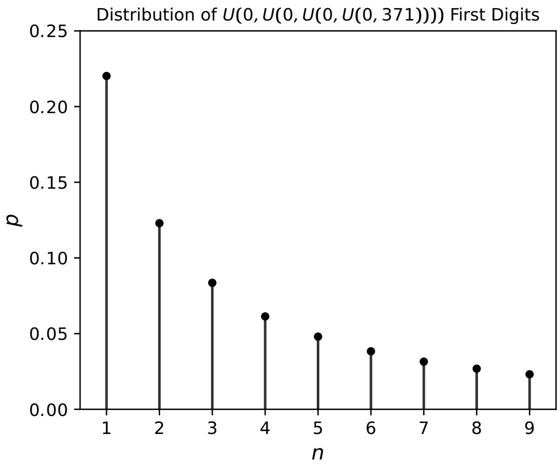
ICMA393 Discrete Simulation: HW3

Mahidol University International College
Austin Jetrin Maddison
October 28, 2024

Problem 1

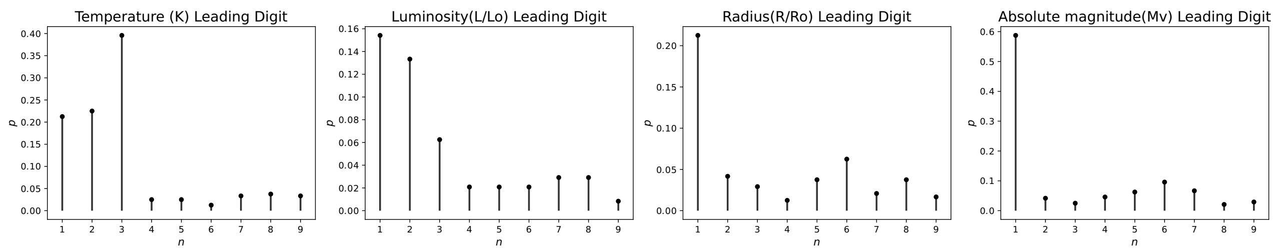
I ran the simulation for N=100000 iterations, for the $U(0, U(0, U(0, U(0, 371))))$.

```
def chain_dist(chains=4, L=0, U=371):
    if chains == 0:
        return np.random.randint(L, U)

    U_new = chain_dist(chains-1, L, U)
    if U_new == L : return U_new

    return np.random.randint(L, U_new)

f = np.vectorize(lambda x : chain_dist())
leading_digit =  np.vectorize(lambda _: int(str(
    np.abs(_))[0]))

N = 1000000
res = leading_digit( f( np.zeros(N) ) )
```



Problem 2

```
df = pd.read_csv('../data/stars.csv')
# ['Temperature (K)', 'Luminosity(L/Lo)', 'Radius(R/Ro)','Absolute magnitude(Mv)']

for i, col in enumerate(cols):
    res = leading_digit(df[col].values)
    plot(res)
```



Problem 3

```
# (a)
C.guess([0, 1, 1, 4, 9, 25, 64, 169, 441, 1156])
```

ans = $\frac{(-x^2+x)}{(x^3-2*x^2-2*x+1)}$

```
# (b)
C.guess([i**2 for i in range(10)])
```

ans = $\frac{(-x^2-x)}{(x^3-3*x^2+3*x-1)}$

```
# (c)
C.guess([int((i/2)**3)//1 for i in range(30)])
```

ans = $\frac{(x^10-2*x^9+4*x^8-2*x^7+3*x^6-x^5+2*x^4+x^2)}{(x^11-3*x^10+3*x^9-x^8-x^3+3*x^2-3*x+1)}$

Problem 4

Guess the C-finite relation of the number of palindromic compositions of n.

```
def find_all_sums(target_sum):
    def find_sums_recursive(remaining_sum, possible_numbers, current_combination, all_combinations):
        if remaining_sum == 0:
        all_combinations.append(current_combination)
        return
        for x in possible_numbers:
        if x <= remaining_sum:
        find_sums_recursive(remaining_sum - x, possible_numbers, current_combination + [x],
            all_combinations)

    possible_numbers = [num for num in range(1, target_sum + 1)]
    all_combinations = []
    find_sums_recursive(target_sum, possible_numbers, [], all_combinations)
    return all_combinations

def is_palindromic(sequence):
    if len(sequence) < 2:
        return True
    return sequence == sequence[::-1]

def find_all_palindromic_sums(target_sum):
    return [s for s in find_all_sums(target_sum) if is_palindromic(s)]

calc_length_of_palindromic_sums = lambda x : len(find_all_palindromic_sums(x))
guess_c_finite([ calc_length_of_palindromic_sums(i) for i in range(1, 20)])
```

ans = $c_0 = 0, c_1 = -2$

Problem 5

a) Find a formula for a_n

```
c1 = symbols('c1')
c2 = symbols('c2')
c3 = symbols('c3')
c4 = symbols('c4')
c5 = symbols('c5')
c6 = symbols('c6')

eqs = []
for i in range(0,6):
    eqs += [(Eq(2 ** i * (c1 + i*c2 + i**2* c3 + i**3 * c4) + 3 ** i * (c5 + c6*i),
        given_cfinite_seq(i)))]
sol = solve(eqs, (c1, c2, c3, c4, c5, c6))
# sol = {c1: -1, c2: 0, c3: -9, c4: 0, c5: 1, c6: 1}

print(f"= 2^n({sol[c1]} + n*{sol[c2]} + n^2* {sol[c3]} + n^3 * {sol[c4]}) + 3^n * ({sol[c5]} + {
    sol[c6]}*n)")
```

ans = $2^n(-1 + 0 * n + n^2 * -9 + n^3 * 0) + 3^n * (1 + 1 * n)$

b) Is there a linear recurrence for the same sequence for a smaller value of d? If not, explain why.

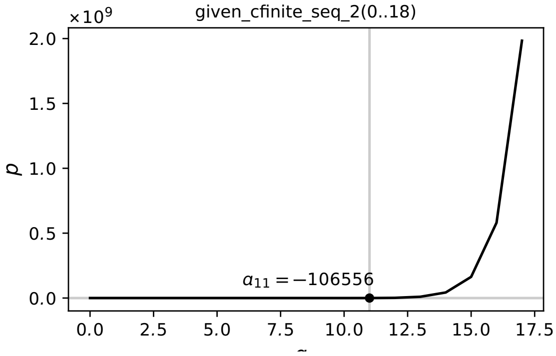
```
seq d=5 guess_c_finite() =  {c0: -12, c1: 57, c2: -134, c3: 156, c4: -72}
seq d=6 guess_c_finite() =  {c0: -12, c1: 57, c2: -134, c3: 156, c4: -72}
```

Index	seq d=5	seq d=6
0	0	0
1	-14	-14
2	-121	-121
3	-548	-548
4	-1915	-1915
5	-5774	-5774
6	-15697	-15697
7	-39080	-39080
8	-88663	-88663
9	-176930	-176930
10	-273085	-273085
11	-106556	-106556
12	1596221	1596221
13	9852298	9852298
14	42826775	42826775
15	163194544	163194544
16	580733777	580733777
17	1983473590	1983473590

It seems as though both sequences share the same recursive relation as the same output despite one having a smaller d than the other.

c.) Are there infinitely many negative terms in the sequence, or are there only a finite number of them? If there are infinite, explain why. If there are finite, find the largest index n for which an < 0.

No there isn't infinitely many negatives, as you can observe from the table prior or look at the plot on the right, at some point, the sequence exponentially increases and becomes greater than zero at n=12, making n=11 the last negative term.



Source Code

https://github.com/AustinMaddison/discrete-simulation/tree/main/hw3/source