Ch6. Factors and tables

Jongrak Jeong

2023-01-14

1. Foractors and table

In R, “factor” is ‘categorical’ or ‘discrete’ variable(vector)

character vector or numeric vector > factor

Factor?

```
set.seed(1)
alpha <- sample(c("A", "B", "C"), 25, replace = T)
f <- factor(alpha)

alpha
```

```
##  [1] "A" "C" "A" "B" "A" "C" "C" "B" "B" "C" "C" "A" "A" "A" "B" "B" "B" "B" "C"
## [20] "A" "C" "A" "A" "A" "A"
```

```
f
```

```
##  [1] A C A B A C C B B C C A A A B B B B C A C A A A A
## Levels: A B C
```

```
z <-- sample(1:5, 25, replace = T)
z
```

```
##  [1] -5 -5 -2 -2 -1 -4 -1 -4 -3 -2 -2 -4 -4 -4 -2 -4 -1 -1 -4 -1 -2 -3 -2 -2 -5
```

```
g <- factor(z)
g
```

```
##  [1] -5 -5 -2 -2 -1 -4 -1 -4 -3 -2 -2 -4 -4 -4 -2 -4 -1 -1 -4 -1 -2 -3 -2 -2 -5
## Levels: -5 -4 -3 -2 -1
```

```
data.frame(f = f, g = g)
```

```
##    f  g
## 1  A -5
## 2  C -5
## 3  A -2
## 4  B -2
## 5  A -1
## 6  C -4
## 7  C -1
## 8  B -4
## 9  B -3
## 10 C -2
```

```
## 11 C -2
## 12 A -4
## 13 A -4
## 14 A -4
## 15 B -2
## 16 B -4
## 17 B -1
## 18 B -1
## 19 C -4
## 20 A -1
## 21 C -2
## 22 A -3
## 23 A -2
## 24 A -2
## 25 A -5
```

```
str(data.frame(f = f, g = g))
```

```
## 'data.frame':    25 obs. of  2 variables:
##  $ f: Factor w/ 3 levels "A","B","C": 1 3 1 2 1 3 3 2 2 3 ...
##  $ g: Factor w/ 5 levels "-5","-4","-3",..: 1 1 4 4 5 2 5 2 3 4 ...
```

table(), addmargins()

```
table(f)
```

```
## f
##  A  B  C
## 11  7  7
```

```
tab <- table(f, g)
tab
```

```
##    g
## f   -5 -4 -3 -2 -1
##   A  2  3  1  3  2
##   B  0  2  1  2  2
##   C  1  2  0  3  1
```

```
addmargins(table(f))
```

```
## f
##   A   B   C Sum
##  11   7   7  25
```

```
addmargins(tab)
```

```
##      g
## f     -5 -4 -3 -2 -1 Sum
##   A    2  3  1  3  2  11
##   B    0  2  1  2  2   7
##   C    1  2  0  3  1   7
##   Sum  3  7  2  8  5  25
```

```
class(tab)
```

```
## [1] "table"
```

```
dim(tab)
```

```
## [1] 3 5
```

```
rownames(tab)
```

```
## [1] "A" "B" "C"
```

```
colnames(tab)
```

```
## [1] "-5" "-4" "-3" "-2" "-1"
```

2. summarize the data: tapply() and aggregate()

tapply(x, f, function)

```
set.seed(2); x <- round(rnorm(25, 50, 10))
data.frame(x = x, f = f)
```

```
##     x f
## 1  41 A
## 2  52 C
## 3  66 A
## 4  39 B
## 5  49 A
## 6  51 C
## 7  57 C
## 8  48 B
## 9  70 B
## 10 49 C
## 11 54 C
## 12 60 A
## 13 46 A
## 14 40 A
## 15 68 B
## 16 27 B
## 17 59 B
## 18 50 B
## 19 60 C
## 20 54 A
## 21 71 C
## 22 38 A
## 23 66 A
## 24 70 A
## 25 50 A
```

```
tapply(x, f, median)
```

```
##  A  B  C
## 50 50 54
```

```
tapply(x, f, max)
```

```
##  A  B  C
## 70 70 71
```

```
tapply(x, f, min)
```

```
##  A  B  C
## 38 27 49
```

```
tapply(x, f, function(t) max(t) - min(t)) # apply new function to x (factor: f)
```

```
##  A  B  C
## 32 43 22
```

split() and sapply(): when target is not vector but data frame

```
s <- split(data.frame(x = x, z = z), f)

s
```

```
## $A
##     x  z
## 1  41 -5
## 3  66 -2
## 5  49 -1
## 12 60 -4
## 13 46 -4
## 14 40 -4
## 20 54 -1
## 22 38 -3
## 23 66 -2
## 24 70 -2
## 25 50 -5
##
## $B
##     x  z
## 4  39 -2
## 8  48 -4
## 9  70 -3
## 15 68 -2
## 16 27 -4
## 17 59 -1
## 18 50 -1
##
## $C
##     x  z
## 2  52 -5
## 6  51 -4
## 7  57 -1
## 10 49 -2
## 11 54 -2
## 19 60 -4
## 21 71 -2
```

```
class(s)
```

```
## [1] "list"
```

```
sapply(s, apply, 2, median)
```

```
##    A  B  C
## x 50 50 54
## z -3 -2 -2
```

aggregate(x, list(f, g), function)

```
aggregate(data.frame(x = x, z = z)$x, list(f), sum)
```

```
##   Group.1   x
## 1       A 580
## 2       B 361
## 3       C 394
```

```
aggregate(data.frame(x = x, z = z)$x, list(f, g), sum)
```

```
##    Group.1 Group.2   x
## 1        A      -5  91
## 2        C      -5  52
## 3        A      -4 146
## 4        B      -4  75
## 5        C      -4 111
## 6        A      -3  38
## 7        B      -3  70
## 8        A      -2 202
## 9        B      -2 107
## 10       C      -2 174
## 11       A      -1 103
## 12       B      -1 109
## 13       C      -1  57
```

3. cut()

cut(x, breaks): numric x > breaks > factors (binning)

```
x <- runif(100, 0, 10)
x
```

```
##   [1] 0.07109038 0.14693911 6.83403423 9.29720222 2.75401199 8.11859695
##   [7] 7.85878913 9.88902156 6.13952910 7.10185730 7.70027857 8.86984157
##  [13] 6.25121730 2.60300035 8.59073118 4.37488002 3.88144758 4.61501105
##  [19] 2.18675193 0.65935510 2.75701027 3.10381097 0.42175526 1.84673463
##  [25] 1.83373228 7.55462416 2.88059732 8.67844662 4.02642736 5.72685004
##  [31] 3.50642575 6.71998928 0.25050357 4.01101038 1.99976530 8.56525001
##  [37] 9.71515429 3.23722437 7.33191433 3.40068240 9.76755185 3.97016412
##  [43] 3.79998879 5.60387630 4.63808179 1.96776827 4.26943403 0.93025187
##  [49] 1.15309127 4.40031654 2.00934730 4.27639073 9.80599982 8.28922126
##  [55] 2.86973855 5.95916897 8.98971946 4.53377000 1.47417779 1.28676983
##  [61] 0.24656338 7.36311375 3.73358564 5.74376940 8.25328013 8.13695674
##  [67] 8.72696340 1.10554900 9.52700237 5.69002081 0.36868471 2.45290916
##  [73] 9.78884799 8.85737232 2.40982898 7.57211570 5.62836519 3.05103095
##  [79] 6.93654087 3.35945604 2.06109444 9.19276256 0.22812450 9.63759745
##  [85] 3.15865244 6.65608417 5.33543303 8.17796719 1.85263510 3.99517552
##  [91] 1.78453260 2.85434211 6.29469827 3.00100282 4.43673962 7.30200940
##  [97] 6.68163537 3.11657001 4.78578083 2.91410151
```

```r
y <- 5 + 0.5 * (x - 5) + rnorm(100)
x.cut <- cut(x, 0:10)
class(x.cut)
```

```
## [1] "factor"
```

```r
x.cut
```

```
##   [1] (0,1]  (0,1]  (6,7]  (9,10] (2,3]  (8,9]  (7,8]  (9,10] (6,7]  (7,8]
##  [11] (7,8]  (8,9]  (6,7]  (2,3]  (8,9]  (4,5]  (3,4]  (4,5]  (2,3]  (0,1]
##  [21] (2,3]  (3,4]  (0,1]  (1,2]  (1,2]  (7,8]  (2,3]  (8,9]  (4,5]  (5,6]
##  [31] (3,4]  (6,7]  (0,1]  (4,5]  (1,2]  (8,9]  (9,10] (3,4]  (7,8]  (3,4]
##  [41] (9,10] (3,4]  (3,4]  (5,6]  (4,5]  (1,2]  (4,5]  (0,1]  (1,2]  (4,5]
##  [51] (2,3]  (4,5]  (9,10] (8,9]  (2,3]  (5,6]  (8,9]  (4,5]  (1,2]  (1,2]
##  [61] (0,1]  (7,8]  (3,4]  (5,6]  (8,9]  (8,9]  (8,9]  (1,2]  (9,10] (5,6]
##  [71] (0,1]  (2,3]  (9,10] (8,9]  (2,3]  (7,8]  (5,6]  (3,4]  (6,7]  (3,4]
##  [81] (2,3]  (9,10] (0,1]  (9,10] (3,4]  (6,7]  (5,6]  (8,9]  (1,2]  (3,4]
##  [91] (1,2]  (2,3]  (6,7]  (3,4]  (4,5]  (7,8]  (6,7]  (3,4]  (4,5]  (2,3]
## Levels: (0,1] (1,2] (2,3] (3,4] (4,5] (5,6] (6,7] (7,8] (8,9] (9,10]
```

```r
cbind(x, x.cut)
```

```
##                 x x.cut
##   [1,] 0.07109038     1
##   [2,] 0.14693911     1
##   [3,] 6.83403423     7
##   [4,] 9.29720222    10
##   [5,] 2.75401199     3
##   [6,] 8.11859695     9
##   [7,] 7.85878913     8
##   [8,] 9.88902156    10
##   [9,] 6.13952910     7
##  [10,] 7.10185730     8
##  [11,] 7.70027857     8
##  [12,] 8.86984157     9
##  [13,] 6.25121730     7
##  [14,] 2.60300035     3
##  [15,] 8.59073118     9
##  [16,] 4.37488002     5
##  [17,] 3.88144758     4
##  [18,] 4.61501105     5
##  [19,] 2.18675193     3
##  [20,] 0.65935510     1
##  [21,] 2.75701027     3
##  [22,] 3.10381097     4
##  [23,] 0.42175526     1
##  [24,] 1.84673463     2
##  [25,] 1.83373228     2
##  [26,] 7.55462416     8
##  [27,] 2.88059732     3
##  [28,] 8.67844662     9
##  [29,] 4.02642736     5
##  [30,] 5.72685004     6
##  [31,] 3.50642575     4
##  [32,] 6.71998928     7
##  [33,] 0.25050357     1
```

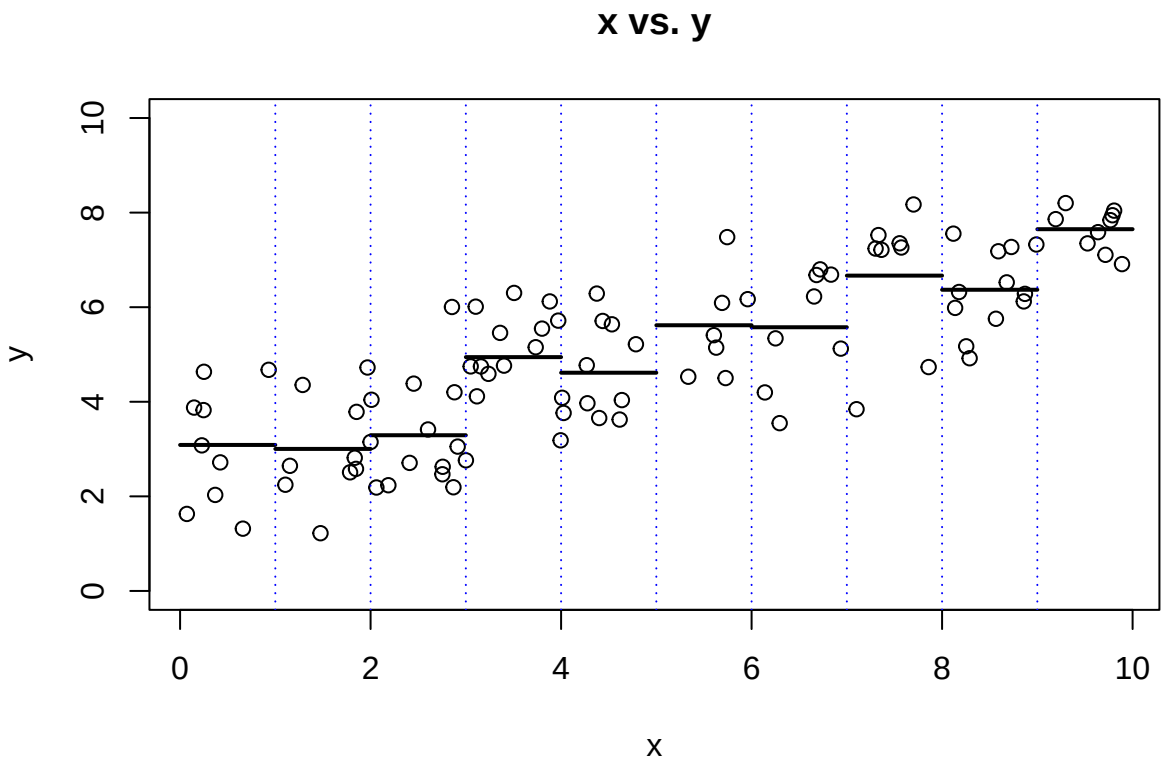
```
##  [34,] 4.01101038     5
##  [35,] 1.99976530     2
##  [36,] 8.56525001     9
##  [37,] 9.71515429    10
##  [38,] 3.23722437     4
##  [39,] 7.33191433     8
##  [40,] 3.40068240     4
##  [41,] 9.76755185    10
##  [42,] 3.97016412     4
##  [43,] 3.79998879     4
##  [44,] 5.60387630     6
##  [45,] 4.63808179     5
##  [46,] 1.96776827     2
##  [47,] 4.26943403     5
##  [48,] 0.93025187     1
##  [49,] 1.15309127     2
##  [50,] 4.40031654     5
##  [51,] 2.00934730     3
##  [52,] 4.27639073     5
##  [53,] 9.80599982    10
##  [54,] 8.28922126     9
##  [55,] 2.86973855     3
##  [56,] 5.95916897     6
##  [57,] 8.98971946     9
##  [58,] 4.53377000     5
##  [59,] 1.47417779     2
##  [60,] 1.28676983     2
##  [61,] 0.24656338     1
##  [62,] 7.36311375     8
##  [63,] 3.73358564     4
##  [64,] 5.74376940     6
##  [65,] 8.25328013     9
##  [66,] 8.13695674     9
##  [67,] 8.72696340     9
##  [68,] 1.10554900     2
##  [69,] 9.52700237    10
##  [70,] 5.69002081     6
##  [71,] 0.36868471     1
##  [72,] 2.45290916     3
##  [73,] 9.78884799    10
##  [74,] 8.85737232     9
##  [75,] 2.40982898     3
##  [76,] 7.57211570     8
##  [77,] 5.62836519     6
##  [78,] 3.05103095     4
##  [79,] 6.93654087     7
##  [80,] 3.35945604     4
##  [81,] 2.06109444     3
##  [82,] 9.19276256    10
##  [83,] 0.22812450     1
##  [84,] 9.63759745    10
##  [85,] 3.15865244     4
##  [86,] 6.65608417     7
##  [87,] 5.33543303     6
```

```
##  [88,] 8.17796719     9
##  [89,] 1.85263510     2
##  [90,] 3.99517552     4
##  [91,] 1.78453260     2
##  [92,] 2.85434211     3
##  [93,] 6.29469827     7
##  [94,] 3.00100282     4
##  [95,] 4.43673962     5
##  [96,] 7.30200940     8
##  [97,] 6.68163537     7
##  [98,] 3.11657001     4
##  [99,] 4.78578083     5
## [100,] 2.91410151     3
```

```
y.local <- aggregate(y, list(x.cut), mean)
y.local
```

```
##    Group.1        x
## 1    (0,1] 3.087039
## 2    (1,2] 3.003359
## 3    (2,3] 3.292717
## 4    (3,4] 4.944227
## 5    (4,5] 4.614255
## 6    (5,6] 5.618110
## 7    (6,7] 5.575787
## 8    (7,8] 6.667601
## 9    (8,9] 6.368346
## 10  (9,10] 7.649198
```

```
plot(x, y, ylim = c(0, 10), main = "x vs. y")
segments(0:9, y.local$x, 1:10, y.local$x, lwd = 2)
abline(v= 1:9, lty = "dotted", col = "blue")
```



4. Application: Major League Baseball

```
library(Lahman)
data(Salaries)
str(Salaries)
```

```
## 'data.frame':    26428 obs. of  5 variables:
##  $ yearID  : int  1985 1985 1985 1985 1985 1985 1985 1985 1985 1985 ...
##  $ teamID  : Factor w/ 35 levels "ANA","ARI","ATL",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ lgID    : Factor w/ 2 levels "AL","NL": 2 2 2 2 2 2 2 2 2 2 ...
##  $ playerID: chr  "barkele01" "bedrost01" "benedbr01" "campri01" ...
##  $ salary  : int  870000 550000 545000 633333 625000 800000 150000 483333 772000 250000 ...
```

```
Salaries.2013 <- subset(Salaries, yearID == 2013)
attach(Salaries.2013)
```

```
table(teamID)
```

```
## teamID
## ANA ARI ATL BAL BOS CAL CHA CHN CIN CLE COL DET FLO HOU KCA LAA LAN MIA MIL MIN
##   0  30  27  26  29   0  28  26  25  28  25  24   0  22  27  26  32  24  25  27
## ML4 MON NYA NYN OAK PHI PIT SDN SEA SFN SLN TBA TEX TOR WAS
##   0   0  31  30  31  26  28  28  26  28  28  23  29  31  25
```

```
tab <- table(teamID)
teamID.1 <- factor(teamID)
levels(teamID.1) <- names(tab)[tab > 0]
dim(table(teamID.1))
```

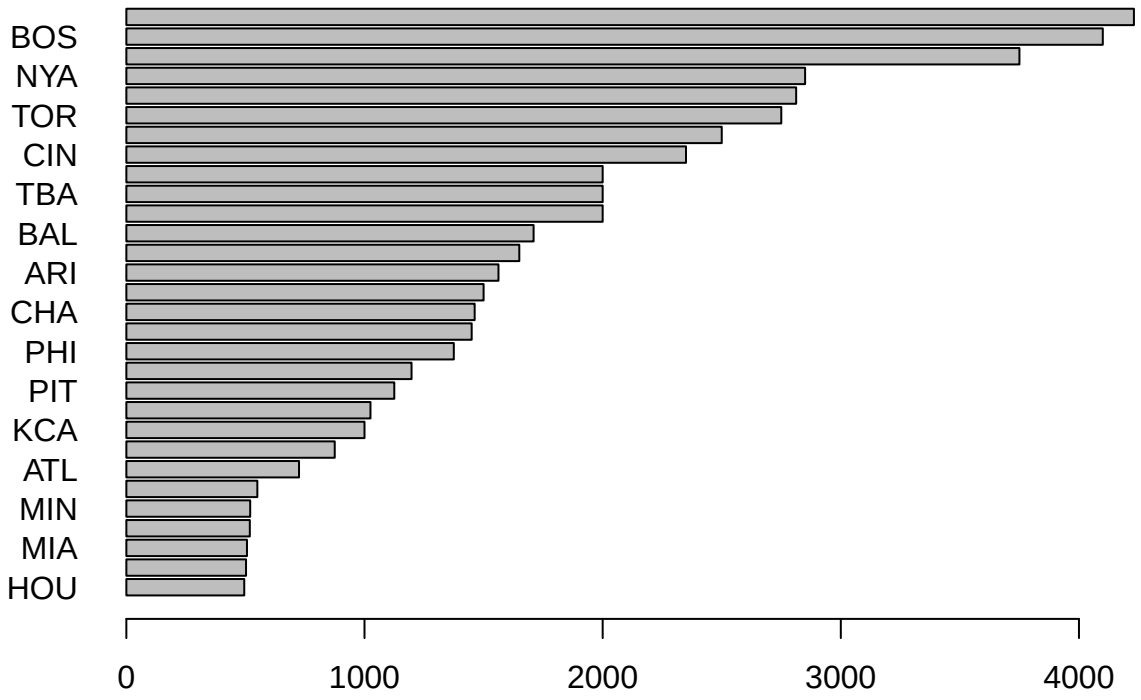
```
## [1] 30
```

```
tapply(salary, teamID.1, median)
```

```
##      ARI      ATL      BAL      BOS      CHA      CHN      CIN      CLE      COL      DET
## 1562500   725000 1710000 4100000 1462500 2000000 2350000   875000 1500000 3750000
##      HOU      KCA      LAA      LAN      MIA      MIL      MIN      NYA      NYN      OAK
##   495000 1000000 2812500 4230750   506450 1450000   520000 2850000   502586   550000
##      PHI      PIT      SDN      SEA      SFN      SLN      TBA      TEX      TOR      WAS
## 1375000 1125000 1197500 1025000 1650000   518500 2000000 2000000 2750000 2500000
```

```
tab <- table(teamID)
par(family = "mono")
barplot(tapply(salary / 1000, teamID.1, median), horiz = T, las = 1,
        main = "median salary (in 1,000)")
```


```
par(mar = c(2, 4, 4, 2))
barplot(sort(tapply(salary / 1000, teamID.1, median)), horiz = T, las = 1,
        main = "median salary (in 1,000)")
```




```
Salaries.2013.s <- split(Salaries.2013, teamID.1)
class(Salaries.2013.s)
```

```
## [1] "list"
```

```
names(Salaries.2013.s)
```

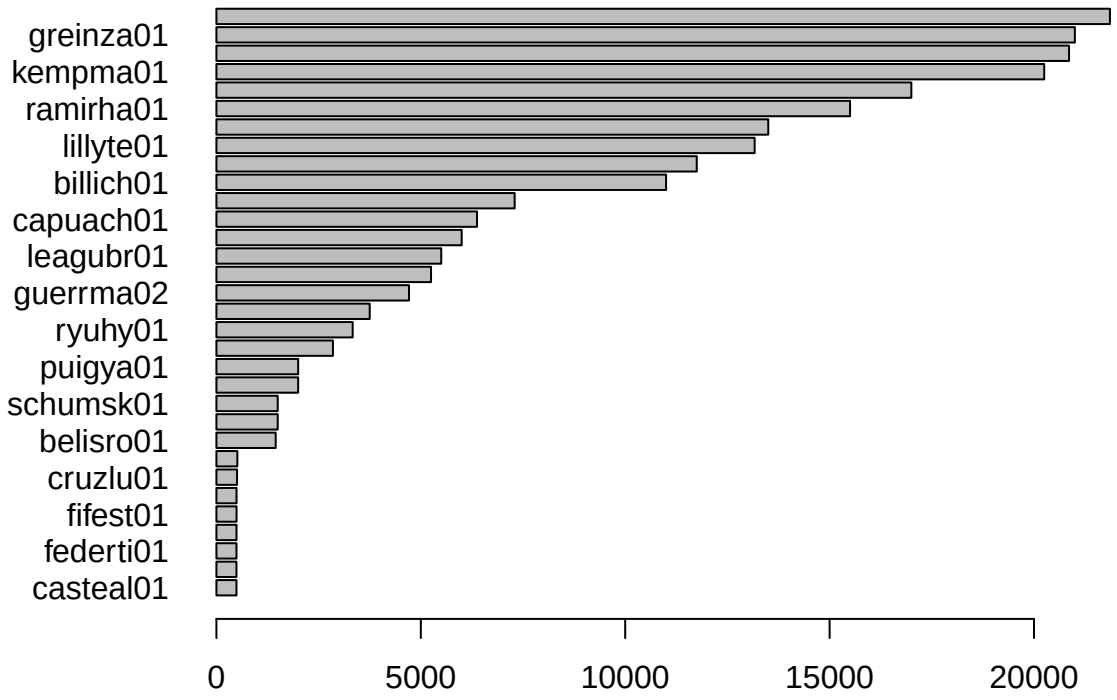
```
##  [1] "ARI" "ATL" "BAL" "BOS" "CHA" "CHN" "CIN" "CLE" "COL" "DET" "HOU" "KCA"
## [13] "LAA" "LAN" "MIA" "MIL" "MIN" "NYA" "NYN" "OAK" "PHI" "PIT" "SDN" "SEA"
## [25] "SFN" "SLN" "TBA" "TEX" "TOR" "WAS"
```

```
attach(Salaries.2013.s$LAN)
```

```
## The following objects are masked from Salaries.2013:
##
##     lgID, playerID, salary, teamID, yearID
```

```
names(salary) <- playerID

par(mar = c(2, 7, 4, 2))
barplot(sort(salary / 1000), horiz = T, las = 1,
        main = "LA Dodgers salary (in 1000)")
```




```
data(People)
str(People)
```

```
## 'data.frame':    20370 obs. of  26 variables:
##  $ playerID    : chr  "aardsda01" "aaronha01" "aaronto01" "aasedo01" ...
##  $ birthYear   : int  1981 1934 1939 1954 1972 1985 1850 1877 1869 1866 ...
##  $ birthMonth  : int  12 2 8 9 8 12 11 4 11 10 ...
##  $ birthDay    : int  27 5 5 8 25 17 4 15 11 14 ...
##  $ birthCountry: chr  "USA" "USA" "USA" "USA" ...
##  $ birthState  : chr  "CO" "AL" "AL" "CA" ...
##  $ birthCity   : chr  "Denver" "Mobile" "Mobile" "Orange" ...
##  $ deathYear   : int  NA 2021 1984 NA NA NA 1905 1957 1962 1926 ...
##  $ deathMonth  : int  NA 1 8 NA NA NA 5 1 6 4 ...
##  $ deathDay    : int  NA 22 16 NA NA NA 17 6 11 27 ...
##  $ deathCountry: chr  NA "USA" "USA" NA ...
##  $ deathState  : chr  NA "GA" "GA" NA ...
##  $ deathCity   : chr  NA "Atlanta" "Atlanta" NA ...
##  $ nameFirst   : chr  "David" "Hank" "Tommie" "Don" ...
##  $ nameLast    : chr  "Aardsma" "Aaron" "Aaron" "Aase" ...
##  $ nameGiven   : chr  "David Allan" "Henry Louis" "Tommie Lee" "Donald William" ...
##  $ weight      : int  215 180 190 190 184 235 192 170 175 169 ...
##  $ height      : int  75 72 75 75 73 74 72 71 71 68 ...
##  $ bats        : Factor w/ 3 levels "B","L","R": 3 3 3 3 2 2 3 3 3 2 ...
##  $ throws      : Factor w/ 3 levels "L","R","S": 2 2 2 2 1 1 2 2 2 1 ...
##  $ debut       : chr  "2004-04-06" "1954-04-13" "1962-04-10" "1977-07-26" ...
##  $ finalGame   : chr  "2015-08-23" "1976-10-03" "1971-09-26" "1990-10-03" ...
##  $ retroID     : chr  "aardd001" "aaroh101" "aarot101" "aased001" ...
##  $ bbrefID     : chr  "aardsda01" "aaronha01" "aaronto01" "aasedo01" ...
```

```
##  $ deathDate   : Date, format: NA "2021-01-22" ...
##  $ birthDate   : Date, format: "1981-12-27" "1934-02-05" ...
```

```
nameLast <- People$nameLast[People$playerID %in% playerID]
names(salary) <- nameLast
```

```
barplot(sort(salary / 1000), horiz = T, las = 1,
        main = "LA Dodgers salary (in 1,000)")
```