

Einführung in gnuplot

Seminar Wissenschaftliches Arbeiten

Jannik Strötgen

Ruprecht-Karls-Universität Heidelberg
Institut für Informatik
Lehrstuhl für Datenbanksysteme
<http://dbs.ifi.uni-heidelberg.de>
stroetgen@uni-hd.de

25. November 2010

UNIVERSITÄT
HEIDELBERG



Motivation

The results of the TempEval-2 Challenge are published using precision (p), recall (r) and their weighted harmonic mean (f -score). The two runs of our system called HeidelTime achieved the best results with an F-score of 86% ($p = 82\%$, $r = 91\%$ and $p = 90\%$, $r = 82\%$, respectively). The f -score of the next best system is 85% ($p = 85\%$, $r = 85\%$). One of the other systems had a precision of 76% and a recall of 66%.

Overall, there were 15 runs of 8 different systems participating in the challenge and the results ...

Motivation

The results of the TempEval-2 Challenge are published using precision (p), recall (r) and their weighted harmonic mean (f -score). The two runs of our system cal-

Gute Darstellung der Ergebnisse?

Overall, there were 15 runs of 8 different systems participating in the challenge and the results ...

Motivation

The results of the TempEval-2 Challenge are published using precision (p), recall (r) and their weighted harmonic mean (f -score). The two runs of our system cal-

Gute Darstellung der Ergebnisse?

- unübersichtlich
- Ergebnisse der meisten Systeme fehlen komplett
- ...

Overall, there were 15 runs of 8 different systems participating in the challenge and the results ...

Motivation

The results of the TempEval-2 Challenge are published using precision (p), recall (r) and their weighted harmonic mean (f -score). The two runs of our system call

Gute Darstellung der Ergebnisse?

- unübersichtlich
- Ergebnisse der meisten Systeme fehlen komplett
- ...

Overall, there were 15 runs of 8 different systems participating in the challenge and the results ...

Bessere Darstellung der Ergebnisse:

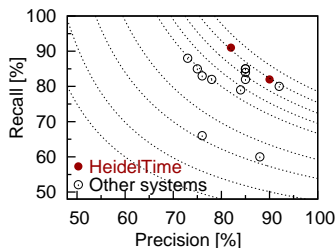


Abbildung: Results of systems participating in the TempEval-2 challenge. F-score contours for reference.

Motivation

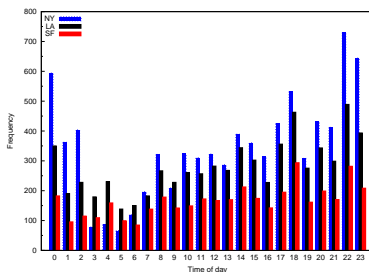
Twitterdaten

Stunde	NY	LA	SF
0	593.8	350.0	182.6
1	362.0	190.67	95.67
2	403.25	228.0	115.0
3	77.86	179.43	109.57
4	86.56	230.33	159.22
5	65.92	138.46	99.54
6	119.38	150.23	85.0
7	196.18	182.73	138.36
8	321.6	266.7	178.6
9	208.69	228.0	142.13
10	324.17	60.92	149.17
11	309.92	256.75	172.58
12	322.46	282.46	167.0
13	286.54	268.23	169.92
14	389.0	344.2	212.6
15	359.82	302.45	174.27
16	314.31	227.62	142.54
17	424.7	356.1	195.2
18	532.38	463.0	293.88
19	308.14	275.57	161.64
20	431.9	343.4	198.9
21	412.09	299.0	170.36
22	730.0	489.17	281.83
23	644.0	393.5	208.5

Motivation

Twitterdaten

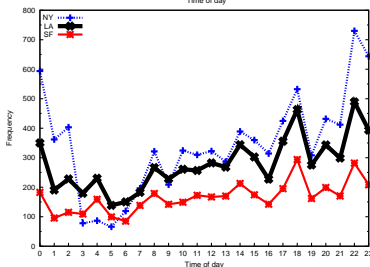
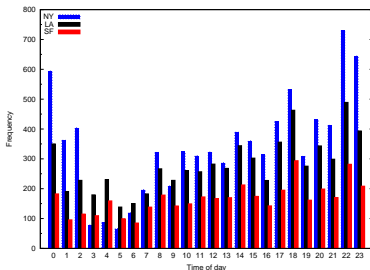
Stunde	NY	LA	SF
0	593.8	350.0	182.6
1	362.0	190.67	95.67
2	403.25	228.0	115.0
3	77.86	179.43	109.57
4	86.56	230.33	159.22
5	65.92	138.46	99.54
6	119.38	150.23	85.0
7	196.18	182.73	138.36
8	321.6	266.7	178.6
9	208.69	228.0	142.13
10	324.17	60.92	149.17
11	309.92	256.75	172.58
12	322.46	282.46	167.0
13	286.54	268.23	169.92
14	389.0	344.2	212.6
15	359.82	302.45	174.27
16	314.31	227.62	142.54
17	424.7	356.1	195.2
18	532.38	463.0	293.88
19	308.14	275.57	161.64
20	431.9	343.4	198.9
21	412.09	299.0	170.36
22	730.0	489.17	281.83
23	644.0	393.5	208.5



Motivation

Twitterdaten

Stunde	NY	LA	SF
0	593.8	350.0	182.6
1	362.0	190.67	95.67
2	403.25	228.0	115.0
3	77.86	179.43	109.57
4	86.56	230.33	159.22
5	65.92	138.46	99.54
6	119.38	150.23	85.0
7	196.18	182.73	138.36
8	321.6	266.7	178.6
9	208.69	228.0	142.13
10	324.17	60.92	149.17
11	309.92	256.75	172.58
12	322.46	282.46	167.0
13	286.54	268.23	169.92
14	389.0	344.2	212.6
15	359.82	302.45	174.27
16	314.31	227.62	142.54
17	424.7	356.1	195.2
18	532.38	463.0	293.88
19	308.14	275.57	161.64
20	431.9	343.4	198.9
21	412.09	299.0	170.36
22	730.0	489.17	281.83
23	644.0	393.5	208.5



Motivation

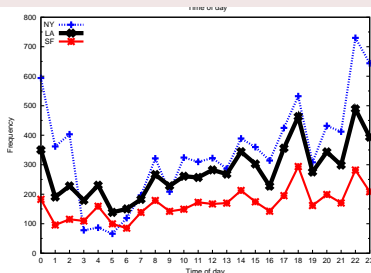
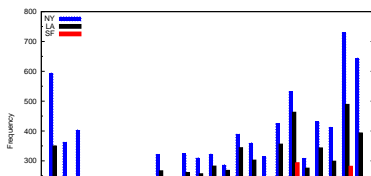
Twitterdaten

Stunde	NY	LA	SF
0	593.8	350.0	182.6
1	362.0	190.67	95.67
2	403.25	228.0	115.0
3	77.86	179.43	109.57
4	86.56	230.33	159.22
5	65.92	138.46	99.54

Sehr häufig gilt:

Grafik "leserlicher" als Tabelle

6	309.92	256.75	172.58
7	322.46	282.46	167.0
8	286.54	268.23	169.92
9	389.0	344.2	212.6
10	359.82	302.45	174.27
11	314.31	227.62	142.54
12	424.7	356.1	195.2
13	532.38	463.0	293.88
14	308.14	275.57	161.64
15	431.9	343.4	198.9
16	412.09	299.0	170.36
17	730.0	489.17	281.83
18	644.0	393.5	208.5



Motivation

Nach dem Motto

“Ein Bild sagt mehr als 1000 Worte”

Grafiken immer dann verwenden, wenn sie sinnvoll sind.

Das heißt, im Text folgt:

- kurze Beschreibung der Ergebnisse
- Querverweis auf die Grafik
- Analyse der Ergebnisse

Outline

- 1** Einführung
 - Plot, plotten, ...
- 2** Fragen
 - Was kann visualisiert werden?
 - Wie können Plots aussehen?
 - Woher kommen die Daten?
 - Wie geht man vor?
- 3** Eigenschaften
- 4** Interaktives Beispiel
- 5** Beispiel mit Messdaten

Outline

1 Einführung

- Plot, plotten, ...

2 Fragen

- Was kann visualisiert werden?
- Wie können Plots aussehen?
- Woher kommen die Daten?
- Wie geht man vor?

3 Eigenschaften

4 Interaktives Beispiel

5 Beispiel mit Messdaten



Einführung - Was ist gnuplot?

Plot, plotten, ...



Einführung - Was ist gnuplot?

Plot, plotten, ...

to plot¹

einen Anschlag ausüben

¹(siehe: <http://dict.leo.org/>)



Einführung - Was ist gnuplot?

Plot, plotten, ...

to plot¹

einen Anschlag ausüben

to plot (math.)¹

graphisch darstellen

¹(siehe: <http://dict.leo.org/>)



Einführung - Was ist gnuplot?

Plot, plotten, ...

to plot¹

einen Anschlag ausüben

to plot (math.)¹

graphisch darstellen

Plot (Definition)

Ein Plot ist die graphische Darstellung von Messdaten oder eines mathematischen Zusammenhangs.

¹(siehe: <http://dict.leo.org/>)



Einführung - Was ist gnuplot?

gnuplot ist

- ein (*das*) Programm zum Plotten
- frei verfügbar
- verfügbar als Sourcecode und als vorkompilierte Programme für alle möglichen Betriebssysteme und Rechnerarchitekturen



Einführung - Was ist gnuplot?

gnuplot ist

- ein (*das*) Programm zum Plotten
- frei verfügbar
- verfügbar als Sourcecode und als vorkompilierte Programme für alle möglichen Betriebssysteme und Rechnerarchitekturen

Eigenschaften:

- kommandozeilenorientiert
- skriptbasiert
- interaktiv

Outline

- 1 Einführung
 - Plot, plotten, ...
- 2 Fragen
 - Was kann visualisiert werden?
 - Wie können Plots aussehen?
 - Woher kommen die Daten?
 - Wie geht man vor?
- 3 Eigenschaften
- 4 Interaktives Beispiel
- 5 Beispiel mit Messdaten

Was kann visualisiert werden?

gnuplot visualisiert:

- Funktionen/Daten in 2D oder 3D

Was kann visualisiert werden?

gnuplot visualisiert:

- Funktionen/Daten in 2D oder 3D

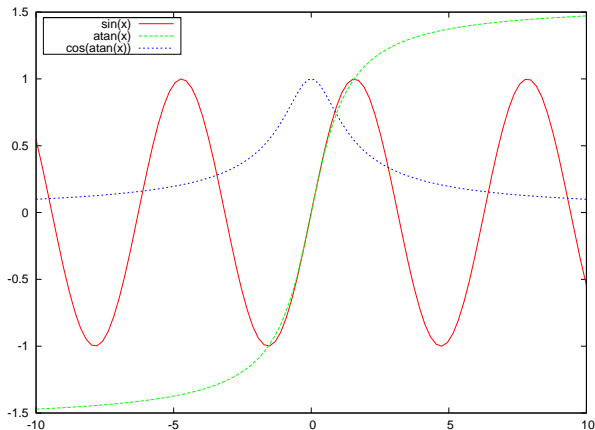
2D Daten beispielsweise als

- Punkte
- Linien
- Linien mit Punkten
- Balken

Wie können Plots aussehen?¹

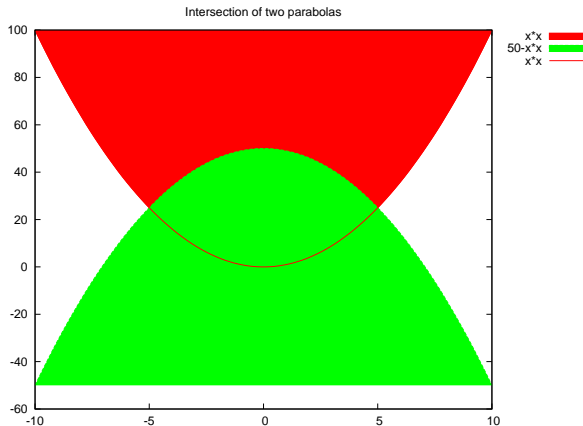
¹Beispiele von <http://gnuplot.sourceforge.net/demo/>

Wie können Plots aussehen?¹



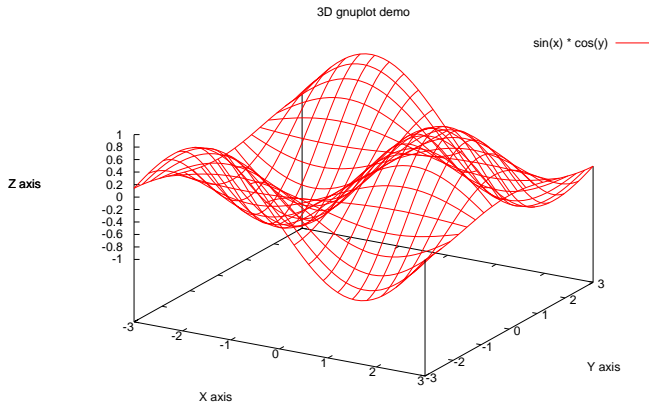
¹Beispiele von <http://gnuplot.sourceforge.net/demo/>

Wie können Plots aussehen?¹



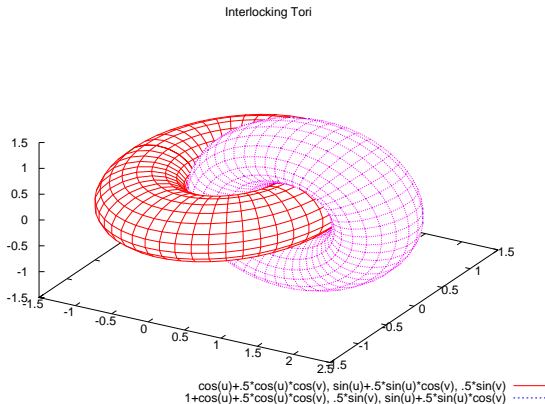
¹Beispiele von <http://gnuplot.sourceforge.net/demo/>

Wie können Plots aussehen?¹



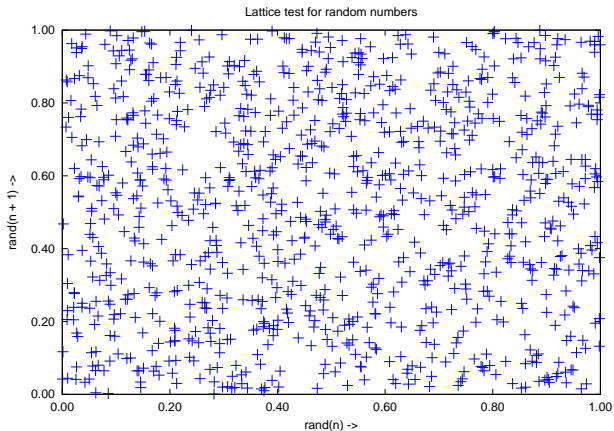
¹Beispiele von <http://gnuplot.sourceforge.net/demo/>

Wie können Plots aussehen?¹



¹Beispiele von <http://gnuplot.sourceforge.net/demo/>

Wie können Plots aussehen?¹



¹Beispiele von <http://gnuplot.sourceforge.net/demo/>

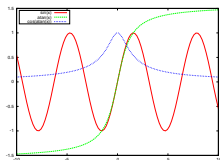
Wie können Plots aussehen?¹



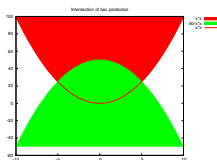
¹Beispiele von <http://gnuplot.sourceforge.net/demo/>

Wie können Plots aussehen?¹

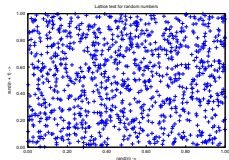
lines



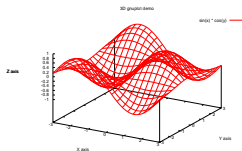
filledcurves



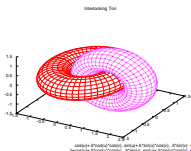
random



3D



3D



smily



¹Beispiele von <http://gnuplot.sourceforge.net/demo/>

Woher kommen die Daten?

- Funktionen: $\sin(x)$, $\text{abs}(x)$, ...
- Einlesen von Daten aus externen Files

Wie geht man vor?

Normalerweise . . .

- man weiß, was man darstellen möchte

Wie geht man vor?

Normalerweise . . .

- man weiß, was man darstellen möchte
- man beginnt interaktiv und passt alle Einstellungen an

Wie geht man vor?

Normalerweise . . .

- man weiß, was man darstellen möchte
- man beginnt interaktiv und passt alle Einstellungen an
- entspricht der Plot den Vorstellungen exportiert man ihn (z.B. als Postscript) und speichert alle Einstellungen als Skript

Outline

- 1 Einführung
 - Plot, plotten, ...
- 2 Fragen
 - Was kann visualisiert werden?
 - Wie können Plots aussehen?
 - Woher kommen die Daten?
 - Wie geht man vor?
- 3 Eigenschaften**
- 4 Interaktives Beispiel
- 5 Beispiel mit Messdaten

Einführung - Eigenschaften

Die wichtigsten Befehle:

- help
- set
- plot (splot)

Einführung - Eigenschaften

Die wichtigsten Befehle:

- help
- set
- plot (splot)
- unset
- reset

Wichtig:

- Befehle kennenlernen
- Terminologie kennenlernen
- *interaktiv* ausprobieren

Einführung - Eigenschaften

Es gibt Standardeinstellungen für alles.

Beispiel:

- grafische Ausgabe: X Server

Dadurch

- kann man sofort beginnen

Einführung - Eigenschaften

Es gibt Standardeinstellungen für alles.

Beispiel:

- grafische Ausgabe: X Server

Dadurch

- kann man sofort beginnen

Aber, das wirklich Tolle:

- man kann (fast) alles verändern

Outline

- 1 Einführung
 - Plot, plotten, ...
- 2 Fragen
 - Was kann visualisiert werden?
 - Wie können Plots aussehen?
 - Woher kommen die Daten?
 - Wie geht man vor?
- 3 Eigenschaften
- 4 **Interaktives Beispiel**
- 5 Beispiel mit Messdaten

Interaktives Beispiel I

gnuplot starten

- Terminal öffnen
- *gnuplot* eingeben
- Kommandofenster mit Prompt:
`gnuplot>`

Interaktives Beispiel I

gnuplot starten

- Terminal öffnen
- *gnuplot* eingeben
- Kommandofenster mit Prompt:

```
gnuplot>
```

gnuplot beenden

```
gnuplot> quit
```

Interaktives Beispiel I

gnuplot starten

- Terminal öffnen
- `gnuplot` eingeben
- Kommandofenster mit Prompt:

```
gnuplot>
```

Los geht's

```
gnuplot> f(x) = x**2
```

```
gnuplot> plot f(x)
```

gnuplot beenden

```
gnuplot> quit
```

Interaktives Beispiel I

gnuplot starten

- Terminal öffnen
- *gnuplot* eingeben
- Kommandofenster mit Prompt:

```
gnuplot>
```

Los geht's

```
gnuplot> f(x) = x**2
```

```
gnuplot> plot f(x)
```

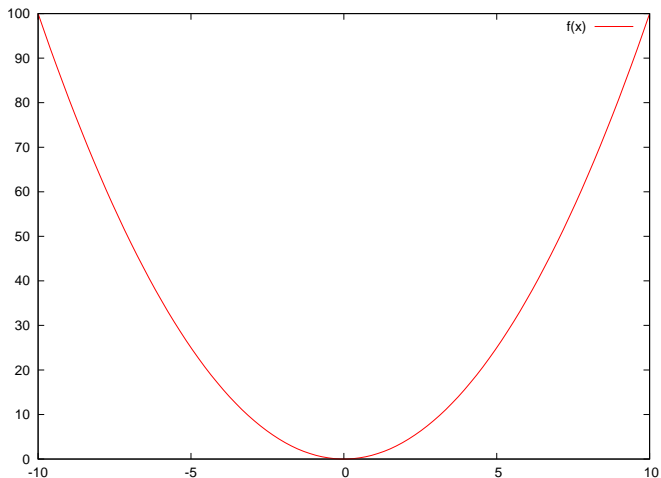
gnuplot beenden

```
gnuplot> quit
```

Alternativ:

```
gnuplot> plot x**2
```

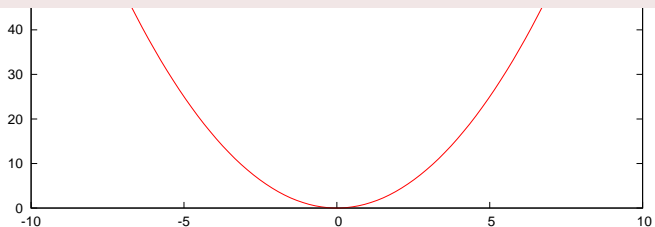
Interaktives Beispiel I



Interaktives Beispiel I



Zufrieden?

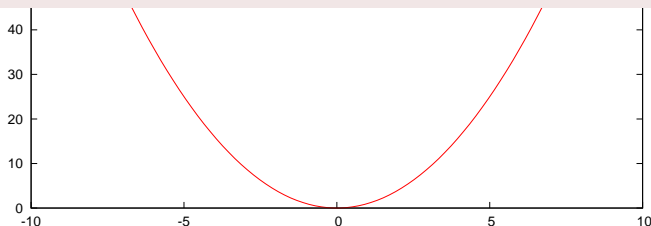


Interaktives Beispiel I

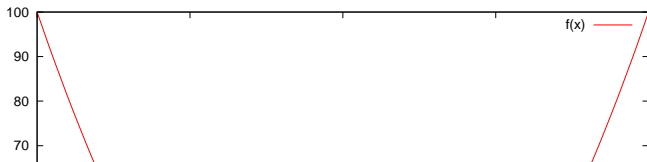


Zufrieden?

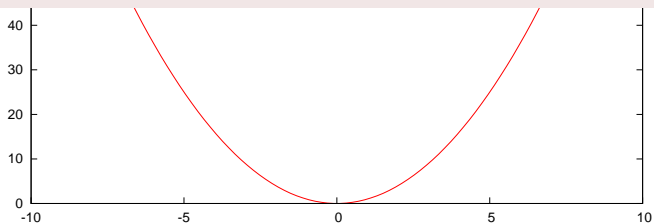
Meistens nicht auf Anhieb, aber (fast) alles ist veränderbar!



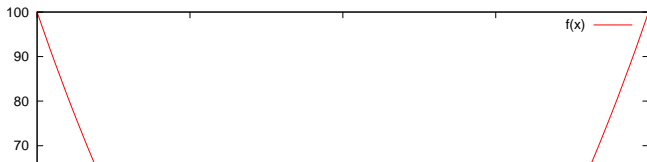
Interaktives Beispiel I



Wir wollen beispielsweise ...

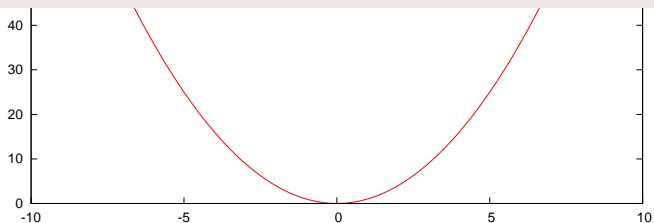


Interaktives Beispiel I



Wir wollen beispielsweise ...

andere Achsenskalierung



Interaktives Beispiel I

Befehle: *xrange*, *yrange*

Details: *help xrange*

Interaktives Beispiel I

Befehle: *xrange*, *yrange*

Details: *help xrange*

```
gnuplot> set xrange[-7.5:7.5]
```

```
gnuplot> set yrange[-25:50]
```

```
gnuplot> plot f(x)
```

Interaktives Beispiel I

Befehle: *xrange*, *yrange*

Details: *help xrange*

```
gnuplot> set xrange[-7.5:7.5]
```

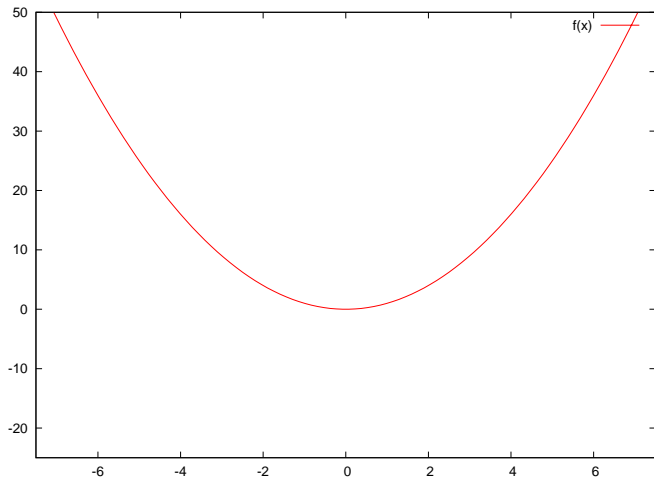
```
gnuplot> set yrange[-25:50]
```

```
gnuplot> plot f(x)
```

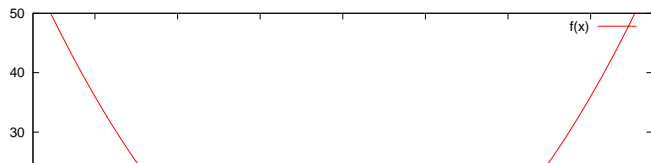
Alternativ:

```
gnuplot> replot
```

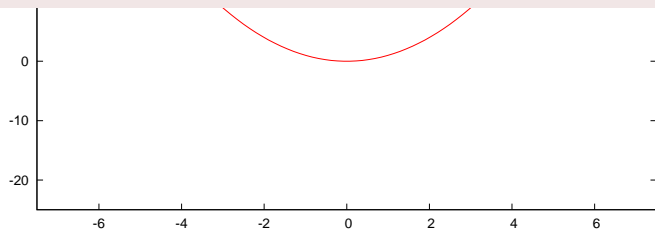
Interaktives Beispiel I



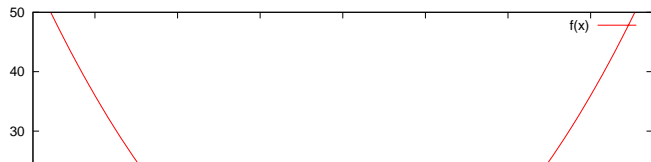
Interaktives Beispiel I



Wir wollen beispielsweise ...

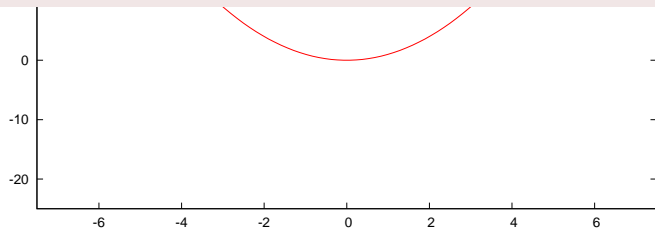


Interaktives Beispiel I



Wir wollen beispielsweise ...

andere Farbe und dickere Linie



Interaktives Beispiel I

Die Plots können verschiedene Farben, Breiten und Muster haben. Farben können auch explizit angegeben werden.

Dafür gibt es die Befehle:

- `linetype (lt)`
- `linecolor (lc)`
- `linewidth (lw)`

Interaktives Beispiel I

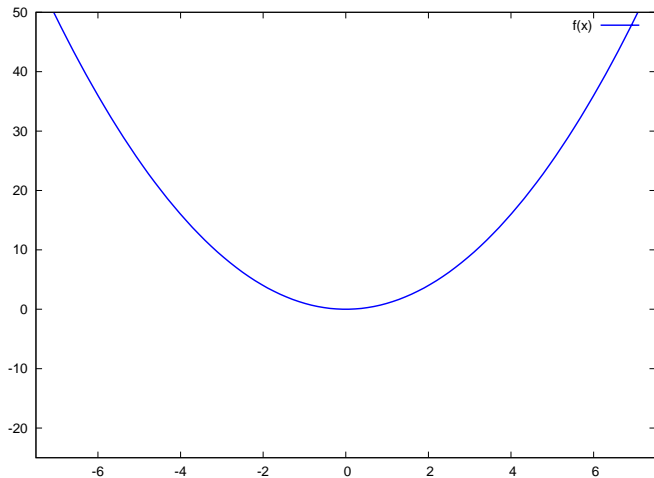
Die Plots können verschiedene Farben, Breiten und Muster haben. Farben können auch explizit angegeben werden.

Dafür gibt es die Befehle:

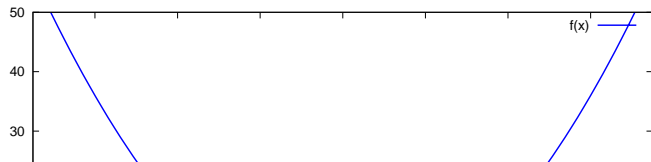
- `linetype (lt)`
- `linecolor (lc)`
- `linewidth (lw)`

```
gnuplot> plot f(x) lt 1 lc rgb 'blue' lw 3
```

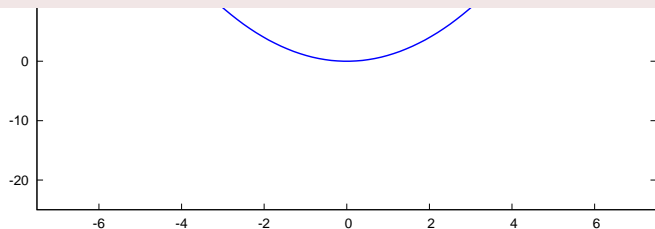
Interaktives Beispiel I



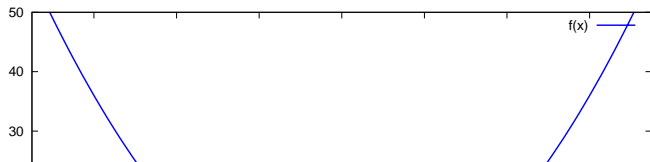
Interaktives Beispiel I



Wir wollen beispielsweise ...

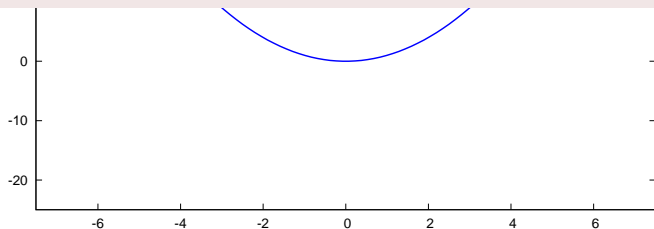


Interaktives Beispiel I



Wir wollen beispielsweise ...

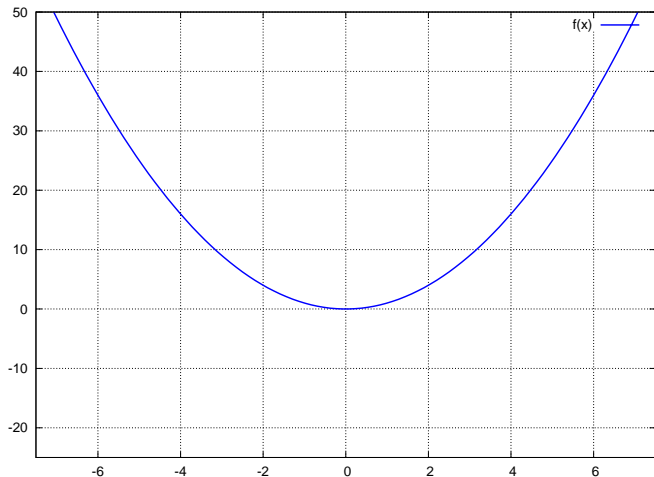
ein Grid



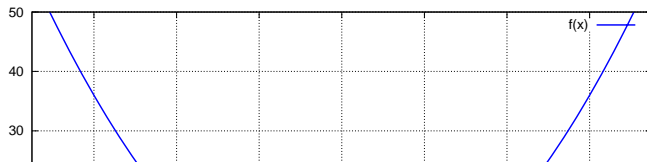
Interaktives Beispiel I

```
gnuplot> set grid  
gnuplot> replot
```

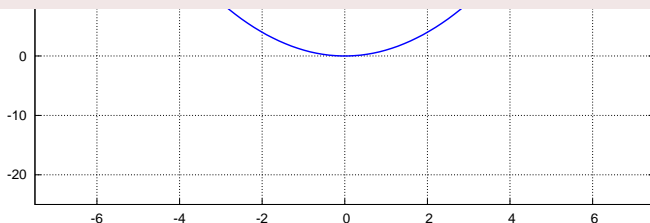
Interaktives Beispiel I



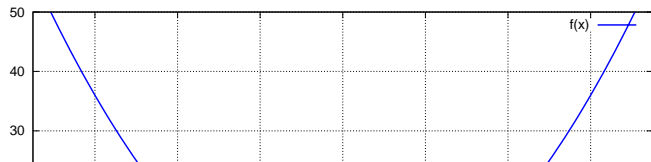
Interaktives Beispiel I



Wir wollen beispielsweise ...

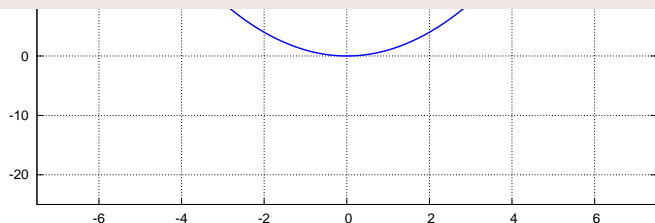


Interaktives Beispiel I



Wir wollen beispielsweise ...

“richtige” x- und y-Achsen



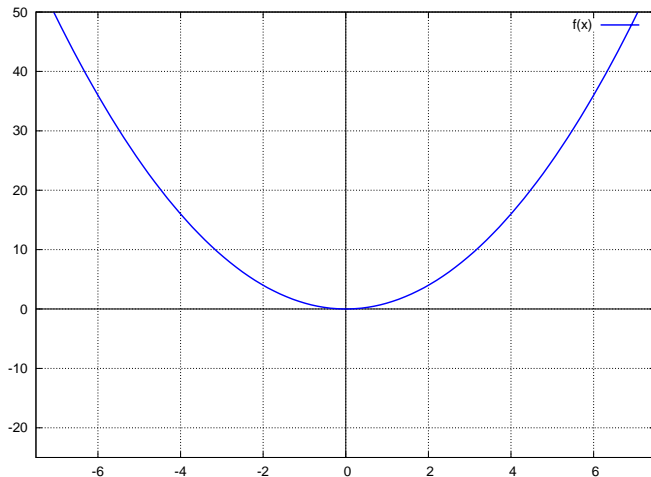
Interaktives Beispiel I

```
gnuplot> set xzeroaxis lt -1
```

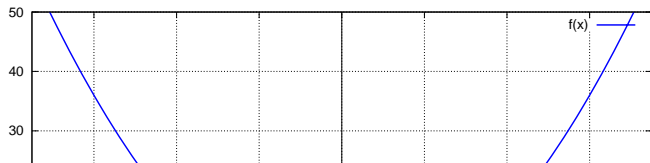
```
gnuplot> set yzeroaxis lt -1
```

```
gnuplot> replot
```

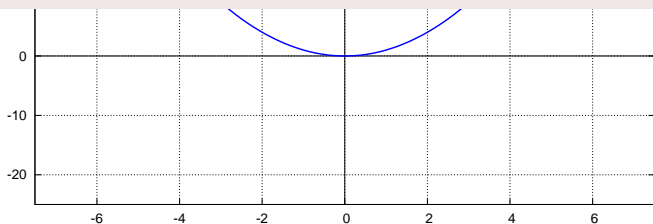
Interaktives Beispiel I



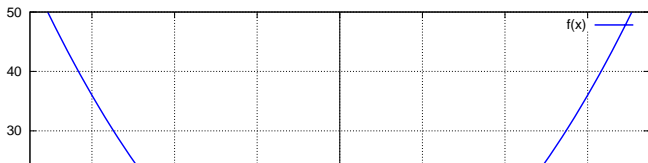
Interaktives Beispiel I



Wir wollen beispielsweise ...

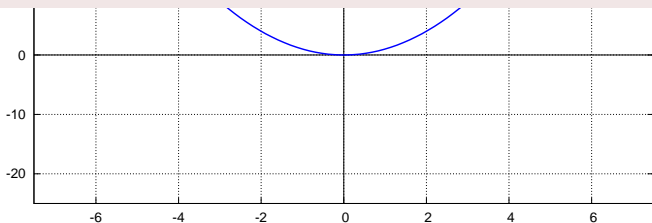


Interaktives Beispiel I



Wir wollen beispielsweise ...

Legende an einer anderen Stelle



Interaktives Beispiel I

Die Legende (*key*) kann innerhalb, außerhalb, oben, unten, rechts, links, mittig, ... plaziert werden.

Höhe, Breite, Beschriftung, Rahmen können ebenfalls geändert werden.

Interaktives Beispiel I

Die Legende (*key*) kann innerhalb, außerhalb, oben, unten, rechts, links, mittig, ... plaziert werden.

Höhe, Breite, Beschriftung, Rahmen können ebenfalls geändert werden.

Legende auszublenden: *unset key*

Details: *help key*

Interaktives Beispiel I

Die Legende (*key*) kann innerhalb, außerhalb, oben, unten, rechts, links, mittig, ... plaziert werden.

Höhe, Breite, Beschriftung, Rahmen können ebenfalls geändert werden.

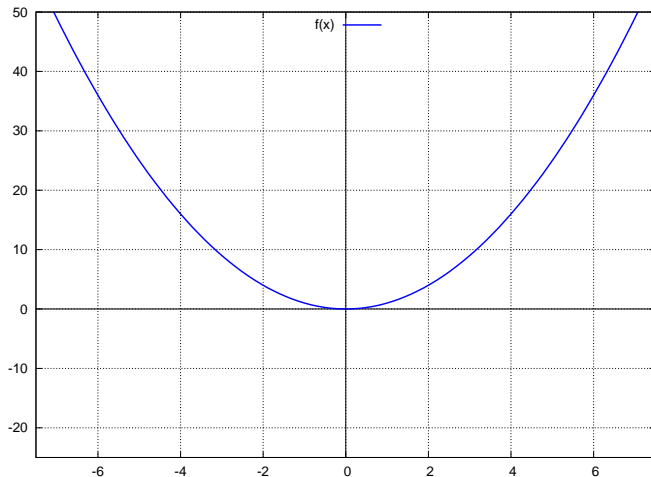
Legende auszublenden: *unset key*

Details: *help key*

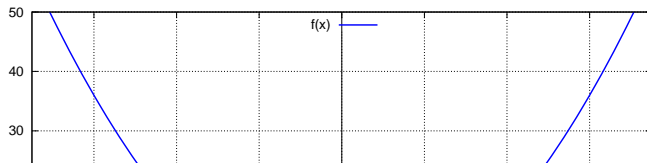
```
gnuplot> set key inside center top
```

```
gnuplot> replot
```

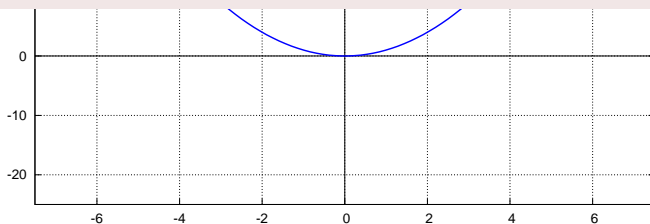
Interaktives Beispiel I



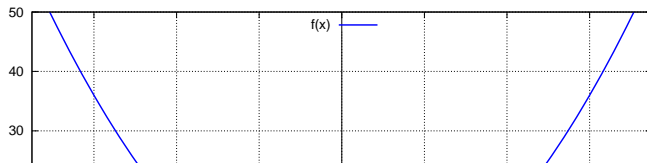
Interaktives Beispiel I



Wir wollen beispielsweise ...

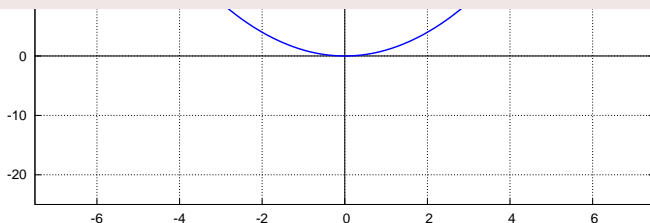


Interaktives Beispiel I



Wir wollen beispielsweise ...

Titel- und Achsenbeschriftung



Interaktives Beispiel I

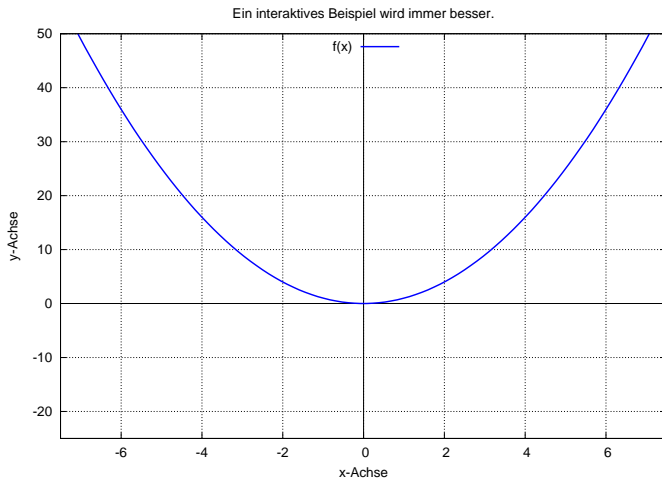
```
gnuplot> set title 'Ein interaktives Beispiel wird immer besser.'
```

```
gnuplot> set xlabel 'x-Achse'
```

```
gnuplot> set ylabel 'y-Achse'
```

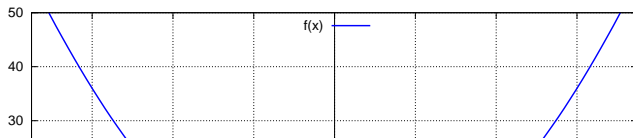
```
gnuplot> replot
```

Interaktives Beispiel I

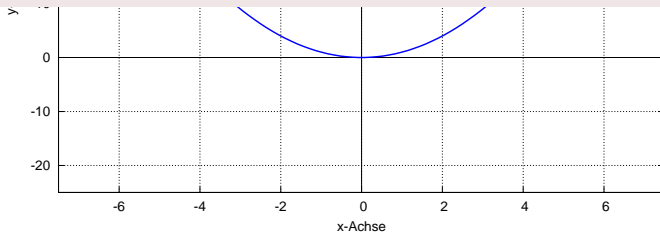


Interaktives Beispiel I

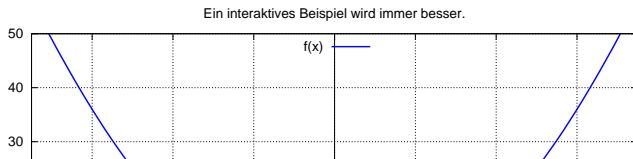
Ein interaktives Beispiel wird immer besser.



Wir wollen beispielsweise ...

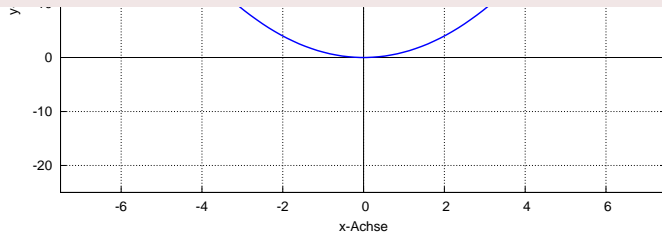


Interaktives Beispiel I



Wir wollen beispielsweise ...

Oben und rechts Achsen ohne Striche.



Interaktives Beispiel I

Die Striche an den Achsen heißen *tics*. Es gibt große und kleine *tics*, für die x-Achse also *xtics* und *mxtics* (minor x tics).

Details: *help xtics*, *help mxtics*

Interaktives Beispiel I

Die Striche an den Achsen heißen *tics*. Es gibt große und kleine tics, für die x-Achse also *xtics* und *mxtics* (minor x tics).

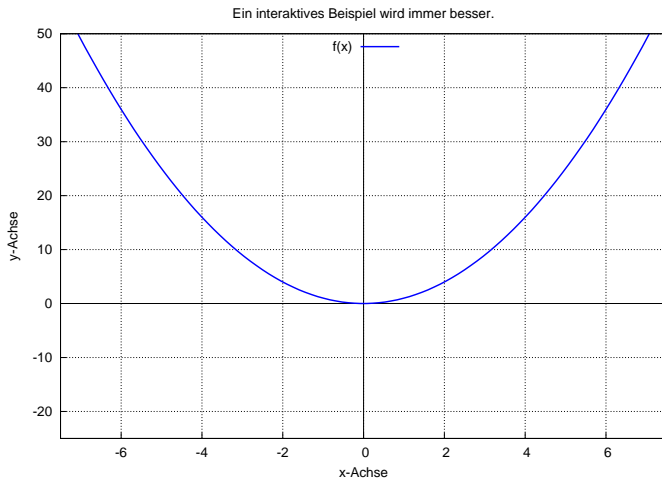
Details: *help xtics*, *help mxtics*

```
gnuplot> set xtics nomirror
```

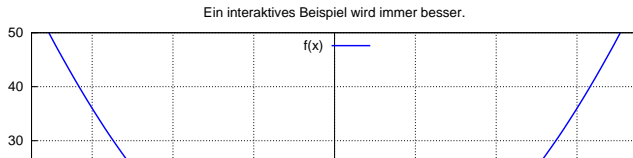
```
gnuplot> set ytics nomirror
```

```
gnuplot> replot
```

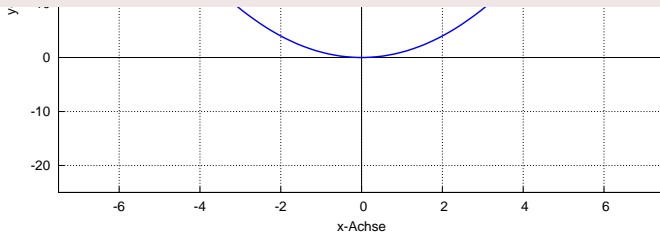
Interaktives Beispiel I



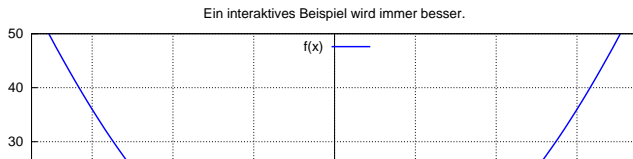
Interaktives Beispiel I



Wir wollen beispielsweise

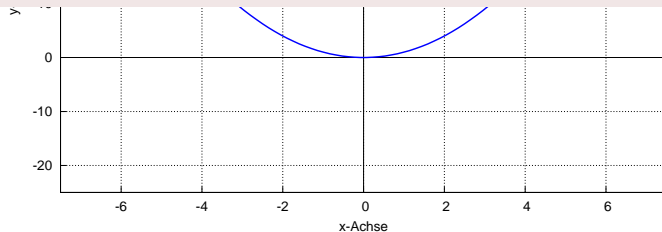


Interaktives Beispiel I



Wir wollen beispielsweise

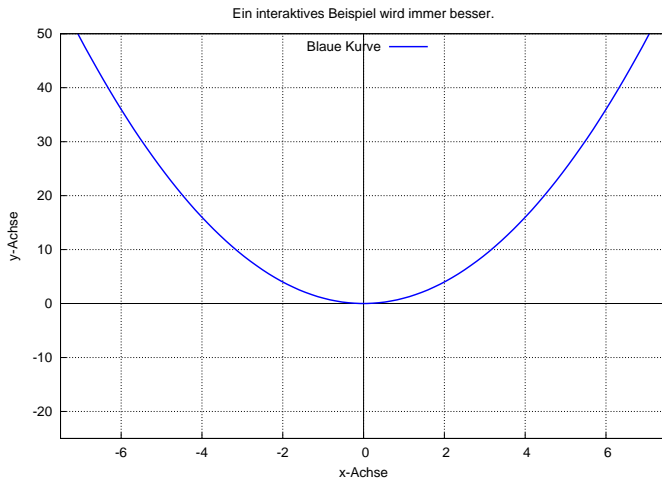
die Kurve umbenennen



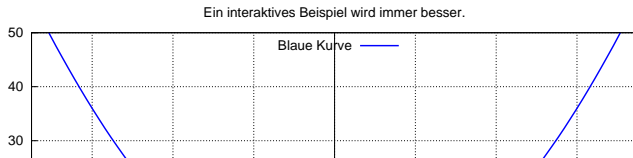
Interaktives Beispiel I

```
gnuplot> plot f(x) lt 1 lc rgb 'blue' lw 3 title 'Blaue Kurve'
```

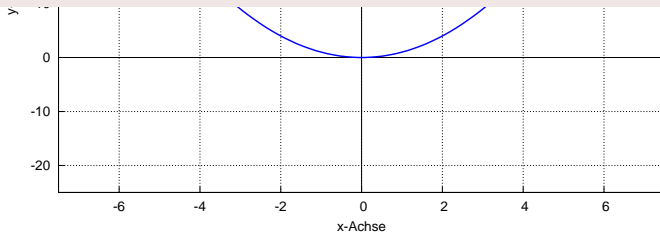
Interaktives Beispiel I



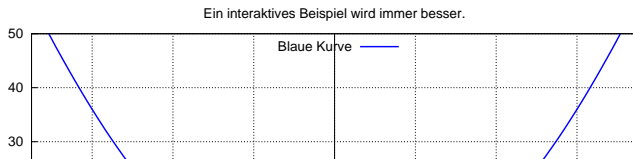
Interaktives Beispiel I



Wir wollen beispielsweise ...

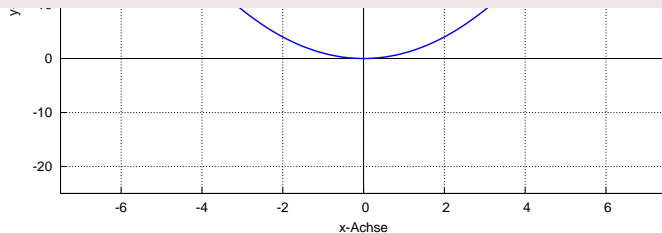


Interaktives Beispiel I



Wir wollen beispielsweise ...

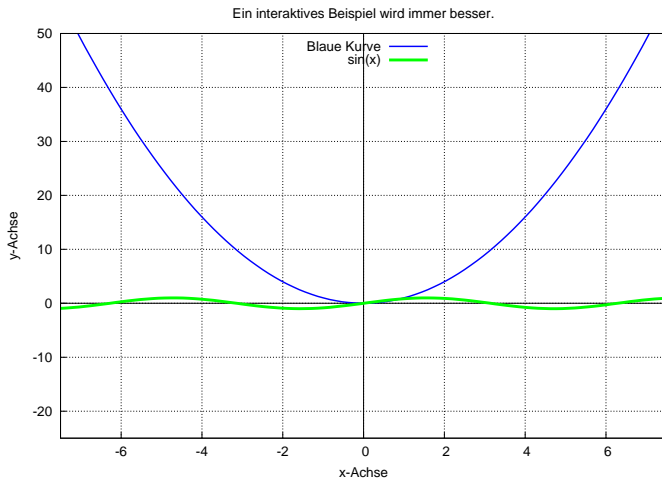
eine zweite Kurve



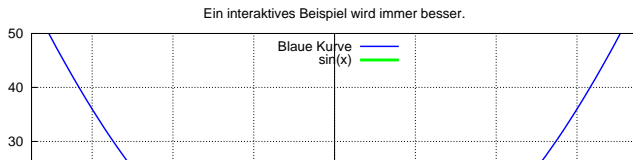
Interaktives Beispiel I

```
gnuplot> plot f(x) lt 1 lc rgb 'blue' lw 3 title 'Blaue Kurve',  
sin(x) lt 1 lc rgb 'green' lw 6
```

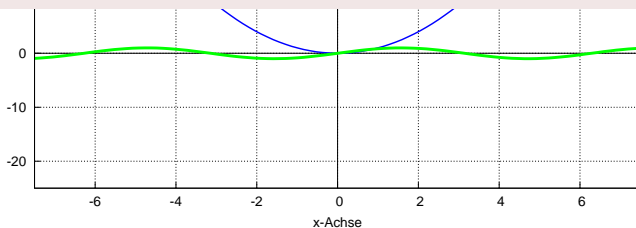
Interaktives Beispiel I



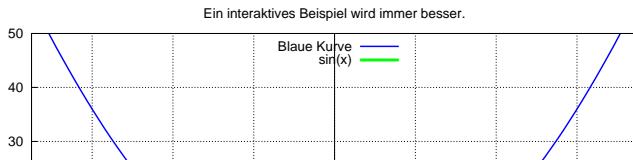
Interaktives Beispiel I



Wir wollen beispielsweise ...

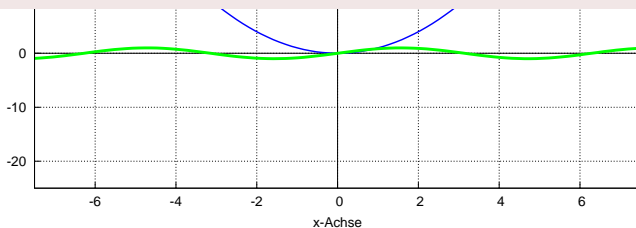


Interaktives Beispiel I



Wir wollen beispielsweise ...

andere Skala für die y-Achse der Sinuskurve



Interaktives Beispiel I

In 2D Bildern gibt es vier Achsen x_1 (unten), x_2 (oben), y_1 (links) und y_2 (rechts).

In 3D Bildern zusätzlich z_1 und z_2 .

Interaktives Beispiel I

In 2D Bildern gibt es vier Achsen x_1 (unten), x_2 (oben), y_1 (links) und y_2 (rechts).

In 3D Bildern zusätzlich z_1 und z_2 .

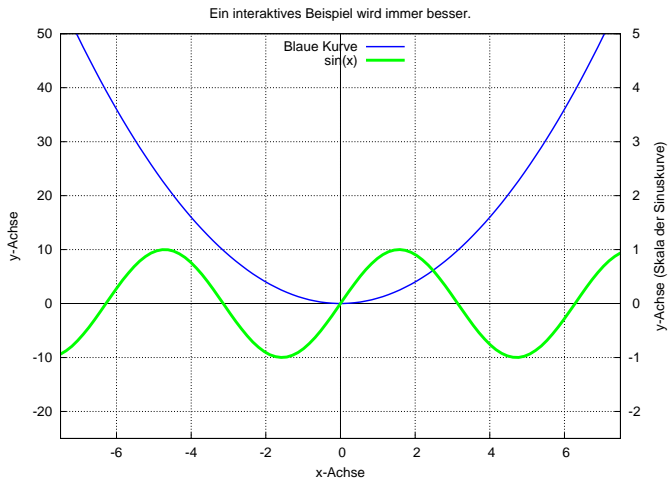
```
gnuplot> set y2range [-2.5:5]
```

```
gnuplot> set y2tics
```

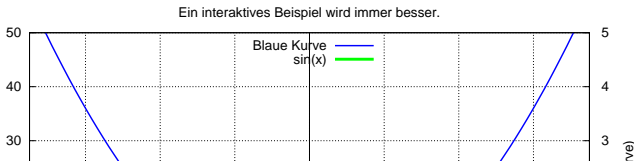
```
gnuplot> set y2label 'y-Achse (Skala der Sinuskurve)'
```

```
gnuplot> plot f(x) lt 1 lc rgb 'blue' lw 3 title 'Blaue Kurve'  
axis x1y1, sin(x) lt 1 lc rgb 'green' lw 6 axis x1y2
```

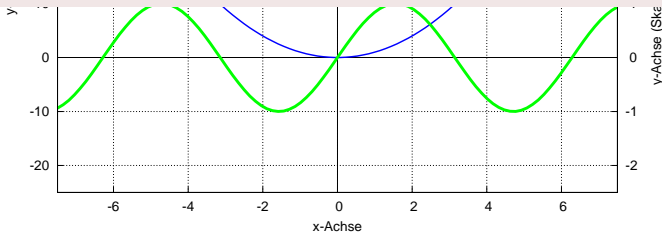
Interaktives Beispiel I



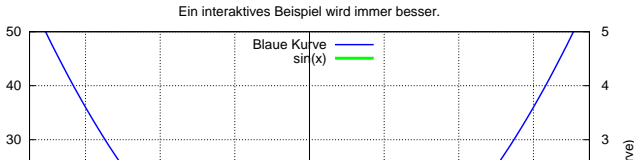
Interaktives Beispiel I



Zufrieden?

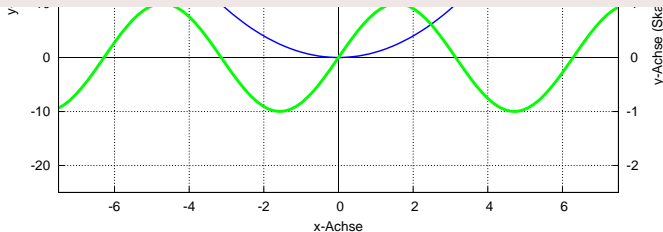


Interaktives Beispiel I



Zufrieden?

Skript speichern, Plot als Datei speichern



Interaktives Beispiel I

Über *set terminal* kann die Ausgabe bestimmt werden und mit *set output* wird die Ausgabedatei angegeben.

Details: *help terminal*, *help output*

Interaktives Beispiel I

Über *set terminal* kann die Ausgabe bestimmt werden und mit *set output* wird die Ausgabedatei angegeben.

Details: *help terminal*, *help output*

```
gnuplot> set terminal postscript color
```

```
gnuplot> set output 'beispiel.ps'
```

```
gnuplot> replot
```

```
gnuplot> save 'beispiel.plt'
```

Interaktives Beispiel I

Über *set terminal* kann die Ausgabe bestimmt werden und mit *set output* wird die Ausgabedatei angegeben.

Details: *help terminal*, *help output*

```
gnuplot> set terminal postscript color
```

```
gnuplot> set output 'beispiel.ps'
```

```
gnuplot> replot
```

```
gnuplot> save 'beispiel.plt'
```

Beim nächsten Mal:

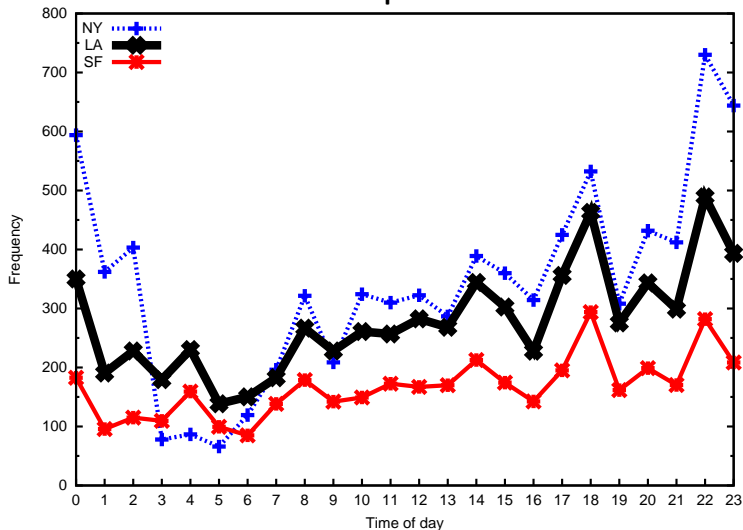
```
gnuplot> load 'beispiel.plt'
```

Outline

- 1 Einführung
 - Plot, plotten, ...
- 2 Fragen
 - Was kann visualisiert werden?
 - Wie können Plots aussehen?
 - Woher kommen die Daten?
 - Wie geht man vor?
- 3 Eigenschaften
- 4 Interaktives Beispiel
- 5 Beispiel mit Messdaten

Beispiel mit Messdaten

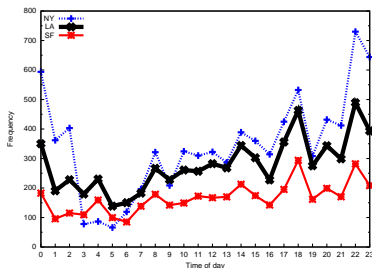
Zurück zu dem Twitter Beispiel



Beispiel mit Messdaten

Das Skript:

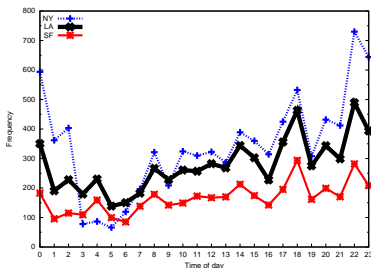
```
set terminal postscript color  
linewidth 2  
set output 'tweets.ps'
```



Beispiel mit Messdaten

Das Skript:

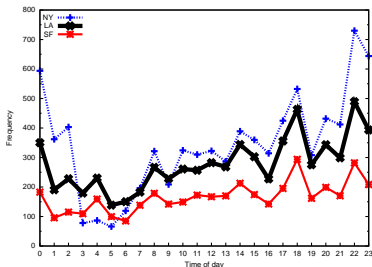
```
set terminal postscript color  
linewidth 2  
set output 'tweets.ps'  
set key inside left top
```



Beispiel mit Messdaten

Das Skript:

```
set terminal postscript color
linewidth 2
set output 'tweets.ps'
set key inside left top
set style data linespoints
set pointsize 2
set xlabel 'Time of day'
set ylabel 'Frequency'
```

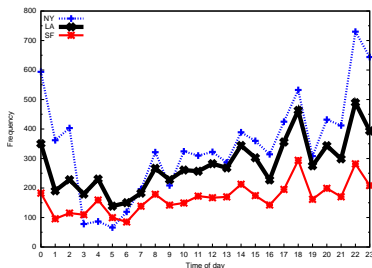


Beispiel mit Messdaten

Das Skript:

```
set terminal postscript color
linewidth 2
set output 'tweets.ps'
set key inside left top
set style data linespoints
set pointsize 2
set xlabel 'Time of day'
set ylabel 'Frequency'

set xtics nomirror
set mytics 2.0
```



Beispiel mit Messdaten

Das Skript:

```
set terminal postscript color
linewidth 2
```

```
set output 'tweets.ps'
```

```
set key inside left top
```

```
set style data linespoints
```

```
set pointsize 2
```

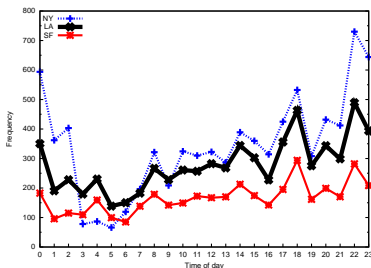
```
set xlabel 'Time of day'
```

```
set ylabel 'Frequency'
```

```
set xtics nomirror
```

```
set mytics 2.0
```

```
plot 'tweet_freq_hourly.dat' using 2 title 'NY' lt 3,
      '' using 3 title 'LA' lt -1,
      '' using 4:xticlabels(1) title 'SF' lt 1
```



Beispiel mit Messdaten

```
plot 'tweet_freq_hourly.dat' using 2 title 'NY' lt 3,  
      " using 3 title 'LA' lt -1,  
      " using 4:xticlabels(1) title 'SF' lt 1
```

Die Daten-Datei:

%#	NY	LA	SF
0	593.8	350.0	182.6
1	362.0	190.7	95.7
2	403.2	228.0	115.0
...			

Beispiel mit Messdaten

```
plot 'tweet_freq_hourly.dat' using 2 title 'NY' lt 3,  
    " using 3 title 'LA' lt -1,  
    " using 4:xticlabels(1) title 'SF' lt 1
```

Die Daten-Datei:

%#	NY	LA	SF
0	593.8	350.0	182.6
1	362.0	190.7	95.7
2	403.2	228.0	115.0
...			

Beispiel mit Messdaten

```
plot 'tweet_freq_hourly.dat' using 2 title 'NY' lt 3,  
      " using 3 title 'LA' lt -1,  
      " using 4:xticlabels(1) title 'SF' lt 1
```

Die Daten-Datei:

%#	NY	LA	SF
0	593.8	350.0	182.6
1	362.0	190.7	95.7
2	403.2	228.0	115.0
...			

Beispiel mit Messdaten

```
plot 'tweet_freq_hourly.dat' using 2 title 'NY' lt 3,  
      " using 3 title 'LA' lt -1,  
      " using 4:xticlabels(1) title 'SF' lt 1
```

Die Daten-Datei:

%#	NY	LA	SF
0	593.8	350.0	182.6
1	362.0	190.7	95.7
2	403.2	228.0	115.0
...			

Beispiel mit Messdaten

```
plot 'tweet_freq_hourly.dat' using 2 title 'NY' lt 3,  
      " using 3 title 'LA' lt -1,  
      " using 4:xticlabels(1) title 'SF' lt 1
```

Die Daten-Datei:

% #	NY	LA	SF
0	593.8	350.0	182.6
1	362.0	190.7	95.7
2	403.2	228.0	115.0
...			

Beispiel mit Messdaten

```
plot 'tweet_freq_hourly.dat' using 2 title 'NY' lt 3,  
      " using 3 title 'LA' lt -1,  
      " using 4:xticlabels(1) title 'SF' lt 1
```

Die Daten-Datei:

%#	NY	LA	SF
0	593.8	350.0	182.6
1	362.0	190.7	95.7
2	403.2	228.0	115.0

...

- Skript immer wieder verwendbar
- Datenfile wird häufig automatisch erstellt

Zusammenfassung

gnuplot visualisiert

- schnell und unkompliziert
- Funktionen aller Art
- Messdaten aller Art

Ideal zum Visualisieren von Ergebnissen und Messungen (mit verschiedenen Einstellungen) in Bachelor-, Master- oder Hausarbeiten!

Quellen und weitere Tutorials (Auswahl)

- Gnuplot Homepage: <http://www.gnuplot.info/>
- Gnuplot Examples:
<http://gnuplot.sourceforge.net/demo/>
- Kurzes Tutorial:
<http://www.duke.edu/~hpgavin/gnuplot.html>
- Gnuplot Grundkurs:
<http://userpage.fu-berlin.de/~voelker/gnuplotkurs/gnuplotkurs.html>
- Gnuplot Skript:
http://www.rz.uni-osnabrueck.de/Zum_Nachlesen/Skripte_Tutorials/index.htm