

# Pelatihan Komputasi dengan Stata

## Modul A - Manajemen Data dan Analisis Deskriptif

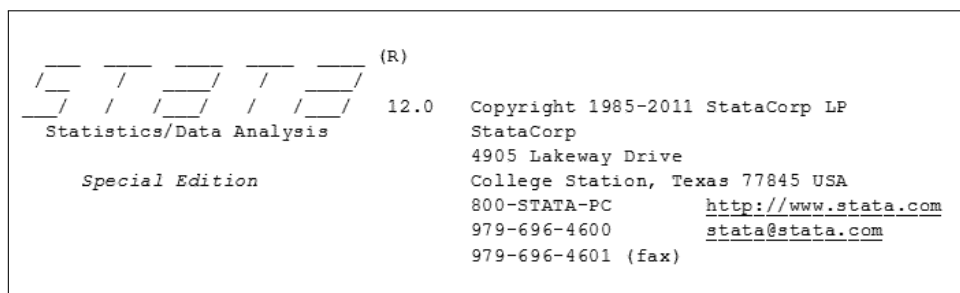
Benedict J. Yappy<sup>1</sup>      Jazman Ihsanuddin

Departemen Ilmu Ekonomi  
Universitas Indonesia

<sup>1</sup> [yappy.benedict@gmail.com](mailto:yappy.benedict@gmail.com) – penyusun.<sup>‡</sup>

## 1 Pendahuluan

Dokumen ini adalah bahan yang digunakan dalam pelatihan “Modul A - Manajemen Data dan Analisis Deskriptif”. Modul ini mencakup deskripsi ringkas fungsi yang akan digunakan dalam *software*, perintah yang digunakan, serta contoh hasil yang diperoleh (bilamana perlu). Dalam paragraf uraian, perintah/*command* Stata dapat dikenali dengan **format** ini, dengan ringkasan perintah baru ditulis di sebelah kanan dan bagian perintah yang diubah sesuai penggunaan dicetak miring. Contoh output Stata dapat dikenali seperti *startscreen* Stata berikut:



```
(R)
-----
Statistics/Data Analysis 12.0 Copyright 1985-2011 StataCorp LP
Special Edition          StataCorp
                        4905 Lakeway Drive
                        College Station, Texas 77845 USA
                        800-STATA-PC http://www.stata.com
                        979-696-4600 stata@stata.com
                        979-696-4601 (fax)
```

Bagi pengguna, perintah `help` berguna untuk melihat bagaimana suatu *command* digunakan, dan opsi perintah apa saja yang tersedia, terutama bila fungsi yang ingin digunakan sudah diketahui. Caranya adalah mengetik `help command` (*yep, it includes help help!*). Informasi lebih lengkap terkait perintah yang diperkenalkan dalam modul dapat diakses dengan perintah ini.

help

Modul ini mencakup seluruh silabus pelatihan yang hendak disampaikan. Dengan kata lain, pembaca dapat menguasai materi yang disampaikan dalam pelatihan dengan membaca dan mereplikasi langkah yang ada dalam modul ini. Pembaca dipersilahkan mengakses materi di <http://benconomy.wordpress.com/tutoring>, tanpa mengubah isinya.

<sup>‡</sup>Powered by L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> with X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X using T<sub>E</sub>Xstudio and Stata.

# Daftar Isi

<b>1</b>	<b>Pendahuluan</b>	<b>1</b>
<b>2</b>	<b>Penggunaan Stata</b>	<b>3</b>
2.1	Bekerja dengan Stata . . . . .	3
2.2	<i>Working Directory</i> . . . . .	3
2.3	Sumber Data . . . . .	4
2.4	Menyimpan Dataset, Perintah dan Output Stata . . . . .	4
<b>3</b>	<b>Bekerja dengan Data</b>	<b>5</b>
3.1	Manajemen variabel . . . . .	6
3.2	Perintah <code>generate</code> . . . . .	6
3.3	Perintah <code>drop</code> . . . . .	6
3.4	Struktur Data . . . . .	6
3.5	Bekerja dengan <code>timevar</code> . . . . .	7
3.6	Peubah untuk <code>varlist</code> . . . . .	8
3.7	Label . . . . .	9
3.8	Perintah <code>encode</code> dan <code>recode</code> . . . . .	10
3.9	Perintah <code>reshape</code> . . . . .	11
3.10	Operator <code>if</code> , <code>in</code> dan <code>by</code> . . . . .	13
3.11	Perintah <code>count</code> dan <code>list</code> . . . . .	13
<b>4</b>	<b>Statistik Deskriptif</b>	<b>14</b>
4.1	Tabel statistik deskriptif . . . . .	14
4.2	Grafik statistik deskriptif . . . . .	15

## 2 Penggunaan Stata

### 2.1 Bekerja dengan Stata

Stata adalah sebuah program statistik yang dapat dikatakan sangat lengkap dari segi kemampuan fungsi statistik yang dapat ditanganinya, yang menjadikan Stata salah satu program paling populer yang digunakan oleh peneliti-peneliti dari berbagai kalangan. Secara garis besar, cara bekerja dalam Stata adalah melalui command line, di mana perintah diketikkan ke dalam panel **command** (dapat diaktifkan dengan *shortcut* Ctrl+1). Walau demikian, Stata memiliki cukup banyak *dialog box* untuk banyak perintahnya, yang menambah fleksibilitas program Stata.

Bila berada dalam panel command, kita dapat scroll command-command sebelumnya dengan Page-up/down, untuk memudahkan mengulangi perintah yang pernah dipanggil. Opsi untuk masing-masing perintah biasanya ditambahkan setelah tanda koma , (hanya ada satu tanda koma diperbolehkan untuk setiap perintah).

Selain panel Command, terdapat juga beberapa panel lain:

1. Results (Ctrl+2) yang melaporkan hasil dari perintah yang dijalankan.
2. Review (Ctrl+3) yang mendaftarkan perintah-perintah yang telah dijalankan sebelumnya.
3. Variables (Ctrl+4) yang mendaftarkan variabel-variabel yang sedang ada dalam dataset/memory.
4. Properties (Ctrl+5) yang memberi keterangan tentang variabel yang dipilih.

### 2.2 Working Directory

Di sebelah kiri bawah tampilan window Stata, akan dituliskan *working directory* Stata. Working directory adalah folder di mana Stata “bekerja”, yaitu tempat di mana Stata akan menyimpan atau mengakses file secara default. Ada baiknya working directory di set ke directory di mana dataset dan file lain terkait pekerjaan tersebut disimpan. Seperti di banyak program lain, untuk mengubah working directory, ketikkan `cd new_path` ke dalam panel command. File yang ada dalam working directory dapat langsung diakses menggunakan *file\_name* (contohnya dataset.dta), sedangkan yang berada di luar working directory hanya dapat diakses menggunakan file path lengkap (contohnya "D:\Data\Dataset.dta"). Dengan kata lain, memindahkan file-file yang digunakan dalam Stata ke dalam working directory akan membantu untuk mengakses file-file tersebut nantinya.

Penggunaan nama file dalam Stata perlu menggunakan tanda petik "...” untuk nama file yang mengandung karakter spasi.

## 2.3 Sumber Data

Data yang akan dianalisis menggunakan Stata dapat berasal dari tiga sumber:

1. File database Stata yang telah disimpan sebelumnya, yang memiliki extension `.dta`. Cara membuka dapat dengan menggunakan command line `use file_name.dta` dialog box (File > Open ; Ctrl+O). use

2. Copy-paste dari program database lain, seperti Microsoft Excel. Pertama, copy data dari program lain tersebut. Kembali di Stata, ketik `edit` di panel Command untuk membuka data editor.. Data dapat langsung di-*paste* (Ctrl+V) ke Stata. Untuk menampilkan dataset tanpa melakukan edit (dan mencegah edit yang tidak disengaja), gunakan perintah `browse`. edit  
browse

Bila data yang di-copy termasuk nama variabel di baris pertama, pilih “**Treat first row as variable names**”. Bila hanya data yang di-copy, pilih “**Treat first row as data**” — Stata akan memberi nama untuk masing-masing variabel (`var1`, `var2`, `dst`). Data yang di-copy dapat berupa data dengan banyak variabel sekaligus (multi-column) maupun satu per satu — variabel baru dapat ditambahkan melalui copy-paste.

3. Membuka file database yang dibuat di program lain. Cara paling mudah melakukan impor database adalah dengan menggunakan opsi di dalam File > Import.

Selain menggunakan database yang telah ada seperti dalam ketiga sumber di atas, Stata juga menyediakan file-file contoh database `.dta` yang siap digunakan dalam setiap instalasi Stata. Untuk melihat daftar file tersebut, ketik `sysuse dir` dalam panel command, dimana file yang ingin dibuka dapat digunakan dengan perintah `sysuse file_name`. sysuse

Stata juga dapat membuka file `.dta` di internet secara langsung dengan menggunakan perintah `webuse URL`. webuse

## 2.4 Menyimpan Dataset, Perintah dan Output Stata

**Dataset** Hasil data yang sudah diolah dengan Stata dapat disimpan dengan perintah `save file_name`. Bila nama file dataset yang hendak disimpan sudah ada, kita perlu menambahkan opsi `replace` ke dalam perintah (`save file_name, replace`). Fungsi Save dan Save As seperti di program lain dapat diakses dengan shortcut Ctrl+S dan Ctrl+Shift+S. save

**Perintah dan *do-file*** “History” perintah-perintah yang dijalankan dalam satu sesi Stata akan dimasukkan dalam panel Review. Kita dapat menyeleksi perintah

yang kita inginkan dari panel Review dan mengeditnya dalam *do-file editor* di Stata.

**Do-file** adalah file text standar dengan extension `.do` (dapat dibuka/diedit dalam text editor seperti Notepad). File `.do` ini menjadi daftar perintah yang dapat dijalankan atau diedit oleh Stata. Untuk menjalankan do-file yang sudah disimpan, gunakan perintah `do file_name` untuk menjalankan do-file seperti biasa, dan perintah `run file_name` untuk menjalankan tanpa laporan dalam panel Results. Untuk mengedit do-file yang telah disimpan sebelumnya, gunakan perintah `doedit file_name`.

do | run

doedit

Menyimpan daftar perintah yang dijalankan oleh Stata sangat penting untuk memastikan bahwa perintah yang diberikan tercatat, sehingga dapat dipertanggungjawabkan dan direplikasi. Penyimpanan daftar perintah tersebut dengan mudah dilakukan dengan **do-file** Stata. Selain itu, eksekusi `do` maupun `run` akan berhenti bila terdapat kesalahan dalam data/command, sehingga sangat berguna untuk melakukan hal-hal yang sifatnya berulang.

**Output (Results)** dalam panel Results dapat disimpan dengan memulai log-file `.smcl`. Detail perintah untuk manajemen log-file sebagai berikut:

log

- `log using file_name`: Menyimpan log ke dalam file dengan menggunakan `file_name.smcl`. Log ini dapat dibuka oleh Stata.
- `log off`: Menghentikan penyimpanan log secara sementara.
- `log on`: Melanjutkan kembali penyimpanan log yang sudah di *off* sebelumnya.
- `log close`: Menyelesaikan penyimpanan log.

Selain dengan menyimpan log, output stata dapat dicopy-paste ke program lain. Untuk meng-copy, highlight area output yang akan di-copy, dan klik kanan. Copy biasa (Ctrl+C) akan menghasilkan teks, copy as table akan menghasilkan tabel dengan karakter tabulator (untuk program spreadsheet seperti Excel), dan copy picture menghasilkan gambar dan mempertahankan tampilan dalam Stata.

### 3 Bekerja dengan Data

Sebagai program statistik, Stata memiliki kemampuan operasi variabel yang cukup luas dan menyeluruh. Pada bagian ini diuraikan bagaimana cara bekerja dengan variabel yang paling umum dipakai. Opsi yang tersedia di Stata lebih luas dari uraian dalam modul ini, sehingga dianjurkan untuk melihat `help` untuk masing-masing perintah.

### 3.1 Manajemen variabel

**Jenis Variabel** Pada dasarnya terdapat dua jenis variabel di Stata, yaitu `string` (dapat bernilai huruf) dan variabel numerik. Data `string` tidak dapat digunakan dalam perintah perhitungan/statistik dalam Stata. Terdapat beberapa jenis variabel numerik.\* Untuk mengubah format penyimpanan variabel, gunakan perintah `recast type varlist` dimana `type` adalah tipe variabel. Variabel `string` tidak dapat diubah menjadi angka. Untuk melihat daftar dan jenis variabel dalam dataset yang dibuka, gunakan perintah `describe`.

recast

describe

**Order** Urutan variabel yang ada dalam Stata dapat diubah dengan menggunakan perintah `order varlist`, di mana variabel `varlist` tersebut akan dipindahkan menjadi variabel pertama. Tambahkan opsi `last` untuk memindahkan ke variabel terakhir; `before(varname)` dan `after(varname)` untuk memindahkan sebelum atau setelah `varname`.

order

### 3.2 Perintah generate

Seperti namanya, perintah `generate varname=expression` berguna untuk menciptakan variabel baru. `Expression` yang dimaksud dapat berupa operasi hitung, variabel lain ataupun pembuatan angka random oleh Stata. Operasi yang didukung dapat dilihat dalam menu `help functions`. Beberapa ekspresi yang umum digunakan yaitu `+`, `-`, `/`, `*`, `exp()` dan `ln()`. Bila variabel yang diciptakan digunakan untuk menggantikan (overwrite) variabel yang sudah ada, maka perintah `replace` dapat digunakan untuk menggantikan `generate`.

generate

replace

Selain perintah `generate`, Stata juga memiliki perintah `egen`. Perintah `egen` dapat melakukan semua operasi yang dapat dilakukan oleh `generate`, ditambah beberapa fungsi lainnya, seperti mengisi dengan pola, statistik variabel, dan lainnya (lihat di `help egen`).

egen

### 3.3 Perintah drop

Perintah `drop varlist` berguna untuk membuang variabel-variabel (`varlist`) yang diinginkan. ‘Lawan’ dari perintah `drop` adalah `keep varlist`, di mana variabel-variabel lain selain `varlist` akan dibuang.

drop

keep

### 3.4 Struktur Data

Terdapat tiga jenis cara penanganan dataset (umum) oleh Stata:

\*Perbedaan jenis variabel dapat mempengaruhi derajat keakurasian, namun berguna untuk menghemat tempat penyimpanan dan waktu kalkulasi (dibahas dalam `help data_type`).

1. Cross section. Struktur data ini adalah struktur data yang dasar di Stata. Tidak ada unsur waktu dalam variabel ini. Untuk kembali ke struktur data Cross Section, setting `tsset` atau `xtreg` perlu di-clear dengan menambahkan opsi `clear` ke perintah `tsset` atau `xtset`.
2. Time Series. Struktur data time series mengikuti satu sampel dalam beberapa periode waktu. Untuk menyatakan struktur data sebagai time series, digunakan perintah `tsset timevar, unitoptions`. `timevar` di Stata bernilai 0 untuk (awal) tahun 1960, dan dihitung ke depan atau belakang sebesar `unitoptions` (`clocktime`, `daily`, `weekly`, `monthly`, `quarterly`, `halfyearly`, `yearly`, `generic`). Bila perubahan tidak berjalan sebesar 1 unit, tambahkan opsi `delta(#)`. `tsset`
3. Panel. Struktur data panel mengikuti lebih dari satu sampel selama lebih dari satu periode. Untuk menyatakan data sebagai data panel, gunakan perintah `xtset panelvar timevar, unitoptions`. `panelvar` adalah ID untuk masing-masing unit sampel yang diobservasi. `xtset`

### 3.5 Bekerja dengan `timevar`

Seperti yang ditulis di atas, `timevar` yang digunakan untuk identifikasi waktu harus berada dalam format numerik, dengan spesifikasi SIF (*Stata internal form*) berupa berapa unit setelah (sebelum) tahun 1 Januari 1960 00:00.00. Format ini tidak selalu mudah untuk diperoleh, karena data yang diperoleh berada dalam format HRF (*human readable form*). Untuk mengubahnya ke dalam format SIF yang dapat di-`tsset`, kita dapat menggunakan perhitungan manual untuk selisih dari awal tahun 1960 tersebut, namun cara ini dapat dikatakan tidak praktis. Stata menyediakan konversi antar SIF dan HRF sebagai berikut.

1. Pastikan variabel waktu HRF (`timevar_str`) telah memiliki format penamaan dan frekuensi yang konsisten. Contohnya, untuk data kuartal, data harus konsisten suati format (seperti 1960Q1, 1960Q2 dst dan tidak berubah menjadi 1970M3, 1970M6 dst).
2. Pelajari format data waktu untuk membentuk *mask* yang akan dikenali Stata. Stata akan menggunakan tanda baca (`.`, `-`, `_`) untuk memisahkan antar komponen waktu, Stata tidak mengenali elemen lain selain waktu (unsur tersebut tidak diikuti dalam mask dengan karakter `#`). Beberapa contoh untuk tanggal 16 Oktober 1992:
  - Oct 16, 1992 akan memiliki mask `MDY`
  - Friday, Oct 16, 1992 akan memiliki mask `#MDY`

- 16 Oct 1992 akan memiliki mask `DMY`
  - 161092 akan memiliki mask `DM19Y` (19Y untuk 2 digit pertama angka tahun, untuk format tahun 2 digit)
  - 1992W42 akan memiliki mask `YW` (mingguan)
  - 1992M10 akan memiliki mask `YM` (bulanan)
  - 1992Q3 akan memiliki mask `YQ` (kuartalan)
  - 1992S2 akan memiliki mask `YH` (semesteran)
3. Ciptakan variabel baru `timevar` untuk konversi ke dalam bentuk SIF. Format perintahnya adalah `generate timevar = function(timevar_str,mask)`. Variabel baru `timevar` sudah dapat digunakan untuk `tsset`. Supaya variabel waktu tersebut tampil dalam format yang dapat dibaca, perlu dilakukan format dengan perintah `format timevar %fmt`. Aplikasinya:

format

```
generate timevar = date(timevar_str, "DMY"), format timevar %td
dan tsset timevar, daily) untuk data harian
```

```
generate timevar = weekly(timevar_str, "YW"), format timevar %tw
dan tsset timevar, weekly) untuk data mingguan
```

```
generate timevar = monthly(timevar_str, "YM"), format timevar %tm
dan tsset timevar, monthly) untuk data bulanan
```

```
generate timevar = quarterly(timevar_str, "YQ"), format timevar
%tq dan tsset timevar, quarterly) untuk data kuartalan
```

```
generate timevar = halfyearly(timevar_str, "YH"), format timevar
%th dan tsset timevar, halfyearly) untuk data semesteran
```

### 3.6 Peubah untuk `varlist`

**Wildcards** Sebagian besar perintah Stata lainnya juga dapat menggunakan wildcard untuk `varlist`. Beberapa wildcard tersebut:

- `*` — digunakan sebagai substitusi satu atau lebih karakter. Contohnya, `Ind*` akan masuk untuk Indonesia dan India; `*Inc` akan masuk untuk Stata Inc dan Microsoft Inc.
- `?` — digunakan untuk substitusi satu karakter. Contohnya, `p?p?` akan masuk untuk papa maupun popo.



- `~` — digunakan sebagai substitusi di tengah. Contohnya, `pan~an` akan masuk untuk panduan dan panutan.
- `_all` — digunakan sebagai substitusi untuk semua variabel.

**Operator** Terdapat dua jenis operator dalam operasi variabel Stata, yaitu operator jenis data dan operator waktu (kedua jenis operator ini dapat dikombinasikan). Operator-operator ini ditambahkan dengan tanda titik sebelum nama variabel (contohnya, `Li.varname`). Berikut adalah uraian operator tersebut.

- `i` — Operator ini menyatakan bahwa variabel diperlakukan sebagai variabel faktor/kualitatif. Penggunaan operator ini memudahkan pekerjaan, karena tidak perlu lagi menciptakan *dummy* untuk variabel-variabel kualitatif.
- `c` — Operator ini menyatakan bahwa variabel diperlakukan sebagai variabel *continuous*, yaitu berisi nilai.
- `#` — Operator ini adalah perintah untuk membuat interaksi antar variabel. Contohnya, `var1#var2`.
- `L` — Operator ini berguna untuk mengambil mengambil nilai sebelumnya (*lag*) dari variabel. Lag dapat dispesifikasikan dengan angka (`L3`) ataupun kombinasi huruf (`LLL`).
- `F` — Operator ini berguna untuk mengambil mengambil nilai berikutnya (*forward*) dari variabel. Forward dapat dispesifikasikan dengan angka (`F3`) ataupun kombinasi huruf (`FFF`).
- `D` — Operator ini berguna untuk perubahan nilai (*difference*) dari variabel. Difference orde lebih tinggi dapat dispesifikasikan dengan angka (`D2`) ataupun kombinasi huruf (`DD`).

### 3.7 Label

Label dalam Stata berguna untuk membantu pengguna. Contohnya, nama variabel dalam Stata tidak dapat mengandung karakter spasi. Berikut ini dibahas penggunaan `label` yang ada dalam stata.

label

`label data "label"` memberikan keterangan label untuk keseluruhan dataset.

`label variable varname "label"` memberikan keterangan label untuk suatu variabel, yang muncul dalam panel variabel maupun perintah `describe`.

`label define labelname # "label"` mendefinisikan sebuah label untuk memberi label keterangan untuk masing-masing nilai angka (#), yang dapat digunakan untuk variabel numerik. Fitur ini berguna terutama untuk variabel kualitatif. Satu angka # mendapat satu label, sehingga ulangi bagian # "label" untuk masing-masing label dalam sate set labelname.

Tambahkan salah satu dari opsi `add` (hanya menambah label baru), `modify` (mengubah label yang sudah ada), dan `replace` (mendefinisikan ulang labelname) bila labelname sudah ada sebelumnya.

`label values varlist "labelname"` meng-assign sate set value label (*labelname*) yang sudah ada (di-define) ke suatu variabel (*varlist*).

`label dir` memperlihatkan daftar value label yang ada.

`label list` memperlihatkan daftar value label beserta nilai-nilai dan label keterangan masing-masing.

### 3.8 Perintah `encode` dan `recode`

Stata tidak dapat melakukan perhitungan statistik untuk variabel dengan format string, sehingga variabel kualitatif (seperti nama negara) perlu dikonversi menjadi numerik (diberi nomor ID). Untuk melakukannya, terdapat command `encode varname, generate newvar`. Selain menciptakan angka-angka untuk masing-masing string, perintah `encode` juga membuat dan meng-assign value label dengan labelname yang sama dengan `newvar`. Untuk mengubah angka yang sudah ada, digunakan perintah `recode varname, generate newvar`. Ilustrasi dari perintah-perintah tersebut adalah sebagai berikut:

negara_str		negara_num		benua	
Data teks		No	Label	No	Label
Indonesia		1	Indonesia	1	Asia
China		2	China	1	Asia
Jepang	<code>encode</code>	3	Jepang	1	Asia
Korea		4	Korea	1	Asia
Inggris		5	Inggris	2	Eropa
Jerman		6	Jerman	2	Eropa
Prancis		7	Prancis	2	Eropa
Amerika Serikat		8	Amerika Serikat	3	Amerika

Untuk transformasi di atas, kedua perintah yang digunakan adalah:

```
encode negara_str, generate(negara_num)
```

```
recode negara_num (1 2 3 4 = 1 "Asia") (5/7 = 2 "Eropa")
(8 = 1 "Amerika"), generate (benua)
```

### 3.9 Perintah **reshape**

Dataset yang diperoleh untuk analisis dapat berbentuk “long” ataupun “wide”. Untuk mengilustrasikannya, berikut adalah beberapa bentuk yang berbeda untuk dataset yang sama.

country	time	variable	data
China	1997	gdp	7.90E+12
China	1997	investment	2.60E+12
China	1997	savings	3.30E+12
China	1998	gdp	8.40E+12
China	1998	investment	2.90E+12
China	1998	savings	3.40E+12
China	1999	gdp	9.00E+12
China	1999	investment	3.10E+12
China	1999	savings	3.40E+12
Indonesia	1997	gdp	6.30E+14
Indonesia	1997	investment	1.80E+14
Indonesia	1997	savings	1.80E+14
Indonesia	1998	gdp	9.60E+14
Indonesia	1998	investment	2.40E+14
Indonesia	1998	savings	2.10E+14
Indonesia	1999	gdp	1.10E+15
Indonesia	1999	investment	2.20E+14
Indonesia	1999	savings	1.50E+14
UK	1997	gdp	8.40E+11
UK	1997	investment	1.40E+11
UK	1997	savings	1.40E+11
UK	1998	gdp	8.80E+11
UK	1998	investment	1.60E+11
UK	1998	savings	1.50E+11
UK	1999	gdp	9.30E+11
UK	1999	investment	1.60E+11
UK	1999	savings	1.40E+11

Country, time, variable (long)

Konversi antara berbagai bentuk/dimensi (termasuk ketiga tabel dalam halaman ini) dapat dilakukan dengan command **reshape** dari Stata, yang akan dibahas di halaman berikut ini.

[benconomy.wordpress.com](http://benconomy.wordpress.com)

country	variable	data97	data98	data99
China	gdp	7.9e+12	8.4e+12	9.0e+12
China	investment	2.6e+12	2.9e+12	3.1e+12
China	savings	3.3e+12	3.4e+12	3.4e+12
Indonesia	gdp	6.3e+14	9.6e+14	1.1e+15
Indonesia	investment	1.8e+14	2.4e+14	2.2e+14
Indonesia	savings	1.8e+14	2.1e+14	1.5e+14
UK	gdp	8.4e+11	8.8e+11	9.3e+11
UK	investment	1.4e+11	1.6e+11	1.6e+11
UK	savings	1.4e+11	1.5e+11	1.4e+11

Country, variable (long), time (wide)

country	time	gdp	investment	savings
China	1997	7.9e+12	2.6e+12	3.3e+12
China	1998	8.4e+12	2.9e+12	3.4e+12
China	1999	9.0e+12	3.1e+12	3.4e+12
Indonesia	1997	6.3e+14	1.8e+14	1.8e+14
Indonesia	1998	9.6e+14	2.4e+14	2.1e+14
Indonesia	1999	1.1e+15	2.2e+14	1.5e+14
UK	1997	8.4e+11	1.4e+11	1.4e+11
UK	1998	8.8e+11	1.6e+11	1.5e+11
UK	1999	9.3e+11	1.6e+11	1.4e+11

Country, time (long), variable (wide)

Sebelum melakukan perintah `reshape`, beberapa hal perlu diketahui terlebih dahulu. Pertama, dimensi sekarang dari dataset (apakah wide atau long). Kedua bagaimana dataset akan dibentuk (dimensi wide dan long yang diinginkan). Ketiga, memastikan bahwa dataset memenuhi syarat untuk dilakukan `reshape`: (1) Terdapat `stubnames` yang menghubungkan data-data wide `j`, yang dalam contoh di atas adalah awalan data (`data97`, `data98`, `dst`). (2) Tidak terdapat lebih dari satu nilai untuk satu kolom  $x_{ij}$ . Sintaks dan penjelasan perintah `reshape` adalah sebagai berikut:

`reshape wide/long stubname, i(i_var) j(j_var) string`

reshape

- `wide|long` — Bentuk yang dituju setelah melakukan perintah, apakah wide atau long. Perlu diketahui bahwa dapat dilakukan beberapa perintah `reshape`, untuk menyesuaikan dengan struktur data.  
Dalam contoh tadi, `reshape wide` dapat dilakukan sekali lagi untuk membuat `country`, `variable` dan/atau `time` semuanya sebagai variabel wide (`j`).
- `stubnames` — Nama variabel (untuk `reshape long`) atau bagian umum dari nama variabel (`reshape wide`). Penambahan memberikan informasi di mana nama variabel baru ditempatkan.  
Contoh dalam dataset tadi, (`data`) dapat digunakan untuk `data97`, `data 98 dst`. `Reshape` tidak dapat digunakan tanpa `stubnames`, contohnya untuk variabel `gdp`, `investment dst` (`rename` menjadi `datagdp`, `datainvestment`).
- `i(i_var)` — Variabel-variabel yang berada dalam bentuk long, yang mengidentifikasi observasi secara unik.
- `j(j_var)` — Variabel lama yang akan di-expand (`reshape wide`), atau variabel baru yang akan diciptakan sebagai identifier (`reshape long`).  
Berikut opsi `j` untuk contoh tadi. Untuk `reshape long`, berikan nama variabel baru (`country`, `time`, atau `variable`). Untuk `reshape wide`, berikan nama variabel lama yang akan di expand (`country`, `time`, atau `variable`).
- `string` — Tambahkan opsi `string` bila `j_var` berisi huruf.  
Dalam contoh tadi, tambahkan opsi `string` bila `j_var` adalah `country` atau `variable` yang adalah data `string`. Jangan tambahkan opsi `string` bila `j_var` adalah `time` yang adalah data numerik. Perhatikan, untuk `reshape wide`, isi variabel yang akan dikonversi menjadi long adalah karakter huruf atau `_` (tidak boleh mengandung karakter non-huruf seperti spasi dan titik).

Perubahan dataset oleh perintah `reshape` dapat di “undo”: Setelah `reshape long`, perintah `reshape wide` (tanpa argumen/opsi) akan mengembalikan dataset seperti sebelum `reshape long` tadi dilakukan; demikian juga sebaliknya. Periksa kembali dataset setelah `reshape` untuk memastikan dataset seperti yang diinginkan, sebelum melakukan kalkulasi statistik atau `save` dataset.

### 3.10 Operator `if`, `in` dan `by`

**Akhiran** `if expression` berguna untuk membuat Stata melakukan perintah hanya pada observasi yang memenuhi kriteria (`expression`) yang dibuat dalam `if`. Berbagai operator dapat digunakan untuk `expression`, seperti `>`, `>=`, `<`, `<=` dan `==` (gunakan dua (*bukan satu!*) tanda sama dengan). `if` dapat digunakan pada sebagian besar perintah Stata. Tambahkan `if expression` pada *akhir* perintah yang diinginkan.

if

**Akhiran** `in obsno` — Berguna untuk membuat Stata melakukan perintah hanya pada nomor observasi yang dispesifikasikan (`obsno`). Nomor observasi 1 2 3 4 juga dapat dinyatakan dengan 1/4 dan 1 2/4. `in` dapat digunakan pada sebagian besar perintah Stata. Tambahkan `in obsno` pada *akhir* perintah yang diinginkan.

if

**Awalan** `by (varlist)`: — Berguna untuk membuat Stata melakukan perintah secara berkelompok (berulang) dengan variabel dalam `varlist`. `by` memerlukan dataset untuk diurutkan (`sort`) sesuai dengan `varlist` (`sort varlist`). Tambahkan `by(varlist)`: *sebelum* mengetikkan perintah yang diinginkan.

by

sort

Awalan `bysort` digunakan dengan cara yang sama dengan awalan `by`. Perbedaannya, `bysort` termasuk perintah `sort`, sehingga dataset akan diurutkan sesuai dengan `varlist` terlebih dahulu.

bysort

### 3.11 Perintah `count` dan `list`

Perintah `count` berguna untuk menghitung dan menampilkan jumlah observasi yang memenuhi kriteria. Kriteria yang dimaksud dapat dispesifikasikan dengan operator `if`, `in` ataupun `by`. Eksekusi `count` sangat cepat dibandingkan dengan perintah statistik lainnya, sehingga perintah ini biasanya digunakan untuk mengecek suatu dataset (terutama dengan menggunakan `do-file`).

count

Perintah `list varlist` berguna untuk menampilkan nama variabel dalam `varlist` dan nilai-nilainya. Perintah `list` sering digunakan dengan operator `if`, `in` atau `by`. Sama seperti `count`, eksekusi perintah `list` sangat cepat dan biasanya digunakan untuk mengecek suatu dataset. Di samping itu, penggunaan `list` biasanya dapat mende-  
teksi kesalahan data secara lebih mudah dibandingkan `edit` ataupun `browse`.

list

## 4 Statistik Deskriptif

### 4.1 Tabel statistik deskriptif

Tabel statistik deskriptif membantu menampilkan isi dari variabel secara ringkas dengan berbagai statistik. Semua tabel statistik berikut dapat digunakan dengan opsi `if`, `in` maupun `by`.

Perintah `summarize varlist` berguna untuk menampilkan statistik deskriptif seperti mean (rata-rata), median, dan data terkecil (min) dan terbesar (max) untuk suatu variabel secara keseluruhan. Bila tidak terdapat `varlist`, perintah ini akan menampilkan statistik deskriptif untuk seluruh variabel. Tambahkan opsi `detail` untuk mendapatkan statistik deskriptif yang lebih detail.

summarize

Perintah `tabulate varname1 varname2` berguna untuk membuat tabulasi (perhitungan seberapa banyak suatu variabel muncul) dalam satu atau lebih variabel. Bila hanya satu variabel yang ditabulasi, maka cukup menggunakan `varname1`. Tambahkan `varname2` bila menggunakan tabulasi dua variabel. Beberapa opsi yang tersedia untuk tabulasi dua variabel adalah `column`, `row` dan `cell` yang membuat tabulasi menampilkan frekuensi relatif (persentasi) dari nilai suatu nilai dibandingkan masing-masing keseluruhan kolom, baris dan total.

tabulate

Perintah `table` “menggabungkan” kekuatan dari `summarize` dan `tabulate`, di mana tabulasi (maksimal 4) statistik deskriptif dapat ditampilkan untuk dua atau lebih variabel. Sintaks perintah `table` adalah sebagai berikut:

`table varname(s), contents(clist)`

table

`varnames` adalah variabel-variabel yang akan ditabulasi. Variabel pertama akan menjadi variabel yang ditampilkan sebagai baris (i); sedangkan variabel kedua menjadi variabel kolom j. Variabel ketiga dan seterusnya akan menjadi super-kolom (kolom dalam kolom).

`clist` adalah jenis statistik yang akan ditampilkan untuk masing-masing nilai. `clist` dapat menampilkan `freq` (frekuensi) ataupun statistik lainnya: `mean`, `sd`, `max`, `min`, `median`, `sum` (spesifikasikan variabel yang hendak dideskripsikan setelah opsi ini).

`by(varlist)` Penambahan opsi ini akan membuat super-row (baris dalam baris) dengan variabel `varlist` (maksimum 4 level).

## 4.2 Grafik statistik deskriptif

Stata memiliki kemampuan grafis yang cukup komprehensif. Selain melalui perintah, menu untuk grafik tersedia dalam menu **Graphics** dengan dialog box yang cukup membantu proses pembuatan grafik yang diinginkan. Untuk memudahkan, perintah grafik akan diperkenalkan menggunakan contoh.

Perintah `histogram varname` berguna untuk membuat diagram batang. Tambahkan opsi `normal` untuk menambahkan grafik distribusi normal. Contoh perintah yang dapat dijalankan:

```
sysuse citytemp
histogram tempjuly
histogram tempjuly, normal
```

Dalam kedua contoh tersebut, Stata akan menampilkan jumlah dan lebar diagram batang secara otomatis. Untuk mengubah jumlah tersebut, gunakan opsi `bin(#)` dan `width(arg1#)`. Untuk diagram batang tampilan jumlah frekuensi dan persentase, tambahkan opsi `frequency` dan `percentage`. Bila ingin menggunakan satu diagram batang untuk setiap nilai, tambahkan opsi `discrete`.

Perintah `twoway (plot)` berguna untuk membuat grafik antar variabel. Bila lebih dari satu `plot`, tanda kurung `()` untuk menandai masing-masing plot. Elemen definisi `(plot)` adalah sebagai berikut:

**jenis plot** — Dua pilihan yang sering digunakan adalah `scatter`, dan `line`. Selain plot data langsung, dapat juga ditambahkan plot trend/regresi, yaitu `lfit` (linear fit) `qfit` (quadratic fit). Pilihan `lfitci` dan `qfitci` menampilkan *confidence interval* yang dapat diset dengan tambahan opsi grafis `level(#)`. Perlu diperhatikan bahwa jenis plot tertentu dapat memiliki sintaks yang berbeda dengan yang dibahas di sini (contohnya `rarea`).

**varlist y** — Dapat terdapat lebih dari satu variabel y untuk satu definisi plot (diplot masing-masing).

**varname x** — Hanya terdapat satu variabel x untuk satu definisi plot.

**opsi** — Plot dapat menggunakan beberapa skala y maupun x, yang dispesifikasikan dengan `yaxis(#)` dan `xaxis(#)`. Terdapat banyak opsi lain yang tergantung pada jenis plot. Contohnya, kita dapat menspesifikasikan jenis marker untuk `scatter` (lihat `help marker_options`) dan garis (lihat `help line_options`) untuk `line`.

Perintah `histogram` dan `twoway` yang diperkenalkan di atas adalah “singkatan” dari perintah `graph` Stata, di mana sintaks lengkap adalah `graph twoway`. Khusus `scatter` dan `line`, sintaks `twoway` dapat dihilangkan (contohnya, `line varname`).

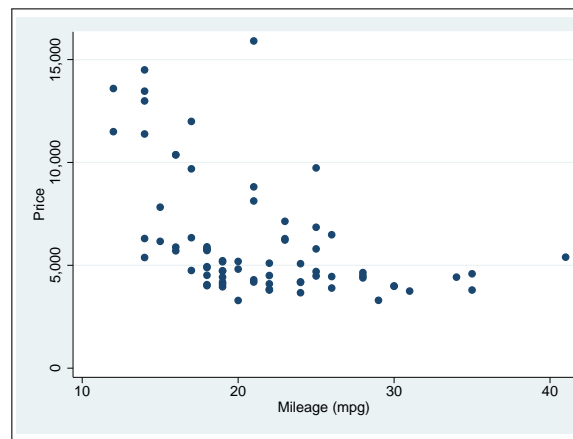
histogram

twoway

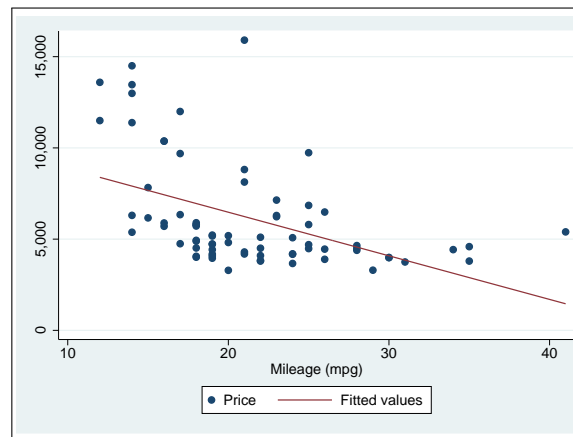
Hasil dari perintah grafik Stata dapat disimpan ke dalam file grafis untuk digunakan dalam program lainnya. Perintah untuk mengekspor grafik yang sedang terbuka tersebut adalah `graph export filename.ext`. `.ext` diperlukan untuk menspesifikasikan tipe file yang diekspor, seperti `.pdf`, `.wmf`, `.eps` dan `.png`.

Berikut adalah beberapa contoh perintah penggunaan grafik `twoway` dengan menggunakan dataset bawaan Stata.

```
sysuse auto
twoway scatter price mpg
```

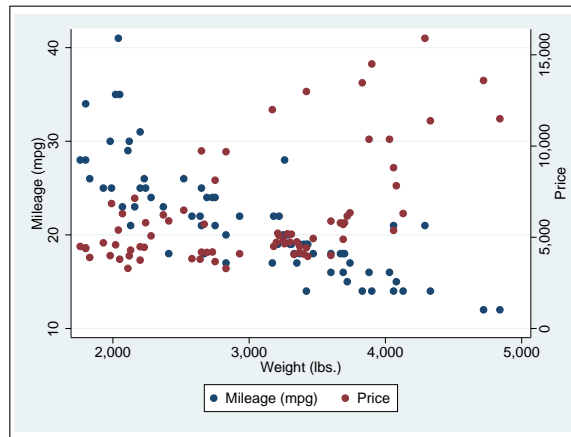


```
twoway (scatter price mpg) (lfit price mpg)
```

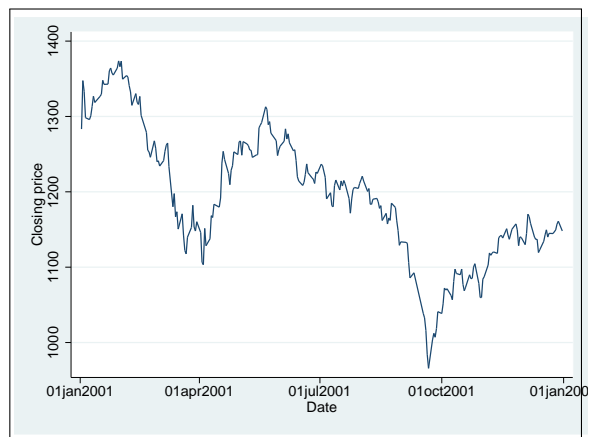




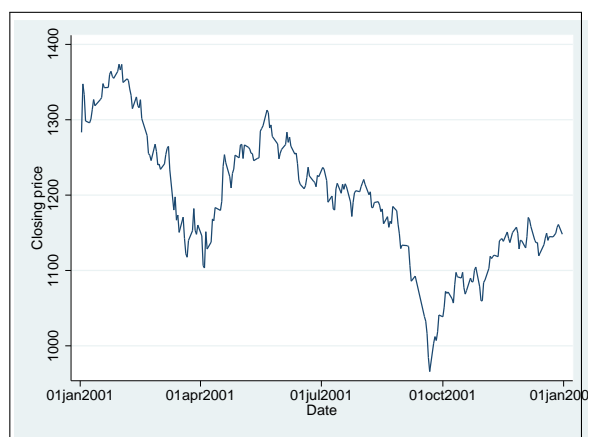
```
twoway (scatter mpg weight) (scatter price weight, yaxis(2))
```



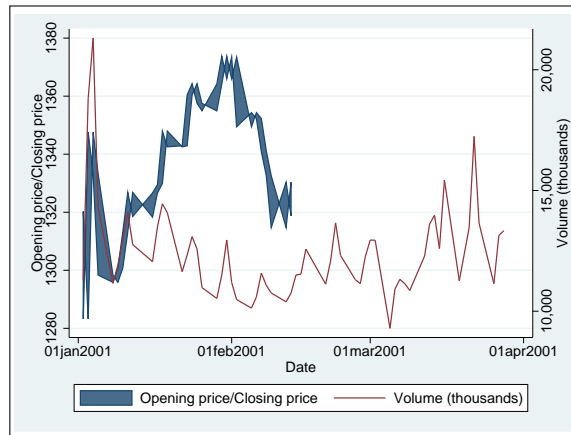
```
sysuse sp500  
twoway line close date
```



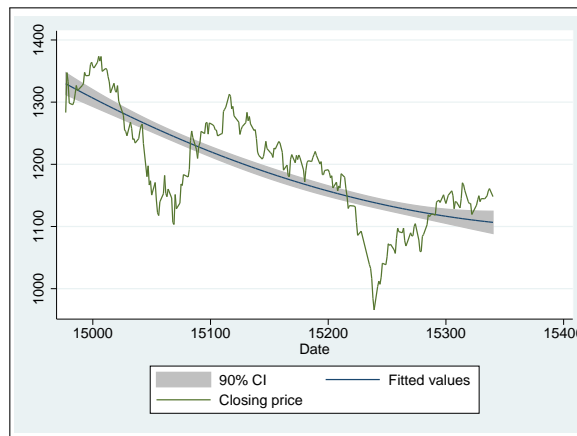
```
twoway line close date, xscale(log)
```



twoway (rarea open close date in 1/30) (line volume in 1/60,  
yaxis (2))



twoway (qfitci close date, level(90)) (line close date)



sysuse citytemp  
histogram tempjuly, normal

