

Trabajo Obligatorio - Análisis de una Serie Temporal de Ventas

Estudiantes: Javier Ramas (107758) - Antonia Rodriguez (260101) - Sebastián Zanotta (252045)

Objetivo:

Analizar el dataset de una empresa ficticia que comercia productos en online, algo similar a lo que hace Amazon.

Utilizaremos la biblioteca `astsa` como hasta ahora, y en algunos casos la biblioteca `forecast` que tiene algunas funciones que simplifican este proceso.

```
In [1]: #install.packages("forecast")
#install.packages("astsa")
#install.packages("ggplot2")
#install.packages("tidyverse")
#install.packages("dplyr")
library(forecast)
library(astsa)
library(ggplot2)
library(tidyverse)
lag <- stats::lag
#library(magrittr)
#library(openxlsx)
library(dplyr)
```

Registered S3 method overwritten by 'quantmod':

```
method          from
as.zoo.data.frame zoo
```

Registered S3 methods overwritten by 'forecast':

```
method          from
fitted.fracdiff fracdiff
residuals.fracdiff fracdiff
```

Attaching package: 'astsa'

The following object is masked from 'package:forecast':

gas

— Attaching packages —

verse 1.2.1 —

```
✓ tibble 3.0.3    ✓ purrr 0.3.4
✓ tidyr  1.1.0    ✓ dplyr  1.0.0
✓ readr  1.3.1    ✓ stringr 1.4.0
✓ tibble 3.0.3    ✓ forcats 0.5.0
```

— Conflicts —

conflicts() —

```
✗ dplyr::filter() masks stats::filter()
✗ dplyr::lag()     masks stats::lag()
```

tidy

tidyverse_

```
In [2]: ventas <- read.csv("Global Superstore.csv", sep = ",")
ventas
```

A data.frame: 51290 × 24

Row.ID	Order.ID	Order.Date	Ship.Date	Ship.Mode	Customer.ID	Customer.Name
<int>	<fct>	<fct>	<fct>	<fct>	<fct>	<fct>
32298	CA-2012-124891	31-07-2012	31-07-2012	Same Day	RH-19495	Rick Hansen
26341	IN-2013-77878	05-02-2013	07-02-2013	Second Class	JR-16210	Justin Ritter
25330	IN-2013-71249	17-10-2013	18-10-2013	First Class	CR-12730	Craig Reiter
13524	ES-2013-1579342	28-01-2013	30-01-2013	First Class	KM-16375	Katherine Murray
47221	SG-2013-4320	05-11-2013	06-11-2013	Same Day	RH-9495	Rick Hansen
22732	IN-2013-42360	28-06-2013	01-07-2013	Second Class	JM-15655	Jim Mitchum
30570	IN-2011-81826	07-11-2011	09-11-2011	First Class	TS-21340	Toby Swindell
31192	IN-2012-86369	14-04-2012	18-04-2012	Standard Class	MB-18085	Mick Brown
40155	CA-2014-135909	14-10-2014	21-10-2014	Standard Class	JW-15220	Jane Waco
40936	CA-2012-116638	28-01-2012	31-01-2012	Second Class	JH-15985	Joseph Holt
34577	CA-2011-102988	05-04-2011	09-04-2011	Second Class	GM-14695	Greg Maxwell
28879	ID-2012-28402	19-04-2012	22-04-2012	First Class	AJ-10780	Anthony Jacobs
45794	SA-2011-1830	27-12-2011	29-12-2011	Second Class	MM-7260	Magdelene Morse

Row.ID	Order.ID	Order.Date	Ship.Date	Ship.Mode	Customer.ID	Customer.Name
<int>	<fct>	<fct>	<fct>	<fct>	<fct>	<fct>
4132	MX-2012-130015	13-11-2012	13-11-2012	Same Day	VF-21715	Vicky Freymann
27704	IN-2013-73951	06-06-2013	08-06-2013	Second Class	PF-19120	Peter Fuller
13779	ES-2014-5099955	31-07-2014	03-08-2014	Second Class	BP-11185	Ben Peterman
36178	CA-2014-143567	03-11-2014	06-11-2014	Second Class	TB-21175	Thomas Boland
12069	ES-2014-1651774	08-09-2014	14-09-2014	Standard Class	PJ-18835	Patrick Jones
22096	IN-2014-11763	31-01-2014	01-02-2014	First Class	JS-15685	Jim Sink
49463	TZ-2014-8190	05-12-2014	07-12-2014	Second Class	RH-9555	Ritsa Hightower
46630	PL-2012-7820	08-08-2012	10-08-2012	First Class	AB-600	Ann Blume
31784	CA-2011-154627	29-10-2011	31-10-2011	First Class	SA-20830	Sue Ann Reed
21586	IN-2011-44803	02-05-2011	03-05-2011	First Class	JK-15325	Jason Klamczynski
13528	ES-2013-2860574	27-02-2013	01-03-2013	Second Class	LB-16795	Laurel Beltran
1570	US-2014-133193	31-07-2014	01-08-2014	First Class	NP-18325	Naresj Patel
3484	MX-2014-165309	05-09-2014	08-09-2014	First Class	VD-21670	Valerie Dominguez
30191	IN-2011-10286	17-12-2011	20-12-2011	First Class	PB-19210	Phillip Breyer
11645	ES-2011-4699764	14-03-2011	17-03-2011	Second Class	EB-14110	Eugene Barchas
37311	CA-2013-159016	11-03-2013	12-03-2013	First Class	KF-16285	Karen Ferguson
22999	IN-2012-44810	25-02-2012	25-02-2012	Same Day	BP-11230	Benjamin Patterson
:	:	:	:	:	:	:

Row.ID	Order.ID	Order.Date	Ship.Date	Ship.Mode	Customer.ID	Customer.Name
<int>	<fct>	<fct>	<fct>	<fct>	<fct>	<fct>
40079	CA-2013-116911	16-09-2013	21-09-2013	Standard Class	JD-16150	Justin Deggeller
43851	RS-2014-3790	17-12-2014	20-12-2014	Second Class	DP-3000	Darren Powers
51055	CA-2014-6810	23-12-2014	23-12-2014	Same Day	VM-11835	Vivian Mathis
6367	MX-2011-131688	22-12-2011	26-12-2011	Standard Class	BP-11230	Benjamin Patterson
13800	ES-2014-5679681	05-08-2014	11-08-2014	Standard Class	NM-18520	Neoma Murray
15975	ES-2011-4021337	10-05-2011	14-05-2011	Standard Class	MM-18055	Michelle Moray
22589	IN-2014-32000	25-11-2014	30-11-2014	Standard Class	AO-10810	Anthony O'Donnell
27081	ID-2012-36725	10-08-2012	11-08-2012	First Class	EK-13795	Eileen Kiefer
34657	US-2012-142020	04-06-2012	08-06-2012	Second Class	TC-21535	Tracy Collins
36834	US-2013-167472	07-06-2013	08-06-2013	First Class	CK-12595	Clytie Kelty
38414	CA-2011-143168	18-10-2011	23-10-2011	Second Class	IG-15085	Ivan Gibson
47274	SA-2014-10	09-04-2014	13-04-2014	Standard Class	JW-6075	Julia West
5465	MX-2014-101980	17-04-2014	20-04-2014	Second Class	AA-10480	Andrew Allen
5596	MX-2013-134600	30-09-2013	02-10-2013	Second Class	SS-20410	Shahid Shariari
9309	MX-2014-162803	11-09-2014	12-09-2014	First Class	SL-20155	Sara Luxemburg
10410	ES-2011-1406120	03-12-2011	03-12-2011	Same Day	LR-17035	Lisa Ryan

Row.ID	Order.ID	Order.Date	Ship.Date	Ship.Mode	Customer.ID	Customer.Name
<int>	<fct>	<fct>	<fct>	<fct>	<fct>	<fct>
31558	US-2014-155299	09-06-2014	13-06-2014	Standard Class	DI-13600	Dorris liebe
37361	CA-2012-111780	25-12-2012	30-12-2012	Second Class	RA-19285	Ralph Arnett
43818	NI-2011-4700	08-09-2011	10-09-2011	Second Class	PO-8865	Patrick O'Donnell
46231	ZI-2011-4350	21-03-2011	26-03-2011	Standard Class	AS-285	Alejandro Savely
46582	TU-2014-6730	29-11-2014	30-11-2014	First Class	KF-6285	Karen Ferguson
6039	MX-2014-169530	09-06-2014	11-06-2014	First Class	HG-15025	Hunter Glantz
9922	MX-2012-100258	28-12-2012	31-12-2012	First Class	KM-16375	Katherine Murray
24105	IN-2014-72327	30-05-2014	30-05-2014	Same Day	KH-16330	Katharine Harms
24175	IN-2014-57662	05-08-2014	10-08-2014	Standard Class	DB-13270	Deborah Brumfield

El dataset se compone de 24 columnas y 51290 filas que representan un pedido cada una, se presentan los datos de ventas comenzando en el año 2011 y finalizando en el 2015. A continuación se presentan los nombres de las columnas que componen las variables de cada pedido. De estas utilizaremos para nuestro análisis con la fecha de compra "Order.Date" y el valor de la venta "Sales".

```
In [3]: colnames(ventas)
```

```
'Row.ID' · 'Order.ID' · 'Order.Date' · 'Ship.Date' · 'Ship.Mode' · 'Customer.ID' ·
'Customer.Name' · 'Segment' · 'City' · 'State' · 'Country' · 'Postal.Code' ·
'Market' · 'Region' · 'Product.ID' · 'Category' · 'Sub.Category' · 'Product.Name' ·
'Sales' · 'Quantity' · 'Discount' · 'Profit' · 'Shipping.Cost' · 'Order.Priority'
```

A continuación identificaremos los valores faltantes y a que variable pertenecen para luego eliminarlos y que no afecten la integridad de los datos ni el análisis posterior.

```
In [4]: sum(is.na(ventas))
```

41296

```
In [5]: ColumnasconNA<-" "
for (i in c(1:ncol(ventas)))
{
  len<-length(grep("TRUE", is.na(ventas[,i])))
  if(len > 0){
    ColumnasconNA<-paste(colnames(ventas[i]), ":", len, ColumnasconNA)
  }
}
ColumnasconNA
```

'Postal.Code : 41296 '

Se observa que todos los valores ausentes se encuentra en la variable (columna) de código postal. Dado que no es una variable que utilizaremos en el trabajo decidimos eliminarla del dataset.

Para trabajar como timeseries vamos a transformar las fechas que originalmente se encontraban como "factor" convirtiendolas a "date", esto nos permitirá poder separar las fechas por año, mes y semana.

```
In [6]: ventas$Order.Date<-as.Date(ventas$Order.Date, "%d-%m-%Y")
```

```
In [7]: ventas$año <- as.numeric(format(ventas$Order.Date, '%Y'))
ventas$mes <- as.numeric(format(ventas$Order.Date, '%m'))
ventas$semana <- as.numeric(format(ventas$Order.Date, '%W'))
```



```
In [8]: ventas
```

A data.frame: 51290 × 27

Row.ID	Order.ID	Order.Date	Ship.Date	Ship.Mode	Customer.ID	Customer.Name
<int>	<fct>	<date>	<fct>	<fct>	<fct>	<fct>
32298	CA-2012-124891	2012-07-31	31-07-2012	Same Day	RH-19495	Rick Hansen
26341	IN-2013-77878	2013-02-05	07-02-2013	Second Class	JR-16210	Justin Ritter
25330	IN-2013-71249	2013-10-17	18-10-2013	First Class	CR-12730	Craig Reiter
13524	ES-2013-1579342	2013-01-28	30-01-2013	First Class	KM-16375	Katherine Murray
47221	SG-2013-4320	2013-11-05	06-11-2013	Same Day	RH-9495	Rick Hansen
22732	IN-2013-42360	2013-06-28	01-07-2013	Second Class	JM-15655	Jim Mitchum
30570	IN-2011-81826	2011-11-07	09-11-2011	First Class	TS-21340	Toby Swindell
31192	IN-2012-86369	2012-04-14	18-04-2012	Standard Class	MB-18085	Mick Brown
40155	CA-2014-135909	2014-10-14	21-10-2014	Standard Class	JW-15220	Jane Waco
40936	CA-2012-116638	2012-01-28	31-01-2012	Second Class	JH-15985	Joseph Holt
34577	CA-2011-102988	2011-04-05	09-04-2011	Second Class	GM-14695	Greg Maxwell
28879	ID-2012-28402	2012-04-19	22-04-2012	First Class	AJ-10780	Anthony Jacobs

Row.ID	Order.ID	Order.Date	Ship.Date	Ship.Mode	Customer.ID	Customer.Name
<int>	<fct>	<date>	<fct>	<fct>	<fct>	<fct>
45794	SA-2011-1830	2011-12-27	29-12-2011	Second Class	MM-7260	Magdelene Morse
4132	MX-2012-130015	2012-11-13	13-11-2012	Same Day	VF-21715	Vicky Freymann
27704	IN-2013-73951	2013-06-06	08-06-2013	Second Class	PF-19120	Peter Fuller
13779	ES-2014-5099955	2014-07-31	03-08-2014	Second Class	BP-11185	Ben Peterman
36178	CA-2014-143567	2014-11-03	06-11-2014	Second Class	TB-21175	Thomas Boland
12069	ES-2014-1651774	2014-09-08	14-09-2014	Standard Class	PJ-18835	Patrick Jones
22096	IN-2014-11763	2014-01-31	01-02-2014	First Class	JS-15685	Jim Sink
49463	TZ-2014-8190	2014-12-05	07-12-2014	Second Class	RH-9555	Ritsa Hightower
46630	PL-2012-7820	2012-08-08	10-08-2012	First Class	AB-600	Ann Blume
31784	CA-2011-154627	2011-10-29	31-10-2011	First Class	SA-20830	Sue Ann Reed
21586	IN-2011-44803	2011-05-02	03-05-2011	First Class	JK-15325	Jason Klamczynski
13528	ES-2013-2860574	2013-02-27	01-03-2013	Second Class	LB-16795	Laurel Beltran
1570	US-2014-133193	2014-07-31	01-08-2014	First Class	NP-18325	Naresj Patel
3484	MX-2014-165309	2014-09-05	08-09-2014	First Class	VD-21670	Valerie Dominguez
30191	IN-2011-10286	2011-12-17	20-12-2011	First Class	PB-19210	Phillip Breyer
11645	ES-2011-4699764	2011-03-14	17-03-2011	Second Class	EB-14110	Eugene Barchas
37311	CA-2013-159016	2013-03-11	12-03-2013	First Class	KF-16285	Karen Ferguson
22999	IN-2012-44810	2012-02-25	25-02-2012	Same Day	BP-11230	Benjamin Patterson

Row.ID	Order.ID	Order.Date	Ship.Date	Ship.Mode	Customer.ID	Customer.Name
<int>	<fct>	<date>	<fct>	<fct>	<fct>	<fct>
:	:	:	:	:	:	:
40079	CA-2013-116911	2013-09-16	21-09-2013	Standard Class	JD-16150	Justin Deggeller
43851	RS-2014-3790	2014-12-17	20-12-2014	Second Class	DP-3000	Darren Powers
51055	CA-2014-6810	2014-12-23	23-12-2014	Same Day	VM-11835	Vivian Mathis
6367	MX-2011-131688	2011-12-22	26-12-2011	Standard Class	BP-11230	Benjamin Patterson
13800	ES-2014-5679681	2014-08-05	11-08-2014	Standard Class	NM-18520	Neoma Murray
15975	ES-2011-4021337	2011-05-10	14-05-2011	Standard Class	MM-18055	Michelle Moray
22589	IN-2014-32000	2014-11-25	30-11-2014	Standard Class	AO-10810	Anthony O'Donnell
27081	ID-2012-36725	2012-08-10	11-08-2012	First Class	EK-13795	Eileen Kiefer
34657	US-2012-142020	2012-06-04	08-06-2012	Second Class	TC-21535	Tracy Collins
36834	US-2013-167472	2013-06-07	08-06-2013	First Class	CK-12595	Clytie Keltly
38414	CA-2011-143168	2011-10-18	23-10-2011	Second Class	IG-15085	Ivan Gibson
47274	SA-2014-10	2014-04-09	13-04-2014	Standard Class	JW-6075	Julia West
5465	MX-2014-101980	2014-04-17	20-04-2014	Second Class	AA-10480	Andrew Allen
5596	MX-2013-134600	2013-09-30	02-10-2013	Second Class	SS-20410	Shahid Shariari
9309	MX-2014-162803	2014-09-11	12-09-2014	First Class	SL-20155	Sara Luxemburg
10410	ES-2011-1406120	2011-12-03	03-12-2011	Same Day	LR-17035	Lisa Ryan

Row.ID	Order.ID	Order.Date	Ship.Date	Ship.Mode	Customer.ID	Customer.Name
<int>	<fct>	<date>	<fct>	<fct>	<fct>	<fct>
31558	US-2014-155299	2014-06-09	13-06-2014	Standard Class	DI-13600	Dorris liebe
37361	CA-2012-111780	2012-12-25	30-12-2012	Second Class	RA-19285	Ralph Arnett
43818	NI-2011-4700	2011-09-08	10-09-2011	Second Class	PO-8865	Patrick O'Donnell
46231	ZI-2011-4350	2011-03-21	26-03-2011	Standard Class	AS-285	Alejandro Savely
46582	TU-2014-6730	2014-11-29	30-11-2014	First Class	KF-6285	Karen Ferguson
6039	MX-2014-169530	2014-06-09	11-06-2014	First Class	HG-15025	Hunter Glantz
9922	MX-2012-100258	2012-12-28	31-12-2012	First Class	KM-16375	Katherine Murray
24105	IN-2014-72327	2014-05-30	30-05-2014	Same Day	KH-16330	Katharine Harms
24175	IN-2014-57662	2014-08-05	10-08-2014	Standard Class	DB-13270	Deborah Brumfield

A continuación agruparemos las ventas por año y semana sumando las mismas para luego generar un dataset que contenga solo esas tres variables. Es decir número total de ventas por semana y año.

```
In [9]: ventas1 <- ventas %>%
  group_by(año, semana) %>%
  summarise(Total_Sales = sum(Sales))

`summarise()` regrouping output by 'año' (override with `groups` argument)
```

```
In [10]: ventas1
```

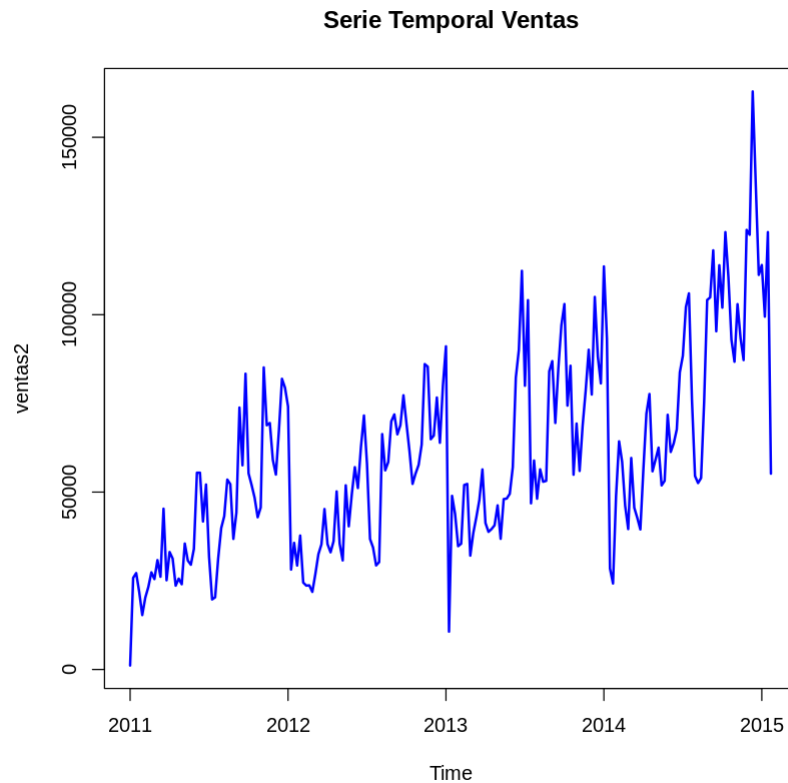
A grouped_df: 212 × 3

año	semana	Total_Sales
<dbl>	<dbl>	<dbl>
2011	0	1122.783
2011	1	25827.283
2011	2	27169.672
2011	3	21799.136
2011	4	15337.029
2011	5	20268.441
2011	6	23255.994
2011	7	27372.644
2011	8	25479.347
2011	9	30824.712
2011	10	26137.756
2011	11	45315.857
2011	12	25148.896
2011	13	33104.578
2011	14	31269.551
2011	15	23633.672
2011	16	25605.471
2011	17	24022.956
2011	18	35495.099
2011	19	30693.585
2011	20	29526.258
2011	21	33984.827
2011	22	55457.941
2011	23	55462.548
2011	24	41714.262
2011	25	52137.213
2011	26	31717.094
2011	27	19743.923
2011	28	20345.803
2011	29	31321.396
⋮	⋮	⋮
2014	23	88435.48
2014	24	102155.83
2014	25	106010.71

año	semana	Total_Sales
<dbl>	<dbl>	<dbl>
2014	26	75860.89
2014	27	54529.96
2014	28	52551.24
2014	29	53963.25
2014	30	75333.98
2014	31	104145.32
2014	32	104876.65
2014	33	118144.36
2014	34	95327.16
2014	35	113942.36
2014	36	101984.15
2014	37	123273.14
2014	38	110573.46
2014	39	92938.16
2014	40	86737.32
2014	41	102970.47
2014	42	93373.00
2014	43	87211.24
2014	44	123896.23
2014	45	122530.74
2014	46	162917.66
2014	47	136854.97
2014	48	111222.48
2014	49	112000.04

Finalmente convertimos ese dataset "ventas1" en una Time Series para poder trabajar utilizando las herramientas aprendidas en clase y la graficamos para ver su forma.


```
In [11]: ventas2 <- ts(ventas1$Total_Sales, start = 2011, frequency = 52)
ts.plot(ventas2, col= 4,main="Serie Temporal Ventas",lwd=2)
options(repr.plot.width=14, repr.plot.height=6)
```



Analisis estacional

La serie presenta una tendencia creciente sostenida a lo largo del tiempo. Anualmente las ventas aumentan hacia fin de año y luego bajan de manera abrupta en las primeras semanas del año siguiente. Esto se debe a un aumento sobre el final del año debido a las fiestas tradicionales.

A continuación ajustaremos un modelo lineal para intentar entender la incidencia de la variable "week" en el comportamiento de la variable "ventas".

```
In [12]: week = ventas1$semana/52
t = time(ventas2) - 2011
fit = lm(ventas2~t+week)
summary(fit)
```

Call:

```
lm(formula = ventas2 ~ t + week)
```

Residuals:

Min	1Q	Median	3Q	Max
-71664	-8848	-1103	9279	61989

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	11321	2712	4.174	4.4e-05 ***
t	12776	990	12.904	< 2e-16 ***
week	44361	3959	11.205	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16460 on 209 degrees of freedom

Multiple R-squared: 0.6481, Adjusted R-squared: 0.6448

F-statistic: 192.5 on 2 and 209 DF, p-value: < 2.2e-16

Definimos: Coeficiente de Determinación

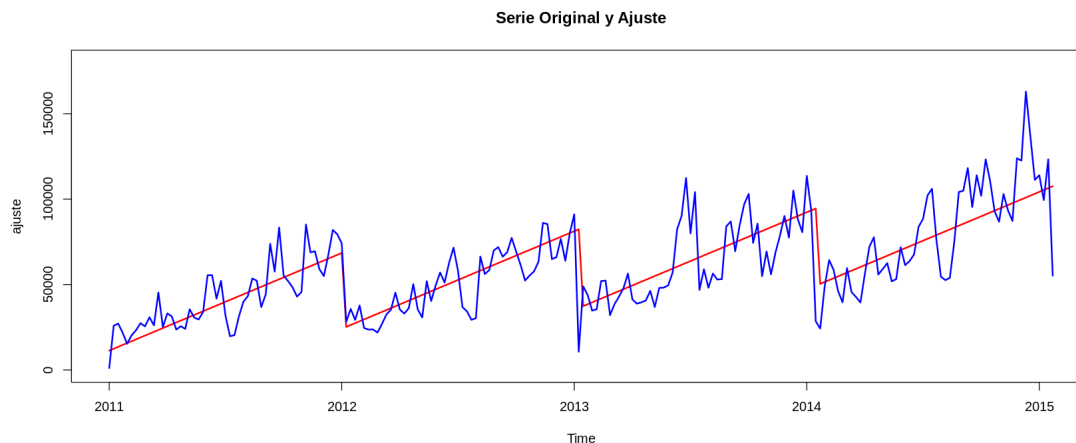
$$R^2 = \frac{SSR}{SST} = \frac{SST - SSE}{SST}$$

Entonces R^2 es una medida de correlación de nuestras variables, o bien cuánto mejora el ajuste en términos relativos respecto a la media.

- SST = Suma total de cuadrados
- SSE = Suma total de cuadrados debido al error (variación sin explicar)
- SSR = Suma total de cuadrados debido a la regresión (variación explicada)

Del análisis anterior surge que: El factor week es significativo y para poder plotear el ajuste, tenemos que convertir los valores a serie temporal. A su vez, para analizar los residuos debemos hacer lo mismo.

```
In [13]: ##genero una serie temporal sincronizada con la serie original y las
predicciones del modelo lineal
ajuste= fitted(fit)
ajuste=ts(ajuste, start = 2011, freq = 52)
ts.plot(ajuste,ylim=c(0,180000), col= 2,lwd=2, main="Serie Original
y Ajuste")
lines(ventas2,col= 4,lwd=2)
```



Definimos: Error Cuadrático Medio de Predicción (RMSE)

Lo que hace aquí es realizar la mejor predicción posible usando el modelo ajustado de los valores de la serie y considerar la varianza del error cometido en la predicción. De nuevo, queremos el menor RMSE posible.

$$RMSE = \sqrt{\frac{SSE}{W}} = \sqrt{\frac{1}{W} \sum_{i=1}^N w_i u_i^2}$$

```
In [14]: #La funcion sigma extrae el RMSE del ajuste
RMSE = sigma(fit)
RMSE
```

16457.2186926841

```
In [15]: AIC = AIC(fit)
BIC = BIC(fit)
AIC
BIC
```

4723.02077143078

4736.44711652947

El modelo aproxima bastante bien la tendencia creciente anual y sostenida en el tiempo. También acompaña las caídas abruptas que se dan al principio de cada año. A continuación haremos el gráfico de los residuos (como serie temporal) para ver si obtenimos ruido blanco.

Análisis de Fourier

Ahora utilizaremos Senos y cosenos para ajustar la tendencia anual. Una serie de Fourier es una serie infinita que converge puntualmente a una función periódica y continua a trozos (o por partes). Las series de Fourier constituyen la herramienta matemática básica del análisis de Fourier empleado para analizar funciones periódicas a través de la descomposición de dicha función en una suma infinita de funciones sinusoidales mucho más simples (como combinación de senos y cosenos con frecuencias enteras).

Con esta herramienta trataremos de aproximar una rampa periódica por cada vez más frecuencias. En el límite de infinitas frecuencias sería exactamente igual, pero en este caso el ajuste anterior (rampa) es mejor. Esto es culpa de la discontinuidad en las compras realizadas en las fiestas de fin de año. Son necesarias muchas funciones de seno y coseno para ajustar esa caída tan abrupta.

```
In [16]: t = time(ventas2) - 2011
sint = sin(2*pi*t)
cost = cos(2*pi*t)
sint1= sin(2*pi*2*t)
cost1= cos(2*pi*2*t)
sint2= sin(2*pi*3*t)
cost2= cos(2*pi*3*t)
fit5 = lm(ventas2~t+sint+cost+sint1+cost1+sint2+cost2)
summary(fit5)
```

Call:

```
lm(formula = ventas2 ~ t + sint + cost + sint1 + cost1 + sint2 +
    cost2)
```

Residuals:

Min	1Q	Median	3Q	Max
-51508	-10711	-937	10397	61503

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	32706.3	2352.2	13.905	< 2e-16	***
t	13269.4	1008.1	13.163	< 2e-16	***
sint	-15882.3	1683.7	-9.433	< 2e-16	***
cost	2284.5	1631.0	1.401	0.163	
sint1	-7379.1	1657.1	-4.453	1.39e-05	***
cost1	2509.8	1634.4	1.536	0.126	
sint2	-974.9	1648.6	-0.591	0.555	
cost2	2136.4	1638.6	1.304	0.194	

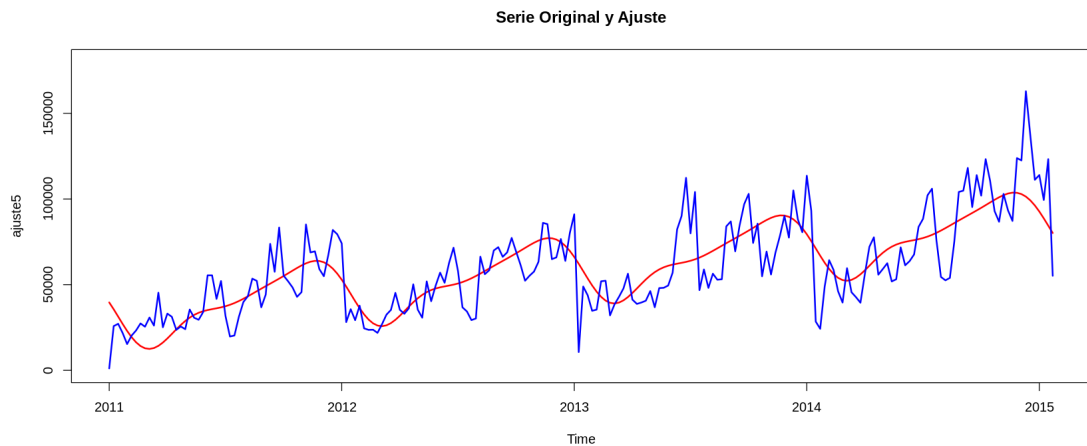
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16900 on 204 degrees of freedom

Multiple R-squared: 0.6378, Adjusted R-squared: 0.6254

F-statistic: 51.33 on 7 and 204 DF, p-value: < 2.2e-16

```
In [17]: ajuste5= fitted(fit5)
ajuste5=ts(ajuste5, start = 2011, freq = 52)
ts.plot(ajuste5,ylim=c(0,180000), col= 2,lwd=2, main="Serie Original
y Ajuste")
lines(ventas2,col= 4,lwd=2)
```



```
In [18]: RMSEF = sd(residuals(fit5))
RMSEF
RMSE
```

16616.9003279461

16457.2186926841

```
In [19]: AICF = AIC(fit5)
BICF = BIC(fit5)
```

AICF

AIC

BICF

BIC

4739.13400470303

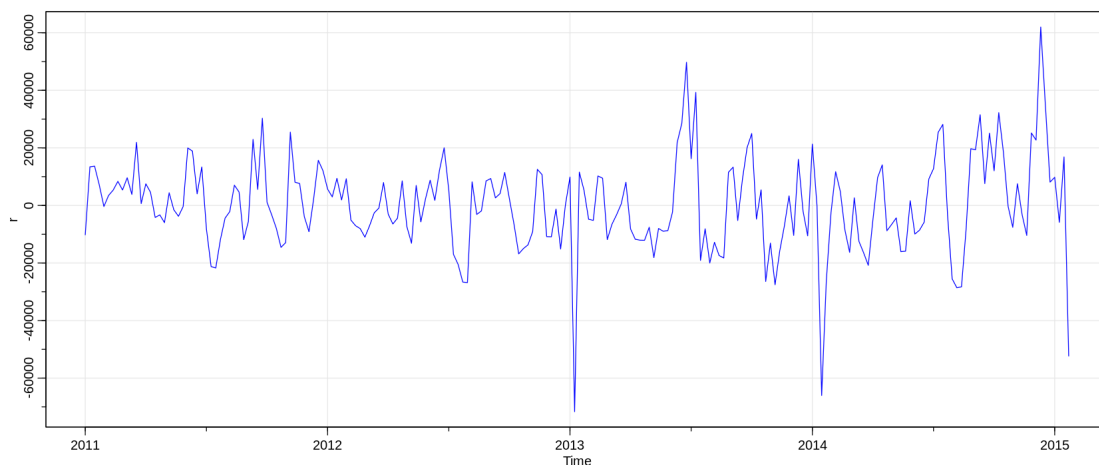
4723.02077143078

4769.34328117508

4736.44711652947

Vemos que el ajuste utilizando el factor week es mejor que el análisis de Fourier por lo tanto vamos a realizar el estudio de los residuos del primer modelo.

```
In [20]: #Generamos una serie temporal con los valores de los residuos y los
         #ploteamos
         r=ts(residuals(fit),start=2011,freq=52)
         tsplot(r, lwd=1, col=4)
```



Observamos que la gráfica no se asemeja a la de ruido blanco.

Periodograma

El *periodograma* es una forma sistemática de recorrer las correlaciones anteriores y calcular cuáles son significativas.

Haremos un primer análisis para ver el periodograma de la serie "ventas2" y luego haremos lo mismo para los residuos obtenidos en el primer modelo, intentando en esta última situación conseguir mayor información para continuar nuestro análisis.

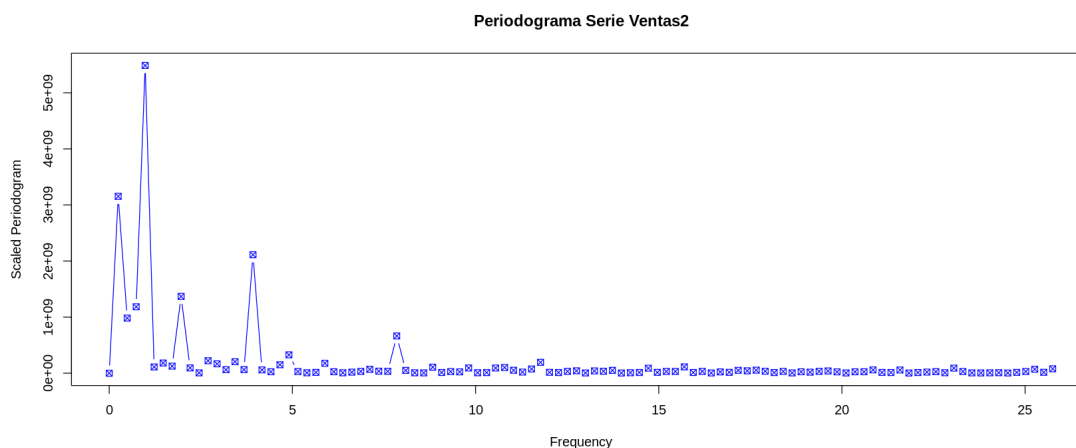
A su vez, debido a la *FFT: Fast Fourier Transform* muchos de los cálculos usados en una regresión pueden ser reusados para calcular otras, llevando a un algoritmo eficiente.

```
In [21]: n=length(t)
         n
```

```
In [22]: options(repr.plot.width=14, repr.plot.height=6)

I = abs(fft(ventas2-mean(ventas2)))^2/n
P = (52/n)*I[1:(n/2)]
f = (0:(n/2-1))/n*52

plot(f, P, type="b", xlab="Frequency", ylab="Scaled Periodogram", ma
in="Periodograma Serie Ventas2", col=4, lwd=1, pch=7)
```

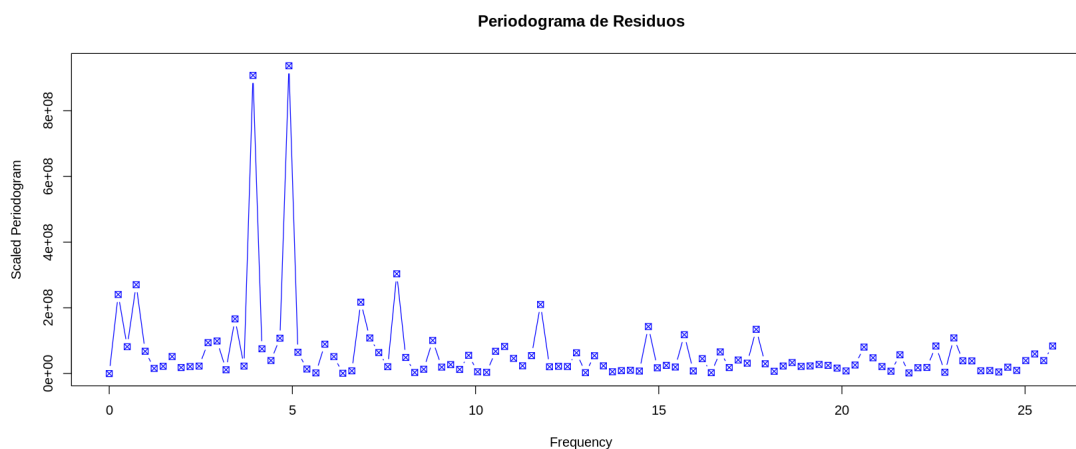


Vemos que las primeras semanas son las más significativas y que se repite su comportamiento año a año, en donde vemos una caída importante de las ventas, una correlación inversa. Ese comportamiento es validado al hacer el estudio del periodograma con los residuos.

```
In [23]: options(repr.plot.width=14, repr.plot.height=6)

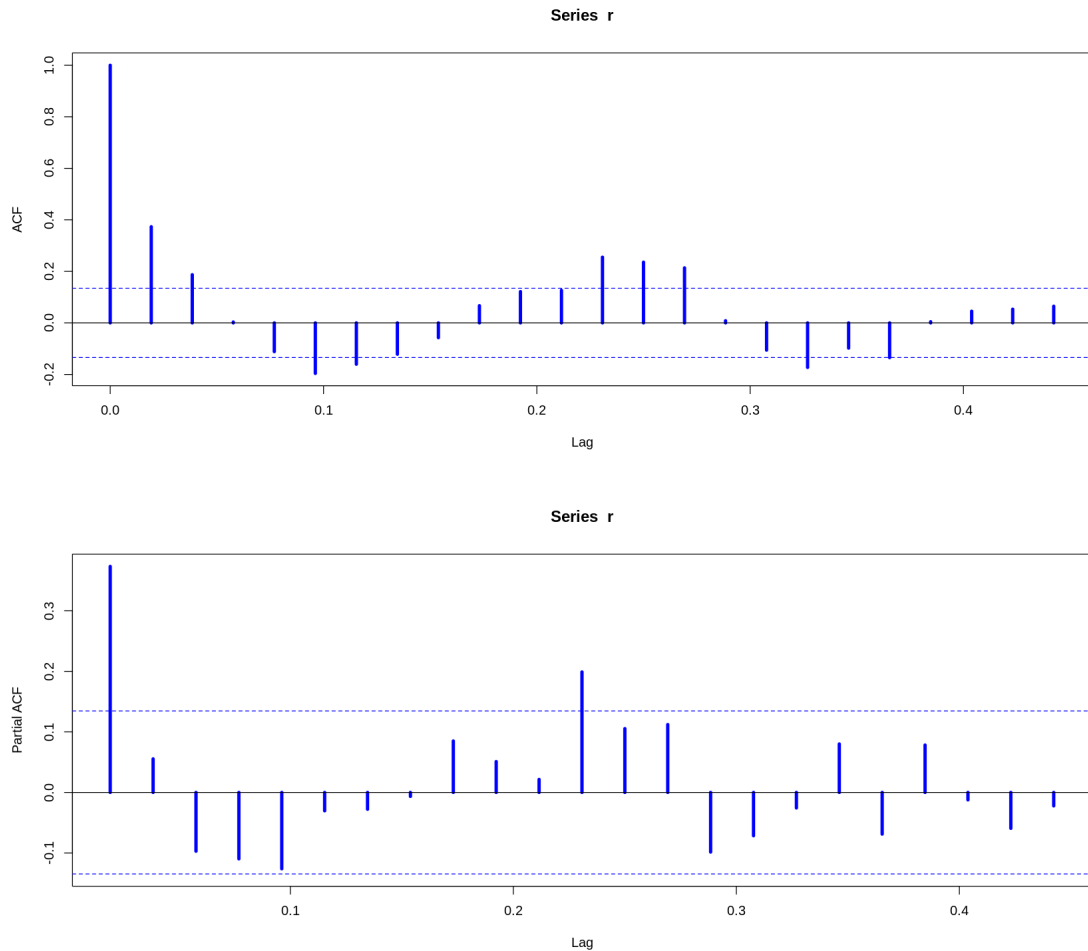
I = abs(fft(r))^2/n
P = (52/n)*I[1:(n/2)]
f = (0:(n/2-1))/n*52

plot(f, P, type="b", xlab="Frequency", ylab="Scaled Periodogram", ma
in="Periodograma de Residuos", col=4, lwd=1, pch=7)
```



Para finalizar el análisis del modelo haremos el cálculo de autocorrelación y la autocorrelación parcial.

```
In [24]: acf(r, lwd=4, col=4)
         pacf(r, lwd=4, col=4)
```



Analicemos la ACF y PACF

Al realizar la ACF vemos que hay una pequeña correlación negativa aproximadamente cada 0,2. La PACF corta luego del 12 así que podemos probar con un modelo $AR(12)$. Aquí conviene por simplicidad el comando Arima de la biblioteca forecast. Puede usarse también arima, que es el básico de R.

Ajustamos un modelo $AR(12)$ al residuo.


```
In [25]: fit1 = Arima(r,order=c(12,0,0),include.mean = FALSE)
summary(fit1)
```

```
Series: r
ARIMA(12,0,0) with zero mean

Coefficients:
          ar1          ar2          ar3          ar4          ar5          ar6          ar7
ar8
      0.3269   0.0730   -0.0622   -0.0496   -0.0850   -0.0180   -0.0213
-0.0311
s.e.   0.0691   0.0744   0.0746   0.0744   0.0744   0.0739   0.0764
0.0756
          ar9          ar10          ar11          ar12
      0.0757   0.043   -0.0446   0.2226
s.e.   0.0758   0.076   0.0758   0.0721

sigma^2 estimated as 217373132:  log likelihood=-2330.1
AIC=4686.2   AICc=4688.04   BIC=4729.84

Training set error measures:
              ME          RMSE          MAE          MPE          MAPE          MAS
E          ACF1
Training set 68.84815 14320.23 10214.18 21.06129 215.4928 0.716565
7 -0.02115123
```

```
In [26]: #Si hacemos r-ajuste lo que estamos obteniendo es residuals(fit) por
lo que esto da lo mismo
RMSE1 = sd(residuals(fit1))
RMSE1
```

```
14353.9583729288
```

```
In [27]: AIC1 = AIC(fit1)
BIC1 = BIC(fit1)
```

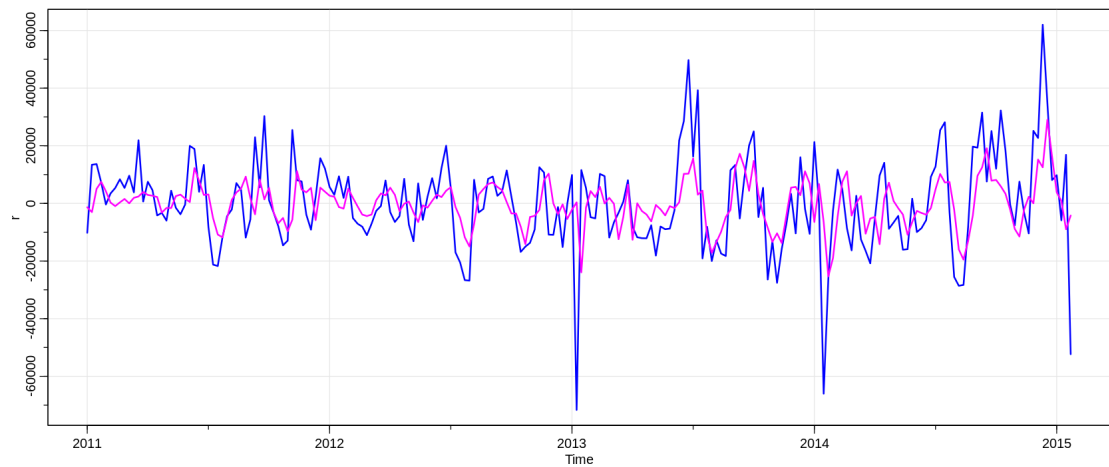
```
AIC1
BIC1
```

```
4686.20205695222
```

```
4729.83767852295
```

```
In [28]: tsplot(r, col=4, lwd=2)
```

```
#Notar que el ajuste de Arima preserva la time series, por lo que no  
tengo que reconstruir los tiempos.  
ajustel = fitted(fit1)  
lines(ajustel,col=6, lwd=2)
```



```
In [29]: #Calculo el desvio estandar de los residuos para ver que coincide co  
n el RMSE  
#La diferencia radica en como considera los grados de libertad en el  
denominador  
#Esta diferencia es mínima para un no. grande de datos  
sd(r-ajustel)
```

```
14353.9583729288
```

Revisemos el resultado del fit1 del modelo Arima, y decidimos ajustar un modelo de regresión lineal con el coeficiente ar12 que es significativo.

```
In [30]: summary(fit1)
```

```
Series: r
ARIMA(12,0,0) with zero mean

Coefficients:
          ar1      ar2      ar3      ar4      ar5      ar6      ar7
ar8
    0.3269   0.0730  -0.0622  -0.0496  -0.0850  -0.0180  -0.0213
-0.0311
s.e.   0.0691   0.0744   0.0746   0.0744   0.0744   0.0739   0.0764
0.0756
          ar9      ar10      ar11      ar12
    0.0757   0.043  -0.0446   0.2226
s.e.   0.0758   0.076   0.0758   0.0721

sigma^2 estimated as 217373132:  log likelihood=-2330.1
AIC=4686.2   AICc=4688.04   BIC=4729.84

Training set error measures:
              ME      RMSE      MAE      MPE      MAPE      MAS
E      ACF1
Training set 68.84815 14320.23 10214.18 21.06129 215.4928 0.716565
7 -0.02115123
```

Los coeficientes significativos son el primero y el de lag 12 (ar12), ya que están a más de dos standard error (s.e.) de 0. Podemos intentar ajustar un modelo de la forma:

$$r_t = \phi r_{t-12}$$

y olvidarnos del resto. Sin embargo, en Arima no podemos forzar los coeficientes a 0. Hay que hacerlo a mano, por ejemplo mediante mínimos cuadrados con la serie laggeada.

```
In [31]: dos_series = ts.intersect(r ,r_12 = lag(r,-12))

fit2 = lm(r ~ 0 + r_12 , data = dos_series)
summary(fit2)
```

```
Call:
lm(formula = r ~ 0 + r_12, data = dos_series)

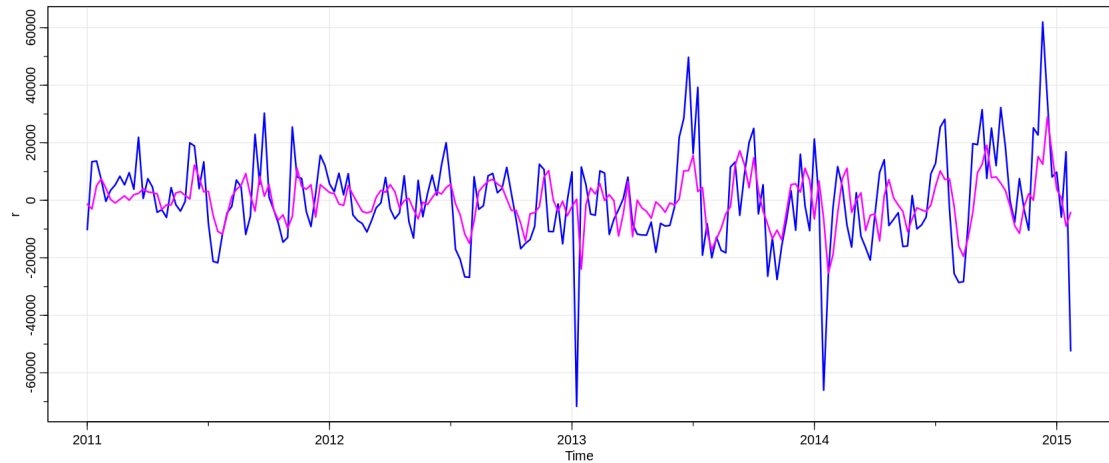
Residuals:
    Min       1Q   Median       3Q      Max
-66494  -9593  -1349    8343   59660

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
r_12    0.30712      0.07378   4.163 4.68e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16000 on 199 degrees of freedom
Multiple R-squared:  0.0801,    Adjusted R-squared:  0.07547
F-statistic: 17.33 on 1 and 199 DF,  p-value: 4.683e-05
```

```
In [32]: tsplot(r, col=4, lwd=2)
```

```
#recupero los datos fitteados y hago una time series (como uso lm, a  
quí no se hace solo)  
ajuste2 = fitted(fit2)  
ajuste2 = ts(ajuste2, start=2011, freq=52)  
lines(ajuste1, col=6, lwd=2)
```



```
In [33]: #Calculo el RMSE de este nuevo ajuste
```

```
RMSE2 = sd(residuals(fit2))  
RMSE2
```

```
15995.1063265997
```

Vemos que este modelo no es mejor que Arima porque tenemos un resultado de RMSE mayor. Por lo tanto nos quedamos con el modelo anterior.

SARIMA

Intentaremos probar un ajuste con el modelo SARIMA. Este modelo capta el comportamiento puramente estacional de una serie, en forma similar, como hemos visto, se realiza para la componente regular o no estacional. Luego haremos también una predicción de seis meses para este modelo.

```
In [34]: fit3= Arima(r, order = c(1, 0, 0), seasonal = list ( order = c(1, 0,
0) ,period = 12), include.mean=FALSE)
summary(fit3)
#tenemos un ciclo de 3 meses que vemos en el periodograma
```

Series: r

ARIMA(1,0,0)(1,0,0)[12] with zero mean

Coefficients:

	arl	sarl
	0.3485	0.2100
s.e.	0.0679	0.0742

sigma^2 estimated as 221151190: log likelihood=-2336.87

AIC=4679.73 AICc=4679.85 BIC=4689.8

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MA
SE						
Training set	-7.293951	14800.84	10447.98	17.89519	209.2579	0.7329679

ACF1

Training set -0.001947362

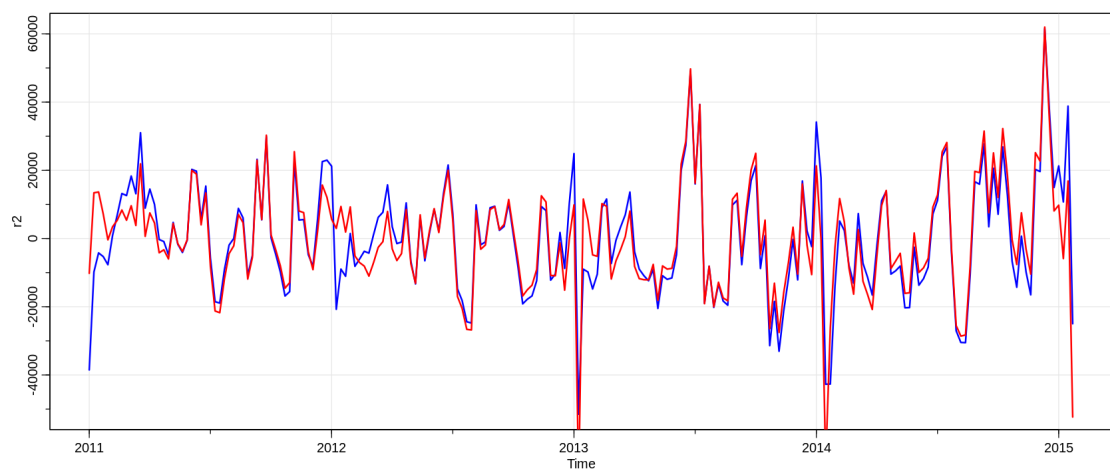
```
In [35]: RMSE3 = sd(residuals(fit3))
RMSE1
RMSE3
```

14353.9583729288

14835.8695865329

Comparamos los residuos de los ajustes realizados.

```
In [36]: r2=ts(residuals(fit5),start=2011,freq=52)
tsplot(r2, lwd=2, col=4)
lines(r,col= 2,lwd=2)
```



CONCLUSION

Luego de analizar varios modelos y utilizar las técnicas vistas en clase, concluimos que el modelo que mejor ajusta a los residuos es SARIMA 12. Entendemos que tiene un valor un poco mayor de RMSE con relación al ARIMA de orden 12 pero solo utiliza dos coeficientes.

Predicción de seis meses para adelante (en semanas)

Para continuar el análisis vamos a realizar una serie de predicciones de los valores de las ventas a futuro usando la función "predict". Estos puntos de tiempo tienen que ser la *continuación* de las series *t* y *week* en este caso.

```
In [37]: ## Construimos los nuevos datos a evaluar el modelo de regresión lineal

#Agregamos tiempos para 24 semanas en el futuro. para el calculo de
by tomamos 1/52
new_t = seq(2015,2015.4615,by=0.01923) - 2011
#Lo convierto en serie temporal
new_t = ts(new_t,start=2015,freq=52)
#Le agrego los factores
new_week = new_t - 4

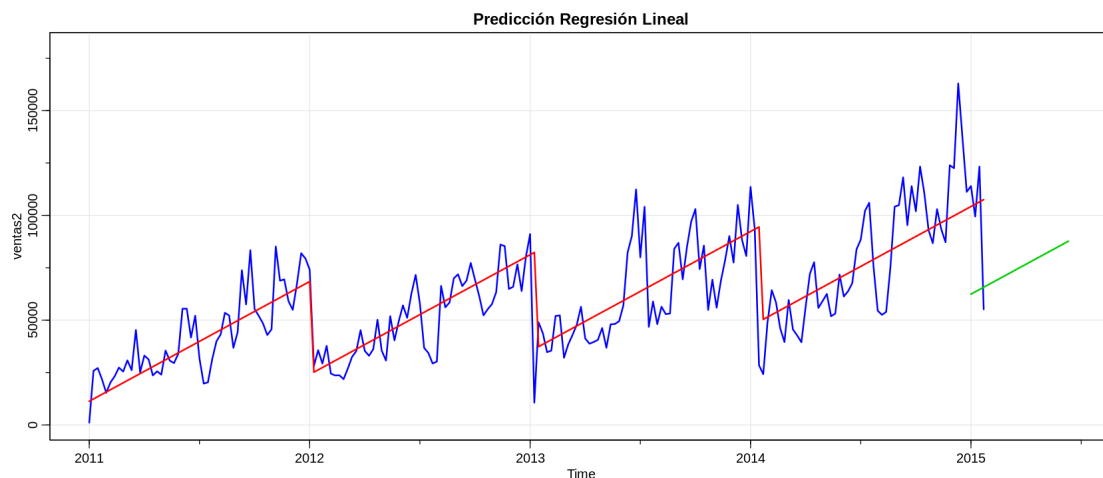
#Predigo sobre los nuevos datos.
new_data = data.frame(t=new_t,week=new_week)
prediccion= predict(fit, new_data);

#Convierto el resultado en time series
prediccion = ts(prediccion,start=2015,freq=52)
prediccion
```

A Time Series:

1: 62423.0878380963 **2:** 63521.8305245417 **3:** 64620.5732109741 **4:** 65719.3158974195
5: 66818.0585838519 **6:** 67916.8012702973 **7:** 69015.5439567298 **8:**
70114.2866431752 **9:** 71213.0293296076 **10:** 72311.772016053 **11:** 73410.5147024854
12: 74509.2573889308 **13:** 75608.0000753632 **14:** 76706.7427618086 **15:**
77805.485448241 **16:** 78904.2281346864 **17:** 80002.9708211188 **18:** 81101.7135075642
19: 82200.4561940097 **20:** 83299.1988804421 **21:** 84397.9415668875 **22:**
85496.6842533199 **23:** 86595.4269397653 **24:** 87694.1696261977

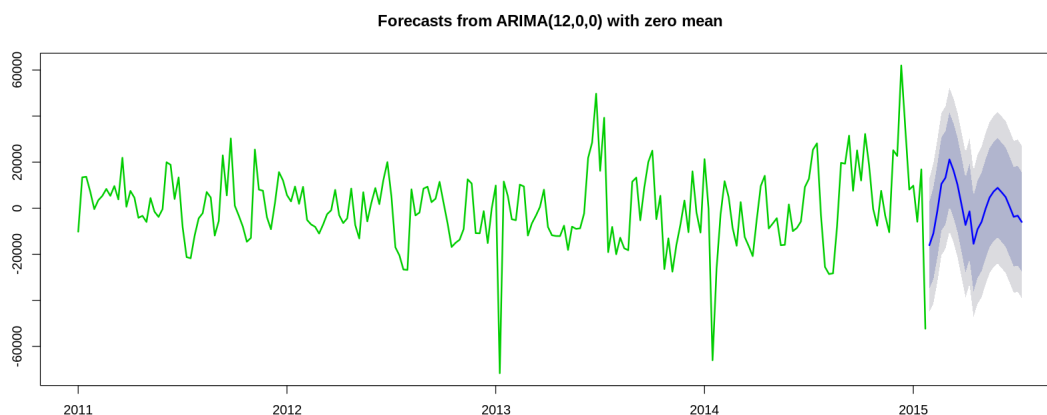
```
In [38]: tsplot(ventas2,xlim=c(2011,2015.4615),ylim=c(0,180000), lwd=2, col=
4, main="Predicción Regresión Lineal")
lines(ajuste,lwd=2, col=2)
lines(prediccion,lwd=2, col=3)
```



Ahora haremos la predicción de 6 meses para adelante del modelo ARIMA que ajustamos sobre los residuos, utilizando la función "forecast".

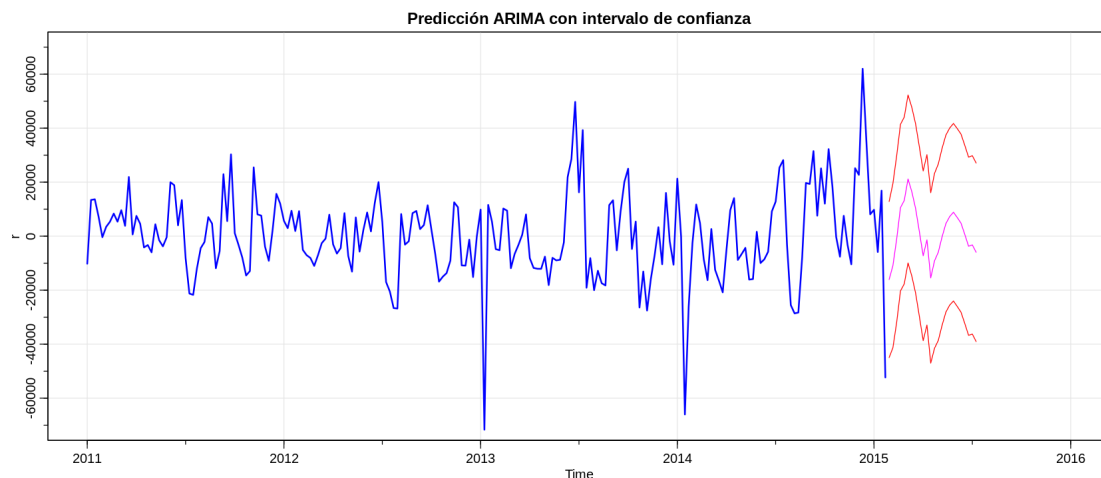
```
In [39]: prediccion1 = forecast(fit1,h=24)

plot(prediccion1,col=3, lwd=2)
```



```
In [40]: #Podemos extraer la media de la predicción y los extremos del intervalo de confianza al 95%
prediccion1_media = prediccion1$mean
prediccion1_lower = prediccion1$lower[,2] #La segunda columna de lower tiene el valor del IdC al 95% por abajo
prediccion1_upper = prediccion1$upper[,2] #La segunda columna de upper tiene el valor del IdC al 95% por arriba
```

```
In [41]: tsplot(r,xlim=c(2011,2016),ylim=c(-70000,70000), lwd=2, col=4, main
="Predicción ARIMA con intervalo de confianza")
lines(prediccion1_medio,col=6)
lines(prediccion1_lower,col=2)
lines(prediccion1_upper,col=2)
```

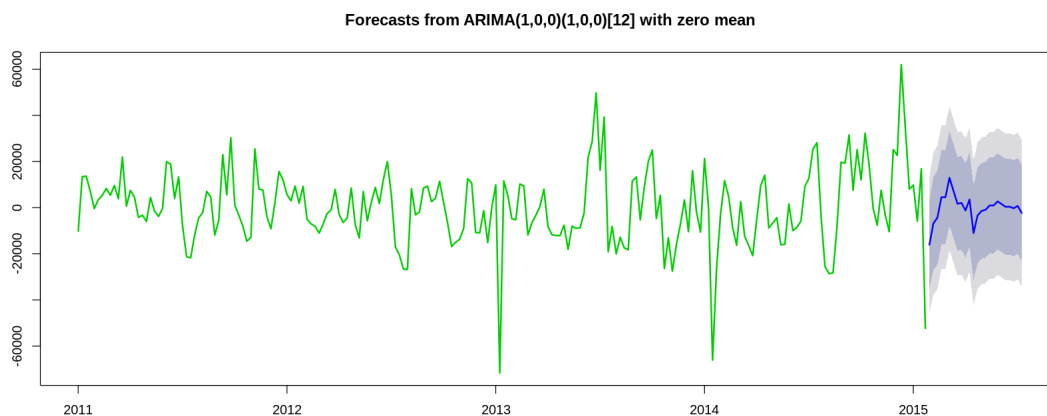


Nota sobre los intervalos de confianza:

Formalmente, el intervalo anterior asume que la varianza solo es producto de la parte AR ajustada a los residuos. En realidad también se agrega incertidumbre debido al modelo lineal ajustado para tendencia y estacionalidad. Sin embargo, no es fácil combinar ambas cosas. De todos modos en general constituye una buena estimación.

Ahora haremos la predicción de 6 meses para adelante del modelo SARIMA que ajustamos sobre los residuos, utilizando la función "forecast".

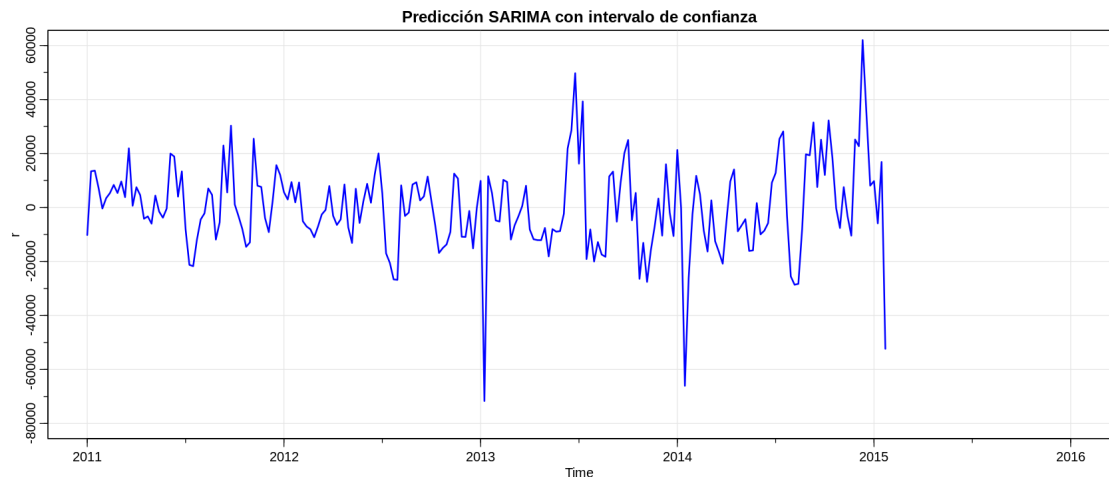
```
In [42]: #predicción de SARIMA
prediccion2 = forecast(fit3,h=24)
plot(prediccion2,col=3, lwd=2)
```




```
In [43]: tsplot(r,xlim=c(2011,2016),ylim=c(-80000,60000), lwd=2, col=4,main="
Predicción SARIMA con intervalo de confianza")
lines(prediccion2_media,col=3)
lines(prediccion2_lower,col=1)
lines(prediccion2_upper,col=1)
```

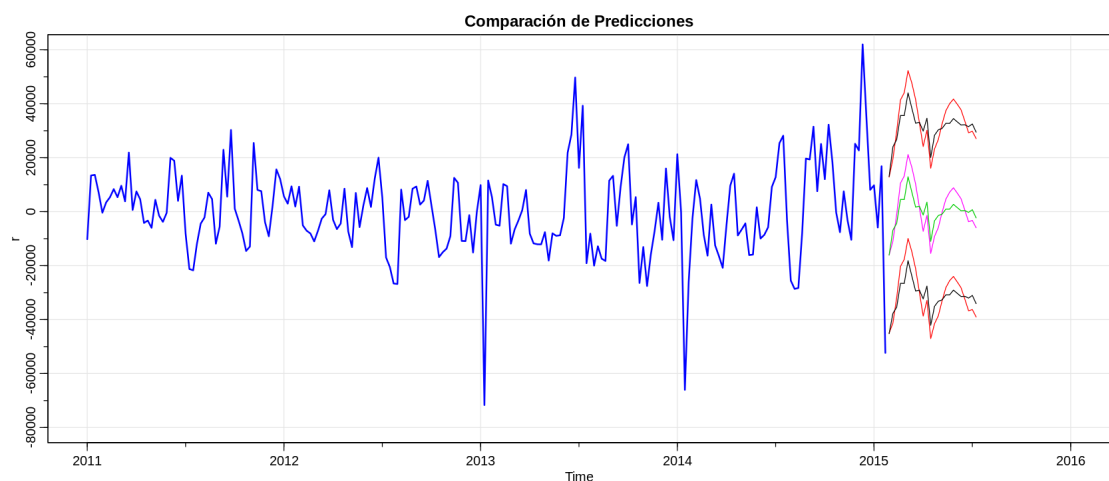
Error in lines(prediccion2_media, col = 3): object 'prediccion2_media' not found
Traceback:

```
1. lines(prediccion2_media, col = 3)
```



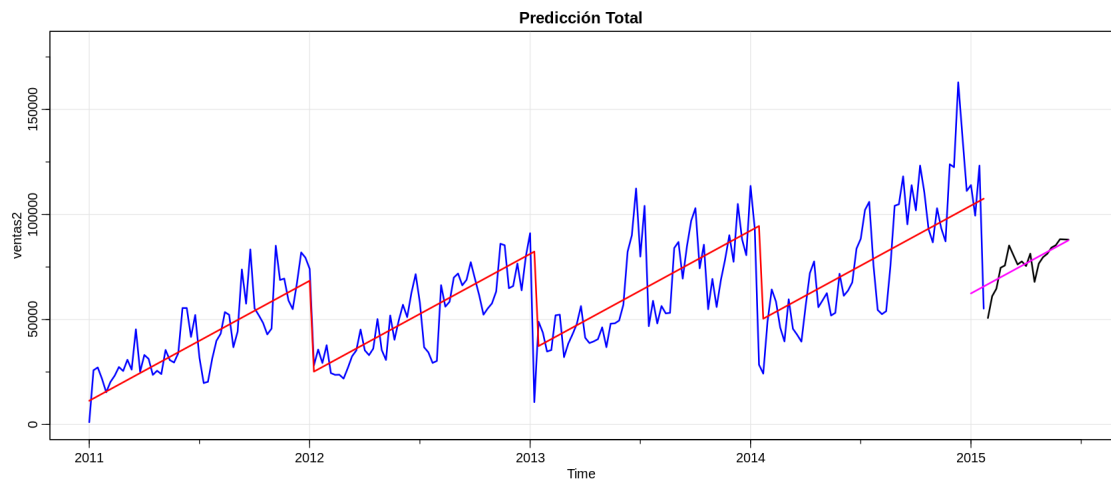
```
In [44]: #Podemos extraer la media de la predicción y los extremos del intervalo de confianza al 95%
prediccion2_media = prediccion2$mean
prediccion2_lower = prediccion2$lower[,2] #La segunda columna de lower tiene el valor del IdC al 95% por abajo
prediccion2_upper = prediccion2$upper[,2] #La segunda columna de upper tiene el valor del IdC al 95% por arriba
```

```
In [45]: tsplot(r,xlim=c(2011,2016),ylim=c(-80000,60000), lwd=2, col=4, main="
Comparación de Predicciones")
lines(prediccion1_media,col=6)
lines(prediccion1_lower,col=2)
lines(prediccion1_upper,col=2)
lines(prediccion2_media,col=3)
lines(prediccion2_lower,col=1)
lines(prediccion2_upper,col=1)
```



La siguiente gráfica contiene la predicción del ajuste de rampa y la predicción de los valores de la serie ventas. Dichos valores se calcularon sumando los del ajuste de regresión lineal más los valores de los residuos calculados con SARIMA

```
In [46]: predicciont = prediccion + prediccion2_media
tsplot(ventas2,xlim=c(2011,2015.4615),ylim=c(0,180000), lwd=2, col=
4, main="Predicción Total")
lines(ajuste,lwd=2, col=2)
lines(predicciont,lwd=2, col=1)
lines(prediccion,lwd=2, col=14)
```



Correlación entre Variables

Analizaremos si existe correlación entre la variable ventas que hemos analizado hasta el momento y la variable descuento, para verificar si hay mayores ventas al tener mayor descuento.

```
In [47]: ventas3 <- ventas %>%  
  group_by(año, semana) %>%  
  summarise(Total_Sales = sum(Sales),  
    Total_Discount = mean(Discount))  
  
ventas3
```

```
`summarise()` regrouping output by 'año' (override with `.groups` argument)
```

A grouped_df: 212 × 4

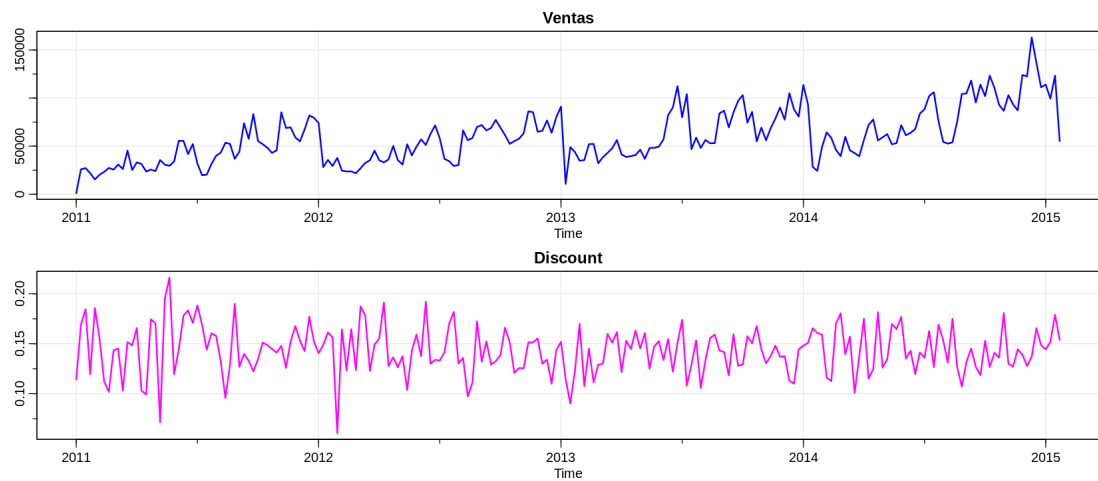
año	semana	Total_Sales	Total_Discount
<dbl>	<dbl>	<dbl>	<dbl>
2011	0	1122.783	0.11428571
2011	1	25827.283	0.16883871
2011	2	27169.672	0.18446018
2011	3	21799.136	0.11970833
2011	4	15337.029	0.18576923
2011	5	20268.441	0.15511111
2011	6	23255.994	0.11197959
2011	7	27372.644	0.10184946
2011	8	25479.347	0.14342056
2011	9	30824.712	0.14509524
2011	10	26137.756	0.10303093
2011	11	45315.857	0.15165049
2011	12	25148.896	0.14841935
2011	13	33104.578	0.16574194
2011	14	31269.551	0.10301613
2011	15	23633.672	0.09918033
2011	16	25605.471	0.17450000
2011	17	24022.956	0.17049180
2011	18	35495.099	0.07131507
2011	19	30693.585	0.19486765
2011	20	29526.258	0.21629032
2011	21	33984.827	0.11945455
2011	22	55457.941	0.14443434
2011	23	55462.548	0.17815179
2011	24	41714.262	0.18348454
2011	25	52137.213	0.17088889
2011	26	31717.094	0.18843137
2011	27	19743.923	0.16876471

año	semana	Total_Sales	Total_Discount
<dbl>	<dbl>	<dbl>	<dbl>
2011	28	20345.803	0.14409259
2011	29	31321.396	0.16047407
:	:	:	:
2014	23	88435.48	0.1360703
2014	24	102155.83	0.1627106
2014	25	106010.71	0.1265843
2014	26	75860.89	0.1690978
2014	27	54529.96	0.1536996
2014	28	52551.24	0.1309851
2014	29	53963.25	0.1749516
2014	30	75333.98	0.1269556
2014	31	104145.32	0.1070773
2014	32	104876.65	0.1320354
2014	33	118144.36	0.1451002
2014	34	95327.16	0.1269356
2014	35	113942.36	0.1186103
2014	36	101984.15	0.1529028
2014	37	123273.14	0.1266500
2014	38	110573.46	0.1408589
2014	39	92938.16	0.1360838
2014	40	86737.32	0.1806848
2014	41	102970.47	0.1301197
2014	42	93373.00	0.1270182
2014	43	87211.24	0.1444659
2014	44	123896.23	0.1390185
2014	45	122530.74	0.1277062

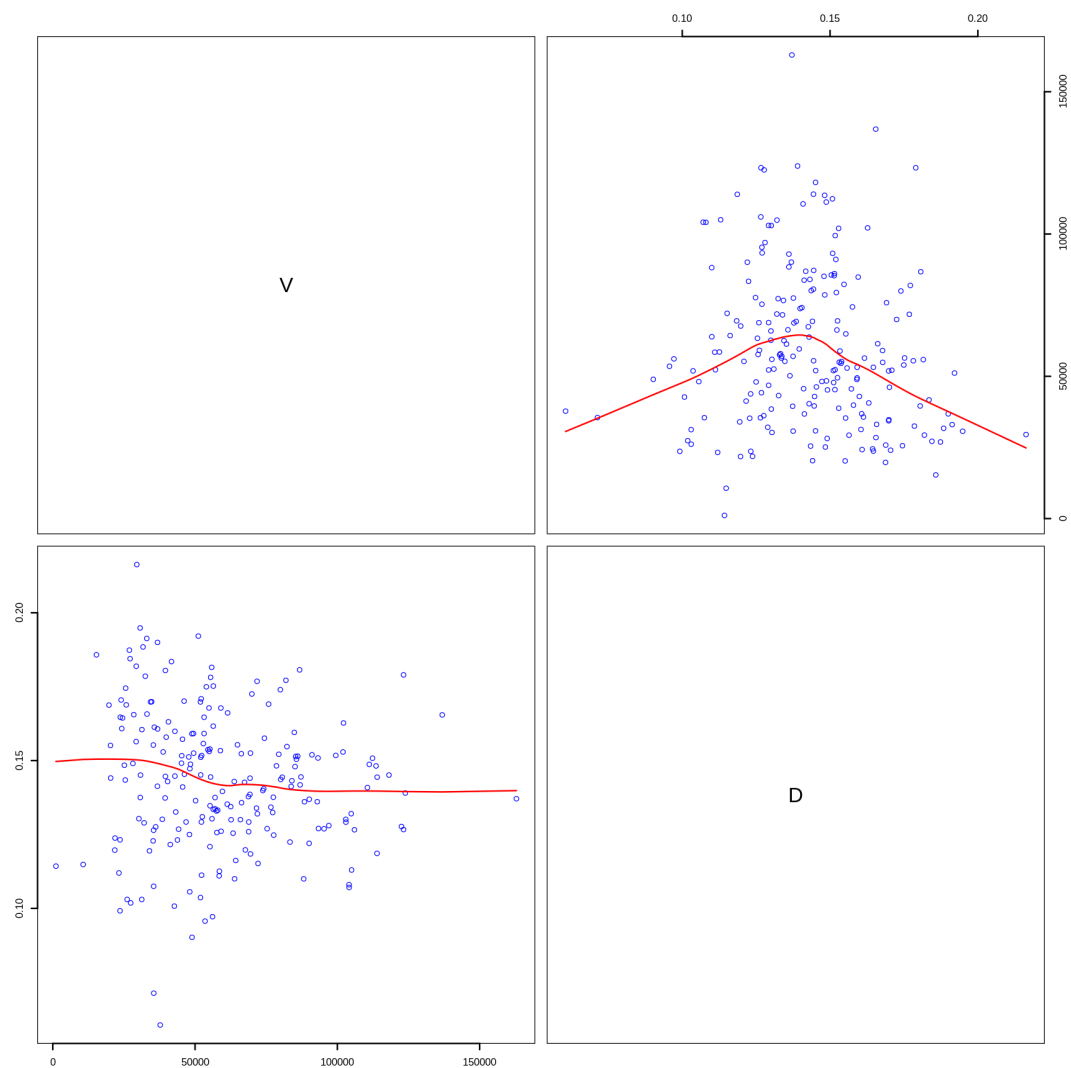
```
In [48]: V <- ts(ventas3$Total_Sales, start = 2011, frequency = 52)
D <- ts (ventas3$Total_Discount, start = 2011, frequency = 52)
```

```
In [49]: par(mfrow=c(2,1))
V=V;
D=D;

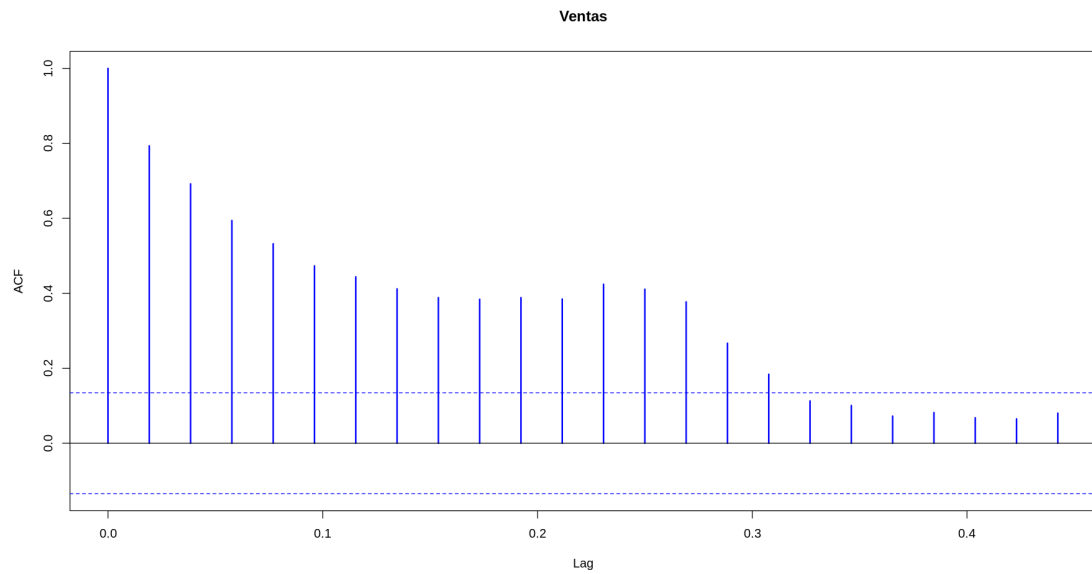
tsplot(V, main="Ventas", ylab="", lwd=2, col=4)
tsplot(D, main="Discount", ylab="", lwd=2, col=6)
```



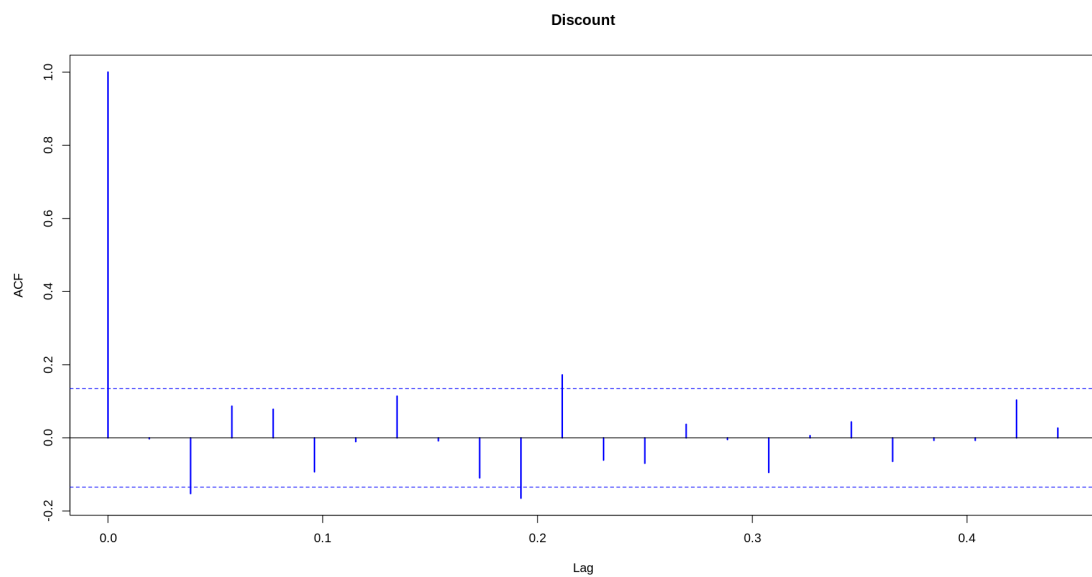
```
In [50]: options(repr.plot.width=15, repr.plot.height=15)
pairs(cbind(V, D), col=4, panel=panel.smooth, lwd=2)
```



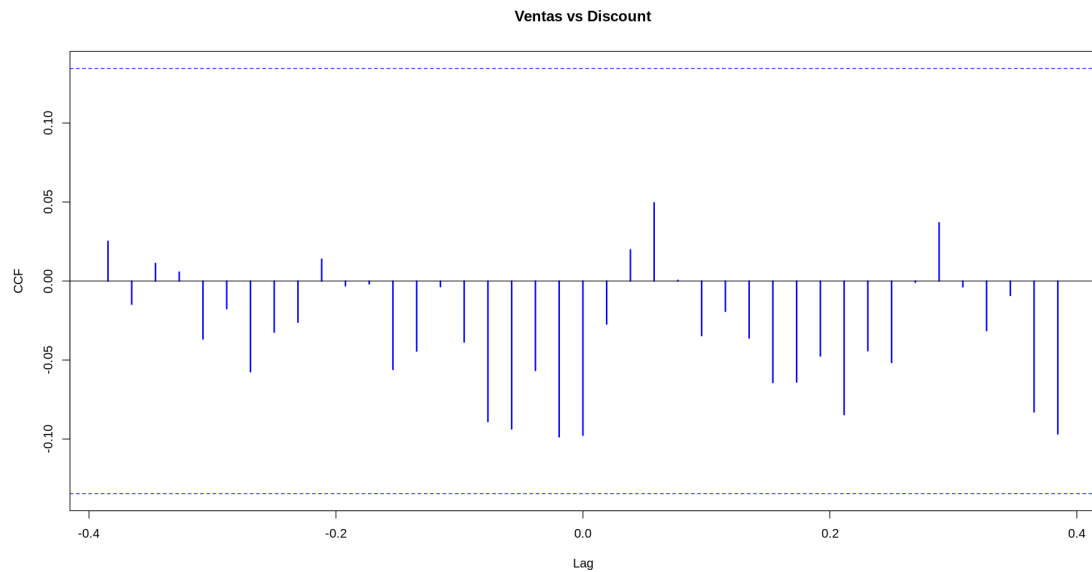
```
In [51]: options(repr.plot.width=15, repr.plot.height=8)
         acf(V, main="Ventas", col=4, lwd=2)
```



```
In [53]: options(repr.plot.width=15, repr.plot.height=8)
         acf(D, main="Discount", col=4, lwd=2)
```



```
In [54]: options(repr.plot.width=15, repr.plot.height=8)
ccf(D, V, main="Ventas vs Discount", ylab="CCF", col=4, lwd=2)
```



Analizando los resultados obtenidos podemos afirmar que no hay correlación entre las variables ventas y descuentos. Procederemos a estudiar la correlación con otra variable. Usaremos a continuación el Shipping Cost


```
In [55]: ventas4 <- ventas %>%  
  group_by(año, semana) %>%  
  summarise(Total_Sales = sum(Sales),  
    Total_Scost = sum(Shipping.Cost))  
  
ventas4
```

```
`summarise()` regrouping output by 'año' (override with `.groups` argument)
```

A grouped_df: 212 × 4

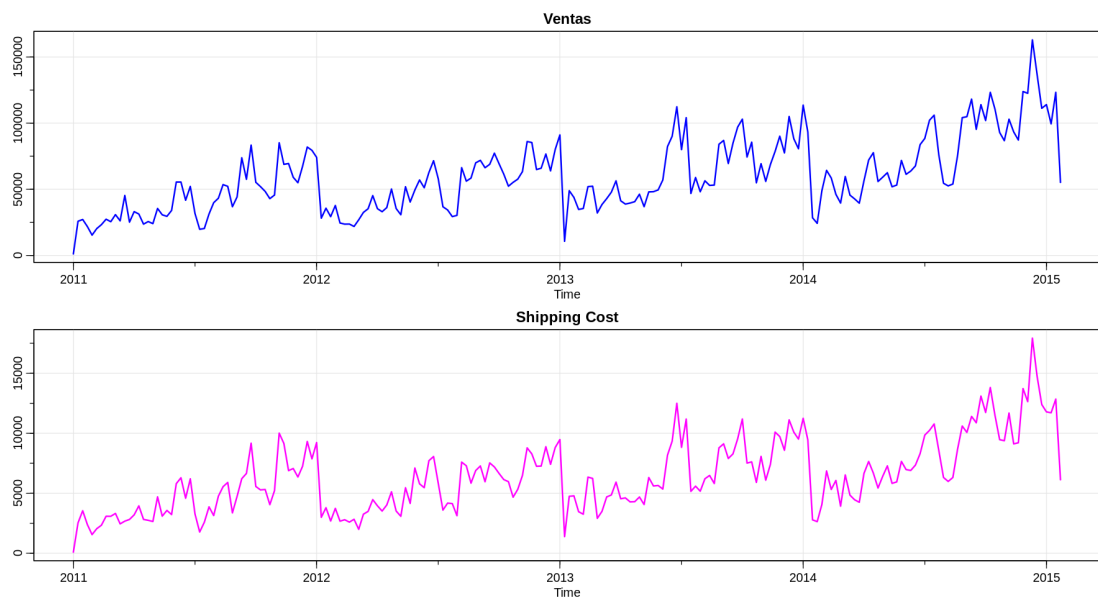
año	semana	Total_Sales	Total_Scost
<dbl>	<dbl>	<dbl>	<dbl>
2011	0	1122.783	88.77
2011	1	25827.283	2534.90
2011	2	27169.672	3543.94
2011	3	21799.136	2387.35
2011	4	15337.029	1552.90
2011	5	20268.441	2047.14
2011	6	23255.994	2337.25
2011	7	27372.644	3084.14
2011	8	25479.347	3080.32
2011	9	30824.712	3319.03
2011	10	26137.756	2448.46
2011	11	45315.857	2674.81
2011	12	25148.896	2817.00
2011	13	33104.578	3200.73
2011	14	31269.551	3950.36
2011	15	23633.672	2819.59
2011	16	25605.471	2744.19
2011	17	24022.956	2645.98
2011	18	35495.099	4698.57
2011	19	30693.585	3092.10
2011	20	29526.258	3568.69
2011	21	33984.827	3223.45
2011	22	55457.941	5800.51
2011	23	55462.548	6290.07
2011	24	41714.262	4585.40
2011	25	52137.213	6202.64
2011	26	31717.094	3278.00
2011	27	19743.923	1769.86

año	semana	Total_Sales	Total_Scost
<dbl>	<dbl>	<dbl>	<dbl>
2011	28	20345.803	2583.99
2011	29	31321.396	3867.31
⋮	⋮	⋮	⋮
2014	23	88435.48	9848.27
2014	24	102155.83	10236.07
2014	25	106010.71	10774.84
2014	26	75860.89	8552.72
2014	27	54529.96	6313.21
2014	28	52551.24	5984.05
2014	29	53963.25	6314.16
2014	30	75333.98	8663.30
2014	31	104145.32	10602.23
2014	32	104876.65	10068.67
2014	33	118144.36	11403.55
2014	34	95327.16	10873.14
2014	35	113942.36	13096.01
2014	36	101984.15	11733.79
2014	37	123273.14	13812.96
2014	38	110573.46	11495.97
2014	39	92938.16	9464.43
2014	40	86737.32	9379.60
2014	41	102970.47	11682.34
2014	42	93373.00	9114.63
2014	43	87211.24	9207.69
2014	44	123896.23	13724.40
2014	45	122530.74	12635.76

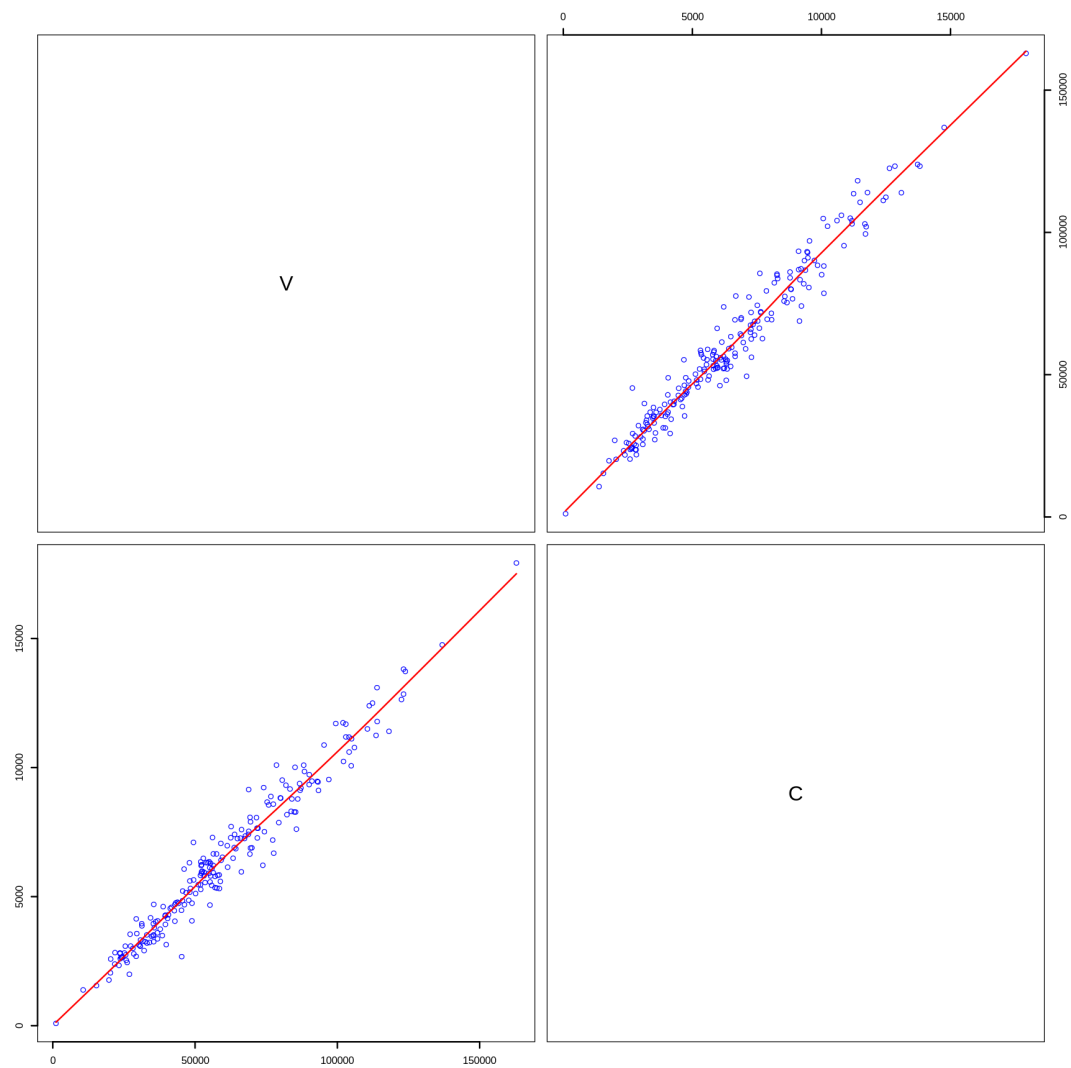
```
In [56]: C <- ts(ventas4$Total_Scost, start = 2011, frequency = 52)
```

```
In [57]: par(mfrow=c(2,1))
V=V;
C=C;

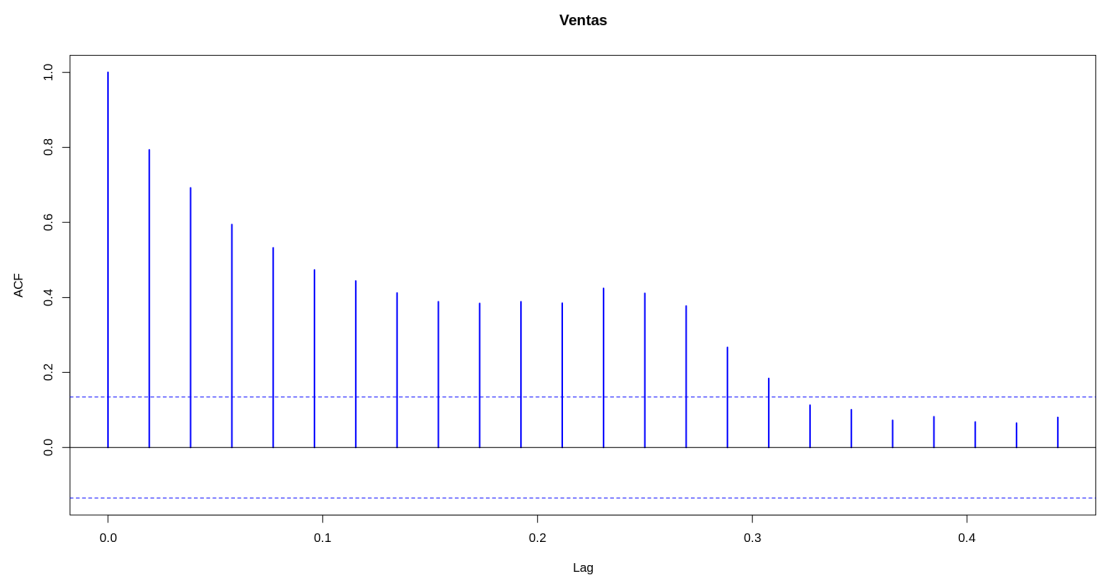
tsplot(V, main="Ventas", ylab="", lwd=2, col=4)
tsplot(C, main="Shipping Cost", ylab="", lwd=2, col=6)
```



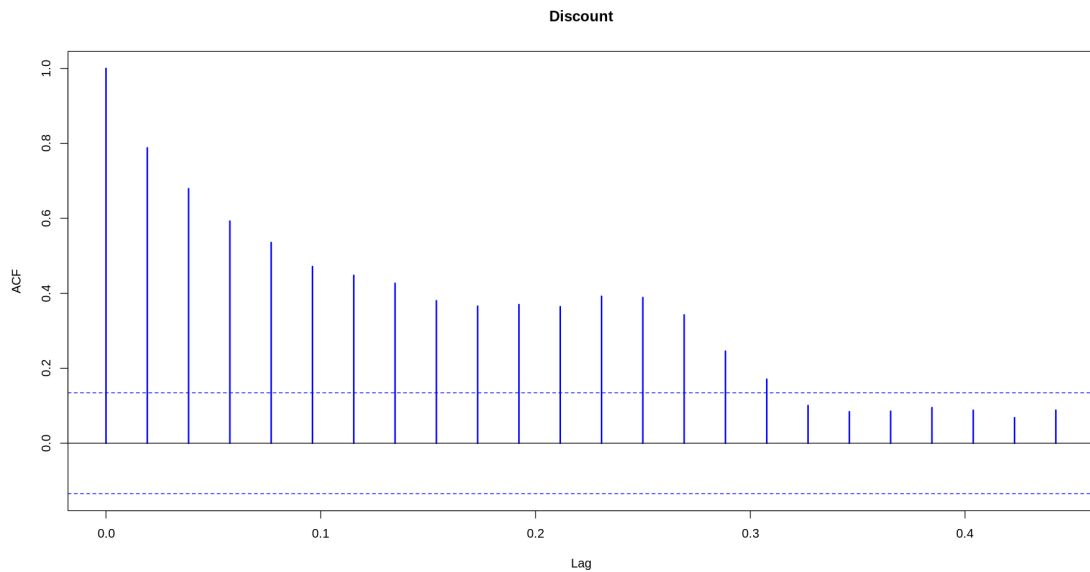
```
In [58]: options(repr.plot.width=15, repr.plot.height=15)
pairs(cbind(V, C), col=4, panel=panel.smooth, lwd=2)
```



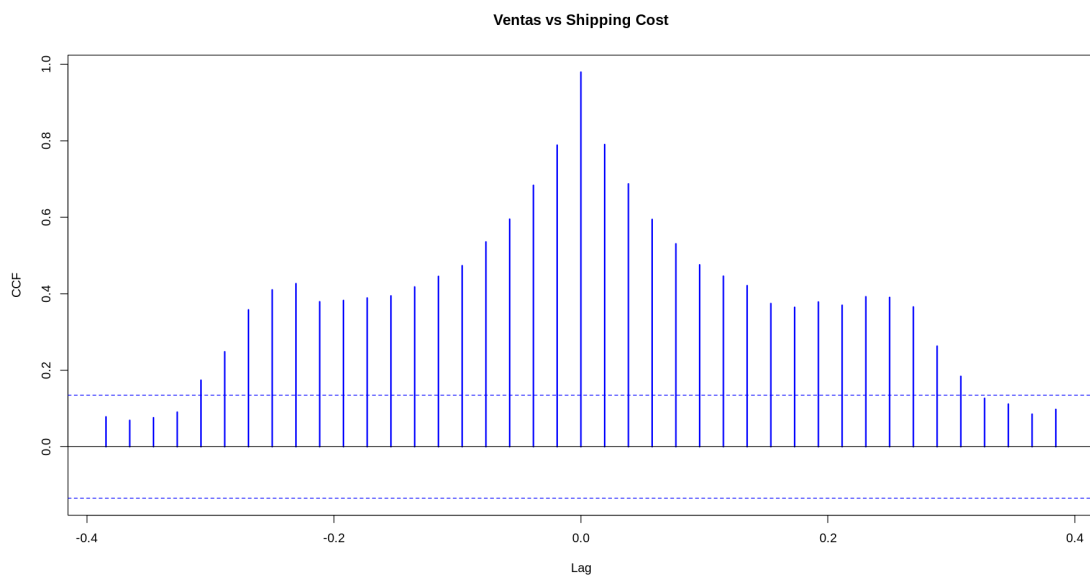
```
In [59]: options(repr.plot.width=15, repr.plot.height=8)
acf(V, main="Ventas", col=4, lwd=2)
```



```
In [60]: options(repr.plot.width=15, repr.plot.height=8)
         acf(C, main="Discount", col=4, lwd=2)
```



```
In [63]: options(repr.plot.width=15, repr.plot.height=8)
         ccf(V, C, main="Ventas vs Shipping Cost", ylab="CCF", col=4, lwd=2)
```



Podemos ver que el costo de envío (shipping cost) se comporta de manera similar a la ventas. Al hacer las gráficas de ACF visualizamos su comportamiento casi identico. Al hacer la correlación cruzada vemos que existe, es positiva y casi simétrica. Podemos suponer que el costo de envío es calculado en base a un porcentaje de las ventas, en el entorno del 10%.