

## ATK tähtitieteessä

### Osa 5 - IDL datan sovitus ja muita ominaisuuksia

3. toukokuuta 2017



## IDL - esimerkiksi linfit

- ▶ Sovitetaan esimerkiksi suora datapisteisiin joihin lisätään keinotekoisesti hälyä.:

```
IDL> x=findgen(50)/5.  
IDL> y=x+randomn(seed,50)  
IDL> sovitus=linfit(x,y)  
IDL> print,sovitus  
0.264205 0.989497  
IDL> sovitus_y=sovitus(0)+sovitus(1)*x  
IDL> sovitus_x=x  
IDL> pisteplot=plot(x,y,symbol='square',linestyle='none')  
IDL> sovitusplot=plot(sovitus_x,sovitus_y,symbol='none',,$  
IDL> color='red',thick=3,/overplot)
```

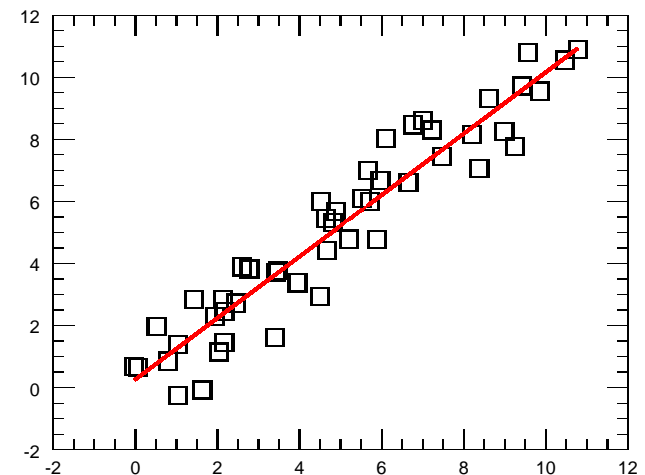


## IDL - datan sovitus

- ▶ IDL sisältää monia yleisimpiä funktioita, joita voi helposti sovittaa datapisteisiin.
- ▶ Jos valmiista funktioista ei löydy mieleistä, voi oman mielivaltaisen sovitusfunktion myös määrittellä.
- ▶ Esimerkiksi:
  - ▶ `linfit` – sovittaa suoran datapisteisiin.
  - ▶ `poly_fit` – n:n asteen polynomisovitus.
  - ▶ `curvefit` – oman erikseen määriteltävän funktion sovitus.
- ▶ IDL manuaalista löytyy kattavat ohjeet näiden käytölle.



## IDL - esimerkiksi linfit



## Kuvaajien tallentaminen tiedostoon, plot-proseduuri

- ▶ Yleensä pelkkä kuvaajien katselu näytöltä ei riitä, vaan ne halutaan myös tallentaa myöhempää käyttöä varten.
- ▶ Plot-proseduurilla tehdyt kuvaavat on mahdollista tallentaa .ps (post script) tiedostoiksi. Tämä tiedostomuoto on ikään kuin .pdf-tiedostojen yksinkertaisempi muoto.
- ▶ Tallentaminen tapahtuu seuraavasti:
  - ▶ Määritellään piirto ps-tiedostoon:  
`set_plot, 'ps'`
  - ▶ Nimetään tiedosto ja määrätään mahdollisia lisäparametreja:  
`device, filename='tiedosto.ps', /landscape, /color`
  - ▶ Piirretään kuvaaja kuten normaalisti:  
`plot, x, y`
  - ▶ Suljetaan tiedosto:  
`device, /close`
  - ▶ Palautetaan piirto ikkunaan:  
`set_plot, 'x'`



## Kuvaajien tallentaminen tiedostoon, plot-proseduuri

- ▶ Valmiin tiedoston voi muuttaa Linux-komentorivillä pdf-tiedostoksi komennolla:
  - ▶ `ps2pdf tiedosto.ps`
- ▶ Tai vaikka png-kuvatiedostoksi:
  - ▶ `convert tiedosto.ps tiedosto.png`
- ▶ Kuten saattoi paljastua tämä tapa ei ole kovin joustava. Saatavilla monia apuohjelmia tämän vastaavan prosessin tekoon, mutta niiden käytöstä vasta myöhemmillä kursseilla.



## Kuvaajien tallentaminen tiedostoon, plot()-funktio

- ▶ Käyttämällä plot()-funktion interaktiivista ikkunaa on kuvaaja mahdollista tallentaa suoraan graafisten valikoiden kautta.
- ▶ Toinen vaihtoehto on käyttää seuraavia komentoja:
  - ▶ Kuten normaalisti ohjataan plot()-funktio johonkin muuttujaan:  
`a_plot=plot(x,y)`
  - ▶ Nyt kuvaaja voidaan tallentaa suoraan käyttäen komentoa muuttuja.save. Esimerkiksi:  
`a_plot.save, 'tiedosto.pdf', /landscape`
- ▶ Molemmat tavat ovat siis huomattavasti yksinkertaisempia kuin plot-proseduuria käytettäessä. Tämän takia onkin syytä välillä miettiä kumpaa plot-komentoa käyttää kuvaajien tekemiseen.



## Monta kuvaajaa samaan ikkunaan, plot-proseduuri

- ▶ On myös varsin käyttökelpoista saada useampi kuvaaja samaan ikkunaan.
- ▶ Käytettäessä plot-proseduuria, voidaan ikkuna jakaa useampaan elementtiin käyttäen IDL systeemimuuttujaa, jonka syntaksi on seuraava:  
`!p.multi=[indeksi mihin piirretään ensimmäisenä, kuvaajien lukumäärä horisontaalisesti, kuvaajien määrä vertikaalisesti]`
- ▶ Eli jos haluamme esimerkiksi neljä kuvaajaa samaan ikkunaan 2x2 muotoon, ja piirto aloitetaan vasemmasta yläkulmasta voidaan asettaa:  
`!p.multi=[0, 2, 2]`
- ▶ On muistettava aina palauttaa oletus takaisin, eli yksi kuvaaja per ikkuna:  
`!p.multi=0`



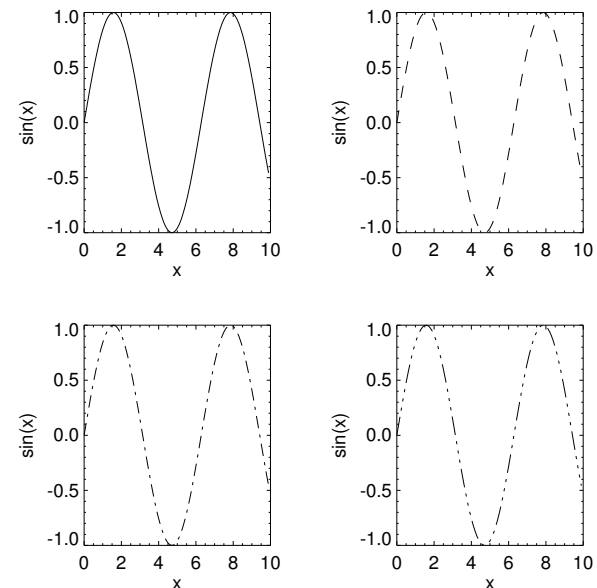
## Monta kuvaajaa samaan ikkunaan, plot-proseduuri

- ▶ Esimerkiksi neljä kuvaajaa samaan ikkunaan:

```
IDL> x=findgen(100.)/10. & y=sin(x)
IDL> window,/free,xsize=800,yysize=800
IDL> !p.multi=[0,2,2]
IDL> plot,x,y,xtitle='x',ytitle='sin(x)'
IDL> plot,x,y,xtitle='x',ytitle='sin(x)',linestyle=2
IDL> plot,x,y,xtitle='x',ytitle='sin(x)',linestyle=3
IDL> plot,x,y,xtitle='x',ytitle='sin(x)',linestyle=4
IDL> !p.multi=0
```

◀ ▶ ⏪ ⏩ 🔍

## Monta kuvaajaa samaan ikkunaan, plot-proseduuri



◀ ▶ ⏪ ⏩ 🔍

## Monta kuvaajaa samaan ikkunaan, plot()-funktio

- ▶ Plot()-funtiota käytettäessä monen kuvaajan ikkuna määritellään plot()-kutsussa käyttämällä layout-avainsanaa.
- ▶ Tämän syntaksi on seuraava:  
`layout=[kuvaajien lukumäärä horisontaalisesti,  
kuvaajien määrä vertikaalisesti,  
indeksi mihin piirretään]`
- ▶ Tämän lisäksi on käytettävä avainsanaa /current ohjaamaan komento aktiiviseen ikkunaan.
- ▶ Nyt verrattuna !p.multi-parametriin, indeksit lasketaan eri tavalla:
  - ▶ !p.multi – vasemman yläkulman indeksi on 0.
  - ▶ /layout – vasemman yläkulman indeksi on 1.

◀ ▶ ⏪ ⏩ 🔍

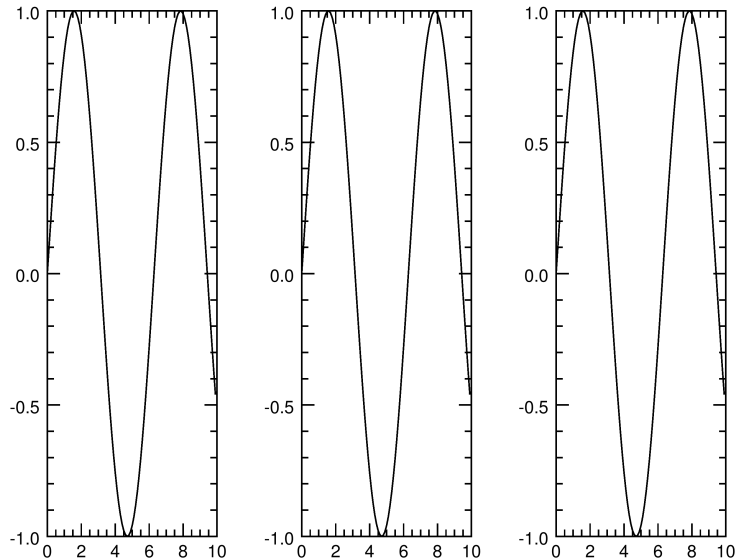
## Monta kuvaajaa samaan ikkunaan, plot()-funktio

- ▶ Esimerkiksi kolme kuvaajaa samaan ikkunaan vierekkäin:

```
IDL> x=findgen(100.)/10. & y=sin(x)
IDL> a.plotti=plot(x,y,layout=[3,1,1])
IDL> b.plotti=plot(x,y,layout=[3,1,2],linestyle='dash',/current)
IDL> c.plotti=plot(x,y,layout=[3,1,3],linestyle='dot',/current)
```

◀ ▶ ⏪ ⏩ 🔍

## Monta kuvaajaa samaan ikkunaan, plot()-funktio



## Tekstin lisääminen kuvaajiin

- ▶ Tekstin lisääminen plot-proseduurin kuvaajiin komennolla:
  - ▶ `xyouts, xkoordinaatti, ykoordinaatti, 'teksti'`
  - ▶ Käytännöllinen avainsana on `/data`, jolloin koordinaatit ovat kuvaajan data-arvoina. Muutoin koordinaatit ovat välillä `[0,1]` eli ns. ikkunan sisäisinä koordinaatteina.
  - ▶ Esimerkiksi kohtaan `x=1, y=2` suuremmalla fontilla:  
`xyouts, 1, 2, 'teksti', /data, size=4`
- ▶ Vastaavasti `plot()`-funktionalle jälleen oma komento:
  - ▶ `teksti=text(xkoordinaatti, ykoordinaatti, '$teksti')`
  - ▶ Jälleen voi käyttää avainsanaa `/data`.
  - ▶ Esimerkiksi kohtaan `x=1, y=2` suuremmalla fontilla:  
`omateksti=text(1, 2, 'teksti', /data, $font_size=6.)`



## Datasetilyksen lisääminen kuvaajiin

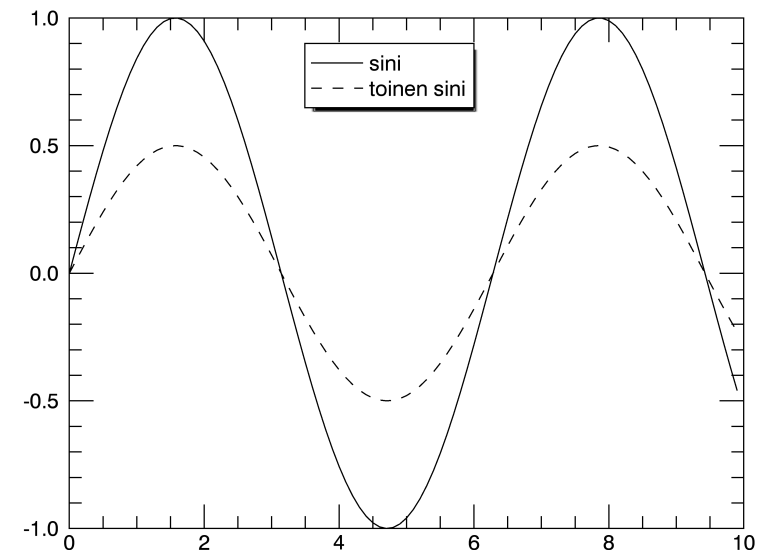
- ▶ Plot-proseduurille ei ole olemassa helppoa valmista IDL:n mukana tulevaa tapaa lisätä datasetilystä kuvaajiin.
- ▶ Toisaalta `plot()`-funktionalle tämä on helppoa:
  - ▶ Luodaan ensin kuvaajat ja nimetään ne käyttäen `name`-avainsanaa:

```
IDL> x=findgen(100.)/10. & y=sin(x)
IDL> a=plot(x,y,name='sini')
IDL> b=plot(x,y/2.,linestyle='dash',/overplot, $
      name='toinen sini')
```
  - ▶ Ja selitys voidaan lisätä `legend`-funktionalle, jossa määritellään kohdekuvaajat ja selityslaatikoiden paikka. Esimerkissä paikka on ilmoitettu jälleen datayksiköissä:

```
IDL> c=legend(target=[a,b],position=[6.,0.9],/data)
```



## Datasetilyksen lisääminen kuvaajiin



## NASA IDL Astro kirjasto

- ▶ NASA:n ylläpitämä yleisten tähtitieteellisen IDL-ohjelmien kirjasto löytyy osoitteesta:
  - ▶ <http://idlastro.gsfc.nasa.gov/>
- ▶ Kirjasto sisältää monia hyödyllisiä apuohjelmia muun muassa datan käsittelyyn ja sitä päivitetään jatkuvasti.
  - ▶ Harjoituksissa haette kirjastosta ohjelmat 'glactc.pro' ja 'bprecess.pro'. Nämä ohjelmat muuttavat rektaskension ja deklinaation galaktisiin koordinaatteihin.
  - ▶ Myöhemmillä kursseilla harjoituksissa käytetään kirjaston ohjelmia mahdollisesti enemmänkin.