

Διαγράμματα με το πακέτο pgfplots

Αποστολος Συρόπουλος

28ης Οκτωβρίου 366

671 33 Ξάνθη

H/T: *asyropoulos at yahoo dot com*

Ένα πρόβλημα που αντιμετωπίζει οποιοσδήποτε γράφει επιστημονικά κείμενα είναι ο τρόπος με τον οποίο θα δημιουργήσει τα διαγράμματα που πιθανά συνοδεύουν τα κείμενά του. Το πακέτο `pgfplots` είναι μια εξαιρετική λύση για όποιον χρησιμοποιεί το \LaTeX για την δημιουργία των κειμένων του. Στο παρόν περιγράφεται η βασική χρήση του πακέτου, δηλαδή, η δημιουργία απλών διαγραμμάτων και πώς βάζουμε σημεία και κείμενο σε αυτά. Επιπλέον, παρουσιάζονται κάποια πιο εξεζητημένα θεματά, καθώς και ο τρόπος με τον οποίο κάποιος μπορεί να δημιουργήσει ραβδογράμματα και κυκλικά διαγράμματα (διαγράμματα πίτας).

Diagrams with `pgfplots`, by *Apostolos Syropoulos* — Most authors of science papers and books need to create plots and diagrams to display scientific data. The problem of course is which tool to use for this task. In general, people prefer to use a tool from a suit of tools which is familiar to them. The package `pgfplots` is the ideal package for people who use \LaTeX to prepare their documents. This paper describes the basic use of the package, i.e., how to create a simple diagram and how to place points and text on it. In addition, it presents some specialized features and how one can create create bar charts and pie charts.

Εισαγωγή

Στο τεύχος 34–35 του *Εύτυπου* παρουσιάσαμε τη χρήση των πακέτων `TikZ` και `PGF` για τη δημιουργία σχημάτων [1]. Αν και είναι εξαιρετικά χρήσιμο να μπορούμε να δημιουργούμε σχήματα, εξίσου χρήσιμο είναι να μπορούμε να δημιουργούμε διαγράμματα. Όταν ετοιμάζουμε τεχνικό κείμενο, πολλές φορές χρειαζόμαστε να δημιουργήσουμε και να προσθέσουμε σ’ αυτό διαγράμματα. Τότε συνήθως καταφεύγουμε σε άλλα «εξωτερικά» εργαλεία, όπως το `Calc` του `LibreOffice`, το `gnuplot` ή κάποια εμπορικά προγράμματα.

Ο συγγραφέας του παρόντος άρθρου, όποτε χρειάστηκε να δημιουργήσει διαγράμματα για κείμενο, δεν σκέφτηκε να χρησιμοποιήσει κάποιο εξωτερικό εργαλείο. Ο λόγος είναι επειδή θέλει να έχει τον έλεγχο των γραμματοσειρών που χρησιμοποιούνται. Κατ’ αυτόν τον τρόπο, το διάγραμμα δεν μοιάζει σαν ξένο σώμα

στο τελικό αποτέλεσμα. Αυτός είναι και ο λόγος που χρησιμοποίησε το πακέτο `pgfplots` [2], το οποίο στην ουσία κάνει χρήση των δυνατοτήτων που παρέχουν τα πακέτα `TikZ` και `PGF`.

Το πρώτο παράδειγμα

Όπως έχω αναφέρει και προηγουμένως [1], καλό είναι τα διαγράμματα να γίνονται σε ξεχωριστά αρχεία και εμείς απλά να φορτώνουμε στο κύριο έγγραφο μόνο την τελική εικόνα. Ο λόγος φυσικά είναι ότι το κύριο έγγραφο γίνεται λιγότερο πολύπλοκο.

Ας δούμε τώρα τη γενική δομή αρχείου που μπορούμε να χρησιμοποιήσουμε για να δημιουργήσουμε ένα διάγραμμα με το πακέτο `pgfplots`:

```

1 \documentclass{standalone}
2 \usepackage{xltextra}
3 \usepackage{xgreek}
4 \usepackage{pgfplots}
5 \usepackage{unicode-math}
6 \begin{document}
7   \setmainfont[Mapping=tex-text,Ligatures=Common]{Linux Libertine O}
8   \setmathfont[Scale=MatchUppercase]{Asana Math}
9 \begin{tikzpicture}
10  ... Εντολές pgfplots ...
11 \end{tikzpicture}
12 \end{document}

```

Η κλάση εγγράφου `standalone` είναι μια μινιμαλιστική κλάση η οποία δίνει ένα αρχείο PDF που καταλαμβάνει μόνο τον χώρο που απαιτείται για το σχήμα. Έτσι δεν χρειάζεται να κόψουμε το παραγόμενο αρχείο. Επίσης, ο αναγνώστης μπορεί να χρησιμοποιεί όποια γραμματοσειρά θέλει.

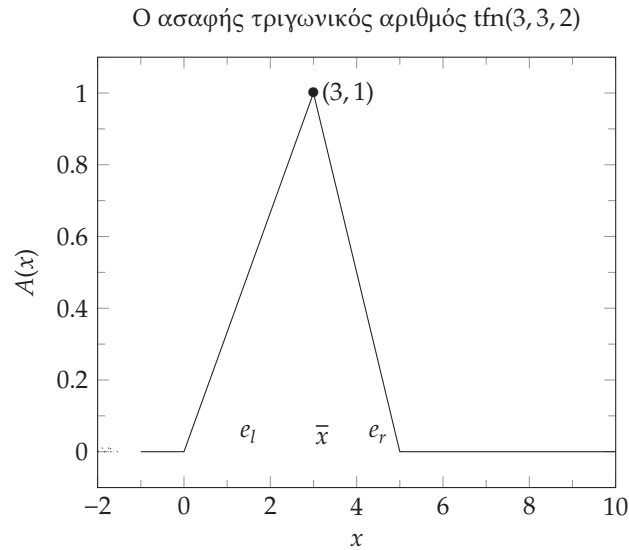
Ένα απλό διάγραμμα

Στην Εικόνα 1 παρουσιάζεται ένα τυπικό διάγραμμα το οποίο μπορεί να δημιουργηθεί με τη χρήση του πακέτου `pgfplots`. Όλο τα δεδομένα του διαγράμματος σημειώνονται σε ένα περιβάλλον `axis` ενώ οι παράμετροι του διαγράμματος δίνονται ως κατ’ επιλογή ορίσματα του περιβάλλοντος. Για παράδειγμα, ορίστε πώς ξεκινάει ο κώδικας που χρησιμοποιήσαμε για τη δημιουργία του διαγράμματος της Εικόνας 1:

```

1 \begin{axis}[
2   title={0 ασαφής τριγωνικός αριθμός  $\mathord{\mathrm{tfn}}(3,3,2)$ },
3   xlabel={ $x$ },
4   ylabel={ $A(x)$ },

```



Εικόνα 1: Τυπικό διάγραμμα το οποίο δημιουργήθηκε με το πακέτο *pgfplots*.

```

5   xmin=-2, xmax=10,
6   minor x tick num=1,
7   ]

```

Στη γραμμή 2 βάζουμε τον τίτλο του διαγράμματος. Φυσικά, αν δε θέλουμε να έχουμε τίτλο στο διάγραμμά μας, απλά δεν βάζουμε τη σχετική εντολή. Προφανώς μπορούμε να κάνουμε το ίδιο με όλες τις άλλες παράμετρους. Στη γραμμή 3 βάζουμε το κείμενο (τίτλο) που θα εμφανιστεί στον οριζόντιο άξονα, ενώ στη γραμμή 4 βάζουμε τον τίτλο του κάθετου άξονα. Στη γραμμή 5, βάζουμε το πεδίο τιμών του οριζόντιου άξονα. Τα `xmin` και `xmax` ορίζουν την ελάχιστη και τη μέγιστη τιμή αντίστοιχα που θα πάρει ο οριζόντιος άξονας. Στη γραμμή 6 σημειώνουμε σε ποια σημεία του οριζόντιου άξονα θα μπαίνουν οι υποδιαιρέσεις του (οι λεπτές γραμμές που μπαίνουν στους άξονες). Αν θέλουμε, μπορούμε να βάλουμε υποδιαιρέσεις και στον κατακόρυφο άξονα με την παρακάτω εντολή:

```
minor y tick num=1
```

Φυσικά μπορούμε να δηλώσουμε πιο πολύπλοκα πράγματα όπως φαίνεται στον παρακάτω κώδικα:

```

xmin = 0.000, xmax = 0.018, ymin = 2, ymax = 9,
xtick = {0.000,0.002,...,0.016}, extra y ticks = {3,5,7},
minor x tick num=1, minor y tick num=4,

```

Η μπορούμε ακόμα να δηλώσουμε πως θέλουμε το κόμμα (,) να είναι το σημείο δεκαδικής υποδιαστολής, αντί για την τελεία (.):

```
\begin{axis}[/pgf/number format/.cd,
    use comma,
    ...
]
```

Στο συγκεκριμένο παράδειγμα, ορίζουμε τις μέγιστες και ελάχιστες τιμές στους δύο άξονες, τα σημεία στα οποία θα μπουν οι βασικές υποδιαίρέσεις στον οριζόντιο άξονα και πού θα μπουν οι επιπλέον βασικές υποδιαίρέσεις στον κατακόρυφο άξονα. Ακόμη ορίζουμε και τις μικρές υποδιαίρέσεις.

Το επόμενο βήμα στη δημιουργία ενός διαγράμματος είναι η εισαγωγή των σημείων του. Ένας τρόπος είναι να αναγκάσουμε το $\text{Xe}_{\text{L}}\text{T}_{\text{E}}\text{X}$ να διαβάσει τα δεδομένα από ένα αρχείο:

```
\addplot [black] table {triangle.dat};
```

Στην περίπτωση αυτή θα χαραχθεί μια μαύρη γραμμή και τα σημεία δεν θα φαίνονται. Το αρχείο που περιέχει τα δεδομένα θα πρέπει να έχει την παρακάτω μορφή:

x_0	$f(x)$
-1	0
-0.75	0
-0.5	0
-0.25	0

Τα σημεία μπορούν να παραχθούν από ένα πρόγραμμα ή να είναι δεδομένα τα οποία έχουμε συλλέξει και καταγράψει μόνοι μας. Για παράδειγμα, για να δημιουργήσουμε το διάγραμμα της Εικόνας 1, χρησιμοποιήσαμε το πρόγραμμα Perl που ακολουθεί:

```
1 open(OUT, ">triangle.dat")|| die "can't create file triangle.dat\n";
2 print OUT "x_0\tf(x)\n";
3 for( $x=-1; $x<=10; $x+=0.25 ) {
4     print OUT $x,"\t";
5     $fx = min(max(0, 1-(3-$x)/3), max(0, 1-($x-3)/2));
6     print OUT $fx,"\n";
7 }
8 close OUT;
```

Το πρόγραμμα πολύ απλά δημιουργεί το αρχείο εξόδου και, για $-1 \leq x \leq 10$, τυπώνει ανά 0,25 τις τιμές του x και του $f(x)$. (Οι συναρτήσεις \min και \max έχουν οριστεί ξεχωριστά και φυσικά ο ορισμός τους είναι πολύ απλός.)

Ένας άλλος τρόπος εισαγωγής των σημείων είναι απευθείας μέσα στον κώδικα $\text{T}_{\text{E}}\text{X}$:

```
\addplot[color=red,mark=x] coordinates {
  (2,-2.8559703)
  (3,-3.5301677)
  . . . . .
  (8,-7.9377747)
};
```

Βλέπουμε πως απλά γράφουμε `coordinates` και στη συνέχεια βάζουμε ένα αριστερό άγκιστρο (`{`), ακολουθούν τα ζεύγη τιμών χωρισμένα με κόμμα και κλείνουμε με ένα δεξιό άγκιστρο (`}`). Για κάθε ζεύγος τιμών σημειώνουμε μια αριστερή παρένθεση, την πρώτη τιμή, ένα κόμμα, τη δεύτερη τιμή και δεξιά παρένθεση. Στο παράδειγμά μας, η γραμμή θα είναι κόκκινη, ενώ τα σημεία θα σημειώνονται με ένα `x`. Άλλα σύμβολα που μπορούν να χρησιμοποιηθούν είναι: `*` (παράγει έναν μικρό κυκλικό δίσκο), `square` (τετράγωνο), `triangle` (τρίγωνο), `diamond` (ρόμβος) και `pentagon` (πεντάγωνο). Αν τα ονόματα συμβόλων ακολουθούνται από ένα `*`, τότε εμφανίζονται μαύρα. Φυσικά μπορούμε να τα κάνουμε και χρωματιστά.

Αντί να εισαγάγουμε τα σημεία ενός διαγράμματος, μπορούμε να βάλουμε το ίδιο το $X^2 - x + 4$ να κάνει τις πράξεις! Για παράδειγμα η εντολή

```
\addplot {x^2 - x +4};
```

θα σχεδιάσει την παράσταση $x^2 - x + 4$. Εκτός από απλές αλγεβρικές παραστάσεις, το πακέτο *pgfplots* μπορεί να κάνει τις πράξεις `+` (πρόσθεση), `-` (αφαίρεση), `*` (πολλαπλασιασμός) και `/` (διαίρεση), και να υπολογίζει τις μαθηματικές συναρτήσεις `abs` (απόλυτη τιμή), `round` (στρογγύλεμα κατά πάνω), `floor` (στρογγύλεμα κατά κάτω), `mod` (υπόλοιπο ακέραιας διαίρεσης), `max`, `min`, `sin`, `cos`, `tan`, `deg` (μετατροπή ακτινίων σε μοίρες), `rad` (μετατροπή μοιρών σε ακτίνια), `atan`, `asin`, `acos`, `cot`, `sec`, `cosec`, `exp`, `ln`, `sqrt`, οι σταθερές `pi` (π) και `e`, `^` (ύψωση σε δύναμη), `factorial` (παραγοντικό, αλλά σε νεότερες εκδόσεις μπορεί κάποιος να γράψει `5!` το οποίο είναι ισοδύναμο με το `factorial 5`), `rand` (ψευδοτυχαίος αριθμός μεταξύ -1 και 1), `rnd` (ψευδοτυχαίος αριθμός μεταξύ 0 και 1), κ.ά.

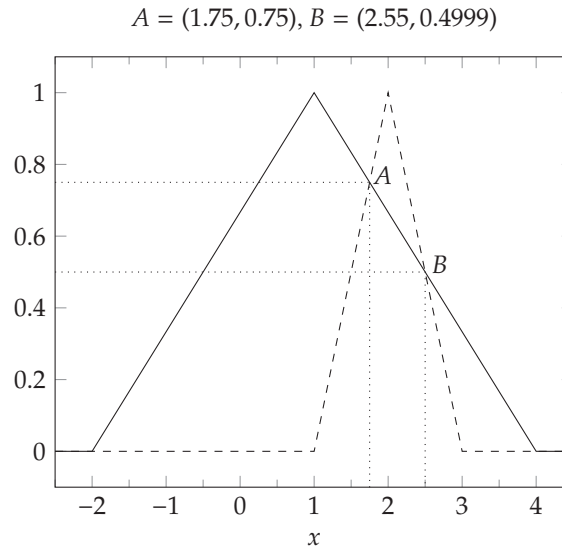
Έτσι για παράδειγμα αν θέλουμε να σχεδιάσουμε την καμπύλη του ημιτόνου από $-\pi$ ως π , μπορούμε να χρησιμοποιήσουμε την παρακάτω εντολή:

```
\addplot [domain=-pi:pi,] {sin(deg(x))};
```

Αν δεν θέλουμε να βάλουμε κάτι άλλο στο διάγραμμα, απλά κλείνουμε το περιβάλλον `axis` και μαζί και τα υπόλοιπα περιβάλλοντα:

```
\end{axis}
\end{tikzpicture}
\end{document}
```

Αν προσέξετε το διάγραμμα της Εικόνας 1 θα δείτε πως υπάρχουν και κάποιες διακεκομμένες γραμμές. Αυτές σχεδιάζονται με τις παρακάτω εντολές:



Εικόνα 2: Τυπικό παράδειγμα διαγράμματος με δύο γραφικές παραστάσεις.

```
\draw [dotted] (-2,1) -- (3,1);
\draw [dotted] (3,1) -- (3,0);
\draw [dotted] (0,0) -- (5,0);
```

Η εντολή `\draw` είναι εντολή του πακέτου TikZ και την περιγράψαμε στο προηγούμενο άρθρο [1]. Ότι απομένει μπήκε στο διάγραμμα με τις παρακάτω εντολές:

```
\node at (axis cs:3.2,0.05) {\overline{x}};
\node at (axis cs:1.5,0.05) {$e_l$};
\node at (axis cs:4.5,0.05) {$e_r$};
\node at (axis cs:3.8,1) {(3,1)};
\node at (axis cs:3,1) {\bullet};
```

Όπως η εντολή `\draw` έτσι και η εντολή `\node` ορίζεται από το πακέτο TikZ. Επειδή όμως θέλουμε να βάλουμε διάφορα αντικείμενα (π.χ. κείμενο ή μαθηματικές παραστάσεις) σε συγκεκριμένα σημεία του διαγράμματος, χρησιμοποιούμε τον προσδιορισμό `axis cs:`.

Διπλά διαγράμματα

Στην Εικόνα 2 έχουμε ένα διάγραμμα με τις γραφικές παραστάσεις δύο διαφορετικών συναρτήσεων. Όπως και στο προηγούμενο παράδειγμα, η λιστα με τα ζεύγη

Διαγράμματα με το πακέτο *pgfplots*

7

τιμών που χρησιμοποιήθηκαν δημιουργήθηκε από κάποιο εξωτερικό πρόγραμμα. Για να μπορέσουμε να έχουμε δύο γραφικές παραστάσεις, χρησιμοποιήσαμε δύο εντολές `\addplot`:

```
\addplot [black] table[x=x0,y=y1] {minExample.dat};
\addplot [dashed] table[x=x0,y=y2] {minExample.dat};
```

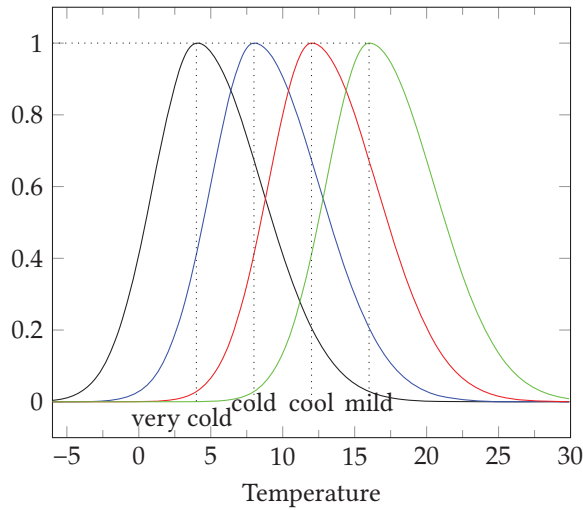
Στον παραπάνω κώδικα, βλέπουμε επίσης πως χρησιμοποιήσαμε το ίδιο εξωτερικό αρχείο δεδομένων, αλλά διαφορετικές στήλες. Παρακάτω φαίνονται τα περιεχόμενα του αρχείου δεδομένων:

```
x0 y1 y2
-2.5 0 0
-2.4 0 0
...
-1.8 0.06666666 0
-1.7 0.1 0
-1.6 0.13333333 0
-1.5 0.16666666 0
...
1 1 1.110223e-15
1.1 0.96666666 0.10000000
1.2 0.93333333 0.20000000
1.3 0.89999999 0.30000000
...
2.7 0.43333333 0.29999999
2.8 0.39999999 0.19999999
2.9 0.36666666 0.09999999
3 0.33333333 0
3.1 0.29999999 0
...
4.3 0
4.4 0
```

Στην Εικόνα 3 βλέπουμε ένα διάγραμμα με πολλές γραφικές παραστάσεις. Ο πιο απλός τρόπος δημιουργίας αυτού του διαγράμματος είναι με τη χρήση τεσσάρων εντολών `\addplot`:

```
\addplot [black] table[x=x0,y=y1] {Lgaussian.dat};
\addplot [blue] table[x=x0,y=y2] {Lgaussian.dat};
\addplot [red] table[x=x0,y=y3] {Lgaussian.dat};
\addplot [green] table[x=x0,y=y4] {Lgaussian.dat};
```

Παρακάτω φαίνονται οι πρώτες πέντε γραμμές του αρχείου `Lgaussian.dat` με τα δεδομένα για τον σχεδιασμό των καμπυλών.



Εικόνα 3: Τυπικό παράδειγμα διαγράμματος με πολλές γραφικές παραστάσεις.

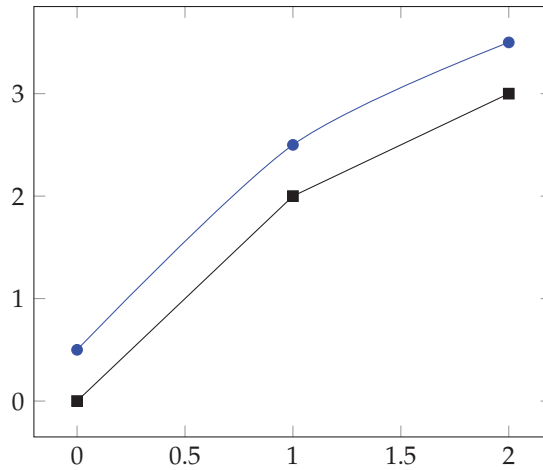
```
x0 y1 y2 y3 y4
-6 0.00387592 1.866446e-05 1.522997e-08 2.100409e-12
-5.9 0.00431784 2.179371e-05 1.859160e-08 2.680547e-12
-5.8 0.00481723 2.541934e-05 2.267002e-08 3.417122e-12
-5.7 0.00536841 2.961521e-05 2.761242e-08 4.351260e-12
```

Όλα οι γραφικές παραστάσεις που παρουσιάστηκαν ως τώρα δημιουργήθηκαν με πάρα πολλά σημεία (ζεύγη τιμών), με αποτέλεσμα να μοιάζουν σαν αληθινές καμπύλες. Αν όμως έχουμε λίγα σημεία, τότε τα σημεία ενώνονται με ευθύγραμμα τμήματα η γραφική παράσταση είναι μια τεθλασμένη γραμμή. Στο διάγραμμα της Εικόνας 4, η πρώτη γραμμή είναι ακριβώς η τεθλασμένη που ενώνει τα λίγα ζεύγη τιμών (τετράγωνα). Η δεύτερη γραμμή είναι μια ομαλή καμπύλη που ενώνει τρία άλλα ζεύγη τιμών (κύκλοι). Η δεύτερη γραμμή σχεδιάστηκε αυτομάτως από το πακέτο `rgfplots` με κάποιο αλγόριθμο προσαρμογής.¹ Ο κώδικας που παράγει αυτές τις γραμμές είναι ο εξής:

```
\addplot [sharp plot,mark=square*] coordinates {(0,0) (1,2) (2,3)};
\addplot [smooth,blue,mark=*] coordinates {(0,0.5) (1,2.5) (2,3.5)};
```

Η παράμετρος `sharp plot` δεν χρειάζεται. Απλά μήκε για να φανεί η διαφορά. Επίσης όταν βάλουμε μια παράμετρο, τότε πρέπει να ορίσουμε και το είδος του συμβόλου που θα χρησιμοποιηθεί για τα σημεία.

¹Για εκείνους που γνωρίζουν από αριθμητική ανάλυση, αξίζει να πούμε ότι ο αλγόριθμος που χρησιμοποιεί το `rgfplots` βασίζεται στις κυβικές καμπύλες Bézier.



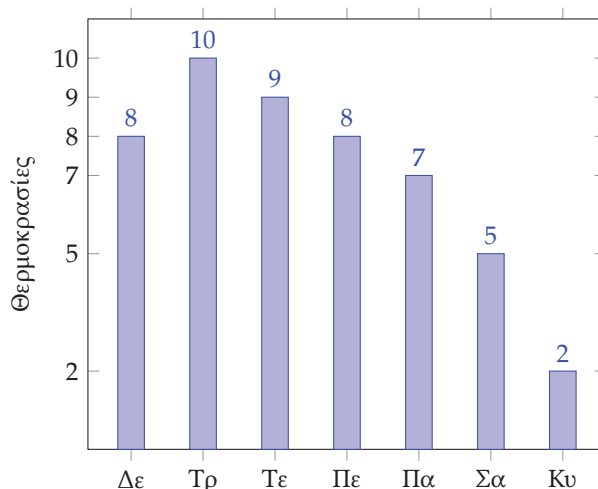
Εικόνα 4: Ένωση σημείων σε τεθλασμένη γραμμή και σε ομαλή καμπύλη γραμμή.

Ραβδογράμματα

Υπάρχουν αρκετά ακόμη είδη διαγραμμάτων, τα οποία μπορεί κάποιος να δει σε διάφορες εφαρμογές λογιστικού φύλλου όπως το Calc, κ.ά. Το πιο χαρακτηριστικό είδος διαγράμματος είναι το ραβδόγραμμα, όπως αυτό της Εικόνας 5. Το ενδιαφέρον σε αυτό το διάγραμμα είναι πως έχουμε *συμβολικές (κατηγορηματικές)* συντεταγμένες μιας και στον οριζόντιο άξονα έχουμε τις ημέρες της εβδομάδας. Ας δούμε τον κώδικα που δημιουργεί αυτό το διάγραμμα:

```
\begin{axis}[ybar, ymin=0,
  ylabel={Θερμοκρασίες},
  symbolic x coords={Δε, Τρ, Τε, Πε, Πα, Σα, Κυ},
  ytick=data,
  nodes near coords, nodes near coords align={vertical},
]
\addplot coordinates {(Δε,8) (Τρ,10) (Τε,9) (Πε,8) (Πα,7) (Σα,5) (Κυ,2)};
\end{axis}
```

Με την παράμετρο *ybar* ορίζουμε πως θέλουμε να έχουμε κατακόρυφες ράβδους (μπάρες). Αν θέλαμε οριζόντιες ράβδους, θα χρησιμοποιούσαμε την παράμετρο *xbar*. Η τέταρτη παράμετρος είναι αυτή που καθορίζει πως ο οριζόντιος άξονας έχει συμβολικές τιμές. Στην περίπτωση μας, οι τιμές αυτές είναι οι ημέρες της εβδομάδας. Οι παράμετροι σε σχόλιο είναι αυτές που βάζουν τους αριθμούς πάνω από τις μπάρες. Αν τις σβήσουμε το διάγραμμα θα είναι αρκετά διαφορετικό από αυτό



Εικόνα 5: Τυπικό παράδειγμα ραβδογράμματος.

που βλέπουμε στην Εικόνα 5. Τέλος, όπως είναι φυσικό, δίνουμε τα ζεύγη τιμών ως όρισμα της εντολής `\addplot`.

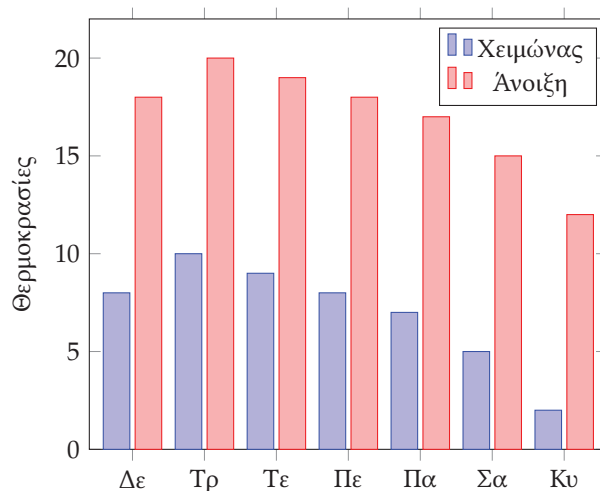
Προσέξτε πως γράφουμε τις συμβολικές τιμές όπως ακριβώς τις δηλώσαμε παραπάνω. Τι γίνεται όμως αν θέλουμε να έχουμε ένα πιο πολύπλοκο διάγραμμα όπως αυτό της Εικόνας 6; Προφανώς πρέπει να χρησιμοποιήσουμε δύο εντολές `\addplot`, όπως στον κώδικα που ακολουθεί:

```
\addplot coordinates {(Δε,8) (Τρ,10) (Τε,9) (Πε,8)
                    (Πα,7) (Σα,5) (Κυ,2)};
\addplot coordinates {(Δε,18) (Τρ,20) (Τε,19) (Πε,18)
                    (Πα,17) (Σα,15) (Κυ,12)};
\legend{Χειμώνας, Άνοιξη}
```

Στην τελευταία γραμμή του κώδικα, βλέπουμε πως υπάρχει μια νέα εντολή: η `\legend`, η οποία δέχεται ως ορίσματα λέξεις ή μικρά κείμενα τα οποία χωρίζονται με κόμμα και τα οποία μπαίνουν στο υπόμνημα του ραβδογράμματος. Το τελευταίο πράγμα που αξίζει να σημειώσουμε είναι πως αν θέλουμε να γράψουμε ολόκληρα τα ονόματα των ημερών, τότε πρέπει να χρησιμοποιήσουμε την παρακάτω παράμετρο:

```
x tick label style={rotate=45,anchor=east}
```

Φυσικά μπορούμε να αλλάξουμε τη γωνία καθώς και τον προσανατολισμό των ονομάτων στον οριζόντιο άξονα.



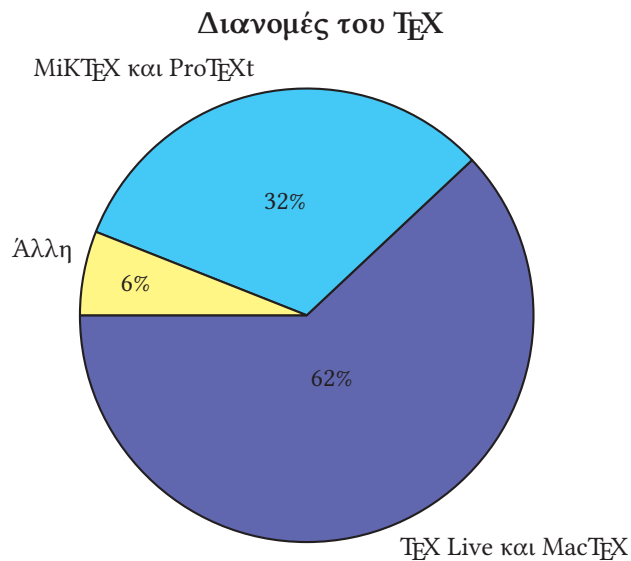
Εικόνα 6: Ραβδογράμμο με πολλές ράβδους.

Κυκλικά διαγράμματα

Το πακέτο *pgfplots* παρέχει πολλές δυνατότητες αλλά δεν μας επιτρέπει να δημιουργήσουμε κυκλικά διαγράμματα, τα γνωστά και ως «διαγράμματα πίτας». Υπάρχει ωστόσο το πακέτο *pgf-pie*, το οποίο έχει σχεδιαστεί για να δημιουργεί μόνο κυκλικά διαγράμματα, όπως αυτό της Εικόνας 7. Ακολουθεί ο κώδικας με τον οποίο δημιουργήθηκε αυτό το διάγραμμα:

```
\begin{tikzpicture}
  \pie [rotate = 180]{
    62/\TeX\ Live και Mac\TeX,
    32/MiK\TeX\ και Pro\TeX t, 6/Άλλη
  };
  \node[above,font=\large\bfseries] at (current bounding box.north)
  {Διανομές του \TeX};
\end{tikzpicture}
```

Η εντολή `\pie` είναι αυτή που δημιουργεί το διάγραμμα. Η παράμετρος `rotate = 180` επιτρέπει την περιστροφή του διαγράμματος κατά 180°. Η παράμετρος `text = legend` δημιουργεί το υπόμνημα, δηλαδή έναν μικρό πίνακα που παρουσιάζει τις αντιστοιχίες τιμών και χρωμάτων. Οι τιμές έχουν τη γενική μορφή `<Αριθμός>/<Κείμενο>` και οι αριθμοί θα πρέπει να έχουν άθροισμα 100.



Εικόνα 7: Κυκλικό διάγραμμα.

Επίλογος

Στο παρόν άρθρο, παρουσιάσαμε διαγράμματα που δημιουργούνται με το πακέτο `pgfplots`. Βεβαίως, ό,τι παρουσιάσαμε εδώ αποτελεί μόνον μια μικρή εισαγωγή στο ζήτημα της δημιουργίας διαγραμμάτων με το \LaTeX . Για περισσότερα, ο αναγνώστης μπορεί να καταφύγει στον οδηγό του πακέτου `pgfplots` [2]. (Σημειώστε πως εδώ χρησιμοποιήσαμε παντού το \XeLaTeX , για την εύκολη εισαγωγή Ελληνικών, αλλά πιστεύουμε πως είναι κατανοητό τι εννοούμε.) Ελπίζουμε να έχετε πεισθεί πως δεν είναι πάντα απαραίτητα τα εξωτερικά προγράμματα για τη δημιουργία πολύ όμορφων διαγραμμάτων.

Αναφορές

- [1] A. Συρόπουλος, «Γραφικά με το πακέτο TikZ/PGF». *Εϋτυπον*, τχ. 34–35 (2016), σσ. 29–43.
- [2] C. Feuersänger, “`pgfplots` – Create normal/logarithmic plots in two and three dimensions,” version 1.15, June 7, 2017. URL: <https://ctan.org/pkg/pgfplots>. (Ανακτήθηκε στις 3 Οκτωβρίου 2017.)