



Veri Biliminde Konular: Zaman Serilerinde Anomali ve Öngörü

Nurhak ÖZAT

May 19, 2018

Tez Danışmanı: Doç. Dr. Atabey KAYGUN

Contents

1 Zaman Serisi ve Temel Kavramlar	4
2 Zaman Serisi Bileşenleri :	5
2.1 Trend :	5
2.2 Mevsimsel Bileşen:	5
2.3 Çevrimsel Bileşen:	5
2.4 Düzensiz Bileşen (Rassal Bileşen)	6
3 Geleneksel Zaman Serisi Ayırışım Yöntemleri	6
4 Doğrusal Zaman Serisi Analizi:	6
4.1 1- Otoregresif Süreç (AutoRegressive Process)	6
4.2 2- Hareketli Ortalama Süreci (Moving Average)	7
4.3 3- Karma Otoregresif Hareketli Ortalama Süreci (ARMA)	7
4.4 4- Homojen Durağan Olmayan Süreç (ARIMA)	7
5 Box-Jenkins Yöntemi (ARIMA)	9
6 Durağanlık	9
6.1 Trend Durağanlık ve Fark Durağanlık	10
6.2 Durağanlık Analizi	11
6.2.1 Birim Kök Analizi	11
6.2.2 Dickey Fuller Testi (DF)	11
6.2.3 Genişletilmiş Dickey-Fuller Testi (ADF)	11
6.2.4 Phillips- Perron Testi (PP)	12
6.2.5 Kwiatkowski-Phillips-Schmidt-Shin Testi (Kpss)	12
6.2.6 Ng Perron Testi	12
7 Normallik Testi- Shapiro Wilk Test	12
8 Otokorelasyon	13
8.1 Grafik Yorumu	13
8.1.1 Otokorelasyon Katsayısı (ACF)	13
8.1.2 Kısmi Otokorelasyon Fonksiyonu (PACF)	14
8.2 Otokorelasyon Testleri	14
8.2.1 Durbin- Watson d Testi	14
8.2.2 Wellis Testi:	14
8.2.3 Breush ve Godfrey Testi:	15
9 Giriş	16
10 Natural Gas Futures	16
10.1 Decompose - Geleneksel Zaman Serisi Ayırışım Yöntemleri	18
10.2 Normallik Testi- Shapiro Wilk Test	23
10.3 Otokorelasyon	24
10.4 Durağanlık Analizi	29
10.5 Arima	32

10.6	Zaman Serilerinin Trend Bileşeninden Arındırılması (de-trend)	34
10.7	Zaman Serilerinin Mevsimsel Bileşeninden Arındırılması (de-seasonalize)	36
10.8	Natural Gas Futures Zaman Serisi Analizi Özet	41
10.9	Forecast	42
10.9.1	Üstel Düzleştirme -Exponential Smooting	42
10.9.2	Öngörü Doğruluğunu Hesaplama - Accuracy Fonksiyonu	43
10.9.3	Basit Üstel Düzleştirme Yöntemi (Durağan Seriler için):	47
10.9.4	Holt ÜsteL Düzleştirme Yöntemi: (Doğrusal Trendi Olan Seriler için)	49
10.9.5	Holt-Winters Yöntemi: (Doğrusal Trendi ve Mevsimlik Etkileri Olan Seriler İçin)	51
10.9.6	Arima Modeli	54
10.9.7	Natural Gas Futures - Zaman Serilerinde Öngörü Özet	58
11	Florida Field Production of Crude Oil	58
11.1	Decompose- Geleneksel Zaman Serisi Ayrışım Yöntemleri	59
11.2	Normallik Testi - Shapiro Wilk Test	65
11.3	Otokorelasyon	66
11.4	Durağanlık Analizi	71
11.5	Arima	74
11.6	Zaman Serilerinin Mevsimsellikten Arındırılması (de-seasonalize)	78
11.7	Florida Field Production of Crude Oil- Zaman Serisi Analizi Özet	83
11.8	Forecast	84
11.8.1	Üstel Düzleştirme Yöntemi- Exponential Smooting	85
11.8.2	Basit Üstel Düzleştirme Yöntemi (Durağan Seriler için)	87
11.8.3	Holt Üstel Düzleştirme Yöntemi D: (Doğrusal Trendi Olan Seriler için)	89
11.8.4	Holt-Winters Yöntemi: (Doğrusal Trendi ve Mevsimlik Etkileri Olan Seriler için)	91
11.8.5	Florida Field Production of Crude Oil - Zaman Serilerinde Öngörü Özet	95
12	Kaynakça	96

```
In [1]: # install.packages("Quandl", repos='http://cran.us.r-project.org')
#install.packages("forecast", repos='http://cran.us.r-project.org')
library(Quandl)
library(forecast)
library(tseries)
```

Warning message:

"package 'Quandl' was built under R version 3.4.3"Loading required package: xts
Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

as.Date, as.Date.numeric

Warning message:

"package 'forecast' was built under R version 3.4.3"Warning message:
"package 'tseries' was built under R version 3.4.3"

1 Zaman Serisi ve Temel Kavramlar

Zaman serisi, bir değişkenin zaman içindeki hareketini gözlemleyen, gözlem sonuçlarının zamana göre dağılım gösterdiği serilerdir. Bütün değerleri belirli bir zaman farkı ile art arda gelen nümerik verilerden oluşur. Bu değerler günlük, haftalık, aylık, yıllık olabilirler.

Zaman serisi değişkeni, n zaman serisindeki veri sayısı olmak üzere; $x(t)$, ($t= 1, 2, \dots, n$) - ilk gözlemlenen veri $x(1)$, - ikinci gözlemlenen veri $x(2)$, - son gözlemlenen veri $x(n)$ ile ifade edilir.

Sürekli Zaman Serileri; zaman içinde sürekli olarak kaydedilebilen verilere sahip serilerdir elektrik sinyalleri, voltaj, ses titreşimleri gibi mühendislik alanlarına ait serilerdir.

Kesikli Zaman Serileri; belli aralıklarda elde edilebilen verilere sahip serilerdir. faiz oranı, satış hacmi, üretim miktarı gibi iktisadi serilerdir.

Zaman serilerinde veriler nicel ya da nitel verilerden oluşur.

- Nitel (Kalitatif) Yöntemler : Öngörü işlemi ile bilgilerden yararlanılarak geleceğe ait tahmin yapılmaktadır. Dolayısıyla, öngörü yöntemleri tecrübeye, kararlara, bilirkişinin düşüncelerine dayanmaktadır. Bu tür öngörü yöntemlerine genel olarak nitel (kalitatif) yöntemler denir.
- Nicel (Kantitatif) Yöntemler : Öngörü yöntemleri subjectif kararlardan ziyade elde edilen verilerin yapısını açıklayabilen modellere dayanırsa bu tür modellere dayanan öngörü yöntemlerine genel olarak nicel (Kantitatif) yöntemler denir.

Zaman Serisi analizinde,iki çeşit değişken sayısı bulunur.

- Tek Değişkenli (Univariate Time Series) : Eğer zaman serisi tek değişkenin zaman içerisindeki hareketi incelenir. ,
- Çok Değişkenli (Multivariate Time Series) : Eğer birden fazla değişkenin zaman içindeki değişimlerini incelendiğinde kullanılır.

2 Zaman Serisi Bileşenleri :

Zaman serilerinin gözlemlenen değerleri arasında artma, azalma ya da belli dönemlerde sabit kalma gibi değişimler gerçekleşebilir. Bu değişimler çeşitli nedenleri olabilir. Bu değişimlerin incelenmesi , bir öngörü yapılmak istendiğinde yardımcı olur. Nedeni ise zaman serileri gelecekte de benzer özellikler gösterebilir. Bu değişimler zaman serilerinin bileşenlerini oluşturur. Bu değişimler dört grupta incelenir:

- Trend bileşen
- Mevsimsel bileşen
- Çevrimsel bileşen (Konjoktürel)
- Düzensiz Bileşen (Rassal Bileşen)

2.1 Trend :

Zamana göre gözlemlenen bir değişkenin uzun dönemde gösterdiği artış veya azalışa denir. İki farklı şekilde ifade edilebilir.Trend iki şekilde olabilir:

- Doğrusal Trend
- Doğrusal Olmayan Trend

Gözlemlerin toplanış sırası ile aldığı değerler arasındaki korelasyonun önemini belirlemek için trend analizleri yapılır. Trend analizi matematiksel bir tekniktir ve bu analiz ile geçmişe ait sonuçları kullanarak geleceğe yönelik sonuçlar için tahmin yapılır. Serinin gözlem sayısı arttıkça trende sahip olmadığı daha kolay anlaşılır. Zaman içerisinde artış veya azalış göstermeyen, aynı düzeyde kararlılık gösteren serilerin ise trendi yoktur.

Trend doğrusal olabileceği gibi üstel de olabilir. Trend tamamen öngörülebiliyorsa deterministik, tamamen öngörüleliyorsa rassaldır.

2.2 Mevsimsel Bileşen:

Sistematiik (gözlemlenebilir ve ölçülebilir) etkileri olan faktörlerdir. Mevsim etkisinde olan zaman serileri yılın bazı dönemlerinde diğer dönemlere göre daha yüksek ve ya daha düşük değerler ulaşır. Örnek olarak; bazı dönemlerde doğalgaz kullanımının artması veya azalması verilebilir.Mevsimsellik farklı biçimlerde (Anneler Günü, Babalar Günü, Öğretmenler günü , Dini bayramlar da yapılan alışverişteki artışlar)gözlemlenebilir. Zaman serilerinde mevsimsellik etkisi bir yıl ve daha az zaman süresinde tekrarlanan periyodik dalgalanmalar olarak tanımlanır.

Zaman serisi analizlerinde mevsimsel etkinin olup olmadığını belirlemek için bir çok yöntem kullanılır. Zaman serisinde mevsimsel bileşen varsa , veriden mevsimsel dalgalanmaları arındırılması için en çok kullanılan yöntemler mutlak ve nispi hareketli ortalamalar ve ortalama yüzde yöntemleridir.

2.3 Çevrimsel Bileşen:

Mevsimsel değişimle ilişkisi olmayan daha uzun zaman aralığındaki dönemsel değişimdir.Trend doğrusu veya eğrisinin etrafındaki uzun süreli dalgalanmalardır. Örnek olarak; Ekonominin maksimum olduğu noktada oluşan bir kriz ile birlikte ekonominin düşüşe başlamasını , devamında belli bir süre hareketsizlik gözlenmesi ,ardından yeniden kıvılcımlarla canlanmaya başlamasını ve bu sürecin tekrarlanıp devam etmesini verebiliriz.

2.4 Düzensiz Bileşen (Rassal Bileşen)

Rassal veya geçici olarak ortaya çıkan , trend , mevsimsel ve çevrimsel bileşenlerin dışında kalan bileşenlere veya belirli bir modele uymayan dalgalanmalara düzensiz bileşen denir. Değişimin zamanı ve şiddeti tahmin edilemez. Serideki yanıtıcı hareketlerdir.

3 Geleneksel Zaman Serisi Ayırışım Yöntemleri

Serinin trend konjonktürel ve mevsimsel etkisi altında kaldığını varsayılmaktadır. Toplamsal Ayırıştırma (Additive)ve Çarpımsal Araştırma (Multiplicative) yöntemleri olmak üzere iki yöntemi vardır.

Toplamsal Ayırıştırma (Additive) :

$$\hat{Y}_{d_t} = T_t + S_t + C_t + I_t$$

Çarpımsal Araştırma (Multiplicative):

$$\hat{Y}_{d_t} = T_t \cdot S_t \cdot C_t \cdot I_t$$

4 Doğrusal Zaman Serisi Analizi:

4.1 1- Otoregresif Süreç (AutoRegressive Process)

Otoregresif süreç AR (p) ile ifade edilir ve serinin mevcut değeri önceki p adet değere bağlıdır.

p'ninci dereceden otoregresif süreç: Y_t değeri sadece bir önceki değerine değil önceki bir çok değere bağlı olabilir.

$$\hat{Y}_t = \epsilon_t + a_0 + \sum_{i=1}^{\ell} a_i \cdot y_{t-i}$$

Başka bir şekilde ifade edilirse, bağımlı değişken geçmişteki değerinin bir fonksiyonudur.(p otoregresif sürecin derecesini vermektedir.)

a : sabit terim

$a_1 \dots$: Gecikmeli değerlerin şimdiki değerle olan ilişkisi

ϵ : hata terimi – rassal şoklar

Otoregresif süreç durağan olma koşulu

$$|a_n| \leq 1 \text{ ve } \sum_{i=1}^{n-1} a_i \leq 1$$

olmasına bağlıdır.

Otoregresif modeli özetlemek gerekirse:

Tüm gecikmeli değerler Y_t üzerinde birikimli bir etkiye sahiptir. Dolayısıyla otoregresif modellerin uzun süreli bir etkiye sahiptir. Rassal değişken ile y_t arasındaki korelasyon sıfır olduğundan otoregresif modelleri En küçük kareler yöntemini kullanarak tahmin edilebilir. Otoregresif süreçlerde ACF gecikmeden sonra üstel olarak azalmakta ve PACF değeri de sıfır değerini almaktadır.

4.2 2- Hareketli Ortalama Süreci (Moving Average)

Serinin gecikmeli hata terimi, şu andaki hata terimini etkilemesine hareketli ortalama süreci denir. Değişkenin tahmin değeri hata teriminin tahmin değeri ile ilişkilidir. Modelin derecesi geçmiş dönemdeki hata terim sayısına göre belirlenir. Örneğin geçmiş dönemde bir hata varsa modelin derecesi 1, n tane hata varsa modelin derecesi n olmaktadır. Hareketli ortalama sürece MA (q) olarak ifade edilir ve $Y_t \sim N(0, s_e^2)$ olacak şekilde

$$\hat{Y}_{d_t} = b_0 + \sum_{i=1}^q b_i e_{t-i}$$

kurulur.

b_i : bilinmeyen sabit parametrelerdir.

4.3 3- Karma Otoregresif Hareketli Ortalama Süreci (ARMA)

Zaman serilerinde veriler bazen hem otokorelasyon hem de kısmi otokorelasyon fonksiyonlarını belirli bir gecikmede kesilmeden sıfıra doğru çok yavaş hareket edebilir. Bu durumda seri hem otoregresif hem de hareketli ortalama bileşenlerini aynı anda içerebilir. Özetle seri modelini hem AR, hem de MA bileşenleriyle p ve q dereceden olmak üzere ARMA(p,q) olarak tanımlanır

Arma sürecinin özetlemek gerekirse:

Karma otoregresif hareketli ortalama sürecinin ARMA (p,q) otokorelasyon fonksiyonu ve otoregresif sürecin AR(p) sürecinin özelliklerini taşır.

ARMA(p,q) modelinde büyük gecikme uzunluklarında ACF ve PACF değerleri sıfıra yaklaşır. Model de durağanlığı AR(p) koşulu ile , tersine çevirebilirliği ise MA(q) koşuluyla aynıdır.

4.4 4- Homojen Durağan Olmayan Süreç (ARIMA)

Fark alma işlemiyle, durağan olmayan modellerin durağan hale dönüştürülmüş serilere uygulanmasıyla elde edilen modellere “durağan olmayan doğrusal stokastik modeller” denir. Bu tür serilere uygulanan model ARIMA modelidir. Bu modeller d dereceden farkı alınmış serilere uygulanan, değişkenin t-dönemindeki değerinin belirli sayıdaki geri dönem değerleri ile aynı dönemdeki hata teriminin doğrusal bir fonksiyonu olarak ifade edildiği AR ve değişkenin t-dönemindeki değerinin aynı dönemdeki hata terimi ve belirli sayıda geri dönem hata terimlerinin doğrusal fonksiyonu olarak ifade edildiği MA modellerinin birer birleşimidir. Farklı bir şekilde ifade edilirse ARIMA, otoregresif (AR), entegre (I) ve hareketli ortalama (MA) süreçlerinden oluşur. Modelin gösterimi ARIMA (p, d, q) şeklinde ifade edilir. Burada p değişkeni otoregresif (AR) model derecesini, q değişkeni hareketli ortalama (MA) derecesini ve d değişkeni de fark alma derecesini gösterir.

Eğer d=1 ise orijinal seri birinci derecen farkı alınarak durağan hale getirilmiştir, d=0 olduğunda ise orijinal seri zaten durağandır ve farkının alınmasına da gerek yoktur. Bu durumda ARIMA(p,d=0,q)=ARMA(p,q) olur.

Örnek olarak ARIMA(3,2,2) alınırsa; zaman serisini durağan olması için farkının iki kez (d= 2) alınması gerekir. İlk farkı alınan zaman serisini ARMA(2,2) süreci olarak yazılabilir. Bu süreçte otoregresif modeli (AR) iki terimiyle ve hareketli ortalama modeli (MA) iki terimiyle modellenebilir.

Bilgi Kriterleri :

1. Akaiki bilgi kriteri (Akaike information criterion-AIC)

Akaiki, veri kümeleri için istatistiksel göreceli model ölçüsüdür. AIC modeli verilerin mmodelerin her birini göreceli olarak tahmin ediyor. Doğru modelin seçilmesini sağlar. Verilen bilgileri kullanara; model verileri oluşturur, işlem temsil etmek için kullanılır ve bunlara dayanarak göreceli bir tahmin sunmaktadır. Bu sayede ,modelin uyum ölçüsü ve model karmaşıklığı anlaşılır.

$$AIC = 2k - 2\ln(L)$$

k: tahmin edilen parametre sayısı

L: maksimum likelihood

Model seçiminde en düşük AIC değeri seçilir.

2. Schwarz Bayesian kriteri (BIC)

Schwarz Bayesian bilgi kriteri (BIC) değeri, otoregresif sürecin gecikme derecesinin belirlenmesinde kullanılır.Akaiki bilgi kriteri ile yakından ilişkilidir.

$$BIC = \ln(n) \cdot k - 2\ln(L)$$

L = modelin olabilirlik fonksiyonunun maksimum değeri

x = gözlenen veriler

n = veri noktalarının sayısı

k = model tarafından tahmin edilen parametre sayısı

Model seçiminde en düşük BIC değeri seçilir.

3. Hannan Quinn bilgi kriteri (HOİC)

$$HOC = -2L_{max} + 2k \cdot \ln(\ln(n))$$

Bilgi kriterleri iki bileşeni vardır:

- Kalanların karelerinin toplamı
- Eklenen her parametrenin neden olduğu serbestik derecesi kaybı için modelin tabi tutulduğu ceza

bileşenleridir. Modele eklenen her yeni değişken, yani her yeni gecikme, ceza teriminin değerini arttırır. Aynı zamanda kalanların karelerinin değerini düşürmektedir. Akaiki bilgi kriteri (AIC) değeri, otoregresif sürecin gecikme derecesinin belirlenmesinde kullanılır.Schwarz Bayesian bilgi kriteri (BIC) değeri, otoregresif sürecin gecikme derecesinin belirlenmesinde kullanılır.

BIC , AIC ile karşılaştırıldığında edildiğinde örnek büyüklüğü orta veya geniş ise gecikme sayısı az olan AR modelin seçimine eğilim söz konusudur.

5 Box-Jenkins Yöntemi (ARIMA)

George E.P. BOX ve Gwilym M. JENKINS tarafından durağan tek değişkenli zaman serilerinin analizi için geliştirilen (Yaffee and McGee 2000) ve öngörü uygulamalarında kullanılan Box-Jenkins yöntemi, ARIMA modelleri olarak da adlandırılmaktadır. Box-Jenkins yöntemi ele alınan zaman serisinin özelliklerine göre belirlenen çeşitli modeller arasında uygun olanını seçerek tahmin etme sürecini kapsamaktadır. Zaman serisi modellerinde serinin kendi iç dinamiği önemli olup oluşturulan modeldeki değişken, kendi gecikmeli değerleri ve hata teriminin gecikmeli değerleriyle açıklanmaktadır. Box ve Jenkins uygulama aşamaları:

- Yöntemi Belirleme :

Durağan zaman serisine ARMA modellerinden herhangi birinin aday olarak belirlenmesi aşamasıdır. Yani ARMA (p,q) modelinde p ve q nun derecesinin bulunması aşamasıdır. Değerleri otoregresif ve hareketli ortalamalar sürecinin özelliklerinden faydalanılarak bulunur. ARMA modelleri otokorelasyon ve kısmi otokorelasyon fonksiyonları sayesinde karakterize edilebilir.

- Parametre Tahmini :

ARMA modellerinde parametrelerini tahmin edilebilmesi için bir çok yöntem bulunmaktadır. Yöntemler olabilirlik fonksiyonuna dayalı yöntemler ya da eğrisel en küçük kareler yöntemleridir. ARMA modelinin parametreleri olabilirlik fonksiyonunun maksimize edilebilir ya da hata kareler fonksiyonunun minimize edilmesiyle bulunabilir.

- Teşhis Etme :

Modelin yeterliliğini kontrol edilmesidir. Zaman serisi modeli belirlenip parametreleri tahmin edildikten sonra ayırt edici testler yardımıyla belirlenen modelin ne kadar doğru olduğu ortaya konulabilir. Modelin artık terimleri, test süreci için önemlidir. Belirlenen model yeterli ise artıklar yaklaşık olarak bir beyaz gürültü sürecidir (white noise). Beyaz gürültü sürecinde olması, ARIMA (p,d,q) modeli öngörü için yeterli düzeydedir.

LjungBox testi, Modelin artık terimlerinin standart normal dağılımlı rasgele değişkenlerin dizisinden oluşan bir beyaz gürültü sürecinde olup olmadığının test etmek için kullanılmaktadır.

6 Durağanlık

Serinin durağan olması, zaman serisinde ortalaması, varyansı ve kovaryansı zaman içinde sabit kalmasıyla söylenir. Durağan zaman serisinde ard arda gelen iki değer arasındaki fark zamanın kendisinden kaynaklanmamakta, sadece zaman aralığından kaynaklanmaktadır. Bundan dolayı serinin ortalaması zamanla değişmemektedir. Y_t serisi tanımlayalım.

$$E(Y_t) = \mu$$

$$var(Y_t) = \gamma_0$$

$$cov(Y_t, Y_{t+k}) = \gamma_k$$

Başlangıç noktasını t'den t + k'ye kaydığımızı düşünelim. Y durağan ise Y_t ve Y_{t+k} serilerinin ortalama, varyans ve kovaryansları aynı olmalıdır. Eğer $k = 0$ ise $cov(Y_t, Y_{t+0}) = var(Y_t) = \sigma^2$ 'dir

Durağan olmayan seriler birim kök içerirler. Bir serideki birim kök sayısı serinin durağan olana dek alınması gereken fark sayısına eşittir. Y_t serisi $d=1$ farkı alınınca durağan hale geliyorsa seri 1. dereceden durağandır denir ve $I(1)$ olarak gösterilir. Genel olarak seri d kez farkı alınınca durağan oluyorsa seri d . dereceden durağandır denir ve $I(d)$ ile gösterilir.

6.1 Trend Durağanlık ve Fark Durağanlık

Trend durağan olmayan süreçler için durağanlaştırmak için iki temel yöntem vardır:

- Durağan olmayan zaman serisi için kurulacak regresyon denkleminde, seri trend (zaman) üzerine regrese edilir. Daha sonra bu regresyondan elde edilen kalıntılar üzerinde analizler yapılır.
- Zaman serisine trend bir regresör olarak ilave edilerek gerekli analizler yapılır.

$$Y_t = \mu + B_t + e_t$$

(Y_t : deterministik trend, $e_t \sim IID(0, \sigma^2)$: durağan stokastik bileşen)

Rassal Yürüyüş:

Rassal yürüyüşü en basit şekilde şöyle gösterilir:

$$Y_t = Y_{t-1} + u_t$$

(u_t : beyaz gürültü)

Markov 1.derece özbağımlısal tasarımıyla rassal yürüyüş bağlantısı :

$$Y_t = \rho \cdot Y_{t-1} + u_t, -1 < \rho < 1$$

Rassal yürüyüşte $\rho = 1$ olduğu için, bu sürece “birim kök” (unit root) süreci de denilmektedir. Rassal yürüyüş sürecinde u_t sarsıntıları kalıcıdır:

$$Y_1 = Y_0 + u_1$$

$$Y_2 = Y_1 + u_2 = Y_0 + u_1 + u_2$$

$$Y_3 = Y_2 + u_3 = Y_0 + u_1 + u_2 + u_3$$

T dönemindeki deger :

$$Y_t = Y_0 + \sum_{i=1}^t u_i$$

Herhangi bir dönemdeki değerin daha önceki tüm rassal sarsıntıların toplamı olmasına, rassal yürüyüşün “sonsuz bellek” (infinite memory) özelliği de denir. $E(u_t) = 0$ olduğundan, $E(Y_t) = Y_0$ olduğuna dikkat edelir. Başka bir ifadeyle Y_t ’nin ortalaması sabittir. Rassal hatalar toplandığı için, $var(Y_t)$ sürekli artmakta ve böylece duraganlık varsayımı engellenmemiş olmaktadır. Y_t ’nin varyansının $var(Y_t) = t\sigma^2$ olduğu gösterilebilir. Buna göre, t sonsuza giderken varyans da sonsuza gitmektedir

6.2 Durağanlık Analizi

Durağanlık analizini grafiksel analiz, kolegram analiz ve birim kök analizi kullanılarak yapılmaktadır.

6.2.1 Birim Kök Analizi

Serinin durağanlığını ve durağanlık derecesini belirlemede kullanılır. Birim Kök Analizi'nde 5 farklı yöntem bulunmaktadır.

- Dickey Fuller Testi (DF)
- Genişletilmiş Dickey-Fuller Testi (ADF)
- Phillips- Perron Testi (PP)
- Kwiatkowski-Phillips-Schmidt-Shin Testi (Kpss)
- Ng Perron Testi

En çok kullanılan DF, ADF ve PP testleridir.

6.2.2 Dickey Fuller Testi (DF)

Serinin durağan olmadığı (birim köke sahip olduğu) boş hipotezinin , durağan olduğu (birim kök olmadığı) alternatif hipotezine göre sınanmasıdır. Y_t değişkeninin bu dönemde aldığı değerin geçen dönemdeki değeri olan ile ilişkisi,

$$Y_t = p \cdot Y_{t-1} + u_t$$

şeklinde ifade edilir. (u_t : kalıntı terim)

Yukarıdaki model 1. dereceden otoregresif AR(1) modelidir. ρ katsayısı 1 eşit olursa birim kök sorunu (durağan olmama durumu) ortaya çıkmaktadır ve model $p = 1$ ise

$$Y_t = Y_{t-1} + u_t$$

şeklini almaktadır. Sonuç bir önceki dönemde iktisadi değişkenin değerinin ve dolayısıyla o dönemde maruz kaldığı şokun olduğu gibi sistemde kalması anlamına gelir. Bu şokların kalıcı nitelikte olması serinin durağan olmaması ve zaman içinde gösterdiği trendin stokastik olması anlamına gelir.

$p < 1$ ise geçmiş dönemlerdeki şoklar belli bir süre etkilerini sürdürseler de, bu etki giderek azalacak ve kısa bir dönem sonra tamamen ortadan kalkacaktır.

6.2.3 Genişletilmiş Dickey-Fuller Testi (ADF)

Dickey-Fuller testi, hata terimlerinin otokorelasyon içermesi halinde kullanılamamaktadır. Zaman serisinin gecikmeli değerleri kullanılarak hata terimindeki otokorelasyon ortadan kaldırılabilir. Dickey-Fuller bağımlı değişkenin gecikmeli değerlerini, bağımsız değişken olarak modele dahil eden yeni bir test geliştirmiştir. Bu test Genişletilmiş Dickey-Fuller testidir. Burada gecikmeli değişkene ait uygun gecikme mertebesi belirlenirken Akaike ve Schwarz kriterlerinden yararlanılmaktadır.

ADF denklemi :

$$\delta Y_t = a + b_t + \gamma Y_{t-1} + u_{t-1} + c \sum_{i=1}^n \delta Y_t$$

δ : delta

6.2.4 Phillips- Perron Testi (PP)

Phillips-Perron birim kök testi ise hata teriminin zayıf derecede bağımlı olmasına ve heterojen olarak dağılmasına izin vermektedir. Bu sayede otokorelasyon sorunu ortaya çıkmamaktadır.

6.2.5 Kwiatkowski-Phillips-Schmidt-Shin Testi (Kpss)

KPSS testinde amaç gözlenen serideki deterministik trendi arındırarak serinin durağan olmasını sağlamaktır. Bu testte kurulan birim kök hipotezi ADF testinde kurulan hipotezlerden farklıdır. Sıfır hipotezi serinin durağan olduğunu ve birim kök içermediğini, buna karşın alternatif hipotez ise seride birim kök olduğunu ve durağan olmadığını ifade eder. Boş hipotezdeki durağanlık trend durağanlıktır. Çünkü seriler trendden arındırılmışlardır.

Trendden arındırılan seride birim kök olmaması, serinin trend durağanlığını gösterir. KPSS testinin en önemli özelliği bir veya daha büyük bir MA yapısı içeren serilerde ADF' nin aksine gücünün azalmamasıdır.

KPSS testi LM testi ile benzer biçimde belirlenmektedir. Dolayısıyla LM istatistiğinin oluşumu önemlidir. LM testinde boş hipotez, rassal yürüyüşün sıfır varyansa sahip olduğunu ve serinin deterministik trend, rassal yürüyüş ve durağan kalıntılar toplamından oluştuğunu ima eder.

6.2.6 Ng Perron Testi

Ng-Perron birim kök testi, Phillips-Perron birim kök testinde ortaya çıkan hata teriminin hacmindeki çarpıklığın düzeltilmesi için geliştirilmiştir. PP testinde serilerde negatif hareketli ortalama yapısı olduğunda, büyük oranda hata teriminde örneklem çarpıklığı olmaktadır. DF testlerinde bu durum çok büyük bir sorun yaratmamaktadır.

NG - Perron testi bu nedenle PP testlerini ve bilgi kriterlerini modifiye etmektedir. MSB ve MPT lerde sıfır hipotezi birim kökün olmadığını , alternatif hipotez ise birim kökün varlığını ifade eder. adf.test ve pp.test için: - H0: birim kök içerir. - HA: birim kök içermez, durağandır.

kpss.test için: - H0: level veya trend durağandır. - HA: durağan değildir.

Test sonuçlarına göre adf ve pp testleri için p-value (test istatistiğinin olasılık değeri) %5'den küçük ise (%95 güven seviyesinde) testin boş hipotezi ret edilir. Kpss test için ise durumun tam tersi doğrudur.

7 Normallik Testi- Shapiro Wilk Test

Veri seti üzerinde yapılan bir çok veri analizi verinin normal dağıldığı varsayımına dayanır. Veri analizine başlamadan önce verinin normal dağılıma uygun olup olmadığına bakılmalıdır. Normallik kontrolü için yaygın olarak kullanılan testlerden birisi de Shapiro-Wilks testidir.

Shapiro-Wilk (1965) testi, verinin sıra istatistiklerinin uygun bir lineer bileşenin karesinin, kareler toplamına bölümü uygulanarak elde edilir.

$$W = \frac{(\sum_{i=1}^n a_i x_i)^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \text{ ve } a_i = (a_1, a_2, \dots, a_n)$$

$a_i = (a_1, a_2, \dots, a_n)$: standart normal dağılımda $N(0,1)$ ' da n adet sıra istatistiğinin beklenen değerlerinin vektörünü ifade eder.

Sadece numeric (numeric factor) veriyi, argüman olarak shapiro.test(veri) fonksiyonuna eklenir.

8 Otokorelasyon

Otokorelasyon durumunda parametrelerin en küçük kareler tahmincileri sapmasız ve tutarlıdır, ancak etkin değildir. Hata teriminin varyansının tahmincisi sapmalıdır ve bu yüzden parametrelerin varyansları da sapmalı olur. Pozitif otokorelasyon varsa sapma negatif olur. Varyanslar olduğundan küçük bulunur. Bunun sonucunda t test istatistiği değeri büyük çıkar. Böylece anlamsız bir katsayının anlamlı olma olasılığı artar. R_2 de yükselir. Dolayısıyla F değeri olduğundan büyük bulunur. Sonuç olarak t ve F testleri güvenilirliğini yitirip yanıltıcı sonuç verirler. Otokorelasyonu durumunu göstermek için 2 yöntem kullanılır.

- Grafik Yorumu
- Otokorelasyon Testleri (Durbin-Watson d , Wallis , King , Breush ve Godfrey)

NOT: Zaman serilerinin durağan olması demek, zaman içinde varyansın ve ortalamanın sabit olması ve gecikmeli iki zaman periyodundaki değişkenlerin kovaryansının değişkenler arasındaki gecikmeye bağlı olup, zamana bağlı olmamasıdır.

8.1 Grafik Yorumu

8.1.1 Otokorelasyon Katsayısı (ACF)

Otokorelasyon katsayısı, zaman serisiyle bu serinin gecikmeli serileri arasındaki ilişkiyi gösterir. Serideki bütün gecikmeler otokorelasyon fonksiyonunu oluşturmaktadır. Otokorelasyona; ihmal edilen değişkenler, modelin fonksiyonel biçiminin doğru belirlenmemesi, verilerle ilgili ölçme hataları, tesadüfi olarak ortaya çıkan savaş, kuraklık gibi olayların, etkilerini ortaya çık-tıkları dönemlerden sonra da sürdürmesi neden olabilirler.

$$E(u_i, u_j) = 0, i \neq j$$

şeklinde ifade edilir.

$$ACF(k) = \frac{\sum_{t=1+k}^T (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^T (Y_t - \bar{Y})^2}$$

Eğer bir seri tesadüfi ise, her hangi bir gecikmede yani Y_t ve Y_{t-k} arasındaki otokorelasyon-lar sıfıra yakın olmaktadır. Bu durumda zaman serisinin ardışık değerlerinin birbirleriyle ilişkisi yoktur.

Eğer bir zaman serisi trende sahipse ise, Y_t ve Y_{t-1} arasında yüksek korelasyon bulunacaktır. Birkaç gecikmeden sonra otokorelasyon katsayıları hızla sıfıra yaklaşacaktır. İlk gecikmede otokorelasyon kat sayısı 1'e yakındır. İkinci gecikmede de oldukça yüksektir. Ve daha sonra hızla azalır.

Hata terimleri arasında bir ilişki varsa yani otokorelasyon durumu söz konusu ise

$$E(u_i, u_j) \neq 0, i \neq j$$

şeklinde ifade edilir.

8.1.2 Kısmi Otokorelasyon Fonksiyonu (PACF)

Kısmi korelasyon; ilk başta bilinmeyen zaman serisinin değerleri olan p ve q değerlerinin kaçınıcı dereceden olduğunu bulmak için uygulanır. Kısmi korelasyon, gecikmeli değişkenler arasındaki ilişkiyi ifade eder. Yani kısmi korelasyon diğer bütün gecikmeli gözlemlerin etkisinden arındırıldıktan sonra x_t değişkeni ile bu değişkenden herhangi bir k-gecikmesiyle elde edilen x_{t+k} değişkeni arasındaki ilişkiyi inceler. Kısmi korelasyon PACF ile gösterilir.

8.2 Otokorelasyon Testleri

8.2.1 Durbin- Watson d Testi

Sabit parametresi olan modellerde 1.dereceden otokorelasyonun varlığını test etmek için kullanılır.

Testin 4 aşamaması vardır. Sırasıyla incelersek ; - Hipotezin kurulması

H_0 : $\rho=0$ ise otokorelasyon yoktur. H_1 : $\rho \neq 0$ ise otokorelasyon vardır.

- Tablo değerlerinin incelenmesi

Durbin Watson tablo değerlerinden d istatistiğinin seçilen bir anlamlılık düzeyi ile gözlem sayısı ve açıklayıcı değişken sayısına göre alt (d_L) ve üst (d_U) sınır değerleri bulunur.

- Durbin Watson d istatistiği hesaplanması

$$d = \frac{\sum_{t=1}^n (e_t - e_{t-1})^2}{\sum_{t=1}^n (e_t)^2}$$

- Karşılaştırma yapılması ve karar verilmesi

Eğer $0 < d < d_L$ ise pozitif otokorelasyon vardır.

Eğer $d_L \leq d \leq d_U$ ise karar verilmemektedir.

Eğer $d_U < d < 4-d_U$ ise otokorelasyon yoktur.

Eğer $4-d_U \leq d \leq 4-d_L$ ise karar verilmemektedir.

Eğer $4-d_L < d < 4$ ise negatif otokorelasyon vardır.

Durbin-Watson testini : - Model sabit terimsiz ise, - Bağımsız X değişkenleri stokastikse, - Otokorelasyonun derecesi 1'den büyük ise, - Zaman serisinde ara yıllar noksan ise, - Modelde bağımsız değişken olarak gecikmeli bağımlı değişken varsa uygulanamaz.

Durbin-Watson d istatistiği tablosu $n < 15$ için d_L ve d_U değerlerini vermemektedir. Bu durumda, Von-Neumann testi kullanılmaktadır.

8.2.2 Wellis Testi:

3'er aylık verilerde 4.dereceden otokorelasyon varlığını test etmek için kullanılır. Durbin Watson testinin, 4.dereceden otokorelasyon varlığını test etmek için düzenlenmiş halidir.

8.2.3 Breush ve Godfrey Testi:

Yuksek dereceden otokorelasyonun varliginin test edilmesi icin kullanilir.Orjinal denklemde bagimli degiskenin gecikmeli halinin bagimsiz degiskenler arasinda yer almasi durumunda da kullanilir.

Bagimli degisken: Orjinal model kalintilari

Bagimsiz degiskenler: Orjinal modelin aciklayici degiskenleri ve incelenen otokorelasyonun derecesi sayisinda gecikme katsayisi modelde yer alır.

Örneğin 2.dereceden otokorelasyonun varligi arastiriliyorsa, orjinal modelin bagimsiz degiskenleri ve

$$u \cdot (t - 1), u \cdot (t - 2)$$

katsayilari yeni modelde bagimsiz degiskenler arasinda yer alır. Test istatistigi:

$$n \cdot R_2$$

ile yeni modelin karesi hesaplanır, n ile carpilarak test istatistigi elde edilir.

9 Giriş

Bitirme çalışmasında toplamsal ve çarpımsal ayrışma modellerini analiz etmek ve modeller için öngöründe bulunabilmek için iki farklı veri inceleyeceğiz. İlk verimiz ABD de kullanılan doğal gaz tüketimi hakkında olup çarpımsal ayrışma modele örnektir. İkinci verimiz Florida'daki ham petrolün üretimi hakkında olup toplamsal ayrışma modeline örnektir. Verilerin hangi modele ait olduğu ve öngörü analizleri aşağıda detaylı olarak inceleyeceğiz.

10 Natural Gas Futures

Verinin Yayıncısı: U.S. Energy Information Administration Data

ABD ulusal ve devlet verilerinin; elektrik, kömür, doğal gaz ve petrol gibi tüm ana enerji ürünleri üzerindeki üretim, tüketim ve diğer göstergelere ilişkisini göstermektedir.

Verinin Adı: Natural Gas Futures Contract 3

ABD ulusal ve devlet verilerinin; doğal gaz üretim, tüketim ve diğer göstergelere ilişkisini göstermektedir. Veri zaman serisi olarak ilerlemektedir.

Verinin Sıklığı: Aylık

Verinin Zaman Aralığı: 31.01.1994- 31.03.2018

Verinin Alındığı Yer: Quandl

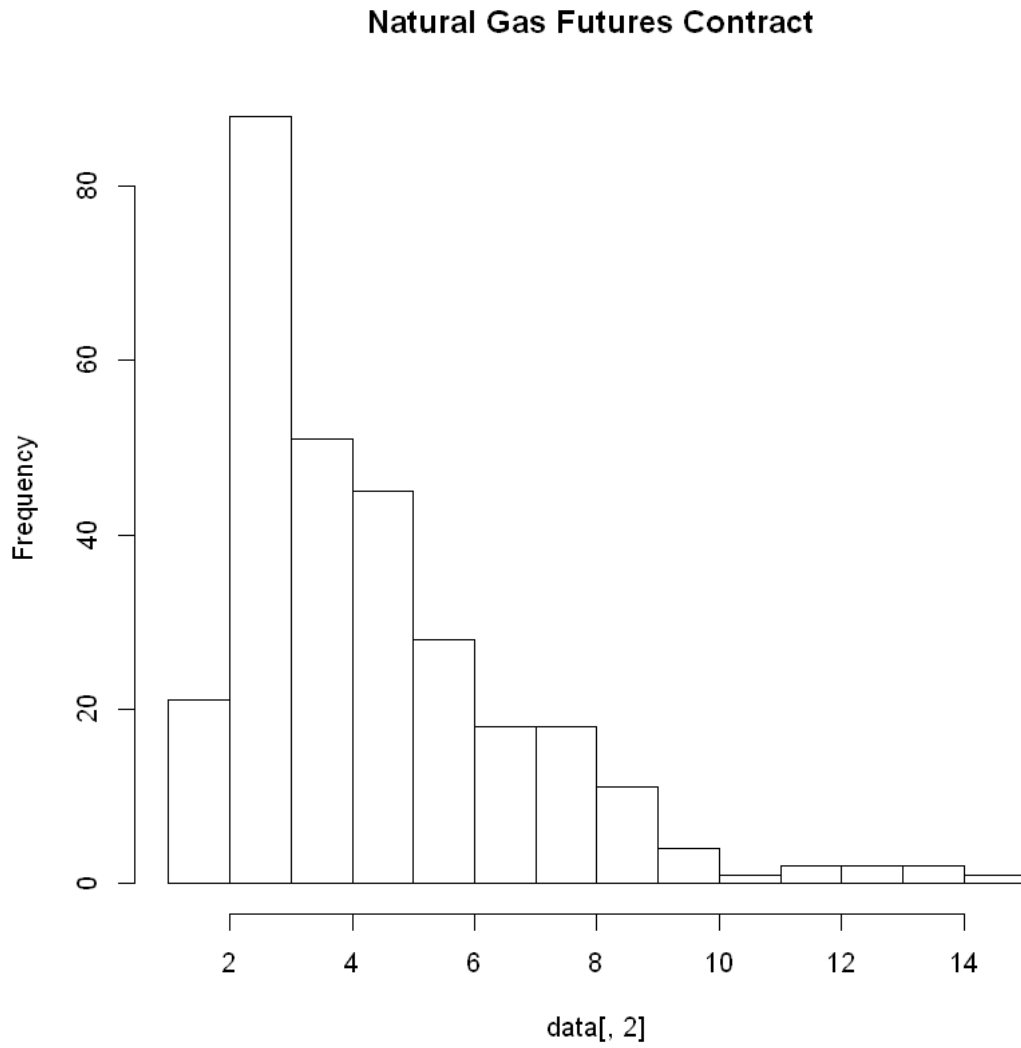
Quandl yatırım profesyonellerine hizmet veren finansal, ekonomik ve alternatif veriler için bir platformdur. Verileri 500'den fazla yayıncıdan sağlar. Quandl'ın verilerine bir API aracılığıyla erişilebilir. Bir çok programlama dili için API erişimi (R, Python, Matlab, Maple ve Stata) vardır.

Excel eklentisi, stok fiyat bilgisi dahil olmak üzere verilere erişim sağlar.

```
In [2]: data<-Quandl("EIA/NG_RNGC3_M", api_key="BtbwkANCR4aHKBscujRb", collapse="monthly")
```

Veriyi quandl veri sitesinden alıyoruz. Veri iki sütundan oluşmaktadır. İlk sütun tarih ikinci sütun ise kullanım miktarıdır.

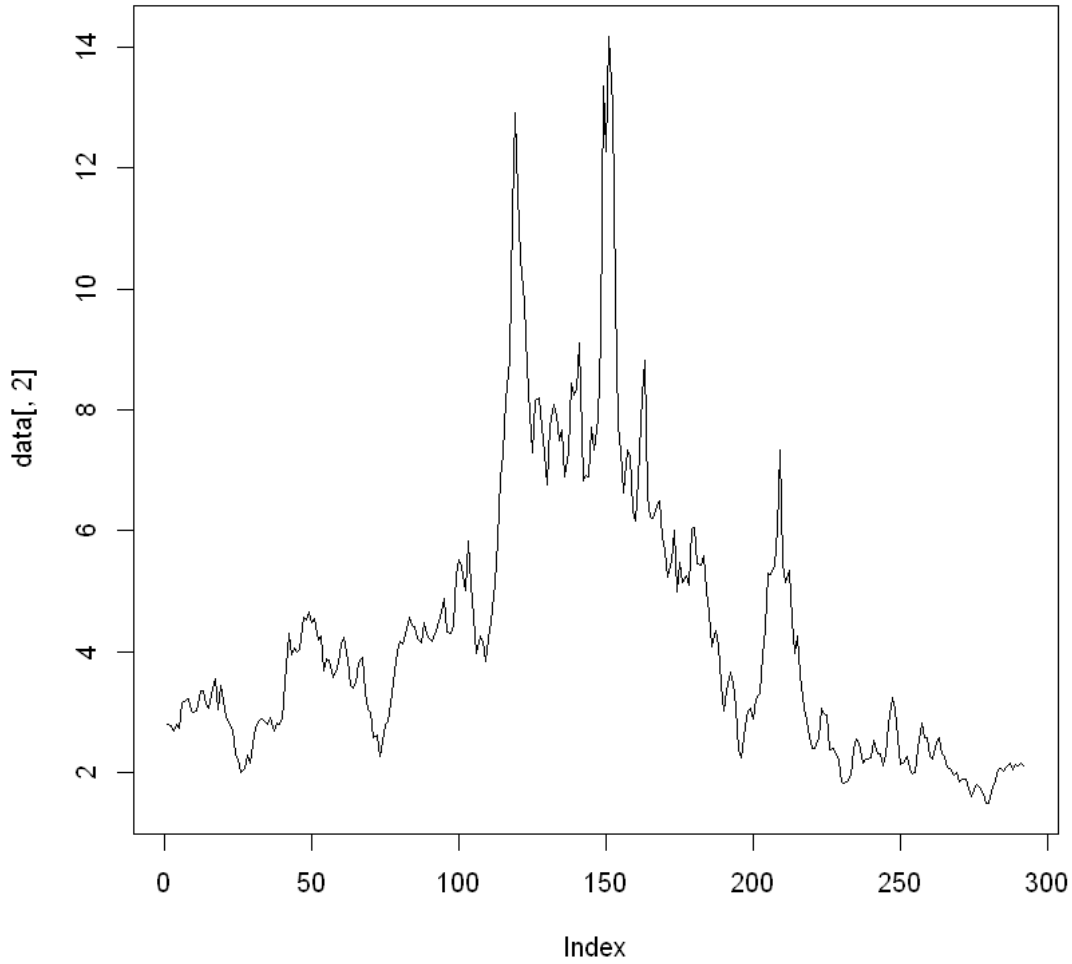
```
In [3]: hist(data[,2], main="Natural Gas Futures Contract ")
```

Aşağıda verimizi çizgizel grafik gösterimini görüyoruz.

```
In [4]: plot(data[,2],type="l", main="Natural Gas Futures Contract ")
```

Natural Gas Futures Contract



10.1 Decompose - Geleneksel Zaman Serisi Ayırışım Yöntemleri

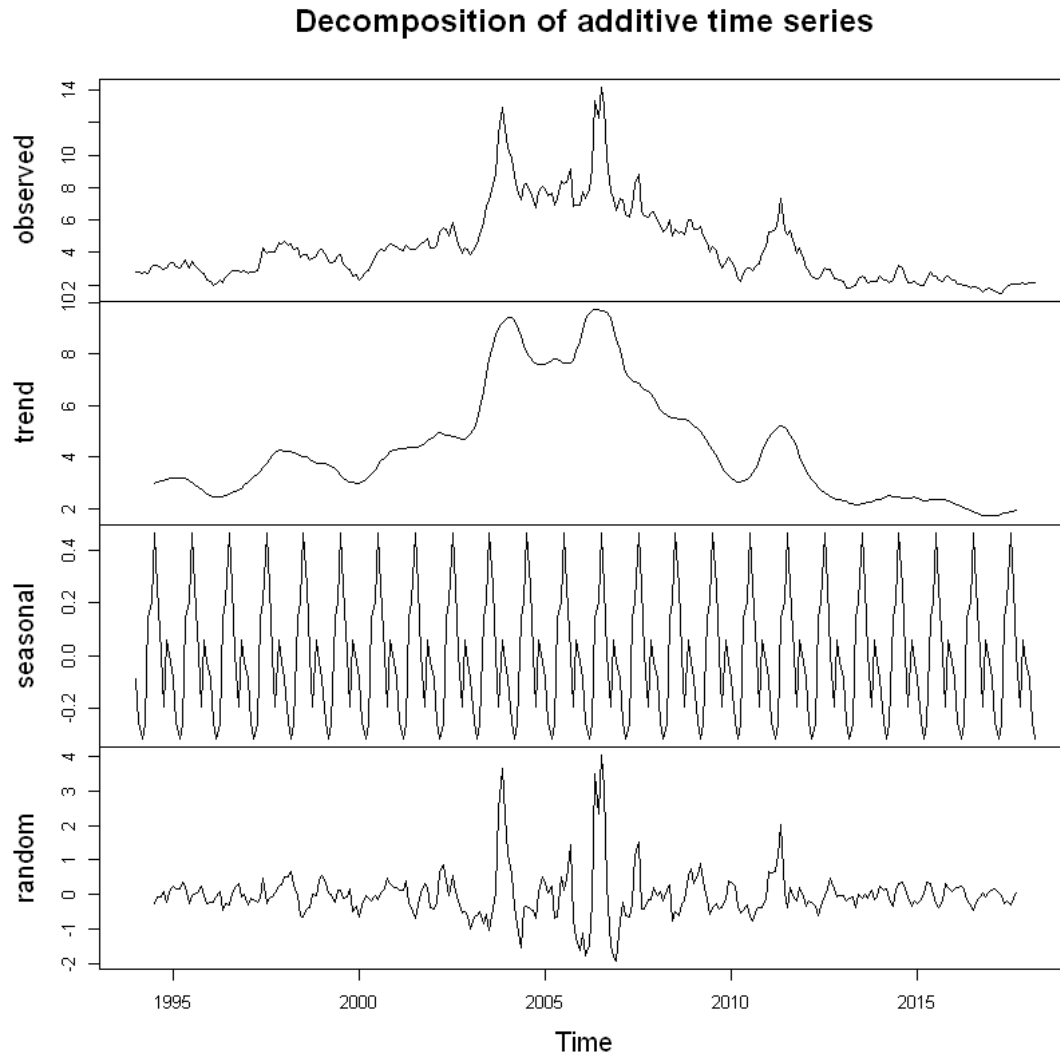
Veriyi sezonsal veriden ayırmak için R paketinin decompose fonksiyonunu kullanıyoruz. Decompose fonksiyonu ile veri sezonsal etkiden ve trend etkisinden arınmaktadır. Veri aylık olduğundan dolayı sıklığı 12 olarak alıyoruz.

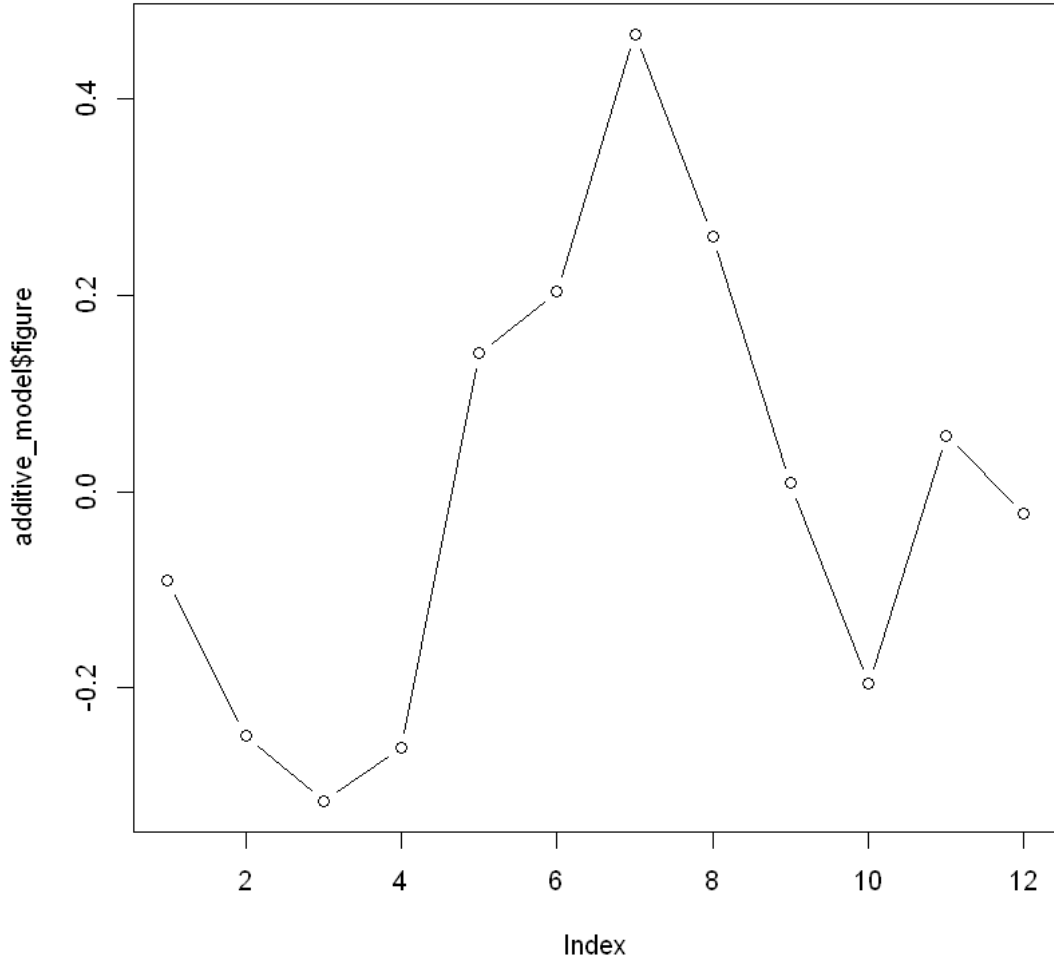
Ek olarak verinin ayırışma yönteminin, toplamsal ya da çarpımsal olduğunu anlamak için `type= c("additive")` ve `type= c("multiplicative")` fonksiyonu kullanıyoruz.

Dönemsel mevsimselliği incelemek için ise R 'ın figure fonksiyonu kullanılmaktadır.

Aşağıda zaman serisi (ts) verimize toplamsal ayırıştırma tipine decompose fonksiyonu uygulanıyor. Böylelikle verimizdeki sezonsal ve trendsel etkiden çıkartmış oluyoruz. Plot fonksiyonu ile yaptığımız modeli grafiksel olarak gösteriyoruz. Figure fonksiyonu ile de zaman serimizin 12 aylık sezonsal etkisini gösteriyoruz.

```
In [5]: additive_model<-decompose(ts(data[,2],start = c(1994, 1), end = c(2018, 3), freq=12),type="additive")
plot(additive_model)
plot(additive_model$figure,type="b")
```



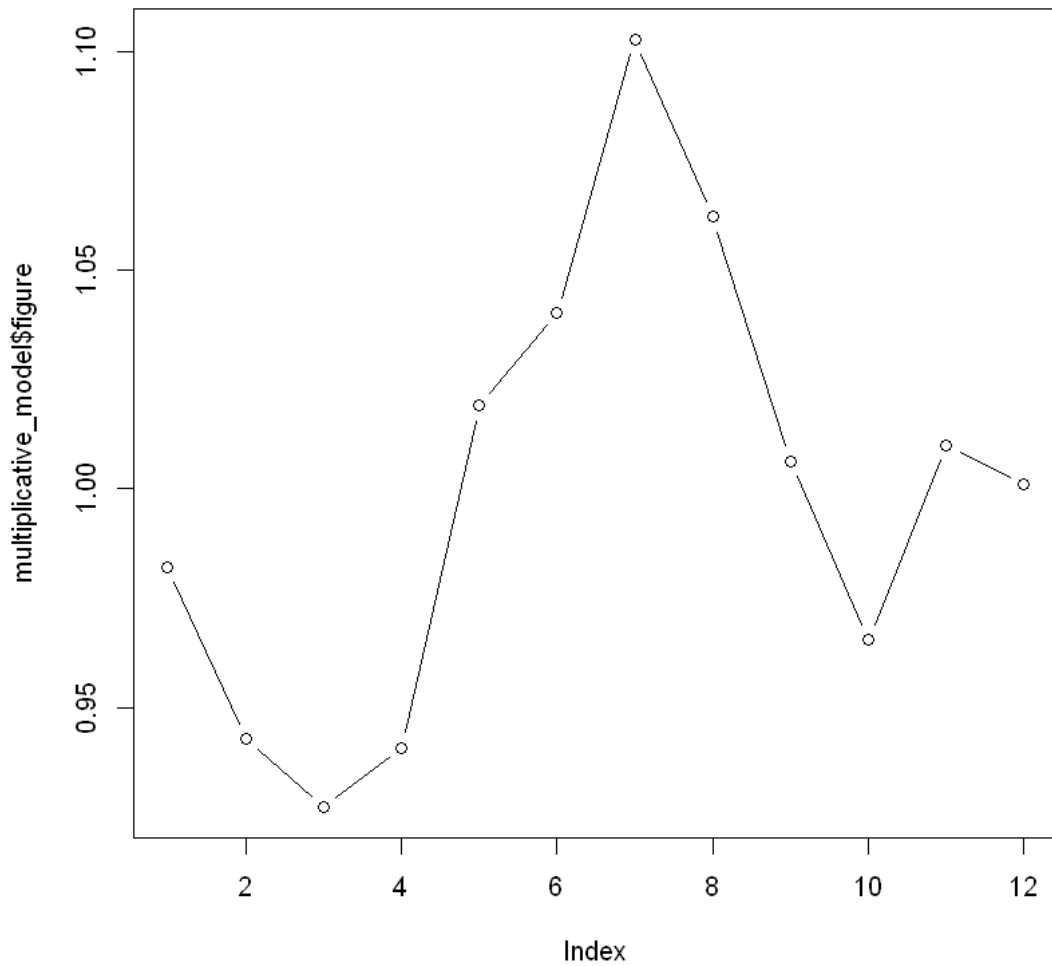


Toplamsal ayrışma modelini incelediğimizde trend bileşeninin 1994 yıllarında düşük seviyelerde olduğunu görüyoruz. Ancak 2000 yılından itibaren trend değerleri artmıştır. Trends el etkiyi incelediğimizde 2003-2008 arası trend bileşeninin değerleri en yüksek noktalara ulaşmıştır. 2013'ten itibaren ise trends el etki giderek azaldığını görüyoruz. Sezon sal etkiyi incelersek veri bütün yıllarda aynı sezon sal değerleri aldığını görüyoruz.

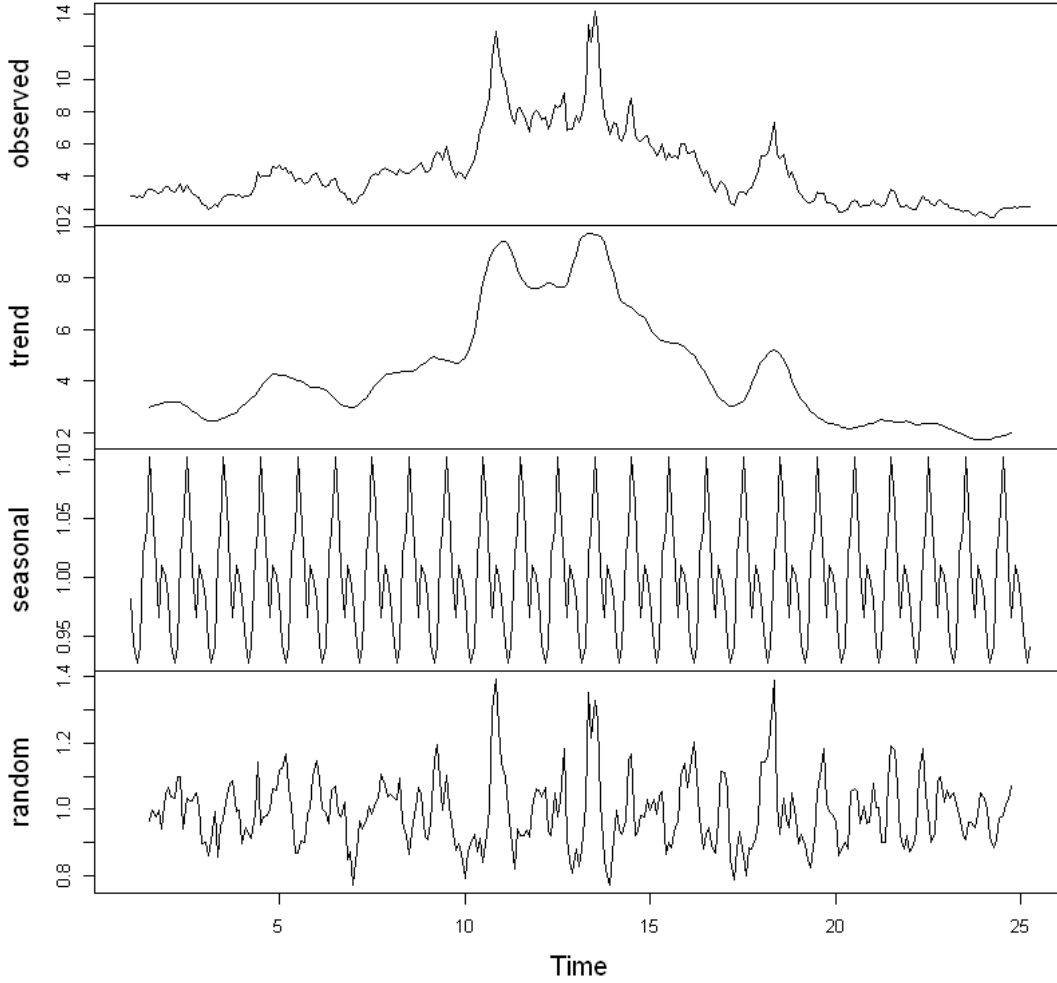
Figure fonksiyonu ile gelen mevsimsellik etki incelendiğinde veri şubat ayında en düşük seviyede, haziran ayında ise en yüksek mevsimsellikten etkilendiğini görebiliyoruz.

Aşağıda zaman serisi (ts) verimize çarpımsal ayrıştırma tipine decompose fonksiyonu uygulanıyor. Böylelikle verimizdeki sezonsal ve trendsel etkiden çıkartmış oluyoruz. Plot fonksiyonu ile yaptığımız modeli grafiksel olarak gösteriyoruz. Figure fonksiyonu ile de zaman seriminin 12 aylık sezonsal etkisini gösteriyoruz.

```
In [6]: multiplicative_model<-decompose(ts(data[,2], freq=12), type= c("multiplicative"))  
       plot(multiplicative_model$figure, type="b")  
       plot(multiplicative_model)
```



Decomposition of multiplicative time series



Çarpımsal ayrışma modelini incelediğimizde trend bileşeninin 1994 yıllarında düşük seviyelerde olduğunu görüyoruz. Ancak 2000 yılından itibaren trend değerleri artmıştır. Trends el etkiyi incelediğimizde 2003-2008 arası trend bileşeninin değerleri en yüksek noktalara ulaşmıştır. 2013'ten itibaren ise trends el etki giderek azaldığını görüyoruz. Sezon el etkiyi incelersek veri bütün yıllarda aynı sezon el değerleri aldığını görüyoruz. Random değerlerinin sürekli değişiklik gösterdiğini görüyoruz.

Figure fonksiyonu ile gelen mevsimsellik etki incelendiğinde veri şubat ayında en düşük seviyede, haziran ayında ise en yüksek mevsimsellikten etkilendiğini görebiliyoruz.

Toplamsal ve çarpımsal ayrışma modellerinde en büyük farklılık random değerlerinde gözlenmektedir. Toplamsal modelde random değerler -2 ile 4 arasında seyrederken , çarpımsal modelde ise 0.8 ile 1.4 arasında seyretmektedir.

Toplamsal ve çarpımsal ayrışma model incelendiğinde büyük bir farklılık görünmediğinden bu aşamada karar verilememektedir. Bu yüzden ilerleyen aşamalarda veriye arima modeli uygulayacağız.

Summary fonksiyonu ile verinin özetini göstermiş olunur.

Aşağıda asıl verimize, toplamsal ayrıştırma sezonsal ve trendsel etkiden ayrılmış verimize, çarpımsal ayrıştırma ile sezonsal ve trendsel etkiden ayrılmış verimizin özetlerini görüyoruz.

```
In [7]: summary(data[,2])
summary(na.omit(additive_model$random))
summary(na.omit(multiplicative_model$random))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.490	2.572	3.669	4.308	5.352	14.178

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-1.925256	-0.323617	-0.062930	-0.001936	0.196409	4.021991

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.7722	0.9172	0.9859	0.9938	1.0545	1.3901

10.2 Normallik Testi- Shapiro Wilk Test

Verinin normalliğini kontrol etmek için shapiro.test fonksiyonu kullanılmaktadır. P değeri testen ne kadar emin olduğunu göstermektedir. P değerinde güven aralığını genelde %95 olarak alınır, yani p değeri 0.05'tan küçük ise yaptığımız testen emin olabiliriz. W değeri 1 değerine ne kadar yakınsa veri normalliğe o derece yakındır.

Shapiro testini toplamsal ve çarpımsal ayrıştırma modeline uygulanır.

```
In [8]: shapiro.test(na.omit(additive_model$random))
shapiro.test(na.omit(multiplicative_model$random))
```

Shapiro-Wilk normality test

```
data: na.omit(additive_model$random)
W = 0.79155, p-value < 2.2e-16
```

Shapiro-Wilk normality test

```
data: na.omit(multiplicative_model$random)
W = 0.96499, p-value = 2.579e-06
```

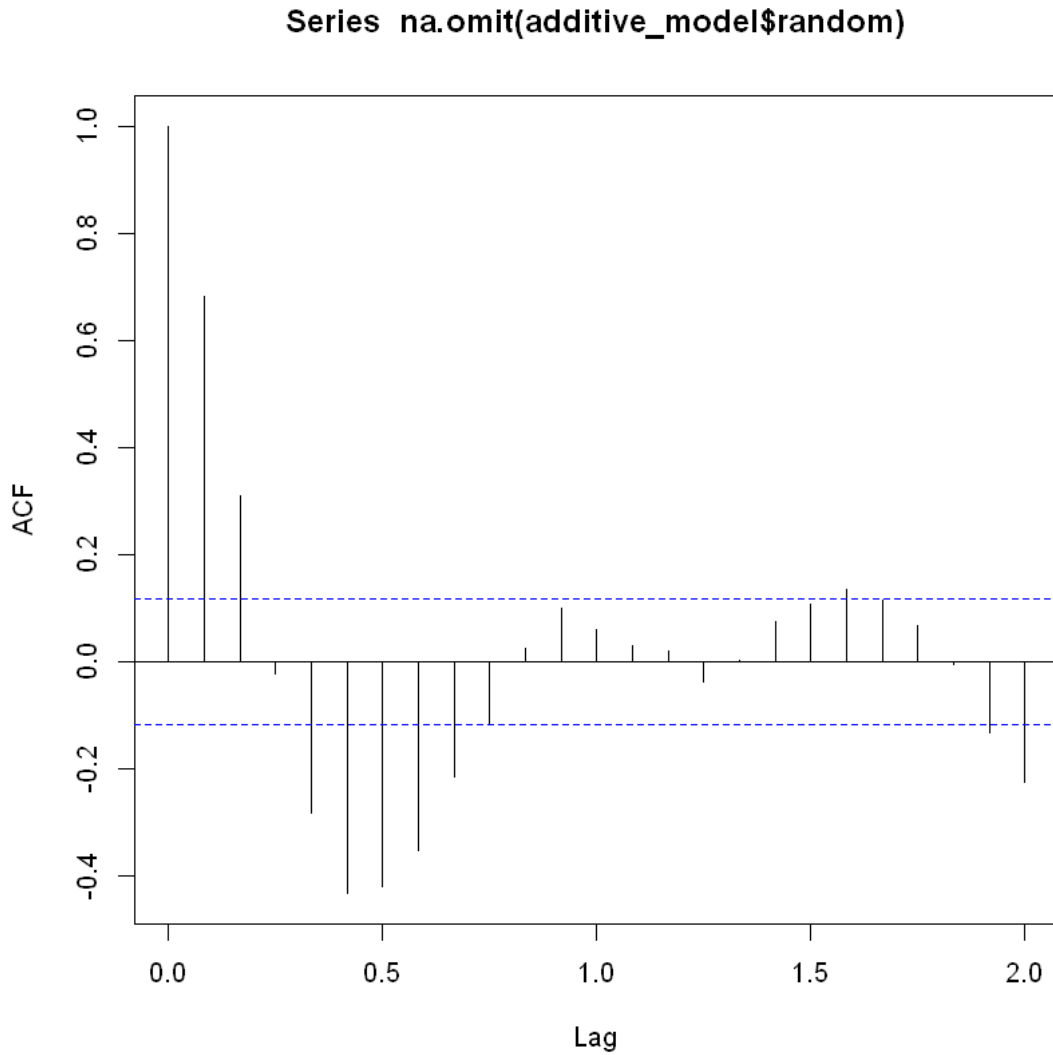
Toplamsal ve çarpımsal ayrıştırma modeline uygulanan normallik testi incelendiğinde p değerlerinin oldukça küçük, neredeyse 0 yakın olduğunu görüyoruz. Yaptığımız iki testte de oldukça eminiz. W değeri incelendiğinde ise çarpımsal modelin 1 daha yakın olduğu görülmektedir. Verimiz normaldir, diyebiliriz.

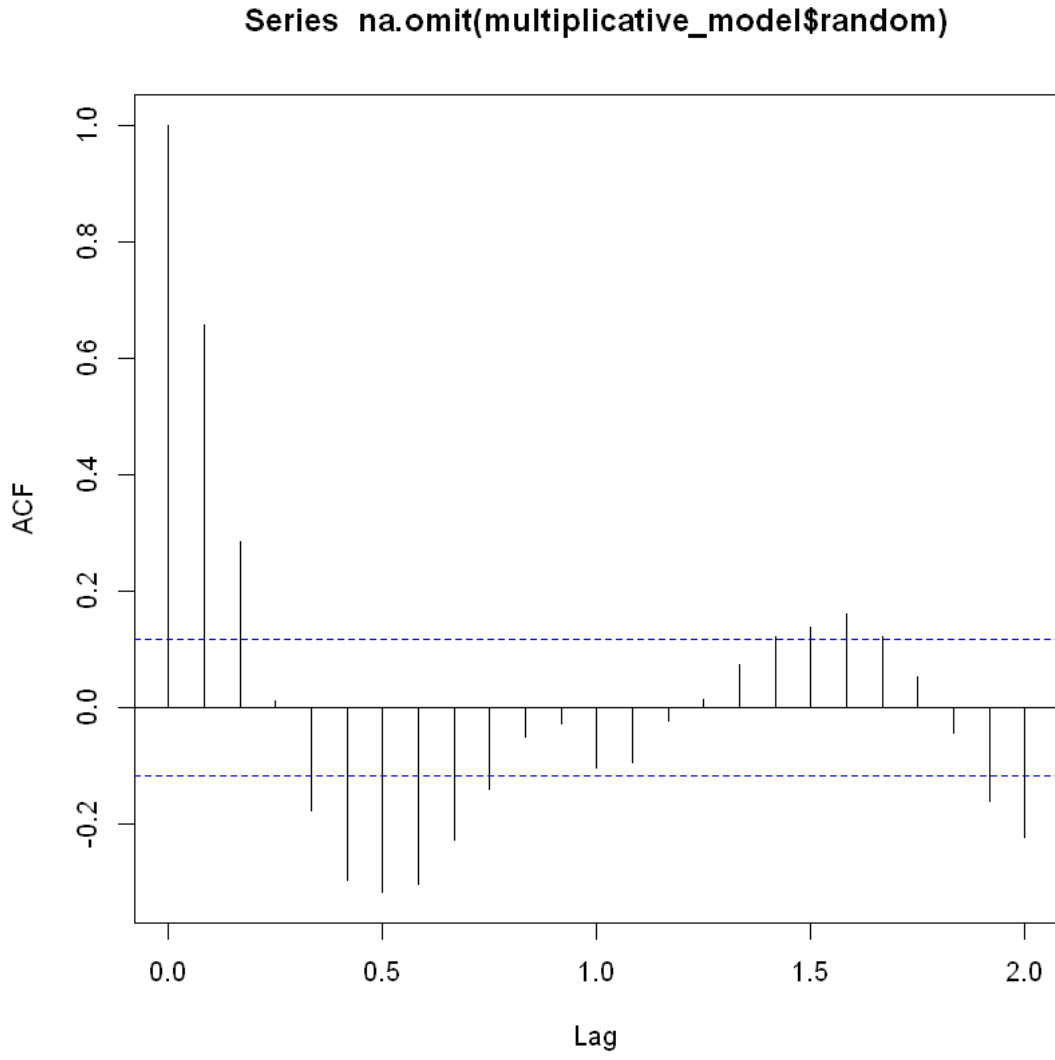
10.3 Otokorelasyon

Zaman serisiyle bu serinin gecikmeli serileri arasındaki ilişkiyi göstermek için R'ın acf ve pacf fonksiyonunu toplamsal ve çarpımsal ayrışma modeline uygulanır.

Otokorelasyon fonksiyonu ile durağanlıkta ölçülebilir. Eğer ACF(otokorelasyon fonksiyonu) çok yüksek bir değerden başlayıp çok yavaş küçülüyorsa, bu seri için durağan değildir diyebiliriz.

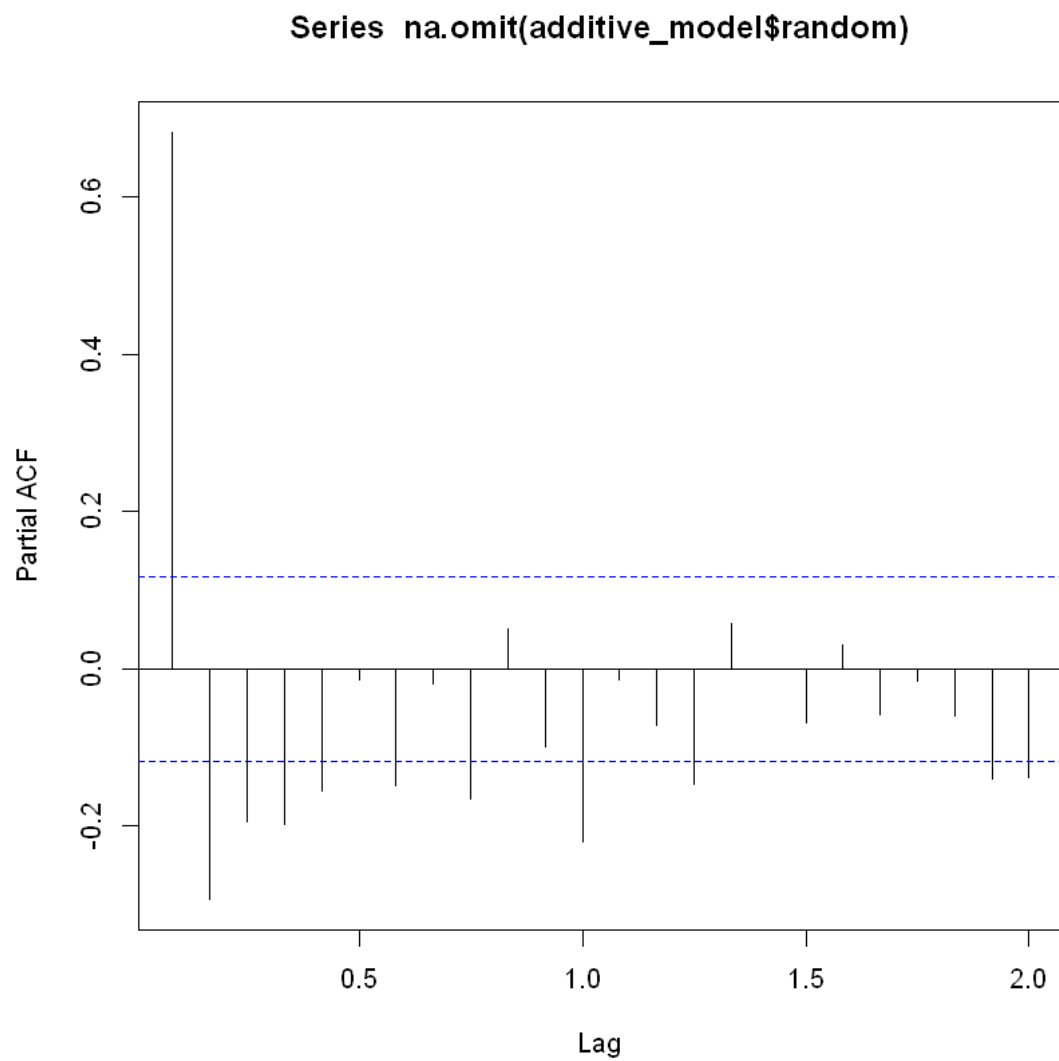
```
In [9]: acf(na.omit(additive_model$random))  
        acf(na.omit(multiplicative_model$random))
```



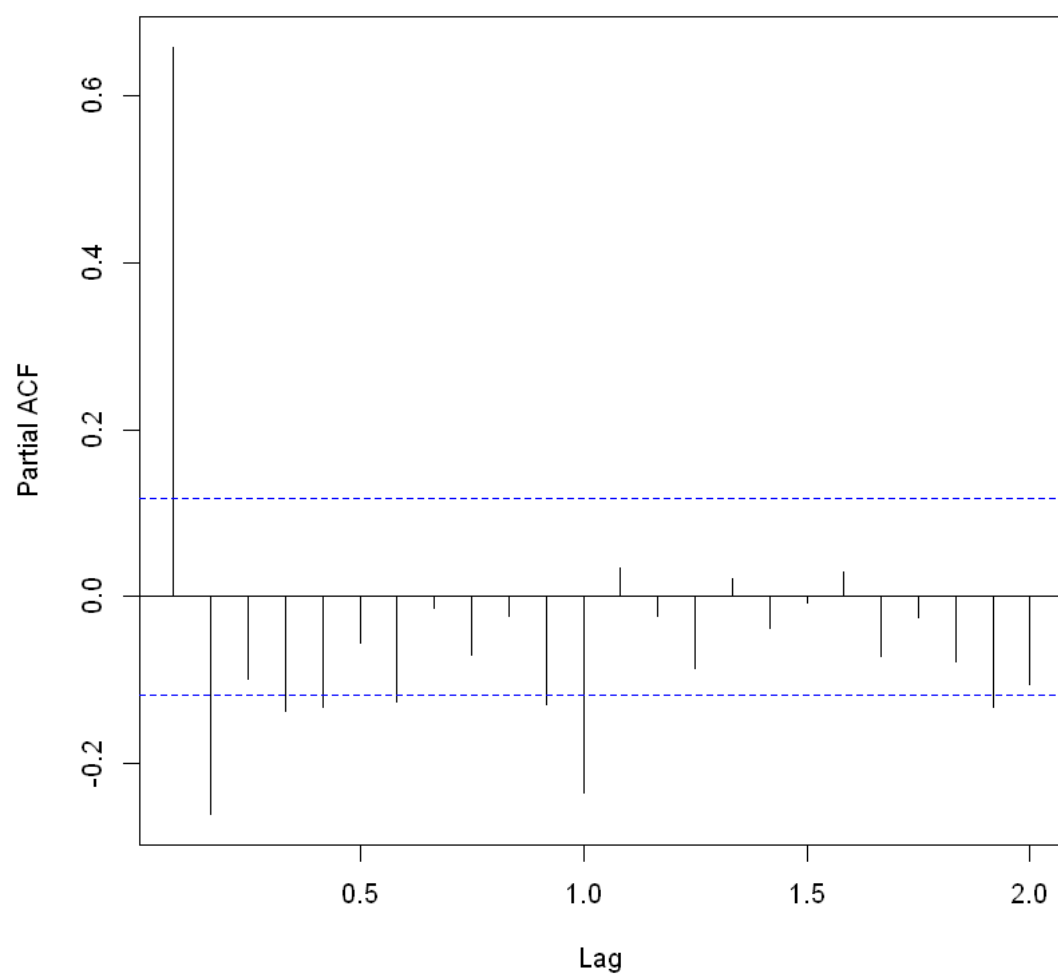


Toplamsal ve çarpımsal ayrıştırma modelleri için acf değerlerini incelediğimizde iki durumda durağan halde olduğunu söyleyebiliriz.

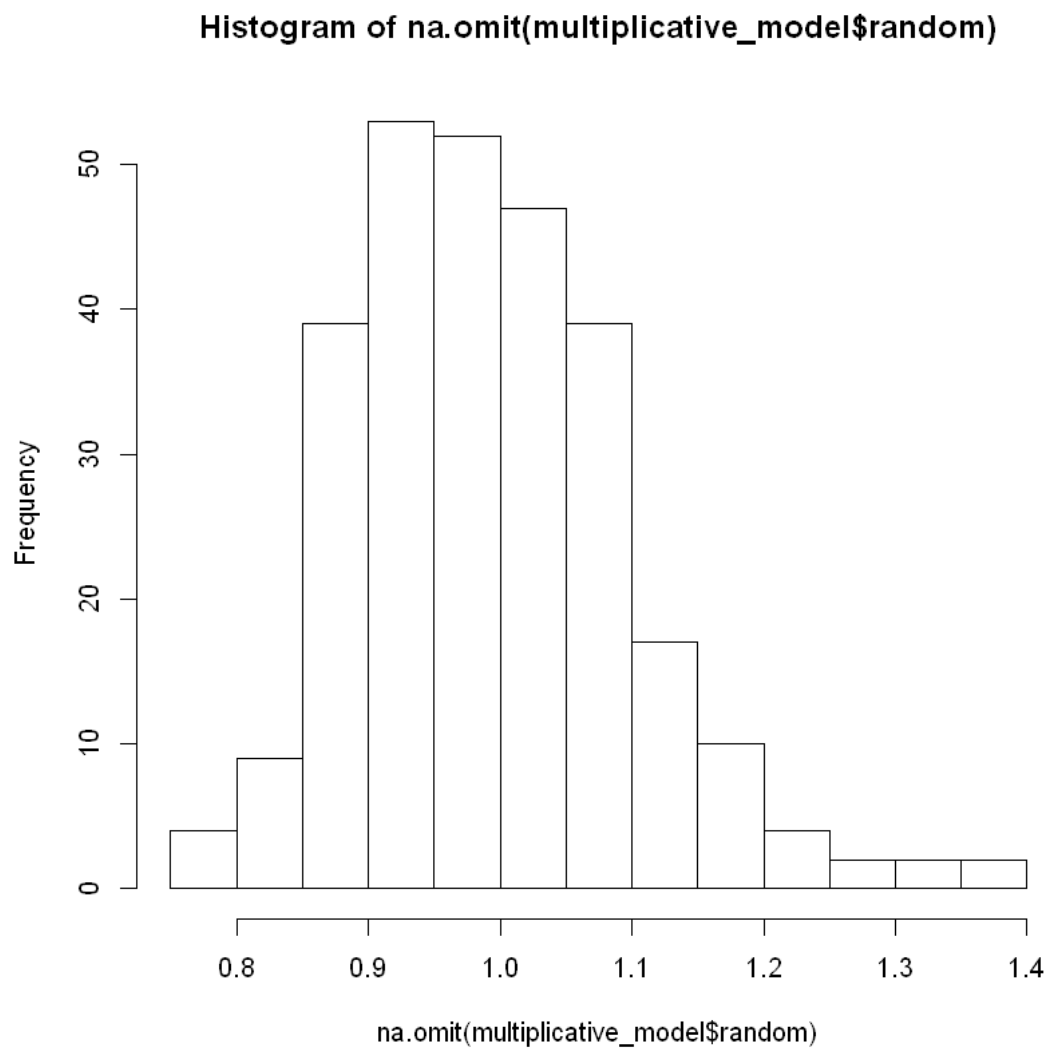
```
In [10]: pacf(na.omit(additive_model$random))  
         pacf(na.omit(multiplicative_model$random))
```



Series na.omit(multiplicative_model\$random)



```
In [11]: hist(na.omit(multiplicative_model$random))
```



10.4 Durağanlık Analizi

Veride durağanlık durumunu belirlemek için sırasıyla ;Augmented Dickey-Fuller Test-Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test - Phillips-Perron Birim Kök Testi uygulanır.

Augmented Dickey Fuller Test İlk olarak Augmented Dickey Fuller testini verinin bileşenlerden ayrılmamış haline, toplamsal ve çarpımsal ayrışma modellerine sırasıyla uyguluyarak verinin durağanlığını kontrol ediyoruz. ADF testini R içinde bulunan `adf.test` fonksiyonu ile elde edebiliriz.

```
In [12]: adf.test(data[,2])
          adf.test(na.omit(additive_model$random))
          adf.test(na.omit(multiplicative_model$random))
```

Augmented Dickey-Fuller Test

```
data: data[, 2]
Dickey-Fuller = -2.3992, Lag order = 6, p-value = 0.408
alternative hypothesis: stationary
```

```
Warning message in adf.test(na.omit(additive_model$random)):
"p-value smaller than printed p-value"
```

Augmented Dickey-Fuller Test

```
data: na.omit(additive_model$random)
Dickey-Fuller = -8.8024, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary
```

```
Warning message in adf.test(na.omit(multiplicative_model$random)):
"p-value smaller than printed p-value"
```

Augmented Dickey-Fuller Test

```
data: na.omit(multiplicative_model$random)
Dickey-Fuller = -8.2717, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary
```

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test İlk olarak Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test testini verinin bileşenlerden ayrılmamış haline, toplamsal ve çarpımsal ayrışma modellerine sırasıyla uyguluyarak verinin durağanlığını kontrol ediyoruz. KPSS testini R içinde bulunan `kpss.test` fonksiyonu ile elde edebiliriz.

```
In [13]: kpss.test(data[,2], null="Trend")
          kpss.test(na.omit(additive_model$random), null="Trend")
          kpss.test(na.omit(multiplicative_model$random), null="Trend")
```

```
Warning message in kpss.test(data[, 2], null = "Trend"):
"p-value smaller than printed p-value"
```

KPSS Test for Trend Stationarity

```
data: data[, 2]
KPSS Trend = 1.2645, Truncation lag parameter = 3, p-value = 0.01
```

```
Warning message in kpss.test(na.omit(additive_model$random), null = "Trend"):
"p-value greater than printed p-value"
```

KPSS Test for Trend Stationarity

```
data: na.omit(additive_model$random)
KPSS Trend = 0.0083309, Truncation lag parameter = 3, p-value = 0.1
```

```
Warning message in kpss.test(na.omit(multiplicative_model$random), null = "Trend"):
"p-value greater than printed p-value"
```

KPSS Test for Trend Stationarity

```
data: na.omit(multiplicative_model$random)
KPSS Trend = 0.010033, Truncation lag parameter = 3, p-value = 0.1
```

Phillips–Perron Birim Kök Testi Phillips–Perron Birim Kök Testi testini verinin bileşenlerden ayrılmamış haline, toplamsal ve çarpımsal ayrışma modellerine sırasıyla uyguluyarak verinin durağanlığını kontrol ediyoruz. Phillips–Perron Birim Kök testini R içinde bulunan `pp.test` fonksiyonu ile elde edebiliriz.

```
In [14]: pp.test(data[,2])
          pp.test(na.omit(additive_model$random))
          pp.test(na.omit(multiplicative_model$random))
```

Phillips-Perron Unit Root Test

```
data: data[, 2]
Dickey-Fuller Z(alpha) = -12.85, Truncation lag parameter = 5, p-value
= 0.3898
alternative hypothesis: stationary
```

```
Warning message in pp.test(na.omit(additive_model$random)):
"p-value smaller than printed p-value"
```

Phillips-Perron Unit Root Test

```
data: na.omit(additive_model$random)
Dickey-Fuller Z(alpha) = -89.666, Truncation lag parameter = 5, p-value
= 0.01
alternative hypothesis: stationary
```

```
Warning message in pp.test(na.omit(multiplicative_model$random)):
"p-value smaller than printed p-value"
```

Phillips-Perron Unit Root Test

```
data: na.omit(multiplicative_model$random)
Dickey-Fuller Z(alpha) = -93.661, Truncation lag parameter = 5, p-value
= 0.01
alternative hypothesis: stationary
```

Verinin durağanlığını ölçmek için üç farklı test uyguladık. Bunlar `adf`, `kpss` ve `pp` testleridir. ADF ve PP testlerinin sonuçlarına göre $p = 0.01 < 0.05$ çıkmıştır. Yani yapılan testten oldukça eminiz. Dickey Fuller değerleri de oldukça düşüktür. Verimiz için toplamsal ve çarpımsal model için durağan diyebiliriz. KPSS testini incelediğimizde ise $p = 0.1$ çıkmıştır.

10.5 Arima

R'da olan `auto.arima` fonksiyonu ile verinin kalan sezonsal, trend ve diğer etkenleri kontrol eder ve onlardan arındırır. Veriye `auto.arima` uygulandığında AIC , BIC ve AICc değerlerinin en düşük olanı analiz için daha kabul edilebilir bir değer almaktadır. ARIMA için p,d,q fonksiyonları ile verinin karmaşıklığı fark edilir ve en sade , en kolay model seçilir.

```
In [15]: auto.arima(ts(data[,2]))
         auto.arima(ts(na.omit(additive_model$random)))
         auto.arima(ts(na.omit(multiplicative_model$random)))
```

```
Series: ts(data[, 2])
```

```
ARIMA(2,1,3)
```

```
Coefficients:
```

	ar1	ar2	ma1	ma2	ma3
	-0.1193	0.8104	0.2938	-0.8454	-0.2725
s.e.	0.0761	0.0812	0.0869	0.0789	0.0626

```
sigma^2 estimated as 0.3433: log likelihood=-255.04
```

```
AIC=522.08 AICc=522.37 BIC=544.12
```

```
Series: ts(na.omit(additive_model$random))
```

```
ARIMA(5,0,1) with zero mean
```

```
Coefficients:
```

	ar1	ar2	ar3	ar4	ar5	ma1
	0.0428	0.4151	-0.1535	-0.1207	-0.2632	0.7428
s.e.	0.1316	0.1130	0.0631	0.0614	0.0572	0.1276

```
sigma^2 estimated as 0.2156: log likelihood=-179.43
```

```
AIC=372.86 AICc=373.27 BIC=398.28
```

```
Series: ts(na.omit(multiplicative_model$random))
```

```
ARIMA(2,0,0) with non-zero mean
```

```
Coefficients:
```

	ar1	ar2	mean
	0.8284	-0.2604	0.9939
s.e.	0.0576	0.0575	0.0106

```
sigma^2 estimated as 0.005939: log likelihood=321.52
```

```
AIC=-635.04 AICc=-634.9 BIC=-620.5
```

Verinin hiçbir bileşenden ayrılmadığı durumda `auto.arima` uygulandığında $p = 2$, $d = 1$ ve $q = 3$ çıkmaktadır. $d = 1$ olması verinin durağan olmadığını gösterir.

Verinin toplamsal ayrışma modeline auto.arima uygulandığında p= 5 , d=0 ve q=1 çıkmaktadır. d=0 veri durağandır.

Verilerde çarpımsal modele auto.arima uygulandığında p= 1 , d=0 ve q=1 çıkmaktadır. d=0 veri durağandır.

Hangi ayrışma modelini seçmemiz gerektiğine karar vermek için ilk olarak p,d,q değerlerini karşılaştırmak gerekmektedir. Verinin durağan modeli seçilmesi gerektiğinden hiçbir bileşenden ayrılmadığı model elenir. Toplamsal ve çarpımsal model karşılaştırıldığında çarpımsal modelin daha basit olduğu görülür. İkinci olarak (Seçtiğimiz modelin doğruluğundan emin olmak için) auto.arima nın bize vermiş olduğu AIC,AICc ve BIC değerlerinin en küçük olanı seçilmesi gerekmektedir. Ayrışma modelleri için incelendiğinde çarpımsal modelin daha uygun olduğu görülmektedir. . Formülü :

$$\hat{Y}_{d_t} = 0.5006Y_{t-1}E$$

d=0 olduğunda orijinal seri zaten durağandır ve farkının alınmasına da gerek yoktur.

Veride sezonsal etkinin olup olmadığını kontrol etmek için veriden hiçbir bileşenin çıkarılmadığı haline auto.arima uygulanır.Bunun için arimanın argümanlarından sseasonal.test'ten yararlanılır.

```
In [16]: auto.arima(ts(data[,2]))
          auto.arima(ts(data[,2]) ,seasonal.test = c("ocsb", "ch"))
```

```
Series: ts(data[, 2])
```

```
ARIMA(2,1,3)
```

```
Coefficients:
```

	ar1	ar2	ma1	ma2	ma3
	-0.1193	0.8104	0.2938	-0.8454	-0.2725
s.e.	0.0761	0.0812	0.0869	0.0789	0.0626

```
sigma^2 estimated as 0.3433: log likelihood=-255.04
AIC=522.08 AICc=522.37 BIC=544.12
```

```
Series: ts(data[, 2])
```

```
ARIMA(2,1,3)
```

```
Coefficients:
```

	ar1	ar2	ma1	ma2	ma3
	-0.1193	0.8104	0.2938	-0.8454	-0.2725
s.e.	0.0761	0.0812	0.0869	0.0789	0.0626

```
sigma^2 estimated as 0.3433: log likelihood=-255.04
AIC=522.08 AICc=522.37 BIC=544.12
```

Herhangi bir işlem yapılmayan veriye auto.arima ile sezonsal test uygulandığında ARIMA modelinde; p,d,q değerlerinde ve AIC , AICc ,BIC değerlerinde herhangi bir farklılık görünmemektedir. Bir farklı ifade ile veride sezonsal etki bulunmamaktadır.

Normallik testini seçtiğimiz çarpımsal modele tekrar uygulanır.

```
In [17]: white_noise<-arima(ts(na.omit(multiplicative_model$random)), order=c(1,0,1))
        shapiro.test(white_noise$residuals)
```

Shapiro-Wilk normality test

```
data: white_noise$residuals
W = 0.97796, p-value = 0.0002546
```

Normallik testine göre p değeri oldukça düşük ve w değeri yüksektir. Artık verinin beyaz gürültü olduğunu ve veride herhangi bir bileşen kalmadığını söylenebilir.

10.6 Zaman Serilerinin Trend Bileşeninden Arındırılması (de-trend)

Lineer trend barındıran zaman serilerini trendden arındırmak için genellikle doğrusal regresyonu kullanılmaktadır. Zaman serisine doğrusal regresyon uygulanırsa, trend etkisi içinde olmayan zaman serisi elde edilir.

Doğrusal regresyon dışında da farklı yöntemler kullanılarak zaman serisi verilerini trendlerden arındırılabilir. Örneğin; AR(1), fourier ve wavelet dönüşümleri, kalman filtreleri gibi yöntemler kullanılabilir, ya da forecast kütüphanesi içerisindeki seasonaldummy ve fourier fonksiyonları kullanılarak de-trend yapılabilir.

Model sonucu kalıntılar de-trend edilmiş zaman serisidir.

Doğrusal regresyon analizini gerçekleştiren lm() fonksiyonu ilk olarak bağımlı değişkeni, ardından bağımsız değişkenleri parametre olarak almaktadır. Bağımlı değişkenden sonra tilda (~) işareti, birden fazla bağımsız değişkenimiz varsa değişkenlerin arasına artı (+) işareti konulması unutulmamalıdır.

Verimizi trendsel etkiden arındırmak için hem doğrusal regresyon yöntemini hem de 1. derece otoregresif süreç yöntemi uygulanır.

Trendselsel etkiden arındırıldıktan sonra normallik testi uygulanarak yaptığımız modelin doğruluğundan ve normal olup olmadığına karar verebiliriz.

```
In [18]: trModel <- lm(data[,2]~ c(1:length(data[,2])))
        trResid <- resid(trModel)

        trModel_AR1 <- arima(x=data[,2], order=c(1,0,0))
        trResid_AR1 <- resid(trModel_AR1)
```

```
In [19]: par(mfrow=c(1,2))
        plot(trResid, type="l", col="blue", main="de-trended (regresyon)")
        plot(trResid_AR1, type="l", col="red", main="de-trended (ar(1))")
        shapiro.test(trResid)
        shapiro.test(trResid_AR1)
```

Shapiro-Wilk normality test

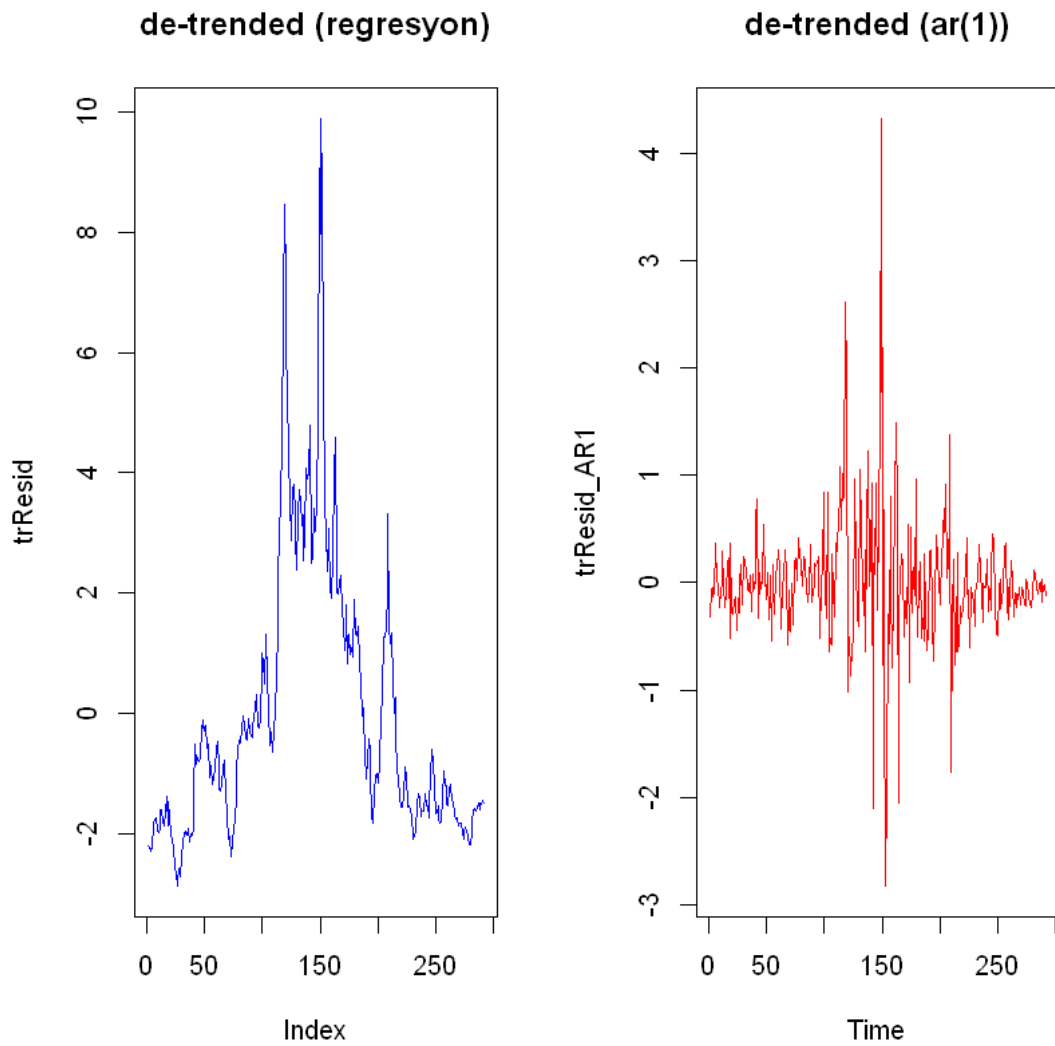
```
data: trResid
```

W = 0.83751, p-value < 2.2e-16

Shapiro-Wilk normality test

data: trResid_AR1

W = 0.82996, p-value < 2.2e-16



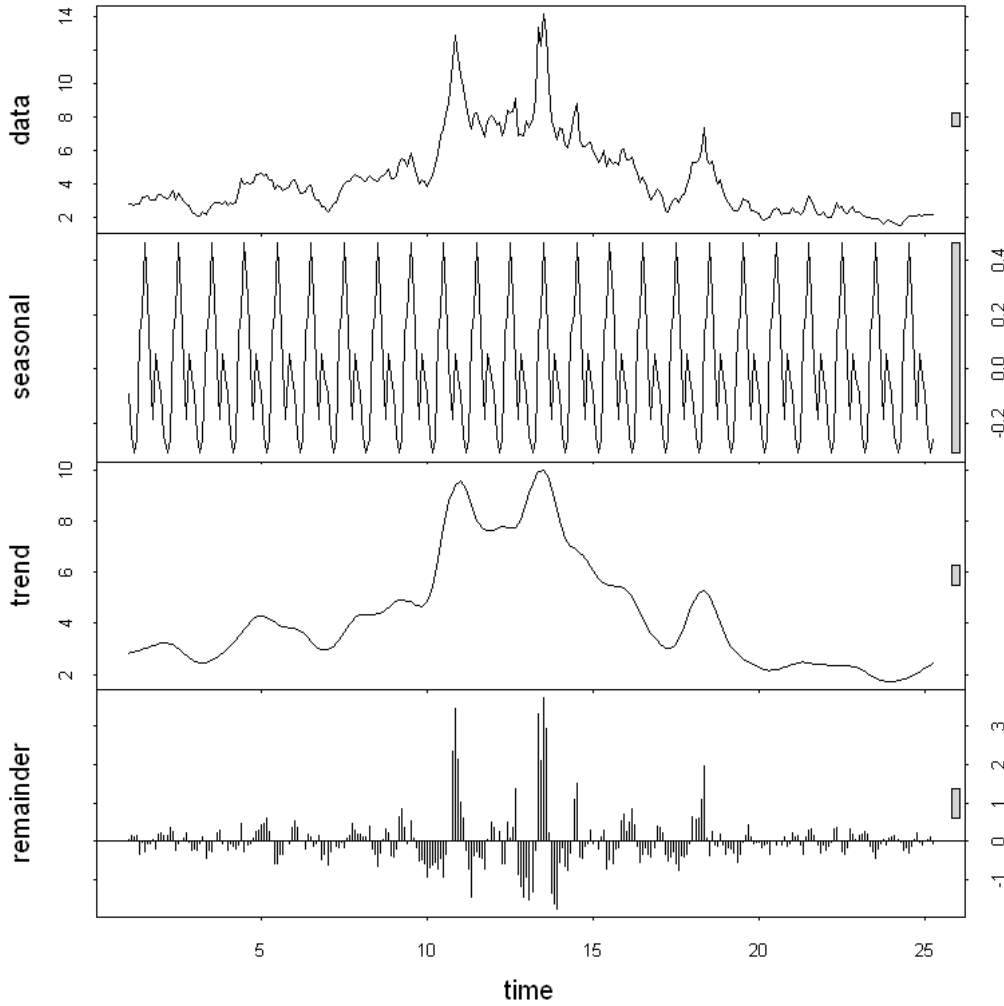
Doğrusal regresyon ve otoregresif süreç karşılaştırıldığında otoregresif sürecin daha tutarlı olduğu , doğrusal regresyonun ise sürekli inişli ve çıkışlı bir durumda olduğu gözlenmektedir.

Normallik testleri açısından karşılaştırsak iki sonuçta da p değerinin güven aralığını sağladığı yani $p < 0.05$ olduğunu ve w değerinin 1 yakın olduğunu görmekteyiz. İki trendsel etkiden arındırılmış veri için normal denilebilir.

10.7 Zaman Serilerinin Mevsimsel Bileşeninden Arındırılması (de-seasonalize)

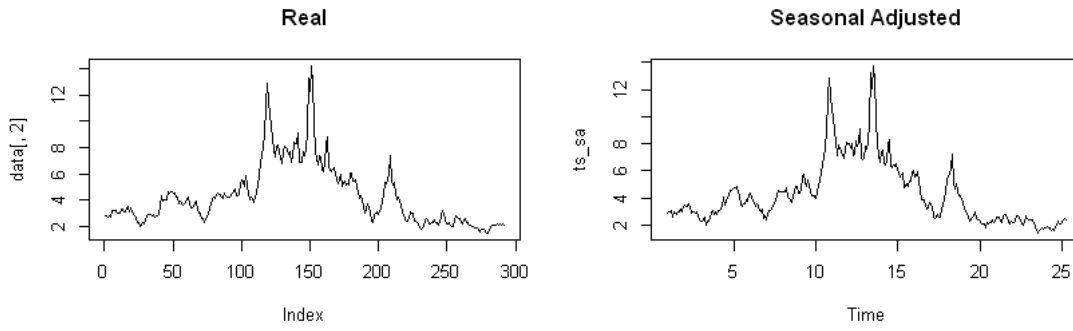
Zaman serisinin mevsimsellik deseni hakkında fikir verir ve mevsimsel etkiler olmadan verilerin modellenmesine yardımcı olur. Zaman serisini mevsimsellikten arındırmak için aşağıdaki iki adım kullanılabilir. 1. Adım: `stl()` fonksiyonu ile zaman serisi ayrılır. 2. Adım: `forecast` kütüphanesi içerisindeki `seasadj()` fonksiyonu kullanılır.

```
In [20]: datastl<- stl(ts(data[,2], freq=12), s.window="periodic")
plot(datastl)
```



```
In [21]: ts_sa <- seasadj(datast1)

In [22]: par(mfrow=c(3,2))
          plot(data[,2], type="l", main="Real")
          plot(ts_sa, type="l", main="Seasonal Adjusted")
```



Normallik testini ölçmek için sezonsal etkisi çıkarılmış veriye shapiro.test uygulanır.

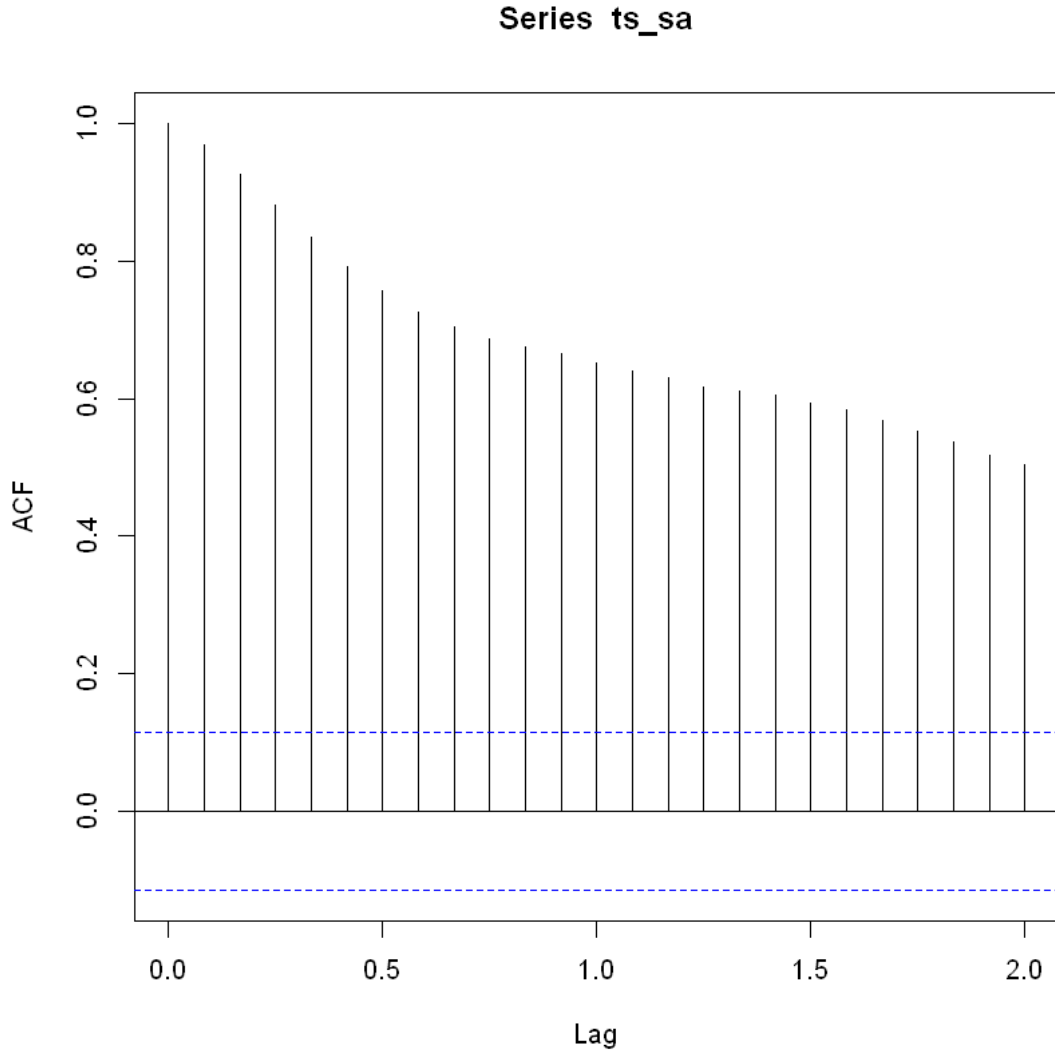
```
In [23]: shapiro.test(ts_sa)
```

Shapiro-Wilk normality test

```
data: ts_sa  
W = 0.86198, p-value = 1.769e-15
```

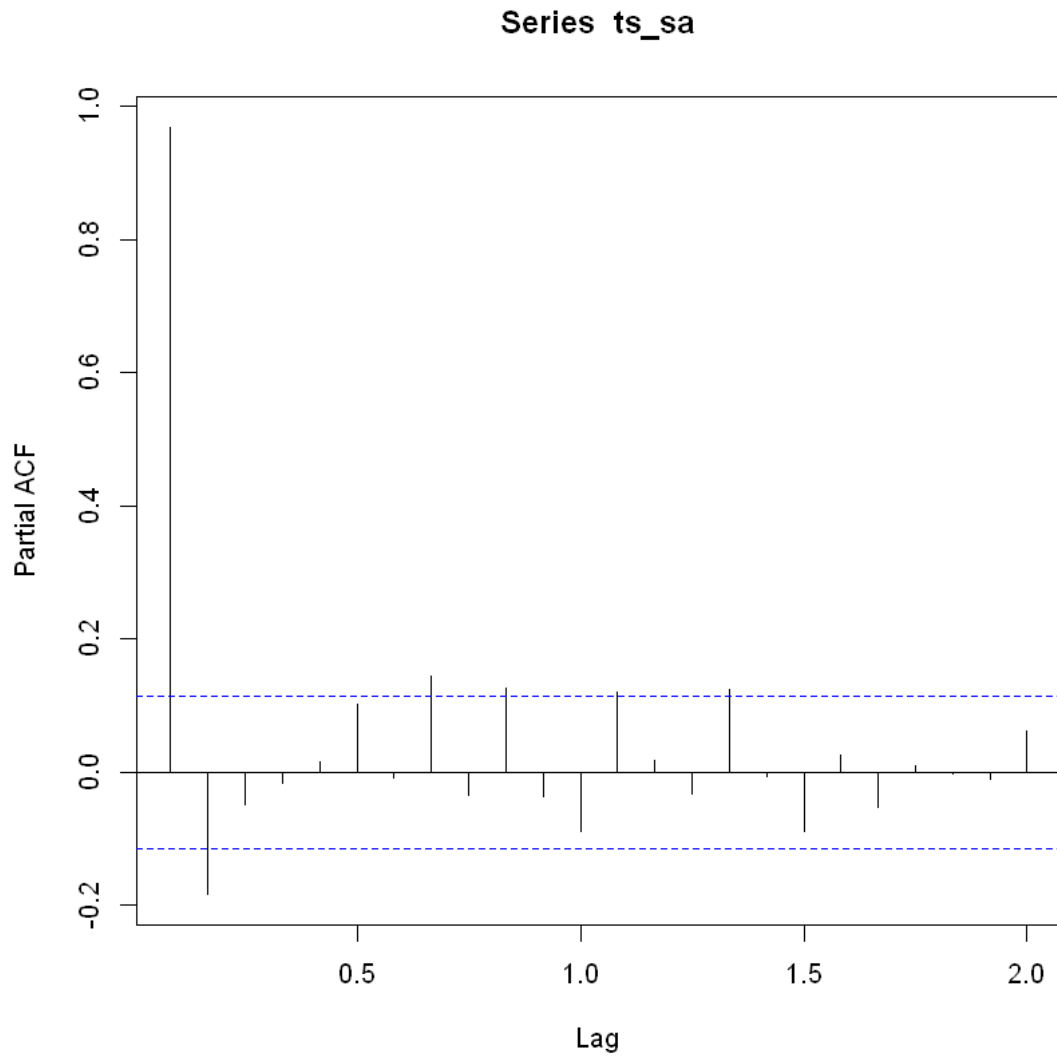
Durağanlık testine göre $p < 0.05$ ve w değeri 1'e yakındı. Bu durum için yapılan sezonsal testin sonucunda doğruluktan eminiz diyebiliriz.

```
In [24]: acf(ts_sa)
```



Otokorelasyon katsayısı incelendiğinde verinin yavaş yavaş azaldığı gözlenmektedir. Bu durumda ts_sa mevsimsel etkiden arındırılmış zaman serimizin durağan olmadığını göstermektedir.

```
In [25]: pacf(ts_sa)
```



Sezonsal etkiden arındırılan verinin (ts_sa) , hala sezonsal etki içerip içermediğini anlamak (mevsimsellik farkını bulmak) için nsdiffs fonksiyonu uygulanır.

```
In [26]: nsdiffs(ts_sa)
```

0

Mevsimsellik farkı sıfırdır. Mevsimsel etki yoktur.

10.8 Natural Gas Futures Zaman Serisi Analizi Özet

Verinin Adı: Natural Gas Futures Contract 3

Verinin Alındığı Yer: Quandl

Verinin Sıklığı: Aylık

Verinin Zaman Aralığı: 31.01.1994- 31.03.2018

Veride iki sütun bulunmaktadır. Sütunlardan birisi zaman, diğeri ise kullanılan değerdir. Amaç zaman serisindeki gözlenen değerlerinin trend, mevsimsel, çevresel ve rastgele etkenlerden ayırmak ve doğru modeli seçmektir. Bu doğrultuda üç yöntem kullandık.

1. Yöntem : R'da bulunan decompose fonksiyonu ile seriyi mevsimsel ve trendsel etkenlerden ayırmaktır. Decompose fonksiyonunu zaman serilerin hep toplamına hem de çarpımına ayrı ayrı uyguladık. Daha net sonuç veren veri seçilir. Toplamsal ayrışma modelini incelediğimizde veride trendsel etkinin sürekli değiştiğini ve mevsimsel etkinin aynı olduğunu görüyoruz. Çarpımsal model de aynı sonuca varıyoruz. İki ayrıştırma modeli arasındaki fark random bileşende ortaya çıkmaktadır. Ancak kullanılan veride toplanan ve çarpılan veride net bir karar veremiyoruz. Verimizin zaman bileşenlerinden ayrıldıktan sonraki normallik durumunu kontrol etmek için veriye normallik testi(shapiro.test) uyguladık, iki modelin de p değerleri güven aralığında ($p < 0.05$) çıktı. Sonuç olarak iki model içinde verimizin bileşenlerden ayrılmış haline normal diyebiliriz. Zaman bileşenlerinden ayrılmış olan verimizin durağanlığını kontrol etmek için adf, kpss ve pp testlerini uyguladık. Üç testin sonucu da verimizin toplamsal ve çarpımsal model için durağan olduğunu gösterdi. Sonraki aşamalarda decompose fonksiyonu ile veremediğimiz hangi modelin uygun olduğu kararını anlamak için arima testini uyguladık. Arima ile gelen p,q,d değerlerinden en basit olanını ve AIC, AICc ve BIC değerlerinden en düşük olanını yani çarpımsal modelin daha uygun olduğuna karar verdik. Ek olarak herhangi bir bileşenden ayrılmamış veriye arima ile sezonsal etkinin olup olmadığını ölçtük. Sonucunda ise herhangi bir farklılık olmadığını yani verimizde sezonsal bir etkinin olmadığını söyleyebiliriz. Decompose ve sonrasında uygulanan auto.arima ile kalan verimize normallik testini tekrar uygularsak güven aralığını sağladığını ($p = 0.0002966 < 0.05$) ve $w = 0.97827$ yani 1 oldukça yakın olduğunu görüyoruz. Bu da verimizin normal olduğunu söyler. Ek olarak artık verimize beyaz gürültü verisi diyebiliriz.
2. Yöntem : Zaman serisini trend bileşenlerinden ayırmak için doğrusal regresyon ve AR(1) modelini kullandık. Verilerin trendsel etkisinden ayrıldığı durumda normallik değerlerini ölçmek için her iki (Doğrusal regresyon, AR(1)) süreç için güven aralığını ($p = 2.2e - 16 < 0.05$ ve $w = 0.83..$) sağladığını gördük.
3. Yöntem : Zaman serisini mevsimsel bileşenlerinden ayırmak için R'da bulunan stl fonksiyonunu kullandık. Verilerin mevsimsel etkisinden ayrıldığı durumda normallik değerlerini ölçmek için normallik (shapiro.test) testini uyguladık. Güven aralığını ($p = 2.098e - 15 < 0.05$ ve $w = 0.86274$) sağladığını gördük. Ancak mevsimsellik etkisinden arındırılmış veri için acf testi uygulandığında verinin yavaş yavaş azaldığı gözlenmektedir. Bu durum için verinin normal olmadığı söylenmektedir.

Ek olarak uyguladığımız nsdiffs fonksiyonu ile birlikte verinin mevsimsel etki içerip içermediğine, ya da kalıp kalmadığına karar verilmektedir. Fonksiyonu uyguladığımızda görülüyor ki veri mevsimsel bileşen içermemektedir.

Yukarıdaki üç yöntemi karşılaştırdığımızda (normalliklerini) kullanmamız gereken modelin çarpımsal ve kullanmamız gereken yöntemin sırasıyla decompose ve arima olmasına karar verdik.

10.9 Forecast

In [27]: `require(forecast)`

10.9.1 Üstel Düzleştirme -Exponential Smooting

Verideki son değişim ve sıçramaları incelenerek tahminlerin ya da öngörülerin devamlı olarak güncelleştirildiği bir yöntemdir. Deterministik ve stokastik trende sahip olan tüm serilere uygulanabilmektedir. Üstel düzeltme yönteminde gelecek dönemin (t+1 döneminin) tahminin hesabında son döneme ait tahmin ile bu tahminden elde edilen hatanın bir kısmı kullanılmaktadır.

$$Y_{t+1} = Y_t + a(e_t)$$

ile ifade edilir.

$$S_t = S_{t-1} + a(Y_t - S_{t-1})$$

S_t : Bir sonraki (gelecek) dönemin, yani (t+1)'inci dönemin tahmini
 S_{t-1} : Son döneme ait tahmin (bu tahmin (t-1)'inci dönemde yapılır.
 Y_t : Son döneme ait gerçek değer
 a : Düzleştirme katsayısı yada ağırlık $0 < a < 1$
 $Y_t - S_{t-1}$: Son döneme ait tahmin hatası

$$Y_{t+1} = S_t = a(Y_{t+1}) + (1 - a)S_{t-1}$$

Gelecek dönemin tahmini=ağırlık.(son dönemin gözlem değeri)+(1-ağırlık).(son döneme ait tahmin) Genel terimini yazmak istenirse:

$$\hat{Y}_{t+1} = S_t = a \sum_{i=1}^t (1 - a)^i \cdot Y_{t-i}$$

Üstel düzeltme yöntemi uyguladığımızda trend ve mevsimsel bileşenler için toplamda 15 metot vardır. - Trendsel Bileşen : N(None) - A(Additive)- A_d (Additive Damped) - M(Multiplicative) - M_d (Multiplicative Damped) - Sezonsal Bileşen: N(None) - A(Additive)- M(Multiplicative)

ETS ile gelen metodta Ets(error, trend , seasonal) sonuçları elde edilir.Örnek olarak birkaç model incelenirse:

ETS(A,A,A) : Holt-Winters' metodunun toplamsal ayrışması ve toplamsal ayrışmadaki hatalar yöntemi

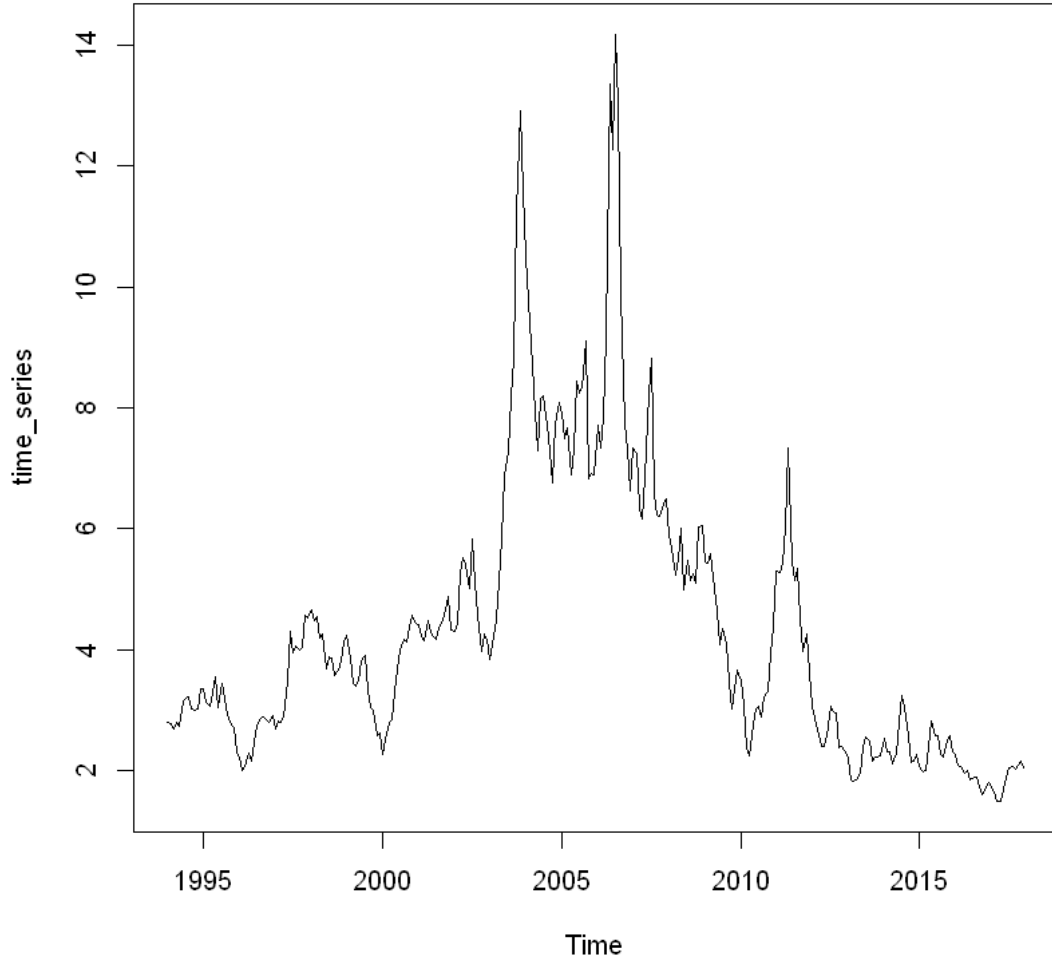
ETS(M,A,M) : Holt-Winters' metodunun çarpımsal ayrışması ve çarpımsal ayrışmadaki hatalar yöntemi

ETS(A, A_d , N) : Toplamsal hata ve sönümlü trendsel etki içerip , sezonsal etki içermeyen yöntem

Üssel düzgünleştirme de 2 farklı yöntem vardır. Bu iki yöntem için de R da bulunan fonksiyonlar kullanılır. - ets() fonksiyonu - HoltWinters() fonksiyonu

İlk olarak verimizin zaman serisi olduğunu gösteriyoruz. Zaman serisi sıklığını aylık olarak tanımlıyoruz. Zaman aralığımızı 01.1994 - 12.2017 kadar olan süreyi alıyoruz. 01.2018-03.2018 süresini de test verisi olarak alıyoruz.

```
In [28]: time_series <-ts(data[,2], start=c(1994, 1), end=c(2017, 12), freq=12)
plot(time_series)
```



Aşağıda üstel düzleştirme yönteminin ets() fonksiyonu kullanılmış halini uygulayacağız.

10.9.2 Öngörü Doğruluğunu Hesaplama - Accuracy Fonksiyonu

Öngörü doğruluğunu hesaplamak için R'in accuracy fonksiyonunu kullanılır. Accuracy fonksiyonu ile gelen verilerde:

ME:Ortalama Hata

$$ME = 1/n \sum_{t=1}^n e_t$$

Bu sonucun dezavantajı,Pozitif ve negatif hata değerlerinin birbirini iptal edebileceğidir, bu yüzden bu ölçü genel uyumun çok iyi bir göstergesi değildir.

MAE : Ortalama Mutlak Hata

$$MAE = 1/n \sum_{t=1}^n |e_t|$$

Bu değer 0 ise, uyum (tahmin) mükemmeldir. Ortalama karesel hata değerine kıyasla, bu uyum ölçüsü aykırı değerleri "vurgulamak", yani benzersiz veya nadir büyük hata değerleri MAE'yi MSE değerinden daha az etkileyecektir.

MSE: Ortalama Hata Kare

$$MSE = 1/n \sum_{t=1}^n e_t^2$$

MPE: Ortalama Yüzde Hata

$$MPE = 1/n \sum_{t=1}^n PE_t$$

Bu değer PE değerlerinin ortalaması olarak hesaplanır.

MAPE: Ortalama Mutlak Yüzde Hata

$$MAPE = 1/n \sum_{t=1}^n |PE_t|$$

e_t : t dönemindeki hata

PE_t : t dönemindeki yüzde hata

$PE_t = [(Y_t - F_t) / Y_t] \cdot 100$

n: dönem sayısı

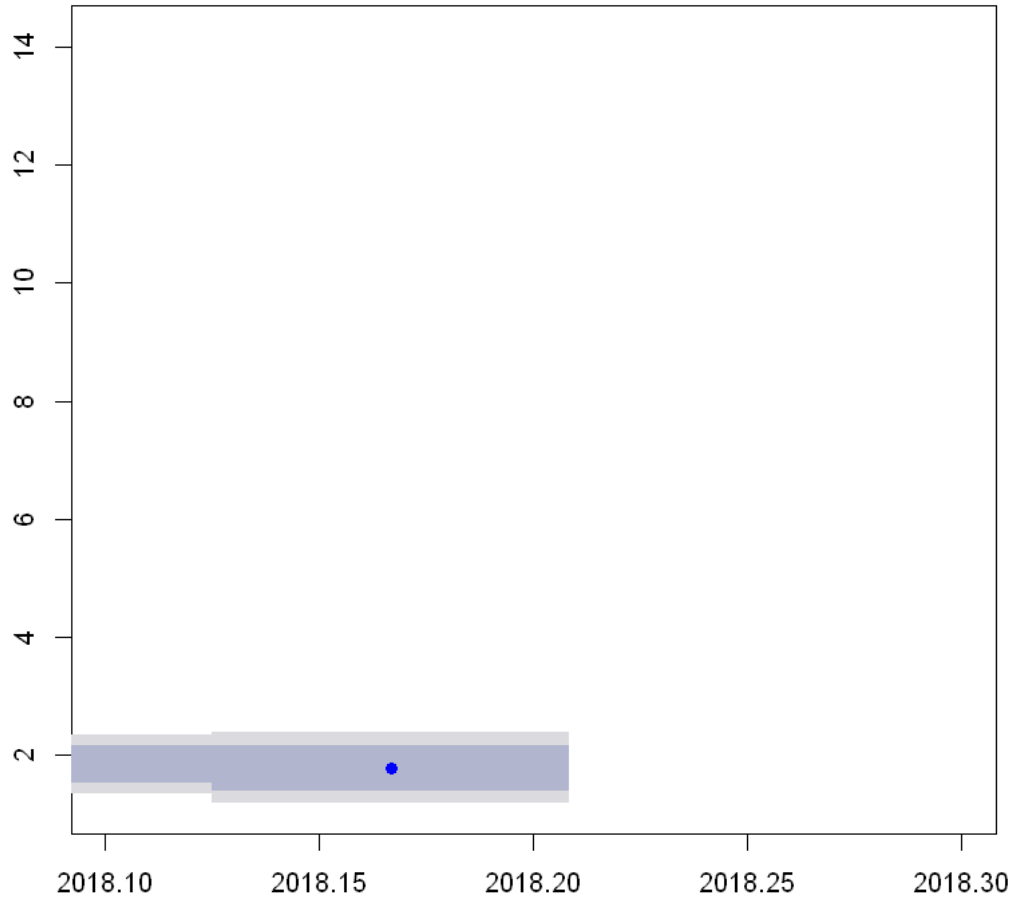
Öngörü uygulandığında çıkan değerlerde 80 ve 95 güven aralığını belli etmektedir. Gri olan bölgeler öngöründen %95 emin olduğumuzu , mavi bölge ise öngöründen %80 emin olduğumuzu göstermektedir.

Öngörü modellerinden ilk olarak ets modelini deniyoruz. Test modülümüzü 2018.01 ve 2018.03 arasına uyguluyoruz.

```
In [29]: model11 <-ets(time_series)
          data_forecast_ets <- forecast(model11, h=3)
          accuracy(data_forecast_ets)
          plot(data_forecast_ets, ,xlim=c(2018.1,2018.3))
          data_forecast_ets
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.001786263	0.5867771	0.3525912	-0.490289	7.254289	0.2780497	0.1595532
	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95		
Jan 2018	1.974613	1.739441	2.209785	1.614948	2.334278		
Feb 2018	1.846689	1.522929	2.170450	1.351541	2.341838		
Mar 2018	1.787294	1.397092	2.177495	1.190532	2.384055		

Forecasts from ETS(M,Ad,A)



Verimize ets() fonksiyonu ile öngörü hesapladığımızda bize ETS(M, A_d ,M) argümanlarını vermiştir. Bu da verimizde çarpımsal hatanın, sönümlü trendin ve çarpımsal mevsimsel etkinin sahip olduğunu gösteriyor. Çıkan sonuçları orjinal değerler ile karşılaştırıldığında :

Orjinal değerler:

- 2018.03 2.784
- 2018.02 2.695
- 2018.01 2.800

Öngörü hesabındaki güven aralığı %95 olan noktalardaki sonuçlar ise:

- 2018.03 2.660738
- 2018.02 2.514919
- 2018.01 2.431952

olarak hesaplanmıştır. İki değer arasındaki farklar incelenirse : - 2018.03 0.089 - 2018.02 0.18001 - 2018.01 0.368048

olarak hesaplanmıştır.

Düşük bir α değeri, geçmiş ortalamaya daha fazla ağırlık verir ve rassal dalgalanmaların etkisini azaltır. Yüksek α değeri ise talepteki değişikliklere daha fazla tepki verilmesini sağlar.

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

```
Forecast method: ETS(M,Ad,A)
```

```
Model Information:
```

```
ETS(M,Ad,A)
```

```
Call:
```

```
ets(y = time_series)
```

```
Smoothing parameters:
```

```
alpha = 0.9999
```

```
beta = 0.0066
```

```
gamma = 1e-04
```

```
phi = 0.9255
```

```
Initial states:
```

```
l = 2.6888
```

```
b = -0.0089
```

```
s=0.0223 0.045 -0.051 0.0241 0.1746 0.2976
```

```
0.1571 -0.0229 -0.2054 -0.2297 -0.1699 -0.0417
```

```
sigma: 0.0929
```

```
      AIC      AICc      BIC  
1071.518 1074.061 1137.452
```

```
Error measures:
```

```
              ME      RMSE      MAE      MPE      MAPE      MASE  
Training set -0.001786263 0.5867771 0.3525912 -0.490289 7.254289 0.2780497
```

```
ACF1
```

```
Training set 0.1595532
```

```
Forecasts:
```

```
      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95  
Jan 2018      1.974613 1.739441 2.209785 1.614948 2.334278  
Feb 2018      1.846689 1.522929 2.170450 1.351541 2.341838  
Mar 2018      1.787294 1.397092 2.177495 1.190532 2.384055
```

ETS(M, A_d , M) için α değeri 0.988 çıkmıştır. Bu da talepteki değişikliklere daha fazla tepki vereceğini söylemektedir.

10.9.3 Basit Üstel Düzleştirme Yöntemi (Durağan Seriler için):

Bu yöntem trende ve mevsimsel dalgalanmaya sahip olmayan sadece bir ortalama düzey etrafında hareket eden serilerin analizinde uygulanmaktadır. Bu yöntemde serinin tahminin elde edilebilmesi için aşağıdaki formülünden yararlanılmaktadır.

Düzleştirme katsayısı(α) tahmin hatasını en küçük yapan sabit bir değerdir. Düzleştirme katsayısı (α) 1 değerine yakın ise son gözlemlerin değerleri doğrudan serinin tahminini etkiler ve bu durum tahmin serisinde aşırı sıçramalara neden olabilir. Düzleştirme katsayısının 0 değerine yakın olması durumunda orijinal serideki değişimler tahmin serisini pek etkileyemez. Tahmin serisindeki ilk verilerle son veriler arasında önemli bir fark olmaz. Bu iki durum da tahminlerin güvenilir olması bakımından istenilen bir özellik değildir.

Aşağıda üstel düzleştirme yönteminin HoltWinters() fonksiyonu kullanılmış halini uygulayacağız. Bu model trend, sezonsal bileşenleri de içermemediği durumdur.

```
In [31]: model12 <- HoltWinters(time_series, beta=FALSE, gamma=FALSE)
        model12
        data_forecast_simple <- forecast(model12, h=3)
        accuracy(data_forecast_simple)
        plot(data_forecast_simple, ,xlim=c(2018.1,2018.3))
        data_forecast_simple
```

Holt-Winters exponential smoothing without trend and without seasonal component.

Call:

```
HoltWinters(x = time_series, beta = FALSE, gamma = FALSE)
```

Smoothing parameters:

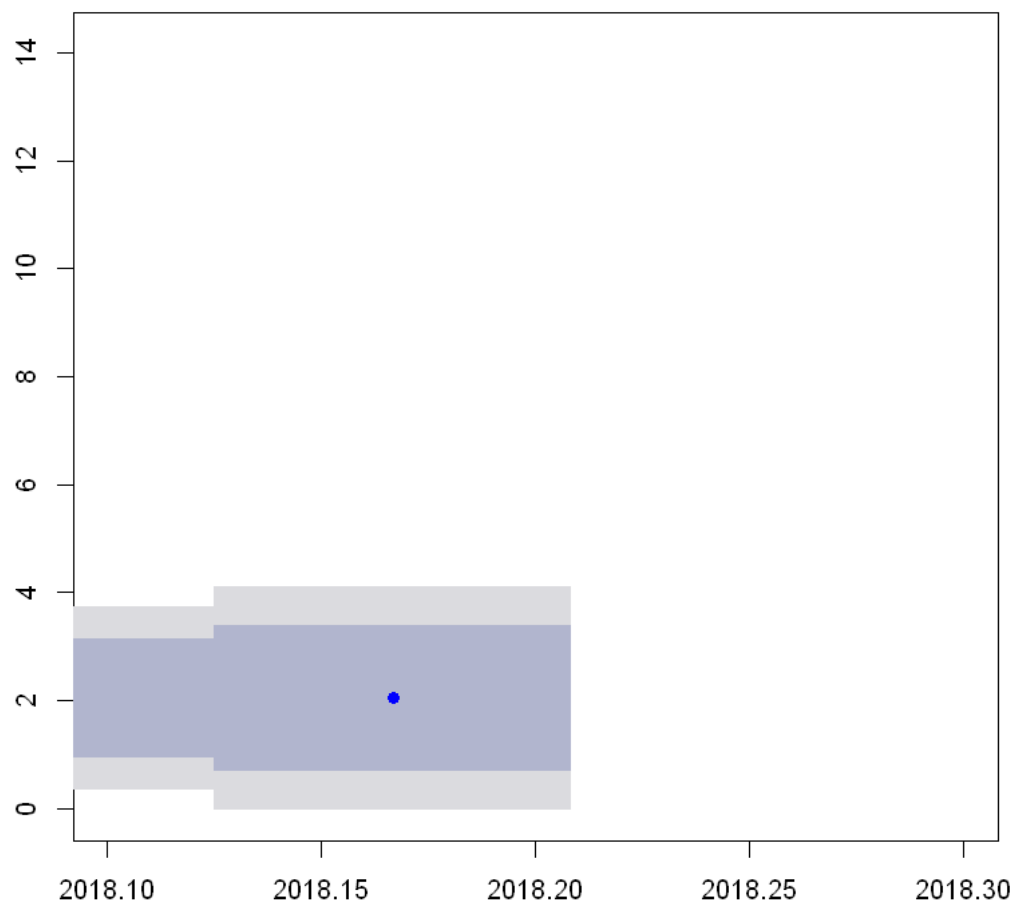
```
alpha: 0.9999286
beta : FALSE
gamma: FALSE
```

Coefficients:

```
[,1]
a 2.038008
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.002665668	0.6110392	0.3739591	-0.6182691	7.84757	0.2949002	0.1759845
	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95		
Jan 2018	2.038008	1.2535693	2.822447	0.8383126	3.237703		
Feb 2018	2.038008	0.9286838	3.147332	0.3414431	3.734573		
Mar 2018	2.038008	0.6793851	3.396631	-0.0398264	4.115842		

Forecasts from HoltWinters



```
In [32]: summary(data_forecast_simple)
```

Forecast method: HoltWinters

Model Information:

Holt-Winters exponential smoothing without trend and without seasonal component.

Call:

```
HoltWinters(x = time_series, beta = FALSE, gamma = FALSE)
```

Smoothing parameters:

alpha: 0.9999286


```
beta : FALSE
gamma: FALSE
```

```
Coefficients:
      [,1]
a 2.038008
```

```
Error measures:
```

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	-0.002665668	0.6110392	0.3739591	-0.6182691	7.84757	0.2949002

```
ACF1
Training set 0.1759845
```

```
Forecasts:
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2018	2.038008	1.2535693	2.822447	0.8383126	3.237703
Feb 2018	2.038008	0.9286838	3.147332	0.3414431	3.734573
Mar 2018	2.038008	0.6793851	3.396631	-0.0398264	4.115842

Düzleştirme katsayı değeri arttıkça daha büyük tahmin farklılıkları oluşmaktadır. Basit üstel düzleştirme yöntemi uyguladığımızda gelen alpha değeri 0.9999435’dir. Bu sonuçta 1’e çok yakın ve değerın çok büyük olduğunu, bileşenlere yüksek tepki vereceğini gösterir. Hata oranı fazladır. O yüzden öngörü yaparken bu model seçilmemelidir.

10.9.4 Holt Üstel Düzleştirme Yöntemi: (Doğrusal Trendi Olan Seriler için)

Trende sahip mevsimsel dalgalanması olmayan serilerin tahmin işleminde Holt üstel düzleştirme yöntemi kullanılmaktadır.

```
In [33]: model13 <- HoltWinters(time_series, gamma=FALSE)
model13
data_forecast_holt <- forecast(model13, h=3)
accuracy(data_forecast_holt)
plot(data_forecast_holt, ,xlim=c(2018.1,2018.3))
data_forecast_holt
```

Holt-Winters exponential smoothing with trend and without seasonal component.

```
Call:
```

```
HoltWinters(x = time_series, gamma = FALSE)
```

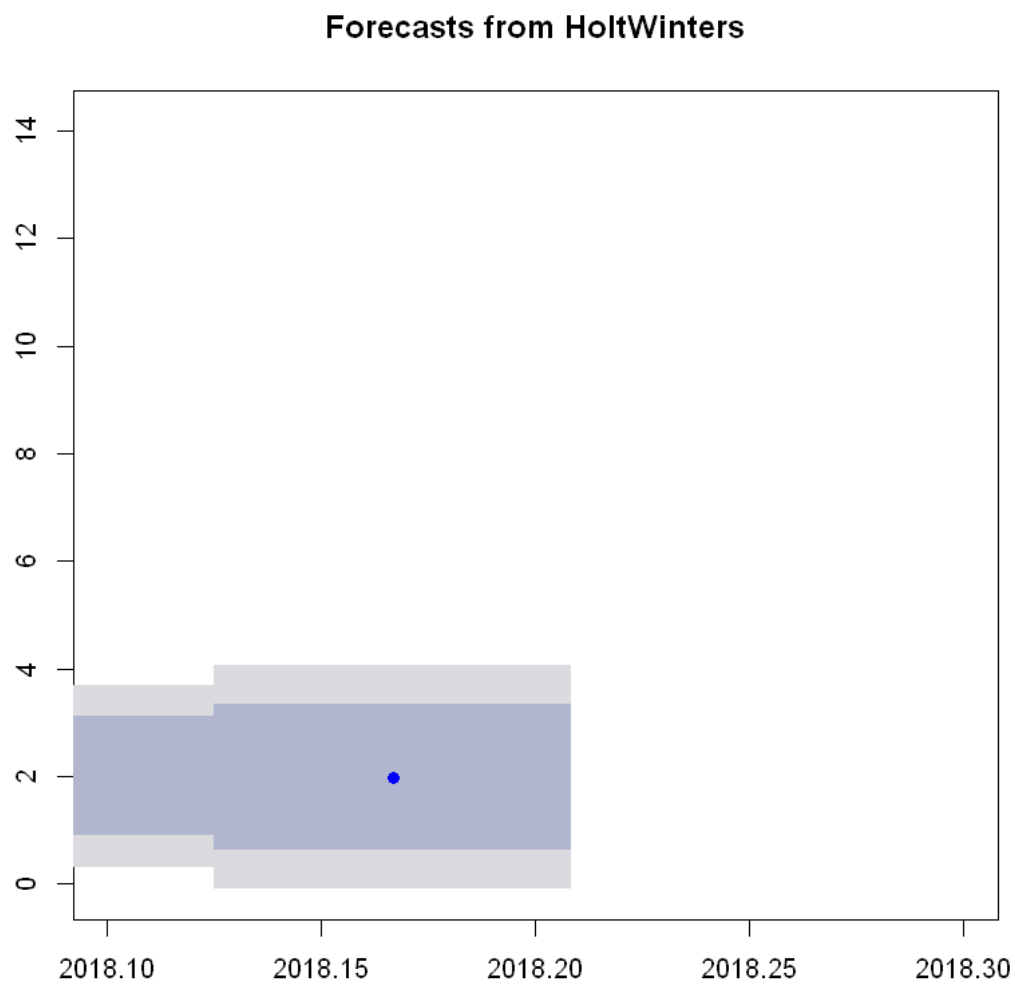
```
Smoothing parameters:
```

```
alpha: 1
beta : 0
gamma: FALSE
```

```
Coefficients:
      [,1]
```

a 2.038
b -0.019

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.01639161	0.6123117	0.3755175	-0.06358456	7.867708	0.2961291	0.1759017
	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95		
Jan 2018	2.019	1.2331972	2.804803	0.8172183	3.220782		
Feb 2018	2.000	0.8887070	3.111293	0.3004241	3.699576		
Mar 2018	1.981	0.6199496	3.342050	-0.1005469	4.062547		



```
In [34]: summary(data_forecast_holt)
```

Forecast method: HoltWinters

Model Information:

Holt-Winters exponential smoothing with trend and without seasonal component.

Call:

```
HoltWinters(x = time_series, gamma = FALSE)
```

Smoothing parameters:

alpha: 1

beta : 0

gamma: FALSE

Coefficients:

[,1]

a 2.038

b -0.019

Error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	0.01639161	0.6123117	0.3755175	-0.06358456	7.867708	0.2961291

ACF1

Training set 0.1759017

Forecasts:

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2018	2.019	1.2331972	2.804803	0.8172183	3.220782
Feb 2018	2.000	0.8887070	3.111293	0.3004241	3.699576
Mar 2018	1.981	0.6199496	3.342050	-0.1005469	4.062547

Düzleştirme katsayı değeri arttıkça daha büyük tahmin farklılıkları oluşmaktadır. Basit üstel düzleştirme yöntemi uyguladığımızda gelen alpha değeri 1'dir. Bu sonuçta değerin çok büyük olduğunu gösterir. Hata oranı fazladır. O yüzden öngörü yaparken bu model seçilmemelidir.

10.9.5 Holt-Winters Yöntemi: (Doğrusal Trendi ve Mevsimlik Etkileri Olan Seriler İçin)

Holt-Winters Yöntemi üç temel düzeltme esitliğine sahiptir. Temel, trend ve mevsimsellik için olan bu üç düzeltme sabiti ile yapılan düzeltmeler, Holt'un Doğrusal Yöntemi'ne benzer olarak tek toplam esitlikte toplanarak mevsimsellik esitliği elde etmektedir.

```
In [35]: model14 <-HoltWinters(time_series)
model14
data_forecast_holtWinter <- forecast(model14, h=3)
accuracy(data_forecast_holtWinter)
plot(data_forecast_holtWinter, ,xlim=c(2018.1,2018.3))
data_forecast_holtWinter
```

Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:

```
HoltWinters(x = time_series)
```

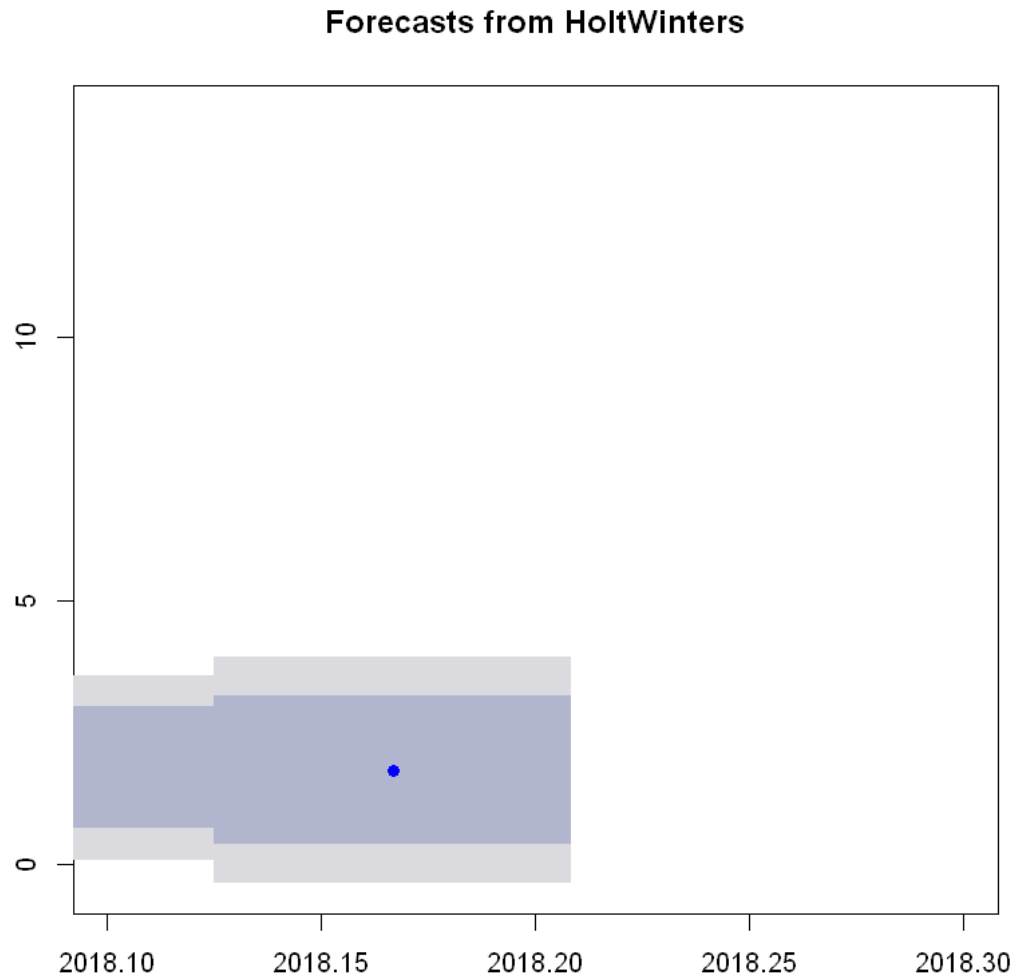
Smoothing parameters:

```
alpha: 1
beta : 0
gamma: 0
```

Coefficients:

```
      [,1]
a    1.93268403
b     0.01312719
s1    0.09744097
s2   -0.11518403
s3   -0.18280903
s4    0.05598264
s5    0.32977431
s6   -0.11422569
s7    0.14994097
s8    0.13856597
s9   -0.09776736
s10  -0.16885069
s11  -0.19818403
s12   0.10531597
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.01718173	0.6298921	0.3989148	-1.049968	8.759911	0.31458	0.1432632
	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95		
Jan 2018	2.043252	1.2348476	2.851657	0.80690399	3.279600		
Feb 2018	1.843754	0.7004976	2.987011	0.09529398	3.592215		
Mar 2018	1.789257	0.3890587	3.189454	-0.35216134	3.930674		



Düzleştirme katsayı değeri arttıkça daha büyük tahmin farklılıkları oluşmaktadır. Basit üstel düzleştirme yöntemi uyguladığımızda gelen alpha değeri 1'dir. Bu sonuçta değerin çok büyük olduğunu gösterir. Hata oranı fazladır. O yüzden öngörü yaparken bu model seçilmemelidir.

```
In [36]: summary(data_forecast_holtWinter)
```

```
Forecast method: HoltWinters
```

```
Model Information:
```

```
Holt-Winters exponential smoothing with trend and additive seasonal component.
```

```
Call:
```

```
HoltWinters(x = time_series)
```

Smoothing parameters:

alpha: 1
beta : 0
gamma: 0

Coefficients:

```
      [,1]  
a      1.93268403  
b      0.01312719  
s1     0.09744097  
s2    -0.11518403  
s3    -0.18280903  
s4     0.05598264  
s5     0.32977431  
s6    -0.11422569  
s7     0.14994097  
s8     0.13856597  
s9    -0.09776736  
s10   -0.16885069  
s11   -0.19818403  
s12    0.10531597
```

Error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	-0.01718173	0.6298921	0.3989148	-1.049968	8.759911	0.31458

ACF1

Training set 0.1432632

Forecasts:

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2018	2.043252	1.2348476	2.851657	0.80690399	3.279600
Feb 2018	1.843754	0.7004976	2.987011	0.09529398	3.592215
Mar 2018	1.789257	0.3890587	3.189454	-0.35216134	3.930674

10.9.6 Arima Modeli

Zaman serilerinde öngörü analizi yapmanın bir diğer yolu da arima modeli uygulamaktır.

```
In [37]: time_series2<-ts(data[,2], start=c(1994, 1), end=c(2017, 12), freq=12)
```

```
In [38]: model21 <- arima(time_series2, order=c(1,0,1))  
model21  
data_forecast_arima <- forecast(model21, h=3)  
accuracy(data_forecast_arima)  
plot(data_forecast_arima, ,xlim=c(2018.1,2018.3))  
data_forecast_arima
```

```
Call:
arima(x = time_series2, order = c(1, 0, 1))
```

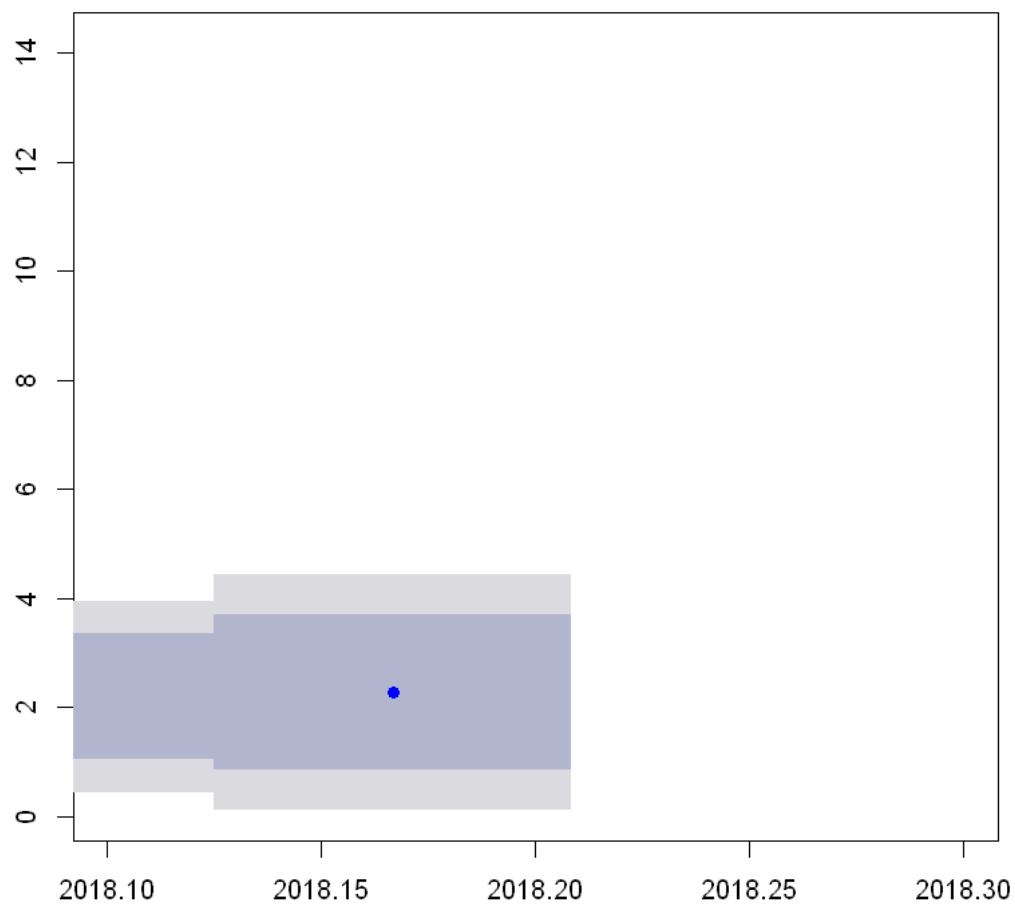
Coefficients:

```
      ar1      ma1  intercept
      0.9542 0.1786      4.0934
s.e.  0.0173 0.0550      0.8477
```

```
sigma^2 estimated as 0.3541:  log likelihood = -260.53,  aic = 529.07
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.006323252	0.5950555	0.3656777	-1.272849	7.823498	0.2883696	0.01722624
Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95			
Jan 2018	2.096852	1.3342576	2.859446	0.9305645	3.263139		
Feb 2018	2.188226	1.0358981	3.340553	0.4258927	3.950558		
Mar 2018	2.275417	0.8585829	3.692252	0.1085561	4.442279		

Forecasts from ARIMA(1,0,1) with non-zero mean



```
In [39]: summary(data_forecast_arima)
```

Forecast method: ARIMA(1,0,1) with non-zero mean

Model Information:

Call:

```
arima(x = time_series2, order = c(1, 0, 1))
```

Coefficients:

ar1	ma1	intercept
0.9542	0.1786	4.0934

s.e. 0.0173 0.0550 0.8477

sigma^2 estimated as 0.3541: log likelihood = -260.53, aic = 529.07

Error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	0.006323252	0.5950555	0.3656777	-1.272849	7.823498	0.2883696
	ACF1					
Training set	0.01722624					

Forecasts:

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2018	2.096852	1.3342576	2.859446	0.9305645	3.263139
Feb 2018	2.188226	1.0358981	3.340553	0.4258927	3.950558
Mar 2018	2.275417	0.8585829	3.692252	0.1085561	4.442279

Aşağıda genel olarak bütün doğrulukları göstereceğiz.Hatırlamak için
ME:Ortalama Hata
MAE : Ortalama Mutlak Hata
MSE: Ortalama Hata Kare
MPE: Ortalama Yüzde Hata
MAPE: Ortalama Mutlak Yüzde Hata

In [40]: accuracy(data_forecast_ets)
accuracy(data_forecast_simple)
accuracy(data_forecast_holt)
accuracy(data_forecast_holtWinter)
accuracy(data_forecast_arima)

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.001786263	0.5867771	0.3525912	-0.490289	7.254289	0.2780497	0.1595532
	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.002665668	0.6110392	0.3739591	-0.6182691	7.84757	0.2949002	0.1759845
	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.01639161	0.6123117	0.3755175	-0.06358456	7.867708	0.2961291	0.1759017
	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.01718173	0.6298921	0.3989148	-1.049968	8.759911	0.31458	0.1432632
	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.006323252	0.5950555	0.3656777	-1.272849	7.823498	0.2883696	0.01722624

Yapılan 5 farklı öngörü modelinin hata değerlerini karşılaştırmak için literatürde en çok kullanılan ve tercih edilen Ortalama Mutlak Hata (MAE) değerini kullanılırız. Bu karşılaştırmaya göre ets() yöntemi en uygun yöntemdir.

Not: Hatırlarsak alpha değerleri ets() modülünde diğer modüllere göre daha düşüktür. Yani,ets() modülünde koşullara verdiği tepki, diğer modellerde verdiği tepkiden daha az, daha tutarlıdır.

10.9.7 Natural Gas Futures - Zaman Serilerinde Öngörü Özeti

Verimizde öngörü yapabilmek için tahminleme verisi ve test verisi olmak üzere iki veri seti oluşturduk. Tahminleme verisini Ocak 1994 ile Aralık 2017 arasını, test verisi için son üç ayı yani Ocak 2018 ile Mart 2018 arasında aldık. Sırasıyla ets, holtwinters ve arima modellerini uyguladık. Hata terimleri ve alpha değerleri incelendiğinde ets() modelinin öngörü için daha uygun olduğuna karar verdik.

İlk verimiz ile ilgili analiz ve öngörü hesaplamalarını yapmış olduk. Şimdi ikinci verimiz olan petrol verilerini inceleyeceğiz.

11 Florida Field Production of Crude Oil

Verinin Yayıncısı: U.S. Energy Information Administration Data

ABD ulusal ve devlet verilerinin; elektrik, kömür, doğal gaz ve petrol gibi tüm ana enerji ürünleri üzerindeki üretim, tüketim ve diğer göstergelere ilişkisini göstermektedir.

Verinin Adı: Florida Field Production of Crude Oil

Florida'nın ham petrol üretiminin göstergelere ilişkisini göstermektedir. Veri zaman serisi olarak ilerlemektedir.

Verinin Sıklığı: Aylık

Verinin Zaman Aralığı: 31.01.1981- 28.02.2018

Verinin Alındığı Yer: Quandl

Quandl yatırım profesyonellerine hizmet veren finansal, ekonomik ve alternatif veriler için bir platformdur. Verileri 500'den fazla yayıncıdan sağlar. Quandl'ın verilerine bir API aracılığıyla erişilebilir. Bir çok programlama dili için API erişimi (R, Python, Matlab, Maple ve Stata) vardır.

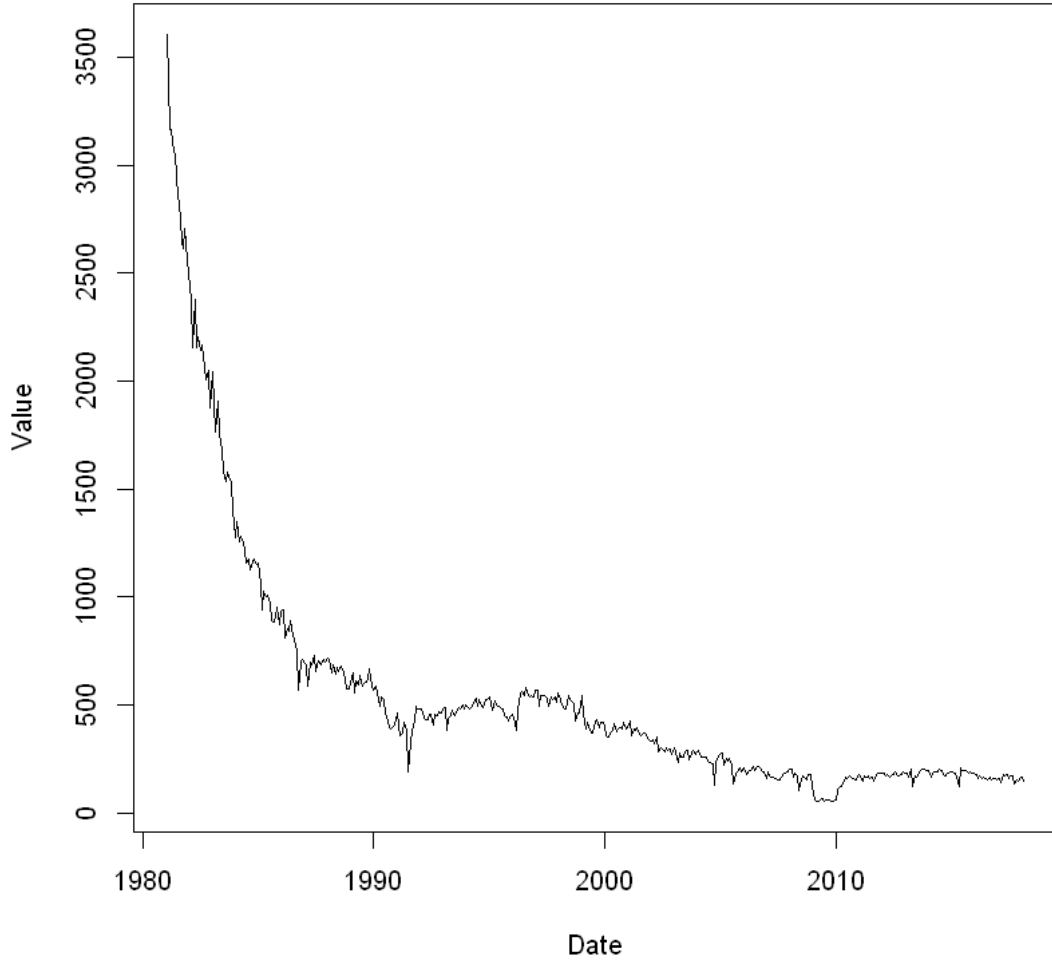
Excel eklentisi, stok fiyat bilgisi dahil olmak üzere verilere erişim sağlar.

```
In [41]: data2<-Quandl("EIA/PET_MCRFPFL1_M", api_key="BtbwkANCR4aHKBscujRb", collapse="monthly")
```

Veriyi quandl veri sitesinden alıyoruz. Veri iki sütundan oluşmaktadır. İlk sütun tarih ikinci sütun ise üretim miktarıdır.

Aşağıda verimizi çizgizel grafik gösterimini görüyoruz.

```
In [42]: rownames(data2) <- data2[,1]
plot(data2,type="l")
```



Grafik incelendiğinde petrol üretimi gittikçe azaldığı fark ediliyor.

11.1 Decompose- Geleneksel Zaman Serisi Ayırışım Yöntemleri

Veriyi sezonsal veriden ayırmak için R paketinin decompose fonksiyonunu kullanıyoruz. Decompose fonksiyonu ile veri sezonsal etkiden ve trend etkisinden arınmaktadır. Veri aylık olduğundan dolayı sıklığı 12 olarak alıyoruz.

Ek olarak verinin ayırışma yönteminin, toplamsal ya da çarpımsal olduğunu anlamak için `type= c("additive")` ve `type= c("multiplicative")` fonksiyonu kullanıyoruz.

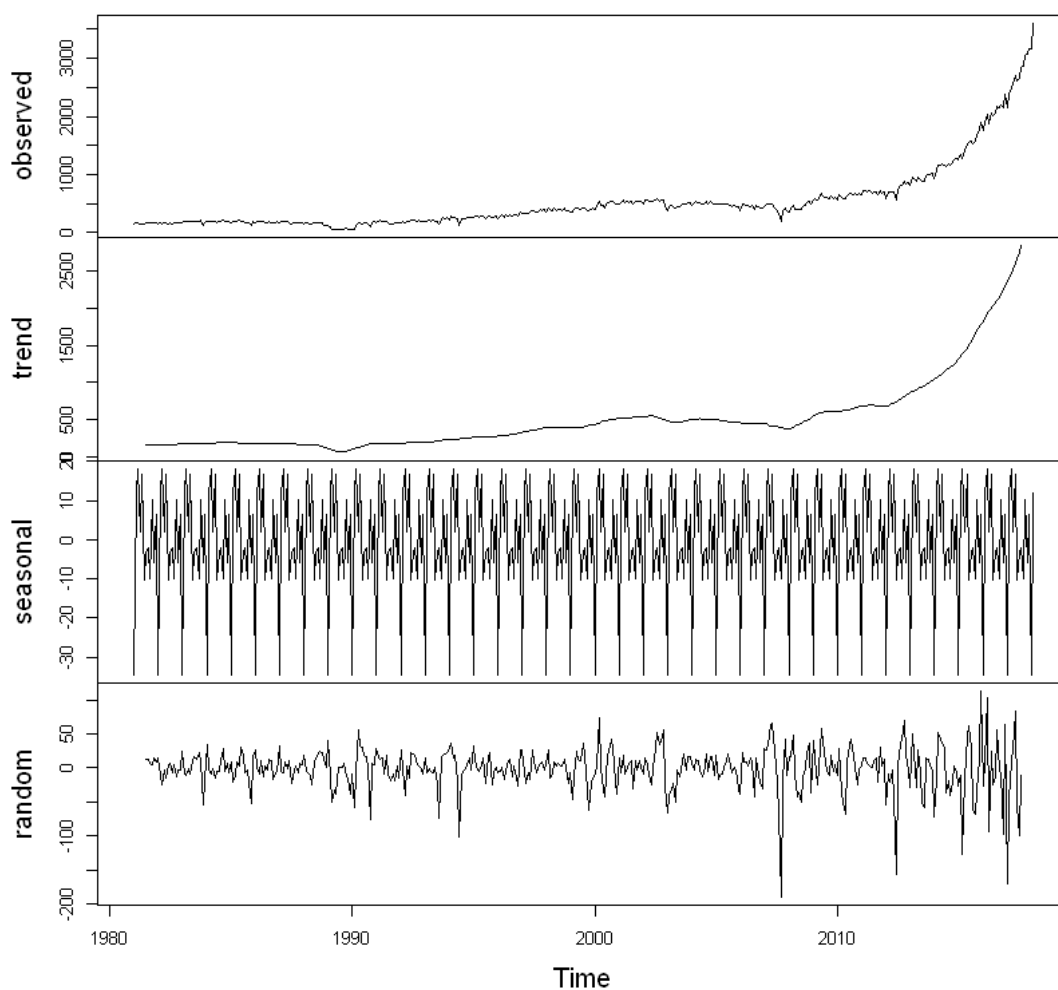
Dönemsel mevsimselliği incelemek için ise R 'ın figure fonksiyonu kullanılmaktadır.

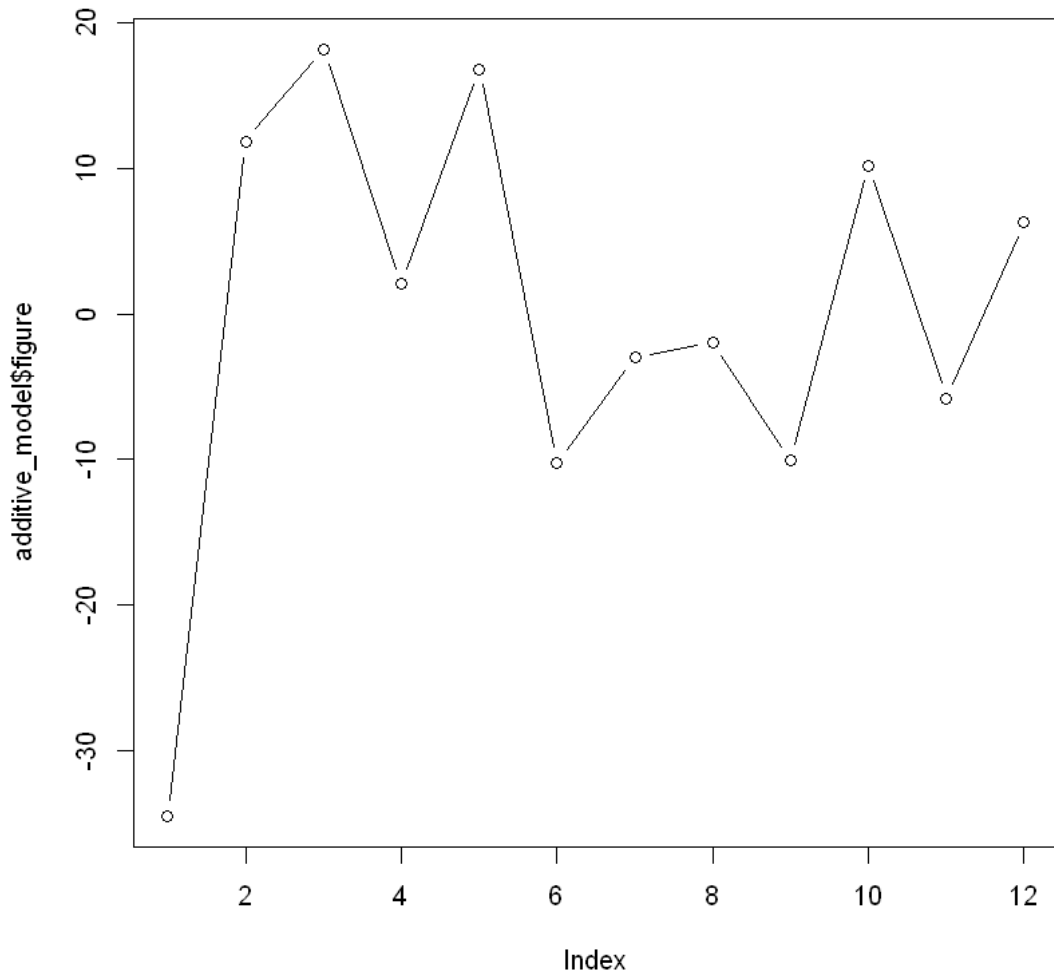
Aşağıda zaman serisi (ts) verimize toplamsal ayırışım tipine decompose fonksiyonu uygulanıyor. Böylelikle verimizdeki sezonsal ve trendsel etkiden çıkartmış oluyoruz. Plot fonksiyonu ile yaptığımız modeli grafiksel olarak gösteriyoruz. Figure fonksiyonu ile de zaman serimizin 12 aylık sezonsal etkisini gösteriyoruz.

```
In [43]: additive_model<-decompose(ts(data2[,2], start = c(1981, 1), end = c(2018, 2),freq=12),t
summary(additive_model)
plot(additive_model)
plot(additive_model$figure,type="b")
```

	Length	Class	Mode
x	446	ts	numeric
seasonal	446	ts	numeric
trend	446	ts	numeric
random	446	ts	numeric
figure	12	-none-	numeric
type	1	-none-	character

Decomposition of additive time series





Toplamsal ayrışma modelini incelediğimizde trend bileşeninin 1990 yılında en düşük etkide bulunduğunu ancak sonrasında sürekli arttığı gözlenmektedir. Sezonluk etkiyi incelersek veri bütün yıllarda aynı sezonluk değerleri aldığını görüyoruz.

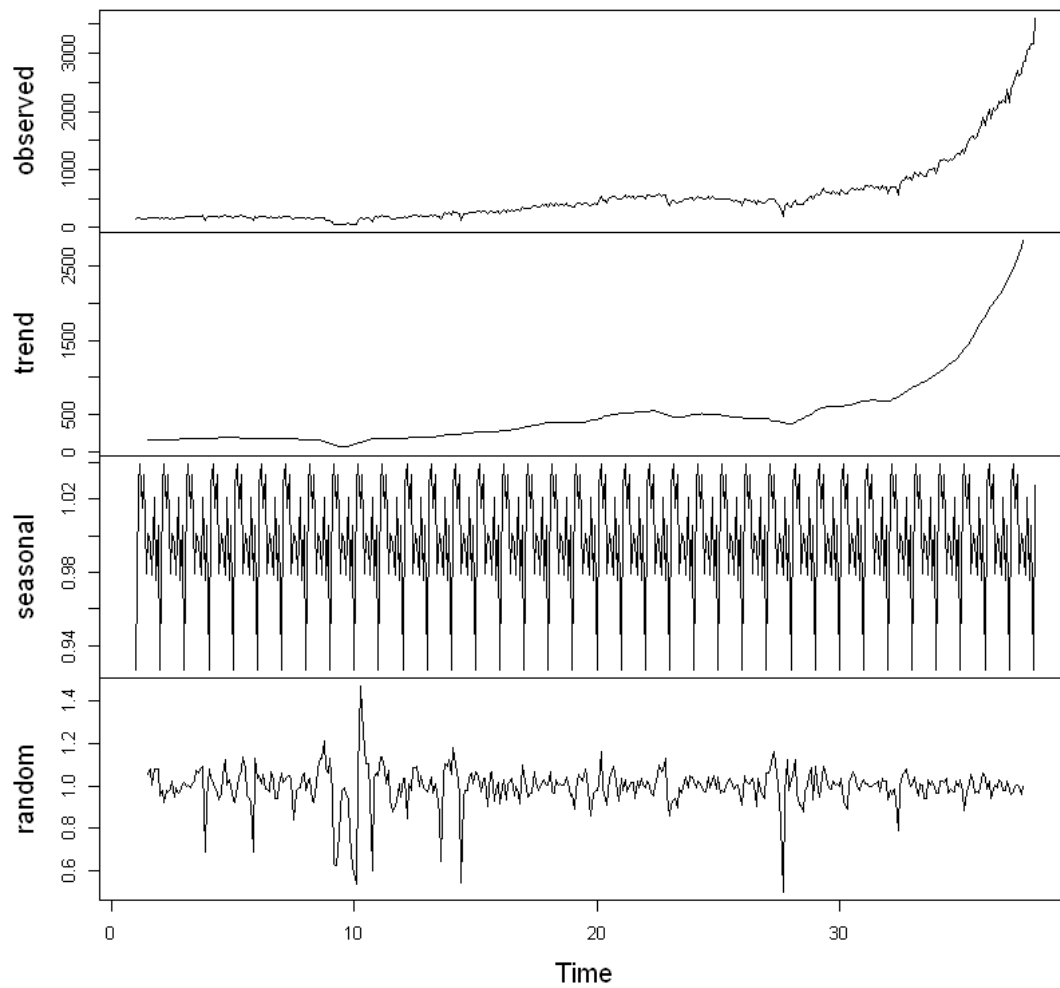
Figure fonksiyonu ile gelen mevsimsellik etki incelendiğinde veri mart ayında en yüksek seviyede, ocak ayında ise en düşük seviyede mevsimsellikten etkilendiğini görebiliyoruz.

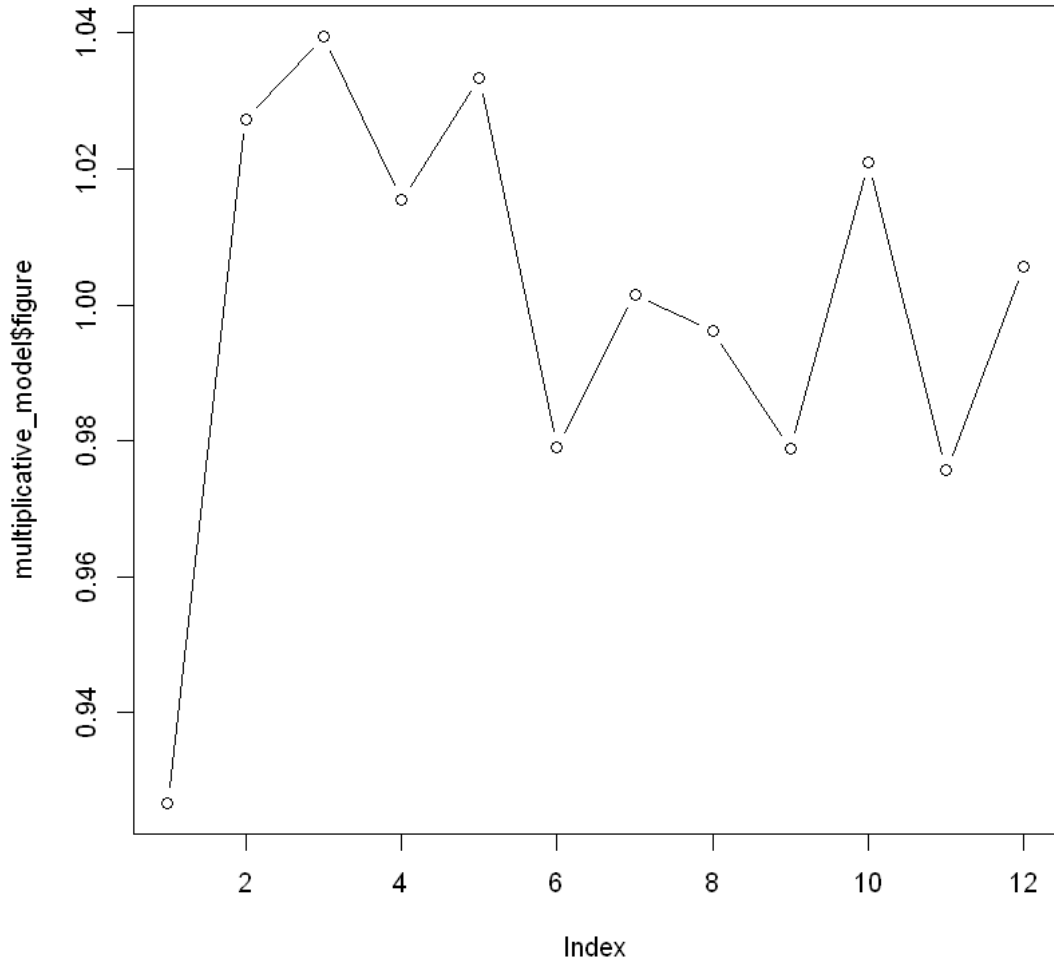
Aşağıda zaman serisi (ts) verimize çarpımsal ayrıştırma tipine decompose fonksiyonu uyguluyoruz. Böylelikle verimizdeki sezonsal ve trendsel etkiden çıkartmış oluyoruz. Plot fonksiyonu ile yaptığımız modeli grafiksel olarak gösteriyoruz. Figure fonksiyonu ile de zaman serimizin 12 aylık sezonsal etkisini gösteriyoruz.

```
In [44]: multiplicative_model<-decompose(ts(data2[,2], freq=12), type= c("multiplicative"))
summary(multiplicative_model)
plot(multiplicative_model)
plot(multiplicative_model$figure, type="b")
```

	Length	Class	Mode
x	446	ts	numeric
seasonal	446	ts	numeric
trend	446	ts	numeric
random	446	ts	numeric
figure	12	-none-	numeric
type	1	-none-	character

Decomposition of multiplicative time series





Çarpımsal ayrışma modelini incelediğimizde trend bileşeninin 1990 yılında en düşük etkide bulunduğunu ancak sonrasında sürekli arttığı gözlenmektedir. Sezonluk etkiyi incelersek veri bütün yıllarda aynı sezonluk değerleri aldığını görüyoruz.

Figure fonksiyonu ile gelen mevsimsellik etki incelendiğinde veri mart ayında en yüksek seviyede, ocak ayında ise en düşük seviyede mevsimsellikten etkilendiğini görebiliyoruz.

Toplamsal ve çarpımsal ayrışma model grafiksel olarak incelendiğinde random kısımlarında büyük farklılıklar gözlenmektedir. Grafiksel olarak karar verilirse additive model daha uygunudur. Ancak doğruluğundan emin olmak için ilerleyen aşamalarda veriye arima modeli uygulayacağız. Summary fonksiyonu ile verinin özetini göstermiş oluruz.

Aşağıda asıl verimize, toplamsal ayrıştırma sezonsal ve trendsel etkiden ayrılmış verimize, çarpımsal ayrıştırma ile sezonsal ve trendsel etkiden ayrılmış verimizin özetlerini görüyoruz.

```
In [45]: summary(data2[,2])
summary(na.omit(additive_model$random))
summary(na.omit(multiplicative_model$random))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
53.0	185.5	390.0	539.2	563.8	3606.0

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-190.9838	-12.0257	0.9057	-1.0926	15.3770	112.8715

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.4991	0.9697	1.0046	0.9970	1.0395	1.4679

11.2 Normallik Testi - Shapiro Wilk Test

Verinin normallliğini kontrol etmek için shapiro.test fonksiyonu kullanılmaktadır. P değeri testen ne kadar emin olduğunu göstermektedir. P değerinde güven aralığını genelde %95 olarak alınır, yani p değeri 0.05'tan küçük ise yaptığımız testen emin olabiliriz. W değeri 1 değerine ne kadar yakınsa veri normallğe o derece yakındır.

Shapiro testini toplamsal ve çarpımsal ayrışma modeline uygulanır.

```
In [46]: shapiro.test(na.omit(additive_model$random))
shapiro.test(na.omit(multiplicative_model$random))
```

Shapiro-Wilk normality test

```
data: na.omit(additive_model$random)
W = 0.90008, p-value = 2.956e-16
```

Shapiro-Wilk normality test

```
data: na.omit(multiplicative_model$random)
W = 0.80709, p-value < 2.2e-16
```

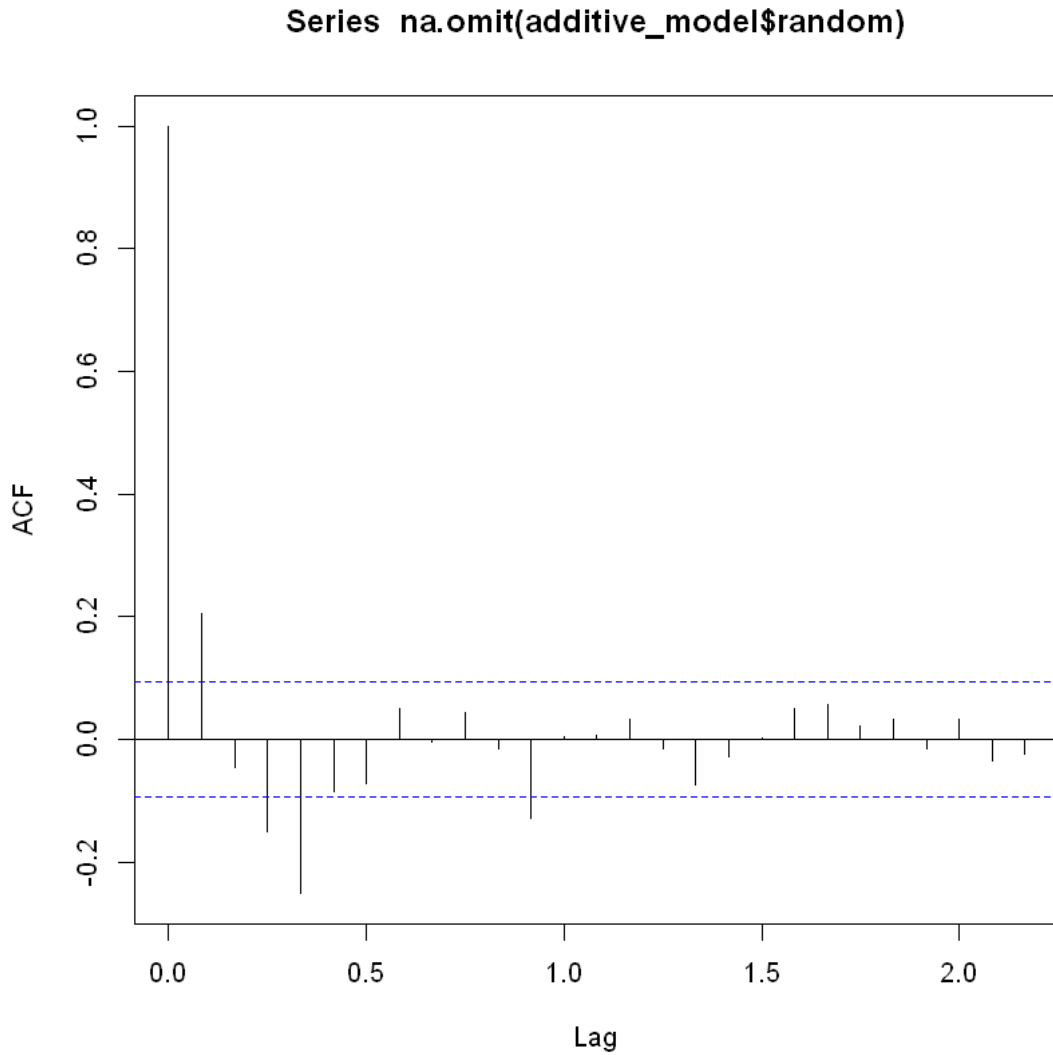
Toplamsal ve çarpımsal ayrışma modeline uygulanan normallik testi incelendiğinde p değerlerinin oldukça küçük, neredeyse 0 yakın olduğunu görüyoruz. Yaptığımız iki testte de oldukça eminiz. W değeri incelendiğinde ise toplamsal modelin 1 daha yakın olduğu görülmektedir. Verimiz normaldir, diyebiliriz.

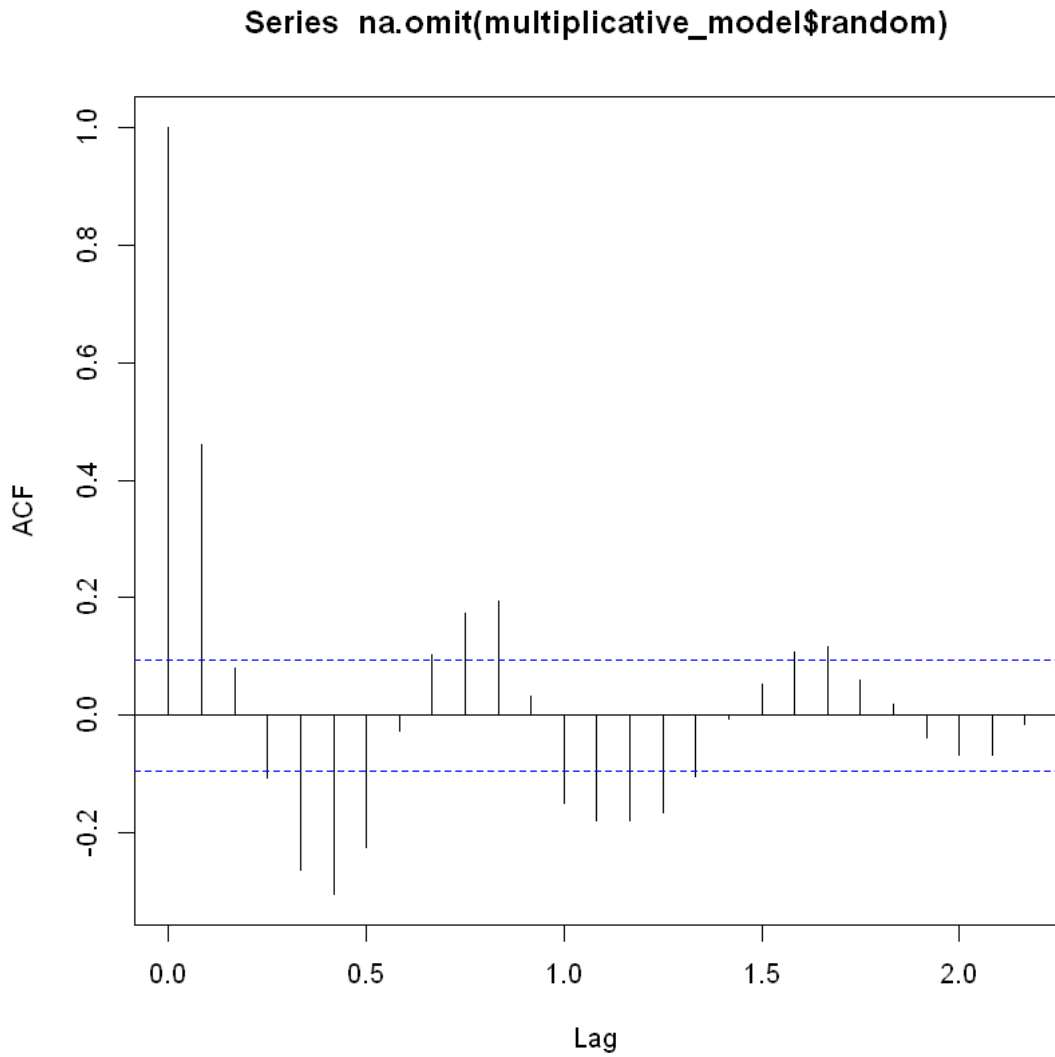
11.3 Otokorelasyon

Zaman serisiyle bu serinin gecikmeli serileri arasındaki ilişkiyi göstermek için R'ın `acf` ve `pacf` fonksiyonunu toplamsal ve çarpımsal ayrışma modeline uygulanır.

ACF eğer çok yüksek bir değerden başlayıp çok yavaş küçülüyorsa, bu serinin durağan olmadığına bir göstergesidir.

```
In [47]: acf(na.omit(additive_model$random))  
         acf(na.omit(multiplicative_model$random))
```

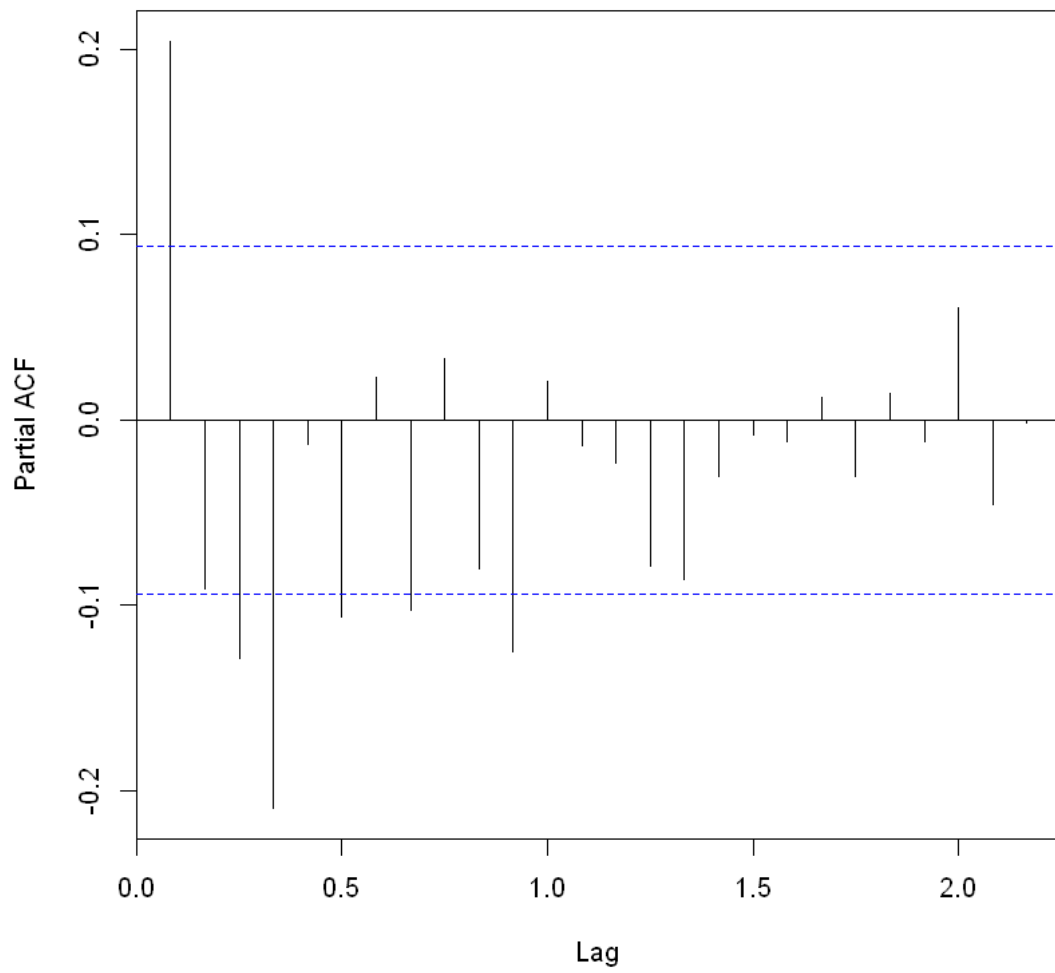




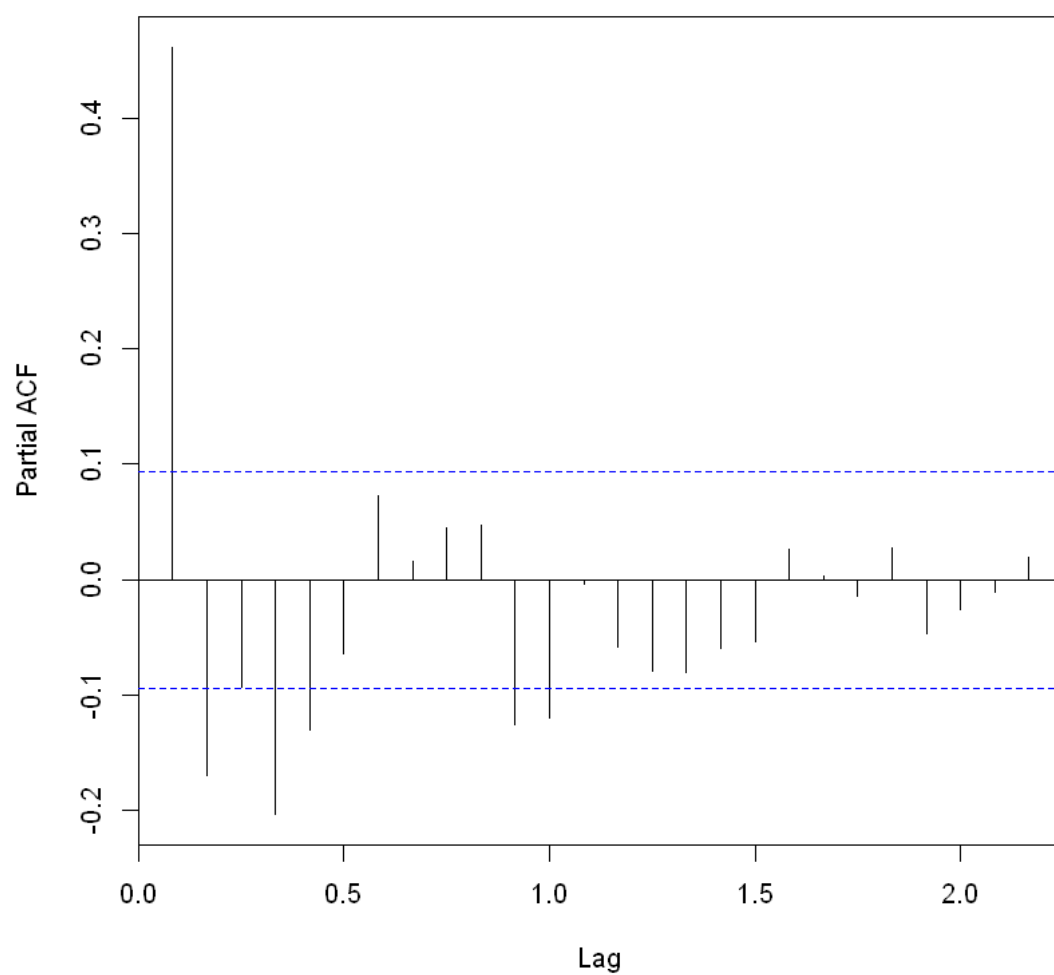
Toplamsal ve çarpımsal ayrıştırma modelleri için acf değerlerini incelediğimizde iki durumda durağan halde olduğunu söyleyebiliriz.

```
In [48]: pacf(na.omit(additive_model$random))  
         pacf(na.omit(multiplicative_model$random))
```

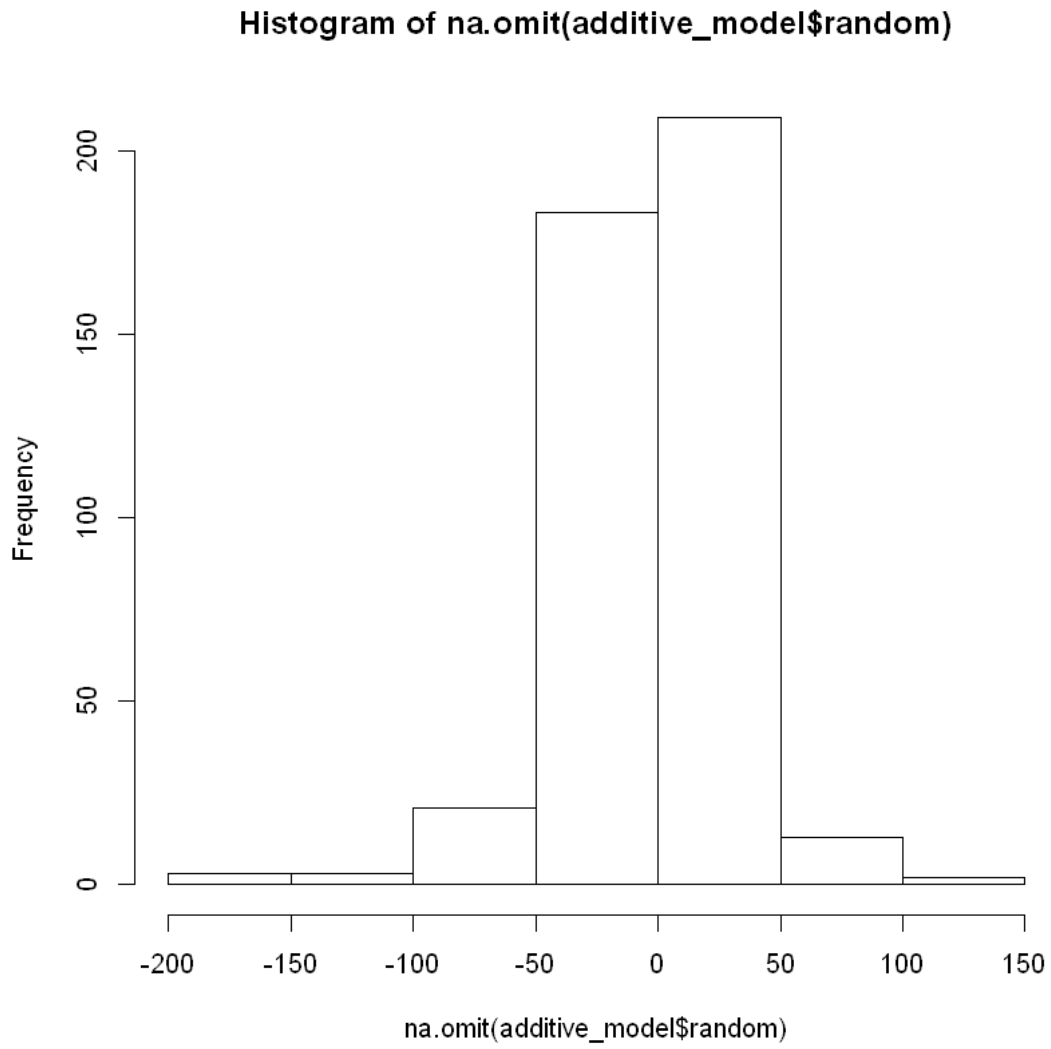
Series na.omit(additive_model\$random)



Series na.omit(multiplicative_model\$random)



```
In [49]: hist(na.omit(additive_model$random))
```



11.4 Durağanlık Analizi

Veride durağanlık durumunu belirlemek için sırasıyla ;Augmented Dickey-Fuller Test-Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test - Phillips-Perron Birim Kök Testi uygulanır.

Augmented Dickey Fuller Test İlk olarak Augmented Dickey Fuller testini verinin bileşenlerden ayrılmamış haline, toplamsal ve çarpımsal ayrışma modellerine sırasıyla uyguluyarak verinin durağanlığını kontrol ediyoruz. ADF testini R içinde bulunan `adf.test` fonksiyonu ile elde edebiliriz.

```
In [50]: adf.test(data2[,2])
          adf.test(na.omit(additive_model$random))
          adf.test(na.omit(multiplicative_model$random))
```

```
Warning message in adf.test(data2[, 2]):
"p-value greater than printed p-value"
```

Augmented Dickey-Fuller Test

```
data: data2[, 2]
Dickey-Fuller = 6.9986, Lag order = 7, p-value = 0.99
alternative hypothesis: stationary
```

```
Warning message in adf.test(na.omit(additive_model$random)):
"p-value smaller than printed p-value"
```

Augmented Dickey-Fuller Test

```
data: na.omit(additive_model$random)
Dickey-Fuller = -9.7154, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary
```

```
Warning message in adf.test(na.omit(multiplicative_model$random)):
"p-value smaller than printed p-value"
```

Augmented Dickey-Fuller Test

```
data: na.omit(multiplicative_model$random)
Dickey-Fuller = -8.2286, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary
```

Yapılan ADF testteki sonuçlar: Additive ve multiplicative modeline göre veriler güven aralığı ($p < 0.05$) içerisinde, yani veri durağandır.

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test İlk olarak Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test testini verinin bileşenlerden ayrılmamış haline, toplamsal ve çarpımsal ayrışma modellerine sırasıyla uyguluyarak verinin durağanlığını kontrol ediyoruz. KPSS testini R içinde bulunan `kpss.test` fonksiyonu ile elde edebiliriz.

```
In [51]: kpss.test(data2[,2], null="Trend")
          kpss.test(na.omit(additive_model$random), null="Trend")
          kpss.test(na.omit(multiplicative_model$random), null="Trend")
```

```
Warning message in kpss.test(data2[, 2], null = "Trend"):
"p-value smaller than printed p-value"
```

KPSS Test for Trend Stationarity

```
data: data2[, 2]
KPSS Trend = 1.0851, Truncation lag parameter = 4, p-value = 0.01
```

```
Warning message in kpss.test(na.omit(additive_model$random), null = "Trend"):
"p-value greater than printed p-value"
```

KPSS Test for Trend Stationarity

```
data: na.omit(additive_model$random)
KPSS Trend = 0.036044, Truncation lag parameter = 4, p-value = 0.1
```

```
Warning message in kpss.test(na.omit(multiplicative_model$random), null = "Trend"):
"p-value greater than printed p-value"
```

KPSS Test for Trend Stationarity

```
data: na.omit(multiplicative_model$random)
KPSS Trend = 0.016544, Truncation lag parameter = 4, p-value = 0.1
```

Yapılan KPSS testteki sonuçlar:

Additive ve multiplicative modeline göre veriler güven aralığı ($p > 0.05$) içerisinde, yani veri durağandır.

Phillips–Perron Birim Kök Testi Phillips–Perron Birim Kök Testi testini verinin bileşenlerden ayrılmamış haline, toplamsal ve çarpımsal ayrışma modellerine sırasıyla uyguluyarak verinin durağanlığını kontrol ediyoruz. Phillips–Perron Birim Kök testini R içinde bulunan `pp.test` fonksiyonu ile elde edebiliriz.

```
In [52]: pp.test(data2[,2])
          pp.test(na.omit(additive_model$random))
          pp.test(na.omit(multiplicative_model$random))
```

```
Warning message in pp.test(data2[, 2]):
"p-value greater than printed p-value"
```

Phillips-Perron Unit Root Test

```
data: data2[, 2]
Dickey-Fuller Z(alpha) = 17.551, Truncation lag parameter = 5, p-value
= 0.99
alternative hypothesis: stationary
```

```
Warning message in pp.test(na.omit(additive_model$random)):
"p-value smaller than printed p-value"
```

Phillips-Perron Unit Root Test

```
data: na.omit(additive_model$random)
Dickey-Fuller Z(alpha) = -280.28, Truncation lag parameter = 5, p-value
= 0.01
alternative hypothesis: stationary
```

```
Warning message in pp.test(na.omit(multiplicative_model$random)):
"p-value smaller than printed p-value"
```

Phillips-Perron Unit Root Test

```
data: na.omit(multiplicative_model$random)
Dickey-Fuller Z(alpha) = -196.72, Truncation lag parameter = 5, p-value
= 0.01
alternative hypothesis: stationary
```

Yapılan Phillips-Person testteki sonuçlar:

Additive ve multiplicative modeline göre veriler güven aralığı ($p < 0.05$) içerisindedir, yani veri durağandır.

11.5 Arima

R'da olan auto.arima fonksiyonu ile verinin kalan sezonsal, trend ve diğer etkenleri kontrol eder ve onlardan arındırır. Veriye auto.arima uygulandığında AIC , BIC ve AICc değerlerinin en düşük olanı analiz için daha kabul edilebilir bir değer almaktadır. ARIMA için p,d,q fonksiyonları ile verinin karmaşıklığı fark edilir ve en sade , en kolay model seçilir.

```
In [53]: auto.arima(ts(data2[,2]))
          auto.arima(ts(na.omit(additive_model$random), start=c(1981, 1), end=c(2018, 1), freq=12))
          auto.arima(ts(na.omit(multiplicative_model$random), start=c(1981, 1), end=c(2018, 1), freq=12))
```

```
Series: ts(data2[, 2])
```

```
ARIMA(0,2,2)
```

```
Coefficients:
```

	ma1	ma2
	-1.4603	0.5198
s.e.	0.0439	0.0435

```
sigma^2 estimated as 2159: log likelihood=-2335.03
```

```
AIC=4676.06 AICc=4676.11 BIC=4688.35
```

```
Series: ts(na.omit(additive_model$random), start = c(1981, 1), end = c(2018, 1), freq = 12)
```

```
ARIMA(2,0,1) with non-zero mean
```

```
Coefficients:
```

	ar1	ar2	ma1	mean
	1.0579	-0.3104	-0.8928	-1.0126
s.e.	0.0519	0.0456	0.0314	0.6058

```
sigma^2 estimated as 885.3: log likelihood=-2139.52
```

```
AIC=4289.03 AICc=4289.17 BIC=4309.52
```

```
Series: ts(na.omit(multiplicative_model$random), start = c(1981, 1), end = c(2018, 1), freq = 12)
```

```
ARIMA(2,0,3)(0,0,1)[12] with non-zero mean
```

```
Coefficients:
```

	ar1	ar2	ma1	ma2	ma3	sma1	mean
	1.5297	-0.8885	-1.0847	0.3008	0.2076	-0.1940	0.9973
s.e.	0.0499	0.0461	0.0736	0.0774	0.0615	0.0509	0.0034

```
sigma^2 estimated as 0.005781: log likelihood=518.12
```

```
AIC=-1020.24 AICc=-1019.91 BIC=-987.46
```

Veriye, son(şubat ayı) verisi eklenmeden önce arima modeli uyguladığımda sonuç aşağıdaki gibi olmuştur.

Verinin hiçbir bileşenden ayrılmadığı durumda auto.arima uygulandığında $p=0$, $d=2$ ve $q=2$ çıkmaktadır. $d=2$ olması verinin durağan olmadığını gösterir.

Verinin toplamsal ayrışma modeline auto.arima uygulandığında $p=2$, $d=0$ ve $q=1$ çıkmaktadır. $d=0$ veri durağandır.

Verilerde çarpımsal modele auto.arima uygulandığında $p=2$, $d=0$ ve $q=3$ çıkmaktadır. $d=0$ veri durağandır.

Hangi ayrışma modelini seçmemiz gerektiğine karar vermek için ilk olarak p, d, q değerlerini karşılaştırmak gerekmektedir. Verinin durağan modeli seçilmesi gerektiğinden hiçbir bileşenden ayrılmadığı model elenir. Toplamsal ve çarpımsal model karşılaştırıldığında toplamsal modelin daha basit olduğu görülür. İkinci olarak (Seçtiğimiz modelin doğruluğundan emin olmak için) auto.arima'nın bize vermiş olduğu AIC, AICc ve BIC değerlerinin en küçük olanı seçilmesi gerekmektedir. Ayrışma modelleri için incelendiğinde toplamsal modelin daha uygun olduğu görülmektedir.

Formülü :

$$\hat{Y}_{d_t} = 1.0579Y_{t-1} - 0.3104Y_{t-2} + E$$

$d=0$ olduğunda orijinal seri zaten durağandır ve farkının alınmasına da gerek yoktur.

Aşağıda her ay güncellenen verimizde şubat ayı eklendiği şekline arima modeli uygulayacağız.

```
In [54]: auto.arima(ts(data2[,2]))
          auto.arima(ts(na.omit(additive_model$random)))
          auto.arima(ts(na.omit(multiplicative_model$random)))
```

```
Series: ts(data2[, 2])
ARIMA(0,2,2)
```

Coefficients:

	ma1	ma2
	-1.4603	0.5198
s.e.	0.0439	0.0435

```
sigma^2 estimated as 2159: log likelihood=-2335.03
AIC=4676.06 AICc=4676.11 BIC=4688.35
```

```
Series: ts(na.omit(additive_model$random))
ARIMA(2,0,1) with non-zero mean
```

Coefficients:

	ar1	ar2	ma1	mean
	1.0680	-0.3148	-0.9066	-0.8773
s.e.	0.0516	0.0466	0.0291	0.5558

```
sigma^2 estimated as 902.4: log likelihood=-2090.78
AIC=4191.55 AICc=4191.69 BIC=4211.92
```

```
Series: ts(na.omit(multiplicative_model$random))
ARIMA(2,0,1) with non-zero mean

Coefficients:
          ar1          ar2          ma1          mean
      1.3402   -0.5220   -0.8800   0.9967
s.e.  0.0704    0.0408    0.0793   0.0025

sigma^2 estimated as 0.006167:  log likelihood=490.07
AIC=-970.14   AICc=-970   BIC=-949.78
```

Ancak her ay güncellenen veride, yani şubat ayı eklenen veriye arima modeli uygulandığında sonuç aşağıdaki gibi olmuştur.

Verinin hiçbir bileşenden ayrılmadığı durumda auto.arima uygulandığında $p=0$, $d=2$ ve $q=2$ çıkmaktadır. $d=2$ olması verinin durağan olmadığını gösterir.

Verinin toplamsal ayrışma modeline auto.arima uygulandığında $p=2$, $d=0$ ve $q=1$ çıkmaktadır. $d=0$ veri durağandır.

Verilerde çarpımsal modele auto.arima uygulandığında $p=2$, $d=0$ ve $q=1$ çıkmaktadır. $d=0$ veri durağandır.

Hangi ayrışma modelini seçmemiz gerektiğine karar vermek için ilk olarak p, d, q değerlerini karşılaştırmak gerekmektedir. Verinin durağan modeli seçilmesi gerektiğinden hiçbir bileşenden ayrılmadığı model elenir. Toplamsal ve çarpımsal model karşılaştırıldığında toplamsal modelin eşit olduğu görülür. İkinci olarak (Seçtiğimiz modelin doğruluğundan emin olmak için) auto.arima'nın bize vermiş olduğu AIC, AICc ve BIC değerlerinin en küçük olanı seçilmesi gerekmektedir.

Geri sapmalardan en düşük olanını yani çarpımsal modeli seçersek:

Formülü :

$$\hat{Y}_{d_t} = 1.3402Y_{t-1} + -0.5220Y_{t-2} + E$$

$d=0$ olduğunda orijinal seri zaten durağandır ve farkının alınmasına da gerek yoktur.

Bu durumda iki arima modelini karşılaştırsak anlayacağımız, veride bir değer (bir aylık verinin) dahi modeli değiştirebildiğini görmüş bulunmaktayız.

Veride sezonsal etkinin olup olmadığını kontrol etmek için veriden hiçbir bileşenin çıkarılmadığı haline auto.arima uygulanır. Bunun için arimanın argümanlarından sseasonal.test'ten yararlanılır.

```
In [55]: auto.arima(ts(data2[,2]))
         auto.arima(ts(data2[,2]), seasonal.test = c("ocsb", "ch"))
```

```
Series: ts(data2[, 2])
ARIMA(0,2,2)
```

```
Coefficients:
          ma1          ma2
      -1.4603   0.5198
s.e.  0.0439   0.0435
```

```
sigma^2 estimated as 2159:  log likelihood=-2335.03
AIC=4676.06   AICc=4676.11   BIC=4688.35
```

```
Series: ts(data2[, 2])
ARIMA(0,2,2)
```

```
Coefficients:
          ma1      ma2
      -1.4603  0.5198
s.e.    0.0439  0.0435
```

```
sigma^2 estimated as 2159:  log likelihood=-2335.03
AIC=4676.06   AICc=4676.11   BIC=4688.35
```

Herhangi bir işlem yapılmayan veriye auto.arima ile sezonsal test uygulandığında ARIMA modelinde; p,d,q değerlerinde ve AIC , AICc ,BIC değerlerinde herhangi bir farklılık görünmemektedir. Bir farklı ifade ile veride sezonsal etki bulunmamaktadır.

Normallik testini seçtiğimiz toplamsal modele tekrar uygulanır.

```
In [56]: white_noise2<-arima(ts(na.omit(additive_model$random)), order=c(2,0,1))
        shapiro.test(white_noise2$residuals)
```

Shapiro-Wilk normality test

```
data: white_noise2$residuals
W = 0.87153, p-value < 2.2e-16
```

Normallik testine göre p değeri oldukça düşük ve w değeri yüksektir. Artık verinin beyaz gürültü olduğunu ve veride herhangi bir bileşen kalmadığını söylenebilir.

11.6 Zaman Serilerinin Mevsimsellikten Arındırılması (de-seasonalize)

Zaman serisinin mevsimsellik deseni hakkında fikir verir ve mevsimsel etkiler olmadan verilerin modellenmesine yardımcı olur. Zaman serisini mevsimsellikten arındırmak için aşağıdaki iki adım kullanılabilir. 1. Adım: stl() fonksiyonu ile zaman serisi ayrılır. 2. Adım: forecast kütüphanesi içerisindeki seasadj() fonksiyonu kullanılır.

```
In [57]: datastl2<- stl(ts(data2[,2], freq=12), s.window="periodic")
          summary(datastl2)
          plot(datastl2)
```

Call:

```
stl(x = ts(data2[, 2], freq = 12), s.window = "periodic")
```

Time.series components:

seasonal	trend	remainder
Min. : -34.17510	Min. : 63.888	Min. : -183.62055
1st Qu.: -9.65014	1st Qu.: 182.538	1st Qu.: -12.84803
Median : -1.33071	Median : 392.483	Median : 0.94450
Mean : -0.03514	Mean : 540.407	Mean : -1.18602
3rd Qu.: 13.54174	3rd Qu.: 545.367	3rd Qu.: 15.28002
Max. : 18.50047	Max. : 3368.702	Max. : 218.79751

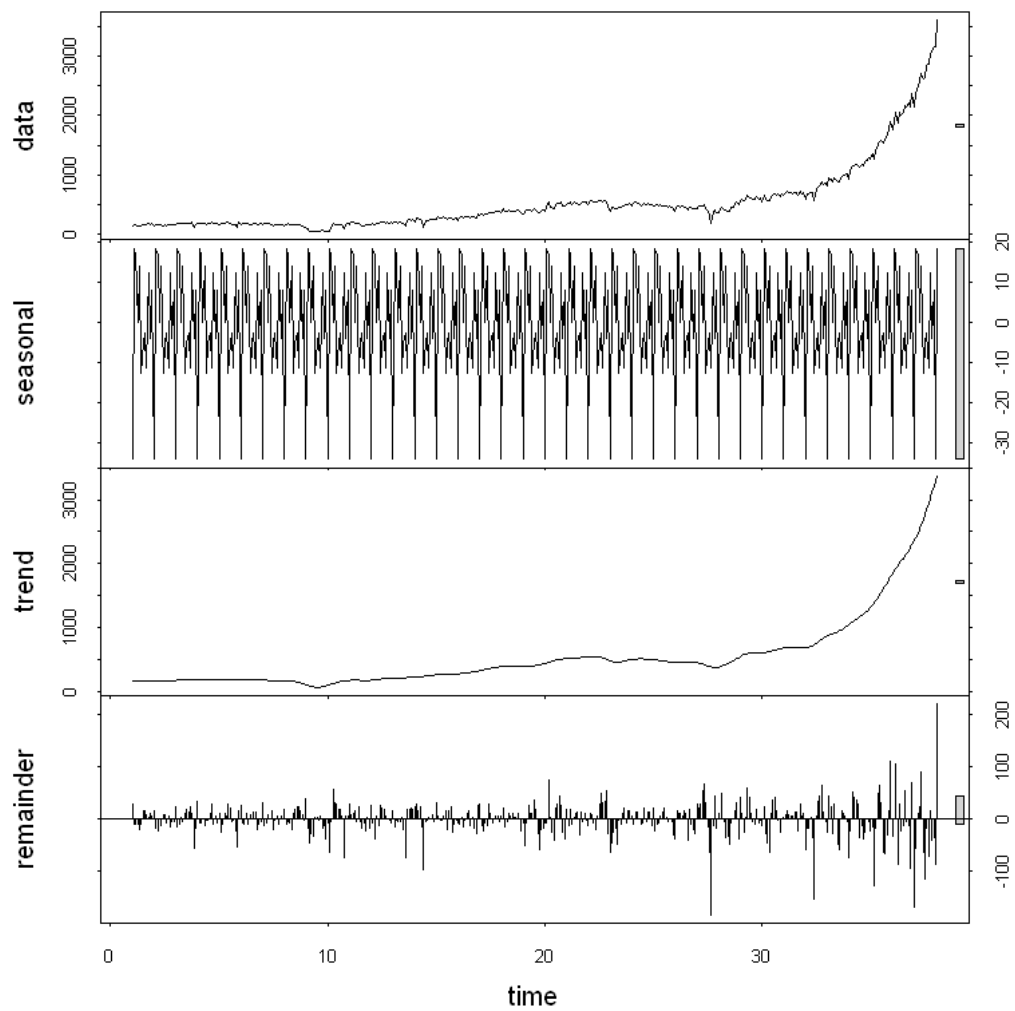
IQR:

STL.seasonal	STL.trend	STL.remainder	data
23.19	362.83	28.13	378.25
% 6.1	95.9	7.4	100.0

Weights: all == 1

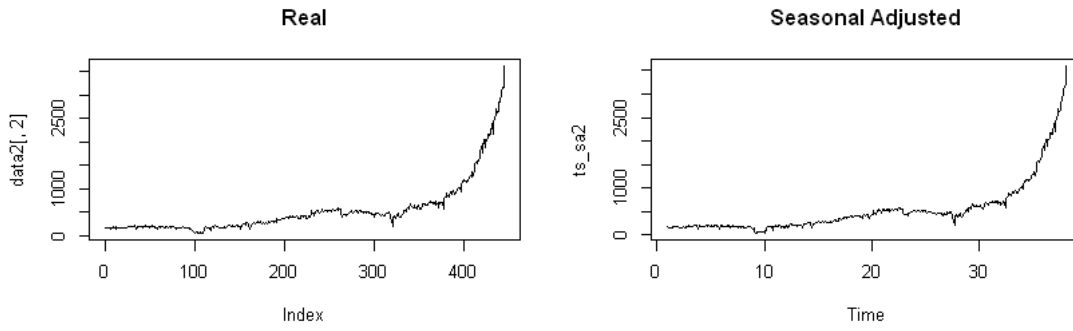
Other components: List of 5

```
$ win : Named num [1:3] 4461 19 13
$ deg : Named int [1:3] 0 1 1
$ jump : Named num [1:3] 447 2 2
$ inner: int 2
$ outer: int 0
```



```
In [58]: ts_sa2 <- seasadj(dataastl2)
```

```
In [59]: par(mfrow=c(3,2))
plot(data2[,2], type="l", main="Real")
plot(ts_sa2, type="l", main="Seasonal Adjusted")
```



Normallik testini ölçmek için sezonsal etkisi çıkarılmış veriye shapiro.test uygulanır.

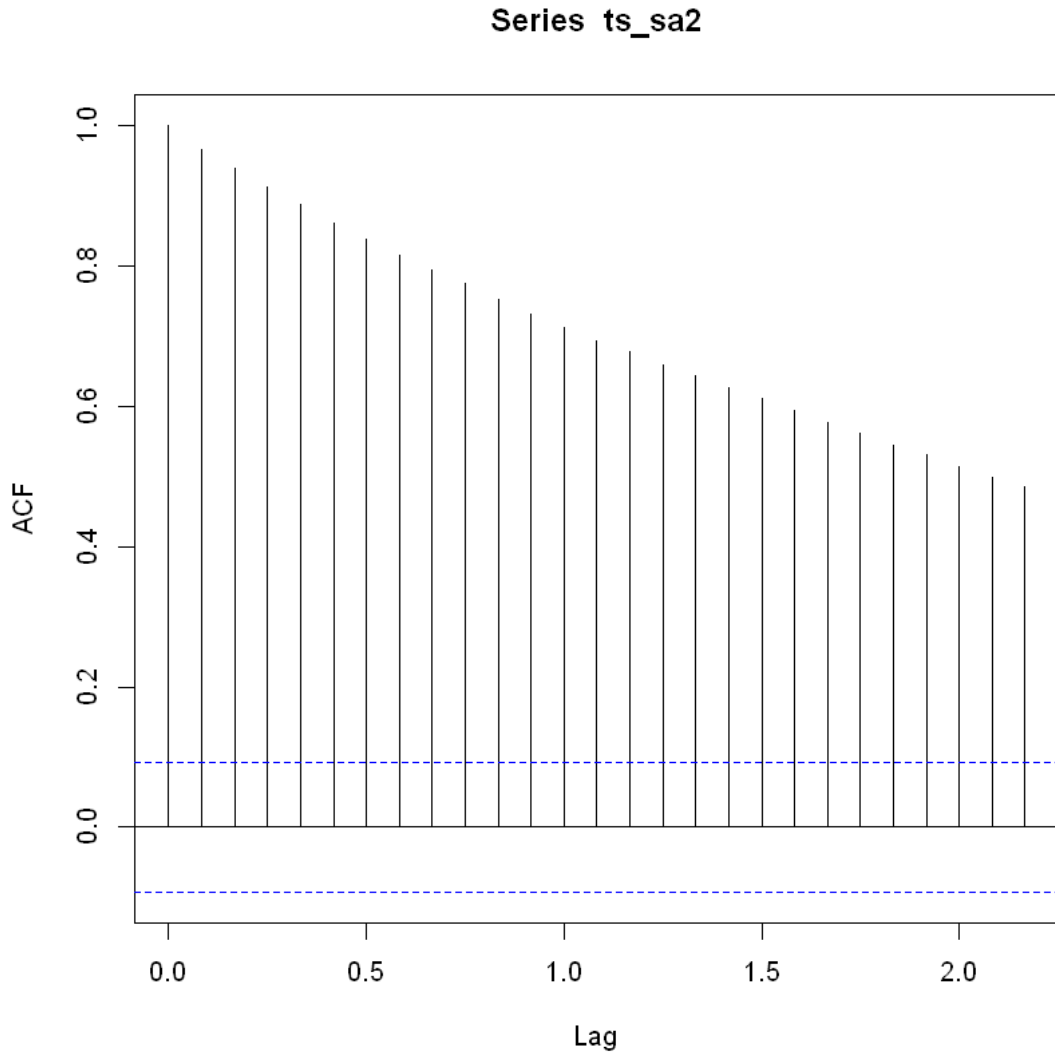
```
In [60]: shapiro.test(ts_sa2)
```

Shapiro-Wilk normality test

```
data: ts_sa2  
W = 0.66646, p-value < 2.2e-16
```

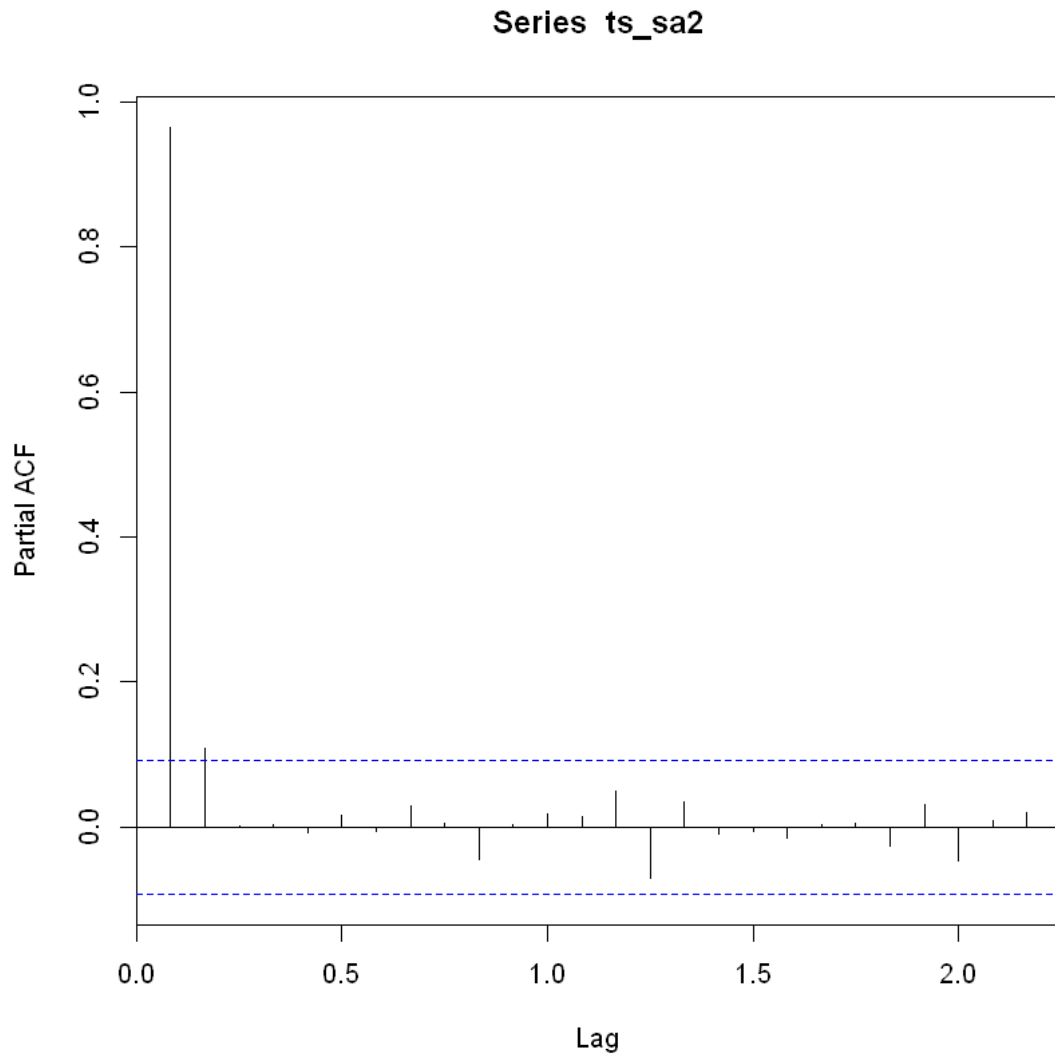
Durağanlık testine göre w değeri 1'den oldukça düşüktür. Bu yüzden ilk yol olarak yapılan çarpımsal modele ilk başta decompose ve sonrasında da auto.arima modelinin uygulandığı analiz yöntemi tercih edilir.


```
In [61]: acf(ts_sa2)
```



Otokorelasyon katsayısı incelendiğinde verinin yavaş yavaş azaldığı gözlenmektedir. Bu durumda ts_sa mevsimsel etkiden arındırılmış zaman serimizin durağan olmadığını göstermektedir.

```
In [62]: pacf(ts_sa2)
```



Sezonsal etkiden arındırılan verinin (ts_sa2) , hala sezonsal etki içerip içermediğini anlamak (mevsimsellik farkını bulmak) için nsdiffs fonksiyonu uygulanır.

```
In [63]: nsdiffs(ts_sa2)
```

0

Mevsimsellik farkı sıfırdır. Mevsimsel etki yoktur.

11.7 Florida Field Production of Crude Oil- Zaman Serisi Analizi Özet

Verinin Adı: Florida Field Production of Crude Oil

Verinin Alındığı Yer: Quandl

Verinin Sıklığı: Aylık

Verinin Zaman Aralığı: 31.01.1981- 28.02.2018

Veride iki sütun bulunmaktadır. Sütunlardan birisi zaman, diğeri ise kullanılan değerdir. Amaç zaman serisindeki gözlenen değerlerinin trend, mevsimsel, çevresel ve rastgele etkenlerden ayırmak ve doğru modeli seçmektir. Bu doğrultuda iki yöntem kullandık.

1. Yöntem : R'da bulunan decompose fonksiyonu ile seriyi mevsimsel ve trendsel etkenlerden ayırmaktır. Decompose fonksiyonunu zaman serilerin hep toplamına hem de çarpımına ayrı ayrı uyguladık. Daha net sonuç veren veri seçilir. Toplamsal ayrışma modelini incelediğimizde trendsel etkinin 1990 yıllarında en düşük etkiye sahip olduğunu ancak sonrasında sürekli arttığını ve mevsimsel etkinin aynı olduğunu görüyoruz. Çarpımsal model de aynı sonuca varıyoruz. İki ayrıştırma modeli arasındaki fark random bileşende ortaya çıkmaktadır. Ancak kullanılan veride toplanan ve çarpılan veride net bir karar veremiyoruz. Verimizin zaman bileşenlerinden ayrıldıktan sonraki normallik durumunu kontrol etmek için veriye normallik testi(shapiro.test) uyguladık, iki modelin de p değerleri güven aralığında ($p < 0.05$) çıktı. W değerleri karşılaştırıldığında ise toplamsal ayrışma modelinin daha iyi sonuç verdiğini görüyoruz. Sonuç olarak iki model içinde verimizin bileşenlerden ayrılmış haline normal diyebiliriz. Zaman bileşenlerinden ayrılmış olan verimizin durağanlığını kontrol etmek için adf, kpss ve pp testlerini uyguladık. Üç testin sonucu da verimizin toplamsal ve çarpımsal model için durağan olduğunu gösterdi. Sonraki aşamalarda decompose fonksiyonu ile veremediğimiz hangi modelin uygun olduğu kararını anlamak için arima testini uyguladık. Veriye bir aylık (şubat ayı) eklenmeden önceki durumda arima vurduğumuzda; Arima ile gelen p,q,d değerlerinden en basit olanını ve AIC, AICc ve BIC değerlerinden en düşük olanını yani toplamsal modele daha uygun olduğuna karar verdik. Ancak şubat ayı da eklenen veride p,q,d değerlerinin toplamsal ve çarpımsal model için aynı olduğunu ,ama AIC , AICc ve BIC değerlerinin çarpımsal modelde daha düşük olduğunu gözlemledik. Burdan verinin bir aylık değerinin arima modellemesinde büyük bir etki sahibi olduğunu fark ettik. Ek olarak herhangi bir bileşenden ayrılmamış veriye arima ile sezonsal etkinin olup olmadığını ölçtük. Sonucunda ise herhangi bir farklılık olmadığını yani verimizde sezonsal bir etkinin olmadığını söyleyebiliriz. Decompose ve sonrasında uygulanan auto.arima ile kalan verimize normallik testini tekrar uygularsak güven aralığını sağladığını ($p = 2.2e - 16 < 0.05$) ve $w = 0.87153$ yani 1 oldukça yakın olduğunu görüyoruz. Bu da verimizin normal olduğunu söyler. Ek olarak artık verimize beyaz gürültü verisi diyebiliriz.
2. Yöntem :Zaman serisini mevsimsel bileşenlerinden ayırmak için R'da bulunan stl fonksiyonunu kullandık. Verilerin mevsimsel etkisinden ayrıldığı durumda normallik değerlerini ölçmek için normallik (shapiro.test) testini uyguladık. Sonucunda $p = 2.2e - 16 < 0.05$ ve $w = 0.66646$ değerlerini elde ettik. Ancak w değerleri 1'den oldukça düşüktür. Veri için normal dağılmamıştır diyebiliriz. Ek olarak mevsimsellik etkisinden arındırılmış veri için acf testi uygulandığında verinin yavaş yavaş azaldığı gözlenmektedir. Bu durum için verinin normal olmadığı söylenmektedir.

Ek olarak uyguladığımız nsdiffs fonksiyonu ile birlikte verinin mevsimsel etki içerip içermediğine, ya da kalıp kalmadığına karar verilmektedir. Fonksiyonu uyguladığımızda görülüyor ki veri mevsimsel bileşen içermemektedir.

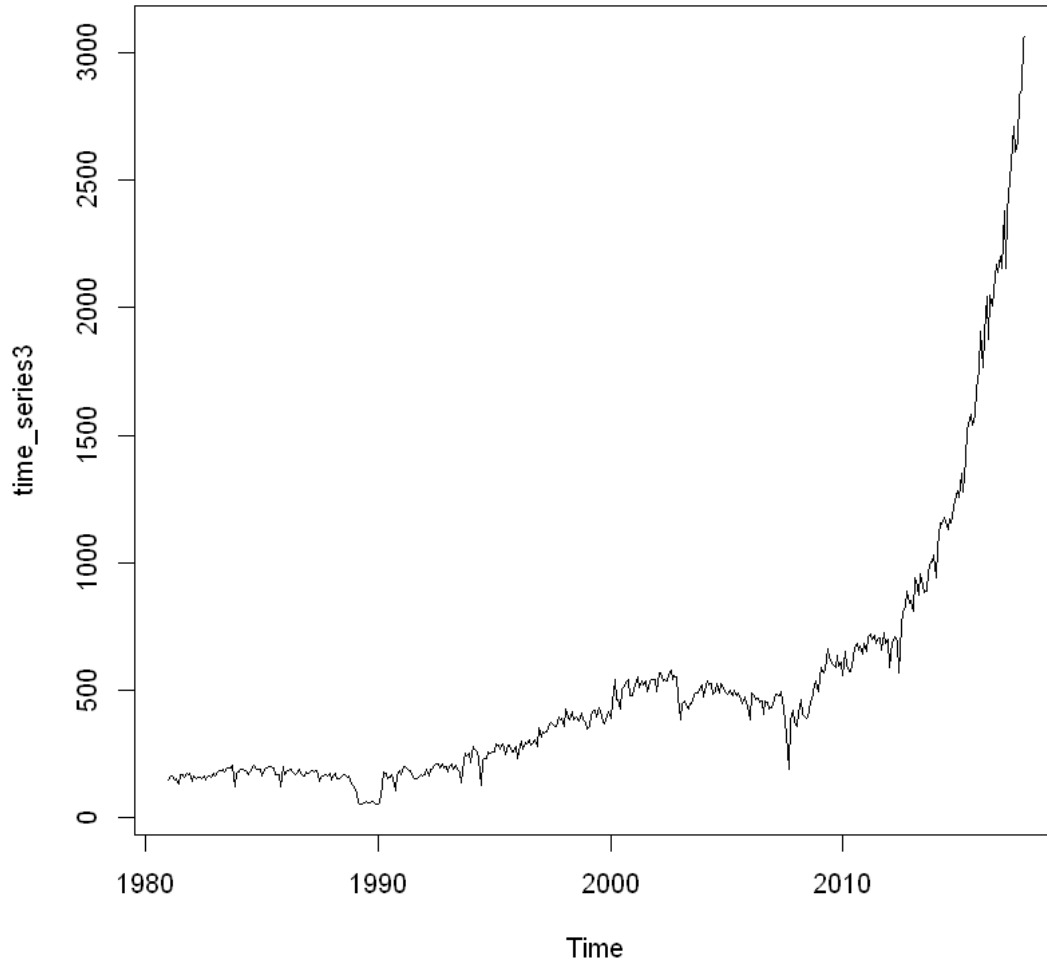
Yukarıdaki üç yöntemi karşılaştırdığımızda (normalliklerini) kullanmamız gereken modelin çarpımsal ve kullanmamız gereken yöntemin sırasıyla decompose ve arima olmasına karar verdik.

11.8 Forecast

```
In [64]: require(forecast)
```

İlk olarak verimizin zaman serisi olduğunu gösteriyoruz. Zaman serisi sıklığını aylık olarak tanımlıyoruz. Zaman aralığımızı 01.1981 - 11.2017 kadar olan süreyi alıyoruz. 12.2017-02.2018 süresini de test verisi olarak alıyoruz.

```
In [65]: time_series3 <-ts(data2[,2], start=c(1981, 1), end=c(2017, 11), freq=12)  
plot(time_series3)
```



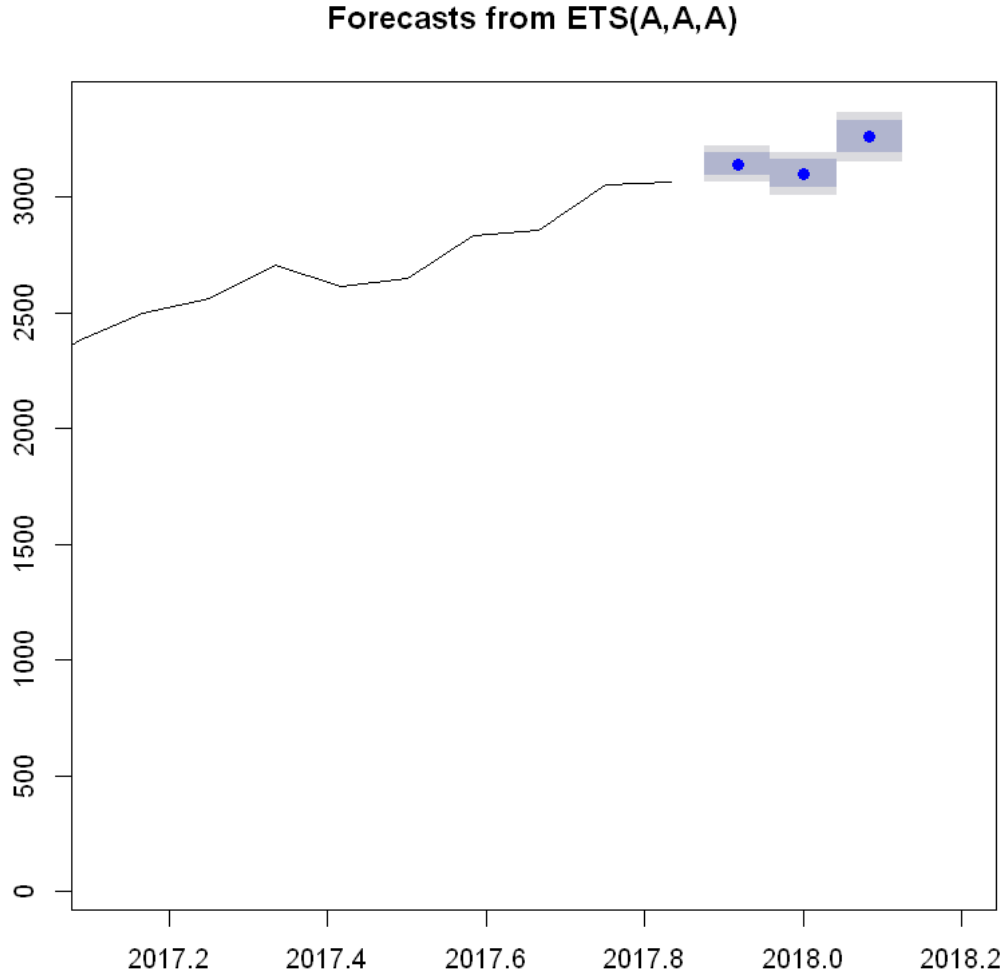
11.8.1 Üstel Düzleştirme Yöntemi- Exponential Smooting

Aşağıda üstel düzleştirme yönteminin ets() fonksiyonu kullanılmış halini uygulayacağız. Öngörü doğruluğunu hesaplamak için R'ın accuracy fonksiyonunu kullanılır. Accuracy fonksiyonu ile gelen verilerde:

Öngörü modellerinden ilk olarak ets modelini deniyoruz.

```
In [66]: model11 <-ets(time_series3)
         data_forecast_ets <- forecast(model11, h=3)
         accuracy(data_forecast_ets)
         plot(data_forecast_ets, ,xlim=c(2017.12,2018.2))
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	2.774238	39.27737	26.06343	-0.3375955	7.387411	0.2734303	0.04872081



Verimize ets() fonksiyonu ile öngörü hesapladığımızda bize ETS(A,A,A) argümanlarını vermiştir. Bu sonuçta öngöründe toplamsal hata, toplamsal trend ve toplamsal sezonsal etkinin uygulandığı yöntemdir denilebilir.

```
In [67]: summary(data_forecast_ets)
```

```
Forecast method: ETS(A,A,A)
```

```
Model Information:
```

```
ETS(A,A,A)
```

```
Call:
```

```
ets(y = time_series3)
```

```
Smoothing parameters:
```

```
alpha = 0.6024
```

```
beta = 0.0461
```

```
gamma = 0.1034
```

```
Initial states:
```

```
l = 166.9415
```

```
b = 2.3989
```

```
s=4.2413 -11.3018 5.7337 -9.4752 -6.5625 -2.0671
```

```
-5.4372 19.2079 3.535 19.4916 9.5222 -26.8879
```

```
sigma: 39.2774
```

```
      AIC      AICc      BIC  
5985.646 5987.086 6055.237
```

```
Error measures:
```

```
              ME      RMSE      MAE      MPE      MAPE      MASE  
Training set 2.774238 39.27737 26.06343 -0.3375955 7.387411 0.2734303
```

```
              ACF1  
Training set 0.04872081
```

```
Forecasts:
```

```
      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95  
Dec 2017      3142.650 3092.314 3192.986 3065.667 3219.632  
Jan 2018      3101.060 3041.067 3161.054 3009.309 3192.812  
Feb 2018      3260.957 3191.518 3330.395 3154.760 3367.153
```

ETS yöntemiyle gelen alpha değeri 0.6024 çıkmıştır. Bu sonuçta değişen bileşenlerden ortalama bir derecede etkinleneceğini göstermektedir.

11.8.2 Basit Üstel Düzleştirme Yöntemi (Durağan Seriler için)

Aşağıda üstel düzleştirme yönteminin HoltWinters() fonksiyonu kullanılmış halini uygulayacağız. Bu model trend, sezonsal bileşenleri de içermemediği durumdur.

```
In [68]: model12 <- HoltWinters(time_series3, beta=FALSE, gamma=FALSE)
          model12
          data_forecast_simple <- forecast(model12, h=3)
          accuracy(data_forecast_simple)
          plot(data_forecast_simple, ,xlim=c(2017.11,2018.2))
```

Holt-Winters exponential smoothing without trend and without seasonal component.

Call:

```
HoltWinters(x = time_series3, beta = FALSE, gamma = FALSE)
```

Smoothing parameters:

alpha: 0.7810215

beta : FALSE

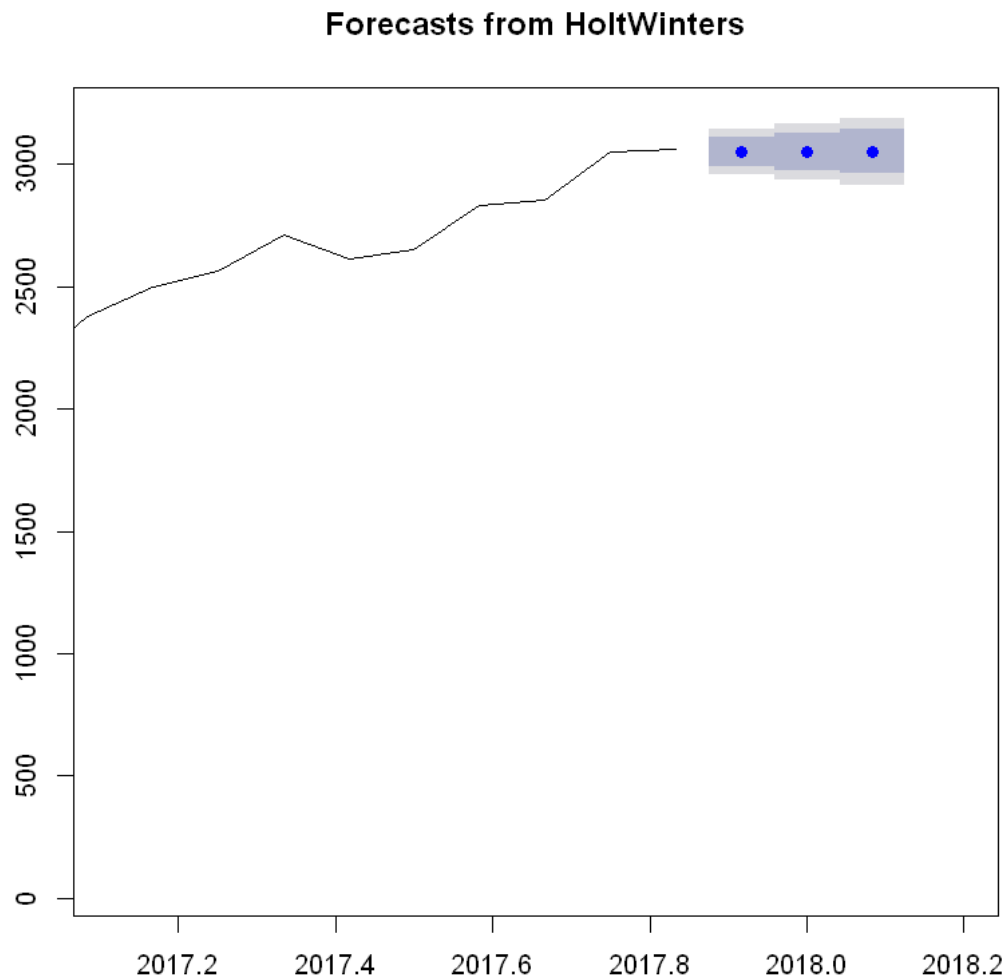
gamma: FALSE

Coefficients:

[,1]

a 3051.299

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	8.410204	47.56481	30.51237	0.03430261	7.450028	0.320104	-0.06453352



```
In [69]: summary(data_forecast_simple)
```

Forecast method: HoltWinters

Model Information:

Holt-Winters exponential smoothing without trend and without seasonal component.

Call:

```
HoltWinters(x = time_series3, beta = FALSE, gamma = FALSE)
```

Smoothing parameters:

alpha: 0.7810215


```

beta : FALSE
gamma: FALSE

Coefficients:
      [,1]
a 3051.299

Error measures:
              ME      RMSE      MAE      MPE      MAPE      MASE
Training set 8.410204 47.56481 30.51237 0.03430261 7.450028 0.320104
              ACF1
Training set -0.06453352

Forecasts:
      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
Dec 2017      3051.299 2991.235 3111.363 2959.439 3143.159
Jan 2018      3051.299 2975.086 3127.512 2934.741 3167.857
Feb 2018      3051.299 2961.805 3140.793 2914.430 3188.168

```

Düzleştirme katsayı değeri (alpha) arttıkça daha büyük tahmin farklılıkları oluşmaktadır. Basit üstel düzleştirme yöntemi uyguladığımızda gelen alpha değeri 0.7810215'dir.

11.8.3 Holt Üstel Düzleştirme Yöntemi D: (Doğrusal Trendi Olan Seriler için)

Trende sahip mevsimsel dalgalanması olmayan serilerin tahmin işleminde Holt üstel düzleştirme yöntemi kullanılmaktadır.

```

In [70]: model13 <- HoltWinters(time_series3, gamma=FALSE)
        model13
        data_forecast_holt <- forecast(model13, h=3)
        accuracy(data_forecast_holt)
        plot(data_forecast_holt, ,xlim=c(2017.11,2018.2))
        data_forecast_holt

```

Holt-Winters exponential smoothing with trend and without seasonal component.

```

Call:
HoltWinters(x = time_series3, gamma = FALSE)

```

```

Smoothing parameters:
alpha: 0.5032057
beta : 0.09194811
gamma: FALSE

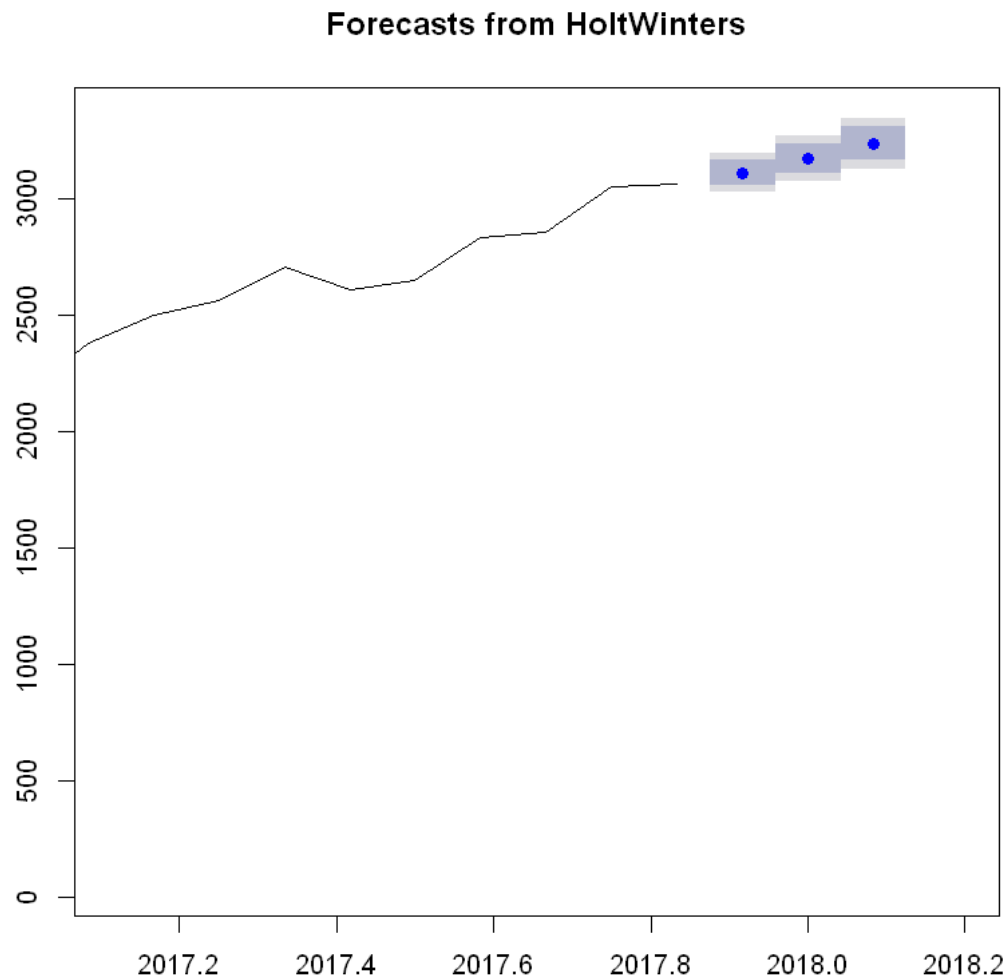
```

```

Coefficients:
      [,1]
a 3050.39138
b   62.40611

```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	2.323311	43.32455	29.40548	-0.857256	7.895946	0.3084917	0.02210322
	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95		
Dec 2017	3112.797	3057.292	3168.303	3027.909	3197.686		
Jan 2018	3175.204	3111.871	3238.537	3078.344	3272.063		
Feb 2018	3237.610	3166.164	3309.056	3128.343	3346.877		



```
In [71]: summary(data_forecast_holt)
```

Forecast method: HoltWinters

Model Information:

Holt-Winters exponential smoothing with trend and without seasonal component.

Call:

```
HoltWinters(x = time_series3, gamma = FALSE)
```

Smoothing parameters:

```
alpha: 0.5032057
beta : 0.09194811
gamma: FALSE
```

Coefficients:

```
      [,1]
a 3050.39138
b   62.40611
```

Error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	2.323311	43.32455	29.40548	-0.857256	7.895946	0.3084917	0.02210322

Forecasts:

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Dec 2017	3112.797	3057.292	3168.303	3027.909	3197.686
Jan 2018	3175.204	3111.871	3238.537	3078.344	3272.063
Feb 2018	3237.610	3166.164	3309.056	3128.343	3346.877

Düzleştirme katsayı değeri arttıkça daha büyük tahmin farklılıkları oluşmaktadır. Basit üstel düzleştirme yöntemi uyguladığımızda gelen alpha değeri 0.5032057'dir. Bu sonuçta değerin düşük olduğunu gösterir. Hata oranı ortalama değerdedir. Diğer modellerde incelenip en uygun öngörü model seçilmelidir.

11.8.4 Holt-Winters Yöntemi: (Doğrusal Trendi ve Mevsimlik Etkileri Olan Seriler için)

Holt-Winters Yöntemi üç temel düzeltme esitliğine sahiptir. Temel, trend ve mevsimsellik için olan bu üç düzeltme sabiti ile yapılan düzeltmeler, Holt'un Doğrusal Yöntemi'ne benzer olarak tek toplam esitlikte toplanarak mevsimsellik esitliği elde etmektedir.

```
In [72]: model14 <-HoltWinters(time_series3)
        model14
        data_forecast_holtWinter <- forecast(model14, h=3)
        accuracy(data_forecast_holtWinter)
        plot(data_forecast_holtWinter, ,xlim=c(2017.1,2018.2))
        data_forecast_holtWinter
```

Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:

```
HoltWinters(x = time_series3)
```

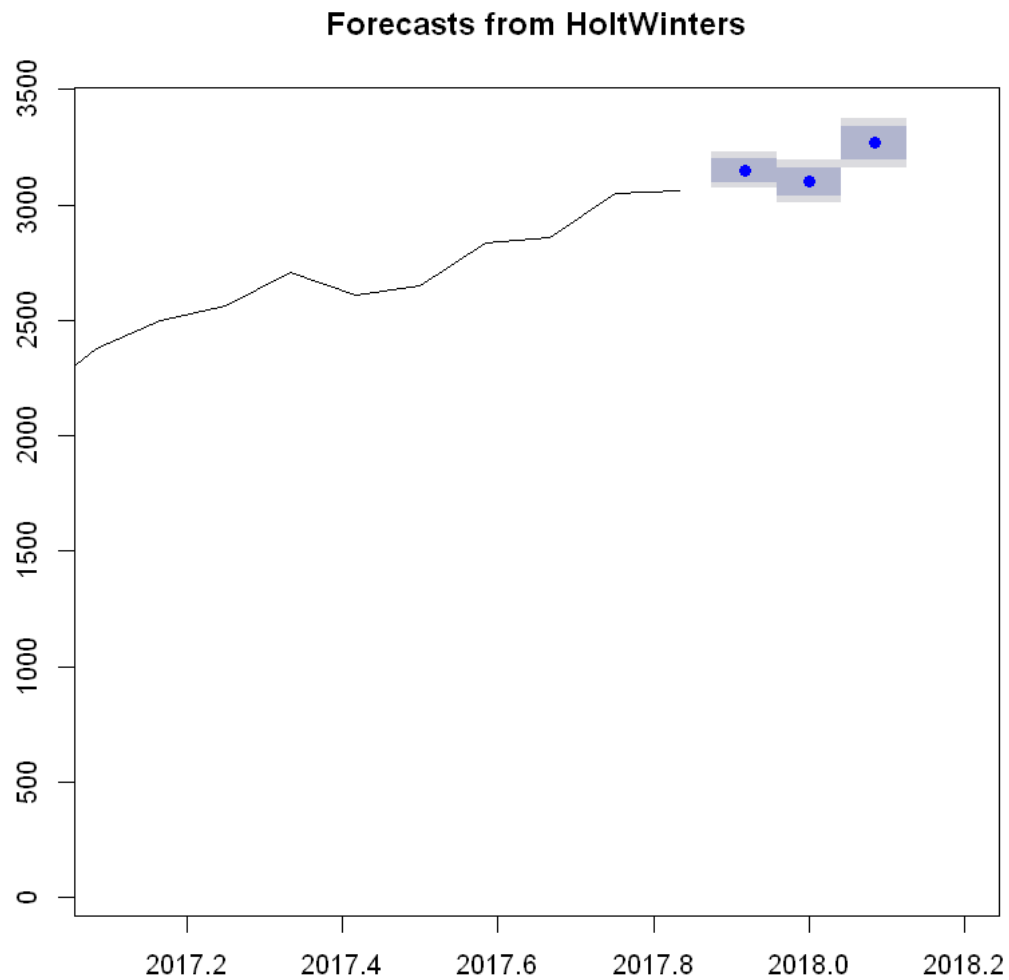
Smoothing parameters:

```
alpha: 0.6076044
beta : 0.07480512
gamma: 0.2946021
```

Coefficients:

```
      [,1]
a 3051.309931
b   58.897088
s1  38.646959
s2 -67.966539
s3  39.639302
s4  49.920330
s5  11.787548
s6  51.898138
s7 -17.438469
s8  -7.923449
s9  15.317955
s10 -10.934216
s11  40.844742
s12   6.342074
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	3.013304	39.89378	26.40596	-0.2875972	7.228475	0.2770238	0.05248844
	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95		
Dec 2017	3148.854	3097.815	3199.893	3070.796	3226.912		
Jan 2018	3101.138	3040.179	3162.096	3007.909	3194.366		
Feb 2018	3267.640	3197.022	3338.259	3159.639	3375.642		



```
In [73]: summary(data_forecast_holtWinter)
```

Forecast method: HoltWinters

Model Information:

Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:

```
HoltWinters(x = time_series3)
```

Smoothing parameters:

alpha: 0.6076044

beta : 0.07480512
gamma: 0.2946021

Coefficients:

[,1]
a 3051.309931
b 58.897088
s1 38.646959
s2 -67.966539
s3 39.639302
s4 49.920330
s5 11.787548
s6 51.898138
s7 -17.438469
s8 -7.923449
s9 15.317955
s10 -10.934216
s11 40.844742
s12 6.342074

Error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	3.013304	39.89378	26.40596	-0.2875972	7.228475	0.2770238

ACF1

Training set 0.05248844

Forecasts:

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Dec 2017	3148.854	3097.815	3199.893	3070.796	3226.912
Jan 2018	3101.138	3040.179	3162.096	3007.909	3194.366
Feb 2018	3267.640	3197.022	3338.259	3159.639	3375.642

Düzleştirme katsayı değeri arttıkça daha büyük tahmin farklılıkları oluşmaktadır. Basit üstel düzleştirme yöntemi uyguladığımızda gelen alpha değeri 0.6076044'dir. Bu sonuçta değerin büyük olduğunu gösterir. Hata oranı fazladır. O yüzden öngörü yaparken bu model seçilmemelidir.

Aşağıda genel olarak bütün doğrulukları göstereceğiz.Hatırlamak için

ME:Ortalama Hata

MAE : Ortalama Mutlak Hata

MSE: Ortalama Hata Kare

MPE: Ortalama Yüzde Hata

MAPE: Ortalama Mutlak Yüzde Hata

```
In [74]: accuracy(data_forecast_ets)
         accuracy(data_forecast_simple)
         accuracy(data_forecast_holt)
         accuracy(data_forecast_holtWinter)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	2.774238	39.27737	26.06343	-0.3375955	7.387411	0.2734303	0.04872081
	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	8.410204	47.56481	30.51237	0.03430261	7.450028	0.320104	-0.06453352
	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	2.323311	43.32455	29.40548	-0.857256	7.895946	0.3084917	0.02210322
	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	3.013304	39.89378	26.40596	-0.2875972	7.228475	0.2770238	0.05248844

Yapılan 5 farklı öngörü modelinin hata değerlerini karşılaştırmak için literatürde en çok kullanılan ve tercih edilen Ortalama Mutlak Hata (MAE) değerini kullanılırız. Bu karşılaştırmaya göre ets yöntemi en uygun yöntemdir.

11.8.5 Florida Field Production of Crude Oil - Zaman Serilerinde Öngörü Özet

Verimizde öngörü yapabilmek için tahminleme verisi ve test verisi olmak üzere iki veri seti oluşturduk.Tahminleme verisini Ocak 1981 ile Kasım 2017 arasını , test verisi için son üç ayı yani Aralık 2017 ile Şubat 2018 arasında aldık. Sırasıyla ets,holtwinters modellerini uyguladık. Hata terimleri incelendiğinde arima modelinin öngörü için daha uygun olduğuna karar verdik.

Petrol verileri ile ilgili analiz ve öngörü hesaplamalarını yapmış olduk.

12 Kaynakça

- Brockwell, Richard A. Davis (2002). Introduction to Time Series and Forecasting
- Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (1994). Time Series Analysis, Forecasting and Control, 3rd ed. Prentice Hall, Englewood Cliffs, NJ.
- Cryer, J. D. and Chan, K. S. (2008). Time series analysis: with applications in R. Springer, 106-107.
- Dickey, D. A. and W. A. Fuller (1981) Likelihood ratio statistics for autoregressive time series with a unit root, *Econometrica*, 49, 1057–1071.
- Frances PH (1991). Seasonality, non-stationarity, and the forecasting of monthly time series. *International Journal of Forecasting*, 7: 199–208.
- Introduction to Forecasting with ARIMA in R, <https://www.datascience.com/blog/introduction-to-forecasting-with-arima-in-r-learn-data-science-tutorials>
- Kayım, H., 1985. İstatistiksel ön tahmin yöntemleri: Hacettepe Üniversitesi İktisadi ve İdari Bilimler Fakültesi.
- Montgomery, D. C., Johnson, L. A. and Gardiner, J. S., 1990. Forecasting and time series analysis: McGraw-Hill Companies.
- The R Project for Statistical Computing , <https://www.r-project.org/>