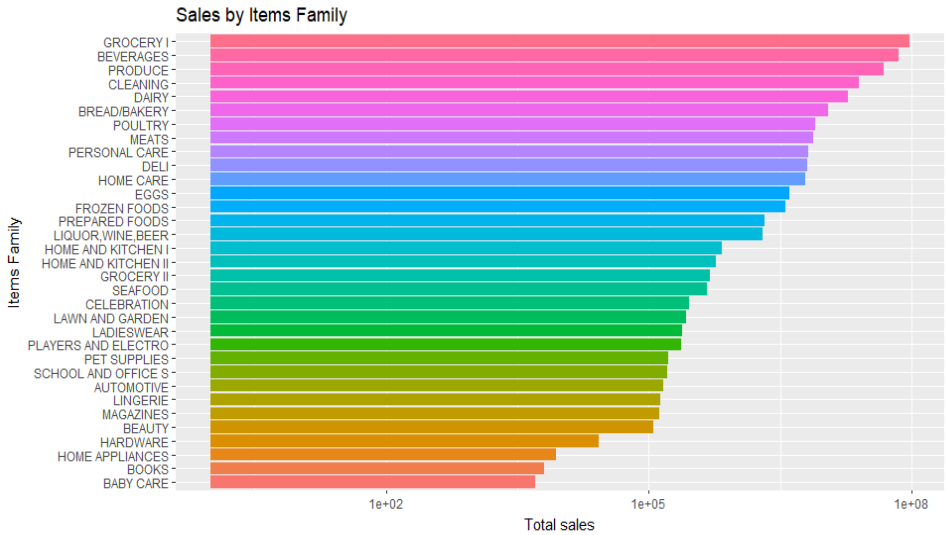
***Fuente**: Elaboración propia*

En la Figura 13 se puede visualizar que existen varios tipos de tiendas, A, B, C, D, E. Las tiendas de tipo A y D son las que tienen las ventas más altas, lo que a su vez indica que se encuentran en las principales ciudades como Quito, Guayaquil, Ambato, Cuenca, Manta y Santo Domingo.

Así mismo, existe una relación entre los tipos de tiendas y una formación de 17 clúster, curiosamente se aprecia que solo el clúster 10 se repite en 3 de los 5 tipos de tiendas. El tamaño del clúster varía de acuerdo al número de tipo de tiendas que agrupe.

Si bien es muy importante conocer las ventas que realizan las tiendas, es fundamental conocer la composición de las ventas que tienen los miles de productos que tiene en sus estanterías La Favorita.

Figura 14. Ranking de ventas de productos por familia



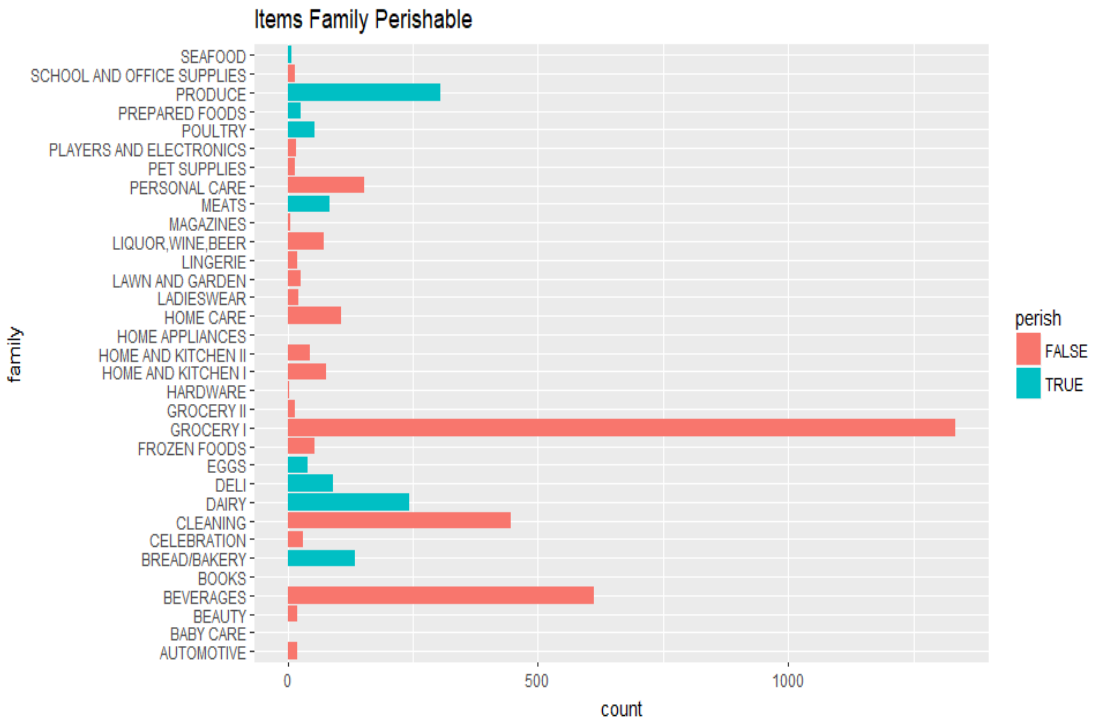
***Fuente**: Elaboración propia*

En la figura 14 gracias a los metados podemos ver como los productos se encuetran calificados por una variable llamada familia.

Según esta clasificación las 6 familias de productos más vendidos son las siguientes:

- Abarrotes, por ejemplo aceites, cereales, fideos, etc.
- Bebidas
- Produce, en esta variable podrian estar frutas, verduras, entre otros productos.
- Limpieza
- Lechería
- Pan / panadería

Figura 15. Clasificación de productos perecibles



***Fuente**: Elaboración propia*

Es importante mencionar que hay 33 tipos de familias de productos. En las figuras 14 y 15 se pueden apreciar todos los tipos de familia.

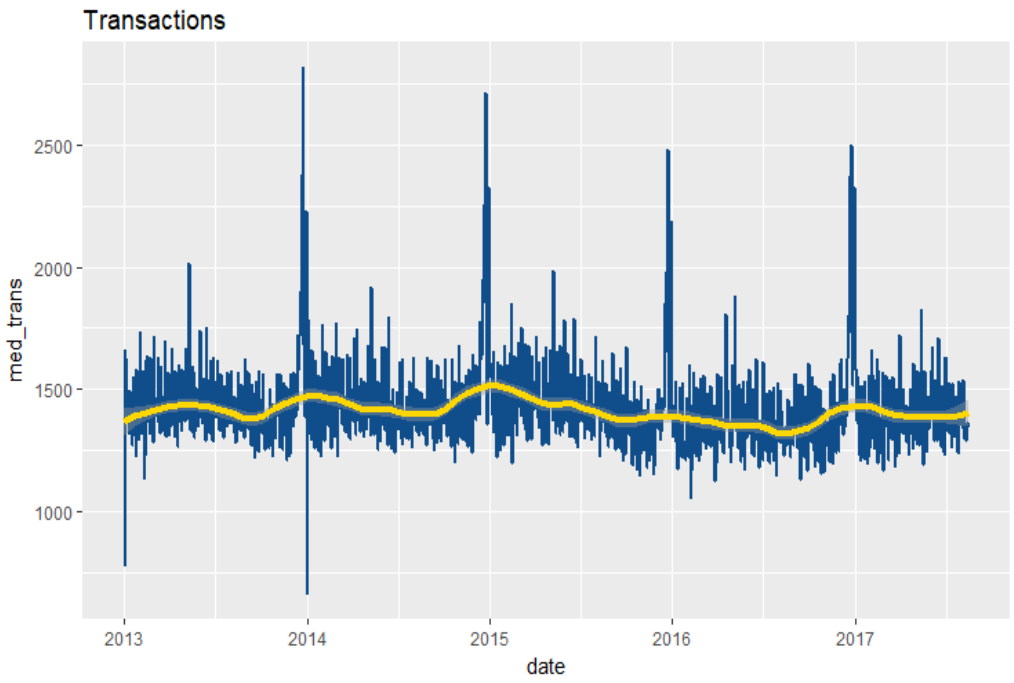
Sin embargo, aquí se evidencia también la cantidad de artículos que tiene cada tipo de familia y también si es o no perecible.

Las 6 familias con mayor variedad de productos, tiene aproximadamente más de 200 productos diferentes y son las mismas que obtienen más ventas.

De igual manera, se visualiza que de los 6 productos más vendidos 4 no son perecibles. Esta es otra manera de entender porque el número de devoluciones es ínfimo en relación al total de ventas.

Una manera de medir el comportamiento de las ventas es analizar el número de transacciones que se realizan, tal como muestra la figura 16.

Figura 16. Comportamiento de las transacciones



***Fuente**: Elaboración propia*

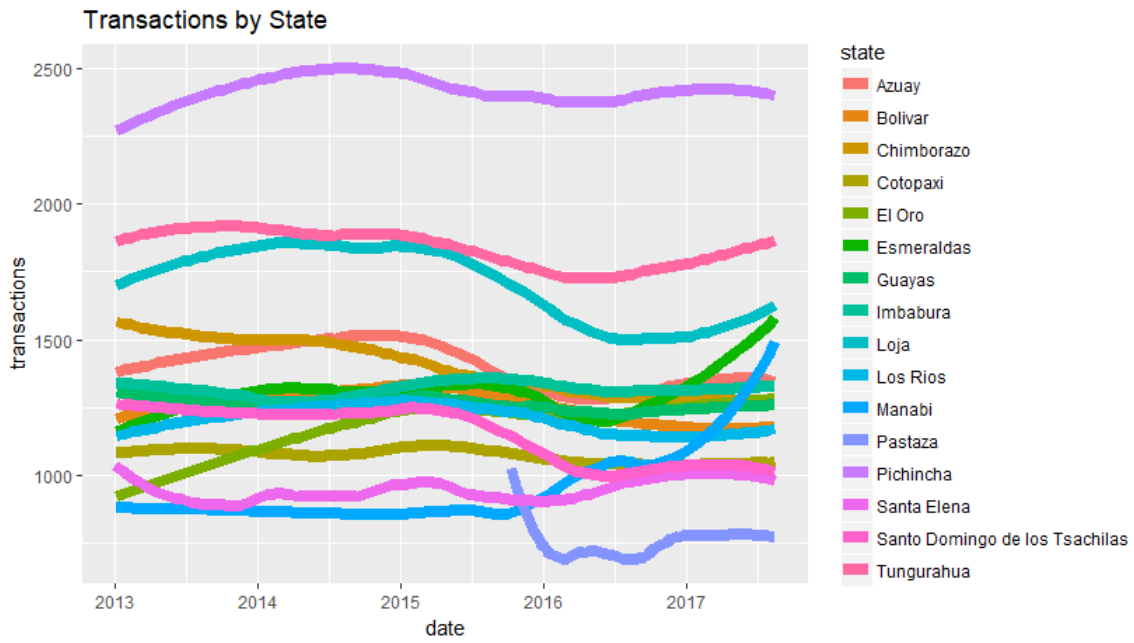
A diferencia de las ventas, que solo se han tomado los datos del último año; las transacciones si se están visualizando desde el año 2013.

La grafica presentada en modo de serie temporal indica que se tienen una media diaria en torno a las 1500 transacciones diarias, que además se mantiene constante en el horizonte temporal. Eso es un indicativo que las ventas tienen el mismo comportamiento.

A simple vista también es evidente que el volumen de transacciones tiene un pico extremadamente alto al final de cada año, lo cual está relacionado con la época navideña que es la temporada en donde en la mayoría de empresas las ventas crecen exponencialmente. Sobre todo, si se toma en cuenta que Corporación La Favorita tiene tiendas de autoservicios a nivel nacional.

Por lo antes mencionado la gráfica 16. muestra un comportamiento normal, dado que mantiene una media estable. Además, como los datos se encuentran a nivel diario lo habitual es que se aprecie una varianza un poco alta.

Figura 17. Transacciones por Provincias



***Fuente**: Elaboración propia*

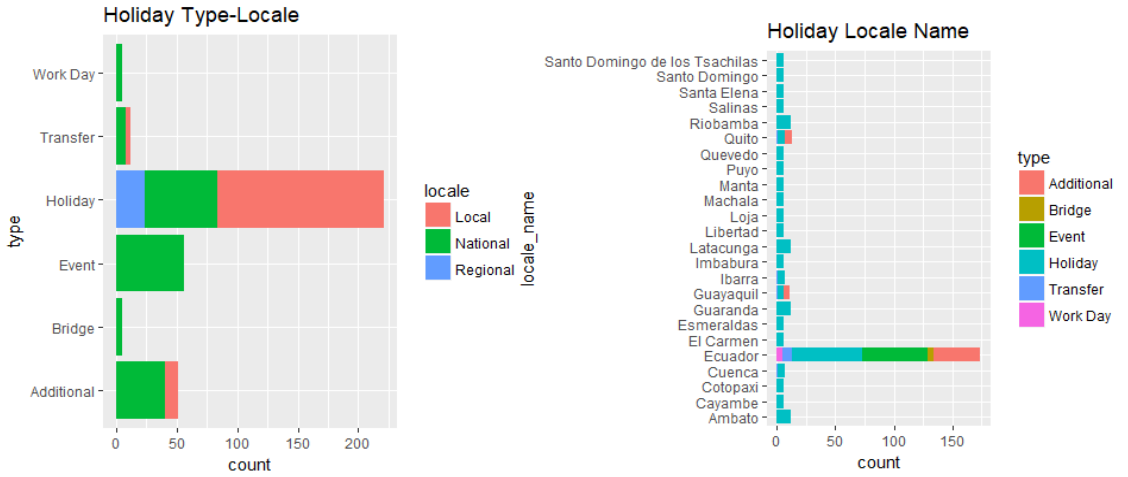
La figura 17 muestra la cantidad de transacciones que se realizan en cada Provincia, que muestra diferenciación de las tiendas en la Provincia de Pichincha, con sus tiendas más importantes en Quito.

Además, hay varios aspectos que destacar que se mencionan a continuación:

- La figura muestra que el año 2015 se apertura la tienda de Pastaza, que comenzó con un buen volumen de transacciones que a partir del año 2016 se mantienen como las más bajas.
- El número de transacciones en Manabí creció exponencialmente a partir del año 2016. Su explicación radica en que el abril del 2016 hubo un terremoto de 7.8 en la escala sismológica de Richter, que afectó principalmente a esta Provincia en las que hay varias tiendas de la Favorita. Además, el gobierno ecuatoriano realizó importantes inversiones en la compras de víveres para los afectados por el terremoto.
- De igual manera se ve un crecimiento exponencial, pero de menor dimensión en la Provincia de Esmeraldas que fue la segunda con mayor afectación luego del terremoto de abril del 2016.

Los días festivos son muy importantes porque en el Ecuador existe una ley de feriados nacionales, que cada vez que hay una festividad de carácter importante si cae en mitad de semana es transferida al fin de semana para promover el turismo.

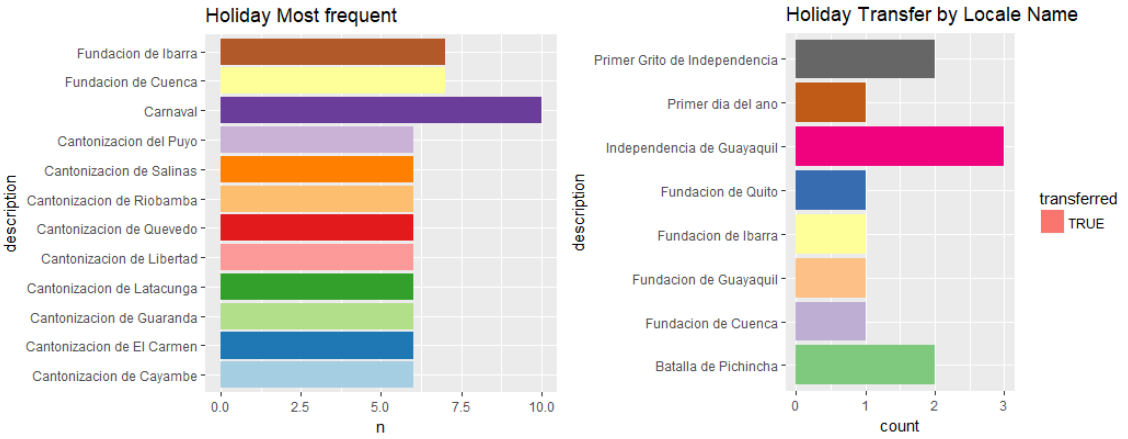
Figura 18. Composición de días festivos



***Fuente**: Elaboración propia*

El lado izquierdo de la figura 18, muestra la composición del tipo de días festivos o vacacionales los cuales son de tipo nacional, regional y local. Mientras que el lado derecho muestra además una relación en torno a las localidades que los componen. Solo Quito y Guayaquil tienen más de un tipo de feriados.

Figura 19. Recurrencia de días festivos



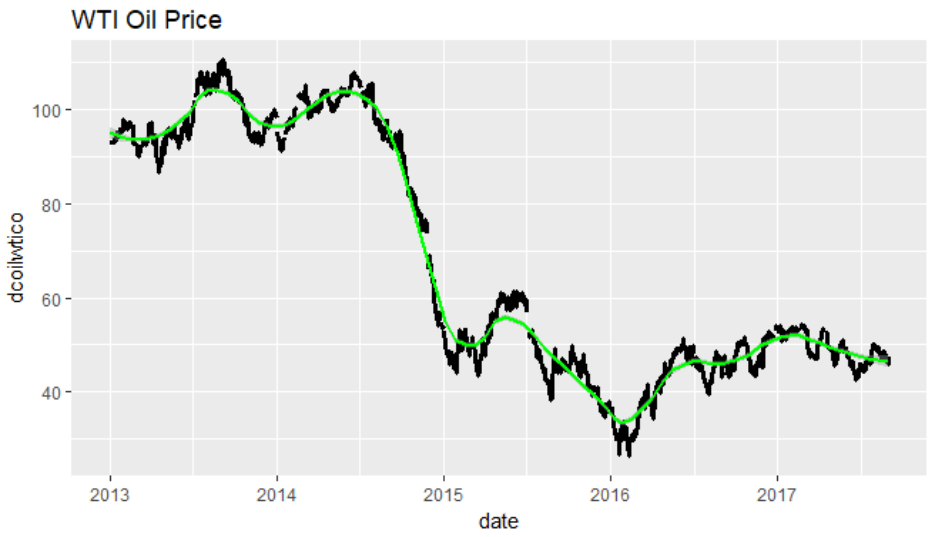
***Fuente**: Elaboración propia*

La figura 19 muestra los días con más recurrencia en los feriados, el carnaval es de gran importancia. Así mismo tenemos los días festivos que tienen como fecha de celebración un día entre semana, pero que han sido transferido al fin de semana. Por ejemplo, la Batalla del Pichincha celebrada el 24 de mayo, es un día que no se labora en el Ecuador; si esa fecha es un día miércoles el día vacacional es transferido al día viernes.

Ecuador es un país con una economía que tiene en el petróleo una de las principales fuentes de ingresos, la base de datos incluye los precios del petróleo desde el 2013 al 2017.

La figura 20 muestra que los precios del petróleo cayeron de manera significativa en el año 2015, por lo cual sería interesante saber si esto afectó a las tiendas de la Favorita.

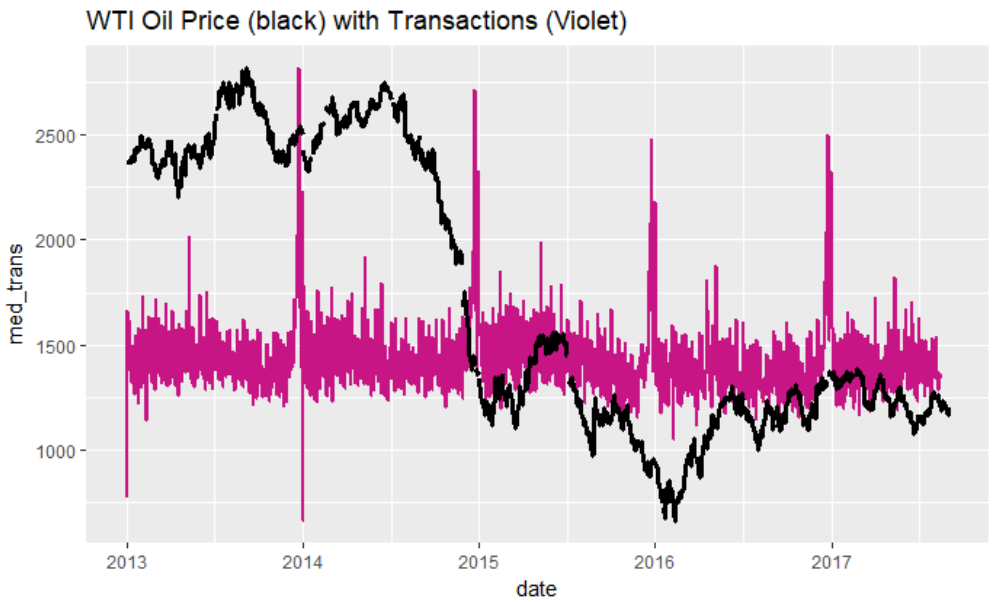
Figura 20. Precio internacional del petróleo ecuatoriano



***Fuente**: Elaboración propia*

En la figura 21 se puede apreciar que la caída de los precios del petróleo no afectó al nivel de transacciones que se realizan diariamente en las tiendas de La Favorita.

Figura 21. Precio del petróleo vs transacciones de tiendas

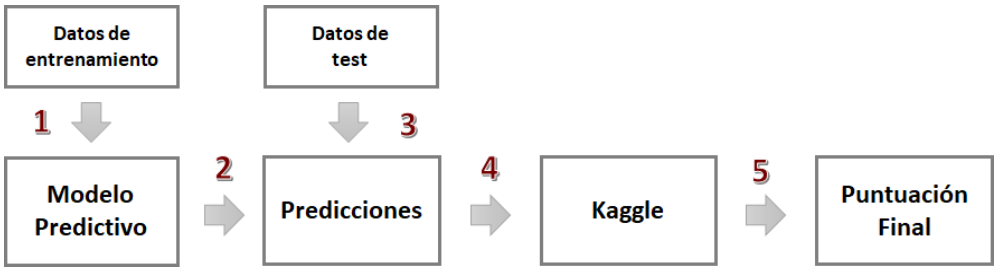


***Fuente**: Elaboración propia*

4 Elaboración de predicciones

En este apartado se describe el proceso de construcción de los diferentes modelos predictivos. La utilización de los datos puede diferir entre los diferentes algoritmos, pero la estructura en que se realizaran las predicciones será la misma para todos los modelos. Figura 22 muestra dicha estructura.

Figura 22.Estructura de modelos predictivos



***Fuente**: Elaboración propia*

4.1 Media histórica

Este modelo al ser muy básico será la base para tomar una dirección respecto a los modelos que se construirán más adelante.

Los datos de entrenamiento contienen las siguientes variables: Id, date, store_nbr, item_nbr, sales, onpromotion.

El periodo de predicción es desde el 16-08-2017 hasta el 31-08-2017, es decir 16 días. Los datos de entrenamiento para este modelo serán de un año aproximadamente, desde el 01-08-2016 hasta el 15-08-2017.

En este modelo se utilizarán 3 variables:

- store_nbr
- item_nbr
- sales.

Se creará una nueva variable con el nombre de **med_sales** que es igual a la media de histórica de ventas por cada combinación tienda y artículo.

Antes de realizar este cálculo los valores negativos que corresponden a devoluciones serán reemplazados por ceros.

Tabla 5. Media histórica de ventas

Filter

	item_nbr	store_nbr	med_sales
1	96995	1	1.440000
2	96995	2	1.272727
3	96995	3	1.212766
4	96995	4	1.142857
5	96995	5	1.705882
6	96995	6	1.476190
7	96995	7	1.227273
8	96995	8	1.547945
9	96995	25	1.071429
10	96995	27	1.117647

Showing 1 to 10 of 170,909 entries

***Fuente**: Elaboración propia*

La tabla 5 muestra cómo se produjeron 170909 observaciones por cada combinación artículo/tienda, las cuales se fusionarán con los datos de **test.**

Los datos de test contienen 3370464 observaciones, luego de la fusión con la media histórica calculada se generaron 1766944 valores perdidos.

El formato de subida de datos de Kaggle no acepta valores perdidos, solo valores ≥0.

Los valores perdidos corresponden a los artículos que no se encontraban en los datos de entrenamiento y que había que predecir.

Estos valores perdidos fueron reemplazados mediante diferentes métodos:

- Mediante la media total de ventas
- Con valores igual a uno
- Con valores igual a cero

La última opción fue la más eficiente porque tendía a sobreestimar menos.

En la tabla 6 se muestran las puntuaciones del modelo, una vez que fueron subidos a la plataforma de Kaggle.

Tabla 6. Puntuación Media Histórica

Submission and Description	Private Score	Public Score
Mean Histor (version 1/1) a month ago by augusmendoza From "Mean Histor" Script	0.807	0.819

***Fuente**: Captura de pantalla de Kaggle*

4.2 ARIMA

Es necesario indicar que se modelaron varios modelos ARIMA con la finalidad de buscar el más óptimo que obtenga la mejor puntuación en la plataforma de Kaggle.

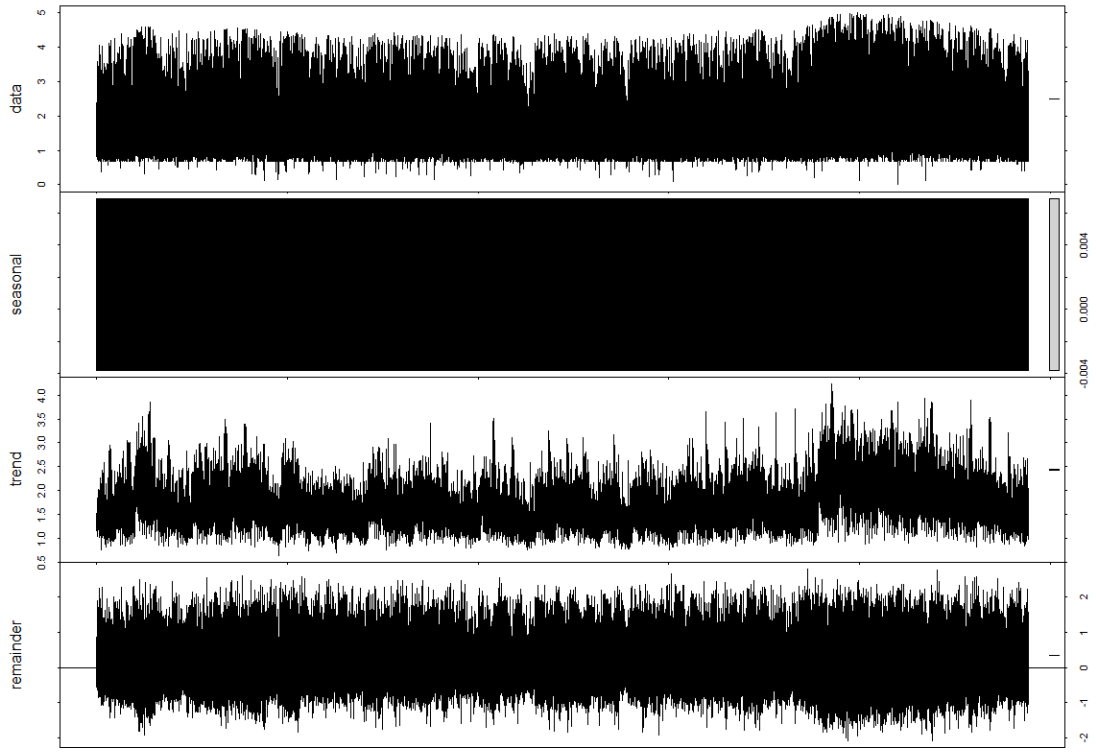
Los datos de entrenamiento para este modelo van desde el 01-08-2016 hasta el 15-08-2017, obteniendo la información por cada artículo-tienda.

Se ha obtenido la media de ventas para cada combinación articulo-tienda, previamente se ha transformado a escala logarítmica los datos y se eliminaron los valores negativos.

Los datos de ventas antes mencionados se pusieron en formato de series temporales y luego fueron limpiados mediante una herramienta disponible en el lenguaje de programación que se realizaron los modelos**.**

La figura 23 muestra cómo se pasó a descomponer la serie, mediante una frecuencia semanal tomando el cuanta el periodo de compras que se realizan en los centros de autoservicios.

Figura 23. Serie temporal descompuesta



***Fuente**: Elaboración propia*

Luego se realizó la prueba aumentada de Dickey-Fuller (ADF) para medir la estacionariedad de la serie.

Figura 24. Test de Dickey-Fuller (ADF)

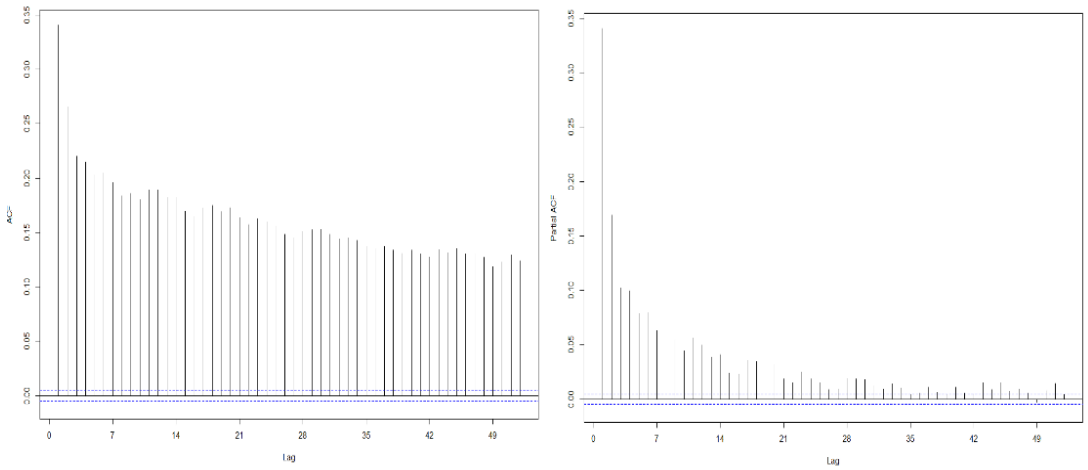
```
        Augmented Dickey-Fuller Test

data:  count_ma
Dickey-Fuller = -29.256, Lag order = 55, p-value = 0.01
alternative hypothesis: stationary
```

***Fuente**: Elaboración propia*

La figura 24 muestra que el resultado indica que la serie es estacionaria.

Es importante visualizar los gráficos de autocorrelación, para tener idea de cuál podría ser el orden del modelo. La figura 25 muestra los ACF/ PACF.

Figura 25. Gráficos de autocorrelación



***Fuente**: Elaboración propia*

Otra de las herramientas útiles del lenguaje de programación utilizado para la construcción de los modelos de series temporales es la función **auto.arima** que busca el mejor modelo **ARIMA** según los valores AIC o BIC, tal como lo muestra la figura 26.

Figura 26. Función auto.arima

```
> auto.arima(deseasonal_sales, seasonal = TRUE)
Series: deseasonal_sales
ARIMA(1,1,2)(1,0,2)[7] with drift

Coefficients:
         ar1      ma1     ma2    sar1     sma1    sma2  drift
      0.6997  -1.5018  0.5072  0.0338  -0.0133  0.0233      0
s.e.  0.0098   0.0116  0.0112  0.2332   0.2330  0.0033      0

sigma^2 estimated as 0.4046:  log likelihood=-165174.8
AIC=330365.7   AICc=330365.7   BIC=330446.1
```

***Fuente**: Elaboración propia*

La función **auto.arima** puede usarse también con la finalidad de tener un modelo base hasta encontrar el mejor modelo posible ajustando manualmente los parámetros, para obtener el AIC o BIC más bajos.

Finalmente, luego de ajustar parámetros y probar varios modelos; en la figura 27 se aprecia el mejor modelo posible que consiguió la puntuación más alta en Kaggle.

Figura 27. Modelo ARIMA optimo

```
ARIMA(5,1,7)(1,0,2)[7]

Coefficients:
          ar1      ar2     ar3     ar4     ar5      ma1     ma2      ma3      ma4     ma5     ma6     ma7     sar1    sma1     sma2
      -0.7578  -0.2804  0.3157  0.8110  0.5745  -0.0379  -0.4112  -0.6558  -0.6112  0.0822  0.5369  0.0998  -0.4512  0.4272  -0.0107
s.e.   0.0472   0.0501  0.0401  0.0505  0.0570   0.0469   0.0357   0.0365   0.0816  0.0382  0.0485  0.0046   0.1620  0.1622   0.0044

sigma^2 estimated as 0.402:  log likelihood = -164626,  aic = 329284.1
```

***Fuente**: Elaboración propia*

Es decir, un modelo ARIMA Estacional, o SARIMA.

En Kaggle fue subido mediante el nombre de Kernel Arima, obteniendo una puntuación más alta que todos los demás modelos. La tabla 7, detalla las puntuaciones:

Tabla 7. Puntuaciones Arima

Submission and Description	Private Score	Public Score
Kernel arima (version 5/5) 5 days ago by augusmendoza From "Kernel arima" Script	0.806	0.816
Arima mean (version 10/10) 5 days ago by augusmendoza From "Arima mean" Script	0.877	0.887
Modelo Arima 3 (version 2/2) 5 days ago by augusmendoza From "Modelo Arima 3" Script	0.931	0.942
Sencillo Arima (version 11/11) 5 days ago by augusmendoza From "Sencillo Arima" Script	1.307	1.338

***Fuente**: Captura de pantalla de Kaggle*

La puntuación del mejor modelo ARIMA es ligeramente superior a la del modelo de la media histórica simple, el resto de modelos ARIMA obtuvieron puntajes inferiores.

4.3 Predictor ingenuo estacional (Snaive)

La construcción de un predictor ingenuo, puede incluir la estacionalidad o no estacionalidad de los datos, este modelo si tomará en cuenta dicha característica.

La función **snaive** del lenguaje de programación utilizado devuelve los pronósticos y los intervalos de predicción de un modelo ARIMA (0,0,0) $(0,1,0)_m$ donde m es el período estacional.

La metodología para la construcción del modelo será diferente a la de los modelos ARIMA en el apartado anterior.

Se crearán 2 variables adicionales en los datos de entrenamiento, como muestra la siguiente tabla.

Tabla 8.Nuevas variables en datos de entrenamiento

id	date	store_nbr	item_nbr	unit_sales	OnPromotion	store_item_nbr	daydate
111649701	2017-04-07	1	96995	2	FALSE	1_96995	viernes
112696044	2017-04-17	1	96995	1	FALSE	1_96995	lunes
113101376	2017-04-21	1	96995	2	FALSE	1_96995	viernes
113206073	2017-04-22	1	96995	3	FALSE	1_96995	sábado
113638799	2017-04-26	1	96995	1	FALSE	1_96995	miércoles
113841982	2017-04-28	1	96995	1	FALSE	1_96995	viernes
114944159	2017-05-08	1	96995	1	FALSE	1_96995	lunes
115047931	2017-05-09	1	96995	1	FALSE	1_96995	martes

***Fuente**: Elaboración propia*

La información de las variables **ítem_nbr** y **store_nbr** se recogerán en una sola variable con el nombre de **store_item_nbr**.

Se crea la variable **daydate,** que indica el día de la semana en que aparece una venta por cada artículo/tienda, esto es posible gracias a la información de la variable **date.**

Los datos de las ventas unitarias serán transformados a valores logarítmicos. Los valores negativos de las ventas unitarias se reemplazarán con ceros.

Posteriormente los datos serán filtrados obteniendo las ventas unitarias por cada día de la semana, como se muestra la tabla 9.

Tabla 9. Filtrado de ventas por día de la semana

	date	daydate	store_item_nbr	unit_sales
1	2016-08-01	lunes	1_103520	1.3862944
2	2016-08-01	lunes	1_103665	1.0986123
3	2016-08-01	lunes	1_105574	2.0794415
4	2016-08-01	lunes	1_105575	2.6390573
5	2016-08-01	lunes	1_105577	1.0986123
6	2016-08-01	lunes	1_105693	0.6931472
7	2016-08-01	lunes	1_105737	2.0794415

***Fuente**: Elaboración propia*

Posteriormente, se han creado 7 marcos de datos del formato que se muestra en la tabla 10.

Tabla 10.Marco de datos para la predicción

	store_item_nbr	2016-08-01	2016-08-08	2016-08-15	2016-08-22	2016-08-29	2016-09-05	2016-09-12	2016-09-19	2016-09-26	2016-10-03
1	1_1000866	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2	1_1001305	0.0000000	0.6931472	1.6094379	1.6094379	0.0000000	0.6931472	0.0000000	1.6094379	0.0000000	1.0986123
3	1_1003679	0.0000000	0.0000000	0.0000000	0.6931472	0.6931472	0.0000000	0.0000000	0.6931472	0.0000000	0.6931472
4	1_1004545	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
5	1_1004550	2.0794415	2.5649494	2.3978953	2.4849066	2.3025851	2.1972246	2.7080502	2.5649494	2.0794415	1.3862944
6	1_1004551	2.0794415	2.6390573	2.0794415	1.6094379	2.0794415	2.1972246	1.6094379	1.9459101	2.4849066	2.0794415

***Fuente**: Elaboración propia*

Cada marco de datos representa un día de la semana, por ese motivo son 7. En cada marco se recoge la información de artículo/tienda por cada día de la semana.

Nuevamente se ha realizado una limpieza de datos, reemplazando los valores negativos con ceros.

Mediante la función **snaive** se realza el cálculo para cada uno de los 7 marcos de datos. En este sentido, hay que recordar que las predicciones son desde el 16-08-2017 (miércoles) hasta el 16-08-2017 (jueves).

Por lo cual para cada día de la semana la predicción será igual, es decir; la predicción de los lunes será igual a la del siguiente lunes, y así para cada día de la semana.

La tabla 11 muestra lo mencionado en el párrafo anterior.

Tabla 11.Estructura de predicciones

Store_item_nbr	VENTAS					
	Miércoles	Jueves	..	..	Miércoles	Jueves
	16/8/2017	17/8/2017	...	...	30/8/2017	31/8/2017
1_1001305	0,6931472	0,0000000	...	...	0,6931472	0,0000000
1_1003679	1,0986123	1,0986123	...	...	1,0986123	1,0986123
1_1004545	0,0000000	0,0000000	...	...	0,0000000	0,0000000
1_1004550	2,7725887	0,0000000	...	...	2,7725887	0,0000000
1_1004551	2,3978953	3,7135721	...	...	2,3978953	3,7135721
.....	.....	.....	...	...	.....	.....

***Fuente**: Elaboración propia*

Posteriormente se combinaron las predicciones con los datos de test, esto provocó la generación de valores perdidos que fueron reemplazados con ceros debido a que es la mejor opción para evitar una sobreestimación.

Los datos fueron ajustados al formato de Kaggle. Obteniendo las puntuaciones que se visualizan en la tabla 12.

Tabla 12. Puntuaciones Snaive

Submission and Description	Private Score	Public Score
Snaive predictor (version 4/4) 13 days ago by augusmendoza From "Snaive predictor" Script	0.737	0.711

***Fuente**: Captura de pantalla de Kaggle*

Los resultados han tenido una mejora significativa respecto a los modelos ya presentados en los apartados anteriores.

4.4 Predictor de Suavizado Exponencial (ETS)

En vista de lo aprendido en los modelos anteriores, la creación de la variable **store_item_nbr** es fundamental.

En este modelo a partir de la creación de la variable mencionada, se trabajará con las 3 variables que muestra la tabla 13.

Tabla 13. Variables del Modelo ETS

	date	store_item_nbr	unit_sales
1	2016-08-01	1_103520	1.3862944
2	2016-08-01	1_103665	1.0986123
3	2016-08-01	1_105574	2.0794415
4	2016-08-01	1_105575	2.6390573
5	2016-08-01	1_105577	1.0986123
6	2016-08-01	1_105693	0.6931472
7	2016-08-01	1_105737	2.0794415
8	2016-08-01	1_105857	1.7917595
9	2016-08-01	1_106716	1.0986123
10	2016-08-01	1_108698	1.3862944

***Fuente**: Elaboración propia*

Es decir, se tiene el valor unitario de ventas por cada tienda/artículo desde el 01/08/2016 hasta el 15/08/2017. Esa información es agrupada es un solo marco de datos, como se muestra a continuación.

Tabla 14. Marco de datos para la predicción

	store_item_nbr	2016-08-01	2016-08-02	2016-08-03	2016-08-04	2016-08-05	2016-08-06	2016-08-07
1	1_1000866	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2	1_1001305	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.6931472
3	1_1003679	0.0000000	0.0000000	0.0000000	1.0986123	1.0986123	0.0000000	0.0000000
4	1_1004545	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
5	1_1004550	2.0794415	1.7917595	2.7080502	2.9444390	3.6109179	3.6888795	1.7917595
6	1_1004551	2.0794415	2.3025851	2.5649494	2.8332133	3.2958369	3.4657359	1.3862944
7	1_1009512	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
8	1_1009539	0.6931472	0.0000000	0.0000000	0.6931472	0.0000000	1.0986123	0.0000000
9	1_1009997	1.0986123	0.6931472	1.3862944	1.3862944	1.3862944	1.6094379	1.0986123
10	1_1009998	0.6931472	0.0000000	1.6094379	1.0986123	1.0986123	0.0000000	0.0000000

***Fuente**: Elaboración propia*

En la tabla 14, se han eliminado valores negativos en las ventas unitarias y luego han sido transformados a logaritmos.

Mediante la función **ets** del lenguaje de programación utilizado se realizarán los cálculos para cada una de las 170909 combinaciones tienda/artículo, en torno a la media. Es decir, para los 16 días de pronósticos, las ventas serán exactamente iguales.

La tabla 15 muestra el formato inicial en que se han obtenido las predicciones:

Tabla 15. Estructura de predicciones ETS

	store_item_nbr	2017-08-16	2017-08-17	2017-08-18	..	2017-08-30	2017-08- 31
result.1	1_1000866	9.781011e-02	9.781011e-02	9.781011e-02	..	9.781011e-02	9.781011e-02
result.2	1_1001305	4.436100e-01	4.436100e-01	4.436100e-01	..	4.436100e-01	4.436100e-01
result.3	1_1003679	2.959845e-01	2.959845e-01	2.959845e-01	..	2.959845e-01	2.959845e-01
result.4	1_1004545	9.293216e-02	9.552286e-02	9.811356e-02	..	1.292019e-01	1.317926e-01
result.5	1_1004550	2.587310e+00	2.587310e+00	2.587310e+00	..	2.587310e+00	2.587310e+00
result.6	1_1004551	2.394609e+00	2.394609e+00	2.394609e+00	..	2.394609e+00	2.394609e+00
result.7	1_1009512	1.385950e-01	1.385950e-01	1.385950e-01	..	1.385950e-01	1.385950e-01
result.8	1_1009539	5.152130e-02	5.152130e-02	5.152130e-02	..	5.152130e-02	5.152130e-02

***Fuente**: Elaboración propia*

Luego de realizar otra limpieza y posterior ajuste de datos al formato de subida requerido por la plataforma Kaggle, se obtuvieron las puntuaciones mostradas en la tabla 16.

Tabla 16. Puntuaciones ETS

Submission and Description	Private Score	Public Score
Model ETS (version 7/7) 21 days ago by augusmendoza From "Model ETS " Script	0.597	0.555

***Fuente**: Captura de pantalla de Kaggle*

El modelo ha superado a los modelos anteriormente descritos, siendo además sencillo.

Mediante la función **ets,** también se realizaron predicciones tomando previamente 7 marcos de datos, filtrados por cada día de la semana. Es decir, la predicción de los lunes será igual a del siguiente lunes, y así para cada día de la semana.

En este sentido, lo mencionando anteriormente resulta en cálculos muchos más complejos, en donde una vez ajustados los resultados al formado requerido por Kaggle las puntuaciones obtenidas son mostradas en la tabla 17.

Tabla 17.Puntuaciones Day ETS

Submission and Description	Private Score	Public Score
Mean Day ETS (version 5/5) 21 days ago by augusmendoza From "Mean Day ETS " Script	0.592	0.571

***Fuente**: Captura de pantalla de Kaggle*

La puntuación privada tuvo una mejora poco significativa respecto al mejor modelo (ETS normal) y una disminución significativa respecto a la puntuación pública.

5 Resultados

A medida que se han ido construyendo los diferentes modelos predictivos se ha mostrado las puntuaciones que estos han obtenido. Sin embargo, la tabla 18 muestra las puntuaciones de todos agrupados en función de la puntuación privada, misma que fue la que se tomó en cuenta para decidir los ganadores.

Tabla 18. Puntuaciones finales de los modelos predictivos

Submission and Description	Private Score	Public Score
Mean Day ETS (version 5/5) 14 days ago by augusmendoza From "Mean Day ETS " Script	0.592	0.571
Model ETS (version 7/7) 21 days ago by augusmendoza From "Model ETS " Script	0.597	0.555
Snaive predictor (version 4/4) 14 days ago by augusmendoza From "Snaive predictor" Script	0.737	0.711
Kernel arima (version 5/5) 6 days ago by augusmendoza From "Kernel arima" Script	0.806	0.816
Mean Histor (version 1/1) a month ago by augusmendoza From "Mean Histor" Script	0.807	0.819
Arima mean (version 10/10) a day ago by augusmendoza From "Arima mean" Script	0.877	0.887
Modelo Arima 3 (version 2/2) 25 days ago by augusmendoza From "Modelo Arima 3" Script	0.931	0.942
Sencillo Arima (version 11/11) 6 days ago by augusmendoza From "Sencillo Arima" Script	1.307	1.338

***Fuente**: Captura de pantalla de Kaggle*

Los resultados obtenidos por el mejor modelo fueron satisfactorios, pero tienen todavía un margen de mejora respecto a los equipos ganadores.

Los 3 equipos ganadores, que lograron el premio económico obtuvieron la puntuación privada mostrada en la tabla 19.

Tabla 19. Puntuación equipos ganadores

In the money Gold Silver Bronze

#	△pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	▲13	w			0.509	117	9mo
2	▲10	SoLucky			0.512	308	9mo
3	▼1	slonoschildpad			0.513	265	9mo

***Fuente**: Captura de pantalla de Kaggle*

6 Conclusiones

La competición Corporación Favorita Grocery Sales Forecasting realizada en la plataforma Kaggle, buscaba el mejor modelo de predicción de ventas para miles de productos de la empresa.

Se trataba de un problema de series temporales, donde para solucionarlo se utilizaron modelos: Media Histórica, Predictor Ingenuo, Arima y de Suavizado Exponencial.

Los diferentes modelos obtuvieron resultados que fueron mejorando, el mejor puntuado en la plataforma fue producto del aprendizaje de ir de modelos sencillos hasta los más complejos.

Los resultados de las predicciones son satisfactorios, se ha obtenido un buen puntaje. Aun así, queda un margen de mejora respecto a las técnicas utilizadas para predecir en problemas de series temporales. Por ejemplo, aplicar algoritmos de redes neuronales.

El problema planteado ha sido el más complejo con el que he podido trabajar y demuestra la importancia de tener claro determinados conceptos sobre series temporales, pero sobre todo que en problemas reales no siempre el modelo más complejo obtiene los mejores resultados de predicción.

Realizar un Análisis Exploratorio de Datos (EDA), ha sido fundamental para entender el comportamiento de muchas variables, sobre todo porque es posible programar y visualizar esos comportamientos mediante graficas muy ilustrativas.

Trabajar con una base de datos de casi 5GB, fue otro reto muy importante debido a la cantidad de computación que requiere procesar tanta información. Tomar la proporción adecuada de esos datos, realizar la limpieza y posteriormente poder hacer predicciones dejó mucho aprendizaje.

Quiero destacar el nivel de programación adquirido en R Studio durante la realización del presente trabajo. Ha sido una evolución muy positiva y espero llegar a un nivel superior, puesto que R Studio y Python son los 2 lenguajes de programación más importantes en el mundo del análisis de datos.

Por último, creo que Kaggle es una plataforma magnifica que te genera más pasión por el análisis de datos, además de brindarte bases de datos con problemas reales que ayudan a mejorar tus competencias profesionales; por lo cual es mi deseo seguir participando en más concursos que me brinden un mayor desarrollo profesional.

7 Bibliografía

Bellosta, C. J. (2018). *R para profesionales de los datos: una introducción.* Madrid. Obtenido de https://datanalytics.com/libro_r/_main.pdf

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control* (5th ed.). New York: Wiley.

Brockwell, P. J., & Davis, R. A. (1987). *Time Series: Theory and Methods.* New York: Springer-Verlag.

Brown, R. G. (1959). *Statistical forecasting for inventory control.* McGraw/Hill.

C.Holt, C. (1957). Forecasting seasonals and trends by exponentially weighted averages. *Office of Naval Research*.

Chatfield, C. (2003). *The Analysis of Time Series: An Introduction, Sixth Edition.* Chapman and Hall/CRC .

Cryer, J. D., & Chan, K.-S. (2008). *Time Series Analysis.* Springer.

Dagum, E. B., & Bianconcini, S. (2016). *Seasonal adjustment methods and real time trend-cycle estimation.* Springer.

Fanaee-T, H., & Gama, J. (2013). *Event labeling combining ensemble detectors and background knowledge, Progress in Artificial Intelligence.* Springer Berlin Heidelberg.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice.* OTexts.

Hyndman, R. J., & Khandakar, Y. (2008). Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software*, 1-22.

Kaggle. (2017). *The Official Blog of Kaggle.com*. Obtenido de http://blog.kaggle.com/2017/06/06/weve-passed-1-million-members/

Kaggle. (2018). *Favorita Grocery Sales Forecasting*. Obtenido de https://www.kaggle.com/c/favorita-grocery-sales-forecasting#description

Kaggle. (2018). *Kaggle terms*. Obtenido de https://www.kaggle.com/terms

Ojeda, F. C. (2014). *Análisis exploratorio y visualización de datos con R.* Obtenido de http://fcharte.com/assets/pdfs/ExploraVisualizaConR-FCharte.pdf

Peña, D., Tiao, G. C., & Tsay, R. S. (2000). *A Course in Time Series Analysis.* New York: Wiley.

Robin, H., Koehler, A. B., Ord, J. K., & Snyder, R. D. (2008). *Forecasting with Exponential Smoothing.* Berlín: Springer.

Secretaria de Gestión de Riesgos del Ecuador. (5 de 2016). *Informes se situacion de terremoto.* Obtenido de https://www.gestionderiesgos.gob.ec/wp-content/uploads/downloads/2016/05/INFORME-n71-SISMO-78-20302.pdf

Superintendencia de Compañias del Ecuador. (2018). *Ranking Empresarial Ecuador 2017*. Obtenido de https://appscvs.supercias.gob.ec/rankingCias/

Theodosiou, M. (2001). Forecasting monthly and quarterly time series using STL decomposition. *International Journal of Forecasting*, 1178–1195.

United Nations Secretary-General. (2014). *World Data Forum*. Obtenido de https://undataforum.org/WorldDataForum/about/

United Nations Secretary-General. (2018). *United Nations Global Pulse*. Obtenido de https://www.unglobalpulse.org/about-new

University of California,Irvine. (2018). *Machine Learning Repository*. Obtenido de https://archive.ics.uci.edu/ml/index.php

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis .* Springer.

Wickham, H., & Grolemund, G. (2016). *R for Data Science.* O'Reilly Media.

Winters, P. R. (1960). Forecasting Sales by Exponentially Weighted Moving Averages. *Management Science*, 324–342.

8 Anexos

En los anexos se presentarán todos los códigos utilizados en R Studio.

8.1 Código del Análisis Exploratorio de Datos (EDA)

```
## Importar librerías
library(wesanderson)
library(RColorBrewer)
library(tidyverse)
library(grid)
library(data.table)
library(timeDate)
library(forecast)
library(tseries)
#=====================================================================
# Cargar los datos

train <- fread("train.csv", skip = 86672217, header =FALSE) #Datos del último
año. Desde el 01 de agosto del 2016.
setnames(train, c("id","date", "store_nbr", "item_nbr", "unit_sales", "OnProm
otion"))

test <- fread("test.csv")
stores <- fread("stores.csv")
items <- fread("items.csv")
trans <- fread("transactions.csv")
oil <- fread("oil.csv")
holidays <- fread("holidays_events.csv")

#=====================================================================# Vi
sualizar la estructura de los datos
summary(train)
glimpse(train)
summary(test)
glimpse(test)
summary(stores)
glimpse(stores)
summary(items)
glimpse(items)
summary(trans)
glimpse(trans)
summary(oil)
glimpse(oil)
summary(holidays)
glimpse(holidays)

#=====================================================================
# Nuevo formatos de fecha, factores y lógicos a determinadas variables.
train <- train %>%
    mutate(date = parse_datetime(date),
         store_nbr = as.factor(store_nbr),
         item_nbr = as.factor(item_nbr))
```

```
test <- test %>%
  mutate(date = parse_datetime(date),
         store_nbr = as.factor(store_nbr),
         item_nbr = as.factor(item_nbr))

stores <- stores %>%
  mutate(city = as.factor(city),
         state = as.factor(state),
         type = as.factor(type),
         store_nbr = as.factor(store_nbr))

items <- items %>%
  mutate(family = as.factor(family),
         class = as.factor(class),
         perish = as.logical(perishable),
         item_nbr = as.factor(item_nbr))

trans <- trans %>%
  mutate(date = parse_datetime(date),
         store_nbr = as.factor(store_nbr))

oil <- oil %>%
  mutate(date = parse_datetime(date))

holidays <- holidays %>%
  mutate(type = as.factor(type),
         locale = as.factor(locale),
         locale_name = as.factor(locale_name))

#########################################################################
#La funcion Multiplot para usarla se define de acuerdo al siguiente link
# http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/

multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  plots <- c(list(...), plotlist)
  numPlots = length(plots)
  if (is.null(layout)) {
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }
  if (numPlots==1) {
    print(plots[[1]])
  } else {
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))
    for (i in 1:numPlots) {
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))
      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

```
#########################################################################
#Visualización datos de entrenamiento
1 <- train %>%
  select(unit_sales, date) %>%
  group_by(date) %>%
  filter(unit_sales > 0) %>%
  summarise(sales = sum(unit_sales)) %>%
  ggplot(aes(sales)) +
  geom_histogram(fill = "red", bins = 50) +
  labs(title = "Positive distribution Unit_sales-All Stores")+
  theme(legend.position = "none")

p2 <- train %>%
  filter(unit_sales < 0) %>%
  ggplot(aes(-unit_sales)) +
  geom_histogram(fill = "blue", binwidth = 0.1) +
  scale_x_log10() +
  scale_y_log10() +
  labs(title = "Negative distribution unit sales = returns")

p3 <- train %>%
  filter(unit_sales > 0) %>%
  ggplot(aes(unit_sales, fill = OnPromotion)) +
  geom_density(bw = 0.2, alpha = 0.5) +
  scale_x_log10()+
  labs(title = "Unit sales distribucion- On Promotion")

p4 <- train %>%
  select(store_nbr, unit_sales) %>%
  group_by(store_nbr)%>%
  summarise(sales = sum(unit_sales)) %>%
  ggplot(aes(reorder(store_nbr, sales, FUN = min), sales,
             fill = reorder(store_nbr, sales, FUN = min))) +
  geom_col() +
  theme(legend.position = "none") +
  coord_flip()+
  labs( title = "Total sales by store")

layout <- matrix(c(1,2,3,1,2,3,1,2,3,4,4,4,4,4,4,4,4,4,4,4,4),7,3,byrow=TRUE)
multiplot(p1, p2, p3, p4, layout=layout)

#########################################################################
#Visualización de Stores por localización y tipo

foo <- train %>%
  select(store_nbr, unit_sales) %>%
  left_join(stores, by = "store_nbr")

p1 <-foo %>%
  group_by(city) %>%
  summarise(sales = sum(unit_sales)) %>%
  ggplot(aes(reorder(city, sales, FUN = min), sales,
             fill = reorder(city, sales, FUN = min))) +
  geom_col() +
  theme(legend.position = "none") +
  labs(title = "Total sales by City", x = "City", y = "Total sales") +
  coord_flip()
```

```
p2 <-foo %>%
  group_by(state) %>%
  summarise(sales = sum(unit_sales)) %>%
  ggplot(aes(reorder(state, sales, FUN = min), sales,
             fill = reorder(state, sales, FUN = min))) +
  geom_col() +
  theme(legend.position = "none") +
  labs(title ="Total sales by State", x = "State", y = "Total sales") +
  coord_flip()

p3 <-foo %>%
  group_by(type) %>%
  summarise(sales = sum(unit_sales)) %>%
  ggplot(aes(reorder(type, sales, FUN = min), sales,
             fill = reorder(type, sales, FUN = min))) +
  geom_col(fill= brewer.pal(n = 5, name = "Spectral")) +
  theme(legend.position = "none") +
  labs(title ="Total sales by Type", x = "Type", y = "Total sales") +
  coord_flip()

p4 <-foo %>%
  group_by(cluster) %>%
  summarise(sales = sum(unit_sales)) %>%
  ggplot(aes(reorder(cluster, sales, FUN = min), sales,
             fill = reorder(cluster, sales, FUN = min))) +
  geom_col(fill= 'RoyalBlue') +
  theme(legend.position = "none") +
  labs(title ="Total sales by Cluster", x = "Type", y = "Total sales") +
  coord_flip()

p5 <- foo %>%
  group_by(cluster, type,cluster) %>%
  summarise(sales = sum(unit_sales)) %>%
  ggplot()+
  aes(type,sales, fill= cluster, label= cluster) +
  geom_col( colour = "white")+
  geom_text(position = position_stack(vjust = 0.4), color = "black", size= 3)
+
  scale_fill_distiller(name= 'cluster',
                       type= 'seq',
                       palette=1)+
  labs(title ="Relation Cluster-type-sales")+
  coord_flip()

p6 <-stores %>%
  ggplot(aes(state, fill = city)) +
  geom_bar() +
  theme() +
  labs(title ="Relation State-City")+
  coord_flip()


layout <- matrix(c(1,2,1,2,3,4,3,4,5,6,5,6),6,2,byrow=TRUE)
multiplot(p1, p2, p3, p4, p5,p6, layout=layout)
```

```
#Visualización de Items

p1 <-items %>%
  ggplot(aes(family, fill = perish)) +
  geom_bar() +
  theme() +
  labs(title ="Items Family Perishable ")+
  coord_flip()
foo <- train %>%
  select(item_nbr, unit_sales) %>%
  mutate(item_nbr = as.character(item_nbr)) %>%
  left_join(items %>% mutate(item_nbr = as.character(item_nbr)), by = "item_n
br")

p2 <-foo %>%
  group_by(family)  %>%
  summarise(sales = sum(unit_sales)) %>%
  ungroup() %>%
  mutate(family = str_sub(family, start = 1, end = 19)) %>%
  ggplot(aes(reorder(family, sales), sales,
             fill = reorder(family, sales)))+
  geom_col() +
  scale_y_log10() +
  scale_x_discrete()+
  theme(legend.position = "none") +
  labs(title = "Sales by Items Family", x = "Items Family", y = "Total sales"
) +
  coord_flip()

p3 <-foo %>%
  group_by(item_nbr) %>%
  summarise(sales = sum(unit_sales)) %>%
  arrange(desc(sales)) %>%
  head(15) %>%
  ggplot(aes(reorder(item_nbr, sales,FUN = min), sales,
             fill = reorder(item_nbr, sales, FUN = max))) +
  geom_point(size = 3,color= 'darkslateblue')+
  theme(legend.position = "none") +
  labs(title = "Best top 15 Items Sales", x = "Items", y = "Total sales") +
  coord_flip()

p4 <-foo %>%
  group_by(class) %>%
  summarise(sales = sum(unit_sales)) %>%
  arrange(desc(sales)) %>%
  head(15) %>%
  ggplot(aes(reorder(class, sales,FUN = min), sales,
             fill = reorder(class, sales, FUN = min))) +
  geom_point(size = 3,color= 'brown') +
  theme(legend.position = "none") +
  labs(title = "Best top 15 Items Class Sales", x = "Items Class", y = "Total
sales") +
  coord_flip()

layout <- matrix(c(1,2,1,2,3,4,3,4),4,2,byrow=TRUE)
multiplot(p1, p2, p3, p4, layout=layout)
```

```
#Visualización de Transactions

p1 <-trans %>%
  group_by(date) %>%
  summarise(med_trans = median(transactions)) %>%
  ggplot(aes(date, med_trans)) +
  geom_line(color = "dodgerblue4", size= 1) +
  geom_smooth(method = "loess", color = "gold1", span = 1/5, size= 1.5)+
  labs(title = "Transactions")

foo <- stores %>%
  select(store_nbr, state) %>%
  left_join(trans, by = "store_nbr")

p2 <-foo %>%
  arrange(desc(state)) %>%
  ggplot(aes(date, transactions, color = state)) +
  geom_smooth(method = "loess", span = 1/2, se = FALSE, size= 3)+
  labs(title = "Transactions by State")

layout <- matrix(c(1,1,2,2),2,2,byrow=TRUE)
multiplot(p1, p2, layout=layout)

#########################################################################
#Visualización de Oil

p1 <- oil %>%
  ggplot(aes(date, dcoilwtico)) +
  geom_line(color = "black", size= 1.2) +
  geom_smooth(method = "loess", color = "green", span = 1/5)+
  labs(title = "WTI Oil Price ")

  foo <- trans %>%
  group_by(date) %>%
  summarise(med_trans = median(transactions))

oil_med <- oil %>%
  filter(date > min(foo$date))

oil_med <- oil_med %>%
mutate(dcoilwtico = ( min(foo$med_trans, na.rm = TRUE) +
(dcoilwtico-min(dcoilwtico, na.rm = TRUE))/(max(dcoilwtico, na.rm = TRUE)
        - min(dcoilwtico, na.rm = TRUE)) *
  (max(foo$med_trans, na.rm = TRUE) - min(foo$med_trans, na.rm = TRUE)) ))

p2<-foo %>%
  ggplot(aes(date, med_trans)) +
  geom_line(color= 'mediumvioletred', size= 0.8) +
  geom_line(data = oil_med, aes(date, dcoilwtico), colour = "black", size= 1.
2) +
  labs(title = "WTI Oil Price (black) with Transactions (Violet)")

layout <- matrix(c(1,1,2,2),2,2,byrow=TRUE)
multiplot(p1, p2, layout=layout)
```

```
#########################################################################
#Visualization Holiday

p1 <- holidays %>%
  ggplot(aes(type, fill = locale)) +
  geom_bar() +
  theme()+
  coord_flip()+
  labs(title = "Holiday Type-Locale")

p2 <-holidays %>%
  ggplot(aes(locale_name, fill = locale)) +
  geom_bar() +
  theme()+
  coord_flip()+
  labs(title = "Holiday Locale")

p3 <-holidays %>%
  ggplot(aes(locale_name, fill = type)) +
  geom_bar() +
  theme()+
  coord_flip()+
  labs(title = "Holiday Locale Name")

p4 <- holidays %>%
  group_by(description) %>%
  count() %>%
  arrange(desc(n)) %>%
  head(12) %>%
  ggplot(aes(description, n)) +
  geom_col(fill= brewer.pal(n = 12, name = "Paired"))+
  theme() +
  coord_flip()+
  labs(title = "Holiday Most frequent")

p5 <-holidays %>%
  filter(transferred > 0) %>%
  ggplot(aes(description, fill = transferred))+
  geom_bar()+
  geom_bar(fill= brewer.pal(n = 8, name = "Accent")) +
  coord_flip()+
  labs(title = "Holiday Transfer by Locale Name")

p6 <-holidays %>%
  filter(transferred > 0) %>%
  ggplot(aes(description, fill = locale))+
  geom_bar()+
  coord_flip()+
  labs(title = "Holiday Transfer by Locale")

layout <- matrix(c(1,2,3,4,5,6),2,3,byrow=TRUE)
multiplot(p1, p2, p3, p4,p5,p6, layout=layout)
```

8.2 Código de media histórica

Para que este código funcione se necesita ejecutar el código anterior (EDA) debido a que se realizaron transformaciones de algunas variables.

Dicho lo anterior, el código es el siguiente:

```
#Reemplazar valores negativos con ceros
train$unit_sales[train$unit_sales < 0] <- 0

# Expresar unit_sales en logaritmo
train$unit_sales <- log1p(train$unit_sales)

# Calcular la media de ventas para por cada artículo y tienda
sales <- train %>%
  group_by(item_nbr,store_nbr) %>%
  summarise(med_sales = mean(unit_sales))
# Merging with the test

tmp_test <- merge(test, sales, all.x = TRUE)
tmp_test$med_sales[is.na(tmp_test$med_sales)]<- 0
tmp_test$med_sales[tmp_test$med_sales< 0]<- 0

setnames(tmp_test, c("store_nbr","item_nbr","id","date","OnPromotion",
"unit_sales"))
submission <- tmp_test[,c("id","unit_sales")]
submission <- arrange(submission, id) # Orden directo

write.csv(submission,file= "media_historica.csv", row.names = FALSE)
```

8.3 Código ARIMA

Para que este código funcione se necesita ejecutar el código anterior (EDA) debido a que se realizaron transformaciones de algunas variables.

Dicho lo anterior, el código es el siguiente

```
#Librerías adicionales
library(forecast)
library(tseries)

############################################
#ARIMA...
sales <- train %>%
  group_by(store_nbr,item_nbr) %>%
  summarise(med_sales = mean(unit_sales))

sales$med_sales[sales$med_sales< 0] <- 0
sales$med_sales <- log1p(sales$med_sales)
```

```
#Dar formato de series temporales
count_ts = ts(sales$med_sales)
sales$med_sales_ts= tsclean(count_ts,replace.missing = TRUE, lambda = NULL)

#descomponer serie
count_ma = ts(na.omit(sales$med_sales_ts), frequency=7)
decomp = stl(count_ma, s.window="periodic")
deseasonal_sales <- seasadj(decomp)
plot(decomp)

#Estacionaridad
adf.test(count_ma, alternative = "stationary")
#La serie no es estacionaria

#Autocorrelacion
Acf(count_ma, main='')
Pacf(count_ma, main='')

#Primera diferencia
count_d1 = diff(deseasonal_sales, differences = 1)
plot(count_d1)

adf.test(count_d1, alternative = "stationary")

Acf(count_d1, main='')
Pacf(count_d1, main='')

#Auto arima
auto.arima(deseasonal_sales, seasonal = TRUE)

fit <- arima(deseasonal_sales, order=c(5,1,7),seasonal=list(order=c(1,0,2)))
tsdisplay(residuals(fit), lag.max=45, main='')
fcast <- forecast(fit, h=16)

prediction<-data_frame(fcast$x)
setnames(prediction, c("unit_sales"))

f_train <- data.frame(sales, prediction)

pre_test <- merge(test, f_train, all.x = TRUE)
pre_test$unit_sales[is.na(pre_test$unit_sales)]<- 0
pre_test$unit_sales[pre_test$unit_sales<0]<- 0

submission <- pre_test[,c("id","unit_sales")]
write.csv(submission,file= "arima_mn.csv", row.names = FALSE, col.names =
FALSE)
```

8.4 Código predictor ingenuo

Para que este código funcione se necesita ejecutar el código anterior (EDA) debido a que se realizaron transformaciones de algunas variables.

Dicho lo anterior, el código es el siguiente.

```
#Librerías
library(forecast)
library(reshape2)
library(foreach)
library(lubridate)
############
train$store_item_nbr <- paste(train$store_nbr, train$item_nbr, sep="_")
test$store_item_nbr <- paste(test$store_nbr, test$item_nbr, sep="_")
train$daydate <- paste(weekdays(train$date))

sales <- train[, c('date','daydate','store_item_nbr', 'unit_sales')]

sales$unit_sales[sales$unit_sales < 0] <- 0
sales$unit_sales <- log1p(sales$unit_sales)

#Ventas unitarias por día de la semana
sales_md1<- filter(sales,daydate =="lunes")
sales_md2<- filter(sales,daydate =="martes")
sales_md3<- filter(sales,daydate =="miércoles")
sales_md4<- filter(sales,daydate =="jueves")
sales_md5<- filter(sales,daydate =="viernes")
sales_md6<- filter(sales,daydate =="sábado")
sales_md7<- filter(sales,daydate =="domingo")

# Ajustar los datos al formato largo para series temporales

sales_lf1 <- dcast(sales_md1, store_item_nbr ~ date, value.var =
"unit_sales", fill = 0)
sales_lf2 <- dcast(sales_md2, store_item_nbr ~ date, value.var =
"unit_sales", fill = 0)
sales_lf3 <- dcast(sales_md3, store_item_nbr ~ date, value.var =
"unit_sales", fill = 0)
sales_lf4 <- dcast(sales_md4, store_item_nbr ~ date, value.var =
"unit_sales", fill = 0)
sales_lf5 <- dcast(sales_md5, store_item_nbr ~ date, value.var =
"unit_sales", fill = 0)
sales_lf6 <- dcast(sales_md6, store_item_nbr ~ date, value.var =
"unit_sales", fill = 0)
sales_lf7 <- dcast(sales_md7, store_item_nbr ~ date, value.var =
"unit_sales", fill = 0)

sales_cl1<- sales_lf1[, 1]
sales_cl1<-data_frame(sales_cl1)
```

```
sales_cl2<- sales_lf2[, 1]
sales_cl2<-data_frame(sales_cl2)

sales_cl3<- sales_lf3[, 1]
sales_cl3<-data_frame(sales_cl3)

sales_cl4<- sales_lf4[, 1]
sales_cl4<-data_frame(sales_cl4)

sales_cl5<- sales_lf5[, 1]
sales_cl5<-data_frame(sales_cl5)

sales_cl6<- sales_lf6[, 1]
sales_cl6<-data_frame(sales_cl6)

sales_cl7<- sales_lf7[, 1]
sales_cl7<-data_frame(sales_cl7)

#Ciclo de compras semanal

sales_ts1 <- ts(sales_lf1, frequency = 7)
sales_ts2 <- ts(sales_lf2, frequency = 7)
sales_ts3 <- ts(sales_lf3, frequency = 7)
sales_ts4 <- ts(sales_lf4, frequency = 7)
sales_ts5 <- ts(sales_lf5, frequency = 7)
sales_ts6 <- ts(sales_lf6, frequency = 7)
sales_ts7 <- ts(sales_lf7, frequency = 7)
#######################################
fc_mat1 <- matrix(NA,nrow=nrow(sales_ts1),ncol=1)
fc_mat2 <- matrix(NA,nrow=nrow(sales_ts2),ncol=1)
fc_mat3 <- matrix(NA,nrow=nrow(sales_ts3),ncol=1)
fc_mat4 <- matrix(NA,nrow=nrow(sales_ts4),ncol=1)
fc_mat5 <- matrix(NA,nrow=nrow(sales_ts5),ncol=1)
fc_mat6 <- matrix(NA,nrow=nrow(sales_ts6),ncol=1)
fc_mat7 <- matrix(NA,nrow=nrow(sales_ts7),ncol=1)
#######

fc_mat1 <- foreach(i=1:nrow(sales_ts1),.combine=rbind,
.packages=c("forecast")) %dopar% {
  fc_mat1 <- forecast(snaive(sales_ts1[i,]),h=1)$mean
}

fc_mat2 <- foreach(i=1:nrow(sales_ts2),.combine=rbind,
.packages=c("forecast")) %dopar% {
  fc_mat2 <- forecast(snaive(sales_ts2[i,]),h=1)$mean
}

fc_mat3 <- foreach(i=1:nrow(sales_ts3),.combine=rbind,
.packages=c("forecast")) %dopar% {
  fc_mat3 <- forecast(snaive(sales_ts3[i,]),h=1)$mean
}

fc_mat4 <- foreach(i=1:nrow(sales_ts4),.combine=rbind,
.packages=c("forecast")) %dopar% {
  fc_mat4 <- forecast(snaive(sales_ts4[i,]),h=1)$mean
}
```

```
fc_mat5 <- foreach(i=1:nrow(sales_ts5),.combine=rbind,
.packages=c("forecast")) %dopar% {
  fc_mat5 <- forecast(snaive(sales_ts5[i,]),h=1)$mean
}

fc_mat6 <- foreach(i=1:nrow(sales_ts6),.combine=rbind,
.packages=c("forecast")) %dopar% {
  fc_mat6 <- forecast(snaive(sales_ts6[i,]),h=1)$mean
}

fc_mat7 <- foreach(i=1:nrow(sales_ts7),.combine=rbind,
.packages=c("forecast")) %dopar% {
  fc_mat7 <- forecast(snaive(sales_ts7[i,]),h=1)$mean
}
################

s_predict1 <- as.data.frame(cbind(sales_cl3,fc_mat3, row.names= NULL))
s_predict1$date <- as.Date("2017-08-16")
setnames(s_predict1, c("store_item_nbr", "unit_sales", "date"))

s_predict2 <- as.data.frame(cbind(sales_cl4,fc_mat4, row.names= NULL))
s_predict2$date <- as.Date("2017-08-17")
setnames(s_predict2, c("store_item_nbr", "unit_sales", "date"))

s_predict3 <- as.data.frame(cbind(sales_cl5,fc_mat5, row.names= NULL))
s_predict3$date <- as.Date("2017-08-18")
setnames(s_predict3, c("store_item_nbr", "unit_sales", "date"))

s_predict4 <- as.data.frame(cbind(sales_cl6,fc_mat6, row.names= NULL))
s_predict4$date <- as.Date("2017-08-19")
setnames(s_predict4, c("store_item_nbr", "unit_sales", "date"))

s_predict5 <- as.data.frame(cbind(sales_cl7,fc_mat7, row.names= NULL))
s_predict5$date <- as.Date("2017-08-20")
setnames(s_predict5, c("store_item_nbr", "unit_sales", "date"))

s_predict6 <- as.data.frame(cbind(sales_cl1,fc_mat1, row.names= NULL))
s_predict6$date <- as.Date("2017-08-21")
setnames(s_predict6, c("store_item_nbr", "unit_sales", "date"))

s_predict7 <- as.data.frame(cbind(sales_cl2,fc_mat2, row.names= NULL))
s_predict7$date <- as.Date("2017-08-22")
setnames(s_predict7, c("store_item_nbr", "unit_sales", "date"))
s_predict8 <- as.data.frame(cbind(sales_cl3,fc_mat3, row.names= NULL))
s_predict8$date <- as.Date("2017-08-23")
setnames(s_predict8, c("store_item_nbr", "unit_sales", "date"))

s_predict9 <- as.data.frame(cbind(sales_cl4,fc_mat4, row.names= NULL))
s_predict9$date <- as.Date("2017-08-24")
setnames(s_predict9, c("store_item_nbr", "unit_sales", "date"))

s_predict10 <- as.data.frame(cbind(sales_cl5,fc_mat5, row.names= NULL))
s_predict10$date <- as.Date("2017-08-25")
setnames(s_predict10, c("store_item_nbr", "unit_sales", "date"))
```

```
s_predict11 <- as.data.frame(cbind(sales_cl6,fc_mat6, row.names= NULL))
s_predict11$date <- as.Date("2017-08-26")
setnames(s_predict11, c("store_item_nbr", "unit_sales", "date"))

s_predict12 <- as.data.frame(cbind(sales_cl7,fc_mat7, row.names= NULL))
s_predict12$date <- as.Date("2017-08-27")
setnames(s_predict12, c("store_item_nbr", "unit_sales", "date"))

s_predict13 <- as.data.frame(cbind(sales_cl1,fc_mat1, row.names= NULL))
s_predict13$date <- as.Date("2017-08-28")
setnames(s_predict13, c("store_item_nbr", "unit_sales", "date"))

s_predict14<- as.data.frame(cbind(sales_cl2,fc_mat2, row.names= NULL))
s_predict14$date <- as.Date("2017-08-29")
setnames(s_predict14, c("store_item_nbr", "unit_sales", "date"))

s_predict15 <- as.data.frame(cbind(sales_cl3,fc_mat3, row.names= NULL))
s_predict15$date <- as.Date("2017-08-30")
setnames(s_predict15, c("store_item_nbr", "unit_sales", "date"))

s_predict16 <- as.data.frame(cbind(sales_cl4,fc_mat4, row.names= NULL))
s_predict16$date <- as.Date("2017-08-31")
setnames(s_predict16, c("store_item_nbr", "unit_sales", "date"))

################################
s_predict_all<- rbind(s_predict1,s_predict2,s_predict3,s_predict4,
                      s_predict5, s_predict6,s_predict7,s_predict8,
                      s_predict9,s_predict10,s_predict11,s_predict12,
                      s_predict13,s_predict14,s_predict15,s_predict16)

# Post procesado y limpieza final

pre_test <- merge(test, s_predict_all, all.x = TRUE)
pre_test$unit_sales[pre_test$unit_sales<0]<- 0
pre_test$unit_sales[is.na(pre_test$unit_sales)]<- 0
pre_test$unit_sales <- expm1(pre_test$unit_sales)

# Predicciones en formato Kaggle

submission <- pre_test[,c("id","unit_sales")]
write.csv(submission,file= "snaive.csv", row.names = FALSE)
```

8.5 Códigos ETS

Para que estos códigos funcionen se necesita ejecutar el código anterior (EDA) debido a que se realizaron transformaciones de algunas variables.

Hay 2 versiones del código, la primera es la siguiente:

```
#Librerías

library(forecast)
library(reshape2)
library(foreach)
```

```
#Nueva variable con información store-item
train$store_item_nbr <- paste(train$store_nbr, train$item_nbr, sep="_")
test$store_item_nbr <- paste(test$store_nbr, test$item_nbr, sep="_")

sales <- train[, c('date','store_item_nbr', 'unit_sales')]
sales$unit_sales[sales$unit_sales < 0] <- 0
sales$unit_sales <- log1p(sales$unit_sales)

# Ajustar los datos al formato largo para series temporales

sales_lf <- dcast(sales, store_item_nbr ~ date, value.var = "unit_sales",
fill = 0)
sales_cl <- sales_lf[, 1]
sales_cl<-data_frame(sales_cl)

#Ciclo de compras semanal
sales_ts <- ts(sales_lf, frequency = 7)

#Crear matriz para predicciones
fc = 16
fc_mat <- matrix(NA,nrow=nrow(sales_ts),ncol=fc)

#Forecasting
registerDoMC(detectCores()-1)

fc_mat <- foreach(i=1:nrow(sales_ts),.combine=rbind, .packages=c("forecast"))
%dopar% {
  fc_mat <- forecast(ets(sales_ts[i,]),h=fc)$mean
}

#Post procesado

fc_mat[fc_mat < 0] <- 0
colnames(fc_mat) <- as.character(seq(from = as.Date("2017-08-16"),
                                     to = as.Date("2017-08-31"),
                                     by = 'day'))

predict <- as.data.frame(cbind(sales_cl, fc_mat))
colnames(predict)[1] <- "store_item_nbr"

predict_lf <- melt(predict, id = 'store_item_nbr',
                   variable.name = "date",
                   value.name = 'unit_sales')

predict_lf$date <- as.Date.factor(predict_lf$date)
predict_lf$unit_sales <- as.numeric(predict_lf$unit_sales)

predict_lf$unit_sales <- expm1(predict_lf$unit_sales)

# Post procesado y limpieza final

pre_test <- merge(test, predict_lf, all.x = TRUE)
pre_test$unit_sales[is.na(pre_test$unit_sales)]<- 0
pre_test$unit_sales[pre_test$unit_sales<0]<- 0

# Predicciones en formato Kaggle

submission <- pre_test[,c("id","unit_sales")]
write.csv(submission,file= "Model_ets.csv", row.names = FALSE)
```

La segunda versión, la cual calcula un modelo ETS por cada día de la semana para todas las combinaciones tienda-artículo es la siguiente:

```
#Librerías
library(forecast)
library(reshape2)
library(foreach)
library(lubridate)

###########

train$store_item_nbr <- paste(train$store_nbr, train$item_nbr, sep="_")
test$store_item_nbr <- paste(test$store_nbr, test$item_nbr, sep="_")
train$daydate <- paste(weekdays(train$date))

sales <- train[, c('date','daydate','store_item_nbr', 'unit_sales')]

sales$unit_sales[sales$unit_sales < 0] <- 0
sales$unit_sales <- log1p(sales$unit_sales)

#Ventas unitarias por dia de la semana

sales_md1<- filter(sales,daydate =="lunes")
sales_md2<- filter(sales,daydate =="martes")
sales_md3<- filter(sales,daydate =="miércoles")
sales_md4<- filter(sales,daydate =="jueves")
sales_md5<- filter(sales,daydate =="viernes")
sales_md6<- filter(sales,daydate =="sábado")
sales_md7<- filter(sales,daydate =="domingo")

# Ajustar los datos al formato largo para series temporales

sales_lf1 <- dcast(sales_md1, store_item_nbr ~ date, value.var =
"unit_sales", fill = 0)
sales_lf2 <- dcast(sales_md2, store_item_nbr ~ date, value.var =
"unit_sales", fill = 0)
sales_lf3 <- dcast(sales_md3, store_item_nbr ~ date, value.var =
"unit_sales", fill = 0)
sales_lf4 <- dcast(sales_md4, store_item_nbr ~ date, value.var =
"unit_sales", fill = 0)
sales_lf5 <- dcast(sales_md5, store_item_nbr ~ date, value.var =
"unit_sales", fill = 0)
sales_lf6 <- dcast(sales_md6, store_item_nbr ~ date, value.var =
"unit_sales", fill = 0)
sales_lf7 <- dcast(sales_md7, store_item_nbr ~ date, value.var =
"unit_sales", fill = 0)

##########################

sales_cl1<- sales_lf1[, 1]
sales_cl1<-data_frame(sales_cl1)

sales_cl2<- sales_lf2[, 1]
sales_cl2<-data_frame(sales_cl2)
```

```
sales_cl3<- sales_lf3[, 1]
sales_cl3<-data_frame(sales_cl3)

sales_cl4<- sales_lf4[, 1]
sales_cl4<-data_frame(sales_cl4)

sales_cl5<- sales_lf5[, 1]
sales_cl5<-data_frame(sales_cl5)

sales_cl6<- sales_lf6[, 1]
sales_cl6<-data_frame(sales_cl6)

sales_cl7<- sales_lf7[, 1]
sales_cl7<-data_frame(sales_cl7)

#Ciclo de compras semanal
sales_ts1 <- ts(sales_lf1, frequency = 7)
sales_ts2 <- ts(sales_lf2, frequency = 7)
sales_ts3 <- ts(sales_lf3, frequency = 7)
sales_ts4 <- ts(sales_lf4, frequency = 7)
sales_ts5 <- ts(sales_lf5, frequency = 7)
sales_ts6 <- ts(sales_lf6, frequency = 7)
sales_ts7 <- ts(sales_lf7, frequency = 7)

#######################################
fc_mat1 <- matrix(NA,nrow=nrow(sales_ts1),ncol=1)
fc_mat2 <- matrix(NA,nrow=nrow(sales_ts2),ncol=1)
fc_mat3 <- matrix(NA,nrow=nrow(sales_ts3),ncol=1)
fc_mat4 <- matrix(NA,nrow=nrow(sales_ts4),ncol=1)
fc_mat5 <- matrix(NA,nrow=nrow(sales_ts5),ncol=1)
fc_mat6 <- matrix(NA,nrow=nrow(sales_ts6),ncol=1)
fc_mat7 <- matrix(NA,nrow=nrow(sales_ts7),ncol=1)
#######

fc_mat1 <- foreach(i=1:nrow(sales_ts1),.combine=rbind,
.packages=c("forecast")) %dopar% {
  fc_mat1 <- forecast(ets(sales_ts1[i,]),h=1)$mean
}

fc_mat2 <- foreach(i=1:nrow(sales_ts2),.combine=rbind,
.packages=c("forecast")) %dopar% {
  fc_mat2 <- forecast(ets(sales_ts2[i,]),h=1)$mean
}

fc_mat3 <- foreach(i=1:nrow(sales_ts3),.combine=rbind,
.packages=c("forecast")) %dopar% {
  fc_mat3 <- forecast(ets(sales_ts3[i,]),h=1)$mean
}

fc_mat4 <- foreach(i=1:nrow(sales_ts4),.combine=rbind,
.packages=c("forecast")) %dopar% {
  fc_mat4 <- forecast(ets(sales_ts4[i,]),h=1)$mean
}
```

```
fc_mat5 <- foreach(i=1:nrow(sales_ts5),.combine=rbind,
.packages=c("forecast")) %dopar% {
  fc_mat5 <- forecast(ets(sales_ts5[i,]),h=1)$mean
}

fc_mat6 <- foreach(i=1:nrow(sales_ts6),.combine=rbind,
.packages=c("forecast")) %dopar% {
  fc_mat6 <- forecast(ets(sales_ts6[i,]),h=1)$mean
}

fc_mat7 <- foreach(i=1:nrow(sales_ts7),.combine=rbind,
.packages=c("forecast")) %dopar% {
  fc_mat7 <- forecast(ets(sales_ts7[i,]),h=1)$mean
}
################

s_predict1 <- as.data.frame(cbind(sales_cl3,fc_mat3, row.names= NULL))
s_predict1$date <- as.Date("2017-08-16")
setnames(s_predict1, c("store_item_nbr", "unit_sales", "date"))

s_predict2 <- as.data.frame(cbind(sales_cl4,fc_mat4, row.names= NULL))
s_predict2$date <- as.Date("2017-08-17")
setnames(s_predict2, c("store_item_nbr", "unit_sales", "date"))

s_predict3 <- as.data.frame(cbind(sales_cl5,fc_mat5, row.names= NULL))
s_predict3$date <- as.Date("2017-08-18")
setnames(s_predict3, c("store_item_nbr", "unit_sales", "date"))

s_predict4 <- as.data.frame(cbind(sales_cl6,fc_mat6, row.names= NULL))
s_predict4$date <- as.Date("2017-08-19")
setnames(s_predict4, c("store_item_nbr", "unit_sales", "date"))


s_predict5 <- as.data.frame(cbind(sales_cl7,fc_mat7, row.names= NULL))
s_predict5$date <- as.Date("2017-08-20")
setnames(s_predict5, c("store_item_nbr", "unit_sales", "date"))

s_predict6 <- as.data.frame(cbind(sales_cl1,fc_mat1, row.names= NULL))
s_predict6$date <- as.Date("2017-08-21")
setnames(s_predict6, c("store_item_nbr", "unit_sales", "date"))

s_predict7 <- as.data.frame(cbind(sales_cl2,fc_mat2, row.names= NULL))
s_predict7$date <- as.Date("2017-08-22")
setnames(s_predict7, c("store_item_nbr", "unit_sales", "date"))

s_predict8 <- as.data.frame(cbind(sales_cl3,fc_mat3, row.names= NULL))
s_predict8$date <- as.Date("2017-08-23")
setnames(s_predict8, c("store_item_nbr", "unit_sales", "date"))

s_predict9 <- as.data.frame(cbind(sales_cl4,fc_mat4, row.names= NULL))
s_predict9$date <- as.Date("2017-08-24")
setnames(s_predict9, c("store_item_nbr", "unit_sales", "date"))

s_predict10 <- as.data.frame(cbind(sales_cl5,fc_mat5, row.names= NULL))
s_predict10$date <- as.Date("2017-08-25")
setnames(s_predict10, c("store_item_nbr", "unit_sales", "date"))
```

```
s_predict11 <- as.data.frame(cbind(sales_cl6,fc_mat6, row.names= NULL))
s_predict11$date <- as.Date("2017-08-26")
setnames(s_predict11, c("store_item_nbr", "unit_sales", "date"))

s_predict12 <- as.data.frame(cbind(sales_cl7,fc_mat7, row.names= NULL))
s_predict12$date <- as.Date("2017-08-27")
setnames(s_predict12, c("store_item_nbr", "unit_sales", "date"))

s_predict13 <- as.data.frame(cbind(sales_cl1,fc_mat1, row.names= NULL))
s_predict13$date <- as.Date("2017-08-28")
setnames(s_predict13, c("store_item_nbr", "unit_sales", "date"))

s_predict14<- as.data.frame(cbind(sales_cl2,fc_mat2, row.names= NULL))
s_predict14$date <- as.Date("2017-08-29")
setnames(s_predict14, c("store_item_nbr", "unit_sales", "date"))

s_predict15 <- as.data.frame(cbind(sales_cl3,fc_mat3, row.names= NULL))
s_predict15$date <- as.Date("2017-08-30")
setnames(s_predict15, c("store_item_nbr", "unit_sales", "date"))

s_predict16 <- as.data.frame(cbind(sales_cl4,fc_mat4, row.names= NULL))
s_predict16$date <- as.Date("2017-08-31")
setnames(s_predict16, c("store_item_nbr", "unit_sales", "date"))

##################################
s_predict_all<- rbind(s_predict1,s_predict2,s_predict3,s_predict4,
                      s_predict5, s_predict6,s_predict7,s_predict8,
                      s_predict9,s_predict10,s_predict11,s_predict12,
                      s_predict13,s_predict14,s_predict15,s_predict16)

# Post procesado y limpieza final

pre_test <- merge(test, s_predict_all, all.x = TRUE)
pre_test$unit_sales[pre_test$unit_sales<0]<- 0
pre_test$unit_sales[is.na(pre_test$unit_sales)]<- 0
pre_test$unit_sales <- expm1(pre_test$unit_sales)

# Predicciones en formato Kaggle

submission <- pre_test[,c("id","unit_sales")]
write.csv(submission,file= "Model_DayMeanEts.csv", row.names = FALSE)
```