

Capstone CYO Project

Bill James (BCDL) / jamesbcdl@gmail.com (mailto:jamesbcdl@gmail.com)

May 2, 2019

Executive Summary

The Project This document reports on the examination of one year's worth (2018) of service desk data - incident and service request tickets - submitted as part of the overall technology group support activity. The data was sourced from the organization's Cherwell application. This analysis is intended to provide insight into duration times, Service Level Agreement (SLA) target achievement, group performance, and, if possible, to provide a mechanism to forecast a given duration time - and alternatively, when an SLA target might be breached.

Fiindings The data, while extensive, has certain flaws, the biggest of which is that it reflects inconsistent recordkeeping on part of the staff. Most staff close out a ticket immediately, but some batch their recordkeeping to take place weeks or even months after the fact. As a result, there are many outliers that have no real reason for being there, and therefore the findings have to be considered provisional - until the data quality can be improved.

The data indicate that:

- SLA targets - with some exceptions - are almost never met
- Prioritization, particularly for service requests, is not practical or meaningful
- There are wide variations in staff performance (in average event duration times)
- Until recordkeeping is more consistent, and prioritization is more scientific, it is not possible to predict ticket closure times beyond basic averaging

On the positive side, it is also true that focusing on better recordkeeping and more disciplined prioritization of a limited number of problem categories could significantly improve overall metrics and performance.

Recommendations The data suggests that the service desk management and team take the following steps:

- *Drive overall duration times down* through a combination of more consistent recordkeeping across the group and more dedicated management / oversight to ensure this is happening. This will enable statistics to be far more reliable because outliers will be eliminated, but more importantly, it is an easy path to much improved group performance metrics.
- *Rethink SLAs* so that true priority 1's are acted on first. Part of this is to rethink the prioritization scheme; nearly all (99%) of the service requests are prioritized as priority 3's effectively having no prioritization at all. Instead, it becomes a large pile of equally important tickets, making it difficult for staff to differentiate between the truly important. Management may want to consider a 2-level prioritization scheme for both incidents and service requests - for example, 1 day or 2 days for incidents, and 5 days / 10 days for service requests. This, of course, depends on whether or not the clients being supported feel these are fast enough resolution times. That may take some interviewing / surveying to confirm.
- *Stress individual improvement* by making staff aware of their statistics. The wide variations in performance suggest it is possible the staff are unaware of the importance management is putting on SLA achievement and duration times. If staff were shown their performance statistics and provided with weekly updates, it is likely we would see a significant change in their performance.
- *Overall* there are several overarching recommendations:

- Management must be clear on the metrics they wish to achieve, communicate these to the staff, and build the necessary fields into Cherwell to gather the data to track them. For example, a more granular taxonomy for ticket classification, and perhaps the revised prioritization scheme recommended above.
- Adopt a weekly aging report to focus staff attention on what did not get closed within (the revised) SLA targets - and discuss why (as a team). A sample is included in the appendix.
- Run the statistics produced by this code periodically - quarterly or semi-annually - to see what progress is being made toward management goals.

Background and Overview

The Service Desk is seeking to improve its services by better managing customer tickets, particularly as they relate to SLAs (service level agreements). There are several interconnected problems to solve:

- *Ticket Classification Accuracy* (both incidents and service requests) so tickets are properly prioritized and enable truly urgent items to be addressed first.
- *Customer Expectation Transparency* so that clients have a more accurate window for ticket resolution. Workflow Improvements are needed to address the steps and tasks that slow the process down and lack of consistency.
- *Breach Forecasting* by determining what factors (alone or in combination) have the highest probability of causing breaches of the SLA, and identifying those for action.
- *Ticket Forensic Review* of tickets that breached their SLA through systematic examination of all events/logs that led to the breach.

The **desired end state** is that clients submitting tickets will receive immediate confirmation that their request has been received followed by a highly-reliable estimate when the request will be resolved. TSG (Technology Service Group) staff can, with great confidence, assume that any tickets on the top of the queue are the ones most important to address. Improved classification and forecasting reduce SLA breaches, and the management team is able to set more precise SLA parameters with no degradation in accuracy. The workflow operates consistently among all team members, and in those rare cases when an SLA is not met, all of the necessary data and reports are available to identify the problems so they can be addressed.

In looking at the desired end state:

- Some elements are management activities and are already possible. These include:
 - Providing clients with immediate confirmation
 - Enabling staff to address the highest priority items first
 - Consistent workflow among all team members
- The others can be addressed - at least in part - through management steps supported by insights from an analysis like this one:
 - Improved classification and forecasting
 - Setting more precise SLA parameters with no degradation in accuracy
 - Data and reports for forensics

This analysis looks at the full year of 2018 incident and service request data to gain insight to address these latter three elements. The analysis considers the following questions for the data collectively and for both incidents and service requests separately:

- What are the type and nature of the problems?
- What kinds of problems are more prevalent?

- What are the characteristics of those problems?
 - Minimum to address
 - Maximum times to address
 - Average times to address
 - The standard deviation of duration times

The analysis also seeks to determine if there are any discernable *patterns* in the data, for example:

- Do some kinds of problems take longer - and which?
- Do some owners resolve problems more quickly than others?
 - Which owners?
 - Which problems in particular?
- Can we accurately predict how long those problems will take?

Approach

A spreadsheet was extracted from Cherwell covering all events (incidents and service requests) for calendar 2018. The data set included a number of fields, and the following were selected for use in the analysis:

- Incident ID
- Created Date Time
- SLA Resolve By Deadline
- SLA_Time
- Closed Date Time
- Dur_Time
- Status
- Short Description
- Description
- Category
- Incident Type
- Call Source
- Cause
- Owner_ID
- Priority

The analysis then took the following steps:

- Examined the shape of the data and the general characteristics
- Determined that the data contained many outliers that skewed the data
- Chose a better “maximum duration” value for a more normal data set
- Split the analysis to look at incidents and service requests separately
- Broke these two groups down into 89 lower-granularity problem categories
- Generated a number of outputs for management, including:
 - Basic descriptive statistics to be compared with SLA goals
 - Views of the top problem generators and their characteristics
 - Views of the problem owners and their speed and efficiency in addressing
- Provided several views of how to predict likely problem duration
- Developed a set of recommendations on how to proceed

Before proceeding, it will be useful to understand a few terms as used in the analysis:

- An *event* refers to any service desk incident or service request
- An *incident* refers to an urgent event that must be addressed within 2 days
- A *service request* refers to an event of lower urgency that can be addressed within 10 days
- A *category* refers to a type of incident or service request, for example, a password reset, a security breach, or a request to setup and install a computer.
- A *priority* is assigned to every event using the numbers 1 - 5. A priority 1 is the most urgent.
- *Duration time* refers to the number of days between event creation and event closure. It is measured in calendar days, excluding weekends, giving an approximation of the number of business days it took to close an event.
- An *SLA* is a service level agreement. It is the stated commitment to the organization as to when the event will be addressed. For incidents, this commitment is currently 2 days; for service requests, it is 10 days.
- An *owner* is a member of the service desk staff who addresses a given event.

Descriptive Statistics

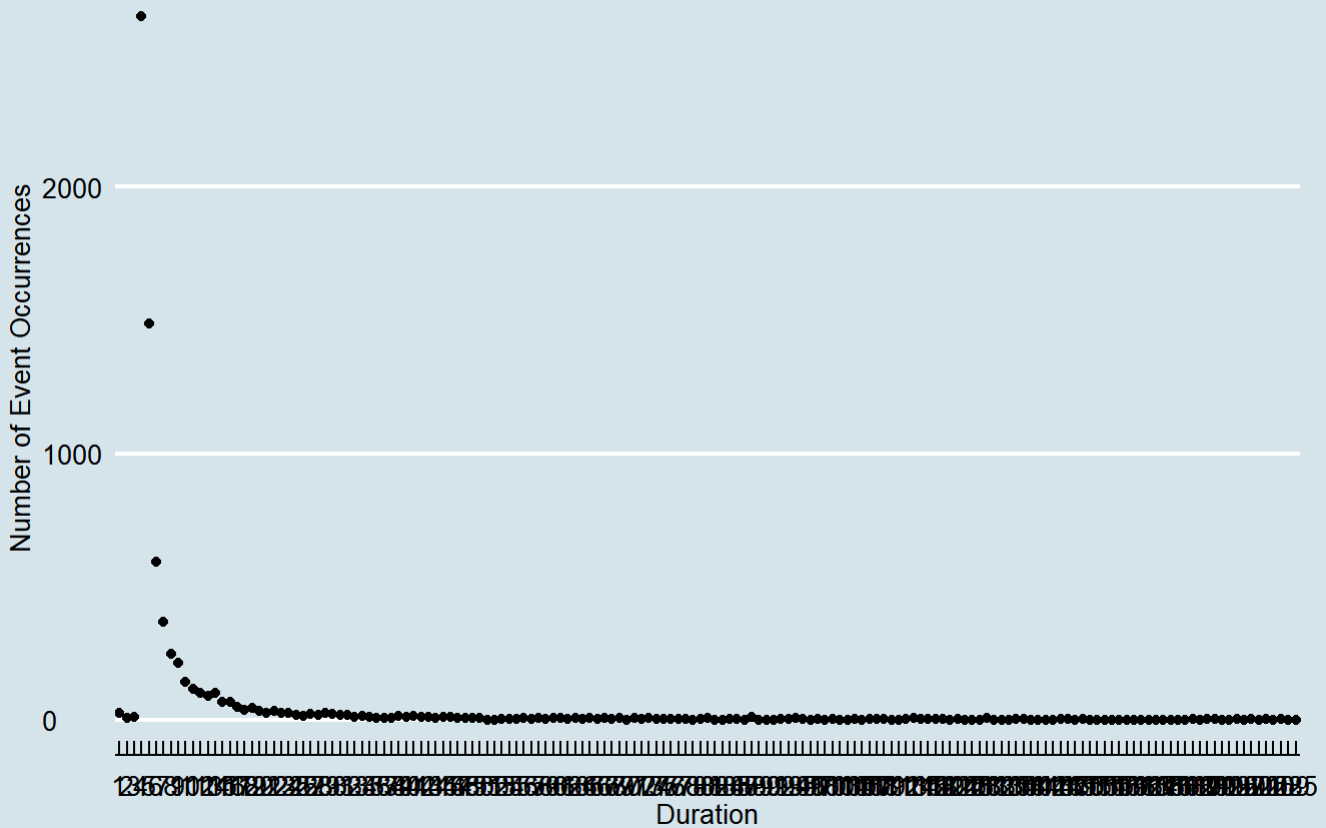
We will start the analysis by looking at the general shape and characteristics of the data, determine how many levels down we need to go to get the necessary granularity, and then look at the descriptive statistics for those groupings.

General Characteristics of the Data

The original data set had 7125 events overall, split between 1522 incidents (21%) and 5603 service requests (79%).

When we plot the curve for the entire data set, it does not suggest a normal distribution.

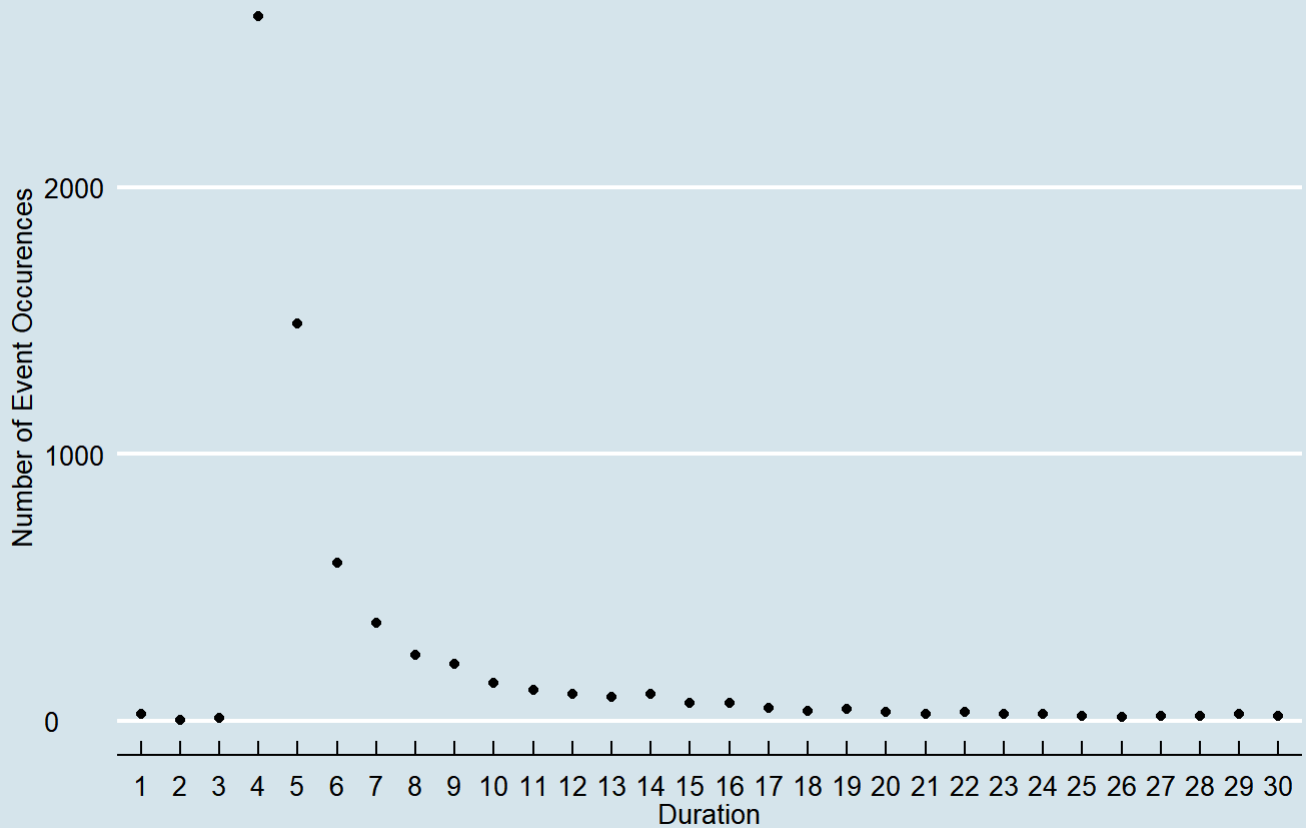
All Events - Duration Time and Frequency



Data points appear to be bunched to the left (duration times in the range of 0-30 days or so), and there are many outliers, some of which approach a full year in duration. A discussion with the business unit about this revealed that much of this may be due to inconsistent recordkeeping; while some staff enter a closure date as soon as an event has been addressed, others may batch their recordkeeping to occur every few weeks or longer, thereby adding unusually high (and consequently incorrect) durations to the mix.

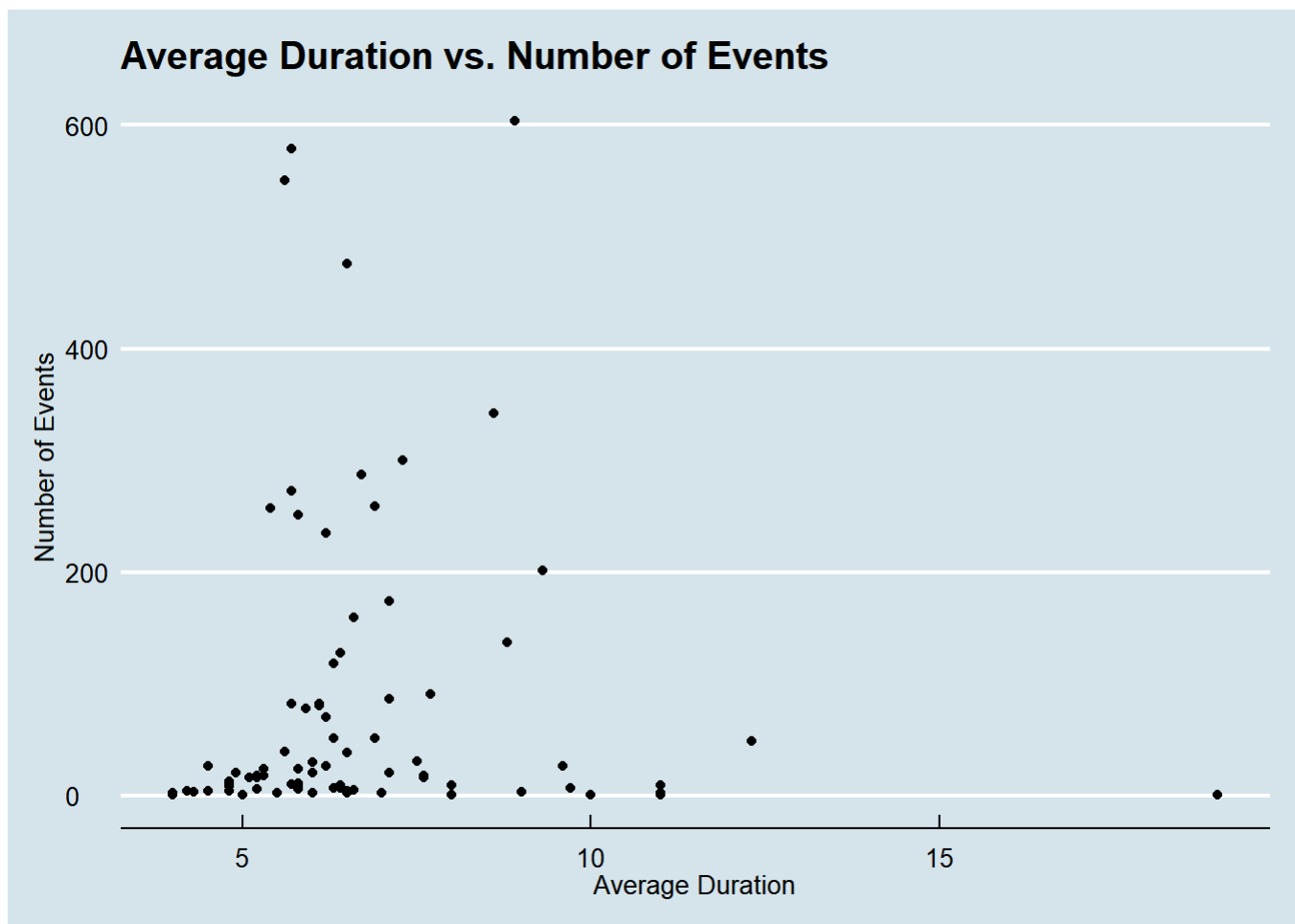
This insight suggests we may get a more accurate / normalized view by cutting off the duration times at some lower value, thereby removing the real outliers - the effect of bad recordkeeping. Looking at the charts below, it appears that a more reasonable point is about 30 days. We will use that value.

Events Duration Time and Frequency - 30 Day Cutoff



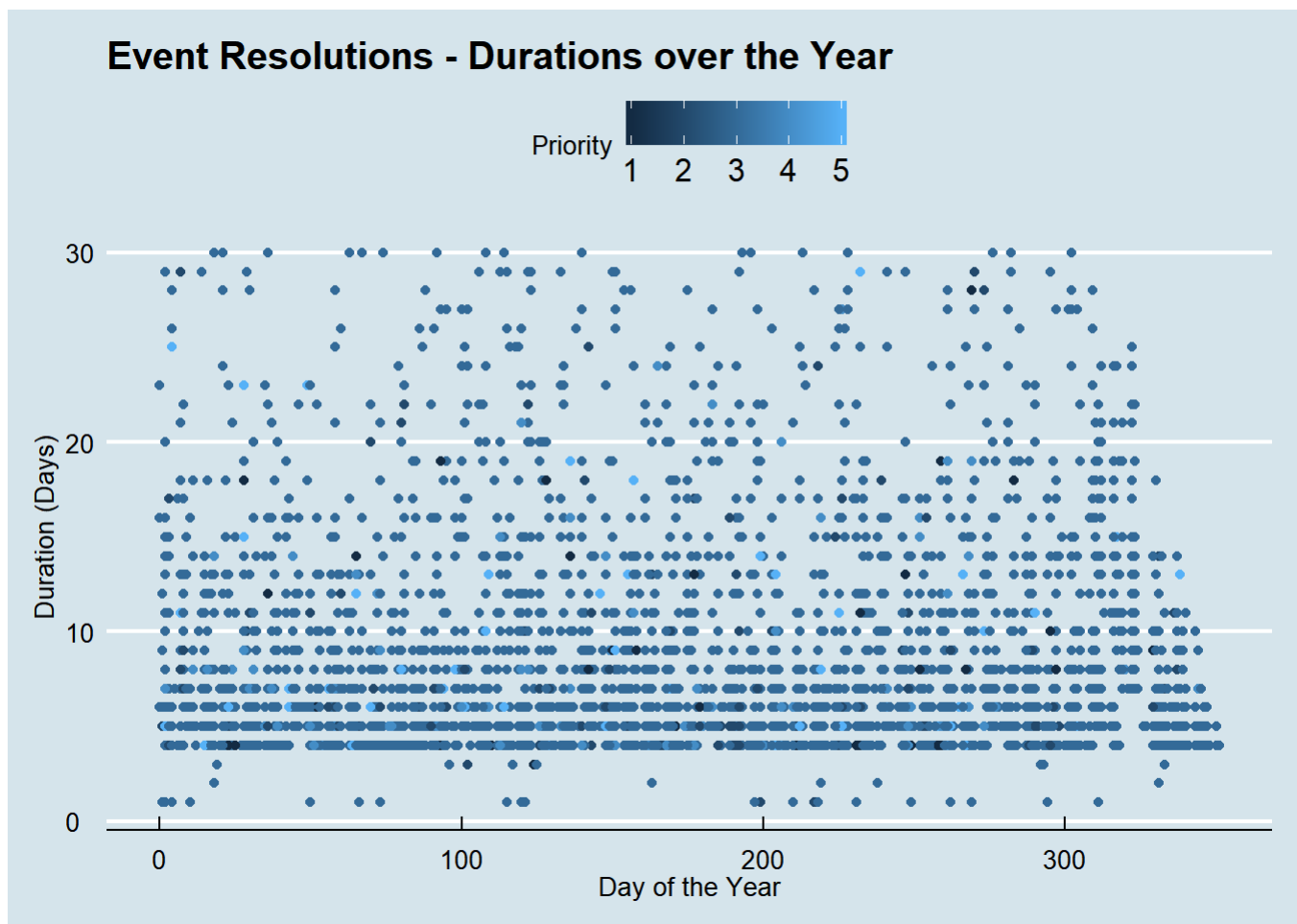
The “revised”30-day maximum” view shows high numbers of events clustered in the 0 to about 16 day range, with a few events occurring after that. The number of events starts to approach zero (1 or 2 occurrences for a given duration) beyond 30 days. For this analysis, we are going to assume that the data is “normal enough” and can provide us with enough confidence to look at basic statistics and then make predictions later on.

When plotting the number of events against average duration, we see a more pronounced clustering of events - this time in the 4-12 day range. There are still a few outliers but very few. This further suggests that the “real” group performance is in a shorter range of durations - likely less than 30 days and perhaps even around 20 days.



At this point we now know that most events - outliers aside - probably take a relatively few days to resolve.

The next view is to look at the distribution of events throughout the year relative to priority and duration. In this next plot, darker dots are high priority items, lighter dots are lower priority.



We see a couple of dynamics going on:

- There is a pervasive clustering between 3 and 10 days and then we see the data spread out a bit in the 10-30 day range. This clustering also shows where the mean and median are likely to be.
- The plot also suggests that there are *many* events that are not meeting the SLA goals of 2 days for incidents and 10 days for service requests. We also see many priority 1's (the darkest dots) well above those SLA's. In a perfect world, all dark dots would be in the 0-2 day range, and the others between 3 and 10.

The business unit has stated that incidents are more important to analyze and understand than service requests, simply because they have greater urgency. Therefore looking at all events together limits our insight; we will next look at them separately to see what additional information we can gather.

An Overview of Incidents

When we reduce the dataset to those events with a maximum duration of 30 days, we end up with about 1450 incidents, or about 22% of the total.

The average (mean) duration time for incidents is just over six days (which is three times the SLA) with a median of five days. With no limits on duration times, that number is between 11 and 12 days, giving us our first indication that inconsistent recordkeeping - if that indeed is the explanation for long duration times - has a very pronounced effect. This provides management with an obvious area for improvement - if the goal is to meet the 2-day SLA target.

Total incidents : 1451

Maximum duration : 30

Minimum duration : 1

Average duration : 6.1

Median duration : 5

Number above median: 469

Number at median : 345

Number below median: 637

Range of duration : 29

An Overview of Service Requests

We see that there is no significant difference between incident and service request duration statistics. The average is slightly higher but the mean is the same. The good news is that, *on average*, the service request SLA goal of 10 days is being met. On the other hand, we also learn that many service requests exceed the SLA target.

Total Service Requests : 5209

Maximum duration : 30

Minimum duration : 1

Average duration : 7

Median duration : 5

Number above median: 2020

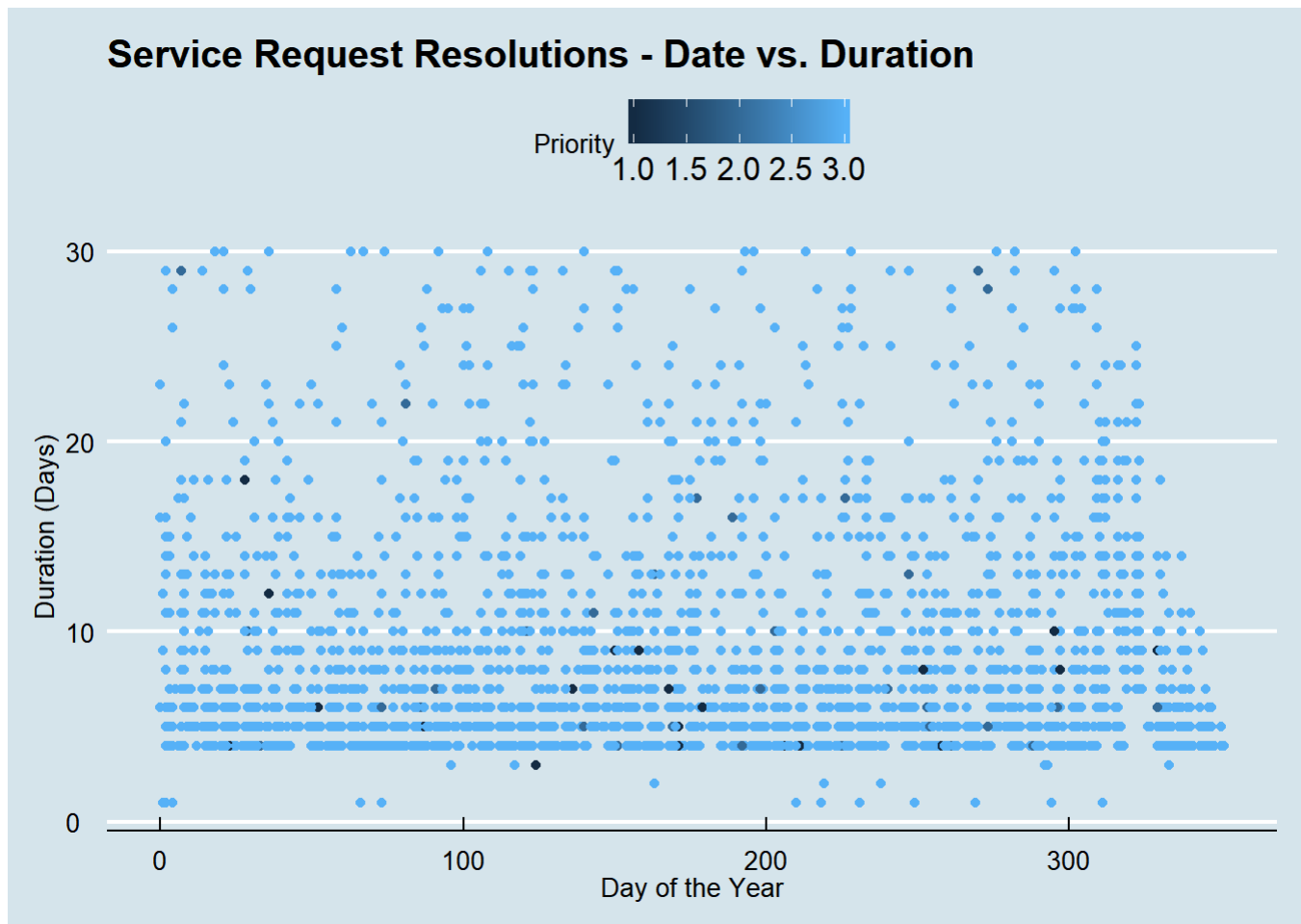
Number at median : 1143

Number below median: 2046

Range of duration : 29

Number above SLA target : 780

One big difference is that there are very few priority 1 and 2 service requests; the vast majority are priority 3's. On the one hand, we would expect service requests to have a lower proportion of 1's and 2's relative to incidents (and this is the case), but we also see many of the 1's and 2's that do exist (darker colored dots) higher up the scale in duration times. Those should be in the 0-2 day range.



Looking at Subcategories

Although these data points are interesting, splitting all events into incidents and service requests still gives us limited insight - there is not enough data to tell us just where the problems are, what to work on, or how to forecast likely durations for new problems. More granularity is needed. To aid in this, we took the step of constructing a 3-level hierarchy / taxonomy to classify every event into one of 89 categories and thereby to provide significantly more granularity. The original data contained two problem description fields, and this was used to make the various classifications. These classifications and their validations are available by inspecting the .csv source file.

The taxonomy has three primary categories (software, infrastructure and other). Note that these could be used for further / higher level analysis at some point. The next level has nine categories - four mapped into software, four more into infrastructure, and a single "other." The final level has 68 categories mapped into the four software categories, 18 into the infrastructure categories, and three mapped into the "other" category.

Analysis Categories / Taxonomy

##	L1N	L1C	L2N	L2C	L3N	L3C
## 1	Software	1	Business Apps	1	Acrobat Pro	1
## 2		1		1	Adobe Air	2
## 3		1		1	Adobe Analytics	3
## 4		1		1	Adobe CC	4
## 5		1		1	Adobe Flash	5
## 6		1		1	Adobe Illustrator	6
## 7		1		1	Adobe Reader	7
## 8		1		1	BetterCloud	8
## 9		1		1	Carbonite	9
## 10		1		1	Caspio	10
## 11		1		1	Ceridian	11
## 12		1		1	Cherwell	12
## 13		1		1	Concord	13
## 14		1		1	Creative Cloud	14
## 15		1		1	Cyberduck	15
## 16		1		1	Docusign	16
## 17		1		1	Dropbox	17
## 18		1		1	Enterprise	18
## 19		1		1	Go-Global	19
## 20		1		1	GoToMeeting	20
## 21		1		1	Greenshot	21
## 22		1		1	iContact	22
## 23		1		1	InCopy	23
## 24		1		1	InDesign	24
## 25		1		1	Intacct	25
## 26		1		1	JSHOL	26
## 27		1		1	K4	27
## 28		1		1	Kace	28
## 29		1		1	Keychain	29
## 30		1		1	Lastpass	30
## 31		1		1	LibreOffice	31
## 32		1		1	Livelink	32
## 33		1		1	Marketing Cloud	33
## 34		1		1	Merlin	34
## 35		1		1	M-Files	35
## 36		1		1	Multimedia	36
## 37		1		1	Omnifocus	37
## 38		1		1	Photoshop	38
## 39		1		1	Pulse Secure	39
## 40		1		1	Pycharm	40
## 41		1		1	Salesforce	41
## 42		1		1	SentinelOne	42
## 43		1		1	Skype	43
## 44		1		1	Slack	44
## 45		1		1	Trello	45
## 46		1		1	UltiPro	46
## 47		1		1	Visio	47
## 48		1		1	Zoom	48
## 49		1	Browsers	2	Chrome	49
## 50		1		2	Firefox	50
## 51		1		2	IE / Edge	51
## 52		1		2	Browsers - Other	52

## 53	1	Google Apps	3	Docs and Sheets	53
## 54	1		3	Calendar	54
## 55	1		3	Mail	55
## 56	1		3	Groups	56
## 57	1		3	Folders	57
## 58	1		3	Sites	58
## 59	1		3	Chats or Google+	59
## 60	1		3	Google Apps - Other	60
## 61	1	MS-Office	4	MS Excel	61
## 62	1		4	MS Access	62
## 63	1		4	MS Outlook	63
## 64	1		4	MS Publisher	64
## 65	1		4	MS Windows	65
## 66	1		4	MS Word	66
## 67	1		4	MS Powerpoint	67
## 68	1		4	MS Office Other	68
## 69 Infrastructure	2	User Platforms	5	Monitors	69
## 70	2		5	Computer Accessories	70
## 71	2		5	Setup and installs	71
## 72	2		5	Updates / Upgrades	72
## 73	2		5	Move / Close / Remove	73
## 74	2		5	Functional Issues	74
## 75	2		5	Infrastructure Placeholder	75
## 76	2	Printing & Faxing	6	Printing / Scanning	76
## 77	2		6	Faxes	77
## 78	2	Telecomms	7	Network / Connectivity	78
## 79	2		7	Phones	79
## 80	2		7	Teleconferencing	80
## 81	2	Security & Access	8	Access or Login	81
## 82	2		8	Password	82
## 83	2		8	File Access	83
## 84	2		8	Breaches and Phishing	84
## 85	2		8	Security - Other	85
## 86	2		8	Security Placeholder	86
## 87	Other	GA and Other	9	Outage or error	87
## 88	3		9	Training	88
## 89	3		9	GA and Other	89

Classifying each of the events into these categories took significant time - about 40 hours for the full year of data (including time to debug errors in consistency and validation calculations). However, the effort was valuable, because it made it possible to determine what event types were causing the majority of the problems and how long, on average, it was taking to address them. It also laid the groundwork for a predictive model, because we had a dataset analogous to the movielens dataset: instead of movies, users, and ratings, we had categories, owners, and durations.

When we look at ALL events taken together, we can identify those that are causing most of the problems. Of the 89 possible event categories, relatively few are causing 80% of the problems - in fact, it is an almost perfect “80 / 20 rule”:

A total of 18 combinations account for 80% of all tickets submitted

L1	L2	L3	Events	MaxDur	MinDur	AvgDur	MMSpread	L3Name
----	----	----	--------	--------	--------	--------	----------	--------

	L1	L2	L3	Events	MaxDur	MinDur	AvgDur	MMSpread	L3Name
71	2	5	71	604	30	1	8.9	29	Setup and installs
80	2	8	81	579	30	1	5.7	29	Access or Login
55	1	3	55	551	27	3	5.6	24	Mail
87	3	9	89	476	30	1	6.5	29	GA and Other
73	2	5	73	342	30	4	8.6	26	Move / Close / Remove
72	2	5	72	300	30	4	7.3	26	Updates / Upgrades
78	2	7	79	287	29	1	6.7	28	Phones
60	1	3	60	273	30	1	5.7	29	Google Apps - Other
74	2	5	74	259	29	4	6.9	25	Functional Issues
81	2	8	82	257	29	2	5.4	27	Password
47	1	1	48	251	30	1	5.8	29	Zoom
75	2	6	76	235	25	1	6.2	24	Printing / Scanning
40	1	1	41	201	30	1	9.3	29	Salesforce
18	1	1	18	174	29	4	7.1	25	Enterprise
70	2	5	70	159	30	4	6.6	26	Computer Accessories
35	1	1	35	137	30	4	8.8	26	M-Files
85	3	9	87	128	20	1	6.4	19	Outage or error
77	2	7	78	118	28	3	6.3	25	Network / Connectivity

Just to be clear, this view includes **all** events (incidents and service requests combined) so it is still not completely useful. Nonetheless, the additional granularity provides a few insights:

- The top problem-causers are a mix of software, infrastructure, and “other”
- Many of the top problems appear to be less technical (logins, moves, mail)
- Being lower tech, the average durations *might* be relatively easy to reduce

As an explanatory note, “GA and Other” is a general / administrative category. Tickets in this category are often vague, having problem descriptions such as:

- Is Sally Schwartz’s PC PD00396 ok after water leak on P4?
- Hardware Support
- Alarm in Tel/Data Room
- Paco Has Connection Issues
- Laptop Docking Station Not Working, Need Power Cable For Nearby Monitor

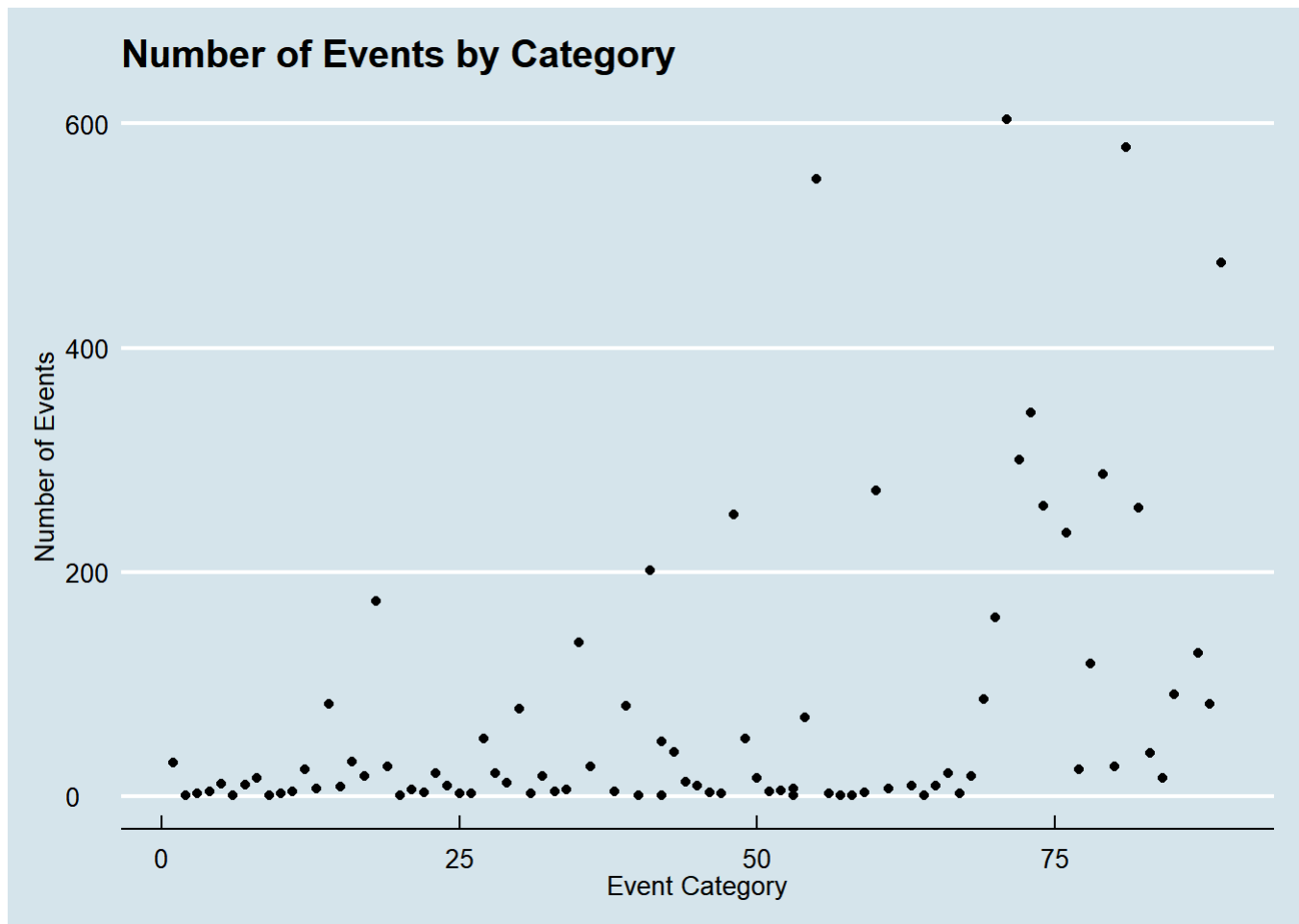
...and are therefore not easily categorized otherwise.

One other note on the taxonomy: the organization relies on many applications to run the business and these were called out in greater detail than might otherwise have been necessary (see taxonomy numbers 1-48). However, it was considered useful information, as it would enable analysis on those applications that were causing most of the

problems and presumably more of the staff's time - making them more expensive to support.

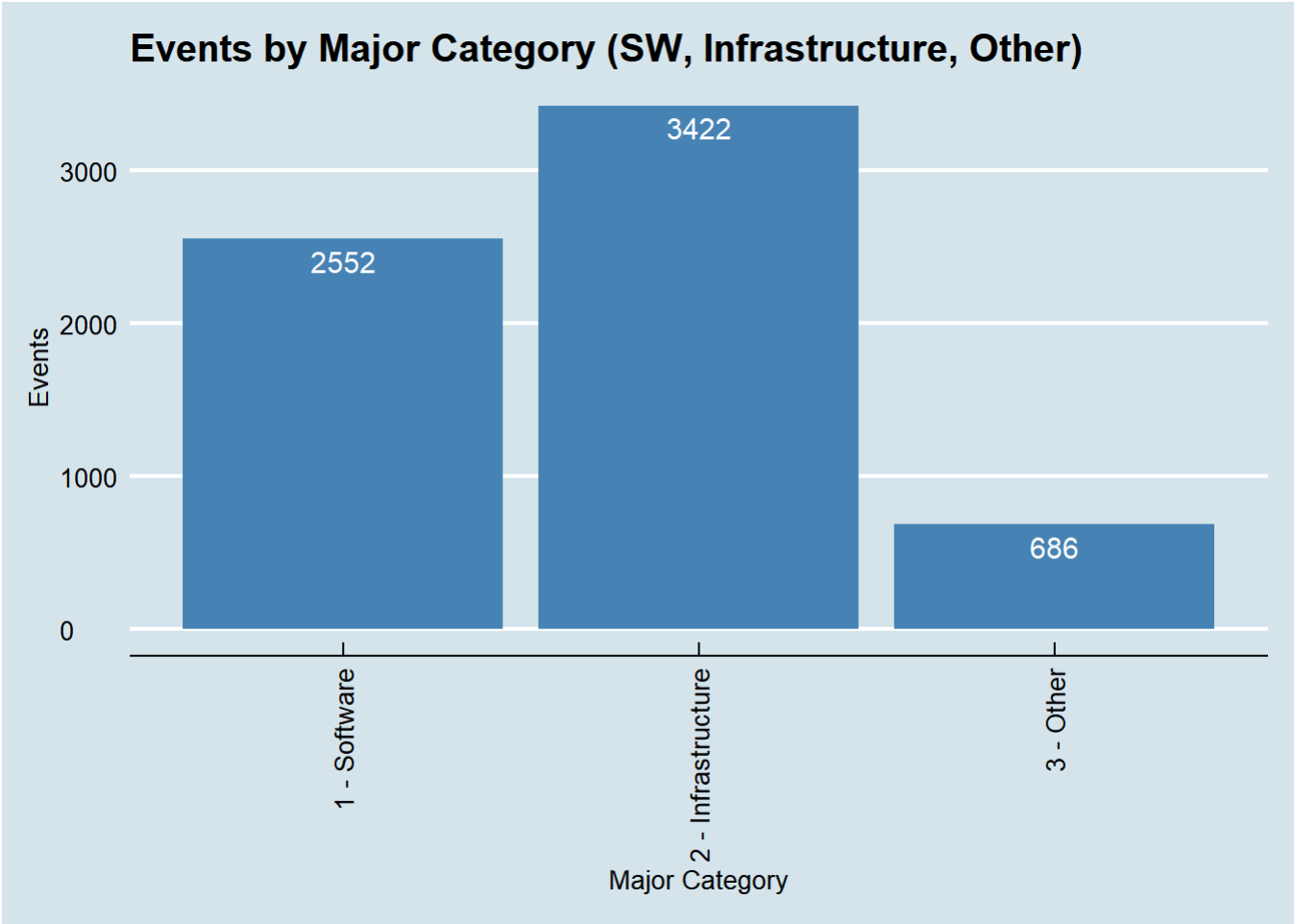
Because there are more categories for business applications (68), we see a few dynamics when we plot these categories by number of events for each:

- There are fewer events per category (left-hand portion of the plot). There are a total of 2552 software events - 38% of all events - and approximately 53 tickets per event on average. .
- Because there are fewer categories under infrastructure and "other" there tend to be more events under each (right-hand side of the plot). There are 3423 infrastructure events (51% of all events, with an average of 190 events per category) and 686 "other" events (10% of all events, with an average of 229 events per category).



As noted above, this is useful; when looking at the portfolio of business applications, we can see that Google Apps and Zoom head the list in number of tickets. Presumably, this implies higher staff hours and support costs.

we can also look at this distribution at a higher level - and perhaps simpler - with this bar chart:



Even these event subcategories - interesting and useful as they are - are not getting us to the bottom of either incidents or service requests, so we will look at those two major groups separately, and look at the 89 category types within each group.

Incidents

When we look at incidents broken out by category, we see some interesting data points:

- Functional issues are not only the largest incident category, but they also tend to take longer to resolve than most other categories.
- Access and login issues have very high average durations, posing the question, is it really taking that long to provide access, and if so, why? And further, are those problems interrupting the business flow?
- Zoom seems problematic and deserves further discussion by the service desk team to understand where these problems are coming from. Are they with the Zoom rooms primarily (hardware and infrastructure) or with the users themselves (lack of experience or training)?
- None of these top problem areas have acceptable SLA times - even with a 30-day cutoff for the dataset; average times are considerably over the SLA target. This chart may be another starting point for management discussion with the staff.

A total of 17 combinations out of 64 account for 80% of the incidents

L1	L2	L3	Events	MaxDur	MinDur	AvgDur	MMSpread	L3Name
----	----	----	--------	--------	--------	--------	----------	--------

	L1	L2	L3	Events	MaxDur	MinDur	AvgDur	MMSpread	L3Name
51	2	5	74	192	29	4	7.0	25	Functional Issues
57	2	8	81	151	19	1	5.3	18	Access or Login
64	3	9	89	109	25	1	6.1	24	GA and Other
30	1	1	48	102	23	1	5.7	22	Zoom
52	2	6	76	101	25	1	5.8	24	Printing / Scanning
62	3	9	87	86	20	1	6.3	19	Outage or error
55	2	7	79	81	29	4	7.5	25	Phones
48	2	5	71	61	19	4	5.9	15	Setup and installs
58	2	8	82	53	21	2	5.4	19	Password
54	2	7	78	52	28	3	6.3	25	Network / Connectivity
37	1	3	55	49	16	4	5.6	12	Mail
49	2	5	72	27	25	4	7.4	21	Updates / Upgrades
31	1	2	49	25	30	4	7.2	26	Chrome
21	1	1	30	23	9	4	5.0	5	Lastpass
12	1	1	18	22	6	4	4.4	2	Enterprise
46	2	5	69	21	26	1	8.0	25	Monitors
47	2	5	70	21	12	4	5.5	8	Computer Accessories

We also broke out the top five incident categories to see if there was anything to be learned by looking at the distribution of priorities, and if there was any correlation between events, priorities and resolution times. For example, if higher priorities had lower resolution times, that would be a reasonable behavior and vice-versa. Further, correlations might tell us something about what happens to durations when the number of incidents increases.

Top Incident Number 1 - Functional Issues

In looking at the priority breakouts, we would expect duration times to lengthen with priority number, but never beyond two days. We see that this is not really the case:

Priority	Events	MaxDur	MinDur	AvgDur	MMSpread
1	11	18	4	7.2	14
2	30	24	4	7.8	20
3	82	21	4	6.2	17
4	34	22	4	7.5	18
5	35	29	4	7.6	25

A breakout like this provides some useful observations for team discussion and goal setting. Points that might be worthwhile considering:

- Getting duration times down. This seems like a clear opportunity - especially priority 1s and 2s. This is not a statistical issue but rather a management one, that is, by:
 - Ensuring that events get the right priority classification
 - Keeping the top priority items visible to the team on a daily basis
 - Tracking their closure via an aging report
 - Rewarding staff for addressing items in priority order and reducing resolution times
- Looking at the number of events are bunched around Priority 3. With the current distribution we might ask if there *really* are that many, and if so, do we have a very practical prioritization scheme?
- Rethinking incident prioritization. Perhaps something simpler, such as “**fix on same day**” and “**fix by end of day 2**” would work better.

Correlations

A correlation is when one variable moves with another. Sometimes this is positive (when temperatures go up, the faster air molecules move) or negative (the more rabbits in a garden, the fewer the lettuces we have). Correlations range from -1 (strong negative) to +1 (strong positive). A correlation around 0 indicates randomness / no correlation. Correlations start to become interesting past the 0.5 mark, and especially around the 0.8 or 0.9 mark.

We look at these to see what might be driving what. In this case:

- *Maximum duration* is moving with priority (and this would be expected). That is, with a higher priority *number* (which means a lower priority), we see maximum duration times going up at a 0.77 correlation, which is moderately strong. This may imply that staff are (quite rightly) leaving lower priority items to when they can get to them.
- Events seem to drive *average duration times* negatively at a relatively strong correlation of -0.77 - that is, as the number of incidents goes up, the average time to close on them seems to go down. While this is counter-intuitive, there could be reasons for this such as staff jumping on a problem set more intensely during a crisis period. However, it deserves discussion with the service desk staff - is this *really* going on, and why.

	Pr iority	Events	MaxDur Min	Dur	AvgDur
Priority	1.0000000	0.3133895	0.7738232	NA	0.1253137
Events	0.3133895	1.0000000	0.0592279	NA	-0.7751687
MaxDur	0.7738232	0.0592279	1.0000000	NA	0.4615802
MinDur	NA	NA	NA	1	NA
AvgDur	0.1253137	-0.7751687	0.4615802	NA	1.0000000

Top Incident Number 2 - Access and Login

In this 2nd category, we notice that:

- Most are priority 3 to 5; there are very few 1's. This is a little surprising as usually needing to login to a system or desktop is an urgent issue.
- Average durations are high - even priority 1's are 2x the SLA target.

Given that access and login issues are often resolvable quickly, this seems like an opportunity to dramatically lower average resolution times. And if the incident involves some kind of approval chain, then reclassify it as a service request.

Priority	Events	MaxDur	MinDur	AvgDur	MMSpread
1	1	4	4	4.0	0
2	15	8	4	4.9	4
3	69	19	1	5.6	18
4	46	14	4	5.0	10
5	20	14	4	5.5	10

Correlations

When we look at correlations, we see:

- When the priority goes up (a higher number means a lower priority), both average and maximum durations increase (0.77 and 0.70 respectively). This makes sense (again, lower priority items should take longer).
- Similarly, when the number of events goes up, both the average and maximum duration times go up with correlations of 0.71 and 0.91 (moderately to very strong). Again, seems right.
- Oddly, the more events, the *shorter* the minimum duration time with a strong correlation of 0.79 - this needs discussion (why is this happening?).
- Average duration and maximum duration are traveling very close together (0.9 correlation) suggesting that when events pile up, time suffers. Seems right.

	Priority	Events	MaxDur	MinDur	AvgDur
Priority	1.0000000	0.4022247	0.7029595	0.0000000	0.7702012
Events	0.4022247	1.0000000	0.9097099	-0.7996618	0.7082246
MaxDur	0.7029595	0.9097099	1.0000000	-0.6882472	0.9135627
MinDur	0.0000000	-0.7996618	-0.6882472	1.0000000	-0.5270463
AvgDur	0.7702012	0.7082246	0.9135627	-0.5270463	1.0000000

Top Incident Number 3 - GA (General Administration) and Other

As before, priorities are bunched at priority 3, but even more so. Priority 2's have an unusually long average duration relative to other priorities. This suggests that:

- Prioritization is difficult and therefore the default is generally 3
- Once prioritized, it is difficult to believe the prioritization, and therefore problems may be being addressed more arbitrarily, (with 2's suffering the most)
- There may be some better way to communicate the urgency of items which are in a category that admittedly does not lend itself to a sense of urgency. Alternatively, more precise classification of the kind of problem may give it more visibility and therefore a shorter duration time.

Priority	Events	MaxDur	MinDur	AvgDur	MMSpread
1	1	4	4	4.0	0
2	5	16	4	10.0	12
3	78	25	1	6.0	24
4	13	19	4	6.1	15
5	12	8	4	5.1	4

Correlations

The correlations here are odd, but perhaps explainable:

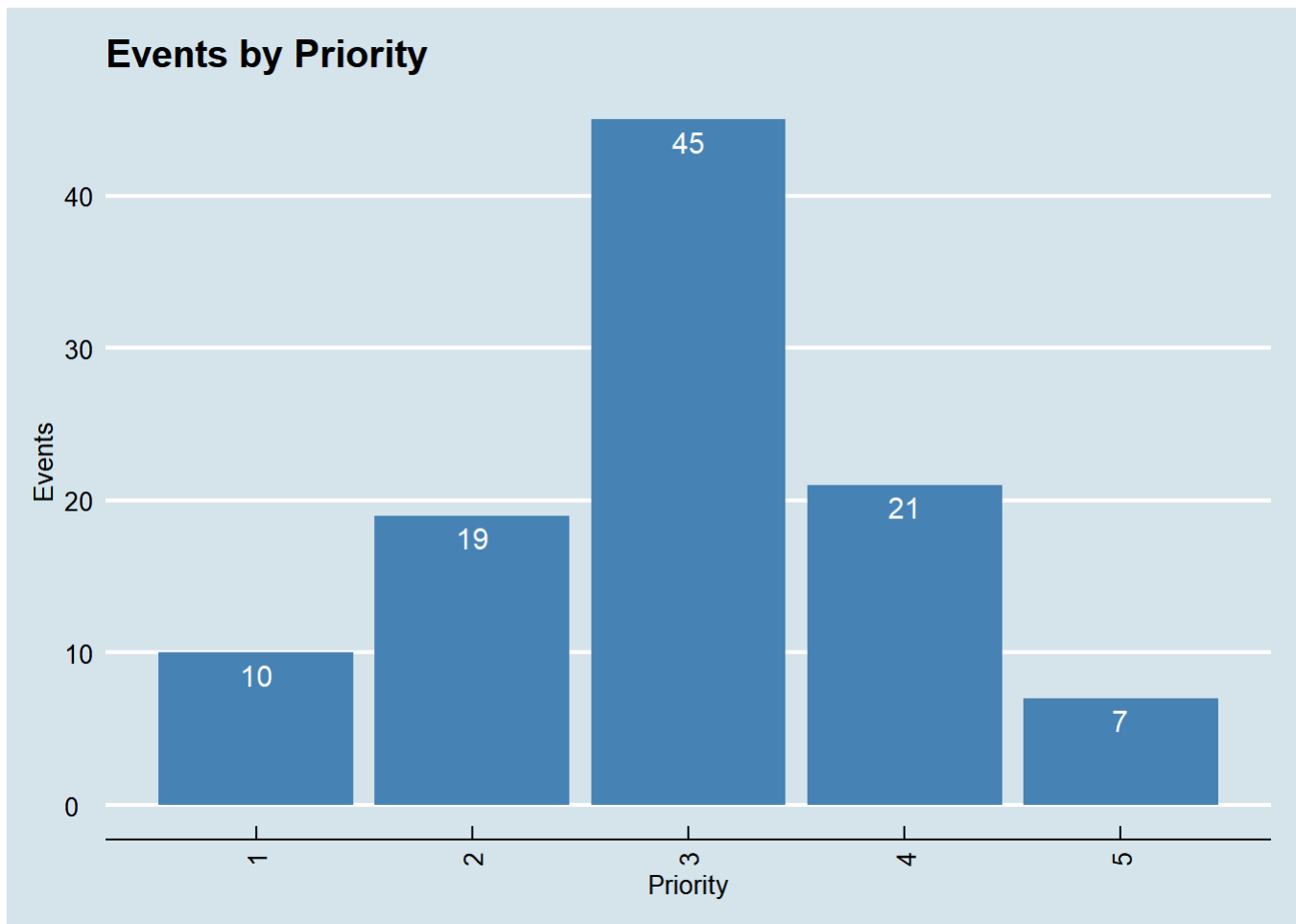
- There seems to be no correlation between priority and time - in that incidents are resolved without regard to their original priority
- There is a strong correlation between both maximum and minimum durations *at the same time*. This is very strange, suggesting that as work piles up, tickets either get addressed very quickly or very slowly, with fewer in the middle / average time.

	P riority	Events	MaxDur	MinDur	AvgDur
Priority	1.0000000	0.1491301	0.2059766	0.0000000	-0.1186406
Events	0.1491301	1.0000000	0.7515504	-0.9877236	-0.0612660
MaxDur	0.2059766	0.7515504	1.0000000	-0.7017560	0.4367330
MinDur	0.0000000	-0.9877236	-0.7017560	1.0000000	0.0592176
AvgDur	-0.1186406	-0.0612660	0.4367330	0.0592176	1.0000000

Top Incident Number 4 - Zoom

It is not surprising that Zoom made the top 10 list. It is the standard for teleconferencing for the organization and most conference rooms are set up to be Zoom rooms. Therefore there are multiple opportunities for things to go wrong: first with the user; second, with the physical equipment in the room; and third, with the connectivity within the infrastructure.

Priority	Events	MaxDur	MinDur	AvgDur	MMSpread
1	10	19	4	6.3	15
2	19	9	4	5.1	5
3	45	23	1	6.3	22
4	21	10	4	5.0	6
5	7	7	4	4.6	3



The “curve” of the data here has more of the look of a normal distribution - most values are in the middle (priority 3) with roughly equal numbers of events in priority 2 and 4, and again with priorities 1 and 5. This is a little more like we would expect as the norm.

Average durations are still not meeting SLAs overall and are particularly problematic for Priority 1’s (highest average duration, second highest maximum duration).

Correlations

As we have seen in a couple of other instances, in this case, priority is driving average duration downward, which is the reverse of what we would expect; because a higher priority number means a lower priority / urgency, we would expect average duration to go up, not down.

In the case of events, we are seeing that the number of incidents *increase* the maximum duration while simultaneously *decreasing* the minimum duration. As seen earlier, is this because as the work piles up, things tend to get done either very fast or very slow?

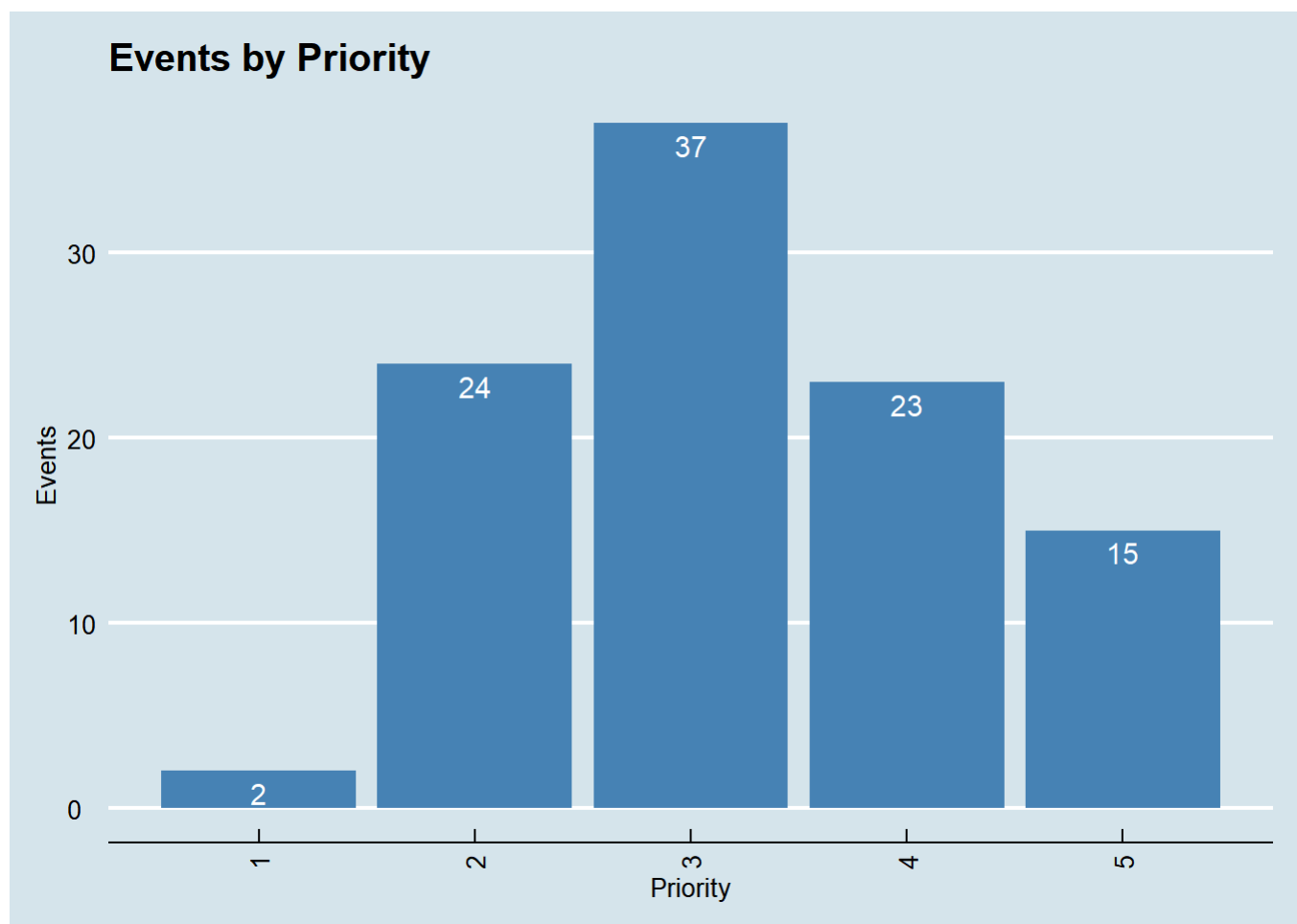
	Priority	Events	MaxDur	MinDur	AvgDur
Priority	1.0000000	-0.0422766	-0.5205805	0.0000000	-0.7011227
Events	-0.0422766	1.0000000	0.6406350	-0.9192425	0.5013576
MaxDur	-0.5205805	0.6406350	1.0000000	-0.7522167	0.9711930
MinDur	0.0000000	-0.9192425	-0.7522167	1.0000000	-0.5949223
AvgDur	-0.7011227	0.5013576	0.9711930	-0.5949223	1.0000000

Top Incident Number 5 - Printing / Scanning

As in the case of Zoom, the distribution is more normal-like but to a lesser degree; while priorities 2, 3, and 4 look about right, there is an imbalance between priorities 1 and 5.

As we have seen before, the average durations for the various priorities seem more random than purposeful, and the average duration for priority 1 seems particularly problematic. It suggests (as similar cases have suggested) that service desk staff need to assign priorities accurately, and then focus on the top priorities first. This would likely produce (over time) a normally distributed data shape with very low average durations (under 2 days) for higher priority tickets.

Priority	Events	MaxDur	MinDur	AvgDur	MMSpread
1	2	13	4	8.5	9
2	24	25	1	6.4	24
3	37	11	3	5.2	8
4	23	11	4	4.7	7
5	15	25	4	7.3	21



Correlations

The only strong correlation in this case is the number of events and average duration. It is (again) odd that the more tickets, the lower the average duration.

	Priority	Events	MaxDur	MinDur	AvgDur
Priority	1.0000000	0.3070780	0.2151657	0.3638034	-0.4195924
Events	0.3070780	1.0000000	-0.1480028	-0.4200523	-0.8525313
MaxDur	0.2151657	-0.1480028	1.0000000	-0.4696682	0.3567237
MinDur	0.3638034	-0.4200523	-0.4696682	1.0000000	0.1588544
AvgDur	-0.4195924	-0.8525313	0.3567237	0.1588544	1.0000000

Service Requests

We now turn our attention to **service requests**, following a path similar to the one we took for incidents. Looking through the data file, we find that there are 5209 service requests, and learn that:

```
## A total of 18 combinations out of 86 account for 80% of the service requests
```

	L1	L2	L3	Events	MaxDur	MinDur	AvgDur	MMSpread	L3Name
70	2	5	71	543	30	1	9.2	29	Setup and installs
55	1	3	55	502	27	3	5.6	24	Mail
79	2	8	81	428	30	1	5.9	29	Access or Login
86	3	9	89	367	30	1	6.6	29	GA and Other
72	2	5	73	323	30	4	8.7	26	Move / Close / Remove
71	2	5	72	273	30	4	7.3	26	Updates / Upgrades
60	1	3	60	269	30	1	5.8	29	Google Apps - Other
77	2	7	79	206	29	1	6.4	28	Phones
80	2	8	82	204	29	4	5.4	25	Password
40	1	1	41	188	30	1	9.5	29	Salesforce
18	1	1	18	152	29	4	7.5	25	Enterprise
47	1	1	48	149	30	4	5.9	26	Zoom
69	2	5	70	138	30	4	6.8	26	Computer Accessories
74	2	6	76	134	24	4	6.5	20	Printing / Scanning
35	1	1	35	126	30	4	9.0	26	M-Files
14	1	1	14	75	23	4	5.8	19	Creative Cloud
83	2	8	85	70	30	4	7.4	26	Security - Other
73	2	5	74	67	24	4	6.6	20	Functional Issues

When we look at the top five service requests, we find that they:

- Have 300 or more tickets (each)
- Constitute 42% of *all* requests
- Have an average duration ranging from 5.6 to 9.2 days (under the SLA)
- Do not appear to be overly complex problems to solve

Taken together, this provides a very good opportunity to reduce duration times and improve customer satisfaction. Service requests, because their SLA target is considerably more forgiving than incidents, have a much higher success rate - 85% of service requests are completed in 10 days or less.

Let's look a little closer at the top 5 generators of service request tickets.

Top Service Request Number 1 - Setup and Installs

Not surprisingly, with all of the coming and going of staff and interns, setup and install requests lead the service requests. Because these are rarely incidents (generally they are planned some time in advance), we expect that they will be comfortably within the SLA target; however, we see that, on average, it is pushing the limit at over 9 days. Another improvement opportunity.

We also see that essentially *ALL* of the requests (in fact all but 9) are priority 3, and that there are **NO** priority 4 or 5 service requests. It turns out that there are no priority 4 and 5 service requests in any of the data. This implies that service requests have, in effect, no real prioritization. This might be fine, but the sheer volume of them (over 5,000 annually) suggests that we need *some* mechanism to determine what to work on next. If something is truly critical / urgent, make it an incident. If it is truly a service request, they need to be grouped in some way that implies priority / order in which to address.

Priority Breakdown for service request category 1: Setup and installs

Priority	Events	MaxDur	MinDur	AvgDur	MMSpread
1	2	16	4	10.0	12
2	7	12	4	5.7	8
3	534	30	1	9.3	29
4	0	0	0	0.0	0
5	0	0	0	0.0	0

Correlations

There are two strong correlations here - priority is driving both average duration and minimum duration (making them shorter). Neither of these makes much sense, as it implies that the lower the priority (again, the higher the priority number, the lower the priority), the faster it gets done. This needs discussion - is it because there is effectively no priority here (that is, the 9 non-priority 3 requests are skewing the data)? Or is it because tickets are being taken at random from this massive pile of priority 3's and being closed in a variety of ways? This is worth management time to figure out.

	Priority	Events	MaxDur	MinDur	AvgDur
Priority	1.0000000	-0.0073132	-0.5555839	-0.9258201	-0.8383308
Events	-0.0073132	1.0000000	0.8258184	-0.2089693	0.5012070

	Priority	Events	MaxDur	MinDur	AvgDur
MaxDur	-0.5555839	0.8258184	1.0000000	0.3662943	0.8954448
MinDur	-0.9258201	-0.2089693	0.3662943	1.0000000	0.6820197
AvgDur	-0.8383308	0.5012070	0.8954448	0.6820197	1.0000000

Top Service Request Number 2 - Mail (Email)

The sheer magnitude of email usage makes it no surprise that this is high on the list. Once again, however, we see that only 7 of 502 requests are *not* priority 3. As a result, we have data questions similar to those above.

Additionally, however, it is disturbing to see the few priority 1's hav such long average duration times, *especially with* a 30-day cutoff. The good news is that priority 2's and 3's are well within the range of acceptable SLA times.

```
## Priority Breakdown for service request category 2: Mail
```

	Priority	Events	MaxDur	MinDur	AvgDur	MMSpread
	1	5	20	5	12.2	15
	2	2	7	5	6.0	2
	3	495	27	3	5.5	24
	4	0	0	0	0.0	0
	5	0	0	0	0.0	0

Correlations

We see again the strange behavior of minimum and average duration times going down as the priority number goes higher. In addition, we are seeing that, as the number of events (service requests) increases, so does the maximum duration. Taken together, this implies (as before) that the mass of priority 3's really means there is, practically speaking, no prioritization, and as events pile up, some of those 3's age for a long time.

	Priority	Events	MaxDur	MinDur	AvgDur
Priority	1.0000000	-0.0086010	-0.6094139	-0.9449112	-0.9483714
Events	-0.0086010	1.0000000	0.7488221	0.0969847	0.0930226
MaxDur	-0.6094139	0.7488221	1.0000000	0.6174987	0.7221957
MinDur	-0.9449112	0.0969847	0.6174987	1.0000000	0.9016293
AvgDur	-0.9483714	0.0930226	0.7221957	0.9016293	1.0000000

Top Service Request Number 3 - Access and Login

As in the case of the others, nearly all events are bunched around Priority 3, again suggesting that this is a general oddbins category for any request unless something in the definitively sets it apart. Average durations are respectable (under 6 days in each case), and all the priority 1's are within the SLA. This is what we want to see.

Priority Breakdown for service request category 3: Access or Login

Priority	Events	MaxDur	MinDur	AvgDur	MMSpread
1	7	10	3	5.3	7
2	11	16	4	5.5	12
3	410	30	1	5.9	29
4	0	0	0	0.0	0
5	0	0	0	0.0	0

Correlations

We see, as before, lower priorities (3's, presumably) driving minimum and average durations down and events driving maximum durations up. Again, the latter makes sense, and the former does not. To be fair however, the high percentage of events that are priority 3 make it virtually impossible to see any true correlations clearly.

	Priority	Events	MaxDur	MinDur	AvgDur
Priority	1.0000000	-0.0217900	-0.4539899	-0.8703883	-0.8328230
Events	-0.0217900	1.0000000	0.8521130	-0.1591607	0.4901236
MaxDur	-0.4539899	0.8521130	1.0000000	0.3775854	0.8542977
MinDur	-0.8703883	-0.1591607	0.3775854	1.0000000	0.7690025
AvgDur	-0.8328230	0.4901236	0.8542977	0.7690025	1.0000000

Top Service Request Number 4 - GA and Other

This is a difficult category to deal with, as it contains administrative tasks, notes to users, requests for not otherwise classifiable services, and miscellaneous email. Once again, nearly all requests are priority 3s, and between the variations within the category and the single prioritization, this does not tell us much.

Priority Breakdown for service request category 4: GA and Other

Priority	Events	MaxDur	MinDur	AvgDur	MMSpread
1	2	4	4	4.0	0
2	5	22	5	13.4	17
3	360	30	1	6.6	29
4	0	0	0	0.0	0
5	0	0	0	0.0	0

Correlations

We conce again have the odd coincidence of lower priority items driving the average and minimum durations down. And as before, we see increasing numbers of events driving the maximum duration time up. And as before, the latter makes sense but the former does not - and no doubt the same reasons apply: too many priority 3's to be able to distinguish what is driving what.

	Priority	Events	MaxDur	MinDur	AvgDur
Priority	1.0000000	-0.0088813	-0.3412618	-0.8764598	-0.6079153
Events	-0.0088813	1.0000000	0.7641768	-0.2268697	0.1932564
MaxDur	-0.3412618	0.7641768	1.0000000	0.3374487	0.7755450
MinDur	-0.8764598	-0.2268697	0.3374487	1.0000000	0.7967302
AvgDur	-0.6079153	0.1932564	0.7755450	0.7967302	1.0000000

Top Service Request Number 5 - Move / Close / Remove

Like setups and installs, these events are fairly common, and for the same reason: staff and interns come and go; old equipment is turned in and new equipment issued and so forth.

The bunching at priority 3 is extreme once again, and so once again we lose our ability to see clearly what is actually happening. This is compounded by the two priority 2's sitting out there with an average duration of 29.0 days - suggesting late recordkeeping.

Priority Breakdown for service request category 5: Move / Close / Remove

Priority	Events	MaxDur	MinDur	AvgDur	MMSpread
1	1	4	4	4.0	0
2	2	29	29	29.0	0
3	320	30	4	8.6	26
4	0	0	0	0.0	0
5	0	0	0	0.0	0

Correlations

For what value the correlations may provide, we see higher numbers of events increasing the maximum duration times (this is logical), and the three duration values (average, min, and max) traveling together; that is, they appear to go up together. In the absence of any better data, this seems normal.

	Priority	Events	MaxDur	MinDur	AvgDur
Priority	1.0000000	-0.0044297	-0.3770019	-0.4779867	-0.4838160
Events	-0.0044297	1.0000000	0.6310173	-0.1499599	0.0183386
MaxDur	-0.3770019	0.6310173	1.0000000	0.6723676	0.7872092
MinDur	-0.4779867	-0.1499599	0.6723676	1.0000000	0.9857757

	Priority	Events	MaxDur	MinDur	AvgDur
AvgDur	-0.4838160	0.0183386	0.7872092	0.9857757	1.0000000

Events and Owners

Just as no two event categories are the same - each has unique characteristics, levels of complexity, requirements to address, and so forth, so staff members have unique characteristics in terms of ability, experience, speed with which they can address a given problem, and so forth. This section looks at service desk staff and sees how they are performing relative to:

- The sheer numbers of incidents and service requests
- How long it takes them on average to address an event
- The standard deviation of those durations (speed to address all)

The standard deviation is included because it is a proxy for how long it takes a given person to get a given set of tickets closed. In this context, it describes the spread in days to address the first 68% of the events handled. It is useful to compare average duration and standard deviation when considering a given person's performance. We want *both* to be low - not only the average time it takes to address an issue, but also, the number of days it takes to address a given set of issues (that is, a sample).

Using the 68-95-99 rule of standard deviations, a normally distributed data set will have 68% of the data points within the first standard deviation, 95% within two deviations, and 99% within three. In our context, we are talking days of duration. For example, if a staff member typically addresses a given set of service requests in 3.5 days, with a standard deviation of 2 days, this person would likely complete 95% of his or her service requests within 2 standard deviations (4 days), and 99% of them in 6 days. We would like this if we were talking service requests, as all the work would be completed within the SLA target. We would feel less good about it if we were talking incidents.

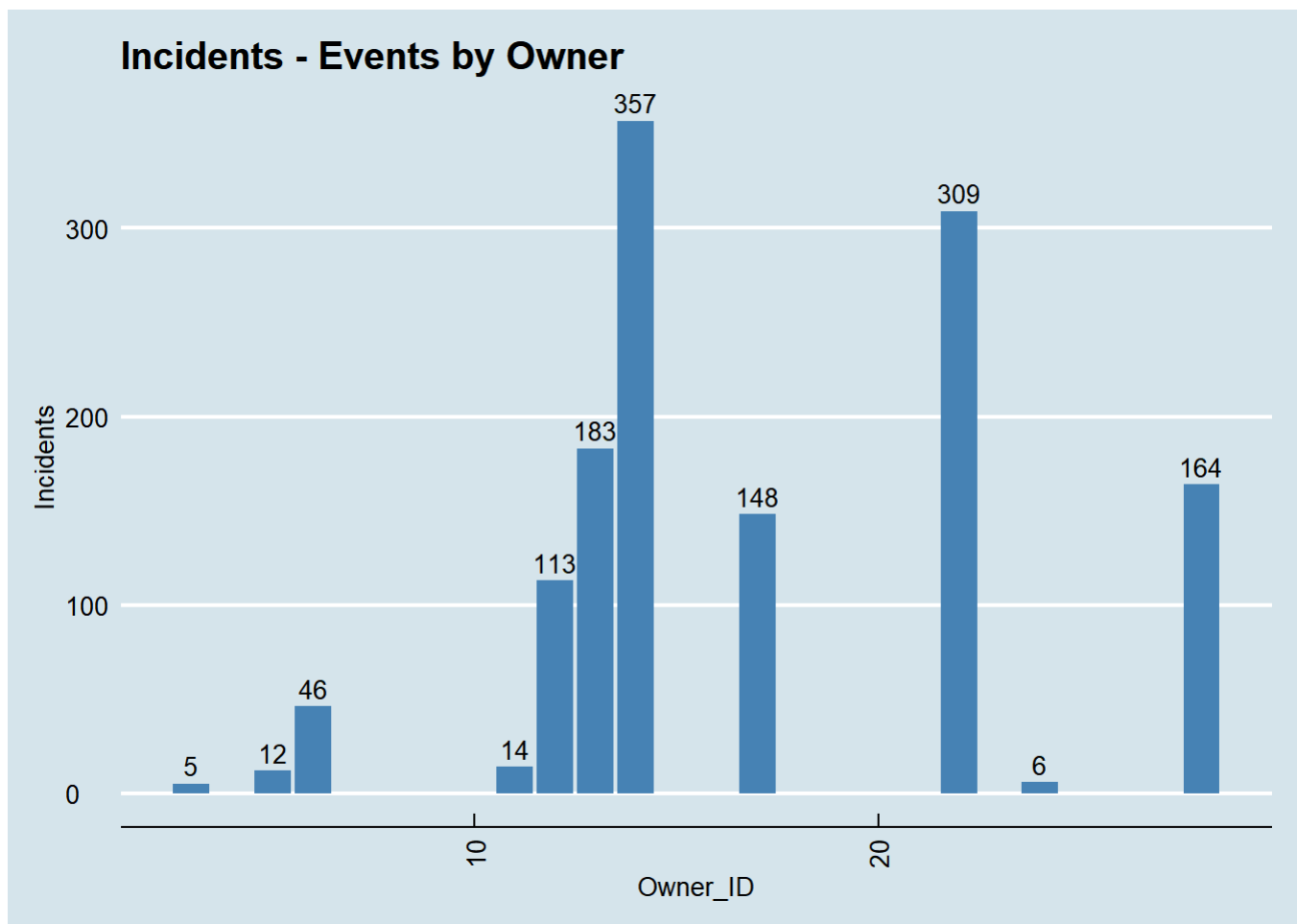
In like manner, a poorer performer might have the same average duration of 3.5 days, but have a standard deviation of 5 days, meaning that it takes 15 days to close a given set of tickets. This is well outside of all service level targets.

Incidents

We look first at how well the staff was able to handle the ~1450 incidents over the year. In the plots below:

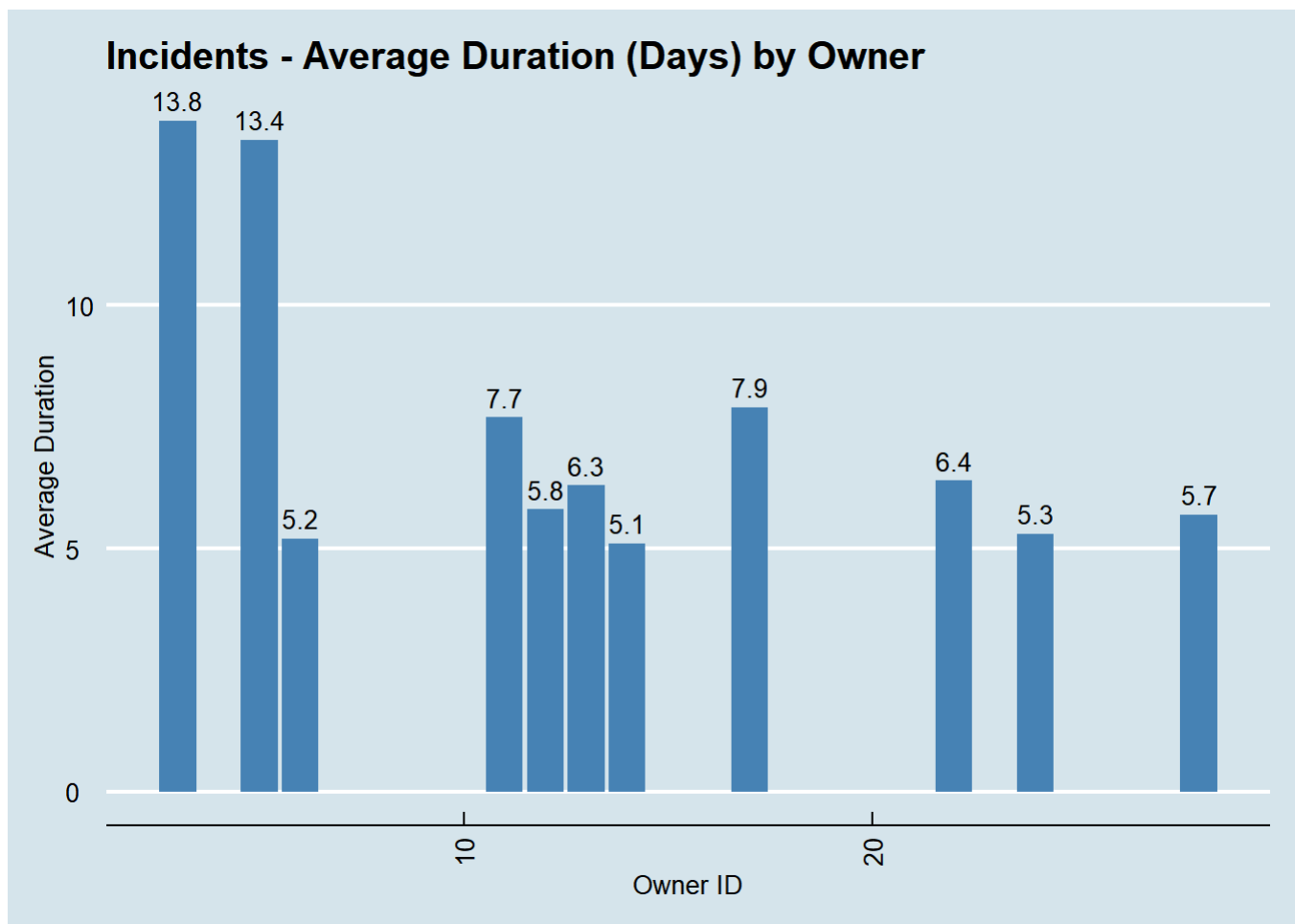
- The first compares owners to the number of incidents handled. More incidents handled is, all things being equal, better.
- The second looks at the average duration time to handle an incident. In this case, we are looking for the lowest possible number
- The last shows the standard deviation. As in the case of average duration, a lower value is an indicator of better performance (fewer days needed to complete a body of work).

We see a tiering effect here - two staffers are handling very high numbers of incidents, a second tier of four handling over 100 incidents, and a final tier of five where very few incidents are handled. To be fair, this is not enough information to make a judgment, as we do not know, for example, if some of these lower numbers are because the given staff member was actually fully engaged with service requests.

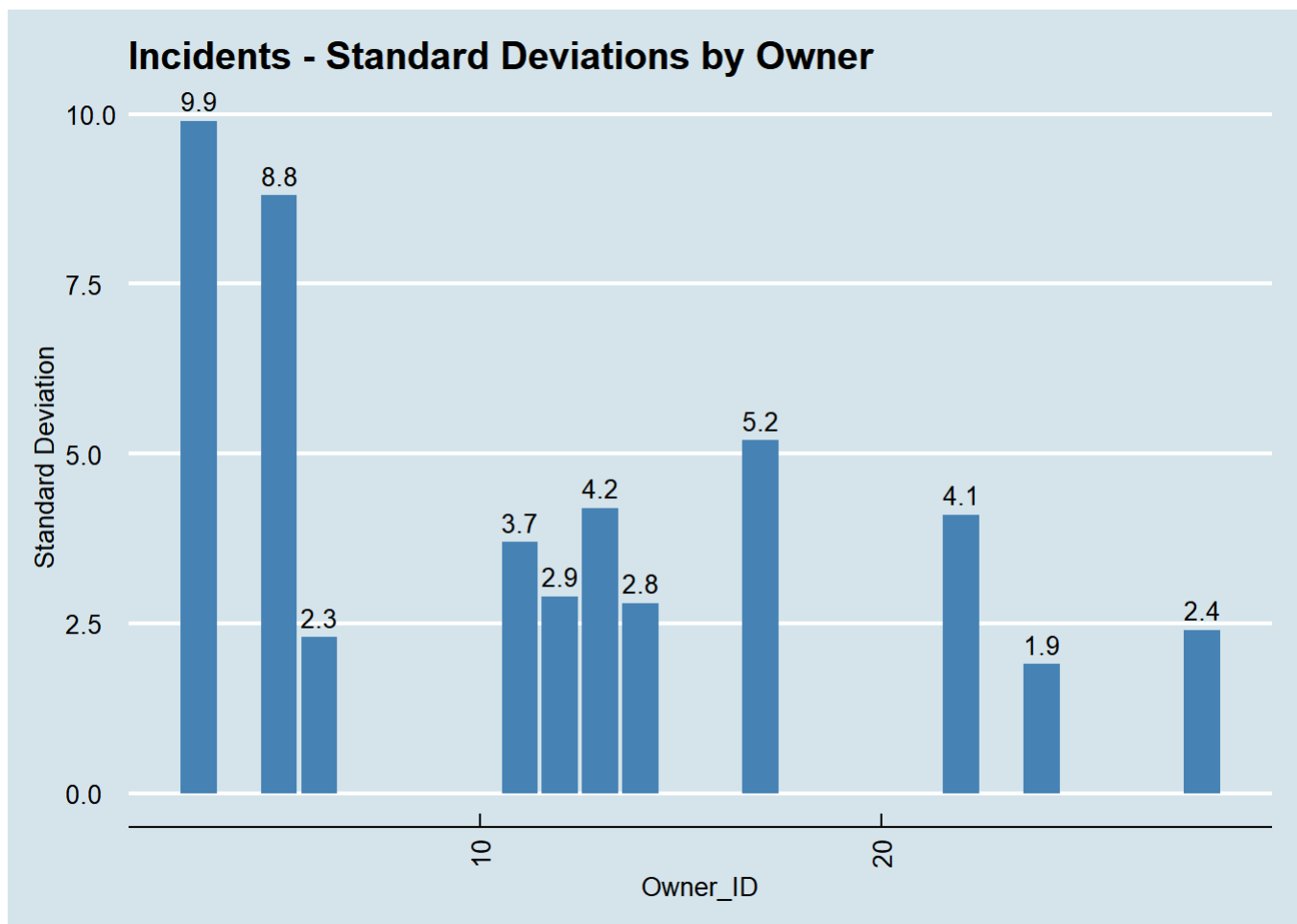


We are pleased to see that we have a couple of potential superstars on the team, but the next question would be, how quickly do they handle all of those? And what about those who handle fewer? Are they able to handle them faster because they have fewer?

Looking at average duration times we see that the two staffers handling the larger numbers of incidents are also very competitive when it comes to duration times. The same is not true of the first staffer, who handled relatively few incidents and also tends to take a good deal longer to close an incident. We have already learned through interviews, however, that in this case we have a situation where recordkeeping is batched; therefore we cannot accurately reflect the time is actually took to close on those tickets.



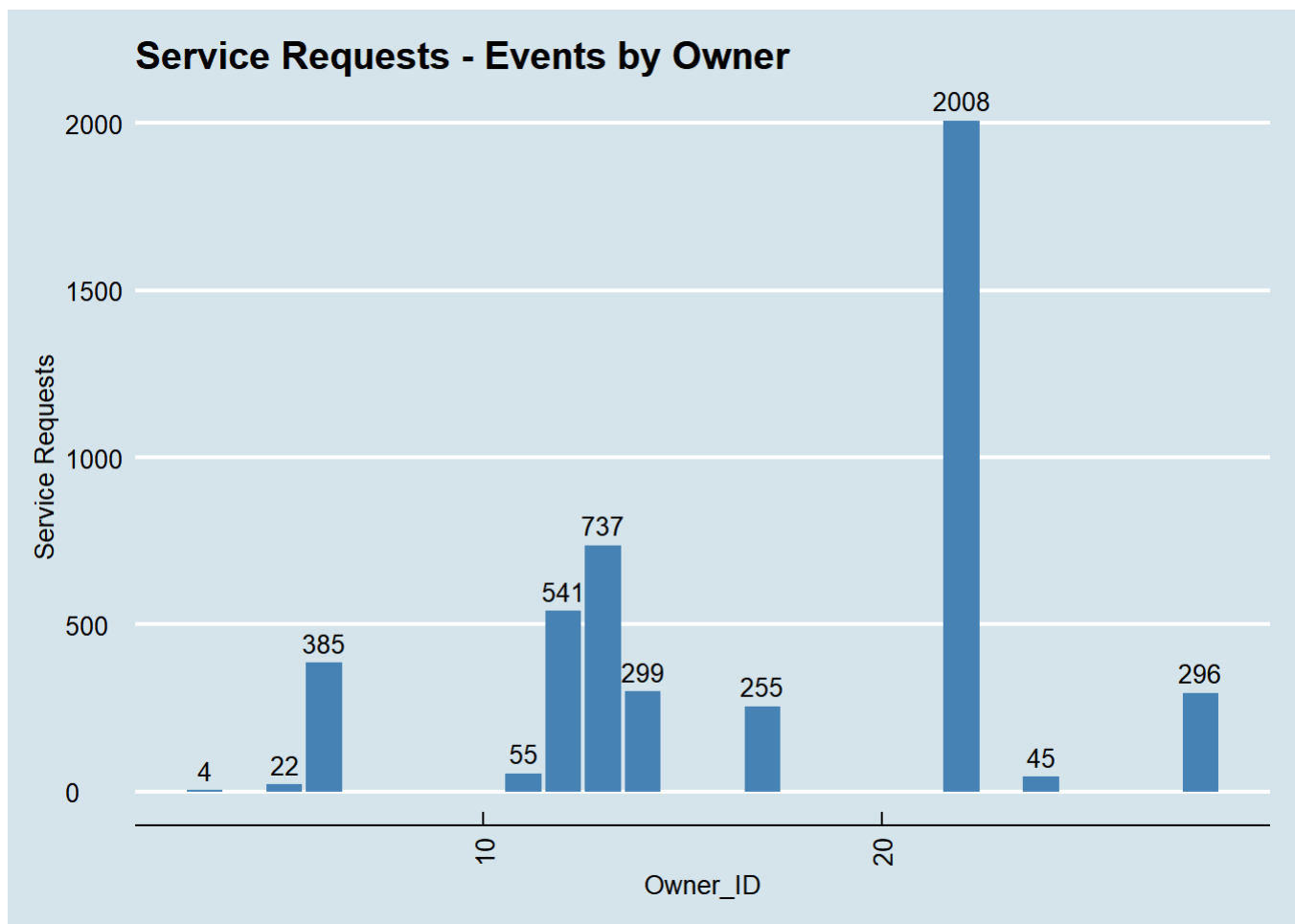
When we look standard deviations, we see that the two staffers handling the most incidents are also very competitive relative to the total time needed to handle a given set of tickets. We also see one staff member with a standard deviation of 1.9, which implies that in that one case, all of that person's tickets would be addressed within the SLA target.



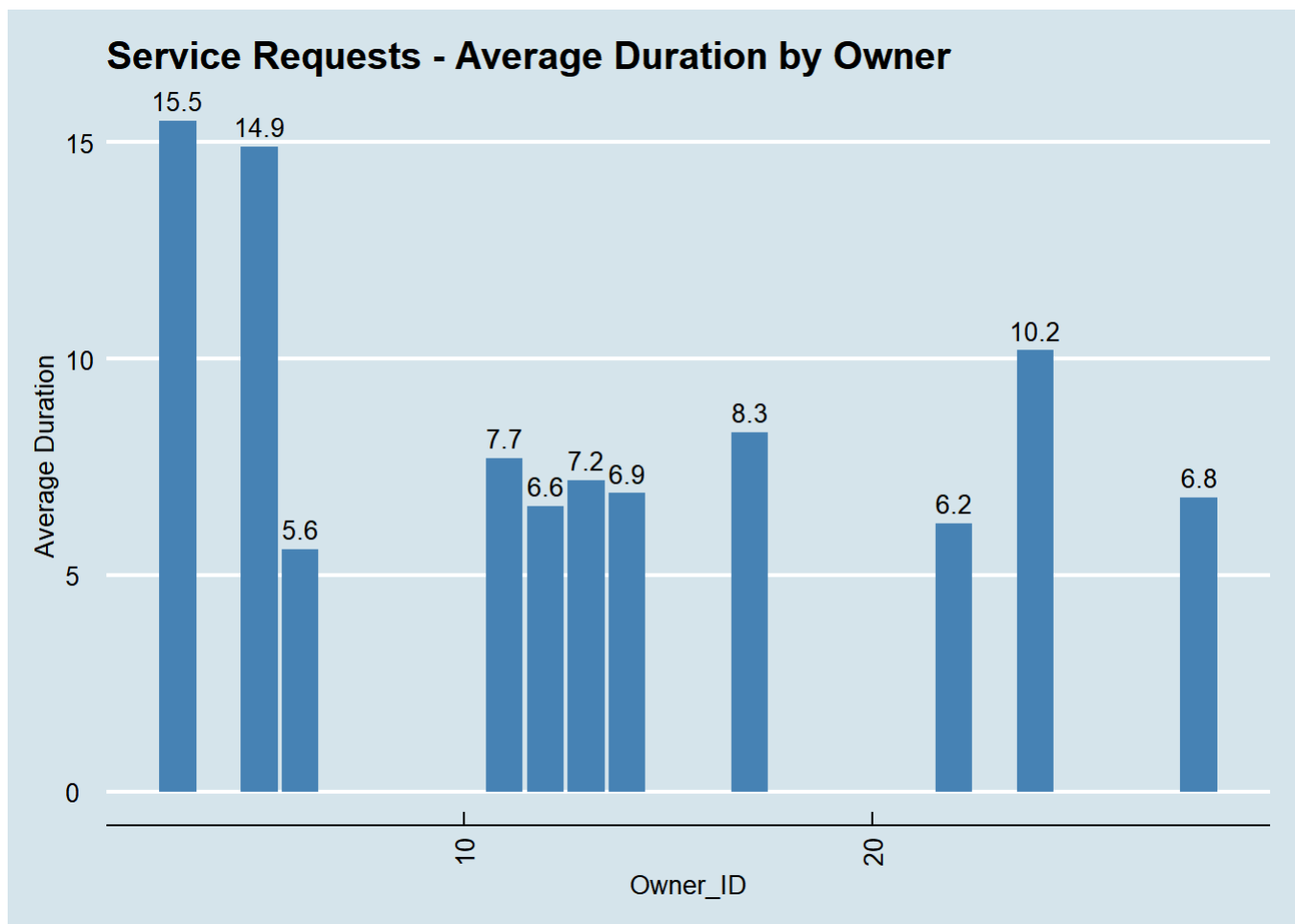
If we think of the standard deviation as a kind of efficiency metric, we see a kind of tiering here as well. Five staff have standard deviations of under 3; four more are in the range of 3-6, and 2 are above that. In this case too, there may be very good reasons for this - a person might be generally assigned to longer-term projects or particularly knotty problems.

Service Requests

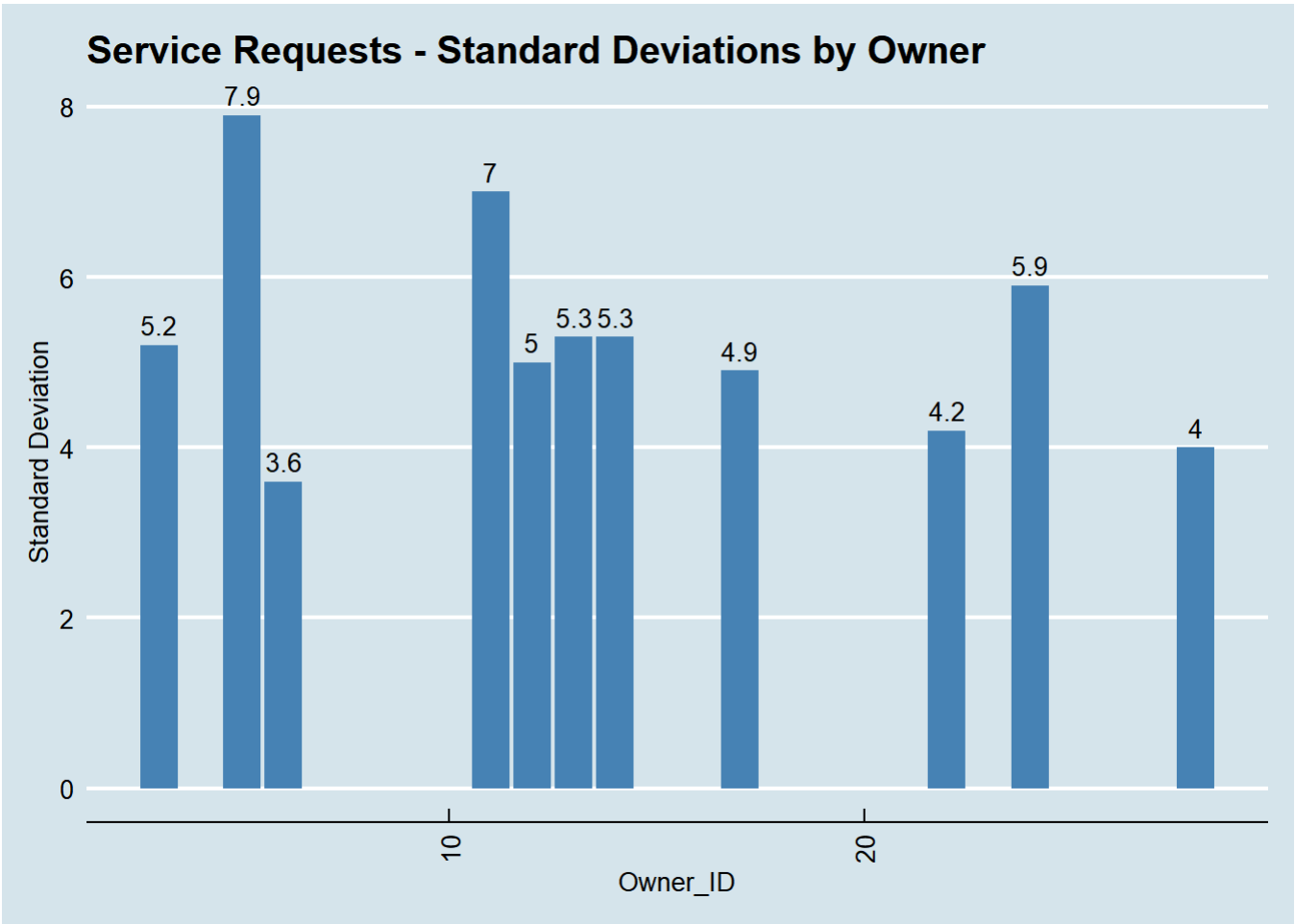
When we look at service requests we see a different set of dynamics for numbers of tickets addressed. One staff member is handling 39% of all requests, and the rest of the team about one-tenth that number (on average). As noted before, there may be very good reasons for this, but we also remember that this same staff member (ID #22) was handling many incident tickets as well.



When we look at average duration, we see more of a two-tier structure. One group tends to get tickets resolved in 5-10 days (within SLA limits), and a second group (the two other members) does not.



In terms of standard deviation, we see five staffers who can get 95% percent of their tickets done within the SLA target (that is, they have a standard deviation of 5 or less). The others would complete at least 68% of their tickets within the SLA target, but no one is performing at a level where all of their work would be within the SLA range.



So who is the best performer overall? Of course, it depends on what you are measuring. But to pose a hypothetical example, we could choose 3 metrics defined this way:

- Volume: the overall number of incidents handled
- Efficiency: how quickly an incident was handled (on average)
- Speed: the number of days it takes to handle a set of requests

We might see a ranking like this, giving a 1 for every first place, a 2 for every second place, and so on. The last column gives the composite ranking and the winner:

##		X Inc	X.1	X.2	X.3	X.4	SerReq	X.5	X.6	X.7	X.8	Both	X.9
##	1	Owner_ID	Vol	Eff	Speed	Total	Rank	Vol	Eff	Speed	Total	Rank	Comb Rank
##	2	3	11	11	11	33	11	11	11	6	28	10	21 10
##	3	5	9	10	10	29	10	10	10	11	31	11	21 10
##	4	6	7	2	2	11	2	4	1	1	6	1	3 1
##	5	11	8	8	6	22	8	8	7	10	25	8	16 8
##	6	12	6	5	5	16	5	3	3	5	11	3	8 5
##	7	13	3	6	8	17	7	2	6	7	15	5	12 6
##	8	14	1	1	4	6	1	5	5	7	17	6	7 4
##	9	17	5	9	9	23	9	7	8	4	19	7	16 8
##	10	22	2	7	7	16	5	1	2	3	6	1	6 2
##	11	24	10	3	1	14	4	9	9	9	27	9	13 7
##	12	28	4	4	3	11	2	6	4	2	12	4	6 2

Any such ranking has to be devised by management - to choose those criteria that are the most important. The above is only for illustrative purposes, but it may prove to be directionally useful.

Building a Predictive Model

It is clear from the analysis so far that there is room for improvement in the performance of the service desk staff. The average durations are over the SLA target for incidents and, while under the SLA for service requests, are not particularly encouraging given the number that are *over* that target.

We also learned that just using simple averages would give us a basic predictive model should the service desk wish to provide clients with an initial estimate to set expectations. Given the wide range of standard deviations, however, this estimate could be misleading if not tied to a specific staff member.

The simplest of all models would be to just take one grand average, but we have already seen that we can do better than that at the category level. The question is, can we do better than that?

At the outset, we have to keep in mind that predictive models seek to weed out the noise in a data set - outliers and oddballs of various kinds. By limiting the duration ceiling to 30 days, *we have effectively removed much of this noise already*. Therefore we have to be prepared to look at model building as a means to verify that this step was a good one, rather than as a means to dramatically improve our ability to estimate durations.

The removal of events with a duration greater than 30 days has improved our estimates by shortening the range of durations and thereby lowering the average. Therefore we expect that a technique such as regularization will not help much. However, we may see some benefit from accounting for category- or owner-specific effects.

This section has two parts. In part one, we make the most use we can of averages by category and owner to provide service desk staff with a reasonable estimate of duration times for a given type of problem handled by a given person. In part two, we see if we can improve on those averages.

Using Averages to Set Expectations

When responding initially to the event, the initial response could be simple: just say “6-7 days on average” as the current average duration for all events - incidents or service requests - are so close (about 7 days if we round the number).

Going one step further, we could just give the average duration for given category. Or, if we know if it is an incident vs. a service request, give the average for that category of that event type. We can go one step further than that, and map a given service desk staff member to a given category and quote the average duration for that person working on that kind of problem. That is about as far as we can go at this point - or at least, until we have better data to work with.

If a category is not listed, the estimate is zero, because the data has indicated that the average duration for that category is zero. A crude model to be sure, but it has two advantages: first, it is simple and practical. Second, it provides a starting point for individuals to up their game and see how far they can drive down the duration times (min, average, and max) for a given category.

Below is such a table mapping problem categories against the staff who have worked on them.

Category	11	12	13	14	17	22	24	28	3	5	6
Access or Login	NA	5	5	5	8	7	NA	6	NA	NA	7
Acrobat Pro	NA	9	4	NA	NA	NA	NA	NA	NA	NA	4
Adobe Analytics	NA	NA	NA	4	NA	NA	NA	NA	NA	NA	NA
Adobe CC	NA	5	NA	NA	NA	5	NA	NA	NA	NA	NA

Category	11	12	13	14	17	22	24	28	3	5	6
Adobe Reader	NA	NA	4	5	NA	NA	NA	NA	NA	NA	NA
BetterCloud	NA	5	5	NA	NA	NA	NA	NA	NA	NA	NA
Breaches and Phishing	8	5	NA	10	NA	11	NA	NA	NA	NA	6
Browsers - Other	NA	4	NA	5	NA	NA	NA	NA	NA	NA	NA
Calendar	NA	NA	5	5	NA	7	NA	NA	NA	NA	4
Caspio	NA	NA	NA	NA	6	NA	NA	NA	NA	NA	NA
Cherwell	NA	NA	NA	NA	NA	4	NA	NA	NA	NA	NA
Chrome	NA	8	8	5	NA	13	NA	NA	NA	NA	4
Computer Accessories	NA	NA	NA	4	6	8	NA	5	NA	NA	NA
Concord	NA	NA	NA	4	NA	NA	NA	NA	NA	NA	NA
Creative Cloud	NA	12	NA	10	NA	6	NA	NA	NA	NA	NA
Cyberduck	NA	4	NA	NA	NA	4	NA	NA	NA	NA	NA
Docs and Sheets	NA	NA	7	6	NA	NA	NA	NA	NA	NA	NA
Dropbox	NA	NA	3	NA	NA	NA	NA	NA	NA	NA	5
Enterprise	NA	5	4	5	NA	4	NA	NA	NA	NA	NA
Faxes	NA	NA	NA	NA	NA	5	5	NA	NA	NA	NA
File Access	NA	7	7	5	NA	13	NA	NA	NA	NA	5
Firefox	NA	NA	NA	5	NA	6	NA	NA	NA	NA	NA
Functional Issues	NA	5	8	5	9	7	4	6	18	NA	4
GA and Other	9	5	6	4	8	6	NA	6	NA	NA	5
Go-Global	NA	NA	NA	NA	NA	5	NA	NA	NA	NA	4
Google Apps - Other	NA	6	NA	4	NA	4	NA	NA	NA	NA	4
Greenshot	NA	NA	NA	NA	NA	8	NA	NA	NA	NA	NA
Groups	NA	NA	NA	NA	NA	7	NA	NA	NA	NA	NA
IE / Edge	NA	NA	NA	NA	NA	NA	NA	5	NA	NA	NA
InCopy	8	4	NA	NA	NA	NA	NA	NA	NA	NA	NA
InDesign	NA	4	NA	4	NA	4	NA	NA	NA	NA	NA
JSHOL	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	5
K4	NA	5	8	NA	NA	7	NA	NA	NA	NA	4
Kace	NA	NA	8	NA	NA	5	NA	NA	NA	NA	NA

Category	11	12	13	14	17	22	24	28	3	5	6
Keychain	NA	NA	4	4	NA	7	NA	NA	NA	NA	4
Lastpass	NA	6	5	5	NA	5	NA	NA	NA	NA	NA
LibreOffice	NA	NA	NA	NA	NA	8	NA	NA	NA	NA	NA
Livelink	NA	NA	NA	NA	NA	5	NA	NA	NA	NA	NA
M-Files	NA	5	NA	NA	NA	4	NA	NA	NA	NA	NA
Mail	6	8	6	4	6	5	NA	6	NA	NA	4
Merlin	4	NA	NA	NA	NA	NA	NA	NA	NA	7	NA
Monitors	NA	6	NA	4	14	6	7	5	NA	NA	4
Move / Close / Remove	NA	5	10	4	7	6	NA	5	NA	NA	NA
MS Excel	NA	NA	7	12	NA	5	NA	NA	NA	NA	NA
MS Office Other	NA	NA	5	4	NA	5	NA	NA	NA	NA	NA
MS Powerpoint	NA	NA	NA	NA	NA	NA	NA	NA	NA	8	NA
MS Windows	NA	NA	NA	5	NA	5	NA	NA	NA	NA	NA
MS Word	NA	4	4	9	NA	9	NA	NA	NA	NA	NA
Network / Connectivity	NA	4	7	6	5	7	NA	5	13	8	4
Outage or error	13	6	6	5	6	7	NA	6	NA	NA	5
Password	NA	NA	5	5	NA	7	NA	6	NA	NA	17
Phones	NA	6	4	4	8	10	NA	7	NA	17	5
Printing / Scanning	6	3	6	4	7	7	4	5	NA	NA	4
Pulse Secure	NA	5	4	8	NA	4	NA	NA	NA	NA	7
Salesforce	NA	5	NA	NA	NA	5	NA	NA	NA	NA	NA
Security - Other	6	5	14	6	NA	7	NA	NA	NA	14	6
SentinelOne	NA	NA	NA	NA	NA	11	NA	NA	NA	NA	NA
Setup and installs	NA	5	5	5	7	7	5	7	NA	NA	5
Skype	NA	NA	9	4	NA	4	NA	6	NA	NA	4
Slack	NA	4	4	4	NA	NA	NA	NA	NA	NA	NA
Teleconferencing	NA	NA	18	4	6	NA	NA	NA	NA	NA	NA
Training	NA	13	6	4	NA	5	NA	NA	NA	NA	NA
Updates / Upgrades	NA	4	6	9	6	7	NA	NA	NA	NA	NA
Zoom	NA	11	6	5	7	6	NA	5	NA	NA	NA

We notice that there are many NAs, indicating that oftentimes a given person does not handle a given category / problem type (and vice-versa). Nonetheless, in the absence of better recordkeeping and because some staff tend to handle certain kinds of problems, it is a reasonable-enough mechanism to predict likely resolution times.

A model to Predict Duration Times More Accurately

To see if a predictive model can improve accuracy, we will focus on incidents for two reasons. First, management has expressed the desire to focus on improving team performance in that area first; second, the tendency for service requests to have no real prioritization (and therefore questionable statistics relative to average durations) makes it problematic to build such a model.

To see if our model is useful, we will first establish a general baseline, and then use the RMSE (residual mean squared error, aka root mean squared error) as a measure of improvement.

RMSE is the standard deviation of the residuals (the prediction errors). Residuals are a measure of how far from the regression line data points are - and how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit.

We calculate RMSE as the square root of the average of the actual duration minus the predicted duration squared. In R programming terms, it is:

```
sqrt(mean((act_duration - pred_duration)^2))
```

RMSE is commonly used in forecasting and regression analysis to verify experimental results. In practical terms, it will tell us how far off the mark (the regression line) our data tends to be, and if we can get closer with some adjustments.

To build the model, we divide the data into a training set and a test set. We will develop a model with the training set and then test it on the test set. This is done to avoid overfitting the model (for example, if we used the entire data set to develop the model and never tested it, we would not know if it gave good predictive results or not). In this instance, we will divide the dataset of all incidents randomly - in effect, taking a statistically useful sample to create the test set. The training set will have 80% of the incidents, the test set 20%.

Step 1: Establish a Baseline - the Average Duration for All Incidents

The simple average for all events is one place - and the easiest place - to start. It provides a benchmark for improvement. As noted elsewhere, the mere act of limiting duration times to 30 days *will significantly improve average duration times* and therefore the average will be better than would ordinarily be the case. When looking at the average and running it through our RMSE formula, we get our benchmark RMSE to see if we can do better.

method	RMSE
Simple Average (Mean) Duration	3.933

So at the very least, we have a starting point - and RMSE for the basic, overall average from all incidents in the 30-day cutoff dataset.

Step 2 - Accounting for Category Bias

Next we introduce a factor we will call “beta c” or b_c , to account for general bias in category durations. We know that different categories are treated differently - some are more important than others and presumably get faster attention. The term b_c represents the average duration for a given category c . The code looks like this:

```
# Calculate  $b_c$  as the average of a given category's duration minus the overall average
mu <- mean(train_set$Dur_Time)

# Set up dataframe to capture the results
category_avgs <- data.frame("L3C" = 1:num_L3,
                             "b_c" = 1:num_L3)

counter <- 1
for(i in 1:num_L3) {
  x <- train_set %>% filter(L3C == i)
  if(nrow(x) != 0) {
    category_avgs[counter,1] <- i
    category_avgs[counter,2] <- mean(x$Dur_Time - mu)
    counter <- counter + 1
  }
  else {
    category_avgs[counter,1] <- i
    category_avgs[counter,2] <- 0
    counter <- counter + 1
  }
}
```

The result is a list with 89 rows, one for each category, and the output looks like this:

```
##   L3C      b_c
## 1    1 -0.4464752
## 2    2  0.0000000
## 3    3  0.0000000
## 4    4 -1.1131419
## 5    5  0.0000000
## 6    6  0.0000000
```

We then join this list to our test set and calculate the new RMSE and see if it is better than our simple average, using this code:

```
# Generate predicted durations returned by the model
predicted_durations <- mu + test_set %>%
  left_join(category_avgs, by = 'L3C') %>%
  pull(b_c)

# Calculate the RMSE
model_1_rmse <- RMSE(predicted_durations, test_set$Dur_Time)

# Add the results to the rmse_results summary
rmse_results <- bind_rows(rmse_results,
  data_frame(method="Category Effects Model",
    RMSE = model_1_rmse))

# Display the result
knitr::kable(rmse_results)
```

method	RMSE
Simple Average (Mean) Duration	3.93300
Category Effects Model	4.12702

We see that the new RMSE value is not lower - the simple average is still better (it has the lower RMSE value). Apparently there are no factors causing a given category type to be significantly more complex or difficult that would cause extraordinarily long duration times - perhaps our 30-day cutoff did the trick already?

Step 3 - Accounting for Who Works the Problem

Next, we will introduce a term we will call “beta o” or b_o , to account for owner-specific effects when handling categories of various kinds. As noted above, this is used to see if there are significantly different factors at work that would cause one person to consistently have *much* better durations than another for a given problem category. As we did with the category effects model, we calculate the b_o value for each of the staff members, join this to the category effects values which we have already joined to the test data. The code to do the calculation looks like this:

```
# Generate predicted durations returned by the model
predicted_durations <- test_set %>%
  left_join(category_avgs, by = 'L3C') %>%
  left_join(b_o_values, by = 'Owner_ID') %>%
  mutate(pred = mu + b_c + b_o_value) %>%
  pull(pred)
```

...and the resulting RMSE is shown below:

method	RMSE
Simple Average (Mean) Duration	3.93300
Category Effects Model	4.12702
Owner Effects Model	4.12515

We see that there is the slightest of improvements - certainly not enough to be significant and definitely not better than the average we have been using. This supports the theory that applying the 30-day cutoff for durations may have already removed most (or all) of whatever owner effects there might have been.

Step 4: Regularization

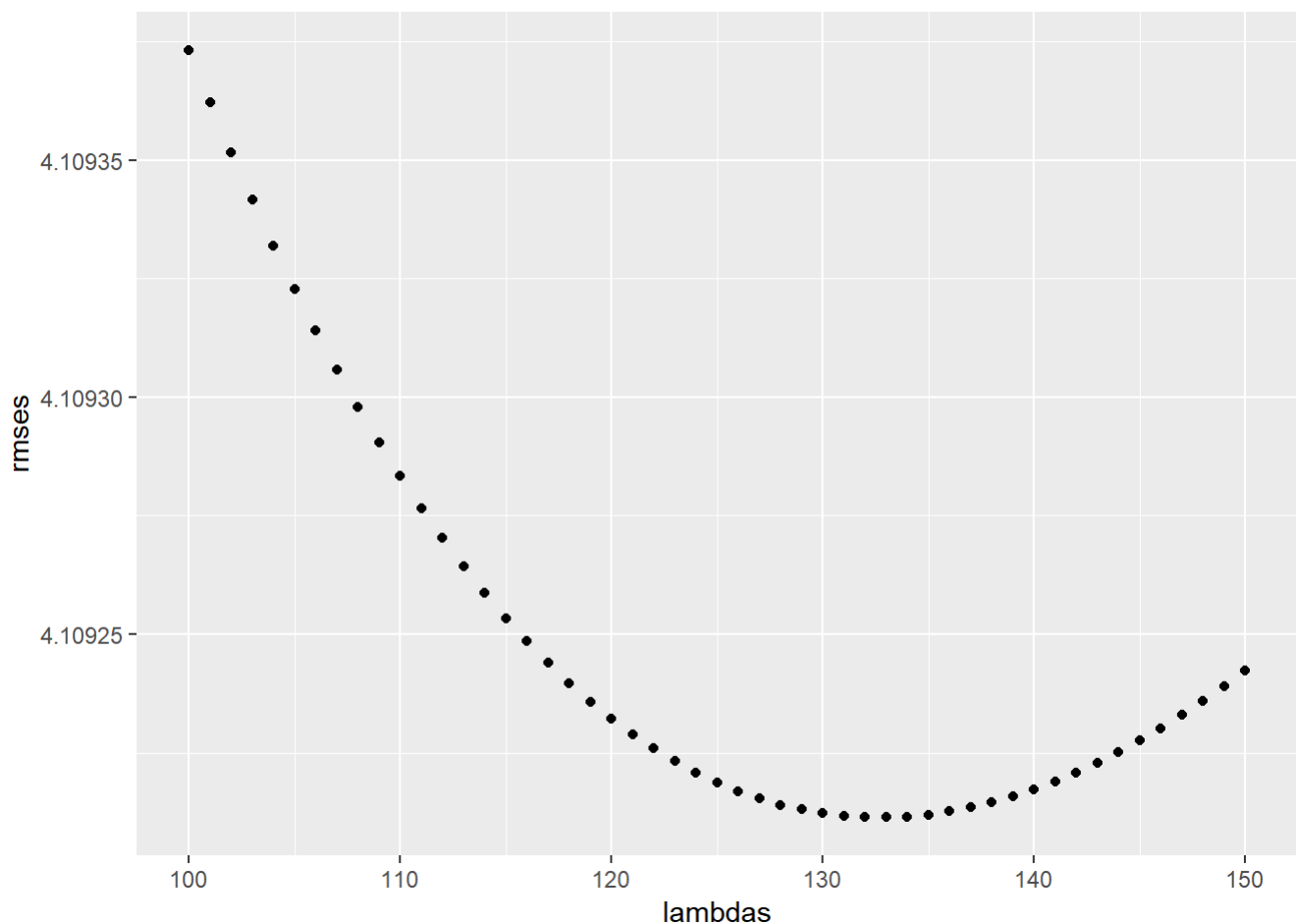
We will now see if regularization can improve the model. Given what we have seen above, we do not expect it to. Regularization is used to remove the effects of extremes. In our case, it would remove the effect of, say, a person joining the service desk for a week, handling a number of requests in record time, and then disappearing. Like a batter who hits .427 in the first two weeks of the season, it is not indicative of what the whole season would be like. Regularization compensates for this by applying a penalty for durations that are either way too high or way too low. It is likely, however, that the 30-day cutoff will already have cured this problem as well - or some of it.

The process is to establish a value - call it lambda - experimentally, and then apply it to the test set that has been joined together with the duration time factors for a given category. The code to do this looks like this to arrive at the optimal lambda value and apply it:

```
# Select a Lambda
lambdas <- seq(100, 150, 1)

# Join to the training set, calc new b_c and prediction
rmsees <- sapply(lambdas, function(l){
  predicted_durations <- test_set %>%
    left_join(sum_category_dur_time, by = 'L3C') %>%
    mutate(b_c = scdt / (n_c + 1)) %>%
    mutate(pred = mu + b_c) %>%
    pull(pred)
  return(RMSE(predicted_durations, test_set$Dur_Time))
})
```

Let's see what this gives us.



```
## [1] "Optimal lambda value: 133"
```

When we calculate the new RMSE, we see that the regularized model does a little better than either of the two above, but still does not beat the simple average.

method	RMSE
Simple Average (Mean) Duration	3.933000
Category Effects Model	4.127020
Owner Effects Model	4.125150
Regularized Category Effects Model	4.109211

In the end, our 30-day cutoff apparently had the most leverage in improving average duration times - it effectively blunt-forced a user effects model **and** regularization simultaneously. The key management takeaway is *get rid of the outliers - anything above 30 days*.

Conclusions

- One of the original goals of this analysis was to discover if there was a means to do breach forecasting (determine if a given ticket would exceed the SLA targets). Given that 1439 of the 1451 incidents (99.2%) exceed SLA targets, it is actually safe to assume that *EVERY* ticket will cause a breach. The problem is not

with forecasting, it is with getting duration times down. It is not nearly so bad with service requests, with only 780 of 5209 (15%) exceeding the SLA. Of course, both of these figures assume a 30-day cutoff on durations.

- The number of very high duration tickets is troublesome. If it is bad recordkeeping, that can be fixed. If we are mis-classifying tickets (for example, using them from everything from major projects to a note to call someone back), then they should not be entered into Cherwell. For example, updating 450 desktops to a new virus software package is *not* a ticket, it is a project. We would expect projects to take weeks or months to finish.
- The service desk staff are likely unaware of their performance, or we would not see so many very high durations with no reasonable explanation. There is a lot of data in this report that management can use to convey the key problems of prioritization and recordkeeping consistency. The manager could hold one or more sessions to talk over the findings and create some action steps to improve.
- On the flip side, the staff may have useful insights as to why some of the odd correlations are occurring (for example, when more events cause maximum duration times to increase but minimum duration times to decrease). One suspects there will be some useful information gained from that discussion.

Recommendations

- One piece of feedback TSG has received from the organization is that they need to be faster in responding to problems and requests. The analysis has uncovered a number of opportunities (all relatively straightforward and easy) to improve duration times and thereby gain a significant jump in speed.
- In this vein, management needs to decide what are the most important metrics; there are many to choose from - for example:
 - Average duration
 - Average or maximum duration for priority 1's
 - Ability to meet SLA targets
 - Number of tickets addressed in a given timeframe
 - ...and so forth
- Once the staff understand the metrics they are going to be measured on, they will likely respond accordingly - and this implies some management mechanism be in place to show them how they are doing (individually and collectively)
- This analysis will remain somewhat flawed until the service desk team dedicates itself to disciplined recordkeeping. This has the following aspects:
 - Adopting a taxonomy that categorizes problems with greater graduality. This will enable the team to identify what problem areas are requiring the most staff time and effort to resolve. One has been proposed here. This, or an alternative, should be built into Cherwell and be a required entry when a ticket is created.
 - Adopting a method for assigning priorities. The 5-level prioritization does not seem to be working. Behavior suggests that there should be perhaps no more than 3:
 - Immediately or same day
 - Within 2 days (incidents) or 5 days (service requests)
 - Best effort depending on other priorities and workload
 - Assuming a reasonable and accurate method of assigning priorities, discipline in addressing them - that is, all Incident 1's are addressed before 2's, and so forth

- Providing accurate descriptions of the problem for every ticket. In many instances, the description field has just a few words or something meaningless to anyone who did not enter the ticket. This can result in misclassification or mistaken prioritization.
- Closing tickets in Cherwell the day they are completed - no more batching of recordkeeping, as it is not possible to generate accurate statistics when some staff are batching their updates.
- Management of the backlog via a daily or weekly aging report so that no high priority ticket gets forgotten, and thereby drives up the average duration times. One is proposed in the appendix.
- Based on the aging report, select 1 or 2 tickets per week on which to conduct a post mortem to determine what when right and what when wrong. Make adjustments in communication, the data, or the process accordingly.

Next Steps

the next steps, which can be taken immediately, are:

- Create a taxonomy that is optimally useful for managing work and priorities. It does not have to be the one in this analysis, but it needs to be granular enough to track improvements in key problem areas.
- Revise the prioritization scheme so it works and drives faster performance
- Build the revised taxonomy and prioritization scheme into Cherwell, so someone entering a ticket *must* enter that information
- Begin classifying tickets under the new rules immediately. The sooner we can gather a year's worth of data (or even 3 to 6 months' worth) the sooner we will know if improved processes and practices are paying off.
- Once we have a critical mass of new data, update this code and re-run everything to see where there have been areas of improvement. Updating code and the attendant tasks should not take more than a few days.

Appendices

Prototype Aging Report

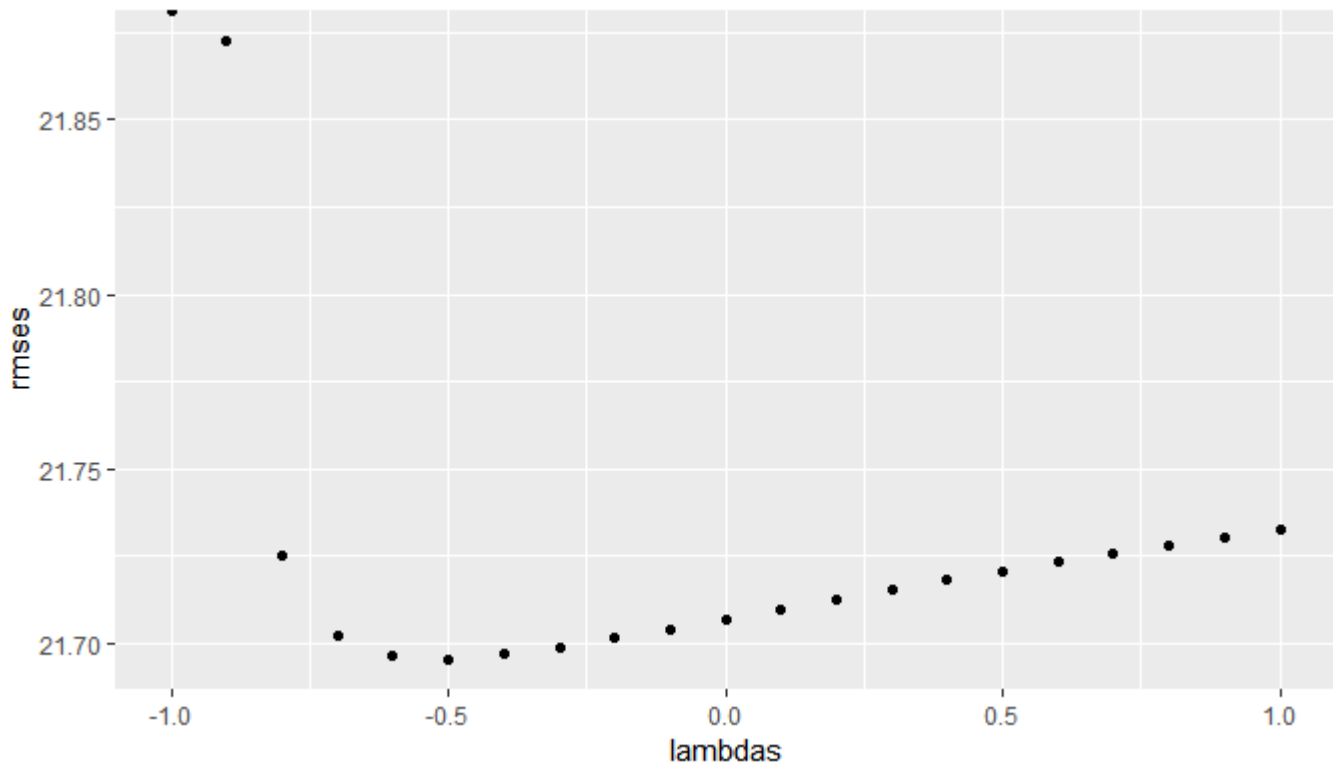
Below we offer the management team a suggested format for a weekly aging report for tracking tickets that are exceeding SLA boundaries. This is easily created from a simple Cherwell extract. Names have been removed and replaced by blanks,

ID	Age	Team	Created	Customer	Status	Owned.By	SLA.Date
30001	61	B&MPS	1/4/2019		In Progress		1/8/2019
29950	62	Business Applications Support	1/2/2019		Assigned		1/4/2019
30711	29	Business Applications Support	2/5/2019		New		2/7/2019
31145	8	Business Applications Support	2/25/2019		In Progress		3/4/2019
22996	357	Developers	3/14/2018		Assigned		3/16/2018
25992	237	Developers	7/11/2018		Assigned		7/13/2018

ID	Age	Team	Created	Customer	Status	Owned.By	SLA.Date
26587	212	Developers	8/6/2018		Assigned		8/6/2018
29282	98	Developers	11/27/2018		In Progress		11/30/2018
31331	1	Hardware	3/5/2019		Assigned		3/18/2019

Output of the Predictive Model with No 30-Day Cutoff

One question that is natural to ask is what would have happened to the predictive model if we had not enforced a cutoff at 30 days? When we run the model under these circumstances, we would have made a lambda selection /between -1.0 and +1.0, with the optimal lambda being negative 0.5.



Alt text

This would have produced model results as follows:

method	RMSE
<chr>	<dbl>
1 Simple Average (Mean) Duration	22.7
2 Category Effects Model	21.7
3 Owner Effects Model	20.8
4 Regularized category Effects Model	21.7

Alt text

In this case, the owner effects model would have produced more accurate results and this is intuitively makes sense if we believe that our 30-day cutoff model effectively removed the owner bias from poor recordkeeping. Without this cutoff, the model has to reinstate it by penalizing those very high duration times.

[End of Document]