

Queues and Wait Times at Canadian Airports

P. Boily (CQADS), Y. Zhao (Carleton), W. Ye (Carleton)
M. Haghighi (CATSA), J. Lavigne (CATSA)



Carleton
UNIVERSITY



CQADS
CAQAD

Abstract

By providing efficient and effective pre-board screening (PBS), the Canadian Air Transport Security Authority (CATSA) ensures that security requirements are met while maintaining an appropriate balance between staffing and the wait time experienced by passengers.

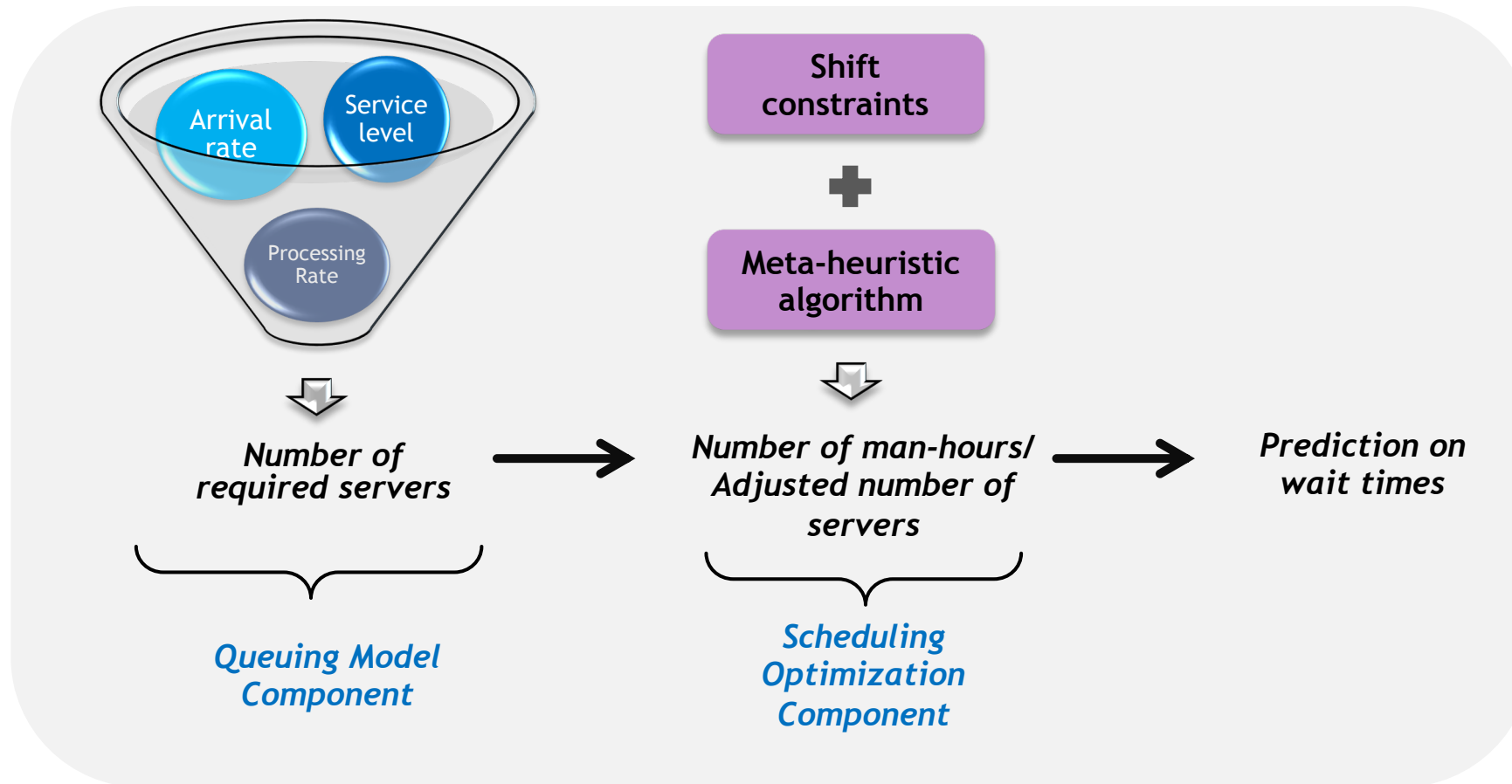
We use **queueing theory** to develop a model which can predict, among other things, the number of servers required to achieve particular service levels based on forecast arrival rates.

In this presentation, we describe the underlying model and discuss some of its possible refinements.

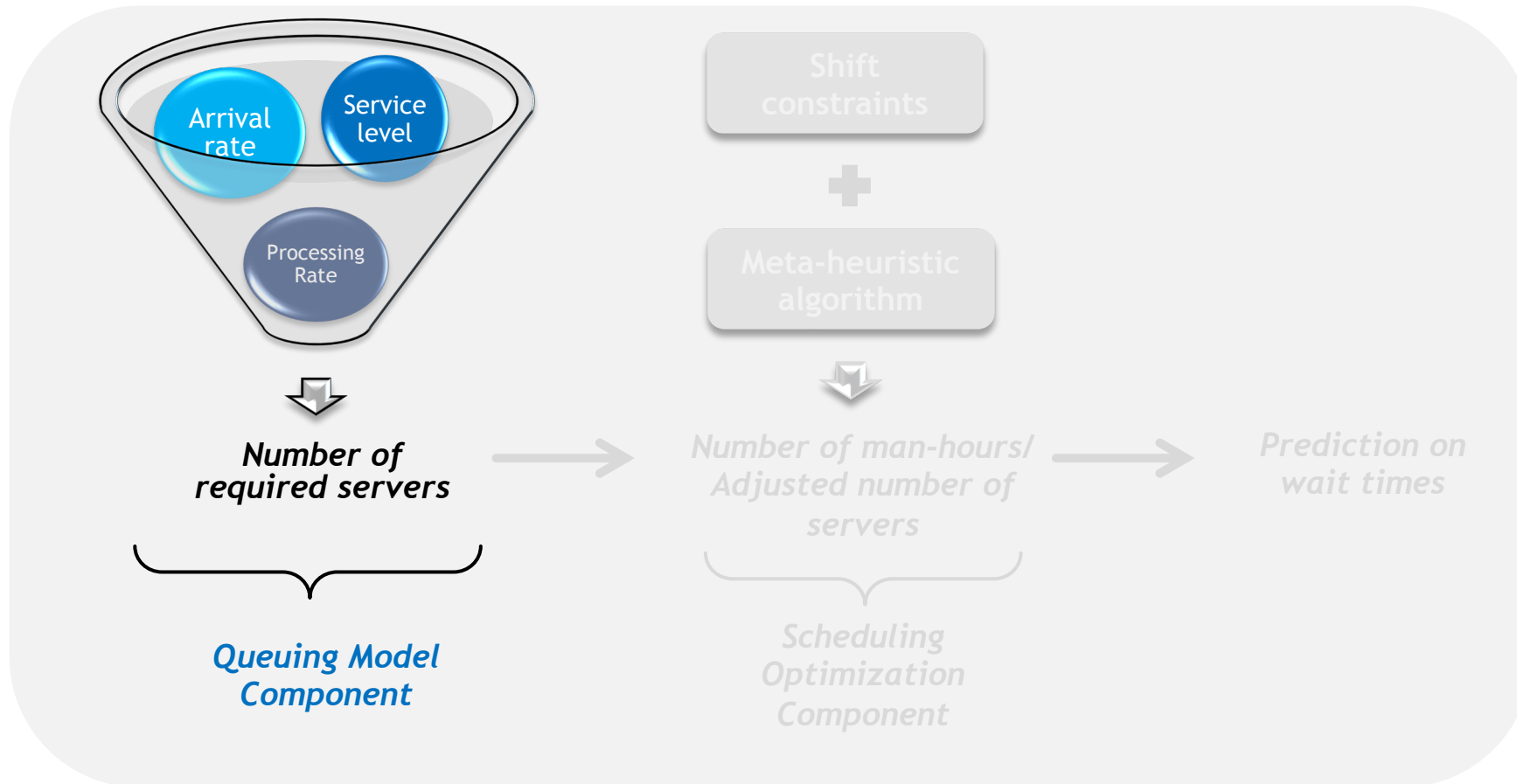
Outline

1. Preliminaries
2. Model Flow
3. $M/M/1$ Queueing Model
4. Regression Model
5. Departure Model
6. Predictions
7. Supplemental Comments
8. Consulting Post-Mortem

CATSA's Wait Time Impact Model



Presentation Focus



Preliminaries

Objectives

For each combination of checkpoint, time period, day of the week, season (cluster), use field data to provide estimates of:

- passenger arrival rates λ
- processing rates μ
- number of servers c

For each cluster, given λ , μ , and c , calculate the quality of service (QoS) level curve $(p_x(x), x)$ (i.e. the percentage p of passengers which will wait less than x minutes, for x fixed).

For each cluster, predict the average number of servers c^* required to achieve a prescribed QoS level (p_x, x) given an arrival profile λ^* .

Preliminaries

Definitions

The **Poisson process** is a stochastic process where the time between any two consecutive events is exponentially distributed with parameter λ .

$M/M/c$ queueing model

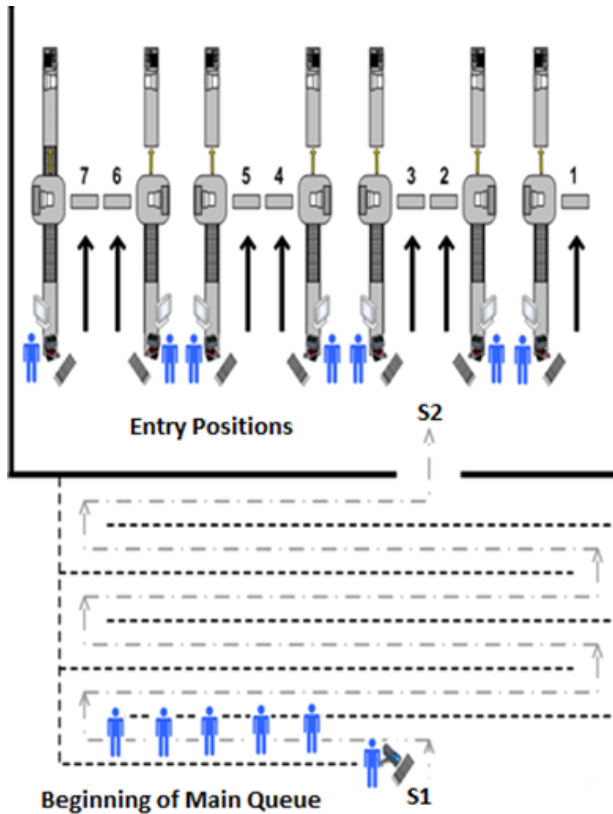
- arrivals form a single queue governed by a Poisson process
- arriving customers are processed by c servers
- service times are exponentially distributed

Various quantities

- **arrival rate:** rate at which passengers arrive for PBS (i.e. passengers per minute)
- **service rate:** processing rate at a screening line (i.e. maximal potential throughput)
- **number of servers:** number of screening lines
- **service level:** % of people waiting less than a given number of minutes at a checkpoint

Preliminaries

Description of the PBS Process



Passengers enter the main queue, where their boarding pass may be scanned at S_1 .

Once they reach the end of the main queue, their boarding pass is scanned at S_2 and they are sent to one of the active lines for processing.

In practice, it may often happen that only the S_2 reading is available.

Preliminaries

Available Data Sources

Raw Data: for each passenger reaching the end of the main queue, this dataset records

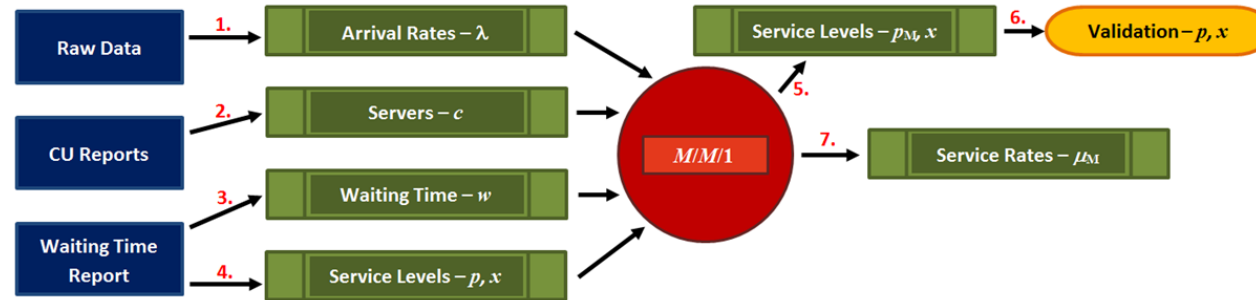
- the date
- the scan time upon entering the main queue (S_1)
- the scan time upon exiting the main queue (S_2)
- the wait time between S_1 and S_2

Passengers may not have been scanned upon entering the main queue.

Checkpoint Utilization Report: for each day of the year and each 15-minute block, this dataset records the maximum number of open lines (servers).

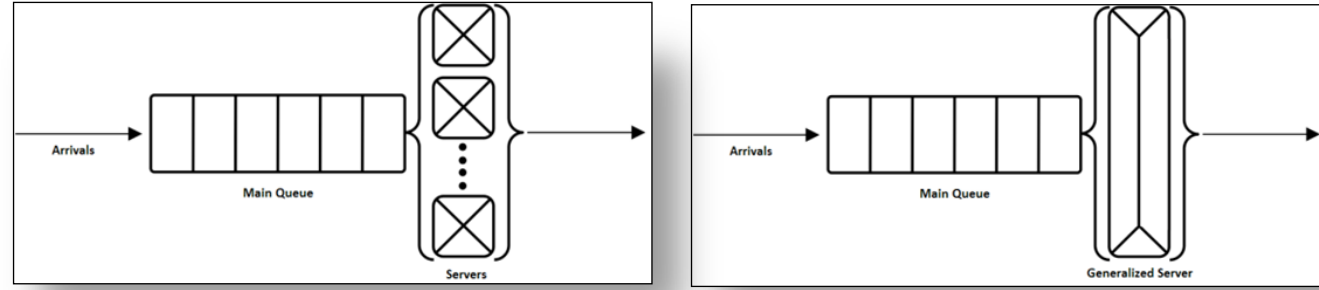
Waiting Time Report: consists of the subset of Raw Data for which S_1 and S_2 are both available. Observations for which the wait time exhibits outlying behaviour have been removed by CATSA.

Model Flow I — $M/M/1$



$M/M/1$ Queuing Model

Generalized Servers



Number of servers varies with time — problematic since service rate estimates with $M/M/c$ depend on the number of open servers:

- there are times when all servers are busy
- others when a number of open servers are idle
- the number of open servers changes according to some vacation policy (difficult to model)

Circumvent this issue (without invoking Vacation Models):

- $M/M/c$ queue viewed as $M/M/1$ queue where servers are hidden behind a generalized server
- Service rates can be estimated independently of the number of servers
- Major theorems still hold with $c = 1$, but related quantities are easier to compute

Theory of $M/M/1$ systems is not sufficient to recover (and predict) the number of servers for the checkpoint: **look for link between arrival rate, estimated service rate, and number of servers.**

$M/M/1$ Queuing Model

“Clusters”

Need to group the data into meaningful “clusters” exhibiting similar properties (i.e. **properties that can be characterized by the same Poisson process**).

- Allows for proper estimation of queuing model parameters (arrival rates, processing rates, etc.)

The selection of the appropriate “cluster” size relies on finding a balancing point between two extremes:

- If clusters cover too long a period of time, the single Poisson process assumption may fail.
- If clusters cover too short a period of time, they are unlikely to exhibit the statistical behaviour of the process.

Preliminary analysis of the model’s accuracy based on: **checkpoint, weekly patterns** (day of week vs wkday/wkend), **seasonal patterns** (season vs month) and **daily patterns** (2-hr period vs 4-hr period).

- The cluster combination that produced the most encouraging queueing results when compared against actual reports was: checkpoint, weekday/weekend, season, 4 hour-period

$M/M/1$ Queuing Model

Average Arrival Rate I

Some boarding passes are not scanned at S_1 so data cannot be used to derive the cluster arrival rates.

$S_1 - S_2$ line-up is a birth-death process (i.e. a reversible 1-D Markov chain)

- forward chain $S_1 - S_2$ and reverse are stochastically identical and the arrival epochs of the reversed chain are the departure epochs of the forward chain

BURKE'S THEOREM for $M/M/c$ queues holds

- *Consider an $M/M/c$ queue in the steady state with arrivals modeled by a homogeneous Poisson process with rate parameter λ . Then the departure process is also a homogeneous Poisson process with rate parameter λ .*

All S_1 arrivals will eventually leave at S_2 and so the fluctuations at S_2 follow the same statistical property governing arrivals to the queue: **arrival rates can be estimated by using data readings at S_2** within a given cluster.

M/M/1 Queuing Model

Average Arrival Rate II

Let the number of arrivals in the cluster by time t is denoted by $N(t)$. Then

- $N(t)$ is a **counting process** ... (obviously satisfied ✓)
- with **independent** and **stationary** increments, (satisfied with introduction of clusters ✓)
- the number of arrivals in any time interval of length t is **Poisson-distributed** with mean λt , i.e. for all $s, t \geq 0$,

$$P[N(t + s) - N(s) = n] = e^{-\lambda t} \frac{(\lambda t)^n}{n!}, \quad n = 0, 1, \dots$$

(holds if the **inter-arrival times** are i.i.d. exponential random variables with the same rate λ ... analysis of S_2 in the raw data suggests that this is a decent assumption to make ✓)

Conclusion: arrivals follow a homogeneous (roughly) Poisson process, in each cluster.

M/M/1 Queuing Model

Average Number of Servers

Number of active servers at each checkpoint can be adjusted at any moment during each time period, in order to **accommodate fluctuations in arrivals**.

The CU reports record the **maximum** number of simultaneously active servers during 15-minute block.

Discrepancy between the actual numbers and the reported number is fairly small, due to the short time duration of the blocks.

Data is not available for smaller time scales.

Average is computed over all 15-minute blocks within each cluster.

M/M/1 Queuing Model

Average Wait Time

Not all wait time data is available — if S_1 data is **representative of the overall raw data**, the “real” wait time distribution can be estimated from the subset:

- Since the full wait time data is inaccessible, representativeness is hard to prove

Possible reasons why a raw data observation may not be included in the wait time report include:

1. scanned at S_1 , but the calculated wait time $w = S_2 - S_1$ is an outlier compared to neighbours
2. not scanned at S_1 because the scanner was overwhelmed by incoming traffic
3. main queue was empty and passenger was processed immediately, leading to $w = 0$

Reason 3 may introduce **bias** if many such observations were removed, which could affect the predicted QoS levels in the small wait time regime.

Possible to enter the main queue in one cluster and leave it in another. **Convention:** cluster in which w is recorded is the cluster in which S_2 falls.

M/M/1 Queuing Model

Service Rate

For generalized *M/M/1* queues,

$$p(x) = P(W_q \leq x) = 1 - \rho e^{-(\mu-\lambda)x} \quad \text{and} \quad \overline{W}_q = \frac{\rho}{\mu-\lambda}$$

where $\rho = \lambda/\mu$ is the **traffic intensity** of the system.

If arrival rate λ is known and average wait time $\overline{W}_q$ can be computed by other means, then it is possible to recover service rate μ :

$$\hat{\mu}_M = \frac{\overline{W}_q \lambda + \sqrt{(\overline{W}_q \lambda)^2 + 4 \overline{W}_q \lambda}}{2 \overline{W}_q}$$

QoS levels: $\hat{p}_M(x) = 1 - \hat{\rho}_M e^{-(\hat{\mu}_M - \lambda)x} \in (0,1)$ if $\hat{\rho}_M = \frac{\lambda}{\hat{\mu}_M} < 1$. Can still provide QoS levels when $\hat{\rho}_M > 1$ (assumptions fail)

$M/M/1$ Queuing Model

Validation

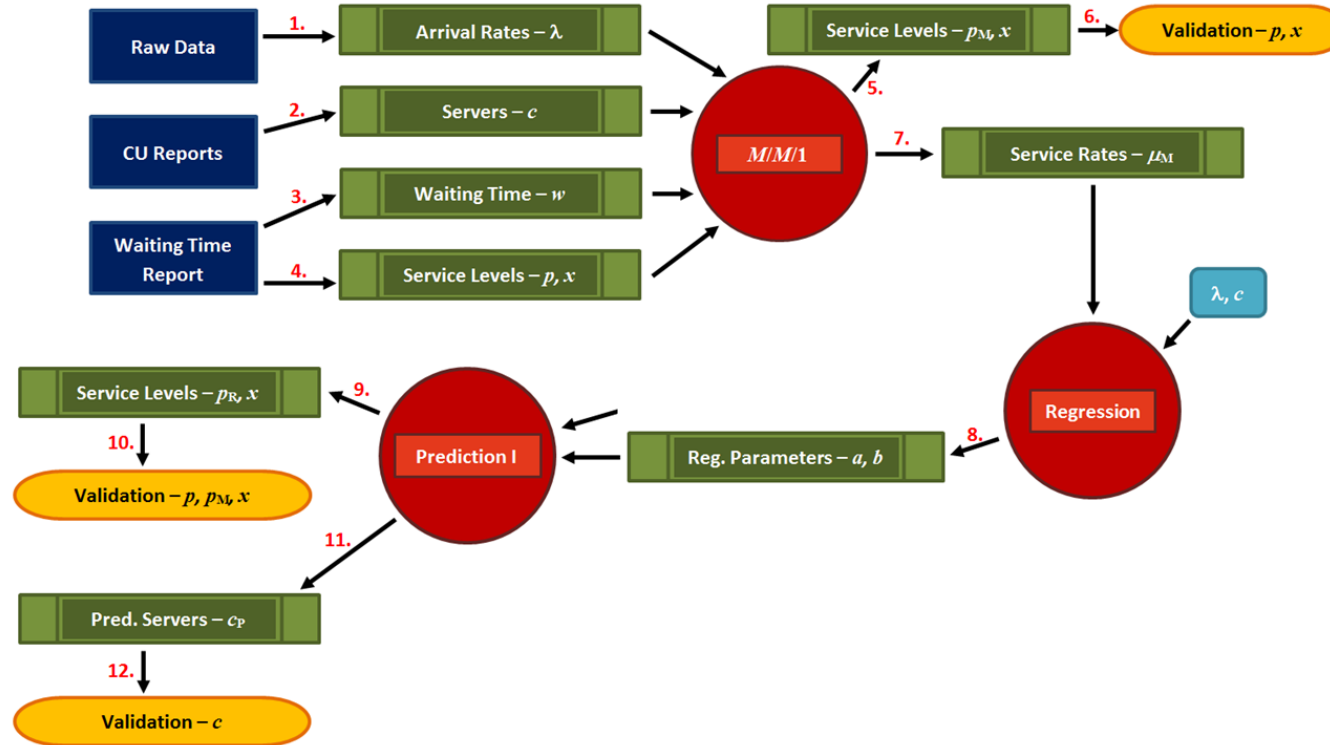
Relations do not hold if the generalized $M/M/1$ hypothesis fails

At this stage, the simplest way to validate is to compare the wait times generated by the model to those of the empirical data: are the estimated QoS levels $(\hat{p}_{M,x}, x)$ “close to” the empirical QoS levels (p_x, x) ?

We found that the generalized $M/M/1$ assumption, while not exact, remains reasonable to make at the checkpoint level

Still can't extract c without additional assumptions

Model Flow II — Regression



Regression Model

Linking λ , μ , and c

Regression assumption: on a quarterly level, the cluster processing rates μ is a function of

- the number of active servers c (hidden behind the generalized server) and
- the arrival rates λ

Simplest form: $\mu = \mu(\lambda, c) = ac + b\lambda$ (other forms possible as well)

Do quarterly clusters observations $\left(\frac{\lambda}{c}, \frac{\hat{\mu}_M}{c}\right)$ lie on a line?

Observations weighted by number of arrivals

Regression Model

Service Rate and Performance Levels

Service rate estimates: $\hat{\mu}_R = \hat{a}c + \hat{b}\lambda$, with optimal regression parameters $\hat{a}, \hat{b}$ (quarterly and seasonal estimates).

QoS level curves $(\hat{p}_R, x)$ estimates:

$$\hat{p}_R = 1 - \frac{\lambda}{\hat{a}c + \hat{b}\lambda} e^{-(\hat{a}c + \hat{b}\lambda - \lambda)x}$$

as long as $\hat{\rho}_R = \frac{\lambda}{\hat{\mu}_R} < 1$ (some complications may occur)

Additional assumptions:

- quarterly regressions produce good fits
- there is a quarterly characteristic to the service rate

Regression Model

Predicted Number of Servers

1. Start with $p = 1 - \frac{\lambda}{ac+b\lambda} e^{-(ac+b\lambda-\lambda)x}$, where $p, \frac{\lambda}{ac+b\lambda} \in (0,1)$
2. Re-arrange terms $\Rightarrow (ac + b\lambda)e^{(ac+b\lambda)x} = \frac{\lambda}{1-p} e^{\lambda x}$
3. Multiply by x on both sides $\Rightarrow (ac + b\lambda)xe^{(ac+b\lambda)x} = \frac{\lambda x}{1-p} e^{\lambda x}$
4. Set $y = (ac + b\lambda)x$ and $z = \frac{\lambda x}{1-p} e^{\lambda x} \Rightarrow ye^y = z$
5. Solve for $y \Rightarrow (ac + b\lambda)x = y = W_0(z) = W_0\left(\frac{\lambda x}{1-p} e^{\lambda x}\right)$
6. Solve for $c \Rightarrow c_R = \frac{1}{ax} \left[W_0\left(\frac{\lambda x}{1-p} e^{\lambda x}\right) - b\lambda x \right]$

The “physics” of the problem also require that $c \in (0, \text{maximum \# of servers})$

Regression Model Validation

Relations do not hold if the combined **M/M/1 + Regression** hypotheses fails

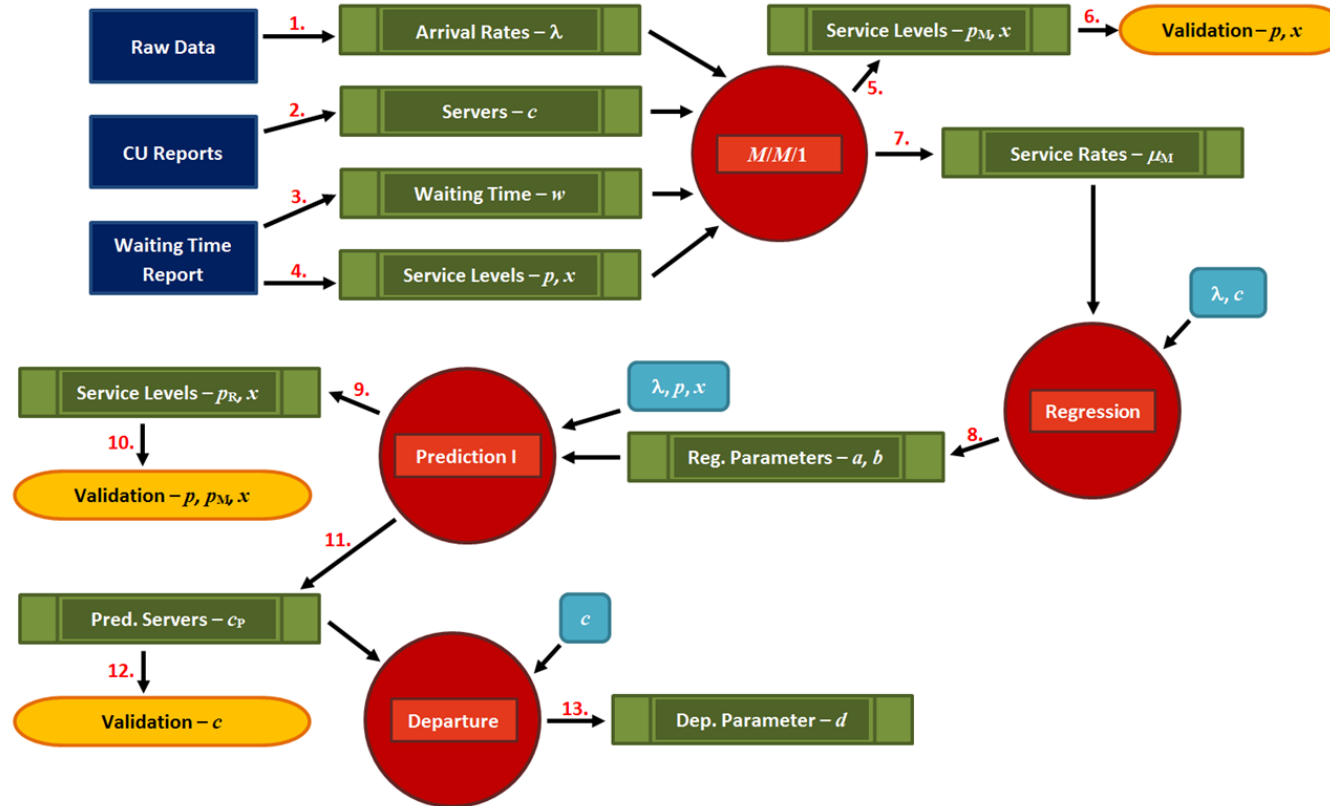
As before, we can compare the estimated QoS levels $(\hat{p}_{R,x}, x)$ to the empirical QoS levels (p_x, x)

But now we can also compare the predicted # of servers c_R to the empirical # of servers c

We found that the combined assumptions, while proving slightly less valid than the *M/M/1* hypothesis on its own, still provided reasonably close QoS estimates at the quarter and checkpoint levels (regression function $\mu = \mu(\lambda, c)$ adds some uncertainty)

Big advantage: can extract/predict the number of active servers c

Model Flow III — Departure



Departure Model

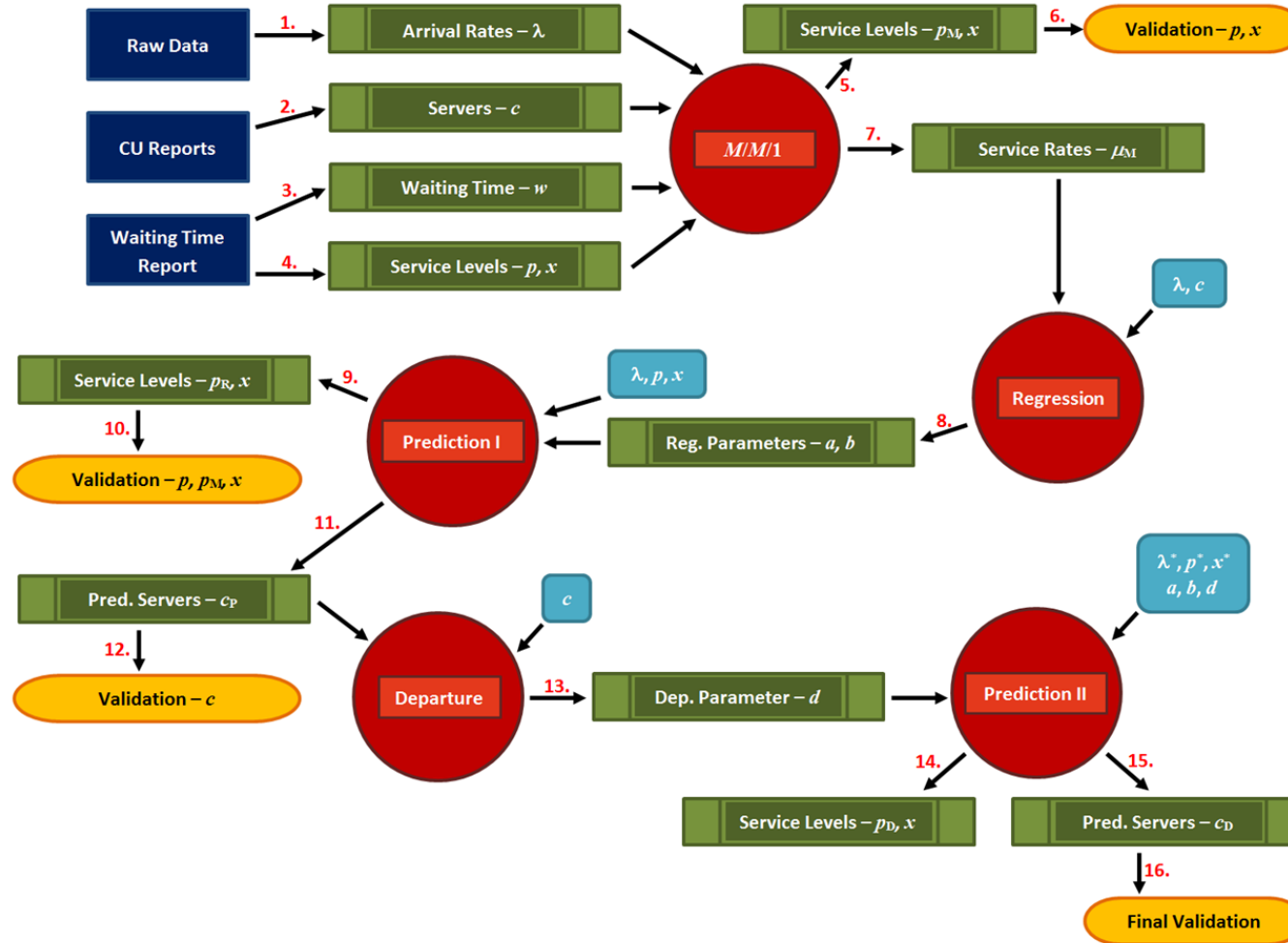
Predicted Number of Servers vs. Actual Number of Servers

For any given checkpoint, quarter, and cluster, we can compare the actual number of open servers c (given by the *CU Report*), and the predicted value c_R , given the actual arrival rate λ and the actual QoS level (p, x) .

Plotting c_R against c for all clusters strongly suggests that these were linked at the checkpoint level according to $c = \hat{d}c_R$, where $\hat{d}$ is the regression estimate of the **checkpoint departure parameter** d .

Computed values of d near 1 for nearly all checkpoints further validate the combined model.

Model Flow IV — Final Predictions



Combined Model

Final Predictions and Validation

Given an arrival rate λ , a desired QoS level (p, x) , and checkpoint parameters (a, b, d) , the predicted required number of servers is

$$c_D = d \cdot c_R = \frac{d}{ax} \left[W_0 \left(\frac{\lambda x}{1-p} e^{\lambda x} \right) - b\lambda x \right]$$

It makes little sense to compare the predicted value c_D with the actual number of servers c found in the historical data;

- the prediction depends not only on the forecasted arrival rate (which is likely to be different from the historical rate), but also on the attained QoS level (for which an independent forecast is unavailable)

Supplemental Comments

Accuracy and Potential Issues

The **$M/M/1$** model on its own provides the best QoS levels estimates, while the best estimates for the average number of active servers are provided by the **Departure** model.

- Some loss of information is inevitable due to the necessity of simplification assumptions

Possible issues which could affect the model's accuracy:

- Underlying arrival processes are roughly Poisson, wait time distributions are roughly conditionally exponential for each cluster; depending on the distance between the theoretical process and the empirical data, the $M/M/1$ assumption may be inappropriate
- Wait time distributions may be strongly biased due to missing S_1 scans; no easy way to verify how representative it is
- Server vacation policy is unknown, and may not be uniformly adhered to
- Actual c crudely approximated by maximum number of active lines within a 15 minute block
- Service rates seem to depend on other factors, not just c and the λ

Supplemental Comments

Refinements and Recommendations

Different functional forms $\mu = \mu(\lambda, c)$

Model built using 2 years of data instead of a single year

Number of clusters

Queueing Approach vs. Simulation (of missing scans)

Illustration

Arrival Rates

(arbitrary example — not an actual CATSA checkpoint)

Cluster			# of Hours	Count	Avg Arrival Rate
	Week day	0:00 4:00	260	844	0.055
		4:00 8:00	260	129,069	8.274
		8:00 12:00	260	97,949	6.279
		12:00 16:00	260	84,548	5.420
		16:00 20:00	260	78,964	5.062
		20:00 0:00	260	33,061	2.119
	Week-end	0:00 4:00	104	1,076	0.172
		4:00 8:00	104	39,674	6.358
		8:00 12:00	104	31,200	5.000
		12:00 16:00	104	26,136	4.188
		16:00 20:00	104	28,129	4.508
		20:00 0:00	104	10,013	1.605

Illustration

Average Number of Servers

(arbitrary example — not an actual CATSA checkpoint)

Cluster				Avg # of Servers	Distribution of # of Active Servers								
					0	1	2	3	4	5	6	7	8
	Week day	0:00	4:00	0.14	86.7%	13.0%	0.3%	-	-	-	-	-	-
		4:00	8:00	5.38	-	7.6%	11.3%	4.4%	5.6%	11.3%	21.3%	20.4%	18.1%
		8:00	12:00	4.63	-	-	0.9%	10.3%	35.2%	33.6%	19.0%	1.0%	0.1%
		12:00	16:00	4.19	-	-	3.1%	21.9%	37.6%	27.8%	9.2%	0.4%	-
		16:00	20:00	3.78	-	-	17.7%	30.7%	22.3%	17.6%	9.2%	2.5%	-
		20:00	0:00	0.58	49.4%	42.9%	7.5%	0.2%	-	-	-	-	-
	Week-end	0:00	4:00	0.21	82.5%	13.9%	3.6%	-	-	-	-	-	-
		4:00	8:00	4.56	-	1.9%	9.4%	10.1%	20.7%	32.2%	18.8%	7.0%	-
		8:00	12:00	3.92	-	-	1.0%	31.3%	46.6%	18.0%	2.4%	0.7%	-
		12:00	16:00	3.41	-	-	6.3%	51.9%	36.8%	4.6%	0.5%	0.0%	-
		16:00	20:00	3.60	-	0.7%	17.5%	38.2%	18.8%	15.6%	8.2%	0.5%	0.5%
		20:00	0:00	1.47	0.2%	56.3%	39.7%	3.8%	-	-	-	-	-

Illustration

Average Wait Time and Performance Levels

(arbitrary example — not an actual CATSA checkpoint)

Cluster			Count	Avg Wait	Performance					
					5m	10m	15m	20m	25m	30m
	Week day	0:00 4:00	-	-	-	-	-	-	-	-
		4:00 8:00	50,132	6.564	57.1%	79.0%	88.7%	93.5%	96.6%	98.8%
		8:00 12:00	43,033	4.466	68.9%	89.2%	96.5%	99.5%	99.8%	99.9%
		12:00 16:00	32,380	5.374	64.1%	81.8%	92.6%	97.6%	99.3%	99.9%
		16:00 20:00	29,279	5.373	68.0%	81.8%	90.9%	95.8%	97.8%	99.1%
		20:00 0:00	4,511	2.975	86.3%	96.7%	99.9%	100%	100%	100%
	Week-end	0:00 4:00	204	3.992	70.6%	99.5%	100%	100%	100%	100%
		4:00 8:00	14,450	4.520	68.4%	86.4%	96.8%	99.3%	100%	100%
		8:00 12:00	12,638	3.317	82.3%	95.0%	96.8%	98.2%	99.7%	100%
		12:00 16:00	11,938	3.043	83.0%	95.6%	98.5%	99.9%	100%	100%
		16:00 20:00	8,625	5.247	60.5%	80.6%	94.3%	98.9%	100%	100%
		20:00 0:00	1,529	2.382	88.9%	100%	100%	100%	100%	100%

Illustration

Estimates — $M/M/1$

(arbitrary example — not an actual CATSA checkpoint)

Cluster			Est Serv Rate	Est ρ	Estimated Performance (M/M/1)					
					5m	10m	15m	20m	25m	30m
	Week day	0:00 4:00	-	-	-	-	-	-	-	-
		4:00 8:00	8.423	0.982	53.5%	78.0%	89.6%	95.1%	97.7%	98.9%
		8:00 12:00	6.495	0.967	67.2%	88.9%	96.2%	98.7%	99.6%	99.9%
		12:00 16:00	5.600	0.968	60.7%	84.0%	93.5%	97.4%	98.9%	99.6%
		16:00 20:00	5.242	0.966	60.7%	84.0%	93.5%	97.3%	98.9%	99.6%
		20:00 0:00	2.414	0.878	79.9%	95.4%	98.9%	99.8%	99.9%	100.0%
	Week-end	0:00 4:00	0.311	0.554	72.3%	86.2%	93.1%	96.5%	98.3%	99.1%
		4:00 8:00	6.572	0.967	66.8%	88.6%	96.1%	98.7%	99.5%	99.8%
		8:00 12:00	5.285	0.946	77.3%	94.5%	98.7%	99.7%	99.9%	100.0%
		12:00 16:00	4.495	0.932	79.8%	95.6%	99.1%	99.8%	100.0%	100.0%
		16:00 20:00	4.691	0.961	61.5%	84.6%	93.8%	97.5%	99.0%	99.6%
		20:00 0:00	1.950	0.823	85.4%	97.4%	99.5%	99.9%	100.0%	100.0%

Illustration

Regression

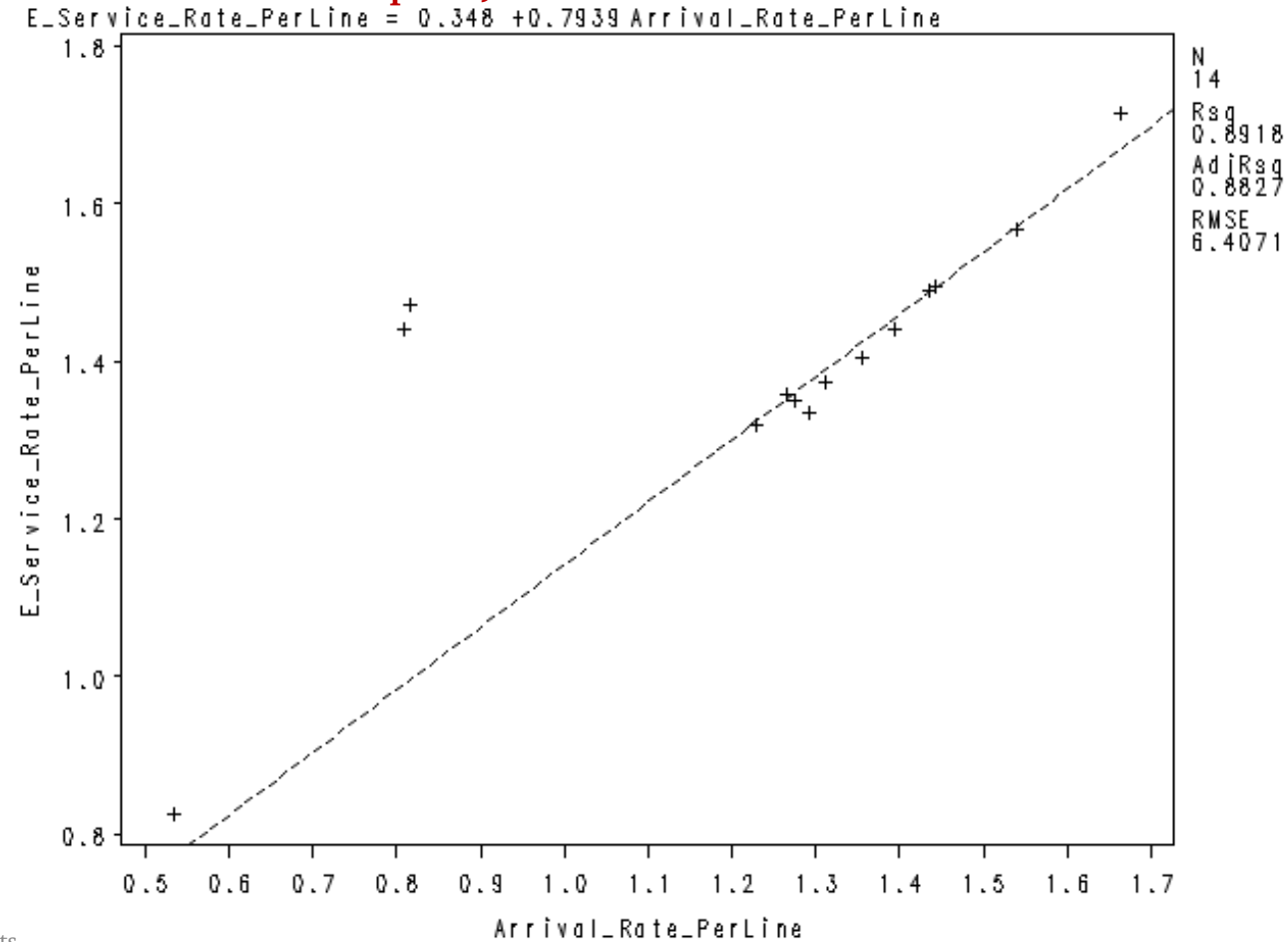
(arbitrary example — not an actual CATSA checkpoint)

Cluster			Avg # of Servers	Arrival Rate	Est Serv Rate	Arr Rate / Server	Serv Rate / Server
	Week day	0:00 4:00	0.14	0.055	0.000	0.405	0.000
		4:00 8:00	5.38	8.274	8.423	1.539	1.567
		8:00 12:00	4.63	6.279	6.495	1.356	1.403
		12:00 16:00	4.19	5.420	5.600	1.292	1.335
		16:00 20:00	3.78	5.062	5.242	1.341	1.388
		20:00 0:00	1.58	2.119	2.414	1.337	1.524
	Week-end	0:00 4:00	0.21	0.172	0.311	0.815	1.471
		4:00 8:00	4.56	6.358	6.572	1.394	1.441
		8:00 12:00	3.92	5.000	5.285	1.276	1.349
		12:00 16:00	3.41	4.188	4.495	1.228	1.318
		16:00 20:00	3.60	4.508	4.691	1.253	1.304
		20:00 0:00	1.47	1.605	1.950	1.091	1.326

Illustration

Regression

(arbitrary example — not an actual CATSA checkpoint)



Illustration

Estimates — $M/M/1$ + Regression

(arbitrary example — not an actual CATSA checkpoint)

Cluster			Class	Reg Serv Rate	Reg ρ	Estimated Performance ($M/M/1$ + Reg)					
						5m	10m	15m	20m	25m	30m
	Week day	0:00 4:00	-	-	-	-	-	-	-	-	-
		4:00 8:00	1.5	8.264	1.001	-	-	-	-	-	-
		8:00 12:00	1.4	6.591	0.953	80.0%	95.8%	99.1%	99.8%	100%	100%
		12:00 16:00	1.3	5.804	0.934	86.3%	98.0%	99.7%	100%	100%	100%
		16:00 20:00	1.3	5.338	0.948	76.2%	94.0%	98.5%	99.6%	99.9%	100%
		20:00 0:00	1.3	2.237	0.947	47.5%	70.9%	83.9%	91.1%	95.1%	97.3%
	Week-end	0:00 4:00	-	-	-	-	-	-	-	-	-
		4:00 8:00	1.4	6.600	0.963	71.3%	91.4%	97.4%	99.2%	99.8%	99.9%
		8:00 12:00	1.3	5.383	0.929	86.3%	98.0%	99.7%	100%	100%	100%
		12:00 16:00	1.2	4.584	0.914	87.4%	98.3%	99.8%	100%	100%	100%
		16:00 20:00	1.3	4.892	0.922	86.5%	98.0%	99.7%	100%	100%	100%
		20:00 0:00	1.1	1.852	0.867	74.8%	92.7%	97.9%	99.4%	99.8%	99.9%

Illustration

Predicted Number of Servers

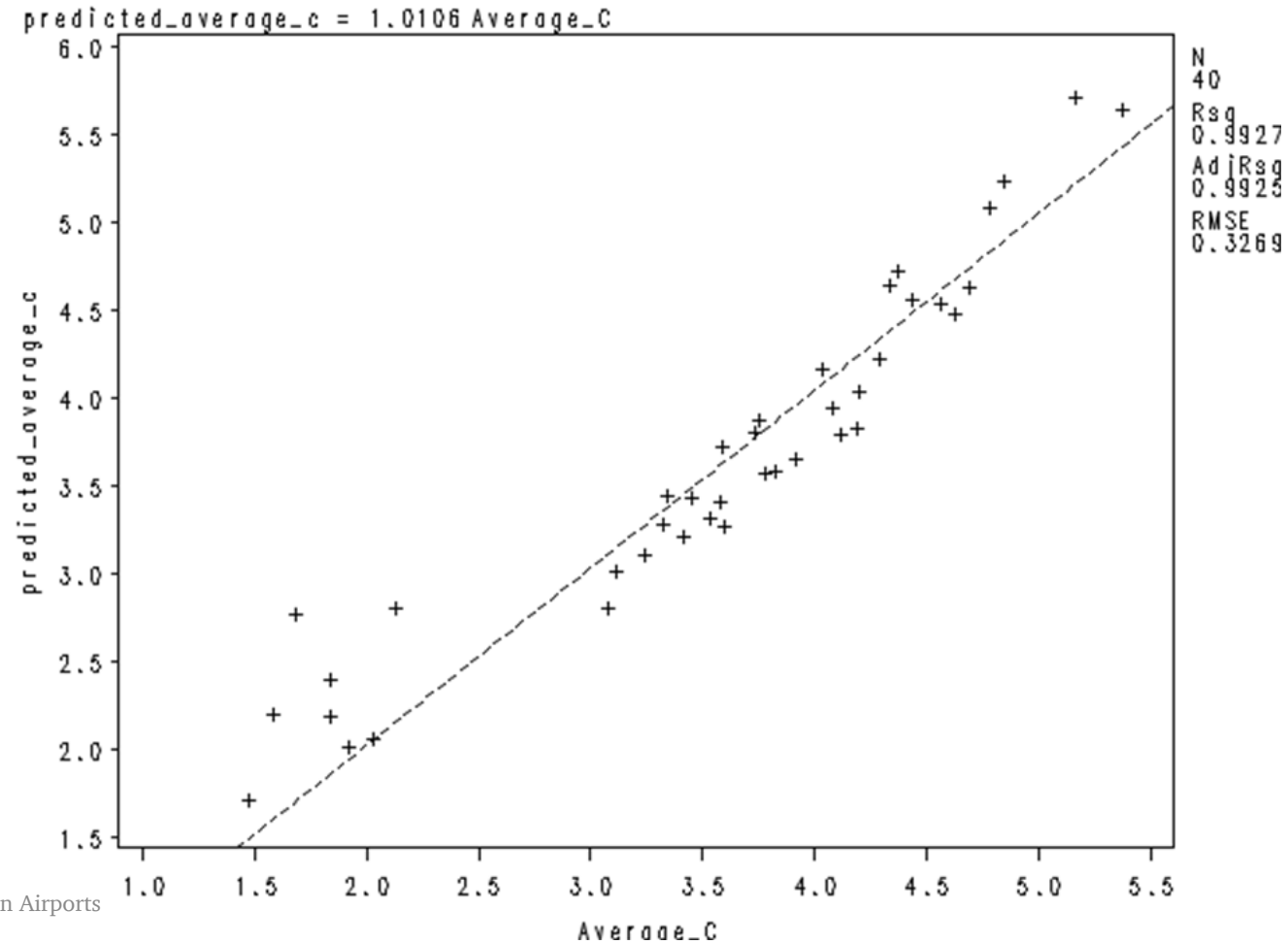
(arbitrary example — not an actual CATSA checkpoint)

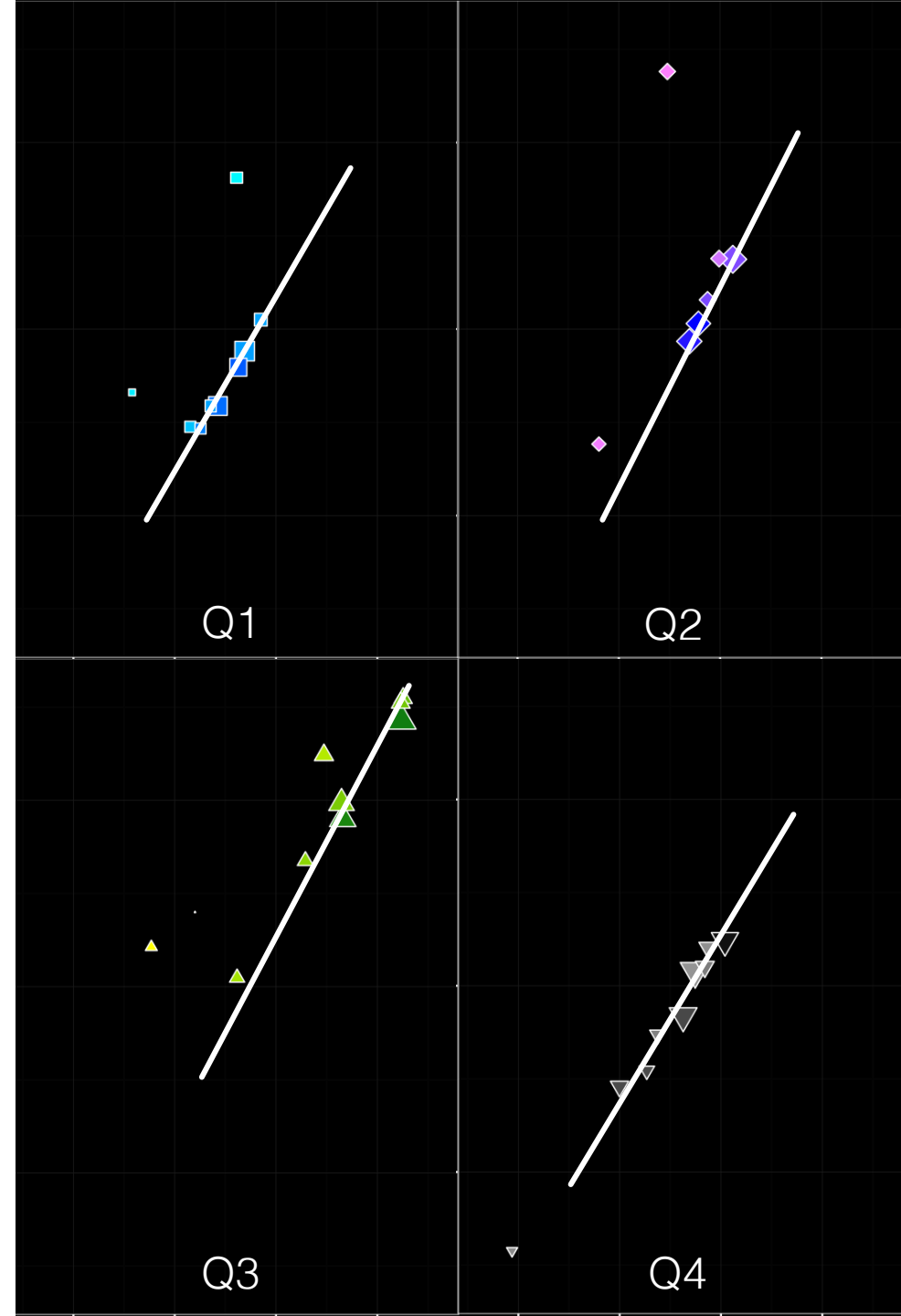
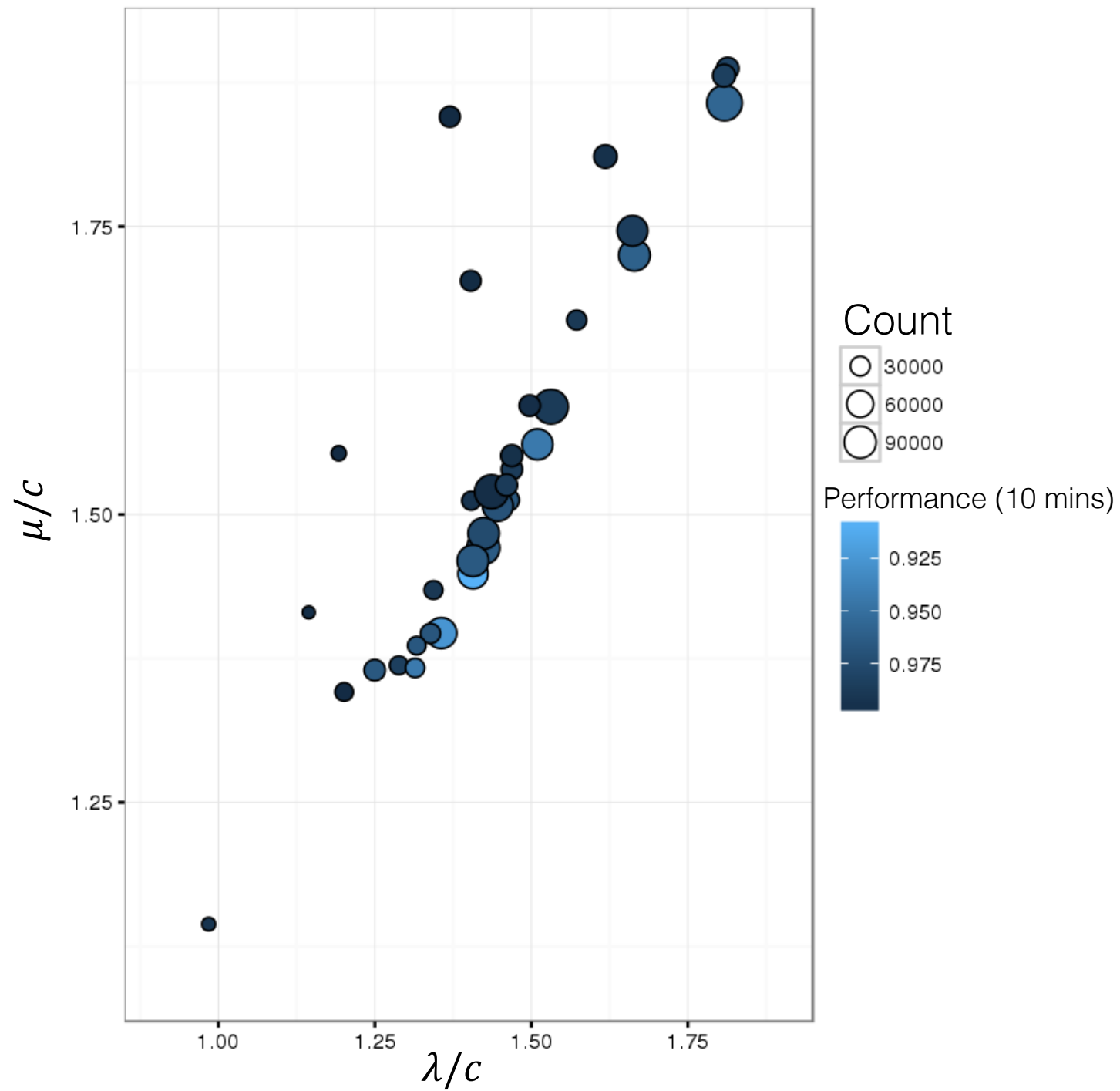
Cluster		Actual # Servers	Pred # Servers
	Week-day	0:00 4:00	0.136 -
		4:00 8:00	5.375 5.643
		8:00 12:00	4.629 4.474
		12:00 16:00	4.193 3.829
		16:00 20:00	3.775 3.572
		20:00 0:00	1.585 2.198
	Week-end	0:00 4:00	0.212 -
		4:00 8:00	4.560 4.535
		8:00 12:00	3.918 3.650
		12:00 16:00	3.411 3.205
		16:00 20:00	3.599 3.264
		20:00 0:00	1.471 1.701

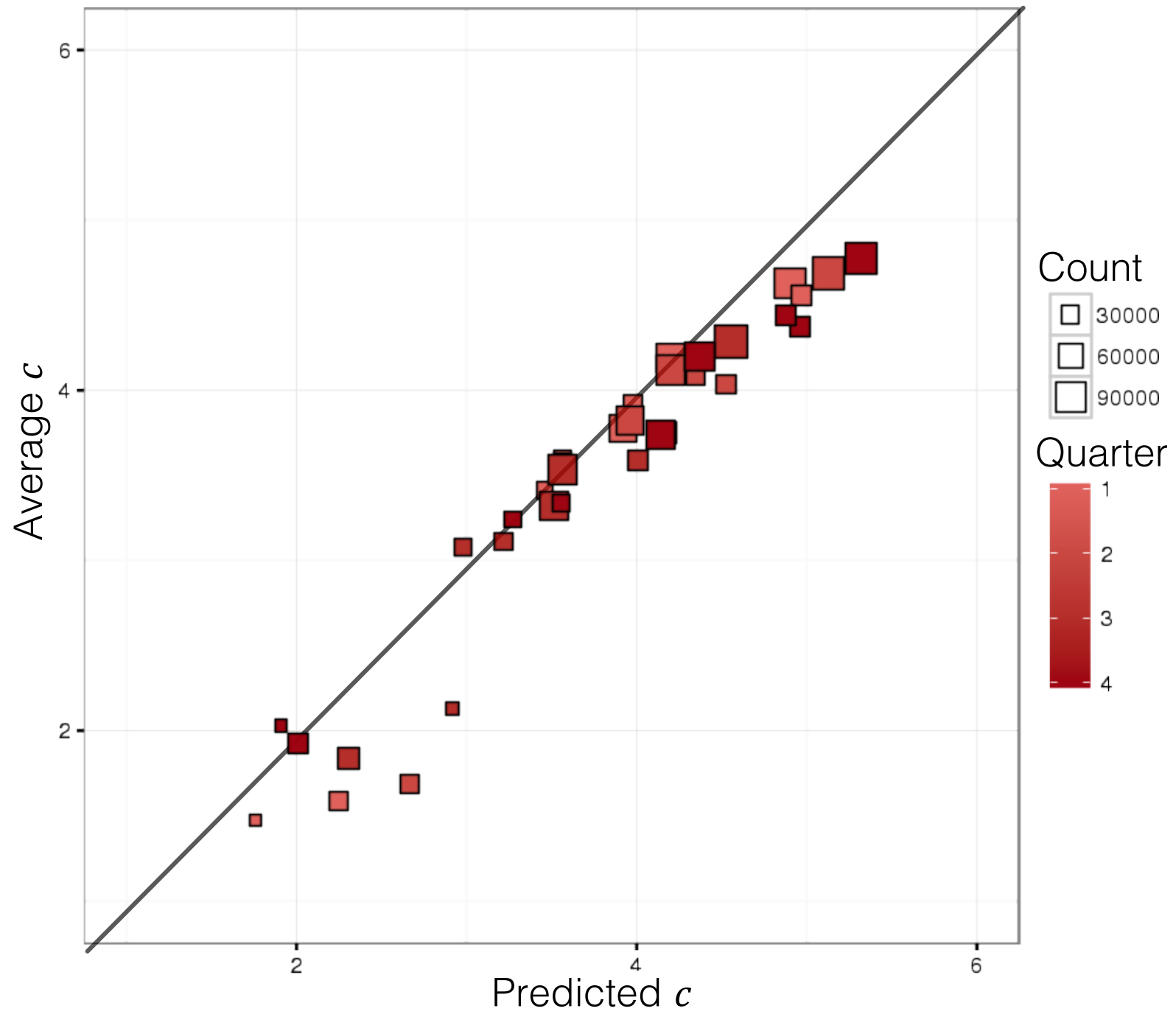
Illustration

Departure

(arbitrary example — not an actual CATSA checkpoint)







Consulting Post-Mortem

Clients were mathematicians: knew what they wanted, and had a pretty good idea of how to get it

Clients could probably have done this on their own, but were not being taken seriously by non-technical stakeholders

Another approach (simulation) had been used and failed to provide useful results, so there was scepticism on the part of stakeholders

Academics were not highly regarded, so consultant angle had to be played up

We were able to help them find a data reporting error through preliminary analysis, which helped solidify our credentials

Still in use as of 2018, although the model performs better at the national level than at the airport level.