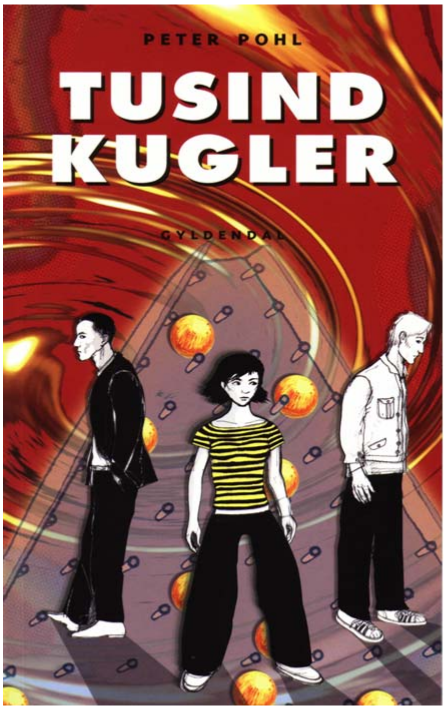
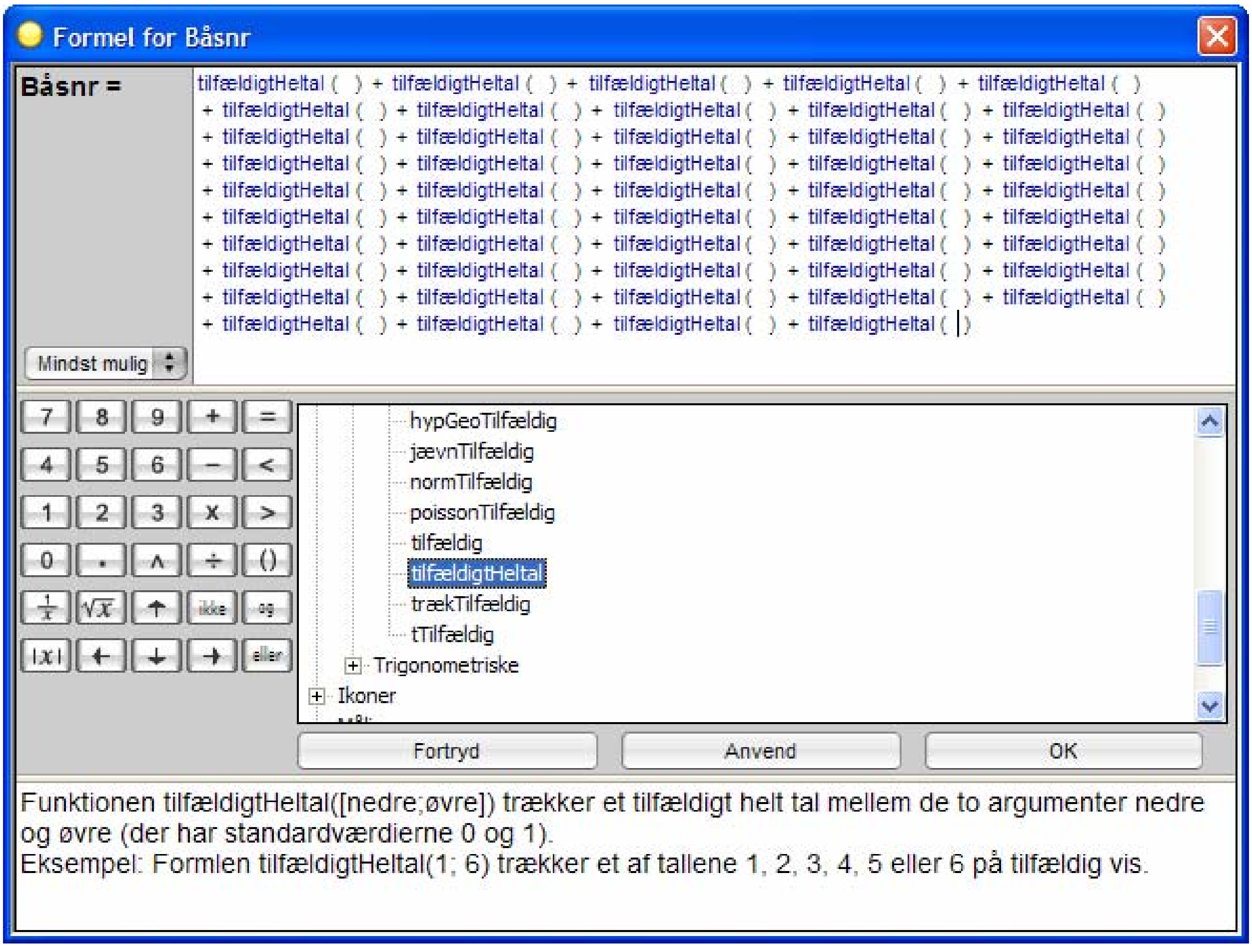
Tusind kugler



I en nyligt udkommen ungdomsroman om en ung svensk gymnasiepiges fortrædeligheder bruges et kuglespil ('galtonbræt') som en metafor for tilværelsens tilfældige forudsigelighed. Den matematisk begavede hovedperson Åsa er dybt fascineret af de naturvidenskabelige forklaringsmodeller med deraf følgende spændinger for hende i forholdet til hendes mandlige lærer i samfundsfag. I et appendiks beskriver forfatteren nu nærmere virkningen af et sådant kuglespil med 50 opsamlingsbåse, hvor de 1000 kugler skiftevis triller til højre og venstre i 49 trin når de rammer piggene i kuglespillet.

Bogen giver en meget fin beskrivelse af det random walk mønster (diskret normalfordeling), som hovedpersonen Åsa er så fascineret af, og som hun bruger megen tid på at studere; først konkret med et galtonbræt hun har fået forærende som barn, siden elektronisk efter at hun som fjortenårig har fået en computer at lege med. Men har forfatteren selv leget med bogens kuglespil? Er fx beskrivelsen af variationen i antallet af kugler der havner i de enkelte bokse realistisk? Hvordan kan man nu undersøge det?

Det er ikke særligt nemt at konstruere et præcist kuglespil, men vi kan nemt **simulere** kuglespillet i et databehandlingsprogram, fx et regneark, en grafisk lommeregner, TI-Interactive eller som her DataMeter. Ved at udnytte tilfældighedsgeneratoren **tilfældigtHeltal(0;1)**, der veksler tilfældigt mellem tallene 0 og 1, kan vi simulere udfaldet af en enkelt kugle ved at lægge 49 af disse stokastiske funktioner sammen fx i grupper af 7. Det gør nok indskrivningen af formlen lidt nemmere at bruge **kopiér** og **indsæt** flittigt undervejs! I DataMeter trækker vi altså en tabel ind i dokumentet, opretter en variabel **Båsnr**, tænder for **Vis formler** i **Tabel**-menuen og indskriver formlen (hvor det er vigtigt at huske at benytte **semikolon** som skilletegn); eller endnu nemmere, vi ignorerer argumenterne, da de under alle omstændigheder har standardværdierne 0 og 1. Endelig giver vi tabellen (og dermed datasættet) titlen **Tusind kugler**:



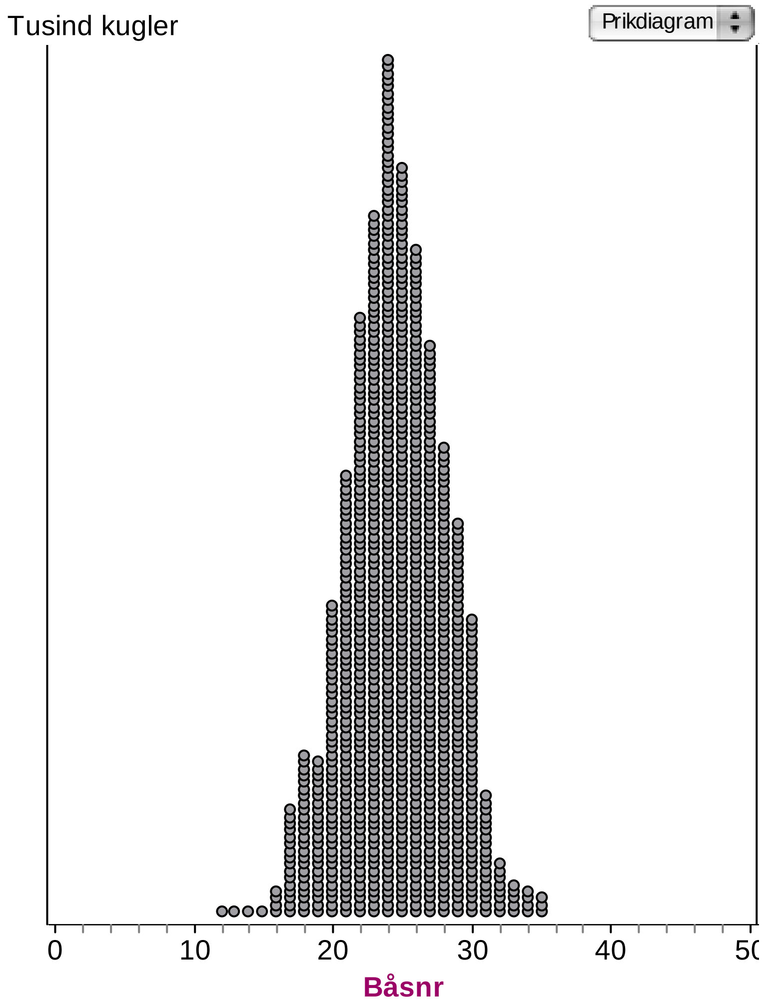
Bemærkning: Jo, der findes mere elegante måder at udføre eksperimentet på, men dette er nok den nemmeste i første omgang, og den kan bruges i alle slags databehandlingsprogrammer! Vi vil senere se på mere avancerede varianter, der trækker på mere specielle faciliteter, som ikke nødvendigvis findes i et databehandlingsprogram.

Derefter oprettes 1000 nye data (ved at højreklikke og vælge **Tilføj data**) og vi har fået opbygget udfaldene for de 1000 kugler.

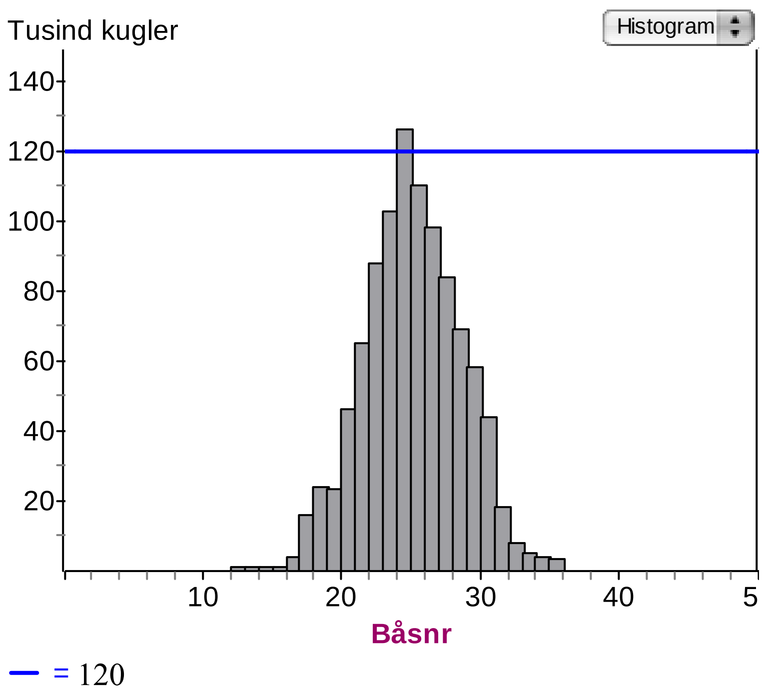
Tusind kugler

	Båsnr
=	tilfældigtHeltal ( ) +
1	26
2	23
3	18
4	31
5	22
6	22
7	28
8	24
9	25
10	20

Vi er nu klar til at oprette et **histogram** for fordelingen ved først at trække et grafvindue ind i dokumentet og dernæst trække variablen **Båsnr** ind i grafrummet på førsteaksen. Vi kan således efter få minutter ved selvsyn se hvordan et realistisk forløb af kuglespillet ser ud. Faktisk kan vi ved at strække et prikdiagram få en rimelig realistisk fornemmelse af kuglebanens udseende.



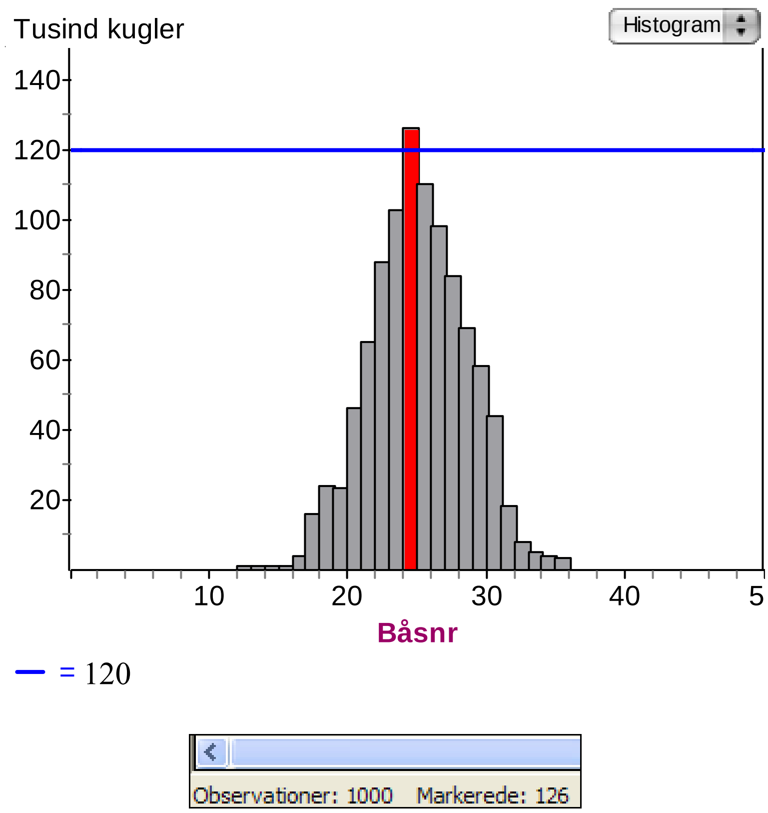
Men tilbage til histogrammet (hvor intervalbredden sættet til 1):



I teksten står nu følgende påstand:

I hver af de to midterste båse havnede mellem 110 og 115 kugler, nogle gange lidt flere, nogle gange lidt færre, men aldrig mere end 120, hvilket spillets konstruktør må have været klar over, idet båsene ikke kan rumme mere end 120 kugler. Nogle gange var der en eller to kugler mere i den højre bås, nogle gange i den venstre. For det meste var der lige mange og aldrig færre end 104.

Først er der påstanden om at der aldrig falder mere end 120 kugler i de to midterste båse og at det derfor ikke er nødvendig at lade opsamlingsbåsene rumme mere end 120 kugler. Det kan vi som vist checke ved at højreklikke og vælge **Plot Funktion** og dernæst indskrive den konstante funktion 120 i formeleditoren. Allerede i det første forsøg kan vi altså se at vi har ramt loftet idet opsamlingsbåsen nr. 24 rummer 126 kugler som det også fremgår af statusbjælken, når man fører musen hen til opsamlingsbåsen og marker den:



Vi gentager nu simuleringen for at udføre en hurtig optælling af hvor hyppigt vi bryder loftet. I DataMeter sker det fx ved at bruge kommandoen **Gentag simulering** til at gentage kuglespillet. Kommandoen udføres enten med tastaturgenvejen **Ctrl-U** (simUlering) eller den kan findes i øverste højre hjørne af et udfoldet datasæt. I mit dokument ser jeg for eksempel at 5 ud af de første 10 kuglespil gennembryder loftet. Vi gennembryder altså loftet ca. halvdelen af gangene.

Forfatteren har altså ikke selv prøvet at lege med et kuglespil for så ville han have haft en meget klarere fornemmelse af variationen i spillene.

Mellemspil:
Vi kan udbygge eksperimentet med en **måling** til at finde fordelingen for hvor ofte loftet gennembrydes. Vi må da først konstruere en hyppighedstabel for kuglespillet, så vi kan finde den maksimale hyppighed. I DataMeter sker det ved at trække beregningsværktøjet ned i dokumentet og derefter trække variablen **Båsnr** ind i tabellen, men husk at *holde* **SKIFT**-tasten nede, for tricket består i at håndtere **Båsnr** som en kategoriseret variabel:

Tusind kugler

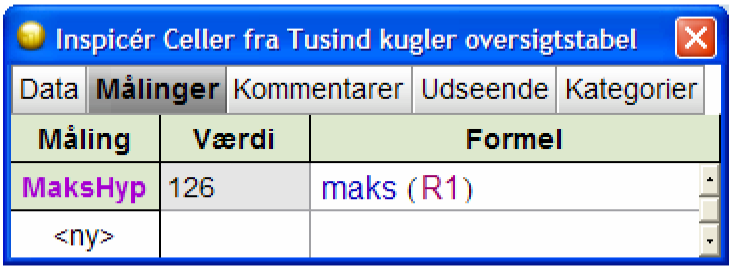
Båsnr	12	1
	13	1
	14	1
	15	1
	16	4
	17	16
	18	24
	19	23
	20	46
	21	65
	22	88
	23	103
	24	126
	25	110
	26	98
	27	84
	28	69
	29	58
	30	44
	31	18
	32	8
	33	5
	34	4
	35	3
	Søjle total	1000

R1 = tæl ( )

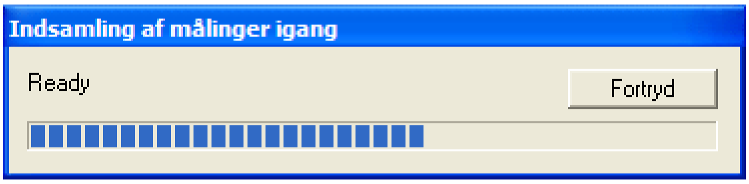
Celler fra Tusind kugler oversigtstabel

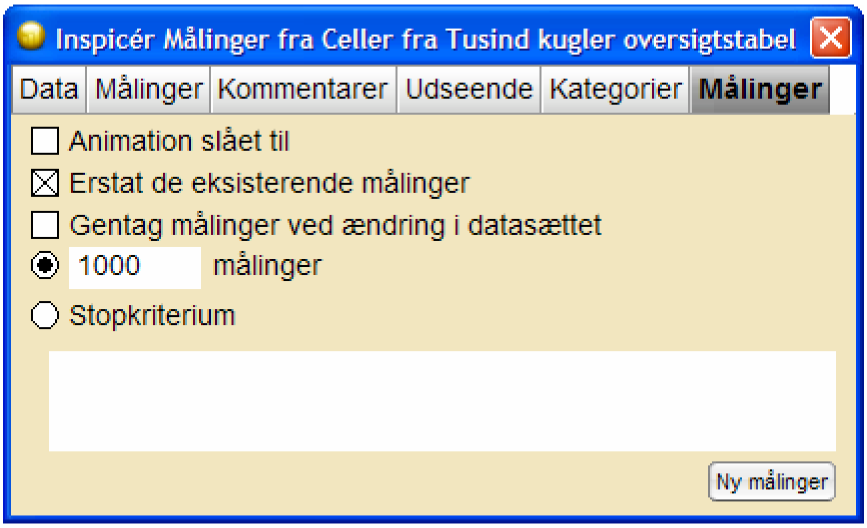
	Båsnr	R1	<ny>
1	12	1	
2	13	1	
3	14	1	
4	15	1	
5	16	4	
6	17	16	
7	18	24	
8	19	23	
9	20	46	
10	21	65	
11	22	88	
12	23	103	
13	24	126	
14	25	110	
15	26	98	
16	27	84	
17	28	69	
18	29	58	
19	30	44	
20	31	18	
21	32	8	
22	33	5	
23	34	4	
24	35	3	

I dette tilfælde var det altså bås nr. 24 der fik det største antal kugler, nemlig 138 (og dermed stødte mod loftet). Men resultatet af en sådan hyppighedstabel kan som vist samles i et selvstændigt afledet datasæt ved at højreklikke og vælge menupunktet **Overfør celler til nyt datasæt**. Og når vi først har fået overført hyppighederne til et afledet datasæt kan vi udføre en måling på dette afledte datasæt **Celler fra oversigtstabel for Tusind kugler**. Dermed kan vi trække den maksimale hyppighed ud som en måling. Da målingerne ligger gemt i selve datasættet dobbeltklikker vi på dette og åbner dets inspektør, hvorefter vi skifter til fanebladet **Måling**:



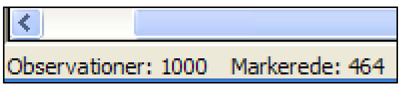
Så snart vi har oprettet målingen **MaksHyp** kan vi udføre gentagne målinger fra Datasæt-menuen eller ved at højreklikke på datasættet **Celler fra oversigtstabel for Tusind kugler**. Vi vælger da at gentage målingen 1000 gange (det kræver lidt tålmodighed, da vi i så fald skal kaste de 1000 kugler 1000 gange):

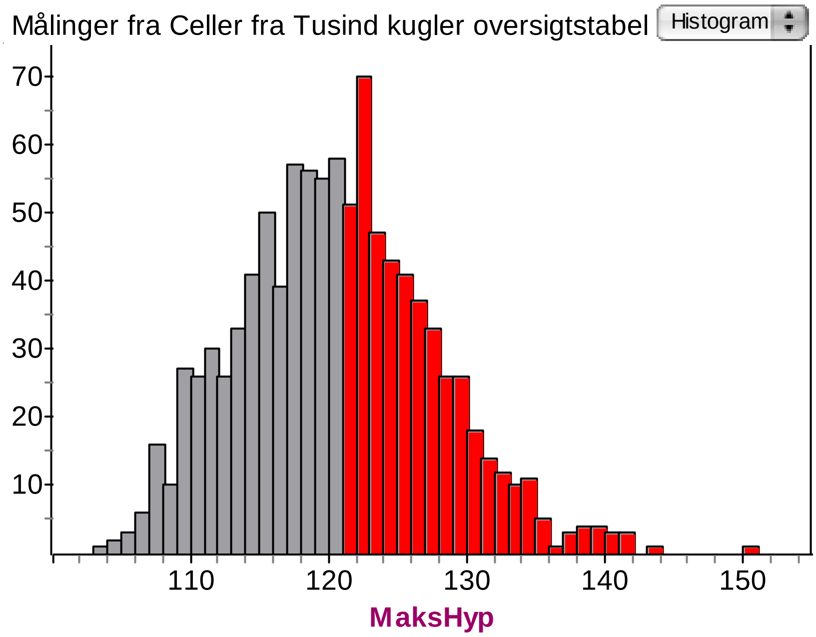




Først udføres målingen automatisk fem gange med animationen slået til. Ved at dobbeltklikke på datasættet for målingerne kan vi efterfølgende som vist slå animationen fra og udføre 1000 nye målinger, der erstatter de første.

Vi trækker derefter den gentagne måling af **MaksHyp** ind i et grafrum og markere samtidigt målingerne fra 121 og frem. Som det ses overskrider en betydelig del af de tusinde målinger den kritiske værdi 120. I modsætning til tekstens påstand er det altså et højst normalt fænomen at kuglerne løber ud over den øvre grænse. Ved at markere området til højre for den kritiske grænse kan vi hurtigt tælle hvor mange gange ud af 1000, idet resultatet aflæses på statusbjælken når vi fører markøren op til datasættet for de gentagne målinger:





I 46% af tilfældene overskrider vi altså den kritiske grænse i en af de centrale opsamlingsbåse. Derimod kommer vi kun meget sjældent under 104, præcis som bogen påstår.

Målinger fra Celler fra Tusind kugler...

	464
	1

R1 = tæl (MaksHyp > 120)
R2 = tæl (MaksHyp < 104)

Ved hjælp af sådanne simple eksperimenter kan vi også hurtigt checke nogle af de andre påstande, fx påstanden om hvor meget antallet af kugler i de to centrale båse ligner hinanden:

Nogle gange var der en eller to kugler mere i den højre bås, nogle gange i den venstre. For det meste var der lige mange.

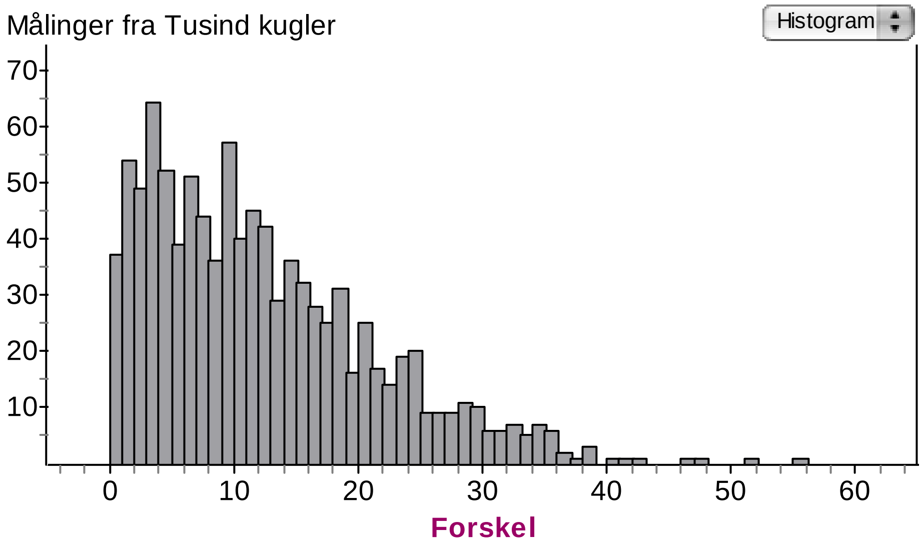
Påstanden kan give anledning til at formode at der typisk er næsten lige mange kugler i de centrale båse med numrene 24 og 25. For at undersøge dette indføres målingen **Forskel** i det oprindelige datasæt:

Inspicér Tusind kugler

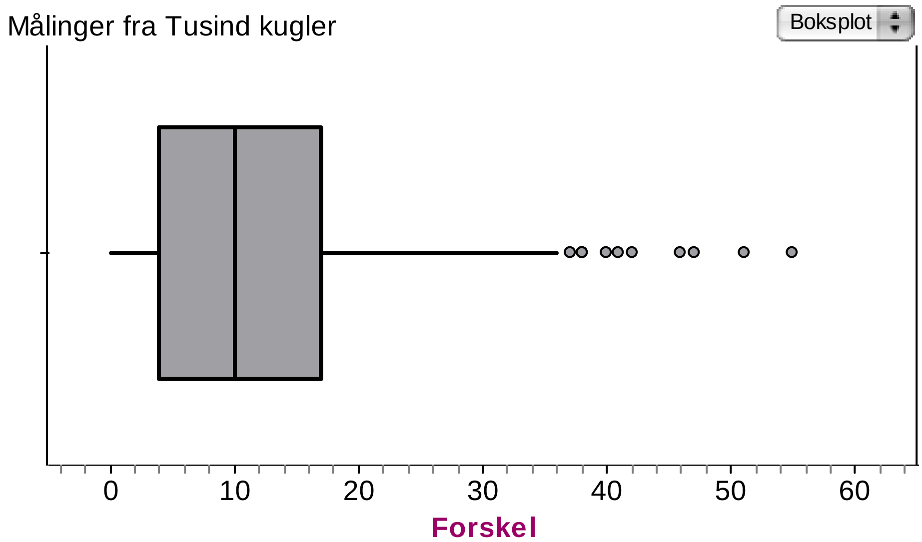
Data | **Målinger** | Kommentarer | Udseende | Kategorier

Måling	Værdi	Formel
Forskel	34	abs (tæl (BåsNr = 24) − tæl (Båsnr = 25))
<ny>		

Som før gentager vi målingen 1000 gange, hvilket igen kræver en del tålmodighed, da vi jo igen skal kaste de 1000 kugler i alt 1000 gange. Trækker vi derefter den gentagne måling **Forskel** ind på førsteaksen i et grafrum og opretter histogram og **Boksplot** fås følgende:



Der er intet der tyder på at der for det meste er lige mange kugler i de to centrale opsamlingsbåse. Det sker kun i under 4% af tilfældene.



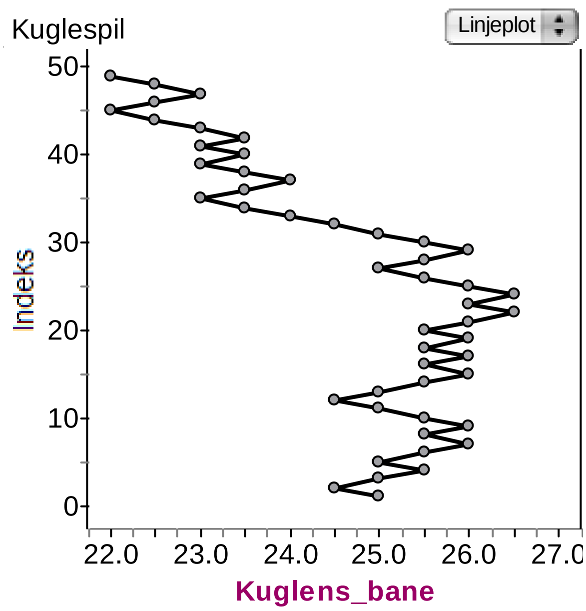
Tilsvarende vil forskellen på antallet af kugler i de to centrale opsamlingsbåse i halvdelen af tilfældene ligge mellem 4 og 17 med en median på 10.

Slut på mellemspil:

Avanceret simulering af den enkelte kugles bane

Som lovet viser vi nu en mere sofistikeret simulering af en kuglebane. Vi opretter da et datasæt **kuglespil** for den enkelte kugles bane. De enkelte opsamlingsbåse nummereres fra 0 til 49. Vi starter da midt mellem de to centrale båse 24 og 25, dvs. startpositionen er 24.5. Hver gang vi rammer en pind falder kuglen tilfældigt til venstre eller højde, dvs. hver gang lægger vi enten ½ til den forrige position eller vi trækker ½ fra den forrige position (i tilfældig rækkefølge). Da vi skal passere 49 pinde opretter vi 49 nye data og indskriver derfor formlen (med semikolon som skilletegn!):

Formlen har altså den følgende betydning: Hver ny værdi af kuglens bane afhænger af den foregående, idet vi på tilfældig vis lægger ±½ til den foregående position. Hvis der ikke er nogen foregående position benytter vi startpositionen 24.5.



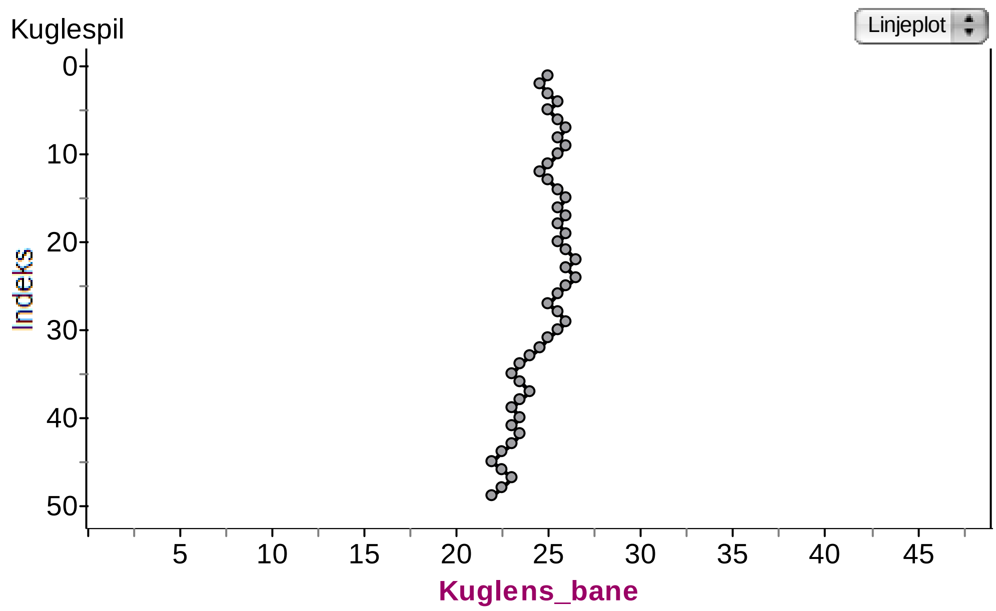
Trækker vi **kuglens_bane** ind på første aksen og vælger et **linjeplot** fås netop en grafisk fremstilling af banen, med den ene fejl at kuglen bevæger sig *opad* andenaksen. Vi kan bedre følge kuglens bane *ned* gennem kuglespillet ved at vende aksen. Det sker ved at åbne for grafinspektøren og sætte **yModsatskala** til **sand** (og samtidigt sætte **xNedre** til 0 og **xØvre** til 50):

Inspicér graf hørende til Kuglespil

Data | **Egenskaber**

Parameter	Værdi
punktstørrelse	5
xNedre	0
xØvre	49
yNedre	-2.5
yØvre	52.5
xModsatskala	falsk
yModsatskala	sand
xAutotilpas	sand
yAutotilpas	sand

Ved at taste **Ctrl-U** (for **simUlering**) kan vi nu spille kuglespil!



Men nu er vi jo faktisk mest interesseret i slutpositionen! Vi indfører derfor en måling af slutpositionen, der er givet ved funktionen **sidste**:

Inspicér Kuglespil

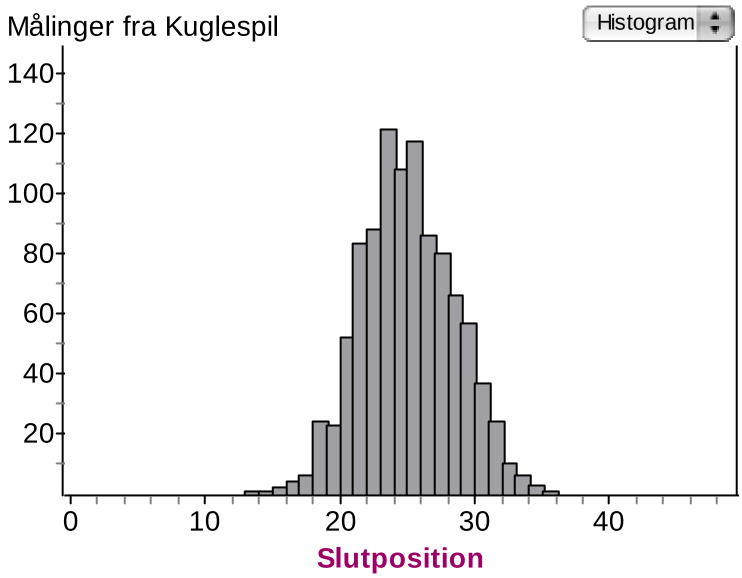
Data | **Målinger** | Kommentarer | Udseende | Kategorier

Måling	Værdi	Formel
Slutposition	22	sidste (Kuglens_bane)
<ny>		

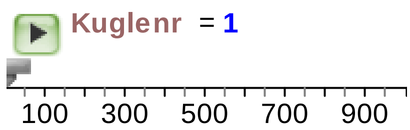
Vi kan da tage **gentagne målinger** af slutpositionen. Faktisk har vi brug for 1000 gentagne målinger for at lade i alt 1000 kugler falde ned gennem kuglespillet! Derefter trækkes **Slutposition** ind i et grafrum for at få oprettet et histogram over slutpositionerne for de 1000 kugler:

Målinger fra Kuglespil

	Slutposition
1	26
2	28
3	24
4	23
5	22
6	24
7	24
8	27
9	22
10	25



Herefter kan vi fortsætte som før. Men når vi nu er i gang med visualiseringerne kan vi også lige se på hvordan kuglerne fyldes op i båsene. Vi indfører da en dynamisk parameter **Kuglenr** med heltallige værdier fra 1 til 1000:



Inspicér parameter

Egenskaber

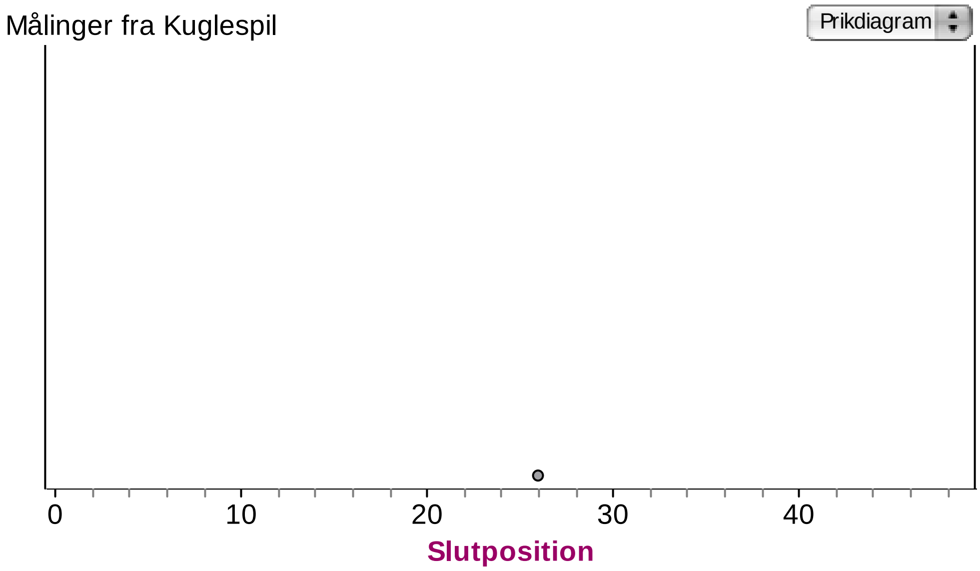
Parameter	Værdi	Formel
Kuglenr	1	
Opdateringsfrekvens		
Slutværdi_	1000	
Startværdi_	1	
BegrænsTilMultiplaAf	1	
VendAksen	falsk	

Derefter sætter vi et **filter** på grafen for **Målinger fra Slutspil**, så vi kun ser kuglerne med numre (indeks) op til **kuglenr**. Det er mest overbevisende, hvis vi vælger et **prikdiagram** og låser grafrummet til at gå fra 0 til 49:

Inspicér graf hørende til Målinger fra Kuglespil

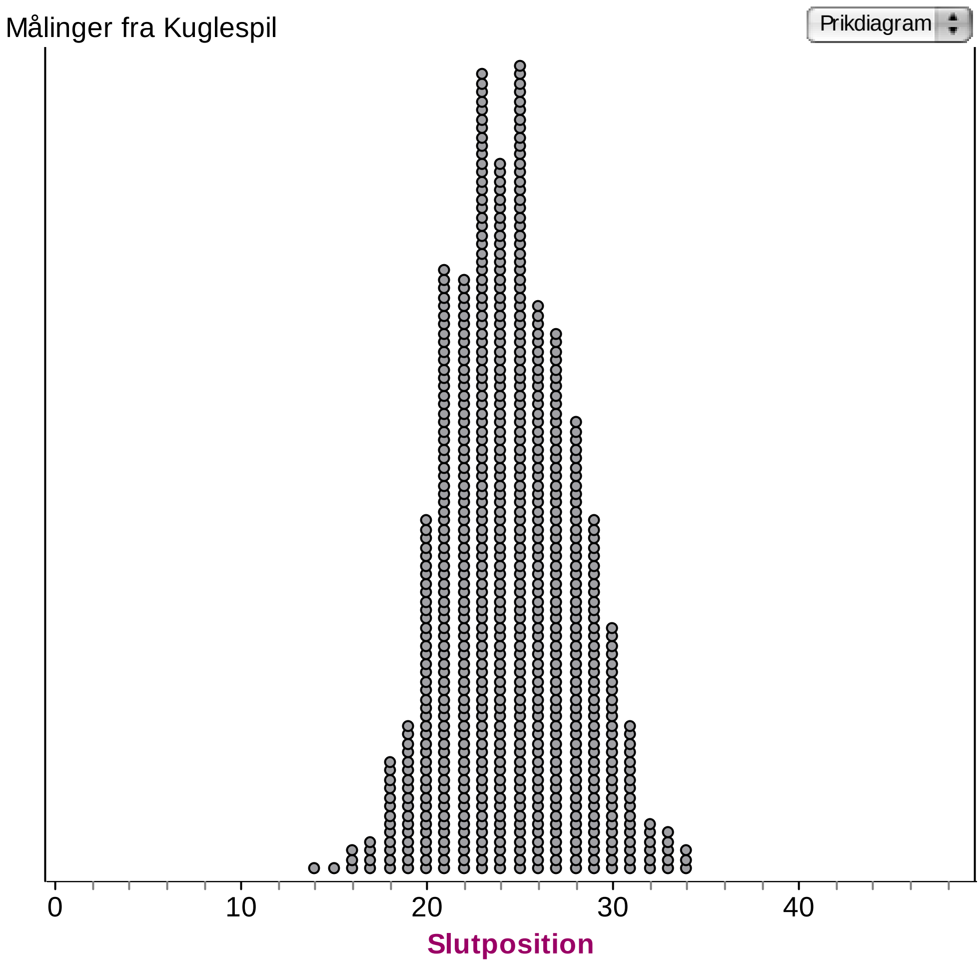
Data Egenskaber

Parameter	Værdi
punktstørrelse	5
xNedre	-0.5
xØvre	49.5
xModsatskala	falsk
xAutotilpas	falsk



indeks ≤ kuglenr

Trækker vi nu i den dynamiske parameter ser vi kuglerne én for én fylde båsene op i kuglespillet:



indeks ≤ kuglenr

Sætter vi den dynamiske parameter til at animere forløbet ved at klikke på trekanten ▶ er det endda som om der er optaget en film, der først spilles forfra og dernæst bagfra!

Udfordring:

Bogen **Tusind kugler** handler om tre unges skæbner: Markus, Åsa og Allan. Markus, Åsa og Allan gik i klasse sammen, da de var børn; Åsa og Markus var kærester, Markus og Allan var tætte venner. Siden udvikledes deres liv sig i vidt forskellige retninger, og deres veje skiltes. Men en dag krydses deres veje igen! I bogen fortæller Åsa Markus om sine overvejelser omkring en opgave hun er ved at skrive om en ungdomsforbryder, som netop viser sig at være Allan.

Åsa er meget fascineret af samspillet mellem orden og tilfældighed, og ikke mindst af hvordan orden opstår spontant ud af tilfældighed: Den enkelte kugle falder tilfældigt ned gennem kuglespillet, men tusinde kugler frembringer en klokkeformet fordeling, normalfordelingen, og der er i praksis meget snævre grænser for hvor meget det enkelte kuglespil kan afvige fra den ideelle normalfordeling. Større afvigelser peger utvetydigt på at nogle har pillet ved spillet og trukket det 'skævt'. Hun er fx så fortrolig med spillet at hun med det samme opdager at hendes lillebror har sat en mønt fast under den ene side af spillet. Men selv når man forsøger at ændre spillereglerne, hvilket hun har eksperimenteret meget med på sin computer, har kuglerne en forunderlig tendens til stadigvæk at samle sig i en normalfordeling om end den flytter og skifter form i forhold til den oprindelige normalfordeling. Normalfordelingen er meget stædig: den bryder kun sammen, hvis man direkte bryder spillereglerne.

Åsa bruger sine erfaringer med kuglespillet til at opdage snyd i den virkelige verden, som fx når journalisterne overdriver de forbrydelser der tilskrives den ukendte kriminelle som Markus ved i virkeligheden er Allan. Hun kan se at mønstrene ikke længere passer sammen og hun får Markus med på at opspore Allan for at finde ud af hvad der i virkeligheden er foregået.

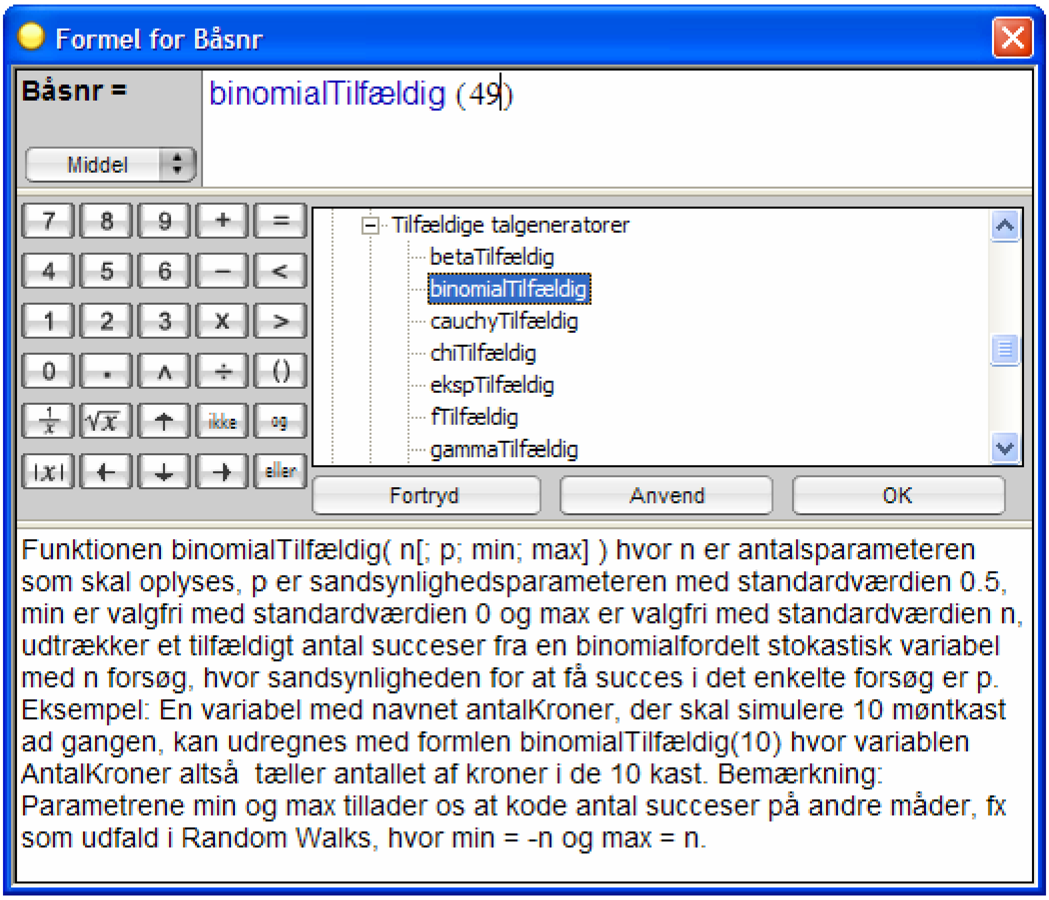
Med udgangspunkt i en sådan tekst kan man stille forskellige opgaver til projekter i sandsynlighedsregning, fx:

Prøv om du kan simulere et skævt kuglespil, hvor sandsynligheden p ikke er den samme for at falde til venstre som til højre. Hvad sker der med formen af slutspillet?

Prøv også om du kan simulere et kuglespil, hvor sandsynlighederne afhænger af, hvor langt ude du er i kuglespillet. Du får da brug for at kunne overføre positionen i kuglespillet til din stokastiske funktion, dvs. parameteren p skal nu også afhænge af positionen.

Afsluttende bemærkning om binomialfordeling

I det foregående har jeg ikke forudsat noget større kendskab til sandsynlighedsregning. Men det er klart at hvis klassen i forvejen har haft et systematisk forløb om binomialfordelinger, så er det i realiteten en symmetrisk binomialfordeling vi simulerer. Vi kan da trække på den indbyggede tilfældighedsgenerator for binomialfordelingen:



Formlen **binomialTilfældig(49)** simulerer altså netop slutpositionen for en kugle i et kuglespil med 49 pinde, idet fordelingen som udgangspunkt er symmetrisk. Læg mærke til at vi også kan simulere en klassisk random Walk gående fra -49 til 49 ved hjælp af kommandoen

binomialTilfældig(49;0.5;-49;49) .

Random Walk med 49 skridt

	Random_Walk
=	binomialTilfældig $(49;\ 0.5;\ -49;\ 49)$
1	-9
2	-5
3	5
4	-1
5	-3
6	9
7	-7
8	1
9	5
10	11