

Χρήση της R για επιστημονικό προγραμματισμό

Karline Soetaert

Κέντρο Εκβολών Ποταμών και Θαλάσσιας Οικολογίας

Ολλανδικό Ινστιτούτο Οικολογίας

Ολλανδία

Σεπτέμβριος 2008

Μετάφραση από την Αγγλική Γλώσσα:

Παύλος Φλωράτος

Κεφάλαιο 1

Το λογισμικό R

Η R (βασική ομάδα εξέλιξης της R, 2007) είναι η ανοιχτή (βλ δωρεάν) έκδοση της γλώσσας S. Είναι περισσότερο γνωστή ως ένα πακέτο το οποίο εκτελεί στατιστική ανάλυση και γραφικά. Παρ' όλα αυτά, η R είναι πολύ περισσότερα: είναι μία γλώσσα υψηλού επιπέδου με την οποία μπορεί κανείς να εκτελέσει πολύπλοκους υπολογισμούς, να εφαρμόσει νέες μεθόδους και να επιτύχει στοιχεία υψηλής ποιότητας. Η R έχει λειτουργίες υψηλού επιπέδου που μπορούν να εφαρμοστούν σε πίνακες, να εκτελέσουν αριθμητική ολοκλήρωση, και εξελιγμένη στατιστική, οι οποίες μάλιστα μπορούν εύκολα να τεθούν σε εφαρμογή και οι οποίες την καθιστούν ιδανική για οπτικοποίηση δεδομένων, στατιστική ανάλυση και μαθηματικά μοντέλα. Σκοπός αυτών των σημειώσεων είναι η γνωριμία και εξοικείωσή σας με τη γλώσσα R. Οι σημειώσεις βασίζονται σε ένα βιβλίο (Soetaert και Herman, 2008) σχετικά με το οικολογικό μοντέλο στο οποίο η R χρησιμοποιείται ευρέως για ανάπτυξη, εφαρμογή και οπτικοποίηση μοντέλων προσομοίωσης.

Υπάρχουν πολλές εξαιρετικές πηγές για την εκμάθηση της R (ή της S) γλώσσας. Η R συνοδεύεται από αρκετά εγχειρίδια χρήσης τα οποία μπορεί να συμβουλευτεί κανείς μέσα από το κύριο πρόγραμμα της R. (Βοήθεια/Εγχειρίδια) Το R-intro.pdf είναι μια καλή αρχή. Πολλές άλλες καλές εισαγωγές στην R είναι διαθέσιμες, ορισμένες δωρεάν στο διαδίκτυο και προσβάσιμες διαμέσου της ιστοσελίδας (www.r-project.org.) Εγώ προτιμώ την εισαγωγή στην R της Petra Kuhner και του Bill Venables (2005) αλλά προσέξτε: αυτή η «εισαγωγή» αποτελείται από περισσότερες από 300 σελίδες!

1.1 Εγκατάσταση της R

Η R μπορεί να κατέβει από την ακόλουθη ιστοσελίδα: <http://www.r-project.org/> Επιλέξτε την προμεταγλωττισμένη δυαδική κατανομή. Σε αυτήν την ιστοσελίδα θα βρείτε επίσης χρήσιμη τεκμηρίωση. Για να χρησιμοποιήσετε την R για τα παραδείγματα αυτής της παρουσίασης αρκετά πακέτα χρειάζεται να κατεβούν.

- deSolve. Εκτελεί ολοκλήρωση (Soetaert και άλλοι, 2008).
- rootSolve. Βρίσκει την ρίζα των εξισώσεων (Soetaert, 2008).
- scatterplot3d. Για 3-D γραφικά (Ligges και Machler, 2003).
- seacarb. Υδατική χημεία (Proye και άλλοι, 2007).
- marelac. Σύνολα δεδομένων, συναρτήσεις και σταθερές για τις θαλάσσιες και λιμναίες επιστήμες (Soetaert και Meysman, 2008).

Αν τρέχετε την R στα Windows, το κατέβασμα ορισμένων εκδοχών μπορεί να γίνει καλύτερα μέσα από το ίδιο το πρόγραμμα της R. Επιλέξτε στο μενού την επιλογή "packages/install

packages” επιλέξτε μία παρεμφερή ιστοσελίδα (πχ France (Paris)) και επιλέξτε το πακέτο που χρειάζεστε. Αν εγκαταστήσετε το πακέτο “marelac” τότε όλα τα άλλα πακέτα θα εγκατασταθούν αυτόματα επίσης.

1.2 Άλλο χρήσιμο λογισμικό

Προτιμώ να εκτελώ την R μέσα από τον συντάκτη **Tinn-R**, ο οποίος μπορεί να κατέβει από τα URL <http://sourceforge.net/projects/tinn-r> και <http://www.sciviews.org/Tinn-R>. Αυτός ο συντάκτης περιέχει σύνταξη και βοήθεια συμβατές με την R. Κατεβάστε την πιο πρόσφατη έκδοση του Tinn-R και εγκαταστήστε τη.

Μέσα από το πρόγραμμα Tinn-R, ανοίγετε την R μέσω του μενού (R/start preferred Rgui)

Κεφάλαιο 2

Γρήγορη επισκόπηση της R

Ο κώδικας R είναι ιδιαίτερα αναγνώσιμος, εφόσον αντιληφθείτε ότι:

- “<-” είναι ο τελεστής εκχώρησης
- οτιδήποτε ξεκινά με «#» θεωρείται σχόλιο
- Η R είναι ευαίσθητη ως προς τα κεφαλαία και τα μικρά γράμματα: το “a” και το “A” είναι δύο διαφορετικά αντικείμενα

2.1 Κονσόλα έναντι σεναρίων

Υπάρχουν δύο τρόποι με τους οποίους εργαζόμαστε στην R:

1. Μπορούμε να πληκτρολογούμε εντολές μέσα στο παράθυρο **κονσόλας** της R σε γραμμή εντολών «>» και να χρησιμοποιούμε την R ως έναν ισχυρό επιστημονικό υπολογιστή τσέπης.

Για παράδειγμα, η εισαγωγή στο παράθυρο κονσόλας:

```
> pi*0.795^2 ; 25*6/sqrt(67) ; log(25)
```

θα μας δώσει ως απάντηση:

```
[1] 1.985565  
[1] 18.32542  
[1] 3.218876
```

Εδώ τα sqrt και log είναι ενσωματωμένες λειτουργίες στην R. Το π είναι ενσωματωμένη σταθερά. Το ερωτηματικό (;) χρησιμοποιείται για να διαχωρίζει εντολές στην R.

Στο παράθυρο κονσόλας, τα πλήκτρα βέλους <UP> και <DOWN> χρησιμοποιούνται για την πλοήγηση διαμέσου εκ των προτέρων πληκτρολογημένων προτάσεων.

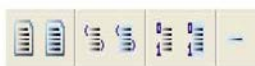
2. Εναλλακτικά, μπορούμε να δημιουργήσουμε **σενάρια R** σε έναν συντάκτη (π.χ. Tinn-R) και να τα αποθηκεύσουμε σε ένα αρχείο (“filename.R”) για επαναχρησιμοποίηση αργότερα. Τα σενάρια R είναι ακολουθίες από εντολές R και εκφράσεις. Αυτά τα σενάρια πρέπει να υποβάλλονται στην R πριν εκτελεστούν.

Αυτό μπορεί να γίνει με διάφορους τρόπους:

- Πληκτρολογώντας, στο παράθυρο κονσόλας της R:

```
> source ("filename.R")
```

- Ανοίγοντας το αρχείο, αντιγράφοντας το σενάριο R στο πρόχειρο (CTRL-C) και επικολλώντας το (CTRL-V) στο παράθυρο κονσόλας της R.
- Αν δεν χρησιμοποιείτε τον συντάκτη tinn-R, το αρχείο ανοίγει σαν ένα σενάριο R μέσα από την κονσόλα της R. Αφού επιλέξετε το σενάριο και πατήσετε το κουμπί οι εντολές εκτελούνται και ο κέρσορας μετακινείται στην επόμενη σειρά.
- Αν χρησιμοποιείτε τον συντάκτη tinn-R, μπορείτε είτε να υποβάλλετε ολόκληρο το αρχείο (πλήκτρα 1,2), επιλεγμένα τμήματα του κειμένου (πλήκτρα 3,4) , να υποβάλλετε τα επιλεγμένα τμήματα (πλήκτρα 5,6) ή γραμμή ανά γραμμή (τελευταίο πλήκτρο).



Καθ' όλη την έκταση αυτών των σημειώσεων, χρησιμοποιείται η ακόλουθη σύμβαση:

```
> 3/2
```

Δηλώνει είσοδο στο παράθυρο κονσόλας (ο κέρσορας είναι «>»)

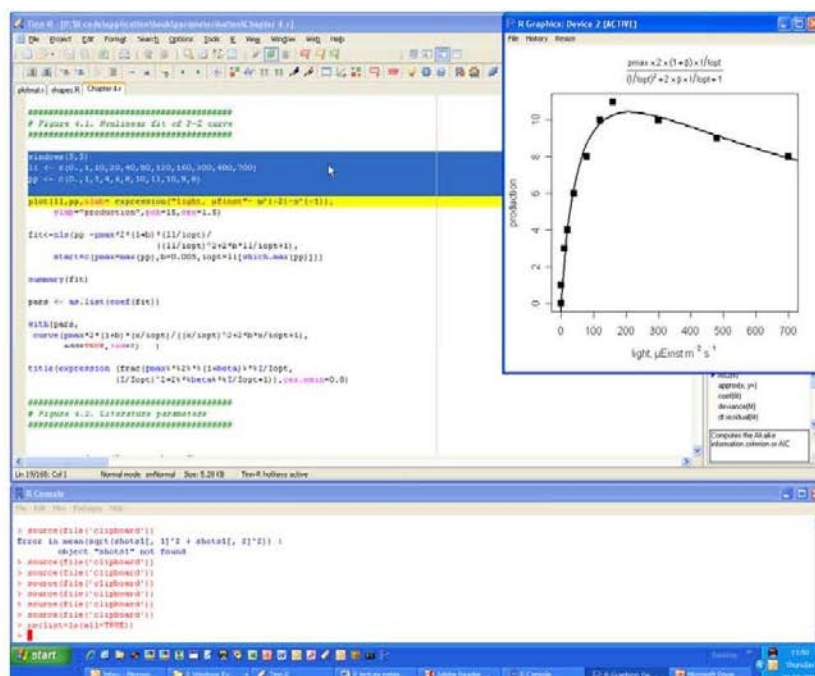
```
[1] 1
```

είναι η έξοδος R, όπως είναι γραμμένη στο παράθυρο κονσόλας

```
getwd()
```

Είναι η δήλωση R σε ένα αρχείο σεναρίου (απουσιάζει ο κέρσορας)

Μια αποτύπωση οθόνης μίας τυπικής συνεδρίασης στο Tinn-R, με τον συντάκτη Tinn-R (άνω παράθυρο) και την κονσόλα R (κάτω παράθυρο) δίνεται παρακάτω. Ένα αρχείο σεναρίου έχει ανοιχθεί στον συντάκτη Tinn-R – παρατηρήστε την ευαίσθητη σύνταξη ως προς τα συμφραζόμενα (πράσινο=σχόλια, μπλε=διακεκομμένες λέξεις, ροζ=παραμέτροι R). Αρκετές γραμμές κώδικα R έχουν επιλεγεί (μπλε περιοχή) και σταλεί στην κονσόλα R η οποία έχει δημιουργήσει το παράθυρο γραφικών που παρουσιάζεται ανεξάρτητο από τα άλλα παράθυρα.



2.2 Λαμβάνοντας βοήθεια, παραδείγματα, παρουσιάσεις

Η R έχει εκτεταμένες υπηρεσίες **βοήθειας**. Πέρα από το παράθυρο Βοήθειας που ανοίγει από το μενού Βοήθειας, ή τις υπηρεσίες βοήθειας HTML, είναι επίσης διαθέσιμη από τον κέρσορα της γραμμής εντολών.

Για παράδειγμα, πληκτρολογώντας:

```
> ?log  
> ?sin  
> ?sqrt  
> ?round  
> ?$special
```

Θα εξηγήσει για λογαρίθμους και εκθετικές συναρτήσεις, τριγωνομετρικές συναρτήσεις και άλλες συναρτήσεις

```
> ?Arithmetic
```

Κατατάσσει σε λίστα τις αριθμητικές λειτουργίες της R

```
> help.search("factor")
```

Κατατάσσει εμφανίσεις της λέξης <factor> στις εντολές της R

Μερικές φορές η καλύτερη βοήθεια παρέχεται από την ενεργή λίστα του ηλεκτρονικού ταχυδρομείου. Αν έχετε ένα συγκεκριμένο πρόβλημα, απλώς πληκτρολογήστε R:<problem> στη μηχανή αναζήτησης. Υπάρχει η πιθανότητα κάποιος να έχει αντιμετωπίσει το ίδιο πρόβλημα και να έχει ήδη επιλυθεί.

Τα περισσότερα από τα βοηθητικά αρχεία επίσης περιλαμβάνουν **παραδείγματα**. Μπορείτε να τα τρέξετε όλα χρησιμοποιώντας την εντολή R “example” .

Για παράδειγμα, πληκτρολογώντας στο παράθυρο της κονσόλας

```
> example(matrix)
```

θα τρέξουν όλα τα παραδείγματα των βοηθητικών αρχείων των πινάκων `matrix`

```
> example(pairs)
```

θα τρέξουν όλα τα παραδείγματα των βοηθητικών αρχείων των ζευγών `pairs` (δοκιμάστε το αυτό! Η εντολή `pairs` είναι ένας πολύ ισχυρός τρόπος για την απεικόνιση των σχέσεων ζευγών).

Εναλλακτικά, μπορείτε να επιλέξετε ένα παράδειγμα, να το αντιγράψετε στο πρόχειρο (CTRL-C για τους χρήστες των Windows) και μετά να το επικολλήσετε (CTRL-V) στο παράθυρο της κονσόλας.

Επιπλέον, το κυρίως λογισμικό της R και πολλά πακέτα της R συνοδεύονται από υλικό **επίδειξης**. Πληκτρολογώντας:

```
> demo()
```

θα σας δώσει μια λίστα από διαθέσιμες επιδείξεις στο κύριο λογισμικό.

```
> demo(graphics)
```

θα σας επιδείξει μερικές απλές δυνατότητες γραφικών.

2.3 Μικρά πράγματα που πρέπει να θυμάστε

1. Τα ονόματα διαδρομής στην R γράφονται με δεξιές καθέτους, “/” αν και στα Windows χρησιμοποιούνται οι ανάστροφες κάθετοι “\”. Ως εκ τούτου, για να δημιουργήσουμε έναν φάκελο εργασίας στην R:

```
setwd("C:/R code/")
```

2. Αν μια πρόταση σε μία γραμμή είναι συντακτικά σωστή, η R θα την εκτελέσει, ακόμη και αν η πρόθεση είναι να προχωρήσει στην επόμενη γραμμή. Για παράδειγμα αν γράφουμε:

```
A <- 3 + cos(pi)
- sqrt(5)
[1] -2.236068
```

Τότε το A θα λάβει την τιμή $(3 - \cos(\pi))$ και η R θα τυπώσει την τιμή $(-\sqrt{5})$.

Αντιθέτως στις επόμενες σειρές:

```
3 + cos(pi) -
sqrt(5)
[1] -0.236068
```

Η R θα τυπώσει την τιμή $3 + \cos(\pi) - \sqrt{5}$: καθώς η πρόταση στην πρώτη σειρά δεν ήταν συντακτικά ολοκληρωμένη, η R έχει (σωστά) υποθέσει ότι συνεχιζόταν στην επόμενη σειρά.

Προσέξτε αν θέλετε να διαμοιράσετε (χωρίσετε) μία πολύπλοκη εντολή σε αρκετές σειρές. Αυτά τα λάθη είναι πολύ δύσκολο να ανιχνευθούν, είναι λοιπόν καλύτερα να αποφεύγονται.

2.4 Ασκήσεις – χρησιμοποιώντας την R ως αριθμομηχανή

Είναι πολύ βολικό να χρησιμοποιείτε την R ως μια ισχυρή αριθμομηχανή. Αυτό μπορεί να γίνει καλύτερα μέσα από την κονσόλα της R

1. Χρησιμοποιήστε την κονσόλα για να υπολογίσετε την τιμή

- $(4/6*8-1)^{2/3}$
- $\ln(20)$
- $\log_2(4096)$
- $2*\pi*3$
- $e^{2+\cos(\pi/8)}$
- $\sqrt{2.3^2 + 5.4^2 - 2*2.3*5.4*\cos(\pi/8)}$

Συμβουλή: ίσως χρειαστεί να κοιτάξετε στα βοηθητικά αρχεία για μερικές από αυτές τις συναρτήσεις πληκτρολογώντας ?«+» θα ανοίξει ένα βοηθητικό αρχείο με τα συνηθισμένα αριθμητικά σύμβολα πράξεων.

2. Τώρα γράψτε τις εντολές R σε ένα αρχείο σεναρίου χρησιμοποιώντας τον συντάκτη Tinn-R. Δοκιμάστε τους διαφορετικούς τρόπους με τους οποίους μπορείτε να υποβάλλετε τις εντολές στην R.

Κεφάλαιο 3

Προγραμματισμός με διαφορετικές εκδοχές της R

3.1 Αριθμοί, διανύσματα, πίνακες και συστοιχίες

3.1.1 Ανάθεση εργασίας

Όταν χρησιμοποιούνται παραλλαγές, χρειάζεται να διαχωριστούν αρχικά με αριθμούς

```
> A <- 1  
> B <- 2  
> A+B  
[1] 3
```

η R μπορεί να δεχθεί ως επιχειρήματα για τις λειτουργίες της απλούς **αριθμούς**, **διανύσματα**, **πίνακες** ή **συστοιχίες**

```
> V<-factorial(10)
```

Υπολογίζει $10!$ ($=1*2*3*...*10$). Ο διαχειριστής «<-» **αναθέτει** το αποτέλεσμα αυτού του υπολογισμού στην παραλλαγή V. Η V μπορεί τότε να χρησιμοποιηθεί στους ακόλουθους υπολογισμούς:

```
> V/10  
[1] 362880
```

Σημειώστε ότι η ανάθεση μιας τιμής στη V δεν εκφράζεται στο παράθυρο. Για να εμφανίσουμε τη V απλώς γράφουμε:

```
> V  
[1] 3628800
```

Εναλλακτικά, μπορούμε να αναθέσουμε το αποτέλεσμα των υπολογισμών σε μία παραλλαγή ΚΑΙ να παρακολουθήσουμε τα αποτελέσματα, περικλείοντας την εντολή σε παρενθέσεις:

```
> (X <- sin(3/2*pi) )  
[1] -1
```

Εκτός από ακεραίους, πραγματικούς και μιγαδικούς αριθμούς, επίσης αναγνωρίζει το άπειρο (`inf`) και μη αριθμούς (`NaN`).

```
> 1/0  
> 0/0  
> 1e-8 * 1000
```

(όπου η σημείωση “e-8” δηλώνει 10^{-8})

3.1.2 Διανύσματα

Τα **διανύσματα** αποτελούνται από μια διατεταγμένη συλλογή αριθμών και μπορούν να δημιουργηθούν με πολλούς τρόπους:

- Χρησιμοποιώντας την συνάρτηση-R “vector”
- Η συνάρτηση “c ()” συνδυάζει αριθμούς σε ένα διάνυσμα¹
- Ο διαχειριστής «:» δημιουργεί μια σειρά τιμών, κάθε μία μεγαλύτερη ή μικρότερη από την προηγούμενη:
- Μια πιο γενική αλληλουχία (σειρά) μπορεί να παραχθεί από τη λειτουργία της R “seq” .
- Ίδια ποσότητα επαναλαμβάνεται χρησιμοποιώντας την λειτουργία της R “rep”.
- ...

Για παράδειγμα, οι εντολές

```
> c(0, pi/2, pi, 3*pi/2, 2*pi)
[1] 0.000000 1.570796 3.141593 4.712389 6.283185

> seq(from=0,to=2*pi, by=pi/2 )
[1] 0.000000 1.570796 3.141593 4.712389 6.283185

> seq(0, 2*pi, pi/2 )
[1] 0.000000 1.570796 3.141593 4.712389 6.283185
```

Θα δημιουργήσουν όλες ένα διάνυσμα αποτελούμενο από 0, π ... 2*π.

Σημειώστε ότι η λειτουργία της R “seq” λαμβάνει ως εισερχόμενες (μεταξύ άλλων) τις παραμέτρους ‘from’ , ‘to’ και ‘by’ (2^ο παράδειγμα).

Αν η σειρά διατηρηθεί, χρειάζεται να αποσαφηνιστούν με το όνομά τους (3^ο παράδειγμα).

Η επόμενη εντολή υπολογίζει το ημίτονο ενός διανύσματος και εξάγει το αποτέλεσμα:

```
> sin( seq(0, 2*pi, pi/2 ))
[1] 0.00000e+00 1.00000e+00 1.224606e-16 -1.00000e+00 -2.449213e-16
```

Η λειτουργία “rep” χρησιμοποιείται για επανάληψη στοιχείων:

```
> rep(1,times=5)
[1] 1 1 1 1 1

> rep(c(1,2),times=5)
[1] 1 2 1 2 1 2 1 2 1 2

> c(rep(1,5),rep(2,5))
[1] 1 1 1 1 1 2 2 2 2 2
```

Οι επόμενες εντολές:

```
> V <- 1:100
> sqrt(V)
[1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490
...
```

¹ Αυτή είναι ίσως η πιο σημαντική λειτουργία της R.

Δημιουργούν μια ακολουθία ακεραίων ανάμεσα στο 1 και το 100 και παίρνουν την τετραγωνική ρίζα όλων, δείχνοντας το αποτέλεσμα στην οθόνη.

Ο διαχειριστής ‘<-’ αναθέτει την ακολουθία στη V. Μερικά άλλα παραδείγματα του διαχειριστή «:» είναι:

```
> (V <- 0.5:10.5)
[1] 0.5 1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5 10.5
> 6:1
[1] 6 5 4 3 2 1
```

Τέλος, οι εντολές

```
> V <- vector(length=10)
> FF <- vector()
```

παράγουν ένα διάνυσμα V αποτελούμενο από 10 στοιχεία, και ένα διάνυσμα FF αγνώστου μήκους.

Σημείωση: Ένα αξιοσημείωτο στοιχείο της R είναι ότι τα στοιχεία ενός διανύσματος μπορούν να ονομαστούν

```
> (fruit <- c(banana=1, apple=2, orange=3))
banana apple orange
      1      2      3
> names(fruit)
[1] "banana" "apple" "orange"
```

3.1.3 Πίνακες

Οι πίνακες μπορούν επίσης να δημιουργηθούν με αρκετούς τρόπους:

- Μέσω της λειτουργίας της R “matrix”.
- Μέσω της λειτουργίας της R “diag” η οποία κατασκευάζει έναν διαγώνιο πίνακα.
- Οι λειτουργίες “cbind” και “rbind” προσθέτουν στήλες και σειρές σε έναν υπάρχοντα πίνακα, ή σε ένα άλλο διάνυσμα.

Η εντολή

```
> A <- matrix(nrow=2, data=c(1,2,3,4))
```

Δημιουργεί έναν **πίνακα A**, με δύο σειρές, και καθώς υπάρχουν τέσσερα στοιχεία, με δύο στήλες. Σημειώστε ότι τα δεδομένα εισάγονται ως διάνυσμα (χρησιμοποιώντας τη λειτουργία c()).

Οι δύο επόμενες εντολές απεικονίζουν τον πίνακα που ακολουθείται από την τετραγωνική ρίζα των στοιχείων του:

```
> A
      [,1] [,2]
[1,]    1    3
[2,]    2    4

> sqrt(A)
      [,1] [,2]
[1,] 1.000000 1.732051
[2,] 1.414214 2.000000
```

Από σφάλμα, η R γεμίζει έναν πίνακα **με διάταξη στήλης** (δείτε το παράδειγμα παραπάνω)
Παρ' όλα αυτά, αυτό μπορεί να διορθωθεί εύκολα, χρησιμοποιώντας την παράμετρο
byrow:

```
> (M <- matrix(nrow=4, ncol=3, byrow=TRUE, data=1:12))
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
[4,]   10   11   12
```

Ο **ενιαίος πίνακας** δημιουργείται χρησιμοποιώντας τη λειτουργία της R “diag”:

```
> diag(1,nrow=2)
      [,1] [,2]
[1,]    1    0
[2,]    0    1
```

Τα **ονόματα** των στηλών και των σειρών τοποθετούνται ως ακολούθως:

```
> rownames(A) <- c("x","y")
> colnames(A) <- c("c","b")
> A
   c b
x 1 3
y 2 4
```

Σημειώστε ότι χρησιμοποιούμε τη λειτουργία “c ()” εδώ! Τα ονόματα των σειρών και των στηλών είναι στην πραγματικότητα διανύσματα που περιέχουν χορδές.

Οι πίνακες μπορούν επίσης να δημιουργηθούν συνδυάζοντας (**καμπυλωτά**) διανύσματα, πχ με διάταξη σειράς

```
> V <- 0.5:5.5
> rbind(V,sqrt(V))
      [,1] [,2] [,3] [,4] [,5] [,6]
V 0.5000000 1.500000 2.500000 3.500000 4.500000 5.500000
 0.7071068 1.224745 1.581139 1.870829 2.121320 2.345208
```

t(A) θα **μετατοπίσει** τον πίνακα A (ανταλλάσσοντας σειρές και στήλες)

```
> t(A)
   x y
c 1 2
b 3 4
```

3.1.4 Συστοιχίες

Οι συστοιχίες είναι πολυδιάστατες γενικεύσεις των πινάκων· οι πίνακες και οι συστοιχίες στην R είναι στην πραγματικότητα διανύσματα με χαρακτηριστικά διάστασης. Μια πολυδιάστατη συστοιχία δημιουργείται ως ακολούθως:

```
> AR <- array(dim=c(2,3,2),data=1)
```

Σε αυτή την περίπτωση AR είναι μια συστοιχία $2 \times 3 \times 2$ και τα στοιχεία της είναι όλα 1.

3.2 Διαστάσεις

Οι εντολές:

```
> length(V)
> dim(A)
> ncol(M)
> nrow(M)
```

Θα επιστρέψουν το μήκος (συνολικός αριθμός στοιχείων) του (διανύσματος ή πίνακα) V, τη διάσταση του πίνακα ή της συστοιχίας A, και τον αριθμό των στηλών και των σειρών του πίνακα M αντιστοίχως.

3.3 Επιλέγοντας και αναιρώντας στοιχεία

Για να επιλέξουμε υποσύνολα διανυσμάτων ή πινάκων μπορούμε είτε

- Να καθορίσουμε τον αριθμό των στοιχείων που θέλουμε (απλή κατάταξη)
- Να καθορίσουμε ένα διάνυσμα με λογικές τιμές (TRUE/FALSE) για να δείξουμε ποια στοιχεία να συμπεριλάβουμε (TRUE) και ποια να μη συμπεριλάβουμε (FALSE). Αυτό χρησιμοποιεί λογικές προτάσεις

3.3.1 Απλή κατάταξη

Τα στοιχεία των διανυσμάτων, πινάκων και συστοιχιών κατατάσσονται χρησιμοποιώντας «[]»

```
M[1, 1]
M[1, 1:2]
M[1:3, c(2,4)]
```

Παίρνει το στοιχείο της πρώτης σειράς, πρώτη στήλη, ενός πίνακα M (1^η σειρά), έπειτα επιλέγει τις εισαγωγές στην πρώτη σειρά και τις δύο πρώτες στήλες (2^η σειρά) και μετά τα στοιχεία των τριών πρώτων σειρών και τα 2^η και 4^η στήλη του πίνακα M (3^η σειρά). Αν μια καταχώρηση παραληφθεί, τότε όλες οι σειρές (1^η κατάταξη παραλειπόμενη) ή οι στήλες (2^η κατάταξη παραλειπόμενη) επιλέγονται. Στο ακόλουθο

```
M[, 2] <- 0  
M[1:3, ] <- M[1:3, ] * 2
```

Πρώτον όλα τα στοιχεία στη 2^η στήλη (1^η σειρά) του M μηδενίζονται και έπειτα τα στοιχεία στις τρεις πρώτες σειρές του M πολλαπλασιάζονται επί 2 (2^η σειρά)

Παρόμοιες μέθοδοι επιλογής εφαρμόζονται στα διανύσματα:

```
V[1:10]  
V[seq(from=1,to=5,by=2)]
```

Η εντολή στην πρώτη σειρά, παίρνει τα πρώτα δέκα στοιχεία του διανύσματος V, ενώ στη δεύτερη σειρά, επιλέγονται το 1^ο, 3^ο και 5^ο στοιχείο του διανύσματος V

3.3.2 Λογικές προτάσεις

Οι λογικές προτάσεις χρησιμοποιούνται συχνά για την επιλογή στοιχείων από διανύσματα και πίνακες τα οποία ανταποκρίνονται σε ορισμένα κριτήρια. Η R διακρίνει τις λογικές μεταβλητές TRUE και FALSE, εκπροσωπούμενες από τους ακεραίους 1 και 0.

```
> ?Comparison  
> ?Logic
```

Θα κατατάξει τους σχετικούς και λογικούς φορείς που είναι διαθέσιμοι στη R. Η ακόλουθη εντολή θα δώσει TRUE για τις τιμές της ακολουθίας V που είναι θετικές:

```
> (V <- seq(-2,2,0.5))  
[1] -2.0 -1.5 -1.0 -0.5  0.0  0.5  1.0  1.5  2.0  
  
> V > 0  
[1] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
```

Ενώ

```
> V[V > 0]  
[1] 0.5 1.0 1.5 2.0
```

θα επιλέξει τις θετικές τιμές από το V,

```
> V[V > 0] <- 0
```

θα μηδενίσει όλα τα θετικά στοιχεία στο V,

```
> sum(V < 0)  
[1] 4
```

θα επιστρέψει τον αριθμό σε αρνητικά στοιχεία: αθροίζει τις TRUE (:1) τιμές, και

```
> V [V != 0]
[1] -2.0 -1.5 -1.0 -0.5  0.5  1.0  1.5  2.0
```

θα εκθέσει όλα τα μη μηδενικά στοιχεία από το V (“!” ισοδυναμεί με “not”) Λογικά τεστ μπορούν επίσης να συνδυαστούν, χρησιμοποιώντας «|» (ή εντολή “or”) και «&» (“and”)

```
> V [V<(-1) | V>1]
[1] -2.0 -1.5  1.5  2.0
```

θα εκθέσει όλες τις τιμές από το V που είναι <-1 και >1. Σημειώστε ότι έχουμε συμπεριλάβει το «-1» ανάμεσα σε αγκύλες (μπορείτε να καταλάβετε γιατί είναι απαραίτητο αυτό;) Τέλος,

```
> which (V == 0)
[1] 5

> which.min (V)
[1] 1
```

θα επιστρέψει τον κατάλογο των στοιχείων της τιμής 0 και των ελαχίστων.

3.4 Διαγράφοντας στοιχεία

Όταν ο κατάλογος προηγείται από ένα “-” το στοιχείο διαγράφεται

```
M[,-1]
```

θα δείξει τα περιεχόμενα του πίνακα M εκτός από την πρώτη στήλη

```
x <- x[-1]
M <- M[-1, ]
V <- V[-V>=0]
```

θα διαγράψει το 1^ο στοιχείο του x, (1^η σειρά) την 1^η σειρά του M (2^η σειρά) και τα θετικά στοιχεία του V (3^η σειρά)

Για περισσότερες πληροφορίες πληκτρολογήστε

```
> ?Extract
```

3.5 Πιο σύνθετες δομές δεδομένων

Η R επίσης επιτρέπει περισσότερες σύνθετες κατασκευές όπως πλαίσια δεδομένων και λίστες

3.5.1 Λίστες

Μια λίστα είναι ένας συνδυασμός από αρκετά αντικείμενα· κάθε αντικείμενο μπορεί να είναι διαφορετικού μήκους

```
> list(Array = AR, Matrix = M)
```

θα συνδυάσει την προκαθορισμένη συστοιχία AR και τον πίνακα M

3.5.2 Πλαίσια δεδομένων

Αυτά είναι συνδυασμοί από διαφορετικούς τύπους δεδομένων (πχ χαρακτήρες, ακέραιοι, λογικοί, πραγματικοί) τοποθετημένους σε πίνακοειδή μορφή:

```
> genus <- c("Sabatieria", "Molgolaimus")
> dens <- c(1,2)
> Nematode <- data.frame(genus=genus, density=dens)
> Nematode
```

	genus	density
1	Sabatieria	1
2	Molgolaimus	2

Στο παράθυρο παραπάνω, το πλαίσιο δεδομένων “Nematode” περιέχει δύο στήλες, μία με χορδές (το όνομα είδους), και μια με τιμές (τις πυκνότητες)

Τα πλαίσια δεδομένων είναι στην πραγματικότητα ειδικές περιπτώσεις από λίστες, αποτελούμενες από διανύσματα ίσου μήκους. Πολλές λειτουργίες των πινάκων λειτουργούν πάνω στα data.frames με έναν μοναδικό τύπο δεδομένων, αλλά υπάρχουν επίσης ξεχωριστές λειτουργίες πάνω στα data.frames.

3.5.3 Επιλέγοντας δεδομένα από data.frames και λίστες

Τα data.frames και οι λίστες μπορούν να γίνουν προσβάσιμα από τα **ονόματά** τους, ή από τις εντολές «`[]`» και «`[[]]`». Το αντικείμενο που θα προκύπτει από μια επιλογή απλής αγκύλης `[]`, θα αποτελεί ένα data.frame αντίστοιχως και μια λίστα το ίδιο· με διπλή αγκύλη `[[]]`, αποκτά κανείς ένα διάνυσμα (data.frames) ή ένα μεταβλητό τύπο δεδομένων (λίστες).

Για παράδειγμα:

```
> Nematode$density/sum(Nematode$density)
[1] 0.3333333 0.6666667
```

θα διαιρέσει όλες τις τιμές πυκνότητας (1,2) δια της αθροιστικής πυκνότητας

```
> mean(Nematode[,2])
```

θα υπολογίσει το μέσο της πυκνότητας του νηματοειδούς (2^η στήλη)

Δοκιμάστε:

```
> Nematode$genus
> Nematode[1]
> Nematode[[1]]
```

Αυτές οι εντολές όλες θα παράγουν τα δύο ονόματα του γένους, αλλά με διαφορετική διάταξη:


```
> ?Extract
```

θα εξηγήσει επίσης τους διάφορους τύπους με τους οποίους εξάγουμε στοιχεία από λίστες και `data.frames`.

3.6 Μετατροπή δεδομένων

Η μετατροπή από ένα τύπο δομών δεδομένων σε έναν άλλο μπορεί να γίνει εύκολα πχ

```
> as.data.frame(M)
> as.vector(A)
```

Αν δεν είστε σίγουροι για τον τύπο μπορείτε να γράψετε:

```
> is.data.frame(M)
> is.vector(A)
```

Ή μπορείτε να απεικονίσετε τα δεδομένα ως

```
> class(M)
```

3.7 Εισαγωγή δεδομένων από εξωτερικές πηγές

Σε όλα τα προηγούμενα παραδείγματα, τα δεδομένα εισήχθηκαν από την κονσόλα (ή από αρχείο σεναρίου) Υπάρχουν άφθονοι τρόποι να εισάγει κανείς δεδομένα από εξωτερικές πηγές. Περισσότερο συχνά, θα χρησιμοποιούμε τις λειτουργίες `read.table`, `read.csv` ή `read.delim` για να διαβάσουμε πίνακες ή πλαίσια δεδομένων γραμμένα σε πινακοειδή μορφή ως αρχεία κειμένου. Η R έχει επίσης πολλά ενσωματωμένα σύνολα δεδομένων. Ταξινομούνται ως:

```
> data()
```

3.8 Ασκήσεις

Η δημιουργία και επεξεργασία πινάκων και διανυσμάτων είναι απαραίτητη αν θέλουμε να χρησιμοποιήσουμε την R ως μαθηματικό εργαλείο. Αν και αυτό έχει εφαρμοστεί με αξιόπιστο τρόπο στην R, δεν είναι απλό για αρχάριους!

Η εξάσκηση αποτελεί τον καλύτερο δάσκαλο, γι' αυτό σας δίνονται πολλές ασκήσεις. Οι περισσότερες ασκήσεις μπορούν να λυθούν με μία μόνο εντολή της R. Παρ' όλα αυτά, καθώς αυτές οι εντολές μπορεί να είναι αρκετά πολύπλοκες, είναι συχνά απλούστερο πρώτα να διαιρεθούν σε μικρότερα μέρη, κι έπειτα να συγχωνευθούν σε μία εντολή.

3.8.1 Διανύσματα, ακολουθίες

Χρησιμοποιήστε την λειτουργία R “`mean`” για να υπολογίσετε το μέσο δύο αριθμών, του 9 και του 17 (ίσως παρατηρήστε ότι αυτό δεν είναι τόσο απλό όσο μπορεί να νομίζατε!)

Διανύσματα V

Χρήση της R για επιστημονικό προγραμματισμό

- Δημιουργήστε ένα διάνυσμα, αποκαλούμενο V, με ζυγούς αριθμούς ανάμεσα στο 16 και το 56. Μη χρησιμοποιήσετε βρόγχους. (συμβουλή: χρησιμοποιήστε την λειτουργία της R “seq”)
- Απεικονίστε αυτό το διάνυσμα
- Ποιο είναι το σύνολο όλων των στοιχείων του V; Μη χρησιμοποιήσετε βρόγχους, υπάρχει μία λειτουργία R που κάνει αυτό – το όνομα αυτής της λειτουργίας είναι συνηθισμένο.
- Απεικονίστε τα πρώτα 4 στοιχεία του V
- Υπολογίστε το γινόμενο των πρώτων 4 στοιχείων του V
- Απεικονίστε το 4^ο, 9^ο και 11^ο στοιχείο του V (συμβουλή: χρησιμοποιείτε τη λειτουργία «c()»)

Διάνυσμα w

- Δημιουργήστε ένα νέο διάνυσμα, w, το οποίο ισούται με το διάνυσμα V, πολλαπλασιασμένο επί 3· απεικονίστε το περιεχόμενό του
- Πόσα στοιχεία του w είναι μικρότερα από 100;
Πρώτα δημιουργήστε ένα νέο διάνυσμα που να περιέχει μόνο τα στοιχεία από w < 100 (ονομάστε το w₁₀₀), έπειτα υπολογίστε το μήκος αυτού του νέου διανύσματος.
- Τώρα κάντε τον ίδιο υπολογισμό, με μία εντολή R.
- Δημιουργήστε μια ακολουθία που περιέχει τις τιμές (1, 1/2, 1/3, 1/4, ..., 1/10)
- Υπολογίστε την τετραγωνική ρίζα κάθε στοιχείου
- Υπολογίστε το τετράγωνο κάθε στοιχείου
- Δημιουργήστε μία ακολουθία με τιμές (0/1, 1/2, 2/3, 3/4, ..., 9/10)

Διάνυσμα u

- Δημιουργήστε ένα διάνυσμα U, με 100 τυχαίους αριθμούς, ομοιόμορφα διανεμημένους ανάμεσα στο -1 και το 1. Συμβουλή: η εντολή της R “runif” παράγει ομοιόμορφα διανεμημένους τυχαίους αριθμούς. Χρησιμοποιήστε? runif να δείτε πως δουλεύει.
- Ελέγξτε το φάσμα του u· όλες οι τιμές θα πρέπει να είναι ανάμεσα στο -1 και 1. Συμβουλή: υπάρχει μία λειτουργία R που το κάνει αυτό, το όνομά της είναι συνηθισμένο. Υπολογίστε το άθροισμα και το γινόμενο των στοιχείων του u.
- Πόσα στοιχεία του U είναι θετικά;
- Μηδενίστε όλες τις αρνητικές τιμές του U.
- Ταξινομήστε το U.

Διανύσματα X,Y

- Δημιουργήστε 2 διανύσματα: το διάνυσμα X, με τα στοιχεία: 2,9,0,2,7,4,0 και το διάνυσμα Y με τα στοιχεία 3,5,0,2,5,4,6 (με αυτή τη σειρά) (συμβουλή: χρησιμοποιήστε τη λειτουργία c())
- Διαιρέστε όλα τα στοιχεία του Y με τα στοιχεία του X
- Πληκτρολογήστε τις ακόλουθες εντολές – προσπαθήστε να καταλάβετε:
- X>Y

- $X==0$
- Επιλέξτε όλες τις τιμές του Y που είναι μεγαλύτερες από τις αντίστοιχες τιμές του X
- Επιλέξτε όλες τις τιμές του Y για τις οποίες οι αντίστοιχες τιμές του X είναι 0.
- Αφαιρέστε όλες τις τιμές του Y για τις οποίες οι αντίστοιχες τιμές του X ισούνται με 0.
- Μηδενίστε όλα τα στοιχεία του X που είναι μεγαλύτερα ή ίσα με 7. Δείξτε το X.

3.8.2 Πίνακες

- Χρησιμοποιείτε τη λειτουργία της R “matrix” για να δημιουργήσετε έναν πίνακα με τα ακόλουθα περιεχόμενα

$$\begin{bmatrix} 3 & 9 \\ 7 & 4 \end{bmatrix}$$

- Απεικονίστε τον στην οθόνη
- Χρησιμοποιήστε τη λειτουργία της R “matrix” για να δημιουργήσετε έναν πίνακα με το όνομα «A»

$$\begin{bmatrix} 1 & 1/2 & 1/3 \\ 1/4 & 1/5 & 1/6 \\ 1/7 & 1/8 & 1/9 \end{bmatrix}$$

- Πάρτε το ανάστροφο του A
- Δημιουργήστε έναν νέο πίνακα B, αφαιρώντας τις δύο πρώτες σειρές και τις δύο πρώτες στήλες του A. Απεικονίστε τον στην οθόνη.
- Χρησιμοποιήστε “diag” για να δημιουργήσετε τον ακόλουθο πίνακα με το όνομα “D”.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

- Χρησιμοποιήστε “cbind” και “rbind” για να μεγενθύνετε αυτόν τον πίνακα, έτσι ώστε να αποκτήσετε:

$$\begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 2 & 0 & 4 \\ 0 & 0 & 3 & 4 \\ 5 & 5 & 5 & 5 \end{bmatrix}$$

είναι απλούστερο να το κάνετε αυτό με δύο εντολές (αλλά μπορεί να γίνει με μία!)

- Αφαιρέστε τη δεύτερη σειρά και τη δεύτερη στήλη του προηγούμενου πίνακα.

3.8.3 Ποικιλία των θαλάσσιων νηματοειδών

Θα εργαστούμε τώρα πάνω σε ένα σύνολο νηματοειδών αποτελούμενο από τις πυκνότητες των ειδών των νηματοειδών, που βρίσκονται στα ιζήματα σε μεγάλα βάθη της Μεσογείου, σε βάθη από 160 m έως 1220 m. Οι πυκνότητες εκφράζονται σε αριθμό ατόμων ανά 10 cm^2 .

Οι νηματοειδείς είναι μικρά (<1mm) σκουλήκια, και βρίσκονται γενικά σε αφθονία σε όλα τα θαλάσσια ιζήματα.

Τα δεδομένα (από Soetaert και άλλους, 1991) αρχικά παρουσιασμένα σε μια βάση δεδομένων ACCESS μεταφέρθηκαν σε μορφή πίνακα με τιμές διαχωρισμένες από κόμμα (αποκαλούμενη ως CSV)

Αυτή η μορφή έχει το πλεονέκτημα ότι μπορεί να διαβαστεί εύκολα από επεξεργαστές κειμένου (όπως ο Tinn-R) καθώς επίσης και από προγράμματα υπολογιστικών φύλλων.

Ανοίξτε το αρχείο "nemaspec.csv" στο Tinn-R². Ελέγξτε τη δομή του. Μπορείτε επίσης να ανοίξετε το αρχείο στο EXCEL, αλλά μην ξεχάσετε να το κλείσετε πριν προχωρήσετε. Το EXCEL καταναλώνει πολλούς πόρους και δε θα επιτρέψει σε άλλο πρόγραμμα, όπως η R, να αποκτήσει πρόσβαση σε αρχείο που έχει ανοίξει στο EXCEL.

Στην πρώτη γραμμή είναι ο τίτλος (τα ονόματα των σταθμών), η πρώτη στήλη περιέχει τα ονόματα των ειδών.

Πριν εισάγετε το αρχείο στην R, ελέγξτε τον φάκελο εργασίας:

```
> getwd()
```

Αν το αρχείο με όνομα "nemaspec.csv" δεν βρίσκεται σε αυτό το φάκελο, ίσως χρειαστεί να αλλάξετε το φάκελο εργασίας:

```
> setwd("directory name")
```

(θυμηθείτε ότι η R απαιτεί "/" όπου τα windows χρησιμοποιούν "\\")

Δημιουργήστε ένα αρχείο σεναρίου στο οποίο θα γράψετε τα επόμενα βήματα· υποβάλλετε κάθε γραμμή στην R για να ελέγξει την εγκυρότητά της.

Ανοίξτε το αρχείο που είναι οριοθετημένο με κόμμα, χρησιμοποιώντας την εντολή-R "read.csv". Πληκτρολογήστε ?read.csv για βοήθεια. Προσδιορίστε ότι πρώτη σειρά είναι η κεφαλίδα (header=TRUE) και η πρώτη στήλη περιέχει τα ονόματα των σειρών (row.names=1). Τοποθετήστε τα δεδομένα στο data.frame **Nemaspec**.

```
Nemaspec <- read.csv("nemaspec.csv", header=TRUE, row.names=1)
```

Ελέγξτε τα περιεχόμενα του Nemaspec. Καθώς το σύνολο δεδομένων είναι αρκετά σημαντικό, είναι καλύτερο να εξάγετε μόνο το πρώτο τμήμα των δεδομένων:

```
head(Nemaspec)
```

Τα υπόλοιπα είναι στο χέρι σας:

² Σημείωση: αν δεν έχετε αυτό το αρχείο, μπορεί να βρεθεί στον φάκελο του πακέτου "marelac" Μέσα από την R πληκτρολογήστε: `browseURL(paste(system.file(package="marelac"), "/lecture/", sep=" "))`

- Επιλέξτε τα δεδομένα από τον σταθμό M160b (η 2^η στήλη του Nemaspec)· τοποθετήστε αυτά τα δεδομένα αυτά σε ένα διάνυσμα καλούμενο “dens”.
(θυμηθείτε: για να επιλέξετε μία ολοκληρωμένη στήλη επιλέγετε όλες τις σειρές αφήνοντας τον πρώτο δείκτη κενό).
- Αφαιρέστε από τον διάνυσμα dens, τις πυκνότητες που είναι 0. Απεικονίστε αυτό το διάνυσμα στην οθόνη. (Απάντηση: [1] 6.580261 5.919719 κτλ...)
- Υπολογίστε το N, τη συνολική πυκνότητα των νηματωδών αυτού του σταθμού. Η συνολική πυκνότητα είναι απλώς το άθροισμα όλων των πυκνοτήτων ειδών (δηλαδή τι σύνολο των τιμών του διανύσματος dens). Ποια είναι η τιμή του N; (Απάντηση :699).
- Διαιρέστε τις τιμές στο διάνυσμα dens με την ολική πυκνότητα των νηματωδών N. Τοποθετήστε όλα τα αποτελέσματα στο διάνυσμα **p**, το οποίο τώρα περιέχει τις σχετικές αναλογίες για όλα τα είδη. Το άθροισμα όλων των τιμών στο p πρέπει τώρα να ισούται με 1. Ελέγξτε αυτό.
- Υπολογίστε το S, τον αριθμό των ειδών: αυτό είναι απλώς το μήκος του p· ονομάστε αυτήν την τιμή S. (Απάντηση: S=126)
- Υπολογίστε τις τιμές των δεικτών ποικιλότητας N1 και N2 και Ni, οι οποίες δίνονται από τους ακόλουθους τύπους:

$$N1 = e^{\sum -p_i \log_e(p_i)}$$

$$N2 = 1 / \left(\sum p_i^2 \right)$$

$$Ni = 1 / \max(p_i)$$

Μπορείτε να υπολογίστε καθεμία από αυτές τις τιμές χρησιμοποιώντας μόνο μία πρόταση R! (A: 90.15358, 66.77841, 22.56157)

- Τα 126 νηματοειδή είδη ανά 10 cm² ελήφθησαν με την εξέταση και τον 699 ατόμων. Φυσικά, όσο λιγότερα άτομα απαρτίζουν ένα είδος, τόσο λιγότερα είδη θα συναντήσουμε. Κάποιοι ερευνητές θεωρούν 100, άλλοι 200. Για να τυποποιήσουμε τα αποτελέσματά τους, ο προσδοκώμενος αριθμός των ειδών σε ένα δείγμα μπορεί να επαναυπολογιστεί με βάση έναν συνηθισμένο αριθμό ατόμων. Ο προσδοκώμενος αριθμός των ειδών σε ένα δείγμα με μέγεθος n, το οποίο έχει προέλθει από έναν πληθυσμό μεγέθους N, ο οποίος έχει S είδη δίνεται από:

$$ES(n) = \sum_{i=1}^S \left[1 - \frac{\binom{n-Ni}{n}}{\binom{N}{n}} \right]$$

οπού Ni είναι ο αριθμός των ατόμων στο i-στο είδος στο σύνολο του δείγματος και

$\binom{N}{n}$ είναι ο αποκαλούμενος “διωνυμικός συντελεστής», ο αριθμός των

διαφορετικών συνόλων με μέγεθος n τα οποία μπορούν να επιλεγθούν από ένα σύνολο με συνολικό μέγεθος N.

Στην R, οι διωνυμικοί συντελεστές υπολογίζονται με την πρόταση “choose(N,n)”.

Ποιος είναι ο προσδοκώμενος αριθμός ειδών ανά 100 άτομα;

(n=100,N=699).

(A: ES(100) = 60.68971).

Χρήση της **R** για επιστημονικό προγραμματισμό

- Εκτυπώστε όλους τους δείκτες ποικιλότητες στην οθόνη, οι οποίοι πρέπει να φαίνονται ως εξής:

N	N0	N1	N2	Ni	ESS
699.00000	126.00000	90.15358	66.77841	22.56157	60.68971

Κεφάλαιο 4

Λειτουργίες που ορίζονται από τον χρήστη

4.1 Ορισμός της λειτουργίας

Μία από τις δυνατότητες της R είναι ότι κάποιος μπορεί να φτιάξει λειτουργίες που ορίζονται από τον χρήστη που προστίθενται στις ενσωματωμένες λειτουργίες της R.

Τυπικά, σύνθετες λειτουργίες γράφονται σε αρχεία σεναρίου R, τα οποία μετά υποβάλλονται στην R (βλέπετε παραπάνω). Για παράδειγμα:

```
Circlesurface <- function (radius) pi*radius^2
```

ορίζει μια λειτουργία (η οποία ονομάζεται “Circlesurface”) η οποία δέχεται ως παράμετρο εισόδου μια μεταβλητή ονόματι “radius” και επιστρέφει την τιμή $\pi \cdot \text{radius}^2$ (η οποία είναι η επιφάνεια ενός κύκλου).

Αφού υποβάλλετε αυτή την λειτουργία στην R, μπορούμε να την χρησιμοποιούμε για να υπολογίζουμε τις επιφάνειες κύκλων με δοσμένη ακτίνα:

```
>Circlesurface(10)  
[1] 314.1593
```

ή

```
>Circlesurface(1:100)  
[1] 3.141593 12.566371 28.274334 50.265482 ...
```

το οποίο θα υπολογίσει τις επιφάνειες των κύκλων με ακτίνες 1,2, ..., 100.

Πιο σύνθετες λειτουργίες μπορεί να εξάγουν **περισσότερα από ένα στοιχεία**:

```
Sphere <- function(radius)  
{  
  volume <- 4/3*pi*radius^3  
  surface <- 4 *pi*radius^2  
  return(list(volume=volume,surface=surface))  
}
```

Εδώ αναγνωρίζουμε:

- Την κεφαλίδα της λειτουργίας (1^η γραμμή), η οποία καθορίζει το όνομα της λειτουργίας (Sphere) και την παράμετρο εισόδου (radius).
- Τις προδιαγραφές της λειτουργίας. Καθώς η λειτουργία εμπεριέχει πολλαπλές εντολές, οι προδιαγραφές της λειτουργίας εμπεριέχονται σε άγκιστρα «{...}».

- Οι τιμές που εξάγονται (τελευταία σειρά). Η λειτουργία Sphere θα εξάγει τον όγκο και την επιφάνεια μίας σφαίρας, σε μία λίστα.

Η Γη έχει προσεγγιστικά ακτίνα 6371 km , επομένως ο όγκος της (km^3) και η επιφάνειά της (km^2) είναι:

```
>Sphere(6371)
$volume
[1] 1.083207e+12
$surface
[1] 510064472
```

Η επόμενη εντολή θα παρουσιάσει μόνο τον όγκο των σφαιρών με ακτίνες 1, 2, ...,5

```
>Sphere(1:5)$volume
[1] 4.18879 33.51032 113.09734 268.08257 523.59878
```

Μερικές φορές είναι βολικό να παρέχουμε **προκαθορισμένες τιμές** για τις παραμέτρους εισόδου.

Για παράδειγμα, η επόμενη λειτουργία υπολογίζει την πυκνότητα του «τυπικού μέσου ωκεάνιου νερού» (σε kg m^{-3}), ως μια συνάρτηση της θερμοκρασίας, T, (και για αλμυρότητα=0, πίεση=1 atm) (Millero και Poisson, 1981); η παράμετρος εισόδου είναι προκαθορισμένη και ίση με 20°C:

```
Rho_W <- function(T=20)
{
  999.842594 + 0.06793952 * T - 0.00909529 * T^2 +
  0.0001001685 * T^3 - 1.120083e-06 * T^4 + 6.536332e-09 * T^5
}
```

Τελειώνοντας την πρώτη πρόταση με ένα «+» ξεκαθαρίσαμε ότι η εντολή δεν έχει τελειώσει και συνεχίζει στην επόμενη σειρά. (δείτε κεφάλαιο 2).

Όταν επικαλούμαστε τη λειτουργία χωρίς να προσδιορίζουμε την θερμοκρασία, αυτή χρησιμοποιεί την προκαθορισμένη τιμή:

```
>Rho_W()
[1] 998.2063

>Rho_W(20)
[1] 998.2063

>Rho_W(0)
[1] 999.8426
```

4.2 Προγραμματισμός

Η R έχει όλα τα στοιχεία μίας γλώσσας προγραμματισμού υψηλού επιπέδου

4.2.1 If, else, ifelse δομές:

Προσπαθήστε να κατανοήσετε τα ακόλουθα:

```
Dummy <- function (x)
{
  if ( x<0 ) string <- 'x<0'   else
  if ( x<2 ) string <- '0>=x<2' else
    string <- 'x>=2'
  print(string)
}

> Dummy(-1)
[1] "x<0"

> Dummy(1)
[1] "0>=x<2"

> Dummy(2)
[1] "x>=2"
```

Τώρα που έχουμε καθορίσει την συνθήκη “else” στην ίδια σειρά με το μέρος “if” έτσι ώστε η R να αντιληφθεί ότι η εντολή συνεχίζεται στην επόμενη σειρά!

Δομές if και else οι οποίες αφορούν μόνο μία εντολή μπορούν να συνδυαστούν:

```
> x<-2
> ifelse (x>0, "positive", "negative,0")
[1] "positive"
```

4.2.2 Βρόγχοι

Οι βρόγχοι επιτρέπουν σε ένα σύνολο εντολών να εκτελεστεί πολλαπλές φορές:

Ο βρόγχος “for” επαναλαμβάνει ένα καθορισμένο σύνολο τιμών. Στο παρακάτω παράδειγμα, η μεταβλητή “i” λαμβάνει τις τιμές (1,2,3):

```
> for (i in 1:3) print(c(i,2*i,3*i))
[1] 1 2 3
[1] 2 4 6
[1] 3 6 9
```

Τα “while” και “repeat” θα εκτελούν μέχρι να πληρείται μια καθορισμένη συνθήκη:

```
i<-1 ; while(i<11) {print(i); i<-i+1}
```

Το “break” οδηγεί σε έξοδο από τον βρόγχο

Το “next” σταματά την τρέχουσα επανάληψη και προχωρά στην επόμενη επανάληψη

```
i<-1
repeat
{
  print(i)
  i <-i+1
  if(i>10) break
}
```

Τα άγκιστρα «{...}» περικλείουν πολλαπλές εντολές οι οποίες εκτελούνται σε κάθε επανάληψη.

Σημείωση: Οι βρόγχοι υλοποιούνται πολύ αναποτελεσματικά και θα πρέπει να αποφεύγονται όσο το δυνατόν περισσότερο. Ευτυχώς, η R προσφέρει πολλές εντολές υψηλού επιπέδου οι οποίες λειτουργούν σε διανύσματα και πίνακες. Αυτές θα πρέπει να χρησιμοποιούνται όσο το δυνατόν περισσότερο!

Για περισσότερες πληροφορίες για δομές και βρόγχους με if, πληκτρολογήστε:

```
> ?Control
```

4.3 Πακέτα R

Ένα πακέτο στην R είναι ένα αρχείο το οποίο περιέχει πολλές λειτουργίες οι οποίες εκτελούν συγκεκριμένες σχετικές μεταξύ τους εργασίες. Τα πακέτα μπορούν να κατέβουν από τον ιστότοπο της R (δείτε κεφάλαια 1,2).

Εφόσον εγκατασταθούν, δημιουργούμε μια λίστα με όλα τα διαθέσιμα πακέτα, φορτώνουμε ένα πακέτο και αποκτούμε μία λίστα με τα περιεχόμενά του με τις ακόλουθες εντολές:

```
>library()  
>library(deSolve)  
>library(help=deSolve)  
>help(package=deSolve)
```

4.4 Ασκήσεις

4.4.1 Λειτουργία sphere της R

Επεκτείνετε τη λειτουργία “sphere” με την περίμετρο της σφαίρας στο σημείο της μέγιστης ακτίνας. Ο τύπος για τον υπολογισμό της περιμέτρου ενός κύκλου με ακτίνα r είναι $2\pi r$.

Ποια είναι η περίμετρος της Γης κοντά στον Ισημερινό;

4.4.2 Μια λειτουργία της R για να υπολογίζει συγκεντρώσεις των κορεσμένων οξυγόνων.

Η συγκέντρωση του κορεσμένου οξυγόνου στο νερό ($\mu\text{mol kg}^{-1}$), ως συνάρτηση της θερμοκρασίας (T), και της αλμυρότητας (S) μπορεί να υπολογιστεί από:

$\text{SatOx} = e^A$ όπου

$A = -173.9894 + 25559.07/T + 146.4813 \cdot \log_e(T/100) - 22.204 \cdot T/100 + S \cdot$

$(-0.037362 + 0.016504 \cdot T/100 - 0.0020564 \cdot T/100 \cdot T/100)$

και T είναι η θερμοκρασία Κέλβιν ($T_{\text{kelvin}} = T_{\text{celsius}} + 273.15$)

Εργασίες:

- Δημιουργήστε μία συνάρτηση που να εμπεριέχει αυτόν τον τύπο: οι προκαθορισμένες τιμές για τη θερμοκρασία και την αλμυρότητα είναι 20°C και 35 αντίστοιχα.
- Ποια είναι η συγκέντρωση του κορεσμένου οξυγόνου στις πρότυπες συνθήκες; ($A=225.2346$)
- Υπολογίστε τη συγκέντρωση του κορεσμένου οξυγόνου για ένα εύρος θερμοκρασιών από 0 έως 30°C, με αλμυρότητα 35 (Σημείωση: δεν χρειάζεται να χρησιμοποιήσετε βρόγχους).

4.4.3 Βρόγχοι

Οι αριθμοί Fibonacci υπολογίζονται από την ακόλουθη σχέση: $F_n = F_{n-1} + F_{n-2}$ με $F_1 = F_2 = 1$

Εργασίες:

- Υπολογίστε τους πρώτους 50 αριθμούς Fibonacci· αποθηκεύστε τα αποτελέσματα σε ένα διάνυσμα (χρησιμοποιήστε την εντολή R “vector” για να το δημιουργήσετε). Επιβάλλεται να χρησιμοποιήσετε βρόγχο εδώ.
- Για μεγάλο n , η αναλογία F_n/F_{n-1} τείνει στη «χρυσή τομή»: $(1 + \sqrt{5}) / 2$
- Ποια είναι η τιμή του F_{50}/F_{49} ; Είναι ίση με τη χρυσή τομή;
- Πότε είναι το n αρκετά μεγάλο; (δηλαδή ικανοποιητικά κοντά ($< e^{-6}$) στη χρυσή τομή;

4.4.4 Ποικιλότητα των νηματωδών σε βαθιά θάλασσα – όλοι οι σταθμοί

- Ξεκινώντας από τον κωδικό σας για να υπολογίσετε τους δείκτες ποικιλότητας για τον σταθμό σε βαθιά θάλασσα M160b, γράψτε έναν βρόγχο που κάνει το ίδιο για όλους τους σταθμούς στο Nemaspec.
- Πρώτα δημιουργήστε έναν πίνακα ονόματι **div**, με αριθμό σειρών ίσο με τον αριθμό των υποθαλάσσιων σταθμών, και με 6 στήλες, μία για κάθε δείκτη ποικιλότητας. Αυτός ο πίνακας θα περιέχει τις τιμές ποικιλότητας.
- Τα ονόματα των στηλών του div είναι: «N», «N0», «N1», «N2», «Ninf», «ESS». Τα ονόματα των σειρών του πίνακα div είναι τα ονόματα των σταθμών (= τα ονόματα των στηλών του Nemaspec). Σημείωση: Χρησιμοποιήστε την εντολή της R `colnames()`, `rownames()`
- Τώρα πραγματοποιήστε βρόγχο σε όλες τις στήλες του πλαισίου δεδομένων Nemaspec, υπολογίστε τους δείκτες ποικιλότητας και τοποθετήστε τα αποτελέσματα στην σωστή σειρά του πίνακα div:

```
for (i in 1:ncol(Nemaspec))
{
  # you have to write this part of the code
}
```

- Απεικονίστε τον πίνακα div στην οθόνη.

4.4.5 Δείκτες ποικιλότητας – Συνάρτηση

- Βασιζόμενοι στα αποτελέσματα που συλλέξατε στην άσκηση 4.4.4, δημιουργήστε μία συνάρτηση η οποία θα υπολογίζει τους δείκτες ποικιλότητας για κάθε πίνακα δεδομένων.

4.4.6. Ποικιλότητα αραίωσης (³)

Αν έχετε ακόμη χρόνο και κουράγιο: δοκιμάστε έναν εναλλακτικό τρόπο να υπολογίστε τον αριθμό των ειδών ανά 100 άτομα παίρνοντας τυχαία «υποδείγματα» των 100 ατόμων και υπολογίζοντας τον αριθμό των ειδών από αυτό το υπόδειγμα.

Αν η διαδικασία επαναλαμβάνεται αρκετά συχνά, η τιμή του αριθμητικού μέσου θα πρέπει να συγκλίνει στον προσδοκώμενο αριθμό ειδών, ESS(100) – αυτή είναι η μέθοδος αραίωσης του Sanders (1968).

Θα χρειαστείτε τις ακόλουθες νέες λειτουργίες της R:

- `round` (μετατρέπει πραγματικούς σε ακραίους),
- `cumsum` (δίνει ένα σωρευτικό ποσό)
- `sample` (κάνει μια τυχαία επιλογή στοιχείων)
- `table` (φτιάχνει έναν πίνακα μετρήσεων)

όπως και τις `length`, `mean`.

Ο Hurlbert (1971) έδειξε ότι η αραίωση γενικά υπερεκτιμά τον πραγματικά υπολογισμένο αριθμό ειδών – μπορείτε να επιβεβαιώσετε αυτή την ανακάλυψη;

³ Υποσημείωση Αυτή η ερώτηση απαιτεί σημαντική σκέψη και φαντασία· υπάρχουν αρκετοί τρόποι για να το επιτύχετε αυτό.

Κεφάλαιο 5

Στατιστική

Η R ξεκίνησε ως στατιστικό πακέτο, και ακόμα χρησιμοποιείται κυρίως για αυτό τον σκοπό. Μπορείτε να κάνετε πρακτικά οποιαδήποτε στατιστική ανάλυση στην R. Καθώς υπάρχουν πολλά αρχεία που μπορούν να σας βοηθήσουν με στατιστικές αναλύσεις στην R, δεν θα ασχοληθούμε με το αυτό το θέμα εδώ.

Απλώς χρησιμοποιούμε τα στατιστικά για να σας δείξουμε πώς να χρησιμοποιείτε την R αποτελεσματικά, σε περιπτώσεις που δεν έχετε κάποιο στοιχείο για το πώς να ξεκινήσετε!

5.1 Χρησιμοποιώντας την R σε 4 βήματα

Ας υποθέσουμε ότι θέλετε να εκτελέσετε μία ιεραρχική ομαδοποίηση και να σχεδιάσετε το δενδροδιάγραμμα ενός συνόλου δεδομένων με πολλές μεταβλητές.

Αν δεν το έχετε κάνει ποτέ στην R, εδώ είναι τα βήματα:

1. Βρείτε μία **εντολή** που να εκτελεί τη διαδικασία που επιζητείτε.
για παράδειγμα, χρησιμοποιήστε `help.search ("cluster")` για να λάβετε βοήθεια. Ανάλογα με τον αριθμό των πακέτων που έχετε εγκαταστήσει, η R θα σας παρουσιάσει μια λίστα από πιθανές εντολές των οποίων το αρχείο βοήθειας περιέχει τη λέξη "cluster" – χρησιμοποιήστε μια συνάρτηση από τα στατιστικά των πακέτων (τμήμα του πυρήνα της R).
2. Ανοίξτε το **αρχείο βοήθειας** (`?<name-of-the-function>`) και ελέγξτε τη σύνταξη για αυτή την εντολή. Αν δεν έχετε αρκετό χρόνο για να τη διαβάσετε πλήρως, διαβάστε τουλάχιστον (τμήματα από) τους τομείς "Description", "Usage" και "Examples".
3. Δοκιμάστε τα **παραδείγματα** στο αρχείο βοήθειας. Μπορείτε να:

Τα δοκιμάσετε όλα μαζί (`example(<name-of-the-function>)`).

Εναλλακτικά, μπορείτε να επιλέξετε τις εντολές στον τομέα Παραδείγματα οι οποίες εφαρμόζονται στο πρόβλημά σας, κάνετε αντιγραφή-επικόλληση στο αρχείο σεναρίου σας (`Ctrl-C/Ctrl-V`) και εκτελέστε τις – για παράδειγμα γραμμή ανά γραμμή. Είναι αρκετά πιθανό πώς, αν είναι κατάλληλες για την περίπτωσή σας, θα τις μετατρέψετε έτσι και αλλιώς.

4. **Μετατρέψτε** ένα υπάρχον παράδειγμα έτσι ώστε να ταιριάζει στο πρόβλημά σας.

5.2 Άσκηση: Στατιστική πολλών μεταβλητών στα δεδομένα ειδών νηματωδών

Χρησιμοποιήστε την R για να εκτελέσετε μια στατιστική ανάλυση πολλών μεταβλητών στα δεδομένα των νηματοειδών. Προσοχή: τα δεδομένα νηματοειδών έχουν τους σταθμούς ως στήλες και τα είδη σαν σειρές

- Εκτελέσετε μία ιεραρχική ομαδοποίηση και να σχεδιάσετε το δένδροδιάγραμμα
- Πραγματοποιήστε μία ανάλυση σε κύριες συνιστώσες (PCA) και σχεδιάστε τα αποτελέσματα – μπορείτε επίσης να επαναλάβετε την ανάλυση PCA, αφαιρώντας τους δύο πρώτους σταθμούς!

Κεφάλαιο 6

Γραφικά

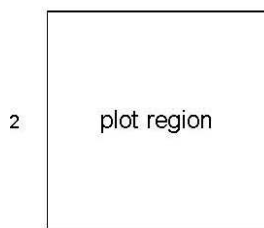
Η R έχει εκτεταμένες γραφικές δυνατότητες. Δοκιμάστε:

```
>demo(graphics)
>demo(image)
>demo(persp)
```

Για να αποκτήσετε μία παρουσίαση των δυνατοτήτων της R, είτε αυτές είναι απλές (1-D, x-y), μοιάζουν με εικόνα (2-D) ή έχουν προοπτική (3-D).

6.1 Βασικά

3



Τα γραφικά απεικονίζονται στο παράθυρο εικόνας που εμφανίζεται ανεξάρτητα από τα άλλα παράθυρα. Αν δεν είναι ακόμη ανοιχτό, μπορεί να

```
windows()
```

ανοίξει γράφοντας:

Μια εικόνα απαρτίζεται από μία περιοχή απεικόνισης η οποία περιβάλλεται από 4 περιθώρια, τα οποία είναι αριθμημένα, σύμφωνα με τη φορά του ρολογιού, από το 1 στο 4, ξεκινώντας από κάτω.

Η R διακρίνει τις εντολές σε:

1. εντολές υψηλού επιπέδου. Από προεπιλογή, αυτές δημιουργούν μία νέα εικόνα, για παράδειγμα:
hist, barplot, pie, boxplot... (1-D απεικόνιση)
plot, curve, matplot, pairs... (X-y απεικονίσεις)
image, contour, filled.contour... (2-D απεικονίσεις επιφάνειας)
persp, scatterplot3d⁴) (3-D απεικονίσεις)
2. εντολές χαμηλού επιπέδου οι οποίες προσθέτουν νέα αντικείμενα σε υπάρχουσες εικόνες, για παράδειγμα:

⁴ Η εντολή scatterplot3d είναι στο πακέτο R “scatterplot3d” το οποίο πρέπει να φορτωθεί προτού γίνει χρήση της.

lines, points, segments, polygon, rect, text, arrows, legend, abline, locator, rug...

Αυτές προσθέτουν αντικείμενα μέσα στην περιοχή απεικόνισης

box, axis, mtext (κείμενο στο περιθώριο), title...

οι οποίες προσθέτουν αντικείμενα στα περιθώρια της απεικόνισης

3. γραφικές παράμετροι οι οποίες ελέγχουν την εμφάνιση των:

αντικειμένων απεικόνισης:

cex (μέγεθος του κειμένου και των συμβόλων), col (χρώματα), font, las

(προσανατολισμός του υπομνήματος των αξόνων), lty (τύπος σειράς), lwd (πλάτος

σειράς), pch (τύπος των σημείων)

παραθύρου γραφικών:

mar (μέγεθος περιθωρίων), mfrow (αριθμός εικόνων σε μία σειρά), mfcol (αριθμός εικόνων σε μία στήλη)

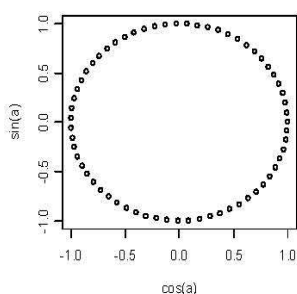
```
> ?plot.default
> ?par
> ?plot.window
> ?points
```

θα ανοίξει τα αρχεία βοήθειας, ενώ:

```
> example(plot.default)
> example(points)
```

θα εκτελέσει τα παραδείγματα, απεικονίζοντας κάθε νέο γράφημα, εφόσον έχετε πατήσει <ENTER> (δοκιμάστε το!)

6.2 x-y απεικονίσεις



Ένας κύκλος μπορεί να απεικονιστεί με σημεία (x,y) όπου $x=r*\cos(\alpha)$ και $y=r*\sin(\alpha)$ με α γωνία από 0 έως 2π και r ακτίνα.

Στο ακόλουθο σενάριο, πρώτα δημιουργούμε μια ακολουθία από τιμές της γωνίας, α , από 0 έως 2π , η οποία απαρτίζεται από 100 τιμές (length.out) και μετά απεικονίζουμε έναν κύκλο με ακτίνα:

```
a <- seq(0,2*pi, length.out=100)
plot(x=cos(a), y=sin(a))
```

Αφού η εντολή “plot” είναι υψηλού επιπέδου, δημιουργεί μία νέα εικόνα.

Από προεπιλογή, η R προσθέτει άξονες, και υπομνήματά τους, και αντιπροσωπεύει τα δεδομένα (x,y) ως μικρές κουκκίδες (σημεία). Παρατηρήστε ότι το γράφημα δεν είναι συμμετρικό:

Τώρα θα δημιουργήσουμε μία πιο περίπλοκη εικόνα η οποία θα μοιάζει με «στόχο» π.χ. για να εξασκούμαστε στην τοξοβολία ή να πετάμε βελάκια.

Αρχικά θα χρησιμοποιήσουμε την ίδια εντολή όπως παραπάνω (“plot”), αλλά θα προσθέσουμε έναν αριθμό γραφικών παραμέτρων οι οποίες θα εξασφαλίζουν ότι:

- Αντί για κουκκίδες, τα σημεία θα ενώνονται με γραμμές (type).
- Η γραμμή θα είναι διπλάσιου πλάτους από την προεπιλεγμένη (lwd).
- Τα υπομνήματα των αξόνων x,y θα πρέπει να είναι άδεια (xlab.ylab)
- Οι άξονες και οι σχολιασμοί των αξόνων θα αφαιρεθούν (axes)
- Το γράφημα θα πρέπει να είναι συμμετρικό, δηλαδή η αναλογία x/y=1 (asp).

```
plot(cos(a),sin(a),type="l",lwd=2,xlab="",ylab="",axes=FALSE,  
     asp=1)
```

Σε αυτή την εικόνα, μπορούμε τώρα να προσθέσουμε διάφορα αντικείμενα χαμηλού επιπέδου:

- Μία σειρά από γραμμές, οι οποίες θα αντιπροσωπεύουν μικρότερους κύκλους

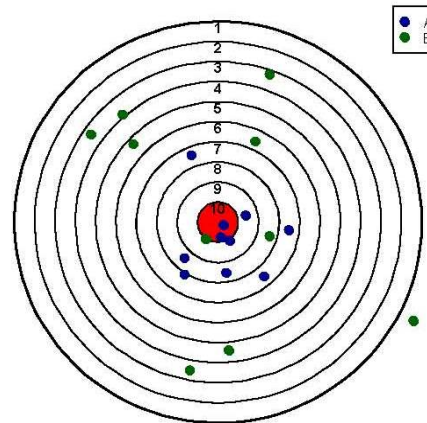
```
for (i in seq( 0.1,0.9,by=0.1)) lines(i*sin(a), i*cos(a))
```

(lines).

```
polygon(sin(a)*0.1,cos(a)*0.1,col="red")
```

- Ένα ενδότατο κόκκινο πολύγωνο (polygon).
- Η βαθμολογία σε πόντους θα αναπαρίσταται με ετικέτες κειμένου, οι οποίες θα κυμαίνονται από 10 έως 1 (text). Όσο πλησιέστερα στο κέντρο, τόσο μεγαλύτερο το σκορ.

```
for (i in 1:10) text(x=0,y=i/10-0.025,labels=11-i,font=2)
```



Τώρα, 2 τοξοβόλοι κάνουν 10 προσπάθειες στον στόχο. Θα μιμηθούμε τα βέλη τους δημιουργώντας (x,y) αριθμούς που ακολουθούν κανονική κατανομή, με μέση τιμή=0 (το κέντρο!) έτσι ώστε η εμπειρία κάθε τοξοβόλου να αναπαρίσταται από την τυπική απόκλιση. Όσο πιο έμπειρος είναι ο τοξοβόλος, τόσο πλησιέστερα θα είναι τα βέλη στο κέντρο, δηλαδή τόσο μικρότερη η τυπική απόκλιση.

- Η εντολή R `rnorm` δημιουργεί αριθμούς που ακολουθούν κανονική κατανομή· θα χρειαστούμε 20 τέτοιους, διατεταγμένους σε έναν πίνακα με 2 στήλες.

```
shots1 <- matrix(ncol=2, data=rnorm(n=20,sd=0.2))
shots2 <- matrix(ncol=2, data=rnorm(n=20,sd=0.5))
```

- Οι προσπάθειες προστίθενται στην απεικόνιση ως πόντοι, `points` αποκαλούμενοι `darkblue` (για τον έμπειρο τοξοβόλο) και `darkgreen` (για τον αρχάριο). Παρατηρήστε ότι χρησιμοποιούμε σημεία με 50% μεγέθυνση (`cex`), και ότι επιλέξαμε σημεία κυκλικού σχήματος (`pch=16`)

```
points(shots1,col="darkblue",pch=16,cex=1.5)
points(shots2,col="darkgreen",pch=16,cex=1.5)
```

- Τέλος, προσθέτουμε ένα υπόμνημα, εξηγώντας ποιος έκανε την κάθε προσπάθεια.

```
legend("topright",legend=c("A","B"),pch=16,
      col=c("darkblue","darkgreen"),pt.cex=1.5)
```

Παρατηρήστε ότι το κείμενο του υπομνήματος και τα χρώματα εισάγονται ως ένα διάνυσμα από χορδές, χρησιμοποιώντας την λειτουργία `c()` (π.χ. `c("A","B")`).

6.3 X-Y απεικονίσεις – υπό όρους αποτύπωση

Ως μία πιο εκλεπτυσμένη παρουσία της χρήσης των συμβόλων στα γραφήματα R, θα εργαστούμε σε ένα πιο βιολογικό παράδειγμα, από το σύνολο δεδομένων R αποκαλούμενο “Orange”.

Το σύνολο δεδομένων περιέχει την περιμέτρο (σε mm, στο ύψος του στήθους) η οποία έχει υπολογιστεί σε διαφορετικές ηλικίες για πέντε πορτοκαλιές. Ξεκινάμε κοιτώντας τα δεδομένα (μόνο ένα κομμάτι τους απεικονίζεται):

```
>Orange
```

	Tree	age	circumference
1	1	118	30
2	1	484	58
...			
8	2	118	33
9	2	484	69
...			
34	5	1372	174
35	5	1582	177

και κάνουμε μία προσεγγιστική απεικόνιση της περιμέτρου σε σχέση με την ηλικία:

```
plot(Orange$age, Orange$circumference, xlab="age,  
days", ylab="circumference, mm", main= "Orange tree growth")
```

(επειδή το Orange είναι data.frame, οι στήλες μπορούν να αναφέρονται με τα ονόματά τους, Orange\$age και Orange\$circumference).

Η έξοδος (στο παρακάτω σχήμα, αριστερά) δείχνει ότι υπάρχει μεγάλη διασπορά, εξαιτίας του γεγονότος ότι τα πέντε δέντρα δεν αναπτύχθηκαν με τον ίδιο ρυθμό.

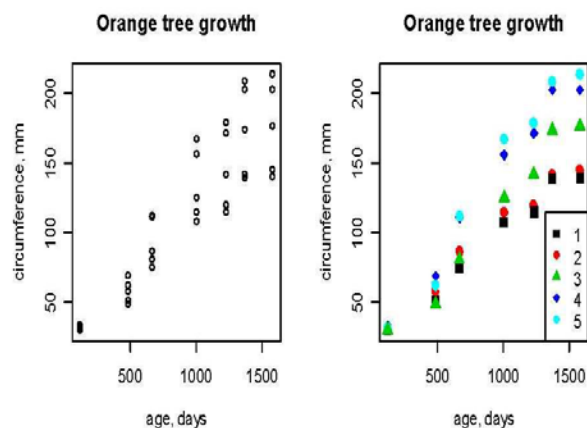
Είναι κατατοπιστικό να απεικονίσουμε τη σχέση μεταξύ περιμέτρου και ηλικίας διαφορετικά για κάθε δέντρο. Στην R, αυτό είναι εύκολο: μπορούμε να εμπεριέχουμε μερικές γραφικές παραμέτρους (τύποι συμβόλων, χρώματα, μέγεθος...) σε συνθήκη με κάποιους «**παράγοντες**». Οι παράγοντες παίζουν πολύ μεγάλο ρόλο στις στατιστικές εφαρμογές της R – για την εφαρμογή μας, είναι αρκετό να γνωρίζουμε ότι οι παράγοντες είναι ακέραιοι, ξεκινώντας από το 1.

Στην εντολή R παρακάτω, χρησιμοποιούμε απλά διαφορετικά σύμβολα (pch) και χρώματα (col) για κάθε δέντρο: pch=(15:20) [Orange\$Tree] σημαίνει ότι, ανάλογα με την τιμή του Orange\$Tree (δηλαδή με τον αριθμό του δέντρου), το σύμβολο (pch) θα πάρει την τιμή 15 (tree=1), 16 (tree=2),... 20 (tree=5). Col=(1:5) [Orange\$Tree] κάνει το ίδιο για το χρώμα του σημείου. Η τελική εντολή προσθέτει ένα υπόμνημα, τοποθετημένο κάτω δεξιά:

```
plot(Orange$age, Orange$circumference, xlab="age,
days", ylab="circumference, mm", main= "Orange tree growth",
pch=(15:20)[Orange$Tree], col=(1:5)[Orange$Tree], cex=1.3)
legend("bottomright", pch=15:20, col=1:5, legend=1:5)
```

Το εξαγόμενο (σχήμα δεξιά) δείχνει ότι το δέντρο 5 μεγαλώνει γρηγορότερα, ενώ το δέντρο 1 μεγαλώνει αργότερα.

(σημείωση: είναι επίσης μορφωτικό να εκτελέσετε τα παραδείγματα στο αρχείο βοήθειας Orange).



6.3.1 Ρυθμοί ανάπτυξης του ζωοπλαγκτόν

Το “Zoogrowth” από το πακέτο “marelac” είναι ένα σύνολο δεδομένων της βιβλιογραφίας, το οποίο συνέλεξε ο Hansen και άλλοι (1997) με μετρήσεις των μέγιστων ρυθμών ανάπτυξης του ζωοπλαγκτόν ως συνάρτηση του όγκου του σώματός τους. Εκτελέστε το παράδειγμα αυτού του συνόλου δεδομένων (θα πρέπει να έχετε φορτώσει το πακέτο “marelac” πρώτα):

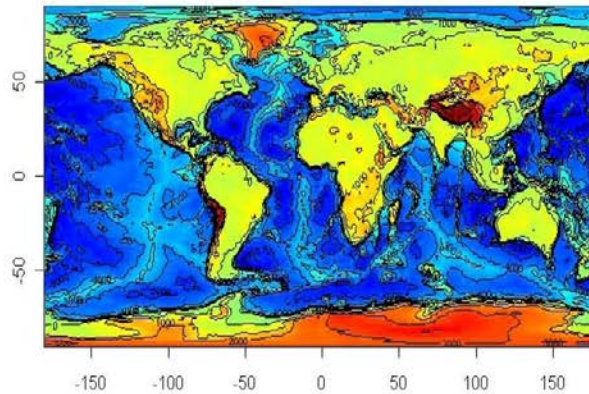
```
require(marelac)
example(Zoogrowth)
```

6.4 Ε ι κ ό ν ε ς κ α ι α π ε ι κ ο ν ί σ ε ι ς π ε ρ ι γ ρ ά μ μ α τ ο ς

Η R έχει κάποιες πολύ χρήσιμες λειτουργίες για να δημιουργεί εικόνες και να προσθέτει περιγράμματα. Για παράδειγμα, το σύνολο δεδομένων “Bathymetry” από το πακέτο marelac μπορεί να χρησιμοποιηθεί για να υπολογίσει την βαθυμετρία (και υψομετρία) των παγκόσμιων ωκεανών (και ξηράς):

```
require(marelac)
image(Bathymetry$x,Bathymetry$y,Bathymetry$z,col=femmecol(100),
      asp=TRUE,xlab="",ylab="")
contour(Bathymetry$x,Bathymetry$y,Bathymetry$z,add=TRUE)
```

Παρατηρήστε τη χρήση του “asp=TRUE”, το οποίο διατηρεί την αναλογία των διαστάσεων.



6.5 Απεικόνιση μίας μαθηματικής συνάρτησης

Γραφικές παραστάσεις για μαθηματικές συναρτήσεις μπορούν να δημιουργηθούν γρήγορα με την εντολή R “curve”:

```
curve(sin(3*pi*x))
```

θα απεικονίσει την γραφική παράσταση για $y=\sin(3\pi x)$, χρησιμοποιώντας τις προεπιλεγμένες ρυθμίσεις (σχ. αριστερά),

ενώ:

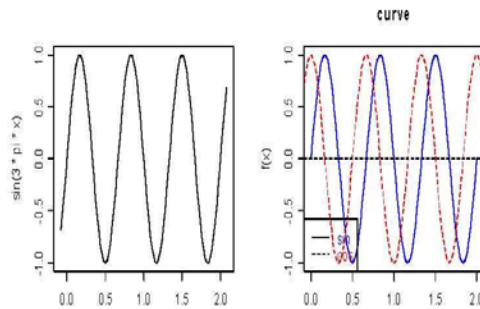
```
curve(sin(3*pi*x),from=0,to=2,col="blue",
      xlab="x",ylab="f(x)",main="curve")
curve(cos(3*pi*x),add=TRUE,col="red",lty=2)
```

Πρώτα σχεδιάζει τη γραφική παράσταση της $y=\sin(3\pi x)$, με μπλε χρώμα (col), και για τιμές του x που κυμαίνονται από 0 έως 2 (from, to), προσθέτοντας έναν κύριο τίτλο (main) και αρίθμηση για τους x και y άξονες. (xlab, ylab) (1^η πρόταση).

Η 2^η πρόταση της R προσθέτει τη συνάρτηση $y=\cos(3\pi x)$, ως μία κόκκινη (col) διακεκομμένη γραμμή (lty). Παρατηρήστε τη χρήση της παραμέτρου “add=TRUE”, αφού εκ προεπιλογής η εντολή curve θα δημιουργήσει μία νέα απεικόνιση.

Οι τελευταίες εντολές προσθέτουν τον x-άξονα, δηλαδή τον οριζόντιο, διακεκομμένο (lty=2), ως γραμμή (abline) για $y=0$ και ύστερα.

```
abline(h=0,lty=2)
legend("bottomleft",c("sin","cos"),text.col=c("blue","red"),lty=1:2)
```



6.6 Πολλαπλές εικόνες

Υπάρχουν διάφοροι τρόποι για να διατάξετε πολλαπλές εικόνες σε μία απεικόνιση:

1. Ο απλούστερος είναι καθορίζοντας τον αριθμό των εικόνων σε μία σειρά (**mfrow**) και σε μία στήλη (**mfc**):

```
par(mfrow=c(3,2))
```

Θα διατάξει τις επόμενες απεικονίσεις σε 3 σειρές και 2 στήλες. Τα γραφήματα θα απεικονίζονται με βάση τη σειρά.

```
par(mfcol=c(3,2))
```

Θα διατάξει τις απεικονίσεις σε 3 στήλες και 2 σειρές, σε ακολουθία με βάση τη στήλη.

Σημειώστε ότι και οι δύο εντολές **mfrow**, **mfc** πρέπει να εισαχθούν ως ένα διάνυσμα. Δοκιμάστε:

```
par(mfrow=c(2,2))
for ( i in 1:4) curve(sin(i*pi*x),0,1,main=i)
```

2. Η λειτουργία R “**layout**” επιτρέπει πολύ περισσότερες σύνθετες διατάξεις απεικόνισης.

6.7 Ασκήσεις

6.7.1 Απλές γραφικές παραστάσεις

- Δημιουργήστε ένα αρχείο σεναρίου το οποίο θα σχεδιάζει την γραφική παράσταση της συνάρτησης $y=x^3\eta\mu^2(3\pi x)$ στο διάστημα $[-2,2]$
- Δημιουργήστε την γραφική παράσταση της συνάρτησης $y=1/\text{συν}(1+x^2)$ στο διάστημα $[-5,5]$

6.7.2 Ανάπτυξη του ανθρώπινου πληθυσμού

Ο ανθρώπινος πληθυσμός (N , σε εκατομμύρια ατόμων) σε μία συγκεκριμένη στιγμή t , μπορεί να περιγραφεί ως μία συνάρτηση του χρόνου (t), της αρχικής πυκνότητας πληθυσμού για $t=t_0$ (N_{t_0}), τη φέρουσα ικανότητα « K » και το ρυθμό αύξησης “ a ” σύμφωνα με την επόμενη εξίσωση:

$$N(t) = \frac{K}{1 + \left[\frac{K - N_{t_0}}{N_{t_0}} \right] e^{-a(t-t_0)}} \quad (5)$$

Για τις ΗΠΑ, η πυκνότητα πληθυσμού το 1900 (N_0) ήταν 76.1 εκατομμύρια· η αύξηση του πληθυσμού μπορεί να εκφραστεί με τιμές παραμέτρων $a=0,02\text{yr}^{-1}$, $K=500$ εκατομμύρια άνθρωποι.

Οι πραγματικές τιμές του πληθυσμού ήταν:

1900	1910	1920	1930	1940	1950	1960	1970	1980
76.1	92.4	106.5	123.1	132.6	152.3	180.7	204.9	226.5

Εργασίες:

- Απεικονίστε την καμπύλη της πυκνότητας του πληθυσμού ως μία παχιά γραμμή, χρησιμοποιώντας τις τιμές παραμέτρου για τις ΗΠΑ.
- Προσθέστε τις καταμετρημένες τιμές του πληθυσμού σαν σημεία. Τελειώστε το γράφημα με τίτλους, ετικέτες κλπ.

6.7.3 Τοξική αμμωνία

Το άζωτο της αμμωνίας απαντάται σε δύο μορφές: το ιόν αμμωνίου (NH_4^+) και την μη ιονισμένη αμμωνία (NH_3). Επειδή η αμμωνία μπορεί να είναι τοξική σε αρκετά υψηλά επίπεδα, είναι συχνά επιθυμητό να γνωρίζουμε τη συγκέντρωσή της.

Η σχετική σημασία τα αμμωνίας (η συνεισφορά της αμμωνίας στο συνολικό άζωτο της αμμωνίας, $\text{NH}_3/(\text{NH}_3 + \text{NH}_4^+)$) είναι συνάρτηση της συγκέντρωσης πρωτονίων $[\text{H}^+]$ και μίας παραμέτρου K_N αποκαλούμενη στοιχειομετρική σταθερά ισορροπίας:

$$p_{[\text{NH}_3]} = \frac{K_N}{K_N + [\text{H}^+]}$$

⁵ Αυτή είναι η λύση της αποκαλούμενης διαφορικής εξίσωσης οργάνωσης (Verhulst, 1838)

Εργασίες:

- Απεικονίστε το σχετικό κλάσμα της τοξικής αμμωνίας προς τη συνολική συγκέντρωση της αμμωνίας ως μία συνάρτηση του pH, όπου $\text{pH} = -\log_{10}([\text{H}^+])$ και για θερμοκρασία 30°C. Χρησιμοποιήστε ένα εύρος του pH από 4 έως 9. Η τιμή της K_N είναι $8 \cdot 10^{-10}$ για θερμοκρασία 30°C.
- Προσθέστε σε αυτή την απεικόνιση το σχετικό κλάσμα της αμμωνίας για τους 0°C· η τιμή της K_N για αυτή τη θερμοκρασία είναι $8 \cdot 10^{-11} \text{ mol kg}^{-1}$.

6.7.4 Το σύνολο δεδομένων iris

Ένα διάσημο σύνολο δεδομένων που είναι τμήμα της R είναι το σύνολο δεδομένων “iris” (Fisher, 1936), το οποίο θα ερευνήσουμε στην συνέχεια.

Δίνει μετρήσεις, σε εκατοστά για το μήκος και το πλάτος του φύλλου κάλυκος και των πετάλων αντίστοιχα, για 50 άνθη των ειδών Iris setosa, Iris versicolor και Iris virginica.

Εργασίες:

- Κοιτάξτε τα δεδομένα
- Ποια είναι η τάξη αυτού του συνόλου δεδομένων; Γιατί;
- Ποιες είναι οι διαστάσεις αυτού του συνόλου δεδομένων; (αριθμός γραμμών, στηλών)
- Δημιουργήστε ένα διάγραμμα διασποράς του μήκους των πετάλων έναντι του πλάτους των πετάλων· δημιουργήστε έναν τίτλο που να πληροφορεί ικανοποιητικά, καθώς και αρίθμηση στους δύο άξονες.
- Επαναλάβετε το ίδιο γράφημα, χρησιμοποιώντας διαφορετικά χρώματα συμβόλων για τα τρία είδη.
- Προσθέστε ένα υπόμνημα στο γράφημα. Κάνετε αντιγραφή-επικόλληση το αποτέλεσμα σε ένα αρχείο WORD.
- Δημιουργήστε ένα σύρμα κατεύθυνσης για το μήκος των φύλλου κάλυκος όπου οι τιμές των δεδομένων είναι διαχωρισμένες σε είδη· χρησιμοποιήστε το πρότυπο ως πρώτο παράδειγμα στο αρχείο βοήθειας “boxplot”.
- Τώρα δημιουργήστε ένα παρόμοιο σύρμα κατεύθυνσης και για τις τέσσερις μορφολογικές μετρήσεις, διατεταγμένο σε 2 γραμμές και 2 στήλες. Πρώτα καθορίστε τη γραφική παράμετρο που διατάσσει τις απεικονίσεις 2 ανά 2.

Κεφάλαιο 7

Άλγεβρα πινάκων

Η άλγεβρα πινάκων είναι πολύ απλή στην R. Πρακτικά οτιδήποτε είναι εφικτό!

Εδώ είναι οι πιο σημαντικές λειτουργίες R που λειτουργούν σε πίνακες:

- `%*%` Πολλαπλασιασμός πινάκων
- `t(A)` μεταφορά του A
- `diag(A)` διαγώνιος του A
- `solve(A)` αντίστροφο του A
- `solve(A,B)` λύνει την $Ax=B$ ως προς x
- `eigen(A)` ιδιοτιμές και ιδιοδιανύσματα για το A
- `det(A)` ορίζουσα του A

Για παράδειγμα:

```
> A <- matrix(nrow=2, data=c(1,2,3,4))
> solve(A) %*% A
```

Θα αντιστρέψει τον πίνακα A (`solve(A)`) και πολλαπλασιάσει με A (`%*%`) δίνοντας τον πίνακα ενότητας:

```
      [,1] [,2]
[1,]    1    0
[2,]    0    1
```

Ενώ το `t(A)` θα μεταφέρει τον πίνακα A (αλλάζει μεταξύ τους γραμμές και στήλες)

```
> t(A)
     1 2
     3 4
```

Το επόμενο σύνολο εντολών θα επιλύσει το γραμμικό σύστημα $Ax=B$ ως προς το άγνωστο διάνυσμα x:

```
> B <- c(5,6)
> solve(A,B)
[1] -1  2
```

Τέλος, οι ιδιοσυχνότητες και τα ιδιοδιανύσματα του A υπολογίζονται χρησιμοποιώντας την λειτουργία της R “`eigen`”. Αυτή η λειτουργία εξάγει μία λίστα που περιέχει τόσο τις ιδιοτιμές (`$values`) όσο και τα ιδιοδιανύσματα (`$vectors`) (οι στήλες).

```
> eigen(A)
$values
[1] 5.3722813 -0.3722813

$vectors
      [,1]      [,2]
[1,] -0.5657675 -0.9093767
[2,] -0.8245648  0.4159736
```

7.1 Ασκήσεις

Άλγεβρα πινάκων 1

- Χρησιμοποιήστε την λειτουργία της R “matrix” για να δημιουργήσετε τους πίνακες που καλούνται «A» και «B»:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 6 & 4 & 1 \\ -2 & 1 & -1 \end{bmatrix}, B = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

- Λάβετε τον αντίστροφο πίνακα του A και τον μεταθετημένο πίνακα του A.
- Πολλαπλασιάστε τον A με τον B.
- Υπολογίστε τις ιδιοτιμές και τα ιδιοδιανύσματα του A.
- Για τον πίνακα A, το x είναι ιδιοδιάνυσμα και το λ ιδιοτιμή του αν $Ax = \lambda x$. Δοκιμάστε το!

Άλγεβρα πινάκων 2

- Δημιουργήστε έναν πίνακα, ονόματι P:

$$P = \begin{bmatrix} 0 & 0.0043 & 0.1132 & 0 \\ 0.9775 & 0.9111 & 0 & 0 \\ 0 & 0.0736 & 0.9534 & 0 \\ 0 & 0 & 0.0452 & 0.9804 \end{bmatrix}$$

- Ποια είναι η τιμή της μεγαλύτερης ιδιοτιμής (η λεγόμενη επικρατούσα ιδιοτιμή) και το αντίστοιχο ιδιοδιάνυσμα;
- Δημιουργήστε έναν νέο πίνακα, T, ο οποίος είναι ίσος με τον P, εκτός από την πρώτη γραμμή, όπου τα στοιχεία είναι 0.
- Τώρα υπολογίστε τον $N = (I - T)^{-1}$ όπου I είναι ο ταυτοτικός πίνακας.⁽⁶⁾

Σύστημα γραμμικών εξισώσεων

⁶ Υποσημείωση: Αυτό είναι το μοντέλο σταδίων για την φάλαινα-δολοφόνο (Caswell, 2001). Η ιδιοτιμή και τα ιδιοδιανύσματα υπολογίζουν το ρυθμό αύξησης και την στατική κατανομή ηλικίας, ο πίνακας N περιέχει το μέσο χρόνο που δαπανάται σε κάθε στάδιο.

- Λύστε το επόμενο σύστημα γραμμικών εξισώσεων για το άγνωστο x :

$$3x_1 + 4x_2 + 5x_3 = 0$$

$$6x_1 + 2x_2 + 7x_3 = 5$$

$$7x_1 + x_2 = 6$$

- Θα χρειαστεί να αναγράψετε πρώτα αυτό το πρόβλημα στη μορφή $Ax=B$, όπου το A περιέχει τους συντελεστές, x τους αγνώστους, και B είναι οι τιμές του δεξιού μέρους. Έπειτα μπορείτε να λύσετε το σύστημα χρησιμοποιώντας τη λειτουργία `R` “solve”.
- Ελέγξτε τα αποτελέσματα (δηλαδή, είναι $Ax=?B$)

Κεφάλαιο 8

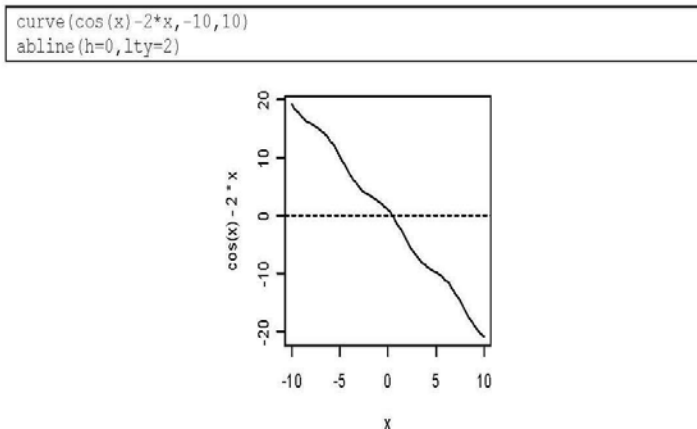
Ρίζες συναρτήσεων

8.1 Ρίζα μίας απλής συνάρτησης

Ας υποθέσουμε ότι θέλουμε να επιλύσουμε το παρακάτω πρόβλημα: $\cos x = 2x$ ως προς x

Μαθηματικά ψάχνουμε τη **ρίζα** της συνάρτησης $y = \cos x - 2x$, αυτή είναι η τιμή του x για την οποία $y=0$. Επειδή η συνάρτηση είναι αρκετά πολύπλοκη δεν είναι δυνατόν να βρούμε μία ακριβή λύση (μία συγκεκριμένη έκφραση) για αυτή τη ρίζα

Είναι πάντα προτιμότερο να σχεδιάσουμε την εξίσωση (1^η γραμμή), και να προσθέσουμε τον άξονα των x (2^η γραμμή).



Το σχήμα δείχνει ότι υπάρχει μια τιμή x , για την οποία $y=0$.

Τώρα η λειτουργία της R “uniroot” μπορεί να χρησιμοποιηθεί για να εντοπίσουμε αυτή την τιμή.

Συναρτήσεις που ζητούν μια ρίζα από μία μη γραμμική εξίσωση γενικά λειτουργούν «επαναληπτικά» δηλ. κινούνται όλο και πλησιέστερα προς τη ρίζα με διαδοχικά βήματα «επαναλήψεις».

Συνήθως δεν είναι εφικτό να βρεθεί η ρίζα ακριβώς, επομένως βρίσκεται κατά προσέγγιση, δηλ. με κάποια συγκεκριμένη προσέγγιση (tol (= ανοχή) έναν πολύ μικρό αριθμό) ⁽⁷⁾

Για να έχει η μέθοδος αποτέλεσμα, θα πρέπει να υπάρχει μία τουλάχιστον ρίζα στο διάστημα. Η παρακάτω εντολή θα επιλύσει ως προς τη ρίζα· δίνει διάφορες τιμές, σε μορφή λίστας

⁷ Ειδικότερα: η ρίζα του $y = \cos x - 2x$ είναι η τιμή του x για την οποία $|\cos x - 2x| < \text{tol}$ ή για την οποία οι διαδοχικές αλλαγές του x είναι $< \text{tol}$

```
>uniroot(f = function(x) cos(x)-2*x, interval=c(-10,10))
$root
[1] 0.4501686
$f.root
[1] 3.655945e-05
$iter
[1] 5
$estim.prec
[1] 6.103516e-05
```

Η πιο σημαντική τιμή είναι η ίδια η ρίζα, η οποία είναι 0.45103686· η τιμή της συνάρτησης στη ρίζα ήταν $3,66 \times 10^{-5}$, η συνάρτηση εκτέλεσε 5 επαναλήψεις.

Στο παράδειγμα, η συνάρτηση ήταν αρκετά απλή για να συμπεριληφθεί στην κλάση της “uniroot”. Το επόμενο κεφάλαιο δίνει ένα πιο πολύπλοκο παράδειγμα από την υδατική χημεία, όπου η προς επίλυση συνάρτηση είναι σημαντικά πιο πολύπλοκη

8.2 Ρίζα μίας πολύπλοκης συνάρτησης: επιλύοντας το pH.

Στα υδατικά διαλύματα, η ρυθμιστική ικανότητα των διαλυμάτων ανόργανων ανθρακικών ενώσεων (DIC)· διοξείδιο του άνθρακα (CO_2), διττανθρακικά ιόντα (HCO_3^-) και ανθρακικά ιόντα (CO_3^{2-}) διατηρούν το pH σε ένα στενό φάσμα. Η ρυθμιστική ικανότητα μετράται με την ολική αλκαλικότητα (TA, όπως καθορίζεται παρακάτω).

Εάν η αλκαλικότητα και η συγκέντρωση του ανόργανου άνθρακα (DIC) είναι γνωστά, είναι δυνατόν να υπολογίσουμε το pH (Park, 1969-Zeebe & Wolf-Gladrow, 2003), επιλύοντας τις παρακάτω εξισώσεις ως προς τη συγκέντρωση των άγνωστων πρωτονίων $[\text{H}^+]$.⁸

$$[\text{HCO}_3^-] = \frac{K_{c1} \cdot [\text{H}^+]}{[\text{H}^+] \cdot [\text{H}^+] + K_{c1} \cdot [\text{H}^+] + K_{c1} \cdot K_{c2}} \cdot \text{DIC} \quad (1)$$

$$[\text{CO}_3^{2-}] = \frac{K_{c1} \cdot K_{c2}}{[\text{H}^+] \cdot [\text{H}^+] + K_{c1} \cdot [\text{H}^+] + K_{c1} \cdot K_{c2}} \cdot \text{DIC} \quad (2)$$

$$\text{TA} = 2[\text{CO}_3^{2-}] + [\text{HCO}_3^-] - [\text{H}^+] \quad (3)$$

Να πως επιλύουμε ως προς τη συγκέντρωση πρωτονίων $[\text{H}^+]$ (ή την τιμή του pH στην R.

Το κόλπο είναι να υπολογίσουμε την αλκαλικότητα βασιζόμενοι σε μία εκτίμηση της συγκέντρωσης πρωτονίων χρησιμοποιώντας την εξίσωση (3) και συγκρίνοντάς την με την τιμή της αλκαλικότητας που μετρήσαμε. Εάν και οι δύο είναι ίσες, εντός των επιπέδων ανοχής, η συγκέντρωση πρωτονίων έχει βρεθεί.

⁸ Στην πράξη είναι δυνατόν να συγχωνεύσουμε τις 3 αυτές εξισώσεις σε μία, αλλά δεν είναι ούτε διδακτικά προφανές, ούτε προγραμματιστικά αποδοτικότερο.

Στην παρακάτω εφαρμογή οι σταθερές διάστασης για τα ανθρακικά άλατα (k_{c1} , k_{c2}) σε αλμυρότητα 0, θερμοκρασία 20 και πίεση 0 υπολογίζονται στο πακέτο R seacarb, τα οποία πρέπει να φορτωθεί πρώτα (require).

Εν συνεχεία ορίζουμε μία συνάρτηση της οποίας η ρίζα πρέπει να βρεθεί (pHfunction). Σε αυτή την συνάρτηση υπολογίζουμε την ολική αλκαλικότητα βασιζόμενοι στην κατ' εκτίμηση τιμή του pH, τις σταθερές διάστασης (k_{c1} , k_{c2}) και την συγκέντρωση DIC. Η διαφορά αυτής της κατ' εκτίμηση αλκαλικότητας (EstimatedAlk) με την πραγματική αλκαλικότητα ευρίσκεται· εάν το pH έχει εκτιμηθεί σωστά τότε η πραγματική και η κατ' εκτίμηση αλκαλικότητα θα είναι ίσες, και η διαφορά θα είναι μηδέν. Επομένως για να βρούμε το pH, πρέπει να βρούμε τη ρίζα της συνάρτησης.

Σημειώστε ότι η μετατροπή από το pH στο H^+ δίνει τη συγκέντρωση σε mol kg^{-1} . Επειδή οι συγκεντρώσεις των άλλων ουσιών είναι σε $\mu\text{mol kg}^{-1}$, μετατρέπονται χρησιμοποιώντας έναν παράγοντα 10^6 .

Περιορίζουμε την περιοχή της ρίζας του pH ανάμεσα στο 0 και το 12 (που είναι αρκετά μεγάλη) και θέτουμε την ανοχή (tol) σε έναν πολύ μικρό αριθμό για να αυξήσουμε την ακρίβεια.

```
require(seacarb)

kc1 <- K1(S=0,T=20,p=0) # Carbonate k1
kc2 <- K2(S=0,T=20,p=0) # Carbonate k2

pHfunction <- function(pH, kc1,kc2, DIC, Alkalinity )
{
  H <- 10^(-pH)
  HCO3 <- H*kc1 / (H*kc1 + H*H + kc1*kc2)*DIC
  CO3 <- kc1*kc2 / (H*kc1 + H*H + kc1*kc2)*DIC

  EstimatedAlk <- -H *1.e6 + HCO3 + 2*CO3
  return ( EstimatedAlk - Alkalinity)
}

uniroot(pHfunction, interval=c(0, 12), tol=1.e-20, kc1=kc1, kc2=kc2,
        DIC=2100, Alkalinity=2200)
```

8.3 Ασκήσεις

8.3.1 Απλές συναρτήσεις

- Βρες την ρίζα της εξίσωσης $e^x = 4x^2$ στο διάστημα $[0,1]$. Σχημάτισε πρώτα την καμπύλη της συνάρτησης.
- Επίλυσε τις εξισώσεις $1000 = \gamma * (3+x) * (1+\gamma)^4$ ως προς γ με το x να παίρνει τιμές από το 1 έως το 100. Σχεδίασε την ρίζα σαν συνάρτηση του x .
Βοήθεια: πρώτα φτιάξε μια σειρά των τιμών του x μετά κάνε ένα βρόγχο για κάθε τιμή x υπολογίζοντας την ρίζα και βάζοντάς την σε ένα διάνυσμα.

8.3.2 Χημεία: pCO_2 η αύξηση αυξάνει την οξύτητα

Το pH μπορεί επίσης να υπολογισθεί με βάση την μέτρηση της αλκαλικότητας και της pCO_2 , της μερικής πίεσης του CO_2 . Για την επίλυση αυτής της εξίσωσης, είναι απλούστερο να χρησιμοποιήσουμε έναν άλλο (ισοδύναμο) τρόπο για να μετρήσουμε τις σχέσεις ανάμεσα στα είδη των DIC.

$$[HCO_3^-] = K_{c1} \cdot \frac{[CO_2]}{[H^+]}$$

$$[CO_3^{2-}] = K_{c2} \cdot \frac{[HCO_3^-]}{[H^+]}$$

Το pCO_2 σχετίζεται με το $[CO_2]$ με την σταθερά του Henry, K_h , η οποία επίσης μπορεί να υπολογιστεί σαν μία συνάρτηση αλκαλικότητας, θερμοκρασίας και πίεσης, χρησιμοποιώντας το πακέτο R seacarb:

$$pCO_2 = \frac{[CO_2]}{K_h}$$

- Υπολογίστε το pH σε ισορροπία με αλκαλικότητα $230 \mu mol\ kg^{-1}$ και την τρέχουσα τιμή pCO_2 στα 360 ppm. Χρησιμοποιήστε το πακέτο seacarb για να υπολογίσετε τις σταθερές διάστασης και της σταθεράς Henry σε θερμοκρασία $20^\circ C$, αλμυρότητα 0 και πίεση 0
(A: $pH=8.19$)
- Το ενδοκυβερνητικό πάνελ για τις κλιματικές αλλαγές προβλέπει για το 2100 ατμοσφαιρική συγκέντρωση του CO_2 ανάμεσα στα 490 και 1250 ppmn, ανάλογα με το κοινωνικό-οικονομικό σενάριο (IPCC 2007). Αυτές οι αυξήσεις pCO_2 κάνουν το νερό πιο όξινο. Σχεδιάστε την καμπύλη pH σαν συνάρτηση των αυξημένων επιπέδων ατμοσφαιρικής pCO_2 . (Υποθέστε ότι η pCO_2 του ωκεανού βρίσκεται σε ισορροπία με την pCO_2 της ατμόσφαιρας). Ποια είναι η μέγιστη πτώση στο pH; (A: σε $pCO_2\ 1250\ ppmn$, το $pH\ 7,68$)

Κεφάλαιο 9

Παρεμβολή, ομαδοποίηση και προσαρμογή καμπύλης

9.1 Παρεμβολή και ομαδοποίηση

Η παρεμβολή και η ομαδοποίηση στην R μπορεί να γίνουν με διάφορους τρόπους.

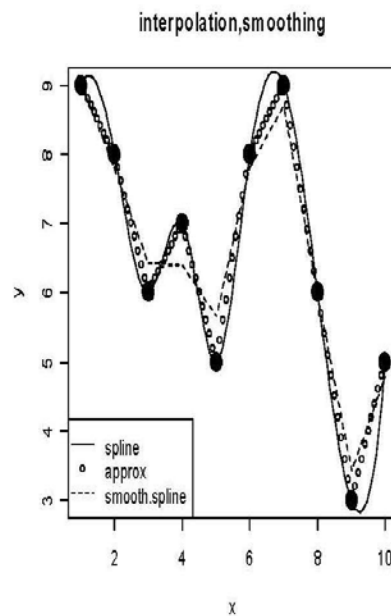
- `approx` γραμμική παρεμβολή ανάμεσα σε σημεία
- `spline` χρήση σφηνοειδούς παρεμβολής η οποία είναι ομαλότερη
- `smooth.spline` ομαδοποιεί ομάδες δεδομένων· αυτό σημαίνει ότι δεν συνδέει τα αρχικά σημεία

Η χρήση των συναρτήσεων αυτών δίνεται σαν παράδειγμα στο παρακάτω σενάριο και το αποτέλεσμα

```
x <- 1:10
y <- c(9,8,6,7,5,8,9,6,3,5)

plot(x,y,pch=16,cex=2,main="interpolation,smoothing")

lines (spline(x,y, n=100),lty=1)
points(approx(x,y, xout=seq(1,10,0.1)),pch=1)
lines (smooth.spline(x,y),lty=2)
legend("bottomleft",lty=c(1,NA,2),pch=c(NA,1,NA),
      legend=c("spline","approx","smooth.spline"))
```



9.2 Προσαρμογή καμπύλης

Η R έχει επίσης διάφορες διαδικασίες προσαρμογής καμπύλης. Ανάλογα με το εάν η συνάρτηση που πρόκειται να προσαρμοστεί είναι γραμμική ή όχι μπορείτε να χρησιμοποιήσετε:

Χρήση της R για επιστημονικό προγραμματισμό

- lm και glm για προσαρμογή γραμμικών μοντέλων και «γενικευμένων» γραμμικών μοντέλων
- nls, nlm, optim, constrOptim για μη γραμμικά μοντέλα

Σαν παράδειγμα τώρα θα προσαρμόσουμε τις τιμές της πυκνότητας του πληθυσμού των ΗΠΑ, σε 10-ετή διαστήματα, με το μοντέλο λογιστικής ανάπτυξης (κεφάλαιο 6.7.2) Το μοντέλο ήταν:

$$N(t) = \frac{K}{1 + \left[\frac{K - N_{t0}}{N_{t0}} \right] e^{-a \cdot (t - t_0)}} ,$$

1900	1910	1920	1930	1940	1950	1960	1970	1980
76.1	92.4	106.5	123.1	132.6	152.3	180.7	204.9	226.5

Ξεκινάμε εισάγοντας δεδομένα

```
year<- seq(1900,1980,by=10)
pop <- c(76.1,92.4,106.5,123.1,132.6,152.3,180.7,204.9,226.5)
```

Η απλούστερη μέθοδος για την προσαρμογή μιας μη γραμμικής καμπύλης είναι χρησιμοποιώντας την συνάρτηση R “nls”

Η συνάρτηση αυτή απαιτεί για είσοδο τον τύπο ($y \sim f(x, \text{παράμετροι})$) και τις αρχικές τιμές των παραμέτρων.

Στο παράδειγμα γ είναι οι τιμές πληθυσμού, f ο σχηματισμός λογιστικής ανάπτυξης. Σαν αρχική συνθήκη, χρησιμοποιούμε: $K=500$, $N_0=76.1$, $a=0.02$

```
fit <- nls(pop~K/(1+(K-N0)/N0*exp(-a*(year-1900))),
start=list(K=500,N0=76.1,a=0.02))
```

Ολοκληρώνουμε την προσαρμογή τυπώνοντας μια *περίληψη* των παραμέτρων προσαρμογής η οποία δείχνει τις εκτιμήσεις των παραμέτρων και τα σταθερά λάθη τους.

Προφανώς, δεν είναι δυνατόν να αποκτήσουμε αξιόπιστες εκτιμήσεις της τιμής του κ βασιζόμενοι στα δεδομένα

```
>summary(fit)
Formula: pop ~ K/(1 + (K - N0)/N0 * exp(-a * (year - 1900)))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
K  1.008e+03   8.932e+02   1.129  0.30209
N0 7.866e+01   2.531e+00  31.084 7.36e-08 ***
a  1.550e-02   2.505e-03   6.188  0.00082 ***
...
```

Οι ίδιες οι τιμές των συντελεστών, εξάγονται χρησιμοποιώντας τη λειτουργία R “coef”

```
>(cc<-coef(fit))
      K      NO      a
1.008226e+03 7.866365e+01 1.550344e-02
```

9.3 Ασκήσεις

Ομαλοποίηση

Ένα ανεμόμετρο μετρά την ταχύτητα του ανέμου σε τρία ωριαία διαστήματα. Μια συγκεκριμένη μέρα οι ταχύτητες αυτές είναι: 5,6,7,8,4,5,3,7,9 τις ώρες 0, 3, ... 24 αντίστοιχα. Για να υπολογίσουμε την αλλαγή του θαλασσίου αέρα χρειαζόμαστε ωριαίες μετρήσεις.

Στόχος:

- Παρεμβάλλετε ωριαίες μετρήσεις στις 3 ώρες
- Φτιάξετε μια καμπύλη με τις παρεμβαλλόμενες τιμές

Προσαρμογή

Η πρωταρχική παραγωγή μετράται με τον ^{14}C από δείγματα φυτοπλαγκτόν, σε διαφορετικές εντάσεις φωτός

Τα δεδομένα είναι:

```
II <- c(0.,1,10,20,40,80,120,160,300,480,700)
pp <- c(0.,1,3,4,6,8,10,11,10,9,8)
```

Προσαρμόστε τις εκτιμήσεις παραγωγής (pp) σαν συνάρτηση της έντασης του φωτός (II) με τις 3 παραμέτρους της εξίσωσης Eilers-Peeters.

Η πρωταρχική παραγωγή υπολογίζεται ως:

$$pp = p_{\max} \cdot \frac{2 \cdot (1 + \beta) \cdot \frac{I}{I_{opt}}}{\left(\frac{I}{I_{opt}}\right)^2 + 2 \cdot \beta \cdot \frac{I}{I_{opt}} + 1}$$

Όπου I είναι το φως και p_{\max} , β και I_{opt} είναι παράμετροι

- Πρώτα σχεδιάστε την αρχική παραγωγή (pp) έναντι του φωτός (II). Χρησιμοποιήστε μεγάλα σύμβολα
- Εν συνεχεία χρησιμοποιήστε τη λειτουργία της R “nls” για να προσαρμόσετε το μοντέλο στα δεδομένα

- Προσθέστε την βέλτιστη καμπύλη στο γράφημα (σημ: χρησιμοποιήστε `coef` για να πάρετε τις καλύτερες τιμές των παραμέτρων)

Κεφάλαιο 10

Διαφορικές εξισώσεις

Οι διαφορικές εξισώσεις εκφράζουν τον ρυθμό μεταβολής ενός παράγοντα σε μία ή περισσότερες διαστάσεις, συνήθως χρόνο και/ή διάστημα.

Ας λάβουμε το ακόλουθο σύστημα δύο διαφορικών εξισώσεων:

$$\begin{aligned}\frac{dA}{dt} &= r \cdot (x - A) - k \cdot A \cdot B \\ \frac{dB}{dt} &= r \cdot (y - B) + k \cdot A \cdot B\end{aligned}$$

- Τα A και B ονομάζονται διαφορικές μεταβλητές (ή μεταβλητές κατάστασης)
- $\frac{dA}{dt}$ είναι η παράγωγος (ή ο ρυθμός μεταβολής).
- r, x, y και k είναι παράμετροι (σταθερές).

Για να επιλύσουμε συστήματα διαφορικών εξισώσεων στην R ορίζουμε μια συνάρτηση (εδώ ονομάζεται `model`) η οποία έχει ως δεδομένα τον χρόνο (t), τις τιμές των μεταβλητών κατάστασης (`state`) και τις τιμές των παραμέτρων (`pars`). Η συνάρτηση αυτή απλά υπολογίζει τον ρυθμό μεταβολής των μεταβλητών κατάστασης (dA και dB) και τις δίνει σαν αποτέλεσμα σε μία λίστα.

Η εντολή της R “`with (as.list (c(state, pars)), { }`” επιτρέπει στις μεταβλητές κατάστασης και τις παραμέτρους να καλούνται με το όνομά τους.

```
model <- function(t,state,pars)
{
  with (as.list(c(state,pars)),
  {
    dA <- r*(x-A)-k*A*B
    dB <- r*(y-B)+k*A*B
    return (list(c(dA,dB)))
  }
)
```

Πριν λύσουμε το μοντέλο αυτό,

- δημιουργούμε μια σειρά τιμών χρόνων στους οποίους θέλουμε δεδομένα (`times`).
- αναθέτουμε αρχικές συνθήκες στις μεταβλητές κατάστασης (`state`) και
- δίνουμε τιμές στις παραμέτρους (`parms`):

```
times <- seq(0,300,1)
state <- c(A=1,B=1)
parms <- c(x=1, y = 0.1, k = 0.05, r = 0.05)
```

Το μοντέλο μπορεί τώρα να επιλυθεί. Για να το κάνουμε χρησιμοποιούμε τη ρουτίνα ολοκλήρωσης της R ode· η οποία μπορεί να βρεθεί στο πακέτο-R `deSolve`. Φορτώνουμε αυτό το πακέτο πρώτα

```
require(deSolve)
```

Για κάθε χρόνο t , “ode” θα καλεί τη λειτουργία “model”, με τις τρέχουσες τιμές των μεταβλητών κατάστασης και των τιμών των παραμέτρων.

Το αποτέλεσμα αποθηκεύεται σε ένα `data.frame`, που ονομάζεται “out”

```
out <- as.data.frame(ode(state,times,model,parms))
```

Το μόνο που πρέπει να κάνουμε τώρα είναι να σχεδιάσουμε το μοντέλο των αποτελεσμάτων. Πριν το κάνουμε ας ρίξουμε μια ματιά στο `data.frame` out:

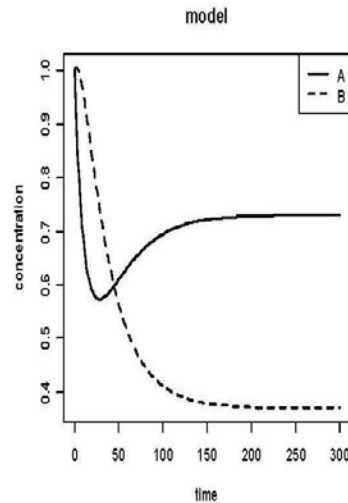
```
>head(out)
  time      A      B
1  0 1.0000000 1.0000000
2  1 0.9523189 1.0037869
3  2 0.9090698 1.0052847
4  3 0.8699233 1.0047146
...
```

Τα δεδομένα ταξινομούνται σε τρεις στήλες: πρώτα ο χρόνος, μετά οι συγκεντρώσεις των A και B. Επειδή το out είναι πλαίσιο δεδομένων μπορούμε να εξάγουμε τα δεδομένα χρησιμοποιώντας τα ονόματά τους (`out$A`, `out$B`).

Πριν σχεδιάσουμε το μοντέλο των αποτελεσμάτων, εκτιμάται το εύρος των συγκεντρώσεων των A και B: αυτό χρησιμοποιείται για να θέσουμε τα όρια του άξονα των y (`ylim`).

Η συνάρτηση R `plot` δημιουργεί ένα καινούριο γράφημα· το `lines` προσθέτει μια γραμμή στο γράφημα· Η `lty` επιλέγει ένα είδος γραμμής· Το `lwd=2` κάνει τη γραμμή διπλάσια σε πάχος από την προκαθορισμένη. Τέλος προσθέτουμε ένα υπόμνημα.

```
ylim <- range(c(out$A,out$B))
plot(out$time,out$A,xlab="time",ylab="concentration",
     lwd=2,type="l",ylim=ylim,main="model")
lines(out$time,out$B,lwd=2,lty=2)
legend("topright",legend=c("A","B"),lwd=2,lty=c(1,2))
```



10.1.1 Ασκήσεις

Μοντέλο Lotka-volterra ⁹

- Γράψτε ένα αρχείο σεναρίου που επιλύει το ακόλουθο σύστημα ODE:

$$\frac{dx}{dt} = a \cdot x \cdot \left(1 - \frac{x}{K}\right) - b \cdot x \cdot y$$
$$\frac{dy}{dt} = g \cdot b \cdot x \cdot y - e \cdot y$$

με αρχικές τιμές $x=300$, $y=100$ και τιμές παραμέτρων $a=0,05$, $\kappa=500$, $b=0,0009$, $g=0,8$, $e=0,03$.

- Φτιάξτε τρεις παραστάσεις, μία για το x και μία για το y , σε συνάρτηση του χρόνου, και μία εκφράζοντας το y σαν συνάρτηση του x (αυτό ονομάζεται γράφημα phase-plane). Ταξινομήστε τα σε δύο στήλες και δύο γραμμές.
- Τώρα τρέξτε το μοντέλο με άλλες αρχικές τιμές ($x=200$, $y=50$): προσθέστε τις γραφικές παραστάσεις (x,y) στο γράφημα phase-plane.

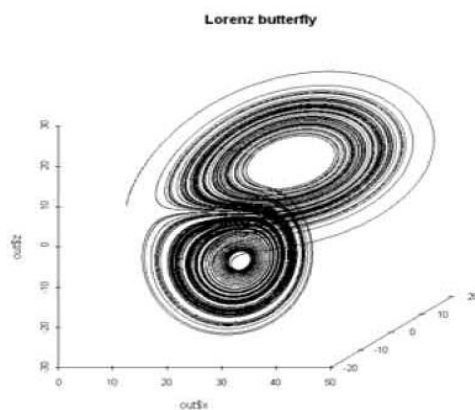
⁹ Τα μοντέλα Lotka-volterra είναι ένα διάσημο είδος μοντέλων που είτε περιγράφουν αλληλεπιδράσεις θύτη-θύματος ή ανταγωνιστικές αλληλεπιδράσεις ανάμεσα σε δύο είδη. Οι A.J. Lotka και V. Volterra σχημάτισαν το αρχικό μοντέλο την δεκαετία του 1920 σχεδόν ταυτόχρονα. (Lotka, Volterra, 1926).

Πεταλούδα

Οι εξισώσεις Lorenz (Lorenz, 1963) ήταν το πρώτο χαοτικό σύστημα διαφορικών εξισώσεων που δημιουργήθηκε για να αναπαραστήσει την εξιδανικευμένη εφαρμογή της ατμόσφαιρας της γης.

$$\begin{aligned}\frac{dx}{dt} &= -\frac{8}{3} \cdot x + y \cdot z \\ \frac{dy}{dt} &= -10 \cdot (y - z) \\ \frac{dz}{dt} &= -x \cdot y + 28y - z\end{aligned}$$

- Χρειάζονται περίπου 10 γραμμές κώδικα R για να παράγουμε τις λύσεις και να τις σχεδιάσουμε.
- Η συνάρτηση “scatterplot3d” από το πακέτο scatterplot3d δημιουργεί 3-D γραφικά. Μπορείτε να αναπαραστήσετε την παρακάτω «πεταλούδα»; Χρησιμοποιήστε σαν αρχικές συνθήκες $x=y=z=1$. βγάλτε δεδομένα για μια χρονική σειρά με εύρος από το 0 μέχρι το 100, με χρονικό διάστημα 0,005.



Κεφάλαιο 11

Τελικά

11.1 Ερωτήσεις

Οι σημειώσεις αυτές έχουν δημιουργηθεί στο WORD (Microsoft). Εάν δεν σας ικανοποιεί η παρουσίαση (ή δεν σας αρέσει το WORD), μια διαφορετική έκδοση υπάρχει σαν βινιέτα στο πακέτο “marelac”. Η βινιέτα αυτή έχει δημιουργηθεί με το Latex και το πακέτο R “Sweave” (Leisch, 2002), το οποίο επιτρέπει να ενσωματώσετε Latex με κώδικα R.

Μέσα από την R, πληκτρολογήστε:

```
> vignette("UsingR")
```

Ή μπορείτε να βρείτε το αρχείο “UsingR.pdf” στον υποφάκελο \inst\doc του πακέτου “marelac”.

Οι απαντήσεις στις ερωτήσεις του κύκλου μαθημάτων υπάρχουν σαν μία βινιέτα R στο πακέτο “marelac”.

Μέσα από την R, πληκτρολογήστε:

```
> vignette("Answers")
```

Ή, μπορείτε να βρείτε το αρχείο “Answers.pdf” στον υποφάκελο \inst\doc του πακέτου “marelac”.

Κεφάλαιο 12

Βιβλιογραφία

Caswell, H. 2001. Matrix population models: construction, analysis, and interpretation. Second edition. Sinauer, Sunderland, Mass.

Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7, Part II, 179-188.

Hurlbert, S. H. (1971). The nonconcept of species diversity: critique and alternative parameters. Ecology 52: 577-586

IPCC. 2007. Climate change 2007. Report of the IPCC, Geneva, Switzerland.

Kuhnert, P. and Venables, P. 2005. An introduction to R: software for statistical modelling & computing. CSIRO Australia. 364 pp. Downloadable from www.rproject.org.

Leisch, F., 2002, Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis. In: eds W. Hardle and Bernd Ronz, Compstat 2002 -Proceedings in Computational Statistics, Physica Verlag, Heidelberg: 575-580.

- Ligges, U. and Mächler, M. (2003). Scatterplot3d - an R Package for Visualizing Multivariate Data. *Journal of Statistical Software* 8(11), 1-20.
- Lotka, A. J. 1925. Elements of physical biology. Baltimore: Williams & Wilkins Co.
- Lorenz, EN, (1963). Deterministic non-periodic flows. *J. Atmos. Sci* 20, 130-141.
- Millero, F.J & Poisson, A. 1981. International one-atmosphere equation of state for seawater. *Deep-Sea Research Vol28A No.6*. 625-629.
- Proye, A., Gattuso, J-P, Epitalon, J-M, Gentili, B, Orr, J. and Soetaert, K. 2007. seacarb: an R-package to calculate parameters of the seawater carbonate system. R package version 1.2.1. <http://www.obs-vlfr.fr/~gattuso/seacarb.php>
- R Development Core Team (2007). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-90005107-0, URL <http://www.R-project.org>.
- Sanders, H. L. (1968). Marine benthic diversity: a comparative study. *Am. Nat.* 102: 243-282
- Soetaert, K., C. Heip & M. Vincx, 1991. Diversity of nematode assemblages along a mediterranean deep-sea transect. *Mar. Ecol. Progr. Ser.* 75: 275-282.
- Soetaert, K. and Herman, PMJ. 2008. A practical guide to ecological modelling. Using R as a simulation platform. Springer.
- Soetaert, K., Petzoldt, T. and R. Woodrow Setzer (2008). deSolve: General solvers for ordinary differential equations (ODE) and for differential algebraic equations (DAE). R package version 1.1.
- Soetaert, K. (2008). rootSolve: Nonlinear root finding, equilibrium and steady-state analysis of ordinary differential equations. R package version 1.1.
- Soetaert, K. and Meysman, F. (2008). marelac: Datasets, constants, conversion factors, utilities for the marine and lacustrine sciences. R package version 1.1.
- Verhulst, P. F. 1838. Notice sur la loi que la population poursuit dans son accroissement. *Correspondance mathématique et physique* 10:113-121.
- Volterra, V. 1926. Variazioni e fluttuazioni del numero d'individui in specie animali conviventi. *Mem. R. Accad. Naz. dei Lincei. Ser. VI, vol. 2*.
- Zeebe, R & Wolf-Gladrow, D.. 2003. CO₂ in Seawater: Equilibrium, kinetics, isotopes. Elsevier.