UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Máster Oficial en Eficiencia Energética

Trabajo Final de Master

Minería de Datos para el Desarrollo y Estudio Comparativo de Modelos Predictivos de la Ocupación, Volumen y Velocidad Medias del tráfico rodado en Estambul

Pablo López Dolz

Tutores: Máximo Méndez Babey
Javier J. Sánchez Medina

18 de diciembre de 2017

$A\ L.M^a.$

Agradecimientos

A todos aquellos profesores y maestros que han pasado al largo de mi vida, motivando la curiosidad, el autoaprendizaje y la autocrítica, herramientas esenciales para algo tan importante como es la felicidad sana en uno mismo, con su pareja y con su entorno.

Agradecer a Javier, a Máximo y a Antonio Carlos por haberme soportado durante el desarrollo de esta tarea y tutorizarme.

A la universidad de Estambul por la aportación de los datos necesarios para realizar este trabajo final de máster.

Y sobretodo a mi familia y amigos, por la suerte que he tenido de coincidir en el tiempo y el lugar. Sin ellos, no estaría aquí escribiendo lo que escribo. Gracias.

Resumen

Este trabajo fin de máster tiene una orientación de trabajo de investigación. Se ha obtenido un dataset de velocidades, ocupación y volumen de tráfico muestreados en tres puntos de la vía D100, que circunvala Estambul, muestreados con una frecuencia de 2 minutos. Con estos datos se ha realizado primero un análisis exploratorio, encontrándose elementos de gran interés.

Con posterioridad a dicho análisis, se ha dedicado un tiempo a feature engineering, para luego emplearse el resto del tiempo del trabajo fin de máster al desarrollo de diversos modelos, empleándose diferentes técnicas del mundo de la estadística, como modelos autoregresivos ARIMA, SARIMA, etc.

Finalmente se realiza un análisis comparativo de los diversos modelos desarrollados, con el fin de determinar cuál ofrece una predicción de más calidad, no sólo con una buena evaluación, sino preservando la generalidad del mismo, con el fin de futuras aplicaciones.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

Hoy en día el parque mundial de automóviles no para de crecer [3], como se observa en la figura 1.1, el uso del automóvil a seguido creciendo y esta última década se ha aumentado el volumen total más de un tercio sobrepasando ya los 1,250,000,000 de vehículos. Según la OICA (Organización Internacional de Constructores de Automóviles) la producción mundial creció un 4.7 % entre 2015 y 2016 según los últimos datos disponibles en diciembre de 2017 [13].

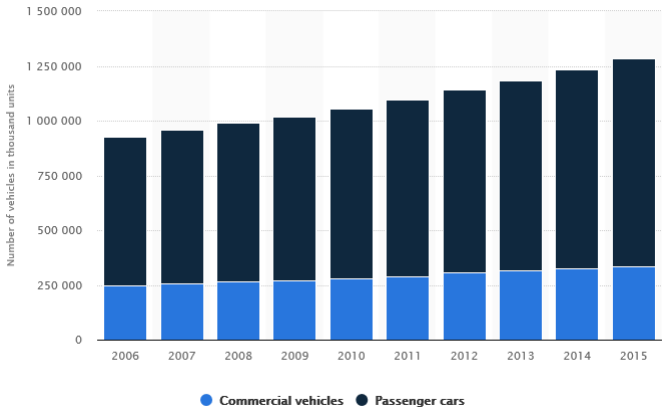

Figura 1.1: Evolución del número de coches en uso a nivel mundial, en miles. Fuente: https://www.statista.com

Este hecho viene ligado a varias problemáticas implicitas, como una mayor ocupación de la calzada, tener más probabilidad de atascos, la necesidad de más plazas de aparcamiento, aumento de la concentración de CO_2 en áreas metropolitanas y consecuente daño a la salud pública [6], colaboración al agotamiento de fuentes de energía no renovables como los hidrocarburos, etc... Y un ejemplo claro del aumento del uso del automóvil se ve en una ciudad importante, donde cualquier persona que haya cogido el coche en hora punta,

se habrá encontrado con serias dificultades para circular con normalidad en ciertos puntos de la red viaria.

Por otra parte también existe una lectura positiva de estos mismos hechos, ya que la mayor parte de los nuevos vehículos sustituyen al antiguo parque automovilístico, aportando mejores prestaciones y menores consumos de hidrocarburos o incluso el uso de otras fuentes de energía para su funcionamiento; además de ofrecer una herramienta de libre movilidad a sus usuarios en los países en vías de desarrollo, como se ve en la siguiente imagen, dónde sin duda se muestra el mayor crecimiento en los últimos años.

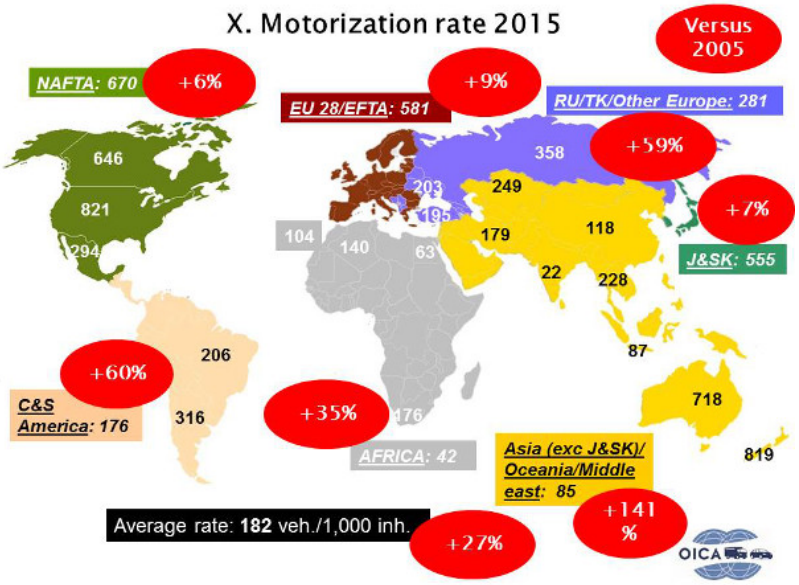

Figura 1.2: Mapamundi con el crecimiento porcentual de los distintos continentes durante el 2005 y 2015. Fuente: http://www.oica.net

Existe gran cantidad de grupos de investigación, ya sean públicos o privados, asociaciones y conjuntamente con apoyos gubernamentales, dedicados a la reducción o eliminación de los efectos negativos del tráfico automovilístico. Como por ejemplo, la producción de coches eléctricos, legislación para empresas productoras de coches y consejos para la reducción del consumo [12] , ampliación de los viales para aumentar la capacidad de la red viaria, gestión eficiente del tráfico,... Este último tema es el foco de este trabajo.

Dentro de la gestión del tráfico, existen varias propuestas: control de los semáforos en tiempo real según la circulación, señales urbana con la capacidad

de adaptar las indicaciones a las condiciones reales, saber el estado del tráfico a partir de los móviles/GPS de los usuarios/conductores, empresas como Waze y Google ya son especialistas como se ve en la imagen 1.3; predicción a corto tiempo de posibles atascos... La mayoría de las propuestas ya llevan un largo camino recorrido y hay un amplio abanico de opciones con las que mejorar las condiciones del tráfico, pero si hay algo claro, es que la gestión del tráfico viene ligada a la gestión de grandes masas de datos (Big Data) sobre el tráfico y la circulación a tiempo real. Medidas tecnológicas que acercan a nuestras ciudades a las denominadas Smart Cities (ciudades inteligentes). Evidentemente, todo esto debe ir acompañado por nuevos sistemas tecnológicos en nuestros coches capaces de aprovechar las ventajas que toda esta gestión aportaría al día a día de los ciudadanos.

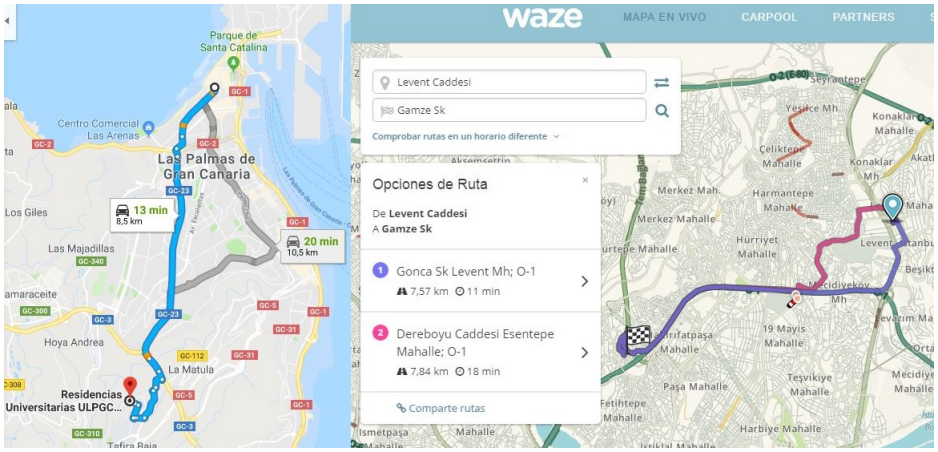

Figura 1.3: Mapas de tráfico en tiempo real, izquierda GoogleMaps y derecha Waze, ofrecen el tiempo aproximado de la ruta escogida teniendo en cuenta el estado de las carreteras.

A partir de una serie temporal con datos de velocidad, ocupación y volumen de 3 puntos de una autovía perteneciente a la red viaria de Estambul, se estudian y desarrollan los diferentes modelos predictivos.

Revisando el estado del arte [11] de los métodos de predicción se ha decidido dividirlo en dos grupos, métodos estadísticos y métodos de aprendizaje por computación (Machine Learning / Deep Learning), este segundo grupo no se abarca en este estudio, se aplaza para futuras investigaciones, aunque ya es sabido que muestra mejores resultados en variables dependientes del tráfico urbano.

Uno de los métodos estadísticos que más se utiliza para la predicción a corto plazo de los atributos correspondientes a series temporales es el modelo ARIMA *(Modelo Autorregresivo-Integrado de Medias Móviles)* [18] y [5], mediante el cual se llegan a estimar futuras series no estacionales con muy buenos resultados, claro está, el modelo hay que ajustarlo personalmente y tiene sus limitaciones a la hora de calcular con un número grande de datos previos. Funciona de una manera óptima para series de frecuencias bajas, anuales, por cuatrimestres, trimestres o incluso meses; pero para frecuencias más bajas mejor utilizar otro modelo o modificar los datos de entrada.

Otro caso que se estudia dentro del trabajo es el modelo SARIMA [15], como se puede intuir, es igual al anterior, aunque añadiendo la estacionalidad (*Seasonal ARIMA*) ayudando al modelo a ser más certero en predicciones futuras de series temporales con una variación periódica marcada, como por ejemplo podrían ser las temperaturas medias en un mismo lugar durante varios años.

También se ponen a prueba modelos encontrados por el autor y que han tenido resultados notables, como funciones implementadas para *R* basada en ajustes de modelos lineales aplicadas a series temporales (tslm) o modelos ETS (Error, Tendencia y Estacionalidad) como el **TBATS** [8], que a diferencia del modelo ARIMA no obliga a que la estacionalidad sea periódica, sino que permite una estacionalidad dinámica.

Finalmente, dentro de los modelos estadísticos, se utiliza el modelo de regresión dinámico-armónico con la ayuda de las transformadas de Fourier [4], obteniendo unos resultados sobresalientes en comparación con los demás métodos probabilísticos.

Antes de finalizar este apartado de la introducción y empezar con los objetivos, mencionar que lo que motivó inicialmente a la realización de un Trabajo Final de Máster de esta temática fue el tiempo que se pasa dentro del coche parado, ya sea en semáforos o momentos de congestión del tráfico. La idea inicial surgió en un semáforo dónde durante minuto y medio, no pasó ningún coche por la intersección mientras había coches esperando en el semáforo rojo y cansado ya, aquí el autor piensa por primera vez en la eficiencia del tráfico urbano.

1.1. Objetivos

1.1.1. Generales

OG-01 Obtener conclusiones y modelos predictivos que ayuden a las futuras investigaciones sobre la estimación del trafico urbano e interurbano y acoten de una manera precisa el caso de estudio en cuestión.

OG-02 Mejorar la integración social de los ciudadanos, aportando una accesibilidad y conectividad más universal, incrementando su calidad de vida y preservando sus condiciones de salud y seguridad [9](People).

OG-03 Reducir la dependencia de los combustibles fósiles y contribuir al descenso de las emisiones de gases de efecto invernadero mediante el uso de medios (en este caso infraestructuras) de transporte cada vez más sostenibles [9](Planet).

OG-04 La gestión del tráfico en tiempo real mediante el uso de sistemas inteligentes de transporte, reduciendo los tiempos de espera y la congestión. *"The use of Intelligent Transport Systems contributes to real-time traffic management, reducing delivery times and congestion for last mile distribution"* [7].

OG-05 Poder extrapolar el presente proyecto/estudio a otros municipios o puntos conflictivos.

OG-06 Desarrollar, innovar y adoptar soluciones y tecnologías de procesamiento masivo de datos e información [1](Reto 7, IV - ii).

1.1.2. Específicos

OE-01 Dar prioridad a ámbitos de especial relevancia en la Comunidad Autónoma como la eco-innovación, o la **innovación en transporte y logística**, servicios y, de forma particular, turismo.[2]

OE-02 Reducir los tiempos de espera de los usuarios de la red de transporte así como su tiempo de trayecto. Centrándose en los puntos conflictivos (cruces con semáforos y rotondas).

OE-03 Aumentar el bienestar de los conductores(ciudadanos, chófer, turistas).

OE-04 Potenciar las relaciones y colaboraciones con otras universidades y sus equipos de investigación. [10] (5.2.c)

1.2. Datos sobre Estambul y localización del estudio

Estambul, conocida históricamente como Bizancio y, posteriormente, Constantinopla, es la ciudad más poblada de Turquía y el centro histórico, cultural y económico del país. La ciudad se encuentra en la región del Mármara (Figura 1.4), tiene una extensión de 1539 Km^2 y una población aproximada de 14,8 millones de habitantes (31 diciembre 2016)

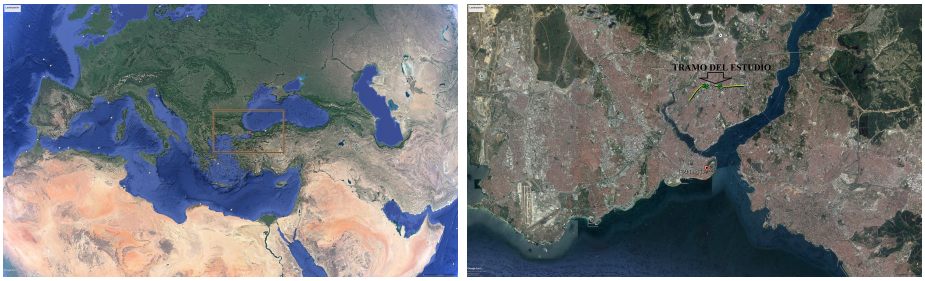

Figura 1.4: Localización en el mapa de la ciudad de Estambul (izq) y plano de la ciudad (der).

En la ciudad de Estambul hay registrados un total de 3,845,349 coches [16] según datos oficiales de diciembre de 2016, aproximadamente 1 cada 5 personas. Y en concreto, el volumen del tráfico medio en la carretera donde se centra el estudio es de unos 62,000 vehículos diarios, alcanzando picos de 6,000 coches en una hora. Es una de las ciudades con mayores problemas de congestión en el mundo debido a la alta densidad de población, la necesidad de desplazarse al lugar de trabajo situado a las afueras de la ciudad o lo suficientemente lejos como para coger el coche y la falta de espacio para las infraestructuras.

Figura 1.5: Imagen de un atasco en la ciudad de Estambul

Los datos disponibles para el estudio corresponden a un histórico del tráfico rodado entre los días 22 y 28 de enero de 2016 en la circunvalación (D-100; O-1) de Estambul (Turquía) en los puntos 363, 533, 544 (Ver figura 1.6).

Figura 1.6: Ortofoto de la zona del estudio y los respectivos puntos de aforo (363, 533, 534).

De estos puntos de aforo se almacenan los datos correspondientes a la velocidad media (S_), el volumen medio (V_) y la ocupación media (O_) de cada carril (_1, _2, _3) cada 2 minutos y las medias de velocidad y ocupación del conjunto de los tres carriles. Anotar que debido a problemas con las infraestructuras existen vacíos temporales en el almacenamiento de datos, habiendo puntos dónde durante 5 horas no se acopia ningún dato en la base, más adelante se explicará como se resolvió el problema.

Capítulo 2

Metodología

En el presente capítulo se describe el diseño del trabajo, dando los detalles suficientes para que otra persona pueda reproducir el estudio y especificando las herramientas utilizadas y los paquetes software necesarios.

2.1. Herramienta R - RStudio

En esta sección se explicará de una manera breve el programa utilizado para el desarrollo del trabajo final de máster, como lo es el *software R y RStudio*, del cual se hizo una introducción a su manejo en el máster.

Figura 2.1: Logo de los programas R y RStudio

R es un entorno y lenguaje de programación con un enfoque al análisis estadístico. Es uno de los lenguajes más utilizados en la rama de la investigación centrada en la estadística, siendo famoso en campos como la minería de datos, la bioinformática, investigación biomédica y matemáticas financieras. Consiste en una programación en código abierto del lenguaje S pero con un soporte de alcance estático, es decir, un nombre/variable dentro del programa siempre se refiere a su entorno léxico local. Ejemplo:*si la función f invoca una función g definida de forma separada, entonces, bajo ámbito léxico, la función g no tiene acceso a las variables locales de f (asumiendo que el texto de g no se encuentra dentro del texto de f).*

RStudio es un entorno de desarrollo integrado *(IDE)* para R, haciendo más manejable el uso de esta herramienta. El software incluye una consola, un editor de código que resalta la sintaxis del programa apoyando su ejecución, así como herramientas de diagramado *(plots)*, historial y gestión del espacio de trabajo. En la figura 2.2 se muestra la interfaz del programa.

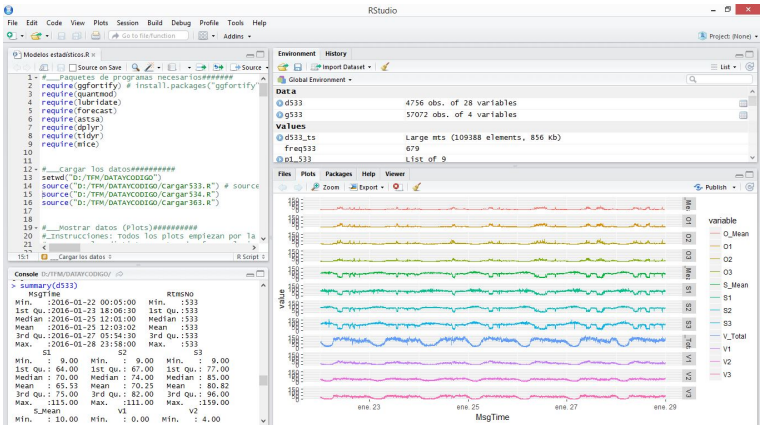

Figura 2.2: Interfaz del programa RStudio.

Una de las ventajas que presenta este tipo de software libre es su amplia utilización por varios equipos de investigación y usuarios particulares que aportan un gran almacén de librerías ahorrando tiempo en la implementación de futuros códigos.

Durante el desarrollo de la memoria se irán explicando las librerías utilizadas, los problemas encontrados con el programa y su solución propuesta.

2.2. Librería/Paquetes R utilizados

En esta sección se exponen las paquetes instalados para el desarrollo de la tarea llevada a cabo, aunque más adelante se justificará de manera individual el uso de algunas de las librerías, aquí se ara un breve resumen de su función.

```
require(ggfortify)
require(quantmod)
require(lubridate)
require(dplyr)
require(tidyr)
```

```
require(astsa)
require(forecast)
require(mice)
```

Ggfortify es un paquete de herramientas de trazado unificadas, comúnmente utilizadas en estadísticas , como GLM, series temporales, familias de PCA, agrupamiento y análisis de supervivencia. El paquete ofrece una única interfaz de trazado para estos resultados de análisis y representa un estilo unificado usando "ggplot2".

El paquete **quantmod** es un marco cuantitativo de modelado financiero, orientado principalmente a al campo de las financias, se puede dar otros usos con objeto en series temporales ya que tiene funciones y paquetes incluidos, como el **xts**, **zoo** y **TTR**.

Lubridate contiene las funciones para trabajar con fechas y tiempos: mediante un análisis rápido y fácil para usar datos de fecha y hora, extracción y actualización de componentes de una determinada fecha y hora (años, meses, días, horas, minutos y segundos), manipulación algebraica de objetos temporales y con lapso de tiempo.

El paquete **dplyr** es una herramienta rápida y consistente para trabajar con objetos similares a marcos de datos, tanto en memoria como sin memoria.

Tidyr es un paquete diseñado específicamente para la ordenación de datos, y funciona bien con conjuntos de datos tratados con **dplyr**.

Astsa son las siglas de *Applied Statistical Time Series Analysis*, este paquete se orienta al análisis estadístico de las series temporales.

El paquete **forecast** contiene las funciones de previsión para las series temporales y los modelos lineales, además de métodos y herramientas para visualizar y analizar los pronósticos de las series temporales univariantes. La función *auto.arima()* pertenece a este paquete.

Finalmente el paquete **mice** sirve para la atribución múltiple mediante ecuaciones encadenadas, como describen Van Buuren y Groothuis-Oudshoorn (2011) en su articulo [17] se utiliza la especificación totalmente controlada ,*Fully Conditional Specification (FCS)*, para conseguir atribuir a valores no disponibles un valor correlativo a los otros atributos y además, cada variable

tiene su modelo de atribución

Existen una gran variedad de paquetes en el ámbito de *R* pero se ha decidido que con estos es suficiente para el desarrollo de la tarea, el siguiente paso es obtener y preparar unos buenos datos de entrada para su posterior estudio.

2.3. Preprocesamiento de los datos obtenidos

Los datos iniciales proporcionados por la Universidad de Estambul residen en un fichero *Excel* con 4 pestañas, 3 de ellas muestran los datos recogidos por las distintas estaciones de aforo: la 533, la 363 y la 534. Cada una de estas pestañas contiene 13 columnas con la siguiente distribución:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	MsgTime	RtmsNo	S1	S2	S3	S_Mean	V1	V2	V3	O1	O2	O3	O_Mean	
2	2016-01-28 23:14:09.000	533	65	75	88	76	16	50	44	5	13	9	9	
3	2016-01-28 23:12:09.000	533	70	77	96	81	15	46	47	7	10	9	8	
4	2016-01-28 23:10:09.000	533	69	72	78	73	31	52	39	8	16	9	11	
5	2016-01-28 23:08:08.000	533	83	78	88	83	21	46	49	4	11	10	8	
6	2016-01-28 23:06:09.000	533	65	72	85	74	36	53	47	11	13	10	11	
7	2016-01-28 23:04:09.000	533	85	82	93	86	18	38	30	5	9	6	6	
8	2016-01-28 23:02:09.000	533	70	75	86	77	19	45	45	6	13	9	9	
9	2016-01-28 23:00:09.000	533	75	74	83	77	20	42	40	6	11	9	8	
10	2016-01-28 22:58:09.000	533	72	72	85	76	23	44	50	7	11	11	9	

534 363 533 coordinates

Figura 2.3: Captura de pantalla del archivo *Excel*, ejemplo del punto 533.

Donde:

- *MsgTime* - corresponde al tiempo en el que se recogen los datos, con el formato (yyyy-mm-dd HH:MM:ss.000) lo que es lo mismo a (año-mes-día Hora).
- *RtmsNo* - muestra el punto de aforo correspondiente.
- *S1, S2* y *S3* - son las velocidades medias de cada carril en (Km/h).
- *S_Mean* - la velocidad media de la calzada, es decir, la media de las 3 componentes anteriores, en (Km/h).
- *V1, V2* y *V3* - es el volumen total de vehículos aforados por cada carril.
- *O1, O2* y *O3* - muestra el porcentaje de ocupación de las distintas líneas, en (%).

- *O_Mean* - la ocupación media de la calzada, es decir, la media de las 3 componentes anteriores, en (%).

Además en el documento, última pestaña atañe a las características de cada punto y sus coordenadas como se muestra en la tabla 2.1 a continuación:

Número del detector	**Nombre del detector**	**Desde**	**Hasta**	**Coordenada X** (Grados)	**Coordenada Y** (Grados)
363	D100 Çağlayan	Okmeydanı	Mecidiyeköy	28.97892104	41.06729962
533	Mecidiyeköy	Mecidiyeköy	Çağlayan	28.99086788	41.06717977
534	Çağlayan Kavşağı	Çağlayan	Okmeydanı	28.97605893	41.06773344

Cuadro 2.1: Características y coordenadas de los puntos de aforo.

A partir de esta hoja de cálculo se exportan tres ficheros "*.csv*", uno por cada punto de aforo, para su posterior manipulación dentro de la herramienta RStudio. En este caso se han nombrado: *363.csv*, *533.csv* y *534.csv*.

Nota: No es necesaria la exportación como ficheros "*.csv*" ya que R dispone de librerías para cargar directamente tablas desde archivos "*.xls / .xlsx*", pero en este trabajo se ha optado por la primera opción que parece más atractiva.

2.3.1. Cargar y visualizar los datos en R

Para cargar los datos, primero se debe situar el directorio de trabajo en la carpeta dónde se encuentren los documentos determinados, en este caso de ejemplo se encuentran en el escritorio dentro de la carpeta llamada"TFM" así que el código inicial es:

```
setwd("C:/.../Desktop/TFM")
```

Una vez establecido el directorio de trabajo ya se pueden se cargar los datos, dándole un nombre a cada tabla *csv*, como se muestra en el siguiente ejemplo:

```
d363 <- read.csv("363.csv")
d533 <- read.csv("533.csv")
d534 <- read.csv("534.csv")
```

Cargados los conjuntos de datos, se comprueba que todo está correcto, en este caso se opta por utilizar la función ***summary()*** que refleja los

cuartiles y los valores mínimo y máximo de cada una de las variables. Se encuentran errores relacionados con la clase de ciertos atributos que deberían ser numéricos y no lo son, para ello lo corregiremos de la siguiente forma:

```
d363$S1 <- as.numeric(d363$S1)
d363$S2 <- as.numeric(d363$S2)
d363$S3 <- as.numeric(d363$S3)
d363$S_Mean <- as.numeric(d363$S_Mean)
d363$O_Mean <- as.numeric(d363$O_Mean)
```

Ulteriormente, realizado este cambio, depende del computador que se este utilizando, suelen aparecer problemas al convertir los datos y aparecen como **Na** (datos no disponibles). Estos normalmente se ignoran evitando así problemas en los futuros cálculos aunque con la consecuencia de mermar la calidad del estudio, por esa razón se opta por utilizar la librería **mice** para autocompletar los datos no disponibles. El código que se muestra a continuación atribuye valores a los Na, estos valores son calculados a partir de correlaciones existentes con los demás atributos.

```
d363_ <- d363[,3:28]
temp.imputed  <- mice(d363_, m = 5, maxit = 50, meth ='pmm'
, seed = 500)
data363 <- complete(temp.imputed,1)
d363 <- cbind(d363[,1:2], data363)
```

Nota: no se añaden las columnas que no influyen a la atribución de los valores no disponibles, como lo serian los atributos RtmsNo y MsgTime, además este último aún no tiene la clase que le corresponde.

Una vez corregidos los valores no disponibles (Na), se crea la variable "Volumen Total", correspondiente al número total de coches que circulan por la calzada como resultante de la suma del volumen de cada uno de los carriles (Vtotal = V1+V2+V3). Seguidamente se da clase de atributo temporal a la variable "MsgTime", en este caso POSIXct , formato para tratar con datos pertenecientes a series temporales posteriormente se ordena. Código utilizado:

```
d363$V_Total <- d363$V1 + d363$V2 + d363$V3
d363$MsgTime <- as.POSIXct(d363$MsgTime)
d363 <- d363[order(d363$MsgTime),]
```

Después de corregir los datos originales y ordenarlos, mediante la función *gather* de la librería ***tidyr*** conseguimos *plotear* los datos de los atributos de cada uno de los puntos de aforo de una manera visual. A continuación se muestra un ejemplo del código para el punto 363 y su consiguiente *plot*, seguido de los *plots* de los puntos 533 y 534::

```
p363 <- ggplot(g363, aes(x=MsgTime,y=value,col=variable))+
geom_line() + facet_grid(variable ~ .)
p363
```

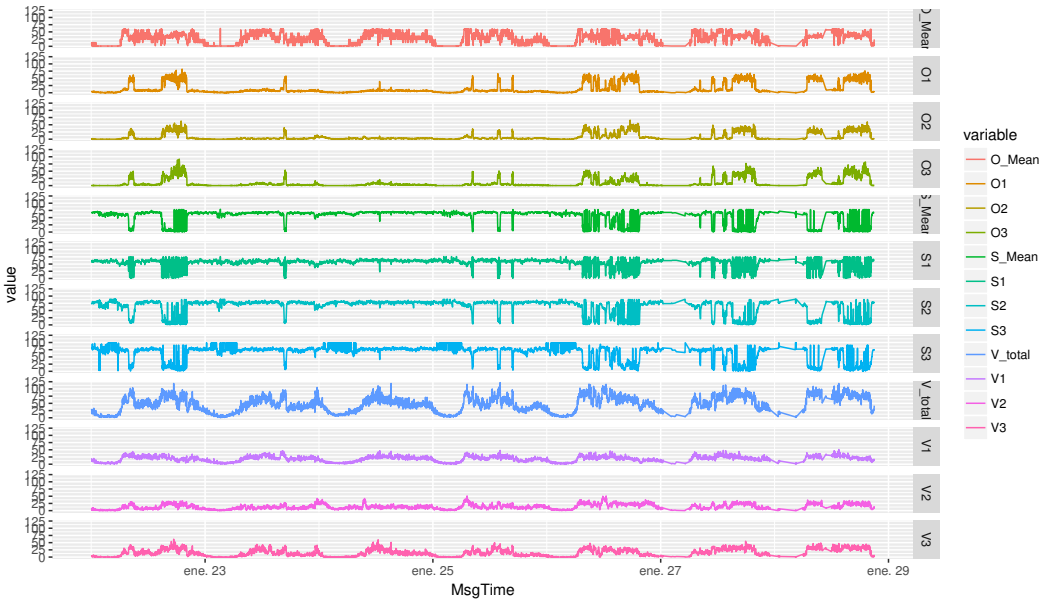

Figura 2.4: Evolución temporal de las variables pertenecientes al punto de aforo 363.

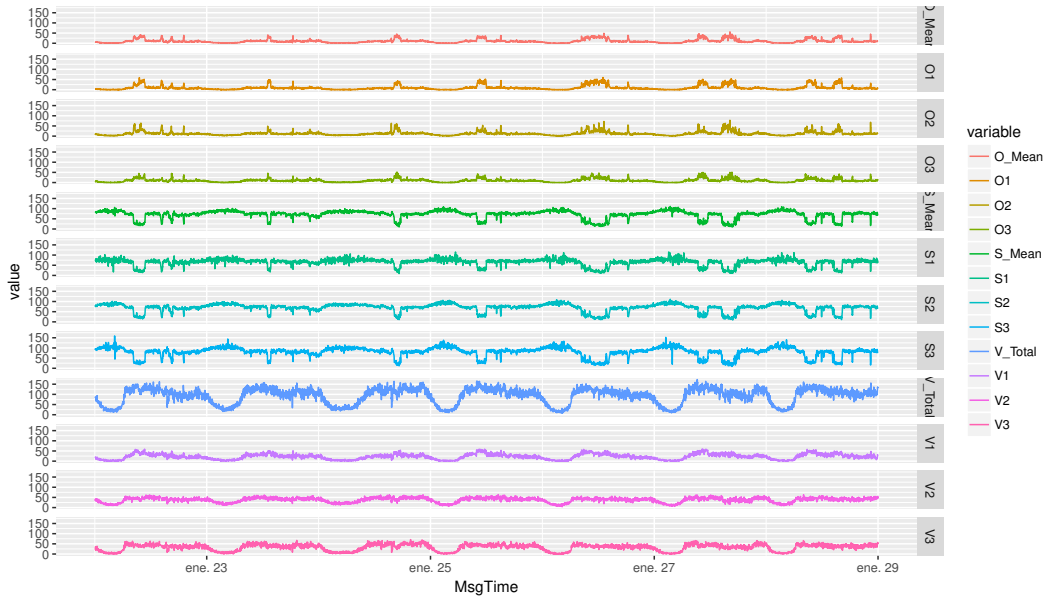

Figura 2.5: Evolución temporal de las variables pertenecientes al punto de aforo 533.

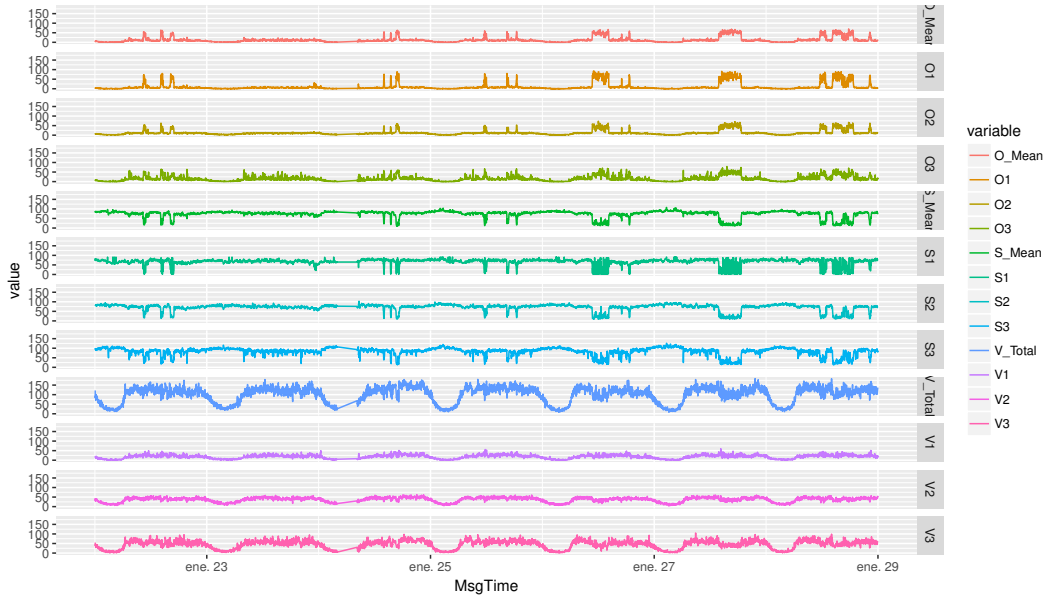

Figura 2.6: Evolución temporal de las variables pertenecientes al punto de aforo 534.

En el código base del trabajo se han añadido unos cuantos gráficos, pero el lector debe saber que las posibilidades con la herramienta R son mucho más grandes de lo que aparece en este trabajo. De este modo aque solamente se muestra lo preciso para explicar los pasos a seguir y que la reproducción del estudio funciona de una manera óptima.

Por ejemplo es posible observar conjuntamente el mismo atributo de los tres carriles superpuestos (Figura 2.7) y sin superponer (Figura 2.8) o los diferentes atributos (Velocidad, Volumen y Ocupación) de una misma calzada (Figura 2.9) y (Figura 2.10). Esto ayuda a una mejor comprensión de los datos y observar sus relaciones.

Cada uno es libre para visualizar los datos de la manera que le satisfaga, el autor cree que el elegir plotear las medias cada media hora en vez de cada dos minutos quita jugo a los gráficos pero le da más "sabor" que ayuda a la reflexión sobre el comportamiento de las variables estudiadas.

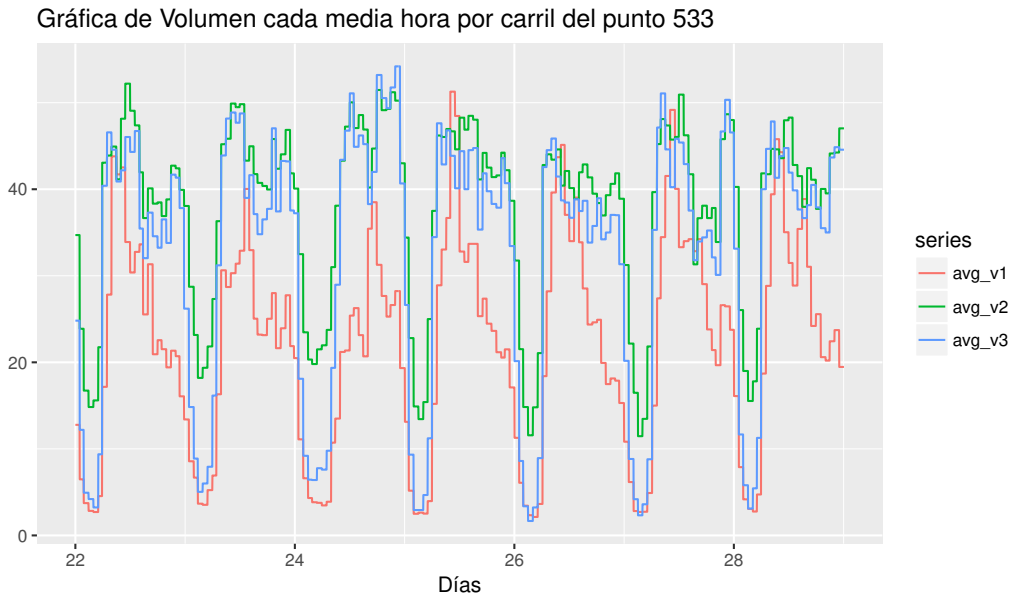

Figura 2.7: Volumen cada dos minutos de cada uno de los 3 carriles. Punto 533.

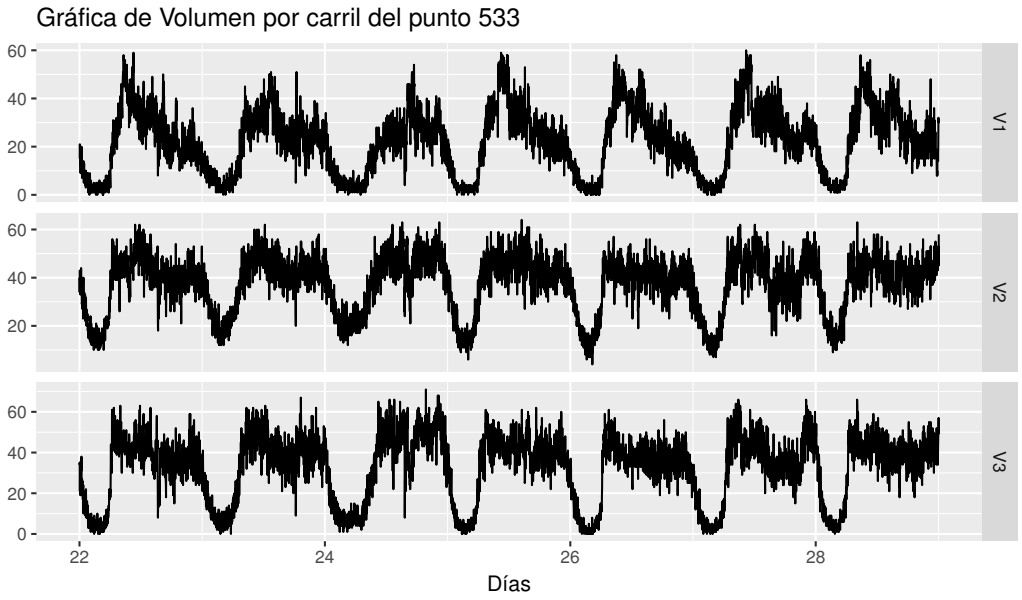

Figura 2.8: Volumen medio de cada uno de los 3 carriles cada media hora, superpuestos. Punto 533.

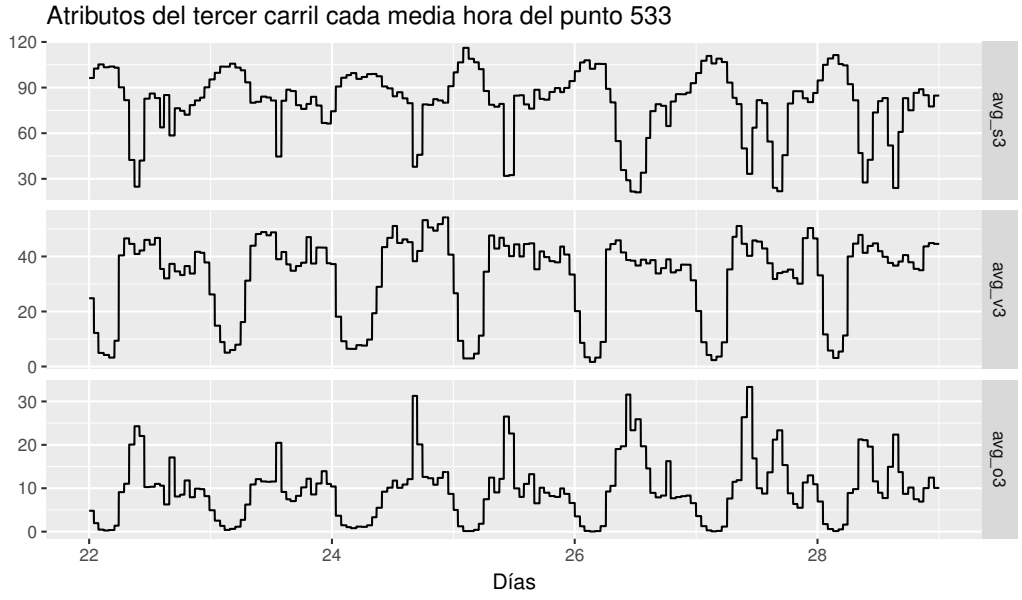

Figura 2.9: Velocidad, Volumen y Ocupación del 3er carril, cada media hora. Punto 533.

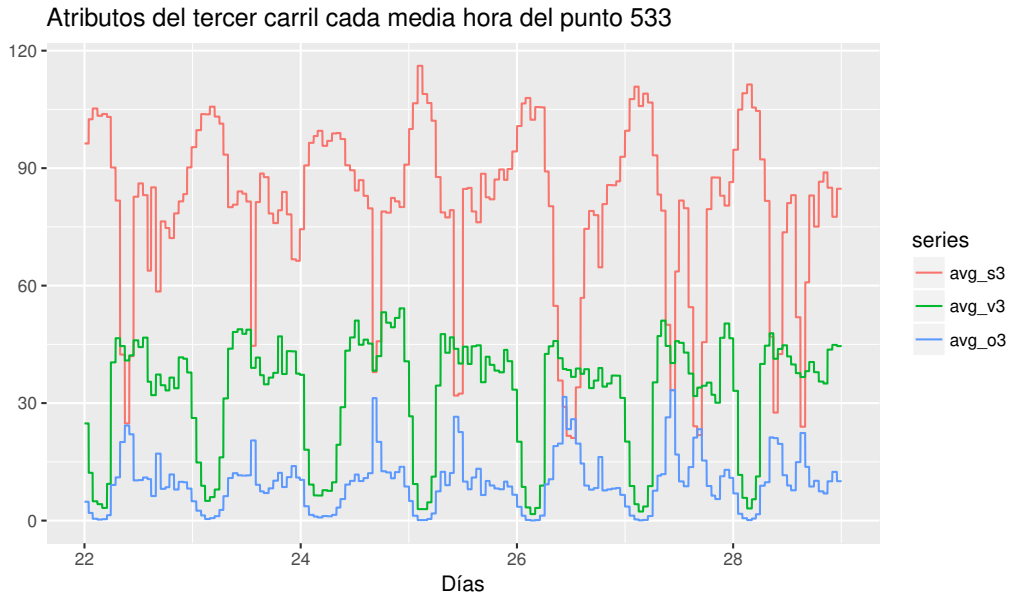

Figura 2.10: Velocidad, Volumen y Ocupación del 3er carril, superpuestos. Punto 533.

2.3.2. Procesamiento de los conjuntos de datos temporales

De las gráficas anteriores (Figuras 2.4, 2.5, 2.6) se observa que en la gráfica 2.4 hay momentos en que el aforo se interrumpe, los dos últimos días en horario de madrugada, cuando el volumen del tráfico es bajo. Para solucionar de una manera eficaz los futuros problemas que puedan surgir, el autor opta por desestimar estos dos últimos días del conjunto de datos ***d363***. También se decide reducir el número de datos escogiendo las medias de ciertas variables, con una frecuencia de media hora, esto favorece a algunos modelos predictivos, como más adelante se verá.

Para la manipulación de series temporales (*Time Series*) existen varias librerías implementadas dentro del ámbito *R*, en el apartado 2.2 se mencionan las que se utilizan a continuación:

Con la librería **lubridate** se es capaz de escoger días concretos o horas concretas, por lo que mediante el uso de los siguientes códigos, se acota el conjunto ***d363***:

```
d363$Tiempo<-ymd_hms(d363$MsgTime)
d363<- mutate(d363,
                yday = yday(Tiempo),
                hour = hour(Tiempo),
                minute = minute(Tiempo),
                halfhour = hour(Tiempo) +
                if_else(minute(Tiempo)>29,0.5,0))

#se acota el conjunto de datos, eliminando los días 27 y 28.
d363 <- subset(d363, yday < 27)
```

Una vez acotado, se generan los datos con formato de serie temporal mediante la función ***ts*** y estableciendo la frecuencia correspondiente, como se calcula en la siguiente parte del código:

```
#Primero se calcula la media de cada atributo, agrupando
# cada 30 min.Luego hará falta.

d363 <-  d363 %>%
  group_by(yday, halfhour) %>%
  mutate(avg_s1 = mean(S1, na.rm = TRUE),
         avg_s2 = mean(S2, na.rm = TRUE),
         avg_s3 = mean(S3, na.rm = TRUE),
         avg_v1 = mean(V1, na.rm = TRUE),
         avg_v2 = mean(V2, na.rm = TRUE),
         avg_v3 = mean(V3, na.rm = TRUE),
         avg_v = mean(V_Total, na.rm = TRUE),
         avg_o1 = mean(O1, na.rm = TRUE),
         avg_o2 = mean(O2, na.rm = TRUE),
         avg_o3 = mean(O3, na.rm = TRUE))

#Establecer la frecuencia para la serie temporal.
freq363 <- round(length(d363$RtmsNo)/(yday(last(d363$Tiempo))
- yday(first(d363$Tiempo))+1))

#Se crea y ordena el conjunto "d363_ts".
d363_ts <- ts(d363[,c(15:19,3:6,20:22,7:9,14,23:26,10:12,27:29)],
              start = 22, frequency = freq363)

#Se generan los distintos atributos con formato ya de "ts"
```

```
t363_S1<-d363_ts[,"S1"]
t363_avg_s1<-d363_ts[,"avg_s1"]
t363_S2<-d363_ts[,"S2"]
t363_avg_s2<-d363_ts[,"avg_s2"]
t363_S3<-d363_ts[,"S3"]
t363_avg_s3<-d363_ts[,"avg_s3"]
t363_V1<-d363_ts[,"V1"]
t363_avg_v1<-d363_ts[,"avg_v1"]
t363_V2<-d363_ts[,"V2"]
t363_avg_v2<-d363_ts[,"avg_v2"]
t363_V3<-d363_ts[,"V3"]
t363_avg_v3<-d363_ts[,"avg_v3"]
t363_V3<-d363_ts[,"V3"]
t363_avg_v3<-d363_ts[,"avg_v3"]
t363_V<-d363_ts[,"V_Total"]
t363_avg_v<-d363_ts[,"avg_v"]
t363_O1<-d363_ts[,"O1"]
t363_avg_o1<-d363_ts[,"avg_o1"]
t363_O2<-d363_ts[,"O2"]
t363_avg_o2<-d363_ts[,"avg_o2"]
t363_O3<-d363_ts[,"O3"]
t363_avg_o3<-d363_ts[,"avg_o3"]
```

Ahora ya se pueden calcular las medias de los atributos con una frecuencia establecida y crear nuevos conjuntos de datos con intervalos de media hora.

```
#Primero se implementa el conjunto de datos con observaciones cada 30 min,
# con esta propuesta:

data363 <- subset(d363, minute == 0 | minute == 59 |
minute == 30 | minute == 31)

# Finalmente se repiten los pasos anteriores con
# el nuevo conjunto.
freqdata363<-round(length(data363$RtmsNo)/(yday(last(data363$Tiempo))
- yday(first(data363$Tiempo))+1))
data363_ts<-ts(d363[,c(15:19,3:6,20:22,7:9,14,23:26,10:12,27:29)],
start = 22, frequency = freq363)

t363_30min_S1<-data363_ts[,"S1"]
```

```
t363_30min_avg_s1<-data363_ts[,"avg_s1"]
t363_30min_S2<-data363_ts[,"S2"]
t363_30min_avg_s2<-data363_ts[,"avg_s2"]
t363_30min_S3<-data363_ts[,"S3"]
t363_30min_avg_s3<-data363_ts[,"avg_s3"]
t363_30min_V1<-data363_ts[,"V1"]
t363_30min_avg_v1<-data363_ts[,"avg_v1"]
t363_30min_V2<-data363_ts[,"V2"]
t363_30min_avg_v2<-data363_ts[,"avg_v2"]
t363_30min_V3<-data363_ts[,"V3"]
t363_30min_avg_v3<-data363_ts[,"avg_v3"]
t363_30min_V3<-data363_ts[,"V3"]
t363_30min_avg_v3<-data363_ts[,"avg_v3"]
t363_30min_V<-data363_ts[,"V_Total"]
t363_30min_avg_v<-data363_ts[,"avg_v"]
t363_30min_O1<-data363_ts[,"O1"]
t363_30min_avg_o1<-data363_ts[,"avg_o1"]
t363_30min_O2<-data363_ts[,"O2"]
t363_30min_avg_o2<-data363_ts[,"avg_o2"]
t363_30min_O3<-data363_ts[,"O3"]
t363_30min_avg_o3<-data363_ts[,"avg_o3"]
```

Finalmente el resultado se observa en la siguiente figura 2.11, donde se tienen, en negro, los datos con frecuencia de 2 min, en rojo, las medias de cada media hora, y en naranja los puntos del nuevo conjunto de datos. Esta operación se puede realizar con cualquiera de los otros atributos.

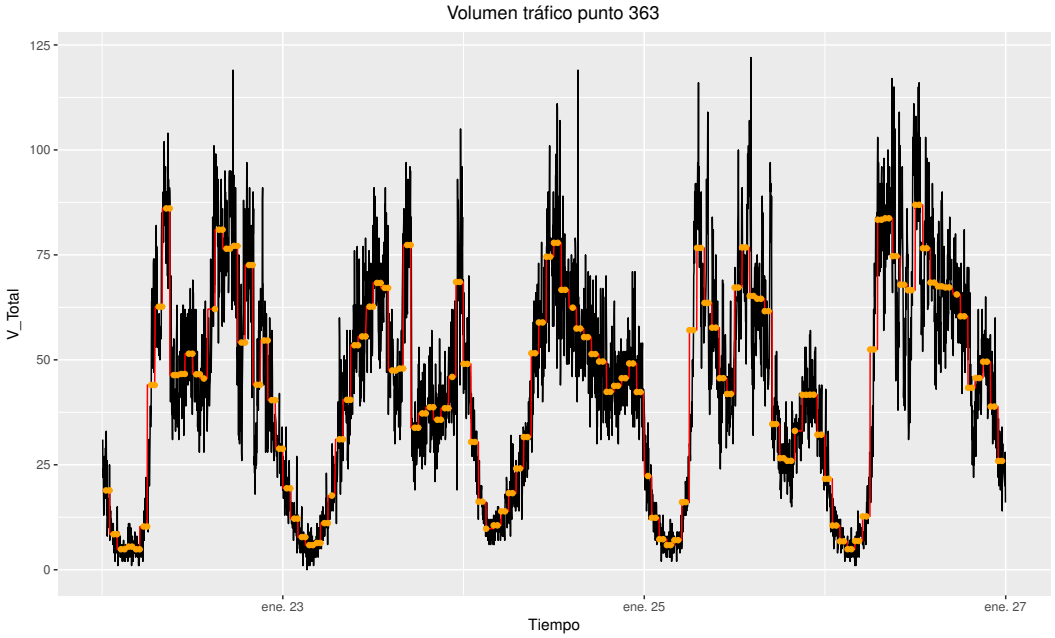

Figura 2.11: Datos de Volumen total de vehículos, cada dos minutos (Negro), media (Rojo), cada media hora (Naranja). Punto 363

Una vez los datos de entrada están preparados ya se puede empezar a comprobar el ajuste de los modelos propuestos a continuación.

2.4. Modelos estadísticos

En esta sección se expondrán los principales modelos estadísticos utilizados durante el desarrollo del trabajo, haciendo una breve explicación de cada uno de ellos, y posteriormente, mostrando el código implementado para la obtención del modelo.

Por hacer una introducción a los modelos estadísticos, mencionar que se trata de un modelo matemático que utiliza la probabilidad y incluye algunas aceptaciones sobre la generación de muestras para parecerse a los datos de un conjunto mayor, es decir, datos que se recogen de un muestreo y se suponen aleatorios, muestran una forma de comportamiento explicada por las muestras recogidas anteriormente.

Finalmente, un modelo estadístico queda especificado por un conjunto de ecuaciones que relacionan diversas variables aleatorias, y en las que pueden

aparecer otras variables no aleatorias.

Nota: Los códigos que se verán en este apartado son ejemplos del código aplicados solamente al atributo del Volumen Total del punto de aforo 533 2.12, aunque en el código general este proceso se haya realizado mediante bucles para cada uno de los atributos y conseguir de una manera eficaz la posterior comparación de resultados.

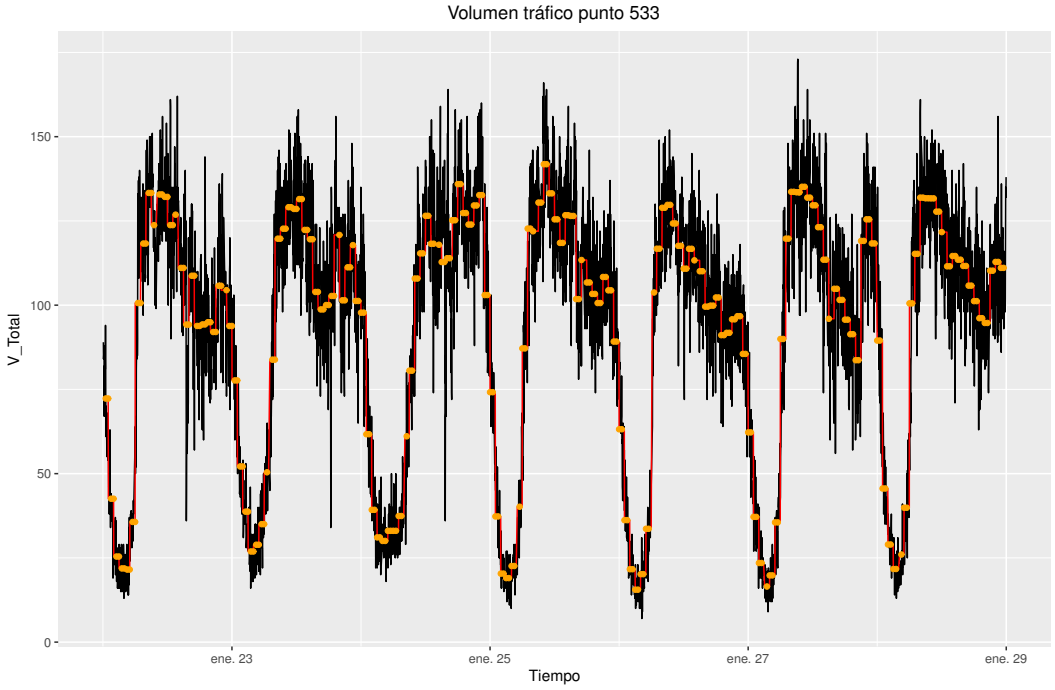

Figura 2.12: Datos de Volumen total de vehículos, cada dos minutos (Negro), media (Rojo), cada media hora (Naranja). Punto 533

Además se examinan los distintos grupos de datos, tanto los originales preprocesados ($f = 2$ min) como los reducidos a dato cada 30 min y así finalmente contrastar resultados. También comentar que se expone el código utilizado para el cálculo del error cuadrático medio (ECM) y los criterios de AIC y BIC, explicados en el punto 2.5 más adelante.

2.4.1. ARIMA

La palabra **ARIMA** es la abreviación de *Modelos Autorregresivos Integrados de Medias Móviles*. Viene del conjunto de los modelos autoregresivos (AR) con el de las medias móviles (MA) y el termino integrados (I), que se refiere a la posibilidad de realizar una diferenciación entre los elementos de la muestra.

AR: Un modelo se define como autorregresivo si la variable endógena de un período t se explica por las observaciones de ella misma haciendo correspondencia a períodos anteriores, añadiéndose, como en los modelos estructurales, un término de error. En procesos estacionarios con distribución normal, bajo determinadas condiciones previas, la teoría estadística de los procesos estocásticos dice que toda Y_t puede expresarse como una combinación lineal de sus valores pasados más un término de error.

Los modelos aotorregresivos se abrevian con la palabra AR tras la que se indica el orden p del modelo: AR(1), AR(2),....etc. El orden del modelo expresa el número de observaciones retasadas de las series temporales analizadas que intervienen en la ecuación. Un modelo AR(p) tendría la siguiente expresión:

$$Y_t = c + \sum_{i=1}^{p} f_i Y_{t-i} + \epsilon_t \tag{2.1}$$

Donde c es una constante, $f_1, ..., f_p$ son los parámetros del modelo y ϵ_t el término de error de los modelos de este tipo que se denominan generalmente **ruido blanco** cuando cumple las tres hipótesis básicas tradicionales:

- Media nula
- Varianza constante
- Covarianza nula entre errores correspondientes a observaciones diferentes

En el método ARIMA este término se trata dentro del modelo de medias móviles.

MA: Un modelo de los denominados de medias móviles es aquel que explica el valor de una determinada variable en un período t en función de un término independiente y una sucesión de términos de error, de innovaciones correspondientes a períodos precedentes, convenientemente ponderados. Estos modelos se denotan normalmente con las siglas MA, seguidos, como en el caso

de los modelos autorregresivos, del orden entre paréntesis. Así, un modelo con **q** términos de error MA(q) respondería a la siguiente expresión:

$$Y_t = \epsilon_t + \sum_{i=1}^{q} g_i \epsilon_{t-i} \tag{2.2}$$

Donde $g_1, ..., g_q$ corresponde a los parámetros del modelo y $\epsilon_t, \epsilon_{t-1}...$ son los términos de error.

Entonces el resultado de un modelo ARMA(p,q) quedaría de la siguiente forma:

$$Y_t = \epsilon_t + \sum_{i=1}^{p} f_i Y_{t-i} + \sum_{i=1}^{q} g_i \epsilon_{t-i} \tag{2.3}$$

Y finalmente, **I:** Corresponde a la componente integrada del modelo, se representa por un orden entre paréntesis denotado por la letra **d**, corresponde a las diferencias que son necesarias para transformar la serie original en estacionaria.

$$\Delta Y_t = Y_t - Y_{t-1} \tag{2.4}$$

Y así el modelo ARIMA (p,d,q) queda representado como:

$$Y_t = -(\Delta^d Y_t - Y_t) + f_0 + \sum_{i=1}^{p} f_i \Delta^d Y_{t-i} + \epsilon_t - \sum_{i=1}^{q} g_i \epsilon_{t-i} \tag{2.5}$$

La idea básica del análisis de series consiste en que cada uno de estos componentes de las series puede ser analizado de forma separada para posteriormente, agregar los análisis parciales en un resultado conjunto.

En ocasiones, el análisis prioriza, se centra sólo en alguno de los componentes sistemáticos por separado (la tendencia, la estacionalidad, el ciclo), en otras ocasiones, como es el caso de la modelización ARIMA, lo que interesa es ir más allá de las componente cíclicas, tendenciales y estacionales, analizando la componente no sistemática, de carácter aparentemente aleatorio, para tratar de identificar algún patrón de interés en su evolución que ayude a entender la progresión de la serie completa.

Así pues, la aplicación de modelos ARIMA suele realizarse por descomposición, analizando en primer lugar la tendencia de la serie, pasando después a observar la estacionalidad y concentrándose después en la identificación del

componente filtrado de tendencia y estacionalidad.

Para ajustar mejor el modelo existen dos funciones que ayudan a determinar los términos *"p"* y *"q"*, estas son la función de auto-correlación (**ACF**) y la función de auto-correlación parcial (**PACF**).

ACF mide la correlación entre las observaciones de una serie temporal separadas por k unidades de tiempo (y_t e y_{t-k}). La función de auto-correlación parcial hace lo mismo pero después de ajustarse para la presencia de los demás términos de desfase más corto (y_{t-1}, y_{t-2}, ..., y_{t-k-1}).

Para el ajuste es necesaria una interpretación de ambos diagramas, que se explican en el siguiente cuadro (2.2)

	ACF	**PACF**
AR	Decrece con el retardo de forma exponencial	No es significativa para retardos > p
MA	No es significativa para retardos > q	Decrece con el retardo de forma exponencial
ARMA	Decrecimiento geométrico dependiente del parámetro autorregresivo	Decrecimiento geométrico dependiente del parámetro de media móvil

Cuadro 2.2: Interpretación de los diagramas ACF y PACF

A continuación se muestra el código utilizado para el ajuste del modelo ARIMA.

Código: Primero se ajusta la serie temporal a un modelo mediante el uso de la función **auto.arima**, que determina los términos (p, d, q) automáticamente reduciendo los errores y basándose en las ACF y PACF. Contrastamos los datos cada dos minutos, con otros de la misma frecuencia y la media del atributo cada media hora, no se espera una gran mejora pero se realiza un test para ver si vale la pena, posteriormente se comprueba también con los datos cada 30 min.

```
#Antes se definen los tamaños de las distintas muestras
n <- length(t533_V)
n_ <- length(t533_30min_V)

# Se calculan los ajustes de los tres conjuntos de datos escogidos.
arimaV_533 <- auto.arima(t533_V) # 2 min
arimaVm_533 <- auto.arima(t533_avg_v) # 2 min con media
arimaV_533ok <- auto.arima(t533_30min_avg_v) # 30 min
```

Para comprobar el ajuste del módelo predictivo generado con el método ARIMA, se carga el paquete ***forecast*** y se utiliza el siguiente código:

```
fcast_arimaV_533 <- forecast(arimaV_533, h = 2*freq533)
fcast_arimaVm_533 <- forecast(arimaVm_533, h = 2*freq533)
fcast_arimaV_533ok <- forecast(arimaV_533ok, h = 2*freqdata533)

parima_V_533 <- autoplot(fcast_arimaV_533)
parima_Vm_533 <- autoplot(fcast_arimaVm_533)
parima_V_533ok <- autoplot(fcast_arimaV_533ok)
```

Donde **h** indica el número de valores a extrapolar (en este caso dos días)

Para calcular el error cuadrático medio:

```
error_arima_V <- fcast_arimaV_533$residuals
error_arima_Vm <- fcast_arimaVm_533$residuals
error_arima_Vok <- fcast_arimaV_533ok$residuals

ecm_arimaV <- sum(error_arima_V^2)/n
ecm_arimaVm <- sum(error_arima_Vm^2)/n
ecm_arima_Vok <- sum(error_arima_Vok^2)/n_
```

Nota: en el punto 2.5 se muestra la función implementada "ECM" para agilizar los cálculos.

Para obtener el AIC y el BIC:

```
arimaV_533$aic
arimaVm_533$aic
arimaV_533ok$aic

arimaV_533$bic
arimaVm_533$bic
arimaV_533ok$bic
```

Como se verá en el punto 2.5 el que menor resultado que se obtenga de los indicadores expuestos significa que el ajuste es mejor.

2.4.2. S-ARIMA

En este caso a la palabra **ARIMA** se le añade la **S** de *Seasonal (Estacionalidad)*, ya que consiste en modalizar no solo la componente regular sino también la componente *estacional*. Este modelo, es parecido al explicado anteriormente y se representa con la siguiente expresión:

$$\boldsymbol{ARIMA\ (p,d,q)\ *\ SARIMA\ (P,D,Q)_S}$$

Donde la primera parte corresponde a la parte regular y la segunda a la estacionalidad. P,D y Q se refieren al numero de *lags* (retardos) escogidos dentro de la estacionalidad S impuesta.

Un indicador de que la serie temporal tiene una componente de estacionalidad es observar los resultados del ACF *(Autocorrelation Function)* y el PACF *(Partial Autocorrelation Function)*, ver si la se muestran indicios de la existencia de un patrón. Este patrón indicará la estacionalidad (S).

Código: Para este apartado se necesita saber los valores de los parámetros del ajuste ARIMA y la estacionalidad, estos valores se pueden obtener simplemente observando los modelos.

```
parima_V_533 # dice los valores
parima_V_533ok # se desestima el conjunto con las medias de atributo,
  al no mostrar resultados más favorables que con un dato cada media hora.
```

Una vez obtenidos los datos p, d y q, se añaden a las nuevas funciones además de la estacionalidad S, y los términos P, D y Q, que con un proceso de ensayos se obtienen los valores óptimos al comprobar los criterios AIC.

```
V_sarima <- sarima(t533_V,3,0,1,0,1,1,freq533)
Vok_sarima <- sarima(t533_30min_avg_v,3,0,1,1,0,1,freqdata533)
V_sarima$AIC
Vok_sarima$AIC
```

Después, escogidos los parámetros que ajustan mejor el modelo, se ejecuta la función ***forecast*** con la predicción para dos días.

```
fcast_sarima_V <- forecast(t533_V, h = 2*freq533)
fcast_sarima_Vok <- forecast(t533_30min_avg_v, h = 2*freqdata533)
psarima_V_533 <- autoplot(fcast_sarima_V)
psarima_Vok_533 <- autoplot(fcast_sarima_Vok)
```

Y finalmente los indicadores, como ya anteriormente se ha calculado el *AIC*, solamente queda obtener los errores cuadráticos medios y el BIC.

```
# ECM
error_sarima_V <- fcast_sarima_V$residuals
ecm_sarima_V <- sum(error_sarima_V^2)/n

error_sarima_Vok <- fcast_sarima_Vok$residuals
esarima_Vok <- sum(error_sarima_Vok^2)/n_

# BIC
V_sarima$BIC
Vok_sarima$BIC
```

2.4.3. Modelo de Regresión Dinámico - Armónico

Con este método de se pueden extraer y extrapolar ajustes a regresiones dependientes del tiempo, ya que mediante coeficientes de fourier correspondientes a los armónicos se explica un porcentaje determinado de la varianza o si no mediante los coeficientees de fourier correspondientes a las bajas frecuencias.

Las primeras aplicaciones en el análisis espectral de series temporales se aplicaron en economía por Nerlove (1964) y Granger (1969). El uso del análisis espectral requiere un cambio en el modo de ver las series económicas, al pasar de la perspectiva del tiempo al dominio de la frecuencia. El análisis espectral parte de la suposición de que cualquier serie Yt, puede ser transformada en ciclos formados con senos u cosenos:

$$Y_t = \eta + \sum_{j=1}^{N} [a_j \cos(2\pi \frac{ft}{n}) + b_j \sin(2\pi \frac{ft}{n})] \tag{2.6}$$

donde η es la media de la serie, a_j y b_j son su amplitud, f son las frecuencias que del conjunto de las n observaciones, t es un indice de tiempo que va de 1 a N, siendo N el numero de periodos para los cuales tenemos observaciones en el conjunto de datos, el cociente $\frac{ft}{n}$ convierte cada valor de t en escala de tiempo en proporciones de $2n$ y rango j desde 1 hasta n siendo $n = \frac{N}{2}$ (es decir, 0,5 ciclos por intervalo de tiempo). Las dinámica de las altas frecuencias (los valores más altos de f) corresponden a los ciclos cortos en tanto que la dinámica de la bajas frecuencias (pequeños valores de f) van a corresponder con los ciclos largos. Si nosotros hacemos que $\frac{ft}{n} = w$ la ecuación 2.6 quedaría,

así :

$$Y_t = \eta + \sum_{j=1}^{N} [a_j \cos(\omega_j) + b_j \sin(\omega_j)] \tag{2.7}$$

El análisis espectral puede utilizarse para identificar y cuantificar en procesos aparentemente no periódicos, sucesiones de ciclos de periodo de corto y largo plazo. Una serie dada Y_t puede contener diversos ciclos de diferentes frecuencias y amplitudes, y esa combinación de frecuencias y amplitudes de carácter cíclico la hacer aparecer como un serie no periódica e irregular. De hecho la ecuación 2.7, muestra que cada observación t de una serie de tiempo, es el resultado sumar los valores en t que resultan de N ciclos de diferente longitud y amplitud, a los que habría que añadir si cabe un termino de error.

Según Harvey (1978), existe una manera de pasar desde el dominio del tiempo al dominio de la frecuencia, multiplicando previamente los datos originales por una matriz ortogonal, W que él sugiere con el elemento (j,t):

$$w_{jt} = \begin{cases} \left(\frac{1}{T}\right)^{\frac{1}{2}} & \forall j = 1 \\ \left(\frac{2}{T}\right)^{\frac{1}{2}} \cos\left[\frac{\pi j(t-1)}{T}\right] & \forall j = 2, 4, 6, .. \frac{(T-2)}{(T-1)} \\ \left(\frac{2}{T}\right)^{\frac{1}{2}} \sin\left[\frac{\pi (j-1)(t-1)}{T}\right] & \forall j = 3, 5, 7, .. \frac{(T-2)}{T} \\ \left(\frac{1}{T}\right)^{\frac{1}{2}} (-1)^{t+1} & \forall j = T \end{cases} \tag{2.8}$$

La matriz W tiene la ventaja de ser ortogonal por lo que $WW^T = I$.

Y una vez acabada la introducción al método, se añade a continuación el código, en el cual se han ido comprobando distintos coeficientes de fourier y finalmente se comprueba cual de todos es el que obtiene un mejor resultado.

Código: En este punto se muestra el código utilizado para calcular los modelos y comprobar cual es el mejor coeficiente que ajuste el modelo, tanto para los datos con frecuencia cada 2 minutos como para los que tienen una observación cada 30 min.

```
# Datos con observaciones cada dos minutos #
# Da error con lambda=0, por lo que se pasa a 1 #
fitV_1 <- auto.arima(t533_V, xreg = fourier(t533_V, K = 1),
seasonal = FALSE, lambda = 1)
fitV_2 <- auto.arima(t533_V, xreg = fourier(t533_V, K = 2),
seasonal = FALSE, lambda = 1)
```

```
fitV_3 <- auto.arima(t533_V, xreg = fourier(t533_V, K = 3),
seasonal = FALSE, lambda = 1)
fitV_4 <- auto.arima(t533_V, xreg = fourier(t533_V, K = 4),
seasonal = FALSE, lambda = 1)
fitV_5 <- auto.arima(t533_V, xreg = fourier(t533_V, K = 5),
seasonal = FALSE, lambda = 1)
fitV_6 <- auto.arima(t533_V, xreg = fourier(t533_V, K = 6),
seasonal = FALSE, lambda = 1)
#fitV_7 <- auto.arima(t533_V, xreg = fourier(t533_V, K = 7),
seasonal = FALSE, lambda = 1)
#fitV_8 <- auto.arima(t533_V, xreg = fourier(t533_V, K = 8),
seasonal = FALSE, lambda = 1)
#fitV_9 <- auto.arima(t533_V, xreg = fourier(t533_V, K = 9),
seasonal = FALSE, lambda = 1)
```

Una vez calculados los ajustes para distintos coeficientes de Fourier, se procede a la visualización y obtención de cual goza de un menor valor de AIC.

```
AIC_V<-c(fitV_1$aic,fitV_2$aic,fitV_3$aic,fitV_4$aic,fitV_5$aic,
fitV_6$aic)#fitV_7$aic,fitV_8$aic,fitV_9$aic)

plot(AIC_V)

which.min(AIC_V)
```

Hallado el coeficiente, el siguiente paso corresponde a generar la predicción del modelo y posterioermente sus errores. Ejemplo para K = 3.

```
# Forecast y plot
pMdA_V_533 <- fitV_3 %>% forecast(xreg = fourier(t533_V, K = 3,
h=2*freq533)) %>% autoplot() + xlab("Día") +
ylab("Volumen Vehículos total - punto 533")

# ECM
ecm_harm_V <- fitV_3$residuals
eharm_V <- sum(ecm_harm_V^2)/n

# BIC
BIC_harm_V <- fitV_3$BIC
```

De seguido se prueba con el atributo con datos cada 30 min:

```
# Datos con observaciones cada treinta minutos #
fitVm_1 <- auto.arima(t533_30min_avg_v, xreg = fourier(t533_30min_avg_v,
 K = 1),seasonal = FALSE, lambda = 0)
fitVm_2 <- auto.arima(t533_30min_avg_v, xreg = fourier(t533_30min_avg_v,
 K = 2),seasonal = FALSE, lambda = 0)
fitVm_3 <- auto.arima(t533_30min_avg_v, xreg = fourier(t533_30min_avg_v,
 K = 3),seasonal = FALSE, lambda = 0)
fitVm_4 <- auto.arima(t533_30min_avg_v, xreg = fourier(t533_30min_avg_v,
 K = 4),seasonal = FALSE, lambda = 0)
fitVm_5 <- auto.arima(t533_30min_avg_v, xreg = fourier(t533_30min_avg_v,
 K = 5),seasonal = FALSE, lambda = 0)
fitVm_6 <- auto.arima(t533_30min_avg_v, xreg = fourier(t533_30min_avg_v,
 K = 6),seasonal = FALSE, lambda = 0)
#fitVm_7 <- auto.arima(t533_30min_avg_v, xreg = fourier(t533_30min_avg_v,
K = 7),seasonal = FALSE, lambda = 0)
#fitVm_8 <- auto.arima(t533_30min_avg_v, xreg = fourier(t533_30min_avg_v,
 K = 8),seasonal = FALSE, lambda = 0)
#fitVm_9 <- auto.arima(t533_30min_avg_v, xreg = fourier(t533_30min_avg_v,
 K = 9),seasonal = FALSE, lambda = 0)

# se han comentado, para no sobrecargar el proceso de cálculo,
# ya que si son necesarios se pueden usar, una vez se vea que el
# modelo mejora mostrando una tendencia al alza de los coeficientes
# de Fourier
```

Igualmente al caso anterior, se obtiene el mejor ajuste dependiendo del coeficiente, su predicción y finalmente los errores.

```
AIC_Vm<-c(fitVm_1$aic,fitVm_2$aic,fitVm_3$aic,fitVm_4$aic,
fitVm_5$aic,fitVm_6$aic)#fitVm_7$aic,fitVm_8$aic,fitVm_9$aic )

plot(AIC_Vm)

which.min(AIC_Vm)#min en k=3

# Forecast y plot
pMdA_Vm_533 <- fitVm_3 %>%
forecast(xreg = fourier(t533_30min_avg_v, K = 3, h=2*freqdata533)) %>% autoplot() +
```

```
# ECM
ecm_harm_Vm <- fitVm_3$residuals
eharm_Vm <- sum(ecm_harm_Vm^2)/n_

# BIC
BIC_harm_V <- fitV_3$BIC
```

Nota: Para agilizar y reducir los cálculos a realizar se optimiza el código creando una función que tiene como objetivo ir haciendo ajustes aumentando el coeficiente de Fourier (**K**) al mismo tiempo que compara los valores de del AIC (Ver 2.5). La función se detiene al obtener tres ajustes no mejores después de encontrar el óptimo y selecciona este como mejor, mostrando finalmente las gráficas de los valores AIC para cada coeficiente y la predicción para los dos días posteriores al fin de la serie temporal. También devuelve una oración que dice el coeficiente K escogido y su valor AIC correspondiente.

Función:

```
Armonico <- function(x)
  AIC_armonico <- c()
  m <- 1
  i <- 1
  fr <- 2*round(length(x)/7) # para 533 y 534
  #fr <- 2*round(length(x)/5) # para 363
  while( i - m < 4 ) # bucle hasta que haga tres
    # comprobaciones despues de encontrar un mínimo
    fit <- auto.arima(x, xreg = fourier(x, K = i),
                      seasonal = FALSE, lambda = 0)
    AIC_armonico[i] <- fit$aic
    m <- which.min(AIC_armonico)
    i <- i + 1

  plot(AIC_armonico)
  fit <- auto.arima(x, xreg = fourier(x, K = m),
                    seasonal = FALSE, lambda = 0)
  autoplot(forecast(fit, xreg = fourier(x, K = m),
                    h = fr))
  return(paste("El coeficiente de Fourier que mejor
               se ajusta a la muestra es, K = ", m,"
```

```
                con un valor del AIC de ",
                round(AIC_armonico[m],digits = 4),  sep=""))
```

Finalmente se explica el último método estadístico utilizado en el presente trabajo.

2.4.4. TBATS

TBATS consiste en un modelo, implementado por de Livera, Hyndman & Snyder [14] que permite una estacionalidad dinámica, no obliga a que sea periódica como ocurre en el modelo ARIMA, es decir, sirve para series temporales con estacionalidades múltiples y complejas.

TBATS es el anacronismo de:

T de regresiones trigonométricas a modelos multi-estacionales (*trigonometric regressors to model multiple-seasonalities*)

B de transformación Box-cox (*Box-Cox transformations*)

A de errores ARMA (*ARMA errors*)

T de tendencia (*trend*)

S de estacionalidad (*seasonality*)

El código es simple ya que esta función **tbats()** se encuentra dentro del paquete ***forecast***. se añade la posibilidad de poder escoger la ventana, rango, de datos que se elija en el comando ***window***. A continuación se muestra el ejemplo con el atributo del volumen total de coches en el punto 533:

```
ptbats_V_533 <- t533_V %>% window(start=22)%>% tbats() %>%
 forecast() %>%autoplot() +xlab("Día") +
 ylab("número total de vehículos - punto 533")

ptbats_Vok_533 <- t533_30min_avg_v %>% window(start=22)%>%
tbats() %>% forecast() %>%autoplot()+
xlab("Día") + ylab("número total de vehículos - punto 533")
```

2.5. Indicadores para establecer el modelo óptimo

En este apartado se comentan los posibles indicadores existentes dentro del ámbito de la predicción de series temporales y de regresión de las mismas que nos indique cual de los modelos utilizados se ajusta mejor a la realidad.

Uno de los más conocidos, es del **error cuadrático medio (ECM)**, resta a un vector de tamaño n llamado de predicciones ($\hat{Y}_i$), un vector con los valores reales (Y_i), explicado de otra forma, a los valores obtenidos por el modelo, se les resta los verdaderos valores. Luego se suma el cuadrado de esta diferencia y finalmente se divide por el tamaño del vector n. Se ve más directo en la siguiente ecuación:

$$ECM = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2 \tag{2.9}$$

En modelos predictivos, este valor evalúa la calidad del modelo en cuanto a su variación y su grado de sesgo.Pero el problema es que este indicador no tiene en cuenta la estructura estocástica del modelo, no informan sobre alguna característica estocástica supuesta sobre el período extramuestral.

Se crea la función **ECM** que calcula el resultado del error cuadrático medio a partir del vector residual "x":

```
#Error cuadrático medio
ECM <- function(x)
  ECM <- 0 #asignar 0 al error
  #x corresponde al vector residuales/errores
  return(sum(x^2)/length(x))
```

También existen varios criterios para la seleccion de modelos como el **Criterio de Información Bayesiano (BIC)** o el **Criterio de Información de Akaike (AIC)**. Ambos criterios están basados en la función de probabilidad del ajuste del modelo, con el plus de que ambos añaden un término de penalización para el número de parámetros que se introducen, para así evitar sobreajustes. Entre los dos criterios, el BIC es el que tiene un término de penalización mayor.

La fórmula del BIC:

$$-2 \cdot \ln p(x|M) \approx \text{BIC} = -2 \cdot \ln \hat{L} + k \ln(n) \tag{2.10}$$

Y la fórmula para AIC es:

$$AIC = 2k - 2\ln(\hat{L}) \tag{2.11}$$

donde:

- $x \rightarrow$ corresponde a los datos obtenidos;
- $n \rightarrow$ Es el tamaño de la muestra;
- $k \rightarrow$ el número de parámetros libres a ser estimados;
- $p(x|M) \rightarrow$ La probabilidad marginal de los datos observados dado el modelo M;
- $\hat{L} \rightarrow$ El máximo valor de la función de verosimilitud del modelo M, i.e. $\hat{L} = p(x|\hat{\theta}, M)$, donde $\hat{\theta}$ son los valores de los parámetros que maximizan la función de verosimilitud.

Estos criterios no proporcionan información en sentido absoluto, es necesario mínimo dos modelos para poder compararlos y elegir el que menor valor obtenga como mejor ajuste.

En los paquetes cargados, una vez se realiza el ajuste ya aparecen estas dos opciones, tanto AIC como BIC y en el caso del error cuadrático medio se calcula en base al error que proporciona el modelo a través de los residuales, elevándolos al cuadrado y dividiendo por el tamaño del atributo.

Se añade la función implementada para comparar los ajustes de los modelos generados con **auto.arima()**, ya que añadiendo el conjunto de matrices formado por el modelo para cada atributo (i.e. Arima_533) devuelve un diagrama de barras con los valores de AIC de cada atributo y los atributos ordenados de menor valor de AIC a mayor:

```
# Archiva AIC arima
ArimaAIC <- function(x)
  x <- x[6:length(x)] # 1:5 corresponden a atributos de Tiempo
  atrib <- names(x) # vector con los nombres de los atributos
  n <- length(atrib) # número de componentes del vector
```

```
Valores_AIC <- c()
for(i in 1:n)
  Valores_AIC[i] <- x[[i]]$aic

barplot(Valores_AIC, names.arg = atrib, col = "blue",
        density = 50, ylab = "AIC", xlab ="Atributos")
return(atrib[order(Valores_AIC)])
```

Ahora una vez explicados los criterios que se siguen para la elección de los mejores modelos y los mejores atributos para ser extrapolados/predichos con mayor acierto, se da comienzo al capítulo dónde se mostraran los resultados obtenidos y los contrastes realizados, para así posteriormente pasar al apartado de las conclusiones.

Capítulo 3

Resultados

En este apartado se muestra un breve resumen de los resultados más significativos.

En el código se recogen todas las variables mediante la función **lapply** (para el caso ARIMA y TBATS), pero al ser 3 puntos de aforo por 9 atributos de partida más 9 generados para ganar más información, acaban siendo 54 procesos para cada modelo, lo que genera que el computador deba utilizar una cantidad de energía y tiempo considerables. Existe la posibilidad de correr solo las líneas del código que interesen, como se ha mostrado en el caso del Volumen total del punto 533.

A continuación se añaden los gráficos de las predicciones generados por los distintos modelos anteriormente expuestos, para así comprobar los resultados de una manera visual, seguidamente se comparan los residuales para ir sacando las conclusiones oportunas:

3.1. ARIMA

Se ha extrapolado la serie a un día más, ya que con dos días el resultado a partir del segundo era constante. Para el método ARIMA, usado con los datos de partida **parima_V_533**, el resultado es:

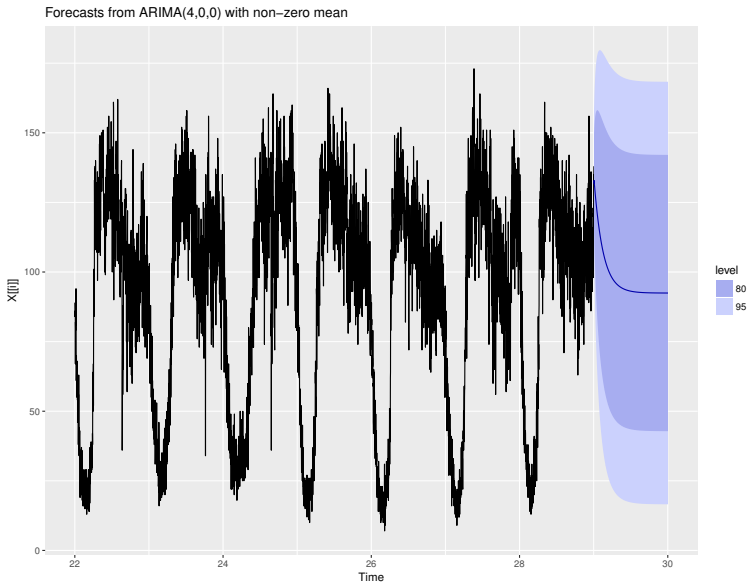

Figura 3.1: Forecast mediante el método Arima, del atributo "Volumen Total" sin cuartar - Punto 533

Con los datos de una misma frecuencia pero valores medios cada media hora, **parima_Vm_533**, el resultado muestra esta forma:

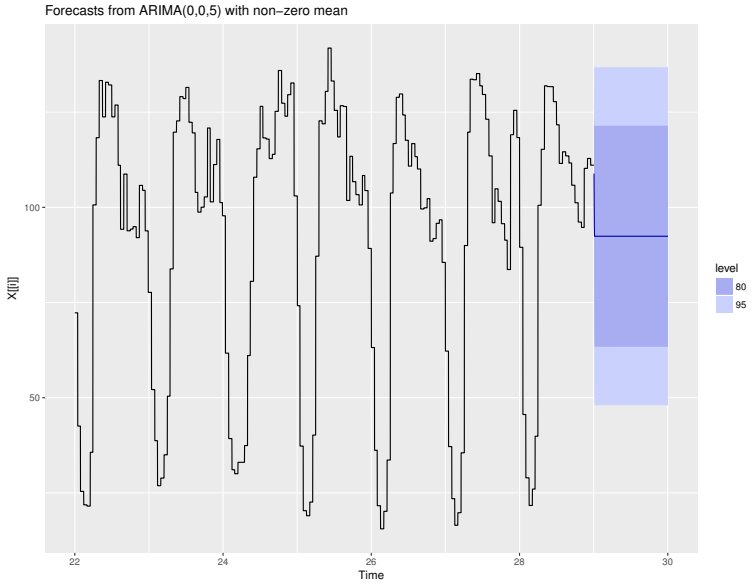

Figura 3.2: Forecast mediante el método Arima, del atributo "V_Total" misma frecuencia, valores medios - Punto 533

A partir de este momento se desestima el seguir haciendo cálculos con las frecuencias de 2 min y las medias cada media hora, ya que el ajuste no presenta mejoras sustanciales comparado con el conjunto de datos que ahora se muestra (Figura 3.3. El método ARIMA con los datos modificados a una observación cada 30 min,**parima_V_533ok** el resultado es:

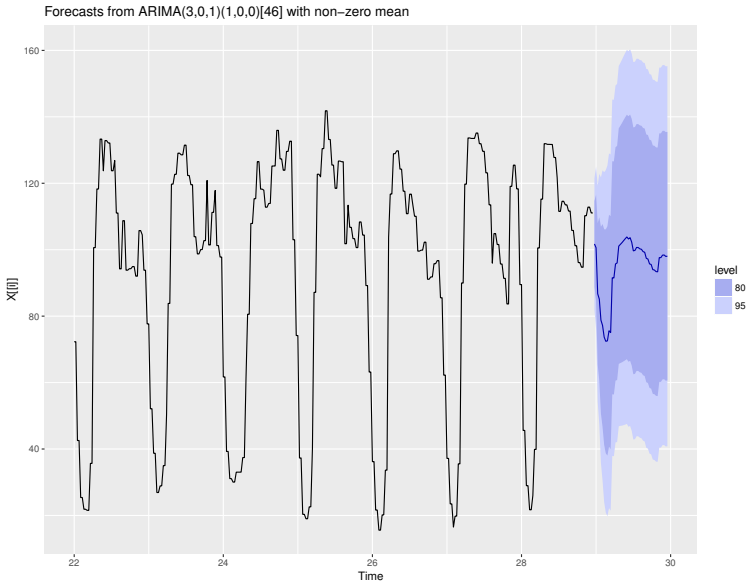

Figura 3.3: Forecast mediante el método Arima, del atributo "Volumen Total" con datos cada 30 minutos - Punto 533

En cuanto a los residuales y valores de los criterios se extrae la siguiente tabla:

Valores residuales	ECM	AIC	BIC
Arima 2 min - Original	160.3085	37658.31	37697.11
Arima 2 min - Media	37.9398	30808.31	30853.58
Arima 30 min - Media	100.0577	2409.86	2436.26

Cuadro 3.1: Valores obtenidos de los indicadores ECM, AIC y BIC para los tres ejemplos con el atributo correspondiente al volumen total de vehículos del punto 533. Con modelo auto.arima()

Aunque el resultado en el ECM sea menor no demuestra que sea un mejor ajuste, hay que fijarse en todos los indicadores, ya que los valores que dan los criterios AIC y BIC son más fidedignos para elegir un mejor modelo.

Ahora se adjuntan los resultados obtenidos mediante la función **ArimaAIC ()** y **ArimaBIC ()** que compara y ordena los atributos según su valor de AIC y BIC de un mismo punto de aforo, ya sea cada 2 min, o cada 30 min (*light*).

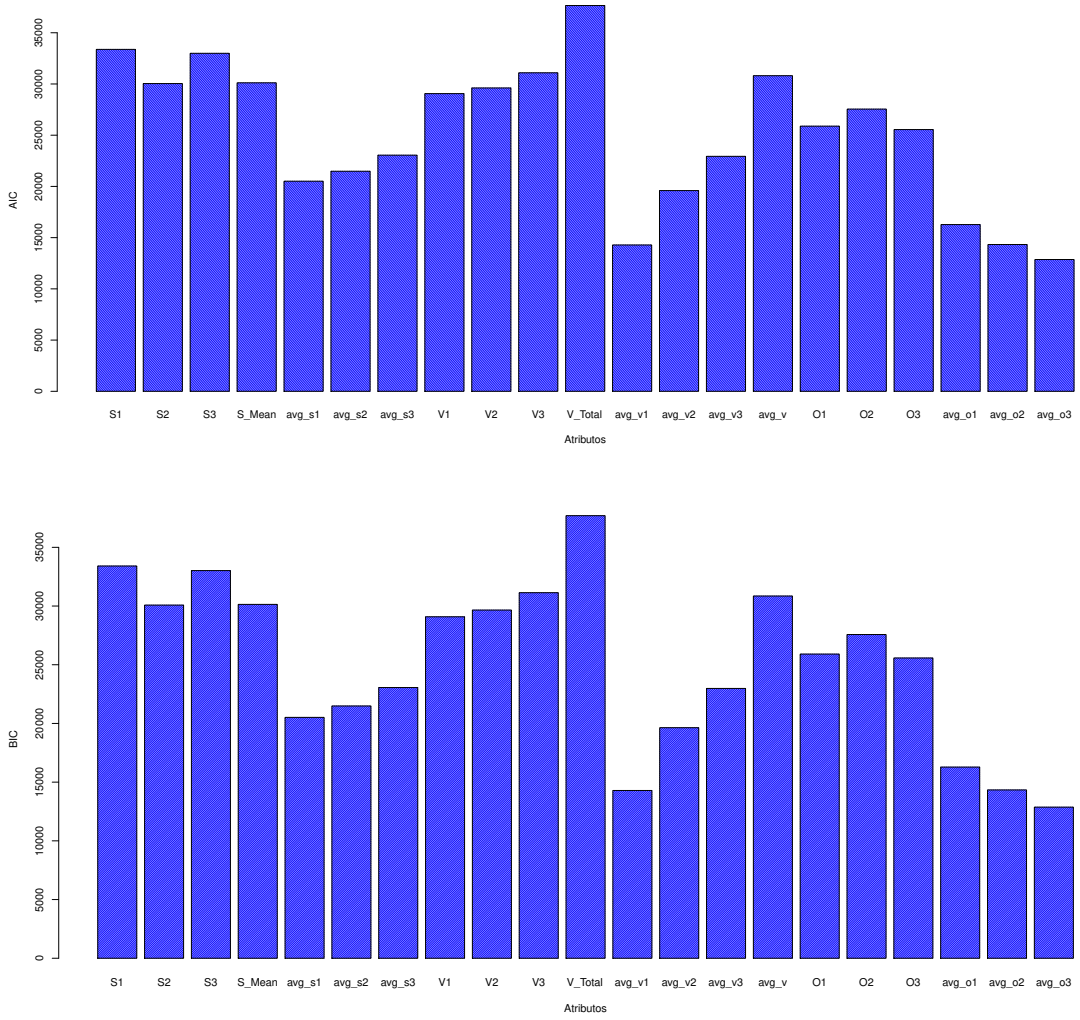

Figura 3.4: Diagrama de barras de los valores AIC y BIC del método Arima de todos los atributos con datos cada 2 minutos - Punto 533

Y la función devolvería el siguiente resultado además del diagrama de barras:

ArimaAIC(Arima_533)
"El atributo avg_o3 tiene un valor de AIC = 12864.822"
"El atributo avg_v1 tiene un valor de AIC = 14284.198"
"El atributo avg_o2 tiene un valor de AIC = 14330.353"
"El atributo avg_o1 tiene un valor de AIC = 16276.393"
"El atributo avg_v2 tiene un valor de AIC = 19593.931"
"El atributo avg_s1 tiene un valor de AIC = 20509.815"
"El atributo avg_s2 tiene un valor de AIC = 21481.396"
"El atributo avg_v3 tiene un valor de AIC = 22936.088"
"El atributo avg_s3 tiene un valor de AIC = 23051.945"
"El atributo O3 tiene un valor de AIC = 25550.663"
"El atributo O1 tiene un valor de AIC = 25882.495"
"El atributo O2 tiene un valor de AIC = 27548.036"
"El atributo V1 tiene un valor de AIC = 29059.921"
"El atributo V2 tiene un valor de AIC = 29611.91"
"El atributo S2 tiene un valor de AIC = 30041.164"
"El atributo S_Mean tiene un valor de AIC = 30114.529"
"El atributo avg_v tiene un valor de AIC = 30808.311"
"El atributo V3 tiene un valor de AIC = 31092.008"
"El atributo S3 tiene un valor de AIC = 32994.113"
"El atributo S1 tiene un valor de AIC = 33377.193"
"El atributo V_Total tiene un valor de AIC = 37658.308"

Los demás resultados numéricos se adjuntan en la parte del Anejo-I, sí se muestran los diagramas obtenidos con los otros puntos de aforo y con las observaciones cada 30 min, deduciendo una gran mejora en los modelos predictivos ajustados a partir de las bases de datos menos pesadas.

Por ejemplo, entre los valores AIC del atributo "avg_o3" del modelo Arima_533 comparado con el modelo Arima_533_light sus valores correspondientes son 12864.822 y 1713.299 (Fig: 3.5) reduciéndose a un 13 % del primer valor AIC, y esto pasa con todos los demás atributos de los otros puntos, reduciendo los valores entre un 7 y un 15 %.

Después de los diagramas de barras, se muestran (Fig:ejemplosarima) otras predicciones de otros puntos de aforo.

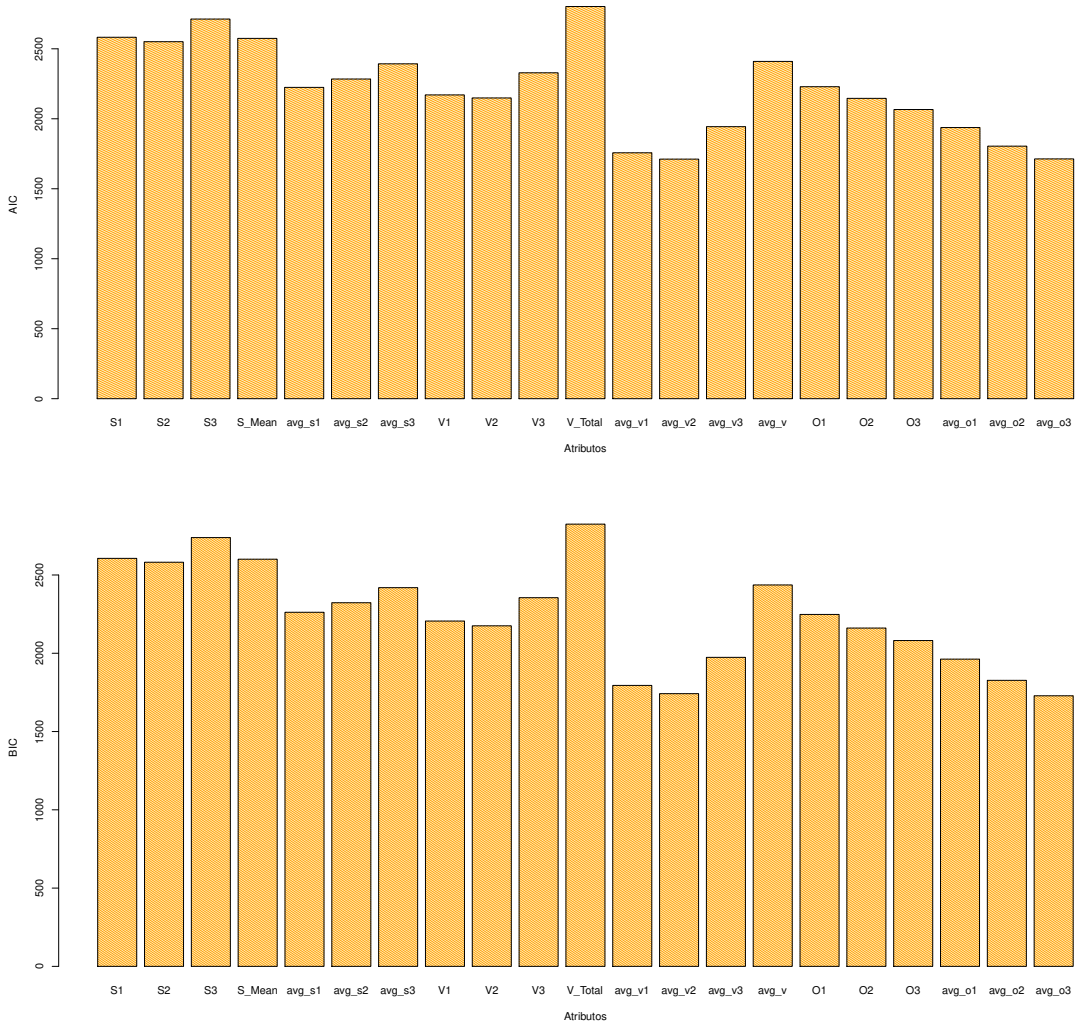

Figura 3.5: Diagrama de barras de los valores AIC y BIC del método Arima de todos los atributos con datos cada 30 minutos - Punto 533

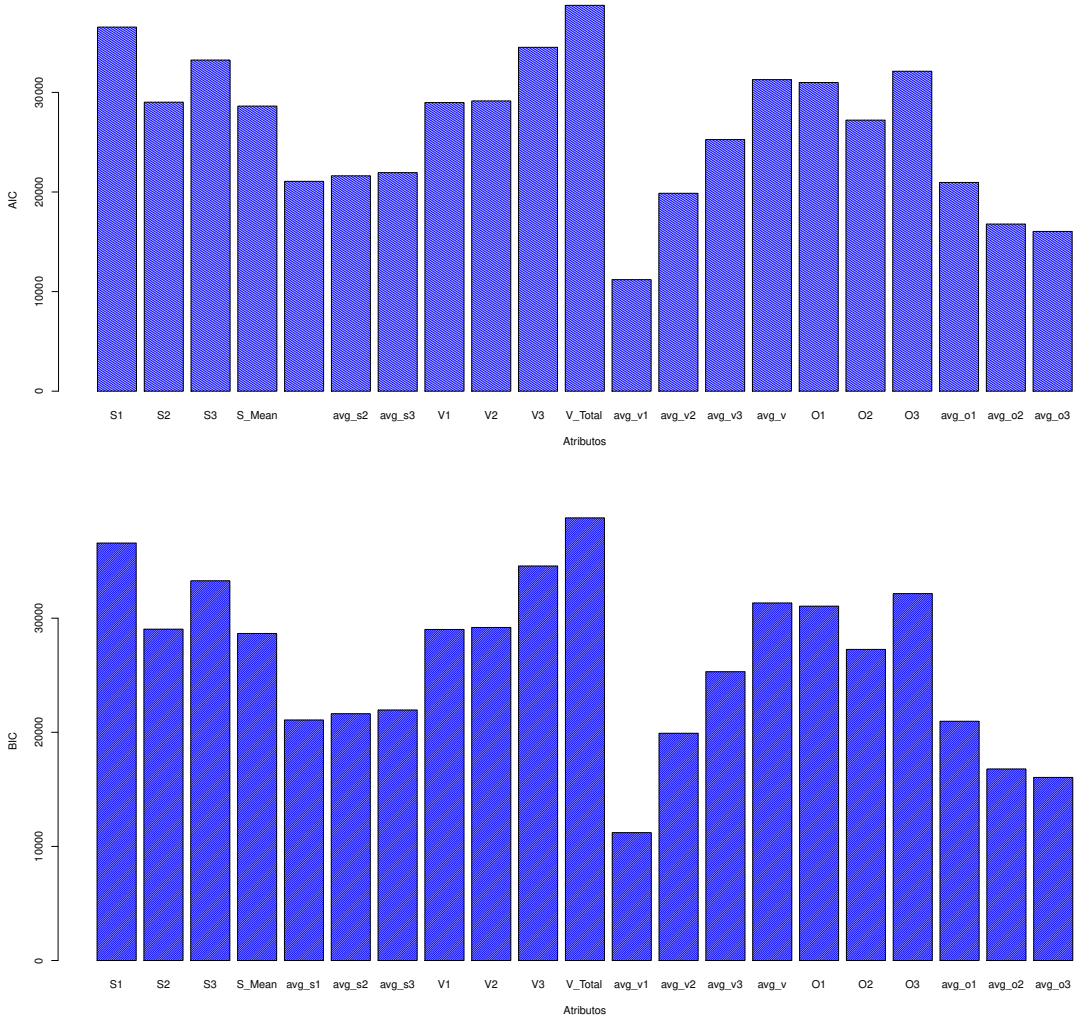

Figura 3.6: Diagrama de barras de los valores AIC y BIC del método Arima de todos los atributos con datos cada 2 minutos - Punto 534

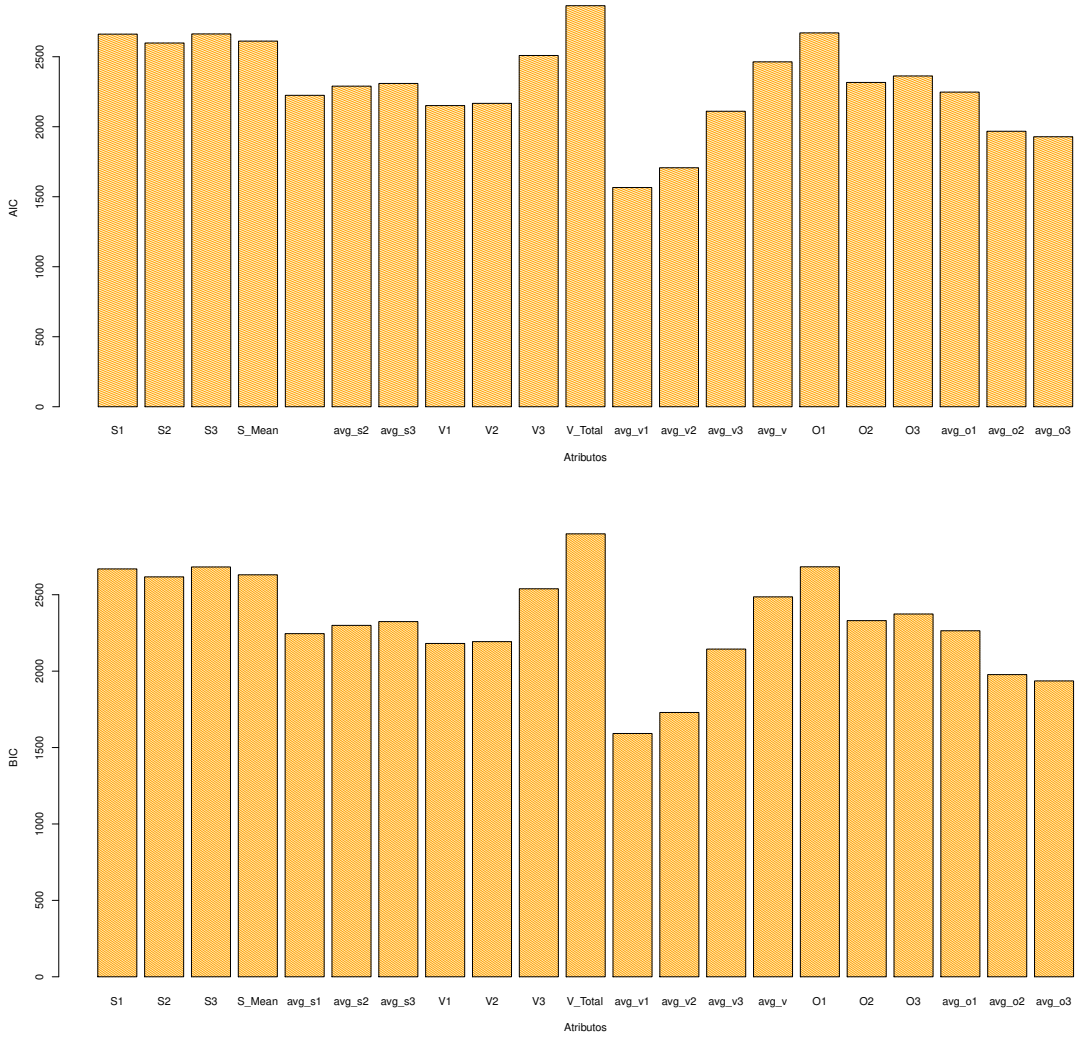

Figura 3.7: Diagrama de barras de los valores AIC y BIC del método Arima de todos los atributos con datos cada 30 minutos - Punto 534

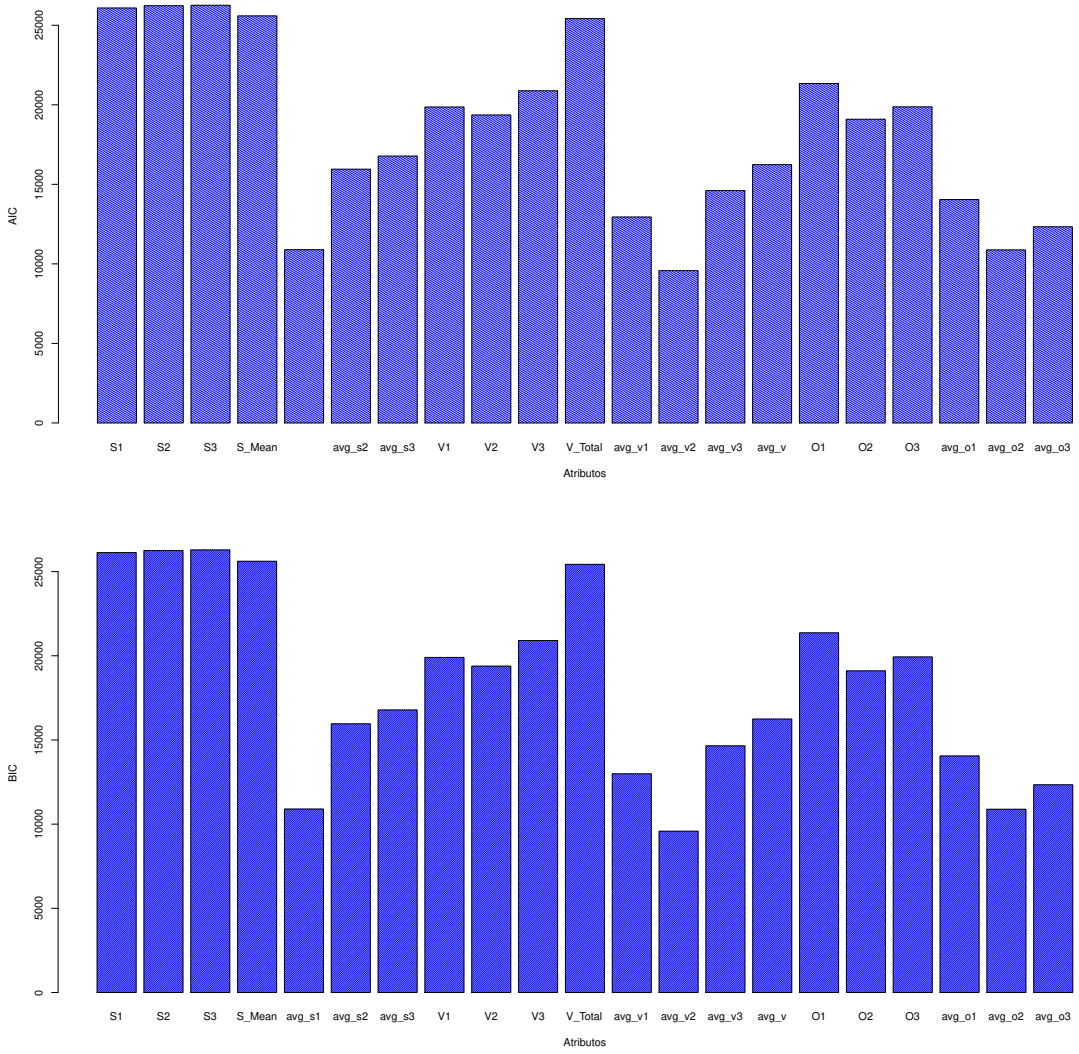

Figura 3.8: Diagrama de barras de los valores AIC y BIC del método Arima de todos los atributos con datos cada 2 minutos - Punto 363

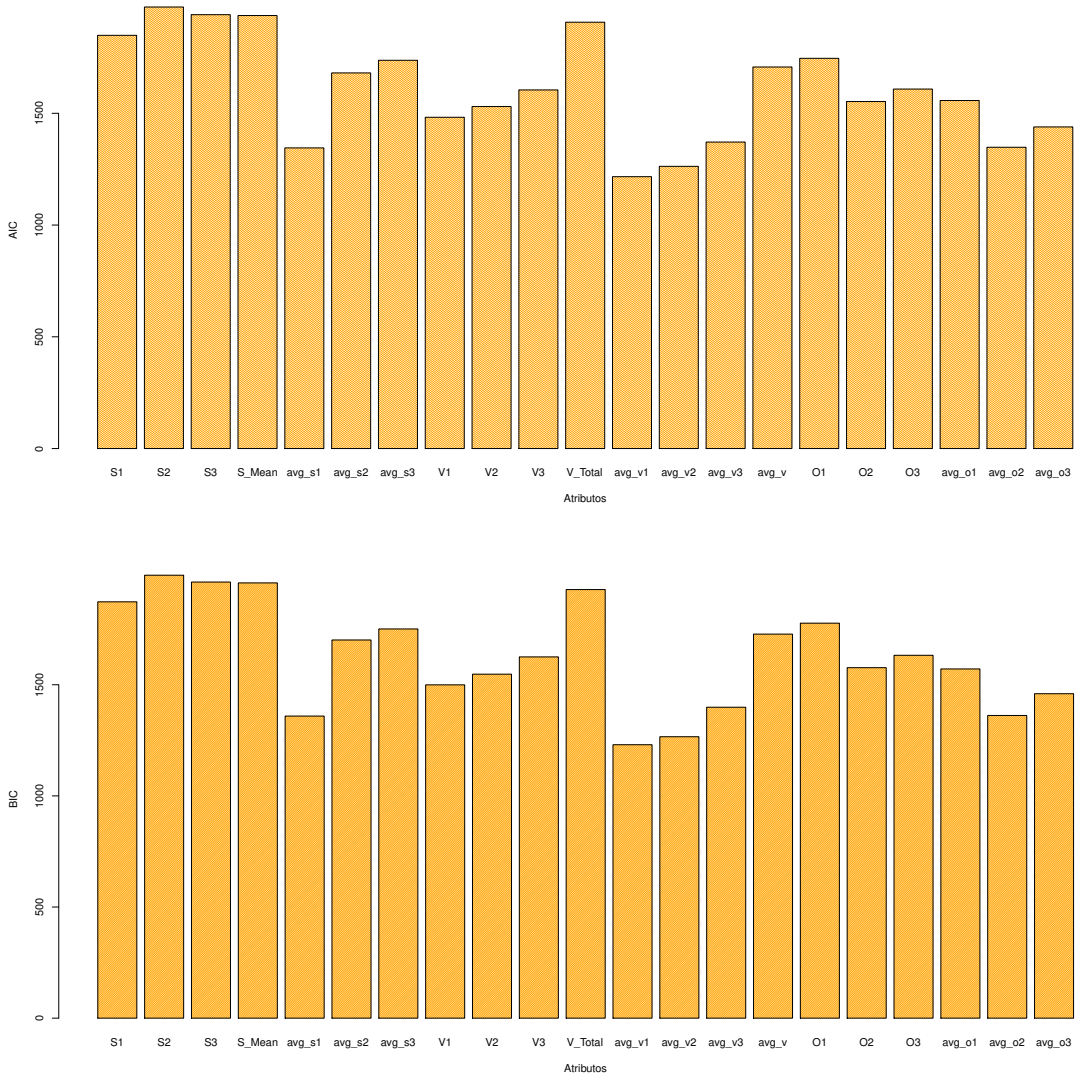

Figura 3.9: Diagrama de barras de los valores AIC y BIC del método Arima de todos los atributos con datos cada 30 minutos - Punto 363

3.2. S-ARIMA

La función Sarima da problemas para conjuntos de datos con frecuencias muy pequeñas en comparación con su estacionalidad, por lo que, en este apartado solo se muestran los resultados obtenidos con los datos tomados cada 30 min.

Para determinar los términos (p, d, q, P, D, Q, S) se ha recurrido al ajuste previamente realizado con el modelo automático de ARIMA, ofreciendo los valores óptimos para el modelo. Con la función **sarima.foi()** se obtiene el ajuste y su gráfica, como muestra la siguiente imagen:

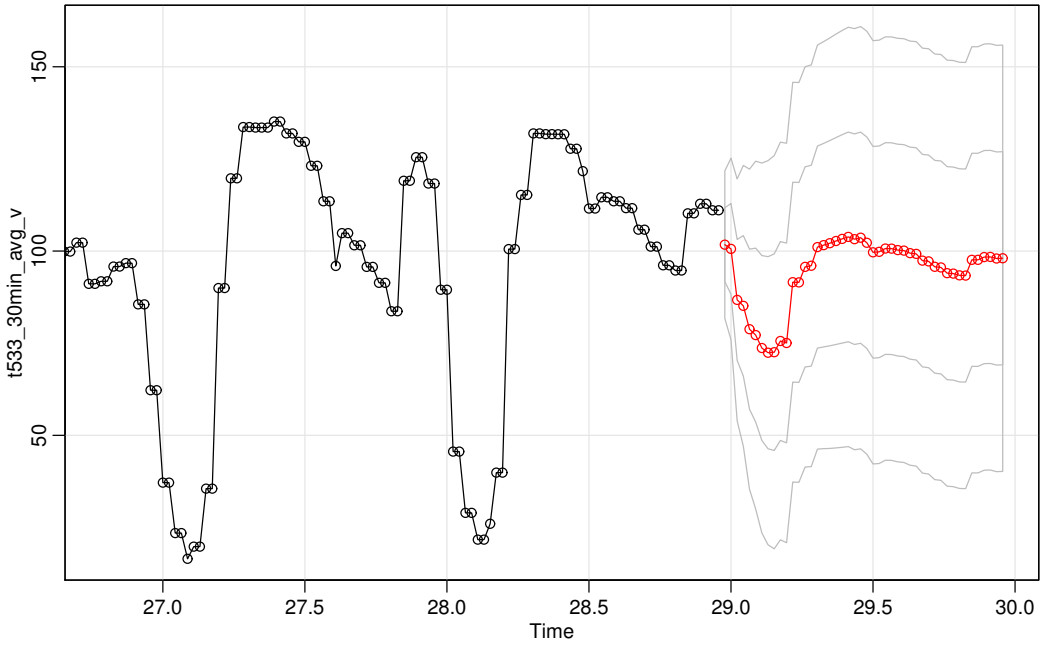

Figura 3.10: Forecast mediante el método S-Arima con la función sarima.for(), del atributo "Volumen Total" con datos cada 30 minutos - Punto 533

Y mediante la función **sarima()** obtenemos los residuales que se añaden a continuación:

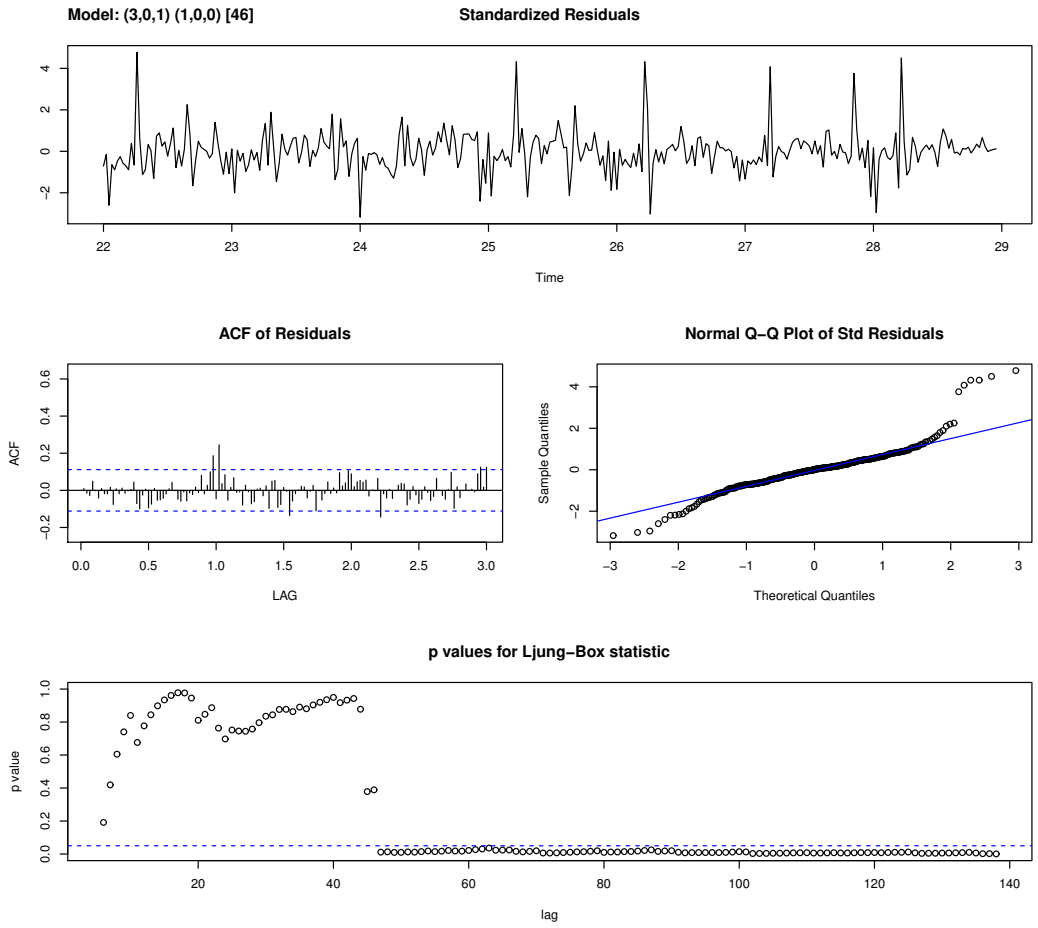

Figura 3.11: Valores e indicadores sobre el ajuste realizado por el modelo sarima propuesto, Residuales, AIC, Q-Q plot y valor -p, de la variable Volumen Total del punto 533.

Los valores del AIC y BIC son 5.643 y 4.713, estos valores del demuestran que existe estacionalidad (mejor ajuste que con ARIMA), no perfecta, ni periódica exacta debido a la variación del tráfico durante los días de la semana, no es lo mismo martes que domingo. Por ello, ahora se procede a modelos dinámicos que permiten ajustar una periodicidad dinámica en el tiempo.

Nota: Los demás valores y gráficos se añaden en el Anejo I, adjuntado al final, pero se muestran los valores del AIC del punto 533 para poder comparar los atributos con este modelo.

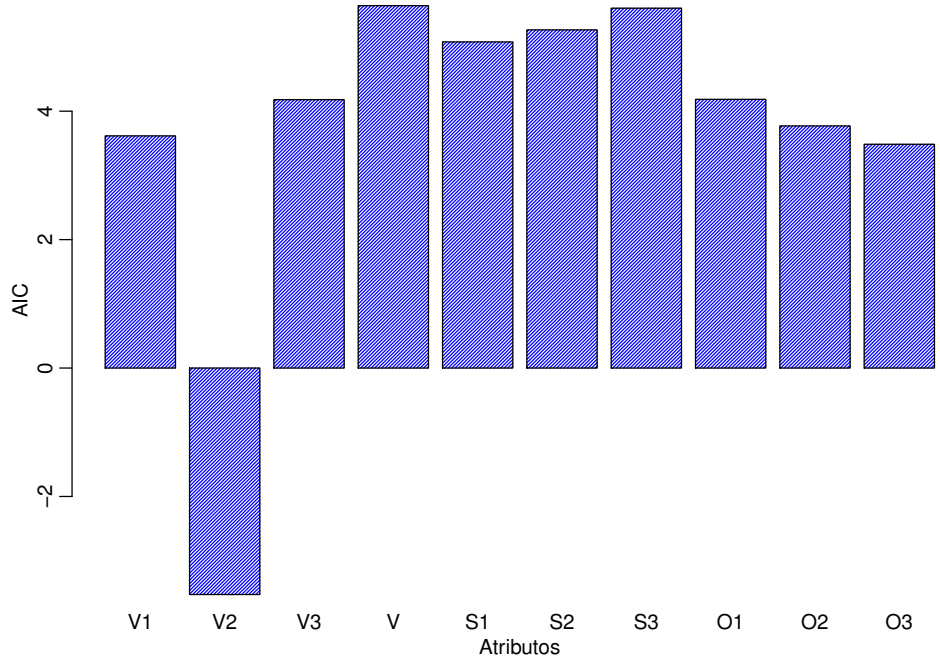

Figura 3.12: Valores AIC de ajuste realizado por el modelo sarima para todos los atributos del punto 533.

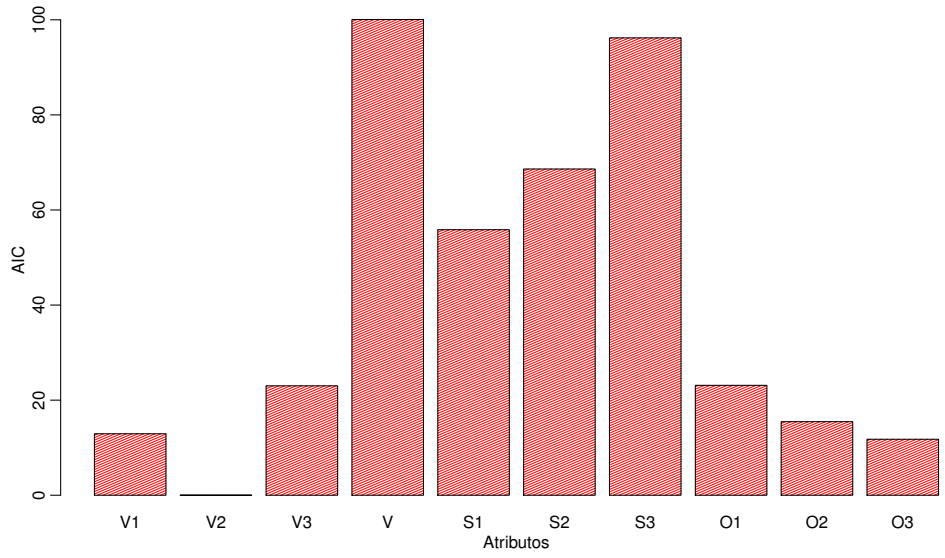

Figura 3.13: Valores ECM de ajuste realizado por el modelo sarima para todos los atributos del punto 533.

3.3. Modelo dinámico Armónico

Con este modelo primero hay que ajustar el coeficiente de Fourier que menor valor del AIC nos de, en este caso se comprobaron los 10 primeros valores de K y este fue el resultado:

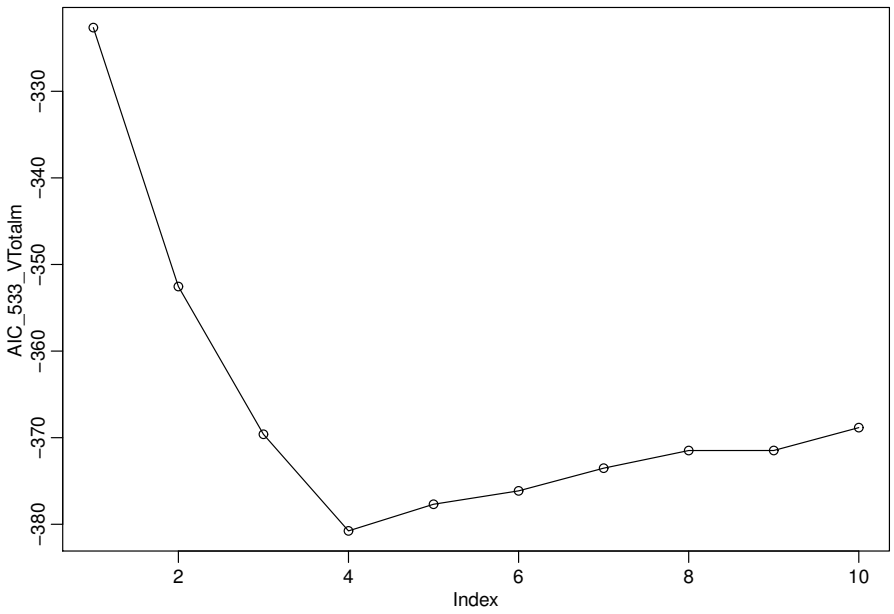

Figura 3.14: Valores del AIC para los primeros valores de K del modelo Armónico, del atributo "Volumen Total" con datos cada 30 minutos - Punto 533

Nota: Como anteriormente ya se ha mencionado, mediante la función implementada **Armonico()** ya se encuentra el mejor coeficiente K cuyo valor AIC del ajuste es menor respecto los demás.

Se ve a simple vista en la Figura 3.14 que el menor valor se corresponde al valor K = 4, por lo que se procede al ajuste con este valor y la visualización de la predicción.

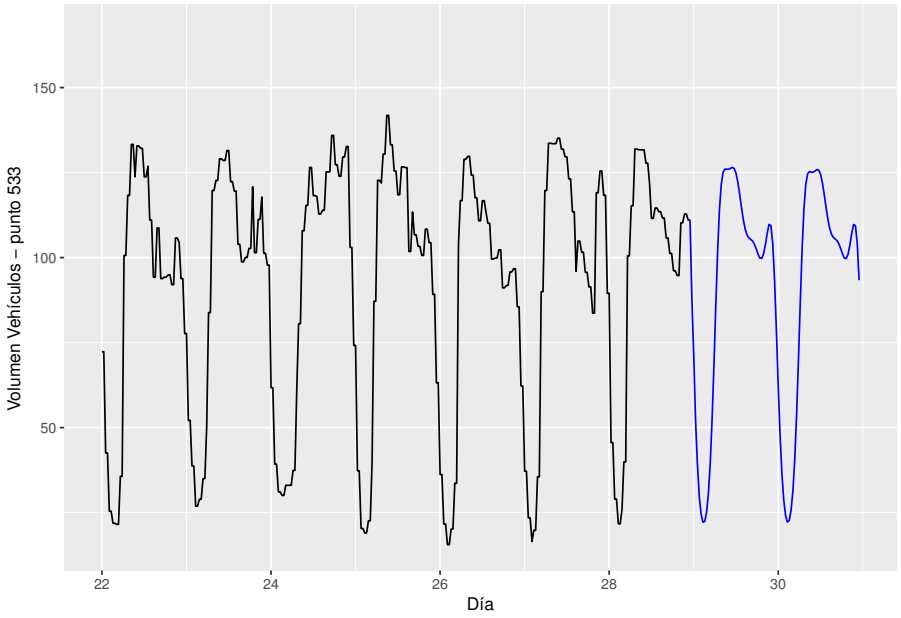

Figura 3.15: Forecast mediante el modelo Armónico (K=4), del atributo "Volumen Total" con datos cada 30 minutos - Punto 533

Se hacen pruebas también para comprobar los ajustes con otros coeficientes de Fourier, en este caso con K= 2 y K = 6.

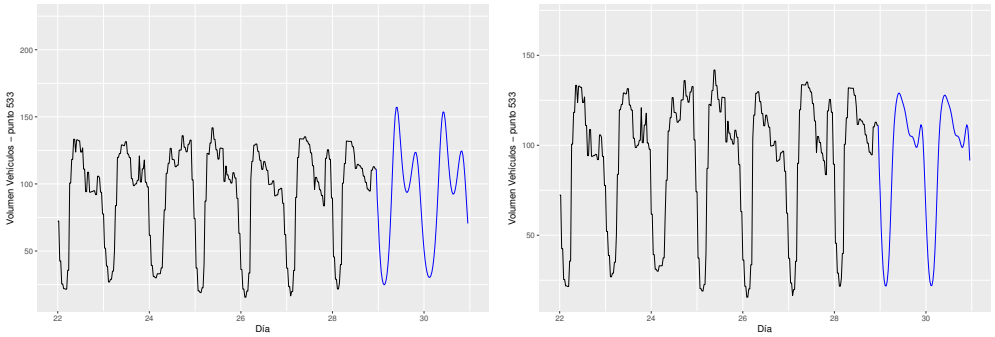

Figura 3.16: Forecast mediante el modelo Armónico K=2 (izq.) y K=6 (der.), del atributo "Volumen Total" con datos cada 30 minutos - Punto 533

También se experimento al cambiar las opciones del ajuste, seleccionando la estacionalidad como "TRUE" y modificando el valor de "lambda", y estos fueron los resultados, para:

- K = 4 :
  AIC = -380.76; BIC = -327.96 ECM = 0.0163

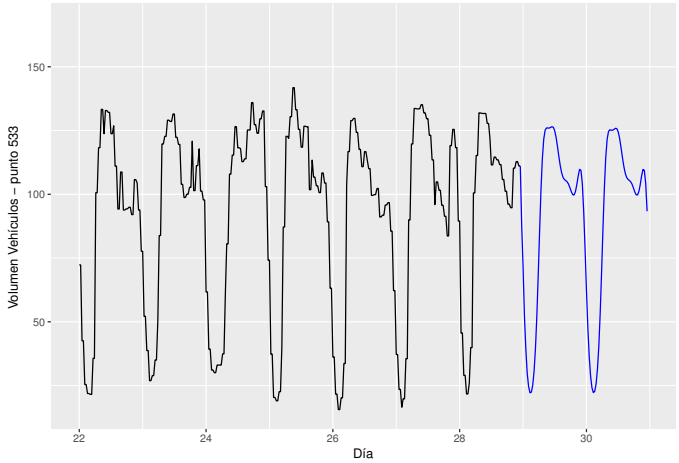

Figura 3.17: Forecast mediante el modelo Armónico (K=4), del atributo "Volumen Total" con datos cada 30 minutos - Punto 533

- K = 4, Seasonal = TRUE :

  AIC = -378.07; BIC = -317.73 ECM = 0.0162

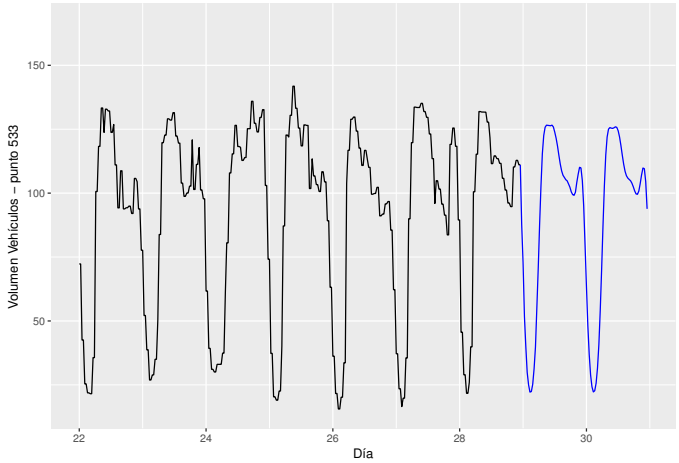

Figura 3.18: Forecast mediante el modelo Armónico (K=4, Seasonal = TRUE), del atributo "Volumen Total" con datos cada 30 minutos - Punto 533

- K = 4, Lambda = 1 :

AIC = 2347.45; BIC = 2404.02 ECM = 79.63

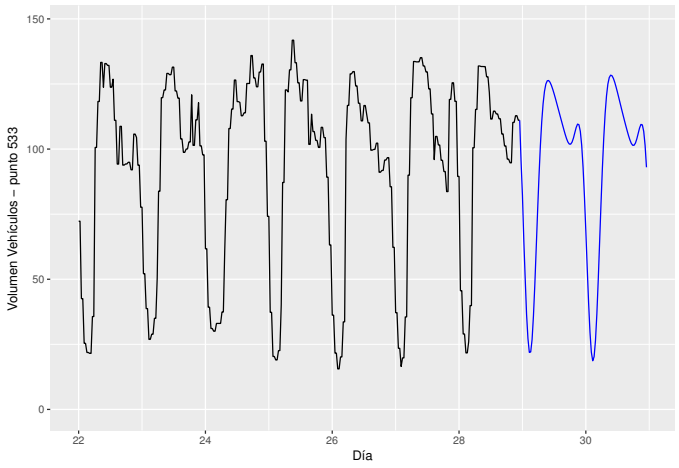

Figura 3.19: Forecast mediante el modelo Armónico (K=4, Lambda = 1), del atributo "Volumen Total" con datos cada 30 minutos - Punto 533

- K = 4, Seasonal = TRUE, Lambda = 1 :

AIC = 2348.47; BIC = 2408.81 ECM = 79.26

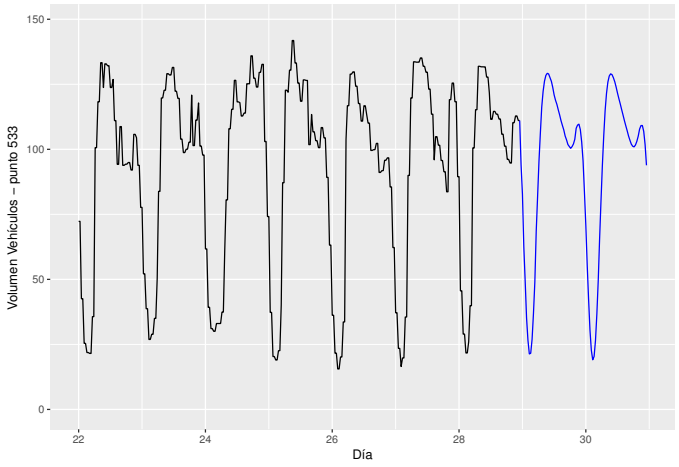

Figura 3.20: Forecast mediante el modelo Armónico (K=4, Seasonal = TRUE, Lambda = 1), del atributo "Volumen Total" con datos cada 30 minutos - Punto 533

3.4. TBATS

A continuación se muestran los resultados obtenidos con el modelo **tbats()** para el caso tratado "Volumen total" del punto 533:

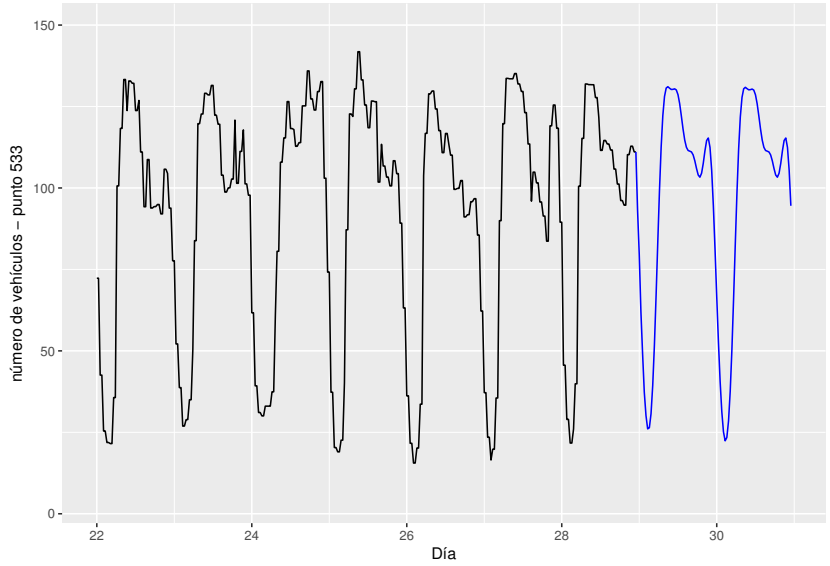

Figura 3.21: Forecast mediante el modelo Tbats, del atributo "Volumen Total" con datos cada 30 minutos - Punto 533

También se añaden los valores de AIC obtenidos del ajuste para los distintos atributos del punto 533, con observaciones cada 30 min:

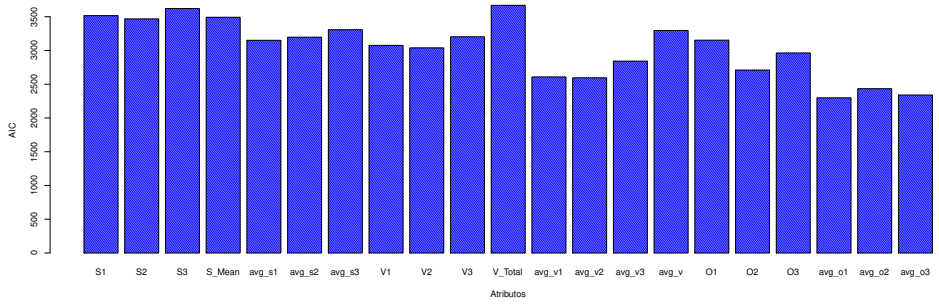

Figura 3.22: Valores AIC con modelo Tbats, de los atributos del punto 533 con datos cada 30 minutos

Los resultados restantes se pueden ver en el Anejo adjunto. Finalmente extraen las conclusiones expuestas en el último capítulo.

Capítulo 4

Conclusión

En este último apartado se zanja el trabajo, resumiendo y sintetizando los resultados obtenidos, se comprueba que los resultados están en línea con los obtenidos por otras publicaciones y exponer las líneas futuras.

En cuento a las muestras para los modelos, los resultados han mostrado mejores resultados haciendo los cálculos con las medias en ciertos intervalos de tiempo, esto le da más equilibrio al modelo y por lo tanto menos errores, pero no sirve para el objetivo de la predicción. En cambio, si de estos valores medios se escoge una observación por cada intervalo de tiempo reduciendo así la muestra, el modelo, aunque pierde información, mejora de una forma considerable la predicción y los ajustes.

De los resultados obtenidos se observa un mejor ajuste de la predicción de las series temporales con los modelos dinámicos armónicos, basados en series de Fourier. Los resultados dados por este tipo de modelo son bastante acertados, excepto en algunos casos puntuales: cuando la muestra no tiene ningún tipo de patrón en su comportamiento o cuando el coeficiente de Fourier utilizado es bajo ($K \leq 2$), en este último caso puede que se ajuste bien pero queda un modelo muy pobre.

En comparación con los otros modelos, el armónico, obtiene un resultado con menor tiempo de cálculo y queda demostrado que al añadir estacionalidad o aumentar el valor de lambda, para estos casos, no mejora el modelo y en el caso de la estacionalidad, aumenta el tiempo del procesamiento.

Entre los atributos que se recogen en el estudio y se analizan, si se ordenan de los que han obtenido un menor error a uno mayor, según la mayoría de modelos probados, se demuestra que la ocupación y el volumen de vehículos son los que muestran un mejor ajuste en comparación con la velocidad. Esto

da a entender que las variables “Ocupación” y “Volumen” tienen un comportamiento menos estocástico que el que se ve las en la variable “Velocidad”. Los errores obtenidos hay que verlos también según el rango de cada atributo ya que no es lo mismo el numero de vehículos por cierto carril(0 - 60), que en la suma de los tres (15 - 180) , el porcentaje de ocupación (0-40 %) o la velocidad (0-100); esto explica algunas diferencias excesivas entre los ajustes de los atributos.

El autor ve la posibilidad de modelar primero el volumen y posteriormente, aprovechando la correlación entre las 3 variables, utilizar un modelo dinámico para ajustar la previsión de los otros atributos, aunque este estudio no entra dentro del ámbito del trabajo. También exponer como líneas futuras, ya que no se ha podido añadir dentro de la carga del trabajo exigido, la comparación de los modelos estadísticos mencionados con técnicas del aprendizaje computado (*Machine Learning*) como el *Support Vector Regression / Machine* (SVR/M).

Bibliografía

[1] *PLAN ESTATAL DE CIENCIA Y TECNOLOGÍA Y DE INNOVACIÓN 2017-2020*.

[2] *ESTRATEGIA DE ESPECIALIZACIÓN INTELIGENTE DE CANARIAS 2014-2020*, 2013.

[3] Xavier Alegret. ¿cuántos coches circulan por el mundo? Technical report, Economiadigital, March 2016. Digital Newspaper, A3Media.

[4] M Beiraghi and AM Ranjbar. Discrete fourier transform based approach to forecast monthly peak load. In *Power and Energy Engineering Conference (APPEEC), 2011 Asia-Pacific*, pages 1–5. IEEE, 2011.

[5] Daniel Billings and Jiann-Shiou Yang. Application of the arima models to urban roadway travel time prediction-a case study. In *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, volume 3, pages 2529–2534. IEEE, 2006.

[6] John F Clarke and Robert B Faoro. An evaluation of co2, measurements as an indicator of air pollution. *Journal of the Air Pollution Control Association*, 16(4):212–218, 1966.

[7] Comisión Europea. *LIBRO BLANCO Hoja de ruta hacia un espacio único europeo de transporte: por una política de transportes competitiva y sostenible*, 2011.

[8] Lucas Dias Condeixa, Leonardo dos Santos Lourenço Bastos, Fernando Luiz Cyrino Oliveira, and Simone DJ Barbosa. Wind speed time series analysis using tbats decomposition and moving blocks bootstrap. *International Journal of Energy and Statistics*, 5(02):1750010, 2017.

[9] Consejo de la Unión Europea. *ESTRATEGIAS NACIONALES Y REGIONALES PARA LA ESPECIALIZACIÓN INTELIGENTE (RIS3)*, 2013.

[10] Ministerio de Economía y Competitividad en colaboración con el Consejo de Política Científica. *ESTRATEGIA ESPAÑOLA DE CIENCIA Y TECNOLOGÍA Y DE INNOVACIÓN 2013-2020*. Gobierno Español.

[11] Gregory C. Reinsel & Greta M. Ljung George E. P. Box, Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.

[12] http://coches.idae.es/portal/Consejos.aspx. *IDAE - Consejos para ahorrar carburante*. Instituto para la Diversificación y el Ahorro de la Energía.

[13] http://www.oica.net/. *WORLD MOTOR VEHICLE PRODUCTION BY COUNTRY AND TYPE - WORLD VEHICLES IN USE - ALL VEHICLES*. OICA-Organisation Internationale des Constructeurs d'Automobiles, 2016.

[14] Alysha M. De Livera, Rob J. Hyndman, and Ralph D. Snyder. Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106(496):1513–1527, 2011.

[15] A. E. Permanasari, I. Hidayah, and I. A. Bustoni. Sarima (seasonal arima) implementation on time series to forecast the number of malaria incidence. In *Proc. Int. Conf. Information Technology and Electrical Engineering (ICITEE)*, pages 203–207, October 2013.

[16] Burak Sansal. http://www.greatistanbul.com/numbers.html.

[17] Stef van Buuren and Karin Groothuis-Oudshoorn. Mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software, Articles*, 45(3):1–67, 2011.

[18] Mascha Van Der Voort, Mark Dougherty, and Susan Watson. Combining kohonen maps with arima time series models to forecast traffic flow. *Transportation Research Part C: Emerging Technologies*, 4(5):307–318, 1996.