

Grafikos naudojimas \LaTeX

Vaidotas Zemlys

2007 lapkričio 20 d.

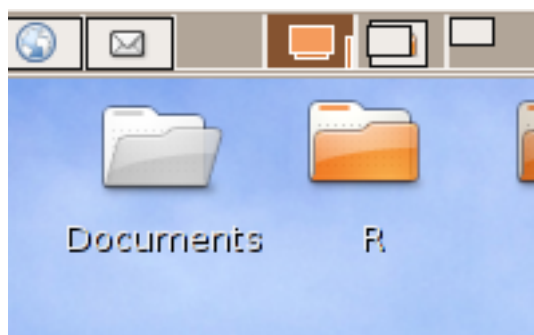
1 Įžanga

Šiame dokumente bus aprašyta, kaip paprastai įtraukti grafinius objektus į savo \LaTeX dokumentą. Taip pat bus aprašyta, kaip eksportuoti grafikus iš programinio paketo R, bei kaip padaryti juose lietuviškas antraštes. Šį dokumentą galima laikyti priedu prie dokumento [1]. Dokumentas yra orientuotas į vartotojus dirbančius su Windows operacine sistema ir įsidiegusius Mi \TeX , bet iš esmės jis tinka bet kuriai operacinei sistemai ir \TeX sistemai. Pastebėtus netikslumus ar iškilusius klausimus siųskite adresu vzdestymas@gmail.com. Šio dokumento `.tex` failą galite rasti mano puslapyje.

2 Grafikos importavimas į \LaTeX

Grafinių paveikslų įkėlimas į \LaTeX realizuojamas `graphicx` paketo pagalba (Išsamiai jo galimybės aprašytos [2]). Pagrindinė jo komanda yra `\includegraphics`. Žemiau esantis paveikslėlis buvo įdėtas tokios komandos pagalba:

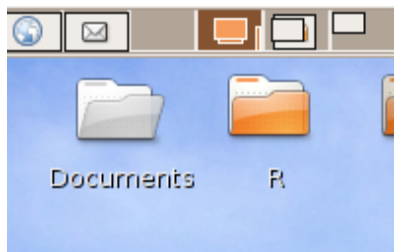
```
\begin{center}  
  \includegraphics{Capture6.png}  
\end{center}
```



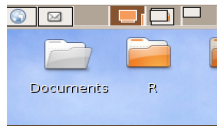
Į \LaTeX dokumentus galima įdėti EPS tipo paveikslus, jeigu dokumentus kompiliuoja su `latex` komanda, ir JPG, PNG, PDF tipo paveikslus, jeigu dokumentus kompiliuoja su `pdflatex` komanda. Panaudojus ne tą komandą dokumentui kompiliuoti, bus išmesta klaida.

JPG ir PNG tipo paveiksai yra „bitmap“ tipo, t.y. juos sudaro taškų matrica. Kiekvienas taškas turi savo spalvą ir juos išdėliojus kaip matricą gauname paveikslėlį. Kadangi kompiuterio ekrane taškas paprastai yra labai mažas, žmogaus akis mato tolydų vaizdą. Bet kuri programa, rodanti JPG ir PNG tipo paveikslus nurodo, kiek taškų yra rodomame paveiksle. Paprastai nurodoma pavidalu 800×600 , kas reiškia, kad paveikslo ilgis yra 800 taškų, o aukštis 600. Kadangi \LaTeX dokumente viskas dėliojama ant įsivaizduojamo popieriaus lapo, kurio dydis yra $297mm \times 210mm$ (A4 formato lapas), tai taškus reikia kažkaip konvertuoti įprastus ilgio vienetus. Tai daroma, laikant, kad viename colyje telpa 72 taškai. Taigi aukščiau įdėto paveikslo kurio išmatavimai taškais yra 200×125 , išmatavimai tampa $71mm \times 44mm$. Toks konvertavimas nevisados tinka, jeigu paveikslo išmatavimas taškais yra didelis. Pavyzdžiui nuotraukų padarytų skaitmeniu fotoaparatu išmatavimai taškais paprastai būna 3072×2034 , taigi jos išmatavimai \LaTeX dokumente taptų $1084mm \times 718mm$ ir paprasčiausiai netilptų į puslapį. Įdedant paveikslą su komanda `\includegraphics` galima kontroliuoti paveikslo išmatavimus. Tai daroma šitaip

```
\begin{center}
  \includegraphics[width=52mm,height=33mm]{Capture6.png}
\end{center}
```



Nurodžius paveikslo ilgį ir plotį \LaTeX atitinkamai paveikslą išstampys. Kad paveikslas neišsikraipytų reikia išlaikyti tą patį santykį tarp ilgio ir aukščio milimetrais, koks buvo taškais. Kokį ilgį ir plotį nurodyti priklauso nuo to, kam dokumentas bus naudojamas. Jeigu skaityti kompiuterio ekrane, kurių raiška paprastai būna apie 100 taškų colyje, grafiko, kurio ilgis yra 100 taškų, nurodžius ilgį $25.4mm$ (plotį atitinkamai), paveikslas, atsidarius PDF dokumentą ir nustatčius 100% zoom, atitinkamai ir užims 100 taškų. Taigi Jeigu jūsų kompiuterio ekrane jis gerai atrodė prieš įdedant į \LaTeX dokumentą, tai jis gerai atrodys ir gautame PDF faile. Jeigu dokumentas bus spausdinamas, tai spausdintuvuose paprastai galima nurodyti, kokia raiška bus spausdinama. Paprastai, tai būna 300, 600 taškų į colį. Taigi, kad paveikslas gerai atrodytų



1 pav.: Nemodifikuotas paveikslėlis

atspausdintas, jo dydis turi būti atitinkamas. Neteisingai nurodžius išmatavimus, atspausdinus, paveikslas gali atrodyti neryškus, arba aiškiai matytis taškai.

Tiesiogiai įdedant paveikslus į dokumentą su `\includegraphics` komanda, \LaTeX įterpia juos, ten kur ta komanda sutinkama. Tekstas prieš ir po grafiko `.tex` faile, lygiai taip pat bus prieš ir po grafiko PDF dokumente. Tai nėra visados pageidaujama, nes paveikslui netilpus į puslapį jis bus nukeltas į kitą puslapį ir tekste atsiras tarpas. To išvengti padeda komanda `figure`:

```
\begin{figure}[t]
  \begin{center}
\includegraphics[width=28mm,height=16mm]{Capture6.png}
  \caption{Nemodifikuotas paveikslėlis}
\label{fig:notmod}
  \end{center}
\end{figure}
```

Jos pagalba \LaTeX traktuoja grafiką kaip „plaukiojantį“ objektą ir pagal tam tikras taisykles apskaičiuoja jo vietą. Jas galima šiek tiek kontroliuoti nurodant parametrus laužtiniuose skliaustuose. Paveikslui 1 buvo nurodytas `t` reiškiantis *top*, t.y. \LaTeX privalėjo bandyti dėti paveikslą viršuje.

Kiti parametrai yra `h` - reiškia *here*, dėti paveikslą čia, `b` - *bottom*, paveikslas apačioje ir `p` - *page*, paveikslas atskirame puslapyje, kuriame yra tik „plaukiojantys“ objektai. Šiuos nustatymus galima kombinuoti, \LaTeX tada bando pritaikyti taisykles eiliškumo tvarka, kol „randa“ tinkamą vietą paveikslui.

Tik su `figure` komanda galima nurodyti paveikslo pavadinimą ir naudoti jį citavimui tekste.

3 Grafikų eksportavimas iš R

Programinis paketas R gali pateikti grafikus visais formatais, kuriuos galima įdėti į \LaTeX dokumentą, t.y. PDF, PS, JPG ir PNG. Eksportavimas iš šių formatų atliekamas funkcijų `pdf`, `postscript`, `jpeg` ir `png` pagalba. Jos turi įvairių nustatymų, mes detalai aptarsime tuos, kurie svarbūs įkeliant grafikus į \LaTeX dokumentus.

3.1 Postscript grafika

Kadangi postscript formatas \LaTeX 'e vaidina gana svarbų vaidmenį, tai visų pirma aptarsime jį. Su kitais formatais viskas bus panašiai. Norint peržiūrėti postscript grafikus, reikia įsidiesti atitinkamas programas. Jos plačiau aprašytos 4 skyriuje. Kadangi postscript grafikus galima paprastai konvertuoti į pdf (kaip tai padaryti aprašyta vėliau) iš principo papildomų programų diegti nereikia.

Grafikos eksportavimą R'e galima realizuoti keliais būdais. Visos grafinės komandos R'e yra atskirtos nuo grafinio formato. Pagal nutylėjimą parašius komandą

```
> plot(rnorm(100))
```

R iš pradžių atidaro Windows langą ir jame pradeda dėti reikalingus grafinius objektus: x, y ašis, taškus ir t. t. Jeigu norime gauti grafiką kitu formatu reikia atitinkamai „atidaryti“ atitinkamo formato „langą“. Tai daroma su atitinkama komanda. Postscript atveju tai būtų šitaip:

```
> postscript(file="test.ps")
```

Gavęs tokią komandą R visas grafinės komandas realizuoja faile `test.ps` (Failas yra padedamas į katalogą, kuriame dirba R, koks tai katalogas galite sužinoti su komanda `getwd()`, o jį nusistatyti su komanda `setwd()`). Atlikus visus norimus grafinius veiksmus, R reikia tai pasakyti. Tai daroma su komanda `dev.off()`. Taigi elementariausia veiksmų seka postscript formato grafiko gavimui yra tokia:

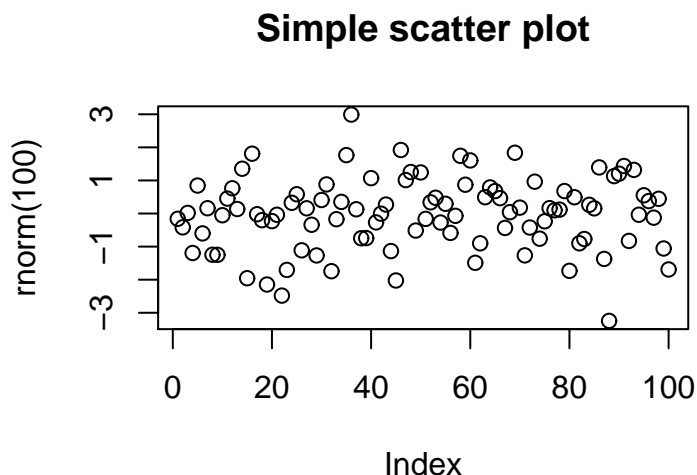
```
> postscript(file="test.ps")
> plot(rnorm(100))
> dev.off()
```

Jeigu norime gauti grafiką kitu pavidalu, vietoj `postscript` reikėtų rašyti atitinkamai `pdf`, `jpeg` arba `png`. Svarbiausiai neužmiršti po grafiko sukūrimo, duoti komandą `dev.off()`. Negavęs tos komandos R'as niekuomet neuždarys failo, netgi jei darant grafiką gavosi klaida. Taigi geriausiai visų pirma įsitikinti, kad grafikas gaunasi toks kokio norite, išsisaugoti veiksmų seką, kurios pagalba jis sukuriamas, ir tada grafiką eksportuoti.

Pagal nutylėjimą R'as sukuria postscript failą, kuriame yra vienas puslapis, su grafiku apatiniame kampe. Įdėjus tokį postscript failą į \LaTeX dokumentą, dokumente atsiras papildomas puslapis su tuo grafiku. Norint įdėti grafiką, kad jis užimtų tik dalį puslapio reikia nurodyti tam tikrus parametrus:

```
> postscript(file="ssplot.ps",horizontal=FALSE,onfile=TRUE,paper="special",
width=4,height=3)
```

Parametrai `horizontal`, `onfile` ir `paper` nurodo R, kad bus kuriamas specialus, ne puslapio formato, postscript failas (jis vadinamas EPS, Encapsulated PostScript), o `width` ir `height` nurodo plotį ir aukštį coliais (1 colis = 2,54cm). Gautas grafikas bus lygiai tokio dydžio, kaip nurodyta, jokios rezoliucijos tam įtakos nedarys. Kadangi A4 formato išmatavimai yra 297mm × 210mm iš karto galima numatyti, kiek eksportuotas grafikas užims vietos.



2 pav.: Paprastas grafikas

Grafikas 2 buvo gautas tokių komandų pagalba:

```
> postscript(file="ssplot.ps",horizontal=FALSE,onfile=TRUE,paper="special",
width=4,height=3)
> plot(rnorm(100),main="Simple scatter plot",pch=1)
> dev.off()
```

Jeigu naudojame PDF \LaTeX postscript grafikų į \LaTeX dokumentų dėti negalima. Bet tai nėra bėda, nes MiK \TeX turi pagalbines komandas `epstopdf` kuri postscript grafikus konvertuoja į pdf. Ją galima iškviešti tiesiog iš R:

```
> system("epstopdf ssplot.ps")
```

ir iš `ssplot.ps` failo bus padarytas `ssplot.pdf` failas kurį jau galima įtraukti į dokumentą kompiliuojamą su PDF \LaTeX .

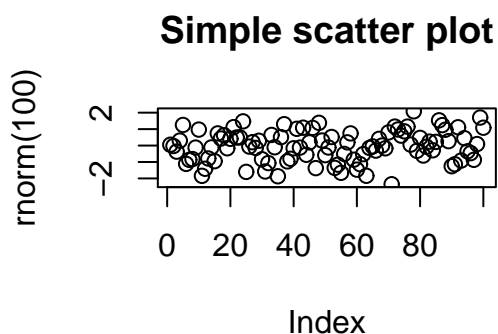
R'o įprastiniai grafiniai nustatymai yra skirti 8in × 6in dydžio grafikams. Jeigu grafikai yra mažesni, kaip pavyzdžiui mūsų atveju, jie gali atrodyti pernelyg suspausti. Tam ištaisyti reikia paprasčiausiai pakeisti kai kuriuos pradinius parametrus:

```
> par(mar=c(4,4,2,2),cex=0.6)
```

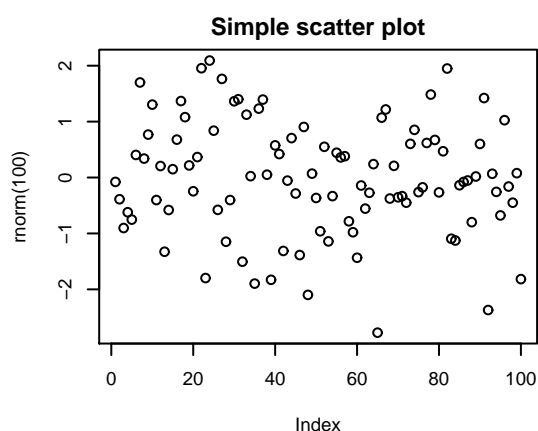
Šią komandą reikia parašyti prieš komandą `postscript`. Parametras `mar` kontroliuoja paraščių dydį, o parametras `cex` grafinio objekto dydį. Grafikai 3 ir 4 buvo sukurti su tom pačiom komandom (aišku tik nurodant skirtingą failo pavadinimą):

```
> postscript(file="ssplot.ps",horizontal=FALSE,onefile=TRUE,paper="special",
width=3,height=3*3/4)
> plot(rnorm(100),main="Simple scatter plot",pch=1)
> dev.off()
```

tik 4 grafikui prieš `postscript` komandą buvo iškviesta `par(mar=c(4,4,2,2),cex=0.6)`.



3 pav.: Įprastiniai nustatymai



4 pav.: `par(mar=c(4,4,2,2),cex=0.6)`

Akivaizdu, kad grafikas 4 kur kas informatyvesnis.

Kokias parametrų `par` ir `cex` reikšmes pasirinkti priklauso nuo kokio dydžio grafiką jūs norite įdėti į savo dokumentą. Prieš dedant grafiką patartina visados paeksperimentuoti su įvairiais dydžiais ir parametrų nustatymais.

Kiekvieną kartą nurodinėti visus komandos `postscript` parametrus yra gana daug pasikartojančio darbo, todėl naudinga susikurti funkciją, kuri tą darbą už jus padarytų. Vieną tokios funkcijos variantą pateiksime čia:

```
d2ps <- function(file="Rplot%d.ps",height=6,width=8,which=dev.cur(),...) {
  if(which!=dev.cur()) {
    cur <- dev.cur()
    dev.set(which)
  }
  else cur <- NULL
```

```

dev.copy(device=postscript,file=file,height=height,width=width,
horizontal=FALSE,onefile=TRUE,paper="special",...)
dev.off()
if(!is.null(cur))dev.set(cur)
else dev.set(which)
}

```

Su šia funkcija darbas vyksta kur kas paprasčiau veiksmų seka:

```

> postscript(file="ssplot.ps",horizontal=FALSE,onefile=TRUE,paper="special",
width=3,height=3*3/4)
> plot(rnorm(100),main="Simple scatter plot",pch=1)
> dev.off()

```

keičia tokia seka:

```

> plot(rnorm(100),main="Simple scatter plot",pch=1)
> d2ps(file="ssplot.ps",width=3,height=3*3/4)

```

Nereikia rašyti komandos `dev.off()` ir grafiką galima apžiūrėti atsidariusiame lange, lyg vyktų įprastinis grafiko žiūrėjimas su R. (Šita funkcija taip pat sėkmingai veikia ir po Linux).

Norint gauti lietuviškas raides grafikuose reikia pasakyti R'ui, kad norime lietuvių kalbos (Linux sistemose su unikodine lokale šitas žingsnis nereikalingas):

```

> Sys.setlocale(locale="lithuanian")

```

ir nurodyti atitinkamus `postscript` komandos parametrus:

```

> postscript(file="sslt.ps",enc="ISOLatin7",family="URWHelvetica",
horizontal=FALSE, onefile=TRUE,paper="special",width=3,height=3*3/4))

```

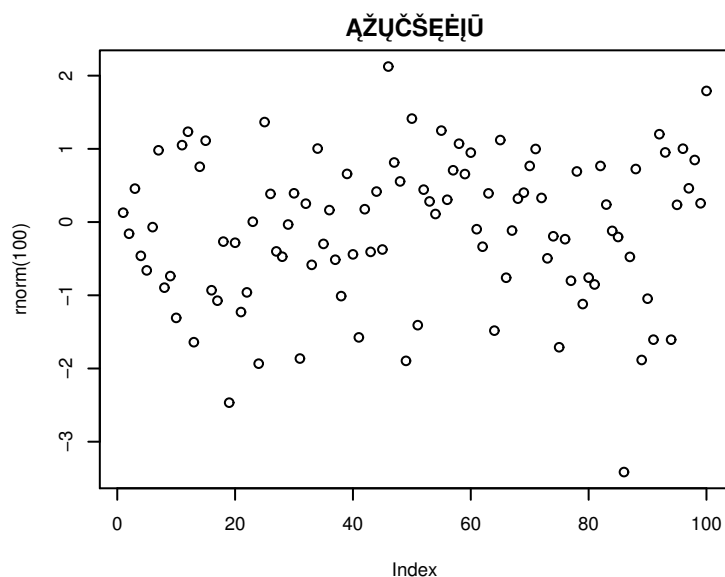
Tada grafike reikia paprasčiausiai naudoti lietuviškas raides:

```

> plot(rnorm(100),main="ĄŽŪČŠĘĮŪ",pch=1)

```

Rezultatą matome grafike 5. Išspaudžia ašarą kiekvienam tikram Lietuvos patriotui. Dėl smulkios klaidos, senesnėse nei 2.6.0 R'o versijose raidė ū gali būti atvaizduota neteisingai. Klaidą galima ištaisyti parengavus `ISOLatin7.enc` failą `grDevices` paketo kataloge, arba įsidiegus naujesnę R versiją.



5 pav.: Lietuviškos raidės!

4 Postscript peržiūra

Postscript failuose yra surašytos Postscript kalbos komandos. Jų interpretavimui reikia šios kalbos interpretatoriaus. Vieną iš interpretatorių Ghostscript parsisiųsti galima iš šio puslapio. Reikia siųstis failą `gs860w32.exe` yra ir atitinkama 64 bitų versija. Postscript failų grafinei peržiūrai reikia įsidiegti programą GSview. Ją galima parsisiųsti iš šio puslapio. Reikia siųstis failą `gsv49w32.exe`. Ir šiai programai egzistuoja 64 bitų versija.

Parsisiuntus ir įsidieigus šias programas (visų pirma Ghostscript, o tik po to GSview!) paspaudus ant ps failų, juos automatiškai turėtų atverti su GSview. GSview programą galima taip pat naudoti PDF failų peržiūrai.

Literatūra

- [1] Zemlys, V. „MiKTeX 2.6 įdiegimas ir sulietuvinimas“
- [2] Reckdahl, K. „Using Imported Graphics in LaTeX and pdfLaTeX“

5 Disclaimer

Šis dokumentas nepretenduoja į išsamų L^AT_EX grafikos galimybių aprašymą. Taip pat jame gali būti įvairių stiliaus, skyrybos ar kitų klaidų. Jei jos pabudino jumyse načį lietuvi kalbininką imkite išeities tekstą ir jas ištaisykite. Pažadu pataisymus įtraukti, ir jus paminėti. Kiekvienas pavyzdys buvo patikrintas kelis kartus, taigi yra didelė tikimybė, kad jie veiks ir jums. Šį dokumentą galima laisvai kopijuoti ir platinti.