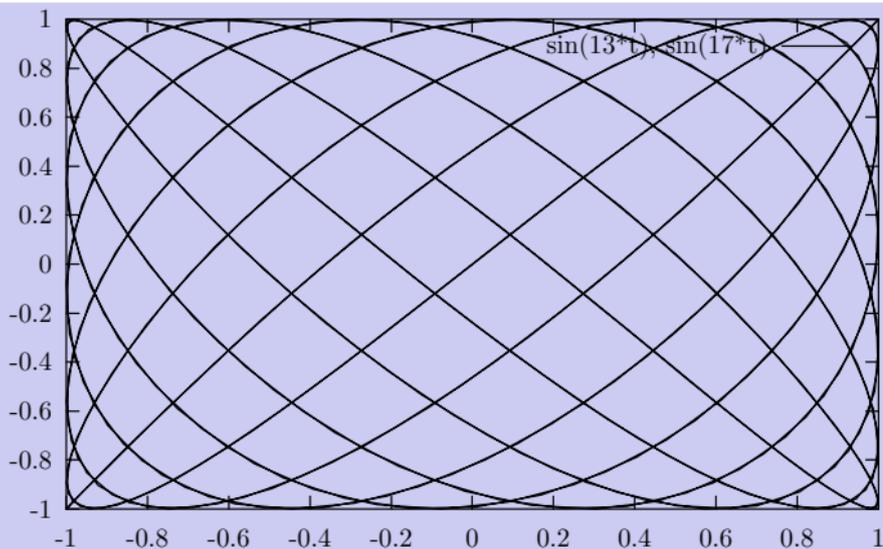


Gnuplot



einstiegsfreundlich

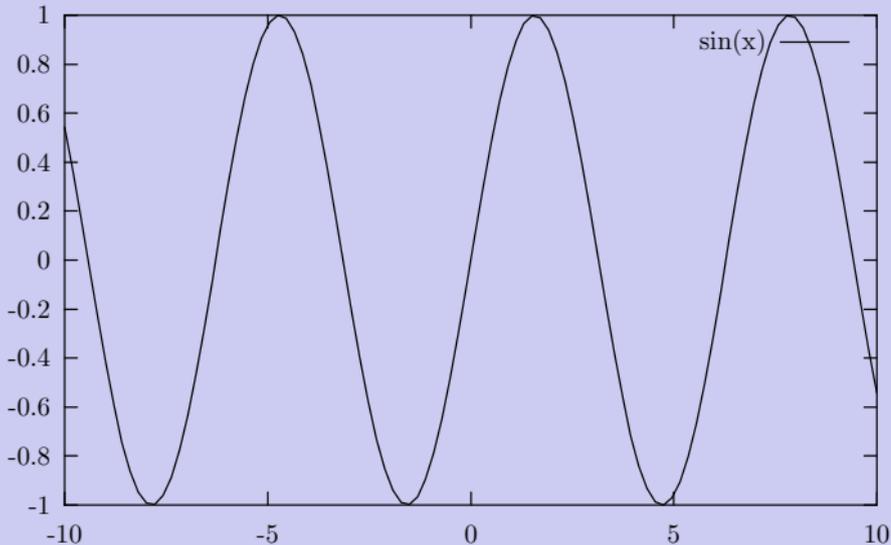
Zu starten durch Eingabe von `gnuplot`.

Angezeigter Prompt:

```
gnuplot>
```

Zum Beispiel eingeben:

```
gnuplot> plot sin(x)
```

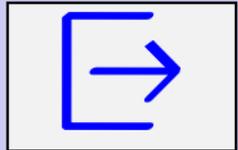
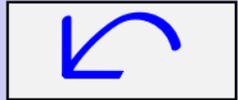
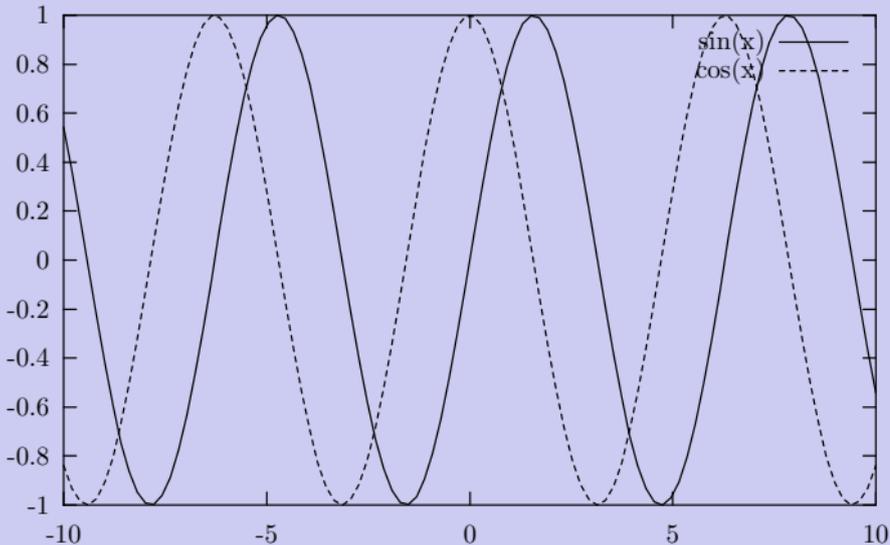


mehrere Plots auf ein Bild

Ab jetzt Befehle ohne Prompt angezeigt.

```
plot sin(x), cos(x)
```

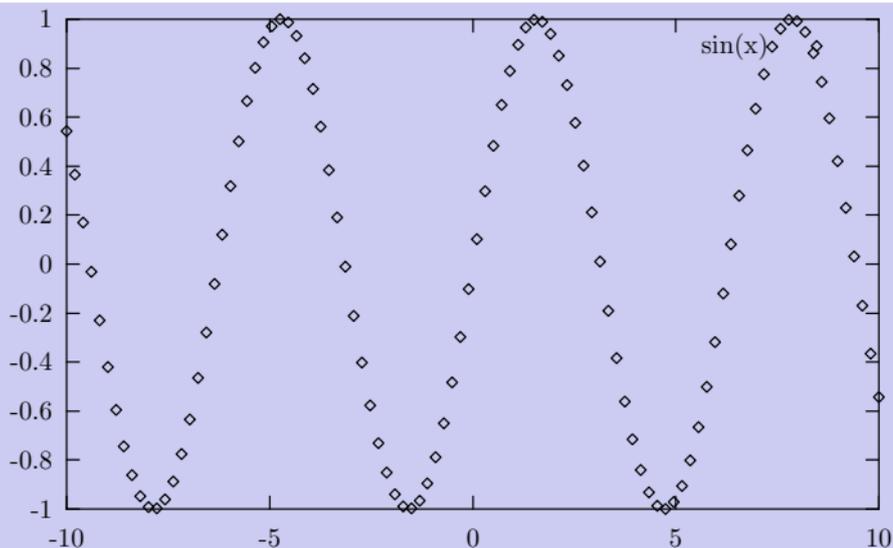
So lassen sich zwei Plots in ein System bringen.



Anzeigen in einzelnen Punkten

Man kann natürlich Plots auch in einzelnen Punkten setzen. Hierbei wird deutlich, in welchem Abstand Gnuplot die Stützstellen setzt.

```
plot sin(x) with points
```

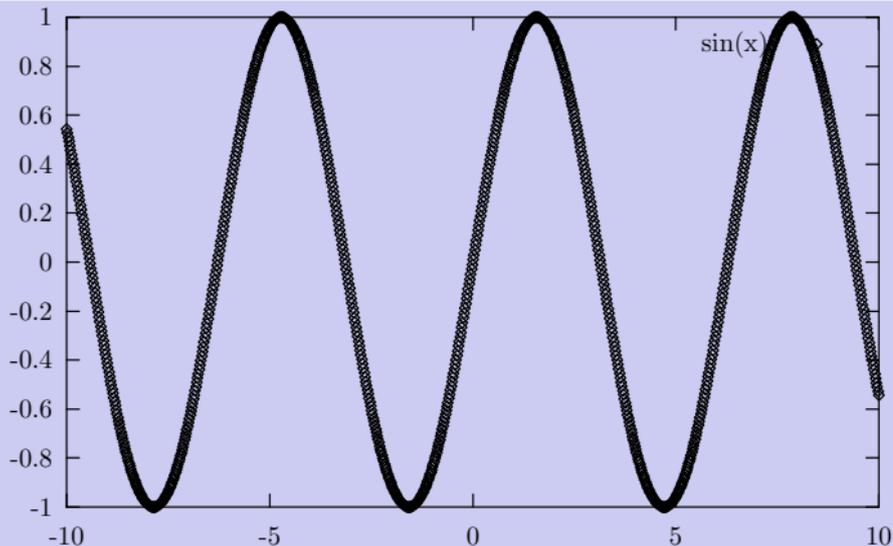


Anzeigen in vielen Punkten

Die Anzahl der Stützstellen lässt sich natürlich auch erhöhen, wenn einem der Plot zu eckig erscheint.

```
set sample 1000  
replot
```

Der letzte Befehl wiederholt den letzten Plot mit den veränderten Einstellungen.

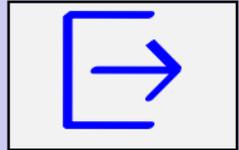
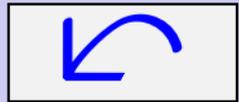
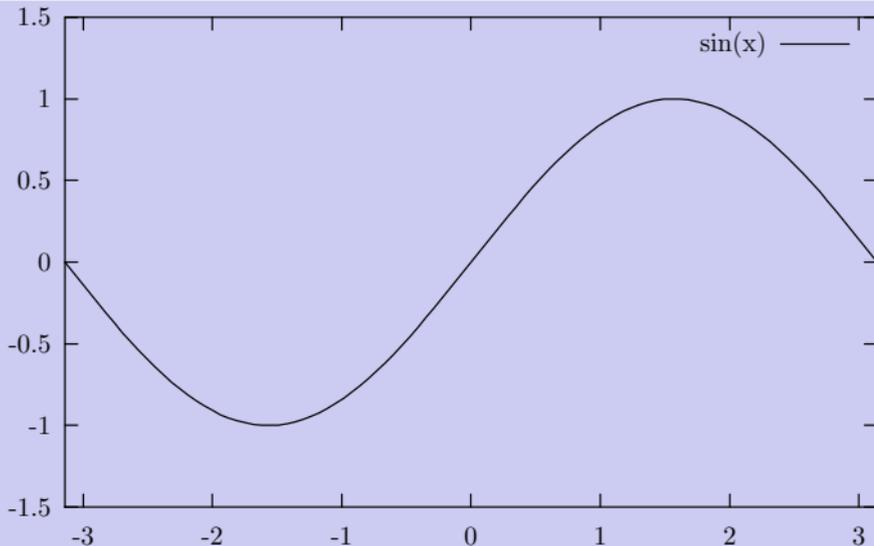


Achsenskalierung ändern

Der Ausschnitt aus dem Plot lässt sich definieren, indem vor der Funktion die Achsenskalierung in eckigen Klammern kommt.

```
plot [-pi:pi][-1.5:1.5] sin(x)
```

Hierbei sieht man auch, dass `pi` vordefiniert ist. Nur die x -Achse wird mit nur einem eckigen Klammerpaar gesetzt, wenn man nur die y -Achse vorgeben will, kann man das mit `plot [] [-1.5:1.5] sin(x)` tun.

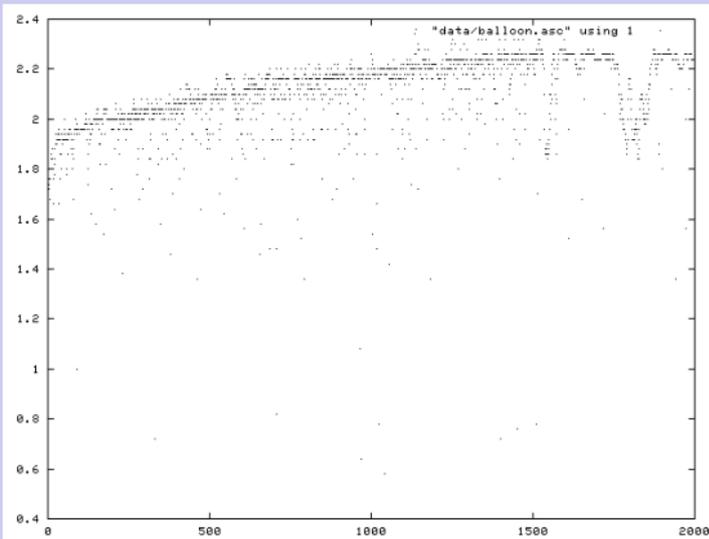


Daten ausgeben

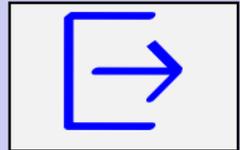
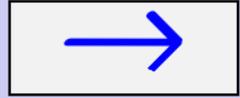
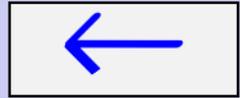
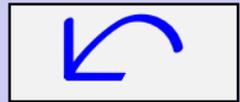
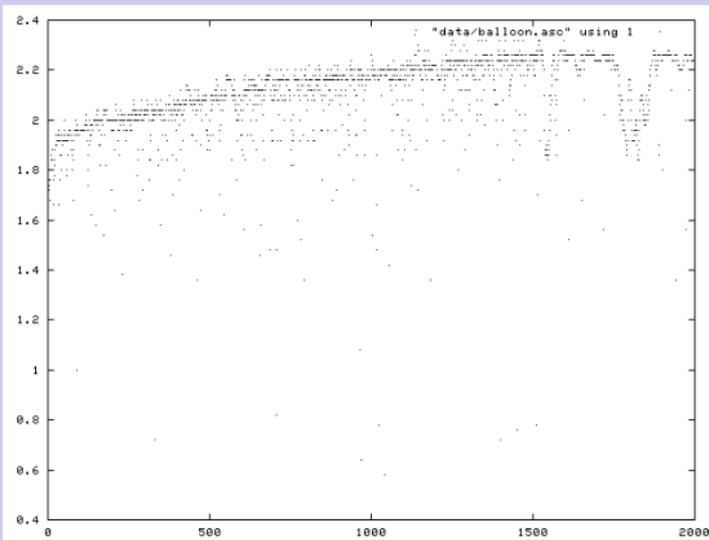
Daten können aus einer Datei ausgelesen werden. Sie sollten dort im ASCII-Tabellenformat stehen.

Beispiel: Ein Wetterballon in der Stratosphäre steigt langsam an und misst Strahlungsdaten zu aufeinanderfolgenden Zeitpunkten. Diese Messreihe soll dargestellt werden.

```
plot "data/balloon.asc" using 1 with points 0
```

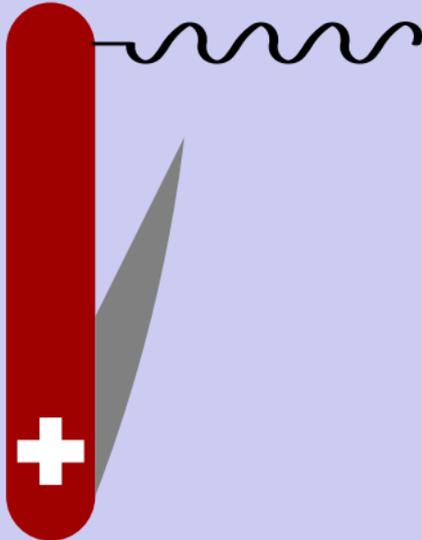


Die Option „with points 0“ gibt eine spezielle Punktform an, hier die kleinste, die es gibt.



vielfältig

Die vielen Möglichkeiten wurden bisher nur angedeutet. Mit Gnu-plot können nicht nur Anzahl der Stützstellen, Punkttyp und Achsen dargestellt werden.

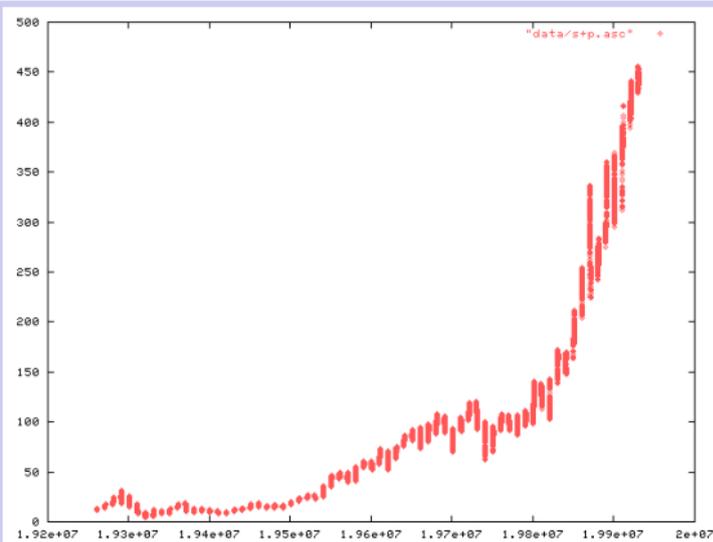


Zeitreihe mit Zeitachse

Beispiel: Der „Standard's and Poor 500 Index“ der Börse an der Wall Street.

Format der Datei: Jahr, Monat, Tag in einer Zahl, dann Kurs.
In den ersten zwei Jahren eine Angabe pro Woche, danach
(börsen)täglich.

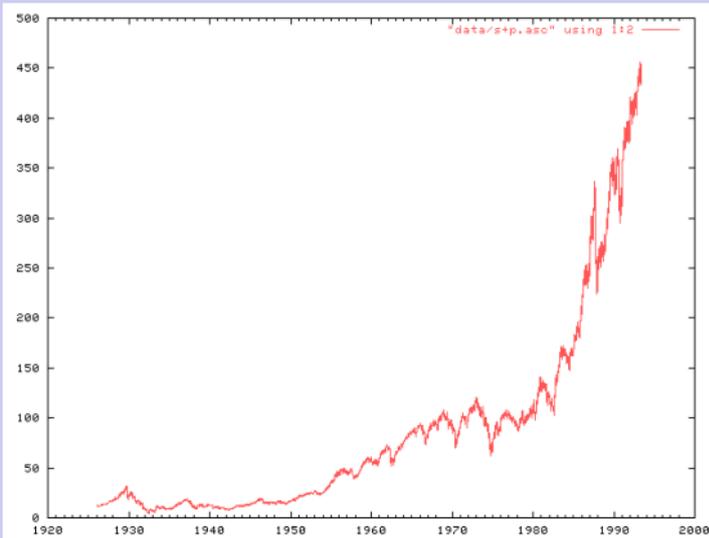
```
plot "data/s+p.asc"
```



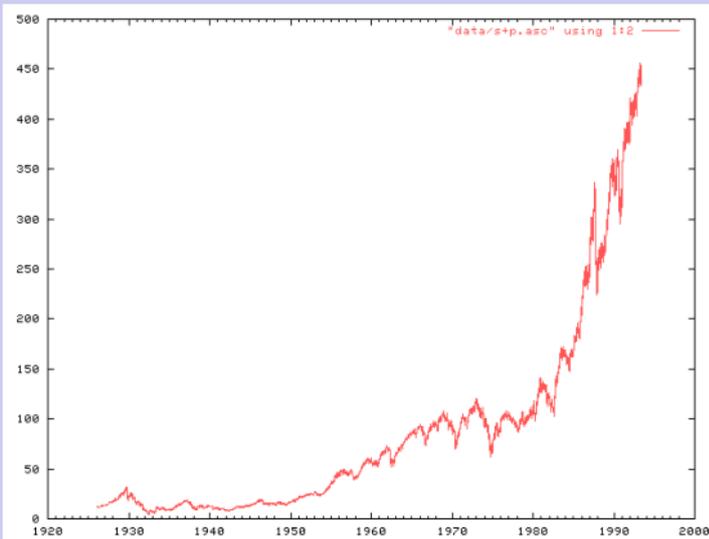
Datumsformat

Wie man gerade sah: Wenn Dateiinhalte geplottet werden, ist der Default Punkte. Bei Zeitreihen sind bisweilen Linien sinnvoller. Gnuplot akzeptiert auch Daten, man muss ihm nur sagen wie.

```
set timefmt "%Y%m%d"  
set xdata time  
set format x "%Y"
```



```
plot "data/s+p.asc" using 1:2 with lines
```

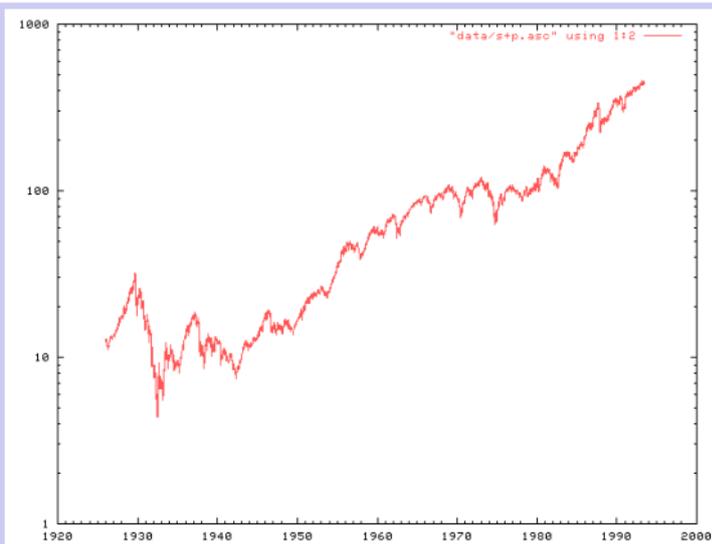


Änderung der Skalen

Einen realistischeren Eindruck von der Entwicklung des Marktes bekommt man, indem man die y -Achse logarithmiert.

```
set logscale y 10  
replot
```

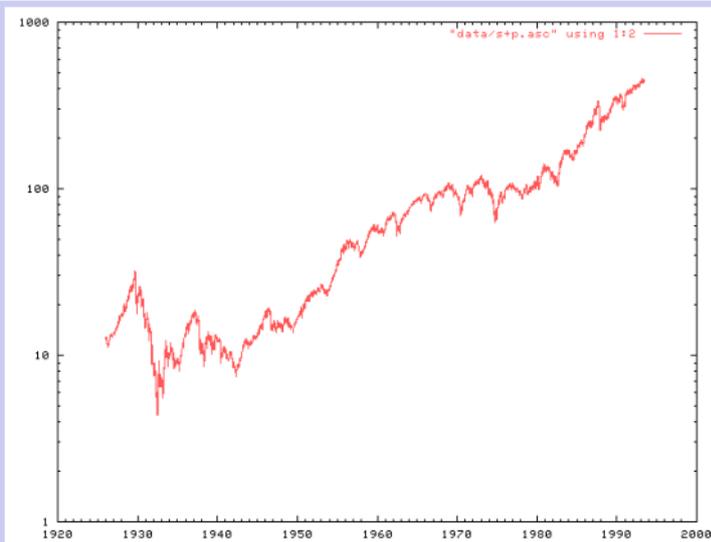
Logarithmieren kann man die Achse zur beliebigen Basis. Die Basis des natürlichen Logarithmus e ist allerdings keine feste Konstante.



Saisoneffekte?

„Sell in May and go away“ ist ein Börsianer-Spruch. Um zu sehen, was dran ist, kann man sich aus jedem Jahr verschiedene Börsentage ausgeben lassen. Jahre sind in den Originaldaten als ‚Blöcke‘ zusammengefasst, also durch Leerzeilen voneinander getrennt.

In Gnuplot kann man systematische Stichproben aus den Daten ziehen. Dies geht blockweise und zeilenweise, und zwar mit **every zi:bi:za:ba:ze:be**. Hierbei ist:



zi Zeileninkrement (,alle **zi** Zeilen in einem Block‘)

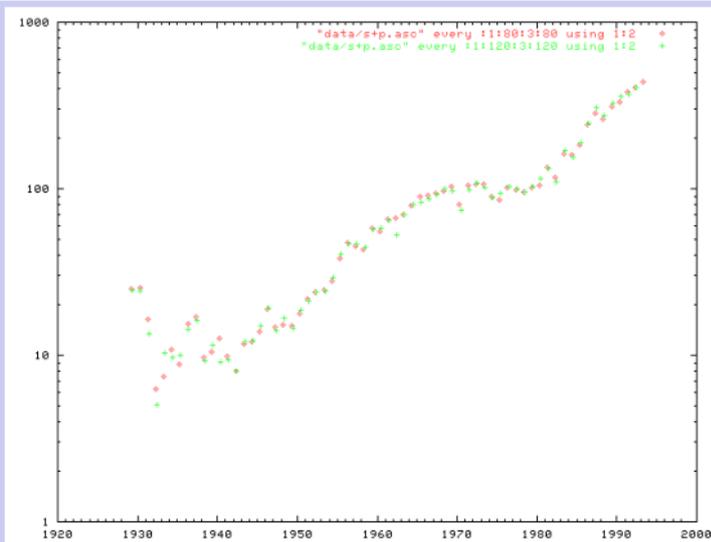
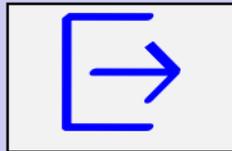
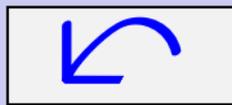
bi Blockinkrement (,alle **bi** Blöcke in Daten‘)

za erste Zeile (,starte mit **za**. Zeile in einem Block‘)

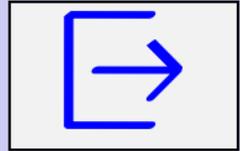
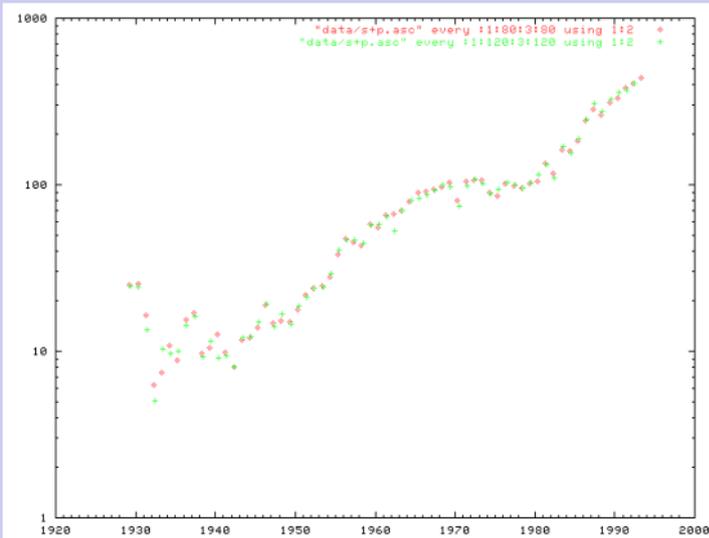
ba erster Block (,starte mit **ba**. Block in Daten‘)

ze letzte Zeile (,ende bei **ze**. Zeile in einem Block‘)

be letzter Block (,ende bei **ba**. Block in Daten‘)



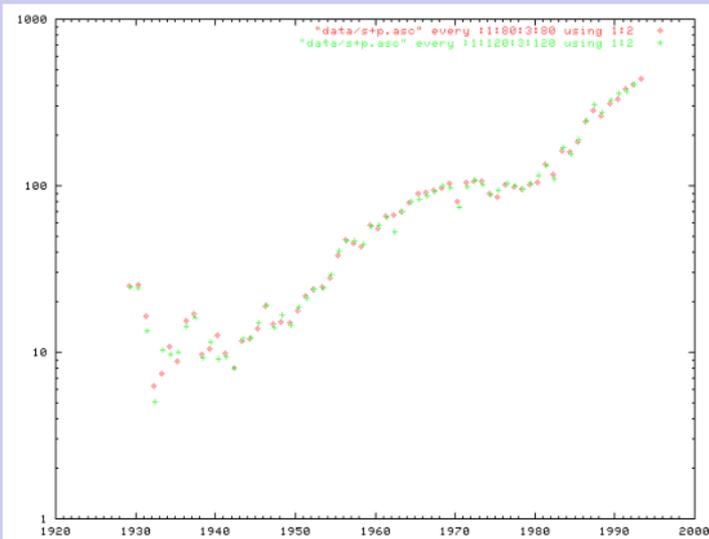
Dabei lassen sich allerdings nicht direkt aufeinanderfolgende Punkte nicht durch Linien miteinander verbinden. Man muss also die Option `with points` wählen.



Außer Punkten und Linien ...

hat man per `with` auch andere Darstellungsmöglichkeiten:

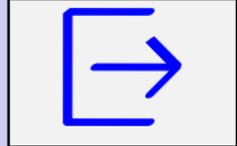
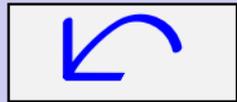
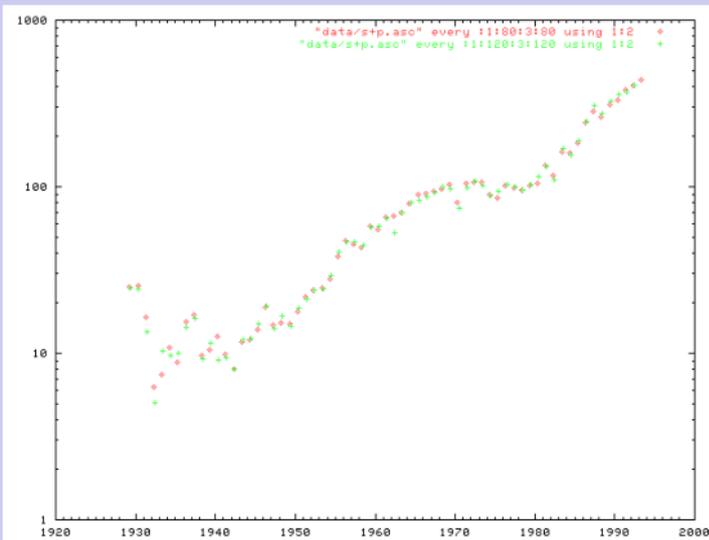
`linespoints`, `impulses`, `dots`, `steps`, `fsteps`, `histeps`,
`errorbars`, `xerrorbars`, `yerrorbars`, `xyerrorbars`, `boxes`,
`boxerrorbars`, `boxxyerrorbars`, `financebars`, `candlesticks`,
`vector`



Obacht bei Syntax

Die Reihenfolge unter den Erweiterungen wie **every**, **using**, **title**, **with** muss peinlichst genau eingehalten werden.

```
plot {<ranges>}  
  {<function> | {"<datafile>" {datafile-modifiers}}}  
  {axes <axes>} {<title-spec>} {with <style>}  
  {, {definitions,} <function> ...}
```



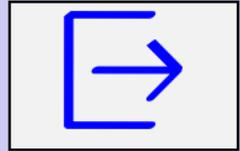
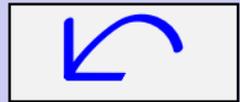
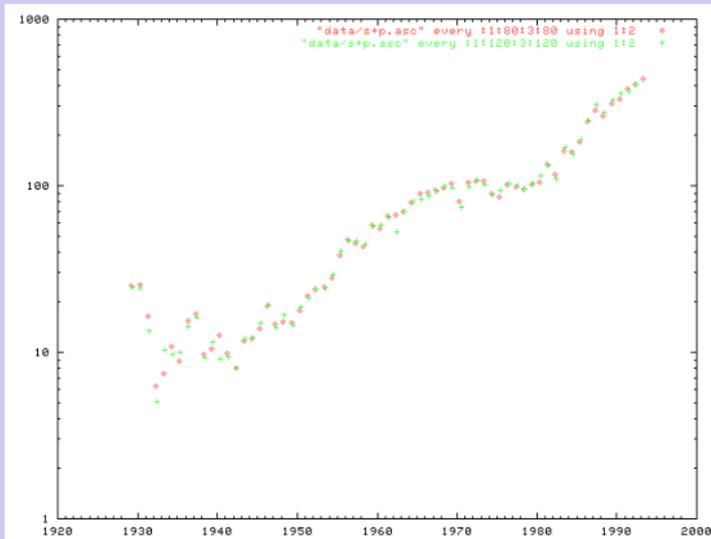
Sonst ist **set** dein Freund

Zum Beispiel:

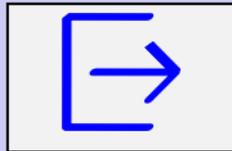
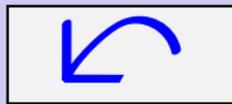
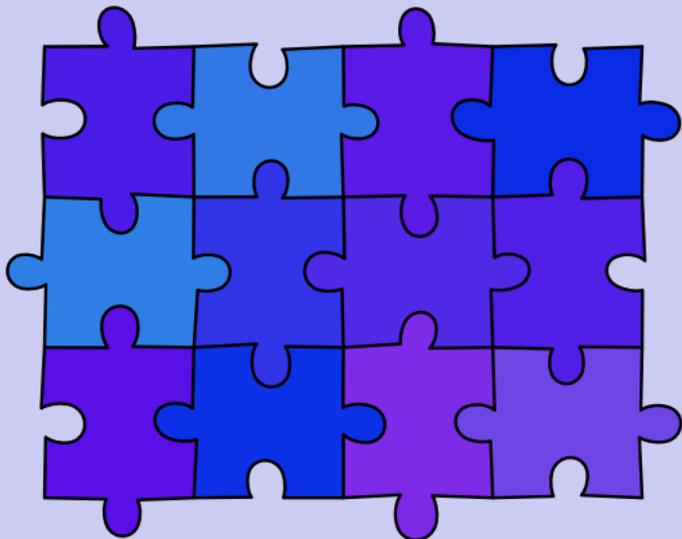
`set parametric` setzt parametrische Kurven (wie im Titel).

`set style` gibt Darstellungsform an (**with** nicht extra nötig).

`set nokey` schaltet Legende aus

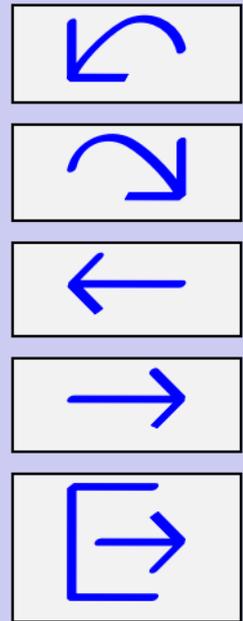
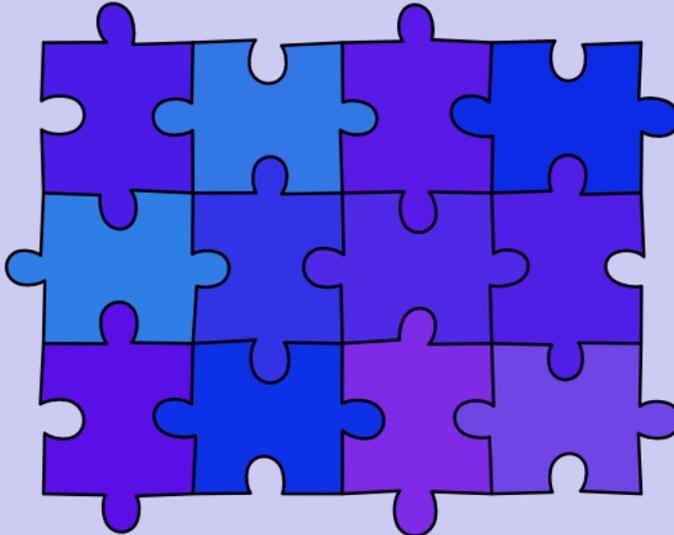


zugänglich



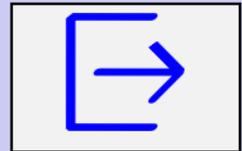
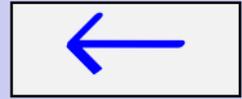
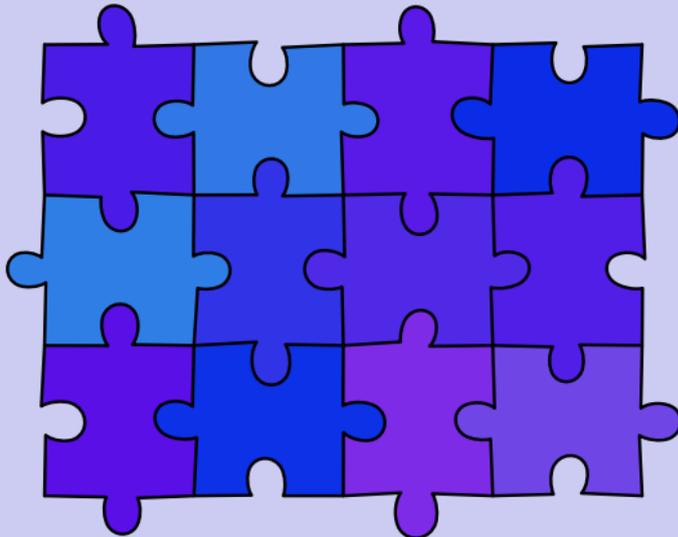
Ausgabegerät definieren

- `set terminal <Name des Gerätes>`
- einfach `set terminal` zeigt alle Möglichkeiten an
- manche Geräte mit spezifischen Optionen, zum Beispiel: `set terminal png gray` erzeugt Graustufen, `set terminal tgif "Helvetica"18` setzt Schriftart und `-grad`



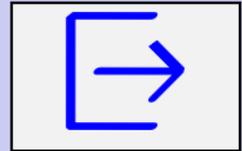
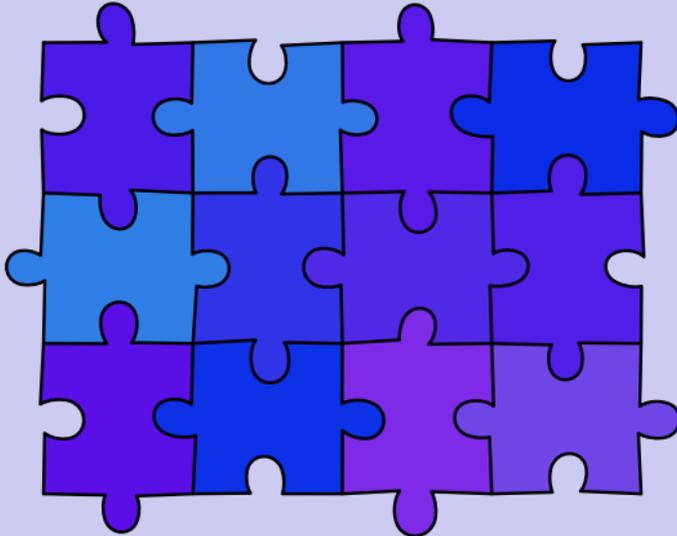
verschiedene Möglichkeiten

- Vektorgrafik (Metapost, CGM, Postscript, ...)
- Bitmapgrafik (PBM, PNG, ...)
- druckerspezifisch (HP, Imagen, Epson, ...)
- programmspezifisch (Autocad, Tgif, $\text{T}_{\text{E}}\text{X}$ (Pstricks, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ picture, eepic), Corel Draw, ...)



Ausgabe in Datei

Die Ausgabe soll natürlich in eine Datei umgeleitet werden. Dies geht mit `set output "dateiname"`.



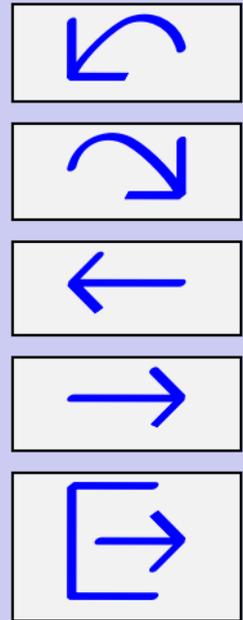
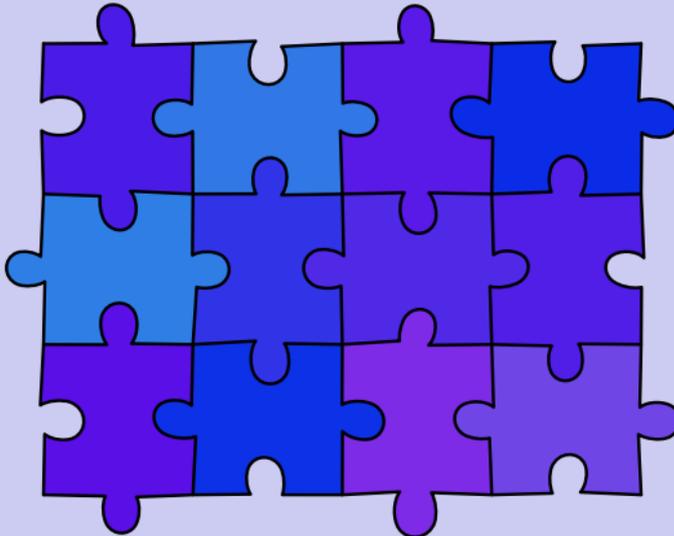
Betrieb im Batch-Modus

Gnuplot kann auch im Batch-Modus betrieben werden. Damit können zum Beispiel Serverstatistiken dargestellt und dynamisch in Webseiten eingebunden werden.

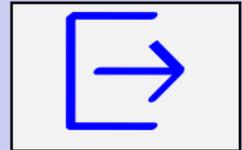
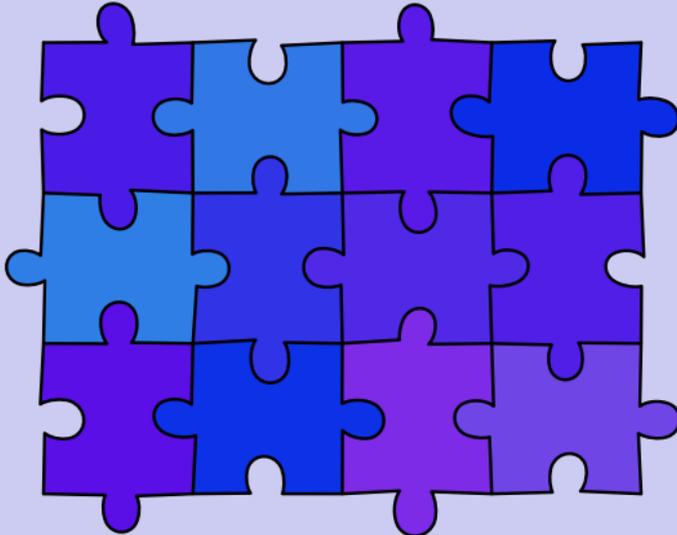
Durch die Eingabe von

```
/home/user> gnuplot "timeplot.gpl"
```

kann eine Grafik direkt erzeugt und gespeichert werden. In der Datei `timeplot.gpl` könnte zum Beispiel stehen:

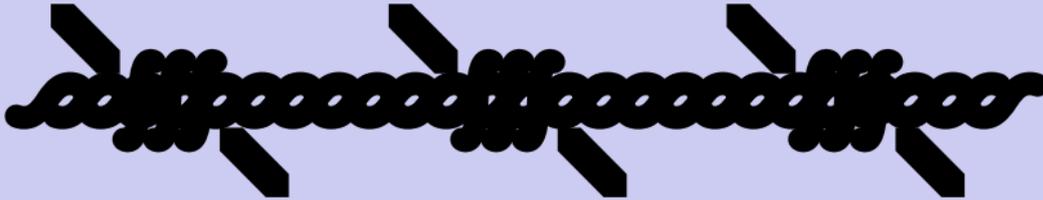


```
set terminal png color
set output "zugriffe.png"
set title "Wöchentliche Zugriffe seit Oktober 1998"
plot "zugriffe.dat" using 1:2 with lines
```



mit Grenzen

- Gnuplot sieht keine Flächenfüllung vor. Daher sind Balken- und Tortengrafiken, Histogramme u. ä. nicht ohne Weiteres zu bekommen.
- Gnuplot ist keine Programmiersprache, daher muss die Vorverarbeitung der Daten anderswo erfolgen.

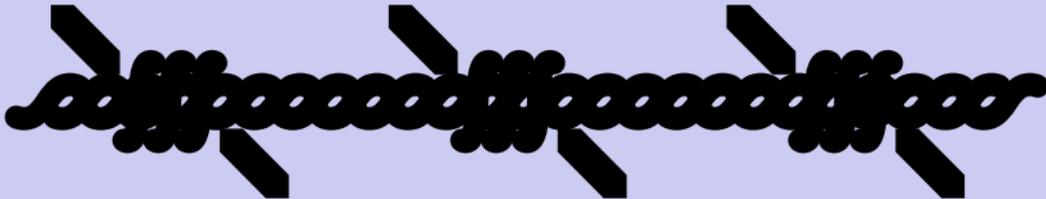


Lizenzfragen

„Gnuplot-FAQ 1.3: Does gnuplot have anything to do with the FSF and the GNU project?

Gnuplot is neither written nor maintained by the FSF. It is not covered by the General Public License, either. It used to be distributed by the FSF, however, due to licensing issues it is no longer.

Gnuplot is freeware in the sense that you don't have to pay for it. However it is not freeware in the sense that you would



be allowed to distribute a modified version of your gnuplot freely. Please read and accept the Copyright file in your distribution.“

Das Copyright erlaubt nicht die Weitergabe modifizierter Quellen, sondern nur von modifizierten Binärdateien mit Quell-Patches.

