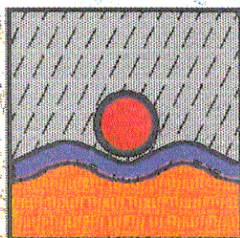




ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΤΜΗΜΑ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΥΔΑΤΙΚΩΝ ΠΟΡΩΝ,
ΥΔΡΑΥΛΙΚΩΝ ΚΑΙ ΘΑΛΑΣΣΙΩΝ ΕΡΓΩΝ

ΣΥΜΠΛΗΡΩΣΗ ΕΛΛΙΠΩΝ
ΥΔΡΟΜΕΤΕΩΡΟΛΟΓΙΚΩΝ ΧΡΟΝΟΣΕΙΡΩΝ
ΣΕ ΚΑΤΑΝΕΜΗΜΕΝΕΣ ΣΧΕΣΙΑΚΕΣ
ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

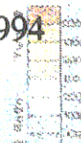


ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
Αντώνης Χριστοφίδης

ΕΠΗΒΛΕΠΩΝ
Δ. Κουτσογιάννης, Λέκτορας

Αθήνα, Φεβρουάριος 1994

ΕΛΛΑΔΑ ΓΕΩΓΡΑΦΙΚΟΣ ΧΑΡΤΗΣ
ΚΑΙΝΗΡΙΑ 1:500000



Η συμβολή μας στην προστασία του περιβάλλοντος

Συνήθως οι επιστημονικές εργασίες και οι τεχνικές εκθέσεις εκτυπώνονται μόνο από τη μια πλευρά κάθε φύλλου χαρτιού, αφήνοντας την από πίσω σελίδα λευκή. Αυτό γίνεται για λόγους ευκολίας, γιατί αν πρόκειται να εκτυπωθούν και οι δυο σελίδες, χρειάζεται περισσότερη φροντίδα κατά την εκτύπωση, για να γυρίζονται τα φύλλα από την άλλη πλευρά. Αν η εκτύπωση γίνει και από τις δυο πλευρές, τότε γίνεται οικονομία χαρτιού χωρίς να δυσχεραίνεται η ανάγνωση· αντίθετα μάλιστα, διευκολύνεται, γιατί κάθε στιγμή ο αναγνώστης έχει δυο σελίδες μπροστά του, αντί για μία.

Σ' αυτή την εργασία έχει γίνει αμφίπλευρη εκτύπωση, πράγμα που συνεπάγεται οικονομία 50%. Ο παραπάνω κόπος που απαιτείται για εκτύπωση από τις δυο πλευρές, με λίγη προσοχή και εξάσκηση, είναι μηδαμινός. *Ας εκτυπώνουμε λοιπόν τις εργασίες μας και από τις δυο πλευρές. Ας προστατέψουμε το περιβάλλον.*

Το σχέδιο του εξωφύλλου που παριστάνει τις καιρικές συνθήκες πάνω σε μια δισκέτα σχεδιάστηκε από την αρχιτέκτονα Άννα Πατρικίου και είναι σήμα του «Υδροσκοπίου».

Χριστοφίδης, Α.

Συμπλήρωση ελλειπών υδρομετεωρολογικών χρονοσειρών σε κατανεμημένες σχεσιακές βάσεις δεδομένων

551.57--dc

Φεβρουάριος 1994

Σύνοψη

Η συμπλήρωση ελλειπουσών τιμών και η επέκταση χρονοσειρών είναι σχετιζόμενες επεξεργασίες που γίνονται είτε για να διευκολύνουν άλλες επεξεργασίες είτε για να βελτιώσουν τα στατιστικά χαρακτηριστικά μιας χρονοσειράς. Σ' αυτή την εργασία εξετάζονται μερικές από τις κυριότερες μεθόδους που έχουν αναπτυχθεί ως σήμερα. Αυτές είναι η μέθοδος της μέσης τιμής, του αντιστρόφου της απόστασης, των υπερετήσιων λόγων, της γραμμικής παλινδρόμησης, της επέκτασης με διατήρηση της διασποράς (MOVE), του απλού μοντέλου AR(1) και PAR(1), και ενός πολυμεταβλητού στοχαστικού μοντέλου. Οι περισσότερες από αυτές υλοποιήθηκαν σε ένα πρόγραμμα υπολογιστή για την εξυπηρέτηση των αναγκών της εθνικής τράπεζας υδρολογικών και μετεωρολογικών δεδομένων με όνομα «Υδροσκόπιο», που βασίζεται σε μια κατανεμημένη σχεσιακή βάση δεδομένων. Ορισμένα σημαντικά στοιχεία του τρόπου υλοποίησης του προγράμματος περιγράφονται στην εργασία. Τέλος, παρουσιάζεται μια εφαρμογή σε ημερήσιες και σε μηνιαίες χρονοσειρές βροχοπτώσεων, με σύντομη αξιολόγηση των μεθόδων.

Abstract

Infilling of missing values and extension of records are related procedures, performed either to facilitate other procedures or to improve the statistical characteristics of a time series. In this thesis, some of the most important existing methods are examined. These are: mean value, reciprocal distance, normal ratio, linear regression, maintenance-of-variance extension (MOVE), use of an AR(1) and PAR(1) model, and use of a multivariate stochastic model. Most of these have been implemented in a computer program in order to serve the needs of the national databank of hydrological and meteorological data, named "Hydroscope", which is based on a distributed relational database. Some significant elements of the implementation are described. Finally, an application on daily rainfall time series, together with a brief evaluation of the methods, is presented.

Λέξεις κλειδιά

ΣΥΜΠΛΗΡΩΣΗ· ΕΛΛΕΙΠΟΥΣΕΣ ΤΙΜΕΣ· ΕΠΕΚΤΑΣΗ ΧΡΟΝΟΣΕΙΡΩΝ· ΓΡΑΜΜΙΚΗ ΠΑΛΙΝΔΡΟΜΗΣΗ· ΔΙΑΤΗΡΗΣΗ ΔΙΑΣΠΟΡΑΣ· ΑΝΑΛΥΣΗ ΧΡΟΝΟΣΕΙΡΩΝ· ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ· ΕΚΤΙΜΗΣΕΙΣ· ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ· ΥΔΡΟΛΟΓΙΑ· ΜΕΤΕΩΡΟΛΟΓΙΑ· ΣΤΑΤΙΣΤΙΚΗ ΑΝΑΛΥΣΗ

Key words

INFILLING; MISSING VALUES; RECORD EXTENSION; RECORD AUGMENTATION; LINEAR REGRESSION; MAINTENANCE OF VARIANCE; TIME SERIES ANALYSIS; DATA PROCESSING; ESTIMATES; DATABASE; HYDROLOGY; METEOROLOGY; STATISTICAL ANALYSIS

Μερικές φορές δεν μας πηγαίνουν στο μυαλό απλά ερωτήματα, κι αν μας πάνε είναι δύσκολο να απαντηθούν. Συμπληρώνουμε και επεκτείνουμε χρονοσειρές για να βελτιώσουμε την εκτίμηση των στατιστικών παραμέτρων. Βελτιώνουμε τις εκτιμήσεις των παραμέτρων για να έχουμε καλύτερα μοντέλα των υδρολογικών διεργασιών. Γιατί όμως χρειαζόμαστε αυτά τα μοντέλα; Ποιος είναι ο στόχος της ανάλυσης χρονοσειρών; Αποσβένεται ο τεράστιος όγκος έρευνας που γίνεται σ' όλο τον κόσμο πάνω σ' αυτόν τον κλάδο; Την απάντηση σ' αυτά τα ερωτήματα δεν βρήκα σε κανένα βιβλίο. Ο κ. Ν. Μαμάσης ήταν αυτός που μου τα εξήγησε, και σ' αυτόν οφείλονται οι πρώτες δύο παράγραφοι της εισαγωγής. Επίσης, μου αποσαφήνισε και πολλά άλλα θεωρητικά ζητήματα.

Ο κ. Ν. Παπακώστας, που μελέτησε τη βάση δεδομένων και το δίκτυο υπολογιστών του Υδροσκοπίου, ήταν μεγάλη πηγή γνώσης σε θέματα πληροφορικής, για μένα και όλους τους άλλους συνεργάτες του. Υπάρχει όμως ένα τίμημα που πληρώνει, όντας τόσο ικανός: βρίσκεται συνεχώς απέναντι σε μια σωρεία ερωτήσεων που κάνουν διάφοροι περίεργοι.

Ο κ. Γ. Τσακαλίας, υπεύθυνος ανάπτυξης εφαρμογών του Υδροσκοπίου, ήταν αυτός που με βοήθησε στην εκμάθηση της γλώσσας προγραμματισμού INGRES/Windows4GL, με καθοδήγησε σε πολλά ζητήματα και εξέφρασε πάρα πολλές ιδέες πάνω στην εφαρμογή συμπληρώσεων. Είναι επίσης αυτός που καθιέρωσε την ελληνική ονομασία της μεθόδου των υπερετήσιων λόγων σε μια παλιότερη εργασία του.

Η κ. Δ. Μπουλουγούρη, βιβλιοθηκονόμος του Τομέα Υδατικών Πόρων, με προμήθευσε με μερικά επιστημονικά άρθρα. Επίσης, μου δάνεισε το μυθιστόρημα-παραμύθι *Ο Άρχοντας των Δαχτυλιδιών* του Τζον Τόλκιν, που πλούτισε με ευχάριστη ποικιλία τις τελευταίες δύσκολες εβδομάδες ετοιμασίας της εργασίας μου. Μάλιστα, αποτέλεσμα των συζητήσεών μας σε βιβλιοθηκονομικά ζητήματα ήταν η διπλωματική μου εργασία να αποκτήσει, μεταξύ σοβαρού και αστείου, ταξινομικό αριθμό Dewey.

Ο κ. Δ. Κουτσογιάννης, επιβλέπων της διπλωματικής, δεν χρειάζεται συστάσεις. Μιλά ο αριθμός των φοιτητών που του ζητούν κάθε χρόνο διπλωματική εργασία.

Περίληψη

Στις υδρομετεωρολογικές χρονοσειρές υπάρχουν συχνά ελλείπουσες τιμές, οι οποίες δυσχεραίνουν ή καθιστούν αδύνατες πολλές επεξεργασίες. Για τη διευκόλυνση αυτών των επεξεργασιών είναι επιθυμητή η εκτίμηση των ελλειπουσών τιμών, ώστε να προκύπτουν συνεχείς χρονοσειρές χωρίς κενά. Επίσης, η επέκταση μιας χρονοσειράς με συσχέτιση με άλλες που προκύπτουν από γειτονικούς μετρητικούς σταθμούς συχνά οδηγεί σε καλύτερες εκτιμήσεις των στατιστικών παραμέτρων της υδρολογικής ή μετεωρολογικής διεργασίας απ' ό,τι η αρχική χρονοσειρά. Μέχρι σήμερα έχουν αναπτυχθεί πολλές μέθοδοι συμπλήρωσης ελλειπουσών τιμών και επέκτασης χρονοσειρών. Αυτή η εργασία ασχολείται με μερικές από τις σημαντικότερες μεθόδους, και εξειδικεύεται στην εφαρμογή τους στην πανελλήνια βάση υδρολογικών και μετεωρολογικών δεδομένων με όνομα «Υδροσκόπιο».

Πρώτα γίνεται μια ανασκόπηση μερικών από τις κυριότερες μεθόδους, που εξετάζονται σε τρεις κατηγορίες: Απλές κλασικές μέθοδοι, μέθοδοι βασισμένες σε γραμμική παλινδρόμηση, και μέθοδοι βασισμένες σε στοχαστικά μοντέλα. Οι απλές κλασικές μέθοδοι, που είναι πιο γρήγορες αλλά ανακριβείς, είναι η μέθοδος της μέσης τιμής, του αντιστρόφου της απόστασης, των υπερετήσιων λόγων, και μια παραλλαγή των δυο τελευταίων, η τροποποιημένη στάθμιση του μέσου όρου. Η γραμμική παλινδρόμηση είναι η κυριότερη κατηγορία, και στην ουσία περιλαμβάνει και τα στοχαστικά μοντέλα. Εξετάζονται απλή και πολλαπλή παλινδρόμηση με ή χωρίς τυχαίο όρο, και παραλλαγές με μηδενικό σταθερό όρο. Επίσης, αναφέρονται και οι μέθοδοι επέκτασης με διατήρηση της διασποράς (MOVE). Τέλος, όσον αφορά στα στοχαστικά μοντέλα, αναφέρεται η χρήση του μοντέλου AR(1) και του PAR(1) και ενός πολυμεταβλητού μοντέλου με αυτοσυσχέτιση και συσχέτιση με γειτονικούς σταθμούς.

Στη συνέχεια εκτίθεται ο τρόπος με τον οποίο οι μέθοδοι έχουν εφαρμοστεί στον υπολογιστή για τις ανάγκες του «Υδροσκόπιου». Γίνεται μια σύντομη αναφορά στο σχήμα της βάσης δεδομένων, αιτιολογείται η επιλογή των γλωσσών προγραμματισμού που έχουν χρησιμοποιηθεί, περιγράφονται ορισμένα καίρια σημεία σχετικά με τη διαχείριση και τη διακίνηση των δεδομένων, και επεξηγείται ο τρόπος με τον οποίο η εφαρμογή εφαρμόζει τις μεθόδους συμπλήρωσης.

Τέλος, επιχειρείται μια σύντομη σύγκριση των μεθόδων με εφαρμογή σε χρονοσειρές ημερήσιων και μηνιαίων βροχοπτώσεων. Η σύγκριση γίνεται εξετάζοντας το συντελεστή απόδοσης των εκτιμήσεων. Καλύτερη αποδεικνύεται η μέθοδος της γραμμικής παλινδρόμησης, αν και προκύπτουν ορισμένα ερωτηματικά σχετικά με τις παραμέτρους ή τις παραλλαγές που συνιστώνται σε κάθε περίπτωση. Τονίζεται ότι η μέθοδος που δίνει το μέγιστο συντελεστή απόδοσης (ελάχιστο τετραγωνικό σφάλμα εκτίμησης) δεν είναι απαραίτητα η καλύτερη· ανάλογα με το στόχο που εξυπηρετεί η συμπλήρωση, μπορεί να χρειάζεται να εξεταστούν και άλλες παράμετροι, όπως ο εκτιμητής της μέσης τιμής ή της διασποράς.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΙΣΑΓΩΓΗ	9
Η ανάγκη για συμπλήρωση ελλειπουσών τιμών των χρονοσειρών	9
Υπενθύμιση της ορολογίας	10
Ανελξεις και χρονοσειρές	10
Ελλειπείς χρονοσειρές και χρονοσειρές αναφοράς	11
ΚΕΦΑΛΑΙΟ 1: ΑΝΑΣΚΟΠΗΣΗ ΤΩΝ ΜΕΘΟΔΩΝ ΣΥΜΠΛΗΡΩΣΗΣ	12
1.1. Χρησιμοποιούμενοι συμβολισμοί	12
Βασικοί συμβολισμοί για όλες τις μεθόδους	12
Περισσότεροι συμβολισμοί για τις μεθόδους της γραμμικής παλινδρόμησης	12
1.2. Απλές κλασικές μέθοδοι	13
Η Μέθοδος της Μέσης Τιμής	13
Η Μέθοδος του Αντιστρόφου της Απόστασης	13
Η Μέθοδος των Υπερετήσιων Λόγων	14
Η Μέθοδος της Τροποποιημένης Στάθμισης του Μέσου Όρου	14
Σύντομη αξιολόγηση των απλών κλασικών μεθόδων	15
1.3. Μέθοδοι βασισμένες στη γραμμική παλινδρόμηση	15
Απλή γραμμική παλινδρόμηση	15
Απλή γραμμική παλινδρόμηση με τυχαίο όρο	15
Πολλαπλή γραμμική παλινδρόμηση	16
Οι μέθοδοι MOVE (Επέκταση με διατήρηση της διασποράς)	17
Γραμμική παλινδρόμηση με μηδενικό σταθερό όρο	18
1.4. Μέθοδοι βασισμένες σε στοχαστικά μοντέλα χρονοσειρών	19
Χρήση μοντέλου AR(1) ή PAR(1)	19
Πολυμεταβλητό μοντέλο	19
ΚΕΦΑΛΑΙΟ 2: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΩΝ ΜΕΘΟΔΩΝ ΣΥΜΠΛΗΡΩΣΗΣ ΣΤΗ ΒΑΣΗ	
ΔΕΔΟΜΕΝΩΝ «ΥΔΡΟΣΚΟΠΙΟ»	21
2.1. Εισαγωγή	21
2.2. Σύντομη περιγραφή της βάσης δεδομένων «Υδροσκόπιο»	21
2.3. Χρησιμοποιηθείσες γλώσσες προγραμματισμού	23
Γλώσσα C: Μαθηματικοί υπολογισμοί και διαδικασίες γενικής χρησιμότητας	23
Γλώσσα SQL: Προσπέλαση της βάσης δεδομένων	24
Γλώσσα INGRES/Windows4GL: Προσαρμοστικό χρήστη	24
Δομή της εφαρμογής	25
2.4. Το δόγμα της συμπλήρωσης	25
2.5. Περιγραφή της εφαρμογής	26
2.6. Διαχείριση και διακίνηση των δεδομένων	28

ΚΕΦΑΛΑΙΟ 3: ΕΦΑΡΜΟΓΗ ΣΕ ΗΜΕΡΗΣΙΕΣ ΚΑΙ ΜΗΝΙΑΙΕΣ ΒΡΟΧΟΠΤΩΣΕΙΣ ΜΕ ΣΥΝΤΟΜΗ

ΑΞΙΟΛΟΓΗΣΗ ΤΩΝ ΜΕΘΟΔΩΝ	32
3.1. Μεθοδολογία.....	32
3.2. Επιλογή σταθμών.....	32
3.3. Συμπληρώσεις ημερήσιων χρονοσειρών: Αποτελέσματα και παρατηρήσεις.....	33
3.4. Συμπληρώσεις μηνιαίων χρονοσειρών: Αποτελέσματα και παρατηρήσεις.....	36
3.5. Περαιτέρω δυνατότητες	36
ΠΑΡΑΡΤΗΜΑ Α: ΔΙΑΓΡΑΜΜΑΤΑ ΣΥΝΤΕΛΕΣΤΩΝ ΑΠΟΔΟΣΗΣ	37
ΠΑΡΑΡΤΗΜΑ Β: ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΦΑΡΜΟΓΗΣ.....	45
ΠΑΡΑΡΤΗΜΑ Γ: ΔΕΙΓΜΑ ΚΩΔΙΚΑ ΕΦΑΡΜΟΓΗΣ.....	56
ΒΙΒΛΙΟΓΡΑΦΙΑ	75

ΕΙΣΑΓΩΓΗ

Η ανάγκη για συμπλήρωση ελλειπουσών τιμών των χρονοσειρών

Ένας από τους στόχους της επιστήμης της υδρολογίας είναι η σύσταση μαθηματικών μοντέλων που να περιγράφουν όσο γίνεται ακριβέστερα μια υδρολογική διεργασία. Τα οφέλη από την ακριβή περιγραφή μιας διεργασίας είναι μεγάλα. Οι διαθέσιμοι πόροι μπορούν να εκτιμηθούν καλύτερα και να γίνει πιο αποτελεσματική εκμετάλλευσή τους. Η πιθανότητα ακραίων γεγονότων μπορεί να υπολογιστεί με μεγαλύτερη ακρίβεια. Έτσι, τα τεχνικά έργα εκμετάλλευσης ή προστασίας από αυτά τα γεγονότα μπορούν να σχεδιαστούν και να υλοποιηθούν με μικρότερο κόστος και μεγαλύτερη αποτελεσματικότητα.

Η κατασκευή ενός μοντέλου που να περιγράφει μια διεργασία γίνεται με την επεξεργασία χρονοσειρών, που είναι γνωστές υλοποιήσεις της διεργασίας για ένα συγκεκριμένο χρονικό διάστημα. Με την εξέταση των χαρακτηριστικών μιας χρονοσειράς μπορούν να εξαχθούν συμπεράσματα για τα χαρακτηριστικά της διεργασίας. Όπως, δηλαδή, εξετάζοντας ένα δείγμα, μπορούμε να βγάλουμε συμπεράσματα για τον πληθυσμό από τον οποίο προέρχεται, έτσι εξετάζοντας μια χρονοσειρά μπορούμε να βγάλουμε συμπεράσματα για τη στοχαστική ανέλιξη που αντιπροσωπεύει τη συγκεκριμένη υδρολογική διεργασία. Επομένως, η προσπάθεια περιγραφής της διεργασίας ανάγεται σε πρόβλημα ανάλυσης χρονοσειρών. Η ανάλυση χρονοσειρών είναι ένας μεγάλος κλάδος της υδρολογίας, και στόχος του είναι η μελέτη και επεξεργασία των χρονοσειρών, ώστε να περιγράφονται όσο γίνεται καλύτερα οι διεργασίες. Από τις πρώτες μορφές επεξεργασίας είναι η συμπλήρωση ελλειπουσών τιμών.

Οι λόγοι για τους οποίους λείπουν τιμές από τις χρονοσειρές είναι πάρα πολλοί. Τα μετρητικά όργανα συχνά παθαίνουν βλάβες και σταματούν να λειτουργούν. Μια τιμή μπορεί να καταγραφεί λανθασμένα από το προσωπικό, ή, σε περίπτωση αυτόματης καταγραφής σε υπολογιστή, να αλλοιωθεί λόγω παρεμβολών. Τέτοιες λανθασμένες τιμές άλλες φορές περνούν απαρατήρητες και άλλες επισημαίνονται με τη βοήθεια ποιοτικών ελέγχων και διαγράφονται, οπότε δημιουργούνται κενά. Εξαιρετικά φυσικά φαινόμενα, όπως κυκλώνες, μετατοπίσεις γαιών, σεισμοί, ή ανθρώπινοι παράγοντες, όπως πόλεμοι και πολιτικές διαταραχές, μπορούν να έχουν ως αποτέλεσμα τη διακοπή της λειτουργίας ενός μετρητικού οργάνου ή σταθμού. Έτσι κατά κανόνα τα δεδομένα που έχουμε στη διάθεσή μας είναι χρονοσειρές με κενά διασπαρμένα σε όλη την έκτασή τους, συνήθως ομαδοποιημένα σε μικρές ομάδες.

Η ύπαρξη ελλειπουσών τιμών δημιουργεί σοβαρά προβλήματα στην επεξεργασία μιας χρονοσειράς. Εργασίες όπως η ανάλυση αυτοσυσχέτισης και η φασματική ανάλυση, που είναι πολύ συνηθισμένες στην πορεία της κατασκευής του μαθηματικού μοντέλου της διεργασίας, δυσχεραίνονται υπερβολικά. Η άθροιση χρονοσειρών σε χρονοσειρές μεγαλύτερου βήματος, όπως η εξαγωγή μηνιαίων από ημερήσιες, καθίσταται δύσκολη. Γι' αυτό είναι επιθυμητή η

εκτίμηση των ελλειπουσών τιμών, ώστε να προκύπτουν τελικά ενιαίες και συνεχείς σειρές αριθμών, τις οποίες μπορούμε να χειριστούμε ευκολότερα. Εκτός όμως από αυτό, η συμπλήρωση των ελλειπουσών τιμών, όταν γίνεται με κατάλληλες μεθόδους και καλές προϋποθέσεις (κυρίως ύπαρξη αξιόπιστων δεδομένων σε γειτονικούς μετρητικούς σταθμούς), μπορεί να οδηγήσει σε καλύτερη εκτίμηση των στατιστικών παραμέτρων, ή σε εκτενέστερες χρονοσειρές με χαρακτηριστικά πιο κοντά στα χαρακτηριστικά της διεργασίας που εκπροσωπούν απ' ό,τι οι αρχικές χρονοσειρές. Γι' αυτό είναι συνηθισμένο όχι μόνο να συμπληρώνονται ελλείπουσες τιμές, αλλά και να γίνεται τεχνητή επέκταση μιας χρονοσειράς με προσθήκη εκτιμημένων τιμών στην αρχή ή στο τέλος της.

Όλα αυτά έχουν κάνει την εκτίμηση ελλειπουσών τιμών ένα σημαντικό τομέα της ανάλυσης χρονοσειρών. Πολλοί ερευνητές ασχολήθηκαν μ' αυτήν και επινόησαν διάφορες μεθόδους συμπλήρωσης και επέκτασης, με κυριότερη τη μέθοδο της παλινδρόμησης, που άρχισε να χρησιμοποιείται ευρέως κυρίως μετά τις εργασίες των *Fiering* [1963] και *Matalas and Jacobs* [1964]. Αργότερα η ανάπτυξη των υπολογιστών έκανε δυνατή την υλοποίηση μεθόδων με πολλές αριθμητικές πράξεις και την εκτενέστερη δοκιμή της αποτελεσματικότητάς τους, μεταξύ άλλων με μεθόδους προσομοίωσης. Η δημιουργία βάσεων υδρολογικών δεδομένων και ο αυτοματισμός πολλών επεξεργασιών διευκόλυνε μεν πολλές διαδικασίες αλλά και δημιούργησε νέα ερωτηματικά. Έτσι, ο τομέας της συμπλήρωσης των χρονοσειρών εξακολουθεί να εξελίσσεται και να βελτιώνεται, όπως και όλος ο κλάδος της ανάλυσης χρονοσειρών.

Μια βάση δεδομένων όπως οι προαναφερθείσες, το «Υδροσκόπιο», δημιουργήθηκε το 1993 στην Ελλάδα, με αποτέλεσμα να συγκεντρωθεί, κατά την υλοποίησή της, πολλή πείρα και σε θέματα πληροφορικής και σε θέματα υδρολογίας και μετεωρολογίας. Η παρούσα εργασία έγινε με κύριο στόχο την εφαρμογή των συμπερασμάτων της στο «Υδροσκόπιο». Οι διάφορες μέθοδοι συμπλήρωσης μελετήθηκαν και εφαρμόστηκαν στον υπολογιστή σε εφαρμογή που αναπτύχθηκε. Στη συνέχεια, για τον έλεγχο τόσο της εφαρμογής όσο και των μεθόδων έγιναν εικονικές συμπληρώσεις γνωστών τιμών.

Σ' αυτή την εργασία γίνεται μια μελέτη των μεθόδων συμπλήρωσης ετήσιων, μηνιαίων και ημερήσιων χρονοσειρών. Αρχικά ανασκοπούνται οι μέθοδοι που έχουν αναπτυχθεί ως σήμερα. Στη συνέχεια γίνεται μια περιγραφή του τρόπου με τον οποίο αυτές εφαρμόζονται στον υπολογιστή, των προβλημάτων που εμφανίζονται και των ερωτηματικών που ανακύπτουν. Στο τέλος παρουσιάζεται μια εφαρμογή σε ημερήσιες και μηνιαίες χρονοσειρές βροχοπτώσεων, παράλληλα με μια συνοπτική σύγκριση των μεθόδων.

Υπενθύμιση της ορολογίας

Ανελίξεις και χρονοσειρές

Η μεταβολή της καταστάσεως ενός συστήματος μέσα στο χρόνο ονομάζεται *ανέλιξη* (*process*). Αν οι παράμετροι που μεταβάλλουν το σύστημα είναι τυχαίες μεταβλητές, όπως

συμβαίνει συνήθως, η ανέλιξη ονομάζεται *στοχαστική*. Για παράδειγμα, η μεταβολή της θερμοκρασίας σε ένα σημείο του χώρου συναρτήσει του χρόνου είναι μία στοχαστική ανέλιξη.

Γενικά οι υδρολογικές ανελιξεις όπως η βροχόπτωση και η απορροφή εξελίσσονται σε συνεχή χρονική κλίμακα. Χρησιμοποιώντας μετρητικά όργανα, μπορούμε να έχουμε, συνήθως υπό μορφή γραφήματος, τη συνεχή συνάρτηση ενός μεγέθους με το χρόνο για ένα ορισμένο χρονικό διάστημα. Μια τέτοια συνάρτηση ονομάζεται *συνεχής χρονοσειρά*. Εντούτοις, οι περισσότερες υδρολογικές ανελιξεις με πρακτικό ενδιαφέρον ορίζονται σε διακριτή χρονική κλίμακα. Λαμβάνοντας τιμές της συνεχούς υδρολογικής ανελιξεως σε διακριτά χρονικά σημεία, ή ολοκληρώνοντας τη συνεχή χρονοσειρά για διαδοχικά χρονικά διαστήματα, παίρνουμε μια σειρά αριθμών, που ονομάζεται *διακριτή χρονοσειρά*. Οι διακριτές χρονοσειρές συνήθως είναι *σταθερού βήματος*, που σημαίνει πως η τιμή της μεταβλητής έχει ληφθεί ανά τακτά χρονικά διαστήματα. Σε μερικές περιπτώσεις όμως, οι τιμές μπορεί να λαμβάνονται σε ακανόνιστα διαστήματα, οπότε η χρονοσειρά λέγεται *μεταβλητού βήματος*. Σ' αυτή την εργασία μάς ενδιαφέρουν μόνο οι διακριτές χρονοσειρές σταθερού βήματος, που στο εξής θα ονομάζονται απλά *χρονοσειρές*.

Ενώ η στοχαστική ανέλιξη είναι έννοια αφηρημένη, γιατί δηλώνει κάτι άγνωστο, η χρονοσειρά είναι συγκεκριμένη, γιατί είναι ακριβώς μια υλοποίηση της ανέλιξης την οποία γνωρίζουμε έχοντας πραγματοποιήσει μετρήσεις. Έτσι κάθε χρονοσειρά εκπροσωπεί την ανέλιξη από την οποία προέρχεται. Οι στατιστικές παράμετροι της χρονοσειράς είναι εκτιμητές των αντίστοιχων παραμέτρων της ανέλιξης. Όσο μεγαλύτερη είναι η χρονοσειρά, τόσο πιο αξιόπιστα εκπροσωπεί την ανέλιξη, όπως ακριβώς συμβαίνει με τους πληθυσμούς και τα δείγματα που τους εκπροσωπούν.

Ελλιπείς χρονοσειρές και χρονοσειρές αναφοράς

Όπως αναφέρθηκε, οι χρονοσειρές περιέχουν συχνά ελλείπουσες τιμές, τις οποίες μπορούμε να συμπληρώσουμε με διάφορες μεθόδους. Μια χρονοσειρά με ελλείπουσες τιμές την οποία πρόκειται να συμπληρώσουμε την ονομάζουμε *ελλιπή χρονοσειρά*.

Οι περισσότερες μέθοδοι συμπλήρωσης συσχετίζουν τις τιμές της ελλιπούς χρονοσειράς με τις τιμές μιας ή περισσότερων χρονοσειρών που προκύπτουν συνήθως από γειτονικούς μετρητικούς σταθμούς. Αυτές οι χρονοσειρές ονομάζονται *χρονοσειρές αναφοράς*. Οι χρονοσειρές αναφοράς μπορεί επίσης να περιέχουν ελλείπουσες τιμές. Όμως, τον όρο *ελλιπής χρονοσειρά* τον εφαρμόζουμε αυστηρά στη χρονοσειρά που πρόκειται να συμπληρώσουμε και όχι σε κάθε χρονοσειρά που περιέχει ελλείπουσες τιμές.

ΚΕΦΑΛΑΙΟ 1: ΑΝΑΣΚΟΠΗΣΗ ΤΩΝ ΜΕΘΟΔΩΝ ΣΥΜΠΛΗΡΩΣΗΣ

1.1. Χρησιμοποιούμενοι συμβολισμοί

Βασικοί συμβολισμοί για όλες τις μεθόδους

Με y συμβολίζονται οι τιμές της ελλιπούς χρονοσειράς, και με $x^{(l)}$ οι τιμές της χρονοσειράς αναφοράς l . Επίσης, ορίζονται τα εξής:

- m : Ο αριθμός των χρονοσειρών αναφοράς.
 y_t : Η ζητούμενη ελλείπουσα τιμή της ελλιπούς χρονοσειράς στο χρόνο t .
 $x_t^{(l)}$: Η τιμή της χρονοσειράς αναφοράς l κατά το χρόνο t , στον οποίο η ελλιπής χρονοσειρά έχει ελλείπουσα τιμή.
 $\bar{y}, \bar{x}^{(l)}$: Οι μέσες υπερετήσιες τιμές της ελλιπούς χρονοσειράς και της χρονοσειράς αναφοράς /αντίστοιχα.
 $s(y), s(x^{(l)})$: Οι τυπικές αποκλίσεις της ελλιπούς χρονοσειράς και της χρονοσειράς αναφοράς /αντίστοιχα.
 $s(x^{(l)} x^{(j)})$: Η συνδιασπορά των χρονοσειρών αναφοράς l και j .
 r_j : Η απόσταση του σταθμού αναφοράς l από το σταθμό της ελλιπούς χρονοσειράς.

Περισσότεροι συμβολισμοί για τις μεθόδους της γραμμικής παλινδρόμησης

Θεωρούμε ότι για την ελλιπή χρονοσειρά και τις χρονοσειρές αναφοράς υπάρχει ένας αριθμός N_1 ημερών, μηνών ή ετών για τα οποία υπάρχουν έγκυρες παρατηρήσεις σε όλες τις χρονοσειρές (περίοδος 1), και $N_1 + N_2$ ημέρες, μήνες ή έτη για τα οποία υπάρχουν έγκυρες παρατηρήσεις μόνο στις χρονοσειρές αναφοράς. Οι ελλείπουσες τιμές της ελλιπούς χρονοσειράς είναι N_2 . Για την ελλιπή χρονοσειρά και μια χρονοσειρά αναφοράς l οι παρατηρημένες τιμές (N_1 για την ελλιπή, $N_1 + N_2$ για τη χρονοσειρά αναφοράς) συμβολίζονται ως εξής:

$$y_1, y_2, \dots, y_{N_1}$$

$$x_1, x_2, \dots, x_{N_1}, x_{N_1+1}, \dots, x_{N_1+N_2}$$

Οι N_1 κοινές παρατηρήσεις δεν αντιστοιχούν γενικά στις πρώτες N_1 παρατηρήσεις, ούτε χρειάζεται να είναι διαδοχικές ή κατ' αύξουσα χρονική σειρά. Εντούτοις, δεν υπάρχει απώλεια της γενικότητας από αυτό το συμβολισμό.

Θεωρούμε ακόμη:

- $\bar{y}_1, \bar{x}_1^{(l)}$: Η μέση τιμή της ελλιπούς χρονοσειράς και της χρονοσειράς αναφοράς l /αντίστοιχα για την περίοδο 1.
 $\bar{x}_2^{(j)}$: Η μέση τιμή της χρονοσειράς αναφοράς j για την περίοδο 2.
 $s_1(x^{(l)}), s_1(x^{(l)} x^{(j)})$, κλπ: Τυπική απόκλιση ή συνδιασπορά για την περίοδο 1. Εντός της παρενθέσεως είναι η χρονοσειρά της οποίας εξάγεται η τυπική

απόκλιση ή οι χρονοσειρές των οποίων εξάγεται η συνδιασπορά, ενώ ο δείκτης δείχνει για ποια περίοδο εξάγεται η συνδιασπορά.

- ρ . Ο συντελεστής γραμμικής συσχέτισης μεταξύ της ανέλιξης που εκπροσωπεί η ελλειπής χρονοσειρά και της ανέλιξης που εκπροσωπεί η χρονοσειρά αναφοράς.
- r . Εκτιμητής του ρ .

1.2. Απλές κλασικές μέθοδοι

Οι μέθοδοι που περιγράφονται σ' αυτό το τμήμα παρέχουν πρόχειρες και γρήγορες εκτιμήσεις των ελλειπουσών τιμών, και μοναδικό τους πλεονέκτημα είναι ότι μπορούν να εφαρμοστούν εύκολα χωρίς πολλές αριθμητικές πράξεις. Όταν όμως υπάρχει βάση δεδομένων και ένα πρόγραμμα που μπορεί να κάνει τις συμπληρώσεις αυτόματα, οι μέθοδοι που είναι βασισμένες στη γραμμική παλινδρόμηση είναι σχεδόν το ίδιο γρήγορες και επομένως προτιμώνται. Γι' αυτό οι μέθοδοι αυτού του τμήματος αναφέρονται για λόγους πληρότητας. Πηγή είναι κυρίως η εργασία της *Foufoula-Georgiou* [1983]. Οι αναφορές της Foufoula-Georgiou σε προηγούμενες εργασίες δεν αναφέρονται εδώ.

Η Μέθοδος της Μέσης Τιμής

Πρόκειται για την απλούστερη μέθοδο συμπλήρωσης, κατά την οποία η ελλείπουσα τιμή τίθεται ίση με το μέσο όρο των υπαρχουσών τιμών της ελλειπούς χρονοσειράς.

$$\hat{y}_t = \bar{y} \quad (1)$$

Για μηνιαίες ή ημερήσιες χρονοσειρές, η τιθέμενη τιμή πρέπει να είναι η μέση μηνιαία ή η μέση ημερήσια αντίστοιχα για το συγκεκριμένο μήνα.

Η Μέθοδος του Αντιστρόφου της Απόστασης

Σ' αυτή τη μέθοδο η ελλείπουσα τιμή εκτιμάται με βάση τις αντίστοιχες τιμές των χρονοσειρών αναφοράς, σύμφωνα με τον εξής τύπο:

$$\hat{y}_t = \sum_{i=1}^m a_i x_i^{(i)} \quad (2)$$

$$\text{όπου } a_i = \frac{\left(\frac{1}{r_i}\right)^p}{\sum_{k=1}^m \left(\frac{1}{r_k}\right)^p} \quad (3)$$

Η παραδοχή που βρίσκεται πίσω από αυτή τη μέθοδο είναι πως όσο πιο κοντά στο σταθμό της ελλειπούς χρονοσειράς βρίσκονται οι σταθμοί των χρονοσειρών αναφοράς, τόσο πιο κοντά βρίσκονται οι μετρημένες τιμές τους. Αυτό όμως ισχύει μόνο αν οι δύο σταθμοί βρίσκονται στην ίδια περιοχή, με ίδια υδρολογικά ή μετεωρολογικά χαρακτηριστικά. Αν μεταξύ των σταθμών υπάρχει ένα βουνό ή μια λίμνη, δεν πρέπει να επιχειρηθεί συμπλήρωση με αυτή τη μέθοδο.

Έχει βρεθεί ότι ο εκθέτης που είναι καταλληλότερος για βροχοπτώσεις είναι $p = 2$.

Η Μέθοδος των Υπερετήσιων Λόγων

Η μέθοδος συσχετίζει την ελλείπουσα τιμή με τις αντίστοιχες των χρονοσειρών αναφοράς με τον ίδιο τύπο (2) όπως η μέθοδος αντιστρόφου αποστάσεως:

$$\hat{y}_t = \sum_{i=1}^m a_i x_t^{(i)}$$

αλλά με άλλους συντελεστές βάρους:

$$a_i = \frac{1}{m} \times \frac{\bar{y}}{\bar{x}^{(i)}} \quad (4)$$

Αυτή η μέθοδος υπερέχει της προηγούμενης, γιατί δίνει μεγαλύτερο βάρος στις χρονοσειρές αναφοράς των οποίων οι υπερετήσιοι λόγοι είναι πλησιέστερα στον υπερετήσιο λόγο της ελλιπούς, και όχι στην πιο κοντινή. Έτσι, δεν υπάρχουν παραδοχές για τη χωρική κατανομή των υδρολογικών ή μετεωρολογικών φαινομένων, οι οποίες είναι συχνά ανακριβείς. Επίσης, αν οι μέσες υπερετήσιες τιμές των χρονοσειρών αναφοράς διαφέρουν σημαντικά από αυτές της ελλιπούς, η μέθοδος του αντιστρόφου της απόστασης δίνει εντελώς λάθος αποτελέσματα, ενώ η μέθοδος των υπερετήσιων λόγων δίνει λογικά.

Η Μέθοδος της Τροποποιημένης Στάθμισης του Μέσου Όρου

Πρόκειται για μια παραλλαγή των μεθόδων αντιστρόφου αποστάσεως και υπερετήσιων λόγων, κατά την οποία η μέση τιμή και η τυπική απόκλιση της συμπληρωμένης χρονοσειράς ταυτίζονται με τα αντίστοιχα χαρακτηριστικά της ελλιπούς. Μπορεί όμως να εφαρμοστεί μόνο όταν συμπληρώνεται ταυτόχρονα ένα μικρό πλήθος ελλειπουσών τιμών, και όχι όταν συμπληρώνονται μία-μία.

Οι μέθοδοι αντιστρόφου αποστάσεως και υπερετήσιων λόγων, όπως αναφέρεται παραπάνω, περιγράφονται αμφότερες από τον τύπο (2):

$$\hat{y}_t = \sum_{i=1}^m a_i x_t^{(i)}$$

Για την ταύτιση της μέσης τιμής και της τυπικής απόκλισης της ελλιπούς χρονοσειράς, ο τύπος μπορεί να τροποποιηθεί ως εξής:

$$\hat{y}_t = \sum_{i=1}^m \beta_i x_t^{(i)} + (\bar{y} - \sum_{i=1}^m \beta_i \bar{x}^{(i)}) \quad (5)$$

$$\text{όπου } \beta_i = a_i \frac{s(y)}{s'(y)} \text{ και } s'(y) = \sqrt{\sum_{i=1}^m \sum_{k=1}^m a_i a_k s(x^{(i)} x^{(k)})}$$

Αυτή η μέθοδος αναφέρεται μόνο για λόγους πληρότητας. Δεν τη θεωρούμε αξιόλογη, αφού η ταύτιση της μέσης τιμής και της διασποράς πριν και μετά τη συμπλήρωση σημαίνει ότι αγνοούνται οι αποκλίσεις από αυτά τα μεγέθη που μπορεί να υπήρχαν σε μια από τις ελλείπουσες τιμές.

Σύντομη αξιολόγηση των απλών κλασικών μεθόδων (Foufoula-Georgiou [1983])

Η αξιολόγηση των μεθόδων γίνεται πειραματικώς θεωρώντας μερικές υπάρχουσες τιμές μιας χρονοσειράς ως ελλείπουσες και συμπληρώνοντάς τις με τις διάφορες μεθόδους. Στη συνέχεια γίνεται σύγκριση των εκτιμημένων τιμών με τις πραγματικές. Τα κυριότερα συμπεράσματα της σύγκρισης είναι τα εξής:

- Η μέθοδος των υπερετήσιων λόγων δίνει τις πιο ακριβείς εκτιμήσεις, και η μέθοδος της μέσης τιμής τις λιγότερο ακριβείς.
- Όλες οι μέθοδοι δίνουν μεροληπτικές εκτιμήσεις της διασποράς και της σειριακής συσχέτισης, αλλά στη μέθοδο μέσης τιμής η μεροληψία είναι εντονότερη.

1.3. Μέθοδοι βασισμένες στη γραμμική παλινδρόμηση

Σε όλες τις μεθόδους που περιγράφονται παρακάτω, πρέπει να λαμβάνεται υπόψη η εποχιακότητα. Αν οι χρονοσειρές είναι ημερήσιες ή μηνιαίες, πρέπει να θεωρηθούν στην ουσία 12 διαφορετικές χρονοσειρές: μια για όλους τους Ιανουαρίους, μια για τους Φεβρουαρίους, κλπ, και να γίνουν 12 γραμμικές παλινδρομήσεις. Στη συνέχεια του κειμένου κάθε φορά αναφέρεται μία μόνο εξίσωση, και εξυπακούεται ότι πρέπει κατά την εφαρμογή να υπάρχει μέριμνα για την εποχιακότητα.

Απλή γραμμική παλινδρόμηση (Matalas and Jacobs [1964])

Η μέθοδος της απλής γραμμικής παλινδρόμησης είναι η υποπερίπτωση της πολλαπλής κατά την οποία υπάρχει μόνο μία χρονοσειρά αναφοράς ($m = 1$). Κατά την απλή γραμμική παλινδρόμηση οι τιμές της ελλιπούς χρονοσειράς συσχετίζονται γραμμικά με τις τιμές μιας άλλης χρονοσειράς, που μπορεί να έχει ίδια ή σχετικά δεδομένα. Για παράδειγμα, μια χρονοσειρά από παροχές μπορεί να συσχετιστεί με μια χρονοσειρά βροχοπτώσεων. Οι ελλείπουσες τιμές εκτιμώνται από τον τύπο

$$y_t = a + bx_t \quad (6)$$

όπου οι παράμετροι a και b εκτιμώνται από τους τύπους

$$\hat{a} = \bar{y}_1 - \hat{b}\bar{x}_1 \quad \text{και} \quad \hat{b} = \frac{s_1(xy)}{s_1^2(x)} = r \frac{s_1(y)}{s_1(x)} \quad (7)$$

όπου $r = \hat{\rho} = \frac{s_1(xy)}{s_1(x)s_1(y)}$ ο συντελεστής γραμμικής συσχέτισης μεταξύ y και x .

Απλή γραμμική παλινδρόμηση με τυχαίο όρο (Matalas and Jacobs [1964])

Όταν δεν χρησιμοποιείται τυχαίος όρος, οι εκτιμημένες τιμές έχουν μεν τη μεγαλύτερη πιθανότητα να είναι πλησιέστερα στις πραγματικές, αλλά προκαλείται υποεκτίμηση της διασποράς του συμπληρωμένου δείγματος. Ένας τρόπος καταπολέμησης αυτού του προβλήματος είναι η τροποποίηση του τύπου (6) με την πρόσθεση ενός τυχαίου όρου ε :

$$y_t = a + bx_t + \varepsilon \quad (8)$$

με $\varepsilon = z s_1(y) \alpha \sqrt{1 - \rho^2}$

όπου:

z: ασυσχέτιστη κανονική τυχαία μεταβλητή με μέση τιμή 0 και διασπορά 1.

ρ: συντελεστής συσχέτισης μεταξύ y και x εκτιμώμενος από τον τύπο

$$r = \hat{\rho} = \frac{s_1(xy)}{s_1(x)s_1(y)} = b \frac{s_1(x)}{s_1(y)}$$

α: συντελεστής εξαρτώμενος από το πλήθος των τιμών των χρονοσειρών:

$$\alpha = \sqrt{\frac{N_2(N_1 - 4)(N_1 - 1)}{(N_2 - 1)(N_1 - 3)(N_1 - 2)}}$$

Ο συντελεστής α, που υπολογίστηκε από τους *Matalas and Jacobs* [1964], απαιτείται για να είναι η αναμενόμενη τιμή της διασποράς της συμπληρωμένης ή επεκτεταμένης χρονοσειράς αμερόληπτη.

Πολλαπλή γραμμική παλινδρόμηση (Gilroy [1970], Moran [1974])

Όταν υπάρχουν περισσότερες από μία χρονοσειρές αναφοράς, μπορεί να γίνει πολλαπλή γραμμική παλινδρόμηση με βάση τον εξής τύπο:

$$y_i = a + \sum_{i=1}^m b_i x_i^{(i)} + \varepsilon \quad (9)$$

όπου οι παράμετροι a και b εκτιμώνται από τις σχέσεις

$$\hat{a} = \bar{y}_1 - \sum_{i=1}^m \hat{b}_i \bar{x}_1^{(i)} \quad (10)$$

$$\hat{b}_i = \sum_{j=1}^m d_1^{(ij)} s_1(x^{(j)} y) \quad (11)$$

και $d_1^{(ij)}$ τα στοιχεία του μητρώου

$$\begin{bmatrix} s_1^2(x^{(1)}) & s_1(x^{(1)}x^{(2)}) & \dots & s_1(x^{(1)}x^{(m)}) \\ s_1(x^{(2)}x^{(1)}) & s_1^2(x^{(2)}) & \dots & s_1(x^{(2)}x^{(m)}) \\ \vdots & \vdots & \ddots & \vdots \\ s_1(x^{(m)}x^{(1)}) & s_1(x^{(m)}x^{(2)}) & \dots & s_1^2(x^{(m)}) \end{bmatrix}^{-1}$$

Ο τυχαίος όρος, αν χρησιμοποιηθεί, δίνεται και πάλι από τον τύπο

$$\varepsilon = z s_1(y) \alpha \sqrt{1 - R^2}$$

όπου R ο συντελεστής πολλαπλής γραμμικής συσχέτισης,

$$R = \pm \sqrt{\frac{\sum_{i=1}^m b_i s_1(x^{(i)} y)}{s_1^2(y)}} \quad (\text{με το πρόσημο της τιμής εντός του απολύτου})$$

και α συντελεστής εξαρτώμενος από το πλήθος των τιμών των χρονοσειρών:

$$\alpha = \sqrt{\frac{N_2(N_1 - 2m - 2)(N_1 - 1)}{(N_2 - 1)(N_1 - m - 2)(N_1 - m - 1)}}$$

Οι μέθοδοι MOVE (Επέκταση με διατήρηση της διασποράς) (Hirsch [1982], Vogel and Stendinger [1985], Grygier and Stendinger [1989])

Όταν στόχος μας είναι η παραγωγή μιας χρονοσειράς που να διατηρεί τα στατιστικά χαρακτηριστικά της πραγματικής χρονοσειράς, η μέθοδος της γραμμικής παλινδρόμησης είναι ακατάλληλη γιατί υποεκτιμά τη διασπορά. Η πρόσθεση τυχαίου όρου λύνει αυτό το πρόβλημα, αλλά έχει δύο μειονεκτήματα: 1) Η εφαρμογή της δίνει διαφορετικές τιμές κάθε φορά. 2) Δεν διατηρεί την αυτοσυσχετιστική δομή που έχουν μερικές χρονοσειρές, όπως οι μηνιαίες σειρές παροχών (Vogel and Stendinger [1985]). Για την επέκταση εποχιακών σειρών παροχών, ο Hirsch [1982] πρότεινε δυο νέες τεχνικές, βασισμένες στην απλή γραμμική παλινδρόμηση. Τις μεθόδους αυτές τις ονόμασε MOVE.1 και MOVE.2 (MOVE = Maintenance-Of-Variance Extension). Ο Hirsch χρησιμοποιεί για την εκτίμηση των τιμών την εξίσωση (6) της απλής γραμμικής παλινδρόμησης, αλλά με διαφορετικούς εκτιμητές για τις παραμέτρους a και b .

Στη μέθοδο MOVE.1, ο Hirsch [1982] επέλεξε τους εκτιμητές των a και b έτσι ώστε αν η (6) χρησιμοποιείτο για την παραγωγή ολόκληρης της χρονοσειράς $y_i, i = 1, \dots, N_1 + N_2$, τότε η μέση τιμή \bar{y}_1 και η διασπορά $s_1^2(y)$ θα ήταν ίδιες με αυτές του αρχικού δείγματος. Οι εκτιμητές που υπολόγισε είναι

$$\hat{a} = \bar{y}_1 - \hat{b}\bar{x}_1 \quad \text{και} \quad \hat{b} = \frac{s_1(y)}{s_1(x)} \quad (12)$$

που διαφέρουν από τους εκτιμητές (7) της παλινδρόμησης στο ότι το $\frac{s_1(y)}{s_1(x)}$ δεν είναι πολλαπλασιασμένο επί r .

Στη μέθοδο MOVE.2, ο Hirsch [1982] επέλεξε τους εκτιμητές ώστε αν η (6) χρησιμοποιείτο για την παραγωγή ολόκληρης της χρονοσειράς $y_i, i = 1, \dots, N_1 + N_2$, τότε η μέση τιμή \bar{y}_1 και η διασπορά $s_1^2(y)$ της παραγόμενης χρονοσειράς θα ήταν ίδιες με αμερόληπτους εκτιμητές της μέσης τιμής και της διασποράς που είχαν υπολογίσει οι Matalas and Jacobs [1964]. Αυτοί οι εκτιμητές της μέσης τιμής και της διασποράς είναι:

$$\hat{\bar{y}} = \bar{y}_1 + \frac{N_2}{N_1 + N_2} r \frac{s_1(y)}{s_2(x)} (\bar{x}_2 - \bar{x}_1) \quad (13)$$

$$\hat{s}^2(y) = \frac{1}{N_1 + N_2 - 1} \left((N_1 - 1)s_1^2(y) + (N_2 - 1)r^2 \frac{s_1^2(y)}{s_1^2(x)} s_2^2(x) + (N_2 - 1)a^2(1 - r^2)s_1^2(y) + \frac{N_1 N_2}{N_1 + N_2} r^2 \frac{s_1^2(y)}{s_1^2(x)} (\bar{x}_2 - \bar{x}_1)^2 \right) \quad (14)$$

Οι εκτιμητές των a και b της εξίσωσης (6) είναι, σ' αυτή την περίπτωση:

$$a = \hat{\bar{y}} - \hat{b}\bar{x} \quad \text{και} \quad \hat{b} = \frac{\hat{s}(y)}{s(x)} \quad (15)$$

Ο Hirsch [1982] δοκίμασε τη μέθοδο της απλής γραμμικής παλινδρόμησης χωρίς και με τυχαίο όρο και τις μεθόδους MOVE.1 και MOVE.2, και συμπέρανε πως οι μέθοδοι MOVE, και πιο πολύ η MOVE.2, είναι πιο αποτελεσματικές από τις άλλες όταν στόχος είναι η τεχνητή παρα-

γωγή χρονοσειρών με ιδιότητες πλησιέστερα στις ιδιότητες των διεργασιών που επιχειρείται να εκπροσωπούν.

Οι *Vogel and Stendinger* [1985] παρατήρησαν ότι συνήθως δεν είναι επιθυμητή η αναπαραγωγή ολόκληρης της χρονοσειράς, αλλά μόνο των τιμών της περιόδου 2, και πρότειναν τη μέθοδο MOVE.3. Στη μέθοδο αυτή, οι παράμετροι εκτιμώνται με στόχο η χρονοσειρά που αποτελείται από τις πραγματικές τιμές για την περίοδο 1 και τις εκτιμημένες για την περίοδο 2 να έχει αναμενόμενη μέση τιμή και διασπορά (ση με τους εκτιμητές (13) και (14) των *Matalas-Jacobs*. Οι εκτιμητές των a και b της εξίσωσης (6) για τη μέθοδο MOVE.3 είναι (κατόπιν μερικών τροποποιήσεων):

$$\hat{a} = \frac{(N_1 + N_2)\hat{y} - N_1\bar{y}_1}{N_2} - \hat{b}\bar{x}_2$$

$$\hat{b} = \sqrt{\frac{(N_1 + N_2 - 1)\hat{s}^2(y) - (N_1 - 1)s_1^2(y) - \frac{N_1(N_1 + N_2)}{N_2}(\hat{y} - \bar{y}_1)^2}{(N_2 - 1)s_2^2(x)}} \quad (16)$$

όπου τα \hat{y} και $\hat{s}(y)$ δίνονται από τους τύπους (13) και (14). Οι *Vogel and Stendinger* εξέτασαν τη μέθοδο MOVE.3 με πειράματα Μόντε-Κάρλο και τη βρήκαν σχεδόν ίδια με τη MOVE.2 όσον αφορά στο μέσο τετραγωνικό σφάλμα των εκτιμητών της μέσης τιμής και της διασποράς της επεκτεταμένης χρονοσειράς.

Οι *Grygier and Stendinger* [1989] επέκτειναν τη μέθοδο MOVE για την περίπτωση πολλών χρονοσειρών αναφοράς, αλλά δεν θα επεκταθούμε περισσότερο γιατί στόχος μας είναι απλώς να δώσουμε μια ιδέα των μεθόδων MOVE.

Γραμμική παλινδρόμηση με μηδενικό σταθερό όρο

Σε πολλές περιπτώσεις επιθυμούμε να χρησιμοποιούμε τους τύπους της γραμμικής παλινδρόμησης (6), (8) ή (9) με σταθερό όρο $a = 0$. Το φυσικό νόημα αυτής της περίπτωσης είναι πως αν σε όλες τις χρονοσειρές αναφοράς η τιμή είναι μηδέν, τότε και στην ελλιπή χρονοσειρά η τιμή πρέπει να είναι μηδέν. Αυτή η παραδοχή είναι, για παράδειγμα, εύλογη στην περίπτωση ημερήσιων χρονοσειρών βροχοπτώσεων, αλλά φυσικά δεν ισχύει στην περίπτωση θερμοκρασιών.

Στη γραμμική παλινδρόμηση με μηδενικό σταθερό όρο, η ελλείπουσα τιμή δίνεται από τον τύπο

$$\hat{y}_t = \sum_{i=1}^m b_i x_t^{(i)} \quad (17)$$

Είναι εύκολο να δείχτεί, με χρήση ελαχίστων τετραγώνων, ότι ο συντελεστής b δίνεται σ' αυτή την περίπτωση από τον (ίδιο τύπο (11) όπως και στην περίπτωση του μη μηδενικού όρου:

$$\hat{b}_i = \sum_{j=1}^m d_1^{(ij)} s_1(x^{(j)} y)$$

με τη διαφορά ότι το $s_1(x^{(j)} y)$, και στην εξίσωση και στο μητρώο που δίνει τα $d_1^{(ij)}$, δεν είναι η συνδιασπορά αλλά το άθροισμα των γινομένων των τιμών της κοινής περιόδου μετρήσεων.

1.4. Μέθοδοι βασισμένες σε στοχαστικά μοντέλα χρονοσειρών (Salas [1993])

Χρήση μοντέλου AR(1) ή PAR(1)

Έστω χρονοσειρά που περιγράφεται από ένα μοντέλο AR(1):

$$y_t = \bar{y} + \varphi(y_{t-1} - \bar{y}) + \varepsilon_t \quad (18)$$

του οποίου οι παράμετροι μπορούν να εκτιμηθούν από τα διαθέσιμα δεδομένα. Αν μια τιμή y_t λείπει αλλά είναι γνωστή η y_{t-1} , τότε η y_t μπορεί να εκτιμηθεί από την παραπάνω εξίσωση. Η χρήση μηδενικού τυχαίου όρου δίνει την «πιθανότερη»¹ τιμή. Αν η χρονοσειρά είναι μηνιαία ή ημερήσια, πρέπει το μοντέλο να είναι εποχιακό, με 12 παραμέτρους φ και 12 μέσες τιμές \bar{y} , μια για κάθε μήνα, και να χρησιμοποιείται για κάθε ελλείπουσα τιμή το κατάλληλο ζεύγος παραμέτρων. Σ' αυτή την περίπτωση, το μοντέλο ονομάζεται PAR(1).

Το μοντέλο AR(1) ή PAR(1) στην ουσία προκύπτει από απλή γραμμική παλινδρόμηση της ελλιπούς χρονοσειράς με τον εαυτό της υστερημένο κατά 1. Αν θεωρήσουμε την υστερημένη κατά 1 ελλιπή χρονοσειρά απλά ως μια χρονοσειρά αναφοράς, τότε η χρήση της μεθόδου της απλής παλινδρόμησης (εξίσωση (6)) θα δώσει εκτιμητές

$$\hat{a} = \bar{y} - \hat{\varphi}\bar{y} \quad \text{και} \quad \hat{b} = \hat{\varphi}$$

που αν αντικατασταθούν στην εξίσωση (6) θα δώσουν την (18).

Ο Salas [1993] αναφέρει ότι το παραπάνω μοντέλο δεν πρέπει να χρησιμοποιείται για τη συμπλήρωση διαδοχικών ελλειπουσών τιμών. Προφανώς αυτό οφείλεται στο ότι αν συμπληρωθούν διαδοχικές τιμές, τότε όλες εκτός από την πρώτη θα είναι συσχετισμένες με τις προηγούμενες τους, που προέρχονται από συμπλήρωση. Αυτό πολλαπλασιάζει τα σφάλματα.

Η χρήση του μοντέλου AR(1) είναι κατάλληλη όταν δεν υπάρχουν γειτονικές χρονοσειρές με τις οποίες να μπορεί να συσχετιστεί η ελλιπής.

Πολυμεταβλητό μοντέλο

Αν κάνουμε συσχέτιση με χρονοσειρές αναφοράς, και παράλληλα χρησιμοποιηθεί και αυτο-συσχέτιση, τότε έχουμε ένα πολυμεταβλητό μοντέλο. Αν y_t είναι οι τιμές της ελλιπούς χρονοσειράς και $x_t^{(1)}, x_t^{(2)}, \dots$ οι τιμές των χρονοσειρών αναφοράς, τότε το μοντέλο αυτό στη γενική μορφή του είναι

$$y_t = a + \sum_{j=1}^p \beta_j y_{t-j} + \sum_{j=0}^p \beta_j^{(1)} x_{t-j}^{(1)} + \sum_{j=0}^{p_2} \beta_j^{(2)} x_{t-j}^{(2)} + \dots + \varepsilon_t \quad (19)$$

Στην πράξη, επειδή η συσχέτιση της ελλιπούς χρονοσειράς με τις χρονοσειρές αναφοράς με υστέρηση μεγαλύτερη από 0 είναι πολύ μικρή για χρονικές κλίμακες μεγαλύτερες ή ίσες της ημέρας, όπως και η αυτοσυσχέτιση με υστέρηση μεγαλύτερη από 1, χρησιμοποιείται το παραπάνω μοντέλο με $p=1$ και $\beta_1 = \beta_2 = \dots = 0$, δηλαδή

$$y_t = a + \beta_1 y_{t-1} + \beta_0^{(1)} x_t^{(1)} + \beta_0^{(2)} x_t^{(2)} + \dots + \varepsilon_t \quad (20)$$

¹ Ο όρος «πιθανότερη» είναι ίσως κατάλληλος για κατανόηση του εννοούμενου, αλλά από μαθηματικής πλευράς είναι λανθασμένος. Μαθηματικώς εννοείται η εκτίμηση με το ελάχιστο τετραγωνικό σφάλμα.

Έστω m ο αριθμός των χρονοσειρών αναφοράς. Αλλάζοντας τους συμβολισμούς και θέτοντας $b_1 = \beta_0^{(1)}, b_2 = \beta_0^{(2)}, \dots, b_m = \beta_0^{(m)}$ και συμβολίζοντας τις τιμές της υστερημένης κατά 1 ελλιπούς χρονοσειράς με $x_t^{(m+1)}$ και το συντελεστή της με $b_{m+1} = \beta_1$, η παραπάνω εξίσωση γράφεται

$$y_t = a + \sum_{i=1}^{m+1} b_i x_t^{(i)} + \varepsilon_t \quad (21)$$

είναι δηλαδή (δια με την εξίσωση (9) της πολλαπλής γραμμικής παλινδρόμησης, όπου η υστερημένη κατά 1 ελλιπής χρονοσειρά έχει θεωρηθεί απλά ως χρονοσειρά αναφοράς.

ΚΕΦΑΛΑΙΟ 2: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΩΝ ΜΕΘΟΔΩΝ ΣΥΜΠΛΗΡΩΣΗΣ ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ «ΥΔΡΟΣΚΟΠΙΟ»

2.1. Εισαγωγή

Η υλοποίηση των μεθόδων συμπλήρωσης σε μια αυτοματοποιημένη εφαρμογή βάσης δεδομένων έχει σοβαρά πλεονεκτήματα. Ο χρήστης μπορεί να εφαρμόσει εύκολα οποιαδήποτε από τις προγραμματισμένες μεθόδους σε οποιαδήποτε δεδομένα, χρησιμοποιώντας ως χρονοσειρές αναφοράς οποιοσδήποτε είναι αποθηκευμένες στη βάση. Αλλιώς, θα έπρεπε ο ενδιαφερόμενος πρώτα να συγκεντρώσει τα δεδομένα σε ένα τετράδιο ή σε μαγνητικό μέσο, στη συνέχεια να τα πληκτρολογήσει ή να τα φορτώσει σε ένα πρόγραμμα φύλλου εργασίας, όπως το Lotus 1-2-3 ή το Microsoft Excel, και τέλος να δώσει οδηγίες στο πρόγραμμα για το πώς να εφαρμόσει τη συμπλήρωση. Αντίθετα, στο «Υδροσκόπιο» ο χρήστης θα έχει εύκολα πρόσβαση σε όλα τα δεδομένα της Ελλάδας μέσα σε λίγα δευτερόλεπτα, ενώ η εφαρμογή συμπλήρωσης τα φορτώνει αυτόματα και γρήγορα και υλοποιεί την επιθυμητή μέθοδο σε ελάχιστο χρόνο.

Υπάρχουν όμως και προβλήματα, που αφορούν κυρίως στην υλοποίηση της εφαρμογής. Όταν ο χρήστης εφαρμόζει τις μεθόδους με το χέρι, έχει μπροστά του και βλέπει όλα τα δεδομένα όλων των χρονοσειρών. Αν υπάρχει πρόβλημα σε κάποιο δεδομένο, μπορεί αμέσως να αποφασίσει τι θα κάνει. Στην περίπτωση της αυτόματης εφαρμογής, ωστόσο, όχι μόνο ο χρήστης δεν βλέπει τα δεδομένα, αλλά ο προγραμματισμός γίνεται πριν αυτά να είναι γνωστά. Πρέπει λοιπόν να καλυφθούν όλες οι περιπτώσεις. Ας πάρουμε για παράδειγμα τη μέθοδο της γραμμικής παλινδρόμησης. Αν μερικές τιμές μιας χρονοσειράς αναφοράς προέρχονται από συμπλήρωση, μπορούν να περιληφθούν στην κοινή περίοδο μετρήσεων ή πρέπει να εξαιρεθούν; Τι συμβαίνει αν προέρχονται από ανόρθωση μη ομογενών δεδομένων ή από διόρθωση δεδομένων που δεν πέρασαν τους βασικούς ελέγχους;

Ενώ αυτά τα ερωτήματα είναι απλά και λίγα, η εφαρμογή πρέπει να είναι ικανή να τα αντιμετωπίσει αποτελεσματικά και όλες οι πιθανές περιπτώσεις πρέπει να προβλεφθούν από πριν. Αυτό, καθώς και η μεγάλη πολυπλοκότητα του σχήματος της βάσης δεδομένων, καθιστούν την εφαρμογή ιδιαίτερα πολύπλοκη. Σ' αυτό το κεφάλαιο επιχειρείται μια περιγραφή του τρόπου με τον οποίο λειτουργεί εσωτερικά η εφαρμογή και το πώς αντιμετωπίζει τα προβλήματα.

2.2. Σύντομη περιγραφή της βάσης δεδομένων «Υδροσκόπιο» (*Papakostas et al [1994], Παπακώστας [1993]*)

Το «Υδροσκόπιο» είναι ένα ερευνητικό πρόγραμμα, που συνίσταται στη δημιουργία μιας πανελληνίας βάσης υδρολογικών και μετεωρολογικών δεδομένων σε ένα δίκτυο υπολογιστών. Για την υλοποίησή του συνεργάζονται 14 ιδρύματα και υπηρεσίες, 5 από τα οποία έχουν στη

διάθεσή τους μετρητικούς σταθμούς και δεδομένα. Ένα δίκτυο υπολογιστών συνδέει μεταξύ τους αυτά τα 14 ιδρύματα, που ονομάζονται *κόμβοι* του δικτύου. Σε 5 από τους κόμβους, που αντιστοιχούν στις υπηρεσίες που έχουν δεδομένα στη διάθεσή τους, θα υπάρχουν αποθηκευμένα τα δεδομένα, τα οποία θα είναι προσπελάσιμα από οποιονδήποτε από τους 14 κόμβους. Έτσι, τα δεδομένα δεν βρίσκονται όλα σε ένα σημείο, αλλά είναι κατανεμημένα στους 5 κόμβους. Μια βάση δεδομένων που λειτουργεί με αυτό τον τρόπο λέγεται *κατανεμημένη*. Ο χρήστης κάθε φορά εργάζεται σε έναν υπολογιστή ενός κόμβου, που είναι γι' αυτόν ο *τοπικός κόμβος*. Οι υπόλοιποι κόμβοι ονομάζονται *απομακρυσμένοι*.

Η χρησιμοποιούμενη βάση δεδομένων, η INGRES, είναι *σχεσιακή*. Αυτό σημαίνει ότι αποτελείται από ένα σύνολο διδιάστατων πινάκων. Σε καθένα από αυτούς τους πίνακες βρίσκεται ένα ορισμένο είδος πληροφορίας. Για παράδειγμα, ο πίνακας σταθμών του Υδροσκοπίου περιέχει τόσες γραμμές όσοι είναι οι καταχωρημένοι σταθμοί στη βάση, και έναν αριθμό στηλών που περιέχουν διάφορες πληροφορίες για τους σταθμούς, όπως είναι ο κωδικός, το όνομα, η λεκάνη απορροής, οι γεωγραφικές συντεταγμένες, η υπηρεσία στην οποία ανήκει, η ημερομηνία έναρξης και παύσης της λειτουργίας του, κλπ. Ο όρος «σχεσιακή» (relational) υπονοεί την ύπαρξη συσχετισμών μεταξύ των πινάκων της βάσης. Για παράδειγμα, για να βρούμε τη βροχόπτωση σε μια ορισμένη ημερομηνία σε έναν ορισμένο σταθμό, πρέπει να ανατρέξουμε στον πίνακα των σταθμών, και με βάση το όνομά του να βρούμε τον κωδικό του. Στη συνέχεια, πρέπει να ανατρέξουμε στον πίνακα των οργάνων και να βρούμε τον κωδικό του οργάνου που βρίσκεται στο συγκεκριμένο σταθμό και του οποίου ο τύπος είναι «ΒΡΟΧΟΜΕΤΡΟ». Τέλος, με βάση τον κωδικό του οργάνου και την ημερομηνία ανατρέχουμε στον πίνακα των δεδομένων και βρίσκουμε τη βροχόπτωση.

Τα δεδομένα που αποθηκεύονται στη βάση καταχωρούνται σε τρία επίπεδα. Στο πρώτο επίπεδο τοποθετούνται δεδομένα πριν γίνει οποιοσδήποτε έλεγχος. Τα δεδομένα εισάγονται στο επίπεδο 1 είτε με απευθείας πληκτρολόγηση, είτε, σε περίπτωση που είναι διαθέσιμο ένα γράφημα, με ψηφιοποίηση, είτε, αν ο μετρητικός σταθμός είναι αυτόματης καταγραφής, από ένα μαγνητικό μέσο. Τα δεδομένα του επιπέδου 1 υφίστανται μερικούς βασικούς ελέγχους, και συγκεκριμένα έλεγχο ακραίων τιμών, εσωτερικής συνέπειας, χρονικής συνέπειας και συνέπειας στο χώρο. Με τη βοήθεια αυτών των ελέγχων εντοπίζονται κυρίως χονδροειδή σφάλματα. Τα δεδομένα που περνούν αυτούς τους ελέγχους μεταφέρονται στο δεύτερο επίπεδο καταχώρησης. Όταν γράφουμε «μεταφέρονται» δεν εννοούμε ότι απομακρύνονται από το επίπεδο 1. Εξακολουθούν να βρίσκονται και στο επίπεδο 1. Τα δεδομένα που δεν περνούν τους ελέγχους είτε διορθώνονται και μεταφέρονται στο επίπεδο 2 διορθωμένα, είτε αγνοούνται, οπότε δημιουργείται ελλείπουσα τιμή. Στην πρώτη περίπτωση, στο επίπεδο 1 εξακολουθεί να βρίσκεται η αρχική, λανθασμένη τιμή, και στο 2 η διορθωμένη, ενώ στη δεύτερη περίπτωση υπάρχει στο επίπεδο 2 μια ελλείπουσα τιμή.

Το επίπεδο 2 περιέχει λοιπόν ελεγμένα και διορθωμένα δεδομένα, που στη συνέχεια υπόκεινται έλεγχο ομογένειας. Αυτά που περνούν τον έλεγχο μεταφέρονται στο επίπεδο 3, ενώ

αυτά που δεν τον περνούν ανορθώνονται και μεταφέρονται διορθωμένα στο επίπεδο 3. Υπάρχει ακόμα πιθανότητα να μην περάσουν τον έλεγχο ομογένειας και να μην ανορθωθούν, αλλά να θεωρηθούν εντελώς λανθασμένα και να αγνοηθούν.

Τέλος, οι συμπληρωμένες τιμές τοποθετούνται στο επίπεδο 3. Σ' αυτή την περίπτωση, είτε δεν υπάρχουν αντίστοιχες στα επίπεδα 1 και 2, είτε υπάρχουν και έχουν αγνοηθεί ως λανθασμένες.

Ο λόγος για τον οποίο υπάρχει αυτός ο διαχωρισμός επιπέδων καταχώρησης είναι η επιθυμία μας να κρατούμε στη βάση όχι μόνο τα διορθωμένα δεδομένα, αλλά, για λόγους πληρότητας, και τα δεδομένα από τα οποία προήλθαν. Συνοψίζοντας, στο επίπεδο 1 υπάρχουν πρωτογενή μη ελεγμένα δεδομένα, στο επίπεδο 2 υπάρχουν ελεγμένα και διορθωμένα, και στο επίπεδο 3 βρίσκονται ομογενοποιημένα δεδομένα καθώς και δεδομένα που προέρχονται από συμπλήρωση ελλειπουσών τιμών. Για μια μελέτη χρησιμοποιούνται τα δεδομένα επιπέδου 3.

2.3. Χρησιμοποιηθείσες γλώσσες προγραμματισμού

Το δίκτυο υπολογιστών του «Υδροσκοπίου» χρησιμοποιεί το λειτουργικό σύστημα UNIX και τη βάση δεδομένων INGRES. Για την υλοποίηση του «Υδροσκοπίου» έχουν χρησιμοποιηθεί 4 γλώσσες προγραμματισμού: C, SQL, INGRES/Windows4GL και PROLOG. Στην εφαρμογή συμπλήρωσης έχουν χρησιμοποιηθεί οι πρώτες τρεις. Αυτές περιγράφονται με συντομία παρακάτω, και παρατίθενται οι λόγοι για τους οποίους έχει χρησιμοποιηθεί η κάθε μια.

Γλώσσα C: Μαθηματικοί υπολογισμοί και διαδικασίες γενικής χρησιμότητας (Kernighan and Ritchie [1988])

Η γλώσσα C είναι πλέον η (ίσως πιο διαδεδομένη) γλώσσα προγραμματισμού, και είναι ό,τι καλύτερο υπάρχει στις γλώσσες γενικής χρησιμότητας. Το κύριο χαρακτηριστικό της είναι ότι είναι «χαμηλού επιπέδου». Αυτός ο όρος σημαίνει πως η C ασχολείται με το ίδιο είδος αντικειμένων με το οποίο ασχολείται ο επεξεργαστής, δηλαδή χαρακτήρες, αριθμούς και διευθύνσεις μνήμης. Αυτά μπορούν να συνδυαστούν και να διακινηθούν με τους συνηθισμένους αριθμητικούς και λογικούς τελεστές που υλοποιούνται από τον επεξεργαστή. Επίσης, μια βασική φιλοσοφία της C είναι ότι ο προγραμματιστής ξέρει τι κάνει. Παράδοξες μετατροπές τύπων, που απαγορεύονται στις περισσότερες γλώσσες, στη C είναι αποδεκτές και κάνουν τη γλώσσα ιδιαίτερα ευέλικτη.

Το γεγονός ότι η C είναι χαμηλού επιπέδου βρίσκεται πίσω από τα υπόλοιπα εμφανιζόμενα πλεονεκτήματά της, με πρώτο τη μεγάλη ταχύτητα εκτέλεσης του προγράμματος. Επειδή τα αντικείμενα και οι εντολές της γλώσσας βρίσκονται πολύ κοντά σ' αυτές καθεαυτές τις δυνατότητες του υπολογιστή, η μετάφραση γίνεται ευθέως χωρίς πολλές απώλειες, με αποτέλεσμα το πρόγραμμα να τρέχει με ταχύτητα σχεδόν ίση με αυτήν ενός προγράμματος γραμμένου σε συμβολική γλώσσα (assembly language). Το άλλο πλεονέκτημα της C είναι ότι η γλώσσα κάνει ακριβώς ό,τι ζητήσει ο προγραμματιστής, ενώ οι γλώσσες ψηλότερου επιπέδου, όπως η

INGRES/Windows4GL, στην οποία κάθε εντολή μπορεί να μεταφράζεται σε ένα μεγάλο σύνολο λειτουργιών, πολλές φορές συμπεριφέρονται απρόβλεπτα και δημιουργούν δυσκολίες.

Τέλος, η γλώσσα C είναι μεταφερτή. Αυτό σημαίνει ότι προγράμματα γραμμένα για έναν τύπο υπολογιστή μπορούν με μικρές ή χωρίς καθόλου μετατροπές να τρέξουν σε άλλον.

Λόγω αυτών των πλεονεκτημάτων της, και κυρίως της μεγάλης ταχύτητας, η C έχει χρησιμοποιηθεί για τις διαδικασίες που κάνουν μαθηματικούς υπολογισμούς, καθώς και για διαδικασίες γενικής χρησιμότητας, όπως απόσπαση της ημέρας, του μήνα ή του έτους από μια συμβολοσειρά που περιέχει μια ημερομηνία.

Γλώσσα SQL: Προσπέλαση της βάσης δεδομένων (Ingres [1991])

Η SQL (*Structured Query Language*) είναι μια γλώσσα που χρησιμοποιείται με τις σχεσιακές βάσεις δεδομένων, και περιέχει εντολές με τις οποίες τα δεδομένα οργανώνονται και διακινούνται.

Υπάρχουν πολλές κατηγορίες SQL. Μεταξύ αυτών είναι η *διαλογική (interactive) SQL* και η *ενσωματωμένη (embedded) SQL*. Η γλώσσα χρησιμοποιείται στη διαλογική μορφή της όταν θέλουμε οι εντολές της γλώσσας να εκτελεστούν κατευθείαν, μόλις τις δώσουμε. Η μορφή που μας ενδιαφέρει εδώ είναι η ενσωματωμένη. Σ' αυτή τη μορφή, οι εντολές της γλώσσας SQL ενσωματώνονται μέσα σε μια άλλη γλώσσα προγραμματισμού, όπως η C, την οποία ονομάζουμε *ξενίστρια γλώσσα*. Οι δυο γλώσσες συνυπάρχουν στον ίδιο κώδικα. Με τη γλώσσα SQL διαβάζονται στοιχεία από τη βάση δεδομένων σε μεταβλητές της ξενίστριας, ή αντιστρόφως, εγγράφονται στοιχεία από τις μεταβλητές της ξενίστριας γλώσσας στη βάση δεδομένων.

Όταν χρησιμοποιείται η ενσωματωμένη SQL, το πρόγραμμα γράφεται και δομείται ουσιαστικά στην ξενίστρια γλώσσα. Όποτε χρειάζεται προσπέλαση της βάσης, γράφεται μια εντολή SQL πριν από την οποία τοποθετείται το πρόθεμα *exec sql*. Για τη μετάφραση του προγράμματος γίνεται πρώτα επεξεργασία του πηγαιού κώδικα από τον *προεπεξεργαστή SQL*, που μεταφράζει όλες τις εντολές των οποίων προηγείται το πρόθεμα *exec sql* σε εντολές της ξενίστριας. Στη συνέχεια ο κώδικας που προκύπτει δίνεται στο μεταφραστή της ξενίστριας.

Γλώσσα INGRES/Windows4GL: Προσαρμοστικό χρήστη (Ingres [1992])

Η INGRES/Windows4GL είναι ένα σύστημα ανάπτυξης εφαρμογών που τρέχουν σε παραθυρικό περιβάλλον. Χάρη στα εργαλεία που υπάρχουν για το σχεδιασμό των παραθύρων της εφαρμογής και στις δυνατότητες της γλώσσας προγραμματισμού, που σημαίνουν εύκολη εκμετάλλευση του παραθυρικού περιβάλλοντος, είναι η καταλληλότερη για το σχεδιασμό του προσαρμοστικού χρήστη (*user interface*).

Τα παραθυρικά περιβάλλοντα, ενώ θεωρούνται φιλικά και απλά για τους χρήστες, είναι ακριβώς το αντίθετο στον προγραμματισμό. Ο τρόπος με τον οποίο τα παράθυρα μιας εφαρμογής επικοινωνούν το ένα με το άλλο και ο μεγάλος αριθμός των πιθανών πορειών που μπορεί να ακολουθήσει η ροή του προγράμματος σε μια παραθυρική εφαρμογή καθιστούν τον προγραμματισμό εξαιρετικά πολύπλοκο. Αποτέλεσμα της ιδιομορφίας των παραθυρικών περιβαλλόντων

ήταν να αναπτυχθεί, παράλληλα με αυτά, μια ειδική κατηγορία προγραμματισμού, ο *αντικειμενοστρεφής*. Σήμερα υπάρχουν πολλά συστήματα ανάπτυξης εφαρμογών (Microsoft Visual Basic, Microsoft Visual C++, INGRES/Windows4GL) που απλοποιούν τον προγραμματισμό σε παραθυρικό περιβάλλον. Όλα αυτά τα συστήματα έχουν στοιχεία από αντικειμενοστρεφή προγραμματισμό.

Η γλώσσα INGRES/Windows4GL, αν και δεν είναι αυστηρά αντικειμενοστρεφής, όπως η C++, περιέχει εντούτοις αρκετά στοιχεία αντικειμενοστρεφούς προγραμματισμού που καθιστούν την ανάπτυξη παραθυρικών εφαρμογών εύκολη και γρήγορη. Επίσης, επειδή είναι σχεδιασμένη για ανάπτυξη εφαρμογών σχετικών με τη βάση δεδομένων, εμπεριέχει όλη σχεδόν τη γλώσσα SQL, με μερικές μόνο μικροαλλαγές στη σύνταξη των εντολών της.

Δομή της εφαρμογής

Ο κορμός της εφαρμογής είναι γραμμένος σε γλώσσα INGRES/Windows4GL (περίπου 4500 γραμμές κώδικα). Ο κώδικας αυτός καλεί συχνά διαδικασίες γραμμένες σε γλώσσα C (περίπου 2000 γραμμές κώδικα) οι οποίες φορτώνουν δεδομένα από τη βάση, εκτελούν μαθηματικούς υπολογισμούς κ.ά. Η γλώσσα SQL χρησιμοποιείται σποραδικά όποτε χρειάζεται να γίνει προσπέλαση της βάσης δεδομένων, είτε ως ενσωματωμένη στη C, είτε ως τμήμα της INGRES/Windows4GL.

Τα δεδομένα της εφαρμογής κρατούνται άλλα σε μεταβλητές της INGRES/Windows4GL και άλλα σε πίνακες της C ορισμένους δυναμικά. Οι πίνακες αυτοί μπορούν να προσπελαστούν μόνο από τη C, αλλά οι δείκτες σ' αυτούς φυλάσσονται σε μεταβλητές της INGRES/Windows4GL. Επομένως, οι διαδικασίες της INGRES/Windows4GL είναι αυτές που διαχειρίζονται όλα τα δεδομένα και που καθοδηγούν, όποτε χρειάζεται, τις διαδικασίες της C ως προς το τι να κάνουν. Επίσης, όλα τα δεδομένα που σχετίζονται με μια συμπλήρωση είναι ομαδοποιημένα σε ένα αντικείμενο, που μπορεί εύκολα να διακινείται μεταξύ διαδικασιών της INGRES/Windows4GL, κάνοντας την εφαρμογή απλούστερη.

2.4. Το δόγμα της συμπλήρωσης

Όπως έχει αναφερθεί στο κεφάλαιο 1, όταν χρησιμοποιείται η μέθοδος της γραμμικής παλινδρόμησης, η εκτιμημένη τιμή έχει μεγαλύτερη πιθανότητα να βρίσκεται κοντά στην πραγματική, δηλαδή το μέσο τετραγωνικό σφάλμα της εκτίμησης θα είναι ελάχιστο. Αυτό όμως δεν σημαίνει ότι η μέθοδος συμπλήρωσης που δίνει τα ελάχιστα τετράγωνα είναι η βέλτιστη: το πρόβλημα σ' αυτή την περίπτωση είναι ότι υποεκτιμάται η διασπορά της χρονοσειράς. Το τίμημα που έχουμε να πληρώσουμε για την «πιθανότερη» εκτίμηση είναι λοιπόν η αλλοίωση των στατιστικών χαρακτηριστικών της χρονοσειράς.

Για να διατηρήσουμε τα στατιστικά χαρακτηριστικά μπορούμε να χρησιμοποιήσουμε μια παραλλαγή της μεθόδου της γραμμικής παλινδρόμησης: είτε, δηλαδή, να προσθέσουμε τυχαίο όρο, είτε να χρησιμοποιήσουμε μια από τις μεθόδους MOVE. Τότε όμως οι εκτιμώμενες τιμές δεν είναι οι «πιθανότερες».

Αυτό το δίλημμα φανερώνει πως για να προχωρήσουμε σε μια συμπλήρωση πρέπει προηγουμένως να γνωρίζουμε σε τι θα χρησιμοποιηθούν οι συμπληρωμένες τιμές. Όμως, στο «Υδροσκόπιο», η συμπλήρωση των ελλειπουσών τιμών είναι μέρος της τακτικής επεξεργασίας των εισαγόμενων δεδομένων, και στόχος της είναι η παραγωγή συνεχών χρονοσειρών για γενική χρήση. Επομένως δεν είναι γνωστό από πριν σε τι θα χρησιμοποιηθούν, ενώ μπορεί να χρησιμοποιηθούν για διαφορετικούς σκοπούς με διαφορετικές ανάγκες.

Η τακτική που ακολουθείται στο «Υδροσκόπιο» πάνω σ' αυτό το ζήτημα είναι η εξής: Γενικά χρησιμοποιείται για τη συμπλήρωση η μέθοδος που δίνει την «πιθανότερη» τιμή, και η εκτιμώμενη τιμή αποθηκεύεται στη βάση στο επίπεδο 3. Αυτό όμως γίνεται για λίγες μόνο τιμές, που αποτελούν μόνο ένα μικρό ποσοστό των τιμών της χρονοσειράς, ώστε να μην αλλοιώνονται σημαντικά τα χαρακτηριστικά της. Αν οι ελλείπουσες τιμές είναι πολλές, αν λείπει π.χ. ένας μήνας από μια ημερήσια χρονοσειρά, τότε δεν γίνεται συμπλήρωση. Τα μικρά κενά λοιπόν συμπληρώνονται ώστε οι χρονοσειρές να αποτελούνται από όσο το δυνατόν μεγαλύτερα ενιαία τμήματα, αλλά τα μεγάλα κενά αφήνονται κενά. Σε κάθε περίπτωση ο ερευνητής που εκπονεί μια μελέτη μπορεί, αν χρειάζεται, να συμπληρώσει τα μεγάλα κενά με τη μέθοδο που αρμόζει στην περίπτωση του. Οι συμπληρώσεις που θα κάνει αποθηκεύονται σε ένα αρχείο για την προσωπική του χρήση. Επίσης, στη βάση δεδομένων είναι αποθηκευμένο και όλο το ιστορικό των συμπληρώσεων. Έτσι, μπορεί να δει κανείς τη μέθοδο και τις παραμέτρους που έχουν χρησιμοποιηθεί, και να αποφασίσει αν θα δεχτεί τις συμπληρωμένες τιμές όπως είναι ή αν θα τις αγνοήσει.

2.5. Περιγραφή της εφαρμογής

Σ' αυτό το τμήμα γίνεται μια περιγραφή της λειτουργίας της εφαρμογής, όσον αφορά στο υδρολογικό μέρος της. Δεν επιχειρείται δηλαδή παρουσίαση του περιβάλλοντος, ούτε των προβλημάτων που αφορούν στη διαχείριση της βάσης δεδομένων ή στον προγραμματισμό.

Η εφαρμογή παρέχει προς το παρόν δυνατότητα χρησιμοποίησης των μεθόδων της μέσης τιμής, αντιστρόφου της απόστασης, υπερετήσιων λόγων, γραμμικής παλινδρόμησης, και του πολυμεταβλητού στοχαστικού μοντέλου (21) της σελίδας 20. Η γραμμική παλινδρόμηση είναι απλή ή πολλαπλή, με ή χωρίς τυχαίο όρο, με δυνατότητα μηδενικού σταθερού όρου. Οι ίδιες επιλογές ισχύουν και για το στοχαστικό μοντέλο, που εμπεριέχει το μοντέλο AR(1) (περίπτωση $m = 0$).

Προκειμένου να γίνει μια συμπλήρωση, ο χρήστης επιλέγει την ελλιπή χρονοσειρά (σταθμός και όργανο), τη μέθοδο συμπλήρωσης, τις χρονοσειρές αναφοράς (εκτός από τις μεθόδους μέσης τιμής και AR(1) ή PAR(1)) και τις χρονικές θέσεις για τις οποίες θα εκτιμηθούν οι ελλείπουσες τιμές. Αυτές θα τις ονομάζουμε στο εξής *θέσεις προς συμπλήρωση*.

Ο χρήστης έχει απόλυτη ελευθερία ως προς τις θέσεις προς συμπλήρωση που θα επιλέξει. Μπορούν να είναι διαδοχικές ή συνεχείς ή διασπαρμένες μέσα στη χρονοσειρά με οποιοδήποτε τρόπο. Αυτό κάνει την εφαρμογή ιδιαίτερα ευέλικτη. Αφού συμπληρωθούν μερικές θέσεις, ο

χρήστης μπορεί να ζητήσει άλλη συμπλήρωση, για άλλες θέσεις, με άλλη μέθοδο ή με άλλες χρονοσειρές αναφοράς. Για παράδειγμα, αν επιχειρείται η συμπλήρωση των τιμών μιας χρονοσειράς από το 1960 ως το 1990, μπορεί να χρησιμοποιηθεί η μέθοδος της απλής γραμμικής παλινδρόμησης με κάποια χρονοσειρά αναφοράς A για το διάστημα 1960-1980, και απλή γραμμική παλινδρόμηση με μια άλλη χρονοσειρά αναφοράς B για το διάστημα 1981-1990. Αυτό είναι εύλογο να γίνει αν έχει λειτουργήσει από το 1981 ένας νέος μετρητικός σταθμός που παρέχει μια χρονοσειρά με καλύτερη συσχέτιση με την ελλειπή. Επίσης, αν συμβεί για κάποια ελλείπουσα τιμή στο διάστημα 1981-90 να υπάρχει ταυτόχρονα ελλείπουσα τιμή και στη χρονοσειρά B, μπορεί ειδικά για εκείνη την τιμή να χρησιμοποιηθεί η χρονοσειρά A.

Οι θέσεις προς συμπλήρωση δεν είναι ανάγκη να αντιστοιχούν σε ελλείπουσες τιμές. Ο χρήστης μπορεί να επιλέξει να πραγματοποιήσει «συμπλήρωση» τιμών που ήδη υπάρχουν (αυτό είναι συνηθισμένο σε μια σύγκριση ή αξιολόγηση των μεθόδων συμπλήρωσης). Σ' αυτή την περίπτωση, η εφαρμογή αγνοεί εντελώς τις υπάρχουσες τιμές που αντιστοιχούν στις θέσεις προς συμπλήρωση, ακριβώς σαν να ήταν ελλείπουσες.

Ο μόνος περιορισμός στην επιλογή των θέσεων προς συμπλήρωση είναι ότι όλες οι χρονοσειρές αναφοράς πρέπει να έχουν υπάρχουσα τιμή στις θέσεις αυτές. Αυτό είναι προφανές, αφού αν δεν συμβαίνει είναι αδύνατο να εφαρμοστεί η μέθοδος στη συγκεκριμένη θέση. Επίσης, πρέπει να αποφεύγεται η ύπαρξη συμπληρωμένων τιμών των χρονοσειρών αναφοράς στις θέσεις προς συμπλήρωση, αλλά η εφαρμογή το επιτρέπει, ενημερώνοντας όμως και προειδοποιώντας το χρήστη κατατοπιστικά σε περίπτωση που εκείνος κάνει μια τέτοια επιλογή. Τέλος, η ύπαρξη μιας ανορθωμένης (ομογενοποιημένης) τιμής χρονοσειράς αναφοράς σε θέση προς συμπλήρωση θεωρείται αποδεκτή.

Η μέθοδος της μέσης τιμής χρησιμοποιεί την εξίσωση (1) (σελ. 13), όπου η μέση τιμή υπολογίζεται από όλες τις τιμές του επιπέδου 3 εκτός από τις συμπληρωμένες, δηλαδή τις τιμές που είτε έχουν περάσει επιτυχώς τον έλεγχο ομογένειας, είτε έχουν αποτύχει και υποστεί ανόρθωση (ομογενοποίηση). Αν η ελλειπής χρονοσειρά είναι μηνιαία ή ημερήσια, τότε το χρησιμοποιούμενο μοντέλο είναι εποχιακό· περιγράφεται δηλαδή από την εξίσωση

$$\hat{y}_i = \bar{y}_i$$

όπου \bar{y}_i η μέση υπερετήσια τιμή για το μήνα i .

Η εφαρμογή της μεθόδου αντιστρόφου αποστάσεως είναι απλή. Γίνεται υπολογισμός των συντελεστών βάρους a_i με την εξίσωση (3) (σελ. 13). Οι αποστάσεις μεταξύ των σταθμών υπολογίζονται από τις συντεταγμένες τους, που είναι αποθηκευμένες στη βάση, ενώ ο εκθέτης p προσδιορίζεται από το χρήστη με ακρίβεια ενός δεκαδικού ψηφίου. Στη συνέχεια υπολογίζονται οι ελλείπουσες τιμές από τον τύπο (2). Στη μέθοδο των υπερετήσιων λόγων, οι συντελεστές βάρους υπολογίζονται από τον τύπο (4) (σελ. 14). Υπολογίζεται μόνο μια υπερετήσια μέση τιμή για κάθε χρονοσειρά, αντίθετα με τη μέθοδο μέσης τιμής, στην οποία υπολογίζονται 12 μέσες τιμές αν η ελλειπής χρονοσειρά είναι μηνιαία ή ημερήσια.

Η μέθοδος της γραμμικής παλινδρόμησης είναι εφαρμογή του τύπου (9) (σελ. 16), ο οποίος καλύπτει και την περίπτωση της απλής γραμμικής παλινδρόμησης όταν $m=1$. Ο χρήστης προσδιορίζει αν θα χρησιμοποιηθεί τυχαίος όρος και αν ο σταθερός όρος θα είναι μηδενικός. Οι παράμετροι a και b , της παλινδρόμησης εκτιμώνται από τους τύπους (10) και (11). Για τον υπολογισμό των παραμέτρων σύμφωνα με αυτές τις σχέσεις απαιτείται ο καθορισμός της κοινής περιόδου μετρήσεων, ή αλλιώς της περιόδου 1. Η περίοδος 1 καθορίζεται λοιπόν ως το σύνολο των χρονικών θέσεων (δηλαδή των ημερών, μηνών ή ετών) για τις οποίες υπάρχουν πραγματικές ομογενείς ή ομογενοποιημένες τιμές στην ελλιπή χρονοσειρά και σε όλες τις χρονοσειρές αναφοράς. Σε περίπτωση που ο σταθερός όρος είναι μηδενικός, χρησιμοποιούνται οι (ίδιοι τύποι, αλλά αντί για διασπορές χρησιμοποιούνται αθροίσματα γινομένων (ροπές περί το μηδέν).

Αν η χρονοσειρά είναι μηνιαία ή ημερήσια, το χρησιμοποιούμενο μοντέλο είναι εποχιακό. Στην ουσία θεωρούνται 12 διαφορετικές χρονοσειρές, μία για κάθε μήνα του έτους, και για κάθε μία από αυτές χωριστά γίνονται οι προαναφερθέντες υπολογισμοί.

Οι τυχαίοι όροι παράγονται από μια σειρά ψευδοτυχαίων αριθμών. Το σημείο έναρξης της παραγωγής των τυχαίων αριθμών καθορίζεται από την ώρα (την οποία παρέχει το λειτουργικό σύστημα) κατά την έναρξη της διαδικασίας συμπλήρωσης.

Τέλος, το πολυμεταβλητό στοχαστικό μοντέλο (21) (σελ. 20) εφαρμόζεται με τον ίδιο τρόπο όπως και η γραμμική παλινδρόμηση, και μάλιστα χρησιμοποιώντας τον ίδιο κώδικα. Απλώς οι χρονοσειρές αναφοράς θεωρούνται περισσότερες κατά μία. Η ελλιπής χρονοσειρά υστερείται κατά 1, και η υστερημένη θεωρείται σαν μια χρονοσειρά αναφοράς χωρίς καμιά ιδιαιτερότητα σε σχέση με τις άλλες χρονοσειρές αναφοράς. Κατόπιν εφαρμόζεται απλά η μέθοδος της γραμμικής παλινδρόμησης. Ο χρήστης και στην περίπτωση του στοχαστικού μοντέλου επιλέγει αν θα είναι μηδενικός ο σταθερός όρος και αν θα χρησιμοποιηθεί τυχαίος όρος.

Όταν η μέθοδος της πολλαπλής γραμμικής παλινδρόμησης χρησιμοποιείται για τη συμπλήρωση ημερήσιων χρονοσειρών βροχοπτώσεων, τότε σε μερικές περιπτώσεις προκύπτει αρνητική βροχόπτωση. Αυτό συμβαίνει όταν η βροχόπτωση παρουσιάζει ασυνήθιστη κατανομή στο χώρο να βρέχει, δηλαδή, σε έναν μόνο από τους γειτονικούς σταθμούς χωρίς να βρέχει στους άλλους. Η αρνητική βροχόπτωση που εμφανίζεται είναι πάντα πολύ κοντά στο μηδέν, π.χ. -0.5 mm . Για την καταπολέμηση αυτού του προβλήματος, αφού γίνει η συμπλήρωση ελέγχεται το πρόσημο της συμπληρωμένης τιμής: αν η τιμή είναι αρνητική, τότε αλλάζεται και τίθεται ίση με το μηδέν.

2.6. Διαχείριση και διακίνηση των δεδομένων

Μεταξύ των πιο σημαντικών ζητημάτων πληροφορικής που πρέπει να λογιστούν είναι η διακίνηση των δεδομένων από τη βάση προς τη μνήμη του υπολογιστή, κυρίως για την περίπτωση των ημερήσιων χρονοσειρών, για τις οποίες ο όγκος των διακινούμενων δεδομένων είναι μεγάλος. Η ανάγνωση των δεδομένων από τη βάση είναι μια σχετικά αργή διαδικασία, ακόμα και όταν τα δεδομένα είναι αποθηκευμένα στον τοπικό κόμβο, ενώ για τους απομακρυσμένους κόμβους είναι βεβαίως πολύ πιο χρονοβόρος. Πρέπει λοιπόν να υπάρχει μέριμνα ώστε να διαβάζεται

ο ελάχιστος αριθμός δεδομένων από τη βάση, αφενός για να μην περιμένει πολλή ώρα ο χρήστης, και κυρίως για να επιβαρύνεται κατά το ελάχιστο το δίκτυο υπολογιστών.

Αυτό το πρόβλημα οδήγησε σε δύο ζητήματα τα οποία έπρεπε να αντιμετωπιστούν αποτελεσματικά. Το πρώτο είναι πως δεν πρέπει να διαβαστούν τα δεδομένα από τη βάση παρά μόνο όταν ξεκινήσει η διαδικασία της συμπλήρωσης. Ας υποθέσουμε, για παράδειγμα, ότι ο χρήστης επιθυμεί να συμπληρώσει τη χρονοσειρά A χρησιμοποιώντας γραμμική παλινδρόμηση με μερικές γειτονικές χρονοσειρές. Κάτι που έχει σημασία να ξέρει ο χρήστης είναι η *πληρότητα* των χρονοσειρών αναφοράς, σε ποια σημεία, δηλαδή, οι χρονοσειρές αναφοράς έχουν ελλείπουσες τιμές. Αυτό εκ πρώτης όψεως μπορεί να επιτευχθεί μόνο αν διαβαστούν τα δεδομένα από τη βάση. Αυτό όμως σημαίνει ότι σε μερικές περιπτώσεις θα διαβαζόντουσαν τα δεδομένα μιας χρονοσειράς από τη βάση μόνο και μόνο για να αποφασίσει ο χρήστης ότι η πληρότητα καθιστά τη χρονοσειρά ακατάλληλη. Για να αποφευχθεί μια τέτοια άσκοπη ανάγνωση των δεδομένων, αποφασίστηκε να υπάρχει στους τοπικούς κόμβους ένας πίνακας χρονοσειρών, ο οποίος περιέχει πληροφορίες πληρότητας για τα δεδομένα όλων των κόμβων. Αυτός ο πίνακας αντιγράφεται στη διάρκεια της νύχτας από κάθε κόμβο στον οποίο υπάρχουν δεδομένα σε όλους τους άλλους κόμβους, και έτσι κάθε 24 ώρες υπάρχει ενημέρωση. Με προσπέλαση αυτού του πίνακα μπορεί να διαπιστωθεί η έκταση της χρονοσειράς και οι ελλείπουσες τιμές σε κάθε επίπεδο. Η εξοικονόμηση οφείλεται αφενός στο ότι ο πίνακας αυτός είναι αποθηκευμένος στον τοπικό κόμβο, και αφετέρου στο ότι τα δεδομένα που περιέχει είναι πολύ λιγότερα.

Κάθε εγγραφή στον πίνακα χρονοσειρών είναι το τμήμα μιας χρονοσειράς δεδομένου επιπέδου στο οποίο όλες οι τιμές είναι υπάρχουσες. Έτσι, η εφαρμογή διαβάζει τον πίνακα και εξάγει συμπεράσματα σχετικά με την επεξεργασία που έχουν υποστεί τα δεδομένα. Οι παραδοχές που γίνονται είναι οι εξής:

- α) Αν μια τιμή υπάρχει σε αμφότερα τα επίπεδα 2 και 3, τότε θεωρείται ως υπάρχουσα που έχει ελεγχθεί για ομογένεια (ενδεχομένως να έχει ανορθωθεί).
- β) Αν μια τιμή δεν υπάρχει σε κανένα από τα επίπεδα 2 ή 3, τότε θεωρείται ως ελλείπουσα.
- γ) Αν μια τιμή υπάρχει μόνο στο επίπεδο 2, τότε θεωρείται ότι δεν έχει ελεγχθεί για ομογένεια.
- δ) Αν μια τιμή υπάρχει μόνο στο επίπεδο 3, τότε θεωρείται πως ήταν ελλείπουσα και έχει συμπληρωθεί.

Σε μερικές περιπτώσεις, όμως, οι παραδοχές αυτές είναι λανθασμένες. Είναι δυνατό να αποφασιστεί κατά τον έλεγχο ομογένειας ότι μια τιμή είναι εντελώς λανθασμένη και να αγνοηθεί. Αν συμβεί κάτι τέτοιο, τότε υπάρχει τιμή στο επίπεδο 2 (λανθασμένη) και στο επίπεδο 3 δεν υπάρχει (ελλείπουσα). Επομένως, σ' εκείνη τη θέση η τιμή είναι ελλείπουσα, ενώ σύμφωνα με τις παραπάνω παραδοχές, θεωρείται μη ελεγμένη για ομογένεια. Επίσης, αν μια τιμή θεωρηθεί λανθασμένη κατά τον έλεγχο ομογένειας και αγνοηθεί, και στη συνέχεια συμπληρωθεί, τότε υπάρχει τιμή και στο επίπεδο 2 (λανθασμένη) και στο επίπεδο 3 (συμπληρωμένη). Σ' εκείνη τη θέση η τιμή είναι συμπληρωμένη, ενώ σύμφωνα με τις παραπάνω παραδοχές θεωρείται υπάρχουσα.

Εντούτοις, η περίπτωση κατά την οποία ένα δεδομένο παραβλέπεται κατά τον έλεγχο ομογένειας είναι σπάνια, γιατί συνήθως οι λανθασμένες τιμές εντοπίζονται από τους ποιοτικούς ελέγχους, οι οποίοι εκτελούνται πριν από τον έλεγχο ομογένειας και είναι οι αρμόδιοι για τη μεταφορά των δεδομένων από το επίπεδο 1 στο επίπεδο 2. Υπάρχει ακόμα και η περίπτωση της διαφοράς μεταξύ των πληροφοριών του πίνακα χρονοσειρών, ο οποίος ενημερώνεται κάθε νύχτα, και των πραγματικών δεδομένων, σε περίπτωση που αυτά έχουν υποστεί κάποια επεξεργασία νωρίτερα εκείνη τη μέρα. Και αυτή η περίπτωση είναι σπάνια. Γι' αυτό, οι παραπάνω παραδοχές θεωρούνται αξιόπιστες. Όταν έρθει η ώρα να διαβαστούν τα δεδομένα από τη βάση, η εφαρμογή επανεξετάζει την πληρότητα, και αν διαπιστώσει πως υπάρχει διαφορά, ενημερώνει το χρήστη και ζητά επιβεβαίωση πριν συνεχίσει.

Δυστυχώς, για να υπολογιστούν πληροφορίες όπως η διασπορά ή άλλες στατιστικές παράμετροι μιας χρονοσειράς (εκτός από τη μέση τιμή), πρέπει να διαβαστούν τα δεδομένα από τη βάση. Ίσως σε μελλοντικές βελτιώσεις των εφαρμογών να υπάρξει η δυνατότητα απομακρυσμένης επεξεργασίας (*remote processing*), ώστε οι υπολογισμοί να εκτελούνται στον απομακρυσμένο κόμβο, αντί να μεταφέρονται τα δεδομένα στον τοπικό.

Το δεύτερο ζήτημα που αντιμετωπίστηκε σχετικά με την ελαχιστοποίηση των αναγνώσεων από τη βάση ήταν η διατήρηση στον τοπικό κόμβο των δεδομένων που διαβάστηκαν από έναν απομακρυσμένο, για ένα χρονικό διάστημα. Αν ένας χρήστης πραγματοποιεί συμπληρώσεις με μερικές χρονοσειρές, είναι πολύ πιθανό να ξαναχρησιμοποιήσει μερικές από αυτές τις χρονοσειρές λίγο αργότερα, σε άλλη διαδικασία συμπλήρωσης. Δεν πρέπει λοιπόν να ξαναδιαβαστούν τα δεδομένα από απομακρυσμένους κόμβους πολλές φορές. Υπήρχαν τρεις τρόποι να αντιμετωπιστεί αυτό το ζήτημα: ή τα δεδομένα, όταν διαβάζονται από απομακρυσμένο κόμβο, να αντιγράφονται σε προσωρινό πίνακα της βάσης δεδομένων (ο οποίος να σβήνεται κάθε νύχτα), είτε να αντιγράφονται σε προσωρινό αρχείο, είτε να παραμένουν στη μνήμη (RAM) του υπολογιστή. Οι πρώτοι δύο από αυτούς θα πολλαπλασίαζαν υπερβολικά την πολυπλοκότητα της βάσης δεδομένων ή των εφαρμογών (περισσότεροι πίνακες στη βάση, πρόβλεψη για διαγραφή των προσωρινών δεδομένων, προβλήματα στη συνεργασία των εφαρμογών κατά την προσπέλαση των προσωρινών πινάκων), και επιλέχτηκε ο τρίτος. Η εφαρμογή κρατά έναν κατάλογο όλων των δεδομένων που έχουν διαβαστεί από τη βάση, και τα κρατά στη μνήμη, ακόμα και αν ο χρήστης τα διαγράψει, μέχρι τη στιγμή που η εφαρμογή σταματά να τρέχει. Έτσι, πριν αναγνωστούν από τη βάση, η εφαρμογή εξετάζει μήπως ήδη βρίσκονται στον κατάλογο που διατηρεί, και, αν ναι, δεν τα διαβάσει.

Αυτό το σύστημα έχει βεβαίως μερικά σημαντικά μειονεκτήματα απέναντι στις άλλες δύο εναλλακτικές λύσεις. Αν ο χρήστης θελήσει να κλείσει την εφαρμογή επειδή θα λείψει για λίγο, τα δεδομένα χάνονται και πρέπει να ξαναδιαβαστούν όταν έρθει. Αν μια άλλη εφαρμογή έχει διαβάσει τα ίδια δεδομένα πριν από λίγο, η εφαρμογή συμπλήρωσης δεν το γνωρίζει και πρέπει επίσης να τα διαβάσει. Όμως, λόγω της πολυπλοκότητας των άλλων λύσεων, η μέθοδος που τελικά υλοποιήθηκε ήταν η μόνη πρακτικά εφαρμόσιμη. Όταν ο χρήστης ζητήσει έξοδο από την

εφαρμογή, αυτή τον προειδοποιεί, σε περίπτωση που υπάρχουν στη μνήμη δεδομένα απομακρυσμένου κόμβου, να μην εξέλθει αν πρόκειται να ξανακάνει συμπληρώσεις λίγο αργότερα.

ΚΕΦΑΛΑΙΟ 3: ΕΦΑΡΜΟΓΗ ΣΕ ΗΜΕΡΗΣΙΕΣ ΚΑΙ ΜΗΝΙΑΙΕΣ ΒΡΟΧΟΠΤΩΣΕΙΣ ΜΕ ΣΥΝΤΟΜΗ ΑΞΙΟΛΟΓΗΣΗ ΤΩΝ ΜΕΘΟΔΩΝ

3.1. Μεθοδολογία

Για την αξιολόγηση των μεθόδων πραγματοποιήθηκαν συμπληρώσεις τιμών που ήδη υπήρχαν, και έγινε σύγκριση των εκτιμημένων τιμών με τις πραγματικές. Οι χρονοσειρές που χρησιμοποιήθηκαν είχαν έκταση περίπου 30 ετών (1960-1990). Αρχικά θεωρήθηκε ως ελλείπον όλο το έτος 1970 (365 τιμές για τις ημερήσιες και 12 για τις μηνιαίες χρονοσειρές) και συμπληρώθηκε με όλες τις μεθόδους. Στη συνέχεια έγινε το ίδιο για το έτος 1971, κλπ, ως το έτος 1979, δηλαδή 10 έτη συνολικά. Ως μέτρο σύγκρισης των μεθόδων χρησιμοποιήθηκε ο *συντελεστής απόδοσης* της εκτίμησης, που ορίζεται ως εξής:

$$EC = 1 - \frac{MSE}{var}$$

όπου *MSE* το μέσο τετραγωνικό σφάλμα της εκτίμησης και *var* η διασπορά του πληθυσμού, που εκτιμάται από τη διασπορά του δείγματος. Ο συντελεστής απόδοσης θεωρήθηκε χωριστά για κάθε μήνα.

Όπως έχει αναφερθεί στο κεφάλαιο 2, αυτός ο τρόπος ελέγχου είναι ορθός μόνο όταν είναι επιθυμητή η «πιθανότερη»¹ εκτίμηση αν επιθυμείται η διατήρηση των στατιστικών χαρακτηριστικών της χρονοσειράς, πρέπει να ληφθούν υπόψη και άλλες παράμετροι.

3.2. Επιλογή σταθμών

Κατά την επιλογή των σταθμών που θα χρησιμοποιόντουσαν για το πείραμα αντιμετώπιστηκαν προβλήματα συνηθισμένα, δυστυχώς, στον ελληνικό χώρο. Σε πάρα πολλές περιπτώσεις παρατηρήθηκε, με σύγκριση με γειτονικούς σταθμούς, μετατόπιση της χρονοσειράς κατά μία μέρα για μεγάλα χρονικά διαστήματα (π.χ. 6 μήνες), προφανώς λόγω λανθασμένης καταγραφής από τον παρατηρητή. Επίσης, παρατηρήθηκαν περιπτώσεις όπως η εξής, στην οποία παρουσιάζονται παρατηρήσεις βροχοπτώσεων σε δυο σταθμούς που απέχουν 5.7 km, με ψηλό συντελεστή συσχέτισης:

	ΚΡΕΜΑΣΤΑ (ΔΕΗ)	ΤΡΙΠΟΤΑΜΟ (ΔΕΗ)
06/07/1970	0	0
07/07/1970	26.0	0
08/07/1970	23.0	40.6
09/07/1970	0	0

¹ Ο όρος «πιθανότερη» είναι ίσως κατάλληλος για κατανόηση του εννοούμενου, αλλά από μαθηματικής πλευράς είναι λανθασμένος. Μαθηματικώς εννοείται η εκτίμηση με το ελάχιστο τετραγωνικό σφάλμα.

Είναι πιθανό τα 40.6 mm βροχόπτωσης στο Τριπόταμο να είναι μοιρασμένα σε δύο μέρες, αλλά ο παρατηρητής να παρέλειψε να πάει μια μέρα στο σταθμό.

Ένα ακόμα πρόβλημα είναι ότι η ημερήσια βροχόπτωση στους σταθμούς της ΕΜΥ μετρείται από τις 8.00 το βράδυ ως τις 8.00 το άλλο βράδυ, ενώ στις υπόλοιπες υπηρεσίες μετρείται από τις 8.00 το πρωί ως τις 8.00 το άλλο πρωί. Έτσι, δεν μπορούν να γίνουν καλές συσχετίσεις ημερήσιων βροχοπτώσεων μεταξύ σταθμών της ΕΜΥ και άλλων υπηρεσιών. Γι' αυτό αποφασίστηκε οι τρεις σταθμοί να είναι της ίδιας υπηρεσίας.

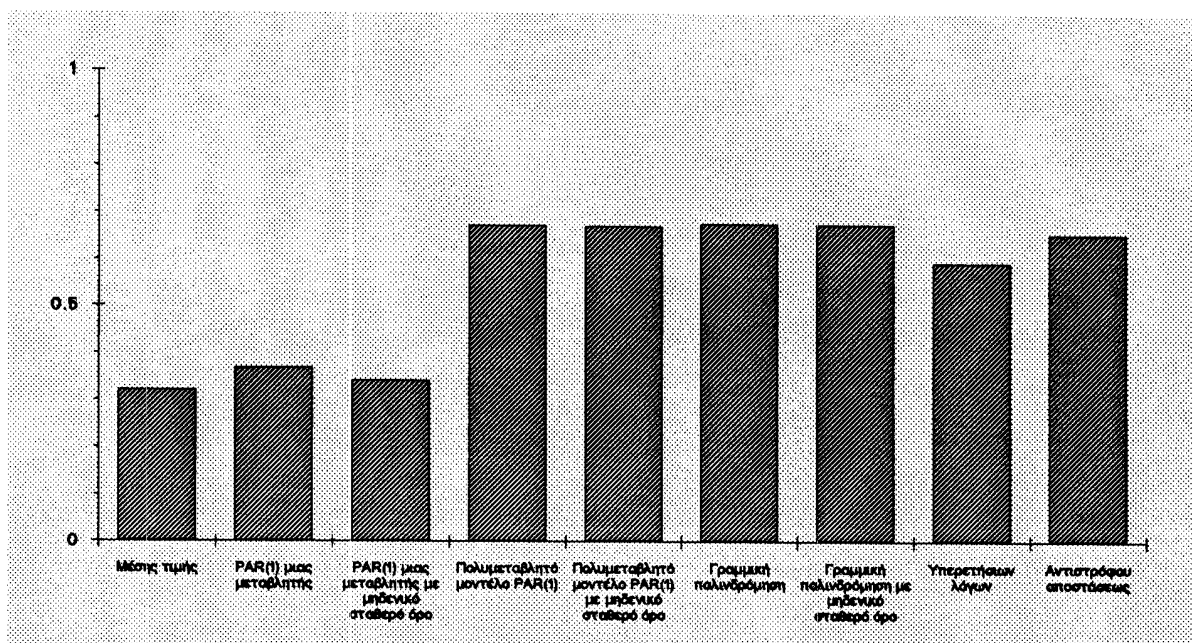
Στη βάση δεδομένων υπήρχαν διαθέσιμες μετρήσεις για 30 περίπου χρόνια (1960-1990) σε 70 περίπου σταθμούς της Στερεάς Ελλάδας, οι περισσότεροι από τους οποίους ήταν της ΔΕΗ. Επιλέχθηκαν οι εξής τρεις σταθμοί της ΔΕΗ, που βρίσκονται στη λεκάνη απορροής του Αχελώου, στην υπολεκάνη του ποταμού Αγραφιώτη:

	Τοποθεσία	Υψόμετρο	Απόσταση από ελλιπή
Ελλιπής Αναφοράς	Επινιανά Ευρυτανίας	1050 m	
Αναφοράς	Μοναστηράκι Μοναστηρακίου Ευρυτανίας	660 m	6.4 km
Αναφοράς	Τροβάτο Ευρυτανίας	1060 m	9.5 km

Οι σταθμοί αυτοί επιλέχθηκαν γιατί ήταν αρκετά κοντά, και, όπως φάνηκε, δεν υπήρχαν μετατοπίσεις των χρονοσειρών σαν αυτές που αναφέρονται παραπάνω. Κατά τα άλλα, η επιλογή τους ήταν τυχαία. Υπάρχουν και άλλες ομάδες σταθμών που θα μπορούσαν να επιλεγούν. Στην πράξη, τα πράγματα είναι βεβαίως δυσμενέστερα, γιατί δεν είναι απαραίτητο να υπάρχουν σταθμοί κοντά σ' αυτόν που συμπληρώνεται. Όσον αφορά στις χρονικές μετατοπίσεις, αυτές μπορούν να εντοπιστούν και να διορθωθούν με ελέγχους συνέπειας στο χώρο.

3.3. Συμπληρώσεις ημερήσιων χρονοσειρών: Αποτελέσματα και παρατηρήσεις

Για κάθε μήνα του έτους έχει υπολογιστεί ο συντελεστής απόδοσης για κάθε μέθοδο και παρατίθεται σε ραβδόγραμμα. Στο σχήμα 1 φαίνονται οι συντελεστές απόδοσης για το μήνα Ιανουάριο. Τα σχήματα για όλους τους μήνες βρίσκονται στο παράρτημα Α (σελ. 37).



Σχήμα 1 - Συντελεστές απόδοσης Ιανουαρίου - ημερήσιες βροχοπτώσεις

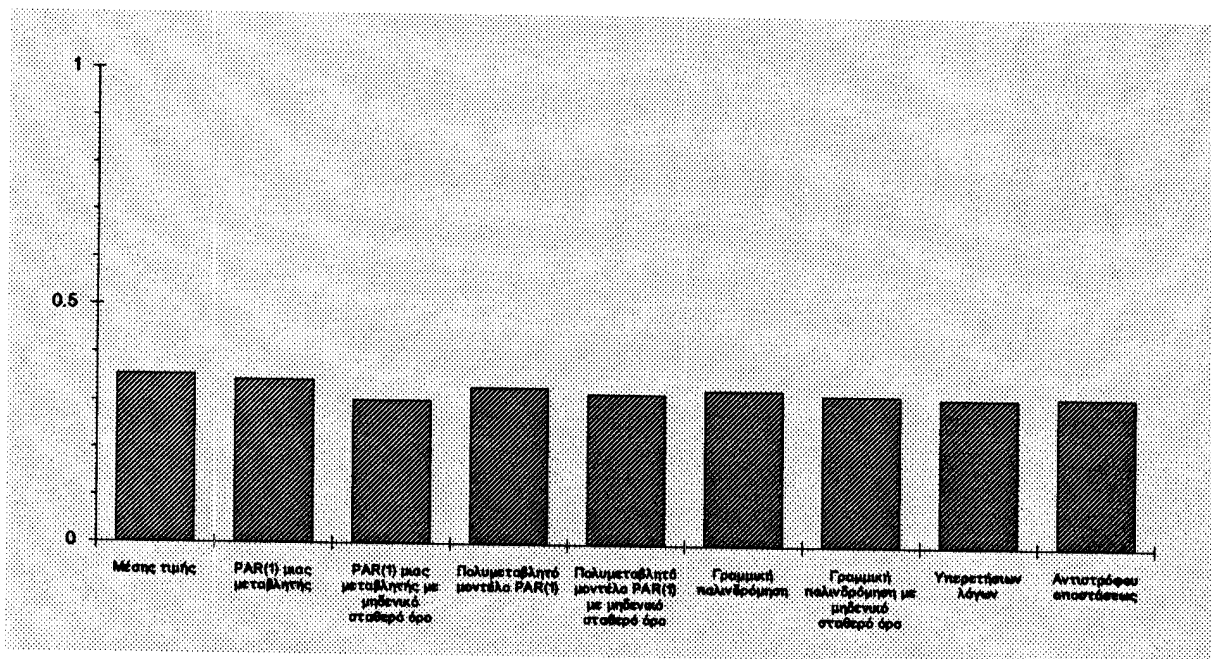
Επίσης, στο παράρτημα Β (σελ. 45) δίνεται ενδεικτικά το αποτέλεσμα που παράγει η εφαρμογή για το έτος 1975, για συμπλήρωση με το πολυμεταβλητό στοχαστικό μοντέλο. Εκεί φαίνονται και οι συντελεστές συσχέτισης μεταξύ της ελλιπούς χρονοσειράς και των δύο χρονοσειρών αναφοράς, καθώς και ο συντελεστής αυτοσυσχέτισης της ελλιπούς με υστέρηση 1. Από τα διαγράμματα και τους συντελεστές συσχέτισης παρατηρούμε τα εξής:

- 1) Οι μέθοδοι PAR(1) και μέσης τιμής, στις οποίες δεν γίνεται συσχέτιση με γειτονικές χρονοσειρές, παρουσιάζουν σημαντικά μικρότερη απόδοση από τις υπόλοιπες μεθόδους.
- 2) Η μέθοδος της γραμμικής παλινδρόμησης και του πολυμεταβλητού στοχαστικού μοντέλου δίνουν όλες την ίδια περίπου απόδοση, που είναι μεγαλύτερη από αυτή όλων των άλλων μεθόδων.
- 3) Η χρήση περισσότερων μεταβλητών, πέραν αυτής που έχει τη μεγαλύτερη συσχέτιση, ελάχιστα ή καθόλου δεν βελτιώνει το αποτέλεσμα. Για παράδειγμα, ο συντελεστής γραμμικής συσχέτισης Ιανουαρίου για συσχέτιση μόνο με το Μοναστηράκι είναι 0.8548. Αν ληφθούν επιπλέον υπόψη το Τροβάτο και η αυτοσυσχέτιση, με αντίστοιχους συντελεστές συσχέτισης 0.6089 και 0.4105, ο συνολικός συντελεστής συσχέτισης για την πολλαπλή γραμμική γραμμική παλινδρόμηση είναι 0.8614, ελάχιστα μεγαλύτερος, δηλαδή, από 0.8548.
- 4) Η αυτοσυσχέτιση είναι πολύ μικρή σε σχέση με τη συσχέτιση με τους γειτονικούς σταθμούς για την περίπτωση των βροχοπτώσεων.
- 5) Οι συντελεστές συσχέτισης για τους χειμερινούς μήνες είναι γενικά ψηλοί, ακόμα και για την αυτοσυσχέτιση. Οι αριθμοί είναι όμως κάπως παραπλανητικοί, γιατί υπάρχουν πολλές μέρες στις οποίες δεν βρέχει καθόλου, οι οποίες λαμβάνονται υπόψη στον υπολογισμό.
- 6) Οι συντελεστές συσχέτισης για τους θερινούς μήνες είναι σημαντικά μικρότεροι απ' ό,τι για τους χειμερινούς, παρόλο που ελάχιστες μέρες βρέχει. Αυτό συμβαίνει γιατί οι βροχές το

καλοκαίρι είναι περισσότερο αποτέλεσμα θερμικών ασταθειών της ατμόσφαιρας και εμφανίζονται τοπικά.

- 7) Η μέθοδος αντιστρόφου αποστάσεως δίνει καλύτερα αποτελέσματα από τη μέθοδο των υπερετήσιων λόγων, που για μερικούς μήνες είναι μάλιστα ελαφρώς καλύτερα και από αυτά της γραμμικής παλινδρόμησης. Αυτό είναι όμως παραπλανητικό, γιατί συμβαίνει οι υπερετήσιοι μέσοι των χρονοσειρών αναφοράς, και κυρίως της πλησιέστερης (5.1 mm) να είναι πολύ κοντά στον υπερετήσιο μέσο της ελλιπούς (4.9 mm). Στη γενική περίπτωση αυτό δεν ισχύει και η μέθοδος αντιστρόφου αποστάσεως δίνει λανθασμένα αποτελέσματα.

Κάτι που δεν φαίνεται στα διαγράμματα είναι η πιθανότητα εντελώς λανθασμένης εκτίμησης. Στο σχήμα 2 φαίνονται οι συντελεστές απόδοσης μόνο για το Σεπτέμβριο του 1975, στον οποίο όλες οι μέθοδοι έχουν δώσει την ίδια περίπου απόδοση, που, ειδικά για τη μέθοδο γραμμικής παλινδρόμησης, είναι πολύ μικρότερη από τη συνηθισμένη (βλ. διάγραμμα Σεπτεμβρίου στο παράρτημα Α).



Σχήμα 2 - Συντελεστές απόδοσης Σεπτεμβρίου 1975 (ημερήσιες βροχοπτώσεις)

Αυτό οφείλεται σε μια ασυνήθιστη βροχοπτώση 31.9 mm στις 4 Σεπτεμβρίου στα Επινιανά (βλ. αποτελέσματα στο παράρτημα Β), ενώ την ίδια μέρα δεν έβρεξε καθόλου στους δύο γειτονικούς σταθμούς. Έτσι, η εκτίμηση για εκείνη τη μέρα είναι 0 ή κοντά στο 0 για όλες τις μεθόδους, ενώ η πραγματική τιμή είναι 31.9 mm. Αυτή η διαφορά είναι καθοριστική για το μέσο τετραγωνικό σφάλμα, αφού οι υπόλοιπες βροχοπτώσεις για το μήνα αυτό είναι 12.9 mm όλες μαζί.

Αυτό το παράδειγμα δείχνει πως όλες οι μέθοδοι έχουν μεγάλα περιθώρια εσφαλμένων εκτιμήσεων, αφού η βροχοπτώση είναι φαινόμενο του οποίου η κατανομή στο χώρο είναι συχνά απρόβλεπτη. Άλλα μεγέθη, όπως η θερμοκρασία, παρουσιάζουν πολύ μεγαλύτερους συντελεστές γραμμικής συσχέτισης, της τάξεως του 0.99, συνέπεια της ομαλότητας που παρουσιάζει η κατανομή τους στο χώρο.

3.4. Συμπληρώσεις μηνιαίων χρονοσειρών: Αποτελέσματα και παρατηρήσεις

Οι συντελεστές απόδοσης παρατίθενται (ενδεικτικά για μερικούς μήνες) σε διαγράμματα στο παράρτημα Α. Είναι παράτολμο να εξαχθούν συμπεράσματα, λόγω της μεγάλης ποικιλίας που παρουσιάζουν τα διαγράμματα. Αυτή οφείλεται κυρίως στο γεγονός ότι τα δείγματα από τα οποία έχει προκύψει το μέσο τετραγωνικό σφάλμα είναι πολύ μικρά (10 τιμές), ενώ στις ημερήσιες βροχοπτώσεις ήταν περίπου 300 τιμές για κάθε δείγμα.

Περισσότερα συμπεράσματα μπορούν να εξαχθούν από τους συντελεστές γραμμικής συσχέτισης, που φαίνονται στο παράρτημα Β στις μηνιαίες βροχοπτώσεις (σελίδα 53), αλλά και η γραμμική παλινδρόμηση έχει γίνει μόνο με 30 περίπου παρατηρήσεις, έναντι 900-1000 για τις ημερήσιες βροχοπτώσεις. Οι παρατηρήσεις που μπορούμε να κάνουμε είναι οι εξής:

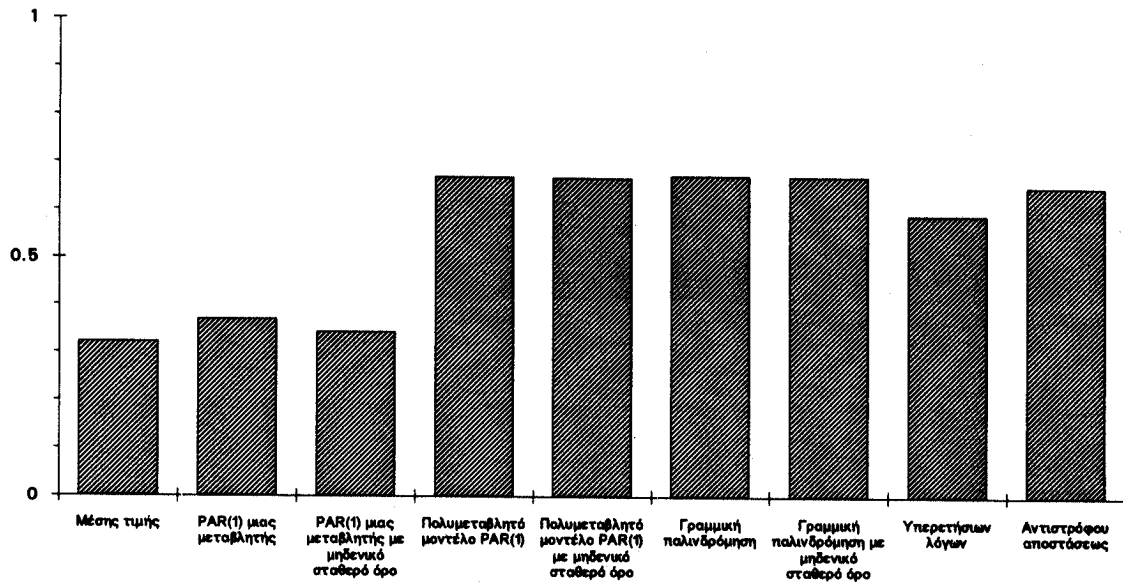
- 1) Οι συντελεστές συσχέτισης φαίνονται ψηλότεροι απ' ό,τι για τις ημερήσιες βροχοπτώσεις.
- 2) Οι συντελεστές συσχέτισης είναι ψηλότεροι για τους χειμερινούς μήνες απ' ό,τι για τους θερινούς.
- 3) Η αυτοσυσχέτιση (συσχέτιση κάθε μήνα με τον προηγούμενο) είναι πολύ μικρή.

3.5. Περαιτέρω δυνατότητες

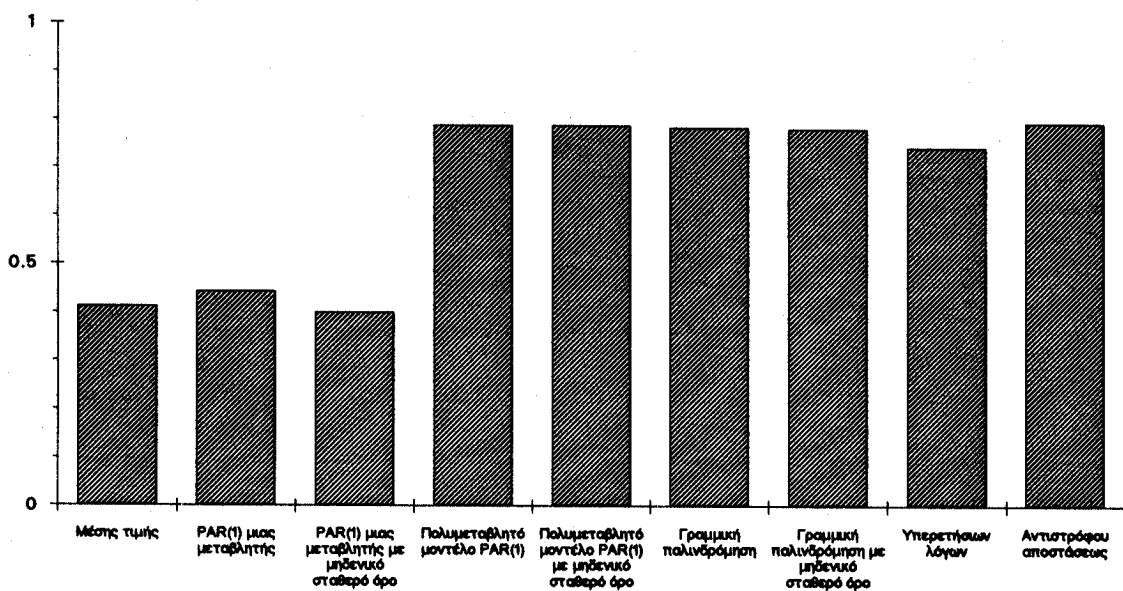
Προκειμένου να εξαχθούν περισσότερα χρήσιμα συμπεράσματα για τις μηνιαίες βροχοπτώσεις, πρέπει να γίνουν περισσότερες συμπληρώσεις, ώστε η εκτίμηση του μέσου τετραγωνικού σφάλματος να προκύπτει από μεγαλύτερο δείγμα. Ακόμα όμως κι αν γινόταν αυτό για όλα τα έτη για τα οποία υπάρχουν παρατηρήσεις (περίπου 30), το δείγμα θα ήταν μικρό. Δείγματα που αποτελούνται από 30 τιμές έχουν μεγάλα περιθώρια σφαλμάτων, όπως φαίνεται από το παράδειγμα Σεπτεμβρίου 1975 που αναφέρθηκε στις συμπληρώσεις ημερήσιων χρονοσειρών.

Μια ακόμα μέθοδος συμπλήρωσης μηνιαίων χρονοσειρών που αξίζει να διερευνηθεί είναι η συμπλήρωση της ημερήσιας χρονοσειράς και η άθροιση της συμπληρωμένης. Αυτή η μέθοδος δοκιμάστηκε ενδεικτικά για το μήνα Ιανουάριο. Οι ημερήσιες βροχοπτώσεις συμπληρώθηκαν (διαδοχικά για τα έτη 1970, 1971 κλπ) με γραμμική παλινδρόμηση με μηδενικό σταθερό όρο. Ο συντελεστής απόδοσης για την εκτίμηση της μηνιαίας τιμής προκύπτει (σος με 0.85, έναντι 0.81 που δίνει η αποτελεσματικότερη (με βάση τα υπάρχοντα δεδομένα) από τις υπόλοιπες μεθόδους, η γραμμική παλινδρόμηση με μηδενικό σταθερό όρο, εφαρμοσμένη κατευθείαν στη μηνιαία χρονοσειρά.

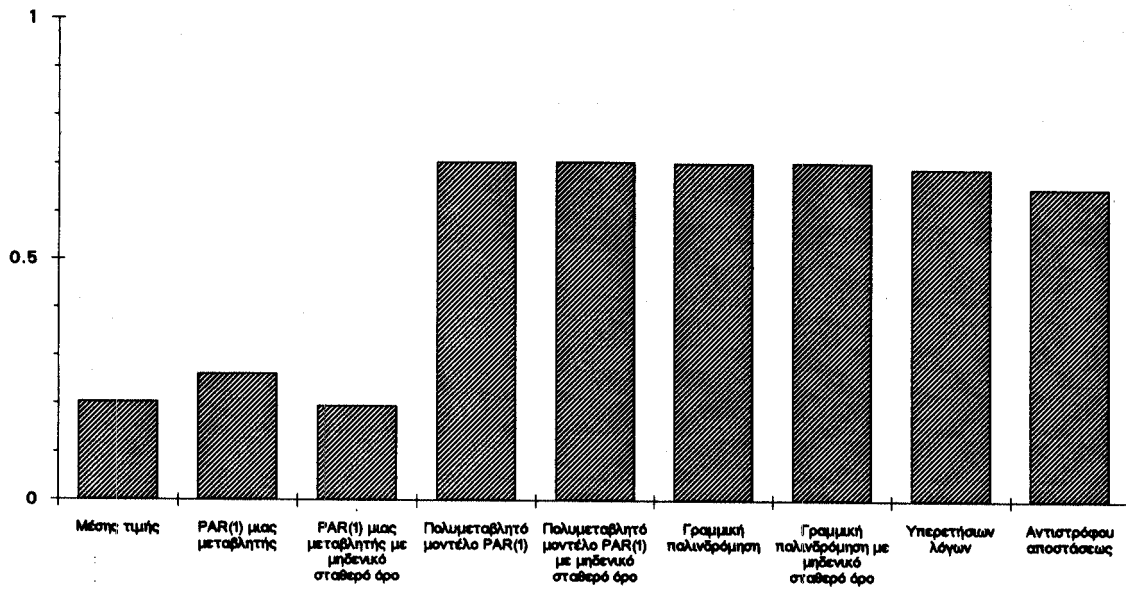
ΠΑΡΑΡΤΗΜΑ Α: ΔΙΑΓΡΑΜΜΑΤΑ ΣΥΝΤΕΛΕΣΤΩΝ ΑΠΟΔΟΣΗΣ



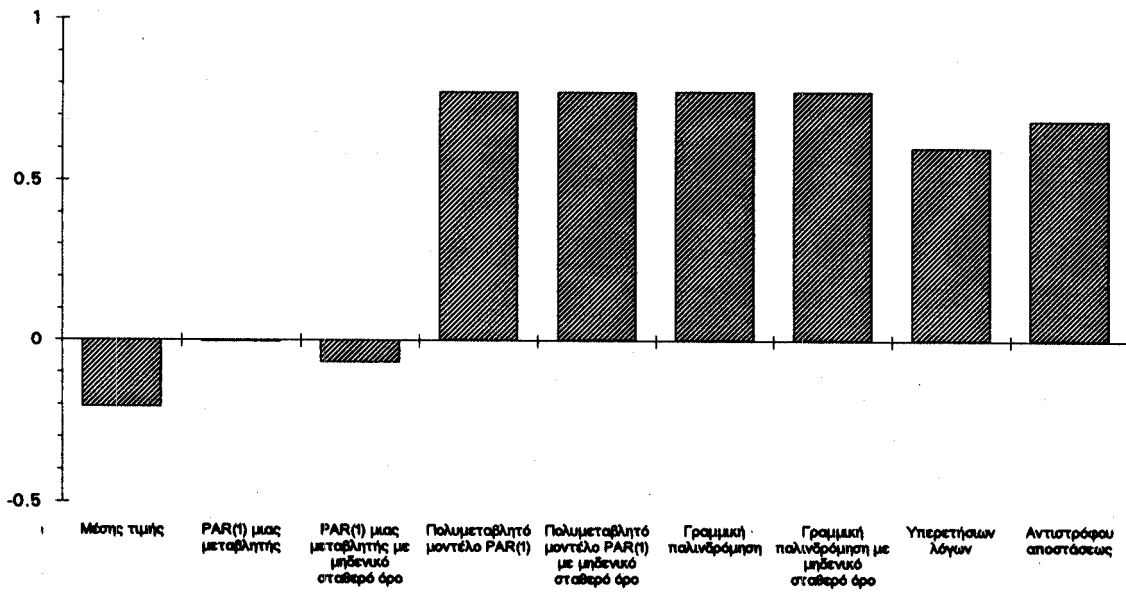
Συντελεστής απόδοσης Ιανουαρίου - ημερήσιες βροχοπτώσεις



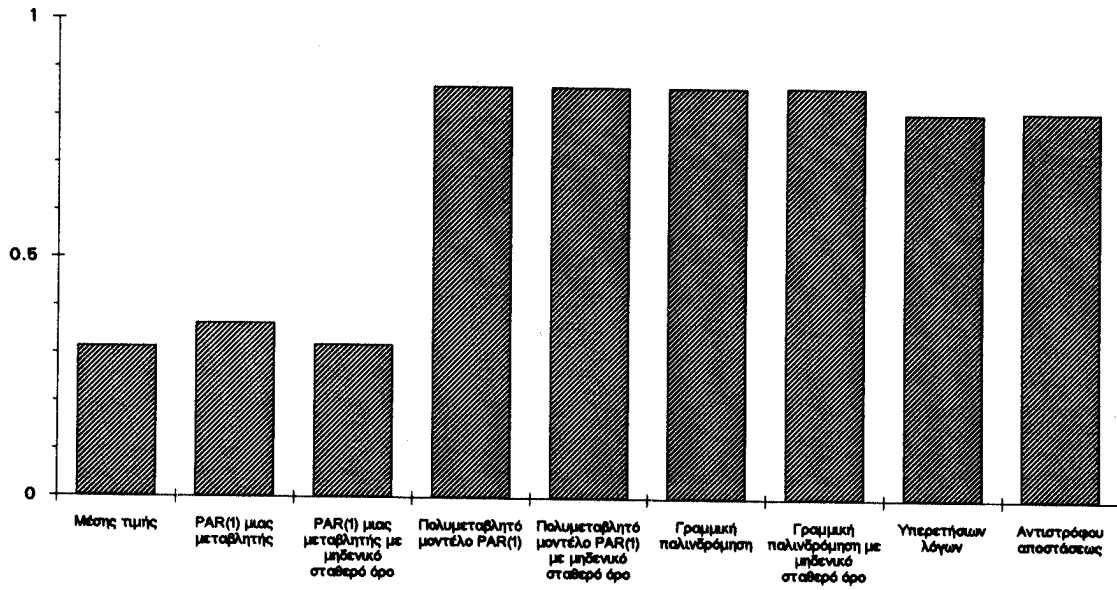
Συντελεστής απόδοσης Φεβρουαρίου - ημερήσιες βροχοπτώσεις



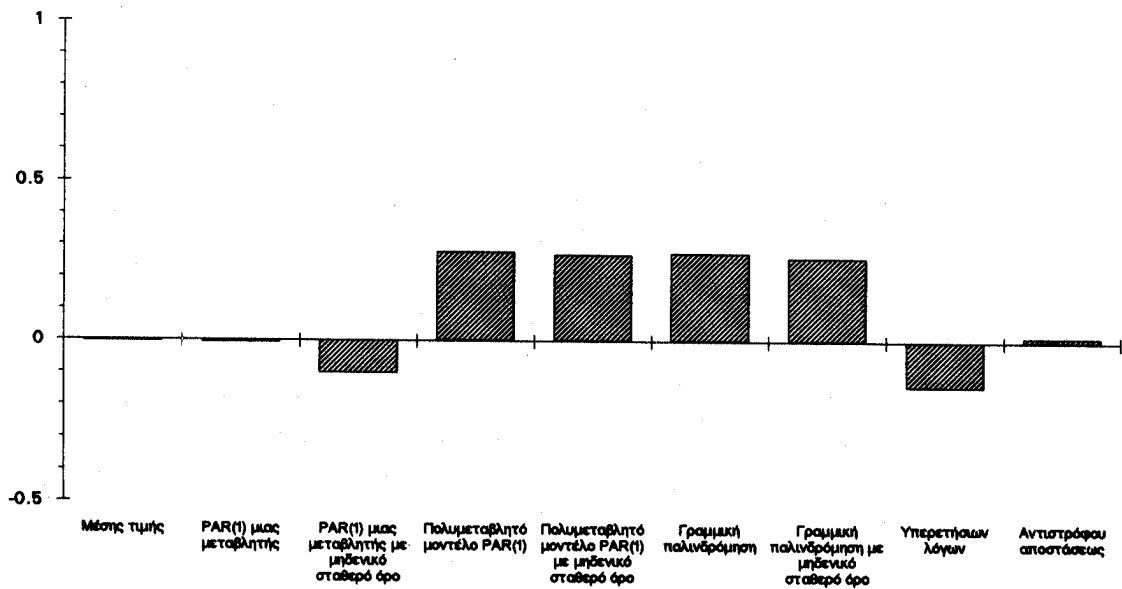
Συντελεστής απόδοσης Μαρτίου - ημερήσιες βροχοπτώσεις



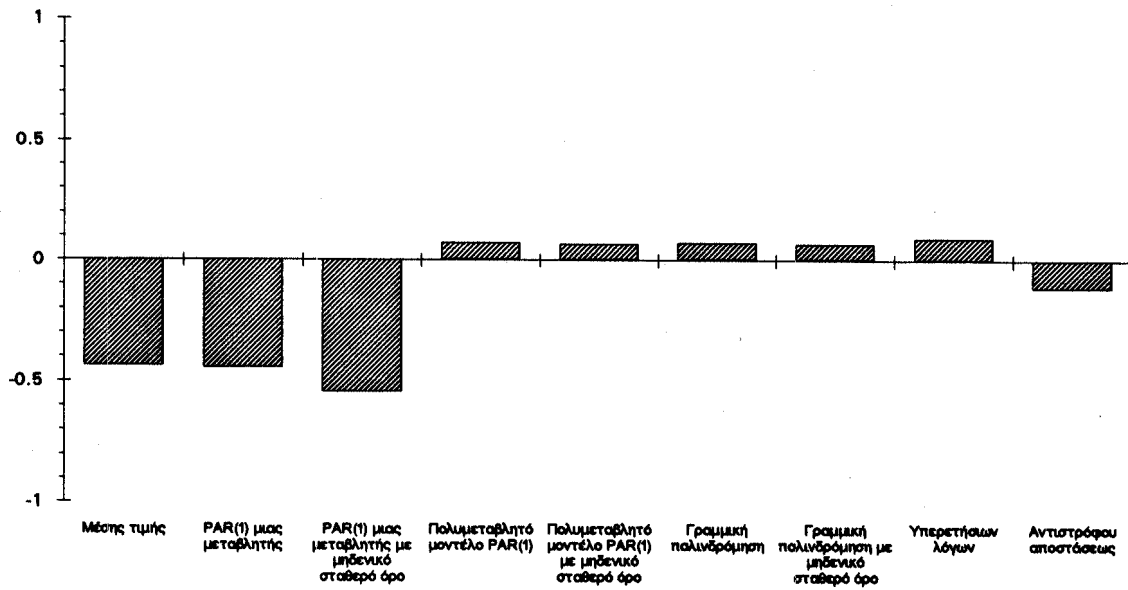
Συντελεστής απόδοσης Απριλίου - ημερήσιες βροχοπτώσεις



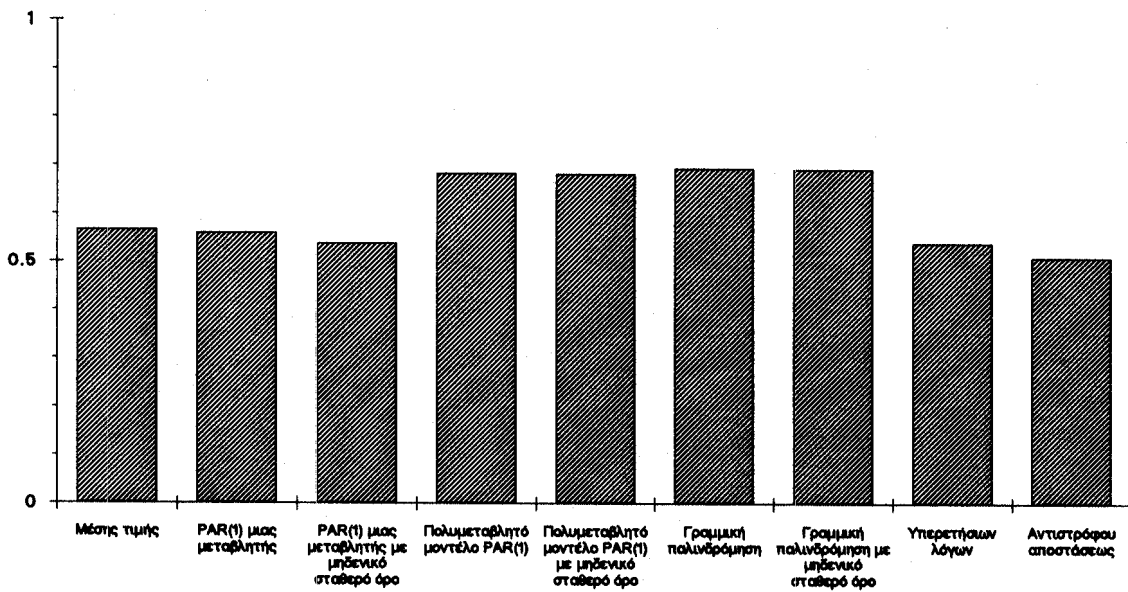
Συντελεστής απόδοσης Μαΐου - ημερήσιες βροχοπτώσεις



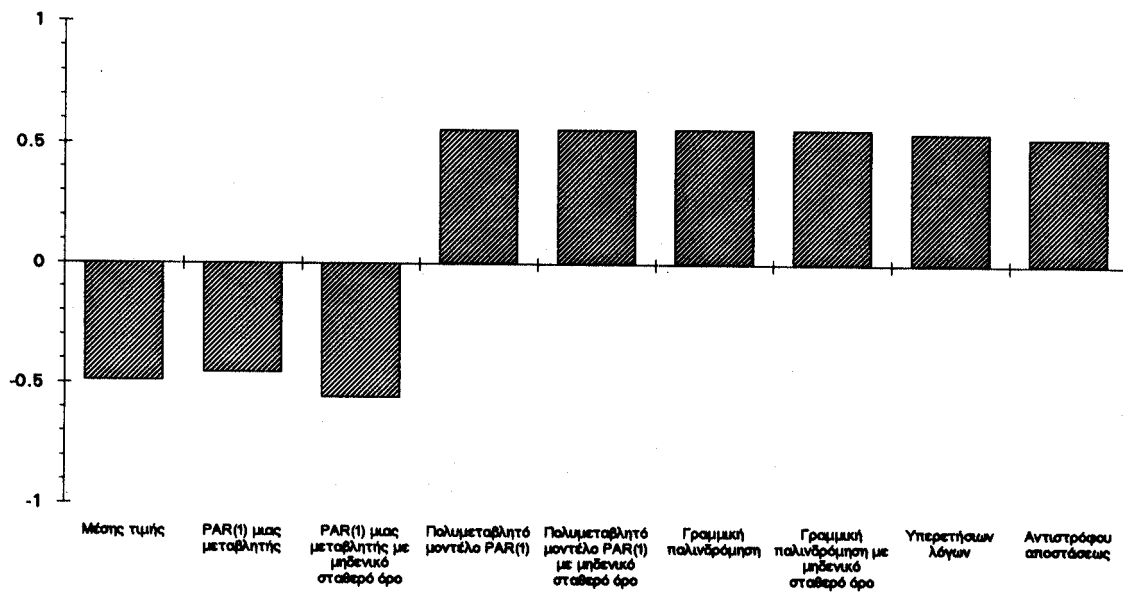
Συντελεστές απόδοσης Ιουνίου - ημερήσιες βροχοπτώσεις



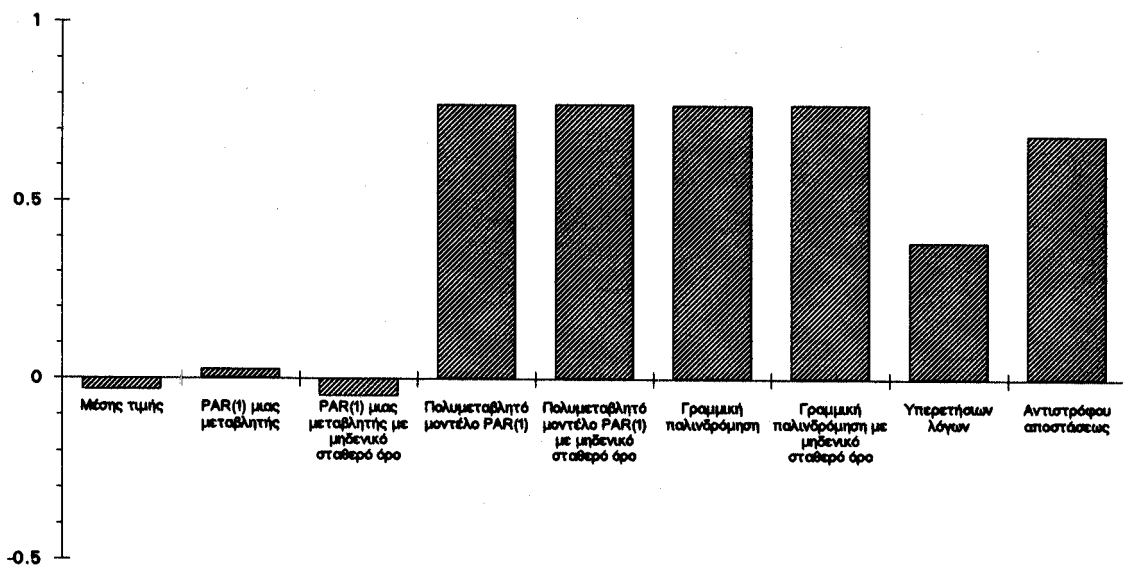
Συντελεστές απόδοσης Ιουλίου - ημερήσιες βροχοπτώσεις



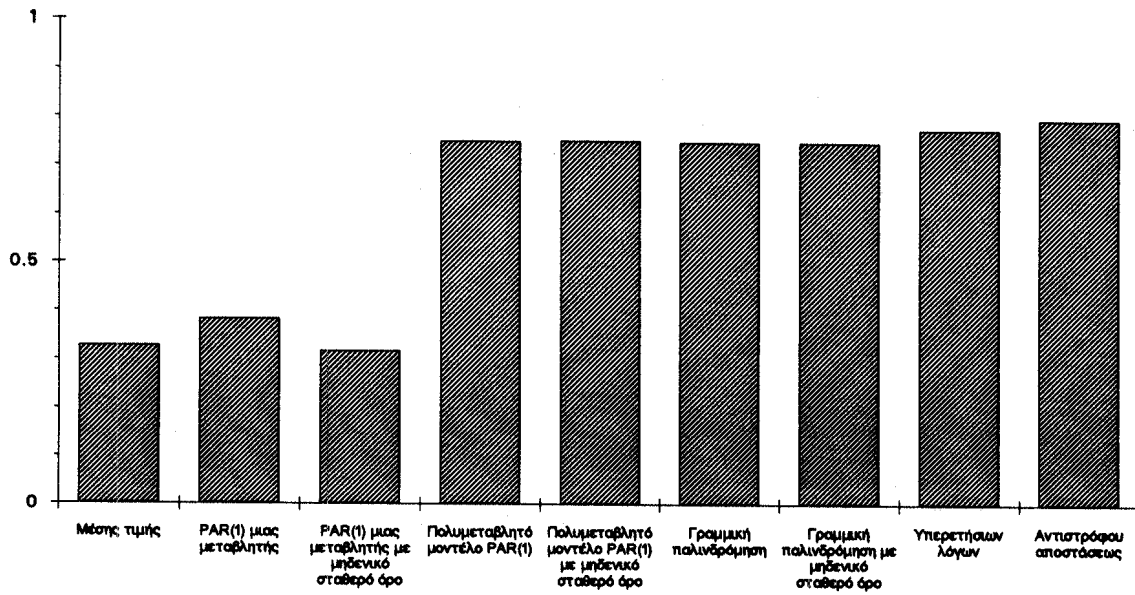
Συντελεστές απόδοσης Αυγούστου - ημερήσιες βροχοπτώσεις



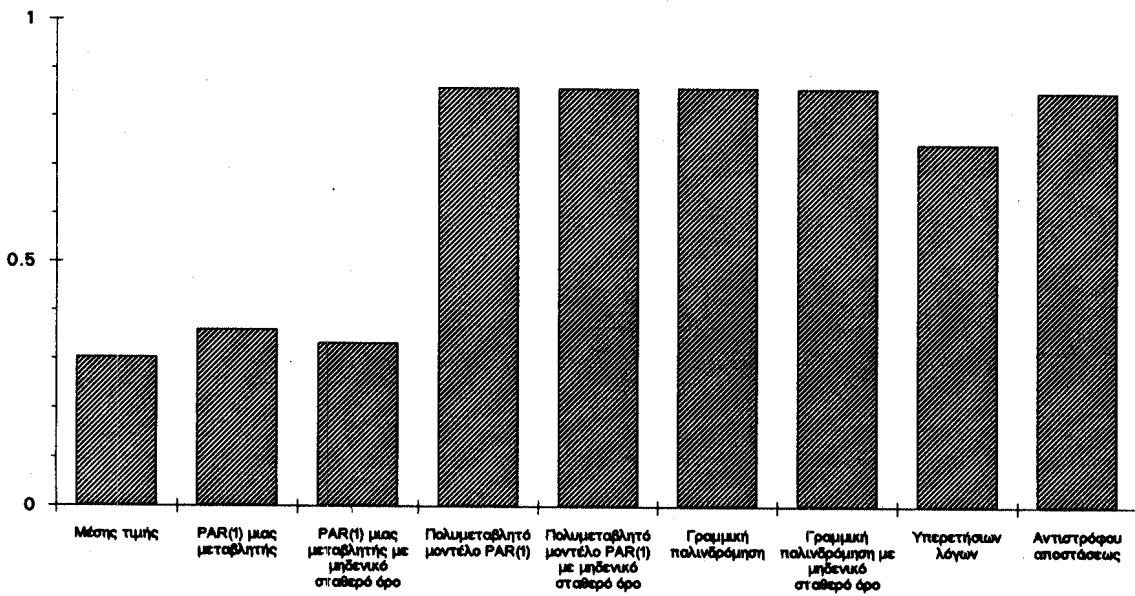
Συντελεστές απόδοσης Σεπτεμβρίου - ημερήσιες βροχοπτώσεις



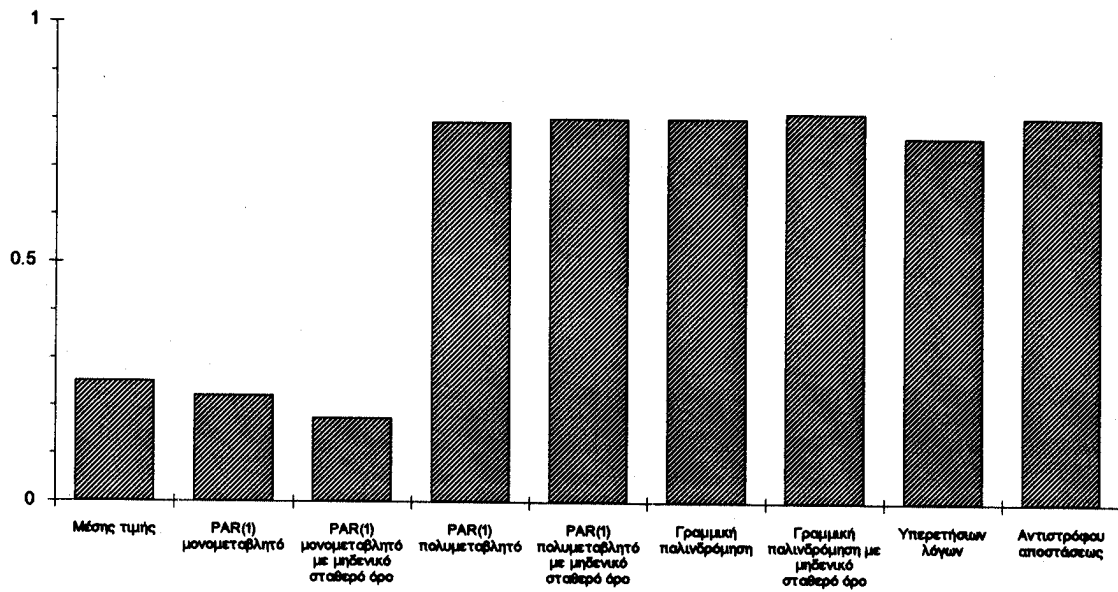
Συντελεστές απόδοσης Οκτωβρίου - ημερήσιες βροχοπτώσεις



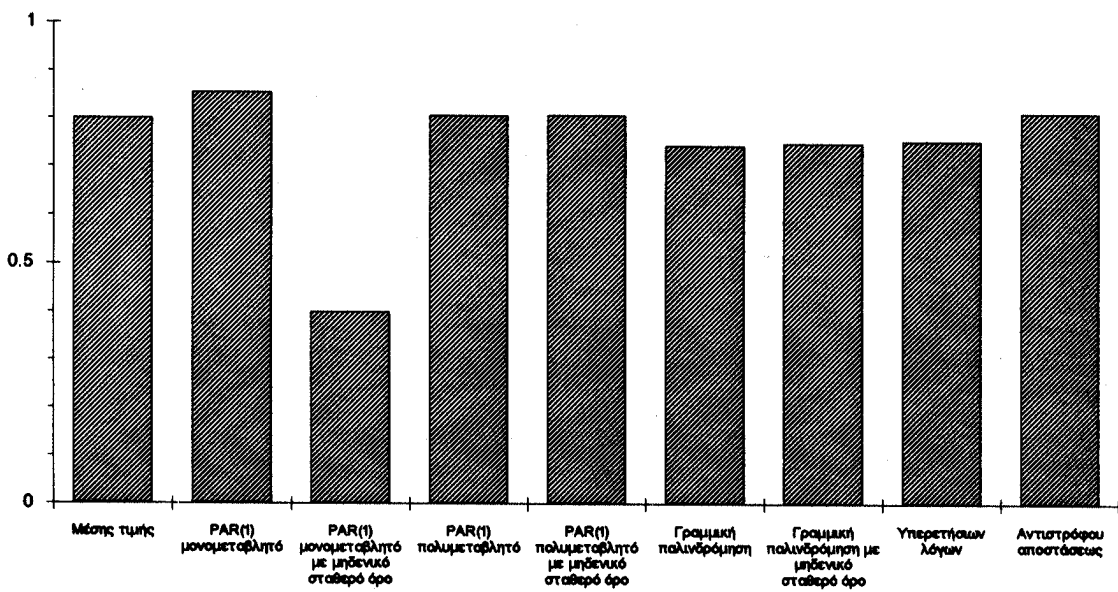
Συντελεστές απόδοσης Νοεμβρίου - ημερήσιες βροχοπτώσεις



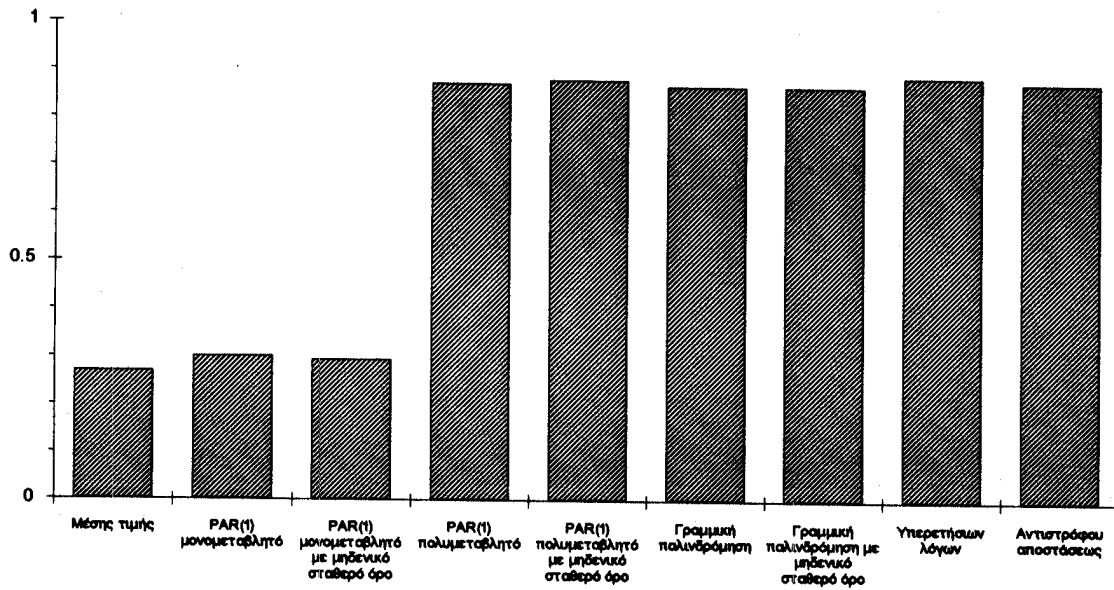
Συντελεστές απόδοσης Δεκεμβρίου - ημερήσιες βροχοπτώσεις



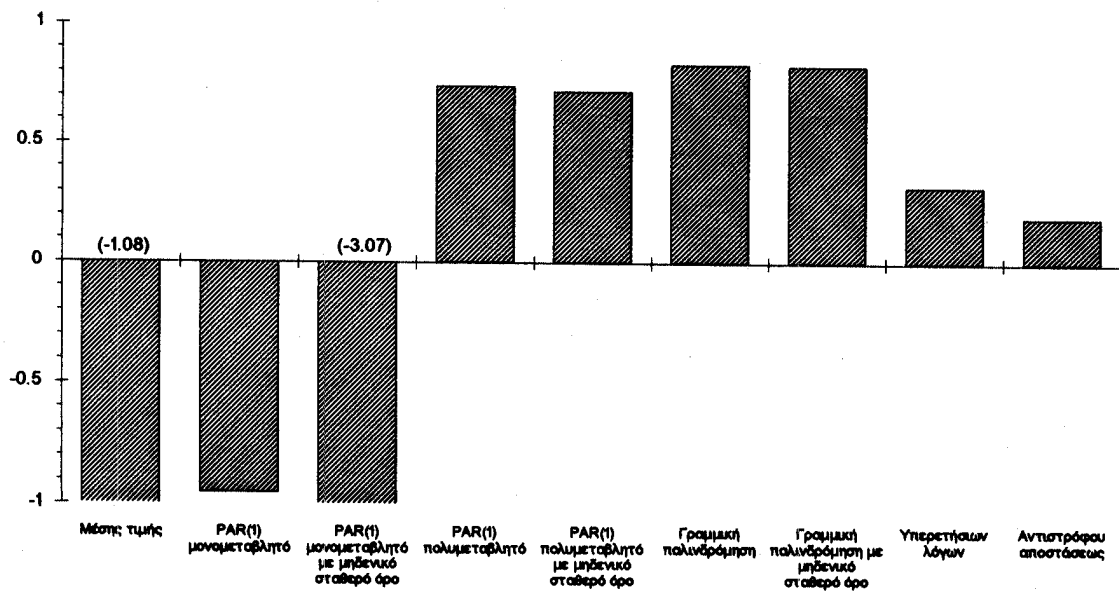
Συντελεστές απόδοσης Ιανουαρίου - μηνιαίες βροχοπτώσεις



Συντελεστές απόδοσης Φεβρουαρίου - μηνιαίες βροχοπτώσεις



Συντελεστές απόδοσης Μαρτίου - μηνιαίες βροχοπτώσεις



Συντελεστές απόδοσης Απριλίου - μηνιαίες βροχοπτώσεις

ΠΑΡΑΡΤΗΜΑ Β: ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΦΑΡΜΟΓΗΣ

ΣΥΜΠΛΗΡΩΣΗ ΕΛΛΕΙΠΟΥΣΩΝ ΤΙΜΩΝ
12 Φεβρουαρίου 1994, 15:53

ΑΠΟΤΕΛΕΣΜΑΤΑ

ΕΛΛΙΠΗΣ ΧΡΟΝΟΣΕΙΡΑ: ΕΠΙΝΙΑΝΑ ΔΕΗ ΒΡΧΜΤΡ 1
ΜΕΣΗ ΤΙΜΗ: 4.9
ΤΥΠΙΚΗ ΑΠΟΚΛΙΣΗ: 13.5

ΜΕΘΟΔΟΣ ΣΥΜΠΛΗΡΩΣΗΣ: PAR(1) με γραμμική παλινδρόμηση, χωρίς τυχαίο όρο, με μηδενικό σταθερό όρο

ΑΡΙΘΜΟΣ ΧΡΟΝΟΣΕΙΡΩΝ ΑΝΑΦΟΡΑΣ: 2

- 1) ΜΟΝΑΣΤΗΡΑΚΙ ΔΕΗ ΒΡΧΜΤΡ 1
- 2) ΤΡΟΒΑΤΟ ΔΕΗ ΒΡΧΜΤΡ 1

Η ελλειπής χρονοσειρά αναφέρεται ως χρονοσειρά 0.
Η ελλειπής με υστέρηση 1 αναφέρεται ως χρονοσειρά 3.

ΚΟΙΝΕΣ ΜΕΤΡΗΣΕΙΣ ΕΤΙΣ ΧΡΟΝΟΣΕΙΡΕΣ ΑΝΑΦΟΡΑΣ (N1+N2): 11564
ΙΑΝ ΦΕΒΡ ΜΑΡΤ ΑΠΡ ΜΑΪ ΙΟΥΝ ΙΟΥΛ ΑΥΓ ΣΕΠΤ ΟΚΤ ΝΟΕΜ ΔΕΚ
961 904 992 960 992 960 992 992 959 961 930 961

ΚΟΙΝΕΣ ΜΕΤΡΗΣΕΙΣ ΣΕ ΟΛΕΣ ΤΙΣ ΧΡΟΝΟΣΕΙΡΕΣ (N1): 11199
ΙΑΝ ΦΕΒΡ ΜΑΡΤ ΑΠΡ ΜΑΪ ΙΟΥΝ ΙΟΥΛ ΑΥΓ ΣΕΠΤ ΟΚΤ ΝΟΕΜ ΔΕΚ
930 876 961 930 961 930 961 961 929 930 900 930

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΧΡΟΝΟΣΕΙΡΩΝ ΑΝΑΦΟΡΑΣ

A/A	1	2	3
r (m)	6431	9483	
μ	5.1	4.1	4.8
σ	13.3	12.3	13.4

όπου

r: Η απόσταση του σταθμού αναφοράς από τον ελλειπή.
μ: Η μέση τιμή της χρονοσειράς.
σ: Η τυπική απόκλιση της χρονοσειράς.

ΣΥΝΤΕΛΕΣΤΗΣ ΓΡΑΜΜΙΚΗΣ ΣΥΣΧΕΤΙΣΗΣ ΜΕ ΕΛΛΙΠΗ ΧΡΟΝΟΣΕΙΡΑ:

	1	2	3
ΙΑΝ	0.8548	0.6089	0.4105
ΦΕΒΡ	0.8760	0.6482	0.5387
ΜΑΡΤ	0.8260	0.6991	0.4164
ΑΠΡ	0.8956	0.7540	0.4240
ΜΑΪ	0.8614	0.6741	0.3405
ΙΟΥΝ	0.6222	0.4839	0.1980
ΙΟΥΛ	0.5624	0.4313	0.1152
ΑΥΓ	0.5804	0.4135	0.1905
ΣΕΠΤ	0.7800	0.6231	0.1840
ΟΚΤ	0.9125	0.6943	0.2803
ΝΟΕΜ	0.8946	0.7124	0.4141
ΔΕΚ	0.8390	0.6736	0.3961

ΕΥΝΤΕΛΕΣΤΗΣ ΠΑΛΙΝΔΡΟΜΗΣΗΣ (b):

	1	2	3
ΙΑΝ	0.715	0.103	0.072
ΦΕΒΡ	0.758	0.155	0.103
ΜΑΡΤ	0.565	0.307	0.051
ΑΠΡ	0.607	0.229	-0.007
ΜΑΪ	0.591	0.224	0.007
ΙΟΥΝ	0.387	0.229	0.066
ΙΟΥΛ	0.370	0.317	0.005
ΑΥΓ	0.430	0.218	0.120
ΣΕΠΤ	0.558	0.321	-0.019
ΟΚΤ	0.858	0.111	-0.010
ΝΟΕΜ	0.909	0.176	-0.051
ΔΕΚ	0.774	0.170	0.024

ΜΕΣΗ ΤΙΜΗ ΚΟΙΝΗΣ ΠΕΡΙΟΔΟΥ:

	1	2	3
ΙΑΝ	8.5	6.6	7.9
ΦΕΒΡ	8.4	6.7	8.6
ΜΑΡΤ	5.7	4.3	5.3
ΑΠΡ	5.3	4.5	4.3
ΜΑΪ	3.8	3.0	3.1
ΙΟΥΝ	1.8	1.7	1.6
ΙΟΥΛ	1.4	1.1	1.1
ΑΥΓ	1.5	1.0	0.9
ΣΕΠΤ	2.1	1.7	2.0
ΟΚΤ	4.5	4.6	4.3
ΝΟΕΜ	8.8	6.9	9.1
ΔΕΚ	10.8	9.4	10.8

ΤΥΠΙΚΗ ΑΠΟΚΛΙΣΗ ΚΟΙΝΗΣ ΠΕΡΙΟΔΟΥ ΠΕΡΙ ΤΟ ΜΗΔΕΝ:

	1	2	3
ΙΑΝ	20.3	17.8	19.2
ΦΕΒΡ	18.8	17.5	19.6
ΜΑΡΤ	14.1	10.9	12.9
ΑΠΡ	12.9	10.7	10.6
ΜΑΪ	9.9	8.2	8.1
ΙΟΥΝ	5.8	5.3	4.9
ΙΟΥΛ	5.6	4.0	4.4
ΑΥΓ	6.2	4.3	4.0
ΣΕΠΤ	7.4	5.7	7.5
ΟΚΤ	13.4	15.3	13.4
ΝΟΕΜ	19.8	18.2	22.1
ΔΕΚ	23.1	23.5	24.7

ΕΥΝΔΙΑΣΠΟΡΑ ΜΕ ΕΛΛΙΠΗ ΓΙΑ ΤΗΝ ΚΟΙΝΗ ΠΕΡΙΟΔΟ ΠΕΡΙ ΤΟ ΜΗΔΕΝ:

	1	2	3
ΙΑΝ	331.2	206.3	150.3
ΦΕΒΡ	322.2	221.3	206.0
ΜΑΡΤ	147.3	96.3	68.1
ΑΠΡ	122.7	85.6	48.0
ΜΑΪ	70.1	45.4	22.7
ΙΟΥΝ	16.1	11.5	4.3
ΙΟΥΛ	14.0	7.6	2.2
ΑΥΓ	19.4	9.7	4.1
ΣΕΠΤ	37.9	23.3	9.1
ΟΚΤ	168.3	146.9	51.8
ΝΟΕΜ	389.4	285.7	202.0
ΔΕΚ	485.0	395.3	245.1

ΕΣΤΑΘΕΡΟΙ ΟΡΟΙ ΠΑΛΙΝΔΡΟΜΗΣΗΣ (a)

ΙΑΝ	ΦΕΒΡ	ΜΑΡΤ	ΑΠΡ	ΜΑΪ	ΙΟΥΝ
0.0	0.0	0.0	0.0	0.0	0.0
ΙΟΥΛ	ΑΥΓ	ΣΕΠΤ	ΟΚΤ	ΝΟΕΜ	ΔΕΚ
0.0	0.0	0.0	0.0	0.0	0.0

ΕΥΝΤΕΛΕΣΤΕΣ ΓΡΑΜΜΙΚΗΣ ΕΥΣΧΕΤΙΣΗΣ (R)

ΙΑΝ ΦΕΒΡ ΜΑΡΤ ΑΠΡ ΜΑΪ ΙΟΥΝ
0.8614 0.8872 0.8521 0.9096 0.8794 0.6753

ΙΟΥΛ ΑΥΓ ΣΕΠΤ ΟΚΤ ΝΟΕΜ ΔΕΚ
0.6268 0.6085 0.8142 0.9168 0.9008 0.8473

ΑΠΟΤΕΛΕΣΜΑΤΑ

Μέσα στις παρενθέσεις φαίνονται οι πραγματικές τιμές της ελλιπούς χρονοσειράς.

	1	2	3	0	
01/01/1975	13.0	0.0	7.1	9.8	(13.4)
02/01/1975	0.0	0.0	13.4	1.0	(0.0)
03/01/1975	0.0	0.0	0.0	0.0	(0.0)
04/01/1975	0.0	0.0	0.0	0.0	(0.0)
05/01/1975	0.0	0.0	0.0	0.0	(0.0)
06/01/1975	0.0	0.0	0.0	0.0	(0.0)
07/01/1975	0.0	0.0	0.0	0.0	(3.0)
08/01/1975	1.2	0.0	3.0	1.1	(0.0)
09/01/1975	16.4	8.8	0.0	12.6	(0.0)
10/01/1975	0.0	4.3	0.0	0.4	(0.0)
11/01/1975	0.0	0.0	0.0	0.0	(0.0)
12/01/1975	0.0	0.0	0.0	0.0	(0.0)
13/01/1975	2.6	1.6	0.0	2.0	(0.0)
14/01/1975	3.4	4.4	0.0	2.9	(2.1)
15/01/1975	0.0	0.0	2.1	0.2	(0.0)
16/01/1975	0.0	0.0	0.0	0.0	(0.0)
17/01/1975	0.0	0.0	0.0	0.0	(0.0)
18/01/1975	0.0	0.0	0.0	0.0	(0.0)
19/01/1975	0.0	0.0	0.0	0.0	(0.0)
20/01/1975	0.0	0.0	0.0	0.0	(0.0)
21/01/1975	0.0	0.0	0.0	0.0	(0.0)
22/01/1975	0.0	0.0	0.0	0.0	(0.0)
23/01/1975	0.0	0.0	0.0	0.0	(0.0)
24/01/1975	0.0	0.0	0.0	0.0	(0.0)
25/01/1975	0.0	0.0	0.0	0.0	(0.0)
26/01/1975	0.0	0.0	0.0	0.0	(0.0)
27/01/1975	0.0	0.0	0.0	0.0	(0.0)
28/01/1975	0.0	0.0	0.0	0.0	(0.2)
29/01/1975	13.0	12.6	0.2	10.6	(10.2)
30/01/1975	2.6	0.0	10.2	2.6	(0.4)
31/01/1975	0.0	0.0	0.4	0.0	(0.0)
01/02/1975	0.0	0.0	0.0	0.0	(0.0)
02/02/1975	0.0	0.0	0.0	0.0	(0.0)
03/02/1975	0.0	1.2	0.0	0.2	(0.0)
04/02/1975	2.3	2.9	0.0	2.2	(0.0)
05/02/1975	0.0	2.2	0.0	0.3	(3.1)
06/02/1975	0.0	0.1	3.1	0.3	(8.9)
07/02/1975	0.0	0.0	8.9	0.9	(0.0)
08/02/1975	0.0	0.0	0.0	0.0	(0.0)
09/02/1975	0.0	0.0	0.0	0.0	(0.0)
10/02/1975	0.0	0.0	0.0	0.0	(1.7)
11/02/1975	0.0	0.0	1.7	0.2	(0.0)
12/02/1975	0.0	0.0	0.0	0.0	(0.0)
13/02/1975	7.0	0.0	0.0	5.3	(5.6)
14/02/1975	57.8	30.8	5.6	49.2	(37.7)
15/02/1975	42.3	39.9	37.7	42.1	(24.0)
16/02/1975	0.0	4.5	24.0	3.2	(0.0)
17/02/1975	10.8	17.4	0.0	10.9	(22.4)
18/02/1975	16.6	25.0	22.4	18.8	(7.8)
19/02/1975	0.0	8.4	7.8	2.1	(0.2)
20/02/1975	0.0	0.6	0.2	0.1	(0.0)
21/02/1975	7.6	6.2	0.0	6.7	(13.6)
22/02/1975	3.0	18.3	13.6	6.5	(5.4)
23/02/1975	0.0	3.9	5.4	1.2	(0.0)
24/02/1975	0.0	6.7	0.0	1.0	(0.0)
25/02/1975	0.0	0.0	0.0	0.0	(0.0)

26/02/1975	0.0	40.3	0.0	6.2	(0.2)
27/02/1975	0.0	0.0	0.2	0.0	(8.3)
28/02/1975	0.0	0.0	8.3	0.9	(0.0)
01/03/1975	0.0	0.0	0.0	0.0	(0.0)
02/03/1975	0.0	0.0	0.0	0.0	(0.0)
03/03/1975	0.0	0.0	0.0	0.0	(0.0)
04/03/1975	0.0	0.0	0.0	0.0	(0.0)
05/03/1975	0.0	0.0	0.0	0.0	(0.0)
06/03/1975	0.0	0.0	0.0	0.0	(0.0)
07/03/1975	0.0	0.0	0.0	0.0	(0.0)
08/03/1975	0.0	0.0	0.0	0.0	(0.0)
09/03/1975	1.0	4.6	0.0	2.0	(2.9)
10/03/1975	6.8	5.4	2.9	5.6	(2.5)
11/03/1975	48.6	19.3	2.5	33.5	(18.4)
12/03/1975	0.5	0.0	18.4	1.2	(0.0)
13/03/1975	0.0	0.0	0.0	0.0	(0.0)
14/03/1975	0.0	0.0	0.0	0.0	(0.0)
15/03/1975	7.4	7.5	0.0	6.5	(7.7)
16/03/1975	0.0	5.5	7.7	2.1	(0.0)
17/03/1975	0.0	0.3	0.0	0.1	(1.1)
18/03/1975	4.8	0.0	1.1	2.8	(1.2)
19/03/1975	0.2	0.4	1.2	0.3	(0.0)
20/03/1975	2.4	17.1	0.0	6.6	(3.2)
21/03/1975	14.5	6.5	3.2	10.4	(11.8)
22/03/1975	12.8	10.6	11.8	11.1	(9.7)
23/03/1975	6.1	9.6	9.7	6.9	(7.5)
24/03/1975	11.0	4.1	7.5	7.9	(19.0)
25/03/1975	84.5	24.3	19.0	56.2	(54.1)
26/03/1975	1.0	0.0	54.1	3.3	(5.2)
27/03/1975	0.0	0.0	5.2	0.3	(0.0)
28/03/1975	0.0	0.0	0.0	0.0	(0.0)
29/03/1975	0.1	0.0	0.0	0.1	(0.0)
30/03/1975	0.0	0.0	0.0	0.0	(0.0)
31/03/1975	0.3	0.0	0.0	0.2	(1.1)
01/04/1975	4.6	3.8	1.1	3.7	(11.3)
02/04/1975	0.3	0.0	11.3	0.1	(0.3)
03/04/1975	4.0	0.0	0.3	2.4	(2.5)
04/04/1975	0.5	0.0	2.5	0.3	(0.6)
05/04/1975	0.0	0.0	0.6	0.0	(0.0)
06/04/1975	0.0	0.0	0.0	0.0	(0.0)
07/04/1975	0.0	0.0	0.0	0.0	(0.0)
08/04/1975	0.0	0.0	0.0	0.0	(0.0)
09/04/1975	0.0	0.0	0.0	0.0	(0.0)
10/04/1975	0.0	0.0	0.0	0.0	(0.0)
11/04/1975	0.6	8.4	0.0	2.3	(0.6)
12/04/1975	0.0	0.0	0.6	0.0	(0.0)
13/04/1975	0.0	4.3	0.0	1.0	(0.3)
14/04/1975	0.0	0.4	0.3	0.1	(0.0)
15/04/1975	0.0	0.0	0.0	0.0	(0.0)
16/04/1975	0.0	0.0	0.0	0.0	(0.0)
17/04/1975	0.4	0.0	0.0	0.2	(0.0)
18/04/1975	3.2	3.8	0.0	2.8	(6.1)
19/04/1975	3.8	15.6	6.1	5.8	(3.1)
20/04/1975	0.0	4.1	3.1	0.9	(0.0)
21/04/1975	0.0	0.0	0.0	0.0	(0.0)
22/04/1975	0.0	1.0	0.0	0.2	(0.0)
23/04/1975	0.0	30.2	0.0	6.9	(0.0)
24/04/1975	5.7	5.9	0.0	4.8	(5.1)
25/04/1975	1.2	5.5	5.1	2.0	(6.6)
26/04/1975	6.6	0.7	6.6	4.1	(2.2)
27/04/1975	11.6	11.2	2.2	9.6	(11.0)
28/04/1975	6.5	5.9	11.0	5.2	(0.5)
29/04/1975	0.0	0.0	0.5	0.0	(0.0)
30/04/1975	0.0	0.0	0.0	0.0	(0.0)
01/05/1975	0.6	0.0	0.0	0.4	(0.0)
02/05/1975	4.0	0.0	0.0	2.4	(0.0)
03/05/1975	0.3	15.5	0.0	3.6	(0.3)
04/05/1975	7.2	4.6	0.3	5.3	(6.5)
05/05/1975	0.2	0.0	6.5	0.2	(0.0)
06/05/1975	0.0	0.0	0.0	0.0	(0.0)
07/05/1975	6.2	0.0	0.0	3.7	(5.7)
08/05/1975	1.0	0.0	5.7	0.6	(1.7)

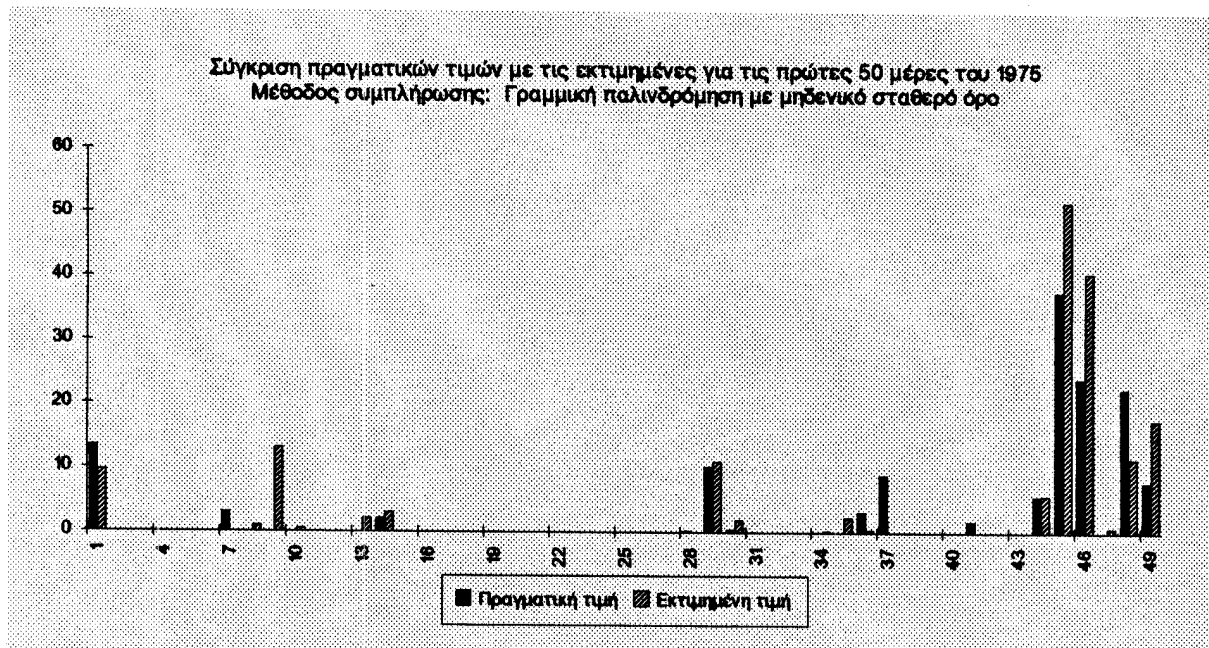
Αποτελέσματα εφαρμογής

09/05/1975	2.9	0.0	1.7	1.7	(2.7)
10/05/1975	0.0	13.1	2.7	3.0	(0.0)
11/05/1975	0.0	0.0	0.0	0.0	(0.0)
12/05/1975	0.0	0.0	0.0	0.0	(0.0)
13/05/1975	2.0	0.0	0.0	1.2	(0.7)
14/05/1975	36.2	0.0	0.7	21.4	(19.3)
15/05/1975	0.0	0.0	19.3	0.1	(0.5)
16/05/1975	0.5	0.0	0.5	0.3	(0.0)
17/05/1975	0.7	0.0	0.0	0.4	(1.5)
18/05/1975	3.3	16.1	1.5	5.6	(7.7)
19/05/1975	0.0	0.0	7.7	0.1	(0.3)
20/05/1975	0.0	0.0	0.3	0.0	(0.0)
21/05/1975	0.0	0.0	0.0	0.0	(0.0)
22/05/1975	2.6	0.0	0.0	1.5	(0.0)
23/05/1975	0.0	0.0	0.0	0.0	(0.0)
24/05/1975	0.0	0.0	0.0	0.0	(0.0)
25/05/1975	0.1	0.0	0.0	0.1	(0.1)
26/05/1975	2.8	0.0	0.1	1.7	(1.5)
27/05/1975	0.0	0.0	1.5	0.0	(0.0)
28/05/1975	0.0	0.0	0.0	0.0	(0.0)
29/05/1975	0.0	0.0	0.0	0.0	(0.0)
30/05/1975	0.0	0.0	0.0	0.0	(0.0)
31/05/1975	0.0	0.0	0.0	0.0	(0.8)
01/06/1975	0.0	0.0	0.8	0.1	(0.0)
02/06/1975	0.0	0.0	0.0	0.0	(0.0)
03/06/1975	0.0	0.0	0.0	0.0	(1.9)
04/06/1975	0.0	0.0	1.9	0.1	(0.0)
05/06/1975	0.0	0.0	0.0	0.0	(0.0)
06/06/1975	0.0	0.0	0.0	0.0	(0.0)
07/06/1975	4.9	19.6	0.0	6.4	(0.0)
08/06/1975	1.8	47.9	0.0	11.6	(0.0)
09/06/1975	0.0	10.6	0.0	2.4	(0.0)
10/06/1975	1.6	0.0	0.0	0.6	(0.0)
11/06/1975	0.0	0.0	0.0	0.0	(0.0)
12/06/1975	0.0	0.0	0.0	0.0	(1.3)
13/06/1975	0.2	0.0	1.3	0.2	(0.0)
14/06/1975	0.0	18.5	0.0	4.2	(29.4)
15/06/1975	0.0	0.0	29.4	1.9	(1.3)
16/06/1975	0.0	0.0	1.3	0.1	(4.6)
17/06/1975	0.0	0.3	4.6	0.4	(0.0)
18/06/1975	0.0	0.0	0.0	0.0	(0.0)
19/06/1975	0.0	0.0	0.0	0.0	(0.0)
20/06/1975	34.8	8.6	0.0	15.4	(2.8)
21/06/1975	18.6	0.0	2.8	7.4	(6.6)
22/06/1975	8.9	0.0	6.6	3.9	(1.3)
23/06/1975	0.0	0.7	1.3	0.2	(0.0)
24/06/1975	0.0	0.0	0.0	0.0	(0.0)
25/06/1975	0.0	0.0	0.0	0.0	(0.0)
26/06/1975	0.0	0.0	0.0	0.0	(0.0)
27/06/1975	0.0	0.0	0.0	0.0	(15.8)
28/06/1975	3.8	0.0	15.8	2.5	(18.0)
29/06/1975	0.8	0.0	18.0	1.5	(5.3)
30/06/1975	0.0	0.0	5.3	0.3	(0.0)
01/07/1975	0.0	0.0	0.0	0.0	(0.0)
02/07/1975	0.0	0.0	0.0	0.0	(0.0)
03/07/1975	5.8	0.0	0.0	2.1	(7.0)
04/07/1975	7.5	0.0	7.0	2.8	(2.8)
05/07/1975	0.0	0.0	2.8	0.0	(0.0)
06/07/1975	0.0	0.0	0.0	0.0	(0.0)
07/07/1975	0.0	0.0	0.0	0.0	(0.0)
08/07/1975	0.0	0.0	0.0	0.0	(0.0)
09/07/1975	0.0	0.0	0.0	0.0	(0.0)
10/07/1975	0.0	0.0	0.0	0.0	(0.0)
11/07/1975	0.0	0.0	0.0	0.0	(0.0)
12/07/1975	0.0	0.0	0.0	0.0	(0.0)
13/07/1975	3.2	0.0	0.0	1.2	(0.2)
14/07/1975	22.0	0.0	0.2	8.1	(23.2)
15/07/1975	0.0	0.0	23.2	0.1	(0.0)
16/07/1975	0.0	0.0	0.0	0.0	(0.0)
17/07/1975	0.0	0.0	0.0	0.0	(0.0)
18/07/1975	0.0	0.0	0.0	0.0	(0.0)
19/07/1975	0.0	0.0	0.0	0.0	(0.0)

20/07/1975	0.0	6.3	0.0	2.0	(0.0)
21/07/1975	5.6	3.4	0.0	3.2	(3.5)
22/07/1975	0.0	0.0	3.5	0.0	(0.0)
23/07/1975	0.0	0.0	0.0	0.0	(0.0)
24/07/1975	0.6	0.0	0.0	0.2	(0.0)
25/07/1975	5.0	0.0	0.0	1.9	(11.5)
26/07/1975	0.0	0.0	11.5	0.1	(0.0)
27/07/1975	0.0	0.0	0.0	0.0	(0.0)
28/07/1975	0.0	0.0	0.0	0.0	(0.0)
29/07/1975	0.0	0.0	0.0	0.0	(0.0)
30/07/1975	0.4	0.0	0.0	0.1	(0.5)
31/07/1975	54.2	0.0	0.5	20.1	(11.6)
01/08/1975	14.0	20.3	11.6	11.8	(4.3)
02/08/1975	0.4	0.0	4.3	0.7	(0.0)
03/08/1975	0.0	4.0	0.0	0.9	(0.0)
04/08/1975	26.7	11.3	0.0	13.9	(3.4)
05/08/1975	0.0	3.1	3.4	1.1	(0.0)
06/08/1975	7.3	25.8	0.0	8.8	(0.0)
07/08/1975	1.5	9.5	0.0	2.7	(0.0)
08/08/1975	9.1	6.4	0.0	5.3	(0.0)
09/08/1975	0.0	15.9	0.0	3.5	(0.0)
10/08/1975	16.2	0.0	0.0	7.0	(0.0)
11/08/1975	0.0	0.0	0.0	0.0	(0.0)
12/08/1975	0.0	0.0	0.0	0.0	(0.0)
13/08/1975	0.0	2.6	0.0	0.6	(0.0)
14/08/1975	0.0	0.0	0.0	0.0	(4.8)
15/08/1975	0.0	0.0	4.8	0.6	(0.0)
16/08/1975	0.0	0.0	0.0	0.0	(0.0)
17/08/1975	0.0	0.2	0.0	0.0	(0.0)
18/08/1975	0.0	0.0	0.0	0.0	(0.0)
19/08/1975	0.0	0.0	0.0	0.0	(2.9)
20/08/1975	0.0	0.0	2.9	0.3	(0.0)
21/08/1975	1.0	0.0	0.0	0.4	(0.0)
22/08/1975	0.0	0.0	0.0	0.0	(0.0)
23/08/1975	0.0	3.5	0.0	0.8	(15.6)
24/08/1975	0.0	0.0	15.6	1.9	(0.0)
25/08/1975	0.0	0.0	0.0	0.0	(0.0)
26/08/1975	0.8	0.0	0.0	0.3	(0.0)
27/08/1975	0.0	0.0	0.0	0.0	(0.0)
28/08/1975	0.0	0.0	0.0	0.0	(0.0)
29/08/1975	0.0	0.0	0.0	0.0	(0.0)
30/08/1975	0.0	0.0	0.0	0.0	(0.0)
31/08/1975	0.0	0.0	0.0	0.0	(0.0)
01/09/1975	0.0	0.0	0.0	0.0	(0.0)
02/09/1975	0.0	3.0	0.0	1.0	(0.0)
03/09/1975	1.3	2.4	0.0	1.5	(0.0)
04/09/1975	0.0	0.0	0.0	0.0	(31.9)
05/09/1975	0.0	0.0	31.9	0.0	(0.0)
06/09/1975	0.0	0.0	0.0	0.0	(0.0)
07/09/1975	0.0	0.0	0.0	0.0	(0.0)
08/09/1975	0.0	1.5	0.0	0.5	(0.0)
09/09/1975	0.0	0.0	0.0	0.0	(0.0)
10/09/1975	0.0	0.0	0.0	0.0	(0.0)
11/09/1975	0.0	0.0	0.0	0.0	(0.0)
12/09/1975	0.9	0.0	0.0	0.5	(3.0)
13/09/1975	2.0	1.3	3.0	1.5	(5.0)
14/09/1975	6.5	4.3	5.0	4.9	(1.7)
15/09/1975	0.0	0.0	1.7	0.0	(0.0)
16/09/1975	0.0	0.0	0.0	0.0	(0.0)
17/09/1975	0.0	0.0	0.0	0.0	(0.0)
18/09/1975	0.4	0.0	0.0	0.2	(3.2)
19/09/1975	0.0	0.0	3.2	0.0	(0.0)
20/09/1975	0.0	0.0	0.0	0.0	(0.0)
21/09/1975	0.0	0.0	0.0	0.0	(0.0)
22/09/1975	0.0	0.0	0.0	0.0	(0.0)
23/09/1975	0.0	0.0	0.0	0.0	(0.0)
24/09/1975	0.0	0.0	0.0	0.0	(0.0)
25/09/1975	0.0	0.0	0.0	0.0	(0.0)
26/09/1975	0.0	0.0	0.0	0.0	(0.0)
27/09/1975	0.0	0.0	0.0	0.0	(0.0)
28/09/1975	0.0	0.0	0.0	0.0	(0.0)
29/09/1975	0.0	0.0	0.0	0.0	(0.0)

30/09/1975	0.0	0.0	0.0	0.0	(0.0)
01/10/1975	0.0	0.0	0.0	0.0	(0.0)
02/10/1975	0.0	0.0	0.0	0.0	(0.0)
03/10/1975	0.0	11.7	0.0	1.3	(0.0)
04/10/1975	0.0	2.4	0.0	0.3	(0.0)
05/10/1975	2.0	0.0	0.0	1.7	(0.1)
06/10/1975	15.2	0.0	0.1	13.0	(11.8)
07/10/1975	0.0	0.0	11.8	0.0	(0.0)
08/10/1975	0.0	0.0	0.0	0.0	(0.0)
09/10/1975	0.0	10.0	0.0	1.1	(0.0)
10/10/1975	21.6	13.2	0.0	20.0	(26.1)
11/10/1975	6.2	9.2	26.1	6.1	(2.4)
12/10/1975	41.0	0.0	2.4	35.2	(5.1)
13/10/1975	0.0	5.1	5.1	0.5	(0.0)
14/10/1975	6.1	0.0	0.0	5.2	(4.5)
15/10/1975	0.0	0.0	4.5	0.0	(0.1)
16/10/1975	0.0	85.1	0.1	9.4	(0.0)
17/10/1975	11.6	45.4	0.0	15.0	(9.9)
18/10/1975	29.3	25.6	9.9	27.9	(39.4)
19/10/1975	46.4	11.9	39.4	40.7	(38.1)
20/10/1975	30.1	0.0	38.1	25.4	(22.0)
21/10/1975	20.8	16.7	22.0	19.5	(16.8)
22/10/1975	0.0	0.0	16.8	0.0	(0.4)
23/10/1975	0.0	4.5	0.4	0.5	(1.6)
24/10/1975	0.0	1.7	1.6	0.2	(0.2)
25/10/1975	0.0	0.0	0.2	0.0	(0.0)
26/10/1975	0.0	0.0	0.0	0.0	(0.0)
27/10/1975	0.0	0.0	0.0	0.0	(0.0)
28/10/1975	0.0	0.0	0.0	0.0	(0.0)
29/10/1975	0.0	0.0	0.0	0.0	(0.0)
30/10/1975	0.0	0.0	0.0	0.0	(0.0)
31/10/1975	0.0	0.0	0.0	0.0	(0.0)
01/11/1975	0.0	0.0	0.0	0.0	(0.0)
02/11/1975	0.0	0.0	0.0	0.0	(0.0)
03/11/1975	0.0	0.0	0.0	0.0	(0.0)
04/11/1975	0.0	0.0	0.0	0.0	(0.0)
05/11/1975	0.0	0.0	0.0	0.0	(0.0)
06/11/1975	0.0	0.9	0.0	0.2	(0.0)
07/11/1975	0.0	0.3	0.0	0.1	(0.0)
08/11/1975	1.0	0.3	0.0	1.0	(0.0)
09/11/1975	13.6	15.1	0.0	15.0	(12.8)
10/11/1975	3.6	0.0	12.8	2.6	(2.3)
11/11/1975	12.3	0.0	2.3	11.1	(21.3)
12/11/1975	7.3	0.0	21.3	5.6	(1.6)
13/11/1975	0.0	0.0	1.6	0.0	(0.0)
14/11/1975	0.0	0.0	0.0	0.0	(0.1)
15/11/1975	21.6	0.0	0.1	19.6	(17.7)
16/11/1975	7.0	0.2	17.7	5.5	(5.0)
17/11/1975	12.0	15.2	5.0	13.3	(10.4)
18/11/1975	15.0	0.5	10.4	13.2	(13.9)
19/11/1975	0.0	0.0	13.9	0.0	(0.0)
20/11/1975	34.2	46.3	0.0	39.2	(30.0)
21/11/1975	0.0	0.0	30.0	0.0	(0.1)
22/11/1975	2.5	35.2	0.1	8.5	(2.8)
23/11/1975	34.4	7.6	2.8	32.5	(27.2)
24/11/1975	7.7	9.7	27.2	7.3	(5.7)
25/11/1975	8.2	0.0	5.7	7.2	(1.7)
26/11/1975	0.0	0.0	1.7	0.0	(4.6)
27/11/1975	0.0	0.0	4.6	0.0	(0.0)
28/11/1975	0.2	0.0	0.0	0.2	(0.0)
29/11/1975	50.4	33.0	0.0	51.6	(4.1)
30/11/1975	0.0	0.2	4.1	0.0	(0.4)
01/12/1975	0.0	0.0	0.4	0.0	(0.0)
02/12/1975	0.0	0.0	0.0	0.0	(0.0)
03/12/1975	2.4	6.5	0.0	3.0	(3.3)
04/12/1975	0.0	3.5	3.3	0.7	(0.6)
05/12/1975	0.0	0.0	0.6	0.0	(0.0)
06/12/1975	0.0	0.0	0.0	0.0	(0.0)
07/12/1975	0.0	0.0	0.0	0.0	(0.0)
08/12/1975	1.9	4.3	0.0	2.2	(6.9)
09/12/1975	0.0	0.0	6.9	0.2	(5.4)
10/12/1975	0.0	0.0	5.4	0.1	(0.0)

11/12/1975	0.0	0.0	0.0	0.0	(0.0)
12/12/1975	0.0	7.3	0.0	1.2	(0.0)
13/12/1975	32.2	0.6	0.0	25.0	(39.6)
14/12/1975	10.7	0.0	39.6	9.2	(5.5)
15/12/1975	0.9	0.0	5.5	0.8	(0.5)
16/12/1975	0.0	0.0	0.5	0.0	(0.0)
17/12/1975	0.0	0.0	0.0	0.0	(0.0)
18/12/1975	43.8	124.5	0.0	55.1	(57.8)
19/12/1975	101.1	54.6	57.8	89.0	(72.2)
20/12/1975	1.4	5.5	72.2	3.8	(5.6)
21/12/1975	0.0	0.0	5.6	0.1	(0.7)
22/12/1975	0.0	0.0	0.7	0.0	(0.5)
23/12/1975	0.0	0.0	0.5	0.0	(0.0)
24/12/1975	0.0	0.0	0.0	0.0	(0.0)
25/12/1975	0.0	0.0	0.0	0.0	(0.0)
26/12/1975	0.0	0.0	0.0	0.0	(0.0)
27/12/1975	0.0	0.0	0.0	0.0	(0.0)
28/12/1975	0.0	0.0	0.0	0.0	(0.0)
29/12/1975	0.0	0.0	0.0	0.0	(0.0)
30/12/1975	0.0	0.0	0.0	0.0	(0.0)
31/12/1975	0.0	0.0	0.0	0.0	(0.0)



ΣΥΜΠΛΗΡΩΣΗ ΕΛΛΕΙΠΟΥΣΩΝ ΤΙΜΩΝ
18 Φεβρουαρίου 1994, 15:48

ΑΠΟΤΕΛΕΣΜΑΤΑ

ΕΛΛΙΠΗΣ ΧΡΟΝΟΣΕΙΡΑ: ΕΠΙΝΙΑΝΑ ΔΕΗ ΒΡΧΜΤΡ 1
ΜΕΣΗ ΤΙΜΗ: 148.4
ΤΥΠΙΚΗ ΑΠΟΚΛΙΣΗ: 150.1

ΜΕΘΟΔΟΣ ΣΥΜΠΛΗΡΩΣΗΣ: PAR(1) με γραμμική παλινδρόμηση, χωρίς τυχαίο όρο

ΑΡΙΘΜΟΣ ΧΡΟΝΟΣΕΙΡΩΝ ΑΝΑΦΟΡΑΣ: 2

- 1) ΜΟΝΑΣΤΗΡΑΚΙ ΔΕΗ ΒΡΧΜΤΡ 1
- 2) ΤΡΟΒΑΤΟ ΔΕΗ ΒΡΧΜΤΡ 1

Η ελλιπής χρονοσειρά αναφέρεται ως χρονοσειρά 0.
Η ελλιπής με υστέρηση 1 αναφέρεται ως χρονοσειρά 3.

ΚΟΙΝΕΣ ΜΕΤΡΗΣΕΙΣ ΣΤΙΣ ΧΡΟΝΟΣΕΙΡΕΣ ΑΝΑΦΟΡΑΣ (N1+N2): 371
ΙΑΝ ΦΕΒΡ ΜΑΡΤ ΑΠΡ ΜΑΪ ΙΟΥΝ ΙΟΥΛ ΑΥΓ ΣΕΠΤ ΟΚΤ ΝΟΕΜ ΔΕΚ
30 31 31 31 31 31 31 31 31 31 31 31

ΚΟΙΝΕΣ ΜΕΤΡΗΣΕΙΣ ΣΕ ΟΛΕΣ ΤΙΣ ΧΡΟΝΟΣΕΙΡΕΣ (N1): 359
ΙΑΝ ΦΕΒΡ ΜΑΡΤ ΑΠΡ ΜΑΪ ΙΟΥΝ ΙΟΥΛ ΑΥΓ ΣΕΠΤ ΟΚΤ ΝΟΕΜ ΔΕΚ
29 30 30 30 30 30 30 30 30 30 30 30

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΧΡΟΝΟΣΕΙΡΩΝ ΑΝΑΦΟΡΑΣ

A/A	1	2	3
r (m)	6431	9483	
μ	156.0	128.7	146.8
σ	142.1	131.9	148.4

όπου

r: Η απόσταση του σταθμού αναφοράς από τον ελλιπή.
μ: Η μέση τιμή της χρονοσειράς.
σ: Η τυπική απόκλιση της χρονοσειράς.

ΣΥΝΤΕΛΕΣΤΗΣ ΓΡΑΜΜΙΚΗΣ ΣΥΣΧΕΤΙΣΗΣ ΜΕ ΕΛΛΙΠΗ ΧΡΟΝΟΣΕΙΡΑ:

	1	2	3
ΙΑΝ	0.8974	0.5673	0.0127
ΦΕΒΡ	0.9110	0.3354	0.3204
ΜΑΡΤ	0.9264	0.5985	0.0872
ΑΠΡ	0.9215	0.8436	0.2602
ΜΑΪ	0.7900	0.5324	0.4073
ΙΟΥΝ	0.7553	0.7469	0.0837
ΙΟΥΛ	0.8041	0.5325	0.0192
ΑΥΓ	0.4672	0.5557	0.2002
ΣΕΠΤ	0.8974	0.7184	0.0831
ΟΚΤ	0.9484	0.7819	0.0315
ΝΟΕΜ	0.9500	0.5731	0.0520
ΔΕΚ	0.9008	0.6593	0.0146

ΣΥΝΤΕΛΕΣΤΗΣ ΠΑΛΙΝΔΡΟΜΗΣΗΣ (b):

	1	2	3
ΙΑΝ	0.894	0.019	0.049
ΦΕΒΡ	1.047	-0.075	0.093
ΜΑΡΤ	0.752	0.072	0.018
ΑΠΡ	0.570	0.153	-0.018
ΜΑΪ	0.561	0.109	0.111
ΙΟΥΝ	0.349	0.283	-0.155
ΙΟΥΛ	0.520	0.229	-0.245
ΑΥΓ	0.238	0.470	0.179

ΣΕΠΤ	0.730	0.106	-0.073
ΟΚΤ	0.932	0.024	-0.040
ΝΟΕΜ	1.150	0.010	-0.206
ΔΕΚ	0.945	0.089	0.161

ΜΕΣΗ ΤΙΜΗ ΚΟΙΝΗΣ ΠΕΡΙΟΔΟΥ:

	1	2	3
ΙΑΝ	270.1	212.5	332.4
ΦΕΒΡ	237.6	194.4	259.5
ΜΑΡΤ	181.5	138.0	242.1
ΑΠΡ	159.9	140.8	165.4
ΜΑΪ	112.3	96.6	129.4
ΙΟΥΝ	54.5	54.1	91.6
ΙΟΥΛ	44.6	35.7	44.1
ΑΥΓ	44.8	31.7	33.8
ΣΕΠΤ	64.3	51.5	31.7
ΟΚΤ	138.7	144.1	58.6
ΝΟΕΜ	263.1	207.2	139.6
ΔΕΚ	334.7	292.2	275.0

ΤΥΠΙΚΗ ΑΠΟΚΛΙΣΗ ΚΟΙΝΗΣ ΠΕΡΙΟΔΟΥ:

	1	2	3
ΙΑΝ	177.6	166.6	194.3
ΦΕΒΡ	134.9	117.4	178.5
ΜΑΡΤ	121.5	85.7	155.2
ΑΠΡ	91.9	85.7	102.9
ΜΑΪ	46.5	67.2	69.5
ΙΟΥΝ	39.8	48.7	40.7
ΙΟΥΛ	43.4	38.4	28.9
ΑΥΓ	28.4	30.6	32.8
ΣΕΠΤ	52.7	43.0	33.4
ΟΚΤ	87.7	135.2	46.8
ΝΟΕΜ	139.2	149.6	89.0
ΔΕΚ	180.7	183.2	168.2

ΣΥΝΔΙΑΣΠΟΡΑ ΜΕ ΕΛΛΙΠΗ ΓΙΑ ΤΗΝ ΚΟΙΝΗ ΠΕΡΙΟΔΟ:

	1	2	3
ΙΑΝ	28467.1	16882.9	440.5
ΦΕΒΡ	19080.0	6111.3	8875.2
ΜΑΡΤ	11582.4	5279.4	1392.4
ΑΠΡ	5887.8	5028.7	-1860.9
ΜΑΪ	1497.1	1456.9	1153.2
ΙΟΥΝ	868.9	1051.9	-98.5
ΙΟΥΛ	1145.5	671.0	-18.2
ΑΥΓ	443.4	568.2	219.2
ΣΕΠΤ	2211.5	1446.5	-129.8
ΟΚΤ	7404.1	9408.4	-131.1
ΝΟΕΜ	22240.2	14421.0	-778.0
ΔΕΚ	32193.6	23889.4	486.6

ΣΤΑΘΕΡΟΙ ΟΡΟΙ ΠΑΛΙΝΔΡΟΜΗΣΗΣ (a)

ΙΑΝ	ΦΕΒΡ	ΜΑΡΤ	ΑΠΡ	ΜΑΪ	ΙΟΥΝ
-8.2	-16.2	14.9	19.7	3.7	23.9
ΙΟΥΛ	ΑΥΓ	ΣΕΠΤ	ΟΚΤ	ΝΟΕΜ	ΔΕΚ
13.2	0.0	8.5	9.1	-1.0	-50.9

ΣΥΝΤΕΛΕΣΤΕΣ ΓΡΑΜΜΙΚΗΣ ΣΥΣΧΕΤΙΣΗΣ (R)

ΙΑΝ	ΦΕΒΡ	ΜΑΡΤ	ΑΠΡ	ΜΑΪ	ΙΟΥΝ
0.8989	0.9187	0.9277	0.9278	0.8245	0.8592
ΙΟΥΛ	ΑΥΓ	ΣΕΠΤ	ΟΚΤ	ΝΟΕΜ	ΔΕΚ
0.8368	0.6081	0.9007	0.9489	0.9565	0.9131

ΑΠΟΤΕΛΕΣΜΑΤΑ

Μέσα στις παρενθέσεις φαίνονται οι πραγματικές τιμές της ελλιπούς χρονοσειράς.

	1	2	3	0	
01/1975	52.2	31.7	91.7	43.6	(29.3)
02/1975	147.4	208.4	29.3	125.2	(138.9)
03/1975	202.0	115.2	138.9	177.4	(145.4)
04/1975	49.0	100.8	145.4	60.4	(50.2)
05/1975	70.6	49.3	50.2	54.3	(49.3)
06/1975	75.4	106.2	49.3	72.7	(88.3)
07/1975	104.3	9.7	88.3	48.0	(60.3)
08/1975	77.0	102.6	60.3	77.4	(31.0)
09/1975	11.1	12.5	31.0	15.6	(44.8)
10/1975	230.3	242.5	44.8	227.9	(178.5)
11/1975	231.0	164.5	178.5	229.6	(161.7)
12/1975	194.4	206.8	161.7	177.1	(198.6)

ΠΑΡΑΡΤΗΜΑ Γ: ΔΕΙΓΜΑ ΚΩΔΙΚΑ ΕΦΑΡΜΟΓΗΣ

Ο κώδικας της εφαρμογής δεν έχει νόημα να παρατεθεί ολόκληρος, γιατί δεν είναι ένα αυτόνομο πρόγραμμα, αλλά τρέχει μόνο σε συνεργασία με τις υπόλοιπες εφαρμογές του Υδροσκοπίου. Γι' αυτό παρατίθεται μόνο ενδεικτικά ένα μέρος του κώδικα. Αρχικά υπάρχουν μερικές διαδικασίες σε INGRES/Windows4GL, και παρακάτω μερικές άλλες σε C.

Ο σχολιασμός του κώδικα έχει γίνει στην αγγλική γλώσσα, σύμφωνα με τις προδιαγραφές για την ανάπτυξη λογισμικού του Υδροσκοπίου (Παπακώστας [1992]).

```

/*****
HYDROSCOPE:  CREATION OF A NATIONAL DATABANK FOR
              HYDROLOGICAL AND METEOROLOGICAL INFORMATION
*****/

/*****

** FRAME

infilling - Starting frame of infilling application
           Perform infilling of missing values of timeseries.

** AUTHOR

Anthony Christofides, NTUA

** DATE

Last modified on 30 January 1994.  Programming began on 14 April 1993.

** PURPOSE

For a specific monthly or yearly time series, the missing values are estimated
by one of a number of methods.  All Methods except for the 'Mean Value' method
use a number of stations, called 'index stations', near the incomplete station,
in order to complete the time series for the incomplete station.

** ARGUMENTS AND RETURN VALUE

Call infilling in the following format:

OPEN|CALL|GOTOFRAME infilling ();

** USER CLASSES

'InfTimeSeries'

This class holds all the information required about an incomplete timeseries or
about the equivalent timeseries of an index station.  The attributes are as
follows:

instrument (integer, nullable)  The instrument code.
TableName (vc41)                The table name where the data are.
time_res_t (varchar7)           MONTHLY or YEARLY.
hydro_year (c1)                 Flag that shows if hydro year or normal year is used.
                                Valid only for time_res_t = YEARLY.
instrument_t (vc8)              The instrument type.
linear (int)                    0 for normal storing, 1 for linear, 2 for differential.

```


StartDate (vc25) The start date of the timeseries.
 EndDate (vc25) The end date of the timeseries.
 station (integer) The code of the station.
 station_name (vc30) The name of the station.
 If the station does not have a name, station_name = VC_UNNAMED.
 service (vc8) The service of the station.
 num1 (smallint) Same as the num1 column of the 'instruments' view of the database.
 x (float) The x of the station.
 y (float) The y of the station.
 distance (float) The distance in metres of the station with the incomplete time series from the station with the current time series.
 completeness (integer) This is a pointer to a C array of short. See function InfilGetCompleteness for an explanation.
 contents (integer) This is a pointer to a C array holding contents of the timeseries. See the file 'infill.sc' for more information.
 a (float) The weighting coefficient in the RD and NR methods.
 average (float) Average value, excluding filled values.
 s (float) The standard deviation, excluding filled values.
 b (float) The linear regression coefficient.
 Additional information about the timeseries is held in InfillingInfo.TimeseriesPtr.

Note: If the 'instrument' and 'time_res_t' are given, all the other attributes can be found. There is a 4GL procedure, 'SetInftsData', that does this little job. A variable of type 'InfTimeSeries' has 'instrument IS NULL' if it does not hold a timeseries. However, even if instrument IS NULL, the timeseries can hold an instrument_t and/or time_res_t. This is useful for the incomplete time series.

'InfillingInfoClass'

This class contains all the data required to define an infilling procedure, plus the results. Its attributes are:

IncTimeSeries (InfTimeSeries) The incomplete time series.
 ValuesToFill (array of StringObject) The dates of the missing values to be filled in.
 virtual (int not null) If this is TRUE, it indicates that the results will not be written to the database.
 existing (int not null) If in one ore more ValuesToFill there is already an existing value, then this attribute takes the value TRUE.
 OldValues (array of NullableIntegerObject) OldValues holds the existing values in the respective dates of ValuesToFill, or NULL if the respective date is a missing value.
 OldStatus (array of IntegerObject) OldStatus holds the existing statuses in the respective dates of ValuesToFill (dummy values if missing value).
 method (i2) A number that defines the infilling method.
 power (float) Applicable only if infilling method is reciprocal distance. It is the power to which the reciprocal distance will be raised.
 mwa (int2) Applicable only if supported by the particular infilling method. It is TRUE if a variation of the method, called 'Modified Weighted Average', is to be used, else it is FALSE.
 WithNoise (int2) Applicable only for linear regression method. It is TRUE if noise is contained.
 WithZeroConstant (int2) Applicable only for linear regression method. It is TRUE if the constant term must be zero.
 IndexTimeseries (ARRAY OF InfTimeSeries) This holds the index for the particular time interval. Not applicable for mean value method.
 UsedTimeseries (ARRAY OF InfTimeSeries) When the data of a timeseries are read from the database in order to make an infilling, it is highly probable that the user will want to make another infilling that uses the same data. Thus, all the data

read is kept in memory, in UsedTimeseries, until the application exits, in order to avoid reading data multiple times from the database. This is more important when the data is on a remote node. All timeseries that have been used go into UsedTimeseries.

N1, N2 (integer) The number of observations for the index and common period of measurements.

Values1 (integer) Pointer to a C array of char[20] holding the dates of period 1.

Values2 (integer) The dates of period 2.

TimeseriesPtr (integer) This is a pointer to a C array holding information about all timeseries. If m is the number of index timeseries, then the array has m+1 elements (0 for incomplete, 1 through m for index), unless the method used is IM_AR1_WITH_LR. In that case, there one more timeseries (number m+1): the lag-1 incomplete.

a (ARRAY OF FloatObject) Contains one or 12 values for the a parameter of linear regression, depending on whether the timeseries is annual or of other time step.

alpha (Float) Noise coefficient for linear regression method.

R (ARRAY OF FloatObject) Linear regression coefficients (1 or 12, depending on time_res_t).

s2 (integer) Pointer to a C array holding 3-dimensional array of covariances, or, in other words, 12 2-dimensional arrays of covariances, one for each month. If the timeseries is annual, only one 2-dimensional array is used. For more information, see InfilCalculateCovariances procedure.

InfStrategyWarning

This class holds information about warnings on the infilling strategy. There is an array of InfStrategyWarning, StrategyWarnings, that holds the information for all the warnings. After this is filled after a series of checks, a frame displays it on the screen. The attributes are as follows:

indx (integer) If it is 0, the warning concerns the incomplete time series. If non-zero, it concerns InfillingInfo.IndexTimeseries[indx].

WarningType (integer) The warning type.

WarningLocation (vc19) The date for which the warning appears.

WarningType can contain one of the following values:

ISW_TC_OK Appears only for indx=0 and WarningLocation within interval to fill, and specifies that in WarningLocation there is already a value of type TC_OK.

ISW_TC_FILLED WarningLocation contains a value which is filled.

ISW_TC_GAP (indx!=0 only) WarningLocation contains a gap.

ISW_TC_NOHOMOCHK WarningLocation contains a value with TC_NOHOMOCHK.

ISW_TC_HOMOGENIZED WarningLocation contains a value with TC_HOMOGENIZED.

ISW_TC_LEVEL2 WarningLocation contains a value which is TC_LEVEL2.

ISW_TC_LEVEL1 WarningLocation contains a value which is TC_LEVEL1.

** CONSTANTS

UNNAMED

This holds the string 'Unnamed'.

TRT_MONTHLY, TRT_ANNUAL, TRT_DAILY

These are used as values for the InfTimeSeries attribute 'time_res_t'. They contain the string used to identify a monthly, annual, or daily time series.

IM_MEANVALUE, IM_RECIPROCALDISTANCE, IM_NORMALRATIO, IM_LINEARREGRESSION, IM_AR1_WITH_LR

Each one is an integer representing an infilling method. The values defined for these constants must match the values defined for the methods in the 'method' option field of the DefineStrategy frame.

IMS_MEANVALUE, IMS_RECIPROCALDISTANCE, IMS_NORMALRATIO, IM_LINEARREGRESSION, IMS_AR1_WITH_LR

These hold the names of the methods.

ISW xxxxxxxxxxxxxxxx

The constants that begin with ISW are described above in the description of the InfStrategyWarning user class.

** GLOBAL VARIABLES

InfillingFrame

See the UserEvents 'InternalError' and 'FatalError' below for an explanation.

*****/

```
INITIALIZE (dummy          = varchar (8) NOT NULL,
            i              = integer NOT NULL,
            j              = integer NOT NULL,
            k              = integer NOT NULL,
            m              = integer not null,
            nullable       = integer with null,
            ev             = EventMsgClass,
            database       = varchar(32) NOT NULL,
            ViewCompletenessFrame = FrameExec,
            ViewCompletenessFrames = ARRAY OF FrameExec,
            /* The above two are for controlling the ViewCompleteness
            * children. The first one is for the incomplete and the
            * second for the index timeseries. If NULL, the respective
            * ViewCompleteness frame is not open.
            */
            ViewWarningsFrame = FrameExec,
            GetValuesToFillFrame = FrameExec,
            SelectInstrumentFrame = FrameExec,
            StrategyWarnings = ARRAY OF InfStrategyWarning,
            NextRow         = integer not null,
            IndexBeingChecked = integer not null,
            TimeseriesChecked = integer not null,
            IsRemote        = integer not null,
            ErrorText       = varchar (255) not null,
            ArrayOfStrObj   = ARRAY OF StringObject,
            Prev_time_res_t = varchar (8) not null,
            date            = varchar (19) not null,
            StartDate       = varchar (19) not null,
            EndDate         = varchar (19) not null,
            time_res_t      = varchar (8) not null,
            completeness    = integer not null
) = BEGIN
```

```
ViewCompletenessFrame = NULL;
ViewWarningsFrame = NULL;
GetValuesToFillFrame = NULL;
SelectInstrumentFrame = NULL;
InfillingInfo.IncTimeseries = InfTimeseries.Create();
InfillingFrame = CurFrame;
InfillingInfo.IncTimeseries.instrument = NULL;
InfillingInfo.IncTimeseries.time_res_t = TRT_MONTHLY;
InfillingInfo.power = 2.0;
InfillingInfo.method = IM_LINEARREGRESSION;
InfillingInfo.virtual = FALSE;
InfillingInfo.existing = FALSE;
InfillingInfo.TimeseriesPtr = 0;
InfillingInfo.Values1 = 0;
InfillingInfo.Values2 = 0;
Prev_time_res_t = InfillingInfo.IncTimeseries.time_res_t;
FIELD (PleaseWait).CurBias = FB_INVISIBLE;
CurFrame.SendUserEvent (eventname = 'SetWindowSize');
CurFrame.SendUserEvent (eventname = 'SetBiases');
```

END;

/*-----*/

ON Click menu.timeseries.CloseBtn,
ON Click CancelBtn,

```

ON Terminate,
ON WindowClose = BEGIN
  /* Release C arrays. */
  i = 1;
  WHILE i <= InfillingInfo.UsedTimeseries.AllRows() DO
    IF InfillingInfo.UsedTimeseries[i].completeness != 0 THEN
      CALLPROC InfilReleaseArray
        (InfillingInfo.
          UsedTimeseries[i].completeness);
      InfillingInfo.UsedTimeseries[i].completeness = 0;
    ENDIF;
    IF InfillingInfo.UsedTimeseries[i].contents != 0 THEN
      CALLPROC InfilReleaseArray
        (InfillingInfo.UsedTimeseries[i].contents);
      InfillingInfo.UsedTimeseries[i].contents = 0;
    ENDIF;
    i = i+1;
  ENDWHILE;

  /* Sometimes, notably in case of sql error (or application bug), it
   * is possible that one of the IndexTimeseries and IncTimeseries does
   * not have an equivalent UsedTimeseries. Thus, These are also freed.
   */
  IF InfillingInfo.IncTimeseries.completeness != 0 THEN
    CALLPROC InfilReleaseArray
      (InfillingInfo.IncTimeseries.completeness);
    InfillingInfo.IncTimeseries.completeness = 0;
  ENDIF;
  IF InfillingInfo.IncTimeseries.contents != 0 THEN
    CALLPROC InfilReleaseArray
      (InfillingInfo.IncTimeseries.contents);
    InfillingInfo.IncTimeseries.contents = 0;
  ENDIF;
  i = 1;
  WHILE i <= InfillingInfo.IndexTimeseries.AllRows() DO
    IF InfillingInfo.IndexTimeseries[i].completeness != 0 THEN
      CALLPROC InfilReleaseArray
        (InfillingInfo.
          IndexTimeseries[i].completeness);
      InfillingInfo.IndexTimeseries[i].completeness = 0;
    ENDIF;
    IF InfillingInfo.IndexTimeseries[i].contents != 0 THEN
      CALLPROC InfilReleaseArray
        (InfillingInfo.IndexTimeseries[i].contents);
      InfillingInfo.IndexTimeseries[i].contents = 0;
    ENDIF;
    i = i+1;
  ENDWHILE;

  IF InfillingInfo.Values1 != 0 THEN
    CALLPROC InfilReleaseArray (InfillingInfo.Values1);
  ENDIF;
  IF InfillingInfo.Values2 != 0 THEN
    CALLPROC InfilReleaseArray (InfillingInfo.Values2);
  ENDIF;
  IF InfillingInfo.TimeseriesPtr != 0 THEN
    CALLPROC InfilReleaseArray (InfillingInfo.TimeseriesPtr);
  ENDIF;
  IF InfillingInfo.s2 != 0 THEN
    CALLPROC InfilReleaseArray (infillingInfo.s2);
  ENDIF;

  InfillingInfo = NULL;
  InfillingFrame = NULL;
  RETURN;
END;

/*-----*/
ON Click menu.help.AboutBtn = BEGIN
  CALLFRAME about (item = INF_ITEM, author=INF_AUTHOR,
    version=INF_VERSION,

```

```

        service = INF_SERVICE, email = NULL, item2=INF_ITEM2);
/* Note: There is also another call to 'about' in the UserEvent
 * 'InternalError'.
 */
END;

/*-----*/
ON Click menu.timeseries.IncStationBtn = BEGIN
    nullable=CALLFRAME SelectInstrument() WITH WindowTitle=VC_INCTIMESERIES;
    i = IFNULL (nullable, -1);
    IF i < 0 OR i = InfillingInfo.IncTimeseries.instrument THEN
        RESUME;
    ENDIF;

    nullable = InfillingInfo.IncTimeseries.instrument; /* Save for later. */
    InfillingInfo.IncTimeseries.instrument = i;
    /* Search UsedTimeseries in case it has already been used. */
    i = 1;
    WHILE i <= InfillingInfo.UsedTimeseries.AllRows() DO
        IF InfillingInfo.UsedTimeseries[i].instrument =
            InfillingInfo.IncTimeseries.instrument
            AND InfillingInfo.UsedTimeseries[i].time_res_t
            = InfillingInfo.IncTimeseries.time_res_t THEN
                InfillingInfo.IncTimeseries=
                    InfillingInfo.UsedTimeseries[i];
            ENDOLOOP;
        ENDIF;
        i = i+1;
    ENDWHILE;

    /* If not found in UsedTimeseries, do now whatever is needed. */
    IF i = InfillingInfo.UsedTimeseries.AllRows()+1 THEN
        IF SetInftsData (InfillingInfo=InfillingInfo, i=0,
            ErrorText = BYREF (ErrorText)) < 0 THEN
            IF ErrorText != '' THEN
                CALLFRAME MessagePopUp(messageText = ErrorText);
            ENDIF;
            InfillingInfo.IncTimeseries.instrument = nullable;
            RESUME;
        ENDIF;
        /* The timeseries will be copied to UsedTimeseries when the
        * data are read from the database.
        */
    ENDIF;

    instrument = InfillingInfo.IncTimeseries.instrument_t + ' '
        +varchar (InfillingInfo.IncTimeseries.num1);
    IF ViewCompletenessFrame IS NOT NULL THEN
        ViewCompletenessFrame.Terminate();
    ENDIF;
    InfillingInfo.ValuesToFill.Clear();
    i = 1;
    WHILE i <= InfillingInfo.IndexTimeseries.AllRows() DO
        IF InfillingInfo.IndexTimeseries[i].instrument
            = InfillingInfo.IncTimeseries.instrument THEN
                CurFrame.SendUserEvent (eventName
                    = 'DeleteIndexTimeseries', messageinteger = i);
            ENDOLOOP;
        ENDIF;
        i = i+1;
    ENDWHILE;
    CALLPROC CalculateAllDistances (InfillingInfo=InfillingInfo);
    IF ViewWarningsFrame IS NOT NULL THEN
        ViewWarningsFrame.SendUserEvent (eventName='StrategyChanged');
    ENDIF;

    CurFrame.SendUserEvent (eventName='CheckInstrumentT');
    CurFrame.SendUserEvent (eventName='SetBiases');
END;

/*-----*/

```

```

/* The following event block checks whether the instrument_t of the incomplete
 * timeseries matches that of the index timeseries. For some infilling methods,
 * this is not required.
 */

```

```
ON UserEvent 'CheckInstrumentT' = BEGIN
```

```

    IF InfillingInfo.method = IM_LINEARREGRESSION OR
    InfillingInfo.method = IM_AR1 WITH LR OR
    InfillingInfo.method = IM_MEANVALUE THEN
        RESUME;
    ENDIF;
    i = 1;
    WHILE i <= InfillingInfo.IndexTimeseries.AllRows() DO
        IF InfillingInfo.IndexTimeseries[i].instrument_t
        != InfillingInfo.IncTimeseries.instrument_t THEN
            ENDLOOP;
        ENDIF;
        i = i+1;
    ENDWHILE;
    IF i = InfillingInfo.indexTimeseries.AllRows()+1 THEN
        RESUME;
    ENDIF;
    i = CALLFRAME InstrumentTInconsist (type = 1);
    IF i != 0 THEN
        InfillingInfo.method = i;
    ELSE
        i = 1;
        WHILE i <= InfillingInfo.IndexTimeseries.AllRows() DO
            IF InfillingInfo.IndexTimeseries[i].instrument_t
            != InfillingInfo.IncTimeseries.instrument_t THEN
                CALLPROC RemoveIndexTimeseries
                (InfillingInfo=InfillingInfo, i=i);
            ELSE
                i=i+1;
            ENDIF;
        ENDWHILE;
    ENDIF;
END;

```

```
END;
```

```
/*-----*/
```

```
ON Click menu.timeseries.IncCompletenessBtn = BEGIN
```

```

    IF ViewCompletenessFrame is null THEN
        ViewCompletenessFrame =
            OPENFRAME ViewCompleteness
            (timeseries=InfillingInfo.IncTimeseries);
    ELSE
        ViewCompletenessFrame.BringToFront();
    ENDIF;
END;

```

```
END;
```

```
/*-----*/
```

```
ON SetValue InfillingInfo.IncTimeseries.time_res_t = BEGIN
```

```

    IF SetInfTsData (InfillingInfo = InfillingInfo, i=0,
    ErrorText = byref (ErrorText)) < 0 THEN
        IF ErrorText != '' THEN
            CALLFRAME MessagePopUp (messagetext = ErrorText);
        ENDIF;
        InfillingInfo.IncTimeseries.time_res_t = Prev_time_res_t;
    ELSE
        IF ViewCompletenessFrame IS NOT NULL THEN
            ViewCompletenessFrame.Terminate();
        ENDIF;
        CurFrame.SendUserEvent (eventname = 'ClearIndexTimeseries');
        InfillingInfo.ValuesToFill.Clear();
        IF ViewWarningsFrame IS NOT NULL THEN
            ViewWarningsFrame.SendUserEvent
            (eventname='StrategyChanged');
        ENDIF;
    ENDIF;
END;

```

```

        ENDIF;
        CurFrame.SendUserEvent (eventname = 'SetBiases');
        Prev_time_res_t = InfillingInfo.IncTimeseries.time_res_t;
    ENDIF;
END;

/*-----*/
ON Click menu.help.InstructionsBtn = BEGIN
    OPENFRAME HelpTopics (frame = VC_INFILLINGTITLE);
    FIELD (menu.help.InstructionsBtn).CurBias = MB_DISABLED;
END;

/*-----*/
ON UserEvent 'HelpTopicsQuit' = BEGIN
    FIELD (menu.help.InstructionsBtn).CurBias = MB_ENABLED;
END;

/*-----*/
ON UserEvent 'SetBiases' = BEGIN
    IF InfillingInfo.IncTimeSeries.instrument IS NULL THEN
        FIELD (menu.timeseries.IncCompletenessBtn).CurBias = MB_DISABLED;
        FIELD (InfillingInfo.IndexTimeseries).CurBias = FB_DIMMED;
        FIELD (menu.timeseries.AddBtn).CurBias = MB_DISABLED;
        FIELD (menu.timeseries.DeleteSelectedBtn).CurBias
            = MB_DISABLED;
        FIELD (menu.timeseries.DeleteAllBtn).CurBias
            = MB_DISABLED;
        FIELD (menu.timeseries.IndexCompletenessBtn).CurBias
            = MB_DISABLED;
        FIELD (menu.ValuesToFill.AddValuesToFillBtn).curbias=MB_DISABLED;
        FIELD (menu.ValuesToFill.DeleteSelectedValueToFillBtn).CurBias =
            MB_DISABLED;
        FIELD (menu.ValuesToFill.DeleteAllValuesToFillBtn).CurBias =
            MB_DISABLED;
        FIELD (InfillingInfo.ValuesToFill).CurBias = FB_DIMMED;
    ELSE
        FIELD (menu.timeseries.IncCompletenessBtn).CurBias = MB_ENABLED;
        FIELD (InfillingInfo.ValuesToFill).CurBias = FB_CHANGEABLE;
        FIELD (menu.ValuesToFill.AddValuesToFillBtn).CurBias=MB_ENABLED;
        IF InfillingInfo.ValuesToFill.AllRows() != 0 THEN
            FIELD (menu.ValuesToFill.DeleteAllValuesToFillBtn)
                .CurBias = MB_ENABLED;
        ELSE
            FIELD (menu.ValuesToFill.DeleteAllValuesToFillBtn)
                .CurBias = MB_DISABLED;
        ENDIF;
        IF FIELD(InfillingInfo.ValuesToFill).ActiveRow != 0 THEN
            FIELD (menu.ValuesToFill.DeleteSelectedValueToFillBtn)
                .CurBias = MB_ENABLED;
        ELSE
            FIELD (menu.ValuesToFill.DeleteSelectedValueToFillBtn)
                .CurBias = MB_DISABLED;
        ENDIF;
        IF InfillingInfo.method = IM_MEANVALUE THEN
            FIELD (menu.timeseries.AddBtn).CurBias
                = MB_DISABLED;
            FIELD (menu.timeseries.DeleteSelectedBtn)
                .CurBias = MB_DISABLED;
            FIELD (menu.timeseries.DeleteAllBtn).CurBias
                =MB_DISABLED;
            FIELD (menu.timeseries.IndexCompletenessBtn)
                .CurBias = MB_DISABLED;
            FIELD (InfillingInfo.IndexTimeseries).CurBias=FB_DIMMED;
        ELSE
            FIELD (InfillingInfo.IndexTimeseries).CurBias
                =FB_CHANGEABLE;
            FIELD (menu.timeseries.AddBtn).CurBias
                = MB_ENABLED;
            IF InfillingInfo.IndexTimeseries.AllRows() != 0 THEN

```

```

        FIELD (menu.timeseries.DeleteAllBtn)
            .CurBias = MB_ENABLED;
    ELSE
        FIELD (menu.timeseries.DeleteAllBtn)
            .CurBias = MB_DISABLED;
    ENDIF;
    IF FIELD(InfillingInfo.IndexTimeseries).ActiveRow != 0
    THEN
        FIELD(menu.timeseries.
            DeleteSelectedBtn).CurBias = MB_ENABLED;
    ELSE
        FIELD (menu.timeseries
            .DeleteSelectedBtn).CurBias=MB_DISABLED;
    ENDIF;
    ENDIF;
ENDIF;

IF InfillingInfo.method = IM_RECIPROCALDISTANCE
OR InfillingInfo.method = IM_NORMALRATIO THEN
    FIELD (InfillingInfo.mwa).CurBias = FB_CHANGEABLE;
ELSE
    FIELD (InfillingInfo.mwa).CurBias = FB_INVISIBLE;
ENDIF;

IF InfillingInfo.method = IM_RECIPROCALDISTANCE THEN
    FIELD (PowerTxt).CurBias = FB_VISIBLE;
    FIELD (InfillingInfo.power).CurBias = FB_CHANGEABLE;
ELSE
    FIELD (PowerTxt).CurBias = FB_INVISIBLE;
    FIELD (InfillingInfo.power).CurBias = FB_INVISIBLE;
ENDIF;

IF InfillingInfo.method = IM_LINEARREGRESSION
OR InfillingInfo.method = IM_AR1_WITH_LR THEN
    FIELD (InfillingInfo.WithNoise).CurBias = FB_CHANGEABLE;
    FIELD (InfillingInfo.WithZeroConstant).CurBias = FB_CHANGEABLE;
ELSE
    FIELD (InfillingInfo.WithNoise).CurBias = FB_INVISIBLE;
    FIELD (InfillingInfo.WithZeroConstant).CurBias = FB_INVISIBLE;
ENDIF;

IF FIELD (InfillingInfo.IndexTimeseries).ActiveRow = 0 THEN
    FIELD(menu.timeseries.IndexCompletenessBtn).CurBias=MB_DISABLED;
ELSE
    FIELD(menu.timeseries.IndexCompletenessBtn).CurBias=MB_ENABLED;
ENDIF;

/* Set title of AR method. */
i=FIELD(InfillingInfo.method).ValueList.IndexByValue(enumvalue=
    IM_AR1_WITH_LR);
IF InfillingInfo.IncTimeseries.time_res_t = TRT_ANNUAL THEN
    FIELD(InfillingInfo.method).ValueList.ChoiceItems[i].EnumText=
    IMS_AR1_WITH_LR;
ELSE
    FIELD(InfillingInfo.method).ValueList.ChoiceItems[i].EnumText=
    IMS_PAR1_WITH_LR;
ENDIF;
i=InfillingInfo.method;
FIELD(InfillingInfo.method).UpdChoiceList();
InfillingInfo.method=i;

/* Finally, set the OkBtn bias. */
FIELD (OkBtn).CurBias = FB_DIMMED;
IF InfillingInfo.IncTimeseries.instrument IS NULL THEN
    RESUME;
ELSEIF InfillingInfo.ValuesToFill.AllRows()=0 THEN
    RESUME;
ELSEIF InfillingInfo.method = IM_MEANVALUE
OR InfillingInfo.method = IM_AR1_WITH_LR THEN
    FIELD (OkBtn).CurBias = FB_CHANGEABLE;
    RESUME;
ELSEIF InfillingInfo.IndexTimeseries.AllRows() = 0 THEN

```



```

        RESUME;
    ELSE
        FIELD (OkBtn).CurBias = FB_CHANGEABLE;
    ENDIF;
END;

```

Αμέσως παρακάτω είναι μερικές από τις διαδικασίες της εφαρμογής σε γλώσσα C. Μερικές περιέχουν και ενσωματωμένη SQL.

```

/*****
        HYDROSCOPE:  CREATION OF A NATIONAL DATABANK
                    FOR HYDROLOGICAL AND METEOROLOGICAL DATA
*****/

/*****
** FILE:      infill.sc -  procedures concerning 'infilling' application
** APPLICATION:  stations
** LANGUAGE:  C
*****/

#include <stdlib.h>
#include <stdio.h>
#include <limits.h>
#include <string.h>
#include <time.h>
#include <math.h>
#include "hydrolib.h"
#include "DateStrings.h"
#include "infill.h"
exec sql include sqlca;
exec sql include sqlda;

exec sql begin declare section;
typedef struct
{
    int status;
    short value0;
} STATUSANDVALUE;
exec sql end declare section;

/* The following need only be sql declare section for debugging purposes. */
exec sql begin declare section;
typedef struct
{
    char StartDate [20];
    char EndDate [20];
    short *completeness;
    STATUSANDVALUE *contents;
    double mean, var;
    double MonthlyMeans[12]; /* Only 1st element valid if TRT_ANNUAL. */
    double b[12];           /* Coefficients of LR. */
} TIMESERIES;
exec sql end declare section;

#define SQLNULL -1

#if defined (DEBUG)
    #define TRACE(p) printf (p); fflush (stdout)
    #define TRACE1(p, n) printf (p, n); fflush (stdout)
    #define TRACE2(p, n1, n2) printf (p, n1, n2); fflush (stdout)
#else
    #define TRACE(p)
    #define TRACE1(p, n)
    #define TRACE2(p, n1, n2)
#endif
#endif

```

```

/*****
** PROCEDURES:      See 'FORMAT' below.
** AUTHOR:         Anthony Christofides, NTUA
** FORMAT
int   InfilCreateSAVArray (int n);
int   InfilCreateSIArray  (int n);
int   InfilCreateTimeseriesArray(int n);
int   InfilCreateDoubleArray  (int n);
void  InfilSetAll          (int p, int i, int status, int value0);
void  InfilSetStatus      (int p, int i, int status);
void  InfilSetValue0     (int p, int i, int value0);
void  InfilGetAll        (int p, int i, int *status, int *value0);
int   InfilGetStatus     (int p, int i);
int   InfilGetValue0    (int p, int i);
void  InfilReleaseArray  (int p);
short InfilGetSIElement (int p, int i);
void  InfilSetSIElement (int p, int i, short value);
void  InfilSetTimeseriesElement (int p, int i, const char *StartDate,
                                const char *EndDate, int completeness,
                                int contents);
void  InfilSetDoubleElement (int p, int i, double *num);
double InfilGetDoubleElement (int p, int i);

** PURPOSE

InfilCreate<something>Array allocates a memory block large enough to hold an
array of <something>, size n. It returns a pointer to the array, cast to int,
or 0 if insufficient memory. <something> can be Double, SI (for short int),
SAV (for STATUSANDVALUE), Timeseries (for TIMESERIES).

The InfilSet... and InfilGet... functions set or get the ith element of the
array pointed to by p. The array elements are numbered starting from 0.

When the job is done, the InfilReleaseArray function must be called to release
the memory assigned to the array.
*****/

int InfilCreateSAVArray (n)
int n;
{
    return (int) malloc (n*sizeof(STATUSANDVALUE));
}

/*-----*/

int InfilCreateSIArray (n)
int n;
{
    return (int) malloc (n*sizeof(short int));
}

/*-----*/

int InfilCreateTimeseriesArray (n)
int n;
{
    return (int) malloc (n*sizeof (TIMESERIES));
}

/*-----*/

int InfilCreateDoubleArray (n)
int n;
{
    return (int) malloc (n*sizeof (double));
}

/*-----*/

InfilSetAll (p, i, status, value0)
int p, i, status, value0;

```

```

{
    STATUSANDVALUE *ptr;

    ptr = (STATUSANDVALUE *) p + i;
    ptr -> status = status;
    ptr -> value0 = value0;
}

/*-----*/
InfilSetStatus (p, i, status)
int p, i, status;
{
    STATUSANDVALUE *ptr;

    ptr = (STATUSANDVALUE *) p + i;
    ptr -> status = status;
}

/*-----*/
InfilSetValue0 (p, i, value0)
int p, i, value0;
{
    STATUSANDVALUE *ptr;

    ptr = (STATUSANDVALUE *) p + i;
    ptr -> value0 = value0;
}

/*-----*/
InfilGetAll (p, i, pstatus, pvalue0)
int p, i;
int *pstatus, *pvalue0;
{
    STATUSANDVALUE *ptr;

    ptr = (STATUSANDVALUE *) p + i;
    *pstatus = ptr -> status;
    *pvalue0 = ptr -> value0;
}

/*-----*/
int InfilGetStatus (p, i)
int p, i;
{
    STATUSANDVALUE *ptr;

    ptr = (STATUSANDVALUE *) p + i;
    return ptr -> status;
}

/*-----*/
int InfilGetValue0 (p, i)
int p, i;
{
    STATUSANDVALUE *ptr;

    ptr = (STATUSANDVALUE *) p + i;
    return ptr -> value0;
}

/*-----*/
InfilReleaseArray (p)
int p;
{
    free ((void *) p);
}

```

```

/*-----*/
short InfilGetSIElement (p, i)
int p, i;
{
    short *ptr;

    ptr = (short *) p + i;
    return *ptr;
}

/*-----*/

InfilSetSIElement (p, i, value)
int p, i;
short value;
{
    short *ptr;

    ptr = (short *) p + i;
    *ptr = value;
}

/*-----*/

InfilSetTimeseriesElement (p, i, StartDate, EndDate, completeness, contents)
int p, i, completeness, contents;
char *StartDate, *EndDate;
{
    TIMESERIES *ptr;

    ptr = (TIMESERIES *)p + i;
    strcpy (ptr -> StartDate, StartDate);
    strcpy (ptr -> EndDate, EndDate);
    ptr -> completeness = (short *) completeness;
    ptr -> contents = (STATUSANDVALUE *) contents;
}

/*-----*/

InfilSetDoubleElement (p, i, num)
int p, i;
double num;
{
    * ((double *) p + i) = num;
}

/*-----*/

double InfilGetDoubleElement (p, i)
int p, i;
{
    return * ((double *) p + i);
}

/*****
** PROCEDURE:      InfilGetCompleteness
** AUTHOR:        ANCHR
** FORMAT:        int InfilGetCompleteness (int instrument, const char *time res t,
**                                     char *StartDate, char *EndDate, char ErrorText[256]);
** PURPOSE

```

The procedure searches the table 'timeseries' of the specified timeseries, reads from there information about the completeness of a timeseries, and puts this information in a C array of short int that it creates. It returns the pointer to the array created, unless error, in which case it returns 0. A message describing the error is put in ErrorText. Furthermore, it puts the first date for which data exist in StartDate, and the last date in EndDate. The array created has one element for each date unit (i.e. day, month or year). The first element corresponds to StartDate, the second to the next date unit, etc. Each

element can have one of the following values:

```
TC_EXISTS      if there is valid data on level 3 (this includes
                data corrected after found to be non-homogeneous)
TC_GAP        if there is a missing value
TC_FILLED     if there is data, but comes from infilling.
TC_NOHOMOCHK  if there is data, but has not been checked for
                homogeneity.
```

The constants are defined in infill.h. Take care to make consistent definitions in INGRES/Windows4GL.

NOTES ON THE FORMAT OF THE TABLE 'timeseries'.

The hydroscope database has a table named 'timeseries', which holds information about timeseries. Its name is not very successful, because each row of the table does not hold a timeseries, but a section of a timeseries. A section is defined as a part of a timeseries that has no missing values. Thus, a timeseries is broken into as many sections as needed, and each one is a row in the table 'timeseries'. This happens for each level separately.

The completeness of a timeseries is found based on the following observations:

- 1) If a value is valid and does not come from infilling (TC_EXISTS) then it exists in both levels 2 and 3.
- 2) If a value is missing (TC_GAP), it is in neither level.
- 3) If a value is filled (TC_FILLED), it exists in level 3, but not in level 2.
- 4) If a value has not been checked for homogeneity (TC_NOHOMOCHK), it exists in level 2, but not in level 3.

The procedure assumes that the opposites are also valid. Thus, it makes a mistake in the following cases:

- 1) If a value has been checked for homogeneity, found incorrect, discarded, then considered missing, and filled, then the wrong value appears in level 2 and the correct in level 3. Thus, this value should be TC_FILLED, but it appears as TC_EXISTS.
- 2) If a value has been checked for homogeneity, found incorrect, discarded, then considered missing, but not yet filled, then the wrong value appears in level 2 and nothing in level 3. Thus, this value should be TC_GAP, but it appears as TC_NOHOMOCHK.

However, the case where a datum is discarded during homogeneity checking is very rare (usually, if data are found to be non-homogeneous, they are corrected, not discarded), and thus the returned results are considered to be correct. Later, when the program reads the actual data from the appropriate tables of the database, it can find out if a mistake has been made. If it is found that one of the above rare cases has been encountered, the user must be warned.

All this alchemy is done in order to avoid reading data from remote nodes. The 'timeseries' table is in the local node.

```
*****/
int InfilGetCompleteness (instrument, time_res_t, StartDate, EndDate, ErrorText)
exec sql begin declare section;
    int instrument;
    char *time_res_t, *StartDate, *EndDate;
    char ErrorText[256];
exec sql end declare section;
{
    short int *ptr, *p, *endp;
    int n;
    exec sql begin declare section;
        char start_date[25], end_date[25];
    exec sql end declare section;

    /* Find the StartDate and EndDate and create the array. */
    exec sql whenever SQLERROR goto SQLERROR;
    exec sql whenever NOT FOUND goto NOTFOUND;
    exec sql select varchar(hdate(min(int4(start_date)))) ,
        varchar(hdate(max(int4(end_date))))
        into :StartDate, :EndDate
        from timeseries
        where instrument = :instrument
        and time_res_t = :time_res_t
        and (level=2 or level=3);
```

```

exec sql commit;
if ((ptr = (short *) InfilCreateSIArray
(n = SubtractDateStrings (EndDate, StartDate,time_res_t)+1 )) == NULL)
    goto MEMORYERROR;
exec sql whenever NOT FOUND continue;

/* Fill the array with TC_TMP1. */
for (p = ptr, endp = ptr+n-1; p <= endp; *(p++) = TC_TMP1)
    ;

/* First search level 2, and put TC_TMP2 if there is a value in each
 * position.
 */
exec sql repeated select start_date, end_date
                    into :start_date, :end_date
                    from timeseries
                    where instrument = :instrument
                    and time_res_t = :time_res_t
                    and level = 2;
exec sql begin;
    p = ptr + SubtractDateStrings (start_date,StartDate,time_res_t);
    endp = p + SubtractDateStrings (end_date,start_date,time_res_t);
    while (p <= endp)
        *(p++) = TC_TMP2;
exec sql end;
exec sql commit;

/* Then search level 3, and act accordingly. */
exec sql repeated select start_date, end_date
                    into :start_date, :end_date
                    from timeseries
                    where instrument = :instrument
                    and time_res_t = :time_res_t
                    and level = 3;
exec sql begin;
    p = ptr + SubtractDateStrings (start_date,StartDate,time_res_t);
    endp = p + SubtractDateStrings (end_date,start_date,time_res_t);
    while (p <= endp)
    {
        if (*p == TC_TMP1)
            *p = TC_FILLED;
        else
            *p = TC_EXISTS;
        ++p;
    }
exec sql end;
exec sql commit;

/* Take care about values which do not exist in level 3. */
endp = ptr + SubtractDateStrings (EndDate, StartDate, time_res_t);
for (p = ptr; p <=endp; ++p)
    switch (*p)
    {
        case TC_TMP1:
            *p = TC_GAP;
            break;

        case TC_TMP2:
            *p = TC_NOHOMOCHECK;
            break;
    }

/* Finished. */
return (int) ptr;

SQLERROR:
InfilReleaseArray ((int) ptr);
exec sql inquire_sql (:ErrorText = errortext);
exec sql rollback;
return 0;

MEMORYERROR:

```

```

strcpy (ErrorText, VC_INSUFFICIENTMEMORY);
return 0;

NOTFOUND:
strcpy (ErrorText, VC_NOTIMESERIES);
return 0;
}

/*****
** PROCEDURE:      InfilGetContents
** AUTHOR:        ANCHR
** FORMAT:        int InfilGetContents (int instrument, const char *time_res_t,
                const char *StartDate, const char *EndDate, int linear,
                char ErrorText [256]);
** PURPOSE:       The procedure selects all the contents of the specified
                timeseries, and puts them in a C array which it creates. It
                returns the pointer to the array. If SQL error or insufficient
                memory, the function returns 0. 'linear' holds 1, 2, or 0 if
                the way of storing data is linear, differential or normal
                respectively.
*****/

exec sql declare CursorName cursor for StatementName;

int InfilGetContents (instrument, time_res_t, StartDate, EndDate,
                    linear, ErrorText)
int instrument, linear;
char *time_res_t;
exec sql begin declare section;
char ErrorText [256], *StartDate, *EndDate;
exec sql end declare section;
{
    char TableName [33], TimeResClause[32];
    char PrevDate [20];
    int j, k, n, length;
    STATUSANDVALUE *p, *q, *ptr;

    exec sql begin declare section;
    char StatementBuffer [512], date [20];
    int status, value0, dummy;
    int offset, i, IntStartDate, IntEndDate;
    short value0null;
    exec sql end declare section;

    exec sql whenever sqlerror goto SQLERROR1;

    TRACE1 ("Getting local contents for instrument %d...\n", instrument);
    if (InfilGetTableName (instrument, time_res_t, TableName, ErrorText))
        return 0;
    if (strncmp (TableName, "aggr", 4) && strncmp(TableName, "x_aggr", 6))
        *TimeResClause = '\0';
    else
    {
        exec sql select time_res into :i from time_res_ts
            where time_res_t = :time_res_t;
        sprintf (TimeResClause, "and time_res=%d", i);
    }
    n = SubtractDateStrings (EndDate, StartDate, time_res_t) + 1;
    if ((ptr = (STATUSANDVALUE *) InfilCreateSAVArray (n)) == NULL)
        goto ERROR;
    exec sql whenever sqlerror goto SQLERROR;
    exec sql select int4(hdate(:StartDate)), int4(hdate(:EndDate))
        into :IntStartDate, :IntEndDate;
    for (i=1, p=ptr; i<=n; ++i, ++p)
        p->value0 = NULL_VALUE0;

    /* Read all level 3 values. */
    sprintf (StatementBuffer, "select status, value0, date, int4(date) "
            "from %s "
            "where instrument = %d "
            "and getstatus(status, 'N')=1 ")

```

```

        "and int4(date)>=%d "
        "and int4(date)<=%d "
        "%s "
        "order by 4 ",
        TableName, instrument, IntStartDate, IntEndDate, TimeResClause);
exec sql prepare StatementName from :StatementBuffer;
exec sql open CursorName for readonly;
strcpy (PrevDate, AddIntToDateString(StartDate,time_res_t,-1));
for (;;)
{
    exec sql fetch CursorName
        into :status, :value0:value0null, :date, :dummy;

    if (sqlca.sqlcode == 100)
        break;
    switch (linear)
    {
        case 0:
            p = ptr+SubtractDateStrings(date,StartDate,time_res_t);
            p -> status = status;
            p -> value0 = value0null ? NULL_VALUE0 : value0;
            break;

        case 1: /* Linear. */
            i = SubtractDateStrings(PrevDate,StartDate,time_res_t);
            j = SubtractDateStrings(date, PrevDate, time_res_t);
            q = ptr+i;
            for (p=q+1, k=1; k<=j; ++p,++k)
            {
                p->status = status;
                p->value0 = value0null ? NULL_VALUE0 :
                    (int)(((double)value0 - q->value0)*k/j+0.5);
            }
            strcpy (PrevDate, date);
            break;

        case 2: /* Differential. */
            i = SubtractDateStrings(PrevDate,StartDate,time_res_t);
            j = SubtractDateStrings(date, PrevDate, time_res_t);
            q = ptr+i;
            for (p=q+1, k=1; k<=j; ++p,++k)
            {
                p->status = status;
                p->value0 = value0null ? NULL_VALUE0 :
                    (int) ((double)value0 * k/j +0.5);
            }
            strcpy (PrevDate, date);
            break;
    }
}
exec sql close CursorName;

/* Now load level 2 values for which there are no level 3 values. */
TRACE ("First select finished. Now searching level 2...\n");
sprintf (StatementBuffer, "select status, value0, date, int4(date) "
        "from %s "
        "where instrument = %d "
        "and getstatus(status, 'M')=1 "
        "and int4(date)>=%d "
        "and int4(date)<=%d "
        "%s "
        "order by 4 ",
        TableName, instrument, IntStartDate, IntEndDate, TimeResClause);
exec sql prepare StatementName from :StatementBuffer;
exec sql open CursorName for readonly;
strcpy (PrevDate, AddIntToDateString(StartDate,time_res_t,-1));
for (;;)
{
    exec sql fetch CursorName
        into :status, :value0:value0null, :date, :dummy;
    if (sqlca.sqlcode == 100)
        break;
}

```



```

switch (linear)
{
    case 0:
        p = ptr+SubtractDateStrings(date,StartDate,time_res_t);
        if (p->value0 == NULL_VALUE0)
        {
            p -> status = status;
            p -> value0 = value0null ? NULL_VALUE0 : value0;
        }
        break;

    case 1: /* Linear. */
        i = SubtractDateStrings(PrevDate,StartDate,time_res_t);
        j = SubtractDateStrings(date, PrevDate, time_res_t);
        q = ptr+i;
        for (p=q+1, k=1; k<=j; ++p,++k)
        {
            if (p->value0 == NULL_VALUE0)
            {
                p->status = status;
                p->value0 = value0null ? NULL_VALUE0 :
                    (int) (((double)value0 - q->value0)*k/j+0.5);
            }
        }
        strcpy (PrevDate, date);
        break;

    case 2: /* Differential. */
        i = SubtractDateStrings(PrevDate,StartDate,time_res_t);
        j = SubtractDateStrings(date, PrevDate, time_res_t);
        q = ptr+i;
        for (p=q+1, k=1; k<=j; ++p,++k)
        {
            if (p->value0 == NULL_VALUE0)
            {
                p->status = status;
                p->value0 = value0null ? NULL_VALUE0 :
                    (int) ((double)value0 * k/j +0.5);
            }
        }
        strcpy (PrevDate, date);
        break;
}
}
exec sql close CursorName;

exec sql commit;

return (int) ptr;

SQLERROR:
InfilReleaseArray ((int) ptr);

SQLERROR1:
exec sql inquire sql (:ErrorText = errortext);
exec sql rollback;
return 0;

ERROR:
strcpy (ErrorText, VC_INSUFFICIENTMEMORY);
return 0;
}

/*****
** PROCEDURE InfilGetTableName
** FORMAT      int InfilGetTableName (int instrument, const char *time_res_t,
**              char *TableName, char *ErrorText);
** PURPOSE     Finds the name of the table of the database that has the data
**              for the instrument with time_res_t. Returns non-zero on error,
**              in which case it puts a message in ErrorText.
*****/

```

```
int InfilGetTableName (instrument, time_res_t, TableName, ErrorText)
exec sql begin declare section;
int instrument;
char *time_res_t;
char TableName [33];
char ErrorText[256];
exec sql end declare section;
{
    exec sql begin declare section;
    char fmt [9];
    exec sql end declare section;

    char *s;

    exec sql whenever sqlerror goto SQLERROR;
    exec sql select distinct table_p, format
        into :TableName, :fmt
        from timeseries
        where instrument = :instrument and time_res_t = :time_res_t;
    exec sql commit;
    if ((s=strchr (TableName, ' ')) != NULL)
        *s = '\0';
    strcat (TableName, "_");
    strcat (TableName, fmt);
    if ((s=strchr (TableName, ' ')) != NULL)
        *s = '\0';
    return 0;

SQLERROR:
exec sql inquire_sql (:ErrorText = errortext);
exec sql rollback;
return 1;
}
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Παπακώστας, Ν., «Προδιαγραφές για την τυποποίηση στην ανάπτυξη λογισμικού», Τεχνική αναφορά Υδροσκοπίου 1/6, Εθνικό Μετσόβιο Πολυτεχνείο, 1992.
- Παπακώστας, Ν., «Σχεδιασμός σχήματος βάσης δεδομένων», Τεχνική αναφορά Υδροσκοπίου 1/15, Εθνικό Μετσόβιο Πολυτεχνείο, 1993.
- Fiering, M. B., "Use of correlation to improve estimates of the mean and variance", *U.S. Geological Survey Professional Paper*, 434-C, pp. C1-C9, 1963.
- Foufoula-Georgiou, E., "Estimation of missing values in monthly rainfall series", *Proceedings of Stormwater and Water Quality Model Users' Group Meeting*, EPA-600/9-83-015, pp. 177-191, 1983.
- Gilroy, E. J., "Reliability of a variance estimate obtained from a sample augmented by multivariate regression", *Water Resources Research*, vol. 6, no. 6, pp. 1595-1600, 1970.
- Grygier, J. C., and J. R. Stedinger, "A generalized maintenance of variance extension procedure for extending correlated series", *Water Resources Research*, vol. 25, no. 3, pp. 345-349, 1989.
- Hirsch, R. M., "A comparison of four record extension techniques", *Water Resources Research*, vol. 18, no. 4, pp. 1081-1088, 1982.
- Ingres, *INGRES/SQL Reference Manual for the UNIX and VMS Operating Systems*, Release 6.4, Ingres Corporation, 1991.
- Ingres, *INGRES/Windows4GL: Application Editor User's Guide for INGRES/Windows4GL*, Ingres Corporation, 1992.
- Kernighan, B. W., and D. M. Ritchie, *The C Programming Language*, Second edition, Prentice Hall, 1988.
- Matalas, N. C., and B. Jacobs, "A correlation procedure for augmenting hydrologic data", *U.S. Geological Survey Professional Paper*, 434-E, pp. E1-E7, 1964.
- Moran, M. A., "On estimators obtained from a sample augmented by multiple regression", *Water Resources Research*, vol. 10, no. 1, pp. 81-85, 1974.
- Papakostas, N., I. Nalbantis, and D. Koutsoyiannis, "Modern Computer Technologies in Hydrologic Data Management", υπό έκδοση στο *Proceedings of the Second European Conference on Advances in Water Resources Technology and Management*, 1994.
- Salas, J. D., "Filling in missing observations and extension of records", *Handbook of Hydrology*, J. R. Maidment (ed), chapter 19, section 4, pp. 19.42-19.48, 1993.
- Vogel, R. M., and J. R. Stedinger, "Minimum variance streamflow record augmentation procedures", *Water Resources Research*, vol. 21, no. 5, pp. 715-723, 1985.