

Vi ska nu göra inställningar för att plotta data i listorna som ett spridningsdiagram. Se till att det inte finns några inmatade funktioner i räknaren. Tryck på tangenten $\boxed{y=}$ för att kontrollera. Man kan ha funktioner inmatade men då får man istället avmarkera funktionen genom att trycka på likhetstecknet.

Ställ in fönstervisningen så här:

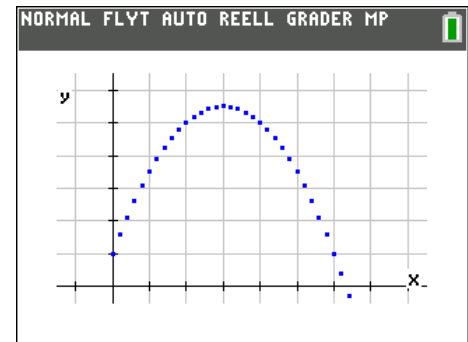
X_{\min} , 1.5, X_{\max} 8,5, Y_{\min} = -0.5, Y_{\max} = 6.5.

b) Importera data till ett Python-skript

- Starta ett nytt skript och döp det till KAP5OVN1
- Skapa två variabellistor (tomma) **xaxel** och **yaxel**.
- Importera modulerna **ti_system** och **tiplotlib** (det finns ingen viss ordning här)
- Skapa en variabel xaxel och sedan från **ti_system**-biblioteket väljer du alternativ **2: var-recall_list ("namn")**. eftersom x-axelns data finns i L_1 ska fältet "namn" betecknat med talet 1.
- På liknande sätt en variabel för y-axelns data och den skriver vi som yaxel= recall_list(2).-

Lärarkommentar: Att skapa tomma listor xaxel=[] och yaxel=[] är inte nödvändigt eftersom de kommer att skapas när listorna L_1 och L_2 återkallas. Det är dock bra att hålla sig till de goda vanor eleverna lärt sig när vi inte använder modulen ti_system.

- Kör skriptet och kontrollera sedan innehållet i dina xaxel- och yaxel-variabler genom att trycka på tangenten $\boxed{\text{vars}}$.
- Återkalla sedan namnet på xaxel- "listan" och validera detta.
- Gör samma sak med listan för y-värden



```

EDITOR: KAP5OVN1
ti_system module
ti_system
1:from ti_system import *
2:var=recall_list("name") 1-6
3:store_list("name",var) 1-6
4:var=recall_RegEQ()
5:while not escape(): [clear]
6:if escape():break [clear]
7:disp_at(row,"text","align")
8:disp_clr() clear text screen
9:disp_wait() [clear]
0:disp_cursor() 0=off 1=on
    
```

```

EDITOR: KAP5OVN1
PROGRAM LINE 0006
from ti_system import *
import tiplotlib as plt
xaxel=[]
yaxel=[]
xaxel=recall_list("1")
yaxel=recall_list("2")_
    
```

```

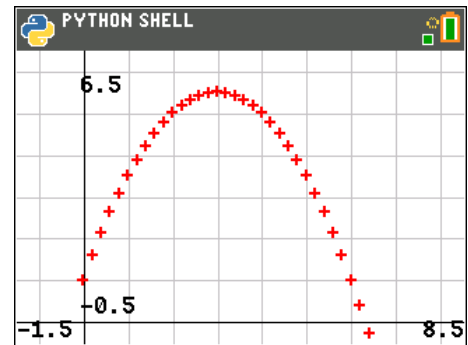
PYTHON SHELL
VARS: KAP5OVN1
>xaxel
yaxel
    
```

c) Grafisk representation

Gör nu inställningar för grafen enligt skärmen här till höger.

```
EDITOR: KAP5OVN1
PROGRAM LINE 0015
xaxel=recall_list("1")
yaxel=recall_list("2")
#inställning för grafen
plt.cls()
plt.window(-1.5,8.5,-0.5,6.5)
plt.grid(1,1,"solid")
plt.axes("on")
plt.color(255,0,0)
plt.scatter(xaxel,yaxel,"+")
plt.show_plot()
```

Exekvera nu skriptet (Run).



2. Exportera data

Skapa ett nytt skript och döp det till KAP5OV2B

Lärarkommentar: Placera markören i slutet av en rad och validera. Ordningen på modulerna har ingen betydelse.

Skapa först en funktion **data (a,b,n)**.

Du skapar sedan två listor med data som grafiskt ska representeras i form av ett spridningsdiagram med värden inom ett intervall **[a; b]**, beräknade med steglängden **n**.

I **listan y** beräknar vi kvadratroten för x-listans värden.

För att skapa dessa datalistor bygger vi en avgränsad loop efter att naturligtvis ha skapat två tomma listor.

```
EDITOR: KAP5OV1B
PROGRAM LINE 0001
from math import *
from ti_system import *
import ti_plotlib as plt
def data(a,b,n):
  x=[]
  y=[]
  for i in range(a,b,n):
    x.append(i)
    y.append(sqrt(x[i]))
  store_list("1",x)
  store_list("2",y)
```

Lärarkommentar: Om du skapar två tomma listor förhindras att ett felmeddelande kommer tillbaka när skriptet körs.

För att exportera listorna till räknaren så är syntaxen:

store_list("list number","name").

listnumret hos räknaren motsvarar listorna L₁ to L₆. Om vi kör skriptet och sedan trycker på tangenten **vars** och sedan skriver

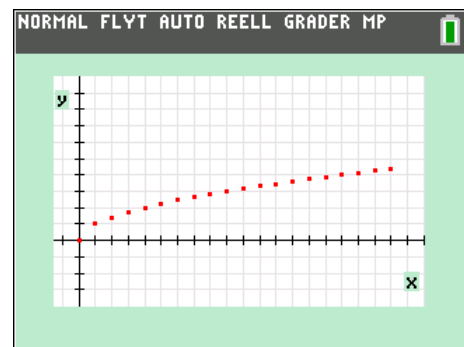
data(0,20,1)

så får vi i editorn i listhanteraren (när vi lämnat Python-appen) i statistik-editorn data enligt skärmen till höger.

L1	L2	L3	L4	L5	1
0	0	-----	-----	-----	
1	1				
2	1.4142				
3	1.7321				
4	2				
5	2.2361				
6	2.4495				
7	2.6458				
8	2.8284				
9	3				
10	3.1623				

L1(1)=0

Om vi nu plottar våra data i ett fönster och ritar ett spridningsdiagram blir det så här:



Lärarkommentar: Utforskande av räknarlistor blir särskilt intressant när vi samlar in data med styrenheten TI-Innovator™ Hub och sedan utforskar rörelse med robotfordonet TI-Innovator™ Rover.

