

# 問題解決技法入門

## 3. Data Analysis

## 2. Data Visualization using R

堀田 敬介

# R と R commander について

- R (アール)

- データ解析・統計処理ソフト
- The R Project for Statistical Computing
  - <https://www.r-project.org/>
    - R is a free software environment for statistical computing and graphics.
  - ※CUI (Character User Interface) を使用するため, 初心者・初級者には敷居が高い

- R commander (アール・コマンダー)

- 初心者・初級者でも R を使用し易くするための GUI (Graphical User Interface) パッケージ. R から呼び出して使う `library(Rcmdr)`
- The R Commander: A Basic-Statistics GUI for R
  - <https://socialsciences.mcmaster.ca/jfox/Misc/Rcmdr/>
    - The R Commander is a graphical user interface (GUI) to the free, open-source [R statistical software](#).

# Outline

## 1. データの準備

- ① データの準備 (csvファイル)
- ② R/R commander の起動
- ③ データの読み込み (csv-fileをR/R commanderで開く)
- ④ データの整備: ケース名の設定

## 2. R commander によるデータの視覚化

- ⑤ 箱ひげ図 box plot, box-and-whisker plot
- ⑥ 幹葉図 stem-and-leaf plot
- ⑦ 散布図 scatter plot
- ⑧ 散布図行列 scatter plot matrix

## 3. R によるデータの視覚化

- ⑨ 多次元尺度法 multi-dimensional scaling

# R commanderでデータの視覚化

## ① データの準備: csv ファイル

2024年プロ野球 セ・パ成績  
(出典: スポーツナビ - Yahoo! JAPAN)

bb2024utf8.csv

※文字コードがutf8  
でないと読みエラー  
が起きる  
(文字コードがsjisだと  
「不正なマルチバイト  
文字エラー」が出て  
読み込みに失敗する)

| チーム    | リーグ | 試合  | 勝利 | 敗戦 | 引分 | 勝率    | 得点  | 失点  | 本塁打 | 盗塁 | 打率    | 防御率   | 失策 |
|--------|-----|-----|----|----|----|-------|-----|-----|-----|----|-------|-------|----|
| 巨人     | セ   | 143 | 77 | 59 | 7  | 0.566 | 462 | 381 | 81  | 59 | 0.247 | 2.490 | 58 |
| 阪神     | セ   | 143 | 74 | 63 | 6  | 0.540 | 485 | 420 | 67  | 41 | 0.242 | 2.500 | 85 |
| DeNA   | セ   | 143 | 71 | 69 | 3  | 0.507 | 522 | 503 | 101 | 69 | 0.256 | 3.070 | 96 |
| 広島     | セ   | 143 | 68 | 70 | 5  | 0.493 | 415 | 419 | 52  | 66 | 0.238 | 2.620 | 66 |
| ヤクルト   | セ   | 143 | 62 | 77 | 4  | 0.446 | 506 | 556 | 103 | 67 | 0.243 | 3.640 | 69 |
| 中日     | セ   | 143 | 60 | 75 | 8  | 0.444 | 373 | 478 | 68  | 40 | 0.243 | 2.990 | 68 |
| ソフトバンク | パ   | 143 | 91 | 49 | 3  | 0.650 | 607 | 390 | 114 | 89 | 0.259 | 2.530 | 53 |
| 日本ハム   | パ   | 143 | 75 | 60 | 8  | 0.556 | 532 | 485 | 111 | 91 | 0.245 | 2.940 | 75 |
| ロッテ    | パ   | 143 | 71 | 66 | 6  | 0.518 | 493 | 495 | 75  | 64 | 0.248 | 3.170 | 71 |
| 楽天     | パ   | 143 | 67 | 72 | 4  | 0.482 | 492 | 579 | 72  | 90 | 0.242 | 3.730 | 64 |
| オリックス  | パ   | 143 | 63 | 77 | 3  | 0.450 | 402 | 448 | 71  | 61 | 0.238 | 2.820 | 78 |
| 西武     | パ   | 143 | 49 | 91 | 3  | 0.350 | 350 | 485 | 60  | 83 | 0.212 | 3.020 | 72 |

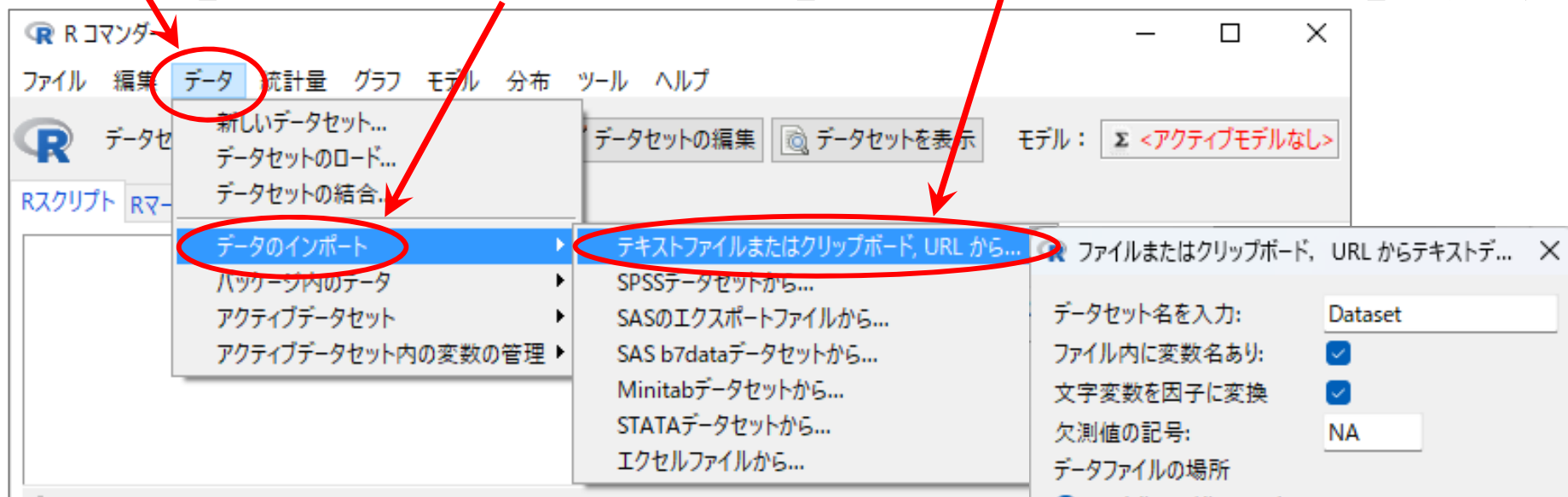
## ② Rの起動: 「R x64 X.X.X」を選択

- 注) x64 = 64bit用のプログラム(アプリ), X.X.X = Rのバージョン
- 注) 32bit PCの場合は, 「R i386 X.X.X」を選択
- 注) 起動すると「R Console」が開く. コマンドプロンプト(>)で  
「library(Rcmdr)」と打ち[Enter] → R commander が起動

# R commanderでデータの視覚化

## ③ データの読込

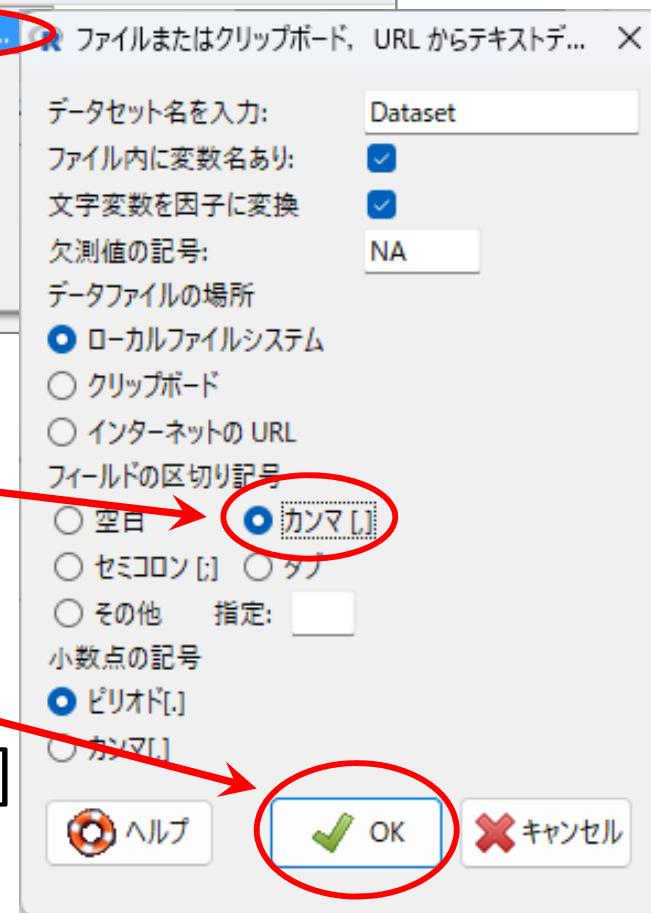
- 「データ」ー「データのインポート」ー「テキストファイル...」を選択



- [フィールドの区切り記号]で[カンマ[,] ]を選び, [OK]クリック

- ①で準備したファイル「\*.csv」を選び[開く]

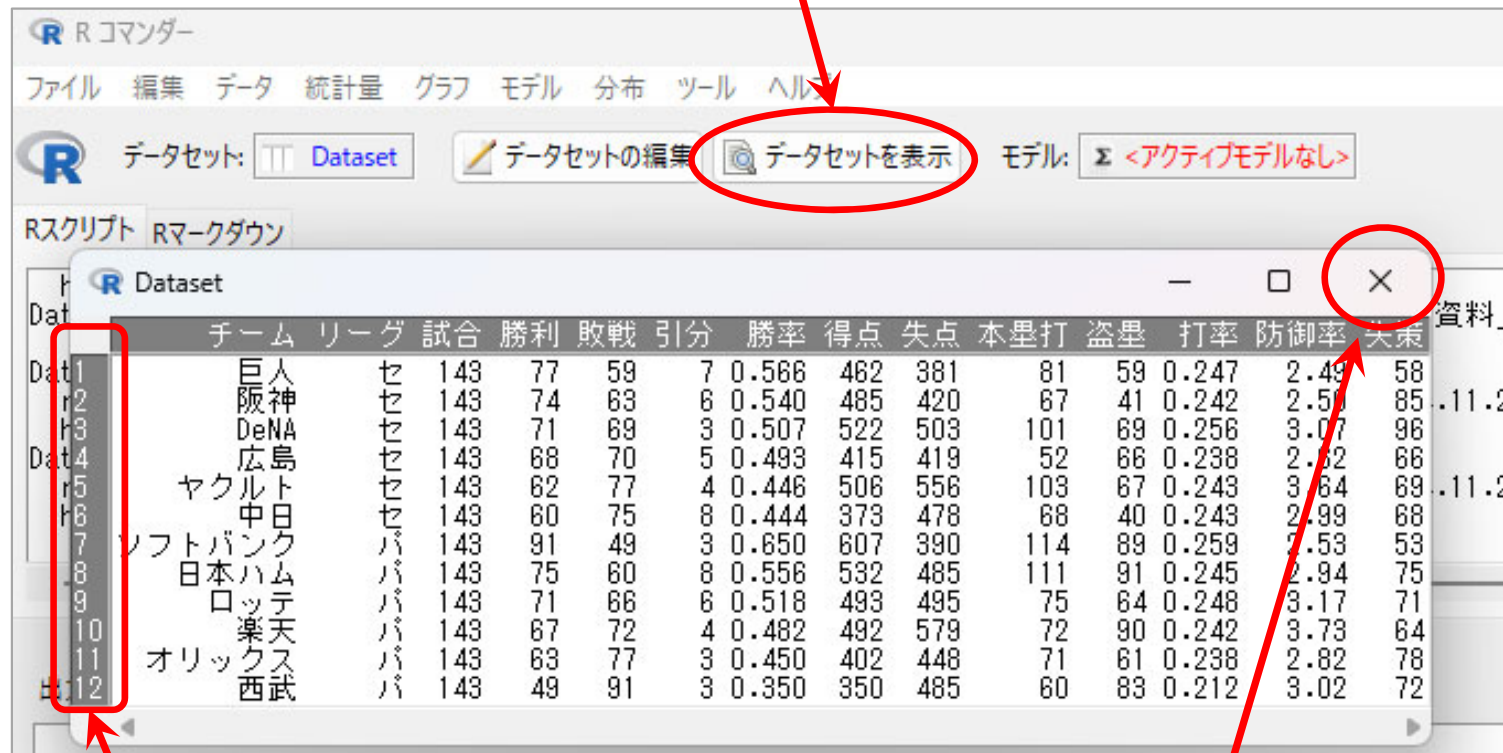
注)「データセットDatasetがすでに存在...上書き...」→[Yes]



# R commanderでデータの視覚化

## ③ データの読込（読み込んだファイルの確認）

- [データセットを表示]ボタンをクリックし、内容を確認

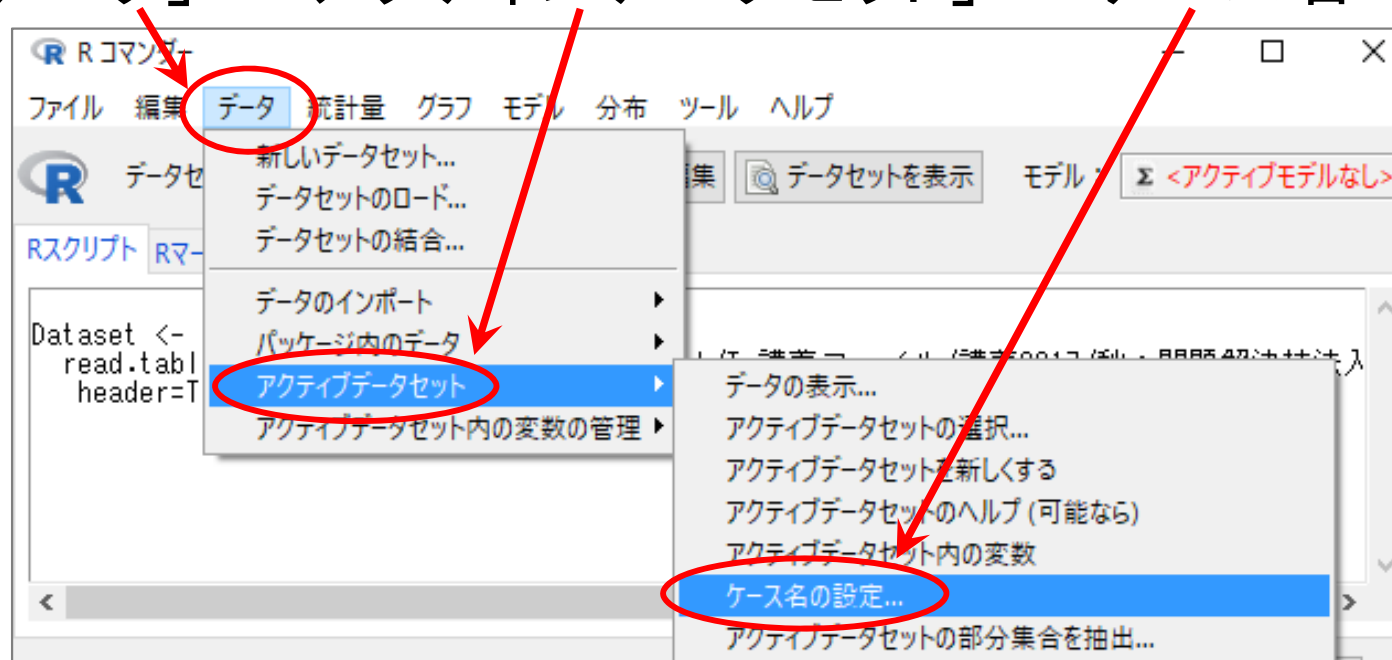


- ケース名（左端）が通し番号（1,2,...,12）であることを確認
- 確認できたら、「Dataset」の「×」をクリックして「閉じる」

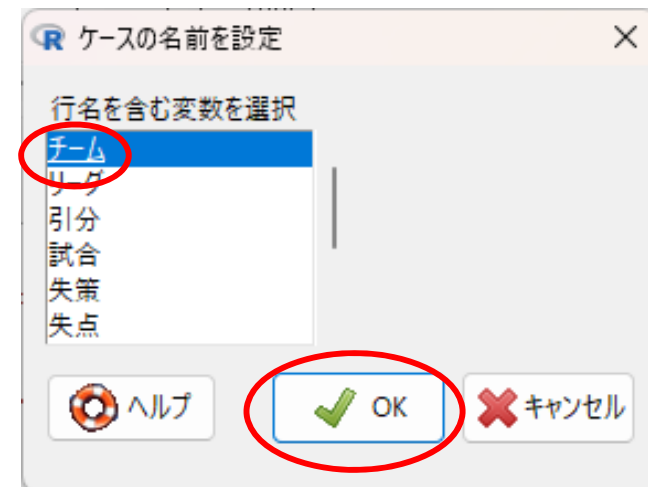
# R commanderでデータの視覚化

## ④ データにケース名を設定する

- 「データ」ー「アクティブデータセット」ー「ケース名の設定」選択



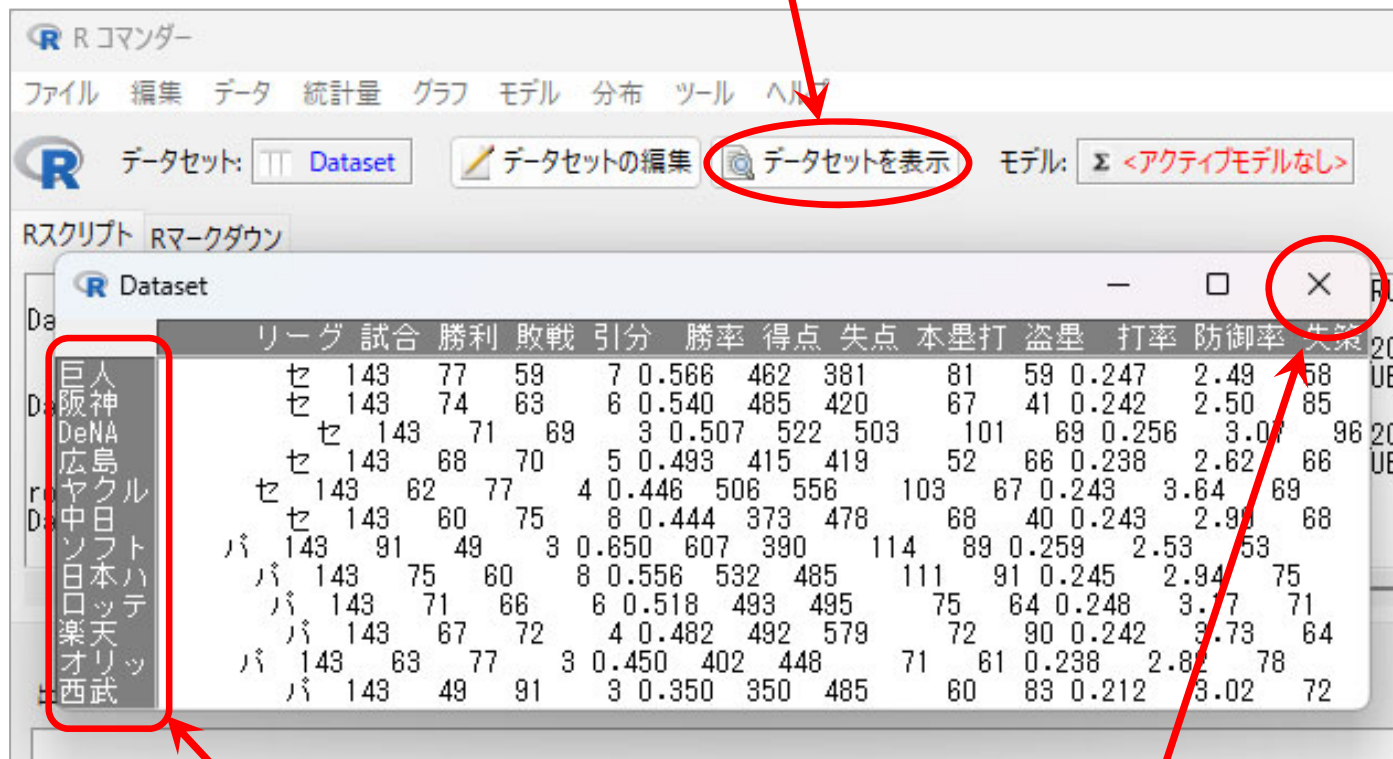
- [行名を含む変数を選択]で  
[チーム]を選び[OK]



# R commanderでデータの視覚化

## ④ データにケース名を設定する(設定確認)

- [データセットを表示]ボタンをクリックし内容を確認



- 指定した変数が**ケース名**になっていることを確認
- 確認できたら、「Dataset」の「×」をクリックして「**閉じる**」

# R commanderでデータの視覚化

## ⑤ 箱ひげ図を描く

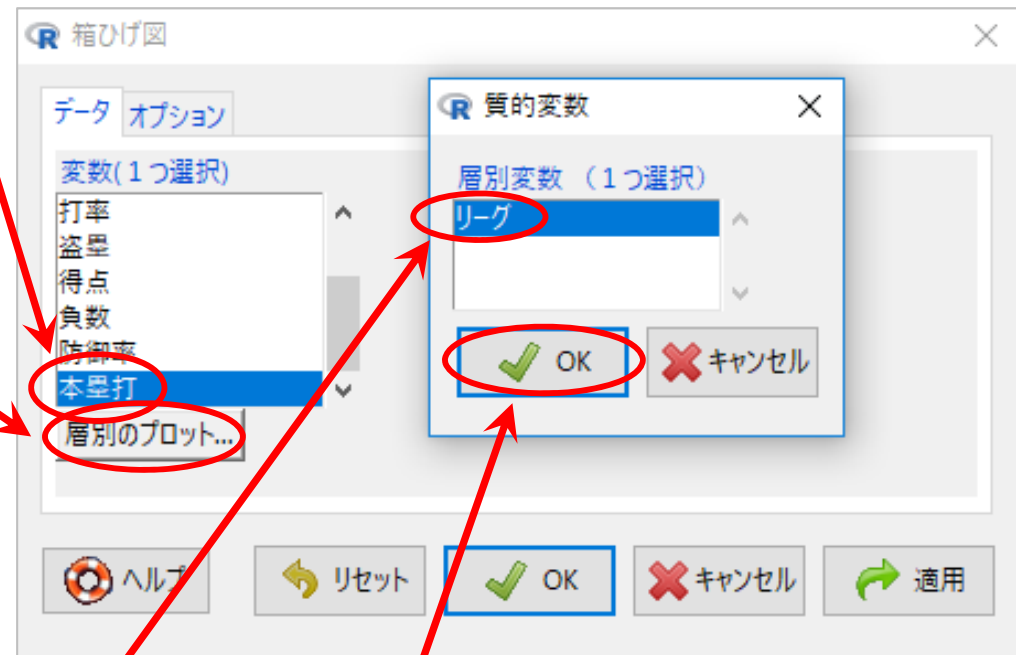
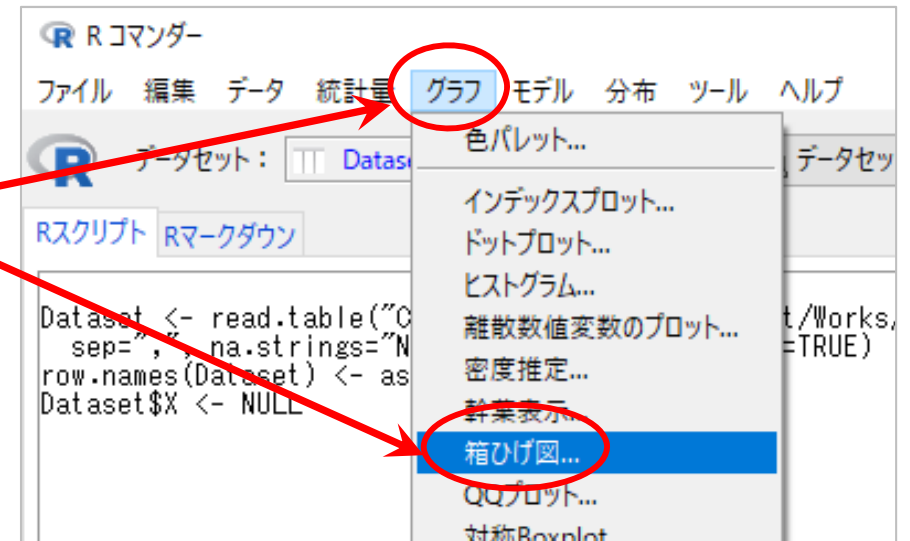
➤ 「グラフ」ー「箱ひげ図」を選択

➤ 『箱ひげ図』で以下を設定

➤ [変数(1つ選択)]で[本塁打]を選択

➤ [層別のプロット]クリック

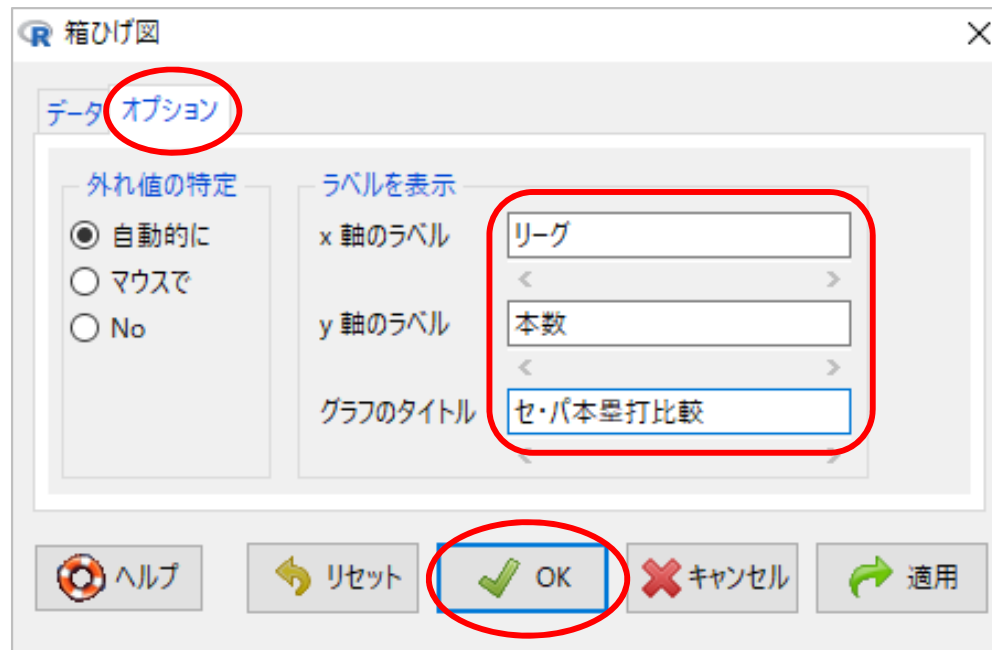
➤ [層別変数(1つ選択)]で[リーグ]を選択し[OK]



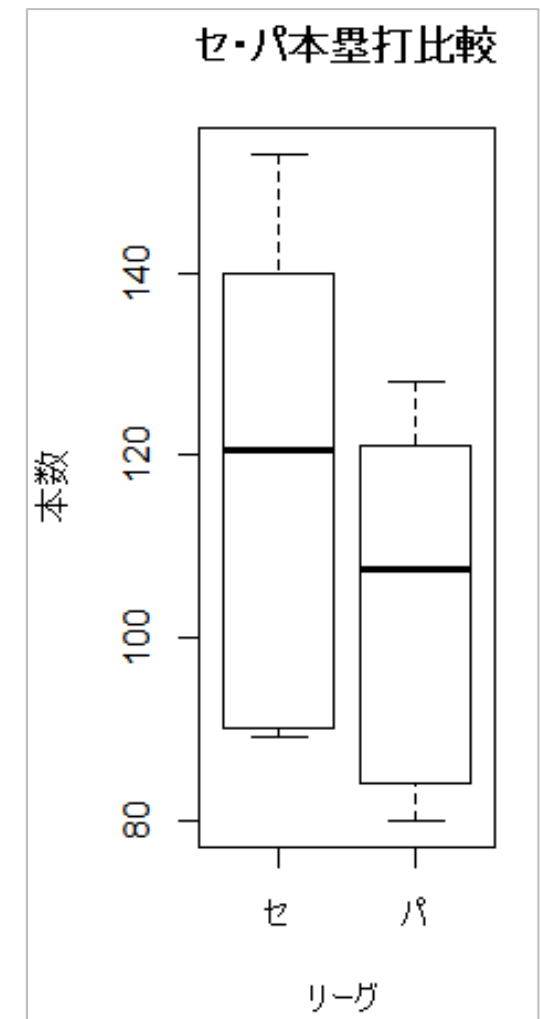
# R commanderでデータの視覚化

## ⑤ 箱ひげ図を描く【完成】

- 『箱ひげ図』の [オプション] タブで以下を設定
  - [ラベルを表示] に, ラベルをそれぞれ適切に設定
    - [X軸のラベル] = リーグ
    - [Y軸のラベル] = 本数
    - [グラフのタイトル] = セ・パ本塁打比較



- 全て設定後, [OK]



# R commanderでデータの視覚化

## ⑥ 幹葉図を描く

➤ 「グラフ」ー「幹葉表示」を選択

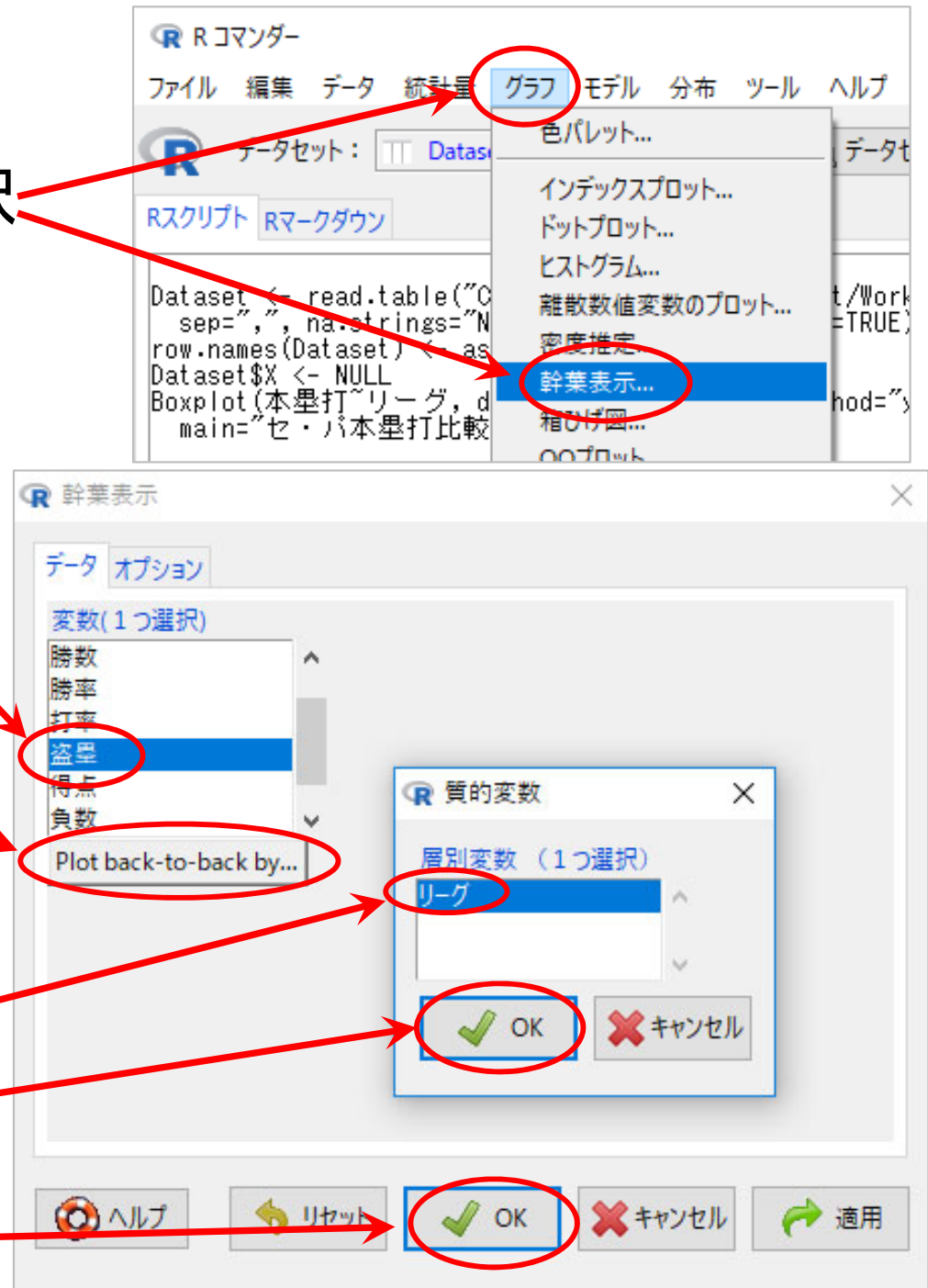
➤ 『幹葉表示』で以下を設定

➤ [変数(1つ選択)]で[盗塁]選択

➤ [Plot back-to-back by..]クリック

➤ [層別変数(1つ選択)]で  
[リーグ]を選択し[OK]

➤ 全て設定後, [OK]



# R commanderでデータの視覚化

## ⑥ 幹葉図を描く【完成】

R コマンダー

ファイル 編集 データ 統計量 グラフ モデル 分布 ツール ヘルプ

データセット: Dataset データセットの編集 データセットを表示 モデル: <アクティブモデルなし>

Rスクリプト Rマークダウン

```
Dataset <- read.table("C:/Users/bkh/Documents/Dat/Works/download/bb2016.csv", header=T,
  sep=";", na.strings="NA", dec=".", strip.white=TRUE)
row.names(Dataset) <- as.character(Dataset$X)
Dataset$X <- NULL
Boxplot(本塁打~リーグ, data=Dataset, id=list(method="y"), xlab="リーグ", ylab="本数",
  main="セ・パ本塁打比較")
library(tcltk, pos=16)
library(aplpack, pos=16)
with(Dataset, stem.leaf.backback(盗塁[リーグ == "セ"], 盗塁[リーグ == "パ"], na.rm=TRUE))
```

出力

> with(Dataset, stem.leaf.backback(盗塁[リーグ == "セ"], 盗塁[リーグ == "パ"], na.rm=TRUE))

1: represents 12, leaf unit: 1

盗塁[リーグ == "セ"]

盗塁[リーグ == "パ"]

|     |     |    |    |     |
|-----|-----|----|----|-----|
| 1   | 9   | 5  | 6  | 1   |
| (3) | 720 | 6  | 7  | 2   |
| 2   | 2   | 8  | 7  | (1) |
|     |     | 9  | 7  | (2) |
|     |     | 10 | 47 |     |
| 1   | 8   | 11 |    |     |
|     |     | 12 |    |     |
|     |     | 13 | 2  | 1   |
|     |     | 14 |    |     |
| n:  | 6   | 6  |    |     |

# R commanderでデータの視覚化

## ⑦ 散布図を描く

➤ 「グラフ」ー「散布図」を選択

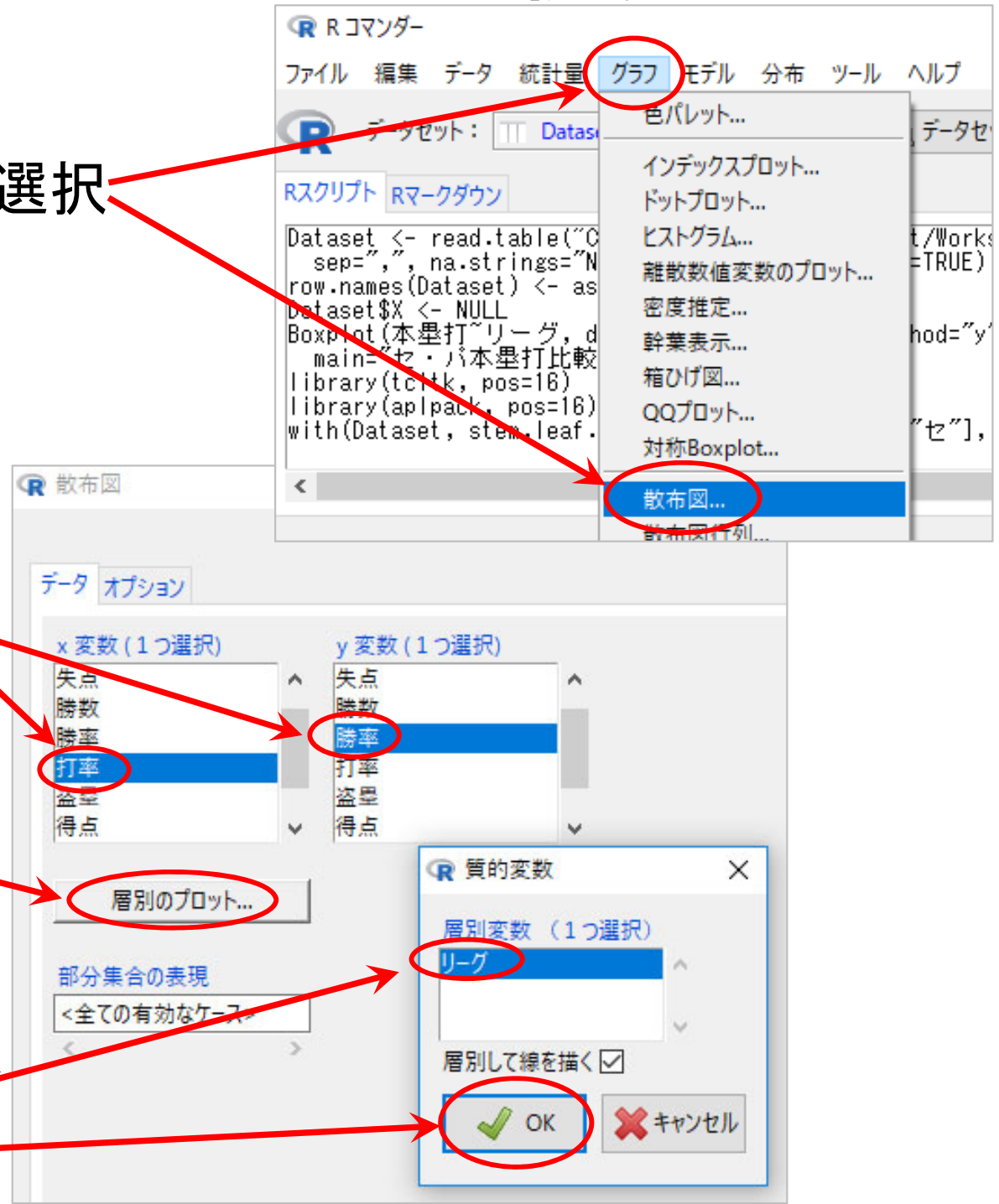
➤ 『散布図』で以下設定

➤ [x変数] = 打率

➤ [y変数] = 勝率

➤ [層別のプロット]クリック

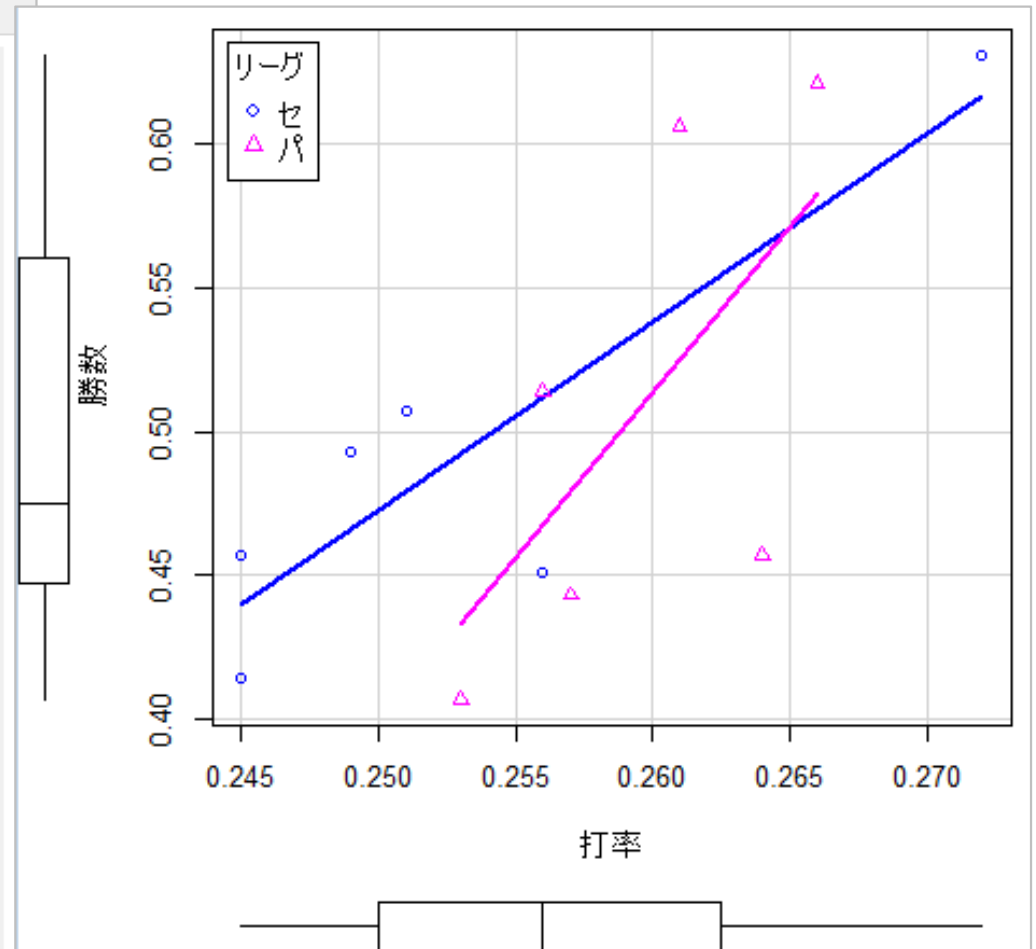
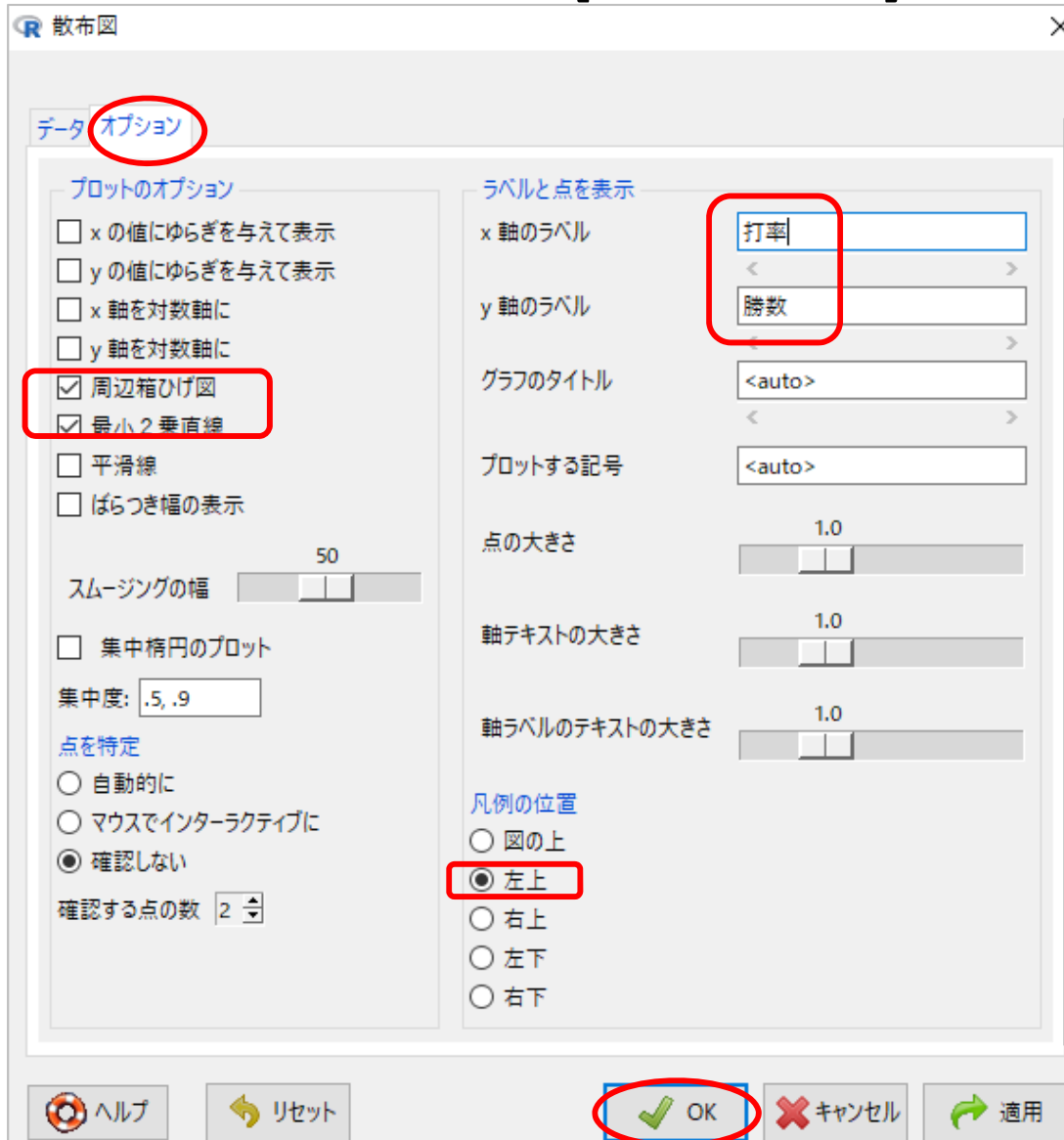
➤ [層別変数(1つ選択)]で  
[リーグ]を選択し [OK]



# R commanderでデータの視覚化

## ⑦ 散布図を描く【完成】

➤ 『散布図』の[オプション]タブで以下を設定後, [OK]



# R commanderで散布図行列描画

## ⑧ 散布図行列を描く

➤ 「グラフ」ー「散布図行列」を選択

➤ 『散布図行列』で以下を設定

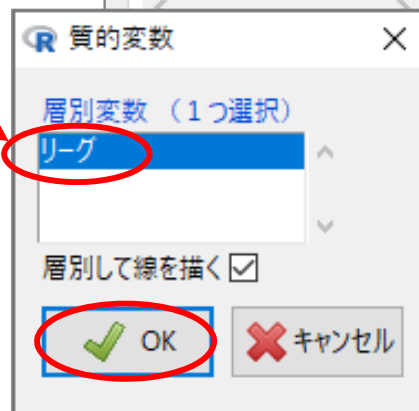
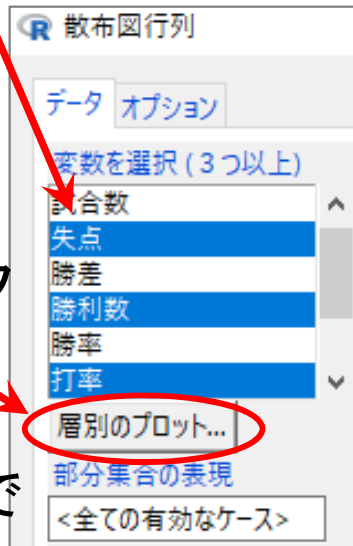
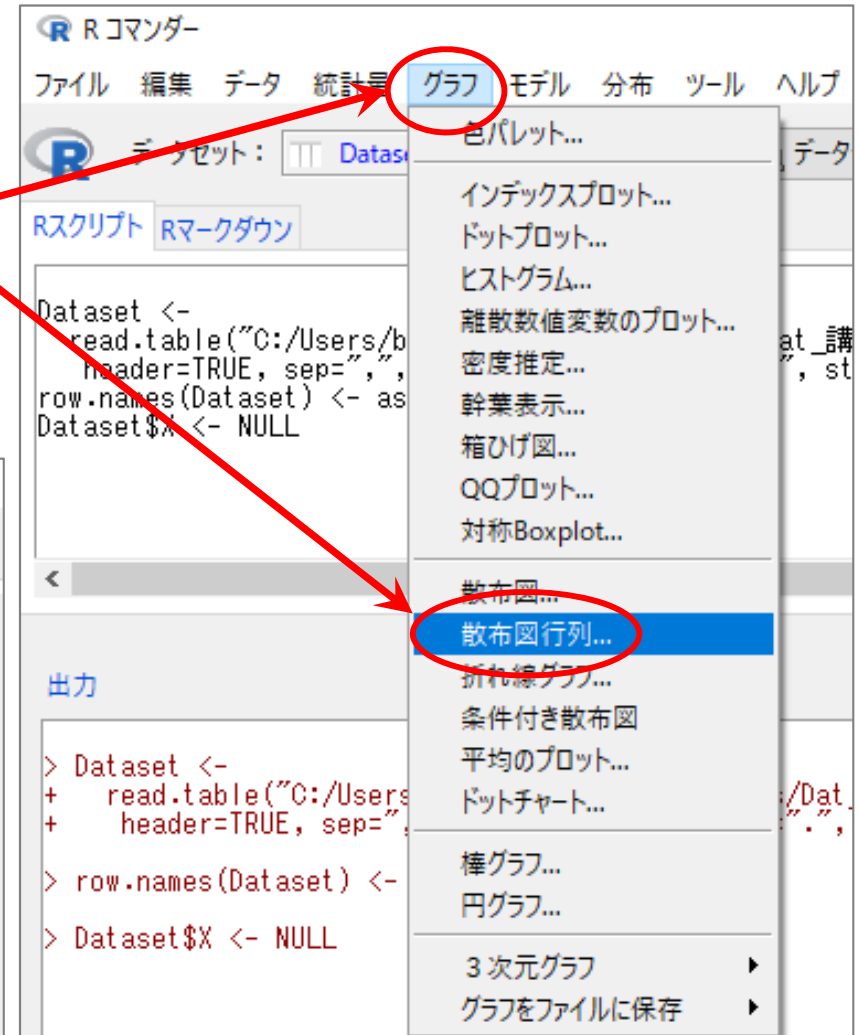
➤ [変数(3つ以上)]で[失点][勝利数][打率][本塁打] 4つを選択

(※複数選択は[Ctrl]キーを押しながら選択)

➤ [層別のプロット]クリック

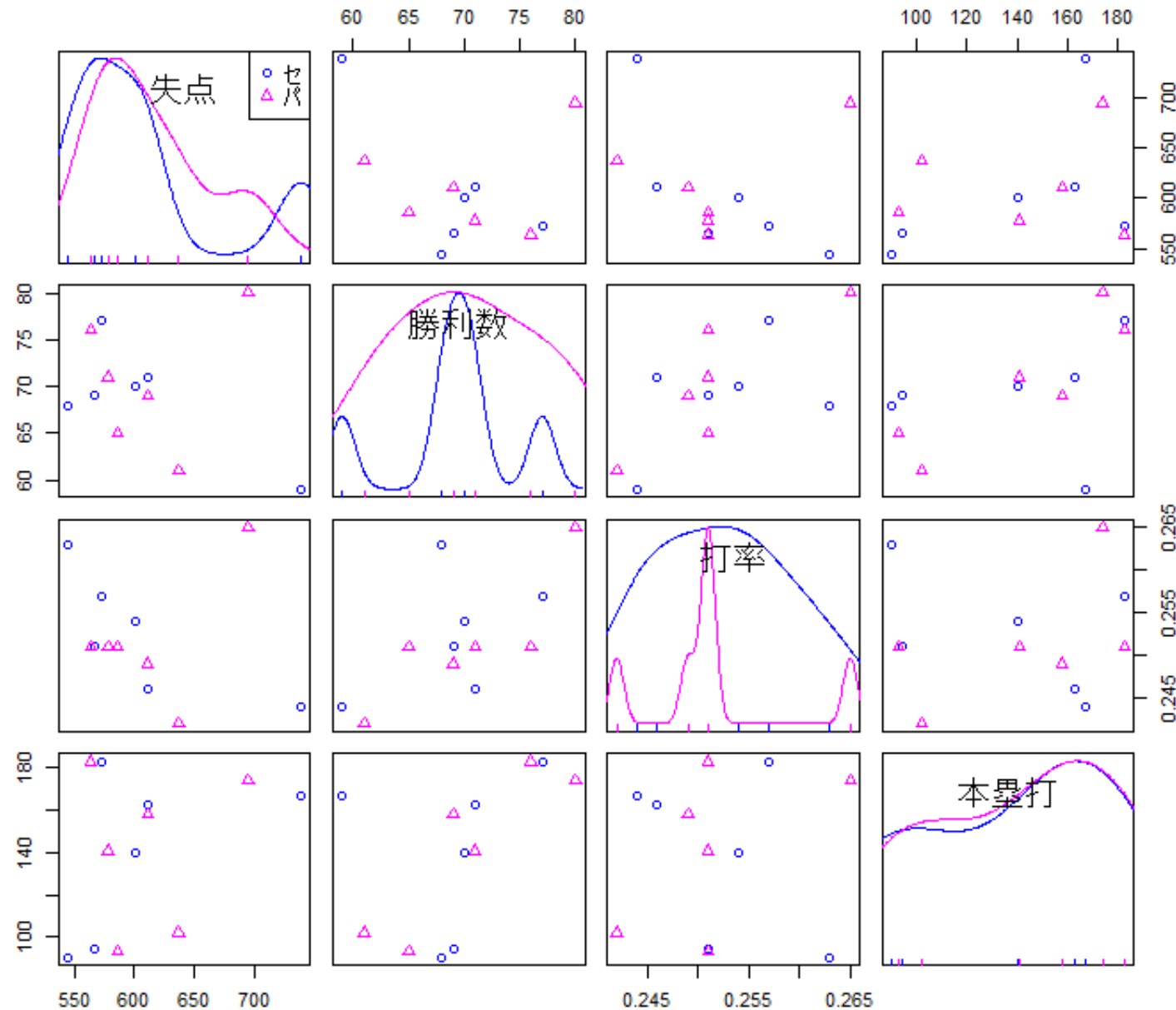
➤ [層別変数(1つ選択)]で[リーグ] 選択し [OK]

➤ [散布図行列]に戻ったら [OK]



# R commanderで散布図行列描画

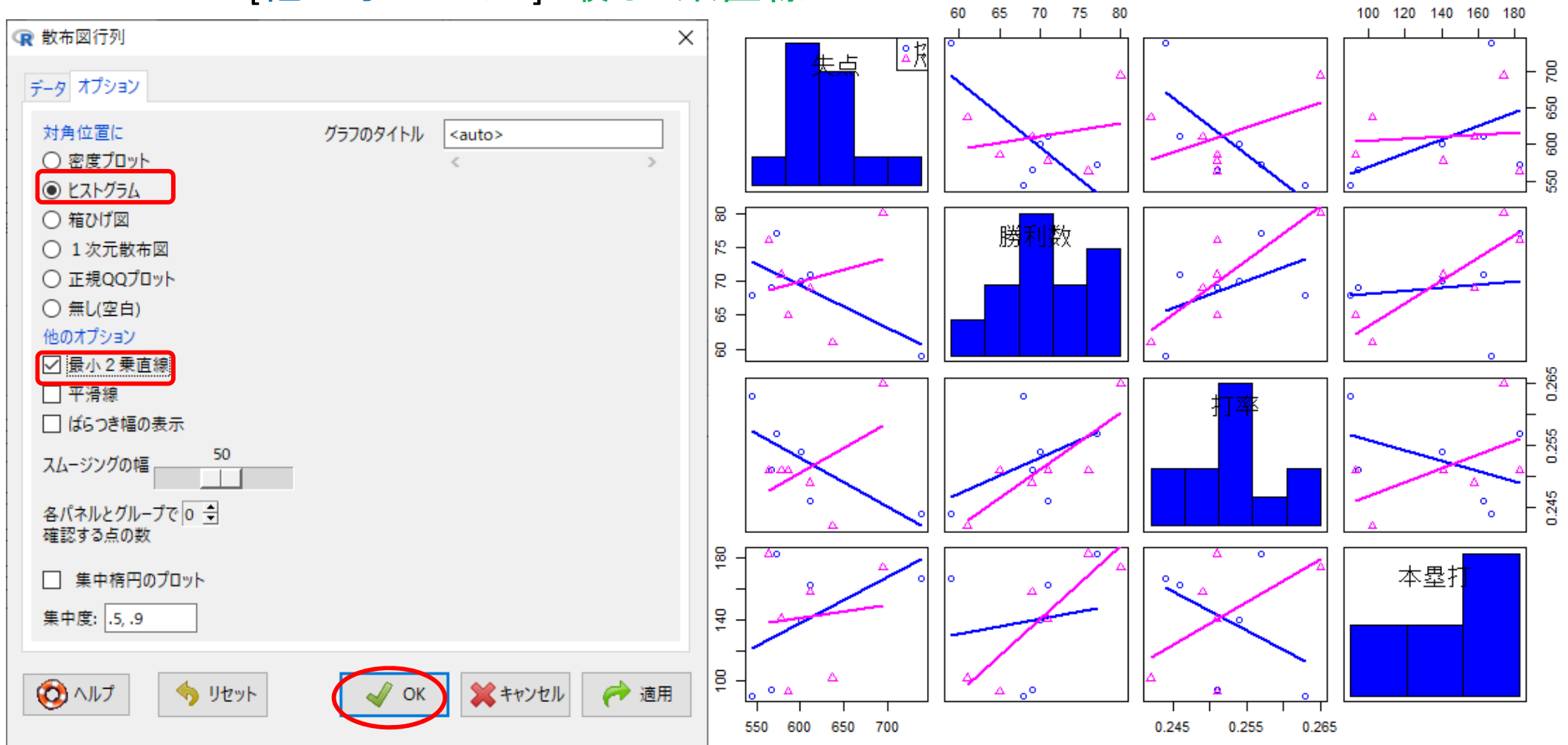
## ⑧ 散布図行列を描く1 【完成1】



# R commanderで散布図行列描画

## ⑧ 散布図行列を描く2 【完成2】

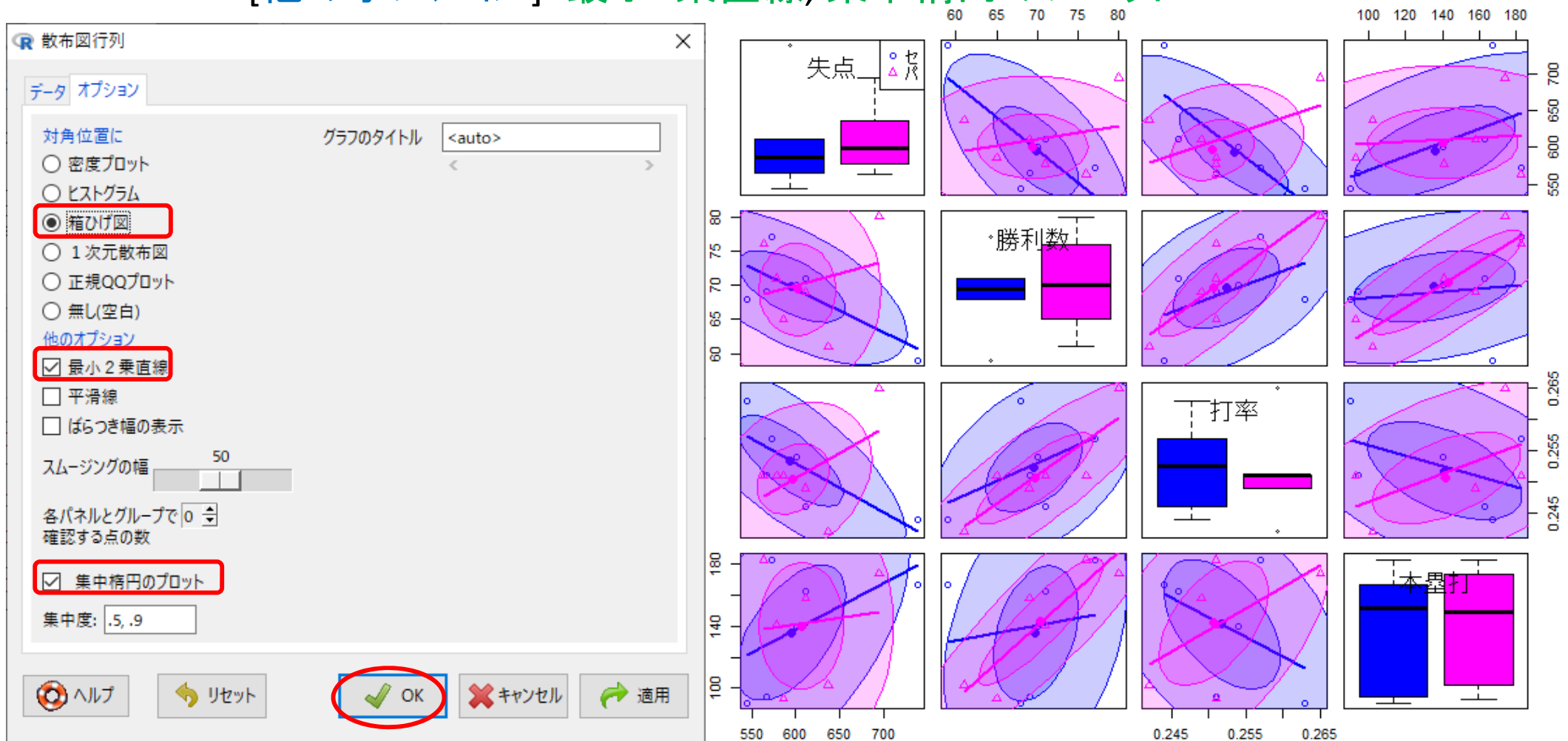
- 再度「グラフ」－「散布図行列」選択し, [オプション]タブで設定
  - [対角位置に]=ヒストグラム
  - [他のオプション]=最小2乗直線



# R commanderで散布図行列描画

## ⑧ 散布図行列を描く3 【完成3】

- 再度「グラフ」－「散布図行列」選択し, [オプション]タブで設定
  - [対角位置に]=箱ひげ図
  - [他のオプション]=最小2乗直線, 集中楕円のプロット



# Rでデータの視覚化

- 多次元尺度法 multi-dimensional scaling
  - 類似度データから類似性が高いものの同士を近くに、低いものの同士を遠くに配置して描画する手法の1つ
- 描画用データファイルの準備
  - 類似度を表した行列形式のデータを **csv ファイル**にし、「マインドキュメント("K:/")」に保存

「yamaote.csv」... JR山手線30駅の駅間所要時間(分)データ

|       | 東京駅 | 神田駅 | 秋葉原駅 | 御徒町駅 | 上野駅 | 鶯谷駅 | 日暮里駅 | 西日暮里駅 | 田端駅 | 駒込駅 | 巣鴨駅 | 大塚駅 | 池袋駅 |
|-------|-----|-----|------|------|-----|-----|------|-------|-----|-----|-----|-----|-----|
| 東京駅   | 0   | 1   | 3    | 5    | 7   | 9   | 11   | 12    | 14  | 16  | 18  | 20  | 22  |
| 神田駅   | 1   | 0   | 2    | 4    | 6   | 8   | 10   | 11    | 13  | 15  | 17  | 19  | 21  |
| 秋葉原駅  | 3   | 2   | 0    | 2    | 4   | 6   | 8    | 9     | 11  | 13  | 15  | 17  | 19  |
| 御徒町駅  | 5   | 4   | 2    | 0    | 2   | 4   | 6    | 7     | 9   | 11  | 13  | 15  | 17  |
| 上野駅   | 7   | 6   | 4    | 2    | 0   | 2   | 4    | 5     | 7   | 9   | 11  | 13  | 15  |
| 鶯谷駅   | 9   | 8   | 6    | 4    | 2   | 0   | 2    | 3     | 5   | 7   | 9   | 11  | 13  |
| 日暮里駅  | 11  | 10  | 8    | 6    | 4   | 2   | 0    | 1     | 3   | 5   | 7   | 9   | 11  |
| 西日暮里駅 | 12  | 11  | 9    | 7    | 5   | 3   | 1    | 0     | 2   | 4   | 6   | 8   | 10  |
| 田端駅   | 14  | 13  | 11   | 9    | 7   | 5   | 3    | 2     | 0   | 2   | 4   | 6   | 8   |
| 駒込駅   | 16  | 15  | 13   | 11   | 9   | 7   | 5    | 4     | 2   | 0   | 2   | 4   | 6   |
| 巣鴨駅   | 18  | 17  | 15   | 13   | 11  | 9   | 7    | 6     | 4   | 2   | 0   | 2   | 4   |
| 大塚駅   | 20  | 19  | 17   | 15   | 13  | 11  | 9    | 8     | 6   | 4   | 2   | 0   | 2   |
| 池袋駅   | 22  | 21  | 19   | 17   | 15  | 13  | 11   | 10    | 8   | 6   | 4   | 2   | 0   |

...

# Rでデータの視覚化

- 多次元尺度法 multi-dimensional scaling で描画
  - 作業フォルダの設定(マイドキュメント("K:/")へ移動)

```
> setwd("K:/")
```

- 作業フォルダの確認/位置取得 get working directory

```
> getwd()
```

- 保存してある csvファイル(yamanote.csv)の読み込み

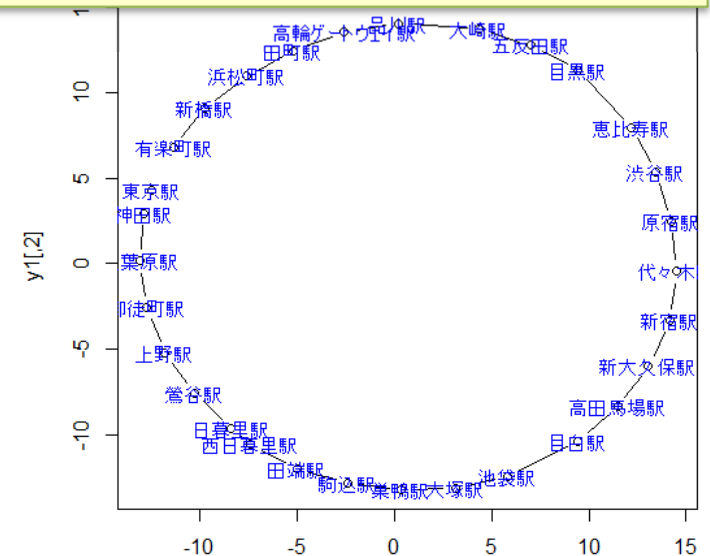
```
> y0 <- read.csv("K:/yamanote.csv", header=T, row.names=1)
```

- (古典的) 多次元尺度法で計算

```
> y1 <- cmdscale(y0)
```

- 描画

```
> plot(y1, type="b")  
> text(y1, names(y0), col="blue")
```



# 参考文献

- ◆ 山本他『Rで学ぶデータサイエンス12統計データの視覚化』共立出版(2013)
- ◆ 奥村晴彦『Rで楽しむ統計』共立出版(2016)
- ◆ J. P. Lander『みんなのR』マイナビ(2015)
- ◆ W. Chang『Rグラフィックス クックブック』オライリー(2013)
- ◆ 青木繁伸『Rによる統計解析』オーム社(2009)
- ◆ 荒木孝治『RとRコマンダーではじめる多変量解析』日科技連(2007)
- ◆ 金明哲『Rによるデータサイエンス』森北出版(2007)
- ◆ 新納浩幸『Rで学ぶクラスタ解析』オーム社(2007)

## もっと知りたい人へ

- 関連する経営学科の授業
  - 「**基礎統計**」(1/2セメ)
  - 「**基礎統計演習**」(3/4セメ)
  - 「**データ処理応用**」(2/3セメ)
  - 「**統計モデル分析**」(5セメ)
  - 「**ビッグデータ・AI演習**」(6セメ)

# Rでデータの視覚化

- csv ファイルをデータとして利用
  - 「マイドキュメント(Y:)」に「R」フォルダをつくり中に保存

bb2018.csv

※) 2018年プロ野球  
セ・パ成績  
(Yahoo Japan!  
Sports naviより)

|         | リーグ | 試合数 | 勝利数 | 敗戦数 | 引分数 | 勝率    | 得点  | 失点  | 本塁打 | 盗塁  | 打率    | 防御率  |
|---------|-----|-----|-----|-----|-----|-------|-----|-----|-----|-----|-------|------|
| 広島      | セ   | 143 | 82  | 59  | 2   | 0.582 | 721 | 651 | 175 | 95  | 0.262 | 4.12 |
| ヤクルト    | セ   | 143 | 75  | 66  | 2   | 0.532 | 658 | 665 | 135 | 68  | 0.266 | 4.13 |
| 巨人      | セ   | 143 | 67  | 71  | 5   | 0.486 | 625 | 575 | 152 | 61  | 0.257 | 3.79 |
| D e N A | セ   | 143 | 67  | 74  | 2   | 0.475 | 572 | 642 | 181 | 71  | 0.25  | 4.18 |
| 中日      | セ   | 143 | 63  | 78  | 2   | 0.447 | 598 | 654 | 97  | 61  | 0.265 | 4.36 |
| 阪神      | セ   | 143 | 62  | 79  | 2   | 0.44  | 577 | 628 | 85  | 77  | 0.253 | 4.03 |
| 西武      | パ   | 143 | 88  | 53  | 2   | 0.624 | 792 | 653 | 196 | 132 | 0.273 | 4.24 |
| ソフトバンク  | パ   | 143 | 82  | 60  | 1   | 0.577 | 685 | 579 | 202 | 80  | 0.266 | 3.9  |
| 日本ハム    | パ   | 143 | 74  | 66  | 3   | 0.529 | 589 | 586 | 140 | 98  | 0.251 | 3.77 |
| オリックス   | パ   | 143 | 65  | 73  | 5   | 0.471 | 538 | 565 | 108 | 97  | 0.244 | 3.69 |
| ロッテ     | パ   | 143 | 59  | 81  | 3   | 0.421 | 534 | 628 | 78  | 124 | 0.247 | 4.04 |
| 楽天      | パ   | 143 | 58  | 82  | 3   | 0.414 | 520 | 583 | 132 | 69  | 0.241 | 3.78 |

- ファイルの読み込み

※1行目にheaderあり

※各行の名称は列1に

```
> dfbb <- read.csv("Y:/R/bb2018.csv", header=T, row.names=1)
```

※ファイルのフルパス

例) YドライブのRフォルダ内にあるbb2018.csvという名前のファイル

# Rでデータの視覚化

- 読込データの確認

- dfbbに代入したdata frame の中身を**全て**表示

```
> dfbb
```

- dfbbに代入したdata frame の中身を**一部(先頭)**表示

```
> head(dfbb)
```

- dfbbに代入したdata frame の中身を**一部(後尾)**表示

```
> tail(dfbb)
```

- dfbbの**項目名**表示 (header=Tで読んだデータ)

```
> names(dfbb)
```

- dfbbの**レコード名**表示 (row.names=1で指定した)

```
> row.names(dfbb)
```

# Rでデータの視覚化

- 箱ひげ図を描画

※dfbb\$本塁打 ... data.frameである dfbbの項目"本塁打"を箱ひげ図のデータとして使用

```
> boxplot(dfbb$本塁打)
```

... ①

- オプションを指定し箱ひげ図を描画

```
> boxplot(dfbb$本塁打, col="tomato", xlab="本塁打", ylab="本数",  
main="12チーム本塁打数の箱ひげ図")
```

... ②

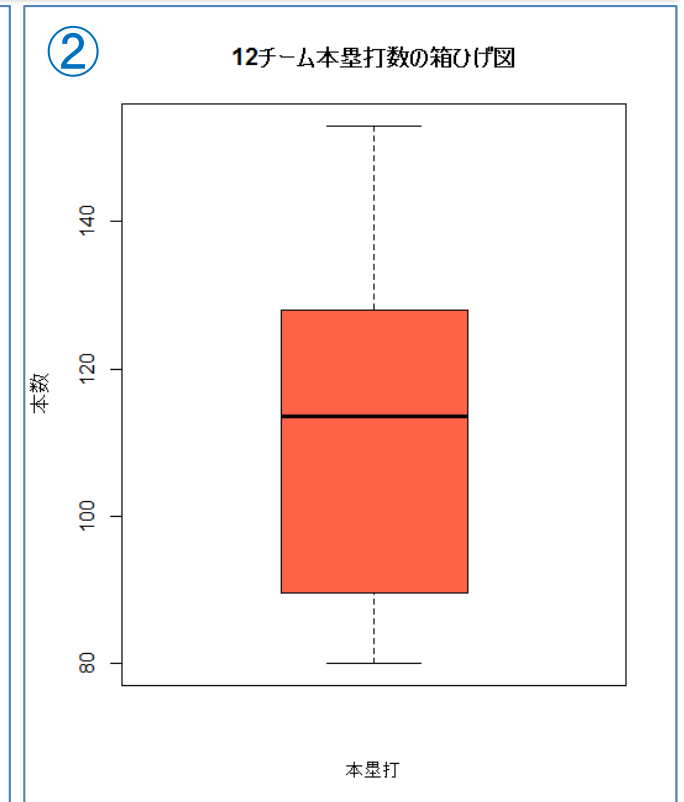
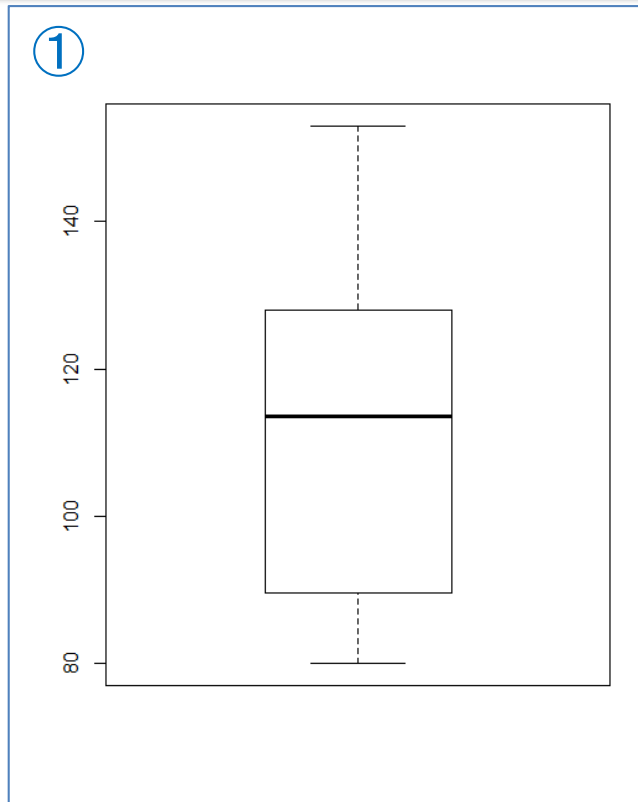
## <オプション>

col ... 色の指定(colour)

xlab ... x軸のラベル(label)

ylab ... y軸のラベル(label)

main ... タイトル



# Rでデータの視覚化

- グループ毎に箱ひげ図を描画

```
> boxplot(dfbb$本塁打~dfbb$リーグ)
```

※項目「リーグ(セ・パ)」毎に描画するよう指定  
(チルド記号 tilde(~)で層別にしたい項目を指定)

## <オプション>

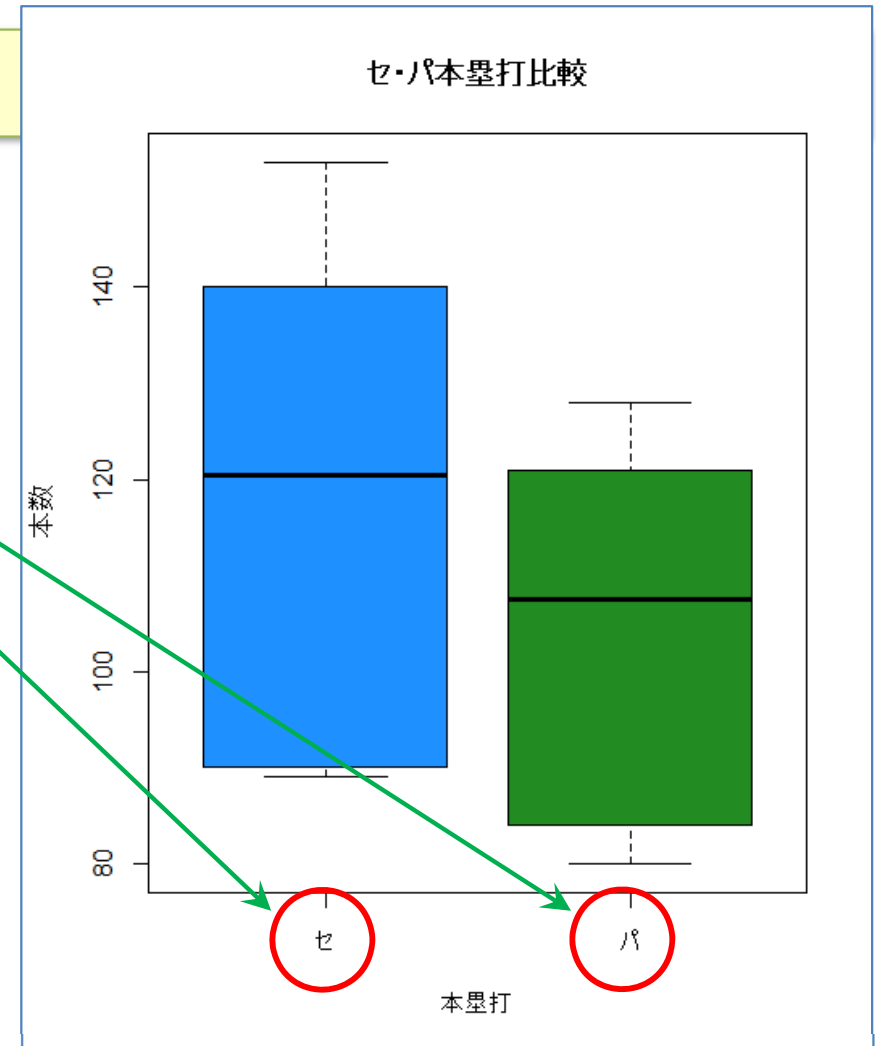
col ... 色の指定(colour)

c("blue", "red", "green") ... 色名のベクトルをつくる

xlab ... x軸のラベル(label)

ylab ... y軸のラベル(label)

main ... タイトル



```
> boxplot(dfbb$本塁打~dfbb$リーグ, xlab="本塁打", ylab="本数",  
col=c("dodgerblue","forestgreen"), main="セ・パ本塁打比較")
```

# Rでデータの視覚化

- 幹葉図 (stem-and-leaf plot) を描画

```
> stem(dfbb$本塁打)
```

The decimal point is 1 digit(s) to the right of the |

8 | 0490

10 | 134

12 | 188

14 | 03

- 幹葉図を描画 (オプション scale=2)

```
> stem(dfbb$本塁打, 2)
```

The decimal point is 1 digit(s) to the right of the |

8 | 049

9 | 0

10 | 1

11 | 34

12 | 188

13 |

14 | 0

15 | 3

※scale数を大きくするとより詳細な幹葉図に  
(default=1)

# Rでデータの視覚化

- csv ファイルをデータとして利用
  - 「マイドキュメント(Y:)」に「R」フォルダをつくり中に保存

bi2016.csv

| 氏名    | チーム | リーグ | 打率     | 試合数 | 打席数 | 打数  | 安打  | 二塁打 | 三塁打 | 本塁打 | 塁打数 | 打点  | 得点  | 三振  | 四球 | 死球 | 犠打 | 犠飛 | 盗塁 | 出塁率   | 長打率   | 得点圏   | 併殺 | 失策 |
|-------|-----|-----|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|-------|-------|-------|----|----|
| 坂本 勇人 | 巨   | セ   | 0.344  | 137 | 576 | 488 | 168 | 28  | 3   | 23  | 271 | 75  | 96  | 67  | 81 | 0  | 1  | 6  | 13 | 0.433 | 0.555 | 0.339 | 6  | 16 |
| 鈴木 誠也 | 広   | セ   | 0.335  | 129 | 528 | 466 | 156 | 26  | 8   | 29  | 285 | 95  | 76  | 79  | 53 | 3  | 3  | 3  | 16 | 0.404 | 0.612 | 0.346 | 10 | 2  |
| 筒香 嘉智 | D   | セ   | 0.322  | 133 | 561 | 469 | 151 | 28  | 4   | 44  | 319 | 110 | 89  | 105 | 87 | 3  | 0  | 2  | 0  | 0.43  | 0.68  | 0.393 | 6  | 2  |
| 菊池 涼介 | 広   | セ   | 0.315  | 141 | 640 | 574 | 181 | 22  | 3   | 13  | 248 | 56  | 92  | 106 | 40 | 0  | 23 | 3  | 13 | 0.358 | 0.432 | 0.343 | 3  | 4  |
| 福留 孝介 | 神   | セ   | 0.311  | 131 | 523 | 453 | 141 | 25  | 3   | 11  | 205 | 59  | 52  | 78  | 61 | 3  | 0  | 6  | 0  | 0.392 | 0.453 | 0.31  | 6  | 1  |
| 山田 哲人 | ヤ   | セ   | 0.304  | 133 | 590 | 481 | 146 | 26  | 3   | 38  | 292 | 102 | 102 | 101 | 97 | 8  | 0  | 4  | 30 | 0.425 | 0.607 | 0.299 | 16 | 5  |
| 村田 修一 | 巨   | セ   | 0.3024 | 143 | 576 | 529 | 160 | 32  | 0   | 25  | 267 | 81  | 58  | 83  | 38 | 5  | 2  | 2  | 1  | 0.354 | 0.505 | 0.305 | 21 | 15 |
| 川端 慎吾 | ヤ   | セ   | 0.3023 | 103 | 458 | 420 | 127 | 22  | 1   | 1   | 154 | 32  | 48  | 31  | 34 | 1  | 1  | 2  | 3  | 0.354 | 0.367 | 0.301 | 13 | 5  |
| 新井 貴浩 | 広   | セ   | 0.3    | 132 | 513 | 454 | 136 | 23  | 2   | 19  | 220 | 101 | 66  | 101 | 54 | 1  | 0  | 4  | 0  | 0.372 | 0.485 | 0.323 | 12 | 5  |

※)2016年プロ野球個人成績(Yahoo Japan! Sports naviより)

- ファイル読込み

```
> dfbi <- read.csv("Y:/R/bi2016.csv", header=T, row.names=1)
```

## 【演習】

箱ひげ図で表示したい項目を1つ選び(例:打率, 安打, 本塁打, 打点, 得点, etc.), 12チーム毎の箱ひげ図を描画せよ.

さらに, 可能なら, 色, x軸ラベル, y軸ラベル, タイトルを適切に設定してみよう

# Rでデータの視覚化

その他のグラフ作成例

棒グラフ

散布図

※これらのグラフを作成したい時は、Excelを使った方が良い

# Rでデータの視覚化

- 棒グラフを作成

※色指定用のベクトル生成. "royalblue"を6回 repeat し,  
"violetred"を6回repeatしたベクトルをつくり cc に代入

```
> cc <- c(rep("royalblue",6), rep("violetred",6))  
> barplot(dfbb$勝数, names.arg=row.names(dfbb), col=cc, xlab="チーム名", ylab="勝数")
```

dfbb\$勝数 ... data.frameである dfbb の項目"勝数"を棒グラフのデータとして使用

names.arg ... それぞれの棒に対応する名称

col ... 棒の色指定

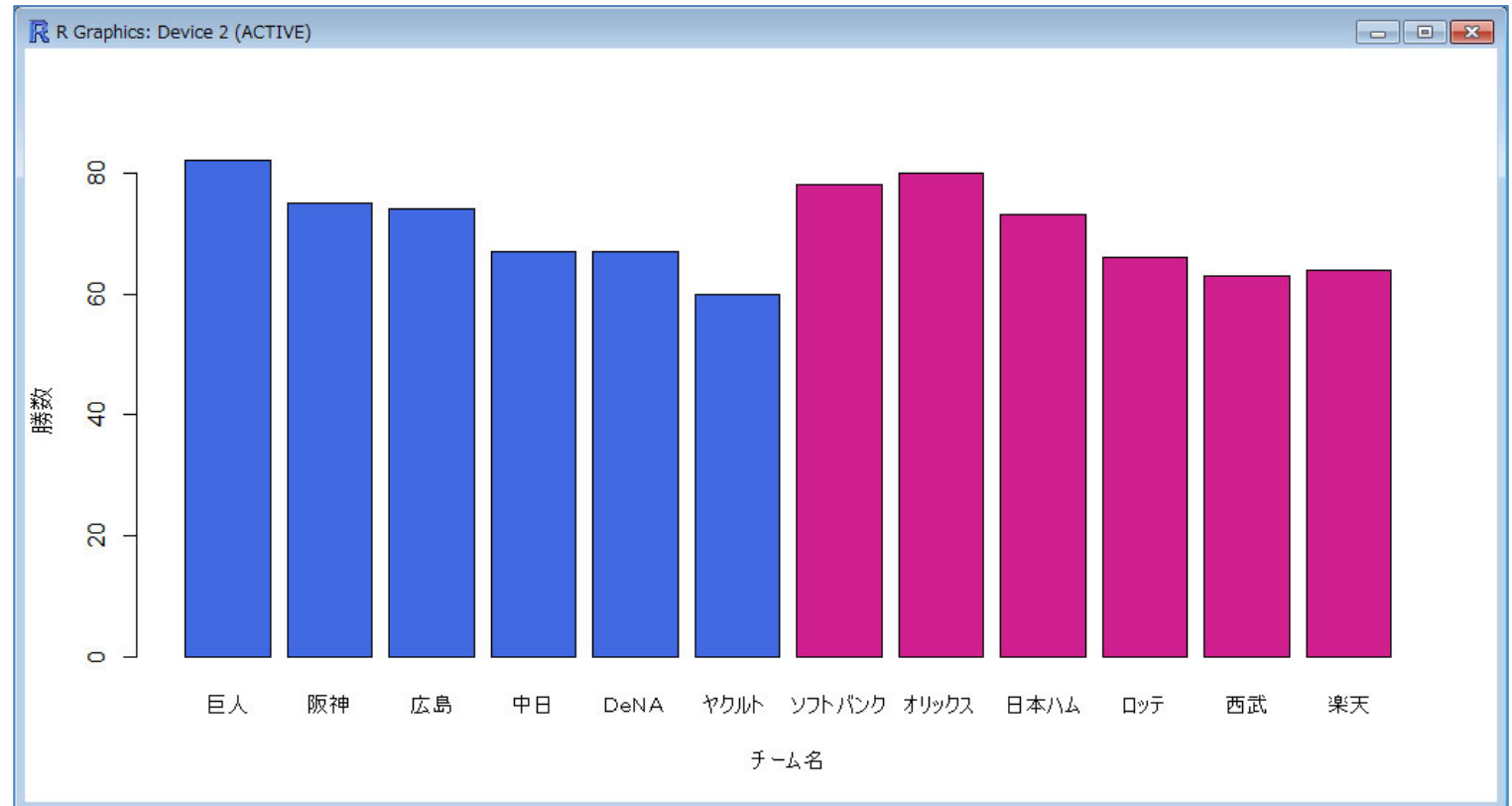
xlab ... x軸のラベル

ylab ... y軸のラベル

- Tips !

```
> colors()
```

※Rで使える657色の名称リスト表示



# Rでデータの視覚化

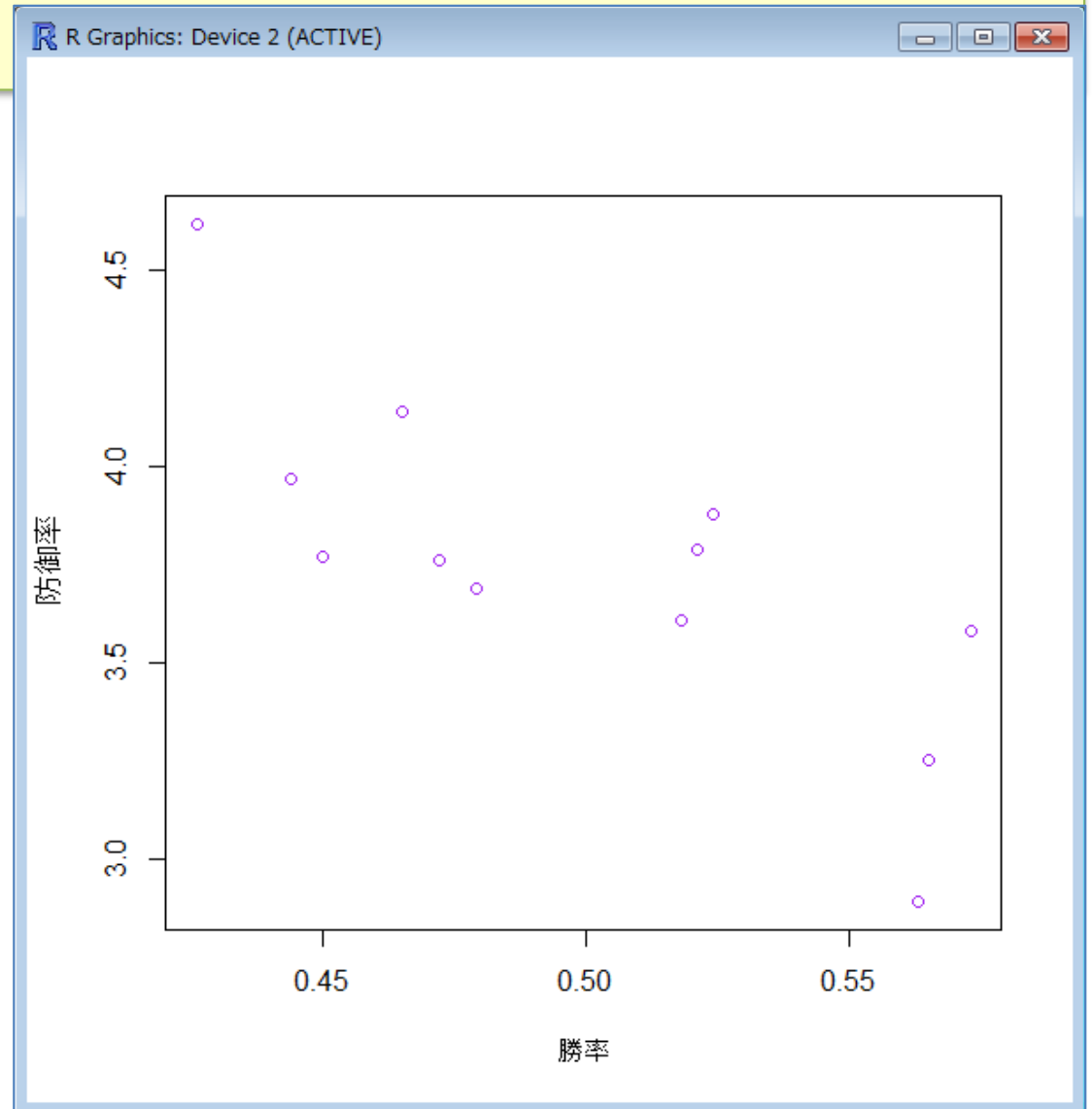
- 散布図を作成(1)

```
> plot(dfbb$勝率, dfbb$防御率, xlab="勝率", ylab="防御率",  
col="purple")
```

x軸を dfbb\$勝率  
y軸を dfbb\$防御率  
のデータを用い散布図を作成

xlab ... x軸ラベルの指定  
ylab ... y軸ラベルの指定  
col ... プロットする点の色指定

dfbb\$勝率 は dfbb[,6] でもよい  
dfbb\$防御率 は dfbb[,12] でもよい



# Rでデータの視覚化

## • 散布図を作成(2)

```
> plot(dfbb[,6], dfbb[,12], xlab="勝率", ylab="防御率", type="b")
```

x軸を dfbb[,6]="勝率"

y軸を dfbb[,12]="防御率"

のデータを用い散布図を作成

xlab ... x軸ラベルの指定

ylab ... y軸ラベルの指定

type ... 描画点の種類

"p" ... points 点 (default)

"l" ... lines 線分

"b" ... both 点と線分 両方

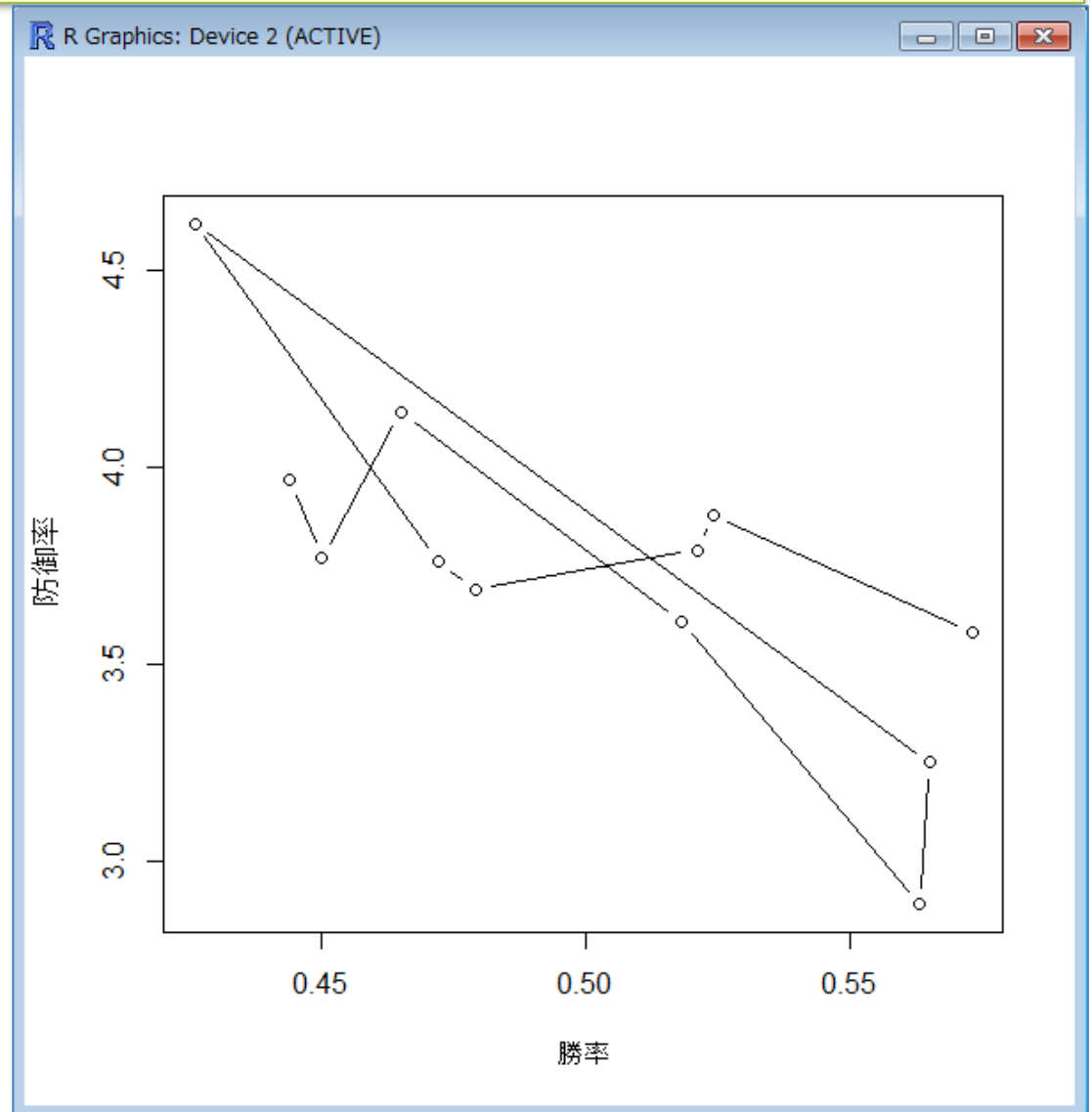
"c" ... "b" から点を抜いたもの

"o" ... overplotted

"h" ... histogram

"s" ... stair steps

"n" ... no plotting 点をかかない



# Rでデータの視覚化

## • 散布図を作成(3)

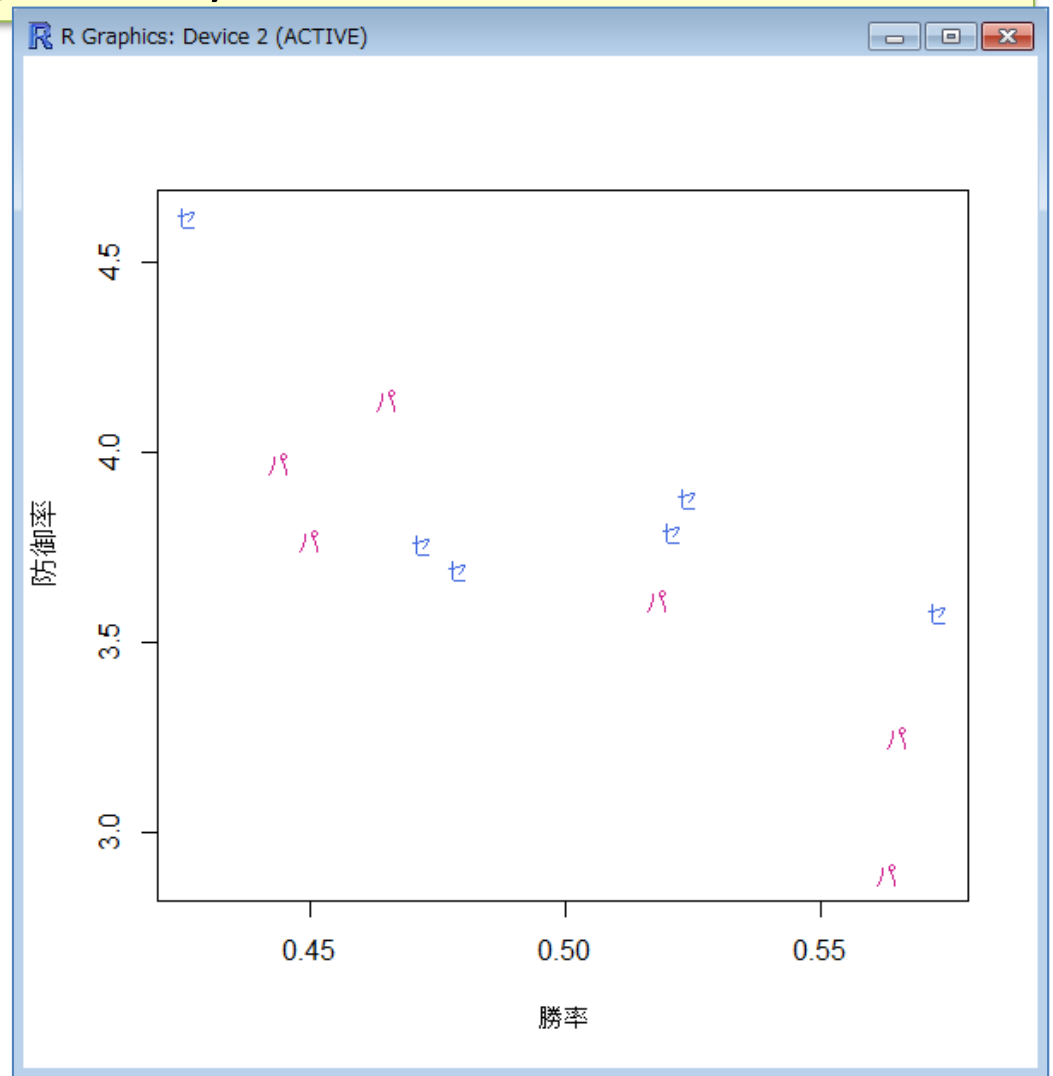
※プロットはせずに、枠・軸だけを描画

```
> plot(dfbb[,6], dfbb[,12], xlab="勝率", ylab="防御率", type="n")  
> text(dfbb[,6], dfbb[,12], dfbb[,1], col=cc)
```

※リーグ名称をプロット点として描く  
(data.frame である dfbb の1列目に  
リーグ名を入れたことを思いだそう!)

※col=cc は色設定を ccにすること  
(cc はリーグ毎の色設定用ベクトル  
として作ったことを思いだそう!)

dfbb[,1] は dfbb\$リーグ でもよい  
dfbb[,6] は dfbb\$勝率 でもよい  
dfbb[,12] は dfbb\$防御率 でもよい



# Rでデータの視覚化

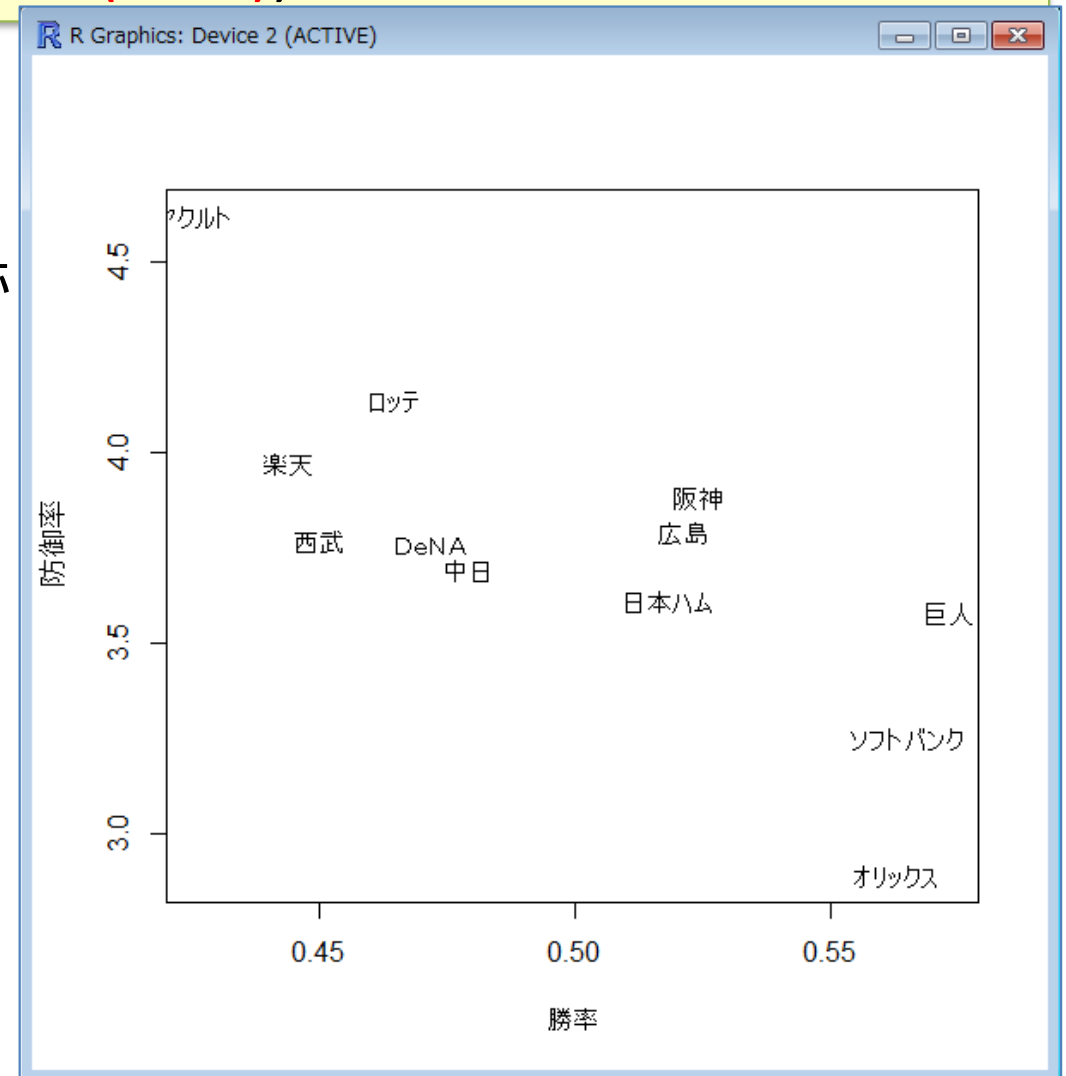
- 散布図を作成(4)

※プロットはせずに、枠・軸だけを描画

```
> plot(dfbb[,6], dfbb[,12], xlab="勝率", ylab="防御率", type="n")  
> text(dfbb[,6], dfbb[,12], row.names(dfbb))
```

※チーム名称をプロット点としてかく  
(read.csvでcsvファイルを読み込んだ時に、row.namesとして1列目のチーム名称を指定したことを思いだそう！)

dfbb[,6] は dfbb\$勝率 でもよい  
dfbb[,12] は dfbb\$防御率 でもよい



## 【参考】

# Rでデータの視覚化

## • 箱ひげ図と散布図を作成(1) -scatterplot()-

```
> install.packages("car")
```

```
> library(car)
```

※scatterplot() の使用準備

package "car" のインストール

package "car" の読み込み

```
> scatterplot(dfbb[,4], dfbb[,8], xlab="負数", ylab="失点")
```

x軸を dfbb[,4]="負数"

y軸を dfbb[,8]="失点"

のデータを用い散布図を作成

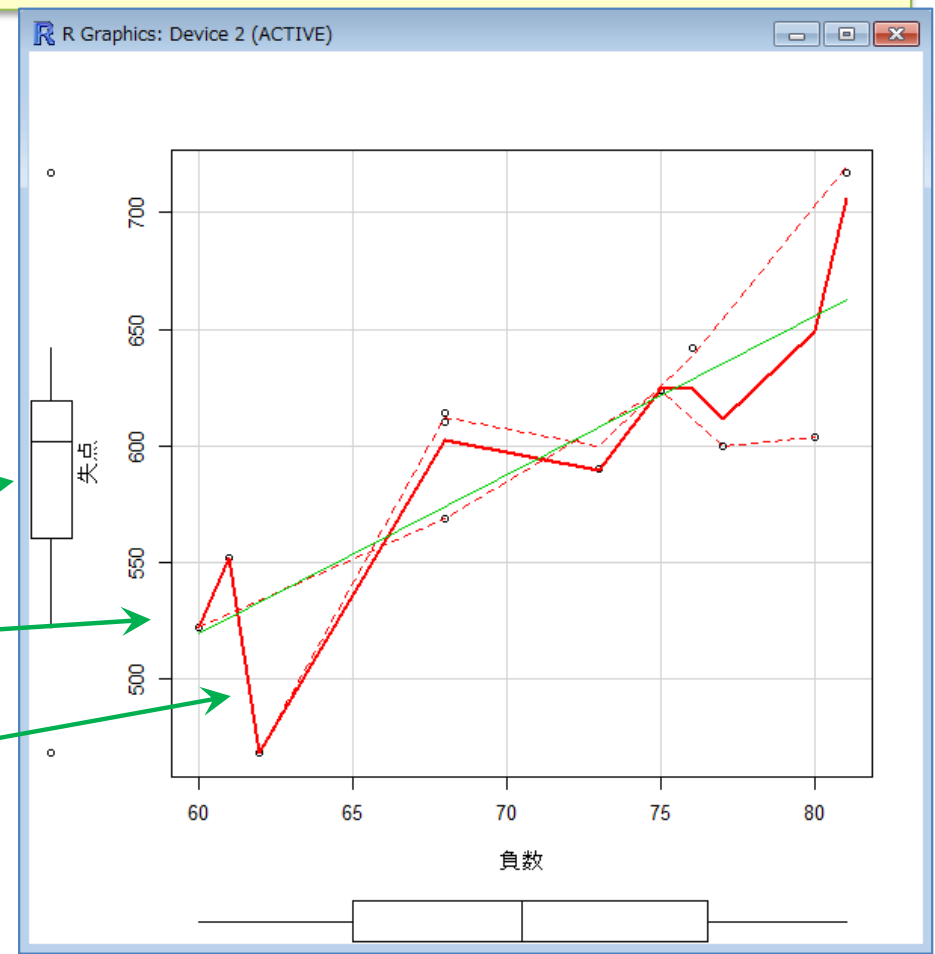
xlab ... x軸ラベルの指定

ylab ... y軸ラベルの指定

※それぞれの軸に、それぞれのデータの箱ひげ図が描かれる

※緑線は回帰直線 regression line

※赤線・赤点線は平滑化線とspan



## 【参考】

# Rでデータの視覚化

## • 箱ひげ図と散布図を作成(2) -scatterplot()-

```
> install.packages("sp")  
> install.packages("maptools")  
> library(sp)  
> library(maptools)
```

※pointLabel() の使用準備

packages "sp","maptools" のインストール

packages "sp", "maptools" の読み込み

(注: 必ず sp → maptools の順!)

## – 点とチーム名を両方プロットする

```
> scatterplot(dfbb[,4], dfbb[,8], xlab="負数", ylab="失点", reg.line=F,  
smooth=F)  
> pointLabel(x=dfbb[,4], y=dfbb[,8], labels=row.names(dfbb))
```

※平滑化線は描かない

※散布図の点のラベルを  
row.names(dfbb)として書く

※回帰直線 regression line  
は描かない(FはFalseの意)

【参考】

# Rでデータの視覚化

