R入門

行動計量学会第15回春のセミナー合宿

横山 暁（帝京大学）

第1部 13:30～15:00

R入門（その1）

基本演算，ベクトル・行列等の操作，データの入出力，データの要約等

高級電卓としてのR

- R Consoleウィンド上に数式を入力することで，高級電卓として利用することができる
- 主な演算子や関数
  - 四則演算の演算子はそれぞれ +, -, *, /
  - 累乗は^もしくは**，円周率はpi，ネイピア数eはexp(1)，対数はlog(真数, 底)
    - 常用対数はlog10(真数)，底が2の対数はlog2(真数)でも表すことができる
  - 三角関数は正弦，余弦，正接それぞれsin, cos, tan
  - 論理和(OR)，論理積(AND)はそれぞれ|, &
    - ベクトルの場合は||, &&
  - 四捨五入はround(数, 桁数)
  - 小数を切り捨てたければfloor()，切り上げたければceiling()
  - 値の比較として>, >=, <, <=, ==, !=
    - それぞれ大なり，大なりイコール，小なり，小なりイコール，等しい，異なる

変数への代入

- 単純な計算だけであれば，R Consoleウィンド上で入力を繰り返せばよい
- 結果を保存したい（後で利用したい）場合には適当な名前の変数に結果を代入するとよい
  - 代入方法は，a<-5（5->aやa=5でもよい）とする
    - 円周率piに別の数字を代入してしまうとpiという変数が上書きされてしまうので注意

データの型

- データの型には以下の5つがある
  - numeric（実数）型
  - complex（複素数）型
  - character（文字列）型
  - logical（論理値）型
  - NULL型
- データの型を調べる
  - mode()関数を用いることでデータの型を調べることができる
  - is.numeric(), is.complex(), is.character(), is.logical(), is.null()という関数でデータの型を個別に調べることもできる
- データの型を変更する
  - as.numeric(), as.complex(), as.charactor(), as.logical(), as.null()という関数でデータの型を変更することができる
    - ただしcharacter型で文字列が数値でないものを数値化しようとすると警告が出てNAとなる

ベクトル

- 最も基本となるデータ構造
  - スカラーも長さ1のベクトル
  - 長さ0のベクトルも存在する
    - numeric(0), character(0)で表される
- ベクトルの入力
  - 開始値:終了値で開始値から終了値まで数が1ずつ増えるベクトルとなる
    - （例）vector.a <- 1:4
  - c(ベクトルの要素のカンマ区切り)
    - （例）vector.b <- c(3, 5, 7, 9)
    - （例）vector.c <- c(1, 2, 3)
- ベクトルの演算
  - ベクトルの積は要素ごとの積
    - 要素が足らない場合には繰り返して使われる
  - 内積は %*%を用いる

ベクトル

- ベクトルの一部を参照する
  - ベクトル[参照したい要素番号]
- ベクトルの一部を変更する
  - ベクトル[参照したい要素番号] <- 代入したい値
- 参照の方法
  - vec[i]・・・i番目を参照
  - vec[i:j]・・・i番目からj番目までを参照
  - vec[c(i, j, k)]・・・i, j, k番目を参照
  - マイナスを使う事で要素を除外して参照することもできる
    - vec[-i]など

行列

- matrix()関数
  - 行列を定義する
  - matirx(data, nrow=行数, ncol=列数, byrow=TRUE)
    - byrow=TRUE (もしくはTや1でもOK）とすると行方向から，FALSE（もしくはFや0）とするか省略と列方向からデータが挿入される
    - データの要素数が行数×列数に一致しないと警告メッセージが出るが行列は作られる
      - 多い場合には切り捨てられ，足らない場合には繰り返し利用される
- 行列の要素の参照や変更
  - 基本的にベクトルの要素の参照の応用
  - 行列[参照したい行番号, 参照したい列番号]
    - 行のみなら行列[行番号,]
    - 列のみなら行列[, 列番号]
- array()関数
  - 行列を拡張した配列
  - array(data, dim=配列の大きさ)

ベクトルや行列の集計に関する主な関数

- ベクトルに用いる関数
  - length(vec)：ベクトルの長さ
  - sum(vec)：合計
  - mean(vec)：平均
  - median(vec)：中央値
  - quantile(vec)：四分位
  - var(vec)：不偏分散
  - max(vec)：最大値
  - min(vec)：最小値
  - range(vec)：範囲
  - prod(vec)：積
- 行列に用いる関数
  - t(mat)：転置
  - nrow(mat)：行数
  - ncol(mat)：列数
  - dim(mat)：行列の大きさ
  - rowSums(mat)：行和
    - apply(mat, 1, sum)
  - colSums(mat)：列和
    - apply(mat, 2, sum)
  - rowMeans(mat)：行平均
    - apply(mat, 1, mean)
  - colMeans(mat)：列平均
    - apply(mat, 2, mean)

ベクトル，行列の結合

- rbind()関数
  - 行方向に結合
    - rbind(ベクトル1, ベクトル2)
    - rbind(行列1, 行列2)
- cbind()関数
  - 列方向に結合
    - cbind(ベクトル1, ベクトル2)
    - cbind(行列1, 行列2)

  - 長さが違うベクトルを結合しようとすると，長さが短い方のベクトルの要素が繰り返して使われる
  - 結合しようとする方向の行数・列数が異なる行列を結合しようとするとエラーとなる

行列のnames属性

- names属性
  - 行や列に名前（ラベル）を付ける
    - x<-matrix(1:6, nrow=2, ncol=3)
    - rownames(x)<-c("ue", "shita")
    - colnames(x)<-c("hidari","mannaka","migi")
  - 名前を初期化する
    - rownames(x)<-NULL
    - colnames(x)<-NULL
  - 行列だけでなくベクトルにも要素にラベルを付けることができる

データフレーム

- データフレーム (data.frame) は複数のデータの型の混在を許す行列形式のデータ
  - データ分析で用いる一般的なデータ構造
  - 同じ長さの複数のベクトルから構成される

  - （例）学生の氏名，テストの得点，成績のデータフレーム

```
seiseki <- data.frame(
        point = c(50, 90, 70, 80),
        hyouka = c("D", "S", "B", "A"),
        row.names = c("A君", "Bさん", "Cさん", "D君")
)
```

リスト

- リスト（list）はあらゆる型のあらゆる構造のデータを要素に持つことができる
  - リスト自身も要素にできる
  - 今まで入力したデータを含めリストにしてみる
  - mylist <- list(1:5, vector.a, matrix(1:9, nrow=3, ncol=3, byrow=T), seiseki)

データの入出力

- コマンドラインからのデータの読み込み
  - スペース区切り，Tab区切り，カンマ区切りのファイルを読み込む
  - read.table()関数もしくはread.csv()関数を用いる
    - scan()関数を用いる方法もある
  - data <- read.table("読み込むファイル", header=F, sep="区切り文字", row.names=1)
    - 読み込むファイルは“C:/hoge/data.txt”の要領でファイルがある場所を指定する
      - 作業フォルダはgetwd()で確認でき，setwd()で変更できる
      - 作業フォルダにファイルがある場合には相対パスでよい
    - headerは1行目がヘッダー行であれはTにする
    - 区切り文字は省略するとスペースもしくはタブとして読み込まれる
    - row.namesは行のラベルがある列番号を指定する（重複したラベルは許されない）
  - CSVファイルを読み込む場合にはread.csv()を用いる
    - 引数の指定の方法はほぼ同じ（header=T, sep=“,”が標準になっている等の違いがある）

データの入出力

- コマンドラインからのデータの書き出し
  - データをファイルに保存する
  - スペース区切り，Tab区切り，カンマ区切り等で出力する
  - write.table()関数もしくはwrite.csv()関数を用いる
  - write.table(data, "保存するファイル名", quote=F, append=T, sep=" ", row.names=T, col.names=T)
    - 標準ではスペース区切りで出力される
    - quoteはFにしておく
    - appendは上書きを許さない場合T（追記される），許す場合F
    - 行名（row.names）と列名（col.names)は標準ではTとなっている
  - CSVファイルで描きだしたい場合にはwrite.csv()関数を使うと便利
    - sep=“,”, dec=“.”が標準になっている等の違いがある

計算結果等の書き出し

- 計算結果等を書き出したい場合はsink()関数を用いると便利

  sink("ファイル名")

  書き出したい計算等

  sink()

  - 必ずsink()で終了しなければならない

スクリプトの作成

- スクリプトを作成する
  - 何度も同じことを実行するときに，いちいちコマンドラインに入力するのではなく，事前に別の場所に入力しておいて実行したほうが便利
  - 「ファイル」から「新しいスクリプト」で開く画面（Rエディタ）に入力をしておく
  - 実行するときには「編集」から「カーソル行または選択中のRコードを実行」もしくは「すべて実行」で実行できる
  - スクリプトを保存しておき，後で「ファイル」から「スクリプトを開く」で開くことができる
  - スクリプトの画面（Rエディタ）には1行ずつコマンドを記載しても良いが，一般的には複数行からなるプログラムを記載する

Rの終了と作業スペース・履歴の扱い

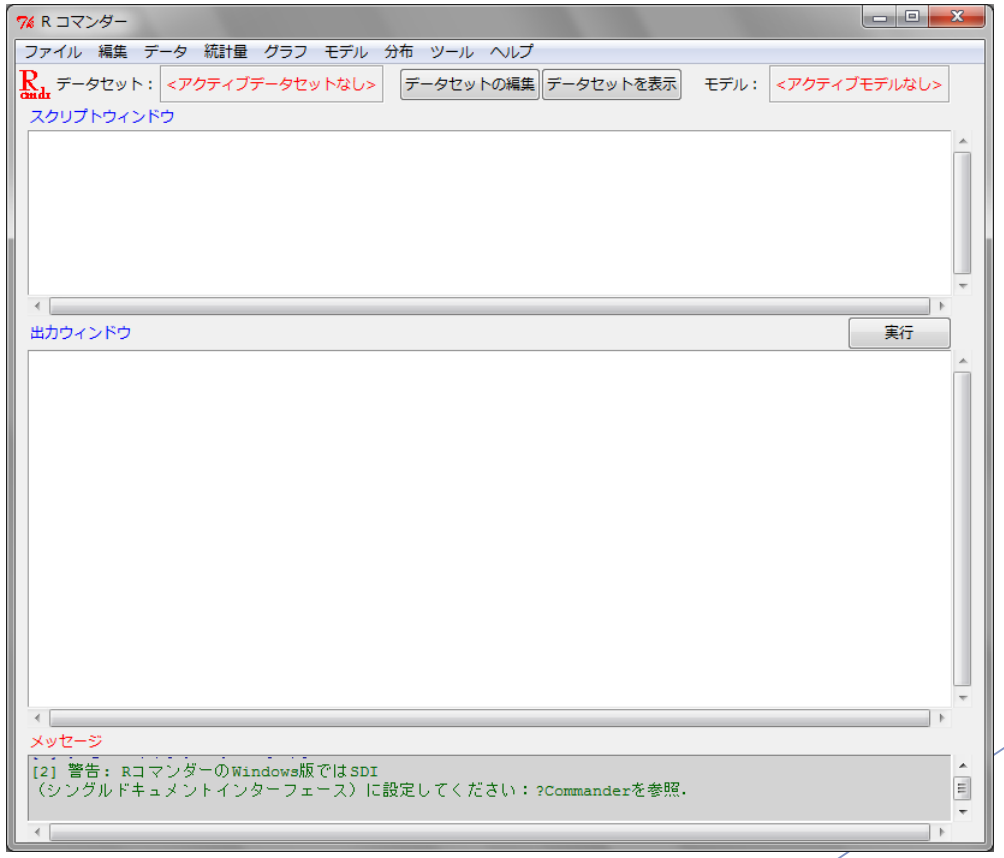
- Rを終了しようとすると「作業スペースを保存しますか」と聞かれる
  - 「はい」とすると，作業フォルダに「.Rdata」や「.Rhistory」というファイルが作成され，次回起動時に自動的に読み込まれる
    - .Rdataには計算に用いたデータ等が，.Rhistoryには入力したコマンド履歴が格納されている
  - 作業フォルダから.Rdataと.Rhistoryを直接削除すると，次回起動時に前回までの結果は読み込まれない（失われる）

オブジェクトの削除

- 変数や関数（以下まとめてオブジェクト）を確認する方法
  - 「その他」→「オブジェクトの一覧」
  - ls()
- オブジェクトを削除する方法
  - 個別に削除するにはrm(オブジェクト名)
  - 全て削除するには「その他」→「すべてのオブジェクトの削除」
    - rm(list=ls(all=TRUE)) でもよい
  - （前回終了時に作業スペースを保存し）起動時に自動的に読み込まれたオブジェクトを上記の方法で削除した場合，Rを終了する際に作業スペースを保存しないと，次回起動した際に削除したはずのオブジェクトが復活するので注意

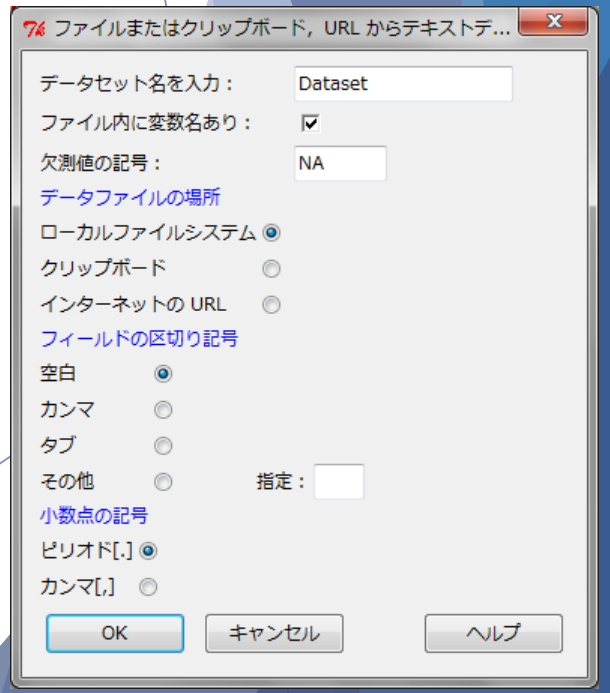
R Commanderの起動

- ここからはR Commanderを利用する
- 「パッケージ」→「パッケージの読み込み…」から「Rcmdr」を選択する



R Commanderを用いたデータの入出力

- データの手入力
  - メニューの「データ」→「新しいデータセット」
  - データセット名（データの名前）を入力
  - データエディタ ウィンドにデータを入力する
    - 「var1」等の変数名の部分をクリックすると変数名や変数型を変更できる
- 外部データの読み込み
  - 「データ」→「データのインポート」
    - 幾つかの形式からデータをインポートできる
    - CSVファイル等は「テキストファイルまたは（略）」からインポートする
      - データセット名の設定や区切り記号の選択が可能

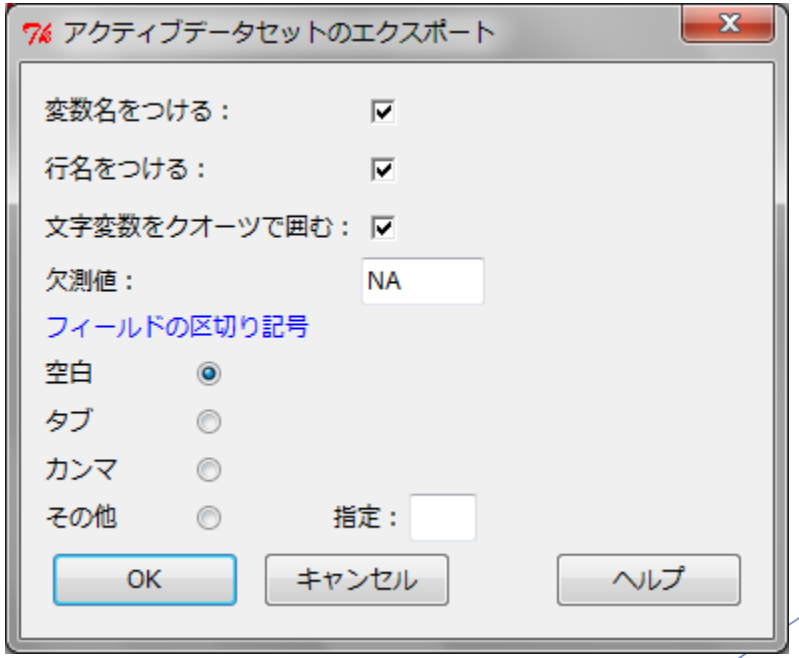


R Commanderを用いたデータの入出力

- アクティブデータセットを編集する
  - R Commander画面上の「データセットの編集」
  - 「データ」→「アクティブデータセット」
  - 「データ」→「アクティブデータセット内の変数の管理」
  - 等でデータや変数を編集できる
    - 例えば・・・
      - 変数の標準化
      - 数値変数を因子に変換（水準名を指定する等ができる）
      - 変数の再コード化（コードの表現を変更したり，新たなグルーピングをすることができる）
      - などなど
      - ※各変換等によって新たに作成される変数は元のデータセットに追加される

R Commanderを用いたデータの入出力

- データの保存，書き出し
  - 「データ」→「アクティブデータセット」→「アクティブデータセットの保存」でRdataの形式でデータを保存できる
  - 「データ」→「アクティブデータセット」→「アクティブデータセットのエクスポート」でテキスト形式（CSV等）でデータを描きだすことができる



準備

- データを読み込む
  - datasetsパッケージに含まれるデータを使用する
    - Titanic
      - Survival of passengers on the Titanic
        - 4次元配列 2201人，4変数を集計したもの
        - Class: 1st, 2nd, 3rd, Crew
        - Sex: Male, Female
        - Age: Child, Adult
        - Survived: No, Yes

準備

- データを読み込む
  - datasetsパッケージに含まれるデータを使用する
    - attitude
      - The Chatterjee–Price Attitude Data
        - 金融団体の30の部署における35人ほどの事務職員を対象として，好ましい回答をしたものの割合
        - rating: 総合評価
        - complaints: 従業員の不満の処理
        - privileges: 特権を認めない
        - learning: 学ぶ機会
        - raises: 成果による昇給
        - critical: 批判的すぎる
        - advace: 昇進

準備

- データを読み込む
  - datasetsパッケージに含まれるデータを使用する
    - mtcars
      - Motor Trend Car Road Tests
        - 1974年の「Motor Trend」誌から抽出した32車種の，燃費の他，デザインなどの評価データ
        - データフレーム: 32×11
        - mpg: 1ガロンあたり走行距離(miles/gallon)
        - cyl: エンジンのシリンダー数
        - disp: 排気量
        - hp: 総馬力
        - drat: Rear axle ratio
        - wt: 重量(lb/1000)
        - qsec: 1/4マイル時間
        - vs: V/S
        - am: トランスミッション(0:オートマティック，1:マニュアル)
        - gear: 前進ギヤ数
        - carb: キャブレター数

質的データの要約（Titanicを使用）

- 1変数のデータの要約
  - 単純集計を行う
    - 「統計量」→「要約」→「頻度分布」
      - 各水準の単純集計と相対度数が得られる
- 2変数データの要約
  - クロス集計表（分割表）を求める
    - 「統計量」→「分割表」→「2元表」
      - 行の変数と列の変数を1つずつ選択
      - クロス集計表およびオプションの設定によっては検定統計量が得られる
- 多変数データの要約
  - 多元表を求める
    - 「統計量」→「分割表」→「多元表」
      - 行の変数，列の変数，コントロール変数をそれぞれ1つずつ選択
      - コントロール変数の因子ごとに分割表が得られる

質的データの視覚化（Titanicを使用）

- 棒グラフ
  - 「グラフ」→「棒グラフ」
- 円グラフ
  - 「グラフ」→「円グラフ」
- 非常に簡単にグラフを描くことができる
- 出力されたグラフ上で右クリックすることでグラフをファイルに保存したり印刷することができる

量的データの要約（attitudeを使用）

- 1変数データの要約
  - 基本統計量を求める
    - 「統計量」→「要約」→「アクティブデータセット」
      - 各変数の最小値，第1～第3四分位，平均値，最大値が求められる
    - 「統計量」→「要約」→ 「数値による要約」
      - 基本統計量を求めたい変数を（1つ以上）選択
      - 平均，（不偏）分散，四分位等が得られる
      - 層別に使える変数があれば，「層別して要約」をすることで，ある質的変数ごとに層別して基本統計量を求めることも可能である

量的データの要約（attitudeを使用）

- 2変数データの要約
  - 相関係数を求める
    - 「統計量」→「要約」→「相関行列」
    - 相関行列を求めたい変数を選択
    - 関数のタイプは「ピアソンの積率相関」
      - いわゆる相関係数

量的データの視覚化（attitudeを使用）

- 箱ひげ図
  - 「グラフ」→「箱ひげ図」
  - 変数を選択
    - 「層別のプロット」を選択することで，質的変数ごとに箱ひげ図を出力することができる
- 折れ線グラフ
  - 「グラフ」→「折れ線グラフ」
    - ※折れ線グラフは本来時系列データに適用するものであり，ここでは紹介だけ
- ヒストグラム
  - 「グラフ」→「ヒストグラム」
  - 変数を選択
  - 区間の数は標準は<auto>になっているが任意に設定することもできる

量的データの視覚化（attitudeを使用）

- 散布図
  - 「グラフ」→「散布図」
  - x変数，y変数を選択
  - 様々なオプションが設定できる
    - 点を確認する
      - 散布図上の点をクリックすることで対象の番号を確認できる
    - 周辺箱ひげ図
      - x軸，y軸の箱ひげ図を出力する
    - 最小2乗直線
      - 最小2乗直線を出力する
    - 層別プロット
      - 質的変数ごとにプロットされる対象が色分けされる

量的データの視覚化（attitudeを使用）

- 3次元散布図
  - 「グラフ」→「3次元グラフ」→「3次元散布図」
    - 目的変数（縦軸）と説明変数2つを選択
    - 得られた3次元散布図はマウスで回転ができる
    - 様々なオプションが設定できる
      - 線形最小2乗
        - 回帰平面が描かれる
      - 層別のプロット
        - 質的変数ごとにプロットされる対象が色分けされる

量的データの視覚化（attitudeを使用）

- 散布図行列
  - 多変数に関する散布図を一度に出力する
  - 「グラフ」→「散布図行列」
    - 対角部分はヒストグラムや箱ひげ図等を選択できる
    - オプションに関しては散布図，3次元散布図とほぼ同様

混合データの要約（Titanic, mtcarsを使用）

- 分割表による要約
  - 「統計量」→「要約」→「数値による要約」
    - 「層別して要約」で層別したい変数を選択する
  - 「統計量」→ 「要約」→「層別の統計量」
  - データによっては層別のために「数値変数を因子に変換」を行う必要があることがある
    - mtcarsのデータでamという質的変数 (0 or 1) を因子に変換（0をATに1をMTにしてみる）

コラム：数値変数を因子にする

- 「データ」→「アクティブデータセット内の変数の管理」→
  - 「数値変数を因子に変換」
    - 数値変数を因子に変換できる
    - 水準名を指定するか，数値をそのまま水準名にする
    - 新しい変数名をつけることもできる
  - 「数値変数を区間で区分」
    - 数値変数を任意の数に区分することができる
    - 水準名や区分の方法を選択する
    - 新しい変数名をつけることもできる

第2部 15:30-17:00

R入門（その2）

回帰分析，ロジスティック回帰分析，主成分分析等

回帰分析

attitudeを使用

回帰分析

- 回帰分析
  - Ratingを目的変数，残りを説明変数にする
  - 「統計量」→「モデルへの適合」→「線形モデル」
    - 因子になっている変数は自動的にダミー変数として扱われる
    - 因子になっている変数を用いない分析であれば「統計量」→「モデルへの適合」→「線形回帰」でもよい

回帰分析

- 出力結果

```
lm(formula = rating ~ advance + complaints + critical + learning +
    privileges + raises, data = attitude)

Residuals:
     Min       1Q   Median       3Q      Max
-10.9418  -4.3555   0.3158   5.5425  11.5990

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.78708   11.58926   0.931 0.361634
advance     -0.21706    0.17821  -1.218 0.235577
complaints   0.61319    0.16098   3.809 0.000903 ***
critical     0.03838    0.14700   0.261 0.796334
learning     0.32033    0.16852   1.901 0.069925 .
privileges  -0.07305    0.13572  -0.538 0.595594
raises       0.08173    0.22148   0.369 0.715480
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.068 on 23 degrees of freedom
Multiple R-squared: 0.7326,     Adjusted R-squared: 0.6628
F-statistic:  10.5 on 6 and 23 DF,  p-value: 1.24e-05
```

回帰分析

- 出力結果の概要
  - Residuals
    - 得られた回帰式に基づいて計算された予測値と実測値の残差の四分位数
  - Coefficients
    - 各変数および定数項における回帰係数，標準誤差，t値，P値（Pr(>|t|)）
    - Signif では，P値に基づいて表示される記号についての説明がなされている
  - Residual standard error
    - 残差の標準誤差
  - Multiple R-squared
    - 重相関係数
  - Adjusted R-squared
    - 自由度調整済み重相関係数

回帰分析

- 回帰係数の信頼区間の計算
  - 「モデル」→「信頼区間」
  - 信頼水準を設定（今回は0.95とする）

	Estimate	2.5 %	97.5 %
(Intercept)	10.78707639	-13.18712881	34.7612816
advance	-0.21705668	-0.58571106	0.1515977
complaints	0.61318761	0.28016866	0.9462066
critical	0.03838145	-0.26570179	0.3424647
learning	0.32033212	-0.02827872	0.6689430
privileges	-0.07305014	-0.35381806	0.2077178
raises	0.08173213	-0.37642935	0.5398936

ロジスティック回帰分析

mtcarsを使用

ロジスティック回帰分析

- 目的変数が連続量ではなく2値（または多値）の時に用いる

- 「統計量」→「モデルへの適合」→「一般化線型モデル」
- 「統計量」→「モデルへの適合」→「多項ロジットモデル」
  - 多項ロジットモデルは目的変数が「数値変数を因子に変換」がなされていないと選択ができない

  - 以降，一般化線形モデルを用いる

ロジスティック回帰分析

- 一般化線形モデル
  - am（2値のもの）を目的変数，mpgを説明変数とする
  - リンク関数族はbinominal
    - gaussian等，別の関数族も利用できる
  - リンク関数はlogit
    - リンク関数族に対応して選択できる関数が表示されている

ロジスティック回帰分析

- 出力結果

```
glm(formula = am ~ mpg, family = binomial(logit), data = mtcars)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5701  -0.7531  -0.4245   0.5866   2.0617

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -6.6035     2.3514  -2.808  0.00498 **
mpg           0.3070     0.1148   2.673  0.00751 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.230  on 31  degrees of freedom
Residual deviance: 29.675  on 30  degrees of freedom
AIC: 33.675

Number of Fisher Scoring iterations: 5
```

ロジスティック回帰分析

- 出力結果の概要
  - Deviance Residuals
    - 得られた回帰式に基づいて計算された予測値と実測値の残差の四分位数
  - Coefficients
    - 各変数および定数項における回帰係数，標準誤差，z値，P値（Pr(>|z|)）
    - Signif では，P値に基づいて表示される記号についての説明がなされている
  - AIC
    - AIC（赤池情報量規準）
  - Number of Fisher Scoring iterations
    - 繰り返し数

ロジスティック回帰分析

- ロジスティック回帰係数の信頼区間の計算
  - 「モデル」→「信頼区間」
  - 信頼水準を設定（今回は0.95とする）

```
                Estimate       2.5 %      97.5 % exp(Estimate)        2.5 %      97.5 %
(Intercept) -6.6035267 -12.3281402 -2.7717638   0.001355579 4.425443e-06 0.06255158
mpg          0.3070282   0.1220088  0.5874914   1.359379288 1.129764e+00 1.79946863
```

  - exp(Estimate) の項目がオッズ比を表している

回帰分析の変数選択について

変数選択について

- 回帰分析やロジスティック回帰分析で変数選択を行う方法
  - まず分析を行う
    - この時には分析に用いる全ての変数で分析を行う
  - 「モデル」→「逐次モデル選択」
    - モデル選択を行う方向（方法）や基準（AICかBIC）を選択できる
  - 次の2枚のスライドでは，回帰分析の結果を利用して，方向は変数減少法，基準はAICで変数選択を実行して得られた出力を示す
    - 全ての変数を用いたときのAICの値や，各変数を除いた時のAICの値が出力される
  - なお，「モデル」→「アクティブモデルを選択」で，変数選択を行うモデル（分析）を選択できる
    - 過去の分析を呼び出せる

変数選択について

- 最初にcriticalが削除される

```
Direction:  backward
Criterion:  AIC

Start:  AIC=123.36
rating ~ complaints + privileges + learning +
    raises + critical + advance

             Df Sum of Sq    RSS    AIC
- critical    1      3.41 1152.4 121.45
- raises      1      6.80 1155.8 121.54
- privileges  1     14.47 1163.5 121.74
- advance     1     74.11 1223.1 123.24
<none>                    1149.0 123.36
- learning    1    180.50 1329.5 125.74
- complaints  1    724.80 1873.8 136.04

Step:  AIC=121.45
rating ~ complaints + privileges + learning +
    raises + advance

             Df Sum of Sq    RSS    AIC
- raises      1     10.61 1163.0 119.73
- privileges  1     14.16 1166.6 119.82
- advance     1     71.27 1223.7 121.25
<none>                    1152.4 121.45
- learning    1    177.74 1330.1 123.75
- complaints  1    724.70 1877.1 134.09
```

変数選択について

- 次にraises
- その次にprivileges
- さらにadvance
- これで終了
  - 残った変数の係数が出力される

```
Step:  AIC=119.73
rating ~ complaints + privileges + learning + advance

             Df Sum of Sq    RSS    AIC
- privileges  1     16.10 1179.1 118.14
- advance     1     61.60 1224.6 119.28
<none>                    1163.0 119.73
- learning    1    197.03 1360.0 122.42
- complaints  1   1165.94 2328.9 138.56

Step:  AIC=118.14
rating ~ complaints + learning + advance

             Df Sum of Sq    RSS    AIC
- advance     1     75.54 1254.7 118.00
<none>                    1179.1 118.14
- learning    1    186.12 1365.2 120.54
- complaints  1   1259.91 2439.0 137.94

Step:  AIC=118
rating ~ complaints + learning

             Df Sum of Sq    RSS    AIC
<none>                    1254.7 118.00
- learning    1    114.73 1369.4 118.63
- complaints  1   1370.91 2625.6 138.16

Call:
lm(formula = rating ~ complaints + learning, data = attitude)

Coefficients:
(Intercept)   complaints     learning
     9.8709       0.6435       0.2112
```

コラム：部分集合の表現とは

- R Commanderで分析を行う際に「部分集合の表現」という欄がある
  - 標準では「<すべての有効なケース>」となっている
  - 分析するデータを条件付けて絞りたい場合，この部分に式を入力する

  - 例：回帰分析でamが1のものだけを用いて分析したい場合
    - am==1

  - 例：2番目から7番目のデータだけを用いて分析したい場合
    - 2:7

コラム：結果等の保存

- R Commanderを利用して計算した結果等を保存する方法
  - 「ファイル」→「スクリプトを保存」
    - 作業フォルダに%logfilenameというファイル名でスクリプトウィンドに表示されている内容が保存される
  - 「ファイル」→「スクリプトを名前を付けて保存」
    - 保存される内容は上記と同様だがファイル名や保存先を任意に変更できる
  - 「ファイル」→「出力を保存」
    - 出力ウィンドに表示される内容が保存される
    - 保存先のフォルダやファイル名はメモ欄に表示されている
  - 「ファイル」→「出力をファイルに保存」
    - 保存される内容は上記と同様だがファイル名や保存先を任意に変更できる

コラム：結果等の保存（続き）

- 「ファイル」→「Rワークプレースの保存」
  - .RData形式でデータやオブジェクト等が保存される
  - 保存先のフォルダやファイル名はメモ欄に表示されている
- 「ファイル」→「Rワークプレースに名前を付けて保存」
  - 保存される内容は上記と同様だがファイル名や保存先を任意に変更できる
- 「モデル」→「計算結果をデータとして保存」
  - 求めた統計量をアクティブデータに追加できる
- 「データ」→「アクティブデータセット」→「アクティブデータセットの保存」
  - （データ名）.RData形式でデータが保存される
- 「データ」→「アクティブデータセット」→「アクティブデータセットのエクスポート」
  - 変数名や行名をつけるかどうか，区切り記号は何にするか等を選択してテキスト形式（.txtや.csv等）でデータを保存できる

主成分分析

attitudeを使用

主成分分析

- 主成分分析
  - 「統計量」→「次元解析」→「主成分分析」
  - 変数はrating以外を選択
    - ちなみに因子になっている変数は表示されない（今回は関係ないが）
  - 「相関行列の分析」，「スクリープロット」，「データセットに主成分得点を保存」にチェック
  - 実行すると保存する主成分数を聞かれる画面が出てくる
    - 「データセットに主成分得点を保存」にチェックをした場合
    - 分析対象とした変数の数まで保存することにする
  - スクリープロットの棒グラフが出力される

主成分分析

- 出力結果

```
> unclass(loadings(.PC))  # component loadings
               Comp.1     Comp.2       Comp.3      Comp.4      Comp.5      Comp.6
advance    -0.3808011 -0.3207060  0.686643142 -0.20574245 -0.25472836  0.41646475
complaints -0.4393752  0.3126424 -0.445166951  0.31601946 -0.19152122  0.61194923
critical   -0.2248130 -0.8022474 -0.457245609 -0.09994698  0.28887525  0.05784728
learning   -0.4614010  0.2170870  0.271981397  0.22479562  0.77564752 -0.11767060
privileges -0.3947108  0.3087507 -0.217413750 -0.81484689 -0.03768625 -0.19029420
raises     -0.4926576 -0.1155323 -0.005604908  0.36510795 -0.46036381 -0.63140375

> .PC$sd^2  # component variances
   Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6
3.1692232 1.0063467 0.7629087 0.5525165 0.3172465 0.1917584

> summary(.PC) # proportions of variance
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4     Comp.5     Comp.6
Standard deviation     1.7802312 1.0031684 0.8734465 0.74331451 0.56324638 0.43790225
Proportion of Variance 0.5282039 0.1677245 0.1271515 0.09208608 0.05287441 0.03195973
Cumulative Proportion  0.5282039 0.6959283 0.8230798 0.91516586 0.96804027 1.00000000
```

主成分分析

- 出力結果の概要
  - > unclass(loadings(.PC))　# component loadings の部分
    - 固有ベクトル
  - > .PC$sd^2　# component variances の部分
    - 固有値
  - > summary(.PC) # proportions of variance の部分
    - Standard deviation
      - 主成分の標準偏差（固有値の平方根）
    - Proportion of Variance
      - 寄与率（固有値の合計に対する個々の主成分の割合）
    - Cumulative Proportion
      - 累積寄与率

主成分分析

- 主成分得点から散布図を描く
  - 「グラフ」→「散布図」
  - x変数にPC1を，y変数にPC2を選択
  - 「点を確認する」にチェック
  - 「最小2乗直線」，「平滑線」，「ばらつき幅の表示」のチェックを外す

クラスター分析

attitudeを使用

クラスター分析

- クラスター分析
  - 階層的クラスター分析
  - 非階層的クラスター分析
    - k-means
    - 重複クラスター分析
    - など
  - 階層的クラスター分析と非階層のk-meansをR Commanderを用いて実習する

階層的クラスター分析

- 階層的クラスター分析
  - 「統計量」→「次元解析」→「クラスタ分析」→「階層的クラスタ分析」
    - 変数を選択（ratingを除くすべて，主成分分析の結果で得られた変数も除く）
    - クラスタリングの方法を選択
    - 距離の測度を選択
    - 「デンドログラムを描く」にチェック
    - R Commanderのスクリプトウィンドウに表示されたplot分の丸括弧の最後に「, hang=-1」を入れて実行することで，足の部分が揃ったデンドログラムが得られる

階層的クラスター分析

- より詳しく分析するためには・・・
  - R Commanderを使わず，R Console画面に直接入力するほうがよい
  - また，データが矩形データではなく，類似度データや距離行列のデータの場合はR Commanderを使ってクラスター分析を行うことができない
  - データを距離行列にする
    - dist(データ, method= "euclidean“)
    - methodはeuclidean, maximum, manhattan, canberra, binary, minkowskiが指定できる（標準はeuclidean）
  - 階層的クラスター分析を行う
    - hclust(距離行列, method= "ward")
    - methodはward, single, complete, average, mcquitty, median, centroidが指定できる（標準はcomplete）

k-means法によるクラスター分析

- 非階層的クラスター分析
  - k-means法
    - 「統計量」→「次元解析」→「クラスタ分析」→「k-平均クラスタ分析」
    - 変数を選択（ratingを除くすべて，主成分分析の結果で得られた変数も除く）
    - クラスタ数：今回は4
    - シードの初期値の数：10
    - 最大繰り返し数：10

    - 「クラスタのサマリの表示」にチェックすることで結果の要約が出力される
    - 「クラスタのバイプロット」にチェックすることで，主成分分析の結果を基にしたバイプロットが出力される

その他

- 今回実習した他にもR Commanderでは
  - 確率分布を描く
  - 確率分布の分位点を得る
  - 各種検定統計量を求める・検定を行う
  - 情報量規準の値を求める
- 等々，様々なことが実行可能である
- さらにR Commanderからさらにパッケージをロードすることで機能拡張もできる
- Rのスクリプトプログラミングと併せて利用すると非常に便利かつ強力なツールとなる

参考文献・参考URL

- 参考文献
  - 大森・阪田・宿久 著 (2011). R Commanderによるデータ解析, 共立出版.
  - 舟尾 著 (2005). The R Tips, 九天社.
    - 第2版はオーム社より出版
- 参考URL
  - R project 公式ページ http://www.r-project.org/
  - RjpWiki （日本語のR情報） http://www.okada.jp.org/RWiki/
  - R Commander Installation Notes http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/installation-notes.html
  - Jin's HP（同志社大学 金明哲先生のページ） http://mjin.doshisha.ac.jp/R/
  - 日本行動計量学会 第14回春の合宿セミナー R入門コース https://appsv.main.teikyo-u.ac.jp/~satoru/societies/bsjspringseminar14/