

Kapitel 5: Använda biblioteksmodulen ti-plotlib

I denna andra övning i kapitel 5 kommer du att upptäcka hur man använder biblioteket **ti_system** för att importera och exportera listor i ett Pythonskript.

I den här lektionen kommer du att utföra linjär modellering med hjälp av data som tidigare angetts i räknarlistorna. Då skriver du ett skript som importerar resultaten av den här modelleringen för att använda dem för en grafisk representation, en interpolering eller extrapolering ...

Scenariot: Vanligtvis är effektuttaget som störst omkring kl. 19. Det är alltså den tidpunkt när ett samhälle konsumerar mest energi. Effektuttaget mäts i watt. Du vill förutsäga konsumtionstopparna nästa helg inom ett område som ligger nära en kontrollcentral. För att åstadkomma denna prognos har du tio värden uppmätta på effektuttaget i MW klockan 19.00 vid olika utetemperaturer.

T (°C)	11	-5	-8	-6	9	14	4	-1	-12	3
Mw	3	5.7	6.6	6.3	3.4	2.7	4	5.1	7.1	4.6

Data är inmatade i räknarens statistikeditor. Temperaturen i L₁ och effektuttaget i MW i L₂.

För att göra en modellering med linjär regression så trycker du på **stat** och sedan **BERÄK** och alternativ 4:Linreg(ax+b).

Övning 2: Modellering

Syfte:

- Utforska en linjär modell
- Importera resultaten av en linjär modellering till ett Pythonskript.

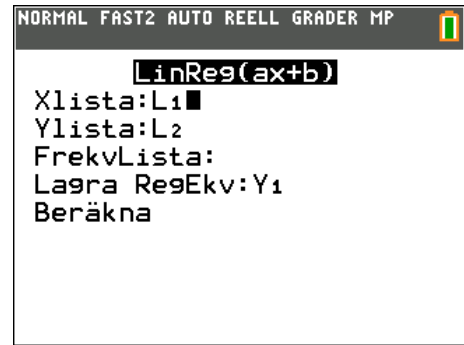


L1	L2	L3	L4	L5	3
11.00	3.00				
-5.00	5.70				
-8.00	6.60				
-6.00	6.30				
9.00	3.40				
14.00	2.70				
4.00	4.00				
-1.00	5.10				
-12.00	7.10				
3.00	4.60				

L3(1)=

REDIGERA BERÄK TESTER	
1:	1-Var-stat
2:	2-Var-stat
3:	Med-Med
4:	LinReg(ax+b)
5:	KvadReg
6:	KubikReg
7:	4-gradsReg
8:	LinReg(a+bx)
9↓:	LnReg

Om du nu trycker på `enter` så kommer följande fönster fram och du fyller nu i enligt skärmen. Variabeln Y1 når du genom att trycka på `alpha` f4 (trace-tangenten) och sedan klickar du i Y1.



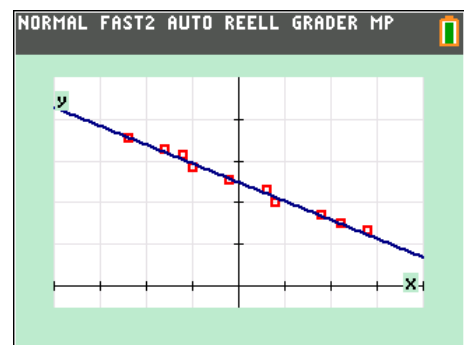
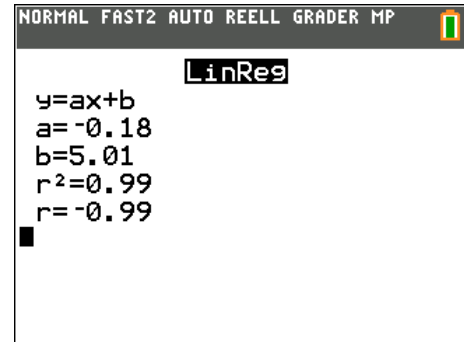
Markera därefter på Beräkna och tryck på `enter`. Nu får du ett resultatfönster på skärmen.

Den linjäras funktion som beskriver hur effektuttaget K i MW beror av temperaturen t kan skrivas som

$$K = -0,18 \cdot t + 5,01$$

r är korrelationskoefficienten och beskriver hur väl datapunkterna är anpassade till den beräknade regressionsekvationen.

Om vi nu plottar datapunkterna och den beräknade regressionslinjen ser det ut som nedan. Vi ser hur väl punkterna följer linjen.



Använda resultaten i ett Python-skript

- Starta ett nytt skript och döp det till KAP5OVN2.
- Lägg till `ti_system` och `ti_plotlib`
- Skapa två listor, `temp[]` och `kons[]`



Återkalla innehållet i L_1 - och L_2 -listorna i sina respektive namn.

Satsen `var-recall_list("name")` finns i menyn för `ti_system`-modulen (se kapitel 5 Övning 1).

Kom också ihåg ekvations-variabeln som är uttrycket för den beräknade linjära modellen.

Nu kan du *testa* ditt skript och be om visning av de olika variablerna.

Tryck på `[vars]` och välj sedan variabel.

Vi ser att regressionsekvationen är importerad som en teckensträng.

För att slutföra den här övningen och repetera de kunskaper som förvärvats i kapitel 4 kan du nu göra en grafisk representation av dina mätningar och den beräknade regressionsmodellen.

Instruktionen `plt.lin_reg ()` motsvarar alternativ **8**: `plt.lin_reg (x-list, y-list,"disp")` på `ti_plotlib`-bibliotekets meny **Draw**.

Obs: Instruktionen `The plt.auto_window (x_list, y_list)` justerar automatiskt fönsterinställningarna. Det motsvarar **ZoomStat** på räknaren.

Lärarkommentar: `lin_reg (xlist, ylist, "disp",)` inkluderar ytterligare en instruktion ger möjlighet att placera uttrycket för regressionsekvationen vid en annan rad. Standardvärdet är rad 11.

```

EDITOR: KAP5OVN2
PROGRAM LINE 0008
from ti_system import *
import ti_plotlib as plt
temp=[]
kons=[]
temp=recall_list("1")
kons=recall_list("2")
ekv=recall_RegEQ()
-
    
```

```

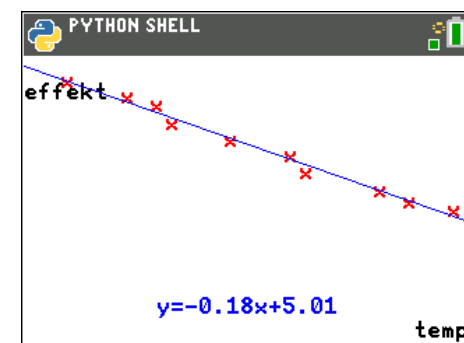
PYTHON SHELL
VARS: KAP5OVN2
▸ekv
kons
temp
    
```

```

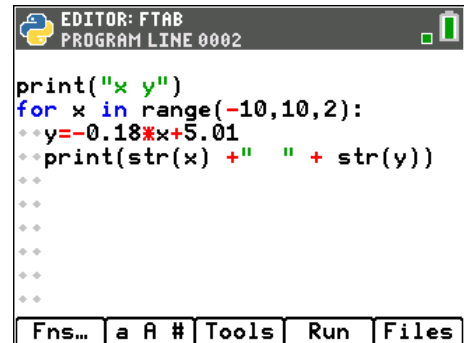
PYTHON SHELL
>>> temp
[11.0, -5.0, -8.0, -6.0, 9.0, 14.0, 4.0, -1.0, -12.0, 3.0]
>>> kons
[3.0, 5.7, 6.6, 6.3, 3.4, 2.7, 4.0, 5.1, 7.1, 4.6]
>>> ekv
'-0.18*x+5.01'
>>> type(ekv)
<class 'str'>
>>> |
    
```

```

EDITOR: KAP5OVN2
PROGRAM LINE 0017
plt.cls()
plt.window(-15,15,-2,8)
plt.color(0,0,0)
plt.labels("temp","effekt")
plt.axes("off")
plt.color(255,0,0)
plt.scatter(temp,kons,"x")
plt.color(0,0,255,)
plt.lin_reg(temp,kons,"center",1)
plt.show_plot()
    
```



Om du vill visa en tabell med funktionsvärden inom ett visst intervall och en viss steglängd kan du skriva in programmet nedan. Det finns en del andra sätt att göra detta på men det här är ett enkelt och kort program.



```
EDITOR: FTAB
PROGRAM LINE 0002

print("x y")
for x in range(-10,10,2):
    y=-0.18*x+5.01
    print(str(x) + " " + str(y))

```

Fns... a A # Tools Run Files