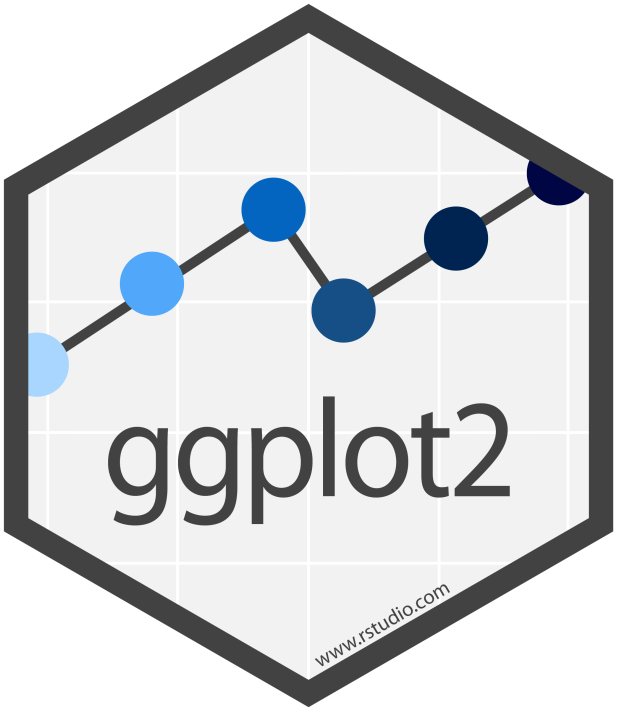


R ile Keşifçi Veri Analizi ve Veri Görselleştirme

Muhammed Fatih TÜZEN

11 01 2022

İÇİNDEKİLER

1 Keşifçi Veri Analizi

Keşifçi Veri Analizi (EDA), verileriniz hakkında bilgi sağlayaya yarayan bir süreçtir. EDA, verileri tanımanıza veya verilerdeki olası özellikler ve ilişkiler hakkında daha derin bir anlayış kazanmanıza yardımcı olabilir. EDA, yeni bir şey değildir, ancak EDA, birkaç nedenden dolayı yakın geçmişte önemli ölçüde büyümüştür:

- Veriler her zamankinden daha hızlı ve daha büyük miktarlarda üretiliyor, bu yüzden incelememiz gereken çok şey var.
- Bilgisayarlar ve yazılımlar (R gibi) EDA yapma fırsatlarını genişletmiştir.
- İstatistiksel model seçeneklerindeki artış, genellikle doğrudan geleneksel bir modele gitmek yerine verilerimize daha yakından bakmamızı gerektirmektedir.

EDA, verilerinizin nihai analizi açısından genellikle istatistiksel değildir, ancak EDA'nın geçiş süreci olarak düşünülmesi gerekir. EDA'dan öğrendikleriniz modellemenize rehberlik edecek ve istatistiksel araçlar hakkında verdiğiniz kararları doğrudan bilgilendirecektir.

1.1 Veri ile Tanışma

Veri analizinin başlangıç aşamasında, verinin yapısına, ne tür değişkenler içerdiğine, çeşitli özet istatistiklerine bakmak ve gerekli ise ne tür dönüşümler yapmak gerektiğini bilmek önemlidir. Bu süreçler daha derin analizlere daha kolay devam edebilmek için de önemlidir. Bunları gerçekleştirmek için hem özet tablolar hem de grafikler yardımıyla verileri tanımak gerekmektedir.

Tek ve iki değişkenli olarak sayısal ve kategorik veri analizi **mpg** verisi kullanılarak yapılacaktır. Bu veri setinde 38 farklı aracın yakıt verileri bulunmaktadır.

```
library(ggplot2)

colnames(mpg)
```

```
##  [1] "manufacturer" "model"        "displ"        "year"         "cyl"
##  [6] "trans"        "drv"          "cty"          "hwy"          "fl"
## [11] "class"
```

```
nrow(mpg)
```

```
## [1] 234
```

```
ncol(mpg)
```

```
## [1] 11
```

```
head(mpg)
```

```
## # A tibble: 6 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 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
```

```
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

```
summary(mpg)
```

```
##  manufacturer          model               displ            year
##  Length:234         Length:234         Min.   :1.600   Min.   :1999
##  Class :character   Class :character   1st Qu.:2.400   1st Qu.:1999
##  Mode  :character   Mode  :character   Median :3.300   Median :2004
##                                        Mean   :3.472   Mean   :2004
##                                        3rd Qu.:4.600   3rd Qu.:2008
##                                        Max.   :7.000   Max.   :2008
##       cyl           trans               drv                 cty
```

```
##  Min.   :4.000   Length:234         Length:234         Min.   : 9.00
##  1st Qu.:4.000   Class :character   Class :character   1st Qu.:14.00
##  Median :6.000   Mode  :character   Mode  :character   Median :17.00
##  Mean   :5.889                                         Mean   :16.86
##  3rd Qu.:8.000                                         3rd Qu.:19.00
##  Max.   :8.000                                         Max.   :35.00
##       hwy             fl                class
##  Min.   :12.00   Length:234         Length:234
##  1st Qu.:18.00   Class :character   Class :character
##  Median :24.00   Mode  :character   Mode  :character
##  Mean   :23.44
##  3rd Qu.:27.00
##  Max.   :44.00
```

```
df <- mpg
df$class <- factor(df$class)
levels(df$class)
```

```
## [1] "2seater"    "compact"    "midsize"    "minivan"    "pickup"
## [6] "subcompact" "suv"
```

```
library(dplyr)
glimpse(df)
```

```
## Rows: 234
## Columns: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "audi", "~
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 quattro", "~
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0, 2.0, 2.~
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1999, 200~
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 8, 8, ~
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)", "auto~
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4", "4", "4~
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 17, 17, 1~
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 25, 25, 2~
## $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p~
## $ class        <fct> compact, compact, compact, compact, compact, compact, com~
```

```
library(funModeling)
profiling_num(df)
```

```
##   variable        mean  std_dev variation_coef   p_01   p_05   p_25   p_50
```

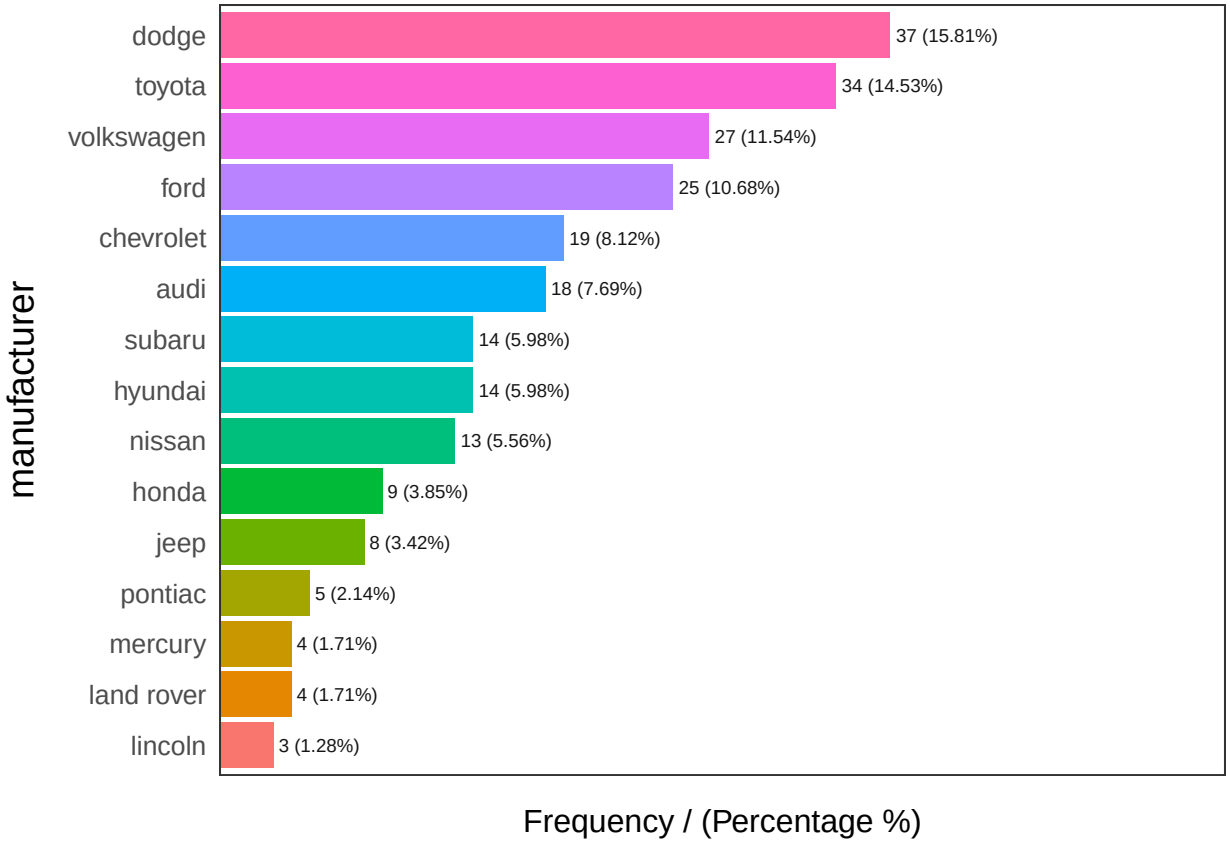
```
## 1    displ    3.471795 1.291959    0.372130002    1.6    1.8    2.4    3.3
## 2     year 2003.500000 4.509646    0.002250884 1999.0 1999.0 1999.0 2003.5
## 3      cyl    5.888889 1.611534    0.273656799    4.0    4.0    4.0    6.0
## 4      cty   16.858974 4.255946    0.252443926    9.0   11.0   14.0   17.0
## 5      hwy   23.440171 5.954643    0.254035837   12.0   15.0   18.0   24.0
##     p_75   p_95    p_99  skewness kurtosis iqr              range_98
## 1    4.6    5.7    6.20 0.4414630 2.107412 2.2            [1.6, 6.2]
## 2 2008.0 2008.0 2008.00 0.0000000 1.000000 9.0          [1999, 2008]
## 3    8.0    8.0    8.00 0.1130695 1.549122 4.0               [4, 8]
## 4   19.0   24.0   28.67 0.7914453 4.468651 5.0            [9, 28.67]
## 5   27.0   32.0   39.68 0.3668650 3.163929 9.0 [12, 39.6799999999999]
##       range_80
## 1     [2, 5.4]
## 2 [1999, 2008]
## 3       [4, 8]
## 4     [11, 21]
## 5   [16.3, 30]
```

```
freq(df)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

```
##    manufacturer frequency percentage cumulative_perc
## 1         dodge        37      15.81           15.81
## 2        toyota        34      14.53           30.34
## 3    volkswagen        27      11.54           41.88
## 4          ford        25      10.68           52.56
## 5     chevrolet        19       8.12           60.68
## 6          audi        18       7.69           68.37
## 7       hyundai        14       5.98           74.35
## 8        subaru        14       5.98           80.33
## 9        nissan        13       5.56           85.89
## 10        honda         9       3.85           89.74
## 11         jeep         8       3.42           93.16
## 12      pontiac         5       2.14           95.30
## 13   land rover         4       1.71           97.01
## 14      mercury         4       1.71           98.72
## 15      lincoln         3       1.28          100.00
```

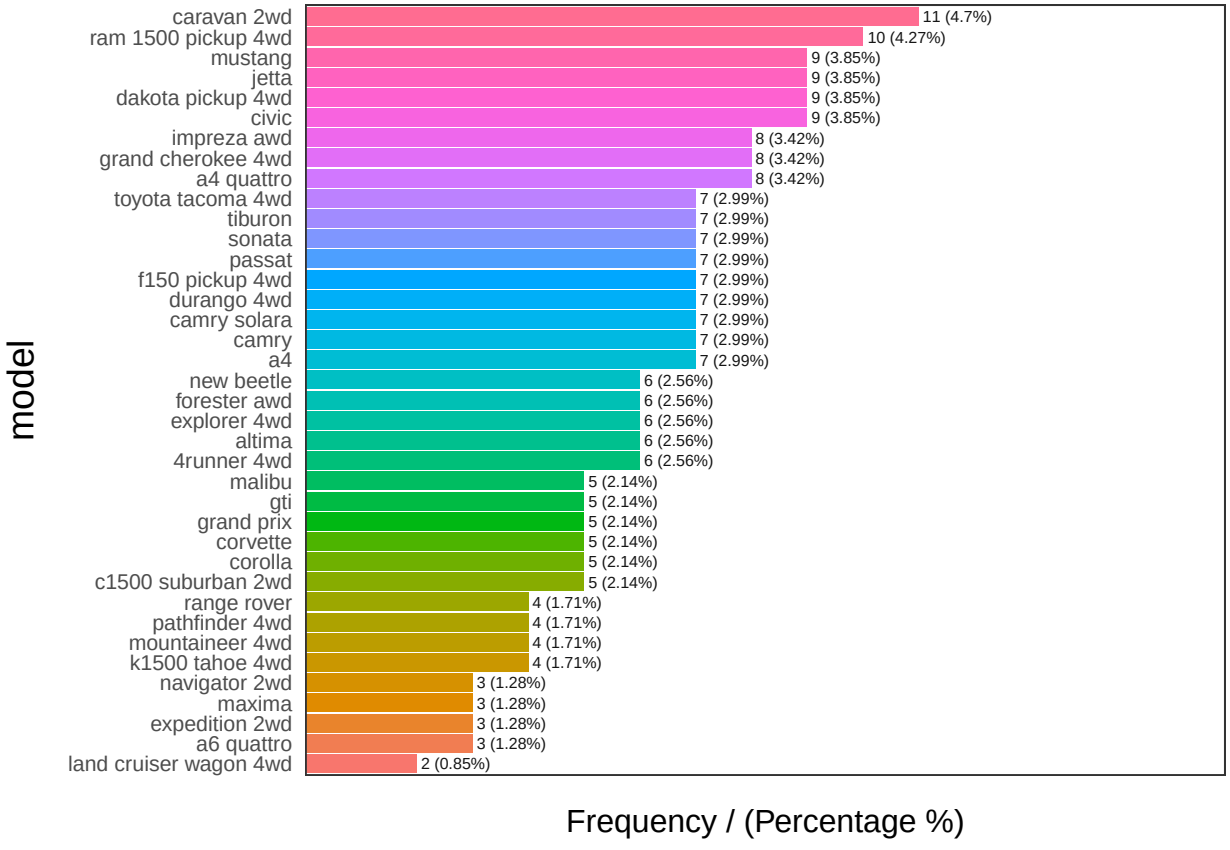
```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```




```
##                    model frequency percentage cumulative_perc
## 1             caravan 2wd        11       4.70            4.70
## 2     ram 1500 pickup 4wd        10       4.27            8.97
## 3                   civic         9       3.85           12.82
## 4       dakota pickup 4wd         9       3.85           16.67
## 5                   jetta         9       3.85           20.52
## 6                 mustang         9       3.85           24.37
## 7              a4 quattro         8       3.42           27.79
## 8       grand cherokee 4wd         8       3.42           31.21
## 9              impreza awd         8       3.42           34.63
## 10                    a4         7       2.99           37.62
## 11                 camry         7       2.99           40.61
## 12           camry solara         7       2.99           43.60
## 13           durango 4wd         7       2.99           46.59
## 14         f150 pickup 4wd         7       2.99           49.58
## 15                 passat         7       2.99           52.57
## 16                 sonata         7       2.99           55.56
## 17                tiburon         7       2.99           58.55
## 18       toyota tacoma 4wd         7       2.99           61.54
## 19             4runner 4wd         6       2.56           64.10
## 20                 altima         6       2.56           66.66
```

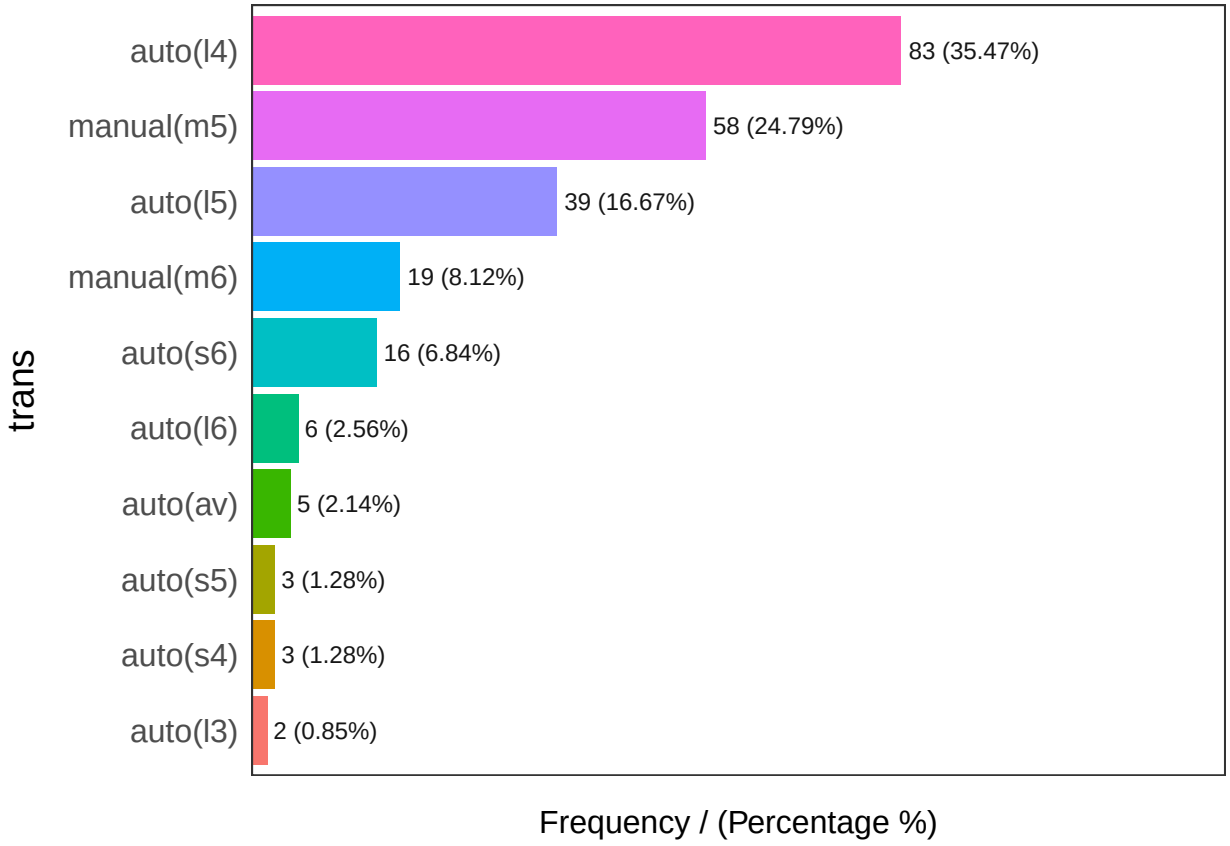
```
## 21           explorer 4wd         6       2.56          69.22
## 22           forester awd         6       2.56          71.78
## 23             new beetle         6       2.56          74.34
## 24     c1500 suburban 2wd         5       2.14          76.48
## 25                corolla         5       2.14          78.62
## 26              corvette         5       2.14          80.76
## 27            grand prix         5       2.14          82.90
## 28                   gti         5       2.14          85.04
## 29                malibu         5       2.14          87.18
## 30        k1500 tahoe 4wd         4       1.71          88.89
## 31        mountaineer 4wd         4       1.71          90.60
## 32         pathfinder 4wd         4       1.71          92.31
## 33            range rover         4       1.71          94.02
## 34             a6 quattro         3       1.28          95.30
## 35         expedition 2wd         3       1.28          96.58
## 36                 maxima         3       1.28          97.86
## 37          navigator 2wd         3       1.28          99.14
## 38 land cruiser wagon 4wd         2       0.85         100.00
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



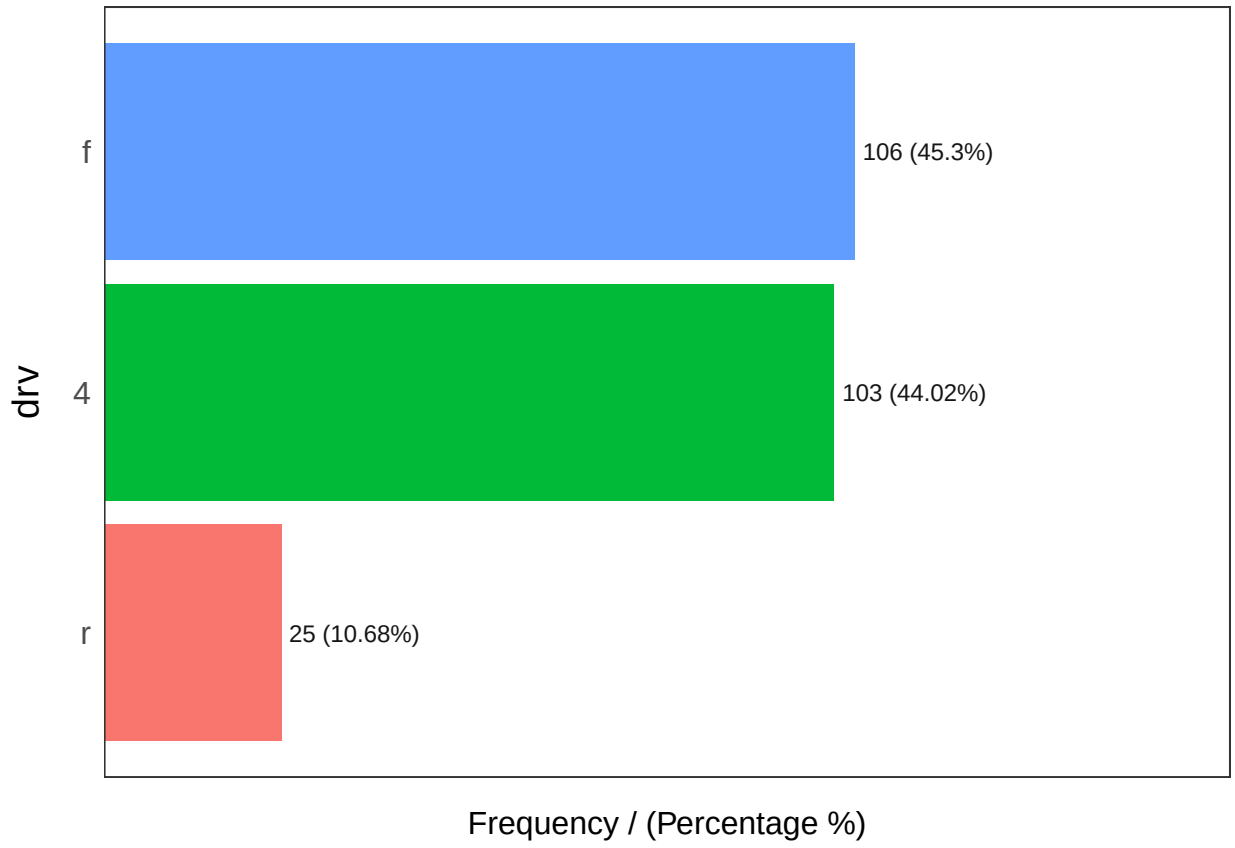
```
##         trans frequency percentage cumulative_perc
## 1    auto(l4)        83      35.47           35.47
## 2  manual(m5)        58      24.79           60.26
## 3    auto(l5)        39      16.67           76.93
## 4  manual(m6)        19       8.12           85.05
## 5    auto(s6)        16       6.84           91.89
## 6    auto(l6)         6       2.56           94.45
## 7    auto(av)         5       2.14           96.59
## 8    auto(s4)         3       1.28           97.87
## 9    auto(s5)         3       1.28           99.15
## 10   auto(l3)         2       0.85          100.00
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



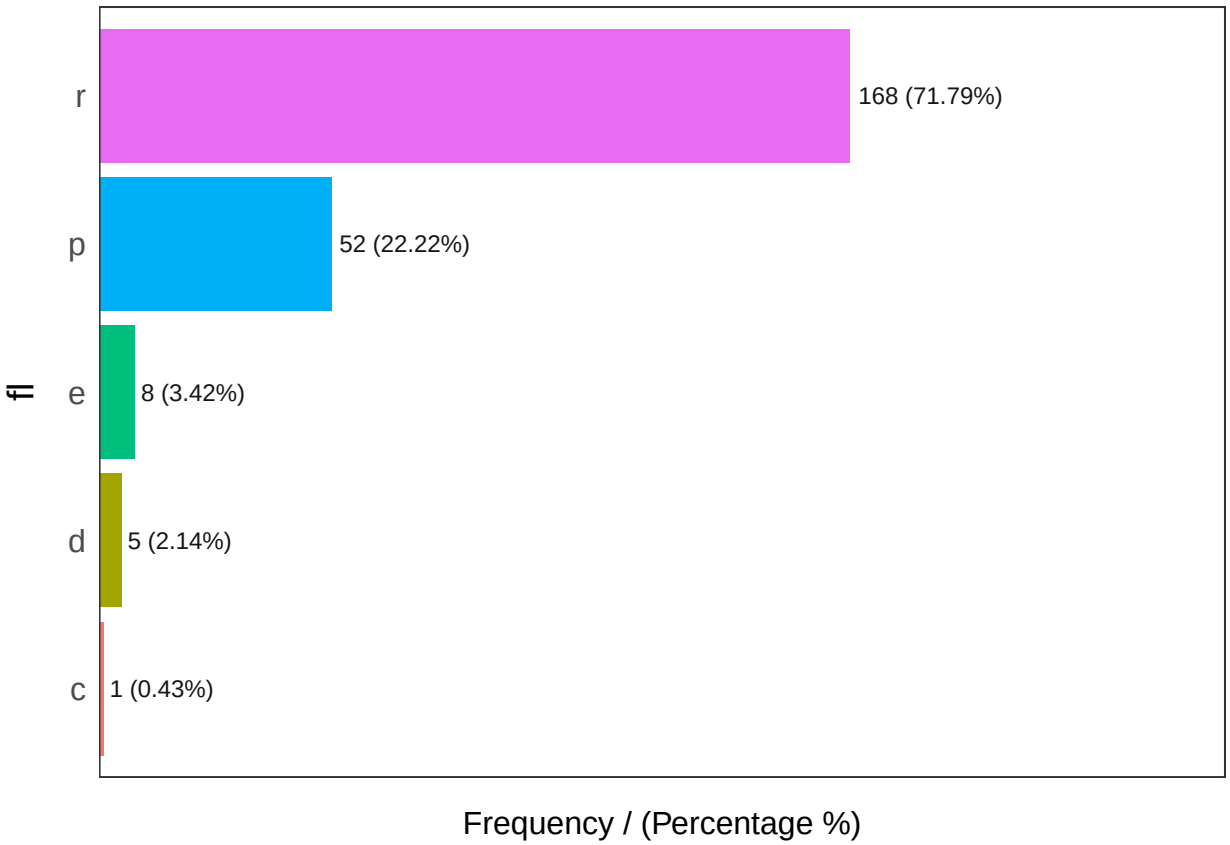

```
##   drv frequency percentage cumulative_perc
## 1   f       106      45.30           45.30
## 2   4       103      44.02           89.32
## 3   r        25      10.68          100.00
```

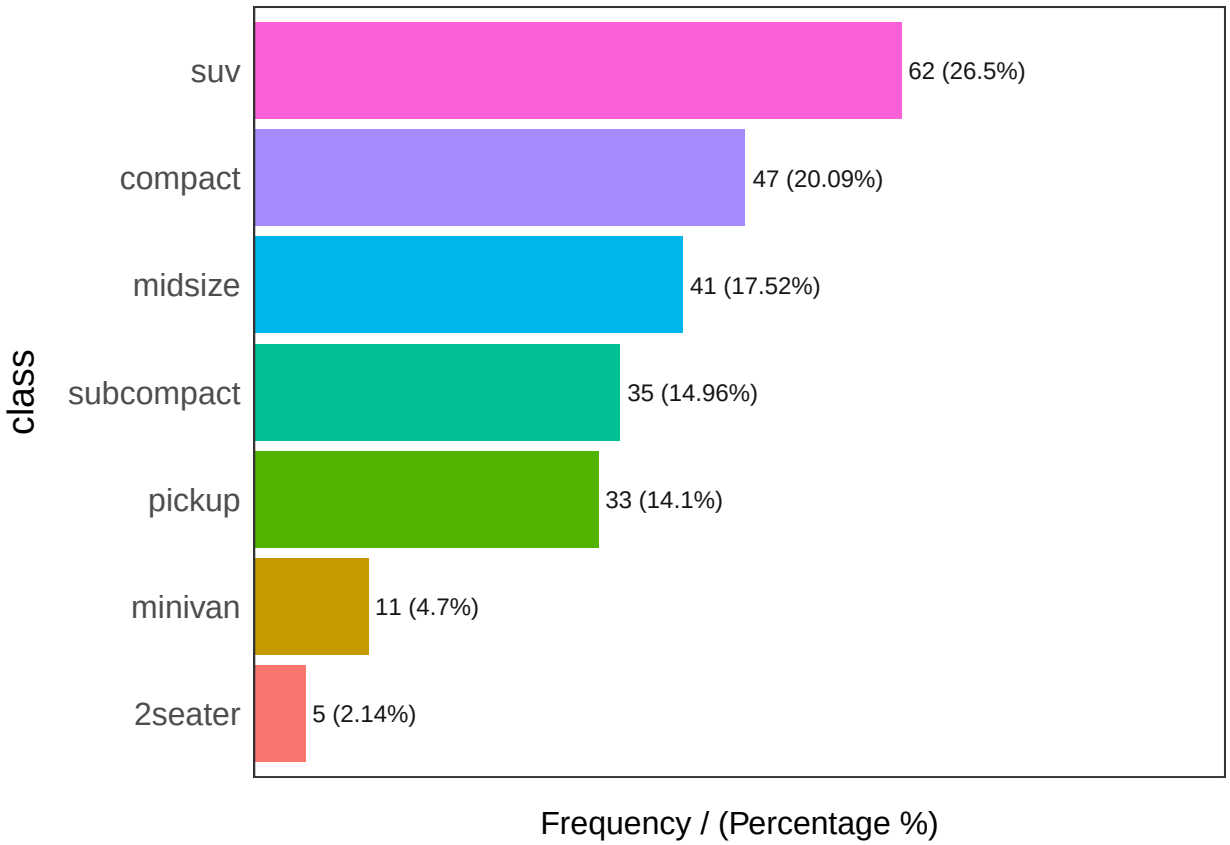
```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```




```
##   fl frequency percentage cumulative_perc
## 1  r       168      71.79           71.79
## 2  p        52      22.22           94.01
## 3  e         8       3.42           97.43
## 4  d         5       2.14           99.57
## 5  c         1       0.43          100.00
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```


fl
r
p
e
d
c
168 (71.79%)
52 (22.22%)
8 (3.42%)
5 (2.14%)
1 (0.43%)
Frequency / (Percentage %)

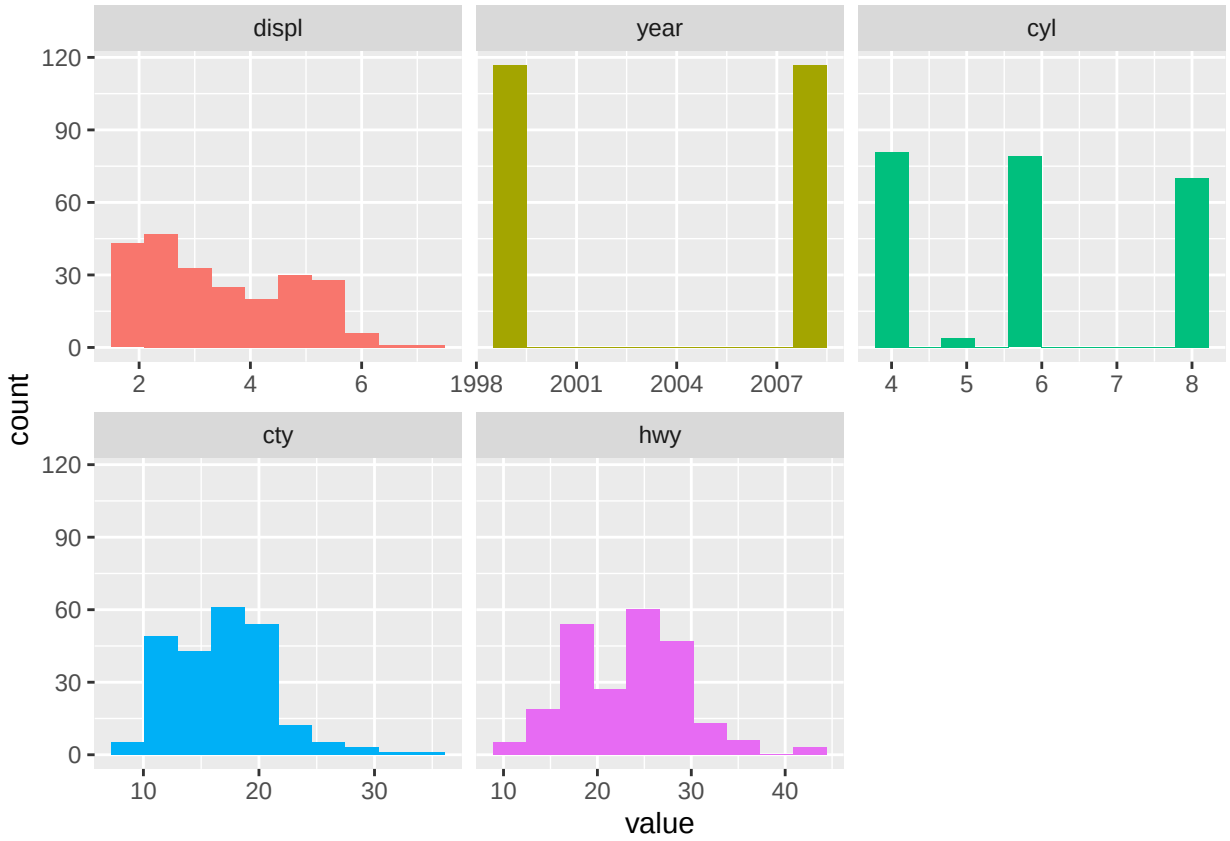



```
##        class frequency percentage cumulative_perc
## 1        suv        62      26.50           26.50
## 2    compact        47      20.09           46.59
## 3    midsize        41      17.52           64.11
## 4 subcompact        35      14.96           79.07
## 5     pickup        33      14.10           93.17
## 6    minivan        11       4.70           97.87
## 7    2seater         5       2.14          100.00
```

```
## [1] "Variables processed: manufacturer, model, trans, drv, fl, class"
```

```
plot_num(df, bins = 10)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



1.2 Sürekli Tek Değişken

```
# cty ve hwy değişkenlerini inceleyelim.
# cty şehiriçi, hwy şehirarasını ifade ediyor.

summary(df$cty)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9.00   14.00   17.00   16.86   19.00   35.00
```

```
var(df$cty)
```

```
## [1] 18.11307
```

```
mean(df$cty)
```

```
## [1] 16.85897
```

```
summary(df$hwy)
```

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

```
var(df$hwy)
```

```
## [1] 35.45778
```

```
mean(df$hwy)
```

```
## [1] 23.44017
```

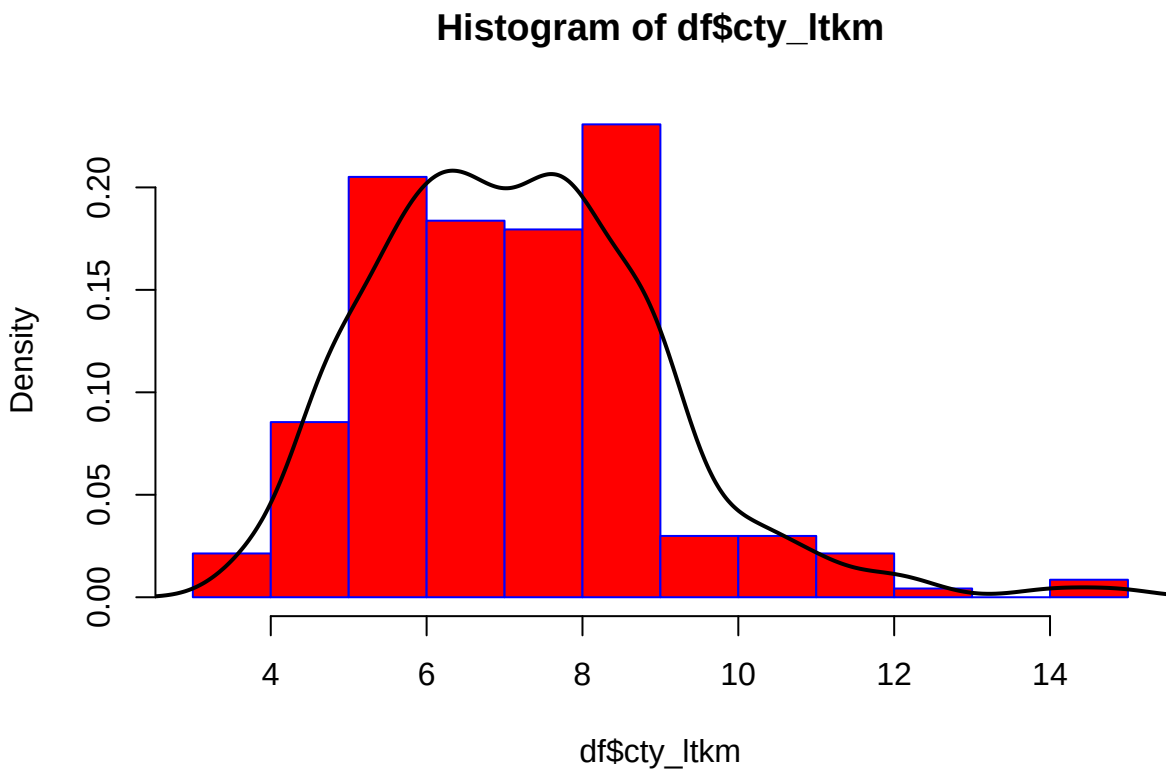
```
# 1 mile= 1.609 km
# 1 galon = 3.79 lt

# litre başına km hesaplama
galonmil_to_ltkm <- function(x){

  km <- x * 1.609/3.79
  return(km)
}

df$cty_ltkm <- galonmil_to_ltkm(df$cty)
df$hwy_ltkm <- galonmil_to_ltkm(df$hwy)
quantile(df$cty_ltkm)
```

```
##        0%       25%       50%       75%      100%
##  3.820844  5.943536  7.217150  8.066227 14.858839
```

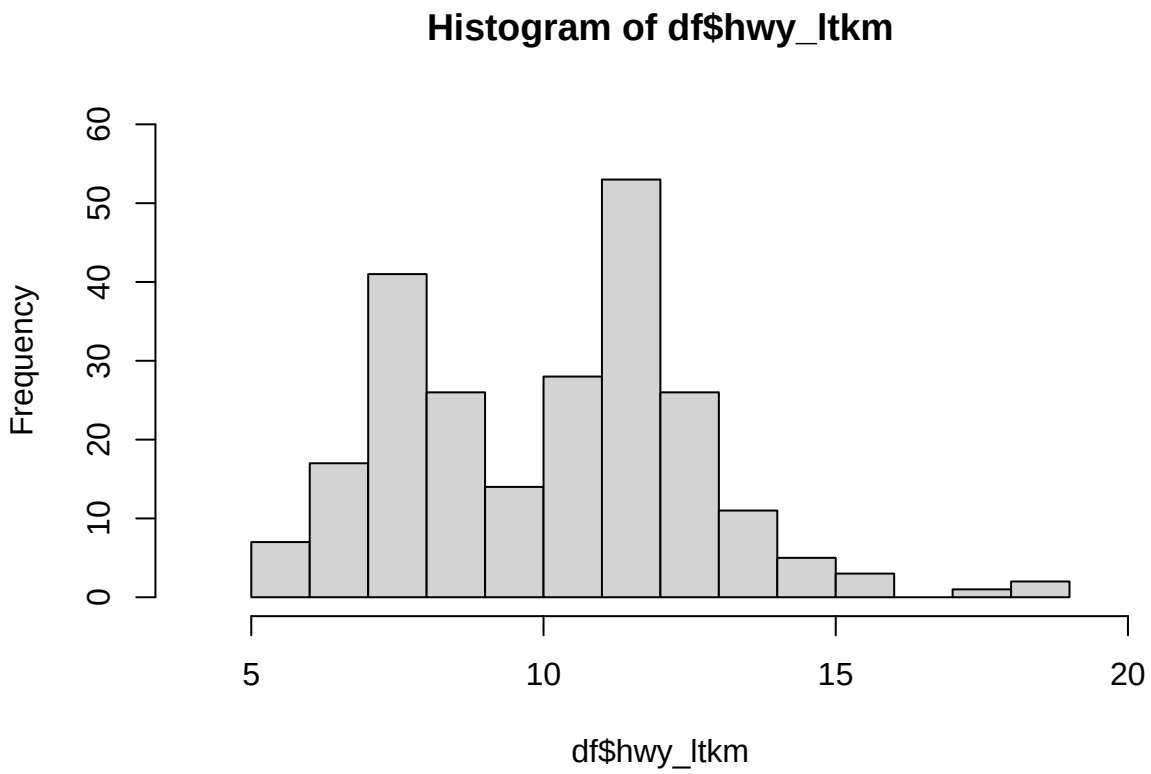
```
# şehiriçi araçların % 75'i 1 lt ile 8.06 km den az yol alıyor.
quantile(df$hwy_ltkm)
```

```
##        0%       25%       50%       75%      100%
##  5.094459  7.641689 10.188918 11.462533 18.679683
```

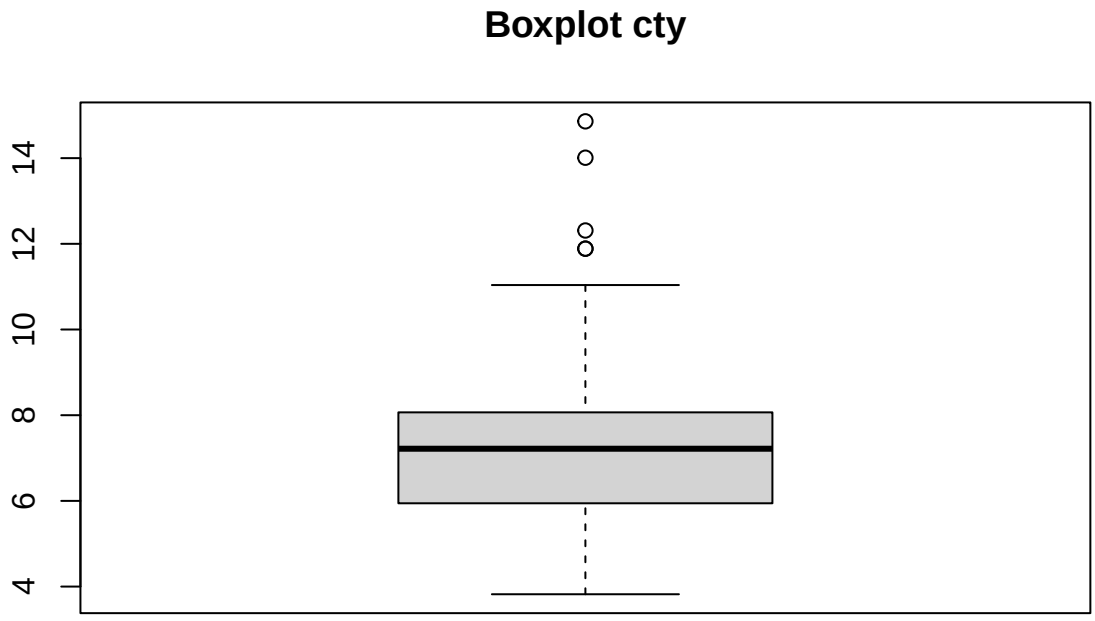
```
# şehirlerarası araçların % 75'i 1 lt ile 11.46 km den az yol alıyor.


# değişken dağılımı için histogram grafiği kullanılabilir.
hist(df$cty_ltkm,freq = FALSE,col = "red",border = "blue")
lines(density(df$cty_ltkm), col = "black", lwd = 2,)
```




```
hist(df$hwy_ltkm,xlim = c(4,20), ylim = c(0,60), breaks = 10)
```



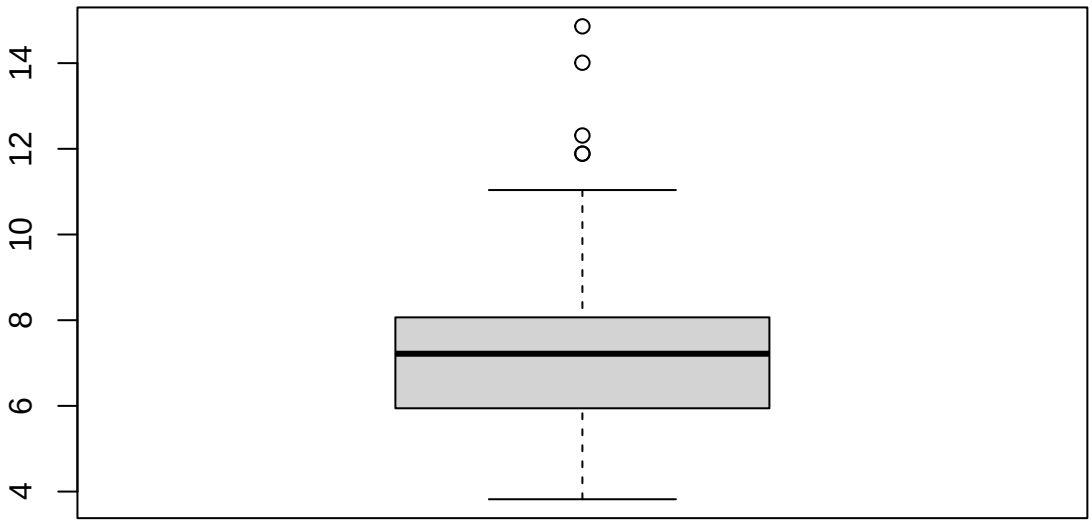

```
# Boxplot
boxplot(df$cty_ltkm, main = "Boxplot cty")
```




```
fivenum(df$cty_ltkm) # minimum, Q1, median, Q3, maximum
```

```
## [1]  3.820844  5.943536  7.217150  8.066227 14.858839
```

```
# outliers
boxplot(df$cty_ltkm)$out
```

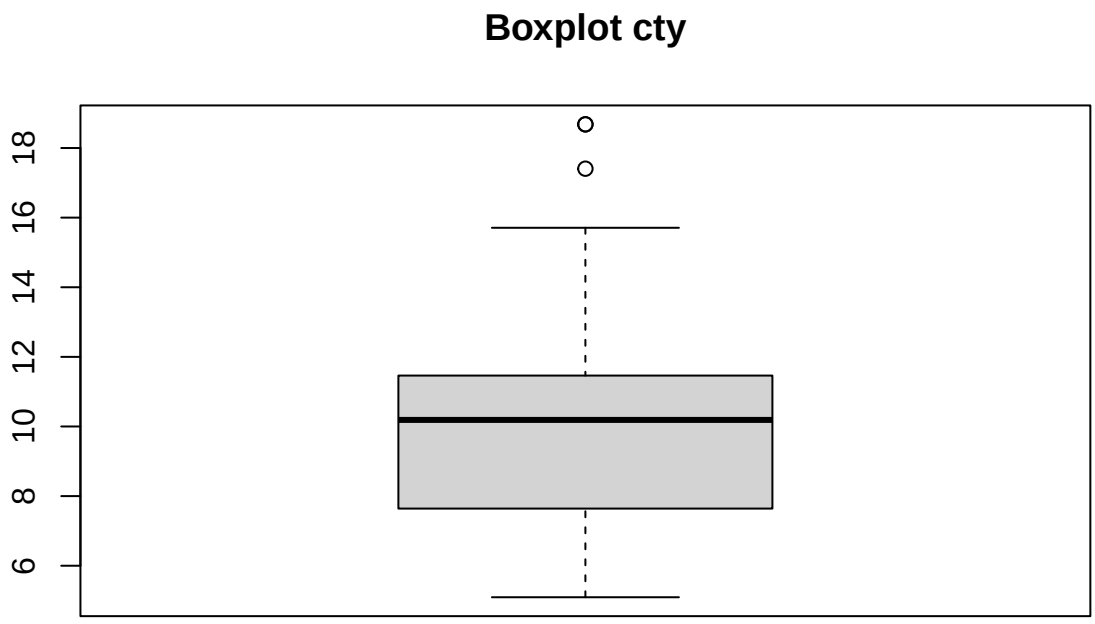



```
## [1] 11.88707 11.88707 14.00976 14.85884 12.31161
```

```
# outliers hangi sıralarda
which(df$cty_ltkm %in% boxplot(df$cty_ltkm)$out)
```

```
## [1] 100 197 213 222 223
```

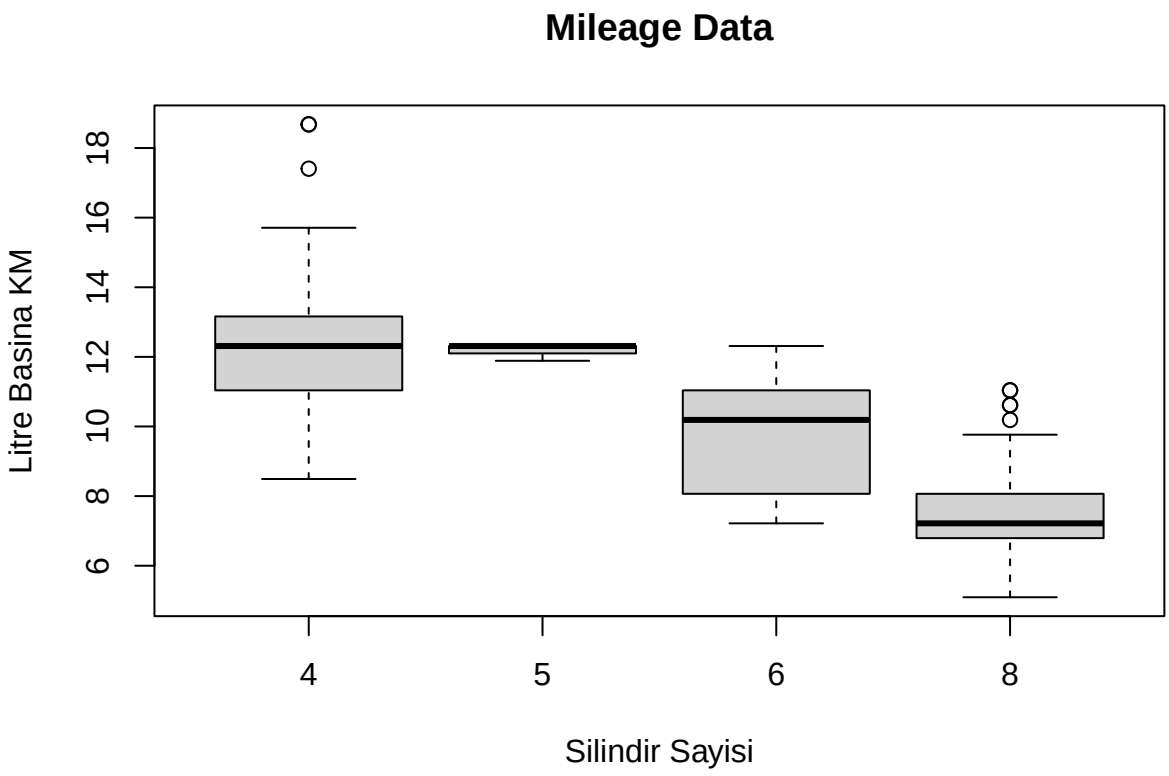
```
boxplot(df$hwy_ltkm, main = "Boxplot cty")
```




```
fivenum(df$hwy_ltkm) # minimum, Q1, median, Q3, maximum
```

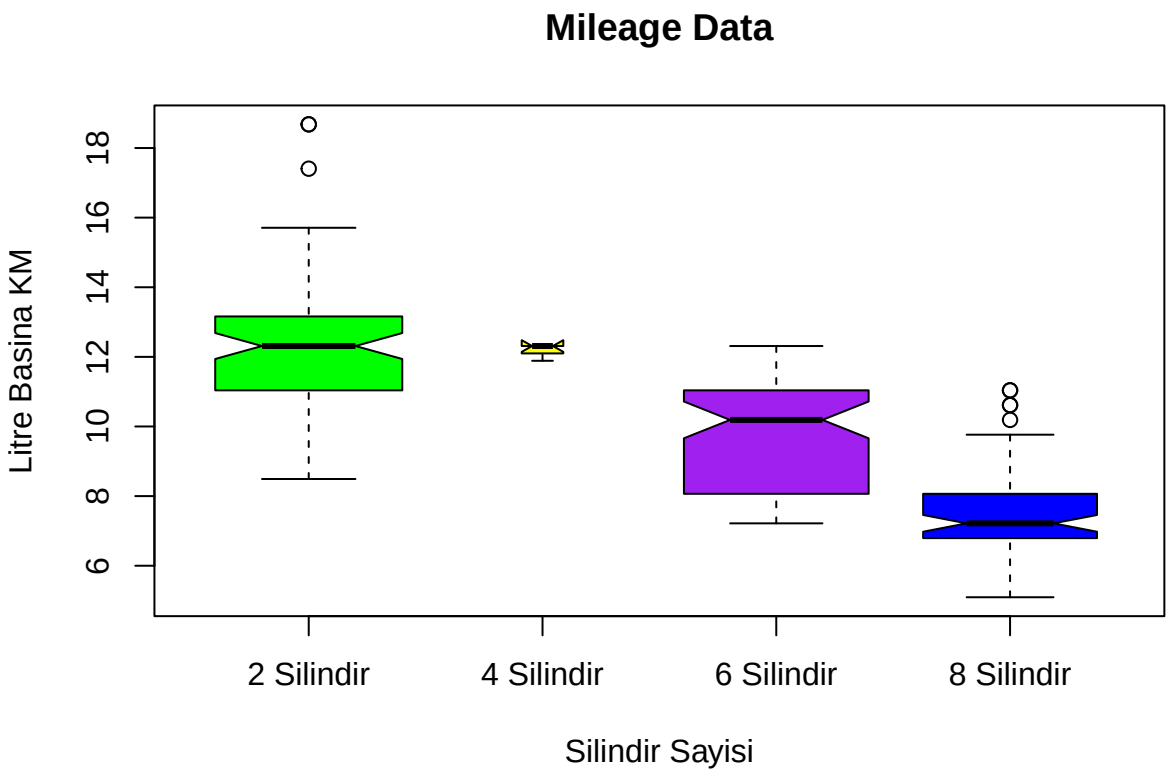
```
## [1]  5.094459  7.641689 10.188918 11.462533 18.679683
```

```
boxplot(hwy_ltkm ~ cyl, data = df, xlab = "Silindir Sayısı",
   ylab = "Litre Başına KM", main = "Mileage Data")
```




```
boxplot(hwy_ltkm ~ cyl, data = df,
   xlab = "Silindir Sayısı",
   ylab = "Litre Başına KM",
   main = "Mileage Data",
   notch = TRUE,
   varwidth = TRUE,
   col = c("green","yellow","purple","blue"),
   names = c("2 Silindir","4 Silindir","6 Silindir","8 Silindir")
)
```

```
## Warning in (function (z, notch = FALSE, width = NULL, varwidth = FALSE, : some
## notches went outside hinges ('box'): maybe set notch=FALSE
```



1.3 Kategorik Tek Değişken

```
# class ve trans değişkenlerine bakalım
# class araç sınıfı, trans ise vites türünü ifade ediyor.

summary(df$class)
```

```
##    2seater    compact    midsize    minivan     pickup subcompact        suv
##          5         47         41         11         33         35         62
```

```
table(df$class)
```

```
##
##    2seater    compact    midsize    minivan     pickup subcompact        suv
##          5         47         41         11         33         35         62
```

```
xtabs(~class,data=df)
```

```
## class
##    2seater    compact    midsize    minivan     pickup subcompact        suv
##          5         47         41         11         33         35         62
```

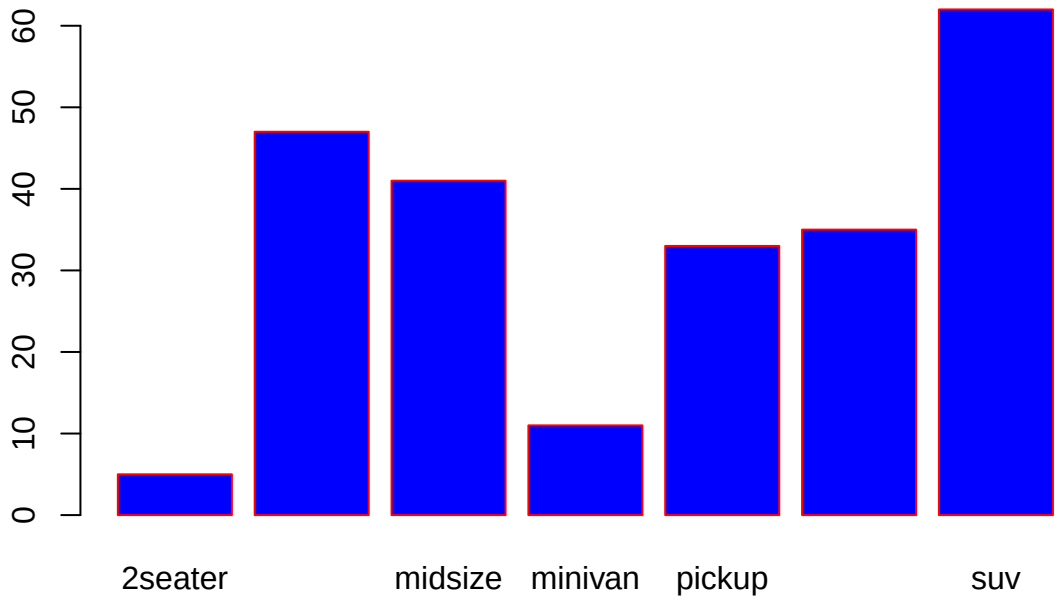
```
table(df$trans)
```

```
##
##   auto(av)   auto(l3)   auto(l4)   auto(l5)   auto(l6)   auto(s4)   auto(s5)
##          5          2         83         39          6          3          3
##   auto(s6) manual(m5) manual(m6)
##         16         58         19
```

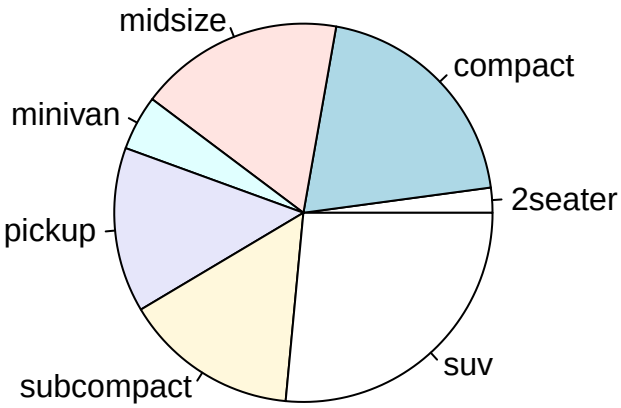
```
prop.table(table(df$class))
```

```
##
##    2seater    compact    midsize    minivan     pickup subcompact        suv
## 0.02136752 0.20085470 0.17521368 0.04700855 0.14102564 0.14957265 0.26495726
```

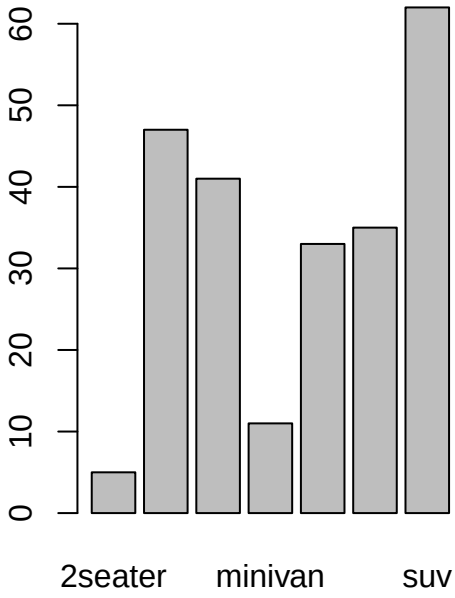
```
tab <- table(df$class)
barplot(tab,col="blue",border="red")
```

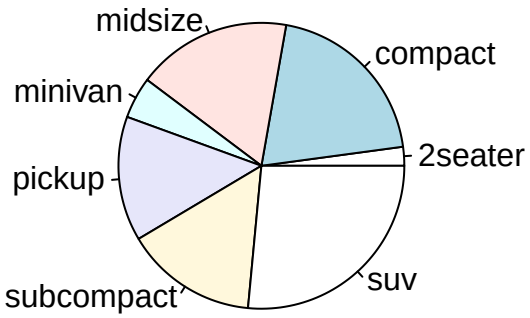



```
pie(tab)
```




```
par(mfrow = c(1, 2))
barplot(tab)
pie(tab)
```





1.4 Sürekli İki Değişken

```
# displ ve cty_ltkm değişkenlerini inceleyelim
# displ motor hacmini ifade ediyor

summary(df$displ)
```

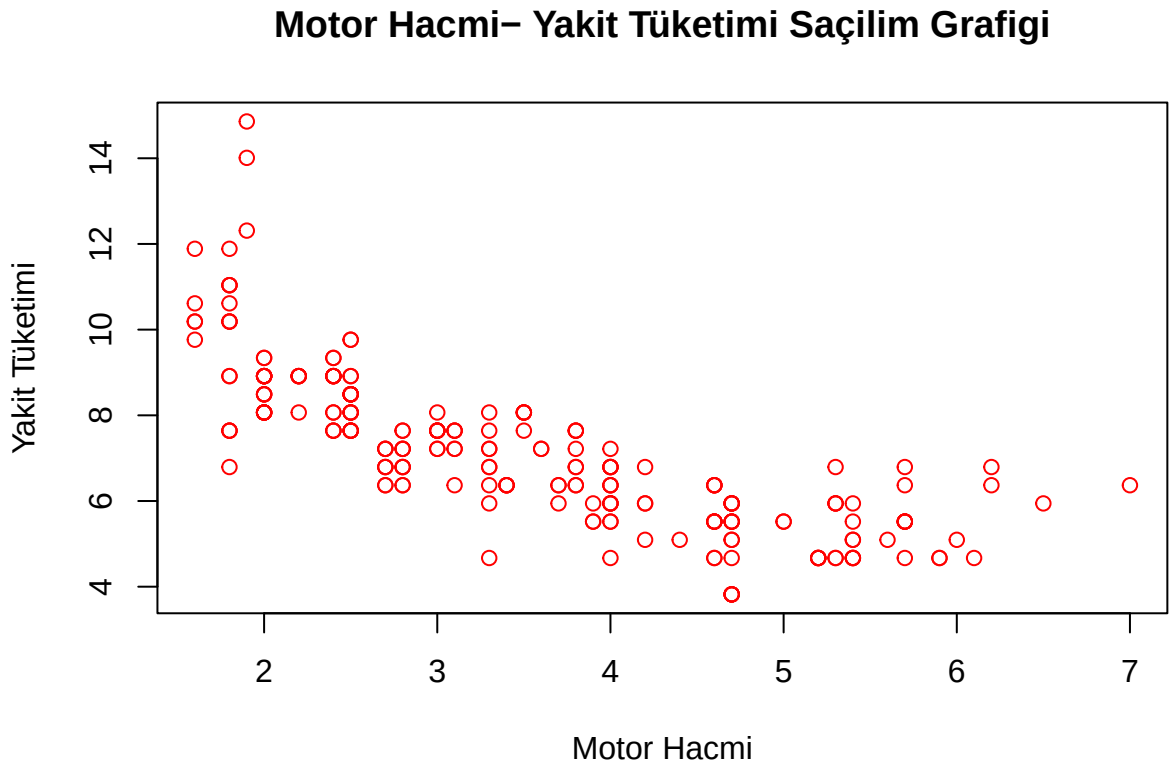
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.600   2.400   3.300   3.472   4.600   7.000
```

```
with(df,cor(displ,cty_ltkm))
```

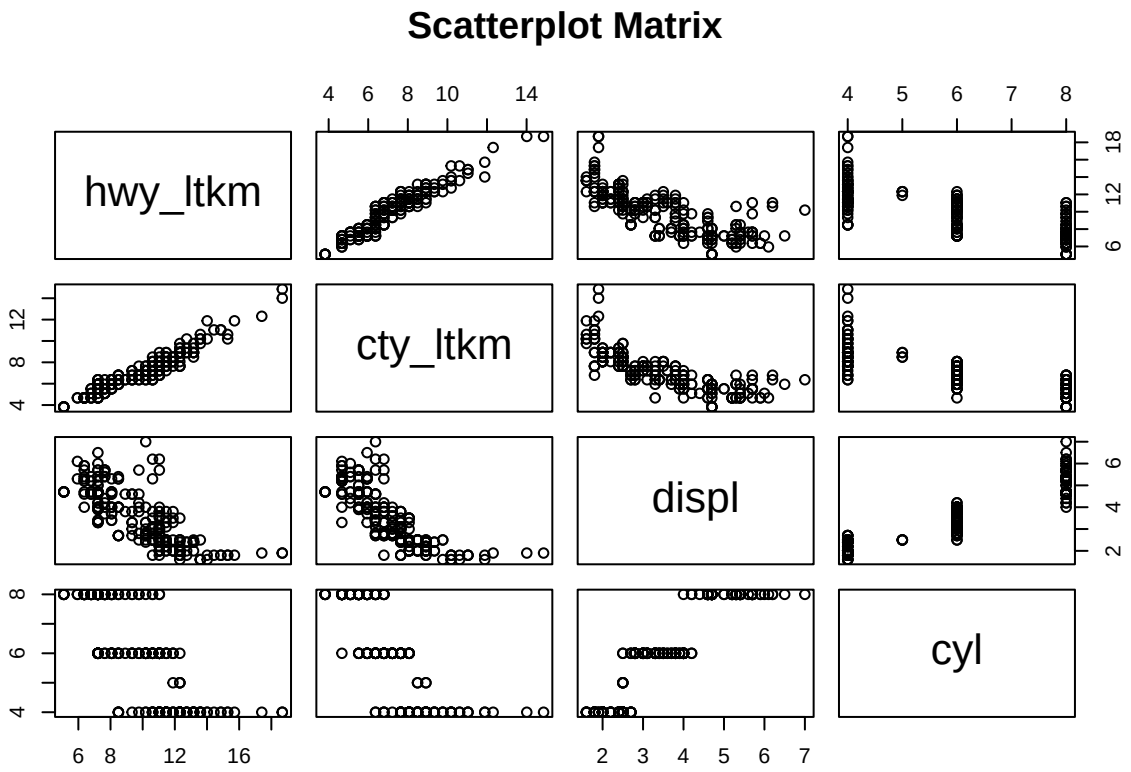
```
## [1] -0.798524
```

```
# motor hacmi ile lt başına km ters ilişkili

plot(df$displ,df$cty_ltkm,
     main = "Motor Hacmi- Yakıt Tüketimi Saçılım Grafiği",
```

```
     col="red",
     xlab = "Motor Hacmi",
     ylab = "Yakıt Tüketimi")
```




```
# birden fazla değişkenin saçılım grafiği
pairs(~hwy_ltkm+cty_ltkm+displ+cyl,data = df,main = "Scatterplot Matrix")
```



1.5 Bir Sürekli Bir Kategorik Değişken

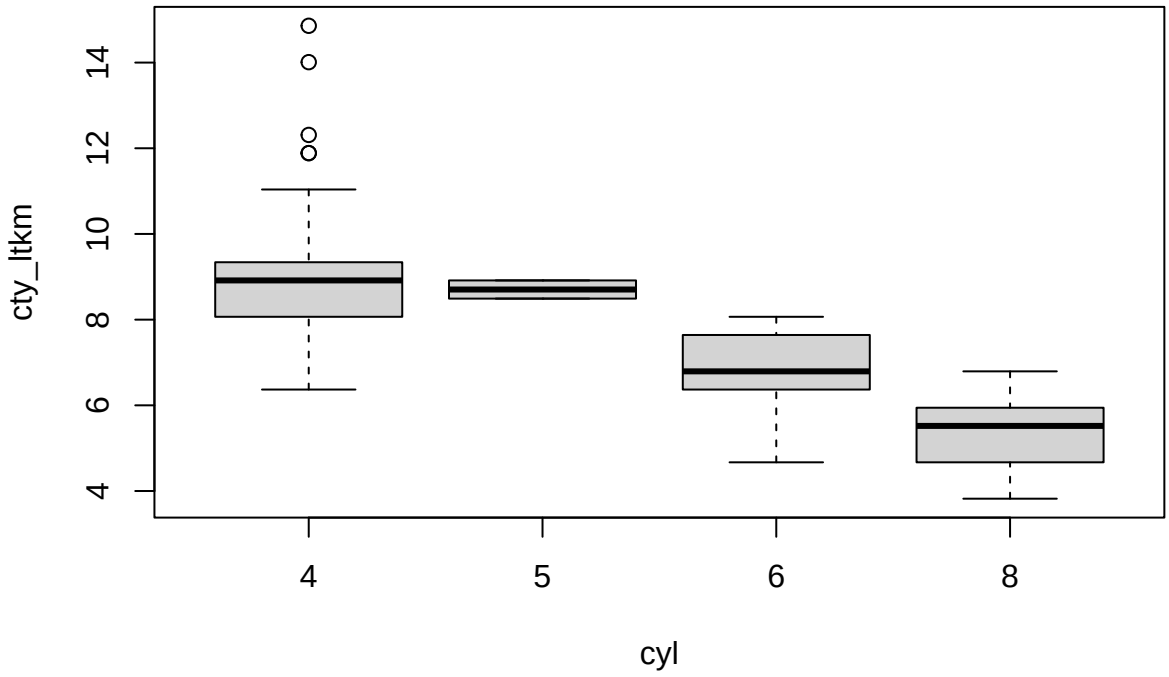
```
# Silindir düzeyinde yakıt tüketimi
tapply(df$cty_ltkm, df$cyl, mean)
```

```
##        4        5        6        8
## 8.920545 8.703034 6.883968 5.337052
```

```
# Same using aggregate()
aggregate(cty_ltkm ~ cyl, data = df, FUN = mean)
```

```
##   cyl cty_ltkm
## 1   4 8.920545
## 2   5 8.703034
## 3   6 6.883968
## 4   8 5.337052
```

```
boxplot(cty_ltkm ~ cyl, data = df)
```



1.6 İki Kategorik Değişken

```
xtabs(~trans+class,data=df)
```

```
##              class
## trans         2seater compact midsize minivan pickup subcompact suv
##   auto(av)          0       2       3       0      0          0   0
##   auto(l3)          0       1       0       1      0          0   0
##   auto(l4)          1       8      14       8     12         11  29
##   auto(l5)          0       4       5       0      8          4  18
##   auto(l6)          0       0       0       2      0          0   4
##   auto(s4)          0       2       1       0      0          0   0
##   auto(s5)          0       2       0       0      0          0   1
##   auto(s6)          1       5       6       0      0          1   3
##   manual(m5)        0      18       9       0      8         16   7
##   manual(m6)        3       5       3       0      5          3   0
```

```
prop.table(table(df$year,df$class),1) # satır toplamları 1' eşittir
```

```
##
##           2seater    compact    midsize    minivan     pickup subcompact
##   1999 0.01709402 0.21367521 0.17094017 0.05128205 0.13675214 0.16239316
##   2008 0.02564103 0.18803419 0.17948718 0.04273504 0.14529915 0.13675214
##
##               suv
##   1999 0.24786325
##   2008 0.28205128
```

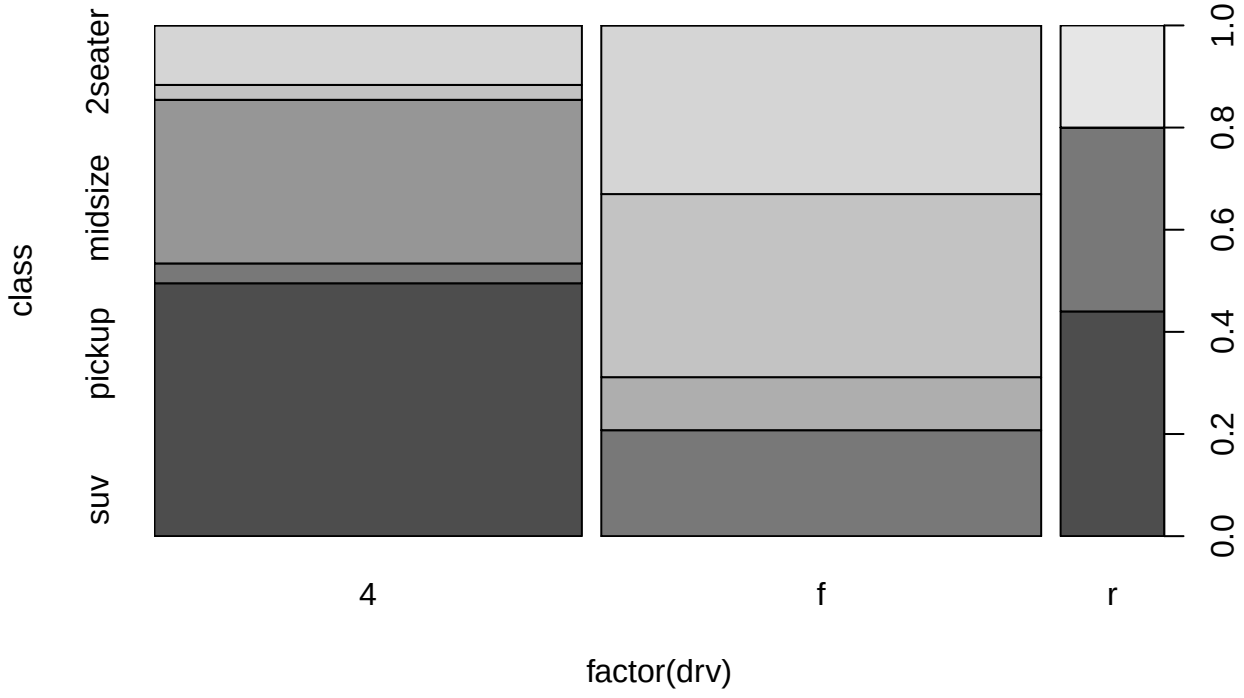
```
prop.table(table(df$year,df$class),2) # sütun toplamları 1' eşittir
```

```
##
##          2seater   compact   midsize   minivan    pickup subcompact       suv
##   1999 0.4000000 0.5319149 0.4878049 0.5454545 0.4848485  0.5428571 0.4677419
##   2008 0.6000000 0.4680851 0.5121951 0.4545455 0.5151515  0.4571429 0.5322581
```

```
proportions(xtabs(~ manufacturer + year, data = df), 1)
```

```
##             year
## manufacturer      1999      2008
##   audi       0.5000000 0.5000000
##   chevrolet  0.3684211 0.6315789
##   dodge      0.4324324 0.5675676
##   ford       0.6000000 0.4000000
##   honda      0.5555556 0.4444444
##   hyundai    0.4285714 0.5714286
##   jeep       0.2500000 0.7500000
##   land rover 0.5000000 0.5000000
##   lincoln    0.6666667 0.3333333
##   mercury    0.5000000 0.5000000
##   nissan     0.4615385 0.5384615
##   pontiac    0.6000000 0.4000000
##   subaru     0.4285714 0.5714286
##   toyota     0.5882353 0.4117647
##   volkswagen 0.5925926 0.4074074
```

```
# araç sınıfı ile drv değişkenine birlikte bakalım
# f = front-wheel drive (önden çekiş),
# r = rear wheel drive (arkadan çekiş),
# 4 = 4wd (4 çeker)

plot(class ~ factor(drv), data = df)
```



1.7 Zaman Serisi

```
AirPassengers
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1949 112 118 132 129 121 135 148 148 136 119 104 118
## 1950 115 126 141 135 125 149 170 170 158 133 114 140
## 1951 145 150 178 163 172 178 199 199 184 162 146 166
## 1952 171 180 193 181 183 218 230 242 209 191 172 194
## 1953 196 196 236 235 229 243 264 272 237 211 180 201
## 1954 204 188 235 227 234 264 302 293 259 229 203 229
## 1955 242 233 267 269 270 315 364 347 312 274 237 278
## 1956 284 277 317 313 318 374 413 405 355 306 271 306
## 1957 315 301 356 348 355 422 465 467 404 347 305 336
## 1958 340 318 362 348 363 435 491 505 404 359 310 337
## 1959 360 342 406 396 420 472 548 559 463 407 362 405
## 1960 417 391 419 461 472 535 622 606 508 461 390 432
```

```
class(AirPassengers)
```

```
## [1] "ts"
```

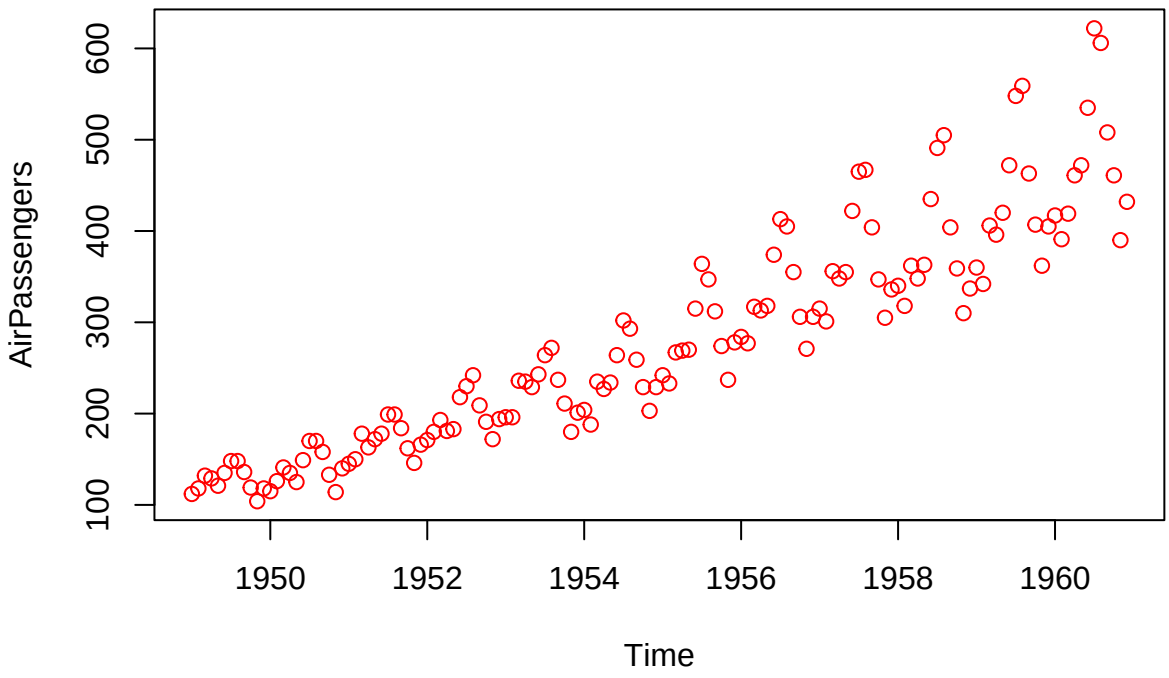
```
diff(AirPassengers) # fark alma
```

```
##       Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1949            6   14   -3   -8   14   13    0  -12  -17  -15   14
## 1950   -3   11   15   -6  -10   24   21    0  -12  -25  -19   26
## 1951    5    5   28  -15    9    6   21    0  -15  -22  -16   20
## 1952    5    9   13  -12    2   35   12   12  -33  -18  -19   22
## 1953    2    0   40   -1   -6   14   21    8  -35  -26  -31   21
## 1954    3  -16   47   -8    7   30   38   -9  -34  -30  -26   26
## 1955   13   -9   34    2    1   45   49  -17  -35  -38  -37   41
## 1956    6   -7   40   -4    5   56   39   -8  -50  -49  -35   35
## 1957    9  -14   55   -8    7   67   43    2  -63  -57  -42   31
## 1958    4  -22   44  -14   15   72   56   14 -101  -45  -49   27
## 1959   23  -18   64  -10   24   52   76   11  -96  -56  -45   43
## 1960   12  -26   28   42   11   63   87  -16  -98  -47  -71   42
```

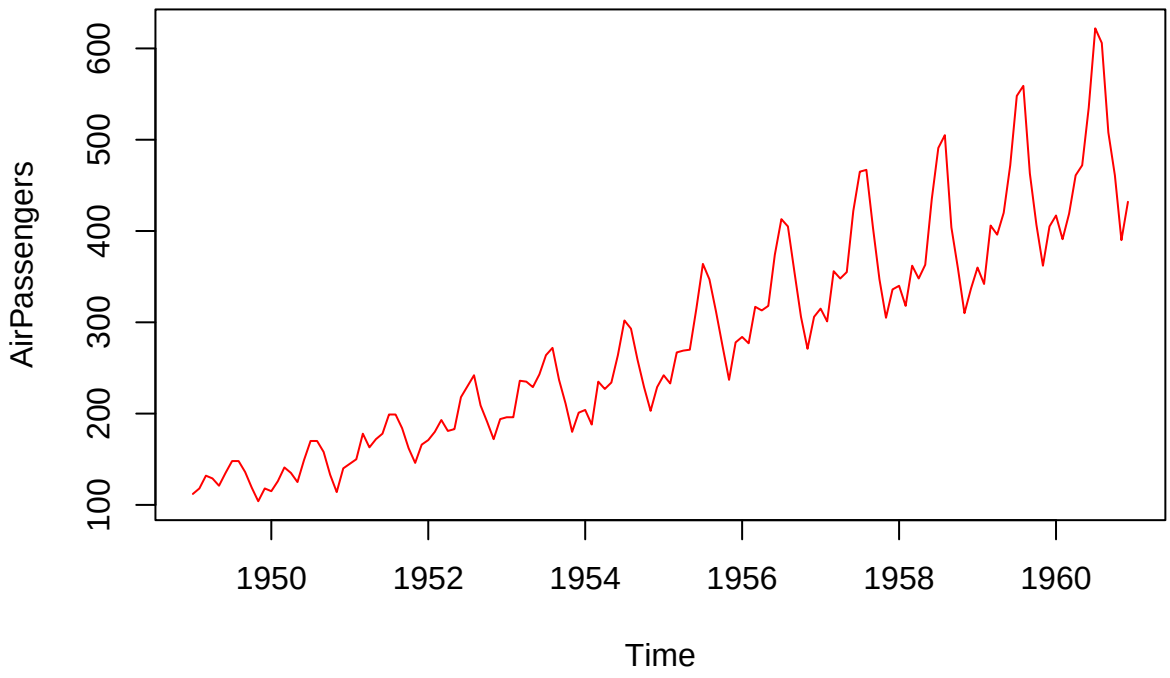
```
stats::lag(AirPassengers,-1) # 1. gecikmesini alma
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1949     112 118 132 129 121 135 148 148 136 119 104
## 1950 118 115 126 141 135 125 149 170 170 158 133 114
## 1951 140 145 150 178 163 172 178 199 199 184 162 146
## 1952 166 171 180 193 181 183 218 230 242 209 191 172
## 1953 194 196 196 236 235 229 243 264 272 237 211 180
## 1954 201 204 188 235 227 234 264 302 293 259 229 203
## 1955 229 242 233 267 269 270 315 364 347 312 274 237
## 1956 278 284 277 317 313 318 374 413 405 355 306 271
## 1957 306 315 301 356 348 355 422 465 467 404 347 305
## 1958 336 340 318 362 348 363 435 491 505 404 359 310
## 1959 337 360 342 406 396 420 472 548 559 463 407 362
## 1960 405 417 391 419 461 472 535 622 606 508 461 390
## 1961 432
```

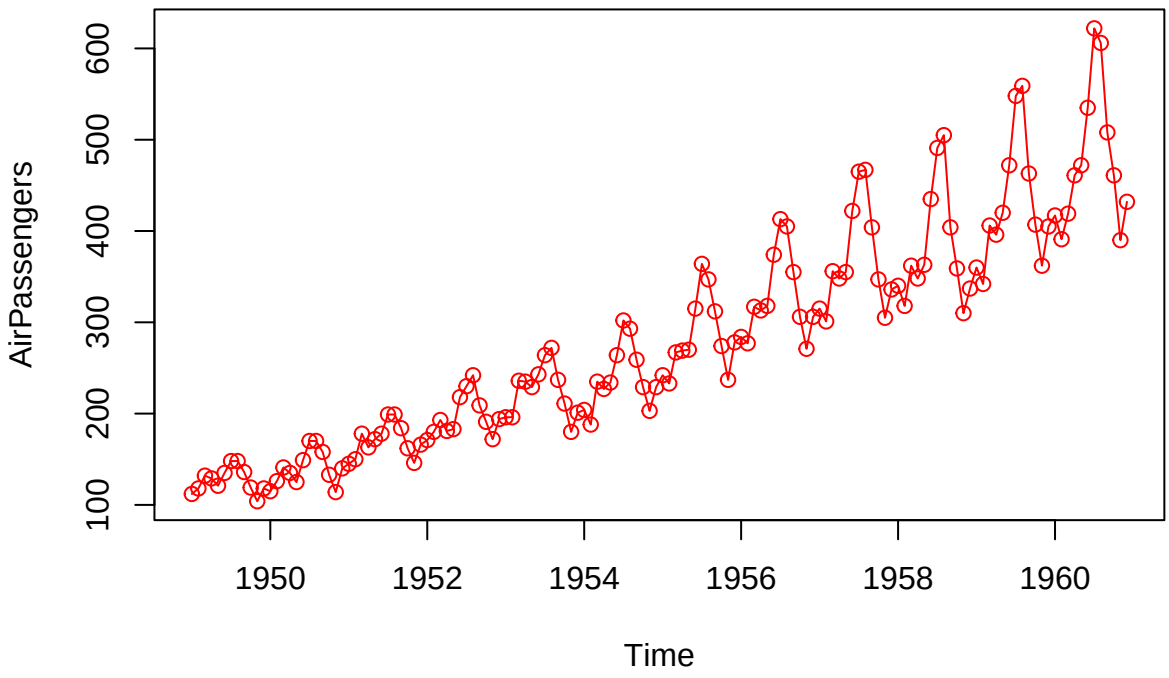
```
plot(AirPassengers,type = "p", col = "red") # points
```



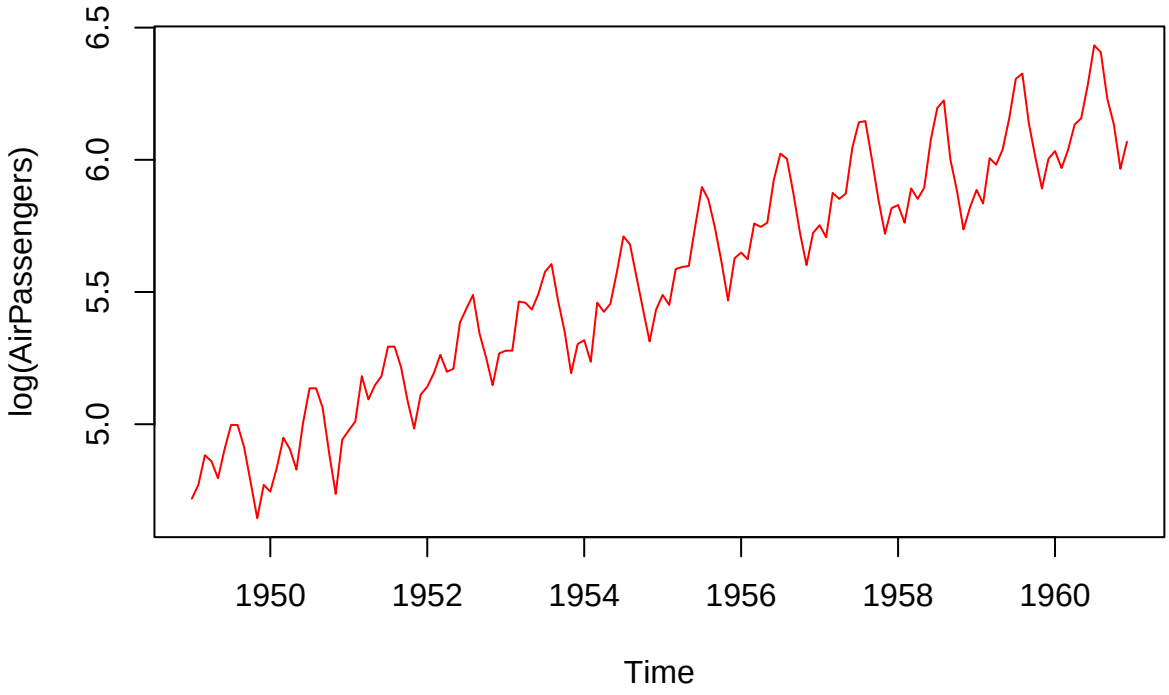

```
plot(AirPassengers,type = "l", col = "red") # line
```



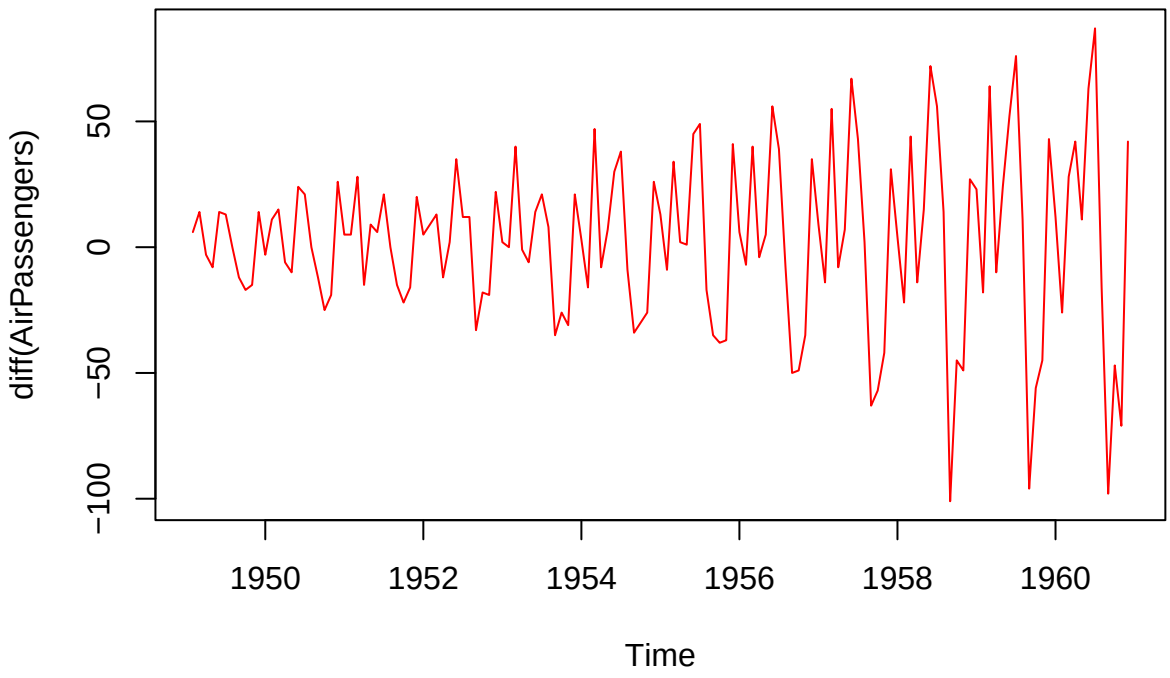

```
plot(AirPassengers,type = "o", col = "red") # points and line
```



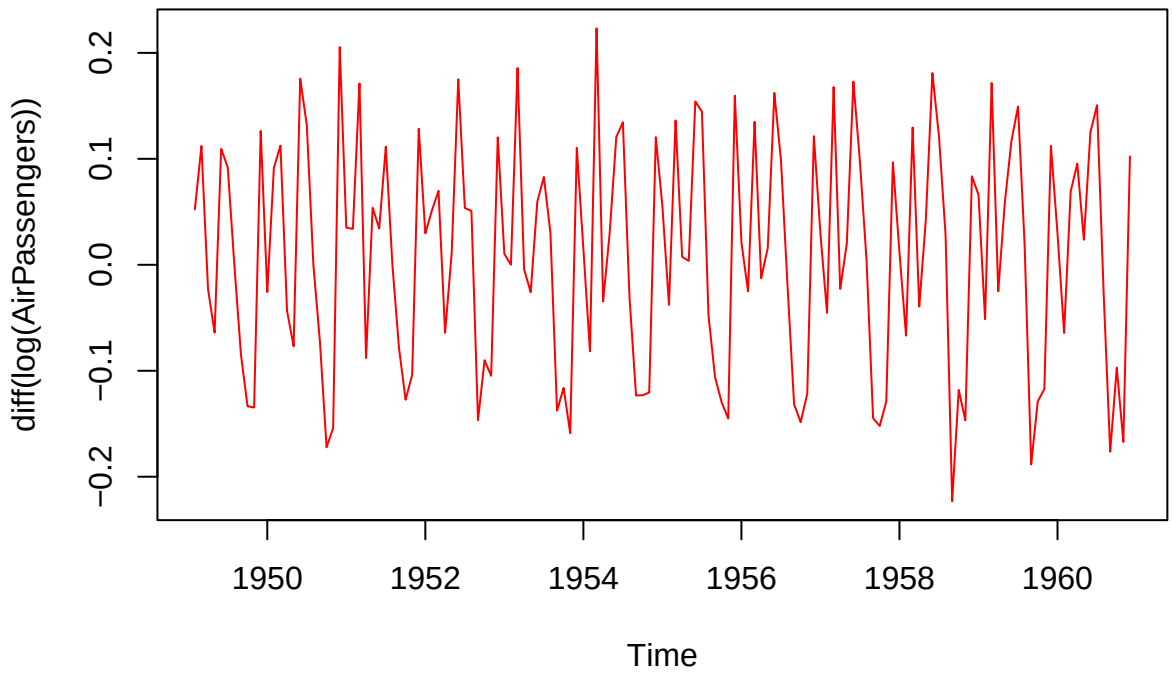

```
plot(log(AirPassengers),type = "l", col = "red") # line
```




```
plot(diff(AirPassengers),type = "l", col = "red") # line
```




```
plot(diff(log(AirPassengers)),type = "l", col = "red") # line
```

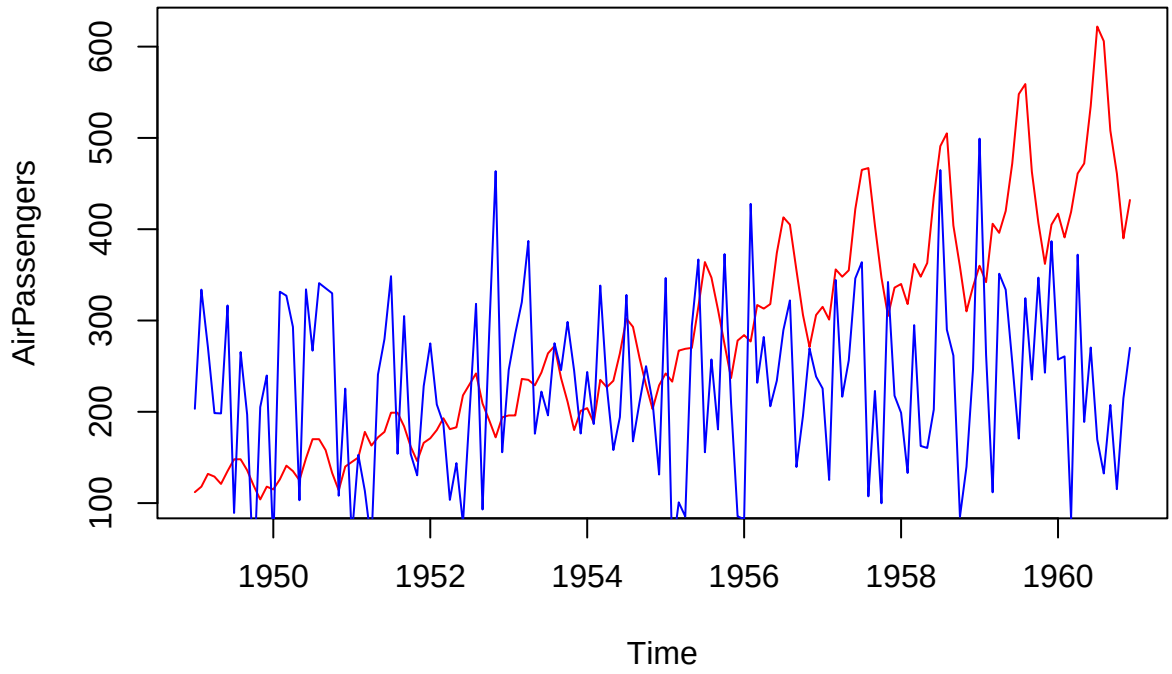



```
# çoklu zaman serisi
ts <- ts(rnorm(length(AirPassengers),250,100),start = c(1949,1),frequency=12)
ts
```

```
##             Jan       Feb       Mar       Apr       May       Jun       Jul
## 1949 203.31848 333.82744 270.78735 198.57787 198.23439 316.49100  89.13946
## 1950  53.80126 331.53917 327.11626 292.96899 103.11414 334.14624 266.96449
## 1951  62.33192 152.83032 113.33284  55.67581 240.70978 279.78878 348.55591
## 1952 274.93190 208.03350 187.25227 103.48012 143.78557  77.84702 197.73105
## 1953 245.95259 285.68621 320.12904 387.16013 175.98127 222.12732 196.07727
## 1954 243.60271 186.58181 338.29303 227.49447 158.09989 194.22725 328.01948
## 1955 346.59356  45.44663 100.81030  85.30848 294.10077 366.83967 155.68778
## 1956  82.33896 427.74568 231.79295 281.97048 206.04296 234.10436 289.46833
## 1957 225.54603 125.30023 344.42958 216.51280 256.34917 346.17421 363.86568
## 1958 198.85214 132.96318 294.97500 162.56757 160.40372 202.45298 464.79507
## 1959 499.24938 262.51232 111.79473 351.24906 333.72574 254.51394 170.65714
## 1960 257.22031 260.69409  83.11639 372.10196 188.83316 270.40711 169.63520
##             Aug       Sep       Oct       Nov       Dec
## 1949 265.45690 196.90419  13.38500 205.11449 239.74799
## 1950 340.97963 335.36794 329.81562 108.05755 225.38383
## 1951 153.96254 304.86259 154.00713 130.42410 228.15230
```

```
## 1952 318.24125  92.97717 284.35375 463.62235 155.69854
## 1953 275.09880 245.65195 298.47982 245.05851 176.18540
## 1954 167.63749 210.33893 249.95129 210.84211 131.26871
## 1955 257.35555 180.70905 372.76562 209.62248  85.47469
## 1956 322.02410 139.52534 195.64538 269.40825 238.62965
## 1957 107.36454 222.84647  99.77136 342.21172 217.63886
## 1958 289.95803 261.55708  84.98523 139.81366 246.55151
## 1959 324.47215 235.28052 346.99891 242.83463 386.88484
## 1960 132.43699 207.50528 115.22319 214.84222 270.04767
```

```
plot(AirPassengers,type = "l",col = "red")
lines(ts, type = "l", col = "blue")
```

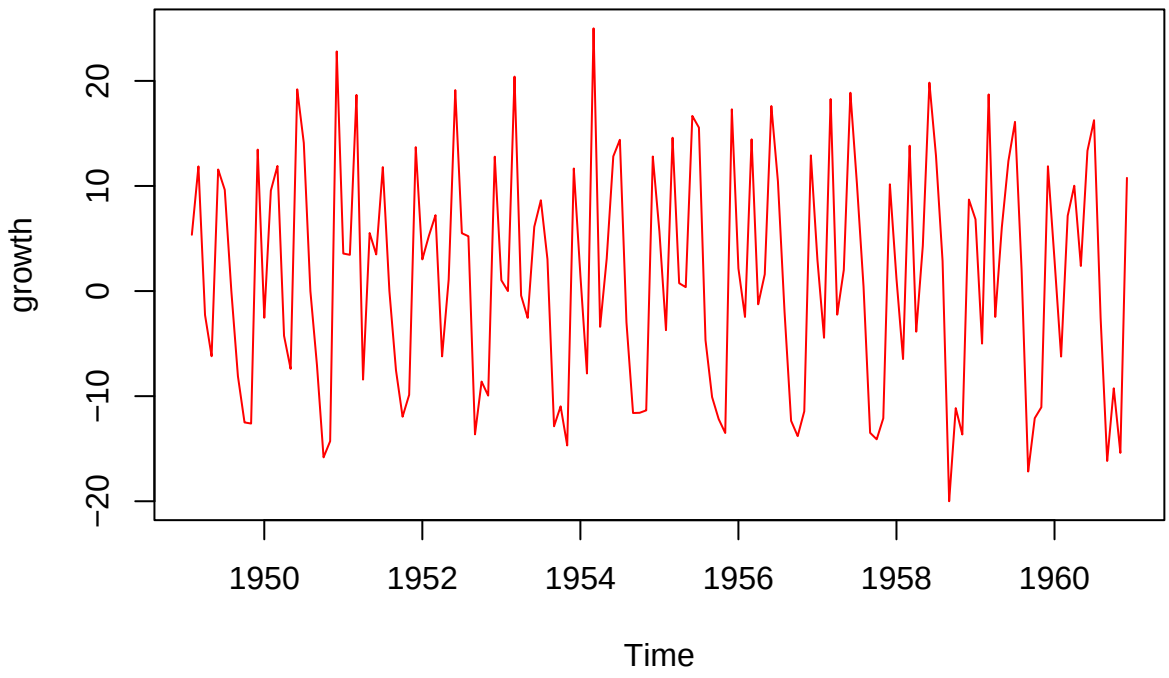



```
# yüzde değişim
growth <- AirPassengers/stats::lag(AirPassengers,-1)*100-100
growth
```

```
##              Jan         Feb         Mar         Apr         May         Jun
## 1949                5.3571429  11.8644068  -2.2727273  -6.2015504  11.5702479
## 1950  -2.5423729   9.5652174  11.9047619  -4.2553191  -7.4074074  19.2000000
```

```
## 1951   3.5714286   3.4482759  18.6666667  -8.4269663   5.5214724   3.4883721
## 1952   3.0120482   5.2631579   7.2222222  -6.2176166   1.1049724  19.1256831
## 1953   1.0309278   0.0000000  20.4081633  -0.4237288  -2.5531915   6.1135371
## 1954   1.4925373  -7.8431373  25.0000000  -3.4042553   3.0837004  12.8205128
## 1955   5.6768559  -3.7190083  14.5922747   0.7490637   0.3717472  16.6666667
## 1956   2.1582734  -2.4647887  14.4404332  -1.2618297   1.5974441  17.6100629
## 1957   2.9411765  -4.4444444  18.2724252  -2.2471910   2.0114943  18.8732394
## 1958   1.1904762  -6.4705882  13.8364780  -3.8674033   4.3103448  19.8347107
## 1959   6.8249258  -5.0000000  18.7134503  -2.4630542   6.0606061  12.3809524
## 1960   2.9629630  -6.2350120   7.1611253  10.0238663   2.3861171  13.3474576
##              Jul         Aug         Sep         Oct         Nov         Dec
## 1949   9.6296296   0.0000000  -8.1081081 -12.5000000 -12.6050420  13.4615385
## 1950  14.0939597   0.0000000  -7.0588235 -15.8227848 -14.2857143  22.8070175
## 1951  11.7977528   0.0000000  -7.5376884 -11.9565217  -9.8765432  13.6986301
## 1952   5.5045872   5.2173913 -13.6363636  -8.6124402  -9.9476440  12.7906977
## 1953   8.6419753   3.0303030 -12.8676471 -10.9704641 -14.6919431  11.6666667
## 1954  14.3939394  -2.9801325 -11.6040956 -11.5830116 -11.3537118  12.8078818
## 1955  15.5555556  -4.6703297 -10.0864553 -12.1794872 -13.5036496  17.2995781
## 1956  10.4278075  -1.9370460 -12.3456790 -13.8028169 -11.4379085  12.9151292
## 1957  10.1895735   0.4301075 -13.4903640 -14.1089109 -12.1037464  10.1639344
## 1958  12.8735632   2.8513238 -20.0000000 -11.1386139 -13.6490251   8.7096774
## 1959  16.1016949   2.0072993 -17.1735242 -12.0950324 -11.0565111  11.8784530
## 1960  16.2616822  -2.5723473 -16.1716172  -9.2519685 -15.4013015  10.7692308
```

```
plot(growth,type = "l", col = "red")
```



2 ggplot2 ile Veri Görselleştirme

Bu bölümde ggplot2 paketi ile verilerin nasıl görselleştirldiğine bakacağız. ggplot2 grafiklerin dil bilgisi **(grammar of graphics)** prensiplerini temel alarak oluşturulmuştur. Bu prensiplere göre her grafik aynı parçalardan oluşturulabilir: bir veri seti, koordinat sistemi, ve "geom"lar - veri noktalarını temsil eden görsel işaretler.

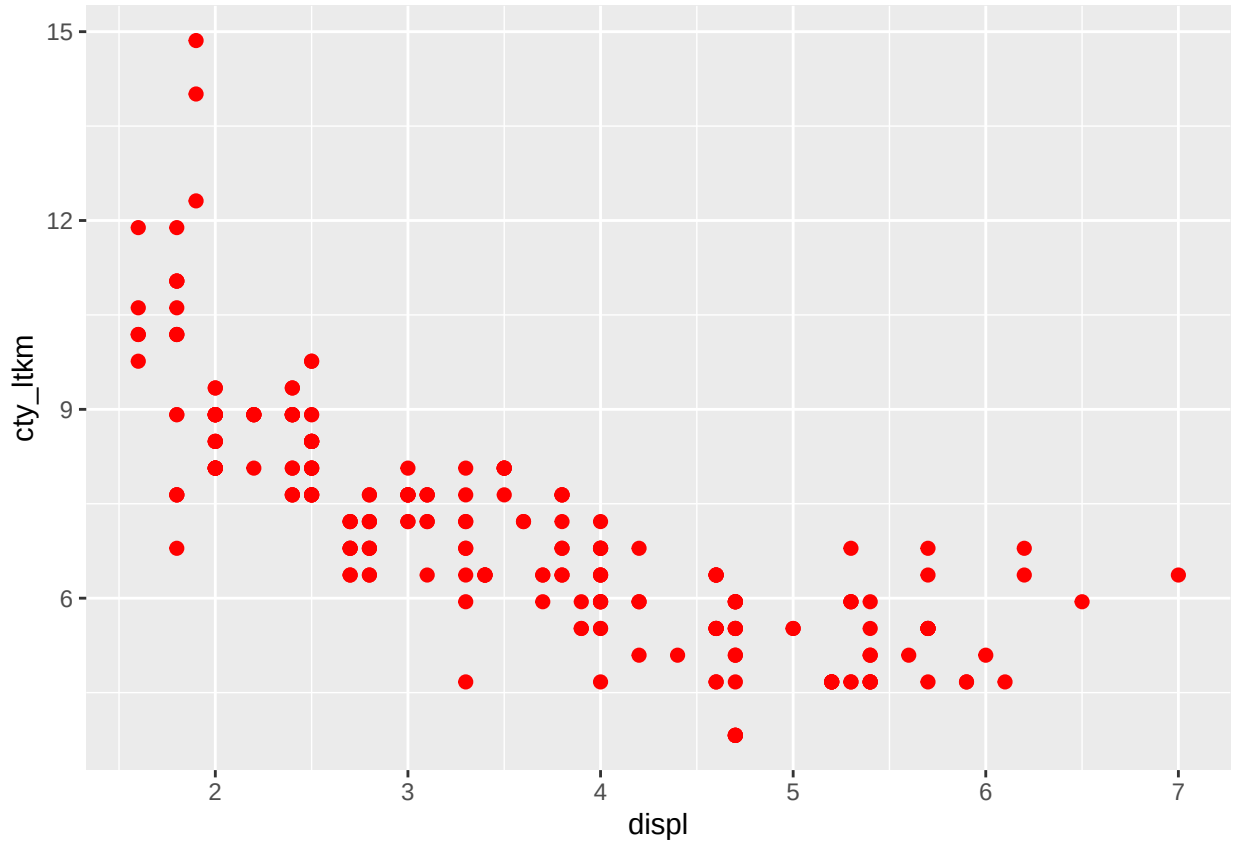
ggplot2 ile veri görselleştirebilmemiz için önce grafik yapısını iyi tanımamız gerekiyor. Yatay eksen x ekseni, dikey eksen ise y ekseni olarak kabul ediliyor. Veri görselleştirmede **`ggplot()`** fonksiyonunu kullanıyoruz. ggplot() fonksiyonu içinde veri seti ismi ve **`aes()`** adlı estetik argümanına yatay ve dikey eksende kullanacağımız değişkenler (sütun isimleri) ile yer veriyoruz. Sonrasında, tercih edeceğimiz grafik tipine göre, **`geom`** fonksiyonlarından birini kullanacağız. Sıklıkla kullanılan geom fonksiyonları şunlardır:

- Nokta grafiği için `geom_point()`
- Çubuk veya sütun grafik için `geom_col()` ve `geom_bar()`
- Çizgi grafiği için `geom_line()`
- Histogram grafiği için `geom_histogram()`

- Boxplot grafiği için `geom_boxplot()`

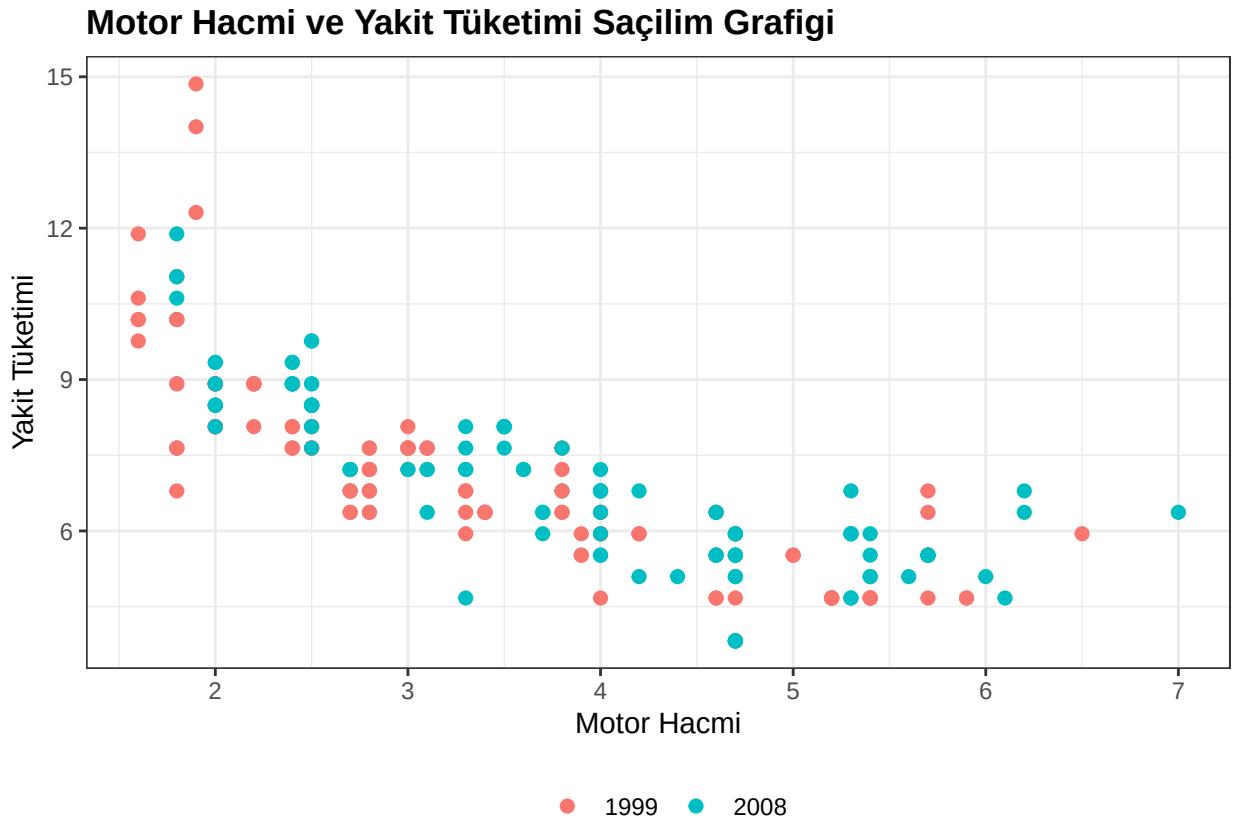
2.1 Saçılım Grafikleri

```
library(ggplot2)
library(dplyr)

p1 <- ggplot(df,aes(x=displ,y=cty_ltkm)) +
  geom_point(size=2,color="red")
p1
```



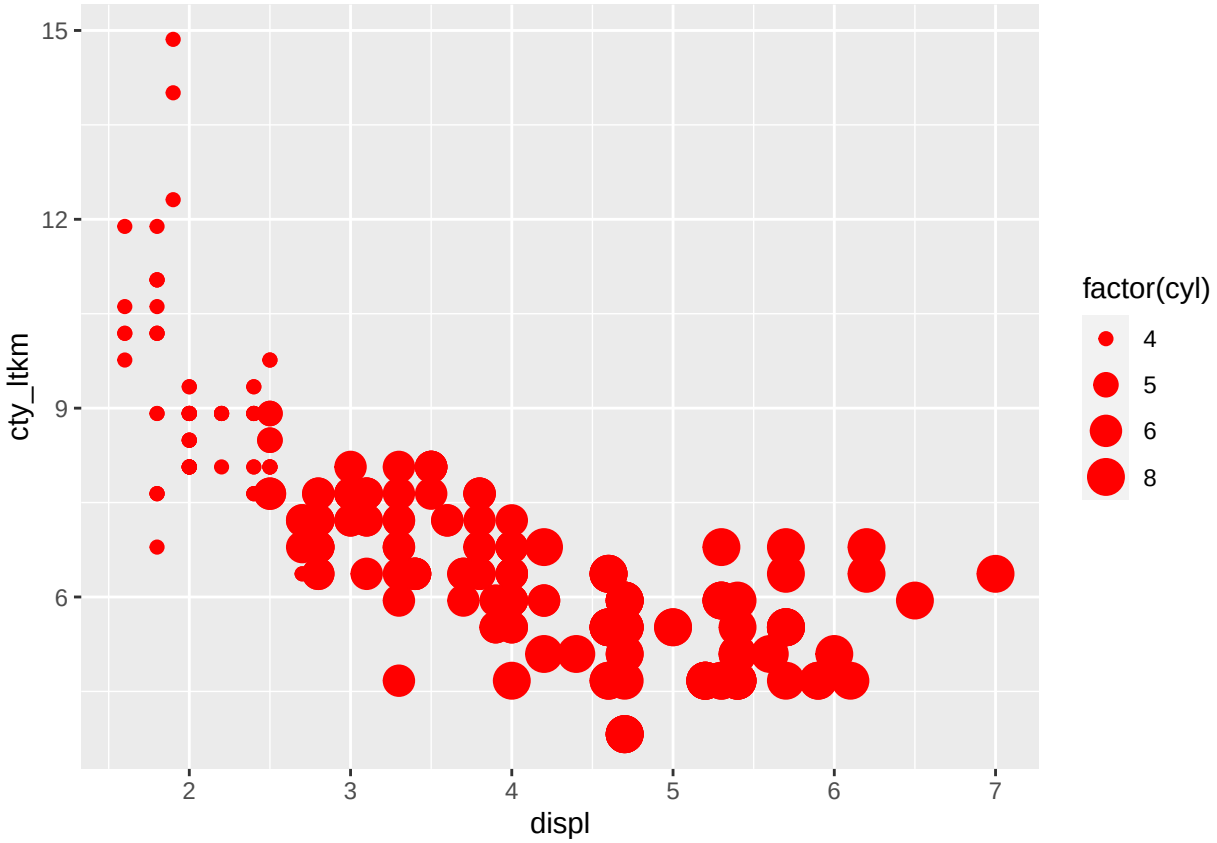

```
# gruplar düzeyinde grafiği çizdirme
p2 <- ggplot(df,aes(x=displ,y=cty_ltkm,colour=as.factor(year))) +
  geom_point(size=2) +
  # grafiğe başlık ekleme
  ggtitle("Motor Hacmi ve Yakıt Tüketimi Saçılım Grafiği") +
  #eksenleri isimlendirme
  xlab("Motor Hacmi") +
  ylab("Yakıt Tüketimi")+
```

```
  theme_bw() + # tema değiştirme
  theme(legend.position = "bottom",  # gruplama değişkeninin poziyounun değiştirme
        plot.title = element_text(face = "bold"), # kalın başlık
        legend.title = element_blank()) # grup başlığını kaldırma
p2
```

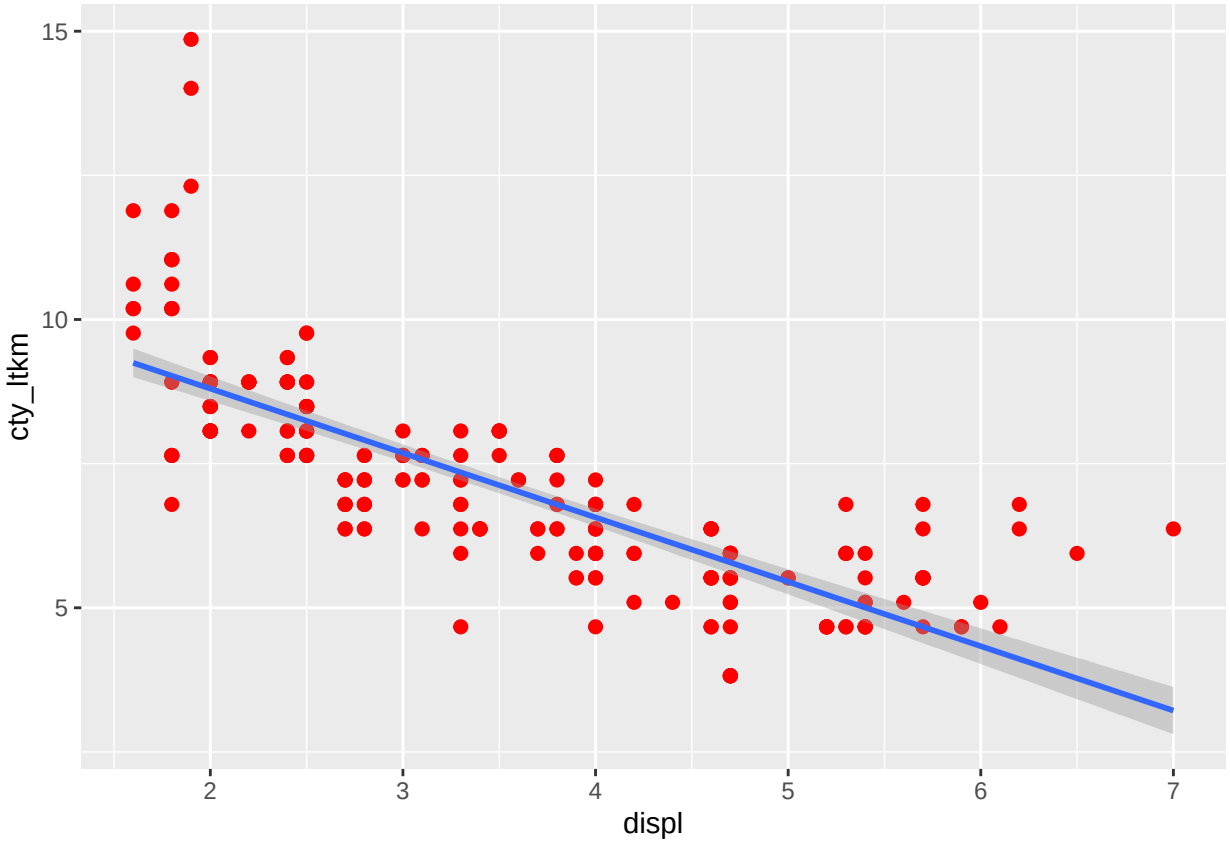



```
ggplot(df,aes(x=displ,y=cty_ltkm)) +
  geom_point(aes(size=factor(cyl)),color="red")
```

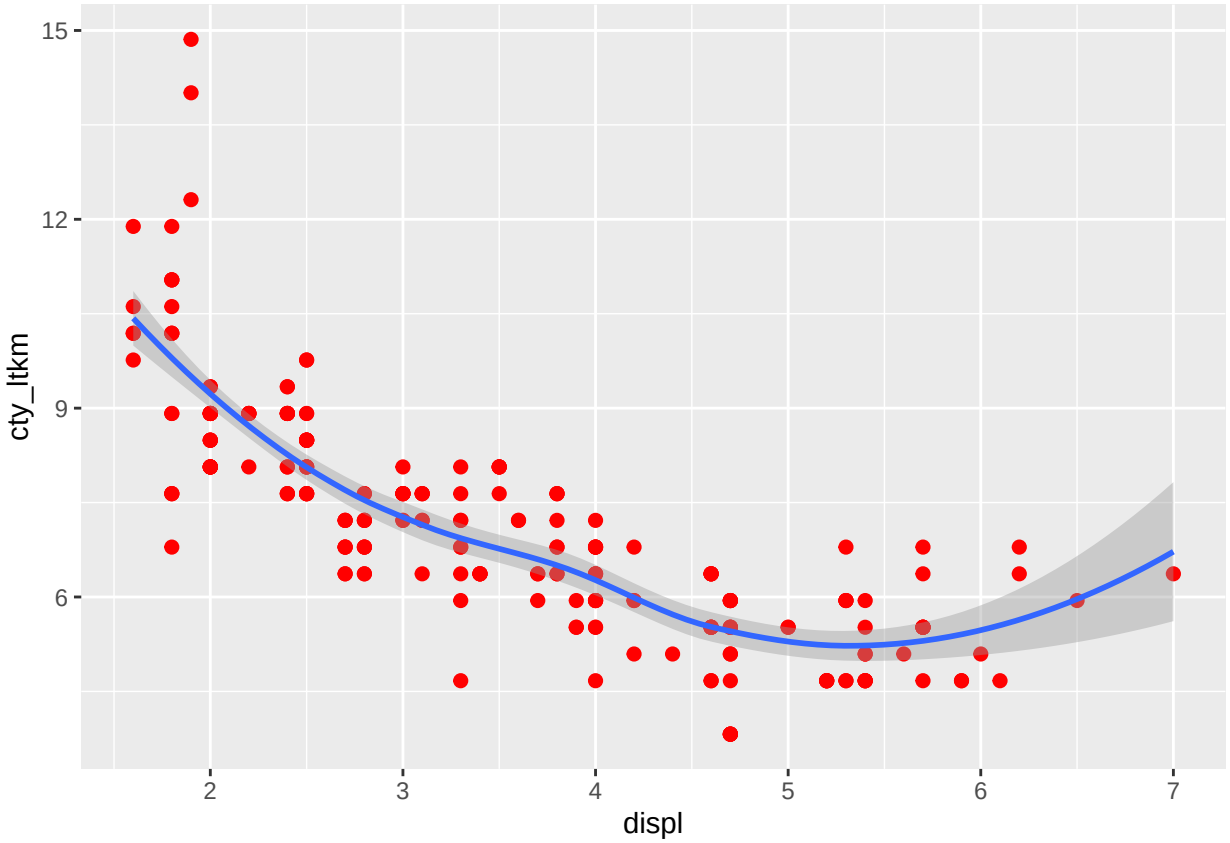
```
## Warning: Using size for a discrete variable is not advised.
```



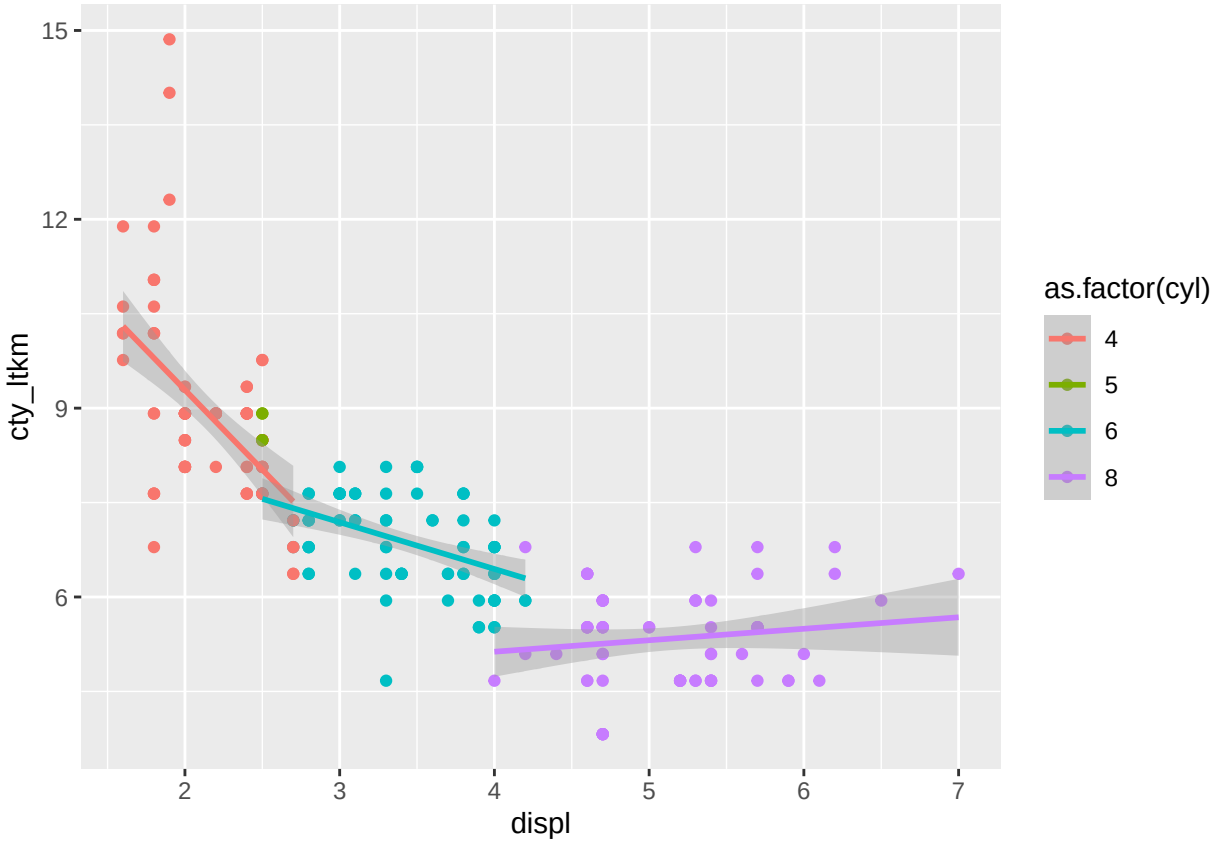

```
# grafiğe model eğrisi ekleme
p1 + geom_smooth(method = lm, se = TRUE)
```




```
p1 + geom_smooth(method = loess, se = TRUE)
```

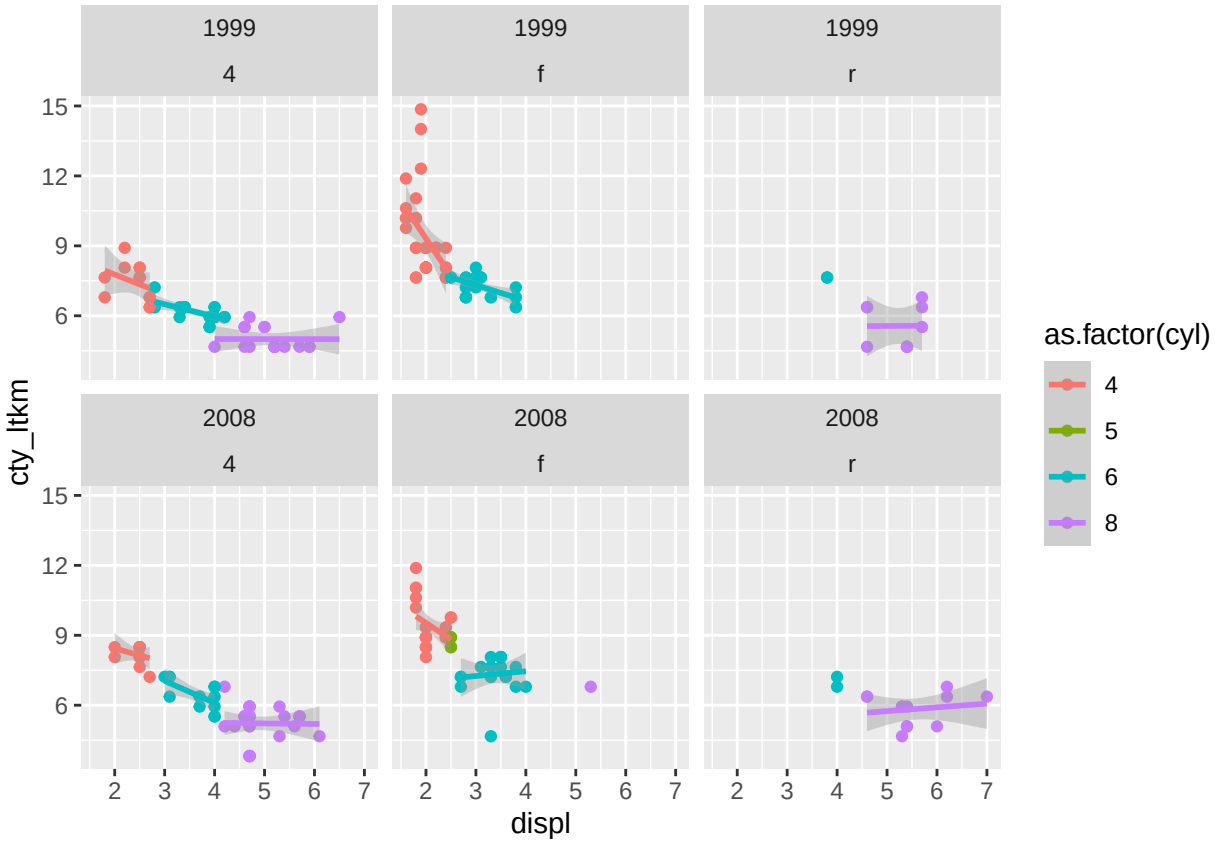



```
# grup düzeyinde model eğrileri ve saçılım grafiği
p3 <- df %>%
  ggplot(aes(x=displ,y=cty_ltkm,color=as.factor(cyl))) +
  geom_point()  +
  geom_smooth(method = lm, se = TRUE)
p3
```

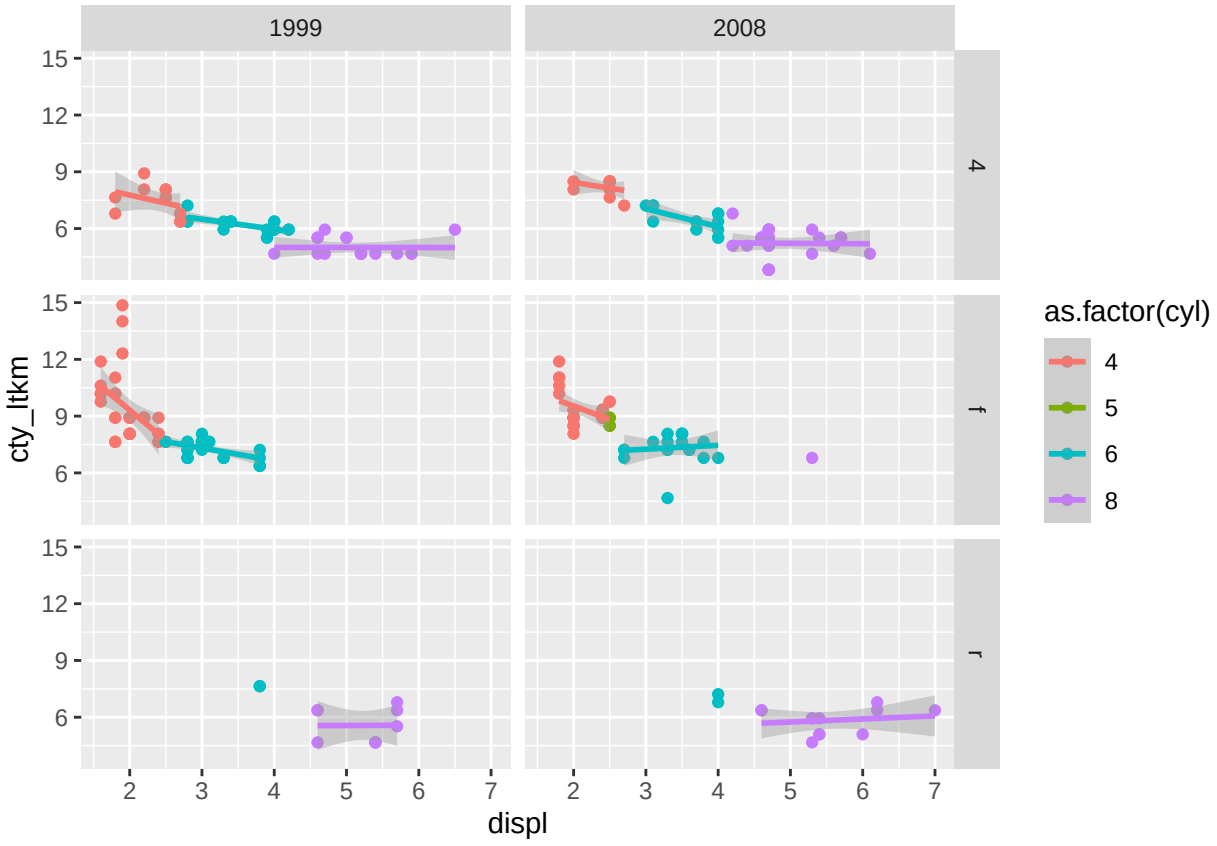



```
# grup ve yıl düzeyinde model eğrileri ve saçılım grafiği
p3 + facet_wrap(~ year)
```

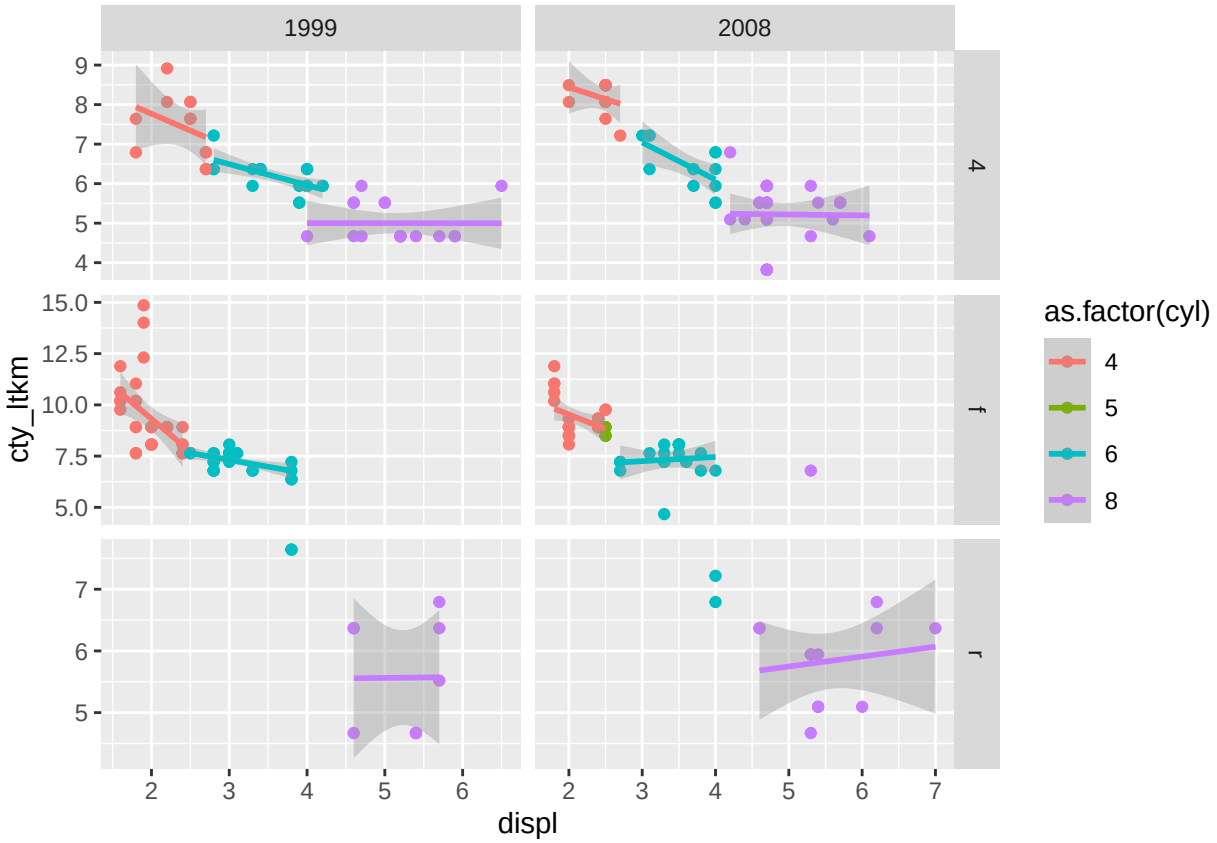

```
p3 + facet_wrap(~ year+drv)
```




```
p3 + facet_grid(drv ~ year) # eksen aralıkları sabit
```




```
p3 + facet_grid(drv ~ year,scales = "free") # eksen aralıkları değişken
```



2.2 Zaman Serisi Grafikleri

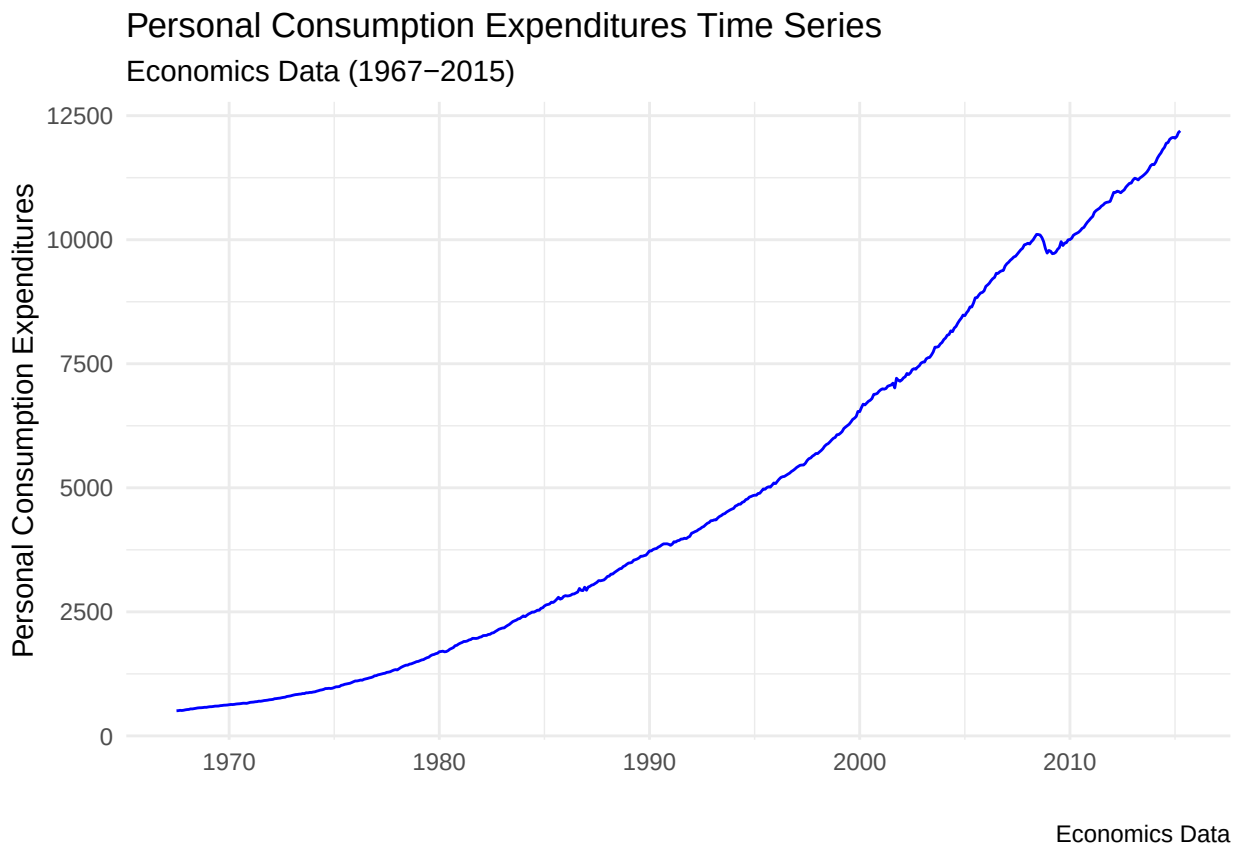
```
economics
```

```
## # A tibble: 574 x 6
##    date         pce    pop psavert uempmed unemploy
##    <date>     <dbl>  <dbl>   <dbl>   <dbl>    <dbl>
##  1 1967-07-01  507. 198712    12.6     4.5     2944
##  2 1967-08-01  510. 198911    12.6     4.7     2945
##  3 1967-09-01  516. 199113    11.9     4.6     2958
##  4 1967-10-01  512. 199311    12.9     4.9     3143
##  5 1967-11-01  517. 199498    12.8     4.7     3066
##  6 1967-12-01  525. 199657    11.8     4.8     3018
##  7 1968-01-01  531. 199808    11.7     5.1     2878
##  8 1968-02-01  534. 199920    12.3     4.5     3001
##  9 1968-03-01  544. 200056    11.7     4.1     2877
## 10 1968-04-01  544  200208    12.3     4.6     2709
## # ... with 564 more rows
```

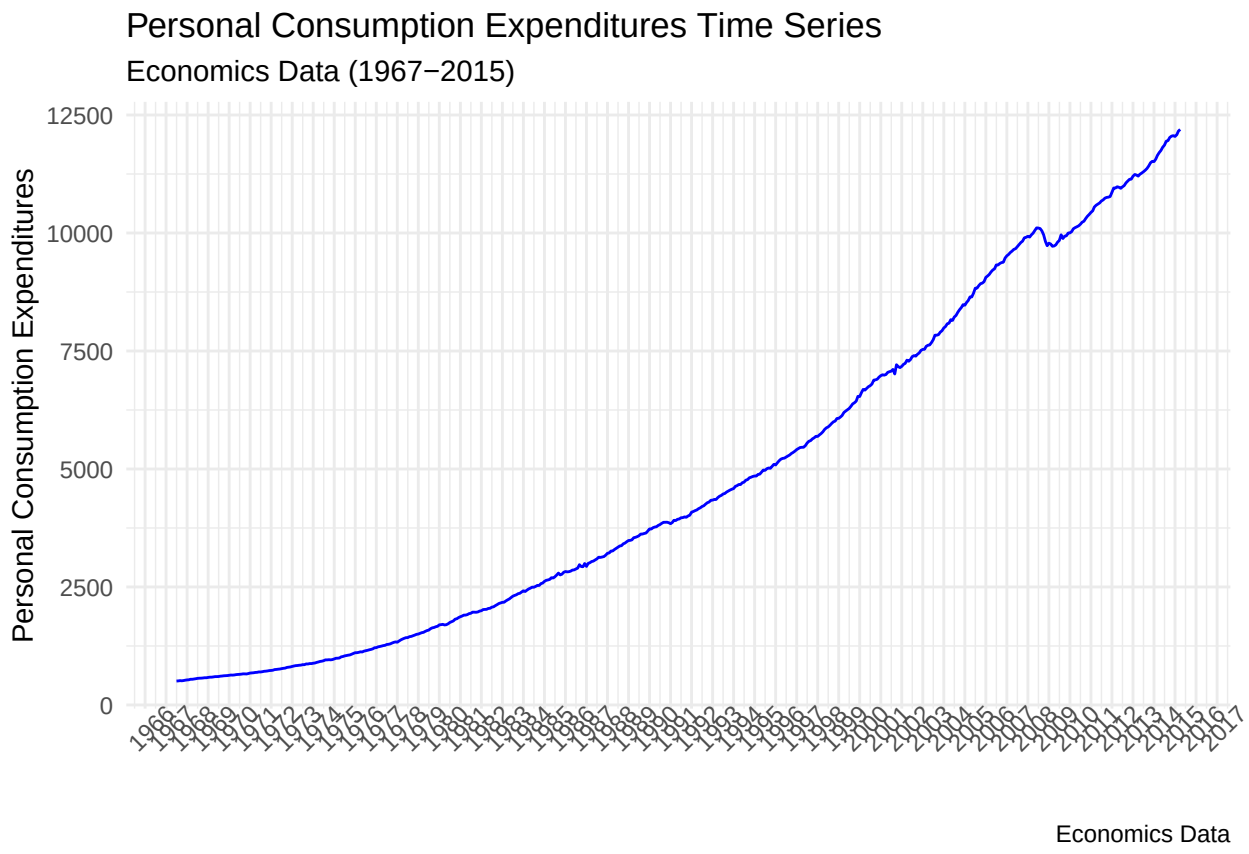
```
summary(economics)
```

```
##       date                 pce              pop            psavert
##  Min.   :1967-07-01   Min.   :  506.7   Min.   :198712   Min.   : 2.200
##  1st Qu.:1979-06-08   1st Qu.: 1578.3   1st Qu.:224896   1st Qu.: 6.400
##  Median :1991-05-16   Median : 3936.8   Median :253060   Median : 8.400
##  Mean   :1991-05-17   Mean   : 4820.1   Mean   :257160   Mean   : 8.567
##  3rd Qu.:2003-04-23   3rd Qu.: 7626.3   3rd Qu.:290291   3rd Qu.:11.100
##  Max.   :2015-04-01   Max.   :12193.8   Max.   :320402   Max.   :17.300
##     uempmed          unemploy
##  Min.   : 4.000   Min.   : 2685
##  1st Qu.: 6.000   1st Qu.: 6284
##  Median : 7.500   Median : 7494
##  Mean   : 8.609   Mean   : 7771
##  3rd Qu.: 9.100   3rd Qu.: 8686
##  Max.   :25.200   Max.   :15352
```

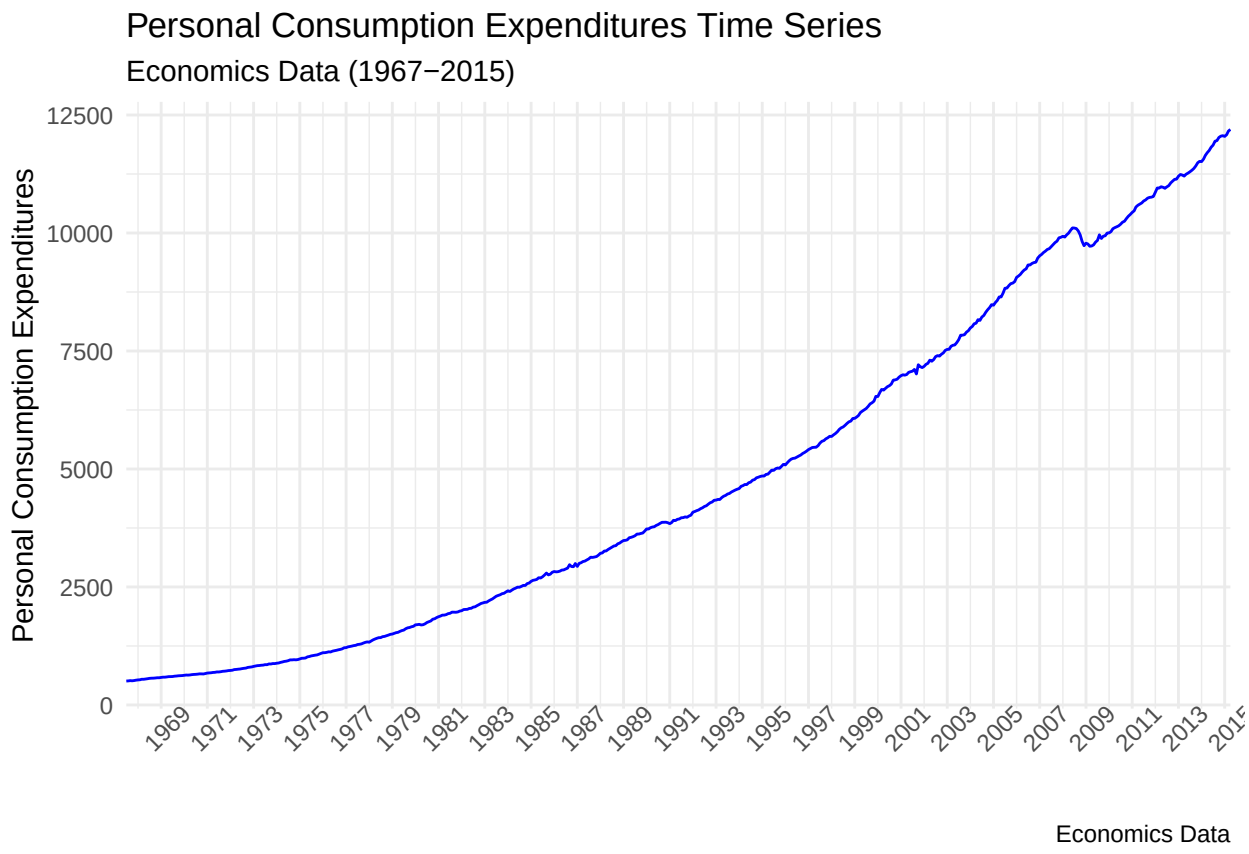
```
p4 <- economics %>%
  ggplot(aes(x=date,y=pce)) +
  geom_line(color="blue") +
  theme_minimal() +
  labs(x = "",
       y = "Personal Consumption Expenditures",
       title = "Personal Consumption Expenditures Time Series",
       caption = "Economics Data",
       subtitle = "Economics Data (1967-2015)")
p4
```



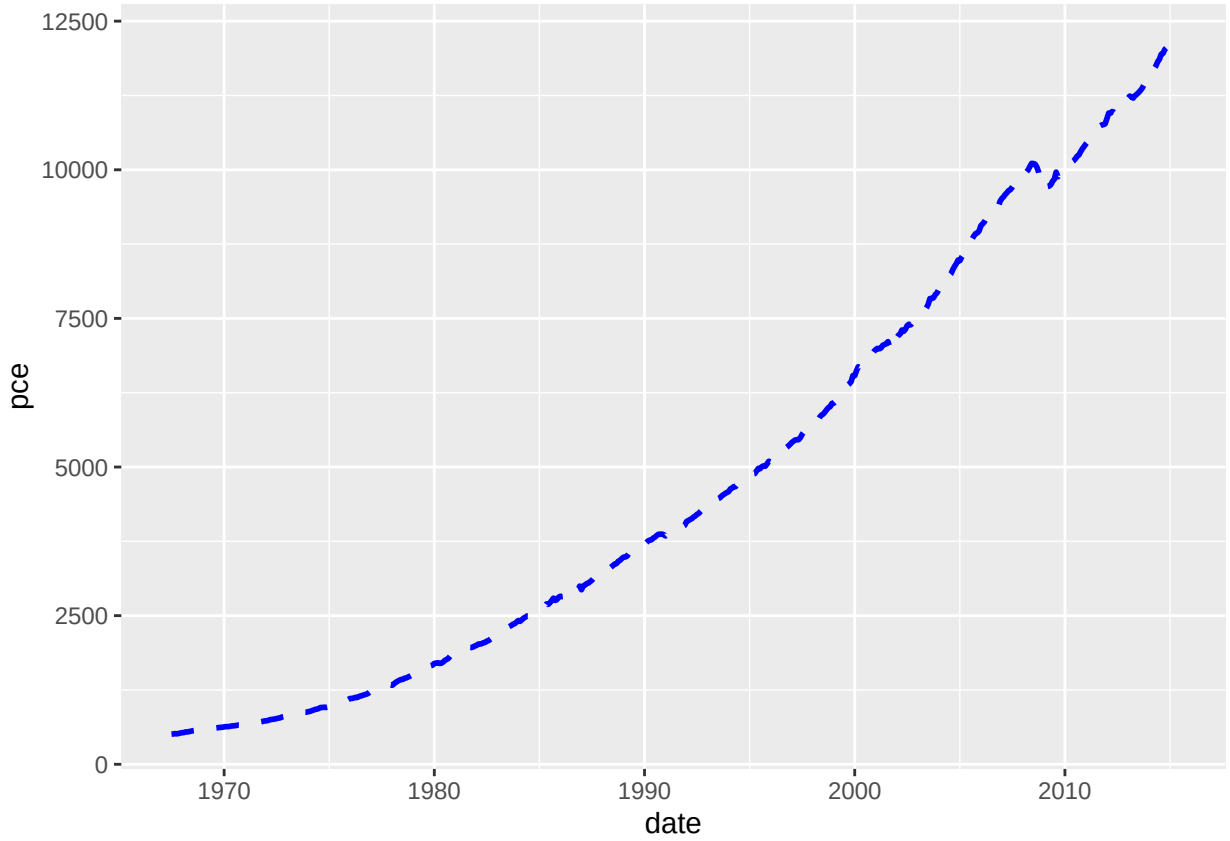

```
# zaman eksenini ayarlama
p4 +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  theme(axis.text.x = element_text(angle = 45), legend.position = "top")
```



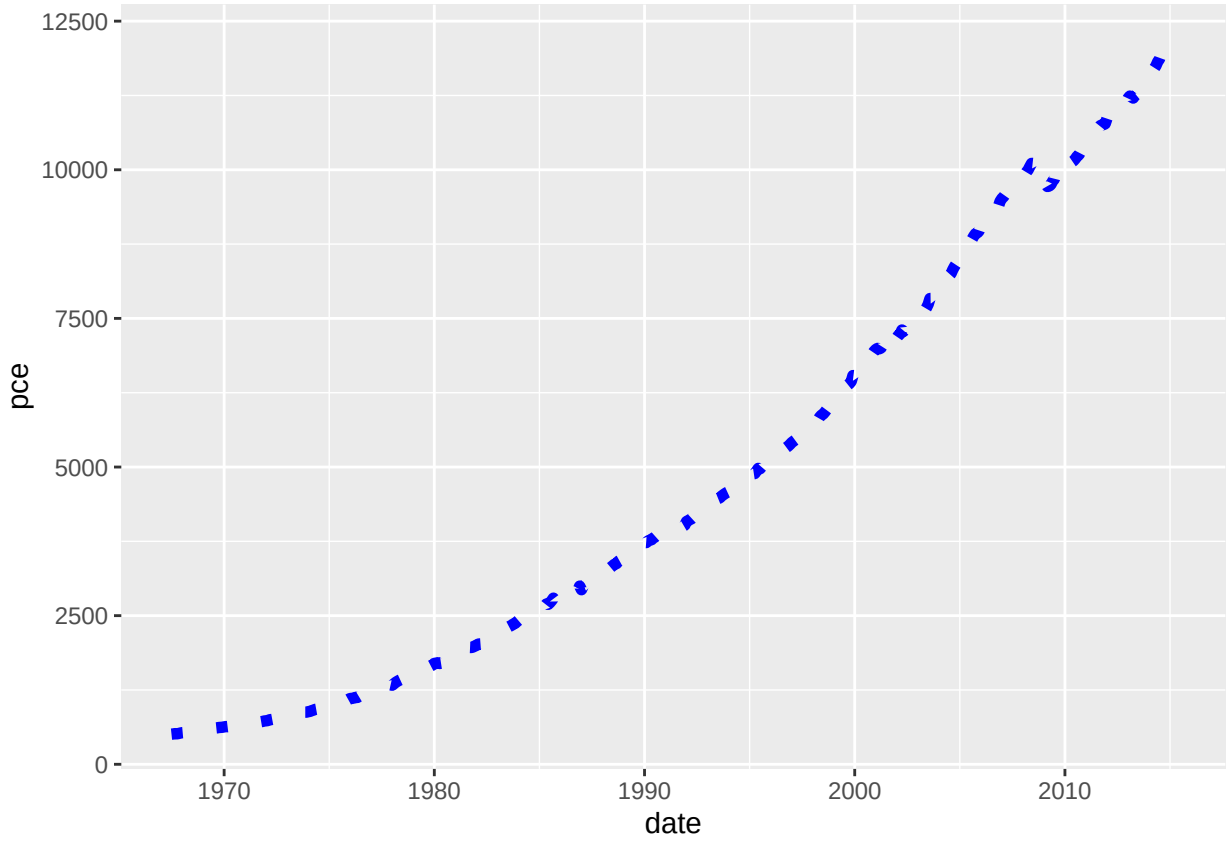

```
p4 +
  scale_x_date(date_breaks = "2 year", date_labels = "%Y",expand = c(0,0)) +
  theme(axis.text.x = element_text(angle = 45), legend.position = "top")
```



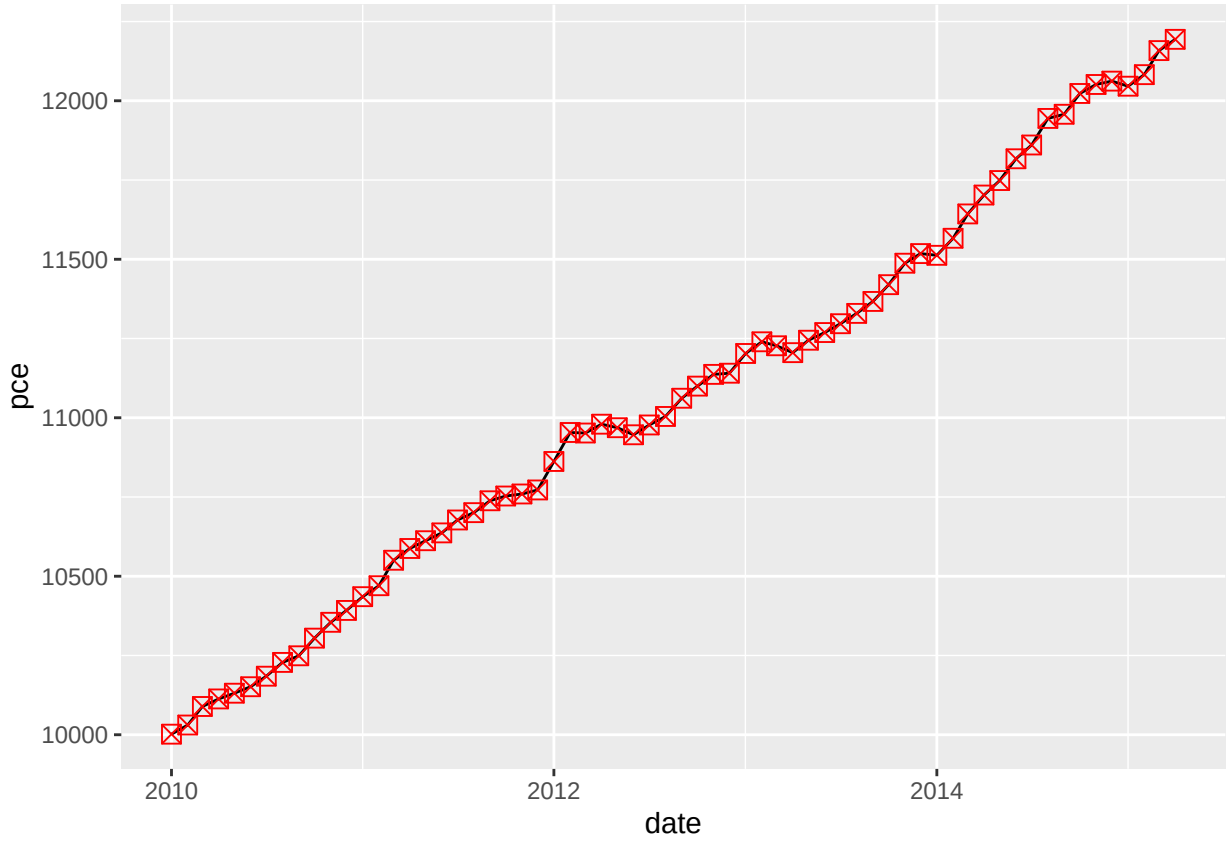

```
# çizgi türü değiştirilebilir
economics %>%
  ggplot(aes(x=date,y=pce)) +
  geom_line(linetype = "dashed", size = 1, colour = "blue")
```



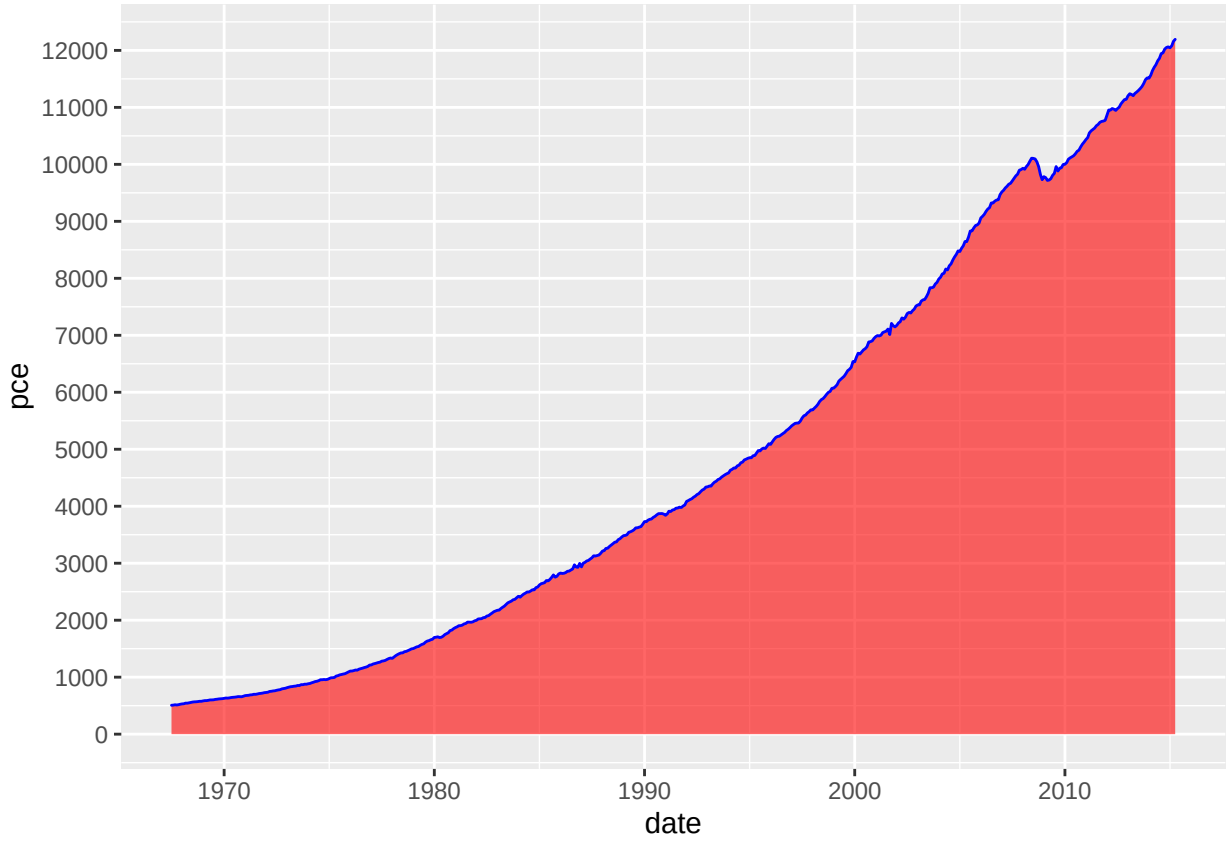

```
economics %>%
  ggplot(aes(x=date,y=pce)) +
  geom_line(linetype = "dotted", size = 2, colour = "blue")
```



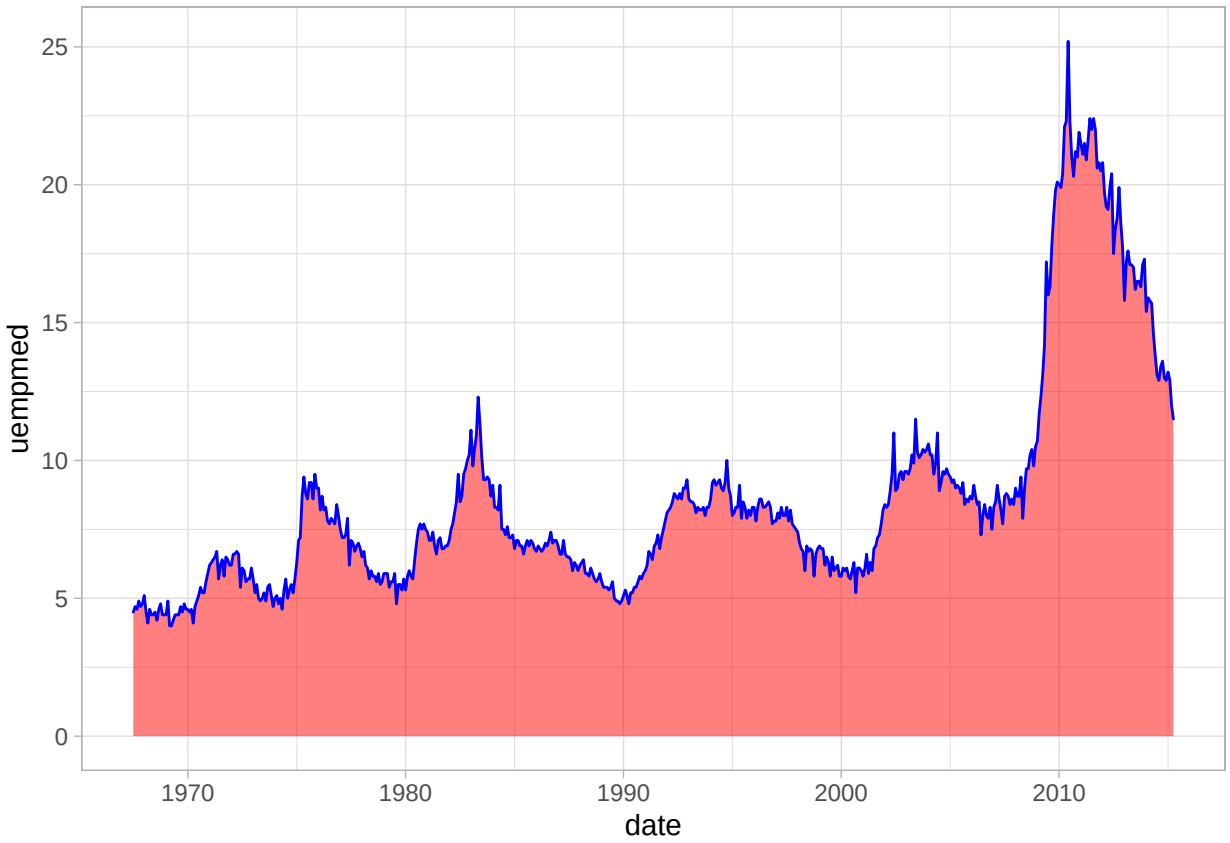

```
# zaman grafiğine noktalar ekleme
economics %>%
  filter(lubridate::year(date) >= 2010) %>%
  ggplot(aes(x=date,y=pce)) +
  geom_line()+
  geom_point(size = 3, shape= 7, colour = "red")
```




```r
# gölgeli zaman grafiği
economics %>%
  ggplot(aes(x=date,y=pce)) +
  geom_area(color="blue",fill="red",alpha=0.6) +
  # y ekseni aralıklarını ayarlama
  scale_y_continuous(breaks = seq(0, max(economics$pce), by = 1000))
```




```
economics %>%
  ggplot(aes(x=date,y=uempmed )) +
  geom_area(color="blue",fill="red",alpha=0.5) +
  theme_light()
```

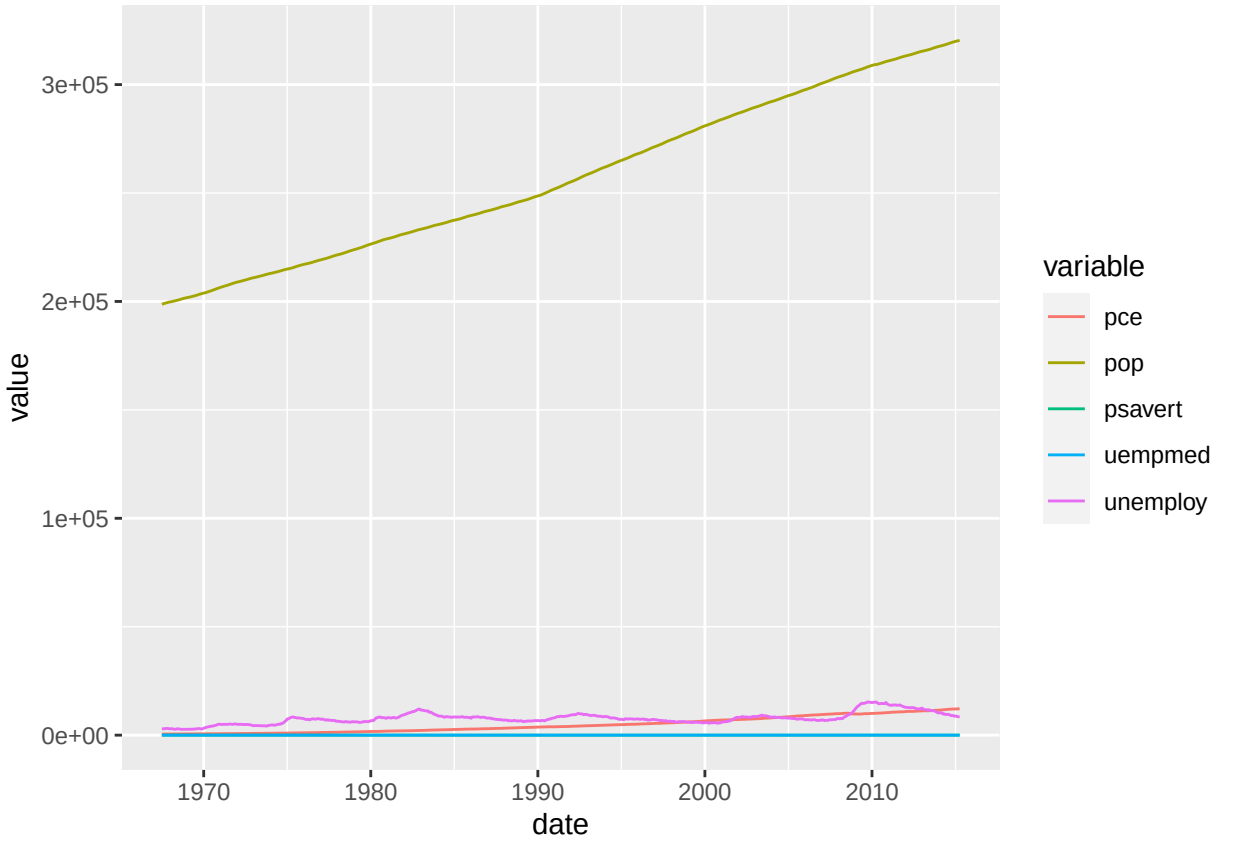



```
# çoklu zaman serisi grafiği
economics_long
```

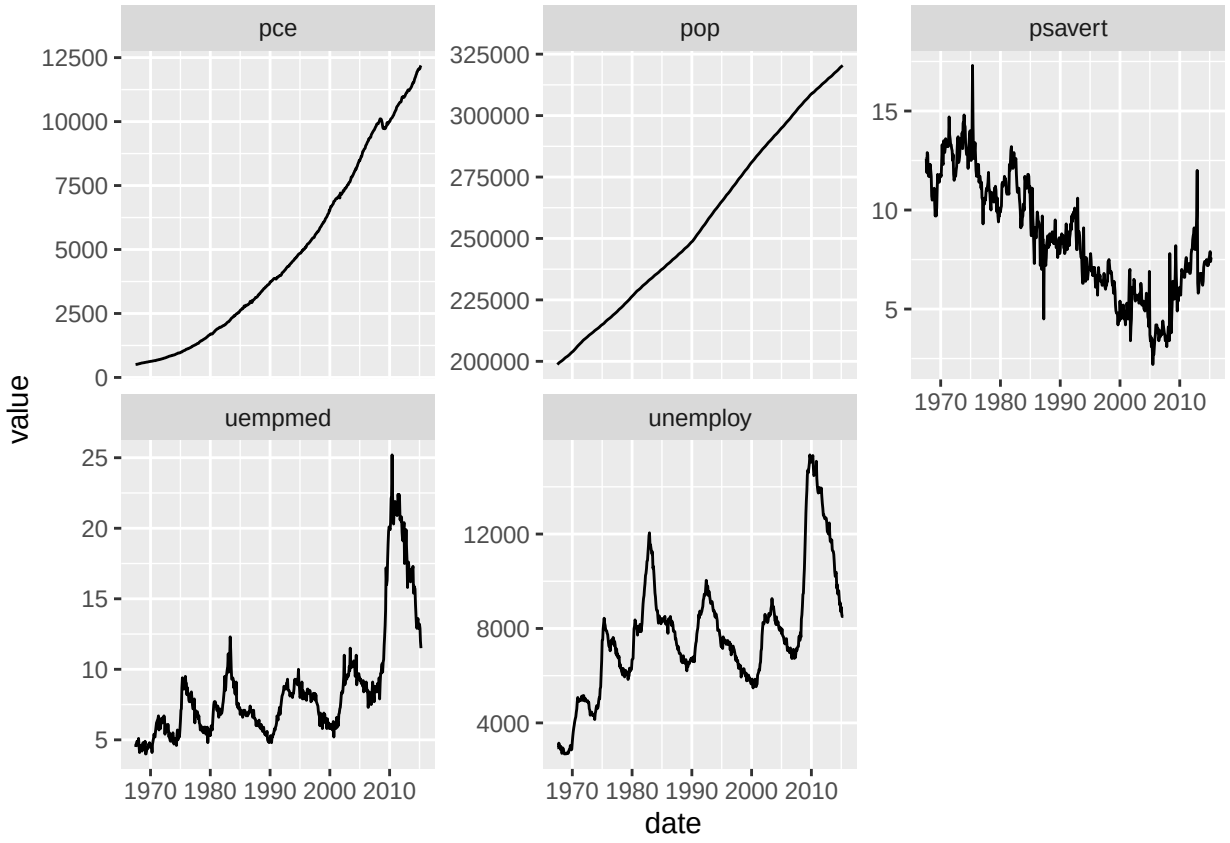
```
## # A tibble: 2,870 x 4
##    date       variable value  value01
##    <date>     <chr>    <dbl>    <dbl>
##  1 1967-07-01 pce       507. 0
##  2 1967-08-01 pce       510. 0.000265
##  3 1967-09-01 pce       516. 0.000762
##  4 1967-10-01 pce       512. 0.000471
##  5 1967-11-01 pce       517. 0.000916
##  6 1967-12-01 pce       525. 0.00157
##  7 1968-01-01 pce       531. 0.00207
##  8 1968-02-01 pce       534. 0.00230
##  9 1968-03-01 pce       544. 0.00322
## 10 1968-04-01 pce       544  0.00319
## # ... with 2,860 more rows
```

```
# serilerin ölçekleri farklı
economics_long %>%
```

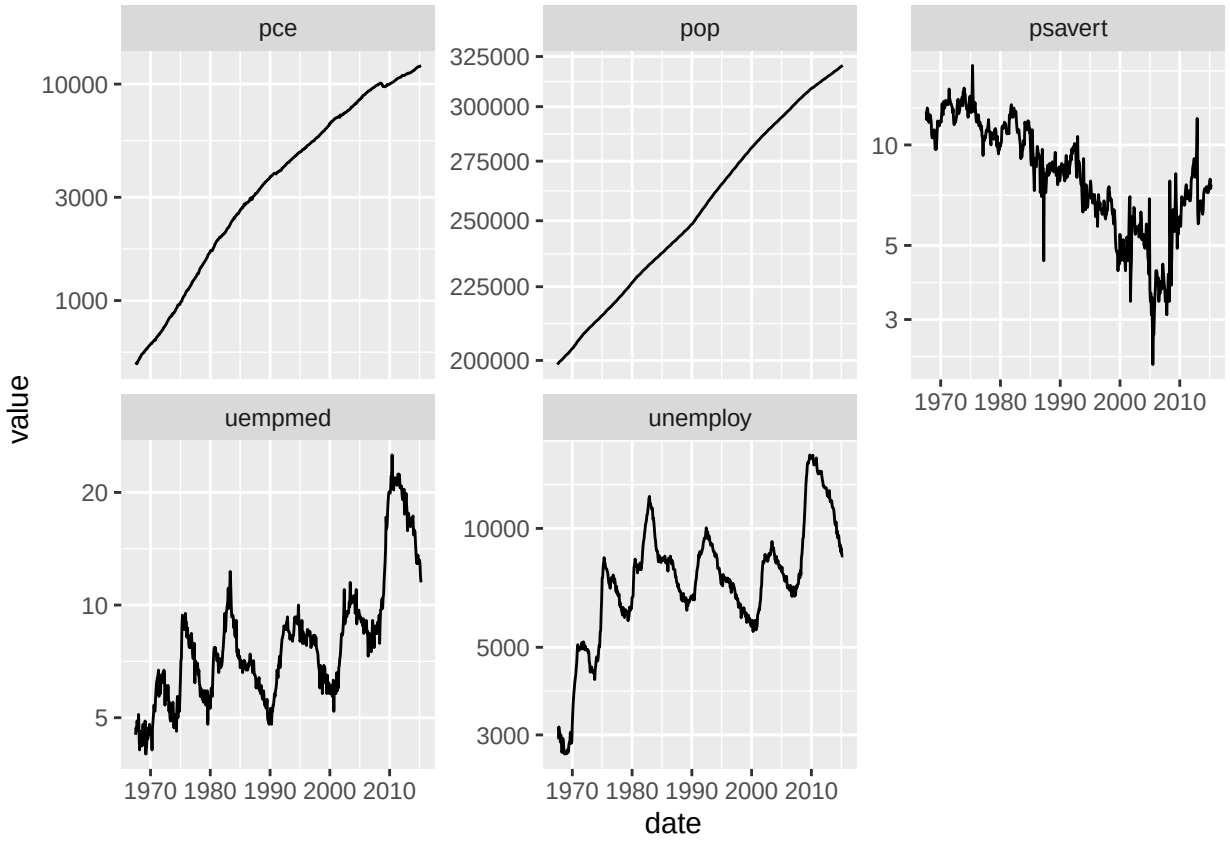
```
  ggplot(aes(x=date,y=value,color=variable))+
  geom_line()
```




```
economics_long %>%
  ggplot(aes(x=date,y=value))+
  geom_line() +
  facet_wrap(~variable,scales = "free_y")
```




```
economics_long %>%
  ggplot(aes(x=date,y=value))+
  geom_line() +
  facet_wrap(~variable,scales = "free_y")+
  scale_y_log10() # y eksenlerinin logatirması alınır
```



2.3 Sütun grafikleri

```
diamonds
```

```
## # A tibble: 53,940 x 10
##    carat cut       color clarity depth table price     x     y     z
##    <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
##  1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
##  2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
##  3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
##  4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
##  5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
##  6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
##  7  0.24 Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
##  8  0.26 Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
##  9  0.22 Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
## 10  0.23 Very Good H     VS1      59.4    61   338  4     4.05  2.39
## # ... with 53,930 more rows
```

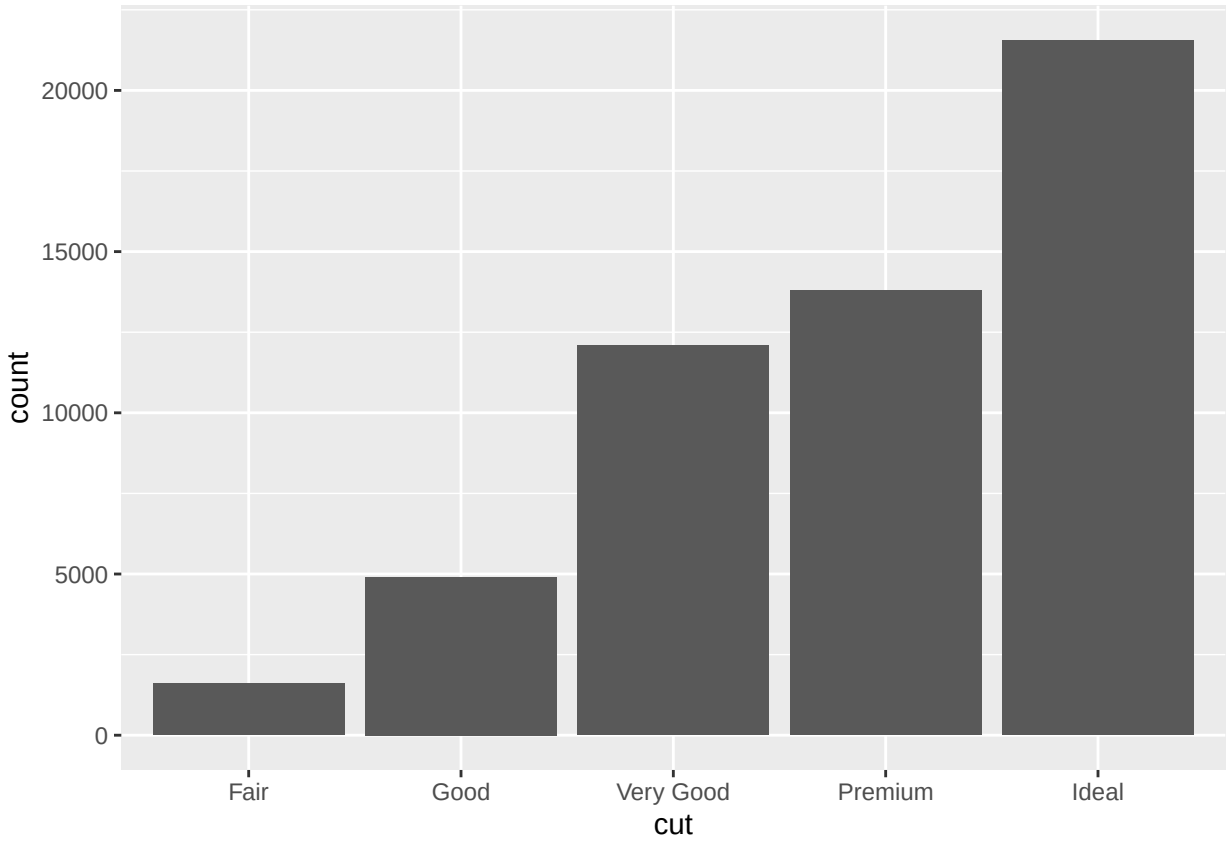
```
glimpse(diamonds)
```

```
## Rows: 53,940
## Columns: 10
## $ carat   <dbl> 0.23, 0.21, 0.23, 0.29, 0.31, 0.24, 0.24, 0.26, 0.22, 0.23, 0.~
## $ cut     <ord> Ideal, Premium, Good, Premium, Good, Very Good, Very Good, Ver~
## $ color   <ord> E, E, E, I, J, J, I, H, E, H, J, J, F, J, E, E, I, J, J, J, I,~
## $ clarity <ord> SI2, SI1, VS1, VS2, SI2, VVS2, VVS1, SI1, VS2, VS1, SI1, VS1, ~
## $ depth   <dbl> 61.5, 59.8, 56.9, 62.4, 63.3, 62.8, 62.3, 61.9, 65.1, 59.4, 64~
## $ table   <dbl> 55, 61, 65, 58, 58, 57, 57, 55, 61, 61, 55, 56, 61, 54, 62, 58~
## $ price   <int> 326, 326, 327, 334, 335, 336, 336, 337, 337, 338, 339, 340, 34~
## $ x       <dbl> 3.95, 3.89, 4.05, 4.20, 4.34, 3.94, 3.95, 4.07, 3.87, 4.00, 4.~
## $ y       <dbl> 3.98, 3.84, 4.07, 4.23, 4.35, 3.96, 3.98, 4.11, 3.78, 4.05, 4.~
## $ z       <dbl> 2.43, 2.31, 2.31, 2.63, 2.75, 2.48, 2.47, 2.53, 2.49, 2.39, 2.~
```

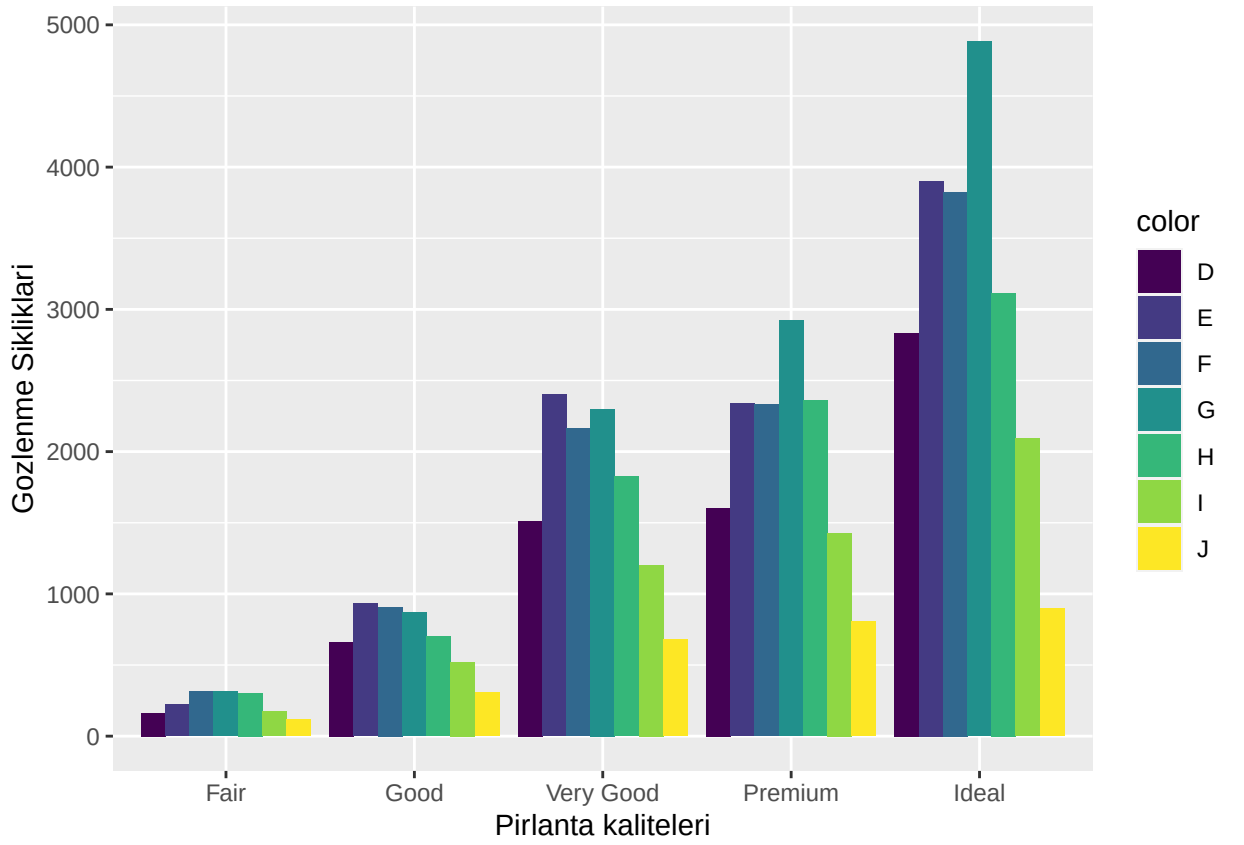
```
summary(diamonds)
```

```
##      carat               cut        color        clarity          depth      
##  Min.   :0.2000   Fair      : 1610   D: 6775   SI1    :13065   Min.   :43.00  
##  1st Qu.:0.4000   Good      : 4906   E: 9797   VS2    :12258   1st Qu.:61.00  
##  Median :0.7000   Very Good:12082   F: 9542   SI2    : 9194   Median :61.80  
##  Mean   :0.7979   Premium   :13791   G:11292   VS1    : 8171   Mean   :61.75  
##  3rd Qu.:1.0400   Ideal     :21551   H: 8304   VVS2   : 5066   3rd Qu.:62.50  
##  Max.   :5.0100                     I: 5422   VVS1   : 3655   Max.   :79.00  
##                                     J: 2808   (Other): 2531                  
##      table           price             x                y         
##  Min.   :43.00   Min.   :  326   Min.   : 0.000   Min.   : 0.000  
##  1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710   1st Qu.: 4.720  
##  Median :57.00   Median : 2401   Median : 5.700   Median : 5.710  
##  Mean   :57.46   Mean   : 3933   Mean   : 5.731   Mean   : 5.735  
##  3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540   3rd Qu.: 6.540  
##  Max.   :95.00   Max.   :18823   Max.   :10.740   Max.   :58.900  
##                                                                   
##        z         
##  Min.   : 0.000  
##  1st Qu.: 2.910  
##  Median : 3.530  
##  Mean   : 3.539  
##  3rd Qu.: 4.040  
##  Max.   :31.800  
##
```

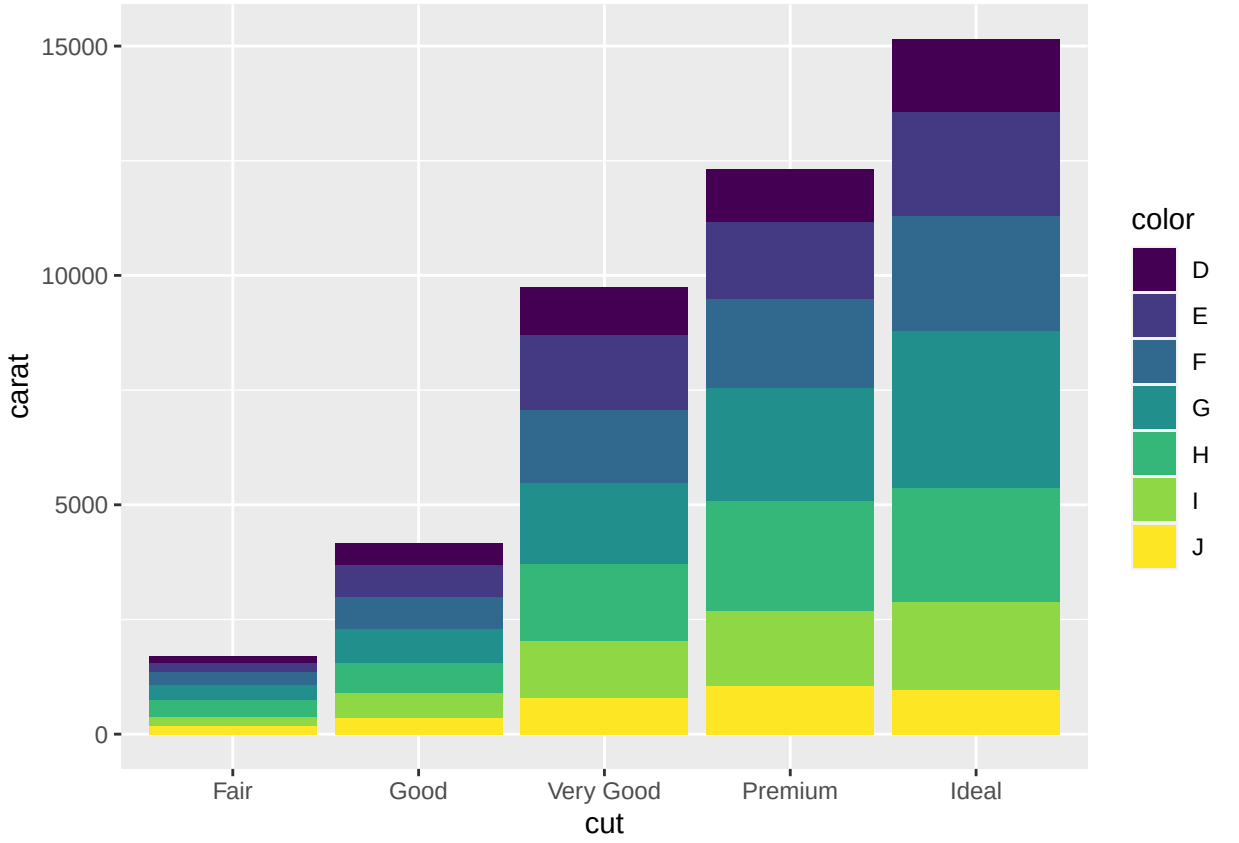
```
# sıklık durumunu görselleştirme
ggplot(diamonds, aes(cut)) +
  geom_bar()
```



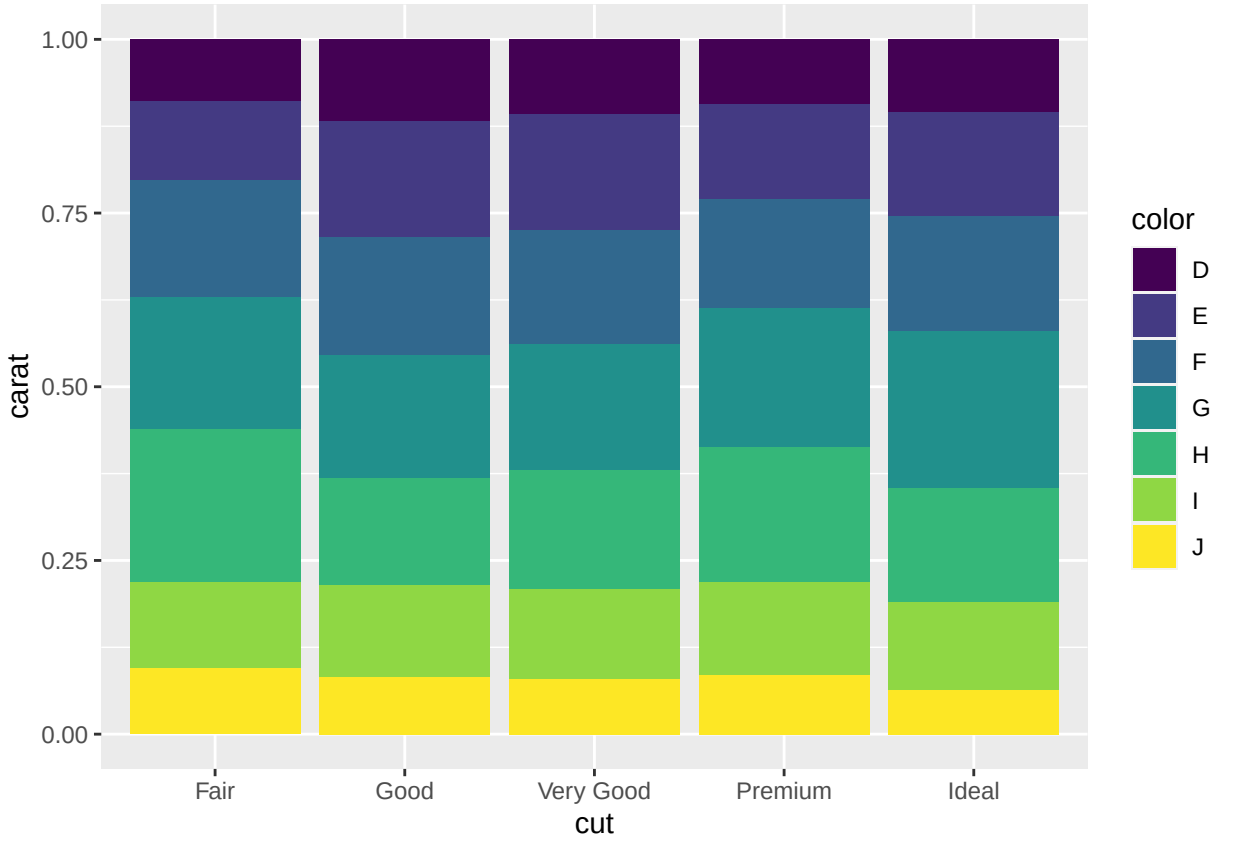

```
ggplot(diamonds, aes(cut, fill = color)) +
  geom_bar(position = position_dodge()) +
  xlab("Pirlanta kaliteleri") +
  ylab("Gozlenme Sikliklari")
```



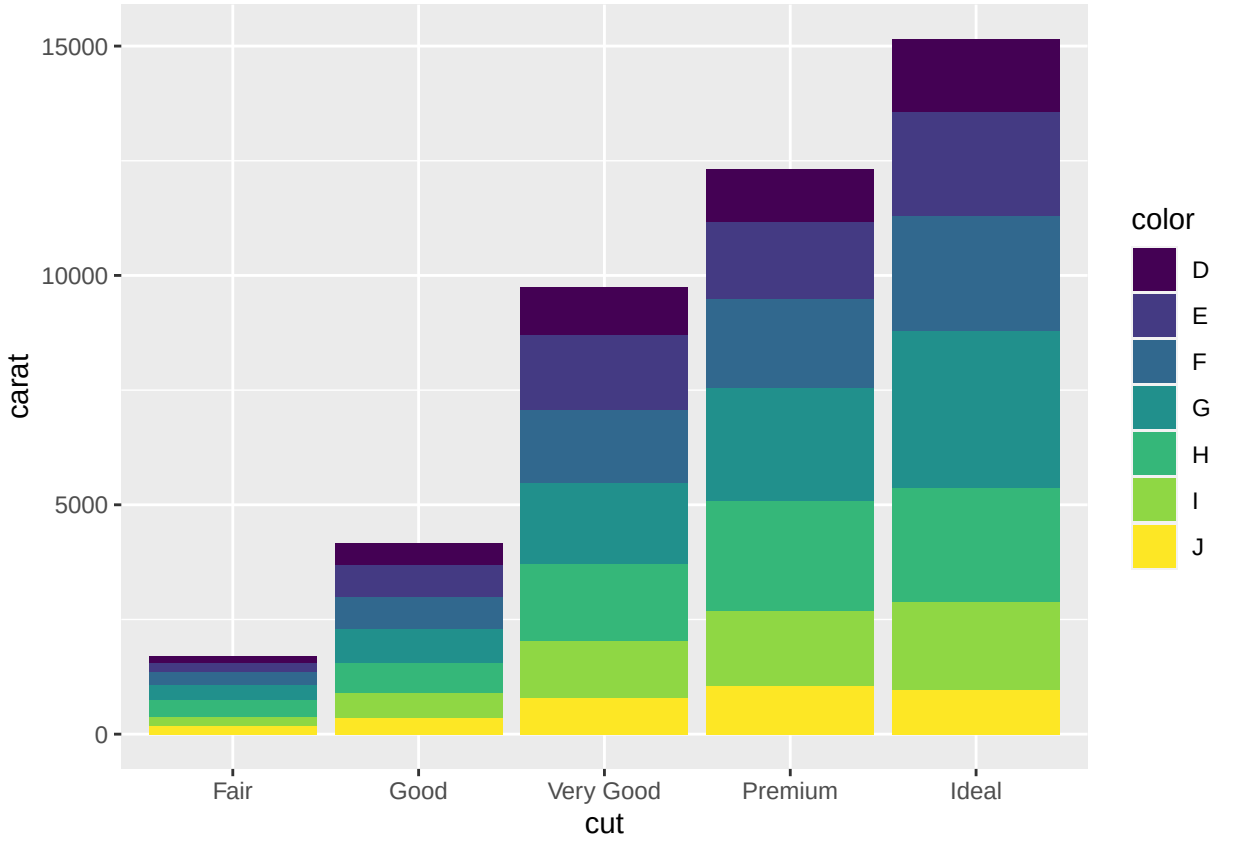

```
ggplot(diamonds, aes(x=cut, y=carat,fill = color)) +
  geom_bar(stat = "identity")
```



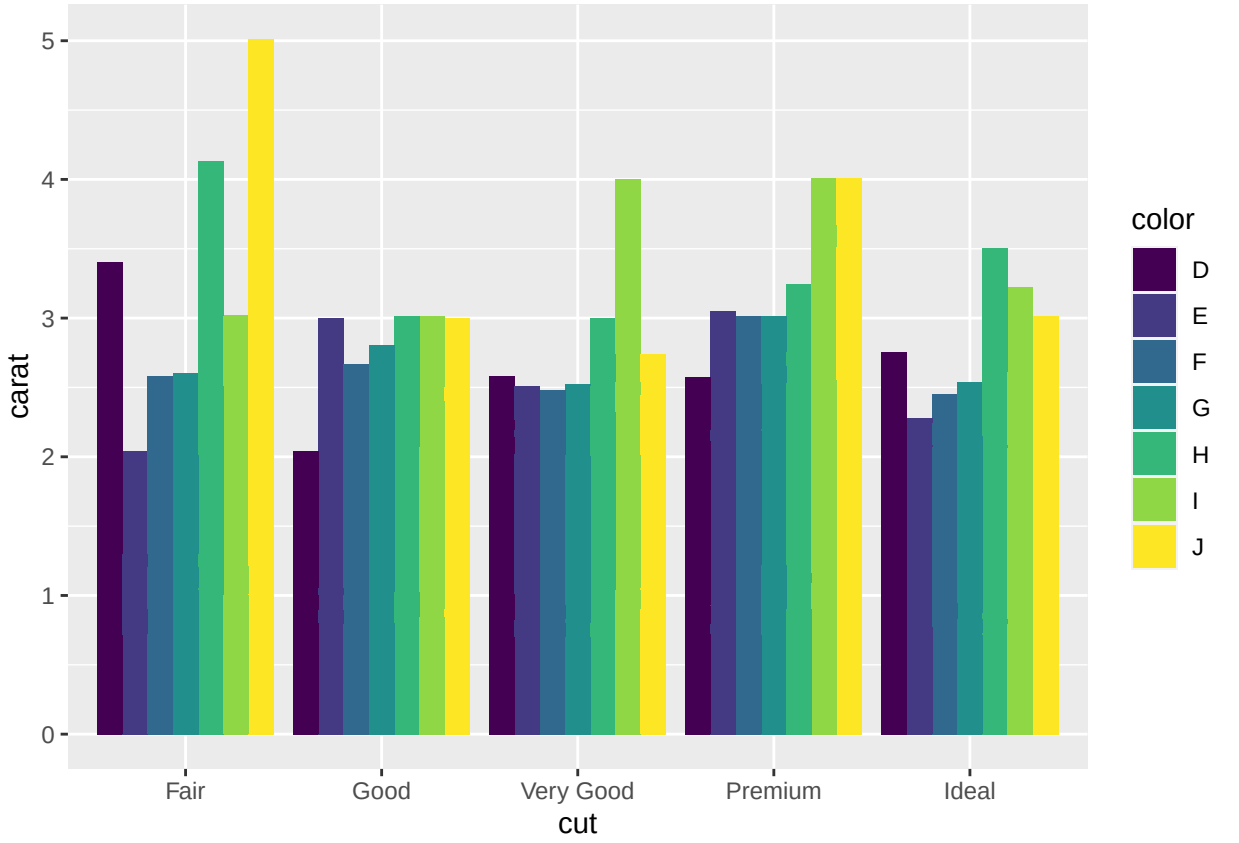

```
ggplot(diamonds, aes(x=cut, y=carat,fill = color)) +
  # fill ile oransal olarak gösterim yapılır
  geom_bar(stat = "identity",position = "fill")
```



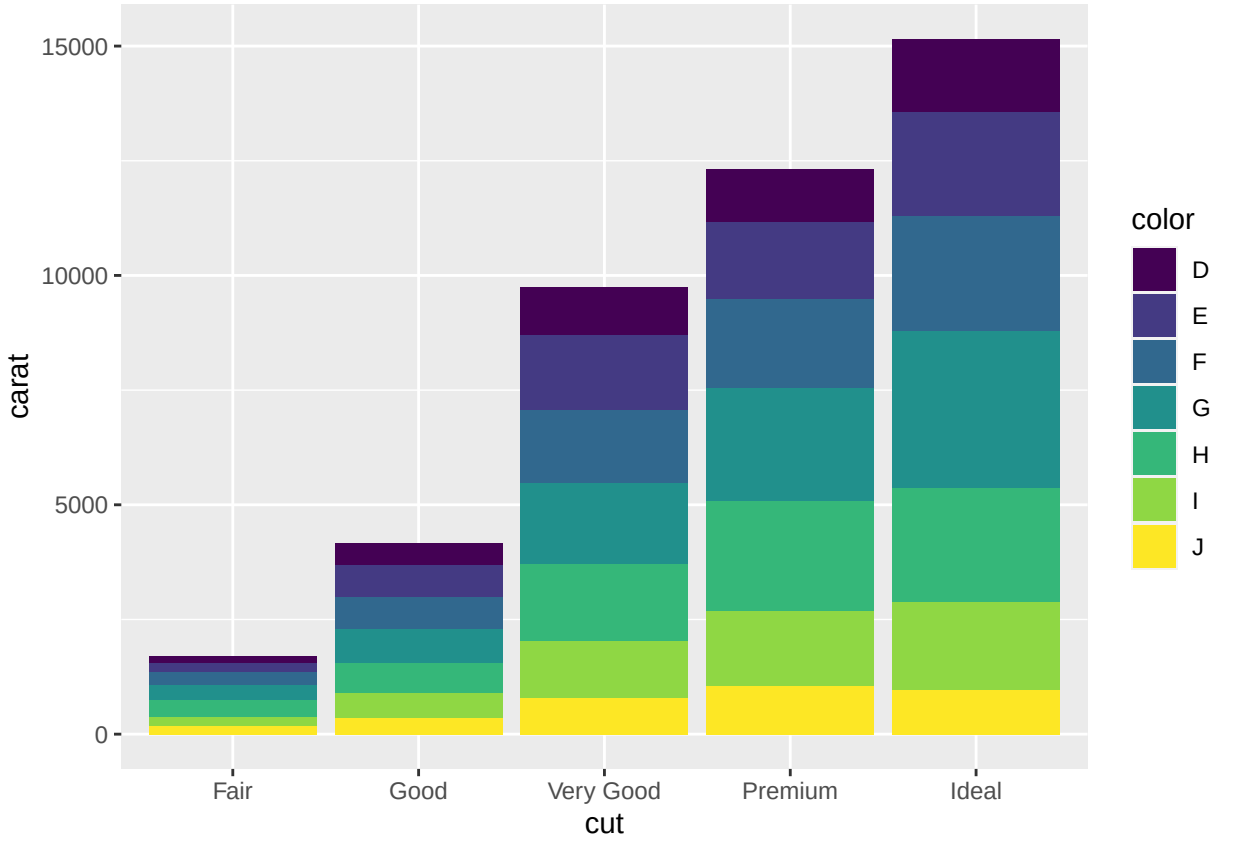

```
ggplot(diamonds, aes(x=cut,y=carat, fill = color)) +
  geom_col() # y ekseni toplanarak yığılmış
```



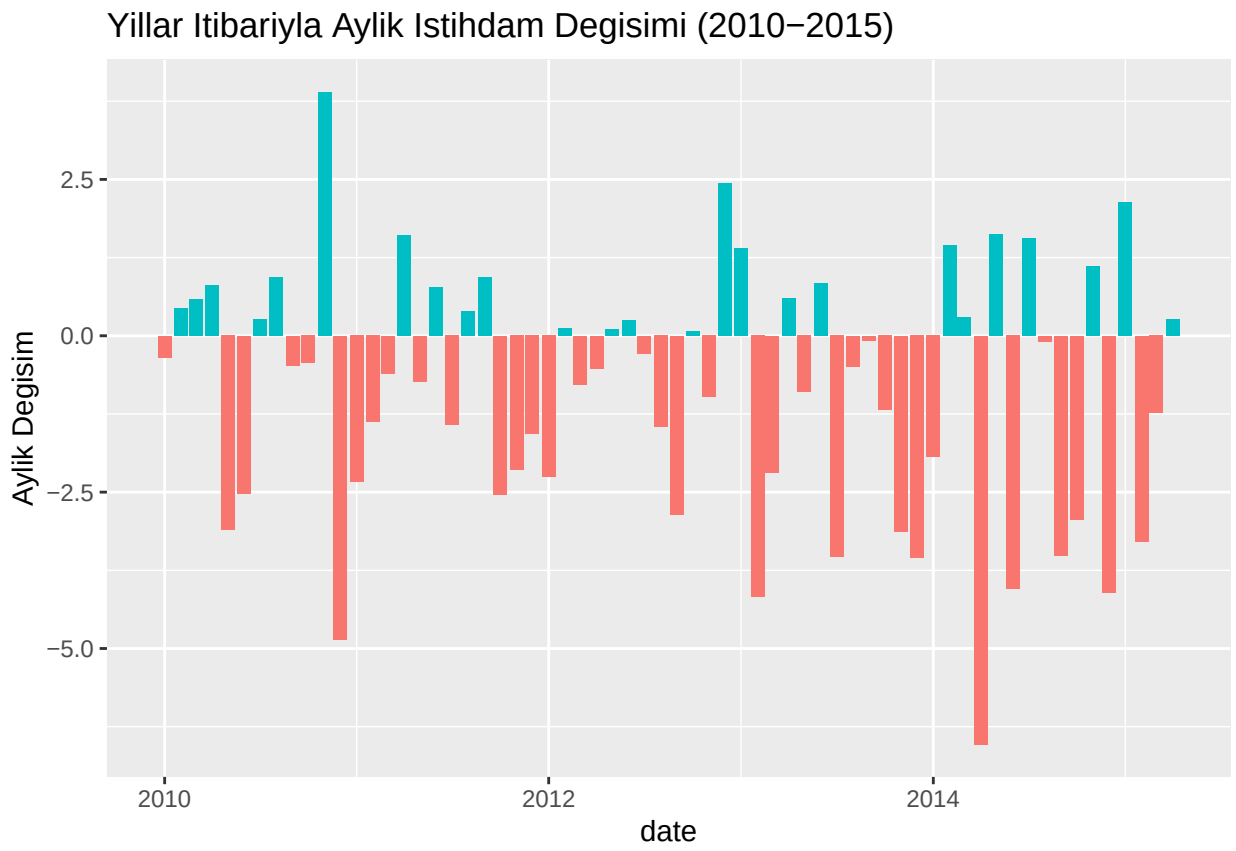

```
ggplot(diamonds, aes(x=cut,y=carat,, fill = color)) +
  geom_col(position = "dodge") # y ekseni değerleri
```




```
ggplot(diamonds, aes(x=cut,y=carat, fill = color)) +
  geom_col(position = "stack")
```

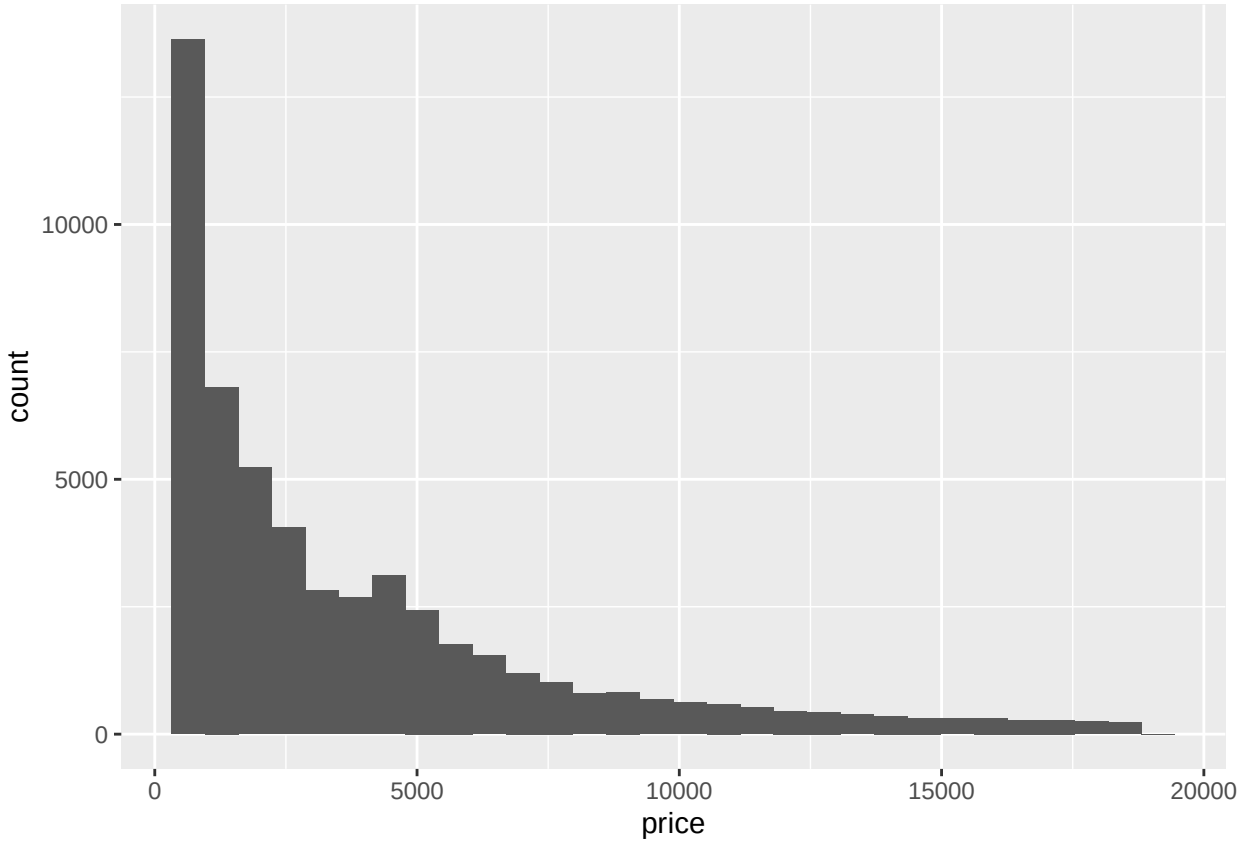



```
# negatif ve pozitif değerli verileri görselleştirme
economics %>%
  mutate(uemploy_mom=unemploy/lag(unemploy ) * 100 - 100,
         growth=ifelse(uemploy_mom>0,"pozitif","negatif")) %>%
  na.omit() %>%
  filter(lubridate::year(date)>=2010) %>%
  ggplot(aes(x=date,y=uemploy_mom,fill=growth))+
  geom_col() +
  theme(legend.position = "none") +
  labs(y="Aylık Değişim",
       title="Yıllar İtibarıyla Aylık İstihdam Değişimi (2010-2015)")
```

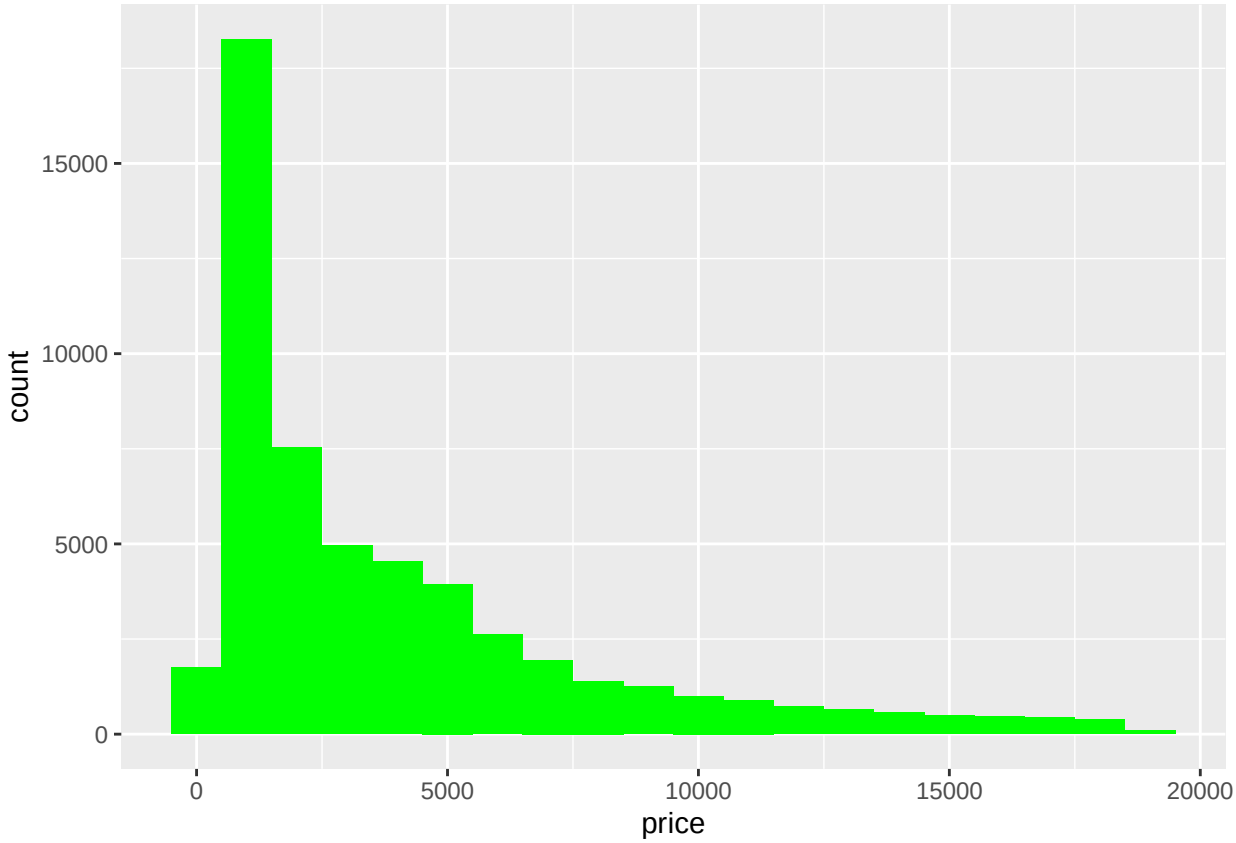


2.4 Dağılım Grafikleri

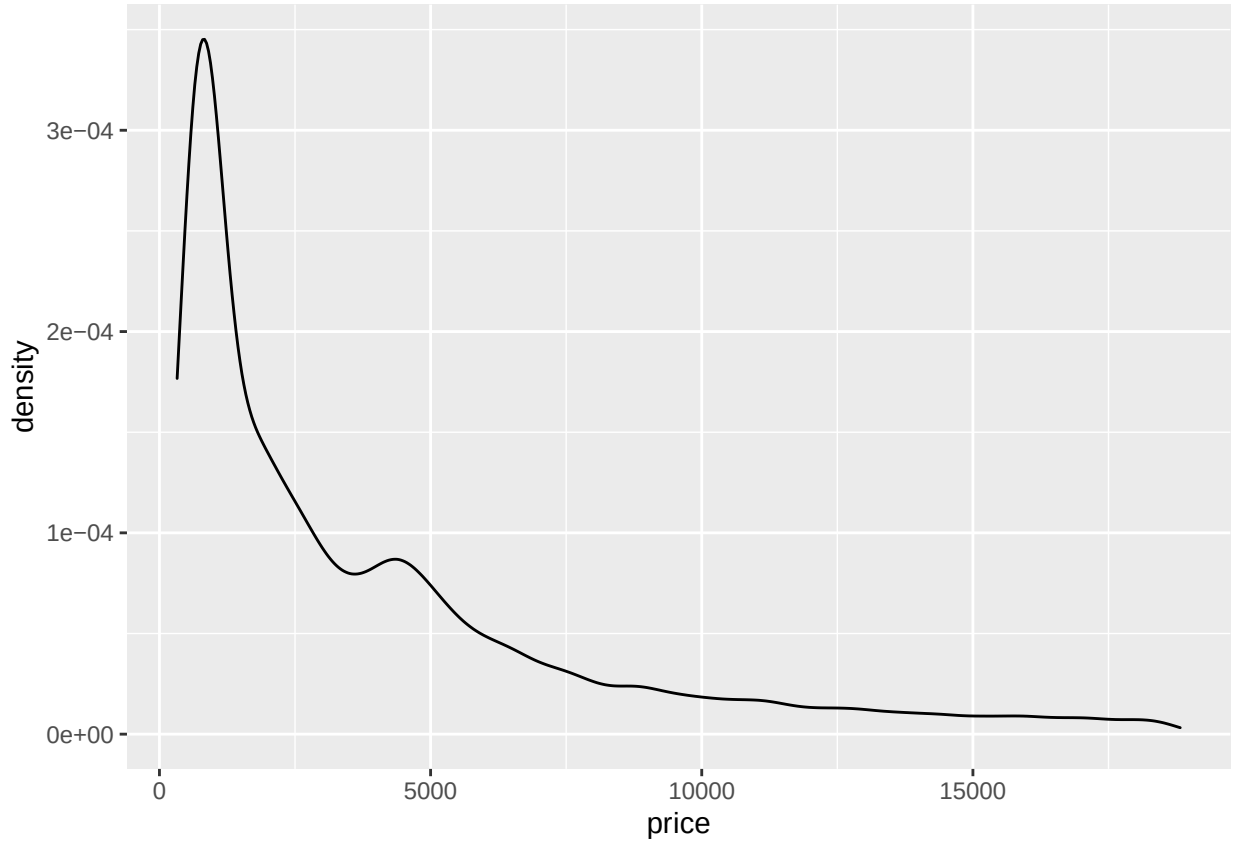
```
ggplot(diamonds, aes(price)) +
  geom_histogram()
```



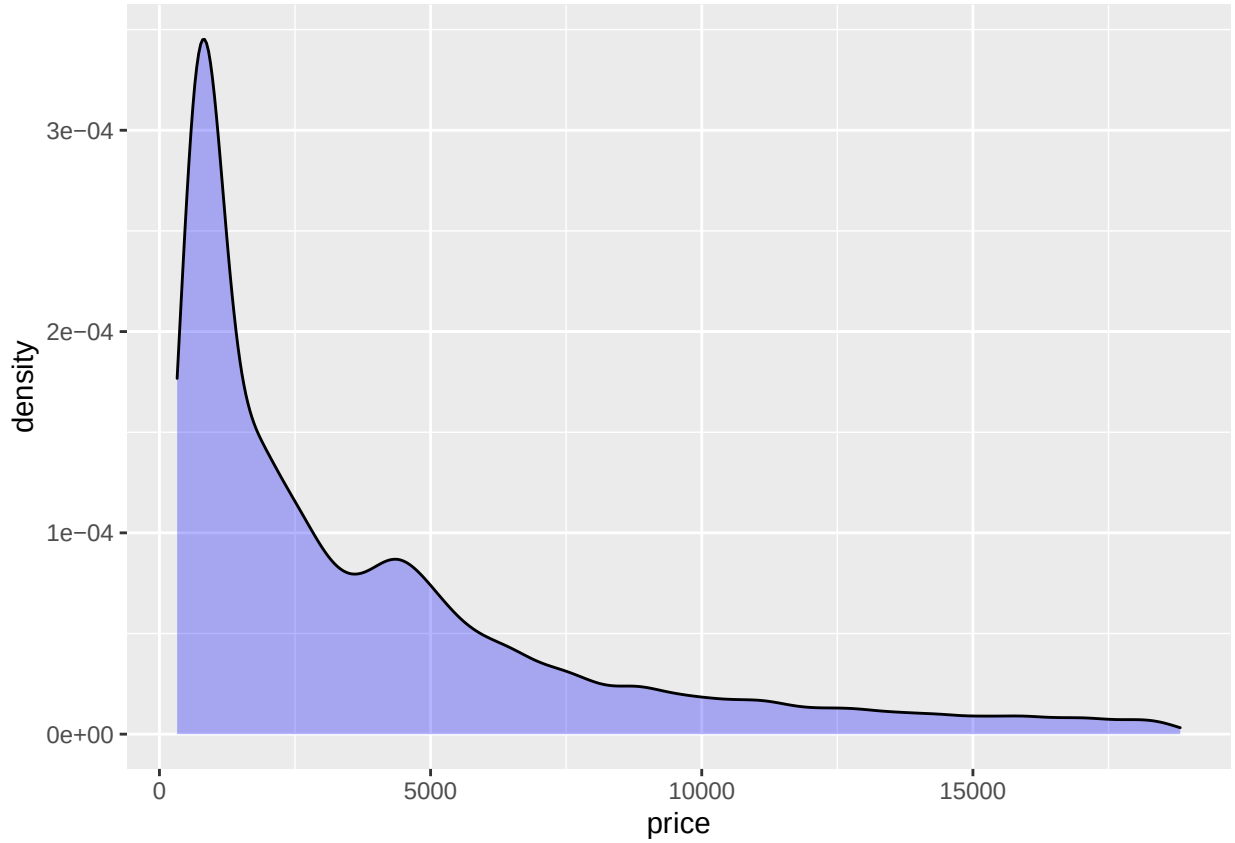

```
ggplot(diamonds, aes(price)) +
  geom_histogram(binwidth = 1000,fill = "green")
```



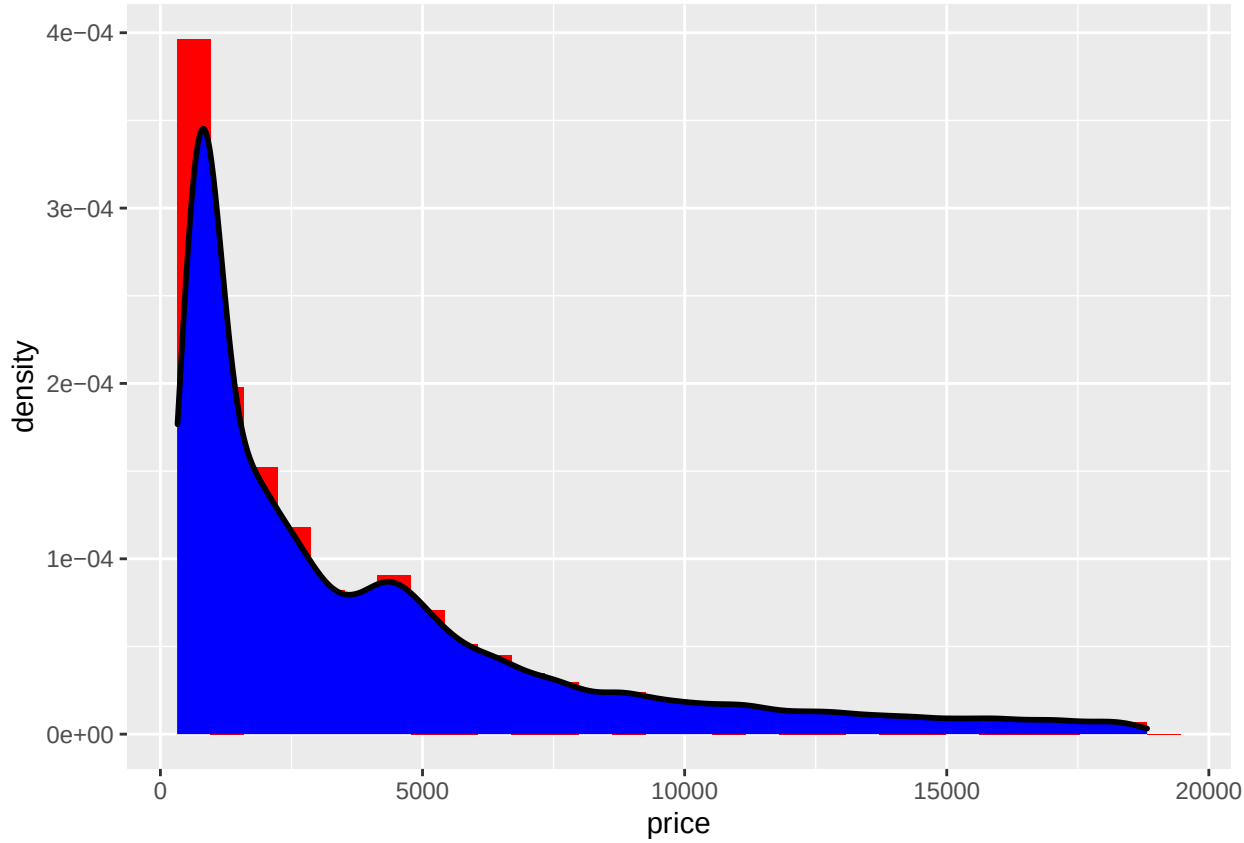

```
ggplot(diamonds, aes(price)) +
  geom_density()
```



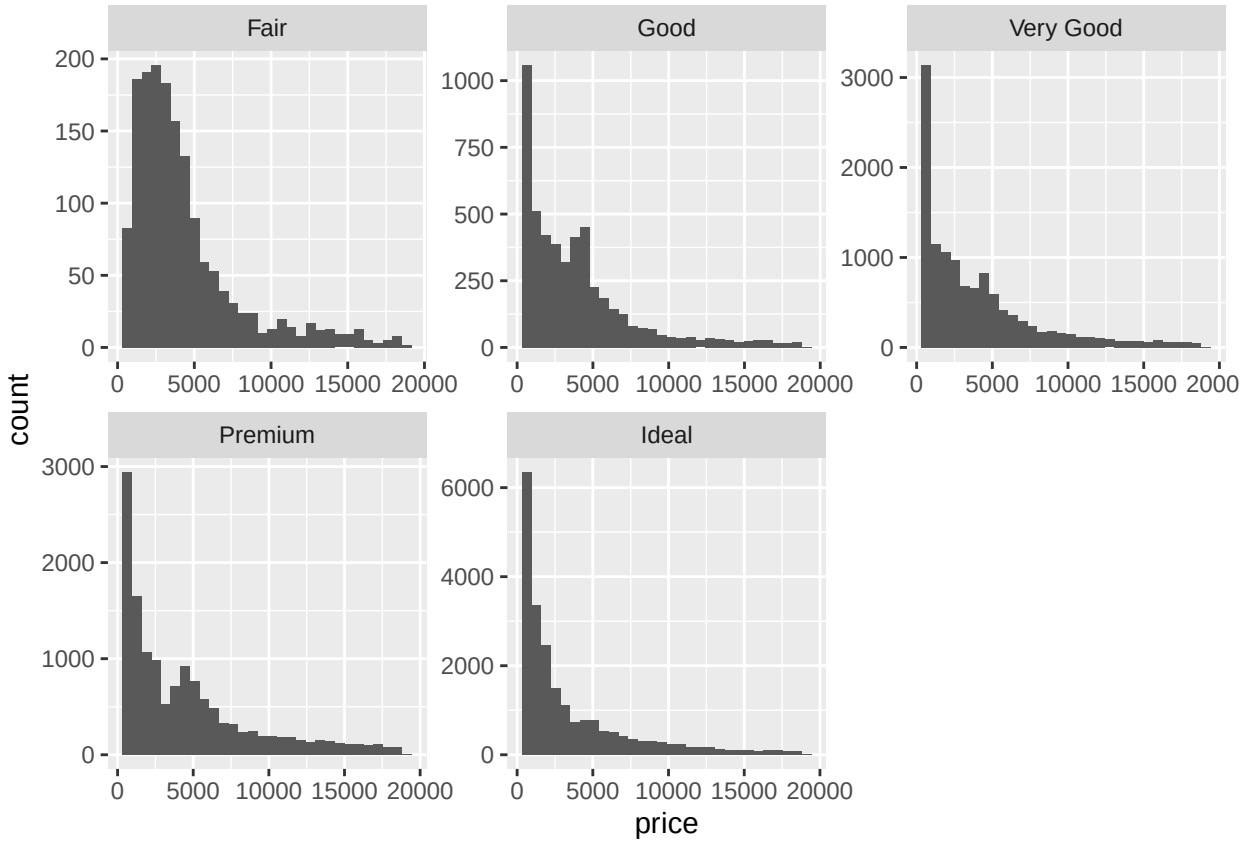

```
ggplot(diamonds, aes(price)) +
  geom_density(alpha = .3, fill = "blue")
```



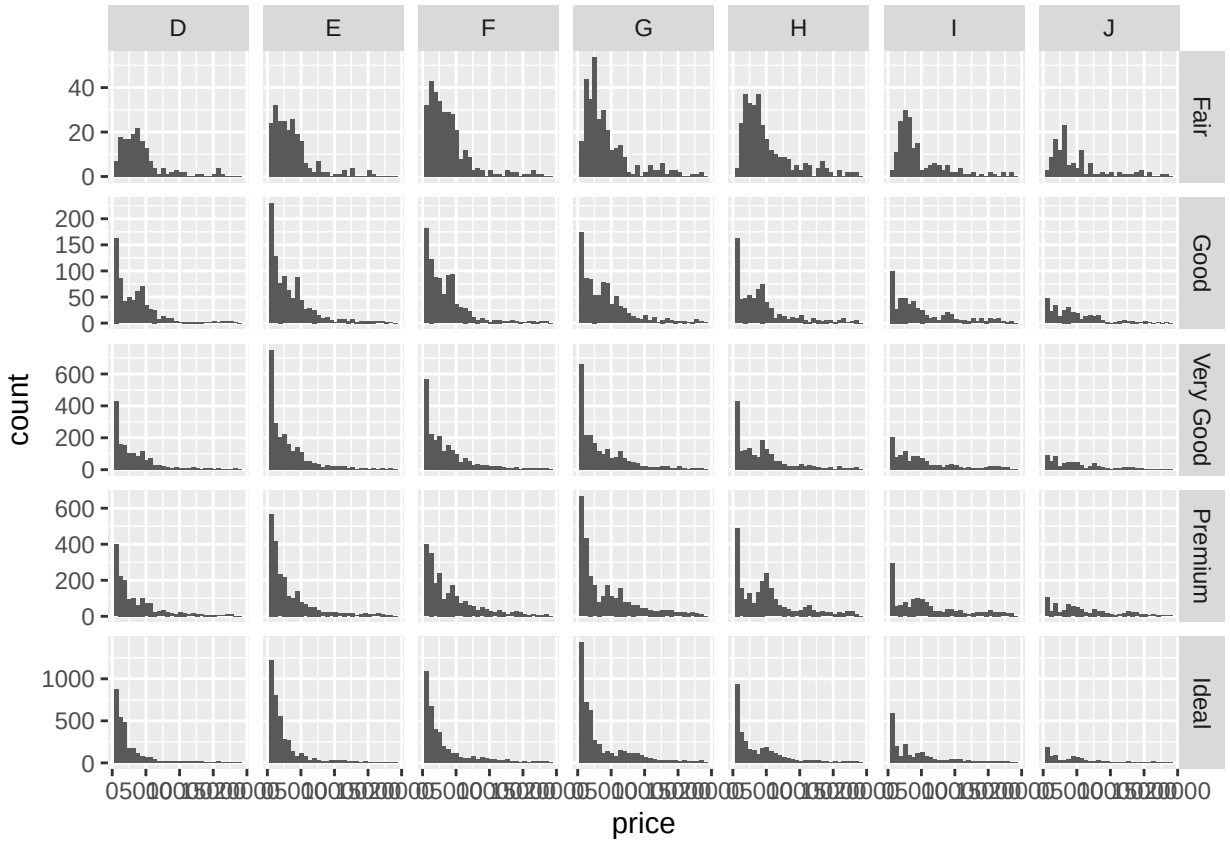

```
ggplot(diamonds, aes(price)) +
  geom_histogram(aes(y = ..density..),fill = "red") +
  geom_density(size=1,fill = "blue")
```



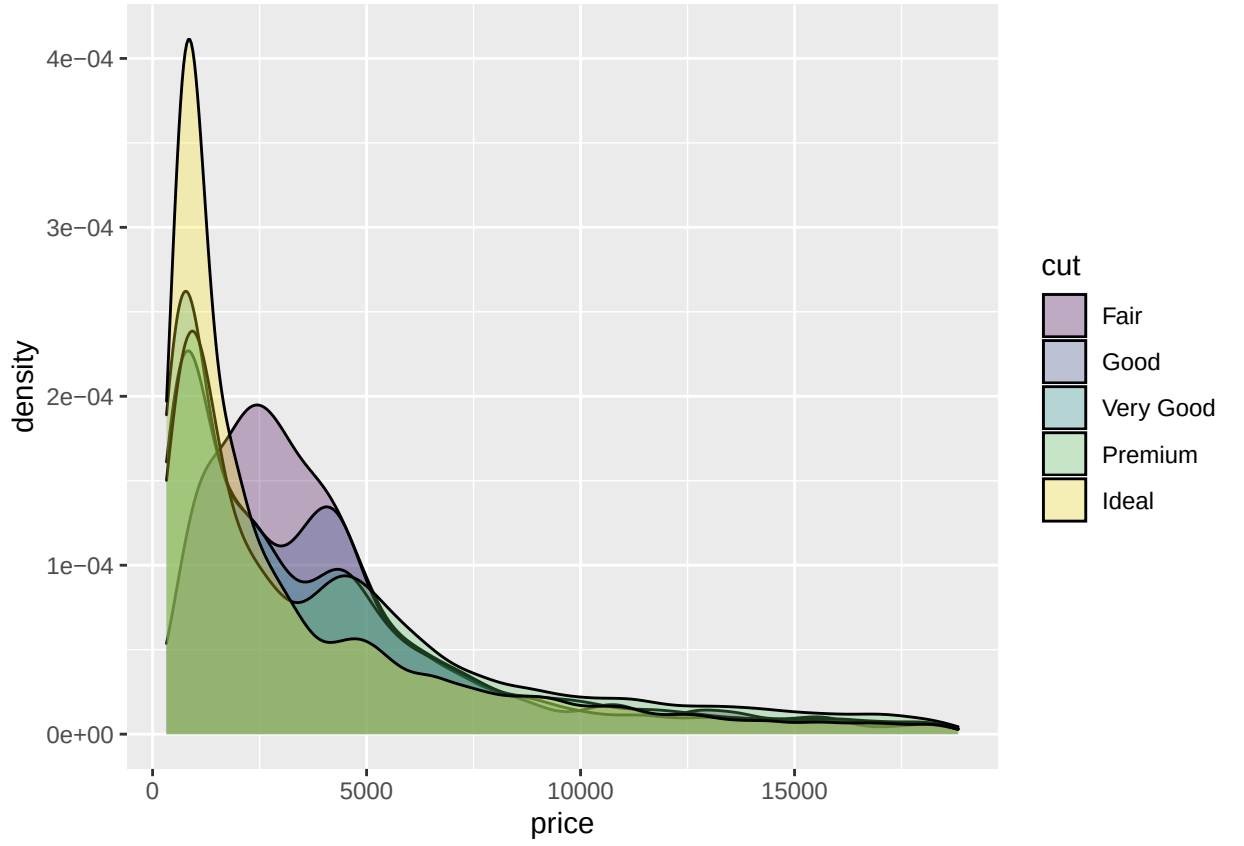

```
ggplot(diamonds, aes(price)) +
  geom_histogram() +
  facet_wrap( ~ cut ,scales = "free" )
```



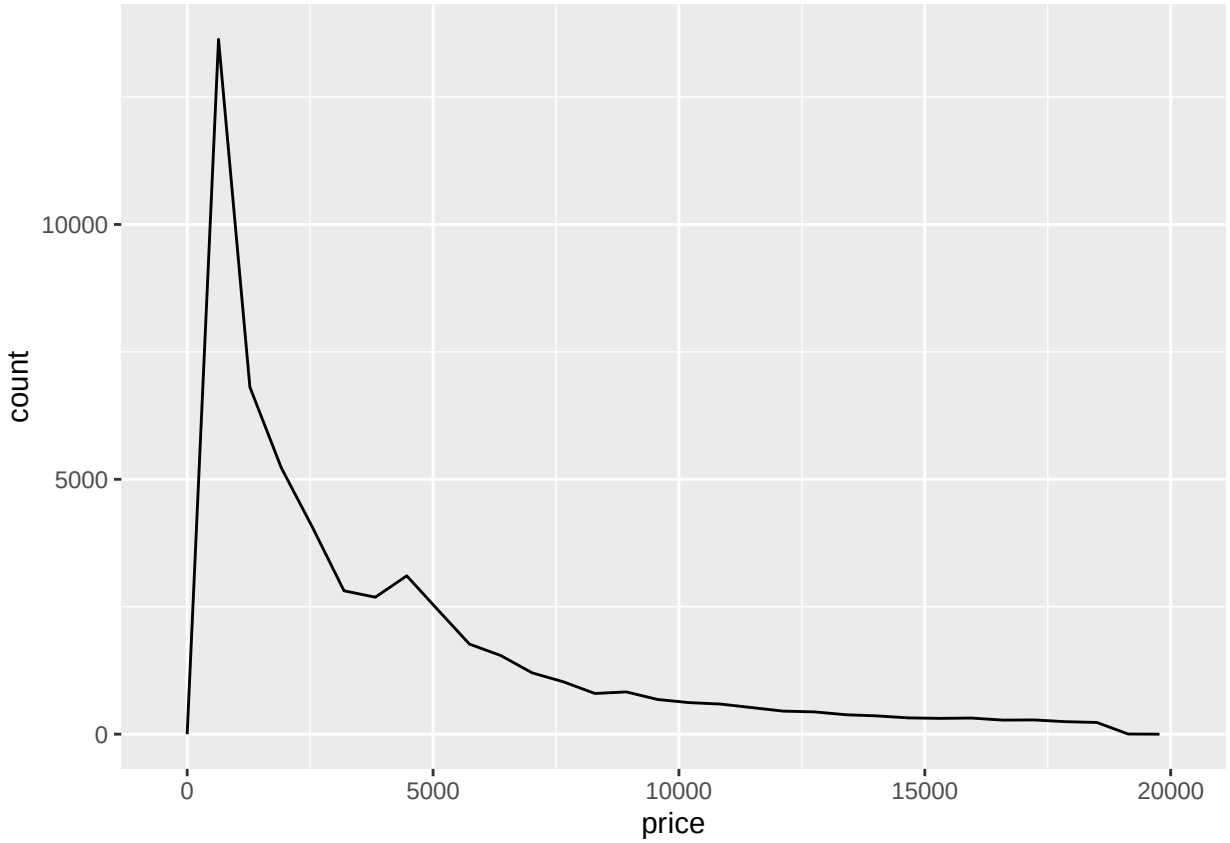

```
ggplot(diamonds, aes(price)) +
  geom_histogram() +
  facet_grid(cut ~ color,scales = "free" )
```



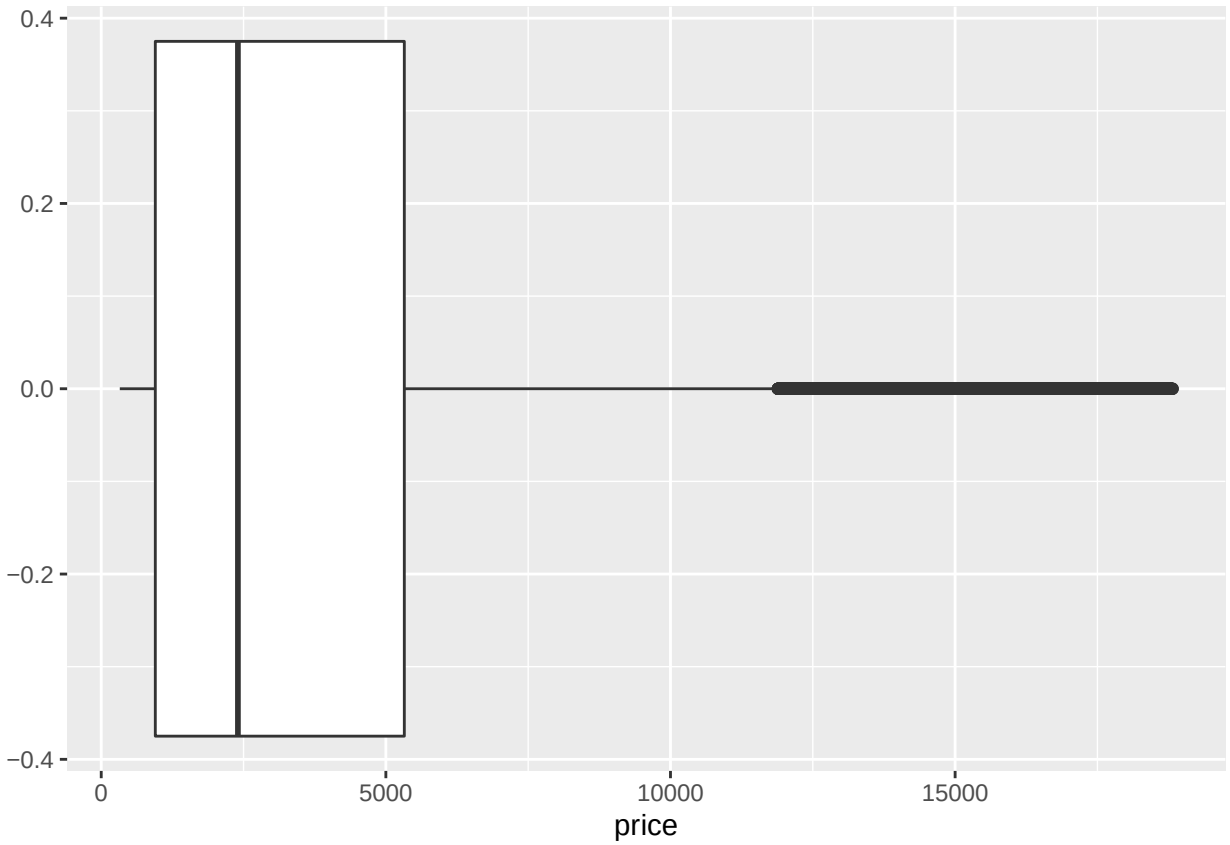

```
ggplot(diamonds, aes(x=price,fill=cut)) +
  geom_density(alpha=.3)
```




```
# freqpoly
ggplot(diamonds, aes(x=price)) +
  geom_freqpoly()
```

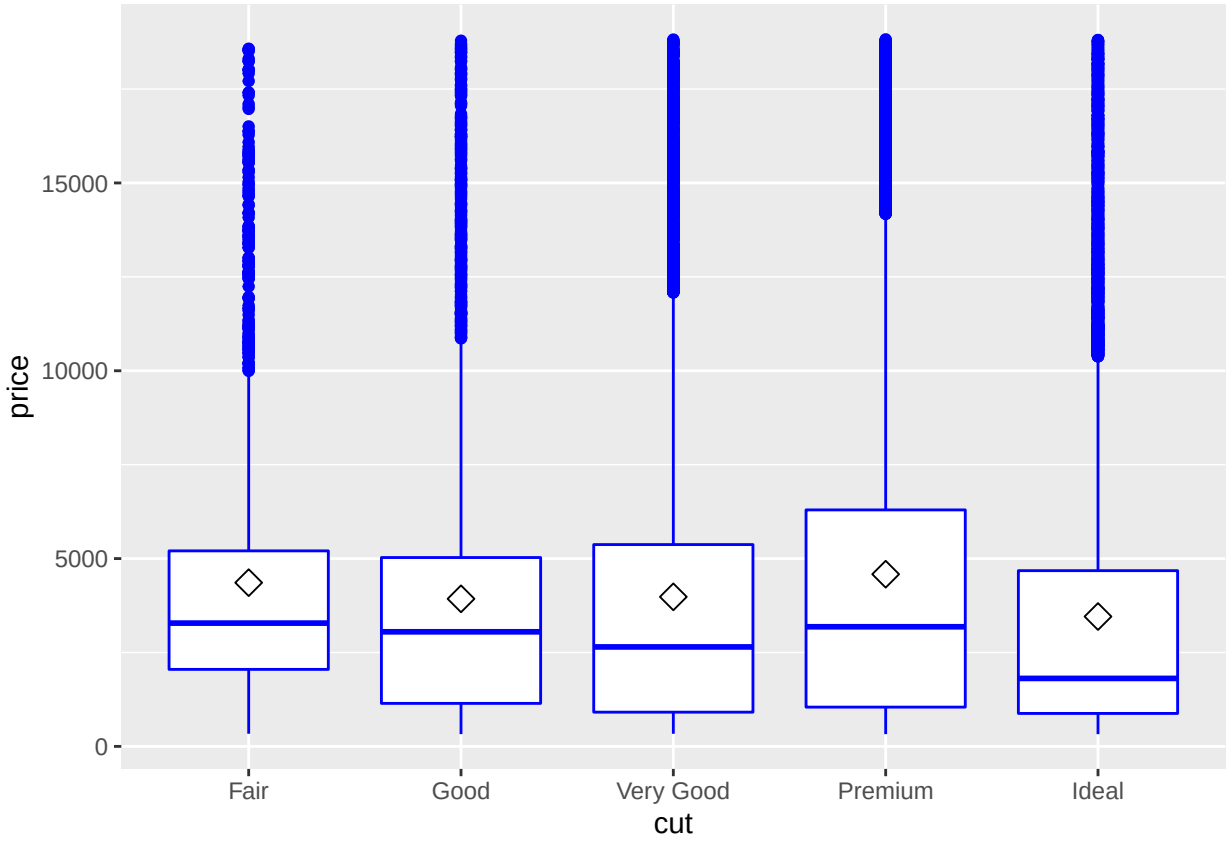



```
# boxplot
ggplot(diamonds, aes(x=price)) +
  geom_boxplot()
```




```
# boxplot'a ortalama eklemek
ggplot(diamonds, aes(x=cut,y=price)) +
  geom_boxplot(color="blue")+
  stat_summary(fun.y = "mean", geom = "point", shape = 5, size = 3)
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```



2.5 Grafikleri Kaydetmek

```
my_plot <- economics %>%
  mutate(uemploy_mom=unemploy/lag(unemploy ) * 100 - 100,
         growth=ifelse(uemploy_mom>0,"pozitif","negatif")) %>%
  na.omit() %>%
  filter(lubridate::year(date)>=2010) %>%
  ggplot(aes(x=date,y=uemploy_mom,fill=growth))+
  geom_col() +
  theme(legend.position = "none") +
  labs(y="Aylık Değişim",
       title="Yıllara göre Aylık İstihdam Değişimi (2010-2015)")

ggsave("myplot.pdf", my_plot, width = 20, height = 8, units = "cm")
ggsave("myplot.png", my_plot,width = 20, height = 8, unit = "cm", dpi = 300)
```