Ch5. Data Frames

Jongrak Jeong

2023-01-13

Default Setting

```
setwd("~/Library/Mobile Documents/com~apple~CloudDocs/Study/2_Data Science/Practice/R Programming by He
```

1. Data frame

Load the data: read csv(comma separated variables)

```
exams <- read.csv("exams.csv", header = T)

str(exams) # overview
```

```
## 'data.frame':    26 obs. of  3 variables:
##  $ course.id: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ mid      : int  8 22 25 25 21 12 12 29 40 25 ...
##  $ final    : int  11 24 31 13 34 26 6 36 34 38 ...
```

```
class(exams)
```

```
## [1] "data.frame"
```

```
dim(exams) # check the number of units and variables/features
```

```
## [1] 26  3
```

```
colnames(exams) # check the variables
```

```
## [1] "course.id" "mid"       "final"
```

```
exams
```

```
##    course.id mid final
## 1          1   8    11
## 2          2  22    24
## 3          3  25    31
## 4          4  25    13
## 5          5  21    34
## 6          6  12    26
## 7          7  12     6
## 8          8  29    36
## 9          9  40    34
## 10        10  25    38
## 11        11   6    21
## 12        12  28    34
```

```
## 13        13  29    36
## 14        14  12    40
## 15        15  24    38
## 16        16  31    20
## 17        17  11    26
## 18        18  33    32
## 19        19  12    17
## 20        20  27    16
## 21        21  34    29
## 22        22  40    36
## 23        23  30    26
## 24        24   2    NA
## 25        25  32    36
## 26        26  25    26
```

```
is.list(exams) # data.frame is also 'list'
```

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

```
length(exams) # The number of variables is length
```

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

```
names(exams) # == colnames()
```

```
## [1] "course.id" "mid"       "final"
```

Indexing: [row, column]

```
exams[exams$course.id == 10, ] # 10th unit
```

```
##    course.id mid final
## 10        10  25    38
```

```
exams[c(10, 20), ] # 10th and 20th unit
```

```
##    course.id mid final
## 10        10  25    38
## 20        20  27    16
```

```
exams[exams$course.id == 10, "final"] # The final grade of 10th unit
```

```
## [1] 38
```

Applying the functions 1

```
# The specific column of data frame is 'vector' > we can apply the functions such as mean() and media()
median(exams[,2]) # median of mid term grades
```

```
## [1] 25
```

```
median(exams$mid)
```

```
## [1] 25
```

Applying the functions 2: NA

```
median(exams$final)
```

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

```
median(exams$final, na.rm = T) # omit 'NA'
```

```
## [1] 29
```

```
# If we consider NA 0
exams$final[is.na(exams$final)] <- 0
exams[21:26, ]
```

```
##    course.id mid final
## 21        21  34    29
## 22        22  40    36
## 23        23  30    26
## 24        24   2     0
## 25        25  32    36
## 26        26  25    26
```

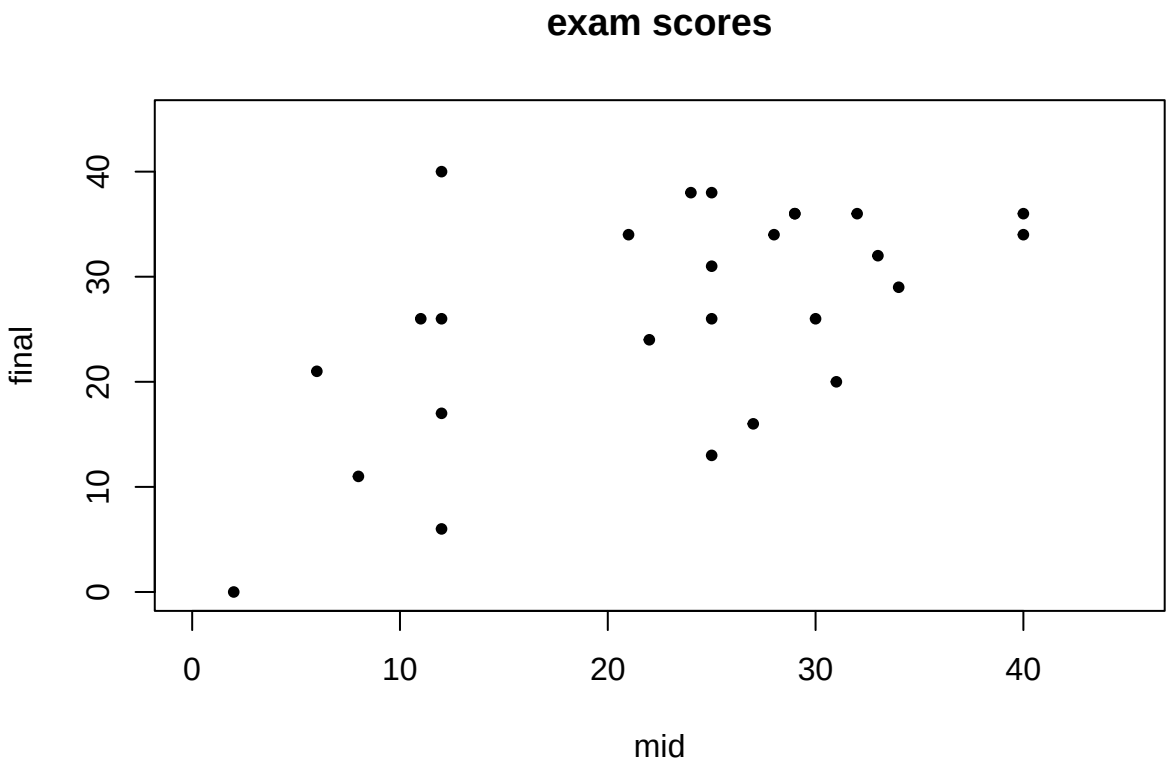
```
median(exams$final)
```

```
## [1] 27.5
```

Scatter plot: plot(x, y, data = DF) or plot(y ~ x, DF)

```
plot(final ~ mid, exams, pch = 20,
     xlim = c(0,45), ylim = c(0, 45),
     main = "exam scores")

with(exams, plot(final ~ mid, pch = 20,
     xlim = c(0,45), ylim = c(0, 45),
     main = "exam scores"))
```



Correlation: cor()

```
cor(exams$mid, exams$final)
```

```
## [1] 0.5709343
```

```
with(exams, cor(mid, final))
```

```
## [1] 0.5709343
```

```
round(with(exams, cor(mid, final)), 3)
```

```
## [1] 0.571
```

2. Merging two data frames

Merging 'exams' with 'book2' - homework and project

```
book2 <- read.csv("book2.csv", header = T)

str(book2)
```

```
## 'data.frame':    26 obs. of  4 variables:
##  $ student.name: chr  "bd" "ce" "cn" "de" ...
##  $ homework    : int  22 36 31 24 37 36 37 28 37 34 ...
##  $ project     : int  NA 7 NA NA 17 10 8 NA 4 10 ...
##  $ course.id   : int  8 23 20 1 16 2 25 4 14 11 ...
```

```
summary(book2) # check NA
```

```
##  student.name          homework        project         course.id
##  Length:26          Min.   : 0.00   Min.   : 4.000   Min.   : 1.00
##  Class :character   1st Qu.:34.00   1st Qu.: 6.250   1st Qu.: 7.25
##  Mode  :character   Median :36.00   Median : 8.000   Median :13.50
##                     Mean   :33.77   Mean   : 9.778   Mean   :13.50
##                     3rd Qu.:38.00   3rd Qu.:10.000   3rd Qu.:19.75
##                     Max.   :40.00   Max.   :23.000   Max.   :26.00
##                                     NA's   :8
```

```
book2$project[is.na(book2$project)] <- 0
```

The facts we should consider

1) two data frames with same order > column binding : class.record <- cbind(exams, book2)
2) two data frames with different order > find and define 'unit identifier' : find 'key' variable with merge() function

Merge()

```
class.record <- merge(exams, book2)
str(class.record)
```

```
## 'data.frame':    26 obs. of  6 variables:
##  $ course.id   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ mid         : int  8 22 25 25 21 12 12 29 40 25 ...
##  $ final       : num  11 24 31 13 34 26 6 36 34 38 ...
```

```
##  $ student.name: chr  "de" "ev" "td" "hi" ...
##  $ homework    : int  24 36 36 28 38 38 31 22 39 40 ...
##  $ project     : num  0 10 0 0 8 0 0 0 20 16 ...
```

```
head(class.record, 5)
```

```
##   course.id mid final student.name homework project
## 1         1   8    11           de       24       0
## 2         2  22    24           ev       36      10
## 3         3  25    31           td       36       0
## 4         4  25    13           hi       28       0
## 5         5  21    34           yk       38       8
```

Without merge()

Sorting book2 by course.id (make same order)

```
i.order <- order(book2$course.id) # index order
book2.sort <- book2[i.order, ] # sort book2
head(book2.sort, 5)
```

```
##    student.name homework project course.id
## 4            de       24       0         1
## 6            ev       36      10         2
## 20           td       36       0         3
## 8            hi       28       0         4
## 25           yk       38       8         5
```

Bind with cbind()

```
class.record.1 <- cbind(exams, book2.sort[ , -4]) # omit 4th column (it is key variable)
class.record.2 <- cbind(exams, book2.sort[ , c(-3, -4)]) # without project grade

head(class.record.1, 5)
```

```
##    course.id mid final student.name homework project
## 4          1   8    11           de       24       0
## 6          2  22    24           ev       36      10
## 20         3  25    31           td       36       0
## 8          4  25    13           hi       28       0
## 25         5  21    34           yk       38       8
```

Application of merge()

merge(x, y, by.x, by.y) or merge(x, y, by)

x, y > target data frames by.x and by.y > key variables (if these are same, we can just use 'by')

Examples: authors and books data (merge() run examples)

```
authors <- data.frame(
    ## I(*) : use character columns of names to get sensible sort order
    surname = I(c("Tukey", "Venables", "Tierney", "Ripley", "McNeil")),
    nationality = c("US", "Australia", "US", "UK", "Australia"),
    deceased = c("yes", rep("no", 4)))
```

```
books <- data.frame(
    name = I(c("Tukey", "Venables", "Tierney",
             "Ripley", "Ripley", "McNeil", "R Core")),
    title = c("Exploratory Data Analysis",
              "Modern Applied Statistics ...",
              "LISP-STAT",
              "Spatial Statistics", "Stochastic Simulation",
              "Interactive Data Analysis",
              "An Introduction to R"),
    other.author = c(NA, "Ripley", NA, NA, NA, NA,
                     "Venables & Smith"))

m1 <- merge(authors, books, by.x = "surname", by.y = "name") # 'surname' variable == 'name' variable

m1
```

```
##    surname nationality deceased                         title other.author
## 1   McNeil   Australia       no     Interactive Data Analysis         <NA>
## 2   Ripley          UK       no            Spatial Statistics         <NA>
## 3   Ripley          UK       no         Stochastic Simulation         <NA>
## 4  Tierney          US       no                     LISP-STAT         <NA>
## 5    Tukey          US      yes     Exploratory Data Analysis         <NA>
## 6 Venables   Australia       no Modern Applied Statistics ...       Ripley
```

3. Apply()

apply(x, 1 or 2, f) # 1: by row, 2: by column

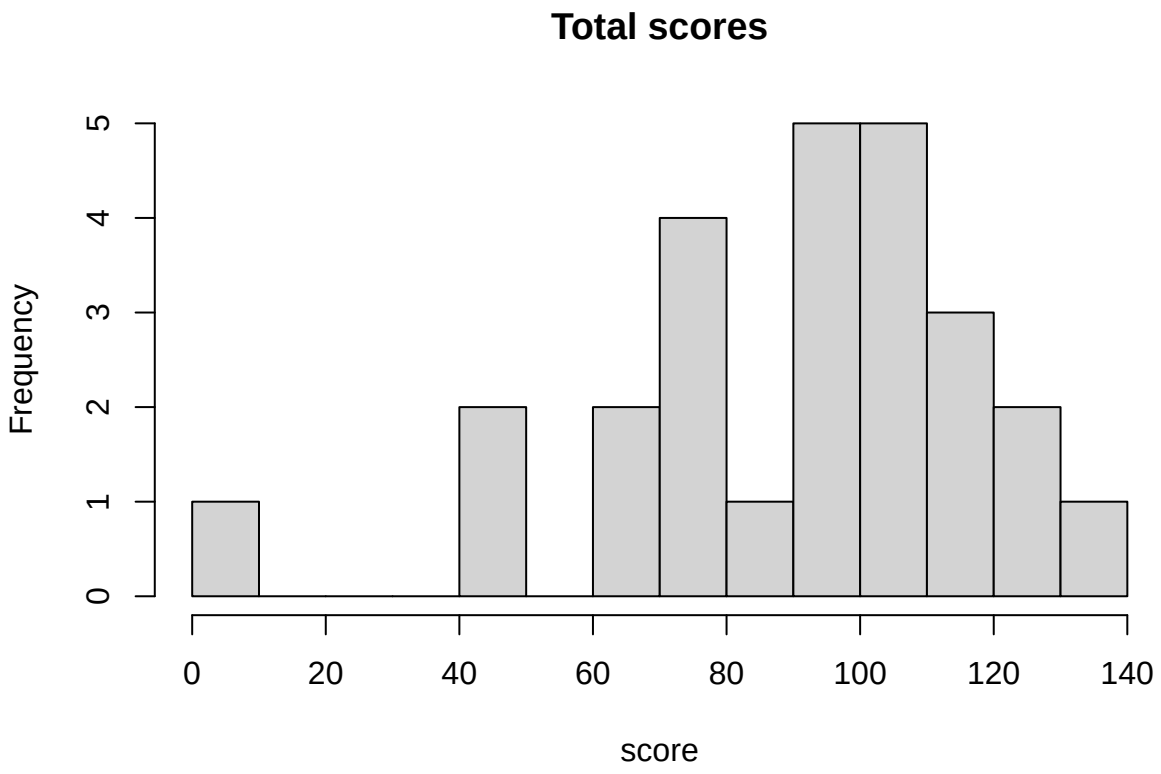
Summation of mid(2nd) column, final(3rd) column, homework(5th) column, and project(6th) column of class.record

```
# each student's total summation > by row
apply(class.record[, c(2, 3, 5, 6)], 1, sum)
```

```
##  [1]  43  92  92  66 101  76  49  87 133 119  71 109 114  93 124 105  77 106  69
## [20]  74 108 122  99   2 113  91
```

```
# make a new variable 'total'
class.record$total <- apply(class.record[, c(2, 3, 5, 6)], 1, sum)

# histogram of total score
hist(class.record$total,
     xlab = "score",
     nclass = 10,
     main = "Total scores")
```



Quantile

```
output <- apply(class.record[, c(2, 3, 5, 6, 7)], 2,
      quantile, prob = c(0, 0.25, 0.5, 0.75, 1))

output
```

```
##        mid final homework project  total
## 0%    2.00  0.00        0    0.00   2.00
## 25%  12.00 20.25       34    0.00  74.50
## 50%  25.00 27.50       36    6.50  92.50
## 75%  29.75 35.50       38    8.75 108.75
## 100% 40.00 40.00       40   23.00 133.00
```

```
class(output) # output of apply() is 'matrix' or 'array'
```

```
## [1] "matrix" "array"
```

4. Some methods for data preprocessing

Character vectors are be changed as 'factor' variables as they are put in a data frame....(???? updated?)

```
set.seed(1)
g <- sample(c("A", "B"), 100, replace = T)
x <- rnorm(100, 50, 10)

D <- data.frame(grp = g, score = x)
str(D)
```

```
## 'data.frame':    100 obs. of  2 variables:
##  $ grp  : chr  "A" "B" "A" "A" ...
##  $ score: num  54 43.9 53.4 38.7 64.3 ...
```

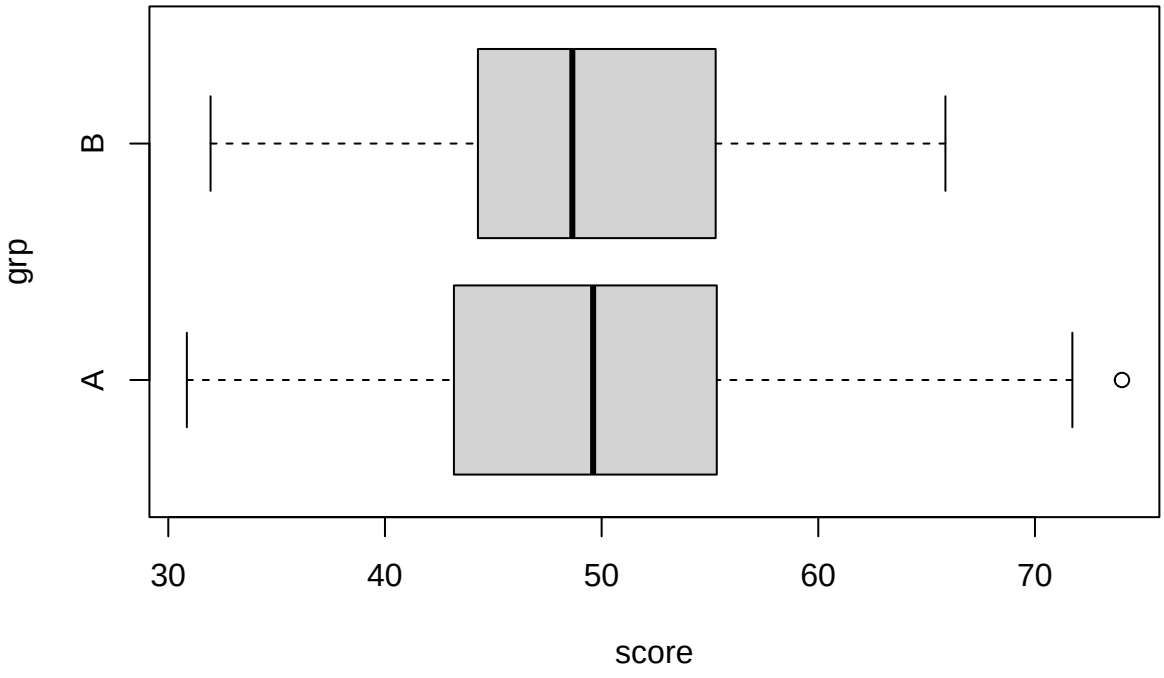
```
head(D)
```

```
##   grp    score
## 1   A 53.98106
## 2   B 43.87974
## 3   A 53.41120
## 4   A 38.70637
## 5   B 64.33024
## 6   A 69.80400
```

```
D$grp <- as.factor(D$grp)
str(D)
```

```
## 'data.frame':    100 obs. of  2 variables:
##  $ grp  : Factor w/ 2 levels "A","B": 1 2 1 1 2 1 1 1 2 2 ...
##  $ score: num  54 43.9 53.4 38.7 64.3 ...
```

```
boxplot(data = D,
        score ~ grp,
        xlab = "score",
        horizontal = T
        )
```



Complete.cases(): row without 'NA' > T, with 'NA' > F

```
set.seed(1)
x <- data.frame(math = rnorm(100), engl = rnorm(100))

x[sample(1:100, 20), 1] <- NA
x[sample(1:100, 20), 2] <- NA
```

```
x.1 <- x[complete.cases(x), ]

sum(complete.cases(x))
```

```
## [1] 66
```

```
str(x.1)
```

```
## 'data.frame':    66 obs. of  2 variables:
##  $ math: num  -0.626 0.184 1.595 0.33 -0.82 ...
##  $ engl: num  -0.6204 0.0421 0.158 -0.6546 1.7673 ...
```

Sorting with the order of a vector

x[order(a),] x[order(a, b),]

Example with mtcars dataset

```
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
# ascending by cyl
mtcars[order(mtcars$cyl), ]
```

```
##                    mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Fiat 128          32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic       30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla    33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona     21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Fiat X1-9         27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2     26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa      30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Volvo 142E        21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C         17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Ferrari Dino      19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
```

```
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
```

```
mtcars[with(mtcars, order(cyl)), ]
```

```
##                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
```

```
# descending by cyl
mtcars[order(-mtcars$cyl), ]
```

```
##                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
```

```
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
mtcars[with(mtcars, order(-cyl)), ]
```

```
##                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
```

```
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
# ascending by cyl and disp
mtcars[order(mtcars$cyl, mtcars$disp), ]
```

```
##                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Toyota Corolla     33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Honda Civic        30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Fiat 128           32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Fiat X1-9          27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Lotus Europa       30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Datsun 710         22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Toyota Corona      21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Porsche 914-2      26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Volvo 142E         21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## Merc 230           22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 240D          24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Ferrari Dino       19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Mazda RX4          21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag      21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Merc 280           19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C          17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Valiant            18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Hornet 4 Drive     21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Merc 450SE         16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL         17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC        15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Maserati Bora      15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## AMC Javelin        15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Dodge Challenger   15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## Camaro Z28         13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Ford Pantera L     15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Hornet Sportabout  18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Duster 360         14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Pontiac Firebird   19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Chrysler Imperial  14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Lincoln Continental 10.4  8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Cadillac Fleetwood  10.4  8 472.0 205 2.93 5.250 17.98  0  0    3    4
```

```
mtcars[with(mtcars, order(cyl, disp)), ]
```

```
##                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Toyota Corolla     33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
```

```
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
```

```
# ascending by cyl and descending by disp
mtcars[order(mtcars$cyl, -mtcars$disp), ]
```

```
##                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Merc 240D        24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230         22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Volvo 142E       21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## Porsche 914-2    26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Toyota Corona    21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Datsun 710       22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Lotus Europa     30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Fiat X1-9        27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Fiat 128         32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic      30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla   33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Hornet 4 Drive   21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Valiant          18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Merc 280         19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C        17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Mazda RX4        21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag    21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Ferrari Dino     19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98  0  0    3    4
```

```
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
```

```
mtcars[with(mtcars, order(cyl, -disp)), ]
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
```

Subset()

subset(x, criteria)

Example with airquality dataset

```
str(airquality)
```

```
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
air <- subset(airquality, Temp > quantile(Temp, 0.95))

# ascending sort
air.sort.a <- air[order(air$Temp), ]
air.sort.a
```

```
##     Ozone Solar.R Wind Temp Month Day
## 42     NA     259 10.9   93     6  11
## 126    73     183  2.8   93     9   3
## 127    91     189  4.6   93     9   4
## 121   118     225  2.3   94     8  29
## 123    85     188  6.3   94     8  31
## 122    84     237  6.3   96     8  30
## 120    76     203  9.7   97     8  28
```

```
# desending sort
air.sort.d <- air[order(-air$Temp), ]
air.sort.d
```

```
##     Ozone Solar.R Wind Temp Month Day
## 120    76     203  9.7   97     8  28
## 122    84     237  6.3   96     8  30
## 121   118     225  2.3   94     8  29
## 123    85     188  6.3   94     8  31
## 42     NA     259 10.9   93     6  11
## 126    73     183  2.8   93     9   3
## 127    91     189  4.6   93     9   4
```

5. Application: Pitching records and salaries of Park Chan Ho

Load dataset

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

```
## 'data.frame':    49430 obs. of  30 variables:
##  $ playerID: chr  "bechtge01" "brainas01" "fergubo01" "fishech01" ...
##  $ yearID  : int  1871 1871 1871 1871 1871 1871 1871 1871 1871 1871 ...
##  $ stint   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ teamID  : Factor w/ 149 levels "ALT","ANA","ARI",..: 97 142 90 111 90 136 111 56 97 136 ...
##  $ lgID    : Factor w/ 7 levels "AA","AL","FL",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ W       : int  1 12 0 4 0 0 0 6 18 12 ...
##  $ L       : int  2 15 0 16 1 0 1 11 5 15 ...
```

```
##  $ G       : int  3 30 1 24 1 1 3 19 25 29 ...
##  $ GS      : int  3 30 0 24 1 0 1 19 25 29 ...
##  $ CG      : int  2 30 0 22 1 0 1 19 25 28 ...
##  $ SHO     : int  0 0 0 1 0 0 0 1 0 0 ...
##  $ SV      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ IPouts  : int  78 792 3 639 27 3 39 507 666 747 ...
##  $ H       : int  43 361 8 295 20 1 20 261 285 430 ...
##  $ ER      : int  23 132 3 103 10 0 5 97 113 153 ...
##  $ HR      : int  0 4 0 3 0 0 0 5 3 4 ...
##  $ BB      : int  11 37 0 31 3 0 3 21 40 75 ...
##  $ SO      : int  1 13 0 15 0 0 1 17 15 12 ...
##  $ BAOpp   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ ERA     : num  7.96 4.5 27 4.35 10 0 3.46 5.17 4.58 5.53 ...
##  $ IBB     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ WP      : int  7 7 2 20 0 0 1 15 3 44 ...
##  $ HBP     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ BK      : int  0 0 0 0 0 0 0 2 0 0 ...
##  $ BFP     : int  146 1291 14 1080 57 3 70 876 1059 1334 ...
##  $ GF      : int  0 0 0 1 0 1 1 0 0 0 ...
##  $ R       : int  42 292 9 257 21 0 30 243 223 362 ...
##  $ SH      : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ SF      : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ GIDP    : int  NA NA NA NA NA NA NA NA NA NA ...
```

```
## '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 ...
```

Subset

```
Pitching.1 <- subset(Pitching, playerID == "parkch01")
Salaries.1 <- subset(Salaries, playerID == "parkch01")

head(Pitching.1)
```

```
##       playerID yearID stint teamID lgID  W  L  G GS CG SHO SV IPouts   H  ER HR
## 29683 parkch01   1994     1    LAN   NL  0  0  2  0  0   0  0     12   5   5  1
## 30247 parkch01   1995     1    LAN   NL  0  0  2  1  0   0  0     12   2   2  1
## 30851 parkch01   1996     1    LAN   NL  5  5 48 10  0   0  0    326  82  44  7
## 31448 parkch01   1997     1    LAN   NL 14  8 32 29  2   0  0    576 149  72 24
## 32028 parkch01   1998     1    LAN   NL 15  9 34 34  2   0  0    662 199  91 16
## 32667 parkch01   1999     1    LAN   NL 13 11 33 33  0   0  0    583 208 113 31
##        BB  SO BAOpp   ERA IBB WP HBP BK BFP GF   R SH SF GIDP
## 29683   5   6 0.294 11.25   0  0   1  0  23  1   5  0  0    0
## 30247   2   7 0.143  4.50   0  0   0  1  16  0   2  0  0    0
## 30851  71 119 0.209  3.64   3  4   4  3 477  7  48  8  1    9
## 31448  70 166 0.213  3.38   1  4   8  1 792  1  80  9  5    8
## 32028  97 191 0.244  3.71   1  6  11  2 946  0 101 11 10   25
## 32667 100 174 0.276  5.23   4 11  14  1 883  0 120 10  5    9
```

```
head(Salaries.1)
```

```
##       yearID teamID lgID playerID  salary
## 6965    1994    LAN   NL parkch01  109000
## 7898    1995    LAN   NL parkch01  114000
## 8849    1996    LAN   NL parkch01  124000
## 9780    1997    LAN   NL parkch01  270000
## 10757   1998    LAN   NL parkch01  700000
## 11762   1999    LAN   NL parkch01 2300000
```

Merge

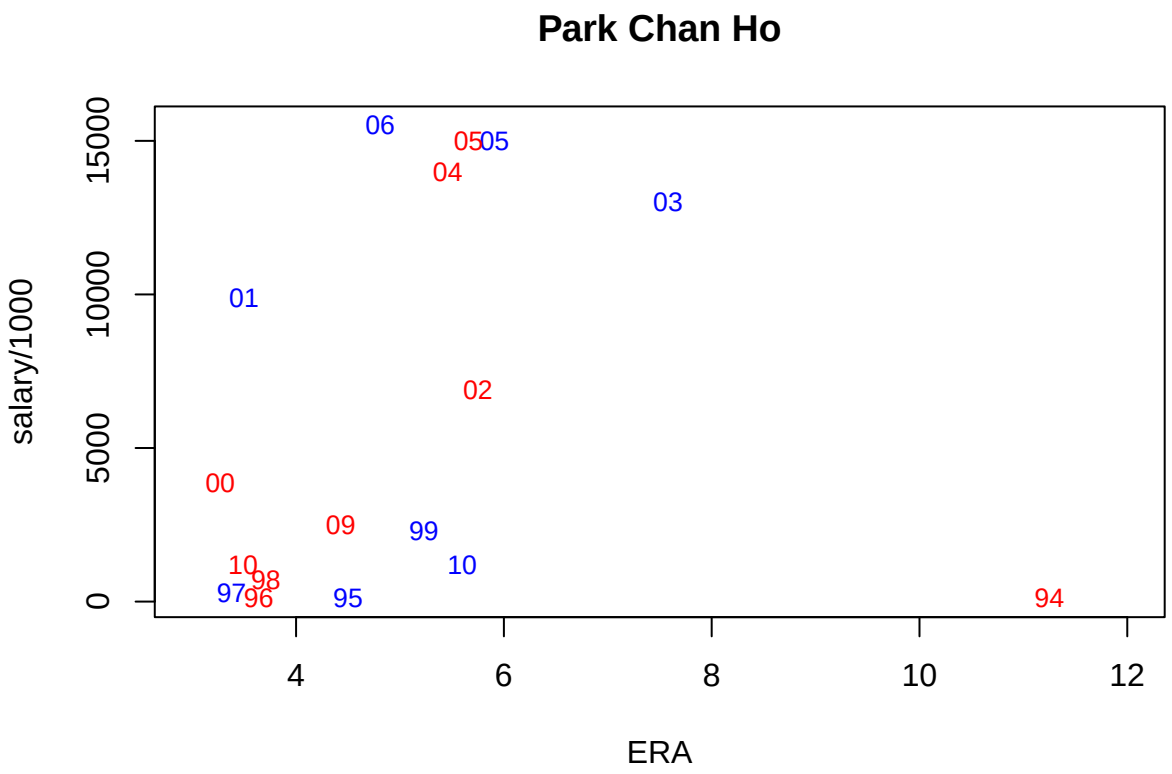
```
Park <- merge(Pitching.1, Salaries.1, by = "yearID")

Park[, c(1:5, 34)]
```

```
##    yearID playerID.x stint teamID.x lgID.x   salary
## 1    1994   parkch01     1      LAN     NL   109000
## 2    1995   parkch01     1      LAN     NL   114000
## 3    1996   parkch01     1      LAN     NL   124000
## 4    1997   parkch01     1      LAN     NL   270000
## 5    1998   parkch01     1      LAN     NL   700000
## 6    1999   parkch01     1      LAN     NL  2300000
## 7    2000   parkch01     1      LAN     NL  3850000
## 8    2001   parkch01     1      LAN     NL  9900000
## 9    2002   parkch01     1      TEX     AL  6884803
## 10   2003   parkch01     1      TEX     AL 13000000
## 11   2004   parkch01     1      TEX     AL 14000000
## 12   2005   parkch01     2      SDN     NL 15000000
## 13   2005   parkch01     1      TEX     AL 15000000
## 14   2006   parkch01     1      SDN     NL 15505142
## 15   2009   parkch01     1      PHI     NL  2500000
## 16   2010   parkch01     1      NYA     AL  1200000
## 17   2010   parkch01     2      PIT     NL  1200000
```

Plot

```
with(Park, plot(ERA, salary / 1000,
                xlim = c(3, 12),
                type = "n",
                main = "Park Chan Ho"))
with(Park, text(ERA, salary / 1000,
                substring(yearID, 3, 4),
                col = c("red", "blue"),
                cex = 0.8))
```


Park Chan Ho
salary/1000
0
5000
10000
15000
4
6
8
10
12
ERA
06
05
05
04
03
01
02
00
09
99
10
10
98
97
96
95
94