Coffee bean titration

MCSB Bootcamp — Mathematical and Computational Track

Jun Allard

```
import numpy as np
import matplotlib.pyplot as plt
```

The coffee bean model

A jar holds `N=10` scoops of coffee. Each day you drink one scoop's worth and top the jar back up with decaf, so the fraction of the jar that is still caffeinated shrinks by a factor of $1 - 1/N$ every day.

```
N = 10  # number of scoops in each jar
n_max = 2 * N  # max number of days to simulate

x = np.zeros(n_max)  # fraction caffeinated
x[0] = 1.0  # initial fraction caffeinated

for n in range(1, n_max):

    x[n] = (1 - 1 / N) * x[n - 1]

# finished loop through days

days = np.arange(1, n_max + 1)

fig, ax = plt.subplots()
ax.plot(days, x, "-ok")
ax.set_ylabel("Fraction caffeinated")
ax.set_xlabel("Days")
plt.show()
```

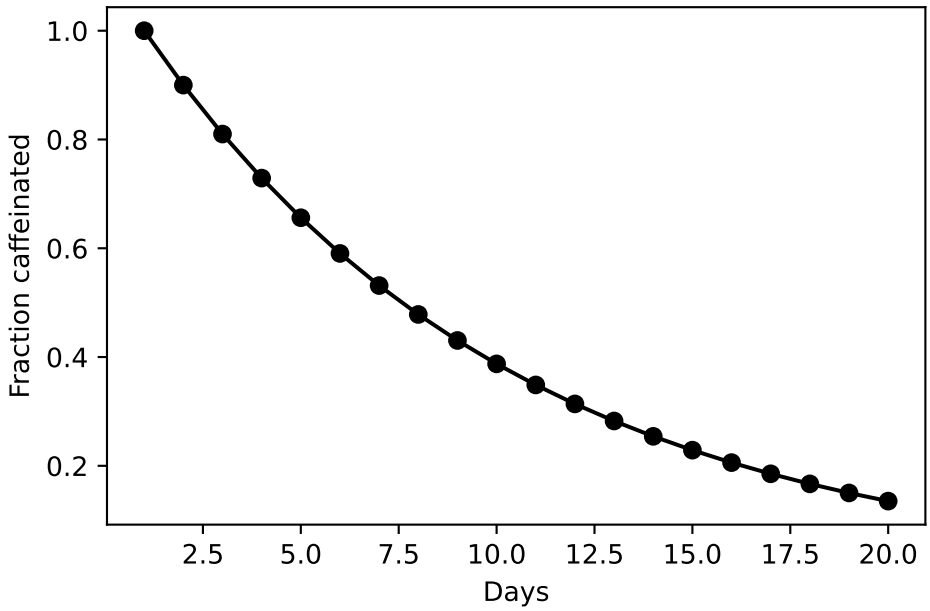


Figure 1: Fraction of the jar still caffeinated, day by day, for a ten-scoop jar.

Caffeine level trajectory analysis

```
query_day=9

fig, ax = plt.subplots()
ax.plot(days, x, "-ok")
ax.plot(days[query_day], x[query_day], "o", color="tab:red", markersize=12)
ax.annotate(
    f"x[{days[query_day]}] = {x[query_day]:.2f}",
    xy=(days[query_day], x[query_day]),  # the point the arrow lands on
    xytext=(days[query_day] + 3, x[query_day] + 0.2),  # where the text sits
    arrowprops=dict(arrowstyle="->", color="tab:red"),
    color="tab:red",
)
ax.set_ylabel("Fraction caffeinated")
ax.set_xlabel("Days")
plt.show()

print(f"After {days[query_day]} days, the fraction of the jar still caffeinated is {x[query_
```

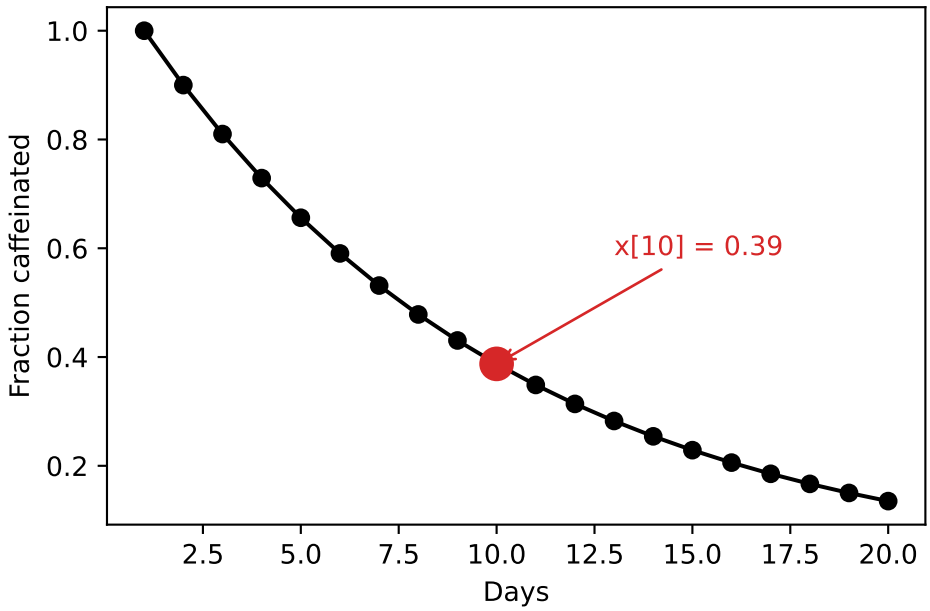


Figure 2: The same run, with the eleventh day marked and labelled.

```
After 10 days, the fraction of the jar still caffeinated is 0.39.
```

When does the caffeine level fall below 50%?

```
below = x < 0.5  # one True/False per day

first_index = np.argmax(below)

print(
    f"The caffeinated fraction first drops below 50% on day {days[first_index]}, "
    f"when it is {x[first_index]:.2f}."
)
```

```
The caffeinated fraction first drops below 50% on day 8, when it is 0.48.
```