2023 年 4 月 17 日

『多変量解析（データサイエンス大系）』

（松井秀俊 著，学術図書出版社）

章末問題解答例

第 2 章

2-1 (a)

(a) 線形単回帰モデルは，1 つの目的変数と，それに関連すると考えられる 1 つの説明変数との関係をモデル化したもので，正しい．(b) 観測された値とは大幅に異なる値を説明変数に用いた場合は外挿になり，目的変数の適切な予測ができない場合があるので誤り．(c) 線形単回帰モデルが観測値に完全に当てはまっている場合は，決定係数は 1 になるので誤り．(d) 回帰係数の推定値が求まったことで，説明変数と目的変数間の相関関係は導けるかもしれないが，因果関係までは明らかにできないので誤り．

2-2 (2.8) 式を $f(\beta_0, \beta_1, \sigma^2)$ とおく．$f(\beta_0, \beta_1, \sigma^2)$ を $\beta_0, \beta_1, \sigma^2$ でそれぞれ偏微分することで，次を得る．

$$\frac{\partial f(\beta_0, \beta_1, \sigma^2)}{\partial \beta_0} = \frac{1}{\sigma^2} \sum_{i=1}^{n} \{y_i - (\beta_0 + \beta_1 x_i)\} = 0, \tag{1}$$

$$\frac{\partial f(\beta_0, \beta_1, \sigma^2)}{\partial \beta_1} = \frac{1}{\sigma^2} \sum_{i=1}^{n} x_i \{y_i - (\beta_0 + \beta_1 x_i)\} = 0, \tag{2}$$

$$\frac{\partial f(\beta_0, \beta_1, \sigma^2)}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_{i=1}^{n} \{y_i - (\beta_0 + \beta_1 x_i)\}^2 = 0. \tag{3}$$

(1), (2) 式を書き換えることで，それぞれ次を得る．

$$\bar{y} - (\beta_0 + \bar{x}\beta_1) = 0, \tag{4}$$

$$\frac{1}{n} \sum_{i=1}^{n} x_i y_i - \left(\bar{x}\beta_0 + \frac{1}{n} \sum_{i=1}^{n} {x_i}^2 \beta_1 \right) = 0. \tag{5}$$

(4), (5) 式から β_0 を消去することで，次を得る．

$$S_{xy} - S_{xx}\beta_1 = 0, \quad \beta_1 = \widehat{\beta}_1 = \frac{S_{xy}}{S_{xx}}. \tag{6}$$

(6) 式を (4) 式に代入して，

$$\beta_0 = \widehat{\beta}_0 = \overline{y} - \widehat{\beta}_1 \overline{x}.$$

これで β_0, β_1 の最尤推定量が得られた．

最後に，σ^2 の最尤推定量は，(3) 式を σ^2 について解き，さらに $\widehat{\beta}_0, \widehat{\beta}_1$ を代入することで次のように得られる．

$$\widehat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} \left\{ y_i - \left(\widehat{\beta}_0 + \widehat{\beta}_1 x_i \right) \right\}^2.$$

2-3 残差は $r_i = y_i - \widehat{y}_i$ で与えられることを考えると，元の y_i は，上に凸な放物線状に分布したデータと考えられる．

2-4 次のプログラム

```
res = lm(Employed~Unemployed, data=longley)
summary(res)
```

を実行することで，回帰係数の推定値として $\widehat{\beta}_0 = 59.287$, $\widehat{\beta}_1 = 0.019$, 決定係数 $R^2 = 0.253$ を得る．

2-5 残差プロットの表示は省略．1 つ目は，残差がランダムに分布しているため妥当と考えられる．2 つ目は，残差が放物線状に分布しているため不適切と考えられる．3 つ目は，外れ値の影響で他のデータの残差が直線状の傾向を持っているため不適切と考えられる．4 つ目は，説明変数の値が異なる点が 1 点しかないことから，そもそも線形単回帰モデルを適用することが不適切と考えられる．

第 3 章

3-1 (d)

(a) 線形重回帰モデルは，ある説明変数の値が変化したとき，他の説明変数でなく目的変数の値の変化量を定量化するために用いられるものなので，誤り．(b) 標準化されていないデータについては，必ずしも回帰係数の大きさの順とは限らないため，誤り．(c) サンプルサイズが説明変数の数よりも大きくても，多重共線性があれば最小二乗推定値を計算できない．(d) 実際には目的変数と関連のない説明変数が加わっても決定係数は大きくなるため，適切に変数選択を行うためには自由度調整済み決定係数

や AIC が用いられるため，適切である．

3-2 $p=1$ の場合，行列 X は次で与えられる．

$$X=\begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}.$$

これより，$X^\top X$ とその逆行列はそれぞれ次で与えられる．

$$X^\top X=\begin{pmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n {x_i}^2 \end{pmatrix}$$

$$\begin{aligned}\left(X^\top X\right)^{-1} &= \frac{1}{n\sum_{i=1}^n {x_i}^2-\left(\sum_{i=1}^n x_i\right)^2}\begin{pmatrix} \sum_{i=1}^n {x_i}^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{pmatrix} \\ &= \frac{1}{S_{xx}}\begin{pmatrix} \sum_{i=1}^n {x_i}^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{pmatrix}.\end{aligned}$$

また，$X^\top \boldsymbol{y}=\begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{pmatrix}$ より，これらを合わせて最小二乗推定量 $\widehat{\boldsymbol{\beta}}$ は次のように計算される．

$$\begin{aligned}\widehat{\boldsymbol{\beta}} &= \left(X^\top X\right)^{-1}X^\top \boldsymbol{y} \\ &= \frac{1}{S_{xx}}\begin{pmatrix} \left(\sum_{i=1}^n {x_i}^2\right)\left(\sum_{i=1}^n y_i\right)-\left(\sum_{i=1}^n x_i\right)\left(\sum_{i=1}^n x_i y_i\right) \\ -\left(\sum_{i=1}^n x_i\right)\left(\sum_{i=1}^n y_i\right)+n\sum_{i=1}^n x_i y_i \end{pmatrix} \\ &= \frac{1}{S_{xx}}\begin{pmatrix} n\overline{y}(S_{xx}+n\overline{x}^2)-n\overline{x}(S_{xy}+n\overline{x}\overline{y}) \\ nS_{xy} \end{pmatrix} \\ &= \frac{1}{S_{xx}}\begin{pmatrix} n\overline{y}S_{xx}-n\overline{x}S_{xy} \\ nS_{xy} \end{pmatrix}\end{aligned}$$

$$= \begin{pmatrix} \overline{y} - \overline{x}\dfrac{S_{xy}}{S_{xx}} \\ \dfrac{S_{xy}}{S_{xx}} \end{pmatrix}.$$

以上より，$\widehat{\beta}_0, \widehat{\beta}_1$ は (2.5) 式に一致する．

3-3 次のようなプログラムを実行することで確認できる．

```
1 data(allometry)
2 X = as.matrix(allometry[, c(2,3,4)])
3 X = cbind(1, X)
4 y = allometry[, 5]
5 bhat = solve(t(X)%*%X)%*%t(X)%*%y
6 bhat
```

3-4 次のプログラム

```
1 result = lm(Employed~., data=longley)
2 step(result)
```

を実行することで，選択された変数の組合せと AIC の値は次のようになる．

`GNP.deflator`, `GNP`, `Unemployed`, `Armed.Forces`, `Population`, `Year`, AIC $= -33.22$

`GNP`, `Unemployed`, `Armed.Forces`, `Population`, `Year`, AIC $= -35.16$

`GNP`, `Unemployed`, `Armed.Forces`, `Year`, AIC $= -36.8$

3-5 次のプログラムを実行する．

```
1 lm(y~X)
```

問題のプログラム 2 行目で用いられる乱数によって結果が異なるため，出力結果は省略する．上のプログラムより出力される回帰係数の推定値と，問題のコード 4 行目の値（真の回帰係数）を比較すると，真の回帰係数をよく近似できていることがわかる．

第 4 章

4-1 (a)

(a) ダミー変数は，値に順序関係が発生しないように，2 つの数値のみで表されるので正しい．(b) ロジスティック回帰モデルは，目的変数が連続

値ではなく二項分布に従う確率変数である場合に用いられるため，誤り．(c) ニュートン-ラフソン法は，ロジスティック回帰モデルを最尤法で推定する場合に用いられるアルゴリズムなので，誤り．(d) AIC の計算で用いられる対数尤度関数の値はモデルの種類によって異なるため，誤り．

4-2 (4.2) 式より

$$\pi + \pi \exp(\beta_0 + \beta_1 x_1) = \exp(\beta_0 + \beta_1 x),$$

$$\frac{\pi}{1-\pi} = \exp(\beta_0 + \beta_1 x_1)$$

となる．この式の両辺に自然対数（log）をとればよい．

4-3 ソースコード 4.1 の 43 行目の 0.5 という値を変えることで出力される値を見る．0.5 より大きければ予測値は 0 となりやすく，0.5 より小さければ 1 となりやすくなる．

4-4 対数尤度を計算する場合は，確率関数 $f(y_i \,|\, x_i, m_i; \beta_0, \beta_1)$ の対数をとったものについて和をとると計算しやすい．

$$\begin{aligned}
\log L(\beta_0, \beta_1) &= \sum_{i=1}^{n} \log f(y_i \,|\, x_i, m_i; \beta_0, \beta_1) \\
&= \sum_{i=1}^{n} \log \left\{ {}_{m_i}\mathrm{C}_{y_i} \pi_i^{y_i} (1-\pi_i)^{m_i - y_i} \right\} \\
&= \sum_{i=1}^{n} \log {}_{m_i}\mathrm{C}_{y_i} + \sum_{i=1}^{n} y_i \log \pi_i + \sum_{i=1}^{n} (m_i - y_i) \log(1-\pi_i).
\end{aligned}$$

最後の式は厳密には β_0, β_1 の関数にはなっていないので，

$\pi_i = \dfrac{\exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0 + \beta_1 x_i)}$ を代入する．すると，

$$\begin{aligned}
\log L(\beta_0, \beta_1) &= \sum_{i=1}^{n} \log {}_{m_i}\mathrm{C}_{y_i} + \sum_{i=1}^{n} y_i [\beta_0 + \beta_1 x_i - \log\{1 + \exp(\beta_0 + \beta_1 x_i)\}] \\
&\quad + \sum_{i=1}^{n} (m_i - y_i)[-\log\{1 + \exp(\beta_0 + \beta_1 x_i)\}] \\
&= \sum_{i=1}^{n} \log {}_{m_i}\mathrm{C}_{y_i} + \sum_{i=1}^{n} y_i (\beta_0 + \beta_1 x_i) - \sum_{i=1}^{n} m_i \log\{1 + \exp(\beta_0 + \beta_1 x_i)\}.
\end{aligned}$$

4-5 次のプログラムを実行することで，目的変数の予測値を 0, 1 として出力できる．

```
result = glm(location~., family="binomial", data=icecream)
phat = predict(result, type="response")
as.numeric(phat > 0.5)
```

予測値は 0 ならば Dappermarkt，1 ならば Oosterpark に対応する．

第 5 章

5-1 (c)

(a) 正解率が 1 に近くても，場合によっては感度や特異度が非常に小さい場合があり分類結果としては不適切な場合があるので，誤り．(b) フィッシャーの線形判別分析では，群間変動が大きく，群内変動が小さくなるような方向を求めるものなので，誤り．(c) 各群で共通の分散共分散行列を用いた場合は，マハラノビス距離に基づく分類ルールとフィッシャーの判別関数による分類ルールは一致するので，正しい．(d) 非線形判別は線形判別よりも柔軟な決定境界を構築できるが，感度や特異度といった分類指標が必ずよくなるとは限らないので誤り．

5-2 ［**第 1 版第 1 刷の注**：問題文を次のように修正します．
表 5.5 の 2 つの混同行列から，偽陽性率，偽陰性率，F 値をそれぞれ求めよ．］

線形判別：FPR $= 0.213$, FNR $= 0.045$, F $= 0.912$.

2 次判別：FPR $= 0.045$, FNR $= 0.246$, F $= 0.845$.

5-3 ［**第 1 版第 1 刷の注**：カーネル判別は実行時にエラーが出るため除外します．］
次のプログラムを実行することで，線形判別と 2 次判別それぞれの混同行列を出力できる．

```
X = as.matrix(X)
y = as.numeric(y)
# 線形判別実行
result.lda = lda(X, y)
pred.lda = predict(result.lda)
table(y, pred.lda$class)
# 2次判別実行
result.qda = qda(X, y)
pred.qda = predict(result.qda)
table(y, pred.qda$class)
```

混同行列は，それぞれ次のようになる．

	$\widehat{Y}=1$	$\widehat{Y}=0$
$Y=1$	50	0
$Y=0$	0	50

	$\widehat{Y}=1$	$\widehat{Y}=0$
$Y=1$	50	0
$Y=0$	0	50

5-4 ［**第 1 版第 1 刷の注**：問題を次のものに差し替えてください．
［R 使用］ ソースコード 5.1 の `x1`, `x2` に対して，次のプログラムを実行することでそれぞれの標本平均，標本不偏分散共分散行列を計算できる．

```
ma = apply(x1, 2, mean)  # x1の標本平均ベクトル
mb = apply(x2, 2, mean)  # x2の標本平均ベクトル
Sa = var(x1)  # x1の標本不偏分散共分散行列
Sb = var(x2)  # x2の標本不偏分散共分散行列
```

このことを利用して，がく片の長さが 5.5，がく片の幅が 3.0 の観測値に対してフィッシャーの線形判別関数 (5.9) 式の値を R で計算せよ．］

次のプログラムを実行することで，観測値 $(5.5, 3.0)^\top$ に対するフィッシャーの線形判別関数の値を求めることができる．

```
z = c(5.5, 3.0)
Sw = 1/(50+50-2)*((50-1)*Sa + (50-1)*Sb)
h0 = t(ma-mb) %*% solve(Sw) %*% (z - (ma+mb)/2)
```

`h0` の値は -1.73 となる．この値は負なので，この観測値は versicolor（`x2` の群）と分類される．

5-5 ソースコード 5.5 の 7 行目の `lda` を `qda` に変更すればよい．品種 virginica を真とみなしたときの偽陽性率は 0.02，偽陰性率は 0.02 となる．

第 6 章

6-1 (d)

(a) 線形分離可能な場合は，データを完全に分離できるような直線は一般的に無数に存在するため，誤り．(b) サポートベクターマシンは，分離超平面から最も近い観測値までの距離を最大化することを目的としているため，誤り．(c) スラック変数の総和は大きすぎても小さすぎても，分類性能は悪化する可能性があるため，誤り．(d) カーネル SVM を適用することで，高次元空間上で SVM を実行し，結果として元の空間では非線形な決定境界が得られるので，正しい．

6-2 超平面上の任意の 2 点を $\boldsymbol{a}, \boldsymbol{b}$ とすると，超平面の式より

$$w_0 + \boldsymbol{w}^\top \boldsymbol{a} = w_0 + \boldsymbol{w}^\top \boldsymbol{b} = 0,$$

$$\boldsymbol{w}^\top (\boldsymbol{b} - \boldsymbol{a}) = 0$$

となる．これは $\boldsymbol{b} - \boldsymbol{a}$ と $\boldsymbol{w}$ が直交することを意味してるため，超平面と $\boldsymbol{w}$ は直交する．

6-3 (6.4) 式の制約について両辺を d で割り，$\widetilde{w}_0, \widetilde{\boldsymbol{w}}$ で表すことで，(6.6) 式の制約を得る．また，(6.5) 式の $\widetilde{\boldsymbol{w}}$ の両辺についてベクトルの大きさ（ノルム）をとると，$\|\widetilde{\boldsymbol{w}}\| = \dfrac{1}{\|\boldsymbol{w}\|}\|\boldsymbol{w}\| = \dfrac{1}{d}$ となる．したがって，$\|\boldsymbol{w}\|$ の最小化は d の最大化と等価である．以上より，(6.6) 式の制約つき最適化問題を得る．

6-4 次のプログラムを実行することで，線形 SVM を適用できる．

```
library(lgrdata)
library(e1071)
data(icecream)

X = as.matrix(icecream[, c(1,2)])
y = icecream[, 3]
X = scale(X)

res = ksvm(X, y, kernel="vanilladot", C=10)
```

6-5 たとえば，次のプログラムにより線形 SVM とカーネル SVM を実行できる．

```
library(kernlab) #ksvm関数実行のため
library(mlbench) #mlbench.spiralsデータ読み込みのため

# 線形分離可能でないデータ
dat = mlbench.spirals(200,cycles=1,sd=0.05) # 200
    個の学習データを作成

x = dat$x #入力データ
y = dat$classes #ラベル

# 線形 SVM
linsvm = ksvm(y~x, kernel="vanilladot")
# 予測値出力
predict1 = predict(linsvm)
# 混同行列
```

```
table(y, predict1, dnn=c("正解", "予測"))

#     予測
#正解   1  2
#    1 50 50
#    2 50 50

# カーネル SVM
rbfsvm = ksvm(y~x, kernel="rbfdot", kpar=list(sigma=1))
# 予測値出力
predict2 = predict(rbfsvm)
# 混同行列
table(y, predict2, dnn=c("正解", "予測"))

#      予測
#正解    1   2
#    1 100   0
#    2   1  99
```

この例の場合，混同行列から線形 SVM の誤分類率は 0.5, カーネル SVM の誤分類率は 0.005 である．ただし，これは学習データに対する誤分類率であることに注意したい．

第 7 章

7-1 (d)

(a) 決定木とフィッシャーの線形判別は分類方式が異なるため，まったく同じデータであっても分類結果は異なるので誤り．(b) ジニ係数は，分割を行う特徴量やその境界を決定するために用いられる指標で，分割数の選択に用いられるものではないので誤り．(c) 決定木における樹形図が大きくなる（分割数が大きくなる）と，観測されているデータ（学習データ）に対する予測精度は高くなるが，新たに観測されるデータに対しては必ずしもそうならないので誤り．(d) ランダムフォレストは複数の複製データに対して決定木を適用しそれを融合するので，どのような条件分岐に基づいて予測が行われるかという解釈が難しくなる．よって正しい．

7-2 $X_1 = 30$ より右の小領域を R_1，左の小領域を R_2 とおくと，誤り率は $E_1 = 0, E_2 = \dfrac{1}{4}$，ジニ係数は $E_1 = 0, E_2 = \dfrac{3}{8}$ である．

7-3 樹形図を上から辿ると，virginica に分類される．

7-4 ［ヒント：予測結果の出力には `predict` 関数を使用し，その引数に `type="class"` を追加する．］

次のプログラムを実行することで，`predict` 関数によりそれぞれの方法による予測値が出力される．これを比較すればよい．

```
library(lgrdata)
library(tree)
library(randomForest)

# icecreamデータ
data(icecream)
# 決定木
fm = as.formula(location ~ ., icecream)
result1 = tree(fm, data=icecream)
plot(result1); text(result1)
predict(result1, type="class")

# ランダムフォレスト
result2 = randomForest(location ~ ., data = icecream)
predict(result2)
```

7-5 略．

第 8 章

8-1 (b)

(a) クラスター分析は一般的に特徴量に対応するデータを用いるもので，ラベルに対応するデータは必要ないので誤り．(b) 単連結は 2 つのクラスターのうち最も近い観測値間の距離で，完全連結は 2 つのクラスターのうち最も遠い観測値間の距離で求められるので，正しい．(c) 階層型クラスタリングではクラスター数 1 の場合からはじめ，クラスター数を 1 つずつ増やしていく方法なので，クラスター数を指定しておく必要はなく，誤り．(d) 標準化せずにクラスタリングを行うと，相対的にスケールの小さい特徴量の距離が適切に反映されない可能性があるので，誤り．

8-2 ユークリッド距離は $\sqrt{29}$，マンハッタン距離は 9 となる．ユークリッド距離の方が小さい．

8-3 ［**第 1 版第 1 刷の注**：問題文の `UScities` は正しくは `UScitiesD` です．］

次のプログラムを実行することで，ウォード法による階層型クラスタリン

グのデンドログラムが得られる.

```
res = hclust(UScitiesD, method="ward.D2")
plot(res)
```

クラスター数が 3 となるように分割すると，“Seattle, LosAngeles, San-Francisco” からなるクラスター，“Denver, Houston” からなるクラスター，“New York, Washington.DC, Atlanta, Chicago” からなるクラスターになる.

8-4 次のプログラムを入力し実行することで，アヤメのデータに対する階層型クラスタリングにより得られるデンドログラムが描画される.

```
distdata = dist(data)
res = hclust(distdata, method="ward.D2")
plot(res)
```

サンプルサイズが 150 と多いのでデンドログラムでは見づらいが，setosa は 1 つのクラスターにまとまっている一方で，versicolor と virginica は 2 つのクラスターで混在している様子が見える.

8-5 たとえば，次のプログラムにより，10 回の K-平均法の結果のうち `tot.withinss` が最小になるクラスタリング結果を出力できる.

```
library(cluster)
res_d2 = 1:10  # D2を格納する変数
res_center = matrix(nrow=75, ncol=10) #
     クラスター結果を格納する変数
# nstart=1として K-平均法を 10回繰り返す
for(k in 1:10){
  res = kmeans(ruspini, 4, nstart=1)
  res_center[, k] = res$cluster #クラスター番号格納
  res_d2[k] = res$tot.withinss  # D2の値を格納
}
idx = which.min(res_d2)  # tot.withinssが最小になった繰り返し番号
res_center[, idx]  # tot.withinssが最小になったクラスタリング結果
```

第 9 章

9-1 (a), (d)

(a) 主成分分析では，データを 1 次元に圧縮したときに，その分散が最大となる方向を求めるものであり，正しい．(b) 第 1 主成分得点の平均値

と，第 1 主成分のもつ情報の大きさに関連はないので誤り．(c) 寄与率は第 1 主成分が最大で，以降第 2，第 3 の順に小さくなるものなので，寄与率が最大となる主成分を選択するという方法は適切でなく誤り．(d) 主成分分析は，重要な情報を集約することができる方法なので，回帰分析やクラスター分析の前に行う方法も考えられており，正しい．

9-2 与えられた分散共分散行列の大きい順に 2 つの固有値に対応する固有ベクトルをそれぞれ求めると，それぞれ

$$\boldsymbol{w}_1 = (-0.560, -0.385, -0.734)^\top, \quad \boldsymbol{w}_2 = (0.816, -0.408, -0.408)^\top$$

となる．これらがそれぞれ第 1 主成分，第 2 主成分の重みベクトルに対応する．

9-3 次のプログラムを実行することで，第 1 主成分，第 2 主成分の重みベクトルを出力できる．

```
res = prcomp(Data, scale=T)
res$roadings[, 1:2]
```

第 1 主成分は CAtBat（打席数）や CHits（安打数）など打者の能力が正の値を示しているため，打者の総合的な能力を表していると考えられる．第 2 主成分は，1986 年の成績が正，通算成績が負の値を示していることから，他のシーズンに比べて 1986 年のシーズンに活躍したかを表していると考えられる．

9-4 次のプログラムにより，主成分得点に対する線形回帰モデルの推定ができる．

```
res.pc = prcomp(X) # 主成分分析実行
Z = X %*% res.pc$rotation[, 1:2]  # 主成分得点

# 線形回帰
res.reg1 = lm(y~Z[,1])  # 第 1主成分のみ用いた場合
summary(res.reg1)
res.reg2 = lm(y~Z)  # 第 1，第 2主成分を用いた場合
summary(res.reg2)
```

lm 関数のの実行結果に対して summary 関数を実行することで，第 1 主成分のみを用いた主成分回帰では決定係数は 0.9143，第 1，第 2 主成分を用

いた場合は決定係数は 0.9289 となる．

第 10 章

10-1 (b)

(a) 探索的因子分析では，共通因子と変数のつながりを固定するもので，共通因子の数を 1 つに固定するとは限らないので，誤り．(b) 因子分析モデルは，因子負荷量が未知であるだけでなく，共通因子も未観測であるため，正しい．(c) 因子負荷量は，複数の共通因子によって役割分担ができていた方が，共通因子の解釈がしやすいため，誤り．(d) 因子回転は共通因子の解釈をわかりやすくするために用いられるもので，独自因子を減らすために用いられるものではなく，誤り．

10-2 (10.4) 式の要素から，第 j 変数 X_j の分散は

$$\mathrm{Var}[X_j] = \sum_{l=1}^{m} {\lambda_{jl}}^2 + {\sigma_j}^2$$

で与えられる．これより，

$$1 = \frac{\sum_{l=1}^{m} {\lambda_{jl}}^2}{\mathrm{Var}[X_j]} + \frac{{\sigma_j}^2}{\mathrm{Var}[X_j]}$$

となる．右辺第 1 項は共通性 (10.5) 式そのものであることと，独自性は 1-共通性であることから，右辺第 2 項が独自性になる．

10-3 ［**第 1 版第 1 刷の注**：［R 使用］とありますが，R は使用しません．］

各因子のもつ意味は，次のように解釈できる．

因子 1：走力および幅跳びの能力

因子 2：投てき力

因子 3：跳躍力

10-4 R の `factanal` 関数の実行結果のうち，`Proportion Var` に寄与率がまとめられている．

```
data = read.csv("score.csv")
res1 = factanal(data, 1); res1
res2 = factanal(data, 2); res2
res3 = factanal(data, 3); res3
```

このプログラムを実行することで，因子数 1, 2, 3 での寄与率は，それぞれ次であることがわかる．

因子数 1：0.475

因子数 2：0.472 0.326

因子数 3：0.473 0.319 0.166

10-5 ソースコード 10.3 において，`res1`, `res2`, `res3` から独自性（`Uniqueness`）を出力できる．共通性は 1-独自性により与えられるが，因子回転を行う場合でも行わない場合でも独自性は変わらないため，共通性も同じである．

第 11 章

11-1 (c)

(a) 多次元尺度構成法は距離行列に対応するデータさえあれば実行できるので，正しい．(b) 対応分析はクロス集計表のデータから行われるので，正しい．(c) 正準相関分析では，分析に用いた 2 つの変数群のうち小さいほうの変数の数だけ正準相関係数が得られるため，誤り．(d) 構造方程式モデルは，線形重回帰モデルや因子分析を包含したモデルとみなすことができるので，正しい．

11-2 次のプログラムを実行することで，多次元尺度構成法により得られる 2 次元の値から，距離行列を求めることができる．

```
Jpdata = read.csv("Japandist.csv")
# 距離行列計算
D = dist(Jpdata[, -1])
pref = Jpdata[, 1]
# 多次元尺度構成法適用
res = cmdscale(D, 2)
# 多次元尺度構成法により得られる座標から距離行列計算
D_cmd = dist(res)
```

得られた `D` と `D_cmd` の値を比較すると，`D_cmd` はある程度 `D` に近いことがわかる．

11-3 `caith` データは，目の色（blue, light, medium, dark）と髪の色（fair, red, medium, dark, black）についてのクロス集計表である．このデータに対して対応分析を行うプログラムを，次に示す．

```
library(MASS)
library(ca)
?caith # caithデータのヘルプ
res = ca(caith)
plot(res) # 各変量の散布図
```

このプログラムを実行することで出力される図では，青点が目の色，赤点が髪の色に対応しており，目の色が青または明るい人は赤髪または金髪の傾向が高く，目の色が黒の人の髪の色は暗めまたは黒である傾向が強いことがわかる．また，目の色がミディアムの人は髪の色もミディアムである．

11-4 がく片の長さ，幅を第 1 変数群，花弁の長さ，幅を第 2 変数群として正準相関分析を行うプログラムを，次に示す．

```
# 2変数群のデータ
X = iris[, c(1,2)]
Y = iris[, c(3,4)]
# 正準相関分析実行
res = cancor(X, Y)
# 第 1，第 2正準相関係数
res$cor
# 変数群 1の第 1，第 2正準重みベクトル
res$xcoef
# 変数群 2の第 1，第 2正準重みベクトル
res$ycoef
```

第 1 正準重みベクトルは，がく片の長さ，幅の順に $(-0.086, 0.070)^\top$，花弁の長さ，幅の順に $(-0.069, 0.057)^\top$ である．このことから，がく片が短く太い個体は，花弁も短く太い傾向にあることを表した軸と考えられる．第 2 正準重みベクトルは，がく片の長さ，幅の順に $(0.005, 0.176)^\top$，花弁の長さ，幅の順に $(-0.157, 0.394)^\top$ である．このことから，第 2 正準相関係数は特にがく片と花弁の幅と，花弁の長さ（短さ）に注目した軸と考えられる．

11-5 ［**第 1 版第 1 刷の注**：問題を次のものに差し替えてください．

11.4.3 項で扱った `Bollen` データの潜在変数および観測変数を RAM (11.15) 式に対応させたものを，$\boldsymbol{f} = (f_1, f_2, f_3)^\top$, $\boldsymbol{X} = (x_1, x_2, x_3, y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)$ とする．このとき，図 11.6 に示すパス図および `sem` 関数の出力結果から，観測変数ベクトル $\boldsymbol{X}$ に対する観測変数ベクトル $\boldsymbol{f}$ の係数からなる行列 A_b の要素を記せ．］

図 11.6 を RAM で表したとき，A_b に対応する行列は次で与えられる．

$$A_b = \begin{pmatrix} 1.00 & 0 & 0 \\ 2.18 & 0 & 0 \\ 1.82 & 0 & 0 \\ 0 & 1.00 & 0 \\ 0 & 1.19 & 0 \\ 0 & 1.17 & 0 \\ 0 & 1.25 & 0 \\ 0 & 0 & 1.00 \\ 0 & 0 & 1.19 \\ 0 & 0 & 1.17 \\ 0 & 0 & 1.25 \end{pmatrix}$$