Problem Statement

Supposed that the Government is planning to open a Volunteer-Run Trap-Neuter-Return (TNR) facility, we would like to know which Queue type I, i∈{Priority Queue (PQ),Shortest Queue (SQ)}, is the most effective one for transporting the cat into the TNR stations at the facility.

Each TNR facility has 3 stations. When a stray cat arrives to the TNR facility, they will queue before entering a TNR station. There are two Queue types, Priority Queue (PQ), Shortest Queue (SQ). For Priority Queue the stray cats will prioritise queue in the West, followed by central then the east. For Shortest Queue the stray cats will enter the station with the lowest queue size at that moment.

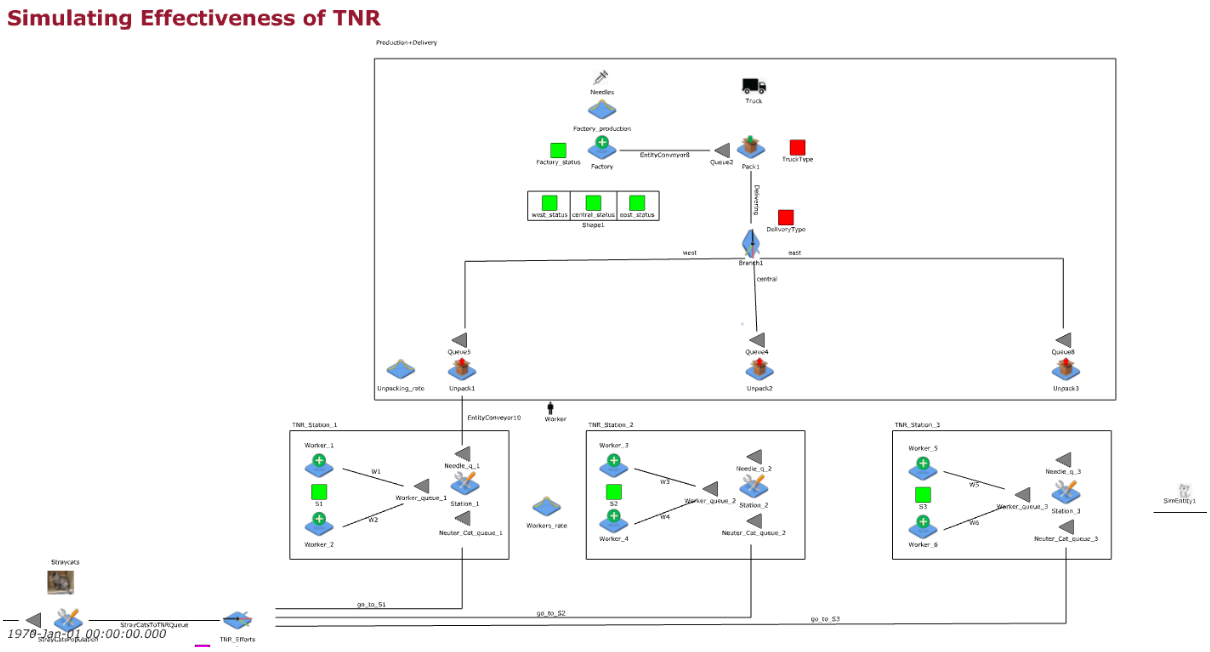
For each station, there are 2 Volunteers. At the start of initialization, both workers are already in queue. Due to the TNR machine limitation, only a single worker may operate it at a time. The machine has a processing rate of about 5 seconds. For the neutering process to occur, the following conditions must be met: 1 volunteer,1 needle and 1 stray must be present in the queues. While one worker operates, the other watches from behind. After a worker has finish operation, he proceeds to take a break. The worker waiting behind moves up the queue and follows the same set of conditions to start operation.

The needles are delivered to the TNR station at a

Supposed that each Volunteers only work 1 hour per day in the TNR facility and Volunteers work at a working rate normally distribution at V~N(μ_V=3 min,σ_V=5 min), Stray Cats gets transported into the TNR stations at a normal distribution of C~N(μ_C=5s,σ_C= 20s)

Needles delivery

Stray cat dying



Output Analysis #1

We would like to compare which queueing type, Priority Queue type or Shortest Queue type is the most efficient one.

Approach

We will do a hypothesis testing with α= 0.05 to see whether there is a difference in the average queueing time between the two different Queue types.

Let μ_{PQ} be the expected average queueing time measure from the Priority Queue type.

Let μ_{SQ} be the expected average queueing time measure from the Shortest Queue type.

Let H_0 be the hypothesis that there is no difference between the expected average queueing time measure from the Priority Queue type and the expected average queueing time measure from the Shortest Queue type.

Let H_1 be the hypothesis that there is a difference between the expected average queueing time measure from the Priority Queue type and the expected average queueing time measure from the Shortest Queue type.

$$H_0: \mu_{PQ} - \mu_{SQ} = 0$$
$$H_1: \mu_{PQ} - \mu_{SQ} \neq 0$$

The average queueing time of stray cats for queueing in each TNR station for both Queue types are run 100 times each, each run lasting for an hour.

Let $x_{i,k,j}$ be the average queueing time of stray cats in respective Queue type i $\in${Priority Queue (PQ), Shortest Queue (SQ)}, Station k $\in${1,2,3}, with run j $\in${1,2,3,….,100}

We then find the average queueing time for each Queueing type across the 3 stations to be $X_{i,j}$ where: $X_{i,j} = x_{i,1,j} + x_{i,2,j} + x_{i,3,j}$

Let $Z_j = X_{PQ,j} - X_{SQ,j}$ to be the difference between the two Queueing types

With Z_j we can identify a confidence interval of $\zeta = \mu_{PQ,} - \mu_{SQ,}$ such that $E(Z_j) = \zeta$

Let $\bar{Z}_j = \frac{1}{j}\sum_1^j Z_j$ be the sample mean,

Let $\hat{\sigma}_j^2 = \frac{1}{j-1}\sum_1^j \left(Z_j - \bar{Z}_j\right)^2$ be the sample variance

95% confidence interval for $\mu_{PQ} - \mu_{SQ}$ is

$$\left(\bar{Z}_j - t_{j-1,\frac{\alpha}{2}} * \frac{\hat{\sigma}_j}{\sqrt{j}}, \bar{Z}_j + t_{j-1,\frac{\alpha}{2}} * \frac{\hat{\sigma}_j}{\sqrt{j}}\right)$$

There is 3 possible outcomes:

1) If ‘0’ lies within the confidence interval, it means that both Queue types are equally efficient.
2) If ‘0’ lies to the left of the confidence interval, it means that PQ is more efficient than SQ
3) If ‘0’ lies to the right of the confidence interval, it means that SQ is more efficient than PQ

Data collected from simulation

Common Random Number: To ensure that we can compare the two Queuing types with better confidence, we subject both queueing types to the same random conditions. This is done through setting the attribute 'NumberOfReplication' in the Global Variable 'Simulation' to 100. This means that the 100 unique seeds use for the 100 runs for PQ will be the same 100 unique seeds use for the 100 runs for SQ. The 'ReportDirecttory' is where the data file will be stored.

Input Editor - Simulation

Key Inputs | Options | Multiple Runs

Keyword	Default	Value
Description	*None*	
RunDuration	8760.0 h	1 h
InitializationDuration	0.0 h	
ExitAtStop	FALSE	FALSE
GlobalSubstreamSeed	*this.ReplicationNu*	
PrintReport	FALSE	TRUE
ReportDirectory	*Configuration File*	'D:\Downloads (D;)\JAAM\output'
RunOutputList	*None*	

Input Editor - Simulation

Key Inputs | Options | Multiple Runs

Keyword	Default	Value
ScenarioIndexDefinitionList	*None*	
StartingScenarioNumber	1	
EndingScenarioNumber	1	
NumberOfReplications	1	100
NumberOfThreads	1	4
PrintReplications	TRUE	
PrintConfidenceIntervals	TRUE	FALSE
PrintRunLabels	TRUE	

The expression logger creates a txt file that logs all the average queue time data after an hour.

Input Editor - ExpressionLog1

Key Inputs | Options | Graphics

Keyword	Default	Value
Description	*None*	
DataSource	*None*	{ [Neuter_Cat_queue_1].AverageQueueTime } { [Neuter_Cat_queue_2].AverageQueueTime } { [Neuter_Cat_queue_3].AverageQueueTime }
SeparateFiles	FALSE	
IncludeInitialization	TRUE	
StartTime	0.0 h	
EndTime	Infinity h	
Interval	*None*	1 h
StateTraceList	*None*	
ValueTraceList	*None*	
WatchList	*None*	
VerifyWatchList	FALSE	
WatchListCondition	*FALSE*	

The data is then exported into an excel file for data wrangling and analysis.

Priority Queue

=SUM(output__3[@[x_PQ,1,j]:[x_PQ,3,j]])

x_PQ,1,j	x_PQ,2,j	x_PQ,3,j	X_PQ,j
0.10681565176152	0.09321315785819	0.06084022097222	j]:[x_PQ,3,j]])
0.09554084190123	0.03538586465873	0.00000000000000	0.13092671
0.09871792176136	0.04901680273016	0.09871328694444	0.24644801
0.09785736558081	0.03802664127604	0.12109647291667	0.25698048
0.10290480782946	0.03071250957562	0.13888888888889	0.27250621
0.10119255477904	0.03487901596354	0.10989464750000	0.24596622
0.09934778558081	0.06486673839869	0.06187426736111	0.22608879
0.10164014724806	0.04789606909314	0.12450090694444	0.27403712
0.10008367875000	0.03685297124603	0.10854396388889	0.24548061
0.09759024783333	0.07661398254630	0.00000000000000	0.17420423
0.10396378677003	0.06520035319801	0.10702901388889	0.27619315
0.10169614465762	0.04513200224537	0.11500281055556	0.26183096
0.09698536304321	0.04453471269444	0.05063608111111	0.19215616
0.09947690530934	0.04442077240196	0.13888888888889	0.28278657
0.10436139134921	0.08430308732253	0.03809604574074	0.22676052
0.10033923487374	0.04233527669841	0.13888888888889	0.2815634
0.10772530248645	0.07917592796547	0.09426371013889	0.28116494
0.10499270542328	0.07814842418860	0.10210543916667	0.28524657
0.10793215096883	0.07025038712963	0.00000000000000	0.17818254
0.09236211617021	0.05028173410590	0.13888888888889	0.28153274
0.09705868380247	0.06382532789062	0.05789638694444	0.2187804
0.09794714497475	0.05763073462698	0.10822592500000	0.2638038
0.10639275248645	0.07874844831140	0.00000000000000	0.1851412
0.10427098953704	0.07434974281863	0.09608070055556	0.27470143

PQ | SQ | Comparing Queues | 1 worker | +

Shortest Queue

=SUM(output__4[@[x_SQ,1,j]:[x_SQ,3,j]])

x_SQ,1,j	x_SQ,2,j	x_SQ,3,j	X_SQ,j
0.0339304535	0.0260488068	0.0616916880	[x_SQ,3,j]])
0.0955408419	0.0353858647	0.0000000000	0.130926707
0.0291521337	0.0236591179	0.0560780933	0.108889345
0.0312174244	0.0241020349	0.0656120769	0.120931536
0.1029048078	0.0307125096	0.1388888889	0.272506206
0.0284929415	0.0203362714	0.0783394738	0.127168687
0.0993477856	0.0648667384	0.0618742674	0.226088791
0.0315673758	0.0220738479	0.0709738781	0.124615102
0.1000836788	0.0368529712	0.1085439639	0.245480614
0.0975902478	0.0766139825	0.0000000000	0.17420423
0.0341017412	0.0231367813	0.0445709275	0.10180945
0.1016961447	0.0451320022	0.1150028106	0.261830957
0.0304801943	0.0265264475	0.0838468460	0.140853488
0.0994769053	0.0444207724	0.1388888889	0.282786567
0.0314731861	0.0293598299	0.0441744207	0.105007437
0.1003392349	0.0423352767	0.1388888889	0.2815634
0.1077253025	0.0791759280	0.0942637101	0.281164941
0.0258580103	0.0236765596	0.0541712146	0.103705785
0.1079321510	0.0702503871	0.0000000000	0.178182538
0.0923621162	0.0502817341	0.1388888889	0.281532739
0.0304943040	0.0274326604	0.0810844310	0.139011395
0.0979471450	0.0576307346	0.1082259250	0.263803805
0.1063927525	0.0787484483	0.0000000000	0.185141201
0.0422097927	0.0272092310	0.0464820266	0.11590105

PQ | SQ | Comparing Queues | 1 worker | +

	A	B	C	D	E	F
1	**X_PQ,j**	**X_SQ,j**	**Z**			
2	0.260869	0.121671	0.139198		**Zbar**	0.03075
3	0.130927	0.130927	0		**sigma**	0.057032
4	0.246448	0.108889	0.137559		**t_0.025,99**	2.276003
5	0.25698	0.120932	0.136049		**Low CI**	0.017769
6	0.272506	0.272506	0		**High CI**	0.04373
7	0.245966	0.127169	0.118798			
8	0.226089	0.226089	0			
9	0.274037	0.124615	0.149422			
10	0.245481	0.245481	0			
11	0.174204	0.174204	0			
12	0.276193	0.101809	0.174384			
13	0.261831	0.261831	0			
14	0.192156	0.140853	0.051303			
15	0.282787	0.282787	0			
16	0.226761	0.105007	0.121753			
17	0.281563	0.281563	0			
18	0.281165	0.281165	0			
19	0.285247	0.103706	0.181541			
20	0.178183	0.178183	0			
21	0.281533	0.281533	0			
22	0.21878	0.139011	0.079769			

	A	B	C	D	E	F
1	**X_PQ,j**	**X_SQ,j**	**Z**			
2	0.260869030591926	0.121670948316017	=A2-B2		**Zbar**	=AVERAGE(C2:C101)
3	0.130926706559965	0.130926706559965	=A3-B3		**sigma**	=STDEV.S(C2:C101)
4	0.246448011435967	0.108889344802366	=A4-B4		**t_0.025,99**	=T.INV.2T(0.05,99)
5	0.256980479773516	0.120931536189106	=A5-B5		**Low CI**	=F2-F4*F3/SQRT(J7)
6	0.272506206293962	0.272506206293962	=A6-B6		**High CI**	=F2+F4*F3/SQRT(J7)
7	0.245966218242582	0.127168686714703	=A7-B7			
8	0.226088791340612	0.226088791340612	=A8-B8			
9	0.274037123285643	0.124615101701929	=A9-B9			
10	0.24548061388492	0.24548061388492	=A10-B10			
11	0.17420423037963	0.17420423037963	=A11-B11			
12	0.276193153856919	0.101809450007936	=A12-B12			
13	0.261830957458547	0.261830957458547	=A13-B13			
14	0.192156156848765	0.140853487865206	=A14-B14			
15	0.282786566600192	0.282786566600192	=A15-B15			
16	0.226760524412478	0.105007436729995	=A16-B16			
17	0.281563400461038	0.281563400461038	=A17-B17			
18	0.281164940590803	0.281164940590803	=A18-B18			
19	0.285246568778542	0.103705784549176	=A19-B19			
20	0.178182538098464	0.178182538098464	=A20-B20			

Output result:

95% confidence interval for $\mu_{PQ} - \mu_{SQ}$ is $(0.019433, 0.042066)$

Since 0 lies to the left of the confidence interval, PQ is more efficient than SQ.

Output Analysis #2

We would like to find out what number of workers for each station for Priority Queue type and Shortest Queue type is the most efficient one for different Delivery Types

Approach

We will plot a graph of 100 unique runs lasting for an hour each for average queueing time vs different number of workers.

Let $u_{i,k,d,w,j}$ be the average queueing time of stray cats in respective Queue type i $\in${**Priority Queue (PQ), Shortest Queue (SQ)**}, Station k $\in${1,2,3}, Delivery type d$\in$**{Priority Delivery (PD), Shortest Delivery (SD)}**, number of workers in respective station w $\in Z^+$, with run j $\in${1,2,3,….,100}

We then find the average queueing time for each station over 100 unique runs to be $U_{i,k,d,w}$ where: $U_{i,k,d,w} = \frac{1}{100}\sum_{j=1}^{100} u_{i,k,d,w,j}$

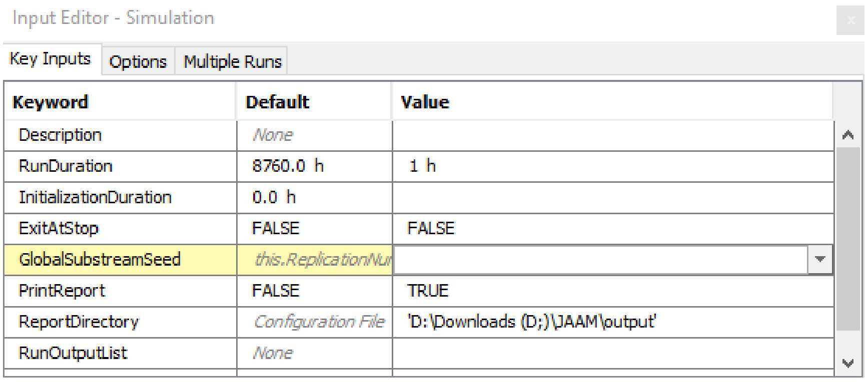
Allowing us to arrive to all possible combinations of outputs with varying w:

1. $U_{PQ,1,PD,w}$ vs $U_{PQ,2,PD,w}$ vs $U_{PQ,3,PD,w}$
2. $U_{SQ,1,PD,w}$ vs $U_{SQ,2,PD,w}$ vs $U_{SQ,3,PD,w}$
3. $U_{PQ,1,SD,w}$ vs $U_{PQ,2,SD,w}$ vs $U_{PQ,3,SD,w}$
4. $U_{SQ,1,SD,w}$ vs $U_{SQ,2,SD,w}$ vs $U_{SQ,3,SD,w}$

We can then plot a line graph of each combination with varying w to help us find which number w would provide us with the smallest queueing time for respective station.

Data collected from simulation

Common Random Number: To ensure that we can compare the two Queuing types with better confidence, we subject both queueing types to the same random conditions. This is done through setting the attribute 'NumberOfReplication' in the Global Variable 'Simulation' to 100. This means that the 100 unique seeds use for the 100 runs for PQ will be the same 100 unique seeds use for the 100 runs for SQ. The 'ReportDirecttory' is where the data file will be stored.



Input Editor - Simulation

Key Inputs | Options | Multiple Runs

Keyword	Default	Value
Description	None	
RunDuration	8760.0 h	1 h
InitializationDuration	0.0 h	
ExitAtStop	FALSE	FALSE
GlobalSubstreamSeed	this.ReplicationNu	
PrintReport	FALSE	TRUE
ReportDirectory	Configuration File	'D:\Downloads (D;)\JAAM\output'
RunOutputList	None	

Input Editor - Simulation

Key Inputs | Options | Multiple Runs

Keyword	Default	Value
ScenarioIndexDefinitionList	None	
StartingScenarioNumber	1	
EndingScenarioNumber	1	
NumberOfReplications	1	100
NumberOfThreads	1	4
PrintReplications	TRUE	
PrintConfidenceIntervals	TRUE	FALSE
PrintRunLabels	TRUE	

The expression logger creates a txt file that logs all the average queue time data after an hour.

Input Editor - ExpressionLog1

Key Inputs | Options | Graphics

Keyword	Default	Value
Description	None	
DataSource	None	{ [Neuter_Cat_queue_1].AverageQueueTime } { [Neuter_Cat_queue_2].AverageQueueTime } { [Neuter_Cat_queue_3].AverageQueueTime }
SeparateFiles	FALSE	
IncludeInitialization	TRUE	
StartTime	0.0 h	
EndTime	Infinity h	
Interval	None	1 h
StateTraceList	None	
ValueTraceList	None	
WatchList	None	
VerifyWatchList	FALSE	
WatchListCondition	FALSE	

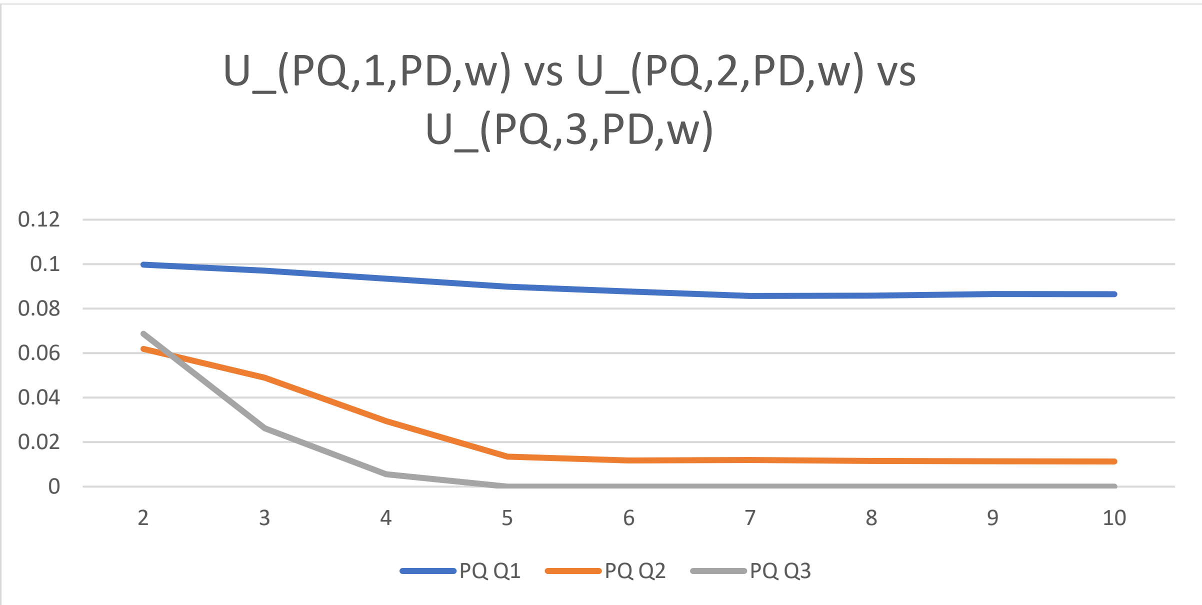
The data is then exported into an excel file for data wrangling and analysis.

	A	B	C	D	E	F	G
1	Truck	PD					
2	Station	PQ			SQ		
3	#workers	Q1	Q2	Q3	Q1	Q2	Q3
4	2	0.09978	0.0619	0.0687	0.07168	0.04806	0.06293
5	3	0.09708	0.04894	0.02623	0.06833	0.0393	0.04872
6	4	0.0935	0.02943	0.00556	0.06327	0.02214	0.04403
7	5	0.08984	0.01349	0	0.05981	0.01238	0.04728
8	6	0.08774	0.01175	0	0.05811	0.01132	0.04982
9	7	0.0857	0.01198	0	0.05826	0.01181	0.05228
10	8	0.08585	0.01153	0	0.05886	0.0113	0.05292
11	9	0.08658	0.01141	0	0.05895	0.01121	0.05345
12	10	0.08648	0.01129	0	0.05882	0.01107	0.05384

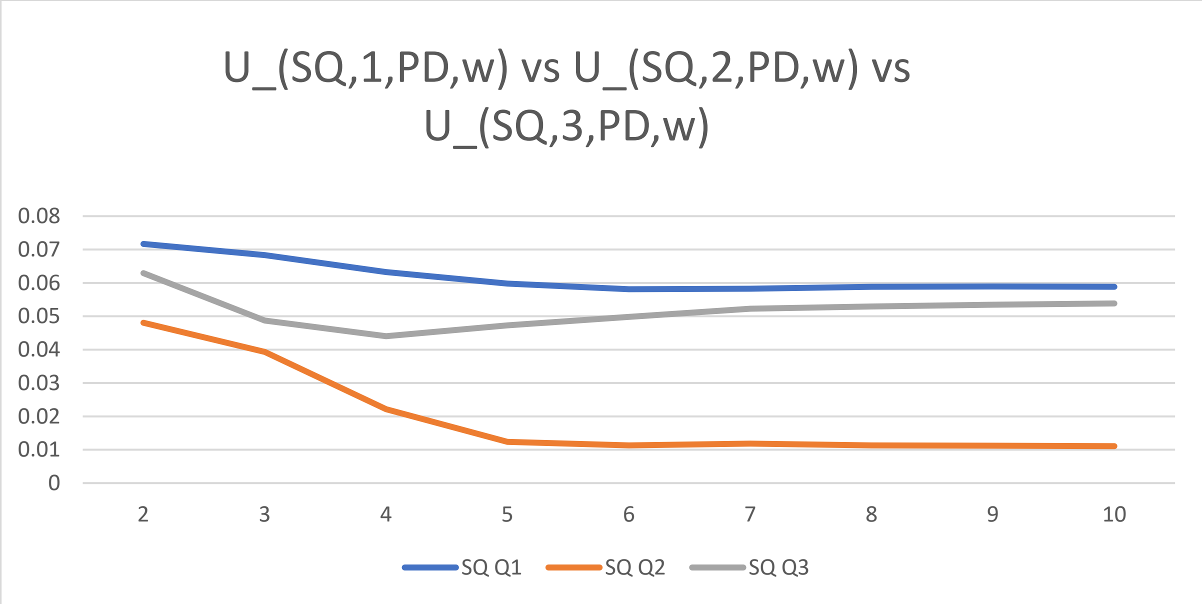
Truck	SD					
station	PQ			SQ		
#workers	Q1	Q2	Q3	Q1	Q2	Q3
2	0.09983	0.06206	0.06667	0.07037	0.04687	0.0639
3	0.09724	0.04896	0.02382	0.06822	0.03885	0.03764
4	0.09364	0.02924	0.00498	0.06326	0.02147	0.03604
5	0.08984	0.01346	0	0.05971	0.01224	0.04059
6	0.08774	0.01175	0	0.05661	0.01173	0.05517
7	0.08539	0.01137	0	0.05675	0.01111	0.05544
8	0.08604	0.01105	0	0.05552	0.01108	0.05929
9	0.08656	0.0109	0	0.05764	0.01069	0.05809
10	0.08658	0.01077	0	0.05698	0.0111	0.05772

Y-axis: average Queuing time , X-axis: # of workers

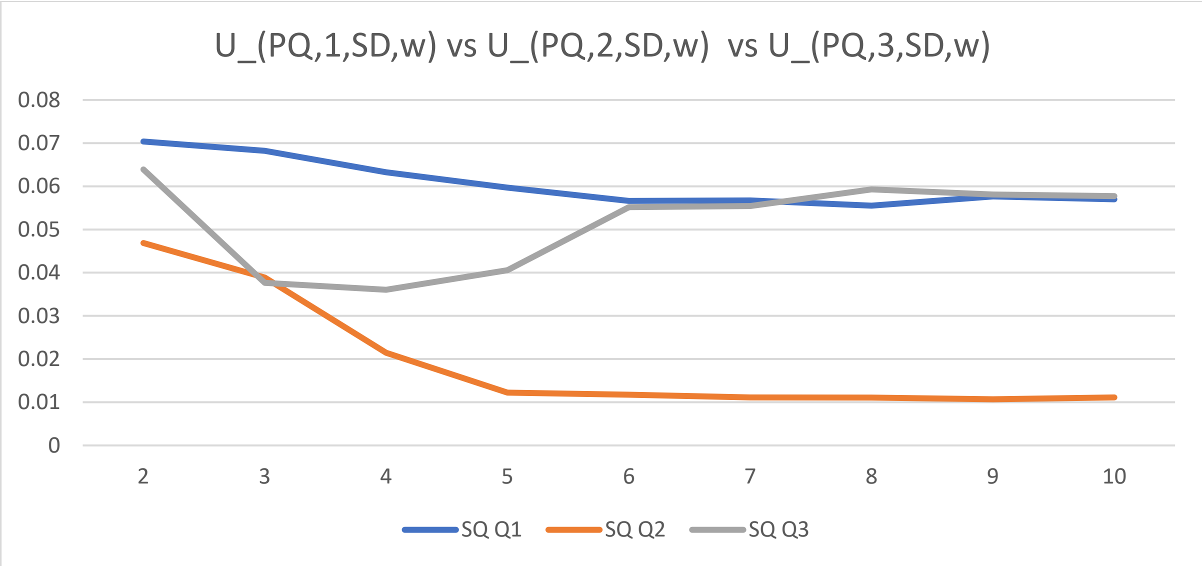
1. $U_{PQ,1,PD,w}$ vs $U_{PQ,2,PD,w}$ vs $U_{PQ,3,PD,w}$



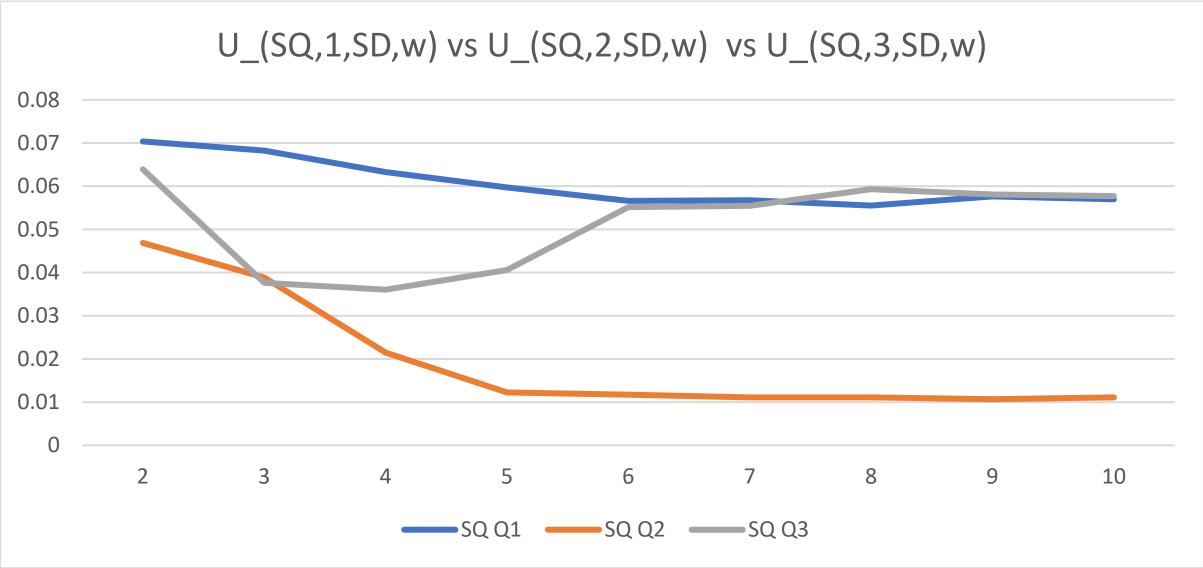
2. $U_{SQ,1,PD,w}$ vs $U_{SQ,2,PD,w}$ vs $U_{SQ,3,PD,w}$



3. $U_{PQ,1,SD,w}$ vs $U_{PQ,2,SD,w}$ vs $U_{PQ,3,SD,w}$



4. $U_{SQ,1,SD,w}$ vs $U_{SQ,2,SD,w}$ vs $U_{SQ,3,SD,w}$

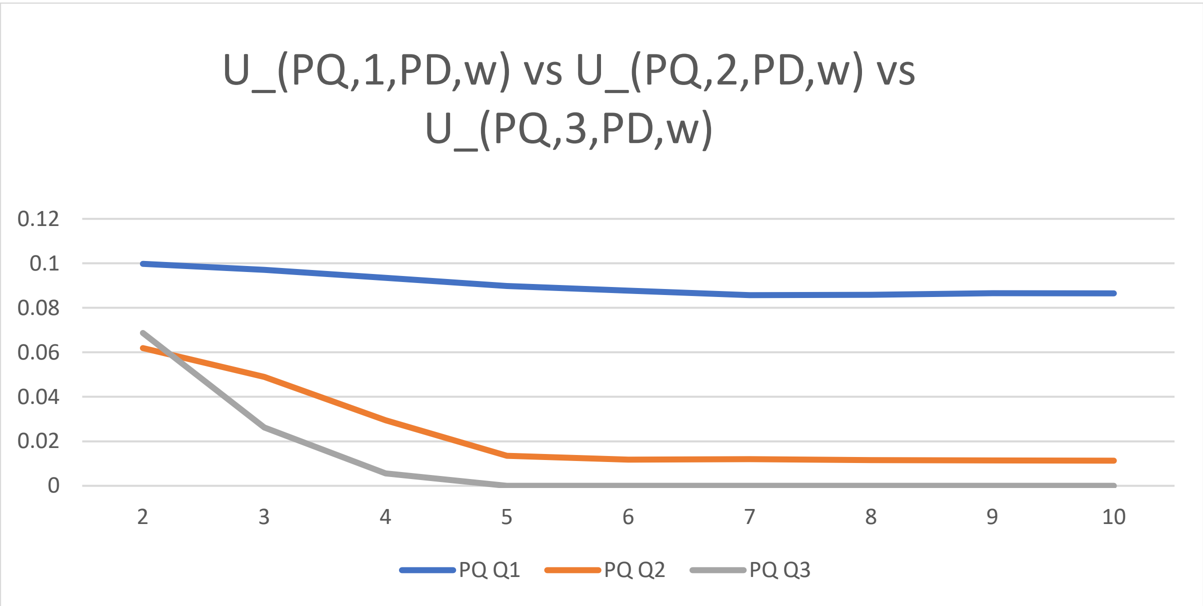


Output result:

Across all four combinations we noticed that combination 1 and 2 can provide the relatively lower average queueing time. Hence, we will now compare combination 1 and 2 with varying workers for each station.

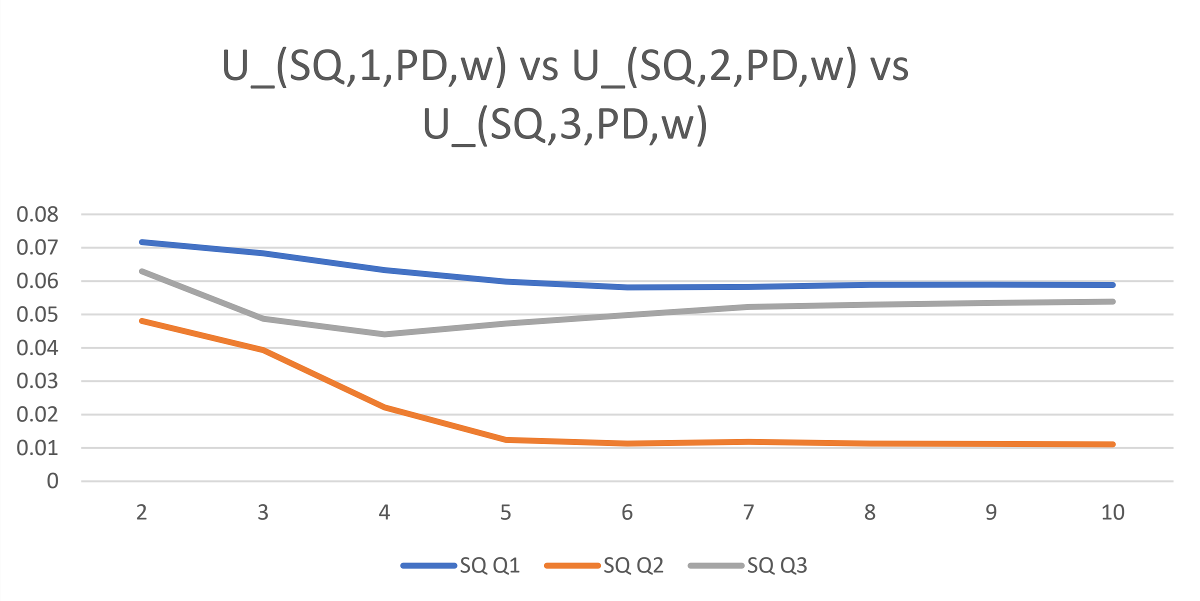
Let w_{stnk} be the number of workers in station k that has the lowest average waiting time by referring to the lowest value of the y axis of the plot.

Combination 1:



w_stn1, w_stn2, w_stn3	P1 Q1	Q2	Q3	Sum of all average queueing times across stations
7, 5, 4	0.08603	0.01371	0	0.099738677
5, 5, 5	0.08984	0.01349	0	0.103334303

Combination 2



w_stn1, w_stn2, w_stn3	SQ Q1	Q2	Q3	Sum of all average queueing times across stations
5, 5, 4	0.05957	0.01282	0.04741	0.119803795
5, 5, 5	0.05981	0.01238	0.04728	0.119474796
5, 5, 7	0.06207	0.01327	0.04109	0.116430403
2, 4, 2	0.06963	0.02961	0.08444	0.183687457
5, 2, 5	0.06256	0.02996	0.02704	0.119562799

We then sum across the stations and found out that the lowest average queueing type is 0.0997 with combination of $U_{PQ,1,PD,7}$ vs $U_{PQ,2,PD,5}$ vs $U_{PQ,3,PD,4}$

Output Analysis #3

Jaamsim allows users to directly infer output data with the graph function. Users are able change inputs before the simulation start, or anytime during the simulation to get different output results.

These are the following User Inputs in Jaamsim:

Initial_State_Inputs	Value(s)
Abandon Rate Min inflation	2
Abandon Rate Max inflation	10
Green: High Demand for Cats Red: Low Demand for Cats	Click_To_Change_DemandType_button

TNR_Stations_Inputs

	No. Of Workers	Max Queue Length	Service Rate	Safety stock
Stn1	2	5	5 s	10
Stn2	2	5	5 s	10
Stn3	2	5	5 s	10

Green: ShortestQueue Red: Priority Q1	Click_To_change_QueueTypeButton
Green: High Workers Speed Rate Red: Low Workers Speed Rate	Click_To_Change_WorkersSpeedType_button
Green: High Unpackaing Speed Rate Red: Low Unpacking Speed Rate	Click_To_Change_UnpackingSpeedType_button

Factory_Inputs	Value(s)
Reorder quantity	20
Green: High Factory ProdRate Red: Low Factory ProdRate	Click_To_Change_WorkType
Green: ShortestQueue Red: Priority Queue	Click_To_Change_DeliveryType
Black: Fast Delivery Red: Slow Delivery	Click_To_Change_TruckSpeedButton

Outputs that users can infer from:

Outputs	Value(s)
Average queuing time Stn1	0.05629485909722222[h]
Average queuing time Stn2	0.08381247055555556[h]
Average queuing time Stn3	0.09526216298611112[h]
# Angel Zombies	{7.0}
# Zombies	{2.0}
# of Mega-Zombies	{0.0}

Approach

We can experiment with **one control variable** (High demand of cats) and have other different user inputs to get different outputs:

Experiment #1:

- All Green and Black button for high speed, fast delivery, high demand of cats
- Pause Simulation at 24hrs,
- Priority Queue
- Priority Delivery

Experiment #2:

- All Red button for low speed, slow delivery
- Keep Demand button Green for high demand for cats
- Pause Simulation at 24hrs,
- Priority Queue
- Priority Delivery

Output result:

Experiment #1

Inputs:

Initial_State_Inputs	Value(s)
Abandon Rate Min inflation	2
Abandon Rate Max inflation	10
Green: High Demand for Cats Red: Low Demand for Cats	Click_To_Change_DemandType_button

TNR_Stations_Inputs

	No. Of Workers	Max Queue Length	Service Rate	Safety stock
Stn1	2	5	5 s	10
Stn2	2	5	5 s	10
Stn3	2	5	5 s	10

Green: ShortestQueue Red: Priority Q1	Click_To_change_QueueTypeButton
Green: High Workers Speed Rate Red: Low Workers Speed Rate	Click_To_Change_WorkersSpeedType_button
Green: High Unpackaing Speed Rate Red: Low Unpacking Speed Rate	Click_To_Change_UnpackingSpeedType_button

Factory_Inputs	Value(s)
Reorder quantity	20
Green: High Factory ProdRate Red: Low Factory ProdRate	Click_To_Change_WorkType
Green: ShortestQueue Red: Priority Queue	Click_To_Change_DeliveryType
Black: Fast Delivery Red: Slow Delivery	Click_To_Change_TruckSpeedButton

Outputs:

Outputs	Value(s)
Average queuing time Stn1	0.01597928673421846[h]
Average queuing time Stn2	0.0069831941877350106[h]
Average queuing time Stn3	0.006756855256813419[h]
# Angel Zombies	{1912.0}
# Zombies	{4.0}
# of Mega-Zombies	{0.0}

Experiment #2

Inputs:

Initial_State_Inputs	Value(s)
Abandon Rate Min inflation	2
Abandon Rate Max inflation	10
Green: High Demand for Cats Red: Low Demand for Cats	Click_To_Change_DemandType_button

TNR_Stations_Inputs

	No. Of Workers	Max Queue Length	Service Rate	Safety stock
Stn1	2	5	5 s	10
Stn2	2	5	5 s	10
Stn3	2	5	5 s	10

Green: ShortestQueue Red: Priority Q1	Click_To_change_QueueTypeButton
Green: High Workers Speed Rate Red: Low Workers Speed Rate	Click_To_Change_WorkersSpeedType_button
Green: High Unpackaing Speed Rate Red: Low Unpacking Speed Rate	Click_To_Change_UnpackingSpeedType_button

Factory_Inputs	Value(s)
Reorder quantity	20
Green: High Factory ProdRate Red: Low Factory ProdRate	Click_To_Change_WorkType
Green: ShortestQueue Red: Priority Queue	Click_To_Change_DeliveryType
Black: Fast Delivery Red: Slow Delivery	Click_To_Change_TruckSpeedButton

Outputs:

Outputs	Value(s)
Average queuing time Stn1	0.12137927857962974[h]
Average queuing time Stn2	0.11729497007762933[h]
Average queuing time Stn3	0.003663646739690723[h]
# Angel Zombies	{1347.0}
# Zombies	{560.0}
# of Mega-Zombies	{27.0}

Comparing Experiment 1 and Experiment 2 with a control variable of high demand of cats suggests that with a working rate, delivery speed, and factory production rate affects the average queueing time for the all stations, and decrease the number of cats getting neutered (higher number of zombies).

Other possible Output analysis:

- Using Priority Queue, compare the average queueing rate for having 2 slow workers for all 3 stations and 1 fast worker for all 3 stations.
- Using Priority Queue and Shortest Delivery repeat the experiment for Output analysis #1.
- Using Shortest Queue and Shortest Delivery repeat the experiment for Output analysis #1.