データサイエンスの基礎

Rによる統計学独習 演習問題解答集

地道 正行

2019 年 2 月 18 日

はじめに

拙著『R による統計学独習』をご購入いただき，まことに有り難うございます.

筆者が学部学生の頃は, パーソナルコンピュータといった個人で自由に利用できる計算機環境が普及しておらず, さらに本書で扱っている R などのフリーのソフトウェアが存在しなかったことから, データから平均・分散を計算したり, ヒストグラムを描くためには, いわゆる「手計算」と「手作業」が必要でした．このような「道具」が利用できない中で, これらの課題に答えるためには, 多大な努力と時間, そして労力が必要であり, そのこと自体が目的になっていたように記憶しています.

現在では, R を利用すれば, 本書で扱っている程度の大きさのデータであれば, これらの結果を得るために要する時間や労力はわずかなものとなりました[1]. ただし, 統計学をはじめて学ぶ方にとって, 何のためにデータの平均や分散の値を求めたり (要約), ヒストグラムなどを描く (可視化) のかということや, これらの結果がどのような意味があるのかといったことを理解することは依然として難しいようです．筆者は, データの要約と可視化を行うことと, 推測統計学における 3 本柱である「推定」･「検定」･「回帰」との関連を強く意識しながら学生自身が学ぶことが, その動機と意味を理解するために重要であることを, 大学での講義を行う中で以前よりもまして強く感じています.

また, データの「要約」と「可視化」と統計的推測における 3 つのテーマをつなぎ止めている「 母集団分布」,「 標本分布」[2] の理解も初学者にとっては難解であるということを学生からのフィードバックによって知ることができました.

幸い R には, 豊富な関数がそろっており, これらを利用して本書の内容や章末ごとに与えられている演習問題を解答することが, はじめて統計学を学ぶ上で理解することが難しいテーマを克服する手がかりとなると思われます．また, 様々な確率分布を R を使って実際に描いてみることが母集団分布の理解に繋がり, R を使って無作為標本を擬似的に発生させ, 統計量の分布を経験的に描くことによって, 標本分布についての理解が深まることが期待されます.

一般に, 学問を学ぶことの本質は, 「知識」(knowledge) を身につけることにあると思われますが, 統計学ではこのことにとどまらず, 「技術」 (technique) を体得することが非常に重要と考えられます．つまり, 専門用語・定理・手法などを記憶し, 理解することにとどまらず, それらを適切に利用して結果を導く「術 (すべ)」を学び, 実際に利用できるようになることが要求されます．このような技術を身につけるために, R を有効に活用し, 実際に演習問題を解けるようになることが

[1] さらに, R がフリー (free) であることも, 特に学生の方にとっては大きな利点かもしれません．なお, ここでフリーであるということは, 文字通り無料のソフトウェアであるとともに, ユーザーが R でいかなるデータ解析を行うことに関しても制約されないという自由も与えられていることに注意が必要です.

[2] あくまでもイメージですが, 統計学における要約と可視化を「筋肉」, 推定・検定・回帰を「骨」と考えた場合, 母集団分布 (確率変数と確率分布も含む), 標本分布 (無作為標本と統計量も含む) はそれらをつなぎ合わせる役目を果たす「靱帯」といえるのかもしれません.

重要と思われます.

末筆ながら, 統計学を学ぶことが, 不確定な現象を科学的に解明するために必要な思考力の涵養へとつながることを期待します. 拙著とこの解答集が統計学を学ぶ初学者をはじめとする多くの方への一助となることを願っています.

2019 年 初春
筆者

目 次

第1章　演習問題解答 (1)

Q 1.1 読者各自のコンピュータ環境に応じた R のインストーラを適当なサイトからダウンロードし, 適切にインストールを行え.

解説: R をインストールするときに, まずどのバージョンをインストールすればよいかが問題となる. R のバージョンは 3.4.3 のようにナンバリングされていることに注意しよう. ここで, 先頭の 3 がメジャーバージョン (major version) であり, 4.3 がマイナーバージョン (minor version) である. R はほぼ 1 年に 1 回のペースでマイナーバージョンの一桁目がアップし, マイナーバージョンの二桁目が数回アップする. 例年では, 3, 4 月にマイナーバージョンの一桁目が一つアップすると同時に二桁目が 0 のものがリリースされる (例えば, 3.4.0). また, 5, 6, 7 月頃にマイナーバージョンの二桁目が 1 のものがリリースされ (例えば, 3.4.1) , その後は 2, 3, 4 等とアップしていく (例えば, 3.4.2, 3.4.3, 3.4.4) [1]. どのバージョンをインストールすればよいかは場合によるけれども, 何らかの改良がなされているという理由から, マイナーバージョンの二桁目が高いものが推奨される.

では, Windows, macOS のそれぞれにインストールを手順のみ以下に与える [2]:

R for Windows のインストール 以下の手順に従ってインストールを行う:

(RW-1) CRAN サイトへアクセス

(RW-2) R for Windows のインストーラのダウンロード

(RW-3) インストーラの実行

(RW-4) R for Windows の実行 (インストール後の動作チェック)

R for macOS のインストール 以下の手順に従ってインストールを行う:

(RM-1) CRAN サイトへアクセス

(RM-2) R for macOS のインストーラのダウンロード

(RM-3) インストーラの実行

(RM-4) R for macOS の実行 (インストール後の動作チェック)

なお, Windows 環境でのインストールでは本文でも述べたけれども, ユーザ名は「半角英数文字」(ASCII コード) を利用することを強く推奨したい. また, macOS ではあまり問題にならないけれども, Windows に関しては, インストール場所としてデフォルトの `C:¥Program Files` は避けた方

---

[1] R の開発バージョンはほぼ毎日のペースでアップデートされている.

[2] R のテキストの中には, インストール手順をスクリーンショット付きで詳しく述べているものもあるけれども, 出版物や文書化されたものは, バージョンアップのスピードが早いため, すぐに陳腐化してしまうことから, ここではあえて与えていない. もしどうしてもインストールの段階のスクリーンショットを参照したい場合は Google 等の検索サイトで「R インストール」と検索してほしい.

がよい場合がある．特に，「コマンドプロンプト」でコマンドラインから R を起動したいユーザは空白を含むフォルダ名が思わぬトラブルを引き起こす可能性があることを知っておくべきである．

Q 1.2 R に関する情報が提供されているサイト (例えば, The R Project や RjpWik) にアクセスし, どのような情報が提供されているかを調べよ．

解説: The R Project のサイトにアクセスすることによって, R の専門誌である *The R Journal* を PDF ファイルでダウンロードすることができる．例えば, 2016 年 12 月にリリースされた第 8 巻第 2 号 (Volume 8/2) の p. 507 の記事 [3] を読むと, 2017 年 1 月 27 日頃にパッケージ数が 10,000 個を超えたことが報告されている [4]．さらに, 2017 年 12 月にリリースされた第 9 巻第 2 号 (Volume 9/2) の p. 564 の記事 [5] によると, 2017 年 11 月 30 日時点でのパッケージ数は, 約 11,875 個となっている．筆者が R を教育・研究に利用するようになった 2000 年前後は数十のオーダーであったことを思い出すにつけ, 現在までの R のパッケージ開発のめざましさを改めて感じる．

[3] Kurt Hornik and Achim Zeileis による CRAN に関する情報を提供した記事 *Changes on CRAN 2016-08-01 to 2017-01-31* である．なお, 著者の一人である Kurt Hornik 氏は当初から R FAQ (Frequently Asked Question) の編集を行っている．

[4] 2010 年頃までは千数百のオーダーであったことから,「手動」で全パッケージをインストールしていたことを思い出す．既に, 手動でパッケージを管理することは難しい...

[5] *Changes on CRAN 2017-06-01 to 2017-11-30*

第2章　演習問題解答 (2)

Q 2.1 演算子 : と関数 seq を使った例をそれぞれ 3 つずつ考えよ.

解答: 演算子 : を使った例題として, 任意の自然数 n に対して,

$$S_n := 1 + \cdots + n = \frac{n(n+1)}{2}$$

が成り立つことに関するものを取り上げる [1]. $n = 100$ を考え, 1 から 100 までの自然数の和

$$1 + \cdots + 100 = \frac{100 \times 101}{2} = 5050$$

が成り立つことを R を使って確かめる. まず, 1 から 100 までの階差 1 の数列を昇順と降順にそれぞれ生成し, さらにそれらを加えることによって,

```
> 1:100
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
 [17]  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32
 [33]  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48
 [49]  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64
 [65]  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
 [81]  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96
 [97]  97  98  99 100
> 100:1
  [1] 100  99  98  97  96  95  94  93  92  91  90  89  88  87  86  85
 [17]  84  83  82  81  80  79  78  77  76  75  74  73  72  71  70  69
 [33]  68  67  66  65  64  63  62  61  60  59  58  57  56  55  54  53
 [49]  52  51  50  49  48  47  46  45  44  43  42  41  40  39  38  37
 [65]  36  35  34  33  32  31  30  29  28  27  26  25  24  23  22  21
 [81]  20  19  18  17  16  15  14  13  12  11  10   9   8   7   6   5
 [97]   4   3   2   1
> 1:100 + 100:1
```

[1] この公式は, 数学者 F. Gauss が小学生のときに算数の授業で利用していたという逸話で有名である. この公式は,

$$S_n := 1 + \cdots + n = n + \cdots + 1$$

より,

$$2S_n = (n+1) + \cdots + (n+1)$$

となり,

$$S_n = \frac{n(n+1)}{2}$$

と導かれる.

```
  [1] 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101
 [17] 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101
 [33] 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101
 [49] 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101
 [65] 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101
 [81] 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101
 [97] 101 101 101 101
```

となり, それらを加えたものを 2 で割ることによって,

```
> sum(1:100 + 100:1)/2
[1] 5050
```

となる. 当然であるが, これは,

```
> sum(1:100)
[1] 5050
```

という結果と一致する.
関数 seq を使った例としては,

```
> seq(1, 30, 3)
 [1]  1  4  7 10 13 16 19 22 25 28
> seq(1, 10)^3
 [1]    1    8   27   64  125  216  343  512  729 1000
> seq(1, 10, length = 20)
 [1]  1.000000  1.473684  1.947368  2.421053  2.894737  3.368421
 [7]  3.842105  4.315789  4.789474  5.263158  5.736842  6.210526
[13]  6.684211  7.157895  7.631579  8.105263  8.578947  9.052632
[19]  9.526316 10.000000
```

などが考えられる.

Q 2.2 以下の数列を生成せよ.

(1) `[1]  1  4  7 10 13 16 19 22 25 28`

(2) `[1]    1    8   27   64  125  216  343  512  729 1000`

(3)
```
 [1]  1.000000  1.473684  1.947368  2.421053  2.894737  3.368421
 [7]  3.842105  4.315789  4.789474  5.263158  5.736842  6.210526
[13]  6.684211  7.157895  7.631579  8.105263  8.578947  9.052632
[19]  9.526316 10.000000
```

解答: (1), (2), (3) についてそれぞれ, Q 2.1 の関数 seq で与えた例が解答となっている. すなわち,

```
> seq(1, 30, 3)
 [1]  1  4  7 10 13 16 19 22 25 28
> seq(1, 10)^3
 [1]    1    8   27   64  125  216  343  512  729 1000
> seq(1, 10, length = 20)
 [1]  1.000000  1.473684  1.947368  2.421053  2.894737  3.368421
 [7]  3.842105  4.315789  4.789474  5.263158  5.736842  6.210526
[13]  6.684211  7.157895  7.631579  8.105263  8.578947  9.052632
[19]  9.526316 10.000000
```

である.

Q 2.3 以下は R に標準的に用意されているオブジェクトである. 実際にコンソールから入力してどのようなものかを確認せよ:

```
LETTERS
letters
month.abb
month.name
```

解答: 実際に入力することによって以下のように与えられる:

```
> LETTERS
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P"
[17] "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
> letters
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p"
[17] "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
> month.abb
 [1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov"
[12] "Dec"
> month.name
 [1] "January"   "February"  "March"     "April"     "May"
 [6] "June"      "July"      "August"    "September" "October"
[11] "November"  "December"
```

Q 2.4 行列 $\mathbf{X}$ と列ベクトル $\boldsymbol{y}$ を

$$\mathbf{X} = \begin{bmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 1 \end{bmatrix}$$

として, 以下の計算を R を用いて行え:

(1) $\mathbf{X}'\mathbf{X}$, (2) $(\mathbf{X}'\mathbf{X})^{-1}$, (3) $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{y}$, (4) $\mathbf{P} := \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$, $\mathbf{Q} := \mathbf{I}_5 - \mathbf{P}$,
(5) $\mathbf{P}'$, $\mathbf{Q}'$, (6) $\mathbf{P}^2$, $\mathbf{Q}^2$, (7) $\mathrm{trace}\mathbf{P}$, $\mathrm{trace}\mathbf{Q}$, (8) $\langle \mathbf{P}\boldsymbol{y}, \mathbf{Q}\boldsymbol{y} \rangle$

解答: まず行列 $\mathbf{X}$ と列ベクトル $\boldsymbol{y}$ を以下のように入力することによって定義する:

```
> (X <- cbind(rep(1 ,5), seq(-2, 2)))
     [,1] [,2]
[1,]    1   -2
[2,]    1   -1
[3,]    1    0
[4,]    1    1
[5,]    1    2
> (y <- matrix(c(1, 2, 1, 2, 1)))
     [,1]
[1,]    1
[2,]    2
[3,]    1
[4,]    2
[5,]    1
```

(1) $\mathbf{X}'\mathbf{X}$ (交差積行列 (cross product matrix)) は,

```
> crossprod(X)
     [,1] [,2]
[1,]    5    0
[2,]    0   10
```

によって求まる.

(2) 交差積行列の逆行列 $(\mathbf{X}'\mathbf{X})^{-1}$ は,

```
> solve(crossprod(X))
     [,1] [,2]
[1,]  0.2  0.0
[2,]  0.0  0.1
```

となる.

(3) $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{y}$ [2] は,

```
> solve(crossprod(X)) %*% t(X) %*% y
     [,1]
[1,]  1.4
[2,]  0.0
```

によって求めることができる.

(4) 行列 $\mathbf{P}$, $\mathbf{Q}$ は以下の入力によって与えられる:

```
> (P <- X %*% solve(crossprod(X)) %*% t(X))
     [,1] [,2] [,3] [,4] [,5]
[1,]  0.6  0.4  0.2  0.0 -0.2
[2,]  0.4  0.3  0.2  0.1  0.0
[3,]  0.2  0.2  0.2  0.2  0.2
[4,]  0.0  0.1  0.2  0.3  0.4
[5,] -0.2  0.0  0.2  0.4  0.6
> (Q <- diag(5) - P)
     [,1] [,2] [,3] [,4] [,5]
[1,]  0.4 -0.4 -0.2  0.0  0.2
[2,] -0.4  0.7 -0.2 -0.1  0.0
[3,] -0.2 -0.2  0.8 -0.2 -0.2
[4,]  0.0 -0.1 -0.2  0.7 -0.4
[5,]  0.2  0.0 -0.2 -0.4  0.4
```

(5) 行列 $\mathbf{P}$, $\mathbf{Q}$ の転置は以下の入力によって求めることができる:

```
> t(P)
     [,1] [,2] [,3] [,4] [,5]
[1,]  0.6  0.4  0.2  0.0 -0.2
[2,]  0.4  0.3  0.2  0.1  0.0
[3,]  0.2  0.2  0.2  0.2  0.2
[4,]  0.0  0.1  0.2  0.3  0.4
[5,] -0.2  0.0  0.2  0.4  0.6
> t(Q)
     [,1] [,2] [,3] [,4] [,5]
[1,]  0.4 -0.4 -0.2  0.0  0.2
[2,] -0.4  0.7 -0.2 -0.1  0.0
[3,] -0.2 -0.2  0.8 -0.2 -0.2
[4,]  0.0 -0.1 -0.2  0.7 -0.4
[5,]  0.2  0.0 -0.2 -0.4  0.4
```

[2] 線形回帰モデル

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

において, $\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{y}$ は, 未知の回帰係数ベクトル $\boldsymbol{\beta}$ の最小自乗推定値のベクトルであることに注意しよう.

この結果から, 行列 $\mathbf{P}$, $\mathbf{Q}$ は対称であることがわかる.

(6) 行列 $\mathbf{P}$, $\mathbf{Q}$ の巾乗は以下の入力によって求めることができる:

```
> P %*% P
     [,1] [,2] [,3] [,4] [,5]
[1,]  0.6  0.4  0.2  0.0 -0.2
[2,]  0.4  0.3  0.2  0.1  0.0
[3,]  0.2  0.2  0.2  0.2  0.2
[4,]  0.0  0.1  0.2  0.3  0.4
[5,] -0.2  0.0  0.2  0.4  0.6
> Q %*% Q
     [,1] [,2] [,3] [,4] [,5]
[1,]  0.4 -0.4 -0.2  0.0  0.2
[2,] -0.4  0.7 -0.2 -0.1  0.0
[3,] -0.2 -0.2  0.8 -0.2 -0.2
[4,]  0.0 -0.1 -0.2  0.7 -0.4
[5,]  0.2  0.0 -0.2 -0.4  0.4
```

この結果から, 行列 $\mathbf{P}$, $\mathbf{Q}$ は巾等であることがわかる. なお, (5), (6) の結果から, 行列 $\mathbf{P}$, $\mathbf{Q}$ は共に射影行列であることがわかる.

(7) トレースは,

```
> sum(diag(P))
[1] 2
> sum(diag(Q))
[1] 3
```

によって求めることができる. なお, 射影行列のトレースは行列のランクを表すことに注意しよう.

(8) $\langle \mathbf{P}\boldsymbol{y}, \mathbf{Q}\boldsymbol{y} \rangle = (\mathbf{P}\boldsymbol{y})'\mathbf{Q}\boldsymbol{y}$ であるので,

```
> t(P %*% y) %*% Q %*% y
              [,1]
[1,] -7.771561e-16
```

より求まる. なお, この結果は射影行列 $\mathbf{P}$, $\mathbf{Q}$ によってベクトル $\boldsymbol{y}$ を射影したものが直交する[3] ことを表していることに注意しよう.

[3] ただし, 計算の丸め誤差のために正確に 0 とはなっていないことに注意しよう.

第3章　演習問題解答 (3)

Q 3.1 以下のようなファイルが作業ディレクトリに用意されているものとする:

データファイル: `firms.rda`(一部)

```
   sale employee    asset code
  77674     1540   121201    1
 682385    32595   720707    1
  80006     1443   138767    1
 251358    51406   306772    1
 171763     1554   204786    1
 395502     6700   497451    1
      :        :        :    :
      :        :        :    :
  89568     1295   228236    2
 101879     2407    83815    2
```

このファイルはデータ間の区切り文字が半角空白のテキストファイルである．なお, 1 行目には 2 行目以降のデータに関する変量名 (`sales`, `employees`, `assets`, `code`) が与えられていることに注意しよう．関数 `read.table` を利用してこのファイルからデータを読み込み `firms.frame` というオブジェクトに代入 (付値) せよ．

解答: 以下のように入力することによって読み込むことができる:

```
> firms.frame <- read.table("firms.rda", header=TRUE)
```

なお, 読み込まれたオブジェクトは以下のように確かめることができる:

```
> head(firms.frame)
   sales employees assets code
1  77674      1540 121201    1
2 682385     32595 720707    1
3  80006      1443 138767    1
4 251358     51406 306772    1
5 171763      1554 204786    1
6 395502      6700 497451    1
```

```
> dim(firms.frame)
[1] 100   4
> str(firms.frame)
'data.frame':	100 obs. of  4 variables:
 $ sales    : int  77674 682385 80006 251358 171763 395502 22716 30967 38745 1981763 ...
 $ employees: int  1540 32595 1443 51406 1554 6700 493 3058 658 44206 ...
 $ assets   : int  121201 720707 138767 306772 204786 497451 26437 40407 28403 2320529 ...
 $ code     : int  1 1 1 1 1 1 1 1 1 1 ...
```

Q 3.2 R には Microsoft Excel [1] (以後 Excel と略す.) で作成されたファイルから直接 R へデータを読み込むための関数がいくつかのパッケージとして提供されている. 以下の手順に従ってパッケージ `readxl` で提供されている `read_excel` を利用してデータを R へ読み込め.

(1) Excel ファイル `babies.xlsx` を作業ディレクトリに保存せよ.
(2) パッケージ `readxl` をインストールせよ.
(3) `library` を利用して `readxl` をロードせよ.

```
> library(readxl)
```

(4) `read_excel` を使って, 以下のように R に読み込め.

```
> babies.readxl.frame <- read_excel("babies.xlsx")
```

(5) この結果と 3.3 節で `read.table` を使って読み込んだ結果とを比較し, どのような違いがあるかを考察せよ.

解答: ファイル `babies.xlsx` が適切なフォルダに保存され, さらにパッケージ `readxl` が適切にインストールされているとして, 以下のように入力すればよい:

```
> library(readxl)
> babies.readxl.frame <- read_excel("babies.xlsx")
```

関数 `read_excel` によって読み込まれたオブジェクト `babies.readxl.frame` と `read.table` を使って読み込んだ結果の違いは以下のように確かめることができる:

```
> str(babies.frame)
```

[1] Microsoft Excel は米国 Microsoft Corporation の米国およびその他の国における登録商標です.

```
'data.frame':	100 obs. of  5 variables:
 $ weight: int  3170 2610 3020 3020 3330 2180 3140 3420 2580 3360 ...
 $ height: num  49.5 45 48.5 49 50 46 50 50 47 49.5 ...
 $ chest : num  33.5 30.5 32.5 31 34 28 32 33 30.5 32 ...
 $ head  : num  34 31.5 32.5 32.5 35 32 31.5 35 32.5 33 ...
 $ gender: Factor w/ 2 levels "female","male": 1 2 2 2 2 2 1 2 1 2 ...
> str(babies.readxl.frame)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':	100 obs. of  5 variables:
 $ weight: num  3170 2610 3020 3020 3330 2180 3140 3420 2580 3360 ...
 $ height: num  49.5 45 48.5 49 50 46 50 50 47 49.5 ...
 $ chest : num  33.5 30.5 32.5 31 34 28 32 33 30.5 32 ...
 $ head  : num  34 31.5 32.5 32.5 35 32 31.5 35 32.5 33 ...
 $ gender: chr  "female" "male" "male" "male" ...
```

この結果から, オブジェクト `babies.frame` のクラスが, `data.frame` であるのに対し, `babies.readxl.frame` のクラスは `data.frame` であると同時に `tbl_df`, `tbl` となっている. これは, 表示などの点で改良がなされたものであることに注意しよう.

Q 3.3 各自が用意したデータファイルを, 本章で扱った関数を利用して R に読み込め.

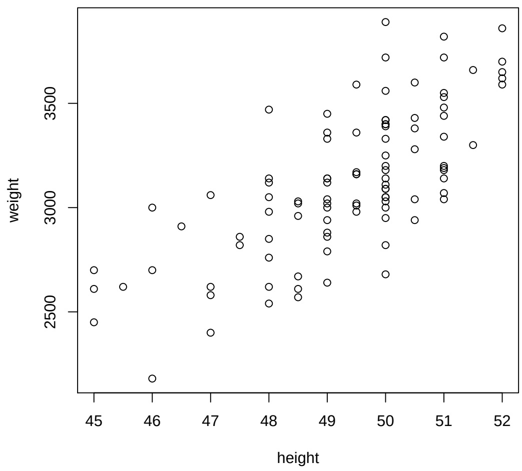
(読者が各自試して頂ければ幸いである.)

第4章　演習問題解答 (4)

Q 4.1 3.3 節で読み込んだデータフレーム babies.frame を利用して, 身長 (height) と体重 (weight) の散布図を plot と ggplot を使って描け.

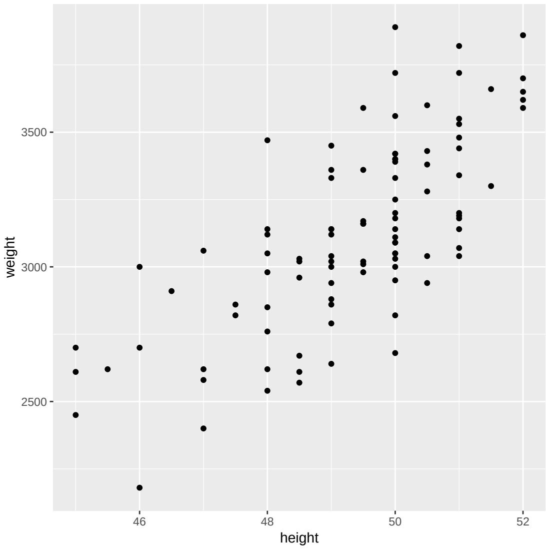
解答: 以下のように入力することによって関数 plot を使って散布図を描くことができる:

```
> plot(weight ~ height, babies.frame)
```



また, ggplot2 パッケージをロードし, 以下のように入力することによって関数 ggplot を使って散布図を描くことができる:

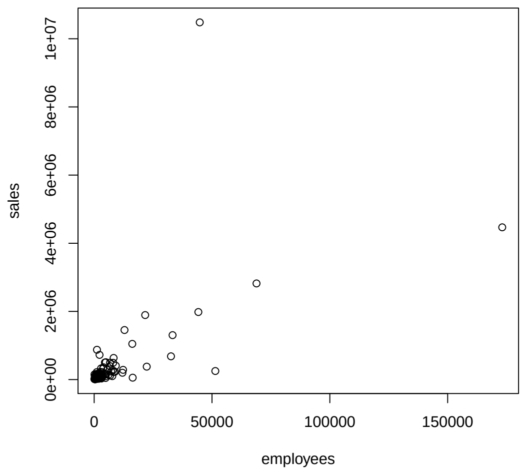
```
> library(ggplot2)
> ggplot(babies.frame, aes(x = height, y = weight)) + geom_point()
```



Q 4.2 演習問題 Q 3.1 で読み込んだデータフレーム `firms.frame` を利用して, 従業員数 (`employees`) と売上高 (`sales`) の散布図を `plot` と `ggplot` を使って描け.

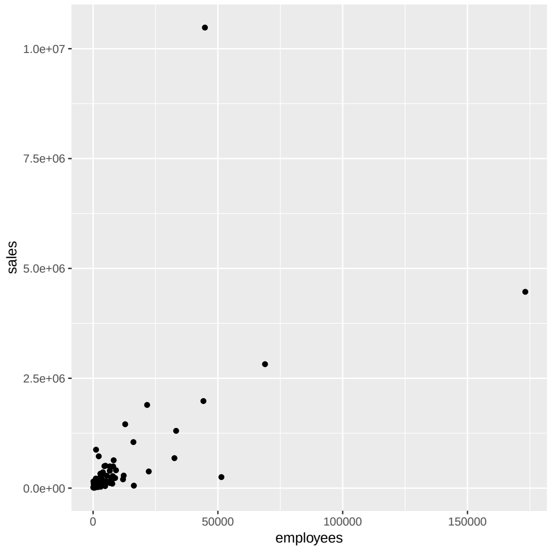
解答: 以下のように入力することによって関数 `plot` を使って散布図を描くことができる:

```
> plot(sales ~ employees, firms.frame)
```



また, `ggplot2` パッケージをロードし, 以下のように入力することによって関数 `ggplot` を使って散布図を描くことができる:

```
> library(ggplot2)
> ggplot(firms.frame, aes(x = employees, y = sales)) + geom_point()
```



これらのプロットから, 散布図におけるデータの点は原点付近に集中的に分布し, 従業員数と売上高は 2 次元の意味で歪みがあることがわかる.

Q 4.3 演習問題 Q 3.3 で各自が読み込んだデータフレームを利用して, 散布図を `plot` と `ggplot` を使って描け.

(読者が各自試して頂ければ幸いである.)

第5章　演習問題解答 (5)

Q 5.1 標本分散を計算するための R 関数が

```
> svar <- function(x) mean((x - mean(x))^2)
```

で定義されることを説明せよ．

解答: ベクトル x にデータ $(x_1,\ldots,x_n)$ が収められているとすると, 偏差のベクトル

$$(x_1-\overline{x},\ldots,x_1-\overline{x})$$

は x-mean(x) というスクリプトによって求めることができ, 偏差平方のベクトル

$$\left((x_1-\overline{x})^2,\ldots,(x_1-\overline{x})^2\right)$$

は (x-mean(x))^2 というスクリプトによって与えられる．さらに偏差平方の平均

$$\frac{1}{n}\left\{(x_1-\overline{x})^2+\cdots+(x_1-\overline{x})^2\right\}=s^2$$

はデータの標本分散となるので, 結局, mean((x-mean(x))^2) が求めるスクリプトとなる．なお, 関数 svar はこれを関数として定義したものである．

実際に, ベクトル x にデータ $(x_1,\ldots,x_n)=(1,2,\ldots,10)$ を代入することによって確かめる:

```
> (x <- seq(10)) # データ生成
 [1]  1  2  3  4  5  6  7  8  9 10
> x - mean(x) # 偏差ベクトル
 [1] -4.5 -3.5 -2.5 -1.5 -0.5  0.5  1.5  2.5  3.5  4.5
> (x - mean(x))^2 # 偏差平方ベクトル
 [1] 20.25 12.25  6.25  2.25  0.25  0.25  2.25  6.25 12.25 20.25
> mean((x - mean(x))^2) # 分散を求める
[1] 8.25
> svar(x) # 関数によって求めた結果とチェック
[1] 8.25
```

Q 5.2 平均絶対偏差 (mean absolute deviation)

$$d := \frac{1}{n}\sum_{i=1}^{n}|x_i - \overline{x}|$$

を計算する関数を作成せよ．ただし, 絶対値 $|\cdot|$ を計算する R 関数は abs である．

解答: 以下のように定義できる:

```
> mad <- function(x) mean(abs(x - mean(x)))
```

この関数の実行例として, 例えば新生児の体重 weight の平均絶対偏差の値を求めると,

```
> mad(weight)
[1] 296.808
```

となる．

Q 5.3 データ $\{x_1, \ldots, x_n\}$ に対する幾何平均 $\overline{x}_G := \sqrt[n]{\prod_{i=1}^{n} x_i}$ と調和平均 $\overline{x}_H := 1\Big/\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{x_i}\right)$ を求める関数 gmean, hmean をそれぞれ定義せよ．

解答: 幾何平均に関して,

$$\log \overline{x}_G = \log\left(\prod_{i=1}^{n} x_i\right)^{1/n} = \frac{1}{n}\sum_{i=1}^{n}\log x_i =: \overline{\log x} \Longleftrightarrow \overline{x}_G = \exp\left(\overline{\log x}\right)$$

が成り立つことから, 以下のようにコーディングできる:

```
> gmean <- function(x) exp(mean(log(x)))
```

このような工夫が必要な理由としては, 一般にデータの積は「大きく」なる傾向にあり, 直接計算することが難しくなる場合に対処するためである．

また, 調和平均は「逆数の平均の逆数」であることから以下のようにコーディングされる:

```
> hmean <- function(x) 1/mean(1/x)
```

この関数の実行例として, 例えば新生児の体重 weight の通常の平均値 (算術平均) と幾何平均, 調和平均の値を求めると,

```
> mean(weight)
[1] 3160.2
> gmean(weight)
[1] 3137.376
> hmean(weight)
[1] 3114.416
```

となり, 「算術平均」>「幾何平均」>「調和平均」の大小関係が成り立つことがみてとれる.

Q 5.4 以下の設問に答えよ.

(1) データに関する平均まわりの k 次中心モーメント

$$m_k := \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^k$$

を求める R 関数 `moment` を定義せよ.

(2) データに対する歪度 (skewness) $\widehat{\beta}_0$ と尖度 (kurtosis) $\widehat{\beta}_1$

$$\widehat{\beta}_0 = \frac{m_3}{m_2^{\frac{3}{2}}}, \quad \widehat{\beta}_1 = \frac{m_4}{m_2^2}$$

を求める R 関数 `skewness`, `kurtosis` を定義せよ.

(3) 3.1 節で読み込んだ新生児の体重データ `weight` に対する歪度, 尖度を計算せよ. また, 正規分布にデータが従う場合に歪度と尖度は, それぞれ, 0, 3 に近い値をとることが知られているけれども, 新生児の体重の場合は正規分布に従っていると考えられるかを考察せよ.

解答: (1) 以下のように定義される:

```
> moment <- function(x, k) mean((x - mean(x))^k)
```

(2) 以下のように定義される:

```
> skewness <- function(x) moment(x, 3)/(moment(x, 2))^(3/2)
> kurtosis <- function(x) moment(x, 4)/moment(x, 2)^2
```

(3) 新生児の体重データの歪度と尖度は以下のように与えられる:

```
> skewness(weight)
[1] 0.1965304
> kurtosis(weight)
[1] 3.085669
```

この結果から, 歪度は微少な正の値となり, 右に歪んでいるけれども, 尖度はほぼ 3 となり正規分布と考えられる.

Q 5.5 フィボナッチ数列:

$$a_0 := 0,\ a_1 := 1,\ a_{n+1} := a_n + a_{n-1}, \quad n = 1, 2, \ldots$$

を生成する関数を定義せよ.

解答: フィボナッチ数列を生成する関数は以下のように定義される:

```
> Fibonacci <- function(n = 0)
+ {
+   if(n < 0){
+     cat("n must be non-negative!")
+   } else if(n == 0){
+     n
+   } else if(n == 1){
+     a <- c(0, 1)
+     a
+   } else {
+     a <- c(0, 1)
+     for(i in 2:n){
+       a <- c(a, a[i] + a[i-1])
+     }
+     a
+   }
+ }
```

ここで, `if` 関数を使って, n が負のときは非負 (`non-negative`) でなければいけないことを表示し, $n = 0, 1$ のときに, それぞれ, $a_0 = 0$, $a_1 = 1$ となるように条件分岐が行われた後, $n \geq 2$ のときは `for` 関数を利用して目的とする n 項までのフィボナッチ数列を生成する繰り返しが行われていることに注意しよう. 実際に, この関数を利用してフィボナッチ数列を生成させると以下のようになる:

```
> Fibonacci(-1)
n must be non-negative!
> Fibonacci(0)
```

```
[1] 0
> Fibonacci(1)
[1] 0 1
> Fibonacci(10)
 [1]  0  1  1  2  3  5  8 13 21 34 55
```

Q 5.6 数列 $a_n = \left(1+\frac{1}{n}\right)^n$ $(n = 1, 2, \ldots)$ を生成する関数を定義し, この数列が $n \to \infty$ のとき, 自然対数の底 $\mathrm{e} \simeq 2.72$ に収束することを, 横軸に n, 縦軸に a_n をとりプロットする関数を定義することによって確かめよ. その際, 適当に $\varepsilon > 0$ を設定し,

$$\forall n \geq N \Rightarrow \ |a_n - \mathrm{e}| < \varepsilon$$

をみたす N がどのような値になるかと共に考察せよ.

解答: 数列 $a_n = \left(1+\frac{1}{n}\right)^n$ が $n \to \infty$ のとき, 自然対数の底 e に収束することは, 解析的には, この数列が上に有界な単調増加列であることから従うことに注意しよう [1).

数列 $a_n = (1 + 1/n)^n$ を生成する関数は以下のように定義される:

```
> an <- function(n) (1 + 1/n)^n
```

例えば, 最初の 10 項は以下のように生成される:

```
> an(seq(10))
 [1] 2.000000 2.250000 2.370370 2.441406 2.488320 2.521626 2.546500
 [8] 2.565785 2.581175 2.593742
```

次に, 横軸に n を, 縦軸に数列 a_n をとった点列をプロットする以下の関数を用意しよう:

```
> plot.an <- function(N, epsilon = 0.1, xl = range(seq(N)), yl=c(2, 3))
+ {
+   n <- seq(N)
+   plot(n, an(n), type="b", xlim = xl, ylim = yl, pch = 20)
+   abline(h = exp(1), lwd = 2)
+   abline(h = c(exp(1) - epsilon, exp(1) + epsilon), lty = 2)
+ }
```

---

1) 解析的な考察のためには, 例えば,

杉浦 光夫 (1980)『解析入門 I』, 東京大学出版会

などを参照してほしい. また, 自然対数の底 e について詳しくは,

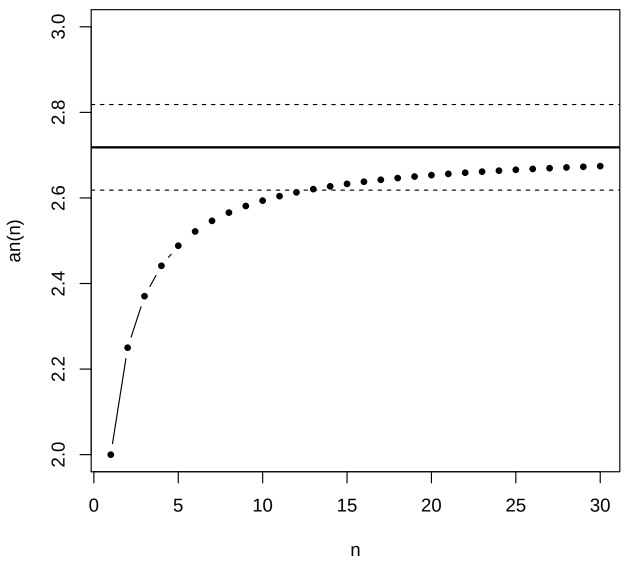
E. マオール (伊理 由美 訳) (1999) 『不思議な数 e の物語』, 岩波書店

を参照してほしい.

この関数では, 数列のプロットに加えて, 収束先の e を表す実線と, 適当に与えられた $\varepsilon(>0)$ に対して, $\mathrm{e}\pm\varepsilon$ の高さをもつ横軸に平行な点線を描いている．これらの直線の間 (帯) の中に点列 (n, a_n) が入ること (そして「再び飛び出さない」こと) が数列が収束している状態を表していることに注意しよう.

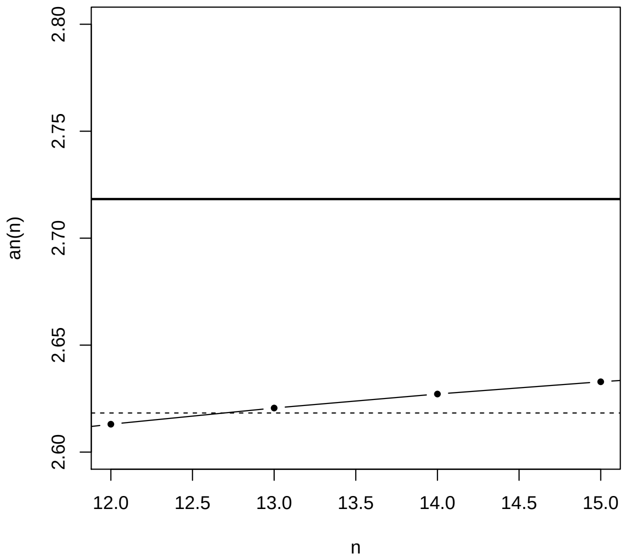
では, 実際に数列の収束状況をプロットによって確認する．まず, $\varepsilon = 0.1$ として, $n = 1, \ldots, 30(= N)$ の範囲でプロットを行う:

```
> plot.an(N = 30)
```



この図からは, $n = 13$ 付近の挙動がわかりづらいので, $n = 12, \ldots, 15$ 付近をズームアップしたものを以下のように描く:

```
> plot.an(N = 30, epsilon = 0.1, xl = c(12, 15), yl = c(2.6, 2.8))
```



この図から, $n \geq 13$ に対して,

$$|a_n - \mathrm{e}| < 0.1(=\varepsilon) \Longleftrightarrow a_n \in (\mathrm{e}-0.1, \mathrm{e}+0.1) = (2.6183, 2.8183)$$

が成り立っていることがわかる. ε を他のもの (例えば, 0.01) にとった場合を考察することも同様の方法できるので, 各自試してほしい.

Q 5.7 数列 $b_n = n\sin\frac{1}{n}$ に対して Q 5.6 と同様の関数を定義し, 収束性を考察せよ.

解答: 数列 $b_n = n\sin\frac{1}{n}$ が $n \to \infty$ のとき 1 に収束することに注意しよう. この数列を生成する関数は以下のように定義される:

```
> bn <- function(n) n*sin(1/n)
```

例えば, 最初の 10 項は以下のように生成される:

```
> bn(seq(10))
 [1] 0.8414710 0.9588511 0.9815841 0.9896158 0.9933467 0.9953768
 [7] 0.9966021 0.9973979 0.9979437 0.9983342
```

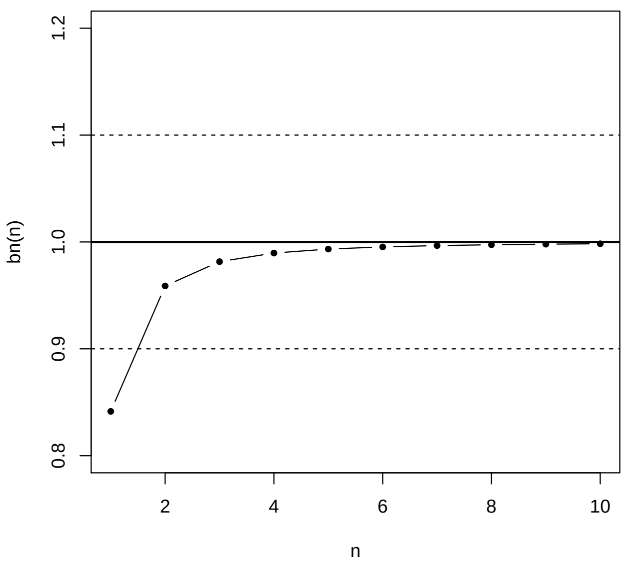
横軸に n を, 縦軸に数列 b_n をとった点列をプロットする以下の関数を用意しよう:

```
> plot.bn <- function(N, epsilon = 0.1, xl = range(seq(N)), yl = c(0.8, 1.2))
+ {
+   n <- seq(N)
+   plot(n, bn(n), type = "b", xlim = xl, ylim = yl, pch = 20)
+   abline(h = 1, lwd = 2)
+   abline(h = c(1 - epsilon, 1 + epsilon), lty = 2)
+ }
```

この関数では, 数列のプロットに加えて, 収束先の 1 を表す実線と, 適当に与えられた $\varepsilon(>0)$ に対して, $1\pm\varepsilon$ の高さをもつ横軸に平行な点線を描いている. これらの直線の間 (帯) の中に点列 (n, b_n) が入ること (そして「再び飛び出さない」こと) が数列が収束している状態を表していることに注意しよう.

では, 実際に数列の収束状況をプロットによって確認する. $\varepsilon=0.1$ として, $n=1,\ldots,10(=N)$ の範囲でプロットを行う:

```
> plot.bn(N = 10)
```

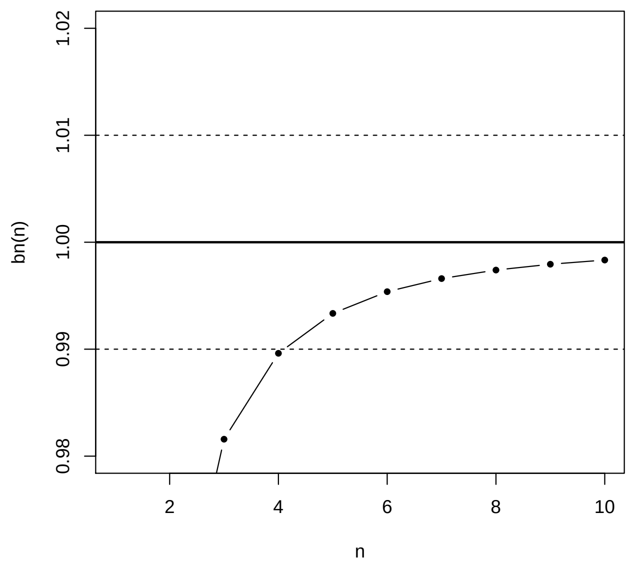


このプロットから, $n\geq 2$ に対して,

$$|b_n-1|<0.1(=\varepsilon) \Longleftrightarrow b_n\in(1-0.1, 1+0.1)=(0.9, 1.1)$$

が成り立つことがわかる. また, $\varepsilon=0.01$ として, $n=1,\ldots,10(=N)$ の範囲でプロットを行う:

```
> plot.bn(N = 10, epsilon = 0.01, yl = c(0.98, 1.02))
```



このプロットから, $n \geq 5$ に対して,

$$|b_n - 1| < 0.01(= \varepsilon) \Longleftrightarrow b_n \in (1 - 0.01, 1 + 0.01) = (0.99, 1.01)$$

が成り立つことがわかる.

Q 5.8 数列 $c_n = \sum_{k=1}^{n} \frac{1}{k} - \log_e(n)$ に対して Q 5.6 と同様の関数を定義し, 収束性を考察せよ.

解答: 数列 $c_n = \sum_{k=1}^{n} \frac{1}{k} - \log_e(n)$ が $n \to \infty$ のとき $\gamma := 0.5772157\ldots$ に収束することに注意しよう. (ここで, γ は**オイラーの定数** (Euler's constant) と呼ばれる.) この数列を生成する関数は以下のように定義される:

```
> cn <- function(n) cumsum(1/seq(n))[n] - log(n)
```

例えば, 最初の 10 項は以下のように生成される:

```
> cn(seq(10))
 [1] 1.0000000 0.8068528 0.7347210 0.6970390 0.6738954 0.6582405
 [7] 0.6469470 0.6384156 0.6317437 0.6263832
```

横軸に n を, 縦軸に数列 c_n をとった点列をプロットする以下の関数を用意しよう:

```
> plot.cn <- function(N, epsilon = 0.1, xl = range(seq(N)), yl = c(0, 1))
+ {
+   n <- seq(N)
+   plot(n, cn(n), type = "b", xlim = xl, ylim = yl, pch = 20)
+   abline(h = -digamma(1), lwd=2)
+   abline(h = c(-digamma(1) - epsilon, -digamma(1) + epsilon), lty = 2)
+ }
```

この関数では, 数列のプロットに加えて, 収束先の γ を表す実線と, 適当に与えられた $\varepsilon(> 0)$ に対して, $\gamma \pm \varepsilon$ の高さをもつ横軸に平行な点線を描いている. なお, R でオイラーの定数 γ を計算する方法は, **ディガンマ関数** (digamma function)

$$\Psi(x) := \frac{d \log \Gamma(x)}{dx} = \frac{\Gamma'(x)}{\Gamma(x)}$$

に関して,

$$\Psi(1) = \Gamma'(1) = -\gamma \Longleftrightarrow \gamma = -\Psi(1)$$

なる関係が成り立つことを利用する. すなわち, R 関数 `digamma` を利用し,

```
> -digamma(1)
[1] 0.5772157
```

と入力することによって求めることができることに注意しよう. なお, ガンマ関数は

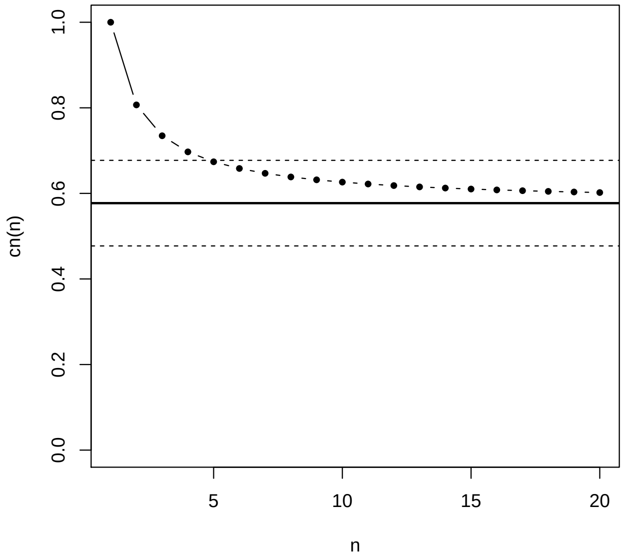
$$\Gamma(x) := \int_0^\infty t^{x-1} \mathrm{e}^{-t} \mathrm{d}t$$

で定義されることを思い出そう.

これらの直線の間 (帯) の中に点列 (n, c_n) が入ること (そして「再び飛び出さない」こと) が数列が収束している状態を表していることに注意しよう.

では, 実際に数列の収束状況をプロットによって確認する. $\varepsilon = 0.1$ として, $n = 1, \ldots, 20(= N)$ の範囲でプロットを行う:

```
> plot.cn(N = 20)
```



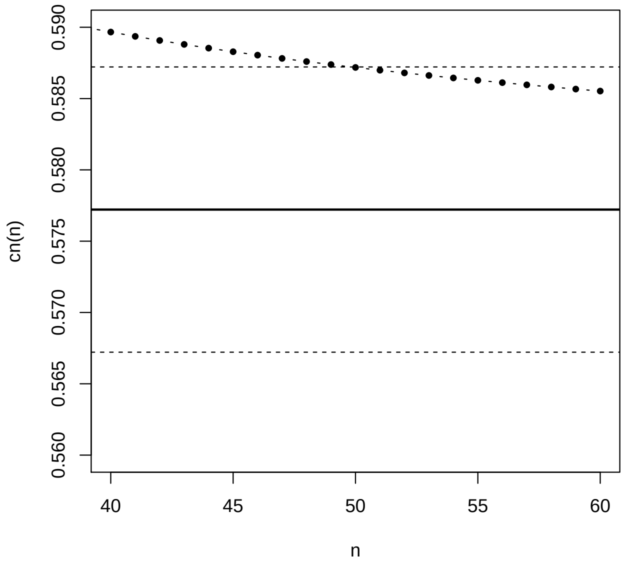
このプロットから, $n \geq 5$ に対して,

$$|c_n - \gamma| < 0.1 (= \varepsilon) \Longleftrightarrow b_n \in (\gamma - 0.1, \gamma + 0.1) = (0.4772, 0.6772)$$

が成り立つことがわかる.

また, $\varepsilon = 0.01$ として, $n = 40, \ldots, 60$ の範囲でプロットを行う:

```
> plot.cn(N = 60, epsilon = 0.01, xl = c(40, 60), yl = c(0.56, 0.59))
```



このプロットから, $n \geq 50$ に対して,

$$|c_n - 1| < 0.01 (= \varepsilon) \iff c_n \in (\gamma - 0.01, \gamma + 0.01) = (0.5672, 0.5872)$$

が成り立つことがわかる. (特に, $c_{50} = 0.58718 \in (0.56722, 0.58722)$ に注意しよう.)

Q 5.9 データ $x_1, \ldots, x_n$ に対する平均値

$$\overline{x}_n := \frac{1}{n} \sum_{i=1}^{n} x_i = \frac{1}{n} (x_1 + \cdots + x_n)$$

に関して以下の設問に答えよ.

(1) 以下のような漸化式の表現をもつことを示せ:

$$\overline{x}_n = \frac{n-1}{n} \overline{x}_{n-1} + \frac{1}{n} x_n, \quad n = 2, 3, \cdots \tag{5.1}$$

ただし, $\overline{x}_1 = x_1$ とおいた.

(2) 適当な N 個の正規乱数を発生させ, 1 個目から n 個目までの平均

$$\overline{x}_n := \frac{1}{n} \sum_{i=1}^{n} x_i, \quad n = 1, \ldots, N$$

を $n = 1, \ldots, N$ の全てについて求める関数を, 漸化式 (5.1) を用いた場合と用いない場合で作成し, そのスピードを比較せよ. ただし, R における実行時間を計測する関数は `system.time` である.

解答: (1) 平均値の定義から,

$$\begin{aligned}\overline{x}_n &= \frac{1}{n}\sum_{i=1}^{n}(x_1+\cdots+x_{n-1}+x_n)\\ &= \frac{1}{n}\sum_{i=1}^{n}(x_1+\cdots+x_{n-1})+\frac{1}{n}x_n\\ &= \frac{n-1}{n}\overline{x}_{n-1}+\frac{1}{n}x_n\end{aligned}$$

より成り立つ.

(2) まず漸化式を用いずに, その都度平均を求めるような関数を以下のように作成する:

```
> mean.sim <- function(x)
+ {
+   N <- length(x)
+   xbar <- rep(0, N)
+   xbar[1] <- x[1]
+   for(i in 2:N) xbar[i] <- mean(x[1:i])
+   list(x = x, N = N, xbar = xbar)
+ }
```

$N = 10000$ として, 標準正規乱数を生成し, この関数を実行した結果を計測する:

```
> set.seed(12345)
> x <- rnorm(10000)
> (res.sim <- system.time(mean.sim(x)))
   user  system elapsed
  0.306   0.059   0.371
```

ここで, `user` の計測値は関数を実行するのにかかった CPU 時間 (秒) を計測したものであることに注意しよう.

次に, 漸化式を利用した関数を以下のように作成する:

```
> mean.sim.rec <- function(x)
+ {
+   N <- length(x)
+   xbar <- rep(0, N)
+   xbar<-x[1]
+   for(n in 2:N) xbar[n] <- (n-1)*xbar[n-1]/n + x[n]/n
+   list(x = x, N = N, xbar = xbar)
+ }
```

この関数を実行したときの計測結果は以下のように与えられる:

```
> (res.sim.rec <- system.time(mean.sim.rec(x)))
   user  system elapsed
   0.01    0.00    0.01
```

これらの結果は, 計算機環境[2] に強く依存するため一概にはいえないが, `user` の計測値に関して `mean.sim` の方が `mean.sim.rec` よりも約 31(= 0.306/0.01) 倍遅いことに注意しよう.

研究: ここで作成した関数は繰り返しに関数 `for` を利用している. 本書の第 5 章の補足でも述べたが, 繰り返しに `for` を利用するのは最終的な手段と考え, できれば R に標準的に用意されている関数 (組み込み関数) を利用することが指摘されている. ここでは, 累積和を求めるための関数 `cumsum` を利用して関数を再実装してみる:

```
> mean.sim.cumsum <- function(x)
+ {
+   N <- length(x)
+   xbar <- cumsum(x)/seq(1,N)
+   list(x = x, N = N, xbar = xbar)
+ }
```

この関数を実行した場合の計測結果は以下のように与えられる:

```
> system.time(mean.sim.cumsum(x))
   user  system elapsed
  0.000   0.000   0.001
```

この結果は計算の実行時間が短いため計測できないことを表しており, 漸化式を利用した関数よりも遙かに早いことがわかる.

これらの結果を, より詳細に比較するためには, `microbenchmark` パッケージを導入することによって以下のように実行することができる:

```
> library(microbenchmark)
> microbenchmark(mean.sim(x), mean.sim.rec(x), mean.sim.cumsum(x), times = 10L)
Unit: microseconds
               expr        min         lq        mean      median
        mean.sim(x) 278342.677 293389.829 313551.0480 300661.9015
    mean.sim.rec(x)   2384.083   2477.212   2786.7775   2723.0840
 mean.sim.cumsum(x)     28.777     30.020    275.7651     37.7485
         uq        max neval cld
```

[2] 現在の, 筆者の計算機環境は, MacBook Pro (15-inch, 2018, プロセッサ 2.9 GHz Intel Core i9) であり, OS は macOS High Sierra (バージョン 10.13.6) である. なお, R のバージョンは, R version 3.5.1 (2018-07-02) が利用されている.

```
   310150.603 420993.040    10   b
     3133.495   3242.262    10  a
       47.579   2417.635    10  a
```

この結果において, 計測時間はマイクロ秒 (100 万分の 1 秒: μs) であり, 計測回数は 10 回であることに注意しよう. 計測時間の中央値 (`median`) の列をみることによって, 各関数の実行速度に桁違いの差があることがわかる. この結果からも, R のプログラムにおいて, 組み込み関数を利用することの重要性が理解できよう.

Q 5.10 作業空間内の全てのオブジェクトを消去する関数を作成せよ.

解答: オブジェクトを消去する関数 `rm` の引数 `list` に作業空間内のオブジェクトのリストを与えればよいので, 以下のような関数を定義すればよい:

```
> balus <- function() rm(list = ls())
```

ただし, ここで定義された関数 `balus` を実行すると, 当たり前であるが作業空間内の全てのオブジェクトが消去されてしまうので十分注意してほしい. なお, R の統合開発環境である RStudio には [Environment] ペインに, 作業空間内の全てのオブジェクトを消去する機能をもつボタン (箒の形をしたもの) が用意されている.

第6章　演習問題解答 (6)

Q 6.1 確率変数 Z が標準正規分布 N$(0,1)$ に従うとき, 以下の確率をもとめよ:
(1) $\mathrm{P}(-1 \le Z \le 1)$, (2) $\mathrm{P}(-2 \le Z \le 2)$, (3) $\mathrm{P}(-3 \le Z \le 3)$, (4) $\mathrm{P}(-6 \le Z \le 6)$

解答: (1), (2), (3) については以下のように求めることができる:

```
> pnorm(1) - pnorm(-1)
[1] 0.6826895
> pnorm(2) - pnorm(-2)
[1] 0.9544997
> pnorm(3) - pnorm(-3)
[1] 0.9973002
```

一方, (4) については同様の方法では

```
> pnorm(6) - pnorm(-6)
[1] 1
```

となって正確な値を得ることはできない. そこで, 計算精度を向上させるパッケージ Rmpfr [1] パッケージを読み込み, 関数 integrateR を以下のように利用することによって求めることができる:

```
> library(Rmpfr)
> integrateR(dnorm, -6, 6, rel.tol = 1e-16, verbose = T)
n= 1, 2^n=       2 | I =    3.19153826751499281, abs.err =      3.191538
n= 2, 2^n=       4 | I =   0.676126099765811173, abs.err =      2.515412
n= 3, 2^n=       8 | I =   0.936176992168244015, abs.err =     0.2600509
n= 4, 2^n=      16 | I =    1.00625511146803892, abs.err =    0.07007812
n= 5, 2^n=      32 | I =   0.999900432691466223, abs.err =   0.006354679
n= 6, 2^n=      64 | I =    1.00000027720224383, abs.err =  9.984451e-05
n= 7, 2^n=     128 | I =   0.999999998216769215, abs.err =  2.789855e-07
n= 8, 2^n=     256 | I =   0.999999998026324777, abs.err =  1.904444e-10
n= 9, 2^n=     512 | I =   0.999999998026824821, abs.err =  5.000445e-13
n=10, 2^n=    1024 | I =   0.999999998026825043, abs.err =  2.220446e-16
```

[1] パッケージ Rmpfr は, R Multiple Precision Floating-point Reliable の略であり, R で多倍長精度の浮動小数点演算を実行するためのものである.

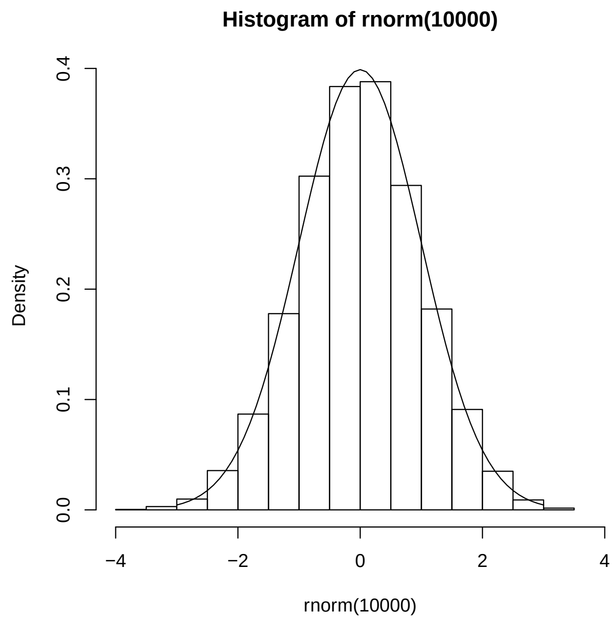
```
n=11, 2^n=      2048 | I =     0.999999998026825043, abs.err =             0
1 with absolute error < 0
```

この結果から, 標準正規分布の $[-6, 6]$ の範囲の面積 (確率) は, 約 0.999999998 となり, この事象は, 約 10 億回に 999, 999, 998 回 (9 億 9 千 9 百 9 十 9 万 9 千 9 百 9 十 8 回) 起こることを表している. 逆にいうと, この事象が起こらない確率は, 約 10 億回に 2 回程度であることに注意しよう.

Q 6.2 各自が定めた μ, σ の値に対する正規分布 $\mathsf{N}(\mu, \sigma^2)$ の乱数を 10000 個生成し, ヒストグラムを描き, さらに密度関数を重ね書きせよ. なお, 密度関数を重ね書きするときには `lines` 関数が利用できることに注意せよ.

解答: $\mu = 0, \sigma = 1$ とする. 以下のように入力することによって描くことができる:

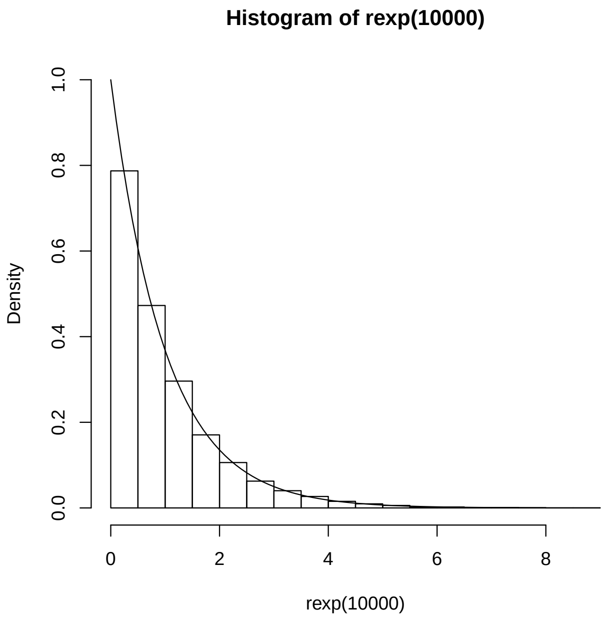
```
> set.seed(12345)
> hist(rnorm(10000), freq=F, xlim = c(-4, 4))
> lines(seq(-3, 3, 0.1), dnorm(seq(-3, 3, 0.1)))
```



Q 6.3 各自が定めた λ の値に対する指数分布 $\mathsf{E_X}(\lambda)$ の乱数を 10000 個生成し, ヒストグラムを描き, さらに密度関数を重ね書きせよ.

解答: $\lambda = 1$ とする. 以下のように入力することによって描くことができる:

```
> set.seed(12345)
> hist(rexp(10000), freq = F, ylim = c(0,1))
> lines(seq(0, 8, 0.1), dexp(seq(0, 8, 0.1)))
```



Q 6.4 確率変数 X の自然対数 $\log X$ が正規分布 N(μ, σ^2) に従うとき, X は対数正規分布 LN(μ, σ^2) に従うといわれる. ある年度の企業の売上高や家計の所得などは対数正規分布に従う例としてしばしばあげられる. 各自が定めた μ, σ に対する対数正規分布の密度関数と分布関数を描く関数を作成せよ. なお, 対数正規分布の R のコーディングは `lnorm` であり, 母数 (μ, σ) のコーディングは, (`meanlog`, `sdlog`) である.

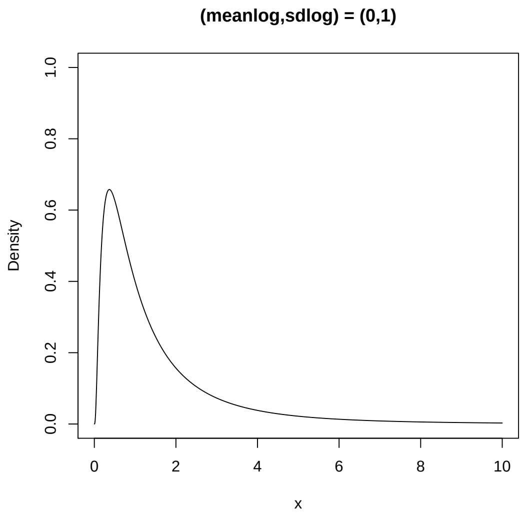
解答: まず, 以下のような関数を用意する:

```
> plot.lnorm <- function(x = seq(0, 10, 0.01), meanlog = 0, sdlog = 1, yup = 1)
+ {
+   plot(x, dlnorm(x, meanlog = meanlog, sdlog = sdlog),
+   type = "l",  ylab = "Density", ylim = c(0, yup))
+   title(paste("(meanlog,sdlog) = (",meanlog, ",", sdlog, ")", sep = ""))
+ }
```

ここで, `x` は密度関数を描く定義域を与える引数であり, `meanlog`, `sdlog` は, 対数正規分布 LN(μ, σ^2) の母数 (μ, σ) のコーディングである. なお, デフォルトとして `meanlog=0`, `sdlog=1` ($(\mu, \sigma) = (0, 1)$) が与えられている. また, `yup` は y 軸の範囲の上限を表す引数である.

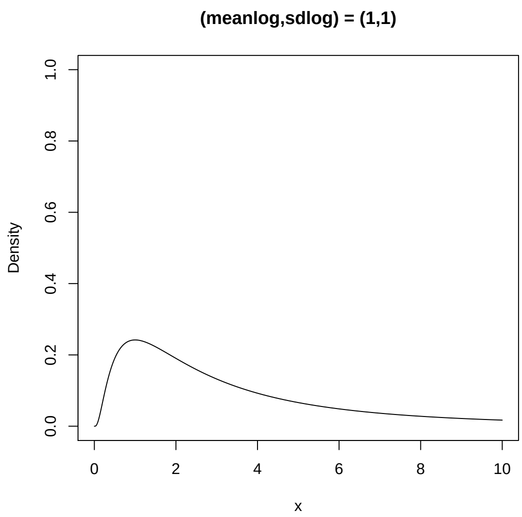
この関数を利用して対数正規分布 LN$(0,1)$ の密度関数は, 以下のように描くことができる:

```
> plot.lnorm()
```



また, 対数正規分布 LN$(1,1)$ の密度関数は, 以下のように描くことができる:

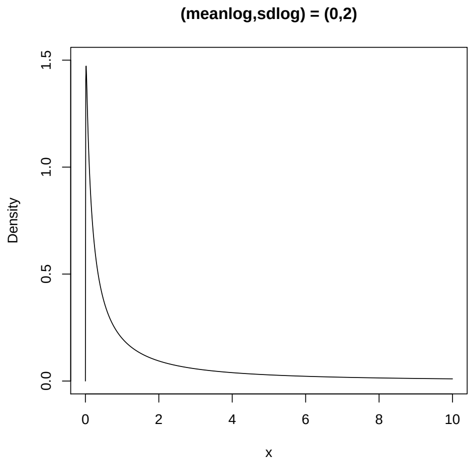
```
> plot.lnorm(meanlog = 1)
```



以上のことから, 対数正規分布は, μ を大きくすることによって,「山」(位置) が右に移動することが予想される.

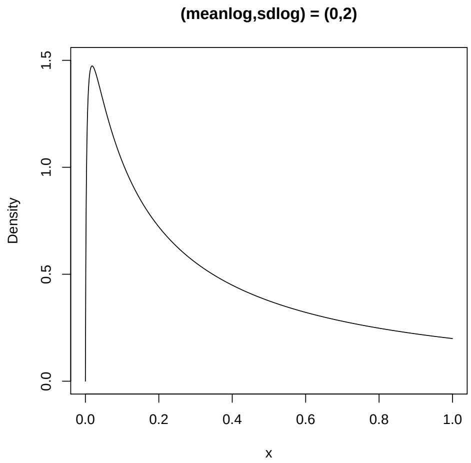
さらに, 対数正規分布 LN$(0,2)$ の密度関数は, 以下のように描くことができる:

```
> plot.lnorm(sdlog = 2, yup = 1.5)
```



原点付近の挙動を詳しくみるためにはプロットの定義域を以下のように変更すればよい:

```
> plot.lnorm(x = seq(0, 1, 0.001), sdlog = 2, yup = 1.5)
```



Q 6.5 確率変数 X, Y がそれぞれ

$$X \sim \mathsf{B_N}(n, p), \quad Y \sim \mathsf{B_E}(k, n-k+1)$$

という分布に従うとき, 以下の等式が成り立つ:

$$\begin{aligned} & \mathrm{P}(X \geq k) = \mathrm{P}(Y \leq p) \\ \Longleftrightarrow & \sum_{x=k}^{n} \binom{n}{x} p^x (1-p)^{n-x} = \frac{1}{B(k, n-k+1)} \int_0^p y^{k-1}(1-y)^{(n-k+1)-1} \mathrm{d}y \end{aligned}$$

このことを R を用いて以下の場合について確かめよ:

(1) $(n, p, k) = (10, 0.5, 2)$ のとき
(2) $(n, p, k) = (10, 0.2, 2)$ のとき
(3) $(n, p, k) = (100, 1/2, 50)$ のとき
(4) $(n, p, k) = (100, 0.2, 50)$ のとき

ただし, $\mathsf{B_E}(k, n-k+1)$ は母数 $(k, n-k+1)$ のベータ分布を表すことに注意しよう.

解答: 以下のようなスクリプトを実行することによって, (1) ∼ (4) の場合に等式が成り立つことがわかる:

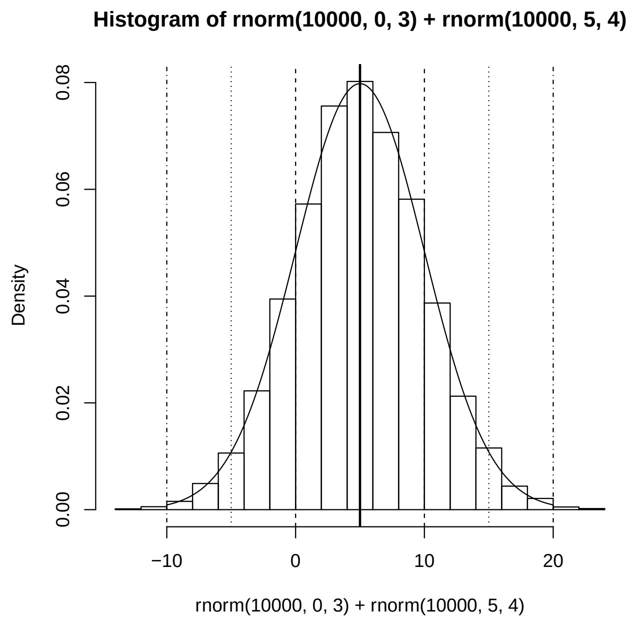
```
> 1 - pbinom(2-1, 10, 1/2); pbeta(1/2, 2, 10-2+1)
[1] 0.9892578
[1] 0.9892578
> 1 - pbinom(2-1, 10, 1/5); pbeta(1/5, 2, 10-2+1)
[1] 0.6241904
[1] 0.6241904
> 1 - pbinom(50-1, 100, 1/2); pbeta(1/2, 50, 100-50+1)
[1] 0.5397946
[1] 0.5397946
> 1 - pbinom(50-1, 100, 1/5); pbeta(1/5, 50, 100-50+1)
[1] 2.139244e-11
[1] 2.139251e-11
```

第7章　演習問題解答 (7)

Q 7.1 確率変数 X と Y がそれぞれ独立に正規分布 $\mathrm{N}(0,3^2)$, $\mathrm{N}(5,4^2)$ に従うとき, 確率変数の和 $X+Y$ の分布は再生性から $\mathrm{N}(5,5^2)$ に従う. このことを乱数を生成し, そのヒストグラムを描くことによって確かめよ.

解答: 独立に正規分布 $\mathrm{N}(0,3^2)$, $\mathrm{N}(5,4^2)$ に従う乱数を 10000 個発生させ, それらの和のヒストグラムを描き, そのヒストグラムに正規分布 $\mathrm{N}(5,5^2)$ の密度関数を重ね書きしたものを以下に与える:

```
> set.seed(12345)
> hist(rnorm(10000, 0, 3) + rnorm(10000, 5, 4), freq = F)
> lines(seq(-10, 20, 0.1), dnorm(seq(-10, 20, 0.1), mean = 5, sd = 5))
> abline(v = 5, lwd = 2)
> abline(v = c(0, 10), lty = 2)
> abline(v = c(-5, 15), lty = 3)
> abline(v = c(-10, 20), lty = 4)
```



この結果から, ヒストグラムと密度関数の重なり具合から, 再生性が成り立っていることが確認

できる．なお，正規分布が $\mu \pm 3\sigma = 5 \pm 15 = (-10, 20)$ の範囲にほぼ全ての確率 (99.7%) をもつことも確認できる．

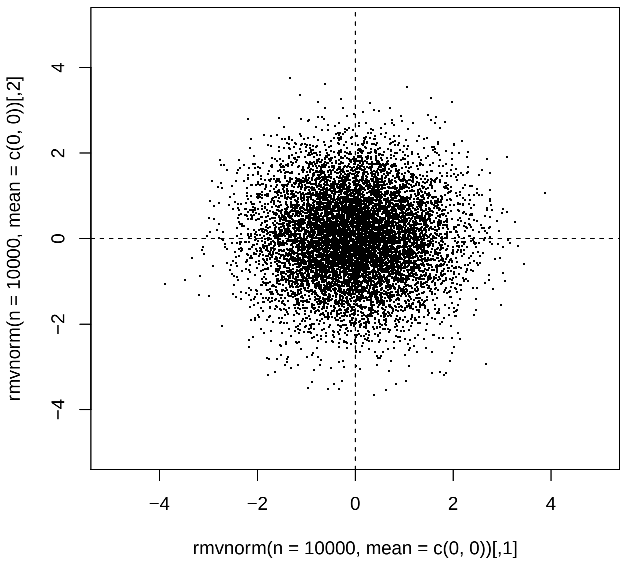
Q 7.2 平均ベクトル $\boldsymbol{\mu}$ と分散共分散行列 $\boldsymbol{\Sigma}$ を適当に与え，2 変量正規分布 $\mathsf{N}_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ に従う乱数を生成し，散布図を描くことによって分布を考察せよ．

解答: まず，平均ベクトルと分散共分散行列として，

$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

を与えた場合 (2 変量標準正規分布) の乱数を生成し，以下のようにプロットする:

```
> library(mvtnorm)
> set.seed(12345)
> plot(rmvnorm(n = 10000,
+          mean = c(0,0)),
+          pch = ".", xlim = c(-5, 5), ylim = c(-5, 5))
> abline(v = 0, h = 0, lty = 2)
```



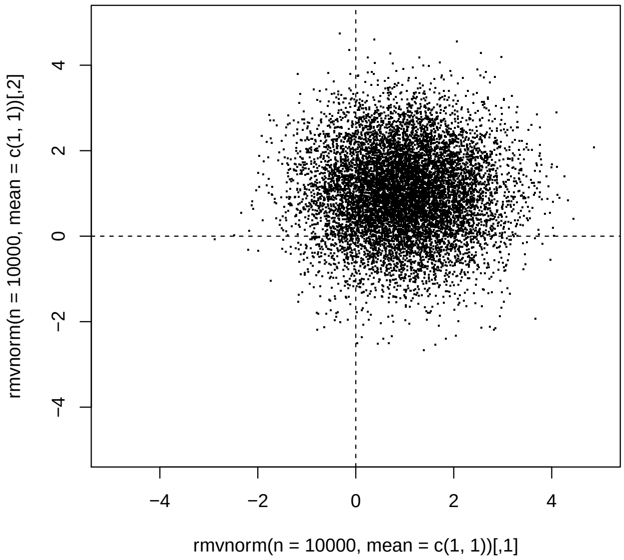
この結果から，2 変量標準正規分布は原点を中心とする同心円上に分布することがわかる．

次に，

$$\boldsymbol{\mu} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

ととった場合は，

```
> set.seed(12345)
> plot(rmvnorm(n = 10000,
+         mean = c(1, 1)),
+         pch = ".", xlim = c(-5, 5), ylim = c(-5, 5))
> abline(v = 0, h = 0, lty = 2)
```



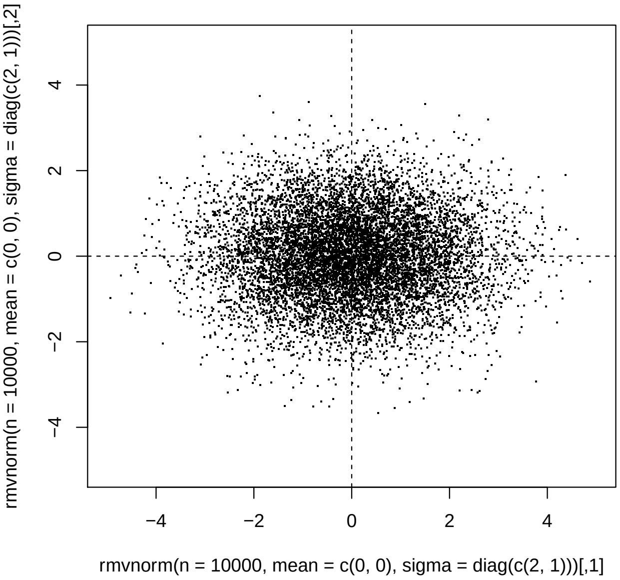
となり, 座標 $(1,1)$ を中心とする同心円上に分布することがわかる．よって, 2 変量標準正規分布の場合と比べて, 中心位置が移動していることがわかる.

また, 2 変量確率変数 (X,Y) が 2 変量正規分布に従うとし, X の分散を 2 に変更した場合, すなわち,

$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \mathbf{I}_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

を与えた場合は,

```
> set.seed(12345)
> plot(rmvnorm(n = 10000,
+         mean = c(0, 0), sigma = diag(c(2, 1))),
+         pch = ".", xlim = c(-5, 5), ylim = c(-5, 5))
> abline(v = 0, h = 0, lty = 2)
```



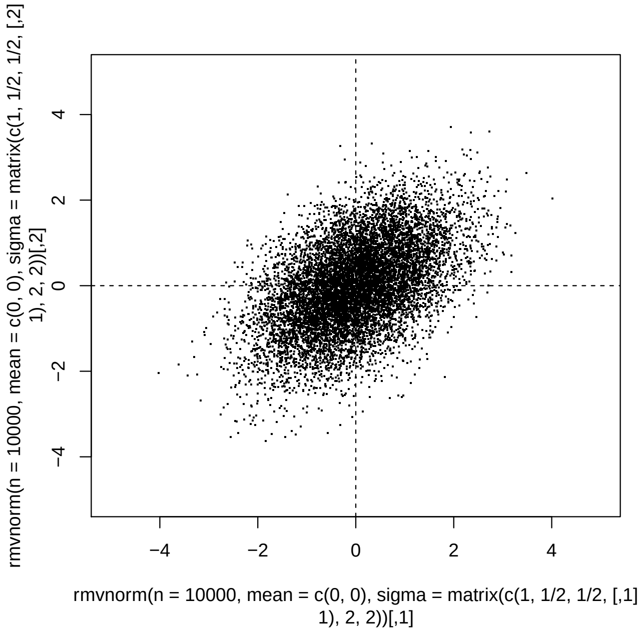
この結果から, 2 変量標準正規分布の場合と比べて x-軸方向に拡がりが大きくなっており, 原点を中心とする横長の楕円上に分布することがわかる. なお, Y の分散を大きくすることによって y-軸方向に拡がりが拡大することも同様の方法によって確かめることができる.

また, 2 変量確率変数 (X, Y) のそれぞれの分散が 1 であり, 相関係数が 1/2 である場合, すなわち,

$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{\Sigma} = \mathbf{I}_2 = \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}$$

を与えた場合は,

```
> set.seed(12345)
> plot(rmvnorm(n = 10000,
+          mean = c(0, 0), sigma = matrix(c(1, 1/2, 1/2, 1), 2, 2)),
+          pch = ".", xlim = c(-5, 5), ylim = c(-5, 5))
> abline(v = 0, h = 0, lty = 2)
```



となり, この場合は, 2 変量標準正規分布の場合と比べて右肩上がりの楕円上に分布することがわかる. なお, 相関係数を負の値, 例えば, $-1/2$ の場合は右肩下がりの楕円になることも同様の方法によって確かめることができる.

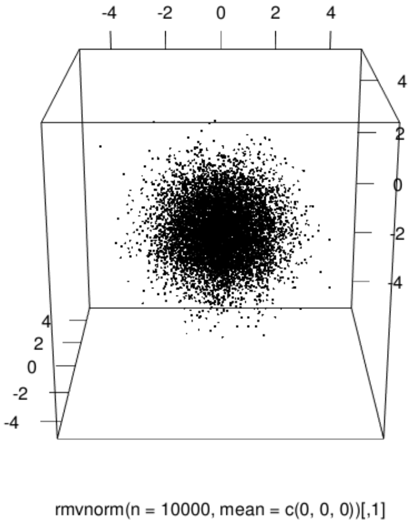
Q 7.3 平均ベクトル $\boldsymbol{\mu}$ と分散共分散行列 $\boldsymbol{\Sigma}$ を適当に与え, 3 変量正規分布 $\mathrm{N}_3(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ に従う乱数を生成し, 3 次元散布図を描くことによって分布を考察せよ.

解答: まず, 平均ベクトルと分散共分散行列として,

$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

を与えた場合 (3 変量標準正規分布) の乱数を生成し, 以下のようにプロットする:

```
> library(rgl)
> library(mvtnorm)
> set.seed(12345)
> plot3d(rmvnorm(n = 10000,
+         mean=c(0, 0, 0)),
+         pch = ".", size = 0.5,
+         xlim = c(-5, 5), ylim = c(-5, 5), zlim = c(-5, 5))
```



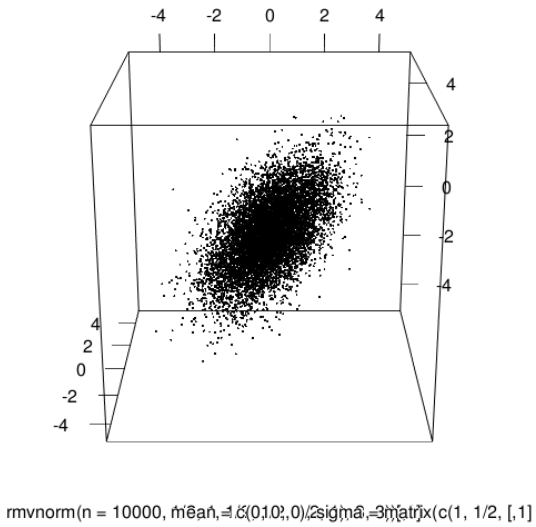
この結果から, 紙面ではわかりづらいかもしれないが, 3 変量標準正規分布は原点を中心とする同心球上に分布することがわかる.

次に, 3 変量確率変数 (X, Y, Z) が 3 変量正規分布に従うとし, 全ての平均 0, 分散が 1, 相関係数が 1/2 の場合, すなわち,

$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \mathbf{I}_3 = \begin{bmatrix} 1 & 1/2 & 1/2 \\ 1/2 & 1 & 1/2 \\ 1/2 & 1/2 & 1 \end{bmatrix}$$

を与えた場合の乱数を生成し, 以下のようにプロットする:

```
> set.seed(12345)
> plot3d(rmvnorm(n = 10000,
+         mean = c(0,0,0),
+         sigma = matrix(c(1, 1/2, 1/2, 1/2, 1, 1/2, 1/2, 1/2, 1), 3, 3)),
+         pch = ".", size = 0.5,
+         xlim = c(-5, 5), ylim = c(-5, 5), zlim = c(-5, 5))
```



この結果から, 3 変量標準正規分布の場合よりも, さらにわかりづらいかもしれないが, この 3 変量正規分布は原点を中心とする「斜めに傾いた」楕円体上に分布することがわかる[1]. このプロットを理解するためには, 紙面では限界があり, 回転や拡大・縮小を行うことが必要となる. 一般に, 3 次元空間の図形を 2 次元の平面で表現した場合は, 奥行きがわからなかったり, 点などが重なってしまう現象 (オーバープロット) が起こり視認性が低下することが知られている. この問題に対して, 回転や拡大・縮小を行うことが可能なグラフィックツールを利用することが一つの解決法である. 幸いここで利用している `rgl` パッケージに用意されている `plot3d` は専用のウィンドウ上でこれらの機能をもつため, 様々な角度とスケールでプロットを眺めることができる. 読者におかれては, ここで与えた母数以外の場合のプロットを是非描いてみてほしい.

[1] いわゆる, ラグビーやアメリカンフットボールで利用されるボールを想像すると理解しやすいもしれない.

第8章　演習問題解答 (8)

Q 8.1 標準正規分布 N$(0,1)$ に従う乱数を n 個生成し, それらの 2 乗を加えることを N 回繰り返し, そのヒストグラムを描き, さらに自由度 n のカイ自乗分布の密度関数を重ね書きする関数を作成せよ.

解答: 以下のような関数を用意する:

```
> plot.sim.chisq <- function(x = seq(0.01, 10, 0.01), N = 100, n = 1, seed = 12345)
+ {
+   set.seed(seed)
+   rn <- matrix(rnorm(N*n)^2, N, n)
+   stat <- apply(rn, 1, sum)
+   if(n == 1)
+   {
+     dens <- dchisq(x, n)
+     dmax <- max(dens)
+     hist(stat, freq = FALSE, xlim = range(x), ylim = c(0, dmax), main = "")
+     title(main = "Histogram of Chisq Random Numbers",
+     sub = paste("(n, N) = (",n, ",", N, ")", sep=""))
+     lines(x, dens)
+   } else
+   dens <- dchisq(x, n)
+   dmax <- max(dens)
+   hist(stat, freq = FALSE, xlim = range(x), ylim = c(0, dmax), main = "")
+   title(main = "Histogram of Chisq Random Numbers",
+   sub = paste("(n, N) = (",n, ", ", N,")", sep = ""))
+   lines(x, dens)
+ }
```

ここで, `x=seq(0.01,10,0.01)` は, プロットする定義域 (デフォルトは区間 $[0.01,10]$ を 0.01 刻みで分割した点の集合) を与える引数であり, `N=100` には繰り返し回数 (デフォルトは 100 回) , `n=1` は, 生成する標準正規乱数の個数, すなわち, カイ自乗分布の自由度 (デフォルトは 1) を与えている. なお, `seed=12345` は乱数のシードである.

この関数を実行することによって, $(n,N)=(1,100),(3,100),(10,100),(10,1000)$ という組み合わせに対する結果を可視化する. 以下のように入力することによって, 図 8.1 のようなプロットが描かれる.

```
> plot.sim.chisq()
> plot.sim.chisq(x = seq(0.01, 15, 0.01), n = 3)
> plot.sim.chisq(x = seq(0.01, 30, 0.01), n = 10)
> plot.sim.chisq(x = seq(0.01, 30, 0.01), n = 10, N = 1000)
```

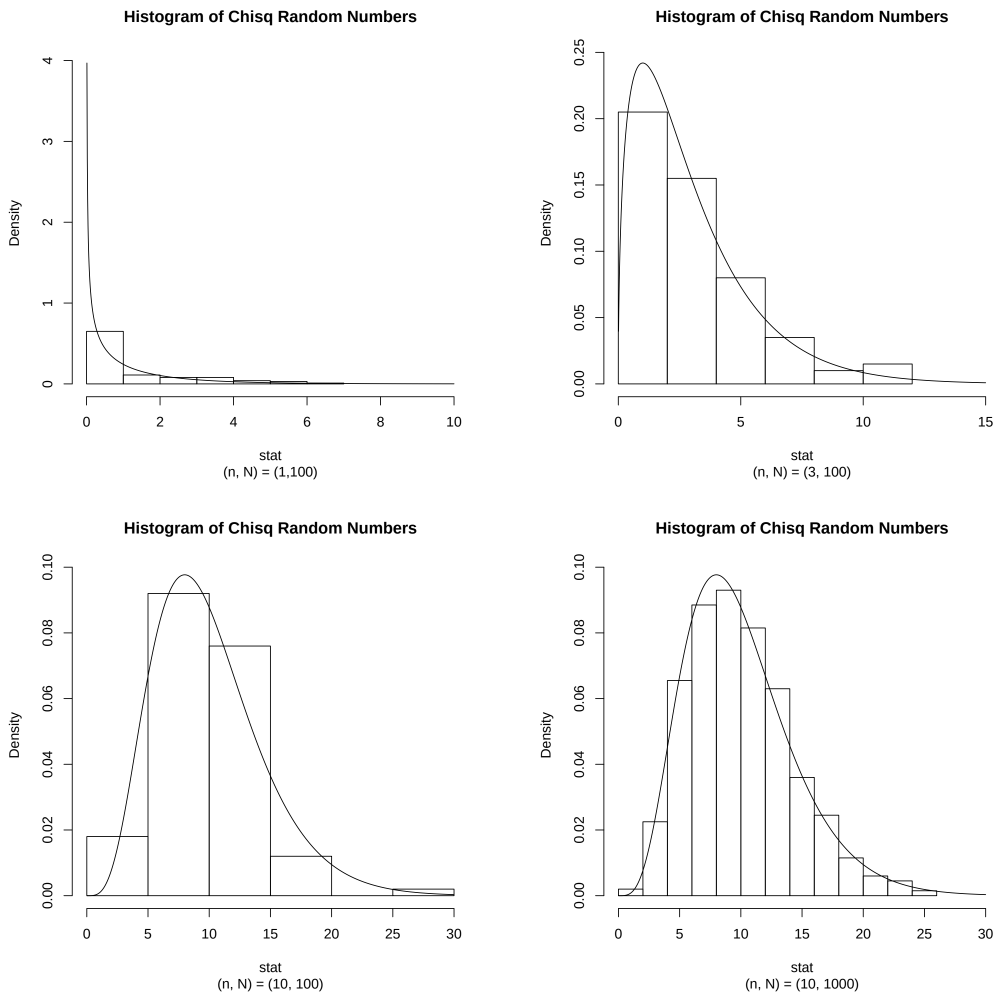


図 8.1: $(n, N) = (1, 100), (3, 100), (10, 100), (10, 1000)$ の場合の結果のプロット

以上の結果から, 独立に標準正規分布に従う乱数の自乗和はカイ自乗分布に従うことを, 乱数を生成することによる経験的な方法 (モンテカルロ法) によって検証することができた. 特に, 繰り返し回数を増やすことによって近似精度が向上することもわかった.

Q 8.2 6.4 節で使った正規分布の与えられた範囲に関する領域を描く関数 `ggplot.pdf.normal.area` (付録 E.4 も参照) を参考にして, 標本分布 (カイ自乗分布, ティー分布, エフ分布) に対する同様の

機能を持つ関数を作成せよ．また, それらを用いて各分布の上側 $100\alpha\%$ 点と上側確率の領域を描け．なお, α の値は各自が適当なものを与えよ．

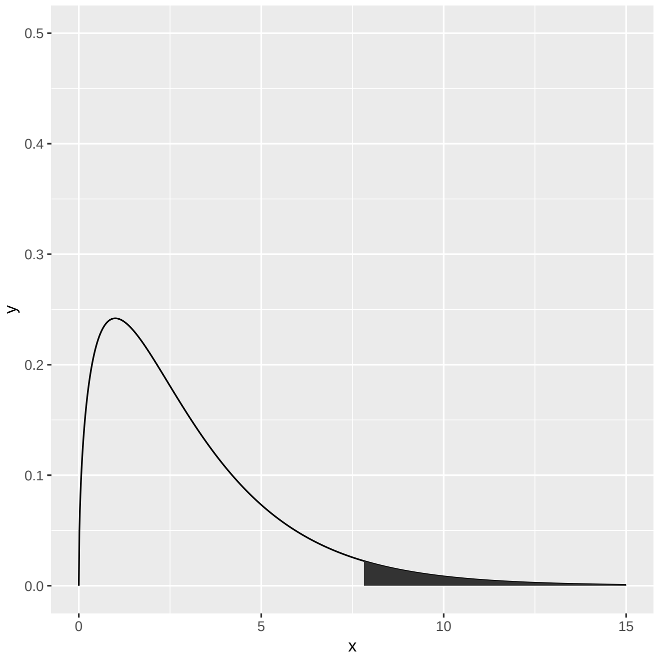
解答: ここでは, カイ自乗分布に対する関数を作成し, いくつかのプロットを行う．なお, この関数と本書にある正規分布に対するものを参考にして, ティー分布とエフ分布に関しては読者諸兄が作成してほしい．

```
> ggplot.pdf.chisq.area <- function(lb, ub, df = 1, limits = c(0, 15), yup = 0.5)
+ {
+   require(ggplot2)
+   x <- seq(limits[1], limits[2], length.out = 1000)
+   xmin <- max(lb, limits[1])
+   xmax <- min(ub, limits[2])
+   areax <- seq(xmin, xmax, length.out = 100)
+   area <- data.frame(x = areax, ymin = 0, ymax = dchisq(areax, df))
+   ggplot() +
+       geom_line(data.frame(x = x, y = dchisq(x, df = df)),
+                 mapping = aes(x = x, y = y)) +
+       geom_ribbon(data = area, mapping = aes(x = x, ymin = ymin, ymax = ymax)) +
+       scale_x_continuous(limits = limits) +
+       ylim(0, yup)
+ }
```

この関数を使って, 自由度 3 のカイ自乗分布の上側 5 % 領域を描いてみよう．

```
> ggplot.pdf.chisq.area(lb = qchisq(0.05, 3, lower.tail = FALSE), ub = 15, df = 3)
```

と入力することによって, 以下のような図を描くことができる:



Q 8.3 確率論・統計学において以下のような主張がある:

> 「標準正規乱数を (1 個) 得るには一様分布 U$(0,1)$ に従う乱数を 12 個加えたものから 6 を引けばよい.」

この主張の理論的な根拠を考察すると共に, 実際に一様乱数を生成して, この主張を検証せよ.

解答: 一般に, 平均 μ, 分散 σ^2 をもつ分布からの独立な確率変数列 $X_1, \ldots, X_n$ に対して, 中心極限定理より,

$$\overline{X}_n \overset{a}{\sim} \mathsf{N}\left(\mu, \frac{\sigma^2}{n}\right) \quad (n \to \infty)$$

が成り立つことに注意すると, 確率変数列 $X_1, \ldots, X_n$ が独立に一様分布 $\mathsf{U}(0,1)$ に従うとき,

$$\mathrm{E}(X_i) = \frac{1}{2}, \quad \mathrm{V}(X_i) = \frac{1}{12}$$

より,

$$\overline{X}_n \overset{a}{\sim} \mathsf{N}\left(\frac{1}{2}, \frac{1}{12n}\right)$$

が漸近的に成り立つ. ここで, $\overline{X}_n = (X_1 + \cdots + X_n)/n$ であることを思い出すと,

$$\left(\overline{X}_n - \frac{1}{2}\right) \overset{a}{\sim} \mathsf{N}\left(0, \frac{1}{12n}\right) \iff (12\overline{X}_n - 6) \overset{a}{\sim} \mathsf{N}\left(0, \frac{12}{n}\right) \iff \frac{12}{n}(X_1 + \cdots + X_n) - 6 \overset{a}{\sim} \mathsf{N}\left(0, \frac{12}{n}\right)$$

となり, $n = 12$ とおくことによって,

$$(X_1 + \cdots + X_{12}) - 6 \overset{a}{\sim} \mathsf{N}(0,1)$$

となり, 題意の主張に関する結果が得られる. ただし, 中心極限定理は n が大きくなったときの漸近的な性質を表すので, ここで, $n=12$ で上の近似が成り立っていることを, 本来であれば解析的に証明する必要があるけれども [1], ここではモンテカルロ法を用いてシミュレーションによって検証する.

シミュレーションを実行するためのアルゴリズムを以下に与える.

一様分布に関するシミュレーションのためのアルゴリズム

(US1) U$(0,1)$ にしたがう大きさ 12 の一様乱数を N 組生成し, 各組ごとの値を加えたものから 6 を引いたものを求める.

(US2) ステップ (US1) によって得られた N 個の値のヒストグラムを描画する.

このアルゴリズムに沿ってシミュレーションを実行する R 関数を以下のように作成する:

```
> sim.uniform.sum <- function(n = 12, N = 100, seed = 12345)
+ {
+   set.seed(seed)
+   X <- matrix(runif(n*N, 0, 1), N, n)
+   vec.sum <- apply(X, 1, sum) - 6
+   list(rn = X, vec.sum = vec.sum, n = n, N = N)
+ }
```

一様分布のパラメータを $(a,b)=(0,1)$, 標本の大きさを $n=12$ で固定し, シミュレーション回数を $N=100, 1000, 10000, 100000$ と変化させることによって分布がどのように変化するかをこの関数を利用して検証しよう.

以下のように入力することによって各シミュレーションの結果をオブジェクト `obj.unif.sum1` $\sim$ `obj.unif.sum4` に保存する.

```
> obj.unif.sum1 <- sim.uniform.sum(N = 100)
> obj.unif.sum2 <- sim.uniform.sum(N = 1000)
> obj.unif.sum3 <- sim.uniform.sum(N = 10000)
> obj.unif.sum4 <- sim.uniform.sum(N = 100000)
```

この結果を図示するための関数を以下のように用意する:

```
> plot.sim.uniform.sum <- function(obj)
+ {
```

[1] 詳細については, 例えば,

S. S. Wilks (1962), *Mathematical Statistics, Second Edition*, John Wiley & Sons, Inc, New York, p. 204, Example

に引用されているラプラスの結果を参照してほしい. (この文献の存在と詳細な計算結果に関して, 大阪大学名誉教授の稲垣宣生先生にご教示いただいた. 感謝の意を表す.)

```
+   n <- obj$n
+   N <- obj$N
+   mu <- 0
+   sigma <- 1
+   hist(obj$vec.sum,freq = FALSE,
+        xlim = c(mu - 4*sigma, mu + 4*sigma),
+        ylim = c(0, 2/(sqrt(2 * pi) * sigma)),
+        main = paste("(n,N)=(", n, ",", N, ")", sep = ""))
+   x <- seq(mu - 4*sigma, mu + 4*sigma, 0.01)
+   lines(x, dnorm(x, mu, sigma))
+   abline(v = mu, lwd = 2, col = 2)
+ }
```

この関数に上で実行したシミュレーションのオブジェクトを以下のように与えることによって描くと, 図 8.2 のように結果が得られる.

```
> plot.sim.uniform.sum(obj.unif.sum1)
> plot.sim.uniform.sum(obj.unif.sum2)
> plot.sim.uniform.sum(obj.unif.sum3)
> plot.sim.uniform.sum(obj.unif.sum4)
```

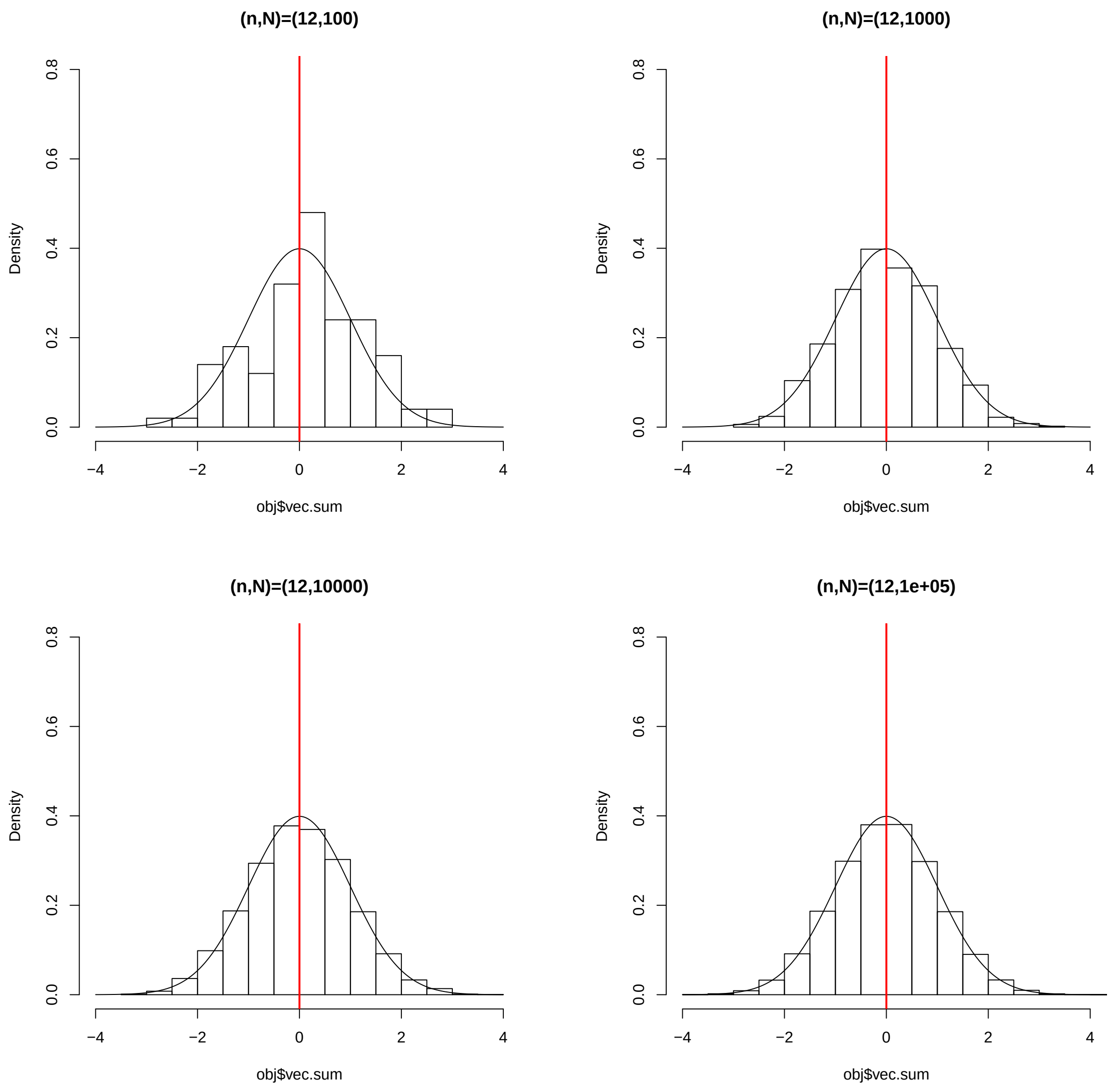
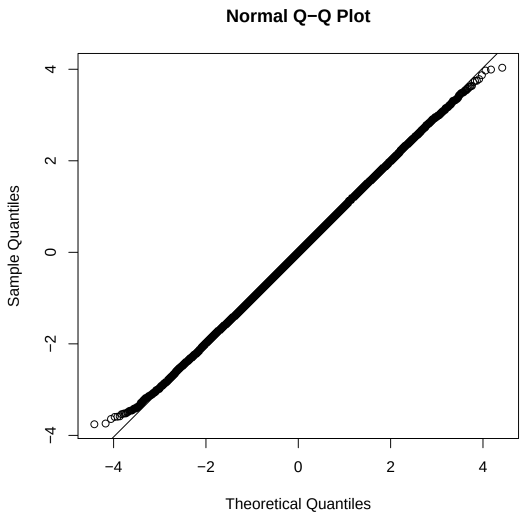


図 8.2: $N = 100, 1000, 10000, 100000$ と変化させたときの一様乱数の和のヒストグラムの変化

以上のシミュレーション結果から, $N = 100000$ 程度発生させたものであれば, 標準正規分布に従う乱数と考えても良さそうである. 検証のために, この場合の正規 Q-Q プロットを以下のように入力することによって与える:

```
> qqnorm(obj.unif.sum4$vec.sum)
> qqline(obj.unif.sum4$vec.sum)
```



以上の結果は, 標準正規乱数と考えることを肯定するものといえる.

第 9 章　演習問題解答 (9)

Q 9.1 確率変数に関する分散公式 (6.9) と同様に, データの分散に関しても以下のような分散公式が成り立つ [1]:

$$s^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \overline{x}^2 \tag{9.1}$$

適当なデータを用いて分散公式が成り立つことを R を用いて確かめよ.

解答: 新生児の体重データ `weight` と自作関数 `svar` を利用して, 以下のように確かめることができる:

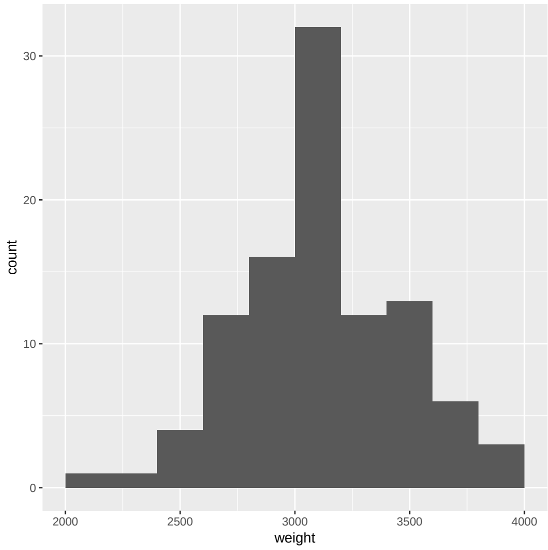
```
> svar(weight)
[1] 143912
> mean(weight^2) - mean(weight)^2
[1] 143912
```

Q 9.2 新生児のデータフレーム `babies.frame` の体重 `weight` を `ggplot2` パッケージに付属の関数 `ggplot` を用いてヒストグラム, ボックスプロット, 正規 Q-Q プロット等を作成して可視化せよ. なお, `ggplot2` パッケージと関数 `ggplot` の利用については 4.2 節を参照してほしい.

解答: ヒストグラムは以下のように描くことができる:

```
> library(ggplot2)
> ggplot(babies.frame, aes(x = weight)) +
+ geom_histogram(breaks = seq(2000, 4000, by = 200))
```

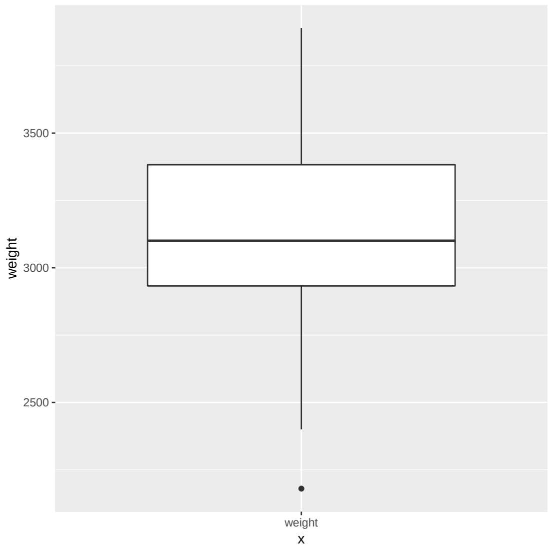
[1] データの分散公式の証明と幾何学的な説明が与えられた補足文書 (補遺) のファイルが裳華房の本書に関するサイトからダウンロード可能となる予定である.



ここで, 階級を 2000 (g) から, 4000 (g) まで階級幅 200 (g) で指定している (`breaks=seq(2000, 4000, by=200)`) ことに注意しよう.

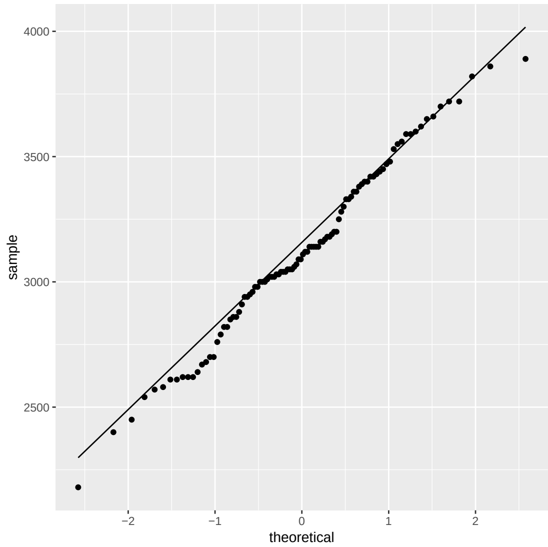
ボックスプロットは以下のように描くことができる:

```
> ggplot(babies.frame, aes(x= "weight", y = weight)) + geom_boxplot()
```



正規 Q-Q プロットは以下のように描くことができる:

```
> p <- ggplot(babies.frame, aes(sample = weight))
> p + stat_qq() + stat_qq_line()
```



この結果から, データフレーム babies.frame の新生児の体重 (weight) も正規分布に従っていることが肯定された.

Q 9.3 GGally パッケージに付属する関数 ggpairs を利用して新生児のデータフレーム babies.frame に対する対散布図を性別によって色分けすることによって作成せよ. この結果から, 各変量間の性別に関する散布図の差異と, 各分布の差異があるかどうかを考察せよ.

解答: 関数 ggpairs の引数 mapping にパッケージ ggplot2 に付属の関数 aes の引数 color に因子ベクトル gender を与えることによって実現できる. (図 9.1 参照.)

```
> library(ggplot2)
> library(GGally)
> ggpairs(babies.frame, mapping = aes(color = gender))
```

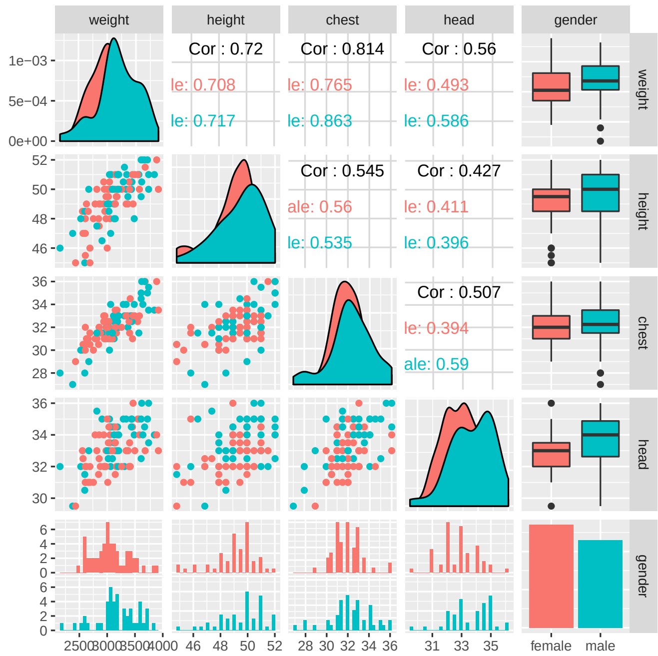


図 9.1: 新生児の対散布図: 性別によって色分けしたもの, ggpairs を利用

この結果から, 各変量間の性別ごとの散布図の差異と, 各分布の差異に関する情報が同時に得られる. すなわち, 体重と胸囲の相関は男子の方が強いことと, 頭囲の分布については男子の方が大きい位置で分布するという情報を同時に得ることができる.

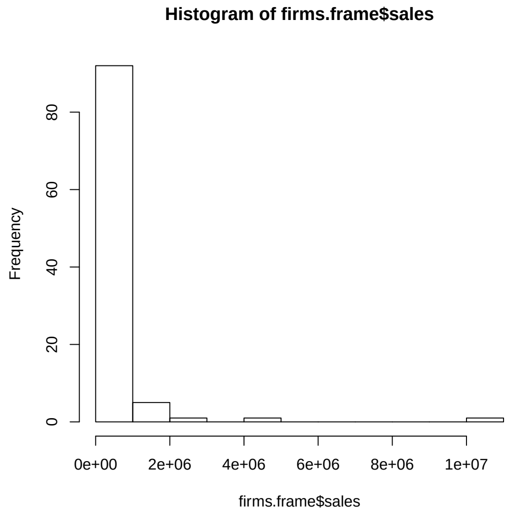
Q 9.4 本章で学んだ 1 変量データの要約と可視化を演習問題 Q 3.1 で読み込んだデータフレーム `firms.frame` の `sales` に対して行え. また, その結果から `sales` はどのような分布に従うかを考えよ.

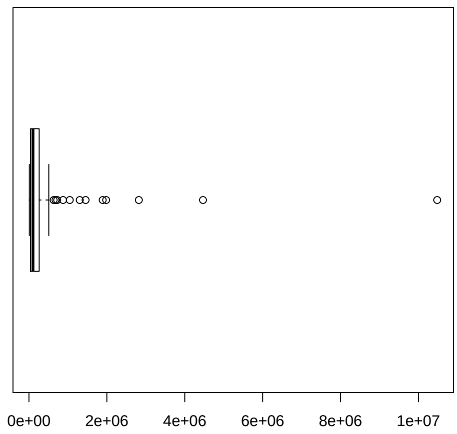
解答: まず, データの要約を行う:

```
> summary(firms.frame$sales)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
    9104    40077   101808   398452   256802 10481166
```

この結果から, 最小値 9104 (約 9 億円) に対して, 最大値 10481166 (約 10.5 兆円) であり, 大きな開きがあることに注意しよう. また, 平均値が 398452.5 (約 3985 億円) に対して, 中央値は 101807.5 (約 1018 億円) であり, この結果から右に歪んだ分布に従うことがわかる. では, このことをヒストグラムとボックスプロットを描くことによって以下のように確かめる:

```
> hist(firms.frame$sales)
> boxplot(firms.frame$sales, horizontal = TRUE)
```





これらのプロットから, 売上高は右に歪んだ分布に従うことがわかる.

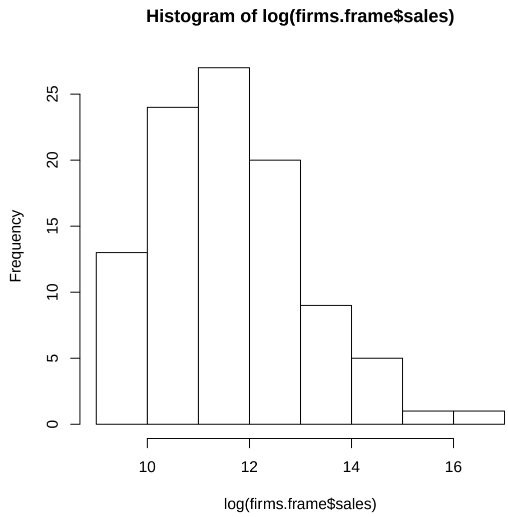
研究: 歪んだデータは, 対数をとることによって対称化できる場合があることが知られており [2], 売上高の対数をとったもののヒストグラムとボックスプロットを描くことによって可視化し, 確かめる:

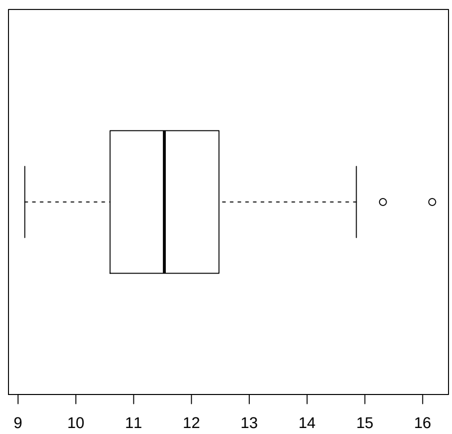
```
> hist(log(firms.frame$sales))
> boxplot(log(firms.frame$sales), horizontal = TRUE)
```

[2] 例えば,

J. W. Tukey (1977) *Exploratory Data Analysis*, Addison-Wesley Publishing Co.

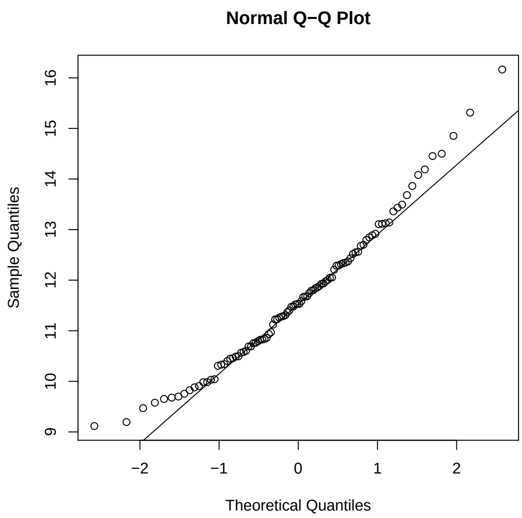
を参照してほしい.





これらのプロットから, 若干右に歪んでいるものの対称な分布に近づけることができた. なお, 演習問題 **Q 6.4** でも扱ったが, 一般に, 観測 (確率変数) X の対数をとったものが正規分布に従う場合は, 対数正規分布 LN(μ, σ^2) に従うが, 売上高の対数をとったものの正規 Q-Q プロットを描くと,

```
> qqnorm(log(firms.frame$sales))
> qqline(log(firms.frame$sales))
```



となり, 分布の「裾」の部分で当てはまりはあまりよくないので, 対数正規分布に従うことを早計

には判断できないが, 分布の中心部分ではある程度当てはまっているように見受けられる [3].

Q 9.5 本章で学んだ多変量データの要約と可視化を演習問題 Q 3.1 で読み込んだデータフレーム firms.frame に対して行え.

解答: まず, 関数 summary を利用して要約する:

```
> firms.frame$code <- as.factor(firms.frame$code)
> summary(firms.frame)
     sales            employees           assets          code  
 Min.   :    9104   Min.   :    73   Min.   :   6718   1:57  
 1st Qu.:   40077   1st Qu.:   845   1st Qu.:  43398   2:43  
 Median :  101808   Median :  2305   Median :  96125         
 Mean   :  398452   Mean   :  7877   Mean   : 440803         
 3rd Qu.:  256802   3rd Qu.:  6295   3rd Qu.: 298850         
 Max.   :10481166   Max.   :173155   Max.   :9011823         
```

ここで, 業種分類コード (code) [4] のオブジェクトの型を因子型 (factor) に変換していることに注意しよう. 前問では, 売上高 (sales) のみを考察し, 右に歪んでいることをみたが, この結果から, 従業員数 (employees) と資産合計 (assets) も右に歪んだ分布からのデータであることがわかる [5]. また, データには, 製造業の方が非製造業よりも企業数が多く含まれていることもわかる.

次に, 分類コード (code) を除いた変量の相関係数行列を求める:

```
> cor(firms.frame[,-4])
              sales employees    assets
sales     1.0000000 0.6042341 0.9308494
employees 0.6042341 1.0000000 0.5388860
assets    0.9308494 0.5388860 1.0000000
```

この結果から, 全ての変量間に正の相関があることがわかる. 特に, 売上高 (sales) と資産合計 (assets) の間には高い正の相関 (0.930849) があることがわかる.

さらに, 業種で分類することによって, 相関係数の値が変化するかどうかを検討する. 以下のように関数 subset を使って, 製造業と非製造業に属する企業を分けて変量間の相関係数行列を求める:

```
> # 製造業
> cor(subset(firms.frame,code == 1, select = -code))
```

[3] もし売上高が対数正規分布に従うかを検証するためには, さらにデータを抽出したものを用意する必要があろう. 実は, このデータのもとになった日経 NEEDS 財務データでは, 2012 年 3 月 31 日決算の 1,162 社のデータで調べたところ, やはり分布の裾の部分では若干の問題があるものの, ほぼ対数正規分布に従うことがわかっている.

[4] 日経業種分類の大分類を表すコードであり, 1 が製造業, 2 が非製造業を表す.

[5] 特に, このデータは東京証券取引所第一部上場企業に関するものであるが, 従業員数については, 73 人の企業もあれば, 173,155 人 (17 万人以上) の企業もあることに注意が必要である.

```
               sales employees    assets
sales      1.0000000 0.9165394 0.9267649
employees 0.9165394 1.0000000 0.7737074
assets     0.9267649 0.7737074 1.0000000
> # 非製造業
> cor(subset(firms.frame,code == 2, select = -code))
               sales employees    assets
sales      1.0000000 0.9080354 0.9325856
employees 0.9080354 1.0000000 0.8422621
assets     0.9325856 0.8422621 1.0000000
```

これらの結果から, 相関係数は総じて高くなり, 業種で分類することによって, より明確に相関関係をみることができる.

以上の結果を関数 ggpairs を利用して色付きの対散布図を描くことによって可視化し, グラフィカルに考察しよう. 関数 ggpairs のオプション mapping にパッケージ ggplot2 に付属の関数 aes の引数 color に因子ベクトル code を与えることによって実現できる (図 9.2 を参照).

```
> library(ggplot2)
> library(GGally)
> ggpairs(firms.frame, mapping = aes(color = code))
```

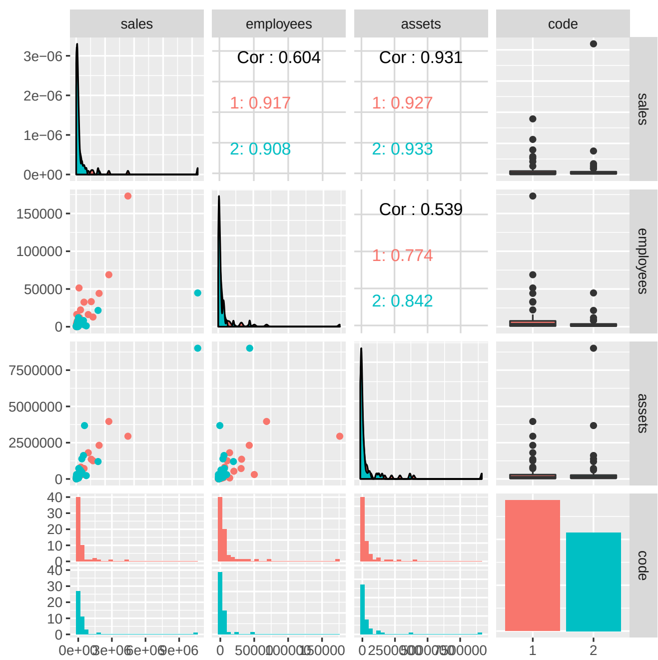


図 9.2: 企業の財務データの対散布図: 性別によって色分けしたもの, ggpairs を利用

この結果から, 製造業の企業が非製造業の企業より多く存在することが (行列の成分に倣って) (4,4) ブロックの棒グラフからわかる. また, 売上高, 従業員数, 資産合計は業種によって分類しても右に歪みをもっていることが対角ブロックの推定された密度関数のプロットと, (1,4), (2,4), (3,4) ブロックのボックスプロット, さらに (4,1), (4,2), (4,3) ブロックの業種別ヒストグラムから分かる. さらに, 各変量間の相関は, 製造・非製造を合併したものよりも, 業種で分けたものの方が高くなる傾向にあるということが, 非対角ブロック (1,2), (1,3), (2,3) の相関係数の値からわかる. なお, この傾向は非対角ブロック (2,1), (3,1), (3,2) の対散布図の結果からも若干推察できる.

Q 9.6 本章で学んだデータの要約と可視化を演習問題 Q 3.3 で各自が読み込んだデータフレームに対して行え.

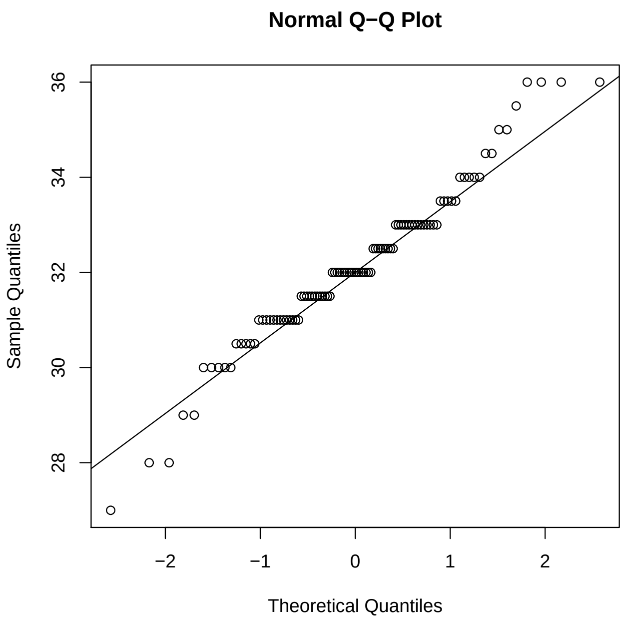
(読者が各自試して頂ければ幸いである.)

第10章　演習問題解答(10)

Q 10.1 新生児のデータフレーム babies.frame における胸囲 (chest) の母平均と母分散の 95% 信頼区間を求めよ.

解答: まず胸囲の分布を確かめるために正規 Q-Q プロットを描く:

```
> qqnorm(babies.frame$chest)
> qqline(babies.frame$chest)
```



この結果から, 胸囲は測定単位に関して丸められていることがうかがえるものの正規分布に従うものと仮定する.

次に, 関数 t.test を利用して母平均の 95% 信頼区間を以下のように求めることができる:

```
> t.test(babies.frame$chest)$conf.int
[1] 31.78866 32.45134
attr(,"conf.level")
[1] 0.95
```

また, 関数 var.interval を利用して母分散の 95% 信頼区間を以下のように求めることができる:

```
> var.interval(babies.frame$chest)$conf.interval
[1] 2.149632 3.763031
```

Q 10.2 母比率 p の簡易的な区間推定

$$[\overline{p}_{\mathrm{L}n},\ \overline{p}_{\mathrm{U}n}] := \left[\overline{\varepsilon}_n - \frac{z(\alpha/2)}{\sqrt{n}}\sqrt{\overline{\varepsilon}_n(1-\overline{\varepsilon}_n)},\quad \overline{\varepsilon}_n + \frac{z(\alpha/2)}{\sqrt{n}}\sqrt{\overline{\varepsilon}_n(1-\overline{\varepsilon}_n)}\right]$$

を行うための R 関数を作成し, 新生児のデータに対する男子の出生率の推定を行え.

解答: 以下のように関数 prop.interval.simple を定義する:

```
> prop.interval.simple <- function(x,alpha = 0.05)
+ {
+   m <- mean(x)
+   n <- length(x)
+   propL <- m - qnorm(1 - alpha/2)*sqrt(m*(1 - m))/sqrt(n)
+   propU <- m + qnorm(1 - alpha/2)*sqrt(m*(1 - m))/sqrt(n)
+   list(n = n, alpha = alpha, m = m, conf.interval = c(propL, propU))
+ }
```

この関数を利用して以下のように新生児のデータに対して男子の出生率を区間推定する:

```
> babies.gender <- as.numeric(babies.frame$gender == "male")
> prop.interval.simple(babies.gender)$conf.interval
[1] 0.3623159 0.5576841
```

この結果と, R に標準的に用意されている関数 prop.test の結果を比較すると,

```
> prop.test(sum(babies.gender), length(babies.gender), correct = FALSE)$conf.int
[1] 0.3656081 0.5573514
attr(,"conf.level")
[1] 0.95
```

となり, こちらの結果の方が若干区間幅が狭くなっていることがわかる. すなわち, 狭い区間で同じ信頼率をもつ区間を作ることができたことになり, 簡易的なものよりもより精度が高いことがわかる.

Q 10.3 R には正規近似を用いずに母比率 p に関する正確な区間推定を行う関数 `binom.test` が用意されている. この関数を利用して新生児のデータに対する男子の出生率の推定を行え.

解答: R に標準的に用意されている関数 `binom.test` を利用して区間推定を行うと,

```
> binom.test(sum(babies.gender), n = length(babies.gender))$conf.int
[1] 0.3598434 0.5625884
attr(,"conf.level")
[1] 0.95
```

となる.

第11章　演習問題解答(11)

Q 11.1 3.3 節で読み込んだ新生児に関するデータフレーム `babies.frame` は 1989 年 12 月に採集されたものである．一方, 昭和 45 年 (1970 年) に実施された乳幼児身体発育調査の結果 [1] によると, 男子の新生児の平均身長は 50.2 cm であると報告されている．約 20 年後に男子の平均身長が変化したかどうかを有意水準 $\alpha = 0.05$ で仮説検定せよ．ただし, 新生児の身長は正規分布に従うものとする.

解答: 男子の新生児の平均身長 μ に関して以下の仮説を検定する:

$$\begin{cases} H_0: & \mu = 50.2 \\ H_1: & \mu \neq 50.2 \end{cases}$$

題意より新生児の身長は, 正規分布に従っていると考えられるので, 正規母集団で母分散を未知として検定を行う．関数 `t.test` を利用し有意水準を $\alpha = 0.05$ として以下のように検定を行うことができる:

```
> with(babies.frame,
+ t.test(height[gender == "male"], mu = 50.2, alternative = "two.sided"))
        One Sample t-test

data:  height[gender == "male"]
t = -2.4475, df = 45, p-value = 0.01835
alternative hypothesis: true mean is not equal to 50.2
95 percent confidence interval:
 49.10227 50.09338
sample estimates:
mean of x
 49.59783
```

ここで, 帰無仮説が $H_0: \ \mu = 50.2$ であることを引数 `mu=50.2` で与えており, 対立仮説 $H_1: \mu \neq 50.2$ であることを引数 `alternative="two.sided"` で与えている．この結果から, 自由度が 45 (`df=45`) であることがわかり, 検定統計量の実現値 (ティー値) が

$$t = -2.4475$$

[1]厚生労働省の Web ページ (`http://www.mhlw.go.jp/`) を参照してほしい.

で与えられることがわかる．棄却域は

$$
\begin{aligned}
R &= (-\infty, -t_{n-1}(\alpha/2)] \text{ または } [t_{n-1}(\alpha/2), \infty) \\
&= (-\infty, -t_{45}(0.025)] \text{ または } [t_{45}(0.025), \infty) \\
&= (-\infty, -2.0141] \text{ または } [2.0141, \infty)
\end{aligned}
$$

であることから,

$$t \in R$$

となり, 帰無仮説 H_0 は棄却できる．よって, 男子の新生児の平均身長は有意水準 $\alpha = 0.05$ (5%) で 50.2 cm から変化したといえる．

Q 11.2 平成 12 年 (2000 年) に実施された乳幼児身体発育調査の結果によると, 女子の新生児の身長の分散は 2.0^2 cm^2 であると報告されている．約 20 年前に調査されたデータフレーム `babies.frame` を使って身長の分散が変化しているかどうかを有意水準 $\alpha = 0.05$ で仮説検定せよ．ただし, 新生児の身長は正規分布に従うものとする．

解答: 女子の新生児の体重の母分散を σ^2 (cm^2) とし, 仮説

$$
\begin{cases}
H_0: & \sigma^2 = 2^2 (= \sigma_0^2) \\
H_1: & \sigma^2 \neq 2^2
\end{cases}
$$

の検定を行う．R には (単一の母集団に対する) 母分散の検定を行う関数は用意されていないため, 付録に与えられている関数 `one.var.test` を利用する．有意水準を $\alpha = 0.05$, 正規母集団として R 関数 `one.var.test` を利用して以下のように検定を行う:

```
> with(babies.frame,
+ one.var.test(height[gender == "female"], sigma2 = 2^2, alternative = "two.sided"))
$u2
[1] 2.459906

$n
[1] 54

$statistic
[1] 32.59375

$df
[1] 53

$critical.value
[1] 34.77633 75.00186

$p.value
[1] 0.02456547

$alternative
[1] "two.sided"

$sigma2
[1] 4

$alpha
[1] 0.05
```

ここで, 帰無仮説が H_0 : $\sigma^2 = 2^2$ であることを引数 `sigma2=2^2` で与えており, 対立仮説 H_1 : $\sigma^2 \neq 2^2$ であることを引数 `alternative="two.sided"` で与えている.

この結果から, 自由度 (`$df`) が $n-1 = 53$ であることがわかり, 検定統計量の実現値 (`$statistic`) が,

$$x^2 = 32.59375$$

で与えられることがわかる.

棄却域は

$$\begin{aligned} R &= (0, \chi_{n-1}(1-\alpha/2)] \quad \text{または} \quad [\chi_{n-1}(\alpha/2), \infty) \\ &= (0, \chi_{53}(0.975)] \quad \text{または} \quad [\chi_{53}(0.025), \infty) \\ &= (0, 34.77633] \quad \text{または} \quad [75.00186, \infty) \end{aligned}$$

であることから,

$$x^2 \in R$$

となり, 帰無仮説 H_0 は棄却される. よって, 女子の新生児の身長の母分散は有意水準 $\alpha = 0.05$ (5%) で 2^2 cm^2 から変化しているといえる.

Q 11.3 データフレーム `firms.frame` を利用して, 東京証券取引所第一部上場企業に関して製造業の比率は 1/2 よりも高いかどうかを有意水準 $\alpha = 0.05$ で仮説検定せよ.

解答: 製造業の比率 p に関する仮説

$$\begin{cases} H_0 : & p = \frac{1}{2} (= p_0) \\ H_1 : & p > \frac{1}{2} \end{cases}$$

の検定を行う.

検定を行う前に, 以下のようにして業種コードを `1` を `1`, `2` を `0` に変換する:

```
> firms.code <- abs(as.numeric(firms.frame$code) - 2)
> firms.code
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [33] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
 [65] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [97] 0 0 0 0
```

この変換された業種コードのオブジェクト `firms.code` を, 付録に与えられている関数 `prop.norm.app.test` を利用し, 有意水準を $\alpha = 0.05$ (デフォルト値) として以下のように検定を行う:

```
> prop.norm.app.test(x = sum(firms.code), n = length(firms.code),
+ p = 0.5, alternative = "greater")
```

```
$x
[1] 57

$n
[1] 100

$statistic
[1] 1.4

$critical.value
[1] 1.644854

$p.value
[1] 0.08075666

$alternative
[1] "greater"

$p
[1] 0.5

$phat
[1] 0.57

$alpha
[1] 0.05
```

ここで, 帰無仮説が $H_0:\ p = \frac{1}{2}$ であることを引数 `p=0.5` で与えており, 対立仮説 $H_1:\ p > \frac{1}{2}$ であることを, 引数 `alternative="greater"` で与えている. この結果から, 検定統計量の実現値 (`$statistic`) が,

$$z = 1.4$$

で与えられることがわかる. 棄却域は `$critical.value` が `1.644854` で与えられるという結果から,

$$R = [z(\alpha), \infty) = [1.644854, \infty)$$

であり,

$$z \notin R$$

より, 帰無仮説 H_0 は棄却できない. よって, 製造業の比率は有意水準 $\alpha = 0.05$ (5%) で $\frac{1}{2}$ よりも高いとはいえない.

第12章　演習問題解答 (12)

Q 12.1 3.3 節で読み込んだ新生児に関するデータフレーム `babies.frame` の体重 `weight` に関して以下の設問に答えよ.

(1) 性別ごとのボックスプロットを描き, 比較せよ.

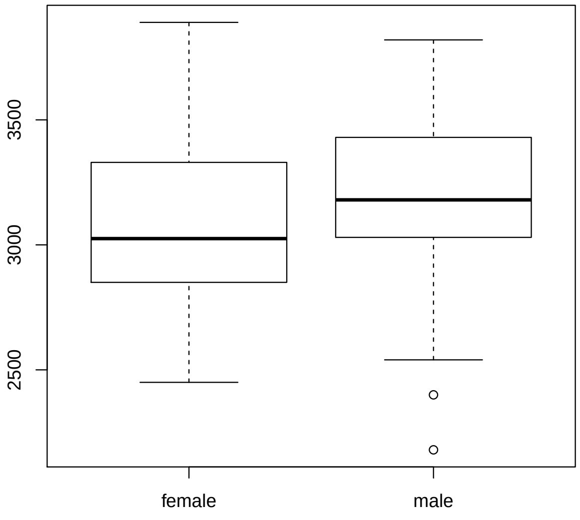
(2) 母平均の差 $\delta = \mu_1 - \mu_2$ に関する以下の仮説検定を行え:

$$\begin{cases} H_0: & \delta = 0 \\ H_1: & \delta < 0 \end{cases}$$

ただし, μ_1, μ_2 は, それぞれ, 女子, 男子の新生児の平均体重とする.

解答: (1) ボックスプロットを描くと,

```
> boxplot(weight ~ gender, babies.frame)
```



となり, この結果から, 男子の体重よりも女子の体重の方が若干軽いことがわかる.

(2) 第 12 章で与えた結果から, 新生児に関する女子と男子の体重の母分散は等分散とする. 関数 `t.test` を用いて検定を以下のように行う:

```
> t.test(weight ~ gender, babies.frame, var.equal = TRUE,alternative="less")
        Two Sample t-test

data:  weight by gender
t = -1.7385, df = 98, p-value = 0.04264
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf -5.428807
sample estimates:
mean in group female   mean in group male
            3064.074             3185.217
```

有意水準は明示されていないがデフォルトで $\alpha = 0.05$ が設定されている. `data:  weight by gender` より, (新生児の) 体重を性別によって分類している. また, `alternative hypothesis: true difference in means is less than 0` より, 対立仮説が $\delta < 0$ の検定を実行していることがわかる. また, `df = 98` より自由度が $n_1 + n_2 - 2 = 98$ で与えられることがわかり, さらに, `t = -1.7385` より, 検定統計量の実現値が

$$t = -1.7385$$

で与えられることがわかる.

棄却域は,

$$\begin{aligned} R &= (-\infty, -t_{n_1+n_2-2}(\alpha)] \\ &= (-\infty, -t_{98}(0.05)] \\ &= (-\infty, -1.66055] \end{aligned}$$

であるので,

$$t \in R$$

となり, 帰無仮説 H_0 は棄却される. よって, 有意水準 $\alpha = 0.05$ で新生児の体重は性別間の母平均の差が 0 より小さいといえる. すなわち, 女子の方が男子よりも軽いと統計的にいえる.

Q 12.2 新生児のデータフレーム `babies.frame` の頭囲 `head` に関して以下の設問に答えよ.

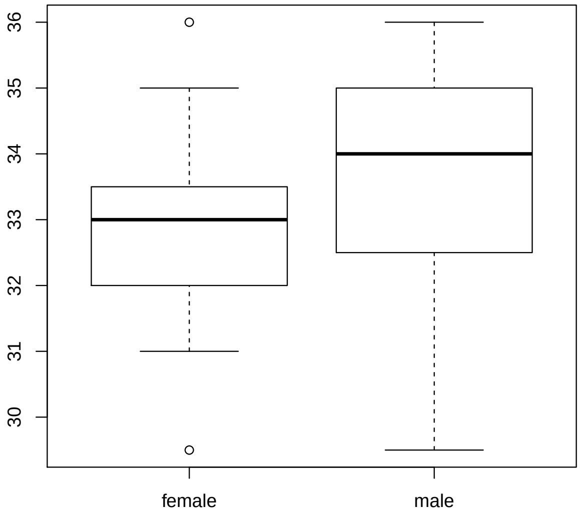
(1) 性別ごとのボックスプロットを描き, 比較せよ.

(2) 等分散であるかどうかを検定せよ.

(3) (2) の結果を考慮して母平均に差があるかどうかを検定せよ.

解答: (1) ボックスプロットを描くと,

```
> boxplot(head ~ gender, babies.frame)
```



となり, この結果から, 男子の頭囲よりも女子の頭囲の方が小さいことがわかる.
(2) 新生児に関する女子と男子の頭囲の母分散比 $\psi = \sigma_1^2/\sigma_2^2$ に関する検定

$$\begin{cases} H_0: & \psi = 1 \\ H_1: & \psi \neq 1 \end{cases}$$

を行う. 関数 `var.test` を利用した検定結果は以下のように与えられる:

```
> var.test(head ~ gender, babies.frame)
        F test to compare two variances

data:  head by gender
F = 0.78541, num df = 53, denom df = 45, p-value = 0.3965
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.4418655 1.3775573
sample estimates:
ratio of variances
         0.7854051
```

ここで, 有意水準は明示されていないがデフォルトで $\alpha = 0.05$ が設定されていることと, `data: head by gender` より, 新生児の頭囲を性別によって分類している. 次に, `alternative hypothesis: true ratio of variances is not equal to 1` より, 対立仮説が $\psi \neq 1$ の検定を実行していることがわかる. また, 自由度が, それぞれ, $n_1 - 1 = 53$ (`num df = 53`), $n_2 - 1 = 45$ (`denom df = 45`) で与えられることがわかり, さらに, `F = 0.78541` より, 検定統計量の実現値が

$$\widehat{\psi} = 0.78541$$

で与えられることがわかる.

棄却域は,

$$\begin{aligned} R &= (0, F_{n_2-1}^{n_1-1}(1-\alpha/2)] \text{ または } [F_{n_2-1}^{n_1-1}(\alpha/2), \infty) \\ &= (0, F_{45}^{53}(0.975)] \text{ または } [F_{45}^{53}(0.025), \infty) \\ &= (0, 0.570143] \text{ または } [1.777475, \infty) \end{aligned}$$

であるので,

$$\widehat{\psi} \notin R$$

となり, 帰無仮説 H_0 は棄却できない. よって, 有意水準 $\alpha = 0.05$ で新生児の頭囲は性別間の分散比が 1 でないといえない.

(3) (2) の結果から, 新生児の頭囲の分散は等分散とし, 以下の頭囲の平均値の差の仮説検定を行う:

$$\begin{cases} H_0: & \delta = 0 \\ H_1: & \delta \neq 0 \end{cases}$$

関数 `t.test` を用いて検定を以下のように行う:

```
> t.test(head ~ gender,babies.frame, var.equal = TRUE)
        Two Sample t-test

data:  head by gender
t = -3.1104, df = 98, p-value = 0.002447
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.3636817 -0.3013747
sample estimates:
mean in group female   mean in group male
            32.78704             33.61957
```

有意水準は明示されていないがデフォルトで $\alpha = 0.05$ が設定されている. `data: head by gender` より, (新生児の) 頭囲を性別によって分類している. また, `alternative hypothesis: true difference in means is not equal to 0` より, 対立仮説が $\delta \neq 0$ の検定を実行していることがわかる. また, `df = 98` より自由度が $n_1 + n_2 - 2 = 98$ で与えられることがわかり, さらに, `t = -3.1104` より, 検定統計量の実現値が

$$t = -3.1104$$

で与えられることがわかる.

棄却域は,

$$\begin{aligned} R &= (-\infty, -t_{n_1+n_2-2}(\alpha/2)] \text{ または } [t_{n_1+n_2-2}(\alpha/2), \infty) \\ &= (-\infty, -t_{98}(0.025)] \text{ または } [t_{98}(0.025), \infty) \\ &= (-\infty, -1.98447] \text{ または } [1.98447), \infty) \end{aligned}$$

であるので,

$$t \in R$$

となり, 帰無仮説 H_0 は棄却される. よって, 有意水準 $\alpha = 0.05$ で新生児の頭囲は性別間の母平均の差が 0 ではないといえる. すなわち, 統計的には女子と男子の頭囲の間に差があるといえる.

Q 12.3 2 組のデータ

$$\{x_1, \ldots, x_{n_1}\}, \quad \{y_1, \ldots, y_{n_2}\}$$

のそれぞれの平均値

$$\overline{x} := \sum_{i=1}^{n_1} x_i/n_1, \quad \overline{y} := \sum_{i=1}^{n_2} y_i/n_2$$

と不偏分散

$$u_1^2 := \sum_{i=1}^{n_1}(x_i - \overline{x})^2/(n_1 - 1), \quad u_2^2 := \sum_{i=1}^{n_2}(y_i - \overline{y})^2/(n_2 - 1)$$

が与えられているとする. 以下の設問に答えよ.

(1) 合併したデータ

$$\{z_1, \ldots, z_n\} := \{x_1, \ldots, x_{n_1}, y_1, \ldots, y_{n_2}\}$$

に関する平均値 $\overline{z}$, 分散 s^2, 不偏分散 u^2 を, 平均値 $\overline{x}$, $\overline{y}$ と不偏分散 u_1^2, u_2^2 を用いて表せ.

(2) データの個数 n_1, n_2, 平均値 $\overline{x}$, $\overline{y}$, 不偏分散 u_1^2, u_2^2 を引数として与えることによって, 母平均の差の検定を行う関数 `mu.test` を作成せよ.

(3) (1) の結果を利用してデータの個数 n_1, n_2, 平均値 $\overline{x}$, $\overline{y}$, 不偏分散 u_1^2, u_2^2 を引数として与えることによって, 合併データに関するデータの個数 n , 平均値 $\overline{z}$, 分散 s^2, 不偏分散 u^2, (分散, 不偏分散にもとづく) 標準偏差 s, u を計算するための R 関数 `restore` を作成せよ.

解答: (1) $n = n_1 + n_2$ に注意すると,

$$\begin{aligned} \overline{z} &= \frac{(n_1\overline{x} + n_2\overline{y})}{n} = \frac{(n_1\overline{x} + n_2\overline{y})}{n_1 + n_2}, \\ s^2 &= \frac{(n_1 - 1)u_1^2}{n_1 + n_2} + \frac{(n_2 - 1)u_2^2}{n_1 + n_2} + \frac{n_1 n_2}{(n_1 + n_2)^2}(\overline{x} - \overline{y})^2, \\ u^2 &= \frac{(n_1 - 1)u_1^2}{n_1 + n_2 - 1} + \frac{(n_2 - 1)u_2^2}{n_1 + n_2 - 1} + \frac{n_1 n_2}{(n_1 + n_2)(n_1 + n_2 - 1)}(\overline{x} - \overline{y})^2 \end{aligned}$$

で与えられる.

(2) 以下のような手順によって検定を行う関数を作成する:

(S1) 母分散比の検定 (当分散性の検定) を実施する. (予備検定と呼ばれる.)

(S2) 等分散である場合は, 平均値の差の検定として, 合併した不偏分散を用いたティー検定の統計量の値 (`test.stat`) と臨界値 (`critical.value`) を計算する.

(S3) 等分散でない場合は, 平均値の差の検定として, ウェルチの検定の統計量の値 (`test.stat`) と臨界値 (`critical.value`) を計算する.

なお, 関数には, 二つの標本の大きさ `n1`, `n2`, それぞれの平均値 `xb`, `yb`, そして不偏分散の値 `u12`, `u22` を必須引数として与え, 各検定を行う際に, 分散比の検定の場合は, 両側検定を行い, 平均値の差の検定を行う際には, 両側 (`"two.sided"`), 下側 (`"less"`), 上側 (`"greater"`) のそれぞれのタイプの仮説をオプショナル引数 `alternative` に与えることによって実施する. また, 母分散比の検定を行う有意水準 `alpha.var` と平均値の差の検定を行う有意水準 `alpha` は異なったものを与える仕様とし, デフォルト値では 0.05 とする.

これらの手順を実装した以下の関数を与える:

```
> mu.test <- function(n1, n2, xb, yb, u12, u22,
+                   alternative = c("two.sided", "less", "greater"),
+                   alpha.var = 0.05, alpha = 0.05)
+ {
+   # Step or Testing Variance Ratio
+   delta <- xb - yb
+   u2 <- (n1 - 1)*u12/(n1 + n2 - 1) + (n2 - 1)*u22/(n1 + n2 - 1)
+           + n1*n2*(xb - yb)^2/((n1 + n2)*(n1 + n2 - 1))
+   psi <- u12/u22
+   critical.value.var <- c(qf(alpha.var/2, n1 - 1, n2 - 1),
+                           qf(1 - alpha.var/2, n1 - 1, n2 - 1))
+   if(psi > critical.value.var[1] & psi < critical.value.var[2])
+     test.method <- "t.test"
+   else test.method <- "Welch.test"
+   # Step of Testing Difference in Mean
+   if(test.method == "t.test")
+   {
+     test.stat <- delta/sqrt(u2*(1/n1 + 1/n2))
+     if(alternative == "two.sided")
+       critical.value <- c(qt(alpha/2, n1 + n2 - 2), t(1 - alpha/2, n1 + n2 - 2))
+     else if(alternative == "less")
+       critical.value <- qt(alpha, n1 + n2 - 2)
+     else if(alternative == "greater")
+       critical.value <- qt(1 - alpha, n1 + n2 - 2)
+   }
+   else if(test.method == "Welch.test")
+   {
+     test.stat <- delta/sqrt(u12/n1 + u22/n2)
+     w1 <- (u12/n1)/(u12/n1 + u22/n2)
+     w2 <- 1 - w1
+     phi <- (w1^2/(n1 - 1) + w2^2/(n2 - 1))^(-1)
+     if(alternative == "two.sided")
+       critical.value <- c(qt(alpha/2, phi), qt(1 - alpha/2, phi))
+     else if(alternative == "less")
+       critical.value <- qt(alpha, phi)
+     else if(alternative ==" greater")
+       critical.value <- qt(1 - alpha, phi)
+   }
+   list(
+     delta = delta, u2 = u2,
+     alpha.var = alpha.var, psi = psi, critical.value.var = critical.value.var,
+     alpha = alpha, test.method = test.method, alternative = alternative,
+     phi = phi, test.stat = test.stat, critical.value = critical.value)
+ }
```

結果はリストで与えられる. 等分散性の検定が実施された結果として, `test.method` に合併した不偏分散を用いたティー検定 (`"t.test"`) またはウェルチの検定 (`"Welch.test"`) のどちらが実行されたかが与えられることに注意しよう.

(3) 以下のような関数によって与えられる:

```
> restore <- function(n1, n2, xb, yb, u12, u22)
+ {
+   n <- n1 + n2
+   m <- (n1*xb + n2*yb)/n
+   s2 <- ((n1 - 1)*u12 + (n2 - 1)*u22)/n + n1*n2*(xb - yb)^2/n^2
+   s <- sqrt(s2)
+   u2 <- n*s2/(n - 1)
+   u <- sqrt(u2)
+   list(n1 = n1, n2 = n2, xb = xb, yb = yb, u12 = u12, u22 = u22,
+        n = n, m = m, s2 = s2, s = s, u2 = u2, u = u)
+ }
```

Q 12.4 平成 12 年 (2000 年) の乳幼児身体発育調査 [1)] によると, 新生児 (単胎) の体重は女子 (x) と男子 (y) に関して, データの個数, 平均値, 不偏分散がそれぞれ以下のように与えられている:

$$n_1 = 1875, \overline{x} = 2980,\ u_1^2 = 380^2,$$
$$n_2 = 1981, \overline{y} = 3050,\ u_2^2 = 400^2$$

ただし, 単位は g である. 以下の設問に答えよ.

(1) Q 12.3 で作成した関数 `mu.test` を用いて, 新生児の体重に関して女子と男子との間に差があるかどうかの検定を行え.

(2) Q 12.3 で作成した関数 `restore` を用いて, 女子と男子を合併した「新生児」に関する平均体重, 不偏分散と標準偏差を求めよ.

解答: (1) 女子と男子の新生児の平均体重に差があるかどうかの仮説検定:

$$\begin{cases} H_0: & \delta = 0 \\ H_1: & \delta \neq 0 \end{cases}$$

を関数 `mu.test` を使って以下のように行う:

```
> mu.test(n1 = 1875, n2 = 1981, xb = 2980, yb = 3050, u12 = 380^2, u22 = 400^2,
+ alternative = "two.sided")
$delta
[1] -70
```

1)厚生労働省の Web ページ (`http://www.mhlw.go.jp/`) を参照してほしい.

```
$u2
[1] 152375

$alpha.var
[1] 0.05

$psi
[1] 0.9025

$critical.value.var
[1] 0.9144678 1.0934105

$alpha
[1] 0.05

$test.method
[1] "Welch.test"

$alternative
[1] "two.sided"

$phi
[1] 3853.947

$test.stat
[1] -5.572771

$critical.value
[1] -1.96058  1.96058
```

この結果から, 母分散比の検定を行うための検定統計量の実現値が

$$\widehat{\psi} = 0.9025$$

で与えられ, 棄却域が

$$R = (0, 0.9144678] \text{ または } [1.0934105, \infty)$$

となることから,

$$\widehat{\psi} \in R$$

となり, 帰無仮説は棄却され, 等分散でないことがわかる. この結果, `$test.method` に `"Welch.test"` が与えられ, ウェルチの検定が行われている. 検定の結果としては, 自由度 (`psi`) が $\phi = 3853.947$ であたえられ, 検定統計量の実現値 (`$test.stat`) が,

$$t = -5.572771$$

であり, 臨界値 (`$critical.value`) より棄却域が,

$$R = (-\infty, -1.96058] \text{ または } [1.96058, \infty)$$

となることから,

$$t \in R$$

となり, 有意水準 $\alpha = 0.05$ で帰無仮説は棄却される. よって, 統計的には新生児の平均体重は性別間に差があるといえる.

(2) (Q12.3) (3) で定義された関数 restore を以下のように実行することによって設問に対する結果を得る:

```
> restore(1875, 1981, 2980, 3050, 380^2, 400^2)
$n1
[1] 1875

$n2
[1] 1981

$xb
[1] 2980

$yb
[1] 3050

$u12
[1] 144400

$u22
[1] 160000

$n
[1] 3856

$m
[1] 3015.962

$s2
[1] 153559.6

$s
[1] 391.8668

$u2
[1] 153599.4

$u
[1] 391.9176
```

この結果から, 合併したデータ

$$\{z_1, \ldots, z_n\} := \{x_1, \ldots, x_{n_1}, y_1, \ldots, y_{n_2}\}$$

に関する平均値 (`m`) が

$$\overline{z} = 3015.962$$

で与えられ, 不偏分散 (`u2`) が

$$u^2 = 153599.4$$

となることがわかる. さらに, 不偏分散にもとづく標準偏差 (`u`) が

$$u = 391.9176$$

となることもわかる.

第13章　演習問題解答 (13)

Q 13.1 以下の設問に答えよ.

(1) 3.3 節で読み込んだデータフレーム babies.frame に対して, 以下のように入力することによって weight を列ベクトル $\boldsymbol{y}$ として定義し, 1 列目に 1 のみの成分からなるベクトルと 2 列目に height を成分として持つ行列 $\mathbf{X}$ を定義せよ:

```
> y <- matrix(babies.frame$weight)
> X <- cbind(rep(1, 100), babies.frame$height)
```

(2) 最小自乗推定値のベクトル $\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{y}$ を計算し, 関数 lm で求めたものと比較せよ.

(3) 射影行列 $\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ を計算し, 回帰ベクトル $\widehat{\boldsymbol{y}} = \mathbf{P}\boldsymbol{y}$ と残差ベクトル $\boldsymbol{e} = \boldsymbol{y} - \widehat{\boldsymbol{y}}$ を求め, 関数 lm で求めたものと比較せよ.

(4) 誤差分散の推定値 $\widehat{\sigma}^2 = \boldsymbol{e}'\boldsymbol{e}/(n-2)$ を計算し, 関数 lm で求めたものと比較せよ.

解答: (1) 題意に与えられているスクリプトを以下のように実行すればよい:

```
> y <- matrix(babies.frame$weight)
> X <- cbind(rep(1, 100), babies.frame$height)
```

(2) 以下のスクリプトを実行することによって比較できる:

```
> solve(t(X) %*% X) %*% t(X) %*% y
           [,1]
[1,] -4534.4097
[2,]   155.1948
> babies.lm <- lm(weight ~ height, babies.frame)
> coef(babies.lm)
(Intercept)      height
 -4534.4097    155.1948
```

すなわち, ベクトルと行列の演算によって得られた最小自乗推定値のベクトルは, 関数 lm によって得られた回帰係数の結果と一致する. ただし, 前者は行列オブジェクトであるのに対して, 後者はベクトルオブジェクトであることに注意しよう.

(3) 以下のスクリプトを実行することによって射影行列, 回帰ベクトル, 残差ベクトルが計算できる:

```
> P <- X %*% solve(t(X) %*% X) %*% t(X)
> yhat <- P %*% y
> e <- y - yhat
```

この結果から, 以下のような入力によって, 回帰ベクトルと残差ベクトルを比較することができる:

```
> as.vector(yhat)
  [1] 3147.735 2449.358 2992.540 3070.138 3225.332 2604.553 3225.332
  [8] 3225.332 2759.748 3147.735 3458.125 3380.527 3225.332 3225.332
 [15] 2604.553 2914.943 2914.943 3380.527 2914.943 3070.138 3147.735
 [22] 2914.943 2992.540 3225.332 3380.527 3147.735 2992.540 3380.527
 [29] 2682.151 2759.748 2914.943 3070.138 3225.332 3070.138 2759.748
 [36] 3302.930 2914.943 3225.332 3302.930 3225.332 3380.527 3302.930
 [43] 3302.930 2837.345 3535.722 3380.527 3147.735 3225.332 3380.527
 [50] 2992.540 3070.138 3380.527 2914.943 2992.540 3380.527 3380.527
 [57] 2914.943 3380.527 3380.527 3535.722 3458.125 3070.138 2449.358
 [64] 3225.332 2526.956 3225.332 3070.138 3070.138 3225.332 2914.943
 [71] 2992.540 3225.332 3302.930 3070.138 2759.748 3070.138 3147.735
 [78] 3225.332 3535.722 3147.735 3225.332 3225.332 3302.930 3147.735
 [85] 3070.138 3535.722 3225.332 3225.332 3225.332 3070.138 3070.138
 [92] 3225.332 2837.345 3535.722 2449.358 3225.332 3380.527 2604.553
 [99] 3225.332 3070.138
> fitted(babies.lm)
       1        2        3        4        5        6        7
3147.735 2449.358 2992.540 3070.138 3225.332 2604.553 3225.332
       8        9       10       11       12       13       14
3225.332 2759.748 3147.735 3458.125 3380.527 3225.332 3225.332
      15       16       17       18       19       20       21
2604.553 2914.943 2914.943 3380.527 2914.943 3070.138 3147.735
      22       23       24       25       26       27       28
2914.943 2992.540 3225.332 3380.527 3147.735 2992.540 3380.527
      29       30       31       32       33       34       35
2682.151 2759.748 2914.943 3070.138 3225.332 3070.138 2759.748
      36       37       38       39       40       41       42
3302.930 2914.943 3225.332 3302.930 3225.332 3380.527 3302.930
      43       44       45       46       47       48       49
3302.930 2837.345 3535.722 3380.527 3147.735 3225.332 3380.527
      50       51       52       53       54       55       56
2992.540 3070.138 3380.527 2914.943 2992.540 3380.527 3380.527
      57       58       59       60       61       62       63
2914.943 3380.527 3380.527 3535.722 3458.125 3070.138 2449.358
      64       65       66       67       68       69       70
3225.332 2526.956 3225.332 3070.138 3070.138 3225.332 2914.943
      71       72       73       74       75       76       77
2992.540 3225.332 3302.930 3070.138 2759.748 3070.138 3147.735
```

```
      78       79       80       81       82       83       84 
3225.332 3535.722 3147.735 3225.332 3225.332 3302.930 3147.735 
      85       86       87       88       89       90       91 
3070.138 3535.722 3225.332 3225.332 3225.332 3070.138 3070.138 
      92       93       94       95       96       97       98 
3225.332 2837.345 3535.722 2449.358 3225.332 3380.527 2604.553 
      99      100 
3225.332 3070.138 

> as.vector(e)

  [1]   22.2649279  160.6417296   27.4597727  -50.1376497  104.6675055
  [6] -424.5531152  -85.3324945  194.6675055 -179.7479600  212.2649279
 [11] -158.1247617  169.4726607 -175.3324945  174.6675055   95.4468848
 [16] -294.9428049  225.0571951  -40.5273393  135.0571951 -210.1376497
 [21] -167.7350721  555.0571951 -422.5402273 -225.3324945 -180.5273393
 [26]   12.2649279  -32.5402273  149.4726607  227.8494624  300.2520400
 [31]  -64.9428049 -430.1376497  -45.3324945 -190.1376497 -359.7479600
 [36]  127.0700831 -374.9428049 -115.3324945  297.0700831 -175.3324945
 [41] -310.5273393   77.0700831  -22.9299169   22.6546176   54.2778159
 [46] -240.5273393   12.2649279  494.6675055 -340.5273393 -382.5402273
 [51] -130.1376497 -190.5273393   65.0571951   37.4597727 -200.5273393
 [56]  439.4726607  205.0571951   99.4726607   59.4726607  164.2778159
 [61]  201.8752383  289.8623503    0.6417296  174.6675055   93.0443072
 [66] -405.3324945   69.8623503  -30.1376497 -275.3324945 -154.9428049
 [71] -322.5402273   24.6675055 -262.9299169   49.8623503 -139.7479600
 [76]   69.8623503  442.2649279 -195.3324945  324.2778159 -127.7350721
 [81] -545.3324945  334.6675055 -362.9299169 -137.7350721 -280.1376497
 [86]   84.2778159  194.6675055  664.6675055  -25.3324945  259.8623503
 [91]  -70.1376497 -135.3324945  -17.3453824  114.2778159  250.6417296
 [96]  164.6675055  339.4726607  395.4468848 -135.3324945  379.8623503

> resid(babies.lm)

          1           2           3           4           5 
 22.2649279 160.6417296  27.4597727 -50.1376497 104.6675055 
          6           7           8           9          10 
-424.5531152 -85.3324945 194.6675055 -179.7479600 212.2649279 
         11          12          13          14          15 
-158.1247617 169.4726607 -175.3324945 174.6675055  95.4468848 
         16          17          18          19          20 
-294.9428049 225.0571951 -40.5273393 135.0571951 -210.1376497 
         21          22          23          24          25 
-167.7350721 555.0571951 -422.5402273 -225.3324945 -180.5273393 
         26          27          28          29          30 
 12.2649279 -32.5402273 149.4726607 227.8494624 300.2520400 
         31          32          33          34          35 
 -64.9428049 -430.1376497 -45.3324945 -190.1376497 -359.7479600 
         36          37          38          39          40 
 127.0700831 -374.9428049 -115.3324945 297.0700831 -175.3324945 
         41          42          43          44          45 
-310.5273393  77.0700831 -22.9299169  22.6546176  54.2778159 
         46          47          48          49          50 
-240.5273393  12.2649279 494.6675055 -340.5273393 -382.5402273 
         51          52          53          54          55 
```

```
   -130.1376497 -190.5273393   65.0571951   37.4597727 -200.5273393
             56           57           58           59           60
    439.4726607  205.0571951   99.4726607   59.4726607  164.2778159
             61           62           63           64           65
    201.8752383  289.8623503    0.6417296  174.6675055   93.0443072
             66           67           68           69           70
   -405.3324945   69.8623503  -30.1376497 -275.3324945 -154.9428049
             71           72           73           74           75
   -322.5402273   24.6675055 -262.9299169   49.8623503 -139.7479600
             76           77           78           79           80
     69.8623503  442.2649279 -195.3324945  324.2778159 -127.7350721
             81           82           83           84           85
   -545.3324945  334.6675055 -362.9299169 -137.7350721 -280.1376497
             86           87           88           89           90
     84.2778159  194.6675055  664.6675055  -25.3324945  259.8623503
             91           92           93           94           95
    -70.1376497 -135.3324945  -17.3453824  114.2778159  250.6417296
             96           97           98           99          100
    164.6675055  339.4726607  395.4468848 -135.3324945  379.8623503
```

以上の結果から, 当てはめ値と残差のベクトルがどちらの方法で求めた結果も数値的には一致することがわかる.

(4) 以下のようにスクリプトを入力することによって比較できる:

```
> t(e) %*% e/(100 - 2)
         [,1]
[1,] 59885.19
> summary(babies.lm)$sigma^2
[1] 59885.19
```

これらの結果は数値的には一致している. ただし, 前者が 1 行 1 列の行列オブジェクトであるのに対して, 後者は長さ 1 のベクトルオブジェクトであることに注意しよう.

Q 13.2 データフレーム `babies.frame` に対して以下の設問に答えよ.

(1) 体重 (`weight`) を 身長 (`height`), 胸囲 (`chest`), 頭囲 (`head`) で重回帰分析せよ. また, その結果から回帰係数の有意性や決定係数などをみることによって回帰の当てはまりの良さなどを考察せよ.

(2) 残差のプロットを行うことによって回帰診断を行い, 結果を考察せよ.

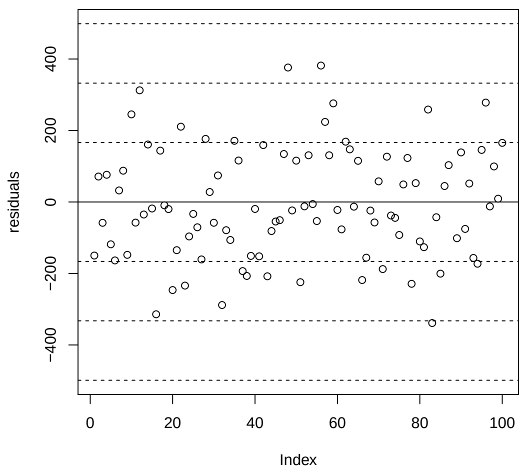
解答: (1) 以下のスクリプトを実行することによって重回帰モデルを当てはめた結果をオブジェクト `babies.lm.whch` に付値し, それを要約することによって重回帰分析を実行する:

```
> babies.lm.whch <- lm(weight ~ height + chest + head, babies.frame)
> summary(babies.lm.whch)
Call:
lm(formula = weight ~ height + chest + head, data = babies.frame)

Residuals:
    Min      1Q  Median      3Q     Max
-338.67 -112.37  -21.09  117.80  561.76

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5531.86     545.50 -10.141  < 2e-16 ***
height         79.00      12.52   6.309 8.61e-09 ***
chest         115.94      12.80   9.055 1.59e-14 ***
head           31.09      14.25   2.182   0.0316 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 166.3 on 96 degrees of freedom
Multiple R-squared:  0.7822,        Adjusted R-squared:  0.7754
F-statistic: 114.9 on 3 and 96 DF,  p-value: < 2.2e-16
```

この結果から, 全ての回帰係数は有意であり, 決定係数と自由度調節済み決定係数も 8 割近い値を示している. なお, 第 13 章の重回帰分析の結果と比較して, 頭囲 (`head`) がモデルに加わることによってどのような変化があったかを各自確かめてほしい.

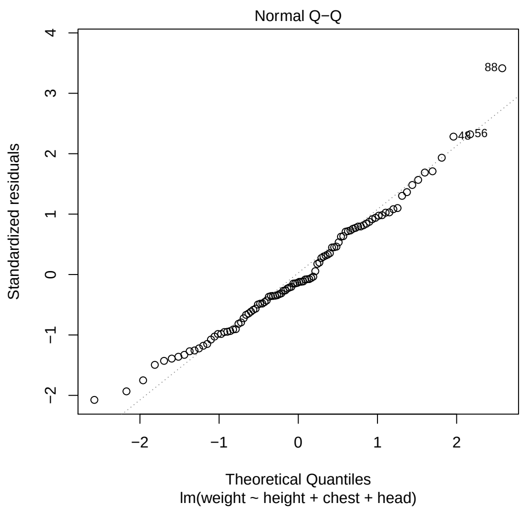
(2) まず, 以下のように入力することによって残差のインデックスプロットを描く:

```
> plot.resid.index(babies.lm.whch)
```



この結果から, 外れ値や周期性などの傾向は見られない.
次に, 以下のように入力することによって残差の正規 Q-Q プロットを描く:

```
> plot(babies.lm.whch, which = 2)
```



この結果から, 裾においては数個の残差が外れるものが存在するけれども全体としては正規性は疑われないであろう.

Q 13.3 演習問題 Q 3.1 で読み込んだデータフレーム `firms.frame` に対して以下の設問に答えよ.

(1) 売上高 (`sales`) を 従業員数 (`employees`) と 資産合計 (`assets`) で重回帰分析せよ. また, その結果から回帰係数の有意性や決定係数などをみることによって回帰の当てはまりなどの良さを考察せよ.

(2) 残差のプロットを行うことによって回帰診断を行い, 結果を考察せよ.

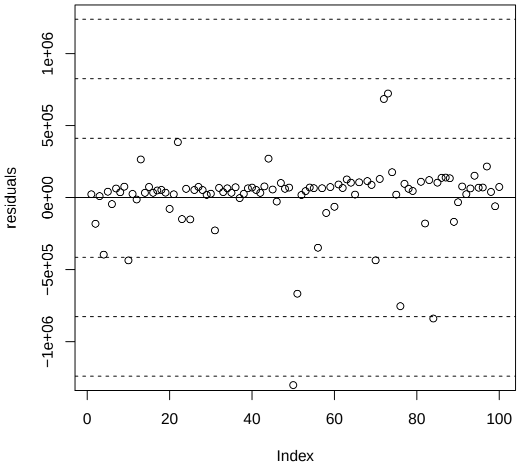
解答: (1) 以下のスクリプトを実行することによって重回帰モデルを当てはめた結果をオブジェクト `firms.lm` に付値し, それを要約することによって重回帰分析を実行する:

```
> firms.lm <- lm(sales ~ employees + assets, firms.frame)
> summary(firms.lm)
Call:
lm(formula = sales ~ employees + assets, data = firms.frame)

Residuals:
     Min       1Q   Median       3Q      Max
-2574848    16910    53991    78107  1972234

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.948e+04  4.530e+04  -1.534 0.128327
employees    8.499e+00  2.441e+00   3.482 0.000747 ***
assets       9.096e-01  4.429e-02  20.541  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 413000 on 97 degrees of freedom
Multiple R-squared:  0.8813,        Adjusted R-squared:  0.8789
F-statistic: 360.2 on 2 and 97 DF,  p-value: < 2.2e-16
```

ここで, 全ての回帰係数は有意であり, 決定係数と自由度調節済み決定係数も 9 割近い値を示している.

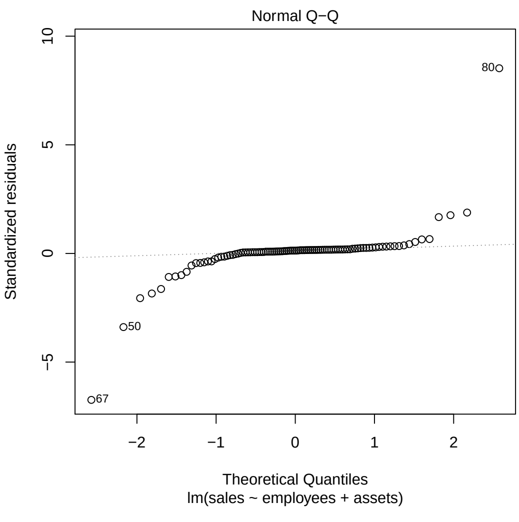
(2) まず, 以下のように入力することによって残差のインデックスプロットを描く:

```
> plot.resid.index(firms.lm)
```



この結果から, 外れ値が存在することがわかるが, 周期性などの傾向は見られない.
次に, 以下のように入力することによって残差の正規 Q-Q プロットを描く:

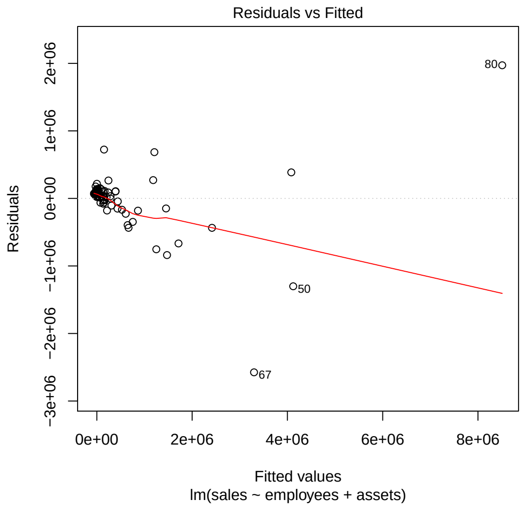
```
> plot(firms.lm, which=2)
```



この可視化の結果から, 残差は正規分布に従っているとは言いがたく, 誤差の正規性に関する仮定は疑われる.

さらに, 本書では与えなかったが, 回帰診断のプロットとして, 当てはめ値 (横軸) と残差 (縦軸) の散布図を描いたものを与える:

```
> plot(firms.lm, which = 1)
```



この結果から, 当てはめ値が大きくなるにつれて, 残差の分散が拡大する傾向がみえる [1]. このような結果からは分散が個体ごとに不均一 (分散不均一) であることが疑われる.

以上の結果を総合すると, 重回帰モデルを当てはめた数値的な結果からは, 当てはまりは良いと考えられるかもしれないけれども, 残差に関する診断の結果からは誤差の正規性や均一性の仮定が疑われ, それによって推定・検定の信頼性に問題があることが推測される.

この結果が示すように, 誤差の正規性が疑われる問題に対して, モデルを改良するいくつかの方法が考えられるが, 一つの解決法は各変量の対数をとったものに線形回帰モデルを当てはめることである. これ以上の考察は, 本書の範囲を超えるため, 割愛するけれども, 以下のスクリプトを実行し, 結果を各自で考察してほしい:

```
firms.lm.log <- lm(log(sales) ~ log(employees) + log(assets), firms.frame)
summary(firms.lm.log)
plot.resid.index(firms.lm.log)
plot(firms.lm.log, which = 2)
plot(firms.lm.log, which = 1)
```

Q 13.4 演習問題 Q 3.3 で各自が読み込んだデータフレームを用いて回帰分析を行え.

(読者が各自試して頂ければ幸いである.)

[1] このような形状のプロットを, ファン形状 (fan shape), 漏斗形状 (funnel shape) と呼ぶことがある.