ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ
SCHOOL OF BUSINESS

ΜΕΤΑΠΤΥΧΙΑΚΟ ΛΟΓΙΣΤΙΚΗΣ & ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΗΣ
MSc IN ACCOUNTING & FINANCE

Χρηστικός Οδηγός STATA

Αθήνα 2024

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Αντί προλόγου

Ο χρηστικός οδηγός **STATA** συντάχθηκε με σκοπό να αποτελέσει ένα βοήθημα για τους φοιτητές του Μεταπτυχιακού Προγράμματος Λογιστικής & Χρηματοοικονομικής του Τμήματος Λογιστικής & Χρηματοοικονομικής του Οικονομικού Πανεπιστημίου Αθηνών. Παρέχει στους φοιτητές βασικές οδηγίες χρήσης του στατιστικού προγράμματος λογισμικού **STATA** προκειμένου να εκτελέσουν βασική επεξεργασία δεδομένων και στατιστική ανάλυση στο πλαίσιο του προαναφερθέντος προγράμματος λογισμικού.

Σημειώνεται ότι ο χρηστικός οδηγός **STATA** δεν μπορεί να καλύψει το σύνολο των δυνατοτήτων του στατιστικού προγράμματος λογισμικού **STATA** αλλά και ούτε τις ενδεχόμενες απαιτήσεις στατιστικής ανάλυσης που δύναται να αναδειχθούν κατά τη διαδικασία εκπόνησης μίας διπλωματικής εργασίας. Τούτων λεχθέντων, υπογραμμίζεται ότι στο διαδίκτυο είναι δυνατόν να ανευρεθεί άφθονο και σχετικά αξιόπιστο υλικό υποστήριξης για την διενέργεια ποικίλων ενεργειών στατιστικής ανάλυσης το οποίο δύναται να αναζητήσει με σχετική ευκολία ένας φοιτητής υπό το πνεύμα της αξιολογική κρίσης του.

Τέλος, την ευθύνη για οποιαδήποτε σφάλματα την έχει ο συντάκτης του ανά χείρας χρηστικού οδηγού υπό την πρόνοια ότι η χρήση του από τους φοιτητές για την εκπόνηση της διπλωματική εργασία δεν τους απαλλάσσει από την προσωπική και αποκλειστική ευθύνη τους αναφορικά με την αξιοπιστία της στατιστικής ανάλυσης, επεξεργασίας και των παρουσίασης δημοσιευμένων αποτελεσμάτων στο πλαίσιο εκπόνησης της διπλωματικής εργασίας τους ή οποιαδήποτε άλλου συναφές παράγωγου πνευματικής εργασίας.

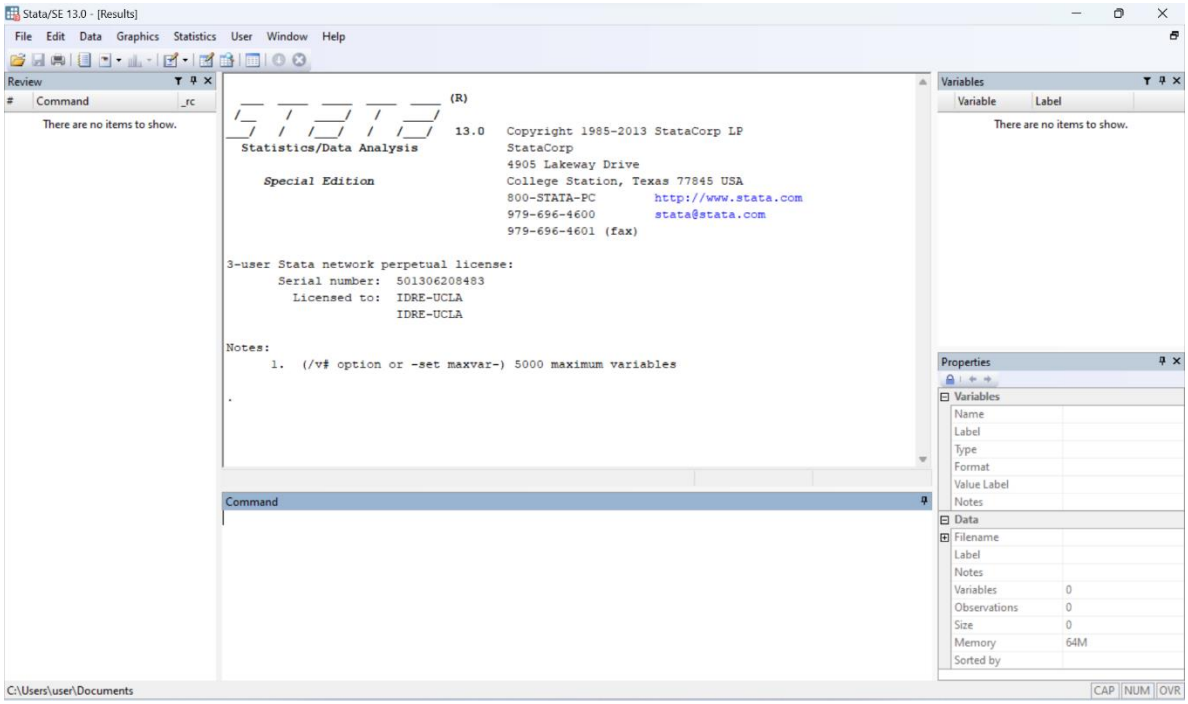
Ορέστης Βλησμάς
Αναπληρωτής Καθηγητής Λογιστικής
Τμήμα Λογιστικής & Χρηματοοικονομικής
Οικονομικό Πανεπιστήμιο Αθηνών

1. Γενικά περί περιβάλλοντος εργασίας STATA

1.1. Γενικό πλαίσιο διεπαφής

Όταν εκτελείται μία συνεδρία στο στατιστικό πρόγραμμα λογιστικού STATA, εμφανίζεται η ακόλουθη οθόνη με πέντε επιμέρους παράθυρα (windows) όπως εμφανίζεται στην Εικόνα 1.

Εικόνα 1: Γενική Άποψη Οθόνης STATA

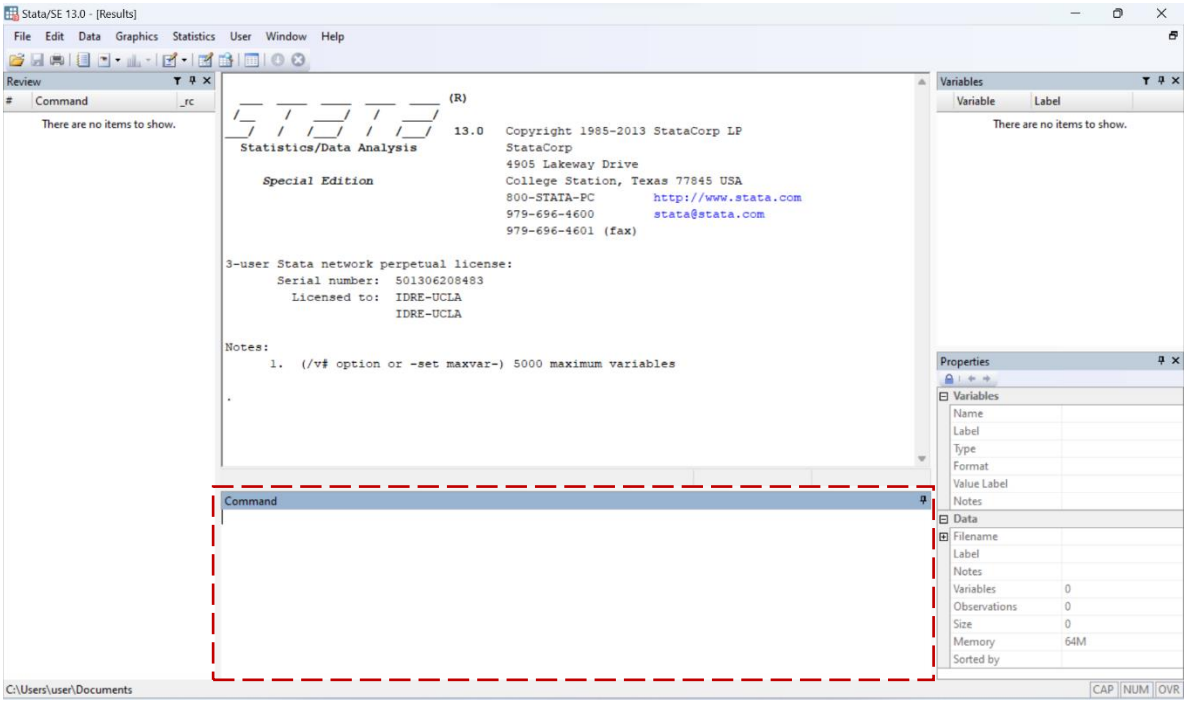


Από τη γενική άποψη της οθόνης STATA της Εικόνας 1 είναι δυνατόν να διαπιστώσουμε τα ακόλουθα επιμέρους παράθυρα (πεδία):

1. Command
2. Result
3. Review
4. Variables
5. Properties

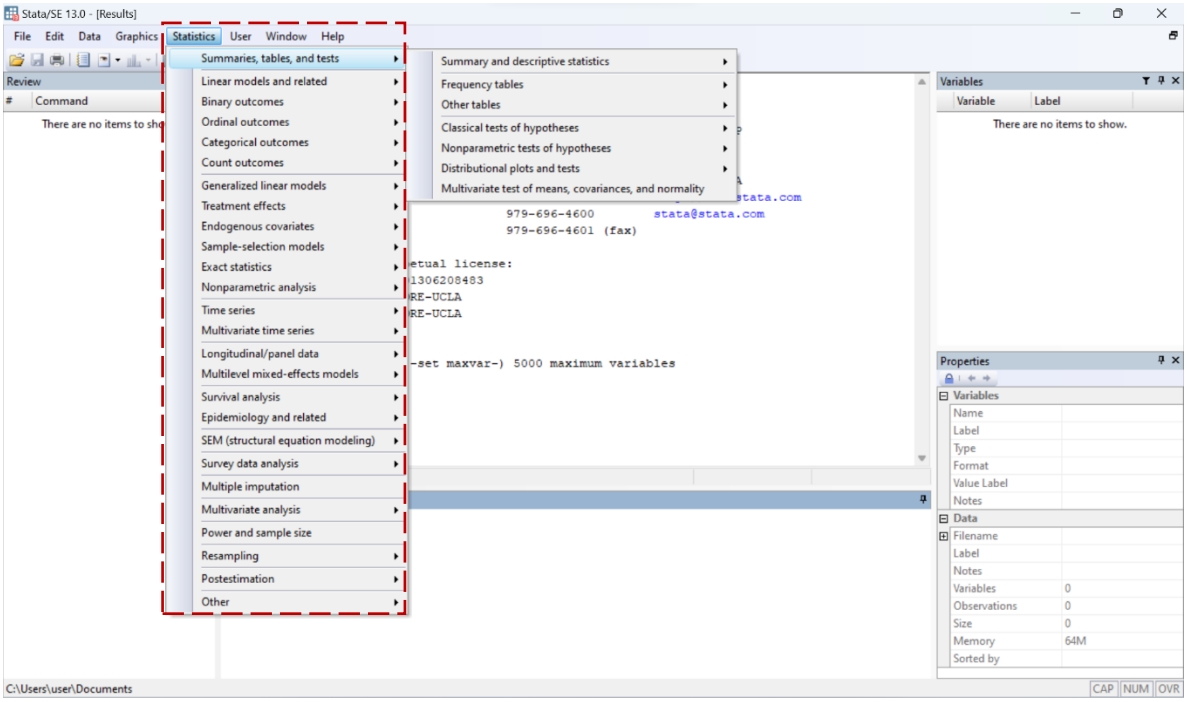
Το περισσότερο σημαντικό παράθυρο (πεδίο) με επικεφαλίδα Command (εντολές). Ο λόγος είναι ότι στο πεδίο αυτό πληκτρολογούμε τις εντολές που δίνουμε στο στατιστικό πρόγραμμα λογισμικού **STATA** προκειμένου να τις εκτελέσει. Στην Εικόνα 2 επισημαίνεται το παράθυρο (πεδίο) με επικεφαλίδα Command.

Εικόνα 2: Παράθυρο (Πεδίο) με Επικεφαλίδα Command



Σημειώνεται ότι στο στατιστικό πρόγραμμα λογισμικού STATA συνηθίζεται να πληκτρολογούνται οι σχετικές εντολές που θα πρέπει να εκτελεσθούν. Παρόλο που αυτό αποτελεί καθιερωμένη πρακτική από τους ερευνητές, το στατιστικό πρόγραμμα λογισμικού STATA δίδει τη δυνατότητα προκαθορισμένων επιλογών μέσω drop-down windows από την πάνω οριζόντια μπάρα (Εικόνα 3).

Εικόνα 3: Εκτέλεση Εντολών μέσω επιλογών Drop-down Windows



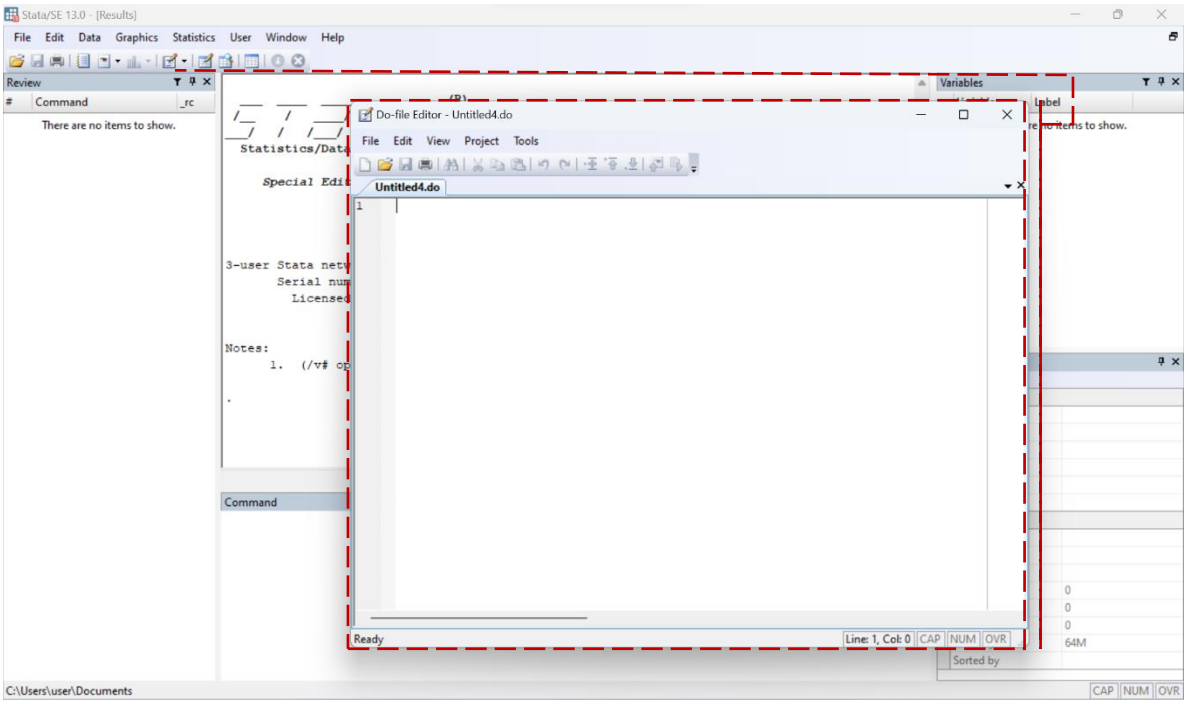
Επειδή το παράθυρο (πεδίο) με επικεφαλίδα Command αποτελεί και χώρο πειραματισμών από τον ερευνητή εναλλακτικών σεναρίων στατιστικής ανάλυσης κάποια εκ των οποίων στην πορεία θα τροποποιηθούν ή θα πρέπει να διατηρηθούν (αποθηκευτούν) για μελλοντική χρήση συνίσταται να γίνεται παράλληλη χρήση του do-editor.

Ο do-editor παρέχει μεταξύ άλλων τις ακόλουθες δυνατότητες:

1. Αποτελεί ένα κειμενογράφο με δυνατότητες αποθήκευσης και ανάκτησης αρχείων εντολών STATA. Τα αρχεία αυτά καλούνται do-files και έχουν προέκταση do, δηλαδή η ονομασία των do-files έχει ως εξής: XXXX.do
2. Δίδει τη δυνατότητα να εκτελεσθούν όλες οι εντολές που περιλαμβάνονται σε ένα do-file ή να επισημανθεί (highlight) ένας αριθμός αυτών και να εκτελεσθεί μόνο αυτός ο αριθμός εντολών.

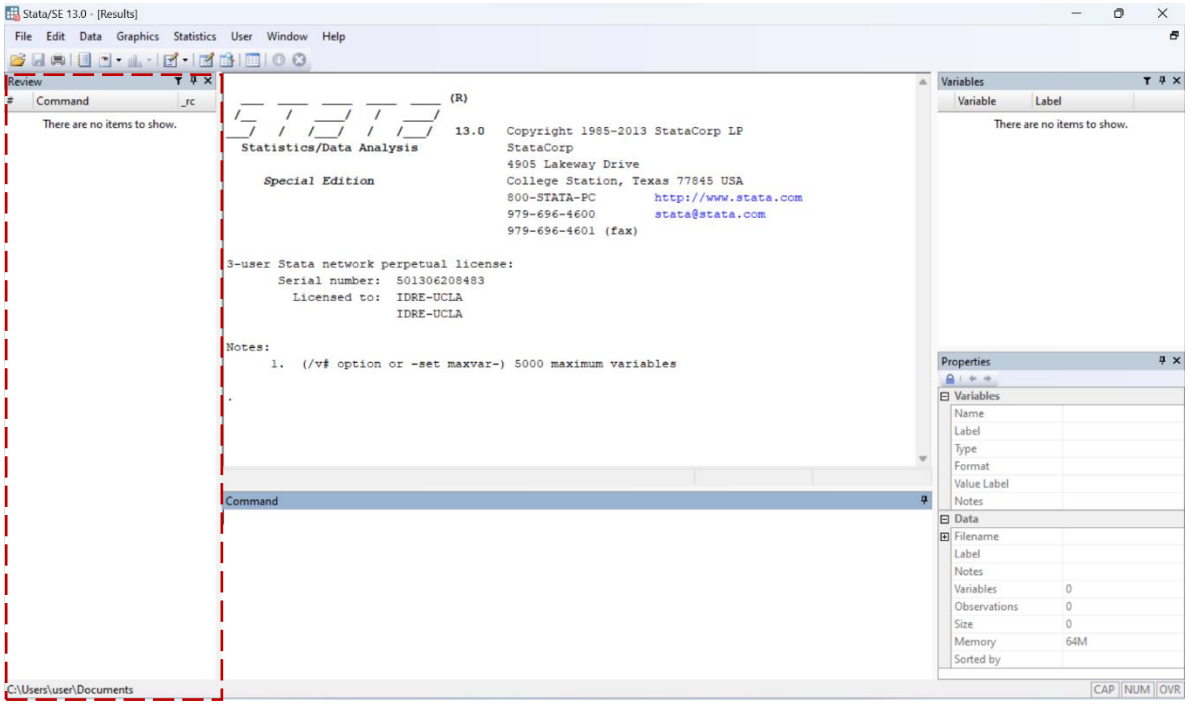
Ο do-editor ενεργοποιείται με την επιλογή  και αναδύεται σχετικό παράθυρο (window). Στην Εικόνα 4 αποδίδεται τα παράπανω γραφικά.

Εικόνα 4: Ο do-editor στο STATA



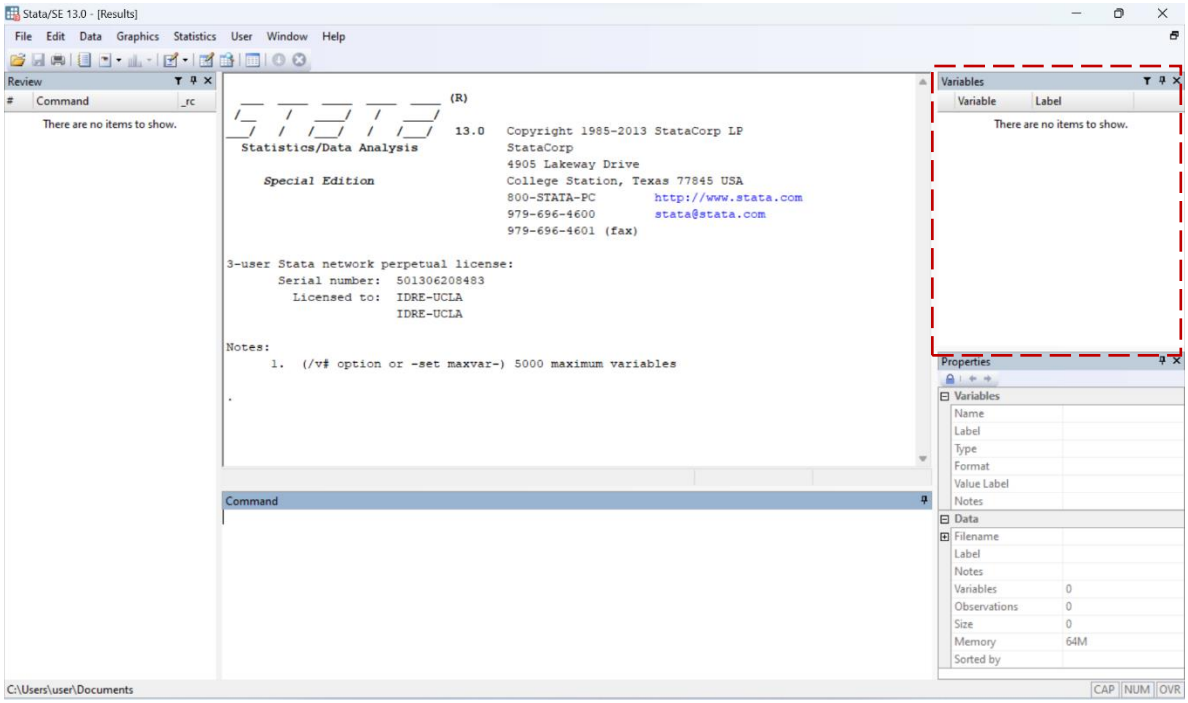
Το παράθυρο (πεδίο) με επικεφαλίδα Review (επισκόπηση) αποθηκεύει προσωρινά όλες τις εντολές με χρονολογική σειρά που έχουν εκτελεστεί σε μία συνεδρία του **STATA**. Αποθηκεύονται ακόμη και οι εντολές οι οποίες δεν ήταν δυνατόν να εκτελεσθούν με κατάλληλη σήμανση λάθους (error code). Στην Εικόνα 5 επισημαίνεται το παράθυρο (πεδίο) με επικεφαλίδα Review.

Εικόνα 5: Παράθυρο (Πεδίο) με Επικεφαλίδα Review



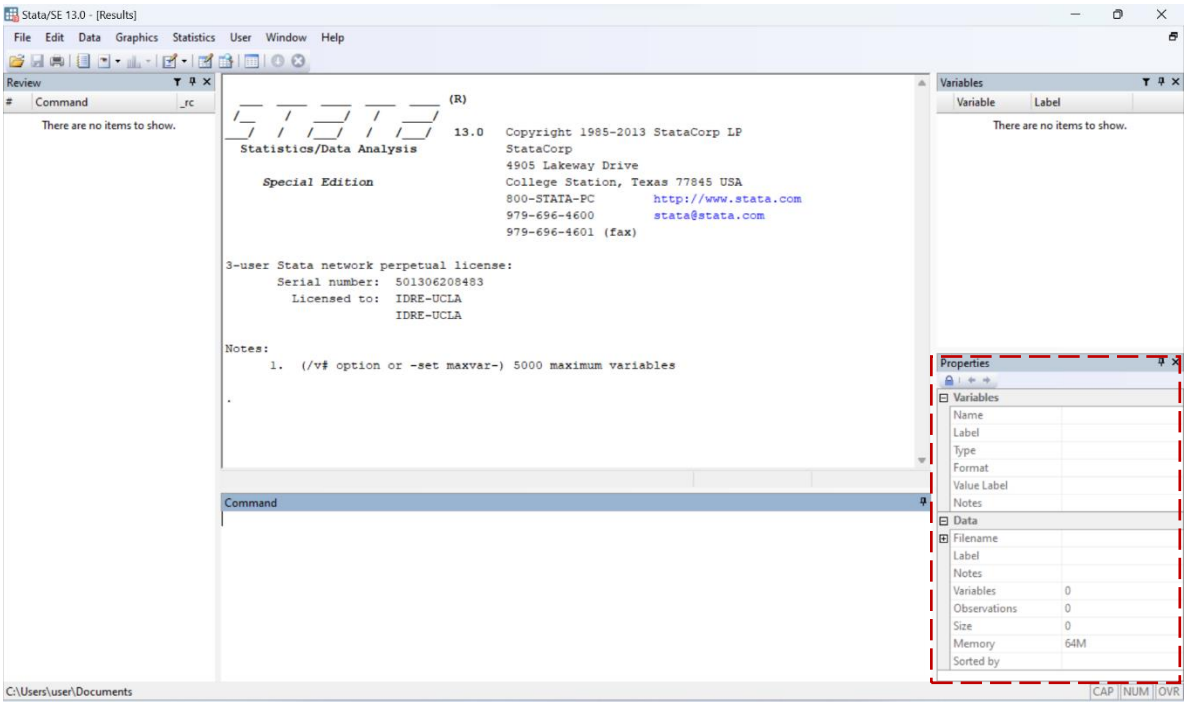
Το παράθυρο (πεδίο) με επικεφαλίδα Variables (μεταβλητές) όπου εμφανίζει το σύνολο των μεταβλητών οι αριθμητικές τιμές των οποίων ευρίσκονται στο data editor κατά τη διάρκεια μίας συνεδρίας του **STATA**. Επίσης εμφανίζεται ο τύπος των μεταβλητών. Στην Εικόνα 6 επισημαίνεται το παράθυρο (πεδίο) με επικεφαλίδα Variables.

Εικόνα 6: Παράθυρο (Πεδίο) με Επικεφαλίδα Variables



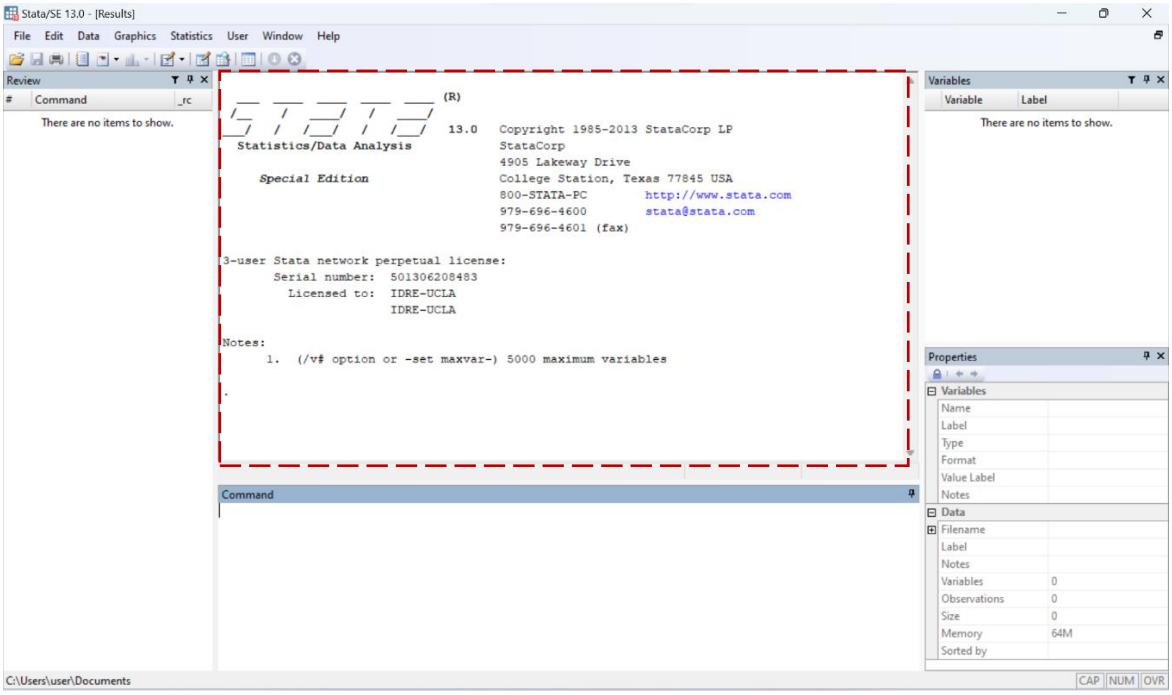
Το παράθυρο (πεδίο) με επικεφαλίδα Properties (ιδιότητες) εμφανίζει τις ιδιότητες της μεταβλητής που έχει επιλεγεί στο παράθυρο (πεδίο) με επικεφαλίδα Variables (μεταβλητές).

Εικόνα 7: Παράθυρο (Πεδίο) με Επικεφαλίδα Properties



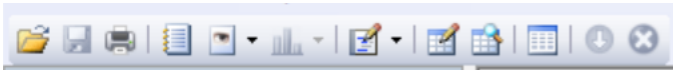
Τέλος τα αποτελέσματα από την εκτέλεση μίας εντολής στο **STATA** εμφανίζονται στο παράθυρο (πεδίο) με Results (αποτελέσματα). Στην Εικόνα 8 επισημαίνεται το παράθυρο (πεδίο) Results.

Εικόνα 8: Παράθυρο (Πεδίο) Αποτελεσμάτων (Results)



1.2. Περί βασικής γραμμής εργαλείων (toolbar)

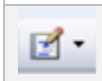
Στο πάνω μέρος της κεντρικής οθόνης μίας συνεδρίας του **STATA** εμφανίζεται η ακόλουθη οριζόντια γραμμή εργαλείων (toolbar):



Τα παραπάνω εικονίδια δίδουν τη δυνατότητα στο χρήση για πιο γρήγορη πρόσβαση σε διάφορες εντολές γενικής φύσης. Ακολουθεί σχετικός πίνακας που παρουσιάζει τη λειτουργίας (δηλαδή την εντολή που εκτελεί) το κάθε ένα από τα παραπάνω εικονίδια.

Πίνακας 1: Βασική Γραμμή Εργαλείων (Toolbar) του STATA

Εικονίδιο	Επεξήγηση
	Η πρώτη επιλογή στη γραμμή εργαλείων είναι για το άνοιγμα διαφόρων τύπων εγγράφων. Κάνοντας κλικ και κρατώντας πατημένο αυτό το κουμπί θα εμφανιστούν τα τελευταία αρχεία δεδομένων που άνοιξαν, επιτρέποντάς σας να επιλέξετε ένα για να το ανοίξετε ξανά. Κάνοντας ένα κλικ σε αυτό το κουμπί θα ανοίξει ένας τυπικός διάλογος ανοίγματος αρχείου. Αυτό θα σας επιτρέψει να περιηγηθείτε στον υπολογιστή σας και να επιλέξετε ποιο αρχείο θα ανοίξετε. Από προεπιλογή, το **STATA** ενεργοποιεί τα αρχεία δεδομένων **STATA**, αλλά μπορείτε επίσης να ανοίξετε do-files ή γραφήματα **STATA** αλλάζοντας τον τύπο αρχείου που είναι ενεργοποιημένος στο πεδίο Μορφή αρχείου του παραθύρου Άνοιγμα.
	Η δεύτερη επιλογή στη γραμμή εργαλείων είναι για την αποθήκευση ενός ανοιχτού αρχείου δεδομένων, είναι ισοδύναμη με την πληκτρολόγηση "save, replace" στο παράθυρο (πεδίο) Command.
	Η τρίτη επιλογή στη γραμμή εργαλείων είναι για εκτύπωση. Κάνοντας κλικ παρατεταμένα θα εμφανιστούν όλα τα ανοιχτά παράθυρα που μπορούν να εκτυπωθούν. Κάνοντας ένα κλικ θα ανοίξει ένας διάλογος εκτύπωσης για όποιο παράθυρο βρίσκεστε αυτήν τη στιγμή.
	Η τέταρτη επιλογή στη γραμμή εργαλείων αφορά το ημερολόγιο (αρχείο καταγραφής ή διαφορετικά log-file) του **STATA**. Ένα ημερολόγιο (log-file) **STATA** περιέχει μια λίστα με κάθε εντολή και αποτέλεσμα που συνέβη κατά την ενεργοποίηση της καταγραφής ημερολογίου μίας συνεδρίας του **STATA**. Η χρησιμότητα του είναι εξαιρετική. Για να ξεκινήσει η καταγραφή ημερολογίου, απλώς κάντε κλικ στο εικονίδιο του ημερολογίου και θα σας ζητήσει να ορίσετε όνομα αρχείου και περιοχή αποθήκευση του (ένα log-file έχει προέκταση smc1o, δηλαδή η ονομασία των log-files έχει ως εξής: XXXX.smc1) και επιλέξτε Έναρξη (Begin). Εναλλακτικά μπορείτε να ανακτήσετε ένα παλαιότερο log-file. Αν στην ίδια συνεδρία STATA κάνετε κλικ για δεύτερη φορά στο σχετικό εικοδίο

	μπορείτε είτε να διακόψετε την τρέχουσα καταγραφή, να αναστείλετε την τρέχουσα καταγραφή ή επαναφέρετε την καταγραφή ημερολογίου.
	Η πέμπτη επιλογή στη γραμμή εργαλείων θα ανοίξει ένα παράθυρο του Viewer ή θα το φέρει στο μπροστινό μέρος. Το παράθυρο Viewer είναι ένας χρήσιμος τρόπος πρόσβασης στα αρχεία βοήθειας. Μπορείτε να το κάνετε αυτό πληκτρολογώντας σχετικές λέξεις στο πλαίσιο αναζήτησης δίπλα στον μεγεθυντικό φακό ή πληκτρολογώντας μια εντολή στο πλαίσιο δίπλα στο κουμπί Find.
	Το έκτο κουμπί φέρνει το παράθυρο του γραφήματος μπροστά. Και πάλι, χρήσιμο αν το χάσετε πίσω από άλλα ανοιχτά παράθυρα.
	Το έβδομο κουμπί σάς επιτρέπει να επεξεργαστείτε το αρχείο των εντολών δηλαδή το do-file. Κάνοντας κλικ θα δημιουργηθεί ένα νέο αρχείο do-file και θα το φέρει στο μπροστινό μέρος. Κάνοντας κλικ παρατεταμένα θα εμφανιστεί μια λίστα με τα ανοιχτά αρχεία do-file και θα σας επιτρέψει να επιλέξετε ένα. Τα αρχεία do-file είναι συνήθως λίστες εντολών που το STATA εκτελεί με προκαθορισμένη σειρά.
	Το όγδοο κουμπί ανοίγει το ενεργό αρχείο δεδομένων (.dta) και το φέρνει στο μπροστινό μέρος, επιτρέποντάς σας να το επεξεργαστείτε.
	Το ένατο κουμπί ανοίγει επίσης το ενεργό αρχείο δεδομένων. Ωστόσο, σε αντίθεση με το πρόγραμμα επεξεργασίας, το πρόγραμμα περιήγησης δεν σας επιτρέπει να αλλάξετε τα δεδομένα. Έτσι είναι χρήσιμο και ασφαλέστερο σε καταστάσεις όπου θέλετε απλώς να δείτε κάτι στα δεδομένα σας.
	Το δέκατο κουμπί επιτρέπει τη διαχείριση των μεταβλητών.
	Το ενδέκατο κουμπί θα εμφανίσει απλώς την επόμενη οθόνη σε μια μεγάλη οθόνη ή εντολή.
	Το τελευταίο κουμπί θα ακυρώσει μια εντολή. Έτσι, αν έβαζα λάθος εντολή, θα μπορούσα να πατήσω αυτό το κουμπί και θα ήταν σαν να μην είχα εκδώσει ποτέ την εντολή. Σημειώστε ότι αυτό δεν είναι κουμπί αναίρεσης και θα ακυρώσει μόνο μια εντολή που εξακολουθεί να εκτελείται. Σημειώστε επίσης ότι είναι κυρίως χρήσιμο εάν η εντολή έχει αρκετά μεγάλη έξοδο ώστε να εμφανίζεται -περισσότερα. Διαφορετικά δεν θα το ακυρώσετε αρκετά γρήγορα. Χρησιμοποιώντας το κουμπί X, εμφανίζεται ως -Break- με κόκκινο χρώμα στο παράθυρο Αποτελέσματα και δεν περιλαμβάνεται στο παράθυρο Review.

2. Διαχείριση δεδομένων στο STATA

2.1. Πηγαία δεδομένα και τρόπος εισαγωγής τους στο STATA

Ο ερευνητής αποθηκεύει τα προς επεξεργασία δεδομένα σε αρχεία τα οποία δύναται να έχουν διάφορες μορφές. Ενίοτε, η μορφή του αρχείου των δεδομένων έχει παραχθεί αυτόματα κατά τη διαδικασία λήψης δεδομένων από μία βάση δεδομένων στην οποία έχει συνδρομητική πρόσβαση ο ερευνητής.

Το **STATA** έχει τρεις κύριες εντολές για την εισαγωγή δεδομένων από άλλα προγράμματα: η infile, η import delimited και η εντολή infix. Επιπλέον, αν υποθέσουμε ότι το πρόγραμμα στο οποίο βρίσκονται αυτήν τη στιγμή τα δεδομένα περιέχει τα δεδομένα σε έναν πίνακα (spreadsheet), η αντιγραφή και η επικόλληση συχνά λειτουργούν. Η μορφή των δεδομένων που θα εισαχθούν θα καθορίσει ποια εντολή θα χρησιμοποιήσετε τελικά. Όλες οι ακόλουθες εντολές πρέπει να πληκτρολογηθούν στο παράθυρο Command. Επίσης το **STATA** θα αρνηθεί να εκτελεστεί εάν έχετε ήδη ανοιχτά δεδομένα, έτσι ώστε να μην χάσετε κατά λάθος αυτό που εργάζεστε τόσο σκληρά.

Αρχεία δεδομένων με τη μορφή ASCII (.csv ή .tab ή .tsv ή .dat.)

Αρχεία δεδομένων με τη μορφή ASCII έχουν συνήθως τις προεκτάσεις .csv ή .tab ή .tsv ή .dat. Όταν τα δεδομένα σας εξάγονται σε μορφή ASCII (μια επιλογή που παρέχει κάθε πρόγραμμα υπολογιστικών φύλλων), χρειάζεστε μόνο την εντολή import delimited (παλαιότερα ήταν η εντολή insheet).

Στο παράθυρο Command πληκτρολογήστε την ακόλουθη εντολή:

```
import delimited using filename.extension
```

Σημειώνεται ότι ο όρος filename.extension αναφέρεται στο πλήρες path του αρχείου στο υπολογιστή σας (η προέκταση του οποίου όπως αναφέρθηκε είναι .csv ή .tab ή .tsv ή .dat). Επίσης λάβετε υπόψη τα ακόλουθα:

- Η εντολή import delimited μπορεί από μόνη της να καταλάβει τη μορφή του αρχείου δεδομένων.
- Εάν τα ονόματα των μεταβλητών δεν υπάρχουν στο αρχείο, η import delimited θα δώσει στις μεταβλητές αυθαίρετα ονόματα (π.χ. v1, v2, κ.λπ). Μπορείτε να επιστρέψετε και να αλλάξετε αργότερα.
- Εάν περιλαμβάνονται τα ονόματα των μεταβλητών, το STATA ξέρει να τα διαβάζει από την επάνω πρώτη γραμμή η οποία είναι αφιερωμένη στα ονόματα των μεταβλητών.
- Εάν το όνομα της μεταβλητής έχει περισσότερους από οκτώ χαρακτήρες, το STATA θα το συντομεύσει κατά την εμφάνιση των δεδομένων.

- Εάν θέλετε οι μεταβλητές να εισαχθούν με δική σας ονομασία (π.χ. έστω 3 μεταβλητές με την ονομασία vr1, vr2 και vr3 αντίστοιχα) τότε η εντολή διαφοροποιείται ως εξής:

```
import delimited vr1, vr2, vr3 using filename.extension
```

Αρχεία δεδομένων με τη μορφή ASCII (.raw ή txt.)

Υποθέτοντας ότι τα δεδομένα σας είναι λιγότερο τακτοποιημένα, θα χρειαστείτε την εντολή infile. Τις περισσότερες φορές αυτό συμβαίνει εάν τα δεδομένα σας είναι σε μορφή ακατέργαστων (.raw) ή κειμένου (.txt). Σε αυτές τις περιπτώσεις, τα δεδομένα συνήθως διαχωρίζονται απλώς με λευκό διάστημα (και όχι κόμματα ή καρτέλες) και μπορεί να έχουν άλλη περίεργη μορφοποίηση.

Στο παράθυρο Command πληκτρολογήστε την ακόλουθη εντολή:

```
infile using filename.extension
```

Σημειώνεται ότι ο όρος filename.extension αναφέρεται στο πλήρες path του αρχείου στο υπολογιστή σας (η προέκταση του οποίου όπως αναφέρθηκε είναι .raw ή .txt. Η εντολή infile θα πρέπει να ακολουθείτε πολλές φορές με κατάλληλους τελεστές ή οδηγίες που δεν είναι δυνατόν να προκαθορισθούν εκ προοιμίου εφόσον τα ενδεχόμενα προβλήματα με τα αρχεία .raw ή .txt δεν μπορούν να προβλεφθούν από τον παρόντα χρηστικό οδηγό.

Αρχεία δεδομένων με τη μορφή υπολογιστικού φύλλου Excell ή SPSS

Εάν εργάζεστε με αριθμούς ή συμβολοσειρές σε ένα υπολογιστικό φύλλο και δεν ενδιαφέρεστε για τη διατήρηση οποιασδήποτε πληροφορίας αλλά της πραγματικής τιμής στο κελί, μπορείτε πάντα να αντιγράψετε και να επικολλήσετε δεδομένα στο **STATA**. Ανοίξτε ένα επεξεργάσιμο παράθυρο δεδομένων στο **STATA** (μέσω του εικονιδίου data editor). Κάντε κλικ για να επισημάνετε το πρώτο κελί στην πρώτη κενή σειρά της πρώτης στήλης στην οποία θέλετε να προσθέσετε δεδομένα. Στη συνέχεια, απλώς επισημάνετε τα κελιά που θέλετε να μεταφέρετε (στο υπολογιστικό φύλλο δεδομένων σας), αντιγράψτε και επικολλήστε τα.

2.2. Προετοιμασία (εκκαθάριση) δεδομένων για στατιστική επεξεργασία στο **STATA**

Στο πεδίο της λογιστικής είναι σύνηθες τα δεδομένα που θα χρησιμοποιηθούν για στατιστική ανάλυση να λαμβάνονται από κάποια συνδρομητική βάση δεδομένων. Όσο και αν η συνδρομητική βάση δεδομένων είναι αξιόπιστη θα πρέπει ο ερευνητής να εξετάσει προσεκτικά τα δεδομένα του. Ειδικότερα θα πρέπει να εκτελεσθούν μία σειρά από εργασίες η οποίες καλούνται εκκαθάριση δεδομένων.

Στο χρηστικό οδηγό **STATA** περιγράφονται οι εξής διαδικασίες:

- Διαγραφή τιμών που δηλώνονται ως μη διαθέσιμες N/A. Δηλαδή αντικατάσταση της ένδειξης N/A σε κενό.
- Διαγραφή τιμών που είναι πρόδηλα εσφαλμένες.
- Μετατροπή με μεταβλητής συμβολοσειράς σε αριθμητική μεταβλητή.
- Μετατροπή δομής δεδομένων από δομή wide σε δομή long.
- Αντιμετώπιση ακραίων τιμών (outliers).

Διαγραφή τιμών που δηλώνονται ως μη διαθέσιμες N/A

Ελέγχοντας το spreadsheet (υπολογιστικό φύλο συνήθως μορφής xls), ο ερευνητής μπορεί να διαπιστώσει σε μερικές περιπτώσεις οι τιμές των μεταβλητών δεν είναι διαθέσιμες. Τούτο δηλώνεται ως N/A ή NA ή na. Ο ερευνητής πριν την εισαγωγή του αρχείου στο STATA θα πρέπει να φροντίσει οι παραπάνω ενδείξεις να έχουν απαλείφει ή αντικατασταθεί με αριθμητικές τιμές. Συνίσταται στο πλαίσιο το MSEXCEL να εκτελεσθεί replace (αντικατάσταση) των παραπάνω ενδείξεων με κενό.

ΠΡΟΣΟΧΗ: αν οι ενδείξεις N/A ή NA ή na αναφέρονται σε μηδενική αξία τότε θα πρέπει να αντικατασταθούν με μηδέν (0) και όχι με κενό διότι ο ερευνητής θα απωλέσει πολύτιμο αριθμό παρατηρήσεων. Για παράδειγμα πολλές εταιρείες δεν πραγματοποιούν έξοδα έρευνας και ανάπτυξης και αντί αυτό να δηλωθεί με την τιμή 0 δηλώνεται με την ένδειξη περί μη διαθεσιμότητας δεδομένων.

Μετά τη αντιμετώπιση των τιμών που δηλώνονται ως μη διαθέσιμες θα γίνει η εισαγωγή των δεδομένων στο STATA όπως αναφέρεται στην παράγραφο 2.1. και θα ακολουθήσουν τα επόμενα βήματα.

Διαγραφή τιμών που είναι πρόδηλα εσφαλμένες

Ως πρόδηλα εσφαλμένο θεωρείται οτιδήποτε αντιβαίνει θεμελιώδεις αρχές. Παραδείγματα είναι πωλήσεις με αρνητικές τιμές, κυκλοφορούν ενεργητικό με αρνητικές τιμές, κ.λπ. Έχοντας μεταφέρει τα δεδομένα του ο ερευνητής στο **STATA** μπορεί να αξιοποιήσει τις εντολές drop και keep σε συνδυασμό με το τελεστή if. Διακρίνονται κάποιες περιπτώσεις:

- Αν ο ερευνητής θέλει να απορρίψει εξολοκλήρου από το σύνολο των δεδομένων του μίας λίστα μεταβλητών (π.χ. έστω 3 μεταβλητές με την ονομασία vr1, vr2 και vr3 αντίστοιχα), τότε θα αξιοποιήσει την εντολή drop:

```
drop vr1, vr2, vr3
```

- Αν ο ερευνητής θέλει να απορρίψει τις τιμές μίας μεταβλητής που δεν ικανοποιούν κάποια κριτήρια τότε θα αξιοποιήσει την εντολή drop και τον τελεστή if. Για παράδειγμα, η απόρριψη των τιμών της μεταβλητής vr1 οι οποίες είναι μικρότερες του μηδέν μπορεί να επιτευχθεί ως εξής:

```
drop if vr1<0
```

Εναλλακτικά μπορεί να χρησιμοποιηθεί η εντολή keep και να επιτευχθεί το ίδιο αποτέλεσμα ως εξής:

```
keep vr1>=0
```

- Ένα άλλο παράδειγμα θα ήταν η απόρριψη των τιμών της μεταβλητής vr1 οι οποίες είναι μικρότερες του μηδέν και μεγαλύτερες του 10. Με την εντολή drop επιτυγχάνεται ως εξής:

```
drop if vr1<0 & vr1>10
```

ενώ με την εντολή keep η σύνταξη έχει ως εξής:

```
keep if vr1>=0 & vr1<=10
```

- Πέραν από τους τελεστές & (και η συνθήκη αυτή μαζί με τις προηγούμενες συνθήκες) ή if (εάν ισχύει η συνθήκη αυτή) υπάρχει ο τελεστής or (αν ισχύει αυτή η συνθήκη ανεξάρτητα από τις προηγούμενες). Ο τελεστής or υποδηλώνεται στο **STATA** με το σύμβολό |. Για παράδειγμα, έστω ότι ο ερευνητής θα ήθελα να απορρίψει τιμές μεταβλητής vr1 οι οποίες είναι ίσες του 5 ή του 10. Η σύνταξη της εντολής είναι:

```
drop if vr1==5 | vr1==10
```

Το ίδιο αποτέλεσμα επιτυγχάνεται με την εντολή keep κατευθύνοντας το **STATA** να κρατήσει όλες τις τιμές που είναι διάφορες του 5 ή του 10:

```
keep if vr1=!5 | vr1=!10
```

Σημείωση: όταν εμπρός από μία τιμή τεθεί το ! τότε υποδηλώνεται διάφορο (π.χ. !5 σημαίνει διάφορο του 5). Το ίδιο ισχύει για το σύνολο μίας μεταβλητής (π.χ. !vr1 σημαίνει διάφορο της vr1 αν και σε αυτή την περίπτωση συνδυάζεται με ένα πεδίο τιμών της vr1).

- Ωστόσο επειδή σε μια μεταβλητή μόνο μερικές τιμές μπορεί να είναι πρόδηλα εσφαλμένες τότε θα πρέπει να κάνετε generate και replace με if. Για παράδειγμα αν η μεταβλητή vr δεν πρέπει να έχει αρνητικές τιμές τότε ορίζουμε μία νέα μεταβλητή, έστω vr1 με τη σύνταξη της εντολής ν είναι:

```
generate vr1=.
replace vr1=vr if vr>=0
```

Από την παραπάνω ανάλυση ο αναγνώστης μπορεί να συνοψίσει τα ακόλουθα:

- Οι εντολές drop και keep λειτουργούν αντίστροφα διότι η πρώτη αναφέρεται σε απόρριψη ενός υποσυνόλου δεδομένων ενώ η δεύτερη αναφέρεται στη διατήρηση του συμπληρωματικού υποσυνόλου.
- Οι εντολές του **STATA** αποκτούν ιδιαίτερη δυναμική όταν συνδυασθούν με τους τελεστές. Οι τελεστές εισάγουν συνθήκες υπό τις οποίες εκτελείται μία εντολή. Οι συνηθέστεροι τελεστές είναι οι ακόλουθοι:
    - If: η εντολή εκτελείται όταν ισχύουν οι συνθήκη(ες) που ακολουθούν μετά το if.

- &: χρησιμοποιείται για να υποδηλώσει ότι η εντολή θα εκτελεσθεί όταν ισχύει ΚΑΙ (and) η συνθήκη που ακολουθεί το &.
- |: χρησιμοποιείται για να υποδηλώσει ότι η εντολή θα εκτελεσθεί όταν ισχύει η συνθήκη που ακολουθεί το | ΑΝΕΞΑΡΤΗΤΑ (or) από τις υπόλοιπες συνθήκες.
- !: χρησιμοποιείται για να υποδηλώσει ότι η εντολή θα εκτελεσθεί όταν ΔΕΝ ισχύει η συνθήκη που ακολουθεί το !.

- Το σύμβολο ! χρησιμοποιείται για να υποδηλώσει το διάφορο από μία τιμή η μία μεταβλητή).
- Τα σύμβολα ισότητας/ανισότητας κατά τη σύνταξη των εντολών **STATA** έχουν ως εξής:
  - == : ίσο με μία αριθμητική ή λογική τιμή.
  - >= : μεγαλύτερο ή ίσο με μία αριθμητική η λογική τιμή.
  - > : μεγαλύτερο από μία αριθμητική η λογική τιμή.
  - <= : μικρότερο ή ίσο με μία αριθμητική ή λογική τιμή.
  - < : μικρότερο από μία αριθμητική ή λογική τιμή.
  - != : διάφορο από μια αριθμητική ή λογική τιμή.

Σημειώνεται επίσης ότι μία μεταβλητή συμβολοσειράς μπορεί να αποδίδει μια ποιοτική μεταβλητή. Η συμμετοχή της ποιοτικής μεταβλητής στη στατιστική ανάλυση συνεπάγεται απόδοση αριθμητικών τιμών στις διάφορες καταστάσεις της και τούτο επιτυγχάνεται με την εντολή encode.

Μετατροπή με μεταβλητής συμβολοσειράς σε αριθμητική μεταβλητή

Το **STATA** αναγνωρίζει κατά κύριο λόγο μία μεταβλητή ως συμβολοσειρά (string variable) ή ως αριθμητική (numeric). Οι μεταβλητές συμβολοσειράς μπορεί να περιέχουν και χαρακτήρες και αριθμούς, οι αριθμητικές μεταβλητές μόνο αριθμούς. Στο πλαίσιο της οικονομετρικής στατιστικής ανάλυσης, ο ερευνητής θα πρέπει να βεβαιωθεί ότι όλες οι μεταβλητές που θα αξιοποιήσει είναι αριθμητικές. Αν υπάρχουν μεταβλητές ως συμβολοσειρές θα πρέπει να μετατραπούν σε αριθμητικές.

Η εντολή που χρησιμοποιείται για να μετατραπεί μία συμβολοσειρά σε αριθμητική μεταβλητή είναι η destring. Διακρίνονται κάποιες περιπτώσεις:

- Αν ο ερευνητής θέλει να δημιουργήσει μία νέα αριθμητική μεταβλητή (π.χ. έστω με την ονομασία vr1_DS) που θα αποδίδει με αριθμητικές τιμές μία μεταβλητή συμβολοσειράς (π.χ. έστω με την ονομασία vr1_S) χωρίς να καταργηθεί η μεταβλητή συμβολοσειράς, τότε θα αξιοποιήσει την εντολή destring:

```
destring  vr1_S, generate (vr1_DS)
```

- Η παραπάνω διαδικασία μπορεί να εκτελεσθεί για παραπάνω από μία μεταβλητές. Αν ο ερευνητής θέλει να δημιουργήσει νέες αριθμητικές

μεταβλητές (π.χ. έστω τρεις νέες μεταβλητές με την ονομασία vr1_DS, vr2_DS και vr3_DS) που θα αποδίδουν με αριθμητικές τιμές τρεις αντίστοιχες μεταβλητές συμβολοσειράς (π.χ. έστω με την ονομασία vr1_S, vr2_S και vr3_S αντίστοιχα) χωρίς να καταργηθούν οι μεταβλητές συμβολοσειράς, τότε θα αξιοποιήσει την εντολή destring με την εξής σύνταξη:

```
destring  vr1_S vr2_S vr3_S, generate (vr1_DS vr2_DS vr3_DS)
```

- Αν ο ερευνητής δεν επιθυμεί τη δημιουργία νέων μεταβλητών αλλά απευθείας την αντικατάσταση της συμβολοσειράς με τα αριθμητικά ισοδύναμα της τότε αντί για την επιλογή generate αξιοποιεί την επιλογή replace και δεν αναφέρει ονόματα νέων μεταβλητών. Η σχετική σύνταξη έχει ως εξής:

```
destring  vr1_S, replace
```

και για περισσότερες μεταβλητές:

```
destring  vr1_S vr2_S vr3_S, replace
```

Μερικές φορές το **STATA** αναγνωρίζει αριθμητικές μεταβλητές ως μεταβλητές συμβολοσειράς και ο λόγος μπορεί να είναι (α) εμπεριέχεται κάποιος ειδικός χαρακτήρας (#, $, % κ.λπ) στις αριθμητικές τιμες (π.χ. αναγράφεται 10% αντί 0.10 ή $ 1,000 αντί σκέτο 1,000) ή (β) μεταξύ του ακέραιου και του δεκαδικού μέρους ενός αριθμούς παρεμβάλλεται το κόμμα (,) αντί της τελείας (.) όπως επιβάλλει το αγγλοσαξονικό σύστημα αρίθμησης. Στην περίπτωση (α) και υποθέτοντας ότι ο ειδικός χαρακτήρας είναι $, η σύνταξη της εντολής destring έχει ως εξής:

```
destring  vr1_S, replace ignore ($)
ή
destring  vr1_S, generate (vr1_DS) ignore ($)
```

Στην περίπτωση (β), η σύνταξη της εντολής destring έχει ως εξής:

```
destring  vr1_S, replace dpcomma
ή
destring  vr1_S, generate (vr1_DS) dpcomma
```

Η εντολή για τη μετατροπή μίας μεταβλητής συμβολοσειράς σε αριθμητική μεταβλητή είναι tostring με τις δυνατότητες generate (νέα μεταβλητή) ή replace (την ίδια μεταβλητή). Επειδή δεν είναι συνήθης η περίπτωση αυτή δεν γίνεται περεταίρω ανάλυση επί του θέματος.

Μετατροπή δομής δεδομένων από δομή wide σε δομή long

Τα δεδομένα που λαμβάνονται από μία συνδρομητική βάση δεδομένων δύναται να αφορούν μεταβλητές που δύναται να χαρακτηριστούν ως είτε χρονολογικές σειρές, είτε διαστρωματικές παρατηρήσεις, είτε δεδομένα τύπου πάνελ:

- Χρονολογική σειρά: μία χρονολογική ακολουθία τιμών για μία περίπτωση. Παραδείγματα: η αξία των πωλήσεων μίας εταιρείας της τελευταίας

δεκαετίας, η αξία των μηνιαίων πωλήσεων μίας εταιρείας για το χρονικό διάστημα από το 1990 έως και το 2020, το μηνιαίο βάρος ενός αθλητή τους τελευταίους 36 μήνες.

- Διαστρωματικά δεδομένα: μία ακολουθία τιμών για όλες τις περιπτώσεις σε ένα χρονικό σημείο, Παραδείγματα: το ύψος των πωλήσεων ενός αριθμού εταιρειών για τη λογιστική χρήση 2024, η αξία των μηναίων πωλήσεων ενός αριθμού εταιρειών για το μήνα Ιανουάριο 2024, το βάρος ενός αριθμού αθλητών για το μήνα Ιανουάριο 2024.
- Δεδομένα τύπου πάνελ: συνδυασμός χρονολογικής σειράς και διαστρωματικών δεδομένων, δηλαδή ένας αριθμός χρονολογικών ακολουθιών τιμών και έναν αριθμό περιπτώσεων. Παραδείγματα: το ύψος των πωλήσεων ενός αριθμού εταιρειών της τελευταίας δεκαετίας, η αξία των μηναίων πωλήσεων ενός αριθμού εταιρειών για το χρονικό διάστημα από το 1990 έως και το 2020, το βάρος ενός αριθμού αθλητών τους τελευταίους 36 μήνες.

Στην ενότητα αυτή θα επικεντρωθούμε στη δομή των δεδομένων τύπου πάνελ. Δίδεται έμφαση στα δεδομένα πάνελ διότι είναι η συνήθης περίπτωση. Η δομή των δεδομένων τύπου πάνελ όταν γίνεται η αποθήκευση σε ένα φύλλο εργασίας xls με την μορφή wide.

Έστω ότι έχουμε αποθηκεύσει δεδομένα για δύο εταιρείες (την A και την B) αναφορικά με την αξία των assets και των sales. Τα δεδομένα αφορούν τρία έτη: 2000, 2001 και 2002. Σε αυτή την περίπτωση, η δομή wide έχει ως εξής:

Firm_id	Assets2000	Assets2001	Assets2002	Sales200	Sales2001	Sales2002
A	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX
B	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX

Είναι εμφανές ότι οι εταιρείες θέτονται στην πρώτη στήλη, μετά ακολουθούν τρεις στήλες με την αξία των assets για διαδοχικά τρία έτη και έπειτα άλλες τρεις στήλες με την αξία των sales για διαδοχικά τρία έτη. Η πρώτη στήλη αντιπροσωπεύει τη μεταβλητή Firm_id και αφορά το μοναδικό κωδικό που διακρίνεται η κάθε εταιρεία στο δείγμα των δεδομένων.

Η στατιστική ανάλυση στο πλαίσιο του **STATA**, προϋποθέτει ότι η δομή των δεδομένων θα είναι long. Η δομή long στο προηγούμενο παράδειγμα έχει ως εξής:

Firm_id	Time_id	Assets	Sales
A	2000	XXXXX	XXXXX
A	2001	XXXXX	XXXXX
A	2002	XXXXX	XXXXX
B	2000	XXXXX	XXXXX
B	2001	XXXXX	XXXXX
B	2002	XXXXX	XXXXX

Το χαρακτηριστικό της δομής long σε σχέση με τη δομή wide είναι ότι εισάγεται ανεξάρτητη μεταβλητή δηλωτική του χρόνου (Time_id) και για κάθε μεταβλητή (assets, sales) αφιερώνεται μία στήλη.

Προκειμένου η δομή των δεδομένων να μετατραπεί από wide σε long θα πρέπει να εκτελεσθεί η εντολή reshape. Με βάση αναφοράς το προηγούμενο παράδειγμα διαπιστώνεται ότι στην περίπτωση της δομή wide υπάρχει μία επαναληπτικότητα στις μεταβλητές assets και sales που ακολουθεί το λογικό κανόνα ΟΝΟΜΑ|ΕΤΟΣ (Assets2000, Assets2001, Assets2002, Sales2000, Sales2001, Sales2002). Έστω ότι υπάρχει μία δομή δεομένων με την μεταβλητή vrA για πέντε διαδοχικά έτη (άρα εμφανίζονται οι μεταβλητές vrA2000, vrA2001, vrA2002, vrA2003 και vrA2004). Έστω ότι στη δομή wide υπάρχει η αριθμητική μεταβλητή Firm_id δηλωτική της εταιρείας. Η σύνταξη της εντολής reshape έχει ως εξής:

```
reshape long vrA, i(Firm_id) j(Time_id)
```

Σημειώνονται τα εξής:

- Η έκφραση j(Time_id) δίδει εντολή στο STATA να δημιουργήσει μία αριθμητική τιμή δηλωτική του χρόνου με την ονομασία Time_id.
- Αν ο ερευνητής επιθυμεί να επαναφέρει τη δομή wide τότε εκτελεί την ακόλουθη εντολή:

  ```
  reshape wide
  ```

Αντιμετώπιση ακραίων τιμών (outliers)

Η ύπαρξη ακραίων τιμών μίας ή περισσότερων μεταβλητών είναι δυνατόν να επηρεάσουν τα αποτελέσματα της στατιστική ανάλυσης. Έχουν διαμορφωθεί δύο αντιλήψεις επί του τρόπου αντιμετώπισης των ακραίων τιμών. Η πρώτη διατείνεται ότι ο ερευνητής δεν πρέπει να πειράξει τις ακραίες τιμές διότι εμπεριέχουν πληροφόρηση που πρέπει να ληφθεί υπόψη από την εμπειρική ανάλυση. Η δεύτερη θεωρεί ότι οι ακραίες τιμές αποτελούν μία πηγή σφάλματος και άρα πρέπει να αντιμετωπισθούν με έναν από τους ακόλουθους τρόπους:

- Διαγραφή όλων των τιμών που ευρίσκονται στο χαμηλότερο ή υψηλότερο 5% (ή 1%) του εύρους τιμών της μεταβλητής (trimming).
- Αντικατάσταση όλων των τιμών που ευρίσκονται στο χαμηλότερο 5% (ή 1%) του εύρους τιμών της μεταβλητής με την τιμή της που αντιστοιχεί στο 5% (ή 1%) του εύρους τιμών της ΚΑΙ αντικατάσταση όλων των τιμών που ευρίσκονται στο υψηλότερο 5% (ή 1%) του εύρους τιμών της μεταβλητής με την τιμή της που αντιστοιχεί στο 95% (ή 99%) του εύρους τιμών της (winsorization).
- Οποιοδήποτε άλλο τρόπο υποδεικνύει η εξειδικευμένη βιβλιογραφία.

Συνίσταται ο ερευνητής να ακολουθήσει το εξής τρόπο σκέψης αναφορικά με την αντιμετώπιση των ακραίων τιμών:

- Μελέτη της συμπεριφοράς του δείγματος με τη χρήση της περιγραφικής σταστικής ανάλυσης για να διαπιστώσει την έκταση του προβλήματος και αν θα πρέπει να προβεί σε αντιμετώπιση του θέματος των ακραίων τιμών

ή να τις αγνοήσει. Η εκτέλεση της περιγραφικής στατιστικής ανάλυσης περιγράφεται σε επόμενες παραγράφους του χρηστικού οδηγού.

- Προσεκτική μελέτη της προηγουμένης ερευνητικής βιβλιογραφίας προκειμένου να διαπιστωθούν ο τρόπος με τον οποίο η διεθνή βιβλιογραφία αντιμετωπίζει το ζήτημα των ακραίων τιμών.

Αρχικά θα εξετασθεί η διαδικασία winsorization. Αρχικά θα πρέπει να εγκατασταθεί η εντολή winsor2 στο **STATA**. Αυτό επιτυγχάνεται με την ακόλουθη εντολή:

```
ssc install winsor2
```

Εν συνεχεία, αν έχει ληφθεί απόφαση για winsorization με όριο το 1% και 99% της κατανομής των τιμών της αριθμητικής τιμής της vr1, η σύνταξη της εντολής winsor2 έχει ως εξής:

```
winsor2 vr1, cuts (1 99) suffix(_new)
```

Η επιλογή suffix(_new) κατευθύνει το **STATA** να δημιουργήσει μία νέα αριθμητική μεταβλητή με προέκταση ονόματος το _new (στο παράδειγμα μας θα δημιουργηθεί η μεταβλητή vr1_new από την αρχική μεταβλητή vr1 και την προέκταση _new). Επίσης, σημειώνεται ότι ο ερευνητής μπορεί να θέσει άλλα όρια στην κατανομή της αριθμητικής μεταβλητής που θα εκτελεσθεί η διαδικασία winsorization. Για παράδειγμα μπορεί να τεθεί το κάτω όριο στο 7% και το πάνω όριο στο 98%. Τότε η σύνταξη της εντολής winsor2 έχει ως εξής:

```
winsor2 vr1, cuts (9 98) suffix(_new)
```

Τέλος ο ερευνητής μπορεί να επιθυμεί να μην εκτελείτε η διαδικασία winsorization αλλά να γίνεται διαγραφή όλων των τιμών που ευρίσκονται στο χαμηλότερο ή υψηλότερο 5% (ή 1%) του εύρους τιμών της μεταβλητής (trimming). Σε αυτή την περίπτωση προστίθεται στην εντολή winsor2 η επιλογή trim, δηλαδή η σύνταξη της εντολής για trimming της μεταβλητής vr1 διαμορφώνεται ως εξής:

```
winsor2 vr1, cuts (1 99) suffix(_new)  trim
```

ΓΕΝΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ ΑΝΑΦΟΡΙΚΑ ΜΕ ΤΗΝ ΠΡΟΕΤΟΙΜΑΣΙΑ ΤΩΝ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΣΤΑΤΙΣΤΙΚΗ ΑΝΑΛΥΣΗ

- Στην ενότητα 2.2. παρουσιάσθηκαν οι επιμέρους διαδικασίες προετοιμασία των δεδομένων για επεξεργασία στο **STATA**. Είναι εμφανές ότι κατά την εκτέλεση των παραπάνω διαδικασιών και για κάθε διαδικασία είναι πιθανόν να απορρίπτονται παρατηρήσεις από το αρχικό δείγμα και άρα το μέγεθος του να μειώνεται σταδιακά. Συνίσταται ο ερευνητής να καταγράφει και να παρακολουθεί αναλυτικά τον αριθμό των παρατηρήσεων που απορρίπτονται για κάθε μία διαδικασία για τους ακόλουθους λόγους:
    - Δύναται να προκύψει μία δραματική πτώση του αριθμού των παρατηρήσεων. Τούτο μπορεί να οφείλεται είτε σε σφάλμα του ερευνητή και άρα θα πρέπει να διορθωθεί είτε εκ της φύσεως της διαδικασίας

και κατά συνέπεια ενδέχεται ο ερευνητής να προβεί σε εναλλακτικό (αλλά μεθοδολογικά ορθό) σχεδιασμό προετοιμασίας δεδομένων.

- Είναι δυνατόν να ζητηθεί κατά την αξιολόγηση της διπλωματικής εργασίας πληροφορίες αναφορικά με το αρχικό δείγμα των παρατηρήσεων και το τρόπο με τον οποίο προέκυψε το τελικό δείγμα παρατηρήσεων με το οποίο εκτελέσθηκε η στατιστική ανάλυση.

- Οι διαδικασίες προετοιμασίας των δεδομένων για επεξεργασία στο **STATA** που περιγράφηκαν στην ενότητα αυτή ενδέχεται να εμπλουτισθούν με πρόσθετες διαδικασίες προετοιμασίας δεδομένων υπό το πρίσμα ότι η ειδικότερη ερευνητική περιοχή στην οποία λαμβάνει χώρα η εκπόνηση της διπλωματικής εργασίας έχει διαμορφώσει μία πάγια ερευνητική προσέγγιση επί του σχετικού θέματος.
- Το τελικά διαμορφωμένο δείγμα παρατηρήσεων θα πρέπει να αποθηκευτεί με τη μορφή αρχείου δεδομένων **STATA** δηλαδή αποθηκευτεί με την προέκταση ονόματος .dta. Συνίσταται, οι περαιτέρω εργασίες να εκτελούνται με αντίγραφο του αρχείου αυτού για λόγους ασφάλειας και ακεραιότητας των δεδομένων.

2.3. Διαχείριση δεδομένων στο περιβάλλον εργασίας **STATA**

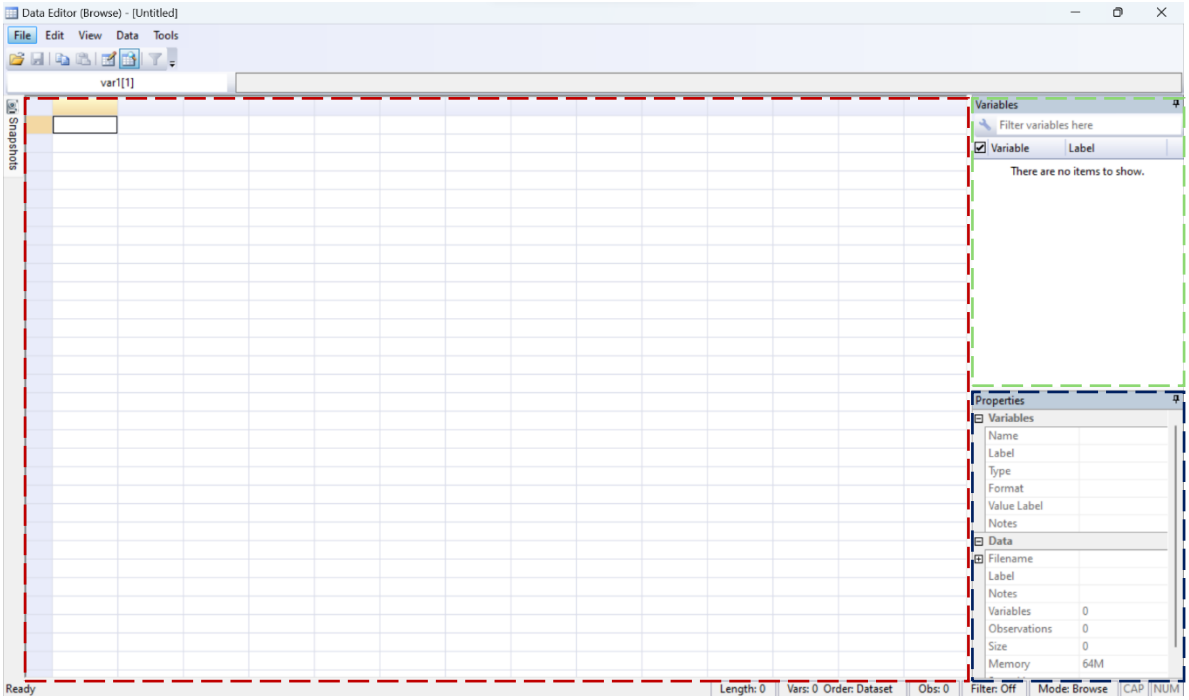
Πληκτρολογήστε browse στο παράθυρο (πεδίο) με επικεφαλίδα Command και με τρόπο αυτό μπορείτε να μεταβείτε στο παράθυρο (πεδίο) με επικεφαλίδα Data Editor, δηλαδή στο περιβάλλον περιήγησης δεδομένων του STATA. Εναλλακτικά, το παραπάνω μπορεί να επιτευχθεί αν ο χρήστης κάνει κλικ στη βασική γραμμή εργαλείων (toolbar) το ένατο κουμπί που δίδει τη δυνατότητα περιήγησης στα δεδομένων ενός ενεργού αρχείου δεδομένων (.dta). Το γράφημα του ένατου κουμπιού φαίνεται στην επόμενη γραμμή.

Σημειώνεται ότι ο χρήστης μπορεί να επιλέξει το όγδοο κουμπί στη βασική γραμμή εργαλείων (toolbar) και να μεταβεί στο παράθυρο (πεδίο) με επικεφαλίδα Data Editor, αλλά σε αυτή την περίπτωση έχει τη δυνατότητα να πραγματοποιήσει και επεξεργασία των δεδομένων. Το γράφημα του όγδοου κουμπιού φαίνεται στην επόμενη γραμμή.

Το παράθυρο (πεδίο) με επικεφαλίδα Data Editor εμφανίζεται στην Εικόνα 8. Από την Εικόνα 8 είναι εμφανές ότι η βασική δομή του παράθυρου (πεδίου) με επικεφαλίδα Data Editor περιλαμβάνει κάποια επιμέρους πεδία:

- Το πεδίο οπτικοποίησης των μεταβλητών και των αριθμητικών τιμών τους το οποίο έχει επισημανθεί στην Εικόνα 8 με ένα κόκκινο πλαίσιο. Επειδή έχει ανοιχθεί ένα κενό από δεδομένα αρχείο δεν εμφανίζεται καμία παρατήρηση.
- Το πεδίο οπτικοποίησης των μεταβλητών με επικεφαλίδα Μεταβλητές (Variables) το οποίο έχει επισημανθεί στην Εικόνα 8 με ένα πράσινο πλαίσιο.
- Το πεδίο οπτικοποίησης των ιδιοτήτων των μεταβλητών με επικεφαλίδα Ιδιότητες (Properties) το οποίο έχει επισημανθεί στην Εικόνα 8 με ένα μπλε πλαίσιο.

Εικόνα 9: Το Περιβάλλον Επεξεργασίας - Περιήγησης Δεδομένων (Data Editor - Browse) του STATA



Όταν περιηγείστε στα δεδομένα σας, μπορεί να συνειδητοποιήσετε ότι έχετε εισαγάγει κάτι λάθος. Κάντε κλικ στο κουμπί Επεξεργασία (Edit) στη γραμμή εργαλείων του παραθύρου του προγράμματος περιήγησης δεδομένων για να μεταβείτε στη λειτουργία επεξεργασίας. Εάν η Stata σας ρωτήσει εάν είστε βέβαιοι ότι θέλετε να αποχωρήσετε από τη λειτουργία περιήγησης, κάντε κλικ στο "Yes - Ναι".

Το γράφημα του κουμπιού Επεξεργασία (Edit) φαίνεται στην επόμενη γραμμή.

Σημειώστε το παράθυρο Ιδιότητες στο πρόγραμμα περιήγησης δεδομένων – χρησιμοποιήστε αυτό αντί για το παράθυρο Ιδιότητες στην κύρια διάταξη **STATA**

κατά την επεξεργασία των μεταβλητών σας στο πρόγραμμα περιήγησης δεδομένων.

Για να προσθέσετε μια ετικέτα (label) μεταβλητής, επιλέξτε πρώτα οποιοδήποτε κελί στη στήλη της μεταβλητής που θέλετε να αλλάξετε. Στη συνέχεια, κάντε διπλό κλικ στο κελί στα δεξιά του κελιού Label στο παράθυρο Ιδιότητες του προγράμματος περιήγησης δεδομένων, πληκτρολογήστε τι θέλετε να είναι γνωστή η μεταβλητή και πατήστε enter. Παρόμοια διαδικασία μπορεί να ακολουθηθεί και για την ετικέτα των δεδομένων της μεταβλητής.

Σημειώνονται τα ακόλουθα:

- Τα δεδομένα δύναται να αναφέρονται σε δεδομένα μεταβλητής συμβολοσειράς (string variable) ή σε δεδομένα αριθμητικής μεταβλητής (numeric). Τα δεδομένα μεταβλητής συμβολοσειράς εμπεριέχουν σύμβολα, διάφορους χαρακτήρες, αριθμητικές τιμές, κ.λπ. Τα δεδομένα αριθμητικής μεταβλητής εμπεριέχουν μονάχα αριθμητικές τιμές.
- Τα αριθμητικά δεδομένα (συμπεριλαμβανομένων των μεταβλητών ημερομηνίας) μπορούν να διατίθενται σε διάφορα μεγέθη. Υπάρχουν μεταβλητές που αποτελούνται μόνο από τους αριθμούς 0 και 1. Άλλες μεταβλητές εκφράζονται σε ακέραιους αριθμούς ή σε φυσικούς αριθμούς. Μερικές μπορεί να έχουν και δεκαδικό μέρος. Οι βασικοί τύποι αριθμητικών δεδομένων που εμφανίζονται στο STATA είναι οι ακόλουθοι:
    - byte: ακέραιος αριθμός που λαμβάνει τιμές από -127 έως 100.
    - int: ακέραιος αριθμός που λαμβάνει τιμές από -32,767 έως 32,740.
    - long: ακέραιος αριθμός που λαμβάνει τιμές από -2,147,483,647 έως 2,147,483,620.
    - float: πραγματικός αριθμός με ακρίβεια 8 δεκαδικών ψηφίων.
    - double: πραγματικός αριθμός με ακρίβεια 16 δεκαδικών ψηφίων.

2.4. Διαχείριση μεταβλητών στο περιβάλλον εργασίας **STATA**

Εντολή generate

Είναι δυνατόν να δημιουργήσετε μία νέα μεταβλητή στο **STATA** αξιοποιώντας την εντολή generate. Η εντολή generate μας επιτρέπει να δημιουργήσουμε μία νέα μεταβλητή ως αποτέλεσμα μίας μαθηματικής έκφρασης ή οποία μπορεί να συμπεριλαμβάνει ήδη ορισμένες μεταβλητές. Η γενική σύνταξη της εντολής generate στο παράθυρο Command έχει ως εξής:

```
generate ονομασία μεταβλητής = μαθηματική έκφραση
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής, Ακολουθούν διάφορα υποθετικά παραδείγματα:

```
generate  x = 5
generate y = 4*15
generate z = y/x
```

Τότε θα δημιουργηθούν οι ακόλουθες περιπτώσεις (υποθέτουμε ότι αναφερόμαστε σε πέντε περιπτώσεις):

x	y	z
5	60	12
5	60	12
5	60	12
5	60	12
5	60	12

Παρατηρείται ότι οι μεταβλητές x και y δημιουργούνται ως από αμιγώς αριθμητικές πράξεις ενώ η μεταβλητή z ως μαθηματική (συναρτησιακή) έκφραση των προηγούμενων μεταβλητών.

Σημειώνονται τα ακόλουθα:

- Η εντολή generate δύναται να συναχθεί με επιλογές που αφορούν συνθήκες υπό τις οποίες εκτελείται (θυμηθείτε από προηγούμενη ανάλυση (if, &, |).
- Αν θέσετε την νέα μεταβλητή ίση με _n θα δημιουργήσετε μία μεταβλητή που θα δίνει αύξων αριθμό στις παρατηρήσεις σας.
- Μπορείτε να δημιουργήσετε τις χρονικές υστερήσεις μίας μεταβλητής χρησιμοποιώντας την μαθηματική έκφραση _n. Για παράδειγμα θέλετε να δημιουργήσετε μία νέα μεταβλητή με όνομα lag1_vr1 η οποία να περιέχει την πρώτη χρονική υστέρηση της υπάρχουσας μεταβλητής vr1:

  ```
  generate lag_1vr1=vr1[_n-1]
  ```

  Εναλλακτικά χρησιμοποιείται τον τελεστή L1. Τότε η σύνταξη της εντολής θα είναι:

  ```
  generate lag_1vr1=L1.vr1
  ```

- Γενικά για να δημιουργήσετε την K υστέρηση, δημιουργείται μία νέα μεταβλητή με όνομα lagK_vr1 η οποία να περιέχει την K χρονική υστέρηση της υπάρχουσας μεταβλητής vr1:

  ```
  generate lag_Kvr1=vr1[_n-K]
  ```

  Εναλλακτικά χρησιμοποιείται τον τελεστή LK. Τότε η σύνταξη της εντολής θα είναι:

  ```
  generate lag_1vr1=LK.vr1
  ```

- Μπορείτε να δημιουργήσετε νέες μεταβλητές που να αντιστοιχούν στις μελλοντικές τιμές μίας προυπάρχουσας. Δημιουργείται μία νέα μεταβλητή με όνομα for_Kvr1 η οποία να περιέχει την K μελλοντική τιμή μίας υπάρχουσας μεταβλητής vr1:

  ```
  generate for_Kvr1=vr1[_n-K]
  ```

Εναλλακτικά χρησιμοποιείται τον τελεστή FK. Τότε η σύνταξη της εντολής θα είναι:

```
generate lag_1vr1=FK.vr1
```

- Αν θέσετε την νέα μεταβλητή ίση με _N θα δημιουργήσετε μία μεταβλητή με μοναδική τιμή το σύνολο των παρατηρήσεων του δείγματός σας.

Εντολή egen

Η εντολή egen στο **STATA** δίνει δυναμικές δυνατότητες στην δημιουργία και διαχείριση των μεταβλητών. Επειδή η παρουσίαση της υπερβαίνει του σκοπούς του χρηστικού οδηγού δεν θα αναλυθεί αλλά στο Παράρτημα Α δίδεται ο επίσημος οδηγός σύνταξης της εντολής.

3. Περιγραφική στατιστική ανάλυση στο STATA

3.1. Αναφορές περιγραφικής στατιστικής στο **STATA**

Εντολή describe

Τα περιγραφικά στατιστικά στοιχεία είναι ζωτικής σημασίας για την κατανόηση της φύσης των δεδομένων σας. Παρέχει μια βασική περιγραφή των δεδομένων σας και σας επιτρέπει να εξερευνήσετε τις μορφές ("μορφή εμφάνισης") των μεταβλητών. Μία βασική εντολή στο **STATA** για τη λήψη περιγραφικών στατιστικών είναι η describe. Στο παράθυρο Command πληκτρολογήστε την ακόλουθη εντολή:

```
describe
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο δεδομένων εμφανίζονται τα ακόλουθα στο παράθυρο Results:

```
Contains data from C:\Program Files\Stata18\ado\base/n/nlsw88.dta
 Observations:         2,246                  NLSW, 1988 extract
    Variables:            17                  1 May 2022 22:52
                                              (_dta has notes)
-------------------------------------------------------------------------------
-------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
-------------------
idcode          int     %8.0g                 NLS ID
age             byte    %8.0g                 Age in current year
race            byte    %8.0g      racelbl    Race
married         byte    %8.0g      marlbl     Married
never_married   byte    %16.0g     nev_mar    Never married
grade           byte    %8.0g                 Current grade completed
collgrad        byte    %16.0g     gradlbl    College graduate
south           byte    %9.0g      southlbl   Lives in the south
smsa            byte    %9.0g      smsalbl    Lives in SMSA
c_city          byte    %16.0g     ccitylbl   Lives in a central city
industry        byte    %23.0g     indlbl     Industry
occupation      byte    %22.0g     occlbl     Occupation
union           byte    %8.0g      unionlbl   Union worker
wage            float   %9.0g                 Hourly wage
hours           byte    %8.0g                 Usual hours worked
ttl_exp         float   %9.0g                 Total work experience (years)
tenure          float   %9.0g                 Job tenure (years)
-------------------------------------------------------------------------------
```

Η ερμηνεία των παραπάνω έχει ως εξής:

- Ο παραπάνω πίνακας παρέχει μια περίληψη ολόκληρου του συνόλου δεδομένων. Για παράδειγμα, το Variables: 17 υποδηλώνει ότι υπάρχουν 17 μεταβλητές στο σύνολο δεδομένων.
- Ο τύπος αποθήκευσης μας βοηθά να κατανοήσουμε τον τύπο δεδομένων μιας μεταβλητής. Οι μεταβλητές στο STATA μπορούν να έχουν διαφορετικούς τύπους δεδομένων, όπως int, byte, float, double κ.λπ. Το int υποδεικνύει ότι η μεταβλητή είναι ακέραιου τύπου δεδομένων (ακέραιος είναι ένας ακέραιος αριθμός που δεν έχει δεκαδικό ή κλασματικό στοιχείο).

Ο τύπος δεδομένων byte υποδεικνύει ότι η μεταβλητή αποθηκεύεται ως ακέραιος αριθμός εντός περιορισμένου εύρους (χρησιμοποιούνται συνήθως όταν έχετε κατηγορικές ή τακτικές μεταβλητές με μικρό αριθμό διακριτών τιμών). Το float υποδηλώνει ότι η μεταβλητή αποθηκεύεται ως αριθμοί κινητής υποδιαστολής με δεκαδικούς. Το Str υποδηλώνει ότι η μεταβλητή είναι αποθηκευμένη ως συμβολοσειρά (κείμενο). Ο διπλός τύπος δεδομένων αναφέρεται σε μια μεταβλητή που αποθηκεύει αριθμητικές τιμές τόσο ως ακέραιους όσο και ως κλασματικές τιμές με υψηλό βαθμό ακρίβειας.

- Η μορφή εμφάνισης αναφέρεται στη μορφή στην οποία εμφανίζονται οι αριθμητικές τιμές όταν προβάλλουμε ή εκτυπώνουμε τα δεδομένα στο Stata. Για παράδειγμα, το %9,0g στη μεταβλητή wage υποδεικνύει ότι όταν προβάλλετε ή εκτυπώνετε τις τιμές "wage", θα εμφανίζονται με συνολικό πλάτος 9 χαρακτήρων. Το "%g" υποδεικνύει ότι είναι μια γενική μορφή, η οποία είναι μια ευέλικτη μορφή που εμφανίζει τις τιμές με συμπαγή και ευανάγνωστο τρόπο.

Εντολή summarize

Για να λάβουμε διάφορα στατιστικά μεγέθη στο **STATA**, μπορούμε να χρησιμοποιήσουμε την εντολή summarize ή su (νεότερες εκδόσεις του **STATA**. Στο παράθυρο Command πληκτρολογήστε την ακόλουθη εντολή:

```
summarize
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο δεδομένων εμφανίζονται τα ακόλουθα στο παράθυρο Results:

Variable	Obs	Mean	Std. dev.	Min	Max
idcode	2,246	2612.654	1480.864	1	5159
age	2,246	39.15316	3.060002	34	46
race	2,246	1.282725	.4754413	1	3
married	2,246	.6420303	.4795099	0	1
never_marr~d	2,246	.1041852	.3055687	0	1
grade	2,244	13.09893	2.521246	0	18
collgrad	2,246	.2368655	.4252538	0	1
south	2,246	.4194123	.4935728	0	1
smsa	2,246	.7039181	.4566292	0	1
c_city	2,246	.2916296	.4546139	0	1
industry	2,232	8.189516	3.010875	1	12
occupation	2,237	4.642825	3.408897	1	13
union	1,878	.2454739	.4304825	0	1
wage	2,246	7.766949	5.755523	1.004952	40.74659
hours	2,242	37.21811	10.50914	1	80
ttl_exp	2,246	12.53498	4.610208	.1153846	28.88461
tenure	2,231	5.97785	5.510331	0	25.91667

Για κάθε μία μεταβλητή του δείγματος δίδεται ο αριθμός των παρατηρήσεων (Obs) ο μέσος όρος (Mean), η τυπική απόκλιση (Std. dev.), το ελάχιστο (Min) και το μέγιστο (max) της.

Μπορούμε να λάβουμε μία πιο αναλυτική εικόνα της περιγραφικής στατιστικής των μεταβλητών μας αν στην εντολή summarize προσθέσουμε την παράμετρο detail. Στο παράθυρο Command πληκτρολογήστε την ακόλουθη εντολή:

```
summarize, detail
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο δεδομένων εμφανίζονται τα ακόλουθα στο παράθυρο Results:

```
                        NLS ID
-------------------------------------------------------------
      Percentiles      Smallest
 1%           45              1
 5%          269              2
10%          521              3       Obs               2,246
25%         1366              4       Sum of wgt.       2,246

50%         2614                      Mean           2612.654
                       Largest       Std. dev.      1480.864
75%         3903          5154
90%         4651          5156       Variance        2192957
95%         4925          5157       Skewness       -.0232308
99%         5104          5159       Kurtosis        1.828053

                    Age in current year
-------------------------------------------------------------
      Percentiles      Smallest
 1%           34             34
 5%           35             34
10%           35             34       Obs               2,246
25%           36             34       Sum of wgt.       2,246

50%           39                      Mean            39.15316
                       Largest       Std. dev.      3.060002
75%           42             45
90%           44             45       Variance       9.363614
95%           44             46       Skewness       .2003234
99%           45             46       Kurtosis       1.932389

                           Race
-------------------------------------------------------------
      Percentiles      Smallest
 1%            1              1
 5%            1              1
10%            1              1       Obs               2,246
25%            1              1       Sum of wgt.       2,246

50%            1                      Mean            1.282725
                       Largest       Std. dev.      .4754413
75%            2              3
90%            2              3       Variance       .2260444
95%            2              3       Skewness       1.284394
99%            3              3       Kurtosis       3.409155
```

Είναι εμφανές ότι παραπάνω εντολές (summarize και describe) δύναται να αφορούν συγκεκριμένες μεταβλητές. Έστω ότι εστιάζουμε σε τρεις μεταβλητές την vr1, vr2, και την vr3 τότε η σύνταξη των εντολών στο **STATA** θα ήταν:

```
describe/summarize vr1 vr2 vr3
```

Σημειώνεται ότι ο χρήστης του **STATA** μπορεί να εξειδικεύσει τον τρόπο που θα εκτελεσθούν οι άνω εντολές (όπως άλλωστε και όλες οι εντολές στο **STATA**) κάνοντας χρήση των τελεστών (if/by). Για παράδειγμα, αν στα δεδομένα μας υπάρχει μία μεταβλητή με ονομασία vrCAT η οποία κατηγοριοποιεί τα δεδομένα μας σε πέντε μεγάλες κατηγορίες με βάση κάποιο κριτήριο (όπως για παράδειγμα το μέγεθος της εταιρείας σε πολύ μικρή, μικρή, μεσαία, μεγάλη και πολύ μεγάλη με δηλωτικές αριθμητικές τιμές 1, 2, 3, 4 και 5 αντίστοιχα). Με την επιλογή by δύναται να λάβουμε τα περιγραφικά στατιστικά μεταβλητών (άρα εντολή summarize) για κάθε μία κατηγορία:

```
by vrCAT, sort: summarize
```

Εναλλακτικά μπορεί να επιθυμούσαμε τα περιγραφικά στατιστικά για δύο κατηγορίες με δηλωτικές τιμές της μεταβλητής vrCAT να είναι η 1 και η 4. Σε αυτή την περίπτωση μπορούμε να χρησιμοποιήσουμε τον τελεστή if:

```
summarize if vrCAT==1 & vrCAT==4
```

Εννοείται ότι τα παραπάνω μπορούν να εκτελεσθούν για συγκεκριμένες μεταβλητές.

Εντολή codebook

Η εντολή codebook στο **STATA** είναι ένα πολύτιμο εργαλείο για τη λήψη λεπτομερών πληροφοριών σχετικά με τις μεταβλητές σε ένα σύνολο δεδομένων. Παρέχει πληροφορίες για ονόματα μεταβλητών, ετικέτες τιμών, τύπους δεδομένων, συνοπτικά στατιστικά στοιχεία και άλλες σχετικές λεπτομέρειες.

Στο παράθυρο Command πληκτρολογήστε την ακόλουθη εντολή:

```
codebook
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο δεδομένων και για μία μεταβλητή εμφανίζονται τα ακόλουθα στο παράθυρο Results:

```
-------------------------------------------------------------------------------
wage                                                          Hourly wage
-------------------------------------------------------------------------------

                  Type: Numeric (float)

                 Range: [1.0049518,40.74659]          Units: 1.000e-07
        Unique values: 967                        Missing .: 0/2,246

                  Mean: 7.76695
             Std. dev.: 5.75552

           Percentiles:     10%       25%       50%       75%       90%
                         3.22061   4.25926   6.27227   9.59742   12.7778
```

Σημειώνονται τα ακόλουθα:

- Η εντολή codebook χωρίς εξειδίκευση εκτελείται για το σύνολο των μεταβλητών και όταν τεθούν ονόματα μεταβλητών μετά από αυτήν τότε εκτελείται για τις συγκεκριμένες μεταβλητές.

- Η εντολή codebook δύναται να εκτελεσθεί σε συνδυασμό με τελεστές (if, and, or και by).
- Ειδικότερη περίπτωση είναι ο συνδυασμός της εντολής codebook με την επιλογή compact οπότε και δίνονται συμπυκνωμένα αποτελέσματα. Η σύνταξη έχει ως εξής:

```
Codebook, compact
```

Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο δεδομένων και για μία μεταβλητή εμφανίζονται τα ακόλουθα στο παράθυρο Results:

```
Variable       Obs Unique      Mean       Min       Max  Label
-------------------------------------------------------------------------------
idcode        2246   2246  2612.654         1      5159  NLS ID
age           2246     13  39.15316        34        46  Age in current year
race          2246      3  1.282725         1         3  Race
married       2246      2  .6420303         0         1  Married
never_marr~d  2246      2  .1041852         0         1  Never married
grade         2244     16  13.09893         0        18  Current grade completed
collgrad      2246      2  .2368655         0         1  College graduate
south         2246      2  .4194123         0         1  Lives in the south
smsa          2246      2  .7039181         0         1  Lives in SMSA
c_city        2246      2  .2916296         0         1  Lives in a central city
industry      2232     12  8.189516         1        12  Industry
occupation    2237     13  4.642825         1        13  Occupation
union         1878      2  .2454739         0         1  Union worker
wage          2246    967  7.766949  1.004952  40.74659  Hourly wage
hours         2242     62  37.21811         1        80  Usual hours worked
ttl_exp       2246   1546  12.53498  .1153846  28.88461  Total work experience (years)
tenure        2231    259   5.97785         0  25.91667  Job tenure (years)
-------------------------------------------------------------------------------
```

3.2. Αναφορές συχνοτήτων στο **STATA**

Ένας πίνακας συχνοτήτων δείχνει την κατανομή των παρατηρήσεων με βάση τις επιλογές σε μια μεταβλητή. Οι πίνακες συχνότητας είναι χρήσιμοι για να κατανοήσουμε ποιες επιλογές εμφανίζονται περισσότερο ή λιγότερο συχνά στο σύνολο δεδομένων. Αυτό είναι χρήσιμο για να κατανοήσετε καλύτερα κάθε μεταβλητή και να αποφασίσετε εάν οι μεταβλητές πρέπει να επανακωδικοποιηθούν ή όχι.

Εντολή tab

Η εντολή tab δίνει το πίνακα συχνοτήτων μίας μεταβλητής δηλαδή την συχνότητα (πλήθος) εμφανίσεων της κάθε μίας τιμής της. Η σύνταξη της εντολής tab γίνεται με αναφορά στην μεταβλητή που μας ενδιαφέρει και εκτελείται για μία μεταβλητή. Έστω, ότι εστιάζουμε στη μεταβλητή vr1 τότε η σύνταξη της εντολής tab στο παράθυρο Command του **STATA** θα ήταν:

```
tab vr1
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο

δεδομένων και για μία μεταβλητή (married = παντρεμένος ή όχι) εμφανίζονται τα ακόλουθα στο παράθυρο Results:

```
    Married |      Freq.     Percent        Cum.
------------+-----------------------------------
     Single |        804       35.80       35.80
    Married |      1,442       64.20      100.00
------------+-----------------------------------
      Total |      2,246      100.00
```

Ο παραπάνω πίνακας δείχνει ότι 804 άτομα στο σύνολο δεδομένων είναι άγαμοι, που αποτελεί το 35,80% της συνολικής παρατήρησης. Ομοίως, ο πίνακας δείχνει επίσης ότι 1.442 άτομα στο σύνολο δεδομένων είναι παντρεμένα, που αποτελεί το 64,20% της συνολικής παρατήρησης.

Αν επιθυμούμε και τη δημιουργία γραφήματος συχνοτήτων, τότε προσθέτουμε την επιλογή plot sort. Δηλαδή η σύνταξη της εντολής tab στο παράθυρο Command του **STATA** θα ήταν:

```
tab vr1, plot sort
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο δεδομένων και για μία μεταβλητή (occupation = επάγγελμα) εμφανίζονται τα ακόλουθα στο παράθυρο Results:

```
            Occupation |      Freq.
-----------------------+------------+-----------------------------------------
                 Sales |        726 |****************************************
Professional/Technical |        317 |*****************
              Laborers |        286 |****************
        Managers/Admin |        264 |**************
             Operatives |        246 |*************
                 Other |        187 |**********
    Clerical/Unskilled |        102 |******
              Craftsmen |         53 |***
              Transport |         28 |**
                Service |         16 |*
          Farm laborers |          9 |*
     Household workers |          2 |
                Farmers |          1 |
-----------------------+------------+-----------------------------------------
                  Total |      2,237
```

Η εντολή tab όταν εκτελεσθεί για δύο μεταβλητές μας δίδει τη δυνατότητα να εξετάσουμε τη συνδυαστική κατανομή συχνοτήτων (crosstab). Η συνδυαστική κατανομή συχνοτήτων είναι χρήσιμη αν θέλουμε να λάβουμε την κοινή κατανομή δύο μεταβλητών σε ένα σύνολο δεδομένων. Για να επιτευχθεί αυτό μετά την εντολή tab πληκτρολογείτε την ονομασία δύο μεταβλητών (έστω των μεταβλητών vr1 και vr2) λάβετε τη διασταύρωση των κατηγορικών μεταβλητών στο παράθυρο Command του **STATA:**

```
tab vr1 vr2
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο

δεδομένων και για δύο μεταβλητές (race/colleqe graduate) εμφανίζονται τα ακόλουθα στο παράθυρο Results:

College graduate	Race: White	Black	Other	Total
Not college grad	1,217	480	17	1,714
College grad	420	103	9	532
Total	1,637	583	26	2,246

Σημειώνονται τα ακόλουθα:

- Αν επιθυμούμε να λάβουμε την συνδυαστική κατανομή συχνοτήτων σε όρους ποσοστών και όχι σε απόλυτα μεγέθη χρησιμοποιείται στην εντολή tab αξιοποιείται η επιλογή row nofreq και η σύνταξη έχει ως εξής:

  ```
  tab vr1 vr2, row nofreq
  ```

  Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο δεδομένων και για δύο μεταβλητές (race/colleqe graduate) εμφανίζονται τα ακόλουθα στο παράθυρο Results:

College graduate	Race: White	Black	Other	Total
Not college grad	71.00	28.00	0.99	100.00
College grad	78.95	19.36	1.69	100.00
Total	72.89	25.96	1.16	100.00

- Αν επιθυμούμε να λάβουμε την συνδυαστική κατανομή συχνοτήτων σε όρους ποσοστών και σε απόλυτα μεγέθη χρησιμοποιείται στην εντολή tab αξιοποιείται η επιλογή row και η σύνταξη έχει ως εξής:

  ```
  tab vr1 vr2, row
  ```

  Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο δεδομένων και για δύο μεταβλητές (race/colleqe graduate) εμφανίζονται τα ακόλουθα στο παράθυρο Results:

Key
frequency
row percentage

College graduate	Race: White	Black	Other	Total
Not college grad	1,217	480	17	1,714
	71.00	28.00	0.99	100.00
College grad	420	103	9	532
	78.95	19.36	1.69	100.00
Total	1,637	583	26	2,246
	72.89	25.96	1.16	100.00

- Ο παραπάνω πίνακας αναφέρει ποσοστό για σειρές. Αν θέλουμε να πάρουμε το ποσοστό στήλης αντί για γραμμές, πληκτρολογούμε:

  ```
  tab vr1 vr2, column
  ```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο δεδομένων και για δύο μεταβλητές (race/colleqe graduate) εμφανίζονται τα ακόλουθα στο παράθυρο Results:

```
+-------------------+
| Key               |
|-------------------|
|     frequency     |
| column percentage |
+-------------------+

                |               Race
College graduate |     White      Black      Other |     Total
-----------------+---------------------------------+----------
Not college grad |     1,217        480         17 |     1,714
                 |     74.34      82.33      65.38 |     76.31
-----------------+---------------------------------+----------
    College grad |       420        103          9 |       532
                 |     25.66      17.67      34.62 |     23.69
-----------------+---------------------------------+----------
           Total |     1,637        583         26 |     2,246
                 |    100.00     100.00     100.00 |    100.00
```

- Ο παραπάνω πίνακας αναφέρει ποσοστό για σειρές. Αν θέλουμε να πάρουμε το ποσοστό στήλης καιγραμμές, πληκτρολογούμε:

```
tab vr1 vr2, col row
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο δεδομένων και για δύο μεταβλητές (race/colleqe graduate) εμφανίζονται τα ακόλουθα στο παράθυρο Results:

```
+--------------------+
| Key                |
|--------------------|
|     frequency      |
|  row percentage    |
| column percentage  |
+--------------------+

                |               Race
College graduate |     White      Black      Other |     Total
-----------------+---------------------------------+----------
Not college grad |     1,217        480         17 |     1,714
                 |     71.00      28.00       0.99 |    100.00
                 |     74.34      82.33      65.38 |     76.31
-----------------+---------------------------------+----------
    College grad |       420        103          9 |       532
                 |     78.95      19.36       1.69 |    100.00
                 |     25.66      17.67      34.62 |     23.69
-----------------+---------------------------------+----------
           Total |     1,637        583         26 |     2,246
                 |     72.89      25.96       1.16 |    100.00
                 |    100.00     100.00     100.00 |    100.00
```

Εντολή tab1

Αν θέλουμε διακριτά τους πίνακες συχνοτήτων για παραπάνω από μία μεταβλητές τότε χρησιμοποιείται η εντολή tab1 αντί της εντολής tab. Έστω ότι εστιάζουμε σε τρεις μεταβλητές την vr1, vr2, και την vr3 τότε η σύνταξη των εντολών στο **STATA** θα ήταν:

```
tab1 vr1 vr2 vr3
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο δεδομένων και για τρεις μεταβλητές (married/south/race) εμφανίζονται τα ακόλουθα στο παράθυρο Results:

```
-> tabulation of married

    Married |      Freq.     Percent        Cum.
------------+-----------------------------------
     Single |        804       35.80       35.80
    Married |      1,442       64.20      100.00
------------+-----------------------------------
      Total |      2,246      100.00

-> tabulation of south

   Lives in |
  the south |      Freq.     Percent        Cum.
------------+-----------------------------------
  Not south |      1,304       58.06       58.06
      South |        942       41.94      100.00
------------+-----------------------------------
      Total |      2,246      100.00

-> tabulation of race

       Race |      Freq.     Percent        Cum.
------------+-----------------------------------
      White |      1,637       72.89       72.89
      Black |        583       25.96       98.84
      Other |         26        1.16      100.00
------------+-----------------------------------
      Total |      2,246      100.00
```

Εντολή tabstat

Το tabstat είναι μια άλλη εντολή που παρέχει συνοπτικά στατιστικά στοιχεία. Η γενική σύνταξη της εντολής tabstat περιλαμβάνει τα ονόματα των μεταβλητών ενδιαφέροντος (έστω τρεις μεταβλητές: vr1, vr2 και vr3) και την πρόταση s() όπου μέσα στην παρένθεση αναφέρονται τα στατιστικά μέτρα που μας ενδιαφέρουν [έστω mean (μέσος όρος) semean (τυπικό σφάλμα το μέσου όρου) median (διάμεσος) sd (τυπική απόκλιση) var (διακύμανση) skew (λοξότητα) k (κύρτωση) count (αριθμός παρατηρήσεων) sum (άθροισμα) range (διάστημα) min (ελάχιστη τιμή) max (μέγιστη τιμή)]. Έστω ότι εστιάζουμε σε τρεις μεταβλητές την vr1, vr2, και την vr3 τότε η σύνταξη της εντολής tabstat στο **STATA** θα ήταν:

```
tabstat vr1 vr2 vr3, s(mean semean median sd var skew k count sum range min max)
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο δεδομένων και για επτά μεταβλητές (age/married/collgrad/south/c_city/union/wage) εμφανίζονται τα ακόλουθα στο παράθυρο Results:

```
   Stats |       age   married  collgrad     south    c_city     union      wage
---------+------------------------------------------------------------------------
    Mean |  39.15316  .6420303  .2368655  .4194123  .2916296  .2454739  7.766949
se(mean) |  .0645679   .010118  .0089731  .0104147  .0095926  .0099336  .1214451
     p50 |        39         1         0         0         0         0   6.27227
      SD |  3.060002  .4795099  .4252538  .4935728  .4546139  .4304825  5.755523
Variance |  9.363614  .2299298  .1808408  .2436141  .2066738  .1853151  33.12604
Skewness |  .2003234 -.5925296  1.237816  .3266212  .9168961   1.18283  3.096199
Kurtosis |  1.932389  1.351091   2.53219  1.106681  1.840698  2.399088  15.85446
       N |      2246      2246      2246      2246      2246      1878      2246
     Sum |     87938      1442       532       942       655       461  17444.57
   Range |        12         1         1         1         1         1  39.74164
     Min |        34         0         0         0         0         0  1.004952
     Max |        46         1         1         1         1         1  40.74659
------------------------------------------------------------------------------
```

Επίσης υπαρχει η δυνατότητα τα εξάγονται συνοπτικά στατιστικά στοιχεία ανά κατηγορία δεδομένων με κριτήριο τις τιμές μίας μεταβλητής. Τούτο επιτυγχάνετια

με την επιλογή by. Έστω ότι εστιάζουμε σε τρεις μεταβλητές την vr1, vr2, και την vr3 και ότι επιθυμούμε συνοπτικά στατιστικά στοιχεία ανά τιμή της μεταβλητής vr4 τότε η σύνταξη της εντολής στο **STATA** θα ήταν:

```
tabstat vr1 vr2 vr3, by (vr4) s(mean se median sd var skew k count sum range min max)
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο δεδομένων και για επτά μεταβλητές (age/married/collgrad/south/c_city/union/wage) εμφανίζονται τα ακόλουθα στο παράθυρο Results ανά κατηγορία δεδομένων με βάση τις τιμές της μεταβλητής race:

race	age	married	collgrad	south	c_city	union	wage
White	39.27245	.7025046	.2565669	.3457544	.2211362	.2232077	8.082999
	.0760678	.0113025	.0107977	.0117588	.0102605	.0113245	.1471846
	39	1	0	0	0	0	6.545891
	3.077691	.457296	.4368717	.4757589	.4151389	.4165504	5.955069
	9.472181	.2091196	.1908569	.2263466	.1723403	.1735143	35.46285
	.1514898	-.8859315	1.114778	.6486171	1.343883	1.329465	3.00474
	1.919862	1.784875	2.24273	1.420704	2.806021	2.767478	14.74577
	1637	1637	1637	1637	1637	1353	1637
	64289	1150	420	566	362	302	13231.87
	12	1	1	1	1	1	39.19313
	34	0	0	0	0	0	1.004952
	46	1	1	1	1	1	40.19808
Black	38.81132	.4699828	.1766724	.6397942	.490566	.3013972	6.844558
	.1234292	.0206883	.0158092	.0198991	.020722	.0205211	.2102342
	38	0	0	1	0	0	5.434783
	2.980246	.4995268	.3817187	.4804722	.5003403	.4593235	5.076187
	8.881865	.249527	.1457092	.2308536	.2503404	.210978	25.76767
	.3449691	.1202856	1.695517	-.5824029	.0377426	.8656266	3.516731
	2.029956	1.014469	3.874778	1.339193	1.001425	1.749309	21.15914
	583	583	583	583	583	501	583
	22627	274	103	373	286	151	3990.377
	11	1	1	1	1	1	39.59522
	34	0	0	0	0	0	1.151368
	45	1	1	1	1	1	40.74659
Other	39.30769	.6923077	.3461538	.1153846	.2692308	.3333333	8.550781
	.6367447	.0923077	.0951486	.0638971	.088712	.0982946	1.021653
	39	1	0	0	0	0	7.560383
	3.246774	.4706787	.4851645	.3258126	.4523443	.4815434	5.20943
	10.54154	.2215385	.2353846	.1061538	.2046154	.2318841	27.13816
	.0047392	-.8333333	.6467617	2.407717	1.040532	.7071068	1.428553
	1.622899	1.694444	1.418301	6.797101	2.082707	1.5	5.799663
	26	26	26	26	26	24	26
	1022	18	9	3	7	8	222.3203
	10	1	1	1	1	1	23.99913
	34	0	0	0	0	0	1.80602
	44	1	1	1	1	1	25.80515
Total	39.15316	.6420303	.2368655	.4194123	.2916296	.2454739	7.766949
	.0645679	.010118	.0089731	.0104147	.0095926	.0099336	.1214451
	39	1	0	0	0	0	6.27227
	3.060002	.4795099	.4252538	.4935728	.4546139	.4304825	5.755523
	9.363614	.2299298	.1808408	.2436141	.2066738	.1853151	33.12604
	.2003234	-.5925296	1.237816	.3266212	.9168961	1.18283	3.096199
	1.932389	1.351091	2.53219	1.106681	1.840698	2.399088	15.85446
	2246	2246	2246	2246	2246	1878	2246
	87938	1442	532	942	655	461	17444.57
	12	1	1	1	1	1	39.74164
	34	0	0	0	0	0	1.004952
	46	1	1	1	1	1	40.74659

Παρακάτω παρουσιάζονται τα διαθέσιμα στατιστικά που μπορεί να ένας χρήστης να ζητήσει στην πρόταση s(), δηλαδή εντός παρενθέσεως.

statname	Definition	*statname*	Definition
mean	mean	p1	1st percentile
count	count of nonmissing observations	p5	5th percentile
n	same as count	p10	10th percentile
sum	sum	p25	25th percentile
max	maximum	median	median (same as p50)
min	minimum	p50	50th percentile (same as median)
range	range = max − min	p75	75th percentile
sd	standard deviation	p90	90th percentile
variance	variance	p95	95th percentile
cv	coefficient of variation (sd/mean)	p99	99th percentile
semean	standard error of mean (sd/$\sqrt{n}$)	iqr	interquartile range = p75 − p25
skewness	skewness	q	equivalent to specifying p25 p50 p75
kurtosis	kurtosis		

3.3. Εξαγωγή αποτελεσμάτων περιγραφικής στατιστικής από το **STATA**

Εντολή outreg2

Η εξαγωγή αποτελεσμάτων περιγραφικής στατιστικής από το **STATA** μπορεί να γίνει με τη χρήση της εντολής outreg2. Η εντολή αυτή όμως θα πρέπει να εγκατασταθεί στο **STATA** και για το λόγο αυτό θα πρέπει να εκτελεσθεί ο ακόλουθος κώδικας στο **STATA**:

```
ssc install outreg2
```

Μετά την εκτέλεση της εντολής regress ή xtreg εκτελείται η εντολή outreg2 και η γενική σύνταξη της έχει ως εξής:

```
outreg2 using ONOMA_ΑΡΧΕΙΟΥ.doc, replace sum(log)
```

Όπου στη θέση ONOMA_ΑΡΧΕΙΟΥ δίνεται μία ονομασία με λατινικούς χαρακτήρες. Προσοχή γιατί η επιλογή replace θα αντικαταστήσει προϋπάρχουν αρχείο με το ίδιο όνομα.

Εάν θέλετε να εξαγάγετε τον πίνακα αποτελεσμάτων στο Excel, χρησιμοποιήστε την επέκταση *.xls αντί να χρησιμοποιήσετε *.doc

Αν επιθυμείτε να εκτελεσθεί η εντολή για συγκεκριμένες μεταβλητές (να εξαχθούν τα αποτελέσματα της περιγραφικής στατιστικής για επιλεγμένες μεταβλητές) τότε προσθέτετε στο τέλος της επιλογή keep () και μέσα στην παρένθεση βάζετε τα ονόματα των μεταβλητών (εστω vr1, vr2 και vr3). Η γενική σύνταξη της έχει ως εξής:

```
outreg2 using ONOMA_ΑΡΧΕΙΟΥ.doc, replace sum(log) keep(vr1 vr2 vr3)
```

Η εντολή outreg2 μπορεί να εκτελεσθεί και για κατηγορίες δεδομένων. Για παράδειγμα, αν στα δεδομένα μας υπάρχει μία μεταβλητή με ονομασία vrCAT η οποία κατηγοριοποιεί τα δεδομένα μας σε πέντε μεγάλες κατηγορίες με βάση κάποιο κριτήριο (όπως για παράδειγμα το μέγεθος της εταιρείας σε πολύ μικρή, μικρή, μεσαία, μεγάλη και πολύ μεγάλη με δηλωτικές αριθμητικές τιμές 1, 2, 3, 4 και 5

αντίστοιχα). Με την επιλογή by δύναται να λάβουμε ξεχωριστούς πίνακες με τα περιγραφικά στατιστικά μεταβλητών για κάθε μία κατηγορία δεδομένων:

```
by vrCAT, sort: outreg2 using ONOMA_ΑΡΧΕΙΟΥ.doc, replace sum(log)
```

3.4. Στατιστικός έλεγχος t-student για μέσο όρο στο **STATA**

Η κατανομή t-student, που αναπτύχθηκε πριν από περισσότερα από 100 χρόνια, χρησιμοποιείται για διάφορους σκοπούς στατιστικών. Η διαδικασία που συνήθως ονομάζεται t-test, ωστόσο, αναφέρεται σε μια δοκιμή της διαφοράς μεταξύ δύο μέσων ή αν ο μέσος όρος μία μεταβλητής είναι στατιστικά διάφορος από μία υποθετική τιμή.

Εντολή ttest

Ο στατιστικός έλεγχος t-test μπορεί να εκτελεσθεί στο πλαίσιο του STATA με τη χρήση της εντολής ttest. Ανάλογα με την επιλογή που προσάψουμε στην εντολή ο έλεγχος του μέσου όρου μπορεί να αφορά: (α) τη διαφορά μέσου όρου μεταξύ δύο ανεξάρτητων μεταβλητών που αφορούν το συνολικό δείγμα παρατηρήσεων, (β) τη διαφορά μέσου όρου της ίδιας μεταβλητής υπολογισμένοι για δύο υποσύνολα δεδομένων από το συνολικό δείγμα παρατηρήσεων, και (γ) τη διαφορά μέσου όρου μιας μεταβλητής από συγκεκριμένη υποθετική τιμή.

Έστω ότι επιθυμείτε να εξετασθεί η μέσου όρου μεταξύ δύο ανεξάρτητων μεταβλητών που αφορούν το συνολικό δείγμα παρατηρήσεων (έστω των μεταβλητών vs1 και vs2), με τη χρήση της εντολής ttest. Η σύνταξη της εντολής στο **STATA** θα ήταν:

```
ttest vr1==vr2
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής.

Ενδέχεται να μας ενδιαφέρει να ελέγξουμε διαφορά μέσου όρου της ίδιας μεταβλητής υπολογισμένοι για δύο υποσύνολα δεδομένων από το συνολικό δείγμα παρατηρήσεων. Υποθέστε ότι η μεταβλητή vrBIN είναι δυαδική και αναφέρεται στο αν υπαρχει μία ιδιότητα σε μία παρατήρηση. Τότε μπορούμε να ελέγχουμε αν ο μέσος όρος της μεταβλητής vr1 διαφοροποιείται μεταξύ δύο υποσυνόλων δεδομένων εκ των οποίων το ένα έχει την ιδιότητα και το άλλο δεν την έχει. Η σύνταξη της εντολής στο **STATA** θα ήταν:

```
ttest vr1, by vrBIN
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής.

Ενδέχεται να επιθυμούμε να εξετάσουμε αν ο μέσος όρος μίας μεταβλητής είναι ίσος ή διάφορος από μία υποθετική τιμή (έστω z). Η σύνταξη της εντολής στο **STATA** θα ήταν:

```
ttest vr1=z
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής.

Υπάρχουν μερικές επιλογές που μπορούν να προστεθούν: unequal (ή un) ενημερώνει το **STATA** ότι οι διακυμάνσεις των δύο ομάδων δεδομένων πρέπει να θεωρηθούν ως άνισες. Το welch (ή w) ζητά από το **STATA** να χρησιμοποιήσει την προσέγγιση του Welch στο τεστ t (που έχει σχεδόν το ίδιο αποτέλεσμα με άνισους βαθμούς ελευθερίας) και τέλος, με το level(99) (συντομογραφία l(99)) εσείς μπορεί, σε αυτήν την περίπτωση, να ζητήσει ένα επίπεδο εμπιστοσύνης 99 τοις εκατό αντί του προεπιλεγμένου επιπέδου 95, το οποίο χρησιμοποιείται για τον υπολογισμό των διαστημάτων εμπιστοσύνης.

Σημειώνεται ότι ειδικότερη περίπτωση είναι ο έλεγχος διαφοράς μέσου όρου της ίδιας μεταβλητής υπολογισμένοι για περισσότερα από δύο υποσύνολα δεδομένων από το συνολικό δείγμα παρατηρήσεων.

Υποθέστε ότι η μεταβλητή vrCAT είναι μια κατηγορική μεταβλητή και αναφέρεται σε διάφορες κατηγορίες/ιδιότητες. Τότε μπορούμε να ελέγχουμε αν ο μέσος όρος της μεταβλητής vr1 διαφοροποιείται μεταξύ των διαφόρων υποσυνόλων δεδομένων εκ των οποίων το κάθε ένα αναφέρεται σε συγκεκριμένη κατηγορία. Η σύνταξη της εντολής στο **STATA** θα ήταν:

```
anonva vr1, by vrCAT
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής.

3.5. Ανάλυση συσχετίσεων στο **STATA**

Η συσχέτιση μεταβλητών μαζί είναι μια μέθοδος για να ελεγχθεί εάν υπάρχει στατιστικά σημαντική σχέση μεταξύ συνεχών μεταβλητών. Αυτό είναι χρήσιμο εάν θέλετε να μάθετε εάν υπάρχει μια σχέση και εάν πρέπει να διερευνήσουμε περαιτέρω αυτήν τη σχέση με άλλες στατιστικές μεθόδους. Υπάρχουν πολλά τεστ για τη συσχέτιση συνεχών μεταβλητών μαζί και σε αυτόν τον οδηγό θα επικεντρωθούμε στο δείγμα συσχέτισης Pearson και Spearmans rho.

Εντολές corr και pwcorr

Όταν εξετάζουμε τη συσχέτιση μεταξύ μεταβλητών που λαμβάνουν αριθμητικές τιμές οι οποίες πραγματικοί αριθμοί (η τουλάχιστον μία μεταβλητή είναι) τότε υπολογίζεται η συσχέτιση δείγματος Pearson. Οι εντολές στο **STATA** για την εξαγωγή συσχέτισης δειγματος Pearson είναι δύο: corr και pwcorr.

Υπάρχουν δύο είδη διαφορών μεταξύ των δύο εντολών. Το πρώτο είναι ότι με την corr, το **STATA** χρησιμοποιεί τη διαγραφή λίστας. Δηλαδή, ο πίνακας συσχέτισης υπολογίζεται μόνο για εκείνες τις περιπτώσεις που δεν έχουν τιμή που λείπει σε καμία από τις μεταβλητές της λίστας. Αντίθετα, η εντολή pwcorr χρησιμοποιεί διαγραφή κατά ζεύγη. Με άλλα λόγια, κάθε συσχέτιση υπολογίζεται για όλες τις περιπτώσεις που δεν λείπουν τιμές για αυτό το συγκεκριμένο ζεύγος μεταβλητών.

Διαφορές υπάρχουν επίσης και στις επιλογές που μπορούν να χρησιμοποιηθούν κατά την εκτέλεση της κάθε μίας εντολής.

Έστω ότι μας ενδιαφέρει να προσδιορίσουμε τη συσχέτιση δείγματος τριών μεταβλητών (vr1, vr2 και vr3) με τη χρήση της εντολής corr. Τότε η σύνταξη της εντολής στο **STATA** θα ήταν:

```
corr vr1 vr2 vr3
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Η εντολή corr έχει δύο επιλογές με τα δηλωτικά m και c. Η επιλογή m θα εμφανίσει τον μέσο όρο, την τυπική απόκλιση, το ελάχιστο και το μέγιστο κάθε μεταβλητής και η σύνταξη της εντολής στο **STATA** είναι:

```
corr vr1 vr2 vr3, m
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Η επιλογή c θα τον πίνακα συνδιακύμανσης αντί του πίνακα συσχέτισης και η σύνταξη της εντολής στο **STATA** είναι:

```
corr vr1 vr2 vr3, m
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Φυσικά, οι επιλογές m και c μπορούν να συνδυαστούν:

```
corr vr1 vr2 vr3, m c
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Έστω, τώρα, ότι μας ενδιαφέρει να προσδιορίσουμε τη συσχέτιση δείγματος τριών μεταβλητών (vr1, vr2 και vr3) με τη χρήση της εντολής pwcorr. Τότε η σύνταξη της εντολής στο **STATA** θα ήταν:

```
pwcorr vr1 vr2 vr3
```

Μερικές από τις επιλογές της εντολής pwcorr είναι οι: o, sig p(.1) star(.05). Η επιλογή o εμφανίζει τον αριθμό των παρατηρήσεων. Η επιλογή p(.1) λέει στο **STATA** να εμφανίζει μόνο συσχετίσεις με επίπεδο σημασίας 0,1 ή καλύτερο (δηλαδή χαμηλότερο) και το star(.05) ζητά από το Stata να εμφανίσει ένα αστέρι με κάθε συσχέτιση που είναι σημαντικός στο 0,05 ή καλύτερο . Και πάλι, οποιοσδήποτε συνδυασμός αυτών των επιλογών είναι δυνατός. Έστω, τώρα, ότι μας ενδιαφέρει να προσδιορίσουμε τη συσχέτιση δείγματος τριών μεταβλητών (vr1, vr2 και vr3) με τη χρήση της εντολής pwcorr και με όλες τις παραπάνω επιλογές. Τότε η σύνταξη της εντολής στο **STATA** θα ήταν:

```
pwcorr vr1 vr2 vr3, o sig p(.1) star (.05)
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής pwcorr για ένα υποθετικό αρχείο δεδομένων (με επιλογές sig p(.1) και star(.05)) και για έξι μεταβλητές (csat/expense/percent/income /high/college) εμφανίζονται τα ακόλουθα στο παράθυρο Results:

	csat	expense	percent	income	high	college
csat	1.0000					
expense	-0.4663*	1.0000				
	0.0006					
percent	-0.8758*	0.6509*	1.0000			
	0.0000	0.0000				
income	-0.4713*	0.6784*	0.6733*	1.0000		
	0.0005	0.0000	0.0000			
high	0.0858	0.3133*	0.1413	0.5099*	1.0000	
	0.5495	0.0252	0.3226	0.0001		
college	-0.3729*	0.6400*	0.6091*	0.7234*	0.5319*	1.0000
	0.0070	0.0000	0.0000	0.0000	0.0001	

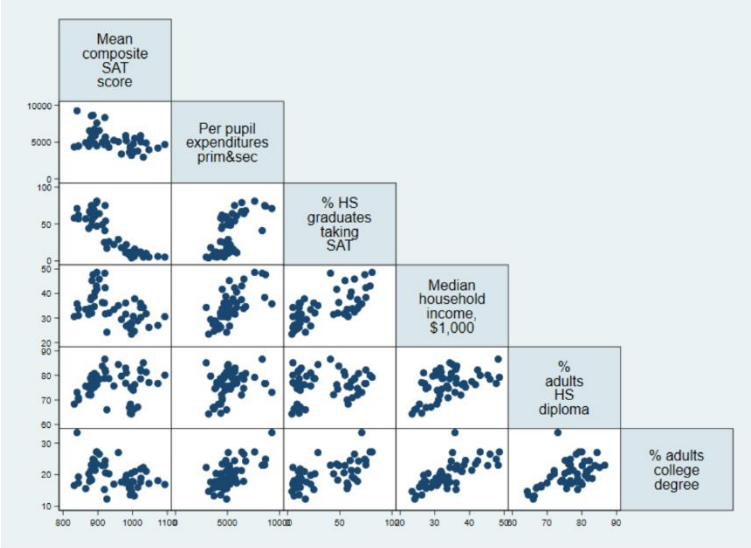
Στον πίνακα, οι αριθμοί είναι συντελεστές συσχέτισης Pearson, κυμαίνονται από -1 σε 1. Για διευκόλυνση της ερμηνευτικής σημειώνονται τα ακόλουθα αναφορικά με την τιμή τη συσχέτισης (r):

- Αν $r = \pm 1$ υπάρχει τέλεια γραμμική συσχέτιση.
- Αν $-0,3 \leq r < 0,3$ δεν υπάρχει γραμμική συσχέτιση. Αυτό, όμως, δεν σημαίνει ότι δεν υπάρχει άλλου είδους συσχέτιση μεταξύ των δύο μεταβλητών.
- Αν $-0,5 < r \leq -0,3$ ή $0,3 \leq r < 0,5$ υπάρχει ασθενής γραμμική συσχέτιση.
- Αν $-0,7 < r \leq -0,5$ ή $0,5 \leq r < 0,7$ υπάρχει μέση γραμμική συσχέτιση. Αν $-0,8 < r \leq -0,7$ ή $0,7 \leq r < 0,8$ υπάρχει ισχυρή γραμμική συσχέτιση.
- Αν $-1 < r \leq -0,8$ ή $0,8 \leq r < 1$ υπάρχει πολύ ισχυρή γραμμική συσχέτιση.
- Θετικές τιμές του r δεν υποδηλώνουν, κατ' ανάγκην μεγαλύτερο βαθμό γραμμικής συσχέτισης από το βαθμό γραμμικής συσχέτισης που υποδηλώνουν αρνητικές τιμές του r .
- Ο βαθμός γραμμικής συσχέτισης καθορίζεται από την απόλυτη τιμή του r και όχι από το πρόσημο του r . Το πρόσημο του r καθορίζει το είδος, μόνο, της συσχέτισης (θετική ή αρνητική). Μας πληροφορεί δηλαδή για το αν αύξηση της μιας μεταβλητής αντιστοιχεί σε αύξηση ή σε μείωση της άλλης μεταβλητής. Για παράδειγμα η τιμή 9 $r = -0$, δείχνει ισχυρότερη γραμμική συσχέτιση από την τιμή 8 $r = 0$, ενώ οι τιμές 6 $r = -0,6$ και $r = 0$, δείχνουν ίδιο βαθμό γραμμικής συσχέτισης αλλά αντίθετο είδος.

Η εντολή graph matrix παράγει μια γραφική αναπαράσταση του πίνακα συσχέτισης δείχνοντας μια σειρά από διαγράμματα διασποράς για όλες τις μεταβλητές. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής για ένα υποθετικό αρχείο δεδομένων και για έξι μεταβλητές (csat/expense/percent/income /high/college) αν δώσουμε την ακόλουθη εντολή στο **STATA**:

```
graph matrix  vr1 vr2 vr3, half
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής:



Εντολή spearman

Για την εξέταση της συσχέτισης μεταξύ μεταβλητών όπου κάποια/ες από αυτές είναι διατεταγμένες είναι περισσότερο δόκιμο να χρησιμοποιηθεί ως μέτρο συσχέτισης το Spearman rho. Η σχετική εντολή στο **STATA** είναι η spearman. Έστω, τώρα, ότι μας ενδιαφέρει να προσδιορίσουμε τη συσχέτιση δείγματος τριών μεταβλητών (vr1, vr2 και vr3) με τη χρήση της εντολής spearman και με όλες τις παραπάνω επιλογές. Τότε η σύνταξη της εντολής στο **STATA** θα ήταν:

```
spearman vr1 vr2 vr3
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής.

3.6. Εξαγωγή αποτελεσμάτων συσχέτισης από το **STATA**

Εντολή asdoc

Η εξαγωγή αποτελεσμάτων περιγραφικής στατιστικής από το **STATA** μπορεί να γίνει με τη χρήση της εντολής asdoc. Η εντολή αυτή όμως θα πρέπει να εγκατασταθεί στο **STATA** και για το λόγο αυτό θα πρέπει να εκτελεσθεί ο ακόλουθος κώδικας στο **STATA**:

```
ssc install asdoc
```

Η εντολή asdoc συνδυάζεται με την εντολή corr (ή pwcorr) η γενική σύνταξη της έχει ως εξής:

```
asdoc corr (ή pwcorr), save(ONOMA_ΑΡΧΕΙΟΥ)
```

Όπου στη θέση ONOMA_ΑΡΧΕΙΟΥ δίνεται μία ονομασία με λατινικούς χαρακτήρες. Προσοχή γιατί η επιλογή replace θα αντικαταστήσει προϋπάρχουν αρχείο με το ίδιο όνομα.

Εάν θέλετε να εξαγάγετε τον πίνακα αποτελεσμάτων στο Excel, χρησιμοποιήστε την επέκταση *.xls αντί να χρησιμοποιήσετε *.doc

Αν επιθυμείτε να εκτελεσθεί η εντολή για συγκεκριμένες μεταβλητές (να εξαχθούν τα αποτελέσματα της περιγραφικής στατιστικής για επιλεγμένες μεταβλητές) βάζετε τα ονόματα των μεταβλητών (έστω vr1, vr2 και vr3) μετά την εντολή corr (ή pwcorr)

```
asdoc corr (ή pwcorr) vr1 vr2 vr3, save(ONOMA_ΑΡΧΕΙΟΥ)
```

4. Ανάλυση παλινδρόμησης στο STATA

4.1. Εκτίμηση γραμμικού (μονομεταβλητό) υποδείγματος παλινδρόμησης με το **STATA**

Εντολή regress

Για να εκτελέσετε ένα γραμμικό (μονομεταβλητό) υπόδειγμα γραμμικής παλινδρόμησης που αφορά μια εξαρτημένη μεταβλητή (vrY) και μια ανεξάρτητη μεταβλητή (vrX1), τότε αξιοποιείται η εντολή regress και η σύνταξη της εντολής στο **STATA** θα ήταν:

```
regress vrY vrX1
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής regress για ένα υποθετικό αρχείο δεδομένων εμφανίζονται τα ακόλουθα στο παράθυρο Results:

```
      Source |       SS           df       MS      Number of obs   =        51
-------------+----------------------------------   F(1, 49)        =     13.61
       Model |  48708.3001         1  48708.3001   Prob > F        =    0.0006
    Residual |   175306.21        49  3577.67775   R-squared       =    0.2174
-------------+----------------------------------   Adj R-squared   =    0.2015
       Total |   224014.51        50   4480.2902   Root MSE        =    59.814

------------------------------------------------------------------------------
        csat |      vrY   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     expense |     vrX1   .0060371    -3.69   0.001    -.0344077   -.0101436
       _cons |   1060.732   32.7009    32.44   0.000     995.0175    1126.447
------------------------------------------------------------------------------
```

Ερμηνευτική αποτελεσμάτων:

- Prob > F = 0,0006 : Αυτή είναι η τιμή p του μοντέλου. Ελέγχει τη μηδενική υπόθεση ότι το R-squared είναι ίσο με 0. Για να απορρίψουμε τη μηδενική υπόθεση, συνήθως χρειαζόμαστε μια τιμή p μικρότερη από 0,05. Εδώ, η τιμή p του 0,0006 υποδεικνύει μια στατιστικά σημαντική σχέση μεταξύ X και Y.
- R-squared = 0,2174: Το R-squared δείχνει το ποσό της διακύμανσης του Y που εξηγείται από το X. Σε αυτή την περίπτωση η μεταβλητή vrX1 εξηγεί το 22% της διακύμανσης στη μεταβλητή vrY.
- Adj R-squared = 0,2015 : Το Adj R-squared δείχνει το ίδιο με το R-squared, αλλά προσαρμόζεται από τον αριθμό των περιπτώσεων και τον αριθμό των μεταβλητών. Όταν ο αριθμός των μεταβλητών είναι μικρός και ο αριθμός των περιπτώσεων είναι πολύ μεγάλος, τότε το Adj R-square είναι πιο κοντά στο R-square.
- Root MSE = 59.814 : ρίζα μέσου τετραγώνου σφάλματος. Όσο πιο κοντά στο μηδέν, καλύτερα η εφαρμογή.
- Ο εκτιμώμενος συντελεστής της vrX1 είναι -,0222756. Αυτό σημαίνει ότι για κάθε αύξηση της μεταβλητής vrX1 κατά μία μονάδα, η τιμή της μεταβλητής vrY μειώνεται κατά 0,022 μονάδες.

- Οι τιμές t ελέγχουν τη μηδενική υπόθεση ότι κάθε συντελεστής είναι 0. Για να το απορρίψετε, χρειάζεστε μια τιμή t μεγαλύτερη από 1,96 (για 95% εμπιστοσύνη). Μπορείτε να λάβετε τις τιμές t διαιρώντας τον συντελεστή με το τυπικό σφάλμα του. Οι τιμές t δείχνουν επίσης τη σημασία μιας μεταβλητής στο μοντέλο.
- P>|t| = 0,001: Η τιμή p τη μηδενική υπόθεση ότι ο συντελεστής είναι ίσος με 0 (δηλ. δεν υπάρχει σημαντική επίδραση). Για να απορριφθεί αυτό, η τιμή p πρέπει να είναι μικρότερη από 0,05 (μπορείτε επίσης να επιλέξετε ένα άλφα 0,10). Σε αυτή την περίπτωση, η μεταβλητή vrX1 είναι στατιστικά σημαντική για την εξήγηση της συμπεριφοράς της μεταβλητής vrY.

Εντολή predict

Ενίοτε μας ενδιαφέρει μετά την εκτέλεσης του υποδείγματος γραμμικής παλινδρόμησης να αποθηκευτούν σε την μορφή μίας νέας μεταβλητής οι προβλεπόμενες τιμές της εξαρτημένης μεταβλητής για να μπορούν να συγκριθούν με τις πραγματικές τιμές της εξαρτημένης μεταβλητής. Για το λόγο αυτό συνίσταται ακριβώς μετά την εντολή regress (στην επόμενη γραμμή του πεδίου των εντολών του **STATA**) να εκτελεσθεί η εντολή predict η οποία συνοδεύεται με το επιθυμητό όνομα της νέας μεταβλητής που περιλαμβάνει τις προβλεπόμενες τιμές της εξαρτημένης μεταβλητής το οποίο είναι το όνομα της εξαρτημένης μεταβλητής με το πρόθεμα _predict. Δηλαδή στην περίπτωση της εξαρτημένης μεταβλητής με ονομασία vrY η νέα μεταβλητή θα ονομαστεί vrY_predict και η ακολουθία εντολών στο **STATA** θα έχει ως εξής:

```
regress vrY vrX1
predict vrY_predict
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής.

Εντολή predict e, resid

Ως κατάλοιπα σε μία εκτίμηση ενός υποδείγματος γραμμικής παλινδρόμησης ορίζεται η διαφορά μεταξύ των πραγματικών τιμών της εξαρτημένης μεταβλητής (έστω με την ονομασία vrY) και των αντίστοιχων προβλεπόμενων τιμών με βάση το εκτιμηθέν υπόδειγμα γραμμικής παλινδρόμησης (έστω με την ονομασία PvrY). Συνήθως τα κατάλοιπα συμβολίζονται με e και μετά την εκτέλεση ενός υποδείγματος γραμμικής παλινδρόμησης θα πρέπει να αποθηκεύονται προκειμένου να λάβουν χώρα κάποιοι απαραίτητοι στατιστικοί έλεγχοι στο πλαίσιο του ελέγχου των παραδοχών της εκτίμησης ενός υποδείγματος γραμμικής παλινδρόμησης με τη μέθοδο των ελαχίστων τετραγώνων.

Για το λόγο αυτό συνίσταται ακριβώς μετά την εντολή regret (στην επόμενη γραμμή του πεδίου των εντολών του **STATA**) να εκτελεσθεί η εντολή predict e,

resid η οποία θα οδηγήσει στη δημιουργία μίας νέας μεταβλητής με την ονομασία e. Σε αυτή την περίπτωση η ακολουθία εντολών στο **STATA** θα έχει ως εξής:

```
regress vrY vrX1
predict e, resid
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής.

Εντολές swilk, sfrancia και sktest

Ένας βασικός έλεγχος στο πλαίσιο της στατιστικής ανάλυσης είναι αν οι τιμές μίας αριθμητικής μεταβλητής ακολουθούν την κανονική κατανομή. Για παράδειγμα, μία από τις βασικές παραδοχές του υποδείγματος γραμμικής παλινδρόμησης είναι ότι τα κατάλοιπα της παλινδρόμησης ακολουθούν την κανονική κατανομή. Τρεις γνωστοί έλεγχοι αναφορικά με την υπόθεση της κανονικότητας είναι οι Shapiro-Wilk test, Shapiro-Francia test και Skewness and Kurtosis test και οι αντίστοιχες εντολές στο STATA είναι αντίστοιχα οι ακόλουθοι: swilk, sfrancia, sktest.

Η μηδενική υπόθεση για αυτά τα τεστ είναι ότι η μεταβλητή είναι κανονικά κατανεμημένη. Εάν η τιμή p-value του τεστ είναι μικρότερη από κάποιο επίπεδο σημαντικότητας, τότε μπορούμε να απορρίψουμε τη μηδενική υπόθεση και να συμπεράνουμε ότι υπάρχουν επαρκή στοιχεία για να πούμε ότι η μεταβλητή δεν κατανέμεται κανονικά.

Έστω ότι επιθυμούμε να ελέγξουμε την υπόθεση περί κανονικής κατανομής της αριθμητικής μεταβλητής vr1 με τον έλεγχο swilk. Η σύνταξη της εντολής στο **STATA** θα ήταν:

```
swilk vr1
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής regress για ένα υποθετικό αρχείο δεδομένων εμφανίζονται τα ακόλουθα στο παράθυρο Results:

Variable	Obs	W	V	z	Prob>z
displacement	74	0.92542	4.803	3.423	0.00031

Ερμηνευτική αποτελεσμάτων:

- Αριθμός παρατηρήσεων: 74. Αυτός είναι ο αριθμός των παρατηρήσεων που χρησιμοποιήθηκαν στο τεστ.
- W: 0,92542. Αυτό είναι το στατιστικό του τεστ.
- Prob>z: 0,00031. Αυτή είναι η τιμή p-value που σχετίζεται με τον έλεγχο της μηδενικής υπόθεσης. Δεδομένου ότι η τιμή p είναι μικρότερη από 0,05, μπορούμε να απορρίψουμε τη μηδενική υπόθεση του τεστ. Έχουμε επαρκή στοιχεία για να πούμε ότι η μεταβλητή vr1 δεν κατανέμεται κανονικά.

Μπορούμε επίσης να εκτελέσουμε τη δοκιμή Shapiro-Wilk σε περισσότερες από μία μεταβλητές ταυτόχρονα, παραθέτοντας πολλές μεταβλητές μετά την εντολή swilk:

Έστω ότι επιθυμούμε να ελέγξουμε την υπόθεση περί κανονικής κατανομής της αριθμητικής μεταβλητής vr1 με τον έλεγχο sfrancia. Η σύνταξη της εντολής στο **STATA** θα ήταν:

```
sfrancia vr1
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής regress για ένα υποθετικό αρχείο δεδομένων εμφανίζονται τα ακόλουθα στο παράθυρο Results:

Variable	Obs	W'	V'	z	Prob>z
displacement	74	0.93011	4.975	3.110	0.00094

Ερμηνευτική αποτελεσμάτων:

- Αριθμός παρατηρήσεων: 74. Αυτός είναι ο αριθμός των παρατηρήσεων που χρησιμοποιήθηκαν στο τεστ.
- W: 0,93011. Αυτό είναι το στατιστικό τεστ για το τεστ.
- Prob>z: 0,00094. Αυτή είναι η τιμή p-value που σχετίζεται με τον έλεγχο της μηδενικής υπόθεσης. Δεδομένου ότι η τιμή p είναι μικρότερη από 0,05, μπορούμε να απορρίψουμε τη μηδενική υπόθεση του τεστ. Έχουμε επαρκή στοιχεία για να πούμε ότι η μεταβλητή μετατόπιση δεν κατανέμεται κανονικά.

Παρόμοια με τη δοκιμή Shapiro-Wilk, μπορείτε να εκτελέσετε τη δοκιμή Shapiro-Francia σε περισσότερες από μία μεταβλητές ταυτόχρονα, αναφέροντας πολλές μεταβλητές μετά την εντολή sfrancia.

Έστω ότι επιθυμούμε να ελέγξουμε την υπόθεση περί κανονικής κατανομής της αριθμητικής μεταβλητής vr1 με τον έλεγχο sktest. Η σύνταξη της εντολής στο **STATA** θα ήταν:

```
sktest vr1
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής regress για ένα υποθετικό αρχείο δεδομένων εμφανίζονται τα ακόλουθα στο παράθυρο Results:

	Skewness/Kurtosis tests for Normality			joint	
Variable	Obs	Pr(Skewness)	Pr(Kurtosis)	adj chi2(2)	Prob>chi2
displacement	74	0.0337	0.2048	5.81	0.0547

Ερμηνευτική αποτελεσμάτων:

- Αριθμός παρατηρήσεων: 74. Αυτός είναι ο αριθμός των παρατηρήσεων που χρησιμοποιήθηκαν στο τεστ.
- adj chi(2): 5,81. Αυτό είναι το στατιστικό τεστ Chi-Square για το στατιστικό έλεγχο.

- Prob>chi2: 0,0547. Αυτή είναι η τιμή p-value που σχετίζεται με τον έλεγχο της μηδενικής υπόθεσης. Δεδομένου ότι η τιμή p είναι μικρότερη από 0,05, μπορούμε να απορρίψουμε τη μηδενική υπόθεση του τεστ. Έχουμε επαρκή στοιχεία για να πούμε ότι η μεταβλητή μετατόπιση δεν κατανέμεται κανονικά.

Εντολές estat hettest και estat imtest, white

Για δεδομένες τιμές των ανεξάρτητων μεταβλητών, η διακύμανση της κατανομής των τιμών του διαταρακτικού όρου είναι σταθερή (ομοσκεδαστικότητα). Αν η παραπάνω παραδοχή δεν ισχύει τότε εμφανίζεται το πρόβλημα της ετεροσκεδαστικότητας. Δύο σύνηθες έλεγχοι για τη διάγνωση του προβλήματος της ετεροσκεδαστικότητας είναι ο έλεγχος Breusch–Pagan–Godfrey και ο έλεγχος White. Οι αντίστοιχες εντολές για τον έλεγχο ετεροσκεδαστικότητας είναι η εντολή estat hettest και η εντολή estat imtest, white.

Για το λόγο αυτό συνίσταται ακριβώς μετά την εντολή regress (στην επόμενη γραμμή του πεδίου των εντολών του **STATA**) να εκτελεσθεί μία από τις δύο εντολές. Για παράδειγμα αν επιλεγεί η εντολή estat hettest και η ακολουθία εντολών στο **STATA** θα έχει ως εξής:

```
regress vrY vrX1
estat hettest
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Η μηδενική υπόθεση είναι ότι έχουμε ομοσκεδαστικότητα και απορρίπτεται (δεν γίνεται αποδεκτή) αν το p-value του στατιστικού X^2 είναι μικρότερο του 5% (δηλαδή θα δείτε στο report το εξής: prob>chi2 = XXX < 0.05).

Αν επιλεγεί η εντολή estat imtest, white και η ακολουθία εντολών στο **STATA** θα έχει ως εξής:

```
regress vrY vrX1
estat imtest, white
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Και την περίπτωση αυτή η μηδενική υπόθεση είναι ότι έχουμε ομοσκεδαστικότητα και απορρίπτεται (δεν γίνεται αποδεκτή) αν το p-value του στατιστικού X^2 είναι μικρότερο του 5% (δηλαδή θα δείτε στο report το εξής: prob>chi2 = XXX < 0.05).

Επιλογή robust

Αν διαπιστωθεί πρόβλημα ετεροσκεδαστικότητας τότε η γραμμική παλινδρόμηση θα πρέπει να εκτελεσθεί με την επιλογή robust. Έστω ότι επιθυμείτε να εκτελέσετε ένα απλό μοντέλο γραμμικής παλινδρόμησης που αφορά μια εξαρτημένη μεταβλητή (vrY) και μια ανεξάρτητη μεταβλητή (vrX1) και έχετε διαπιστώσει το

πρόβλημα της ετεροσκεδαστικότητας, τότε αξιοποιείται η εντολή regress με την επιλογή robust και η σύνταξη της εντολής στο **STATA** θα ήταν:

```
regress vrY vrX1, robust
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής.

Εντολή estat bgodfrey

Για δεδομένες τιμές των ανεξάρτητων μεταβλητών, οι τιμές του διαταρακτικού όρου είναι μηδέν δεν συσχετίζονται μεταξύ τους (απουσία αυτοσυσχέτισης). Ένας συνήθης έλεγχος για τη διάγνωση του προβλήματος της αυτοσυσχέτισης είναι ο έλεγχος Breusch–Godfrey. Η εντολή για τον έλεγχο Breusch–Godfrey είναι η εντολή estat bgodfrey και εκτελείται μετά την εκτέλεση της εντολής regress (στην επόμενη γραμμή του πεδίου των εντολών του **STATA**). Η ακολουθία εντολών στο **STATA** θα έχει ως εξής:

```
regress vrY vrX1
estat bgodfrey, lag(1)
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Η μηδενική υπόθεση είναι ότι δεν έχουμε αυτοσυσχέτιση απορρίπτεται (δεν γίνεται αποδεκτή) αν το p-value του στατιστικού X^2 είναι μικρότερο του 5% (δηλαδή θα δείτε στο report το εξής: prob>chi2 = XXX < 0.05).

Εντολή ovtest

Μία συνηθισμένη ερευνητική ανησυχία είναι η εσφαλμένη εξειδίκευση της μαθηματικής έκφρασης της εξίσωσης παλινδρόμησης. Μία από τις πιθανές αιτίες είναι η παράλειψη ανεξάρτητης μεταβλητής (omitted-variable bias). Ένας συνήθης έλεγχος για τη διάγνωση του προβλήματος εσφαλμένης εξειδίκευσης εξαιτίας της παράλειψης ανεξάρτητης μεταβλητής είναι ο έλεγχος Ramsey RESET. Η εντολή για τον έλεγχο Ramsey RESET είναι η εντολή ovtest και εκτελείται μετά την εκτέλεση της εντολής regress (στην επόμενη γραμμή του πεδίου των εντολών του **STATA**). Η ακολουθία εντολών στο **STATA** θα έχει ως εξής:

```
regress vrY vrX1
ovtest
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Χάρη παραδείγματος έστω ότι μετά την εκτέλεση της εντολής regress για ένα υποθετικό αρχείο δεδομένων εμφανίζονται τα ακόλουθα στο παράθυρο Results:

```
Ramsey RESET test using powers of the fitted values
       Ho:  model has no omitted variables
                  F(3, 38) =      2.15
                  Prob > F =      0.1096
```

Η μηδενική υπόθεση είναι ότι δεν έχουμε σφάλμα παράλειψης μεταβλητής απορρίπτεται (δεν γίνεται αποδεκτή) αν το p-value του στατιστικού X^2 είναι μικρότερο του 5% (δηλαδή θα δείτε στο report το εξής: prob>F = XXX < 0.05).

4.2. Εκτίμηση γραμμικού (πολυμεταβλητό) υποδείγματος παλινδρόμησης με το **STATA**

Στο γραμμικό (πολυμεταβλητό) υπόδειγμα παλινδρόμησης μία εξαρτημένη μεταβλητή (έστω η vrY) εκφράζεται ως συνάρτηση περισσότερων της μίας ανεξάρτητων μεταβλητών (έστω ν: vrX1, vrX2, … vrXν). Ότι έχει ειπωθεί στην παράγραφο 4.1. για το γραμμικό (μονομεταβλητ'ο) υπόδειγμα στο **STATA** ισχύει και για το γραμμικό (πολυμεταβλητό) υπόδειγμα παλινδρόμησης σε γενικές γραμμές. Απλά στο π γραμμικό (πολυμεταβλητό) υπόδειγμα παλινδρόμησης ο ερευνητής θα πρέπει να μεριμνήσει και για την ύπαρξη του προβλήματος της πολυσυγγραμικότητας.

Για να εκτιμήσετε ένα υπόδειγμα (πολυμεταβλητής) γραμμικής παλινδρόμησης που αφορά μια εξαρτημένη μεταβλητή (vrY) και ένα πλήθος ανεξάρτητων μεταβλητών (έστω τρεις με την ονομασία vrX1, vrX2 και vrX3 αντίστοιχα), τότε αξιοποιείται η εντολή regress και η σύνταξη της εντολής στο **STATA** θα ήταν:

```
regress vrY vrX1 vrX2 vrX3
```

Εντολή vif

Μεταξύ των ανεξάρτητων μεταβλητών δεν πρέπει να υπάρχει γραμμική συσχέτιση. Όταν υπάρχει πολυσυγγραμμικότητα σε ένα μοντέλο, τα τυπικά σφάλματα μπορεί να διογκωθούν. Το **STATA** θα αφαιρέσει μία από τις μεταβλητές για να αποφευχθεί η διαίρεση με το μηδέν στη διαδικασία OLS.

Η εντολή στο **STATA** για τον έλεγχο της πολυσυγγραμμικότητας είναι vif (variance inflation factor) και εκτελείται μετά την εκτέλεση της εντολής regress (στην επόμενη γραμμή του πεδίου των εντολών του **STATA**). Έστω ότι θέλετε να ελέγξετε την ύπαρξη πολυσυγγραμικότητας για ένα υπόδειγμα (πολυμεταβλητής) γραμμικής παλινδρόμησης που αφορά μια εξαρτημένη μεταβλητή (vrY) και ένα πλήθος ανεξάρτητων μεταβλητών (έστω τρεις με την ονομασία vrX1, vrX2 και vrX3 αντίστοιχα), τότε η ακολουθία εντολών στο **STATA** θα έχει ως εξής:

```
regress vrY vrX1 vrX2 vrX3
vif
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής. Έστω σε ένα υποθετικό παράδειγμα ενός πολυμεταβλητού γραμμικού μοντέλου με οκτώ ανεξάρτητες μεταβλητές (csat expense percent income high college i.region) εκτελείται η εντολή vif και δίδει το ακόλουθο αποτέλεσμα:

```
. vif
    Variable |       VIF       1/VIF
-------------+----------------------
     expense |      3.18    0.314656
     percent |      3.88    0.257790
      income |      4.78    0.209068
        high |      4.71    0.212167
     college |      4.34    0.230156
      region |
           2 |      3.57    0.279850
           3 |      4.18    0.239156
           4 |      1.80    0.556855
-------------+----------------------
    Mean VIF |      3.81
```

Ένα VIF > 10 ή ένα 1/VIF < 0,10 υποδηλώνει την παρουσία πολυσυγγραμμικότητας στο μοντέλο.

4.3. Ανάλυση παλινδρόμησης με δεδομένα πάνελ στο **STATA**

Εντολή xtset

Υπενθυμίζεται ότι τα δεδομένα τύπου πάνελ αποτελούν ένα συνδυασμό χρονολογικής σειράς και διαστρωματικών δεδομένων, δηλαδή ένας αριθμός χρονολογικών ακολουθιών τιμών και έναν αριθμό περιπτώσεων. Παραδείγματα: το ύψος των πωλήσεων ενός αριθμού εταιρειών της τελευταίας δεκαετίας, η αξία των μηναίων πωλήσεων ενός αριθμού εταιρειών για το χρονικό διάστημα από το 1990 έως και το 2020, το βάρος ενός α-ριθμού αθλητών τους τελευταίους 36 μήνες.

Το **STATA** δεν μπορεί να γνωρίζει εκ προοιμίου αν το αρχείο με τα δεδομένα που ανοίξαμε αφορά πάνελ δεδομένα. Τούτο θα πρέπει να δηλωθεί αμέσως αφού ανοίξουμε ένα αρχείο με δεδομένα πάνελ και κάθε φορά που το ανοίγουμε. Η σχετική εντολή είναι η xtset. Έστω ότι η αριθμητική μεταβλητή firm_id χρησιμοποιείται για να διακριτοποιηθούν οι εταιρείες στο δείγμα πάνελ και η αριθμητική μεταβλητή time_id για να διακριτοποιηθούν οι χρονικές περίοδοι (ημέρες, μήνες, τετράμηνα, έτη κ.λπ). Η σύνταξη της εντολής xtset στο στο **STATA** θα έχει ως εξής:

```
xtset firm_id time_id
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής.

Εντολή xtreg

Για την εκτίμηση ενός υποδείγματος (μονομεταβλητής ή πολυμεταβλητής) γραμμικής παλινδρόμησης με δεδομένα πάνελ δεν χρησιμοποιείται η εντολή regress αλλά η εντολή xtreg (έχει προηγηθεί η εκτέλεση της εντολής xtset).

Έστω ότι θέλετε να εκτιμήσετε ένα υπόδειγμα (πολυμεταβλητής) γραμμικής παλινδρόμησης με δεδομένα πάνελ που αφορά μια εξαρτημένη μεταβλητή (vrY) και ένα

πλήθος ανεξάρτητων μεταβλητών (έστω τρεις με την ονομασία vrX1, vrX2 και vrX3 αντίστοιχα) με δεδομένα πάνελ, τότε η εντολή στο **STATA** θα έχει ως εξής:

```
xtreg vrY vrX1 vrX2 vrX3
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής.

Η εκτίμηση ενός υποδείγματος γραμμικής παλινδρόμησης με δεδομένα πάνελ θα πρέπει να συνοδεύεται με όλους τους ελέγχους που περιεγράφηκαν στις παραγράφους 4.1. και 4.2. (ετεροσκεδαστικότητα, αυτοσυσχέτιση, πολυσυγγραμικότητα) με τις αντίστοιχες εντολές και με τις αντίστοιχε δυνατότητες διόρθωσης (δες επιλογή robust στην περίπτωση της ετεροσκεδαστικότητας).

Επιπρόσθετα όταν πραγματοποιείται η εκτίμηση ενός υποδείγματος γραμμικής παλινδρόμησης με δεδομένα πάνελ θα πρέπει ο ερευνητής να προβληματισθεί για το γεγονός ότι το δείγμα του αφορά μία χρονική περίοδο αλλά συντίθεται από διαφορετικές περιπτώσεις δηλαδή εταιρείες η κάθε μία εκ των οποίων έχει τα δικά της ιδιαίτερα χαρακτηριστικά. Τούτο δυνητικά εισάγει μία μεταβλητότητα στη συμπεριφορά του δείγματος που θα πρέπει να ληφθεί υπόψη κατά την εκτίμηση του υποδείγματος.

Προκειμένου να ελεγχθεί η παραπάνω μεταβλητότητα είναι δυνατόν να εκτιμηθεί το υπόδειγμα γραμμικής παλινδρόμησης με δεδομένα πάνελ με την τεχνική της εκτίμησης σταθερών επιδράσεων ή με την τεχνική της εκτίμησης τυχαίων επιδράσεων.

Επιλογή fe

Εκτίμηση με σταθερές επιδράσεις – fixed effects (FE). Όταν πραγματοποιούμε μια εκτίμηση γραμμικής παλινδρόμησης δεδομένων πάνελ με FE υποθέτουμε ότι τα χαρακτηριστικά μίας περίπτωσης (εταιρείας) μπορεί να επηρεάσουν ή να προκαταλάβουν τη συμπεριφορά της εξαρτημένης μεταβλητής. Αν αυτά τα χαρακτηριστικά δεν μοντελοποιηθούν τότε θα παραβιάζεται η παραδοχή περί ανεξαρτησίας των ανεξαρτήτων μεταβλητών μίας γραμμικής παλινδρόμησης με το σφάλμα εκτίμησης (κατάλοιπα). Η εκτίμηση γραμμικής παλινδρόμησης δεδομένων πάνελ με FE επιχειρεί να αφαιρέσει την επίδραση αυτών των αμετάβλητων στο χρόνο χαρακτηριστικών, και επομένως, μπορούμε να εκτιμήσουμε την καθαρή επίδραση των ανεξάρτητων μεταβλητών στην εξαρτημένη μεταβλητή.

Το μοντέλο παλινδρόμησης FE υποθέτει ότι για κάθε μία εταιρεία στο δείγμα μας αντιστοιχεί μία δυαδική μεταβλητή η οποία λαμβάνει την τιμή 1 όταν τα αριθμητικά δεδομένα των υπολοίπων μεταβλητών (της εξαρτημένης και των ανεξάρτητων) αναφέρονται σε αυτή της εταιρεία διαφορετικά λαμβάνει την τιμή 0. Περαιτέρω, η κάθε δυαδική μεταβλητή αντιπροσωπεύει τις επιδράσεις των διαχρονικά σταθερών χαρακτηριστικών της εταιρείας στην συμπεριφορά της εξαρτημένης μεταβλητής (vrY).

Ο ερευνητής έχει δύο επιλογές. Η πρώτη είναι να εισάγει στο γραμμικό μοντέλο παλινδρόμησης τόσες δυαδικές μεταβλητές όσες και οι εταιρείες που εντοπίζονται στο δείγμα του. Με άλλα λόγια αν υποθέσουμε ότι το δείγμα αποτελείται από δυο εταιρείες με αντίστοιχες δυαδικές μεταβλητές D1 και D2 και επιθυμεί να εκτιμήσει ένα υπόδειγμα (πολυμεταβλητής) γραμμικής παλινδρόμησης με δεδομένα πάνελ που αφορά μια εξαρτημένη μεταβλητή (vrY) και ένα πλήθος ανεξάρτητων μεταβλητών (έστω τρεις με την ονομασία vrX1, vrX2 και vrX3 αντίστοιχα) με δεδομένα πάνελ, τότε η εντολή στο **STATA** θα έχει ως εξής:

```
xtreg vrY D1 D2 vrX1 vrX2 vrX3
```

Δηλαδή θα εισάγει τις δυαδικές μεταβλητές ως ανεξάρτητες ενώ αν τις αγνοούσε η αντίστοιχη εντολή στο **STATA** θα είχε ως εξής:

```
xtreg vrY vrX1 vrX2 vrX3
```

Ο παραπάνω τρόπος συνίσταται όταν ο αριθμός των εταιρειών είναι μικρός και ο αριθμός των χρονικών περιόδων είναι μεγάλος, αλλά ακόμη και σε αυτή την περίπτωση δημιουργούνται περαιτέρω προβληματισμοί:

- Η εισαγωγή ανεξαρτήτων μεταβλητών σε ένα οποιοδήποτε υπόδειγμα γραμμικής παλινδρόμησης αυξάνει την πιθανότητα εμφάνισης του προβλήματος της πολυσυγγραμικότητας.
- Είναι δυνατόν να ασκηθεί κριτική ότι ο ερευνητής θα έπρεπε να λάβει υπόψη και την αλληλεπίδραση των δυαδικών μεταβλητών με τις υπόλοιπες ανεξάρτητες μεταβλητές του υποδείγματος.

Για το λόγο αυτό η βιβλιογραφία προτείνει να ορισθεί το οικονομετρικό υπόδειγμα της γραμμικής παλινδρόμησης θεωρώντας ότι το σφάλμα εκτίμησης (δηλαδή το κατάλοιπο) είναι συνάρτηση των δυαδικών μεταβλητών. Τούτο επιτυγχάνεται με την επιλογή fe (σταθερές επιδράσεις) και κατά συνέπεια η εντολή στο **STATA** θα έχει ως εξής:

```
xtreg vrY vrX1 vrX2 vrX3, fe
```

Επιλογή re

Εκτίμηση τυχαίων επιδράσεων – random effects (RE). Εναλλακτικά ό ερευνητής μπορεί να υποθέσει ότι τα ιδιαίτερα χαρακτηριστικά της κάθε περίπτωσης, δηλαδή της κάθε εταιρείας, επηρεάζουν αποκλειστικά τη συμπεριφορά των σφαλμάτων εκτίμησης με τυχαίο μεταβλητό τρόπο. Η θεώρηση αυτή μας δίνει τη δυνατότητα αν επιθυμούμε (αν ο αριθμός των εταιρειών είναι μικρός) στο ίδιο υπόδειγμα να εισάγουμε δυαδικές μεταβλητές που να αντιστοιχούν σε κάθε μία εταιρεία αλλά και ταυτόχρονα να παραμετροποιήσουμε τις τυχαίες επιδράσεις των ιδιαίτερων χαρακτηριστικών των εταιρειών στο υπόδειγμα μας.

Έστω ότι επιθυμούμε να εκτιμήσουμε ένα υπόδειγμα (πολυμεταβλητής) γραμμικής παλινδρόμησης με δεδομένα πάνελ που αφορά μια εξαρτημένη μεταβλητή (vrY)

και ένα πλήθος ανεξάρτητων μεταβλητών (έστω τρεις με την ονομασία vrX1, vrX2 και vrX3 αντίστοιχα) με δεδομένα πάνελ, και με τυχαίες επιδράσεις. Τούτο επιτυγχάνεται με την επιλογή re (τυχαίες επιδράσεις) και κατά συνέπεια η εντολή στο **STATA** θα έχει ως εξής:

```
xtreg vrY vrX1 vrX2 vrX3, fe
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής.

Έλεγχος hausman

Τελικά για την εκτίμηση ενός υπόδειγμα (πολυμεταβλητής) γραμμικής παλινδρόμησης με δεδομένα πάνελ τι θα πρέπει να επιλέξουμε; Σταθερές επιδράσεις ή τυχαίες επιδράσεις; Για να απαντήσουμε το ερώτημα αυτό θα πρέπει να εκτελέσουμε το έλεγχο Hausman.

Έστω ότι επιθυμούμε να εκτιμήσουμε ένα υπόδειγμα (πολυμεταβλητής) γραμμικής παλινδρόμησης με δεδομένα πάνελ που αφορά μια εξαρτημένη μεταβλητή (vrY) και ένα πλήθος ανεξάρτητων μεταβλητών (έστω τρεις με την ονομασία vrX1, vrX2 και vrX3 αντίστοιχα) με δεδομένα πάνελ. Για να εξάγουμε στατιστικό συμπέρασμα αν θα επιλέξουμε στην εκτίμηση μας σταθερές επιδράσεις (επιλογή fe) ή τυχαίες επιδράσεις (επιλογή re) εκτελούμε τον έλεγχο hausman. Η ακολουθία των εντολών στο STATA είναι η εξής:

```
xtreg vrY vrX1 vrX2 vrX3, fe
estimates store fixed
xtreg vrY vrX1 vrX2 vrX3, re
estimates store random
hausman fixed random
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της ακολουθίας των εντολών. Αν εστιάσουμε στο τέλος των αναφερθέντων αποτελεσμάτων θα παρατηρήσουμε τα στατιστικά αποτελέσματα του ελέγχου hausman. Έστω για παράδειγμα ότι είναι τα ακόλουθα:

```
                b = Consistent under H0 and Ha; obtained from xtreg.
 B = Inconsistent under Ha, efficient under H0; obtained from xtreg.

Test of H0: Difference in coefficients not systematic

    chi2(2) = (b-B)'[(V_b-V_B)^(-1)](b-B)
            =   5.99
Prob > chi2 = 0.0500
```

Η μηδενική υπόθεση είναι ότι πρέπει να επιλεχθούν οι τυχαίες επιδράσεις απορρίπτεται (δεν γίνεται αποδεκτή) αν το p-value του στατιστικού X^2 είναι μικρότερο του 5% (δηλαδή θα δείτε στο report το εξής: prob>chi2 = XXX < 0.05). Στο παραπάνω παράδειγμα θα επιλεχθούν οι τυχαίες επιδράσεις.

Επιλογή cluster(Firm_id)

Έχουμε ήδη αναλύσει το τρόπο με τον οποίο αντιμετωπίζεται το πρόβλημα της ετεροσκεδαστικότητας (βλέπε επιλογή robust). Αναφορικά με την αυτοσυσχέτιση και εφόσον διενεργηθεί ο έλεγχος Breusch–Godfrey όπου διαπιστωθεί η ύπαρξη προβλήματος αυτοσυσχέτισης, τότε ο ερευνητής δύναται να το αντιμετωπίσει με σχετική επιλογή κατά τη σύνταξη της εντολής xtreg.

Έστω ότι επιθυμούμε να εκτιμήσουμε ένα υπόδειγμα (πολυμεταβλητής) παλινδρόμησης που αφορά μια εξαρτημένη μεταβλητή (vrY) και ένα πλήθος ανεξάρτητων μεταβλητών (έστω τρεις με την ονομασία vrX1, vrX2 και vrX3 αντίστοιχα) με δεδομένα πάνελ. Έστω ότι διαπιστώθηκε πρόβλημα αυτοσυσχέτισης. Σε αυτή την περίπτωση ορίζουμε μεταβλητή Firm_id η οποία αναφέρεται στο μοναδικό κωδικό που διακρίνεται η κάθε εταιρεία στο δείγμα των δεδομένων. Εκτελούμε την εντολή xtreg με την επιλογή cluster(Firm_id):

```
xtreg vrY vrX1 vrX2 vrX3, cluster(Firm_id)
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής.

Συνδυάζοντας επιλογές στην εκτέλεση της εντολής xtreg

Οι επιλογές fe, re, robust, cluster(Firm_id) είναι δυνατόν να χρησιμοποιηθούν συνδυασμένα ανάλογα με τη περίπτωση. Έστω ότι επιθυμούμε να εκτιμήσουμε ένα υπόδειγμα (πολυμεταβλητής) παλινδρόμησης που αφορά μια εξαρτημένη μεταβλητή (vrY) και ένα πλήθος ανεξάρτητων μεταβλητών (έστω τρεις με την ονομασία vrX1, vrX2 και vrX3 αντίστοιχα) με δεδομένα πάνελ. Ο παρακάτω πίνακας δίδει διάφορες επιλογές κατά τηνς εκτέλεση της εντολής xtreg.

Πίνακας 2: Συνδυασμός Επιλογών για την Εκτέλεση της Εντολής xtreg

Επιδράσεις	Ετεροσκεδα-στικότητα	Αυτοσυ-σχέτιση	Σύνταξη
Σταθερές	Όχι	Όχι	xtreg vrY vrX1 vrX2 vrX3, fe
Σταθερές	Ναι	Όχι	xtreg vrY vrX1 vrX2 vrX3, fe robust
Σταθερές	Όχι	Ναι	xtreg vrY vrX1 vrX2 vrX3, fe cluster(Firm_id)
Σταθερές	Ναι	Ναι	xtreg vrY vrX1 vrX2 vrX3, fe robust cluster(Firm_id)
Τυχαίες	Όχι	Όχι	xtreg vrY vrX1 vrX2 vrX3, re
Τυχαίες	Ναι	Όχι	xtreg vrY vrX1 vrX2 vrX3, re robust
Τυχαίες	Όχι	Ναι	xtreg vrY vrX1 vrX2 vrX3, re cluster(Firm_id)
Τυχαίες	Ναι	Ναι	xtreg vrY vrX1 vrX2 vrX3, re robust cluster(Firm_id)

4.4. Διάφορες εντολές εκτίμησης μοντέλων διακριτών μεταβλητών στο **STATA**

Εντολή logit

Στη λογαριθμιστική παλινδρόμηση (logit) η εξαρτημένη μεταβλητή ενός υποδείγματος παλινδρόμησης είναι ο λογάριθμος της σχέσης P/1-P (λόγου της συχνότητας-πιθανότητας εμφάνισης του γεγονότος προς τη συχνότητα – πιθανότητα της μη εμφάνισής του). Έστω ότι επιθυμούμε να εκτιμήσουμε ένα υπόδειγμα (πολυμεταβλητής) λογαριθμικής παλινδρόμησης που αφορά μια εξαρτημένη μεταβλητή (vrY) και ένα πλήθος ανεξάρτητων μεταβλητών (έστω τρεις με την ονομασία vrX1, vrX2 και vrX3 αντίστοιχα). Τούτο επιτυγχάνεται με την εντολή logit και η σύνταξη της εντολής στο **STATA** θα έχει ως εξής:

```
logit vrY vrX1 vrX2 vrX3, fe
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής.

Εντολή cmxtmixlogit

Στη λογαριθμιστική παλινδρόμηση (logit) είναι δυνατόν να εκτελεσθεί και για δεδομένα τύπου πανελ. Αντί της εντολής logit εκτελείται την εντολή cmxtmixlogit

4.5. Εξαγωγή αποτελεσμάτων εκτίμησης μοντέλων παλινδρόμησης από το **STATA**

Εντολή outreg2

Η εξαγωγή αποτελεσμάτων εκτίμησης μοντέλων παλινδρόμησης από το **STATA** μπορεί να γίνει με τη χρήση της εντολής outreg2. Η εντολή αυτή όμως θα πρέπει να εγκατασταθεί στο **STATA** και για το λόγο αυτό θα πρέπει να εκτελεσθεί ο ακόλουθος κώδικας στο **STATA**:

```
ssc install outreg2
```

Μετά την εκτέλεση της εντολής regress ή xtreg εκτελείται η εντολή outreg2 και η γενική σύνταξη της έχει ως εξής:

```
outreg2 using ΟΝΟΜΑ_ΑΡΧΕΙΟΥ.doc, replace
```

Όπου στη θέση ΟΝΟΜΑ_ΑΡΧΕΙΟΥ δίνεται μία ονομασία με λατινικούς χαρακτήρες. Προσοχή γιατί η επιλογή replace θα αντικαταστήσει προϋπάρχουν αρχείο με το ίδιο όνομα.

Εάν θέλετε να εξαγάγετε τον πίνακα αποτελεσμάτων στο Excel, χρησιμοποιήστε την επέκταση *.xls αντί να χρησιμοποιήσετε *.doc

Έστω ότι θέλετε να εκτιμήσετε ένα υπόδειγμα (πολυμεταβλητής) γραμμικής παλινδρόμησης με δεδομένα πάνελ που αφορά μια εξαρτημένη μεταβλητή (vrY) και ένα πλήθος ανεξάρτητων μεταβλητών (έστω τρεις με την ονομασία vrX1, vrX2 και vrX3

αντίστοιχα) με δεδομένα πάνελ και να εξάγεται τα αποτελέσματα σε ένα αρχείο με την ονομασία results, τότε η ακολουθία εντολών στο **STATA** θα έχει ως εξής:

```
xtreg vrY vrX1 vrX2 vrX3
outreg2 using results.doc, replace
```

Στο παράθυρο Results θα εμφανισθεί το αποτέλεσμα της εντολής.

Μπορούμε να προσθέσουμε μια νέα στήλη σε έναν πίνακα αποτελεσμάτων χρησιμοποιώντας την επιλογή append.

Η εντολή outreg2 μπορεί να εκτελεσθεί και για κατηγορίες δεδομένων. Για παράδειγμα, αν στα δεδομένα μας υπάρχει μία μεταβλητή με ονομασία vrCAT η οποία κατηγοριοποιεί τα δεδομένα μας σε πέντε μεγάλες κατηγορίες με βάση κάποιο κριτήριο (όπως για παράδειγμα το μέγεθος της εταιρείας σε πολύ μικρή, μικρή, μεσαία, μεγάλη και πολύ μεγάλη με δηλωτικές αριθμητικές τιμές 1, 2, 3, 4 και 5 αντίστοιχα). Με την επιλογή by δύναται να λάβουμε ξεχωριστούς πίνακες με τα αποτελέσματα εκτίμησης γραμμικής παλινδρόμησης για κάθε μία κατηγορία:

```
by vrCAT, sort: xtreg vrY vrX1 vrX2 vrX3
outreg2 using ONOMA_ΑΡΧΕΙΟΥ.doc, replace sum(log)
```

5. Παράρτημα Α: Οδηγός Σύνταξης Εντολής egen στο STATA

Title

stata.com

egen — Extensions to generate

Description

`egen` creates a new variable of the optionally specified storage type equal to the given function based on arguments of that function. The functions are specifically written for `egen`, as documented below or as written by users.

Quick start

Generate `newv1` for distinct groups of `v1` and `v2`, and create and apply value label `mylabel`

```
egen newv1 = group(v1 v2), label(mylabel)
```

Generate `newv2` equal to the minimum of `v1`, `v2`, and `v3` for each observation

```
egen newv2 = rowmin(v1 v2 v3)
```

Generate `newv3` equal to the overall sum of `v1`

```
egen newv3 = total(v1)
```

Same as above, but calculate total within each level of `catvar`

```
egen newv3 = total(v1), by(catvar)
```

Generate `newv4` equal to the number of nonmissing numeric values across `v1`, `v2`, and `v3` for each observation

```
egen newv4 = rownonmiss(v1 v2 v3)
```

Same as above, but allow string values

```
egen newv4 = rownonmiss(v1 v2 v3), strok
```

Generate `newv5` as the concatenation of numeric `v1` and string `v4` separated by a space

```
egen newv5 = concat(v1 v4), punct(" ")
```

Menu

Data > Create or change data > Create new variable (extended)

Syntax

egen [*type*] *newvar* = *fcn*(*arguments*) [*if*] [*in*] [, *options*]

by is allowed with some of the egen functions, as noted below.

Depending on *fcn*, *arguments* refers to an expression, *varlist*, or *numlist*, and the *options* are also *fcn* dependent. *fcn* and its dependencies are listed below.

anycount(*varlist*), values(*integer numlist*)
may not be combined with by. It returns the number of variables in *varlist* for which values are equal to any integer value in a supplied *numlist*. Values for any observations excluded by either if or in are set to 0 (not missing). Also see anyvalue(*varname*) and anymatch(*varlist*).

anymatch(*varlist*), values(*integer numlist*)
may not be combined with by. It is 1 if any variable in *varlist* is equal to any integer value in a supplied *numlist* and 0 otherwise. Values for any observations excluded by either if or in are set to 0 (not missing). Also see anyvalue(*varname*) and anycount(*varlist*).

anyvalue(*varname*) , values(*integer numlist*)
may not be combined with by. It takes the value of *varname* if *varname* is equal to any integer value in a supplied *numlist* and is missing otherwise. Also see anymatch(*varlist*) and anycount(*varlist*).

concat(*varlist*) [, format(%*fmt*) decode maxlength(#) punct(*pchars*)]
may not be combined with by. It concatenates *varlist* to produce a string variable. Values of string variables are unchanged. Values of numeric variables are converted to string, as is, or are converted using a numeric format under the format(%*fmt*) option or decoded under the decode option, in which case maxlength() may also be used to control the maximum label length used. By default, variables are added end to end: punct(*pchars*) may be used to specify punctuation, such as a space, punct(" "), or a comma, punct(,).

count(*exp*) (allows by *varlist*:)
creates a constant (within *varlist*) containing the number of nonmissing observations of *exp*. Also see rownonmiss() and rowmiss().

cut(*varname*), { at(*numlist*) | group(#) } [icodes label]
may not be combined with by. Either at() or group() must be specified. When at() is specified, it creates a new categorical variable coded with the left-hand ends of the grouping intervals specified in the at() option. When group() is specified, groups of roughly equal frequencies are created.

at(*numlist*) with *numlist* in ascending order supplies the breaks for the groups. *newvar* is set to missing for observations with *varname* less than the first number specified in at() and for observations with *varname* greater than or equal to the last number specified in at().

group(#) specifies the number of equal-frequency grouping intervals when breaks are not specified. Specifying this option automatically invokes icodes.

icodes requests that the codes 0, 1, 2, etc., be used in place of the left-hand ends of the intervals.

label requests that the integer-coded values of the grouped variable be labeled with the left-hand ends of the grouping intervals. Specifying this option automatically invokes icodes.

diff(*varlist*)
may not be combined with by. It creates an indicator variable equal to 1 if the variables in *varlist* are not equal and 0 otherwise.

ends(*strvar*) [, punct(*pchars*) trim [head|last|tail]]
may not be combined with by. It gives the first "word" or head (with the head option), the last "word" (with the last option), or the remainder or tail (with the tail option) from string variable *strvar*.

head, last, and tail are determined by the occurrence of *pchars*, which is by default one space (" ").

The head is whatever precedes the first occurrence of *pchars*, or the whole of the string if it does not occur. For example, the head of "frog toad" is "frog" and that of "frog" is "frog". With punct(,), the head of "frog,toad" is "frog".

The last word is whatever follows the last occurrence of *pchars* or is the whole of the string if a space does not occur. The last word of "frog toad newt" is "newt" and that of "frog" is "frog". With punct(,), the last word of "frog,toad" is "toad".

The remainder or tail is whatever follows the first occurrence of *pchars*, which will be the empty string "" if *pchars* does not occur. The tail of "frog toad newt" is "toad newt" and that of "frog" is "". With punct(,), the tail of "frog,toad" is "toad".

The trim option trims any leading or trailing spaces.

fill(*numlist*)
may not be combined with by. It creates a variable of ascending or descending numbers or complex repeating patterns. *numlist* must contain at least two numbers and may be specified using standard *numlist* notation; see [U] 11.1.8 numlist. if and in are not allowed with fill().

group(*varlist*) [, missing autotype label[(*lblname*[, replace truncate(#)])]]
may not be combined with by. It creates one variable taking on values 1, 2, ... for the groups formed by *varlist*. *varlist* may contain numeric variables, string variables, or a combination of the two. The order of the groups is that of the sort order of *varlist*.

missing indicates that missing values in *varlist* (either . or "") are to be treated like any other value when assigning groups. By default, any observation with a missing value is assigned to the group with *newvar* equal to missing (.).

autotype specifies that *newvar* be the smallest *type* possible to hold the integers generated. The resulting *type* will be byte, int, long, or double.

label or label(*lblname*) creates a value label for *newvar*. The integers in *newvar* are labeled with the values of *varlist* or their value labels, if they exist. label(*lblname*) specifies *lblname* as the name of the value label. If label alone is specified, the name of the value label is *newvar*. label(..., replace) allows an existing value label to be redefined. label(..., truncate(#)) truncates the values contributed to the label from each variable in *varlist* to the length specified by the integer argument #.

iqr(*exp*)[, autotype] (allows by *varlist*:)
creates a constant (within *varlist*) containing the interquartile range of *exp*. autotype specifies that *newvar* be the smallest *type* possible to hold the result. The resulting *type* will be byte, int, long, or double. Also see pctile().

kurt(*exp*) (allows by *varlist*:)
returns the kurtosis (within *varlist*) of *exp*.

mad(*exp*) (allows by *varlist*:)
returns the median absolute deviation from the median (within *varlist*) of *exp*.

`max(`*exp*`)` [`, missing`] (allows by *varlist*:)
creates a constant (within *varlist*) containing the maximum value of *exp*. `missing` indicates that missing values be treated like other values.

`mdev(`*exp*`)` (allows by *varlist*:)
returns the mean absolute deviation from the mean (within *varlist*) of *exp*.

`mean(`*exp*`)` (allows by *varlist*:)
creates a constant (within *varlist*) containing the mean of *exp*.

`median(`*exp*`)` [`, autotype`] (allows by *varlist*:)
creates a constant (within *varlist*) containing the median of *exp*. `autotype` specifies that *newvar* be the smallest *type* possible to hold the result. The resulting *type* will be `byte`, `int`, `long`, or `double`. Also see `pctile()`.

`min(`*exp*`)` [`, missing`] (allows by *varlist*:)
creates a constant (within *varlist*) containing the minimum value of *exp*. `missing` indicates that missing values be treated like other values.

`mode(`*varname*`)` [`, minmode maxmode nummode(`*integer*`) missing`] (allows by *varlist*:)
produces the mode (within *varlist*) for *varname*, which may be numeric or string. The mode is the value occurring most frequently. If two or more modes exist or if *varname* contains all missing values, the mode produced will be a missing value. To avoid this, the `minmode`, `maxmode`, or `nummode()` option may be used to specify choices for selecting among the multiple modes. `minmode` returns the lowest value, and `maxmode` returns the highest value. `nummode(#)` returns the #th mode, counting from the lowest up. `missing` indicates that missing values be treated like other values.

`pc(`*exp*`)` [`, prop`] (allows by *varlist*:)
returns *exp* (within *varlist*) scaled to be a percentage of the total, between 0 and 100. The `prop` option returns *exp* scaled to be a proportion of the total, between 0 and 1.

`pctile(`*exp*`)` [`, p(#) autotype`] (allows by *varlist*:)
creates a constant (within *varlist*) containing the #th percentile of *exp*. If `p(#)` is not specified, 50 is assumed, meaning medians. `autotype` specifies that *newvar* be the smallest *type* possible to hold the result. The resulting *type* will be `byte`, `int`, `long`, or `double`. Also see `median()`.

`rank(`*exp*`)` [`, field|track|unique`] (allows by *varlist*:)
creates ranks (within *varlist*) of *exp*; by default, equal observations are assigned the average rank. The `field` option calculates the field rank of *exp*: the highest value is ranked 1, and there is no correction for ties. That is, the field rank is 1 + the number of values that are higher. The `track` option calculates the track rank of *exp*: the lowest value is ranked 1, and there is no correction for ties. That is, the track rank is 1 + the number of values that are lower. The `unique` option calculates the unique rank of *exp*: values are ranked 1, . . . , #, and values and ties are broken arbitrarily. Two values that are tied for second are ranked 2 and 3.

`rowfirst(`*varlist*`)`
may not be combined with by. It gives the first nonmissing value in *varlist* for each observation (row). If all values in *varlist* are missing for an observation, *newvar* is set to missing for that observation.

`rowlast(`*varlist*`)`
may not be combined with by. It gives the last nonmissing value in *varlist* for each observation (row). If all values in *varlist* are missing for an observation, *newvar* is set to missing for that observation.

rowmax(*varlist*)
may not be combined with by. It gives the maximum value (ignoring missing values) in *varlist* for each observation (row). If all values in *varlist* are missing for an observation, *newvar* is set to missing for that observation.

rowmean(*varlist*)
may not be combined with by. It creates the (row) means of the variables in *varlist*, ignoring missing values. For example, if three variables are specified and, in some observations, one of the variables is missing, in those observations *newvar* will contain the mean of the two variables that do exist. Other observations will contain the mean of all three variables. If all values in *varlist* are missing for an observation, *newvar* is set to missing for that observation.

rowmedian(*varlist*)
may not be combined with by. It gives the (row) median of the variables in *varlist*, ignoring missing values. If all values in *varlist* are missing for an observation, *newvar* is set to missing for that observation. Also see rowpctile().

rowmin(*varlist*)
may not be combined with by. It gives the minimum value in *varlist* for each observation (row). If all values in *varlist* are missing for an observation, *newvar* is set to missing for that observation.

rowmiss(*varlist*)
may not be combined with by. It gives the number of missing values in *varlist* for each observation (row).

rownonmiss(*varlist*)[, strok]
may not be combined with by. It gives the number of nonmissing values in *varlist* for each observation (row).

String variables may not be specified unless the strok option is also specified. When strok is specified, *varlist* may contain a mixture of string and numeric variables.

rowpctile(*varlist*) [, p(#)]
may not be combined with by. It gives the #th percentile of the variables in *varlist*, ignoring missing values. If p() is not specified, p(50) is assumed, meaning medians. If all values in *varlist* are missing for an observation, *newvar* is set to missing for that observation. Also see rowmedian().

rowsd(*varlist*)
may not be combined with by. It creates the (row) standard deviations of the variables in *varlist*, ignoring missing values. If all values in *varlist* are missing for an observation, *newvar* is set to missing for that observation.

rowtotal(*varlist*) [, missing]
may not be combined with by. It creates the (row) sum of the variables in *varlist*, treating missing values as 0. If missing is specified and all values in *varlist* are missing for an observation, *newvar* is set to missing for that observation.

sd(*exp*) (allows by *varlist*:)
creates a constant (within *varlist*) containing the standard deviation of *exp*. Also see mean().

seq() [, from(#) to(#) block(#)] (allows by *varlist*:)
returns integer sequences. Values start from from() (default 1) and increase to to() (the default is the maximum number of values) in blocks (default size 1). If to() is less than the maximum number, sequences restart at from(). Numbering may also be separate within groups defined by *varlist* or decreasing if to() is less than from(). Sequences depend on the sort order of observations, following three rules: 1) observations excluded by if or in are not

counted; 2) observations are sorted by *varlist*, if specified; and 3) otherwise, the order is that when called. No *arguments* are specified.

`skew(`*exp*`)` (allows `by` *varlist*`:`)
returns the skewness (within *varlist*) of *exp*.

`std(`*exp*`)` [`, mean(#) sd(#)`] (allows `by` *varlist*`:`)
creates the standardized values (within *varlist*) of *exp*. The options specify the desired mean and standard deviation. The default is `mean(0)` and `sd(1)`, producing a variable with mean 0 and standard deviation 1 (within each group defined by *varlist*).

`tag(`*varlist*`)` [`, missing`]
may not be combined with `by`. It tags just one observation in each distinct group defined by *varlist*. When all observations in a group have the same value for a summary variable calculated for the group, it will be sufficient to use just one value for many purposes. The result will be 1 if the observation is tagged and never missing, and 0 otherwise. Values for any observations excluded by either `if` or `in` are set to 0 (not missing). Hence, if `tag` is the variable produced by `egen tag = tag(`*varlist*`)`, the idiom `if tag` is always safe. `missing` specifies that missing values of *varlist* may be included.

`total(`*exp*`)` [`, missing`] (allows `by` *varlist*`:`)
creates a constant (within *varlist*) containing the sum of *exp* treating missing as 0. If `missing` is specified and all values in *exp* are missing, *newvar* is set to missing. Also see `mean()`.

Remarks and examples

stata.com

Remarks are presented under the following headings:

See Mitchell (2020) for numerous examples using `egen`.

Summary statistics

The functions `count()`, `iqr()`, `kurt()`, `mad()`, `max()`, `mdev()`, `mean()`, `median()`, `min()`, `mode()`, `pc()`, `pctile()`, `sd()`, `skew()`, and `total()` create variables containing summary statistics. These functions take a `by ...:` prefix and, if specified, calculate the summary statistics within each by-group.

▷ Example 1: Without the by prefix

Without the `by` prefix, the result produced by these functions is a constant for every observation in the data. For instance, we have data on cholesterol levels (`chol`) and wish to have a variable that, for each patient, records the deviation from the average across all patients:

```
. use https://www.stata-press.com/data/r18/egenxmpl
. egen avg = mean(chol)
. generate deviation = chol - avg
```

◁

▷ Example 2: With the by prefix

These functions are most useful when the by prefix is specified. For instance, assume that our dataset includes dcode, a hospital–patient diagnostic code, and los, the number of days that the patient remained in the hospital. We wish to obtain the deviation in length of stay from the median for all patients having the same diagnostic code:

```
. use https://www.stata-press.com/data/r18/egenxmpl2, clear
. by dcode, sort: egen medstay = median(los)
. generate deltalos = los - medstay
```

◁

▷ Example 3: sum() function and egen total()

Distinguish carefully between Stata's sum() function and egen's total() function. Stata's sum() function creates the running sum, whereas egen's total() function creates a constant equal to the overall sum, for example,

```
. clear
. set obs 5
Number of observations (_N) was 0, now 5.
. generate a = _n
. generate sum1 = sum(a)
. egen sum2 = total(a)
. list
```

	a	sum1	sum2
1.	1	1	15
2.	2	3	15
3.	3	6	15
4.	4	10	15
5.	5	15	15

◁

Definitions of egen summary functions

The definitions and formulas used by egen summary functions are the same as those used by summarize; see [R] summarize. For comparison with summarize, mean() and sd() correspond to the mean and standard deviation. total() is the numerator of the mean, and count() is its denominator. min() and max() correspond to the minimum and maximum. median()—or, equally well, pctile() with p(50)—is the median. pctile() with p(5) refers to the 5th percentile, and so on. iqr() is the difference between the 75th and 25th percentiles.

The mode is the most common value of a dataset, whether it contains numeric or string variables. It is perhaps most useful for categorical variables (whether defined by integers or strings) or for other integer-valued values, but mode() can be applied to variables of any type. Nevertheless, the modes of continuous (or nearly continuous) variables are perhaps better estimated either from inspection of a graph of a frequency distribution or from the results of some density estimation (see [R] kdensity).

Missing values need special attention. `egen` *newvar* `= mode(`*varname*`)` calculates the mode of all nonmissing observations, and the variable *newvar* containing the mode is filled in for all observations, even those for which *varname* is missing (except for observations excluded using an `if` or `in` statement). This allows use of `mode()` as a simple way to impute categorical variables.

Missing values are by default excluded from the determination of modes (whether missing is defined by the period `[.]` or extended missing values `[.a, .b, ..., .z]` for numeric variables or the empty string `[""]` for string variables). However, missing may be the most common value in a variable, and you want `mode()` to report this value as the mode. To include missing values as possible values for the mode, use the `missing` option. See *Missing values* below for more on missing values.

`mad()` and `mdev()` produce alternative measures of spread. The median absolute deviation from the median and even the mean deviation will both be more resistant than the standard deviation to heavy tails or outliers, in particular from distributions with heavier tails than the normal or Gaussian. The first measure was named the MAD by Andrews et al. (1972) but was already known to K. F. Gauss in 1816, according to Hampel et al. (1986). For more historical and statistical details, see David (1998) and Wilcox (2003, 72–73).

Missing values

Missing values in the argument to `egen` functions (typically, *varname*, an expression, or *varlist*) are generally handled in one of three ways. Functions that calculate a single statistic for *varname* or an expression (for example, `mean()` and `total()`) fill in the result for all observations, including those for which *varname* or the expression is missing.

Functions that calculate results that potentially differ observation by observation (for example, `group()` and `rank()`) generally generate missing values for the result for observations where *varname* or the expression is missing.

Functions that take *varlist* (for example, `rowmean()`) generally generate a missing value for the result only when every variable in *varlist* is missing for that observation.

▷ Example 4: How missing values are handled

Here's an example of how `mean()` handles missing values.

```
. use https://www.stata-press.com/data/r18/egenxmpl1, clear
. egen y = mean(x)
. list x y
```

	x	y
1.	0	3
2.	5	3
3.	2	3
4.	5	3
5.	3	3
6.	.	3
7.	.a	3

The result `y` is filled in for all observations, including the 6th and 7th observations where `x` is missing. If you do not want this behavior, you can explicitly exclude missing values using an `if` statement.

```
. egen z = mean(x) if !missing(x)
(2 missing values generated)
. list x z
```

	x	z
1.	0	3
2.	5	3
3.	2	3
4.	5	3
5.	3	3
6.	.	.
7.	.a	.

Other functions, such as `group()`, by default exclude missing values. If you want to treat missing values just like other values and let them be part of the enumerated groups as well, use the `missing` option.

```
. egen g1 = group(x)
(2 missing values generated)
. egen g2 = group(x), missing
. list x g1 g2
```

	x	g1	g2
1.	0	1	1
2.	5	4	4
3.	2	2	2
4.	5	4	4
5.	3	3	3
6.	.	.	5
7.	.a	.	6

With the `missing` option, the missing values "." and ".a" are placed in two distinct groups, the 5th and 6th groups, in the result g2.

Here's an example of how `rowmean()` and `rowtotal()` handle missing values.

```
. egen m = rowmean(x1 x2 x3 x4)
(1 missing value generated)
. egen t1 = rowtotal(x1 x2 x3 x4)
. egen t2 = rowtotal(x1 x2 x3 x4), missing
(1 missing value generated)
. list x1 x2 x3 x4 m t1 t2
```

	x1	x2	x3	x4	m	t1	t2
1.	2	6	4	8	5	20	20
2.	9	.	0	3	4	12	12
3.	.	.a	.b	2	2	2	2
4.	.	.a	3	6	4.5	9	9
5.	4	5	5	2	4	16	16
6.	7	8	4	5	6	24	24
7.	.b	.a	.	.	.	0	.

`rowmean()` uses all the nonmissing values to calculate the mean of a row, ignoring any missing values. In the first row, all four variables are nonmissing, so the result is the mean of these four values. In the second row, three variables are nonmissing, and the result is the mean of these three values. In the third row, only one variable is nonmissing, and the result is simply the mean of this one value, that is, the value itself.

`rowtotal()` is similar to `rowmean()`, except that by default the total is 0 when all four variables are missing. See the 7th observation in this example. The result `t1` is 0 in this case. If you want `rowtotal()` to behave like `rowmean()`, use the `missing` option. The result `t2` is produced with this option, and you can see it is missing for the 7th observation, just like the `rowmean()` result.

Several egen functions have a `missing` option. See *Syntax* for the description of what `missing` does with each function that has this option—or better yet create a simple example, and run the function with and without the `missing` option.

◁

Generating patterns

To create a sequence of numbers, simply "show" the `fill()` function how the sequence should look. It must be a linear progression to produce the expected results. Stata does not understand geometric progressions. To produce repeating patterns, you present `fill()` with the pattern twice in the *numlist*.

▷ Example 5: Sequences produced by fill()

Here are some examples of ascending and descending sequences produced by `fill()`:

```
. clear
. set obs 12
Number of observations (_N) was 0, now 12.
. egen i = fill(1 2)
. egen w = fill(100 99)
. egen x = fill(22 17)
. egen y = fill(1 1 2 2)
. egen z = fill(8 8 8 7 7 7)
. list, sep(4)
```

	i	w	x	y	z
1.	1	100	22	1	8
2.	2	99	17	1	8
3.	3	98	12	2	8
4.	4	97	7	2	7
5.	5	96	2	3	7
6.	6	95	-3	3	7
7.	7	94	-8	4	6
8.	8	93	-13	4	6
9.	9	92	-18	5	6
10.	10	91	-23	5	5
11.	11	90	-28	6	5
12.	12	89	-33	6	5

◁

▷ Example 6: Patterns produced by fill()

Here are examples of patterns produced by `fill()`:

```
. clear
. set obs 12
Number of observations (_N) was 0, now 12.
. egen a = fill(0 0 1 0 0 1)
. egen b = fill(1 3 8 1 3 8)
. egen c = fill(-3(3)6 -3(3)6)
. egen d = fill(10 20 to 50   10 20 to 50)
. list, sep(4)
```

	a	b	c	d
1.	0	1	-3	10
2.	0	3	0	20
3.	1	8	3	30
4.	0	1	6	40
5.	0	3	-3	50
6.	1	8	0	10
7.	0	1	3	20
8.	0	3	6	30
9.	1	8	-3	40
10.	0	1	0	50
11.	0	3	3	10
12.	1	8	6	20

◁

▷ Example 7: seq()

`seq()` creates a new variable containing one or more sequences of integers. It is useful mainly for quickly creating observation identifiers or automatically numbering levels of factors or categorical variables.

```
. clear
. set obs 12
```

In the simplest case,

```
. egen a = seq()
```

is just equivalent to the common idiom

```
. generate a = _n
```

a may also be obtained from

```
. range a 1 _N
```

(the actual value of _N may also be used).

In more complicated cases, `seq()` with option calls is equivalent to calls to the versatile functions `int` and `mod`.

```
. egen b = seq(), b(2)
```

produces integers in blocks of 2, whereas

```
. egen c = seq(), t(6)
```

restarts the sequence after 6 is reached.

```
. egen d = seq(), f(10) t(12)
```

shows that sequences may start with integers other than 1, and

```
. egen e = seq(), f(3) t(1)
```

shows that they may decrease.

The results of these commands are shown by

```
. list, sep(4)
```

	a	b	c	d	e
1.	1	1	1	10	3
2.	2	1	2	11	2
3.	3	2	3	12	1
4.	4	2	4	10	3
5.	5	3	5	11	2
6.	6	3	6	12	1
7.	7	4	1	10	3
8.	8	4	2	11	2
9.	9	5	3	12	1
10.	10	5	4	10	3
11.	11	6	5	11	2
12.	12	6	6	12	1

All of these sequences could have been generated in one line with `generate` and with the use of the `int` and `mod` functions. The variables `b` through `e` are obtained with

```
. gen b = 1 + int((_n - 1)/2)
. gen c = 1 + mod(_n - 1, 6)
. gen d = 10 + mod(_n - 1, 3)
. gen e = 3 - mod(_n - 1, 3)
```

Nevertheless, `seq()` may save users from puzzling out such solutions or from typing in the needed values.

In general, the sequences produced depend on the sort order of observations, following three rules:

1. observations excluded by `if` or `in` are not counted;
2. observations are sorted by *varlist*, if specified; and
3. otherwise, the order is that specified when `seq()` is called.

◁

The `fill()` and `seq()` functions are alternatives. In essence, `fill()` requires a minimal example that indicates the kind of sequence required, whereas `seq()` requires that the rule be specified through options. There are sequences that `fill()` can produce that `seq()` cannot, and vice versa. `fill()` cannot be combined with `if` or `in`, in contrast to `seq()`, which can.

Marking differences among variables

▷ Example 8: diff()

We have three measures of respondents' income obtained from different sources. We wish to create the variable `differ` equal to 1 for disagreements:

```
. use https://www.stata-press.com/data/r18/egenxmpl3, clear
. egen byte differ = diff(inc*)
. list if differ==1
```

	inc1	inc2	inc3	id	differ
10.	42,491	41,491	41,491	110	1
11.	26,075	25,075	25,075	111	1
12.	26,283	25,283	25,283	112	1
78.	41,780	41,780	41,880	178	1
100.	25,687	26,687	25,687	200	1
101.	25,359	26,359	25,359	201	1
102.	25,969	26,969	25,969	202	1
103.	25,339	26,339	25,339	203	1
104.	25,296	26,296	25,296	204	1
105.	41,800	41,000	41,000	205	1
134.	26,233	26,233	26,133	234	1

Rather than typing `diff(inc*)`, we could have typed `diff(inc1 inc2 inc3)`.

◁

Ranks

▷ Example 9: rank()

Most applications of `rank()` will be to one variable, but the argument *exp* can be more general, namely, an expression. In particular, `rank(-`*varname*`)` reverses ranks from those obtained by `rank(`*varname*`)`.

The default ranking and those obtained by using one of the `track`, `field`, and `unique` options differ principally in their treatment of ties. The default is to assign the same rank to tied values such that the sum of the ranks is preserved. The `track` option assigns the same rank but resembles the convention in track events; thus, if one person had the lowest time and three persons tied for second-lowest time, their ranks would be 1, 2, 2, and 2, and the next person(s) would have rank 5. The `field` option acts similarly except that the highest is assigned rank 1, as in field events in which the greatest distance or height wins. The `unique` option breaks ties arbitrarily: its most obvious use is assigning ranks for a graph of ordered values. See also `group()` for another kind of "ranking".

```
. use https://www.stata-press.com/data/r18/auto, clear
(1978 automobile data)
. keep in 1/10
(64 observations deleted)
. egen rank = rank(mpg)
. egen rank_r = rank(-mpg)
. egen rank_f = rank(mpg), field
```

```
. egen rank_t = rank(mpg), track
. egen rank_u = rank(mpg), unique
. egen rank_ur = rank(-mpg), unique
. sort rank_u
. list mpg rank*
```

	mpg	rank	rank_r	rank_f	rank_t	rank_u	rank_ur
1.	15	1	10	10	1	1	10
2.	16	2	9	9	2	2	9
3.	17	3	8	8	3	3	8
4.	18	4	7	7	4	4	7
5.	19	5	6	6	5	5	6
6.	20	6.5	4.5	4	6	6	4
7.	20	6.5	4.5	4	6	7	5
8.	22	8.5	2.5	2	8	8	2
9.	22	8.5	2.5	2	8	9	3
10.	26	10	1	1	10	10	1

Standardized variables

▷ Example 10: std()

We have a variable called age recording the median age in the 50 states. We wish to create the standardized value of age and verify the calculation:

```
. use https://www.stata-press.com/data/r18/states1, clear
(State data)
. egen stdage = std(age)
. summarize age stdage
```

Variable	Obs	Mean	Std. dev.	Min	Max
age	50	29.54	1.693445	24.2	34.7
stdage	50	6.41e-09	1	-3.153336	3.047044

```
. correlate age stdage
(obs=50)
```

	age	stdage
age	1.0000	
stdage	1.0000	1.0000

`summarize` shows that the new variable has a mean of approximately zero; 10^{-9} is the precision of a `float` and is close enough to zero for all practical purposes. If we wanted, we could have typed `egen double stdage = std(age)`, making `stdage` a double-precision variable, and the mean would have been 10^{-16}. In any case, `summarize` also shows that the standard deviation is 1. `correlate` shows that the new variable and the original variable are perfectly correlated.

We may optionally specify the mean and standard deviation for the new variable. For instance,

```
. egen newage1 = std(age), sd(2)
. egen newage2 = std(age), mean(2) sd(4)
. egen newage3 = std(age), mean(2)
. summarize age newage1-newage3
```

Variable	Obs	Mean	Std. dev.	Min	Max
age	50	29.54	1.693445	24.2	34.7
newage1	50	1.28e-08	2	-6.306671	6.094089
newage2	50	2	4	-10.61334	14.18818
newage3	50	2	1	-1.153336	5.047044

```
. correlate age newage1-newage3
(obs=50)
```

	age	newage1	newage2	newage3
age	1.0000			
newage1	1.0000	1.0000		
newage2	1.0000	1.0000	1.0000	
newage3	1.0000	1.0000	1.0000	1.0000

◁

Row functions

▷ Example 11: rowtotal()

`generate`'s `sum()` function creates the vertical, running sum of its argument, whereas `egen`'s `total()` function creates a constant equal to the overall sum. `egen`'s `rowtotal()` function, however, creates the horizontal sum of its arguments. They all treat missing as zero. However, if the `missing` option is specified with `total()` or `rowtotal()`, then *newvar* will contain missing values if all values of *exp* or *varlist* are missing.

```
. use https://www.stata-press.com/data/r18/egenxmpl4, clear
. egen hsum = rowtotal(a b c)
. generate vsum = sum(hsum)
. egen sum = total(hsum)
. list
```

	a	b	c	hsum	vsum	sum
1.	.	2	3	5	5	63
2.	4	.	6	10	15	63
3.	7	8	.	15	30	63
4.	10	11	12	33	63	63

◁

▷ Example 12: rowmean(), rowmedian(), rowpctile(), rowsd(), and rownonmiss()

`summarize` displays the mean and standard deviation of a variable across observations; program writers can access the mean in `r(mean)` and the standard deviation in `r(sd)` (see [R] **summarize**). egen's `rowmean()` function creates the means of observations across variables. `rowmedian()` creates the medians of observations across variables. `rowpctile()` returns the #th percentile of the variables specified in *varlist*. `rowsd()` creates the standard deviations of observations across variables. `rownonmiss()` creates a count of the number of nonmissing observations, the denominator of the `rowmean()` calculation:

```
. use https://www.stata-press.com/data/r18/egenxmpl4, clear
. egen avg = rowmean(a b c)
. egen median = rowmedian(a b c)
. egen pct25 = rowpctile(a b c), p(25)
. egen std = rowsd(a b c)
. egen n = rownonmiss(a b c)
. list
```

	a	b	c	avg	median	pct25	std	n
1.	.	2	3	2.5	2.5	2	.7071068	2
2.	4	.	6	5	5	4	1.414214	2
3.	7	8	.	7.5	7.5	7	.7071068	2
4.	10	11	12	11	11	10	1	3

◁

▷ Example 13: rowmiss()

`rowmiss()` returns k − `rownonmiss()`, where k is the number of variables specified. `rowmiss()` can be especially useful for finding casewise-deleted observations caused by missing values.

```
. use https://www.stata-press.com/data/r18/auto3, clear
(1978 automobile data)
. correlate price weight mpg
(obs=70)
```

	price	weight	mpg
price	1.0000		
weight	0.5309	1.0000	
mpg	-0.4478	-0.7985	1.0000

```
. egen excluded = rowmiss(price weight mpg)
. list make price weight mpg if excluded~=0
```

	make	price	weight	mpg
5.	Buick Electra	.	4,080	15
12.	Cad. Eldorado	14,500	3,900	.
40.	Olds Starfire	4,195	.	24
51.	Pont. Phoenix	.	3,420	.

◁

▷ Example 14: rowmin(), rowmax(), rowfirst(), and rowlast()

`rowmin()`, `rowmax()`, `rowfirst()`, and `rowlast()` return the minimum, maximum, first, or last nonmissing value, respectively, for the specified variables within an observation (row).

```
. use https://www.stata-press.com/data/r18/egenxmpl5, clear
. egen min = rowmin(x y z)
(1 missing value generated)
. egen max = rowmax(x y z)
(1 missing value generated)
. egen first = rowfirst(x y z)
(1 missing value generated)
. egen last = rowlast(x y z)
(1 missing value generated)
. list, sep(4)
```

	x	y	z	min	max	first	last
1.	-1	2	3	-1	3	-1	3
2.	.	-6	.	-6	-6	-6	-6
3.	7	.	-5	-5	7	7	-5
4.	.	.	.	.	.	.	.
5.	4	.	.	4	4	4	4
6.	.	.	8	8	8	8	8
7.	.	3	7	3	7	3	7
8.	5	-1	6	-1	6	5	6

◁

Categorical and integer variables

▷ Example 15: anyvalue(), anymatch(), and anycount()

`anyvalue()`, `anymatch()`, and `anycount()` are for categorical or other variables taking integer values. If we define a subset of values specified by an integer *numlist* (see [U] **11.1.8 numlist**), `anyvalue()` extracts the subset, leaving every other value missing; `anymatch()` defines an indicator variable (1 if in subset, 0 otherwise); and `anycount()` counts occurrences of the subset across a set of variables. Therefore, with just one variable, `anymatch(`*varname*`)` and `anycount(`*varname*`)` are equivalent.

With the `auto` dataset, we can generate a variable containing the high values of `rep78` and a variable indicating whether `rep78` has a high value:

```
. use https://www.stata-press.com/data/r18/auto, clear
(1978 automobile data)
. egen hirep = anyvalue(rep78), v(3/5)
(15 missing values generated)
. egen ishirep = anymatch(rep78), v(3/5)
```

Here it is easy to produce the same results with official Stata commands:

```
. generate hirep = rep78 if inlist(rep78,3,4,5)
. generate byte ishirep = inlist(rep78,3,4,5)
```

However, as the specification becomes more complicated or involves several variables, the egen functions may be more convenient.

◁

▷ Example 16: group()

group() maps the distinct groups of a varlist to a categorical variable that takes on integer values from 1 to the total number of groups. order of the groups is that of the sort order of *varlist*. The *varlist* may be of numeric variables, string variables, or a mixture of the two. The resulting variable can be useful for many purposes, including stepping through the distinct groups easily and systematically and cleaning up an untidy ordering. Suppose that the actual (and arbitrary) codes present in the data are 1, 2, 4, and 7, but we desire equally spaced numbers, as when the codes will be values on one axis of a graph. group() maps these to 1, 2, 3, and 4.

We have a variable agegrp that takes on the values 24, 40, 50, and 65, corresponding to age groups 18–24, 25–40, 41–50, and 51 and above. Perhaps we created this coding using the recode() function (see [U] 13.3 Functions and [U] 26 Working with categorical data and factor variables) from another age-in-years variable:

```
. generate agegrp=recode(age,24,40,50,65)
```

We now want to change the codes to 1, 2, 3, and 4:

```
. egen agegrp2 = group(agegrp)
```

◁

▷ Example 17: group() with missing values

We have two categorical variables, race and sex, which may be string or numeric. We want to use ir (see [R] Epitab) to create a Mantel–Haenszel weighted estimate of the incidence rate. ir, however, allows only one variable to be specified in its by() option. We type

```
. use https://www.stata-press.com/data/r18/egenxmpl6, clear
. egen racesex = group(race sex)
(2 missing values generated)
. ir deaths smokes pyears, by(racesex)
  (output omitted)
```

The new numeric variable, racesex, will be missing wherever race or sex is missing (meaning . for numeric variables and "" for string variables), so missing values will be handled correctly. When we list some of the data, we see

```
. list race sex racesex in 1/7, sep(0)
```

	race	sex	racesex
1.	White	Female	1
2.	White	Male	2
3.	Black	Female	3
4.	Black	Male	4
5.	Black	Male	4
6.	.	Female	.
7.	Black	.	.

group() began by putting the data in the order of the grouping variables and then assigned the numeric codes. Observations 6 and 7 were assigned to racesex = . because, in one case, race was not known, and in the other, sex was not known. (These observations were not used by ir.)

If we wanted the unknown groups to be treated just as any other category, we could have typed

```
. egen rs2 = group(race sex), missing
. list race sex rs2 in 1/7, sep(0)
```

	race	sex	rs2
1.	White	Female	1
2.	White	Male	2
3.	Black	Female	3
4.	Black	Male	4
5.	Black	Male	4
6.	.	Female	6
7.	Black	.	5

The resulting variable from `group()` does not have value labels. Therefore, the values carry no indication of meaning. Interpretation requires comparison with the original *varlist*. To get value labels, we specify the option `label`.

```
. egen rs3 = group(race sex), missing label
. list race sex rs3 in 1/7, sep(0)
```

	race	sex	rs3
1.	White	Female	White Female
2.	White	Male	White Male
3.	Black	Female	Black Female
4.	Black	Male	Black Male
5.	Black	Male	Black Male
6.	.	Female	. Female
7.	Black	.	Black .

The numeric values of the generated variable `rs3` are the same as `rs2`, but `rs3` has a value label that indicates the categories of `race` and `sex` that define the groups. The value label created by `group()` uses the actual values of the categorical variables or their value labels, if they exist. In this case, the categorical variables `race` and `sex` are numeric variables with value labels, so their value labels were used to create the value label for `rs3`. ◁

String variables

Concatenation of string variables is provided in Stata. In context, Stata understands the addition symbol `+` as specifying concatenation or adding strings end to end. `"soft" + "ware"` produces `"software"`, and given string variables `s1` and `s2`, `s1 + s2` indicates their concatenation.

The complications that may arise in practice include wanting 1) to concatenate the string versions of numeric variables and 2) to concatenate variables, together with some separator such as a space or a comma. Given numeric variables `n1` and `n2`,

```
. generate newstr = s1 + string(n1) + string(n2) + s2
```

shows how numeric values may be converted to their string equivalents before concatenation, and

```
. generate newstr = s1 + " " + s2 + " " + s3
```

shows how spaces may be added between variables. Stata will automatically assign the most appropriate data type for the new string variables.

▷ Example 18: concat()

`concat()` allows us to do everything in one line concisely.

```
. egen newstr = concat(s1 n1 n2 s2)
```

carries with it an implicit instruction to convert numeric values to their string equivalents, and the appropriate string data type is worked out within `concat()` by Stata's automatic promotion. Moreover,

```
. egen newstr = concat(s1 s2 s3), p(" ")
```

specifies that spaces be used as separators. (The default is to have no separation of concatenated strings.)

As an example of punctuation other than a space, consider

```
. egen fullname = concat(surname forename), p(", ")
```

Noninteger numerical values can cause difficulties, but

```
. egen newstr = concat(n1 n2), format(%9.3f) p(" ")
```

specifies the use of format `%9.3f`. This is equivalent to

```
. generate str1 newstr = ""
. replace newstr = string(n1,"%9.3f") + " " + string(n2,"%9.3f")
```

See [FN] **String functions** for more about `string()`.

◁

As a final flourish, the `decode` option instructs `concat()` to use value labels. With that option, the `maxlength()` option may also be used. For more details about `decode`, see [D] **encode**. Unlike the `decode` command, however, `concat()` uses `string(`*varname*`)`, not `""`, whenever values of *varname* are not associated with value labels, and the `format()` option, whenever specified, applies to this use of `string()`.

▷ Example 19: ends()

The `ends(`*strvar*`)` function is used for subdividing strings. The approach is to find specified separators by using the `strpos()` string function and then to extract what is desired, which either precedes or follows the separators, using the `substr()` string function.

By default, substrings are considered to be separated by individual spaces, so we will give definitions in those terms and then generalize.

The head of the string is whatever precedes the first space or is the whole of the string if no space occurs. This could also be called the first "word". The tail of the string is whatever follows the first space. This could be nothing or one or more words. The last word in the string is whatever follows the last space or is the whole of the string if no space occurs.

To clarify, let's look at some examples. The quotation marks here just mark the limits of each string and are not part of the strings.

	head	tail	last
"frog"	"frog"	""	"frog"
"frog toad"	"frog"	"toad"	"toad"
"frog toad newt"	"frog"	"toad newt"	"newt"
"frog toad newt"	"frog"	" toad newt"	"newt"
"frog toad newt"	"frog"	"toad newt"	"newt"

The main subtlety is that these functions are literal, so the tail of `"frog  toad newt"`, in which two spaces follow `"frog"`, includes the second of those spaces, and is thus `" toad newt"`. Therefore, you may prefer to use the `trim` option to trim the result of any leading or trailing spaces, producing `"toad newt"` in this instance.

The `punct(`*pchars*`)` option may be used to specify separators other than spaces. The general definitions of the `head`, `tail`, and `last` options are therefore interpreted in terms of whatever separator has been specified; that is, they are relative to the first or last occurrence of the separator in the string value. Thus, with `punct(,)` and the string `"Darwin, Charles Robert"`, the head is `"Darwin"`, and the tail and the last are both `" Charles Robert"`. Note again the leading space in this example, which may be trimmed with `trim`. The punctuation (here the comma, ",") is discarded, just as it is with one space.

pchars, the argument of `punct()`, will usually, but not always, be one character. If two or more characters are specified, these must occur together; for example, `punct(:;)` would mean that words are separated by a colon followed by a semicolon (that is, `:;`). It is not implied, in particular, that the colon and semicolon are alternatives. To do that, you would have to modify the programs presented here or resort to first principles by using `split`; see [D] **split**.

With personal names, the `head` or `last` option might be applied to extract surnames if strings were similar to `"Darwin, Charles Robert"` or `"Charles Robert Darwin"`, with the surname coming first or last. What then happens with surnames like `"von Neumann"` or `"de la Mare"`? `"von Neumann, John"` is no problem, if the comma is specified as a separator, but the `last` option is not intelligent enough to handle `"Walter de la Mare"` properly.

◁

Acknowledgments

The `cut()` function was written by David Clayton (retired) of the Cambridge Institute for Medical Research and Michael Hills (1934–2021) of the London School of Hygiene and Tropical Medicine.

Many of the other `egen` functions were written by Nicholas J. Cox of the Department of Geography at Durham University, UK, and coeditor of the *Stata Journal* and author of *Speaking Stata Graphics*.

References

Andrews, D. F., P. J. Bickel, F. R. Hampel, P. J. Huber, W. H. Rogers, and J. W. Tukey. 1972. *Robust Estimates of Location: Survey and Advances*. Princeton, NJ: Princeton University Press.

Cappellari, L., and S. P. Jenkins. 2006. Calculation of multivariate normal probabilities by simulation, with applications to maximum simulated likelihood estimation. *Stata Journal* 6: 156–189.

Cox, N. J. 2009. Speaking Stata: Rowwise. *Stata Journal* 9: 137–157.

——. 2014. Speaking Stata: Self and others. *Stata Journal* 14: 432–444.

——. 2020. Speaking Stata: More ways for rowwise. *Stata Journal* 20: 481–488.

——. 2021. Speaking Stata: Ordering or ranking groups of observations. *Stata Journal* 21: 818–837.

——. 2022. Speaking Stata: The largest five—A tale of tail values. *Stata Journal* 22: 446–459.

——. 2023. Speaking Stata: Replacing missing values: The easiest problems. *Stata Journal* 23: 884–896.

Cox, N. J., and C. B. Schechter. 2018. Speaking Stata: Seven steps for vexatious string variables. *Stata Journal* 18: 981–994.

David, H. A. 1998. Early sample measures of variability. *Statistical Science* 13: 368–377. https://doi.org/10.1214/ss/1028905831.

Gallup, J. L. 2019. Grade functions. *Stata Journal* 19: 459–476.

Hampel, F. R., E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. 1986. *Robust Statistics: The Approach Based on Influence Functions*. New York: Wiley.

Huber, C. 2014. How to simulate multilevel/longitudinal data. *The Stata Blog: Not Elsewhere Classified*. http://blog.stata.com/2014/07/18/how-to-simulate-multilevellongitudinal-data/.

Kohler, U., and J. Zeh. 2012. Apportionment methods. *Stata Journal* 12: 375–392.

Mitchell, M. N. 2020. *Data Management Using Stata: A Practical Handbook*. 2nd ed. College Station, TX: Stata Press.

Pinzon, E. 2015. Fixed effects or random effects: The Mundlak approach. *The Stata Blog: Not Elsewhere Classified*. http://blog.stata.com/2015/10/29/fixed-effects-or-random-effects-the-mundlak-approach/.

Rios-Avila, F. 2020. Recentered influence functions (RIFs) in Stata: RIF regression and RIF decomposition. *Stata Journal* 20: 51–94.

Salas Pauliac, C. H. 2013. group2: Generating the finest partition that is coarser than two given partitions. *Stata Journal* 13: 867–875.

Weiss, M. 2009. Stata tip 80: Constructing a group variable with specified group sizes. *Stata Journal* 9: 640–642.

Wilcox, R. R. 2003. *Applying Contemporary Statistical Techniques*. San Diego, CA: Academic Press.

Also see

[D] **collapse** — Make dataset of summary statistics

[D] **generate** — Create or change contents of variable

[U] **13.3 Functions**