

## 1. Excel VBA の NDB オープンデータへの活用

### 1-1. NDB (National Database of Health Insurance Claims and Specific Health Checkups of Japan) オープンデータとは

厚生労働省では、「高齢者の医療の確保に関する法律」に基づき、2009（平成 21）年からレポート情報並びに特定健康診査（以下「特定健診」という）・特定保健指導情報を収集した、NDB オープンデータの運用が開始された。本データは、日本の医療ビッグデータを扱った統計資料の 1 つとしてヘルスケアに関心を有する多くの国民に役立ててもらふことを期待して公表されている。国民皆保険下にある日本では全数に近い割合で国民の医療動向を把握することができるため、保健・医療を対象とした政策の立案や研究を遂行するにあたって、世界でも有数の規模と悉皆性を誇るデータベースである<sup>1) 2)</sup>。

第 5 回公表データ（平成 29 年度分）のうち、本演習の題材とする特定健診検査項目※<sup>1</sup>については、「BMI」、「腹囲」、「空腹時血糖」、「HbA1c」、「収縮期血圧」、「拡張期血圧」、「中性脂肪」、「HDL コレステロール」、「LDL コレステロール」、「AST」、「ALT」、「 $\gamma$ -GT」、「貧血検査」、「眼底検査」、「尿蛋白」、「尿糖」が、特定健診質問票項目※<sup>2</sup>については、「標準的な質問票 22 項目」の約 2,8600 万件分の集計データが公表されている（対象年齢：40～74 歳）<sup>3)</sup>。



図 1

#### 引用文献

1) 第 5 回 NDB オープンデータ概要：[000539640.pdf \(mhlw.go.jp\)](https://www.mhlw.go.jp/stf/shingi2/shingi2_000539640.pdf)

2) 第 5 回 NDB オープンデータ 【解説編】

厚生労働省保険局医療介護連携政策課 保険データ企画室令和 2 年 12 月

3) 標準的な健診・保健指導プログラム（平成 30 年版） 厚生労働省 健康局 H30 4 月

### 1-2. 医療情報データベースをさらに知りたい人へ

日本の医療ビッグデータを扱った統計資料は、大きく分けて民間が提供している医療情報データベースと政府主管の大規模医療情報データベースがある。

民間が提供している医療情報データベース：健保組合パネルレセプトデータ (JMDC)

DPC 病院データ (MDV)：急性期病院 (DPC 参加病院)

院外調剤レセプトデータ (JMIRI)

政府主管の大規模医療情報データベース：NDB, MID-NET

※1 : 特定健診検査項目

**特定健康診査受診結果通知表**

項 目	基準値	今	前	前	前々	前
		年 月 日	年 月 日	年 月 日	年 月 日	年 月 日
身 体 計 測	身 長 (cm)					
	体 重 (kg)					
	腹 囲 (cm)					
	B M I					
血 圧	収 縮 期 血 圧 (mmHg)					
	拡張期血圧 (mmHg)					
血中脂質検査	中 性 脂 肪 (mg/dl)					
	HDL-コレステロール (mg/dl)					
	LDL-コレステロール (mg/dl)					
	Non-HDLコレステロール (mg/dl)					
肝機能検査	G O T (IU/l)					
	G P T (IU/l)					
	γ - G T P (IU/l)					
血 糖 検 査 (Fasting blood sugar test)	空腹時血糖 (mg/dl)					
	ヘモグロビンA1c (NGSP値) (%)					
	随 時 血 糖 (mg/dl)					
尿 検 査	糖					
	蛋 白					

图 2

(裏面)

メタボリックシンドローム判定			
----------------	--	--	--

(備考)

1. この用紙は、日本工業規格A列4番とすること。
2. 「性別」の欄は、該当しない文字を抹消すること。
3. 基準値を外れている場合には、「\*」を測定結果欄に記入すること。
4. 「メタボリックシンドローム判定」の欄は、「基準該当／予備群該当／非該当」を記入すること。

图 3

### 標準的な質問票

图 4

2

## 第2部 (データ一編)

医科診療行為 (A~N)

歯科診療行為 (A～O)

•

特定保険医療材料

特定健診

特定健診（質問票）

**特定健診**




- **BMI** 脂連動性男児性年輪齡級別分布 [59K B]
- **GOT(AST)** 脂連動性男児性年輪齡級別分布 [48K B]
- **GPT(ALT)** 脂連動性男児性年輪齡級別分布 [48K B]
- **HbA1c** 脂連動性男児性年輪齡級別分布 [64K B]
- **HDLコレステロール** 脂連動性男児性年輪齡級別分布 [48K B]
- **LDLコレステロール** 脂連動性男児性年輪齡級別分布 [65K B]
- **ア-GTP(γ-GTP)** 脂連動性男児性年輪齡級別分布 [49K B]
- **ヘモグロビン** 脂連動性男児性年輪齡級別分布 [59K B](K)
- **血圧の平均値** 脂連動性男児性年輪齡級別分布 [199K B]
- **拡張期血圧** 脂連動性男児性年輪齡級別分布 [54K B]
- **脈波検査(SCOT分類)** 脂連動性男児性年輪齡級別分布 [34K B](K)
- **脈波検査(コースワグナー分類)** 脂連動性男児性年輪齡級別分布 [29K B](K)
- **脈波検査(シェイエ分類:H)** 脂連動性男児性年輪齡級別分布 [58K B](K)
- **脈波検査(シェイエ分類:S)** 脂連動性男児性年輪齡級別分布 [54K B](K)
- **空腹時血糖** 脂連動性男児性年輪齡級別分布 [55K B]
- **空腹時血圧** 脂連動性男児性年輪齡級別分布 [54K B]
- **中性脂肪** 脂連動性男児性年輪齡級別分布 [49K B]
- **尿酸** 脂連動性男児性年輪齡級別分布 [54K B]
- **尿酸** 脂連動性男児性年輪齡級別分布 [39K B]
- **尿酸** 脂連動性男児性年輪齡級別分布 [49K B]

(※)これらの抽出レコードは、NDBの特定健診情報のうち、基本情報レコードと健診結果・問診結果情報レコードを集計したものです。

- ④ [△ヘモグロビン 都道府県別性年齢階級別分布](#) [54K]B(※)
- ⑤ [△眼底検査\(SCOTT分類\) 都道府県別性年齢階級別分布](#) [58K]B(※)
- ⑥ [△眼底検査\(キスフグナー分類\) 都道府県別性年齢階級別分布](#) [54K]B(※)
- ⑦ [△眼底検査\(シェイェン分類:H\) 都道府県別性年齢階級別分布](#) [53K]B(※)
- ⑧ [△眼底検査\(ジョエイ分類:S\) 都道府県別性年齢階級別分布](#) [59K]B(※)

(※)これらの抽出レコードは、NDBの特定健診情報のうち、基本情報レコード、健診結果・問診結果情報レコード、詳細情報レコードを集計したものです。

**特定健診(質問票)**

-  [標準的な質問票\(質問項目1\)](#) [46KB]
-  [標準的な質問票\(質問項目2\)](#) [46KB]
-  [標準的な質問票\(質問項目3\)](#) [46KB]
-  [標準的な質問票\(質問項目4\)](#) [45KB]

特定健診項目：BMI

都道府県名		検査値階層 (kg/m <sup>3</sup> )	全体																				中計								
			男										女																		
			40～44歳		45～49歳		50～54歳		55～59歳		60～64歳		65～69歳		70～74歳		40～44歳		45～49歳		50～54歳			55～59歳		60～64歳		65～69歳		70～74歳	
			人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数		人数	人数	人数	人数	人数	人数	人数	人数
北海道	30.0以上	8,934	9,133	6,897	5,538	3,811	2,565	1,307	38,185	3,909	4,131	3,507	3,018	2,473	2,696	2,188	21,922														
	25.0以上30.0未満	28,755	31,129	27,842	27,636	24,229	20,705	13,368	173,664	9,648	11,106	10,688	10,616	10,238	13,091	11,308	76,782														
	20.0以上25.0未満	52,188	51,186	44,544	45,216	39,263	35,835	25,454	293,686	33,500	35,679	33,313	31,930	30,210	36,404	29,913	230,945														
	15.0以上20.0未満	6,156	5,481	4,516	4,438	3,843	3,406	2,362	30,202	13,845	12,595	11,337	9,992	8,073	8,182	6,041	70,065														
	18.5未満	2,533	2,122	1,777	1,880	1,769	1,531	1,105	12,717	9,192	8,040	7,725	7,207	5,645	5,518	4,104	47,433														
青森県	30.0以上	2,441	2,208	1,579	1,238	871	678	434	9,449	1,174	1,242	1,053	982	911	969	777	7,100														
	25.0以上30.0未満	7,190	7,570	7,234	7,278	6,302	6,000	4,220	45,794	2,783	3,258	3,491	3,778	4,111	5,210	4,509	27,140														
	20.0以上25.0未満	12,323	12,763	11,795	12,557	11,087	11,173	8,419	80,117	8,719	9,412	9,640	10,195	10,421	12,702	10,259	71,340														
	18.5以上20.0未満	1,431	1,419	1,224	1,346	1,166	1,129	853	8,568	3,179	3,000	2,717	2,705	2,380	2,457	1,783	18,222														
	18.5未満	554	547	495	588	492	525	432	3,633	1,934	1,816	1,662	1,803	1,531	1,540	1,181	11,445														
岩手県	30.0以上	2,329	2,240	1,707	1,334	969	734	472	9,780	1,161	1,180	1,031	1,016	902	1,074	862	7,220														
	25.0以上30.0未満	7,389	7,764	7,505	7,995	6,963	6,932	5,170	49,190	2,747	3,232	3,365	3,868	4,225	5,523	5,026	27,988														
	20.0以上25.0未満	13,393	12,760	12,200	13,312	11,206	10,367	85,583	9,100	9,266	10,203	10,926	10,601	13,720	11,379	7,556	73,561														
	18.5以上20.0未満	1,699	1,482	1,361	1,482	1,338	1,197	892	9,534	2,316	2,911	2,663	2,703	2,398	2,686	1,983	18,358														
	18.5未満	698	653	563	680	564	545	450	4,135	2,134	1,846	1,822	1,848	1,598	1,711	1,426	12,269														

图 5

4  
5

图 6

都道府県別、性、年齢別に集計されている。

## 2. 簡単なプログラムの作成

Excel では、プログラミング言語の 1 つである Visual Basic が利用可能である。Excel 内で利用できる Visual Basic は、Excel、Word、Access などのようなアプリケーションソフトウェア用に改良されたもので、Visual Basic for Applications (VBA) と呼ばれている。この VBA を用いると、Excel 上のデータに対して様々な処理を行うことができる。NDB オープンデータは Excel 形式のデータとして公開されているので、VBA をそのまま利用することが可能である。ただし、VBA を利用するためには、プログラミング言語の基本的な事項について理解しておく必要がある。ここでは、それを順番に説明していくが、授業では最初に 8. NDB オープンデータへの応用から進めて、必要に応じて基本的な事項に戻るようにする。

## 2-1. プログラムの編集

Excel でプログラムを作成するときは、専用のエディタである Visual Basic Editor (VBE)を使用する。まず、簡単なプログラムを作成してみる。



- (1) リボンの「開発」タブの「コード」グループにある「Visual Basic」ボタンをクリックする。
- (2) VBE が起動するので、メニューバーの「挿入」、「標準モジュール」を選択する。
- (3) Module1 が開くので、そこに例 1 を入力してみる。プログラムを入力するときは、特に日本語の入力が必要とき以外、常に日本語入力システムをオフにして半角英数モードにする。

## 例 1

```
Sub No1()  
    MsgBox "BMI が 25 以上は肥満傾向です"  
End Sub
```

例 1 は画面上に文字列を表示するプログラムである。最初の行の Sub はプログラムの開始、No1 はプログラムの名前、最後の行の End Sub はプログラムの終了を表しており、プログラムはこの間に記述する。なお、()は入力しなくても自動的に追加される。「Msgbox」は画面に文字列を表示する命令で、表示される文字列はプログラム上の命令と区別するために" "で囲まれている。" "が全角文字にならないよう注意する。「Msgbox」の左側に空白があるが、このように空白を入れると、プログラムの構成が理解し易くなる。空白は「Space」キーだが、行頭の空白は「Tab」キーを利用する。また、「Msgbox」などプログラムの命令は大文字でも小文字でも大丈夫で、小文字ですべて入力しても自動的に適切な表記に変換される。

### 2-2. プログラムの実行と終了

- (1)実行：VBE のツールバーのボタン  をクリックする。
- (2)停止：VBE のツールバーのボタン  をクリックする。また、プログラムを強制終了したい場合は「Esc キー」を押すか、「Ctrl+Pause キー」を押す。

### 2-3. プログラムの修正

例えば、「Msgbox」を「Magbox」と間違えて入力して実行した場合は、図 7 のようなメッセージが表示され、間違えた部分を知らせてくれる。この場合は、「OK」ボタンを押して、青く反転されている箇所を修正すればよい。

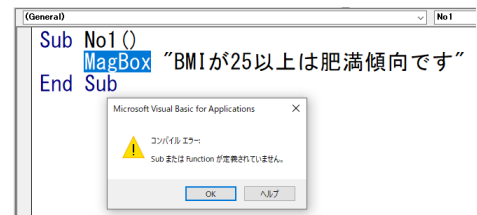


図 7

### 2-4. プログラムの保存

ユーザー定義関数のところで説明したように、プログラムを含む Excel のファイルを保存するときは、通常の Excel ブック「.xlsx」形式ではなく、Excel マクロ有効ブック「.xlsm」形式で保存する。

### 2-5. セキュリティの確認

アプリケーションソフトウェア上でプログラムが利用できると、作業の効率化や様々な処理が行えるなどメリットがある一方で、コンピュータウイルスの原因となったりするなど負の面がある。このため、Excel ではプログラムの利用に関して、全く使用できないようにしたり、制限付きで使えるようにしたりと、セキュリティのレベルを設定できるようになっている。その設定は以下の手順で確認できる。

- (1)「ファイル」タブ、[オプション] をクリックする。
- (2)「Excel のオプション」の窓が表示されるので、左側にある「トラストセンター」(Excel のバージョンによってはセキュリティセンター) をクリックし、その中の「トラストセンターの設定」をクリックする。
- (3)「マクロの設定」のところで「警告を表示してすべてのマクロを無効にする」にチェックが付いていることを確認する。

(補足)

「警告を表示せずにすべてのマクロを無効にする」にすると、セキュリティが最も高くなる。プログラムを利用しないときはこの設定が安全であるが、今回はプログラムを利用するので警告を発して、有効化できる設定にする。

### 3. 変数

プログラムでは数値や文字などのデータ扱うので、それらを格納するための記憶領域が必要となる。これを変数と呼ぶ。変数は、数値や文字を格納する箱のようなものと考えればよく、表計算のセルと似ている。プログラミング言語によっては、変数を使うためには最初に変数を宣言しておく必要があり、そのときにデータの種類に応じて変数の型も指定する。

(補足)

VBA では変数宣言をしなくても変数を利用できるが、ここでは変数の理解を深めるため変数宣言を行うようにする。なお、変数宣言しない変数は Variant 型というほぼすべてのデータ形式に対応できる型として扱われる。

#### 3-1. 変数の宣言

変数の型には、基本的なものとして、整数型、浮動小数点型、文字型の 3 つがある。例として整数型の変数を宣言してみる。

```
Dim x As Integer
```

これは、変数 x を整数型として宣言している。浮動小数点型は

```
Dim x As Double
```

文字型は

```
Dim x As String
```

となる。x は変数名なので、y でも z でも xy でもよい。変数名の長さは 255 文字以内である。変数名には英数字と \_ (アンダースコア) が使用できるが、変数名の先頭はアルファベットから始める。すなわち、1x や \_x などといった変数名は使えない。また、変数が複数ある場合は、

```
Dim x As String, y As Integer
```

のように、Dim 内に「,」で区切って記述することができる。次のプログラムは文字型変数を使った例である。No\_1\_1 の最後の End Sub で改行して、その下側に入力する。

例 2

```
Sub No2()  
    Dim x As String  
    x = "BMI が 25 以上は肥満傾向です"  
    MsgBox x  
End Sub
```

「=」は代入演算子と呼ばれ、右辺の値を左辺に代入する役割をもつ。演算子とは演算の種類を表す記号であり、ここでは左辺の x という文字型変数に右辺の文字列を代入している。最後に x に格納された文字列を「Msgbox」で表示している。次は、整数型変数を使ったプログラムである。

例 3

```
Sub No3()  
    Dim x As Integer, y As Integer  
    x = 20: y = 10  
    MsgBox x & "+" & y & "=" & x + y  
    MsgBox x & "-" & y & "=" & x - y  
End Sub
```

通常 1 行には 1 つの命令しか書けないが、「:」で区切ると、複数の命令を記述することができる。また「&」は文字列を結合する演算子である。文字列演算子以外にも算術演算子、比較演算子などがある。算術演算子には次のようなものがある。

表 1

演算子	例	意味	優先順位
+	$x + y$	$x$ と $y$ を加算する	5
-	$x - y$	$x$ を $y$ で減算する	5
*	$x * y$	$x$ と $y$ を乗算する	2
/	$x / y$	$x$ を $y$ で除算する	2
¥	$x ¥ y$	$x$ を $y$ で除算して商を求める	3
Mod	$x \text{ Mod } y$	$x$ を $y$ で除算して余りを求める	4
^	$x ^ y$	$x$ を $y$ 乗する	1

演算子は計算のときに優先順位があるので、それを変更したい場合は()を付ける。例えば、 $x = 1 + 2 * 3$  において、 $1 + 2$  を先に計算したいときは、 $x = (1 + 2) * 3$  とする。

### 3-2. 配列

配列は、同じデータ型(整数型、浮動小数点型、文字列型など)をもつ変数をまとめて扱う場合に使用する。変数がデータを格納する 1 つの箱だとすれば、配列は複数の箱から構成される。配列を用いたプログラムを以下に示す。

例 4

```
Sub No4()
    Dim x(2) As Double, goukei As Double
    x(0) = 40: x(1) = 30: x(2) = 70
    goukei = x(0) + x(1) + x(2)
    MsgBox "合計点=" & goukei
End Sub
```

最初の宣言の部分  $x(2)$  は、実数型のデータを 3 個格納する配列であることを示しており、配列の要素はそれぞれ、 $x(0)$ ,  $x(1)$ ,  $x(2)$  で参照できる。

## 4. 繰り返し処理

繰り返し処理とは、ある条件が満たされるまで指定された部分の処理を反復して行い続けることである。Visual Basic では、この繰り返し処理の命令として、For 文や Do while 文がある。

### 4-1. For 文

For 文は、指定された回数だけ処理を繰り返す命令で、以下のように記述する。

```
For i = m To n Step q
    処理
Next
```

この命令では、変数  $i$  を  $m$  から  $n$  まで  $q$  ずつ変化させながら処理を繰り返している。1 ずつ変化させる場合には、Step  $q$  の部分を省略できる。この For 文を使って、簡単なプログラムを作成してみる。

例 5

```
Sub No5()
    Dim x(2) As Double, goukei As Double, i As Integer
    x(0) = 40: x(1) = 30: x(2) = 70
    goukei = 0
    For i = 0 To 2
```

```

        goukei = goukei + x(i)
    Next i
    MsgBox "合計点=" & goukei
End Sub

```

このプログラムでは、For 文を使用して 3 回繰り返し処理を行っている。まず、For 文に入る前に  $\text{goukei} = 0$  により  $\text{goukei}$  に 0 を代入して初期化している。最初の For 文の処理では  $i=0$  であり、 $\text{goukei} = \text{goukei} + x(0)$  が実行される。このとき、変数  $\text{goukei}$  には 0 が代入されているので、右辺の値は  $0+x(0)$  となり、これが左辺の  $\text{goukei}$  に代入される。2 回目の処理では、 $i$  の値が 1 つ増えているので  $i=1$  となり、 $\text{goukei} = \text{goukei} + x(1)$  が実行される。このとき  $\text{goukei}$  には 1 回目の処理で  $x(0)$  が代入されているので、右辺の値は  $x(0)+x(1)$  となり、これが左辺の  $\text{goukei}$  に代入される。3 回目の処理が終わったときには、 $\text{goukei}$  は  $x(0)+x(1)+x(2)$  となるので、最後にこれを MsgBox で表示している。

#### 4-2. Do while 文

Do while 文は、ある条件式が満たされるまで処理を繰り返す命令で、次のように記述する。

```

Do While 条件式
    処理
Loop

```

この命令では、条件式が真である間、処理を繰り返す。Do While 文を使って、簡単なプログラムを作成してみる。

##### 例 6

```

Sub No6()
    Dim x(3) As Double, goukei As Double, i As Integer
    x(0) = 40: x(1) = 30: x(2) = 70
    goukei = 0
    i = 0
    Do While i <= 2
        goukei = goukei + x(i)
        i = i + 1
    Loop
    MsgBox "合計点=" & goukei
End Sub

```

変数  $i$  を  $i=0$  で初期化して、Do While 文中で  $i=i+1$  により、処理が繰り返されるたびに  $i$  を 1 ずつ増加させている。 $i \leq 2$  により、 $i$  が 3 となったときに処理は終了する。 $\leq$  は比較演算子で、他に以下のようなものがある。

表 2

演算子	例	意味
=	$i = 20$	$i$ は 20 と等しい
<>	$i <> 20$	$i$ は 20 と等しくない
<	$i < 20$	$i$ は 20 より小さい
>	$i > 20$	$i$ は 20 より大きい
<=	$i \leq 20$	$i$ は 20 以下
>=	$i \geq 20$	$i$ は 20 以上

## 5. 条件分岐

通常プログラムでは、1行ずつ書かれた命令が上から下へと処理される。一方、For 文や Do While 文などの繰り返し処理は、この原則とは異なった処理の流れを実現する。このような命令を制御命令という。その他の制御命令として、条件分岐がある。条件分岐の命令には、If 文や Select 文がある。

### 5-1. If 文

If 文は条件式によって処理を分岐させるための命令で、以下のように記述する。

```
If 条件式 Then
    処理
End If
```

この命令は、条件式が真の場合に処理を実行する。もし、条件式が偽の場合でも何らかの処理を実行したい場合には、次のように記述する。

```
If 条件式 Then
    処理 1
Else
    処理 2
End If
```

この命令は、条件式が真の場合に処理 1 を偽の場合には処理 2 を実行する。条件式を複数にして分岐を増やすこともできる。

```
If 条件式 1 Then
    処理 1
ElseIf 条件式 2 Then
    処理 2
Else
    処理 3
End If
```

この命令は、条件式 1 が満たされた場合には処理 1 を、それ以外で条件式 2 が満たされた場合には処理 2 を、それ以外では処理 3 を実行する。条件式がさらに多い場合には、ElseIf を追加する。If 文を使って、簡単なプログラムを作成してみる。

例 7

```
Sub No7()
    Dim x As Integer
    x = InputBox("年齢を入力して下さい。")
    If x >= 20 Then
        MsgBox "成人"
    ElseIf x >= 0 Then
        MsgBox "未成年"
    Else
        MsgBox "年齢の範囲が正しくありません"
    End If
End Sub
```

このプログラムは、自然数を入力したときにその大きさに応じて、画面にメッセージを表示する。Elseif x >= 0 Then の部分は、「前の条件 x >= 20 を満たしていない」かつ「x >= 0」という意味になるので、0 以上 20 未満という条件になる。



## 5-2. Select 文

Select 文は、変数の値によって条件を分岐する。

```
Select Case 変数
    Case 値 1
        処理 1
    Case 値 2
        処理 2
    Case Else
        処理 3
End Select
```

この命令は、変数の値が、値 1 に一致する場合は処理 1 を、それ以外で値 2 に一致する場合は処理 2 を、それ以外では処理 3 を実行する。Select 文を使って、簡単なプログラムを作成してみる。なお、Case のところで複数の値や値の範囲を指定したいときに、次のように書くこともできる。

表 3

書式	意味
Case 値 1, 値 2	値 1 または値 2 の場合
Case 値 1 to 値 2	値 1 から値 2 までの場合
Case is < 値 1	値 1 未満の場合
Case is <= 値 1	値 1 以下の場合
Case is >= 値 1	値 1 以上の場合
Case is > 値 1	値 1 より大きい場合

### 例 8

```
Sub No8()
    Dim x As Integer, y As Integer
    x = InputBox("整数を入力して下さい。")
    y = x Mod 2
    Select Case y
        Case 0
            MsgBox "偶数です。"
        Case Else
            MsgBox "奇数です。"
    End Select
End Sub
```

このプログラムは、入力された整数が偶数か奇数かを判別する。x Mod 2 より x を 2 で割ったときの余りを計算し、その結果が 0 であれば偶数、それ以外は奇数としている。

## 6. オブジェクトの操作

Excel は、ブック、ワークシート、セルなどの幾つかのオブジェクトから構成される。そのため、Visual Basic には、これらのオブジェクトを操作するための様々な命令がある。例えば、ワークシート Sheet1 にあるセル A1 に 100 の値を入力するための命令は、以下のように記述する。

### 例 9

```
Sub No9()
    Worksheets("Sheet1").Cells(1,1).Value = 100
End Sub
```

Cells(1,1)は Range("A1")でもよい。オブジェクトを操作する命令は、プロパティとメソッドに分類される。プロパティはオブジェクトの名前や色といった性質、特徴を設定するときに、メソッドはオブジェクトを削除したりアクティブにしたりといった動作をさせるときに使用される。

#### 6-1. セルに関する命令の例

命令	処理内容	利用例
Activate	セルをアクティブにする	Worksheets("Sheet1").Range("A1").Activate A1 セルをアクティブにする。
Select	セルを選択する	Worksheets("Sheet1").Range("A1:B3").Select (Select は、複数のセル範囲を選択できる。)
Range(セル番号)	セルを参照する	Range("A1:B3").Value = 9 A1:B3 の範囲のセルの値を 9 にする。
ActiveCell	アクティブなセルを参照する	ActiveCell.Font.Bold = True アクティブセルを太字にする。
Cells(行, 列)	1 つのセルを参照する	Worksheets("Sheet1").Cells(2,3).Select 2 行 3 列目のセル(C2 セル)を選択する。
Value	セルの値を取得/設定する	Worksheets("Sheet1").Range("A1").Value=100 A1 セルの値を 100 にする。

#### 6-2. ワークシートに関する命令の例

命令	処理内容	利用例
Activate	ワークシートをアクティブにする	Worksheets("Sheet1").Activate (Activate メソッドはセルでもワークシートでも使える。)
ActiveSheet	アクティブなワークシートを参照する	MsgBox ActiveSheet.Name アクティブなワークシート名を表示する。
Name	シートの名前を取得/設定する	Worksheets("Sheet1").Name="data" Sheet1 の名前を data にする。
Copy (before, after)	ワークシートをコピーする	Worksheets("s1").Copy before:= Worksheets("s2") s1 を s2 の前にコピーする。
Add	ワークシートを追加する	Worksheets.add
Delete	ワークシートを削除する	Worksheets("Sheet1").Delete Sheet1 を削除する。

#### 6-3. VBA でのワークシート関数の利用法

VBA で Excel のワークシート関数を利用する命令は、WorksheetFunction.関数名である。例えば、WorksheetFunction.Average(セル範囲)は、セル範囲のデータの平均値を計算する。同様に WorksheetFunction.Min(セル範囲)は最小値、WorksheetFunction.Max(セル範囲)は最大値をそれぞれ計算する。セル範囲を指定する場合は、Range(Cells(行, 列), Cells(行, 列))で表し、WorksheetFunction.Average(Range(Cells(1,1), Cells(5,1)))では、A1 セルから A5 セルまでに格納された 5 個の値の平均値を計算する。

#### 6-4. プログラムにおけるワークシートの利用

このプログラムは、「sheet1」シート(図 8)から、系列ごと(教科ごと)に得点を読み込んで、優、良、可、不可、範囲外の数を読み、その結果を「sheet2」シートに出力する(図 9)。範囲外を設けているのは、例えばテストを受けていない場合などを想定している。その場合は 0~100 以外の数字、例えば 999 等を入力する。

	A	B	C	D
1	国語	英語	数学	
2	55	70	54	
3	63	41	80	
4	90	76	83	
5	15	65	50	
6	21	80	30	
7	65	74	32	
8	92	36	50	
9	100	50	77	
10	20	5	34	
11	24	70	47	
12				
13				
14	成績集計		グラフ作成	
15				
16				
17				

図 8

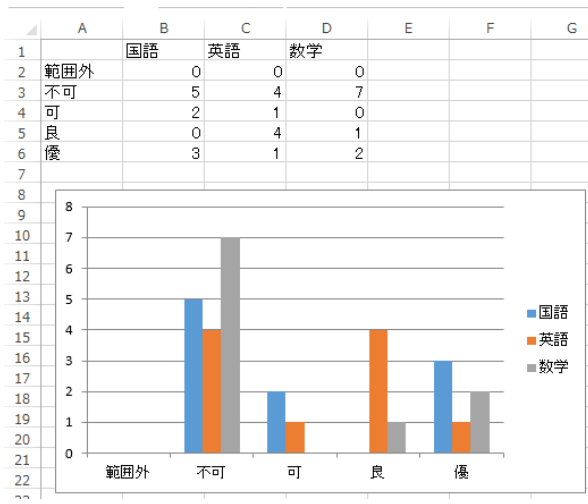


図 9

### 例 10

Sub No10()

Dim i As Integer, j As Integer, k As Integer, y(4) As Integer

Dim ws1 As Worksheet, ws2 As Worksheet

Set ws1 = ActiveWorkbook.Worksheets("sheet1")

Set ws2 = ActiveWorkbook.Worksheets("sheet2")

ws2.Cells(2, 1).Value = "範囲外": ws2.Cells(3, 1).Value = "不可"

ws2.Cells(4, 1).Value = "可": ws2.Cells(5, 1).Value = "良": ws2.Cells(6, 1).Value = "優"

j = 1

Do While j < 30

k = 0

Do Until IsEmpty(ws1.Cells(k + 2, j))

k = k + 1

Loop

If k = 0 Then

Exit Do

End If

Erase y

For i = 1 To k

If ws1.Cells(i + 1, j).Value < 0 Then

y(0) = y(0) + 1

ElseIf ws1.Cells(i + 1, j).Value < 60 Then

y(1) = y(1) + 1

ElseIf ws1.Cells(i + 1, j).Value < 70 Then

y(2) = y(2) + 1

ElseIf ws1.Cells(i + 1, j).Value < 80 Then

y(3) = y(3) + 1

ElseIf ws1.Cells(i + 1, j).Value <= 100 Then

y(4) = y(4) + 1

Else

y(0) = y(0) + 1

End If

Next

For i = 1 To 5

ws2.Cells(i + 1, j + 1).Value = y(i - 1)

Next

ws2.Cells(1, j + 1).Value = ws1.Cells(1, j).Value

```

    j = j + 1
Loop
End Sub

```

プログラムの説明

Do Until IsEmpty(セル番号): セルが空白になるまで処理を繰り返す。

Exit Do: Loop から抜ける。

Set ws1 = ActiveWorkbook.Worksheets("sheet1"): 変数 ws1 が現在アクティブなブックの「sheet1」シートを参照するように設定する。

Erase y: 配列 y を初期化する。

(2)プログラムの説明

Do While j < 30 は系列の繰り返し処理で 30 未満と仮定している。ここでの系列は各科目のことで j=1 で国語、j=2 で英語、j=3 で数学を参照する。得点が入力されていない列があった場合には Exit Do より Do While j ~ Loop を抜けるようにしている。得点が入力されている場合は配列 y に成績の分布を格納するようにして、その結果を「sheet2」シートへ出力する。

## 6-5. プログラムにおけるユーザー定義関数の利用

例 10 のプログラムにおいて、If 文で作成している成績を判定している部分を、1 つの独立した特定の処理とみたとき、それを関数として定義できる。関数を利用して例 10 を書き直したものが例 11 になる。

例 11

```

Sub No11()
Dim i As Integer, j As Integer, k As Integer, y(4) As Integer, p As Integer
Dim ws1 As Worksheet, ws2 As Worksheet, DataRange As Range
Set ws1 = ActiveWorkbook.Worksheets("sheet1")
Set ws2 = ActiveWorkbook.Worksheets("sheet2")
ws2.Cells(2, 1).Value = "範囲外": ws2.Cells(3, 1).Value = "不可"
ws2.Cells(4, 1).Value = "可": ws2.Cells(5, 1).Value = "良": ws2.Cells(6, 1).Value = "優"
j = 1
Do While j < 30
    k = 0
    Do Until IsEmpty(ws1.Cells(k + 2, j))
        k = k + 1
    Loop
    If k = 0 Then
        Exit Do
    End If
    Erase y
    For i = 1 To k
        p = getgrade(ws1.Cells(i + 1, j).Value)
        y(p) = y(p) + 1
    Next
    For i = 1 To 5
        ws2.Cells(i + 1, j + 1).Value = y(i - 1)
    Next
    ws2.Cells(1, j + 1).Value = ws1.Cells(1, j).Value
    j = j + 1
Loop
End Sub

```

```

Function getgrade(x As Double) As Integer
    If x < 0 Then
        getgrade = 0
    ElseIf x < 60 Then
        getgrade = 1
    ElseIf x < 70 Then
        getgrade = 2
    ElseIf x < 80 Then
        getgrade = 3
    ElseIf x <= 100 Then
        getgrade = 4
    Else
        getgrade = 0
    End If
End Function

```

#### プログラムの説明

getgrade という関数を作成し、入力された得点に応じて 0 から 4 までの整数を返すようにしている。例 11 の方が例 10 よりも処理の流れが見やすく、また、例えば、国語と数学をクロスさせて、国語が優で数学が良の人の数を数えるといった拡張も容易になる。クロスさせた場合のプログラムは例 12 となる。

#### 例 12

```

Sub No12()
    Dim i As Integer, j As Integer, k As Integer, p As Integer, q As Integer
    Dim r As Integer, M As Integer, N As Integer, z(4, 4) As Integer
    Dim ws1 As Worksheet, ws2 As Worksheet
    Set ws1 = ActiveWorkbook.Worksheets("sheet1")
    Set ws2 = ActiveWorkbook.Worksheets("sheet3")
    Set DataRange = ws1.Cells(1, 1).CurrentRegion
    N = DataRange.Columns.Count
    M = DataRange.Rows.Count - 1
    N = j - 1
    r = 0
    For j = 1 To N
        For i = j + 1 To N
            Erase z
            For k = 1 To M
                p = getgrade(ws1.Cells(k + 1, j).Value)
                q = getgrade(ws1.Cells(k + 1, i).Value)
                z(p, q) = z(p, q) + 1
            Next
            ws2.Cells(1 + r * 10, 1).Value = ws1.Cells(1, j).Value & "\" & ws1.Cells(1, i).Value
            ws2.Cells(2 + r * 10, 1).Value = "範囲外": ws2.Cells(1 + r * 10, 2).Value = "範囲外"
            ws2.Cells(3 + r * 10, 1).Value = "不可": ws2.Cells(1 + r * 10, 3).Value = "不可"
            ws2.Cells(4 + r * 10, 1).Value = "可": ws2.Cells(1 + r * 10, 4).Value = "可"
            ws2.Cells(5 + r * 10, 1).Value = "良": ws2.Cells(1 + r * 10, 5).Value = "良"
            ws2.Cells(6 + r * 10, 1).Value = "優": ws2.Cells(1 + r * 10, 6).Value = "優"
            For p = 1 To 5
                For q = 1 To 5
                    ws2.Cells(p + 1 + r * 10, q + 1).Value = z(p - 1, q - 1)
                Next
            Next
        Next
    Next

```

```

        r = r + 1
    Next
Next
End Sub

```

## プログラムの説明

Set DataRange = ws1.Cells(1, 1).CurrentRegion: ワークシート ws1 の A1 セルを含むデータが存在する領域のセル範囲を変数 DataRange に格納する。図 8 の例では A1:C11 のセル範囲になる。

N = DataRange.Columns.Count: DataRange で指定されたセル範囲の列数を数える。上記のセル範囲の場合は 3 となる。次の DataRange.Rows.Count は行数を数える。図 8 では 1 行目に見出しがあるので、数値データがある行の数はそれから 1 引いたものになり、M に代入している。

## (2)プログラムの説明

2つの系列のすべての組み合わせ、例えば国語と英語などを For j, For i のループにより参照する。まず、For j のループで j=1 により国語が指定され、次に For i のループで i=j+1 すなわち i=2 で英語が指定される。For k のループで成績分布を求める。p に j で指定された教科の成績(最初は国語)、q に i で指定された教科(最初は英語)の成績が格納されるようにする。k のループが終了すると i が更新され、i=3 により数学を指定する。国語\数学も同様に k のループにより成績分布が求められる。i のループが終了すると j が更新され、j=2 により英語を指定する。For i=j+1 となっているため、すべての組み合わせが終了している国語、及び、自分自身の英語は飛ばされ、数学を指定することになる。このように、for 文が入れ子構造になっているものを多重ループと呼ぶ。プログラムを実行すると、すべての系列の組み合わせの成績分布が出力される(図 10)。

	A	B	C	D	E	F
1	国語\英語	範囲外	不可	可	良	優
2	範囲外	0	0	0	0	0
3	不可	0	1	1	2	1
4	可	0	1	0	1	0
5	良	0	0	0	0	0
6	優	0	2	0	1	0
7						
8						
9						
10						
11	国語\数学	範囲外	不可	可	良	優
12	範囲外	0	0	0	0	0
13	不可	0	5	0	0	0
14	可	0	1	0	0	1
15	良	0	0	0	0	0
16	優	0	1	0	1	1
17						
18						
19						
20						
21	英語\数学	範囲外	不可	可	良	優
22	範囲外	0	0	0	0	0
23	不可	0	2	0	1	1
24	可	0	1	0	0	0
25	良	0	3	0	0	1
26	優	0	1	0	0	0

図 10

## 7. コマンドボタンによるプログラムの実行

### 7-1. ActiveX コントロール

プログラムの実行を、VBE からではなくワークシート上で行えると便利である。そのため、ActiveX コントロールを利用し、コマンドボタンをワークシート上に配置して、プログラムを記述してみる。

(1)リボンの「開発」タブの「コントロール」グループにある「挿入」ボタンをクリックし、表示されるメニューの中に「ActiveX コントロール」があることを確認する。

(2)「ActiveX コントロール」(図 11)において「コマンドボタン」をクリックし、ワークシート上でマウスをドラッグしボタンの大きさを指定し、ボタンを配置する。

(3)CommandButton1 という名前のボタンが表示されるので、そのボタンを選択した状態で右ボタンを押し、表示されるメニューの中から「プロパティ」をクリックする。

(4)プロパティの窓(図 12)が表示されるので、そこで「Caption」を探し、その右側に表示されている文字列「CommandButton1」をクリックし、その名前を「成績集計」に変更する(図 8)。最後に「プロパティ」の窓を閉じる。

(5)今度は、そのコマンドボタンに例 11 のプログラムをコピーする。そのために、コマンドボタンを選択した状態で右ボタンを押し、表示されるメニューの中から「コ

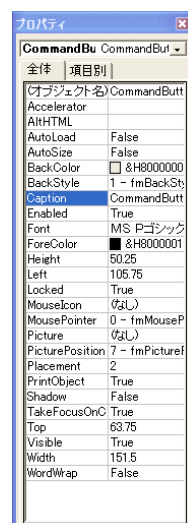
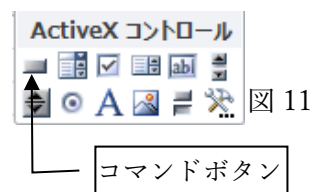


図 12

ードの表示」をクリックする。

(6)VBEが開くので、そこへコピーする。

(7)コピーが終えたら、リボンの「開発」タブの「コントロール」グループにある「デザインモード」ボタンをクリックする。

(8)最後に、コマンドボタンをクリックして、計算できるか確認する。

## 7-2. コマンドボタンによるグラフ描画

例 13 を参考にして、得られた結果をグラフ化してみること。また、コマンドボタンにより、グラフ描画を自動化してみること(図 9)。

例 13(番号は説明のために使うもので、入力しなくてよい。)

```
Sub No13()  
[1] Dim ws1 As Worksheet  
[2] Set ws1 = ActiveWorkbook.Worksheets("sheet2")  
[3] ws1.Select  
[4] ws1.ChartObjects.Add(10, 100, 350, 200).Select  
[5] ActiveChart.ChartType = xlColumnClustered  
[6] ActiveChart.SeriesCollection.NewSeries  
[7] ActiveChart.SeriesCollection(1).Values = ws1.Range("B2:B6")  
[8] ActiveChart.SeriesCollection(1).Name = ws1.Range("B1")  
[9] ActiveChart.SeriesCollection(1).XValues = ws1.Range("A2:A6")  
[10] ActiveChart.SeriesCollection.NewSeries  
[11] ActiveChart.SeriesCollection(2).Values = ws1.Range("C2:C6")  
[12] ActiveChart.SeriesCollection(2).Name = ws1.Range("C1")  
[13] ActiveChart.SeriesCollection(2).XValues = ws1.Range("A2:A6")  
[14] ActiveChart.SeriesCollection.NewSeries  
[15] ActiveChart.SeriesCollection(3).Values = ws1.Range("D2:D6")  
[16] ActiveChart.SeriesCollection(3).Name = ws1.Range("D1")  
[17] ActiveChart.SeriesCollection(3).XValues = ws1.Range("A2:A6")  
End Sub
```

### プログラムの説明

[1][2]. 変数 ws1 が「sheet2」シートを参照するように設定する。

[3]. 「sheet2」シートを選択する。

[4]. 「sheet2」シートにグラフを挿入する。挿入する場所と大きさは Add(Left, Top, Width, Height)で指定する。Left は左端からの位置、Top は上端からの位置、Width はグラフの幅、Height はグラフの高さを指定する。

[5]. グラフの種類を縦棒にする。

[6]. グラフに新しいデータ系列を追加する。SeriesCollection(1)は 1 番目の系列という意味。2 番目の系列を選択する場合は、SeriesCollection(2)とする。

[7]. 追加されたデータ系列の数値が格納されているセル範囲を「sheet2」シートの「B2:B6」とする。

[8]. 追加されたデータ系列の名前が格納されているセル番地を「sheet2」シートの「B1」とする。

[9]. 追加されたデータ系列の項目が格納されているセル番地を「sheet2」シートの「A2:A6」とする。

[10]-[13]. 2 番目の系列を追加し、その後は ActiveChart.SeriesCollection(2)として、2 番目の系列の折れ線を 1 番目の系列と同様に設定している。

[14]-[17] 3 番目の系列を追加し、その後は ActiveChart.SeriesCollection(3)として、3 番目の系列の折れ線を 1 番目、2 番目の系列と同様に設定している。

## 8. NDB オープンデータへの応用

図 13 は特定健診に関する NDB オープンデータの 1 つであり、都道府県、年齢ごとにクロス集計された BMI データである。また、図 14 は同じく収縮期血圧データである。BMI データでは、年齢の範囲を 40 歳から 74 歳までとし男女別に 5 歳刻みで 7 階層に分けており、検査値である BMI を 5 階層に分けている。収縮期血圧データでは、年齢の階層は BMI データと同じで、検査値である収縮期血圧を 6 階層に分けている。すなわち、BMI データ(図 13)と収縮期血圧データ(図 14)は、ほぼ同じフォーマットであり、異なるところは検査値の階層数だけである。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	特定健診(BMI):H29年度	※集計結果が10未満の場合は「J」で表示(10未満の箇所が1箇所の場合は10以上の最小値を全て「J」で表示)																
2	都道府県名	検査値階層 (kg/m <sup>2</sup> )	全体															
3			次															
4			男															
5			40～44歳	45～49歳	50～54歳	55～59歳	60～64歳	65～69歳	70～74歳	中計	40～44歳	45～49歳	50～54歳	55～59歳	60～64歳	65～69歳	70～74歳	中計
6			人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数
7	北海道	30.0以上	8,934	9,133	6,897	5,538	3,811	2,565	1,307	38,185	3,908	4,131	3,507	3,018	2,473	2,696	2,188	21,922
8		25.0以上30.0未満	28,755	31,129	27,842	27,636	24,229	20,705	13,368	173,664	9,648	11,106	10,688	10,613	10,328	13,091	11,308	76,782
9		20.0以上25.0未満	52,188	51,186	44,544	45,216	39,263	35,835	25,454	293,686	33,500	35,679	33,313	31,930	30,210	36,404	29,913	230,949
10		18.5以上20.0未満	6,156	5,481	4,516	4,438	3,843	3,406	2,362	30,202	13,845	12,595	11,337	9,892	8,073	8,182	6,041	70,065
11		18.5未満	2,533	2,132	1,777	1,880	1,769	1,531	1,105	12,717	8,192	8,040	7,725	7,207	5,645	5,518	4,104	47,431
12		30.0以上	2,441	2,208	1,579	1,338	871	678	434	3,449	1,174	1,242	1,053	982	911	969	777	7,108
13	青森県	25.0以上30.0未満	7,190	7,570	7,234	7,278	6,302	6,000	4,200	46,794	2,783	3,258	3,491	3,778	4,111	5,210	4,509	27,140
14		20.0以上25.0未満	12,323	12,763	11,795	12,557	11,087	11,173	8,419	80,117	8,719	9,412	9,640	10,195	10,421	12,702	10,259	71,348
15		18.5以上20.0未満	1,431	1,419	1,224	1,345	1,165	1,129	853	8,568	3,179	3,000	2,717	2,705	2,380	2,457	1,783	18,221
16		18.5未満	554	547	495	588	492	525	432	3,633	1,834	1,816	1,662	1,803	1,513	1,540	1,181	11,449
17	岩手県	30.0以上	2,329	2,240	1,705	1,394	906	734	472	9,780	1,161	1,180	1,031	1,016	902	1,074	862	7,226
18		25.0以上30.0未満	7,389	7,764	7,507	7,735	6,693	6,832	5,170	48,190	2,747	3,232	3,365	3,868	4,225	5,523	5,026	27,886
19		20.0以上25.0未満	13,293	12,760	12,281	13,212	11,827	12,306	9,804	85,583	9,100	9,557	9,296	10,203	10,601	13,528	11,279	73,564
20		18.5以上20.0未満	1,699	1,492	1,403	1,523	1,339	1,197	892	9,534	3,216	2,911	2,663	2,703	2,338	2,680	1,983	18,554
21		18.5未満	698	635	563	680	564	545	450	4,135	2,134	1,846	1,822	1,848	1,598	1,711	1,426	12,385

図 13

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	特定健診(収縮期血圧):H29年度	※集計結果が10未満の場合は「J」で表示(10未満の箇所が1箇所の場合は10以上の最小値を全て「J」で表示)																
2	都道府県名	検査値階層 (mmHg)	全体															
3			男															
4			40～44歳	45～49歳	50～54歳	55～59歳	60～64歳	65～69歳	70～74歳	中計	40～44歳	45～49歳	50～54歳	55～59歳	60～64歳	65～69歳	70～74歳	中計
5			人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数
6	北海道	180以上	276	395	470	651	747	735	463	3,737	131	239	287	315	376	482	472	2,302
7		160以上180未満	1,516	2,231	2,557	3,377	3,797	3,810	2,694	19,962	572	965	1,330	1,583	1,861	2,784	2,687	11,802
8		140以上160未満	8,969	11,713	12,868	15,817	16,149	15,998	11,313	92,827	2,903	4,829	6,354	7,353	8,791	12,822	11,696	54,738
9		120以上140未満	15,058	16,921	16,440	17,526	16,170	15,116	11,140	106,381	5,132	7,292	8,778	9,729	10,728	14,876	13,615	70,150
10		100以上120未満	28,066	27,459	23,093	21,862	18,052	15,038	10,023	143,593	12,791	14,962	14,520	14,921	14,230	16,739	13,579	102,161
11		120未満	44,679	40,315	30,147	25,476	17,998	13,345	7,963	179,923	48,555	43,220	34,895	28,851	20,744	18,187	11,513	205,965
12	青森県	180以上	70	85	104	141	139	132	117	788	37	44	56	65	84	109	84	479
13		160以上180未満	359	478	590	769	831	854	692	4,573	150	245	370	390	472	704	664	2,995
14		140以上160未満	2,199	2,890	3,173	4,100	4,022	4,280	3,359	24,023	788	1,260	1,661	2,244	2,709	3,904	3,839	16,405
15		120以上140未満	4,613	5,210	4,926	5,265	4,750	4,977	3,711	33,472	1,677	2,391	2,869	3,392	3,951	5,354	4,670	24,304
16		100以上120未満	6,564	6,551	6,039	5,775	4,890	4,546	3,367	37,672	3,268	3,811	4,254	4,509	4,700	5,637	4,584	30,742
17		120未満	10,133	9,293	7,495	6,325	5,244	4,716	3,110	47,025	11,969	10,976	9,353	8,864	7,419	7,171	4,691	60,343
18	岩手県	180以上	64	94	120	156	148	161	105	848	40	60	76	70	108	147	129	630
19		160以上180未満	430	610	653	950	938	1,038	797	5,416	188	295	390	476	637	920	895	3,791
20		140以上160未満	2,819	3,469	3,734	4,687	4,694	5,142	4,074	28,619	875	1,380	1,955	2,476	3,251	4,961	4,794	19,692
21		120以上140未満	4,325	4,723	4,839	5,347	4,825	5,051	4,012	33,122	1,649	2,311	2,671	3,392	3,925	5,365	5,054	24,367
22		100以上120未満	7,118	6,665	6,325	6,378	5,176	5,125	3,951	40,738	3,560	4,091	4,180	4,788	4,769	5,919	4,890	32,161
23		120未満	10,651	9,320	7,787	7,126	5,546	5,196	3,849	49,475	12,047	10,589	8,914	8,434	7,040	7,205	4,844	59,073

図 14

### 8-1. Excel の数式による NDB オープンデータの処理

ここでは、BMI データや収縮期血圧データにおいて、健康寿命に悪影響を及ぼすと考えられる検査値の範囲を設定し、その人数をリスク人口として算出する。例えば、BMI が 25 以上をリスクと考えたとき、北海道の男性の 40～44 歳を対象とすると、データは表 4 のようになっているため、リスク人口は、 $28755+8934=37689$  となる。

表 4

30.0以上	8,934
25.0以上30.0未満	28,755
20.0以上25.0未満	52,188
18.5以上20.0未満	6,156
18.5未満	2,533

また、リスク割合をリスク人口/全体の人口とすると  
 $(28755+8934)/(2533+6156+52188+28755+8934)=0.382373232\cdots$  となる。

NDB オープンデータが採用している年齢区分における日本の人口を図 15 に示す。このデータでは、40-44 歳の人口は 4914018 となっているが、これにさきほどのリスク割合をかけたものを調整リスク人口とすると、 $4914018 \times 0.382373232\cdots = 1878988.94\cdots$  となる。これを全年齢区分に適用した結果が図 16 となる。すべての年齢区分で調整リスク人口を合計したものを全調整リスク人口とし、全日本人人口も同様にすべての年齢区分で日本の人口を合計したものと定義とする。全調整リスク割合を下記 3 つのもので定義する。



全調整リスク割合(男) = 全調整リスク人口(男) / 全日本人人口(男)  
 全調整リスク割合(女) = 全調整リスク人口(女) / 全日本人人口(女)  
 全調整リスク割合(男女計) = 全調整リスク人口(男女計) / 全日本人人口(男女計)

この3つを計算したものを図 16 右側に示す。このような計算は Excel の数式で簡単に行うことができるが、都道府県ごとに同じことを何度も繰り返す必要がある。

## 8-2. VBA による NDB オープンデータの処理

この処理に VBA を利用すると、作業を効率的に行うことができる。BMI データに対して処理を VBA で記述したものが例 14\_3 である。ただし、例 14\_3 は少し複雑であるので、それを順番に説明していく。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1																		
2			40～44歳	45～49歳	50～54歳	55～59歳	60～64歳	65～69歳	70～74歳	中計	40～44歳	45～49歳	50～54歳	55～59歳	60～64歳	65～69歳	70～74歳	中計
3		日本人人口	4914018	4354877	3968311	3729523	4151119	4659662	3582440	29359950	4818200	4307927	3961985	3785723	4303891	4984205	4113371	30275302

図 15

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	特定健診(BMI)	H29年度 ※集計結果が10未満の場合は「-」で表示(10未満の箇所が1箇所の場合は10以上の最小値を全て「-」で表示)																				
2																						
3																						
4	都道府県名																					
5			40～44歳	45～49歳	50～54歳	55～59歳	60～64歳	65～69歳	70～74歳	中計	40～44歳	45～49歳	50～54歳	55～59歳	60～64歳	65～69歳	70～74歳	中計				
6			人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	全調整リスク人口			
7			人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	人数	男	女	計	
8	北海道	リスク割合	0.382373	0.406477	0.405943	0.391628	0.384557	0.363355	0.336613	0.386266	0.193412	0.212953	0.213234	0.217192	0.225652	0.239593	0.252007	0.220741				
9		日本人人口	4914018	4354877	3968311	3729523	4151119	4659662	3582440	29359950	4818200	4307927	3961985	3785723	4303891	4984205	4113371	30275302	29359950	30275302	59635252	
10		調整リスク人口	1879989	1770159	1610909	1460585	1596343	1693113	1205897	11340743	9318963	917396	8448307	8222306	9711807	1194179	1036600	6662992	11215966	67183027	17934298	
11																			0.3820169	0.221907	0.3007532	全調整リスク割合
12	青森県	リスク割合	0.402314	0.396988	0.394724	0.370148	0.360127	0.342374	0.32414	0.374374	0.222441	0.240282	0.244788	0.244567	0.259723	0.270085	0.285591	0.25319				
13		日本人人口	4914018	4354877	3968311	3729523	4151119	4659662	3582440	29359950	4818200	4307927	3961985	3785723	4303891	4984205	4113371	30275302	29359950	30275302	59635252	
14		調整リスク人口	1976979	1737544	1566387	1380476	1494928	1595346	1161212	10991602	1071764	1035117	9696465	9259815	1117819	1346158	1174741	7665404	10912872	76413069	18554179	
15																			0.3716925	0.2523941	0.3111277	全調整リスク割合

図 16

まず、2-1 節を参考にして VBA を入力するためにエディタを起動する。次に、標準モジュールに例 14\_1 を入力し、results シートに文字列を表示してみる。結果は図 17 のようになる。

### 例 14\_1

```
Sub BMI_1()
[1] Dim i As Integer, j As Integer
[2] Dim total As Double, risk As Double
[3] Dim sum As Double, sum_m As Double, sum_f As Double
[4] Dim sum_risk As Double, sum_risk_m As Double, sum_risk_f As Double
[5] Dim risk_p(15) As Double, population(15) As Double
[6] Dim ws1 As Worksheet, ws2 As Worksheet
[7] Set ws1 = ActiveWorkbook.Sheets("BMI")
[8] Set ws2 = ActiveWorkbook.Sheets("results")
[9] ws2.Select
[10]ws2.Cells(1, 2) = "BMI"
[11]ws2.Cells(2, 2) = "男"
[12]ws2.Cells(2, 3) = "女"
[13]ws2.Cells(2, 4) = "計"
End Sub
```

### プログラムの説明

- [1]-[6]. 変数宣言を行う。(3-1 節を参照)
- [7]. 変数 ws1 が「BMI」シートを参照するように設定する。(6-4 節を参照)
- [8]. 変数 ws2 が「results」シートを参照するように設定する。(6-4 節を参照)
- [9]. 「results」シートを選択する。(6-1 節を参照)
- [10]-[13]. 「results」シートの指定した場所に文字列を代入する。(6 章例 9 を参照)

	A	B	C	D
1		BMI		
2		男	女	計
3				

図 17

次に、例 14\_2 を入力し、results シートに都道府県名を表示してみる。結果は図 18 のようになる。

#### 例 14\_2

```
Sub BMI_2()  
[1] Dim i As Integer, j As Integer  
[2] Dim total As Double, risk As Double  
[3] Dim sum As Double, sum_m As Double, sum_f As Double  
[4] Dim sum_risk As Double, sum_risk_m As Double, sum_risk_f As Double  
[5] Dim risk_p(15) As Double, population(15) As Double  
[6] Dim ws1 As Worksheet, ws2 As Worksheet  
[7] Set ws1 = ActiveWorkbook.Sheets("BMI")  
[8] Set ws2 = ActiveWorkbook.Sheets("results")  
[9] ws2.Select  
[10]ws2.Cells(1, 2) = "BMI"  
[11]ws2.Cells(2, 2) = "男"  
[12]ws2.Cells(2, 3) = "女"  
[13]ws2.Cells(2, 4) = "計"  
[14]For j = 1 To 47  
[15]  ws2.Cells(j + 2, 1) = ws1.Cells(5 * (j - 1) + 6, 1)  
[16]Next  
End Sub
```

#### プログラムの説明

[14]-[16]. 都道府県の繰り返し処理。(4-1 節を参照)

[15]. 「BMI」シートに格納されている都道府県の文字列を「results」シートに代入する。「BMI」シートでは北海道は A6、青森は A11、岩手は A16 と 5 個飛ばしになっているので ws1.Cells(5 \* (j - 1) + 6, 1) と指定している。(6-1 節を参照)

	A	B	C	D
1		BMI		
2		男	女	計
3	北海道			
4	青森県			
5	岩手県			
6	宮城県			
7	秋田県			
8	山形県			
9	福島県			
10	茨城県			
11	栃木県			
12	群馬県			
13	埼玉県			
14	千葉県			
15	東京都			
16	神奈川県			
17	新潟県			

図 18

最後に例 14\_3 を入力すると、図 20 の結果を得る。

#### 例 14\_3

```
Sub BMI_3()  
[1] Dim i As Integer, j As Integer  
[2] Dim total As Double, risk As Double  
[3] Dim sum As Double, sum_m As Double, sum_f As Double  
[4] Dim sum_risk As Double, sum_risk_m As Double, sum_risk_f As Double  
[5] Dim risk_p(15) As Double, population(15) As Double  
[6] Dim ws1 As Worksheet, ws2 As Worksheet  
[7] Set ws1 = ActiveWorkbook.Sheets("BMI")  
[8] Set ws2 = ActiveWorkbook.Sheets("results")  
[9] ws2.Select  
[10]ws2.Cells(1, 2) = "BMI"
```

```

[11]ws2.Cells(2, 2) = "男"
[12]ws2.Cells(2, 3) = "女"
[13]ws2.Cells(2, 4) = "計"
[14]For j = 1 To 47
[15]  ws2.Cells(j + 2, 1) = ws1.Cells(5 * (j - 1) + 6, 1)
[16]  For i = 1 To 16
[17]    total=WorksheetFunction.Sum(Range(ws1.Cells(5*(j-1)+6, i+2), ws1.Cells(5*(j-1)+6+4, i+2)))
[18]    risk = WorksheetFunction.Sum(Range(ws1.Cells(5*(j-1)+6, i+2), ws1.Cells(5*(j-1)+6+1, i+2)))
[19]    population(i - 1) = Worksheets("basic_data").Cells(3, i + 2)
[20]    risk_p(i - 1) = risk / total * population(i - 1)
[21]  Next
[22]  sum = 0: sum_m = 0: sum_f = 0
[23]  sum_risk = 0: sum_risk_m = 0: sum_risk_f = 0
[24]  For i = 0 To 6
[25]    sum = sum + population(i) + population(i + 8)
[26]    sum_risk = sum_risk + risk_p(i) + risk_p(i + 8)
[27]    sum_m = sum_m + population(i)
[28]    sum_risk_m = sum_risk_m + risk_p(i)
[29]    sum_f = sum_f + population(i + 8)
[30]    sum_risk_f = sum_risk_f + risk_p(i + 8)
[31]  Next
[32]  ws2.Cells(j + 2, 2) = sum_risk_m / sum_m
[33]  ws2.Cells(j + 2, 3) = sum_risk_f / sum_f
[34]  ws2.Cells(j + 2, 4) = sum_risk / sum
[35]Next
End Sub

```

#### プログラムの説明

[16]-[21]. 年齢の繰り返し処理。都道府県の繰り返し処理の中にあるので 2 重ループとなっている。(6-5 節例 12 を参照)

[17]. ある年齢幅における全体の人口を計算し、変数 total に代入している。WorksheetFunction.Sum は合計を計算するワークシート関数で、セル範囲は Range(ws1.Cells(5\*(j-1)+6, i+2), ws1.Cells(5\*(j-1)+6+4, i+2)) により指定している。

	A	B	C
1	特定健診(BMI):H29年度 ※集計結果が10未満の場合		
2			
3			
4	都道府県名	検査値階層 (kg/m <sup>2</sup> )	40～44歳
5			人数
6		30.0以上	8,934
7		25.0以上30.0未満	28,755
8	北海道	20.0以上25.0未満	52,188
9		18.5以上20.0未満	6,156
10		18.5未満	2,533
11		30.0以上	2,441
12		25.0以上30.0未満	7,190
13	青森県	20.0以上25.0未満	12,323
14		18.5以上20.0未満	1,431
15		18.5未満	554

図 19

Range 内の cells 命令で指定された座標は、図 19 の通り NBA データの規則性に基づいている。(6-3 節を参照)

[18]. リスク人口を計算し、変数 risk に代入している。Range(ws1.Cells(5\*(j-1)+6, i+2), ws1.Cells(5\*(j-1)+6+1, i+2))により、セル範囲を指定している。total の計算と違うところは、終点を表す ws1.Cells(5\*(j-1)+6+1, i+2)のところで、30.0 以上の下の 25.0 以上 30.0 未満の人数を指定している。

[19]. 「basic\_data」シートに格納されている各年齢区分の日本の人口データを配列 population に代入している。(3-2 節を参照)

[20]. リスク割合を計算し、変数 risk\_p に代入している。(3-2 節を参照)

[22]-[23]. 変数を 0 で初期化している。(4-1 節例 5 を参照)

[24]-[31]. 合計の計算を行うための繰り返し処理。(4-1 節例 5 を参照)

[25]. 全日本人人口(男女計)を計算し、変数 sum に代入している。sum = sum + population(i)+population(i+8)は、最初の i=0 のとき population(0)の 40-44 歳の男性、population(8)の 40-44 歳の女性、i=1 のとき population(1)の 45-49 歳の男性、population(9)の 45-49 歳の女性を sum に足していき、最後 i=6 のとき population(6)の 70-74 歳の男性、population(14)の 70-74 歳の女性を足している。(4-1 節例 5 を参照)

[26]. 全リスク人口(男女計)を計算し、変数 sum\_risk に代入している。(4-1 節例 5 を参照)

[27]. 全日本人人口(男)を計算し、変数 sum\_m に代入している。(4-1 節例 5 を参照)

[28]. 全リスク人口(男)を計算し、変数 sum\_risk\_m に代入している。(4-1 節例 5 を参照)

[29]. 全日本人人口(女)を計算し、変数 sum\_f に代入している。(4-1 節例 5 を参照)

[30]. 全リスク人口(女)を計算し、変数 sum\_risk\_f に代入している。(4-1 節例 5 を参照)

[32]. 全調整リスク割合(男)を計算し、results シートに代入している。(6-1 節を参照)

[33]. 全調整リスク割合(女)を計算し、results シートに代入している。(6-1 節を参照)

[34]. 全調整リスク割合(男女計)を計算し、results シートに代入している。(6-1 節を参照)

同様に収縮期血圧データに対して処理を VBA で記述したものが例 15 である。

#### 例 15

```
Sub systolic_blood_pressure()
[1] Dim i As Integer, j As Integer
[2] Dim total As Double, risk As Double
[3] Dim sum As Double, sum_m As Double, sum_f As Double
[4] Dim sum_risk As Double, sum_risk_m As Double, sum_risk_f As Double
[5] Dim risk_p(15) As Double, population(15) As Double
[6] Dim ws1 As Worksheet, ws2 As Worksheet
[7] Set ws1 = ActiveWorkbook.Sheets("収縮期血圧")
[8] Set ws2 = ActiveWorkbook.Sheets("results")
[9] ws2.Select
[10]ws2.Cells(1, 2 + 3) = "収縮期血圧"
[11]ws2.Cells(2, 2 + 3) = "男"
[12]ws2.Cells(2, 3 + 3) = "女"
[13]ws2.Cells(2, 4 + 3) = "計"
[14]For j = 1 To 47
[15]  For i = 1 To 16
[16]    total = WorksheetFunction.Sum(Range(ws1.Cells(6*(j-1)+6, i+2), ws1.Cells(6*(j-1)+6+5, i+2)))
[17]    risk = WorksheetFunction.Sum(Range(ws1.Cells(6*(j-1)+6, i+2), ws1.Cells(6*(j-1)+6+2, i+2)))
[18]    population(i - 1) = Worksheets("basic_data").Cells(3, i + 2)
[19]    risk_p(i - 1) = risk / total * population(i - 1)
[20]  Next
[21]  sum = 0: sum_m = 0: sum_f = 0
[22]  sum_risk = 0: sum_risk_m = 0: sum_risk_f = 0
```

```

[23] For i = 0 To 6
[24]     sum = sum + population(i) + population(i + 8)
[25]     sum_risk = sum_risk + risk_p(i) + risk_p(i + 8)
[26]     sum_m = sum_m + population(i)
[27]     sum_risk_m = sum_risk_m + risk_p(i)
[28]     sum_f = sum_f + population(i + 8)
[29]     sum_risk_f = sum_risk_f + risk_p(i + 8)
[30] Next
[31] ws2.Cells(j + 2, 2 + 3) = sum_risk_m / sum_m
[32] ws2.Cells(j + 2, 3 + 3) = sum_risk_f / sum_f
[33] ws2.Cells(j + 2, 4 + 3) = sum_risk / sum
[34] Next
End Sub

```

## プログラムの説明

ほぼ例 14\_3 と同じであるが、変数 ws1 が「収縮期血圧」シートを参照するように設定している。また、収縮期血圧は 6 階層になっているので、[16][17]の Range 命令で 6\*(j-1)等となっている。また、「results」シートに結果を出力させる[31]-[33]において、ws2.Cells(j + 2, 2 + 3)等で、+3 により BMI の結果の右側に出力させるようにしている。出力結果は図 21 となる。

	A	B	C	D	E	F	G
1		BMI					
2		男	女	計			
3	北海道	0.382017	0.221907	0.300733			
4	青森県	0.371692	0.252394	0.311128			
5	岩手県	0.371198	0.250283	0.309813			
6	宮城県	0.369857	0.232416	0.300082			
7	秋田県	0.363783	0.245596	0.303782			
8	山形県	0.342412	0.232926	0.286829			
9	福島県	0.370905	0.252422	0.310754			
10	茨城県	0.355955	0.225372	0.289662			
11	栃木県	0.346791	0.226218	0.285579			
12	群馬県	0.330579	0.213484	0.271133			
13	埼玉県	0.339293	0.20493	0.27108			
14	千葉県	0.351573	0.205232	0.277279			
15	東京都	0.333197	0.182379	0.256631			
16	神奈川県	0.330205	0.181614	0.25477			
17	新潟県	0.302395	0.191435	0.246063			
18	富山県	0.328332	0.194207	0.26024			
19	石川県	0.324268	0.192147	0.257194			
20	福井県	0.325296	0.192762	0.258012			

図 20

	A	B	C	D	E	F	G
1		BMI			収縮期血圧		
2		男	女	計	男	女	計
3	北海道	0.382017	0.221907	0.300733	0.226146	0.159827	0.192478
4	青森県	0.371692	0.252394	0.311128	0.20406	0.144158	0.17365
5	岩手県	0.371198	0.250283	0.309813	0.223364	0.167043	0.194771
6	宮城県	0.369857	0.232416	0.300082	0.215368	0.154752	0.184595
7	秋田県	0.363783	0.245596	0.303782	0.255082	0.179567	0.216745
8	山形県	0.342412	0.232926	0.286829	0.231462	0.166328	0.198395
9	福島県	0.370905	0.252422	0.310754	0.211106	0.156514	0.183391
10	茨城県	0.355955	0.225372	0.289662	0.199933	0.147842	0.173488
11	栃木県	0.346791	0.226218	0.285579	0.219492	0.162115	0.190363
12	群馬県	0.330579	0.213484	0.271133	0.219932	0.168714	0.19393
13	埼玉県	0.339293	0.20493	0.27108	0.207666	0.156416	0.181648
14	千葉県	0.351573	0.205232	0.277279	0.194674	0.14887	0.17142
15	東京都	0.333197	0.182379	0.256631	0.181471	0.129362	0.155016
16	神奈川県	0.330205	0.181614	0.25477	0.18297	0.133549	0.15788
17	新潟県	0.302395	0.191435	0.246063	0.203687	0.138851	0.170771
18	富山県	0.328332	0.194207	0.26024	0.197809	0.145402	0.171204
19	石川県	0.324268	0.192147	0.257194	0.213053	0.151387	0.181747
20	福井県	0.325296	0.192762	0.258012	0.225602	0.162873	0.193756

図 21