**YILDIZ TEKNİK ÜNİVERSİTESİ**

**IST4122 ZAMAN SERİSİ DERSİ PROJE ÖDEVİ**

Asst. Prof. Elif TUNA



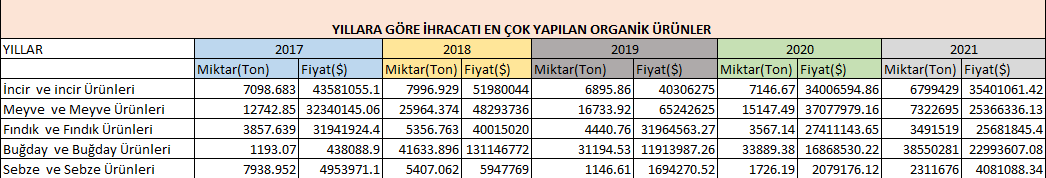
Leyla Tunç 20023036

Aylin Öztürk 18023049

Fatma Erdem 19023017

**BİRİNCİ VERİ: İHRACAT VERİ SETİ**

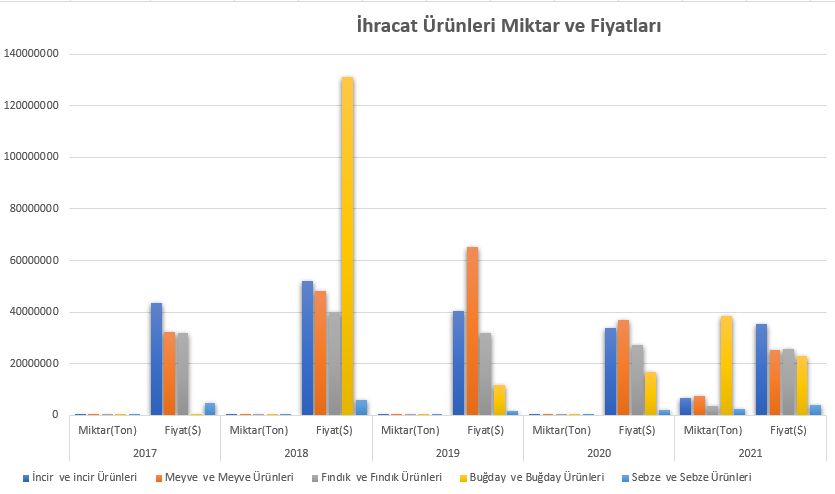
Verimiz; 2017-2021 yılları arasında ihracatı yapılan incir ve incir ürünleri, meyve ve meyve ürünleri, fındık ve fındık ürünleri, buğday ve buğday ürünleri ve sebze sebze ürünleri olmak üzere miktar ve fiyat bakımından görmekteyiz.



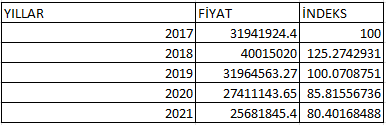
**İNDEKS HESABI**

**1) Basit İndeks**

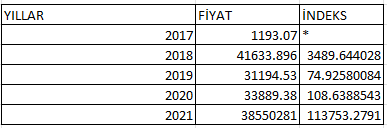
Tabloyu baz alarak yıllara göre ihracatı en çok yapılan organik ürünler verilmiştir. 2017 yılını temel alarak fındığın satış fiyatındaki değişmeyi bulalım.



Görmekte olduğumuz veri tablosundan hareketle görselleştiren bu grafikte ihracatı yapılan ürünlerin yıllara göre miktar ve fiyatları mevcuttur.Göze çarpan 2018 yılında buğday fiyatının pik yaptığıdır. Ayrıca 2021 yılında hasadı yapılan buğday miktarı fiyatından daha fazla olduğu görülmektedir.2019 yılında da meyve ve meyve ürünlerinin fiyatları artmıştır.

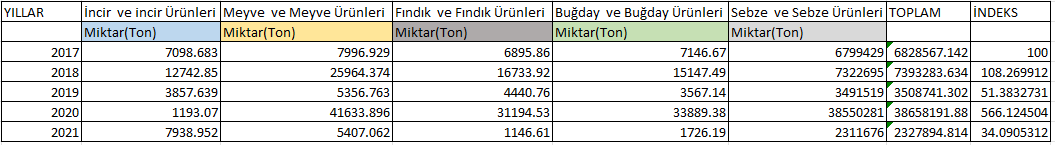


Yıllara göre ihracatı en çok yapılan organik ürünler yukarıdaki tabloda verilmiştir.Yıllara göre buğdayın satış miktarları arasındaki değişmeyi bulalım.

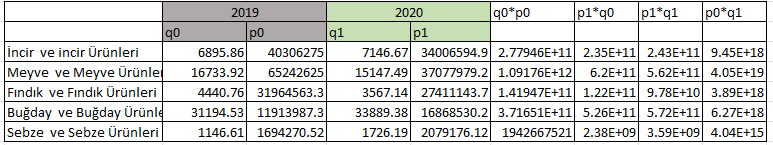


**2) Bileşik İndeks**

Yıllara göre ihracat yapılan ürünler yukarıdaki tabloda verilmiştir.Bu verileri kullanarak 2017 yılını temel alan toplam miktar indeksini hesaplayalım.

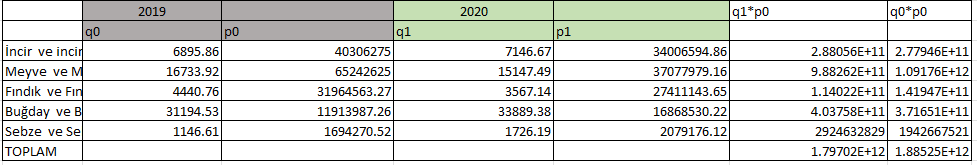


Ürünlerin 2021 yılındaki miktarları 2017 yılı miktarına göre %34.09 artmıştır. Aşağıdaki tabloda tüm p0, p1, q0 ve q1 değerlerinin hesapları verilmiştir.



i. Lespayres Miktar İndeksi

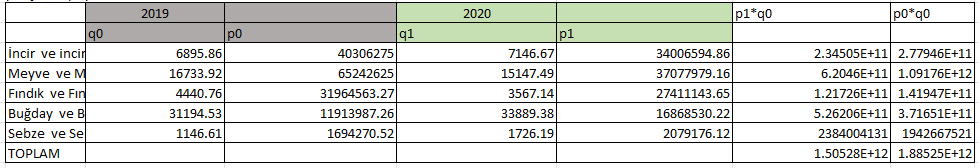
2019 yılını temel alarak 2020 yılı için lespeyres miktar indeksi hesabını bulalım.



Buradan Lespayres indeksi = I :95.31998154 olarak 5 temel gıda maddesi miktarı 2019 yılına göre 4.69 azalmıştır diyebiliriz.

ii. Lespayres Fiyat İndeksi

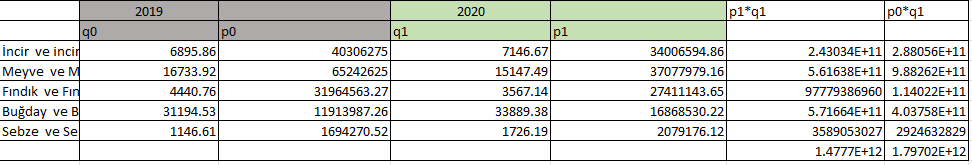
2019 yılını temel alarak 2020 yılı için lespeyres fiyat indeksini hesaplayalım.



Buradan Lespayres fiyat indeksi= 79.84506686 olarak buluruz.Söz konusu 5 gıda maddesi 2020 yılında 2019 yılına göre 20.16 oranında azalmıştır.

iii. Paasche İndeksi

2019 Yılını temel alarak 2020 yılı için paasche fiyat endeksi hesaplayalım.



Buradan paasche indeksi = 79.84506686 . Söz konusu 5 gıda maddesi 2020 yılında 2019 yılına göre 17.77 oranında azalmıştır.

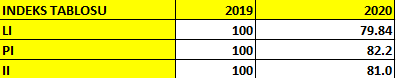
iv. Fisher İndeksi

2019 Yılını temel alarak 2020 yılı için fisher fiyat endeksi hesaplayalım.

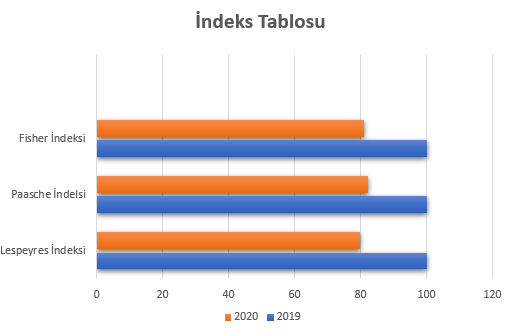
Fisher İndeksi = √(Laspeyres İndeksi \* Paasche İndeksi) olduğunu biliyoruz. O zaman; 81.02912973 olarak buluruz.

Söz konusu 5 gıda maddesi 2020 yılında 2019 yılına göre 18.971 oranında azalmıştır.

Özet olarak şöyle bir tablo oluşur:



**3) Enflasyon Oranlarının Karşılaştırılması**

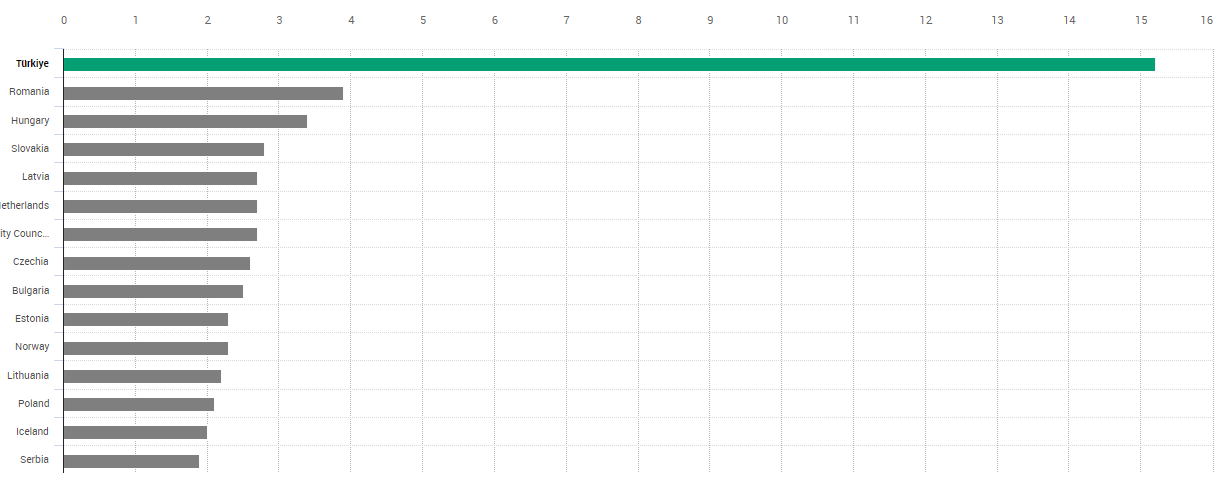


Enflasyon Oranlarının Karşılaştırılması

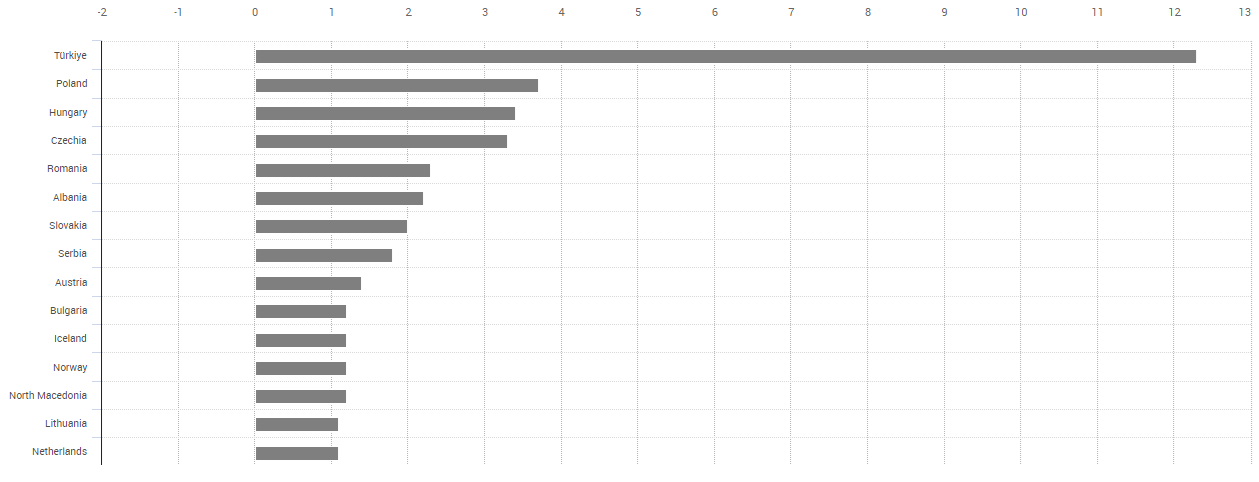
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2017 | 2018 | 2019 | 2020 | 2021 |
| 11.1 | 16.3 | 15.2 | 12.3 | 19.6 |

Yıllara Göre Türkiye Enflasyon Oranları

Türkiye’nin 2017 ile 2021 yılları arasında enflasyon oranları verilmiştir.En yüksek oranı 2021 yılında 19.6 ile karşılaşmaktayız. Lespeyres, Paasche ve Fisher indeksleri ile karşılaştırırsak 2019 yılında aynı ilerlerken,2020 yılında ise Paasche indeksi diğer indekslere kıyasla daha fazladır.



2019 yılı avrupa ülkeleri ile Türkiye enflasyon oranları karşılaştırılması



2020 yılı avrupa ülkeleri ile Türkiye enflasyon oranları karşılaştırılması:

**İKİNCİ VERİ: ROBBERIES VERİ SETİ**

Veri seti, Ocak 1966'dan Ekim 1975'e kadar Boston'daki aylık silahlı soygunların sayısını veren bir veri setidir. Değerler birer sayıdır ve 118 gözlem vardır.

|  |
| --- |
| Months: Aylar |
| Robberies: Soygunlar |

**1) GEREKLİ ZAMAN SERİLERİ**

Gerekli Kütüphaneler:

library(tidyverse)

library(tidyr)

library(tidyquant)

library(fpp3)

library(fpp2)

library(seasonal)

library("TTR")

library(ggplot2)

library(ggseas)

library(seasonal)

library(stats)

Veriyi Yükleme:

rob <- read.csv("Robberies.csv",sep=";")

rob

Burada robberies veri setini rob adlı değişkene atadık.

rob\_ts <- ts(rob$Robberies,

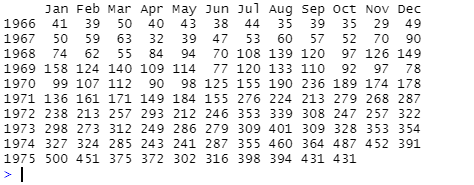
start = c(1966, 01),

end = c(1975, 10),

frequency = 12)

rob\_ts

“rob” adlı veri setini zaman serisine dönüştürdük.Verinin başlangıç tarihi 1966 Ocak ayı olduğundan, bunu “c(1966, 01)” olarak verdik. Bitiş tarihi de aynı şekilde 1975 yılının Ekim ayı. Son argüman ise frekans, yani verinin hangi sıklıkla gözlemlendiği.Bizim veri setimiz de aylık gözlemlenen bir veri seti olduğundan, bir yıl içerisinde 12 gözlem barındırıyor. Yani frekans değeri 12.Ve aşağıdaki çıktıyı elde edebiliriz.

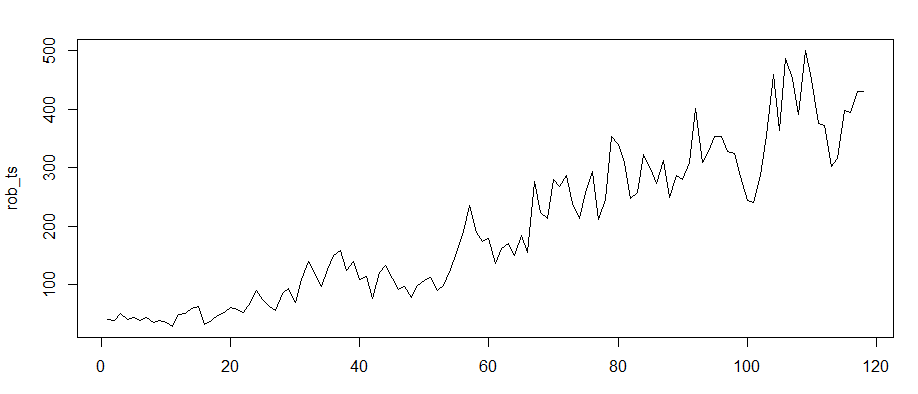


Zaman Serisi Grafiği:

Robberies veri setini ts() fonksiyonu ile zaman serisi olarak aldık ve plot.ts() ile zaman serisi grafiği çizdirelim.

rob\_ts <- ts(rob$Robberies)

plot.ts(rob\_ts)



Grafiğe baktığımızda verimizin trendini çizgi grafiği ile görebiliyoruz. X ekseninde bulunan veriler bize 1den 118’e kadar aylarımızı göstermektedir.

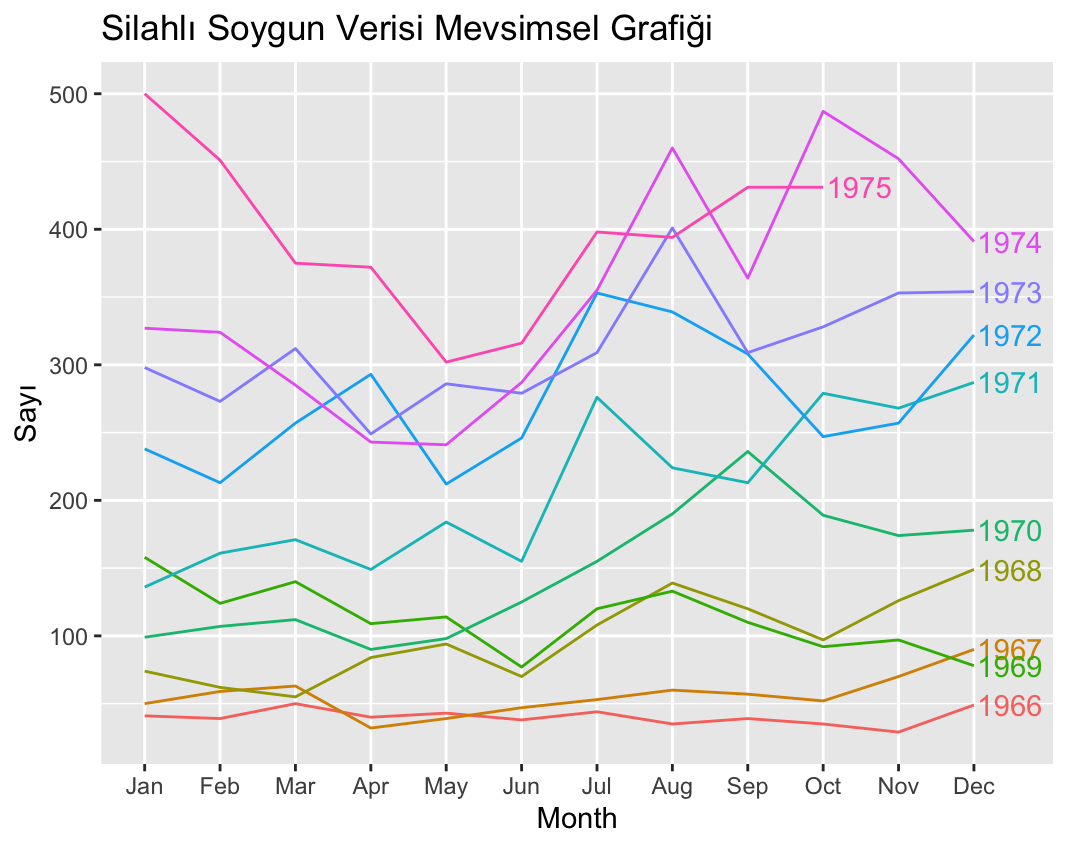
Bu veride mevsimselliği net bir şekilde göremiyoruz. Serideki mevsimselliği görmenin en iyi yollarından biri, mevsimsellik grafiğini çizdirmektir. R daki kodu şöyledir:

ggseasonplot(x = rob\_ts,

year.labels = TRUE) +

ylab("Sayı") +

ggtitle("Silahlı Soygun Verisi Mevsimsel Grafiği")



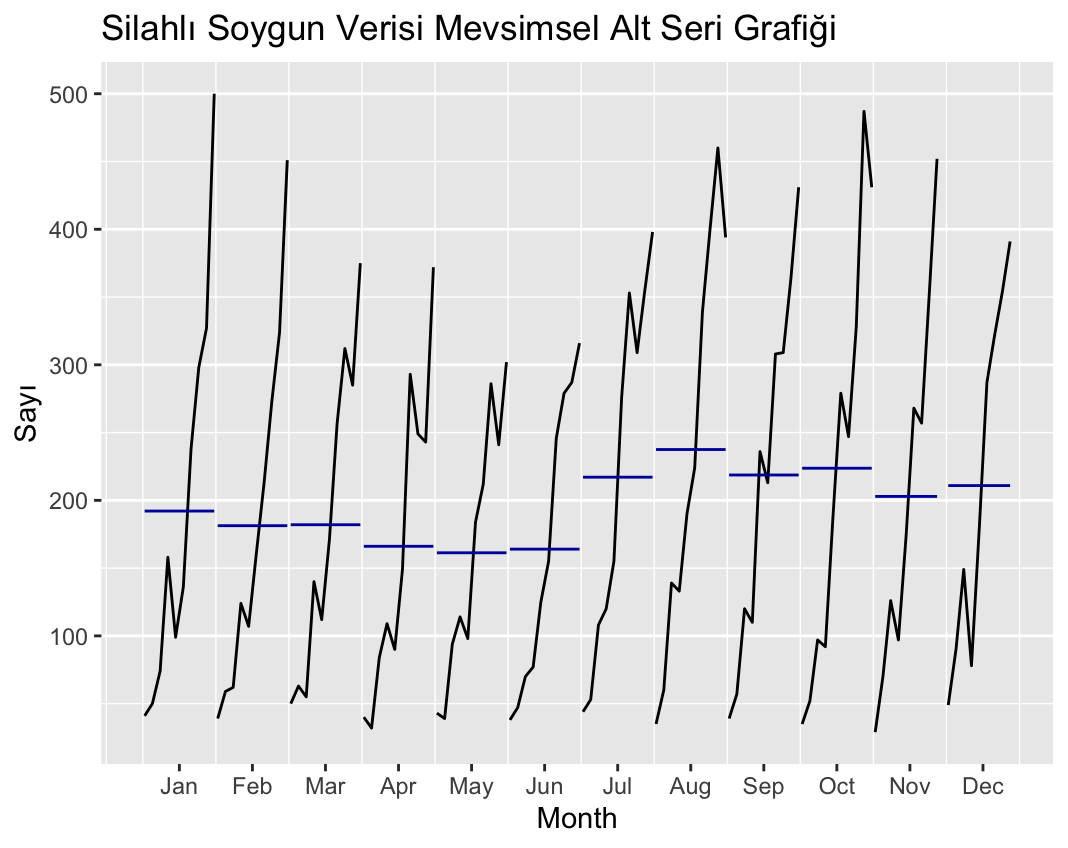
Örneğin yukarıda yera alan mevsimselliği gösteren grafiğimizde her Mart ayında suç oranlarında bir düşüş olduğunu görüyoruz. Ancak verimiz yıllara göre oldukça dalgalı olduğu için grafiğe bakarak net kararlar vermek doğru sonucu bize vermeyecektir.

Başka bir mevsimsellik grafiği kullanarak da mevsimselliğe göz atabiliriz. Bu da bize mevsimsel alt serileri gösterecek. Aşağıdaki kod bloğu ile görelim

ggsubseriesplot(x = rob\_ts) +

ylab("Sayı") +

ggtitle("Silahlı Soygun Verisi Mevsimsel Alt Seri Grafiği")



Bu grafik, mevsimselliği daha iyi anlamamıza yardımcı oluyor. Mavi çizgiler ortalama değerleri gösteriyor.

**2) AYRIŞTIRMA YÖNTEMLERİ**

i. Toplamsal Ayrıştırma:

Trend, konjonktür ve mevsim etkileri ile hata bileşeninin ortaya çıkarılması amacı ile kullanılan yöntem ise ayrıştırma yöntemi olarak adlandırılmaktadır.

Bu aşamada gözlenen zaman serisi ile ilgili iki varsayım yapılır.

𝑌\_𝑡=𝑇\_𝑡+𝑀\_𝑡+𝐷\_𝑡

Çarpımsal modele göre

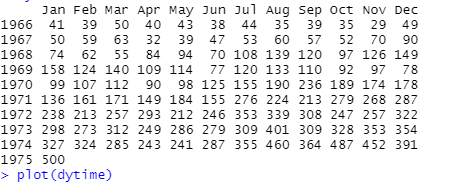
𝑌\_𝑡=𝑇\_𝑡∗𝑀\_𝑡∗𝐷\_𝑡

Toplamsal Yöntemle bileşenleri bulursak;

dytime <- ts(rob$Robberies, frequency=12, start=c(1966),end=c(1975))

dytime

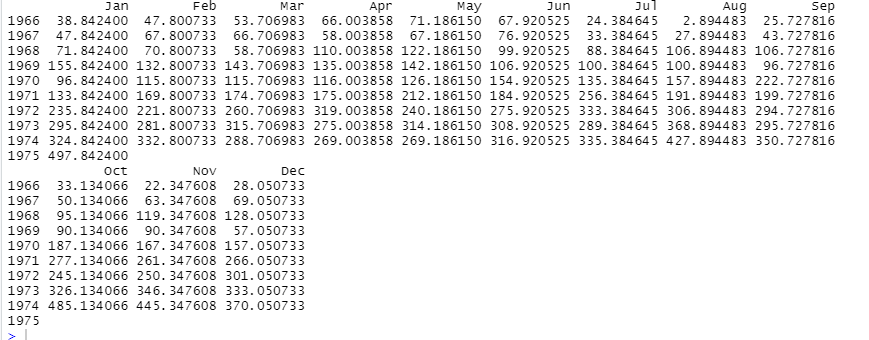
plot(dytime)



Bileşenlerin mevsimsel olarak düzeltip, değerlerini görmek istersek;

dytimeadjusted <- dytime - dycomponents$seasonal

dytimeadjusted



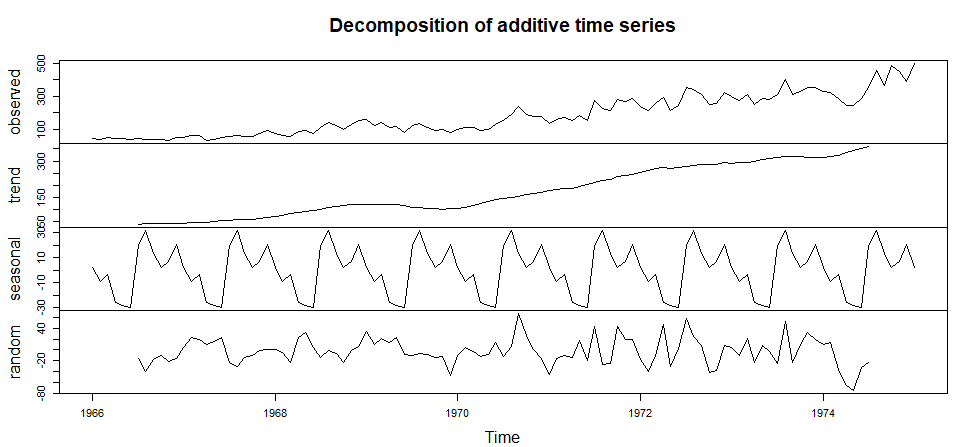
Burada en büyük mevsimsel faktör 1975 yılının Ocak ayının ayına aittir (yaklaşık 498) ve en düşük değer 1966 yılının Temmuz ayına aittir (yaklaşık 3),

Zaman serisinin tahmini trendini, mevsimsel ve düzensiz bileşenlerini "plot()" işlevini kullanarak R programlama da çizebiliriz;

dycomponents <- decompose(dytime)

dycomponents

plot(dycomponents)



Tüm bileşenleri tek bir tablo üzerinden görebiliriz. Mevsimsel, trend, rassal ve gözlenen olarak bileşenlerin grafikleri tablodan okumak mümkündür.

ii. Otokorelasyon Grafiği:

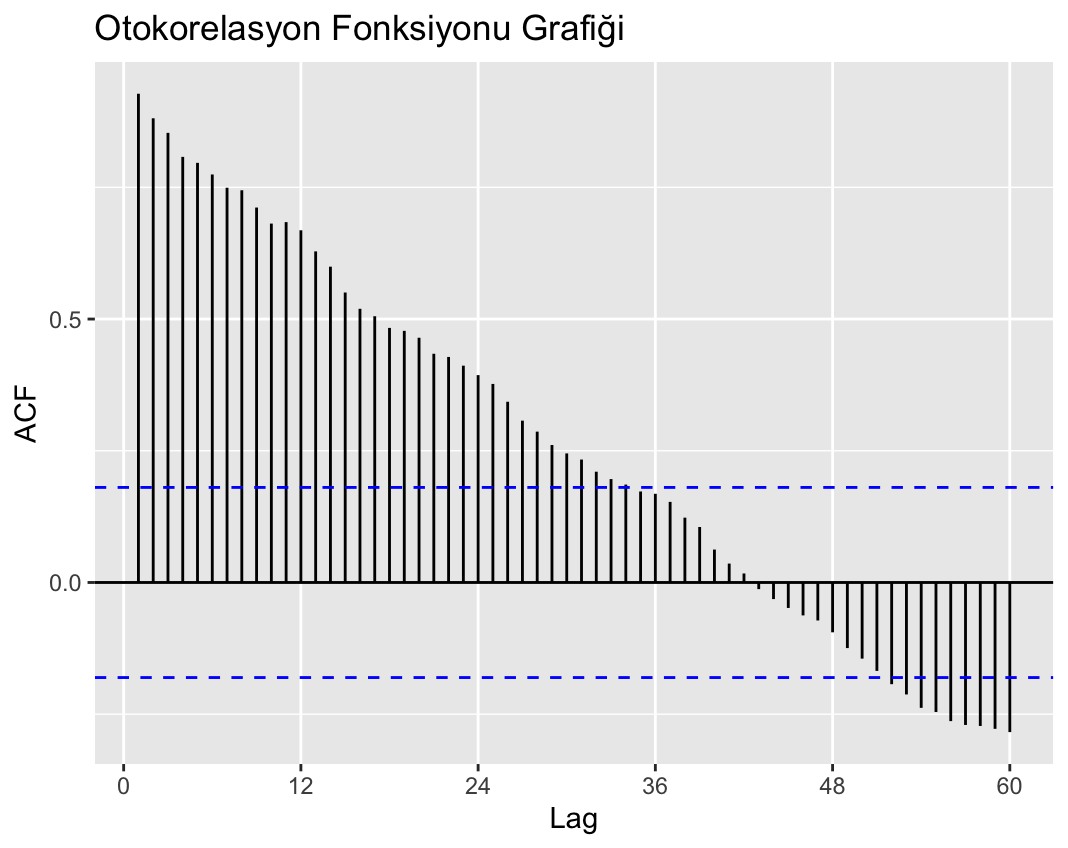
ggAcf(rob\_ts, lag = 60) +

ggtitle("Otokorelasyon Fonksiyonu Grafiği")

ggAcf(rob\_ts, lag = 60) +

ggtitle("Otokorelasyon Fonksiyonu Grafiği")





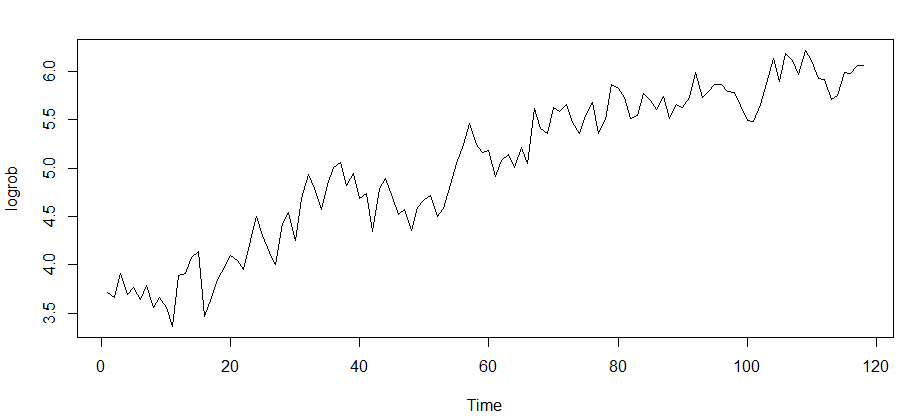
Otokorelasyon, bir zaman serisinin geçmiş değerleri ile arasındaki doğrusal ilişkinin derecesini ölçer. Nasıl iki değişken arasındaki doğrusal ilişkinin derecesini korelasyon ile ölçüyorsak, buradaki ikinci değişkenin de elimizdeki zaman serisinin geçmiş değerleri olduğunu düşünebiliriz. Otokorelasyon da bu değişken ile serinin kendi değerleri arasındaki doğrusal ilişkiyi ölçer.Öncelikle, grafiğin genel anlamda azalarak ilerlediğini görüyoruz. Bu da aslında serinin geçmiş değerleri ile arasındaki otokorelasyonun gittikçe düştüğünü gösteriyor.

iii. Logaritmik Dönüşüm:

Sırada verimizin logaritmik dönüşümü var. Veri logaritmik dönüşümle verilerdeki büyüme oranını dengelemek veya değişen varyansı stabilize edilir.

logrob<- log(rob\_ts)

plot.ts(logrob)



Bu grafikte gittikçe artan bir logaritmik dönüşüm görmekteyiz ve bu da veri setinde ki i büyümenin logaritmik bir şekilde arttığını göstermekte.

iv. Autoplot:

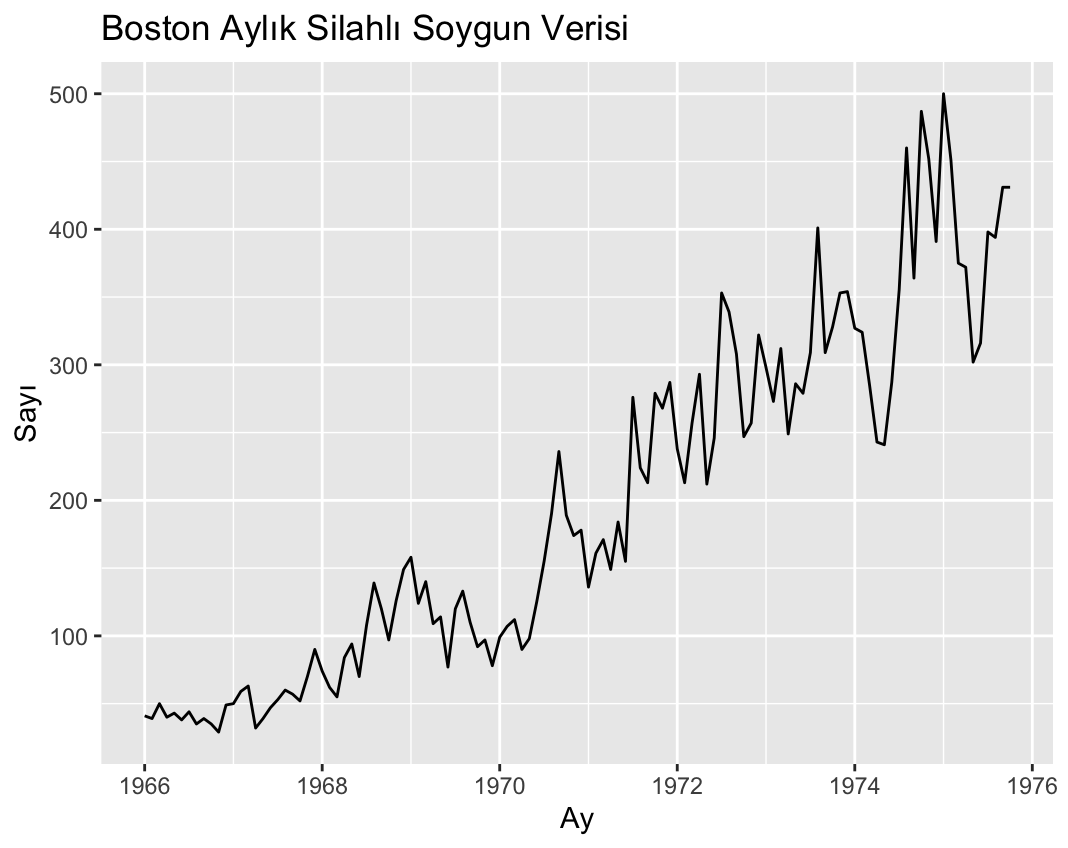
“autoplot” komutu verinin grafiğe dökülmesini sağlıyor. Autoplot, otomatik olarak içerisine verilen nesneye uygun bir grafik bulmaya çalışır. Zaman serisine en uygun grafik de çizgi grafiğidir.

autoplot(rob\_ts) +

ggtitle("Boston Aylık Silahlı Soygun Verisi") +

xlab("Ay") +

ylab("Sayı")



Grafiğe baktığımızda yukarı yön ve oldukça yüksek bir trendin var olduğunu görebiliyoruz.Yani verinin değişkenliği serinin farklı seviyelerinde çoğu kez değişiyor.1966 yılında soygun sayısı 50’nin altındayken, 1975’te 500 civarlarına kadar çıktığını görebiliyoruz. Bu da trendimizin büyüklüğünü ve suç sayısındaki artışı ortaya koyuyor.

**3) NAIVE METHOD**

Zaman serileri analizinde yaygın olarak kullanılan yöntemlerden biri "naive method" veya "naive forecast" olarak adlandırılan basit bir yöntemdir. Bu yöntemde, gelecekteki değerleri tahmin etmek için yalnızca en son gözlemin kullanıldığı basit bir yaklaşım benimsenir.

Naive method, zaman serisinin trendi, mevsimselliği veya diğer yapısal özellikleri göz önünde bulundurmadan, yalnızca en son gözlem değerini gelecekteki tüm dönemler için tahmin olarak kullanır. Şimdi R programlamada naive metodu yazalım

f\_Linear <- forecast(Model\_Linear, h = 22, newdata = data.frame(easter = easter(test\_ts)))

f\_Mean <- meanf(train\_ts, h = 22)

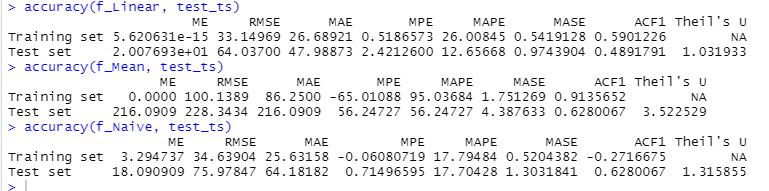
f\_Naive <- naive(train\_ts, h = 22)

accuracy(f\_Linear, test\_ts)

accuracy(f\_Mean, test\_ts)

accuracy(f\_Naive, test\_ts)

“accuracy” fonksiyonu, bizim için gerekli hata metriklerini hesaplayacak. Dikkat edelim ki tahminlerin hata metriklerini test set üzerinde hesaplıyoruz.



MAE değeri lineer model için 26.7 iken, ve Naive model için ise 25.6. RMSE değeri ise lineer model için 33.14, ve Naive model için 34.63. Bu durumda MAE baz alındığında en iyi model Naive iken, RMSE baz alındığında en iyi model lineer model olarak karşımıza çıkıyor.

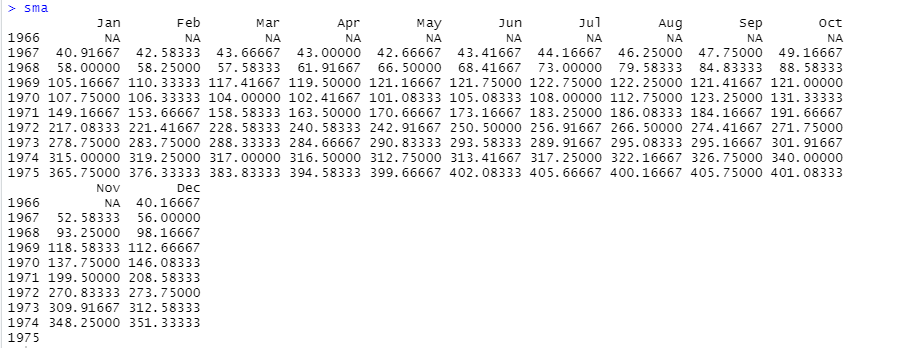
**4) HAREKETLİ ORTALAMALAR:**

1)Basit Hareketli Ortalama

Hareketli ortalama (moving average), belirli bir zaman periyodu boyunca bir veri serisinin ortalamasını hesaplamak için kullanılan bir istatistiksel yöntemdir. Hareketli ortalama, verilerin düzensiz dalgalanmalarını düzeltmek ve genel bir eğilimi göstermek için kullanılabilir.R programında hesaplamasını yaparsak;

sma <- SMA(rob\_ts, n = 12)

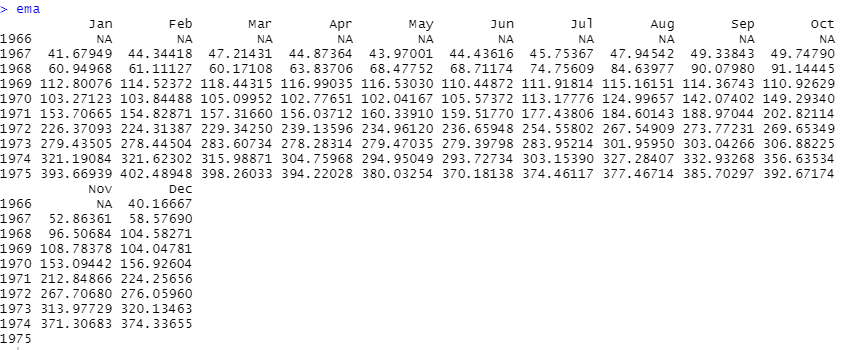
sma



2)Üssel Hareketli Ortalama

ema <- EMA(rob\_ts, n = 12)

ema

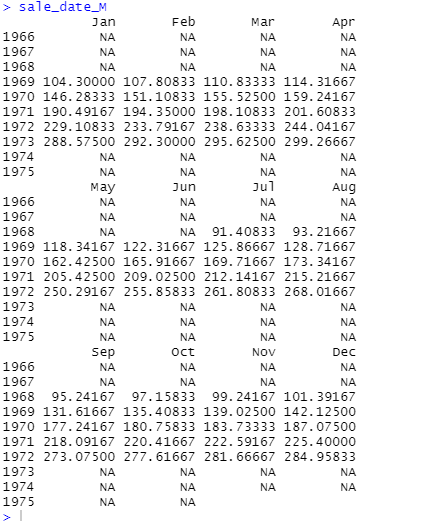


3) Merkezi Hareketli Ortalama

sale\_date\_M<-ma(rob\_ts,order=60,centre=TRUE)

logrob<- log(rob\_ts)

plot.ts(logrob)



Merkezi Hareketli Ortalama (Centered Moving Average), veri serisindeki dalgalanmaları ortadan kaldırmak ve düzgünleştirme yapmak için kullanılan bir istatistiksel yöntemdir.

Hareketli ortalama, belirli bir zaman periyodu boyunca verilerin ortalamasını alarak bir trend çizgisini hesaplar.

Bu grafikte de verimizin güçlü bir trende sahip olduğunu görüyoruz.

ZAMAN SERİSİNDE BİLEŞENLER

Trend: Trend, zaman serisinin genel bir yükselme veya düşüş eğilimini temsil eder. Trend, uzun vadeli değişimleri ifade eder ve serinin temel yapısını oluşturur. Trend genellikle doğrusal veya düzgün bir şekilde artan veya azalan bir eğri olarak görülür.

Mevsimsellik: Mevsimsellik, zaman serisinde düzenli aralıklarla tekrarlanan desenleri ifade eder. Örneğin, bir yıl içinde aylar veya mevsimler arasında tekrarlanan belirli bir desen mevcut olabilir. Mevsimsellik, özellikle mevsimsel verilerin analizinde önemlidir.

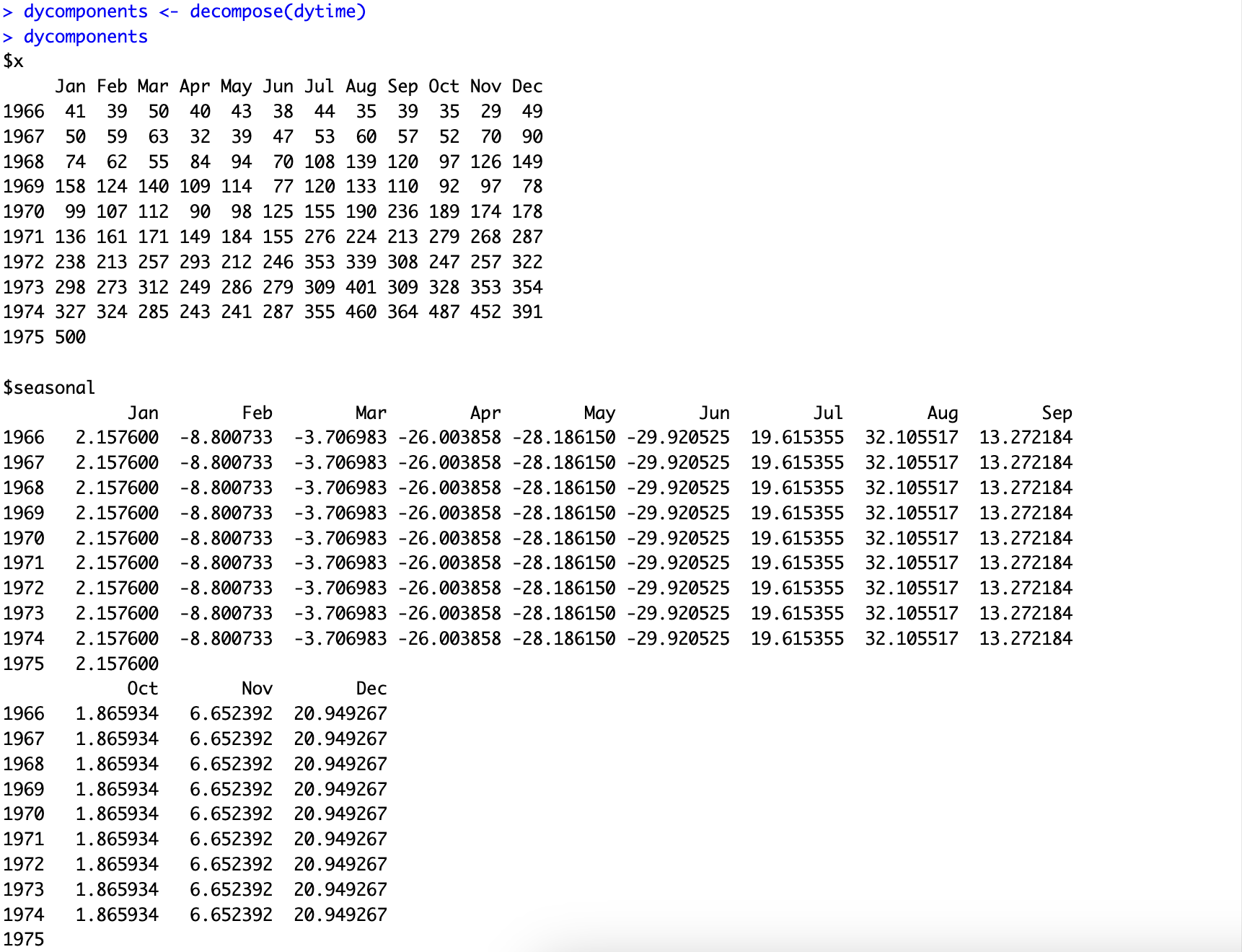
Döngü: Döngü, uzun vadeli dalgalanmaları ifade eder. Trendden daha uzun bir periyoda sahip olan döngüler genellikle ekonomik veya iş döngüleri olarak bilinir. Döngüler genellikle trende göre daha düzensiz ve daha uzun vadeli değişimlerdir.

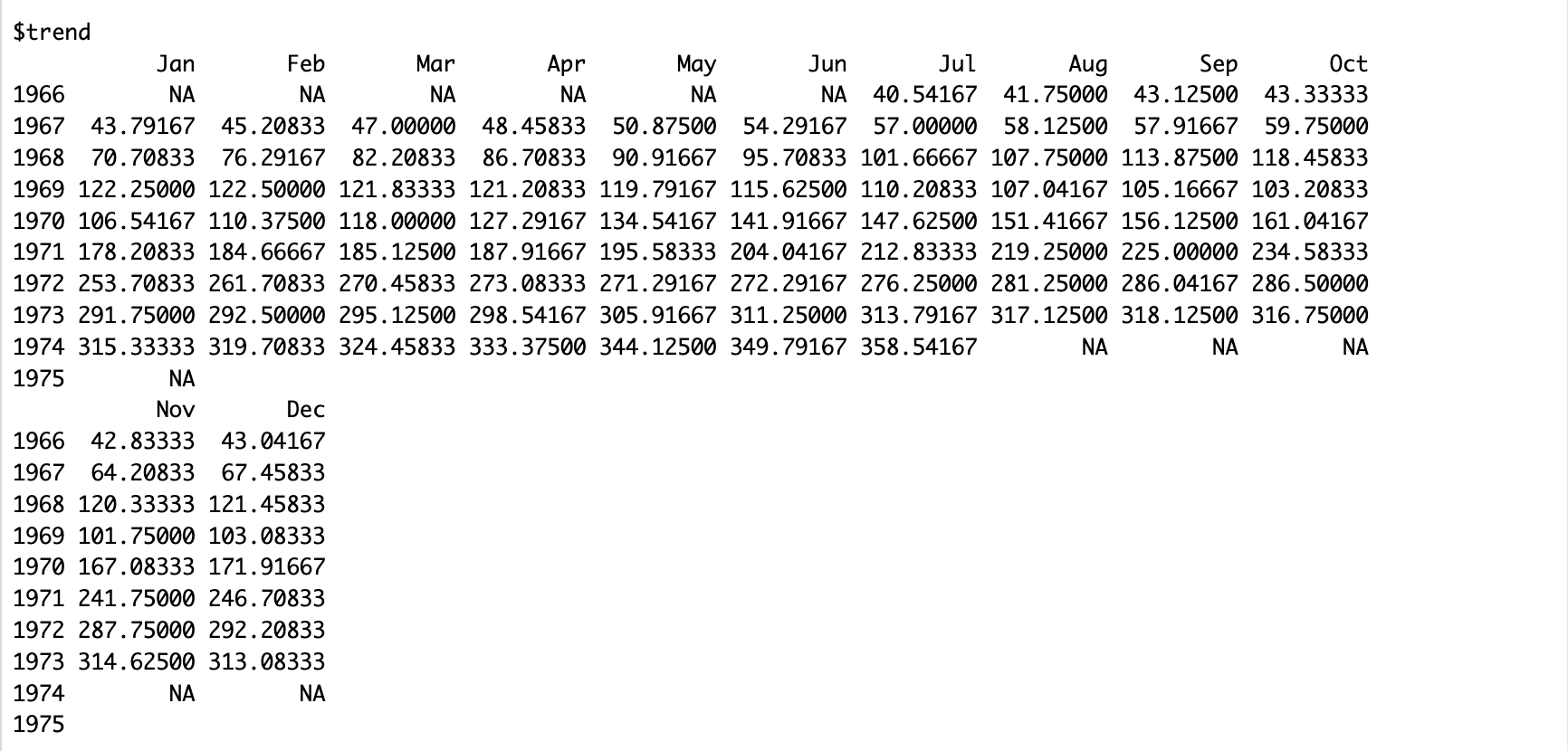
İstenmeyen Değişimler veya Gürültü: Zaman serisindeki rastgele veya istenmeyen dalgalanmaları ifade eder. Bu gürültü bileşeni, trend, mevsimsellik veya döngüden bağımsız olarak ortaya çıkabilir ve zaman serisindeki istatistiksel dalgalanmalardır.

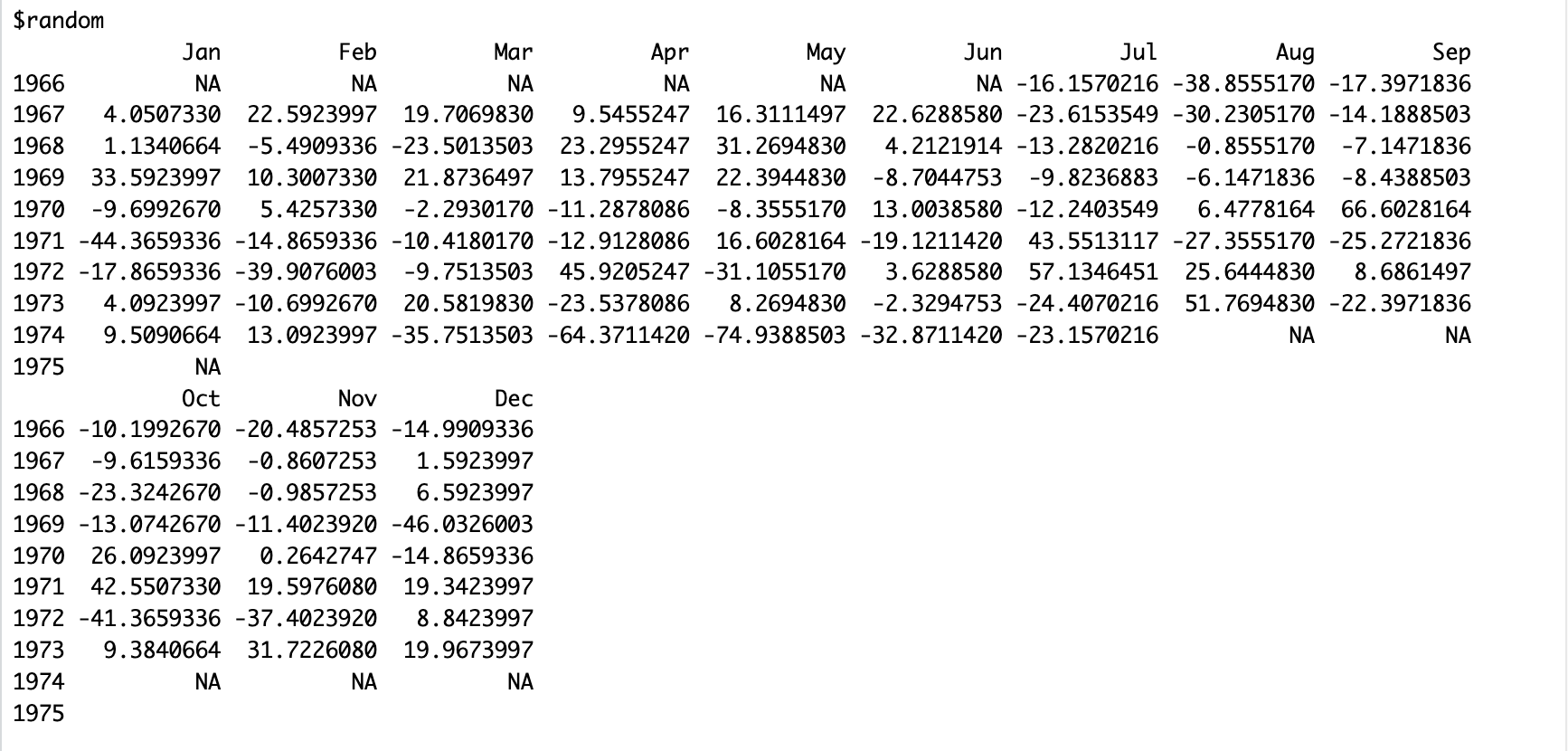
R programında aşağıdaki yazım ile 4 bileşenin değerlerini inceleyebiliriz;

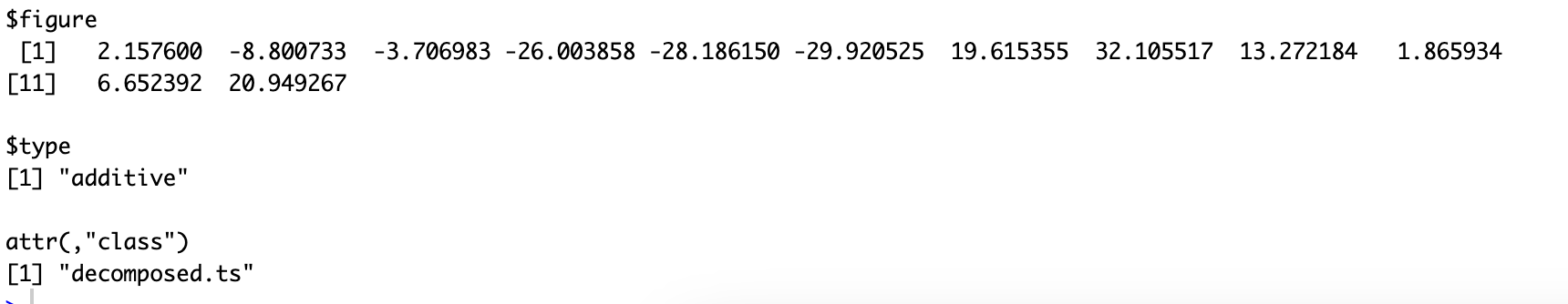
dycomponents <- decompose(dytime)

dycomponents



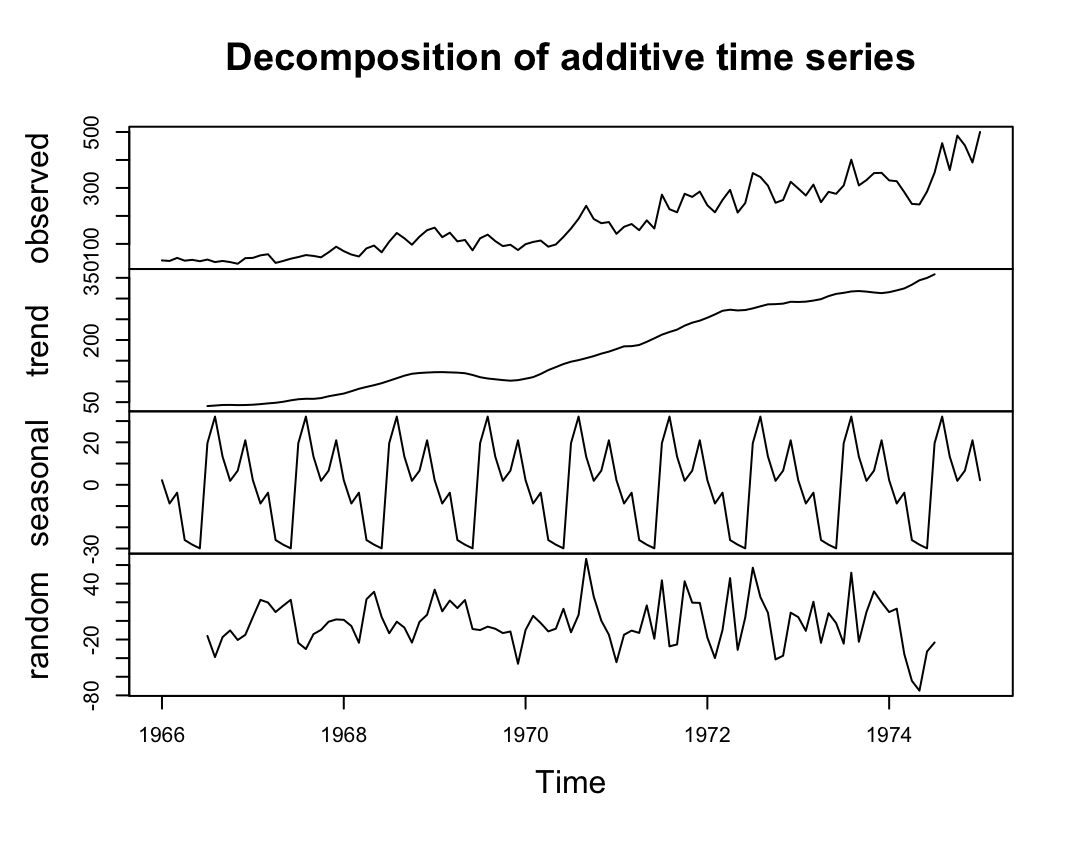






Grafik ile tek bir tabloda görselleştirmek istersek;

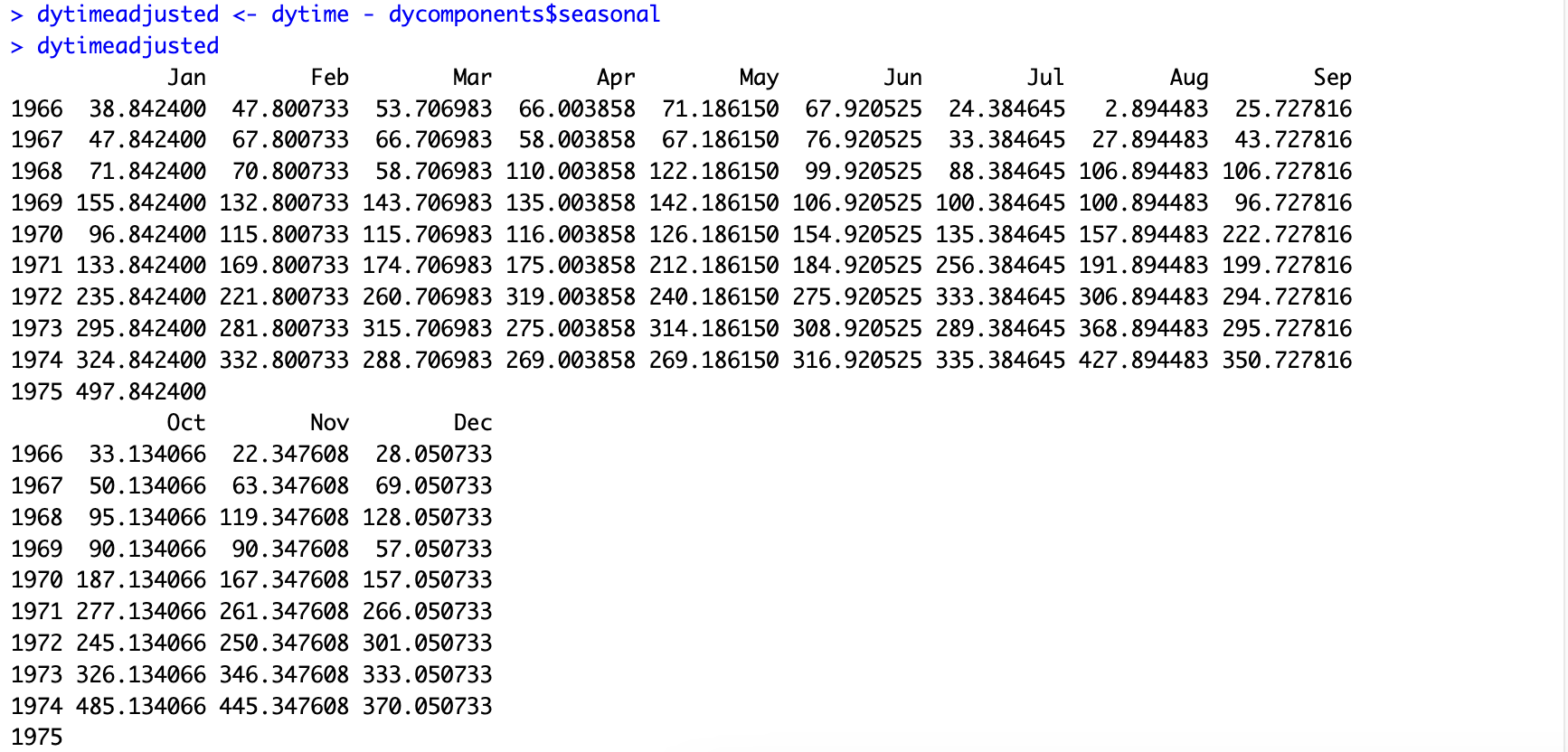
plot(dycomponents)



Mevsimsel Bileşen

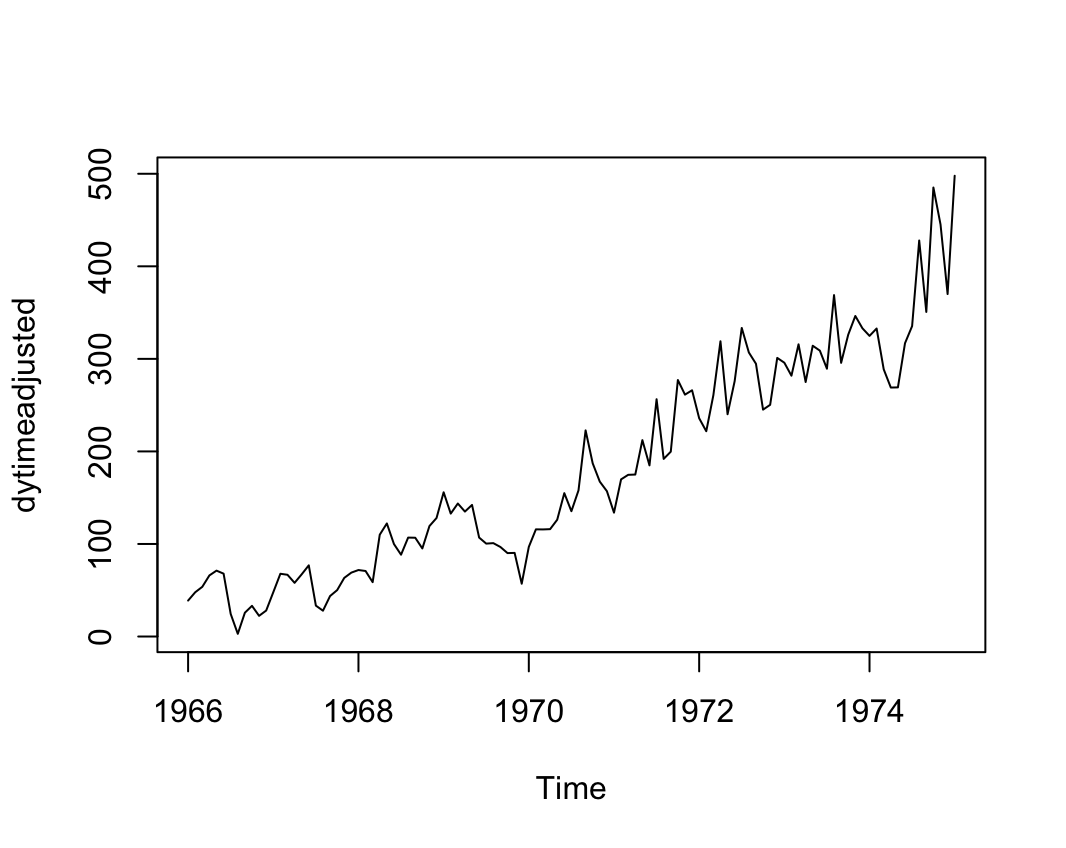
dytimeadjusted <- dytime - dycomponents$seasonal

dytimeadjusted



Düzeltilmiş zaman serisi grafığı:

plot(dytimeadjusted)



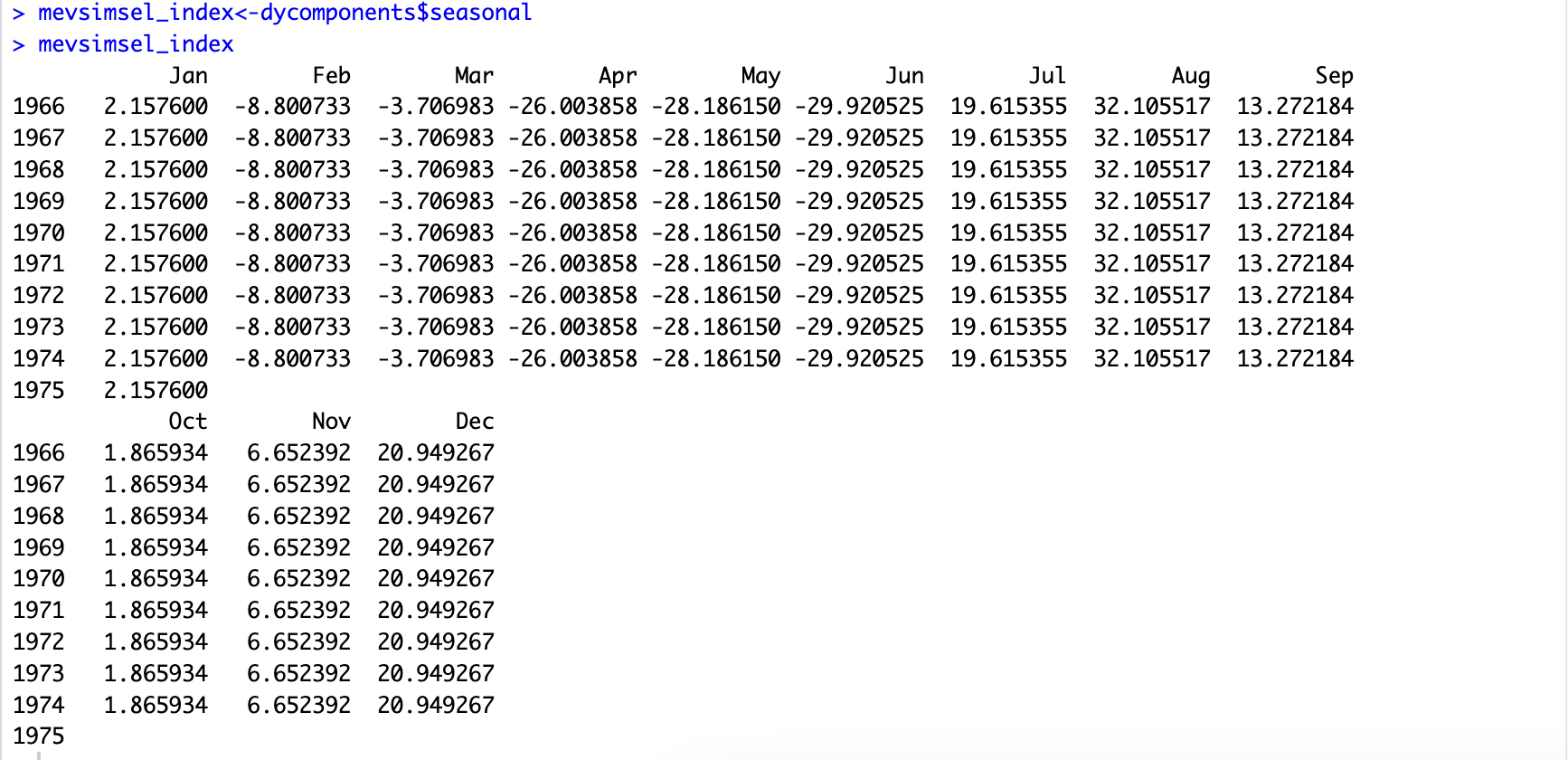
Mevsimsel İndeks

Mevsimsel indeks, bir zaman serisinin mevsimsel etkilerini ölçmek ve analiz etmek için kullanılan bir ölçüdür. Mevsimsel indeks, bir zaman serisi içinde belirli bir mevsimsel dönemin (genellikle bir yıl) ortalama etkisini ifade eden bir değerdir. Bu indeks, bir mevsimsel bileşenin göreceli gücünü veya etkisini nicel olarak ifade etmek için kullanılır.

Mevsimsel indeksler, genellikle 0 ile 100 arasında bir ölçek üzerinde ifade edilir. Mevsimsel indeks değeri 100'e yaklaştıkça, mevsimsel etkinin gücü artar. Tam tersi olarak, 0'a yaklaştıkça mevsimsel etki zayıflar.

mevsimsel\_index<-dycomponents$seasonal

mevsimsel\_index



Mevsimsel indeksi en yüksek olan Agustos ayıdır.Mevsimsel etkinin gücü bu ayda etkilidir.

Model Eğitimi:

Modelleri kurmadan önce ilk aşama olarak veriyi train ve test setler olarak ikiye ayırıyoruz. Modelimizi train setiyle eğitiyoruz. Daha sonra modelin performansını test set üzerinde ölçüyoruz..

train\_ts <- window(rob\_ts, end = c(1973, 12))

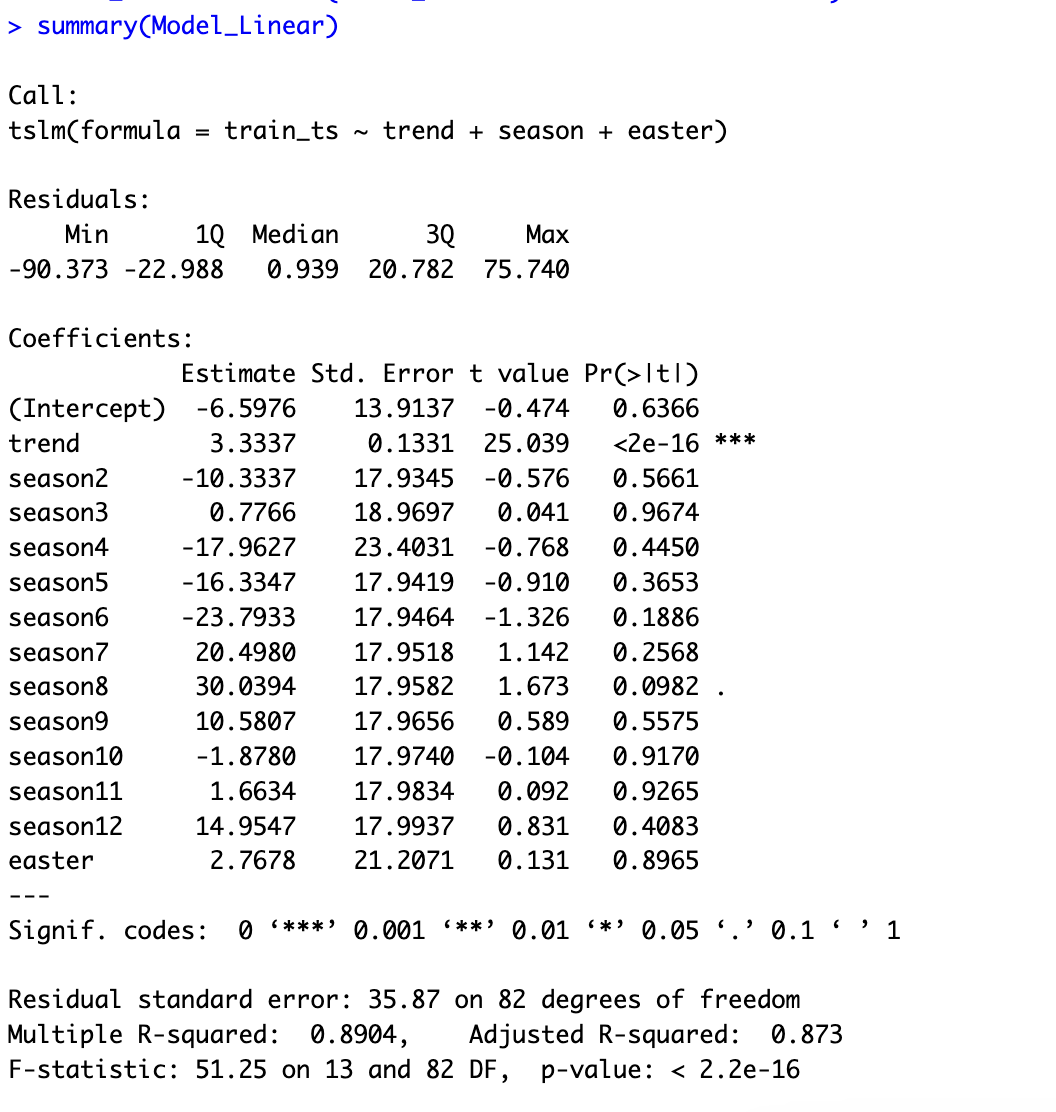
test\_ts <- window(rob\_ts, start = c(1974, 01))

Aşağıdaki kod bloğuyla training set üzerinde ilk modeli kuruyoruz. “easter” fonksiyonu, üstte de bahsettiğim paskalya bayramının sene içerisinde ne tarihe denk geldiğini hesaplayıp o tarihteki gözlemlere “1” , diğerlerine “0” değeri atayarak “easter” değişkenini oluşturuyor. Bu şekilde kukla değişken oluşmuş durumda. Bazı gözlemlerde 1 değeri dışında ondalıklı sayılıar bulunuyor ki bu da paskalya bayramının iki aya yayıldığı durumlarda oluyor. Bu değişkeni oluşturmanın amacı, üstte de söylediğimiz gibi kukla değişken kullanımını göstermek ve tatil günlerinde de suç sayısının artabileceğini düşünmek. Daha çok gösterim amacıyla kullanıyoruz.

easter<- easter(train\_ts)

Model\_Linear <- tslm(train\_ts ~ trend + season + easter)

summary(Model\_Linear)



Trend değişkenine bakacak olursak, seride aydan aya ortalama 3.33 adetlik yükselen bir trend olduğunu görüyoruz. Buradaki adet tabii ki soygun sayısına denk geliyor. Ayrıca, ikinci ay birinci aya göre ortalama 10.33 adet daha düşük soyguna sahipken, onikinci ay birinci aya göre ortalama 14.95 adet daha yüksek soyguna sahip.Veri aylık bazda olduğu için, mevsimsellik değişkeni de bize mevsimselliği aylık bazda veriyor.Easter değişkeni ise paskalya bayramının olduğu tarihlerde ortalama 2.77 adet daha fazla soygun görüldüğünü gösteriyor.

v. Metod Karşılaştırması:

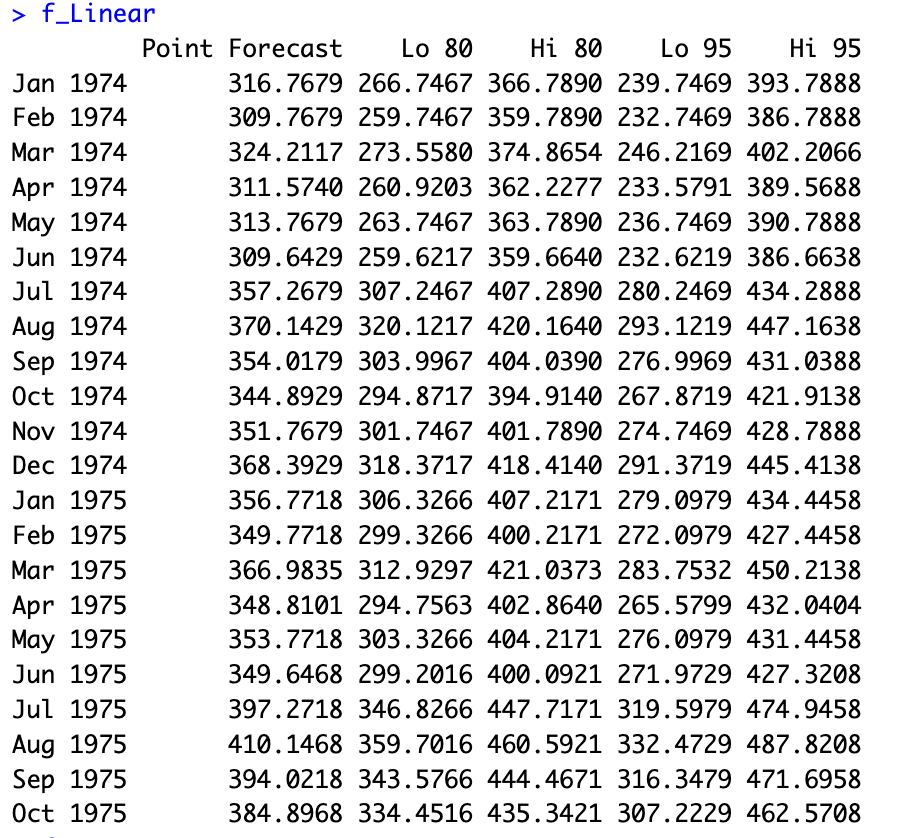
Kısaca değinecek olursak, ortalama metodu, tahminlenecek olan bağımlı değişkeni, serinin geri kalanının ortalamasına eşit kabul eder ve o şekilde tahmin eder. Yani elinizdeki zaman serisinin ortalamasını alır ve bu serinin bir sonraki değerini de o ortalama değeri olarak tahmin etmiş olur.

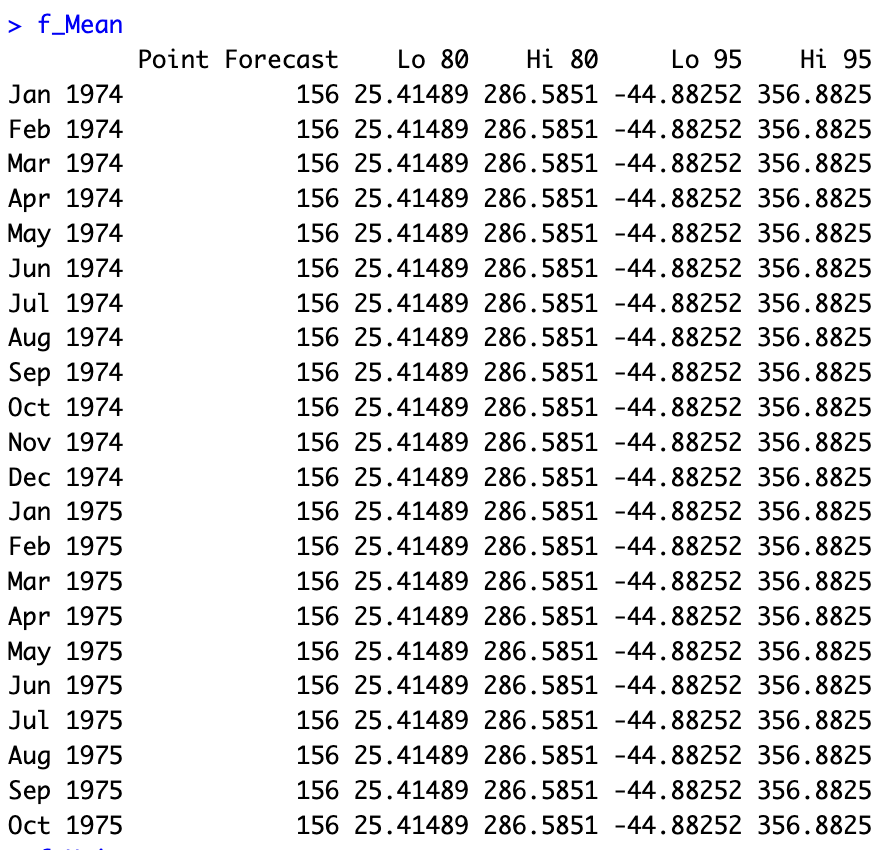
Naive metodda ise elinizdeki serinin son değerini, direkt olarak tahmin değeri olarak belirlersiniz.

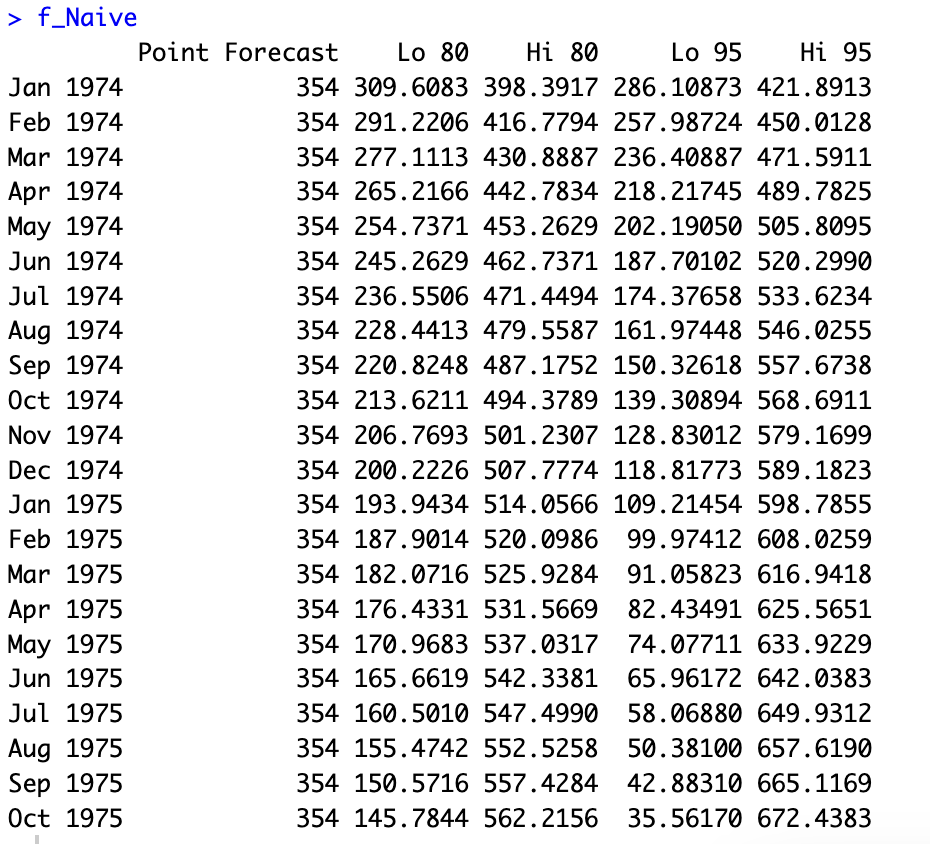
f\_Linear <- forecast(Model\_Linear, h = 22, newdata = data.frame(easter = easter(test\_ts)))

f\_Mean <- meanf(train\_ts, h = 22)

f\_Naive <- naive(train\_ts, h = 22)





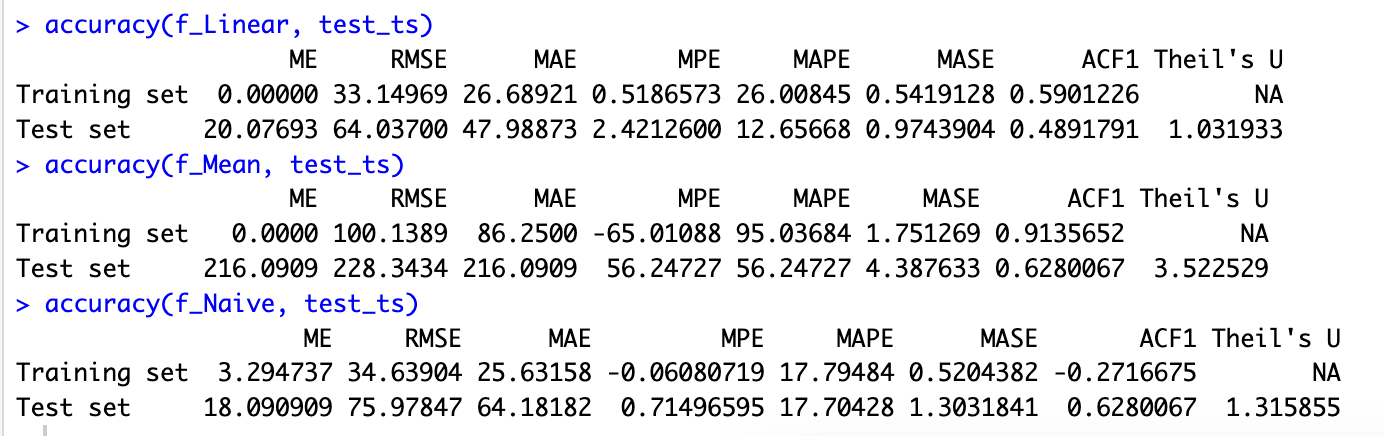


“accuracy” fonksiyonu, bizim için gerekli hata birimlerini hesaplayacak.

accuracy(f\_Linear, test\_ts)

accuracy(f\_Mean, test\_ts)

accuracy(f\_Naive, test\_ts)



Bu durumda MAE baz alındığında en iyi model Naive iken, RMSE baz alındığında en iyi model lineer model olarak karşımıza çıkıyor. Lineer model MAE ve RMSE bazında daha iyi sonuçlar ortaya çıkardığını söyleyebiliriz.

**REFERANSLAR**

* <https://ec.europa.eu/eurostat/databrowser/view/tec00118/default/table?lang=en>
* <https://tim.org.tr/files/downloads/Strateji_Raporlari/TIM_Ihracat_2021_Raporu.pdf>
* <https://www.kaggle.com/datasets/sujan97/monthly-armed-robberies-in-boston>
* <https://www.tarimorman.gov.tr/Sayfalar/AnaSayfa.aspx>