

# Zaman Serileri Analizi Emir Doğan

temel\_islemler.Rdm

2023-11-15

Muhammed Emir Doğan

Veriyi Kaggle'dan aldım. Bu veri 2000 ile 2022 yılları arasında Gümüş fiyatlarının günlük Açılış, En Yüksek, En Düşük, Kapanış fiyatlarını ve günlük hacmini Dolar (USD) cinsinden göstermektedir. Bu veri borsa verisi olduğundan hafta sonları yoktur ve bu nedenle verimizi aylık veriye çevirmemiz gerekmektedir.

```
library(lubridate)
library(fpp2)
library(dplyr)
library(readr)
silver <- read.csv("C:/Users/emird/OneDrive/Masaüstü/time_series_vize/silver.csv")

View(silver)
head(silver)
```

##	Date	Open	High	Low	Close	Volume	Currency
## 1	2000-01-04	5.420	5.420	5.32	5.375	27560	USD
## 2	2000-01-05	5.375	5.380	5.16	5.210	13515	USD
## 3	2000-01-06	5.205	5.215	5.15	5.167	4729	USD
## 4	2000-01-07	5.170	5.215	5.15	5.195	5375	USD
## 5	2000-01-10	5.190	5.230	5.17	5.190	4278	USD
## 6	2000-01-11	5.190	5.220	5.19	5.195	18507	USD

Zamana bağlı aylık kapanış fiyatlarını inceleyelim. Önce günlük zaman serisi oluşturup tarih kısmını yıl-ay-hafta olarak ayrıştırarak aylığa çeviriyoruz.

```
silver_ts <- ts(silver$Close, start = decimal_date(as.Date("2000-01-04")), frequency = 365)
head(silver_ts)
```

```
## Time Series:
## Start = 2000.00819672131
## End = 2000.02189535145
## Frequency = 365
## [1] 5.375 5.210 5.167 5.195 5.190 5.195
```

Verimizi tarih kısmını yıl-ay-hafta olarak ayrıştıralım.

```
silver$Year <- year(silver$Date)
silver$Month <- month(silver$Date)
silver$Week <- week(silver$Date)

as_tibble(silver)
```

```
## # A tibble: 5,708 x 10
##   Date      Open  High   Low Close Volume Currency Year Month Week
##   <chr>      <dbl> <dbl> <dbl> <dbl> <int> <chr>   <dbl> <dbl> <dbl>
## 1 2000-01-04  5.42  5.42  5.32  5.38  27560 USD    2000     1     1
## 2 2000-01-05  5.38  5.38  5.16  5.21  13515 USD    2000     1     1
## 3 2000-01-06  5.20  5.22  5.15  5.17   4729 USD    2000     1     1
## 4 2000-01-07  5.17  5.22  5.15  5.20   5375 USD    2000     1     1
## 5 2000-01-10  5.19  5.23  5.17  5.19   4278 USD    2000     1     2
## 6 2000-01-11  5.19  5.22  5.19  5.20  18507 USD    2000     1     2
## 7 2000-01-12  5.20  5.22  5.12  5.14  10288 USD    2000     1     2
## 8 2000-01-13  5.14  5.18  5.12  5.15   5478 USD    2000     1     2
## 9 2000-01-14  5.16  5.18  5.13  5.15   9314 USD    2000     1     2
## 10 2000-01-18 5.16  5.19  5.12  5.16   6171 USD    2000     1     3
## # i 5,698 more rows
```

```
View(silver)
```

Artık verimizi aylık veri haline getirebiliriz.

```
silver%>%group_by(Year,Month)%>%summarise(mean_close = mean(Close)) -> aylık_veri
```

```
## 'summarise()' has grouped output by 'Year'. You can override using the
## '.groups' argument.
```

```
silver%>%group_by()
```

```
## # A tibble: 5,708 x 10
##   Date      Open  High   Low Close Volume Currency Year Month Week
##   <chr>      <dbl> <dbl> <dbl> <dbl> <int> <chr>   <dbl> <dbl> <dbl>
## 1 2000-01-04  5.42  5.42  5.32  5.38  27560 USD    2000     1     1
## 2 2000-01-05  5.38  5.38  5.16  5.21  13515 USD    2000     1     1
## 3 2000-01-06  5.20  5.22  5.15  5.17   4729 USD    2000     1     1
```

```
## 4 2000-01-07 5.17 5.22 5.15 5.20 5375 USD 2000 1 1
## 5 2000-01-10 5.19 5.23 5.17 5.19 4278 USD 2000 1 2
## 6 2000-01-11 5.19 5.22 5.19 5.20 18507 USD 2000 1 2
## 7 2000-01-12 5.20 5.22 5.12 5.14 10288 USD 2000 1 2
## 8 2000-01-13 5.14 5.18 5.12 5.15 5478 USD 2000 1 2
## 9 2000-01-14 5.16 5.18 5.13 5.15 9314 USD 2000 1 2
## 10 2000-01-18 5.16 5.19 5.12 5.16 6171 USD 2000 1 3
## # i 5,698 more rows
```

```
head(aylık_veri)
```

```
## # A tibble: 6 x 3
## # Groups:   Year [1]
##   Year Month mean_close
##   <dbl> <dbl>     <dbl>
## 1  2000     1     5.24
## 2  2000     2     5.28
## 3  2000     3     5.11
## 4  2000     4     5.09
## 5  2000     5     5.05
## 6  2000     6     5.04
```

**Soru 1 : Zaman serisi grafiği çizdiriniz.**

```
aylık_silver_ts <- ts(aylık_veri$mean_close, start = c(2000,1), frequency = 12)
autoplot(aylık_silver_ts)+
  xlab("Date")+
  ylab("Silver Price")+
  ggtitle("Monthly Silver Price")
```



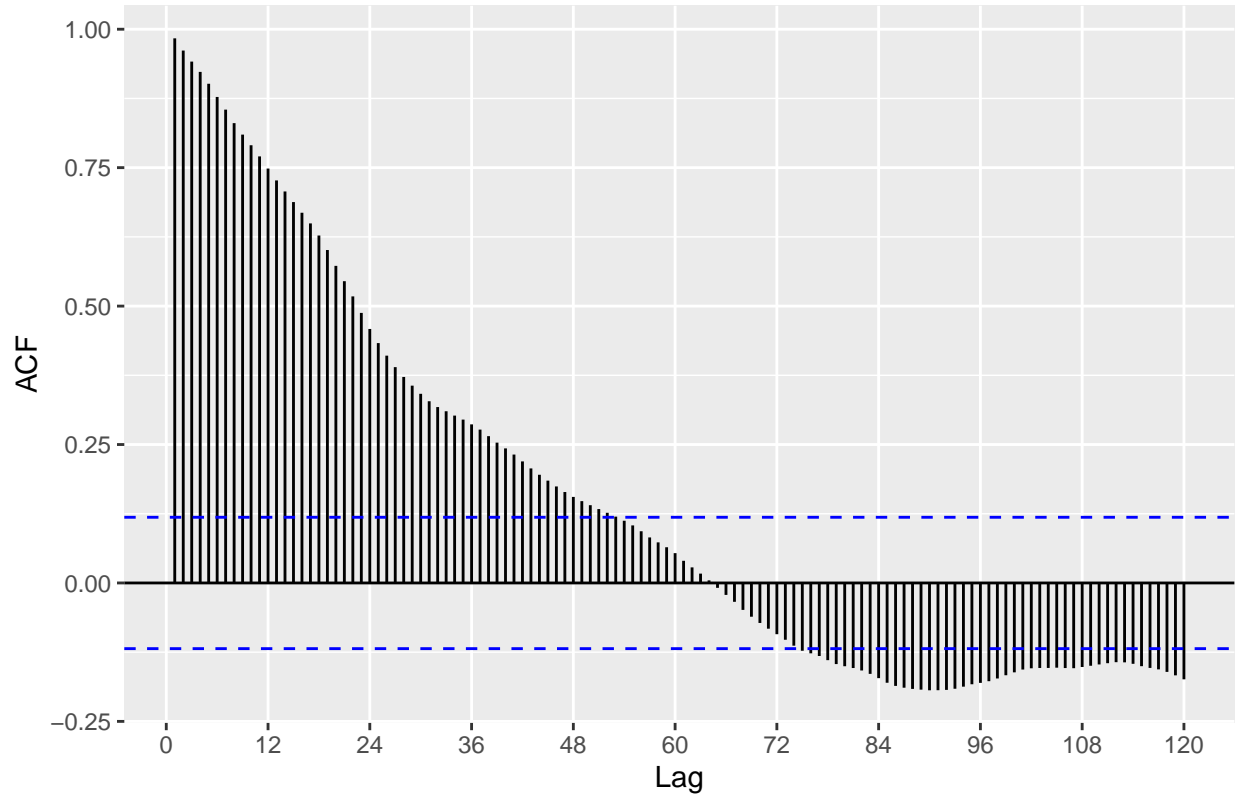
**Soru 2 :** Verinizde otokorelasyon olup olmadığını ACF grafiği ve Ljung Box test kullanarak inceleyip yorumlayınız.

Grafikte 120 gecikme yani 120 önceki değere kadar otokorelasyon hesaplamasını söyledik. Son değerler ilk değerlere göre çok büyük olduğu için böyle bir grafik ortaya çıkmıştır. Yani burada bir mevsimsellikten söz edemeyiz ancak grafiğe bakarak bir trend olduğunu görmek mümkündür.

Ljung Box testi kullandığımızda ise  $p\text{-value} < 0.05$ . Buradaki  $H_0$ :Otokorelasyon yoktur hipotezi  $p\text{-value} < 0.05$  olduğundan reddedilir. Yani otokorelasyon vardır ancak bir mevsimsellikten söz edemeyiz.

```
ggAcf(aylık_silver_ts, lag.max = 120) +  
ggtitle("Otokorelasyon Fonksiyonu Grafiği")
```

Otokorelasyon Fonksiyonu Grafigi



```
length(aylık_silver_ts)/5
```

```
## [1] 54.6
```

```
Box.test(aylık_silver_ts, lag = 10, type = "Lj")
```

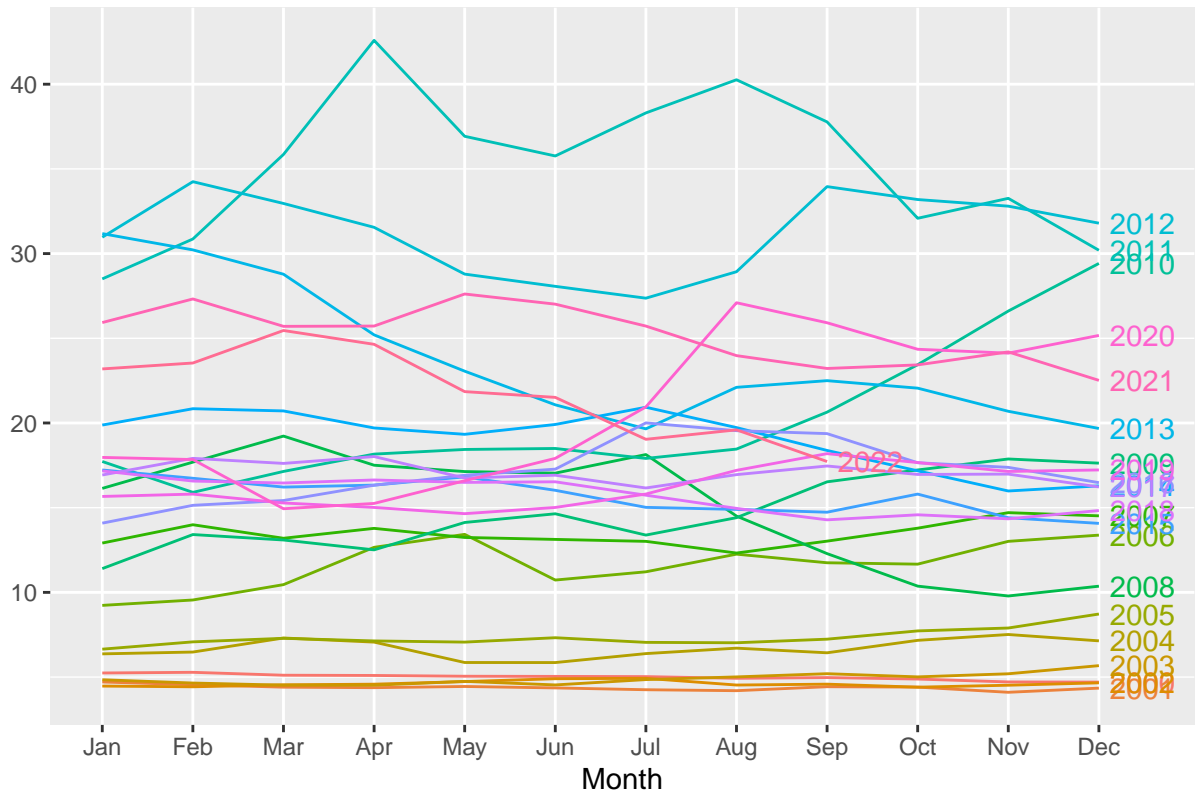
```
##  
## Box-Ljung test  
##  
## data:  aylık_silver_ts  
## X-squared = 2218.1, df = 10, p-value < 2.2e-16
```

**Soru 3 :** Verinizde mevsimsellik olup olmadığını mevsimsellik grafikleri kullanarak inceleyiniz.

Mevsimsellik grafiklerine baktığımızda her sezon farklı bir grafik olduğundan mevsimsellik yoktur denilebilir.

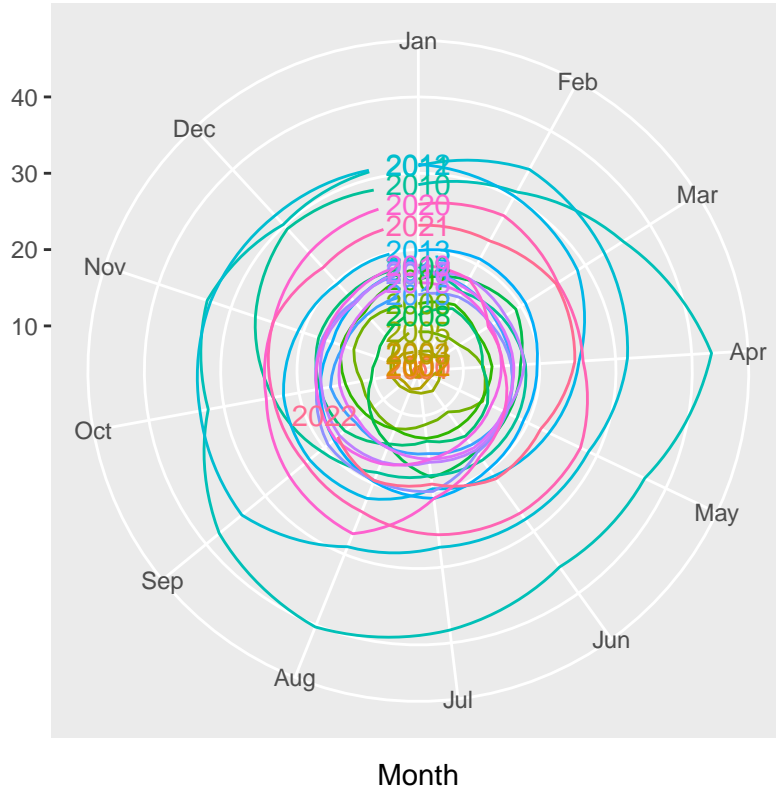
```
ggseasonplot(aylık_silver_ts, year.labels = TRUE)
```

Seasonal plot: aylık\_silver\_ts



```
ggseasonplot(aylık_silver_ts, year.labels = TRUE, polar = TRUE)
```

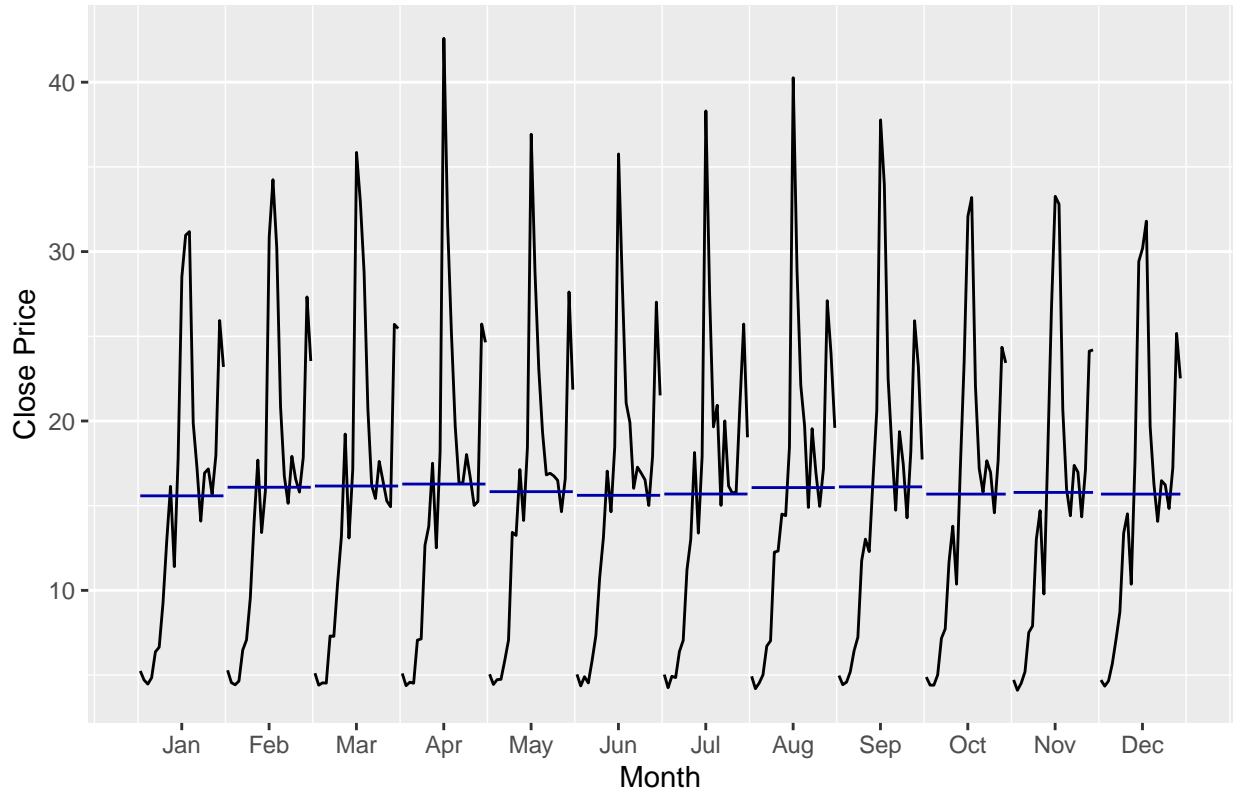
Seasonal plot: aylık\_silver\_ts



Yatay çizgiler her ayın ortalamasını gösterir ancak bu grafiğe de bakarsak aylık ortalamalar sabit gözükmemektedir ve aylık ortalamalarda bir mevsimsellik yoktur.

```
ggsubseriesplot(x = aylık_silver_ts) +  
ylab("Close Price") +  
ggtitle("Aylık Gümüş Fiyatları Mevsimsel Alt Seri Grafiği")
```

Aylık Gümüş Fiyatları Mevsimsel Alt Seri Grafiği



**Soru 4 :** Verinizi son 20 gözlemi test ve öncesini train olarak ayırıştırın.

```
set.seed(123)
train_ts <- window(aylık_silver_ts, end = c(2021, 01))
test_ts <- window(aylık_silver_ts, start = c(2021, 02))
# length(test_ts)
```

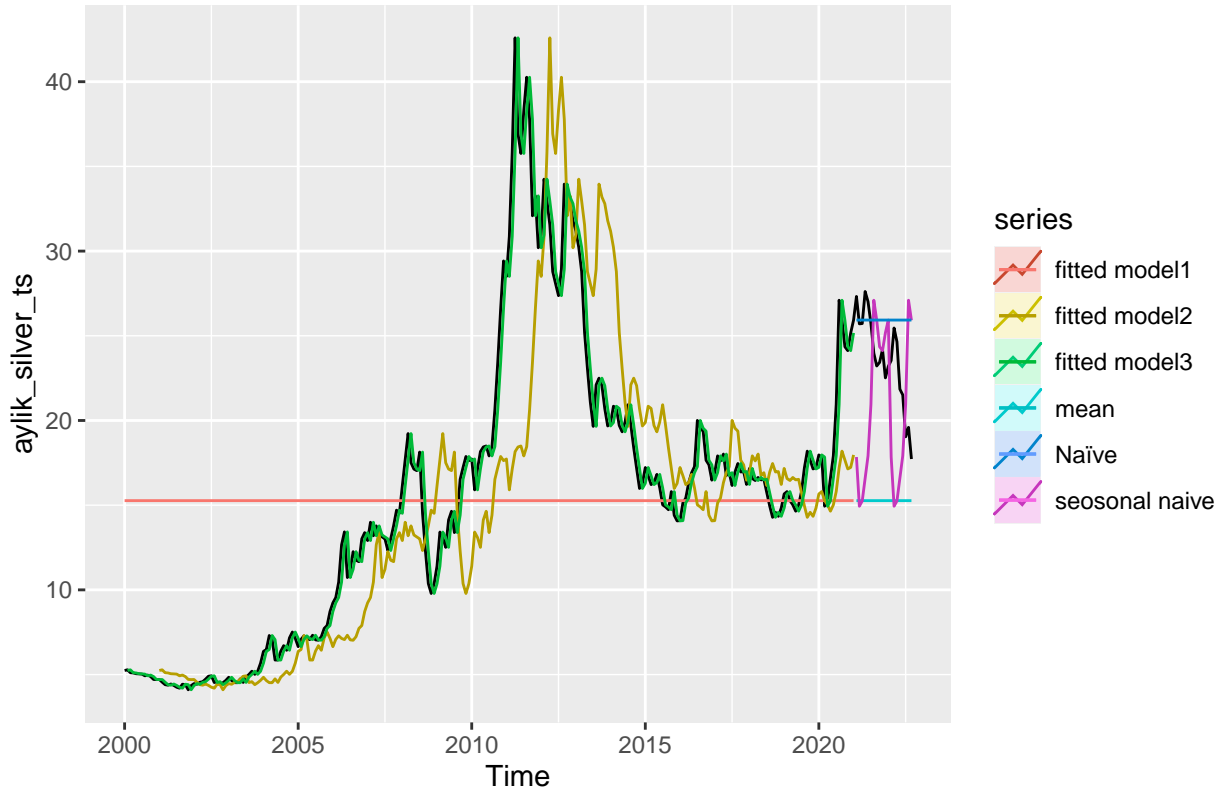
```
set.seed(123)
model1 <- meanf(train_ts, 20)
model2 <- snaive(train_ts, 20)
model3 <- rwf(train_ts, 20)
```

**Soru 5 :** Bu kurduğumuz modellerin test set ve training set tahmin performanslarını grafik üzerinden gösteriniz.

```
autoplot(aylık_silver_ts)+autolayer(model1, series = "mean", PI = FALSE)+
  autolayer(model2, series = "seasonal naive", PI = FALSE)+autolayer(model3, series = "Naive ", PI = FALSE)
```



```
autolayer(fitted(model1), series = "fitted model1")+autolayer(fitted(model2), series = "fitted model2")
autolayer(fitted(model3), series = "fitted model3")
```



**Soru 6 : Kurduğumuz modellerin test seti ve training set üzerindeki RMSE değerlerini bulunuz.**

Model1 için Training set RMSE'si “8.456276” iken Test set RMSE'si “8.790334” 'dir.

Model2 için Training set RMSE'si “5.434236” iken Test set RMSE'si “6.979750” 'dir.

Model3 için Training set RMSE'si “1.388501” iken Test set RMSE'si “3.495002” 'dir.

```
set.seed(123)
accuracy(model1,test_ts)
```

```
##               ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -3.519843e-16 8.456276 6.428988 -45.49923 69.79622 1.779261
## Test set      8.381554e+00 8.790334 8.381554 34.54819 34.54819 2.319645
##               ACF1 Theil's U
## Training set 0.9806533      NA
## Test set      0.6914063 5.777789
```

```
accuracy(model2,test_ts)
```

```
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 0.8134038 5.434236 3.613292 3.683342 20.13761 1.000000 0.9314089
## Test set      3.0219905 6.979750 5.995692 10.378609 25.04883 1.659343 0.7197811
##               Theil's U
## Training set      NA
## Test set      4.69796
```

```
accuracy(model3,test_ts)
```

```
##               ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.08209336 1.388501 0.8794047 0.3971108 5.185827 0.2433805
## Test set      -2.27936759 3.495002 2.6959295 -11.1542600 12.676880 0.7461145
##               ACF1 Theil's U
## Training set 0.2157025      NA
## Test set      0.6914063 2.829055
```

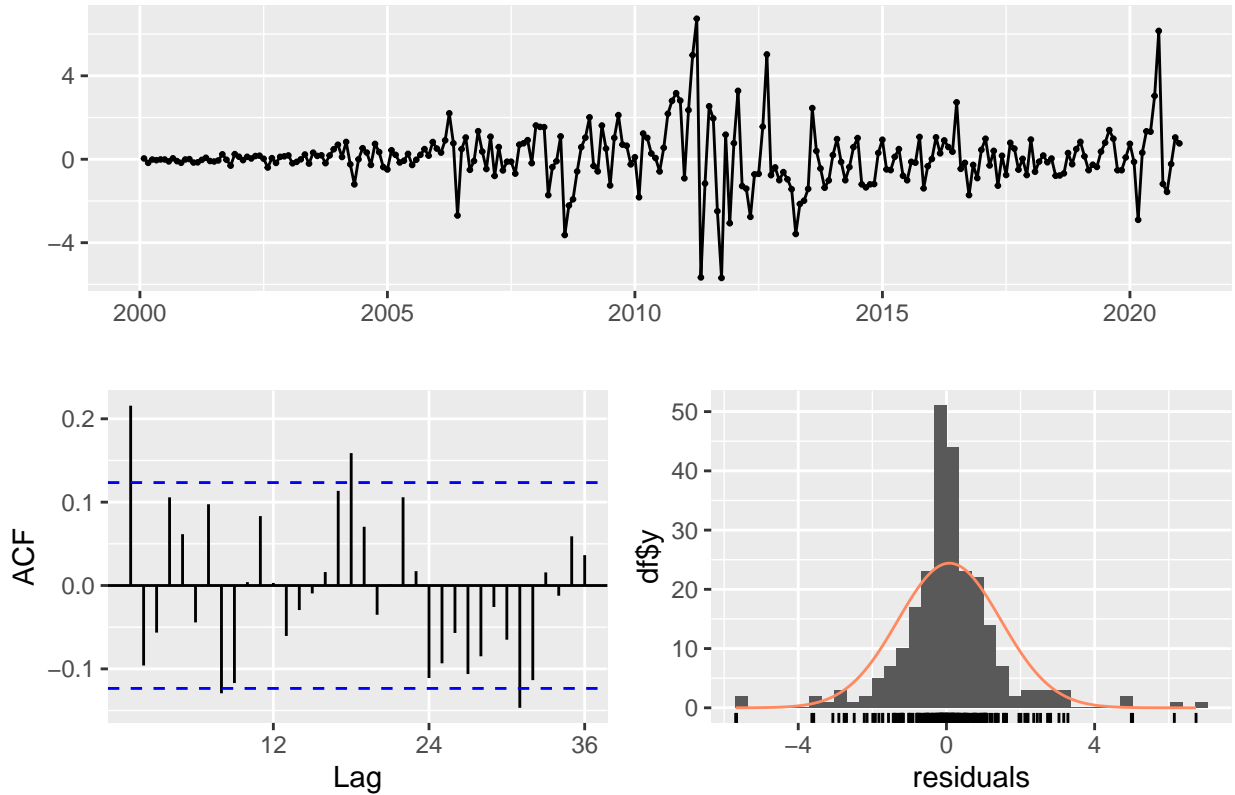
**Soru 7 : Kurduğunuz 3 modelden test seti forecasting performansı daha iyi olan için model varsayımlarını kontrol edip yorumlayınız.**

Forecasting performansı en iyi olan model “model3” olduğundan “model3” için varsayımları kontrol edelim.

Burada  $H_0$ :Otokorelasyon yoktur hipotezidir ve  $p\text{-value} = 0.0008231$  yani 0’a çok yakın olduğundan  $H_0$ ’ı reddettik. Yani otokorelasyon vardır denilebilir.

```
checkresiduals(model3)
```

Residuals from Random walk



```
##  
##  Ljung-Box test  
##  
## data:  Residuals from Random walk  
## Q* = 51.831, df = 24, p-value = 0.0008231  
##  
## Model df: 0.   Total lags used: 24
```

Soru 8 : Test seti forecasting performansı daha iyi olan modele uygun lambda değerini belirleyip box-cox dönüşümünü yapalım. Bu kurduğumuz modelin test set RMSE değeri nedir ?

Kurduğumuz modeller arasında forecasting performansı (mape ve rmse değerlerine baktığımızda) en iyi naive model olarak gözük-mektedir. Naive modele Box-Cox dönüşümü uygulayarak kurduğumuz modelin test set RMSE değeri “5.592709” olarak bulunmuştur.

```
set.seed(123)  
lamda<-BoxCox.lambda(train_ts)
```

```
model3<-rwf(train_ts,lambda = lamda, h = 50, biasadj = TRUE)
accuracy(model3, aylık_silver_ts)
```

```
##           ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set      NaN      NaN      NaN      NaN      NaN      NaN      NA
## Test set      -4.259812  5.592709  4.48776 -20.12386  20.95515  1.242014  0.7491774
##           Theil's U
## Training set      NA
## Test set      4.486449
```