Bilge K., David L., Iryna T., Liam D.

Introduction

Passenger flow through airport checkpoints is critical to maintain a high level of operational efficiency and service quality. As air travel demand can be unpredictable at times, airport managers must accurately estimate passenger arrival patterns to allocate servers accordingly. Reliable estimates are essential to determine the appropriate number of active servers needed at any given time to avoid long wait times and ensure customer satisfaction. The analysis of predicting passenger flow and controlling server allocation allows managers to optimize recourses.

ABC aviation is an international cooperation that manages many airports. As part of their continuous improvement initiative, ABC hired Professor Boily to build a model predicting service effectiveness of airport security lines using temporal cycle patterns. Although this model has been helpful in predicting service quality, Professor Boily's model assumes demand is based purely on cyclical temporal patterns. This is a very strong assumption given the aviation industry lives at the heart of the globalization. Thus, if any there are any drastic changes in demand, the model's performance may deteriorate. Some examples that could cause a change of demand:

1. Airline routes and scheduling – cancellations and new additions for operational efficiency.
2. Geopolitical environment – changes in visa requirements
3. Global Conflicts - wars for example, Russia-Ukrainian war
4. Special Events – Olympics, Super Bowl, Royal Weddings etc.

To address these blind spots, ABC aviation has asked their team of data scientist to build a similar prototype of Boily's models built on forecasted demand. By doing so, the model will be more versatile and adaptable to changing circumstances.

Part I – Preliminary Data Exploration

Source or datafile name	Variable	Type	Range	Description	NAs	Examples
dat_P_sub_c.csv, dat_f_sub.csv	Flight_ID	int	18095 – 21678	Unique identifier assigned per flight	0	18095
dat_P_sub_c.csv	Departure_Date	date	2028-09-01 – 2028-12-31	The date of flight	0	2028-09-01
dat_P_sub_c.csv, 20262030.csv	Pass_ID	int	5348206 – 6438752	Unique identifier assigned per passenger	0	5348206
dat_P_sub_c.csv, 20262030.csv	S2	date time	2028-09-01 6:37:00 AM – 2028-12-31 8:09:00 PM	Scan time of passenger when exiting the queue	0	2028-09-01 6:37:00 AM
dat_P_sub_c.csv	Wait_Time	int	1 – 75	Time a passenger waits in the queue for a service counter	16,871	1
dat_P_sub_c.csv	C_Start	int	1 – 3	Count of active service counters at the time a passenger is scanned	0	1
dat_P_sub_c.csv, dat_f_sub.csv, 20262030.csv	Sch_Departure	date time	2028-09-01 8:06:00 AM – 2028-12-31 8:06:00 PM	Scheduled departure time of flight	0	2028-09-01 8:06:00 AM
dat_P_sub_c.csv, dat_f_sub.csv	Act_Departure	date time	2028-09-01 8:06:00 AM – 2028-12-31 9:40:00 PM	Actual departure time of flight	0	2028-09-01 8:06:00 AM
20262030.csv	order	int	1186 - 9974043	Ordinal ranking / order	0	9968506
Created in new file	Sch_Dep_Day	int	1 – 28	Day number of scheduled flight	0	10
Created in new file	Sch_Dep_Month	int	9 – 12	Month number of scheduled flight	0	9
Created in new file	Sch_Dep_Year	int	2028	Year number of scheduled flight	0	2028
Created in new file	Arrivals	int	0 - 230	Number of passengers arriving per hour	0	120
Created in new file	Average_Lambda	float	0.0000 – 3.8418	Average arrival rate per S2 hour	0	0.0508
Created in new file	Hour_Bin	date time	2028-09-01 6:00:00 AM – 2028-12-30 10:00:00 PM	Binned S2 per hour	0	2028-09-01 6:00:00 AM

Created in new file	Poisson_Pass_Rate	float	0.0 – 1.0	Probability of hour passing KS Exp Test	0	0.6
Created in new file	Wait_Time_Bin	char	1-5 – 71-75	Binned Wait_Time	1,700	6-10
Created in new file	min_lambda_bin	categ	0.0508 – 2.6038	Binned min lambda after k-means clustering	0	1.9436
Created in new file	mean_lambda_bin	float	0.4161 – 2.9848	Mean lambda per k-means cluster	0	2.2197
Created in new file	max_lambda_bin	float	0.6381 – 3.8418	Binned max lambda after k-means clustering	0	2.5881

Table 1. Data dictionary for variables that were used in the analysis. The file dat_P_sub_c.csv contains primarily passenger data and metadata on flights. dat_f_sub.csv contains primarily flight data and metadata on passengers.

Table 1 provides a summary of all key variables used in this analysis, along with their sources, data types, ranges, descriptions and examples. This table is crucial for understanding the structure of the datasets. Variables include flight-level identifiers (e.g. Flight_ID, Sch_Departure, Act_Departure), passenger level identifiers (e.g. Pass_ID, S2, Wait_Time, C_Start) and derived features (e.g. Average_Lamda, Arrivals). Derived features allow modelling passenger arrival rates, queueing behavior and server requirements. Most significantly Average_Lambda, min_lambda_bin, mean_lambda_bin, and max_lambda_bin play a significant role in clustering hours and linking clusters to operational decisions such as staffing levels. Additionally, Poisson_Pass_Rate indicates the degree to which the observed data aligns with a Poisson process assumption, ensuring the validity of modeling approaches. Building on this data exploration, the next step is to develop a model that can effectively forecast passenger arrivals to make operational decisions such as staffing.

Clustering Data with similar Poisson Process

Clustering data begins with selecting an appropriate granularity. The granularity chosen to evaluate Poisson process, is by the hour. This choice is quite arbitrary, made for the convenience for analysis, explainability to business partners and ease of implementation as workforces are typically scheduled into hourly shifts.

The objective is to build a model to forecast lambda, the defining parameter of the Poisson process, for any hour based on future scheduled information. The idea is that the arrival rate at security checkpoints depends on the number of passengers scheduled to depart. Given airlines and airports recommend to passengers to arrive 2 hours prior to departure, the model will use predictors $x_1 = scheduled\ passengers\ taking\ off\ in\ 1\ hour$ and $x_2 = number\ of\ passengers\ taking\ off\ in\ 2\ hours$ to predict $\hat{\lambda}$. To achieve this, a training set for the initial regression model must be built.

Note that $\hat{\lambda}$ is estimated from the interarrival times. Given S2 is rounded to the nearest minute, there are many passengers where the timestamp is identical. Under a Poisson process assumption, this is not possible. Thus, seconds are added by sampling from a uniform distribution on [0,60] to add seconds. After calculating the interarrival rate between each arrival, the avg_lambda is obtained. To confirm the assumption of the Poisson process, a KS test on the interarrivals with the exponential distribution of with avg_lambda (alpha = 0.05) is performed. Depending on the cluster, this test may pass/or fail depending on the values of the selected from the uniform distribution. This simulation is run 10 times to get a sense of the impact of the uniform sample. A preview of output of the simulation, which is used as the training set of the regression model is below (**Table 2**).

Hour Bin	Arrivals Count	Successful KS Tests	Unsuccessful KS Test	$\hat{\lambda}$	x_1	x_2
9/1/2028 9:00	25	9	1	0.581001454	51	0
9/1/2028 10:00	39	10	0	0.702791999	0	38
9/1/2028 11:00	14	10	0	0.269327587	38	24
9/1/2028 12:00	32	10	0	0.540933312	24	61

Table 2. Hourly Simulation Summary and Estimated Arrival Rates $\hat{\lambda}$ for Regression Training

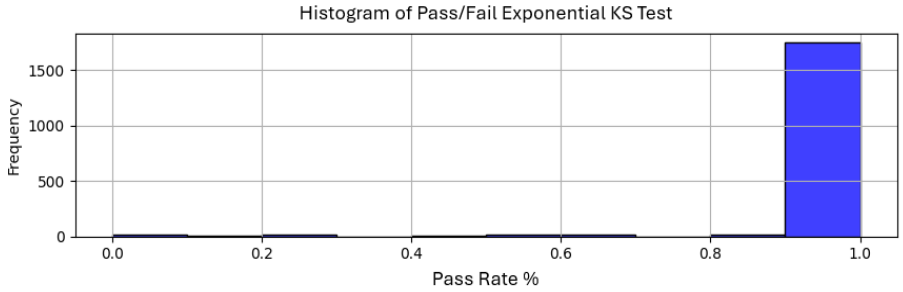

Figure 1. Histogram of Distribution of Hourly KS Test Pass Rates for Exponential fit.

Based on **Table 2** and **Figure 1**, hours generally pass the KS test. Further investigation is required to access why some do not. We further illustrate this on the heat map below (**Figure 2**).

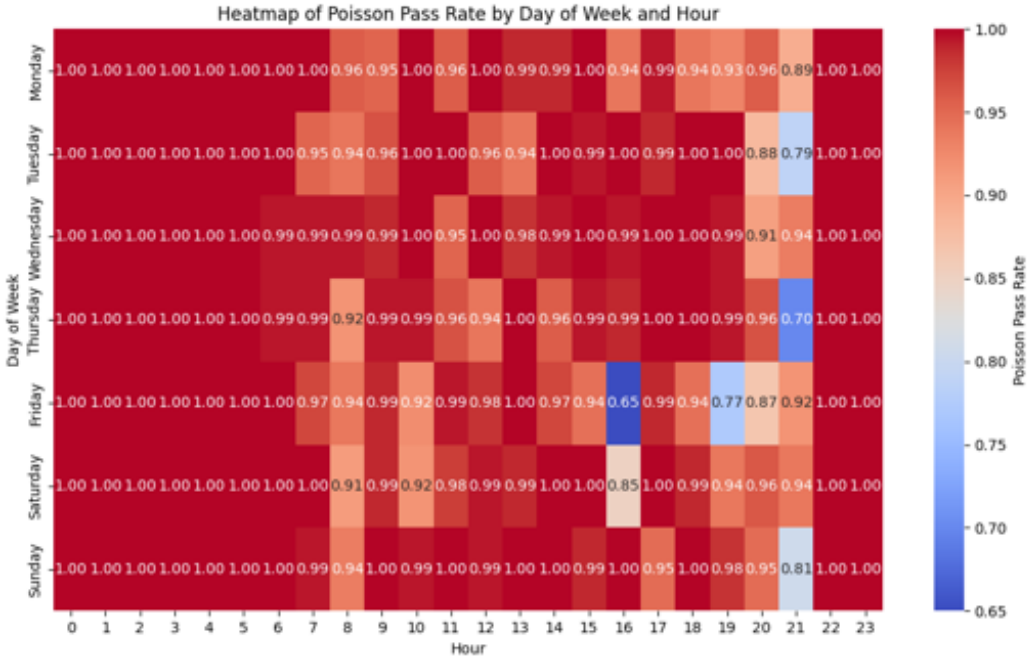

Figure 2. Heatmap of Poisson Pass Rate by Day of Week and Hour.

Figure 2 shows the pass rates of every hour of each day, on whether they follow a Poisson Distribution. In Layman's terms, this shows how often arrivals in each hour follow a standard Poisson distribution with parameter λ. The heatmap shows that an overwhelming amount of our data, the vast majority follow a Poisson distribution with parameter λ, thus validating the modeling assumption needed for estimating λ using interarrival times. This supports the application of a regression-based method for forecasting arrival rates, as the foundation assumption of a Poisson process is met in most hourly segments.

Following, the linear regression model is $\hat{\lambda} = b_1 x_1 + b_2 x_2 + \epsilon$, such that ϵ is distributed normally with a mean of 0 and a standard deviation of σ^2, which is visualized below. The adjusted R^2 is 0.9088 via **Figure 4**. The intercept is intentionally left out, since if there are no scheduled flights in the next two hours, there are also no arrivals. This assumption was confirmed by failing to reject the null hypothesis.

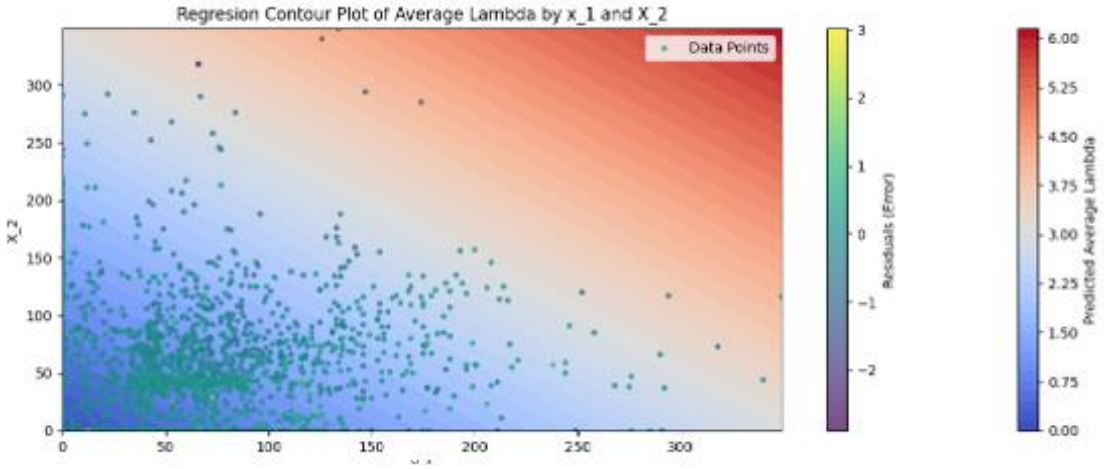

Figure 3. Regression Contour plot of $\hat{\lambda}$ by x_1 and x_2.

The regression contour plot in **Figure 3** provides a visual representation of how the predicted arrival rate $\hat{\lambda}$ varies based on values of x_1 and x_2, the number of passengers scheduled to depart in the next one and two hours, respectively. The color gradient illustrates increasing values of $\hat{\lambda}$, confirming that the model behaves as expected: predicted arrival rates increase as the number of scheduled passengers increases. The dense clustering of data points along the contour lines also indicates a stable and consistent relationship between the predictors and the response variable. This alignment between the data and the contour levels further supports the effectiveness of the regression model and the validity of using scheduled flight data as a predictor for passenger arrival rates at security. Using standard OLS, the model is $\hat{\lambda} = 0.0086x_1 + 0.0090x_2$. The summary of the model is below.

```
R-squared: 0.9089
Adjusted R-squared: 0.9088

Full Model Summary:
                            OLS Regression Results
==============================================================================
Dep. Variable:         Average_Lambda   R-squared (uncentered):                0.909
Model:                            OLS   Adj. R-squared (uncentered):           0.909
Method:                 Least Squares   F-statistic:                       1.444e+04
Date:                Sun, 06 Jul 2025   Prob (F-statistic):                     0.00
Time:                        12:06:23   Log-Likelihood:                      -788.32
No. Observations:                2897   AIC:                                   1581.
Df Residuals:                    2895   BIC:                                   1593.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
X1             0.0086      0.000     76.951      0.000       0.008       0.009
X2             0.0090      0.000     80.258      0.000       0.009       0.009
==============================================================================
Omnibus:                      948.102   Durbin-Watson:                   2.070
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            17003.833
Skew:                           1.081   Prob(JB):                         0.00
Kurtosis:                      14.670   Cond. No.                         1.92
==============================================================================
```

Figure 4. R Output Summary of Regression Model Estimating $\hat{\lambda}$ Using Scheduled Passenger Counts.

All in all, the regression model demonstrates strong potential for accurately clustering hours based on predicted arrival rates. While minor limitations exist, especially around early morning and final closing hours, the model overall captures the flow of passenger traffic throughout the day. By leveraging scheduled flight data to predict $\hat{\lambda}$, the model provides a practical, interpretable, and data-driven method for anticipating security checkpoint demand. These predicted $\hat{\lambda}$ values are then used to group hours into distinct clusters, each representing a Poisson process with a specific arrival intensity. This clustering not only reflects operational demand levels but also sets the foundation for optimizing staffing decisions, improving passenger experience, and informing future airport resource planning.

Lambda-Based Clustering for Dynamic Server Allocation

Lambda Range	Mean Lambda	Number of Scanned Passengers	Number of Hours
[0.00, 0.00]	0.000	94	1,088
(0.00, 0.64]	0.416	17,250	771
(0.64, 1.04]	0.861	71,113	1,438
(1.04, 1.44]	1.229	95,200	1,362
(1.44, 1.94]	1.660	95,025	1,039
(1.94, 2.59]	2.220	72,171	634
(2.59, 3.84]	2.985	45,998	302

Table 3. dat_P_sub_c Clusters of Arrival Rates Based on Passenger Arrival Volume of Each Hour.

In contrast to Boily's clustering method, the clustering of lambda values via numbers of passengers scanned is much more dynamic and granular. Additionally, it doesn't require an entire new set of annual data to capture major changes in airline routes and scheduling, geopolitics, global conflicts, and special events as mentioned in the introduction. This lambda-based clustering approach groups hours by their similarity in arrival rates, having operational demand levels, allowing for tailored staffing and resource allocation strategies. Each cluster reflects a distinct arrival rate at security. Arrival rates $\hat{\lambda}$ were grouped into 6 clusters, apart from the first row of **Table 3**, which represents the 0 case, which are outliers.

Lambda Range	Mean Lambda	Counter 1	Counter 2	Counter 3	Mean Counter
(0.00, 0.64]	0.416	98.1%	1.9%	0.0%	1.02
(0.64, 1.04]	0.861	91.5%	8.4%	0.1%	1.09
(1.04, 1.44]	1.229	79.3%	20.5%	0.2%	1.21
(1.44, 1.94]	1.660	54.9%	44.1%	1.1%	1.46
(1.94, 2.59]	2.220	30.6%	68.1%	1.3%	1.71
(2.59, 3.84]	2.985	23.5%	68.7%	7.9%	1.84

Table 4. dat_P_sub_c Service Counter Usage by Lambda Cluster.

Since managers oversee service counters and can adjust number of active servers to accommodate passenger traffic, the distribution of passengers per counter is crucial. **Table 4** shows the distribution of open service counters, as well as the average number of service counters open per cluster. These values were generated by developing counts of C_Start (the number of active servers open when a passenger arrived) for every 5-minute block of the arrival hour, then the counts and averages taken per cluster.

This distribution highlights how staffing decisions scale with operational demand. Not only does the average number of counters increase with higher arrival rates, but the proportion of counters shifts noticeably. For instance, in the [0.64, 1.04] lambda cluster, 91.5% of passengers were processed with a single counter. However, in the highest lambda cluster [2.59, 3.84], this drops to just 23.5%, with nearly 70% of processing shifting to two counters and an additional 7.9% requiring three. This demonstrates that airport operations do not rely solely on averages but instead reflect strategic adjustments in real time, transitioning toward more parallel processing as demand rises to sustain performance targets.

Mean Lambda	Lambda Range	Avg Wait Time	NA	1-5 min	6-10 min	11-15 min	16-20 min	21-25 min	26-30 min	31-35 min	36-40 min	41-45 min	46-50 min	51-55 min	56-60 min	66-70 min	71-75 min
0.416	(0.00, 0.64]	2.82	22%	97%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
0.861	(0.64, 1.04]	3.83	12%	89%	98%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
1.229	(1.04, 1.44]	5.58	12%	74%	93%	98%	100%	100%	100%	100%	100%	100%	100%	100%	NA	100%	100%
1.660	(1.44, 1.94]	6.58	14%	61%	86%	94%	98%	99%	100%	100%	100%	100%	100%	100%	100%	100%	100%

2.220	(1.94, 2.59]	6.88	16%	61%	85%	95%	98%	99%	100%	100%	100%	100%	100%	100%	100%	100%	100%
2.985	(2.59, 3.84]	9.35	21%	52%	75%	87%	94%	98%	99%	100%	100%	100%	100%	100%	100%	100%	100%

Table 5. dat_P_sub_c passenger wait times for service counter per cluster.

Using only cases where Wait_Time was available, **Table 5** displays the cumulative distribution of wait times per cluster. In the (1.94, 2.59] range with an average arrival rate of 2.220, the average wait time is 6.88 minutes, and 95% of passengers are processed within 11–15 minutes. Yet, in the (2.59, 3.84] range with an average arrival rate of 2.985, the average wait time is 9.35 minutes, and only 87% of passengers are processed within 11–15 minutes. This average wait time increases, and number of customers served in 15 minutes decrease as arrival rates increase. The goal of serving 85% of customers within 15 minutes becomes more difficult, requiring potential staff expansions.

Mean Lambda	Lambda Range	Avg Wait Time	Est. Service Rate (û)	Est. traffic intensity (ρ̂)	5 min	10 min	15 min	20 min	25 min	30 min	35 min	40 min	45 min	50 min	55 min
0.416	(0.00, 0.64]	2.82	0.64	0.65	79%	93%	98%	99%	100%	100%	100%	100%	100%	100%	100%
0.861	(0.64, 1.04]	3.83	1.07	0.80	72%	90%	97%	99%	100%	100%	100%	100%	100%	100%	100%
1.229	(1.04, 1.44]	5.58	1.39	0.89	60%	82%	92%	96%	98%	99%	100%	100%	100%	100%	100%
1.660	(1.44, 1.94]	6.58	1.80	0.92	54%	77%	89%	94%	97%	99%	99%	100%	100%	100%	100%
2.220	(1.94, 2.59]	6.88	2.36	0.94	53%	76%	88%	94%	97%	98%	99%	100%	100%	100%	100%
2.985	(2.59, 3.84]	9.35	3.09	0.97	42%	66%	80%	88%	93%	96%	97%	98%	99%	99%	100%

Table 6. dat_P_sub_c service rates per cluster

The average wait time of a passenger in the queue was extracted from **Table 5** using the dat_P_sub_C dataset, where Wait_Time data was available. Using the average arrival rate (Mean Lambda) and the average wait time, we estimated the service rate for each cluster. The service rate μ was computed using the formula:

$$\hat{\mu} = \frac{\bar{W}q\lambda \pm \sqrt{(\bar{W}q\lambda)^2 + 4\bar{W}q\lambda}}{2\bar{W}q\lambda}$$

This service rate was then used to calculate the estimated traffic intensity $\hat{\rho} = \frac{\lambda}{\hat{\mu}}$. For all clusters, traffic intensity was found to be less than 1, indicating the queues are stable and service capacity is sufficient. With traffic intensity established, we estimated the quality-of-service levels using the M/M/1 queue cumulative distribution function:

$$\hat{\mathbb{Q}} = 1 - \hat{\rho}e^{-(\hat{\mu}-\lambda)x}$$

These quality-of-service levels are shown in **Table 6** and are based on x = 15-minute time increments. In general, as the average arrival rate increases, the percentage of passengers that are serviced within fifteen minutes decreases. For every cluster, 100% of passengers should be serviced within 55 minutes, or as fast as within 25 minutes when the arrival rate is low.

This analysis is critical for informing cluster-based staffing and optimizing service allocation. Maintaining at least 85% of passengers being served within 15 minutes requires dynamically adjusting the number of servers. Comparing per-hour lambdas with cluster-based lambdas, the cluster approach offers improved performance, achieving a stronger balance between service quality and operational efficiency.

Moving From M/M/1 to M/M/C

To evaluate the effectiveness of the cluster-based approach in predicting operational needs, we examine the relationship between arrival rates λ, number of counters opened, and service rates across different λ-clusters. The focus is on two primary outputs which are how closely actual service rates align with predicted ones and how well the number of counters can be predicted based on λ values.

Lambda Range	Mean Counter	Predicted Counters
(0.00, 0.64]	1.02	0.769460694
(0.64, 1.04]	1.09	1.05022439
(1.04, 1.44]	1.21	1.258637518
(1.44, 1.94]	1.46	1.494179265
(1.94, 2.59]	1.71	1.794296255
(2.59, 3.84]	1.84	2.199478294

Figure 5. Predicted vs. Observed Number of Service Counters per λ Range.

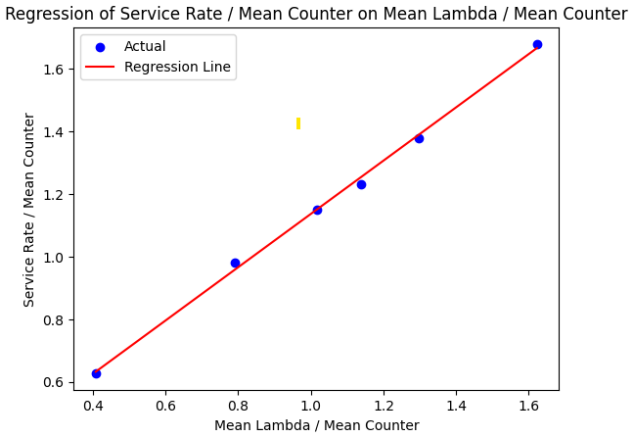

Figure 6. Linear Relationship Between Arrival Rate per Counter and Observed Service Rate.

Figure 6 and **5** show a close alignment between the predicted number of service counters and the actual mean counters observed per lambda cluster provides strong support for the assumptions underpinning the model. In particular, the regression of service rate per counter against arrival rate per counter, illustrated in the regression graph, demonstrates a nearly perfect linear relationship, with an R^2 value of 0.998 via **Figure 7**. This high degree of fit suggests that, on average, counters are staffed proportionally to demand, and that the modeling approach accurately captures how operational staffing adjusts to fluctuations in arrival rates. The consistency between predicted and actual values indicates that managers likely respond systematically to passenger volumes, opening the appropriate number of counters to maintain quality-of-service targets. This regression not only validates our assumption that λ/μ ratios at each counter remain relatively stable across clusters, but shows how well the clustering framework for guides dynamic staffing decisions.

OLS Regression Results

Dep. Variable:	Service Rate / Mean Counter	R-squared:	0.998
Model:	OLS	Adj. R-squared:	0.997
Method:	Least Squares	F-statistic:	1975.
Date:	Sun, 06 Jul 2025	Prob (F-statistic):	1.53e-06
Time:	19:39:40	Log-Likelihood:	16.824
No. Observations:	6	AIC:	-29.65
Df Residuals:	4	BIC:	-30.06
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.2851	0.021	13.371	0.000	0.226	0.344
Mean Lambda / Mean Counter	0.8515	0.019	44.445	0.000	0.798	0.905

Omnibus:	nan	Durbin-Watson:	1.921
Prob(Omnibus):	nan	Jarque-Bera (JB):	0.382
Skew:	0.330	Prob(JB):	0.826
Kurtosis:	1.955	Cond. No.	5.68

Figure 7. R Summary of OLS Regression of Clustering Model.

Assuming the goal is to have 85% of passengers process in under t =10 min, we can recover the following counter predictions. Effectively this means increasing counter when λ > 1.5.

Looking at the predicted service rate vs actual service rate, and the predicted number of counters vs mean counters, the assumption made by our model seem to hold up well. Although the caveat, is that further testing on unseen data is necessary to confirm generalizability.

Mean Lambda (λ)	Lambda Range	Mean Counter (c)	Service Rate (μ)	μ/c	λ/c	Pred. Service Rate	t<5	t<10	t<15	T<20	T<25	T<30	T<35	T<40	T<45	T<50	T<55
0.416	(0.00, 0.64]	1.02	0.64	0.6275	0.4078	0.65	79%	93%	98%	99%	100%	100%	100%	100%	100%	100%	100%
0.861	(0.64, 1.04]	1.09	1.07	0.9817	0.7899	1.04	67%	87%	95%	98%	99%	100%	100%	100%	100%	100%	100%
1.229	(1.04, 1.44]	1.21	1.39	1.1488	1.0157	1.39	61%	83%	92%	97%	98%	99%	100%	100%	100%	100%	100%
1.66	(1.44, 1.94]	1.46	1.8	1.2329	1.1370	1.83	61%	83%	93%	97%	99%	99%	100%	100%	100%	100%	100%
2.22	(1.94, 2.59]	1.71	2.36	1.3801	1.2982	2.38	58%	81%	91%	96%	98%	99%	100%	100%	100%	100%	100%
2.985	(2.59, 3.84]	1.84	3.09	1.6793	1.6223	3.07	35%	57%	71%	81%	87%	91%	94%	96%	97%	98%	99%

Figure 8. Quality of Service Wait Time Distributions by λ Range.

Next Steps to Validate This Project:

1. Based on years20262030 CSV file, calculate x_1 and x_2 for every hour, then predict lambda.
2. Add Avg number of counters
3. Based on lambda and counters, use regression model (λ, μ, counter) to create prediction of service rate for that hour. Compare with actual service rates of that hour.
4. Observe the difference in forecasted service rates vs actual service rate. If these rates are similar, then a successful model independent on temporal cycles has been created.

To improve overall approach, one should consider reducing the ranges of lambda clusters to create more clusters. These cluster ranges were kept large for the convenience of visualizing results and working though methodology. The same is true with the regression model to predict lambda for a given hour. Rather than using passengers scheduled to take off in 1 and 2 hours, it might be better to consider 4 increments of 30 min. This model will do a better job at detecting users arriving early in the hour and taking off later within the same hour. Other issues with this model is that it may be irresponsible to extrapolate to beyond the lambda in the test set. Although linear regression is a global structure, very large lambda could cause the model to have a traffic intensity > 1 and diverge.

Summary

All in all, this report presents a data-driven approach to improving airport security checkpoint operations by forecasting passenger arrival rates and dynamically allocating service counters. Using scheduled flight data, a regression model was developed to estimate hourly arrival rates $\hat{\lambda}$, which were then used to cluster hours into groups with similar demand profiles.

These clusters informed optimal staffing levels based on real-time passenger volumes, transitioning from a basic M/M/1 queueing model to a more realistic M/M/c framework. The analysis showed strong alignment between predicted and actual service counters, with regression results validating the proportional relationship between arrival and service rates. Furthermore, quality-of-service metrics indicated that maintaining service targets, such as processing at least 85% of passengers within 15 minutes, becomes increasingly challenging as demand rises, but can be addressed through appropriate staff scaling. Overall, the model demonstrated operational feasibility and robustness, offering airport managers a practical tool for efficient, demand-responsive resource planning.

Appendix

Teammate	Contributions
Bilge Kirilmis	Writing
David Liu	Data dictionary, Writing
Iryna Tkachenko-Riek	Data dictionary, M/M/1 queuing model, editing
Liam Dubé	Establish business scenario, Data Exploration, Stationarity, Independence testing, Clustering methodology design, Regression Model Design and Model Testing, Writing