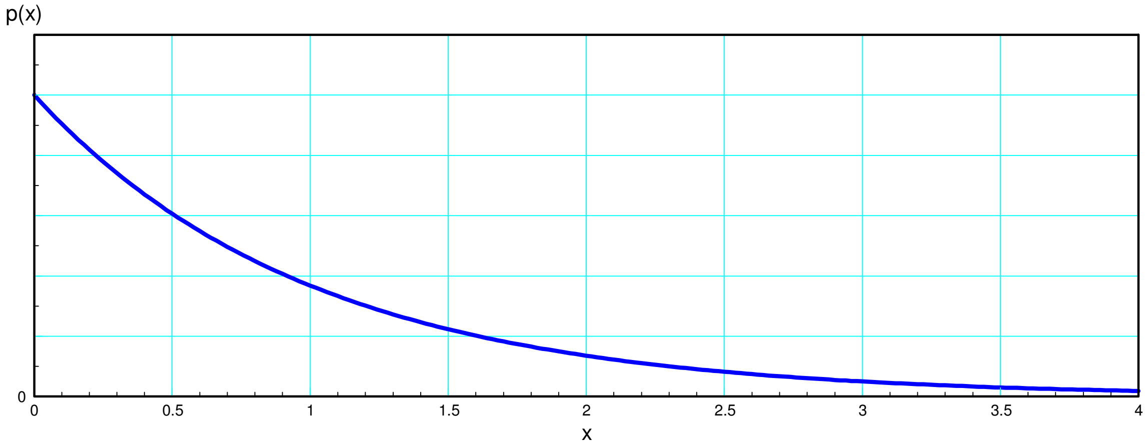
Exponential Distribution

Another common continuous probabilty distribution is the exponential distribution. With this distribution, the pdf is exponential:

$$f_X(x) = \begin{cases} a\, e^{-ax} & 0 < x < \infty \\ 0 & otherwise \end{cases}$$



pdf for an exponential distribution

Examples of where this type of distibution is encountered is:

- Probability that the length of a telephone call is less than x minutes
- Probability that the next atom will decay within x seconds
- Time it takes for the next customer to arrive at a store

Exponential distributions also lead into queueing theory: how long a customer will have to wait in line to be served.

Parameters of Exponential Distributions

pdf: As stated before

$$pdf(x) = a\, e^{-ax}$$

cdf: The cdf is the integral of the pdf

$$cdf(x) = \int_0^x (a\, e^{-at})\, dt$$

$$cdf(x) = 1 - e^{-ax} \qquad x > 0$$

Moment Generating Function: Start with this since it makes everything else a lot easier

$$\psi(s) = \left(\tfrac{a}{s+a}\right)$$

To be a valid probability, the area under the curve (i.e. the zeroth moment) must be 1.00000

$$\psi(s=0) = 1 = \left(\tfrac{a}{a}\right)$$

Mean: The mean of an exponential distribution is

$$\bar{x} = \int_0^\infty p(x)\, x\, dx$$

$$\bar{x} = \int_0^\infty (ae^{-ax})\, x\, dx$$

$$\bar{x} = \left(\tfrac{1}{a}e^{-ax}(-ax-1)\right)_0^\infty$$

$$\bar{x} = \left(\tfrac{1}{a}\right)$$

This can also be found using the moment generating funciton

$$\psi(s) = \left(\tfrac{a}{s+a}\right)$$

$$\bar{x} = m_1 = \psi'(0)$$

$$\bar{x} = \left(\tfrac{a}{(s+a)^2}\right)_{s=0} = \tfrac{1}{a}$$

Variance: Use the moment generating function - it's a lot easier

$$s^2 = m_2 - m_1^2$$

$$m_2 = \psi''(0)$$

$$m_2 = \left(\tfrac{2a(s+a)}{(s+a)^4}\right)_{s=0} = \left(\tfrac{2}{a^2}\right)$$

so

$$s^2 = \left(\tfrac{2}{a^2}\right) - \left(\tfrac{1}{a}\right)^2$$

$$s^2 = \left(\tfrac{1}{a^2}\right)$$

Matlab Example:

Generate 10 random variables with an exponential distibution where a = 1

Solution: Determine the probability using the *rand* function in matlab

```
p = rand
```

Convert to x using the cdf

$$p = cdf(x) = 1 - e^{-ax}$$

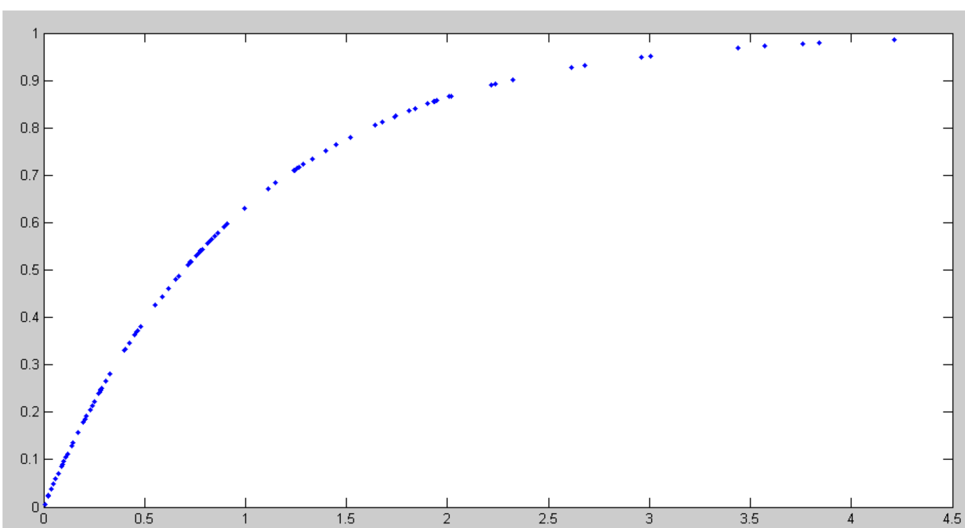
$$x = -\left(\frac{1}{a}\right) \ln(1-p)$$

In matlab

```
p = rand(10,1);

a = 1;
x = -(1/a) * log(1-p)

[p, x]

    0.6862    1.1590
    0.9593    3.2004
    0.4850    0.6635
    0.7880    1.5511
    0.7550    1.4066
    0.7228    1.2831
    0.2646    0.3074
    0.6884    1.1660
    0.9323    2.6930
    0.0902    0.0946
```

Plotting these together give you the cdf: (plotted with 100 points for x)

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



Plotting x vs. probability gives the cdf for an exponential distribution

Example: A block of radioactive material is sitting next to a Geiger counter. In 30 minutes, the Geiger counter detects 100 atoms decaying. Determine the pdf and CDF for the time until the next atom decays (in minutes).

Solution: The average number of atoms decaying per minute is

$$\bar{x} = \frac{100 \text{ atoms}}{30 \text{ minutes}} = 3.33 \frac{\text{atoms}}{\text{min}} = \frac{1}{a}$$

$$a = 0.3$$

pdf:

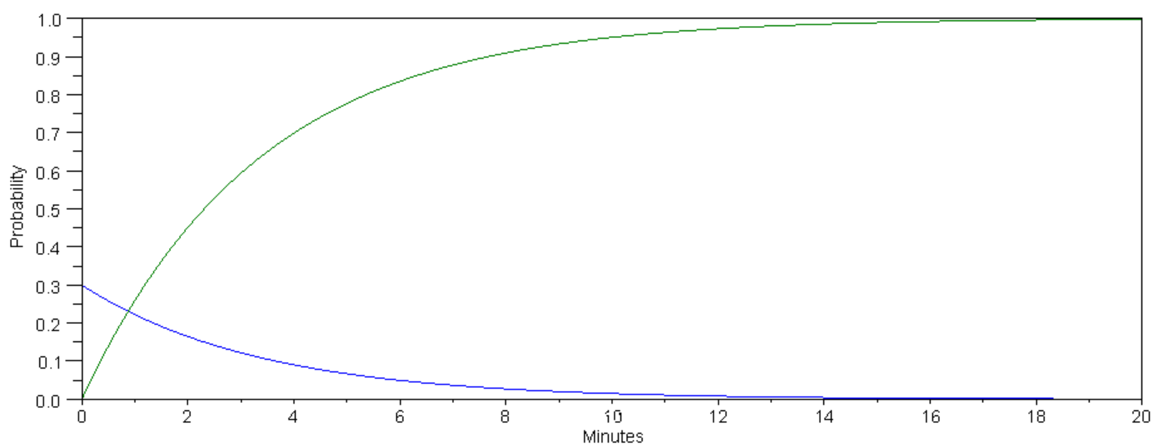
$$f_X(x) = \begin{cases} 0.3 \cdot e^{-0.3x} & 0 < x < \infty \\ 0 & otherwise \end{cases}$$

CDF:

$$F_X(x) = \begin{cases} 1 - e^{-0.3x} & 0 < x < \infty \\ 0 & otherwise \end{cases}$$

Matlab

```
x = [0:0.01:20]';

p = 0.3 * exp(-0.3*x);

C = 1 - exp(-0.3*x);

plot(x,p,x,C)
xlabel('Minutes');
ylabel('Probability');
```



pdf (blue) and CDF (green) for time until you detect an atom decaying

Queing Theory

One place where exponential distributions are often used are in determining how many servers are needed at a restaraunt.

For example, assume

- Customers arrive at a fast-food restaraunt with
  - An exponential distribution, with
  - An average of one customer every minute.
- It takes an average of 30 seconds to serve each customer.

How long will the longest wait be for a 1 hour shift?

Start with a simulation for 5 customers. This number can be increased later.

Next, define a bunch of arrays:

- Tarr[N]: The time the Nth customer arrives
- Tser[N]: The time the Nth customer get to the front of the line and places his/her order
- Tdone[N]: The time the Nth customer gets his/her order and leaves
- Twait[N]: The time the Nth customer spends waiting in line
- Queue[N]: The queue size for the Nth customer. How many people are in line when he/she arriaves.

Once these variables are defined, start with the first customer:

- The time the first customer arrives is an exponential distribution with a mean of 60 seconds.
- There's no-one in line when the 1st customer arrives, so he/she is served right away.
- The time the 1st customer's order is completed (Tdone) is then the time he/she was served, plus the time it takes to complete that order (an exponential distribution with a mean of 30 seconds0.

The matlab code for the 1st customer would be:

```
N = 5;

Tarr = zeros(N,1);
Tser = zeros(N,1);
Tdone = zeros(N,1);
Twait = zeros(N,1);
Queue = zeros(N,1);

% Start with the first customer
n = 1;
Tarr(n) = -60*log(1 - rand);
Tser(n) = Tarr(n);
Twait(n) = 0;
Tdone(n) = Tser(n) -30*log(1-rand);
Queue(n) = 0;

[Tarr, Tser, Twait, Tdone, Queue]
```

The The reslting arrays are then (numbers vary)

```
Tarr     Tser     Twait    Tdone     Q
68.77    68.77        0    196.52    0
 0        0           0      0       0
 0        0           0      0       0
 0        0           0      0       0
 0        0           0      0       0
```

Now that the first customer has arrived and had their order completed, bring in more customers.

- Each subsequent customer arrives ar a random time equal to
  - The time the previous customer arrived, plus
  - A random time with an exponential distribution with a mean of 60 seconds.
- The time the next customer is served and can place his/her order is the maximum of
  - The time that customer arrived, and
  - The time the previous customer finished their order and was served.
- Once served, the time the next customer is done is the time he/she was served plus the time it takes to complete their order (a random time with an exponential distirbution with a mean of 30 seconds)

```
for n=2:N
   Tarr(n) = Tarr(n-1) - 60*log(1-rand);
   Tser(n) = max(Tarr(n), Tdone(n-1));
   Tdone(n) = Tser(n) - 30*log(1-rand);
end

[Tarr, Tser, Twait, Tdone, Queue]
```

Running a simulation results in the following (numbers vary each run)

```
arrive   serve    wait   done   queue
 31.0    31.0       0    50.3     0
 45.1    50.3       0    83.2     0
130.1   130.1       0   145.4     0
140.4   145.4       0   166.2     0
146.1   166.2       0   201.9     0
```

Finally, add in the wait time and the queue size.

- The wait time is the difference between when the Nth customer arrievs (Tarr) and when he/she is served (Tser).
- The queue size is the number of previous customers who are still in line when the Nth customer arrives

```
for n=2:N
  Tarr(n) = Tarr(n-1) - 60*log(1-rand);
  Tser(n) = max(Tarr(n), Tdone(n-1));
  Tdone(n) = Tser(n) - 30*log(1-rand);
  Twait(n) = Tser(n) - Tarr(n);
  Queue(n) = sum( Tdone(1:n-1)>Tarr(n));
end
```

A sanple output for 5 custumers looks like this:

```
arrive   serve   wait   done   queue
 10.8     10.8    0      12.8    0
 91.6     91.6    0     160.7    0
107.6    160.7   53.0   168.2    1
141.4    168.2   26.7   169.7    2
208.1    208.1    0     212.0    0
```

- Next, start with the first customer. For that customer, c

The overall Matlab code for 30 customers is:

```
N = 30;

Tarr = zeros(N,1);
Tser = zeros(N,1);
Tdone = zeros(N,1);
Twait = zeros(N,1);
Queue = zeros(N,1);

% Start with the first customer
n = 1;
Tarr(n) = -60*log(1-rand);
Tser(n) = Tarr(n);
Twait(n) = 0;
Tdone(n) = Tser(n) -30*log(1-rand);
Queue(n) = 0;
for n=2:N
  Tarr(n) = Tarr(n-1) - 60*log(1-rand);
  Tser(n) = max(Tarr(n), Tdone(n-1));
  Tdone(n) = Tser(n) - 30*log(1-rand);
  Twait(n) = Tser(n) - Tarr(n);
  Queue(n) = sum( Tdone(1:n-1)>Tarr(n));
end

[Tarr, Tser, Twait, Tdone, Queue]
```

with outputs:

arrive	serve	wait	done	queue
0.3	0.3	0	8.2	0
10.1	10.1	0	26.1	0
33.4	33.4	0	134.4	0
116.6	134.4	17.8	160.4	1
209.0	209.0	0	220.1	0
229.2	229.2	0	233.7	0
239.5	239.5	0	253.8	0
245.7	253.8	8.1	265.0	1
343.4	343.4	0	387.5	0
358.9	387.5	28.6	417.5	1
429.5	429.5	0	438.0	0
479.8	479.8	0	512.1	0
540.4	540.4	0	543.5	0
590.8	590.8	0	674.2	0
610.8	674.2	63.4	755.4	1
655.7	755.4	99.8	759.3	2
671.0	759.3	88.3	763.4	3
835.4	835.4	0	871.1	0
854.7	871.1	16.4	892.7	1
857.9	892.7	34.8	956.8	2

In this case, the maximum wait time and maximum queue size can be seen as

- Max wait time = 99.8 seconds
- Max queue size = 3

These are important numbers:

- If the wait time is too large, customers will turn around and leave.
- If the queue size is too large, again, customers will turn around and leave.
- Both are lost revenue.

If either gets too large, some employees should be reassigned to work the cash registers, reducing the service time.

On the other hand, if the wait time and queue size is too small, you can reassign some servers to other tasks.

Simulation Runs

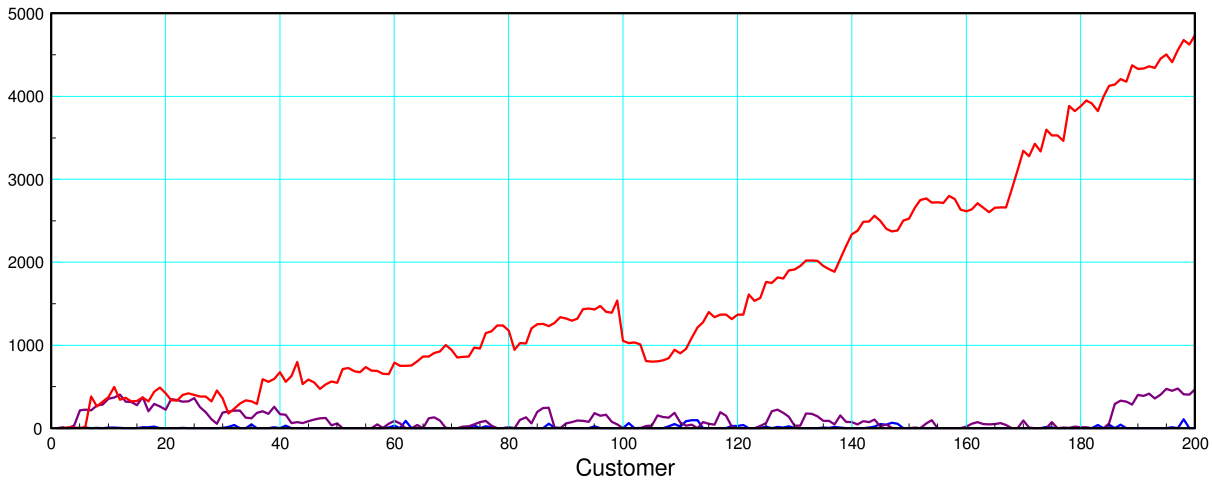
By running simulations, you can see the impact of changing the time it takes to serve a customer. In the following plots, it assumes:

- Mean arrival time is 60 seconds, and
- Mean time to complete an order is
  - 40% of the arrivla time (24 seconds)
  - 80% of the arrival time (48 seconds), and
  - 120% of the arrival time.

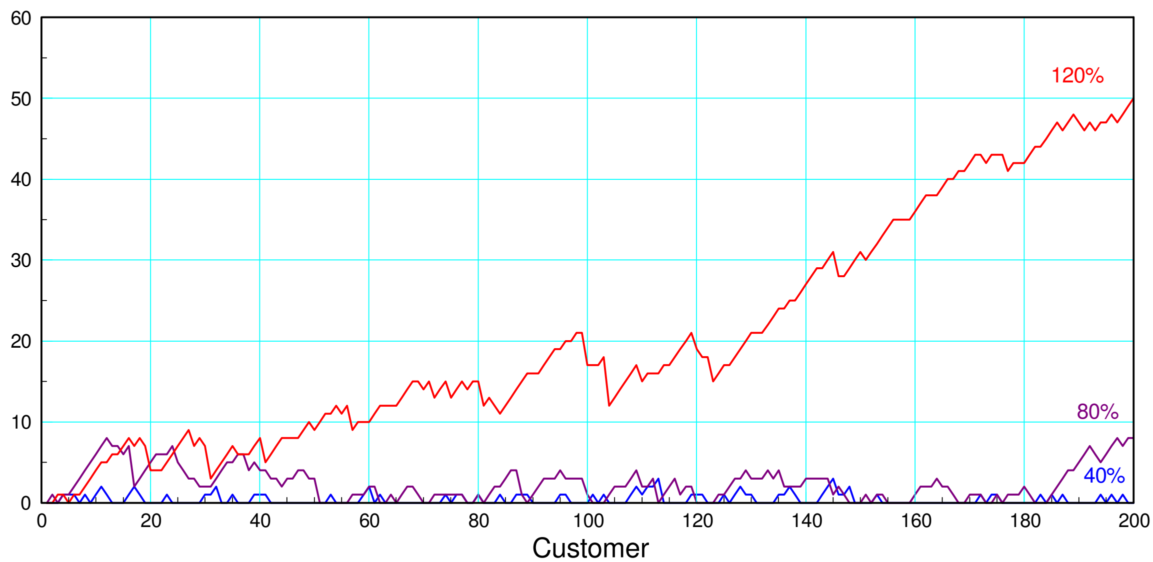
Run each simulation for 200 customers.

First, let's look at the waiting time for each customer.

- When customers are arriving faster than orders are being filled (red line), the waiting time and the queue size increase without limit.
- When orders are completed 20% faster than customers arrive (purple lines), the wait times and queue sizes can still get fairly large.
- When orders are completed 60% faster than customers arrive (blue lines), the queue size and waiting times remain small.



Waitimg Time when the time to complete an order is 40% (blue), 80% (purple) and 120% (red) of the arrival time.



Queue size when the time to complete an order is 40% (blue), 80% (purple) and 120% (red) of the arrival time.

Summary

Exponential distributions model many systems

- Time of a phone call
- Time until an atom decays
- Time until the next customer arrives

Exponential distribution also lead to a field called queeing theory. This looks at modeling the time customers have to wait as well as the number of people standing in line for a given arrival time and time to complete orders.