ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Γλώσσες Προγραμματισμού

Γραφικές παραστάσεις - R

Χαρά Πρασσά, chprassa@gmail.com

Acknowledgments

- Οι διαφάνειες αντιστοιχούν στο 5ο κεφάλαιο του βιβλίου ‘Ανάλυση δεδομένων με την R. Νικολάου, Χ., Εκδόσεις Δίσιγμα, 2019’.

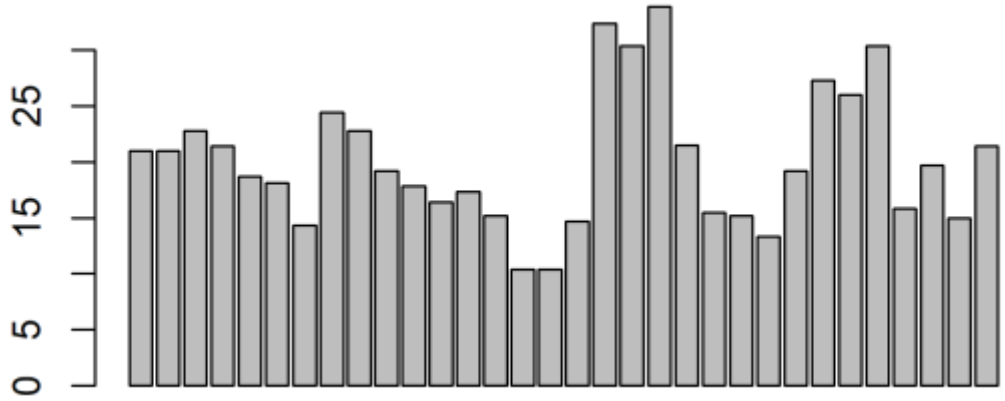
Ραβδογράμματα με την barplot()

Ο τρόπος δημιουργίας γραφημάτων στην R διαφέρει από τα προγράμματα που έχετε χρησιμοποιήσει μέχρι τώρα (Excel ή Matlab) με την έννοια ότι δεν βασίζεται σε ένα γραφικό περιβάλλον αλλά στη γραμμή εντολών.

1. Δημιουργία απλού ραβδογράμματος σε ένα μικρό σετ δεδομένων από το dataframe mtcars που μπορούμε να φορτώσουμε από τη βασική έκδοση της R:

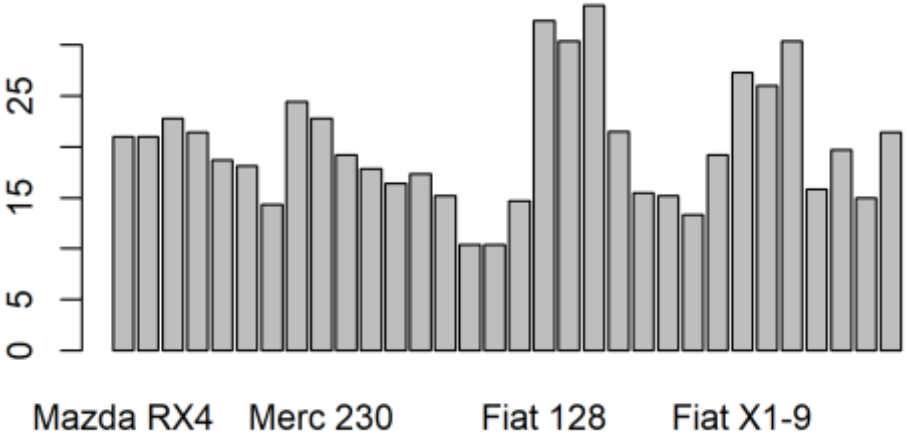
```
> str(mtcars)
'data.frame':   32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

> barplot(mtcars$mpg)



2. Τίτλος δεδομένων στον οριζόντιο άξονα (names=):

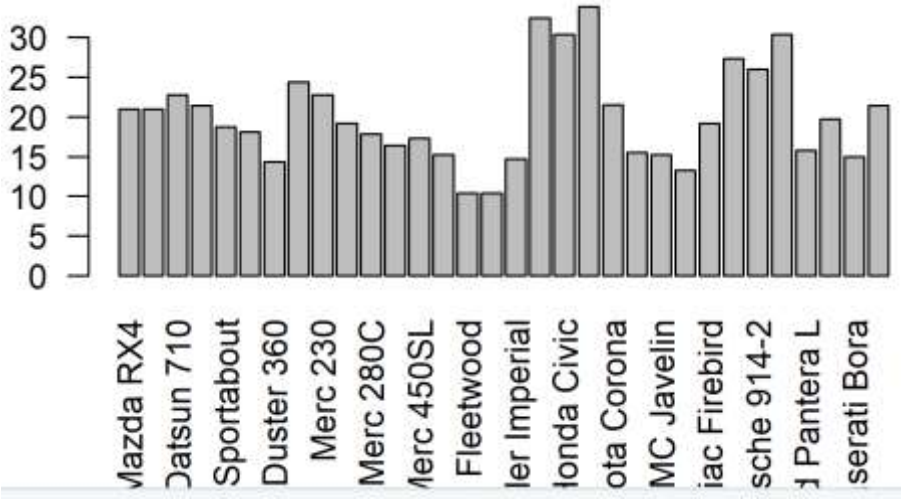
```
> barplot(mtcars$mpg, names=rownames(mtcars))
```



Η απόδοση των ονομάτων των μοντέλων κάτω από τις μπάρες που αντιστοιχούν στις τιμές τους δεν είναι σωστή καθώς τα ονόματα πέφτουν το ένα πάνω στο άλλο (εξαιτίας του πλήθους τους)

3. Αλλαγή διάταξης δεδομένων στους άξονες (las=):

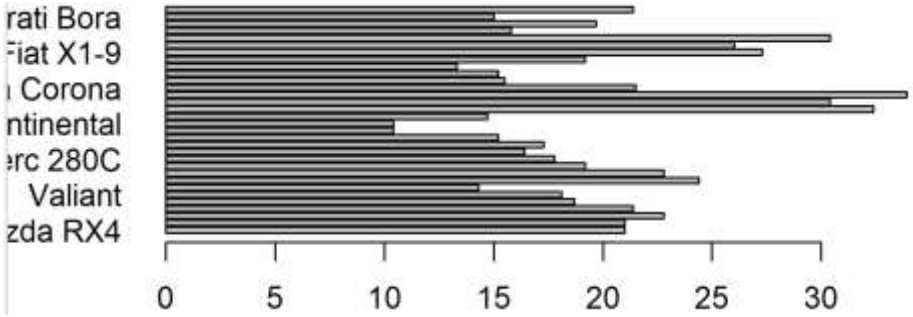
```
> barplot(mtcars$mpg, names=rownames(mtcars),las=2)
```



las=2, οι τιμές και των δύο αξόνων γίνονται κάθετες ως προς αυτούς

4. Αλλαγή διάταξης ραβδογράμματος (horiz=):

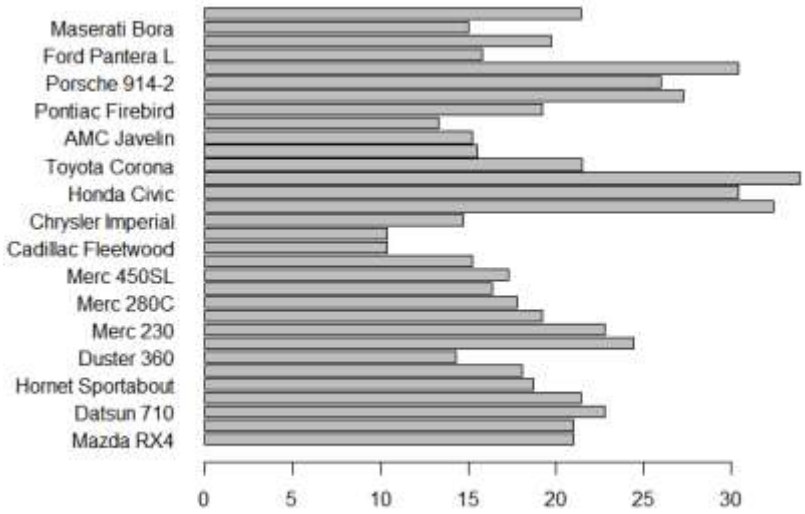
```
> barplot(mtcars$mpg, names=rownames(mtcars),las=1,horiz=T)
```



Αντιστροφή του προσανατολισμού, τα δεδομένα αναπαριστάνονται με οριζόντιες μπάρες. Το πρόβλημα δημιουργείται από τα μικρά περιθώρια και το μεγάλο μέγεθος γραμματοσειράς.

5. Αλλαγή περιθωρίων πλαισίου (par(mar)=c()):

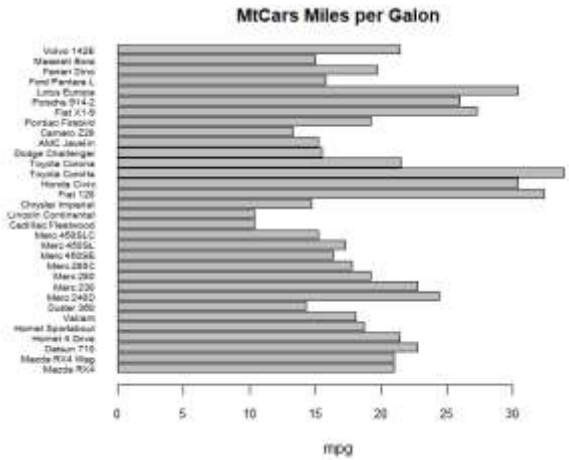
```
> par(mar=c(5,10,2,2))
> barplot(mtcars$mpg, names=rownames(mtcars),las=1,horiz=T)
```



Η αλλαγή επιτυγχάνεται μέσω της συνάρτησης *par()*, η οποία ορίζει παραμέτρους γραφικών, τον καθορισμό των περιθωρίων μέσω της *mar=c(bottom, left, top, right)*. Για τις ανάγκες του γραφήματος θέλουμε μια μεγάλη τιμή στο αριστερό περιθώριο για αυτό το θέτουμε ίσο με 10.

6. Τίτλοι αξόνων και γραφήματος (xlab/main/cex):

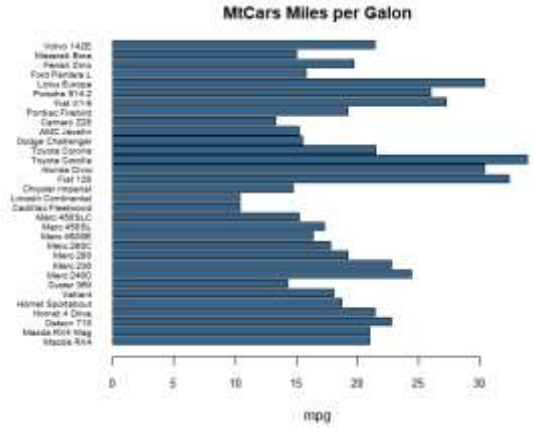
```
> par(mar=c(5,10,2,2))
> barplot(mtcars$mpg, names=rownames(mtcars), las=1,horiz=T, xlab='mpg', main='MtCars Miles per Galon', cex.main =1.2, cex.names=0.6, cex.axis=0.8)
```



Με την *xlab=* θέτουμε τον τίτλο του *x*-άξονα, με την *main=* δίνουμε γενικό τίτλο στο γράφημα ενώ με τις παραμέτρους τύπου *cex.X()=* ορίζουμε το σχετικό μέγεθος της γραμματοσειράς για το αντίστοιχο στοιχείο (X=main, names, axis).

7. Χρώμα (col=):

```
> barplot(mtcars$mpg, names=rownames(mtcars),las=1,horiz=T,xlab='mpg',main='MtCars Miles per Galon', cex.main =1.2,cex.names=0.6,cex.axis=0.8,col='steelblue4')
```



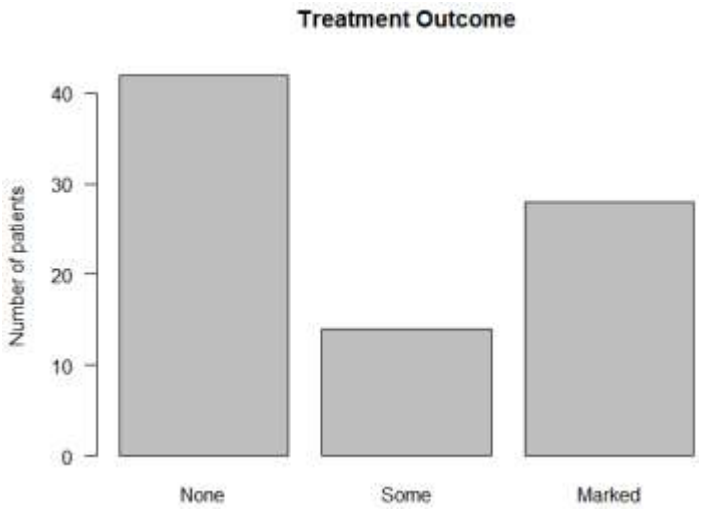
Σε ότι αφορά στα χρώματα, η R διαθέτει μια εξαιρετικά πλούσια παλέτα που κανείς μπορεί να δει απλά γράφοντας: ***colors()***

- Για να δημιουργήσουμε πιο σύνθετα ραβδογράμματα, θα εστιάσουμε σε ένα νέο, πιο σύνθετο σετ δεδομένων που δεν συμπεριλαμβάνεται στη βασική έκδοση της R. Για να αποκτήσουμε πρόσβαση σε αυτό χρειάζεται να εγκαταστήσουμε μέσω της *install.packages*(' ') και να φορτώσουμε μέσω της *library*(' ') την βιβλιοθήκη *vcd*.
- Η εν λόγω βιβλιοθήκη περιέχει τους αριθμούς ταυτότητας 84 ασθενών με αρθρίτιδα που λαμβάνουν αγωγή με ένα κανονικό ή ένα εικονικό φάρμακο (Placebo). Για κάθε ασθενή μας παρέχεται το είδος της αγωγής, το φύλο, η ηλικία και το αποτέλεσμα της θεραπείας με τρία πιθανά αποτελέσματα που αντιστοιχούν στη βελτίωση των συμπτωμάτων (marked, none και some).
- Το ακόλουθο ραβδόγραμμα απεικονίζει τον αριθμό των ασθενών που έδειξαν κάθε αποτέλεσμα:

```
> install.packages('vcd')
```
(χρειάζεται να τρέξει μόνο μια φορά)

```
> library('vcd')
```
(πρέπει να εκτελείται κάθε φορά που η R εκκινείται εκ νέου)

```
> str(Arthritis)
'data.frame':    84 obs. of  5 variables:
 $ ID       : int  57 46 77 17 36 23 75 39 33 55 ...
 $ Treatment: Factor w/ 2 levels "Placebo","Treated": 2 2 2 2 2 2 2 2 2 2 ...
 $ Sex      : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
 $ Age      : int  27 29 30 32 46 58 59 59 63 63 ...
 $ Improved : Ord.factor w/ 3 levels "None"<"Some"<..: 2 1 1 3 3 3 1 3 1 1 ...
```




```
> par(mar=c(5,5,5,5))
```
(επαναφέρουμε τα περιθώρια σε συμμετρικά επίπεδα)

```
> barplot(table(Arthritis$Improved),las=1,main='Treatment Outcome', ylab='Number of patients')
```

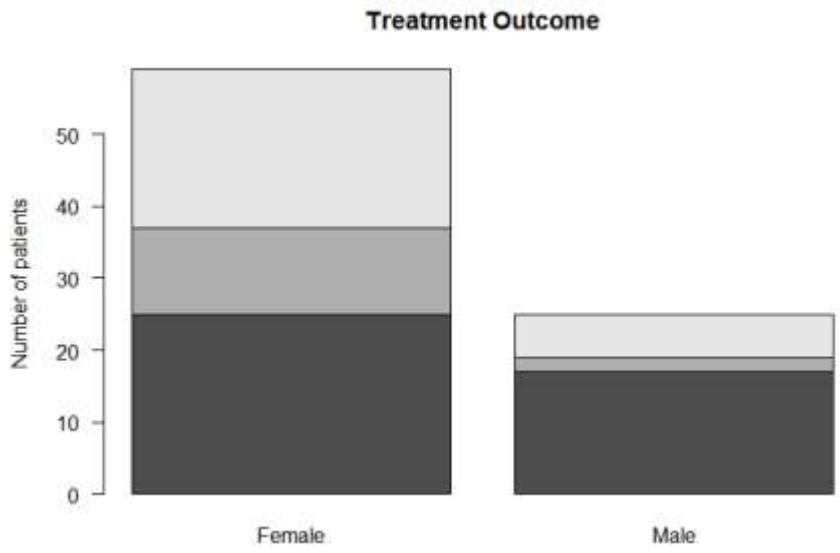
- Αν εφαρμόσουμε την table() σε δύο παράγοντες του dataframe, θα δημιουργήσουμε πιο σύνθετους πίνακες σύμπτωσης (contingency table).

```
> table(Arthritis$Improved,Arthritis$Sex)
```

	Female	Male
None	25	17
Some	12	2
Marked	22	6

Πίνακας με αριθμό ατόμων κάθε φύλου που εμφάνισαν αντίστοιχη ανταπόκριση στη θεραπεία.

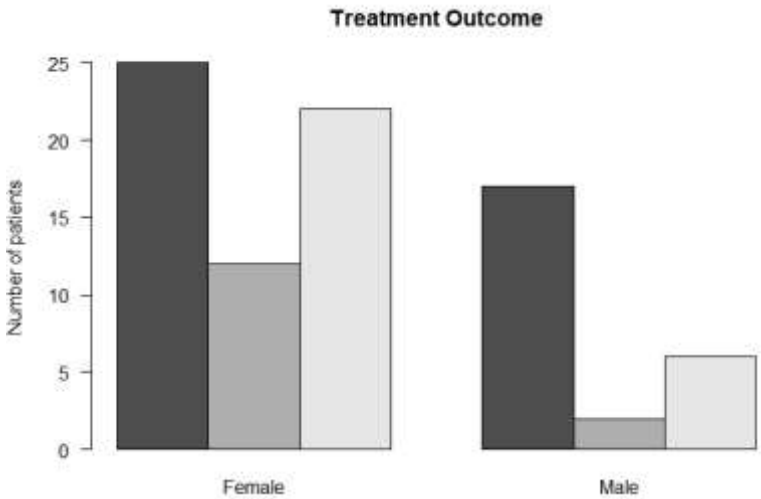
```
> barplot(table(Arthritis$Improved,Arthritis$Sex),las=1,main='Treatment Outcome',ylab='Number of patients')
```



Στοιβαγμένα ραβδογράμματα (stacked barplots) σε περιπτώσεις πινάκων δύο διαστάσεων.

- Για να ζητήσουμε από την *barplot()*να σχεδιάσει όλα τα δεδομένα σε ξεχωριστές ράβδους αρκεί να ορίσουμε την παράμετρο *beside()=T:*

```
> barplot(table(Arthritis$Improved,Arthritis$Sex),las=1,main='Treatment Outcome',ylab='Number of patients', beside=T)
```



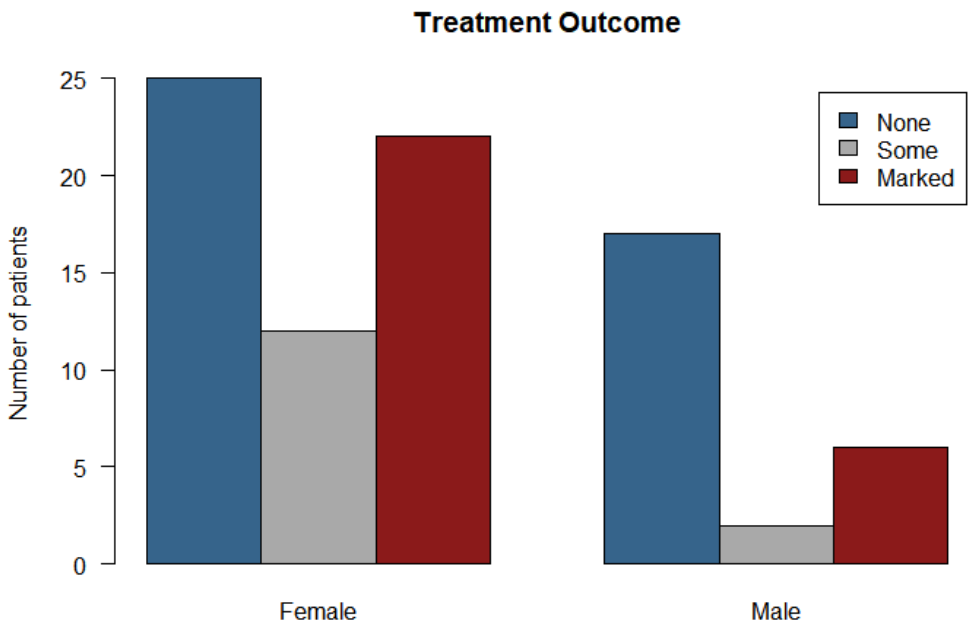
Κάθε δέσμη δεδομένων έχει ξεχωριστή ράβδο, αλλά χρειαζόμαστε και ένα υπόμνημα ώστε να ορίσουμε ποια ράβδος είναι ποια.

- Μια καλή προγραμματιστική τακτική είναι να δημιουργήσουμε μια ξεχωριστή μεταβλητή με τα δεδομένα που θα απεικονίσουμε

```
> datatab <- table(Arthritis$Improved,Arthritis$Sex)
> datatab

       Female Male
  None     25   17
  Some     12    2
  Marked   22    6
```

- Στη συνέχεια θα την χρησιμοποιήσουμε αρχικά στην *barplot()* και μετά μέσα στην ίδια συνάρτηση μέσω της παραμέτρου *legend*.

```
> barplot(table(Arthritis$Improved,Arthritis$Sex),las=1,main='Treatment Outcome', ylab='Number of patients', beside=T, legend=rownames(datatab), col=c('steelblue4','dark grey','firebrick4'))
```



- Από το γράφημα παρατηρούμε ότι υπάρχουν περισσότερες γυναίκες παρά άνδρες ασθενείς. Παρατηρούμε επίσης ότι μεταξύ ανδρών και γυναικών υπάρχουν διαφορές στην κατανομή της έκβασης της θεραπείας με περισσότερες αναλογικά γυναίκες να εμφανίζουν μερική βελτίωση σε σχέση με τους άντρες. Αντίθετα στους άντρες φαίνεται πάλι αναλογικά να υπάρχει ένα μεγαλύτερο ποσοστό με μηδενική βελτίωση. Για να έχουμε μια καλύτερη εικόνα χρειαζόμαστε αντί των απόλυτων αριθμών, το ποσοστό των αποτελεσμάτων της θεραπείας για κάθε φύλο ξεχωριστά.

```
> datatab
       Female  Male
  None      25   17
  Some      12    2
  Marked    22    6
```

- Θα πάρουμε το άθροισμα ανά στήλη με την συνάρτηση colSums():

```
> colSums(datatab)
Female   Male
   59     25
```

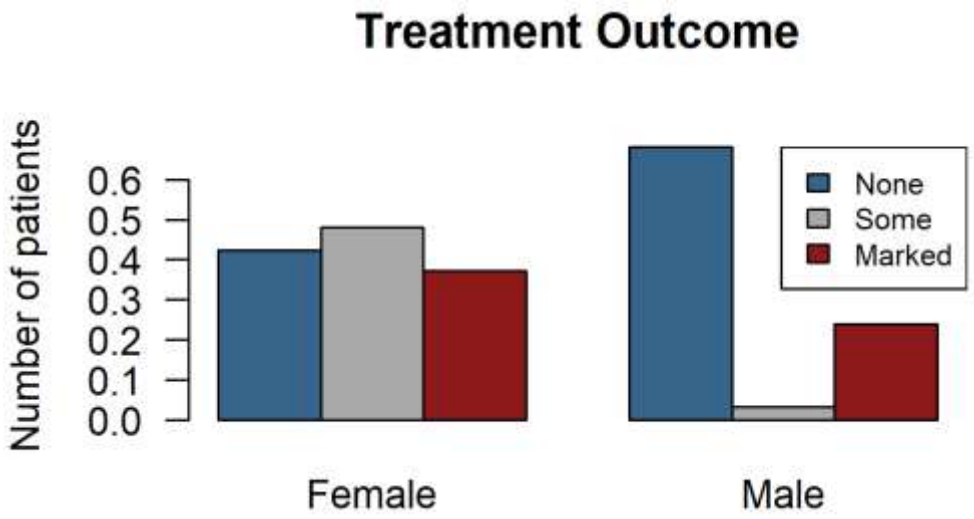
- Θυμηθείτε ότι το ίδιο μπορεί να γίνει με την γενική εφαρμογή συνάρτησης ανά στήλη ως εξής:

```
> apply(datatab,2,sum)
Female   Male
   59     25
```

- Διαίρεση του πίνακα με τα αθροίσματα δίνει έναν πίνακα σύμπτωσης που τώρα περιέχει κανονικοποιημένα ποσοστά.

```
> normdatatab <- datatab/colSums(datatab)
> normdatatab

          Female       Male
  None   0.42372881 0.68000000
  Some   0.48000000 0.03389831
  Marked 0.37288136 0.24000000
```

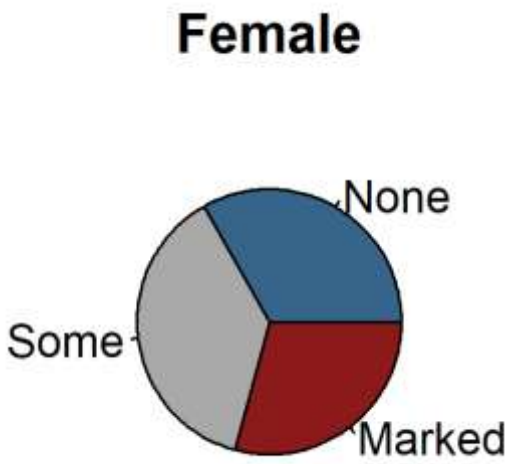
```
> barplot(normdatatab,las=1,main='Treatment Outcome',ylab='Number of patients', beside=T,
col=c('steelblue4','dark grey','firebrick4'))
> legend("topright", legend=rownames(normdatatab), fill=c('steelblue4','dark grey','firebrick4'), cex = 0.75)
```



- Το τελευταίο γράφημα αποτυπώνει ξεκάθαρα τις διαφορετικές τάσεις στην απόκριση στη θεραπεία μεταξύ των δύο φύλων. Οι άνδρες φαίνεται να ανταποκρίνονται λιγότερο σε σχέση με τις γυναίκες.

- Η βασική συνάρτηση της R για τη δημιουργία τέτοιων γραφημάτων είναι η *pie()*. Η χρήση της δεν διαφέρει από αυτή των περισσότερων συναρτήσεων γραφικών. Για παράδειγμα, στην περίπτωση του πίνακα που μόλις δημιουργήσαμε μπορούμε να επιλέξουμε να σχεδιάσουμε τα δεδομένα για τα αποτελέσματα θεραπείας για τις γυναίκες με την μορφή μιας πίτας ως εξής:

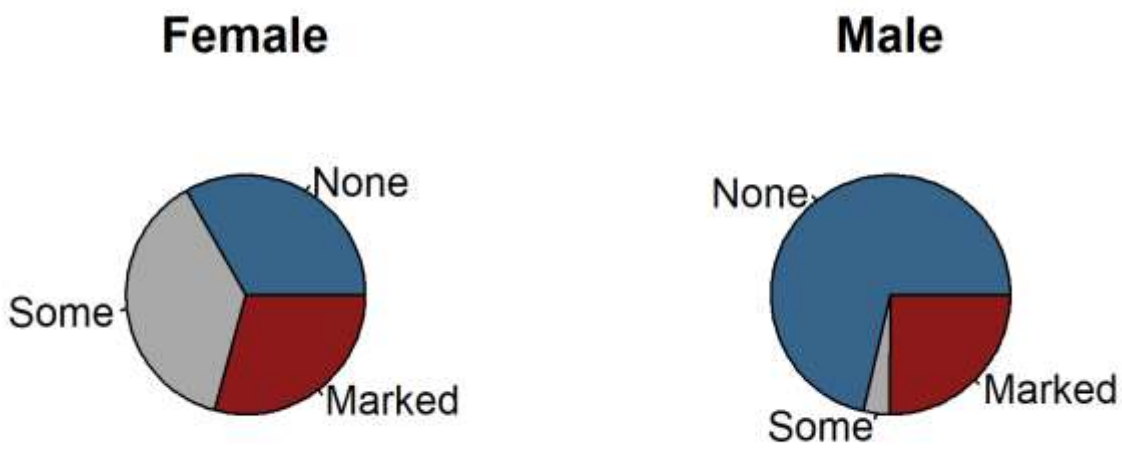
```
> pie(normdatatab[,1],labels=rownames(normdatatab),col=c('steelblue4','dark grey', 'firebrick4'),
main='Female')
```



- Στο παράδειγμα μας θα χρειαστούμε μια επιπλέον πίτα για να δούμε τι συμβαίνει με τους άντρες ασθενείς. Για να γίνει κάτι τέτοιο θα χρειαστεί να χωρίσουμε το πλαίσιο του γραφήματος σε δύο ίσα μέρη και να σχεδιάσουμε μια πίτα σε καθένα από αυτά.

- Ο διαχωρισμός της οθόνης επιτυγχάνεται με την κλήση μιας διαφορετικής επιλογής της συνάρτησης *par()*. Η επιλογή για τον διαχωρισμό της οθόνης ονομάζεται *mfrow()* και δέχεται ένα διάνυσμα μήκους 2 που αντιστοιχεί στον αριθμό των α) σειρών και β) των στηλών στις οποίες θα διαχωριστεί το πλαίσιο του γραφήματος. Σε αυτή την περίπτωση, εφόσον θέλουμε να σχεδιάσουμε δύο πίτες δίπλα-δίπλα, θα χωρίσουμε το πλαίσιο σε δύο στήλες (και με μία μόνο σειρά).

```
> par(mfrow=c(1,2))
> pie(normdatatab[,1],labels=rownames(normdatatab),col=c('steelblue4','dark grey', 'firebrick4'), main='Female')
> pie(normdatatab[,2],labels=rownames(normdatatab),col=c('steelblue4','dark grey', 'firebrick4'), main='Male')
```



Για να δημιουργήσετε επόμενο γράφημα, θυμηθείτε να επαναφέρετε το πλαίσιο πίσω σε 1 σειρά και σε 1 στήλη. Αυτό γίνεται με την εντολή:

```
> par(mfrow=c(1,1))
```

Διαγράμματα σκέδασης για συζευγμένα δεδομένα με την plot()

- Τα ραβδογράμματα και οι πίτες είναι επαρκείς αναπαραστάσεις για σύνολα δεδομένων μιας διάστασης, υπό την έννοια ότι εξετάζουμε αριθμητικές τιμές που αναφέρονται σε ορισμένες κατηγορικές περιπτώσεις. Αν θέλουμε να αναπαραστήσουμε δεδομένα που έρχονται σε ζεύγη μπορούμε να χρησιμοποιήσουμε τα διαγράμματα σκέδασης (scatterplots), στα οποία συζευγμένα δεδομένα αναπαρίστανται ως σημεία σε ένα δισδιάστατο χώρο. Η συνάρτηση *plot()* της R είναι η πιο απλή λύση για αυτό.

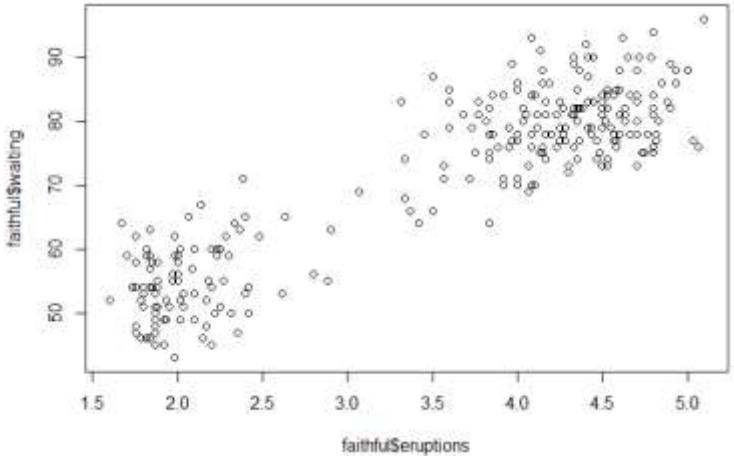
- Παράδειγμα συζευγμένων δεδομένων με το dataframe *faithful*. Το *faithful* περιέχει μετρήσεις για το χρόνο διάρκειας των εκρήξεων του Old Faithful, ενός διάσημου θερμοπίδακα που βρίσκεται στο εθνικό πάρκο Yellowstone του ΗΠΑ. Οι χρόνοι διάρκειας των εκρήξεων αναφέρονται έναντι του χρόνου αναμονής μεταξύ τους για 271 διαδοχικές εκρήξεις.

```
> head(faithful)
  eruptions waiting
1     3.600      79
2     1.800      54
3     3.333      74
4     2.283      62
5     4.533      85
6     2.883      55
```

- Στην απλούστερη δυνατή προσέγγιση, θα θέλαμε να δούμε αν υπάρχει μια σύνδεση μεταξύ των δύο διαφορετικών δεδομένων, δηλαδή εάν η διάρκεια της έκρηξης σχετίζεται κατά κάποιο τρόπο με τους χρόνους αναμονής.

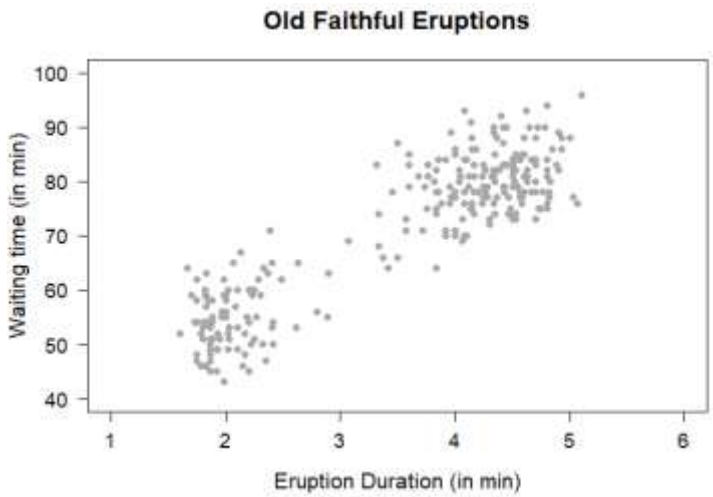
Διαγράμματα σκέδασης για συζευγμένα δεδομένα με την plot()

> plot(faithful$eruptions, faithful$waiting)



Απλό Scatterplot καθώς βλέπουμε μόνο σημεία ως λευκούς κύκλους με ελάχιστο σχολιασμό σε ό,τι αφορά στους τίτλους, στους άξονες κλπ.

> plot(faithful$eruptions,faithful$waiting, col='dark grey', pch=19, xlim=c(1,6),ylim=c(40,100),xlab='Eruption Duration (in min)', ylab='Waiting time (in min)',main='Old Faithful Eruptions', cex.axis=1.2, cex.lab=1.3, cex.main=1.5, las=1)



- Αλλαγή του χρώματος των λευκών κύκλων σε γκρι με την επιλογή, *col=*
- Κάνουμε τους κύκλους μεγαλύτερους με την επιλογή, *pch=19*. Θα μπορούσαμε να έχουμε γραφικά στοιχεία σχήματος άστρων αντί για κύκλους, θέτοντας *pch=8*, ή τετραγώνων θέτοντας *pch=15*.
- Ρυθμίζουμε το μέγεθος της γραμματοσειράς για τις αριθμητικές τιμές των αξόνων με το cex.axis=, τις ετικέτες των αξόνων με το cex.lab= και του κυρίως τίτλου με το cex.main= (<1 για μικρότερες γραμματοσειρές και >1 για μεγαλύτερες).

- Αν έχουμε δισδιάστατα δεδομένα, στα οποία η πρώτη μεταβλητή μπορεί να θεωρηθεί ως μια σειρά από κάποιο μέγεθος. Αυτό μπορεί να είναι δεδομένα χώρου ή χρόνου ή ακόμα και απλές απαριθμήσεις μιας ταξινομημένης μεταβλητής (1,2,3,… κ.λπ.)
- *Παράδειγμα*: Για το πλαίσιο δεδομένων *faithful* να σχεδιάσετε τη διάρκεια κάθε έκρηξης για μια σειρά που να αποτελείται από τις πρώτες 100 εκρήξεις. Οι διάρκειες θα είναι ίσες με:

```
> durations <- faithful$eruptions[1:100]
```

Στο αρχικό ερώτημα υπονοείται και μια επιπλέον μεταβλητή που δεν περιέχεται αναλυτικά στο *dataframe*, δηλαδή η σειρά των εκρήξεων από το 1 έως το 100. Παρότι δεν την έχουμε, μπορούμε εύκολα να τη δημιουργήσουμε φτιάχνοντας μια σειρά από ακέραιους αριθμούς από το 1 έως το 100. Η συνάρτηση για τη δημιουργία σειρών είναι η *seq(from, to, by)*. Η χρήση της παρουσιάζεται παρακάτω:

```
> seq(from=1,to=10,by=1)
[1]  1  2  3  4  5  6  7  8  9 10
```

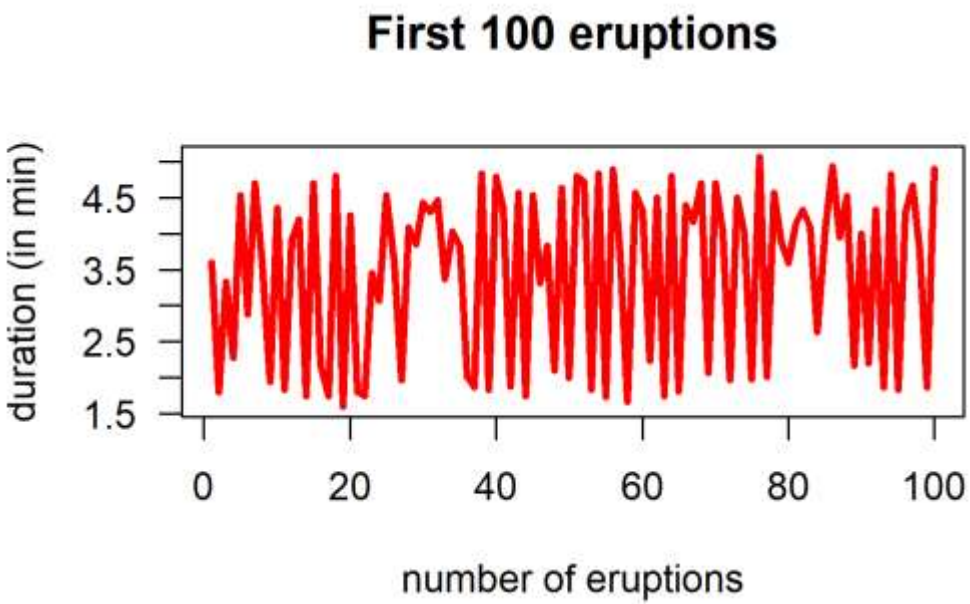
Η *seq()* δημιουργεί μια ακολουθία όχι μόνο ακεραίων αλλά και πραγματικών αριθμών μεταξύ ενός κατώτερου και ανώτερου ορίου μαζί με ένα βήμα.

```
> seq(from=10,to=-10,by=-2.5)
[1]  10.0   7.5   5.0   2.5   0.0  -2.5  -5.0  -7.5 -10.0
```

Επιστρέφοντας στο πρόβλημα μας, δημιουργούμε μια σειρά i η οποία είναι η σειρά διαδοχής των εκρήξεων:

```
> i <- seq(from=1,to=100,by=1)
```

```
> plot(i,faithful$eruptions[i],type='l',col='red',lwd=3,las=1,ylab='duration (in min)', xlab='number of eruptions', main='First 100 eruptions')
```



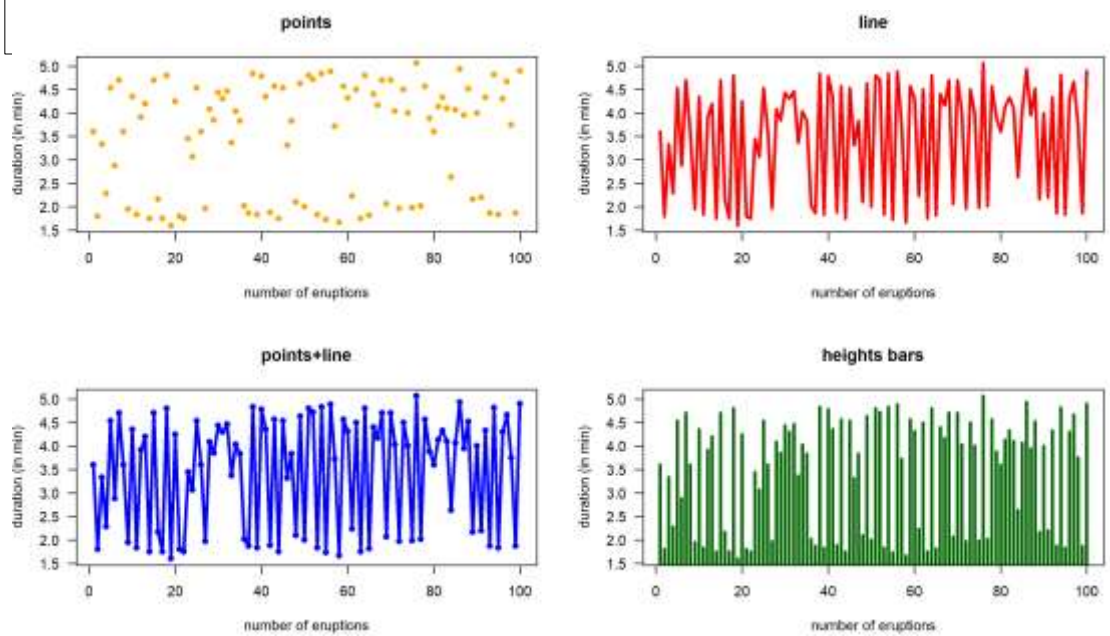
type = 'l' : επιλογή συνεχούς γραμμής. Εναλλακτικοί τύποι διαγραμμάτων είναι τα 'p' (απλά σημεία), 'h' (πλήρεις γραμμές ύψους), 'o' (γραμμές με σημεία).

lwd = '3' : αλλάζει το πάχος της γραμμής (η τιμή default είναι το 1)

Χρήση της plot() για σειριακά δεδομένα

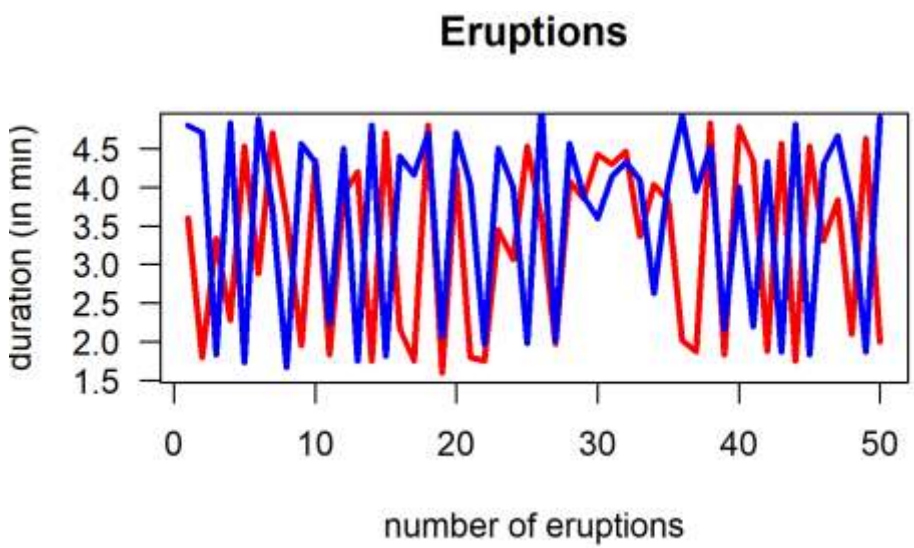
- Στο παρακάτω τετραπλό διάγραμμα φαίνονται και οι τέσσερις τρόποι για το ίδιο dataset (το πλαίσιο διαγράμματος χωρίζεται σε 2x2 τμήματα)

```
par(mfrow=c(2,2))
plot(i,faithful$eruptions[i],type='p',pch=16,col='orange',las=1,ylab='duration (in min)', xlab='number of eruptions', main='points')
plot(i,faithful$eruptions[i],type='l',lwd=3,col='red',las=1,ylab='duration (in min)', xlab='number of eruptions', main='line')
plot(i,faithful$eruptions[i],type='o',pch=16,lwd=3,col='blue',las=1,ylab='duration (in min)', xlab='number of eruptions', main='points+line')
plot(i,faithful$eruptions[i],type='h',lwd=3,col='dark green',las=1,ylab='duration (in min)', xlab='number of eruptions', main='heights bars')
```



- Αν θέλουμε να δούμε περισσότερα από ένα σετ δεδομένων στον y άξονα στο ίδιο διάγραμμα, χρειάζεται μαζί με την *plot()* να χρησιμοποιήσουμε μια επιπλέον συνάρτηση την *lines()*. Η *lines()* προσθέτει ένα σετ δεδομένων στις ίδιες συντεταγμένες x-άξονα με την απαραίτητη προϋπόθεση τα διανύσματα να είναι ίδιου μήκους, το κάνει δε πάνω στο ίδιο διάγραμμα και για τον λόγο αυτό πάντοτε θα πρέπει να έχει προηγηθεί μια κλήση της *plot()*.
- Παράδειγμα: Θα σχεδιάσουμε τις πρώτες 50 και τις επόμενες 50 διάρκειες εκρήξεων:

```
i <- seq(from=1,to=50,by=1)
durationsA<- faithful$eruptions[1:50]
durationsB<- faithful$eruptions[51:100]
plot(i,durationsA,type='l',col='red',lwd=3,las=1,ylab='duration (in min)', xlab='number of eruptions', main='Eruptions')
lines(i,durationsB,type='l',col='blue',lwd=3)
```

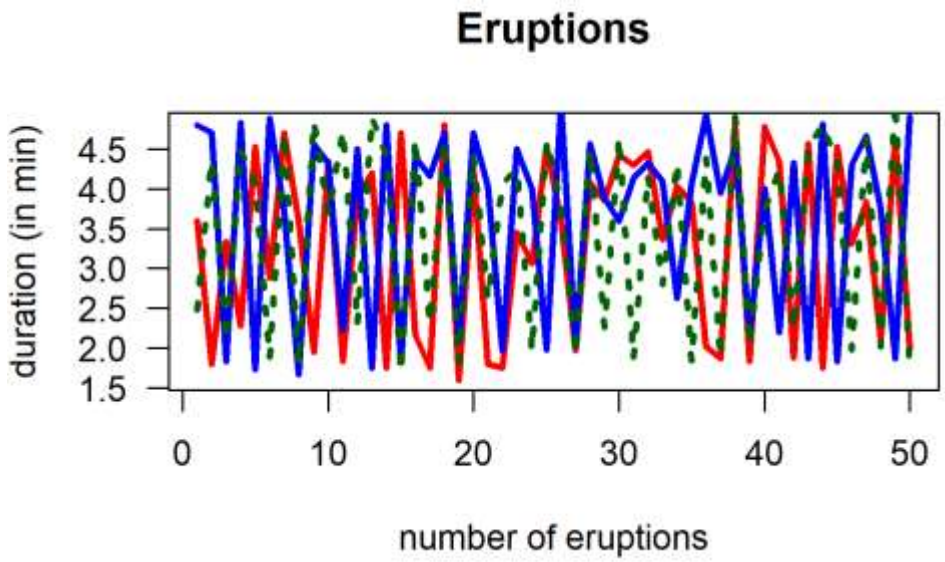


Πολλαπλά διαγράμματα σε γραφική με την lines()

- *Η εκτέλεση της lines()* μπορεί να γίνει όσες φορές θέλουμε, εφόσον το διάγραμμα εξακολουθεί να είναι ευανάγνωστο. Αν π.χ. προσθέσουμε ακόμα πενήντα εκρήξεις με μια πράσινη γραμμή:

```
i <- seq(from=1,to=50,by=1)
durationsA<- faithful$eruptions[1:50]
durationsB<- faithful$eruptions[51:100]
plot(i,durationsA,type='l',col='red',lwd=3,las=1,ylab='duration (in min)', xlab='number of eruptions', main='Eruptions')
lines(i,durationsB,type='l',col='blue',lwd=3)
durationsC<- faithful$eruptions[101:150]
lines(i,durationsC, type='l',col='dark green', lwd=3, lty=3)
```

- Αλλαγή του τύπου της γραμμής με την παράμετρο, lty=
- Συνεχής γραμμή (default), lty=1
- Γραμμή διακεκομμένη με παύλες, lty=2
- Γραμμή διακεκομμένη με τελείες, lty=3

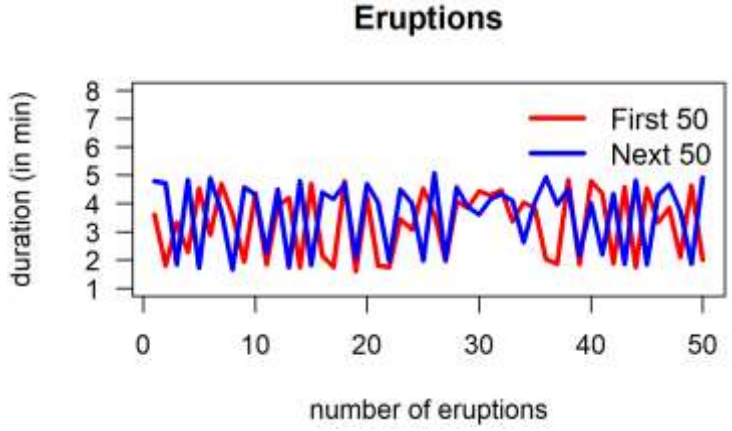


Πολλαπλά διαγράμματα σε γραφική με την lines()

- Η *R* δημιουργεί υπομνήματα με την συνάρτηση *legend()*, της οποίας οι παράμετροι καθορίζουν τη θέση, τον τύπο (γραμμές, σημεία), τα χρώματα κλπ. Η *legend()* εκτελείται στο τέλος μιας σειράς κλήσεων των *plot()* και *lines()*.

```
# Δημιουργία τιμών x-άξονα
i <- seq(from=1,to=50,by=1)
# Δημιουργία σετ δεδομένων
durationsA<- faithful$eruptions[1:50]
durationsB<- faithful$eruptions[51:100]
#Δημιουργία διαγράμματος με το πρώτο σετ
plot(i,durationsA,type='l',col='red',lwd=3,las=1,ylab='duration (in min)', xlab='number of eruptions', main='Eruptions',ylim=c(1,8))
#Προσθήκη δεύτερου διαγράμματος
lines(i,durationsB,type='l',col='blue',lwd=3)
#Προσθήκη υπομνήματος
legend('topright',c('First 50', 'Next 50'),col=c('red','blue'),lwd=c(3,3),bty='n',cex=1.1)
```

- Η legend() παίρνει μια σειρά από παραμέτρους όπως η θέση (topright, topleft, bottomleft κλπ), ακολουθούμενα από διανύσματα που περιέχουν το κείμενο, τα χρώματα γραμμών και τα πλάτη γραμμών.
- Η παράμετρος bty='n' λέει στην R να μην σχεδιάσει ένα κουτί γύρω από το υπόμνημα και το cex καθορίζει το μέγεθος του (default=1).

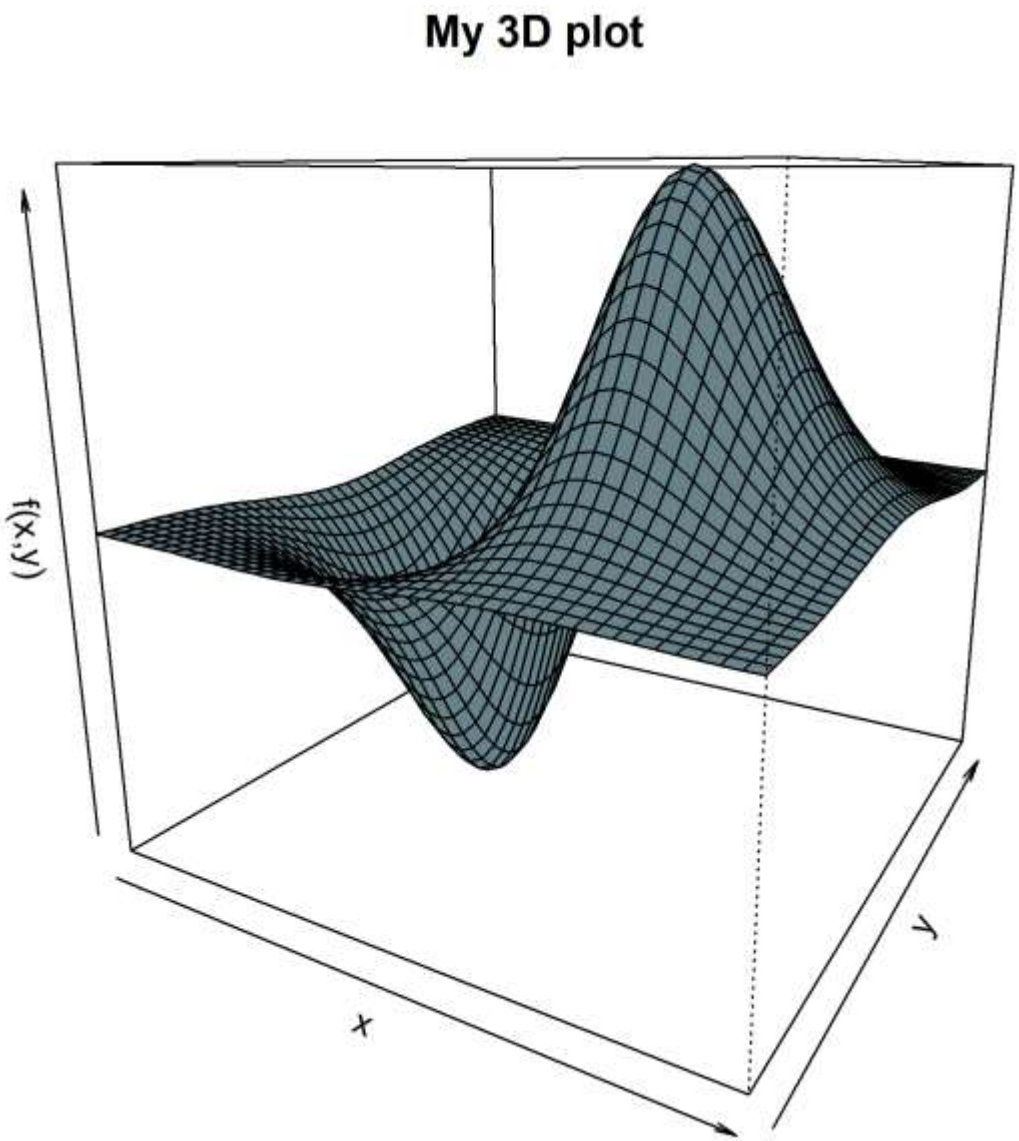


- Η *συνάρτηση persp()* λαμβάνει ως όρισμα 3 μεταβλητές από τις οποίες, οι πρώτες δύο *(x, y)* θα πρέπει να είναι διανύσματα και να είναι αυτές που ορίζουν ένα πλέγμα ή την κύρια επιφάνεια. Η τρίτη *z* πρέπει να είναι ένας πίνακας διαστάσεων *[x, y]* που αντιπροσωπεύει κάποιο είδος ύψους για κάθε συντεταγμένη 2D του πλέγματος.

Παράδειγμα: Γραφική παράσταση συνάρτησης $z = f(x, y) = xe^{-x^2-y^2}$

```
x <- seq(from=-2,to=2,by=0.1)

y <- seq(from=-2,to=2,by=0.1)

function1 <- function (x, y) {

  return (x*exp(-((x^2)+(y^2))))

}

z <- outer(x, y, function1)

persp(x, y, z, phi=15, theta=30, col='lightblue4', main='My 3D plot’, xlab="x", ylab="y",
zlab="f(x,y) ")
```

- **Δημιουργία δύο διανυσμάτων x και y με ίσα μήκη. Έτσι οι διαστάσεις του πλέγματος είναι 41x41.**
- **Με τη συνάρτηση outer υπολογίζεται εύκολα και ο πίνακας με στοιχεία f(xi,yj), όπου f(x,y) συνάρτηση δύο μεταβλητών**
- **(phi, theta): καθορίζουν την οπτική γωνία του γραφήματος**


My 3D plot
f(x,y)
x
y