Robust Regresyon Uygulama-2

ELİF EKMEKCİ

2023-06-03

Kullanacağımız paketleri yükleyelim

```
library(carData)
library(car)
library(faraway)
library(MASS)
```

```
summary(Prestige)
```

```
##    education          income          women           prestige    
##  Min.   : 6.380   Min.   :  611   Min.   : 0.000   Min.   :14.80  
##  1st Qu.: 8.445   1st Qu.: 4106   1st Qu.: 3.592   1st Qu.:35.23  
##  Median :10.540   Median : 5930   Median :13.600   Median :43.60  
##  Mean   :10.738   Mean   : 6798   Mean   :28.979   Mean   :46.83  
##  3rd Qu.:12.648   3rd Qu.: 8187   3rd Qu.:52.203   3rd Qu.:59.27  
##  Max.   :15.970   Max.   :25879   Max.   :97.510   Max.   :87.20  
##      census       type   
##  Min.   :1113   bc  :44  
##  1st Qu.:3120   prof:31  
##  Median :5135   wc  :23  
##  Mean   :5402   NA's: 4  
##  3rd Qu.:8312            
##  Max.   :9517            
```

Verimizi incelediğimizde eksik gözlemler olduğunu görüyoruz. Çalışmamızın ilerleyen bölümlerinde bununla ilgili bir düzeltme yapmamız gerekiyor.

```
mod <-lm(prestige~.,data=Prestige)
summary(mod)
```

```
## 
## Call:
## lm(formula = prestige ~ ., data = Prestige)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -12.9863  -4.9813   0.6983   4.8690  19.2402 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.213e+01  8.018e+00  -1.513  0.13380
## education    3.933e+00  6.535e-01   6.019 3.64e-08 ***
## income       9.946e-04  2.601e-04   3.824  0.00024 ***
## women        1.310e-02  3.019e-02   0.434  0.66524
## census       1.156e-03  6.183e-04   1.870  0.06471 .
## typeprof     1.077e+01  4.676e+00   2.303  0.02354 *
## typewc       2.877e-01  3.139e+00   0.092  0.92718
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.037 on 91 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.841,  Adjusted R-squared:  0.8306
## F-statistic: 80.25 on 6 and 91 DF,  p-value: < 2.2e-16
```

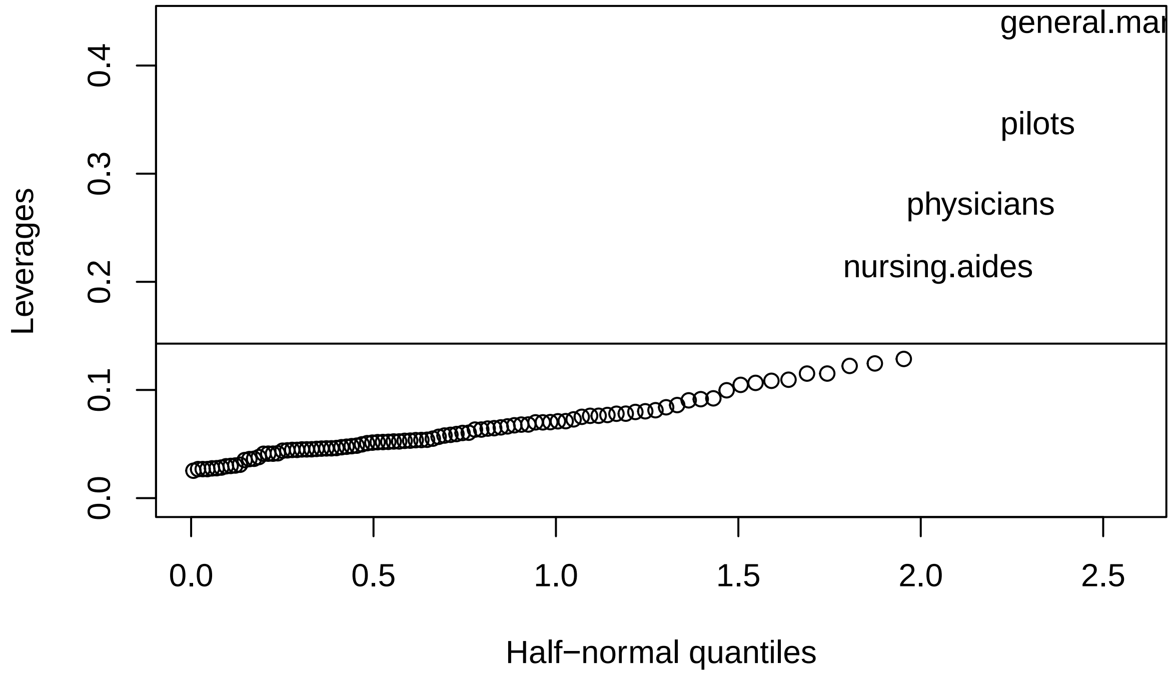
Birinci aşamada leverage point varlığını araştıralım. İlk olarak önerilen yönteme göre bakalım.

```
Prestige1<-Prestige[which(is.na(Prestige$type)=="FALSE"),]
# eksik gozlemler icin bu kodu yazdik
# missing value (NA) degerlerini cikarttik
cutpoint<-2*sum(hatvalues(mod))/nrow(Prestige1)
rownames(Prestige1)[which(hatvalues(mod)>cutpoint)]
```

```
## [1] "general.managers" "physicians"       "nursing.aides"    "pilots"
```

Şimdi de half normal plot üzerinden bakalım

```
jobs<-rownames(Prestige1)
halfnorm(hatvalues(mod),labs=jobs,ylab="Leverages",4)
abline(h=cutpoint)
```



Outliers

```
## Benferroni Correction ile
rstud<-rstudent(mod)
cutpoint<-qt(0.05/(2*nrow(Prestige1)),nrow(Prestige1)-sum(hatvalues(mod))-1)
max(abs(rstud))
```

```
## [1] 2.970091
```

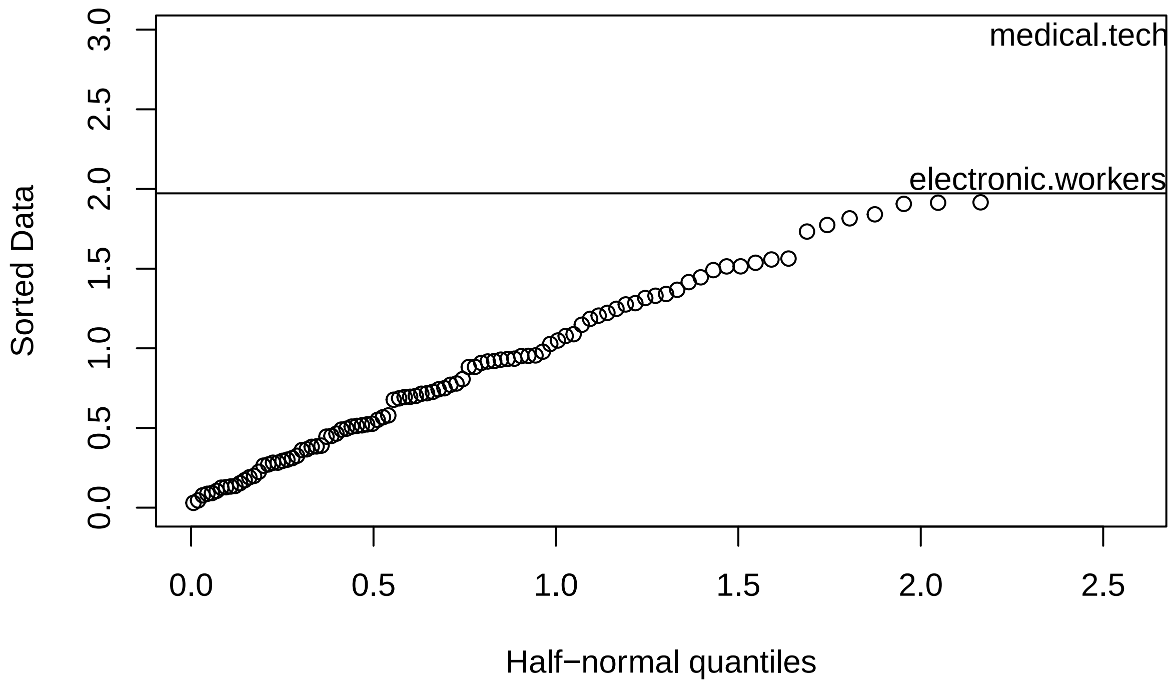
```
outlierTest(mod) # ile de yapabiliriz
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##                     rstudent unadjusted p-value Bonferroni p
## medical.technicians 2.970091          0.0038164        0.374
```

```
## Benferroni correction yapmadan
cutpoint2<-qt(0.05/2,nrow(Prestige1)-sum(hatvalues(mod)-1))
rownames(Prestige1)[which(rstud>abs(cutpoint2))]
```

```
## [1] "medical.technicians" "electronic.workers"
```

```
halfnorm(rstud,labs=rownames(Prestige1),2)
abline(h=abs(cutpoint2))
abline(h=cutpoint)
```



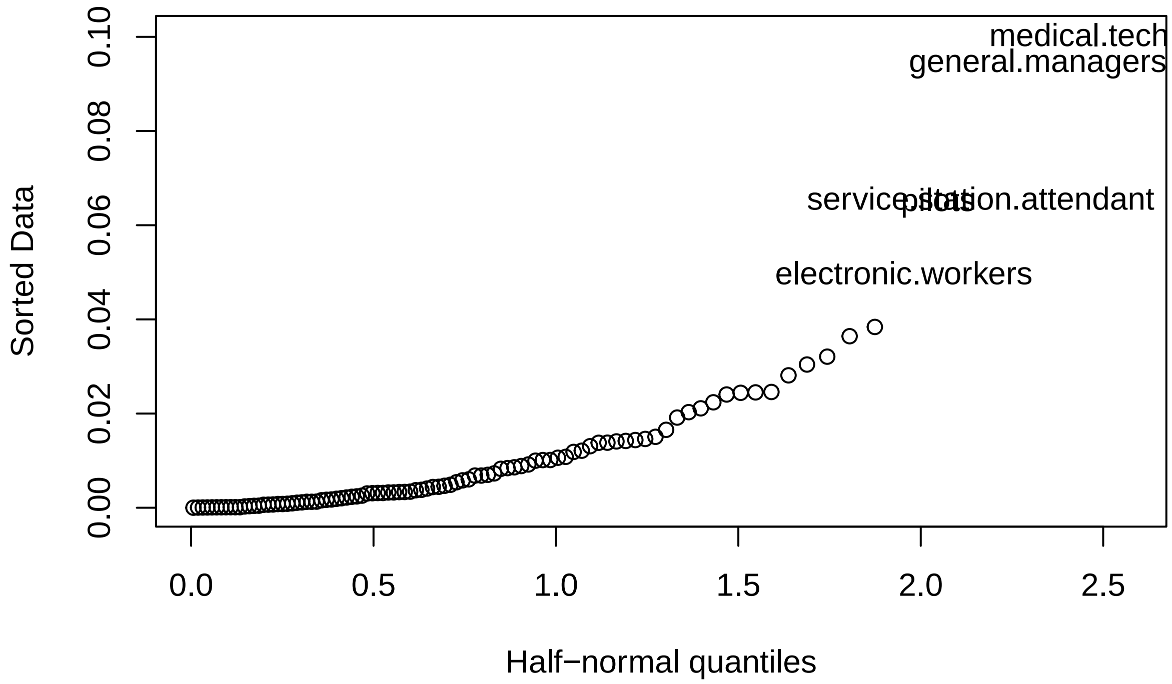
```
outlierTest(mod)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##                        rstudent unadjusted p-value Bonferroni p
## medical.technicians 2.970091          0.0038164        0.374
```

Etkili Gözlemler

Cook Distance

```
p<-sum(hatvalues(mod))
n<-nrow(Prestige1)
jobs<-row.names(Prestige1)
cook<-cooks.distance(mod)
cutpoint<-qf(0.5,p,n-p)
halfnorm(cook,labs=jobs,5)
abline(h=cutpoint)
```



DFBETA

```
dfbeta<-dfbeta(mod)
cut<-2/sqrt(n)
which(abs(dfbeta[,2])>cut)
```

```
##    general.managers medical.technicians 
##                   2                  31
```

Buraya kadar outlier, leverage ve etkili gözlem olup olmadığını kontrol ettik. Şimdi veriyi test ve train olarak ayırıp robust regresyon modeli uygulayalım

```
set.seed(124)
n<-nrow(Prestige1)
index<-sample(1:n,round(0.8*n))
```

```
training<-Prestige1[index,]
test<-Prestige1[-index,]
lmod<-lm(prestige~.,data=training)
```

```
library(caret)
ctrl<-trainControl(method='cv', number=10)
```

```
X<-model.matrix(lmod)[,-1]
y<-training$prestige
cv.lm<-train(X, y,method='rlm',trControl=ctrl)
print(cv.lm)
```

```
## Robust Linear Model
##
## 78 samples
##  6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 72, 70, 70, 70, 70, 70, ...
## Resampling results across tuning parameters:
##
##   intercept  psi           RMSE      Rsquared   MAE
##   FALSE      psi.huber     7.039363  0.8646686  5.715574
##   FALSE      psi.hampel    7.045841  0.8646356  5.759307
##   FALSE      psi.bisquare  7.046819  0.8640915  5.731056
##    TRUE      psi.huber     6.999862  0.8626104  5.784885
##    TRUE      psi.hampel    7.022545  0.8616450  5.873559
##    TRUE      psi.bisquare  7.030396  0.8617377  5.821449
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were intercept = TRUE and psi = psi.huber.
```

En uygun model intercept = TRUE and psi= psi.huber. olarak bulundu Bu yüzden ilk olarak interceptli Huber modele bakalım

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
model<-rlm(prestige~.,data=training)
fits<-predict(model,test)
accuracy(test$prestige,fits)
```

```
##                 ME     RMSE      MAE      MPE     MAPE
## Test set 2.826103 8.516491 7.202853 5.943733 16.24308
```

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

Test seti üzerindeki RMSE değerini hesaplayalım

```
rmse(test$prestige,fits,nrow(test))
```

```
## [1] 8.516491
```

Test seti üzerindeki rˆ2 değerini hesaplayalım

```
rsquare(test$prestige,fits)
```

```
## [1] 0.7220612
```

Şimdi interceptsiz modele bakalım

```
nointerceptmodel<-rlm(prestige~0+.,data=training)
```

nointerceptmodel için outlier test yapalım

```
outlierTest(nointerceptmodel)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##                             rstudent unadjusted p-value Bonferroni p
## service.station.attendant -2.360202           0.021057           NA
```

service.station.attendant değişkeni outlier olarak bulundu.