UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

LOS VINOS DE VAR

ALMUNO:
DEL VALLE LÓPEZ JOSÉ CARLOS

PROFESOR:
ACT. DAVID CHAFFREY MORENO FERNÁNDEZ
FACULTAD DE CIENCIAS, UNAM

MÉXICO, D.F. NOVIEMBRE 2016

LOS VINOS DE VAR

SERIES DE TIEMPO

José Carlos Del Valle López

22 de noviembre de 2016

Resumen del proyecto

El siguiente proyecto es un análisis de las series de tiempo empleando el modelo clásico de descomposición, se pretende fragmentar nuestro proceso en una componente de tendencia, una componente estacional y un proceso de ruido blanco.

Nuestra serie de tiempo son las ventas de vino a lo largo de 120 meses.

Aunado a ello, aplicaremos nuestros conocimientos en programación lineal para maximizar las ganancias de la venta de vinos sujeto a distintas restricciones.

Haremos uso de distintos softwares que nos ayudarán a resolver nuestro problema, es concreto: *R*, *ITSM* y *MatLab*

Índice

1. Introducción

Tras una larga negociación, los propietarios del afamado viñedo de Var, conocido por producir los vinos *Cõtes de Provence, Coteaux Varois* y *Vins de Bandol*, han accedido a cedernos una parte de su producción.

Tomando el control en la prodicción de los *Vins de Bandol*, nuestro objetivo es obtener la mayor ganancia posible de las ventas, también tomemos en cuenta que contamos con $100,000 y una cuenta bancaria que nos ofrece un 4 % de interés mensual.

1.1. Historia del viñedo

El lugar se edificó en el siglo XIX por un propietario industrial lilés, el *Château* se halla sobre una ladera orientada hacia el Sur. Con su frontera en el puerto de Cassis.
Durante esa época era reconocido por la alta aristocracia de aquella provincia.
En 1969 Paul y Pierre Bunan emprendieron la replantación de este viñedo ahora compuesto de cincos cepas.

Poseé el suelo más favorable, el mejor soleamiento y las cepas más antiguas.
Sus estrechas terrazas en ladera permite recibir de una hasta tres hileras de cepas, subiendo frente al sol, hasta el viñedo.[**?**]

2. Estructura del problema

2.1. ¿Cómo ganaremos dinero?

El costo de producción de una botella es de \$10.

El valor de una botella de vino en el mercado es de \$25, es decir, ganamos \$15 por unidad.

El costo de almacenamiento de botella de vino durante un mes es de \$10, es decir, si la guardmos un mes y la vendemos el siguiente nuestra ganancie es solamente de \$5.

Una botella sin almacenar será donada a la caridad, con ello las pérdidas son de \$10 por unidad.

Nuesra producción máxima es de 2300 unidades mensuales y el almacén cuenta con una capacidad de 1500 botellas.

Además, hay periodos en los que la demanda superará nuestra producción, por lo que debemos elegir entre almacenar cierta cantidad el mes anterior o regalarla para minimizar costos.

En vista de ello, una botella almacenada 2 meses representará una pérdida de \$5, por lo que se elegirá donarla, entonces, si nuestra demanda supera nuestra cantidad almacenada más la producción, elegiremos no satisfacer esa demanda por completo.

2.2. ¿Qué información tenemos?

Afortunadamente, los dueños llevaban un registro mensual de sus ventas mostrado en la figura [2.2.1].

Gracias a estos datos nos enfocaremos en predecir las ventas de los siguientes 22 meses.

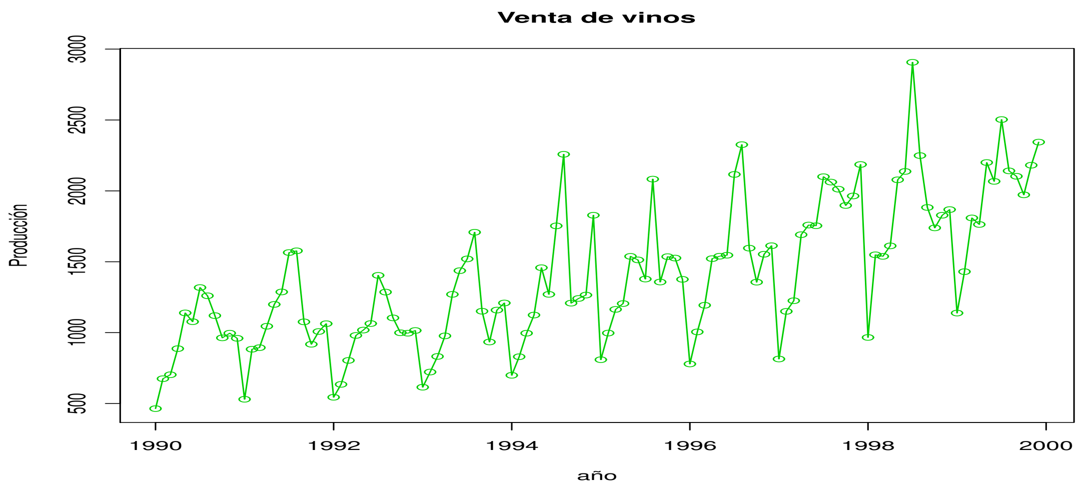


Figura 1: [2.2.1]

3. Resolución del problema en R

3.1. Serie de tiempo en R

Para obtener más información sobre nuestra base, haremos uso de R. El código se hallará en conjunto con los datos arrojados por el software.

Antes que nada debemos cargar las librerías que nos serán de utilidad para las series de tiempo.

```
library(MASS)
library(forecast)
library(tseries)
library(itsmr)
```

Primero cargamos la base.

```
base<-read.table(choose.files())
```

Ahora toca convertir nuestra base en una serie de tiempo, sabemos que nuestros datos son registrados mensualmente, es decir, hay una frecuencia de 12, y dado que tenemos 120 registros, tomaremos una diferencia de 10 años entre el primer y el último dato, el inicio será en enero de 1990 por poner un ejemplo

```
serie<-ts(base,start=c(1990,1),end=c(1999,12), freq=12)
```

Con el comando summary examinaremos la información básica de nuestros datos.

```
summary(serie)
```

Min.	:	464
1st Qu.	:	1004
Median	:	1286
Mean	:	1393
3rd Qu.	:	1753
Max.	:	2907

Si deseamos ver los promedios de ventas por periodo usaremos el siguiente comando:

```
for(i in 1:12){
mp<-mean(serie[seq(i,120,12)])
print(mp)}
```

Enero	735.8	Julio	1856.4
Febrero	987.6	Agosto	1895
Marzo	1115.8	Septiembre	1461
Abril	1280.6	Octubre	1355.5
Mayo	1519.8	Noviembre	1447.6
Junio	1515.2	Diciembre	1546.2

A partir de esto, podemos observar que hay mucha diferencia entre las ventas mensuales máximas y mínimas reportadas, además, las ventas máximas exceden nuestra capacidad de producción mensual, por lo que será necesario recurrir a las botellas almacenadas en más de una ocasión, también es bueno observar que ni la media muestral ni la media por periodo superan nuestra capacidad de producción, y si nuestra serie presenta una componente de tendencia y una estacionaria, seremos capaces de satisfacer la demanda del mercado en la mayoría de los periodos.

Ya definida nuestra serie, echaremos un vistazo a nuesras gráficas: La que corresponde a nuestra serie de tiempo [3.1.1] y la que corresponde a nuestra base [3.1.2].

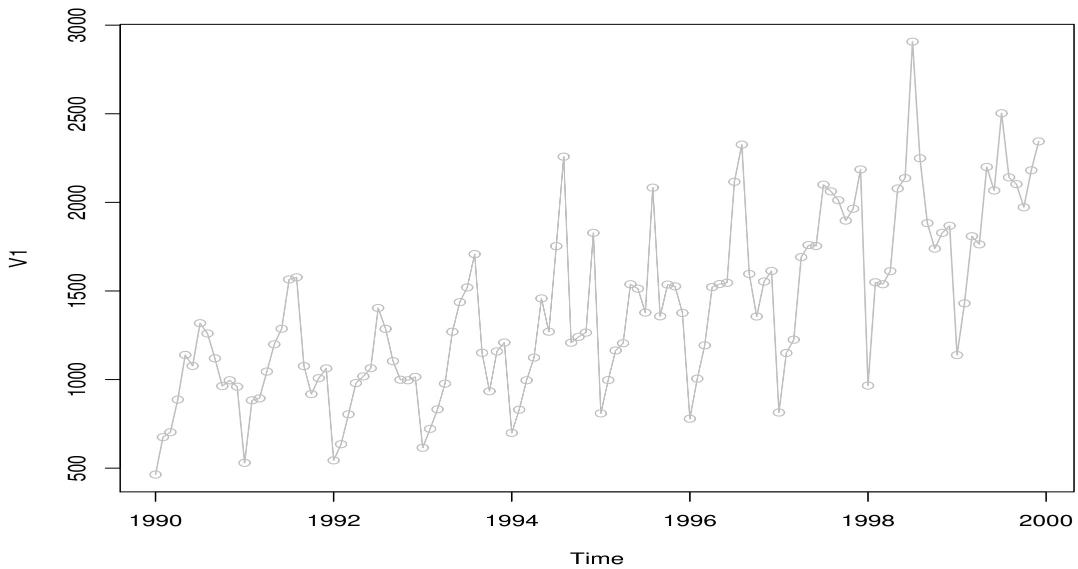


Figura 2: [3.1.1]

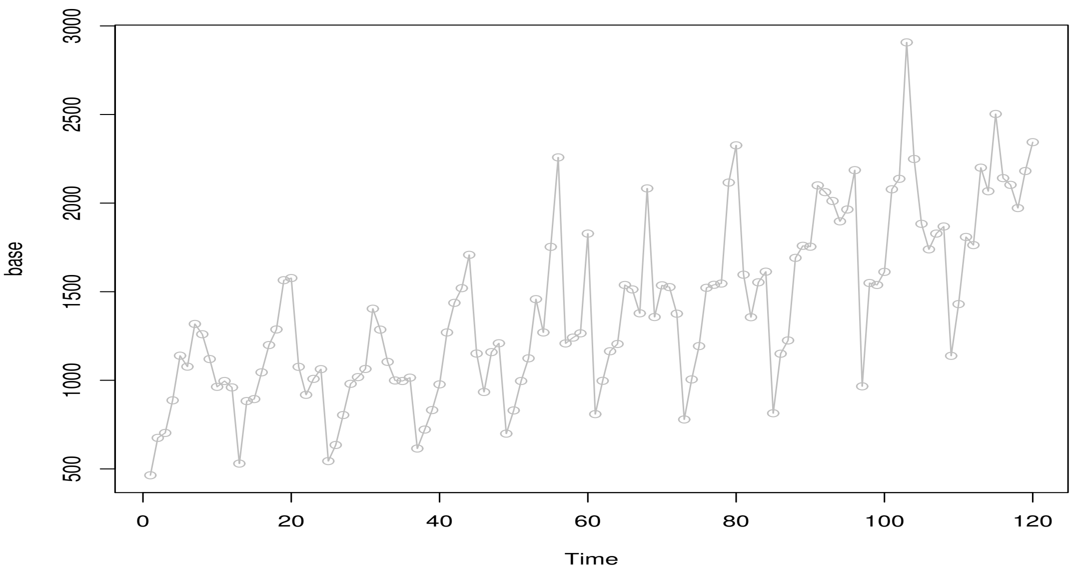


Figura 3: [3.1.2]

No existe mucha diferencia entre los gráficos anteriores, el [3.1.1] marca los periodos en meses y [3.1.2] en años. De los gráficos anteriores podemos destacar lo siguiente: Existe un patrón que se repite cada 12 meses.
El mes con menor número de ventas es enero.
En contraste, los meses con más ventas son julio y en especial agosto.
Ha habido un pequeño crecimiento en las ventas durante cada año.

3.2. Box-Cox y Máximo Verosimil

En esta sección obtendremos el Máximo Verosimil a partir de Box-Cox, que nos ayudará a aplicar la descomposición a nuestra serie. Para emplear Box-Cox escribiremos:

```
boxcox(serie~1)
```

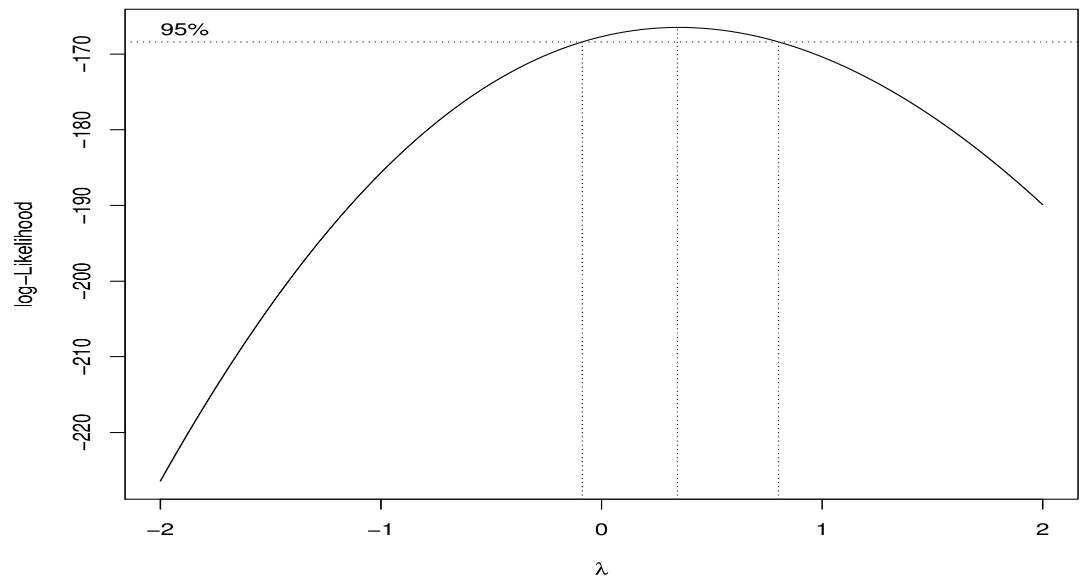


Figura 4: [3.2.1]

La gráfica [3.2.1] nos muestra el perfil log-verosimilitud para el parámetro de la transformación de potencia de Box-Cox.

Ahora, para hallar el lambda máximo verosimil de Box-Cox, debemos ingresar el siguiente comando:

```
lamb<-boxcox(serie~1)$x[which.max(boxcox(serie~1)$y)]
```

Lo que nos arrojó un valor de 0.3434343

Ahora veremos nuestros datos en función de nuestra serie y el lambda que obtuvimos con el siguiente comando:

```
transbc<-BoxCox(serie,lamb)
```

3.3. Descomosición en componentes

Nuestra serie de tiempo se compone de tres distintas componentes, una estacional, una tendencia y un proceso de ruido blanco. La forma de visualizar dentro de *R* cada una de las componentes es a través del comando:

```
stl(transbc,s.window="periodic")
```

Es posible que nos arroje un error: .Error in stl() : only univariate series are allowed", en caso de presentarse, hay convertir en vector y nuevamente en una serie de tiempo nuestra serie antes definida como *transbc* e la siguiente forma.

```
dummyVector = as.vector(t(transbc))
tsData = ts(dummyVector, start=c(1990,1),end=c(1999,12), frequency = 12)
stl = stl(tsData, "periodic")
plot(stl)
```

Lo que nos arroja la figura[3.3.1], donde data es nuestra serie original, seasonal es nuestra componente estacional, trend nuestra componente de tendencia y remainder nuestro proceso de ruido blanco:

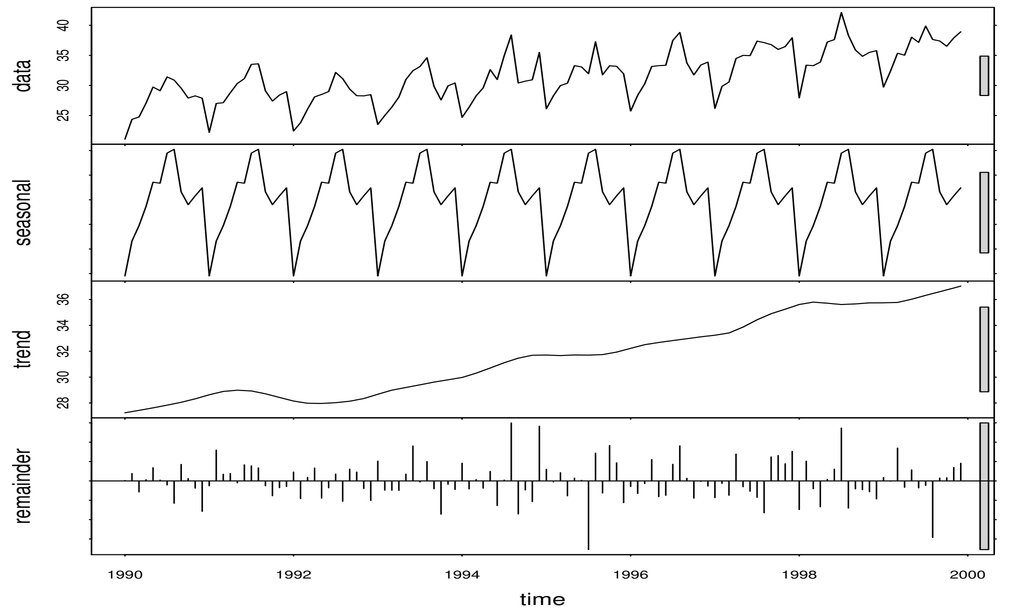


Figura 5: [3.3.1]

3.4. Eliminación de componentes de tendencia y estacional

Ya que nuestra serie pudo ser descompuesta en una componente estacional, una comonente de tendencia y un proceso de ruido blanco, ahora se puede hacer uso de la diferenciación para deshacernos de dichas componentes.

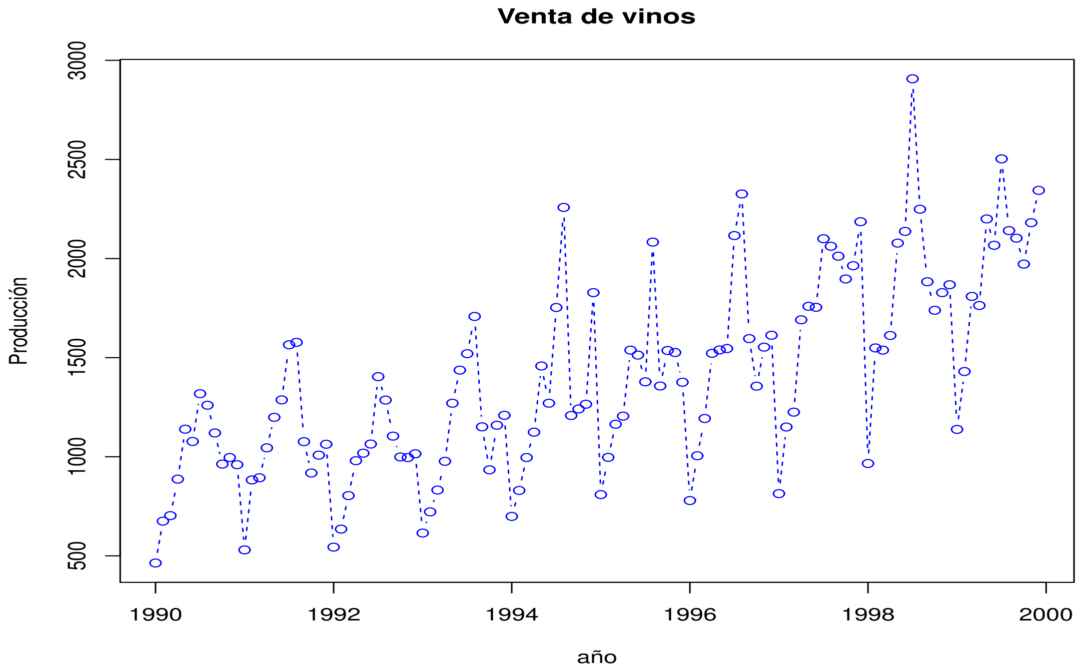
Ahora quitaremos la componente estaconal con el comando diff conocido como operador de diferenciación, en donde lag es el exponente o mejor dicho el factor de retraso.

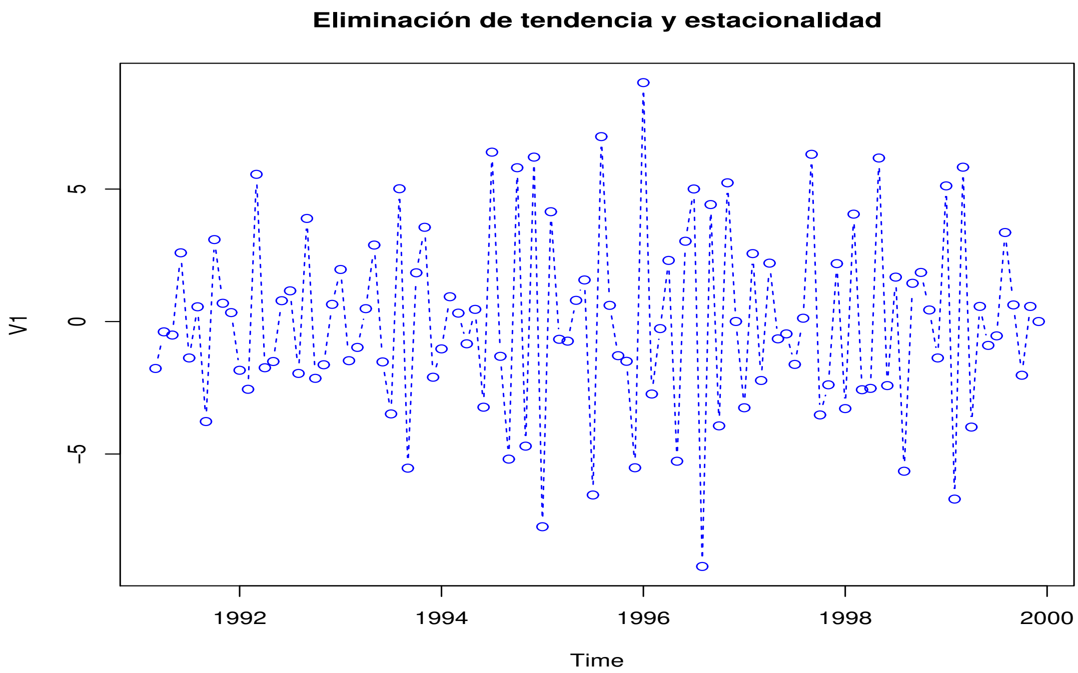
```
transbcs<-diff(transbc,lag=12)
```

Para retirar la tendencia el proceso es análogo, sólo es necesario agregar el comando differences, en donde indicamos el número entero que indica el orden de la diferencia.

```
transbcst<-diff(transbcs,lag=1,differences=2)
```

Si queremos ver la diferencia entre nuestra nueva serie sin tendencia ni estacionalidad podemos graficar transbcst.





3.5. Pruebas de estacionariedad

De la misma manera haremos las pruebas de estacionalidad, requeriremos que el p valor sea menor a .05 para rechazar la hipótesis de estacionalidad, para ello realizaremos la siguiente prueba:

```
adf.test(transbcst)
pp.test(transbcst)
```

Para ambas pruebas, el p valor es = 0.01 y nos advierte que es menor a ello, por lo que podemos rechazar la hipótesis sin temor a equivocarnos, es decir, hemos retirado la componente estacional con éxito.

3.6. ACF

Para graficar la funcion de Autocovarianza haremos uso del comando

```
autocov<-acvf(transbcst)
```

A partir de ésta función podemos conocer la funcion de autocorrelación que definiremos como

```
autocor<-autocov/autocov[1]
```

Esta funcion nos ayudará a determinar el valor de q para nuestro ajuste, sin embargo, dentro de R el proceso se vuelve engorroso para ver las bandas a partir de las cuales podemos elegir nuestra q adecuada, por lo que haremos uso del software *ITSM* para auxiliarnos a partir de aquí.

4. ITSM

Por medio de *ITSM* complementaremos los pasos que hemos hecho en *R* para solucionar nuestro problema.
Lo primero que debemos hacer es abrir la base de datos (definida como *base* dentro de *R*) en *ITSM*, para ello iremos a file/project/open y abriremos nuestra base original.

4.1. Eliminación de nuestras componentes

Para eliminar nestras componentes de tendencia y estacional, acudiremos al menu Transform/Box-Cox e introduciremos el lambda que obtuvimos dentro de *R*: 0.3434343 .

Ahora, iremos a Transform/Classical, presionamos en el checkbox de Seasonal Fit y colocaremos la frecuencia de nuestra serie, es decir 12, lo que nos arroja nuestra serie como un proceso de ruido blanco que podemos ver a continuación:

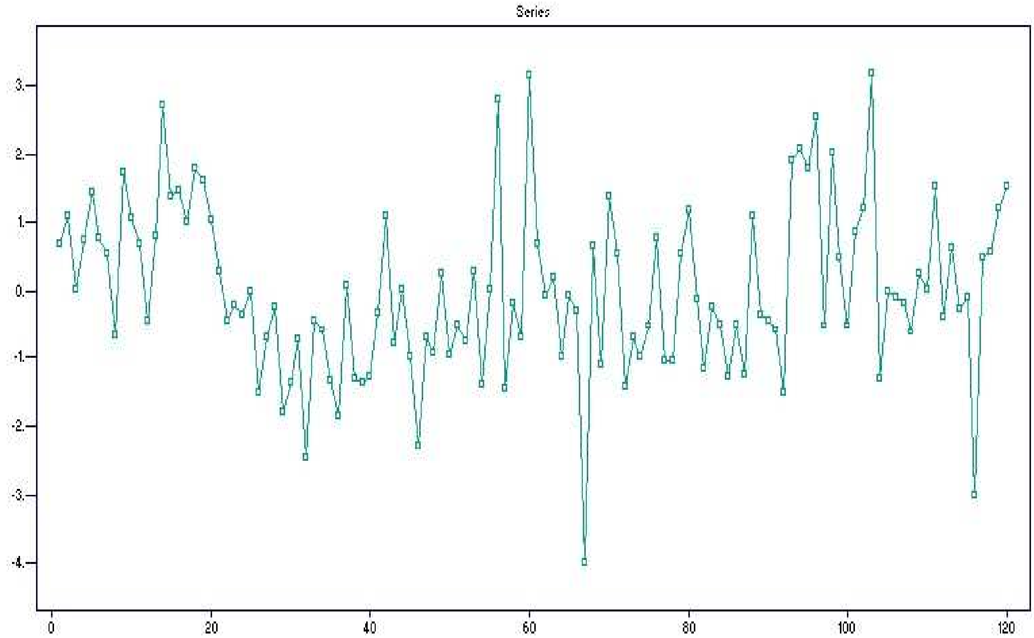


Figura 6: [3.3.1]

4.2. Normalidad del proceso de ruido blanco

Una vez que eliminamos nuestras componentes de tendecia y estacional, nuestra serie ahora debe de ser un proceso de ruido blanco, para corroborarlo veremos si cumple con las pruebas de aleatoriedad. Nuestra hipótesis nula es que nuestra muestra se comporta como una distribución normal.

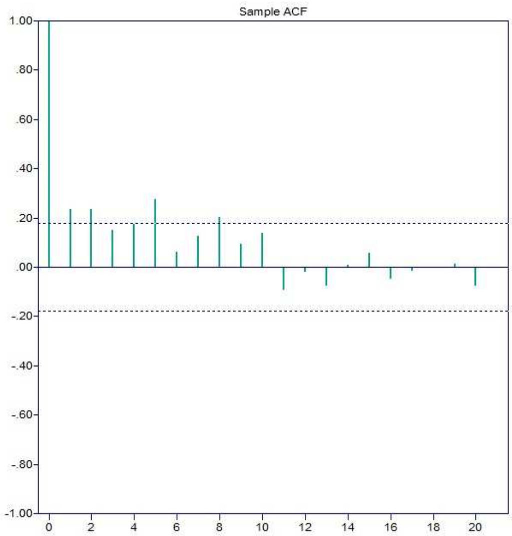
Para ello iremos a statistics/Residual Analysis/Test of Randomness, dejaremos el valor en 20 y presionaremos *OK*. Lo cual nos muestra los siguientes resultados:

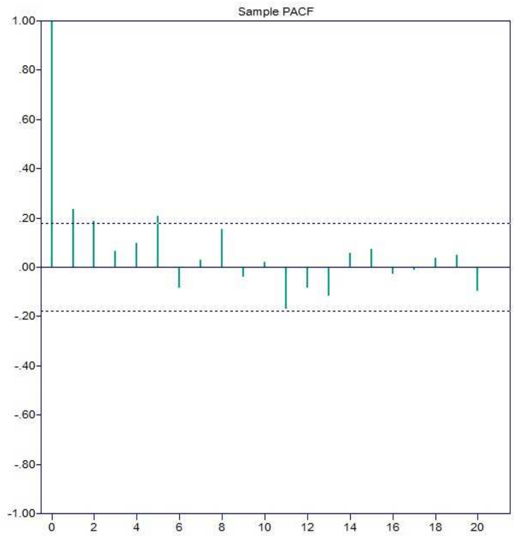
Ljung - Box statistic = 44.094 Chi-Square (12)
p-value = .00001
McLeod - Li statistic = 10.514 Chi-Square (12)
p-value = .57100
Turning points = 80.000 AN(78.667,sd = 4.5838)
p-value = .77114
Diff sign points = 57.000 AN(59.500,sd = 3.1754)
p-value = .43111
Rank test statistic = .35760E+04 AN(.35700E+04,sd = .22044E+03)
p-value = .97829
Jarque-Bera test statistic (for normality) = 1.4947 Chi-Square (2)
p-value = .47363
Order of Min AICC YW Model for Residuals = 5

Dado que el p valor en las pruebas de normalidad es > 0.05 (Omitiendo el Ljung-Test) entonces no rechazamos la hipótesis y podemos corroborar la normalidad y concluir que nuestro proceso resultante *transbcst* tiene un comportamiento de tipo normal que podemos tratar como un proceso de ruido blanco.

4.3. ACF Y PACF

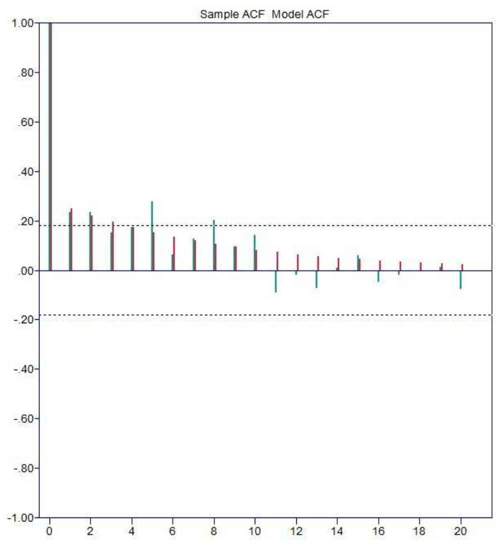
La funcion de autocorrelación y autocorrelacion parcial serán esenciales para determinar los valores p y q de nuestro modelo. La funcion de autocorrelación puede determinar los rezagos apropiados p en un modelo uno ARMA (p,q), en contraste, la función de autocorrelación parcial nos proporciona el valor de q para nuestro modelo. Para visualizar ambas funciones, basta con ir al menú Model/ACF/PACF con h=20, lo que nos arroja el siguiente resultado:

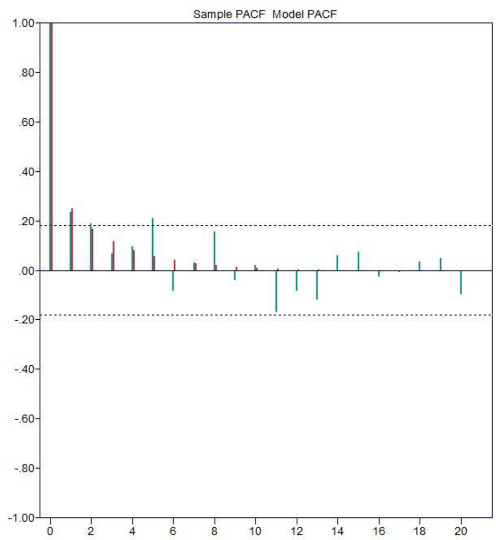




Gracias a la gráfica, podemos decir que nuestro modelo tiene parámetros $p = 8$ y $q = 5$.

Una vez obtenidos los valores de p y q, haremos la estimación de nuestr modelo, para ello, debemos ir al menú Estimation/AutoFit y colocar el orden de nuestro modelo, es decir AR (max) order = 8 y MA (max) order = 5. Ahora solo resta comparar nuestro modelo con un ARMA(8,5).

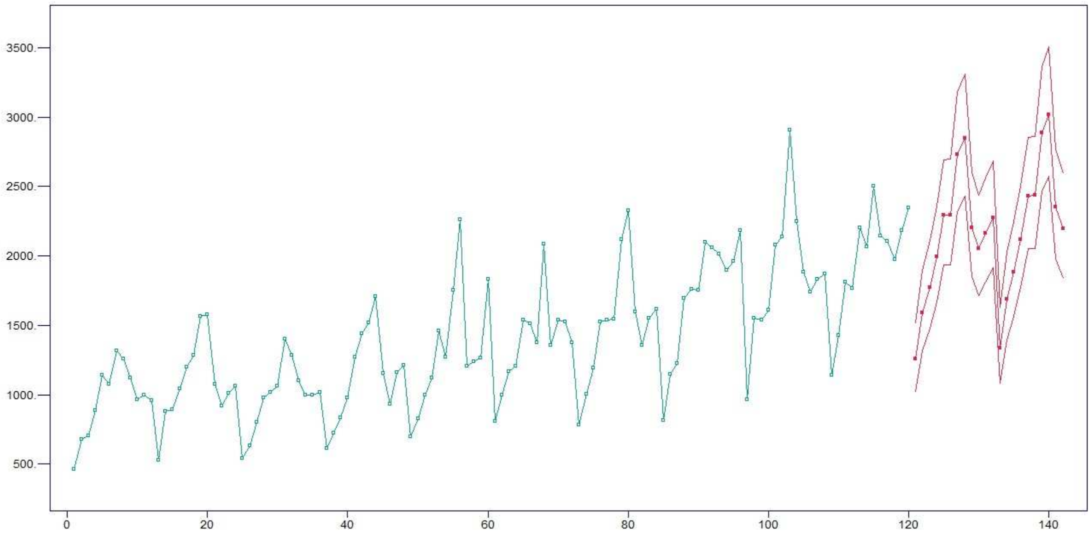




En el resultado de nuestra ACF hay una similitud pesente, no obstante, dentro de nuestra PACF hay pequeños valores que difieren, sin embargo, conforme más avanzamos éstos siguen disminuyendo.

Predicción

Ahora, es momento de predecir los 22 meses a futuro de nuestra serie, para ello, debemos ir al menú Forecasting/ARMA, indicar que queremos 22 predicciones a futuro y que grafique los intervalos de confianza, una vez hecho esto obtenemos:



Y más importante aún, la información de la gráfica (presionando click derecho/info):

Step	Prediction	Lower	Upper
1	.12560E+04	.10260E+04	.15176E+04
2	.15904E+04	.13169E+04	.18989E+04
3	.17754E+04	.14777E+04	.21101E+04
4	.19940E+04	.16694E+04	.23574E+04
5	.22936E+04	.19346E+04	.26938E+04
6	.22959E+04	.19350E+04	.26985E+04
7	.27276E+04	.23207E+04	.31788E+04
8	.28479E+04	.24277E+04	.33132E+04
9	.22056E+04	.18512E+04	.26018E+04
10	.20546E+04	.17162E+04	.24340E+04
11	.21650E+04	.18140E+04	.25580E+04
12	.22765E+04	.19130E+04	.26828E+04
13	.13372E+04	.10835E+04	.16271E+04
14	.16892E+04	.13917E+04	.20257E+04
15	.18855E+04	.15648E+04	.22465E+04
16	.21163E+04	.17694E+04	.25052E+04
17	.24313E+04	.20502E+04	.28563E+04
18	.24369E+04	.20551E+04	.28626E+04
19	.28885E+04	.24602E+04	.33632E+04
20	.30163E+04	.25752E+04	.35043E+04
21	.23505E+04	.19776E+04	.27667E+04
22	.21948E+04	.18388E+04	.25932E+04

5. Toma de decisiones y programación lineal

5.1. Planteamiento del problema

Ahora que hemos hecho la predicción de nuestras observaciones 22 meses a futuro, podremos indicar el numero óptimo de botellas a producir a lo largo de dichos periodos. No olvidemos que contamos con una cuenta de \$100,000 en el banco con un 4 % de interés, la pregunta a plantearnos ahora es: ¿Invertiremos en el banco ó en el viñedo?.
Una botella tiene un costo de \$25 en el mercado y su costo de producción es de \$10, significa que cada botella vendida representa una ganancia de \$15, es decir, nuestra ganancia es del 150 %.
Una botella que fue almacenada un mes, al haber un costo de almacenamiento de \$10, representa una ganancia es de \$5, es decir, nuestro rendimiento es del 50 %.
Si una botella no se vende después de haber sido alacenada la regalaremos, es decir, tendrems pérdidas del 100 %.
Eso quiere decir que siempre invertiremos en el viñedo procurando que ninguna botella esté más de un mes almacenada.

Aunado a ello, significa que siempre que nuestras ventas sean menores a 2300 unidades, decidiremos vender y no almacenar el periodo anterior. Si nuestras ventas son mayores a 2300 durante un segundo mes y mayores a 800 el mes anterior pero menores a 2300, el primer mes decidiremos producir las ventas mensuales más el excedente del segundo mes y ese excedente almacenarlo.

5.2. Maximización de ganancias

Tomemos como ejemplo los periodos 114 y 115, que tuvieron ventas de 2067 y 2503 unidades respectivamente, para el planteamiento de nuestro problema, haremos uso de los datos de la sección 2. Primero definamos nuestras variable a partir de la información que tenemos, es decir:

x_1 =Venta de vinos durante este mes
x_2 =Venta de vinos almacenados durante este mes
x_3 =Venta de vinos durante el siguiente mes
x_4 =Venta de vinos almacenados durante el siguiente mes

Cabe destacar que sólo tomamos en cuenta éste mes y el siguiente, pues el almacenamiento no tiene sentido si excedemos del mes, pues nos generarían pérdidas, pues si almacenamos dos meses obtendíamos -\$5 por cada venta. -\$10 de producción, -\$20 de almacenamoento y +\$25 de venta.
Nuestro objetivo en programación lineal para este problema será maximizar nuestras ganancias, es decir nuestra función de utilidad U.
Donde si definimos $15x_1$ como la ganancia por la venta de vinos durante este mes y análogamente para cada variable, entonces nuestra función de utilidad se define como:

$$U = 15x_1 + 5x_2 + 15x_3 + 5x_4$$

Que para los periodos 114 y 115 sólo debemos sustituir:
$x_1 = 2067$
$x_2 = 0$ (pues este mes nuestras ventas fueron menores a 2300)
$x_3 = 2300$(el máximo de nuestra producción)
$x_4 = 203$ (que almacenamos el mes anterior para venderlos este)

Por lo que nuestra utilidad fue de:

$$U = 15(2067) + 5(0) + 15(2300) + 5(203)$$

$$U = 66520$$

Debemos tener cuidado, pues U representa la ganancia de los meses 114 y 115, si queremos la de un mes en concreto, debemos entonces restar lo que corresponde al otro mes.

5.3. Restricciones

Primero debemos definir entre que valores se encuentran nuestras variables x
$0 \leqslant x_1 \leqslant 2300$
$0 \leqslant x_2 \leqslant 1500$
$0 \leqslant x_3 \leqslant 2300$
$0 \leqslant x_4 \leqslant 1500$

Ahora es cuando tomaremos en cuenta nuestros intervalos de confianza: para el periodo 120 nuestras ventas fueron de 2344 y para el 121 diremos que se encuentran entre 1026 y 1517, entonces:
$x_1 + x_2 \geqslant 2344$
$x_1 + x_2 \leqslant 2344$
$x_3 + x_4 \geqslant 1026$
$x_3 + x_4 \leqslant 1517$

Que se puede escribir como:
$-x_1 - x_2 \leqslant -2344$
$x_1 + x_2 \leqslant 2344$
$-x_3 - x_4 \leqslant -1026$
$x_3 + x_4 \leqslant 1517$

Y ya tenemos nuestro planteamiento como ejercicio de programación lineal, ahora haremos uso de *MatLab* para darle una solución rápida.

6. Programación lineal en MatLab

Dentro de *MatLab* existen funciones que nos ayudarán a resolver nuestro problema de programación lineal.
Lo primero que debemos hacer es ingresar los datos dentro del programa como se muestra a continuación:

```
A=[-1 -1 0 0; 1 1 0 0;0 0 -1 -1;0 0 1 1 ];
```

A es nuestra matriz de coeficientes de cada x.

```
b=[-2344 2344 -1026 1517];
```

b es la matriz de coeficientes de de las retricciones.

```
f=[-15 -5 -15 -5];
```

f es la matriz de coeficientes de U, los valores los colocamos negativos pues tratamos un problema de maximización.

```
Aeq=[];
beq=[];
```

Aeq y beq se definen si son igualdades, y al estar tratando desigualdades lo dejaremos vacío.

```
lb=[0 0 0 0];
```

lb es la cota inferior del valor de las x.

```
ub=[2300 1500 2300 1500];
```

lb es la cota superior del valor de las x.
Finalmente, para encontrar el valor óptimo de cada x, utilizaremos el comando *linprog*

```
x = linprog(f,A,b,Aeq,beq,lb,ub)
```

Lo que nos arroja el siguiente resultado:

$$
\begin{aligned}
x_1 &= \quad 2300 \\
x_2 &= \quad 0044 \\
x_3 &= \quad 1517 \\
x_4 &= \quad 0000
\end{aligned}
$$

Que al sustituir en $U = 15x_1 + 5x_2 + 15x_3 + 5x_4$ obtendremos nuestra ganancia por el mes 120 y 121

$$U = 15(2300) + 5(44) + 15(1517) + 5(0)$$
$$U = 57475$$

Si deseamos hacer lo mismo para los demás periodos, obtendremos siempre el valor de los intervalos de confianza superiores, Para cambiar el periodo, solo basta con cambiar el valor de b:

```
b=[-(ICL de x1) (ICU de x1) -(ICL de x2) (ICU de x2)];
```

En donde ICL es el intervalo de confianza inferior y ICU el intervalo de confianza superior.

7. Conclusiones

La opción más loable se creería que son los valores de la predicción redondeados a números enteros y no los de ICU, sin embargo analizemos los dos casos posibles.

El primero es si hay exceso de demanda, de suceder eso, estaríamos perdiendo, pues lo invertido en el banco lo pudimos invertir en producir más botellas.

El segundo caso es si la demanda no es suficiente y nos vemos obligados a almacenar, en tal caso, sólo ganaríamos $5 por unidad (aún así más que en el banco) y produciríamos el valor predicho menos el excedente.

Supongamos el peor de los casos, en el que producimos el máximo y se vende el mínimo para 121 y 122.

Entonces $x_1 = 1026, x_2 = 491, x_3 = 1407, x_4 = 581$ aún si esto se repite todos los periodos, dado que ningun ICU ni ICL sperarán los 1500 (se hacen las restas entre el ICU e ICL y todas son $\leq$ a 1500), significa que siempre ganamos más que si lo invertimos en el banco. En el mejor de los casos ganaríamos aún más.

Por ello siempre conviene producir el máximo, pues en el peor de los casos gano más que el banco debido ya que no hay una diferencia mayor a 1500 entre los intervalos de confianza superiores e inferiores en ningún periodo.

Por lo tanto, mi producción será:

Producción
.15176E+04
.18989E+04
.21101E+04
.23574E+04
.26938E+04
.26985E+04
.31788E+04
.33132E+04
.26018E+04
.24340E+04
.25580E+04
.26828E+04
.16271E+04
.20257E+04
.22465E+04
.25052E+04
.28563E+04
.28626E+04
.33632E+04
.35043E+04
.27667E+04
.25932E+04

A. Código:

```
##Descargar install.packages("MASS")
##Haremos uso de R y la librería de MASS para obtener más información
##(MASS es para box cox)
##install.packages("MASS")
library(MASS)
##install.packages("astsa")
library(astsa)
##para BoxCox
##install.packages("forecast")
library(forecast)
library(itsmr)
library(tseries)
##Colocaremos nuestra serie dentro de R
##base<-read.table("ubicación del archivo") escogemos el .tsm
base<-read.table(choose.files())
##te muestra lo que hay en la base
ls()
##convertiremos nuestras observaciones en una serie de tiempo con el comando "ts" (t
##(120 meses son 10 años, freq= numero de observaciones cada año)
serie<-ts(base,start=c(1990,1),end=c(1999,12), freq=12)
##veamos nuestra serie

plot(serie)
win.graph()
ts.plot(base)

##dev.new() es igual a win.graph

##vamos a quedarnos con la sig grafica

serie<-ts(base,start=c(1990,1),end=c(1999,12), freq=12)
##tipo de linea y color, lwd=2 es el grueso, type=l

plot(serie,lty=2,col=12,xlab="año",ylab="Producción",main="Venta de vinos",type="b")

##prueba de boxcox y grafica el valor de lambda max verosimil. (~ se llama cilo omg)
boxcox(serie~1)
## (buscar interpretación)
##hallar el lambda max verosimil

lamb<-boxcox(serie~1)$x[which.max(boxcox(serie~1)$y)]

##mandamos llamar la funcion para ver la aplicacion de boxcox

transbc<-BoxCox(serie,lamb)
```

```
stl(transbc,s.window="periodic")
##es posible que nos arroje un error: "Error in stl() : only univariate series are a

dummyVector = as.vector(t(transbc))
tsData = ts(dummyVector, start=c(1990,1),end=c(1999,12), frequency = 12)
stl = stl(tsData, "periodic")
plot(stl)

##quitar la componente estaconal (diff es el dorito lag es el exponente (factor de r

transbcs<-diff(transbc,lag=12)

##quitar la tendencia

transbcst<-diff(transbcs,lag=1,differences=2)

win.graph()
plot(transbcst,lty=2,col=12,main="Eliminación de tendencia y estacionalidad",type="b

win.graph()
plot(serie,lty=2,col=12,xlab="año",ylab="Producción",main="Venta de vinos",type="b")
####el getwd() lo pongo en la carpeta donde guarde las series de tiempo

adf.test(transbcst)
pp.test(transbcst)

autocov<-acvf(transbcst)
plot(0:40,autocov,type="h")

autocor<-autocov/autocov[1]
plot(0:40,autocor,type="h")
```

Referencias

[1] http://www.vinos-fr.com/provence/bunan/

[2] Box, G. E., Jenkins, G. M., & Reinsel, G. C. (2013). Time series analysis: forecasting and control. John Wiley & Sons.

[3] https://www.mathworks.com/help/optim/ug/linprog.html