Simulating a doctor’s waiting room

Goal: Simulate the queuing behavior at a doctor’s office.

Assumptions:

1. Patients arrive on the minutes between 9:00 am and 11:59 am.
2. At most one patient arrives during any minute.
3. The probability that a patient arrives in any one minute is 0.075.
4. Each patient needs to meet with the doctor for 15 minutes.

Expectations:

1. How many patients does the doctor expect to see in the day?
2. How much time does the doctor expect to spend with patients?
Simulating a doctor’s waiting room

How do we simulate the arrival of patients?

Assumptions:
1. Patients arrive on the minutes between 9:00 am and 11:59 am.
2. At most one patient arrives during any minute.
3. The probability that a patient arrives in any one minute is 0.075.

We let $i$ be the counter for time. $i = 0$ occurs at time 9:00 am.

$$i = ____$$ occurs at time 11:59 am.

We set up a For loop:

For[$i = 0$, $i ____$, $i ____$, 
newPatient = If[RandomReal[] <= 0.075, 1, 0] ]
Dealing with the waiting

How do we simulate “waiting” in a “waiting room”?  
An arriving patient will wait when ____________________________.

We will keep track of the following variables:

\[ n_{\text{wait}} = \text{The number of patients waiting.} \]

\[ \text{endTime} = \text{The time when the current patient finishes with the doctor.} \]

\[ \text{busy} = \begin{cases} 1 & \text{if the doctor is busy} \\ 0 & \text{if the doctor is free} \end{cases} \]


Simulating a doctor’s waiting room

Now translate the flowchart into an algorithm:

**Pseudocode:**

- Zero out the counters.
- For $i$ from 0 up to 180,
  - If the doctor is finishing with a patient at time $i$, then set busy=0.
  - Determine if a new patient arrives (random, probability 0.075). If so, set newPatient to 1; otherwise set newPatient to 0.
  - If newPatient == 1, add one to the number waiting ($nwait++$)
  - If the doctor is not busy AND there is a patient waiting,
    - Subtract one from the number waiting ($nwait--$)
    - Set the doctor to be busy (busy = 1)
    - Set the time when the doctor is not busy (endTime = $i + 15$).
Dealing with the waiting

```plaintext
nwait = 0; busy = 0; endTime = 0;
For[i = 0, i < 180, i++,
    If[endTime == i, busy = 0];
    newPatient = If[RandomReal[] <= 0.075, 1, 0];
    If[newPatient == 1, nwait++];
    If[busy == 0 && nwait > 0,
        nwait--; busy = 1; endTime = i + 15];
]

For i from 0 up to 180,
    - If the doctor is finishing with a patient at time i, then set busy=0.
    - Determine if a new patient arrives (random, probability 0.075).
      If so, set newPatient to 1; otherwise set newPatient to 0.
    - If newPatient == 1, add one to the number waiting (nwait++)
    - If the doctor is not busy AND there is a patient waiting,
      - Subtract one from the number waiting (nwait--)
      - Set the doctor to be busy (busy = 1)
      - Set the time when the doctor is not busy to i + 15.
```
What does the simulation tell us?

We did the simulation, but what was the point?

- How much of the day will the doctor be busy?
- What is the average number of patients in the waiting room?
- How many people are in the waiting room at noon?
- How late will the doctor stay after noon?

What statistics do we need to keep track of to answer these questions?

This is just one instance; the power of simulation comes from running the model many times and understanding the average behavior.
Gathering data

How many people are in the waiting room at noon?

```plaintext
nwait = 0; busy = 0; endTime = 0;
For[i = 0, i < 180, i++,
    If[endTime == i, busy = 0];
    newPatient = If[RandomReal[] <= 0.075, 1, 0];
    If[newPatient == 1, nwait++];
    If[busy == 0 && nwait > 0,
        nwait--; busy = 1; endTime = i + 15];
];

nwait ← [Outputs the value after the loop ends.]
```
Running many trials

- How many people are in the waiting room at noon?

Simulate 1000 times using a Table command, generate a histogram.

```
trials = Table[
  nwait = 0; busy = 0; endTime = 0;
  For[i = 0, i < 180, i++,
    If[endTime == i, busy = 0];
    newPatient = If[RandomReal[] <= 0.075, 1, 0];
    If[newPatient == 1, nwait++];
    If[busy == 0 && nwait > 0,
      nwait--; busy = 1; endTime = i + 15];
  ],
  nwait ], {j, 1000}]
Mean[trials] ←− [Average: 3.105]
Histogram[trials]
```
Gathering data

How late will the doctor stay after noon?

```math
nwait = 0; busy = 0; endTime = 0;
For[i = 0, i < 180, i++,
   If[endTime == i, busy = 0];
   newPatient = If[RandomReal[] <= 0.075, 1, 0];
   If[newPatient == 1, nwait++];
   If[b \text{usy} = 0 \text{ and } n \text{wait} > 0,
      nwait--; busy = 1; endTime = i + 15];
];
```

*****

↑ [How to calculate how long the doctor has to stay after noon?]

Mean: 51 minutes (does this make sense?)
Gathering and plotting time-dependent data

How much of the day will the doctor will be busy?

Keep track of if the doctor is busy by using a variable isBusy.

\[ \text{nwait} = 0; \text{busy} = 0; \text{endTime} = 0; \]
\[ \text{For}[i = 0, i < 180, i++, \]
\[ \quad \text{If}[\text{endTime} == i, \text{busy} = 0]; \]
\[ \quad \text{newPatient} = \text{If}[\text{RandomReal[]} \leq 0.075, 1, 0]; \]
\[ \quad \text{If}[\text{newPatient} == 1, \text{nwait}++]; \]
\[ \quad \text{If}[\text{busy} == 0 \&\& \text{nwait} > 0, \]
\[ \quad \quad \text{nwait}--; \text{busy} = 1; \text{endTime} = i + 15]; \]
\[ \quad \text{isBusy}[i]=\text{busy}; \quad \text{[this copies busy into isBusy[i].]} \]
\[ ]; \quad \downarrow \text{[Puts data into a list.]} \]
\[ \text{busyList} = \text{Table}[\text{isBusy}[i], i, 0, 179] \]
\[ \text{Total}[\text{busyList}] \quad \text{[Total time busy.]} \]
\[ \text{ListLinePlot}[\text{busyList}] \]
Gathering and plotting time-dependent data

What is the average number of patients in the waiting room?

Keep track of number of waiting patients by using a variable `numWait`.

```math
nwait = 0; busy = 0; endTime = 0;
For[i = 0, i < 180, i++,
   If[endTime == i, busy = 0];
   newPatient = If[RandomReal[] <= 0.075, 1, 0];
   If[newPatient == 1, nwait++];
   If[buzy == 0 && nwait > 0,
      nwait--; busy = 1; endTime = i + 15];
   numWait[i]=nwait; ← [this copies nwait into numWait[i].]
];
waitList=Table[numWait[i],i,0,179]
m=Mean[waitList] ← [Average patients.]
p1=ListLinePlot[waitList];
p2=Plot[m,x,0,179];
Show[p1,p2]
```