



# Varsayımlar



## Normallik



Dr. Kübra Atalay Kabasakal  
Bahar 2023

# ilk olarak veriyi okuma

```
library(haven)
screen <- read_sav("SCREEN.sav")
head(screen)
```

```
## # A tibble: 6 × 7
##   subno timedrs attdrug atthouse income mstatus race
##   <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl> <dbl>
## 1     1       1       8      27       5       2       1
## 2     2       3       7      20       6       2       1
## 3     3       0       8      23       3       2       1
## 4     4      13       9      28       8       2       1
## 5     5      15       7      24       1       2       1
## 6     6       3       8      25       4       2       1
```

# Eksik veri ile baş etme

```
screen <-
  screen %>%
  mutate(income = ifelse(is.na(income),
  mean(income, na.rm =TRUE), income)) %>% #ortalama atama
  na.omit() #liste bazında silme
summary(screen)
```

```
##      subno      timedrs      attdrug      atthouse
## Min.   : 1.0   Min.   : 0.000   Min.   : 5.00   Min.   : 2.00
## 1st Qu.:136.8  1st Qu.: 2.000   1st Qu.: 7.00   1st Qu.:21.00
## Median :313.5  Median : 4.000   Median : 8.00   Median :24.00
## Mean    :317.3  Mean    : 7.914   Mean    : 7.69   Mean    :23.54
## 3rd Qu.:483.2  3rd Qu.:10.000   3rd Qu.: 9.00   3rd Qu.:27.00
## Max.    :758.0  Max.    :81.000   Max.    :10.00   Max.    :35.00
##      income      mstatus      race
## Min.   : 1.000   Min.   :1.00   Min.   :1.000
## 1st Qu.: 3.000   1st Qu.:2.00   1st Qu.:1.000
## Median : 4.000   Median :2.00   Median :1.000
## Mean    : 4.208   Mean    :1.78   Mean    :1.086
## 3rd Qu.: 6.000   3rd Qu.:2.00   3rd Qu.:1.000
## Max.    :10.000  Max.    :2.00   Max.    :2.000
```

# Veri setindeki kayıp veriler

- `atthouse` değişkeninde bir kayıp değer bulunmaktadır ve **liste bazında silme yöntemi** ile veri setinden çıkarılmıştır.
- Veri setinde **income** değişkeni 26 kayıp değere sahiptir ve bu sayı örneklemi  $\%5$ 'inden fazladır. Eğer bu değişken araştırma açısından öneme sahip değilse, veri setinden çıkarılabilir, aksi halde kayıp verinin tahmin edilmesi yöntemlerinden biri kullanılabilir.
- **income** değişkenindeki kayıp değerler için kayıp verinin tahmin edilmesi yöntemlerinden ortalamanın yerleştirilmesi kullanılarak kayıp değer **yerine değişkenin ortalama değeri (4.21 değeri) yerleştirilmiştir**.

## Uç değerler

- Uç değerler hem I. tip hem de II. tip hatalara neden olurlar ve sonuçların genellenebilirliğini düşürürler.
- Veri setinde uç değer bulunmasının 4 nedeni olabilir:
  - Verinin veri dosyasına **yanlış girilmesi**
  - **Kayıp veri kodlamasında hata yapılması**
  - Uç değerin örneklemin alındığı **evrenin üyesi olmaması**
  - Uç değerin örneklemin alındığı evrenin üyesi olması ancak değişkenin evrendeki dağılımının normal dağılıma göre **aşırı değerlere sahip olması**
- Hatalı veri girişi ve kayıp değer kodlaması kolaylıkla bulunup düzeltilebilir ancak 3. ve 4. durumlar arasında ayırm yapıp uç değerin veri setinden silinip silinmemesine karar vermek oldukça güçtür

## Uç değerlerin Belirlenmesi

- Tek değişkenli uç değerlerin belirlenmesi çok değişkenli uç değerlerin belirlenmesine göre daha kolaydır.
- İki kategorili değişkenler için, **eşit büyüklükte olmayan kategorilerde yanlış kategoride gözlenen bir değer olasılıkla uç değerdir.**
- Rummel (1970) iki kategorili bir değişken için kategorilerden biri örneklemdeki bireylerin **%90'ını diğer ise %10'unu içeriyorsa, değişkenin analiz dışı bırakılmasını önermektedir.**

# Uç değerlerin Belirlenmesi

```
library(summarytools)
freq(screen$mstatus) %>%
  kable(format='markdown',
    caption="Frekans Tablosu", digits = 2)
```

Table: Frekans Tablosu

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
1	102	21.98	21.98	21.98	21.98
2	362	78.02	100.00	78.02	100.00
	0	NA	NA	0.00	100.00
Total	464	100.00	100.00	100.00	100.00

# Uç değerlerin Belirlenmesi

```
freq(screen$race) %>%  
  kable(format='markdown',  
        caption="Frekans Tablosu",digits = 2)
```

Table: Frekans Tablosu

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
1	424	91.38	91.38	91.38	91.38
2	40	8.62	100.00	8.62	100.00
	0	NA	NA	0.00	100.00
Total	464	100.00	100.00	100.00	100.00

## Uç değerlerin Belirlenmesi

- İki kategorili değişkenlerden **race** değişkeninin kategorilere dağılımları incelendiğinde kategoriler arasında yaklaşık **91/9 (10.1:1)** oranı olduğu görülmektedir. Bu **oran oldukça yüksektir**. Değişken **araştırma için önemli değilse çıkarılabilir**, aksi halde değişkenle ilgili sonuçlar yorumlanırken bu durum göz önüne alınmalıdır.
- İki kategorili değişkenlerden **mstatus** değişkeninin kategorilere dağılımları incelendiğinde kategoriler arasında yaklaşık **78/22 (3.5:1)** oranı olduğu görülmektedir. Bu oran kabul edilebilir bir orandır.

- 
-  [summarytools](#) için

## Uç değerlerin Belirlenmesi

- Sürekli değişkenler için tek değişkenli uç değerleri belirlemenin bir yolu, değişkene ait bütün değerlerin ortalama 0, standart sapma 1 olacak şekilde standart değerlere (z puanlarına) dönüştürülmesidir. Tek değişkenli **uç değerler çok büyük z puanlarına sahiptirler.**
- Örneklem büyüklüğü 100 veya daha az olduğunda, eğer herhangi bir gözlemin **z puanı  $\pm 3.0$  veya daha fazlaysa, gözlem uç değerdir.**
- Örneklem büyüklüğü 100'den fazla olduğunda, eğer herhangi bir gözlemin **z puanı  $\pm 4.0$  veya daha fazlaysa, gözlem uç değerdir.**
  - Bu yöntem eşit aralık veya eşit oran düzeyinde ölçülen değişkenler için veya sürekli değişken olarak ele alınan sıralama ölçüğünde ölçülen değişkenler için geçerli olup sınıflama düzeyinde ölçülen değişkenler için geçerli değildir.

# Uç değerlerin Belirlenmesi

```
library(outliers)
library(knitr)
z.scores <- screen %>%
  na.omit() %>%
  select(2:5) %>%
  scores(type = "z") %>%
  round(2)
```

```
summarytools::descr(z.scores,
  stats      = c("min", "max"),
  transpose  = TRUE,
  headings   = FALSE) %>%
  kable()
```

	Min	Max
attdrug	-2.33	2.00
atthouse	-4.80	2.56
income	-1.36	2.46
timedrs	-0.72	6.67

# Uç değerlerin Belirlenmesi

- **timedrs** değişkeni için z puanlarının maksimum değerin 4.0'ten büyük olduğu,**atthouse** değişkeni z puanlarının içinse minimum değerin -4.0'ten küçük olduğu görülmektedir. Diğer değişkenler için değerler beklenen sınırlar içerisinde

```
DT:::datatable(z.scores)
```

Show 10 entries

Search:

	timedrs	attdrug	atthouse	income
1	-0.63	0.27	0.77	0.34
2	-0.45	-0.6	-0.79	0.76
3	-0.72	0.27	-0.12	-0.51
4	0.46	1.13	0.99	1.61
5	0.65	-0.6	0.1	-1.36
6	-0.45	0.27	0.33	-0.09
7	-0.54	-0.6	1.44	0.76
8	-0.72	-0.6	0.1	0.76
9	-0.08	-0.6	-0.79	-0.94

## Uç değerlerin Belirlenmesi

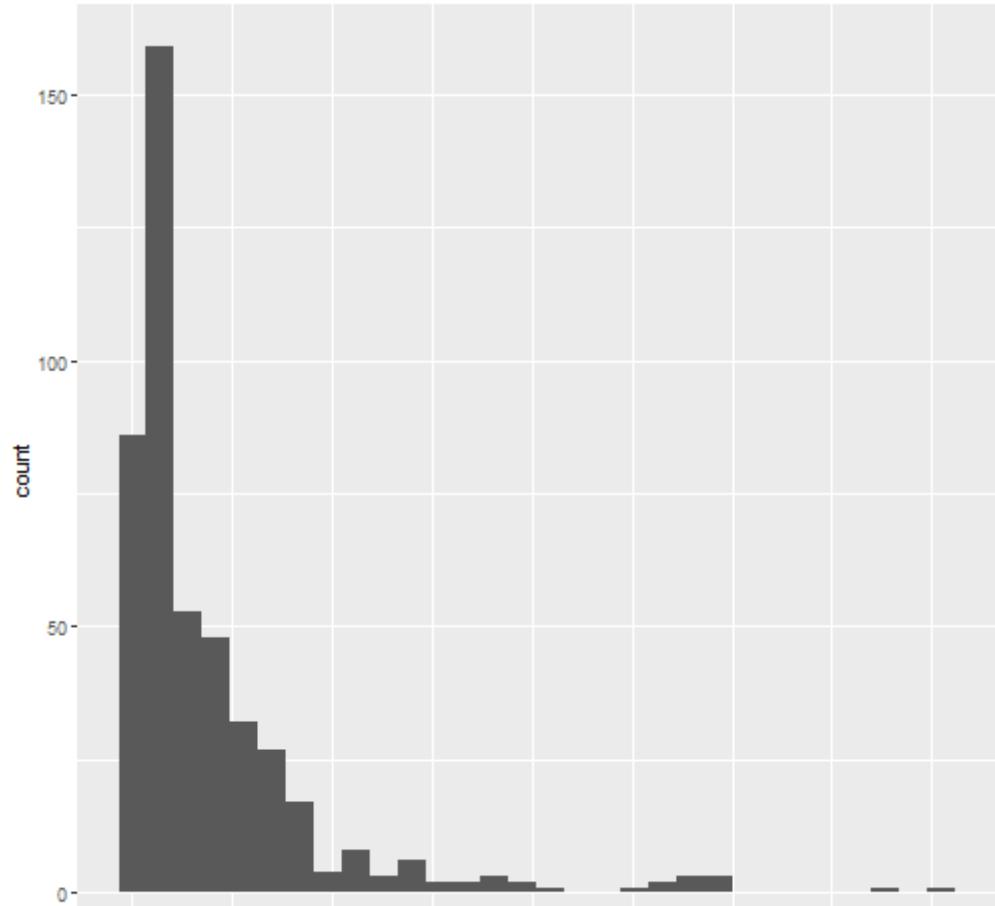
- Tek değişkenli uç değerleri saptamak için grafiksel yöntemlerden de yararlanılabilir (Örneğin, histogramlar, kutu grafikleri, normal olasılık grafikleri gibi).
- **Histogramlar** kolay anlaşılan ve yorumlanan grafiklerdir ve uç değerlerin belirlenmesine yardımcı olabilirler. Genellikle ortalamanın yakınındaki çoğu gözlemle birlikte ortalamanın iki yönüne doğru uzanan gözlemler vardır. **Uç değer dağılımının geri kalanıyla bağlantısı bulunmayan gözlemdir.**
- **Kutu** grafikleri de basittir. Medyan etrafındaki gözlemler kutu içine alınır. **Kutudan çok uzağa düşen gözlemler uç değerdir.**
- Normal **olasılık grafikleri** değişkenlerin dağılımlarının normalliliğinin değerlendirilmesinde oldukça kullanışlıdır. **Uç değerler de bu grafiklerde gözlenebilir; diğerlerinden önemli derecede uzakta bulunan nokta uçdeğerdir.**

# Uç değerlerin Belirlenmesi

```
library(ggplot2)
timedrs_plt <-
ggplot(screen) +aes(x = timedrs) +
  geom_histogram()
```

timedrs\_plt

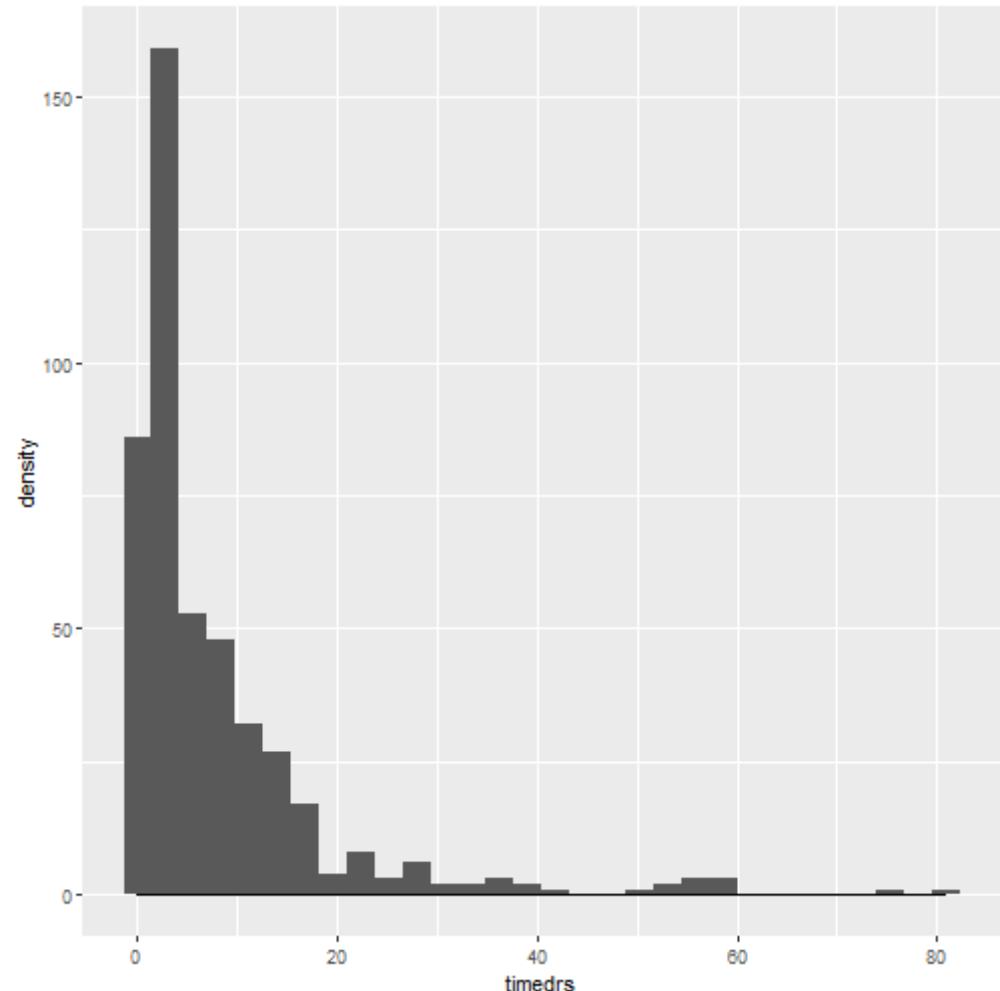
## `stat\_bin()` using `bins = 30`. Pick better value



# Uç değerlerin Belirlenmesi

```
library(ggplot2)
timedrs_plt <-
timedrs_plt +
geom_density(fill = "#0c4c8a",alpha = 0.5)
```

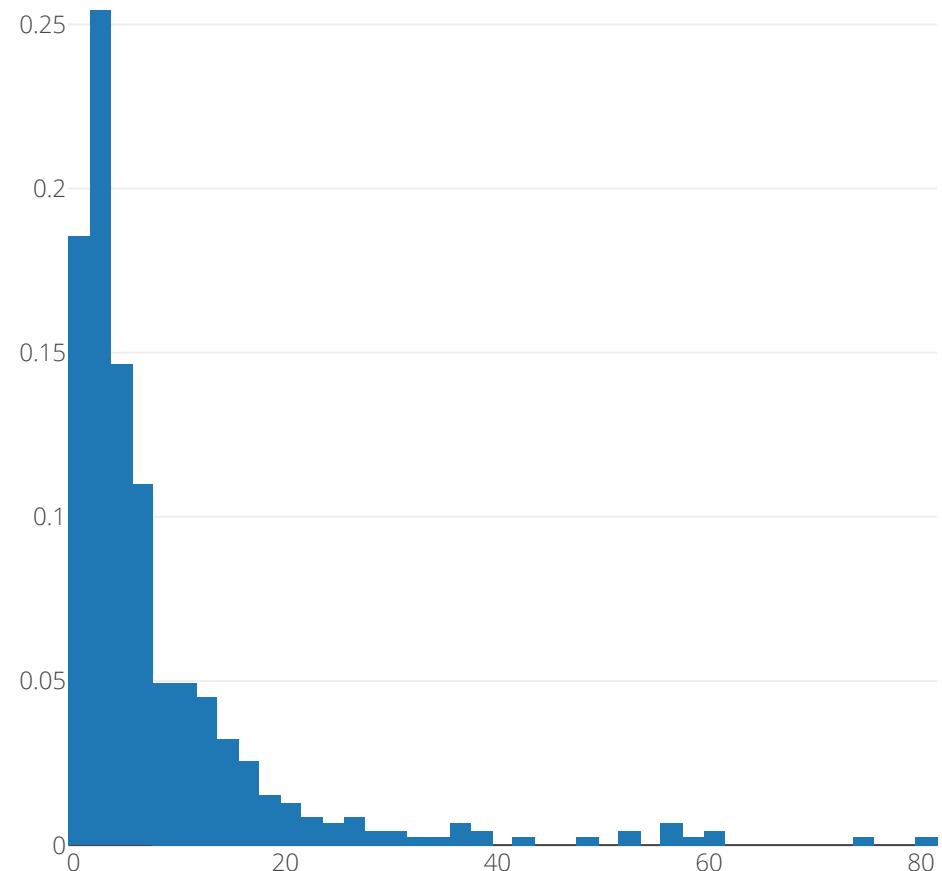
timedrs\_plt



# Uç değerlerin Belirlenmesi

```
library(plotly)
timedrs_plt <-
plot_ly(x = screen$timedrs,
type = "histogram",
histnorm = "probability")
```

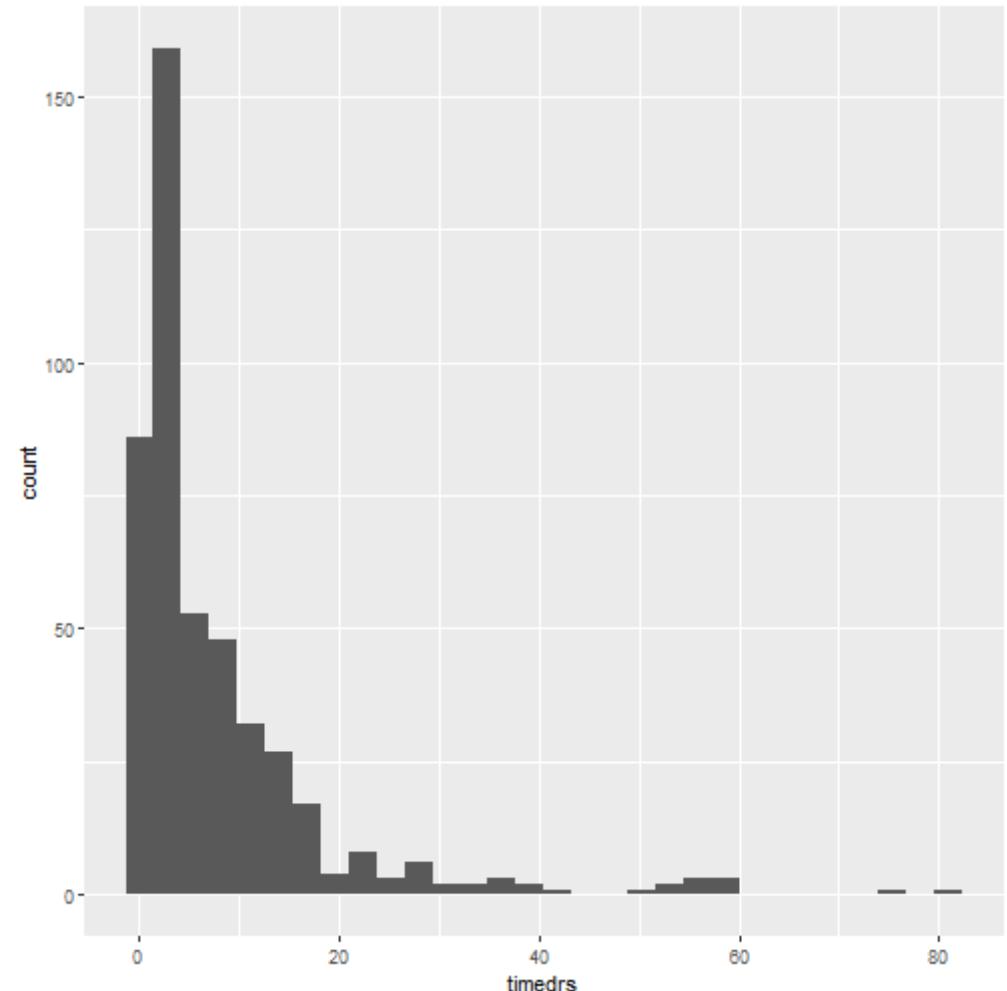
timedrs\_plt



# Uç değerlerin Belirlenmesi

```
library(ggpmisc)
timedrs_plt <-
ggplot(data = screen, aes(x = timedrs)) +
geom_histogram(bins = 30)
```

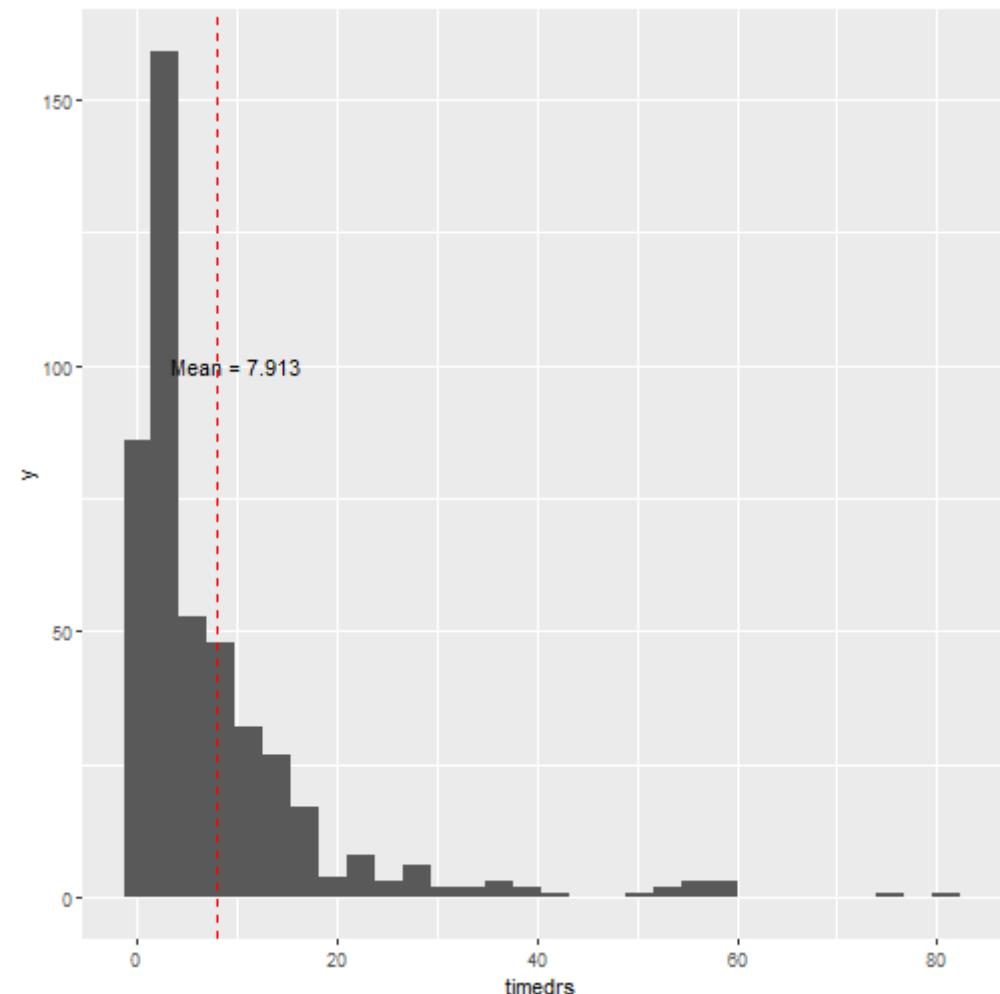
timedrs\_plt



# Uç değerlerin Belirlenmesi

```
library(ggpmisc)
timedrs_plt <-
  timedrs_plt +
  geom_vline(xintercept =7.914,
             color = "red",
             linetype = "dashed") +
  annotate("text",
           label = "Mean = 7.913",
           x = 10, y = 100,
           color = "black")
```

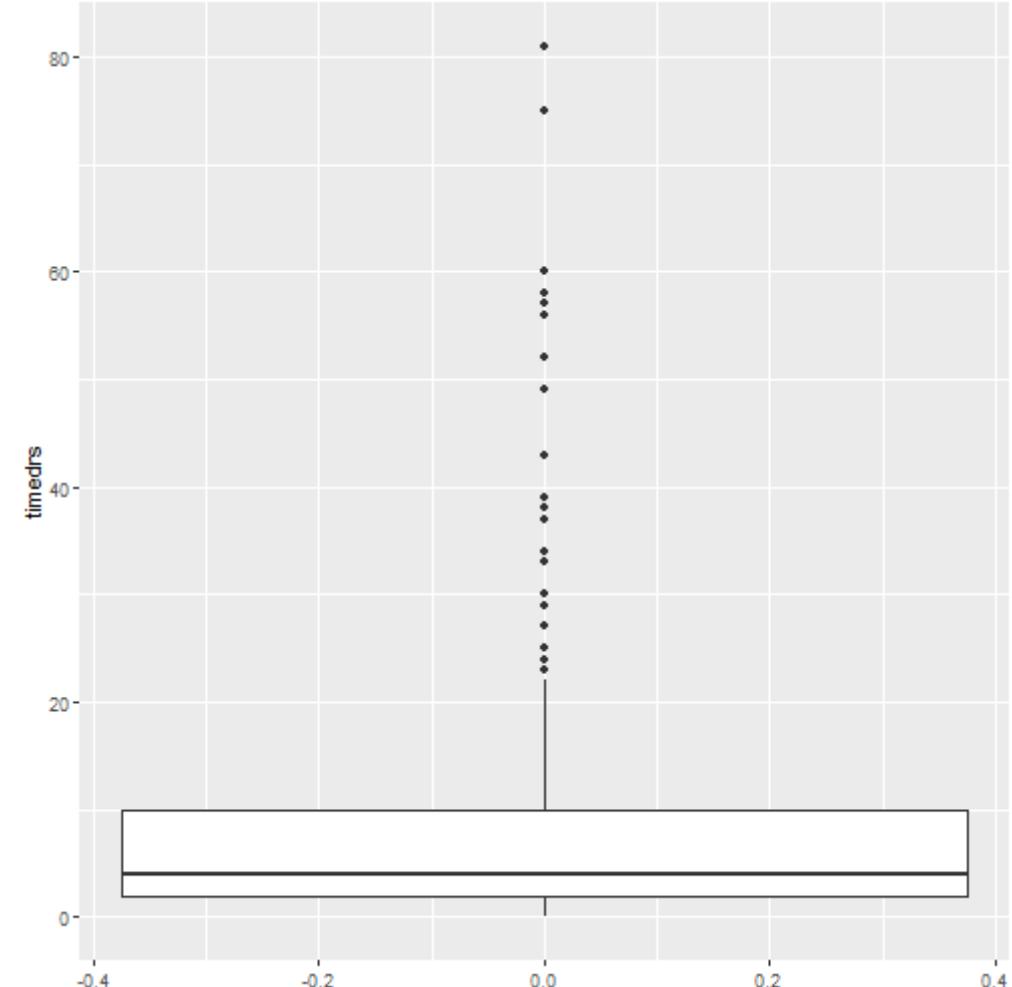
timedrs\_plt



# Uç değerlerin Belirlenmesi

```
timedrs_plt <-  
ggplot(screen, aes(y = timedrs)) +  
  geom_boxplot()
```

```
timedrs_plt
```



# Uç değerlerin Belirlenmesi

```
boxplot.stats(screen$timedrs)$out
```

```
## [1] 60 23 39 33 38 34 27 30 25 49 60 27 27 52 24 57 52 58 57 43 37 75 29
## [24] 30 25 37 56 29 37 81 27 23
```

```
out <- boxplot.stats(screen$timedrs)$out
out
```

```
## [1] 60 23 39 33 38 34 27 30 25 49 60 27 27 52 24 57 52 58 57 43 37 75 29
## [24] 30 25 37 56 29 37 81 27 23
```

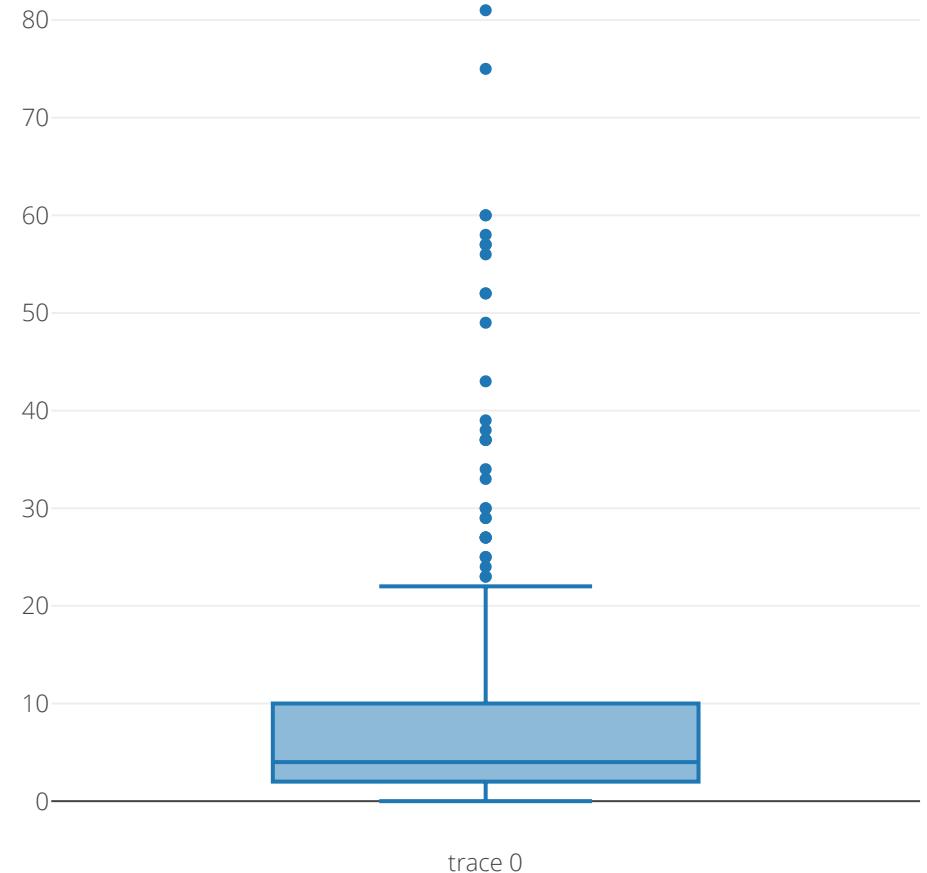
```
out_ind <- which(screen$timedrs %in% c(out))
out_ind
```

```
## [1] 40 64 67 76 79 96 102 117 150 163 168 170 178 193 203 206 213
## [18] 249 274 278 285 289 309 342 344 362 367 374 388 404 408 443
```

# Uç değerlerin Belirlenmesi

```
library(plotly)  
  
timedrs_plt <-  
plot_ly(y = screen$timedrs, type = 'box')
```

timedrs\_plt

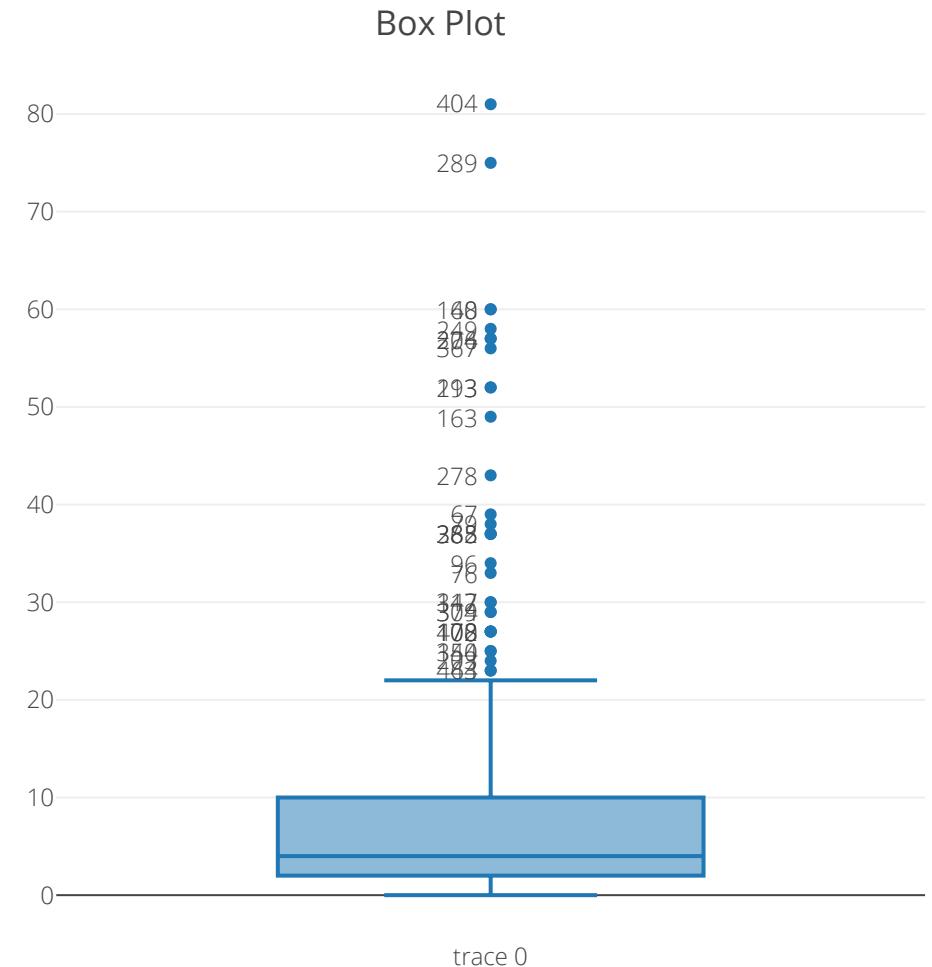


# Uç değerlerin Belirlenmesi

```
library(plotly)

timedrs_plt <-
timedrs_plt %>%
layout(title = 'Box Plot',
annotations = list(
x = -0.01,
y = boxplot.stats(screen$timedrs)$out,
text = paste(out_ind),
showarrow = FALSE,
xanchor = "right"
)
)
```

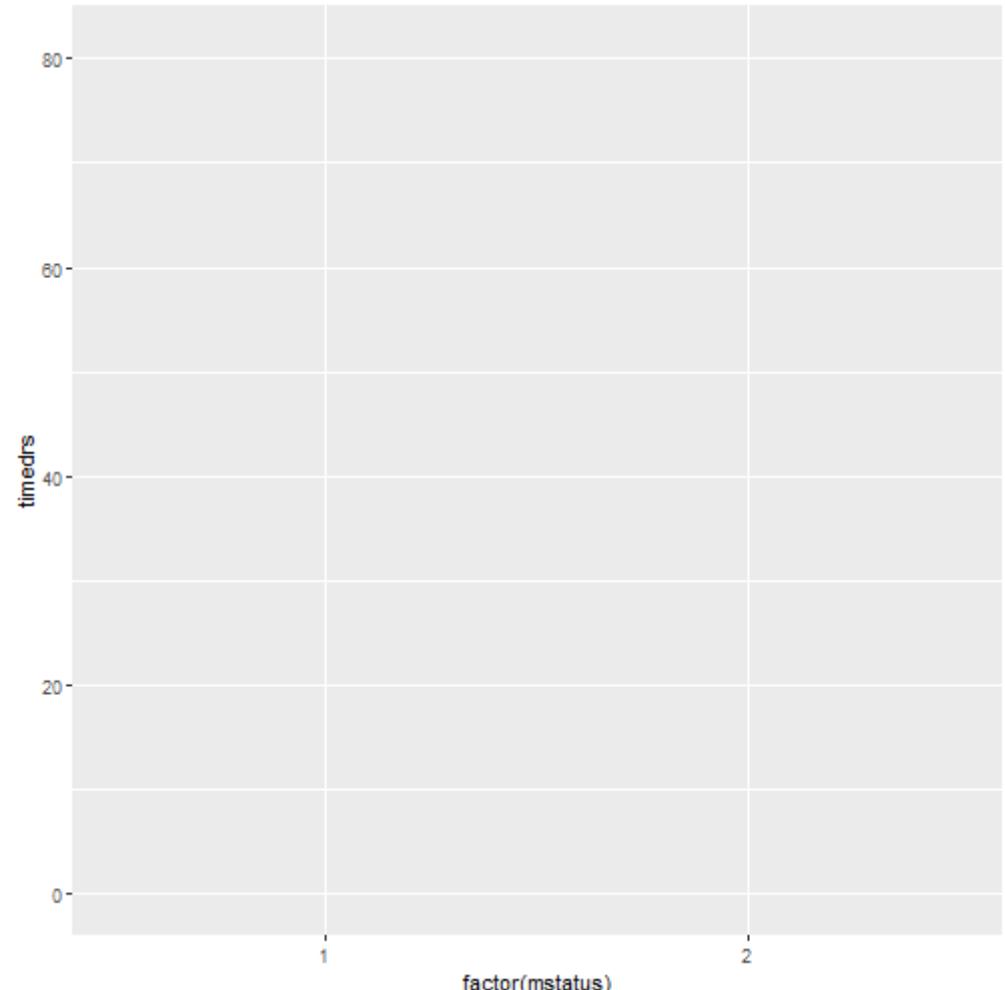
timedrs\_plt



# Uç değerlerin Belirlenmesi

```
timedrs_plt <-  
ggplot(screen,  
aes(x = factor(mstatus),  
y = timedrs,  
fill = factor(mstatus)))
```

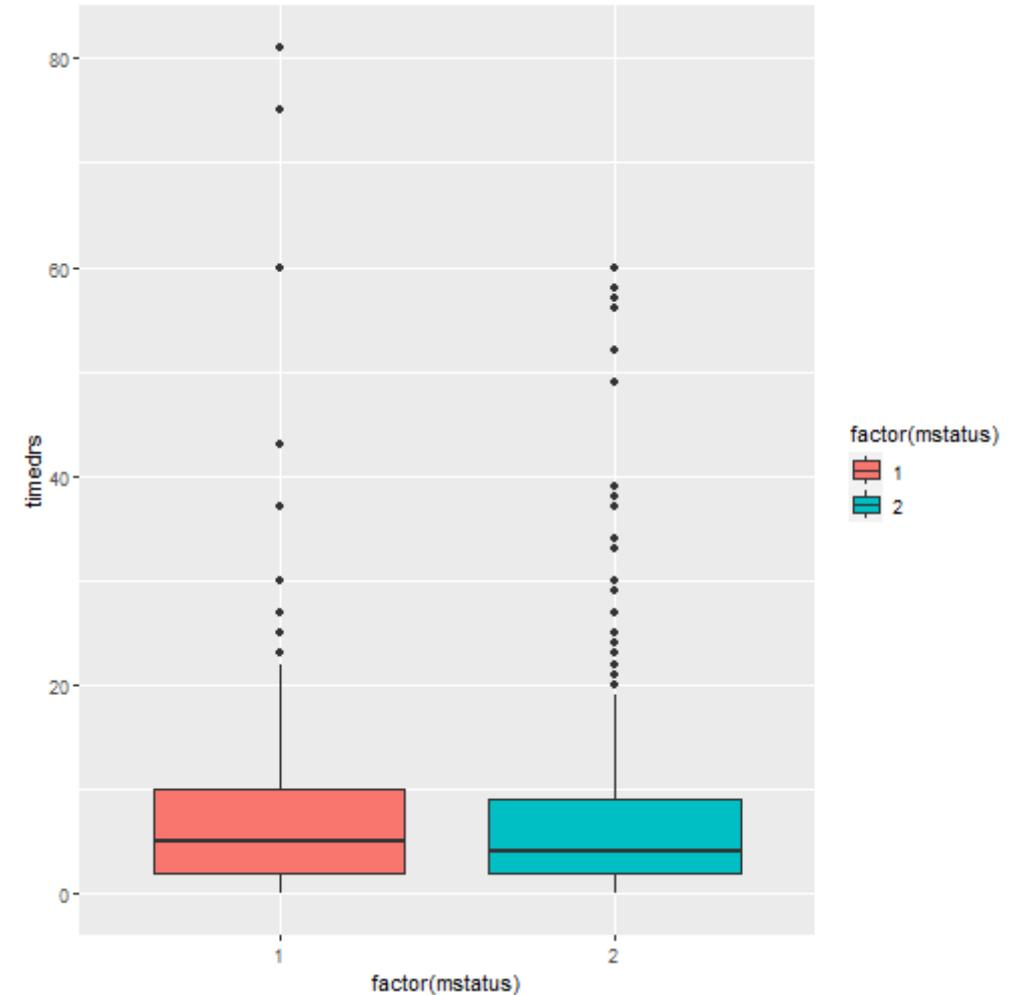
timedrs\_plt



# Uç değerlerin Belirlenmesi

```
timedrs_plt <-  
timedrs_plt+  
geom_boxplot()
```

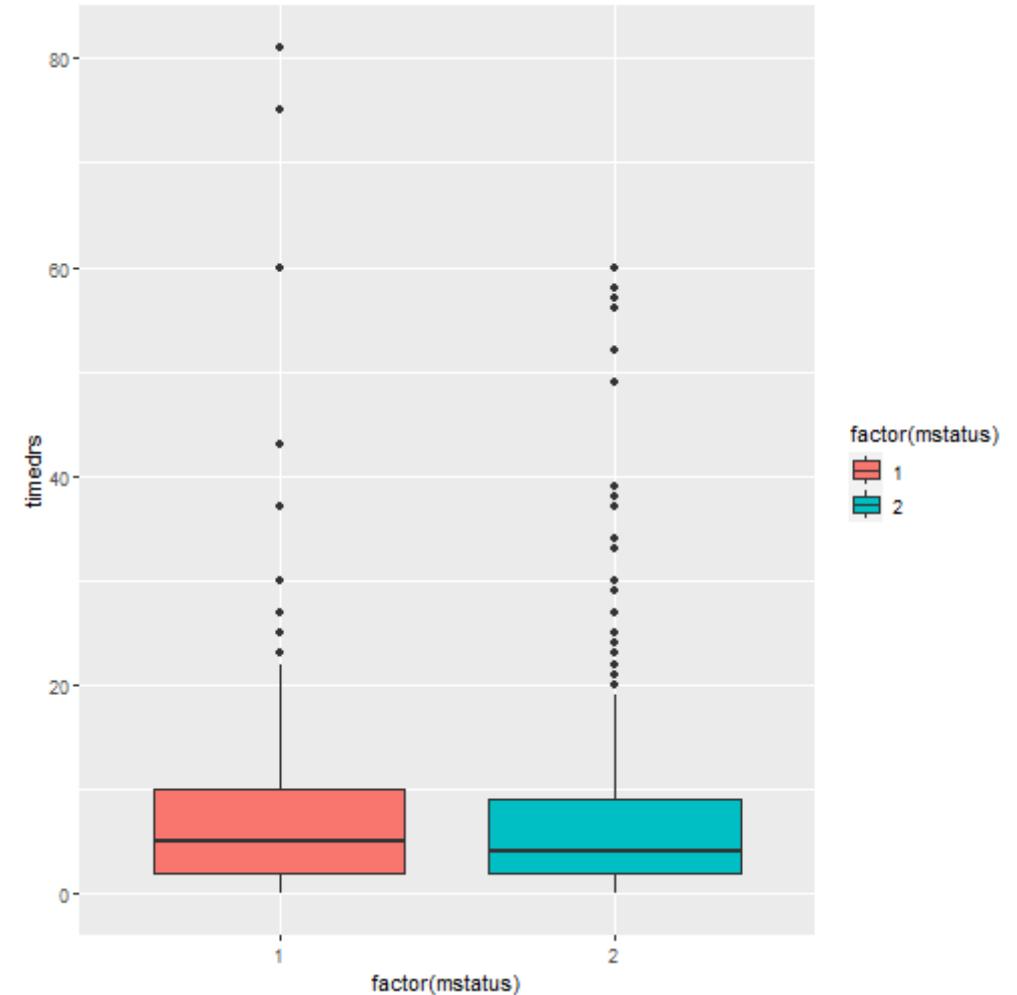
```
timedrs_plt
```



# Uç değerlerin Belirlenmesi

```
timedrs_plt <-  
timedrs_plt+  
geom_boxplot()
```

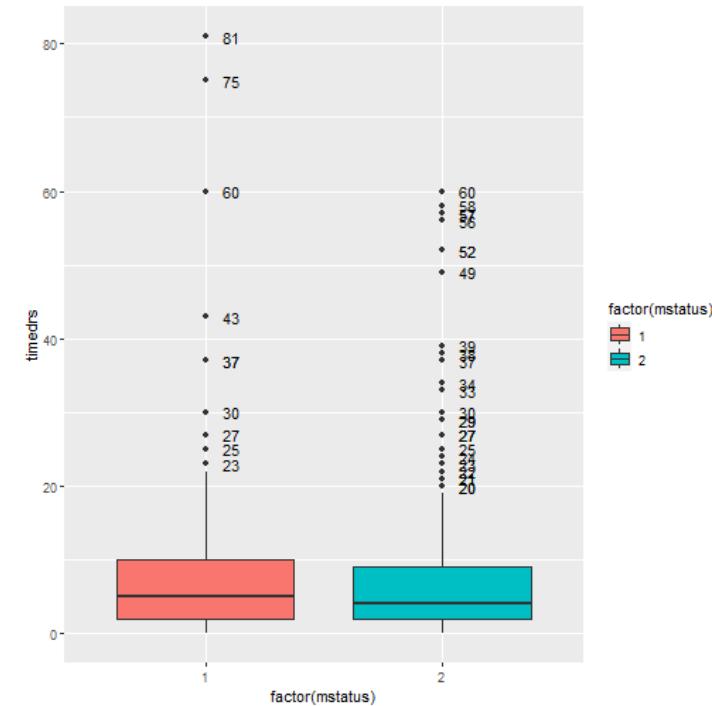
```
timedrs_plt
```



# Uç değerlerin Belirlenmesi

```
timedrs_plt <-
timedrs_plt +
stat_summary(
aes(label = round(stat(y), 1)),
geom = "text",
fun = function(y)
{o <- boxplot.stats(y)$out;
if(length(o) == 0) NA else o },
hjust = -1
)
```

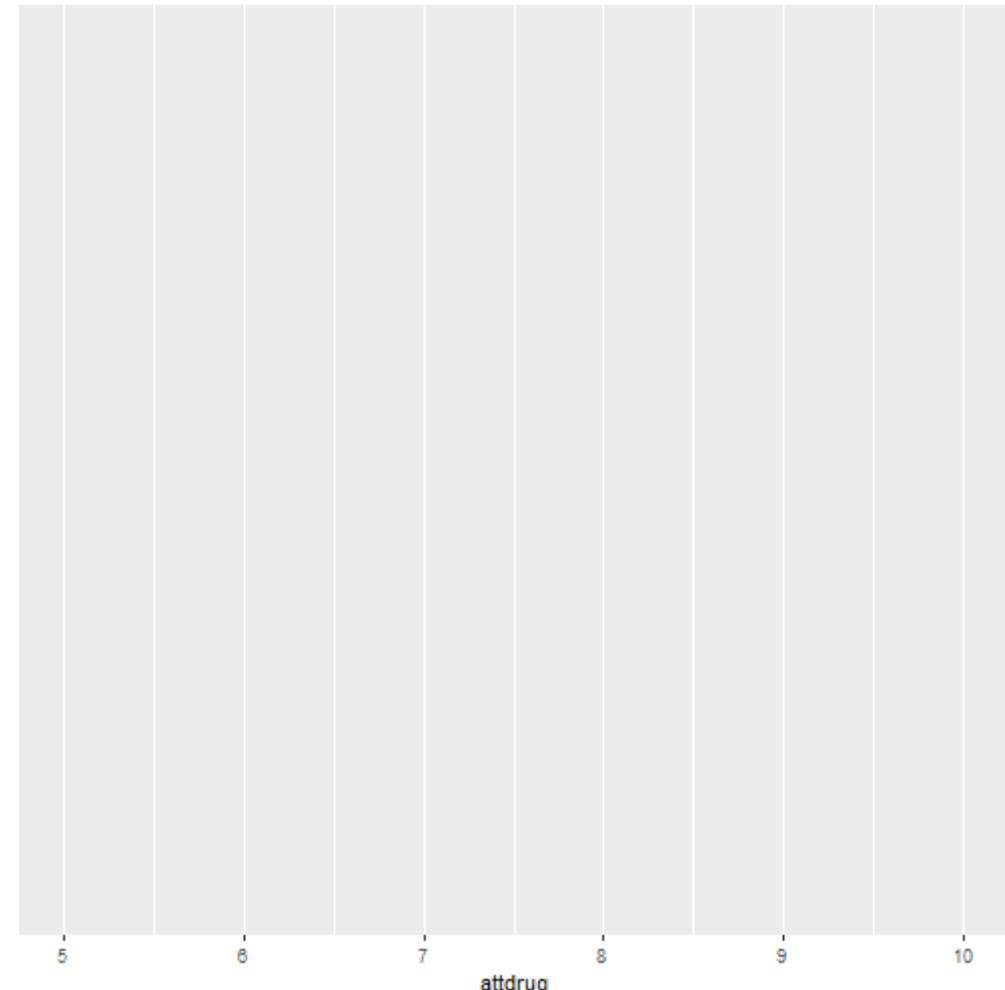
timedrs\_plt



# Uç değerlerin Belirlenmesi

```
attdrug_plt <-  
ggplot(screen) +  
aes(x = attdrug)
```

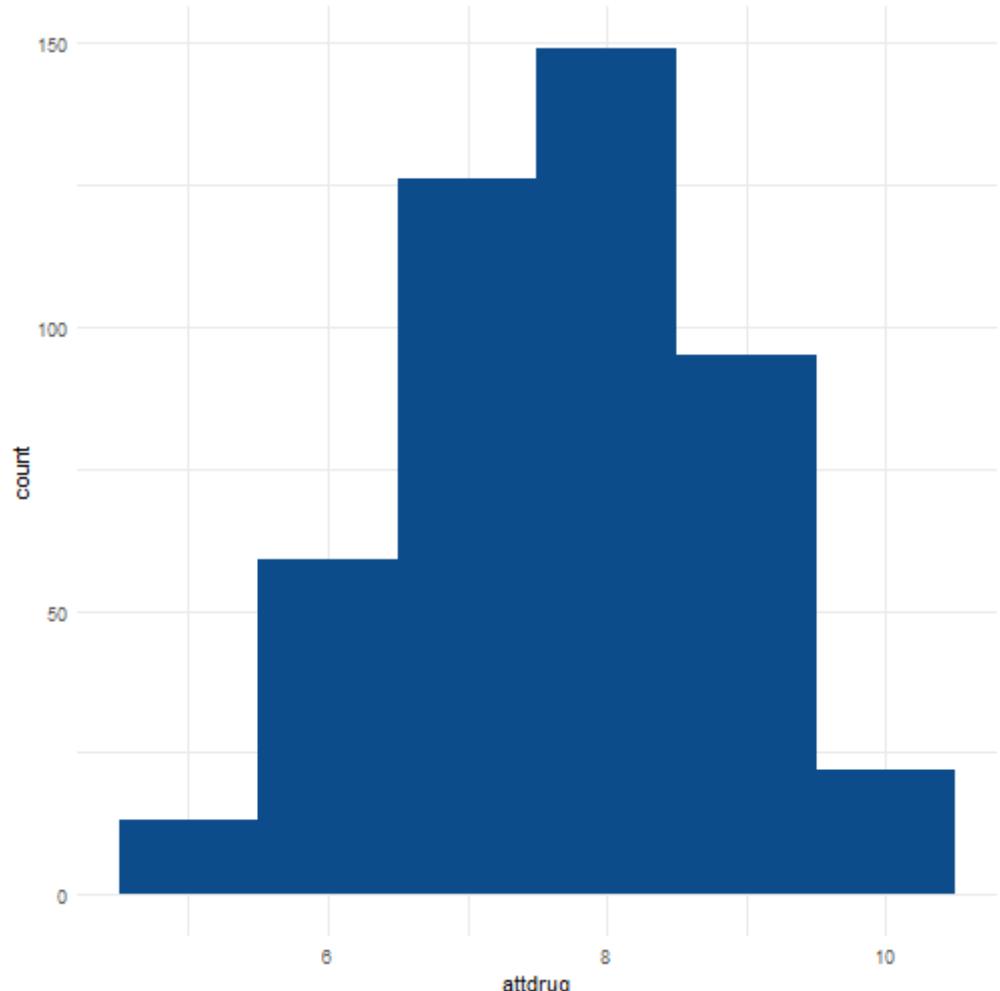
```
attdrug_plt
```



# Uç değerlerin Belirlenmesi

```
attdrug_plt <-  
attdrug_plt +  
geom_histogram( bins = 6, fill = "#0c4c8a") +  
theme_minimal()
```

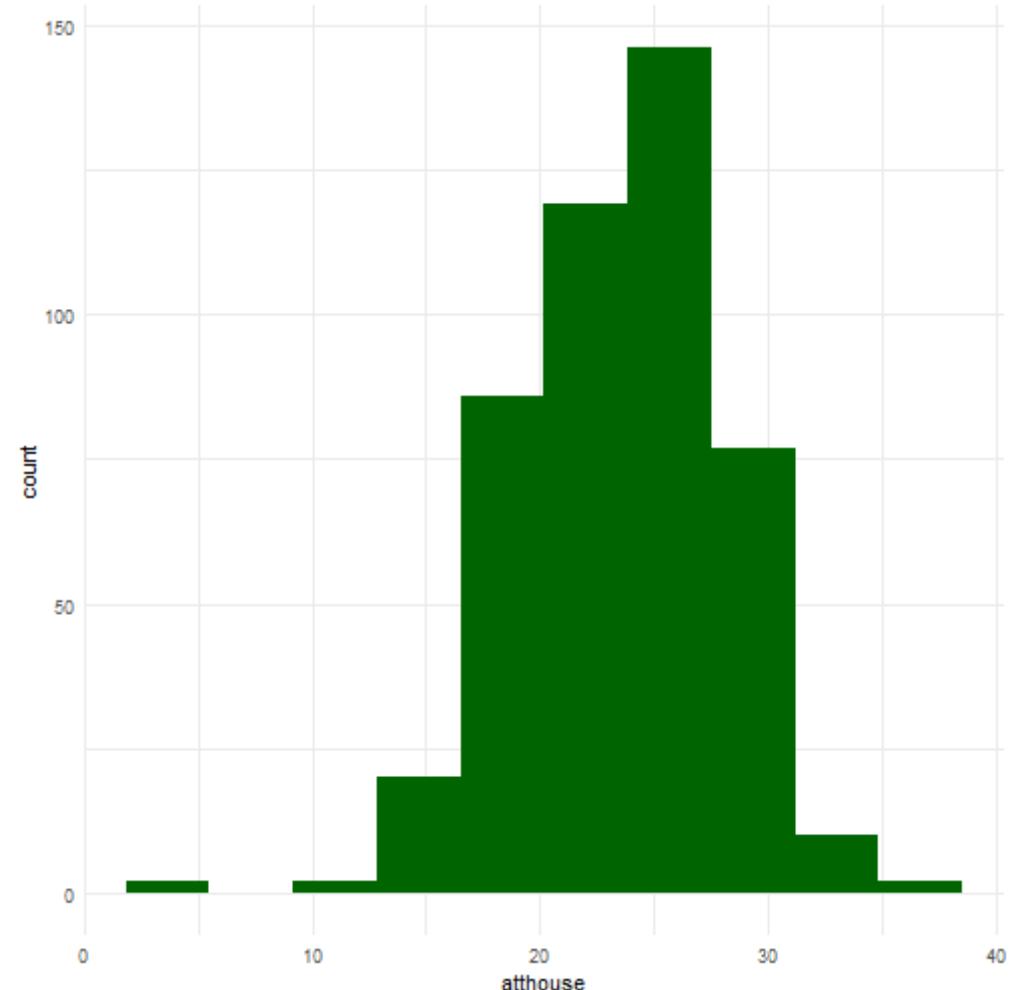
```
attdrug_plt
```



# Uç değerlerin Belirlenmesi

```
atthouse_plt <-
ggplot(screen) +
aes(x = atthouse) +
geom_histogram( bins = 10, fill = "darkgreen")
theme_minimal()
```

```
atthouse_plt
```

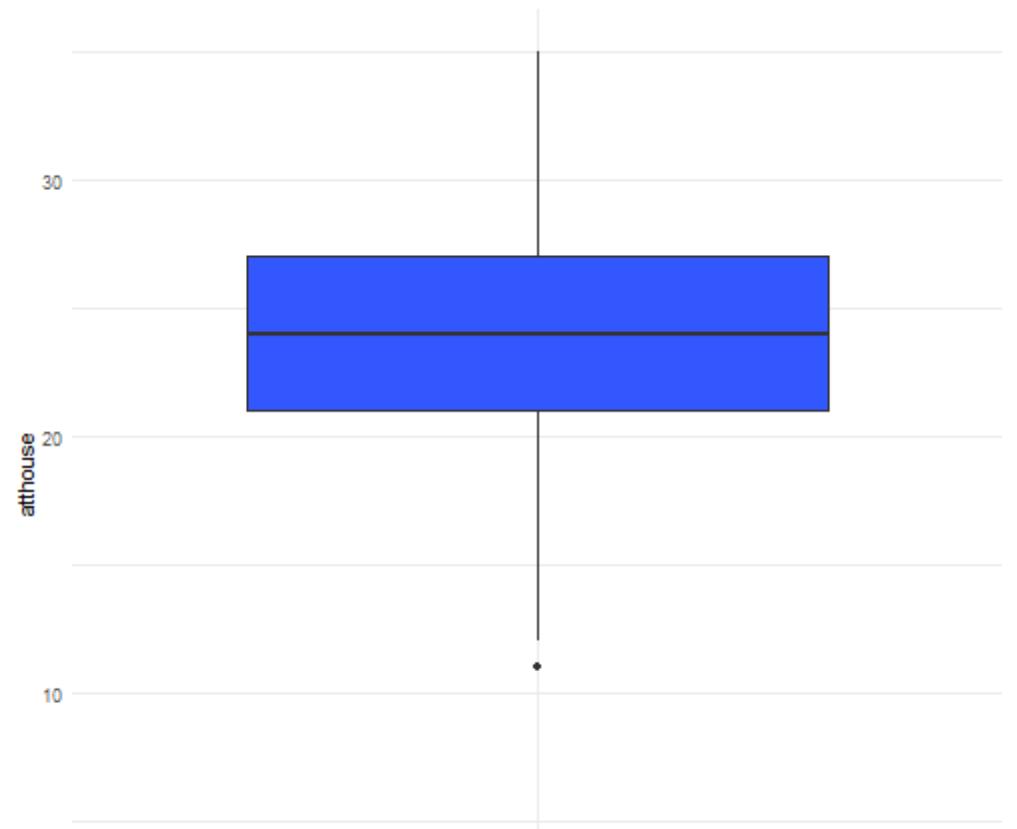


# Uç değerlerin Belirlenmesi

## 🔗 Color codes

```
atthouse_plt <-  
ggplot(screen) +  
aes(x = "", y = atthouse) +  
geom_boxplot(fill = "#3357FF") +  
theme_minimal()
```

atthouse\_plt



## Uç değerlerin Belirlenmesi

- Veri setinde potansiyel **tek değişkenli uç değerler tespit edildiğinde, önce uç değerin nedeni araştırılmalıdır.** Eğer veri girişinde hata varsa veya kayıp veri kodlanırken hata yapıldıysa düzeltilmelidir.
- Bunun dışındaki nedenlerde **değişkenin dönüştürülmesinin uygun olup olmayacağına karar verilmelidir.**
- **Dönüşümler hem dağılımların normallliğini geliştirir** hem de tek değişkenli uç değerleri dağılımin merkezine çekerler ve etkisini azaltırlar.
- **Dönüşümme karar verilirse çok değişkenli uç değerler incelenmeden dönüşüm yapılmalıdır.** Çünkü çok değişkenli uç değerlerin belirlenmesinde kullanılan istatistikler normal dağılımı gerektirir.

## Uç değerlerin Belirlenmesi

- Veri setinde iki değişken – **timedrs ve atthouse** üç değerlere sahiptir.
- **timedrs** değişkeni için üç değer olarak belirlenen değerlerin beklenen değerlerin üstünde olduğu ancak veri girişinde hata bulunmadığı rapor edilmiş, bu değerlere sahip bireylerin veri setinde kalmasına karar verilmiştir.
- **athouse** değişkeni için üç değerler olarak belirlenen değerler diğer değerlerden kopuktur. Bu değerlerin evren için beklenen değerler mi olduğuna veya veri girişinde hata yapılmış yapılmadığına karar verilmelidir.
- Her iki durumda da veri setinde 260. ve 296. satırda yer alan 2 birey (346 ve 407 subno.lu bireyler) veri setinden çıkarılabilir. 2 bireyin veri setinden çıkarılması sonucu örneklem büyütüğü 462'ye eşit olacaktır.

# Uç değerlerin Belirlenmesi

```
screen[c(260,298),]
```

```
## # A tibble: 2 × 7
##   subno timedrs attdrug atthouse income mstatus race
##   <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl> <dbl>
## 1     346        2        8        2        1        1     1
## 2     407        2        8        2        4        1     1
```

```
screen <- screen[-c(260,298),]
```

# Mahalanobis Uzaklığı

- Çok değişkenli üç değerleri belirlemenin bir yolu **Mahalanobis uzaklığını** hesaplamaktır. Mahalanobis uzaklığı z puanının çok boyutlu versiyonudur. Bir gözlemin, dağılımın kovaryansı (çok boyutlu varyansı) verildiğinde, **dağılımın ağırlık merkezinden (çok boyutlu ortalamasından) uzaklığını ölçer.**
- **Mahalonobis uzaklığı** ki-kare dağılımı gösterir (serbestlik derecesi hesaplamada kullanılan değişken sayısına eşittir) ve ki-kare dağılımı kullanılarak değerlendirilebilir. Eğer hesaplanan Mahalonobis uzaklığının **gözlenme olasılığı 0.001 veya daha küçükse gözlem uçdeğerdir.**
- Bu yöntem eşit aralık veya eşit oran düzeyinde ölçülen değişkenler için veya sürekli değişken olarak ele alınan sıralama ölçüğünde ölçülen değişkenler için geçerli olup sınıflama düzeyinde ölçülen değişkenler için geçerli değildir.

# Mahalanobis Uzaklığı

- Mahalanobis Uzaklığı hesaplama

```
library(psych)
veri <- screen[,1:5]
md <- mahalanobis(veri, center = colMeans(veri), cov = cov(veri))
md
```

```
## [1] 3.7855174 4.5414927 3.5010769 7.2813646 5.4572396 2.8965495
## [7] 5.8078984 3.8794776 4.7511661 7.4154047 10.6020996 5.2491208
## [13] 6.0737317 3.2718847 12.3164634 4.4407488 4.8361596 6.3628056
## [19] 4.1265239 10.7975454 5.7484910 10.5727271 3.4984523 4.8334420
## [25] 4.7006156 9.5885717 3.5312950 5.5896224 5.1433235 7.2593681
## [31] 5.2943018 4.3283138 4.4507527 2.5727236 8.7821612 6.1025712
## [37] 7.2382743 4.0468912 7.7316581 28.8987230 5.4170887 4.1667572
## [43] 5.3116126 5.1569082 7.3460683 2.3625482 3.6737604 4.0534574
## [49] 3.3853834 3.4657620 4.5780857 3.0608354 4.6916196 4.4893953
## [55] 4.0245315 2.0344652 3.1050893 9.8614394 6.7505963 3.0433786
## [61] 6.5567014 3.3054220 2.4455501 7.0248226 5.6672530 2.9891672
## [67] 11.5828872 2.5441926 1.7667950 4.5960680 14.5333387 2.6837064
## [73] 6.4811658 3.0011028 2.3057045 9.7329514 1.7895523 5.3934136
## [79] 13.3716971 2.3110731 5.8720105 10.7725170 8.5039953 2.4574823
## [85] 8.8868422 4.3797949 3.6144474 4.7011680 3.7145569 4.4318054
## [91] 3.7427603 2.7907114 3.4843374 1.9955713 5.0384718 7.8181717
## [97] 2.7505160 4.1944071 9.6736148 3.3155116 3.8999808 6.5893242
```

# Mahalanobis Uzaklığı

Kritik değer belirleme

```
library(psych)
alpha <- .001
cutoff <- (qchisq(p = 1 - alpha, df = ncol(veri)))
cutoff
```

```
## [1] 20.51501
```

# Mahalanobis Uzaklığı

Kritik değer belirleme

```
ucdegerler <- which(md > cutoff)  
veri[ucdegerler, ]
```

```
## # A tibble: 9 × 5  
##   subno timedrs attdrug atthouse income  
##   <dbl>    <dbl>    <dbl>    <dbl>    <dbl>  
## 1     48      60       7      24       1  
## 2    235      60      10      29       4  
## 3    276      57       9      24       2  
## 4    291      52       8      19       1  
## 5    330      58       7      29       4  
## 6    370      57       8      23       4  
## 7    398      75       9      33       9  
## 8    502      56       8      19       3  
## 9    548      81       8      24       9
```

```
data_temiz <- veri[-ucdegerler, ]
```

# Mahalanobis Uzaklığı

- Mahalonobis uzaklığı değerleri kare ile değerlendirilir (serbestlik derecesi bağımsız değişken sayısına eşittir). Buna göre 20.51501 kritik değerinden büyük olan değerler 0.001 alfa düzeyinde istatistiksel olarak anlamlı olarak değerlendirilir.
- 548, 398, 48, 235, 330, 502, 276, 291 ve 370 subno.lu bireyler için Mahalonobis uzaklık değerleri kritik değerden büyüktür. Bu gözlemler çok değişkenli üç değerler olarak değerlendirilir.

```
veri[ucdegerler, ]
```

```
## # A tibble: 9 × 5
##   subno timedrs attdrug atthouse income
##   <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1     48      60       7      24       1
## 2    235      60      10      29       4
## 3    276      57       9      24       2
## 4    291      52       8      19       1
## 5    330      58       7      29       4
## 6    370      57       8      23       4
## 7    398      75       9      33       9
## 8    502      56       8      19       3
## 9    548      81       8      24       9
```

## Çok Değişkenli Normallik Sayıltısı

- Çok değişkenli **normallik her bir değişkenin ve değişkenlerin bütün doğrusal kombinasyonlarının normal dağıldığı sayıltısıdır.**
- Sayıltının karşılanması durumunda **analizin artıkları (hataları) da normal dağılır.**
- Bir değişken çok değişkenli normal dağılıma sahipse, tek değişkenli normal dağılıma da sahiptir ancak bunun tersi sağlanmayabilir; iki veya daha fazla tek dağılımlı normallik gösteren değişkenler çok değişkenli normallik göstermeyebilir.

## Çok Değişkenli Normallik Sayıltısı

- Çok değişkenli normallik sayıltısını test etmek için doğrudan bir test bulunmadığından, genellikle her bir değişken ayrı ayrı test edilir ve eğer her bir değişken normal dağılım gösteriyorsa, çok değişkenli normal oldukları varsayıılır.
  - Not: **Her bir değişkenin normal olarak dağılımı çok değişkenli normallik için gereklidir ancak yeterli değildir.**
- Normallliğin değerlendirilmesi için hem istatistiksel hem de grafiksel yöntemler vardır.
  - İstatistiksel yöntemler normallik için hipotez testlerini içerir.
  - Grafiksel yöntemler histogram ve normallik grafiklerinin incelenmelerini içerir.

# Normallik Sayıltısı

Normallliğin iki bileşeni vardır: **Çarpıklık ve basıklık**

- Bir değişkene ait dağılım normal olduğunda, değişkenin çarpıklık ve basıklık değerleri sıfıra eşittir.
- Kural olarak eğer değişkenin **çarpıklık ve basıklık değerleri -1.0 ile +1.0 arasındaysa, değişkenin normale oldukça yakın olduğu söylenebilir.**
- Hem çarpıklık hem de basıklık için istatistiksel anlamlılık testleri vardır. Bu testlerde z dağılımı kullanılarak elde edilen çarpıklık veya basıklık değeri sıfır ile karşılaştırılır:

# Normallik Sayıltısı

```
library(sur)
attach(screen)

skew(timedrs)
```

```
## [1] 3.226868
```

```
se.skew(timedrs)
```

```
## [1] 0.1135929
```

```
skew.ratio(timedrs)
```

```
## [1] 28.4073
```

# Normallik Sayıltısı

```
library(moments)
library(labelled)
# jarque.test(remove_labels(timedrs))
```

```
# jarque.test(remove_labels(attdrug))
# jarque.test(remove_labels(atthouse))
```

---

📎 summarytools için

## Normallik Sayıltısı

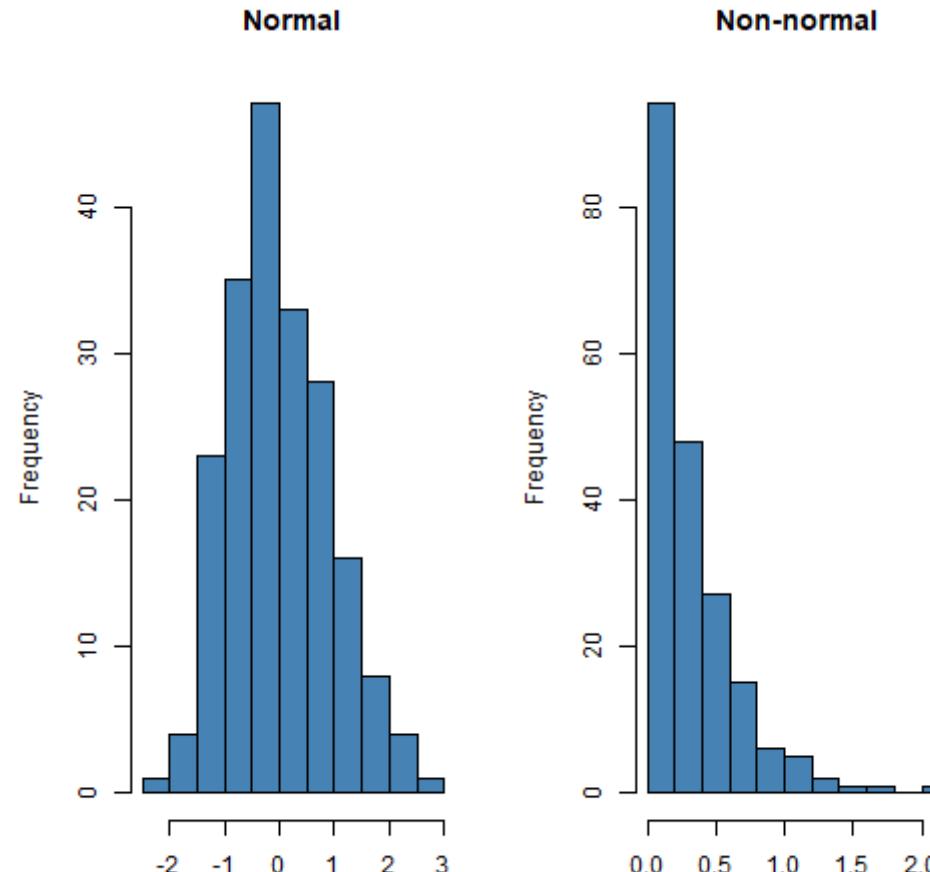
Histogramlar normallliğin değerlendirilmesi için kullanılan önemli grafiklerdir.

- Histogramların yanı sıra beklenen normal olasılık grafikleri de kullanılabilir.
- Bu grafiklerde gözlenen değerlerin yığılmalı dağılımı, normal bir dağılımın yığılmalı dağılımıyla karşılaştırılır. Grafikte gözlenen değerler X, beklenen değerler ise Y ekseninde yer alır.
- Eğer dağılım normalse, her bir gözleme ait nokta sol alttan sağ üste uzanan köşegenin üzerine düşer. Normallikten uzaklaşıldıkça noktalar da köşegenden uzaklaşır.

# Normallik Sayıltısı

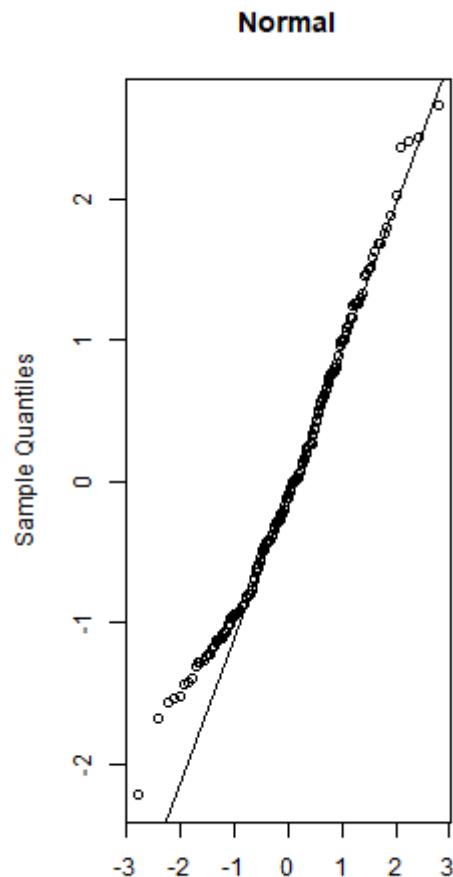
```
set.seed(0)
normal <- rnorm(200)
non_normal<-
rexp(200, rate=3)
```

```
par(mfrow=c(1,2))
hist(normal, col='steelblue', main='Normal')
hist(non_normal, col='steelblue', main='Non-normal')
```

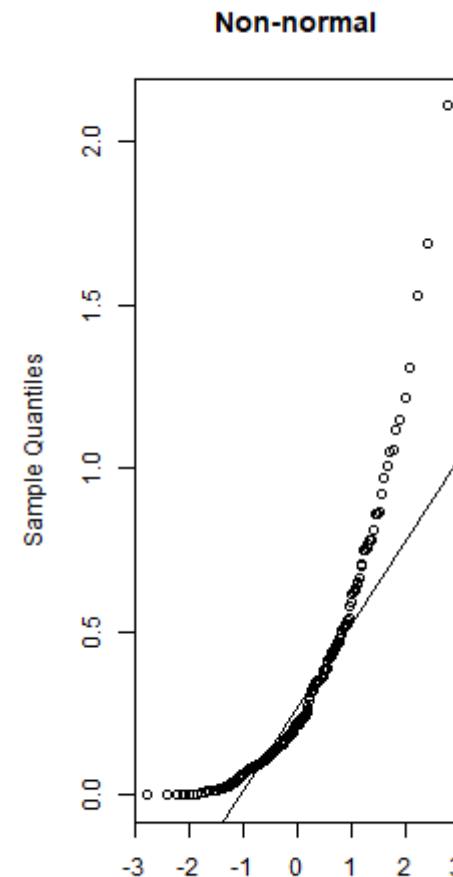


# Normallik Sayıltısı

```
par(mfrow=c(1,2))
qqnorm(normal, main='Normal')
qqline(normal)
```

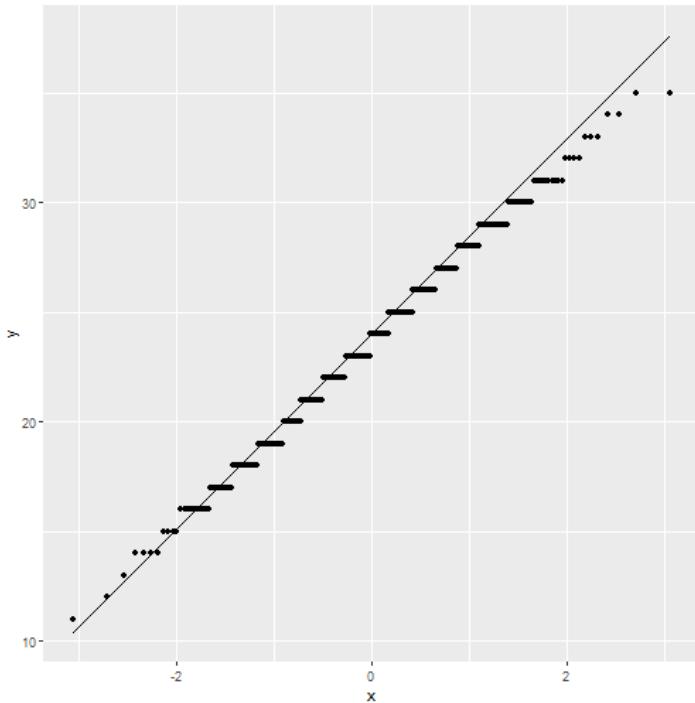


```
par(mfrow=c(1,2))
qqnorm(non_normal, main='Non-normal')
qqline(non_normal)
```

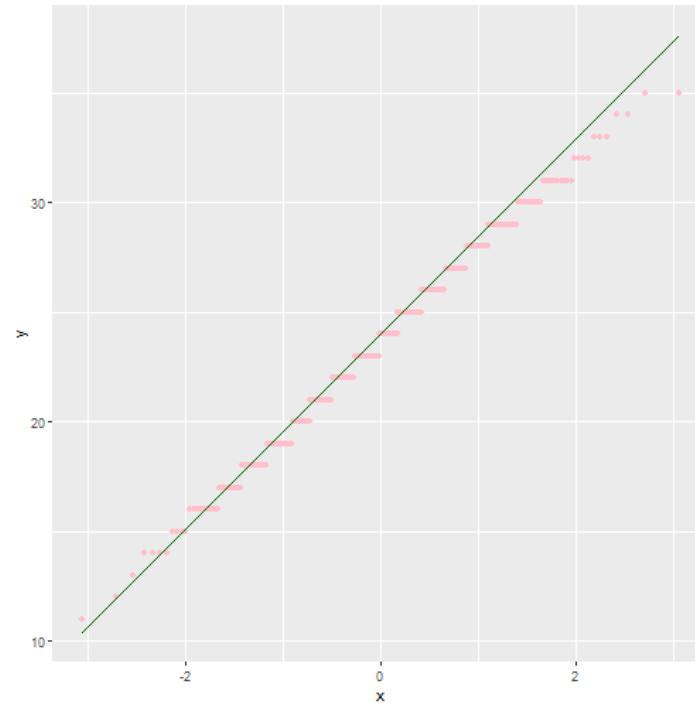


# Normallik Sayıltısı

```
ggplot(data = data_temiz, aes(sample = atthouse)) +  
  geom_qq() +  
  geom_qq_line()
```

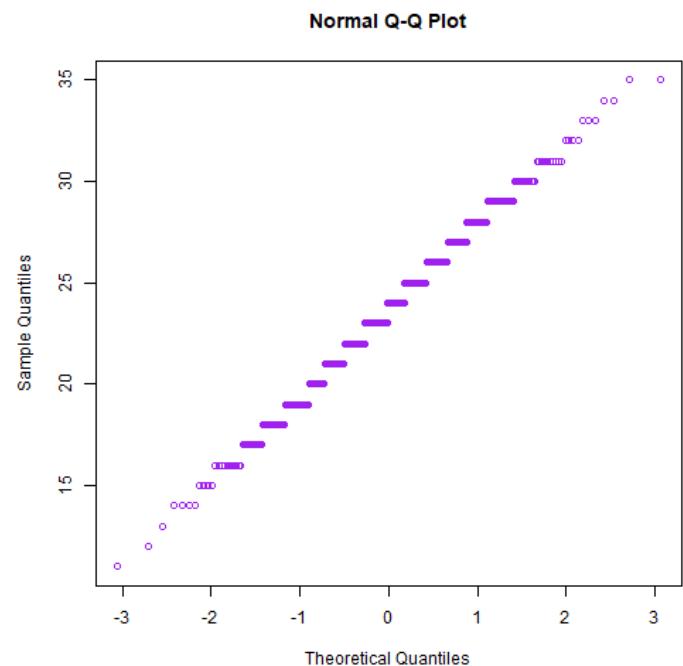


```
ggplot(data = data_temiz, aes(sample = atthouse)) +  
  geom_qq(color = "pink") +  
  geom_qq_line(color = "dark green")
```



# Normallik Sayıltısı

```
qqnorm(data_temiz$atthouse ,  
       col = "purple")
```

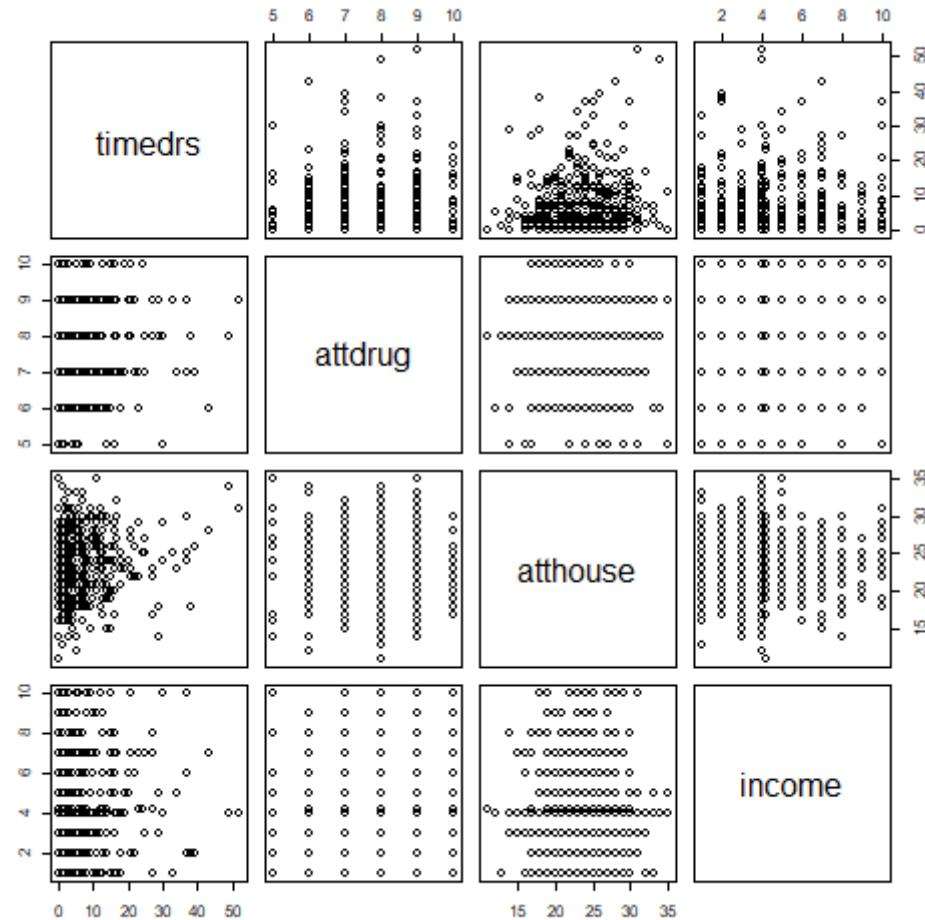


# Varyansların homejenliği

- Varyansların homojenliği (homoscedasticity), bağımlı değişken(ler)in bağımsız değişken(ler)in aralığı boyunca aynı düzeyde varyansa sahip olduğu sayiltisidir.
- Çoğu durumda, bağımsız değişkenin her bir değerinde bağımlı değişkenin çok farklı değerleri bulunur. Bu ilişkinin ele alınabilmesi için bağımlı değişkenin değerlerinin varyansı, bağımsız değişkenin her değerinde oldukça eşit olmalıdır.
- **Varyansların homojenliği normallik sayiltisi ile ilişkilidir.** Çok değişkenli normallik sayiltisi karşılandığında, değişkenler arasındaki ilişkiler homojendir.
- **Varyansların heterojenliği değişkenlerden birinin normal dağılım göstermemesinden veya bağımsız değişkendeki hatalı ölçümlerden kaynaklanabilir.**

# Varyansların homejenliği

```
pairs(data_temiz[,2:5])
```



# Veri Dönüşümü

- Normallik ve varyansların homojenliği sayıltıları ihlal edildiği zaman veri dönüşümü düşünülebilir. **Ancak veri dönüştürüldüğü zaman yorumlanmasıının da güçleşebileceği göz önünde bulundurulmalıdır.**
- Veri dönüştürmede değişkenlerin normallikten ne kadar uzaklaştıkları önemlidir.
- Eğer dağılım normalden **orta derecede farklılık gösteriyorsa, ilk olarak karekök dönüşümü denenir.**
- Eğer dağılım normalden **önemli derecede farklılık gösteriyorsa, log dönüşümü denenir.**
- Eğer dağılım normalden **ciddi derecede farklılık gösteriyorsa, ters dönüşümü denenir.**

## Veri Dönüştürme

- Veri dönüştürmede değişkenlerin normallikten ne yönde uzaklaştıkları önemlidir.
- Eğer sola çarpıklık varsa, değişkenin yansıtılması ve yansıtılma sonucu sağa çarpık şekilde dönüşen dağılım üzerinden dönüştürme işlemlerinin yapılması önerilir.
- Değişkeni yansıtmak için önce dağılımdaki en yüksek değer bulunur ve bu değere 1 eklerek sabit bir değer elde edilir. Sonra dağılımdaki her bir değer sabit değerden çıkarılarak yeni bir değişken elde edilir. Böylece dönüştürme işleminden önce sola çarpık dağılım sağa çarpık dağılıma dönüştürülmüş olur.
- Veri dönüştürme işlemlerinden sonra sayıtlar tekrar kontrol edilmelidir.

# Veri Dönüşümü

Dağılım	Dönüşüm
Orta düzeyde pozitif çarpık	Karekök
Yüksek düzeyde pozitif çarpık	Logaritma
Aşırı düzeyde pozitif çarpık	Ters Çevirme
Orta düzeyde negatif çarpık	Yansıtma ve karakök
Yüksek düzeyde negatif çarpık	Yansıtma ve logaritma
Aşırı düzeyde negatif çarpık	Yansıtma ve ters çevirme

# Veri Dönüşümü

```
ltimedrs <- log(timedrs+1)
describe(timedrs)
```

```
##      vars     n   mean     sd median trimmed   mad min max range skew kurtosis
## X1      1 462 7.94 10.97      4    5.64 4.45    0   81    81 3.22    12.79
##           se
## X1 0.51
```

```
describe(ltimedrs)
```

```
##      vars     n   mean     sd median trimmed   mad min max range skew kurtosis
## X1      1 462 1.71 0.96    1.61     1.7 0.76    0 4.41   4.41 0.22    -0.21
##           se
## X1 0.04
```

## Çoklu Bağlantı ve Tekillik

- (Çoklu) bağlantı ve tekillik bağımsız değişkenler arasındaki **korelasyon çok yüksek** olduğunda ortaya çıkan problemlerdir.
  - (Çoklu) bağlantıda değişkenler arasındaki korelasyon çok yüksektir.
  - Tekillikte değişkenler fazlalıktır; değişkenlerden biri analizdeki iki veya daha fazla değişkenin bileşimidir.
- Değişkenler (çoklu) bağlantılıysa veya tekilse, gereksiz bilgi içerirler ve analizde bu değişkenlerin hepsine ihtiyaç yoktur. Bu değişkenlerin hepsinin modele yer olması modeldeki hataları artırır ve analizi zayıflatır.

## Çoklu Bağlantı ve Tekillik

- Bağlantı problemini belirlemek için bağımsız değişkenler arasındaki iki değişkenli korelasyon katsayılarını içeren korelasyon matrisi incelenebilir.
- Örneğin, **iki değişken arasındaki korelasyon katsayısının 0.90 veya 0.90'dan daha yüksek olması bağlantı problemine işaretettir.**
  - Not: Yüksek korelasyon değerlerinin bulunmaması, bağlantı probleminin olmadığı anlamına gelmez. Bağlantı ilgili bağımsız değişken dışındaki diğer bağımsız değişkenlerden iki veya daha fazlasının bir aradaki etkisinden kaynaklanabilir ki bu durumda çoklu bağlantı söz konusudur.
  - Çoklu bağlantının değerlendirilmesi için her bir bağımsız değişkenin diğer bağımsız değişkenler tarafından ne ölçüde açıklandığının tespit edilmesi gereklidir.

## Çoklu Bağlantı ve Tekillik

- Çoklu bağlantının belirlenmesinde her bir değişken için SMC (squared multiple correlation,  $R^2$ ) değeri incelenebilir.  $R^2$  değeri regresyon modelinde belli bir **bağımsız değişkenin gözlenen varyansının diğer bütün bağımsız değişkenler tarafından açıklanan miktarıdır.**
- $R^2$  değeri bağımsız değişkenlerden birinin (Örneğin, X1) bağımlı değişken, diğer bağımsız değişkenlerinse bağımsız değişken (Örneğin, X2, X3 gibi) olarak ele alındığı bir regresyon modeli kurularak hesaplanır.
- $R^2$  değeri yüksekse, **değişken diğer değişkenlerle oldukça ilişkilidir ve yüksek değerler çoklu bağlantıya işaretettir.**
- $R^2$  değeri 1'e eşitse, değişken diğer değişkenlerle mükemmel derecede ilişkilidir ve bu değer tekilliğe işaretettir.

## Çoklu Bağlantı ve Tekillik

- Çoklu bağlantının belirlenmesinde her bir değişken için tolerans (tolerance) ( $1 - R^2$ ) değeri incelenebilir. **Bu değer belli bir bağımsız değişkenin gözlenen varyansının modeldeki diğer bağımsız değişkenler tarafından açıklanmayan miktarıdır.**
- Örneğin, X1 değişkeninin gözlenen varyansının yaklaşık %25'i modeldeki diğer bağımsız değişkenler tarafından açıklanırsa ( $R^2 = 0.25$ ), X1 değişkeninin tolerans değeri yaklaşık 0.75'tir ( $1 - R^2 = 1 - 0.25 = 0.75$ ).
- Tolerans değerinin yüksek olması gereklidir. Daha düşük tolerans değerleri, daha yüksek derecede çoklu bağlantı anlamına gelir. **Tolerans değeri için önerilen kesme değeri 0.10'dur.** Bu değer bir bağımsız değişken ve diğer bağımsız değişkenler arasında 0.95 değerinde bir çoklu korelasyona karşılık gelmektedir.

## Çoklu Bağlantı ve Tekillik

- Çoklu bağlantının belirlenmesinde her bir değişken için **VIF** değeri incelenebilir. **VIF** değeri tolerance değerinin tersi alınarak hesaplanır. ( $1/(1 - R^2)$ )
- Örneğin  $X_1$  değişkenin tolerans değeri yaklaşık 0.75 ise **VIF** değeri 1.33 olacaktır. **VIF** değerinin karekökü çoklu bağlantıdan kaynaklı standart hatanın artma derecesini yansıtır.
- **VIF** değerinin kesem değeri 10'dur. Dolayısıla standart hatalar hiç çoklu bağlantı bulunmayan duruma oranla üç kattan daha fazla artacaktır.

## Çoklu Bağlantı ve Tekillik

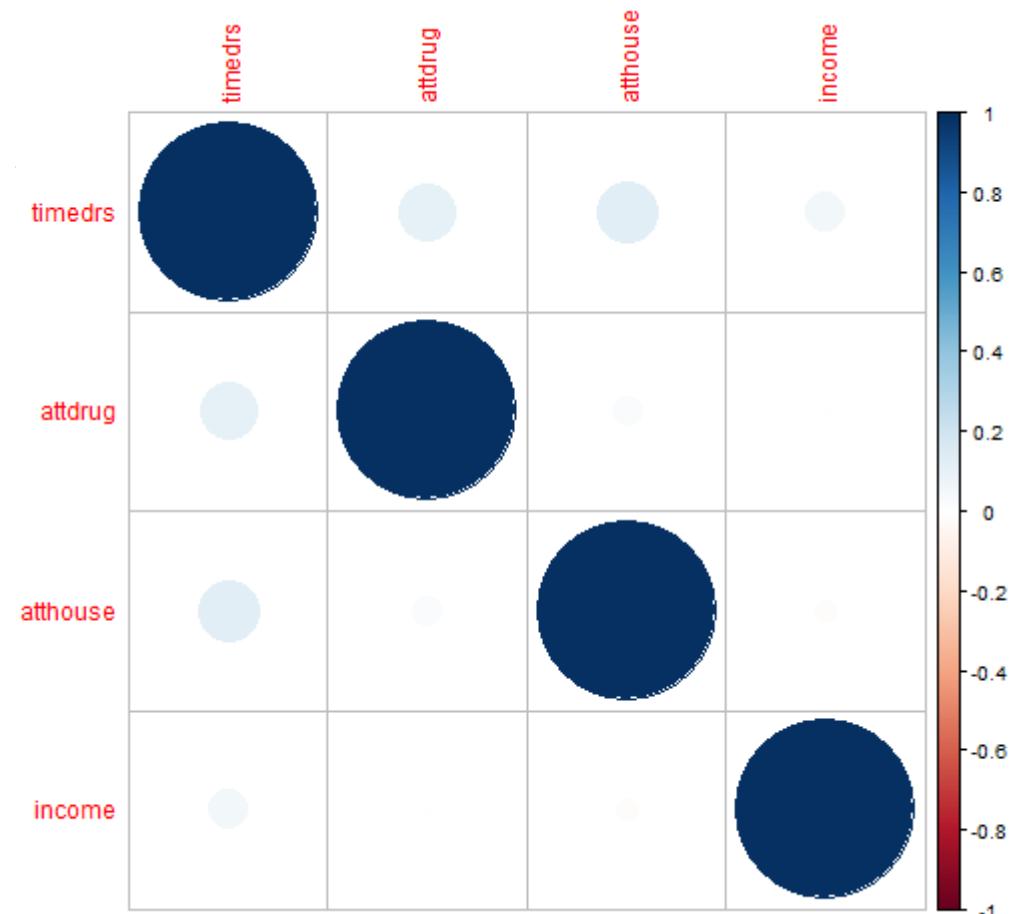
- Çoklu bağlantı problemi belirlenirse,
- Birinci seçenek çoklu bağlantıya neden olan değişkenlerden en az birisinin analizden çıkarılmasıdır.
- İkinci seçenek çoklu bağlantıya neden olan değişkenlere ait değerlerin toplanması veya ortalamasının alınmasıdır.
- Üçüncü seçenek temel bileşenlerin hesaplanması ve analizlerde temel bileşenlerin kullanılmasıdır.

# Çoklu Bağlantı ve Tekillik

```
library(corrplot)
cor(screen[,2:5])
```

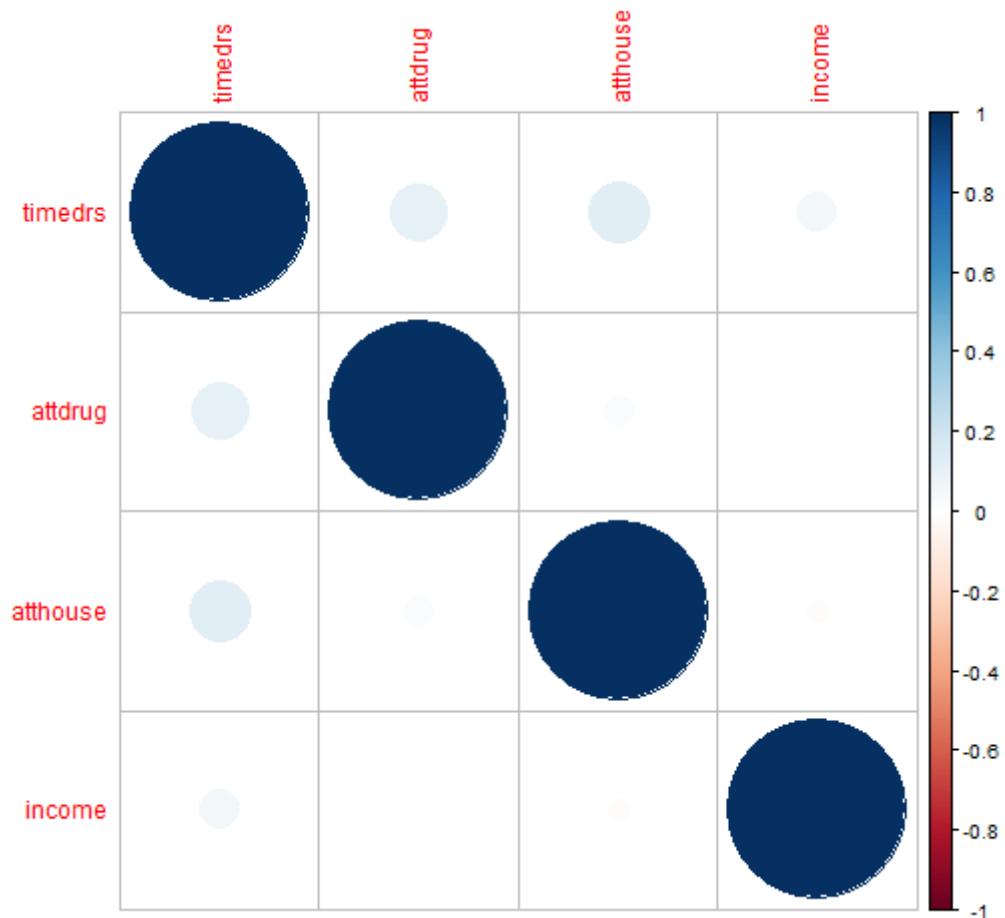
```
##          timedrs      attdrug     atthouse
## timedrs 1.00000000 0.103581423 0.12367560
## attdrug  0.10358142 1.000000000 0.02961376
## atthouse 0.12367560 0.029613756 1.00000000
## income   0.05116209 0.001887284 -0.01563911
```

```
corrplot(cor(screen[,2:5]))
```



# Çoklu Bağlantı ve Tekillik

```
corplot(cor(screen[,2:5]))
```



# Çoklu Bağlantı ve Tekillik

```
model <- lm(subno ~ timedrs + attdrug + atthouse + income + race+ mstatus ,  
data = screen)  
library(olsrr)  
ols_vif_tol(model)
```

```
##    Variables Tolerance      VIF  
## 1   timedrs 0.9689049 1.032093  
## 2   attdrug 0.9877891 1.012362  
## 3   atthouse 0.9750074 1.025633  
## 4   income 0.7715697 1.296059  
## 5   race 0.9872218 1.012944  
## 6   mstatus 0.7724256 1.294623
```

teşekkürler

