



Statistics • Visualization • Data manipulation • Reporting

GETTING STARTED WITH STATA

日本語マニュアル

FOR WINDOWS

Statistical software for data science

25年以上の経験と実績でお客様をサポートします。



STATA ガイド Getting Started Windows®版

リリース 18

訳 ライトストーン

Translated by LightStone Corp.



A Stata Press Publication

StataCorp LLC

College Station, Texas



Copyright© 1985-2023 StataCorp LP

All rights reserved

Version 18

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845

Typeset in TEX

このマニュアルは著作権で保護されています。無断転載を禁じます。**StataCorp LP** がソフトウェアおよびマニュアルを使用する目的で発行するライセンス諸条件により許諾された場合を除き、本マニュアルのいかなる部分も、**StataCorp LP** からの書面による事前の許諾なしに、いかなる形式または手段（電子的、機械的、複写、録音等を含む）による複製、検索システムへの保存、または転記を禁じます。禁反言またはそれ以外のものにより、明示または黙示を問わず、いかなる知的財産権に対するライセンスも、本文書によって譲渡されることはありません。

StataCorp は、本マニュアルを「現状のまま」で提供し、特定の目的への商品性や適合性の黙示的な保証を含み、それに限定しない明示または黙示されたいかなる保証も行いません。**StataCorp** は本マニュアル内で説明されている製品およびプログラムを予告なしに改善または変更を行うことがあります。

本マニュアルで記述されているソフトウェアはライセンス許諾または非開示許諾に基づきます。当該許諾に基づく場合に限り、ソフトウェアの複製の作成が許可されます。**DVD**、**CD**、ディスク、ディスクレット、テープ、あるいはそれ以外のメディアにバックアップおよびアーカイブ目的以外で複製することは法律に違反するものです。

この付属メディア内に出てくる自動車のデータセットの著作権は ©1979 by **Consumers Union of U.S., Inc., Yonkers, NY 10703-1057** にあり、再現するに当たり **CONSUMER REPORTS, April 1979** から許可を得ました。

特定のアイコンは **Axialis SA** からライセンス供与されています。それらは **Axialis SA** の所有物であり、複製または配布することはできません。

Stata, STATA, Stata Press, Mata, mata, and NetCourse は **StataCorp LP** の登録商標です。 **Stata** と **Stata Press** は国連の **World Intellectual Property Organization** に登録した商標です。 **NetCourseNow** は **StataCorp LP** の登録商標です。

それ以外のブランドまたは製品名はそれぞれの会社の登録商標です。

ソフトウェアに関する著作権の情報は「**help copyright**」と **Stata** 内で打ち込んでください。

ソフトウェアに関する引用は次のように行ってください。

StataCorp. 2023. *Stata: Release 18*. Statistical Software. College Station, TX: StataCorp LLC.

目次

1. Stata の紹介—サンプルセッション	9
2. Stata のユーザインターフェイス	40
3. Stata のユーザインターフェイス	54
4. ヘルプ・ヒントを見つける	61
5. Stata のデータセットを開く・保存する	71
6. データエディタを使用する	74
7. 変数マネージャを使用する	100
8. データをインポートする	105
9. データのラベリング	112
10. データのリストと基本コマンドの構文	121
11. 新しい変数を作成する	134
12. 変数やデータを削除する	144
13. do ファイルエディタを使用する—Stata の自動化	149
14. データを作図する	164
15. グラフを編集する	167
16. ログを使い結果の保存や印刷を行う	170
17. ウィンドウやフォントの設定をする	176
18. Stata について詳しく学ぶ	178
19. Stata のアップデートと拡張—インターネットでの機能	184
A. Stata のトラブルシューティング	193
B. 上級者向け Stata の使用法	196
C Windows 版 Stata について—追加要素	206

Stata の他のマニュアル参照について

本マニュアルを読み進めて行くと、他の **Stata** のマニュアルを参照している箇所があります。たとえば、次のように表現されます。

[\[U\] 26 Overview of Stata estimation commands](#)

[\[R\] regress](#)

[\[D\] reshape](#)

1 行目は *User's Guide* の 26 章、*Overview of Stata estimation commands* を参照しています。2 行目は *Base Reference Manual* の **regress** を、3 行目は *Data Management Reference Manual* の **reshape** を参照しています。

上記 [U] のように **Stata** のマニュアルには略称が割り振られています。

[GSM]	<i>Getting Started with Stata for Mac</i>
[GSU]	<i>Getting Started with Stata for Unix</i>
[GSW]	<i>Getting Started with Stata for Windows</i>
[U]	<i>Stata User's Guide</i>
[R]	<i>Stata Base Reference Manual</i>
[ADAPT]	<i>Stata Adaptive Designs: Group Sequential Trials Reference Manual</i>
[BAYES]	<i>Stata Bayesian Analysis Reference Manual</i>
[BMA]	<i>Stata Bayesian Model Averaging Reference Manual</i>
[CASUAL]	<i>Stata Casual Inference and Treatment-Effects Estimation Reference Manual</i>
[CM]	<i>Stata Choice Models Reference Manual</i>
[D]	<i>Stata Data Management Reference Manual</i>
[DSGE]	<i>Stata Dynamic Stochastic General Equilibrium Models Reference Manual</i>
[ERM]	<i>Stata Extended Regression Models Reference Manual</i>
[FMM]	<i>Stata Finite Mixture Models Reference Manual</i>
[FN]	<i>Stata Functions Reference Manual</i>
[G]	<i>Stata Graphics Reference Manual</i>
[IRT]	<i>Stata Item Response Theory Reference Manual</i>
[LASSO]	<i>Stata Lasso Reference Manual</i>
[XT]	<i>Stata Longitudinal-Data/Panel-Data Reference Manual</i>
[META]	<i>Stata Meta-Analysis Reference Manual</i>
[ME]	<i>Stata Multilevel Mixed-Effects Reference Manual</i>
[MI]	<i>Stata Multiple-Imputation Reference Manual</i>
[MV]	<i>Stata Multivariate Statistics Reference Manual</i>
[PSS]	<i>Stata Power, Precision, and Sample-Size Reference Manual</i>
[P]	<i>Stata Programming Reference Manual</i>
[RPT]	<i>Stata Reporting Reference Manual</i>
[SP]	<i>Stata Spatial Autoregressive Models Reference Manual</i>
[SEM]	<i>Stata Structural Equation Modeling Reference Manual</i>
[SVY]	<i>Stata Survey Data Reference Manual</i>
[ST]	<i>Stata Survival Analysis Reference Manual</i>
[TABLES]	<i>Stata Customizable Tables and Collected Results Reference Manual</i>
[TS]	<i>Stata Time-Series Reference Manual</i>
[I]	<i>Stata Index</i>

[M]

Mata Reference Manual

このマニュアルについて

これは、**Windows®**版 **Stata** のマニュアルです。**Mac®**版 **Stata** ご使用の方は「**Stata Getting Started Mac 版**」を、**Unix®**版 **Stata** をご使用の方は「**Stata Getting Started Unix 版**」をそれぞれご覧ください。このマニュアルは **Stata** 入門者から **Stata** の **Windows** 版を初めて使用する、という方向けに作成しました。既存ユーザには **Windows** 版 **Stata** の新機能のチュートリアルとしてもご利用いただけます。

このマニュアルの本編が **19** 章、付録が **3** 章です。付録には **Windows** 版 **Stata** 専用の情報を記載しました。

ユーザ登録をした方は複数のテクニカルサポートを利用できます。[\[GSW\] 4 Getting Help\(ヘルプ・ヒントを見つける\)](#) では **Stata** のコマンドや機能を学ぶ手助けとなるリソースの紹介をしています。リソースの1つとして **Stata** のウェブサイト (<https://www.stata.com>) があります。サイト上にはよくある質問 (**FAQ**) 等、多くの情報があります。ウェブサイトと [\[GSW\] 19 Updating and extending Stata—Internet functionality \(Stata のアップデートと拡張—インターネットでの機能\)](#) で説明されている資料を参考にしても分からないところがある場合、[\[U\] 3.8 Technical Support](#) を参照してください。

マニュアルを使用する

Stata 入門者はこのマニュアルを演習テキストとして、各例題を実際にコンピュータで操作しながら学ぶことをお勧めします。例題はステップ形式になっているので同じデータを複数回にわたり使用していきます。ちょうど統計学そのものに多くの手法と奥深さがあるように、**Stata** は奥深く豊富な統計機能を持つソフトウェアです。例題に取り組むことで統計の知識も身に付くので、実際にデータ分析を行う際に練習の効果が表れるはずです。

これは **Stata** 入門者向けのマニュアルですが、熟練ユーザでもこのマニュアルから学べることもあるかもしれません。熟練ユーザはまず目次を見て、何か新しいものがないか、忘れていないか、と項目を確認してみてください。

1. Stata の紹介—サンプルセッション

1.1 Stata の紹介

この章ではサンプルのワークセッションを実際に使用しながら **Stata** で実行出来る基本的な操作を説明します。データセットを開く、データセットの内容を調べる、記述統計量を求める、グラフを作成する、そして簡単な回帰分析を行う等の操作を紹介します。いずれも導入的な簡単な内容です。**Stata** で何が出来るのか、どのように動作するのかを理解する助けとなります。説明はなるべく簡潔に記します。必要に応じてこのマニュアルの別の箇所、もしくはシステムヘルプや他のマニュアルへのリファレンス情報を示しますので参照してください。メニューとダイアログによる操作と、コマンドによる操作を併用して説明するので、どちらの操作も体験できます。**Stata** のメニュー表記が本書と異なる場合、読みやすいように本書と同じ言語に変更することをお勧めします。変更方法については [\[P\] set locale ui](#) をご覧ください。

コンピュータの前に座り、この本で勉強していきましょう。

1.2 Stata の紹介

このセッションではアメリカ国内における **1978** 年の年代物の自動車販売データを利用します。

カーソルを合わせてクリックを行う操作はメニュー>メニュー内項目>サブメニュー項目>などのように表記します。コマンドウィンドウを使用して操作を行う場合は、(.)の後に続くコマンドを画面下部にあるコマンドと書かれた小さなウィンドウに入力してください。何かコマンドの構成の中で気を付けるべきことがある場合、“構文メモ”として記します。

では、まず **automobile** データセットをロードしましょう。このデータセットは **Stata** に初めから入っています。メニューを使用して、以下のように操作します。

1. ファイル>例題データセット... と操作します。
2. 表示されるウィンドウ内から **Example datasets installed with Stata** をクリックします。
3. **auto.dta** (リストの一番上)の横にある **use** をクリックします。

このコマンドの結果は 4 つに分類する事ができます。

- 中央に位置する大きな結果ウィンドウには次のコマンドを表示します：

```
. sysuse auto  
(1978 automobile data)
```

このウィンドウはコマンドとその結果を表示します。コマンド「`sysuse auto.dta`」はピリオド(.)の後に太字で書かれます。結果 (**1978 automobile data**) は標準字体で、データセットの簡単な説明と一緒に表示します。


メモ： コマンドの意味や使用方法が知りたい場合、コマンドウィンドウに「`help` 半角スペース」の後に「コマンド名」をひとつ入力すると、そのコマンドに関するヘルプを表示します。または、メニューバーでヘルプ > 検索... と選択すると、さらに詳しい情報がいつでも検索できます。

- 画面左側の縦長な履歴ウィンドウに結果ウィンドウと同じコマンド、`sysuse auto.dta` を表示します。コマンドの成功・失敗 (エラー)にかかわらず履歴ウィンドウは **Stata** が実行したすべてのコマンドを表示します。これらのコマンドは簡単に再実行できます。詳しくは [\[GSW\] 2 Stata user interface \(Stata のユーザインターフェイス\)](#) をご覧ください。
- 画面右上の小さな変数ウィンドウには変数の一覧を表示します。
- 画面右下にある、小さなプロパティウィンドウにはデータセットの 1 番目の変数、`make` についての情報を表示します。

コマンドウィンドウに「`sysuseauto`」と入力してから **Enter** キーを押してもデータセットは開きます。これも一度試してみてください。`sysuse` はサンプル (システム) データセットをロード (使用) するコマンドです。このセッションで利用するように、**Stata** のコマンドはとても単純なのでコマンドウィンドウに直接入力して使用するほうが作業時間を短くできます。**Stata** を日常的に使用する場合、使用頻度の高いコマンドを覚えておくと効率的に作業できるようになります。

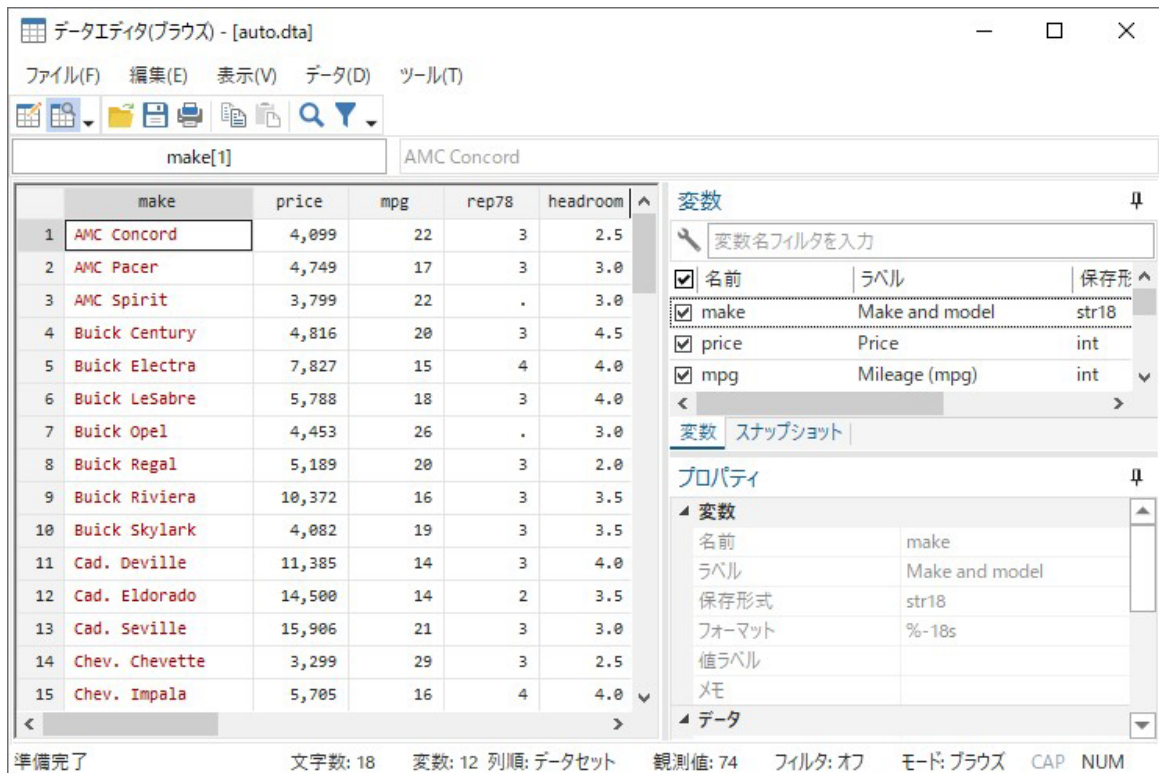
構文メモ： 上記の例では `sysuse` が **Stata** のコマンドで、`auto` は **Stata** のデータファイルの名前です。

1.3 簡単なデータ管理

データセットはデータエディタで見ることができます。データエディタ(ブラウズ)ボタンをクリックするか、データ > データエディタ > データエディタ (ブラウズ) とメニュー操作するか、「browse」コマンドをコマンドウィンドウに入力すると、同じようにデータエディタが開きます。

構文メモ：データエディタボタンをクリックするだけでは、データセットは変更されないため、コマンドは発行されません。

データエディタを開くと Stata がデータを表形式で表示します。これはすべての Stata のデータセットについて同じです。列は変数を、行は観測値 (データ) を表します。変数には分かりやすい名前が付き、観測値には番号が振られます。



	make	price	mpg	rep78	headroom
1	AMC Concord	4,099	22	3	2.5
2	AMC Pacer	4,749	17	3	3.0
3	AMC Spirit	3,799	22	.	3.0
4	Buick Century	4,816	20	3	4.5
5	Buick Electra	7,827	15	4	4.0
6	Buick LeSabre	5,788	18	3	4.0
7	Buick Opel	4,453	26	.	3.0
8	Buick Regal	5,189	20	3	2.0
9	Buick Riviera	10,372	16	3	3.5
10	Buick Skylark	4,082	19	3	3.5
11	Cad. Deville	11,385	14	3	4.0
12	Cad. Eldorado	14,500	14	2	3.5
13	Cad. Seville	15,906	21	3	3.0
14	Chev. Chevette	3,299	29	3	2.5
15	Chev. Impala	5,705	16	4	4.0

データは複数の色で表示されます。一見すると黒は数値を、他の色は文字を示しているようです。では、確認してみましょう。変数 `make` の下にあるセルを、1つクリックします。ウィンドウ上の入力ボックス (ウィンドウ内の上部、ボタンのあるツールバーの下にある灰色のエリア) には車のメーカーが表示されます。変数 `foreign` が見えるまで右にスクロールし、その列のセルを1つクリックします。クリックしたセ

ルは“Domestic”と表示していますが、入力ボックスには **0** が表示されます。Stata はデータ分類のカテゴリを数字で保存します。しかし、数値のままではその意味が分かりづらいので、一目で分類内容が伝わるように文字を表示できます。これを値ラベルと呼びます。最後に変数 **rep78** は数値データを表しているように見えますが、いくつかのセルはピリオド (.) だけを表示しています。このピリオドは欠損値を表します。

データエディタで見るデータは見やすいですが、データセットについての情報は限られます。データを分析するとき、何を表すデータなのか、どのように保存しているのか、という詳細が分かると便利です。データエディタを閉じて Stata のメインウィンドウに戻ります。

データセットの構造は **describe** コマンドで詳しく確認できます。データ > データの内容表示 > メモリ/ファイル内のデータの内容表示 とメニュー選択し OK をクリックするか、コマンドウィンドウに「describe」と打ち込み、**Enter** キーを押します。どちらの方法でも同じ結果を表示します。

```
. describe
```

Contains data from C:\Program Files\Stata18\ado\base/a/auto.dta

Observations: 74 1978 automobile data

Variables: 12 13 Apr 2022 17:45
(_dta has notes)

Variable name	Storage type	Display format	Value label	Variable label
make	str18	%-18s		Make and model
price	int	%8.0gc		Price
mpg	int	%8.0g		Mileage (mpg)
rep78	int	%8.0g		Repair record 1978
headroom	float	%6.1f		Headroom (in.)
trunk	int	%8.0g		Trunk space (cu. ft.)
weight	int	%8.0gc		Weight (lbs.)
length	int	%8.0g		Length (in.)
turn	int	%8.0g		Turn circle (ft.)
displacement	int	%8.0g		Displacement (cu. in.)
gear_ratio	float	%6.2f		Gear ratio
foreign	byte	%8.0g	origin	Car origin

Sorted by: foreign

リストの一番上にデータセット全体の情報、たとえばデータの保存場所、メモリの使用量や最終保存時間を簡潔に表示します。太字の **1978 automobile data** はデータセットが開かれた時に表示される簡単な説明

で、**Stata** ではこれをデータセットのラベルと呼びます。 `_dta has notes` の部分はデータセットにメモが添付されていることを表します。メモの内容はコマンドウィンドウに「notes」と打ち **Enter** キーを押すと結果ウィンドウで確認できます。

```
. notes
_dta:
  1. From Consumer Reports with permission
```

元データに関する簡単なメモを見ることができます。

`describe` コマンドによるリストを見返すと、元のデータ以外の情報を **Stata** が保持していることがわかります。

全ての変数には次に示すフィールドが用意されています。

- **variable name** (変数名) には **Stata** で操作するためのデータの名前が入ります。 **variable name** は **Stata** が利用する **name** の 1 つです。詳しくは [\[U\] 11.3 Naming conventions](#) をご覧ください。
- **storage type** (保存タイプ) はデータ保存形式を示します。現時点では `str` がつくタイプは文字列(テキスト)変数を表し、その他のタイプは数値であることを理解していれば十分です。このデータセットの中にはありませんが、**Stata** では任意の長い文字列 `strL`(スタール)も使用出来ます。`strL` はバイナリ形式も格納できます。詳しくは [\[U\] 12.4 Strings](#) をご覧ください。
- **display format** は表形式でデータを表示する時に利用します。詳しくは [\[U\] 12.5 For - mats:Controlling how data are displayed](#) をご覧ください。
- **value label** (値ラベル)はデータセットの中に値ラベルを含む場合に記述が表れます。これは各観測値に文字を紐づけることで文字列を表示します。詳しくは [\[GSW\] 9 Labeling data](#) (データのラベリング) と [\[U\] 12.6.3 Value labels](#) をご覧ください。
- **variable label** (変数ラベル)はデータ作成者以外でも変数の情報が分かるように用意されました。この変数ラベルは表作成時に使用します。

データセットにはデータのみではなく、より多くの情報を付加できます。これらの情報があればデータ作成者以外の研究者にとっても便利です。

`describe` コマンドはデータ構成に関する情報をユーザに提供しますが、データについてはほとんど説明しません。このデータの要約を表示するには 統計 > 要約/表/検定 > 要約と記述統計量 > 記述統計量 と操作し、

OK ボタンをクリックします。あるいはコマンドウィンドウに「`summarize`」と打ち込み、**Enter** キーを押します。結果はデータセット内すべての変数に関する記述統計量を表形式で出力します。

. summarize					
Variable	Obs	Mean	Std. dev.	Min	Max
make	0				
price	74	6165.257	2949.496	3291	15906
mpg	74	21.2973	5.785503	12	41
rep78	69	3.405797	.9899323	1	5
headroom	74	2.993243	.8459948	1.5	5
trunk	74	13.75676	4.277404	5	23
weight	74	3019.459	777.1936	1760	4840
length	74	187.9324	22.26634	142	233
turn	74	39.64865	4.399354	31	51
displacement	74	197.2973	91.83722	79	425
gear_ratio	74	3.014865	.4562871	2.19	3.89
foreign	74	.2972973	.4601885	0	1

この簡単な記述統計量から、データの様子が少し分かります。まず価格 (`price`) が現代の車とは全く異なります。アンティーク並みの古い車なので不思議ではありません。また、燃費 (`mpg`) も決してよくありません。自動車愛好家ならば他の細かい特徴からも性能について想像できるでしょう。

さらに重要なポイントが 2 つあります。

- 変数 `make` の観測値 (`Obs`) が 0 です。この変数は文字列 (テキスト) の変数で、数値データはありません。
- 変数 `rep78` は他の数値的な観測数よりも 5 つ少なくなっています。これは `rep78` に 5 つの欠損値があることを示しています。

`summarize` コマンドと `describe` コマンドを使用すれば、データセットの概要を確認できます。Stata にはデータセットをより深く、細部にわたり説明をするコマンドとして `codebook` があり、構成、内容、変数の値など幅広く表示します。コマンドウィンドウに「codebook」と入力して **Enter** キーを押すか、メニューからデータ>データの内容表示>コードブックの表示と選択し **OK** をクリックします。このシンプルなコマンド **1** つで多くの情報を表示します。必要に応じて結果ウィンドウをスクロールバックし、今までの出力結果も確認しましょう。これから変数 `make`, `rep78`, `foreign` の出力について詳しく見ていきます。

調査を始めるにあたり **1** つの変数、たとえば `make` だけに `codebook` コマンドを実行します。この操作もコマンドとメニュー、どちらからでも実行できます。メニュー操作で変数を選ぶには、まずメニューからデータ>データの内容表示>コードブックの表示と操作してダイアログを開きます。ダイアログを使用して変数 `make` にだけ `codebook` を適用する場合、次に示す **2** つの方法があります。

- 変数欄に直接「make」と入力します。
- 変数欄は直接入力の外にリストから選択もできるようになっています。欄の右端にあるドロップダウンを示す下三角形をクリックすると、データセット内にある変数のリストを表示します。このリスト変数 `make` を選択すると、編集エリアに `make` が入ります。

もっとも、コマンドウィンドウに「codebook make」と入力し、**Enter** キーを押すのが一番簡単です。出力した結果は次の通りです。

```
. codebook make

-----
make                                     Make and model
-----
                                Type: String (str18), but longest is str17

Unique values: 74                      Missing "": 0/74

Examples: "Cad. Deville"
           "Dodge Magnum"
           "Merc. XR-7"
           "Pont. Catalina"

Warning: Variable has embedded blanks.
```


出力結果の最初の列は変数名 (**make**) と変数ラベル (**Make and model**) を表しています。変数は文字列 (**string**) として保存されています。文字列は最長 **17** 文字 (**str17**) ですが、**18** 文字 (**str18**) で保存しているようです。全ての値がユニークなので、必要に応じて変数 **make** は観測値の識別子になります。識別子は複数の元データからデータセットを取りまとめる時や、データ内からエラーを抽出するのに便利です。欠損値 (**missing**) はありませんが、**make** の文字列の中にスペースがあります。変数 **make** が一単語 (スペースなし) の文字列変数だと想定しているなら、気を付けなければなりません。

構文メモ: 「**codebook make**」コマンドは引数として **varlist** (変数リスト) を使用するコマンドの一例です。

次に変数 **foreign** から値ラベルについて学びましょう。この変数のコードブック出力を確認します。コマンドウィンドウにコマンドを入力する方が簡単なので「**codebook foreign**」と入力します。(以降、「**Enter** キーを押す」という記述は省略します。) 次のような出力結果になります。

```
. codebook foreign
```

```
foreign                                     Car origin
```

```

      Type: Numeric (byte)
      Label: origin

      Range: [0,1]
Unique values: 2                               Units: 1
                                           Missing .: 0/74

      Tabulation: Freq.   Numeric   Label
                   52         0   Domestic
                   22         1   Foreign
```

出力された表から、変数 **foreign** の値は **0** と **1** だけなのでダミー変数だと分かります。変数には値ラベルがあり、**0** の時には“**Domestic**”、**1** の時は“**Foreign**”と数字の代わりに表示します。このデータ表示形式の利点は **2** つあります。

- 変数が使用するメモリ量を減らします。数値の場合、容量は 1 バイトのみですが、文字列“Domestic”の場合 8 バイトになります。詳しくは [\[U\] 12.2.2 Numeric storage types](#) をご覧ください。
- ダミー変数として統計モデルに組み込むことができます。詳しくは [\[U\] 26 Working with categorical data and factor variables](#) をご覧ください。

最後にラベル付けが不十分で、欠損値がある例を変数 `rep78` から見ていきましょう。コマンドウィンドウに

「`codebook rep78`」を入力し、実行すると次のようになります。

```
. codebook rep78
```

```
rep78                                     Repair record 1978
```

```

                                Type: Numeric (int)
                                Range: [1,5]
                                Unique values: 5
                                Units: 1
                                Missing .: 5/74

                                Tabulation: Freq. Value
                                           2 1
                                           8 2
                                           30 3
                                           18 4
                                           11 5
                                           5 .

```

`rep78` はカテゴリー変数のようです。しかしデータにはこれ以上の説明がないので、カテゴリー分けした数字が何を意味するのか分かりません。(値にラベルを付けるには [\[GSW\] 6 Using the Data Editor \(データエディタを使用する\)](#) の「データを変更する」と [\[GSW\] 9 Labeling data \(データのラベリング\)](#) をご覧ください。) この変数には欠損値が 5 つあります。これは 5 つの車種の修理記録(`repairrecord`)が存在しないことを示します。データエディタを使用してこれらの 5 つの観測値を詳しく確認します。「簡単なデータ管理」の冒頭で説明したように、データエディタ (ブラウザ) を出力するコマンドは `browse` でした。変数 `rep78` では欠損値だけを確認したいので、次のようにコマンドウィンドウに入力します。

```
. browse if missing(rep78)
```

	make	price	mpg	rep78	headroom
3	AMC Spirit	3,799	22	.	3.0
7	Buick Opel	4,453	26	.	3.0
45	Plym. Sapporo	6,486	26	.	1.5
51	Pont. Phoenix	4,424	19	.	3.5
64	Peugeot 604	12,990	14	.	3.5

表示されたデータエディタを見ると、「.」の値は欠損値であることが分かります。他の変数にも欠損値があっても問題はありません。「.」は数値欠損値のデフォルトの表示形式です。また、Stata では「.a」から「.z」までのユーザ欠損値を設定できますが、このデータセットの中にはありません。詳しくは [\[U\] 12.2.1 Missing values](#) をご覧ください。確認が終了したら、ウィンドウ右上の **x** ボタンをクリックしてデータエディタ (ブラウザ) を閉じます。

構文メモ：上記のように `if` コマンドを使用すると観測値 (データ) のサブセットを表示します。

データを一通り確認してもなぜ特定の値が欠損しているのか分かりません。この場合、データの出典元にはじめから数値が無い可能性と、誤って数値を省いた可能性を確認します。変数 `make` の値はユニークなので修理記録に欠損値がある車の情報をリストすれば情報の有無を確認できます。メニューおよびダイアログで操作します。

1. データ > データの内容表示 > データの一覧表示と選択します。
2. 変数欄の右端にある下三角形をクリックして変数名を表示します。
3. その中から `make` を選んで変数欄に入力します。
4. ダイアログ内の `by/if/in` タブをクリックします。
5. `missing(rep78)` を条件式ボックスに打ち込みます。

6. 適用をクリックします。すると、ダイアログは開いたままでコマンドを実行します。コマンドを試すとき、調べるとき、そして複雑なものを作成するとき等に適用ボタンはとても便利です。このサンプルでは基本的に適用を使用します。ここで **OK** を押してダイアログを閉じて構いません。

コマンドウィンドウに「`list makeif missing(rep78)`」と入力しても上記メニュー操作と同じ結果になります。`list` コマンドは観測値 (データ) のリストを作るものであり、コマンド入力の方が簡単です。出力結果を次に示します。

```
. list make if missing(rep78)
```

	make
3.	AMC Spirit
7.	Buick Opel
45.	Plym. Sapporo
51.	Pont. Phoenix
64.	Peugeot 604

データの出典元にはこれ以上の情報が無く、この欠損値をなくすことはできません。詳しくは [\[GSW\] 10 Listing data and basic command syntax \(データのリストと基本コマンドの構文\)](#) で `list` コマンドの機能をご覧ください。

構文メモ：このコマンド (`if` 条件と `missing()` 関数) は私たちに 2 つの新しいコンセプトを提供します。`if` 条件は `if` 以下の条件に当てはまる観測値にのみコマンドを実行します。詳しくは [\[U\] 11.1.3 if exp](#) をご覧ください。`missing()` 関数は各観測値に欠損値があるかどうかを調べます。詳細は [\[FN\] Programming functions](#) をご覧ください。

では、データセットそのものが分かってきたのでデータ自体の調査に移りたいと思います。

1.4 記述統計量



前のセクションから、`summarize` コマンドは簡単な記述統計量をすべての変数について出力することが分かりました。データの記述統計量を見たところ、車の価格であるにもかかわらず、価格がとても安い事が気

になります(1978年なので安いのは当たり前ですが)。この変数 `price` をより詳しく調べる為、以下のよう
に操作します。

1. 統計 > 要約/表/検定 > 要約と記述統計量 > 記述統計量 を選択します。
2. 変数欄に直接 `price` と入力するか、右の下三角形のリストから選びます。
3. オプション内の 追加の統計量を表示する のラジオボタンを選択します。
4. 適用をクリックします。

構文メモ：結果ウィンドウからも分かるように、「`summarize price, detail`」とコマンドウィンドウに入
力しても結果は同じです。カンマの後の部分は **Stata** コマンドではオプションを表します。つまり、以下の構
文では `detail` はオプションの例となります。

. summarize price, detail				
Price				
Percentiles		Smallest		
1%	3291	3291		
5%	3748	3299		
10%	3895	3667	Obs	74
25%	4195	3748	Sum of wgt.	74
50%	5006.5		Mean	6165.257
		Largest	Std. dev.	2949.496
75%	6342	13466		
90%	11385	13594	Variance	8699526
95%	13466	14500	Skewness	1.653434
99%	15906	15906	Kurtosis	4.819188

出力結果から、このデータセット内の車の値段の中央値はわずか**\$5,006.50**だと分かります。そして高価な
車4台はすべて**\$13,400**から**\$16,000**の範囲にあります。この最も高価な価格帯にある車を詳しく調べるに
は(そしてデータエディタを少し使うには)まずデータエディタ(ブラウズ)ボタンを押します。データ
エディタが開いたら観測値フィルタボタンを押すと観測値フィルタダイアログが出てきます。式によ
るフィルタ欄に「`price > 13000`」と打ち込むと**\$13,000**より高い車のみを表示します。

観測値フィルタ

式によるフィルタ:
 ...

観測値によるフィルタ

範囲限定子を使用する:

観測値の番号による
 開始: 終了:

ライブフィルタを使用する

フィルタを適用するボタンを押すと最高価格帯にある 4 台の車が表示され、2 つは **Cadillac** 車 (変数 **make** の前半が **Cad.**) で残りの 2 つは **Lincoln** 車 (変数 **make** の前半が **Linc.**) です。この 4 台は決して燃費が良い車ではありません。

データエディタ(ブラウズ) - [auto.dta]

ファイル(F) 編集(E) 表示(V) データ(D) ツール(T)

make[12] Cad. Eldorado

	make	price	mpg	rep78	headroom
12	Cad. Eldorado	14,500	14	2	3.5
13	Cad. Seville	15,906	21	3	3.0
27	Linc. Mark V	13,594	12	3	2.5
28	Linc. Versailles	13,466	14	3	3.5

変数

変数名フィルタを入力

名前 | ラベル | 保:
 make | Make and Model | str
 price | Price | int

変数 スナップショット

プロパティ

変数
 名前 | make
 ラベル | Make and Model

準備完了 文字数: 18 変数: 12 列順: データセット 観測値: 4 / 74 フィルタ: オン モード: ブラウズ CAP NUM

先ほどデータの内容を簡単に確認した時、外国製の車の修理記録の方が良かったようなので、これから外国製の車と修理記録の関係について調べようと思います。(ここで、カテゴリー1、2、3、4、5 が何を意味するのか分かりませんが、Chevy の Monza (カテゴリー 2) は壊れやすいと評判でした。) では、データセット内の外国製の車の割合と、各修理記録の割合を見てみましょう。これは一元表(one-way table) で確認できます。外国製の車に関する表を作成するには次のように操作します。統計 > 要約/表/検定 > 度数分布表 > 一元配置表と選択しカテゴリ変数欄でドロップダウンリストから変数 **foreign** を選択します。適用を押すと次の結果を表示します。

```
. tabulate foreign
```

Car origin	Freq.	Percent	Cum.
Domestic	52	70.27	70.27
Foreign	22	29.73	100.00
Total	74	100.00	

この結果からデータセット内の約 **70%** は国内製 (**domestic**) すなわちアメリカ製で、**30%** は外国製 (**foreign**) だと分かります。この表の **Car type** 欄では **0** と **1** の数値ではなく、見やすくなるように値ラベルを使用しています。

構文メモ：結果ウィンドウから、この一元表は **tabulate** コマンドの後に変数名 **foreign** を加えることでも作成できます。修理記録 (**rep78**) の一元表を作成するにはコマンドウィンドウに「**tabulate rep78**」と入力しましょう。次のように、カテゴリ別に表示されます。

```
. tabulate rep78
```

Repair record 1978	Freq.	Percent	Cum.
1	2	2.90	2.90
2	8	11.59	14.49
3	30	43.48	57.97
4	18	26.09	84.06
5	11	15.94	100.00
Total	69	100.00	

このカテゴリ「**3**」が何を意味するのかは分かりません。しかし、ほとんどの車は **3** 以上のカテゴリに入っています。おそらく、カテゴリ **1** は最も悪い (修理記録の) 評価を、**5** は良い評価を表しているのでしょう。この推測を元にデータセットの説明を続けていきます。度数 (**Freq.**) が **74** では無く **69** なので **5** つの欠損値の存在が確認できます。

外国製と国内製の修理記録を比較するには **2** つの一元表よりは、むしろ **1** つの二元表の方が適しているの
でそれを作成します。メニューで次のような操作をします。

1. 統計 > 要約/表/検定 > 度数分布表 > 二元配置表/関連係数 を選択します。
2. 行の変数にドロップダウンリストから `rep78` を選びます。
3. 列の変数にも同じように `foreign` を選びます。
4. 変数 `foreign` 内にはパーセント表示があるほうが良いのでセルの内容の行内の相対度数にチェックを付けます。
5. 適用をクリックします。

出力結果は次のようになります。

```
. tabulate rep78 foreign, row
```

Repair record		Car origin		Total
1978	Domestic	Foreign		
1	2 100.00	0 0.00	2 100.00	
2	8 100.00	0 0.00	8 100.00	
3	27 90.00	3 10.00	30 100.00	
4	9 50.00	9 50.00	18 100.00	
5	2 18.18	9 81.82	11 100.00	
Total	48 69.57	21 30.43	69 100.00	

出力結果から、修理記録では外国製の車の方が国内製の物よりも全般的に良いことが分かります。ダイアログには他の仮説検定のコマンドがありますが、この場では省きます。

構文メモ：結果ウィンドウの表示から「`tabulate rep78 foreign, row`」をコマンドウィンドウに打ち込めば同じ表が出力できます。つまり、`tabulate` コマンドの後に変数を2つ入力すると二元表を作成します。`row` がオプションとしてあるのは、ダイアログで「行内の相対度数」を選択したからです。`row` オプションを使用することで `tabulate` コマンドをデフォルトから変更できます。

次に外国製と国内製の燃費を比較したいと思います。それぞれの記述統計量を見ることから始めましょう。`if` 条件を使用し、変数 `mpg` を `foreign` で分けてから `summarize` コマンドを実行します。

```
. summarize mpg if foreign==0
```

Variable	Obs	Mean	Std. dev.	Min	Max
mpg	52	19.82692	4.743297	12	34


```
. summarize mpg if foreign==1
```

Variable	Obs	Mean	Std. dev.	Min	Max
mpg	22	24.77273	6.611187	14	41

結果から外国製の車の方が燃費は良いようです。次にこの結果の検定を行きましょう。

構文メモ：相等性の検定には等号2個 (`==`) が必要です。等号2個はプログラミングを行った経験がある方は見覚えがあるかもしれませんが、等号2個を使う構文は、**Stata** 初心者によく見られるエラー原因の1つなので気を付けてください。相等性を“完全に等しい”(だから、等号2個で強調している)として考えるとタイピングのミスは少なくなります。

記述統計量を出力するには他に2つの方法があります。こちらのほうが操作としては簡単です。1つ目の方法は今説明した方法を1回の操作で行います。2つのサブセット (**Domestic** と **Foreign**) にそれぞれコマンドを実行します。メニューでは以下の手順で操作します。

1. 統計 > 要約/表/検定 > 要約と記述統計量 > 記述統計量 を選択して、リセットボタン  を押します。
2. 変数欄のドロップダウンリストから `mpg` を選びます。
3. (未選択ならば) オプション内の標準の表示を選択します。
4. `by/if/in` タブをクリックします。

5. グループごとにコマンドを実行する のチェックボックスにチェックを付けます。
6. グループ変数欄にリストから **foreign** 選ぶか、直接入力します。
7. 適用をクリックします。

先程の表と一致する結果が出力されます。この方法は、数値ではなく値ラベル (**Domestic** と **Foreign**) が使われているため、上記 2 つのコマンドより見やすくなっています。グループを分類する変数の値を考える必要なく表が作成できます。

```
. by foreign, sort: summarize mpg
```

```
-> foreign = Domestic
```

Variable	Obs	Mean	Std. dev.	Min	Max
mpg	52	19.82692	4.743297	12	34

```
-> foreign = Foreign
```

Variable	Obs	Mean	Std. dev.	Min	Max
mpg	22	24.77273	6.611187	14	41

構文メモ：この相等性に関するコマンドはこれまでのコマンドとは少し異なります。この構文には **by** プレフィックスという前置コマンドが含まれます。**by** プレフィックスは独自のオプションとして主に「**sort**」があり、類似するデータを隣り合わせた状態で概要にまとめることができます。この **by** プレフィックスはデータ操作の理解とサブ集団 (**subpopulation**) で作業する際に大切なポイントになります。必要であればメモを補い、詳細の確認は [\[U\] 11.1.2 by varlist](#) と [\[U\] 13.7 Explicit subscripting](#) をご覧ください。**Stata** には他にもコマンドに特殊効果を付与する前置コマンドがあります。詳しくは [\[U\] 11.1.10 Prefix commands](#) をご覧ください。

生産地ごとの車の燃費を表にして比較します。つまり、変数 **foreign** の一元表 (**foreign** 対 **domestic**) の中に燃費の記述統計量を組み込みます。メニューから 統計 > 要約/表/検定 > 度数分布以外の表 > 平均/標準偏差/度数 と操作し、ダイアログの変数 1 に **foreign** を、変数の要約を表示するに **mpg** を入力します。そして適用をクリックすると以下のような表を出力します。

結果の表から、外国製の車の平均燃費と国内製の車の平均燃費は異なると結論付けることができます。本来ならデータ分析を始める前にこの検定を行う方が良いでしょう。「`ttest mpg, by(foreign)`」コマンドは簡単なので覚えておくと便利です。不均一な分散の場合は異なる `t` 値を求めるオプションや自由度の近似計算を行うオプションがありますのでご自由にお試しください。

構文メモ：`by()` オプションは先程使用した `by` プレフィックスとは異なるものです。似たようなコンセプトを使用していますが、用法が違います。`by()` オプションは `t` 検定の専用オプションです。

1.6 記述統計量—相関行列

ここから路線を変更してカテゴリー間の関係から数値間の方に焦点をあてます。たとえば燃費と車重に相関があるか調べてみます。メニューから、統計 > 要約/表/検定 > 要約と記述統計量 > 相関と共分散 を選択しま

す。`mpg` と `weight` を入力またはリストから選択し、適用をクリックします。結果ウィンドウに `mpg` と `weight` の相関行列を表示します。

```
. correlate mpg weight
(obs=74)
```

	mpg	weight
mpg	1.0000	
weight	-0.8072	1.0000

コマンド入力の場合は「`correlate mpg weight`」です。相関は負の相関を示しています。これは重い車ほど多くの力を必要とするので、納得できる結果です。

では国内製と外国製の車で燃費と車重の相関を比較するために、今までに学んだ `by` プレフィックスの知識を使います。`correlate` ダイアログをアクティブにします。閉じた場合は先程と同じようにダイアログを開きます。

`by/if/in` タブをクリックし、グループごとにコマンドを実行する のチェックボックスにチェックをつけ、グループ変数に `foreign` を入力します。適用を押すと `Domestic` と `Foreign` に分かれた相関を表示します。記

述統計量セクションで使用した「by foreign, sort:」を「correlate mpg weight」コマンドの前に入力しても同じものが出力されます。

```
. by foreign, sort: correlate mpg weight
```

```
-> foreign = Domestic
(obs=52)
```

	mpg	weight
mpg	1.0000	
weight	-0.8759	1.0000

```
-> foreign = Foreign
(obs=22)
```

	mpg	weight
mpg	1.0000	
weight	-0.6829	1.0000

結果の表より、国内製 (Domestic) のほうが強い相関があることが分かります。

構文メモ：この例では `correlate` コマンドを使用して 2 つの変数の相関を確認しました。Stata は任意の変数の数で相関行列を作成します。たとえば、5 つの変数を使用すると以下のような出力になります。

```
. correlate mpg weight length turn displacement
(obs=74)
```

	mpg	weight	length	turn	displacement
mpg	1.0000				
weight	-0.8072	1.0000			
length	-0.7958	0.9460	1.0000		
turn	-0.7192	0.8574	0.8643	1.0000	
displacement	-0.7056	0.8949	0.8351	0.7768	1.0000

これは、説明変数 (predictor variable) の共線性を調査する時などに役立ちます。

1.7 データの作図

今までの作業から分かったことがいくつかあります。まず国内製と外国製の車では平均燃費 (MPG) が異なります。修理記録も異なることが分かりました。最後に燃費と車重で負の相関を予想通り見つけ、国内製の方がより強く相関していました。

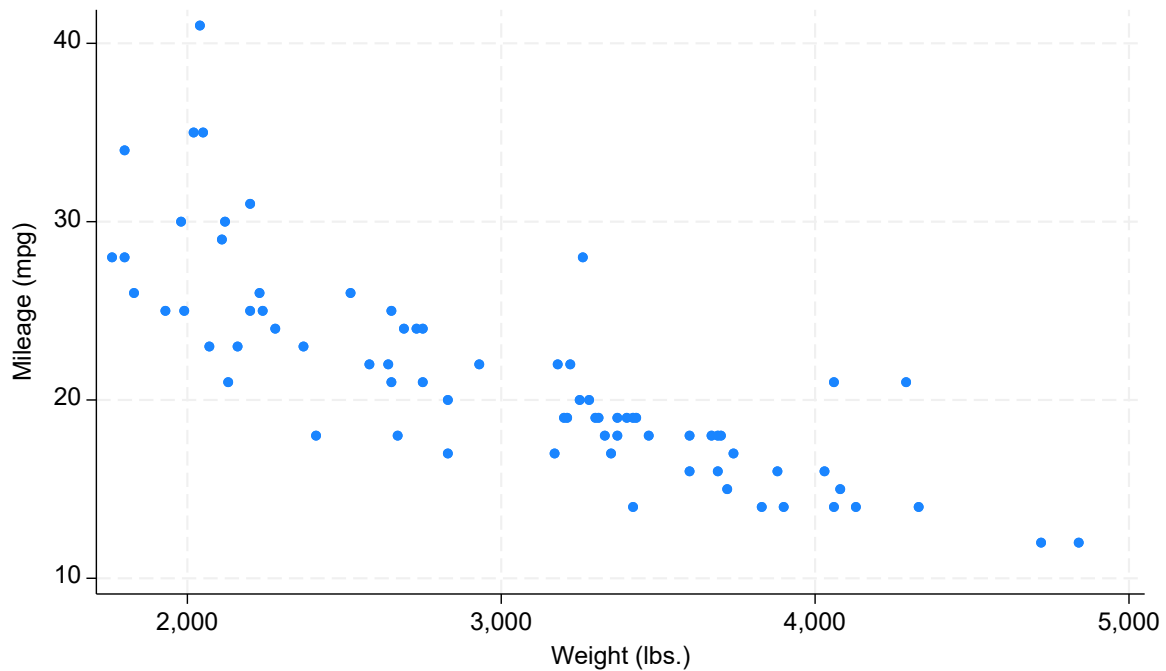
これから回帰モデル作成を見据えて燃費 (MPG) と車重 (weight) について確認していきます。まずは相関グラフを作図しましょう。mpg 対 weight の散布図から始めます。コマンドを使用して作図するには単純に「scatter mpg weight」と入力します。グラフをカスタマイズする場合はメニューを使って次のように操作します。

1. グラフィックス > 二元グラフ (散布図/折れ線など) を選択します。
2. 作成... ボタンをクリックします。
3. プロットカテゴリとタイプを選択する の枠にある、基本的なグラフのラジオボタンを選択します。
(未選択の場合)
4. 基本的なグラフ:(タイプを選択) の中からマーカー (散布図) を選択します。(未選択の場合)
5. プロットタイプ:(散布図) 枠の y 変数に mpg を、x 変数に weight をそれぞれドロップダウンリストから選択します。
6. 適用ボタンをクリックします。


メニューで実行した操作のコマンドを結果ウィンドウに表示します。

```
. twoway (scatter mpg weight)
```

実行したコマンドは初めに紹介したコマンドより少し複雑です。複雑になるには理由があり、複雑なコマンドのほうがグラフの統合やグラフの重ね合わせも行えるからです。これから実際に操作する中で確認してください。それでは作成したグラフを見てみましょう。



グラフから、mpg と weight には非線形かつ負の相関(右下がりの分布)があると分かります。

メモ：グラフを作成すると、結果ウィンドウの上に **Graph** ウィンドウが表示されます。**Stata** のメインウィンドウをクリックすると結果ウィンドウを最前面に配置します。グラフをもう一度確認したい場合は、グラフウィンドウを前面にボタン  をクリックすると、再び **Graph** ウィンドウが最前面になります。グラフウィンドウを前面にボタンについての詳細は [\[GSW\] 14 Graphing data \(データを作成する\)](#) をご覧ください。

国内製と外国製、それぞれの相関関係がどのように異なるのか散布図で見てください。それぞれのカテゴリーの散布図と全体の散布図を同時に表示します。

構文メモ：現在サブグループを見ているので、**by** プレフィックスで作図できそうです。実際に試してみましょう。

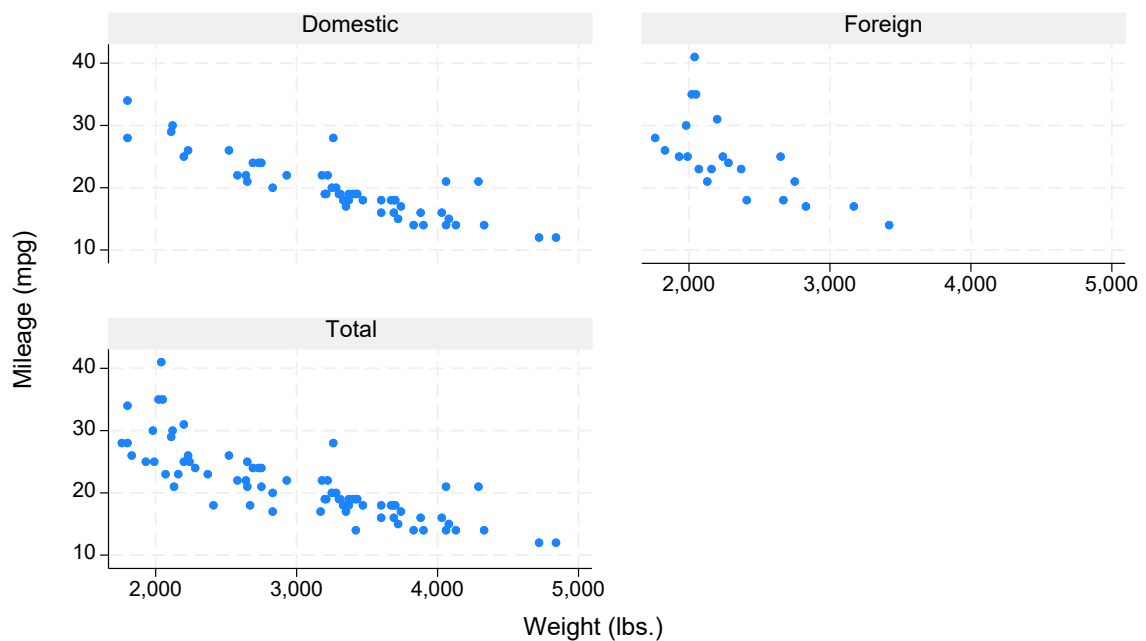
先程と同じように操作します。

1. グラフィックス > 二元グラフ (散布図/折れ線など) をメニューから選択します。
2. プロット 1 のダイアログ (先程グラフを作成したダイアログ) がまだ開いている場合、OK ボタンをクリックしてステップ 4 から操作してください。

3. 前のページに示した手順にしたがいグラフを作成します。
4. `twoway` - 二元グラフダイアログにある `by` 条件タブをクリックします。
5. 変数のユニーク値ごとのサブグラフを作成する のチェックボックスにチェックを付けます。
6. 変数欄に `foreign` を入力します。
7. 合計を含むグラフを追加する のチェックボックスにチェックを付けます。
8. 適用ボタンをクリックします。

作成したコマンドとグラフは次の通りです。

```
. twoway (scatter mpg weight), by(foreign, total)
```



Graphs by Car origin

どちらのカテゴリーも非線形な関係が成り立っている事が分かります。

構文メモ：サブグループごとのグラフを統合するとき (統合グラフ)、`by` プレフィックスではなく `by()` オプションを使用しました。`by` プレフィックスを使用すると、統合グラフではなく別々のグラフを作成します。

1.8 フィットモデル：線形回帰

グラフで特徴をつかんだので、車重とカテゴリーで燃費を予測する回帰モデルを作成します。変数の関係は非線形だと分かります。よって、車重の二次式として燃費をモデリングしてみます。**Domestic**と**Foreign**からは燃費と車重の関係が若干異なることがわかります。ダミー変数として **foreign** を加え、後でこの変数が正しく違いを表しているのか確認します。では、次のモデルをフィットしてみましょう。

$$\text{mpg} = \beta\beta_0 + \beta\beta_1\text{weight} + \beta\beta_2\text{weight}^2 + \beta\beta_3\text{foreign} + \epsilon\epsilon$$

foreign は既にダミー変数 (0 か 1) ですが、**weight** の二乗値を作成する必要があります。メニュー操作でも新しい変数を作成できますが、コマンド入力の方が簡単です。次のようにコマンドウィンドウに入力しましょう。

```
. generate wtsq = weight^2
```

必要な変数がすべて揃ったので、線形回帰を行います。メニュー操作で実行し、後でコマンドを確認しましょう。統計 > 線形モデル他 > 線形回帰 をメニューから選びます。ダイアログの従属変数欄に **mpg** を、独立変数欄に **weight**, **wtsq**, **foreign** を入力し、適用をクリックします。**regress** のコマンド文と、分散分析表が表示されます。

```
. regress mpg weight wtsq foreign
```

Source	SS	df	MS	Number of obs	=	74
Model	1689.15372	3	563.05124	F(3, 70)	=	52.25
Residual	754.30574	70	10.7757963	Prob > F	=	0.0000
Total	2443.45946	73	33.4720474	R-squared	=	0.6913
				Adj R-squared	=	0.6781
				Root MSE	=	3.2827

mpg	Coefficient	Std. err.	t	P> t	[95% conf. interval]
weight	-.0165729	.0039692	-4.18	0.000	-.0244892 - .0086567
wtsq	1.59e-06	6.25e-07	2.55	0.013	3.45e-07 2.84e-06
foreign	-2.2035	1.059246	-2.08	0.041	-4.3161 -.0909002
_cons	56.53884	6.197383	9.12	0.000	44.17855 68.89913

出力に問題は無さそうなので、車のカテゴリーごとの散布図上に予測値を作図しましょう。そのためには予測、またはフィットした値が必要です。これはメニュー操作で実行できますが、簡単なのでコマンドウィ

ンドウに入力しましょう。まずは予測値を格納する新規変数、`mpghat` を作りましょう。次のコマンドを入力します。

```
. predict mpghat
(option xb assumed; fitted values)
```

このコマンドを入力するとメッセージが 1 行だけ表示されます。画面右上にある変数ウィンドウを下までスクロールすると変数 `mpghat` が確認できます。`mpghat` が作成された状態でこのコマンドをもう一度実行しても、既存データは上書きされません。


```
. predict mpghat
variable mpghat already defined
r(110);
```

`predict` のように、回帰実行後に続けて利用するコマンドのことをポスト推定 (**postestimation**) コマンドと呼びます。新しい変数 `mpghat` には次の数式で計算した値が入ります。

$$-0.0165729\text{weight} + 1.59 \times 10^{-6}\text{wtsq} - 2.2035\text{foreign} + 56.53884$$

モデルの推定後には予測値の計算はもちろん、モデルを改良するための様々な機能が利用できるようになります。詳しくは [\[U\] 20 Estimation and postestimation commands](#) をご覧ください。

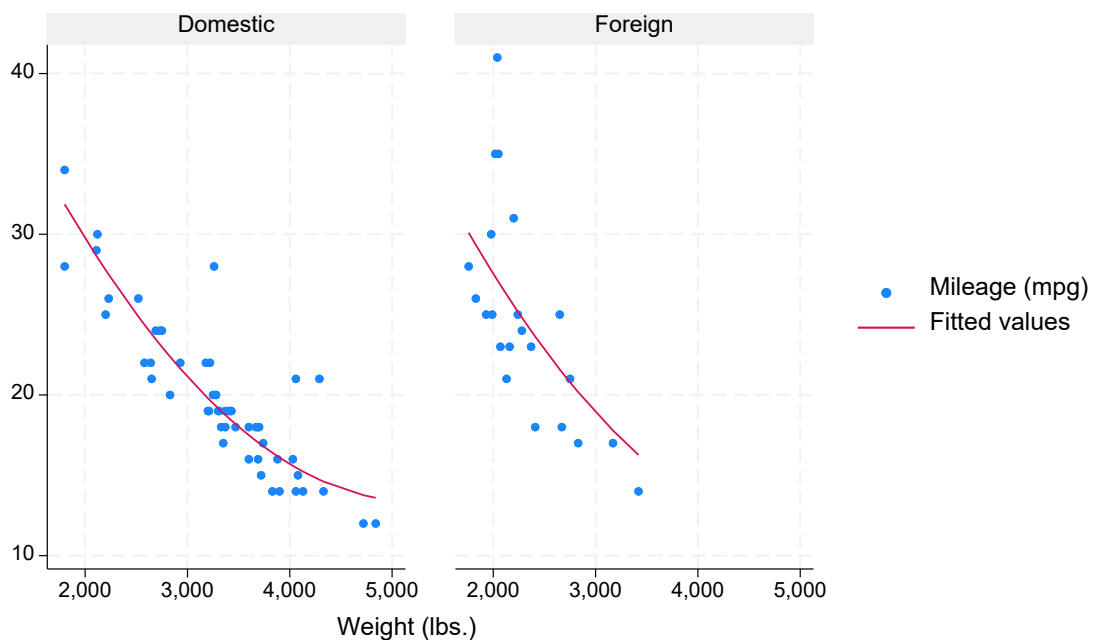
では、国内製と外国製のグラフの上に予測値を書き込んでダミー変数の適切さを確認しましょう。データと予測曲線を同じグラフ上に作図し、適切なシフトパラメータの判断をします。両方のグラフを一度に作図できるので、実際にやってみましょう。メニューとダイアログを使うには以下の手順で操作します。

1. グラフィックス > 二元グラフ (散布図/折れ線など) をメニューから選択します。
2. プロットの定義に他のプロットが残っている場合、リセットボタン  をクリックします。
3. `mpg` 対 `weight` のグラフを作成します：
 - (a) 作成... ボタンをクリックし、プロット 1 ダイアログを開きます。
 - (b) 基本的なグラフとマーカー (散布図) が選択されていることを確認します。

- (c) プロットタイプの y 変数に **mpg** を、x 変数に **weight** をそれぞれ選択します。
- (d) OK をクリックします。
4. **mpghat** 対 **weight** のグラフを作成します：
- (a) 作成... ボタンをクリックします。
- (b) 基本的なグラフと線を選択します。
- (c) プロットタイプの y 変数に **mpghat** を、x 変数に **weight** を選択します。
- (d) x 変数でソートのチェックボックスをチェックします。これでデータ内の順番ではなく、**weight** の昇順で直線を作成します。
- (e) **OK** をクリックします。
5. 2つのプロット、国内製と外国製を同じグラフ内に表示します：
- (a) **by** 条件タブをクリックします。
- (b) 変数のユニーク値ごとのサブグラフを作成するのチェックボックスにチェックを付けます。
- (c) 変数欄に **foreign** を入力します。
6. 適用ボタンをクリックします。

コマンド文とグラフは次のようになります。

```
. twoway (scatter mpg weight) (line mpghat weight, sort), by(foreign)
```



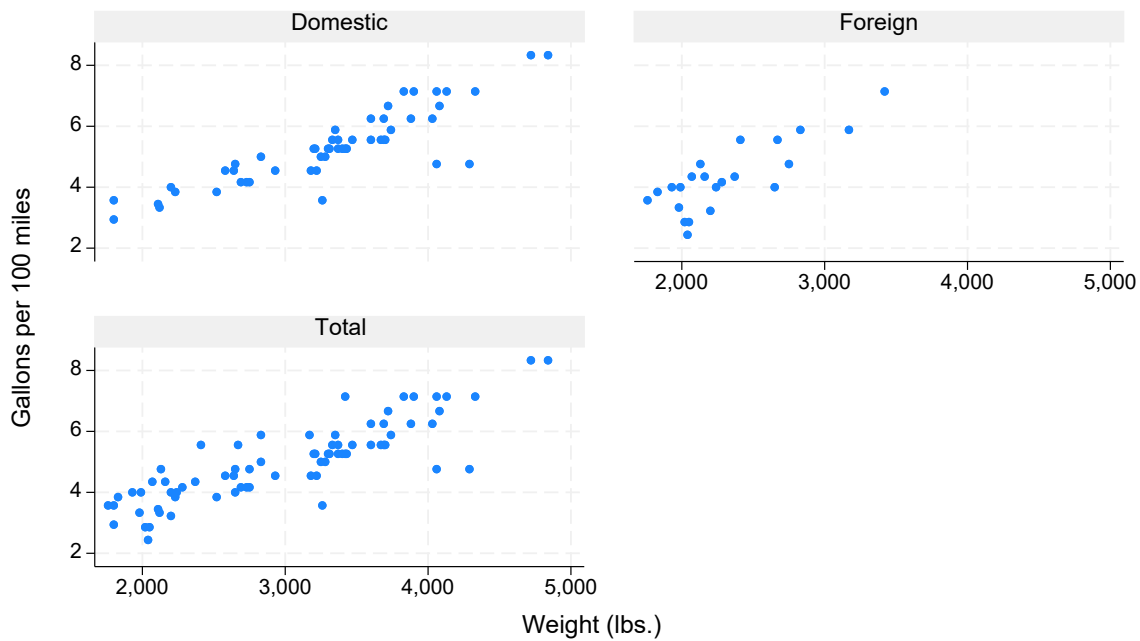
この例から、`scatter` と `line` のコマンドを別々のカッコに入れて同時に実行すると、重ね書きできることが分かります。このように、散布図と曲線を重ねる場合はカッコを使用してください。このグラフは良くフィットしています。

良いグラフができたので、技術者である友人にこのグラフを見てもらうことにしましょう。今までの結果ではなくグラフだけを印刷するには、**Graph** ウィンドウ内で `ファイル > 印刷...` と操作して印刷します。

印刷したグラフを友人に見せたところ、どうやら間違いがあるようです。「違う。」と言われてしまいました。「**2,000** ポンド(約 **900kg**) を **1** マイル (約 **1.6km**) 動かすのには **1,000** ポンド (約 **450kg**) を同じ距離動かすときの約 **2** 倍のエネルギーが必要となる。つまり、ガソリンの消費量も比例し、**2** 倍の量を消費するはず。 **mile/gallon** は重さの二次式でなく、一次式になるはず。」とのことです。

友人が言ったことを検証してみましょう。まずは距離単位毎のエネルギー (**gallon/mile**) 変数を作成し、散布図を作図します。以下が必要なコマンドです。このコマンドはセッション内で使用したものに似ています。この中に初めて見るコマンドが **1** つあります。「`label variable`」コマンドは変数 `gp100m` に変数ラベルを設定します。結果として次のようなグラフを作成します。

```
. generate gp100m = 100/mpg
. label variable gp100m "Gallons per 100 miles"
. twoway (scatter gp100m weight), by(foreign, total)
```



友人が正しいという結論が出たところで、回帰をやり直してみましょう。

```
. regress gp100m weight foreign
```

Source	SS	df	MS	Number of obs	=	74
Model	91.1761694	2	45.5880847	F(2, 71)	=	113.97
Residual	28.4000913	71	.400001287	Prob > F	=	0.0000
Total	119.576261	73	1.63803097	R-squared	=	0.7625
				Adj R-squared	=	0.7558
				Root MSE	=	.63246

gp100m	Coefficient	Std. err.	t	P> t	[95% conf. interval]
weight	.0016254	.0001183	13.74	0.000	.0013896 .0018612
foreign	.6220535	.1997381	3.11	0.003	.2237871 1.02032
_cons	-.0734839	.4019932	-0.18	0.855	-.8750354 .7280677

記述統計量のセクションで1978年代の外国製の車の燃費が国内製の物より良かった理由は軽かったからだ、ということが分かります。このモデルによると、国内製の車と同じ重さの外国製の車は100マイル（約160km）あたり、追加で8分の5ガロン（約2.35L）のガソリンを消費することになります。以上で、この一連の分析を終了します。

1.9 コマンドとメニューの違い

今までのセッションで **Stata** はメニュー操作とコマンドウィンドウのどちらからでも操作できることが分かりました。慣れてきたら、よく使用するコマンドはコマンドウィンドウで素早く実行し、グラフを作成するような複雑な操作はメニューとダイアログを利用すると効率的です。

Stata のコマンド構文 (command syntax) には一貫性があります。基本的にコマンドは次の構文を使用します。[] 内の項目はオプションであり、変数名は **varlist** に入力します。

[prefix:] command [varlist] [if] [in] [weight] [, options]

一般的なルール：

- ほとんどのコマンドでは機能を変化させる前置コマンドを併せて利用できます。詳しくは [\[U\] 11.1.10 Prefix commands](#) をご覧ください。頻繁に使われる前置コマンドは **by** プレフィックスです。
- **varlist** の指定がない場合、全ての変数を使用します。
- **if** と **in** はコマンドの実行対象となるデータに条件を付加します。
- **options** (オプション) はコマンドの機能を修正します。
- 各コマンドの構文はシステムヘルプとマニュアルで確認できます。
- **Stata** のコマンド構文はここで紹介した以上に多くのものが存在しますが、とりあえずこれまでの知識で使い始めてみましょう。詳しくは [\[U\] 11 Language syntax](#) と「**help language**」コマンドをご覧ください。

in と **weight** 以外のコマンドは、この節で既に使用してきました。システムヘルプにはすべてのコマンド構文と例題が載っています。詳しくは [\[GSW\] 4 Getting help \(ヘルプ・ヒントを見つける\)](#) をご覧ください。文法は基本的に同じですから、新しいコマンドの用法もすぐに分かりますし、他の研究者の分析内容を理解して読み解く際にも便利です。

以前使用した **summarize** コマンドをもとに構文を読んでみましょう。**summarize** の構文は **Stata** コマンドの典型例です。

`summarize [varlist] [if] [in] [weight] [, options]`

以下の内容を読み取ることができます。

コマンドのみ :	<code>summarize</code>
コマンドと varlist (変数リスト) :	<code>summarize mpg</code> <code>summarize mpg weight</code>
コマンド (と変数リスト) と if 条件文 :	<code>summarizeif mpg>20</code> <code>summarize mpg weight if mpg>20</code>


など

`summarize` の詳細は [\[R\] summarize](#) またはヘルプ > Stata コマンド... と選択してから `summarize` と入力します。

1.10 作業内容を記録する

作業記録 (ログ) を取るとそれまでの結果を見返し、変更内容を確認できるので便利です。ログの取り方は

[\[GSW\] 16 Saving and printing results by using logs \(ログを使い結果の保存や印刷を行う\)](#) で説明します。ログにはコマンドと出力結果が含まれるので、コマンド構文を学んでおけば自分が実行したコマンドをすぐに思い出すことができます。

結果ウィンドウが表示する内容を全てログとして記録するには、ノートのように見えるログボタン  をクリックします。普通のファイルと同じように、このログファイルを保存する場所を選び、ファイルに名前を付けます。ログ記録 (ロギング) を始めてから終えるまでの間、結果ウィンドウに表示されたすべてのものをログファイルは保存します。

1.11 まとめ

この章では **Stata** の機能を簡単に紹介しました。このままマニュアルを読み進み、作業を続けてください。このマニュアルを一通り読み終えた上で、**User's Guide** をご参照ください。

2. Stata のユーザインターフェイス

2.1 ウィンドウ

The screenshot shows the Stata 17.0 main window with the following panes and callouts:

- 履歴ウィンドウ (History Window):** 実行した操作の履歴を表示 (Display history of operations performed).
- 結果ウィンドウ (Results Window):** 実行した操作の結果を表示 (Display results of operations performed).
- コマンドウィンドウ (Command Window):** コマンドを入力 (Enter commands).
- 変数ウィンドウ (Variable Window):** 使用できる変数の一覧を表示 (Display a list of available variables).
- プロパティウィンドウ (Properties Window):** 変数ウィンドウで選択した変数の詳細を表示 (Display details of the variable selected in the variable window).

Additional callouts in the image include:

- 現在の作業フォルダ (Current working folder) pointing to the command window.
- 現在のログ状況 (Current log status) pointing to the command window.
- 現在のコマンドログ状況 (Current command log status) pointing to the command window.

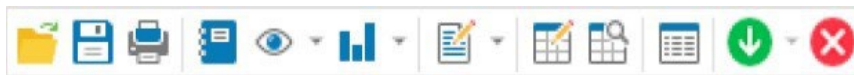
Stata のメインウィンドウは履歴、結果、コマンド、変数、プロパティの 5つのウィンドウで構成されます。結果ウィンドウ以外は各ウィンドウ独自の名前が上部のタイトルバーに表れます。この 5つのウィンドウは、基本的に Stata の使用中は常に開いています。これらとは別に、ビューワ、データエディタ、変数マネージャ、do ファイルエディタ、Graph、グラフエディタという、用途ごとに特化したウィンドウがあります。詳しくはマニュアルの後半に記します。

ウィンドウを開くまたは他ウィンドウの背面にあるウィンドウを表示するにはウィンドウメニューから対応するウィンドウを選択します。あるいはツールバーで対応するアイコンをクリックします。キーボード操作では、**Ctrl+Tab** キーを押すとメインウィンドウ内のウィンドウを順に巡ることができます。また、**Alt+Tab** キーで Stata 以外のウィンドウを含む、すべてのウィンドウを巡回します。Stata の多くのウィンドウ内で右クリックを行うとコンテキストメニューで利用できる機能が表示されます。ウィンドウによってコンテキストメニューの内容は異なりますが、主にテキストのコピー、ウィンドウの設定変更、ウィンドウの

印刷などのコマンドがあります。テキストのコピーや印刷を行う場合、ミス回避するためにもメニューバーよりも右クリックを利用することをお勧めします。

2.2 ツールバー

メインツールバーを次に示します。



ツールバーには頻繁に利用するコマンドがボタンとして配置されています。ボタンが何の機能なのか忘れてしまった場合、マウスカーソルをボタン上に移動するとヒントが表示されます。

矢印の付いたボタンでは、矢印をクリックするとさらに小さなメニューを表示します。ツールバーボタンと機能の概要は次の通りです。

	開く	Stata のデータセットを開きます。ボタンをクリックすると開くダイアログを表示します。
	保存	メモリ内にある現在の Stata データセットを保存します。
	印刷	印刷したいウィンドウのリストを表示します。ウィンドウの名前を選んでください。
	ログ	新しいログの開始、現在のログの終了、中断、再開、のいずれかを選択して実行します。ログファイルについては [GSW] 16 Saving and printing results by using logs (ログを使い結果の保存や印刷を行う) をご覧ください。
	ビューワ	ビューワウィンドウを新たに開く、または既に開いているウィンドウを最前面にします。ボタンをクリックすると新規ビューワを開き、隣の矢印でビューワを選択すると最前面にします。詳しくは [GSW] 3 Using the Viewer (ビューワを使う) をご覧ください。

	グラフ	Graph ウィンドウを最前面にします。ボタンをクリックすると直近に選択したグラフを、隣の矢印をクリックすると選択したグラフを最前面に表示します。詳しくは [GSW] 14 Graphing data (データを作図する) 内にあるグラフボタンをご覧ください。
	do ファイルエディタ	do ファイルエディタウィンドウを新たに開く、または既に開いているウィンドウを最前面にします。ボタンをクリックすると新規 do ファイルエディタが開き、隣の矢印をクリックすると、既に開いているウィンドウを選択して最前面にします。詳しくは [GSW] 13 Using the Do-file Editor—automating Stata (do ファイルエディタを使用する—Stata の自動化) をご覧ください。
	データエディタ (編集)	データエディタウィンドウを開くか、既に開いているウィンドウを最前面にします。詳しくは [GSW] 6 Using the Data Editor (データエディタを使用する) をご覧ください。
	データエディタ (ブラウズ)	データエディタをブラウズモードで開きます。詳しくは [GSW] 6 Using the Data Editor (データエディタを使用する) 内のブラウズモードをご覧ください。
	変数マネージャ	変数マネージャを開きます。詳しくは [GSW] 7 Using the Variables Manager (変数マネージャを使用する) をご覧ください。
	より多くの結果を表示	長い出力表示の途中で一時停止した時に次の画面を表示します。隣の矢印をクリックしてコマンドを最後まで実行させることもできます。詳しくは [GSW] B.8 More をご覧ください。
	中断	実行中のタスクを中止します。詳しくは [GSW] 10 Listing data and basic command syntax (データのリストと基本コマンドの構文) をご覧ください。

2.3 コマンドウィンドウ

Stata はコマンドウィンドウに入力したコマンドを実行します。コマンドウィンドウでは基本的なテキスト編集、コピーと貼り付け、コマンド履歴、ファンクションキーのマッピング、変数名コンプリート (**variable-name completion**) をサポートしています。

コマンドウィンドウをアクティブにして次のように操作してみましょう。

Page Up	コマンド履歴を遡ります
Page Down	コマンド履歴を進めます
Tab	途中まで入力した変数名の続きを自動で補完するか、または複数の候補があるときは候補の一覧を表示します。さらに入力続けて候補を狭めることができます。入力を二重引用符 (") で始めたときはファイル名についての自動補完を行います。

コマンドウィンドウ用のキーボードショートカットについては [\[U\] 10 Keyboard use](#) をご覧ください。

コマンド履歴は同一セッション内で実行したコマンドを振り返るのに有効で、編集を加えて再実行もできます。

Stata のダイアログで実行されたコマンドについてもコマンド履歴に含まれているので、ダイアログを再び開くことなくコマンドを振り返り、必要に応じて再実行できます。

2.4 結果ウィンドウ

結果ウィンドウはセッション中のすべてのコマンドとテキスト形式の結果を表示します。結果ウィンドウをスクロールして操作やコマンドを確認するより、結果ウィンドウの検索バーを利用の方が簡単に操作できます。デフォルトでは検索バーは非表示になっており、表示するには **編集 > 検索** と操作します。

結果ウィンドウのウィンドウ内で右クリックをしてコンテキストメニューから結果のクリアを選択すると、結果ウィンドウのコンテンツを削除できます。このアクションはやり直しができません。

2.5 変数ウィンドウ

変数ウィンドウは設定したプロパティとデータセット内の変数をリスト形式で表示します。デフォルトではすべての変数と変数ラベルを表示します。変数ウィンドウでヘッダーを右クリックすると、プロパティの表示を選択できます。

変数ウィンドウ内で一度クリックすると変数を選択できます。複数の変数を選択する場合、それが隣り合っていない時は **Ctrl** キー + マウスクリックを使い、隣り合っている時は **Shift** キー + マウスクリックを使って一度に選択します。

変数ウィンドウで変数をダブルクリックすると、コマンドウィンドウのカーソル位置にその変数を表示します。

変数ウィンドウの左端の列を「ワンクリック貼り付け列」と呼びます。ワンクリック貼り付け列の上にマウスを移動し、矢印が出たらクリックします。この方法でもコマンドウィンドウに変数名を入力できます。変数ウィンドウの他の列と同じように、ワンクリック貼り付け列も表示・非表示を切り替えることができます。

変数ウィンドウは変数の並び替えやフィルタリングの機能をサポートしています。変数名フィルタを入力した欄に入力したテキストで変数ウィンドウ内の変数にフィルタをかけることができます。フィルタは表示されている列全てに適用され、**1** つでも条件に当てはまる変数を抽出します。デフォルトでは大文字・小文字を区別しません。エリア左側にあるレンチをクリックするとこの設定条件を変更できます。さらに変数に関する他の情報の挿入・削除も可能です。

変数ウィンドウのヘッダーをクリックすると変数の並び替えができます。**1** 回クリックで昇順、**2** 回クリックで降順、そして **3** 回クリックすると元のデータセット順に並べます。変数ヘッダーにある項目をクリックすると変数ウィンドウの表示のみがその項目を使ってアルファベット順になります。変数ウィンドウの並び替えはリアルタイムに行われます。よって、変数ウィンドウをあるプロパティで並び替えた後に変更を加え

ると、自動的に並び順を更新します。変数ウィンドウの変数の表示順を並べ替えても、データセット内の順番は変わりません。

変数の上で右クリックするとコンテキストメニューが表示されます。

- 選択した変数のみ維持 選択した変数だけをデータセットのメモリに残します。この操作はユーザーに承認を求めます。メモリ内のデータセットのみ変更するので、ディスクに保存したデータセットには影響しません。詳しくは [\[GSW\] 12 Deleting variables and observations](#) をご覧ください。
- 選択した変数を削除 データセットメモリから、選択した変数を削除します。この操作はユーザーに承認を求めます。上記のように、これはメモリ内のデータセットのみ変更するので、ディスクに保存したデータセットには影響しません。詳しくは [\[GSW\] 12 Deleting variables and observations](#) をご覧ください。
- 変数リストをコピー 選択した変数名をクリップボードにコピーします。
- すべて選択 フィルタの条件に当てはまる全ての変数を選択します。フィルタの指定がない場合、すべての変数を選択します。
- コマンドウィンドウに変数リストを送る 選択した変数を全てコマンドウィンドウに表示します。
- フォント... フォントダイアログを開きます。変数ウィンドウの表示フォントを変更します。

コンテキストメニューの項目は通常の **Stata** コマンドと同じです。つまり、右クリックから操作を行うのはコマンドウィンドウに直接コマンドを入力するのと同じです。

変数ウィンドウを非表示にする時はウィンドウの 閉じる ボタンをクリックします。

非表示の変数ウィンドウを再び表示するには、ウィンドウ > 変数ウィンドウ と操作します。

2.6 プロパティウィンドウ

プロパティウィンドウは変数とデータセットのプロパティを表示します。変数ウィンドウで変数を選択すると、そのプロパティを表示します。変数ウィンドウで複数の変数を選択する場合、その選択したすべての変数に共通するプロパティをプロパティウィンドウに表示します。


プロパティウィンドウのタイトルバーにあるロックアイコンをクリックすると、選択した変数のプロパティを編集できます。デフォルトでは編集できません。プロパティのロックを解除すると、変数またはデータセットのプロパティを自由に編集できます。編集操作はコマンドとして結果と履歴ウィンドウに直接表示されます。当然コマンドログにも記録します。メモ(Notes)の管理、変数と値ラベルの変更、表示形式も変更を行うにはプロパティウィンドウを使用するのが最も簡単です。詳しくは [\[D\] notes](#)、[\[D\] label](#)、[\[D\] format](#) をご覧ください。


ロックアイコンの隣にある矢印ボタンをクリックすると、変数ウィンドウ内の上または下にある変数が選択され、それらのプロパティを表示します。プロパティウィンドウを非表示にするにはウィンドウの閉じるボタンをクリックします。再びプロパティウィンドウを表示するには、ウィンドウ > プロパティ と選択してください。

より細かく変数の管理を行う場合は、優れたユーザインターフェイスを備えている変数マネージャを利用してください。変数マネージャの詳細は [\[GSW\] 7 Using the Variables Manager \(変数マネージャを使用する\)](#) を参照してください。

2.7 履歴ウィンドウ

履歴ウィンドウは入力したコマンドの履歴を表示します。デフォルトでは、成功したコマンドは黒で、失敗したコマンドはエラーコードと共に赤で表示します。

ツールバーには履歴ウィンドウの表示を変更するツールが2つあり、フィルタボタン  を押すと、履歴ウィンドウのタイトルバーの表示方法を変更します。コマンドフィルタを入力した欄に入力した文字列で履歴ウィンドウにフィルタをかけます。デフォルトでは大文字・小文字を区別せずにテキストを検索します。デフォ

ルトの設定を変更するときはエリア左側にあるレンチマークをクリックします。エリア右側の感嘆符  をクリックすると、エラーがあったコマンドが非表示になります。

これらのツールを使用してもコマンドの履歴が削除される訳ではありません。ただ表示に影響を与えるだけです。

履歴ウィンドウからコマンドを入力する時は次のようにします。

- リスト上のコマンドを **1** 回クリックすると、再びコマンドウィンドウに表示します。
- リスト上のコマンドをダブルクリックすると再実行します。再実行したコマンドは履歴ウィンドウのコマンド履歴に再度登録されます。

履歴ウィンドウで右クリックをすると、コンテキストメニューを表示します。

- **切り取り**： 選択したコマンドを履歴ウィンドウから切り取り、クリップボードに移します。
- **コピー**： 選択したコマンドをクリップボードにコピーします。
- **削除**： 選択したコマンドを履歴ウィンドウから削除します。
- **すべて選択**： 履歴ウィンドウ内のコマンドをすべて選択します。 ● **すべてクリア** 履歴ウィンドウ内のコマンドをすべて削除します。
- **選択範囲を実行**： 選択したすべてのコマンドを再実行し、コマンド履歴に追加します。エラーコマンドも含め、選択したコマンドはすべて再実行します。エラーが発生しても途中で止まりません。
- **選択範囲を do ファイルエディタへ送る**： 選択したすべてのコマンドを **do** ファイルエディタウィンドウに表示します。
- **すべて保存...**： 履歴の保存ダイアログを開き、履歴ウィンドウにあるすべてのコマンドを **do** ファイルとして保存します。(do ファイルに関する詳細は [\[GSW\] 13 Using the Do-file Editor— automating Stata \(do ファイルエディタを使用する—Stata の自動化\)](#) をご覧ください。
- **選択範囲を保存...**： 履歴の保存ダイアログを開きます。履歴ウィンドウで選択したコマンドを **do** ファイルに保存します。

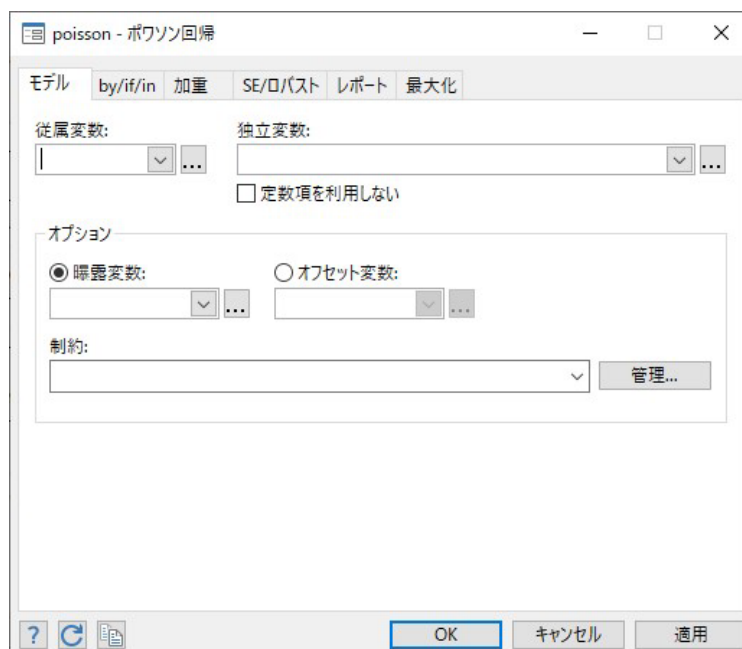
- **フォント...** : フォントダイアログを開きます。これで履歴ウィンドウ内の内容を表示するフォントを変更します。

2.8 メニューとダイアログ

Stata を操作する方法は 2 通りあります。1 つ目はメニューを使用する方法で、2 つ目はコマンドウィンドウを使用する方法です。[GSW] 1 **Introducing Stata—sample session (Stata の紹介—サンプルセッション)** の中にあるサンプルセッションを操作して分かるように、どちらの方法にも便利なところはありました。このセッションではメニューとダイアログについてより詳しく紹介をします。

データ、グラフィックス、統計のメニューに **Stata** のほぼすべてのコマンドがあります。**Stata** はプログラミング機能を備えているので、ユーザは独自のダイアログやメニューを作成できます。作成したメニュー項目はユーザメニューに入ります。初期状態では空欄のサブメニューがあるだけです。例として、ポアソン回帰を行う場合を紹介します。

Stata の「**poisson**」コマンドを入力するか、統計 > アウトカム (カウント) > ポアソン回帰 とメニュー操作をすると、次のダイアログを表示します。

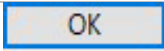

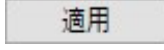





このダイアログは `poisson` コマンドで実行できるすべての機能を提供します。従属および独立変数は数値である必要があるため、数値が入力されている変数のみがコンボボックスで選択可能になります。ダイアログのタブを変更することで `poisson` コマンドが備える数多くのオプションを選ぶことができます。選択したコマンドのダイアログを見ると、機能の概略が分かります。

多くのダイアログには `by/if/in` と加重タブがあります。推定に利用する標本をコントロールしたり、データに重み付けを行うときに利用します。Stata のプログラミング言語に関する詳細は [\[U\] 11 Language syntax](#) をご覧ください。

推定を行うコマンドの多くには最大化タブがあり、そこでは最大化に関するオプションを設定します。[\(\[R\] Maximize をご覧ください。 \)](#) たとえば、最適化のための反復回数の上限を設定できます。

ほとんどのダイアログには上記 `poisson` ダイアログの下部にある 6 つのボタンと同じものがあります。

	OK	ダイアログの設定に応じてコマンドを実行します。実行後、ダイアログを閉じます。
	キャンセル	何もしないでダイアログを閉じます。右上にある赤い x 印のボタンと同じです。
	適用	OK と同じようにコマンドを実行しますが、ダイアログは開いたままです。そのままダイアログに変更を加え、再適用すると新たにコマンドを実行します。この機能は、不慣れなコマンドの利用や、複雑なグラフの作成中に役立ちます。
	ヘルプ	ヘルプシステムを起動します。ボタンをクリックするとそのダイアログに関連のあるヘルプファイルを表示します。画面上でこのボタンを押すと <code>poisson</code> のヘルプファイルにつながります。ヘルプファイルの <code>Options</code> セクションではタブごとにオプションを解説します。
	リセット	ダイアログをデフォルトの状態にリセットします。ダイアログは開く度に最後の状態を覚えています。ダイアログの画面をデフォルトに戻したい場合、このボタンをクリックしてください。
	コマンドをクリップボードにコピーする	適用ボタンのように機能しますが、コマンドを実行する代わりにクリップボードにコピーします。このコマンドは他のところ (例えば <code>do</code> ファイルエディタ) で使用できます。

ダイアログを通して入力したコマンドも手入力したものと全く同じです。コマンドは結果ウィンドウに表示され、実行後は履歴ウィンドウで確認できます。メニュー操作で実行したコマンドの出力を詳しく見るとで **Stata** のコマンド構文を学ぶことができます。

メニューからの操作のほか、**2**つの方法でダイアログを開くことができます。目的のコマンドがメニュー内のどこにあるのか忘れてしまったときには次の方法を試してください。コマンドウィンドウに「db コマンド名」の形で入力します。

```
. db poisson
```

また、ダイアログを開くためのメニューコマンドはヘルプファイルからも調べられます。詳しくは[\[GSW\] 4](#)

[Getting help \(ヘルプ・ヒントを見つける\)](#) をご覧ください。

このマニュアルを読み進めて行くと **Stata** コマンドを直接入力する例が出てきます。例に書かれている通りに入力してコマンドを実行します。しかし、ダイアログを使用して同じ操作を行うこともできるので **db** コマンドを試してみてください。**db** コマンドはコマンドに対応するダイアログを素早く開きます。

2.9 現在の作業フォルダ

「ウィンドウ」セクションのスクリーンショットを見ると、**Stata** のメインウィンドウ一番下にあるステータスバーに、**Stata** をインストールしたときに設定した作業フォルダ(**working directory**)を表示します。この例の中では **C:\¥Users¥Stata¥Documents¥Stata** となっていますが、ユーザのコンピュータではそれぞれ設定した作業フォルダを表示しているはずです。グラフやデータセットを「save ファイル名」のコマンドで保存すると、保存先のフォルダが作業フォルダになります。これはメニューから操作されているファイルのアクション、たとえば **ファイル> 保存** または **ファイル> 開く...** などには影響を与えません。**Stata** の起動後には現在の作業フォルダを **cd** コマンドで変更できます。詳細は[\[D\] cd](#) をご覧ください。メインウィンドウには常に現在の作業フォルダ名を表示するので、グラフやデータセットの保存先を簡単に確認できます。

2.10 Stata ウィンドウの繊細なコントロール

Stata を起動すると、履歴、結果、コマンド、変数、プロパティウィンドウをメインウィンドウ内に表示します。履歴、変数、プロパティ、コマンドウィンドウはそれぞれお互いにリンクしていて、メインウィンドウの端にあります。各ウィンドウの端をドラッグすると大きさを変更できます。更に、メインウィンドウ内でタイトルバーをドラッグして位置を変更したり、使用しないウィンドウをタイトルバーにある画鋏アイコンをクリックして非表示にすることもできます。

結果ウィンドウは **Stata** のメインウィンドウの残っている空間を使用します。メインウィンドウ以外のウィンドウは自由に移動できます。

このデフォルトの機能は他のソフトウェアでもよく見かける物なのでこれ以上は説明しません。メインウィンドウ内にあるウィンドウを自由に動かしてみましょう。ウィンドウの操作に慣れている場合は、これ以降のセクションを飛ばしてください。更にカスタマイズを行いたい時、または技術的な解説を読みたいときは、このまま読み進めてください。

2.11 ウィンドウの種類

Windows 版 **Stata** にはドッキングとノンドッキングウィンドウの **2** 種類のウィンドウタイプがあります。履歴、コマンド、変数、プロパティウィンドウはドッキングウィンドウです。他のウィンドウはノンドッキングウィンドウです。

ドッキングウィンドウには、他のドッキングウィンドウと連携して使用できる、という特性があります。具体的には、**1** つのウィンドウをスプリッタ (分割線) で分けて共有するリンク機能や、各ウィンドウをタブで表し **1** つのウィンドウを共有するタブ機能があります。また必要でない時は自動的に非表示にすることもできます。ノンドッキングウィンドウにはこのような特性はありません。

2 種類のウィンドウを簡単に比較すると次のようになります。

ドッキングウィンドウ

- 他のドッキングウィンドウにリンクでき、**1** つのウィンドウをスプリッタで共有できます
- 他のウィンドウと同じ枠を共有し、タブ化して切り替えることができます
- 使用していない時は自動的に非表示になるように設定できます

ノンドッキングウィンドウ

- **Stata** のメインウィンドウとは独立しています (ただし結果ウィンドウは除く)
- 他のウィンドウにリンクできません
- 他のウィンドウとタブ化でつなげることはできません
- 使用されていない時は自動的に非表示になるように設定できません

Stata を起動すると、履歴、結果、コマンド、変数、プロパティウィンドウをメインウィンドウ内に表示します。履歴、コマンド、変数、プロパティウィンドウはドッキングウィンドウで、最初はお互いに隣り合っている、リンクしています。結果ウィンドウはノンドッキングウィンドウです。

2.12 ドッキングウィンドウ

ドッキングウィンドウはタイトルバーをドラッグすることで、メインウィンドウ内を移動させることができます。Stata のメインウィンドウまたは他のドッキングウィンドウ上へこのボックスをドラッグするとドッキングガイドを表示します (図 1)。このガイド上にタイトルバーをドラッグしてくると、ドロップ時にウィンドウがどこにどのくらいの表示になるかをプレビューできます。

ボックスを外側のドッキングガイドに合わせてドロップすると、ウィンドウをガイド下のウィンドウにリンクします。例として、図 1 では履歴ウィンドウを変数ウィンドウの上にドラッグした時のドッキングガイドの様子を表します。マウスを下側のドッキングガイドに合わせて離すと、履歴ウィンドウは変数ウィンドウにリンクし、ウィンドウの下半分が履歴ウィンドウになります。

ドッキングウィンドウがリンクすると、分割線 (スプリッター) を 2 つのウィンドウの間に表示します。このスプリッターはリンクしたウィンドウの大きさを変更でき、片方のウィンドウを大きくすると、もう片方は小さくなります。

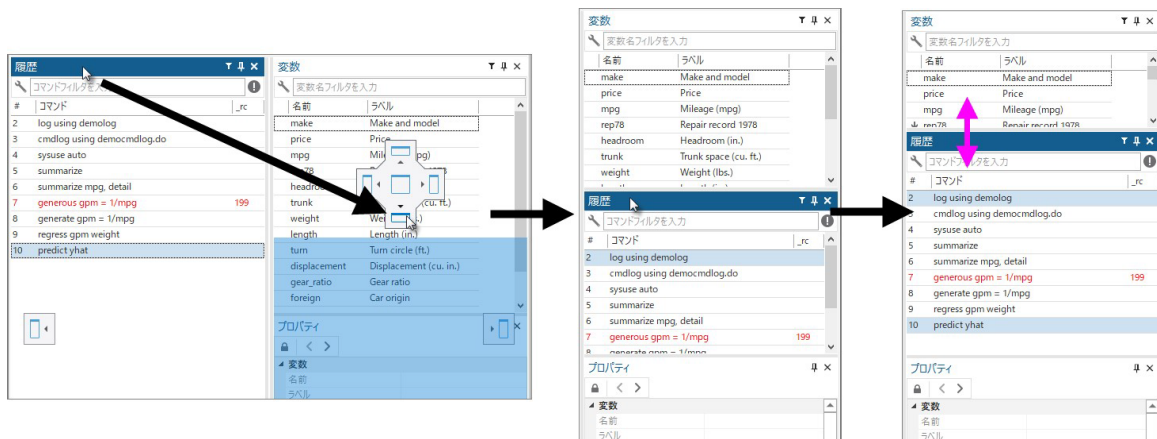


図 1: 2つのウィンドウをリンクする

ボックスを中央のドッキングガイド上にドロップすると (図 2)、ガイドの下にあるウィンドウと結合し、タブとして挿入します。ドッキングウィンドウをそれぞれタブで表し、新しいウィンドウの下に表示します。タブをクリックするとそのウィンドウが前面に移動します。タブとなったウィンドウを切り離すに

は、ウィンドウのタブをドッキングウィンドウからドラッグし、ドッキングガイドへと移動します。タブのついているドッキングウィンドウを他のドッキングウィンドウにリンクすることも可能です。

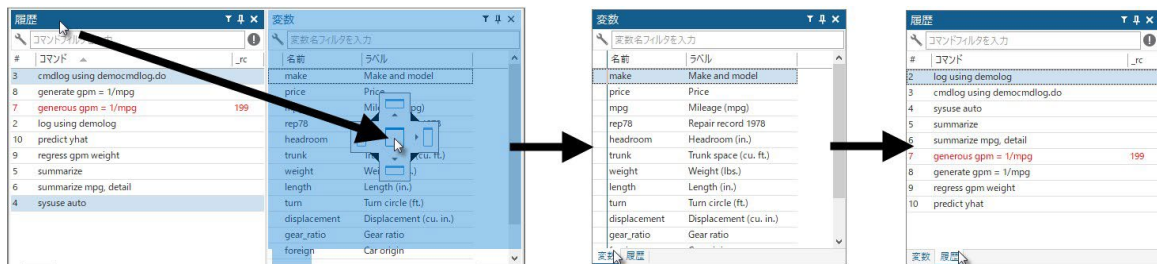


図 2: 2つのウィンドウを統合する

2.13 自動非表示とピンニング

ドッキングウィンドウは「自動非表示(Auto Hide)」の機能を備えています。ドッキングウィンドウが使われていない時は非表示にします。メインウィンドウ内のドッキングウィンドウのタイトルバーに画鋏アイコンが表示されている時、この機能を利用できます。

ドッキングウィンドウで「自動非表示」を行うには、タイトルバーの画鋏アイコンをクリックし(図 3)、画鋏アイコンの向きを水平にします。もう一度クリックすると「自動非表示」は解除され、画鋏アイコンの向きは垂直に戻ります。

ドッキングウィンドウの「自動非表示」を実行すると、使われていない時はウィンドウが非表示になり、メインウィンドウの端にタブとして表示されます。ウィンドウを表示するにはマウスカーソルをタブの上に移し、再び非表示にするにはマウスをウィンドウ内から動かします。ウィンドウをフォーカスしているとき(タイトルバーを選択しているとき)に他のウィンドウをクリックすると非表示になります。

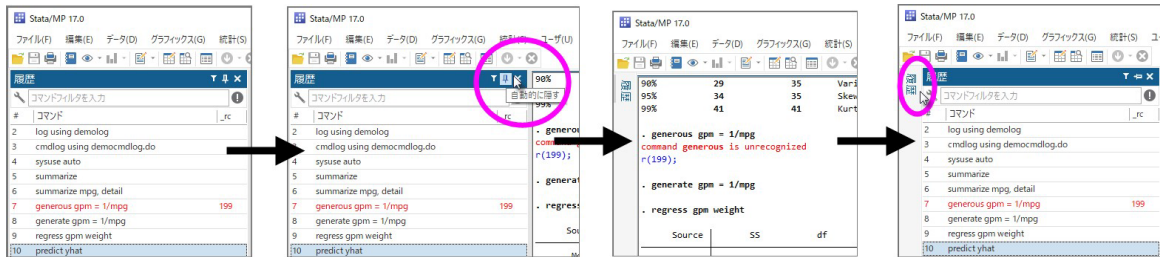


図 3：「自動非表示 (Auto Hide)」を行う

履歴ウィンドウの「自動非表示」をデフォルトの位置で実行した場合、表示した時にコマンドと結果ウィンドウ上の左側部分を隠します。この問題を回避するためには、履歴ウィンドウをメインウィンドウの右端に結合してから「自動非表示」を実行してください。

2.14 ノンドッキングウィンドウ


Graph、データエディタ、**do** ファイルエディタ、ビューワ、ダイアログのことをノンドッキングウィンドウと呼びます。ノンドッキングウィンドウは常にメインウィンドウから独立しています。結果ウィンドウは **Stata** で最も重要なウィンドウなので、メインウィンドウの中心に表示します。

3. Stata のユーザインターフェイス

3.1 ビューワの目的







ビューワは **Stata** 内で多岐にわたって活用できる便利なツールです。 **Stata** でヒントを得るために利用しますが、ビューワが提供するものはヘルプシステムだけではありません。ビューワでは、第三者が作成した拡張プログラムであるコミュニティ投稿コマンド(**community-contributed features**)の追加・削除・管理ができます。また、現在あるいは過去の **Stata** ログ、および **Stata** 形式(**SMCL**)またはテキスト形式(**ASCII**)のファイルも表示・印刷できます。更にウェブブラウザを使ってハイパーリンク先を表示できます。

この章は一般的なビューワの利用方法、ボタン、ビューワに入力できるコマンドについて説明します。ビューワに関する詳しい情報は [\[GSW\] 4 Getting help \(ヘルプ・ヒントを見つける\)](#) を、コミュニティ投稿コマンドのインストールについては [\[GSW\] 19 Updating and extending Stata—Internet functionality \(Stata のアップデートと拡張—インターネットでの機能\)](#) をご覧ください。

新しいビューウインドウ (または、既存のビューウインドウに新しいタブ) を開くにはビューボタン  をクリックするか、ウィンドウ > ビュー > 新規ビュー と選択します。いずれかの方法で開いたビューは複数のタスクへのリンクを表示します。

3.2 ビューのボタン




ビューのツールバーには複数のボタンに加え、コマンドボックスと検索ボックスがあります。

	戻る	画面を1つ戻ります。
	次へ	画面を1つ進みます。これは、最低でも一度は戻ったことを前提にします。
	再読み込み	ビューの内容を更新します。これはビューを開いた後に何かに変更された場合に使用できます。
	印刷	ビューの内容を印刷します。
	テキストを検索	ビューの下部に検索バーを開きます (下記参照)。
	検索	ビュー内でヘルプ検索の対象範囲を選択します。検索バーは現在のビュー内でのテキスト検索に使用します。

検索バーをウィンドウ下部に表示するには検索ボタン(上記参照)をクリックします。

✖ 検索:
前へ
次へ
 全てをハイライト
 大文字・小文字を区別してマッチ

検索バーには専用のボタンやエリアがあります。

	閉じる	検索バーを閉じます。
	検索	検索したいテキストを入力します。検索のオプションを変更するにはチェックボックスを使用します。
	前へ	検索テキストの直前の候補に移動します。ビューの先頭まで戻って次の検索対象がない時、自動的に最後の候補に戻ります。

<input type="checkbox"/> 次へ	次へ	検索テキストの次の候補に移動します。ビューワの最後まで検索を行って次の検索対象がない時、自動的に先頭に戻ります。
<input checked="" type="checkbox"/> 全てをハイライト	全てをハイライト	チェックを付けると、検索対象全てをハイライト表示します(デフォルトでは黄色)。チェックがない場合、現在の検索対象だけをハイライトします(デフォルトでは黒)。デフォルトではチェックは付いています。
<input type="checkbox"/> 大文字・小文字を区別してマッチ	大文字・小文字を区別してマッチ	チェックを付けると、大文字と小文字を区別して扱います。このチェックがついているときに This を検索すると、 this はヒットしません。チェックがついていない時は大文字と小文字の同一として認識されるので、 This で検索をすると this も検索結果としてヒットします。デフォルトでは、このチェックは付いていません。

3.3 ビューワの機能

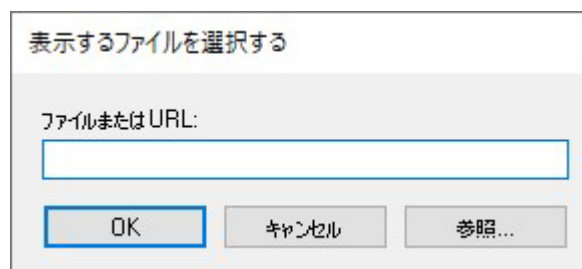
ビューワはウェブのブラウザと似ており、リンクがあります(デフォルトでは、青文字で表示)。リンクをクリックすると関連のあるヘルプトピックを表示し、第三者が作成したソフトをインストールおよび管理できます。マウスカーソルをリンク上に移動すると、そのリンクで実行される動作をウィンドウ下部にあるステータスバーに表示します。リンクが「**help logistic**」の時、このリンクをクリックすると **logistic** コマンドのヘルプファイルを表示します。ビューワのリンク上で **Shift+**クリックすると、リンク先を新しいビューワウィンドウに表示します。現在のビューワに新しいタブでリンク先を表示するには **Ctrl+**左クリックと操作します。

新しいビューワを開くには ウィンドウ > ビューワ > 新規ビューワ と選択するか、メインウィンドウにあるツールバーのビューワボタンをクリックします。コマンドウィンドウに「**help**」コマンドを入力しても同じように新しいビューワを開きます。

複数のビューウィンドウが開いている時に、1つのビューを最前面に移動するには ウィンドウ>ビュー と選択し、そのリストから該当するビューを選びます。全ビューを閉じる を選択すると開いているすべてのビューウィンドウとタブを閉じます。

3.4 SMCL ファイルを含むローカルテキストファイルを表示する

Stata のヘルプファイルを表示する以外にも、ビューを使用して **Stata Markup and Control Language(SMCL)**ファイルを見ることができます。SMCL は通常、作業のログを取るときに作成します(詳しくは [\[GSW\] 16 Saving and printing results by using logs \(ログを使い結果の保存や印刷を行う\)](#)をご覧ください。)また、ビューではテキストファイルも表示できます。ファイル>表示 と操作すると、ファイルを開いて内容を確認できます。



開きたいファイル名を入力して OK を押すか、参照...ボタンを押してファイルダイアログを開き、ファイル名を選択します。

ログの記録を継続中の場合、現在のログをビューで確認できます。ビューを開いておくと結果ウィンドウを後ろにスクロールする必要がなくなるという利点があります。結果ウィンドウでは新たな結果により表示が移動しますが、ビューでは移動しないからです。それは現在のログファイルを参照する場合、ファイル>ログ>表示 と選択すると表示するファイルを選択するダイアログが開きますが、パスとファイル名はログのそれがすでに入っています。OK をクリックすると、ログをビューに表示します。詳しくは [\[GSW\] 16 Saving and printing results by using logs \(ログを使い結果の保存や印刷を行う\)](#) をご覧ください。

3.5 インターネット上のファイルを閲覧する


インターネット上のファイルを閲覧する時は、ローカルファイルを表示する場合と同様に操作します。ただし、参照... ボタンでファイルでなく閲覧したい URL を、たとえば

<https://www.stata.com/man/readme.smcl> と入力します。ウェブ上のテキストか SMCL ファイルを見る時にだけビューワを使用してください。HTML ファイルのアドレスを入力すると、HTML ソースを表示します。

3.6 ビューワ内をナビゲートする

ウィンドウのスクロールバーの他に、キーボードの矢印上下キーまたは **Page Up/Page Down** キーでもウィンドウのナビゲートができます。矢印 (上下) キーを使用すると、1 行ごとにウィンドウがスクロールします。Page Up/Page Down キーを使用すると、1 画面ごとにウィンドウがスクロールします。

3.7 印刷

ビューワを印刷するには目的のビューワウィンドウ内で右クリックをし、コンテキストメニューから印刷... を選びます。または、メインウィンドウでメニューから **ファイル > 印刷 > ビューワタイトル** と選択するか、印刷ボタン  を押して開いているウィンドウのリストを表示し、印刷するものを選びます。

3.8 ビューワのタブ

ビューワウィンドウ内に複数のタブを作成できます。異なるファイル、または同じファイルの別の部分をそれぞれのタブで表示できます。新しいタブを開くボタン  をクリックすると現在のウィンドウに新しいタブが作成されます。ウィンドウのタブバーでタブをドラッグして順番を入れ替えることができます。タブをビューワウィンドウの本文にドラッグ&ドロップするとメニューが開きます。**New Horizontal Tab Group** を選ぶと、ビューワ本文を上下に分割します。この機能は同一ファイル内の異なる場所を 1 つの画面に表示する時などに便利です。同様に **New Vertical Tab Group** を選択すると、ビューワ本文を左右に分割します。この機能はファイルを隣り合わせて比較する時などに便利です。

一度上下または左右のタブグループを作成した後に別のタブをグループ内にドラッグ&ドロップすると、新しいタブのウィンドウを追加するか、または、そのタブを選択したグループに移動できます。

3.9 ビューワウィンドウで右クリックする

ビューワウィンドウで右クリックすると以下のオプションを含むコンテキストメニューを表示します。

- すべて選択 ウィンドウ上にあるすべてのテキストを選択します。
- ユーザ設定... ビューワの設定を編集します。
- フォント... ビューワのフォントを変更します。
- 印刷... ビューワの内容を印刷します。

これらの項目は、クリックした対象によって変更されます。

3.10 ビューワでヘルプを検索する

ビューワの検索ボックスでは文書(ヘルプ文書を含む)を検索できます。虫眼鏡をクリックします。そして全てを検索、ドキュメントと FAQ の検索、インターネットリソースの検索の中から **1** つを選びます。そして検索語句を検索ボックスに入力し、**Enter** キーを押します。ビューワをヘルプとして使用するための詳細は [\[GSW\] 4 Getting help \(ヘルプ・ヒントを見つける\)](#) をご覧ください。

3.11 ビューワのコマンド

ビューワの中でリンクやボタンをクリックして行える全ての操作は、ビューワのコマンドボックス(ウィンドウの上部)か **Stata** のコマンドラインにコマンドを入力して実行することも可能です。ビューワで実行できるコマンドをいくつかご紹介します。

- ヘルプを参照する (詳しくは [\[GSW\] 4 Getting help \(ヘルプ・ヒントを見つける\)](#) をご覧ください)。
 - 「contents」と打ち込んで **Stata** のヘルプシステムを表示します。
 - 「コマンド名」を入力して **Stata** のコマンドヘルプを参照します。
 - 「キーワード」を入力してトピックからドキュメンテーション、**FAQ**、インターネットリソースを検索します。
- 検索する (詳しくは [\[GSW\] 4 Getting help \(ヘルプ・ヒントを見つける\)](#) をご覧ください)。
 - 「search キーワード」と打ち込んでそのトピックに関する文書、**FAQ**、インターネット上のリソースを検索します。
 - 「search キーワード, local」と入力すると、そのトピックに関する文書と **FAQ** のみを検索します。
 - 「search キーワード, net」と入力すると、インターネット上の資料のみを検索します。

- コミュニティ投稿プログラムを検索し、インストールする (詳しくは [\[GSW\] 4 Getting help \(ヘルプ・ヒントを見つける\)](#) と [\[GSW\] 19 Updating and extending Stata—Internet functionality \(Stata のアップデートと拡張—インターネットでの機能\)](#) をご覧ください)。
 - 「net from <https://www.stata.com/>」を入力すると **Stata Journal**、**Stata Technical Bulletin**、コミュニティ投稿プログラムをインターネット上で検索してインストールできます。
 - 「ado」と打つとコミュニティ投稿のインストール済みプログラムを確認できます。
 - 「ado uninstall」と入力すると、該当するコンピュータにインストールされているコミュニティ投稿プログラムをアンインストールできます。
- ビューワでファイルを見る
 - 「view ファイル名.smcl」で **SMCL** ファイルを表示します。
 - 「view ファイル名.txt」でテキストファイルを表示します。
 - 「view ファイル名.log」でテキストのログファイルを表示します。
- 結果ウィンドウでファイルを見る
 - 「type ファイル名.smcl」とコマンドウィンドウに入力すると **SMCL** ファイルを結果ウィンドウに表示します。
 - 「type ファイル名.txt」とコマンドウィンドウに打ち込むとテキストファイルを結果ウィンドウに表示します。
 - 「type ファイル名.log」とコマンドウィンドウに入力するとテキストログファイルを結果ウィンドウに表示します。
- ブラウザを起動して HTML ファイルを表示する
 - 「browse URL」を入力してブラウザを起動します。

3.12 コマンドウィンドウからビューワを使う

「help コマンド名」をコマンドウィンドウに打ち込むと新規ビューワに指示されたコマンドのヘルプを表示します。

4. ヘルプ・ヒントを見つける

4.1 システムヘルプ

Stata のヘルプシステムには分析上役立つ情報が豊富に用意されていますので是非ご利用ください。一般的に次の手順に沿って目的の統計およびデータ管理用のコマンドについて調べることができます。

1. ヘルプ > 検索... の中から 全てを検索 を選び、トピックまたはキーワードを入力します。このコマンドは新しいビューウィンドウを開き、**Stata** のコマンド、**Stata Journal** 内のリファレンス記事、**Stata** のウェブ上にある **FAQ (Frequently Asked Questions)** へのリンク、**Stata** の **YouTube** チャンネルにある動画へのリンク、厳選した外部ウェブサイトへのリンク、コミュニティ投稿コマンドへのリンク等の情報を表示します。
2. これらの結果を確認してください。結果の中に参考になりそうなコマンドがあった場合、そのコマンド名のリンクをクリックし、ヘルプファイルを開きます。
3. 表示されたヘルプファイルをご一読ください。
4. より詳しいヘルプが必要な場合は、コマンド名のリンクをクリックして **PDF** 文書を開いてください。
5. 検索したファイルが目的のものでなかった場合、ビュー右上の 関連項目ボタン をクリックして関連するヘルプファイルのリストから他のリンクを選択します。あるいは、戻る ボタンをクリックして直前の文書に戻り、他のヘルプファイルを検索します。
6. ヘルプファイルを開いた状態でもコマンドウィンドウにコマンドを入力して目的のコマンドを実行できます。ビュー右上のダイアログボタンをクリックすると表示しているヘルプのコマンドダイアログを開きますので直接ダイアログからコマンドを実行できます。
7. 検索をやり直す場合は、新たなキーワードをビューウィンドウの検索ボックスに入力します。
8. ドキュメントと FAQ の検索 でキーワード検索を行った場合、**Stata** はコマンド、**Stata Journal** 記事とソフトウェア、**FAQ**、動画を検索します。インターネットリソースの検索 を選択すると、**Stata** はコミュニティ投稿コマンド (**Stata Journal** またはそれ以外の場所) を検索します。詳細は[\[GSW\] 19 Updating and extending Stata—Internet functionality \(Stata のアップデートと拡張—インターネットでの機能\)](#) をご覧ください。

では例題を使用してヘルプシステムの説明を行います。実際にコンピュータを動かしながら読み進めてみましょう。

たとえば、与えられたアンティークカーデータセットの内容を確認します。似たようなことを[\[GSW\] 1 Introducing Stata—sample session \(Stata の紹介—サンプルセッション\)](#)のサンプルセッションで行ったような気はしますが詳しいコマンドを覚えていないことにしましょう。

そこで、まずは「`sysuse auto, clear`」とコマンドウィンドウに入力してデータセットを開きます。

(`clear` オプションについての詳細は [\[GSW\] 5 Opening and saving Stata datasets \(Stata のデータセットを開く・保存する\)](#) をご覧ください。)

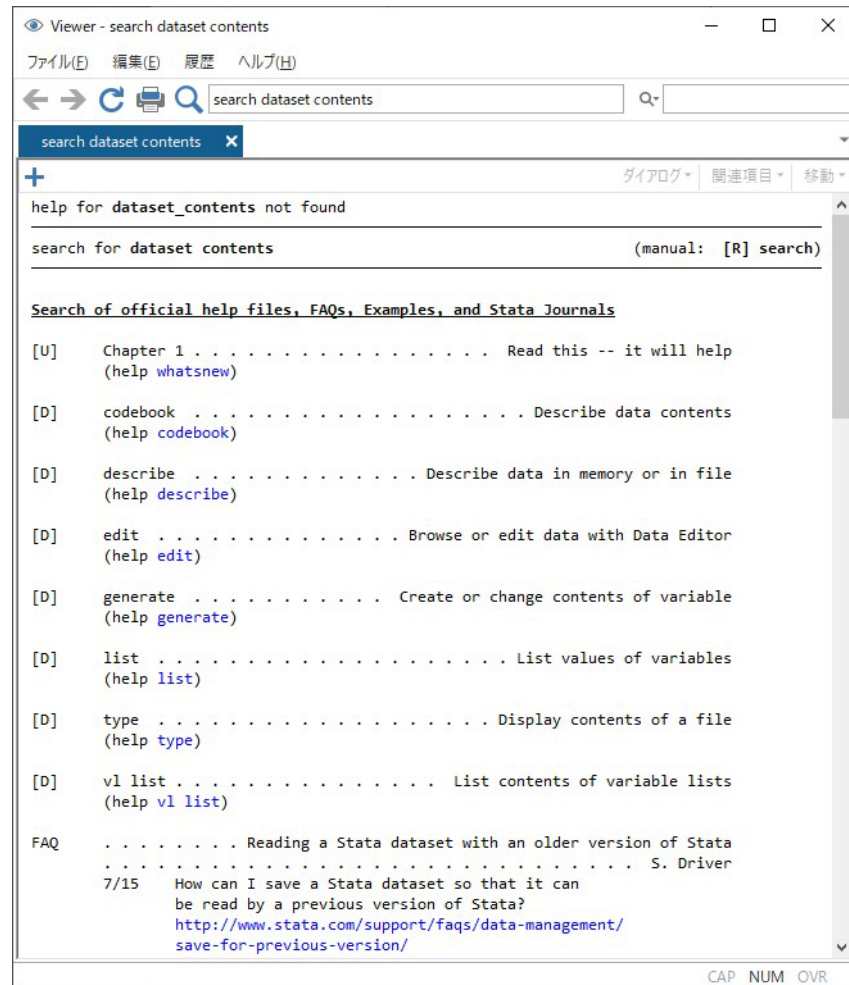
上記手法で調べていきます。

1. メニューから ヘルプ > 検索... と選択します。
2. 全てを検索 のラジオボタンが選択してあることを確認してください。
3. 「dataset contents」と検索ボックスに打ち込み、OK をクリックするか **Enter** を押します。

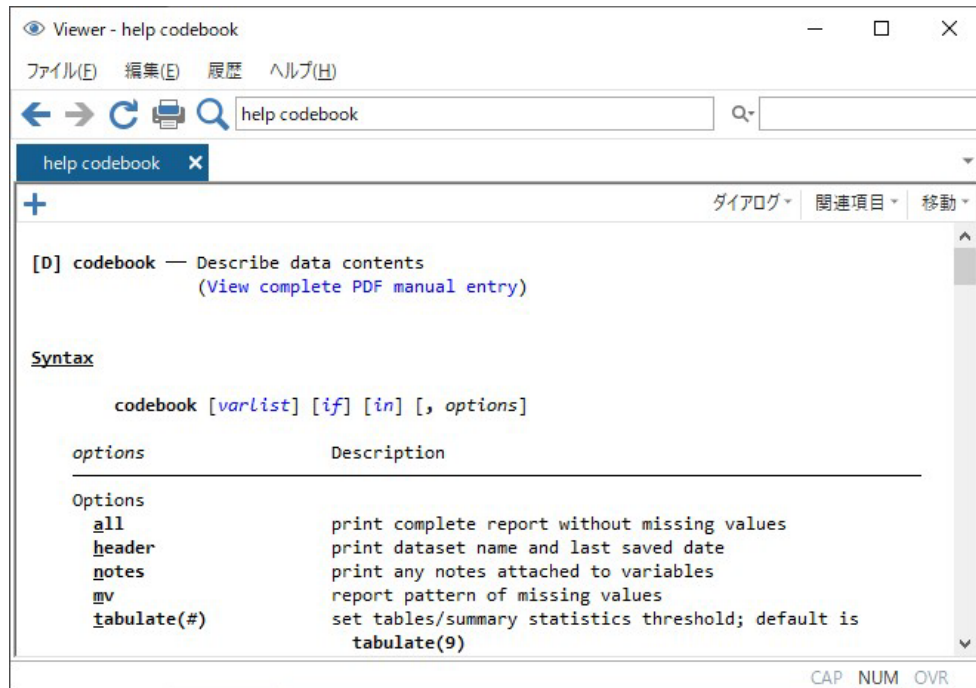
Enter キーを押す前のウィンドウは次のようになります。



4. **Stata** はコマンド、リファレンスマニュアル、**Stata Journal**、**Stata** のウェブ上の **FAQ**、コミュニティ投稿コマンドの中を“dataset contents”のキーワードで検索します。結果は次の通りです。



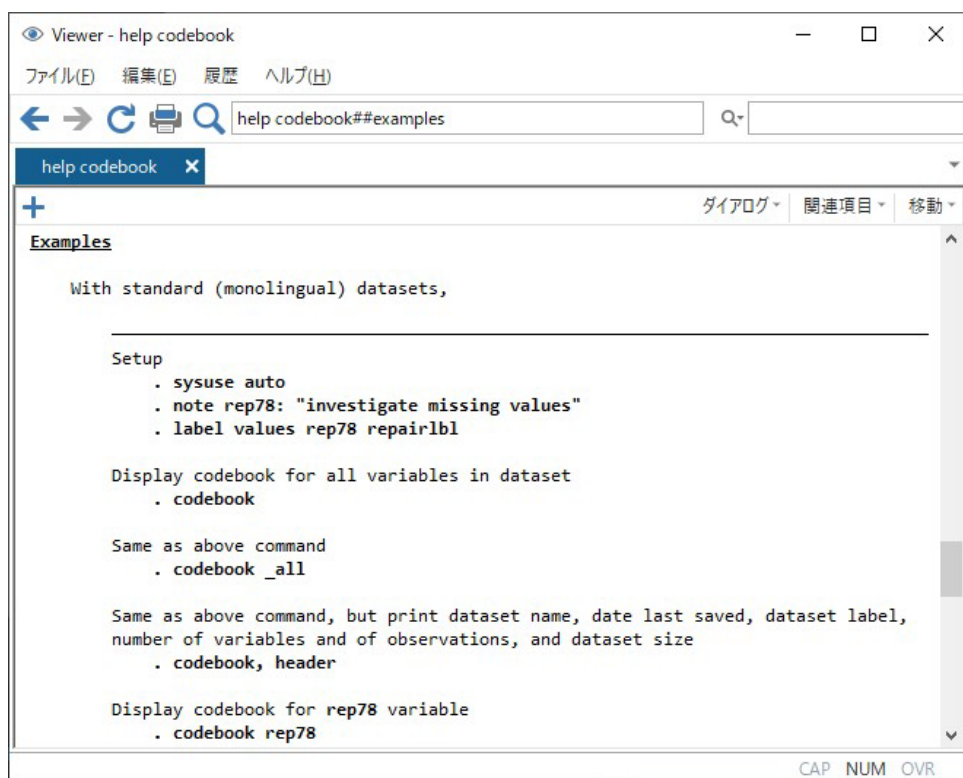
5. 検索結果を見ると、「codebook」と「describe」コマンドが使えそうです。データセットの中身を知りたいので、「codebook」コマンドを確認しましょう。**[D]**は**Data Management Reference Manual**でcodebookコマンドについて調べることを意味しています。(help codebook)の中にあるcodebookリンクはcodebookコマンドのヘルプファイルがシステムヘルプにあることを示しています。これが今探していたものです。
6. codebookのリンクをクリックします。リンクは多様なリソース、たとえばStataコマンド、ダイアログ、ウェブページ等に繋がっています。目的のリンクからcodebookコマンドのヘルプファイルに移動します。



7. 開いた画面はコマンドのヘルプの典型的な例です。コマンドのヘルプファイルは、上から順に次のような内容のセクションに分かれています。
- (a) クイックアクセスツールバー (タブのすぐ下にあるバー)には **3** つのボタンがあります。
 - i. ダイアログ ボタンはコマンドに関連するダイアログへのリンクを表示します。
 - ii. 関連項目 ボタンは関連する **PDF** 文書またはヘルプファイルのリンクを表示します。
 - iii. 移動 ボタンは現在のヘルプファイル内のセクションへのリンクを表示します。
 - (b) ヘルプファイルの **2** 行目 **View complete PDF manual entry** リンクがあります。このリンクをクリックすると、ヘルプファイルのコマンド、ここでは **codebook** のドキュメンテーションを **PDF** ビューワーで開きます。
 - (c) コマンド構文、つまり **Stata** が読み取るコマンド構成のルールについて **[]** は、**codebook** コマンドに対してカッコ内の引数が任意である事を表します。引数を指定する場合は **varlist**、**if** 条件、**in** 条件の他にいくつかのオプションを選択できます。(使用可能なオプションはコマンドにより大きく異なります。) オプションはコマンドのすぐ下に記載してあり、詳しい説明はヘルプファイルの後半にあります。コマンド構文は、[\[GSW\] 10 Listing data and basic command syntax \(データのリストと基本コマンドの構文\)](#) で詳しく学びます。
 - (d) コマンドの概略を示します。**codebook** はデータセットの各変数の情報を示すだけの簡単なもので、短い説明しかありません。

- (e) このコマンドで使用できるオプションです。ここのリストは構文における解説と比べると、各オプションについてさらに詳しく説明しています。たとえば、`mv` オプションは欠損値がある時に、欠損値の出現には何らかのパターンがあるか調べるものです。このオプションはデータ整理や補完 (imputation) に役立ちます。
- (f) コマンド使用例があります。この `codebook` の例題は、実際に 1 ステップずつ操作していく例題です。必要なデータは **Stata** と共にシステムにインストールされているかインターネットから無料でダウンロードできます。
- (g) コマンド実行後にメモリ上に確保される情報です。この内容は基本的にプログラマー向けです。

では、移動 ボタンをクリックして **Examples** を選ぶか、**Examples** までスクロールダウンします。例題に挑戦してみましょう。例題の先頭部分スクリーンショットは次の通りです。



```

Viewer - help codebook
ファイル(F) 編集(E) 履歴  ヘルプ(H)
help codebook##examples
help codebook
+
Examples
With standard (monolingual) datasets,

Setup
. sysuse auto
. note rep78: "investigate missing values"
. label values rep78 repairlbl

Display codebook for all variables in dataset
. codebook

Same as above command
. codebook _all

Same as above command, but print dataset name, date last saved, dataset label,
number of variables and of observations, and dataset size
. codebook, header

Display codebook for rep78 variable
. codebook rep78
CAP NUM OVR

```

4.2 ヘルプを探す

検索... では統計、グラフ、データ管理、プログラミングに関する機能を検索できます。これらは公式的なリリースに組み込まれているかユーザ定義コマンドとして使用できます。検索するトピックを入力するときは、統計の正しい表現を使用してください。たとえば、「**Mann-Whitney**」と入力します。また、複数のキーワードを「**regression residuals**」のように入力することも可能です。

検索... を使用するときには「正しい英語」と「正しい統計用語 (英語)」を使用してください。コマンド名が分かっている、直接ヘルプファイルを表示したい場合はメニューから ヘルプ > Stata のコマンド... と選び、コマンド名を入力します。あるいはビューワの上部の検索欄にコマンド名を入力し **Enter** キーを押します。

ヘルプはキーワードがコマンドかトピックかを区別します。これはいくつかのコマンド名はそのままトピックとしても使用できるからです。たとえば「**logistic**」は Stata のコマンドです。ここでヘルプメニューから Stata のコマンド... を選び「**logistic**」と入力すると、直接 **logistic** コマンドのヘルプファイルに移動します。一方、検索... を選択し、「**logistic**」と打ち込むと、検索結果としてロジスティック回帰に関する多くのコマンドを表示します。既に記述した通り、検索はビューワ内部からでも行えます。検索する場合はコマンドをビューワのコマンドボックスに直接入力します。または検索ボックスの左にある虫眼鏡をクリックし、検索の範囲を設定した後にキーワードを **search** ボックスに入力して **Enter** キーを押します。

4.3 ヘルプと検索のコマンド

ヘルプシステムはコマンドウィンドウからも利用可能です。ある特定のコマンドのヘルプを見たい時に特に便利です。次に使用できるコマンドをリストで記します。

- 「**help** コマンド名」を入力すると、ヘルプ > Stata のコマンド... と選択してコマンド名を入力するのと同じになります。このヘルプファイルは新しいビューワウィンドウに表示します。
- 「**search** 項目」をコマンドウィンドウに入力すると、ヘルプ > 検索... の後に 全てを検索 を選んで項目を入力した場合と同じになります。検索結果は新しいビューワウィンドウに表示します。
- 「**search** 項目, **local**」をコマンドウィンドウに入力すると、ヘルプ > 検索... の後に ドキュメントと

FAQ の検索 を選んで項目を入力した場合と同じになります。ビューウィンドウではなく結果ウィンドウに検索結果を表示します。

- 「search 項目, net」をコマンドウィンドウに入力すると、ヘルプ>検索... の後にインターネットリソースの検索 を選択して項目を入力した場合と同じになります。ビューウィンドウではなく結果ウィンドウに検索結果を表示します。

詳しくはユーザガイド内の [\[U\] 4 Stata's help and search facilities](#) と [\[U\] 4.8 search: All the details](#) でコマンド言語版のヘルプシステムについての情報をご覧ください。search コマンドは、特にここで説明していない機能 (たとえば作者検索) を備えています。

4.4 Stata リファレンスマニュアルと User's Guide

全ての Stata リファレンスマニュアルは PDF ファイルでソフト内に含まれています。マニュアル自体は内部にクロスリファレンスが多くあり、クリックで関連するページを移動できます。順序に沿って文書を読む必要はありません。

ヘルプファイル内の多くのリンクは PDF マニュアルにつながっています。リンクをクリックすると、マニュアル内にある多くの情報をご利用いただけます。Stata のヘルプシステムは多くの内容を含みますが、マニュアルの中にある情報のほんの一部でしかありません。

Stata のリファレンスマニュアルはアルファベット順に並んでいます。Getting Started マニュアルの各 OS 版は、それぞれ索引がついています。全マニュアルの統合的な索引は、Stata Index に記載されています。コマンドについて調べたい時に、まずこの統合索引を確認してみると良いでしょう。

索引語の collapse, egen, summarize のようにそれ自身も Stata のコマンドとなるものも多くあります。

検索結果およびヘルプファイルの [\[R\] ci](#)、[\[R\] regress](#)、[\[R\] ttest](#) などという表記は Base Reference Manual を参照します。この他にも [\[P\] PyStata integration](#) があり、これは Programming Reference

Manual を、[U] 12 **Data** は **User's Guide** を参照します。マニュアルのリストとそれぞれの略語表記についてはこのマニュアルの目次直後にある「**Stata** の他のマニュアル参照について」をご覧ください。

これらのリファレンスマニュアルの使用に関するアドバイスは [GSW] 18 **Learning more about Stata** (**Stata** についてもっと詳しく学ぶ)、または [U] 1.2 **The User's Guide and the Reference manuals** をご覧ください。

4.5 Stata 動画

Stata YouTube channel は **Stata** を知る上で優れたリソースです。各トピックが短めの動画の中で実際に **Stata** を操作しながら紹介されます。扱っているトピックは、データ管理、グラフィックス、要約統計、仮説検定から、マルチレベルモデル、構造方程式モデルなどの高度な話題にまで及びます。

トピックに沿って一連の動画を再生するプレイリストも用意されています。たとえば、「**Power and sample size calculations**」のプレイリストには 2 つの独立した群や対応のある 2 群における検出力、標本の大きさ、効果量の計算方法を紹介する動画がまとまっています。「**Survival analysis**」では生存分析のためのデータ設定処理、基本的な記述分析、グラフ化、生存関数や生命表の計算という一連の流れが閲覧できます。「**Time series**」では時系列分析用のデータ設定処理、時系列グラフの作成、推定式での演算子 (オペレータ) の使用方法、ARMA や ARIMA モデルでのフィットの操作法がご覧になれます。また、「**Back-to-school video**」では学生、**Stata** を初めてまたは久しぶりに使う方のための動画を集めています。

<https://www.stata.com/links/video-tutorials/>には、最新のものを含む動画がトピックごとに紹介されています。<https://www.youtube.com/user/statacorp/>からは、プレイリストへ直接アクセスできます。

4.6 The Stata Journal

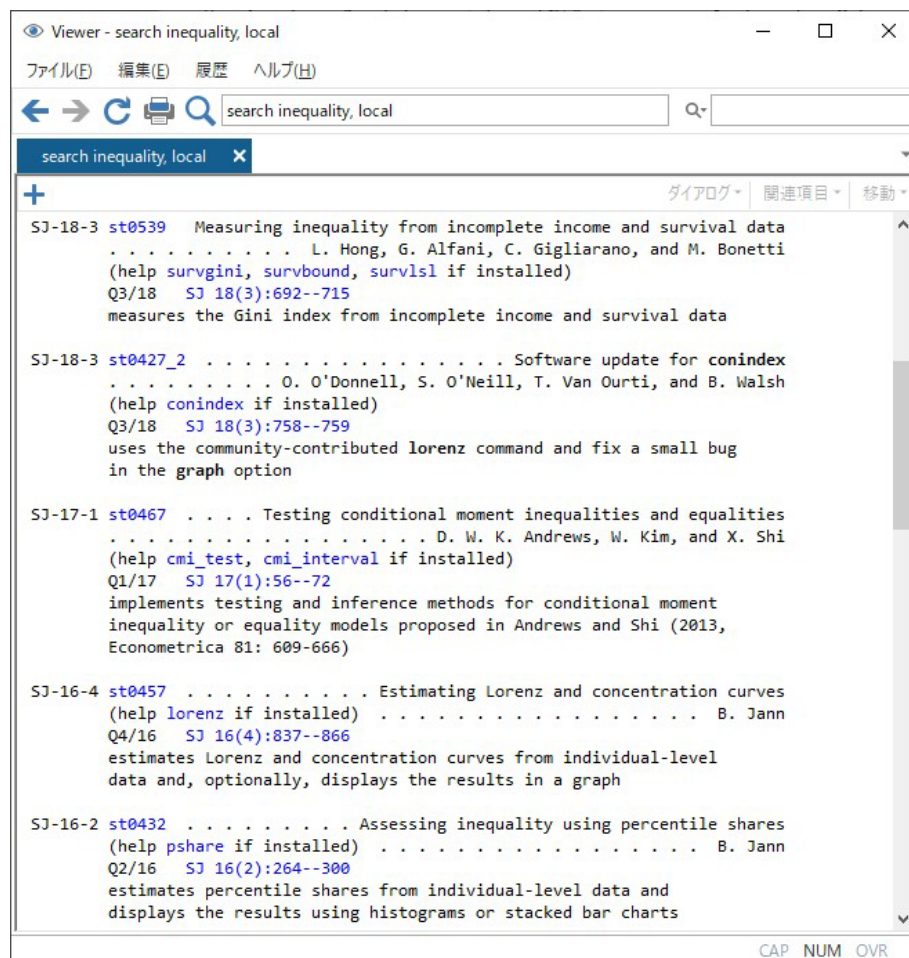
Stata で検索を行っている時、**Stata Journal** へのリンクをよく見かけます。

Stata Journal は印刷および電子版の機関誌で、年 4 回発行しています。内容は統計、データ分析、教授方法、**Stata** 言語の有効活用法などにわたります。**Stata Journal** は査読付きの論文ですが、短い注釈やコメ

ント、コラム、ヒント、書評など、統計を様々な分野で利用する人に有益な情報も合わせて掲載しています。初心者から熟練者までのすべての **Stata** ユーザを対象にした機関誌です。 **Stata** の熟練度は問いません。統計、実験計画、データ管理、グラフ作成、結果発表などを、 **Stata** で行う研究者のための論文誌です。詳しくは <https://www.stata-journal.com> をご覧ください。このウェブページではアーカイブの閲覧、購読登録、特定の記事の PDF 版を購入できます。

この **Stata Journal** には論文で説明しているプログラムとデータセットが対になって用意されています。プログラムとデータセットは購読者だけでなく **Stata** ユーザならばインターネット経由でダウンロードしインストールできます。詳しくは [\[R\] net](#) と [\[R\] sj](#) をご覧ください。

Stata Journal には **measures of inequality** についての記事が複数あります。ヘルプ>検索... と選択し、ドキュメントと **FAQ** の検索を選んだ後、「**meta analysis**」と入力してから少しスクロールします。



SJ-18-3 は volume 18 の 3 番目に発行した **Stata Journal** です。 **st0539** はパッケージ番号を表します。 **st** はこのパッケージが **Stata Journal** の“**statistics** (統計)”カテゴリに分類されることを意味します。続く

記述はソフトウェアパッケージのタイトルと著者名です。続くカッコ内にはパッケージ内で使用されるコミュニティ投稿コマンドが記述されます。**SJ**のリンク（たとえば、[SJ 18\(3\):692--715](#)）をクリックするとブラウザが起動して **Stata Journal** のウェブサイトに移動します。ここで要約や記事をダウンロードできます。各検索結果は、コミュニティ投稿コマンドの簡単な説明で記述が終了します。

Stata Journal のウェブサイトでは 3 年以上経過した記事は全て無料でダウンロードできます。このソフトウェアのインストールについての詳細は [\[GSW\] 19 Updating and extending Stata— Internet functionality \(Stata のアップデートと拡張—インターネットでの機能\)](#) 中の「コミュニティ投稿のプログラムをダウンロードする」を参照してください。そして便利なインターフェイスやリソースは **Statistical Software Components (SSC)** アーカイブから利用できます。詳細は [\[R\] ssc](#) をご覧ください。

ユーザには **Stata Journal** を購読することをお勧めします。詳しくは [\[U\] 3.4 The Stata Journal](#) をご覧ください。

他のプログラムやデータセットを無料でダウンロードできるサイトへのリンクは **Stata** のウェブサイト (<https://www.stata.com/links/>) に掲載されています。

5. Stata のデータセットを開く・保存する

5.1 データセットをロードし、保存する

Stata のデータセットを開き、保存するには他のコンピュータアプリケーションと同じように操作します。しかし、2つの相違点があります。1つ目はコマンドウィンドウからファイルを開き、保存できる点です。2つ目は、Stata は一度に1つのデータセットしかアクティブな状態にできないという点です。つまり、メモリ上には一度に複数のデータセットを読み込むことができます(詳しくは [\[D\] frames Intro](#) をご覧ください)。ビューワで複数ファイルの閲覧はできますが、データセットの使用は1つのみです。新しいデータセットを開くときには必ず既存のデータセットを閉じてください。この章ではデータセットを開き、保存する全ての方法を紹介します。

Stata のデータセットは複数の方法で開くことができます。ほとんどの方法は他のアプリケーションでなじみ深いものでしょう。

- Stata のデータファイル (拡張子が **.dta**) をダブルクリックします。
メモ：拡張子はコンピュータのシステム設定によっては非表示になっている場合もあります。
- ファイル > 開く... とメニュー操作するか、開くボタンを押してファイルへ行きます。
- ファイル > データのサブセットを開く... と操作してファイル名、観測値の範囲、使用する変数を選択します。
- ファイル > 最近のファイル > ファイル名、と選択します。
- 「use ファイル名」とコマンドウィンドウに入力します。このコマンドは、現在の作業フォルダ (**working directory**)内を入力した「ファイル名」で検索します。ファイルが他のフォルダにあるなら、目的のフォルダのパスを指定しなければなりません。もしパスやファイル名のどこかにスペース(空白)があるなら、そのファイル名をダブルクォテーション (“”) の中に入れるよう注意してください。詳しくは [\[U\] 11.6 Filenaming conventions](#) をご覧ください。
- 「sysuse ファイル名」とコマンドウィンドウに入力します。このコマンドは、**adopath** と呼ばれるディレクトリ内にあるファイル名を検索します。通常、このコマンドは Stata をインストールする時にソフトウェアと共にインストールした例題のデータセットを検索しますが、それ以外でも、自分のデータセットに簡単にアクセスする手段として使用できます。**adopath** についての詳細は [\[P\] sysdir](#) をご覧ください。
- 「webuse ファイル名」とコマンドウィンドウに入力します。**webuse** コマンドは Stata マニュアルで使用するデータセットにアクセスする際に使用します。たとえば、「webuse lbw」は **logistic** コマンドのマニュアル文書内で使用している **lbw** データセットをロードします。詳細は [\[D\] webuse](#) をご覧ください。

開いているフレームでデータセットを新たに開くと、**Stata** はメモリ内にあるフレームのデータセットを破棄します。他のフレームにあるデータセットは影響を受けません。**Stata** は現在開いているフレームのデータセットに変更を加えた場合、操作を通じて強制しない限りデータセットの破棄を拒否します。コマンドウィンドウ以外の方法でファイルを開こうとするとポップアップを表示します。一度変更を加えたデータセットからコマンドウィンドウを使用して他のデータを開こうとすると、以下のエラーメッセージを表示します。

```
. sysuse auto
no; dataset in memory has changed since last saved
r(4);
```

この機能は誤ってデータを消す可能性を低くします。

名前を付けていないデータセットを保存する(または既存のデータセットを新しい名前でも保存する)には次のように操作します。

1. ファイル>名前を付けて保存... と選択します。
2. 「save ファイル名」とコマンドウィンドウに打ち込みます。

Stata 13 の形式でデータセットを保存するには、次のようにします。

1. ファイル>名前を付けて保存... を選び、ファイルの種類のドロップダウンリストから **Stata 13 データ (*.dta)** を選びます。
2. 「saveold ファイル名」とコマンドウィンドウに打ち込みます。

変更したデータセットを保存する (または元のデータセットを上書きする) には次のように操作します。

1. ファイル>保存と選択します。
2. はいボタンをクリックします。
3. 「save, replace」とコマンドウィンドウに入力します。

データセットを上書きするとデータセットを元に戻すことはできません。重要なデータセットに関しては元のデータのコピーをバックアップとしておくか、変更後の内容を新しい名前でも保存します。ワープロのド

キュメント (**Microsoft Word** など) とほぼ同様に操作できますが、一度上書き保存をするとデータセットの回復がほぼできなくなるという違いもあります。

重要：データセットに施した変更は保存するまで完了しません。この間の作業はデータセットのメモリで行われており、データファイルそのものではありません。ほぼすべてのコンピュータアプリケーションはこのように動作しています。

既存のデータセットを保存しない場合、そのデータセットをクリアすれば新たなデータセットを開けます。他のファイルを開くにはコマンドウィンドウに「**use** (開きたい) ファイル名, **clear**」と入力します。

6. データエディタを使用する

6.1 データエディタ


データエディタは現在メモリ内にあるデータをスプレッドシート形式で表示します。この画面を使用してデータの入力、編集、データセットの属性編集を行えます。変数名、ラベル、表示形式の設定や値ラベルを作成する時は、この属性を編集します。

データエディタはデータの表示だけでなく、変数の変更とプロパティの変更を行う、変数ウィンドウとプロパティウィンドウも表示します。これはメインウィンドウにある、同名のウィンドウと同じように機能します。

データエディタで行った操作も、あたかもコマンドウィンドウに直接入力したかのように結果ウィンドウへ表示されます。つまり、何を行ったのか記録できるのでデータエディタを通じてコマンドを学ぶこともできます。

データエディタは作業中表示しておくことができ、データの様子を見ながら作業できます。データを誤って変更しないようにデータエディタには2つの表示モードがあります。編集モードで編集を加え、ブラウズモードで表示のみを行います。ブラウズモードではデータエディタウィンドウ内でデータを編集できません。データをチェックする時などはブラウズモードを使い、編集が必要な場合にのみ編集モードに切り替えて使用することをお勧めします。










メニューでファイル>印刷...と選択すると、データエディタの内容を印刷できます。

この章ではデータ入力および編集を実際に行います。また変数とプロパティのウィンドウを使用して値の変更も行います。ではデータエディタ (編集) ボタン  をクリックして編集モードのデータエディタを開きましょう。

6.2 データエディタ内のボタン

データエディタのツールバーには標準ツールバーボタンに加え、いくつか新しいボタンがあります。



	編集モード	データエディタを編集モードに切り替えます。
	ブラウズモード	データエディタをブラウズモードに切り替え、データを変更することなく、閲覧します。
	開く	Stata データセットを開きます。
	保存	データエディタに表示されているデータセットを保存します。
	印刷	データエディタに表示されている変数を印刷します。
	コピー	現在選択しているものをクリップボードにコピーします。
	貼り付け	クリップボードの内容を貼り付けます。貼り付ける内容の左上となるセルを1つ指定します。注意：既存のデータは上書きされます。
	検索	データエディタ内を搜索する検索バーを開きます。
	観測値フィルタ	データエディタで表示する観測値にフィルタをかけます。このボタンはデータセット内のサブセットを表示するのに便利です。

データエディタでアクティブなセルを動かす時は次のようにします。

- 右に移動するには **Tab** キーか、右矢印キーを押します。
- 左に移動するには **Shift+Tab** キーか、左矢印キーを押します。
- 下に移動するには **Enter** キーか、下矢印キーを押します。
- 上に移動するには **Shift+Enter** キーか、上矢印キーを押します。

目的のセルをクリックすると直接選択できます。

データエディタ内で右クリックをするとコンテキストメニューを表示します。このメニューを使ってデータや表示内容を変更できます。このメニューを使ってデータや表示内容を変更できます。詳しくは章の後半で説明します。実際にデータエディタを右クリックすると、次に示すようなコマンドを表示します。

- **コピー** クリップボードにデータをコピーします。
- **貼り付け** クリップボードのデータを貼り付けます。
- **特殊な貼り付け...** より繊細な区切り文字をコントロールし、データのプレビューを確認しながらクリップボードのデータを貼り付ける事ができます。

- **すべて選択** データエディタ内に表示しているデータをすべて選択します。これはデータにフィルタがかかるか、変数が非表示になっている場合はデータセット全体とは異なる可能性もあります。
- **データ** 以下の項目を含むサブメニューを表示します。
 - **変数を挿入...** 新規の変数を作成するダイアログを表示し、現在のカーソル位置に挿入します。
 - **変数を追加...** 新規の変数を作成するダイアログを表示し、データセットの開始または終了位置に追加します。
 - **変数の内容を変更...** 選択した変数の値を置き換えるダイアログを表示します。
 - **観測値を挿入...** 新規の空の観測値を作成するダイアログを表示し、現在のカーソル位置に挿入します。
 - **観測値を追加...** 新規の空の観測値を作成するダイアログを表示し、データセットの開始または終了位置に追加します。
 - **データをソート...** 選択された変数でデータをソートします。
 - **値ラベル** 値ラベルの表示・管理のサブメニューを開きます。
 - **値ラベルの管理** 「値ラベルの管理」ダイアログを開きます。
 - **選択した変数のみ維持** 選択したデータのみをデータセットに残します。これ以外のデータはデータセットからドロップします(取り除きます)。この操作はメモリ内のデータにのみ適用され、ディスク上のデータには影響はしません。
 - **選択範囲を削除** 選択したデータをドロップします。これは選択範囲が変数全体(列全体)または観測値全体(行全体)である時のみドロップできます。
 - **変数を文字列から数値に変換...** 文字列変数を数値変数に変換する際に使用します。なお、文字列変数に数値形式を指定する記号が含まれていると、より変換しやすくなります。
 - **数値変数から文字列変数に変換...** 数値変数を文字列変数に変換する際に使用します。
 - **文字列をラベル付数値にエンコード...** 文字列を値とするカテゴリー変数を数値を値とするよう符号化(エンコード)します。その際、カテゴリーは表やグラフ内の表示は文字列のままになります。

- ラベル付数値を文字列にデコード... エンコードされたカテゴリ変数を、元の文字列変数に変換します。

- **選択した観測値/変数を固定する** 選択した観測値または列を固定します。一つ以上の変数（列）を選択したときは**選択した変数を固定する**が表示されます。一つ以上の観測値（行）を選択したときは**選択した観測値を固定する**が表示されます。
- **選択した列の幅をリセットする** 選択した列をデフォルトの幅にリセットします。
- **選択した変数を非表示** 選択した変数を非表示にします。
- **選択した変数のみ表示** 選択した変数以外を非表示にします。
- **フィルタ解除** 全てのフィルタを解除し、すべての変数を表示します。
- **ユーザ設定...** データエディタの設定を編集します。
- **フォント...** データエディタのフォントを変更します。

6.3 データ入力


データエディタにデータを入力するのはスプレッドシートにデータを入力する事と似ています。しかし、相違点の1つとしてデータエディタは観測値というコンセプトがあり、データ入力をスマートに行えます。例題を使い、この内容を確認しましょう。お手元のコンピュータで操作しながら進んでください。作業を始めるには空のデータセットが必要です。必要に応じてデータセットを保存し、コマンドウィンドウに「clear」と入力してください。

メモ：コマンドウィンドウに「describe, short」（または省略して「d, s」）と入力してデータが変更されたことを確認します。データが変更されている場合は、**Stata** が適切なメッセージを表示します。

次に示すデータを、これから説明する方法に従って入力してみましょう。

Make	Price	MPG	Weight	Gear ratio
VW Rabbit	4697	25	1930	3.78
Olds 98	8814	21	4060	2.41
Chev. Monza	3667		2750	2.73
AMC Concord	4099	22	2930	3.58
Datsun 510	5079	24	2280	3.54
	5189	20	3280	2.93
Datsun 810	8129	21	2750	3.55

3番目の車のMPGと6番目の車のMake(車種)はわかりません。

まず、データエディタを編集モードで開きましょう。データエディタ(編集)ボタンをクリックするか、コマンドウィンドウに「edit」と入力すると、空のデータエディタウィンドウが開きます。既にデータが存在する場合は、コマンドウィンドウに「clear」と入力します。Stata はアクティブなセルをハイライトして表示し、カーソル位置ボックス(ウィンドウ上部、左側のボックス)の隣にある入力ボックスに **varname [obsnum]** と表示します。後ほど詳しく見ていきますが、このセル参照機能を使ってデータセット内を移動することもできます。データエディタはデフォルトで **1 行 1 列目** から始まります。まだデータが無く、変数名もないので位置ボックスには **var1[1]** と表示されます。

データは横(観測値ごと)か縦(変数ごと)に入力します。観測値ごとにデータを入力する場合、値を入力するごとに **Tab** キーを押すと左から順に移動しながら入力できます。この例題の場合、「**VW Rabbit**」と打った後に **Tab** キーを押します。そして「**4697**」と入力して **Tab** キーを押します。このように **1 番目** の観測値(行)を全て入力します。

1 番目 の観測値(行)の入力が終わったら、**2 行 1 列目** のセルを選択します。クリックして選択するか、キーボードを使用して移動しましょう。この時点では次のような画面を表示します。



2 行目 (観測 2) も **1 行目** と同じ要領で入力します。最後の列 (**var5**) まで入力した後に **Tab** キーを押すと自動的にカーソルが **3 行 1 列目** に移動します。これは **1 行目** の入力の後、変数の数を Stata が認識したことによるものです。

残りのデータの 입력は各セル間を **Tab** キーで移動しながら行い、欠損値(空欄)に関してはデータを入力せずに、そのまま **Tab** キーで移動します。

変数(列)ごとにデータを入力する場合、値を入力するごとに **Enter** キーを押して下方向に移動します。欠損値に関しては **2** 回連続で **Enter** キーを押します。初めの変数の 입력が終わったら、**1** 行 **2** 列目のセルを選んで次のデータ **Price** (価格) を 입력します。全てのデータが 입력できるまで繰り返します。

6.4 データ入力に関するメモ

データ入力に関する注意点を説明します。

- データセット内に空の行や列は作成できません。
新しい変数や観測値を入力する際は向かって左上角から順番に入力します。もし列または行を飛ばしてしまった場合、そこは自動的に欠損値になります。
- 文字列 (**string**) と値ラベルは数値とは異なる色を表示します。
データエディタ内の様々な変数タイプを区別するために、文字列 (テキスト)、値ラベル (詳しくは [\[GSW\] 9 Labeling data \(データのラベリング\)](#) をご覧ください)、それ以外の値を異なる色で表示します。文字列と値ラベルの色を変更するにはデータエディタウィンドウで右クリックをしてユーザー設定... を選びます。
- ピリオド (.) は欠損値を示します。
Stata にはシステム欠損値 (.) があり、拡張欠損値が **'a'** から **'z'** まであります。デフォルトではこのシステム欠損値 (.) を使用します。
- **Tab** キーについて
先の例で見たように、**1** 行目の観測値を全て入力すると、**Stata** は変数の数を認識します。**2** 行目の最後の観測値で **Tab** キーを押すと自動的に **3** 行 **1** 列目に戻ります。
- カーソル位置ボックスはカーソル位置の表示とカーソル移動に使用できます。
カーソル位置ボックスは現在選択しているセルの位置を表示します。たとえば、**var3[4]** のようになっているとき、選択セルは変数 **var3** の **4** 行目にあることを示します。特定のセルへ移動するには、カーソル位置ボックスに変数名と行番号を入力すれば移動できます。たとえば、変数 **var1** の **2** 行目のセルを選択する場合は「**var1 2**」とカーソル位置ボックスに入力して **Enter** キーを押します。

- 文字列をダブルクォーテーションで囲む必要はありません。
Stata が一度でもその変数を **string** (文字列) であると認識すれば、たとえその値が数値に見えても、ダブルクォーテーション (" ") で囲む必要はありません。たとえば、郵便番号 (**ZIP code**) をテキストとして入力するには、**1** 行目は郵便番号をダブルクォーテーションで囲みます ("**02173**") が、**2** 行目以降では不要です。
- 矢印キーは内容に依存します。
セルを選択して新しいデータ入力後に矢印キーを使用すると、この変更を反映して次のセルに移動します。セルをダブルクリックすると内容を編集できます。この場合、左右の矢印キーはセル内のデータ上でカーソルを移動させます。
- セルに施した変更点は破棄できます。

データ入力中に変更をキャンセルするときは、**Esc** キーを押すか他のセルを **1** 回クリックしてください。

6.5 変数の名前およびフォーマットの変更

「データ入力」セクションでデータを入力しましたが、変数名がデフォルトの **var1, var2, ..., ver5** となっています。これらの変数名を変更し、データセットの列タイトルに対応するようにしましょう。そして変数に説明を付け加え、さらに変数のフォーマットも変更したいと思います。

変数名、ラベル、そしてフォーマットの変更については **price** を例にしながら手順に沿って操作します。それから変数にメモを付けます。操作を始めるには、変数 **var2** を変数ウィンドウでクリックします。すると、変数 **var2** のプロパティがプロパティウィンドウに表示され、編集可能になります。これから順に変数 **var2** のプロパティを変更しましょう。

1. 名前欄で **var2** をクリックします。選択後「**price**」と打ち込んで名前を上書きします。
2. **price** の下のラベル欄をクリックします。
3. 変数を説明するラベルを設定します。たとえば「**Price in dollars**」と入力します。
4. フォーマット欄の **%9.0g** をクリックします。
5. クリックすると表示される省略符号(...)をクリックします。書式作成ダイアログが開きます。

6. ここには多くのフォーマットがあり、中でも時間に関係したものが大多数を占めます。数字の中にカンマ(,)を入りたいので書式のプロパティボックス内の 出力する数値をカンマで区切る のチェックボックスにチェックを付け、OK ボタンを押します。
7. メモ欄をクリックします。
8. クリックすると表示される省略符号(...)をクリックします。price のメモ というダイアログが開きます。
9. 追加 ボタンをクリックし、何かメモを入力します。
10. 入力が終わったら、適用 ボタンに続けて 閉じる ボタンをクリックします。変数 price にメモが付きしました。
11. メモの前に表示される ボタンをクリックするとプロパティウィンドウに先程作成したメモを表示します。

他の変数を編集するには変数ウィンドウで目的の変数をクリックします。var1 を「make」、var3 を「mpg」、var4 を「weight」、var5 を「gear_ratio」に変更します。

The screenshot shows the Stata Data Editor window with the following data table:

	make	price	mpg	weight	gear_ratio
1	VW Rabbit	4,697	25	1930	3.78
2	Olds 98	8,814	21	4060	2.41
3	Chev.Monza	3,667	.	2750	2.73
4	AMC Concord	4,099	22	2930	3.58
5	Datsun 510	5,079	24	2280	3.54
6		5,189	20	3280	2.93
7	Datsun 810	8,129	21	2750	3.55

The 'Properties' panel on the right shows the variable 'make' selected, with its label and storage type (str11) visible.

ここで、変数名に関するルールをいくつか確認しましょう。

- 大文字と小文字は区別します。

Make, make, MAKE はすべて異なる名前になります。変数名を Make、Price、MPG 等のように大文字も使用して入力した場合、今後、同じように大文字で入力する必要があります。全て小文字で入力の方が簡単でしょう。

- 変数名は **1-32** 文字の長さで入力してください。
- 使用できる記号は半角英数 (**A-Z, a-z, 0-9**)、アンダースコア (**_**)、記号以外の **Unicode** 文字です。
- 変数名の **1** 文字目はアルファベット、アンダースコア、**Unicode** 文字のいずれかにします。変数名の **1** 文字目にアンダースコアを使用出来ませんが、これはお勧めしません。**Stata** では一時的に利用する変数名には先頭にアンダースコアを付けるという流儀があります。したがって普通の変数の先頭にアンダースコアを付けることは避けるべきでしょう。

変数名と値ラベルに関する詳細は [\[GSW\] 9 Labeling data \(データのラベリング\)](#) を、表示形式に関する詳細は [\[U\] 12.5 Formats: Controlling how data are displayed](#) を参照してください。

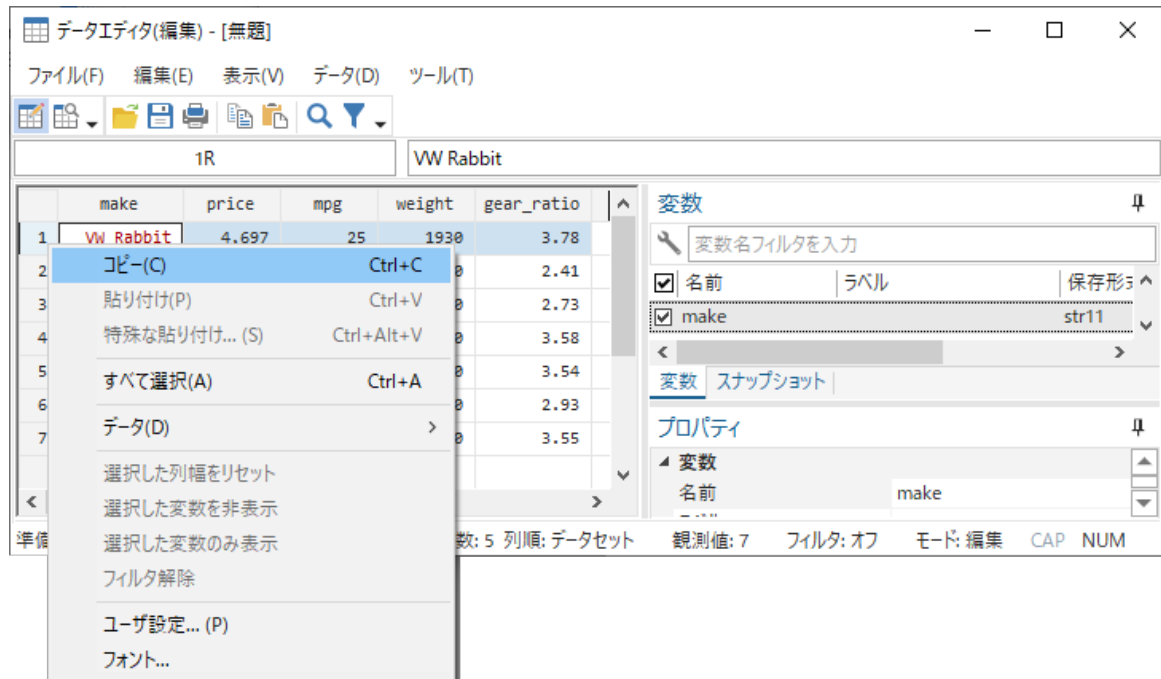
6.6 データのコピーと貼り付け

データのコピーと貼り付けにはデータエディタを使用します。他のスプレッドシートやデータベースからデータを移動する場合はこの方法を利用すると簡単です。

1. 以下の方法のうち **1** つを使い、**Stata** にコピーする内容を選択します。
 - 変数名または列ヘッダーを **1** 回クリックして列を選択します。
 - 行番号または行ヘッダーを **1** 回クリックして行を選択します。
 - マウスをクリック & ドラッグしてセルの範囲を選択します。
2. 選択した範囲内で右クリックを行い、コピーを選んでクリップボードにコピーします。
3. クリップボードのデータを貼り付けるセル範囲の左上角を右クリックします。そのセル内で右クリックして貼り付けを選びます。

コピーと貼り付けを説明するために先程の例で作成したデータの **1** 行目をコピーし、データセットの最後に貼り付けましょう。

まずは行番号 (一番左側の番号) をクリックします。これで行のデータを全て選択します。続けて右クリックし (マウスを動かす必要はありません)、コピーを選択します。



8行1列目を選択し、右クリックします。そして表示されたメニューから貼り付けを選びます。1行目がうまく複製できたことが分かります。

6.7 コピーと貼り付けに関するメモ

- 先ほどの例はデータエディタ内でのコピーと貼り付けに関するものです。この方法とほぼ同じ方法で **Stata** と他のアプリケーション間でコピーと貼り付けを行えます。**Stata** と他のアプリケーションの間でコピーと貼り付けができるかどうか確認するには、実際にやってみると良いでしょう。表計算ソフト、データベースソフト、ワープロソフトなど、コピー元のデータに何か明確な区切りがある場合はうまくコピーできるはずですが。編集 > 特殊な貼り付け... を使うと、他のフォーマットにも対応するオプションが提供されます。単純な貼り付け (**paste**) コマンドでうまくいかない時はこの、編集 > 特殊な貼り付け... を試してください。ファイルをインポートする場合についての詳細は [\[GSW\] 8 Importing data \(データのインポート\)](#) をご覧ください。
- 値ラベルがついているデータのコピーと貼り付けには選択肢があります。つまり、値ラベルをテキストとして貼り付けることができますし、元の数値データを貼り付けることができます。値ラベルをテキストとしてコピーするのがデフォルトの設定です。他のオプションを使用するにはデータエディタ

内で右クリックをして データ > 値ラベル > すべての値ラベルを非表示 と操作するか、メインメニューから ツール > 値ラベル > すべての値ラベルを非表示 と操作してください。

6.8 データを変更する

名前から予想できるようにデータエディタはデータセットの編集を行うツールです。先の例で示したように、データの編集、変数の説明や表示オプションの編集を行えます。

実際に **automobile** のデータセットに変更を加えて、データエディタとそのコマンド履歴の使用方法を見てみたいと思います。このセクションではテキストのスナップショットも取りながら作業します。これで何かを間違えた場合は、間違える前のデータセットに戻すことができます。


これからデータセットの調査、値ラベルの作成、変数 **trunk** の削除、**100** マイル (約 **160km**) あたりのガソリン消費量を示す新しい変数を作成します。この一連の操作でデータエディタを使用する時の基礎を学べます。

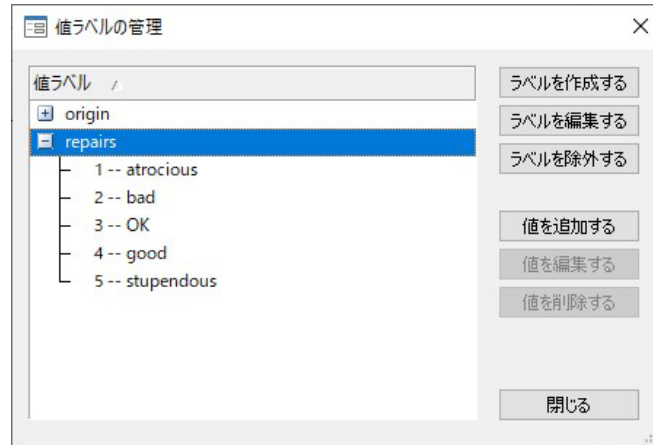
はじめに、コマンドウィンドウに「**sysuse auto**」と入力します。このセクションの前に例題を行っている場合、エラーが表示されてメモリ内のデータは失われると表示されます。このメッセージはメモリ上にある編集済みのデータを削除する前に、ユーザに保存の確認を求めるものです。既存のデータセットを保存する場合は ファイル > 保存 と操作してファイルを目的のフォルダに保存します。保存しない場合はコマンドウィンドウに「**clear**」と入力して **Enter** キーを押し、データを削除します。その後、**auto** データセットをロードしてください。

auto のデータセットをロードしたらデータエディタを表示してください。

1. 祖父の乗っていた車が **Toronado** でした。お酒落な車である一方、大量にガソリンを必要とした気がします。この車がデータセットに掲載があるか調べるとします。編集 > 検索 を選択し、**Toronado** と入力して **Enter** キーを押します。その結果、この車の情報を発見し、燃費が **16 mile/gallon** であったことが分かります。

2. この中で最も燃費が良い車と悪い車を調べましょう。まず、**mpg** 列の列ヘッダーを右クリックします。そして、コンテキストメニューから データ > データをソート... を選択します。どのようにソートを行うのか尋ねるダイアログが表示されます。デフォルトは昇順です。**OK** をクリックします。(Stata にはリサンプリング機能があるので、並び替えを行う場合は注意が必要です。) データが **mpg** の昇順で並べ替えられます。燃費が最悪い車 (**mpg** の数値が小さい車) がスクリーンの上にあります。データをスクロールダウンするとデータセットの最後に燃費が最も良い車があります。変数 **mpg** を選択した後に ツール > データをソート... とメニューから選んでも同じ結果になります。
3. 修理記録も比較したいので、次は変数 **rep78** でソートしてみましょう。(今行ってください。) **Starfire** と **Firebird** の修理記録は良くありません。ですが今は修理記録が良い車を調べたいので、この画面では分かりません。データセットの下までスクロールダウンもできますが、今回はカーソル位置ボックスを使う方が速いです。「**rep78 74**」と入力し **Enter** キーを押すと、**rep78[74]** セルがアクティブになります。**rep78** の最後の 5 つのセルは点 (.) になっており、これは欠損値を表します。ここで、いくつか注意点があります。
 - ソート結果から分かるように、Stata は欠損値を数値より大きいものとして扱います。技術的に書くと、「**rep78 >= .**」は **missing(rep78)** と同じ事になります。
 - このデータを見ただけでは分からない事として、Stata は複数の欠損値指標を使用します。(.) は Stata のデフォルトまたはシステム欠損値指標で、**.a, .b, ..., .z** は Stata の拡張欠損値です。拡張欠損値は欠損の理由を区別するのに役立ちます。
 - 各欠損値指標は欠損値内でソートし、(**. < .a < .b < ... < .z**) のようになります。[\[U\] 12.2.1 Missing values](#) をご覧ください。
4. この変数 **rep78** の値に意味を持たせたいと思います。変数ウィンドウの **rep78** をクリックします。
5. プロパティウィンドウの 値ラベル 欄をクリックし、省略符号 (...) のボタンをクリックします。これで値ラベルの管理ダイアログを開きます。では新しい値ラベルを定義しましょう。
 - (a) ラベルを作成する ボタンをクリックします。するとラベルを作成するダイアログが開きます。
 - (b) ラベル名、たとえば「**repairs**」をラベル名ボックスに入力します。
 - (c) **Tab** キーを押すか値ボックス内をクリックします。
 - (d) 値に **1** と打ち込んで、**Tab** キーを押します。そして「**atrocious**」をラベルに入力します。
 - (e) **Enter** キーを押して値とラベルのペアを作成します。

- (f) ステップ **d** と **e** を繰り返してすべてのペアを作成します。2 を「**bad**」、3 を「**OK**」、4 を「**good**」、5 を「**stupendous**」として作成しましょう。
- (g) **OK** をクリックして値ラベルを作成するのを終了します。
- (h) 折りたたみ解除ボタン  をクリックして作成したラベルを確認します。



間違いがある場合、ラベルを編集するボタンを押してラベルを編集します。

- (i) 閉じる ボタンを押して値ラベルの管理ダイアログを閉じます。ラベルを作成したので、**rep78** 変数にラベルを付けます。値ラベル欄の右側にある下矢印ボタンをクリックして **repairs** ラベルを選択してください。すると値の代わりにラベルを表示します。
6. 仮に欠損値ではなく、元のデータが分かった時は **rep78** にその値を入力できます。その場合、直接セルに値を入力すれば大丈夫です。または欠損値のセル内を右クリックし、データ > 値ラベル > 変数 '**repairs**' に値ラベルを割当 と選択してラベルを選択します。値ラベルに複数の選択肢があるときに便利です。
7. 次に変数 **trunk** を削除します。変数 **trunk** の変数名 (列の一番上) で右クリックを行い、表示するコンテキストメニューでデータ > 選択範囲を削除 を選びます。データをメモリ上から削除する場合は、確認のダイアログを表示します。そこで、はい ボタンをクリックします。
8. 最後に、**100** マイル (約 **160km**) を走るのに必要なガソリンの量を表す変数を **1** つ作成しましょう。
- (a) どのセルでも構わないので右クリックをし、データ > 変数を追加... を選択して **generate** ダイアログを開きます。
- (b) 変数名欄に「**gp100m**」と打ち込みます。
- (c) 値または式を指定する のラジオボタンが選択されていることを確認してから「**100/mpg**」と入力します。右隣りにある 作成... ボタンを押すと数式ビルダダイアログ

を表示しますが、この数式は簡単なので直接入力します。(ここでちょっと休憩をして数式ビルダの機能を確認するのも良いでしょう。)

- (d) 新たな変数の位置 欄には、データセットの最後に追加するとあるのを確認します。
- (e) **OK** をクリックします。右にスクロールすると新しく作成した変数を確認できます。

このデータ編集セッションではデータエディタを使用してデータを編集してきました。結果ウィンドウを見てみると、コマンドとその出力結果を確認できます。また、履歴ウィンドウではデータエディタで実行したコマンドも見ることが可能です。編集コマンドを後で使用する場合は、次の操作で保存します。

1. データエディタが最後に出力したコマンドを履歴ウィンドウ内でクリックします。
2. データエディタを開いてからすぐに実行したコマンド「`sort mpg`」をスクロールで探し、**Shift+クリック**してこれまでの内容を選択します。
3. 選択したコマンド上で右クリックします。
4. 選択範囲を **do** ファイルエディタへ送る を選びます。

このように操作することによって、選択したコマンドを **do** ファイルエディタに保存できます。この後、再実行できます。**do** ファイルエディタについては [\[GSW\] 13 Using the Do-file Editor— automating Stata \(do ファイルエディタを使用する—Stata の自動化\)](#) で説明します。**do** ファイルについてのヘルプは [\[U\] 16 Do-files](#) をご覧ください。

もしこのデータセットを保存する場合はメインメニューで **ファイル>名前を付けて保存...** と操作し、新しい名前を付けて保存します。元のデータセットを上書きしないように注意しましょう。

6.9 スナップショットで作業する

データエディタでは作業中のデータセットならいつでもスナップショットを残すことができます。スナップショットは **Stata** を閉じる時に削除されるので、一時ファイルの扱いになります。それでもスナップショットの用途は幅広く、次のような使用例があります。

- データを一時的なコピーとしてメモリに残して、他のデータセットを開いて見ることができます。
- 作業内容を保存できるので、万一データを失くした時でも再現できます。
- 分析中のデータセットを自由に加工して保存できます。

引き続き **auto.dta** を使用します。このセクションから作業を開始する場合はコマンドウィンドウに「sysuse

auto」と入力してデータセットを開きます。(「data in memory being lost」というエラーメッセージを表示した場合、「clear」をするか既存データを保存してください。詳しくは[\[GSW\] 5 Opening and saving](#)

Stata

datasets (Stata のデータセットを開く・保存する) をご覧ください。) データエディタを開いてウィンドウ右上のスナップショットタブをクリックすると、次のようなウィンドウが開きます。データセットを開くところから作業を始めている場合は、変数 **rep78** は値ラベルではなく数値を表示します。



開始直後、スナップショットツールバーにはアクティブなボタンは1つしかありません。アクティブなボタン、追加ボタン $+$ をクリックします。するとダイアログが開きますので、スナップショット用の名前(ラベル)の入力を行います。

今回は「Start」というラベルを付けましょう。そして **Enter** を押します。スナップショットはリストに追加され、ツールバーにある他のボタンもアクティブになります。スナップショットウィンドウには次のボタンが表示されます。

$+$	追加	時間スタンプとラベルの付いた新しいスナップショットを保存します。
$-$	はずす	スナップショットを削除します。これは一時的なスナップショットファイルを削除しますがメモリ内のデータには影響しません。
	ラベルの変更	選択したスナップショットのラベルを編集します。

Restore	リストア	メモリ内のデータを選択したスナップショットのデータで置き換えます。データを置き換える場合、操作を確認するダイアログを表示します。
---------	------	--

ではいま紹介したデータエディタのツールを実際に使用してデータセットの内容を変更してください。変更後に新たなスナップショットを作成し、「**Changed**」とラベルを付けます。スナップショットウィンドウを開いて「**Start**」を選択し、ダブルクリックをするかリストアボタンを押して「**Start**」を復元します。そして先程の変更したファイルに戻るには「**Changed**」スナップショットを復元します。

これらのスナップショットは、削除するか **Stata** の該当セッションを閉じない限り使用できます。つまり、**1**つのデータセットで作業しながら、別のデータのスナップショットを開くことができます。スナップショットの用途は千差万別なので、ご自身に合ったものを探してみてください。ただし、スナップショットは一時的なファイルなので、後から利用するデータはしっかりと保存してください。

6.10 データエディタと日付

データエディタには日付を取り扱う **2**つの特殊なツール(フォーマットと日付マスク)があります。これらの動作を確認するには他のデータセットを開く必要があります。データセットを保存するか **clear** をして、コマンドウィンドウに「**sysuse sp500**」と入力します。データエディタを開いて、内容を見てみましょう。

The screenshot shows the Stata Data Editor interface. The main window displays a table with the following data:

	date	open	high	low	close
1	02jan2001	1320.28	1320.28	1276.05	1283.27
2	03jan2001	1283.27	1347.76	1274.62	1347.56
3	04jan2001	1347.56	1350.24	1329.14	1333.34
4	05jan2001	1333.34	1334.77	1294.95	1298.35
5	08jan2001	1298.35	1298.35	1276.29	1295.86
6	09jan2001	1295.86	1311.72	1295.14	1300.8
7	10jan2001	1300.8	1313.76	1287.28	1313.27
8	11jan2001	1313.27	1332.19	1309.72	1326.82
9	12jan2001	1326.82	1333.21	1311.59	1318.55
10	16jan2001	1318.32	1327.81	1313.33	1326.65
11	17jan2001	1326.65	1346.92	1325.41	1329.47
12	18jan2001	1329.89	1352.71	1327.41	1347.97

The right-hand pane shows the variable properties for 'date':

名前	ラベル	保存形式
date	Date	int
open	Opening price	float
high	High price	float

At the bottom of the window, the status bar indicates: 準備完了 (Ready), 変数: 7 列順: データセット (Variables: 7, Column order: Data set), 観測値: 248 (Observations: 248), フィルタ: オフ (Filter: Off), モード: 編集 (Mode: Edit), CAP NUM.

データの最初の日付として **2001 年 1 月 2 日 (January 2, 2001)** と記載してある日付変数 **date** があります。この表示形式は **Stata** のデフォルト形式です。

まずは日付形式を変更することから始めましょう。

1. データエディタウィンドウの右側にある変数ウィンドウ内で変数 **date** を選びます。
2. プロパティウィンドウでフォーマットを選び、省略記号(...)ボタンをクリックします。
3. 書式作成ダイアログからこの日付形式について **3** つの情報を読み取れます。
 - このデータは日次(**daily**)の時系列データです。ダイアログを見ても分かるように、**Stata** は金融業界等で使用する日付の種類にも対応します。
 - ダイアログの下には、**Stata** のデフォルト日付形式として「**%td**」を表示します。このフォーマット記号のついている変数は日次の時系列データであると解釈します。
 - このデフォルト形式は、たとえば **07apr2021** ように表示されます。(これはデータエディタで見ると一目でわかります。)
4. 書式作成ダイアログの右上にあるサンプルにはさまざまな日付形式が用意されています。ここではその中から「**April 07, 2021**」をクリックします。どのようにこの形式が記号で記述されるのかをダイアログの下で見ることができます。
5. **OK** をクリックして書式作成ダイアログを閉じます。日付の表示形式が変更されました。

この方法で日付形式を簡単に変更できます。日付と日付形式に関する情報は [\[D\] Datetime](#) をご覧ください。

それでは、日付をいくつか変更し、形式の変更を簡単にできる様子を確認しましょう。これは実際の表示形式がどのような場合でも同じです。データエディタの左上角を見ると、時間/日付入力マスク(**Time/Date input mask**) エリアがあり、そこには **DMY** と記載されています。データ編集時の日付表示形式はここで変更します。デフォルトでは、このマスクは **DMY** となっています。これは日付要素の入力順番が日・月・年ならば、日付は様々な様式で入力できることを示しています。実際にやってみましょう。

1. 変数 **date** の 1 行目の観測値をクリックします。するとカーソル位置ボックスには **date[1]** と表示します
2. 「**18jan2021**」と入力して **Enter** キーを押します。Stata は **DMY** 入力マスクを理解し、選択したセルに新しい日付を入力します。
3. 「**30042021**」と入力して **Enter** キーを押します。今回、デリミタ (区切り) はありませんが、入力マスクは入力した情報を正確に認識して表示します。
4. 時間/日付入力マスクエリア内をクリックし、ドロップダウンメニューから **MDY** を選びます。
5. 変数 **date** 内にある観測値のどれかをクリックします。
6. 「**March 15, 2021**」と入力して **Enter** キーを押します。すると、他の観測値と同じ形式で表示します。

結果ウィンドウには、データ編集のコマンドを表示します。この場合、データエディタへの直接入力の方が効率的です。

次は他のデータセットを利用しますので「**clear**」と打ち込み、**Enter** を押しましょう。

6.11 データエディタのアドバイス

先の例からも分かるように、データエディタ内の小さな間違いがデータセットに大きな問題を引き起こすこともあります。データ編集の方法には細心の注意を払わなければなりません

- 生データの管理に気を配る人はデータエディタのような編集機能は危険だと感じるはずですが。なぜなら、うっかりデータを変更してしまう可能性があるからです。データを表示するだけなら、データエディタの編集モードを使わないでください。代わりにデータエディタのブラウズモードを使用してください(あるいは **browse** コマンドを使用してください)。
- データを編集する場合、データセットの表示領域を制限すれば誤ってデータを変更することはありません。たとえば、**rep78** の欠損値だけを直す必要がある場合、その欠損値だけを表示するように工夫してください。これで編集する変数および観測値以外の値は変更 (破損) できなくなります。この方法は後で実際に試します。
- 注意をしても間違いは起きてしまうことがあります。それでもデータエディタは結果ウィンドウにコマンドの記録があるので他のソフトよりもデータの破損等に関しては安心です。この機能を使っ

て出力結果をログとして記録し、変更の記録を作ることができます。この記録から変更が意図したものと一致しているか確認できます。ログファイル作成についての情報は [\[GSW\] 16 Saving and printing results by using logs](#) (ログを使い結果の保存や印刷を行う) をご覧ください。

6.12 フィルタと非表示

これからデータエディタ内で表示制限する方法を紹介します。データエディタのアドバイスセクションで説明した表示制限に使用するだけでなく、大きなデータセットの内容を見る際にも役立ちます。どちらの場合でも、データ全てではなくデータの一部、たとえば変数や観測値の一部を表示する時にとっても便利です。更に、例題の中で変数の順序も変更します。この操作方法はグラフィカルなインターフェイスとコマンドの両方で紹介します。

コマンドウィンドウに「`sysuse auto`」と打ち込んで、データセット **automobile** データセットを開きます。もしエラーメッセージが出てきたら、**clear** を実行して再度試してください。データセットをうまく読み込んだら、データエディタを開きます。


たとえば、変数 **rep78** の欠損値のみを編集するとしましょう。どの観測値について作業をしているのか確認するために車のメーカー (**make**) も表示しますが、他の変数は必要ありません。ここではとても大きなデータセットで作業を行っているとは仮定して操作します。

1. 作業を始める前にデータエディタウィンドウの変数ウィンドウで次の操作を試してください。
 - (a) 変数をリスト上でドラッグ&ドロップして配置を変えます。するとデータエディタ内の変数列の順番が変わります。データセットの元の順番には変更ありません。
 - (b) 1列目のチェックボックスからチェックを数個外し、データエディタ内でその変数を非表示にします。
 - (c) ウィンドウ上部の変数名フィルタを入力欄にキーワードを入力します。メインウィンドウの変数ウィンドウのように、デフォルトでは変数または変数ラベルについて、大文字と小文字の区別をつけずにフィルタ内の言葉について検索します。左側にあるレンチをクリックすると、この機能を変更できます。また変数の情報を含む列の追加と削除も行えます。変数ウィンドウで行うフィルタリングは変数ウィンドウ上の表示を変更するもので、観測値の表示は制御しません。一通り終了したらフィルタのテキストを消去します。

2. 変数ウィンドウの変数（どれでも構いません）を右クリックしてコンテキストメニューからすべて選択を選びます。
3. 一度、チェックのついているチェックボックスの1つをクリックし、すべてのチェックを外します。
4. 変数 **make** をクリックして選択し、他の全ての変数の選択を外します。
5. 変数 **make** のチェックボックスをクリックします。
6. 変数 **rep78** のチェックボックスをクリックします。

結果ウィンドウを見ると、コマンドとしては何も実行していません。データエディタで変数を非表示してもデータセットには影響しません。データエディタでの表示だけに影響します。

これで作業する変数だけを表示できたので、間違える可能性を減らしました。では変数 **rep78** で欠損値となっている観測値を表示しましょう。これは簡単に行えます。

1. データエディタツールバーの観測値フィルタボタン  をクリックします。
2. 式によるフィルタ欄に「**missing(rep78)**」と入力します。
3. フィルタを適用するボタンをクリックします。
4. 関数の表記がよくわからない時は式によるフィルタ欄右側にある省略記号 (...) をクリックしてください。数式ビルドダイアログを開きます。このダイアログは **Stata** で使用できる関数のリストを表示します。詳しくは [Stata Functions Reference](#) マニュアルをご覧ください。

これで作業を行う部分だけを表示しました。些細なキーボード操作のミスなどで誤ってデータを削除または変更する可能性も無くなりました。

コマンドウィンドウからデータエディタ内の変数を非表示にし、観測値にフィルタをかける方法も確認しましょう。このコマンドを使用すると先程行ったような表示制限を簡単に行えるようになります。コマンドウィンドウから作業を行う場合は、**edit** コマンド、**varlist** (変数リスト)、**if** と **in** 条件を使用してコマンド文を入力します。**varlist** で表示する変数を制限し、**if** と **in** 条件で観測値の制限を行います ([\[GSW\] 10 Listing data and basic command syntax](#) (データのリストと基本コマンドの構文)にはコマンド文と共に **varlist** と **if** および **in** 条件の使用についての例が多くあります)。では変数 **rep78** の欠損値を修正しましょう。最低限必要な情報は変数

make と変数 **rep78** です。最小限の表示で操作を行い、間違える可能性をできるだけ小さくするには次のコマンド文を入力します。

```
. sysuse auto
(1978 automobile data)

. edit make rep78 if missing(rep78)
```


実行すると、次のような画面が表示されます。



これで間違える可能性を大幅に減らしました。

ここで学んだ内容を忘れずに、今後のデータ編集を行ってください。

6.13 ブラウズモード

誤ったキー操作等でデータを変更したくない時は、データエディタのブラウズモードを使用します。データエディタをブラウズモードで開くためにはデータエディタ (ブラウズ) ボタンをクリックするか、コマンドウィンドウで「**browse**」を実行します。ブラウズモードで作業をしているとき、データを変更できるすべてのコンテキストメニュー、変数のラベル、またはディスプレイ形式に関する操作はできなくなっています。変数のプロパティはプロパティウィンドウで見ることができますが、変更はできません。しかし、観測値や変数を非表示にして表示領域を制限することはできます。この操作はデータセットの変更を伴わないのでブラウズモードでも実行できます。

メモ：データエディタがブラウズモードでも、**Stata** のメニューやコマンドウィンドウを使用して操作すればデータは変更できます。実行したコマンドがどのようにデータセットに影響するのか、直接見ることができます。データエディタの編集モードは本当にデータを変更するときのみ使用してください。

6.14 列ヘッダでの変数ラベル表示と列幅の調整

変数ラベルを列ヘッダに表示することもできます。表示メニューから列ヘッダに変数ラベルを表示するを選択します。変数ラベルの表示が見切れる場合は、列幅をドラッグして調整できます。

The screenshot shows the Stata Data Editor window for 'auto.dta'. The main table displays data for variables: make, price, mpg, and rep78. The 'make' variable has labels like 'AMC Concord', 'AMC Pacer', etc. The 'price' variable has values like 4,099, 4,749, etc. The 'mpg' variable has values like 22, 17, etc. The 'rep78' variable has values like 3, 3, etc. The column headers are 'make', 'price', 'mpg', and 'rep78'. The column widths are adjusted so that the variable labels are visible. The right-hand pane shows the '変数' (Variables) list with '名前' (Name) and 'ラベル' (Label) columns. The 'make' variable is selected, and its label 'Make and model' is visible. The bottom status bar shows '準備完了' (Ready), '文字数: 18' (Character count: 18), '変数: 15' (Variables: 15), '列順: データセット' (Column order: Data set), '観測値: 74' (Observations: 74), 'フィルタ: オフ' (Filter: Off), 'モード: ブラウズ' (Mode: Browse), 'CAP NUM'.

	make	price	mpg	rep78
	Make and model	Price	Mileage (mpg)	Repair record 1978
1	AMC Concord	4,099	22	3
2	AMC Pacer	4,749	17	3
3	AMC Spirit	3,799	22	.
4	Buick Century	4,816	20	3
5	Buick Electra	7,827	15	4
6	Buick LeSabre	5,788	18	3
7	Buick Opel	4,453	26	.

列幅をドラッグして全ての変数ラベルが表示されるように調整してください。列幅の情報はデータセットと一緒に保存されます。列幅よりも長いテキストは見切れて表示されます。データエディタは表示が見切れたセルの情報をツールチップで表示します。セルの上にマウスを重ねると、テキスト全体がツールチップとして表示されます。また、セルをダブルクリックして編集モードにすることで、長いテキストを確認できます。

6.15 行と列の固定

データエディタでは行と列の固定ができます。固定された行または列はスクロールされず、常に表示されます。「sysuse census」とコマンド実行して別のデータに切り替えましょう。これは1980年のアメリカ合衆国の国勢調査データです。データエディタをブラウザモードで起動しましょう。50州のデータがありますが、画面サイズに限りがあるため、全ての観測値と変数を表示することができません。また、ある州や観測値を固定して、他の州と比較したいこともあるでしょう。次の例を見てみましょう。

任意のセル上で右クリックすると、選択した行または列を固定するは無効になっています。

データエディタ(ブラウズ) - [census.dta]

ファイル(F) 編集(E) 表示(V) データ(D) ツール(T)

state[1] Alabama

state	state2	region	pop	poplt5
1 Alabama	Al	South	3,893,888	296,412
2 Alaska				38,949
3 Arizona				213,883
4 Arkansas				175,592
5 California				708,400
6 Colorado				216,495
7 Connecticut				185,188
8 Delaware				41,151
9 Florida				570,224
10 Georgia				414,935
11 Hawaii				77,848
12 Idaho				93,531
13 Illinois				842,241
14 Indiana				418,764
15 Iowa				221,628
16 Kansas				180,877
17 Kentucky				282,731

変数

変数名フィルタを入力

名前 | ラベル

state | State

state2 | Two-letter state abbrevi...

region | Census region

pop | Population

poplt5 | Pop, < 5 year

pop5_17 | Pop, 5 to 17 years

pop18p | Pop, 18 and older

変数 スナップショット

プロパティ

変数

準備完了 文字数: 14 変数: 13 列順: データセット 観測値: 50 フィルタ: オフ モード: ブラウズ CAP NUM

ある変数（例えば state）の列ヘッダ上で一度クリックすると列がハイライト表示されます。そのまま右クリックすると先程と似たメニューが表示されますが、**選択した変数を固定する**が有効になっています。一つ以上の変数を選択して固定できます。

データエディタ(ブラウズ) - [census.dta]

ファイル(F) 編集(E) 表示(V) データ(D) ツール(T)

1C Alaska

state	pop	poplt5	pop5_17	pop18p
1 Alabama	96,412	13,883	75,592	08,400
2 Alaska	85,188	41,151	70,224	14,935
3 Arizona	77,848	93,531	42,241	18,764
4 Arkansas	21,628	80,877	82,731	
5 California				
6 Colorado				
7 Connecticut				
8 Delaware				
9 Florida				
10 Georgia				
11 Hawaii				
12 Idaho				
13 Illinois				
14 Indiana				
15 Iowa				
16 Kansas				
17 Kentucky				

準備完了 文字数: 14 変数: 13 列順: データセット 観測値: 50 フィルタ: オフ モード: ブラウズ CAP NUM

州の情報を固定した後、右方向にスクロールすると、先程まで見えなかった変数（marriageやdivorceなど）が見えます。州の情報は左側に固定されています。

データエディタ(ブラウズ) - [census.dta]

ファイル(F) 編集(E) 表示(V) データ(D) ツール(T)

state[2] Alaska

state	death	marriage	divorce
1 Alabama	35,305	49,018	26,745
2 Alaska	1,604	5,361	3,517
3 Arizona	21,226	30,223	19,908
4 Arkansas	22,676	26,513	15,882
5 California	186,428	210,864	133,541
6 Colorado	18,925	34,917	18,571
7 Connecticut	26,005	26,048	13,488
8 Delaware	5,123	4,437	2,313
9 Florida	104,190	108,344	71,579
10 Georgia	44,230	70,638	34,743
11 Hawaii	4,849	11,856	4,438
12 Idaho	6,753	13,428	6,596
13 Illinois	102,230	109,823	50,997
14 Indiana	47,300	57,853	40,006
15 Iowa	26,348	27,474	11,854
16 Kansas	21,910	24,847	13,410
17 Kentucky	33,765	32,727	16,731

準備完了 文字数: 14 変数: 13 列順: データセット 観測値: 50 フィルタ: オフ モード: ブラウズ CAP NUM

同様に、一つ以上の観測値をハイライトして右クリックすると、行を固定できます。

The screenshot shows the STATA Data Editor window for a dataset named 'census.dta'. The window title is 'データエディタ(ブラウズ) - [census.dta]'. The menu bar includes 'ファイル(F)', '編集(E)', '表示(V)', 'データ(D)', and 'ツール(T)'. The toolbar contains icons for file operations, editing, and viewing. The main area displays a table with columns: 'state', 'state', 'state2', 'region', and 'pop'. The first three rows (Alabama, Alaska, Arizona) are highlighted in blue. A context menu is open over these rows, listing various actions such as 'コピー(C)' (Ctrl+C), '貼り付け(P)' (Ctrl+V), 'すべて選択(A)' (Ctrl+A), and '選択した観測値を固定する' (Fix selected observations), which is currently selected. The status bar at the bottom indicates '準備完了' (Ready), '文字数: 14' (Character count: 14), '変数: 13 列' (Variables: 13 columns), and 'モード: ブラウズ' (Mode: Browse).

row	state	state	state2	region	pop
1	Alabama	Alabama	AL	South	3,893,888
2	Alaska	Alaska	AK	West	
3	Arizona	Arizona	AZ	West	
4	Arkansas	Arkansas	AR	South	
5	California	California	CA	West	
6	Colorado	Colorado	CO	West	
7	Connecticut	Connecticut	CT	NE	
8	Delaware	Delaware	DE	South	
9	Florida	Florida	FL	South	
10	Georgia	Georgia	GA	South	
11	Hawaii	Hawaii	HI	West	
12	Idaho	Idaho	ID	West	
13	Illinois	Illinois	IL	N Cent	
14	Indiana	Indiana	IN	N Cent	
15	Iowa	Iowa	IA	N Cent	
16	Kansas	Kansas	KS	N Cent	
17	Kentucky	Kentucky	KY	South	

観測値の1から3行目を固定し、垂直方向にスクロールして、最初の3つの州と他の州を比較しましょう。

データエディタ(ブラウズ) - [census.dta]

ファイル(F) 編集(E) 表示(V) データ(D) ツール(T)

state[1] Alabama

	state	death	marriage	divorce
1	Alabama	35,305	49,018	26,745
2	Alaska	1,604	5,361	3,517
3	Arizona	21,226	30,223	19,908
39	Rhode Island	9,300	7,490	3,606
40	S. Carolina	25,138	53,915	13,595
41	S. Dakota	6,523	8,800	2,811
42	Tennessee	40,713	59,175	30,206
43	Texas	108,019	181,762	96,809
44	Utah	8,103	16,958	7,802
45	Vermont	4,587	5,226	2,623
46	Virginia	42,496	60,210	23,615
47	Washington	32,012	47,728	28,642
48	W. Virginia	19,186	17,391	10,273
49	Wisconsin	39,255	41,111	17,546
50	Wyoming	3,215	6,868	4,003

変数

変数名フィルタを入力

- 名前 | ラベル
- state | State
- state2 | Two-letter state abbrevi...
- region | Census region
- pop | Population
- poplt5 | Pop, < 5 year
- pop5_17 | Pop, 5 to 17 years
- pop18p | Pop, 18 and older

変数 スナップショット

プロパティ


変数

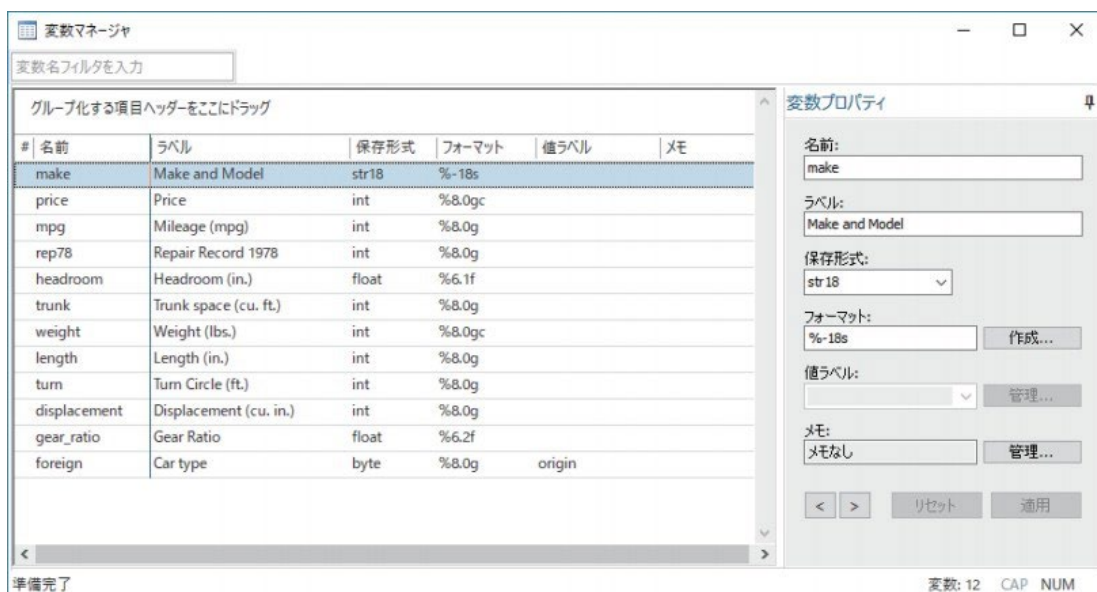
準備完了 文字数: 14 変数: 13 列順: データセット 観測値: 50 フィルタ: オフ モード: ブラウズ CAP NUM

行/列の固定はブラウズモードと編集モードの両方で使用できます。行と列を同時に固定することもできます。固定した行または列の上で右クリックして、**ピン留めされた観測値/変数をクリアする**を選択すると固定が解除されます。

7. 変数マネージャを使用する

7.1 変数マネージャ

この章では **Stata** の変数マネージャについて説明します。コマンドウィンドウで「`sysuse auto,`
`clear`」を実行して **automobile** データセットを開きましょう。変数マネージャを開くにはメインメニューからデータ > 変数マネージャ と操作するか、 ボタンをクリックします。



変数マネージャは変数のプロパティを管理するもので、1つまたは複数の変数を同時に管理できます。変数マネージャでは変数および値ラベルの作成、変数名の変更、表示形式の変更、そしてメモの管理を行えます。また変数をフィルタにかけてグループにまとめることも、変数リストを作成することもできます。この機能は大きなデータセットを管理する時に便利です。

変数マネージャで行った操作は、コマンドウィンドウに直接入力したように結果ウィンドウに表示します。つまり、操作の記録を取ると同時に、変数マネージャを利用することによってコマンドを学べます。

7.2 変数ペイン

変数マネージャの左側ペインは、スクリーン上では特に記載されていませんが、変数ペインと呼ばれます。このペインはデータセット内の変数リストを表示します。このリストは様々な方法で表示できます。

- 左上角にあるフィルタボックスにテキストを入力すると変数にフィルタをかけます。これは似たような名前がっている変数および値ラベルのみを表示するのに役立ちます。
- リストは列タイトルをクリックするとソートできます。
 - (a) 列タイトルを1回クリックすると昇順で表示します。
 - (b) 同じ列タイトルを2回クリックすると変数を降順でソートします。
 - (c) シャープ (#) をクリックすると、ソート順を元の変数の並び順に戻します。
- ソート順は変数マネージャウィンドウの表示のみ変更します。データセット自体の順番は変更出来ません。
- 列タイトルをドラッグ&ドロップで移動すると列の順番を変更できます。デフォルトの順番に戻すには、列タイトルを右クリックして列をデフォルトに戻すを選びます。
- 変数はカテゴリー（列）の値でグループにまとめることができます。列タイトルをウィンドウ上部の **grouping bar**（フィルタと列ヘッダーの間）にドラッグ&ドロップするとグループ化します。グループ化を取り消すには、キャンセルする列タイトルを元の列タイトルバーにドラッグします。以下は **auto.dta** のデータセットを列 **type** でグループ化した例です。



変数ラベルでグループ化するとラベリングしていない変数を確認するのに便利です。

7.3 変数ペインで右クリックする

変数ペインで右クリックすると、多くのタスクを実行できるコンテキストメニューを表示します。

- **変数のプロパティを編集** 変数プロパティペイン（ウィンドウの右側の部分）をアクティブにします。自動的に非表示となっている場合、変数プロパティペインを表示します。
- **選択した変数のみ維持** 選択した変数のみデータセット内に残し、他の変数をドロップ（削除）します。
- **選択した変数を削除** 選択した変数をデータセット内から削除します。
- **選択した変数のメモを管理...** 1つの変数にメモを付けたり削除するウィンドウを開きます。複数の変数を選択している場合、この機能は使用できません。
- **データセットのメモを管理...** データセット全体にメモを付けたり削除するウィンドウを開きます。
- **変数リストをコピー** 選択した変数名をクリップボードにコピーします。
- **すべて選択** 表示しているすべての変数を選択します。フィルタの影響で変数が非表示である場合、この操作では選択されません。
- **コマンドウィンドウに変数リストを送る** 選択した変数名をコマンドウィンドウに入力します。これはソートやグループ化と組み合わせると、大きなデータセット内で変数リストを作成するのに便利です。
- **印刷...** 変数ペインを印刷します。変数マネージャで列幅を変えると、印刷される列幅を変更できます。

7.4 変数プロパティペイン

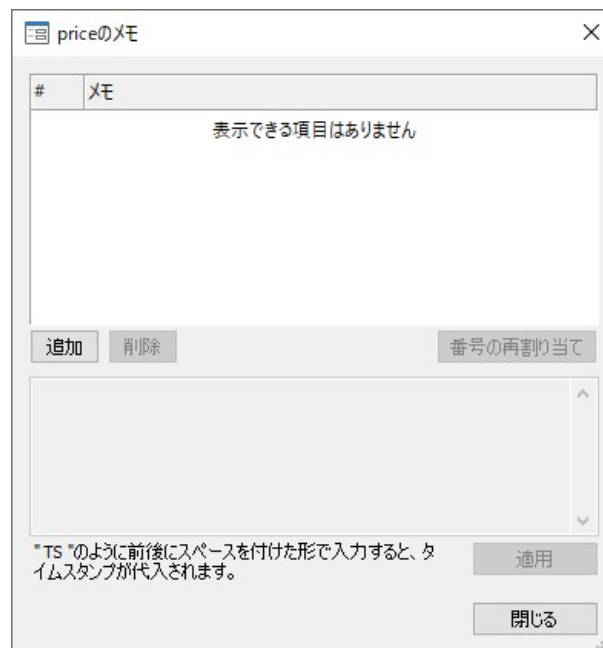
変数ペインで選択した変数（行）のプロパティを変更する時は変数プロパティペインを利用します。変数を1つ選択した場合、その変数のプロパティをすべて変更できます。複数の変数を選択している場合、それらの形式、タイプ、値ラベルを一度に変更できます。このペインは [\[GSW\] 6 Using the Data Editor \(データエディタを使用する\)](#) 内の、「[変数の名前およびフォーマットの変更](#)」では説明したダイアログと同じように操作できます。また次のセクションで示すように **Stata** では変数およびデータセットにメモを追加できます。

変数プロパティペインは [GSW] 2 The Stata user interface (Stata のユーザインターフェイス) の「自動非表示とピンニング」で説明したドッキングウィンドウです。これは画鋲アイコンがウィンドウ右上にあることから分かります。画鋲アイコンをクリックすると不使用時にはウィンドウが自動的に非表示になります。また変数マネージャウィンドウ内でタイトルバーをドラッグすると、他の場所にドッキングできます。

7.5 メモを管理する

Stata では各変数またはデータセット全体にメモを追加できます。このメモはシンプルなテキストメモで、自由に記入できます。たとえばデータセットの出所、変数に関連したデータ採取時の備考、変数の分析手法についてなど、どのようなものでも構いません。

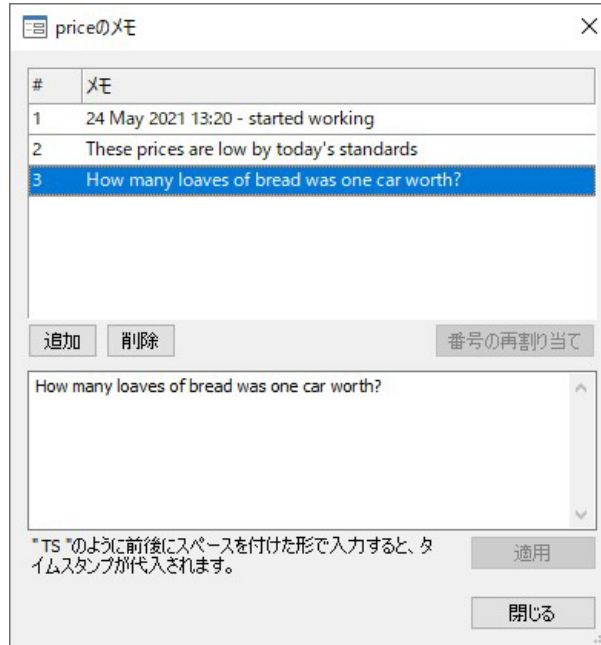
まずは変数ペインで変数を選択することから始めましょう。例として変数 **price** を使用します。変数プロパティペイン内のメモ欄の右隣りにある、**管理...** ボタンをクリックし、以下のダイアログを開きます。



ではメモを追加しましょう。

1. 追加ボタンをクリックしてメモを追加します。

2. 「**TS - started working**」と入力します。「**TS**」とそれに続くスペースで時間スタンプをメモに挿入します。
適用をクリックするとメモを作成します。
3. あと**2**つメモを入力します。下図のように価格についてのメモを追加しましょう。



ではメモの追加・削除・編集をやってみましょう。メモは長期プロジェクト中などに書き残しておくと便利です。メモマネージャ（メモを入力するダイアログ）でメモを変更すると、その度に結果ウィンドウにコマンドを出力します。

8. データをインポートする

8.1 コピーと貼り付け

データをインポートする最も簡単な方法は、表作成のできるアプリケーションからデータをコピーし、データエディタに直接貼り付けることです。これはすべてのスプレッドシート型のアプリケーションからのインポートで使用できます。また、多くのデータベース、複数のワープロやウェブページのアプリケーションにも対応しています。データ範囲すべてをコピーし、それをデータエディタに貼り付けばうまくいくはずです。さらにカンマ (,) で区切られているテキストファイルも、コピーしてデータエディタ内に貼り付けることができます。


たとえば、友人が古い車に関する小さなデータセットを持っているとします。

```
VW Rabbit,4697,25,1930,3.78
Olds 98,8814,21,4060,2.41
Chev.Monza,3667,,2750,2.73
,4099,22,2930,3.58
Datsun 510,5079,24,2280,3.54 Buick
Regal,5189,20,3280,2.93
Datsun 810,8129,,2750,3.55
```

このデータを **Stata** に取り込みましょう。この操作は簡単に行えます。

1. 現在のデータセットに「**clear**」と打ち込んでメモリ内からデータを消します。
2. 上記のデータをコピーします。
3. データエディタを編集モードで開きます。
4. 編集 > 特殊な貼り付け を選びます。
5. **Stata** は列の区切り文字がカンマ (**comma**) であることを認識し、データの見え方をプレビューします。
6. OK ボタンをクリックします。

Stata はうまくデータをインポートしました。

同じデータセットを章の後半でコピーと貼り付け以外の方法でインポートするので、テキストファイルとして保存します。メインの **Stata** ウィンドウに戻り、**do** ファイルエディタボタン  を押して、新しい **do** ファイルエディタウィンドウを開きます。**do** ファイルエディタにデータを貼り付けて保存ボタンをクリックします。作業中のフォルダを参照し、このファイルを「**a few cars.csv**」として保存します。もし作業中のフォルダ (**working directory**) が分からない場合は **Stata** のメインウィンドウの下にある、ステータスバーの左端を見てください。スプレッドシートからデータをコピーする場合、特殊なフォーマットで入力されていることもあります。そのままコピーすると、フォーマットが矩形にならずに崩れてしまう場合もあります。スプレッドシートに空の行や列、重複しているヘッダー、結合したセルがないことを確認してください。これらが残っていると貼り付ける時にトラブルを起こす可能性があります。スプレッドシートが表のように見えるなら基本的には問題はないはずです。

8.2 データをインポートするコマンド

コピーと貼り付けは **Stata** 内にデータをインポートする簡単な方法ですが、はっきりとした操作記録（履歴）が必要な場合、別の方法でデータをインポートしなければなりません。章の残りを使い、コピーと貼り付け以外のインポートコマンドを確認します。また、繰り返し行う作業に適したインポート方法と、様々な形式のデータをインポートする方法も紹介します。

Stata にはデータをインポートするコマンドが複数あります。一般的なテキストファイルのデータセットをインポートする時に使用するコマンドの内、特によく使用する **3** つのコマンドは次の通りです。

- **import delimited** スプレッドシートやデータベースプログラムで作成したテキストファイルを読み取ります。また、さらに全般的に、列を区切っているデリミタ、たとえばカンマ、タブ、セミicolon、スペースが定義されているテキストファイルも読み取れます。
- **infile** スペース区切りのファイルとデータが固定列で定義されているファイルを読み取ります。
- **infix** 固定列になっているファイルを読み取ります。**1** つのデータが途中で分断されても読み取れます。

Stata はこの他にもさまざまなタイプのファイルを読み込むコマンドや、外部のデータベースから直接データを取得するコマンドなどがあります。

- `import excel` **Microsoft Excel** のファイルを直接読み込みます。 `.xls` と `.xlsx` ファイルを読み込みます。
- `import sas` ネイティブ **SAS** ファイルを読み込み、データを **SAS** から **Stata** のフォーマットに変換します。
- `import spss` **IBM SPSS Statistics** ファイルを読み込みます。
- `import sasxport5` **SAS V5 Transport** ファイルを読み込みます。 `import sasxport8` コマンドは **SAS V8 Transport** ファイルを読み込みます。
- `odbc` **ODBC (Open Database Connectivity)** ドライバに対応しているデータベースからインポートできます。
- `jdbc` データベースからデータをロードし、データベースで **SQL** ステートメントを実行し、**JDBC (Java Database Connectivity)** ドライバーを使用してデータベースにデータを挿入できます。

他の形式のインポートも可能です。インポート可能な形式の一覧は [\[D\]import](#) をご覧ください。各コマンドはそれぞれのファイルが特定の形式であることを前提にしています。この章ではファイル形式についてサンプルを用いて説明します。詳細に関しては **Data Management Reference Manual** を参照してください。

8.3 import delimited コマンド

`import delimited` コマンドはスプレッドシートやデータベースのプログラムによって作成されたテキストファイルを読み取るコマンドです。この形式のファイルは、インターネット上でデータをやり取りする中で最も一般的なデータ形式です。全てのスプレッドシートプログラムと多くのデータベースアプリケーションではデータセットをテキストファイルとして保存できます。この時、行をタブまたはカンマで区切って保存します。プログラムによっては列タイトル (**Stata** では変数名) をテキストファイル内に保存できます。

このようなファイルを読み込むにはコマンドウィンドウに「`import delimited` ファイル名」を入力します。この場合ファイル名はインポートするテキストファイルの名前です。`import delimited` コマンドが自動で検出するデリミタ (区切り) はタブかカンマです。また、各列のデータの種類も自動で判別します。フ

ファイル名にスペースが使用されている場合、ダブルクォーテーション ("") 内にファイル名を入力します。目的のファイルが現在の作業フォルダ内にはそのファイルへのパスも忘れずに入力してください。

`import delimited` コマンドはデリミタがタブかカンマである場合、自動的に認識します。他の記号をデリミタとして使用しているファイルの場合、`import delimited` コマンドの `delimiters()` オプションを使用してください。

章の冒頭（「コピーと貼り付け」セクション）で「`a few cars.csv`」というファイルを保存しました。このデータには古い車の `make`, `price`, `MPG`, `weight`, `gear ratio` が入っています。変数名はファイルに含まれていないので `import delimited` コマンドは独自の名前を割り振ります。そして、カンマでフィールドが区切られています。現在開いているデータを一度 `clear` コマンドで消し、`import delimited` コマンドを使用してこのファイルのデータを読み込みましょう。このファイル名にはスペースを使用しているのでダブルクォーテーション ("") を使用します。

```
. clear
. import delimited "a few cars.csv"
(encoding automatically selected: ISO-8859-9)
(5 vars, 7 obs)
```

データエディタで見ると、章の冒頭で紹介したコピーと貼り付けのデータと全く同じものだと分かります。`list` コマンドを使用して結果ウィンドウにデータを表示します。`separator(0)` オプションはデフォルトで 5 行ごとに挿入する水平線を無効にします。

```
. list, separator(0)
```

	v1	v2	v3	v4	v5
1.	VW Rabbit	4697	25	1930	3.78
2.	Olds 98	8814	21	4060	2.41
3.	Chev. Monza	3667	.	2750	2.73
4.		4099	22	2930	3.58
5.	Datsun 510	5079	24	2280	3.54
6.	Buick Regal	5189	20	3280	2.93
7.	Datsun 810	8129	.	2750	3.55

ファイルを取り込む時に変数名を指定する場合、次のように `import delimited` コマンドを実行する時に使用する変数名も入力します。変数名を指定する場合、ファイル名の前に `using` キーワードを入力しなければなりません。

```
. import delimited make price mpg weight gear_ratio using "a few cars.csv"
(encoding automatically selected: ISO-8859-9)
(5 vars, 7 obs)
. list, separator(0)
```

	make	price	mpg	weight	gear_r~o
1.	VW Rabbit	4697	25	1930	3.78
2.	Olds 98	8814	21	4060	2.41
3.	Chev. Monza	3667	.	2750	2.73
4.		4099	22	2930	3.58
5.	Datsun 510	5079	24	2280	3.54
6.	Buick Regal	5189	20	3280	2.93
7.	Datsun 810	8129	.	2750	3.55

表示データに関して、余談ですが、**Stata**は `gear_ratio` を `gear_r~o` として表示します。`gear_r~o` は `gear_ratio` を意味する、固有の省略形です。デフォルトで変数名の文字数が **8** 文字よりも多い場合、省略形を表示します。

`gear_ratio` を省略形で表示しないようにするには `abbreviate(10)` オプションで指定できます。

```
. list, separator(0) abbreviate(10)
```

	make	price	mpg	weight	gear_ratio
1.	VW Rabbit	4697	25	1930	3.78
2.	Olds 98	8814	21	4060	2.41
3.	Chev. Monza	3667	.	2750	2.73
4.		4099	22	2930	3.58
5.	Datsun 510	5079	24	2280	3.54
6.	Buick Regal	5189	20	3280	2.93
7.	Datsun 810	8129	.	2750	3.55

「~」を使用した省略形と `list` コマンドの詳細は、[\[GSW\] 10 Listing data and basic command syntax](#) (データのリストと基本コマンドの構文) をご覧ください。

このデータセットは次の章でも使用するので保存します。コマンドウィンドウに「`save afewcars`」と入力し、**Enter** キーを押します。

`import delimited` コマンドを使用する他に、テキストファイルの内容をコピーしてデータエディタに特殊な貼り付け... で貼り付けるとインポート可能です。その時に表示するダイアログでカンマ (**comma**) をデリミタとして選択します。

デリミタでうまく区切られていないファイルや **1**つの観測が複数行にまたがるデータのために `infile` と `infix` コマンドがあります。このようなファイルのインポートの詳細については [\[D\] import](#) をご覧ください。

8.4 他のソフトウェアからファイルをインポートする

Stata には他アプリケーションで作成したデータを読み取る特殊なコマンドがあります。

`import excel` コマンドは **Microsoft Excel** で作成したファイルを読み取るものです。`import excel` コマンドの詳細は [\[D\] import excel](#) をご覧ください。

`import spss` コマンドは **IBM SPSS Statistics** ファイルの読み取りと作成を行います。詳細は [\[D\] import spss](#) をご覧ください。

`import sas` コマンドは **SAS** ファイルの読み取りと作成を行います。詳細は [\[D\] import sas](#) をご覧ください。

`import sasxport5` コマンドおよび `import sasxport8` は **SAS V5 Transport** ファイルおよびの **SAS V8 Transport** ファイル読み取りと作成を行います。詳細は [\[D\] import sasxport5](#) および [\[D\] import sasxport8](#) をご覧ください。

ODBC に対応しているソフトウェアからインポートする場合、`odbc` コマンドで中間ファイルを作成することなく読み取ります。詳細は [\[D\] odbc](#) をご覧ください。

`jdbc` コマンドは **JDBC** を使用して、データベースへの接続、データのロード、データベースへのデータの挿入、およびデータベースでのクエリの実行を行うことができます。詳細は [\[D\] jdbc](#) をご覧ください。

インポートする方法の一覧は次の通りです。

- **Microsoft Excel** の `.xls` と `.xlsx` 拡張子のファイルの場合、`import excel` コマンドを使用します。
- **IBM SPSS Statistics** の `.sav` 拡張子のファイルの場合、`import spss` コマンドを使用します。
- **Windows** マシンで作成した、**SAS** の `.sas7bdat` 拡張子のファイルの場合、`import sas` コマンドを使用します。
- スプレッドシートまたはデータベースからインポートするファイルがタブ区切りか **CSV** ファイルの場合、`import delimited` コマンドを使用します。
- フィックス形式ファイルをインポートする場合、`infile` コマンドを定義ファイルと共に使用するか、`infix` コマンドを使用します。
- **ODBC** でアクセス可能なデータベースの場合、`odbc` コマンドを使用します。
- **JDBC** でアクセス可能なデータベースの場合、`jdbc` コマンドを使用します。
- **SAS V5 Transport** ファイルの場合、`import sasxport5` コマンドを使用します。
- **SAS V8 Transport** ファイルの場合、`import sasxport8` コマンドを使用します。
- 米連邦準備銀行経済データの場合、`import fred` を使用します。
- **Haver Analytics** データベースのデータを取り込む場合、`import haver` コマンドを使用します。
- **dBASE** ファイルの場合、`import dbase` コマンドを使用します。
- 表形式のデータがある時はデータエディタへ直接コピーと貼り付けを行うことができます。
- 最後に、サードパーティ製のファイル変換プログラムを購入し、ファイル形式を **Stata** 形式に変換するという方法もあります。

9. データのラベリング

9.1 データを見やすくするには

この章ではデータセット、変数、値のラベリングについて説明します。ラベリングはデータそのままでは分かりにくい変数や値に説明を加えることができます。**0**と**1**のダミー変数を使用してデータをカテゴリー分けする時にそのままデータエディタで表形式にしてみても、この**0**と**1**がそれぞれ何を示すのか分かりません。このような場合、値ラベルを使用すると、**0**と**1**に具体的なカテゴリー名を表示できます。たとえば、`afewcars.dta`のデータセットにダミー変数**0**と**1**を表示します。これでは何を示しているのか分かりません。これらの様式は他の人とデータを共有する際に重要な役割を担うとともに、自分自身のためにもなります。結果ウィンドウに結果を表示する時にもラベルが使われるため、データセットを正しくラベリングすると結果も分かりやすくなります。では、例題を使いながらデータセット、変数、値のラベリングの作業を行いましょう。

9.2 データセットの構造 : `describe` コマンド

[GSW] 8 Importing data (データのインポート) 内の「`import delimited` コマンド」セクションの最後で「`afewcars.dta`」というデータセットを保存しました。このデータセットを整え、他の研究者が見たときに理解できるようにします。まずはデータエディタでデータを確認しましょう。

```
. use afewcars
. list, separator(0)
```

	make	price	mpg	weight	gear_r~o
1.	VW Rabbit	4697	25	1930	3.78
2.	Olds 98	8814	21	4060	2.41
3.	Chev. Monza	3667	.	2750	2.73
4.		4099	22	2930	3.58
5.	Datsun 510	5079	24	2280	3.54
6.	Buick Regal	5189	20	3280	2.93
7.	Datsun 810	8129	.	2750	3.55

この表からではデータセット内の値についてはほとんど分かりません。それだけでなく、**price**(価格)や **weight**(車重)の単位も分かりません。また、「**mpg**」の表記はアメリカ以外の人には意味が通じない可能性もあります。データセットの構造を確認するともう少し何か分かるかもしれません。データセットの構造を確認するには **describe** コマンドを使います ([\[GSW\] 1 Introducing Stata—sample session \(Stata の紹介—サンプルセッション\)](#) で一度使用しました)。

```
. describe
```

Contains data from afewcars.dta
Observations: 7
Variables: 5 21 Jul 2023 15:48

Variable name	Storage type	Display format	Value label	Variable label
make	str11	%11s		
price	int	%8.0g		
mpg	byte	%8.0g		
weight	int	%8.0g		
gear_ratio	float	%9.0g		

研究者がデータ分析に使用出来そうな情報はこの中にはほとんどありません。しかし表のはじめの **3** 列から

Stata のデータの扱い方が分かります。

- 変数名 (**variable name**) は各変数を表す固有の名前です。
- 保存タイプ (**storage type**)、またはデータ型は変数内にデータを保存する形式を示します。 **Stata** には **6** 種類の保存タイプがあり、それぞれ必要メモリ量が異なります。
 - 整数 (**integer**) :

保存タイプ **byte** は最小 **-127**、最大 **100** までの整数データを格納でき、**1** 観測につき **1** バイトのメモリを使用します。

保存タイプ **int** は最小 **-32,767**、最大 **32,740** までの整数データを格納でき、**1** 観測につき **2** バイトのメモリを使用します。

整数に関する **long** は最小 **-2,147,483,647**、最大 **2,147,483,620** までの整数データを格納でき、**1** 観測につき **4** バイトのメモリを使用します。
 - 実数 (**real number**) :

保存タイプ **float** は浮動小数で、**8.5** 桁の精度でデータを格納できます。**1** 観測につき **4** バイトのメモリを使用します。

保存タイプ **double** は浮動小数で、**16.5** 桁の精度でデータを格納できます。**1** 観測につき **8** バイトのメモリを使用します。

(c) 文字列 (テキスト) :

1 から **2,04** 文字までの文字列データを格納します。メモリは **ASCII** 文字の場合 **1** 字につき **1** バイト、**Unicode** 文字の場合 **1** 文字につき最大 **4** バイトを使用します。

str1 は **1** バイト長の文字列を記入できます。

str2 は **2** バイト長の文字列を記入できます。

str3 は **3** バイト長の文字列を記入できます。

...

str2045 は **2,045** 文字の長さの記号を記入できます。

(d) また、**Stata** は **strL** (スताल) を使うと、任意の長さの文字列を格納できます。利用できる文字列の長さは **2,000,000,000** 文字までです。また、**strL** はバイナリデータ、よくデータベースで **BLOB** (binary large objects) と呼ばれる文字列も格納できます。これらの詳細はこのページでは紹介しません。

保存タイプは計算の精度とデータセットの容量に影響します。保存タイプに関する詳細は

「**help data types**」のコマンドを実行するか、[\[D\] Data types](#) をご覧ください。

3. 表示形式 (**Display format**) は数値の表示形式をコントロールします。詳しくは [\[U\] 12.5](#)

[Formats:Controlling how data are displayed](#) をご覧ください。デフォルトでは、保存タイプを元に表示形式を選択します。

このデータセットは必要な情報をすべて含めたものにしましょう。

丁寧にラベリングしてあるデータセットを参考にするために、**Stata Press** のアーカイブにあるデータセットを開いてみましょう。これはデータをロードして現在の作業内容を中断する必要も、データセットのコピーをコンピュータ上に準備する必要もありません。(インターネット上で行えることについての詳細は、[\[GSW\] 19 Updating and extending Stata—Internet functionality \(Stata のアップデートと拡張—インターネットでの機能\)](#) をご覧ください。) 実際に行うことは **describe** コマンドで正しいファイルを参照するだけです。

「**describe using** ファイル名」とコマンドウィンドウに入力して実行してください。

```
. describe using https://www.stata-press.com/data/r18/auto
```

Contains data		1978 automobile data		
Observations:	74	13 Apr 2022 17:45		
Variables:	12			

Variable name	Storage type	Display format	Value label	Variable label
make	str18	%-18s		Make and model
price	int	%8.0gc		Price
mpg	int	%8.0g		Mileage (mpg)
rep78	int	%8.0g		Repair record 1978
headroom	float	%6.1f		Headroom (in.)
trunk	int	%8.0g		Trunk space (cu. ft.)
weight	int	%8.0gc		Weight (lbs.)
length	int	%8.0g		Length (in.)
turn	int	%8.0g		Turn circle (ft.)
displacement	int	%8.0g		Displacement (cu. in.)
gear_ratio	float	%6.2f		Gear ratio
foreign	byte	%8.0g	origin	Car origin

Sorted by: foreign

この出力結果の方がより有益な情報を含んでいます。この中でもデータセットを理解する時に必要な情報がラベルに記載してある箇所が3か所あります。

- 1行目の「1978 automobile data」はデータラベルです。データセットの情報が分かります。データセットをラベリングするには、メインメニューから データ > データユーティリティ > ラベルユーティリティ > データセットのラベル と操作するか、プロパティウィンドウのラベル欄に文言を打ち込みます。ただしメインウィンドウから操作する場合、プロパティウィンドウのロックが解除してあることを確認してください。
- 各変数には変数ラベルがついています。変数ラベルには、一般的な会話で用いられる言葉での変数の説明があります。ここには、変数の単位情報も入っています。変数をラベリングするには、変数ウィンドウで変数を選択し、プロパティウィンドウのラベル欄で編集を行います。変数のラベリングは変数マネージャや「label variable」コマンドからも実行できます。
- 変数 **foreign** には値ラベルが付いています。値ラベルは数値変数、たとえば **foreign** の数値を言葉で置き換えます。describe コマンドの出力から、数値変数 **foreign** は値ラベル **origin** と対応することが確

認できます。describe コマンドからは分かりませんが変数 **foreign** は数値 **0** か **1** を持っています。値ラベル **origin** は、**0** は“Domestic”、**1** は“Foreign”と表示します。データを **browse** する（見る）と（方法については [\[GSW\] 6 Using the Data Editor \(データエディタを使用する\)](#) をご覧ください)、変数 **foreign** は値“Domestic”と“Foreign”を表示します。変数内の値をラベル付けするには **2** つの段階を踏む必要があります。まずは値ラベルの定義を行います。この操作はデータエディタまたは変数マネージャのダイアログで行うか、データ > データユーティリティ > ラベルユーティリティ > 値ラベルの管理と選択します。または「label define」と入力しても同じ操作を行えます。ラベルを定義した後に対応する変数と関連付けます。関連付けるには データ > データユーティリティ > ラベルユーティリティ > 変数に値ラベルを割り当てと操作するか、「label values」コマンドを使ってください。

メモ：値ラベルの名前は変数名と同じでも差支えありません。この場合、値ラベルの名前が **foreign** でも問題ありません。

9.3 データセットと変数のラベリング

データセット **afewcars.dta** をロードしてラベルを作成します。この操作は簡単なのでコマンドウィンドウから行いましょう。[\[GSW\] 6 Using the Data Editor \(データエディタを使用する\)](#) の変数の名前および形式の変更では、同じ目的でデータエディタを使用しました。コマンドウィンドウを使用しても、データエディタを使用しても、結果ウィンドウに表示する結果は同じコマンドになります。今回、直接コマンドウィンドウから操作する方法をご紹介します。次のように入力しましょう。

```

. use afewcars

. describe

Contains data from afewcars.dta
Observations:      7
Variables:         5                21 Jul 2023 15:48

```

Variable name	Storage type	Display format	Value label	Variable label
make	str11	%11s		
price	int	%8.0g		
mpg	byte	%8.0g		
weight	int	%8.0g		
gear_ratio	float	%9.0g		

```

Sorted by:

. label data "A few 1978 cars"

. label variable make "Make and model"

. label variable price "Price (USD)"

. label variable mpg "Mileage (miles per gallon)"

. label variable weight "Vehicle weight (lbs.)"

. label variable gear_ratio "Gear ratio"

. describe

Contains data from afewcars.dta
Observations:      7                A few 1978 cars
Variables:         5                21 Jul 2023 15:48

```

Variable name	Storage type	Display format	Value label	Variable label
make	str11	%11s		Make and model
price	int	%8.0g		Price (USD)
mpg	byte	%8.0g		Mileage (miles per gallon)
weight	int	%8.0g		Vehicle weight (lbs.)
gear_ratio	float	%9.0g		Gear ratio

```

Sorted by:
      Note: Dataset has changed since last saved.

. save afewcars2
file afewcars2.dta saved

```

9.4 変数の値をラベリングする

これから新しいダミー変数をデータセットに加えます。ダミー変数の数値 **0** はアメリカ国内メーカーの車を、**1** は他国のメーカーの車をそれぞれ表します。データエディタを開き、今まで学んできた内容をもとに変数 **foreign** を追加しましょう。

```
. list, separator(0)
```

	make	price	mpg	weight	gear_r~o	foreign
1.	VW Rabbit	4697	25	1930	3.78	1
2.	Olds 98	8814	21	4060	2.41	0
3.	Chev. Monza	3667	.	2750	2.73	0
4.		4099	22	2930	3.58	0
5.	Datsun 510	5079	24	2280	3.54	1
6.	Buick Regal	5189	20	3280	2.93	0
7.	Datsun 810	8129	.	2750	3.55	1

もちろん、データエディタを使用して新しい変数を作成する事もできます。(データエディタの使用方法については [\[GSW\] 6 Using the Data Editor \(データエディタを使用する\)](#) をご覧ください)。“0”と“1”がそれぞれ定義しているカテゴリーであるかはこの内容の中では分かりませんが、値ラベルを使用してより明確にしましょう。車について何も知らない人が見ても何を表しているデータなのか分かりやすくするためです。よって、値ラベルを使って誰でもその意味が分かるようにします。

データエディタ内でポイント&クリックインターフェイスを使って値ラベルを作成し添付する作業は、

[\[GSW\] 6 Using the Data Editor \(データエディタを使用する\)](#) の「[データを変更する](#)」セクションで行いまし

た。ここではコマンドウィンドウから直接値ラベルを作成しましょう。

```

. label define origin 0 "Domestic" 1 "Foreign"

. label values foreign origin

. describe

Contains data from afewcars2.dta
Observations:      7                A few 1978 cars
Variables:         6                21 Jul 2023 16:52

```

Variable name	Storage type	Display format	Value label	Variable label
make	str11	%11s		Make and model
price	int	%8.0g		Price (USD)
mpg	byte	%8.0g		Mileage (miles per gallon)
weight	int	%8.0g		Vehicle weight (lbs.)
gear_ratio	float	%9.0g		Gear ratio
foreign	float	%9.0g	origin	

```

Sorted by:
      Note: Dataset has changed since last saved.

. save afewcarslab
file afewcarslab.dta saved

```

値ラベルを定義するには

```
label define ラベル名 値1 "カテゴリー名1" 値2 "カテゴリー名2"...
```

とコマンドウィンドウに入力します。

変数に作成したラベルを付けるには

```
label values 変数名 ラベル名
```

と入力します。

では、先ほど作成した変数とラベルの情報を一度保存しておきましょう。データセットに値ラベルを付けて整えたことと、次の章でもこのデータを使用することを踏まえて、今までのデータセットと混同しないように新しいファイル名で保存します。

値ラベルをポイント&クリックインターフェイスで定義する場合はメインウィンドウおよびデータエディタのプロパティウィンドウか変数マネージャを使用します。詳しくは [\[GSW\] 7 Using the Variables Manager \(変数マネージャを使用する\)](#) をご覧ください。

この他にも値ラベルには使用例や構文があります。具体例は [\[U\] 12.6.3 Value labels](#) をご覧ください。

変数およびデータにメモを付け足すこともできます。メモについては [\[GSW\] 6 Using the Data Editor \(データエディタを使用する\)](#) 内の「[変数の名前および形式の変更](#)」または [\[GSW\] 7 Using the Variables Manager \(変数マネージャを使用する\)](#) 内の「[メモを管理する](#)」セクションをご覧ください。さらに詳しく知りたい場合は「[help notes](#)」とコマンドを打ってシステムヘルプを参照したり、PDF マニュアル内の [\[D\] notes](#) の解説を参照してください。

10. データのリストと基本コマンドの構文

10.1 コマンド構文

この章では **Stata** のコマンド構文の基本を学ぶと共に、データリストの表示をコントロールする方法について学びます。

このマニュアルからも分かるように、**Stata** はメニューとダイアログで操作する方法と、コマンドウィンドウにコマンドを直接入力して操作する方法の 2 つを用意しています。最初は他のソフトウェアのようにメニューで操作していたとしても、コマンドウィンドウに慣れるとより速く、快適に操作できるようになります。コマンド文がシンプルで一貫性のある構文になっているので、コマンドウィンドウを使う方が効率的に操作できます。ここでコマンドウィンドウの使い方に慣れ、より快適に **Stata** を使えるようになりましょう。

「`help language`」コマンドではオンラインヘルプで追加情報と例題を、PDF マニュアルの [\[U\] 11 Language syntax](#) ではコマンド構文に関する詳細の確認をできます。

「`help list`」とコマンドウィンドウに打ち込んで、`list` コマンドの構文を確認しましょう。

`list [varlist] [if] [in] [, options]`

この構文の読み方は次の通りです。

- [] 内の引数はすべて任意です。`list` コマンドの場合、
 - (a) `varlist` は任意です。`varlist` とは変数名のリストです。
 - (b) `if` は任意です。`if` 条件で定めた内容に当てはまるデータにのみコマンドを実行します。`if` 条件の使用例は [\[GSW\] 6 Using the Data Editor \(データエディタを使用する\)](#) で一度紹介しています。
 - (c) `in` は任意です。`in` 条件で指定した行番号にのみコマンドを実行します。
 - (d) , と `options` (オプション) は任意です。コマンド文の他の部分とオプションはカンマで分けます。
- 特に指定がない限り、引数は 1 つのコマンド文で同時に使用できます。`list` コマンドに関しては `varlist` を `if` や `in` 条件と共に使用できます。
- 単語の一部に下線が引いてある場合、下線部は **Stata** が認識する最短の省略形です。下線部よりも長

い省略形なら、**Stata** は正しいコマンドとして認識します。

(a) **list** コマンドでは“**l**”に下線がついているので、**Stata** は **l**, **li**, **lis** を **list** コマンドの省略形として認識します。

- **[]** の外の項目はすべて必須です。**list** コマンドの場合、**list** のみが必須項目となります。

これらのルールを念頭に置いて、それぞれの引数でどのように **list** コマンドの出力が変化するか見ていきましょう。前の章の最後に使用したデータセット、**afewcarslab.dta** を使用します。

10.2 変数リストを使用してデータの一部をリストする

データの一部を選んでリストするには変数リスト (**varlist**) を使います。このセクションで紹介する変数リストの入力方法は、煩わしい入力の手間を省きながら長い変数名でも快適にご利用いただけます。

- **list** コマンドでは **varlist** の入力は任意です。つまり、変数を特に指定しない場合は全ての変数を選択するのと同じです。もう 1 つの考え方は、**list** コマンドはデフォルトで全ての変数にコマンドを実行しますが、**varlist** を使用するとその変数にのみコマンドを実行します。
- 変数のサブセットだけをリストするには「**list make mpg price**」のように入力します。
- **Stata** はキーボード入力を少なくできる省略表記を複数用意しています。

m* は **m** から始まるすべての変数を表します。

price-weight はデータセットの順番で、変数 **price** から変数 **weight** までの全ての変数を表示します。

ma?e は **ma** から始まり、最後の文字が **e** になる変数を表します。

- 「**list gear r~o**」のように、**Stata** が作成した変数固有の省略形でもコマンドを実行できます。省略形が固有の形式ではない場合、**Stata** はエラーメッセージを表示します。

```
. list
```

	make	price	mpg	weight	gear_r~o	foreign
1.	VW Rabbit	4697	25	1930	3.78	Foreign
2.	Olds 98	8814	21	4060	2.41	Domestic
3.	Chev. Monza	3667	.	2750	2.73	Domestic
4.		4099	22	2930	3.58	Domestic
5.	Datsun 510	5079	24	2280	3.54	Foreign
6.	Buick Regal	5189	20	3280	2.93	Domestic
7.	Datsun 810	8129	.	2750	3.55	Foreign

```
. l make mpg price
```

	make	mpg	price
1.	VW Rabbit	25	4697
2.	Olds 98	21	8814
3.	Chev. Monza	.	3667
4.		22	4099
5.	Datsun 510	24	5079
6.	Buick Regal	20	5189
7.	Datsun 810	.	8129

```
. list m*
```

	make	mpg
1.	VW Rabbit	25
2.	Olds 98	21
3.	Chev. Monza	.
4.		22
5.	Datsun 510	24
6.	Buick Regal	20
7.	Datsun 810	.

```
. li price-weight
```

	price	mpg	weight
1.	4697	25	1930
2.	8814	21	4060
3.	3667	.	2750
4.	4099	22	2930
5.	5079	24	2280
6.	5189	20	3280
7.	8129	.	2750

```
. list ma?e
```

	make
1.	VW Rabbit
2.	Olds 98
3.	Chev. Monza
4.	
5.	Datsun 510
6.	Buick Regal
7.	Datsun 810

```
. l gear_r~o
```

	gear_r~o
1.	3.78
2.	2.41
3.	2.73
4.	3.58
5.	3.54
6.	2.93
7.	3.55

10.3 if条件でリストする

if条件は論理表現からどのデータにコマンドを実行するか決めます。if条件が当てはまるデータにはコマンドを実行しますが、当てはまらないものには実行しません。真偽の判断を行う演算子は次の通りです。

<	よりも小さい
<=	以下
==	等しい
>	よりも大きい
>=	以上

!=	等しくない
&	and
/	or
!	not (論理的否定 ; ~も使用できる)
()	カッコは評価の順番を指定するグループ分けに使用する

論理表現では、/ (or) の前に& (and) を判断します (これは算数で足し算の前に掛け算を行うのと同じようなものです)。このルールを使用して **if** 表現を作成できますが、カッコを使用の方が確実になります。詳しくは [\[U\] 13.2 Operators](#) をご覧ください。

```
. list
```

	make	price	mpg	weight	gear_r~o	foreign
1.	VW Rabbit	4697	25	1930	3.78	Foreign
2.	Olds 98	8814	21	4060	2.41	Domestic
3.	Chev. Monza	3667	.	2750	2.73	Domestic
4.		4099	22	2930	3.58	Domestic
5.	Datsun 510	5079	24	2280	3.54	Foreign
6.	Buick Regal	5189	20	3280	2.93	Domestic
7.	Datsun 810	8129	.	2750	3.55	Foreign

```
. list if mpg > 22
```

	make	price	mpg	weight	gear_r~o	foreign
1.	VW Rabbit	4697	25	1930	3.78	Foreign
3.	Chev. Monza	3667	.	2750	2.73	Domestic
5.	Datsun 510	5079	24	2280	3.54	Foreign
7.	Datsun 810	8129	.	2750	3.55	Foreign

```
. list if (mpg > 22) & !missing(mpg)
```

	make	price	mpg	weight	gear_r~o	foreign
1.	VW Rabbit	4697	25	1930	3.78	Foreign
5.	Datsun 510	5079	24	2280	3.54	Foreign

```
. list make mpg price gear if (mpg > 22) | (price > 8000 & gear < 3.5)
```

	make	mpg	price	gear_r~o
1.	VW Rabbit	25	4697	3.78
2.	Olds 98	21	8814	2.41
3.	Chev. Monza	.	3667	2.73
5.	Datsun 510	24	5079	3.54
7.	Datsun 810	.	8129	3.55

```
. list make mpg if mpg <= 22 in 2/4
```

	make	mpg
2.	Olds 98	21
4.		22

上記リストでは **Stata** が欠損値をデータ内の最大値より大きい値として扱うことを確認できます。よって、欠損値のある変数に **if** 表現を使用する場合は、注意が必要です。詳しくは [\[GSW\]6Using the Data Editor \(データエディタを使用する\)](#) をご覧ください。

10.4 if コマンドでリストをする：よくある間違い

このセクションには **list** コマンドを使う時に良くある間違いとその修正例をまとめました。まず誤ったコマンド例を示し、その後に正しいコマンド文を載せます。正しいコマンド文を見る前に間違いが分かるか試してください。

```
. list
```

	make	price	mpg	weight	gear_r~o	foreign
1.	VW Rabbit	4697	25	1930	3.78	Foreign
2.	Olds 98	8814	21	4060	2.41	Domestic
3.	Chev. Monza	3667	.	2750	2.73	Domestic
4.		4099	22	2930	3.58	Domestic
5.	Datsun 510	5079	24	2280	3.54	Foreign
6.	Buick Regal	5189	20	3280	2.93	Domestic
7.	Datsun 810	8129	.	2750	3.55	Foreign

```
. list if mpg=21
```

```
=exp not allowed
```

```
r(101);
```

この例では等号の使い方エラーが発生しています。“等しい”は = ではなく、 == で表します。正しく入力すると次のようになります。

```
. list if mpg==21
```

	make	price	mpg	weight	gear_r~o	foreign
2.	Olds 98	8814	21	4060	2.41	Domestic

次は複数の **if** 表現を組み合わせる時によくある間違いです。

```
. list if mpg==21 if weight > 4000
```

```
invalid syntax
```

```
r(198);
```

```
. list if mpg==21 and weight > 4000
```

```
invalid 'and'
```

```
r(198);
```

2つの **if** 表現を同じコマンド内で使う時は記号 **&** を使用します。**and** や複数の **if** を使用しても **Stata** は認識しません。正しい **if** 表現は「if mpg==21 if weight>4000」ではなく「if mpg==21 & weight>4000」になります。結果を出力すると次のようになります。


```
. list if mpg==21 & weight > 4000
```

	make	price	mpg	weight	gear_r~o	foreign
2.	Olds 98	8814	21	4060	2.41	Domestic

次の例は文字列変数の扱い方についてです。

```
. list if make==Datsun 510
Datsun not found
r(111);
```

文字列は `make=="Datsun 510"` のようにダブルクォテーション ("") の中に入力してください。ダブルクォテーションを使用しないと、**Stata** は **Datsun** のみを存在しない変数だと認識します。よって、「**Datsun not found**」というエラーメッセージを表示します。修正すると次のようになります。

```
. list if make=="Datsun 510"
```

	make	price	mpg	weight	gear_r~o	foreign
5.	Datsun 510	5079	24	2280	3.54	Foreign

最後に文字列とラベルの混同についてです。

```
. list if foreign=="Domestic"
type mismatch
r(109);
```

値ラベルを使う変数は文字列変数のように見えますが、実際はダミー変数なので値として保存してあるのは数値です。変数 **foreign** には値 **0** と **1** があり、**0** に“**Domestic**”、**1** に“**Foreign**”というラベルがついています (詳しくは [\[GSW\] 9 Labeling data \(データのラベリング\)](#) をご覧ください)。値ラベルの元となっている数値を確認するには「**label list**」コマンドを使用します (詳しくは [\[D\] label](#) をご覧ください)。または「**codebook 変数名**」でその変数を確認します。エラーは **if** 表現を `foreign==0` のように変更すれば直ります。

同じ内容を入力するのに別の表記方法を使うと、直接、値ラベル名を使用できます。

```
. list if foreign==0
```

	make	price	mpg	weight	gear_r~o	foreign
2.	Olds 98	8814	21	4060	2.41	Domestic
3.	Chev. Monza	3667	.	2750	2.73	Domestic
4.		4099	22	2930	3.58	Domestic
6.	Buick Regal	5189	20	3280	2.93	Domestic

```
. list if foreign=="Domestic":origin
```

	make	price	mpg	weight	gear_r~o	foreign
2.	Olds 98	8814	21	4060	2.41	Domestic
3.	Chev. Monza	3667	.	2750	2.73	Domestic
4.		4099	22	2930	3.58	Domestic
6.	Buick Regal	5189	20	3280	2.93	Domestic

10.5 in でリストする

in 条件は **numlist** (数字リスト) を使い、リストするデータ範囲を指定します。**numlist** は1つの行番号または、リストの始めと終わりの行番号を「始め/終わり」の形式で使用します。正の数はデータセットの最初から数え始め、負の数はデータセットの最後から数え始めます。次の例題を参考にしてください。

```
. list
```

	make	price	mpg	weight	gear_r~o	foreign
1.	VW Rabbit	4697	25	1930	3.78	Foreign
2.	Olds 98	8814	21	4060	2.41	Domestic
3.	Chev. Monza	3667	.	2750	2.73	Domestic
4.		4099	22	2930	3.58	Domestic
5.	Datsun 510	5079	24	2280	3.54	Foreign
6.	Buick Regal	5189	20	3280	2.93	Domestic
7.	Datsun 810	8129	.	2750	3.55	Foreign

```
. list in 1
```

	make	price	mpg	weight	gear_r~o	foreign
1.	VW Rabbit	4697	25	1930	3.78	Foreign

```
. list in -1
```

	make	price	mpg	weight	gear_r~o	foreign
7.	Datsun 810	8129	.	2750	3.55	Foreign

```
. list in 2/4
```

	make	price	mpg	weight	gear_r~o	foreign
2.	Olds 98	8814	21	4060	2.41	Domestic
3.	Chev. Monza	3667	.	2750	2.73	Domestic
4.		4099	22	2930	3.58	Domestic

```
. list in -3/-2
```

	make	price	mpg	weight	gear_r~o	foreign
5.	Datsun 510	5079	24	2280	3.54	Foreign
6.	Buick Regal	5189	20	3280	2.93	Domestic

10.6 リストの出力をコントロールする

`list` コマンドの出力はオプションを使うことでコントロールできます。使用できるオプションとして、`sepby()` オプションは変数のカテゴリー分類に応じてデータを分けて表示します。`abbreviate()` オプションは出力時に変数名省略を開始する最小文字数を指定できます。`divider` オプションは変数名の間に垂線を引きます。

```
. sort foreign make
. list ma p g f, sepby(foreign)
```

	make	price	gear_r~o	foreign
1.		4099	3.58	Domestic
2.	Buick Regal	5189	2.93	Domestic
3.	Chev. Monza	3667	2.73	Domestic
4.	Olds 98	8814	2.41	Domestic
5.	Datsun 510	5079	3.54	Foreign
6.	Datsun 810	8129	3.55	Foreign
7.	VW Rabbit	4697	3.78	Foreign

```
. list make weight gear, abbreviate(10)
```

	make	weight	gear_ratio
1.		2930	3.58
2.	Buick Regal	3280	2.93
3.	Chev. Monza	2750	2.73
4.	Olds 98	4060	2.41
5.	Datsun 510	2280	3.54
6.	Datsun 810	2750	3.55
7.	VW Rabbit	1930	3.78

```
. list, divider
```


	make	price	mpg	weight	gear_r~o	foreign
1.		4099	22	2930	3.58	Domestic
2.	Buick Regal	5189	20	3280	2.93	Domestic
3.	Chev. Monza	3667	.	2750	2.73	Domestic
4.	Olds 98	8814	21	4060	2.41	Domestic
5.	Datsun 510	5079	24	2280	3.54	Foreign
6.	Datsun 810	8129	.	2750	3.55	Foreign
7.	VW Rabbit	4697	25	1930	3.78	Foreign

separator() オプションコマンドは水平線を指定された区切りで挿入します。特に指定がない場合、separator() オプションのデフォルト値は5です。

```
. list, separator(3)
```

	make	price	mpg	weight	gear_r~o	foreign
1.		4099	22	2930	3.58	Domestic
2.	Buick Regal	5189	20	3280	2.93	Domestic
3.	Chev. Monza	3667	.	2750	2.73	Domestic
4.	Olds 98	8814	21	4060	2.41	Domestic
5.	Datsun 510	5079	24	2280	3.54	Foreign
6.	Datsun 810	8129	.	2750	3.55	Foreign
7.	VW Rabbit	4697	25	1930	3.78	Foreign

10.7 中断

Stata のコマンドを中断する時はメインウィンドウのツールバーにある中断ボタン  をクリックします。中断ボタンをクリックした後の Stata の状態は、現在のコマンドを実行する前の状態に戻りますので安全です。

11. 新しい変数を作成する

11.1 作成と置換

この章では **Stata** で変数の作成や値の変更を行う操作について説明します。データエディタを使う方法はすでに [\[GSW\] 6 Using the Data Editor \(データエディタを使用する\)](#) で紹介しました。ここではコマンドウィンドウを使用する方法を見ていきます。この作業で使う **2**つのコマンドは次の通りです。

- **generate** 新しい変数を作成します。このコマンドの最小省略形は **g** です。
- **replace** 既存の変数の値を変更する時に使用します。このコマンドには省略形は存在しません。

既存のデータを変更するコマンドなので、誤ってデータを変えてしまうことを未然に防ぐためです。

新しい変数を作成する最も基本的なコマンド文の形式は「**generate newvar =exp**」となります。この **exp** は表現 (**expression**) を指します。もちろん、**generate** と **replace** のコマンドは、**if** と **in** 条件と共に使用できます。**Stata** で表現というと、定数、既存の変数、演算子、関数を使って表す式のことです。データセット **auto.dta** 内にある変数を使って例を記すと、**2 + price, weight^2, sqrt(gear_ratio)** となります。

Stata が定義する演算子は次の表の通りです。

算術		論理		大小関係 (数値および文字列)	
+	足し算	!	not	>	よりも大きい
-	引き算	/	or	<	よりも小さい
*	掛け算	&	and	>=	>または等しい
/	割り算			<=	<または等しい
^	べき乗			==	等しい
				!=	等しくない
+	文字列の連結				

Stata は多くの統計、文字列、日付、時系列、プログラミングに関する関数を用意しています。オンラインヘルプを参照するには「`help functions`」コマンドを実行してください。また、PDF マニュアルを直接参照するには [\[D\] functions](#) をご覧ください。

メニューとダイアログから新しい変数を作成して既存の変数を変更することもできます。操作はメニューから データ > データの作成または変更 と行います。このダイアログから使用できる関数の確認もできます。しかし、今回はコマンドウィンドウの簡単な使い方とよくある間違いをお伝えするため、コマンドウィンドウを使って操作します。

Stata にはいくつかのユーティリティコマンドがあり、それを使用して新しい変数を作成できます。

- `egen` コマンドは変数グループを横断して作業する時やデータのグループをまとめて操作する時に便利です。詳細は [\[D\] egen](#) をご覧ください。
- `encode` コマンドはカテゴリーごとの文字列をエンコードした数値に置き換えます。反対に、`decode` コマンドは数値のカテゴリーを文字列に直します。詳細は [\[D\] encode](#) をご覧ください。
- `destring` コマンドは数値であるべき文字列変数、たとえば通貨記号のついた数字等を数値に変換します。数値から文字列に変換するには `tostring` コマンドが便利です。詳細は [\[D\] destring](#) をご覧ください。

この章では、`generate` と `replace` コマンドに集中します。

11.2 generate コマンド

`generate` コマンドはよく使うので、次に述べることはぜひ覚えておいてください。

- `generate` コマンドの基本形は「`generate newvar =exp`」です。`newvar` はこのコマンドで新たに作成する変数の名前、`exp` はその変数を定義する表現です。既存の変数と同名の変数を `generate` コマンドで作成しようとすると、エラーメッセージが表示され、変数作成に失敗します。
- 欠損値を使って代数計算をすると、欠損値を算出します。これは、たとえば `0` を使う割り算や負の数の平方根のように、計算が不可能な数式を計算した場合と同じです。

- 新しい変数 (**newvar**) を作成中に欠損値ができた場合、**Stata** は変数内の欠損値の数を常に報告します。**Stata** が欠損値の数を表示しない場合、欠損値はできなかったことを示します。
- **generate** コマンドを使って新しい変数 (**newvar**) を作成しながら、保存タイプの設定ができます。たとえば、ダミー変数 (**0/1**) は保存タイプ **byte** として作成するといいいでしょう。これはデフォルトの保存タイプである **float** よりもデータ **1** つにつき **3** バイトの節約になるからです。

次の例題はデータセット **afewcarslab** からいくつかの新しい変数を作成できます。このデータセット **afewcarslab** は [\[GSW\] 9 Labeling data \(データのラベリング\)](#) の変数の値をラベリングするで作成したデータセットです。(実際に読み進めながら操作するには **automobile** データセットを開きます。このデータセットを小さくして、リストを短くしています。) 最後の例はダミー変数を作成する方法を示しています。この場合、**3000** ポンド (約 **1350kg**) より重い車を識別します。**Stata** の論理表現では、**1** は“真”、**0** は“偽”となります。この例での **if** 条件は、作成するダミー変数の分類対象である **weight** (車重) が欠損値でないことを確認するために使用しています。

```
. use afewcarslab
(A few 1978 cars)
```

```
. list make mpg weight
```

	make	mpg	weight
1.		22	2930
2.	Buick Regal	20	3280
3.	Chev. Monza	.	2750
4.	Olds 98	21	4060
5.	Datsun 510	24	2280
6.	Datsun 810	.	2750
7.	VW Rabbit	25	1930

```
. * changing MPG to liters per 100km
. generate lphk = 3.7854 * (100 / 1.6093) / mpg
(2 missing values generated)
```

```
. label var lphk "Liters per 100km"
```

```
. * getting logarithms of price
. g lnprice = ln(price)
```

```
. * making an indicator of hugeness
. gen byte huge = weight >= 3000 if !missing(weight)
```

```
. l make mpg weight lphk lnprice huge
```

	make	mpg	weight	lphk	lnprice	huge
1.		22	2930	10.69183	8.318499	0
2.	Buick Regal	20	3280	11.76101	8.554296	1
3.	Chev. Monza	.	2750	.	8.207129	0
4.	Olds 98	21	4060	11.20097	9.084097	1
5.	Datsun 510	24	2280	9.800845	8.532869	0
6.	Datsun 810	.	2750	.	9.003193	0
7.	VW Rabbit	25	1930	9.408812	8.454679	0

11.3 replace コマンド

新しい変数を作成する時は `generate` コマンドを使い、既存の変数の値を変更する時は `replace` コマンドを使用します。ある変数 **A** を作成後に再び変数 **A** を作成しようとするエラーになり、値は上書きされません。変数の値を間違えた時は `replace` コマンドを使って修正します。このように **Stata** は誤ってデータを上書きすることを防ぎますが、`replace` コマンドも省略できると他の **r** から始まるコマンドと間違える可能性があります。それを防ぐために `replace` コマンドは省略できません。**Stata** ではデータ変更ができるコマンドを使う場合は全てのスペルを入力する必要があります。

```
. list make weight
```

	make	weight
1.		2930
2.	Buick Regal	3280
3.	Chev. Monza	2750
4.	Olds 98	4060
5.	Datsun 510	2280
6.	Datsun 810	2750
7.	VW Rabbit	1930

```
. * will give an error because weight already exists
```

```
. gen weight = weight/1000
variable weight already defined
r(110);
```

```
. * will replace weight in lbs by weight in 1000s of lbs
```

```
. replace weight = weight/1000
variable weight was int now float
(7 real changes made)
```

```
.
. list make weight
```

	make	weight
1.		2.93
2.	Buick Regal	3.28
3.	Chev. Monza	2.75
4.	Olds 98	4.06
5.	Datsun 510	2.28
6.	Datsun 810	2.75
7.	VW Rabbit	1.93

たとえば、新しい変数 **predprice** を作成します。この変数は翌年の車の価格を予測するものです。国内製の車の価格は **5%**、外国製の車は **10%** の割合で値上がりすると推定した、とします。

変数を作成するに当たり、まずは先ほどの `generate` コマンドを使用して国内製の車の価格を予測します。この時点では外国製の車の予測価格は欠損値の状態になります。次に `replace` コマンドを使用して外国製の車の値を欠損値から正しい値に変更します。

```
. gen predprice = 1.05*price if foreign==0
(3 missing values generated)

. replace predprice = 1.10*price if foreign==1
(3 real changes made)

. list make foreign price predprice, nolabel
```

	make	foreign	price	predprice
1.		0	4099	4303.95
2.	Buick Regal	0	5189	5448.45
3.	Chev. Monza	0	3667	3850.35
4.	Olds 98	0	8814	9254.7
5.	Datsun 510	1	5079	5586.9
6.	Datsun 810	1	8129	8941.9
7.	VW Rabbit	1	4697	5166.7

変数 `foreign` はダミー変数なので、この変数 `predprice` は次に示すように **1**つのコマンドでも作成できます。

```
. gen predprice2 = (1.05 + 0.05*foreign)*price

. list make foreign price predprice predprice2, nolabel
```

	make	foreign	price	predprice	predprice2
1.		0	4099	4303.95	4303.95
2.	Buick Regal	0	5189	5448.45	5448.45
3.	Chev. Monza	0	3667	3850.35	3850.35
4.	Olds 98	0	8814	9254.7	9254.7
5.	Datsun 510	1	5079	5586.9	5586.9
6.	Datsun 810	1	8129	8941.9	8941.9
7.	VW Rabbit	1	4697	5166.7	5166.7

11.4 generate での文字列変数の作成

Stata は変数作成時の表現で文字列を認識すると、文字列をちょうど格納するのに必要な文字数分の保存タイプを用意します。次の列では変数 **where** は **str1** です。

```
. list make foreign
```

	make	foreign
1.		Domestic
2.	Buick Regal	Domestic
3.	Chev. Monza	Domestic
4.	Olds 98	Domestic
5.	Datsun 510	Foreign
6.	Datsun 810	Foreign
7.	VW Rabbit	Foreign

```
. gen where = "D" if foreign=="Domestic":origin
(3 missing values generated)
```

```
. replace where = "F" if foreign=="Foreign":origin
(3 real changes made)
```

```
. list make foreign where
```

	make	foreign	where
1.		Domestic	D
2.	Buick Regal	Domestic	D
3.	Chev. Monza	Domestic	D
4.	Olds 98	Domestic	D
5.	Datsun 510	Foreign	F
6.	Datsun 810	Foreign	F
7.	VW Rabbit	Foreign	F

```
. describe where
```

Variable name	Storage type	Display format	Value label	Variable label
where	str1	%9s		

Stata は文字列変数で作業するのに便利なツールを備えています。これから変数 **make** を **make** (メーカー) と **model** (モデル) に分けます。そして、分けた **model** と製造場所 (変数 **where**) を合わせた変数を作成します。

```
. gen model = substr(make, substrpos(make, " ")+1,.)
(1 missing value generated)

. gen modelwhere = model + " " + where

. list make where model modelwhere
```

	make	where	model	modelw~e
1.		D		D
2.	Buick Regal	D	Regal	Regal D
3.	Chev. Monza	D	Monza	Monza D
4.	Olds 98	D	98	98 D
5.	Datsun 510	F	510	510 F
6.	Datsun 810	F	810	810 F
7.	VW Rabbit	F	Rabbit	Rabbit F

上記コマンドの説明は次の通りです。

1. `substrpos(s1,s2)` は文字列 **s1** のなかで最初に文字列 **s2** が見つかる位置を整数で返します。見つからなかったときは **0** を返します。上記例の `substrpos(make, " ")` は変数 **make** の中でスペースが先頭から何番目に位置する、という情報を出力します。
2. `substr(s,start,len)` は、文字列 **s** の **start** 番目から **len** 文字分の文字列を取り出します。**len = .** の場合、**start** 番目から最後まで文字列になります。
3. 上記の **1** と **2** を同時に使うこともできます。`substr(s,substrpos(s, " ")+1,.)` は、文字列 **s** の最初の **1** 単語を除いた文字列を与えます。よって、上記例は「変数 **make** の全データから最初のスペースの次の文字から最後までを取り出す」という意味になり、たとえば **1** 行目では“**Rabbit**”を抽出します。
4. 文字列に演算子“+”を使うと、その文字列 (変数) をつなげて **1** つにします。つまり、表現 `"this"+"that"` は `"thisthat"` という文字列になります。上記例では変数 **modelwhere** 作成した時、`model + " " + where` と入力したので変数 **model** と **where** の間にスペース (" ") が入りました。

5. 文字列変数では欠損値の扱いが他の変数とは異なり、空白の文字列 (" ") という認識になります。
変数 `modelwhere` でメーカーやモデルが分からない車 (4行目) では"**D**"と表示します。(Dの前にスペースがあります。)
6. **Unicode** 文字には専用の特別なコマンドが利用できます。詳細は [\[U\] 12.4.2 Handling Uni- code strings](#) をご覧ください。

12. 変数やデータを削除する

12.1 clear, drop, keep コマンド

この章ではデータと変数を削除するツールを紹介します。データエディタでの操作方法は [\[GSW\] 6 Using the Data Editor \(データエディタを使用する\)](#) で紹介しました。ここではコマンドウィンドウを使用する方法を見ていきましょう。

データや他のオブジェクト、たとえば値ラベルをメモリから消すために **Stata** には **clear**、**drop**、**keep** の 3 つのコマンドがあります。このコマンドはメモリ内にだけ影響を与えます。つまり、ディスクに保存してあるものを変更することはありません。

12.2 clear と drop_all コマンド

仮に、何かの分析やシミュレーションを行う中でデータセットをメモリから消去し、別のデータセットをロードして使うとしましょう。その際に今まで使用していたデータセットの内容を保存する必要は無く、ただ何も入っていないデータセットを準備します。上記 3 つのコマンドの内どれを使用するべきか決めるには、何を行いたいのか把握する必要があります。これには普通のデータの他にも作成されるメタデータ（たとえば値ラベル、保存した推定コマンド、保存した行列など）の扱いも含まれます。**clear** コマンドには多様なオプションがありますが、この章では簡単な使用方法だけを確認します。詳しくは **help clear** コマンドを使用するか、[\[D\] clear](#) をご覧ください。

clear コマンドを実行すると、開いているデータセットの全ての変数と値ラベルを削除します。これが通常の用法です。ディスクに保存した結果は削除しないので、新しいデータセットをロードしてから保存した推定結果などを再度分析に利用できます。詳しくは **help postest** コマンドを実行するか、[\[U\] 20 Estimation and postestimation commands](#) をご覧ください。

メモリ上のデータ等、全てを確実に削除したい場合は **clear all** コマンドを使用してください。このコマンドは

Stata のメモリ上のデータおよびメタデータを削除するので、何もない状態から作業を開始できます。最初、グラフウィンドウなどが開いている状態でコマンドを使用すると、そのウィンドウは自動的に閉じるので、驚くかもしれません。この動作で使用中のメモリを空にします。データセットだけを削除して他の物（たとえば値ラベルなど）をそのまま残しておきたい場合、**drop all** コマンドを使用してください。

12.3 drop コマンド

drop コマンドは変数やデータをデータセットのメモリから削除します。

- 変数を取り除く場合、**drop varlist** を使用します。
- データを取り除く場合、**drop** コマンドを **if** か **in** または両方の条件と共に使用します。

ではデータセット **afewcarslab** を使用して **drop** コマンドを確認しましょう。

```
. use afewcarslab
(A few 1978 cars)
```

```
. list
```

	make	price	mpg	weight	gear_r~o	foreign
	VW Rabbit	4697	25	1930	3.78	Foreign
1.	Olds 98	8814	21	4060	2.41	Domestic
2.	Chev. Monza	3667	.	2750	2.73	Domestic
3.	4099	22	2930	3.58	Domestic	
4.	Datsun 510	5079	24	2280	3.54	Foreign
5.	Buick Regal	5189	20	3280	2.93	Domestic
6.	Datsun 810	8129	.	2750	3.55	Foreign
7.						

```
. drop in 1/3
(3 observations deleted)
```

```
. list
```

	make	price	mpg	weight	gear_r~o	foreign
		4099	22	2930	3.58	Domestic
1.	Datsun 510	5079	24	2280	3.54	Foreign
2.	Buick Regal	5189	20	3280	2.93	Domestic
3.	Datsun 810	8129	.	2750	3.55	Foreign
4.						

```
. drop if mpg > 21
(3 observations deleted)
```

```
. list
```

	make	price	mpg	weight	gear_r~o	foreign
1.	Buick Regal	5189	20	3280	2.93	Domestic

```
. drop gear_ratio
```

```
. list
```

	make	price	mpg	weight	foreign
1.	Buick Regal	5189	20	3280	Domestic

```
. drop m*
. list
```

	price	weight	foreign
1.	5189	3280	Domestic

メモリ内のデータにのみこの操作は適用されます。データセットを保存すると変更内容を確定できます。

12.4 keep コマンド

`keep` コマンドは選択した変数または `if` や `in` 表現で指定したデータを除く、全ての変数を削除します。

`keep` コマンドは `drop` コマンドのように `varlist` または `if` や `in` 表現と共に使用できます。しかし、1つのコマンドで使用できるのは `varlist` または `if` や `in` 表現のどちらかだけです。この例の中ではまず `clear` コマンドを使用して、再びデータセット `afewcarslab` をロードしてから作業を始めています。

```
. list
```

	make	price	mpg	weight	gear_r~o	foreign
1.	VW Rabbit	4697	25	1930	3.78	Foreign
2.	Olds 98	8814	21	4060	2.41	Domestic
2.	Chev. Monza	3667	.	2750	2.73	Domestic
3.		4099	22	2930	3.58	Domestic
3.	Datsun 510	5079	24	2280	3.54	Foreign
4.						
5.	Buick Regal	5189	20	3280	2.93	Domestic
6.	Datsun 810	8129	.	2750	3.55	Foreign
7.						

```
. keep in 4/7
```

```
(3 observations deleted)
```

```
. list
```

	make	price	mpg	weight	gear_r~o	foreign
1.		4099	22	2930	3.58	Domestic
2.	Datsun 510	5079	24	2280	3.54	Foreign
3.	Buick Regal	5189	20	3280	2.93	Domestic
4.	Datsun 810	8129	.	2750	3.55	Foreign

```
. keep if mpg <= 21
(3 observations deleted)
```

```
. list
```

	make	price	mpg	weight	gear_r~o	foreign
1.	Buick Regal	5189	20	3280	2.93	Domestic

```
. keep m*
```

```
. list
```

	make	mpg
1.	Buick Regal	20

13. do ファイルエディタを使用する—Stata の自動化


13.1 do ファイルエディタ

Stata には do ファイルエディタと呼ばれる統合型テキストエディタがあり、様々なタスクで利用できます。do ファ

イルエディタという名前は do ファイルという用語から取っています。他のアプリケーションではバッチファイル (batch file) やスクリプト (script) とも呼ばれるものと同じで、do ファイルは Stata の実行コマンドで構成されます。詳しくは

[U] 16 Do-files をご覧ください。do ファイルエディタにはプログラム上の高度な機能が用意されていますが、それだけではなく複数のコマンドから長いプログラムを作成するための機能を用意しています。この機能は、複数の変数に対し同じ手順の操作を繰り返し行うループ設定や、複雑で繰り返しの伴うタスクをインタラクティブに行うのに適しています。

この章の内容を最大限活用していただくには、ご自分のコンピュータで操作しながら読み進めてください。まずは










do ファイルエディタを開くところから始めましょう。do ファイルエディタは do ファイルエディタボタン  をクリックするか、コマンドウィンドウで「doedit」を実行します。







13.2 do ファイルエディタツールバー

do ファイルエディタには 15 個のボタンがあります。その多くは Stata のメインツールバーのボタンと似ており、ほぼ同じ機能を備えています。



ボタンの機能を忘れてしまった時はマウスカースルをボタンの上に移動すると、ツールチップを表示します。

	新規	新規の do ファイルを新しいタブで開きます。
	開く	ディスク内の do ファイルを新しいタブで開きます。
	保存	現在の do ファイルをディスクに保存します。
	印刷	do ファイルエディタの内容を印刷します。
	検索	テキストを検索するための検索ダイアログを開きます。
	切り取り	選択したテキストを切り取り、クリップボードに収納します。
	コピー	選択したテキストをクリップボードにコピーします。
	貼り付け	クリップボードにあるテキストを現在のドキュメントに貼り付けます。
	元に戻す	最後の変更を元に戻します。

	やり直す	最後の「元に戻す (Undo)」を取り消します。
	ブックマークの追加/削除	現在の行にブックマークの挿入と削除を行います。ブックマークは do ファイル内で素早く移動するのに使用でき、長い do ファイルを使う時やデバッグ中に便利です。
	前のブックマーク	前のブックマークに移動します (使用時のみ)。
	次のブックマーク	次のブックマークに移動します (使用時のみ)。
	ファイルをビューワで表示	do ファイルの内容をビューワウィンドウに表示します。これは SMCL タグがあるファイル、例えば log ファイルやヘルプファイルを編集する時に活用できます。
	実行(Do)	do ファイルのコマンドをすべて実行し、その途中経過を含む全ての出力とコマンドを結果ウィンドウに表示します。テキストの一部を選択する時はボタン名が選択範囲を実行 (Do) に変わり、その選択部分のみを実行してコマンドの途中経過も全て表示します。以後このボタンを Do ボタンと呼びます。

13.3 do ファイルエディタを使用する

では、これから [\[GSW\] 1 Introducing Stata—sample session \(Stata の紹介—サンプルセッション\)](#) で行ったように、1978 年製の自動車の燃費について分析しましょう。分析中に大量のコマンドを使用しますが、後でコマンドを再入力せずに同じ分析を繰り返すことを考慮して do ファイルを作成することにします。

do ファイルはコマンドを順番に記入していだけで作成できます。do ファイルでは、各コマンドを上から順に実行します。詳しくは [\[U\] 16 Do-files](#) をご覧ください。

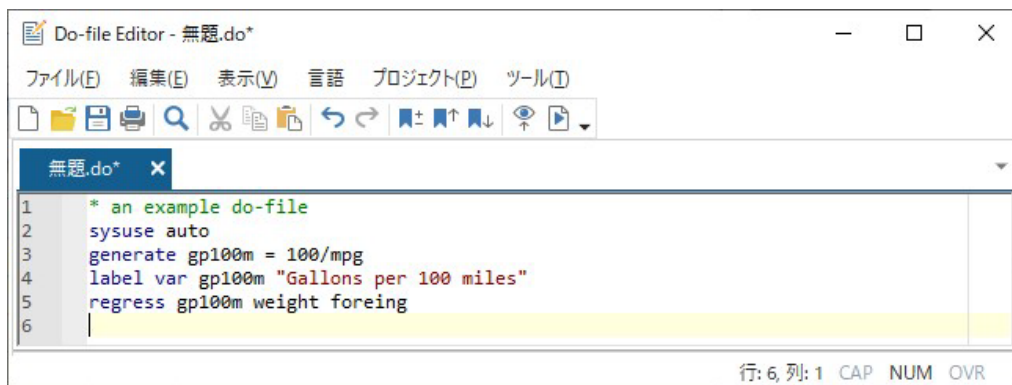
1978 年製の自動車の燃費を分析するため、100 マイル (約 160km) あたりに使用するガソリンの量を表す、新しい変数を作成します。この変数と車重の関係について、国内製と外国製のグループ間の差を見えます。まずは新しい変数で回帰を行います。

操作を始めるには、まず do ファイルエディタボタンを押して編集画面を開きます。do ファイルエディタが開いたら、以下のコマンドを画面に打ち込んでください。5 行目にある変数 **foreign** は意図的にスペルを間違えています。

(ここではよくある間違いをわざと起こしてその解決策を示します。後程、実際に使い始める時に参考にしてください。)

```
* an example do-file sysuse auto
generate gp100m = 100/mpg
label var gp100m "Gallons per 100 miles"
regress gp100m weight foreing
```

データ入力後の **do** ファイルエディタの画面は次のようになります。




do ファイルエディタに文字を入力していくと、必要に応じて文字の色が変わります。**do** ファイルエディタには入力した情報によって表示色を変更する構文ハイライト機能があります。構文要素の色やテキストのプロパティを変更するには **編集 > ユーザ設定...** と操作し、表示されるウィンドウで色タブをクリックします。また、キーワードリストを作成することでご自身の構文ハイライトを定義することも可能です。

構文のハイライト機能は **Stata** のコマンドだけが対象ではありません。**do** ファイルエディタの言語メニューで言語を切り替えることができます。**Stata** では **Markdown** を利用して動的なドキュメントを作成できますので、

Markdown のハイライトも可能となっています。詳細は [\[RPT\] dyndoc](#) をご覧ください。また、**Python** および **Java** のハイライトも可能です。詳細は [\[P\] PyStata integration](#) および [\[P\] Java integration](#) をご覧ください。**Stata** は編集しているファイルの拡張子を元にデフォルトの設定を行います。ファイルに拡張子が設定されていない場合は手動で言語の設定を行う必要があります。

また、**do** ファイルエディタには一度入力した言葉をオートコンプリートする機能があります。候補リストが表示されたら、さらに入力を進めて、候補を狭めることができます。候補リストでは上・下矢印キーで選択するか、入力続けて **1** つの候補に絞り込みます。候補が **1** つになったら、リターンキーで **do** ファイルに入力します。

それでは、**Do** ボタン  をクリックしてコマンドを実行します。**Stata** はコマンドを上から順に実行し、結果を結果ウィンドウに表示します。

```
. do "C:\Users\watabe\AppData\Local\Temp\STD124ec_000000.tmp"
. * an example do-
file . sysuse auto
(1978 automobile data)
. generate gp100m = 100/mpg
. label var gp100m "Gallons per 100 miles"
. regress gp100m weight foreing
variable foreing not found
r(111);
.
end of do-file
```

`do "C:¥..."` コマンドは **Stata** が **do** ファイルエディタからどのようにコマンドを実行したのかを示しています。**Stata** は、保存されていない変更を含むコマンドを **do** ファイルから一時ファイルに保存し、**do** コマンドを発行してそれらを実行します。上手くいったかのように見えたが、**Stata** はエラーを返してきました。初めの **3** 行分のコマンドは問題なく実行できましたが、**4** 行目のわざとスペルミスをしたコマンドでエラーが発生します。**Stata** は **foreing** という誤った名前の変数を見つける事ができません。よって、**do** ファイルエディタに戻ってスペルミスをした最後の行の変数名を **foreign** と正しく入力しましょう。

```

Do-file Editor - 無題.do*
ファイル(F) 編集(E) 表示(V) 言語 プロジェクト(P) ツール(T)
無題.do* x
1  * an example do-file
2  sysuse auto
3  generate gp100m = 100/mpg
4  label var gp100m "Gallons per 100 miles"
5  regress gp100m weight foreign
6
行: 5, 列: 30 CAP NUM OVR

```

再び **Do** ボタンをクリックします。すると、今度は **do** ファイルの **1** 行目で中断しました。既存のデータセットを閉じな

ければ **Stata** で別のファイルを開くことは出来ないからです。

```

. do "C:\Users\watabe\AppData\Local\Temp\STD124ec_000000.tmp"
. * an example do-file
. sysuse auto
no; dataset in memory has changed since last saved
r(4);
.
end of do-file

```

この状態では次のどちらかの操作を行ってください。

- **do** ファイルの一番初めに **clear** コマンドを追加します。これで **Stata** は自動的にメモリをクリアしてから **auto.dta** をロードします。これは便利ですが、**Stata** は予告なしにデータを消去するので注意が必要です。
- **clear** コマンドを手動で入力した後に **do** ファイルを改めて実行します。複雑な **do** ファイルを作成する時には、この手法は少し面倒かもしれません。

簡単なアドバイスをすると、**do** ファイルの作成中（デバック中）は **1** 番目の方法（自動的にメモリを削除）を選択し、**do** ファイルが完成した段階では、次に述べるようにその内容に応じてどちらの手法を使うかを決めてください。かなりの部分を自動化した場合は、**do** ファイルが自動的にメモリを削除する方が便利でしょう。**do** ファイルの使用頻度が低い場合は、自動的にメモリを消

さない方が安心です。このアドバイスを参考に `clear` の使用について検討してください。今回は `clear` オプションを `sysuse` コマンドと組み合わせて `do` ファイル実行前に自動的にメモリをクリアします。

```

Do-file Editor - 無題.do*
ファイル(F) 編集(E) 表示(V) 言語 プロジェクト(P) ツール(T)
無題.do* x
1  * an example do-file
2  sysuse auto, clear
3  generate gp100m = 100/mpg
4  label var gp100m "Gallons per 100 miles"
5  regress gp100m weight foreign
6
行: 6, 列: 1  CAP  NUM  OVR

```

Do ボタンをクリックして `do` ファイルを実行します。

```

. * an example do-file
. sysuse auto, clear
(1978 automobile data)

. generate gp100m = 100/mpg

. label var gp100m "Gallons per 100 miles"

. regress gp100m weight foreign

```

Source	SS	df	MS	Number of obs	=	74
Model	91.1761694	2	45.5880847	F(2, 71)	=	113.97
Residual	28.4000913	71	.400001287	Prob > F	=	0.0000
				R-squared	=	0.7625
				Adj R-squared	=	0.7558
Total	119.576261	73	1.63803097	Root MSE	=	.63246

gp100m	Coefficient	Std. err.	t	P> t	[95% conf. interval]
weight	.0016254	.0001183	13.74	0.000	.0013896 .0018612
foreign	.6220535	.1997381	3.11	0.003	.2237871 1.02032
_cons	-.0734839	.4019932	-0.18	0.855	-.8750354 .7280677

```

.
end of do-file

```

結果が出力できたところで、**do** ファイルエディタでファイル>名前を付けて保存... と選び、**do** ファイルを保存します。分析を進めながら **do** ファイルにコマンドを追加するには、メニューから ファイル>開く... と選択します。分析を行いながら **do** ファイルにコマンドを書き加えていけば、毎回新しい **Stata** のセッションでコマンドを入力し直す手間を省けます。一番初めの **clear** オプションの削除については **do** ファイルの使い方に応じてよく考えてから決めてください。

do ファイルを保存後に、改めて **do** ファイルを実行するにはコマンドウィンドウに「**do** ファイル名」と入力します。この場合、ファイル名は保存した **do** ファイルです。

13.4 ファイルメニュー

do ファイルエディタのファイルメニューには標準的なテキスト編集機能があります。メニュー内には新規ファイルを作成する「新規>ファイル」、既存のファイルを開く「開く...」、ファイルを上書き保存する「保存」、名前を付けてファイルを保存する「名前を付けて保存...」、ファイルを印刷する「印刷...」があります。また、**do** ファイルエディタのツールバーにはそれぞれの機能に対応するボタンがあります。「新規 > ファイル」を選択した後、**do** ファイルエディタウィンドウ内に新たなタブを生成し、空のドキュメントが作成されます。

さらに、新規>プロジェクトによりプロジェクトを作成し、一連のファイルをまとめて管理することもできます。プロジェクトには **do** ファイル、データファイル、グラフファイルなど、必要なファイルを含めることが可能です。プロジェクトマネージャに関する詳細は [\[P\] Project Manager](#) をご覧ください。

13.5 編集メニュー

do ファイルエディタの編集メニューには、標準的な元に戻す、やり直す、切り取り、コピー、貼り付け、削除、検索のコマンドがあります。ツールバーにはそれらのボタンがあり、ワンクリックで使用できます。他にも、編集メニューには次のような機能があります。

- 他の **do** ファイルのコマンドをカーソル位置に挿入するにはメニューから ファイル>ファイルの挿入... と選択します。

- 行を選択 現在の行を選択します。
- 行の削除 現在の行を削除します。
- 検索 > 行に移動... 指定の行番号に移動します。行番号は **do** ファイルエディタウィンドウの左端および右下にあります。
- 上級設定 プログラマー向けのサブメニューを表示します。
 - 右にシフト 1 タブ分のインデントを用意します。
 - 左にシフト 1 タブ分のインデントを解除します。
 - インデント解除 ネスト構造の親に相当する位置にインデントを調整します。
 - コメント/非コメントの切り替え 選択した行頭に//を追加/削除してコメント/非コメント化します。
 - ブロックコメントを追加 選択したブロックの前後に/* */を追加してコメント化します。
 - ブロックコメントを削除 選択したブロックの前後の/* */を削除して非コメント化します。
 - 選択部分を大文字に 選択した文字を全て大文字にします。
 - 選択部分を小文字に 選択した文字を全て小文字にします。
 - 完成語 **do** ファイルエディタに入力されている語をもとに現在の語をオートコンプリートします。候補が複数ある場合は、それら全てを表示します。候補が表示されたら、1つを選択するか、入力続けて候補を狭めていきます。
 - エンコード形式を UTF-8 に変換... 現在のファイルを **UTF-8** エンコード形式に変換します。
 - 改行コードを Mac OSX/Unix 形式に変換(\n) 現在のファイルの改行コードを **Mac OS X/Unix** 形式に変換します。
 - 改行コードを Windows 形式に変換(\r\n) 現在のファイルの改行コードを **Windows** 形式に変換します。
 - タブをスペースに変換する 既存の間隔を維持したまま、タブ文字をスペースに変換します。
 - 先頭のスペースをタブに変換する 先頭のスペースをタブ文字に変換します。タブあたりのスペースの数は環境設定に従います。
 - すべてのスペースをタブに変換する スペースをタブに変換します。タブあたりのスペースの数は環境設定に従います。
 - Unicode のスペースとカーリーダブルクオートを ASCII に変換する この変換は、Unicode で表されるスペースやカーリーダブルクオートを ASCII エンコーディングに変換します。

小括弧 (), 中括弧 { }, 大括弧 [] のマッチとバランス確認も編集メニューから選択できます。メニューから編集 > 検索 > 中カッコのマッチを選択すると、**do** ファイルエディタはカーソルの左右にある記号を確認します。どちらかの記号がエディタにマッチする記号であった場合、その括弧とマッチする (組になる) 括弧の直前にカーソルを移動します。マッチしない場合、カーソルは移動しません。

メニューから 編集 > 検索 > 中括弧のバランス と選ぶと、**do** ファイルエディタはまずカーソルおよび選択箇所の左右を確認します。そして最寄りの { } を含む部分をハイライトします。もう一度中括弧のバランスを選ぶと、次の { } まで選択範囲を拡大します。マッチする記号が無い場合、カーソルは移動しません。{ } を確認する機能は、プログラミングを行う際に **if** 条件を使ったり、ループするような表現やコードのまとまりで作業する際に便利です。詳しくは [\[P\] foreach](#) , [\[P\] forvalues](#) , [\[P\] while](#) , [\[P\] if](#) をご覧ください。

中括弧のバランスについては次の例題で確認してください。**do** ファイルエディタに「**{now {is the} time}**」と入力し、**is** と **the** の間にカーソルを置きます。メニューから 編集 > 検索 > 中括弧のバランス と操作します。

do ファイルエディタは **{is the}** を選択します。もう一度中括弧のバランスを選択すると、今度は **{now {is the} time}** を選択します。

Stata では文字列に **Unicode** 字を含めることができます。**Unicode** 文字のエンコード形式には **UTF-8** が用いられます。(詳しくは [\[U\] 12.4.2 Handling Unicode strings](#) をご覧ください。) ただし、**Stata 13** およびそれ以前のバージョンで作成した **do** ファイル、**ado** ファイルを含むテキストファイルには、アクセント付き文字、中国語、日本語、韓国語、キリル文字等を含む非 **ASCII** 文字が **UTF-8** でエンコードされていないことがあります。そうしたファイルでも、**Stata 15** 以降の **do** ファイルエディタで開くと、エンコード形式を選択する画面になり、選択した形式から **UTF-8** へとエンコード形式を変換する処理が行えます。途中で変換をキャンセルしたり、誤ったエンコード形式を指定するなどしたときは、エンコード形式を **UTF-8** に変換... と選択すると再度変換ができます。変換は編集 > 元に戻す を選択してやり直すことができ、保存しなければ元のファイルが上書きされることはありません。**Stata** データセットを変換したり、複数の **Stata** ファイルを一括で変換する場合には、**unicode translate** コマンドをご利用下さい。

編集ヒント：行番号の列をクリックすると、クリックした行すべてと行末記号を選択できます。この機能は行の削除・

切り取り・貼り付けを行う時に便利です。行番号の列をクリック&ドラッグすると行の範囲を選択できます。

13.6 表示メニュー

do ファイルエディタの表示メニューはディスプレイの表示からズームインやアウトをしたり、編集記号、たとえばタブや行末記号を表示します。

13.7 ツールメニュー

Do ボタンについては既に紹介した通りです。メニューで ツール> 実行 (**do**) と選択するのは実行(**do**) ボタンをクリックするのと同じです。

メニューで ツール> 実行 (**do**) を選択すると、最初の行から現在の行までのすべてのコマンドがコマンドウィンドウに送信されます。このメソッドは、**do** ファイルの一部を実行する簡単な方法です。

また、ツール> すべて実行 (**do**) と選択するとカーソルの行から **do** ファイルエディタ内のコマンドを最後まで実行します。これは **do** ファイルの一部を素早く実行するのに適しています。


ツール> 行を実行 (do) を選択すると、現在選択されている行のコマンドをコマンドウィンドウに送ります。カーソルは空行やコメント行を飛ばして自動的に次の実行行に移ります。行ごとに実行したい場合に便利な機能です。

ツール> サイレント実行 (**run**) を選択すると、ツール> 実行 (**do**) と同様の動作を子ないませんが、**quietly** に実行され、コマンドウィンドウには何も表示されません。

ツール> インクルード実行 (**include**) を選択すると、ローカルマクロ変数を展開して実行 (**do**) と同様の動作を行うことができます。

実行 (**do**) は Stata の **do** コマンドと同様の動作をします。インクルード実行 (**include**) は **include** コマンドと同様の動作をします。詳しくは [\[U\] 16 Do-files](#) をご覧ください。

また、**do** ファイルエディタメニューから ツール> ファイルをビューワで表示 と選択するか、ファイルをビュー

ワで表示ボタン  をクリックすると、ビューワでファイルのプレビューができます。この機能は **Stata** の **SMCL** タグを使用するファイル、たとえばヘルプファイルや **log** ファイルの作成や編集をする時に便利です。

13.8 インタラクティブコマンドを do ファイルとして保存する

Stata でインタラクティブに作業を行っているとき、少し前のコマンドを再実行したいと思うこともあるでしょう。履歴ウィンドウにある一部のコマンドまたは全てのコマンドを選択し、**do** ファイルエディタに送ることができます。別の方法としては、再実行したいコマンドを **do** ファイルとして保存し、**do** ファイルエディタで開くこともできます。また、結果ウィ

ンドウからコマンドをコピーして **do** ファイルエディタに直接貼り付ける事もできます。詳しくは [\[GSW\] 6 Using](#)

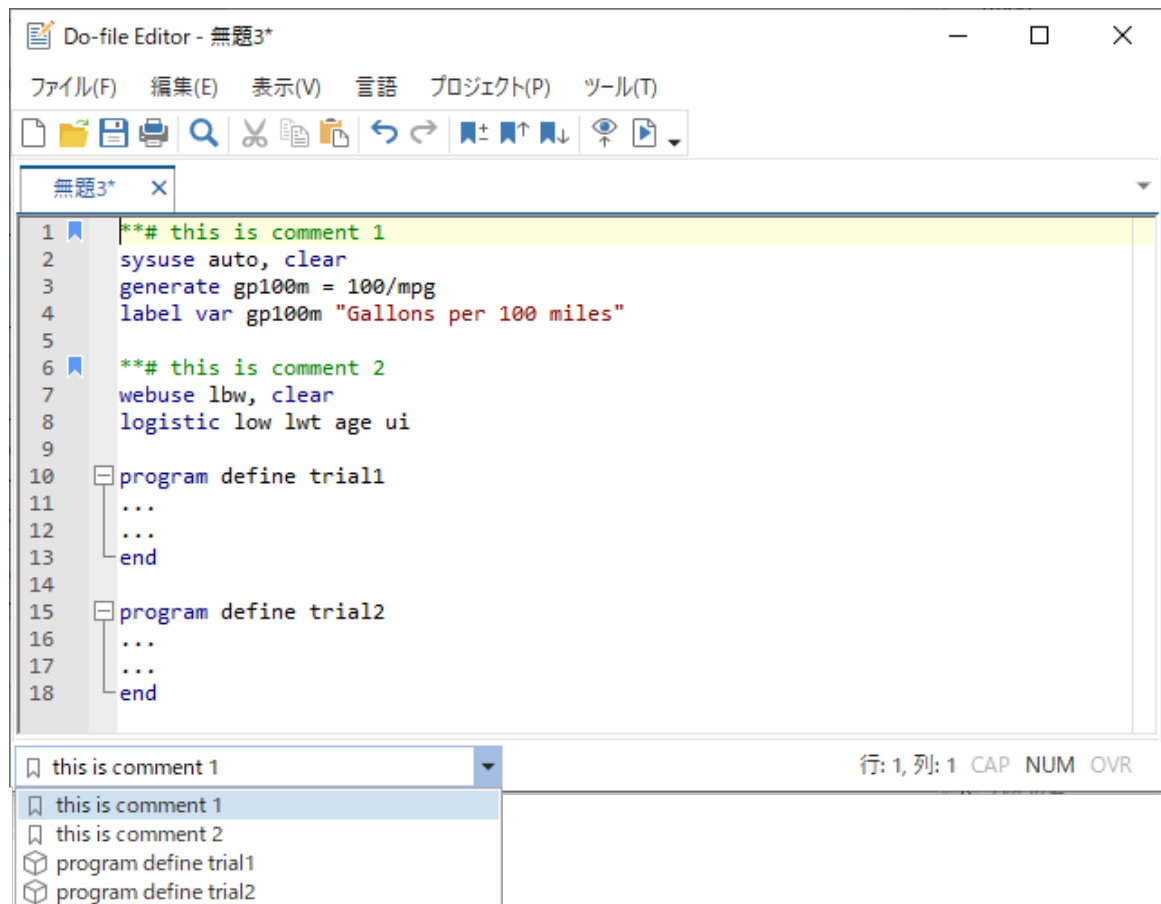
[the Data Editor \(データエディタを使用する\)](#) をご覧ください。更に、[\[R\] log](#) には **cmdlog** (コマンドログ) についての記載があります。**cmdlog** コマンドを使うと **Stata** に入力する全てのコマンドを **do** ファイルに記録できます。

13.9 do ファイルのナビゲート

長い **do** ファイルを操作する場合、ブックマークを使用すると、**do** ファイルを簡単にナビゲートできます。**do** ファイルの重要なセクションの前にブックマークを配置することで、後でそれらのセクションに簡単に戻ることができます。メニューから **編集 > ブックマークの追加/削除** を使用するか、ツールバーのブックマークの追加/削除ボタンをクリックするか、特別なコメント ****#** で始まる行を手動で入力して、ブックマーク行を追加できます。行の残りの部分にある他のすべてのテキストは、ブックマークのタイトルとして扱われます。ブックマークコメントと同じ行に **ado** コードを含めることはできません。そうしないと、ブックマークコメントは無視されます。特別なコメント **//#** を使用してブックマークを追加することもできます。このブックマークコメントは、****#** よりも優れている場合があります。これは **Mata** と **Java** の両方で有効だからです。

編集メニューのオプション、**do** ファイルエディタツールバーのボタン、またはナビゲーションコントロールを使用して、ブックマーク間を移動できます。**do** ファイルエディタのナビゲーションコントロールを使用

すると、ブックマーク間や、**do** ファイルで定義したプログラム間を移動できます。ナビゲーションコントロールからプログラムまたはブックマークを選択すると、**do** ファイル内のそのプログラムまたはブックマークの位置に直接ジャンプします。



ナビゲーションコントロールのブックマークのラベルのインデントレベルを増やすには、ブックマークコメントに#を追加します。例えば、ブックマークコメント **###** ブックマーク 2 は、ブックマークコメント **##** ブックマーク 1 よりも一つのレベルだけ多くインデントされます。

ブックマークは、編集メニューのオプションまたはツールバーのブックマークの追加/削除ボタンを使用して削除することも、行を削除して削除することもできます。行番号の横にある余白をクリックして、ブックマークを追加および削除することもできます。ブックマーク行を追加すると、ブックマークの余白にブックマークアイコンが追加され、スクロール中にわかりやすくなります。

13.10 プロジェクト

上級 Stata ユーザで多くのファイルをプロジェクトの一部として管理している場合、Stata には **do** ファイルエディタで使用できるプロジェクトマネージャがあります。プロジェクトマネージャに関する詳細は [\[P\] Project Manager](#) をご覧ください。

13.11 自動バックアップ

do ファイルエディタは、文書を開くか新しい文書を作成する際にバックアップファイルを作成するようになりました。既存の文書が開かれる際、Stata はその文書のファイル名の前に「~」を付け、拡張子を .stswp としたバックアップファイルと同じディレクトリに作成し、ディスクに保存します。新しい未保存の文書を編集する際は、バックアップファイルを一時ディレクトリに保存します。文書が閉じられると、バックアップファイルも削除されます。ただし、Stata が電源の問題やコンピュータのクラッシュによって正常に終了しない場合、バックアップファイルが残ることがあります。デフォルトでは、Stata は編集が行われた場合、または 200 文字以上の追加または削除が行われた後、文書を約 4 秒ごとにバックアップします。この時間間隔は do ファイルエディタの詳細設定で変更可能であり、バックアップ機能を無効にすることもできます。

do ファイルエディタで文書を開こうとすると、まずバックアップファイルの存在を確認します。バックアップファイルが見つかったら、do ファイルエディタはバックアップファイルが存在することを知らせ、バックアップファイルを復元するか、元の文書を開くか、キャンセルするかを尋ねます。オプションについて逆の順序で説明します。キャンセルを選択すると、文書の開封はキャンセルされ、バックアップファイルはディスクに残ります。元の文書を開くを選択すると、元の文書が do ファイルエディタで開かれ、バックアップファイルはディスクから削除されます。バックアップファイルを復元を選択すると、バックアップファイルが新しい未保存の文書として do ファイルエディタで開かれ、デフォルトのファイル名が元のファイル名に設定され、ファイル名に "Recovered" が追加されます。バックアップファイルはディスクから削除されます。復元した文書は、新しいファイルとして保存するか、元の文書を上書き保存するかでディスクに保存できます。または、保存せずに文書を閉じて変更を無視することもできます。

13.12 ユーザ定義による構文ハイライト

自分で定義したキーワードを構文強調に使用するために、Stata はキーワードを含むテキストファイルを作成できます。テキストファイルの名前は stata-userkeywords.txt とし、キーワードのリストを含む必要があります。キーワードは、スペースやタブの組み合わせで区切ることができ、別々の行に配置できます。キーワードは、Stata の有効なコマンド名のルールに従う必要があります。コメントはサポートされていません。無効なコマンド名とされるキーワードは無視されます。定義できるキーワードの数に制限はありません。ただし、文書の構文強調を行う際に Stata がすべてのキーワードを検索する必要があるため、非常に大きなキーワードの辞書は do ファイルエディタのパフォーマンスに影響を及ぼす可能性があります。

ります。Stata は一意のキーワードの辞書を保持しており、同じキーワードの複数のインスタンスは無視されます。Stata はグローバルなキーワードファイルとローカルなキーワードファイルの両方を検索します。両方のファイルが存在する場合、そのキーワードの辞書が結合されます。グローバルなキーワードファイルは Stata ディレクトリに保存する必要があります。これにより、複数のユーザーと共有する際に、各ユーザーがコピーを持つ必要がなくなります。また、自分専用のローカルなキーワードファイルを作成し、ホームディレクトリに保存することもできます。Stata は起動時にキーワードファイルを読み込みます。Stata が実行中のキーワードファイルへの変更は、Stata を再起動する必要があります。

14. データを作図する

14.1 グラフを使う

Stata はデータからグラフを作成するためのシステムを幅広く取り揃えています。グラフを作成するのに最も重要

なコマンドは **graph** コマンドです。**Stata** ではこのシンプルなコマンドと様々なオプションを組み合わせてグラフを作図します。この章では例題として簡単なグラフを 1 つ作成し、**Graph** ウィンドウの基本を確認します。グラフ作成に関する詳細は [\[G\] Stata Graphics Reference Manual](#) をご覧ください。

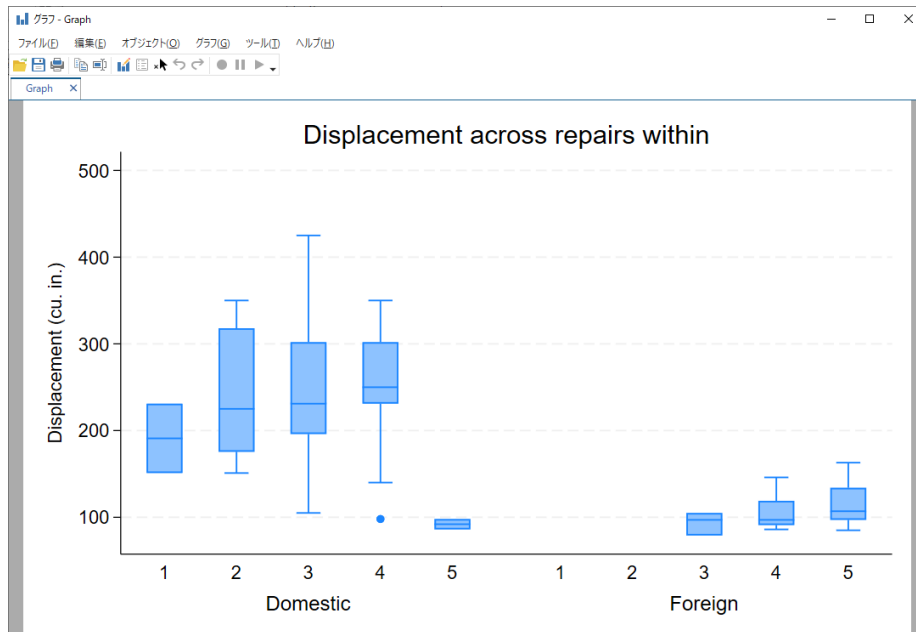
14.2 簡単なグラフのサンプル

[\[GSW\] 1 Introducing Stata—sample session \(Stata の紹介—サンプルセッション\)](#) のサンプルセッションでは、散布図を作成し、それにフィットした回帰線を追加してグリッドを作り、グループごとに比較しました。ここでは、**automobile** データセットを使用して簡単なボックスプロット（箱ひげ図）を作成します。**automobile** データセットの排気量と修理記録の関係をグループ間の差からみていきましょう。では、コマンドウィンドウに「**sysuse auto**」と入力して **Enter** を押し、データセットを開きます。

メニューから **グラフィックス > 箱ひげ図** を選択し、メインタブを選んで、変数欄に「**displacement**」を入力します。次にカテゴリタブを選び、グループ **1** のチェックボックスにチェックを付けて 1 番目のグループ変数を「**rep78**」にします。続いてグループ **2** のチェックボックスにチェックを付けて、「**foreign**」を 2 番目のグループ変数に入力します。最後に、必要に応じてグラフを修正できるように適用ボタンをクリックします。完成したグラフにタイトルを付け忘れたので、ここで追加しましょう。**Graph** ウィンドウを一度閉じて **graph box** ダイアログでタイトルタブを選びます。

そしてタイトルに「**Displacement across Repairs within Origin**」と入力し、再び適用ボタンをクリックします。



この一連の作業でタイトル付のグラフが出来上がります。



14.3 Graph ウィンドウ

Graph ウィンドウは作図したグラフとツールバーを表示します。最初の **4** つのボタンは他のウィンドウでも見慣れて

いる開く、保存、印刷、コピーのボタンです。残り **2** つのボタンは今回初めてなのでここで説明しておきます。

	グラフ名を変更する	グラフの名前を変更します。この機能は、複数のグラフを画面上で同時に開く時に利用します。グラフ名を変更するボタンをクリックするとグラフに名前を付けることができるので、別のグラフを作成してもこのウィンドウは開いたままになります。
	グラフエディタの開始	グラフの編集と加工をするためのグラフエディタを開きます。機能については次の章で紹介します。

これらのボタンの右にある灰色の（アクティブでない）ボタンについては次の章で説明します。

では作成したグラフを保存しましょう。保存ボタンをクリックし、ダイアログでフォルダを選んで名前を入力します。

Graph ウィンドウ内を右クリックしてコンテキストメニューから名前を付けて保存... としても同じ要領で保存できます。

14.4 グラフの保存と印刷


グラフ作成後ならウィンドウ内を右クリックし、名前を付けて保存... を選ぶと保存でき、グラフ表示後にウィンドウ内で右クリックをして印刷... を選ぶと印刷できます。ファイルメニューから保存または印刷を選択しても、保存または印刷できます。複数のウィンドウが開いている時に選択するグラフを間違えないためにも、右クリックをしてから保存または印刷する事をお勧めします。

14.5 Graph ウィンドウで右クリックする

Graph ウィンドウで右クリックをすると、コンテキストメニューに次の項目を表示します。

- 名前を付けて保存... グラフをディスクに保存します。
- コピー クリップボードにグラフをコピーします。
- グラフエディタの開始 グラフエディタを起動します。
- ユーザ設定... グラフの設定を編集します。
- 印刷... **Graph** ウィンドウの内容を印刷します。

14.6 Graph ボタン

Graph ボタン はメインツールバーにあります。このボタンにはアイコンと下矢印の 2 つの部分があります。アイコンをクリックすると開いている **Graph** ウィンドウのうち、一番上にあるウィンドウを最前面に移動します。下矢印をクリックすると開いているグラフウィンドウの一覧を表示します。この一覧からウィンドウを 1 つ選択すると、その選択したグラフが最前面に移動します。**Graph** ウィンドウを保存せずに閉じてから再び同じグラフを開くには、そのグラフを再作成しないといけません。

15. グラフを編集する

15.1 グラフエディタで作業をする

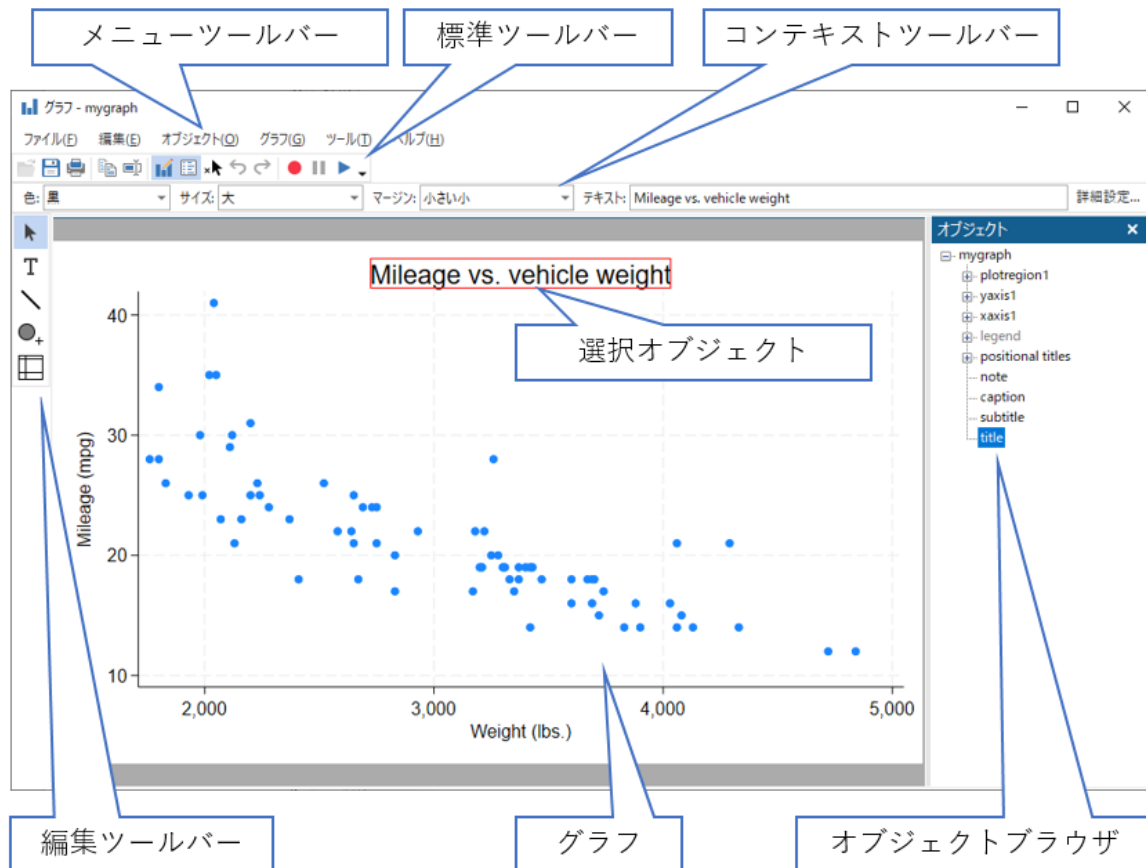
Stata のグラフエディタではグラフ上にテキスト、線、矢印、マーカーなどを自由に追加でき、グラフ上のオブジェクトのほぼ全てが編集可能です。


まずは実際に編集するために例題のグラフを準備しましょう。それからグラフエディタ内のツールを確認します。では、「`sysuse auto`」コマンドを実行して、**automobile** データセットを開きます。グラフを作成するためのコマンドは次の通りです。

```
. scatter mpg weight, name(mygraph) title(Mileage vs. vehicle weight)
```




グラフエディタを開くには、**Graph** ウィンドウで右クリックを行いグラフエディタを開始を選択します。
グラフのタ

イトルを 1 回クリックしてください。グラフエディタとそれぞれの要素の名前を次に示します。




グラフエディタの左側にあるツールを選択してグラフを編集します。デフォルトではポインタ(選択ツール)  が

選択されています。ポインタを使うとオブジェクトの設定を変更でき、そのオブジェクトをグラフ上でドラッグできます。グラフ内のオブジェクトをポインタで選択するとグラフのすぐ上にコンテキストツールバーを表示します。上記例題では、グラフタイトルを選択したので、コンテキストツールバーはタイトルを編集するための内容に変化します。コンテキストツールバーでは選択したオブジェクトの主要なプロパティを簡単に変更できます。オブジェクトを右クリックすると更に細部にわたってプロパティと操作の変更を行えます。**Shift** キーを押しながらオブジェクトをドラッグすると、水平または垂直方向に移動できるようになります。


テキスト、直線、マーカー(ラベル付けオプションあり)を追加するには、次の追加ツール 、 、  を使用します。直線はコンテキストツールバーで矢印に変更できます。デフォルトのプロパティを変更したい場合はテキスト、

線、またはマーカーを追加する前にコンテキストメニューで設定を変更できます。設定を行うと、このセッション以降、追加される全てのオブジェクトのデフォルト設定となります。




ツールに慣れるためにも、いろいろと試してみてください。結果が気に入らない時は、同じツールを使用して元に

戻すか、標準ツールバー（メインメニューの下）にある元に戻すボタンをクリックします。メインメニューで編集 > 元に戻すと操作しても同じです。

オブジェクトのドラッグやプロパティを変更する時は、ポインタ（選択ツール）を再度選択する必要があります。

グリッド編集ツールを選択すると、グラフ上のオブジェクト（軸ラベルや軸タイトル）を赤い罫線で囲まれたエリアに移動できます。この赤い罫線はグリッドと呼ばれ、オブジェクトの移動可能位置を表します。複数のグラフを同じウィンドウ内で表示する時（たとえば、**by** グラフを作成する時）は選択するオブジェクトにより、移動可能なグリッドが異なることもあります。オブジェクトはグリッドの外および他のグリッドへの移動はできません。

グラフの右側にあるオブジェクトブラウザを使用するとオブジェクトを選択できます。このウィンドウはグラフ内のオブジェクトの階層を表示します。オブジェクトブラウザでオブジェクトをクリックまたは右クリックすると、グラフ上でそのオブジェクトに同じことをしている状態になります。

グラフエディタには編集内容を記録して後から他のグラフに反映させる機能があります。記録を開始ボタンをクリックすると、グラフに対して行った編集をすべて記録します。これには「元に戻す」と「やり直す」も含まれます。記録しない編集を行う時は、記録を中断ボタンをクリックします。記録を開始ボタンをもう一度クリックすると、記録を再開します。記録したい内容が終了したら記録を開始ボタンをもう一度押します。ポップアップを表示するので記録を保存します。記録後保存した内容は記録を再生ボタンで見ることができ、それ以後に作成するグラフにも適用できます。記録したものを再生するには、Stata の `graph` コマンドに `play` オプションを付けます。詳しくは [\[G-1\]](#)

[Graph editor](#) 内の「[Graph Recorder](#)」をご覧ください。

グラフエディタを終了するにはメインメニューからファイル>グラフエディタの終了と操作するか、グラフエディタの終了ボタンをクリックします。グラフエディタを終了する時にグラフに変更がある場合、ポップアップを表示してグラフを保存するように促します。グラフを保存しなくても、すぐに変更内容が消えることはありません。しかし、同じウィンドウで新しいグラフを作成すると内容が消えるリスクが高くなります。**Stata**で他のタスクを実行する場合はエディタを中止しなければなりません。

グラフエディタで行える操作一覧の一部を次に示します。

- 線、矢印、テキストを使用して注釈を追加できます。
- グリッドラインやリファレンスラインの追加および削除ができます。
- タイトル、キャプション、メモの追加および編集ができます。
- 散布図 (**scatterplot**) を線 (**line**)、線 + シンボル (**connected**)、面積図 (**area**)、棒 (**bar**)、スパイク (**spike**)、または垂直ドロップライン (**drop line**) に相互変更できます。
- グラフタイトルおよびテキストのサイズ、色、余白、を含むプロパティを変更できます。
- 凡例をグラフ内のどこにでも配置できます。
- グラフの縦横比を変更できます。
- 棒グラフを積み上げに変更することや、その表示をパーセントに変えることができます。
- 軸ラベルの回転や角度の変更ができます。
- カスタムラベルや目盛りを軸に追加します。
- 軸のラベルと目盛りの数および間隔に関するルールを変更します。
- グラフ上のある一点の色、サイズ、シンボルをカスタム化して強調できます。この機能はマーカー棒、スパイク等のグラフで行えます。
- マーカーラベルのテキストやプロパティを変更します。

グラフ上の全オブジェクトのプロパティを編集できるので、結果的にグラフのほぼ全ての内容を変更できます。詳しくは [\[G-1\] Graph editor](#) をご覧いただくか、「`help graph editor`」を実行してください。

16. ログを使い結果の保存や印刷を行う


16.1 Stata でログを使う

Stata で分析を行う時、科学者が実験ノートを取るのと同じように操作を記録しておく、その分析を簡単に繰り返すことができます。集中して作業を行う最中はあたかも全てのやり方や情報を理解したような気になります。ところが、後日この分析を再現しようとする、重要部分の詳細をあいまいに記憶していることがあります。このような状況を回避するために、**Stata** には実験ノートの代わりとしてログファイルがあります。

ログファイルは、結果ウィンドウの内容を記録したものです。全てのコマンドとテキストによる出力を実行順に記録します。つまり、作業しながら実験ノートを自動で作成できます。ログファイルは結果ウィンドウに出力しながらディスク上に作成されるので、万一停電やコンピュータがクラッシュするような事故が起きても、それまでの作業内容は消えません。**Stata** で重要な作業を行う時はログファイルを作ることをお勧めします。

16.2 ログ出力

結果ウィンドウに表示される出力はすべてログファイルに記録できます。ログファイルのファイル形式は2種類あります。デフォルトでは、結果ウィンドウの形式とリンクを全てそのまま保存できる **Stata Markup and Control Language (SMCL)** という形式を使用します。**SMCL** ファイルはビューワで開くと結果ウィンドウの表示と同じように表示します。通常のテキストファイルとしてログが必要な時は、テキスト形式のログファイル (**.log**) として保存します。**SMCL** ファイルはファイル>ログ>変換...メニューで様々なアプリケーションに対応した形式に変換できるので、**SMCL** 形式で保存することをお勧めします。(詳しくは [\[R\] translate](#) をご覧ください。)

ログファイルを開始するにはログボタン  をクリックします。これはディレクトリとファイル名を指定し保存する、一般的なダイアログを表示します。ファイルの拡張子を指定しない時は、拡張子 **.smcl** がファイル名に追加されます。

既存のファイルを指定すると、そのファイルに新しいログを追加するか新しいログで上書きするか尋ねられます。

短いセッションの例は次の通りです。

```

name: <unnamed>
log: C:\Users\watabe\Documents\Stata\log-basic-base.smcl
log type: smcl
opened on: 29 Mar 2023, 17:45:11

. sysuse auto
(1978 automobile data)

. by foreign, sort: summarize price mpg

-> foreign = Domestic

```

Variable	Obs	Mean	Std. dev.	Min	Max
price	52	6072.423	3097.104	3291	15906
mpg	52	19.82692	4.743297	12	34

```

-> foreign = Foreign

```

Variable	Obs	Mean	Std. dev.	Min	Max
price	22	6384.682	2621.915	3748	12990
mpg	22	24.77273	6.611187	14	41

```

. * be sure to include the above stats in report!
. * now for something completely different
. corr price mpg
(obs=74)

```

	price
price	1.0000
mpg	-0.4686
mpg	1.0000

```

. log close
name: <unnamed>
log: C:\Users\watabe\Documents\Stata\log-basic-base.smcl
log type: smcl
closed on: 29 Mar 2023, 17:45:11

```

上記の例の中から何点か確認します。

- ファイルの保存場所、種類、開始時間のスタンプを含むヘッダーもログファイルの一部です。ヘッダーは複数のログファイルを使用する時に便利です。

- アスタリスク (*) から始まっている 2 行はコメントです。Stata はアスタリスクの後ろにあるテキストはコマンドとして認識せず、無視します。つまり、特殊記号などを使いながら好きなコメントを入力できます。コメントはその時の考えを残すのにも適しており、またログファイルをセクション分けすることにも利用できます。
- この例ではログファイルを `log close` コマンドで閉じましたが、Stata を閉じればログファイルも自動的に閉じるので、手動で閉じることは必須ではありません。

それぞれのファイルに名前がある時は、複数のログファイルを同時に開くことができます。詳細は `help log` コマンドで確認してください。

16.3 ログで作業する

ログファイルは Stata のビューワを使います。ファイル > ログ > 表示... を選びます。ログが開いている時は

(ログステータスバーでオン・オフの確認ができます) デフォルトでそのログを表示します。それ以外のファイルを開く

時はログファイルの名前を入力するか、参照... ボタンからファイルのダイアログを開き、リスト内から選んでください。

ビューワウィンドウでは、普通のビューワのファイルと同じようにテキストをコピーして他のビューワやワープロソフトにコピー・貼り付けを行えます。また、コマンドウィンドウや `do` ファイルエディタにも貼り付け可能です。その際の注意点として、結果ではなくコマンドだけをコピーするようにしてください。各コマンド文の冒頭にあるプロンプト (“.”) はコマンドウィンドウが無視するのでコピーしても問題ありません。ワープロソフトで作業を行う時に貼り付けたテキストは、書式設定のないテキストになります。貼り付けの時は固定幅フォント、たとえば **Courier** を使用すると見やすくなります。

自分の考えや分析の流れを見失わないように、現在のログファイルを時折確認するのも大切になります。ビューワウィンドウではログファイルのスナップショットも取れるので、作業をしながらスクロールする必要はありません。最新の結果をビューワに反映させたい時は、再読み込みボタンをクリックします。

ログに関する詳細は [\[U\] 15 Saving and printing output—log files](#) と [\[R\] log](#) をご覧ください。ビューワーに関する詳細は [\[GSW\] 3 Using the Viewer \(ビューワを使う\)](#) をご覧ください。

16.4 ログを印刷する

標準的な **SMCL** のログファイルを印刷する時は、最初にビューワウィンドウで目的のファイルを開きます。そして、ツールバーの印刷ボタンをクリックする、ビューワ内で右クリックして印刷...を選択する、メニューからファイル>印刷を選ぶ、のいずれかで印刷します。印刷ダイアログが表示されるので印刷をクリックすると、印刷設定ダイアログが出てきます。

- ヘッダー、名前、プロジェクトの欄は必要に応じて入力します。行番号を印刷する、ヘッダーを印刷する、ロゴを印刷するのオプションは必要な時にそれぞれのチェックボックスにチェックを付けてください。これらの設定は保存され、次回以降、ログの印刷を行う時に印刷設定に表示されます。
- フォント、余白、配色に関する設定は印刷設定ダイアログ内の設定...をクリックしてプリンタの設定ダイアログを開きます。**Monochrome** は白黒印刷設定、**Color** はデフォルトのカラー印刷設定、**Custom 1** と **Custom 2** はカラー印刷のカスタム設定です。フォントの設定にはフォント...ボタンをクリックします。フォントダイアログはプリンタで使用できる、幅が固定されている（タイプライターフォントとも呼ばれ、**Courier** 等があります）のみを表示します。

PostScript ファイルや **PDF** 形式のログファイルは **translate** コマンドで作成します。詳細は [\[R\] translate](#) をご覧ください。

テキストファイルとしてログを作成した時（拡張子が **.smcl** ではなく **.log**）、ファイルはテキスト編集用アプリケーション、たとえばメモ帳や **Stata** の **do** ファイルエディタ、あるいはワープロソフトで開くことができます。開いたログファイルは編集（見出しやコメントの追加など）し、その上で印刷もできます。ワープロソフトでログファイルを開くと、デフォルトのフォントで表示および印刷します。ログファイルは非固定幅フォント（たとえば **Times New Roman** や

Georgia) では配置が崩れ読みにくくなります。固定幅フォント (Courier New など) ならば配置が崩れることもないので、可能な限り固定幅フォントを使用してください。

Windows では拡張子が .log のファイルはテキストエディタ (たとえばメモ帳) を使用すると良いでしょう。そして必

要に応じ、Windows のアプリケーションで編集や印刷を行ってください。

16.5 do ファイルとしてコマンドを再実行する

Stata は出力結果を除き、コマンドだけをログとして記録できます。これは do ファイルをインタラクティブに使用する際に適しており、このようなファイルは cmdlog ファイルと呼ばれています。cmdlog ファイルを開始するにはコマンドウィンドウに次のように打ち込みます。

```
cmdlog using ファイル名
```

cmdlog ファイルを閉じるには次のように入力します。

```
cmdlog close
```

先程のセッションの内容を cmdlog で保存すると次のようになります。コマンドのみで構成されるので、このまま

do ファイルとして使用できます。

```
sysuse auto by foreign, sort:
summarize price mpg
* be sure to include the above stats in report!
* now for something completely different corr
price mpg
```

cmdlog ファイルを作らずに作業開始した後、cmdlog ファイルが必要になった時は履歴ウィンドウのコマンド一覧を do ファイルとして保存すれば問題ありません。履歴ウィンドウは実行した直近 5000 個のコマンドを記録しています。履歴ウィンドウで右クリックを行い、メニューからすべて保存... を選んでください。こ

の方法は、まずエラーが出力されたコマンドをフィルタにかけて取り除いてから実行してください。詳細については [\[GSW\] 2 The Stata](#)

[user interface \(Stata ユーザーインターフェイス\)](#) 内の「履歴ウィンドウ」をご覧ください。do ファイルエディタに直接コマンドを送りたい時は右クリックで表示するコンテキストメニューから選択範囲を do ファイルエディタへ送るを選びます。この方法はセッション中に入力したコマンドのみでテキストファイルを作成する時に便利です。

詳しくは [\[GSW\] 13 Using the Do-file Editor—automating Stata \(do ファイルエディタを使用する—Stata の自動化\)](#)、[\[U\] 16 Do-files](#)、[\[U\] 15 Saving and printing output—log files](#) をご覧ください。

17. ウィンドウやフォントの設定をする

17.1 ウィンドウの位置、大きさ、フォントを変更して保存する

モニターの画質や好み等により、Stata のウィンドウの表示形式やフォントを変更したい場合もあります。また、グラフなどを論文で利用する際にはフォントの大きさ等について規則があるかもしれません。Stata では上記 2 つのような状況に対応するため、ウィンドウの表示設定を変更できます。

まず、各ウィンドウで設定可能なプロパティを確認した後、初期設定の管理方法について紹介します。

17.2 Graph ウィンドウ

Graph ウィンドウ上で右クリックを行い、コンテキストメニューからユーザ設定... を選ぶと、Graph ウィンドウの設定を変更できます。表示されるダイアログでグラフの表示形式を変更できます。プリンタタブでは印刷時の設定を、クリップボードタブではクリップボードの設定を変更できます。

グラフの初期設定はグラフの見せ方を複数のスキーム (scheme) に分けてコントロールします。スキームを使い分けることによって、目的のグラフを簡単に作成できます。つまり、**The Economist** や **Stata Journal**

向けのスキームもあり、これらの出版物の規定で簡単にグラフを作成できます。スキームを変更しても現在のグラフには反映しません。設定したスキームは次に作成するグラフから適用されます。

17.3 その他全てのウィンドウ

ディスプレイのフォントとフォントサイズの変更は **Stata** のほぼ全てのウィンドウで行えます。

ウィンドウのフォントとフォントサイズを変更できる時は、コンテキストメニューにフォント... が表示されます。フォ

ント... を選択するとフォントダイアログが表示され、フォントの種類とサイズを選ぶことができます。結果、ビューワ、**do** ファイルエディタの各ウィンドウでは固定幅フォントのみ選択できます。固定幅フォントは出力結果の数字と文字を整列させ、読みやすくするために使用します。上記以外のウィンドウでは特に制限はなく、どのフォントでも使用できます。

17.4 配色を変更する

フォントだけでなく、背景や前景のテキストの色を変更することもできます。構文のハイライトカラーを選択すると、**Stata** のコマンドをその他のテキストと異なる色で表示できます。この設定は、**do** ファイルエディタとコマンドウィンドウの両方に適用されます。これらの配色は一般的なユーザ設定の標準タブで制御します。結果とビューワウィンドウでは入力、テキスト、結果、エラー、リンク、選択テキストの表示色を選ぶことができます。それぞれ同じように設定されているので、各ウィンドウで右クリックを行い、それぞれ配色を選ぶか自分で配色を作成します。結果とビューワウィンドウのデフォルト設定はビルトインの“標準配色”になり、これは白い背景に黒いテキストを使用します。他にビルトイン配色と、カスタム配色を設定できるスロットがあります。ビューワの設定変更を行うと一度に全てのビューワウィンドウを変更します。

17.5 複数の設定セットを管理する

Stata の設定は **Stata** を閉じた時に自動的に保存され、次回 **Stata** を開く時にリロードされます。しかし、時には

Stata のウィンドウを並べ替えた後、以前に設定した別の並びに戻したいこともあるでしょう。以前の並びに戻すには変更した設定に名前を付けて保存します。そして必要に応じて、保存した設定をロードします。設定のロード後に変更を加えても、上書き保存をしない限りこの設定が書き換えられる心配はありません。

メニューから編集 > ユーザ設定を開き、設定を管理します。各項目の詳細は次の通りです。

- 設定集を読み込む ロードしたい設定セットを選び、ロードします。設定セットは **Stata** に初めからインストールしてあります。小さなスクリーン向けの物や、**Stata** を使用したプレゼンテーション向けの物もあります。一度目を通してみるといいでしょう。
- 現在の設定を設定セットに保存するには設定集の保存>新規設定集...を選びます。このセットに名前を付け OK をクリックすれば保存できます。
- 設定集の保存メニューを開いて保存すれば、現在の設定セットを上書き保存できます。
- 設定集を削除 削除したい設定セットを選択して削除します。本当に削除して良いか確認されるので OK をクリックしてください。

17.6 ウィンドウを閉じたり開いたりする

結果とコマンドウィンドウ以外のウィンドウは閉じることができます。現在閉じているウィンドウを開く時はウィンドウメニューから開きたいウィンドウを選んでください。

18. Stata について詳しく学ぶ

18.1 Stata を学ぶ上での次のステップ

今までの学習で **Stata** を使用するのに十分な情報をカバーしました。しかし、本マニュアルで説明した内容は基礎的な操作方法に関する事なので、統計の分析とデータ管理分野について学習したい方は次のように操作してください。

- 興味深いデータセットを入手して **Stata** を使いながらいろいろ試してみてください。

- (a) まず、メニューやダイアログを使用して操作を行います。次に、結果ウィンドウにどのようなコマンド

が出力されるのか確認します。**Stata** はシンプルで分かりやすいコマンド構文を使っているの
で、コマンドの理解は比較的容易にできます。また、コマンドは簡単に覚えることができる
ので、コマンド入力を積極的に利用しましょう。結果として、速く作業できるようになりま
す。

- (b) グラフについてはグラフエディタでいろいろ試してみてください。

- コマンドウィンドウを使い始めると、短時間でより多くの操作を行えるようになります。同時に、新たな問

題として、理由の分からないエラーが表示され、戸惑うこともあるでしょう。このような時は、
次のように対応してください。

- (a) 「**help** コマンド名」を実行するか、メニューからヘルプ > **Stata** のコマンド... と選択し、コ
マンド名を入力してください。

- (b) ヘルプファイル内にあるコマンド構文例を確認し、コマンドウィンドウに入力した物と比較
してください。**Stata** は小さな誤字・脱字でもコマンドを実行できなくなるので、細部まで比
べる必要があります。

- メニューでヘルプ > 検索... と選択して **Stata** 内を検索してください。ヘルプファイル内の多くの統
計的ルーチンが載っているので、役に立つでしょう。
- **Stata Index** 内の **Combined subject table of contents** に一通り目を通し、わからない箇所を詳しく
読みます。
- **Stata** を操作しながら **User's Guide** を読んでください。**User's Guide** は最初から最後まで通し
て読むように作成しており、**Stata** のエキスパートになるために必要な情報をほぼ全て含んでいま
す。一読の価値は十分にあるでしょう。それほど上級者向けの技術を身に着ける必要が無い人
は、この章の後半にあるアドバイスを参考にしてください。
- 各リファレンスマニュアルに目を通してください。その際、目的の統計手法のことだけを読んで
ください。また、リンクをたどって他のトピックも必要に応じて参照できます。リファレンスマ
ニュアルは最初から最後まで順に読むようには作られていません。むしろ、百科事典のように必
要な箇所だけを読むようになっています。マニュアルの中で使用するデータセットはメニューか

らファイル> 例題データセット... を選択して表示される、**Stata 18 manual datasets** 項目にあります。データセットを活用して、例題にスムーズに取り組むことができます。

- **Stata** に関する情報は更にあり、よくある質問 (FAQ) もその 1 つです。次の web アドレス (<https://www.stata.com/support/faqs/>) から確認できます。
- **Stata** の資料ページ (<http://www.stata.com/links/>) にはたくさんの役立つリンクがあります。このページには数多くの **Stata** に関するすばらしい資料のリンクがあります。
- **Stata** と統計に関する情報交換を行うリストサーバ、**Statalist** に参加するのもいいでしょう。
- **The Stata Blog: Not Elsewhere Classified** (<https://blog.stata.com/>) では **Stata** 社の社員が書いた記事を読み、**Stata** についての理解を深めることもできます。
- **Stata** の Facebook ページ (<https://facebook.com/statacorp/>) を見たり、
- Instagram (<https://www.instagram.com/statacorp/>) で **Stata** に参加したり、
- LinkedIn (<https://www.linkedin.com/company/statacorp/>) を確認したり、
- Twitter (<https://twitter.com/stata>) でフォローしたりすることで、最新情報を確認できます。
- 様々な分野で統計を使用する研究者に有益な情報を、論文、コラム、本のレビュー等の形で掲載する

Stata Journal を定期購読するのもおすすめです。**Stata Journal** の定期購読申込みは

<https://www.stata-journal.com> からどうぞ。

- **Stata** に関する副読本も提供しています。興味のある人は **Stata Bookstore** (<https://www.stata.com/bookstore/>) を訪ねてみてください。
- **Stata NetCourse®**, **NetCourse 101** は **Stata** について学ぶのに最適なコースです。コースの情報と日程に関しては <http://www.stata.com/netcourse/> をご覧ください。
- **Stata** 社主催の **Stata** レーニングに参加するのもアイデアの 1 つです。コースの情報と日程に関しては <https://www.stata.com/training/classroom-and-web/> をご覧ください。
- **Stata** の動画を見るには、<https://www.youtube.com/user/statacorp/> にアクセスしてください。

18.2 User's Guide およびリファレンスマニュアルのおすすめ項目

User's Guide は最初から最後まで通して読むことを前提として作られています。逆にリファレンスマニュアルは必要な時に、必要な項目だけを参照します。

本マニュアルを読み終えてから **User's Guide** も最初から最後まで読んでいただくのが理想です。しかし **Stata** をすぐにある程度使えるようになりたいという場合は、次に示す **User's Guide** とリファレンスマニュアルの項目を先に参照してください。

この一覧では基本機能を紹介する項目と、見落としがちですが便利な機能について紹介します。

Stata の基礎要素

[\[U\] 11 Language syntax](#)

[\[U\] 12 Data](#)

[\[U\] 13 Functions and expressions](#)

データ管理

[\[U\] 6 Managing memory](#)

[\[U\] 22 Entering and importing data](#)

[\[D\] import](#) — Stata へのデータのインポートに関する概要

[\[D\] append](#) — データセットのアペンド

[\[D\] merge](#) — データセットのマージ

[\[D\] compress](#) — メモリ内のデータの圧縮

[\[D\] frames Intro](#) — フレームに関する概要

グラフィックス

[\[G\] Stata Graphics Reference Manual](#)

再現可能な調査

[\[U\] 16 Do-files](#)

[\[U\] 17 Ado-files](#)

[\[U\] 13.5 Accessing coefficients and standard errors](#)

[\[U\] 13.6 Accessing results from Stata commands](#)

[\[U\] 21 Creating reports](#)

[\[RPT\] Dynamic documents Intro](#) — ダイナミックドキュメントに関する概要

[\[P\] putdocx Intro](#) — Office Open XML (.docx) ファイルの作成に関する概要

[\[P\] putexcel](#) — 分析結果の Excel ファイルへのエクスポート

[P] **putpdf Intro** — PDF ファイルの作成に関する概要

[R] **log** — セッション履歴のファイルへのコピーする

見落としがちな便利な機能

[U] **29 Using the Internet to keep up to date**

[U] **19 Immediate commands**

[U] **24 Working with strings**

[U] **25 Working with dates and times**

[U] **26 Working with categorical data and factor variables**

[U] **27 Overview of Stata estimation commands**

[U] **20 Estimation and postestimation commands**

[R] **estimates** — 推定結果の保存・管理

基本的な統計

[R] **anova** — 分散共分散分析

[R] **ci** — 平均/比率/カウントの信頼区間

[R] **correlate** — 変数の相関

[D] **egen** — **generate** の拡張機能

[R] **regress** — 線形回帰

[R] **predict** — 推定後の予測値、残差等の取得

[R] **regress postestimation** — **regress** 推定後のツール

[R] **test** — 推定後の線形仮説検定

[R] **summarize** — 記述統計量

[R] **table intro** — 頻度、要約、コマンド結果の表の概要

[R] **tabulate oneway** — 一元配置表 [R]

tabulate twoway — 二元配置表

[R] **ttest** — t 検定 (平均比較検定)

行列

[U] **14 Matrix expressions**

[U] **18.5 Scalars and matrices**

[M] **Mata Reference Manual**

プログラミング

[U] [16 Dofiles](#)

[U] [17 Adofiles](#)

[U] [18 Programming Stata](#)

[R] [ml](#) — 最尤法

[P] [Stata Programming Reference Manual](#)

[M] [Mata Reference Manual](#)

システム値

[R] [set](#) — システムパラメータに関する概要

[P] [creturn](#) — **c-class** パラメータの表示

インターネットの情報

Stata の **Web** サイト (<https://www.stata.com>) では、**Stata** の追加情報を得ることができます。**Web** サイトには **FAQ** の確認、他のユーザとの交流の場の提供、**Stata** の公式アップデート情報など、役立つ情報を掲載しています。また、**Stata** と統計に関する情報交換のためのリストサーバである、**Statalist** にも申し込みます。

ここではインターネット上で提供されているインタラクティブなコースである、**Stata NetCourse**® に関する情報も入手できます。このコースの長さも数週間から8週間までの幅がありますので、ご都合に合わせてお選びください。さらに **Stata** はウェブベースの参加型トレーニングやウェビナーも行っています。詳しくは

<https://www.stata.com/learn/>をご覧ください。

ウェブサイト内には **Stata** ユーザが興味を持ちそうな本を取り揃えているページ、**Stata Bookstore** もあります。それぞれの本には技術スタッフからユーザが興味を持ちそうな理由が簡単なコメントとして付いています。

一度 **Stata** のウェブサイトを確認してみてください。 **Stata** をオンライン上で登録し、 **Stata News** の無料購読のお申込みもできます。

本やマニュアル等、 **Stata Press** の出版物の情報は <https://www.stata-press.com> をご覧ください。マニュアル内で使用する例題のデータセットは **Stata Press** のウェブサイトにあります。

Stata Journal は年4回発行し、統計、データ分析、教授方法、 **Stata** 言語の有効な活用法などについての記事を掲載します。ウェブサイト <https://www.stata-journal.com> をご確認ください。

Stata の公式ブログ (<https://blog.stata.com>) には **Stata** のニュースや使用方法に関するアドバイスの記事を掲載しています。ブログに投稿する記事はそれぞれ記名式になっており、 **Stata** の開発・サポート・販売を実際に行う **Stata** 社の社員が書いています。また、 **Stata Blog: Not Elsewhere Classified** には世界中の **Stata** ユーザーが作成した **Stata** に関するブログへのリンクもあります。

Stata の Facebook (<https://facebook.com/statacorp/>)、 Twitter (<https://twitter.com/stata/>)、 Instagram (<https://instagram.com/statacorp/>)、 LinkedIn (<https://www.linkedin.com/company/statacorp/>) といった情報源を活用してみてください。これらは最新の **Stata** に関する情報が確認できます。さらに、 Twitter には、フランス語版：[https://twitter.com/stata fr](https://twitter.com/stata_fr) とスペイン語版：<https://twitter.com/stata es> もあります。 **Stata** の使用方法に関する例題を集めた短い動画は YouTube (<https://www.youtube.com/user/statacorp/>) で見ることができます。

Stata の公式アップデートにアクセスする方法や **Stata** の無料追加の詳細などは [\[GSW\] 19 Updating and extending Stata—Internet functionality \(Stata のアップデートと拡張—インターネットでの機能\)](#) をご覧ください。

19. Stata のアップデートと拡張—インターネットでの機能

19.1 Stata のインターネットでの機能性

Stata はインターネットとうまく動作できるようになっています。自分のコンピュータに保存してあるファイルを開くように、インターネット上のデータセットやヘルプファイルを表示できます。また、ユーザが許可すれば、インターネットを通して自動更新も行えます。さらに、**Stata** の機能を拡張するコミュニティ投稿コマンドのインストールもできます。これらのコマンドは **Stata Journal (SJ)** で発表したものか、ユーザが作成して **Stata** のコミュニティに共有したものです。

この章では **Stata** の可能性を広げる方法を紹介していきます。

19.2 インターネットからのファイルを使用する

Stata は URL をローカルファイルのパスと同じように認識します。ウェブ上のデータセット、グラフ、**do** ファイルを使用する時は、**Stata** で簡単に開く事ができます。1 つ例題を紹介しましょう。

<https://www.stata-press.com/data/>には多くのデータセットがあります。例えば、[\[U\] 11 Language syntax](#) 内で使用するデータセット **census12** は、

<https://www.stata-press.com/data/r18/census12.dta>

にあります。ではデータセットを開いてみましょう。データセットを開くには **use** で始まる、次のようなコマンドを使用します。

```
. use https://www.stata-press.com/data/r18/census12.dta
(1980 Census data by state)
```

```
. describe
```

Contains data from https://www.stata-press.com/data/r18/census12.dta

```
Observations:    50      1980 Census data by state
Variables:       7      6 Apr 2022 15:43
```

Variable name	Storage type	Display format	Value label	Variable label
state	str14	%14s		State
state2	str2	%-2s		Two-letter state abbreviation
region	str7	%9s		Census region
pop	long	%10.0g		Population
median_age	float	%9.2f		Median age
marriage_rate	float	%9.0g		
divorce_rate	float	%9.0g		

```
Sorted by:
```

URL をローカルファイルと同じように認識する機能は **Stata** の至る所で見られます。構文内で「ファイル名」を使用するものは、そこに **URL** を入力することができます。

この機能の例題として **HTTPS** というプロトコルを使用してファイルを取り出す機能があります。**Stata** は **HTTP**、**FTP** の各プロトコルを理解できます。

19.3 Stata のアップデート

Stata の追加情報は大きく分けて 2 つに分類できます。1 つは **Stata** 社が提供するアップデートをはじめとした情報、もう 1 つは世界各国の **Stata** ユーザが作成したコミュニティ投稿プログラムです。コミュニティ投稿プログラムは主に **Stata Journal** で出版され、**Statalist** 等を通して提供されるものを指します。

Stata が提供しているアップデート（以後、公式アップデートと呼びます）とコミュニティ投稿プログラムはインターネットを通して入手します。まずは、公式アップデートから紹介しましょう。**Stata** 社は公式 **Stata** のアップデートファイルを配信します。アップデートファイルは新規機能の追加やバグ修正を行うものがあります。

Stata はデフォルトで自動アップデートチェックがオンになっており、7日ごとにアップデートのチェックを行います。設定の変更や確認にはメニューから編集 > ユーザ設定 > 一般的なユーザ設定... を選択し、インターネットタブをクリックします。

動かす簡単にバックグラウンドで **Stata** を最新版にできるので、自動アップデートチェックはおすすめです。デフォルトの状態ですばらくアップデートの有無をチェックしていない時は、**Stata** 起動時にポップアップを表示して警告します。

手動でアップデートの有無を確認するには、メニューからヘルプ > アップデートのチェックを選択するか、コマンドウィンドウで「`update query`」を実行します。どちらの方法でも、同じように確認できます。チェックが終了すると、アップデートのステータスを結果ウィンドウに表示します。最新版に更新済みの場合は、結果ウィンドウにその内容を表示します。アップデートが必要な時は、結果ウィンドウにその内容と `Install available updates` というリンクを表示します。このリンクをクリックするか「`update all`」コマンドを実行するか、どちらの場合でも、最新版にするために必要なものをダウンロードできます。アップデート終了後に一度 **Stata** は再起動するので、現在のセッションで続けたい作業（例えば、コマンド履歴の保存など）がある時は、アップデートを先送りにする事もできます。

トラブルシューティングメモ：C:\Program Files\Stata18 への書き込み許可がない場合、この方法で公式アップデートをインストールすることはできません。公式アップデートのダウンロードは行えますが、`update` コマンドを使用する必要があります。詳しくは [\[U\] 29 Using the Internet to keep up to date](#) をご覧ください。

19.4 自動アップデートチェック

Stata は自動で定期的なアップデートチェックを行います。デフォルトでは、7日に一度の頻度で、**Stata** 社のウェブサイトではアップデートの有無を確認します。間隔は自動手動を問わず、最後に「`update query`」を実行した時から7日です。チェック間隔の変更も行えます。

Stata のアップデートチェックは、今すぐにチェックを行う、次回 Stata を起動する時に行う、次の更新確認まで先送りにする、という 3 つの選択肢を、インターネットに接続する前に尋ねます。このポップアップを無効にし、確認を求めずにアップデートチェックを行う設定もあります。

アップデートが可能な時、Stata はメッセージを表示します。その後は Stata の指示に従ってアップデートしましょう。

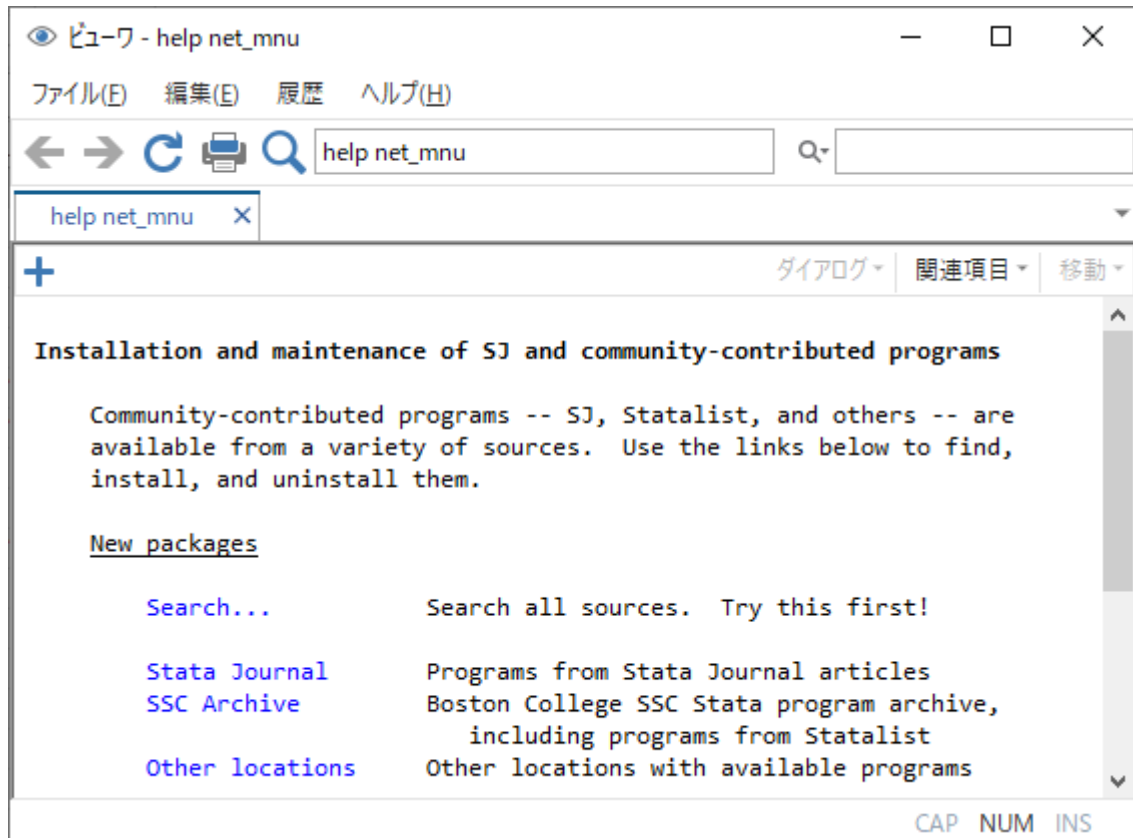
自動アップデートの設定変更は、メニューから 編集 > ユーザ設定 > 一般的なユーザ設定... と選択し、インターネットタブで行います。

19.5 コミュニティ投稿のプログラムをキーワードで探す

Stata にはインターネット上のコミュニティ投稿プログラムを検索するビルトインユーティリティがあります。アクセスするにはメニューからヘルプ > 検索... を選択し、インターネットリソースの検索を選んで、キーワードを入力します。また、ヘルプ > Stata ジャーナル/コミュニティ投稿コマンドを選ぶと、更に細分化した検索用の選択肢を表示します。このユーティリティでは **SJ** プログラムを含む、インターネット上のすべてのコミュニティ投稿プログラムを検索します。ビューワに表示された項目をクリックすると、その項目のヘルプファイルに移動できます。他の検索用コマンド「`search キーワード, net`」の構文については [\[R\]search](#) をご覧ください。

19.6 コミュニティ投稿のプログラムをダウンロードする

コミュニティ投稿プログラムは簡単にダウンロードできます。メニューからヘルプ > Stata ジャーナル/コミュニティ投稿コマンドと選択するところから始めます。



上記ビューワが示すように、まずは検索...を行います。

ロジスティック回帰の適合度に関する情報かコミュニティ投稿プログラムを探していると仮定します。メニューでヘルプ>検索...と選んだ後に全てを検索を選択します。検索ボックスに「**logistic goodness of fit**」と入力してOKボタンをクリックします。

ビューワ - search logistic goodness of fit

ファイル(E) 編集(E) 履歴 ヘルプ(H)

search logistic goodness of fit

search logistic goodness of fit

ダイアログ 関連項目 移動

search for logistic goodness of fit (manual: [R] search)

Search of official help files, FAQs, Examples, and Stata Journals

[R] logistic postestimation Postestimation tools for logistic
(help [logistic postestimation](#))

[R] estat gof Pearson or Hosmer-Lemeshow goodness-of-fit test
(help [logistic estat gof](#))

FAQ What statistical analysis should I use?
. UCLA Academic Technology Services
5/08 <https://stats.idre.ucla.edu/stata/whatstat/what-statistical-analysis-should-i-usestatistical-analyses-using-stata/>

Example Textbook examples: Applied Logistic Regression (2nd Edition)
. UCLA Academic Technology Services
2/08 examples from the book **Applied Logistic Regression**
(2nd Edition) by David Hosmer and Stanley Lemeshow
<https://stats.idre.ucla.edu/other/examples/alr2/>

Example Seminar: Logistic regression with Stata
. UCLA Academic Technology Services
5/07 seminar to help increase skills using logistic
regression analysis
<https://stats.idre.ucla.edu/stata/seminars/stata-logistic/>

SJ-17-4 [gr0071](#) . Calibration of dichot. outcome models with calibration belt
. . G. Mattino, S. Lemeshow, G. Phillips, S. Finazzi, and G. Bertolini
(help [calibrationbelt](#) if installed)
Q4/17 [SJ 17\(4\):1003--1014](#)
implements the calibration belt (a graphical approach to
evaluate the goodness of fit of binary outcome models by
examining the relationship between estimated probabilities
and observed outcome rates) and its associated test

SJ-13-2 [st0299](#) Goodness-of-fit tests for categorical data
. R. Bellocco and S. Algeri
Q2/13 [SJ 13\(2\):356--365](#) (no commands)
discusses choice of analytical units of reference (subjects
or groups of subjects that have the same covariate pattern)
and how that affects the definition of the saturated model
and conclusions from goodness-of-fit tests

CAP NUM INS

1 番目の項目は、ロジスティック回帰の実行後に利用可能な全ての推定後の分析コマンドのヘルプです。

2 番目の項目は、ロジスティック回帰後の適合度統計を計算するための **Stata** ののビルトインコマンド **estat gof** のヘルプです。リンクをクリックして内容を確認してみましょう。次の **3** つの項目は **UCLA**（カリフォルニア州立大学ロサンゼルス校）の **web** サイトの **FAQ** および **examples** です。さらに次の **3** つの項目 **Stata Journal** の論文へのリンクです。

多項ロジット回帰に関心がある場合は、最後のリンクを確認してみましょう。これは **SJ** の **volume 12, number 3** (2012 年の第 3 四半期) に投稿された記事です。画面上の **st0269** のリンクをクリックすると、この記事で紹介されたコマンドの情報ページに移動します。

Viewer - net sj 12-3 st0269

ファイル(F) 編集(E) 履歴 ヘルプ(H)

net sj 12-3 st0269

net sj 12-3 st0269

package **st0269** from <http://www.stata-journal.com/software/sj12-3>

TITLE
SJ12-3 st0269. A generalized Hosmer-Lemeshow...

DESCRIPTION/AUTHOR(S)
A generalized Hosmer-Lemeshow goodness-of-fit test for multinomial logistic regression models
by Morten W. Fagerland, Unit of Biostatistics and Epidemiology, Oslo University Hospital, Norway
David W. Hosmer, Department of Public Health, University of Massachusetts, Amherst, Amherst, MA
Support: morten.fagerland@medisin.uio.no
After installation, type `help mlogitgof`

INSTALLATION FILES [\(click here to install\)](#)
[st0269/mlogitgof.ado](#)
[st0269/mlogitgof.sthlp](#)

ANCILLARY FILES [\(click here to get\)](#)
[st0269/example.do](#)

[\(click here to return to the previous screen\)](#)

CAP NUM INS

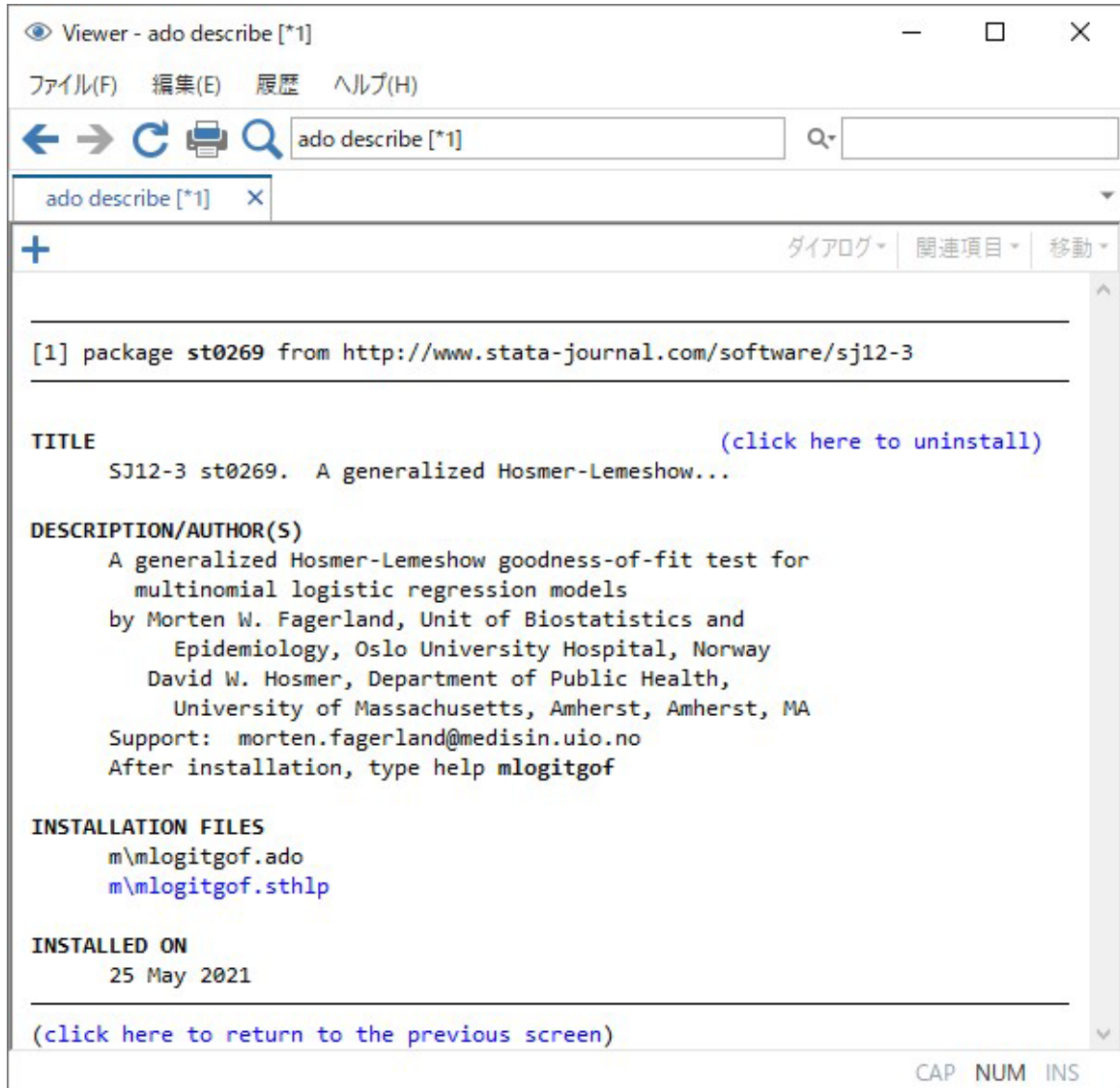
このパッケージには新しいコマンドに対して、1つのヘルプファイルがある事がわかります。

`st0269/mlogitgof.sthlp` リンクをクリックし、`mlogitgof` コマンドがどのようなものか確認してください。このコマンドを実際にインストールして試すには、一度戻るボタンをクリックしてから [click here to install](#) リンクをクリックします。付属の補助ファイル（コマンドの仕組みについて説明を行うものが一般的）も使用するには、[click here to get](#) のリンクをクリックするとダウンロードできます。このようにファイルをダウンロードしても **Stata** の動作に影響はありません。これらのコミュニティ投稿コマンドを安全にアンインストールする方法はこの後すぐに紹介します。

コミュニティ投稿コマンドを最新の状態に更新するには `ado update` コマンドを使用します。「`ado update`」はアップデートの有無の確認を、「`ado update, update`」はアップデートの有無の確認とインストールを行います。このパッケージを削除しましょう。アンインストールはとても簡単で、メニューからヘルプ> **Stata** ジャーナル/コミュニティ投稿コマンドと選択し、List リンクをクリックします。すると、次のような画面を表示します。



この画面でプログラムのオンライン解説を選択すると、インストールしてあるプログラムの解説を表示します。この画像は一番下までスクロールすると確認できるダイアログです。もちろん、実際の画面とこの画像はインストール日付が異なります。



削除（アンインストール）するにはパッケージの説明画面で **click here to uninstall** をクリックし、表示されるダイアログで **OK** をクリックします。

`net` コマンドを使用してコミュニティ投稿プログラムのダウンロードを行う詳細については [\[R\] net](#) をご覧ください。

A. Stata のトラブルシューティング

A.1 Stata が起動しなかったら

Stata を起動しようとして失敗する時、Stata または OS がエラーメッセージを表示します。エラーメッセージが表示された時の対処法はメッセージの種類により異なります。次の対処法を参考にしてください。

Cannot find license file

このメッセージは「ライセンスファイルが見つからない」事を示します。このエラーは深刻なものではなく、Stata は目的のライセンスファイルを発見できなかっただけです。このエラーが表示される一番の理由は、インストール手順が完了していないためです。

DVD と同梱してある、ライセンス用紙のコードをインストール時に入力してロックを解除しましたか。未解除の時は一度 Stata に戻り、初期化手順を完了してください。

Error opening or reading the file

技術的な理由で何か明確なエラーが発生したことを示します。Stata は目的のファイルを発見しましたが、そこで

I/O エラーが発生しています。

このエラーが発生するほぼ唯一の原因はハードディスクのエラーです。その場合、Stata のテクニカルサポートまでご連絡ください。詳しくは [\[U\] 3.8 Technical support](#) をご覧ください。

License not applicable

有効なライセンスはありますが、そのライセンスの情報でインストールした Stata のエディションが一致していないと認識しています。

例えば、Stata/BE 用のライセンスがあるのに間違えて Stata/SE や Stata/MP をインストールしようとしたり、

Stata/SE 用のライセンスを持っているのに Stata/MP のいずれかをインストールしたりする場合があります。このように、お持ちのライセンスとインストールのエディションが一致していない時は、インストール用の DVD を再度読み込ませ、もう一度インストーラーを起動します。そして **Modify** を選び、**Next** ボタンをクリックしてお手元のライセンスと同じエディションの Stata を選んでください。

その他のメッセージ

上記以外のメッセージは **Stata** がライセンスでは許可していない行為を行っているとは判断しています。例えば、ネットワークライセンスをお持ちでは無いのに **Stata** をネットワーク上で起動しようとする事です。しかし、考えられる理由はこれだけではなく、他にも多くの可能性があります。このような場合、次の **2**つのどちらかに分類されます。**1**つは本当にお手元のライセンスでは認められていない行為を行っている事で、もう **1**つはインストールした **Stata** に問題がある場合です。どちらの場合でも、一度お問い合わせください。詳しくは [\[U\] 3.8 Technical support](#) をご覧ください。

A.2 トラブルシューティングのヒント

想定外の問題に直面した時は、まず最新版の **Stata** を使用している事を確認します。アップデートに関する詳細

は [\[GSW\] 19 Updating and extending Stata—Internet functionality \(Stata のアップデートと拡張](#)

[—インターネットでの機能\)](#) をご覧ください。それでも解決しない時は、**Stata** の **Web** サイトにある、ユーザサポートセクションの **Windows** 版・よくある質問 (**FAQ**) (<https://www.stata.com/support/faqs/windows/>) をご覧ください。ウェブページに問題の解決策が載っているかもしれません。**FAQ** にも情報が無い時はサポートチームにお問い合わせいただけますが、その際には可能な限り多くの情報を提供していただいております。

一度コンピュータを再起動して **Stata** を改めて開始します。そして、問題が再び起こるまでの作業手順を書き残しながら現象を再現してみてください。その手順の情報が必要になります。

以前は問題なく動作していた **Stata** が、急に動かなくなった時は、最近インストールしたハードウェアまたはソフトウェアを思い出してください。

更にコンピュータに関する、できるだけ多くの情報を提供してください。**Windows** のバージョン、メモリの残量、プロセッサの種類、コンピュータのメーカー、**Stata** のシリアル番号とリリース日 (**revision date**) が必要です。**Email** で連絡する時はこれらの情報を分かる範囲で教えてください。電話連絡の時はお手元に準備してからご連絡ください。これらの情報は **Stata** のコマンドウィンドウで「**about**」と実行すると表示されます。**about** コマンドではバージョンやリリース日を含む、使用中の **Stata** に関する全ての情報を表示します。

B. 上級者向け Stata の使用法

B.1 Windows プロパティシート

Windows でショートカットをダブルクリックしてアプリケーションを起動する時、実際はショートカットのプロパティシートに定義してあるコマンドを実行しています。ショートカットのプロパティシートを開くにはショートカットを右クリックしてプロパティを選択します。

では **Stata** のショートカットプロパティシートを開きましょう。ショートカットタブをクリックすると、次のように表示がされます。

StataSE

種類： アプリケーション

場所： **Stata18**

リンク先： **C:\Program Files\Stata18\StataSE.exe /UseRegistryStartin**

作業フォルダ： ショートカット

キー： なし

実行時の大きさ： 通常のウィンドウ

使用中の **Windows** のバージョンによりエリアの名称は異なるかもしれません。ファイル名やフォルダの場所は上記の例とは異なる場合もあります。注意事項として、リンク先と作業フォルダの内容の **2** つがあります。リンク先には実際に **Stata** を起動するために実行されるコマンドが入っています。アプリケーションを起動する前に「作業フォルダ」情報を入力しておく、ディレクトリの変更ができます。必要に応じて変更を施し、OK を押すとプロパティシートが保存されます。これにより、**Stata** を様々な作業フォルダで起動できます。必要に応じ、パラメータ **/UseRegistryStartin** を「リ

ンク先」から削除してください。**Stata** のプロパティシート内の「作業フォルダ」欄に、**Stata** で使用するデフォルト作

業フォルダへのパスを設定できます。もちろん、**Stata** 起動後でも手動で変更でき、操作はメニューからファイル > 作業フォルダの変更... と行います。詳しくは [\[D\] cd](#) をご覧ください。

B.2 ショートカットを作成する

Windows のスタートメニューを使用せずに **Stata** を起動できるように、デスクトップにショートカットを作成できます。最も簡単な方法は **Stata** のショートカットをデスクトップにコピーすることです。ショートカットは **Stata** の実行ファイル (**exe** ファイル) からでも作成できます。手順は次の通りです。

1. **C:\Program Files\Stata18** フォルダ、または **Stata** をインストールしたフォルダを開きます。
2. フォルダ内からショートカットを作成する実行ファイル (**exe** ファイル) を探します。64 ビットバージョン

Stata のファイル名は次のようになっています。

64-bit	
Stata/MP:	StataMP-64.exe
Stata/SE:	StataSE-64.exe
Stata/BE:	StataBE-64.exe

右クリックをしてドラッグすれば、デスクトップ上に移動できます。

3. マウスボタンから指を離すと、コンテキストメニューを表示します。そのメニュー内のショートカットをここに作成を選びます。

これでショートカットは完成です。デスクトップ上ではなくフォルダ内にショートカットを作成したい時は、そのショートカットを入れたいフォルダ内にドラッグすれば作成できます。

このショートカットに関するプロパティも一般的な方法で設定できます。ショートカットを右クリックしてプロパティを選びます。必要に応じて、プロパティシートを前のセクション ([\[GSW\] B.1 The Windows Properties Sheet \(Windows プロパティシート\)](#)) にあるように編集してください。

B.3 Stata が起動する時に毎回コマンドを実行する

Stata は起動する時に自動的に **profile.do** ファイルを探します。目的のファイルが存在する場合、その中にあるコマンドを実行します。この時、まずはインストールしてあるフォルダ内を探します。それから、現在の作業フォルダ、パスの示すフォルダ、Windows の **USERPROFILE** の環境変数に定義されているホームディレクトリ、**ado-path** ([\[P\] sysdir](#) 参照) の順に検索します。Stata をインストールする時に指定する、デフォルトの作業フォルダ内に **profile.do** ファイルを入れる事をおすすめします。

異なるディレクトリから起動するショートカットを作成すると、デフォルトの作業フォルダ内の **profile.do** ファイルは実行されません。この機能から、異なる **profile.do** ファイルをプロジェクトごとに作成できます。

例えば、Stata を起動する度に日付の入ったログを取り始めるようにします。デフォルトの作業フォルダ、例えば **C:\Users\Stata\Documents\Stata** に **profile.do** を作成し、次のコマンドを入力します。

```
log using `': display %tCCYY-NN-DD-HH-MM-SS ///
    Clock("`c(current_date)' `c(current_time)'" , "DMYhms")', ///
    name(default_log_file)
```

Stata を起動すると通常の開始画面が開きます。同時に、次の追加コマンドを実行します。

```
Running C:\Users\Stata\Documents\Stata\profile.do ...
```

このコマンドはどのような意味なのでしょう。コマンドを一つずつ確認していきましょう。

- `c(current date)` と `c(current time)` はローカルシステムのマクロで、現在の日付と時間を含むものです。詳細は [\[P\] creturn](#) をご覧ください。
- ローカルマクロの両端にある左 (‘) と右 (’) の引用符は、これらを拡大します。詳細は [\[P\] macro](#) をご覧ください。
- `clock()` 関数は結果として表示する文字列や日付マスクである **"DMYhms"** を使用して、Stata が認識できる日付時間番号を作成します。詳しくは [\[D\] Datetime](#) をご覧ください。
- 「%tCCYY-NN-DD-HH-MM-SS」形式は Stata が算出した数字を「年-月-日-時-分-秒」の形式

で表示するように指定します。これはファイルをきれいにソートする際に便利です。詳しくは [\[D\]](#)

[Datetime display formats](#) をご覧ください。

- 見慣れない「: **display...**」は形式を指定した日付を直接ファイル名として使用するよう指定しま

す。これはインライン拡張のマクロ関数を使用する、上級者向けコンセプトです。詳細は [\[P\]](#)

[macro](#) を参照してください。

- 「**log using**」コマンドはログファイルを開始します。例題は [\[GSW\] 18 Learning more about Stata \(Stata についてもっと詳しく学ぶ\)](#) にありますので、参照してください。
- **name** オプションはログファイルに内部用の「**default_log_file**」を定義し、他のログファイルと名前が被らないようにします。詳細は [\[R\] log](#) をご覧ください。
- 最後に、**///**記号は連続性コメントです。これらの**3**つの行は**1**つのコマンドである事を示しています。コメントに関する詳細は [\[P\] comments](#) をご覧ください。

この**1**つのコマンドに **Stata** のプログラミングを行う上での上級コンセプトがたくさん含まれています。**profile.do** は実行された後には他の **do** ファイルと同じように扱えます。結果としては **Stata** を起動後に「**run**

profile.do」と実行しても同じです。**profile.do** が別にあるのは、わざわざ入力する事無くコマンドを自動で実行するためです。

システム管理者にとっては **sysprofile.do** も便利でしょう。このファイルは基本的に **profile.do** と同じですが、**Stata** は一番はじめに **sysprofile.do** の有無を検索します。この **sysprofile.do** ファイルがあると、**Stata** はファイル内のコマンドを実行します。その後に **profile.do** を探してファイルがある場合、目的のファイルの内容を実行します。

sysprofile.do が便利になるのは、例えば、システム管理者が **Stata** システムディレクトリのパスを変更した時です。このような場合、**sysprofile.do** に次のようなコマンドを組み込んで作成します。

```
sysdir set SITE "%Matador%StataFiles"
```

do ファイルの説明については [\[U\] 16 Dofiles](#) をご覧ください。**do** ファイルはシンプルな **ASCII** テキストファイルで、**Stata** が実行できるコマンドのシーケンスです。

B.4 Stata を開始する他の方法

Stata は初回起動時に保存データの拡張子を **Windows** のレジストリに登録します。その後、**.dta** のデータセット、**.do** の **do** ファイル、**.gph** のグラフファイルのいずれかをダブルクリックすると、新しい **Stata** のインスタンスが開いて目的のファイルを表示します。この時、ダブルクリックしたファイルの存在するフォルダがそのインスタンスの作業フォルダになります。**Stata** の動作は一般的な **Windows** アプリケーションと同じです。データセットをダブルクリックすると、**Stata** が起動し、データセットを開きます。グラフをダブルクリックすると、**Graph** ウィンドウでグラフが開き、**do** ファイルをダブルクリックすると、**do** ファイルエディタで **do** ファイルが開きます。**do** ファイルを直接実行するには **.do** のファイルを右クリックしてください。**Windows 11** では、まず「詳細オプションを表示」を選択し、それから実行アイテムを選ぶ必要があります。メニューアイテムである **Execute(do)** と **Execute Quietly (run)** がコンテキストメニューに組み込まれています。このアイテムのいずれかを選択すると、**Stata** は新たなインスタンスを開き、選択したアイテムを実行します。

新しいインスタンスを開かずに **do** ファイルを編集する、グラフを見る、データセットを開くには **Stata** のメインウィンドウ上にそのファイルをドラッグ&ドロップしてください。

B.5 Stata のバッチモード

大量の分析を行う時は **Stata** のバッチモードが便利です。目的に応じて複数の方法から操作を選ぶ事ができます。

方法 1

作業終了後にログファイルを特定の場所に保存するには、この方法を使います。

Windows 10, 11 では、タスクバーの検索ボックスに「**cmd**」と入力し、**Enter** を押します。

すると、コマンドプロンプトが開きます。**cd** コマンドを使用して、ログファイルの保存場所を現在の作業フォルダに変更します。例えば、**do** ファイル **bigjob** が **C:\Users\someone\statastuff** にあるとします。**do** ファ

イル実行後のログファイルの保存場所を `C:\Users\someone\statalogs` へ変更しましょう。次のように入力し、画面出力をせずにログファイルを目的の場所に保存します。

```
cd C:¥Users¥someone¥statalogs
"C:¥Program Files¥Stata18¥StataSE.exe" /e do "C:¥Users¥someone¥statastuff¥bigjob"
```

この時、**Stata** の実行ファイル (**exe** ファイル) の場所を指定しないとうまくいきません。

`/e` パラメータはバッチモード実行中の動作について指示します。使用可能なパラメータとそれぞれに対応する動作内容は次の通りです。

パラメータ	結果
<code>/b</code>	バックグラウンド (バッチ) モードを設定し、ログファイルを ASCII テキストで出力します。
<code>/e</code>	バックグラウンド (バッチ) モードに切り替えたうえで Stata コマンドが完了した時にその応答をポップアップ表示することなく、ASCII テキストであるログファイルに出力します。
<code>/q</code>	ロゴと初めのメッセージを抑制します。
<code>/rngstream#</code>	乱数生成器を <code>mt64s</code> に設定 (詳細は [R] set rng) し、乱数ストリームを <code>#</code> に設定 (詳細は [R] set rngstream) します。
<code>/s</code>	バックグラウンド (バッチ) モードを設定してログファイルを SMCL テキストで出力します。
<code>/i</code>	Windows タスクバーの Stata アイコンを抑制します。

方法 2

時間指定をする時や、時間を変えてバッチ作業を行う時は、**Windows** に標準的にインストールしてあるタスクスケジューラを使用します。

この方法は少々上級者向けです。また、各 **Windows** のバージョンにより、使用法が若干異なりますが、大まかな流れは次の通りです。

Windows 10, 11 では、タスクバーの検索ボックスからタスクスケジューラを検索します。タスクスケジューラを開き、基本タスクの作成をクリックします。そして基本タスクの作成ウィザードを設定し、**do** ファイ

ルをバッチモードで実行するように設定します。この時、**/b** または **/e** オプションを指定する必要があります。

「開始 (オプション)」 エリアにはログファイルを保存するフォルダのパスを入力します。この **do** ファイルを実行すると全ての出力は表示されず、指定したパスにログファイルを保存します。

全般的なメモ

do ファイル実行中は、**Stata** のアイコンがタスクバーに表示されます。

バッチ作業をキャンセルする事もでき、タスクバーにあるアイコンをクリックするとキャンセルの希望を尋ねるボックスを表示します。

タスクバーのアイコンはちかちかと点滅して作業の完了を知らせます。**do** ファイルが終了したら **Stata** を閉じてください。バッチ作業後に **Stata** を自動的に閉じるにはパラメータ **/b** ではなく **/e** を使用してください。

長い **do** ファイルの場合、必ずしもバッチモードで実行する必要はありません。バッチモードで実行できる **do** ファイルは通常通り、**Stata** のメインウィンドウからも実行できます。その場合、**Stata** を起動し、「log using ファイル名」を実行した後に「**do** ファイル名」を実行してください。あとは **do** ファイルが実行される様子を観察しても良いですし、**Stata** を最小化しても構いません。

B.6 Stata のセッションを複数同時に実行する

Stata のアイコンをダブルクリックまたは **Stata** をいずれかの方法で起動する度に、新しい **Stata** のインスタンスが呼び出されます。つまり、複数の **Stata** のセッションを同時に利用できます。新しい **Stata** のタイトルバーはそのインスタンスの番号を表示します。

B.7 Stata のロケールを変更する

Stata のロケールを英語に変更するには、次のようにします。

```
set locale_ui en
```

お使いの OS と同じロケールに戻すには、次のようにします。

```
set locale_ui default
```

ロケールと **Stata** の関係についての詳細は [\[U\] 12.4.2.4 Locales in Unicode](#) をご覧ください。


B.8 More

結果ウィンドウの出力をポーズを置いて表示するようにするには、「`set more on`」コマンドを実行します。ウィンド

ウに表示しきれない結果があるとき、結果ウィンドウ左下に `-more-` というプロンプトが表示されます。たとえば、多くのデータをリストする時に表示されます。

```
. list make mpg
```

	make	mpg
1.	AMC Concord	22
2.	AMC Pacer	17
3.	AMC Spirit	22
4.	Buick Century	20
5.	Buick Electra	15
6.	Buick LeSabre	18
7.	Buick Opel	26
8.	Buick Regal	20
9.	Buick Riviera	16
10.	Buick Skylark	19
11.	Cad. Deville	14
12.	Cad. Eldorado	14
13.	Cad. Seville	21
14.	Chev. Chevette	29
15.	Chev. Impala	16
16.	Chev. Malibu	22
17.	Chev. Monte Carlo	22
18.	Chev. Monza	24
19.	Chev. Nova	19
20.	Dodge Colt	30
21.	Dodge Diplomat	18
—more		

続きの画面を表示するには、**3**つの方法があります。まず、キーボードのキー（たとえばスペースキー）を押す方法です。次にメインウィンドウのツールバーにあるより多くの結果を表示ボタン をクリックする方法です。最後

に、結果ウィンドウの左下にある **—more—** リンクをクリックする方法です。結果ウィンドウの出力結果で次の行を **1** 行表示するには **Enter** キーを **1** 回押してください。より多くの結果を表示ボタン右端の矢印をクリックし、完了まで実行を選択すると最後まで表示が進みコマンド処理が完了します。

B.9 メモリサイズに関する考慮

Stata のメモリ管理は自動で行われます。少数の Stata ユーザが必要とする、メモリを効率化する小ワザに関しては [\[D\] memory](#) をご覧ください。

C Windows 版 Stata について—追加要素

C.1 Stata のデータセットやグラフを他のプラットフォームで使用する

Stata のデータセット (.dta) およびグラフファイル (.gph) は作成した OS が Mac または Unix でも、プラットフォーム

に依存することなく Windows 版で開く事ができます。また、Mac 版 Stata と Unix 版 Stata のユーザでも Windows 版で作成したファイルを使用できます。FTP を利用して Stata のファイルを送信する時は、ASCII 形式ではなくバイナリ形式を使用してください。

C.2 Stata のグラフを他のドキュメントにエクスポートする

Stata で作成したグラフをワープロソフトやプレゼンテーションソフトで使用する場合があります。グラフをエクスポートする方法は 2 通りあります。1 つはクリップボードを使用してグラフのコピーと貼り付けを行う方法です。もう 1 つは画像ファイル形式で保存し、そのグラフをアプリケーションにインポートする方法です。

C.2.1 クリップボードを使用してグラフをエクスポートする

Stata のグラフを最も簡単にエクスポートする方法はコピーと貼り付けです。

はじめに、グラフを作成するか、作成済みのグラフを再表示します。クリップボードにコピーするには Graph ウィンドウ内を右クリックし、コピーをクリックします。Stata は EMF (Enhanced Metafile) としてグラフのコピーを作成します。EMF を使うと、グラフの貼り付け先となるアプリケーションでは、できるだけ高い解像度で表示します。

メタファイルにはグラフを再作成する時に必要なコマンドが組み込まれています。つまり、メタファイルとは線、点、テキスト、色などの情報をまとめたものです。メタファイルはこれらの情報を構造化することにより、作図プログラムで編集する事ができます。クリップボードにグラフをコピーした後、グラフをインポートするアプリケーションに切り替えてから貼り付けましょう。一般的にこの操作を行うにはメニューから編集 > 貼り付けと操作します。詳しくは各アプリケーションのマニュアル等を参照してください。

C.2.2 グラフを他のファイルでエクスポートする

Stata はグラフを複数の画像ファイル形式でエクスポートできます。グラフを右クリックした後に名前を付けて保

存... を選び、「ファイルの種類」をクリックすると **Stata** は 拡張メタファイル (**EMF, .emf**)、**TIFF** プレビュー付きまたはなしの **Enhanced PostScript (EPS, .eps)**、高品質または最高品質の **Joint Photographic Experts Group (JPEG, .jpg)**、**Portable Document Format (PDF, .pdf)**、**Portable Network Graphics (PNG, .png)**、**PostScript (PS, .ps)**、**Scalable Vector Graphics (SVG, .svg)**、**Tag Image File Format (TIFF, .tif)**の形式で保存できることが確認できます。

EMF、**EPS**、**PDF**、**PS**、**SVG** はベクター形式で、**JPEG**、**PNG**、**TIFF** はビットマップ形式です。グラフのサムネイルを使う時は、**EPS with TIFF preview** を選択してください。このプレビューオプションを選択してもグラフ印刷

に影響はありません。**PNG** はウェブサイト等にグラフを掲載するのに適しています。詳しくは[\[G-2\] graph export](#) をご覧ください。

C.3 ネットワークドライブで **Stata** をインストールする

まず、ネットワークドライブに **Stata** をインストールする前に「同時起動ライセンス」を入手してください。**Stata** をサ

ーバからインストールします。ネットワーク管理者の権限がある時は、**Stata** を直接ネットワークドライブにインストールできます。

ライセンスを初期化するために、**Stata** をインストールしてから、一度実行してください。そして、ワークステーションから **Stata** をインストールしたネットワークドライブが使用できる状態にします。デスクトップ上で右クリックをするか、**Windows** のスタートメニューを開き、新規作成 > ショートカットを選択します。**Stata** 実行ファイルへのパスを編集領域に入力するか、参照してパスを指定してください。ショートカットの名前として **Stata** と入力しましょう。

Stata へのショートカットを作成した後に、右クリックしてプロパティを選択します。**Stata** のデフォルト作業フォルダを設定するには、「作業フォルダ」エリアのパスをユーザがアクセスできるローカルドライブに変更します。ここで指定したドライブにデータセットやグラフ等 **Stata** に関連するファイルを保存します。ワ

ークステーションを 1 人以上のユーザが使用する場合、「作業フォルダ」エリアを、
「%HOMEDRIVE%%HOMEPATH%」という環境変数に変更すると、デフォルトの作業フォルダを各ユーザのホームディレクトリにできます。

C.4 Stata を Python から呼び出す

pystata Python パッケージを使用して、**Stata** を **Python** から呼び出すことができます。これには、**Stata** および **Mata** との対話に使用できる一連の **API** 関数と **IPython magic** コマンドが含まれます。**pystata Python** パッケージの詳細は、オンラインドキュメンテーション (<https://www.stata.com/python/pystata>) または **[P] Pystata module** をご覧ください。

C.5 Stata のライセンスを変更する

すでに **Stata** はインストール済みでライセンスを変更する必要がある時は、**Stata** のメニューから ヘルプ > **Stata** について を選択し、開いた画面左下の「ライセンスの更新...」をクリックします。更新オプションを確認するか、新しいライセンスを入力するかを選びます。