人間行動システム統計演習 A

第 2 回：データセットの操作　担当：中川、寺井、和嶋 *

2015 年 4 月 16 日

1　ファイルのダウンロード

東工大ポータルにログインし、Tokyo Tech OCWi から `ten1.csv`、`ten2.csv`、`pref_car.csv` をダウンロードし、保存してください。

2　データファイルの読み込み

2.1　データファイルの形式

`ten1.csv` と同様の形式のデータを準備すると R に読み込ませることが容易です。各自コピーした `ten1.csv` を開いて確認してみてください

ten1.csv の内容

```
Name,Kokugo,Sansu,Rika,Syakai,Eigo,Blood
Obayashi,66,58,44,32,51,A
Yamamoto,64,68,72,55,61,B
Natsume,63,70,63,55,61,B
Matsuki,67,56,50,39,48,O
         :
```

コピーしてもらったデータファイルは以下のように整形されています。

- テキスト形式
- 行列形式のデータ（行：標本，列：変数）
- データの区切りはカンマ [*1]
- 一行目は変数名
- 欠損値は `NA`

---

* 御質問などは寺井 (terai.a.aa@m.titech.ac.jp) にメールにてご連絡下さい。

*1 スペースやタブでも構いません。

データを自分で加工・作成する人はこの形式で用意しておくと、読み込みが容易になります。Excel 等の表計算ソフトを利用すれば、カンマ区切り、スペース区切り、タブ区切り等のテキスト形式で保存することができます。Excel を用いてカンマ区切りデータを作成する場合は、Excel にデータを入力した後、「オフィスボタン」->「名前を付けて保存」->「その他の形式」をクリックし、「ファイルの種類」欄にて「CSV(カンマ区切り)(*.csv)」を選択したうえで保存を行ってください。[*2]

なお本演習では触れませんが、Excel、Access のファイルや SPSS のデータファイルを直接読むこともできます。さらに、データベースからデータをとりこむことも可能です [*3]。データおよび、変数名、標本名に日本語を使うこともできます。テキスト処理では、極めて便利な機能です。[*4]

2.2 R コマンダーでのデータの読み込み

R コマンダーを用いて、CSV ファイル（カンマ区切り）のデータを読み込みます。

- R コマンダーを立ち上げる (> library(Rcmdr))
- 「データ」→「データのインポート」を選択
- 「テキストファイルまたはクリップボードから」を選択 (CSV ファイルの場合。その他 SPSS などのデータセットから選択する場合は、適宜自分で読み替えてください。)
- 「フィールドの区切り記号」でカンマを指定（カンマ区切りの場合。データにあわせて設定してください。）
- 自分のファイルに合わせた指定（データセット名：ten1、欠損値の記号：NA、小数点の記号：ピリオド）を終えて「OK」をクリック
- 「ファイルを開く」ウインドウが開きますので、ファイルを指定

読み込みが完了すると、R コマンダーのデータセットの部分に「青色」で指定した名前（デフォルトでは「Dataset」）が書かれ、スクリプトウィンドウと出力ウィンドウにデータセットをインポートした際のコマンドとその結果が出力されます。

次に、読み込んだデータを表示します。読み込んだデータが複数の場合は、処理対象のデータセットをアクティブ化させなければなりません。「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選択」をクリックして表示したいデータセットを選択してください。アクティブ化されたデータセットの名前が、R コマンダーのデータセットの部分に「青色」で表示されます。その後、データセット名の横にある「データセットを表示」と書かれたボタンをクリックすることで、データを表示できます。どんな時でも、表示されたデータを見て間違ったデータセットを表示していないか確認しましょう。

[*2] 「テキスト（スペース区切り）(*.prn)」、「テキスト（タブ区切り）(*.txt)」を選択することでタブ区切り、スペース区切りのデータとして保存することも可能です。

[*3] 特別なファイル形式の読み込みについては、help や書籍で調べてください。

[*4] ただし、異なる OS 間では、エラーの原因になります。従って、日本語のテキスト処理を行う場合、および個人でしかデータを扱わない場合を除き、日本語を利用するのは避けるのが無難です。

ファイルに変数名が含まれていない場合は、「V1, V2, V3, ...」という変数名が自動的に付けられます。読み込んだデータの変数名を変更したい場合は、「データ」→「アクティブデータセット内の変数の管理」→「変数名をつけ直す」をクリックし、変更した変数名を選択した後、新しい変数名を入力してください。ten2.csv はファイルに変数名が含まれていません。その点に注意して、ten2 という名前で ten2.csv を読み込んでください。その後、変数名を ten1 と同様のものに変更してください。

また「データセットの編集」をクリックするとデータセットの編集画面が現れます。こちらは自分で編集できるようになっています。しかし、データの編集内容は記録に残りません。編集済みのファイルを使用し、R コマンダー上ではなるべく編集しないことをお勧めします。

2.3 R コンソールでのデータの読み込み

2.3.1 R のディレクトリ操作

R のコンソールにコマンドを入力し、ファイルを読み込むためにはファイルがどのディレクトリ（フォルダ）にあるかを R に知らせる必要があります。その方法として、

- 読み込み時のファイル指定で、ファイル名だけではなくファイルが存在するディレクトリ名も含めて指定する
- R が参照するディレクトリをデータファイルが存在するディレクトリに変更し、読み込み時にはファイル名のみ指定する

方法があります。

ワークディレクトリの変更は「ファイル」->「ディレクトリの変更」(Mac の場合は、「その他」->「作業ディレクトリの変更」）を用い、データが保存されているフォルダ（ディレクトリ）を選択します。

2.4 read.table( ) によるデータファイル読み込み

ファイルの読み込みは read.table( ) 関数を用います。

```
> ten1 <- read.table(file   = "ten1.csv",  # ファイル名の指定
+                    header = TRUE,          # ファイルの一行目は変数名
+                    sep    = ","            # データの区切り記号
+                      ) ↵
```

上記の書式でオブジェクト ten1 に ten1.csv ファイルの内容が読み込まれます。

- データファイルの 1 行目が変数名の場合は header = TRUE を指定します。変数名がない場合は、FALSE と指定します。変数名は v1,v2,v3... と付けられます

- 区切り文字が空白やタブ以外の場合は sep = **“区切り文字”**で指定します [*5]
- 欠損値を表す文字が NA 以外の場合は na.strings = **“欠損値”**で指定します

2.5 変数の参照

読み込んだオブジェクト ten1 の内容を参照します。以下のコマンドはスクリプトウィンドウで実行してください。データセットを行列のように**オブジェクト名** [m, n] と指定することで、m 行 n 列目の要素が参照できます [*6]。指定を省略すると省略された行もしくは列はすべて表示されます。従って、 **オブジェクト名**[,] は、全データを表示します。

```
> ten1[1, ] ↵    # 1 行目データの参照
> ten1[ ,2] ↵ # 2 列目データの参照
```

連続した範囲、飛び飛びの範囲も参照できます。下記の最初の例では、4 人目までの全データを参照しています [*7]。次の例では、連続でない対象の指定方法を示しています。この場合、関数 c( ) を用いて範囲をベクトルで与えます。

```
> ten1[1:4, ] ↵
> ten1[ , c(2:4,7)] ↵
```

データセット型のオブジェクトは、上に加えて、**オブジェクト名$変数名**による変数の参照ができます。

```
> ten1$Blood ↵
```

変数が文字型の場合、値のリストが Levels: に続けて表示されます。

2.5.1 変数の追加・削除

新しい変数を追加する場合は、「データ」→「アクティブデータセット内の変数の管理」→「新しい変数を計算」をクリックします。アクティブデータセットとして ten1 を選択し、「新しい変数名」に ”Newvar”, 「計算式」に” 1” をいれて、「OK」をクリックします。Newvar として定数 1 を持つ変数が作成されます。同様に、「新しい変数名」に ”DifKokugo”, 「計算式」

[*5] 空白の場合 sep = " " 、 タブの場合 sep = "\t" と指定しても構いません。カンマの場合は、 sep = "," と指定します。

[*6] 属性が行列のオブジェクトにも同様の参照できます。

[*7] 同じことを head(**データセット**, n = 4) でも実行できます。また、データの末尾を参照するには、同様に tail(**データセット**, n =) が使えます。n =を省略すると 6 行表示します。

に”Kokugo-mean(Kokugo)” をいれて、「OK」をクリックします。DifKokugo として国語の得点の平均からの偏差を持つ変数が作成されます。

「データ」→「アクティブデータセット内の変数の管理」→「データセットから変数を削除」をクリックし、「Newvar」「DifKokugo」を選択し（Ctrl キーを押しながら選択することで、複数の変数を選択することができます)、「OK」をクリックすることで変数を削除することができます。

2.6 条件に基づく標本抽出

条件に合致する行（オブザベーション）のみを抽出するには、「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部分集合を抽出」をクリックします。「部分集合の表現」に抽出条件”Kokugo > 70”、新しいデータセット名に”KokugoOver70” を記述し、「OK」をクリックすることで、国語の得点が 70 点より大きいオブザベーションのみを抽出した新しいデータセット”KokugoOver70” を作成できます。同様の操作を行い社会の得点が 50 点より大きいオブザベーションのみを抽出し、新しいデータセット”SyakaiOver50” を作成してみてください。

”Blood == "B" & ten1$Syakai > 50” というように、論理演算子を用いることで「血液型がB かつ社会の得点が 50 点を超える」といった条件でのオブザベーションの抽出も可能です。

機能	比較・論理演算子
一致	x == y
不一致	x != y
x は y より大きい	x > y
x は y より大きいか等しい	x >= y
x は y より小さいか等しい	x <= y
条件での論理和	x || y
条件での論理積	x && y
排他的論理和	xor(x,y)

「データ」→「アクティブデータセット」→「欠測値のあるケースを削除」をクリックすることで、欠損のあるオブザベーションを削除することができます。

2.7 データの並び替え

ある変数の値に関して小さい順（昇順）あるいは大きい順（降順）にデータを並び替える作業をソートと言います。ソートを行う場合は、order() 関数を用いてスクリプトウィンドで実行して下さい。order() 関数によりキー変数のソート結果（行番号のリスト）をオブジェクトに代入し、そのオブジェクトを基にデータセットにアクセスし、ソート結果を作成します。

```
> Sansu.o <- ten1[order(ten1$Sansu), ] ↵ # 算数昇順
> Kokugo_do <- ten1[order(ten1$Kokugo, decreasing = TRUE), ] ↵ # 国語降順
```

```
> EigoSansu.o <- ten1[order(ten1$Eigo, ten1$Sansu), ] ↵ # 英語、算数昇順
```

最初は算数の点数の低い順、次は国語の点の高い順、最後は、英語を第 1 基準に、タイの要素に関しては、算数の点数を手がかりに、低い順に並べています。データを降順に並び替えたい場合は order( ) 関数内に、 "decreasing = TRUE" のオプションを指定します。2 つ以上の変数を手がかりにソートしたい場合は、優先度の高い順に当該変数を指定します。最初の変数でタイのものは、2 番の変数の大小関係から並べられます。2 番目以降の変数にもタイの成分がある場合は、出現順に並べられます。ソートした結果、読み込んだデータでの出現順に振られたオブザベーション番号をそのまま保存しているため、行番号はバラバラになります。行番号を整え直す（新たにオブザベーション番号を振りなおす）方法は関数 rownames(**データセット**) <- c(**新しい行番号列**) で行うことができます。

3 複数データセットの結合

3.1 縦につなぐ

「データ」→「データセットの結合」を選択し、結合したい 2 つのデータセットを選択し、「行単位での結合」を選択してください。ten1、ten2 のデータを縦につないぎ、新しいデータセット ten を作成してください。

3.2 横につなぐ

R コマンダーでは共通の変数を明示的に指定してデータを横に結合する機能はありません。そのため、共通の変数を指定してデータを横につなぐには、スクリプトウィンドで関数 merge() を使います。下記の例では、ten1 と pref で共有されている Name の列を指定して、両データセットを横につなぎあわせています。pref_car.csv を pref というデータセットとして読み込んでからコマンドを実行してください。

```
> ten1.pref <- merge(ten1, pref, by.x = "Name", by.y = "Name") ↵
# merge の基本的書式
> ten1All.pref <- merge(ten1, pref, by.x = "Name",
      by.y = "Name", all.x = TRUE) ↵  # ten1 を全て表示
```

1 つ目が基本的な merge() の書式になります。ここでは、ten1、pref、それぞれのデータセットの参照列を明記していますが、同じ列名なので、by = "Name"と指定しても同じ結果が得られます。

2 つ目の書式は、どちらかのデータセットの全データを結果に含めたい場合の書式です。上記例

では、ten1 に含まれる全データを表示するよう指示しています。ten1 にあって pref にないデータについては NA が表示されます。また、全てのデータを表示したい場合は、同オプション部分を all = TRUE とすれば OK です。

共通の変数を明示的に指定せずにデータを横に結合する場合は、「データ」→「データセットの結合」を選択し、結合したい 2 つのデータセットを選択し、「列単位での結合」を選択してください。

4 データセットの書き出し

データセットをファイルに書き出すことができます。「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」をクリックし、諸設定を行った後、データセットをファイル (テキストファイル) に出力させます。

5 データの要約

「統計量」→「要約」→「アクティブデータセット」を選択することで、numeric 型の変数の基本統計量を確認することができます。Min：最小値、1st Qu：第 1 四分位数、Median：中央値、Mean：平均値、3rd Qu：第 3 四分位数、Max：最大値を表します。また、欠損値（NA）の個数を確認することもできます。

「統計量」→「要約」→「数値による要約」をクリックし、処理を行いたい numeric 型の変数と factor 型の変数 (カテゴリ変数) を選択することで、各カテゴリごとの基本統計量を出力できます。mean：平均値、sd：標準偏差、IQR：四分位範囲、0% 25% 50% 75% 100% 各 % タイル点、 n：データ数、 NA：欠損値数です。

カテゴリ変数の頻度集計は「統計量」→「要約」→「頻度分布」を選択し、対象となる変（カテゴリ変数）を選択します。水準ごとの頻度、相対頻度が出力されます。

5.1 R コンソールでの基本統計量の計算

平均、合計、分散、標準偏差、中央値、最小値、最大値、範囲等は、関数を用いて計算することができます。これらはそれぞれ mean(変数名)、sum(変数名)、var(変数名)、sd(変数名)、median(変数名)、min(変数名)、max(変数名)、range(変数名) で求めることができます。欠損がある場合は、sum(**変数名**, na.rm = TRUE) といった指定をしないと、欠損を除外した計算をしてくれませんので注意してください。

6 カテゴリー変数の処理

6.1 数値データのカテゴリー化

ten1.csv、ten2.csv では、Blood は A, B, O, AB のように文字で書かれており、それを読み込んだ場合には文字もしくは文字列として表示され、R の内部では factor 型というカテゴリ変数として認識されています。しかし、データファイル pref_car.csv の Class や Car という変数のように、あらかじめ数値によって入力されている変数は、カテゴリー変数であることを R に認識させる必要があります。「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」をクリックし、対象となる変数を選択します。「水準名を指定」を選択すると、各水準に対する新たなラベルを入力する画面が出力されます。「数値で」を選択すると、各水準に対するラベルとして数値がそのまま用いられます。Class に対し、水準 1, 2, 3 をそれぞれ C1,C2,C3 として factor 型の変数 Class1 を作成してください。

6.2 連続変数をカテゴリー化する

連続変数を任意の区間にカテゴリー化する方法を説明します。「データ」→「アクティブデータセット内の変数の管理」→「連続変数を区間で区分」をクリックします。対象となる変数を選択し、区間の数、水準名、区分の方法等を設定し、「OK」をクリックしてください。

R コマンダーでは任意の区分指定には対応していません。そのため、任意の区分でのカテゴリ化を行う場合はスクリプトウィンドから cut( ) を利用して、連続変数を任意の区間にカテゴリー化してください。

```
# 区間を指定して、カテゴリー化
> ten$KokugoC <- cut(ten$Kokugo, breaks = c(-Inf, 60, 70, 80, Inf),
+ right = FALSE, labels = c("D", "C", "B", "A") ,
+ ordered_result = TRUE) ↵
```

cut( ) の内容ですが、最初にカテゴリー化する変数を指定しています。分割点は、breaks = c(**数値** 1，**数値** 2，**数値** 3，...，**数値** n) という形式で指定します。これにより、数値 1 以上数値 2 未満、数値 2 以上数値 3 未満、…、数値 n-1 以上数値 n 以下と分割されます。数値 1, 数値 n としてデータの最小値、最大値を、事前に調べ、定義することもできますが、代わりにマイナス無限とプラス無限を表す-Inf と Inf で代用することもできます。

right = FALSE の部分は、区間の下限点は区間に含まれるが（〜以上)、上限点は含まれないようにする（〜未満）ことを意味しています [*8]。

[*8] デフォルト設定は、right = TRUE です。従って、この部分を省略すると、“ 〜を超える〜以下 ” になります。

labels はそれぞれの区間の名前です。従って、上記例では、60 点未満に D を、60 点以上 70 点未満に C、70 点以上 80 点未満に B、80 点以上を A に分類しています。

ordered_result は、カテゴリー間の順序関係を持たせるか否か、のオプションです。もちろん、順序関係のないカテゴリー変数に指定することはできませんが、上記例のように順序関係が存在する場合は、指定しておくと解析の際利用することができます。連続変数のカテゴリー化には、以下の書式を利用します。

- cut(**変数名**, breaks = **分割点**, right = FALSE,
  labels = **水準のラベル**, ordered_result = TRUE)

7 コマンドの再利用

7.1 スクリプトファイルの準備

これまで R コマンダーでコマンドを実行する操作を説明してきましたが、この方法は同じコマンドを再利用するのには向いていません。コマンドを効率よく利用するにはコマンドを並べた内容のファイル（**R スクリプト**）を作成します。「ファイル」→「スクリプトの保存（スクリプトに名前をつけて保存)」をクリックすることで実行したスクリプトを保存することができます。

7.2 スクリプトファイルの読み込み

「ファイル」→「スクリプトファイルを開く」をクリックすることで、保存したスクリプトファイルを開くと、保存されたコマンドがスクリプトウィンドに表示されます。実行したいコマンドを選択し、「実行」ボタンをクリックしてください。

8 本日の課題

1. ten1.csv と ten2.csv には実は同じ名前のオブザベーションが存在する。2 つのファイルを読み込み、どちらのデータのオブザベーションか区別するための新しいカテゴリ変数（”Group”）を追加し、２つのデータを（縦に）結合したデータセットを作成する。
2. 1 で作成したデータセットに、算数の成績が平均点以下を”L”、平均点を超えるものを”H”に分類した新しいカテゴリ変数”SansuC”を追加する。
3. 上記を実行できるスクリプトファイル (データの読み込みを含む) を”Kadai2.R”という名前で作成する。

Kadai2.R を、OCWi から提出してください。課題の締め切りは次回の講義の前日です（4 月 22 日 23:59:59)。