

R と RStudio の使い方

芳賀敏郎 (2011) 医薬品開発のための統計解析 第1部 基礎
4 相関・回帰
4.2 相関係数

テキストと利用上の注意

●テキスト

芳賀敏郎（2011）医薬品開発のための統計解析

第1部 基礎 改訂版、サイエンティスト社、p.275

（サイトへアップすることに対して、サイエンティスト社の了解を得ています）

●Rによる解析事例を紹介

R スクリプトの出力結果を紹介します（tidyverse 系には次期バージョンで対応します）

R スクリプト（文字コードUTF-8に設定）を、このサイトから[ダウンロード](#)できます

R スクリプトを [Compile Report] することにより、Word または HTML で見ることができます

R と RStudio の設定と基本的な使い方は「[R と RStudio の使い方](#)」を参照してください

R の出力結果の見方は、テキストとそれを解説した [PDF ファイル](#)を参照してください

グラフ表示は、解析手段として、必要最小限の表現に止めています

●自己責任で利用

上記のことを理解した上で、自己責任により利用してください

第1部 基礎

- 1. 統計の基礎
 - 1.1 宝くじの期待値と分散、1.2 サイコロの目の数の期待値と分散
 - 1.3 分散の加法性・中心極限定理・正規分布、1.4 統計的推測、1.5 モデル
- 2. 1組のデータの解析
 - 2.1 データの特徴の記述、2.2 データのグラフ表示と外れ値
 - 2.3 対数変換と対数正規分布、2.4 平均に関する推測（母標準偏差 σ 既知）
 - 2.5 分散に関する推測、2.6 平均に関する推測（母標準偏差 σ 未知）
- 3. 2組のデータの解析
 - 3.1 データのグラフ化、3.2 平均値の差の t 検定、3.3 分散の違いの検定
 - 3.4 分散が異なる場合の平均値の差の比較
 - 3.5 対応のある場合の平均値の差の t 検定、3.6 検出力と n の決め方
 - 3.7 ノンパラメトリック検定
- 4. 相関・回帰
 - 4.1 散布図、4.2 **相関係数**、4.3 回帰モデルとモデルの推定
 - 4.4 誤差を考慮した推定、4.5 回帰分析適用上の諸問題

●表示4.2.2 積和と相関係数の計算

スクリプトファイル

Green1-4-2a.R

利用した関数

sum、mean、sd、cov

data.frame

方法

x と y にデータを付値

平方和、積和、標準偏差、

自由度、共分散を使って、

相関係数を算出

```
x <- df_ens$height
y <- df_ens$weight
n <- length(df_ens$height)
```

```
Sxx <- sum((x - mean(x))^2) # 平方和
Syy <- sum((y - mean(y))^2) # 平方和
Sxy <- sum((x - mean(x))*(y - mean(y))) # 積和

r1 <- sum((x - mean(x)) / sd(x) * (y - mean(y)) / sd(y))
r1 <- r1 / (n - 1)
r2 <- Sxy / sqrt(Sxx * Syy)
r3 <- Sxy / (n - 1) * 1/(sd(x) * sd(y))
r4 <- cov(x, y) / (sd(x) * sd(y))

col1 <- c("Sxx", "Syy", "Sxy", "n", "r", "r", "r", "r")
col2 <- c(Sxx, Syy, Sxy, n, r1, r2, r3, r4)
data.frame(col1, col2)

## 1 Sxx 974.7750000
## 2 Syy 1821.7750000
## 3 Sxy 553.5250000
## 4 n 40.0000000
## 5 r 0.4153724
```

●表示4.2.2 積和と相関係数の計算

スクリプトファイル

Green1-4-2a.R

利用した関数

sum、mean、sd、cov

data.frame

方法

x と y にデータを付値

平方和、積和、標準偏差、

自由度、共分散を使って、

相関係数を算出

```
x <- df_ens$height
y <- df_ens$weight
n <- length(df_ens$height)
```

```
Sxx <- sum((x - mean(x))^2) # 平方和
Syy <- sum((y - mean(y))^2) # 平方和
Sxy <- sum((x - mean(x))*(y - mean(y))) # 積和

r1 <- sum((x - mean(x)) / sd(x) * (y - mean(y)) / sd(y))
r1 <- r1 / (n - 1)
r2 <- Sxy / sqrt(Sxx * Syy)
r3 <- Sxy / (n - 1) * 1/(sd(x) * sd(y))
r4 <- cov(x, y) / (sd(x) * sd(y))

col1 <- c("Sxx", "Syy", "Sxy", "n", "r", "r", "r", "r")
col2 <- c(Sxx, Syy, Sxy, n, r1, r2, r3, r4)
data.frame(col1, col2)

## 1 Sxx 974.7750000
## 2 Syy 1821.7750000
## 3 Sxy 553.5250000
## 4 n 40.0000000
## 5 r 0.4153724
```

r1 の値
r2~r4 の値は省略

- 表示4.2.4 JMP による相関係数の出力

スクリプトファイル：Green1-4-2a.R

利用した関数：split、cor.test、sapply、
function

方法

データフレームをsplit関数で性別に分け、
sapply関数で男女ごとにcor.testを適用
sapply(リスト、関数)のように使用

2つの
データフレームを
要素とするリスト

lst\$F lst[[1]]
データフレーム

lst\$M lst[[2]]
データフレーム

```
lst <- split(df_big, df_big$sex)
```

```
## $F
##      name age sex height weight
## 1    KATIE  12  F     59      95
## 2   LOUISE  12  F     61     123
## .....
## 38   LINDA  17  F     62     116

## $M
##      name age sex height weight
## 6     TIM  12  M     60      84
## 7    JAMES  12  M     61     128
## .....
## 40 LAWRENCE 17  M     70     172
```

```
sapply(lst, function(i) {
  cor.test(i$height, i$weight,
           alternative = "two.sided",
           method = "pearson",
           conf.level = 0.95)})
```

lst\$M\$weight
または lst[[2]]\$weight

```
(1) i <- lst[[1]]
cor.test(lst[[1]]$height,
         lst[[1]]$weight...
```

```
(2) i <- lst[[2]]
cor.test(lst[[2]]$height,
         lst[[2]]$weight...
```

2つの
データフレームを
要素とするリスト

lst\$F lst[[1]]
データフレーム

lst\$M lst[[2]]
データフレーム

データフレームをsplit関数で性別に分け、
sapply関数で男女ごとにcor.testを適用
sapply(リスト、関数)のように使用

第1引数

第2引数

```
sapply(lst, function(i) {
  cor.test(i$height, i$weight,
           alternative = "two.sided",
           method = "pearson",
           conf.level = 0.95)})
```

```
lst <- split(df_big, df_big$sex)

## $F
##      name age sex height weight
## 1  KATIE  12  F    59      95
## 2  LOUISE  12  F    61     123
## .....
## 38  LINDA  17  F    62     116

## $M
##      name age sex height weight
## 6    TIM  12  M    60      84
## 7   JAMES  12  M    61     128
## .....
## 40 LAWRENCE 17  M    70     172
```

lst\$M\$weight
または lst[[2]]\$weight

● 表示4.2.4 JMP による相関係数の出力

スクリプトファイル：Green1-4-2a.R

利用した関数：split、cor.test、sapply、
function

方法

データフレームをsplit関数で性別に分け、
sapply関数で男女ごとにcor.testを適用
sapply(リスト、関数)のように使用

```
sapply(1st, function(i) {  
  cor.test(i$height, i$weight,  
    alternative = "two.sided",  
    method = "pearson",  
    conf.level = 0.95)})
```

cor.testの結果が男女ごとに出カ

```
## F  
## statistic 4.346776  
## parameter 16  
## p.value 0.000499556  
## estimate 0.7358495  
## null.value 0  
## alternative "two.sided"  
## method "Pearson's product-moment  
correlation"  
## data.name "i$height and i$weight"  
## conf.int Numeric,2
```

```
## M  
## statistic 4.315711  
## parameter 20  
.....
```



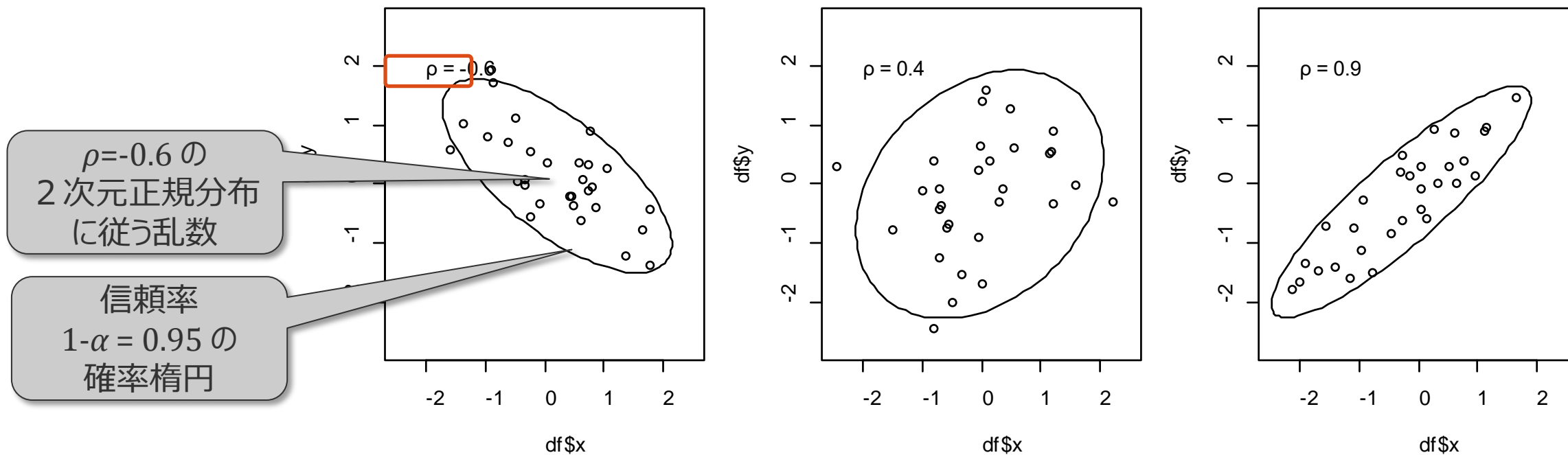
2次元正規分布：確率楕円

●表示4.2.5 2次元正規分布の散布図と相関係数

スクリプトファイル：Green1-4-2b.R

利用した関数：rnorm、plot、polygon、ellipse::ellipse

方法：2次元正規分布に従う乱数を発生（p.262）、plot関数で散布図を描画、
polygon関数とellipse関数で確率楕円を描画（[§4.1](#) 参照）



2次元正規分布：確率楕円

p.209

●表示4.2.5 2次元正規分布の散布図と相関係数

スクリプトファイル：Green1-4-2b.R

利用した関数：rnorm、plot、polygon、
ellipse::ellipse

方法：2次元正規分布に従う乱数を発生（p.262）、
polygon 関数と ellipse 関数で確率楕円を描画

```
rnorm_2dim <- function(rho) {  
  u <- rnorm(30, mean = 0, sd = 1)  
  v <- rnorm(30, mean = 0, sd = 1)  
  x <- u  
  y <- rho * u + sqrt(1 - rho^2) * v  
  df <- data.frame(x, y)  
  return(df)  
}
```

```
sc <- 2.5 # 軸のスケール  
rho <- c(-0.6, 0.4, 0.9)  
for (i in 1:3) {  
  df <- rnorm_2dim(rho[i])  
  plot(df$x, df$y, xlim = c(-sc, sc), ylim = c(-sc, sc), asp = 1)  
  polygon(ellipse(cov(df), centre = colMeans(df), level = 0.90))  
  text(-2, 2, paste("ρ =", rho[i]))  
}
```

乱数を df に付値

散布図を描画

確率楕円を描画

rho[1] = -0.6
rho[2] = 0.4
rho[3] = 0.9

縦軸と横軸の比



2次元正規分布：確率楕円

●表示4.2.5 2次元正規分布の散布図と相関係数

スクリプトファイル：Green1-4-2b.R

利用した関数：rnorm、plot、polygon、
ellipse::ellipse

方法：2次元正規分布に従う乱数を発生（p.262）、
plot 関数で散布図を描画、
polygon 関数と ellipse 関数で確率楕円を描画

cov 関数は、分散共分散行列を生成
($\rho=0.9$ の場合の一例)

```
cov(df)
##           x           y
## x 1.288381 1.127645
## y 1.127645 1.151751
```

共分散
分散

分散共分散行列を生成

$\rho = 0.9$ の
2次元正規分布
に従う乱数 x, y

x と y の平均

```
ellipse(cov(df), centre = colMeans(df),
        level = 0.90)
##           x           y
## [1,] 2.512511065 2.171645665
## [2,] 2.469436096 2.201789905
## [3,] 2.418465411 2.224901187
## [4,] 2.359834262 2.240887177
## [5,] 2.293778726 2.249683596
## . . . . .
## [98,] 2.574491966 2.090776914
## [99,] 2.547606704 2.134592560
## [100,] 2.512511065 2.171645665
```

信頼率



2次元正規分布：確率楕円

p.209

●表示4.2.5 2次元正規分布の散布図と相関係数

スクリプトファイル：Green1-4-2b.R

利用した関数：rnorm、plot、polygon、

ellipse::ellipse

方法：2次元正規分布に従う乱数を発生（p.262）、
polygon 関数と ellipse 関数で確率楕円を描画

```
rnorm_2dim <- function(rho) {  
  u <- rnorm(30, mean = 0, sd = 1)  
  v <- rnorm(30, mean = 0, sd = 1)  
  x <- u  
  y <- rho * u + sqrt(1 - rho^2) * v  
  df <- data.frame(x, y)  
  return(df)  
}
```

```
sc <- 2.5 # 軸のスケール  
rho <- c(-0.6, 0.4, 0.9)  
for (i in 1:3) {  
  df <- rnorm_2dim(rho[i])  
  plot(df$x, df$y, xlim = c(-sc, sc), ylim = c(-sc, sc), asp = 1)  
  polygon(ellipse(cov(df), centre = colMeans(df), level = 0.90))  
  text(-2, 2, paste("ρ =", rho[i]))  
}
```

乱数を df に付値

散布図を描画

確率楕円を描画

相関係数 r の分布と z の分布

- 表示4.2.7 相関係数 r の分布と z の分布
($\rho = 0.0, 0.2, 0.4, 0.6, 0.8$)

スクリプトファイル : Green1-4-2c.R

利用した関数

DescTools::FisherZ、

DescTools::FisherZInv

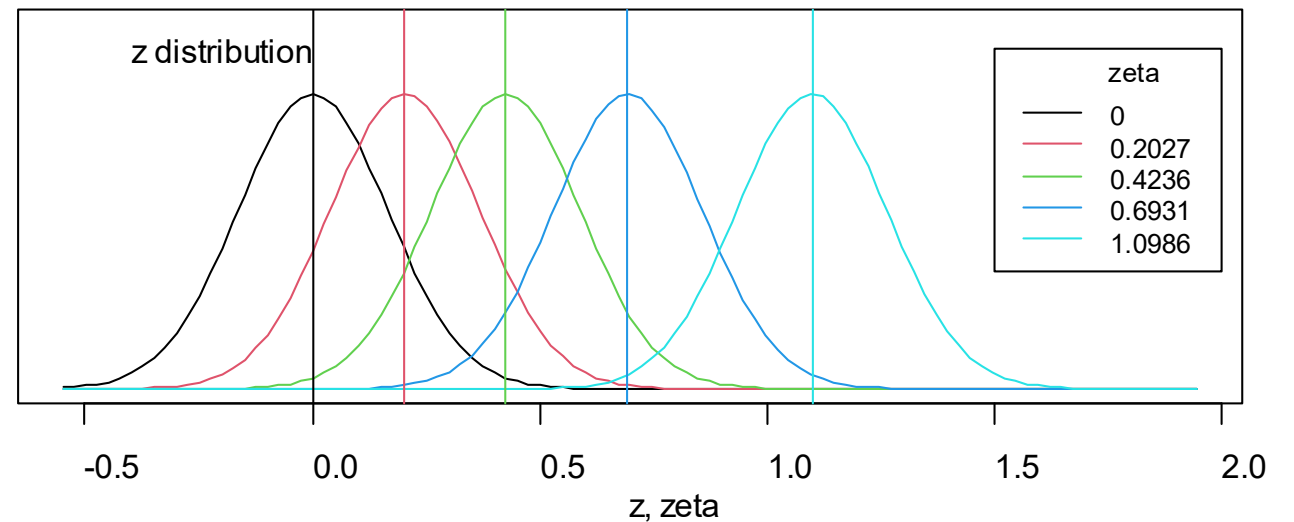
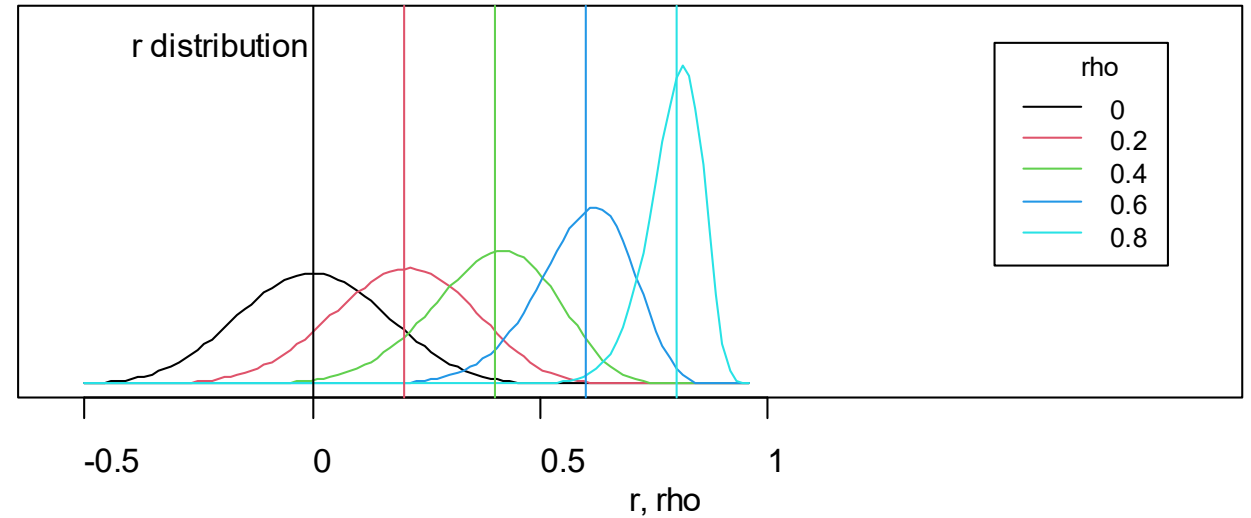
plot、curve、axis、legend、text

function、for

方法

function 関数とFisherZ 関数で分布関数
を作成

plot 関数と curve 関数で作図





相関係数 r の分布と z の分布

- 表示4.2.8 相関係数の検定、区間推定の計算表

スクリプトファイル：Green1-4-2d.R

利用した関数

DescTools::FisherZ、DescTools::FisherZInv、qnorm

方法

FisherZ 関数で ρ を変換した z で正規分布の区間推定

FisherZInv 関数で z の区間推定を r の区間推定に戻す

##		n	rho	alpha	zeta	interval	lowerZ	upperZ	lowerR	upperR	lowerR-rho	rho-upperR
##	[1,]	40	0.0	0.05	0.000	0.322	-0.322	0.322	-0.312	0.312	0.312	0.3120
##	[2,]	40	0.2	0.05	0.203	0.322	-0.119	0.525	-0.119	0.482	0.319	0.2820
##	[3,]	40	0.4	0.05	0.424	0.322	0.101	0.746	0.101	0.633	0.299	0.2330
##	[4,]	40	0.6	0.05	0.693	0.322	0.371	1.020	0.355	0.768	0.245	0.1680
##	[5,]	40	0.8	0.05	1.100	0.322	0.776	1.420	0.651	0.890	0.149	0.0898

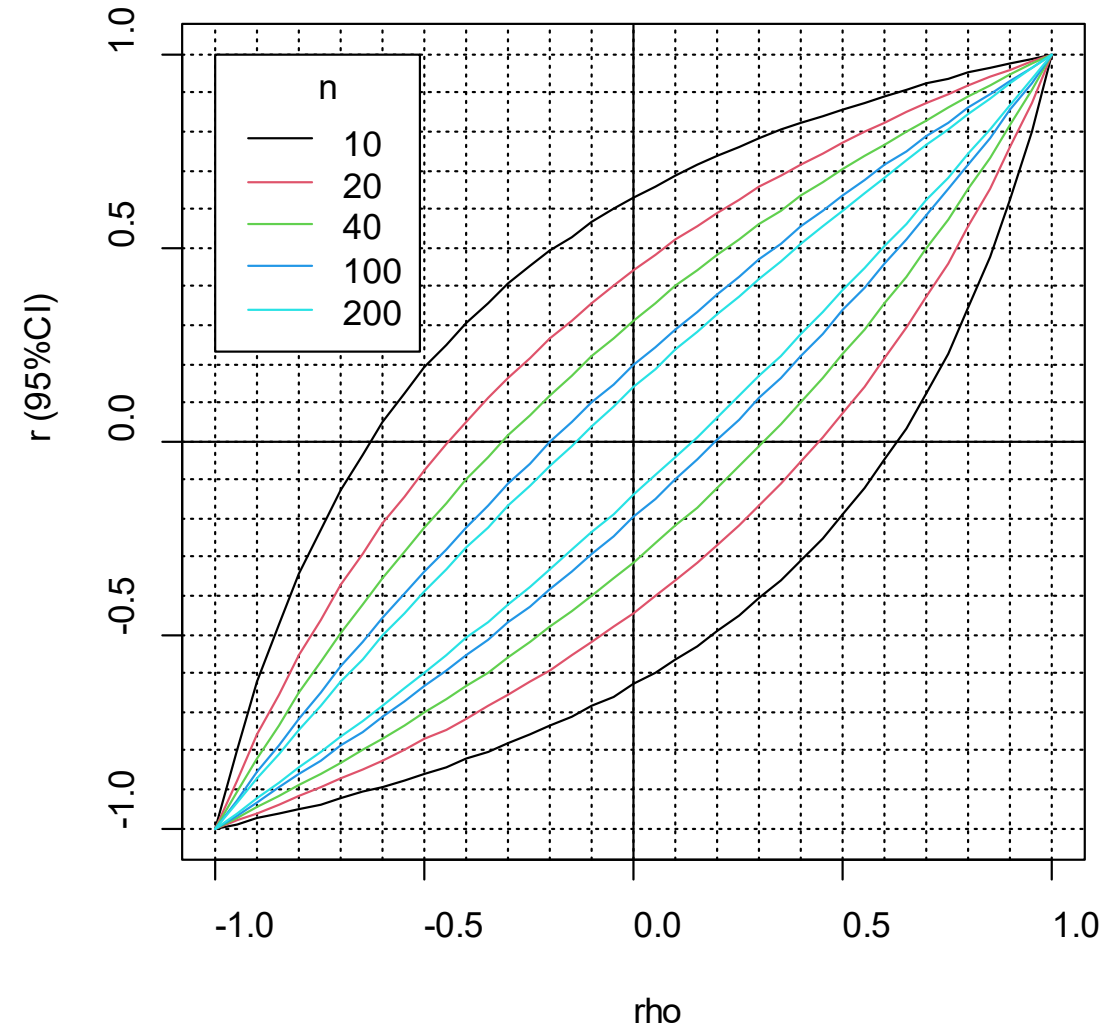
相関係数の検定、区間推定のダイアグラム

- 表示4.2.9 相関係数の検定、区間推定のダイアグラム

スクリプトファイル：Green1-4-2e.R

利用した関数：DescTools::CorCI、sapply、for
plot、abline、lines

方法：CorCI 関数で rho に対応した r の区間推定
(信頼率 $1 - \alpha = 0.95$)、plot 関数で描画



相関係数の検定、区間推定のダイアグラム

p.214

- 表示4.2.9 相関係数の検定、区間推定のダイアグラム

スクリプトファイル：Green1-4-2e.R

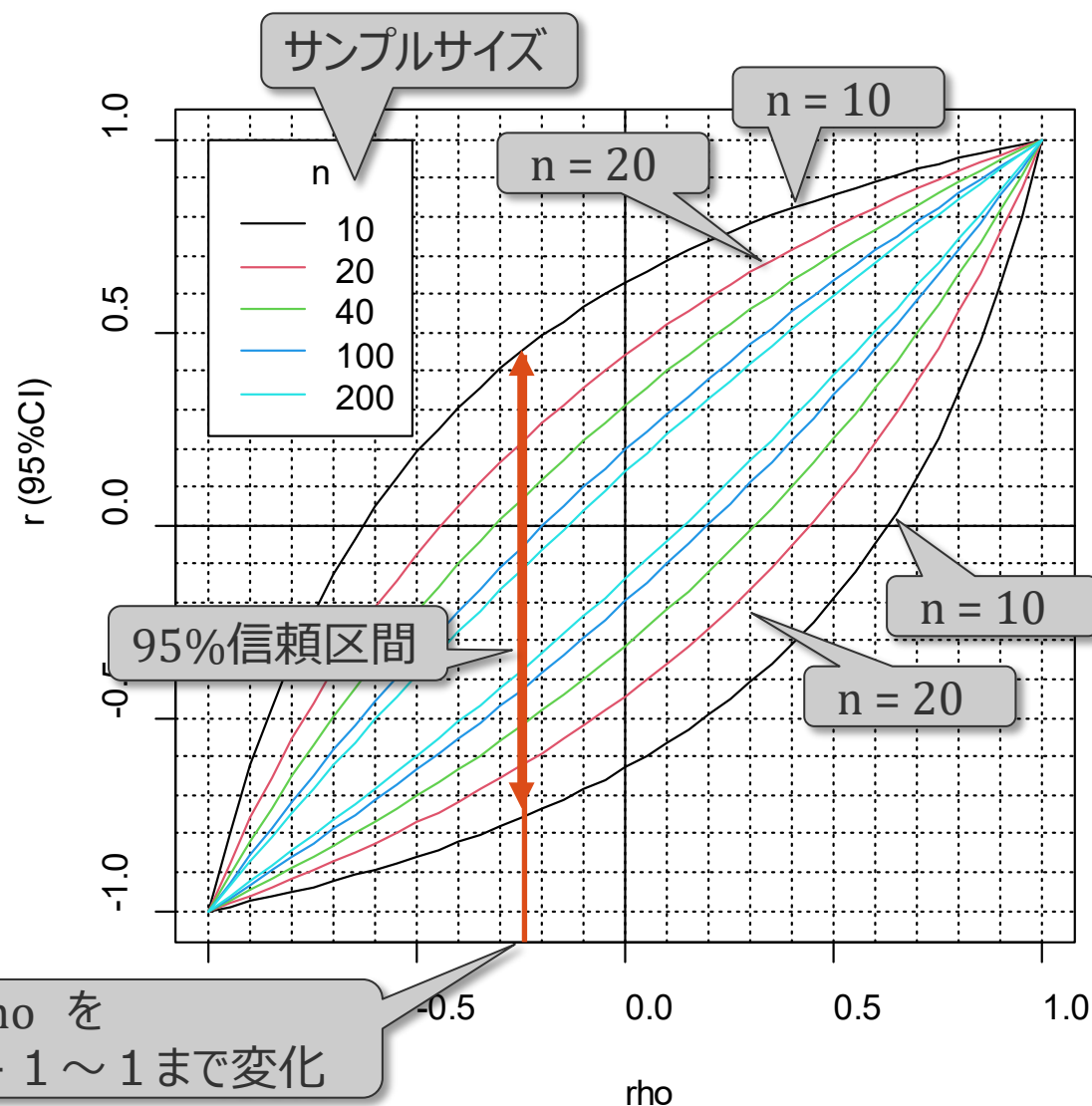
利用した関数：DescTools::CorCI、sapply、for
plot、abline、lines

方法：CorCI関数でrhoに対応したrの区間推定
(信頼率 $1 - \alpha = 0.95$)、plot関数で描画

```
n <- c(10, 20, 40, 100, 200) # サンプルサイズ
rho <- seq(-1, 1, 0.05) # 母相関係数 41個
lst <- list() # 空のリストを作成

for (i in 1:5) {
  nn = n[i]
  rg <- t(sapply(rho, CorCI, n = nn,
                conf.level = 0.95,
                alternative = "two.sided"))

  lst[[i]] <- rg
}
```



相関係数の検定、区間推定のダイアグラム

● 表示4.2.9 相関係数の検定、区間推定のダイアグラム

スクリプトファイル：Green1-4-2e.R

利用した関数：DescTools::CorCI、sapply、for
plot、abline、lines

方法：CorCI関数でrhoに対応したrの区間推定
(信頼率 $1 - \alpha = 0.95$)、plot関数で描画

```
n <- c(10, 20, 40, 100, 200) # サンプルサイズ
rho <- seq(-1, 1, 0.05) # 母相関係数 41個
lst <- list() # 空のリストを作成

for (i in 1:5) {
  nn = n[i]
  rg <- t(sapply(rho, CorCI, n = nn,
                conf.level = 0.95,
                alternative = "two.sided"))

  lst[[i]] <- rg
}
```

```
lst[[1]]
##          cor          lwr.ci          upr.ci
## [1,] -1.00 -1.000000000 -1.000000000
## [2,] -0.95 -0.98841239 -0.79723703
## [3,] -0.90 -0.97635908 -0.62393499
## [4,] -0.85 -0.96381143 -0.47410786
##          . . . . .
## [39,]  0.90  0.62393499  0.97635908
## [40,]  0.95  0.79723703  0.98841239
## [41,]  1.00  1.00000000  1.00000000
```

rho を
-1 ~ 1 まで変化

CorCI(rho, n, conf.level = 0.95,
alternative = "two.sided")
ρと信頼区間の上限と下限の3つの値が返る

相関係数の検定、区間推定のダイアグラム

●表示4.2.9 相関係数の検定、区間推定のダイアグラム

スクリプトファイル：Green1-4-2e.R

利用した関数：DescTools::CorCI, sapply, for

plot,
方法：CorCI関数で
(信頼率 $1 - \alpha =$

```
n <- c(10, 20, 40, 100)
rho <- seq(-1, 1, 0.05)
lst <- list()

for (i in 1:5) {
  nn = n[i]
  rg <- t(sapply(rho, CorCI, n = nn,
                 conf.level = 0.95,
                 alternative = "two.sided"))
  lst[[i]] <- rg
}
```

- (1) rho = -1 CorCI(-1, n[1]
- (2) rho = -0.95 CorCI(-0.95, n[1]
- (3) rho = -0.90 CorCI(-0.90, n[1]
- (4) rho = -0.85 CorCI(-0.85, n[1]
-

```
lst[[1]]
##          cor          lwr.ci          upr.ci
## [1,] -1.00 -1.00000000 -1.00000000
## [2,]  0.95  0.98841239 -0.79723703
## [3,]  0.90  0.97635908 -0.62393499
## [4,]  0.85  0.96381143 -0.47410786
## [5,]  0.80  0.95175804 -0.32427581
## [6,]  0.75  0.94000000 -0.17444374
## [7,]  0.70  0.92875000  0.07538823
## [8,]  0.65  0.91800000  0.32521620
## [9,]  0.60  0.90775000  0.57504417
## [10,] 0.55  0.89800000  0.82487214
## [11,] 0.50  0.88875000  0.97470011
## [12,] 0.45  0.88000000  1.02452808
## [13,] 0.40  0.87175000  1.07435605
## [14,] 0.35  0.86400000  1.12418402
## [15,] 0.30  0.85675000  1.17401199
## [16,] 0.25  0.85000000  1.22384000
## [17,] 0.20  0.84375000  1.27366807
## [18,] 0.15  0.83800000  1.32349604
## [19,] 0.10  0.83275000  1.37332401
## [20,] 0.05  0.82800000  1.42315198
## [21,] 0.00  0.82375000  1.47297995
## [22,] -0.05 0.82000000  1.52280792
## [23,] -0.10 0.81675000  1.57263589
## [24,] -0.15 0.81400000  1.62246386
## [25,] -0.20 0.81175000  1.67229183
## [26,] -0.25 0.81000000  1.72211980
## [27,] -0.30 0.80875000  1.77194777
## [28,] -0.35 0.80800000  1.82177574
## [29,] -0.40 0.80775000  1.87160371
## [30,] -0.45 0.80800000  1.92143168
## [31,] -0.50 0.80875000  1.97125965
## [32,] -0.55 0.80900000  2.02108762
## [33,] -0.60 0.80975000  2.07091559
## [34,] -0.65 0.81000000  2.12074356
## [35,] -0.70 0.81075000  2.17057153
## [36,] -0.75 0.81100000  2.22039950
## [37,] -0.80 0.81175000  2.27022747
## [38,] -0.85 0.81200000  2.32005544
## [39,] -0.90 0.81275000  2.36988341
## [40,] -0.95 0.81300000  2.41971138
## [41,] -1.00 0.81375000  2.46953935
```

相関係数 41個
空のリストを作成

第1引数 第2引数

転置

CorCI(rho, n, conf.level = 0.95, alternative = "two.sided")
 ρ と信頼区間の上限と下限の3つの値が返る

相関係数の検定、区間推定のダイアグラム

● 表示4.2.9 相関係数の検定、区間推定のダイアグラム

スクリプトファイル：Green1-4-2e.R

利用した関数：DescTools::CorCI、sapply、for
plot、abline、lines

方法：CorCI関数でrhoに対応したrの区間推定
(信頼率 $1 - \alpha = 0.95$)、plot関数で描画

```
n <- c(10, 20, 40, 100, 200) # サンプルサイズ
rho <- seq(-1, 1, 0.05) # 母相関係数 41個
lst <- list() # 空のリストを作成

for (i in 1:5) {
  nn = n[i]
  rg <- t(sapply(rho, CorCI, n = nn,
                 conf.level = 0.95,
                 alternative = "two.sided"))
  lst[[i]] <- rg
}
```

```
lst[[1]]
##          cor          lwr.ci          upr.ci
## [1,] -1.00 -1.000000000 -1.000000000
## [2,] -0.95 -0.98841239 -0.79723703
## [3,] -0.90 -0.97635908 -0.62393499
## [4,] -0.85 -0.96381143 -0.47410786
##          . . . . .
## [39,]  0.90  0.62393499  0.97635908
## [40,]  0.95  0.79723703  0.98841239
## [41,]  1.00  1.00000000  1.00000000
```

rho <-
seq(-1, 1, 0.05)

信頼区間の上限と下限
CorCI関数の出力

CorCI(rho, n, conf.level = 0.95,
alternative = "two.sided")
 ρ と信頼区間の上限と下限の3つの値が返る

相関係数の検定、区間推定のダイアグラム

- 表示4.2.9 相関係数の検定、区間推定のダイアグラム

スクリプトファイル：Green1-4-2e.R

利用した関数：DescTools::CorCI、sapply、for
plot、abline、lines

方法：CorCI関数で rho に対応した r の区間推定
(信頼率 $1 - \alpha = 0.95$)、plot 関数で描画

グラフの枠のみ表示

```
plot(NA, xlim = c(-1, 1), ylim = c(-1, 1),  
     xlab = "rho", ylab = "r (95%CI)", asp = 1)
```

```
for (i in 1:5) {  
  lines(lst[[i]][, 1], lst[[i]][, 2],  
        type = "l", col = i)  
  lines(lst[[i]][, 1], lst[[i]][, 3],  
        type = "l", col = i)  
}
```

信頼区間の下限

信頼区間の上限

リスト lst の第 1 要素

```
lst[[1]]  
##          cor          lwr.ci          upr.ci  
## [1,] -1.00 -1.000000000 -1.000000000  
## [2,] -0.95 -0.98841239 -0.79723703  
## [3,] -0.90 -0.97635908 -0.62393499  
## [4,] -0.85 -0.96381143 -0.47410786  
##          . . . . .  
## [39,]  0.90  0.62393499  0.97635908  
## [40,]  0.95  0.79723703  0.98841239  
## [41,]  1.00  1.00000000  1.00000000
```

lst[[1]][, 1]
母相関係数 ρ

lst[[1]][, 2]
信頼区間の下限

lst[[1]][, 3]
信頼区間の上限

相関係数の検定、区間推定のダイアグラム

●表示4.2.9 相関係数の検定、区間推定のダイアグラム

スクリプトファイル：Green1-4-2e.R

利用した関数：DescTools::CorCI、sapply、for
plot、abline、lines

方法：CorCI 関数で rho に対応した r の区間推定
(信頼率 $1 - \alpha = 0.95$)、plot 関数で描画

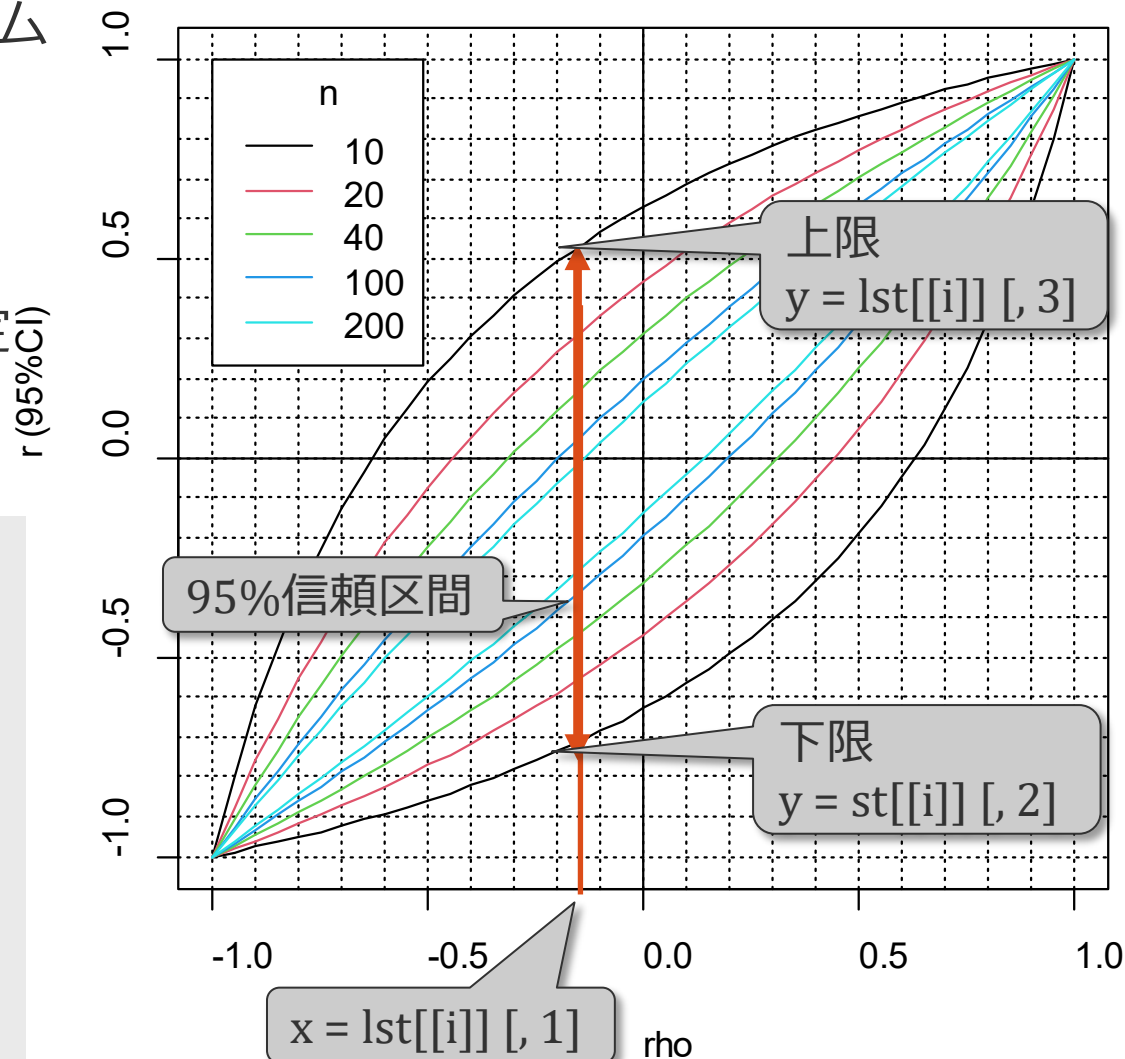
グラフの枠のみ表示

```
plot(NA, xlim = c(-1, 1), ylim = c(-1, 1),  
     xlab = "rho", ylab = "r (95%CI)", asp = 1)
```

```
for (i in 1:5) {  
  lines(lst[[i]][, 1], lst[[i]][, 2],  
        type = "l", col = i)  
  lines(lst[[i]][, 1], lst[[i]][, 3],  
        type = "l", col = i)  
}
```

信頼区間の下限

信頼区間の上限



- 表示4.2.13 分析ツールによる相関係数行列

スクリプトファイル：Green1-4-2f.R

利用した関数：cor

方法：データフレームの df_ens の 2列～11列を cor 関数に渡して、2列ごとに相関係数を算出

2～11列の全行

```
signif(cor(df_ens[, 2:11], method = "pearson"), digits = 3)
```

	age	height	weight	SBP	DBP	TC	HDL	TG	TN
## age	1.0000	-0.24600	-0.354	0.1030	0.0700	-0.15900	0.0251	-0.1620	-0.0491
## height	-0.2460	1.00000	0.415	0.1180	0.0409	-0.00565	0.2760	-0.1550	0.0740
## weight	-0.3540	0.41500	1.000	0.2790	0.1470	0.15700	-0.1540	0.1320	0.2670
## SBP	0.1030	0.11800	0.279	1.0000	0.7650	0.40000	-0.0585	0.4400	0.0692
## DBP	0.0700	0.04090	0.147	0.7650	1.0000	0.43200	-0.2020	0.4240	-0.0635
## TC	-0.1590	-0.00565	0.157	0.4000	0.4320	1.00000	-0.2590	0.5150	0.3060
## HDL	0.0251	0.27600	-0.154	-0.0585	-0.2020	-0.25900	1.0000	-0.3990	-0.1500
## TG	-0.1620	-0.15500	0.132	0.4400	0.4240	0.51500	-0.3990	1.0000	0.0617
## TN	-0.0491	0.07400	0.267	0.0692	-0.0635	0.30600	-0.1500	0.0617	1.0000

相関係数行列

- 表示4.2.14 JMP による相関係数行列と散布図行列(1)

スクリプトファイル

Green1-4-2f.R

利用した関数

cor、

psych::describe

cor(df)

##		TC	HDL	TG	TN	BS
##	TC	1.0000000	-0.25921065	0.51534669	0.30588138	0.23470596
##	HDL	-0.2592107	1.0000000	-0.39871517	-0.14952937	-0.08789918
##	TG	0.5153467	-0.39871517	1.0000000	0.06173477	0.35716629
##	TN	0.3058814	-0.14952937	0.06173477	1.0000000	0.37415410
##	BS	0.2347060	-0.08789918	0.35716629	0.37415410	1.0000000

describe(df)

##	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se	
##	TC	1	40	192.90	28.78	194.5	189.84	22.98	153.0	300.0	147.0	1.55	3.71	4.55
##	HDL	2	40	51.83	17.13	50.0	49.59	10.38	28.0	110.0	82.0	1.74	3.89	2.71
##	TG	3	40	142.50	100.14	117.5	120.88	34.84	50.0	580.0	530.0	2.95	8.93	15.83
##	TN	4	40	7.37	0.36	7.4	7.37	0.30	6.5	8.3	1.8	0.09	0.24	0.06
##	BS	5	40	88.08	9.53	85.0	86.62	7.41	73.0	117.0	44.0	1.28	1.24	1.51

相関係数行列

- 表示4.2.14 JMP による相関係数行列と散布図行列(2)

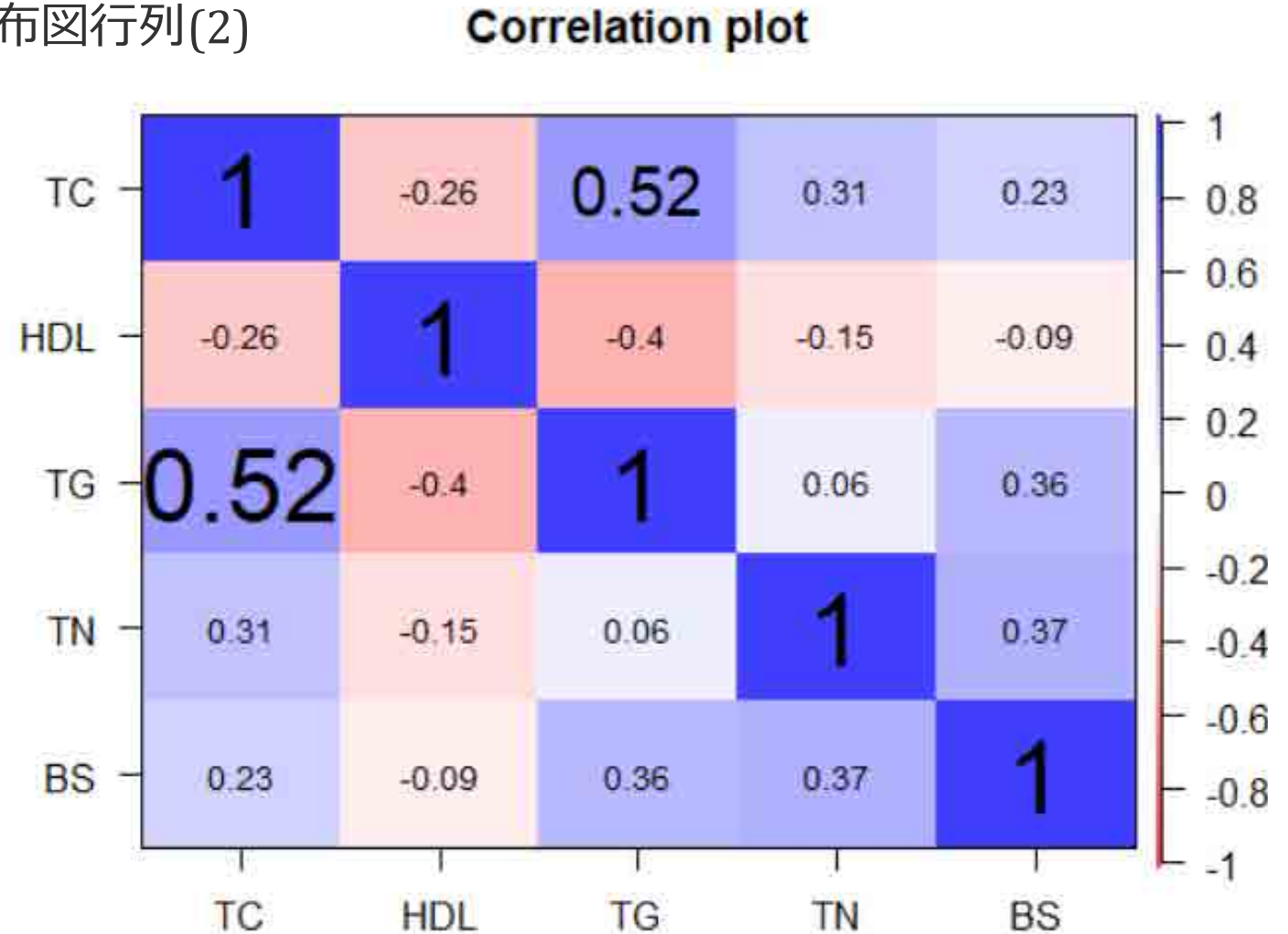
スクリプトファイル：Green1-4-2f.R

利用した関数：psych::cor.plot

方法

視覚化された相関係数行列を得る

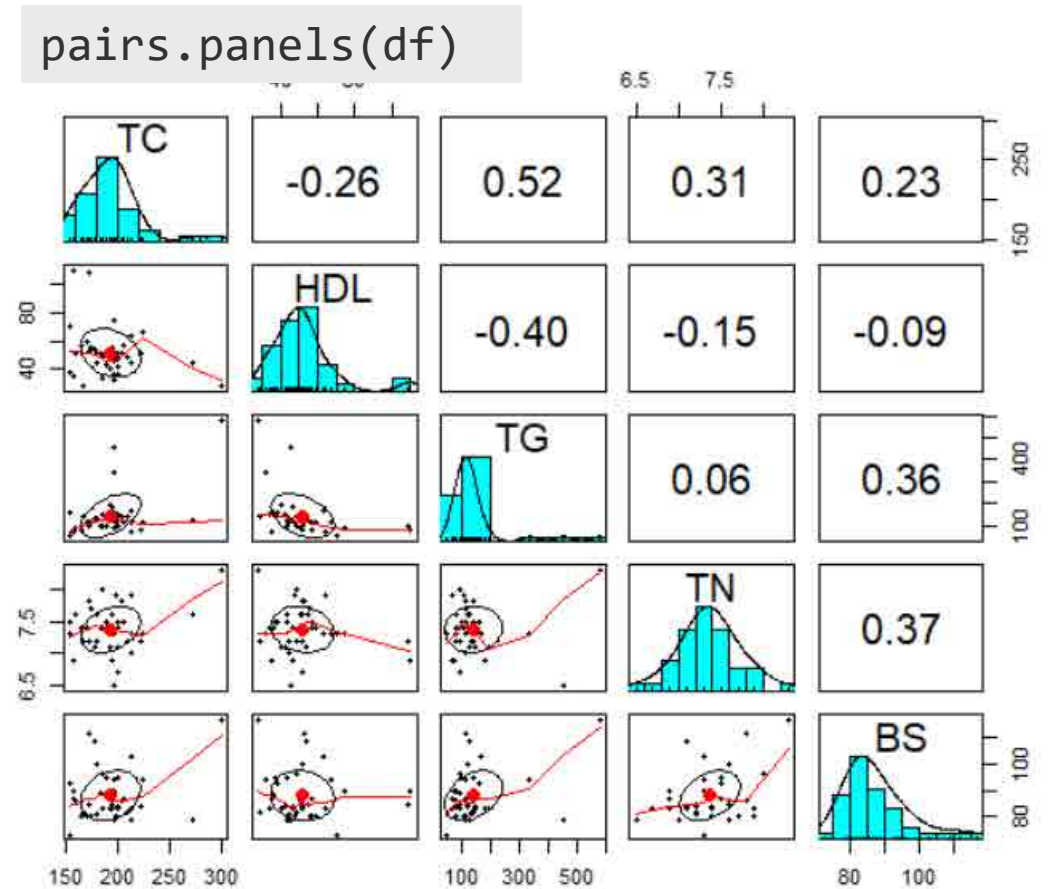
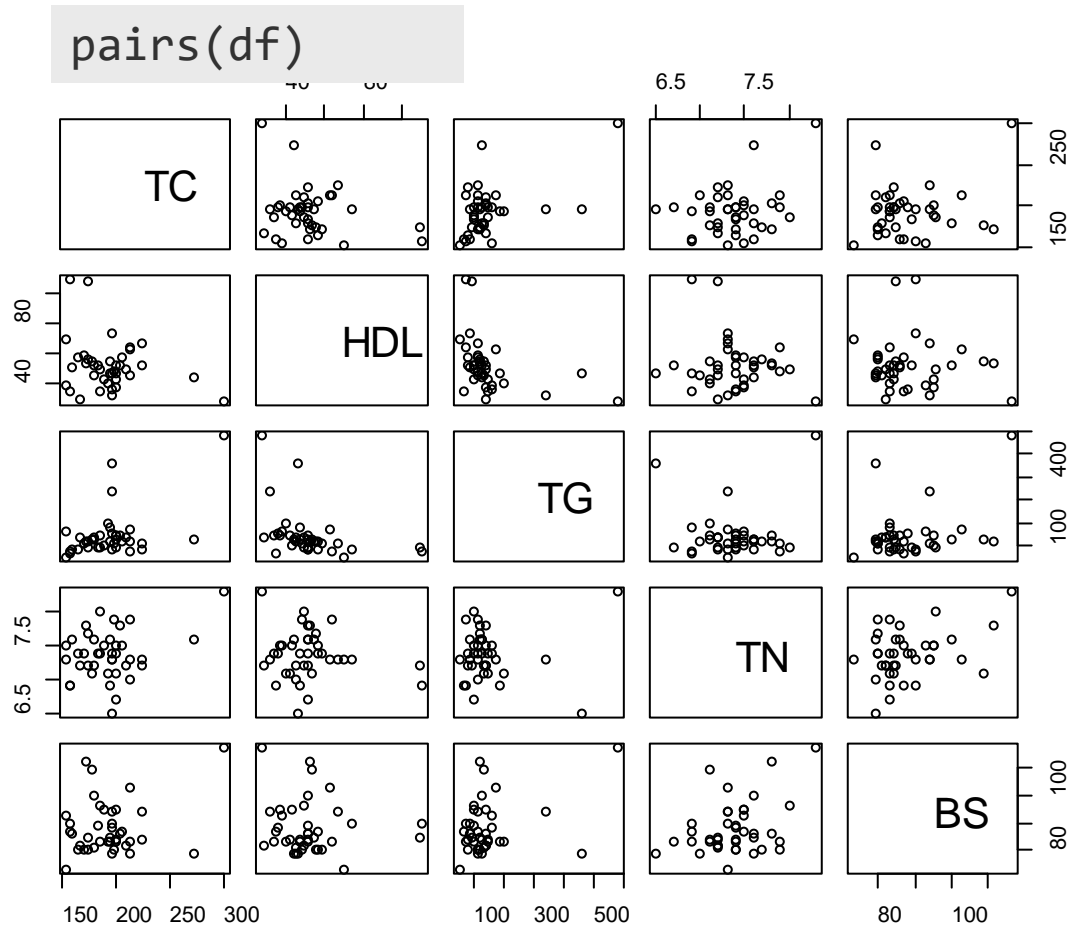
```
cor.plot(df, numbers = TRUE)
```



相関係数行列

- 表示4.2.14 JMP による相関係数行列と散布図行列(3)

スクリプトファイル：Green1-4-2f.R、利用した関数：pairs、spsych::pairs.panels



- 表示4.2.17 JMP による順位相関係数

スクリプトファイル：Green1-4-2f.R、利用した関数：cor、cor.test

または "kendall"

```
cor.test(df_ens$HDL, df_ens$TC, alternative = "two.sided", method = "spearman")  
  
## Spearman's rank correlation rho  
## data: df_ens$HDL and df_ens$TC  
## S = 11876, p-value = 0.4833  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
##      rho  
## -0.114087  
  
cor(df_ens[, 7:11], method = "spearman")  
##           TC           HDL           TG           TN           BS  
## TC    1.00000000 -0.11408703  0.30153067  0.10829778  0.03132499  
## HDL -0.11408703  1.00000000 -0.59333914  0.02722874 -0.01209355  
## TG   0.30153067 -0.59333914  1.00000000 -0.04592871  0.12747468  
## TN   0.10829778  0.02722874 -0.04592871  1.00000000  0.25654331  
## BS   0.03132499 -0.01209355  0.12747468  0.25654331  1.00000000
```



- 作成 片瀬雅彦
- 作成時期 2021年8月18日