

DİNAMİK REGRESYON MODELİ İLE ZAMAN SERİLERİ ANALİZİ

ELİF EKMEKÇİ

2023-06-04

```
library(readxl)
library(lubridate)
library(fpp2)
df <- read_excel("/Users/elif/Desktop/zamanserileri_analizi/altindata.xlsx")
dat <- ts(df[, -1], start=c(2013, 1), frequency = 12)
# date degiskenini cikarip diger 3 degiskeni ts nesnesi olarak tanımladık
```

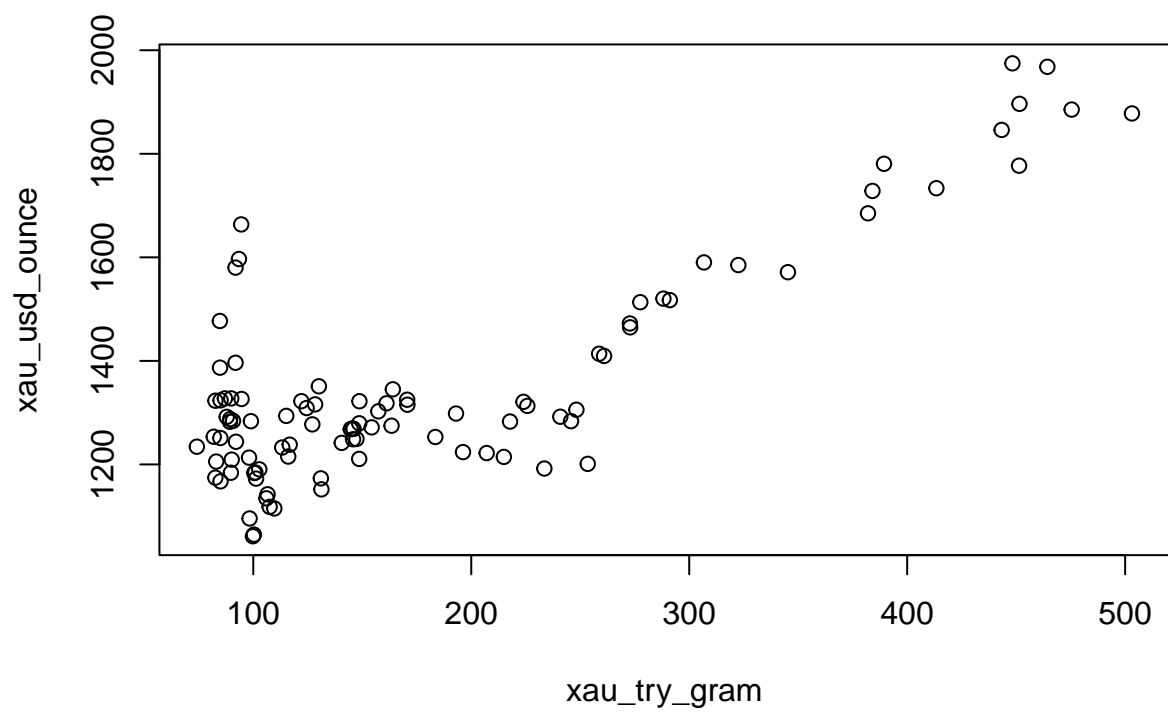
Verimizi incelediğimizde aylık bir veri olduğunu görüyoruz.

Değişkenler arasındaki ilişkileri incelemek için saçılım grafiği çizdirelim.

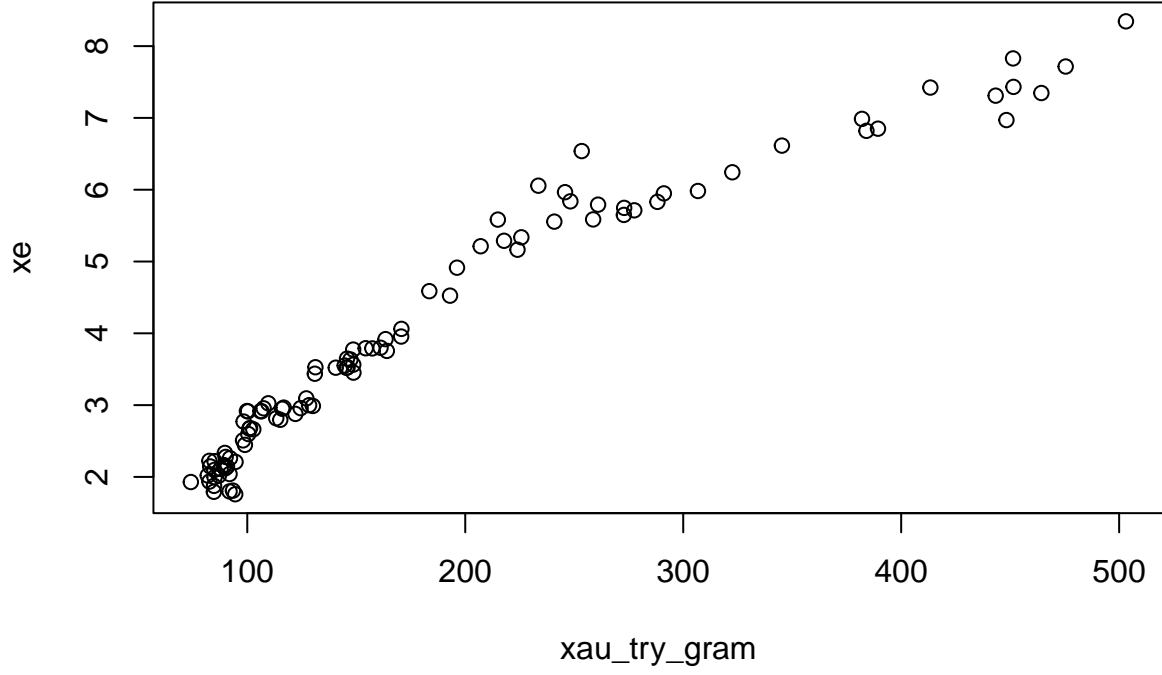
NOT: t ve tkare kullanımına bu adımda karar veriyoruz

İlişki her zaman lineer olmak zorunda değildir. Kuadratik olduğu durumlarda değişkenin hem kendisini hem de karesini modele dahil edebiliyoruz

```
plot(xau_usd_ounce~xau_try_gram, dat)
```



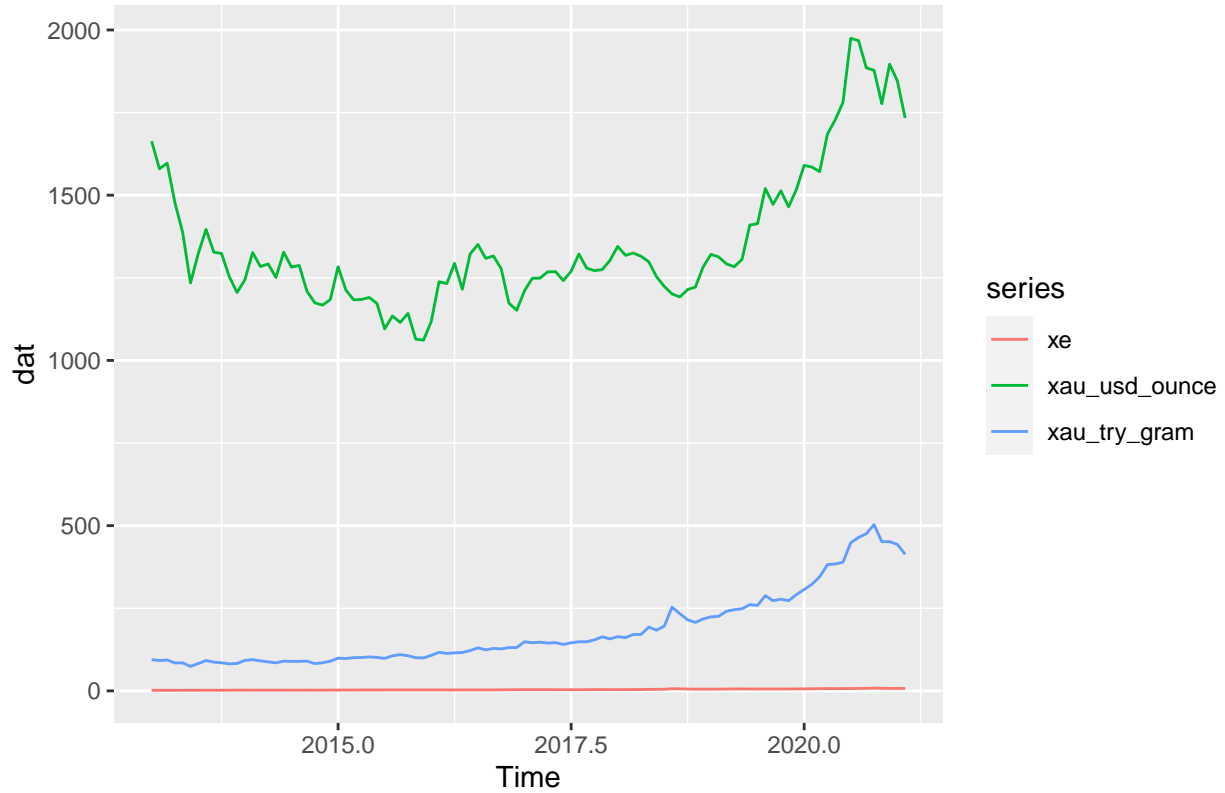
```
plot(xe~xau_try_gram,dat)
```



xau_try_gram ile xe arasındaki ilişki lineer gözüküyor. xau_try_gram değişkeni burada yanıt değişkeni olarak kullanılacak. Diğer grafiği de incelediğimizde lineer gibi gözüküyor. **Bu nedenle direkt değişkenlerin kendilerini kullanabiliriz karelerini küplerini vs kullanmamıza gerek yok**

Değişkenlerimizin autoplot grafiğini çizdirelim

```
autoplot(dat)
```



“xau_try_gram” değişkeni yanıt değişken “xe” ve “xau_usd_ounce” ise bu değişkeni etkileyen değişkenler olacak.

Bunu yapmamız için bizim bir x matrisi oluşturmamız gerekiyor. x matrisi için de xe’yi ve xau_usd_ounce’yi birleştirip bir matris oluşturmamız gerekiyor.

```
xreg <- cbind(xe = dat[, "xe"],xau_usd_ounce = dat[, "xau_usd_ounce"])
# Bunu yapmamız için bir x matrisi oluşturmamız gerekiyor.
# x matrisi için de xe'yi ve xau_usd_ounce'yi birleştirip bir matris oluşturmamız gerekiyor.
fit <- auto.arima(dat[, "xau_try_gram"], xreg = xreg)
# model tahmini yapıyorum (daha sonra tahmin için değişken tahmini yapılacak)
# auto.arima ile xau_try_gram(yanıt değişkeni) için en uygun modeli arıyoruz
# auto.arima icine oluşturdugumuz matrisi yazıyoruz (xreg =xreg) aslında aciklayici degisken matrisi
# auto.arima ile fit adında bir model kurduk
# burada auto.arima dışında farklı modeller de kullanılabilir
summary(fit)
```

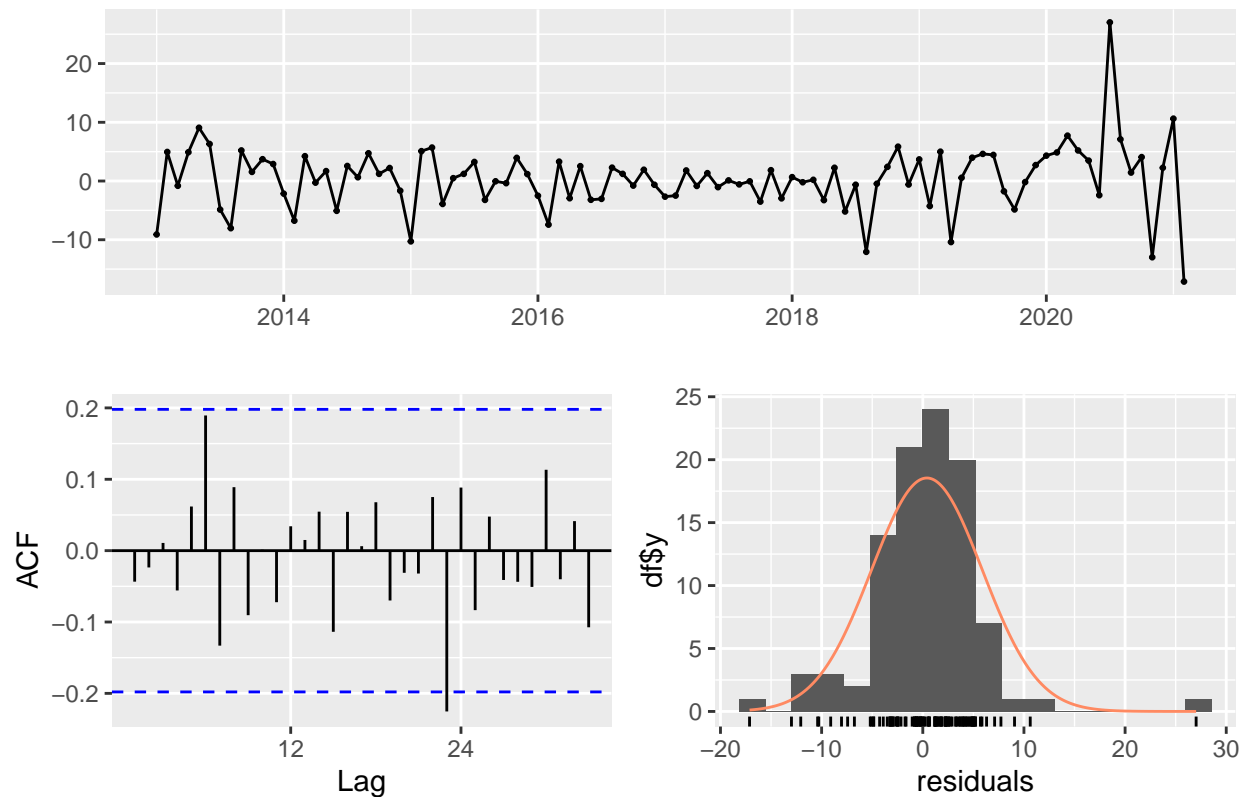
İLK ADIMDA YANIT DEĞİŞKENLERİNİ BİRLEŞTİRİP (xreg) TAHMİN DEĞİŞKENİ İÇİN MODEL KURDUK

```
## Series: dat[, "xau_try_gram"]
## Regression with ARIMA(3,0,0) errors
##
## Coefficients:
```

```
##          ar1      ar2      ar3  intercept      xe  xau_usd_ounce
##      0.8077  0.3888 -0.2562 -182.1586  45.3231      0.1354
## s.e.  0.1102  0.1318   0.1090   19.9919   2.4002      0.0095
##
## sigma^2 = 31.78:  log likelihood = -306.66
## AIC=627.32  AICc=628.56  BIC=645.41
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.4063972 5.461665 3.828073 0.2063788 2.522838 0.07958014
##              ACF1
## Training set -0.0434574
```

```
checkresiduals(fit)
```

Residuals from Regression with ARIMA(3,0,0) errors



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 12.747, df = 17, p-value = 0.7529
##
## Model df: 3.   Total lags used: 20
```

```
library(olsrr)
ols_test_normality(residuals(fit))
```

```
## -----
##          Test          Statistic      pvalue
## -----
## Shapiro-Wilk          0.9089        0.0000
## Kolmogorov-Smirnov     0.1087        0.1833
## Cramer-von Mises       7.0337        0.0000
## Anderson-Darling       1.7403        2e-04
## -----
```

```
nrow(dat)
```

```
## [1] 98
```

summary(fit) çalıştırıldığında **ARIMA(3,0,0)** modelini buldu. Mevsimselliği olmayan bir model. MA kısmı da 0 fark almaya gerek görmedi. **R 3 model** kurdu. Yani y'nin 3 gecikmeliyle model kuruldu

log likelihood = -306.66 olarak hesaplandı. log likelihood değeri ne kadar yüksekse o model bizim için daha iyidir. (en çok olabilirlik metriği)

Şimdi 8 aylık tahmin yapalım. Bunun için önce "xau_usd_ounce" ve "xe" serileri için 8 aylık tahmin yapmalıyız.

2.ADIMDA AÇIKLAYICI DEĞİŞKENLER İÇİN AYRI AYRI MODEL KURUP FORECAST YAPIYORUZ (veriyi test ve train olarak ayırmadık bu yüzden tüm veriyi kullanarak tahmin yapıyoruz yani bu tahmin de train seti tahmini oluyor) **## forecast** yapılacak ancak önce elimizdeki açıklayıcı değişkenlerin tahminini yapıp daha sonra bu tahminlerin tahminini yapıyoruz **##**

```
# aciklayici degiskenlerimiz "xau_usd_ounce" ve "xe" icin auto.arima ile tahmin modeli kurduk
# burada aciklayici degiskenlerin tahmini icin auto.arima kullanildi
fitusd1 <- auto.arima(dat[, "xau_usd_ounce"])
fitxe1 <- auto.arima(dat[, "xe"])
# icine belirtmemiz gerekiyor xe ve xau_usd_ounce (koseli parantez ile)

# 8 adimlik tahmin yapacagiz
# her ikisi icin 8 adimlik tahmin yapip matrise ekliyoruz
# yani forecast yapacagiz ama elimizdeki degiskenlerin de once forecastini yapıyoruz
fusd1 <-forecast(fitusd1,8)
# bu kodu calistirinca point forecast lo80 gibi degiskenleri veriyor (8 adimlik tahmin)
fxe1 <-forecast(fitxe1,8)
summary(fitxe1)
```

```
## Series: dat[, "xe"]
## ARIMA(0,1,0) with drift
##
## Coefficients:
##          drift
##          0.0584
## s.e.  0.0248
##
## sigma^2 = 0.06049: log likelihood = -1.08
## AIC=6.16   AICc=6.28   BIC=11.31
##
## Training set error measures:
```

```
##               ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 1.735215e-05 0.2434233 0.1443457 -0.3965158 3.263285 0.185725
##               ACF1
## Training set 0.0927775
```

```
summary(fitusd1)
```

```
## Series: dat[, "xau_usd_ounce"]
## ARIMA(0,1,0)
##
## sigma^2 = 3658: log likelihood = -535.57
## AIC=1073.13 AICc=1073.17 BIC=1075.71
##
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.7301388 60.17364 46.0683 -0.04940561 3.360069 0.3089706
##               ACF1
## Training set 0.005190971
```

ARIMA(0,1,0) random walk model yani sadece farka dayalı model dinamik regresyon modelimizin 8 adımlık tahminlerini(forecastlerini) elde edelim

Elimde var olan açıklayıcı değişkenlerle bir model kurduk. Kurduğumuz model üzerinden forecast yapacağız. Forecast yaparken x'lerin de forecast değerlerini almamız gerekiyor. Bu bulduğumuz forecastleri diğer forecast içerisinde kullanıyoruz.

Açıklayıcı değişkenlerimizi ne kadar iyi forecast ediyorsak aynı şekilde asıl değişkeni de onlar üzerinden iyi tahmin edeceğiz.

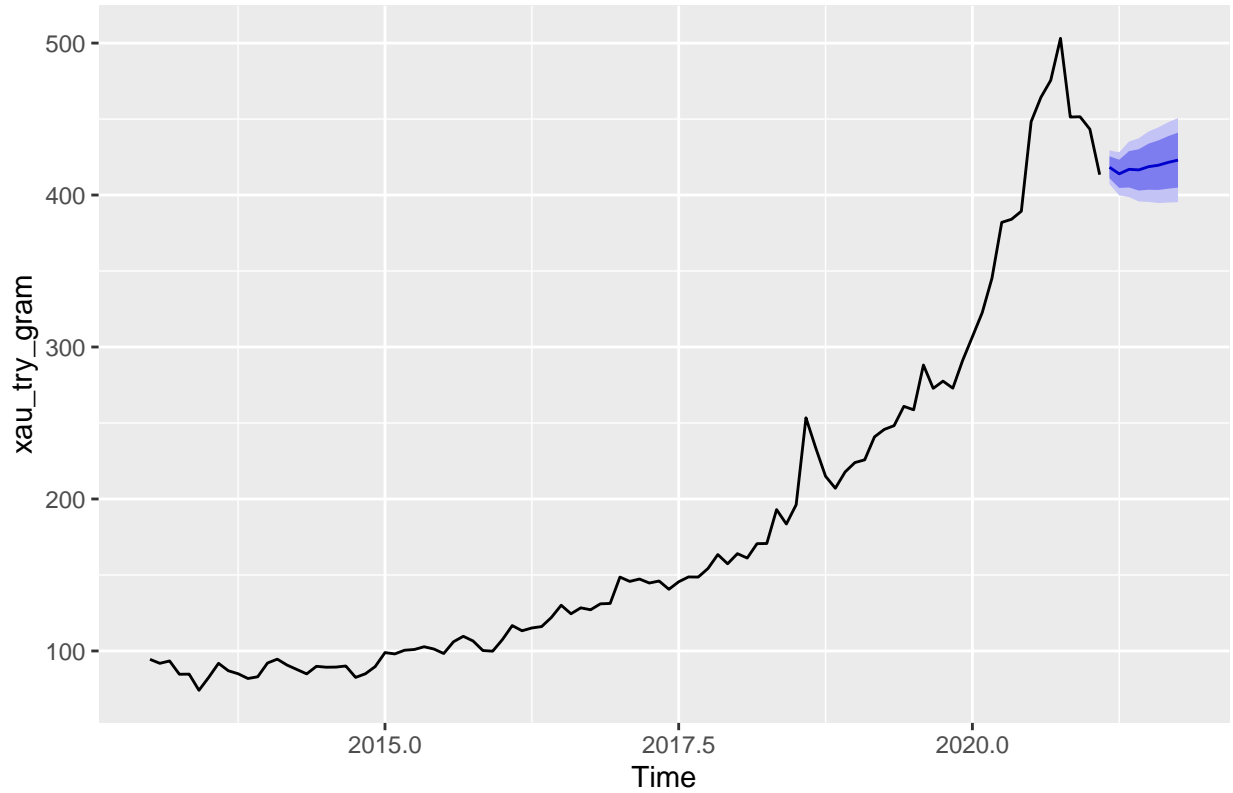
Burada açıklayıcı değişken auto.arima ile tahmin (forecast) edildi ancak ETS model ile ya da kübik spline ile de tahmin edilebilir. Ayrıca Model için auto.arima kullanabiliriz ancak açıklayıcı değişkenler için auto.arima kullanılacak diye bir şey yok yukarıdaki gibi modelleri kullanabiliriz.

bu adımda da değişkenlerimizin tahmin değerlerini kullanarak dinamik regresyon modelini tahmin edelim

DİNAMİK REGRESYON MODELİMİZİN 8 ADIMLIK TAHMİNLERİNİ BULALIM

```
fcast <- forecast(fit,xreg = cbind(xe=as.vector(fxe1$mean),xau_usd_ounce=as.vector(fusd1$mean)))
# fit auto.arima ile kurduğumuz model
# xe=as.vector(fxe1$mean)'yi bu şekilde yazıyoruz yani degiskenleri bu şekilde tanımlamamız gerekiyor
# böylelikle dinamik regresyon modelimizin 8 adimlik tahminlerini(forecastlerini) elde etmis olduk
autoplot(fcast) + ylab("xau_try_gram")
```

Forecasts from Regression with ARIMA(3,0,0) errors



TEST VE TRAIN OLARAK AYRILMIŞ VERİ

Şimdi aynı veriyi test ve train olarak ayıralım Verimizi ilk olarak bir önceki gibi modelleyelim daha sonra trendi de bir değişken olarak modele dahil edip iki modelin performanslarını karşılaştıralım.

```
traindat<-window(dat,end=c(2020,6))
# train set ile dinamik regresyon modeli kuruyoruz
# daha sonra test seti üzerine tahmin yapıyoruz
testdat<-window(dat,start=c(2020,7))
xreg <- cbind(xe = traindat[, "xe"], xau_usd_ounce = traindat[, "xau_usd_ounce"])
# degiskenleri kod icinde tanımladigimiza dikkat !!*****
# ayrica train seti kullandigimiza da dikkat !!!!!

##### DINAMIK REGRESYON MODELİ KURALIM #####
fittrain1 <- auto.arima(traindat[, "xau_try_gram"], xreg = xreg)
# traindat[, "xau_try_gram"] yanıt degiskenini koseli parantez icine yazdik
# dinamik regresyon modeli kurduk

fittrain1 <- auto.arima(traindat[, "xau_try_gram"], xreg = xreg,
                        stepwise = FALSE, approximation = FALSE)
##### auto arima hakkında hatırlatmalar #####
# approximation = FALSE yapmamızın sebebi daha çok model denemesini istememiz
### HATIRLATMA ###
# arimamod<-auto.arima(train,lambda="auto",stepwise = FALSE,approximation = FALSE)
# lambda = "auto" yapınca en uygun lambda degerini buluyor
```



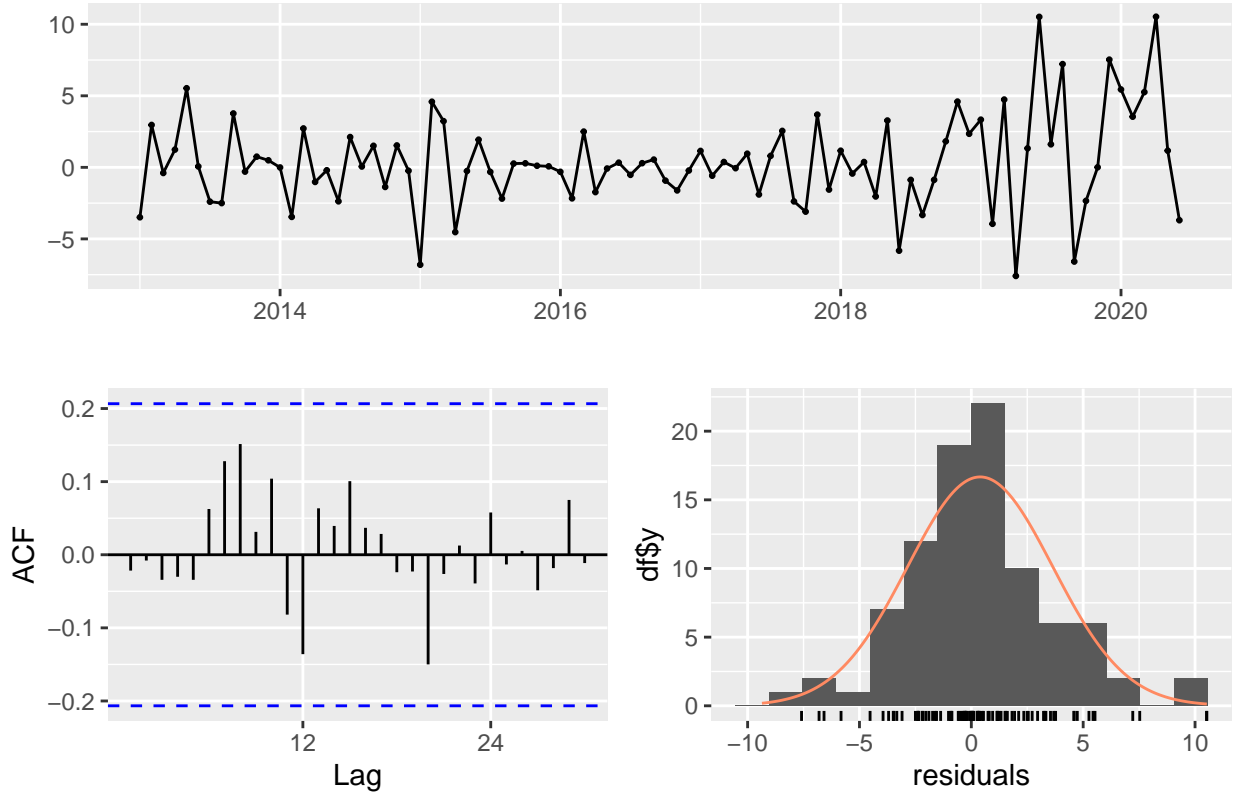
```
# lambda box-cox için gerekli auto yazınca en iyisini buluyor
# arima modelini belirledik
```

```
summary(fittrain1)
```

```
## Series: traindat[, "xau_try_gram"]
## Regression with ARIMA(2,0,3) errors
##
## Coefficients:
##          ar1      ar2      ma1      ma2      ma3  intercept          xe
##      -0.0026  0.9706  1.0428  0.5241  0.4036 -107.6417  41.1656
## s.e.   0.0517  0.0509  0.1233  0.1472  0.1399   24.6952   1.3559
##      xau_usd_ounce
##              0.0954
## s.e.         0.0061
##
## sigma^2 = 11.65: log likelihood = -236.61
## AIC=491.22  AICc=493.47  BIC=513.72
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.3954041 3.258174 2.31872 0.1289404 1.510989 0.06513657
##              ACF1
## Training set -0.0216642
```

```
checkresiduals(fittrain1)
```

Residuals from Regression with ARIMA(2,0,3) errors



```
##
## Ljung-Box test
##
## data: Residuals from Regression with ARIMA(2,0,3) errors
## Q* = 10.615, df = 13, p-value = 0.6431
##
## Model df: 5. Total lags used: 18
```

loglikelihood değerinin büyük olması iyidir. loglikelihood ne kadar büyükse o veri ile model o kadar uyumludur

`summary(fittrain1)` çıktısını incelediğimizde ARIMA(2,0,2)(1,0,0) buldu. Fark almaya gerek yok. Mevsimsellik var.

Saçılım grafiğine baktığımızda kuadratik bir yapı var. Bu nedenle buraya t^2 ekledik (aşağıdaki kod)

Trendli model

```
t<-ts(1:dim(traindat)[1],start=c(2013,1),frequency = 12) # train setin baslangic tarihini yaziyoruz
tkare<-t^2
xreg <- cbind(xe = traindat[, "xe"],xau_usd_ounce = traindat[, "xau_usd_ounce"],t,tkare)
# ek olarak t ve tkare eklendi

##### DINAMIK REGRESYON MODELİ KURALIM #####
```

```

# Biz burada dinamik regresyon için t ve t kare kullandık
fittrain2 <- auto.arima(traindat[, "xau_try_gram"], xreg = xreg,
                        stepwise = FALSE, approximation = FALSE)

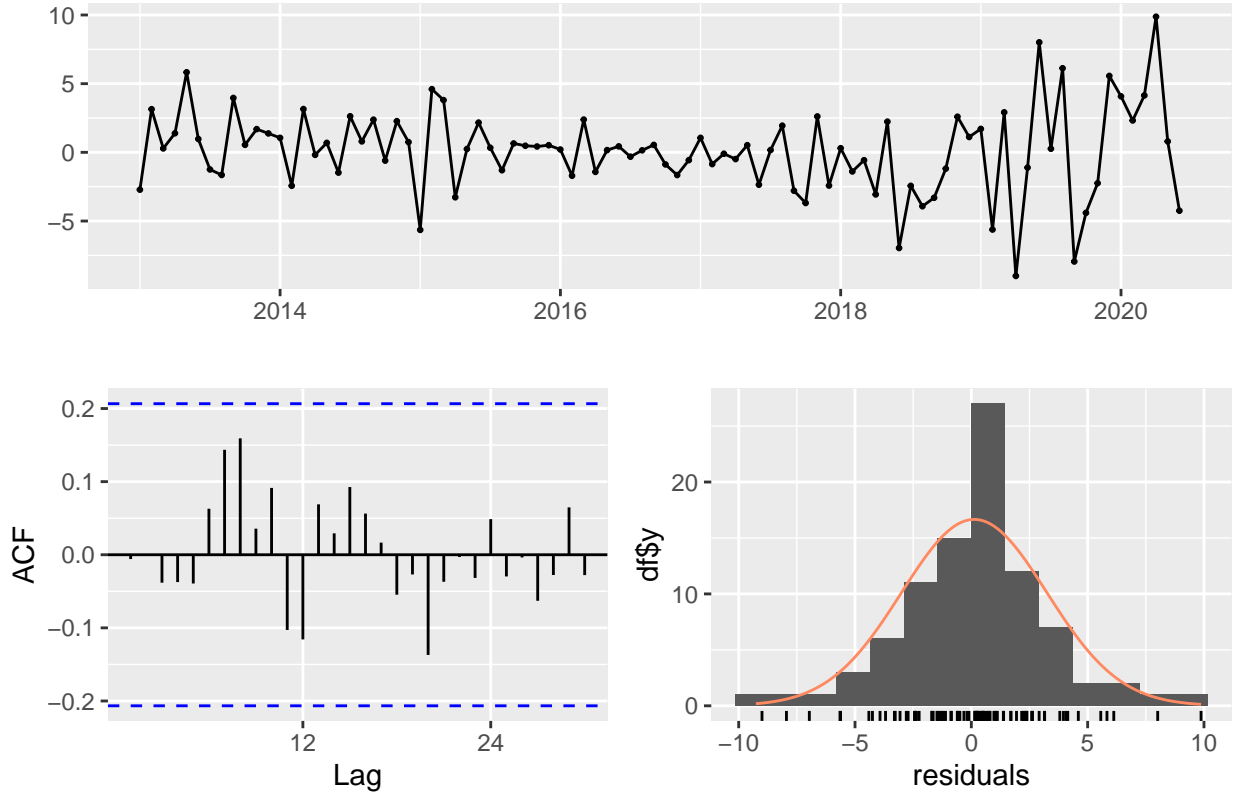
summary(fittrain2)

## Series: traindat[, "xau_try_gram"]
## Regression with ARIMA(2,0,3) errors
##
## Coefficients:
##          ar1      ar2      ma1      ma2      ma3  intercept          xe
##        -0.0694  0.9071  1.0167  0.4759  0.3866  -122.2576  40.3652
## s.e.    0.0708  0.0647  0.1287  0.1681  0.1446   15.4538   1.3693
##      xau_usd_ounce          t      tkare
##              0.0937  -0.5486  0.0128
## s.e.          0.0061   0.5454  0.0058
##
## sigma^2 = 10.91: log likelihood = -231.66
## AIC=485.31  AICc=488.7  BIC=512.81
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.1240805 3.114082 2.28416 0.1883555 1.518935 0.06416572
##              ACF1
## Training set -0.005900055

checkresiduals(fittrain2)

```

Residuals from Regression with ARIMA(2,0,3) errors



```
##  
## Ljung-Box test  
##  
## data: Residuals from Regression with ARIMA(2,0,3) errors  
## Q* = 11.241, df = 13, p-value = 0.5906  
##  
## Model df: 5. Total lags used: 18
```

```
# ***train seti üzerinden dinamik regresyon modeli kurduk***.  
# yukarıda test ve train diye ayırmadığımız için tüm veri seti üzerinde model kurmustuk  
# burada auto.arima yerine baska yontemler denenebilir
```

Regression with ARIMA(2,0,3) modeli belirlendi.

not xau vs arasındaki ilişki lineer yerine kuadratik olsaydı orada da bu şekilde t değişkeni kullanacaktık. ancak onlar lineer olduğu için kullanmadık. hem kendisi hem de karesi kuadratik model. **buna nereden karar veriyoruz?** Grafiklerden(25.satır). Orjinal değişkenlerle yanıt değişkenim arasındaki ilişkiyi autoplot çizdirip bakıyorum. İlişki eğer lineerse direkt yukarıdaki yaptığımız gibi olduğu gibi kullanıyoruz. Ama kuadratik gözüküyorsa açıklayıcı değişkenin hem kendisini hem de karesini kullanıyoruz.

Test seti için forecast yapmak istiyoruz. Öncelikle x'ler için yapacağız training setteki x'ler için

Şimdi her iki yöntemin test seti üzerindeki performansını incelemek için 8 aylık tahmin yapalım. bunun için önce açıklayıcı değişkenleri tahminletip yeni xreg matrisini oluşturmalıyız.

```

fitusd2 <- auto.arima(traindat[, "xau_usd_ounce"])
fitxe2<- auto.arima(traindat[, "xe"])

# 1.model icin forecast
fusd2<-forecast(fitusd2,8)
# xau_usd_ounce degiskeni icin auto.arima ile kurdugumuz modelin tahmin degerleri
fxe2<-forecast(fitxe2,8)
# xe degiskeni icin auto.arima ile kurdugumuz modelin tahmin degerleri
fcasttest1 <- forecast(fittrain1,xreg=cbind(xe=as.vector(fxe2$mean),
                                             xau_usd_ounce=as.vector(fusd2$mean)))

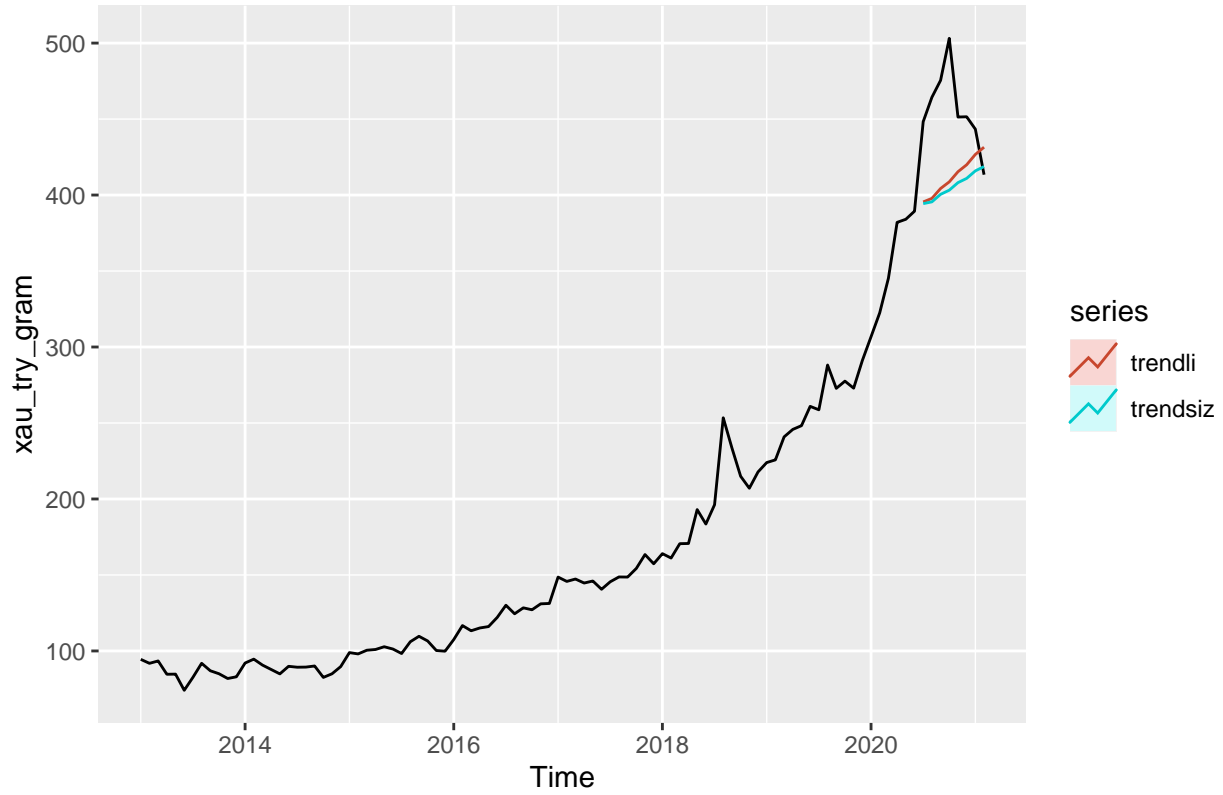
# test seti icin 8 adimlik tahminler

### hatirlatma ####
t<-ts(1:dim(traindat)[1],start=c(2013,1),frequency = 12)
tkare<-t^2
xreg <- cbind(xe = traindat[, "xe"],xau_usd_ounce = traindat[, "xau_usd_ounce"],t,tkare)
fittrain2 <- auto.arima(traindat[, "xau_try_gram"], xreg = xreg,
                       stepwise = FALSE, approximation = FALSE)
#####

# 2.model icin forecast (fittrain2 modeli)
t <- ts(dim(traindat)[1]+1:8,start=c(2020,7),frequency = 12)
# start olarak train setin baslangicini aliyoruz
# 8 adimlik tahmin olduđu icin 8 yazdik
# DİKKAT***** start icine test setinin baslangicini yaziyoruz
## dim(traindat)[1]+1 yerine length(traindat) yazilabilir
tkare <- t^2
fcasttest2 <- forecast(fittrain2,xreg =cbind(xe=as.vector(fxe2$mean),
                                             xau_usd_ounce=as.vector(fusd2$mean),t,tkare))
# fittrain2 trendli (t ve tkareli model 145.satirda)

autoplot(dat[, "xau_try_gram"]) + ylab("xau_try_gram")+
  autolayer(fcasttest2,PI=FALSE,series="trendli")+
  autolayer(fcasttest1,PI=FALSE,series = "trendsiz")

```



```
# trendli t ve tkare iceren
```

```
accuracy(fcasttest1,testdat[,3])
```

```
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set  0.3954041  3.258174  2.31872  0.1289404  1.510989  0.06513657
## Test set     50.4281759  58.670313  51.76972  10.7394310  11.063953  1.45429459
##              ACF1 Theil's U
## Training set -0.0216642      NA
## Test set     0.4106146  2.335618
```

```
accuracy(fcasttest2,testdat[,3])
```

```
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set  0.1240805  3.114082  2.28416  0.1883555  1.518935  0.06416572
## Test set     43.8769985  54.878194  48.40288  9.2727758  10.367596  1.35971459
##              ACF1 Theil's U
## Training set -0.005900055      NA
## Test set     0.451271680  2.174292
```

```
# train ve test seti icin degerleri hesaplattik
```

```
# train ve test seti için ME RMSE MAE MPE MAPE gibi degerleri veriyor
```

İki model kurduk bir modelde xe ve xau_usd_ounce var diğerinde ise bunlara ek t ve t^2 var

EK OLARAK: açıklayıcı değişkene dinamik regresyon uygulama

Dinamik regresyon modeli kullanarak açıklayıcı değişkenlerin tahminini yapabiliriz

```
# dinamik regresyon modeli ile yaptik tabii ki burada da auto.arima degisebilir
fitusd2 <- auto.arima(traindat[, "xau_usd_ounce"])
fitxe2<- auto.arima(traindat[, "xe"])
fusd2<-forecast(fitusd2,8)
fxe2<-forecast(fitxe2,8)
fcasttest1 <- forecast(fittrain1,xreg = cbind(xe=as.vector(fxe2$mean),
                                             xau_usd_ounce=as.vector(fusd2$mean)))

t<-ts(dim(traindat)[1]+1:8,start=c(2020,7),frequency = 12)
# DİKKAT***** start icine test setinin baslangicini yaziyoruz
tkare<-t^2
fcasttest2 <- forecast(fittrain2,
xreg = cbind(xe=as.vector(fxe2$mean),
              xau_usd_ounce=as.vector(fusd2$mean),t,tkare))

accuracy(fcasttest1, testdat[,3])
```

```
##                ME        RMSE        MAE        MPE        MAPE        MASE
## Training set  0.3954041  3.258174  2.31872  0.1289404  1.510989  0.06513657
## Test set     50.4281759  58.670313  51.76972  10.7394310  11.063953  1.45429459
##                ACF1 Theil's U
## Training set -0.0216642      NA
## Test set     0.4106146  2.335618
```

```
accuracy(fcasttest2, testdat[,3])
```

```
##                ME        RMSE        MAE        MPE        MAPE        MASE
## Training set  0.1240805  3.114082  2.28416  0.1883555  1.518935  0.06416572
## Test set     43.8769985  54.878194  48.40288  9.2727758  10.367596  1.35971459
##                ACF1 Theil's U
## Training set -0.005900055      NA
## Test set     0.451271680  2.174292
```

Burada da açıklayıcı değişkenleri tahmin etmek için direkt auto.arima kullanmıştık ancak oradaki açıklayıcı değişkenleri tahmin etmek için de dinamik regresyon modeli kullanabiliriz. t ve t^2 yi değişken olarak kullanarak.

“xau_usd_ounce” ve t ve $tkare$ ile kullanıldı. zamanla xe arasındaki ilişki lineer ancak zamanla xau_usd_ounce arasındaki ilişki kuadratikti. Dolayısıyla xe ile sadece zaman(t) değişkeni xau_usd_ounce içinse t ve $tkare$ beraber kullanıldı.