

Random

Random

Ομοιόμορφη κατανομή

- `import numpy as np`
`x=np.random.uniform(-1,1)`
- Τυχαίος αριθμός ανάμεσα στο $(-1,1)$
- ίση πιθανότητα να είναι οποιοσδήποτε στο $(-1,1)$

Random

Ομοιόμορφη κατανομή

- ```
import numpy as np
N=500
x=range(N)
y=[np.random.uniform(-1,1) for i in x]
```

N τυχαίοι αριθμοί ανάμεσα στο (-1,1)

# Random

Ομοιόμορφη κατανομή - εκτύπωση

```
import matplotlib.pyplot as p
```

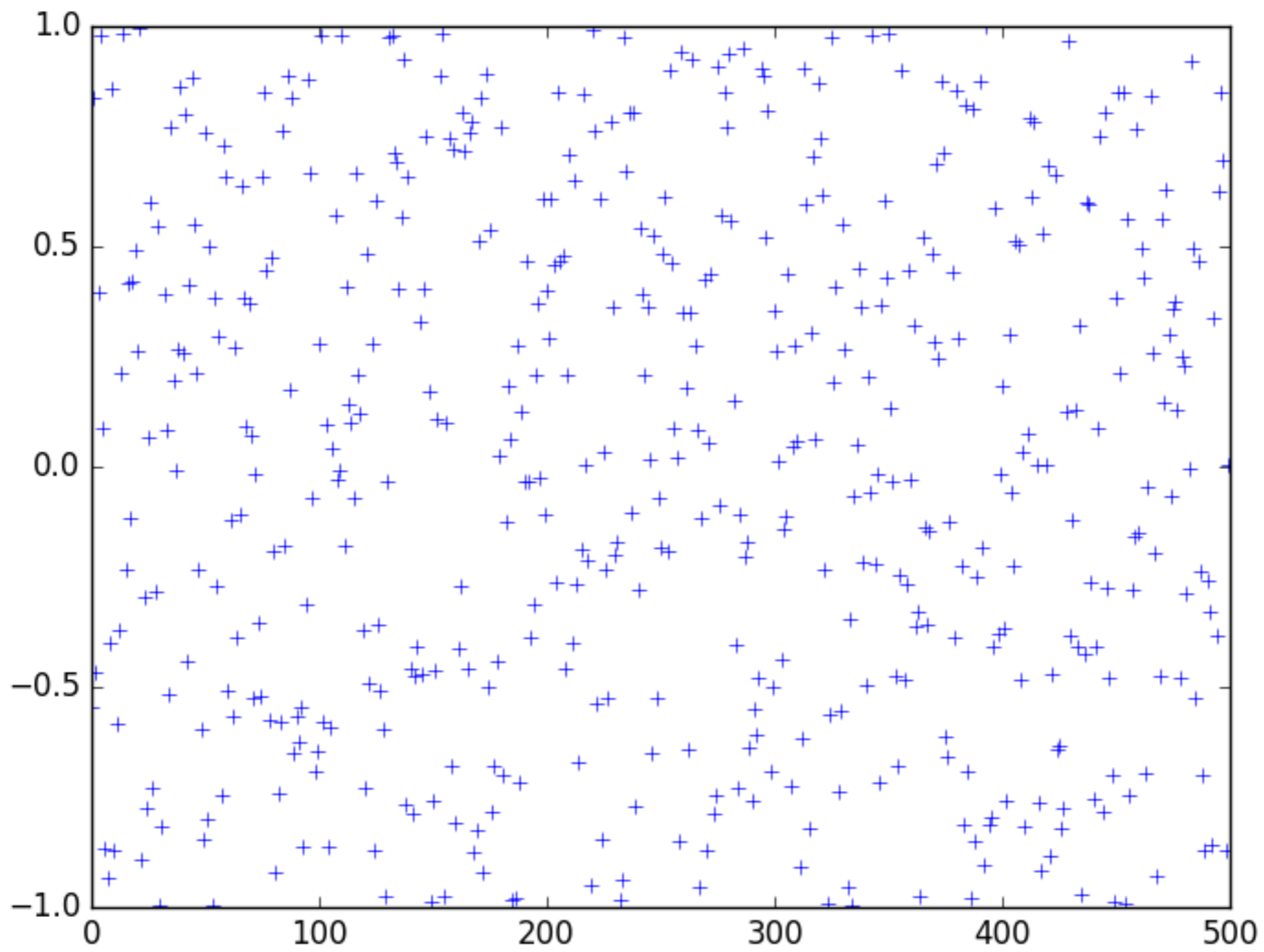
```
p.figure()
```

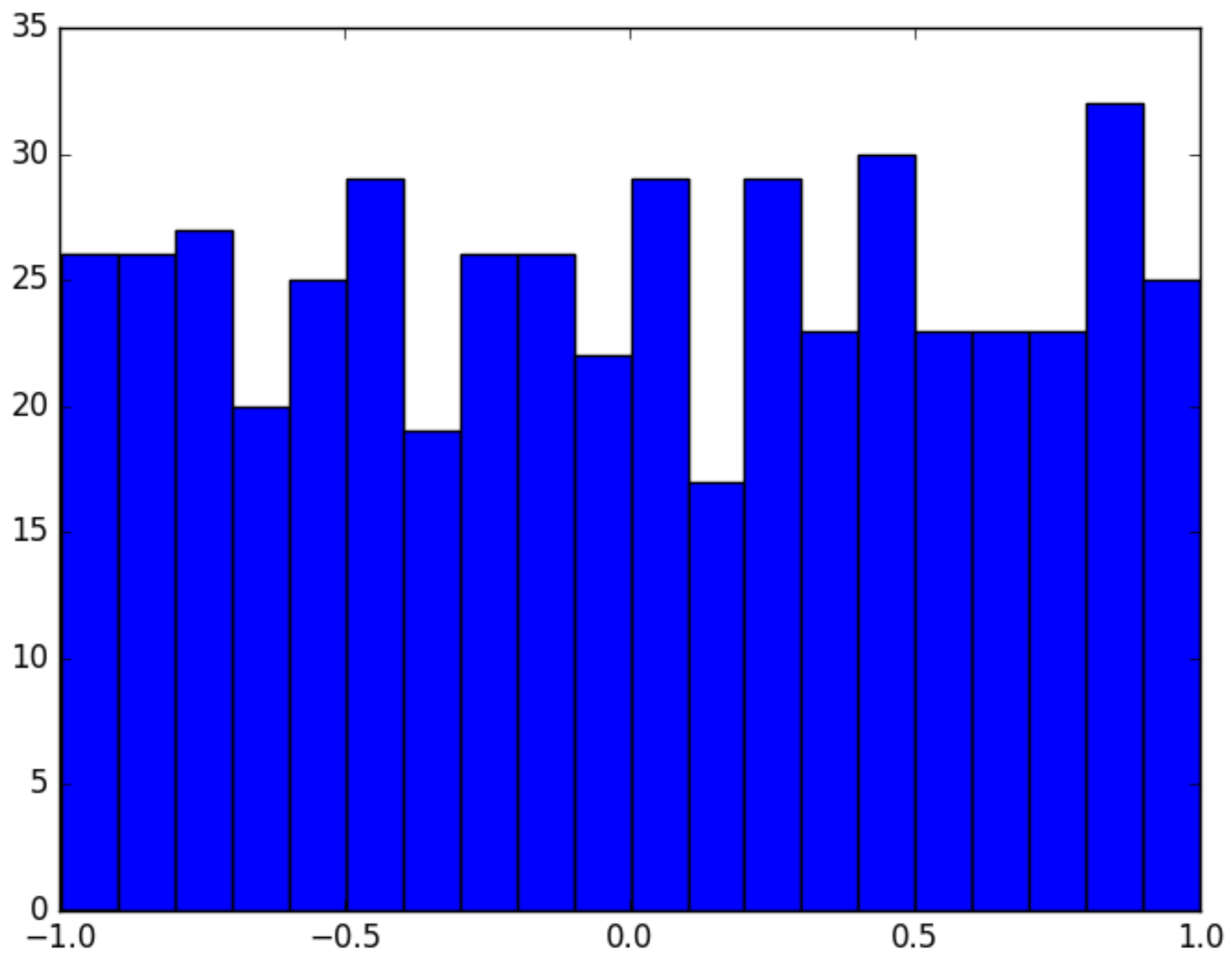
```
p.plot(x,y, '+')
```

```
p.figure()
```

```
p.hist(y,bins=20) #histogram
```

```
p.show()
```





# Random

## Κατανομή Gauss

- ```
import numpy as np
m=0 #mean
s=1 # std.dev
N=1000
x=range(N)
y=np.random.normal(m,s,size=N)
```

N τυχαίοι αριθμοί σύμφωνα με την κατανομή Gauss

Random

Κατανομή Gauss - εκτύπωση

```
import matplotlib.pyplot as p
```

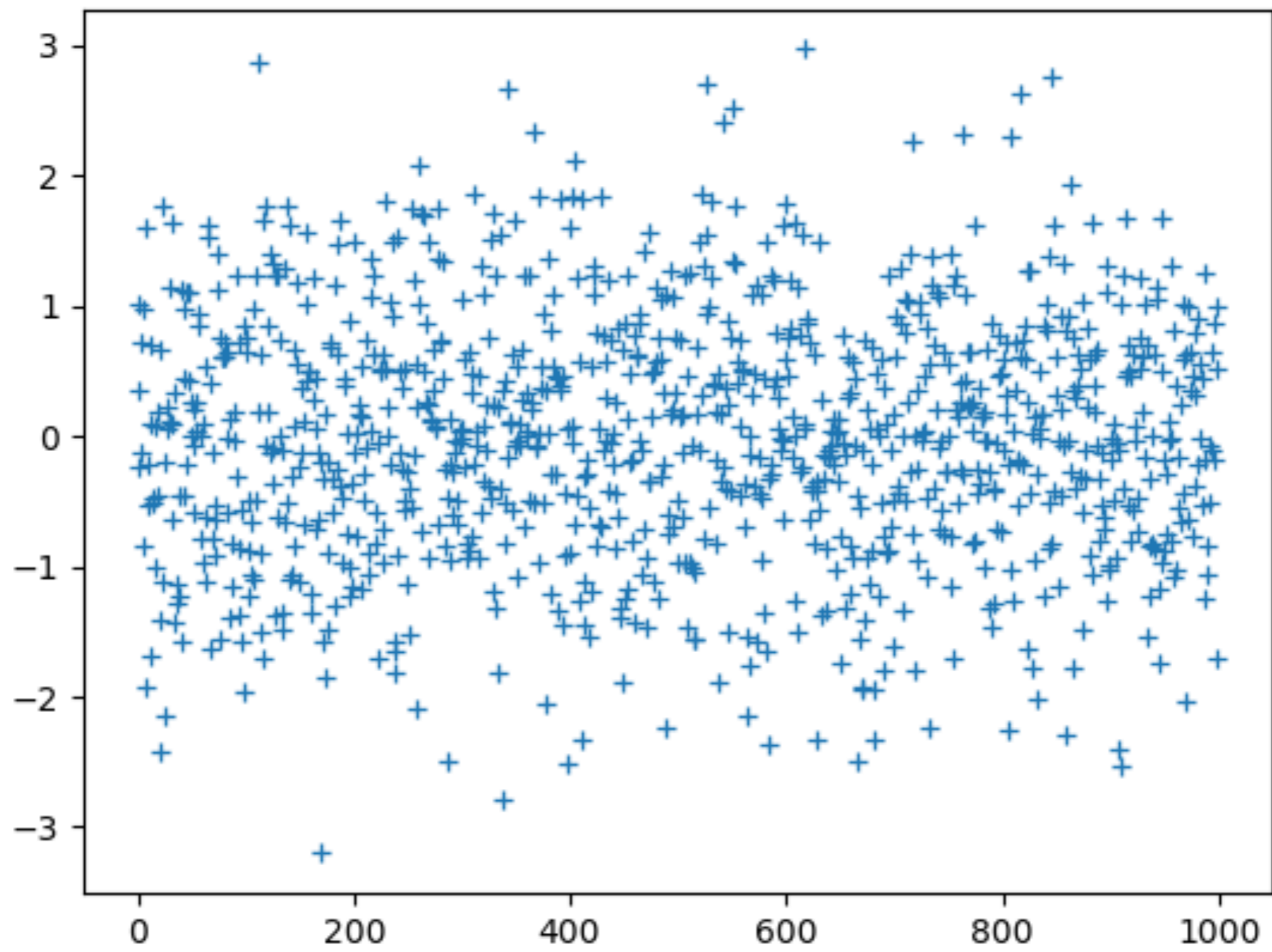
```
p.figure()
```

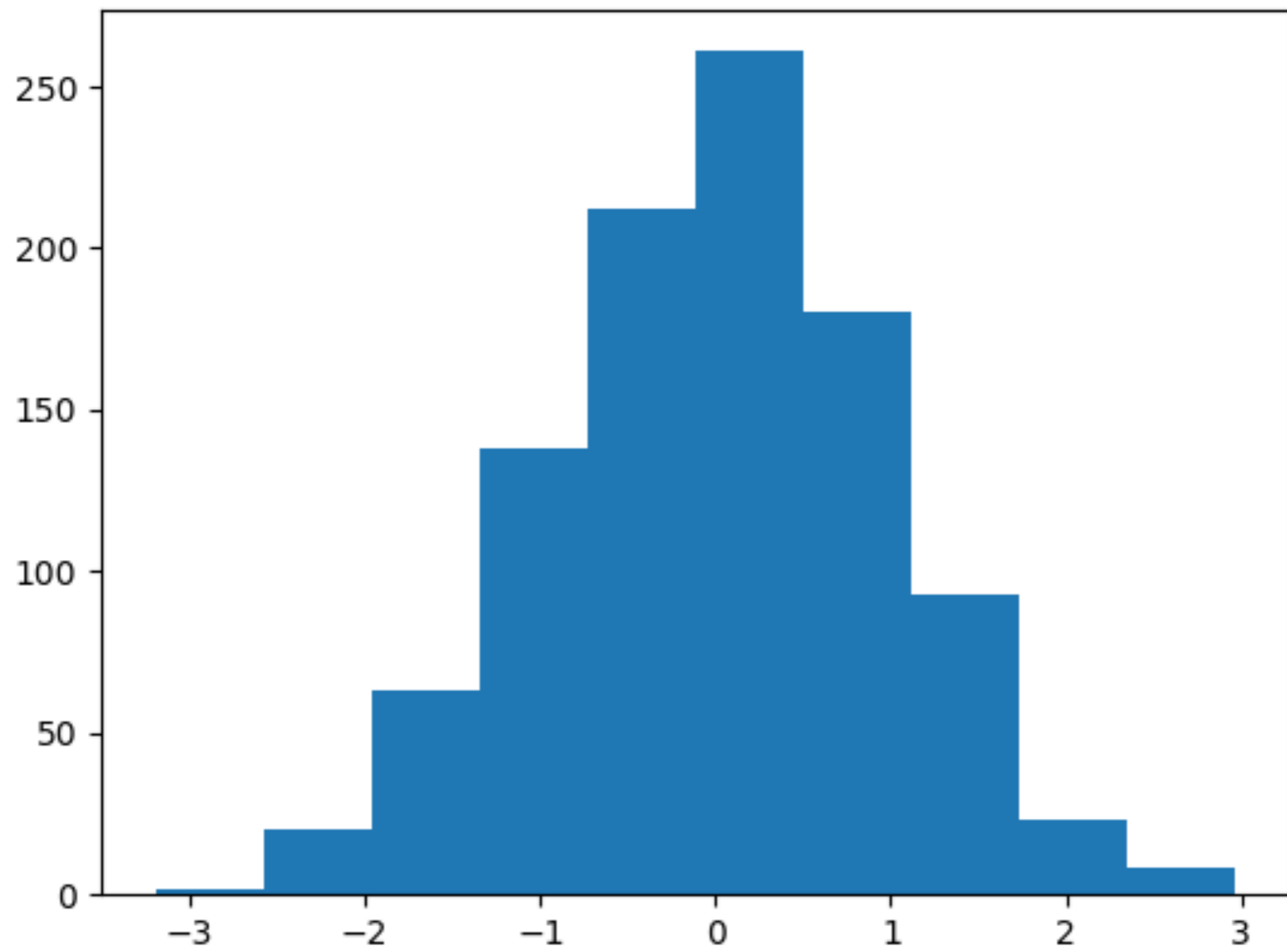
```
p.plot(x,y, '+' )
```

```
p.figure()
```

```
p.hist(y,bins=20) #histogram
```

```
p.show()
```



Random

- Η βιβλιοθήκη μπορεί να “φορτωθεί” και ανεξάρτητα από την `numpy` ως `random` (υπάρχουν μικροδιαφορές με τη `random` της `numpy`)
- `import random # εισαγωγή της βιβλιοθήκης`
- Τυχαία επιλογή ακεραίων αριθμών - ίση πιθανότητα
- `np.random.randint(a,b) #ακέραιοι ανάμεσα στο a και b (χωρίς το b)`
- Ζάρι έχει 6 πλευρές. Η πιθανότητα να έρθει 1 έως 6 είναι ίση

Random

Δύο παίκτες ρίχνουν από ένα ζάρι. Αυτός που έχει το μεγαλύτερο κερδίζει.

Θέλουμε να μετρήσουμε τη πιθανότητα να κερδίσει ο παίκτης με το μαύρο ζάρι.

Πραγματοποιούμε το πείραμα στον Η/Υ καλώντας τη συνάρτηση `randint(1,7)` κάθε φορά που ρίχνουν τα ζάρια, προσομοιώνοντας τη ρίψη του ζαριού

Random

```
import numpy as np
N=100 ← Πόσες φορές θα ρίξουμε τα ζάρια
M=0 ← Μετρητής πόσες φορές το μαύρο είναι μεγαλύτερο
for i in range(N):
    black=np.random.randint(1,7)
    green=np.random.randint(1,7)
    if black>green:
        M+=1
p=M/N ← προσεγγιστική πιθανότητα
print('Approximate probability of /
      Black dice> Green dice /
      after %d throws: %4.2f'%(N,p))
```

Random

- Για να υπολογίσουμε την ακριβή πιθανότητα πρέπει να θεωρήσουμε όλα τα δυνατά ενδεχόμενα και στη συνέχεια να μετρήσουμε πότε το μαύρο ζάρι είναι μεγαλύτερο.

- | | | | |
|----------|----------|---------|----------|
| $(1, 1)$ | $(1, 2)$ | \dots | $(1, 6)$ |
| $(2, 1)$ | $(2, 2)$ | \dots | $(2, 6)$ |
| \vdots | \vdots | \dots | \vdots |
| $(6, 1)$ | $(6, 2)$ | \dots | $(6, 6)$ |

Random

- Χρησιμοποιούμε `list comprehension` για να δημιουργήσουμε μια λίστα με τα ζευγάρια
- `combinations=[(black,green) \`
- `for black in range(1,7) for green in range(1,7)]`
- επιστρέφει τη λίστα
- `[(1,1),(1,2),..., (6,5), (6,6)]`

Random

Θέλουμε να μετρήσουμε πόσα είναι τα “επιθυμητά” ζευγάρια. Δημιουργούμε μια νέα list με τιμές True ή False, ανάλογα αν το αντίστοιχο ζευγάρι είναι επιθυμητό ή όχι.

```
success=[black>green for (black,green) in combinations]
```

επιστρέφει

```
[False,..., True, False,..., True, True, False]
```


Random

- `True` αντιστοιχεί στον αριθμό 1
- `False` αντιστοιχεί στον αριθμό 0
- `True+True` είναι ίσο με 2
- `False+True` είναι ίσο με 1
- `sum(success)` αθροίζει τα στοιχεία της λίστας `success`. Εδώ θα επιστρέψει τον αριθμό των στοιχείων `True`

Random

Υπολογισμός πιθανότητας (μετρώντας τα ενδεχόμενα)

```
combinations=[(black,green) \
               for black in range(1,7) \
               for green in range(1,7)]
success=[black>green \
         for (black,green) in combinations]
M=sum(success)
p=M/len(combinations)
print('True probability of \
      Black dice> Green dice: %4.2f'%p)
```

Random

- Παιχνίδι με ζάρια
- Δύο παίκτες ρίχνουν από ένα ζάρι και κάθε φορά ποντάρουν από ένα ευρώ. Αν το ζάρι ενός παίκτη είναι μεγαλύτερο από του άλλου παίρνει 2 ευρώ.
- Θέλουμε να υπολογίσουμε το κερδος ενός παίκτη αν παίξει N φορές.

Random

```
import numpy
N=input('How many throws: ')
start_capital=input('How much money initially: ')
money=start_capital
for i in range(N):
    money-=1
    black=np.random.randint(1,7)
    green=np.random.randint(1,7)
    if black>green:
        money+=2
profit_total=money-start_capital

print('After %d throws you win %5d money'%
      (N,profit_total))
```