School of Mathematics and Statistics
Te Kura Mātai Tatauranga

DATA304/COMP312/DATA474 **Tutorial 5** **Week 5**

Part 1 — Python

1 We wish to further develop the SimPy simulation model of an M/M/c queueing system from "Bites of SimPy", i.e., start with `q3.py` and make some additions to create `q4.py`.

(a) The idea is to monitor the *number of customers in the system* (call this the *state*). At any *event* (customer arrival, begin service, end service), we need to know if this changes the number of customers in the system, and update the state accordingly.

(1) Use a class variable `n` in the `Arrival` class to keep track of exactly how many customers there are in the system.

(2) Set up a new monitor called `numbermon` in a similar way to `delaymon`.

(3) Modify the `run` method (PEM) within the `Arrival` class so that whenever an event occurs (customer arrival, begin service, end service) we update the state (class variable `Arrival.n`) and then the `numbermon` monitor will `observe` the new value of the state.

(4) Modify the `model` function to return both `W` and `L` in a tuple where

```
L = G.numbermon.timeAverage()
```

See *Lectures on SimPy* (§7.4) for more information about `timeAverage()`.

(5) Add a new list `allL` to the simulation experiment in a similar way to `allW`.

Test your simulation model with $c = 1$, $\lambda = 3$ and $\mu = 5$.

(i) For each performance measure (W and L) produce both a point estimate and a 95% confidence interval from 50 replications.

(ii) *Compare* your results to the expected values of W and L for an M/M/1 queueing system.

(iii) *Determine* whether *Little's Law* appears to hold in your simulation results by forming a confidence interval for λ_{eff}, i.e., each replication will give one estimate of $\lambda_{eff} = \frac{L}{W}$.

(b) Adding to part (a), we wish monitor the *proportion of time the server is busy*. Set up a new monitor called `busymon` in a similar way to `delaymon` and `numbermon`. Modify the `run` method (PEM) within the `Arrival` class so that whenever the state `Arrival.n` is updated, the `busymon` monitor will `observe` a $\boxed{1}$ if there are any customers in the system and a $\boxed{0}$ if there are no customers in the system. Modify the `model` function to return a tuple `(W,L,B)` where `B = G.busymon.timeAverage()` Add a new list `allB` to the simulation experiment in a similar way to `allL` and `allW`. Test your simulation model as in part (a).

pyt260

2 The module `matplotlib` is a Python 2D plotting library which produces publication quality figures. Run the follow fragment of Python code and explain roughly what each of the commands appear to do.

```
import numpy
import matplotlib.pyplot as pl
x = numpy.linspace(-4,4,1000)
y = (1/numpy.sqrt(2*numpy.pi))*numpy.exp(-0.5*x**2)
pl.clf()
pl.plot(x,y)
pl.title("Example: $y=(1/\sqrt{2\pi})e^{-x^2/2}$", fontsize=16)
pl.xlabel("x", fontsize=16)
pl.ylabel("y", fontsize=16)
pl.axis("tight")
pl.savefig("myfig.png")
```

If you do this interactively, no figure will appear; you will need `pl.show()` to display the figure on the screen. pyz133