

## Kapitel 5: Listor, grafik och dynamiska program

## Övning 2: Loopar och listor

I denna aktivitet kommer du att lära dig att styra individuella element i listor och upptäcka en metod att köra program dynamiskt från en matematikruta i appen **Anteckningar**.

## I föregående aktivitet...

... skrev vi ett program som genererade två listor med slumpmässiga heltal och visade dessa listor grafiskt i ett spridningsdiagram i graf-appen. Till höger visar vi utdata från programmet **slumppunkter(n)** i form av ett spridningsdiagram.

Föregående program var begränsat eftersom det bara genererade heltal för x- och y-koordinaterna hos spridningsdiagrammet. I denna aktivitet ska vi utvidga programmet till att omfatta decimaltal inom samma intervall som tidigare. Detta gör att punkterna blir mer tätt packade.

Istället för att använda **randInt(-10, 10)** ska vi nu använda **rand( )**-funktionen för att alstra ett decimaltal mellan 0 och 1 och sedan "skala" så att punkterna fyller skärmen med fönsterinställningen Standard.

Vi ska nu först undersöka **rand( )** and **rand(k)**-funktionerna i Räknare-appen. Vi ser att **rand** utan argument, **rand( )**, genererar ett slumpmässigt decimaltal mellan 0 och 1. Med argumentet 3, får vi en lista med tre decimaltal mellan 0 och 1. Denna lista kan sedan användas som en del av ett uttryck för att anpassa dessa värden så att de går utanför intervallet [0, 1]. **5•rand(3)** ger en lista med tre slumpmässiga decimaltal mellan 0 och 5.

Vi använder nu denna kunskap för att generera ett slumpmässigt decimaltal mellan -10 och 10. Det är ett intervall med bredden 20. För x-listan börjar vi då uttrycket med -10 och lägger till ett slumpmässigt decimaltal mellan 0 och 20 (**20•rand( )**). För y-listan gör vi motsvarande.

**Lärarkommentar:** Med **rand( )** kan vi generera ett slumpmässigt decimaltal i intervallet [A, B]. Uttrycket blir  $n := A + (B - A) \cdot \text{rand}()$ , eller med ord  $n := \text{startvärde} + \text{intervallstorlek} \cdot \text{rand}()$ .

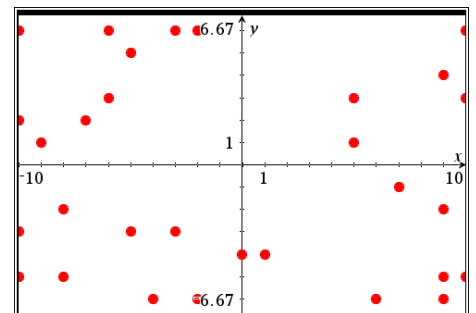
- I programmet **slumppunkter(n)**, ändrar du **randInt**-funktionen till uttryck med **rand( )**-funktionen istället:

**xpunkt := -10 + 20•rand(n)**

**ypunkt:= -6 + 12•rand(n)**

## Syfte:

- Alstra slumpmässiga reella tal i ett önskat intervall
- Använda den interaktiva egenskapen hos en Matematikruta i Antecknings-appen för att få program att köras interaktivt.



|           |                                |
|-----------|--------------------------------|
| rand()    | 0.701089                       |
| rand(3)   | {0.882225, 0.370913, 0.117576} |
| 5•rand(3) | {2.25297, 4.94528, 3.48872}    |
| 5•rand(3) | {2.72034, 3.18915, 4.64755}    |
| 5•rand(3) | {3.37119, 3.99011, 2.76104}    |
|           |                                |

| slumppunkter                      | 2/2 |
|-----------------------------------|-----|
| Define <b>slumppunkter(n)</b> =   |     |
| Prgm                              |     |
| <b>xpunkt</b> := -10 + 20•rand(n) |     |
| <b>ypunkt</b> := -6 + 12•rand(n)  |     |
| EndPrgm                           |     |

## 10 Minutes of Code

### TI-Nspire-teknologi

## KAPITEL 5: ÖVNING 2

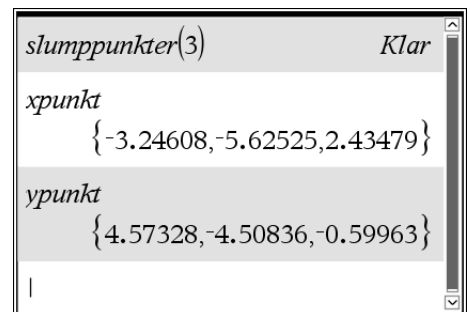
### LÄRARKOMMENTARER

Dessa uttryck genererar  $n$  värden av **xpunkt** mellan -10 och 10 och  $n$  värden av **ypunkt** mellan -6 och 6.

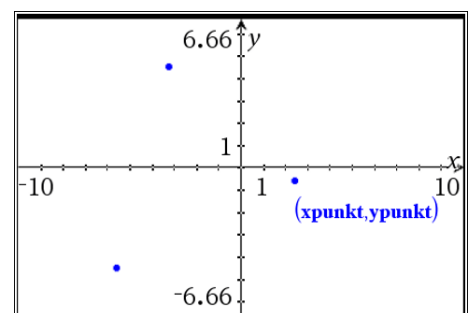
2. Kör programmet appen Räkna och observera värdena på **xpunkt** och **ypunkt**.

**Lärarkommentar:** När du arbetar med en ny funktion i TI-Nspire™ CX kan det vara bra att studera funktionen i appen Räkna först innan du försöker använda den i ett program. Man kan också leta upp funktionen i *katalogen*, där man ofta kan få hjälp med nödvändiga och valfria argument och resultat vid beräkningar med funktionen.

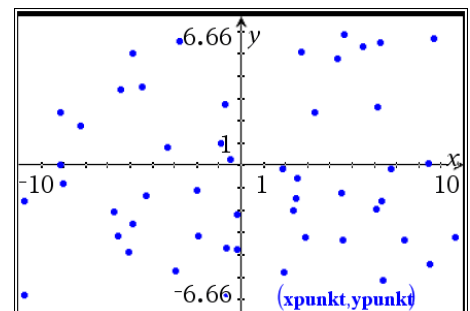
- 3 Titta nu på värdena för **xpunkt** och **ypunkt** i ett spridningsdiagram.



```
slumppunkter(3) Klar
xpunkt
{-3.24608, -5.62525, 2.43479}
ypunkt
{4.57328, -4.50836, -0.59963}
```

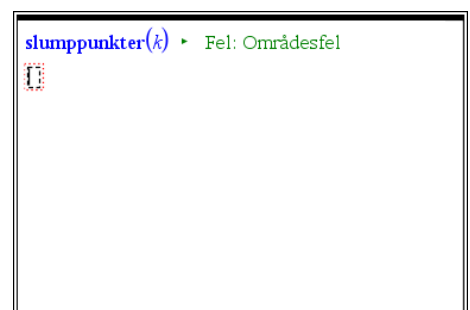


- 4 Kör nu programmet igen med större värden på argumentet  $n$ . Till höger finns en bild efter körning av **slumppunkter(50)**, där decimalvärdena erhöles med funktionen `rand( )`.



### Dynamiska Program

1. För att få programmet att köras dynamiskt så ska du nu lägga till appen Anteckningar till problemet. (När du skapar ett nytt dokument består det av *ett enda* problem med *en enda* sida. Du kan alltid infoga nya problem och lägga till sidor till varje problem.)
2. Infoga en Matematikruta i appen genom att från verktygsmenyn till vänster välja <Infoga> och sedan <Ruta för matematiskt uttryck>. Du kan också direkt trycka Ctrl M.
3. Skriv nu inom rutan namnet på programmet med en variabel (vi använder här  $k$ ) som argument och tryck [enter]. Du kan också klicka på knappen **var** och därifrån välja programmet.
  - Du kommer att se ett felmeddelande eftersom ditt argument är en odefinierad variabel. Detta kommer att lösas inom kort.



```
slumppunkter(k) Fel: Områdesfel
```

Den parameter du använder behöver inte vara samma som argumentet som du använde i programeditorn. Det är värdet som "överförs" till programmets argument.

4. Infoga nu ett skjutreglage på grafsidan som innehåller ett spridningsdiagram. (**Meny > Åtgärder > Infoga skjutreglage**).

**Lärarkommentar:** I nästa avsnitt av denna aktivitet introduceras ett annat sätt att köra ett program och där man använder Matematikrutorna som finns i Anteckningsappen. Denna funktion är unik för TI-Nspire™ CX-plattformen och ger möjlighet att köra ett program när en global variabel som används av programmet eller ett argument till funktionen ändrar värdet. Ändringen utlöser en dynamisk ”omkörning” av programmet utan att behöva starta det på nytt i Räknare-appen.

5. Använd nu samma variabelnamn som du använde som argument i matematikrutan i anteckningsappen. Här använde vi  $k$ .
6. Ställ in värdet till 1, Minimum till 1, maximum till 100 och stegstorleken till 1. Se skärmbilden till höger.
7. Tryck nu på [enter] för att placera skjutreglaget.
8. Flytta reglaget till en plats som passar och tryck på [enter] eller klicka igen för att placera det.

Du ser att alla punkter förutom en försvinner. Detta beror på att programmet svarar på det nya värdet på  $k$  (i vårt exempel) vilket får programmet att svara från Matematikrutan på anteckningssidan.

Om du nu tittar på Anteckningssidan så ser du att felmeddelandet har ersatts av ordet Klar. Programmet fungerar nu som det ska när argumentet  $k$  är definierat.

9. Nu kan du dra i skjutreglaget för att ändra värdet på  $k$ . När värdet på  $k$  ändras så körs programmet igen och det genereras en ny uppsättning av punkter i spridningsdiagrammet.

**Lärarkommentar:** När du med skjutreglaget ändrar värdet med ett steg för argumentet, ritas en helt ny uppsättning punkter upp med en punkt tillagd. En utmanande programmeringsuppgift kan vara att lägga till *en* ny punkt till det *nuvarande* spridningsdiagrammet när man ökar värdet hos reglaget ökar och ta bort den sista plottade punkten när värdet minskar.

