A。 Windows 版 SAS

A.1 Windows 版 SAS の構成

Windows 版の SAS はプロダクトと呼ばれるプログラムの集合から構成されている．もちろん，SAS を利用するために，これらのプロダクトすべてが必要になるわけではない．各自の利用する目的によって，「Base SAS」という基本プロダクトに，必要なプロダクトを組み合せて利用することになる．本書で必要とするプロダクトを表 A.1 に示す．

表 A.1 主なプロダクトとその組み込みに必要なディスク容量（Ver. 9.3）

プロダクト名	プロダクトの内容	必要ディスク容量（MB）
Base SAS	SAS 基本システム	601
SAS/STAT	統計解析	70
SAS/GRAPH	グラフ作成＋地図データなど	390＋1.3 GB
ALS（日本語）	日本語環境サポート	274

A.1.1 必 要 な 環 境

「Windows 版 SAS」をインストールするにあたって，ある程度快適に利用するためには表 A.2 に示したような環境をあらかじめそろえる必要がある．もちろん，簡単な処理が中心であれば，CPU 処理速度，ハードディスクやメモリの容量などはあまり必要ないが，逆に複雑な処理が多ければ，これでは足りないであろう（たとえば，標準的なプロダクトなどをインストールするとハードディスクは 5 GB 以上必要になる）．あくまで，実用上最低限の目安であるが，現在市販されているパソコンであれば，この程度のスペックはクリアしているはずなので心配はない．

「SAS」の最近のバージョンでは，必要なハードディスクの容量が大幅に増えているだけでなく，インストール時の制限もいろいろあるので注意が必要である．たとえば，インストールの前に Windows のアップデートを必ず行い，最新のサービスパックを

表 A.2 SAS（Ver. 9.3）を動かすために必要な環境

ソフトウェア（それぞれ，32 bit 版と 64 bit 版あり）
・Windows XP Professional
・Windows Vista Business または Ultimate
・Windows 7 Professional または Ultimate
・Windows 8 Pro
ハードウェア
・Pentium Ⅳ 1 GHz 以上の CPU をもつパソコン（2 つ以上の CPU が望ましい）
・最低 512 MB 以上のメモリ（Vista 以降は最低 1 GB 以上）
・1 GB 以上の空き領域があるハードディスク （インストール後にメモリと同量のスワップファイル領域が必要なため）
・なるべく高解像度（1024×768 以上）の表示できるディスプレイ
・インストール用の DVD ドライブ

組み込んでおく必要がある．また，Java の Runtime Environment も最新版にアップデートしておいたほうがよい．プログラムのインストールにはアドミニストレータの権限が必要であるが，その名前が日本語（2 バイト文字）の場合には，うまくインストールされない．なお，Windows 8 の「Pro」以外には正式対応していないが，学生用として「Education Analytical Suite」というパッケージが販売されており，このパッケージのみ「Pro」以外もサポートされている．

SAS には，32 ビット版と 64 ビット版の 2 種類があるが，64 ビット版はアプリケーションとして未完成な部分があり，Excel ファイルの読込みや出力など他のアプリケーションと連携をする部分で特殊な設定などが必要になる．通常のデータ数であれば処理速度も変らないので，Excel データを頻繁に利用する場合には，32 ビット版の利用をお勧めする．

A.2 SAS プログラムの構造

まず，いままで SAS のプログラムを作成したことがないという方のために，簡単なプログラム例を使って，プログラムの構造や注意点などを紹介しておこう．「プログラムなど考えたことがない」という人でも，コツさえ覚えれば意外に単純なので安心してほしい．

A.2.1 データステップとプロックステップ

SAS では，プログラム全体が，データの読込みと加工を行うデータステップと統計解析などを行うプロックステップの 2 つの部分に大別できる．

データステップは「DATA ステートメント」で始まり，「RUN ステートメント」で

終わる一連のプログラムの集まりである．通常はその間に，データが納められているファイル名を指定する「INFILE ステートメント」や変数の並びを指定する「INPUT ステートメント」，また各種のデータ変容を行う命令などの指定が可能である．たとえば，以下のような命令の集合がデータステップとなる．

この例では，C ドライブの「data」というディレクトリから「sample.dat」というデータを id，name などの変数名で読み込み，「data1」というデータ名としている．

```
DATA data1;
 INFILE "c:¥data¥sample.dat" MISSOVER;
 INPUT id name $ age 15-16 sex height weight dat1-dat5;
RUN;
```

「DATA ステートメント」の次にはプログラムの中で利用するデータ名を指定する．データ名は標準で半角 8 文字まで使えるので，なるべく後からわかりやすい（日本語ではない）名前をつけたほうがよい．

プロックステップは「PROC ステートメント」で始まり「RUN ステートメント」で終わる一連のプログラムの集まりである．通常は次の例のように「PROC ステートメント」とともに，行う処理の名称と使うデータ名，オプションなどを指定する．さらに，「VAR ステートメント」などで処理に使う変数名やその他の細かな設定を指定する．

ここでは，「data1」というデータに対して，「id と age という変数を印刷しろ」という指定を行ってみよう．

```
PROC PRINT DATA=data1;
VAR id age;
RUN;
```

新しいステップが始まることが明らかであれば，「RUN ステートメント」は省略できるが，間違えやすいので最初のうちは必ずつけるような習慣にしておいたほうがよい．

なお，各ステートメントの最後には「;」を必ずつける．「;」がない場合は，ステートメントが論理的につながっていると判断されるので，トラブルの原因となる．逆に「;」をつければそこで論理的な 1 行が終了するので，「PROC PRINT;RUN;」などのように指定して，1 行に複数のステートメントを指定することもできる．

A.2.2 データの読み込み

a. データの形式　統計に使うデータの基本的な形は，データ例に示したように，ケースごとに横に変数値が並び，各変数値間に半角スペースが 1 つ以上（変数内はスペース不可），欠損値は「.（半角ピリオド）」である．表計算ソフトでも，テキスト形式で保存した場合には，ほぼこのような形式になるので，欠損値がない場合には，そのまま利用できる．ただし，表計算ソフトの欠損値は，テキスト出力の場合，通常スペースになってしまうので注意する．変数間をカンマで区切った CSV 形式などもオプションを指定すれば読み込めるし，必要ならば行や桁を指定して読み込むこともできるので，うまく利用してほしい．なおデータには，トラブルを防ぐために，全角文字を避けて半角の英数字を使うようにしたほうがよい．

［データ例］

```
101 TANAKA    24 1 173.2 74.5 1 2 1 0 2
102 FUKUDA    35 1 165.2 80.3 1 1 2 0 2
103 MAEDA     53 1 168.1 74.2 0 9 2 1 1
104 KOBAYASHI  . 1 170.6 59.3 1 2 2 1 2
105 TAKEDA    29 2 162.5 48.5 0 2 1 0 2
106 HOSHINO   26 2 165.1 49.1 1 1 2 1 2
107 ASADA     23 2 157.3 41.5 9 1 9 1 9
108 IKEDA     25 2 153.5 59.7 1 1 2 1 1
109 OGAWA     21 . 167.3 55.2 0 1 2 1 1
```

b. データファイルの指定方法　データの入っているファイル名の指定には，「INFILE ステートメント」を使う．たとえば，「C:」ドライブの「data」というディレクトリの中に，「sample.dat」という名前のファイルとしてあるならば，

INFILE "c:¥data¥sample.dat" MISSOVER;

のように，ファイル名をディレクトリ名を含めて" "でくくって指定し，必要ならオプションを指定する（ライブラリを指定して読み込む方法もあるが，最初はこの方法のほうが直接的でわかりやすいだろう）．よく使うオプションとしては，例で示した「MISSOVER」の他にも以下のものがあるので，状況に応じて利用してほしい．

LRECL＝300　データの 1 行が長い場合に指定する．この例では 1 行が 300 桁あるとして読み込む．

MISSOVER　指定した変数の数より 1 行当りの変数の数が少ない場合に，次の行の変数を読み込まず欠損値とする．通常は，足りない変数を次の行に探しにいくので，1 ケース 1 行のデータの場合には必ず指定するとよい．

DLM＝",;"　半角スペース以外の文字で変数が区切られているデータを読み込

む場合に指定する．この例では，カンマまたはセミコロンで変数が区切られているものとしてデータを読み込む．

OBS＝20　データの一部だけを読み込むのに使う．この例では，最初から 20 ケースだけ読み込むという指定である．逆に，最初の数例を読み飛ばすための「FIRSTOBS＝5」（5 例目から読み込む）などの指定もできる．

データ数が少ない場合などにはプログラムの中にデータを直接書き込んでしまう方法もある．データステップでは，「DATALINES;」と次の「;」に囲まれた部分をデータとして読み込むので，以下のようなプログラムになる．この場合「INFILE ステートメント」は省略できるが，オプションなどが必要なら「INFILE DATALINES OBS＝5;」などと指定することもできる．

```
DATA data1;
  INPUT id name $ age 15-16 sex height weight dat1-dat5;
DATALINES;
101 TANAKA    24 1 173.2 74.5 1 2 1 0 2
102 FUKUDA    35 1 165.2 80.3 1 1 2 0 2
103 MAEDA     53 1 168.1 74.2 0 9 2 1 1
104 KOBAYASHI  . 1 170.6 59.3 1 2 2 1 2
105 TAKEDA    29 2 162.5 48.5 0 2 1 0 2
106 HOSHINO   26 2 165.1 49.1 1 1 2 1 2
107 ASADA     23 2 157.3 41.5 9 1 9 1 9
108 IKEDA     25 2 153.5 59.7 1 1 2 1 1
109 OGAWA     21 . 167.3 55.2 0 1 2 1 1
;
RUN;
```

c. 変数名の指定方法　変数名の指定には，INPUT ステートメントを使う．最も簡単な指定方法は，INPUT ステートメントの後に変数名を列挙していく方法である．データが 1 ケース 2 行以上になっていたとしても，標準では，順に変数を読み込んでいき，1 行のデータで足らなくなった場合には，次の行のデータを次の変数の値として読み込んでいくので，データの行数を意識する必要はない．文字型の変数の場合は，次の例の「name」のように，変数名の後に「$」をつければ認識する．また，「age」のように，変数の後に「15-16」と指定して，読み込むカラム位置を指定することもできる．

INPUT id name $ age 15-16 sex height weight dat1-dat5;

変数名は，アルファベットから始まり，8 文字以内の英数字とする．「dat1-dat5」の

ように指定すると，dat1 から dat5 までの 5 つの変数を一度に指定できるので，プログラムが簡単になる．

この他にも，カラムコントロールポインタ（@3，+2 など）や行コントロールポインタ（#2 など）を使って，カラムや行を読み飛ばすこともできる．また，1 行に複数のケースが入力されている場合は，「INPUT ステートメント」の最後を「@@ ;」とすることで行継続入力命令とすることもできる．

d. データのインポート　テキスト形式（CSV 形式）のデータファイルを作成しておけば，「ファイルメニュー」の「データのインポート」を使って，プログラムを書かなくても簡単に読み込むことができる．さらに，「PC File Formats」というプロダクトを契約していると Excel（xls 形式および xlsx 形式）や Access などのデータに関しても，同様にして直接簡単に読み込むことができる．

ただし，通常 1 行目を変数名またはラベル名として読み込ませるので，あらかじめ半角英数字に変更して変数名をつけておく必要がある．日本語の変数名の場合には，読込み後，SAS に勝手に変数名を割り付けられるので，プログラム作成時の取扱いが大変になってしまうからである．

e. SAS データセットとライブラリ　データステップで作成したデータのことを，正式には「SAS データセット」と呼ぶ．その名称（データセット名）はライブラリ名とメンバ名からできているが，標準のライブラリ名は一時データセット用の「Work」と決まっているため，SAS プログラム内ではライブラリ名を省略してメンバ名だけで取り扱うことが多い．たとえば，データステップの最初に「DATA data1；」とすることで作成されるデータ「data1」の正式名称は「work.data1」であり，「DATA work.data1;」と指定するのと同じことである．なお，一時データセット用の「Work」の中のデータは，SAS 終了時に自動的に削除されるという特徴がある．

一方，データステップで利用するデータセット名を「sasuser.data1」などと「work」以外のライブラリ名を用いてフルネームで指定すれば，SAS プログラム終了後も「sasuser」というライブラリ（Windows の世界ではフォルダ）の中に「data1」というメンバ（同じくファイル）として残すことができる．一時データセットに対して，このようにして残したデータファイルを永久データセットと呼ぶ．複雑な加工をしたデータを永久データセットとして保存すれば，次回以降にデータステップによる加工をしなくても，データを直接読み込んで簡単に利用できるという利点がある．

ただし，永久データセット用の SAS ライブラリは，後述するように，あらかじめエクスプローラウィンドウで作成しておく必要がある（図 A.8 参照）．つまり，ライブラリ名と Windows のフォルダ名を同じにする必要がない代りに，あらかじめキチン

と関係づけておかなければならないということである.

A.2.3 データの加工

データを加工するためには，計算式や SAS 関数を使った「割当てステートメント（代入文）」による新しい変数の作成や，「IF-THEN/ELSE ステートメント」を使った条件文，「ARRAY ステートメント」と「DO-END ステートメント」の組合せによる複数変数に対する指定などさまざまな方法がある．これらのステートメントの使用例では，①～④の指定を行っている.

①「sex」が欠損値のケースを削除する．②「身長（height）」と「体重（weight）」から「bmi」を計算する．③その値から肥満かどうかを判定し，「himan」という変数に肥満なら 1，そうでなければ 0 を代入する．④「dat1」から「dat5」の 5 つの変数に関しては，値が 9 のときを欠損値とする.

a. データステップ例

```
DATA data1;
  INFILE "C:¥datasample.dat";
  INPUT id name $ age 15-16 sex height weight dat1-dat5;
  ① IF sex=. THEN DELETE;                    ←    ケースの削除
  ② bmi=(weight/height**2)*10000;            ←    新しい変数値の作成
  ③ IF bmi=. THEN himan=.;
      ELSE IF bmi>=25 THEN himan=1;          ←    条件によるリコード
      ELSE himan=0;
  ④ ARRAY d(i) dat1-dat5;
      DO OVER d;
        IF d=9 THEN d=.;                     ←    複数の変数の代入
      END;
RUN;
```

①は，「IF ステートメント」を使った条件文であり，変数「sex」が欠損値の場合は，そのケースを削除しろという意味になる．逆に，変数「sex」が 1 または 2 のケースだけ使いたいという指定方法は，「IF sex＝1 OR sex＝2;」と指定する.

②は，「割当てステートメント」と呼ばれ，右辺の計算結果が左辺の「bmi」という新しい変数の値となる．右辺の計算の部分には SAS 関数と呼ばれるさまざまな関数（整数部分を求める「INT」や絶対値を求める「ABS」など）を利用することもできるが，右辺の計算する変数の中に欠損値が含まれている場合は，左辺の変数の値も欠損値となる.

③は，「bmi」が欠損値ならば，変数「himan」に欠損値を，「bmi」が 25 以上の場合には 1 を，それ以外には 0 を代入することにより，新変数「himan」でリコードし

ている．なお，欠損値は取扱い上，負の無限大となっているので，条件を指定する場合には気をつける必要がある．

④は，「dat1-dat5」という 5 つの変数について，その値が 9 であれば欠損値とするという指定を配列変数（ここでは「d」にあたる）を利用して行っている．なお，ここで「DO OVER d;」という指定以降は，

```
DO i=1 TO 5 BY 1;
   IF d(i)=9 THEN d(i)=.;
   END;
```

という指定と同じ意味になる．

b. データの結合　この他にも，「SET ステートメント」を使って，複数のデータを縦に連結したり，「MERGE ステートメント」と「BY ステートメント」を使って，あるキーワード変数をもとにデータを横に連結したりということも簡単にできる．

たとえば，別々のファイルに入力された男性のデータ（datam）と女性のデータ（dataf）について，

```
DATA alldat;
   SET datam dataf;
```

とすれば，男性のデータの後ろに女性のデータが追加されたものが，「alldat」という新しいデータとしてできる．なお，次の例の dat1，dat2 のように，両方のデータに共通していない変数がある場合には，自動的に変数の数が増えるとともに，欠損値をもつものとして扱われる（変数 dat1，dat2 参照）．

「datam」

OBS	ID	name	age	sex	height	weight	dat1
1	101	TANAKA	24	1	173.2	74.5	1
2	102	FUKUDA	35	1	165.2	80.3	1
3	103	MAEDA	53	1	168.1	74.2	0
4	104	KOBAYASH	.	1	170.6	59.3	1

「dataf」

OBS	ID	name	age	sex	height	weight	DAT2
1	105	TAKEDA	29	2	162.5	48.5	2
2	106	HOSHINO	26	2	165.1	49.1	1
3	107	ASADA	23	2	157.3	41.5	1
4	108	IKEDA	25	2	153.5	59.7	1

「alldat」

OBS	ID	name	age	sex	height	weight	dat1	DAT2
1	101	TANAKA	24	1	173.2	74.5	1	.
2	102	FUKUDA	35	1	165.2	80.3	1	.
3	103	MAEDA	53	1	168.1	74.2	0	.

4	104	KOBAYASH	.	1	170.6	59.3	1	.
5	105	TAKEDA	29	2	162.5	48.5	.	2
6	106	HOSHINO	26	2	165.1	49.1	.	1
7	107	ASADA	23	2	157.3	41.5	.	1
8	108	IKEDA	25	2	153.5	59.7	.	1

また，別々のファイルに入力されたフェイスシートのデータ（dataface）と実験データ（dataex）をケースの「id」の値をキーワードにしてくっつける場合は，

```
PROC SORT DATA=dataface;BY id;
PROC SORT DATA=dataex;BY id;

 DATA newdata;
   MERGE dataface dataex;
   BY id;
```

のようにあらかじめ「SORT プロシージャ」を用いてキーワード変数（BY で指定）の順にソートした後に，「MERGE ステートメント」を利用することにより，すべての変数を含んだ「newdata」という新しいデータができる．

ただし，両方のデータに同じ変数名がある場合には「MERGE ステートメント」の右側のデータの値で上書きされてしまうので注意する必要がある．また，どちらかのデータに同じ「id」をもつケースが複数ある場合は，対応するデータを自動的に複製し，逆に対応する「id」がない場合には，欠損値データで不足部分を補うような仕様になっているので注意する．次の例の「id＝103，id＝104」のデータに具体例を示す．

「dataface」

OBS	ID	name	age	sex
1	101	TANAKA	24	1
2	102	FUKUDA	35	1
3	103	MAEDA	53	1
4	104	KOBAYASH	.	1

「dataex」

OBS	ID	height	weight	dat1
1	101	173.2	74.5	1
2	102	165.2	80.3	1
3	103	168.1	74.2	0
4	103	169.0	75.1	0

「newdata」

OBS	ID	name	age	sex	height	weight	dat1
1	101	TANAKA	24	1	173.2	74.5	1
2	102	FUKUDA	35	1	165.2	80.3	1
3	103	MAEDA	53	1	168.1	74.2	0
4	103	MAEDA	53	1	168.0	75.1	0
5	104	KOBAYASH	.	1	.	.	.

c. その他のステートメント　変数名がわかりにくい場合は，「LABEL ステートメント」で変数にラベルをつけたり（もちろん出力時には表示される），変数の値自体も「FORMAT ステートメント」で変更できる（このステートメントでリコードすることもできる）．

メモリが足りない場合などは，「KEEP ステートメント」や「DROP ステートメント」で必要な変数だけを選んだり，不要な変数を削ることもできるし，「OUTPUT ステートメント」を使ってデータの出力を制御することもできる．

SAS 関数やデータステップのステートメントに関して詳しく知りたい場合は，ヘルプメニューの「このウィンドウの使い方」や「SAS ヘルプとドキュメント」，ツールバーのヘルプボタン（  ），F 1 キーなどで表示させたヘルプ画面から，さまざまな項目を検索することができる．

A.2.4 プロックステップの制御

プロックステップでは，データの処理方法を指定する．一般的にプロックステップの始まりは，

PROC 処理方法 DATA＝処理するデータセット名［オプション］；

という形式になり，この後に処理方法に応じた各種のステートメントが列挙される．なお，処理するデータセット名の指定は省略できる（この場合は直前に作成したデータが利用される）が，複雑なプログラムではどのデータを扱っているのかわかりにくいこともあるので，明示しておいたほうが無難である．

簡単なものをいくつか紹介する．

a. データの出力　データがどのように読み込まれたかを確認するためには，「PROC PRINT;」を使う．

```
PROC PRINT DATA=data1;
RUN;
```

とすることで，読み込まれ，加工されたデータの一覧が以下のように表示される．「VAR ステートメント」を使って，「VAR id sex;」などとすれば，指定した変数だけを出力することも簡単にできる．

OBS	ID	name	age	sex	height	weight	dat1	DAT2	DAT3	DAT4	dat5	bmi	himan	l
1	101	TANAKA	24	1	173.2	74.5	1	2	1	0	2	24.8348	0	6
2	102	FUKUDA	35	1	165.2	80.3	1	1	2	0	2	29.4236	1	6
3	103	MAEDA	53	1	168.1	74.2	0	.	2	1	1	26.2584	1	6
4	104	KOBAYASH	.	1	170.6	59.3	1	2	2	1	2	20.3750	0	6

```
5  105  TAKEDA    29  2  162.5  48.5  0  2  1  0  2  18.3669  0  6
6  106  HOSHINO   26  2  165.1  49.1  1  1  2  1  2  18.0131  0  6
7  107  ASADA     23  2  157.3  41.5  .  1  .  1  .  16.7722  0  6
8  108  IKEDA     25  2  153.5  59.7  1  1  2  1  1  25.3371  1  6
```

なお，文字変数は標準で半角 8 文字なので「name」の一部が欠けてしまっているが，それ以外は指定どおりになっている．もちろん，データステップで「INPUT ステートメント」を指定する際に，「name $」の代りに「name $10.」などと最大文字数を指定すれば，きちんと読み込むことができる．

b.　クロス集計　　データのクロス集計を行うには，「PROC FREQ;」を使う．

```
PROC FREQ DATA=data1;
   TABLES sex sex*himan himan*(dat1-dat3 dat5);
RUN;
```

このように「TABLES ステートメント」に変数を並べれば，次のような結果が得られる．

```
                     FREQ プロシージャ

                                        累積         累積
sex        度数      パーセント         度数      パーセント
-------------------------------------------------------------
1             4         50.00              4         50.00
2             4         50.00              8        100.00

                     表：sex * himan

       sex             himan

       度数            |
       パーセント      |
       行のパーセント  |
       列のパーセント  |       0|       1|  合計
       ----------------+--------+--------+
                     1 |      2 |      2 |      4
                       |  25.00 |  25.00 |  50.00
                       |  50.00 |  50.00 |
                       |  40.00 |  66.67 |
       ----------------+--------+--------+
                     2 |      3 |      1 |      4
                       |  37.50 |  12.50 |  50.00
                       |  75.00 |  25.00 |
                       |  60.00 |  33.33 |
       ----------------+--------+--------+
       合計                   5        3       8
                          62.50    37.50  100.00
```

〈以降省略〉

つまり，「sex」のように変数を単独で指定すれば，単純な度数分布表になり，「sex*

himan」のような形にすれば縦を「sex」，横を「himan」にとったクロス集計表が出力される．また，「himan*(dat1-dat3 dat5)」という形にすれば，これを展開した形の4種類のクロス表（himan*dat1，himan*dat2，himan*dat3，himan*dat5）が一度に指定できる．この他に，「sex*age*himan」のように指定して，多重クロス表を出力することもできる（この例では，sex の階級ごとに「age*himan」のクロス表が出力される）．

なお，変数名の指定の後を「/」で区切って χ^2 検定などの統計値も出力するための「CHISQ オプション」などを入れられる．

より複雑な表を作成するためには，「PROC TABULATE;」を利用する．たとえば，

```
PROC TABULATE DATA=data1;
   CLASS sex himan;
   VAR height weight;
   TABLES (sex ALL)*(himan ALL),(height weight)*MEAN*F=6.1 / RTS=20;
 RUN;
```

のように分類変数を「CLASS ステートメント」で，平均値などを求めたい変数を「VAR ステートメント」で指定しておくと，性別，肥満別（全体を含む）に身長と体重の平均値の一覧が出力できる．なお，指定した6桁中の1桁は小数点のために使われるので，この例のように「F＝6.1」と指定した場合には整数部分4桁，小数点以下1桁で出力されるので注意が必要である．

```
---------------------------------
|                  |height|weight|
|                  |------+------|
|                  | MEAN | MEAN |
|------------------+------+------|
|sex      |himan   |      |      |
|---------+--------|      |      |
|1        |0       | 171.9|  66.9|
|         |--------+------+------|
|         |1       | 166.7|  77.3|
|         |--------+------+------|
|         |ALL     | 169.3|  72.1|
|---------+--------+------+------|
|2        |himan   |      |      |
|         |--------|      |      |
|         |0       | 161.6|  46.4|
|         |--------+------+------|
|         |1       | 153.5|  59.7|
|         |--------+------+------|
|         |ALL     | 159.6|  49.7|
|---------+--------+------+------|
|ALL      |himan   |      |      |
|         |--------|      |      |
|         |0       | 165.7|  54.6|
```

```
|         |---------+-----+-----|
|         |1        | 162.3|  71.4|
|         |---------+-----+-----|
|         |ALL      | 164.4|  60.9|
--------------------------------
```

「TABLES ステートメント」にあるカンマの前後がそれぞれ縦の項目と横の項目になるのが特徴である．もちろん，平均以外にも，ケース数や標準偏差なども簡単に指定できる．なお，オプションの「RTS＝20」は，「左端の列タイトル部分（Row Title Space）を 20 桁にしろ」という指定である．

c. 平均値の計算　平均値などを計算させるためには，「PROC MEANS;」を使う．標準では平均，標準偏差，最大，最小が出力されるが，オプションを指定すればその他の統計量も出力できる．また，「CLASS ステートメント」を追加することでサブグループごとの出力も簡単にできる．さらにここでは，「OUTPUT ステートメント」を利用することにより，身長の平均値を「meanh」という変数名で値としてもつ「out1」というデータセットを作成している．

```
PROC FORMAT;
   VALUE sexf 1="MALE" 2="FEMALE";
PROC MEANS DATA=data1;
   CLASS sex;
    VAR height weight bmi;
  OUTPUT OUT=out1 MEAN=meanh;
 FORMAT sex sexf.;
RUN;
```

この例では，「PROC FROMAT;」で「sex」の変数値に関して，フォーマットをあらかじめ指定するとともに，「PROC MEANS;」で性別に身長，体重，bmi の平均などを出力する際に，「FORMAT ステートメント」を利用している．そのため，実際の出力は，以下のようになる．（SEX の値が「1，2」でない点に注目）

MEANS プロシージャ

sex	オブザベーション数	変数	N	平均	標準偏差	最小値	最大値
MALE	4	height	4	169.2750000	3.4228400	165.2000000	173.2000000
		weight	4	72.0750000	8.9674876	59.3000000	80.3000000
		bmi	4	25.2229335	3.7581546	20.3749541	29.4235764
FEMALE	4	height	4	159.6000000	5.2012819	153.5000000	165.1000000
		weight	4	49.7000000	7.5064417	41.5000000	59.7000000
		bmi	4	19.6223201	3.8707497	16.7722239	25.3371389

なお，「OUTPUT ステートメント」で作成したデータセットは，データステップで

作成したデータと同じように，それ以降のステップで利用できる．

d. 分布の検討　「PROC UNIVARIATE;」を使って，「NORMAL」，「PLOT」などのオプションをつけると，正規性の検定や，ヒストグラム，BOXPLOT などの出力が得られる（ここではオプションの出力は省略する）．このプロシージャも「CLASS ステートメント」を使って，サブグループごとに調べることが簡単にできるが，今回は，「WHERE ステートメント」を使ってプロシージャで分析するケースを選択する方法と，「BY ステートメント」を利用する方法を紹介する．

次の例では，データを「sex」でソートした後で，「himan」が 0 のケースに関してだけ，身長の分布を性別に検討している．

```
PROC SORT DATA=data1;
   BY sex;
RUN;
PROC UNIVARIATE DATA=data1 NORMAL PLOT;
 WHERE himan=0;
  VAR height;
BY sex;
RUN;
```

実際の出力では，次のような統計値が得られる．このプロックステップでは，「FORMAT ステートメント」を指定していないので，「PROC MEANS;」のときの出力と違って，「sex＝2」のままであることにも注意してほしい（一部省略）．また，「WHERE ステートメント」でデータの制限を行ったこと（himan＝0 のデータだけを取り扱った）はログにしか表示されず，結果には明示されないので気をつける必要がある．

```
------------------------------------ sex=2 ------------------------------------

                        UNIVARIATE プロシージャ
                          変数 :  height

                              モーメント

N                          3    重み変数の合計             3
平均                161.633333    合計                 484.9
標準偏差             3.9715656    分散              15.7733333
歪度                -0.9352195    尖度                      .
無修正平方和          78407.55    修正済平方和      31.5466667
変動係数            2.45714514    平均の標準誤差    2.29298447

                            基本統計量
```

位置		ばらつき	
平均	161.6333	標準偏差	3.97157
中央値	162.5000	分散	15.77333
最頻値	.	範囲	7.80000
		四分位範囲	7.80000

〈以下省略〉

A.3　SAS の操作の流れ

SAS を利用する場合に，最も基本的と思われる操作方法を，データの読込みのところで示したデータ例が「C:¥data¥sample.dat」としてあり，プロックステップで説明したようなプログラムを含むプログラムファイル（具体的内容は説明の図中を参照）が「C:¥data¥test.sas」としてあることとして，順を追って紹介する．実際には，同じ命令を何種類かの操作で行えるので，状況に応じて使い分けることができる．

（1）インストールされた SAS のアイコン（  ）をダブルクリックする

最も基本的な SAS の起動方法であり，この操作により画面上に SAS のロゴが表示された後で，デフォルトでは以下の 5 つの基本ウィンドウが開く（図 A.1）．

結果を表示する「結果ウィンドウ」と「アウトプットウィンドウ」は，起動時には

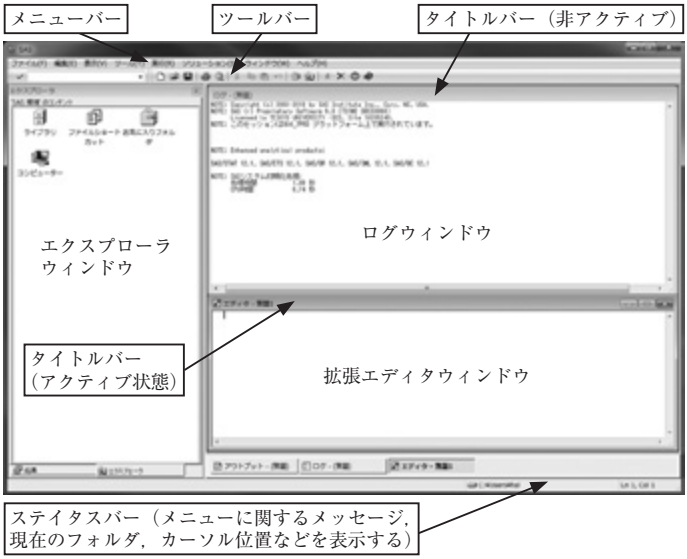

図 A.1　SAS 基本ウインドウの概要（一例）

不要なため他のウィンドウの裏側に表示されているが，下の方にあるタグやボタンを押すと前面に現れるようになっている．

これらのウィンドウは，アクティブ状態から「ツール」メニューの「オプション」を選択することにより，各自が利用しやすいように，表示する文字の大きさやフォントなどを変更することができる．

ただし，メニューバーの内容や，ツールバーのボタンは，どのウィンドウがアクティブになっているのかで変化するので注意すること．

ウィンドウの名称	特徴	表示位置
エクスプローラ	SAS ファイルの管理	起動時・ドッキング時に左
結果	SAS 出力結果を管理	常に左の裏側
ログ	処理状況の表示	起動時右上
拡張エディタ	プログラムの入力・編集・実行	起動時右下
アウトプット	出力結果の表示	起動時右の裏側

（2）「拡張エディタウィンドウ」の領域をクリックすることによりウィンドウをアクティブにし，プログラム入力や編集をする

プログラムは「拡張エディタウィンドウ」内に直接入力するか，すでに存在するファイルを読み込み，必要な部分を編集することになる．そのためには，「拡張エディタウィンドウ」のどこか一部をクリックするか F 5 キーを押すことにより，「拡張エディタウィンドウ」をアクティブ（タイトルバーの色が濃くなっている状態）にする必要がある．タイトルバー部分をダブルクリックするとアクティブになると同時にウィンドウが最大化されるので効率がよいかもしれない．

ファイルを読み込む場合には，ツールバーのオープンボタン（  ）をダブルクリックする方法や，ファイルメニューの「プログラムを開く」から，さらに「ファイル」を選択することにより，オープンダイアログ（図 A.2）を表示させて，ファイル名を

図 A.2 オープンダイアログ

図 A.3　拡張エディタウィンドウ（最大表示）

指定する．通常，SAS のプログラムファイルは，「*.sas」という拡張子がつくことになっており（図 A.2 中のファイルの種類参照），それ以外の拡張子のファイルは標準では表示されないので，他のエディタなどでプログラムを作成する場合にもこの拡張子をつけて保存したほうがよい．図 A.2 ではサンプルプログラムとしてあらかじめ作成した「test.sas」を指定している．ここで，「開く」ボタンをクリックすれば，「拡張エディタウィンドウ」にプログラムが読み込まれて，図 A.3 のように表示される．

なお，「ステイタスバー」の「現在のフォルダ」のアイコン部分をダブルクリックすると，「開く」などの操作時に，最初に指定されるフォルダを変更できるので便利である．

「拡張エディタウィンドウ」内では，通常のエディタと同様に入力（挿入）・削除・検索・置換などの各種の編集操作ができる．他の Windows のワープロやエディタのように，マウスや「SHIFT＋カーソルキー」などで必要な部分をマーク（反転）させた後に，プルダウンメニューやショートカットキー（「CTRL＋X」でカットなど）を利用して行う．また，文字の上書きモードと挿入モードは「INS」キーでも切換えできるが，画面上はカーソルの形程度しか変わらないので注意する必要がある．カット，コピー（CTRL＋C），ペースト（CTRL＋V），アンドゥ（CTRL＋Z）などの命令は，ツールボックスにもそれぞれボタン  としてあるので，そちらを利用することもできる．

「拡張エディタウィンドウ」はいままでの「プログラムウィンドウ」に比べて以下の

点で非常に使いやすくなっている．（なお，「プログラムウィンドウ」自体はなくなってしまったわけではなく，「表示」メニューを使えばいまでも利用できる．）

・プログラムが自動的に色分けされるので，ミスタイプが減少する．

プロシージャ名（紺色）
オプションなどのステートメント（水色）
数値（青緑色）
ファイル名などの文字データ（紫色）
コメント行など（黄緑色）
ミスタイプ（赤色）

・ステップごとに自動的に区切られ，まとめて取り扱えるので，プログラムの構造がわかりやすい．

ステップごとに自動的に横線が入り，さらにステップの頭に付く［−］ボタンをクリックすることにより，ステップ全体を折りたたんで 1 行にまとめて表示することができる．

なお，各ウィンドウは，ウィンドウの名称が書いてあるタイトルバーの部分をダブルクリックしたり，ウィンドウ右上の最大化ボタン（  ）をクリックしたりすることにより拡大するので，先の図のようにあらかじめ拡大しておくと編集が行いやすい．なお，拡大したウィンドウをもとの大きさに戻す場合には，再びタイトル部分をダブルクリックするか，メニューバーの右端にあるウィンドウ表示ボタン（  ）をクリックする．

（3）プログラム入力後，［F 8］キーを押して実行（submit）するか，ツールバーのサブミットボタン（  ）をクリックする

［F 8］キーを押したり，サブミットボタンをクリックしたりすることにより，「拡張エディタウィンドウ」内のすべてのプログラムが実行される．

出力がある場合には，「ログウィンドウ」と「拡張エディタウィンドウ」の裏側にある「アウトプットウィンドウ」が前面に表示されるので，注意すること．

（4）「ログウィンドウ」を調べて，エラーのないことを確認する

通常「ログウィンドウ」には，現在実行されているプログラムとその処理状況に関するメッセージが出力される．また，エラーが起った場合には，処理の途中にカラーのメッセージが出るので確認する（図 A.4 中の 15，18，19 行目参照）．

エラー（赤）やワーニング（緑）のメッセージは，通常のプログラム（黒）やノート（青）とは異なる色で表示される．多くの場合は，その直前に表示されているプログラムの該当箇所にアンダーラインとエラーコードが表示されるので，それを頼りに

図 A.4　ログウィンドウ（最大表示）

エラーの内容を確認することになる（「拡張エディタ」の場合には前述のように文字の色が違うので判別は容易になった）．「DATA」を「DATE」と間違えたような単純な入力ミスの場合には，ワーニングを出すだけで自動的に SAS のプログラムが修正してくれ，そのまま処理を続けてくれる場合もある（図 A.4 中の 18 行目参照）．また，オプションなどのステートメントをミスタイプした場合には，その場所で利用できるステートメントの一覧を表示してくれる場合もある（図 A.4 中の 19 行目参照）．しかし，データの読込みの際にずれがあったり，データの加工の段階で失敗したりするとエラーの原因がわかりにくいことが多いし，最悪の場合は，エラーが出ないで結果だけが間違っていることも起りうるので，エラーメッセージだけでなく，青色のノートの部分（データの読込み件数や変数の数などを表示）にも注意する習慣をつける必要がある．

（5）エラーがあった場合には，「拡張エディタウィンドウ」で修正し，再実行する

まず，「拡張エディタウィンドウ」の領域をクリックするか，ウィンドウ下部の「プログラム名」の書いてあるボタンを押すなどの操作を行い，アクティブにして，修正を加える．

「必要な部分だけをマウスなどを使って指定（文字を反転）する」ことにより，プログラムの部分再実行をすることもできる．データの読込みはうまくいったが，PROC ステップでエラーが出た場合などには，効率的である（図 A.5）．

なお，タイトルバーやボタンの「プログラム名」の後に「*」がついている場合は，

図 A.5 拡張プログラムウィンドウの一部（修正部分）を反転させ実行

なんらかの修正を加えた後で保存していないことを表している．

（6）読み込まれたデータをビューテーブルで確認する

データが具体的にどのように読み込まれているのかは，エクスプローラウィンドウを利用して，ビューテーブルで調べることができる．

前述のように，通常プログラムのデータステップで「DATA data1;」などと指定した場合，実際にはライブラリ「Work」の中に「data1」という一時データセットがつくられる．そこで，「エクスプローラウィンドウ」で「ライブラリ」，「Work」とダブルクリックしていき，最後に「data1」をダブルクリックすると，図 A.6 のようにビューテーブルウィンドウが開いて，実際に読み込まれた内容を Excel のような一覧表で確認することができる．

（なお，図 A.6 では標準設定とは違って，わかりやすいようにエクスプローラウィンドウでは「表示」メニューから「ツリー表示」を追加し，ビューテーブルの列幅も変更してある．）

ビューテーブルウィンドウでは，右ボタンで編集モードにすると簡単なデータの編集や並べ換え，行の追加・削除などもできる．標準的には，テーブルビューと呼ばれる表計算ソフトと同じ表示形式であるが（図 A.6 参照），表示メニューでフォームビューに切り換えると，1 ケースの変数値一覧を表示するモードにすることもできる（図 A.7 参照）．

右上角のめくれている部分をクリックすると，一度に表示できなかった次の変数一

図 A.6　エクスプローラウィンドウ（左側）とビューテーブルウィンドウ（右上）

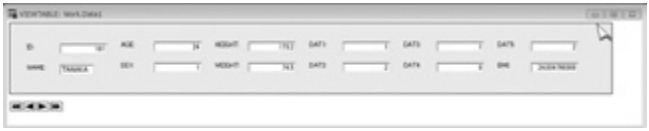

図 A.7　フォームビューのビューテーブルウィンドウ

覧が表示される．またウィンドウの左下にあるボタンで，1 ケースごとに変数値の内容を確認することが可能である．

なお，当然読み込まれた一時的なデータが変更されるだけであり，もとのデータファイルが修正されるわけではないので，注意が必要である．また，再度データを読み込んだりする場合などに，ビューテーブルが開いたままだとキチンと書換えが行われないので，確認が終わったらつねにウィンドウを閉じるように気をつける必要がある．

（7）ライブラリを作成し，永久データセットとして保存する

エクスプローラウィンドウ内のライブラリの部分でマウスの右ボタンメニューで新規作成を選択すると，図 A.8 のようなメニューが表示されて，ライブラリを新規作成することができる．

この例では，ライブラリ名は「test」であり，実際のディレクトリは「c:¥data(C ドライブの data)」が指定されている．そのため，ライブラリを作成後に，データステップの最初で「DATA test.outdata1;」という指定をすれば，C ドライブの「data」というディレクトリ内に「outdata1」という永久データセット（拡張子は「sas7bdat」）

図 A.8 新規ライブラリ作成メニュー

が作成されることになる．図 A.8 のように，右上の「起動時に有効」にチェックを入れておけば，次回以降も SAS 起動時に自動的に同じライブラリが作成される．つまり，次回からは最初にデータステップを実行して，データを読み込んだり加工したりしなくても，「PROC PRINT DATA = test.outdata1;」などプロックステップのデータセット名に直接永久データセット名を指定することで，毎回同じデータを簡単に利用できるわけである．

なお，SAS 以外のソフトで加工したデータを利用したい場合には，利用したいデータを「エクスプローラウィンドウ」で右クリックすれば，「クリップボードにコピー」したり，「Excel で表示」したりするだけでなく，「エクスポート」を利用して「CSV」形式などのファイルに出力することもできる．

（8）エラーがなければ，「アウトプットウィンドウ」で結果を確認する

前述のように，結果出力があると自動的に「アウトプットウィンドウ」が最大表示される．最初に表示されている部分は，結果の最後の部分なので，スクロールバーなどを使って画面を上下させ，結果を確認する（図 A.9 は「PROC FREQ」の出力結果の一部）．

複数のプロシージャを一度に実行した場合には，結果ウィンドウにプロシージャごとにフォルダの表示が現れる（図 A.9 左側参照）．希望する出力をダブルクリックするか，頭についている [+] ボタンをクリックすると，その内容が順に展開してツリー表示される．右クリックで表示されるメニューを使うと，「開く」以外にも「(その部分以下を）すべて展開」を選ぶこともできる（図 A.10 左側参照）．また逆に，不要な部分を [−] ボタンで折りたたんだり，右クリックで「すべて折りたたむ」を選択したりすることもできる．展開したツリー表示の中で，任意の部分をダブルクリックすれば，その部分が「アウトプットウィンドウ」に表示される．

なお，Ver 9.3 になって，出荷時の標準出力の形式が HTML に変更になり，そのため実行結果は「アウトプット」とは別の「結果ビューア」というウィンドウに表示されるようになった．実際には，結果を HTML ファイルとして出力し，それをブラウ

図 A.9　アウトプットウィンドウ（最大表示）

図 A.10　結果ウィンドウ（一部展開後）

ザで表示しているため，（不要な途中の出力を削除するなど）結果ウィンドウで出力結果をうまく管理することができないという難点がある．一方で，表のスタイルをみやすく変更したり，Excel に表を取り込んだりすることも簡単にできるので，最終結果の表示形式としては適しており，「試行錯誤はリスト形式で，最終結果は HTML 形式で」などとうまく使い分けるとよい（本文中の出力例は，リスト形式である）．

2 種類の出力形式の切換えは，「ツール」メニューの「オプション」の「プリファレンス」（図 A.11）から，それぞれ「リスト」（旧型式），「HTML」を選択してチェックする．

図 A.11 出力形式の設定ウィンドウ

ただし，「リスト」形式を利用するときに，「ODS Graphics を利用する」のチェックがついていると，グラフだけ図の形式で標準ディレクトリに次々ファイルとして出力されてしまうことになり扱いにくい（後から個別にファイル削除しないと SAS 終了時もグラフだけファイルとして残ってしまうため）．リスト形式を利用するときには，「ODS Graphics」のチェックも同時に外せば，グラフもいままでの出力形式（本文中のテキスト出力形式）になるのでわかりやすいはずである．

（8）結果を出力する場合には，まずアウトプットウィンドウをアクティブにして，ツールバー印刷プレビューボタン（ ）をクリックして，プレビュー画面で内容を確認したうえで，「印刷」を選択し，プリントダイアログで必要事項を指定して出力する（図 A.12）

図 A.12 プリントダイアログ

文字の大きさやフォントを変更したい場合には，プリントダイアログ内の「印刷設定」ボタンを選んで，「印刷設定」メニューを表示させ必要な部分を変更する（図 A.13）．

さらに，「ページ設定」ボタンで「ページ設定」メニューを表示させれば，用紙のサ

図 A.13　印刷設定メニュー

図 A.14　ページ設定メニュー

イズや印刷の向き，余白などを指定できる（図 A.14）．

なお，ツールバーの印刷ボタン（  ）を押すと，プリントダイアログが表示されずに，すべての結果が一度に出力されてしまうので，注意が必要である．

また，結果ウィンドウで任意の部分をクリックしてから，「ファイル」メニューで「印刷プレビュー」を選択して内容を確認した後に「印刷」すると，指定した部分のみを印刷することもできる．ここでも，右ボタンを押して「印刷」を選択すると，プリントダイアログが表示されずに，すぐに結果が出力されてしまう．

（9）各ウィンドウのすべての内容をファイルに保存したい場合には，そのウィンドウをアクティブにした後で，ファイルメニューの「名前を付けて保存」を選択し，保存する場所や名前の指定を行う（図 A.15）

ただし，保存する目的が，後から Windows のワープロソフトで編集することであ

図 A.15　名前を付けて保存メニュー

れば，直接カット＆ペーストを利用するほうが簡単である．希望する部分をマウスなどを使ってマーク（反転）し，編集メニューの「コピー（CTRL＋C)」を選択すれば，その内容はクリップボードに自動的に取り込まれ，他のソフトで張りつけられるようになる．

ファイルの種類は，アウトプットでは LIST ファイル（拡張子 lst)，ログでは LOG ファイル（拡張子 log)，プログラムでは SAS ファイル（拡張子 sas）に自動的に設定される．

なお，アウトプットとログの保存に関して同一名のファイルが存在する場合には，図 A.16 のメニューが表示されるので，「置換」か「追加」または，「キャンセル（保存しない)」の 3 つの方法を選択することができる．

図 A.16 保存方法選択メニュー

また，出力の一部分だけを保存したい場合などは，結果ウィンドウで右クリックを使って，「名前を付けて保存」を選択することもできる．

なお，どの出力（LIST，LOG，SAS）も通常のテキストファイルなので，メモ帳や Word などで内容を確認したり，編集することができる．また，出力するファイルの種類を変更すれば，出力結果をリッチテキストファイル（RTF）として保存することもできる．

(10) SAS を終了する

通常の Windows ソフトと同じようにウィンドウの左上にある終了ボタン（ ）をクリックするか，ファイルメニューの「終了」で終了することができる．