EIND 464 Assignment 6

4/27/23

Thomas Lipinski

In [ ]:
```
using LinearAlgebra, Distributions, Printf
```

Section 20.2

Problem 4

In [ ]:
```
# The time between arrivals of buses follows an exponential distribution, with a mean
# of 60 minutes. λ is the arrival rate (buses per hour).

λ = 1

# a) What is the probability that exactly four buses will arrive during the next 2 hours?

t = 2
x = 4 # four buses in period t

ArrivalTimes = Poisson(λ*t) # Create a Distribution object to work with
Answer = pdf(ArrivalTimes,x)
@printf "A. The probability of exactly 4 buses arriving
        in the next two hours is: %5.4f. \n" Answer
```

A. The probability of exactly 4 buses arriving in the next two hours is: **0.0902**

In [ ]:
```
# b) That at least two buses will arrive during the next two hours?

Answer = 0
for i in 2:33 # summing values to a fairly high Poisson Input value
    Answer += pdf(ArrivalTimes, i)
end

@printf "B. The probability that at least two buses will arrive during
        the next two hours is: %5.4f. \n" Answer
```

B. The probability that at least two buses will arrive during the next two hours is: **0.5940**

In [ ]:
```
# c ) That no buses will arrive during the next 2 hours?

Answer = pdf(ArrivalTimes,0)

@printf "The probability that no buses arrive during
        the next two hours is: %5.4f. \n" Answer
```

C. The probability that no buses arrive during the next two hours is: **0.1353**.

In [ ]:
```
# d) A bus has just arrived. What is the probability that
# it will be between 30 and 90 minutes before the next bus arrives?

# Look at the continuous aspect for time, so Exponential dist...
InterArrivalTimes = Exponential(λ)

# subtract area to the left of 30 mins from the area to the left of 90 minutes
Answer = cdf(InterArrivalTimes, 3λ/2) - cdf(InterArrivalTimes, λ/2)

@printf "The probability it will be between 30 and 90 minutes
         before the next bus arrives is %5.4f.\n" Answer
```

D. The probability it will be between 30 and 90 minutes before the next bus arrives is **0.3834**.

Problem 7

In [ ]:
```
# An average of 12 jobs per hour arrive at our departmental printer.
# 12 jobs / 60 minutes -> 1 job / 5 minutes
λ = 5 # five minutes per job
λ = 12 # twelve jobs per hour

# a) Use two different computations (one involving the Poisson and another the
# exponential random variable) to determine the probability that no job will arrived
# during the next 15 minutes.
t = 1/4

EXP = Exponential(5)
# 1 - cdf, where cdf if the integral of EXP from 0 to 15
ProbabilityOfNoArrivalsExp = ccdf(EXP,15)

# Just need the probablity of the discrete value 0 on a
POI = Poisson(λ*t)  # Poi dist with parameter 3 jobs / 15 minutes
ProbabilityOfNoArrivalsPoi = pdf(POI,0)

@printf "By the Poisson distribution, the Probability of no buses
        arriving in 15 minutes is %5.4f.\n" ProbabilityOfNoArrivalsPoi
@printf "By the Exponential distribution, the Probability of no
    buses arriving in 15 minutes is %5.4f.\n" ProbabilityOfNoArrivalsExp
```

A. By the Poisson distribution, the Probability of no buses arriving in 15 minutes is **0.0498**.

By the Exponential distribution, the Probability of no buses arriving in 15 minutes is **0.0498**.

In [ ]:
```
# b What is the probability that 5 or fewer jobs will arrive during the next 30 minutes?

POI = Poisson(λ/2)
Answer = cdf(POI,5)
@printf "B. The probability that five or fewer jobs arrive in the next thirty
            minutes is %5.4f.\n" Answer
```

B. The probability that five or fewer jobs arrive in the next thirty minutes is **0.4457**.

Section 20.3

Problem 2

In [ ]:

```
# My home uses two light bulbs. On average, a light bulb lasts for 22 days
# (exponentially distributed). When a light bulb burns out, it takes an average
# of 2 days (exponentially distributed) before I replace the bulb.

# States
# 0 Bulbs working, 1 Bulbs working, 2 Bulbs working

# Birth processes
#   (λ₀  λ₁  λ₂)
λ = [1, 1/2, 0]

# Death processes
#   (μ₀   μ₁   μ₂)
μ = [0, 1/22, 1/11]

# a) Formulate a three-state birth-death model of this situation.

ρ₀ = λ[1]/μ[2]
ρ₁ = λ[2]/μ[3]

# eqn 0:  π₀       +   π₁ + π₂ = 1

# eqn 1: -π₀*ρ₀    +   π₁      = 0

# eqn 2: -π₀*ρ₁*ρ₀ +       π₂  = 0

A = [ 1     1    1;
     -ρ₀    1    0;
     -ρ₀*ρ₁ 0    1]

b = [1;
     0;
     0]

Answer = A \ b

# b) Determine the fraction of the time that both light bulbs are working.
@printf "B. The fraction of the time that both light bulbs
      are working is %5.4f \n" Answer[3]
# c) Determine the fraction of the time that no light bulbs are working.
@printf "C. The fraction of the time that no light
      bulbs are working is %5.4f \n" Answer[1]
```

B. The fraction of the time that both light bulbs are working is **0.8403**

C. The fraction of the time that no light bulbs are working is **0.0069**

Problem 3

In [ ]:

```
F =   0   # Number of pizza restrants in Bloomington
birthRate(F) = max(0, 16-(F/2))

deathRate(z::Float64) = inv((10 + z))

# States holds [Birth rate, death rate, Traffic intensity and steady state probability] fo
States = zeros(Float64, 4, 33)

for i in 0:32
    if i == 0
        States[:,i+1] = [birthRate(i), 0, 1, 0]
    else
        States[:,i+1] = [birthRate(i),
                         deathRate(birthRate(i)),
                         deathRate(birthRate(i-1)) * birthRate(i) / deathRate(birthRate(i))
                         0]
    end
end

States[4,1] = inv(sum(States[3,:]))
for i in 2:33
    States[4,i] = States[3,i] * States[4,1]
end

avResturants = sum(i*States[4,i+1] for i in 0:32)
partB = sum(States[4,i] for i in 20:32)

@printf "A. In steady state, the average number of pizza
                resturants is: %4.2f\n" avResturants
@printf "B. The fraction of the time will there be more than 20
            pizza restaurants in Bloomington is %4.2f\n" partB
# println("| State |   λⱼ    |    μⱼ   |   ρⱼ    |   πⱼ    |")
# println("+-------+---------+---------+---------+---------+")
# for i in 0:32
#     @printf "|%4d   |" i
#     for j in 1:4
#         @printf "%8.3f |" States[j,i+1]
#     end
#     println()
# end
# | State |   λⱼ    |    μⱼ   |   ρⱼ    |   πⱼ    |
# +-------+---------+---------+---------+---------+
# |   0   |  16.000 |   0.000 |   1.000 |   0.004 |
# |   1   |  15.500 |   0.039 |  15.202 |   0.063 |
# |   2   |  15.000 |   0.040 |  14.706 |   0.061 |
# |   3   |  14.500 |   0.041 |  14.210 |   0.059 |
# |   4   |  14.000 |   0.042 |  13.714 |   0.056 |
# |   5   |  13.500 |   0.043 |  13.219 |   0.054 |
# |   6   |  13.000 |   0.043 |  12.723 |   0.052 |
# |   7   |  12.500 |   0.044 |  12.228 |   0.050 |
# |   8   |  12.000 |   0.045 |  11.733 |   0.048 |
# |   9   |  11.500 |   0.047 |  11.239 |   0.046 |
# |  10   |  11.000 |   0.048 |  10.744 |   0.044 |
# |  11   |  10.500 |   0.049 |  10.250 |   0.042 |
# |  12   |  10.000 |   0.050 |   9.756 |   0.040 |
# |  13   |   9.500 |   0.051 |   9.263 |   0.038 |
# |  14   |   9.000 |   0.053 |   8.769 |   0.036 |
# |  15   |   8.500 |   0.054 |   8.276 |   0.034 |
# |  16   |   8.000 |   0.056 |   7.784 |   0.032 |
```

```
# |  17   |   7.500 |   0.057 |   7.292 |   0.030 |
# |  18   |   7.000 |   0.059 |   6.800 |   0.028 |
# |  19   |   6.500 |   0.061 |   6.309 |   0.026 |
# |  20   |   6.000 |   0.062 |   5.818 |   0.024 |
# |  21   |   5.500 |   0.065 |   5.328 |   0.022 |
# |  22   |   5.000 |   0.067 |   4.839 |   0.020 |
# |  23   |   4.500 |   0.069 |   4.350 |   0.018 |
# |  24   |   4.000 |   0.071 |   3.862 |   0.016 |
# |  25   |   3.500 |   0.074 |   3.375 |   0.014 |
# |  26   |   3.000 |   0.077 |   2.889 |   0.012 |
# |  27   |   2.500 |   0.080 |   2.404 |   0.010 |
# |  28   |   2.000 |   0.083 |   1.920 |   0.008 |
# |  29   |   1.500 |   0.087 |   1.438 |   0.006 |
# |  30   |   1.000 |   0.091 |   0.957 |   0.004 |
# |  31   |   0.500 |   0.095 |   0.477 |   0.002 |
# |  32   |   0.000 |   0.100 |   0.000 |   0.000 |
```

A. In steady state, the average number of pizza resturants is: **10.92**

B. The fraction of the time will there be more than 20 pizza restaurants in Bloomington is **0.18**

Section 20.4

Problem 4

In [ ]:
```
# A fast-food restaurant has one drive-through window. An average of 40 customers per
# hour arrive at the window. It takes an average of 1 minute to serve a customer. Assume
# that interarrival and service times are exponential.
λ = 40  # Units are customers per hour
μ = 60  # Units are customers per hour
ρ = λ/μ
# a) On the average, how many customers are waiting in line?

# L = inv(1-ρ)
Lₛ = ρ                    # Time in service
Lᵦ = λ^2 / (μ * (μ - λ)) # Time in queue
L = Lᵦ + Lₛ               # Time in system
# println(L)
@printf "A. The average number of customers waiting in line is: %.2f\n" Lᵦ
```

A. The average number of customers waiting in line is: **1.33**

In [ ]:
```
# b) On the average, how long does a customer spend at the restaurant
# (from time of arrival to time service is completed)?

W = (L*60) / λ
@printf "B. On average, a customer spends %.2f minutes at the resturant.\n" W
```

B. On average, a customer spends **3.00** minutes at the resturant.

In [ ]:
```
# c) What fraction of the time are more than 3 cars waiting for service
#  (this includes the car (if any) at the window)?
P(n) = (1-ρ)*ρ^n # function definition
C = 0
for i in 0:3
    C += P(i)
end
C = 1 - C
@printf "C. The portion of time there are more than three cars
         waiting for service is: %4.4f\n" (C)
```

C. The portion of time there are more than three cars waiting for service is: **0.1975**

In [ ]:
```
# Problem 10

# Consider an airport where taxis and customers arrive (exponential interarrival times)
# with respective rates of 1 and 2 per minute. No matter how many other taxis are
# present, a taxi will wait. If an arriving customer does not find a taxi, the
# customer immediately leaves.

# a) Model this system as a birth-death process (Hint: Determine what the state
#        of the system is at any given time and draw a rate diagram.)
```

Because The number of customers is never greater than 1, the states will be the number of taxis waiting. (As number of taxis can be any discrete nonnegative value)

In state zero: The arrival rate is 1/min, and the departure rate is zero (customers balk if there isn't a taxi available)

In state one: The arrival rate is still 1/min, but the departure rate changes to 2/min.

In state three: The arrival rate is still 1/min, and the departure rate is still 2/min.

In state j: The arrival rate is still 1/min, and the departure rate is still 2/min.

In [ ]:
```
# b) Find the average number of taxis that are waiting for a customer.

# Now for steady state probabilities:
λ = 1 # arrival per minute
μ = 2 # departures per minute
μ₀ = 0 # Zero departures when zero taxis

ρ = λ/μ

π₀ = 1-ρ

πₙ(n) = π₀ * ρ^n

AvTaxis = sum(i*πₙ(i) for i in 0:10000)

@printf "B. The average number of taxis waiting for a customer is: %2.2f\n" AvTaxis
```

B. The average number of taxis waiting for a customer is: **1.00**

C. Suppose all customers who use a taxi pay a $2 fare. During a typical hour, how much revenue will the taxis receive?

Well if each customer has, on average, one taxi waiting for them, then the number of departures in an hour will equal the number of taxi arrivals in the hour. Therefore:

60 taxi arrivals / hour * $2.00 per departure = **120.00**

Section 20.5

Problem 2

In [ ]:
```
# An average of 40 cars per hour (interarrival times are
# exponentially distributed) are tempted to use the drive-in
# window at the Hot Dog King Restaurant. If a total of
# more than 4 cars are in line (including the car at the window) a
# car will not enter the line. It takes an average
# of 4 minutes (exponentially distributed) to serve a car.
c = 4
λ = 40
μ = 15
ρ = λ/μ
```

In [ ]:
```
# a) What is the average number of cars waiting
# for the drive-in window (not including a car at the window)?
# L_q = (λ^2)/(μ*(μ - λ))
π₀ = (1-ρ)/(1-ρ^(c+1))

Ls = 1 - π₀

# Eqn 35 in chapter 20.5
L = (ρ*(1-(c + 1) * ρ^c + c * ρ^(c+1))/((1-ρ^(c+1)) * (1-ρ)))
L_q = L - Ls

@printf "A. The average number of customers in the queue is Lq: %3.3f\n" L_q
```

A. The average number of customers in the queue is Lq: **2.450**

In [ ]:
```
# b) On the average, how many cars will be served per hour?
π₄ = ρ^c * π₀
AvCust = λ * (1 - π₄)
@printf "B. The average number of cars server per hour is %2.2f\n" AvCust
```

B. The average number of cars server per hour is **14.81**

In [ ]:
```
# c) I have just joined the line at the drive-in window.
# On the average, how long will it be before I have received my food?
```

C. Due to the no-memory property of the Exponential distribution, the average time you will wait is the service rate μ, **4 minutes.**

Problem 6

In [ ]:
```
# Two one-man barber shops sit side by side in Dunkirk Square.
# Each can hold a maximum of 4 people, and any
# potential customer who finds a shop full will not wait for a
# haircut. Barber 1 charges $11 per haircut and takes an
# average of 12 minutes to complete a haircut. Barber 2 charges
# $5 per haircut and takes an average of 6 minutes to
# complete a haircut. An average of 10 potential customers per
# hour arrive at each barber shop. Of course, a potential
# customer becomes an actual customer only if he finds that
# the shop is not full. Assuming that interarrival times and
# haircut times are exponential, which barber will earn more money?

# Need to compare customers served per hour * cost.

c = 4
λ = 10 # Customers per hour
# Barber 1

μ₁ = 5
ρ₁ = λ/μ₁
π₀₁ = (1-ρ₁)/(1-ρ₁^(c+1))

π₄₁ = ρ₁^c * π₀₁
E1 = λ * (1 - π₄₁)
@printf "The average hourly earnings for barber 1 are: \$%2.2f\n" E1
# Barber 2
μ₂ = 9.99999
ρ₂ = λ/μ₂
# println((1 - ρ₂^(c + 1)))
π₀₂ = (1 - ρ₂)/(1 - ρ₂^(c + 1))

π₄₂ = ρ₂^c * π₀₂
E2 = λ * (1 - π₄₂)
@printf "The average hourly earnings for barber 2 are: \$%2.2f\n" E2
@printf "Barber 2 earns more money."
```

The average hourly earnings for barber 1 are: $4.84

The average hourly earnings for barber 2 are: $8.00

Barber 2 earns more money.

Section 20.6

Problem 7

In [ ]:

```
# An average of 50 customers per hour arrive at a small post
# office. Interarrival times are exponentially distributed.
# Each window can serve an average of 25 customers per hour.
# Service times are exponentially distributed. It costs
# $25 per hour to open a window, and the post office values
# the time a customer spends waiting in line at $15 per
# customer-hour.

# M/M/s/GD/∞/∞ Queue

λ = 50
μ = 25
ρ = λ / μ
# To minimize expected hourly costs, how many postal windows should be opened?
function probabilityofS(s::Int64, ρ::Float64)
    return (ρ)^s / (factorial(big(s)) * sum(((ρ)^k) for k in 0:s-1))
end

function numCusts(s::Int64, ρ::Float64, μ::Int64)
    Pₛ = probabilityofS(s,ρ)
    if s == 1
        term1 = 0
    else
        term1 = sum(ρ^k / (factorial(big(k)) * float(μ)^(k - s)) * Pₛ for k in 1:s-1)
    end
    return ρ * (Pₛ + term1)
end

function optimal_windows(λ, μ)
    ρ = λ / μ
    min_cost = Inf
    optimal_s = 0

    for s in 1:16
        cost = numCusts(s, ρ, μ)
        if cost < min_cost
            min_cost = cost

            optimal_s = s
            @printf "%d : %.16f\n" s min_cost
        end
    end

    return optimal_s
end

optimal_s = optimal_windows(λ, μ)

# for i in 1:25
#     @printf "%2d : %5.15f\n" i numCusts(i,ρ, μ)
# end
```

In [ ]:

```
λ = 50
μ = 25
ρ = λ / μ

c_open_window = 25
c_waiting = 15

function P0(s, ρ)
    if ρ / s == 1
        return 1 / (s + 1)
    else
        sum_part = sum([(ρ^k) / factorial(big(k)) for k in 0:(s - 1)])
        last_term = (ρ^s) / (factorial(big(s)) * (1 - (ρ / s)))
        return 1 / (sum_part + last_term)
    end
end

function ErlangC(s, ρ)
    sum_term = sum([(ρ^k) / factorial(k) for k in 0:(s - 1)])
    last_term = (ρ^s) / (factorial(s) * (1 - ρ / s))
    return last_term / (sum_term + last_term)
end

function num_cust_in_system(s, ρ)
    L = s * ρ + ρ * ErlangC(s, ρ) / #DIVIDED BY
           (1 - ρ / s)
    return L
end

function hourly_cost(s, λ, μ, c_open_window, c_waiting)
    ρ = λ / (s * μ)
    customers_in_system = num_cust_in_system(s, ρ)
    cost_wait = (customers_in_system - s) * c_waiting
    cost_windows = s * c_open_window
    return cost_wait + cost_windows
end

function optimal_windows(λ, μ, c_open_window, c_waiting)
    min_cost = Inf
    optimal_s = 0

    for s in 1:20
        cost = hourly_cost(s, λ, μ, c_open_window, c_waiting)
        # @printf "%2d : %4.6f\n" s cost
        if cost < min_cost
            min_cost = cost
            optimal_s = s
        end
    end

    return optimal_s
end

optimal_s = optimal_windows(λ, μ, c_open_window, c_waiting)
@printf "The optimal number of postal windows to be opened is: %d\n" optimal_s
```

The optimal number of postal windows to be opened is: **1**

In [ ]:
```
# A muffler shop has three mechanics. Each mechanic
# takes an average of 45 minutes to install a new muffler.
# Suppose an average of 1 customer per hour arrives.
# What is the expected number of mechanics that are busy at any
# given time? Answer this question without
# assuming that service times and interarrival times are exponential.

# each is busy 45/60
# meaning each is free 15/60 -> 1/4

@printf "The expected number of mechanics busy at any given time is %3.3f\n" (3/4)^3
```

The expected number of mechanics busy at any given time is **0.422**

I was told I could turn in Markdown code... Let's find out!