

Calcul 1

Commande :

```
random(2)
```

Résultat :

```
0
```

Calcul 2

Commande :

```
random(2)
```

Résultat :

```
0
```

Calcul 3

Commande :

```
random(2)
```

Résultat :

```
0
```

Calcul 4

Commande :

```
Liste:[]
```

Résultat :

```
[]
```

Calcul 5

Commande :

```
N:30
```

Résultat :

```
30
```

Calcul 6

Commande :

```
for i from 1 thru N do Liste:append(Liste,[random(2)])
```

Calcul 7

Commande :

Liste

Résultat :

```
[1, 0, 1, 1, 1, 0, 1, 1, 1, 0,  
0, 1, 0, 1, 0, 1, 0, 1, 1, 1,  
1, 0, 0, 0, 1, 0, 1, 0, 1, 1]
```

Calcul 8

Commande :

```
entiers:makelist(n,n,1,N)
```

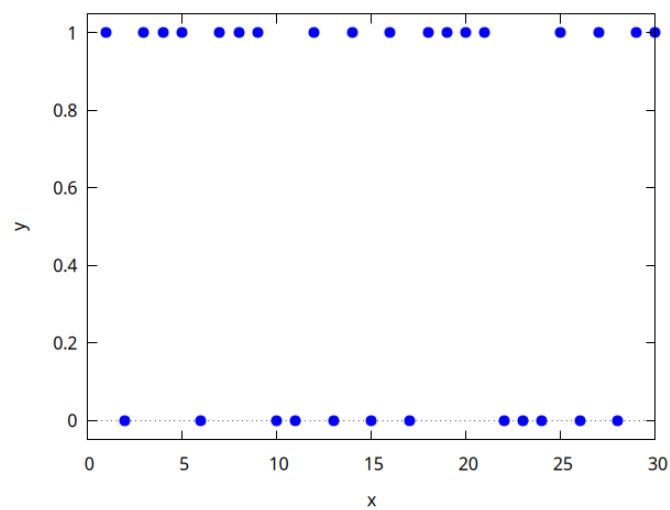
Résultat :

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10,  
11, 12, 13, 14, 15, 16, 17, 18, 19, 20,  
21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
```

Calcul 9

Commande :

```
wxplot2d([discrete, entiers, Liste], [style, points])
```



Calcul 10

Commande :

```
zeroUn: [0, 1]
```

Résultat :

```
[0, 1]
```

Calcul 11

Commande :

```
f1:sum(Liste[k],k, 1, N)/N
```

Résultat :

$$\frac{3}{5}$$

Calcul 12

Commande :

```
f0:1-f1
```

Résultat :

$$\frac{2}{5}$$

Calcul 13

Commande :

```
float(f0)
```

Résultat :

0.4

Calcul 14

Commande :

```
float(f1)
```

Résultat :

0.6

Calcul 15

Commande :

```
ListeFrequences:[f0, f1]
```

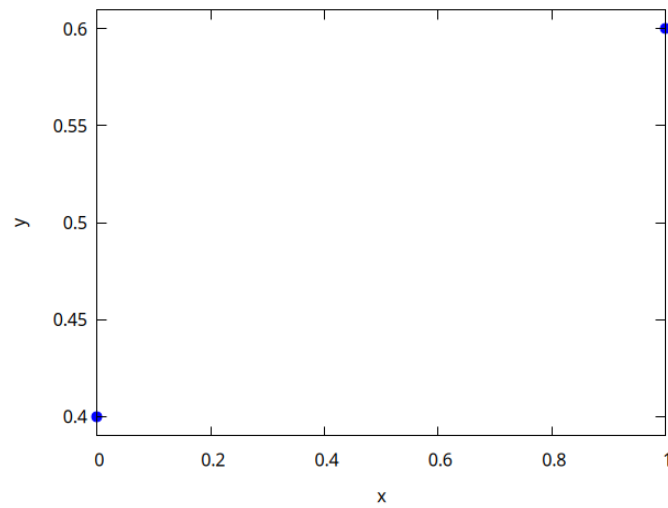
Résultat :

$$\left[\frac{2}{5}, \frac{3}{5} \right]$$

Calcul 16

Commande :

```
wxplot2d([discrete, zeroUn, ListeFrequences], [style, points])
```



Calcul 17

Commande :

```
p:0.3
```

Résultat :

```
0.3
```

Calcul 18

Commande :

```
random(1.0)+ p
```

Résultat :

```
1.152705109085575
```

Calcul 19

Commande :

```
Liste:[]
```

Résultat :

```
[]
```

Calcul 20

Commande :

```
N:30
```

Résultat :

```
30
```

Calcul 21

Commande :

```
for i from 1 thru N do Liste:append(Liste,[floor(random(1.0)+p)])
```

Calcul 22

Commande :

Liste

Résultat :

```
[1, 0, 1, 0, 0, 0, 0, 1, 1, 0,  
1, 0, 1, 0, 0, 0, 1, 1, 0, 1,  
0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
```

Calcul 23

Commande :

```
entiers:makelist(n,n,1,N)
```

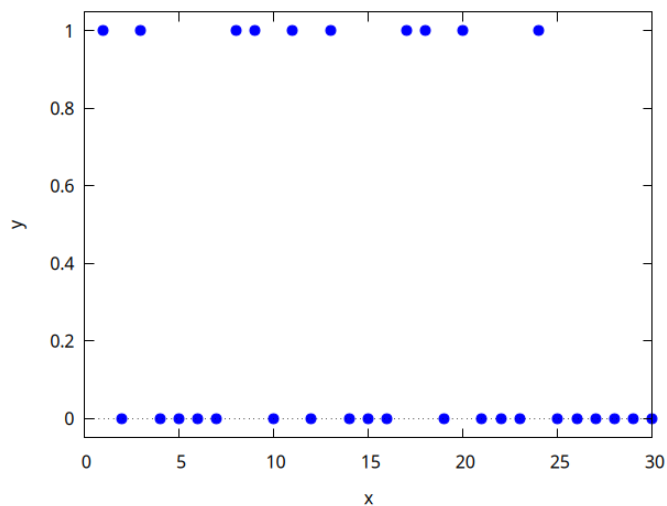
Résultat :

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10,  
11, 12, 13, 14, 15, 16, 17, 18, 19, 20,  
21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
```

Calcul 24

Commande :

```
wxplot2d([discrete, entiers, Liste], [style, points])
```



Calcul 25

Commande :

```
zeroUn: [0, 1]
```

Résultat :

```
[0, 1]
```

Calcul 26

Commande :

```
f1:sum(Liste[k],k, 1, N)/N
```

Résultat :

$$\frac{1}{3}$$

Calcul 27

Commande :

```
f0:1-f1
```

Résultat :

$$\frac{2}{3}$$

Calcul 28

Commande :

```
float(f0)
```

Résultat :

0.6666666666666666

Calcul 29

Commande :

```
float(f1)
```

Résultat :

0.3333333333333333

Calcul 30

Commande :

```
ListeFrequences:[f0, f1]
```

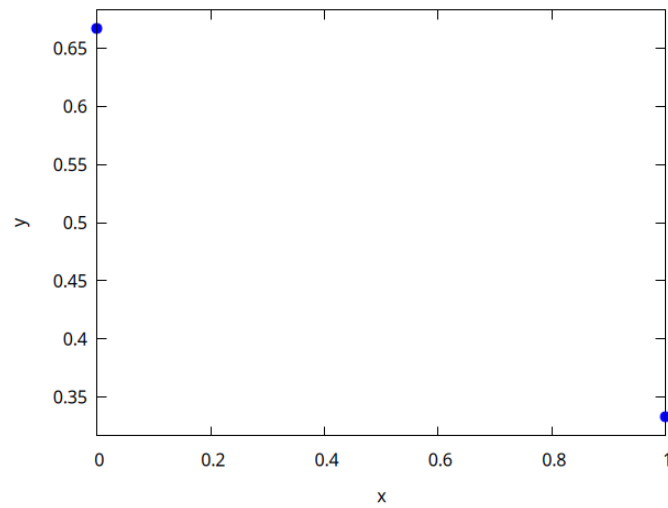
Résultat :

$$\left[\frac{2}{3}, \frac{1}{3} \right]$$

Calcul 31

Commande :

```
wxplot2d([discrete, zeroUn, ListeFrequences], [style, points])
```



Calcul 32

Commande :

```
random(7)
```

Résultat :

0

Calcul 33

Commande :

```
random(7)
```

Résultat :

3

Calcul 34

Commande :

```
random(7)
```

Résultat :

5

Calcul 35

Commande :

```
random(7)
```

Résultat :

1

Calcul 36

Commande :

```
random(7)
```

Résultat :

0

Calcul 37

Commande :

```
random(7)
```

Résultat :

4

Calcul 38

Commande :

```
random(7)
```

Résultat :

3

Calcul 39

Commande :

```
Liste:[]
```

Résultat :

[]

Calcul 40

Commande :

```
N:1000
```

Résultat :

1000

Calcul 41

Commande :

```
for i from 1 thru N do Liste:append(Liste,[random(6)+1])
```

Calcul 42

Commande :

```
Liste
```

Résultat :

[5, 4, 1, 1, 6, 1, 6, 2, 5, 5,
 6, 1, 5, 1, 1, 6, 5, 1, 1, 2,
 6, 4, 4, 4, 2, 4, 3, 2, 6, 3,
 5, 1, 5, 6, 3, 3, 5, 4, 1, 3,
 5, 1, 6, 1, 2, 3, 2, 4, 4, 5,
 4, 1, 1, 4, 6, 4, 4, 6, 6, 4,
 4, 3, 5, 3, 6, 6, 6, 1, 6, 3,
 3, 5, 3, 6, 4, 5, 2, 4, 2, 5,
 1, 5, 6, 5, 3, 5, 3, 6, 2, 3,
 3, 2, 1, 5, 4, 2, 2, 4, 4, 1,
 6, 1, 2, 1, 4, 5, 6, 4, 3, 1,
 3, 6, 1, 5, 1, 5, 2, 3, 2, 6,
 4, 6, 4, 5, 5, 5, 3, 3, 3, 6,
 5, 5, 2, 5, 2, 5, 3, 1, 6, 3,
 3, 5, 1, 5, 2, 5, 1, 2, 6, 4,
 5, 5, 4, 5, 6, 1, 3, 6, 3, 6,
 1, 3, 6, 3, 6, 4, 3, 2, 2, 5,
 6, 1, 6, 2, 1, 5, 3, 3, 3, 1,
 4, 3, 2, 5, 6, 1, 6, 5, 5, 1,
 4, 1, 3, 3, 2, 3, 4, 6, 1, 1,
 6, 6, 2, 6, 2, 2, 4, 4, 1, 3,
 2, 1, 6, 6, 1, 5, 1, 3, 4, 2,
 4, 4, 5, 5, 3, 1, 4, 3, 1, 1,
 6, 4, 6, 6, 5, 6, 5, 6, 6, 5,
 3, 6, 2, 1, 4, 3, 5, 1, 4, 4,
 6, 4, 2, 1, 4, 6, 4, 4, 4, 6,
 4, 4, 1, 1, 1, 5, 2, 3, 5, 2,
 4, 6, 4, 2, 1, 4, 5, 4, 6, 3,
 1, 2, 6, 1, 3, 1, 3, 2, 5, 2,
 3, 3, 4, 5, 1, 3, 4, 5, 1, 6,
 6, 5, 5, 3, 5, 4, 4, 4, 2, 5,
 4, 6, 3, 6, 6, 6, 5, 4, 2, 1,
 2, 2, 5, 6, 3, 5, 5, 4, 5, 2,
 2, 2, 2, 3, 2, 2, 2, 1, 1, 6,
 6, 5, 3, 6, 1, 5, 3, 1, 6, 6,
 2, 5, 1, 5, 3, 4, 2, 1, 2, 3,
 1, 4, 4, 1, 3, 2, 1, 3, 6, 5,
 4, 4, 1, 3, 5, 4, 5, 3, 1, 4,
 4, 2, 4, 5, 4, 6, 4, 5, 1, 1,
 3, 1, 1, 2, 4, 1, 4, 6, 5, 4,
 1, 4, 6, 2, 6, 1, 1, 1, 2, 3,
 2, 6, 6, 5, 6, 6, 4, 1, 4, 6,
 6, 5, 4, 6, 3, 3, 6, 3, 3, 2,
 1, 1, 4, 6, 4, 3, 6, 4, 1, 4,
 4, 2, 6, 1, 3, 5, 2, 2, 1, 1,
 2, 4, 5, 4, 5, 1, 4, 3, 5, 4,
 6, 6, 5, 1, 6, 6, 1, 6, 1, 1,
 4, 1, 6, 6, 5, 5, 6, 2, 1, 2,
 2, 3, 2, 1, 1, 6, 2, 6, 1,
 2, 5, 1, 4, 3, 2, 1, 5, 3, 3,
 2, 2, 4, 4, 4, 2, 3, 4, 6

Calcul 43

Commande :

```
entiers:makelist(n,n,1,N)
```

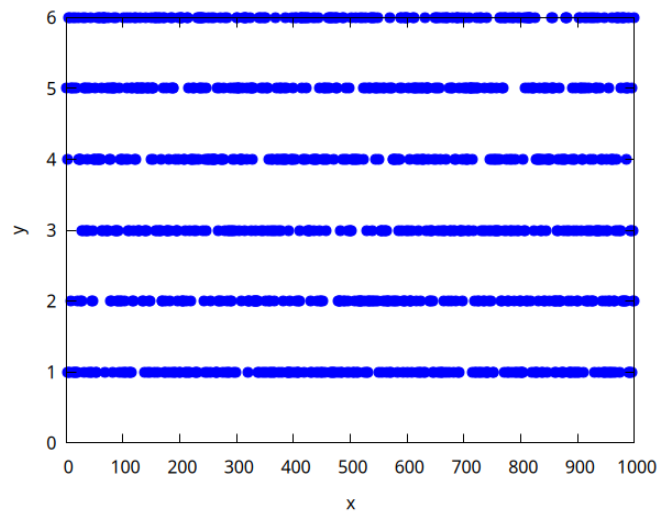
Résultat :

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
61, 62, 63, 64, 65, 66, 67, 68, 69, 70,
71, 72, 73, 74, 75, 76, 77, 78, 79, 80,
81, 82, 83, 84, 85, 86, 87, 88, 89, 90,
91, 92, 93, 94, 95, 96, 97, 98, 99, 100,
101, 102, 103, 104, 105, 106, 107, 108, 109, 110,
111, 112, 113, 114, 115, 116, 117, 118, 119, 120,
121, 122, 123, 124, 125, 126, 127, 128, 129, 130,
131, 132, 133, 134, 135, 136, 137, 138, 139, 140,
141, 142, 143, 144, 145, 146, 147, 148, 149, 150,
151, 152, 153, 154, 155, 156, 157, 158, 159, 160,
161, 162, 163, 164, 165, 166, 167, 168, 169, 170,
171, 172, 173, 174, 175, 176, 177, 178, 179, 180,
181, 182, 183, 184, 185, 186, 187, 188, 189, 190,
191, 192, 193, 194, 195, 196, 197, 198, 199, 200,
201, 202, 203, 204, 205, 206, 207, 208, 209, 210,
211, 212, 213, 214, 215, 216, 217, 218, 219, 220,
221, 222, 223, 224, 225, 226, 227, 228, 229, 230,
231, 232, 233, 234, 235, 236, 237, 238, 239, 240,
241, 242, 243, 244, 245, 246, 247, 248, 249, 250,
251, 252, 253, 254, 255, 256, 257, 258, 259, 260,
261, 262, 263, 264, 265, 266, 267, 268, 269, 270,
271, 272, 273, 274, 275, 276, 277, 278, 279, 280,
281, 282, 283, 284, 285, 286, 287, 288, 289, 290,
291, 292, 293, 294, 295, 296, 297, 298, 299, 300,
301, 302, 303, 304, 305, 306, 307, 308, 309, 310,
311, 312, 313, 314, 315, 316, 317, 318, 319, 320,
321, 322, 323, 324, 325, 326, 327, 328, 329, 330,
331, 332, 333, 334, 335, 336, 337, 338, 339, 340,
341, 342, 343, 344, 345, 346, 347, 348, 349, 350,
351, 352, 353, 354, 355, 356, 357, 358, 359, 360,
361, 362, 363, 364, 365, 366, 367, 368, 369, 370,
371, 372, 373, 374, 375, 376, 377, 378, 379, 380,
381, 382, 383, 384, 385, 386, 387, 388, 389, 390,
391, 392, 393, 394, 395, 396, 397, 398, 399, 400,
401, 402, 403, 404, 405, 406, 407, 408, 409, 410,
411, 412, 413, 414, 415, 416, 417, 418, 419, 420,
421, 422, 423, 424, 425, 426, 427, 428, 429, 430,
431, 432, 433, 434, 435, 436, 437, 438, 439, 440,
441, 442, 443, 444, 445, 446, 447, 448, 449, 450,
451, 452, 453, 454, 455, 456, 457, 458, 459, 460,
461, 462, 463, 464, 465, 466, 467, 468, 469, 470,
471, 472, 473, 474, 475, 476, 477, 478, 479, 480,
481, 482, 483, 484, 485, 486, 487, 488, 489, 490,
491, 492, 493, 494, 495, 496, 497, 498, 499, 500,
501, 502, 503, 504, 505, 506, 507, 508, 509, 510,

Calcul 44

Commande :

```
wxplot2d([discrete, entiers, Liste], [style, points], [y,0,6])
```



Calcul 45

Commande :

```
set:setify(Liste)
```

Résultat :

```
{1,2,3,4,5,6}
```

Calcul 46

Commande :

```
for s in set do print (s)
```

Affichage :

```
1
2
3
4
5
6
```

Calcul 47

Commande :

```
listeFrequences:[]
```

Résultat :

```
[]
```

Calcul 48

Commande :

```
for element in set do (s:0, for i step 1 from 1 thru length(Liste) do (if element=Liste[i] then s:s+1),  
  listeFrequencies:append(listeFrequencies, [s/length(Liste)]))
```

Calcul 49

Commande :

```
float(listeFrequencies)
```

Résultat :

```
[0.185, 0.159, 0.157, 0.165, 0.162, 0.172]
```

Calcul 50

Commande :

```
entiers:makelist(n,n,1,6)
```

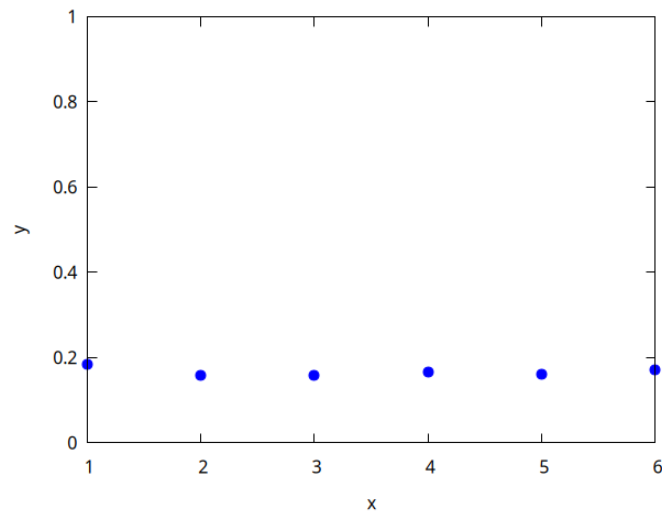
Résultat :

```
[1, 2, 3, 4, 5, 6]
```

Calcul 51

Commande :

```
wxplot2d([discrete, entiers, listeFrequencies], [style, points], [y,0,1])
```



Calcul 52

Commande :

```
nombreSucces(N,p):=block([], somme:0, for i from 1 thru N do (somme:somme+floor(random(1.0)+p)), somme)
```

Résultat :

```
nombreSucces(N,p):=block([],somme:0,  
for i thru N do somme:somme+floor(random(1.0)+p),somme)
```

Calcul 53

Commande :

```
nombreSucces(30,0.7)
```

Résultat :

26

Calcul 54

Commande :

```
Liste:[]
```

Résultat :

[]

Calcul 55

Commande :

```
N:300
```

Résultat :

300

Calcul 56

Commande :

```
p:0.7
```

Résultat :

0.7

Calcul 57

Commande :

```
for i from 1 thru N do Liste:append(Liste,[nombreSucces(N,p)])
```

Calcul 58

Commande :

```
entiers:makelist(n,n,1,N)
```

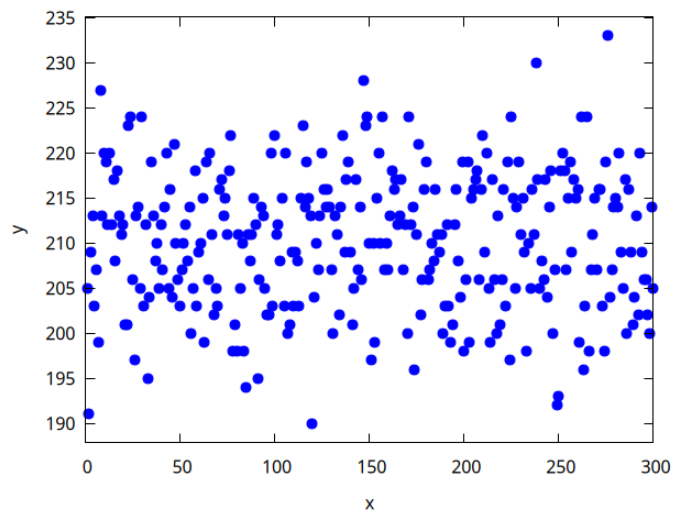
Résultat :

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
61, 62, 63, 64, 65, 66, 67, 68, 69, 70,
71, 72, 73, 74, 75, 76, 77, 78, 79, 80,
81, 82, 83, 84, 85, 86, 87, 88, 89, 90,
91, 92, 93, 94, 95, 96, 97, 98, 99, 100,
101, 102, 103, 104, 105, 106, 107, 108, 109, 110,
111, 112, 113, 114, 115, 116, 117, 118, 119, 120,
121, 122, 123, 124, 125, 126, 127, 128, 129, 130,
131, 132, 133, 134, 135, 136, 137, 138, 139, 140,
141, 142, 143, 144, 145, 146, 147, 148, 149, 150,
151, 152, 153, 154, 155, 156, 157, 158, 159, 160,
161, 162, 163, 164, 165, 166, 167, 168, 169, 170,
171, 172, 173, 174, 175, 176, 177, 178, 179, 180,
181, 182, 183, 184, 185, 186, 187, 188, 189, 190,
191, 192, 193, 194, 195, 196, 197, 198, 199, 200,
201, 202, 203, 204, 205, 206, 207, 208, 209, 210,
211, 212, 213, 214, 215, 216, 217, 218, 219, 220,
221, 222, 223, 224, 225, 226, 227, 228, 229, 230,
231, 232, 233, 234, 235, 236, 237, 238, 239, 240,
241, 242, 243, 244, 245, 246, 247, 248, 249, 250,
251, 252, 253, 254, 255, 256, 257, 258, 259, 260,
261, 262, 263, 264, 265, 266, 267, 268, 269, 270,
271, 272, 273, 274, 275, 276, 277, 278, 279, 280,
281, 282, 283, 284, 285, 286, 287, 288, 289, 290,
291, 292, 293, 294, 295, 296, 297, 298, 299, 300]

Calcul 59

Commande :

```
wxplot2d([discrete, entiers, Liste], [style, points])
```



Calcul 60

Commande :

```
listeFrequences: []
```

Résultat :

```
[]
```

Calcul 61

Commande :

```
listeValeurs: makelist(i,i,0,N)
```

Résultat :

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
 20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
 30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
 40, 41, 42, 43, 44, 45, 46, 47, 48, 49,
 50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
 60, 61, 62, 63, 64, 65, 66, 67, 68, 69,
 70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
 80, 81, 82, 83, 84, 85, 86, 87, 88, 89,
 90, 91, 92, 93, 94, 95, 96, 97, 98, 99,
 100, 101, 102, 103, 104, 105, 106, 107, 108, 109,
 110, 111, 112, 113, 114, 115, 116, 117, 118, 119,
 120, 121, 122, 123, 124, 125, 126, 127, 128, 129,
 130, 131, 132, 133, 134, 135, 136, 137, 138, 139,
 140, 141, 142, 143, 144, 145, 146, 147, 148, 149,
 150, 151, 152, 153, 154, 155, 156, 157, 158, 159,
 160, 161, 162, 163, 164, 165, 166, 167, 168, 169,
 170, 171, 172, 173, 174, 175, 176, 177, 178, 179,
 180, 181, 182, 183, 184, 185, 186, 187, 188, 189,
 190, 191, 192, 193, 194, 195, 196, 197, 198, 199,
 200, 201, 202, 203, 204, 205, 206, 207, 208, 209,
 210, 211, 212, 213, 214, 215, 216, 217, 218, 219,
 220, 221, 222, 223, 224, 225, 226, 227, 228, 229,
 230, 231, 232, 233, 234, 235, 236, 237, 238, 239,
 240, 241, 242, 243, 244, 245, 246, 247, 248, 249,
 250, 251, 252, 253, 254, 255, 256, 257, 258, 259,
 260, 261, 262, 263, 264, 265, 266, 267, 268, 269,
 270, 271, 272, 273, 274, 275, 276, 277, 278, 279,
 280, 281, 282, 283, 284, 285, 286, 287, 288, 289,
 290, 291, 292, 293, 294, 295, 296, 297, 298, 299,
 300]

Calcul 62

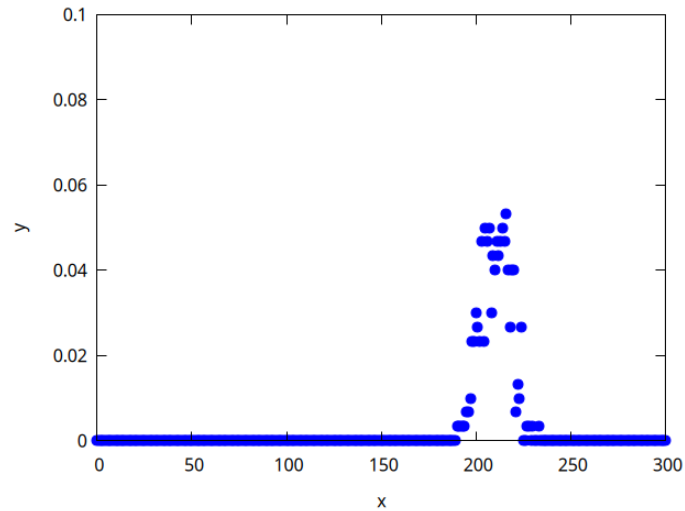
Commande :

```
for element in setify(listeValeurs) do (s:0, for i step 1 from 1 thru length(Liste) do (if element=Liste[i]
  then s:s+1), listeFrequences:append(listeFrequences, [s/length(Liste)]))
```

Calcul 63

Commande :

```
wxplot2d([discrete, listeValeurs, listeFrequences], [style, points], [y,0,0.1])
```



Calcul 64

Commande :

```
declare(N, integer)
```

Résultat :

done

Calcul 65

Commande :

```
assume(N>0)
```

Résultat :

[*redundant*]

Calcul 66

Commande :

```
simulationLoiBinomiale(N,p):=block([], Liste:[], for i from 1 thru N do Liste:append(Liste,[nombreSucces(N,p)])
, listeFrequences:[], listeValeurs:makelist(i,i,0,N), for element in setify(listeValeurs) do (s:0, for i
step 1 from 1 thru length(Liste) do (if element=Liste[i] then s:s+1), listeFrequences:append(
listeFrequences, [s/length(Liste)])), listeFrequences)
```

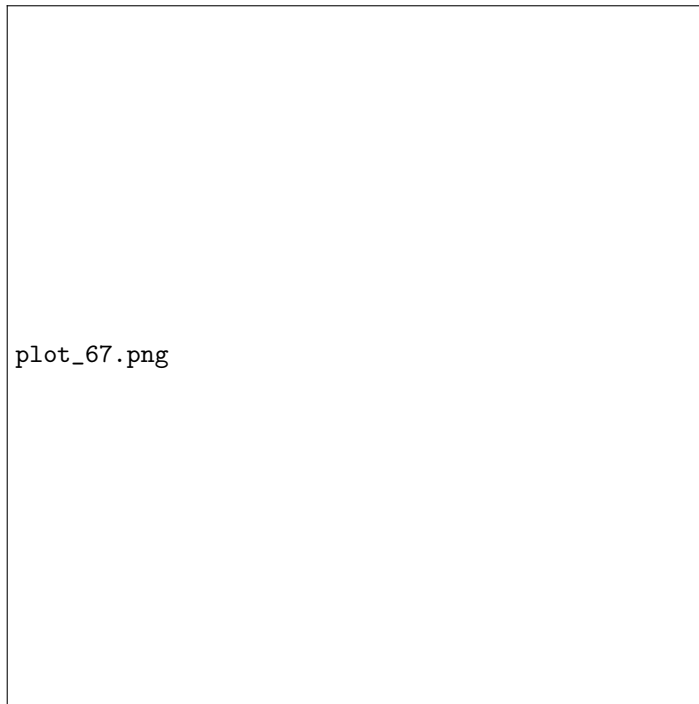
Résultat :

```
simulationLoiBinomiale(N,p):=block([],Liste:[],
for i thru N do
Liste:append(Liste,[nombreSucces(N,p)]),
listeFrequences:[],listeValeurs:makelist(i,i,0,N),
for element in setify(listeValeurs) do
(s:0,
for i thru length(Liste) do
if element = Liste[i] then s:s+1,
listeFrequences
:append(listeFrequences,[s/length(Liste)])),
listeFrequences)
```

Calcul 67

Commande :

```
afficheSimulationLoiBinomiale(N,p):=block([], listeFrequences:simulationLoiBinomiale(N,p), wxplot2d([discrete,  
    makelist(i,i,0,N), listeFrequences], [style, points]))
```



Calcul 68

Commande :

```
afficheSimulationLoiBinomiale(40,0.8)
```

Résultat :

```
[png_file, /tmp/tmpwc3qvxso/plot_67.png ]
```

Calcul 69

Commande :

```
random(1.0)
```

Résultat :

```
0.9565915306933892
```

Calcul 70

Commande :

```
N:100
```

Résultat :

```
100
```

Calcul 71

Commande :

Liste: []

Résultat :

[]

Calcul 72

Commande :

```
for i from 0 thru N do Liste:append(Liste,[random(1.0)])
```

Calcul 73

Commande :

Liste

Résultat :

[0.828985925288555, 0.5394199692841395, 0.9174396575683172, 0.434947961813569, 0.1337394548031956, 0.6199109031765895, 0.145518059882663553, 0.5513819551619559, 0.32907421011590987, 0.16497446191162735, 0.5357217703429666, 0.9987827420870798, 0.2094088740719533, 0.8584669169001706, 0.07363496945585801, 0.5117703297422957, 0.6331760124173405, 0.8060468278269499, 0.15939415894792635, 0.7998066131101857, 0.46696359983748503, 0.6939378638172609, 0.020995123743679844, 0.4197941333396482, 0.888367474553529, 0.5631733174251774, 0.7999789593223028, 0.30804449342967266, 0.1426190095368225, 0.03880055388649639, 0.7255418439374588, 0.8540708477011199, 0.08464118140157817, 0.6071259938228637, 0.5156831619588955, 0.8493439709138926, 0.9409461855987515, 0.6224590118577509, 0.19448001045957186, 0.20255942178616704, 0.34351430594449495, 0.13749583382409503, 0.4179213727981015, 0.9149151812297416, 0.1627976655193919, 0.8325309706275545, 0.3378993354398694, 0.3076555258088567, 0.654526376508893988, 0.4768080204495224, 0.7273199908443835, 0.975339348097342, 0.4642002819700095, 0.8865320027909704, 0.470004377904588927839, 0.9183342640529117, 0.6634540223826384, 0.708712207748802, 0.4557667171151627, 0.41152596195602054, 0.40916530106604143]

Calcul 74

Commande :

```
entiers:makelist(n,n,0,N)
```

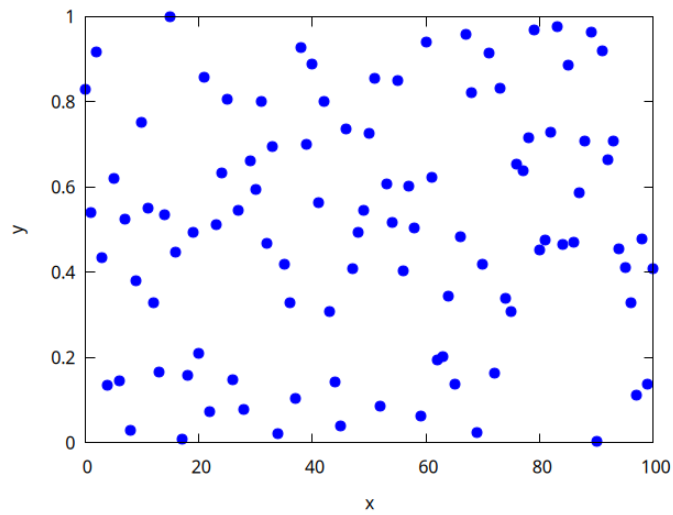
Résultat :

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
40, 41, 42, 43, 44, 45, 46, 47, 48, 49,
50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
60, 61, 62, 63, 64, 65, 66, 67, 68, 69,
70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
80, 81, 82, 83, 84, 85, 86, 87, 88, 89,
90, 91, 92, 93, 94, 95, 96, 97, 98, 99,
100]

Calcul 75

Commande :

```
wxplot2d([discrete, entiers, Liste], [style, points], [y, 0, 1])
```



Calcul 76

Commande :

```
load(distrib)
```

Résultat :

```
/usr/local/share/maxima/5.49.0/share/distrib/distrib.mac
```

Calcul 77

Commande :

```
assume(0<p,p<1)
```

Résultat :

```
[redundant, redundant]
```

Calcul 78

Commande :

```
pdf_bernoulli(1,p)
```

Résultat :

```
0.7
```

Calcul 79

Commande :

```
pdf_bernoulli(0,p)
```

Résultat :

```
0.30000000000000004
```

Calcul 80

Commande :

p:0.3

Résultat :

0.3

Calcul 81

Commande :

zeroUn: [0, 1]

Résultat :

[0, 1]

Calcul 82

Commande :

f1:p

Résultat :

0.3

Calcul 83

Commande :

f0:1-f1

Résultat :

0.7

Calcul 84

Commande :

float(f0)

Résultat :

0.7

Calcul 85

Commande :

float(f1)

Résultat :

0.3

Calcul 86

Commande :

ListeFrequences: [f0, f1]

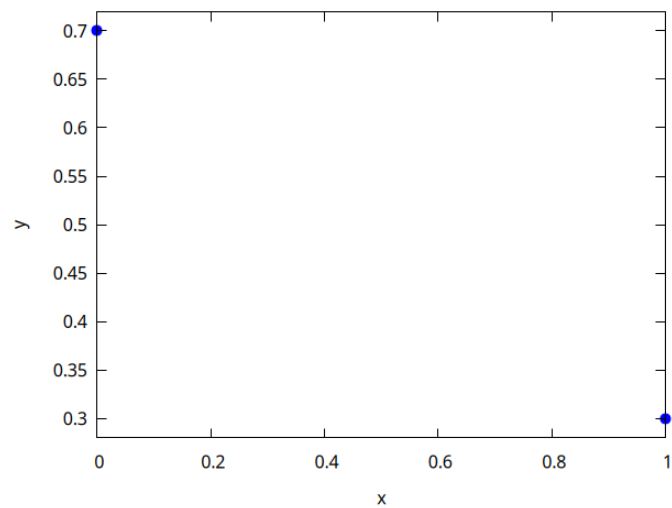
Résultat :

[0.7, 0.3]

Calcul 87

Commande :

```
wxplot2d([discrete, zeroUn, ListeFrequences], [style, points])
```



Calcul 88

Commande :

```
pdf_binomial(5,7,1/6)
```

Résultat :

$$\frac{175}{93312}$$

Calcul 89

Commande :

```
cdf_binomial(5,7,1/6)
```

Résultat :

$$\frac{7775}{7776}$$

Calcul 90

Commande :

```
n:20
```

Résultat :

20

Calcul 91

Commande :

```
p:0.3
```

Résultat :

0.3

Calcul 92

Commande :

a:2

Résultat :

2

Calcul 93

Commande :

b:5

Résultat :

5

Calcul 94

Commande :

`cdf_binomial(b,n,p)-cdf_binomial(a-1,n,p)`

Résultat :

0.4087335696732773

Calcul 95

Commande :

`binomial (5, 2)`

Résultat :

10

Calcul 96

Commande :

N:30

Résultat :

30

Calcul 97

Commande :

`entiers:makelist(n,n,0,N)`

Résultat :

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
30]

Calcul 98

Commande :

```
listeBinomiale:makelist(pdf_binomial(n,N,0.7),n,0,N)
```

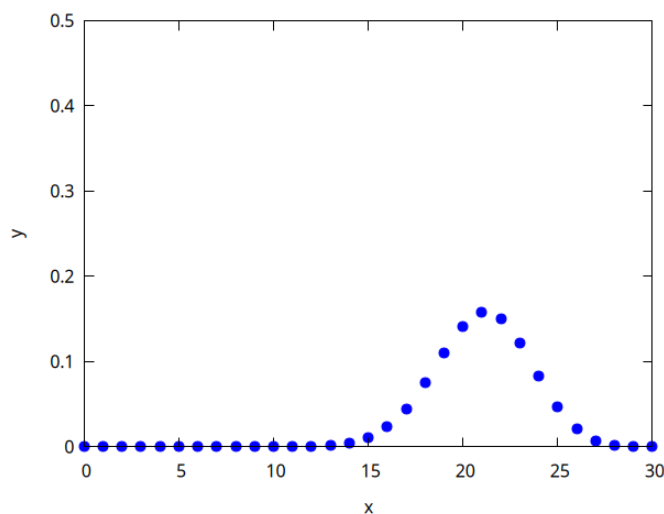
Résultat :

$[2.058911320946499 \times 10^{-16}, 1.441237924662549 \times 10^{-14}, 4.876188311774957 \times 10^{-13}, 1.0619254545643237 \times 10^{-11}, 1.672532590938$
 $2.9592245393542855 \times 10^{-5}, 1.2554285924533328 \times 10^{-4}, 4.6381111887859236 \times 10^{-4}, 0.0014984666917616056, 0.0042456556266578,$
 $0.14156170106200108, 0.15729077895777893, 0.1501411980960617, 0.12185372599100659, 0.0829282301883239, 0.04643980890546137,$
 $2.2539340290692216 \times 10^{-5}]$

Calcul 99

Commande :

```
wxplot2d([discrete, entiers, listeBinomiale], [style, points], [y,0,0.5])
```



Calcul 100

Commande :

```
N:30
```

Résultat :

30

Calcul 101

Commande :

```
entiers:makelist(n,n,0,N)
```

Résultat :

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
30]

Calcul 102

Commande :

```
listeBinomiale:makelist(pdf_binomial(n,N,0.7),n,0,N)
```

Résultat :

$[2.058911320946499 \times 10^{-16}, 1.441237924662549 \times 10^{-14}, 4.876188311774957 \times 10^{-13}, 1.0619254545643237 \times 10^{-11}, 1.672532590938$
 $2.9592245393542855 \times 10^{-5}, 1.2554285924533328 \times 10^{-4}, 4.6381111887859236 \times 10^{-4}, 0.0014984666917616056, 0.0042456556266578,$
 $0.14156170106200108, 0.15729077895777893, 0.1501411980960617, 0.12185372599100659, 0.0829282301883239, 0.04643980890546137,$
 $2.2539340290692216 \times 10^{-5}]$

Calcul 103

Commande :

```
N:30
```

Résultat :

30

Calcul 104

Commande :

```
entiers:makelist(n,n,0,N)
```

Résultat :

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
30]

Calcul 105

Commande :

```
listeBinomiale:makelist(cdf_binomial(n,N,0.7),n,0,N)
```

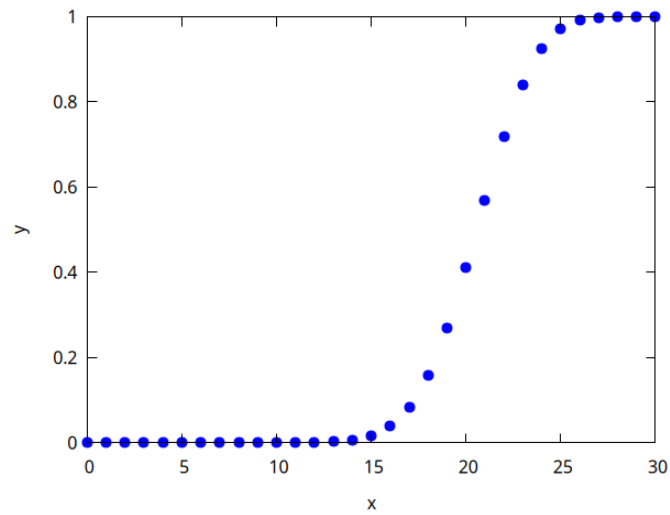
Résultat :

$[2.0589113209465038 \times 10^{-16}, 1.461827037872018 \times 10^{-14}, 5.022371015562152 \times 10^{-13}, 1.1121491647199316 \times 10^{-11}, 1.78374750741$
 $3.6870080768784684 \times 10^{-5}, 1.6241294001411704 \times 10^{-4}, 6.262240588927044 \times 10^{-4}, 0.0021246907506543003, 0.00637034637731208,$
 $0.411191314759272, 0.5684820937170538, 0.7186232918131216, 0.8404770178041268, 0.9234052479924496, 0.9698450568979111, 0.990$
 $1]$

Calcul 106

Commande :

```
wxplot2d([discrete, entiers, listeBinomiale], [style, points], [y,0,1])
```



Calcul 107

Commande :

```
float(pdf_poisson(3,95/100))
```

Résultat :

0.055263680830717164

Calcul 108

Commande :

```
float(cdf_poisson(3,95/100))
```

Résultat :

0.9839255634008381

Calcul 109

Commande :

```
lambda:7.3
```

Résultat :

7.3

Calcul 110

Commande :

```
a:2
```

Résultat :

2

Calcul 111

Commande :

```
b:5
```

Résultat :

5

Calcul 112

Commande :

```
cdf_poisson(b,lambda)-cdf_poisson(a-1,lambda)
```

Résultat :

0.25843623294699997

Calcul 113

Commande :

```
N:30
```

Résultat :

30

Calcul 114

Commande :

```
entiers:makelist(n,n,0,N)
```

Résultat :

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
30]

Calcul 115

Commande :

```
listePoisson:makelist(pdf_poisson(n,21),n,0,N)
```

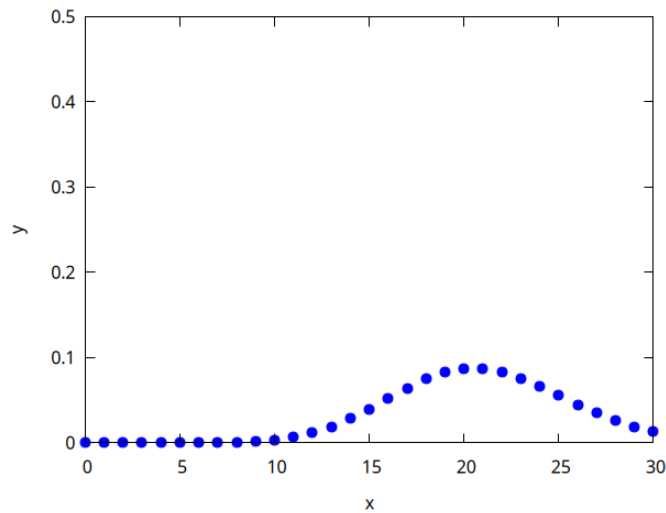
Résultat :

$$\left[e^{-21}, 21 e^{-21}, \frac{441 e^{-21}}{2}, \frac{3087 e^{-21}}{2}, \frac{64827 e^{-21}}{8}, \frac{1361367 e^{-21}}{40}, \frac{9529569 e^{-21}}{80}, \frac{28588707 e^{-21}}{80}, \frac{600362847 e^{-21}}{640}, \frac{1400846643 e^{-21}}{640}, \right. \\ \frac{29417779503 e^{-21}}{6400}, \frac{617773369563 e^{-21}}{70400}, \frac{4324413586941 e^{-21}}{281600}, \frac{90812685325761 e^{-21}}{3660800}, \frac{272438055977283 e^{-21}}{7321600}, \frac{1907066391840981 e^{-21}}{36608000}, \\ \frac{865405750887126927009 e^{-21}}{7567605760000}, \frac{865405750887126927009 e^{-21}}{7567605760000}, \frac{18173520768629665467189 e^{-21}}{166487326720000}, \frac{381643936141222974810969 e^{-21}}{3829208514560000}, \frac{267}{23097785759825920000000} \left. \right]$$

Calcul 116

Commande :

```
wxplot2d([discrete, entiers, listePoisson], [style, points], [y,0,0.5])
```



Calcul 117

Commande :

`N:30`

Résultat :

30

Calcul 118

Commande :

`entiers:makelist(n,n,0,N)`

Résultat :

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
30]

Calcul 119

Commande :

`listePoisson:makelist(cdf_poisson(n,21),n,0,N)`

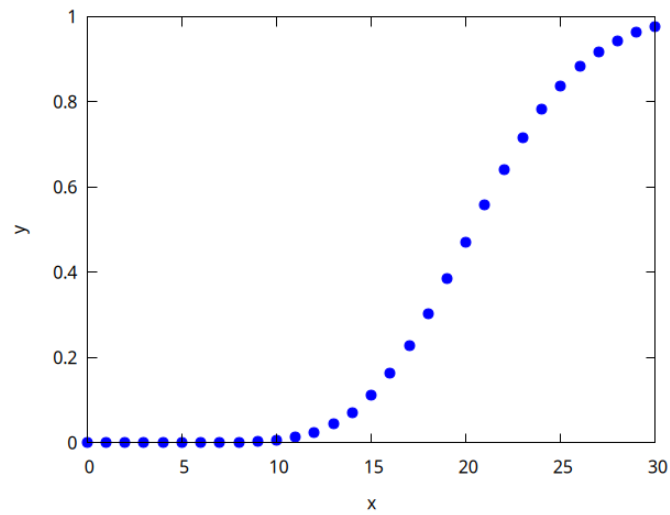
Résultat :

[$Q(1, 21)$, $Q(2, 21)$, $Q(3, 21)$, $Q(4, 21)$, $Q(5, 21)$, $Q(6, 21)$, $Q(7, 21)$, $Q(8, 21)$, $Q(9, 21)$, $Q(10, 21)$,
 $Q(11, 21)$, $Q(12, 21)$, $Q(13, 21)$, $Q(14, 21)$, $Q(15, 21)$, $Q(16, 21)$, $Q(17, 21)$, $Q(18, 21)$, $Q(19, 21)$, $Q(20, 21)$,
 $Q(21, 21)$, $Q(22, 21)$, $Q(23, 21)$, $Q(24, 21)$, $Q(25, 21)$, $Q(26, 21)$, $Q(27, 21)$, $Q(28, 21)$, $Q(29, 21)$, $Q(30, 21)$,
 $Q(31, 21)$]

Calcul 120

Commande :

`wxplot2d([discrete, entiers, listePoisson], [style, points], [y,0,1])`



Calcul 121

Commande :

```
float(pdf_normal(95/100,0,1))
```

Résultat :

0.25405905646918897

Calcul 122

Commande :

```
float(cdf_normal(95/100,0,1))
```

Résultat :

0.828943873691518

Calcul 123

Commande :

```
m:0.95
```

Résultat :

0.95

Calcul 124

Commande :

```
s:1.5
```

Résultat :

1.5

Calcul 125

Commande :

```
a:2
```

Résultat :

2

Calcul 126

Commande :

```
b:5
```

Résultat :

5

Calcul 127

Commande :

```
float(cdf_normal(b,m,s)-cdf_normal(a,m,s))
```

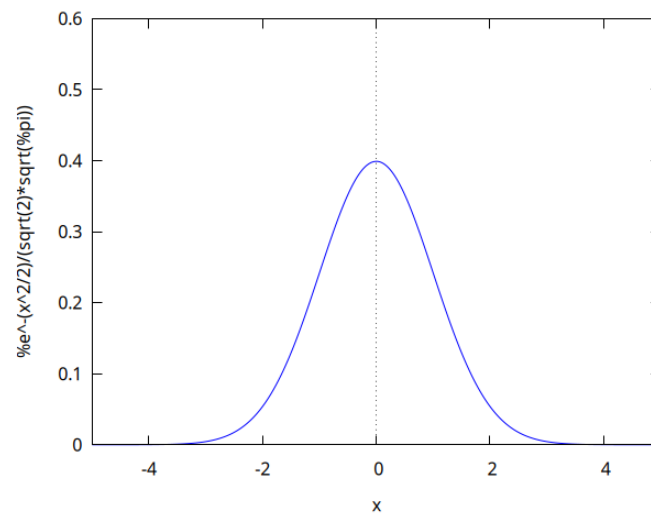
Résultat :

0.23849667842003236

Calcul 128

Commande :

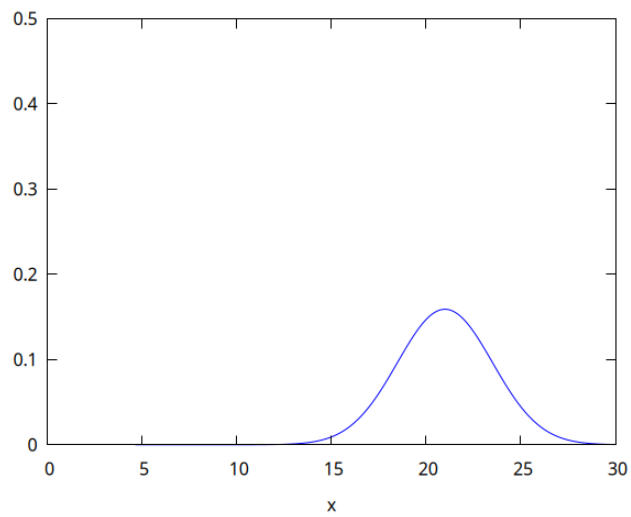
```
wxplot2d(pdf_normal(x,0,1),[x,-5,5], [y,0,0.6])
```



Calcul 129

Commande :

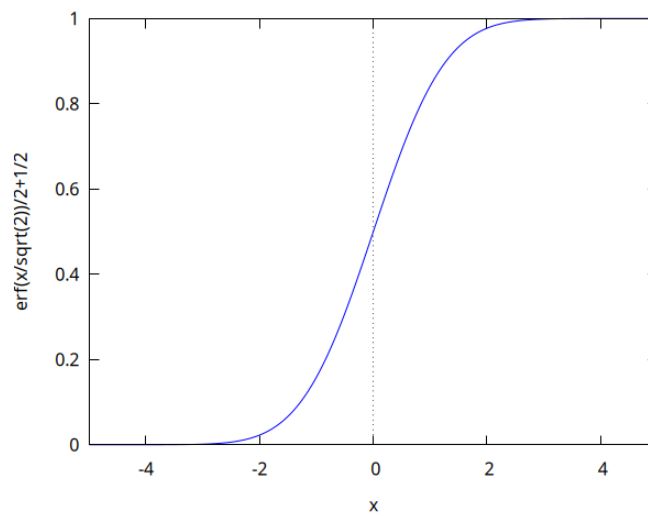
```
wxplot2d(pdf_normal(x,21,sqrt(6.3)),[x,0,30], [y,0,0.5])
```



Calcul 130

Commande :

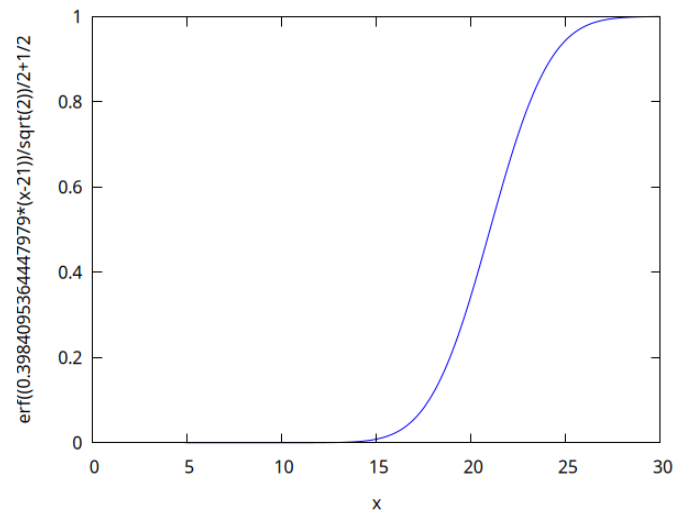
```
wxplot2d(cdf_normal(x,0,1),[x,-5,5], [y,0,1])
```



Calcul 131

Commande :

```
wxplot2d(cdf_normal(x,21,sqrt(6.3)),[x,0,30], [y,0,1])
```



Calcul 132

Commande :

N:30

Résultat :

30

Calcul 133

Commande :

p:0.7

Résultat :

0.7

Calcul 134

Commande :

N*p

Résultat :

21.0

Calcul 135

Commande :

N*(1-p)

Résultat :

9.0000000000000002

Calcul 136

Commande :

```
entiers:makelist(n,n,0,N)
```

Résultat :

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9,  
10, 11, 12, 13, 14, 15, 16, 17, 18, 19,  
20, 21, 22, 23, 24, 25, 26, 27, 28, 29,  
30]
```

Calcul 137

Commande :

```
listeBinomiale:makelist(pdf_binomial(n,N,p),n,0,N)
```

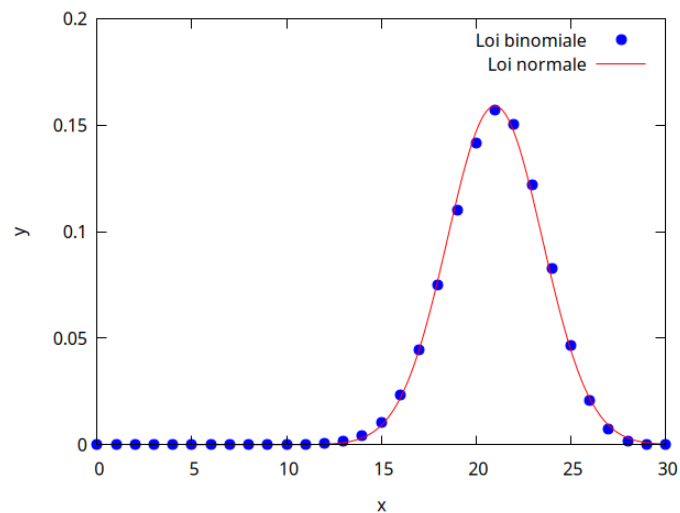
Résultat :

```
[2.058911320946499 × 10-16, 1.441237924662549 × 10-14, 4.876188311774957 × 10-13, 1.0619254545643237 × 10-11, 1.672532590938  
2.9592245393542855 × 10-5, 1.2554285924533328 × 10-4, 4.6381111887859236 × 10-4, 0.0014984666917616056, 0.0042456556266578,  
0.14156170106200108, 0.15729077895777893, 0.1501411980960617, 0.12185372599100659, 0.0829282301883239, 0.04643980890546137,  
2.2539340290692216 × 10-5]
```

Calcul 138

Commande :

```
wxplot2d([[discrete, entiers, listeBinomiale], pdf_normal(x,N*p,sqrt(N*p*(1-p)))], [x,0,N], [style, points,  
lines], [y,0,0.2], [legend, "Loi binomiale", "Loi normale"], [xlabel, "x"], [ylabel, "y"])
```



Calcul 139

Commande :

```
N:40
```

Résultat :

```
40
```

Calcul 140

Commande :

p:0.09

Résultat :

0.09

Calcul 141

Commande :

N*p*(1-p)

Résultat :

3.276

Calcul 142

Commande :

entiers:makelist(n,n,0,N)

Résultat :

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
40]

Calcul 143

Commande :

listeBinomiale:makelist(pdf_binomial(n,N,p),n,0,N)

Résultat :

[0.022996179653995415, 0.09097389753228954, 0.17544965952655842, 0.21979407896733688, 0.20107535246187686, 0.14318332790691
0.0017453752947897993, $4.7078054904420156 \times 10^{-4}$, $1.1252172463419105 \times 10^{-4}$, $2.3969124774147198 \times 10^{-5}$, $4.571818932588516 \times 10^{-6}$,
 $2.5414049390488933 \times 10^{-11}$, $2.3937880116630084 \times 10^{-12}$, $2.044644105866006 \times 10^{-13}$, $1.582572122074978 \times 10^{-14}$, $1.108670030574 \times 10^{-15}$,
 $1.399301071677631 \times 10^{-23}$, $4.464271409109775 \times 10^{-25}$, $1.2417787916823207 \times 10^{-26}$, $2.9772918082293404 \times 10^{-28}$, $6.062358885534 \times 10^{-30}$,
 $1.478088294143457 \times 10^{-42}$]

Calcul 144

Commande :

listePoisson:makelist(pdf_poisson(n,N*p),n,0,N)

Résultat :

[0.02732372244729257, 0.09836540081025325, 0.1770577214584558, 0.21246926575014694, 0.19122233917513223, 0.1376800842060952
0.002752972289451246, $9.009727492749533 \times 10^{-4}$, $2.7029182478248594 \times 10^{-4}$, $7.485004378591919 \times 10^{-5}$, $1.9247154116379215 \times 10^{-5}$,
 $1.501292322368562 \times 10^{-9}$, $2.573643981203249 \times 10^{-10}$, $4.211417423787134 \times 10^{-11}$, $6.591783793753775 \times 10^{-12}$, $9.88767569063066 \times 10^{-13}$,
 $5.0344822410895945 \times 10^{-18}$, $5.846495505781464 \times 10^{-19}$, $6.577307444004147 \times 10^{-20}$, $7.175244484368159 \times 10^{-21}$, $7.5973176893309 \times 10^{-22}$,
 $5.984044682014297 \times 10^{-28}$]

Calcul 145

Commande :

```
wxplot2d([[discrete, entiers, listeBinomiale], [discrete, entiers, listePoisson]], [x,0,N], [style, points, points], [y,0,0.25], [legend, "Loi binomiale", "Loi de Poisson"], [xlabel, "x"], [ylabel, "y"])
```

