UNIVERSIDADE FEDERAL DE UBERLÂNDIA – UFU

Graduação em Ciência da Computação

Atividade Prática 03

GBC065 – Modelagem e Simulação

Uberlândia
2018

Faculdade de Computação

UFU

Atividade Prática 03

Trabalho apresentado à disciplina de Modelagem e Simulação (GBC065), ministrada pelo professor Anderson Rodrigues dos Santos, para o curso de Bacharelado em Ciência da Computação, no período 2018-2, na Universidade Federal de Uberlândia.

Grupo 02 – Integrantes:

Antonio Carlos Neto
11611BCC054

Ronistone Gonçalves dos Reis Júnior
11521BCC018

Uberlândia
2018

Exercise 3.1.2:

(a) Relative to the steady-state statistics in Example 3.1.3 and the statistical equations in Section 1.2, list all of the consistency checks that should be applicable.
R:

- Consistency Checks:
  - $(\bar{w}) = (\bar{d}) + (\bar{s})$;
  - $(\bar{l}) = (\bar{q}) + (\bar{x})$;

- Bônus:
  - $(\bar{r}) = a_n/n$;
  - $(\bar{l}) = (n/c_n)*(\bar{w})$;
  - $(\bar{q}) = (n/c_n)*(\bar{d})$;
  - $(\bar{x}) = (n/c_n)*(\bar{s})$;

(b) Verify that all of these consistency checks are valid.
R:

- Consistency Checks:
  - $(\bar{w}) = (\bar{d}) + (\bar{s})$ => 3.83 = 2.33 + 1.50 ✓
  - $(\bar{l}) = (\bar{q}) + (\bar{x})$ => 1.92 = 1.17 + 0.75 ✓

Exercise 3.1.4:

(a) Conduct a transition-to-steady-state study like that in Example 3.1.3 except for a service time model that is Uniform(1.3, 2.3). Be specific about the number of jobs that seem to be required to produce steady-state statistics.
R:

- Initial Seed = 12345, Número de Jobs = 1 000, Amostra = 20 ;

$(\bar{r})$	$(\bar{w})$	$(\bar{d})$	$(\bar{s})$	$(\bar{l})$	$(\bar{q})$	$(\bar{x})$
1.98	9.33	7.53	1.80	4.64	3.74	0.90

- Initial Seed = 54321, Número de Jobs = 1 000, Amostra = 20 ;

$(\bar{r})$	$(\bar{w})$	$(\bar{d})$	$(\bar{s})$	$(\bar{l})$	$(\bar{q})$	$(\bar{x})$
1.92	10.99	9.20	1.79	5.67	4.75	0.93

- Initial Seed = 2121212, Número de Jobs = 1 000, Amostra = 20 ;

$(\bar{r})$	$(\bar{w})$	$(\bar{d})$	$(\bar{s})$	$(\bar{l})$	$(\bar{q})$	$(\bar{x})$
2.02	10.50	8.70	1.80	5.16	4.28	0.89

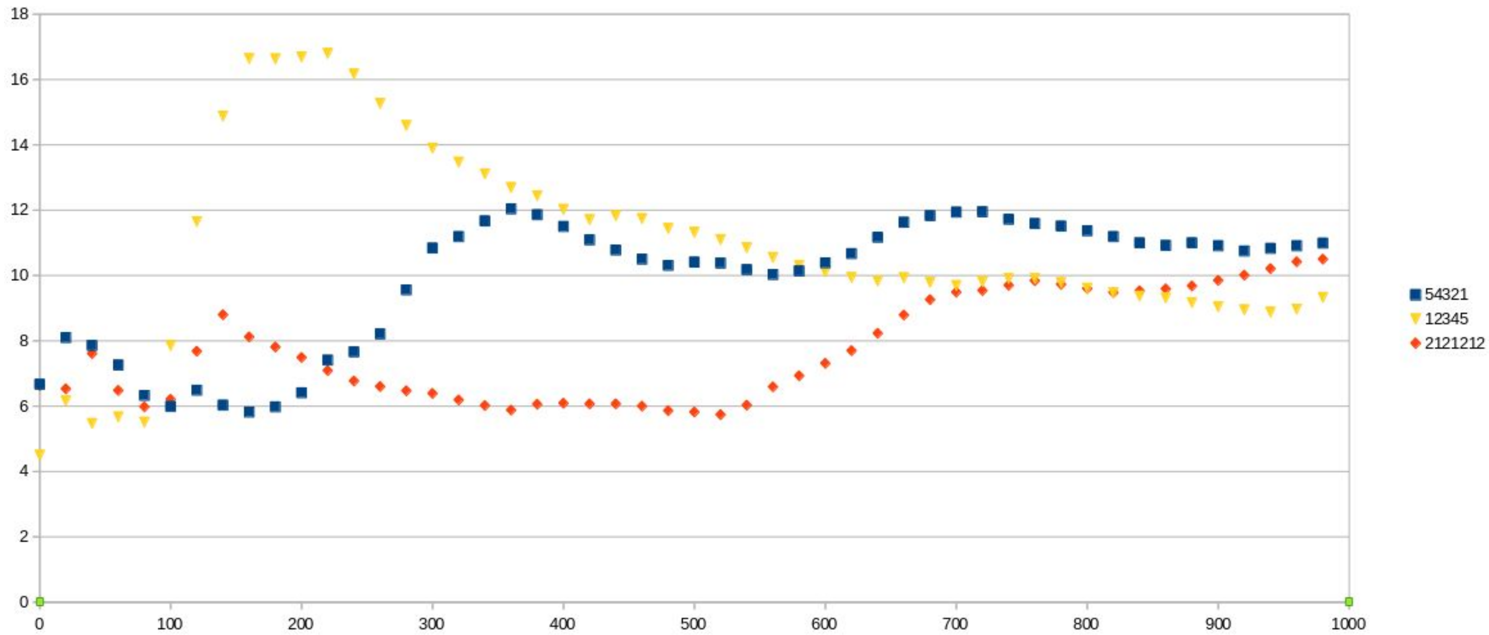


GRÁFICO 1

- Initial Seed = 12345, Número de Jobs = 4 000 000, Amostra = 2 000 ;

$(\bar{r})$	$(\bar{w})$	$(\bar{d})$	$(\bar{s})$	$(\bar{l})$	$(\bar{q})$	$(\bar{x})$
2.00	10.12	8.32	1.80	5.06	4.16	0.90

- Initial Seed = 54321, Número de Jobs = 4 000 000, Amostra = 2 000 ;

$(\bar{r})$	$(\bar{w})$	$(\bar{d})$	$(\bar{s})$	$(\bar{l})$	$(\bar{q})$	$(\bar{x})$
2.00	10.12	8.32	1.80	5.06	4.16	0.90

- Initial Seed = 2121212, Número de Jobs = 4 000 000, Amostra = 2 000 ;

$(\bar{r})$	$(\bar{w})$	$(\bar{d})$	$(\bar{s})$	$(\bar{l})$	$(\bar{q})$	$(\bar{x})$
2.00	10.13	8.33	1.80	5.06	4.16	0.90

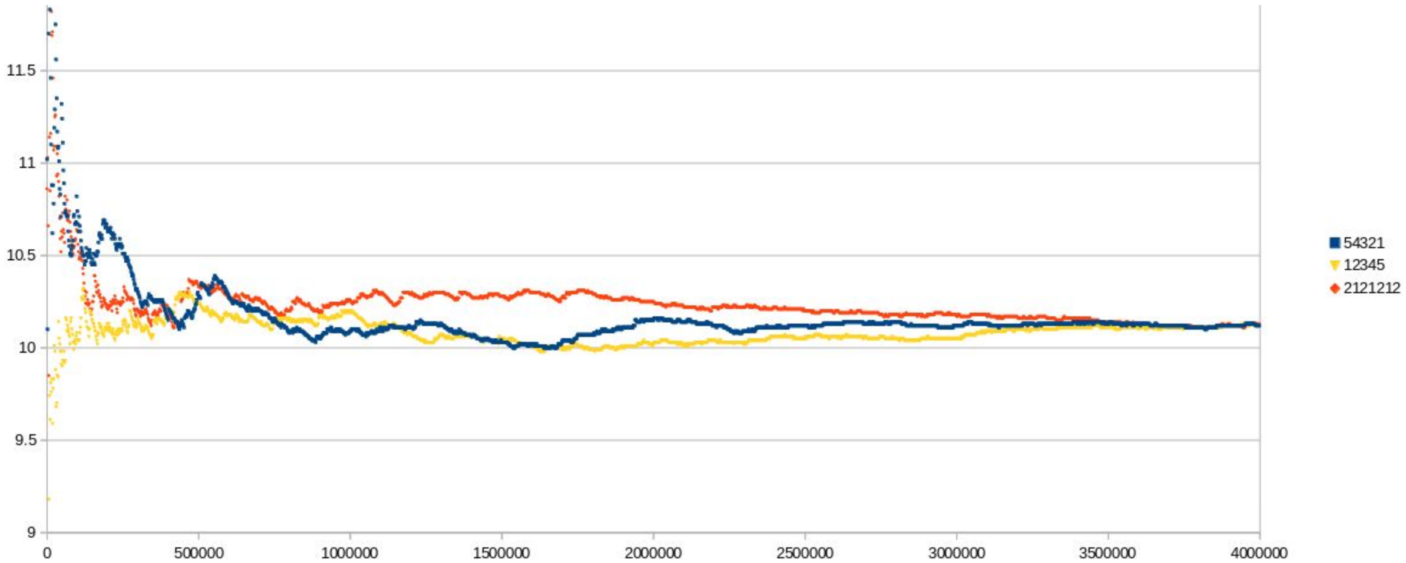


GRÁFICO 2

(b) Comment.
R: Pelo gráfico 1 percebemos que as 3 diferentes **initial seed** convergem para um entre 8 e 12, e a *initial seed* 12345, possui uma variabilidade maior que as outras 2. Pelo gráfico 2 percebemos que eles convergem para um valor aproximadamente de 10.15, e que a *initial seed* 54321 demora mais para aproximar do valor estável. Percebemos que na faixa de 500.000 jobs a média do tempo de wait torna bem próximo do estável.

Exercise 3.1.5:

Considerando o número de Jobs = 10.000, e Initial Seed = 123456789, e fazendo as seguintes alterações:

```c
long Geometric(double p)
{
  return ((long) (log(1.0 - Random()) /log(p))); /* use 0.0 < p < 1.0 */
}
```

```c
double GetService(void)
{
  long k;
  double sum = 0.0;
  long tasks = 1 + Geometric(0.9);

  for (k = 0; k < tasks; k++)
    sum += Uniform(0.1,0.2);
  return (sum);
}
```

- Temos as seguintes medidas:

```
for 10000 jobs
   average interarrival time =   2.00
   average wait ............ =   6.02
   average delay ........... =   4.53
   average service time .... =   1.49
   average # in the node ... =   3.02
   average # in the queue .. =   2.27
   utilization ............. =   0.75
```

(a) Verify that the mean service time in Example 3.1.4 is 1.5.
R: A média do tempo de serviço é 1.49, aproximadamente 1.5.

(b) Verify that the steady-state statistics in Example 3.1.4 seem to be correct.
R: Aumentando o números de jobs chegamos no **steady-state.**

```
for 100000000 jobs
   average interarrival time =   2.00
   average wait ............ =   5.79
   average delay ........... =   4.29
   average service time .... =   1.50
   average # in the node ... =   2.89
   average # in the queue .. =   2.14
   utilization ............. =   0.75
```

- 2.00 = 2.00 ✓
- 5.77 ≅ 5.79 ✓
- 4.27 ≅ 4.29 ✓
- 1.50 = 1.50 ✓
- 2.89 = 2.89 ✓
- 2.14 = 2.14 ✓
- 0.75 = 0.75 ✓

(c) Note that the arrival rate, service rate, and utilization are the same as those in Example 3.1.3, yet all the other statistics are larger than those in Example 3.1.3. Explain (or conjecture) why this is so. Be specific.
R:

- Arrival rate: como não alteramos a função **"double GetArrival(void)"** e nem a **"double Exponential(double m)"** o valor de chegada de todos processos serão o mesmo, ou seja, o tempo entre chegadas será o mesmo.
- Service rate: Por uma coincidência, os valores de tempo de serviço gerados, são basicamente os mesmos, gerando pelas duas formas.
- Utilization: como o service rate é o mesmo, temos que o service node irá ficar ocupado da mesma forma, considerando um grande números de jobs.
    - $(\bar{x}) = (n/c_n)/(\bar{s})$;

Exercise 3.1.6:

(a) Modify program sis2 to compute data like that in Example 3.1.7. Use the functions PutSeed and GetSeed from the library rng in such a way that one initial seed is supplied by the system clock, printed as part of the program's output and used automatically to generate the same demand sequence for all values of s.
R: Quando colocamos PutSeed(-1), ele gera automaticamente de acordo com o relógio do computador.

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (1, 80)

```
average order ............ =  31.35
setup frequency .......... =   0.33
average holding level .... =  35.47
average shortage level ... =   1.51
Dependent Cost ... = 2273.71
```

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (2, 80)

```
average order ............ =  31.35
setup frequency .......... =   0.33
average holding level .... =  35.18
average shortage level ... =   1.55
Dependent Cost ... = 2291.56
```

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (3, 80)

average order .... = 31.35
setup frequency .... = 0.33
average holding level .... = 35.60
average shortage level ... = 1.55
Dependent Cost ... = 2302.11

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (4, 80)

average order .... = 31.35
setup frequency .... = 0.33
average holding level .... = 35.60
average shortage level ... = 1.55
Dependent Cost ... = 2302.11

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (5, 80)

average order .... = 31.35
setup frequency .... = 0.34
average holding level .... = 36.92
average shortage level ... = 1.31
Dependent Cost ... = 2180.30

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (6, 80)

average order .... = 31.35
setup frequency .... = 0.36
average holding level .... = 36.10
average shortage level ... = 0.86
Dependent Cost ... = 1863.56

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (7, 80)

average order .... = 31.35
setup frequency .... = 0.36
average holding level .... = 36.37
average shortage level ... = 0.85
Dependent Cost ... = 1861.34

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (8, 80)

average order .... = 31.35
setup frequency .... = 0.36
average holding level .... = 36.99
average shortage level ... = 0.87
Dependent Cost ... = 1890.99

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (9, 80)

average order .... = 31.35
setup frequency .... = 0.36
average holding level .... = 38.00
average shortage level ... = 0.92
Dependent Cost ... = 1950.55

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (10, 80)

average order .... = 31.35
setup frequency .... = 0.36
average holding level .... = 38.46
average shortage level ... = 0.85
Dependent Cost ... = 1916.02

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (11, 80)

average order .... = 31.35
setup frequency .... = 0.36
average holding level .... = 38.89
average shortage level ... = 0.85
Dependent Cost ... = 1930.10

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (12, 80)

average order .... = 31.35
setup frequency .... = 0.36

average holding level .... = 38.89
average shortage level ... = 0.85
Dependent Cost ... = 1930.10

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (13, 80)

average order .... = 31.35
setup frequency .... = 0.36
average holding level .... = 39.24
average shortage level ... = 0.86
Dependent Cost ... = 1943.40

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (14, 80)

average order .... = 31.35
setup frequency .... = 0.37
average holding level .... = 40.14
average shortage level ... = 0.70
Dependent Cost ... = 1861.69

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (15, 80)

average order .... = 31.35
setup frequency .... = 0.39
average holding level .... = 41.38
average shortage level ... = 0.55
Dependent Cost ... = 1806.34

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (16, 80)

average order .... = 31.35
setup frequency .... = 0.40
average holding level .... = 42.28
average shortage level ... = 0.37
Dependent Cost ... = 1716.44

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (17, 80)

average order .... = 31.35
setup frequency .... = 0.40
average holding level .... = 42.71
average shortage level ... = 0.38
Dependent Cost ... = 1734.38

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (18, 80)

average order .... = 31.35
setup frequency .... = 0.40
average holding level .... = 42.71
average shortage level ... = 0.38
Dependent Cost ... = 1734.38

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (19, 80)

average order .... = 31.35
setup frequency .... = 0.40
average holding level .... = 42.71
average shortage level ... = 0.38
Dependent Cost ... = 1734.38

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (20, 80)

average order .... = 31.35
setup frequency .... = 0.41
average holding level .... = 43.72
average shortage level ... = 0.21
Dependent Cost ... = 1650.94

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (21, 80)

average order .... = 31.35
setup frequency .... = 0.41
average holding level .... = 44.38
average shortage level ... = 0.17
Dependent Cost ... = 1637.00

for 100 time intervals with an average demand of 31.35

and policy parameters (s, S) = (22, 80)

average order .... = 31.35
setup frequency .... = 0.42
average holding level .... = 44.97
average shortage level ... = 0.20
Dependent Cost ... = 1686.86

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (23, 80)

average order .... = 31.35
setup frequency .... = 0.42
average holding level .... = 44.97
average shortage level ... = 0.20
Dependent Cost ... = 1686.86

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (24, 80)

average order .... = 31.35
setup frequency .... = 0.42
average holding level .... = 45.16
average shortage level ... = 0.16
Dependent Cost ... = 1660.97

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (25, 80)

average order .... = 31.35
setup frequency .... = 0.42
average holding level .... = 45.16
average shortage level ... = 0.16
Dependent Cost ... = 1660.97

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (26, 80)

average order .... = 31.35
setup frequency .... = 0.43
average holding level .... = 46.11
average shortage level ... = 0.12
Dependent Cost ... = 1665.20

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (27, 80)

average order .... = 31.35
setup frequency .... = 0.43
average holding level .... = 46.11
average shortage level ... = 0.12
Dependent Cost ... = 1665.20

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (28, 80)

average order .... = 31.35
setup frequency .... = 0.44
average holding level .... = 46.15
average shortage level ... = 0.12
Dependent Cost ... = 1675.67

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (29, 80)

average order .... = 31.35
setup frequency .... = 0.44
average holding level .... = 46.15
average shortage level ... = 0.12
Dependent Cost ... = 1675.67

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (30, 80)

average order .... = 31.35
setup frequency .... = 0.44
average holding level .... = 46.15
average shortage level ... = 0.12
Dependent Cost ... = 1675.67

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (31, 80)

average order .... = 31.35
setup frequency .... = 0.46
average holding level .... = 47.31
average shortage level ... = 0.07

Dependent Cost ... = 1689.75

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (32, 80)

average order .... = 31.35
setup frequency .... = 0.47
average holding level .... = 48.03
average shortage level ... = 0.03
Dependent Cost ... = 1694.90

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (33, 80)

average order .... = 31.35
setup frequency .... = 0.47
average holding level .... = 47.92
average shortage level ... = 0.03
Dependent Cost ... = 1691.80

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (34, 80)

average order .... = 31.35
setup frequency .... = 0.47
average holding level .... = 48.33
average shortage level ... = 0.02
Dependent Cost ... = 1693.87

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (35, 80)

average order .... = 31.35
setup frequency .... = 0.47
average holding level .... = 48.46
average shortage level ... = 0.02
Dependent Cost ... = 1697.42

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (36, 80)

average order .... = 31.35
setup frequency .... = 0.49
average holding level .... = 49.28
average shortage level ... = 0.01
Dependent Cost ... = 1727.15

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (37, 80)

average order .... = 31.35
setup frequency .... = 0.50
average holding level .... = 49.61
average shortage level ... = 0.00
Dependent Cost ... = 1740.95

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (38, 80)

average order .... = 31.35
setup frequency .... = 0.52
average holding level .... = 50.32
average shortage level ... = 0.01
Dependent Cost ... = 1782.64

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (39, 80)

average order .... = 31.35
setup frequency .... = 0.54
average holding level .... = 51.16
average shortage level ... = 0.01
Dependent Cost ... = 1823.28

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (40, 80)

average order .... = 31.35
setup frequency .... = 0.56
average holding level .... = 51.70
average shortage level ... = 0.01
Dependent Cost ... = 1856.69

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (41, 80)

average order .... = 31.35
setup frequency .... = 0.57
average holding level .... = 52.14
average shortage level ... = 0.01
Dependent Cost ... = 1877.69

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (42, 80)

average order .... = 31.35
setup frequency .... = 0.58
average holding level .... = 52.53
average shortage level ... = 0.00
Dependent Cost ... = 1893.74

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (43, 80)

average order .... = 31.35
setup frequency .... = 0.58
average holding level .... = 52.92
average shortage level ... = 0.00
Dependent Cost ... = 1903.49

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (44, 80)

average order .... = 31.35
setup frequency .... = 0.59
average holding level .... = 53.36
average shortage level ... = 0.00
Dependent Cost ... = 1924.49

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (45, 80)

average order .... = 31.35
setup frequency .... = 0.61
average holding level .... = 54.33
average shortage level ... = 0.00
Dependent Cost ... = 1968.42

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (46, 80)

average order .... = 31.35
setup frequency .... = 0.62
average holding level .... = 54.77
average shortage level ... = 0.00
Dependent Cost ... = 1989.42

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (47, 80)

average order .... = 31.35
setup frequency .... = 0.63
average holding level .... = 55.11
average shortage level ... = 0.00
Dependent Cost ... = 2007.92

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (48, 80)

average order .... = 31.35
setup frequency .... = 0.66
average holding level .... = 56.45
average shortage level ... = 0.00
Dependent Cost ... = 2071.13

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (49, 80)

average order .... = 31.35
setup frequency .... = 0.66
average holding level .... = 56.45
average shortage level ... = 0.00
Dependent Cost ... = 2071.13

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (50, 80)

average order .... = 31.35
setup frequency .... = 0.67
average holding level .... = 56.88
average shortage level ... = 0.00
Dependent Cost ... = 2092.13

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (51, 80)

average order .... = 31.35
setup frequency .... = 0.68
average holding level .... = 57.37
average shortage level ... = 0.00
Dependent Cost ... = 2114.13

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (52, 80)

average order .... = 31.35
setup frequency .... = 0.69
average holding level .... = 57.84
average shortage level ... = 0.00
Dependent Cost ... = 2136.13

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (53, 80)

average order .... = 31.35
setup frequency .... = 0.71
average holding level .... = 58.41
average shortage level ... = 0.00
Dependent Cost ... = 2170.13

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (54, 80)

average order .... = 31.35
setup frequency .... = 0.74
average holding level .... = 59.22
average shortage level ... = 0.00
Dependent Cost ... = 2220.38

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (55, 80)

average order .... = 31.35
setup frequency .... = 0.78
average holding level .... = 60.26
average shortage level ... = 0.00
Dependent Cost ... = 2286.38

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (56, 80)

average order .... = 31.35
setup frequency .... = 0.78
average holding level .... = 60.26
average shortage level ... = 0.00
Dependent Cost ... = 2286.38

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (57, 80)

average order .... = 31.35
setup frequency .... = 0.79
average holding level .... = 60.63
average shortage level ... = 0.00
Dependent Cost ... = 2305.63

for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (58, 80)

average order .... = 31.35
setup frequency .... = 0.82
average holding level .... = 61.44
average shortage level ... = 0.00
Dependent Cost ... = 2355.88

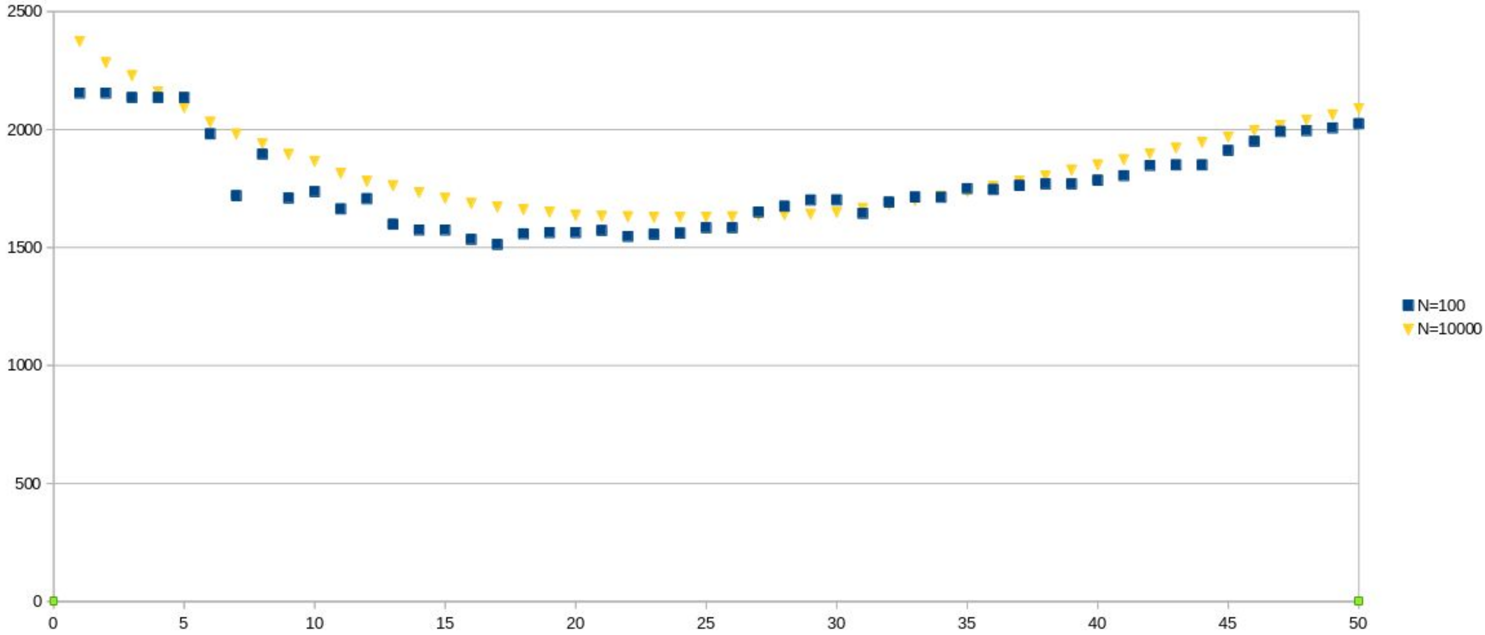
for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (59, 80)

average order .... = 31.35
setup frequency .... = 0.82
average holding level .... = 61.44
average shortage level ... = 0.00
Dependent Cost ... = 2355.88

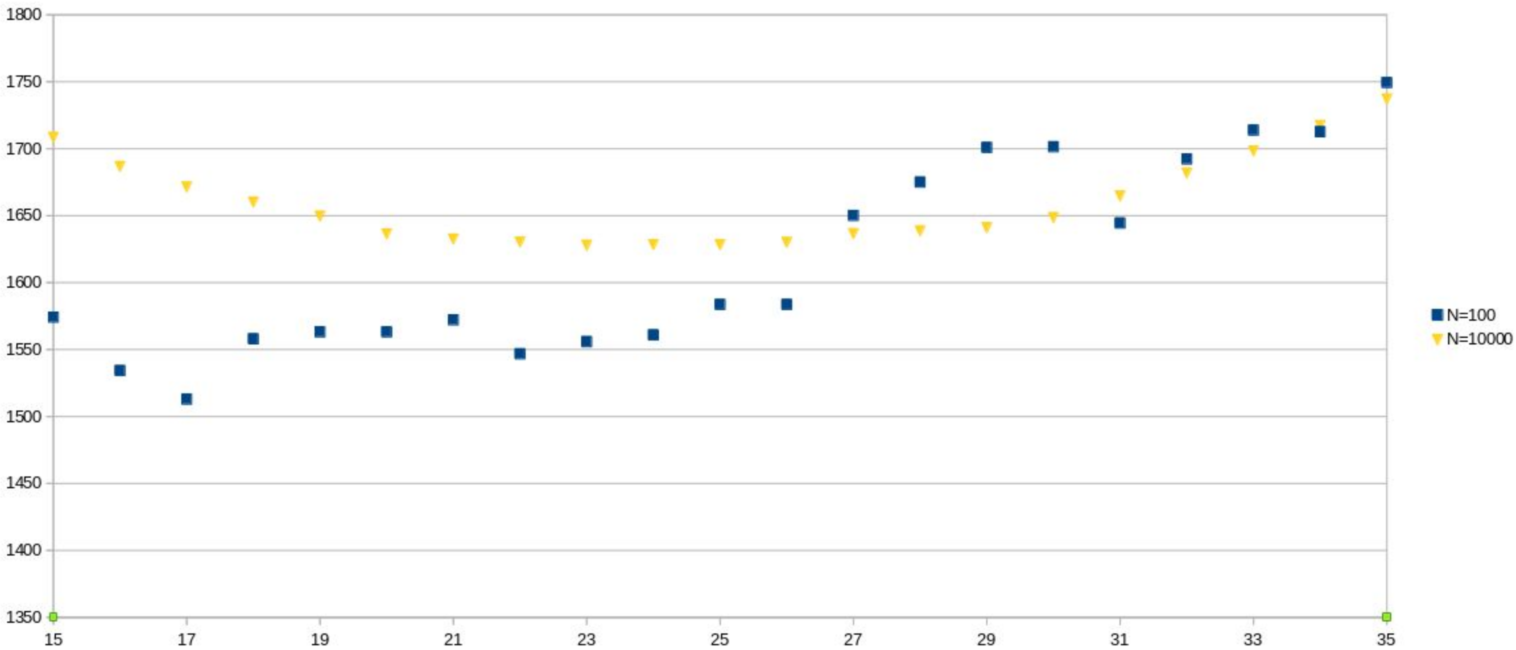
for 100 time intervals with an average demand of 31.35
and policy parameters (s, S) = (60, 80)

average order .... = 31.35
setup frequency .... = 0.83

average holding level .... = 61.73 Dependent Cost ... = 2373.38
average shortage level ... = 0.00



(b) For s = 15, 16, . . . , 35 create a figure (or table) similar to the one in Example 3.1.7.
R:



(c) Comment.

R: Percebemos que quando o s = 17, e o n = 100, temos o menor custo(cost dependent = $1512.96), ou seja, maximizando o lucro total. Quando o n = 10.00, temos que o s = 25 gera o menor cost dependent($1627.73), ou seja, ele gera o menor custo nesse intervalo. Analisando quando o n = 10.000, existe uma menor variação.

Exercise 3.1.7

(a) Relative to Example 3.1.5, if instead the random variate sequence of demands are generated as

di = Equilikely(5, 25) + Equilikely(5, 25)   i = 1, 2, 3, . . .

then, when compared with those in Example 3.1.6, demonstrate that some of the steady-state statistics will be the same and others will not.
R:

- **Example 3.1.6:**

Example 3.1.6 If the *Equilikely*(a, b) demand parameters are set to $(a, b) = (10, 50)$ so that the average demand per time interval is $(a + b)/2 = 30$, then with $(s, S) = (20, 80)$ the (approximate) steady-state statistics program `sis2` will produce are

$\bar{d}$	$\bar{o}$	$\bar{u}$	$\bar{l}^+$	$\bar{l}^-$
30.00	30.00	0.39	42.86	0.26

As illustrated in Figure 3.1.3 (using the same three `rng` initial seeds used in Example 3.1.3) for the average inventory level $\bar{l} = \bar{l}^+ - \bar{l}^-$, at least several hundred time intervals must be simulated to approximate these steady-state statistics. To produce Figure 3.1.3, program `sis2` was modified to print the accumulated value of $\bar{l}$ every 5 time intervals.

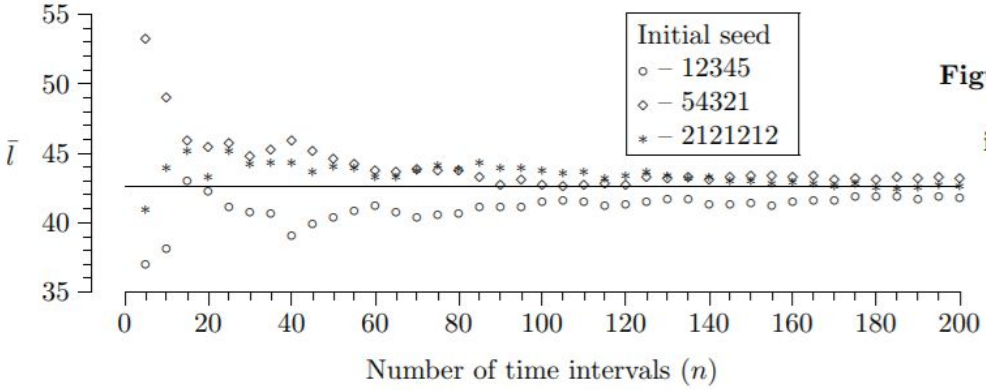


Figure 3.1.3. Average inventory level.

- **Equilikely(10, 50):**

```
for 100000000 time intervals with an average demand of  30.00
and policy parameters (s, S) = (20, 80)

   average order ............ =  30.00
   setup frequency .......... =   0.39
   average holding level .... =  42.81
   average shortage level ... =   0.26
```

- **Equilikely(5, 25) + Equilikely(5,25):**

```
for 100000000 time intervals with an average demand of  30.00
and policy parameters (s, S) = (20, 80)

   average order ............ =  30.00
   setup frequency .......... =   0.39
   average holding level .... =  42.60
   average shortage level ... =   0.18
```

Initial Seed 123456789

```
for 100000000 time intervals with an average demand of  30.00
and policy parameters (s, S) = (20, 80)

   average order ............ =  30.00
   setup frequency .......... =   0.39
   average holding level .... =  42.61
   average shortage level ... =   0.18
```

Initial Seed 12345

(b) Explain why this is so.

R: Primeiramente devemos analisar as diferenças e semelhanças de "**Equilikely(5, 25) + Equilikely(5,25)**" e "**Equilikely(10, 50)**", temos que os dois geram uma sequência uniforme, ou seja, todos os números randômicos gerados têm a mesma probabilidade de saírem, e o intervalo dos números gerados são os mesmos, (a+b)/2 = 30 = 15+15. A diferença é que quando temos a soma de duas sequência uniformes, as extremidades, números pequenos e grandes, são menos prováveis, porém, elas possuem a mesma probabilidade, não influenciando na média, que no caso se mantém nos dois exemplos, como existe um limite mínimo e máximo e mantemos a mesma média de compra, é lógico pensar que o número de compras seja o mesmo, já que o estoque começa e termina no máximo. Considerando que existe uma probabilidade dos números intermediários surgirem mais, temos um menor gasto com holding e shortage, porque há uma chance maior de seu estoque atender o pedido, assim pensamos que o estoque irá descer gradualmente,

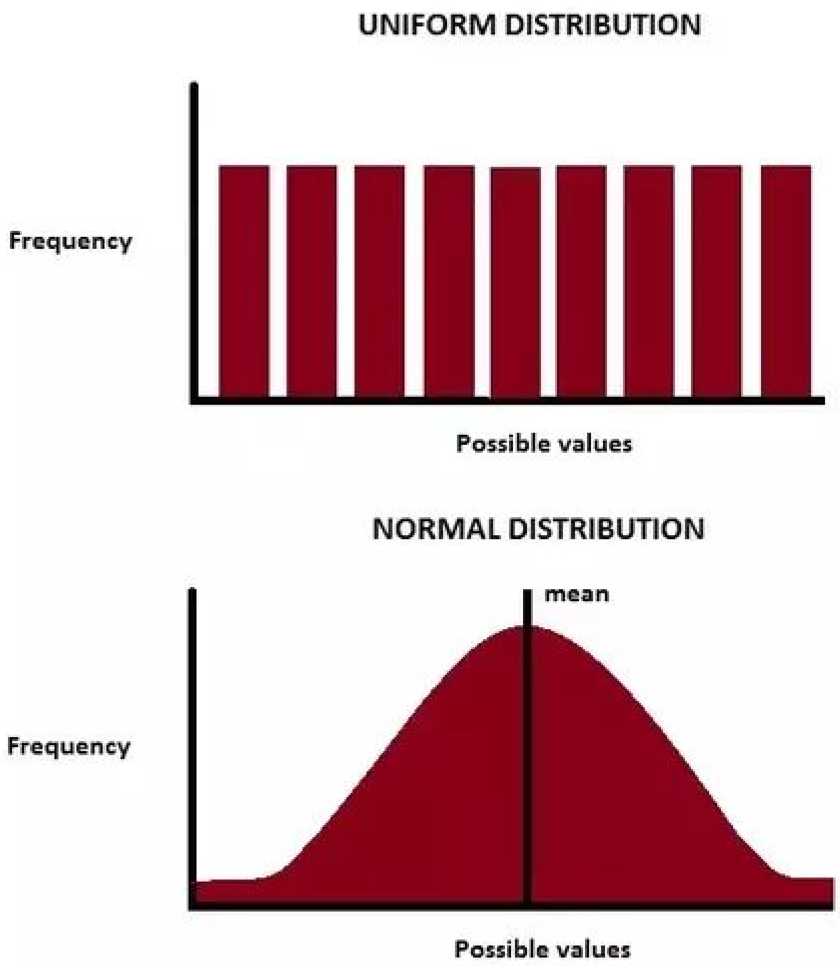
Exercise 3.2.3:

Modify program ssq2 as suggested in Example 3.2.7 to create two programs that differ only in the function GetService. For one of these programs, use the function as implemented in Example 3.2.7; for the other program, use double

```
GetService(void) {
        SelectStream(2);  /* this line is new */
        return (Uniform(0.0, 1.5) + Uniform(0.0, 1.5));
}
```

(a) For both programs verify that exactly the same average interarrival time is produced (print the average with d.dddddd precision). Note that the average service time is approximately the same in both cases, as is the utilization, yet the service nodes statistics $\bar{w}$, d, $\bar{l}$, and $\bar{q}$ are different.
R: Códigos.

(b) Why?
R: Podemos ver que o interarrival não vai se alterar devido ao fato de as arrivals estarem sendo calculados da mesma forma de antes, apenas realizando um desacoplamento da stream. Já para os demais valores temos alterações pois estamos há uma alteração na pdf da geração dos valores, pois como no exemplo geramos o valor de forma uniforme(1.0, 2.0) e no exercício fazemos a alteração para uniforme(0, 1.5) + uniforme(0, 1.5) perdemos a uniformidade da geração pois há maiores possibilidades de valores no meio do intervalo de serem gerados, mas por outro lado devido a isto e ao fato dos valores variarem de 0 a 3.0 e de 1.0 a 2.0, temos assim uma distribuição normal e uma distribuição uniforme, onde a distribuição uniforme possui uma E(X) = (a+b)/2, onde a e b é o limite inferior e superior, enquanto a média da distribuição normal é exatamente o centro da curva, ou seja, o pico da pdf, que para os dois casos são exatamente 1.5. Podemos concluir desta forma que a média de tempo de serviço e a utilização vão tender a serem iguais, já os outros valores variam devido ao fato de como a probabilidade da normal ser maior centro do intervalo causa uma alteração maior pois estes valores são mais sensíveis aos valores do tempo de serviço, que não pode ser visto na média pelo fato de na média valores menores acabam anulando valores maiores.



Exercise 3.3.10

Modifique o programa sis2 para incluir lag na entrega do fornecedor e construir gráficos similares ao Example 3.3.4. Para reproduzir o gráfico sem lag faça com que o Delta (variável que implementa o lag) seja igual a zero e compare os resultados.

R: Não foi possível calcular a fórmula de custo, shortage e holding.