

Método de los cuadrados medios

Estudiante: Adriana Castillo

Generar los números randómicos

Ejemplo 1

```
In [1]: import pandas as pd, math

iteraciones = int(input ("Iteraciones: "))
semilla = int(input ("Semilla: "))
digitos = int(input ("Digitos: "))

val_xn=[]
val_xn_m=[]
val_ui=[]
val_rn=[]
div=[int(str(num).ljust(digitos+1, "0")) for num in [1]]
ui=""
izq=int(digitos/2)
der=digitos-izq

for i in range(iteraciones):
    if(len(str(semilla))>=3):
        if(len(str(semilla*semilla))>digitos):
            val_xn.append(semilla)
            val_xn_m.append(semilla*semilla)
            numero = str(semilla*semilla)
            mitad=math.floor(len(numero)/2)
            if(int(numero)>0):
                for p in range(mitad-izq, mitad+der, 1): ui+=numero[p]
            val_ui.append(ui)
            val_rn.append(int(ui)/div[0])
            semilla=int(val_ui[-1])
            ui=''
        else:
            print('\n* Número de digitos es muy alto')
            break
    else:
        print('\n* * La semilla debe ser mayor a 0 y de 3 digitos')
        break

tabla=pd.DataFrame({"XN ":val_xn, "XN * XN ": val_xn_m,"UI ":val_ui,"RN ":val_rn})
tabla

Iteraciones: 15
Semilla: 7543
Digitos: 4
```

Out[1]:

	XN	XN * XN	UI	RN
0	7543	56896849	8968	0.8968
1	8968	80425024	4250	0.4250
2	4250	18062500	0625	0.0625
3	625	390625	9062	0.9062
4	9062	82119844	1198	0.1198
5	1198	1435204	4352	0.4352
6	4352	18939904	9399	0.9399
7	9399	88341201	3412	0.3412
8	3412	11641744	6417	0.6417
9	6417	41177889	1778	0.1778
10	1778	3161284	1612	0.1612
11	1612	2598544	5985	0.5985
12	5985	35820225	8202	0.8202
13	8202	67272804	2728	0.2728
14	2728	7441984	4419	0.4419

Ejemplo 2

```
In [4]: import pandas as pd, math

iteraciones = int(input ("Iteraciones: "))
semilla = int(input ("Semilla: "))
digitos = int(input ("Digitos: "))

val_xn=[]
val_xn_m=[]
val_ui=[]
val_rn=[]
div=[int(str(num).ljust(digitos+1, "0")) for num in [1]]
ui=""
izq=int(digitos/2)
der=digitos-izq

for i in range(iteraciones):
    if(len(str(semilla))>=3):
        if(len(str(semilla*semilla))>digitos):
            val_xn.append(semilla)
            val_xn_m.append(semilla*semilla)
            numero = str(semilla*semilla)
            mitad=math.floor(len(numero)/2)
            if(int(numero)>0):
                for p in range(mitad-izq, mitad+der, 1): ui+=numero[p]
            val_ui.append(ui)
            val_rn.append(int(ui)/div[0])
            semilla=int(val_ui[-1])
            ui=''
        else:
            print('\n* Número de digitos es muy alto')
            break
    else:
        print('\n* La semilla debe ser mayor a 0 y de 3 digitos')
        break

tabla=pd.DataFrame({"XN ":val_xn, "XN * XN ": val_xn_m,"UI ":val_ui,"RN ":val_rn})
tabla

Iteraciones: 26
Semilla: 52315
Digitos: 6
```

Out[4]:

	XN	XN * XN	UI	RN
0	52315	2736859225	368592	0.368592
1	368592	135860062464	860062	0.860062
2	860062	739706643844	706643	0.706643
3	706643	499344329449	344329	0.344329
4	344329	118562460241	562460	0.562460
5	562460	316361251600	361251	0.361251
6	361251	130502285001	502285	0.502285
7	502285	252290221225	290221	0.290221
8	290221	84228228841	228228	0.228228
9	228228	52088019984	088019	0.088019
10	88019	7747344361	473443	0.473443
11	473443	224148274249	148274	0.148274
12	148274	21985179076	985179	0.985179
13	985179	970577662041	577662	0.577662
14	577662	333693386244	693386	0.693386
15	693386	480784144996	784144	0.784144
16	784144	614881812736	881812	0.881812
17	881812	777592403344	592403	0.592403
18	592403	350941314409	941314	0.941314
19	941314	886072046596	072046	0.072046
20	72046	5190626116	906261	0.906261
21	906261	821309000121	309000	0.309000
22	309000	95481000000	481000	0.481000
23	481000	231361000000	361000	0.361000
24	361000	130321000000	321000	0.321000
25	321000	103041000000	041000	0.041000