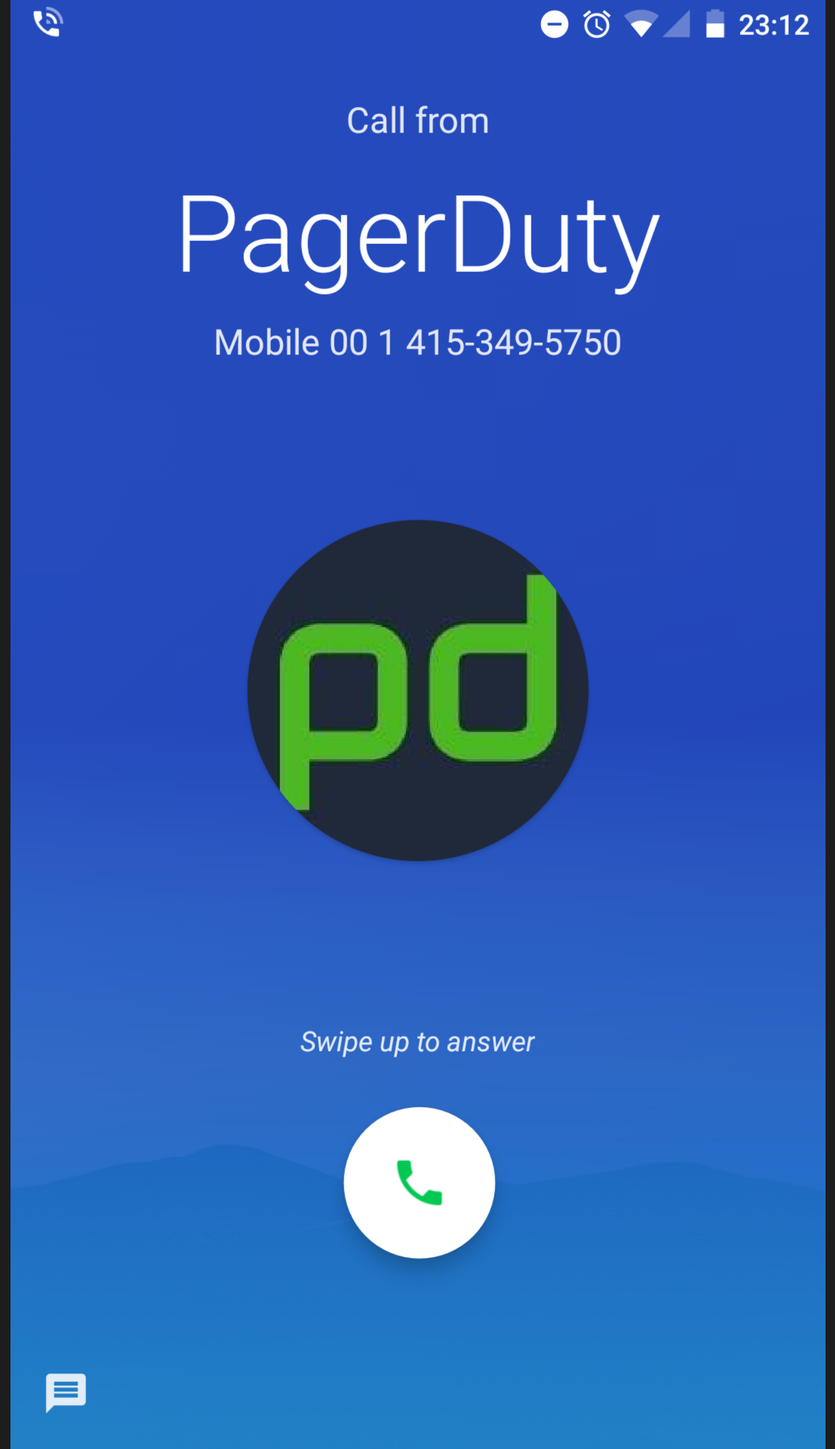
Fixing your noisy pager in 500 easy steps


23:12
Call from
PagerDuty
Mobile 00 1 415-349-5750
Swipe up to answer

🔴 **ServiceDown us-east-1**

Service: Infrastructure
Urgency: ↑ High

Acknowledge **View Details** More actions

🔴 Triggered | 26 September at 16:00 | via Grafana

🔴 **ServiceDown us-east-2**

Service: Infrastructure
Urgency: ↑ High

Acknowledge **View Details** More actions

🔴 Triggered | 26 September at 16:01 | via Grafana

This

is

fixable

Hi

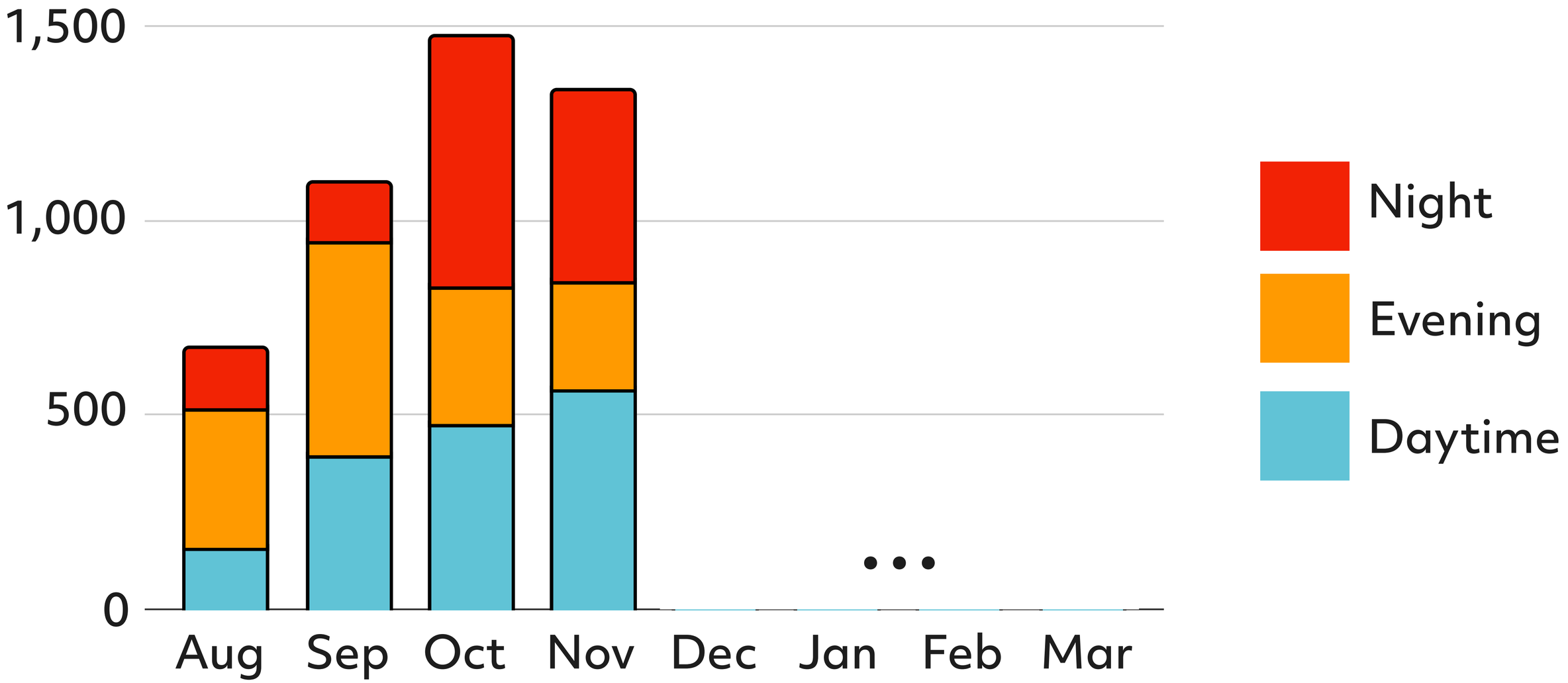
sinjo.dev

sinjo.dev

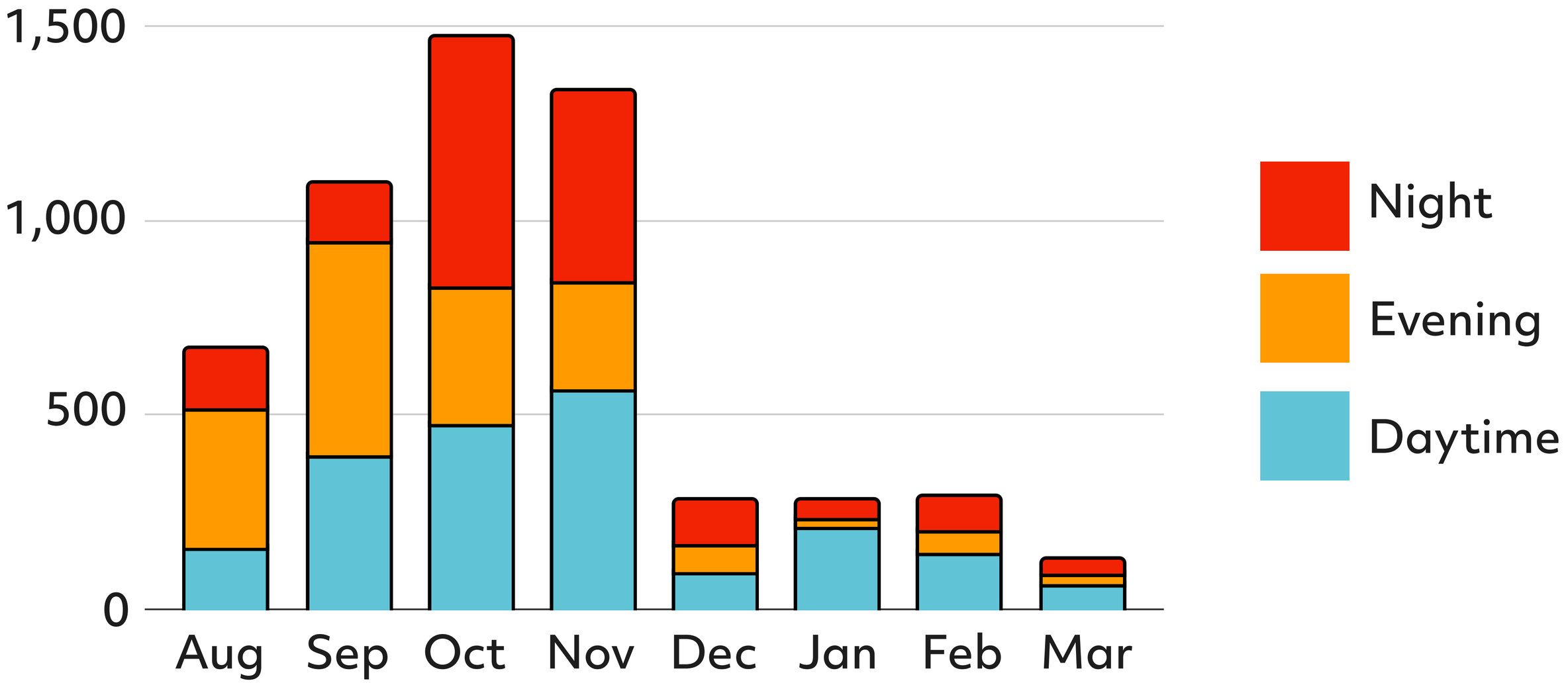
Infra Engineer

PlanetScale

Pages per month (2023–24)

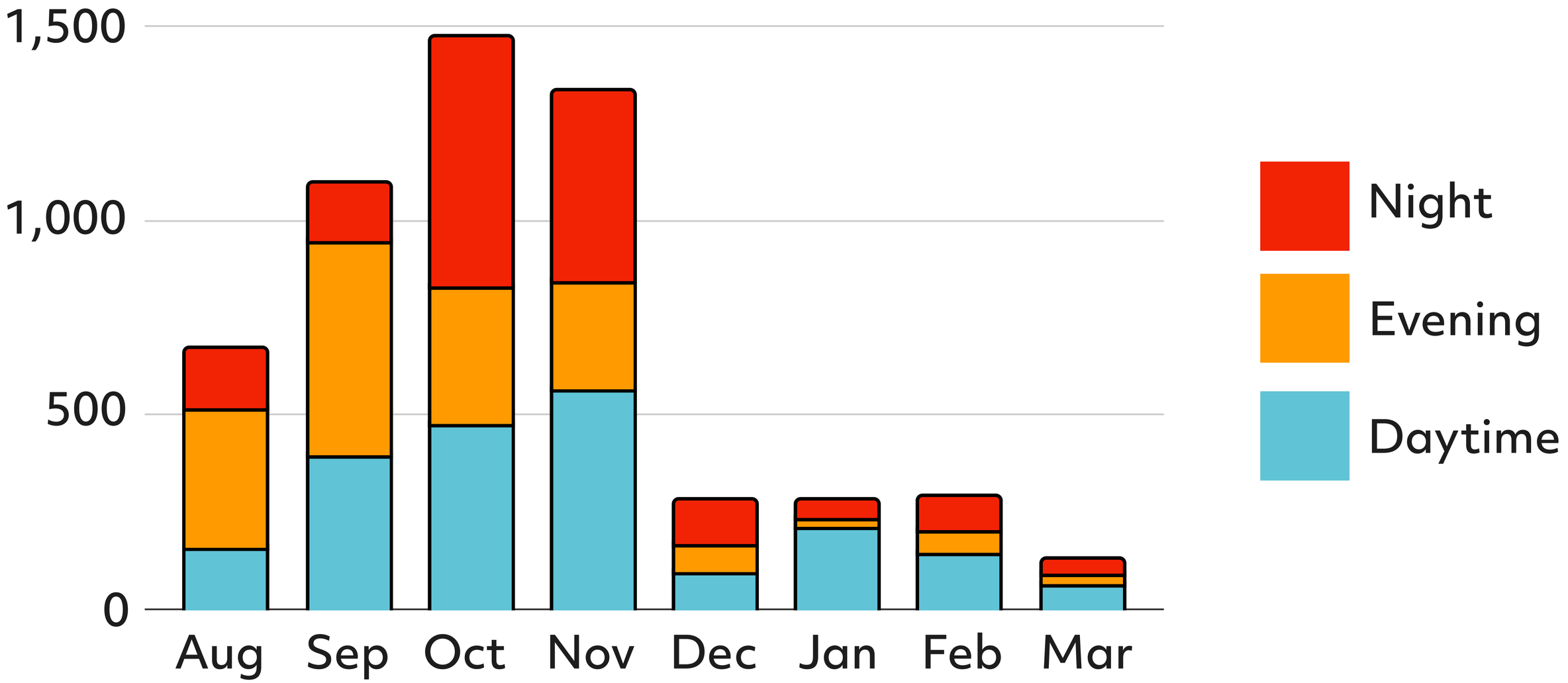


Pages per month (2023–24)

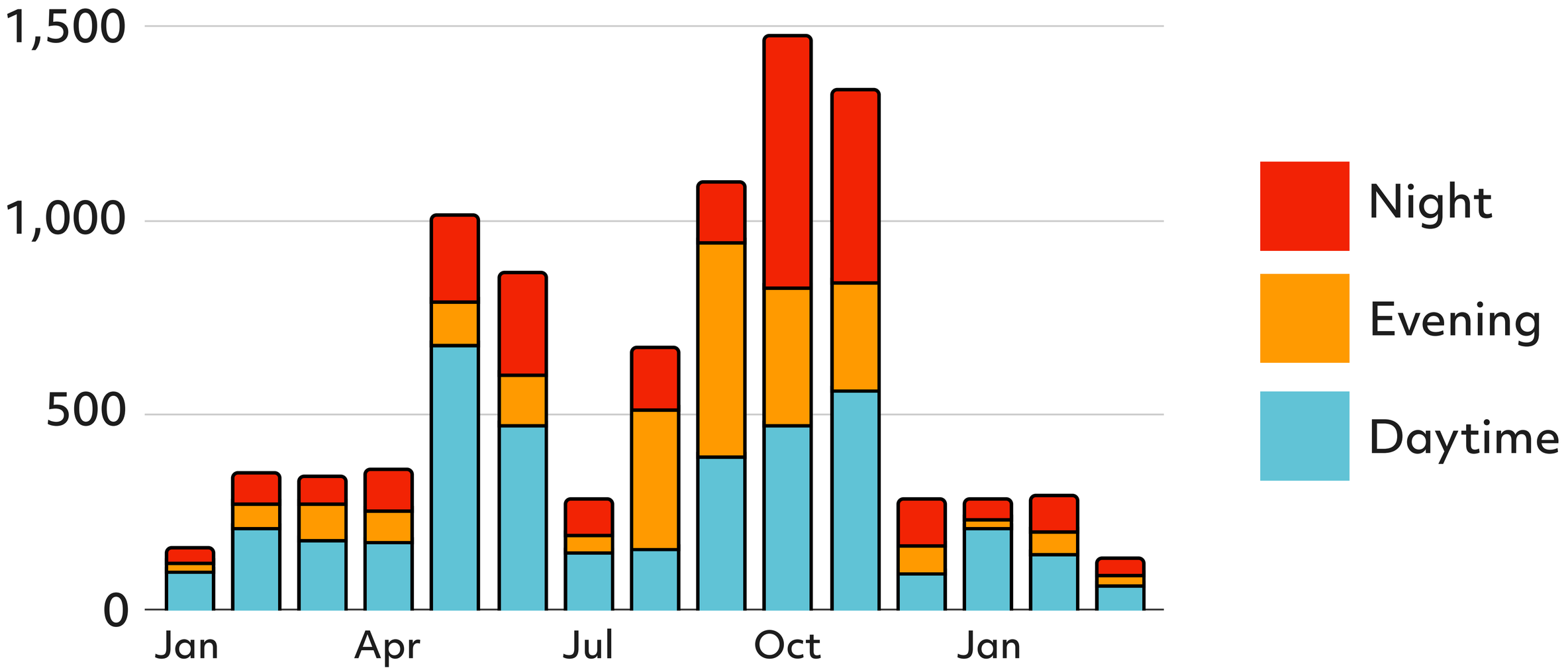


How did we ***get*** here?

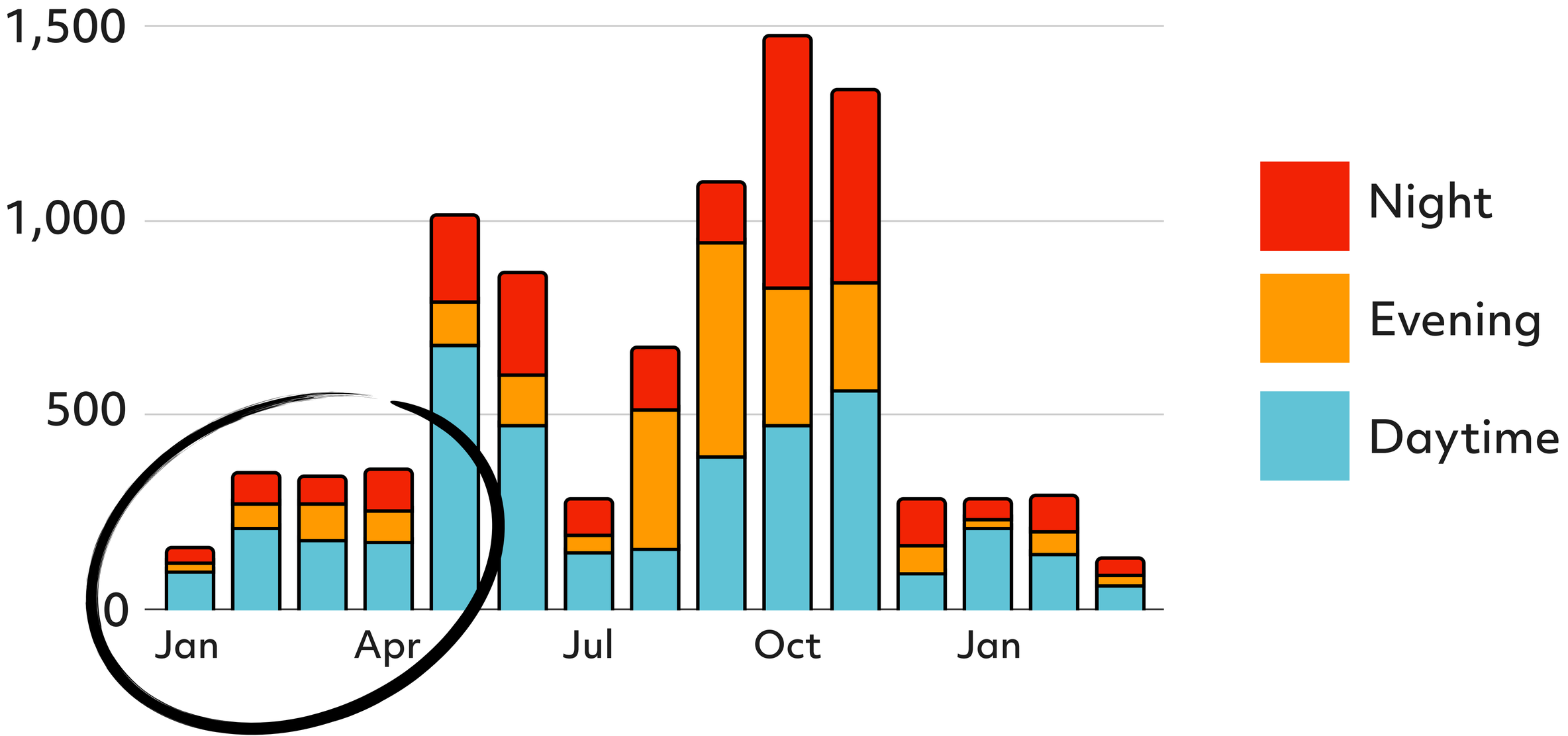
Pages per month (2023–24)



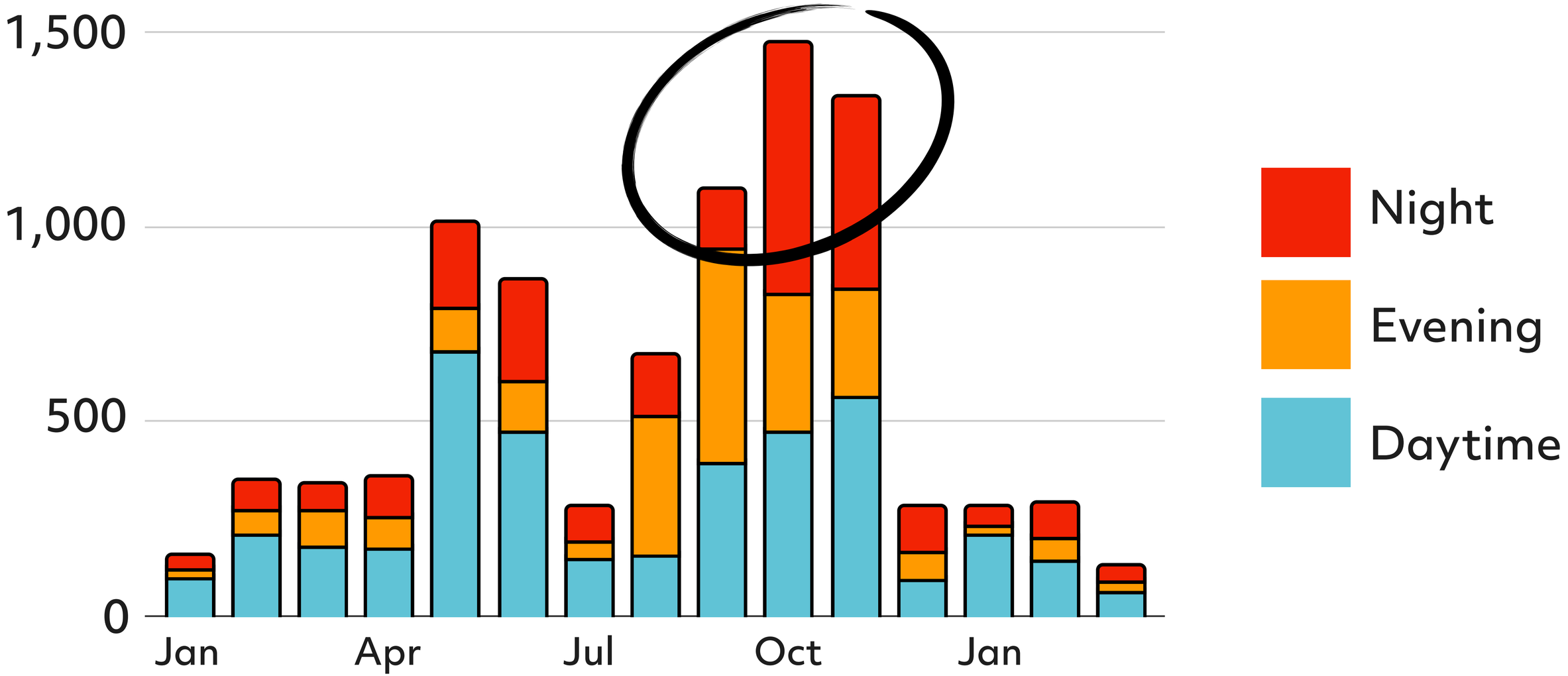
Pages per month (2023–24)



Pages per month (2023–24)



Pages per month (2023–24)



Problem:
convincing
yourselves you
have a problem

We ***didn't*** have those graphs

“Not much to report.”

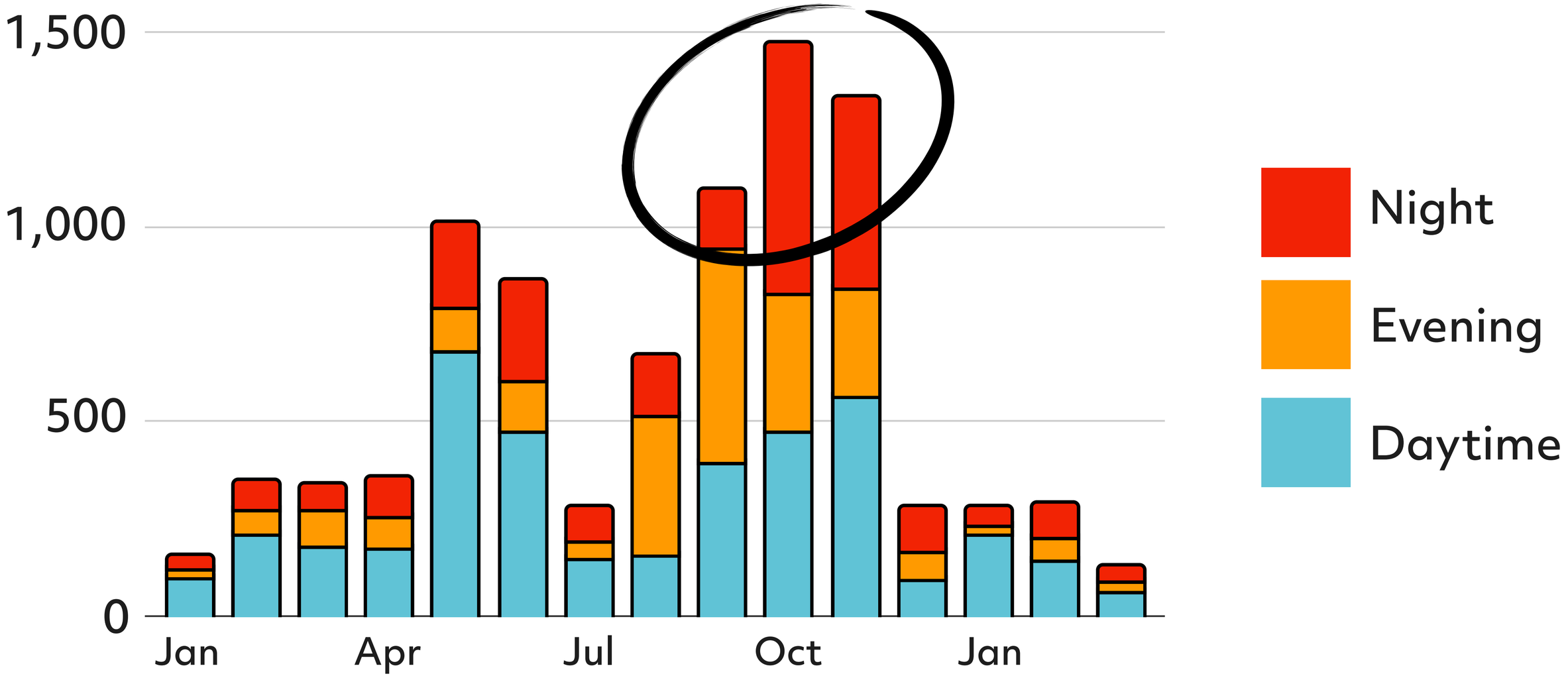
“Nothing interesting from my shift.”

If ***nothing interesting***
happened,
why did a computer
wake you up
every night?

The way out of this situation is

data

Pages per month (2023–24)



So how do we fix it?

1. Group alerts by name

1. Group alerts by name
2. Sort by frequency

1. Group alerts by name
2. Sort by frequency
3. Categorise each alert

Alert frequency

Alert Name	Count
HttpErrorRateHigh	37
TooManyUnhealthyReplicas	21
AutoscalerMaxedOut	5

Categorise

how?

Alerts that are mostly right

vs

Alerts that are mostly wrong

Alerts that are mostly right

vs

Alerts that are mostly wrong

Easier socially

Harder technically

Buy in for

fixing bugs and

improving automation

Alerts that are mostly right

vs

Alerts that are mostly wrong

Easier technically

Harder socially

Changing people's minds is harder than changing a couple of lines of PromQL

Background noise

Alerts that were once useful

Compelling reasons

Compelling reasons

- Pager fatigue: we miss real issues

Compelling reasons

- Pager fatigue: we miss real issues

- Tiredness: people can’t do their best work

Compelling reasons

- Pager fatigue: we miss real issues
- Tiredness: people can’t do their best work
- Learned helplessness: we don’t believe we can improve things

2
choices

Make it more precise

Make it more precise

or

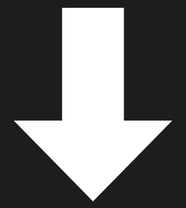
Delete the alert

Bonus

Alerts that are right, but in an annoying way

Excessive urgency

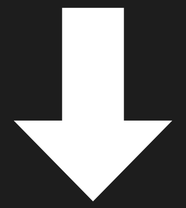
Excessive urgency



Send to Slack or create a ticket

Flappy alerts

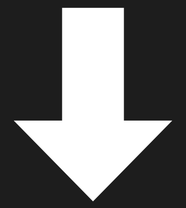
Flappy alerts



Calculate rate over longer window

Pager storms

Pager storms



Use alert grouping/ inhibition

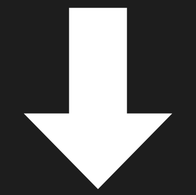
Thorny case:

Real problems in software outside your control

Inside your company,
across team boundaries

Usually fixed by another team?

Usually fixed by another team?



Currently owned by the wrong team

Open source and third-party software

Work with the maintainers!

Possible race condition in EBS volume creation #1951

Closed **Sinjo** opened this issue on Feb 29 · 4 comments · Fixed by #2075

Sinjo commented on Feb 29 • edited by torredil

Hi 👋🏻

I think I've found a race condition that happens very occasionally when creating a PV for a new PVC. The outcome of the race condition is a PVC that is permanently stuck in `Pending` and has to be deleted and recreated.

As a ballpark estimate, based on the number of `CreateVolume` calls per day in CloudTrail and how frequently we see this issue, I would guess that this happens on the order of 1 in 100,000 volume creations.

I've substituted the actual PV and EBS volume names below with placeholders like `pvc-foo` for ease of reading. The identifier of each PVC corresponds directly to the identifier of the EBS volume (i.e. PV `pvc-foo` corresponds to EBS volume `vol-foo`).

https://github.com/kubernetes-sigs/aws-ebs-csi-driver/issues/1951

In the meantime...

protect your team

Automated remediation 101

Restarting the software fixes the software

Restarting the software fixes the software

(temporarily)

Problem shape

Problem shape

- Recurring problem: happens regularly

Problem shape

- Recurring problem: happens regularly
- Reliable detection: highly correlated alert

Problem shape

- Recurring problem: happens regularly
- Reliable detection: highly correlated alert
- Mechanical fix: on-caller follows runbook

Waking someone up to apply a mechanical fix

is a terrible use of their time

Mechanical work is what computers are great at!

We wrote a tool:

`auto-repair`

Write a ***non-paging*** alert that goes off ***before*** your paging one

`auto-repair` (simplified)

auto-repair (simplified)

```
alerts = get("prom:9090/api/v1/alerts")
```

auto-repair (simplified)

```
alerts = get("prom:9090/api/v1/alerts")

issues = filter_fixable(alerts)
```

auto-repair (simplified)

```
alerts = get("prom:9090/api/v1/alerts")

issues = filter_fixable(alerts)

for i in issues do
  // for most issues, restart process
  apply_fix(i)
end
```

It’s that simple

It's that simple

(kinda)

Runaway automation

What the tool
doesn't do is
more important
than what it ***does do***

3 limitations

3 limitations

Don’t restart:

- Too many processes with the same issue

3 limitations

Don’t restart:

- Too many processes with the same issue
- The same instance repeatedly

3 limitations

Don’t restart:

- Too many processes with the same issue
- The same instance repeatedly
- Processes that have already paged

This prevents high ones to low tens of pages per week

(yes, we still file bugs)

What did we learn?

You
need
long-term
buy-in

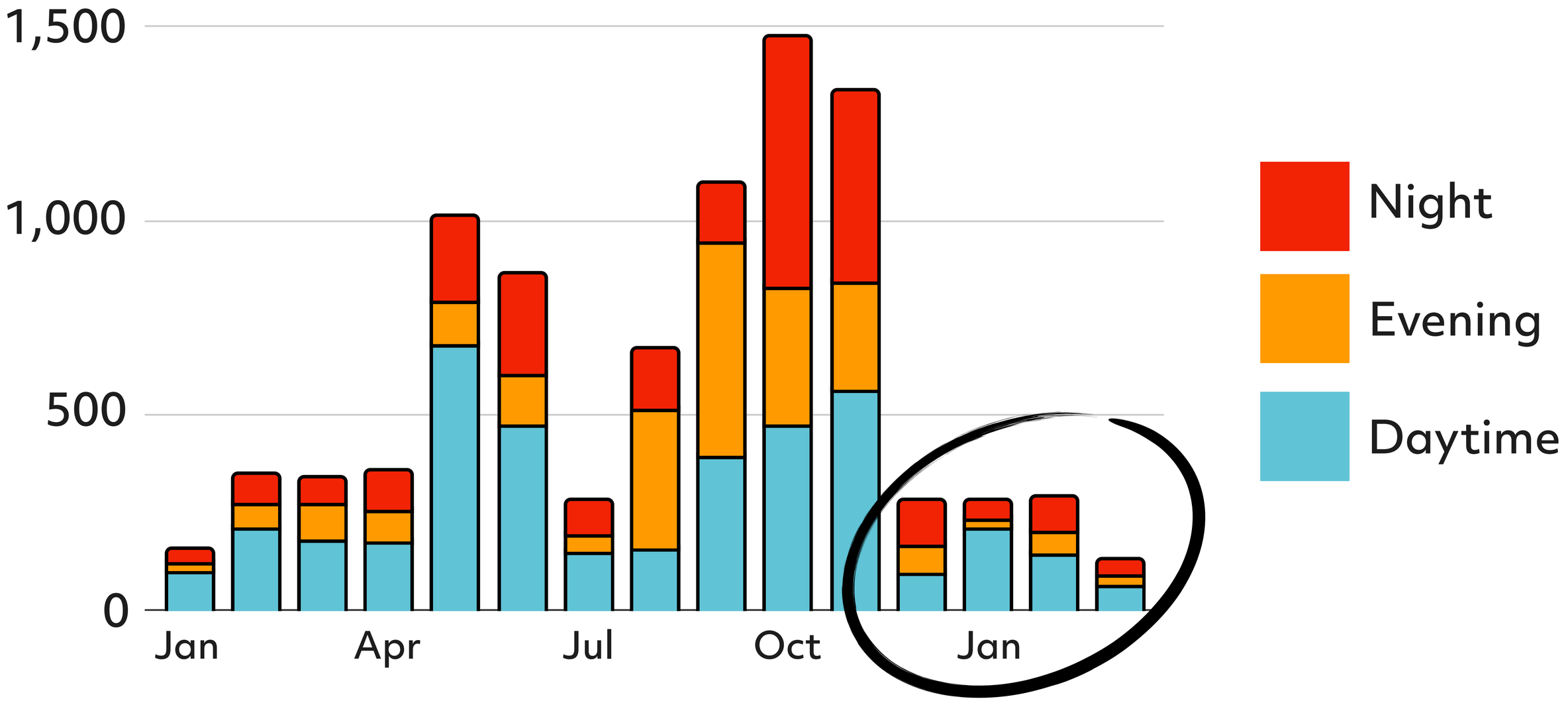
Talk
about
how it
impacts
customers

Embrace hacky fixes that help you survive

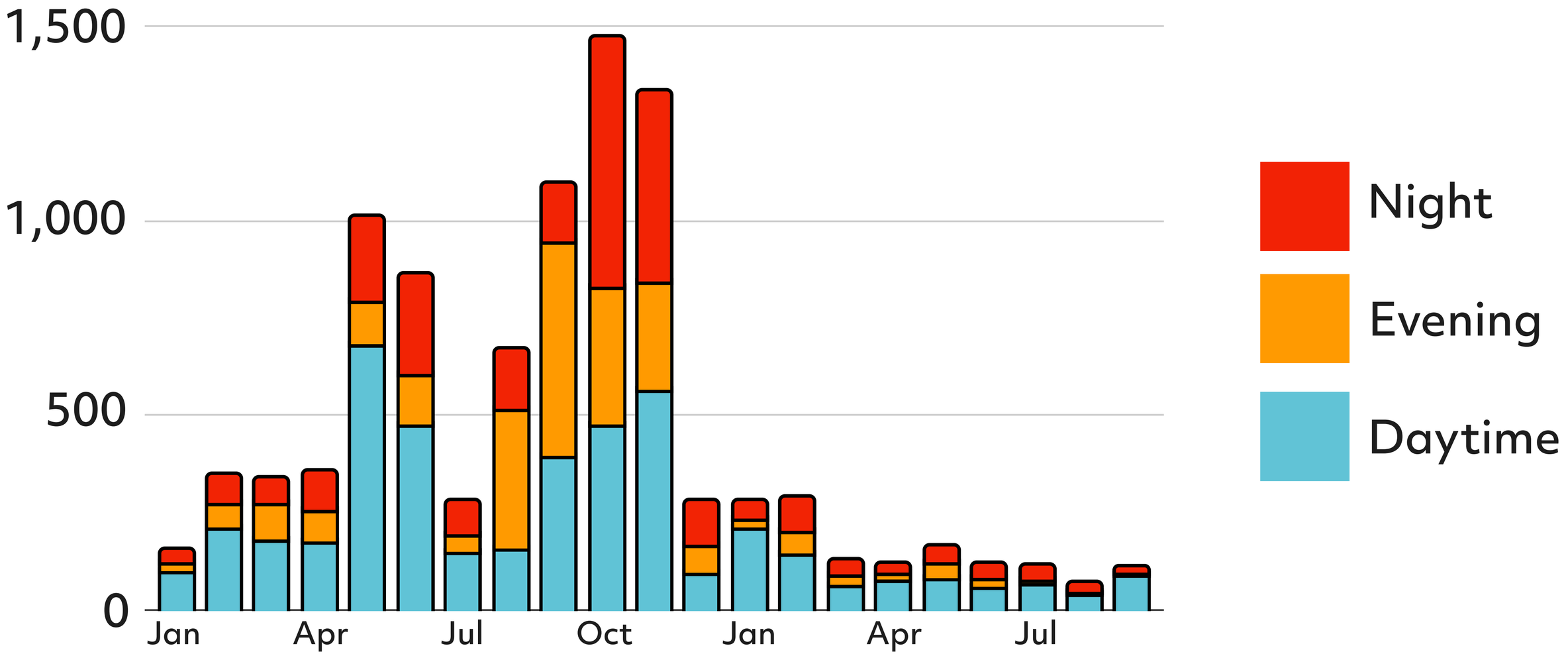
Dumb ideas that work aren't dumb

Good things
happen if you
make it a habit

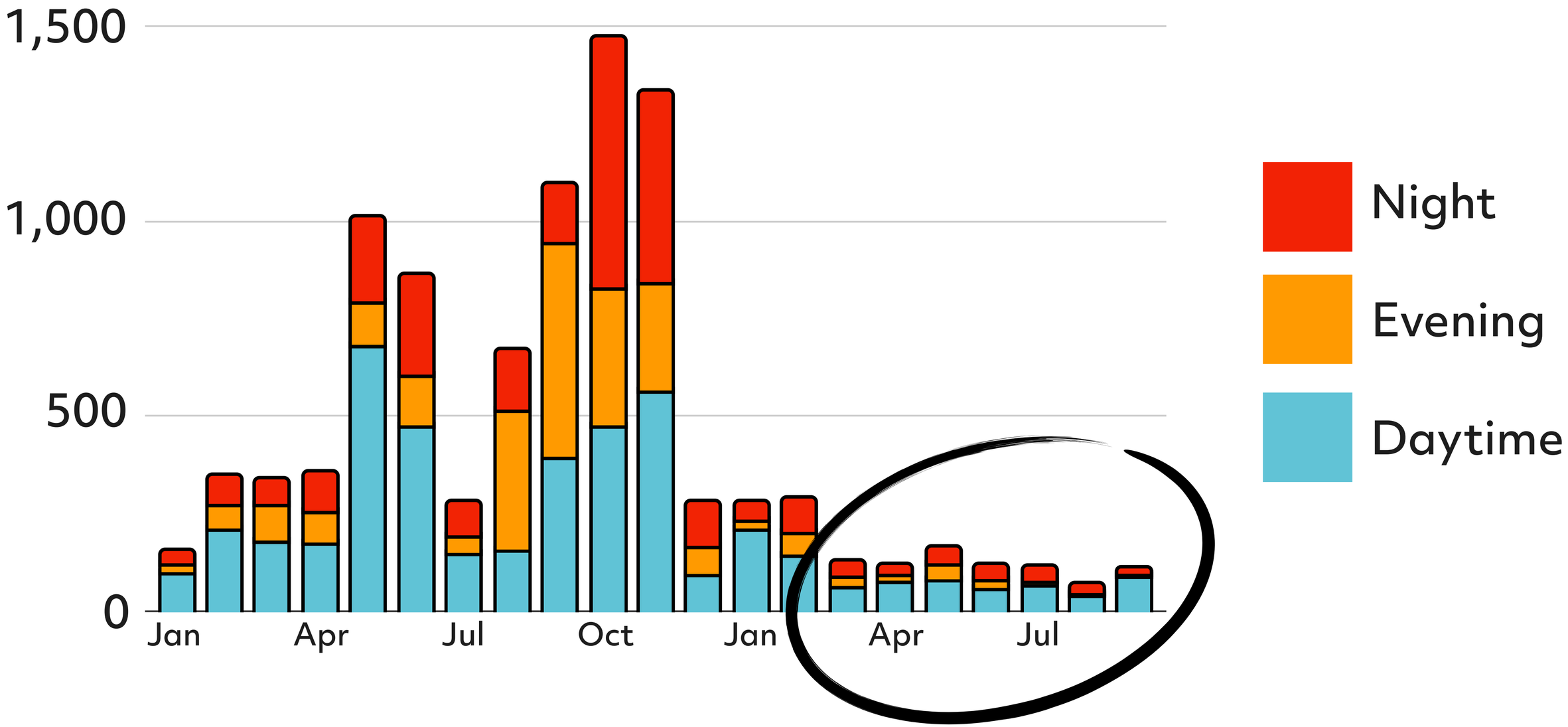
Pages per month (2023–24)



Pages per month (2023–24)



Pages per month (2023–24)



Thank you

✌️❤️

sinjo.dev

@planetscaledata

Image credits

- Analog Alarm Clock in Morning Sunlight - Ruslan Sikunov - https://www.pexels.com/photo/analog-alarm-clock-in-morning-sunlight-19188894/

PlanetScale

Questions?

✌️❤️

sinjo.dev

@planetscaledata