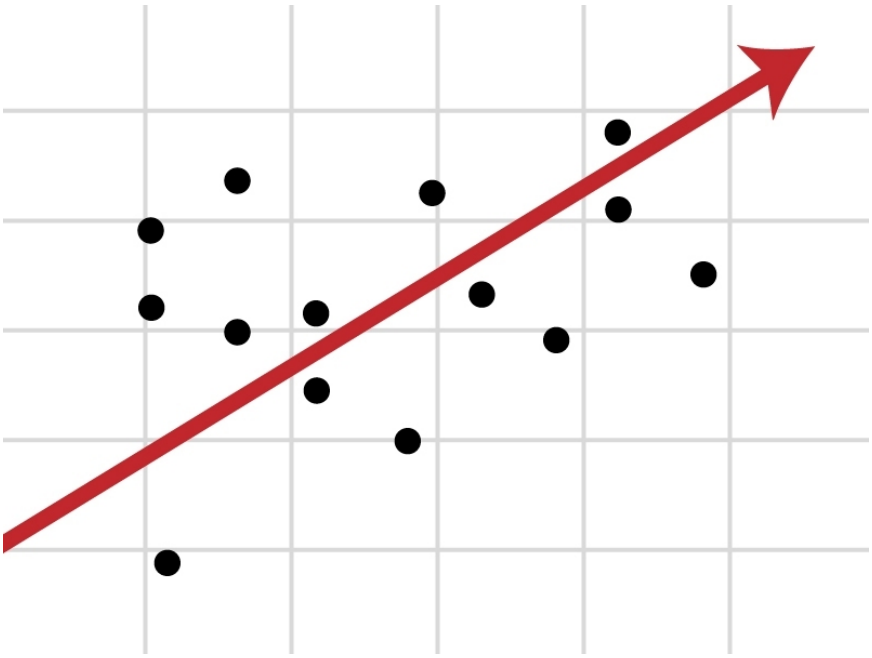


R ile Veri Ön İşleme ve Doğrusal Regresyon

Muhammed Fatih TÜZEN

13 01 2022

İÇİNDEKİLER

1 Veri Ön İşleme **3**

1.1 Eksik Veriler 3

1.1.1 Eksik Verileri Silme 4

1.1.2 İmputasyon 6

1.2 Aykırı Değer Analizi 11

1.2.1 Minumum ve Maximum 11

1.2.2 Histogram 11

1.2.3 Boxplot 12

1.2.4 Yüzdelikler (Percentiles) 14

1.2.5 Z-Skor Yöntemi 15

1.3 Veri Normalleştirme 17

2 Doğrusal Regresyon **21**

1 Veri Ön İşleme

Veri ön işleme; istatistiksel modeller kurulmadan önce veri seti üzerinde yapılan bir takım düzeltme, eksik veriyi tamamlama, tekrarlanan verileri kaldırma, dönüştürme, bütünleştirme, temizleme, normalleştirme, boyut indirgeme vb. işlemlerdir. Bu aşamada ister istemez veri üzerinde bilgi keşfi yapılmış olur. Veri önişleme istatistiksel bir modelleme sürecinin büyük kısmını oluşturmaktadır. Kesin bir rakam olmamakla birlikte modelleme sürecinin yarısından fazlasının bu aşamada harcandığını ifade edebiliriz.

Bu dokümanda veri ön işleme konularında eksik veriler (missing values), aykırı değerler (outliers) ve veri normalleştirme işlemleri R uygulamları ile anlatılacaktır.

1.1 Eksik Veriler

Eksik veriler (kayıp gözlem), veri toplamada kaçınılmaz bir durumdur ve üzerinde dikkatle durulmalıdır. Sistematik bir kayıp gözlem durumu yoksa ortada ciddi bir sorun yoktur. Ama rastgele olmayan bir hata varsa tüm kitleye dair yanlılık olacağı için bu durum göz ardı edilemez.

```
df <- data.frame(weight=c(rnorm(15,70,10),rep(NA,5)),
height=c(rnorm(17,165,20),rep(NA,3)))

set.seed(12345)
rows <- sample(nrow(df))
df2 <- df[rows, ]

# eksik verilerin sorgulanması

is.na(df2) # sorgulanma
```

```
##    weight height
## 14  FALSE  FALSE
## 19   TRUE   TRUE
## 16   TRUE  FALSE
## 11  FALSE  FALSE
## 18   TRUE   TRUE
## 8   FALSE  FALSE
## 2   FALSE  FALSE
## 6   FALSE  FALSE
## 17   TRUE  FALSE
## 13  FALSE  FALSE
## 7   FALSE  FALSE
## 1   FALSE  FALSE
## 15  FALSE  FALSE
```

```
## 10  FALSE  FALSE
## 12  FALSE  FALSE
## 9   FALSE  FALSE
## 4   FALSE  FALSE
## 20   TRUE   TRUE
## 3   FALSE  FALSE
## 5   FALSE  FALSE
```

```
which(is.na(df2)) #konum
```

```
## [1]  2  3  5  9 18 22 25 38
```

```
sum(is.na(df2)) # toplam eksik veri sayısı
```

```
## [1] 8
```

```
colSums(is.na(df2)) # değişken düzeyinde eksik veri sayısı
```

```
## weight height
##      5      3
```

```
df2[!complete.cases(df2), ] #en az bir tane eksik olan satırlar
```

```
##    weight   height
## 19     NA       NA
## 16     NA 154.7867
## 18     NA       NA
## 17     NA 165.4223
## 20     NA       NA
```

```
df2[complete.cases(df2), ]$weight
```

```
##  [1] 74.75368 63.84519 91.38781 51.49316 60.05958 85.82164 65.45696 70.99893
##  [9] 89.85193 79.16534 75.23416 48.13977 66.20157 57.45846 50.42964
```

1.1.1 Eksik Verileri Silme

```r
# eksik veriden tamamen kurtulma
na.omit(df2)
```

```
##      weight   height
## 14 74.75368 143.7565
## 11 63.84519 141.2592
## 8  91.38781 180.4005
## 2  51.49316 139.8969
## 6  60.05958 168.8389
## 13 85.82164 149.6221
## 7  65.45696 157.4419
## 1  70.99893 141.0499
## 15 89.85193 174.8743
## 10 79.16534 162.1014
## 12 75.23416 166.0423
## 9  48.13977 171.3609
## 4  66.20157 145.4789
## 3  57.45846 163.5980
## 5  50.42964 153.8377
```

```r
complete.cases(df2)
```

```
##  [1]  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## [13]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
```

```r
df2[complete.cases(df2), ] # dolu olanlar satırlar
```

```
##      weight   height
## 14 74.75368 143.7565
## 11 63.84519 141.2592
## 8  91.38781 180.4005
## 2  51.49316 139.8969
## 6  60.05958 168.8389
## 13 85.82164 149.6221
## 7  65.45696 157.4419
## 1  70.99893 141.0499
## 15 89.85193 174.8743
## 10 79.16534 162.1014
## 12 75.23416 166.0423
## 9  48.13977 171.3609
## 4  66.20157 145.4789
## 3  57.45846 163.5980
## 5  50.42964 153.8377
```

```
df2[complete.cases(df2), ]$weight # değişken bazında dolu olan satırlar
```

```
##  [1] 74.75368 63.84519 91.38781 51.49316 60.05958 85.82164 65.45696 70.99893
##  [9] 89.85193 79.16534 75.23416 48.13977 66.20157 57.45846 50.42964
```

1.1.2 İmputasyon

```
# eksik verilere basit değer atama
df2$weight2 <- ifelse(is.na(df2$weight),mean(df2$weight, na.rm = TRUE),df2$weight)
sapply(df2, function(x) ifelse(is.na(x), mean(x, na.rm = TRUE), x ))
```

```
##         weight   height  weight2
##  [1,] 74.75368 143.7565 74.75368
##  [2,] 68.68652 157.6334 68.68652
##  [3,] 68.68652 154.7867 68.68652
##  [4,] 63.84519 141.2592 63.84519
##  [5,] 68.68652 157.6334 68.68652
##  [6,] 91.38781 180.4005 91.38781
##  [7,] 51.49316 139.8969 51.49316
##  [8,] 60.05958 168.8389 60.05958
##  [9,] 68.68652 165.4223 68.68652
## [10,] 85.82164 149.6221 85.82164
## [11,] 65.45696 157.4419 65.45696
## [12,] 70.99893 141.0499 70.99893
## [13,] 89.85193 174.8743 89.85193
## [14,] 79.16534 162.1014 79.16534
## [15,] 75.23416 166.0423 75.23416
## [16,] 48.13977 171.3609 48.13977
## [17,] 66.20157 145.4789 66.20157
## [18,] 68.68652 157.6334 68.68652
## [19,] 57.45846 163.5980 57.45846
## [20,] 50.42964 153.8377 50.42964
```

```
library(zoo)
sapply(df2, function(x) ifelse(is.na(x), na.locf(x), x )) # carry forward
```

```
##         weight   height  weight2
##  [1,] 74.75368 143.7565 74.75368
##  [2,] 74.75368 143.7565 68.68652
##  [3,] 74.75368 154.7867 68.68652
##  [4,] 63.84519 141.2592 63.84519
```

```
##  [5,] 63.84519 141.2592 68.68652
##  [6,] 91.38781 180.4005 91.38781
##  [7,] 51.49316 139.8969 51.49316
##  [8,] 60.05958 168.8389 60.05958
##  [9,] 60.05958 165.4223 68.68652
## [10,] 85.82164 149.6221 85.82164
## [11,] 65.45696 157.4419 65.45696
## [12,] 70.99893 141.0499 70.99893
## [13,] 89.85193 174.8743 89.85193
## [14,] 79.16534 162.1014 79.16534
## [15,] 75.23416 166.0423 75.23416
## [16,] 48.13977 171.3609 48.13977
## [17,] 66.20157 145.4789 66.20157
## [18,] 66.20157 145.4789 68.68652
## [19,] 57.45846 163.5980 57.45846
## [20,] 50.42964 153.8377 50.42964
```

```
sapply(df2, function(x) ifelse(is.na(x), na.locf(x,fromlast=TRUE), x ))
```

```
##         weight   height  weight2
##  [1,] 74.75368 143.7565 74.75368
##  [2,] 74.75368 143.7565 68.68652
##  [3,] 74.75368 154.7867 68.68652
##  [4,] 63.84519 141.2592 63.84519
##  [5,] 63.84519 141.2592 68.68652
##  [6,] 91.38781 180.4005 91.38781
##  [7,] 51.49316 139.8969 51.49316
##  [8,] 60.05958 168.8389 60.05958
##  [9,] 60.05958 165.4223 68.68652
## [10,] 85.82164 149.6221 85.82164
## [11,] 65.45696 157.4419 65.45696
## [12,] 70.99893 141.0499 70.99893
## [13,] 89.85193 174.8743 89.85193
## [14,] 79.16534 162.1014 79.16534
## [15,] 75.23416 166.0423 75.23416
## [16,] 48.13977 171.3609 48.13977
## [17,] 66.20157 145.4789 66.20157
## [18,] 66.20157 145.4789 68.68652
## [19,] 57.45846 163.5980 57.45846
## [20,] 50.42964 153.8377 50.42964
```

```
sapply(df2, function(x) ifelse(is.na(x), na.approx(x), x )) # linear interpolation
```

```
##         weight   height  weight2
```

```
##  [1,] 74.75368 143.7565 74.75368
##  [2,] 71.11752 149.2716 68.68652
##  [3,] 67.48135 154.7867 68.68652
##  [4,] 63.84519 141.2592 63.84519
##  [5,] 77.61650 160.8298 68.68652
##  [6,] 91.38781 180.4005 91.38781
##  [7,] 51.49316 139.8969 51.49316
##  [8,] 60.05958 168.8389 60.05958
##  [9,] 72.94061 165.4223 68.68652
## [10,] 85.82164 149.6221 85.82164
## [11,] 65.45696 157.4419 65.45696
## [12,] 70.99893 141.0499 70.99893
## [13,] 89.85193 174.8743 89.85193
## [14,] 79.16534 162.1014 79.16534
## [15,] 75.23416 166.0423 75.23416
## [16,] 48.13977 171.3609 48.13977
## [17,] 66.20157 145.4789 66.20157
## [18,] 61.83001 154.5384 68.68652
## [19,] 57.45846 163.5980 57.45846
## [20,] 50.42964 153.8377 50.42964
```

```
sapply(df2, function(x) ifelse(is.na(x), na.approx(x), x )) # cubic interpolation
```

```
##         weight   height  weight2
##  [1,] 74.75368 143.7565 74.75368
##  [2,] 71.11752 149.2716 68.68652
##  [3,] 67.48135 154.7867 68.68652
##  [4,] 63.84519 141.2592 63.84519
##  [5,] 77.61650 160.8298 68.68652
##  [6,] 91.38781 180.4005 91.38781
##  [7,] 51.49316 139.8969 51.49316
##  [8,] 60.05958 168.8389 60.05958
##  [9,] 72.94061 165.4223 68.68652
## [10,] 85.82164 149.6221 85.82164
## [11,] 65.45696 157.4419 65.45696
## [12,] 70.99893 141.0499 70.99893
## [13,] 89.85193 174.8743 89.85193
## [14,] 79.16534 162.1014 79.16534
## [15,] 75.23416 166.0423 75.23416
## [16,] 48.13977 171.3609 48.13977
## [17,] 66.20157 145.4789 66.20157
## [18,] 61.83001 154.5384 68.68652
## [19,] 57.45846 163.5980 57.45846
## [20,] 50.42964 153.8377 50.42964
```

```r
# KNN (k-nearest neighbor) ile Değer Atama

library(dplyr)
library(DMwR2)
# airquality verisi
df_air <- as_tibble(airquality)
df_air
```

```
## # A tibble: 153 x 6
##    Ozone Solar.R  Wind  Temp Month   Day
##    <int>   <int> <dbl> <int> <int> <int>
##  1    41     190   7.4    67     5     1
##  2    36     118   8      72     5     2
##  3    12     149  12.6    74     5     3
##  4    18     313  11.5    62     5     4
##  5    NA      NA  14.3    56     5     5
##  6    28      NA  14.9    66     5     6
##  7    23     299   8.6    65     5     7
##  8    19      99  13.8    59     5     8
##  9     8      19  20.1    61     5     9
## 10    NA     194   8.6    69     5    10
## # ... with 143 more rows
```

```r
anyNA(df_air)
```

```
## [1] TRUE
```

```r
# airquality verisindeki Wind değişkeninin bazı değerlerini NA yapalım
set.seed(1234)
row_num <- sample(1:nrow(airquality),5)
row_num # bu satırdaki değerlere NA atanacak
```

```
## [1]  28  80 150 101 111
```

```r
airquality_2 <- airquality
airquality_2[row_num,"Wind"] <- NA
airquality_2[row_num,"Wind"]
```

```
## [1] NA NA NA NA NA
```

```
head(airquality_2,20)
```

```
##    Ozone Solar.R Wind Temp Month Day
## 1     41     190  7.4   67     5   1
## 2     36     118  8.0   72     5   2
## 3     12     149 12.6   74     5   3
## 4     18     313 11.5   62     5   4
## 5     NA      NA 14.3   56     5   5
## 6     28      NA 14.9   66     5   6
## 7     23     299  8.6   65     5   7
## 8     19      99 13.8   59     5   8
## 9      8      19 20.1   61     5   9
## 10    NA     194  8.6   69     5  10
## 11     7      NA  6.9   74     5  11
## 12    16     256  9.7   69     5  12
## 13    11     290  9.2   66     5  13
## 14    14     274 10.9   68     5  14
## 15    18      65 13.2   58     5  15
## 16    14     334 11.5   64     5  16
## 17    34     307 12.0   66     5  17
## 18     6      78 18.4   57     5  18
## 19    30     322 11.5   68     5  19
## 20    11      44  9.7   62     5  20
```

```
# k parametresi, verilen bir noktaya en yakın komşuların sayısıdır.
# Örneğin: k=5 olsun. Bu durumda mesafeye (öklit) göre en yakın 5 komşu belirlenir
# ve mesafenin ağırlıklı ortalaması hesaplanır.
# ağırlıklandırma, her komşuya 1 / d ağırlığının verilmesini içerir.
# burada d komşuya olan uzaklıktır.
knn_df_air <- knnImputation(airquality_2, k = 5) # k komşu sayısı

result <- data.frame(row=row_num,
                     orig=airquality[row_num,"Wind"],
                     knn=knn_df_air[row_num,"Wind"])
result
```

```
##   row orig       knn
## 1  28 12.0 10.079819
## 2  80  5.1  8.765250
## 3 150 13.2  9.914454
## 4 101  8.0  6.807361
## 5 111 10.9 11.237192
```

```
mean(result$orig-result$knn)
```

```
## [1] 0.4791848
```

Eksik verilerin analiz edilmesi ve imputasyon konusunda R içerisinde çeşitli kütühaneler bulunmaktadır. Bunlardan en çok bilinenleri **`mice`, `VIM`, `missForest`, `imputation`, `naniar`, `mi`, `Amelia`** paketleridir. Ayrıca **Sosyal Bilimler konuları içerisindeki eksik veriler bölümünden de** yararlanılabilir.

1.2 Aykırı Değer Analizi

Aykırı değer, diğer gözlemlerden uzak olan, yani diğer veri noktalarından önemli ölçüde farklı olan bir veri noktası olan bir değer veya gözlemdir. Bu dokümanda, minimum ve maksimum, histogram, kutu grafiği, (box-plot), yüzdelikler ve Z-Skoru ile aykırı değer analizi anlatılacaktır.

1.2.1 Minumum ve Maximum

```
library(ggplot2)

# mpg verisindeki hwy değişkeni üzerinden inceleyelim
summary(mpg$hwy)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12.00   18.00   24.00   23.44   27.00   44.00
```

```
min(mpg$hwy)
```

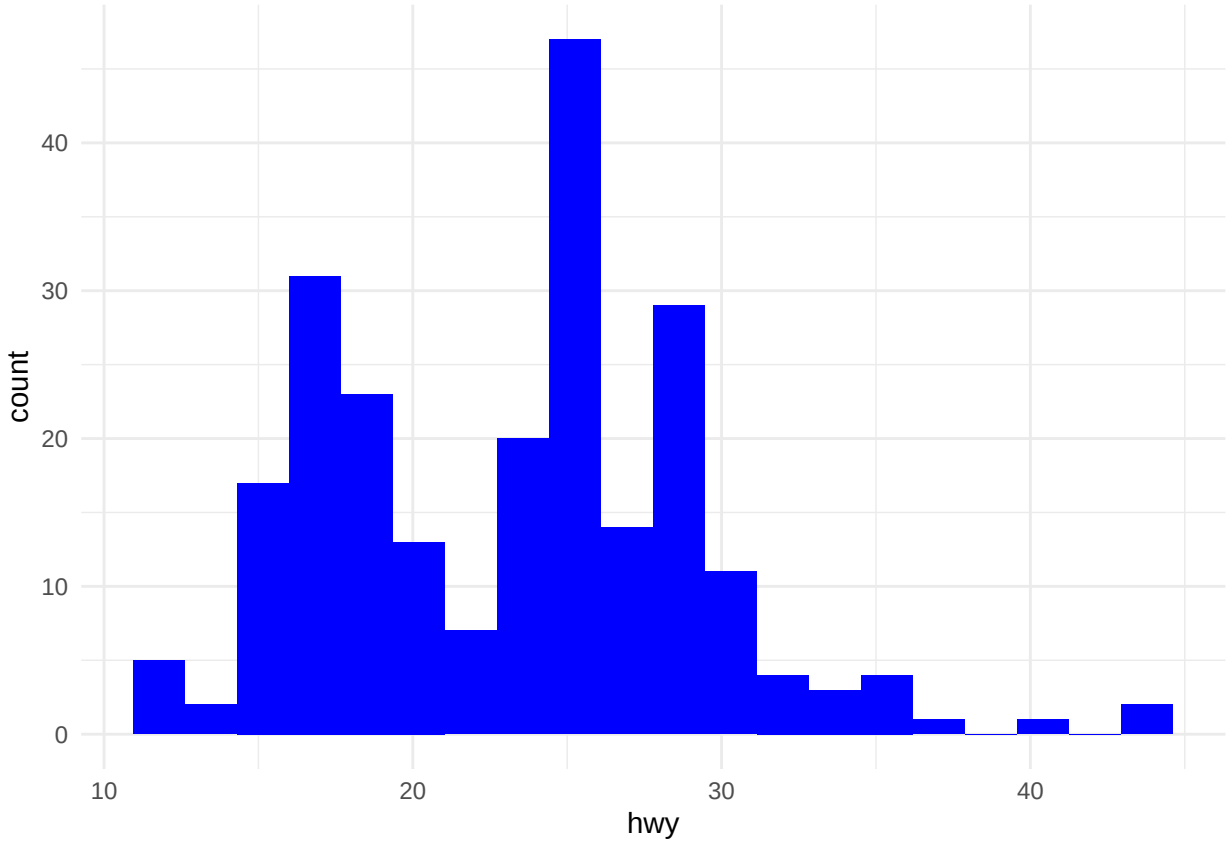
```
## [1] 12
```

```
max(mpg$hwy)
```

```
## [1] 44
```

1.2.2 Histogram

```
ggplot(mpg) +
  aes(x = hwy) +
  geom_histogram(bins = 20, fill = "blue") +
  theme_minimal()
```




```
# grafiğiin sağ tarafında kalan gözlemler şüpheli görünüyor.
```

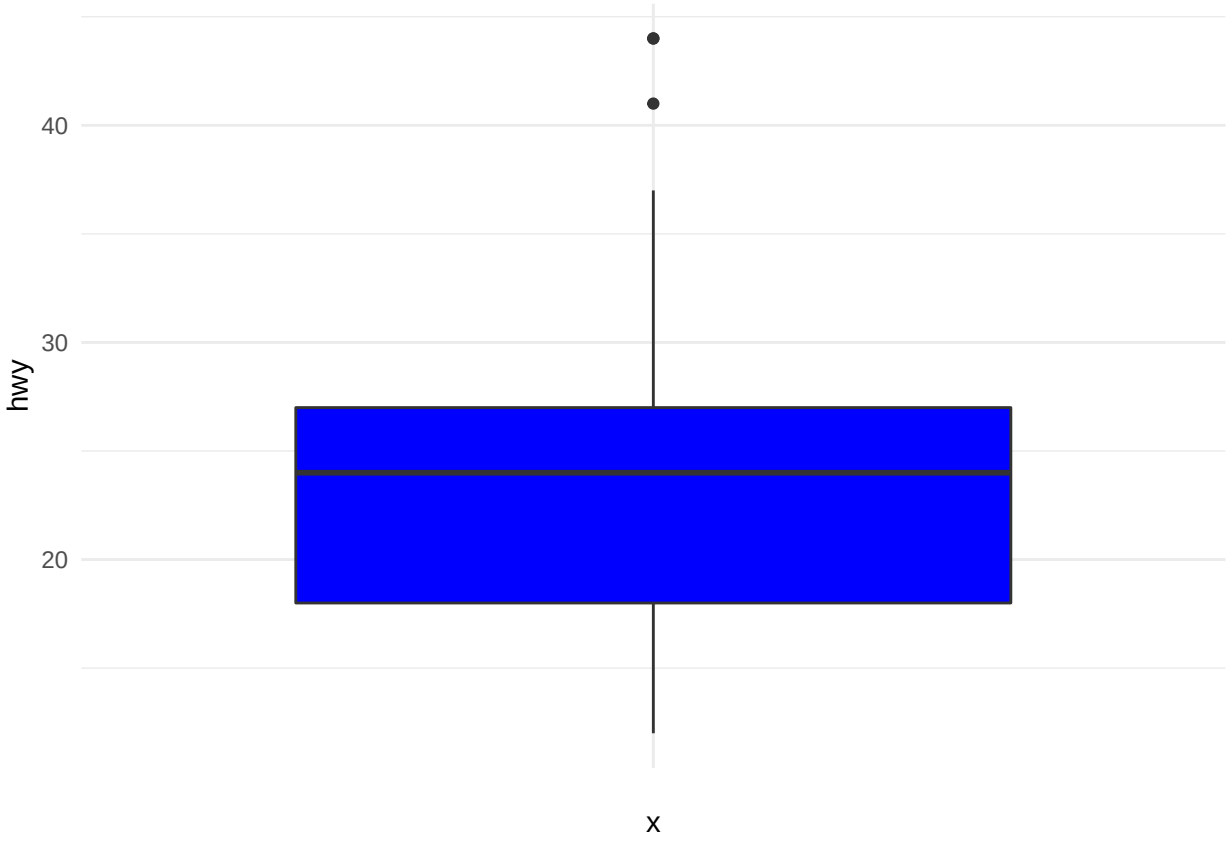
1.2.3 Boxplot

Bir boxplot grafiği, beş konum özetini (minimum, ortanca, birinci ve üçüncü çeyrekler ve maksimum) ve çeyrekler arası aralık (IQR) kriteri kullanılarak şüpheli bir aykırı değer olarak sınıflandırılan herhangi bir gözlemi görüntüleyerek nicel bir değişkeni görselleştirmeye yardımcı olur.

$I = [Q_1 - 1.5 * IQR; Q_3 + 1.5 * IQR]$

IQR ise üçüncü ve birinci çeyrek arasındaki farktır. R içerisindeki `IQR()` fonksiyonu bu amaçla kullanılabilir.

```
ggplot(mpg) +
  aes(x = "", y = hwy) +
  geom_boxplot(fill = "blue") +
  theme_minimal()
```




```
# outlier değrlerine erişim
boxplot.stats(mpg$hwy)$out
```

```
## [1] 44 44 41
```

```
# outier olarak görülen değerlerin konumları
hwy_out <- boxplot.stats(mpg$hwy)$out
hwy_out_sira <- which(mpg$hwy %in% c(hwy_out))
hwy_out_sira
```

```
## [1] 213 222 223
```

```
# outlier olarak görülen satırlar
mpg[hwy_out_sira, ]
```

```
## # A tibble: 3 x 11
##   manufacturer model   displ  year   cyl trans   drv     cty   hwy fl    class
##   <chr>        <chr>   <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
## 1 volkswagen   jetta     1.9  1999     4 manual~ f        33    44 d     compact
## 2 volkswagen   new be~   1.9  1999     4 manual~ f        35    44 d     subcom~
## 3 volkswagen   new be~   1.9  1999     4 auto(l~ f        29    41 d     subcom~
```

1.2.4 Yüzdelikler (Percentiles)

Bu aykırı değer tespiti yöntemi, yüzdelik dilimlere dayalıdır. Yüzdelikler yöntemiyle, 2,5 ve 97,5 yüzdelik dilimlerin oluşturduğu aralığın dışında kalan tüm gözlemler potansiyel aykırı değerler olarak kabul edilecektir. Aralığı oluşturmak için 1 ve 99 veya 5 ve 95 yüzdelikler gibi diğer yüzdelikler de düşünülebilir.

```
alt_sinir <- quantile(mpg$hwy, 0.025)
alt_sinir
```

```
## 2.5%
##   14
```

```
ust_sinir <- quantile(mpg$hwy, 0.975)
ust_sinir
```

```
##  97.5%
## 35.175
```

```
# Bu yönteme göre, 14'ün altındaki ve 35.175'in üzerindeki tüm gözlemler,
# potansiyel aykırı değerler olarak kabul edilecektir.

outlier_sira <- which(mpg$hwy < alt_sinir | mpg$hwy > ust_sinir)
outlier_sira
```

```
##  [1]  55  60  66  70 106 107 127 197 213 222 223
```

```
# Bu yönteme göre 11 adet outlier bulunmuştur.
mpg[outlier_sira,]
```

```
## # A tibble: 11 x 11
##    manufacturer model    displ  year   cyl trans  drv     cty   hwy fl    class
##    <chr>        <chr>    <dbl> <int> <int> <chr>  <chr> <int> <int> <chr> <chr>
##  1 dodge        dakota ~   4.7  2008     8 auto(~ 4         9    12 e     pickup
##  2 dodge        durango~   4.7  2008     8 auto(~ 4         9    12 e     suv
##  3 dodge        ram 150~   4.7  2008     8 auto(~ 4         9    12 e     pickup
##  4 dodge        ram 150~   4.7  2008     8 manua~ 4         9    12 e     pickup
##  5 honda        civic      1.8  2008     4 auto(~ f        25    36 r     subco~
##  6 honda        civic      1.8  2008     4 auto(~ f        24    36 c     subco~
##  7 jeep         grand c~   4.7  2008     8 auto(~ 4         9    12 e     suv
##  8 toyota       corolla    1.8  2008     4 manua~ f        28    37 r     compa~
##  9 volkswagen   jetta      1.9  1999     4 manua~ f        33    44 d     compa~
## 10 volkswagen   new bee~   1.9  1999     4 manua~ f        35    44 d     subco~
## 11 volkswagen   new bee~   1.9  1999     4 auto(~ f        29    41 d     subco~
```

```
# Sınırları biraz daha küçültelim
alt_sinir <- quantile(mpg$hwy, 0.01)
ust_sinir <- quantile(mpg$hwy, 0.99)

outlier_sira <- which(mpg$hwy < alt_sinir | mpg$hwy > ust_sinir)

mpg[outlier_sira, ]
```

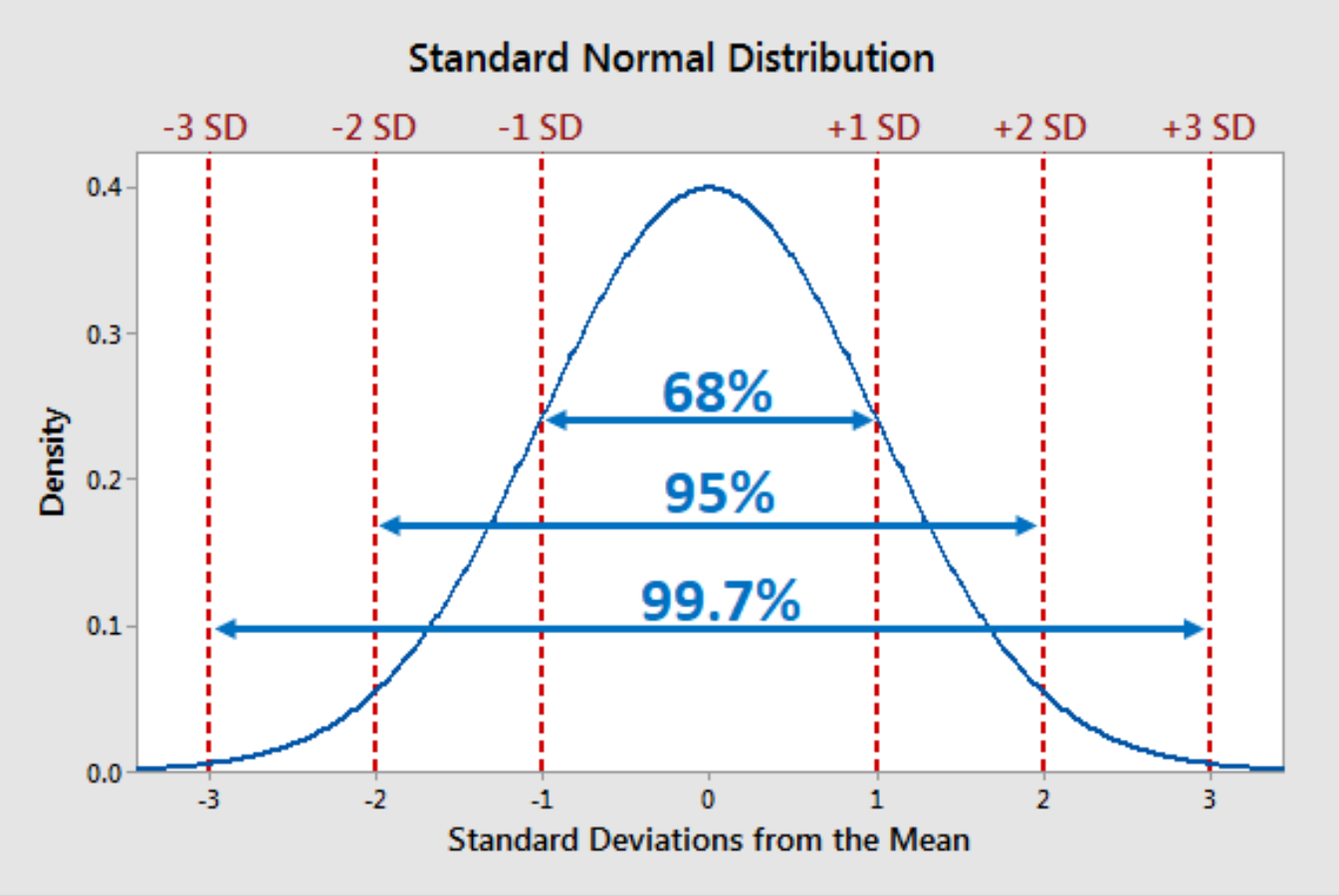
```
## # A tibble: 3 x 11
##   manufacturer model   displ  year   cyl trans   drv     cty   hwy fl    class
##   <chr>        <chr>   <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
## 1 volkswagen   jetta     1.9  1999     4 manual~ f        33    44 d     compact
## 2 volkswagen   new be~   1.9  1999     4 manual~ f        35    44 d     subcom~
## 3 volkswagen   new be~   1.9  1999     4 auto(l~ f        29    41 d     subcom~
```

```
# Buna göre IQR ile elde edildiği gibi 3 adet outlier bulundu.
```

1.2.5 Z-Skor Yöntemi

Aykırı değerlerin tespitinde ortalama ve standart sapmanın kulllanıldığı en bilinen yöntemlerdendir ve aşağıdaki şekilde hesaplanır.

$Z_i = \frac{(X_i - \mu)}{\sigma}$




```
std_z <- function(x){

  z=(x-mean(x))/sd(x)
  return(z)
}

mpg$hwy_std <- std_z(mpg$hwy)
mpg[,c("hwy","hwy_std")]
```

```
## # A tibble: 234 x 2
##      hwy hwy_std
##    <int>   <dbl>
##  1    29   0.934
##  2    29   0.934
##  3    31   1.27
##  4    30   1.10
##  5    26   0.430
##  6    26   0.430
##  7    27   0.598
##  8    26   0.430
```

```
##  9    25   0.262
## 10    28   0.766
## # ... with 224 more rows
```

```
# -3 ve +3 sapma dışında kalanları aykırı değer olarak kabul ediyoruz.
outliers_zskor <- which(mpg$hwy_std < -3 | mpg$hwy_std > +3)
outliers_zskor
```

```
## [1] 213 222
```

```
mpg[outliers_zskor,c() ]
```

```
## # A tibble: 2 x 0
```

```
# bu yönteme göre 2 adet aykırı değer bulunmuştur.
```

1.3 Veri Normalleştirme

Değişkenler farklı ölçeklerde ölçüldüğünde, genellikle analize eşit katkıda bulunmazlar. Örneğin, bir değişkenin değerleri 0 ile 100.000 arasında ve başka bir değişkenin değerleri 0 ile 100 arasında değişiyorsa, daha büyük aralığa sahip değişkene analizde daha büyük bir ağırlık verilecektir. Değişkenleri normalleştirerek, her bir değişkenin analize eşit katkı sağladığından emin olabiliriz. Değişkenleri normalleştirmek için (veya ölçeklendirmek) genellikle min-max ya da z dönüşümü yöntemleri kullanılır.

```
# min-max dönüşümleri

# 0 ile 1 arasi dönüşüm
std_0_1 <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}

#-1 ile +1 arası dönüşüm
std_1_1 <- function(x) {
  ((x - mean(x)) / max(abs(x - mean(x))))
}

# a ile b arası dönüşüm
std_min_max <- function(x,a,b) {
  # a min değer
  # b max değer
  (a + ((x - min(x)) * (b - a)) / (max(x) - min(x)))
```

```
}

set.seed(12345)
dat <- data.frame(x = rnorm(20, 10, 3),
                  y = rnorm(20, 30, 8),
                  z = rnorm(20, 25, 5))
dat
```

```
##            x        y        z
## 1  11.756586 36.23698 30.64255
## 2  12.128398 41.64628 13.09821
## 3   9.672090 24.84537 19.69867
## 4   8.639508 17.57490 29.68570
## 5  11.817662 17.21832 29.27226
## 6   4.546132 44.44078 32.30365
## 7  11.890296 26.14682 17.93451
## 8   9.171448 34.96304 27.83702
## 9   9.147521 34.89699 27.91594
## 10  7.242034 28.70151 18.46601
## 11  9.651257 36.49499 22.29807
## 12 15.451936 47.57467 34.73846
## 13 11.111884 46.39352 25.26795
## 14 11.560649 43.05957 26.75831
## 15  7.748404 32.03417 21.64512
## 16 12.450700 33.92951 26.38977
## 17  7.340927 27.40731 28.45586
## 18  9.005267 16.70360 29.11898
## 19 13.362138 44.14187 35.72533
## 20 10.896171 30.20641 13.26528
```

```
summary(dat)
```

```
##        x                y                z        
##  Min.   : 4.546   Min.   :16.70   Min.   :13.10  
##  1st Qu.: 8.914   1st Qu.:27.09   1st Qu.:21.16  
##  Median :10.284   Median :34.41   Median :27.30  
##  Mean   :10.230   Mean   :33.23   Mean   :25.53  
##  3rd Qu.:11.836   3rd Qu.:42.00   3rd Qu.:29.38  
##  Max.   :15.452   Max.   :47.57   Max.   :35.73
```

```
apply(dat, 2, std_0_1)
```

```
##               x          y           z
##  [1,] 0.6611575 0.63274053 0.775368144
##  [2,] 0.6952505 0.80796300 0.000000000
##  [3,] 0.4700211 0.26373477 0.291705877
##  [4,] 0.3753393 0.02822392 0.733080320
##  [5,] 0.6667578 0.01667340 0.714808256
##  [6,] 0.0000000 0.89848463 0.848779748
##  [7,] 0.6734179 0.30589231 0.213738973
##  [8,] 0.4241150 0.59147416 0.651378062
##  [9,] 0.4219211 0.58933460 0.654866001
## [10,] 0.2471988 0.38864587 0.237228478
## [11,] 0.4681108 0.64109819 0.406585628
## [12,] 1.0000000 1.00000000 0.956385878
## [13,] 0.6020419 0.96173940 0.537838847
## [14,] 0.6431912 0.85374322 0.603705080
## [15,] 0.2936301 0.49659993 0.377728555
## [16,] 0.7248037 0.55799517 0.587417289
## [17,] 0.2562668 0.34672297 0.678727553
## [18,] 0.4088772 0.00000000 0.708033996
## [19,] 0.8083774 0.88880212 1.000000000
## [20,] 0.5822623 0.43739366 0.007383637
```

```
library(dplyr)

dat %>% mutate_all(std_0_1) %>% summary()
```

```
##        x                y                z          
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.4005   1st Qu.:0.3365   1st Qu.:0.3562  
##  Median :0.5261   Median :0.5737   Median :0.6275  
##  Mean   :0.5211   Mean   :0.5354   Mean   :0.5492  
##  3rd Qu.:0.6684   3rd Qu.:0.8194   3rd Qu.:0.7194  
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

```
dat %>% mutate_all(std_1_1) %>% summary()
```

```
##        x                   y                 z          
##  Min.   :-1.000000   Min.   :-1.00000   Min.   :-1.0000  
##  1st Qu.:-0.231502   1st Qu.:-0.37143   1st Qu.:-0.3514  
##  Median : 0.009603   Median : 0.07154   Median : 0.1426  
##  Mean   : 0.000000   Mean   : 0.00000   Mean   : 0.0000  
##  3rd Qu.: 0.282624   3rd Qu.: 0.53057   3rd Qu.: 0.3098  
##  Max.   : 0.918881   Max.   : 0.86789   Max.   : 0.8207
```

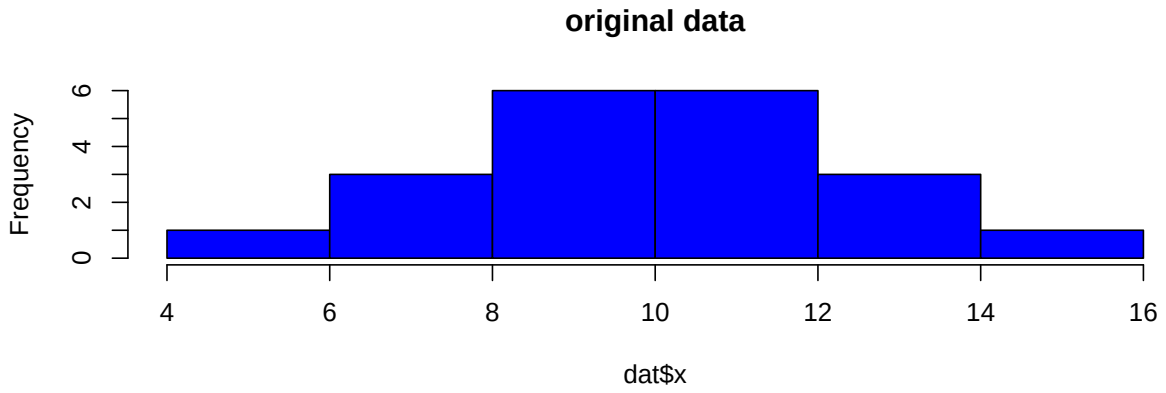
```
dat %>% mutate_all(std_min_max, a = -2, b = 2) %>% summary()
```

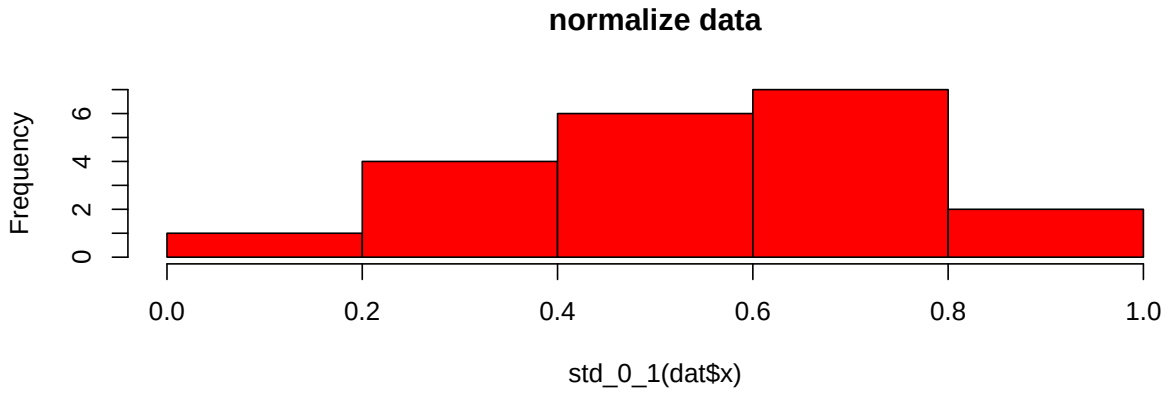
```
##        x                   y                   z
##  Min.   :-2.00000   Min.   :-2.0000   Min.   :-2.0000
##  1st Qu.:-0.39803   1st Qu.:-0.6539   1st Qu.:-0.5751
##  Median : 0.10457   Median : 0.2947   Median : 0.5102
##  Mean   : 0.08455   Mean   : 0.1415   Mean   : 0.1970
##  3rd Qu.: 0.67369   3rd Qu.: 1.2776   3rd Qu.: 0.8775
##  Max.   : 2.00000   Max.   : 2.0000   Max.   : 2.0000
```

```
dat %>% mutate_all(std_z) %>% summary()
```

```
##        x                   y                   z
##  Min.   :-2.27173   Min.   :-1.7088   Min.   :-1.9165
##  1st Qu.:-0.52591   1st Qu.:-0.6347   1st Qu.:-0.6735
##  Median : 0.02182   Median : 0.1223   Median : 0.2732
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.64204   3rd Qu.: 0.9067   3rd Qu.: 0.5937
##  Max.   : 2.08745   Max.   : 1.4831   Max.   : 1.5729
```

```
par(mfrow=c(2,1))
hist(dat$x,main="original data",col="blue")
hist(std_0_1(dat$x),main="normalize data",col="red")
```






```
# bu dönüşümler verinin dağılımını değiştirmemektedir.
```

2 Doğrusal Regresyon

Basit doğrusal regresyon, iki nicel değişken arasındaki doğrusal ilişkiyi değerlendirmeye izin veren istatistiksel bir yaklaşımdır. Daha doğrusu, ilişkinin nicelleştirilmesini ve öneminin değerlendirilmesini sağlar. Çoklu doğrusal regresyon, bu yaklaşımın bir yanıt değişkeni (nicel) ile birkaç açıklayıcı değişken (nicel veya nitel) arasındaki doğrusal ilişkileri değerlendirmeyi mümkün kılması anlamında, basit doğrusal regresyonun bir genellemesidir.

Basit doğrusal regresyonda, değişkenlerden biri yanıt veya bağımlı değişken olarak kabul edilir ve y ekseninde temsil edilir. Diğer değişken ise açıklayıcı veya bağımsız değişken olarak da adlandırılır ve x ekseninde temsil edilir.

Basit doğrusal regresyon, iki değişken arasında doğrusal bir ilişkinin varlığını değerlendirmeye ve bu bağlantıyı nicelleştirmeye izin verir. Doğrusallığın, iki değişkenin doğrusal olarak bağımlı olup olmadığını test etmesi ve ölçmesi anlamında doğrusal regresyonda güçlü bir varsayım olduğuna dikkat etmek gerekmektedir.

Doğrusal regresyonu güçlü bir istatistiksel araç yapan şey, açıklayıcı/bağımsız değişken bir birim arttığında yanıtın/bağımlı değişkenin hangi nicelikle değiştiğini ölçmeye izin vermesidir. Bu kavram doğrusal regresyonda anahtardır ve aşağıdaki soruları yanıtlamaya yardımcı olur:

- Reklama harcanan miktar ile belirli bir dönemdeki satışlar arasında bir bağlantı var mı?
- Tütün vergilerindeki artış tüketimini azaltır mı?
- Bölgeye bağlı olarak bir konutun en olası fiyatı nedir?
- Bir kişinin bir uyarana tepki verme süresi cinsiyete bağlı mıdır?

Basit doğrusal regresyon analizinde, bağımlı değişken y ile bağımsız değişken x arasındaki ilişki doğrusal bir denklem şeklinde verilir.

$$y = \beta_0 + \beta_1 x + \varepsilon$$

Burada, β_0 sayısına kesme noktası denir ve regresyon doğrusu ile y ekseninin (x=0) kesişme noktasını tanımlar. β_1 sayısına regresyon katsayısı denir. Regresyon doğrusu eğiminin bir ölçüsüdür. Böylece β_1, x değeri 1 birim arttığında y değerinin ne kadar değiştiğini gösterir. Model, x ve y arasında kesin bir ilişki verdiği için deterministik bir model olarak kabul edilir.

Ancak birçok durumda, iki değişken x ve y arasındaki ilişki kesin değildir. Bunun nedeni, bağımlı değişken y'nin, tahmin değişkeni x tarafından tam olarak yakalanmayan diğer bilinmeyen ve/veya rastgele süreçlerden etkilenmesidir. Böyle bir durumda veri noktaları düz bir çizgi üzerinde sıralanmaz. Bununla birlikte, veriler hala temeldeki doğrusal bir ilişkiyi takip edebilir. Bu bilinmeyenleri dikkate almak için doğrusal model denklemine ε ile gösterilen rastgele bir hata terimi eklenir ve olasılıklı bir model elde edilir. Burada hata terimi ε_i'nin bağımsız normal dağılımlı değerlerden oluştuğu varsayılır, $e_i \sim N(0, \sigma^2)$.

Doğrusal regresyon modeli hakkında aşağıdaki varsayımlar yapılır:

- Bağımlı değişken tesadüfi bir değişkendir ve normal dağılım göstermektedir.
- Tahmin hataları tesadüfidir ve normal dağılım gösterirler.
- Hatalar birbirinden bağımsızdır (otokorelasyon yoktur).
- Hata varyansı sabittir ve veriler arasında hiç değişmediği varsayılır (eşit varyanslılık-homoscedasticity).
- Eğer çoklu regresyon analizi yapılıyorsa, bağımsız değişkenlerin birbirleri ile bağlantısının olmaması gereklidir. Buna çoklu bağlantı (multicollinearity) olmaması varsayımı adı verilir.

- Bağımlı değişken ile bağımsız değişkenler arasında doğrusal bir ilişki olmalıdır.
- Gözlem sayısı parametre sayısından büyük olmalıdır.

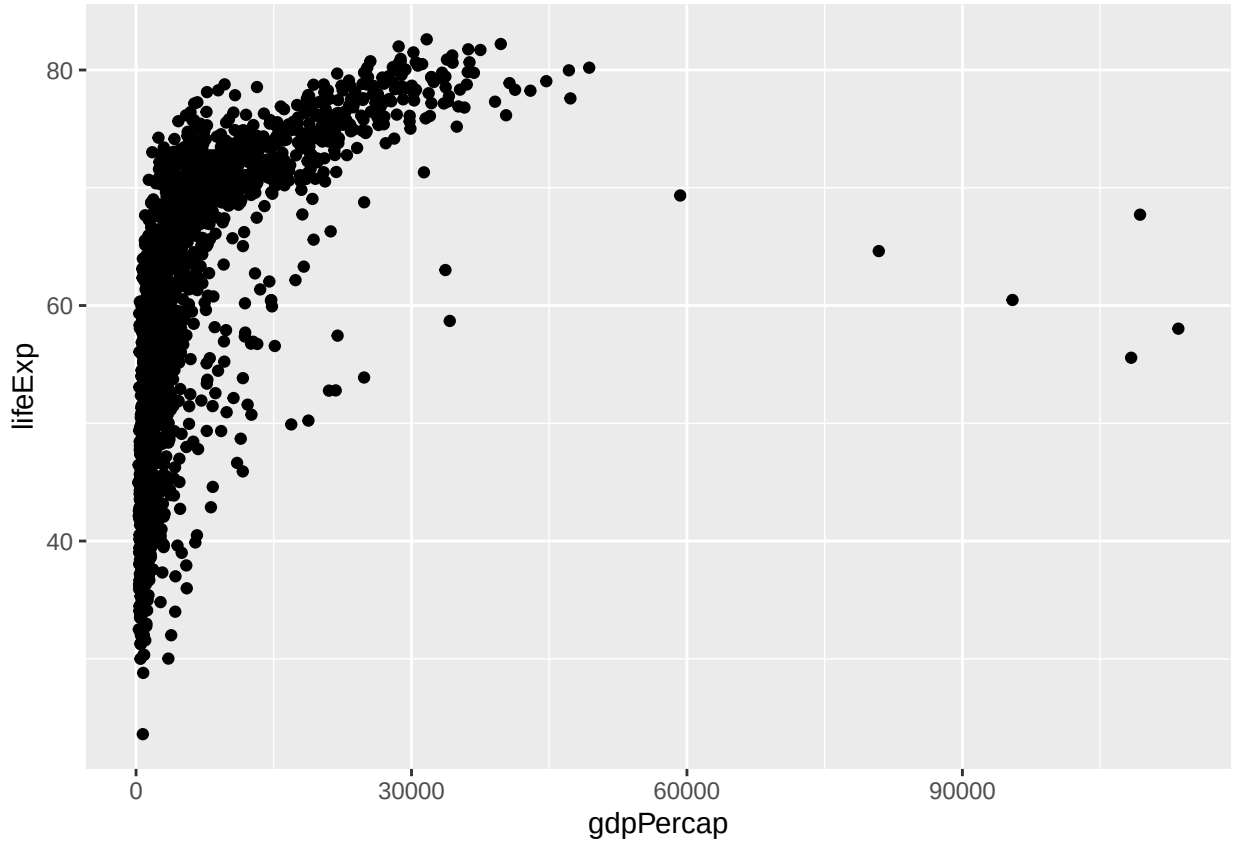
```
library(gapminder)
library(dplyr)
library(ggplot2)

# gapminder veri setine bakalım

glimpse(gapminder)
```

```
## Rows: 1,704
## Columns: 6
## $ country   <fct> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan", ~
## $ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, ~
## $ year      <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, ~
## $ lifeExp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 40.8~
## $ pop       <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372, 12~
## $ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.1134, ~
```

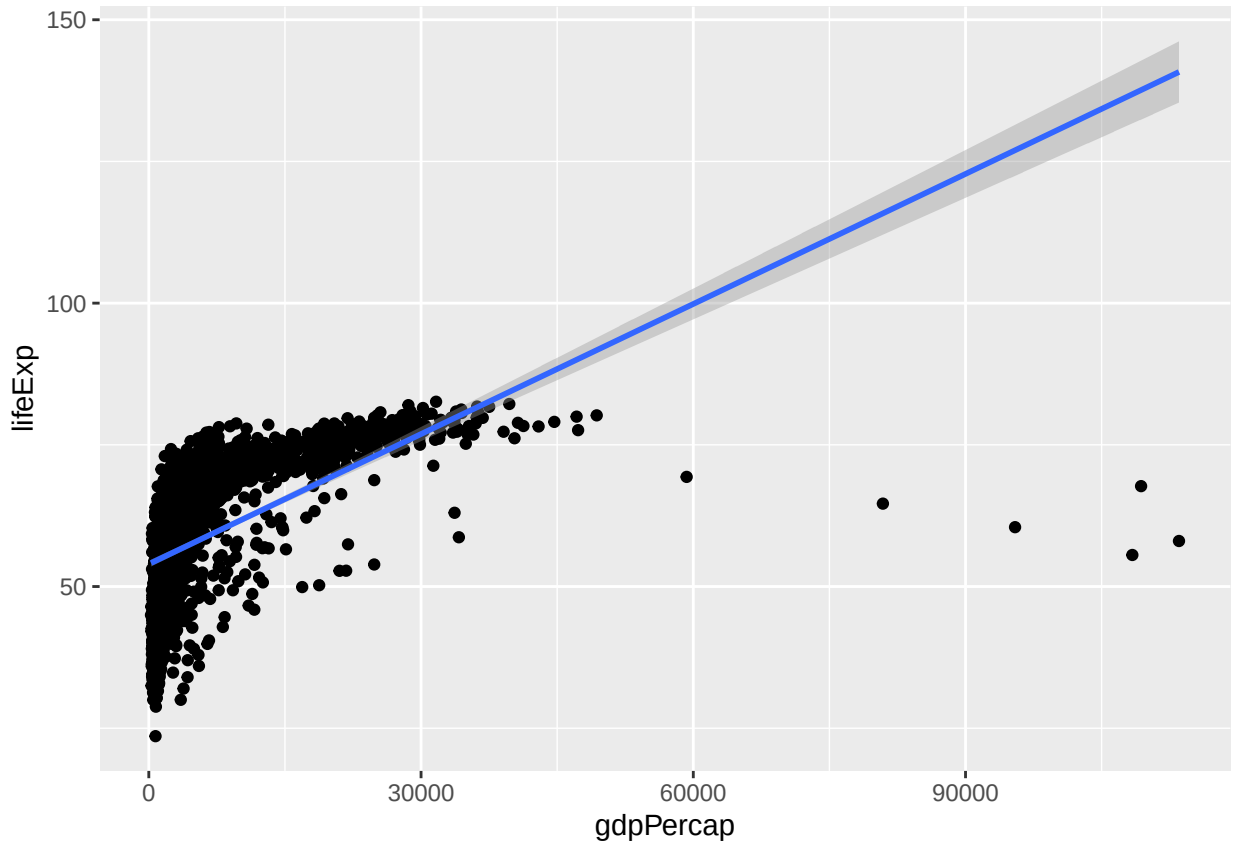
```
summary(gapminder)
```

```
##         country        continent        year         lifeExp
##  Afghanistan:  12   Africa  :624   Min.   :1952   Min.   :23.60
##  Albania    :  12   Americas:300   1st Qu.:1966   1st Qu.:48.20
##  Algeria    :  12   Asia    :396   Median :1980   Median :60.71
##  Angola     :  12   Europe  :360   Mean   :1980   Mean   :59.47
##  Argentina  :  12   Oceania : 24   3rd Qu.:1993   3rd Qu.:70.85
##  Australia  :  12                  Max.   :2007   Max.   :82.60
##  (Other)    :1632
##       pop              gdpPercap
##  Min.   :     60011   Min.   :   241.2
##  1st Qu.:   2793664   1st Qu.:  1202.1
##  Median :   7023596   Median :  3531.8
##  Mean   :  29601212   Mean   :  7215.3
##  3rd Qu.:  19585222   3rd Qu.:  9325.5
##  Max.   :1318683096   Max.   :113523.1
##
```

```
# kişi başına milli gelir ile yaşam beklentisi değişkenlerini görselleştirelim.

ggplot(gapminder, aes(gdpPercap, lifeExp)) +
  geom_point()
```




```
ggplot(gapminder, aes(gdpPercap, lifeExp)) +
  geom_point() +
  geom_smooth(method = "lm",se=TRUE)
```




```
# regresyon modeli kuralım

model1 <- lm(lifeExp ~ gdpPercap, data = gapminder)
model1
```

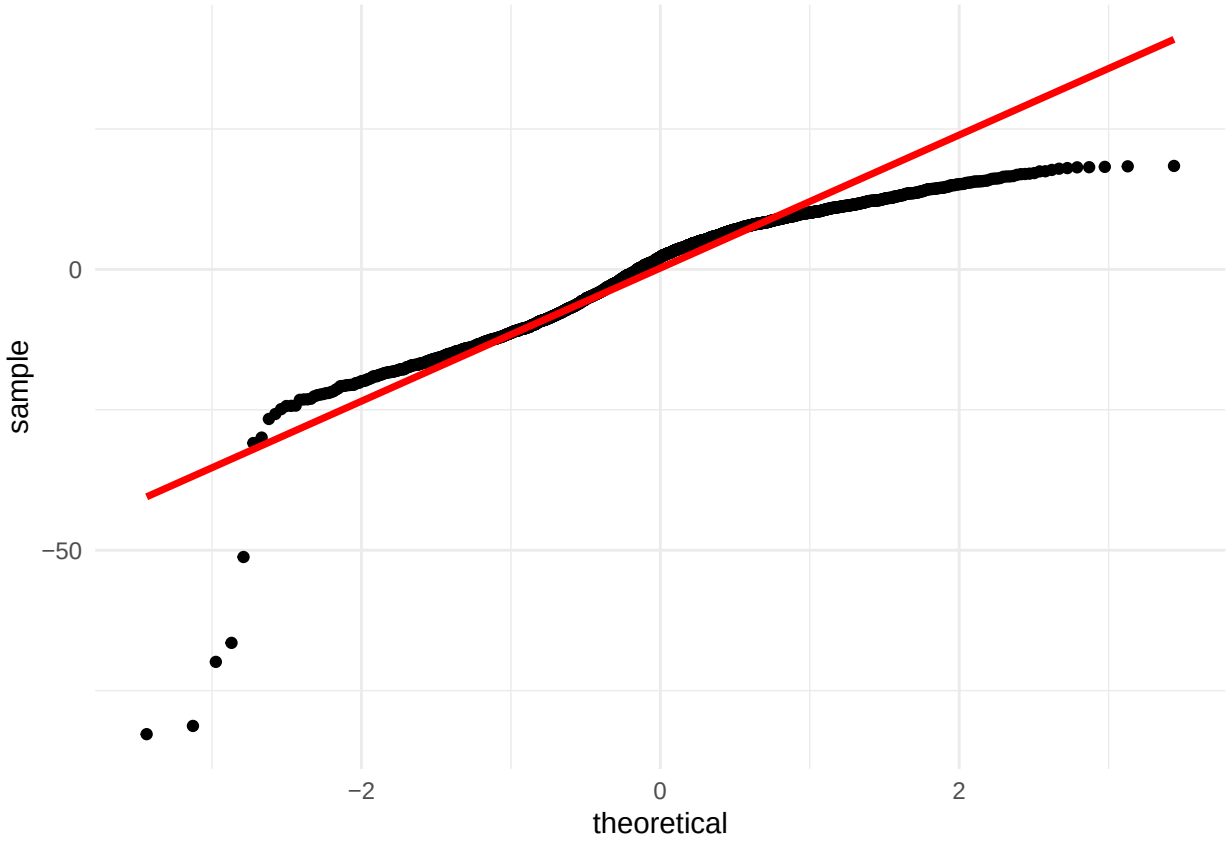
```
##
## Call:
## lm(formula = lifeExp ~ gdpPercap, data = gapminder)
##
## Coefficients:
## (Intercept)    gdpPercap
##  53.9555609    0.0007649
```

Yani burada söyleyebileceğimiz şey, GSYİH'daki her 1 artış için, yaşam beklentisinde 0.0007649 yıllık bir artış görmeyi bekleyebiliriz. Bu özellikle büyük değil - ama o zaman, GSYİH'de tek bir dolarlık artış da çok fazla değil! Modelimizi daha iyi anlayabilmek için model üzerinde summary() fonksiyonunu kullanabiliriz. Ayrıca artıkların normalliğini de bakmak fa fayda var.

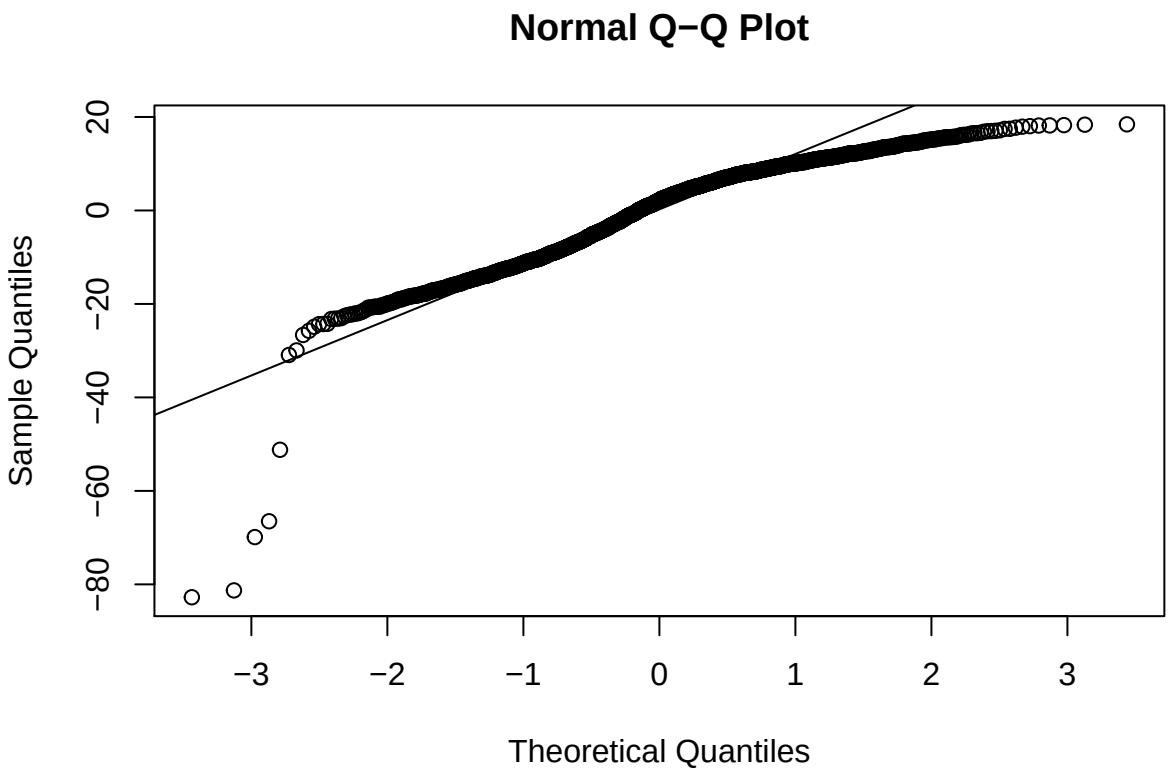
```
summary(model1)
```

```
##
## Call:
## lm(formula = lifeExp ~ gdpPercap, data = gapminder)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -82.754  -7.758   2.176   8.225  18.426
##
## Coefficients:
##                Estimate  Std. Error t value            Pr(>|t|)
## (Intercept) 53.95556088  0.31499494  171.29 <0.0000000000000002 ***
## gdpPercap    0.00076488  0.00002579   29.66 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.49 on 1702 degrees of freedom
## Multiple R-squared:  0.3407, Adjusted R-squared:  0.3403
## F-statistic: 879.6 on 1 and 1702 DF,  p-value: < 0.00000000000000022
```

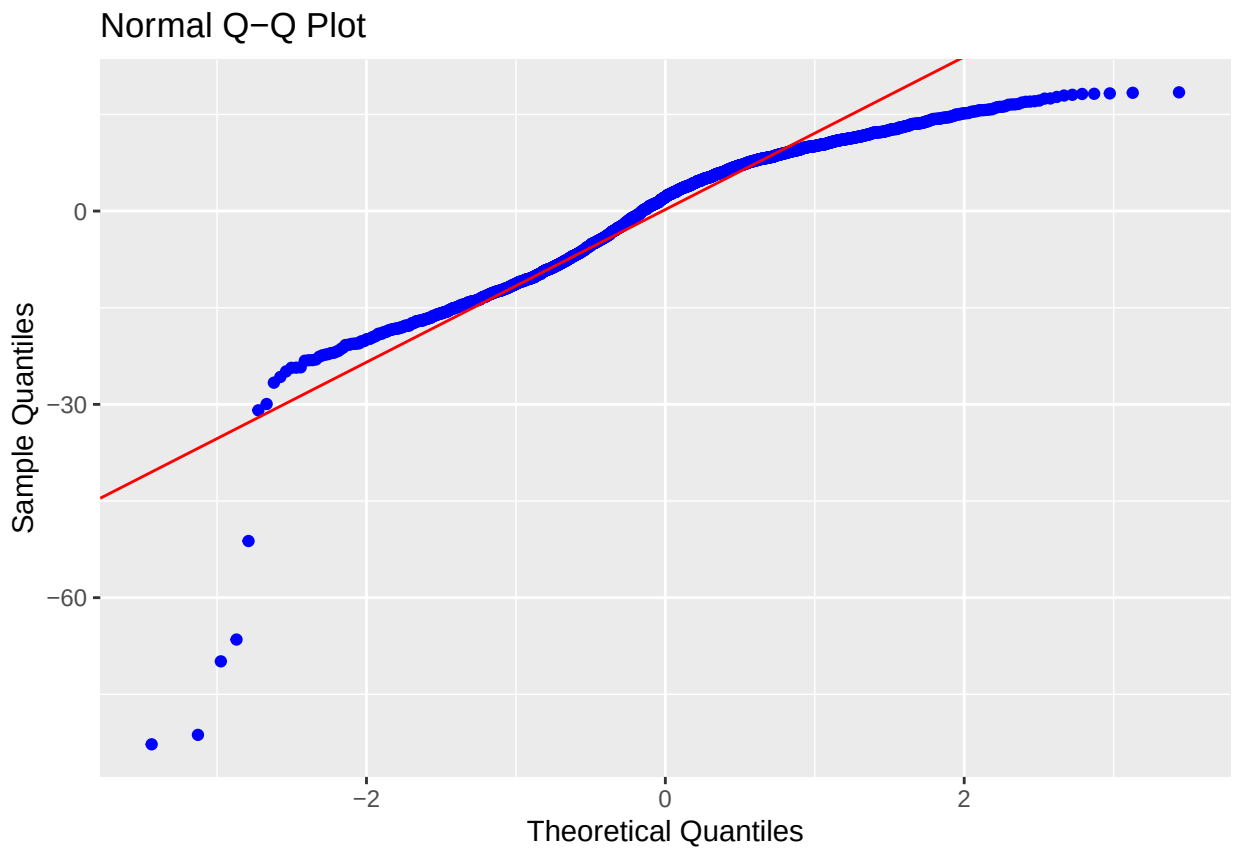
```
#artıkların normalliğini inceleyelim
# 1. yol - ggplot ile
ggplot(mapping=aes(sample = resid(model1))) +
    stat_qq() +
    stat_qq_line(col="red",size=1.25) +
    theme_minimal()
```




```
# 2.yol - base R plot ile
qqnorm(resid(model1))
qqline(resid(model1))
```



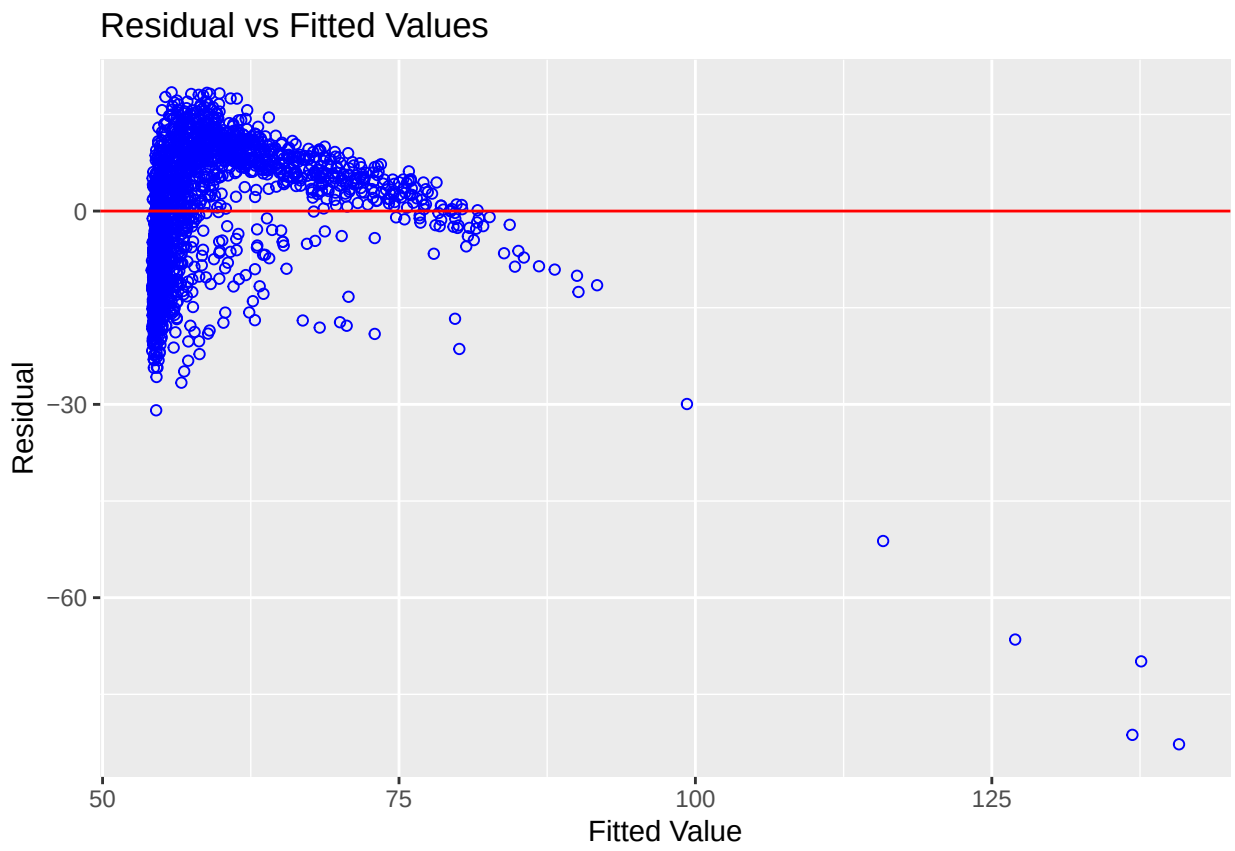

```
# 3. yol olsrr paketi ile (başka paketler de var tabii ki)
library(olsrr)

ols_plot_resid_qq(model1)
```



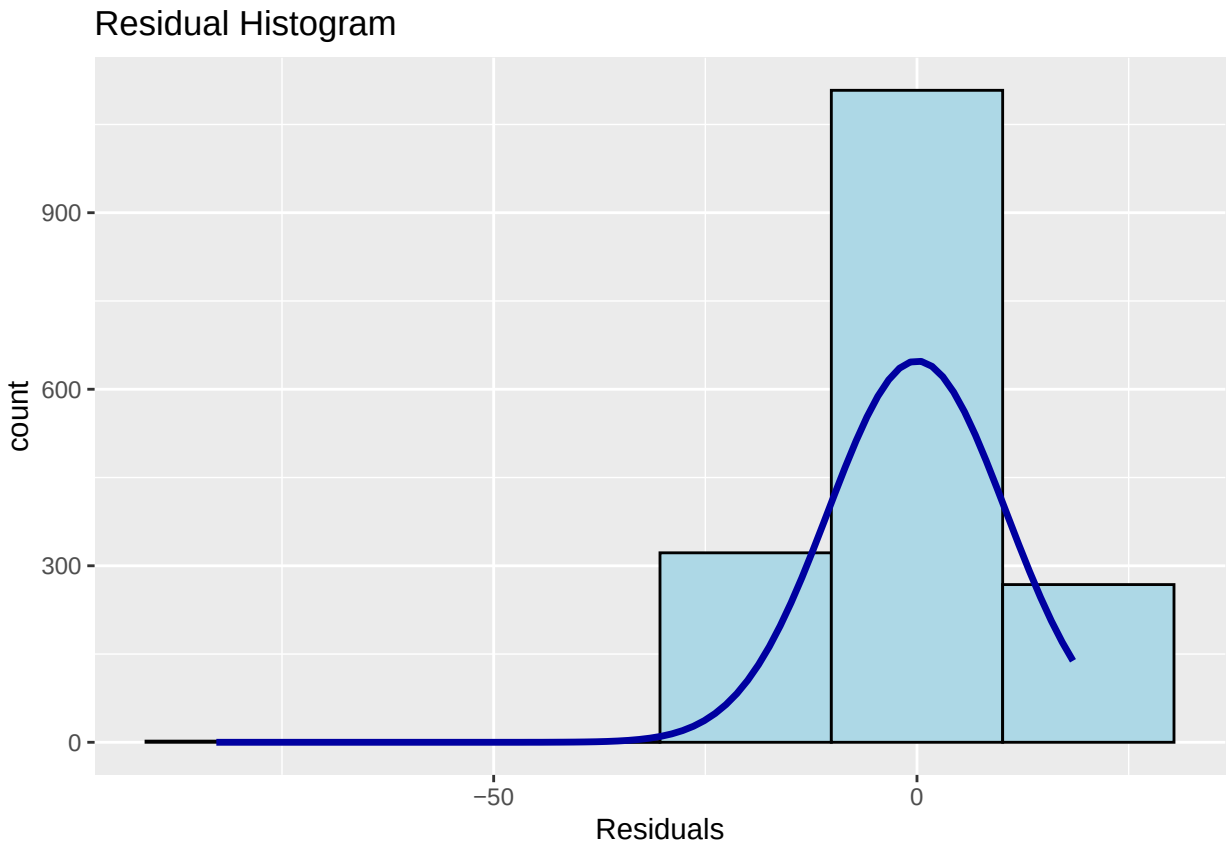

```
ols_test_normality(model1)
```

```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9202         0.0000
## Kolmogorov-Smirnov        0.0887         0.0000
## Cramer-von Mises        138.2622         0.0000
## Anderson-Darling         21.2639         0.0000
## -----------------------------------------------
```

```
ols_plot_resid_fit(model1)
```




```
ols_plot_resid_hist(model1)
```



Artıklara öncelikle bakarsak, bir noktadan sonra sapmalar olduğunu görebiliriz. Bu durum bize artıkarın normal dağılmadığı mesajını veriyor. Normallik testlerinden de bu mesajı doğrulayabiliriz.

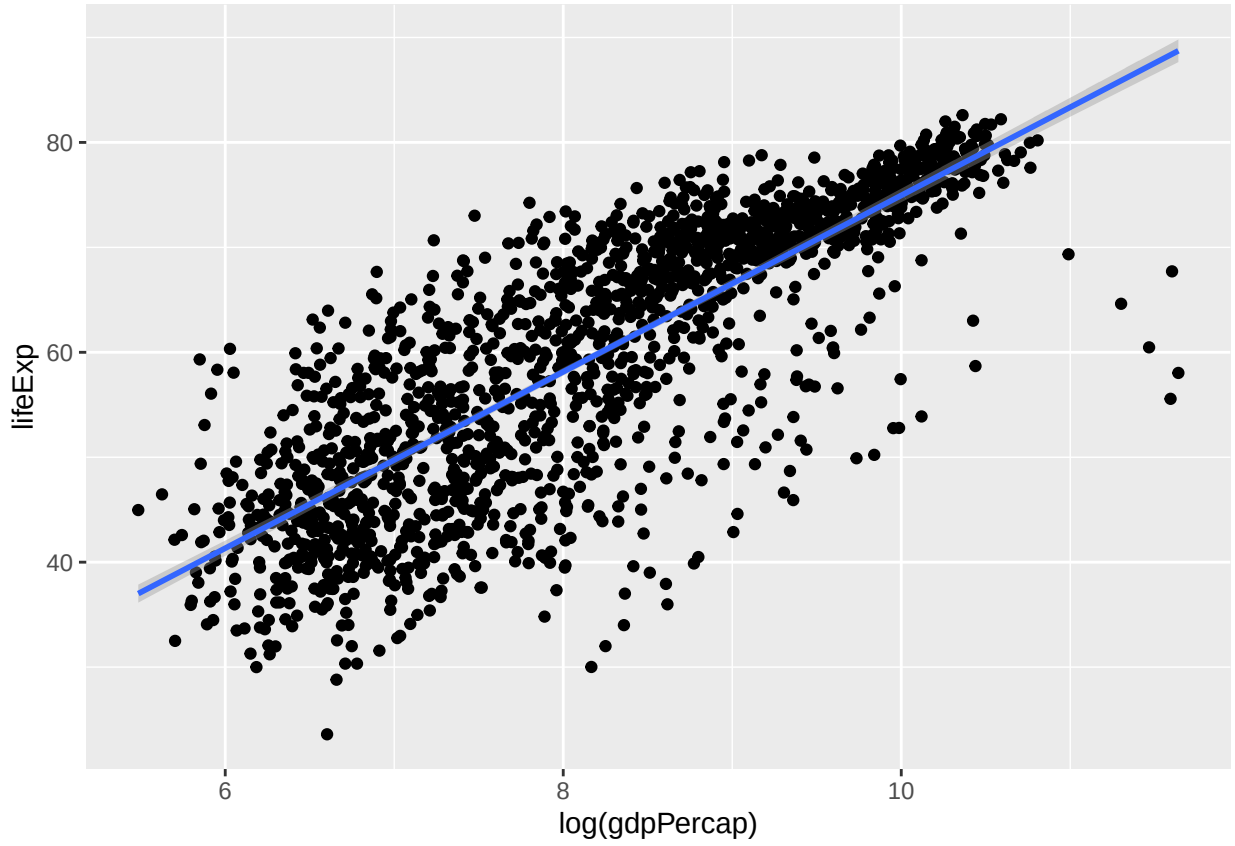
`ols_plot_resid_fit` fonksiyonu doğrusal olmamayı, eşit olmayan hata varyanslarını ve aykırı değerleri tespit etmek için y eksenindeki artıkların ve x eksenindeki kestirim değerlerinin bir dağılım grafiğini üretmektedir. Artıkların aşağıdaki şekilde davranış göstermesi beklenir.

- Artıklar, ilişkinin doğrusal olduğunu gösteren 0 çizgisi etrafında rastgele dağılır.
- Artıklar, hata varyansının homojenliğini gösteren 0 çizgisi etrafında yaklaşık bir yatay bant oluşturur.
- Hiçbir artık, aykırı değer olmadığını gösteren artıkların rastgele modelinden gözle görülür şekilde uzakta değildir.

`summary` fonksiyonu ile modelimizin verilere ne kadar iyi uyduğu hakkında biraz daha bilgi alıyoruz. Genel modelimiz ve her değişken için p-değerlerini görebiliriz. R^2 değeri, veri kümenizdeki varyansın ne kadarının modeliniz tarafından açıklanabileceğini - temel olarak, modelinizin verilere ne kadar iyi uyduğunu gösterir. Bu değer 0 ile 1 arasında değişir ve büyük olması beklenir. Genel olarak, modelinizde kaç değişken kullandığınızı telafi eden düzeltilmiş R^2'yi kullanırız - aksi halde başka bir değişken eklemek her zaman R^2'yi artırır.

Ancak GSYİH'nın logaritması alındığında değişkenlerimiz arasında çok daha normal bir doğrusal ilişki görebiliriz.

```
ggplot(gapminder, aes(log(gdpPercap), lifeExp)) +
  geom_point() +
  geom_smooth(method = "lm",se=TRUE)
```




```
model2 <- lm(lifeExp ~ log(gdpPercap), data = gapminder)
summary(model2)
```

```
##
## Call:
## lm(formula = lifeExp ~ log(gdpPercap), data = gapminder)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.778  -4.204   1.212   4.658  19.285
##
## Coefficients:
##                Estimate Std. Error t value            Pr(>|t|)
```

```
## (Intercept)     -9.1009     1.2277  -7.413    0.000000000000193 ***
## log(gdpPercap)   8.4051     0.1488  56.500 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.62 on 1702 degrees of freedom
## Multiple R-squared:  0.6522, Adjusted R-squared:  0.652
## F-statistic:  3192 on 1 and 1702 DF,  p-value: < 0.00000000000000022
```

R^2 değerimizin arttığını görebiliyoruz. İlk modelde bu değer 0,34 iken ikinci modelde 0,652 olarak bulunmuştur Bu nedenle, verilerimizi log-dönüştürmek, modelimizin verilere daha iyi uymasına yardımcı oluyor gibi görünüyor. Veri setimizdeki continent (kıta) ve year (yıl) değişkenlerini de modele ekleyerek çoklu regresyon analizi sonuçlarına bakalım.

```
model3 <- lm(lifeExp ~ log(gdpPercap) + continent + year, data = gapminder)
summary(model3)
```

```
##
## Call:
## lm(formula = lifeExp ~ log(gdpPercap) + continent + year, data = gapminder)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.0433  -3.2175   0.3482   3.6657  15.1321
##
## Coefficients:
##                       Estimate  Std. Error t value            Pr(>|t|)
## (Intercept)       -465.869597   16.674319  -27.94 <0.0000000000000002 ***
## log(gdpPercap)       5.023835    0.159473   31.50 <0.0000000000000002 ***
## continentAmericas    8.925906    0.462954   19.28 <0.0000000000000002 ***
## continentAsia        7.062939    0.395901   17.84 <0.0000000000000002 ***
## continentEurope     12.507788    0.509676   24.54 <0.0000000000000002 ***
## continentOceania    12.750719    1.274763   10.00 <0.0000000000000002 ***
## year                 0.241637    0.008586   28.14 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.813 on 1697 degrees of freedom
## Multiple R-squared:  0.7982, Adjusted R-squared:  0.7975
## F-statistic:  1119 on 6 and 1697 DF,  p-value: < 0.00000000000000022
```

Bu sonuçlara göre R^2 değeri 0.79'a yükselmiştir. değişken sayısını artırmak model başarısını artırmış görünüyor. Ayrıca katsayıların hepsinin de anlamlı çıktığı göz ardı edilmemelidir.

Afrika kıtası haricinde, veri kümemizdeki kıtaların her biri için bir satır var. Bunun sebebi Afrika kıtası referans kıta olarak burada belirlenmesinden kaynaklanmaktadır. Yani kıtalara göre verileri yorumlarken Afirika kıtasına göre değerlendirme yapılacaktır. Örneğin Avrupa'da olmak ortalama olarak, Afrika'da olmaktan 12.27 yıl daha fazla yaşam beklentisine sahip olmak anlamına gelmektedir.

Model her bağımsız değişkenin birbirinden bağımsız olduğunu varsaymasıdır. Bununla birlikte, bunun GSYİH ve kıta için doğru olmadığından oldukça emin olabiliriz - genellikle Okyanusya'daki çoğu ülkenin, örneğin Afrika'daki çoğu ülkeden daha yüksek kişi başına GSYİH'ya sahip olduğunu varsayabiliriz. Bu nedenle, bu iki değişken arasında bir etkileşim terimi eklemeliyi düşünebiliriz. Bunu, model ifademizde bu terimler arasındaki + yerine * ile değiştirerek yapabiliriz.

```
model4 <- lm(lifeExp ~ log(gdpPercap) * continent + year, data = gapminder)
summary(model4)
```

```
##
## Call:
## lm(formula = lifeExp ~ log(gdpPercap) * continent + year, data = gapminder)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.2340  -2.9548   0.1681   3.3382  14.9649
##
## Coefficients:
##                                    Estimate  Std. Error t value
## (Intercept)                      -476.845698   17.080099 -27.918
## log(gdpPercap)                      4.848360    0.268597  18.051
## continentAmericas                 -13.205551    4.646858  -2.842
## continentAsia                       4.405673    2.655124   1.659
## continentEurope                    30.952598    4.415078   7.011
## continentOceania                   76.626813   35.240823   2.174
## year                                0.247825    0.008681  28.550
## log(gdpPercap):continentAmericas    2.596704    0.555943   4.671
## log(gdpPercap):continentAsia        0.347108    0.346221   1.003
## log(gdpPercap):continentEurope     -1.934524    0.498751  -3.879
## log(gdpPercap):continentOceania    -6.487055    3.605512  -1.799
##                                             Pr(>|t|)
## (Intercept)                      < 0.0000000000000002 ***
## log(gdpPercap)                   < 0.0000000000000002 ***
## continentAmericas                             0.004539 **
## continentAsia                                 0.097239 .
## continentEurope                    0.00000000000341 ***
## continentOceania                              0.029815 *
## year                             < 0.0000000000000002 ***
```

```
## log(gdpPercap):continentAmericas     0.00000323702307 ***
## log(gdpPercap):continentAsia                 0.316216
## log(gdpPercap):continentEurope               0.000109 ***
## log(gdpPercap):continentOceania              0.072164 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.728 on 1693 degrees of freedom
## Multiple R-squared:  0.8045, Adjusted R-squared:  0.8034
## F-statistic: 696.8 on 10 and 1693 DF,  p-value: < 0.00000000000000022
```

Sonuçlar R^2 değerinin 0.80'e yükseldiğini gösteriyor. Şimdi bu modeli kullanarak yeni bir gözlem ile kestirim yapalım.

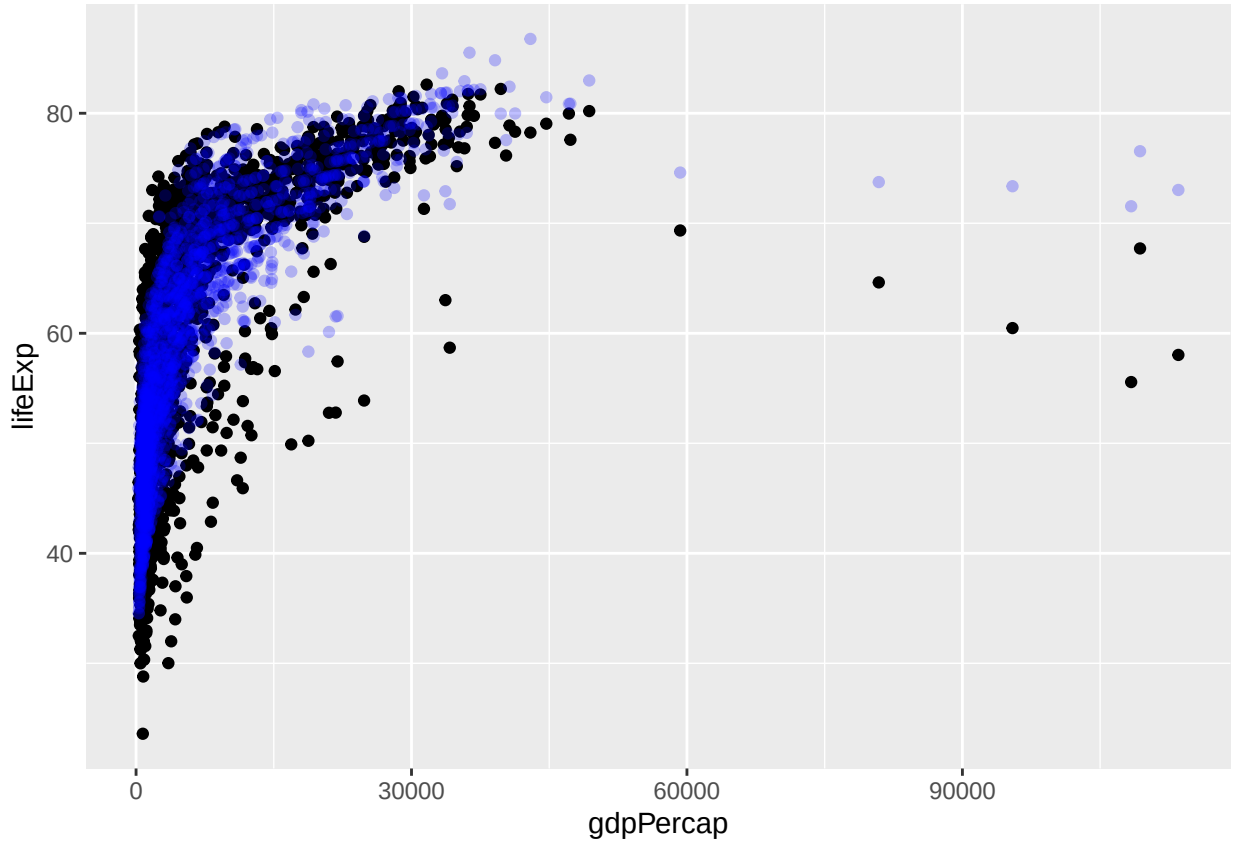
```
# yeni verilerler kestirim
gap_pred <- data.frame(lifeExp=c(70,75,80),
                       gdpPercap=c(9000,12000,15000),
                       continent=c("Asia","Americas","Europe"),
                       year=c(2012,2012,2012))

predict(model4, newdata = gap_pred,interval = "confidence", level = 0.99)
```

```
##        fit      lwr      upr
## 1 73.48759 72.35145 74.62374
## 2 78.50070 77.08816 79.91325
## 3 80.74877 79.69798 81.79956
```

```
# model tahminlerine ulaşmak
fitted <- data.frame(predict=model4$fitted.values)
head(fitted,10)
```

```
##     predict
## 1  45.90794
## 2  47.41599
## 3  48.85531
## 4  49.99046
## 5  50.59449
## 6  52.14781
## 7  54.52173
## 8  55.04663
## 9  54.87211
## 10 55.99799
```

```
# original ve kestirim sonuçlarını görselleştirelim
ggplot(gapminder, aes(gdpPercap)) +
  geom_point(aes(y = lifeExp)) +
  geom_point(aes(y = fitted$predict), color = "blue", alpha = 0.25)
```



Model sonuçları içerisinde bakılması gereken en önemli kısımlardan birisi de artıklardır. Artıklar kullanılarak modellerin başarılarını ölçen metrikler bulunmaktadır. Artıkların ortalaması ya da **RMSE**(Root mean square error) bunlardan bazılarıdır. Ayrıca **AIC**, **BIC**gibi bilgi kriterleri de model başarımlarını ölçmede yardımcı metriklerdir.

```
# Modellerin metriklerini bir araya getirelim

ME <- function(model){
  mean(residuals(model))

}

RMSE <- function(model){
  sqrt(sum(residuals(model)^2) / df.residual(model))
}
```

```
adj.R2 <- function(model){
   summary(model)$adj.r.squared
}

metrics <-
  data.frame(
    model = c("model1", "model2", "model3", "model4"),
    ME = c(ME(model1), ME(model2), ME(model3), ME(model4)),
    AIC = c(AIC(model1), AIC(model2), AIC(model3), AIC(model4)),
    adj.R2 = c(
      adj.R2(model1),
      adj.R2(model2),
      adj.R2(model3),
      adj.R2(model4)
    ),
    RMSE = c(RMSE(model1), RMSE(model2), RMSE(model3), RMSE(model4))
  )

metrics
```

```
##    model                          ME      AIC    adj.R2      RMSE
## 1 model1 -0.000000000000000073877129 12850.41 0.3403256 10.491319
## 2 model2  0.000000000000000005144947 11760.42 0.6520423  7.619535
## 3 model3  0.000000000000000055878558 10843.29 0.7974611  5.813257
## 4 model4  0.000000000000000009246565 10796.71 0.8033820  5.727655
```

Model sonuçlarının daha güzel ve temiz (tidy) bir formatta görünmesi için **broom** paketi kullanılabilir.

```
library(broom)
# Katsayılar düzeyinde sonuçlar
tidy(model4)
```

```
## # A tibble: 11 x 5
##    term                estimate std.error statistic   p.value
##    <chr>                  <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)          -477.      17.1        -27.9  2.07e-141
##  2 log(gdpPercap)          4.85     0.269       18.1  9.42e- 67
##  3 continentAmericas     -13.2      4.65        -2.84 4.54e-  3
##  4 continentAsia           4.41     2.66         1.66 9.72e-  2
##  5 continentEurope        31.0      4.42         7.01 3.41e- 12
##  6 continentOceania       76.6     35.2          2.17 2.98e-  2
##  7 year                    0.248    0.00868     28.5  1.12e-146
```

```
##  8 log(gdpPercap):continentAmericas    2.60    0.556        4.67 3.24e-  6
##  9 log(gdpPercap):continentAsia        0.347   0.346        1.00 3.16e-  1
## 10 log(gdpPercap):continentEurope     -1.93    0.499       -3.88 1.09e-  4
## 11 log(gdpPercap):continentOceania    -6.49    3.61        -1.80 7.22e-  2
```

```
#model düzeyinde sonuçlar
glance(model4)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df logLik    AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>  <dbl>  <dbl>  <dbl>
## 1     0.805         0.803  5.73      697.       0    10 -5386. 10797. 10862.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
# gözlem düzeyinde sonuçlar
augment(model4)
```

```
## # A tibble: 1,704 x 9
##    lifeExp `log(gdpPercap)` continent  year .fitted    .hat .sigma .cooksd
##      <dbl>            <dbl> <fct>     <int>   <dbl>   <dbl>  <dbl>   <dbl>
##  1    28.8             6.66 Asia       1952    45.9 0.00654   5.71 0.00537
##  2    30.3             6.71 Asia       1957    47.4 0.00591   5.71 0.00483
##  3    32.0             6.75 Asia       1962    48.9 0.00544   5.71 0.00433
##  4    34.0             6.73 Asia       1967    50.0 0.00530   5.72 0.00378
##  5    36.1             6.61 Asia       1972    50.6 0.00570   5.72 0.00337
##  6    38.4             6.67 Asia       1977    52.1 0.00547   5.72 0.00288
##  7    39.9             6.89 Asia       1982    54.5 0.00474   5.72 0.00285
##  8    40.8             6.75 Asia       1987    55.0 0.00552   5.72 0.00313
##  9    41.7             6.48 Asia       1992    54.9 0.00715   5.72 0.00350
## 10    41.8             6.45 Asia       1997    56.0 0.00777   5.72 0.00443
## # ... with 1,694 more rows, and 1 more variable: .std.resid <dbl>
```