

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
3 2組のデータの解析
3.7 ノンパラメトリック検定

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

Wilcoxon の順位和検定

p.171

- 表示3.7.3 JMP による順位和検定

スクリプトファイル：Green1-3-7a.R

利用した関数：wilcox.test、coin::wilcox_test (「.」 「_」の違いに注意)

方法：Wilcoxon の順位和検定

```
df1 <- read_excel("Green1-3.xlsx", sheet = "3-2gun-nonpara1")
```

因子型

```
wilcox_test(y ~ factor(group), data = df1,  
            alternative = "two.sided", distribution = "asymptotic")
```

```
##  
## Asymptotic Wilcoxon-Mann-Whitney Test  
##  
## data: y by factor(group) (A, B)  
## Z = 2.0207, p-value = 0.04331  
## alternative hypothesis: true mu is not equal to 0
```

group	y
A	121
A	118
A	110
A	90
B	95
B	34
B	22
B	12

Wilcoxon の順位和検定

p.171

● 表示3.7.3 JMP による順位和検定

スクリプトファイル：Green1-3-7a.R

利用した関数：wilcox.test、coin::wilcox_test (「.」 「_」の違いに注意)

方法：Wilcoxon の順位和検定

```
df1 <- read_excel("Green1-3.xlsx", sheet = "Sheet1")
```

因子型

```
wilcox_test(y ~ factor(group), data = df1,  
            alternative = "two.sided", distribution = "asymptotic")
```

```
##  
## Asymptotic Wilcoxon-Mann-Whitney Test  
##  
## data: y by factor(group) (A, B)  
## Z = 2.0207, p-value = 0.04331  
## alternative hypothesis: true mu is not equal to 0
```

"asymptotic"
：漸近分布による計算、デフォルト
"approximate"
：モンテカルロ近似による計算
"exact" : 正確な計算

表示 3.7.3 と一致
近似精度がよくないとの
警告あり

group	y
A	121
A	118
A	110
A	90
B	95
B	34
B	22
B	12

Wilcoxon の順位和検定

● 表示3.7.3 JMP による順位和検定

スクリプトファイル：Green1-3-7a.R

利用した関数：wilcox.test、coin::wilcox_test (「.」 「_」 の)

方法：Wilcoxon の順位和検定

```
wilcox_test(y ~ factor(group), data = df1,  
            alternative = "two.sided", distribution = "approximate")
```

```
## Approximative Wilcoxon-Mann-Whitney Test  
## data: y by factor(group) (A, B)  
## Z = 2.0207, p-value = 0.0531
```

```
wilcox_test(y ~ factor(group), data = df1,  
            alternative = "two.sided", distribution = "exact")
```

```
## Exact Wilcoxon-Mann-Whitney Test  
##  
## data: y by factor(group) (A, B)  
## Z = 2.0207, p-value = 0.05714
```

"asymptotic"
：漸近分布による計算、デフォルト
"approximate"
：モンテカルロ近似による計算
"exact" : 正確な計算

group	y
A	121
A	118
A	110
A	90
B	95
B	34
B	22
B	12

正確な p 値 ([PDF ファイル](#))
この結果を利用

Wilcoxon の順位和検定 (n が不揃い、同値がある場合)

p.177

- 表示3.7.7 JMP によるノンパラメトリック検定、表示3.7.8 Excelによるノンパラメトリック検定
スクリプトファイル：Green1-3-7a.R、利用した関数：coin::wilcox_test

```
wilcox_test(y ~ factor(group), data = df2,  
            ties.method = "mid-ranks", distribution = "asymptotic")
```

```
##
```

```
## Asymptotic Wilcoxon-Mann-Whitney Test
```

```
## data: y by factor(group) (A, B)
```

```
## Z = -2.4447, p-value = 0.0145
```

表示 3.7.7 と一致

```
wilcox_test(y ~ factor(group), data = df2,  
            ties.method = "mid-ranks", distribution = "exact")
```

```
##
```

```
## Exact Wilcoxon-Mann-Whitney Test
```

```
##
```

```
## data: y by factor(group) (A, B)
```

```
## Z = -2.4447, p-value = 0.01255
```

正確な p 値
この結果を利用

Van der Waerden の検定

● 表示3.7.6 JMP による Van der Waerden 検定

スクリプトファイル：Green1-3-7a.R

利用した関数：coin::normal_test

```
## normal_test(y ~factor(group), data = df1,  
              distribution = "asymptotic",  
              ties.method = "mid-ranks")  
  
## Asymptotic Two-Sample van der Waerden (Normal Quantile) Test  
## data: y by factor(group) (A, B)  
## Z = 1.9944, p-value = 0.04611
```

"asymptotic"
：漸近分布による計算、デフォルト
"approximate"
：モンテカルロ近似による計算
"exact" : 正確な計算

"mid-ranks"
：漸近分布による計算、デフォルト
"average-scores"

表示 3.7.6 と一致

```
normal_test(y ~factor(group), data = df1,  
            distribution = "exact", ties.method = "mid-ranks")  
  
## Exact Two-Sample van der Waerden (Normal Quantile) Test  
## data: y by factor(group) (A, B)  
## Z = 1.9944, p-value = 0.05714
```

正確な p 値
この結果を利用

Van der Waerden の検定 (n が不揃い、同値がある場合)

p.177

- 表示3.7.7 JMP によるノンパラメトリック検定、表示3.7.8 Excelによるノンパラメトリック検定

スクリプトファイル：Green1-3-7a.R

利用した関数：coin::normal_test

```
normal_test(y ~ factor(group), data = df2,  
            distribution = "asymptotic", ties.method = "mid-ranks")
```

```
##
```

```
## Asymptotic Two-Sample van der Waerden (Normal Quantile) Test
```

```
## data: y by factor(group) (A, B)
```

```
## Z = -2.4575, p-value = 0.01399
```

```
## alternative hypothesis: true mu is not equal to 0
```

```
normal_test(y ~ factor(group), data = df2,  
            distribution = "exact", ties.method = "mid-ranks")
```

```
##
```

```
## Exact Two-Sample van der Waerden (Normal Quantile) Test
```

```
## data: y by factor(group) (A, B)
```

```
## Z = -2.4575, p-value = 0.01044
```

表示 3.7.7 と一致

正確な p 値
この結果を利用

Wilcoxon の符号付順位検定

- 表示3.7.10 対応のある順位検定 (Wilcoxon の符号付順位検定)

スクリプトファイル: Green1-3-7a.R

利用した関数: coin::wilcoxsign_test (アンダースコアに注意)

方法

```
df4 <- read_excel("Green1-3.xlsx", sheet = "3-2gun2")
```

```
wilcoxsign_test(df4$after ~ df4$before,  
               zero.method = "Wilcoxon",  
               distribution = "exact")
```

```
##  
## Exact Wilcoxon Sign  
##  
## data: y by x (pos,  
## stratified by bloc  
## Z = 2.3805, p-value = 0.01562  
## alternative hypothesis: true mu is not equal to 0
```

対応するデータを
チルダで結ぶ (注意)

差が 0 になる対の処理方法
"Pratt": 順位変換後に除去
(デフォルト)
"Wilcoxon": 除去後に順位変換

対応がある

ID	before	after
1	3.31	3.39
2	2.87	3.29
3	3.09	3.2
4	2.93	3.21
5	3.18	3.17
6	3.02	3.09
7	2.95	3.17
8	3.05	3.09

Wilcoxon の符号付順位検定

- 表示3.7.10 対応のある順位検定 (Wilcoxon の符号付順位検定)

スクリプトファイル: Green1-3-7a.R

利用した関数: coin::wilcoxsign_test (アンダースコアに注意)

方法:

```
df4 <- read_excel("Green1-3.xlsx", sheet = "3-2gun2")
```

```
wilcoxsign_test(df4$after ~ df4$before,  
               zero.method = "Wilcoxon",  
               distribution = "exact")
```

```
##  
## Exact Wilcoxon Signed-Rank Test  
##  
## data: y by x (pos, neg)  
## stratified by block  
## Z = 2.3805, p-value = 0.01562  
## alternative hypothesis: true mu is not equal to 0
```

"asymptotic"
: 漸近分布による計算、デフォルト
"approximate"
: モンテカルロ近似による計算
"exact" : 正確な計算

表示 3.7.10 と一致
(テキスト p.180 脚注を参照)

ID	before	after
1	3.31	3.39
	2.87	3.29
	3.09	3.2
	2.93	3.21
	3.18	3.17
	3.02	3.09
	2.95	3.17
8	3.05	3.09

●表示3.7.11 データと JMP 出力 ~ 表示3.7.14 ROC 曲線

スクリプトファイル：Green1-3-7b.R

利用したパッケージ

ROCR、**pROC**、**PRROC**、**plotROC**

precrec、**ROCit**、**Epi**、**epicalc**

それぞれ、特徴がある

参照 「Some R Packages for ROC Curves」

(Joseph Rickert, 2019)

方法

群 (sick) と検査値 (obs) の2変量

22組のデータ

群 (sick) は癌/非癌 (1/0) の二値

検査値(obs)は数値データで同値あり

癌/非癌
1/0

検査値
(観測値)

同値あり

全部で22組

```
df <- read_excel("Green1-3.xlsx",
                  sheet = "3-ROC")
df <- data.frame(df)
df
##      sick obs
## 1      0  14
## 2      0  20
## 3      0  32
## 4      1  36
## 5      0  37
## 6      0  40
## 7      1  55
## 8      0  55
. . . . .
## 21     1 278
## 22     1 342
```

ROC 曲線

●表示3.7.11 データと JMP 出力

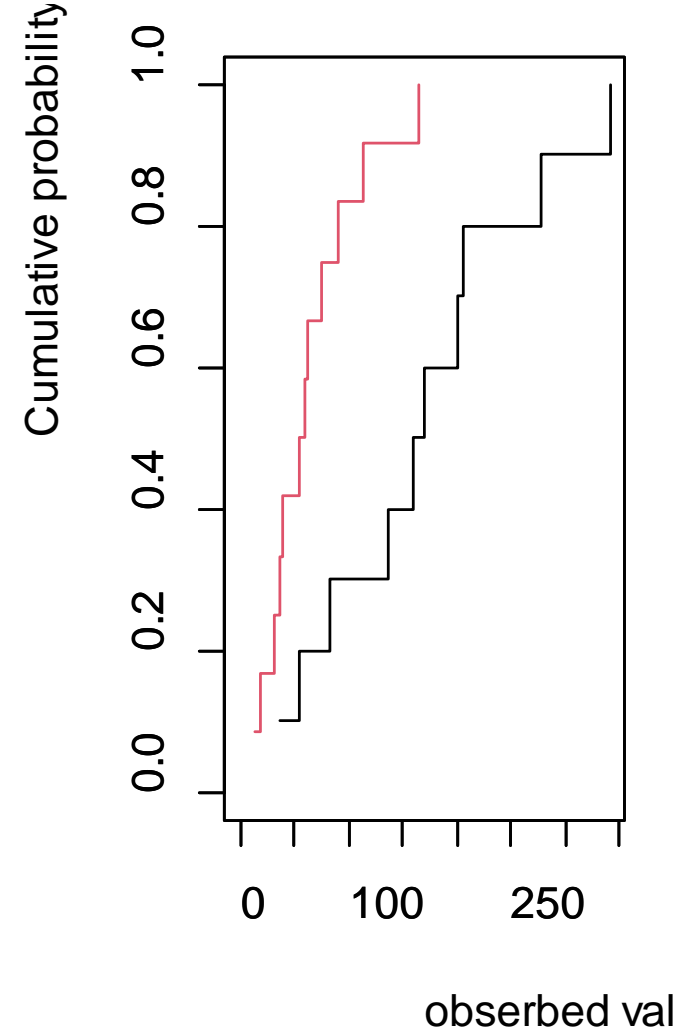
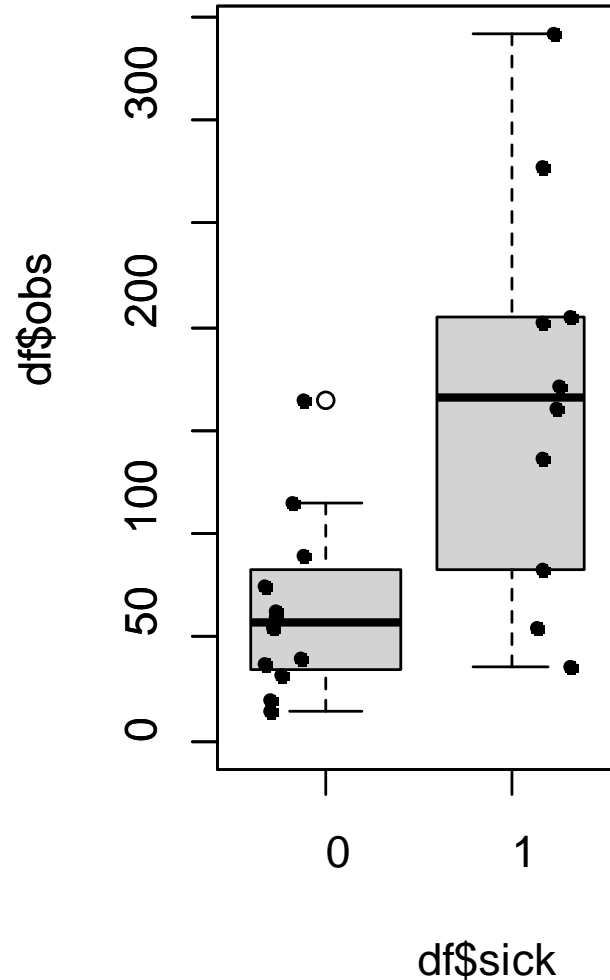
スクリプトファイル：Green1-3-7b.R

利用した関数

boxplot、stripchart、plot、sort、rank
方法

[§3.1](#) 参照

##	sick	obs
## 1	0	14
## 2	0	20
## 3	0	32
## 4	1	36
## 5	0	37
## 6	0	40
## 7	1	55
## 8	0	55
.		
## 21	1	278
## 22	1	342



● ROC テーブル

スクリプトファイル

Green1-3-7b.R

利用した関数

ROCR::prediction、

ROCR::performance、

unlist、ifelse

22 の要素
からなるベクトル

注) 「@」は、

S4 クラスで定義した

スロットにアクセスする

ための演算子

```
# ROC の計算
```

```
pred <- prediction(df$obs, df$sick)
```

オブジェクト @ スロット

```
# ROC テーブル
```

```
cuttoff <- unlist(pred@cuttoffs) # 基準値
tp <- unlist(pred@tp) # true positive 真陽性
tn <- unlist(pred@tn) # true negative 真陰性
fp <- unlist(pred@fp) # false positive 偽陽性
fn <- unlist(pred@fn) # false negative 偽陰性
sensitivity <- tp / (tp + fn) # 感度
specificity <- tn / (fp + tn) # 特異度
fpf <- 1 - specificity # 偽陽性率
accuracy <- (tp + tn) / nrow(df) # 正確度
youden <- sensitivity - fpf
youdenMax <- ifelse(youden == max(youden), "*", "")
```

● ROC テーブル

スクリプトファイル

Green1-3-7b.R

利用した関数

ROCR::prediction、

unlist、ifelse

22 の要素
からなるベクトル

ベクトル演算

```
# ROC の計算
pred <- prediction(df$obs, df$sick)

# ROCテーブル
cutoff      <- unlist(pred@cutoffs) # 基準値、カットオフ値
tp          <- unlist(pred@tp)      # true positive 真陽性
tn          <- unlist(pred@tn)      # true negative 真陰性
fp          <- unlist(pred@fp)      # false positive 偽陽性
fn          <- unlist(pred@fn)      # false negative 偽陰性
sensitivity <- tp / (tp + fn)        # 感度
specificity <- tn / (fp + tn)        # 特異度
fpf         <- 1 - specificity        # 偽陽性率
accuracy    <- (tp + tn) / nrow(df) # 正確度
youden      <- sensitivity - fpf     # youden index
youdenMax   <- ifelse(youden == max(youden), "*", "")
```

● ROC テーブル

スクリプトファイル : Green1-3-7b.R、利用した関数 : ROCR::prediction、unlist、ifelse

##	Cutoff	FPF	Sensitivity	Specificity	YoudenIndex	Max	Accuracy	TP	TN	FP	FN
## 1	Inf	0.00000	0.0	1.00000	0.00000		0.5455	0	12	0	10
## 2	342	0.00000	0.1	1.00000	0.10000		0.5909	1	12	0	9
## 3	278	0.00000	0.2	1.00000	0.20000		0.6364	2	12	0	8
								
## 7	165	0.08333	0.5	0.91667	0.41667		0.7273	5	11	1	5
## 8	160	0.08333	0.6	0.91667	0.51667		0.7727	6	11	1	4
## 9	137	0.08333	0.7	0.91667	0.61667	*	0.8182	7	11	1	3
## 10	115	0.16667	0.7	0.83333	0.53333		0.7727	7	10	2	3
								
## 20	32	0.83333	1.0	0.16667	0.16667			2	10	0	0
## 21	20	0.91667	1.0	0.08333	0.08333		0.5000	10	1	11	0
## 22	14	1.00000	1.0	0.00000	0.00000		0.4545	10	0	12	0
##	max(accuracy) = 0.8182										

*
Youden index
の最大値

● 表示3.7.14 ROC曲線

スクリプトファイル

Green1-3-7b.R

利用した関数

ROCR::prediction、
ROCR::performance、
plot

方法

performance 関数に
prediction 関数の結果
を渡し、

縦軸と横軸を略語で指定

acc:Accuracy、err>Error rate、fpr:False positive rate、fall:Fallout、tpr:True positive rate、
rec:Recall、sens:Sensitivity など多数 (performance 関数のヘルプを参照)

```
# ROC の計算
```

```
pred <- prediction(df$obs, df$sick)
```

```
perf1 <- performance(pred, measure = "tpr", x.measure = "fpr")
```

```
plot(perf1, colorize = TRUE, lwd = 2) # ROC曲線
```

```
perf2 <- performance(pred, measure = "sens", x.measure = "spec")
```

```
plot(perf2, lwd = 2)
```

true positive rate
真陽性率

false positive rate
偽陽性率

sensitivity
感度

specificity
特異度

ROC 曲線

● 表示3.7.14 ROC曲線

スクリプトファイル

Green1-3-7b.R

利用した関数

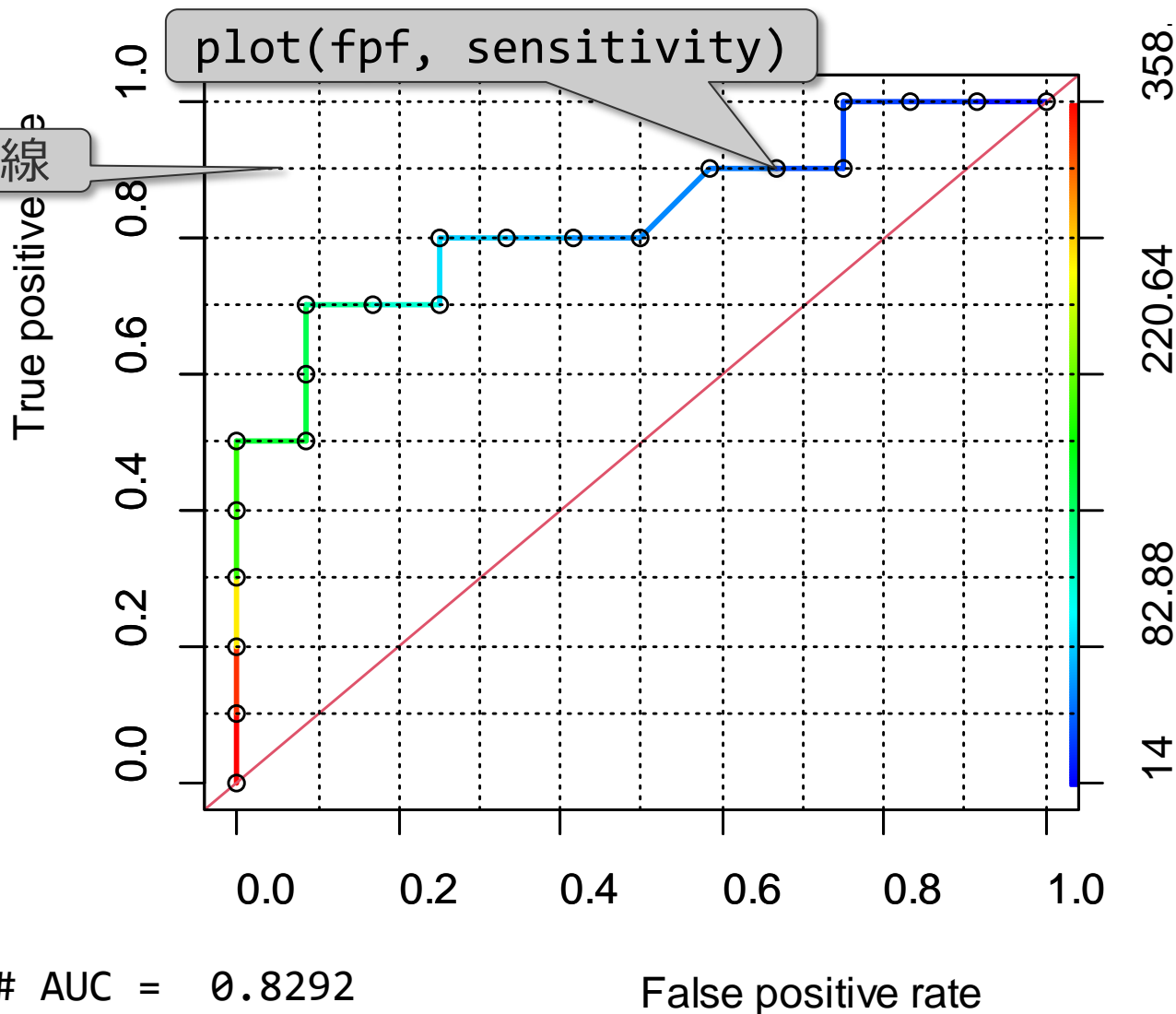
ROCR::performance、plot、abline

```
perf1 <- performance(pred,  
  measure = "tpr", x.measure = "fpr")
```

```
plot(perf1, colorize = TRUE, lwd = 2)  
abline(a = 0, b = 1, col = 2)  
abline(h = (1:10) / 10,  
  v = (1:10) / 10,  
  lty = 3, lwd = 1)
```

```
par(new = TRUE)  
plot(fpf, sensitivity,  
  xlab = "", ylab = "")
```

上書き許可



ROC 曲線

- 表示3.7.14 ROC曲線

スクリプトファイル

Green1-3-7b.R

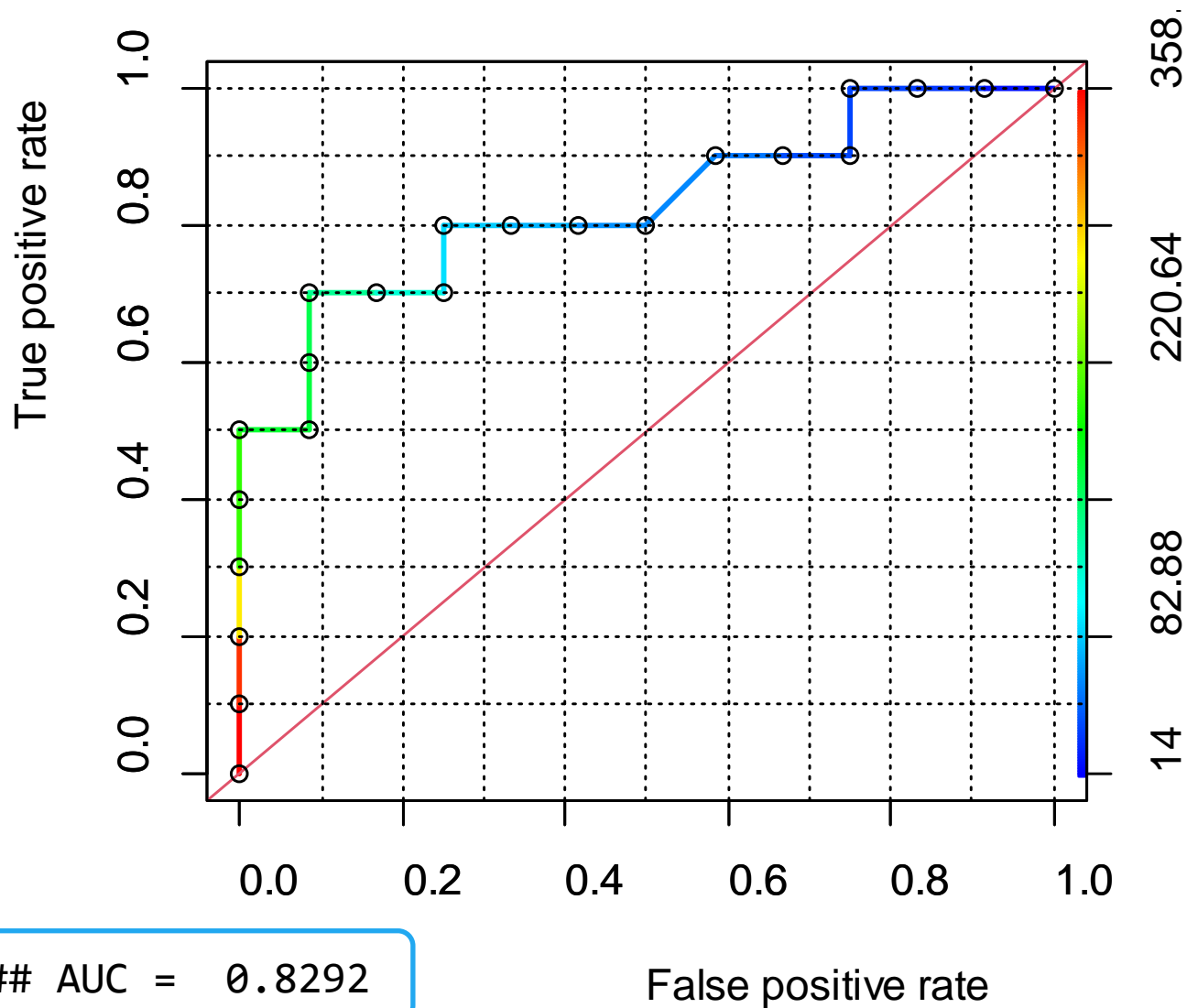
利用した関数

ROCR::performance

as.numeric、cat

```
# AUC value
auc_out <- performance(pred, "auc")
auc <- as.numeric(auc_out@y.values)
cat("AUC = ", auc, "\n")

## AUC = 0.8292
```



AUC = 0.8292

ROC 曲線

p.182

● 感度特異度曲線

スクリプトファイル

Green1-3-7b.R

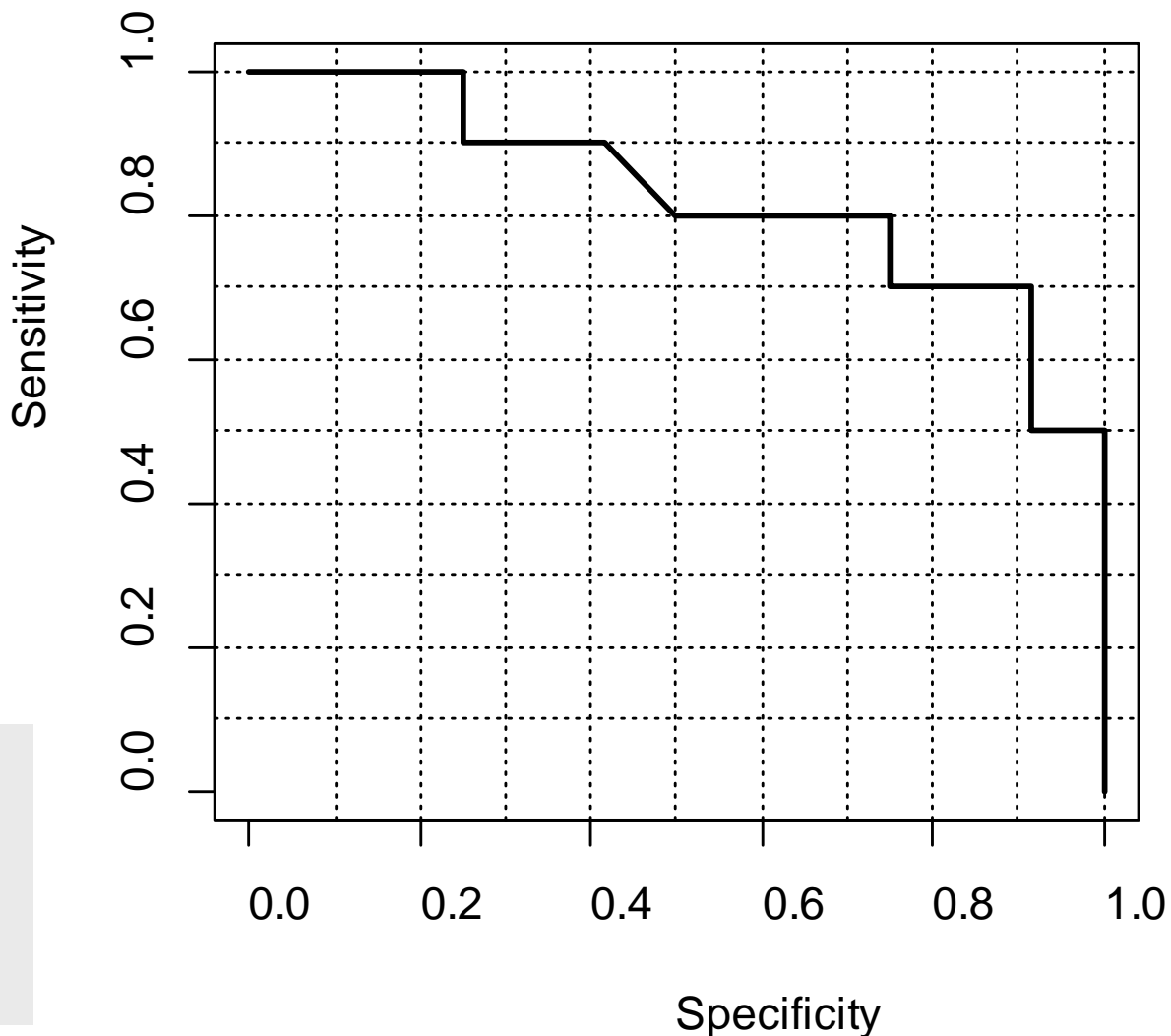
利用した関数

ROCR::performance

ROCR::prediction

plot、abline

```
perf2 <- performance(pred, "sens", "spec")
plot(perf2, lwd = 2)
abline(h = (1:10)/10,
       v = (1:10) / 10, lty = 3)
```



- Precision-Recall曲線

スクリプトファイル

Green1-3-7b.R

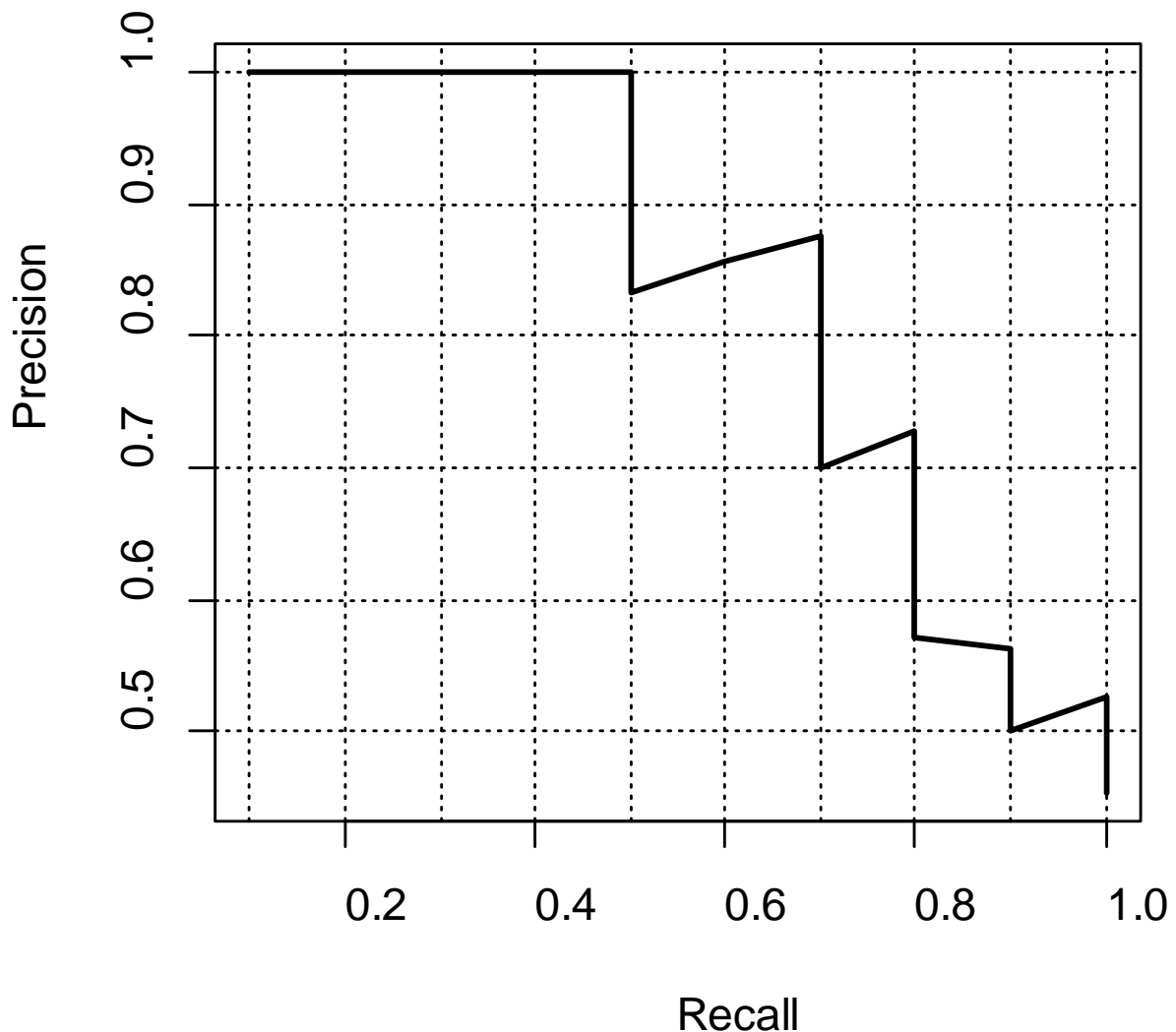
利用した関数

ROCR::performance

ROCR::prediction

plot、abline

```
perf2 <- performance(pred, "prec", "rec")
plot(perf2, lwd = 2)
abline(h = (1:10)/10,
       v = (1:10) / 10, lty = 3)
```



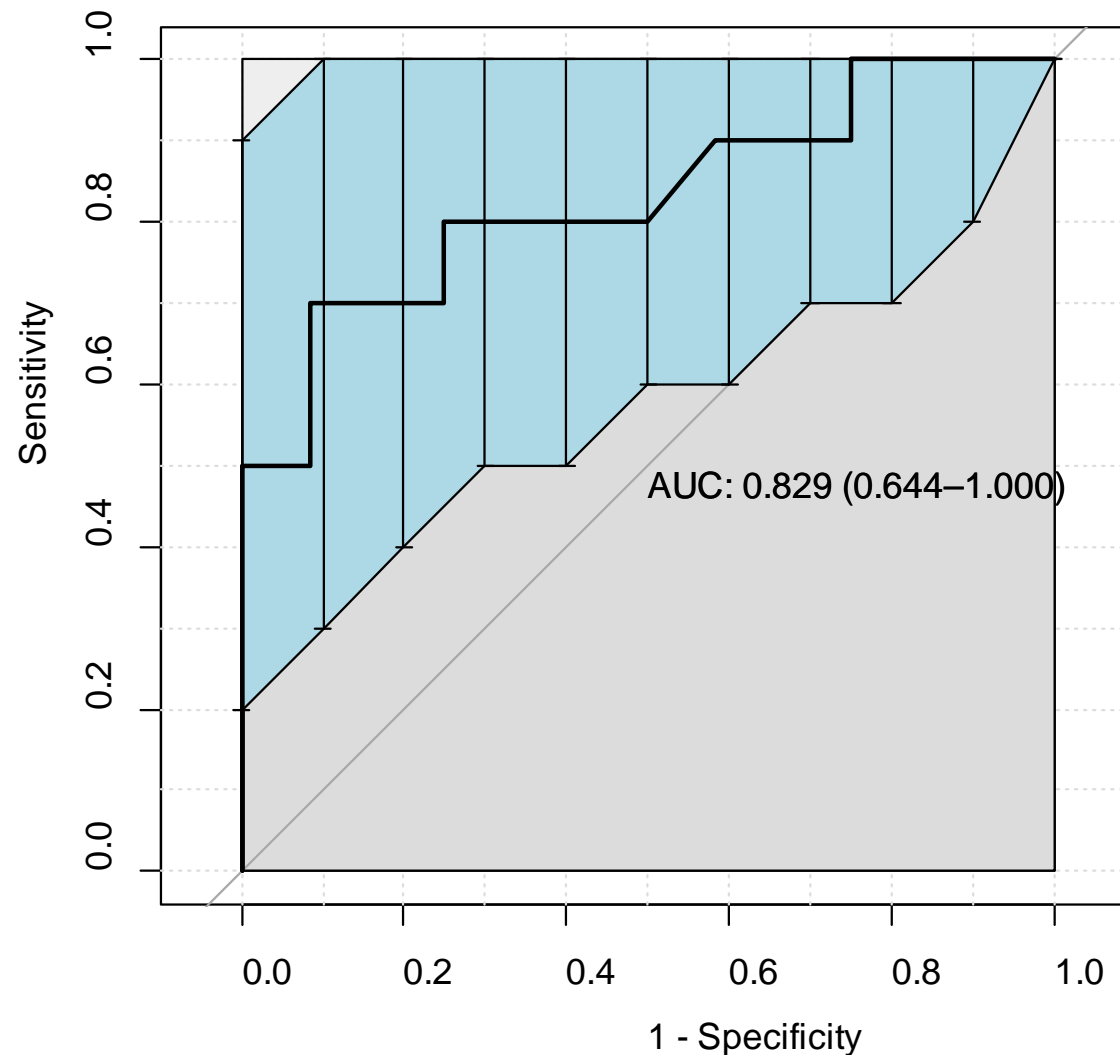
● 表示3.7.14 ROC曲線

スクリプトファイル：Green1-3-7b.R

利用した関数：pROC::roc

方法

```
pROC_out <- roc(df$sick, df$obs,  
               smoothed = FALSE,  
               ci = TRUE,  
               ci.alpha = 0.9,  
               stratified = FALSE,  
               plot = TRUE,  
               auc.polygon = TRUE,  
               max.auc.polygon = TRUE,  
               grid = TRUE,  
               print.auc = TRUE,  
               show.thres = TRUE,  
               legacy.axes = TRUE)
```





- 参考文献

Sing T, Sander O, Beerenwinkel N, Lengauer T. ([2005)

ROCR: visualizing classifier performance in R.

Bioinformatics 21(20):3940-1.

Rickert J. (2019) Some R Packages for ROC Curves.

[https://rviews.rstudio.com/2019/03/01/](https://rviews.rstudio.com/2019/03/01/some-r-packages-for-roc-curves/)

[some-r-packages-for-roc-curves/](https://rviews.rstudio.com/2019/03/01/some-r-packages-for-roc-curves/)

- 作成 片瀬雅彦

- 作成時期 2021年9月5日