ILERI REGRESYON ANALIZI FINAL ODEVI

IremKoyunlu121517056

16 06 2020

NATIONAL STOCK EXCHANGE

KAYNAK: https://www.kaggle.com/atulanandjha/national-stock-exchange-time-series

```
NSTS=read.csv("C:/Users/CASPER/Desktop/infy_stock.csv", header=T)
head(NSTS,10)
```

```
##          Date Symbol Series Prev.Close    Open    High     Low    Last   Close
## 1  2015-01-01   INFY     EQ    1972.55 1968.95 1982.00 1956.90 1971.00 1974.40
## 2  2015-01-02   INFY     EQ    1974.40 1972.00 2019.05 1972.00 2017.95 2013.20
## 3  2015-01-05   INFY     EQ    2013.20 2009.90 2030.00 1977.50 1996.00 1995.90
## 4  2015-01-06   INFY     EQ    1995.90 1980.00 1985.00 1934.10 1965.10 1954.20
## 5  2015-01-07   INFY     EQ    1954.20 1965.00 1974.75 1950.00 1966.05 1963.55
## 6  2015-01-08   INFY     EQ    1963.55 1985.60 1997.00 1950.00 1979.25 1973.45
## 7  2015-01-09   INFY     EQ    1973.45 1980.10 2109.00 1913.05 2075.30 2074.45
## 8  2015-01-12   INFY     EQ    2074.45 2092.00 2119.20 2075.00 2112.95 2115.95
## 9  2015-01-13   INFY     EQ    2115.95 2107.80 2107.80 2075.00 2092.00 2088.90
## 10 2015-01-14   INFY     EQ    2088.90 2098.50 2133.00 2092.60 2129.00 2128.65
##       VWAP   Volume     Turnover Trades Deliverable.Volume X.Deliverble
## 1  1971.34   500691 9.870306e+13  14908             258080       0.5154
## 2  2003.25  1694580 3.394669e+14  54166            1249104       0.7371
## 3  2004.59  2484256 4.979911e+14  82694            1830962       0.7370
## 4  1954.82  2416829 4.724458e+14 108209            1772070       0.7332
## 5  1962.59  1812479 3.557162e+14  62463            1317720       0.7270
## 6  1972.78  3391230 6.690160e+14  92752            2686012       0.7920
## 7  2037.69 11215832 2.285439e+15 359214            3369489       0.3004
## 8  2099.40  3189722 6.696516e+14 107209            1818800       0.5702
## 9  2089.42  2200309 4.597374e+14  66676            1385009       0.6295
## 10 2110.88  2480315 5.235638e+14  53263            1832958       0.7390
```

VERI SETI ACIKLAMASI ;

Ulusal Borsa:Hint bilisim sirketlerinin ulusal borsa veri setidir. Hindistan Ulusal Borsasi (NSE) Mumbai, Maharashtra, Hindistan'da bulunan bir Hint borsasidir. Ulusal Menkul Kiymetler Borsasi (NSE) 1992 yilinda yetkisiz bir elektronik borsa olarak kuruldu. Hindistan Hukumeti'nin talebi uzerine onde gelen finans kuruluslari tarafindan tesvik edildi. Hindistan'in ciro ile yaptigi en buyuk borsa. 1994 yilinda elektronik ekran tabanli ticareti baslatti. Daha sonra, 2000 yilinda ulkede turunun ilk ornegi olan endeks futures ve internet ticareti baslatti. 248 gozlem 15 sutun bulunmakta.

Degiskenler:

Date: Verilerin kaydedildigi tarih.

Symbol: Stokun NSE (Hindistan Ulusal Borsasi) sembolu.

Series: Bu hisse senedinin serisi. (EQ,BE,BL,BT,GC,IL)

Prev Close: Son gun kapanis noktasi.

Open: Mevcut gun acilis noktasi.

High: Mevcut gunun en yuksek noktasi.

Low: Mevcut gunun en dusuk noktasi.

Last: Son islem gununde belirli bir hisse senedi veya borsa endeksi icin son teklif edilen islem fiyati.

Close: Gecerli gun icin kapanis noktasi.

VWAP: Hacim agirlikli ortalama fiyat anlamina gelir, bir varligin belirli bir zaman araligi icin hacme gore agirligi alinmis ortalama fiyatidir.

Volume: Belirli bir zaman diliminde islem goren menkul kiymet tutari. Her alici icin bir satici var ve her islem toplam hacim sayisina katkida bulunuyor.

Turnover: Hisse senedinin o gune kadar toplam cirosu.

Trades: Hisse senedi alim veya satim sayisi.

Deliverable: Bir grup insandan (bugunden once demat hesabida bu hisseleri olan ve bugun satis yapan) baska bir grup insana (bu hisseleri satin almis olan ve bu hisseleri T+2 gunlerine kadar alacak olan) hareket eden hisse miktari. Demat hesabi(Hindistan Internet Borsasi)).

%Deliverble: Bu hisse senedinin teslimat yuzdesi.

VWAP'yi ozellikle guclu bir gosterge haline getiren ortalama fiyat hesaplamasinda hacmi kullanma seklidir. VWAP, hacmin gucuyle fiyat hareketini birlestirerek pratik ve kullanimi kolay bir gosterge yaratir. Alim satim yapanlar VWAP'yi bir trend onaylama ya da giris ve cikis noktalarini belirleme araci olarak kullanabilir. VWAP her gunun basinda sifirlanir. Alim satim yapmak istedigimizde, VWAP'in altinda almak ve ustunde satmak karlidir. Fiyatin bugunku degerinin ustunde mi alim yaptik yoksa altinda mi alim yaptik bunu belirlememizi saglar. Fiyat VWAP uzerinde ise, satmak icin iyi bir gun ici fiyattir. Fiyat VWAP'in altindaysa, satin almak icin iyi bir gun ici fiyatidir.

Degiskenleri Aciklamada Kullanilan Ek Bilgi Kaynaklari:

https://www.investopedia.com/articles/trading/11/trading-with-vwap-mvwap.asp https://www.binance.vision/tr/economics/volume-weighted-average-price-vwap-explained

```
summary(NSTS)
```

```
##      Date              Symbol             Series            Prev.Close
##  Length:248         Length:248         Length:248         Min.   : 937.5
##  Class :character   Class :character   Class :character   1st Qu.:1085.9
##  Mode  :character   Mode  :character   Mode  :character   Median :1149.7
##                                                           Mean   :1551.5
##                                                           3rd Qu.:2125.3
##                                                           Max.   :2324.7
##       Open            High             Low              Last
##  Min.   : 941   Min.   : 952.1   Min.   : 932.6   Min.   : 935.5
##  1st Qu.:1088   1st Qu.:1100.0   1st Qu.:1067.2   1st Qu.:1086.9
##  Median :1150   Median :1159.7   Median :1131.2   Median :1145.6
##  Mean   :1551   Mean   :1566.3   Mean   :1530.1   Mean   :1548.1
##  3rd Qu.:2136   3rd Qu.:2150.0   3rd Qu.:2104.5   3rd Qu.:2125.2
##  Max.   :2328   Max.   :2336.0   Max.   :2292.1   Max.   :2323.2
##      Close             VWAP            Volume            Turnover
##  Min.   : 937.5   Min.   : 941.2   Min.   :  353652   Min.   :3.923e+13
##  1st Qu.:1085.9   1st Qu.:1085.9   1st Qu.: 1722753   1st Qu.:2.847e+14
##  Median :1149.3   Median :1146.2   Median : 2532474   Median :3.625e+14
##  Mean   :1548.0   Mean   :1548.1   Mean   : 2982072   Mean   :4.234e+14
##  3rd Qu.:2125.3   3rd Qu.:2125.1   3rd Qu.: 3567063   3rd Qu.:4.915e+14
##  Max.   :2324.7   Max.   :2322.2   Max.   :19155056   Max.   :2.285e+15
##      Trades       Deliverable.Volume  X.Deliverble
##  Min.   : 13196   Min.   : 166222    Min.   :0.3004
##  1st Qu.: 63052   1st Qu.:1139407    1st Qu.:0.6161
##  Median : 80019   Median :1717132    Median :0.6763
##  Mean   : 92675   Mean   :1940081    Mean   :0.6623
##  3rd Qu.:106617   3rd Qu.:2467728    3rd Qu.:0.7235
##  Max.   :408583   Max.   :9575992    Max.   :0.8532
```

Verimizde duzeltilcek degisken yoktur. Verimizi oncelikle test ve train olarak ayiralim.

```
set.seed(2)
index<-sample(1:nrow(NSTS),round(nrow(NSTS)*0.85))
veritrain<-NSTS[index,]
veritest<-NSTS[-index,]
```

Regresyon modelimizi kuralim;

VWAP, hacmin gucuyle fiyat hareketini birlestirerek pratik ve kullanimi kolay bir gosterge yaratir. Bagimli degiskenimiz VWAP tir.(Yanit degiskeni) Aciklayici degisken olarak da Volume,High,Low,Close,Last kullanarak regresyon modeli kuralim.

```
lmod<-lm(VWAP~Volume+
           High+
          Low+
           Close+
           Last,data=veritrain)
summary(lmod)
```

```
##
## Call:
## lm(formula = VWAP ~ Volume + High + Low + Close + Last, data = veritrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.6284  -1.6843  -0.1007   1.9955  22.2949
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.925e+00  1.404e+00   2.083   0.0385 *
## Volume      -2.945e-07  2.165e-07  -1.360   0.1752
## High         4.031e-01  2.375e-02  16.976  < 2e-16 ***
## Low          2.394e-01  2.472e-02   9.686  < 2e-16 ***
## Close        5.541e-01  8.811e-02   6.289 1.89e-09 ***
## Last        -1.998e-01  8.899e-02  -2.245   0.0258 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.431 on 205 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 5.956e+05 on 5 and 205 DF,  p-value: < 2.2e-16
```

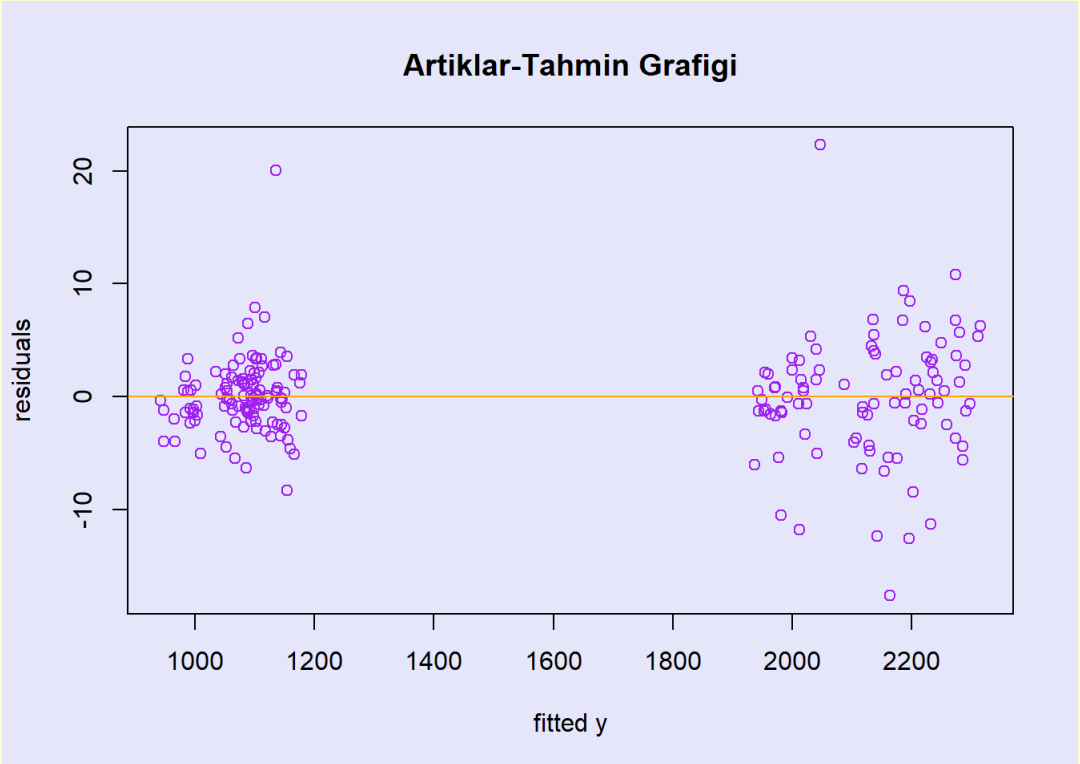
Kurulan regresyon modelinin anlamliligina baktigimizda p value=yaklasik 0 < 0.05 oldugundan kurulan model anlamlidir.

HATA ILE ILGILI VARSAYIMLAR

SABIT VARYANS

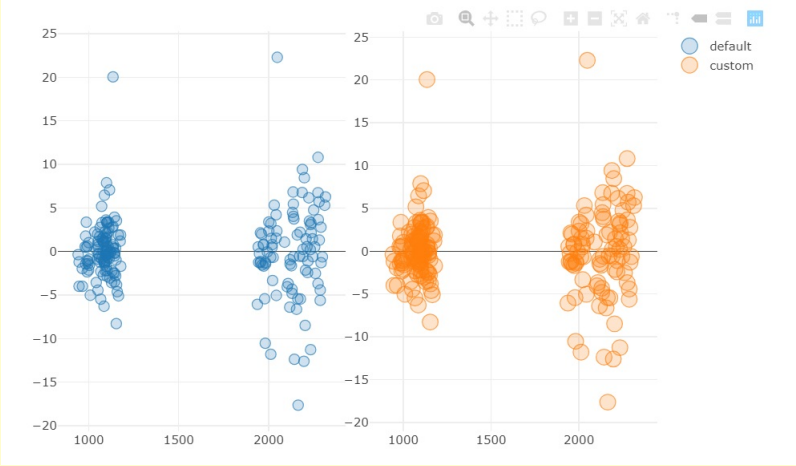
Sabit varyansliligin en kullanisli teshis yontemi artiklara (residuals(lmod)) karsilik tahmin (fitted(lmod)) degerlerinin plotlanmasidir.

```
op = par(bg = "lavender")
plot(fitted(lmod),residuals(lmod), xlab = "fitted y" ,ylab = "residuals",col="purple",main="Artiklar-Tahmin
Grafigi")
abline(h=0,col="orange")

# Interaktif bir sekilde gorsellestirelim;

library(plotly)
```



```
p <- plot_ly(veritrain, x = fitted(lmod), y = residuals(lmod), alpha = 0.3)
subplot(
  add_markers(p, size = 2, name = "default"),
  add_markers(p, size = 2, sizes = c(1, 248), name = "custom")
)
```

```
#Interaktif plotun gorseli;(Html ciktisini pdf e cevirince gozukmedigi icin ekledik)

knitr::include_graphics("sabitvaryans.png")
```



Cizdirdigimiz grafikte sifir etrafinda nasil dagildigini gormek icin h=0 ile yataya cizgi ekledik. Grafigimiz bize duzgun bir sekil vermedigi icin sabit varyansli mi diye emin olamiyoruz. Guvenilir olmasi icin degisken varyanslilik testlerine de bakmaliyiz.

DEGISKEN VARYANSLILIK TESTLERI

BREUSCH-PAGAN TESTI

H0:Heterosce Dosticity (Degisken Varyanslilik) problemi yok. H1:Heterosce Dosticity (Degisken Varyanslilik) problemi vardir.

```
#install.packages("lmtest")
library(lmtest)
bptest(lmod,data=veritrain)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lmod
## BP = 96.298, df = 5, p-value < 2.2e-16
```

Breusch pagan testimizin sonucuna gore p-value=2.2e-16 yaklasik 0 <0.05 oldugundan H0 hipotezi reddedilir yani heterocedosticity (degisken varyanslilik) problemi vardir deriz.

WHITE TEST

```
wmod<-lm(residuals(lmod)^2~fitted(lmod)+fitted(lmod)^2,veritrain)
summary(wmod)
```

```
##
## Call:
## lm(formula = residuals(lmod)^2 ~ fitted(lmod) + fitted(lmod)^2,
##     data = veritrain)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -33.51 -17.66  -8.62  -1.90 468.09
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -11.220010  11.260148  -0.996  0.32019
## fitted(lmod)   0.019624   0.006903   2.843  0.00492 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.79 on 209 degrees of freedom
## Multiple R-squared:  0.03723,    Adjusted R-squared:  0.03262
## F-statistic: 8.082 on 1 and 209 DF,  p-value: 0.004915
```

White testimizin sonucuna gore p-value=0.004915 <0.05 oldugundan H0 hipotezi reddedilir yani heterocedosticity (degisken varyanslilik) problemi vardir.

Varyanlarin homojenligi saglanmamasi durumunda “Yanit degiskeni uzerinde donusum yapmak” veya “Agirlikli En Kucuk Kareler” yontemlerine basvurmaliyiz.

AGIRLIKLI EN KUCUK KARELER YONTEMI

Siradan en kucuk kareler yontemi, hata varyanslarinin sabit oldugunu varsayar(homoscedasticity). Agirlikli en kucuk kareler yontemi bu varsayim saglanmadigi durumlarda kullanilir.

Varyansi buyuk olan degiskenin model uzerinde etkisi fazla olur. EKK nin en iyi calisabilmesi icin hata varyansi σ2i i=1,2,….n birbirine esit olmasi gerekir. Eger σ2i ler esit degil ise agirlikli en kucuk kareler yontemine gecmeliyiz.

Her bir σ2i varyansina karsilik wi= 1/σ2i agirligini tanimlariz.(Agirliklari 1/σ2i almamizin sebebi hepsinin varsayansini 1 e ve birbirlerine esitlemeye calismamizdir. σ2i*(1/σ2i))

Bu sekilde agirligi buyuk olanin agirligini alip, agirligi kucuk olana agirlik yukluyoruz. Buradaki zorluk σ2i parametresinin bilinmemesinden kaynaklanir ve w matrisi kolay belirlenemez.

Agirliklari belirlemek icin ilk olarak EKK regresyon modeli kurulur artiklar elde edilir. Agirliklar belirlendikten sonra agirliklandirilmis EKK kullanilir. Regresyon modeli uzerinden artiklar hesaplanip agirlik incelemesi yapilir. Gercekten kullanilan agirlikla varyans homojen hale gelmis mi diye bakilir. Eger varyans homojenligi saglanmamissa tekrar agirliklandirma yapilir buna iteratif agirliklandirma denir.

Bazi olasi varyans ve standart sapma fonksiyonu tahminleri:

1. Aciklayici degiskenlere karsilik artiklarin grafigini cizdirip megafon sekli varmi diye bakilir. Eger megafon sekli var ise aciklayici degiskenler ile mutlak artiklarin arasinda regresyon modeli kurulur. Bu regresyon modeli uzerinden tahmin degeri elde edilir. Bu elde elilen tahmin degerlerini σi yerine kullaniriz. Agirliklar da 1/σ2i diye olusturulur.
2. Ilk basta kurulan EKK modelindeki tahmin edilen yanitlara (y sapkalara) karsilik ei lerin grafigi megafon seklinde ise artiklarin mutlak degerine karsilik y sapkalarin regresyon modeli kurulur. Bu kurulan modelden elde edilen tahmin degerlerini σi yerine kullaniriz. Agirliklari da wi= 1/σ2i seklinde olustururuz.
3. Aciklayici degiskene karsi ei kare grafigi artan seklindeyse ei karelere karsilik o aciklayici degiskenin regresyon modeli kurulur. Bu modelin tahminlerini σ2i yerine kullaniriz. Agirliklari da wi= 1/σ2i seklinde olustururuz.
4. Tahmin edilen yanitlara (y sapka) karsi ei kare grafigi artan seklindeyse ei karelere karsilik tahmin edilen yanitlara regresyon modeli

kurulur. Bu modelden elde edilen tahminler σ2i tahminleri olarak kullanilir. Agirliklari da wi= 1/σ2i seklinde olustururuz.

Bunlardan hangisi daha uygun gorulurse agirlik o sekilde belirlenmelidir. Simdi verimiz uzerinde bu anlatilanlari yapmaya baslayalim.

```
library(ggplot2)
veritrain$artik<-(residuals(lmod)) #EKK modelinden elde edilen artiklar (residuals)
veritrain$tahmin<-predict(lmod) #EKK modelinden elde edilen tahminler (prediction)
head(veritrain)
```

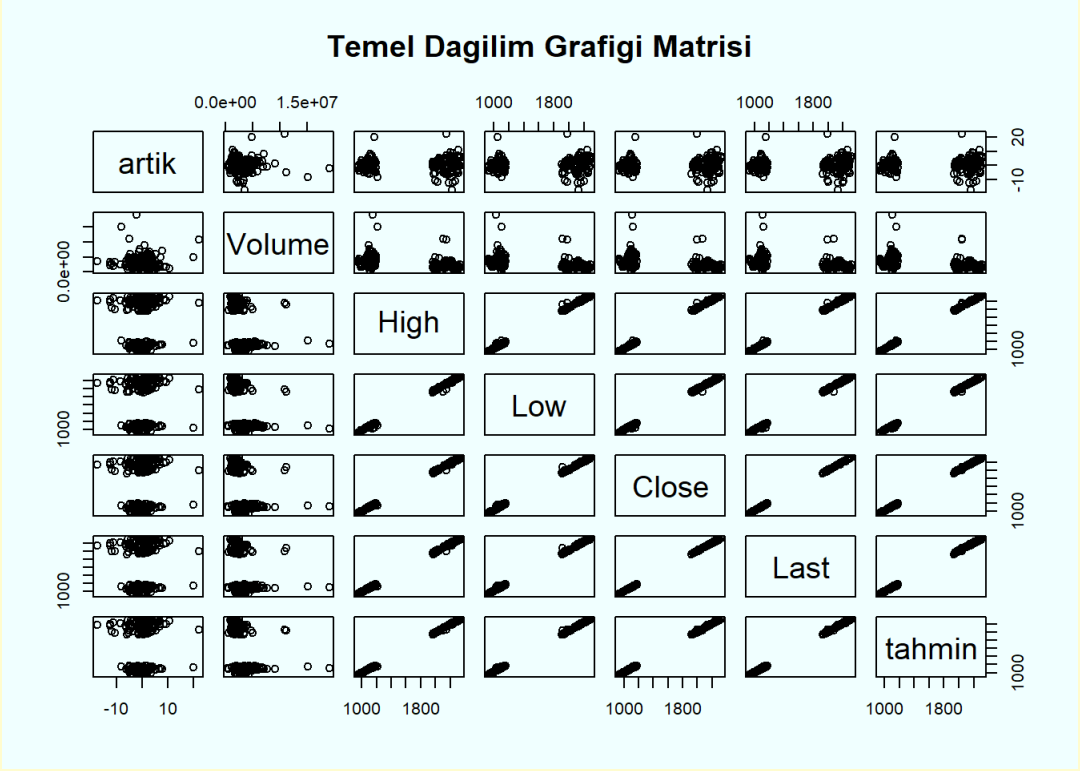
```
##           Date Symbol Series Prev.Close   Open   High    Low    Last   Close
## 85  2015-05-07   INFY     EQ    1922.05 1922.0 1967.1 1910.0 1940.00 1944.00
## 207 2015-10-30   INFY     EQ    1145.05 1145.0 1145.0 1130.0 1132.00 1135.45
## 198 2015-10-16   INFY     EQ    1097.35 1099.0 1101.4 1090.0 1093.00 1094.90
## 6   2015-01-08   INFY     EQ    1963.55 1985.6 1997.0 1950.0 1979.25 1973.45
## 160 2015-08-20   INFY     EQ    1175.55 1159.0 1168.5 1128.0 1140.90 1134.55
## 136 2015-07-17   INFY     EQ     989.05  991.0 1004.5  989.1  999.75 1001.85
##        VWAP  Volume     Turnover Trades Deliverable.Volume X.Deliverble
## 85  1942.88 1694996 3.293173e+14  68774             920685       0.5432
## 207 1135.02 2022338 2.295390e+14  57656            1151943       0.5696
## 198 1094.85 3209496 3.513919e+14  74364            1874068       0.5839
## 6   1972.78 3391230 6.690160e+14  92752            2686012       0.7920
## 160 1140.09 4184557 4.770761e+14 100655            2703228       0.6460
## 136  997.46 1723989 1.719615e+14  59447            1003912       0.5823
##          artik    tahmin
## 85   0.5397104 1942.3403
## 207 -2.4560996 1137.4761
## 198 -0.4453395 1095.2953
## 6    0.8316350 1971.9484
## 160 -3.4670538 1143.5571
## 136 -2.1212970  999.5813
```

EKK modelimizdeki elde edilen artiklar ve tahmin degerlerini verimize degisken olarak ekledik.

Simdi “Bazi olasi varyans ve standart sapma fonksiyonu tahminleri” nden birincisini inceleyelim.

1. Aciklayici degiskenlere karsilik artiklarin grafigini cizdirip megafon sekli varmi diye bakilir. Eger megafon sekli var ise aciklayici degiskenler ile mutlak artiklarin arasinda regresyon modeli kurulur. Bu regresyon modeli uzerinden tahmin degeri elde edilir. Bu elde elilen tahmin degerlerini σi yerine kullaniriz. Agirliklar da 1/σ2i diye olusturulur.

```
op = par(bg = "azure")
pairs(~artik+Volume+
            High+
           Low+
            Close+
            Last+tahmin,data=veritrain, main="Temel Dagilim Grafigi Matrisi")
```



Artiklar ile Volume sacinim grafigi megafon seklindedir .Bu bagimsiz degisken ile artiklarin mutlak degeri arasinda regresyon modeli kuralim.

```
model1<-lm(abs(veritrain$artik)~Volume,data=veritrain)

weights1<-1/(predict(model1))^2 # agirliklar wi= 1/σ2i
veritrain<-veritrain[, -c(16,17)]

weightedleastsquaremod1<-lm(VWAP~Volume+
            High+
           Low+
            Close+
            Last, data= veritrain, weights = weights1)

summary(weightedleastsquaremod1)
```

```
##
## Call:
## lm(formula = VWAP ~ Volume + High + Low + Close + Last, data = veritrain,
##     weights = weights1)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1769 -0.6720 -0.0449  0.6565  6.1822
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.337e+00  1.420e+00   1.646   0.1013
## Volume      -1.913e-07  2.579e-07  -0.742   0.4591
## High         3.629e-01  2.474e-02  14.667  < 2e-16 ***
## Low          2.528e-01  2.699e-02   9.370  < 2e-16 ***
## Close        5.313e-01  8.133e-02   6.532 5.03e-10 ***
## Last        -1.493e-01  8.234e-02  -1.813   0.0712 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.458 on 205 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 7.114e+05 on 5 and 205 DF,  p-value: < 2.2e-16
```

Summary kodumuza baktigimizda Residual standard error: 1.458 ve Adjusted R-squared: 0.9999 cikmistir.

Simdi “Bazi olasi varyans ve standart sapma fonksiyonu tahminleri” nden ikincisini inceleyelim.

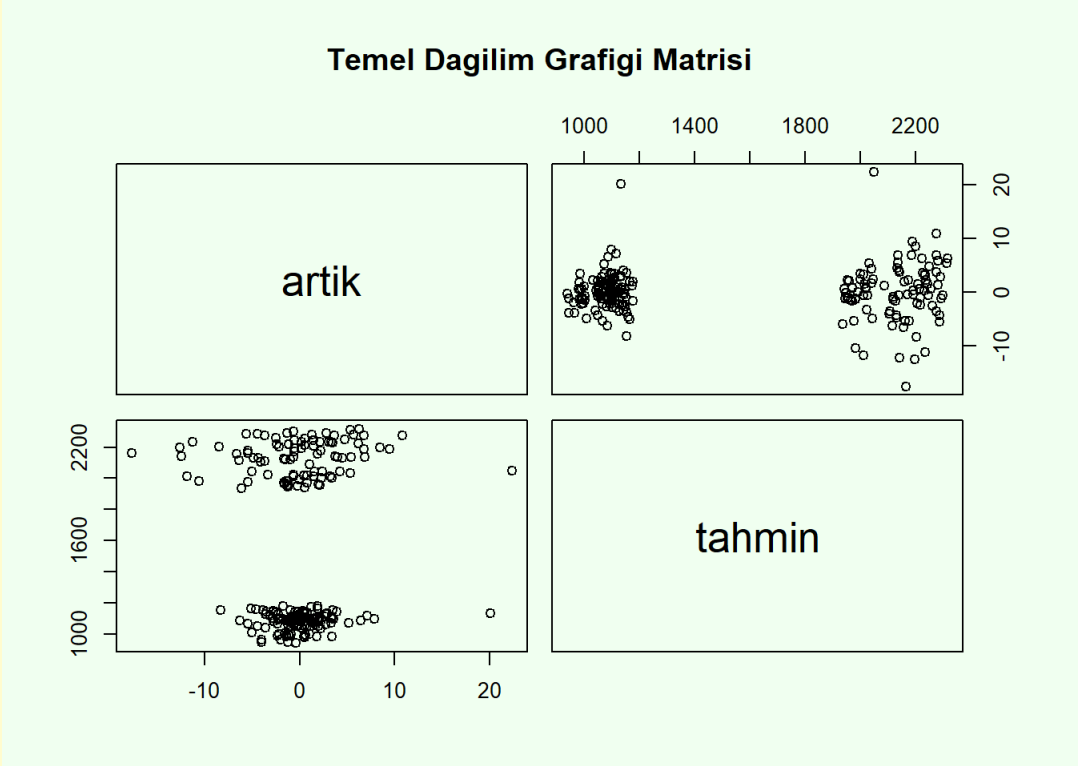
2. Ilk basta kurulan EKK modelindeki tahmin edilen yanitlara (y sapkalara) karsilik ei lerin grafigi megafon seklinde ise artiklarin mutlak degerine karsilik y sapkalarin regresyon modeli kurulur. Bu kurulan modelden elde edilen tahmin degerlerini σi yerine kullaniriz.

Agirliklari da wi= 1/σ2i seklinde olustururuz.

```
library(ggplot2)
veritrain$artik<-(residuals(lmod)) #EKK modelinden elde edilen artiklar (residuals)
veritrain$tahmin<-predict(lmod) #EKK modelinden elde edilen tahminler (prediction)
head(veritrain)
```

```
##           Date Symbol Series Prev.Close    Open    High     Low    Last   Close
## 85  2015-05-07   INFY     EQ    1922.05 1922.0 1967.1 1910.0 1940.00 1944.00
## 207 2015-10-30   INFY     EQ    1145.05 1145.0 1145.0 1130.0 1132.00 1135.45
## 198 2015-10-16   INFY     EQ    1097.35 1099.0 1101.4 1090.0 1093.00 1094.90
## 6   2015-01-08   INFY     EQ    1963.55 1985.6 1997.0 1950.0 1979.25 1973.45
## 160 2015-08-20   INFY     EQ    1175.55 1159.0 1168.5 1128.0 1140.90 1134.55
## 136 2015-07-17   INFY     EQ     989.05  991.0 1004.5  989.1  999.75 1001.85
##        VWAP  Volume     Turnover Trades Deliverable.Volume X.Deliverble
## 85  1942.88 1694996 3.293173e+14  68774             920685       0.5432
## 207 1135.02 2022338 2.295390e+14  57656            1151943       0.5696
## 198 1094.85 3209496 3.513919e+14  74364            1874068       0.5839
## 6   1972.78 3391230 6.690160e+14  92752            2686012       0.7920
## 160 1140.09 4184557 4.770761e+14 100655            2703228       0.6460
## 136  997.46 1723989 1.719615e+14  59447            1003912       0.5823
##          artik    tahmin
## 85   0.5397104 1942.3403
## 207 -2.4560996 1137.4761
## 198 -0.4453395 1095.2953
## 6    0.8316350 1971.9484
## 160 -3.4670538 1143.5571
## 136 -2.1212970  999.5813
```

```
op = par(bg = "honeydew")
pairs(artik~tahmin,data=veritrain, main="Temel Dagilim Grafigi Matrisi")
```



Artiklar ile tahminlerin sacinim grafigi megafon seklinde degildir sonuc olarak bu yontemi verimizde kullanamiyoruz.

Simdi “Bazi olasi varyans ve standart sapma fonkiyonu tahminleri” nden ucuncusunu inceleyelim;

3. Aciklayici degiskene karsi ei kare grafigi artan seklindeyse ei karelere karsilik o aciklayici degiskenin regresyon modeli kurulur. Bu modelin tahminlerini σ2i yerine kullaniriz. Agirliklari da wi= 1/σ2i seklinde olustururuz.

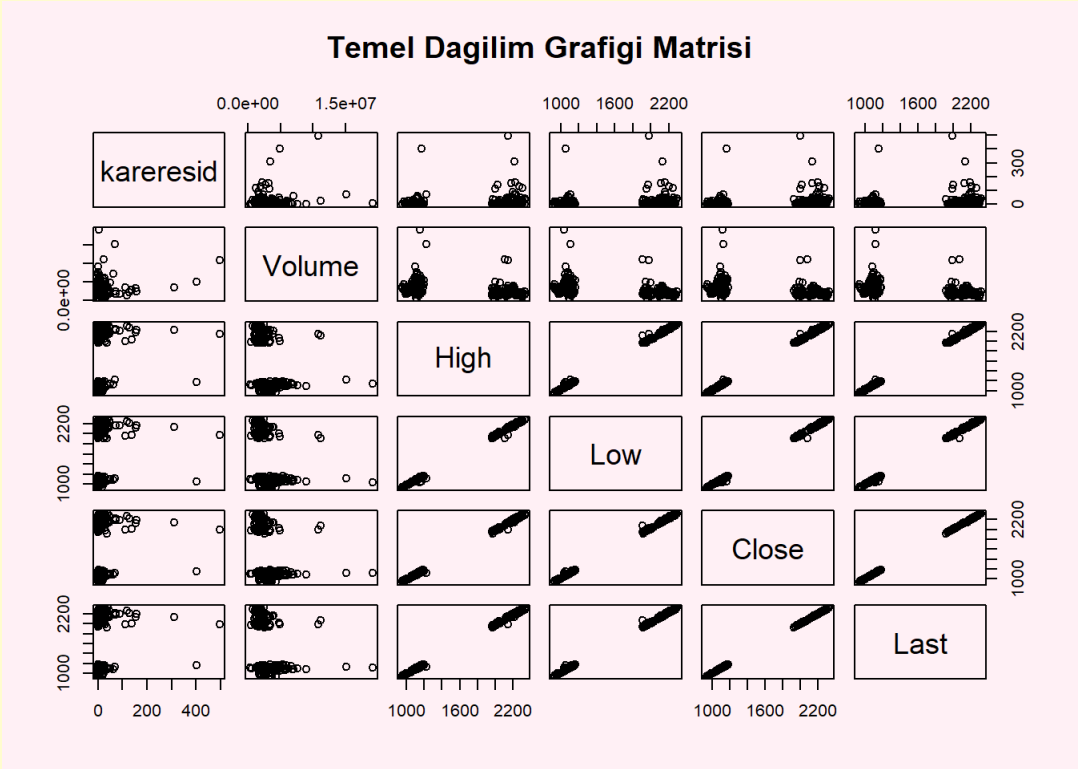
```
library(ggplot2)
veritrain$artik<-(residuals(lmod)) #EKK modelinden elde edilen artiklar (residuals)
veritrain$tahmin<-predict(lmod) #EKK modelinden elde edilen tahminler (prediction)
head(veritrain)
```

```
##            Date Symbol Series Prev.Close    Open    High     Low    Last   Close
## 85   2015-05-07   INFY     EQ    1922.05 1922.0 1967.1 1910.0 1940.00 1944.00
## 207 2015-10-30   INFY     EQ    1145.05 1145.0 1145.0 1130.0 1132.00 1135.45
## 198 2015-10-16   INFY     EQ    1097.35 1099.0 1101.4 1090.0 1093.00 1094.90
## 6    2015-01-08   INFY     EQ    1963.55 1985.6 1997.0 1950.0 1979.25 1973.45
## 160 2015-08-20   INFY     EQ    1175.55 1159.0 1168.5 1128.0 1140.90 1134.55
## 136 2015-07-17   INFY     EQ     989.05  991.0 1004.5  989.1  999.75 1001.85
##         VWAP  Volume     Turnover Trades Deliverable.Volume X.Deliverble
## 85   1942.88 1694996 3.293173e+14  68774             920685        0.5432
## 207 1135.02 2022338 2.295390e+14  57656            1151943        0.5696
## 198 1094.85 3209496 3.513919e+14  74364            1874068        0.5839
## 6    1972.78 3391230 6.690160e+14  92752            2686012        0.7920
## 160 1140.09 4184557 4.770761e+14 100655            2703228        0.6460
## 136  997.46 1723989 1.719615e+14  59447            1003912        0.5823
##           artik    tahmin
## 85    0.5397104 1942.3403
## 207 -2.4560996 1137.4761
## 198 -0.4453395 1095.2953
## 6     0.8316350 1971.9484
## 160 -3.4670538 1143.5571
## 136 -2.1212970  999.5813
```

```
kareresid<-((veritrain$artik)^2)
```

Artiklarimizin karelerine karsilik gelen bagimsiz degiskenlerin grafigi;

```
op = par(bg = "lavenderblush")
pairs(kareresid~Volume+
            High+
           Low+
            Close+
            Last
             ,data=veritrain, main="Temel Dagilim Grafigi Matrisi")
```



Aciklayici degiskene karsi ei kare grafigi artan seklinde oldugu icin ei karelere karsilik o aciklayici degiskenin Regresyon Modeli kurulur.

```
model3<-lm(kareresid~High+Low+Close+Last,data=veritrain)

weights3<-1/(predict(model3))^2 # agirliklar wi= 1/σ2i
veritrain<-veritrain[, -c(16,17)]

weightedleastsquaremod3<-lm(VWAP~Volume+
            High+
           Low+
            Close+
            Last, data= veritrain, weights = weights3)

summary(weightedleastsquaremod3)
```

```
##
## Call:
## lm(formula = VWAP ~ Volume + High + Low + Close + Last, data = veritrain,
##     weights = weights3)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9526 -0.3029 -0.1308  0.0900  3.9331
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.762e+00  1.115e+00   4.272 2.97e-05 ***
## Volume      -1.156e-06  1.544e-07  -7.488 2.04e-12 ***
## High         3.965e-01  8.666e-02   4.575 8.23e-06 ***
## Low          3.529e-01  4.416e-02   7.991 9.53e-14 ***
## Close       -1.127e-01  8.996e-02  -1.252  0.21186
## Last         3.629e-01  1.232e-01   2.946  0.00359 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6213 on 205 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.06e+07 on 5 and 205 DF,  p-value: < 2.2e-16
```

Summary kodumuza baktigimizda Residual standard error: 0.6213 ve Adjusted R-squared: 1 cikmistir.

Simdi “Bazi olasi varyans ve standart sapma fonkiyonu tahminleri” nden dorduncusunu inceleyelim;

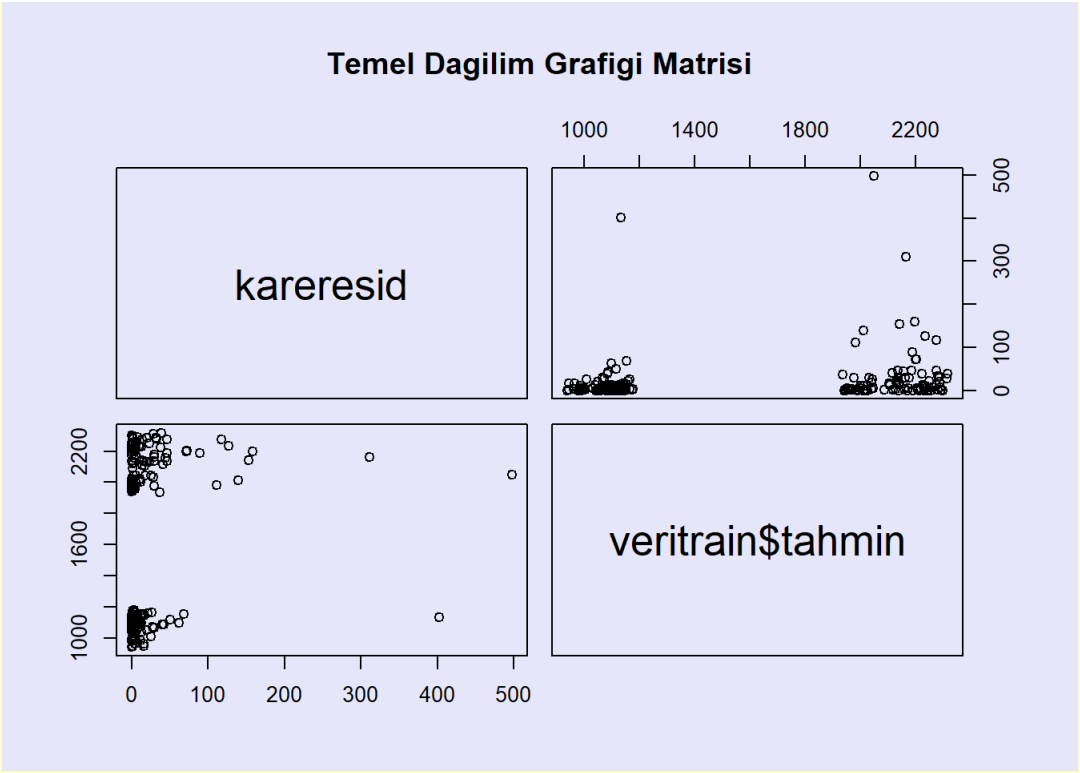
4. Tahmin edilen yanitlara (y sapka) karsi ei kare grafigi artan seklindeyse ei karelere karsilik tahmin edilen yanitlara regresyon modeli kurulur. Bu modelden elde edilen tahminler σ2i tahminleri olarak kullanilir. Agirliklari da wi= 1/σ2i seklinde olustururuz.

Tahmin edilen yanitlara karsilik gelen hatalarin karelerinin grafigi;

```
library(ggplot2)
veritrain$artik<-(residuals(lmod)) #EKK modelinden elde edilen artiklar (residuals)
veritrain$tahmin<-predict(lmod) #EKK modelinden elde edilen tahminler (prediction)
```

```
kareresid<-((veritrain$artik)^2)
```

```
op = par(bg = "lavender")
pairs(kareresid~veritrain$tahmin, main="Temel Dagilim Grafigi Matrisi")
```



Tahmin edilen yanitlara karsilik hatalarin karelerinin grafigi artan seklinde olmadigi icin bu yontemi kullanamayiz.

Simdi bu yontemlerden elde edilen verilerin ozetine bakalim;

```
summary(lmod)
```

```
##
## Call:
## lm(formula = VWAP ~ Volume + High + Low + Close + Last, data = veritrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.6284  -1.6843  -0.1007   1.9955  22.2949
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.925e+00  1.404e+00   2.083   0.0385 *
## Volume      -2.945e-07  2.165e-07  -1.360   0.1752
## High         4.031e-01  2.375e-02  16.976  < 2e-16 ***
## Low          2.394e-01  2.472e-02   9.686  < 2e-16 ***
## Close        5.541e-01  8.811e-02   6.289 1.89e-09 ***
## Last        -1.998e-01  8.899e-02  -2.245   0.0258 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.431 on 205 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 5.956e+05 on 5 and 205 DF,  p-value: < 2.2e-16
```

```
summary(weightedleastsquaremod1)
```

```
##
## Call:
## lm(formula = VWAP ~ Volume + High + Low + Close + Last, data = veritrain,
##     weights = weights1)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1769 -0.6720 -0.0449  0.6565  6.1822
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.337e+00  1.420e+00   1.646   0.1013
## Volume      -1.913e-07  2.579e-07  -0.742   0.4591
## High         3.629e-01  2.474e-02  14.667  < 2e-16 ***
## Low          2.528e-01  2.699e-02   9.370  < 2e-16 ***
## Close        5.313e-01  8.133e-02   6.532 5.03e-10 ***
## Last        -1.493e-01  8.234e-02  -1.813   0.0712 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.458 on 205 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 7.114e+05 on 5 and 205 DF,  p-value: < 2.2e-16
```

```
summary(weightedleastsquaremod3)
```

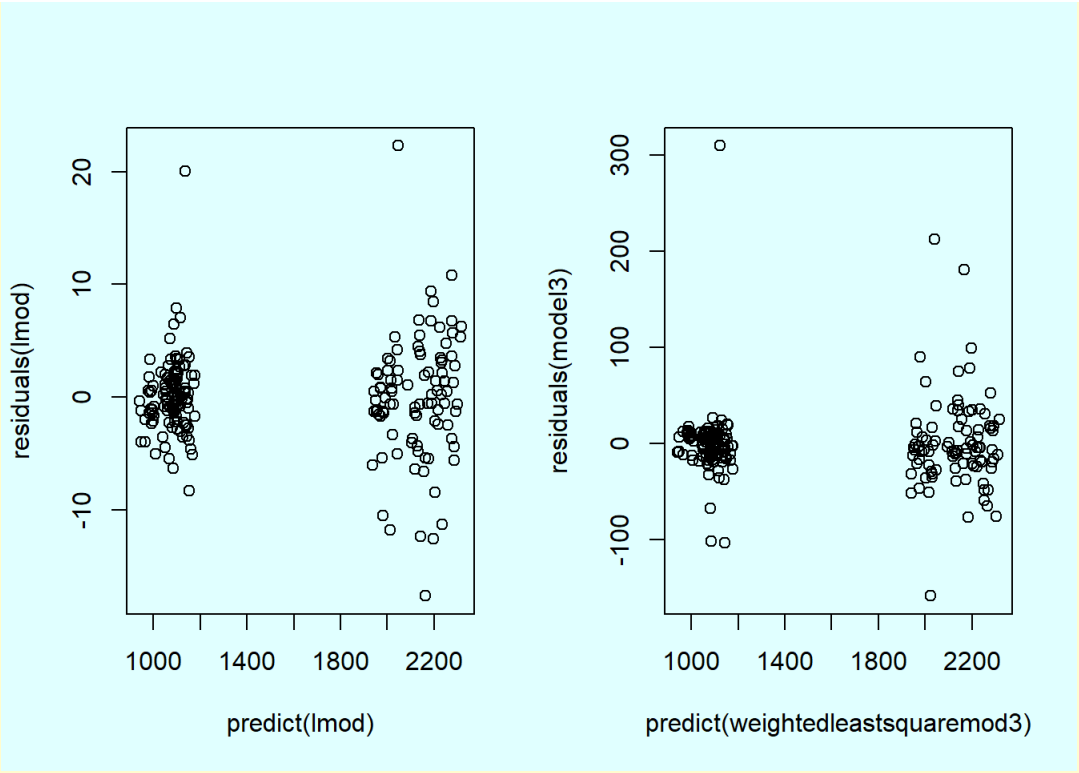
```
##
## Call:
## lm(formula = VWAP ~ Volume + High + Low + Close + Last, data = veritrain,
##     weights = weights3)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9526 -0.3029 -0.1308  0.0900  3.9331
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.762e+00  1.115e+00   4.272 2.97e-05 ***
## Volume      -1.156e-06  1.544e-07  -7.488 2.04e-12 ***
## High         3.965e-01  8.666e-02   4.575 8.23e-06 ***
## Low          3.529e-01  4.416e-02   7.991 9.53e-14 ***
## Close       -1.127e-01  8.996e-02  -1.252  0.21186
## Last         3.629e-01  1.232e-01   2.946  0.00359 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6213 on 205 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.06e+07 on 5 and 205 DF,  p-value: < 2.2e-16
```

EKK modelimizin Residual standard error: 4.431 ve Adjusted R-squared: 0.9999 dir. 1. yontem ile agirliklandirma yaptigimizda Residual standard error: 1.458 ve Adjusted R-squared: 0.9999 dir. 3. yontem ile agirliklandirma yaptigimizda Residual standard error: 0.6213 ve Adjusted R-squared: 1 dir.

En dusuk stanard error a sahip ve en yuksek adjusted r-squared degerine sahip model 3. yontem ile agirliklandirma yaptigimizdaki modeldir. Ayrica agirliklandirma yaptigimda kullanilan bagimsiz degiskenlerimizin katsayilarinda degisimler meydana gelmistir.

Simdi degisken varyanslilik probleminin giderildigini kontrol edelim.

```
op = par(bg = "lightcyan")
par(mfrow=c(1,2))
plot(predict(lmod),residuals(lmod))
plot(predict(weightedleastsquaremod3),residuals(model3))
```




```
library(dplyr)
X<-model.matrix(lmod)
y<-veritrain$VWAP #verimizdeki yanit degiskeni
W<-diag(weights3) #kosegenlerinde agirliklar olan matris
Z<-sqrt(W)%*%X #w matrisinin karekoku ile x in carpimi
yyildiz<-sqrt(W) %*% y
```

Bunlarin tanimlanmasi ile Agirliklandirilmis EKK modeli Basit EKK modeline dondu.

```
Betawls<-solve(t(Z)%*%Z,t(Z)%*%yyildiz) #agirliklandirilmis ekk nin ß larini verir.

cbind(Betawls,coef(weightedleastsquaremod3))
```

```
##                       [,1]          [,2]
## (Intercept)  4.762234e+00  4.762234e+00
## Volume      -1.155990e-06 -1.155990e-06
## High         3.964737e-01  3.964737e-01
## Low          3.529004e-01  3.529004e-01
## Close       -1.126607e-01 -1.126607e-01
## Last         3.629081e-01  3.629081e-01
```

1. sutun donusum ile cikan ß katsayilarini, 2. sutun lm kodu ile elde edilen ß katsayilarini gosterir. Agirliklandirilmis EKK modelindeki ß katsayilari ile donusum yapildiginda cikan ß katsayilari birebir ayni cikti.

```
donart<-yyildiz-Z%*%Betawls #donusturulmus model artiklari
head(cbind(donart,residuals(weightedleastsquaremod3)),15)
```

```
##            [,1]       [,2]
## 85   0.02151942  1.1046539
## 207 -1.41405855 -3.0356582
## 198 -0.10658287 -0.8460218
## 6   -0.23232126 -3.9313823
## 160 -0.28761720 -7.4087633
## 136 -0.38792200 -2.5691977
## 17  -0.18460226 -6.8267114
## 93   0.02680703  0.6512524
## 209  0.11641165  0.5097032
## 204 -0.05901150 -0.5506964
## 125 -2.95255464 -5.0050767
## 41  -0.51538375 -9.2669168
## 178  0.05444156  0.4716302
## 75  -0.16021353 -5.3456626
## 193 -2.00286148 -2.9400807
```

1.sutun donusturulmus artiklari , 2.sutun Agirliklandirilmis EKK modelinin artiklarini gostermektedir. Modelin artiklarina bakildiginda birbirinden ayri cikmistir.

lm modeli ile kurulan Agirliklandirilmis EKK modelinin artiklarini kok icinde w ile carparak donusturulmus modelin artiklari elde edilir.

```
head(cbind(sqrt(W)%*%residuals(weightedleastsquaremod3), donart),10)
```

```
##              [,1]        [,2]
##  [1,]  0.02151941  0.02151942
##  [2,] -1.41405856 -1.41405855
##  [3,] -0.10658287 -0.10658287
##  [4,] -0.23232126 -0.23232126
##  [5,] -0.28761721 -0.28761720
##  [6,] -0.38792200 -0.38792200
##  [7,] -0.18460226 -0.18460226
##  [8,]  0.02680703  0.02680703
##  [9,]  0.11641165  0.11641165
## [10,] -0.05901150 -0.05901150
```

1.sutun donusturulmus artiklari , 2.sutun Agirliklandirilmis EKK modelinin artiklarini gostermekte. Modelin artiklarina bakildiginda birbirleriyle ayni cikmistir.

Eger Agirliklandirilmis EKK uzerinden residuals standart error hesaplarsak;

```
sqrt(sum(residuals(weightedleastsquaremod3)^2)/205)
```

```
## [1] 6.08308
```

Agirliklandirilmis EKK modelinin residuals standat erroru ile ayni cikmamistir.

Simdi Donusturulmus artiklarin standart errorunu hesaplayalim;

```
sqrt(sum(donart^2)/205)
```

```
## [1] 0.6213453
```

Simdi Agirliklandirilmis modelimiz icin BREUSCH-PAGAN TESTI yapalim;

BREUSCH-PAGAN TESTI

H0:Heterosce Dosticity (Degisken Varyanslilik) problemi yok. H1:Heterosce Dosticity (Degisken Varyanslilik) problemi vardir.

```
#install.packages("lmtest")
library(lmtest)
bpmod<-lm(donart^2~ Volume+
            High+
           Low+
            Close+
            Last,data=veritrain)
summary(bpmod)
```

```
##
## Call:
## lm(formula = donart^2 ~ Volume + High + Low + Close + Last, data = veritrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8927 -0.4355 -0.2598 -0.0531 14.6458
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.027e+00  4.324e-01   2.375   0.0185 *
## Volume      -5.007e-08  6.666e-08  -0.751   0.4534
## High        -6.050e-03  7.312e-03  -0.827   0.4090
## Low         -5.454e-03  7.611e-03  -0.717   0.4744
## Close       -3.658e-02  2.713e-02  -1.348   0.1791
## Last         4.776e-02  2.740e-02   1.743   0.0828 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.364 on 205 degrees of freedom
## Multiple R-squared:  0.03642,    Adjusted R-squared:  0.01291
## F-statistic:  1.55 on 5 and 205 DF,  p-value: 0.176
```

BREUSCH-PAGAN Testimizin sonucuna gore p-value: 0.176 > 0.05 oldugundan H0 hipotezi kabul edilir yani degisken varyanslilik problemi yoktur deriz. Goruldugu uzere Agirliklandirma yaparak degisken varyanslilik problemini ortadan kaldirmis olduk.

ACIKLAYICI DEGISKENLERLE ILGILI PROBLEMLER

IC ILISLI (COLLINEARITY)

Iki degisken arasi lineer iliskiyi gosterir.Eger bir aciklayici degisken ve bir diger aciklayici degiskenin veya degiskenlerin lineer bir kombinasyonlari ise bu durumda x transpoz x matrisi (X'X) singuler olur ve tersi alinamaz. Bu durumdan ilgili degiskenlerden biri modelden cikartilarak kurtulunur. Gozlem sayisi arttikca ic iliski durumu azalir.

```
library(dplyr)
nsts <- NSTS%>%select(c("VWAP","Volume","High","Low","Close","Last" ))
head(nsts,10)
```

```
##       VWAP   Volume    High     Low   Close    Last
## 1  1971.34   500691 1982.00 1956.90 1974.40 1971.00
## 2  2003.25  1694580 2019.05 1972.00 2013.20 2017.95
## 3  2004.59  2484256 2030.00 1977.50 1995.90 1996.00
## 4  1954.82  2416829 1985.00 1934.10 1954.20 1965.10
## 5  1962.59  1812479 1974.75 1950.00 1963.55 1966.05
## 6  1972.78  3391230 1997.00 1950.00 1973.45 1979.25
## 7  2037.69 11215832 2109.00 1913.05 2074.45 2075.30
## 8  2099.40  3189722 2119.20 2075.00 2115.95 2112.95
## 9  2089.42  2200309 2107.80 2075.00 2088.90 2092.00
## 10 2110.88  2480315 2133.00 2092.60 2128.65 2129.00
```

```
set.seed(2)
index<-sample(1:nrow(nsts),round(nrow(nsts)*0.85))
veritrain<-nsts[index,]
veritest<-nsts[-index,]
```

```
lmod1<-lm(VWAP~Volume+
            High+
           Low+
            Close+
            Last,data=veritrain)
summary(lmod1)
```

```
##
## Call:
## lm(formula = VWAP ~ Volume + High + Low + Close + Last, data = veritrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.6284  -1.6843  -0.1007   1.9955  22.2949
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.925e+00  1.404e+00   2.083   0.0385 *
## Volume      -2.945e-07  2.165e-07  -1.360   0.1752
## High         4.031e-01  2.375e-02  16.976  < 2e-16 ***
## Low          2.394e-01  2.472e-02   9.686  < 2e-16 ***
## Close        5.541e-01  8.811e-02   6.289 1.89e-09 ***
## Last        -1.998e-01  8.899e-02  -2.245   0.0258 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.431 on 205 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 5.956e+05 on 5 and 205 DF,  p-value: < 2.2e-16
```

Kurulan regresyon modelinin anlamliligina baktigimizda p value=yaklasik 0 < 0.05 oldugundan kurulan model anlamlidir deriz.

Simdi Collinearity teshisi icin korelasyon matrisi, kosul inceksi ve vif e bakacagiz.

KORELASYON MATRISI

Simdi x in korelasyon matrisine bakalim. Bunun icin yanit degiskenini (VWAP) veriden cikartmaliyiz.Geri kalanlarin korelasyon matrisine bakmaliyiz.

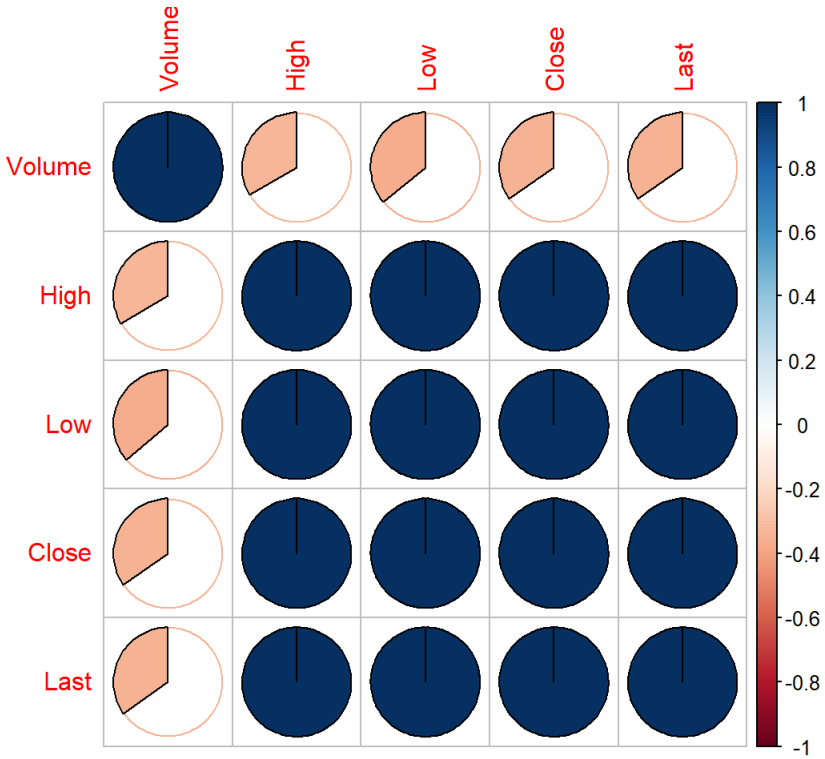
```
cor(veritrain[,-c(1)])
```

```
##            Volume       High        Low      Close       Last
## Volume  1.0000000 -0.3341445 -0.3603287 -0.3479232 -0.3478308
## High   -0.3341445  1.0000000  0.9992291  0.9994961  0.9995049
## Low    -0.3603287  0.9992291  1.0000000  0.9995296  0.9995324
## Close  -0.3479232  0.9994961  0.9995296  1.0000000  0.9999782
## Last   -0.3478308  0.9995049  0.9995324  0.9999782  1.0000000
```

Korelasyon matrisine baktigimizda ornegin en yuksek Close ile Last (aralarindaki korelasyon 0.9999782) bagimsiz degiskenlerinin iliskili oldugu gorulmektedir. Diger bagimsiz degiskenler arasinda da korelasyon oldukca yuksektir.

Bu korelasyonlara simdi korelasyon plotu ile bakalim.

```
library(corrplot)
corrplot(cor(veritrain[,-1]),method = "pie", order="hclust")
```



Korelasyon plotunda Pozitif korelasyonlar mavi, negatif korelasyonlar kirmizi renkte gosterilir.

Korelasyon plotu ile korelasyon matrisimizin sonuclari ayni cikmistir.Bagimsiz degiskenlerin arasinda pozitif yonlu iliskinin yuksek oldugu gorulmektedir. (mavinin tonuna gore en yuksek iliskiden en dusuk iliskiye gore koyu maviden aciga dogru gidiyor.)

KOSUL INDEKSI

Kappa degeri > 30 ise orta derece collinearity , kappa degeri > 100 ise guclu collinearity oldugunu gosterir.

Kurulan regresyon modelimizi matrix haline donusturelim.

```
x <- model.matrix(lmod1)[,-1]
head(x,5)
```

```
##      Volume   High  Low   Close    Last
## 85  1694996 1967.1 1910 1944.00 1940.00
## 207 2022338 1145.0 1130 1135.45 1132.00
## 198 3209496 1101.4 1090 1094.90 1093.00
## 6   3391230 1997.0 1950 1973.45 1979.25
## 160 4184557 1168.5 1128 1134.55 1140.90
```

Verimizden yanit degiskenini cikartip sadece aciklayici degiskenlerden olusan matrix formuna donusturuyoruz.

Simdi x transpoz x in eigen valuelerini hesaplayalim.

```
e <- eigen(t(x)%*%x)$values
e
```

```
## [1] 2.937826e+15 1.109449e+09 3.364775e+04 2.872275e+04 1.275973e+03
```

Kappa degeri : sqrt(En buyuk ozdeger / En kucuk ozdeger)

```
k <- sqrt(max(e)/min(e))
k
```

```
## [1] 1517373
```

Sonucumuzda Kappa=1517373 > 30 oldugundan sonucumuza gore collinearity (ic iliski) problemi vardir.

VIF

Xi degiskenlerinin diger bagimsiz degiskenler ile regresyonundan elde edilen R kare degerlerinin yukekligi collinearity (ic iliski)nin varligini

gosterir. Buna bagli gelistirilmis olcut VIF dir.

VIF degerinin 10 dan buyuk olmasi collinearity (ic iliski) probleminin oldugunu soyler.

Hazir kod ile vif degerlerine bakmak icin car paketi kullanilir.

```
library(car)
vif(lmod1)
```

```
##       Volume         High          Low        Close         Last
##     2.300823  1714.563922  1787.646264 23215.682035 23676.301984
```

Bagimsiz degiskenlerimiz icin hesaplatilan vif degerlerimiz 10 dan buyuk oldugundan bagimsiz degiskenler arasinda collinearity(ic iliski) problemi vardir deriz.

Bu uc tanimlama yontemi de bize bu veride collinearity problemi oldugunu isaret etmektedir.

RIDGE- LASSO - ELASTICNET

RIDGE

Ridge regresyon EKK optimizasyon yontemine bir kisit getirir. Bu kisit ß katsayilarinin karelerinin toplaminin uzerinedir.

Lambda buyudukce ß parametreleri 0 a dogru yaklasir.Parametrelerin buyumesi parametre tahminlerinin varyansini dusurur. Negatif etkisi modele yanlilik katmasidir.

Var(ßsapka ridge)= σ^2 / (x'x + λI) burada λ yi buyuk tutarsak varyans kuculur ayni zamanda yanlilik artar. Ikisi arasinda denge kuracak sekilde λ belirlenmelidir. Multicollinearity problemini halledebilecek en kucuk λ yi belirlemeliyiz.

EKK tahmin edicisi yansiz bir tahmin edici iken Ridge Regresyonunun tahmin edicisi yanli bir tahmin ediciye donusur. Ic iliski durumlarinda varyanslar buyukken Ridge Regresyonunda varyanslar daha kucuktur.

Ridge Regresyon da aciklayici degiskenlerin tamamini modele dahil eder ve kompleks model olusmasini saglar. (Modele degisken ekledikce hata duser fakat kompleks hale de gelebilir.)

```
lambda<-10^seq(-3, 5, length.out = 100) #lamda icin dizi olusturma
```

Ridge de lambda parametresinin secimi icin en iyi yontem Cross Validationdur. Bu sebeple lambdalar icin oncelikle bir dizi olusturduk.

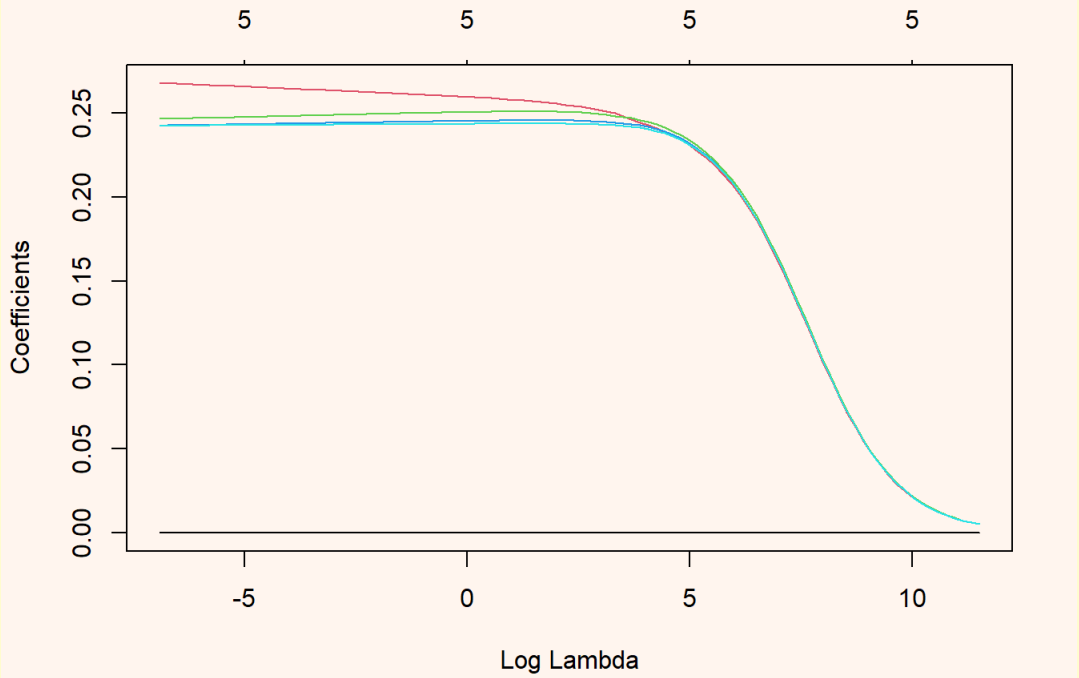
```
x<-as.matrix(veritrain[,-1])
head(x,10)
```

```
##       Volume    High    Low   Close    Last
## 85   1694996 1967.10 1910.0 1944.00 1940.00
## 207  2022338 1145.00 1130.0 1135.45 1132.00
## 198  3209496 1101.40 1090.0 1094.90 1093.00
## 6    3391230 1997.00 1950.0 1973.45 1979.25
## 160  4184557 1168.50 1128.0 1134.55 1140.90
## 136  1723989 1004.50  989.1 1001.85  999.75
## 17   2371582 2225.00 2165.1 2215.05 2211.90
## 93   2568019 2035.70 1986.0 2022.35 2022.20
## 209  1807341 1150.45 1133.4 1145.50 1140.15
## 204  2631874 1155.00 1144.1 1149.40 1147.35
```

Train verimizi matrix haline donusturuyoruz (yanit degiskenini VWAP i cikardik)

Lambdanin farkli degerleri icin degiskenlerin aldigi farkli degerlerin grafigini cizdirelim; Ridge Regresyonda alpha = 0 degerini alir.

```
library(glmnet)
ridgemodel<-glmnet(x,veritrain$VWAP,alpha = 0,lambda = lambda)
op = par(bg = "seashell1")
plot(ridgemodel,xvar = "lambda")
```



Bu grafik lambdanin farkli degerleri icin degiskenlerin aldigi farkli degerlerin grafigidir.

Grafikte kirmizi cizgiye baktigimizda stabil olmayan durum var. Bu degiskenin degeri digerlerine gore cok az bir degisim gosteriyor. Bu multicollinearity nin bir etkisidir.

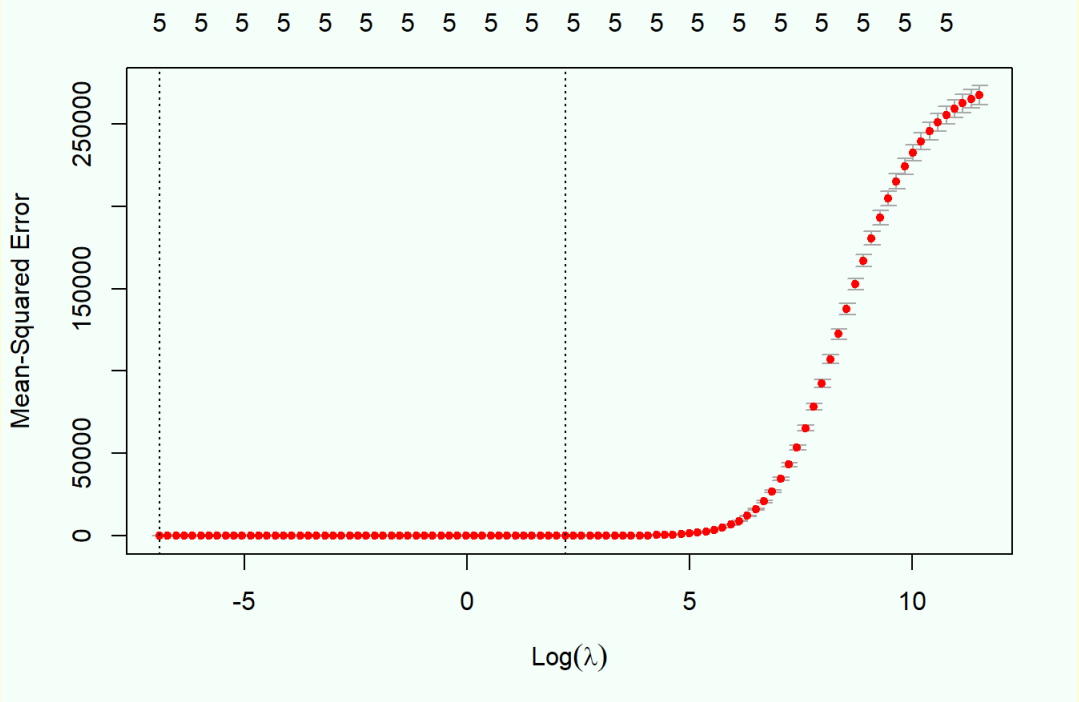
Grafigimizin ust bolumunde gozuken 5 degerleri Ridge Regresyonun hicbir bagimsiz degiskeni atmayip tamamini kullanmasindandir.

Grafikte gorulen her bir cizgi bir degiskene karsilik gelmektedir.

Bu grafik ile model katsayilarinin lambdaya gore nasil degistigini gosterdik. Simdi Cross Validation Yontemi ile optimal lambdayi belirleyelim.

Cross Validationda 9 fold ile model kurulur 10 uncu fold da bu modelin performansi incelenir MSE' ye bakilir. Herbir lambda degeri icin Cross Validation yapilir. Tum bulunan MSE ortalamalari alinir ve minimum RMSE degerini veren lambda degeri secilir.

```
op = par(bg = "mintcream")
cv.fitridge<-cv.glmnet(x,veritrain$VWAP,alpha=0,lambda = lambda) #yukarida olusturulan lambda dizisi icin
plot(cv.fitridge)
```



Grafikte gozuken kirmizi noktalar her lambda degeri icin 10 folddan gelen MSE lerin otalamasidir.

Grafikteki ilk dogru minimum MSE degerini veren lambda degerinin logaritmasini, ikinci dogru ise foldlardan elde edilen MSE degerlerinin standart sapmasinin 1 oldugu lambda degerinin logaritmasini gostermektedir.

Grafigimizin ust bolumunde gozuken 5 degerleri Ridge Regresyonunun tum degiskenleri kullanmasidir.

Simdi optimal lambda degerini kullanarak Ridge Regresyon modelimizi kuralim ve bu modelin test verisi uzerinde RMSE ve R kare degerlerini hesaplayalim.

Performans kiyaslamasi her zaman test veri seti uzerinden yapilir. Cunku multicollinearity nin train'e bir etkisi yoktur ama test'e etkisi vardir.

```
optimumlambda<-cv.fitridge$lambda.min #optimum lambda icin bunlar icerisinden min olan secilir.
optimumlambda
```

```
## [1] 0.001
```

```
lambda_1SE<-cv.fitridge$lambda.1se
lambda_1SE
```

```
## [1] 9.111628
```

Grafigimizde cikan cizgilerimizin yerine bakarsak ;

Ilk Dogru: log(λmin)=log(0.001)= -6.907755

Ikinci Dogru : log(λ1SE)=log(9.111628)= 2.209551

Ridge Regresyon modeli;

```
ridgemodel<-glmnet(x,veritrain$VWAP,alpha=0,lambda=optimumlambda)
```

RMSE ve R kare hesaplayan fonksiyonlar;

```
rmse<-function(true, predicted,n) {sqrt(sum((predicted - true)^2)/n)}

ypredictedridge <- predict(ridgemodel, s = optimumlambda, newx = as.matrix(veritest[,-1]))# kurulan model uzerinden elde edilen tahmin degerleri

rsquare <- function(true, predicted) {
  sse <- sum((predicted - true)^2)
  sst <- sum((true - mean(true))^2)
  rsq <- 1 - sse / sst
  rsq }
```

Test verisi uzerinden Ridge modelinin R karesini hesaplatalim;

```
ridgerkare<-rsquare(veritest$VWAP,ypredictedridge)
ridgerkare
```

```
## [1] 0.9997101
```

Ridge regresyon icin test verisi uzerinden R kare 0.9997101 cikmistir.

Test verisi uzerinden Ridge Modelinin RMSE sini hesaplatalim;

```
ridgermse<-rmse(veritest$VWAP,ypredictedridge,length(veritest$VWAP))
ridgermse
```

```
## [1] 9.104282
```

Ridge Regresyon icin test verisi uzerinden RMSE 9.104282 cikmistir.

AIC ve BIC performans degerlendirme kriterleridir. Modeller arasinda kiyas yaparken kullanilir. Genel olarak AIC ve BIC degerleri daha kucuk olan model diger modellere gore daha iyidir. AIC ve BICde artik degerler kullanilir.

Simdi Test verisi icin artiklar;

```
ridgeartik<-veritest$VWAP-(ypredictedridge)
head(ridgeartik,10)
```

```
##              1
## 4   -7.4986075
## 21   6.1513650
## 30   3.5755672
## 31   0.4488721
## 39 -11.6952021
## 40  11.0478618
## 46 -29.9604325
## 57  -5.2757745
## 58   6.4165778
## 60   5.2276908
```

AIC

```
ridgeaic<-nrow(nsts)*(log(2*pi)+1+log((sum((ridgeartik)^2)/nrow(nsts))))+((length(ridgemodel$VWAP)+1)*2)
ridgeaic
```

```
## [1] 1329.508
```

Ridge Regresyon icin AIC degeri 1329.508 cikmistir.

BIC

```
ridgebic<-nrow(nsts)*(log(2*pi)+1+log((sum((ridgeartik)^2)/nrow(nsts))))+((length(ridgemodel$VWAP)+1)*log(nr
ow(nsts)))
ridgebic
```

```
## [1] 1333.022
```

Ridge Regresyon icin BIC degeri 1333.022 cikmistir.

LASSO

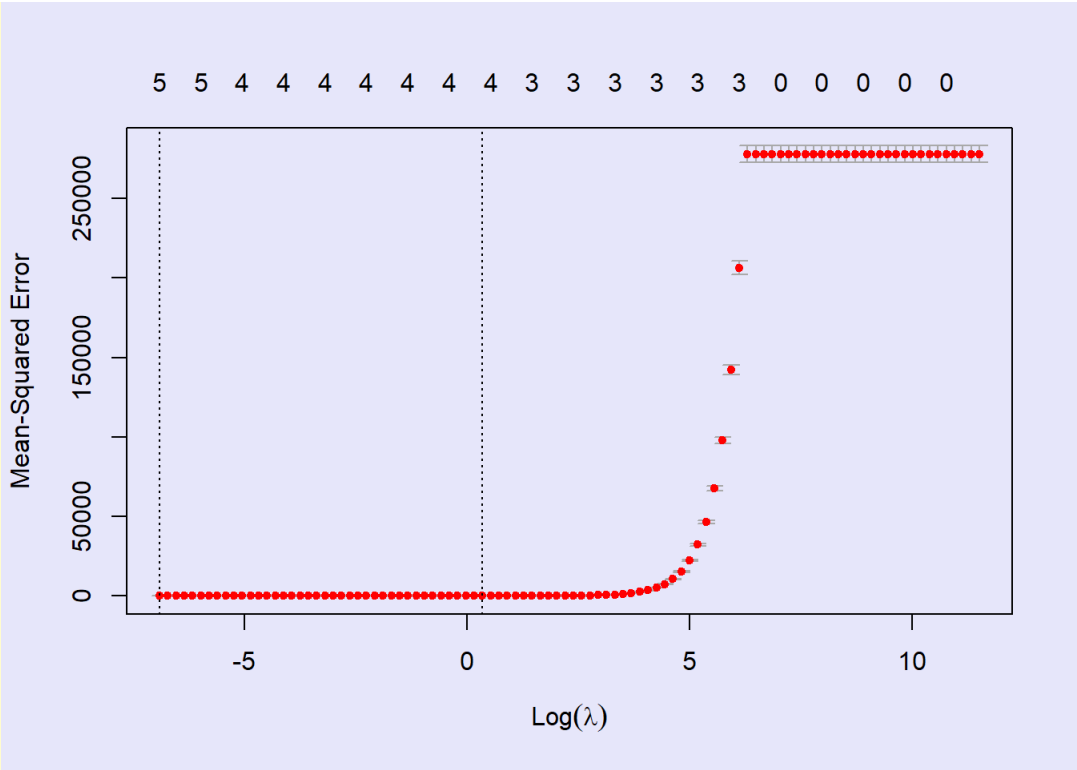
Degisken secimi icin kullanilir. Lassoda katsayilarin bazilari Ridgeden farkli olarak sifirlanmaktadir.

Lasso multicollinearity problemini cozerken ayni zamanda degisken secimi de yapabilme yetenegine sahiptir. Lassoda da λ ceza parametresi vardir. λ buyudukce modelden atilan (disarida birakilan) degisken sayimiz artiyor.

Bazen Lasso cok fazla degiskeni disarida birakmaktadir. Bu modelin tahmin performansini dusurur. Modelde gormek istedigimiz degiskeni goremeyebiliriz.

Ilk olarak Cross Validation ile Optimal Lambda degerini belirleyelim. Lasso Regresyonunda alpha = 1 degerini alir.

```
op = par(bg = "lavender")
cv.fitlasso<-cv.glmnet(x,veritrain$VWAP,alpha=1,lambda = lambda) #lambda dizisinde belirledigimiz lamdalar k
ullanilir.
plot(cv.fitlasso)
```



Grafigimizin ust bolumunde gozuken degerler Lasso Regresyonunun bagimsiz degiskenlerinin tamamini kullanmayarak modelden atmasindan dolayi kaynaklanir.

Grafikte gozuken kirmizi noktalar her lambda degeri icin 10 folddan gelen MSE lerin ortalamasidir.

Grafikteki ilk dogru minimum MSE degerini veren lambda degerinin logaritmasini, ikinci dogru ise foldlardan elde edilen MSE degerlerinin standart sapmasinin 1 oldugu lambda degerinin logaritmasini gostermektedir.

Optimal Lambda degeri;

```
optimallambda<-cv.fitlasso$lambda.min
optimallambda
```

```
## [1] 0.001
```

```
lambda_1SE<-cv.fitlasso$lambda.1se
lambda_1SE
```

```
## [1] 1.417474
```

Grafigimizde cikan cizgilerimizin yerine bakarsak ;

Ilk Dogru: log(λmin)=log(0.001)= -6.907755

Ikinci Dogru : log(λ1SE)=log(1.417474)= 0.3488764

Simdi Optimal Lambda degerini kullanarak Lasso Regresyon modelimizi kuralim ve bu modelin test verisi uzerinde RMSE ve R kare degerlerini hesaplayalim.

Lasso regresyon modeli;

```
lassomodel<-glmnet(x,veritrain$VWAP,alpha=1,lambda=optimallambda)
coef(lassomodel)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)  1.274323e+01
## Volume      -2.739531e-06
## High         8.895992e-01
## Low          7.806363e-02
## Close        2.476695e-02
## Last        -4.611594e-03
```

Burada hicbir degerimiz atilmamistir.

Simdi test verisi uzerinden Lasso Regresyon modelimizin performansina bakalim.

Kurulan model uzerinden elde edilen tahmin degerleri;

```
ypredictedlasso <- predict(lassomodel, s = optimallambda, newx = as.matrix(veritest[,-1]))
head(ypredictedlasso,10)
```

```
##           1
## 4  1962.297
## 21 2138.335
## 30 2298.069
## 31 2303.441
## 39 2283.203
## 40 2257.678
## 46 2231.131
## 57 2229.516
## 58 2208.075
## 60 2198.172
```

Test verisi icin Lasso Regresyon modelinin R karesi;

```
lassorkare<-rsquare(veritest$VWAP,ypredictedlasso)
lassorkare
```

```
## [1] 0.999713
```

Test verisi icin Lasso Regresyon Modelinin R karesi 0.999713 cikmistir.

Test verisi icin Lasso Regresyon Modelinin RMSE si;

```
lassormse<-rmse(veritest$VWAP,ypredictedlasso,length(veritest$VWAP))
lassormse
```

```
## [1] 9.058652
```

Test verisi icin Lasso Regresyon Modelinin RMSE si 9.058652 cikmistir.

Lasso Regresyon Modeli icin artiklar;

```
lassoartik<-veritest$VWAP-(ypredictedlasso)
head(lassoartik,10)
```

```
##              1
## 4   -7.4768073
## 21   6.1054981
## 30   3.5605108
## 31   0.4492543
## 39 -11.6128555
## 40  10.9815825
## 46 -29.8105053
## 57  -5.2459078
## 58   6.3549421
## 60   5.2477094
```

Lassonun artiklari uzerinden AIC;

```
lassoaic<-nrow(nsts)*(log(2*pi)+1+log((sum((lassoartik)^2)/nrow(nsts))))+((length(lassomodel$VWAP)+1)*2)
lassoaic
```

```
## [1] 1327.016
```

Lasso Regresyon icin AIC degeri 1327.016 cikmistir.

Lassonun artiklari uzerinden BIC;

```
lassobic<-nrow(nsts)*(log(2*pi)+1+log((sum((lassoartik)^2)/nrow(nsts))))+((length(lassomodel$VWAP)+1)*log(nr
ow(nsts)))
lassobic
```

```
## [1] 1330.53
```

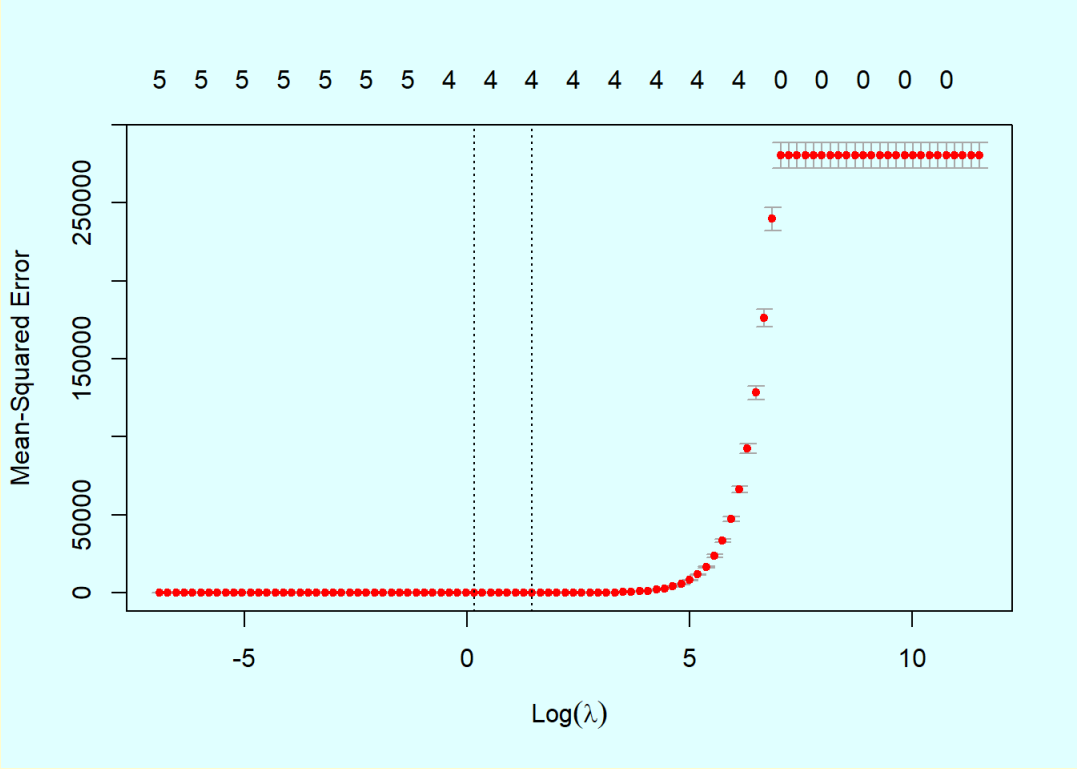
Lasso Regresyon icin BIC degeri 1330.53 cikmistir.

ELASTIC-NET

Elatic net Ridge Regresyon Modeli ile Lasso Regresyon Modelinin bir kombinasyonudur.Ridge tarzi cezalandirma ve Lasso tarzi degisken secimi yapar. Lasso ve Ridge'de bulunan λ parametresi haricinde ikinci bir parametre olan α parametresi de vardir. Ozellikle yuksek korelasyonlu degisken gruplari oldugunda onerilir.

Ilk olarak Cross Validation ile Optimal Lambda degerini belirleyelim. λ= 0.5 alindiginda Elastic Net Regresyon Modeli elde edilir.

```
op = par(bg = "lightcyan")
cv.fitelasticnet<-cv.glmnet(x,veritrain$VWAP,alpha=0.5,lambda = lambda)
plot(cv.fitelasticnet)
```



Grafigimizin ust bolumunde gozuken degerler Elastic Net Regresyonun bagimsiz degiskenlerinin tamamini kullanmayarak modelden atmasindan dolayi kaynaklanir.

Grafikte gozuken kirmizi noktalar her lambda degeri icin 10 folddan gelen MSE'lerin ortalamasidir.

Grafikteki ilk dogru minimum MSE degerini veren lambda degerinin logaritmasini, ikinci dogru ise foldlardan elde edilen MSE degerlerinin standart sapmasinin 1 oldugu lambda degerinin logaritmasini gostermektedir.

Optimal Lambda degeri;

```
optlambda<-cv.fitelasticnet$lambda.min
optlambda
```

```
## [1] 1.176812
```

```
lambda_1SE<-cv.fitelasticnet$lambda.1se
lambda_1SE
```

```
## [1] 4.328761
```

Grafigimizde cikan cizgilerimizin yerine bakarsak ;

Ilk Dogru: log(λmin)=log(1.176812)= 0.1628091

Ikinci Dogru : log(λ1SE)=log(4.328761)= 1.465281

Simdi Optimal Lambda degerini kullanarak Elastic Net Regresyon Modelimizi kuralim ve bu modelin test verisi uzerinde RMSE ve R kare

degerlerini hesaplayalim.

```
elasticmodel<-glmnet(x,veritrain$VWAP,alpha=0.5,lambda=optlambda)
coef(elasticmodel)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept) 3.55769510
## Volume      .
## High        0.35507256
## Low         0.35341373
## Close       0.22439404
## Last        0.06493091
```

5 degiskenli modelimizde Elastic Net Regresyonu 1 degiskeni (Volume) atilmistir.Bu degiskene iliskin katsayilar sifirlanmistir.

Simdi test verisi uzerinden Elastic Net Regresyon Modelimizin performansina bakalim.

Kurulan model uzerinden elde edilen tahmin degerleri;

```
ypredictedelasticnet <- predict(elasticmodel, s = optlambda, newx = as.matrix(veritest[,-1]))
head(ypredictedelasticnet,10)
```

```
##           1
## 4  1958.021
## 21 2142.815
## 30 2299.931
## 31 2301.791
## 39 2270.401
## 40 2264.794
## 46 2209.258
## 57 2223.380
## 58 2215.124
## 60 2195.049
```

Test verisi icin Elastic Net Regresyon modelinin R karesi;

```
elasticrkare<-rsquare(veritest$VWAP,ypredictedelasticnet)
elasticrkare
```

```
## [1] 0.9999421
```

Test verisi icin Elastic Net Regresyon Modelinin R karesi 0.9999421 cikmistir.

Test verisi icin Elastic Net Regresyon Modelinin RMSE si;

```
elasticrmse<-rmse(veritest$VWAP,ypredictedelasticnet,length(veritest$VWAP))
elasticrmse
```

```
## [1] 4.069581
```

Test verisi icin Elastic Net Regresyon Modelinin RMSE si 4.069581 cikmistir.

Elastic Net Regresyon Modeli icin artiklar;

```
elasticartik<-veritest$VWAP-(ypredictedelasticnet)
head(elasticartik,10)
```

```
##             1
## 4  -3.2007819
## 21  1.6249481
## 30  1.6987566
## 31  2.0985350
## 39  1.1892670
## 40  3.8655112
## 46 -7.9381194
## 57  0.8899631
## 58 -0.6941946
## 60  8.3712884
```

Elastic Net in artiklari uzerinden AIC;

```
elasticaic<-nrow(nsts)*(log(2*pi)+1+log((sum((elasticartik)^2)/nrow(nsts))))+((length(elasticmodel$VWAP)+1)*
2)
elasticaic
```

```
## [1] 930.1267
```

Elastic Net Regresyon icin AIC degeri 930.1267 cikmistir.

Elastic Net in artiklari uzerinden BIC;

```
elasticbic<-nrow(nsts)*(log(2*pi)+1+log((sum((elasticartik)^2)/nrow(nsts))))+((length(elasticmodel$VWAP)+1)*
log(nrow(nsts)))
elasticbic
```

```
## [1] 933.6402
```

Elastic Net Regresyon icin BIC degeri 933.6402 cikmistir.

Simdi kurulan modellerde AIC, BIC, RMSE ve R kare degerlerini karsilastirip model secimi yapalim.

```
tablo<-matrix(c(ridgeaic,ridgebic,ridgermse,ridgerkare,
              lassoaic,lassobic,lassormse,lassorkare,
              elasticaic,elasticbic,elasticrmse,elasticrkare),3,4,byrow = TRUE)

row.names(tablo)<-c("Ridge","Lasso","Elasticnet")

colnames(tablo)<-c("AIC","BIC","RMSE","Rkare")

tablo
```

```
##                  AIC       BIC     RMSE     Rkare
## Ridge      1329.5083 1333.0217 9.104282 0.9997101
## Lasso      1327.0161 1330.5296 9.058652 0.9997130
## Elasticnet  930.1267  933.6402 4.069581 0.9999421
```

Secim yaparken AIC BIC ve RMSE degerleri kucuk , R kare degeri buyuk olan model secilmelidir.

Modellerin R2 degerlerini, RMSE degerlerini, AIC, BIC degerlerini karsilastirdigimizda;

En kucuk rmse degeri Elasticnet Regresyon Modelinde (RMSE: 4.069581), En kucuk AIC degeri Elasticnet Regresyon Modelinde (AIC: 930.1267), En kucuk BIC degeri Elasticnet Regresyon Modelinde (BIC: 930.1267), En buyuk R kare degeri Elasticnet regresyon modelinde (R Kare: 930.1267). Bu sebeple calisabilecegimiz en iyi regresyon modeli Elastic Net Regresyon Modelidir. Bu veriyi modellemede Elastic Net Regresyon Modeli daha uygundur.

Multicollinearity nin cozumunde Ridge, Lasso, Elastic Net haricinde Temel Bilesenler Regresyonu da kullanilir. Simdi multicollinearity probleminden Temel Bilesenler Yolu ile kurtulmayi deneyelim.

TEMEL BILESENLER REGRESYONU

Veri setindeki tum degiskenler vektoru ifade eder. Degiskenler arasinda iliski olmadiginda bu vektorler birbirlerine diktir.Temel bilesenler analizinin amaci, X matrisine bir donusum uygulayarak birbirlerine dik(ortogonal) vektorlerden olusan bir sistem elde etmektir. Yuksek boyutlu verilerde dusuk boyutlu dogrusal yapi elde etmek icin kullanilan bir yontemdir. Vektor boyutlari kisa olanlar goz ardi edilir.

```
set.seed(3)
index<-sample(1:nrow(nsts),round(nrow(nsts)*0.95))
veritrain<-nsts[index,]
veritest<-nsts[-index,]
```

Oncelikle Train veri seti uzerinde kurdugumuz EKK Regresyon Modelini kullanarak, hem train hem de test veri seti uzerinde RMSE degerlerini hesaplayalim.

EKK Regresyon Modelimiz;

```
lmod1<-lm(VWAP~Volume+
            High+
           Low+
            Close+
            Last,data=veritrain)
summary(lmod1)
```

```
##
## Call:
## lm(formula = VWAP ~ Volume + High + Low + Close + Last, data = veritrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.954  -1.988   0.094   1.873  23.489
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.155e+00  1.334e+00   1.616    0.108
## Volume      -1.745e-07  2.052e-07  -0.851    0.396
## High         3.884e-01  2.234e-02  17.381  < 2e-16 ***
## Low          2.483e-01  2.384e-02  10.415  < 2e-16 ***
## Close        4.876e-01  8.935e-02   5.457 1.25e-07 ***
## Last        -1.270e-01  8.982e-02  -1.414    0.159
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.437 on 230 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 6.618e+05 on 5 and 230 DF,  p-value: < 2.2e-16
```

Kurulan Regresyon Modelinin anlamliligina baktigimizda p value yaklasik=0< 0.05 oldugundan H0 red edilir yani kurulan model anlamlidir deriz.

Simdi train ve test veri seti uzerinde RMSE degerlerini hesaplayalim.

```
rmse <- function(x,y) sqrt(mean((x-y)^2))
rmse(predict(lmod1), veritrain$VWAP)
```

```
## [1] 4.379819
```

Train veri seti uzerinden kurulan regresyon modelinin RMSE degeri 4.379819 dir.

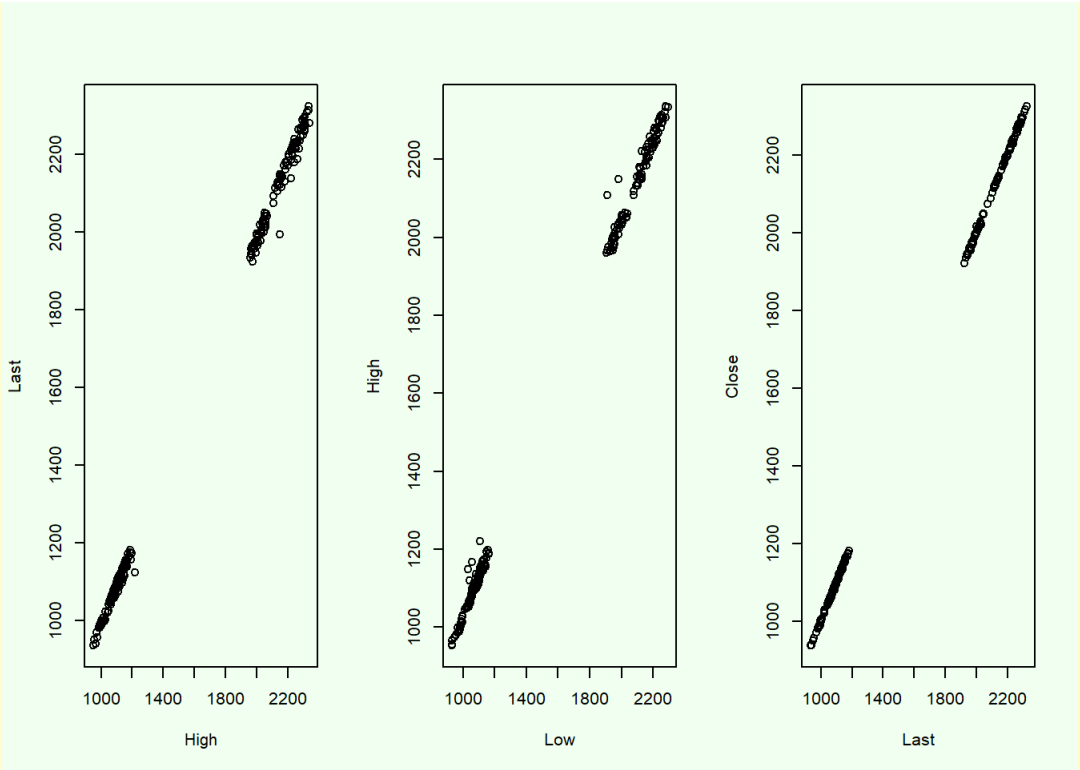
```
rmse(predict(lmod1, veritest), veritest$VWAP)
```

```
## [1] 3.084308
```

Test veri seti uzerinden kurulan regresyon modelinin RMSE degeri 3.084308 dir.

Iki RMSE degeri arasinda fark olmasinin sebebi multicollinearity nin varliginin gostergesidir.

```
par(mfrow=c(1,3),op = par(bg = "honeydew"))
plot(Last~High,veritrain)
plot(High~Low,veritrain)
plot(Close~Last,veritrain)
```



Bazi aciklayici degiskenler arasinda sacinim grafigi cizdirdik. Bunlar arasinda lineer iliski goruluyor. Bu da multicollinearity'nin varliginin bir gostergesidir.

Simdi tum componentleri kullanarak bir Temel Bilesenler Regresyonu kuralim.

```
library(pls)
pcrmodel <- pcr(VWAP~Volume+
                 High+
                Low+
                 Close+
                 Last,data=veritrain,scale=T)

summary(pcrmodel)
```

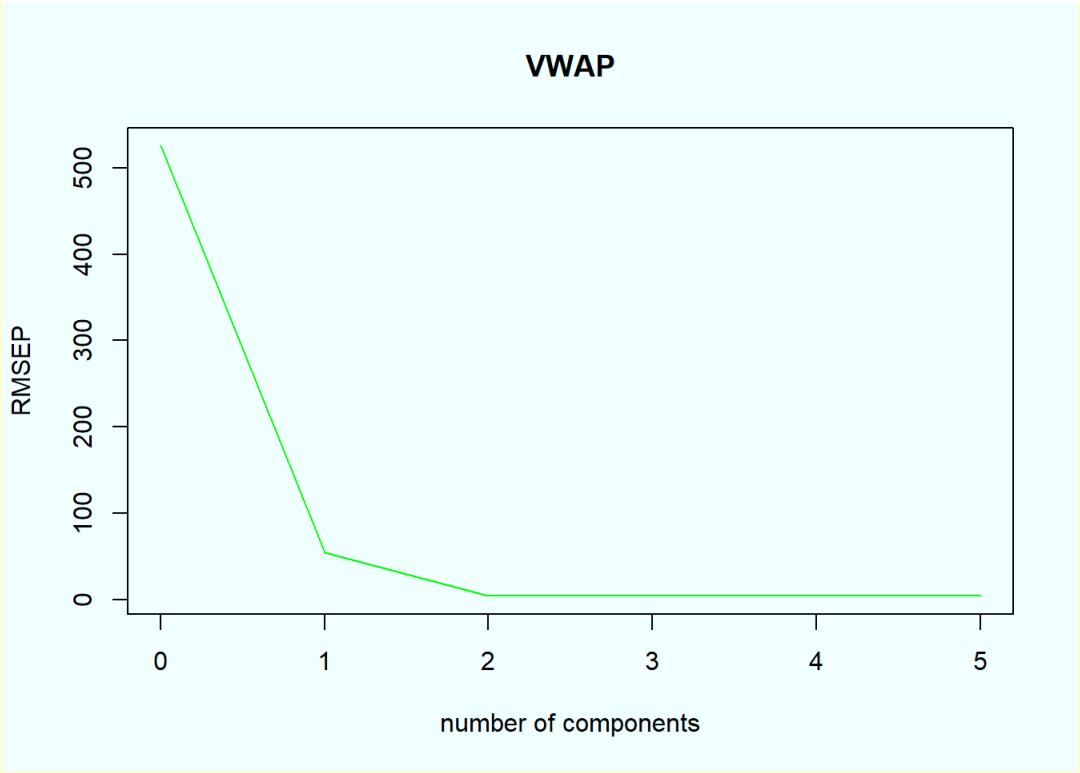
```
## Data:    X dimension: 236 5
##  Y dimension: 236 1
## Fit method: svdpc
## Number of components considered: 5
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps
## X       83.24    99.98    99.99   100.00   100.00
## VWAP    98.94    99.99    99.99    99.99    99.99
```

Ciktinin ilk satiri componentlerin X matrisindeki aciklayicilik oranlarini, ikinci satir ise yanit degiskenindeki aciklayicilik miktarlarini gostermektedir.

Verimizde 5 adet vektor vardi. Besinci component ile yanittaki degiskenligin yaklasik %99.9 aciklanabilmektedir.

Train ne kadar cok componentle calisirsa o kadar iyi sonuc alinir. Cunku multicollinearity nin train uzerinde etkisi yoktur fakat test verisi uzerine etkisi vardir.

```
op = par(bg = "azure")
validationplot(pcrmodel,val.type = "RMSE",col="green")
```



Bu grafik bize her componente karsilik gelen RMSE degerlerini gosterir. Train veri seti uzerinden cizilen grafikte component sayisi arttikca RMSE degerleri dusmektedir.

En buyuk degisim intercepten birinci componente geciste yasanmistir. Ikinci componentten sonra bir degisim olmamistir. Toplamda intercept ile birlikte 6 component var.

Bu grafige iliskin RMSE degerleri;

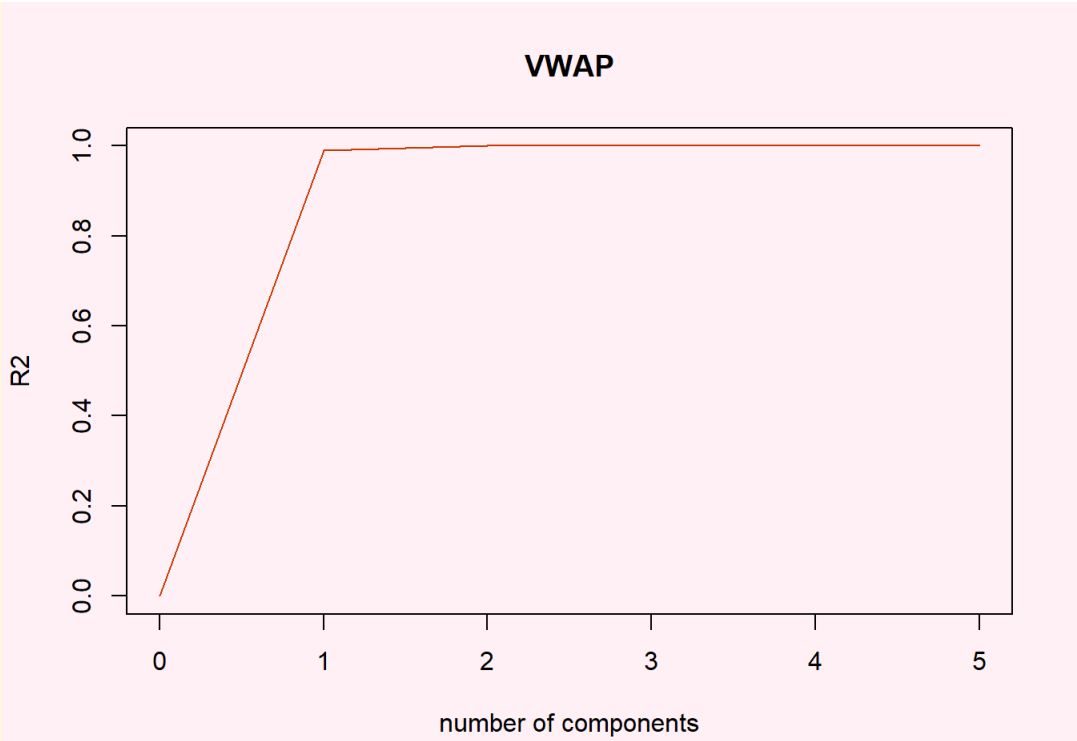
```
pcrmse <- RMSEP(pcrmodel)
pcrmse
```

```
## (Intercept)      1 comps      2 comps      3 comps      4 comps      5 comps
##     525.355       54.197        4.938        4.579        4.492        4.380
```

RMSE degeri component sayisi arttikca azalmaktadir. En dusuk RMSE degeri ilk olarak birinci componentte gorulmektedir.

Simdi RMSE icin yaptiklarimizi R kare icinde yapalim.

```
op = par(bg = "lavenderblush1")
validationplot(pcrmodel,val.type = "R2",col="orangered3")
```



Bu grafik bize her componente karsilik gelen R kare degerlerini gosterir. Train veri seti uzerinden cizilen grafikte component sayisi arttikca R kare degerleri artmaktadir.

En buyuk degisim intercepten birinci componente geciste yasanmis.Ikinci componentten sonra bir degisim olmamistir. Toplamda intercept ile birlikte 6 component var.

Multicollinearity icin train veri setine bakmak uygun olmadigindan test veri setine bakmak daha uygun olacaktir.

Test seti uzerinde component sayilarina gore RMSE degerlerini de RMSEP fonksiyonunu kullanarak bulabiliriz.

```
pcrmse <- RMSEP(pcrmodel,newdata=veritest)
pcrmse
```

```
## (Intercept)      1 comps      2 comps      3 comps      4 comps      5 comps
##     573.657       37.726        3.581        4.152        4.505        3.084
```

Test veri seti uzerinden baktigimizda performans acisindan en kucuk RMSE degeri 3.084 ile besinci componenttedir.

Simdi 5 component ile kurulan modelin katsayilarina bakalim.

```
coef(pcrmodel,ncomp=5)
```

```
## , , 5 comps
##
##               VWAP
## Volume  -0.3619155
## High   206.7179358
## Low    129.5871708
## Close  257.0354530
## Last   -66.9235424
```

Bu cikti bize pcr 5 component icin modelin yanit degiskeni tahminlerini verir.

Simdi bu modeli kullanarak train ve test veri seti uzerinden RMSE degerlerini hesaplayalim.

```
rmse(predict(pcrmodel, ncomp=5), veritrain$VWAP)
```

```
## [1] 4.379819
```

Train veri seti uzerinden RMSE degeri 4.379819 dir.

```
rmse(predict(pcrmodel, veritest, ncomp=5), veritest$VWAP)
```

```
## [1] 3.084308
```

Test veri seti uzerinden RMSE degeri 3.084308 dir.

Componentlerimizin tamamini kullandigimiz icin sonucumuz Ekk modelimizle ayni cikmistir.Optimal component sayisini belirlerken test verisi uzerindeki performansa baktik. Bu is icin Cross Validation Yonteminin kullanilmasi aslinda daha dogru bir yaklasim olacaktir.

CROSS-VALIDATION ILE TEMEL BILESEN ANALIZI

Kac component olmasi gerektigini daha iyi verecek yontemdir. Diger yontemlerde her yapista farkliliklar olmaktadir.

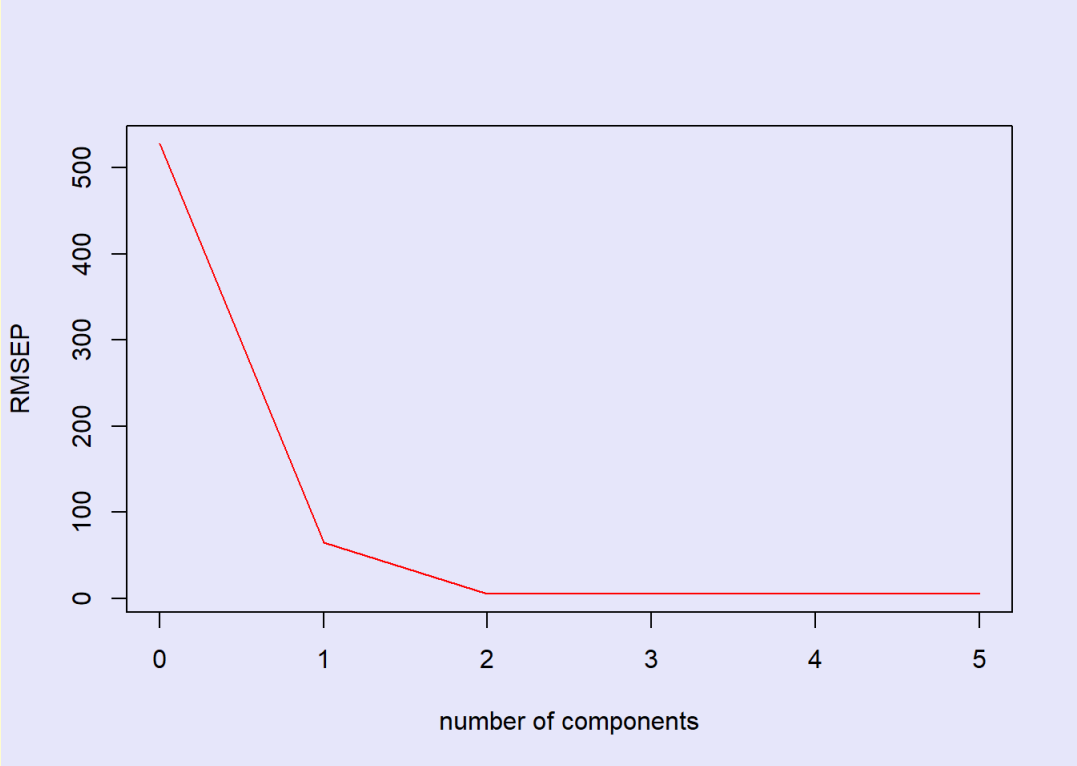
Train veri seti uzerinden Cross Validation hesaplatalim;

```
set.seed(3)
pcrmodel1 <- pcr(VWAP~Volume+
            High+
           Low+
            Close+
            Last,data=veritrain,scale=T,validation="CV")
pcrCV <- RMSEP(pcrmodel1, estimate="CV")
pcrCV
```

```
## (Intercept)       1 comps       2 comps       3 comps       4 comps       5 comps
##     527.591        64.374         5.203         5.158         5.218         5.116
```

RMSE degeri en dusuk besinci componentte (5.116) cikmistir.

```
op = par(bg = "lavender")
plot(pcrCV,main="",col="red")
```



Cross Validation icin componentlere karsilik RMSE degerlerinin grafigi yukarida gosterilmektedir.

Pcrcv icin en kucuk RMSE degerini alan componentin hangisi olduguna bakmak icin;

```
which.min(pcrCV$val)
```

```
## [1] 6
```

Grafikte Cross Validated RMSE degerleri vardir.10 fold Cross Validation ile pcrCV degerlerinin altincisi yani besinci component'e karsilik gelen RMSE degeri minimum cikti. (intercept+5 component)

```
coef(pcrmodel1,ncomp=5)
```

```
## , , 5 comps
##
##               VWAP
## Volume  -0.3619155
## High   206.7179358
## Low    129.5871708
## Close  257.0354530
## Last   -66.9235424
```

Bu cikti bize pcrCV 5 component icin modelin yanit degiskeni tahminlerini verir. Yani tahmin performansimiz besinci componentte en iyi olmaktadir.

Datayi NSTS alarak modelimizi kuralim;

```
model1<-lm(VWAP~Volume+
            High+
           Low+
            Close+
            Last,data=NSTS)
summary(model1)
```

```
##
## Call:
## lm(formula = VWAP ~ Volume + High + Low + Close + Last, data = NSTS)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.7176  -1.8093  -0.0132   1.9931  23.8569
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.983e+00  1.282e+00   1.547   0.1232
## Volume      -1.505e-07  1.993e-07  -0.755   0.4509
## High         3.824e-01  2.165e-02  17.666  < 2e-16 ***
## Low          2.517e-01  2.317e-02  10.861  < 2e-16 ***
## Close        5.419e-01  8.214e-02   6.597 2.62e-10 ***
## Last        -1.785e-01  8.325e-02  -2.144   0.0331 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.374 on 242 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 7.22e+05 on 5 and 242 DF,  p-value: < 2.2e-16
```

Kurulan Regresyon Modelinin anlamliligina baktigimizda p value yaklasik=0< 0.05 oldugundan H0 red edilir yani kurulan model anlamlidir deriz.

```
fit<- fitted(model1)
head(fit,10) #tahmin degerleri
```

```
##        1        2        3        4        5        6        7        8
## 1970.505 2000.941 2000.936 1955.733 1960.802 1972.083 2042.029 2103.662
##        9       10
## 2088.532 2117.493
```

```
resid <-residuals(model1)
head(resid,10) #artiklar
```

```
##          1          2          3          4          5          6          7
##  0.8348789  2.3089608  3.6538460 -0.9128143  1.7875111  0.6971293 -4.3386649
##          8          9         10
## -4.2621433  0.8882247 -6.6132122
```

HATA ILE ILGILI VARSAYIMLAR

NORMALLIK VARSAYIMININ KONTROLU TESTI

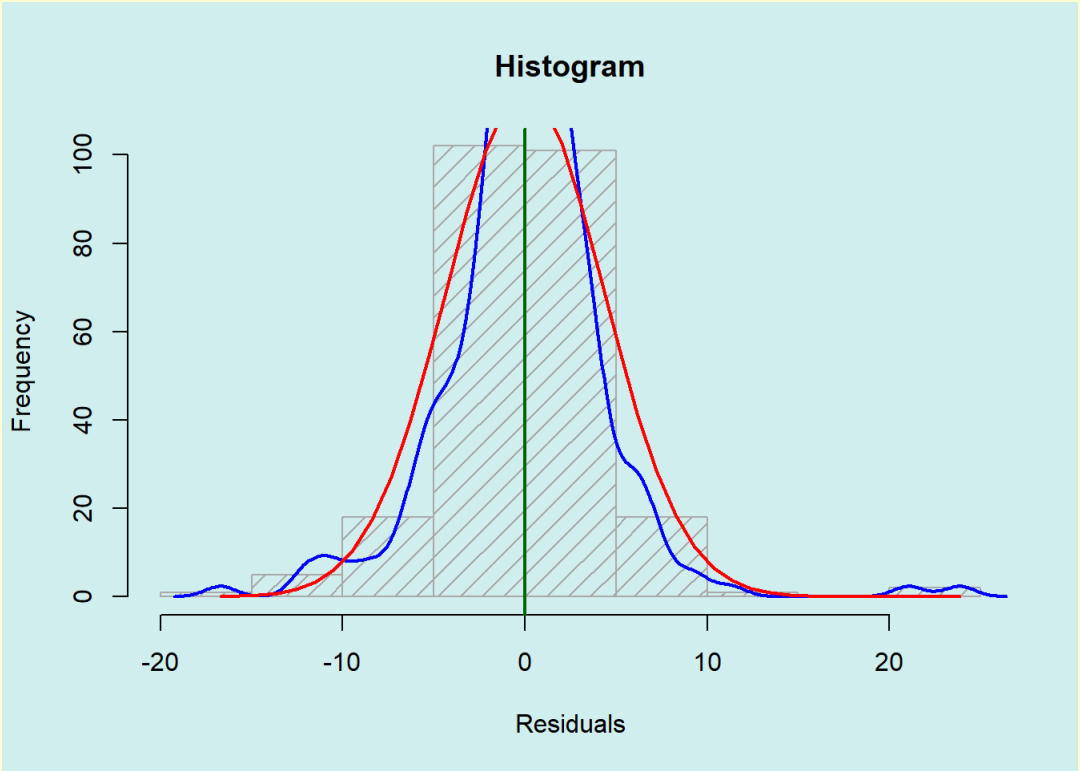
Tum normallik testlerini (Shapiro-Wilk,Kolmogorov-Smirnov,Cramer-von Mises,Anderson-Darling) bir arada gormek icin asagidaki kodu kullanmaliyiz;

```
library(olsrr)
ols_test_normality(model1)
```

```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.915          0.0000
## Kolmogorov-Smirnov        0.0917         0.0309
## Cramer-von Mises         16.8161         0.0000
## Anderson-Darling          4.0377         0.0000
## -----------------------------------------------
```

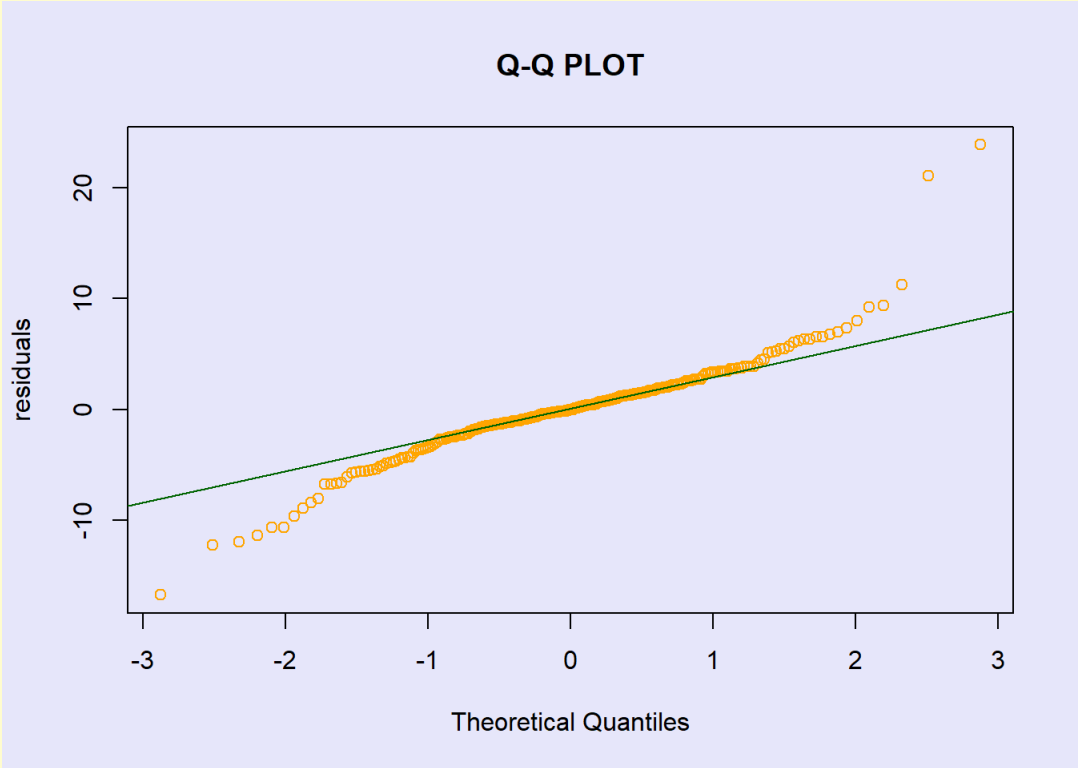
Normallik testlerimizin p-valuelerine baktigimizda 0.05 ten kucuk oldugunu goruyoruz yani artiklarimizin dagilimi normal degildir deriz.

```
op = par(bg = "lightcyan2")
x <-residuals(model1)

#Artiklarin histogrami;
histogram <-hist(x, breaks=10, density=10,col="darkgrey",xlab="Residuals", main="Histogram")
abline(v=mean(x), col="darkgreen", lwd=2)


#Yogunluk egrisi cizme;
multiplier <- histogram$counts / histogram$density
mydensity <- density(x)
mydensity$y <- mydensity$y * multiplier[1]
lines(mydensity,col="blue", lwd=2)

#Normal egrisisin ayni ortalama ve standart sapma ile cizilmesi;
xfit <- seq (min(x), max(x), length=40)
yfit <- dnorm(xfit, mean =mean(x), sd = sd(x))
yfit <- yfit *diff(histogram$mids[1:2]) *length(x)
lines(xfit, yfit, col="red", lwd=2)
```



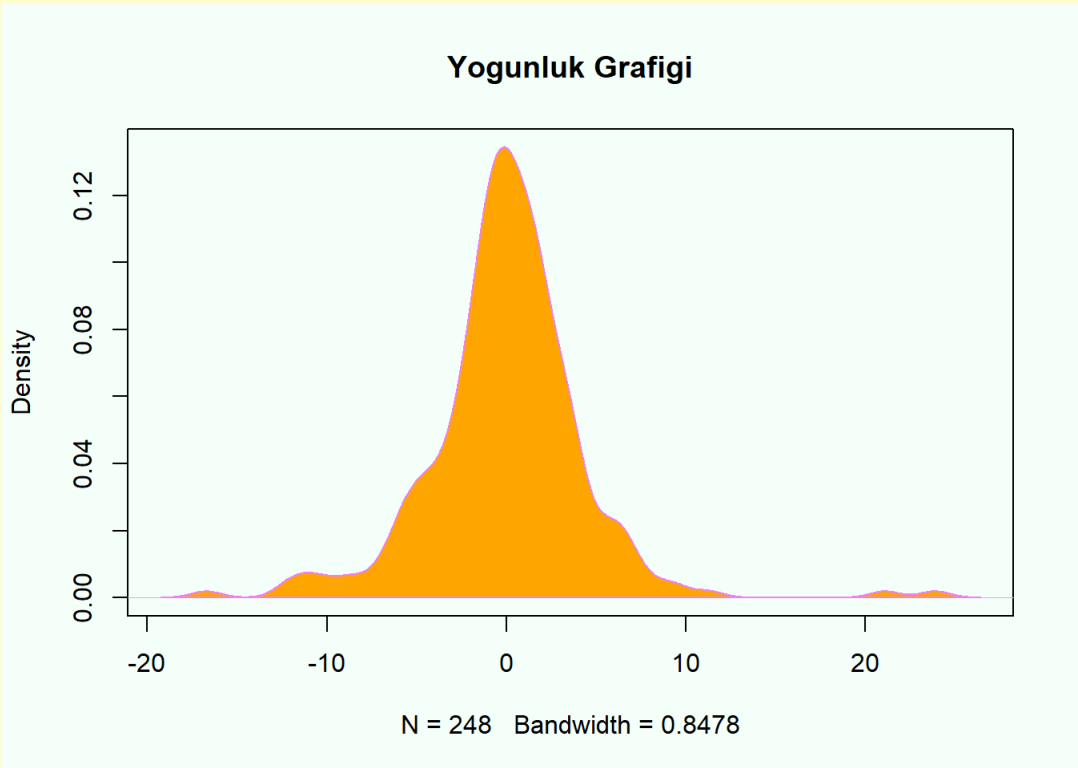
Histogram Grafigimize baktigimizda Kirmizi cizgi bize Normal Dagilim Egrisini verir , Mavi olan cizgi ise sinif orta noktalarindan gecen diyagramdir.Mavi ve Kirmizi cizgilerimiz birbirine benzemedigi icin verinin Normal dagilmadigini soyleyebiliriz.

```
#QQ-plot;
op = par(bg = "lavender")
qqnorm(residuals(model1),ylab="residuals",main="Q-Q PLOT",col="orange")
qqline(residuals(model1),col="darkgreen")
```



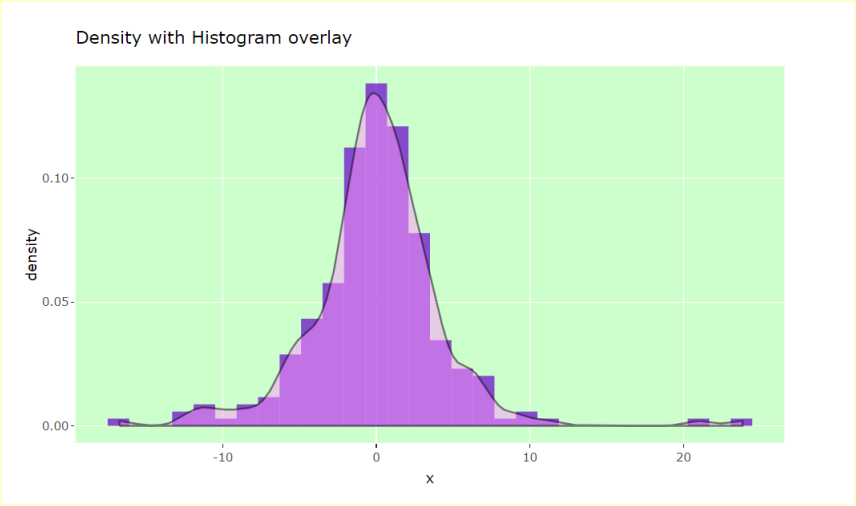
Q-Q Plot Grafigimize baktigimizda verimiz Q-Q Plot cizgisi uzerinde olmadigindan normal dagilim varsayiminin saglanmadigi gorulmektedir.

```
#Density;
op = par(bg = "mintcream")

d <- density(x)
plot(d,main = "Yogunluk Grafigi")
polygon(d, col="orange", border="violet")
```



```
library(plotly)

p <- ggplot(NSTS, aes(x)) +
  geom_histogram(aes(y = ..density..), alpha = 0.7, fill = "#6600CC") +
  geom_density(fill = "#FF99FF", alpha = 0.5) +
  theme(panel.background = element_rect(fill = '#CCFFCC')) +
  ggtitle("Density with Histogram overlay")

fig <- ggplotly(p)
fig
```

```
# Interaktif density plotun gorseli;(Html ciktisini pdf e cevirince gozukmedigi icin ekledik)

knitr::include_graphics("densityplot.png")
```

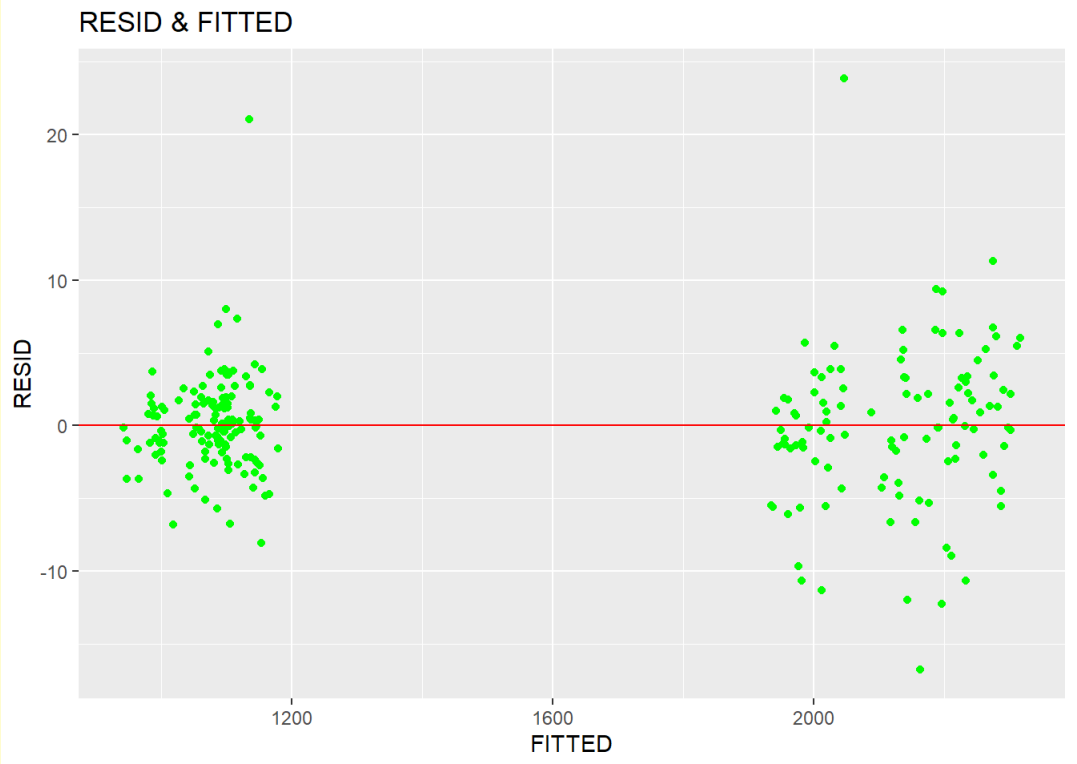


Yogunluk Grafigimize baktigimizda sag kuyruk daha uzun oldugu icin grafigimiz sola carpiktir deriz.

SABIT VARYANSLILIK

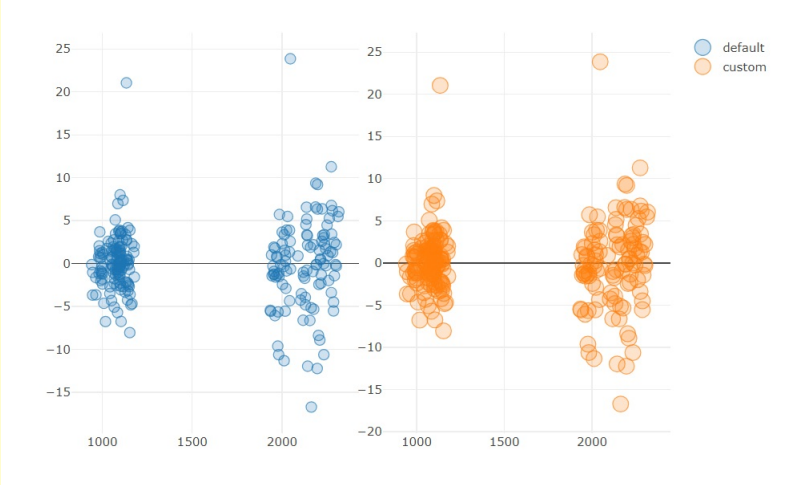
Sabit varyansliligin en kullanisli teshis yontemi artiklara (residuals(lmod)) karsilik tahmin (fitted(lmod)) degerlerinin plotlanmasidir.

```r
library(ggplot2)
ggplot(data=NSTS,mapping=aes(x=fit,y=resid))+
  geom_jitter(color="green")+
  geom_hline(yintercept=0,color="red")+ggtitle("RESID & FITTED ")+xlab(" FITTED ")+ylab("RESID")
```




```r
# Interaktif bir sekilde gorsellestirelim;

p <- plot_ly(veritrain, x = fit, y = resid, alpha = 0.3)
subplot(
  add_markers(p, size = 2, name = "default"),
  add_markers(p, size = 2, sizes = c(1, 248), name = "custom")
)
```

```
#Interaktif plotun gorseli;(Html ciktisini pdf e cevirince gozukmedigi icin ekledik)

knitr::include_graphics("sabitvaryans2.png")
```



Cizdirdigimiz grafigimiz bize duzdun bir sekil vermedigi icin sabit varyansli olup olmadigina emin olamiyoruz.Bu yuzden degisken varyanslilik testlerine basvuruyoruz.

DEGISKEN VARYANSLILIK TESTLERI

BREUSCH-PAGAN TESTI

H0:Heterosce dosticity (Degisken Varyanslilik) problemi yok. H1:Heterosce dosticity (Degisken Varyanslilik) problemi vardir.

```
#install.packages("lmtest")
library(lmtest)
bptest(model1,data=NSTS)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model1
## BP = 107.86, df = 5, p-value < 2.2e-16
```

BREUSCH-PAGAN testimizin sonucuna gore p-value yaklasik=0 <0.05 oldugundan H0 hipotezi reddedilir yani Heteroscedosticity (degisken varyanslilik) problemi vardir.

ILISKILI HATALAR (OTOKORELASYON)

H0: Otokorelasyon yoktur. H1: Otokorelasyon vardir.

```
library(car)
library(lmtest)
dwtest(VWAP~Volume+
            High+
           Low+
            Close+
            Last ,data=NSTS)
```

```
##
##  Durbin-Watson test
##
## data:  VWAP ~ Volume + High + Low + Close + Last
## DW = 1.906, p-value = 0.2024
## alternative hypothesis: true autocorrelation is greater than 0
```

Sonucumuza gore p-value degerimiz 0.2024 cikmistir.P-value degerimiz 0.05'ten buyuk oldugu icin H0 kabul edilir.Bu nedenle Otokorelasyon yoktur deriz.

OLAGAN DISI GOZLEMLER(AYKIRI GOZLEMLERIN

BELIRLENMESI)

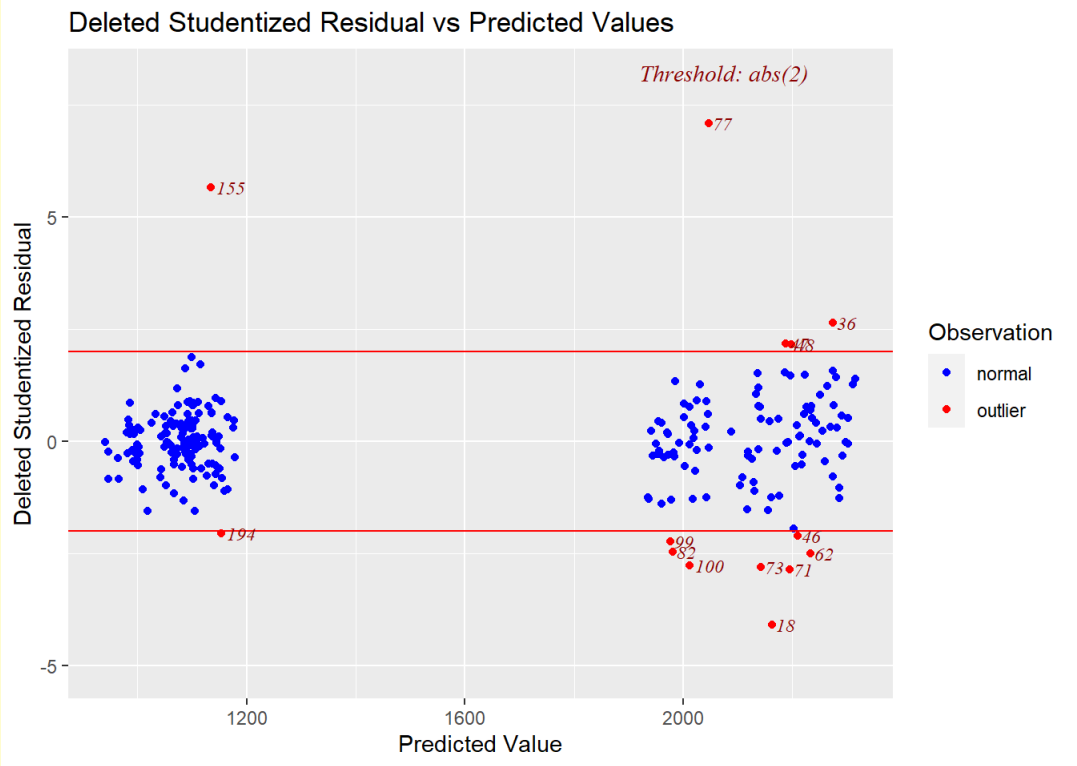
OUTLIER

Model tarafindan iyi tahmin edilemeyen gozlemlere denir.(Hatalari buyuk olan gozlemlere denir.)

Simdi kurulan regresyon modelimizin grafiklerini inceleyelim;

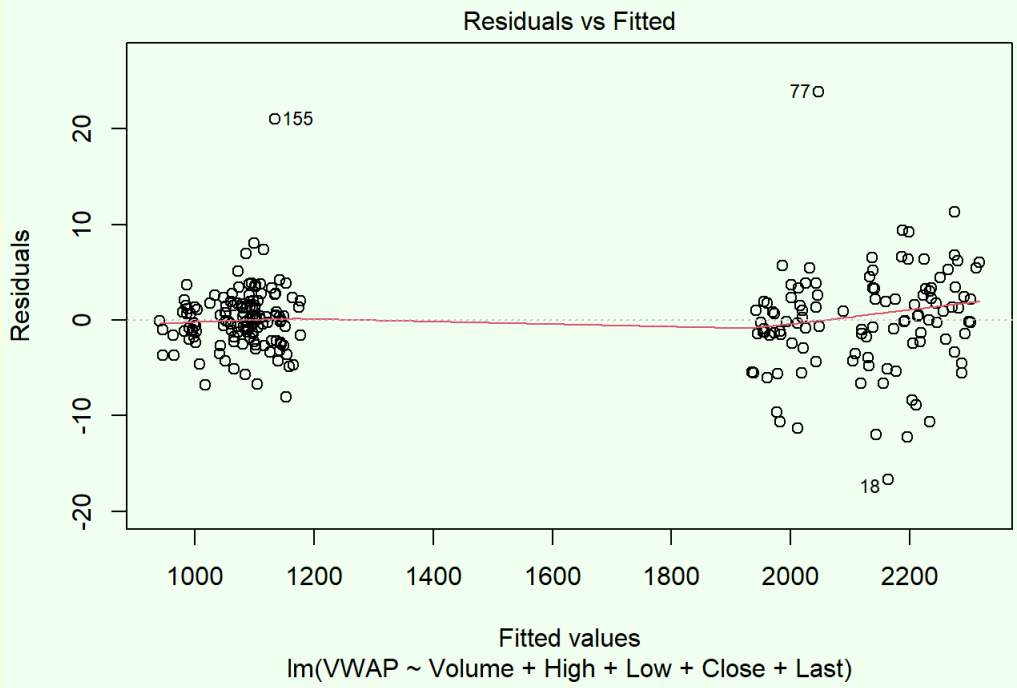
Ilk oncelikle modelimiz icin supheli outlier varsayimlarimiza bakalim;

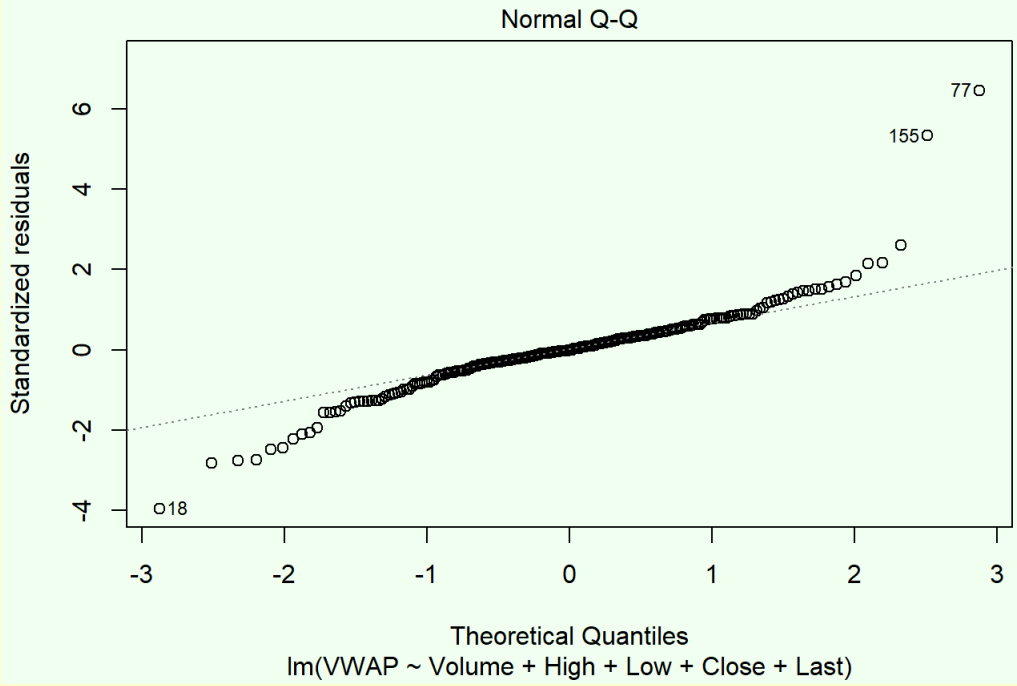
```
ols_plot_resid_stud_fit(model1)
```

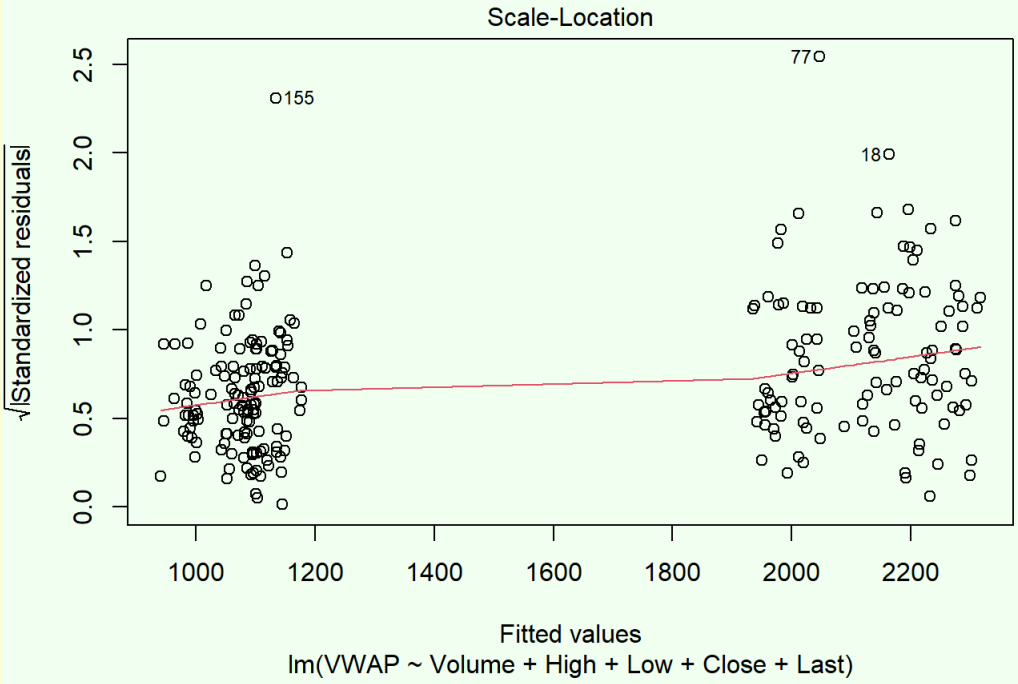


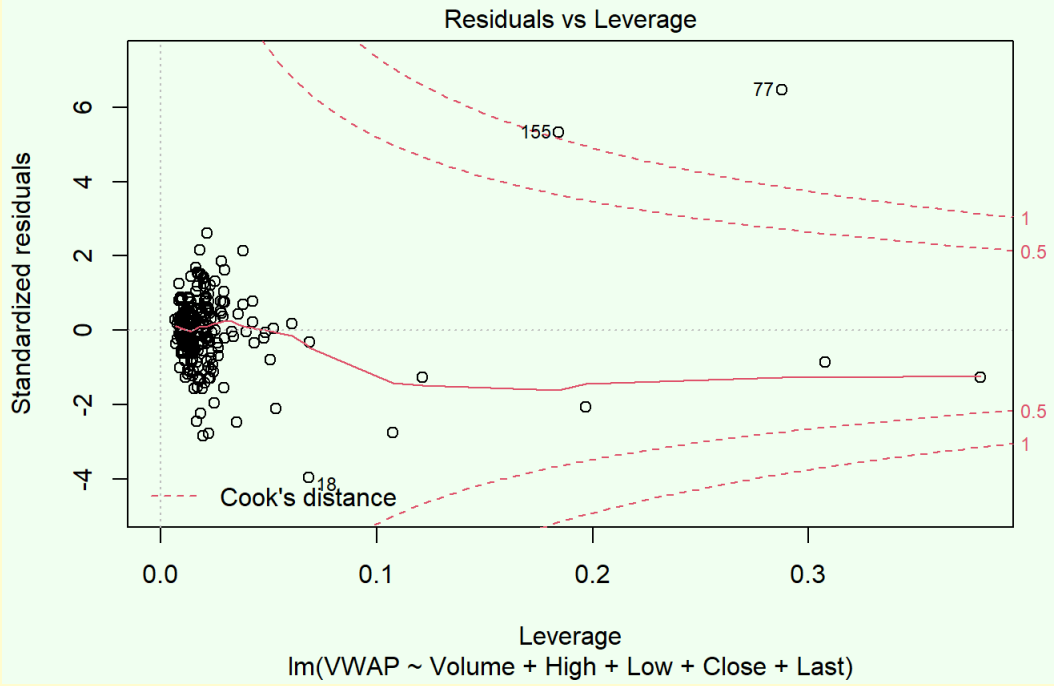
Cizdirdigimiz grafikte kirmizi olan gozlemler supheli outlier degerlerimizdir.Bu grafik sadece varsayim yapmaktadir bu nedenle oncellikle library(faraway) ile plot cizdirerek modelimizde outlier suphesi olan gozlemlerine bakalim;

```
library(faraway)
op = par(bg = "honeydew")
plot(model1)
```


Residuals vs Fitted
Residuals
Fitted values
lm(VWAP ~ Volume + High + Low + Close + Last)
155
77
18
20
10
0
-10
-20
1000
1200
1400
1600
1800
2000
2200


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(VWAP ~ Volume + High + Low + Close + Last)
77
155
18
6
4
2
0
-2
-4
-3
-2
-1
0
1
2
3





1.Grafik icin ; Artiklara karsi cizdirdigimiz grafige baktigimizda varyanslarin homojen olmadigi gorulmektedir.Outlier suphesi olan gozlemlerimiz 18,77 ve 155. gozlem cikmistir.

2.Grafik icin; Normal Q-Q Plot Grafigimize baktigimizda verimiz Q-Q Plot cizgisi uzerinde olmadigindan normal dagilim varsayiminin saglanmadigi gorulmektedir.Outlier suphesi olan gozlemlerimiz 18,77 ve 155. gozlem cikmistir.

3.Grafik icin; Standartlastirilmis artiklarin karekokune karsi cizdirdigimiz grafige baktigimizda varyanslarin homojen olmadigi gorulmektedir.Outlier suphesi olan gozlemlerimiz 18,77 ve 155. gozlem cikmistir.

4.Grafik icin; Cook’s Distance a gore grafikte cikan degerler yani 18,77 ve 155. gozlemlere outlier suphesiyle yaklasilir.

Bu grafiklere baktigimizda 18,77 ve 155. gozlemleri muhtemelen sorunlu olarak tanimlayabiliriz.

Hangi durumlari temsil ettigini gormek icin bu gozlemlere bakmaliyiz;

```
NSTS[c(8, 77,155), ]
```

```
##           Date Symbol Series Prev.Close   Open    High     Low    Last   Close
## 8   2015-01-12   INFY     EQ    2074.45 2092.0 2119.20 2075.00 2112.95 2115.95
## 77  2015-04-24   INFY     EQ    2122.00 2135.9 2150.20 1980.00 1993.00 1995.20
## 155 2015-08-13   INFY     EQ    1144.40 1147.0 1166.85 1055.55 1155.35 1158.00
##        VWAP   Volume     Turnover Trades Deliverable.Volume X.Deliverble
## 8   2099.40  3189722 6.696516e+14 107209            1818800       0.5702
## 77  2070.28 10891363 2.254814e+15 317809            4101422       0.3766
## 155 1155.49  4893980 5.654928e+14 128074            3469113       0.7089
```

Modelimiz icin standartlastirilmis artiklarimiza bakacak olursak;

```
stud <- rstudent(model1)
head(stud,10)
```

```
##          1          2          3          4          5          6          7
##  0.1919764  0.5345281  0.8386665 -0.2134322  0.4107688  0.1606421 -1.2606563
##          8          9         10
## -0.9852964  0.2042155 -1.5290932
```

Standartlastirilmis artiklarimizin mutlakca en buyugune bakmaliyiz.

```
stud[which.max(abs(stud))]
```

```
##       77
## 7.088158
```

En buyuk standartlastirilmis artik (rstudent) degerim 77. gozlem olup degeri mutlakca 7.088158 dir. Fakat aykiri gozlem midir ?

Benferonni duzleltmesi yaparsak; Simdi cut point belirlemeliyiz.

cut point degeri alfa/2n dir. rstudentler (n-p-1) serbestlik dereceli t-dagilimina sahiptir. (p:degisken sayisi +1 , n : gozlem sayisi)

```
qt(0.05/(length(stud)*2) , (length(NSTS$VWAP)-6-1)) #(alfa/2n),(n-p-1)
```

```
## [1] -3.774908
```

Burada en buyuk standartlastirilmis mutlak artik degerini 7.088158 olarak bulmustuk. (|7.088158| > |-3.774908|) oldugundan 77 inci gozlem bizim icin outlierdir.

Bunu hazir kod ile yaptigimizda;

```
library(car)
outlierTest(model1)
```

```
##      rstudent unadjusted p-value Bonferroni p
## 77   7.088158         1.4939e-11   3.7050e-09
## 155  5.658987         4.3065e-08   1.0680e-05
## 18  -4.085940         5.9806e-05   1.4832e-02
```

En buyuk aykiri deger 77. gozleme ait olmakla beraber diger outlier degerler ise 155,18 gozlemlere aittir. Veri setimizde aykiri gozlemler mevcuttur.

Varsayimlar gerceklesmediginde (hatalar normal dagilmadiginda)ve aykiri gozlemler oldugunda ROBUST REGRESYON yontemi kullanilabilir.

ROBUST REGRESYON

Tum regresyon varsayimlari gecerli oldugunda , dogrusal regresyon icin normal EKK tahminleri en uygunudur.Bu varsayimlardan bazilari gecersiz oldugunda , EKK regresyonu kotu performans gosterebilir.

Aykiri gozlemler regresyon dogrusunu kendine gore kendine dogru ceker.Bu sekilde parametre tahminlerini olmasi gerektigi yerden cok uzaga tasiyabilir.Model uzerinde etkileri diger gozlemlerden daha fazla olur.Bu durum bizim model tahmin performansimizi dusurur.Bu durumda robust regresyon kullanilir.

Bu gozlemlerin model uzerinde etkisini dusunerek agirliklandirma yapiyoruz.

Birden cok aykiri gozlem varsa modele bundan etkilenebiliyor(maskeleme-swamping).Bu sekilde kurulan modelde yanlis olmus oluyor ve bu modelin artiklari uzerinden yorum yapmak da cok dogru olmuyor.

Aykiri gozlem calismasi yapilacaksa Robust Regresyon Modeli kurup bu regresyon modelinin artiklari uzerinden konusmak daha dogru olacaktir.

Robust Regresyon iteratif yeniden agirlikli En Kucuk Kareler (IRLS) ile yapilir.Robust Regresyon calistirma komutu MASS paketinde rlm'dir.IRLS icin kullanilabilecek cesitli agirlik fonksiyonlari vardir.

Siradan EKK regresyonu ve robust regresyonun sonuclarini karsilastirirken sonuclar cok farkliysa Robust Regresyondan gelen sonuclar kullanilir.Buyuk farkliliklar, model parametrelerinin aykiri degerlerden buyuk oranda etkilendigini gostermektedir.Farkli agirliklandirmalarin avantajlari ve dezavantajlari vardir.Huber agirliklari siddetli aykiri degerlerde zorluklar yasayabilir ve bisquare agirliklar yakinsamada zorluk yasayabilir veya birden fazla cozum verebilir.

Robust Regresyon modelimizi kuralim.

```
library(MASS)
hubermod <- rlm(VWAP~Volume+
            High+
           Low+
            Close+
            Last  ,data=NSTS)
summary(hubermod)
```

```
##
## Call: rlm(formula = VWAP ~ Volume + High + Low + Close + Last, data = NSTS)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8433  -1.9768  -0.1577   1.9707  30.9003
##
## Coefficients:
##             Value   Std. Error t value
## (Intercept)  1.2445  0.9934     1.2527
## Volume       0.0000  0.0000    -0.1708
## High         0.3197  0.0168    19.0533
## Low          0.2953  0.0180    16.4445
## Close        0.5346  0.0637     8.3969
## Last        -0.1507  0.0645    -2.3361
##
## Residual standard error: 2.932 on 242 degrees of freedom
```

Huber agirliklari kullanarak kurdugumuz Robust Regresyon Modelimizin Residual standard error u 2.932 cikmistir. EKK modelimizde Residual standard error u 4.374 cikmisti.

Iki Regresyon Modelinin katsayilarini yan yana gosterelim;

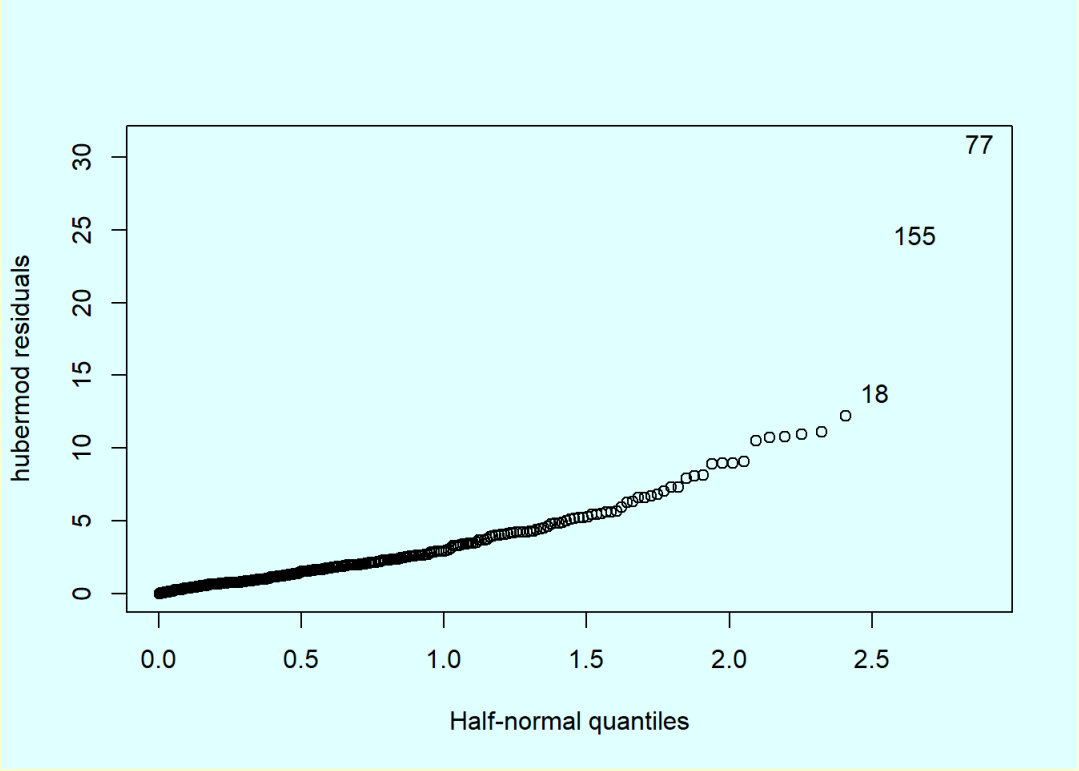
```
cbind(coef(model1),coef(hubermod))
```

```
##                       [,1]          [,2]
## (Intercept)  1.982865e+00  1.244466e+00
## Volume      -1.504960e-07 -2.638095e-08
## High         3.824208e-01  3.196917e-01
## Low          2.516521e-01  2.953356e-01
## Close        5.419047e-01  5.346286e-01
## Last        -1.784653e-01 -1.507446e-01
```

Iki modelin ciktilari arasinda gozle gorulur degisimler vardir.

Simdi Robust Regresyon icin standart artiklari inceleyelim;

```
op = par(bg = "lightcyan1")
halfnorm(residuals(hubermod),3,ylab = "hubermod residuals")
```



Robust resgresyonun ham artiklari 18,155 ve 77. gozlemler olarak cikmistir.

```
stud <- rstudent(hubermod) #robust regresyonunun standart artiklari
stud[which.max(abs(stud))]
```

```
##       77
## 7.978232
```

En buyuk standartlastirilmis artik (rstudent) degerim 77. gozlem olup degeri mutlakca 7.088158 dir.Fakat aykiri gozlem midir ?

Bonferonni duzeltmesi yaparsak;

```
#alfa 0.05
qt(.05/(length(stud)*2),length(NSTS$VWAP)-6-1) #p=bagimsiz degisken sayisi+1
```

```
## [1] -3.774908
```

Burada en buyuk standartlastirilmis mutlak artik degerini 7.088158 olarak bulmustuk. (|7.088158| > |-3.774908|) oldugundan 77 inci gozlem bizim icin outlierdir.

Diger outlier degerler;

```
outlierTest(hubermod)
```

```
##     rstudent unadjusted p-value Bonferroni p
## 77  7.978232          6.006e-14   1.4895e-11
## 155 5.999103          7.222e-09   1.7911e-06
```

En buyuk aykiri deger 77. gozleme ait olmakla beraber diger outlier deger ise 155. gozleme aittir. 18. gozlemimiz outlier olmaktan cikmistir Swamping olmustur. Swamping: Outlier yuzunden aslinda outlier olmayan bir gozlemin outliermis gibi gozukmesidir.

Simdi de Bisquare agirliklandirmasini kullanarak Regresyon Modelimizi kuralim;

```
bisquaremod <- rlm(VWAP~Volume+
            High+
           Low+
            Close+
            Last ,data=NSTS,psi=psi.bisquare)
summary(bisquaremod)
```

```
##
## Call: rlm(formula = VWAP ~ Volume + High + Low + Close + Last, data = NSTS,
##     psi = psi.bisquare)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.8321  -2.0340  -0.2203   2.0211  33.0963
##
## Coefficients:
##             Value   Std. Error t value
## (Intercept)  0.9308  0.9735     0.9561
## Volume       0.0000  0.0000     0.2861
## High         0.2988  0.0164    18.1711
## Low          0.3092  0.0176    17.5705
## Close        0.5274  0.0624     8.4533
## Last        -0.1361  0.0632    -2.1521
##
## Residual standard error: 3 on 242 degrees of freedom
```

Bisquare agirliklari kullanarak kurdugumuz Robust Regresyon Modelimizin Residual standard error u 3 cikmistir. Huber agirliklari kullanarak kurdugumuz Robust Regresyon Modelimizin Residual standard error u 2.932 cikmistir. EKK modelimizde Residual standard error u 4.374 cikmistir.

Huber agirliklari kullanarak kurdugumuz Robust Regresyon Modelimizin Residual standard erroru daha iyi cikmistir.

Simdi tekrardan agirliklara bakalim;

```
biweights <- data.frame(state= NSTS$Date, resid = bisquaremod$resid, weight = bisquaremod$w)
biweights2 <- biweights[order(bisquaremod$w), ]
biweights2[1:10, ]
```

```
##          state      resid     weight
## 77  2015-04-24  33.096310 0.00000000
## 155 2015-08-13  25.793464 0.00000000
## 18  2015-01-27 -12.832060 0.02769650
## 36  2015-02-23  12.635239 0.03687967
## 48  2015-03-11  11.636897 0.09904561
## 82  2015-05-04 -10.747578 0.17252706
## 71  2015-04-16 -10.701817 0.17663408
## 73  2015-04-20 -10.330710 0.21137892
## 100 2015-05-28 -10.105700 0.23333548
## 162 2015-08-24   9.199964 0.32676852
```

Ilk sutun Robust Regresyon Modelinin artiklarini gosterir. Resid ler ne kadar buyukse ona karsilik gelen weight degeri de o kadar kucuk olmaktadir.

Iki modelin residual standart error'lerine bakildigi zaman Huber yontemi daha kucuk residual standart error (2.932) degerine sahiptir.