Mandatory Exercise 1

Børge Olav Haug

September 20, 2018

Problem 1

a) $P(\text{At least four visitors arrive during one minute}) = P(N \geq 4) = 1 - P(N < 4) = 1 - P(N \leq 3) = 1 - \sum_{n=0}^{3} p(n;\lambda) = 1 - \sum_{n=0}^{3} \frac{e^{-\lambda t}(\lambda t)^n}{n!} = 1 - \sum_{n=0}^{3} \frac{e^{-3}3^n}{n!} = 1 - e^{-3}\sum_{n=0}^{3} \frac{3^n}{n!} = 1 - e^{-3}(1 + 3 + \frac{9}{2} + \frac{27}{6}) = 1 - 13e^{-3} \approx 1 - 0.6472 = \underline{\underline{0.3528}}$.

Or, we could've looked up the Poisson distribution in a statistics table, and plugged in $x = 3$ and $\mu = 3$ to find $P(N \leq 3) = 0.6472$ more easily.

$P(\text{A visitor spends more than 5 minutes on the website}) = P(X_i > 5) = 1 - P(X_i \leq 5) = 1 - \int_0^5 f(x;\alpha,\beta)dx = 1 - \int_0^5 \frac{1}{\beta^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-x/\beta} dx = 1 - \int_0^5 \frac{1}{3^2\Gamma(2)} x^{2-1} e^{-x/3} dx$.

For positive integers $\Gamma(n) = (n-1)!$, so $\Gamma(2) = 1! = 1$. Then we get

$P(X_i > 5) = 1 - \int_0^5 \frac{1}{9} x e^{-x/3} dx = 1 - \frac{1}{9}\int_0^5 x e^{-x/3} dx$.

Integrating by parts ($\int u\ dv = uv - \int v\ du$), with $u = x,\ du = dx$ and $dv = e^{-x/3}dx,\ v = -3e^{-x/3}$, we get

$P(X_i > 5) = 1 - \frac{1}{9}\int_0^5 xe^{-x/3}dx =$
$1 - \frac{1}{9}(-3[xe^{-x/3}]_0^5 - (-3)\int_0^5 e^{-x/3}dx) = 1 - \frac{1}{9}(-3[xe^{-x/3}]_0^5 - 9[e^{-x/3}]_0^5) =$
$1 - (-\frac{1}{3}[xe^{-x/3}]_0^5 - [e^{-x/3}]_0^5) = 1 - (-\frac{1}{3}(5e^{-5/3} - 0) - (e^{-5/3} - 1)) =$
$1 - (-\frac{8}{3}e^{-5/3} + 1) =$
$\frac{8}{3}e^{-5/3} \approx \underline{\underline{0.5037}}$.

To simulate the distribution of V_t, we want to generate a series of $Nsim$ numbers from this distribution. To achieve this we can do the following:

1. Generate one random number N from the Poisson distribution (with parameter $\lambda = \lambda t$).
2. Then we generate the X_i by generating N numbers from the gamma distribution (with parameters α and β).
3. Then, to get V_t, we add the X_i together.
4. To get $Nsim$ numbers we just repeat the steps above $Nsim$ times, giving us a vector containing $Nsim$ values for V_t.
5. To estimate $E(V_t)$, we simply add the $Nsim$ numbers together and divide the result by $Nsim$. In R, this can be done by using the `mean` function on the vector containing containing the values for V_t.
6. To estimate $P(V_t > a)$, we count how many times V_t is larger than a, and divide the result by $Nsim$. In R, this can be done by using the `mean` function on the expression $V > a$, where V is the vector containing the values for V_t.

b) See R-code

Problem 2

a) Let U be uniformly distributed on $[0, 1]$. If X follows a distribution of interest and $F(X)$ has an inverse that is easy to find, we can generate a number from this distribution by setting $U = F(X)$, generating a number U, and then applying the inverse transform $F^{-1}(U)$ to get a value for X.

First, let $U = F(X)$. The cumulative distribution function for the Weibull distribution is given by [4, chapter 6.10, p. 203]:

$F(x) = 1 - e^{-\alpha x^\beta}$, for $x \geq 0, \alpha > 0,\ \beta > 0$.

So $U = 1 - e^{-\alpha X^\beta}$. We find the inverse $F^{-1}(U)$ by solving $U = F(X)$ for X:

$U = 1 - e^{-\alpha X^\beta}$

$\Leftrightarrow e^{-\alpha X^\beta} = 1 - U$

$\Leftrightarrow ln(e^{-\alpha X^\beta}) = ln(1 - U)$

$\Leftrightarrow -\alpha X^\beta = ln(1 - U)$

$\Leftrightarrow X^\beta = -\frac{ln(1-U)}{\alpha}$

$\Leftrightarrow ln(X^\beta) = ln(-\frac{ln(1-U)}{\alpha})$

$\Leftrightarrow \beta ln(X) = ln(-\frac{ln(1-U)}{\alpha})$

$\Leftrightarrow ln(X) = \frac{ln(-\frac{ln(1-U)}{\alpha})}{\beta}$

$\Leftrightarrow e^{ln(X)} = e^{\frac{ln(-\frac{ln(1-U)}{\alpha})}{\beta}}$

$\Leftrightarrow X = e^{\frac{ln(-\frac{ln(1-U)}{\alpha})}{\beta}} = e^{ln((-\frac{ln(1-U)}{\alpha})^{\frac{1}{\beta}})} = (-\frac{ln(1-U)}{\alpha})^{\frac{1}{\beta}}$

$\Rightarrow X = F^{-1}(U) = \underline{\underline{(-\frac{ln(1-U)}{\alpha})^{\frac{1}{\beta}}}}$

The algorithm for generating $Nsim$ numbers from the Weibull-distribution then becomes:

1. Generate a number u using the uniform random number generator.
2. Deliver $x = F^{-1}(u) = (-\frac{ln(1-u)}{\alpha})^{\frac{1}{\beta}}$.
3. Repeat the steps above $Nsim$ times.

Or, to avoid for-loops in R:

1. Generate a vector u of size $Nsim$ using the uniform random number generator.
2. Apply $F^{-1}(u) = (-\frac{ln(1-u)}{\alpha})^{\frac{1}{\beta}}$ and return the result.

b) See R-code

c) When $\beta = 1$, $f(x)$ becomes:

$\alpha \cdot 1 \cdot x^{1-1} e^{-\alpha x^1} = \alpha e^{-\alpha x}$,

which is the exponential distribution with parameter $\beta = \frac{1}{\alpha}$ (or $\lambda = \alpha$, depending on the notation). The cumulative distribution function for the exponential distribution is the same as for the Weibull distribution, but with $\beta = 1$: $F(x) = 1 - e^{-\alpha x}$

$P(\text{One pump fails before one year}) = P(X_i < 12) = 1 - e^{-\alpha \cdot 12} =$

$1 - e^{-0.08 \cdot 12} = 1 - e^{-0.96} \approx \underline{\underline{0.6171}}$

i) Operational situation 1: It is sufficient that one pump functions (i.e. the system fails when all pumps have failed):

$P(\text{The pump system fails before one year}) =$

$P(\text{The last pump to fail fails before one year}) =$

$P(\max(X_1, X_2, X_3)) < 12) =$

$P((X_1 < 12) \cap (X_2 < 12) \cap (X_3 < 12)) \underset{independency}{=}$

$P(X_1 < 12) \cdot P(X_2 < 12) \cdot P(X_3 < 12) =$

$(1 - e^{-0.96})^3 \approx \underline{\underline{0.2350}}$ [3, p. 1]

General formula:
$P(\text{The last pump to fail fails before one year}) = (1 - e^{-\alpha \cdot 12})^3$

ii) Operational situation 2: All pumps have to function (i.e. the system fails when the first pump fails):

$P(\text{The pump system fails before one year}) =$

$P(\text{The first pump to fail fails before one year}) =$

$P(\min(X_1, X_2, X_3) < 12)) =$

$1 - P(\min(X_1, X_2, X_3) > 12)) =$

$1 - P((X_1 > 12) \cap (X_2 > 12) \cap (X_3 > 12)) \underset{independency}{=}$

$1 - P(X_1 > 12) \cdot P(X_2 > 12) \cdot P(X_3 > 12) =$

$1 - (1 - P(X_1 < 12)) \cdot (1 - P(X_2 < 12)) \cdot (1 - P(X_3 < 12)) =$

$1 - (1 - (1 - e^{-0.96}))^3 = 1 - (e^{-0.96})^3 = 1 - e^{-2.88} \approx \underline{\underline{0.9439}}$ [3, p. 1]

General formula:

P(The first pump to fail fails before one year) =
$1 - (e^{-\alpha \cdot 12})^3 = 1 - (e^{-\alpha \cdot 36})$

iii) Operational situation 3: 2 out of 3 pumps have to function (i.e. the system fails when at least 2 of the pumps have failed):

P(The pump system fails before one year) =

P(2 or more pumps fail before one year) =

P(All pumps OR

pump 1 AND 2, but NOT 3 OR

pump 1 AND 3, but NOT 2 OR

pump 2 AND 3, but NOT 1, fail before one year) =

$P(((X_1 < 12) \cap (X_2 < 12) \cap (X_3 < 12)) \cup$

$((X_1 < 12) \cap (X_2 < 12) \cap (X_3 > 12)) \cup$

$((X_1 < 12) \cap (X_2 > 12) \cap (X_3 < 12)) \cup$

$((X_1 > 12) \cap (X_2 < 12) \cap (X_3 < 12))) \underset{mutually\ exclusive}{=}$

$P((X_1 < 12) \cap (X_2 < 12) \cap (X_3 < 12)) +$

$P((X_1 < 12) \cap (X_2 < 12) \cap (X_3 > 12)) +$

$P((X_1 < 12) \cap (X_2 > 12) \cap (X_3 < 12)) +$

$P((X_1 > 12) \cap (X_2 < 12) \cap (X_3 < 12)) \underset{independency}{=}$

$P(X_1 < 12) \cdot P(X_2 < 12) \cdot P(X_3 < 12) +$

$P(X_1 < 12) \cdot P(X_2 < 12) \cdot P(X_3 > 12) +$

$P(X_1 < 12) \cdot P(X_2 > 12) \cdot P(X_3 < 12) +$

$P(X_1 > 12) \cdot P(X_2 < 12) \cdot P(X_3 < 12) =$

$P(X_i < 12)^3 + 3 \cdot P(X_i < 12)^2 \cdot P(X_i > 12) =$

$P(X_i < 12)^3 + 3 \cdot P(X_i < 12)^2 \cdot (1 - P(X_i < 12)) =$

$(1 - e^{-0.96})^3 + 3 \cdot (1 - e^{-0.96})^2 \cdot e^{-0.96} \approx \underline{\underline{0.6724}}$

General formula:

$P(\text{2 or more pumps fail before one year}) =$
$(1 - e^{-\alpha \cdot 12})^3 + 3 \cdot (1 - e^{-\alpha \cdot 12})^2 \cdot e^{-\alpha \cdot 12}$

d) See R-code

e) To simulate the distribution of the probability that the system fails within one year, we can do the following:

1. Use the triangle distribution with parameters $x_{min} = 10$, $x_{mode} = 20$ and $x_{max} = 50$ to generate $Nsim$ expected lifetimes.
2. We know from c) that when $\beta = 1$ the Weibull-distribution becomes the exponential distribution with parameter β equal to $\frac{1}{\alpha}$. For the exponential distribution $E(X) = \beta$, and in this case, since $\beta = \frac{1}{\alpha}$, $E(X) = \frac{1}{\alpha}$. Therefore, $\alpha = \frac{1}{E(X)}$, and we can use this to convert the $Nsim$ expectations in step 1 into $Nsim$ α values.
3. Now we can plug the α values from step 2 into the calculations done in c) to get $Nsim$ probability calculations.
   i) To simulate the distribution of
   $P(\text{The last pump to fail fails before one year})$,
   plug the α values into this formula:
   $(1 - e^{-\alpha \cdot 12})^3$
   ii) To simulate the distribution of
   $P(\text{The first pump to fail fails before one year})$,
   plug the α values into this formula:
   $1 - (e^{-\alpha \cdot 12})^3 = 1 - (e^{-\alpha \cdot 36})$
   iii) To simulate the distribution of
   $P(\text{2 or more pumps fail before one year})$,

plug the α values into this formula:
$(1 - e^{-\alpha \cdot 12})^3 + 3 \cdot (1 - e^{-\alpha \cdot 12})^2 \cdot e^{-\alpha \cdot 12}$

f) See R-code

Problem 3

a) We want to simulate the upper section Yatzy game $Nsim$ times. We therefore need:

1. a vector of size $Nsim$ to store all the scores.
2. a for-loop to run the game $Nsim$ times.
3. to keep track of the current game score and update it when simulating the rounds and dice throws.
4. to keep track of what game we're at so we can store the score at the correct position in the vector.

For every game we want to simulate 6 rounds. We therefore need:

1. a for-loop to run 6 rounds.
2. to keep track of what round we're at, because it determines which side of the dice we're interested in and by how much we'll increase the score.

For every round we want to simulate up to 3 dice throws. We therefore need:

1. a for-loop to run up to 3 dice throws.
2. a random number generator that can generate numbers from 1 to 6 in order to simulate the dice throws.
3. to keep track of how many dice we've put aside, because it determines how many numbers we need to draw per dice throw, and if we need to do the second and third dice throw.

From this we can write the following pseudo-code for generating the upper section Yatzy scores:

```
init scores = new vector(size=Nsim):
for i from 1 to Nsim:  // for each game
  init score = 0
  for j from 1 to 6 // for each round
    init n_aside = 0 // number of dice put aside
    for k from 1 to 3:  // for each throw
      values =
        draw (5-n_aside) numbers from discrete U(1,6)
      score += j·(#values showing j)
      n_aside += (#values showing j)
      if n_aside == 5:
        break innermost loop // Proceed to next round
  scores[i] = scores
```

To estimate the probability of getting a score of at least 42 we simply count the number of values in the vector larger than 42, and divide this count by $Nsim$.

When counting the number of successes (a success here being finding a value larger than 42) and dividing by the number of observations (simulations), we can find the number of simulations necessary to be 95% confident that the result doesn't stray more than $e = 0.01$ from the true value by using the formula [1, chap 2_part2, p. 15]:

$n > \frac{1}{e^2}$

$\Rightarrow n > \frac{1}{0.01^2} \Rightarrow n > 10000$

With 10000 simulations, we can be 95 % sure that there is an error of at most 0.01 in the estimated probability.

b) See R-code

References

[1] Jan Terje Kvaløy, *Notes, lecture slides, and code examples from the course STA510 Statistical modelling and simulation*, 2018

[2] Maria L. Rizzo, *Statistical Computing with R*, Chapman & Hall/CRC, London, 1st Edition, 2007

[3] Stian Lydersen, NTNU, and Jan Terje Kvaløy, UiS, *Note on order variables and extreme variables*

[4] Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, and Keying Ye *Probability & Statistics for Engineers and Scientists*, Pearson Education, Inc., Boston, 9th Edition, 2012