

# Aave Grants DAO Activity: Trends in Proposals, Funding, and Categories

Omniacs.DAO

2025-07-30

## Contents

Introduction . . . . .	1
Overall Grants DAO Analysis . . . . .	2
Grant Proposal Volume Over Time . . . . .	2
Grant Amounts . . . . .	7
Grant Categories . . . . .	14
Conclusion . . . . .	26



## Introduction

The Aave Grants DAO exists to provide grants to innovative ideas and projects which have a direct benefit to the broader Aave community. The DAO funds grants from a broad array of categories including developer tooling, events, and applications.

In this analysis, we've collected data on the historical grants that have been submitted to the DAO. The information includes data on the grant proposal itself, as well as information on the reviews of that proposal. This analysis is produced in an effort to provide the utmost transparency about the Aave Grants DAO, and to derive insights into what types of projects have consistently done the best in terms of acceptance, and ultimate outcomes.

This document focuses on an overall analysis, where we look at the overall trajectory of the grant proposals, and break it down by amounts and categories.

```
# Load libraries
library(tidyverse)
library(readr)
library(readxl)
library(lubridate)
library(ngram)
library(knitr)
library(topicmodels)
library(tidytext)
```

```

library(ggrepel)

# Read data
grants <- read_csv("data/ApplicationsAnonymized.csv", show_col_types = FALSE) %>%
  as_tibble() %>%
  unnest() %>%
  rename(`Interview Rationale` = `Interview Rational`)

col_pal <- c("#7DC5EE", "#9D5C9E", "#69B4C1", "#99D17B", "#B4654A",
            "#1E3231", "#F6C0D0", "#556F44", "#283F3B", "#1B2432",
            "#121420", "#EE8434", "#FCFOCC")

```



## Overall Grants DAO Analysis

### Grant Proposal Volume Over Time

Looking broadly at the number of proposals received over time, we see a significant increase in 2022, after a comparative dropoff in late 2021.

```

grants %>%
  mutate(Date = ymd(paste(Month, "-", "01"))) %>%
  group_by(Month = month(Date), Year = year(Date)) %>%
  summarise(Count = n()) %>%
  mutate(Date = ymd(paste(Year, Month, "01"))) %>%
  arrange(Date) %>%
  ggplot(aes(x = Date, y = Count)) +
  geom_line(colour = "#C0E2E7", size = 1.5) +
  geom_point(colour = "#3FB5C3", size = 3) +
  scale_x_date(date_breaks = "1 months", date_labels = "%b %Y") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  labs(
    title = "Number of Grant Proposals Received over Time",
    subtitle = "For Aave Grants"
  ) +
  theme(axis.text = element_text(colour = "#989796"),
        axis.text.x = element_text(angle = 45, hjust = 1))

```

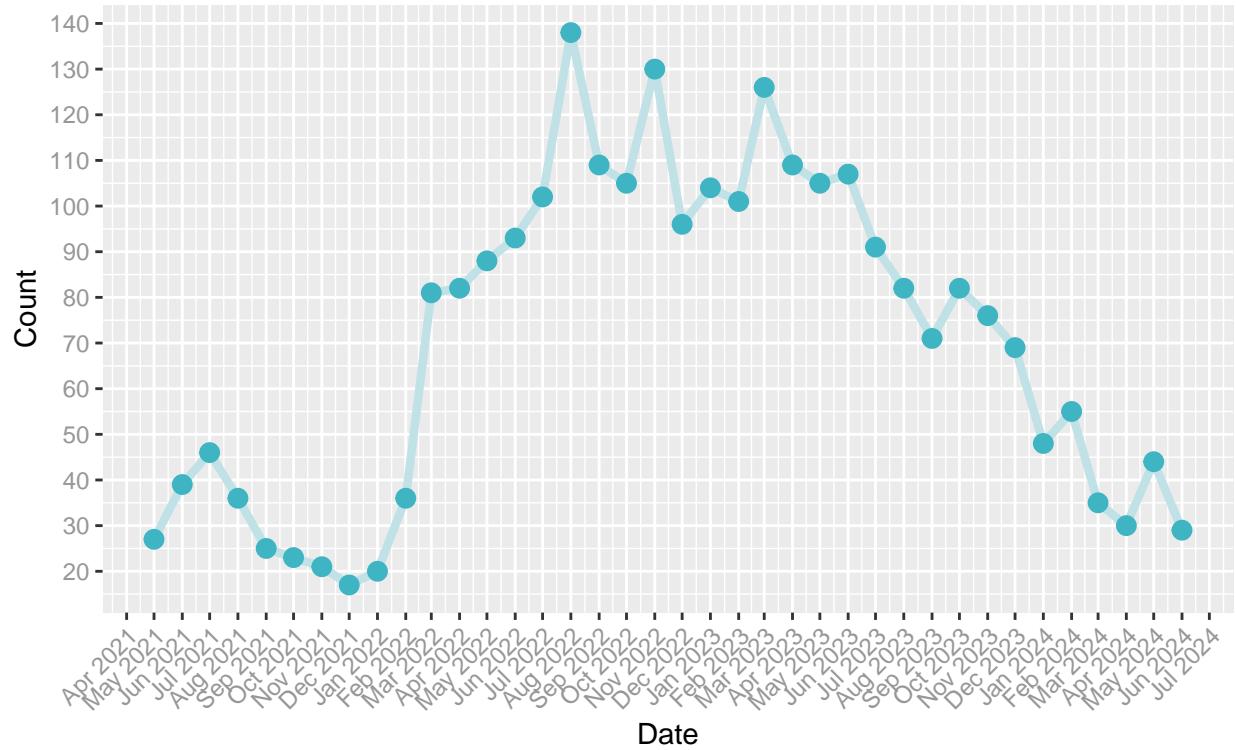
```

## `summarise()` has grouped output by 'Month'. You can override using the
## `.groups` argument.

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

## Number of Grant Proposals Received over Time For Aave Grants



A cumulative view of this data makes the increase in recent months even more obvious, with the number of proposals rising from just over 250 at the start of 2022 to over 600 now.

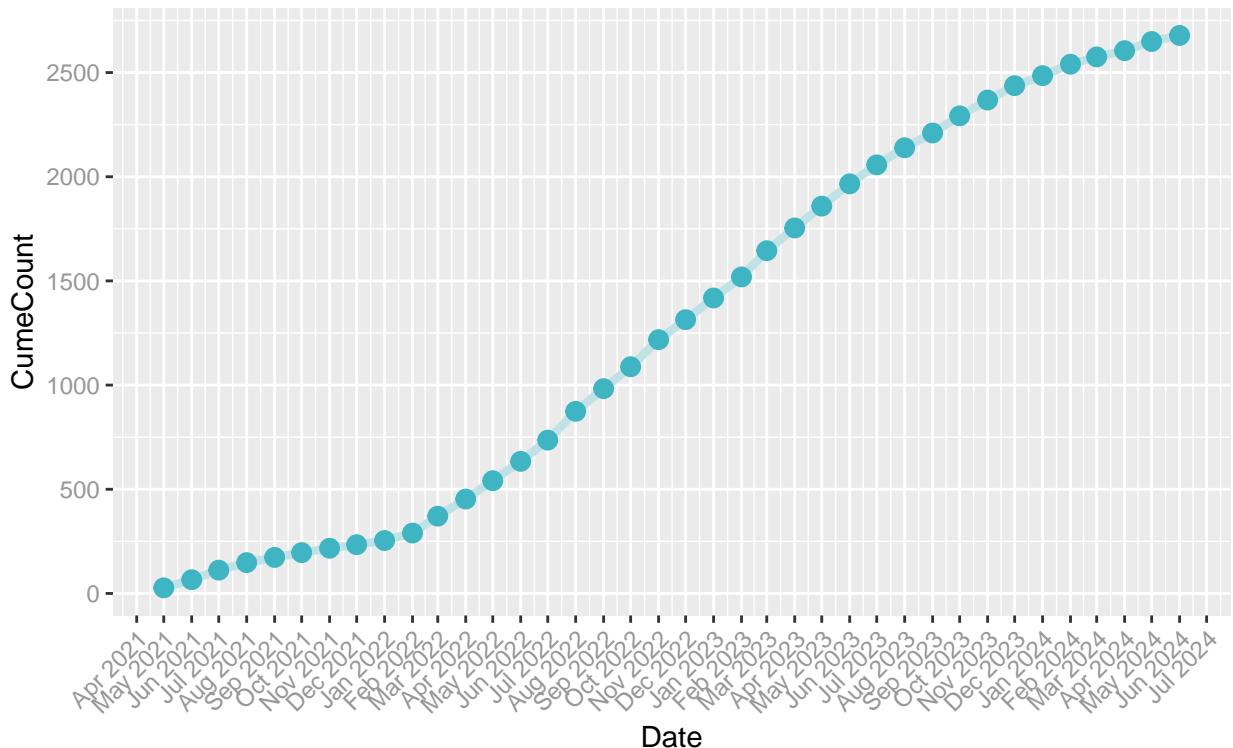
```

grants %>%
  mutate(Date = ymd(paste(Month, "-", "01"))) %>%
  group_by(Month = month(Date), Year = year(Date)) %>%
  summarise(Count = n()) %>%
  mutate(Date = ymd(paste(Year, Month, "01"))) %>%
  ungroup() %>%
  arrange(Date) %>%
  mutate(CumeCount = cumsum(Count)) %>%
  ggplot(aes(x = Date, y = CumeCount)) +
  geom_line(colour = "#COE2E7", size = 1.5) +
  geom_point(colour = "#3FB5C3", size = 3) +
  scale_x_date(date_breaks = "1 months", date_labels = "%b %Y") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  labs(
    title = "Cumulative Number of Grant Proposals Received over Time",
    subtitle = "For Aave Grants"
  ) +
  theme(axis.text = element_text(colour = "#989796"),
        axis.text.x = element_text(angle = 45, hjust = 1))

```

## `summarise()` has grouped output by 'Month'. You can override using the  
## `.`groups` argument.

**Cumulative Number of Grant Proposals Received over Time**  
**For Aave Grants**

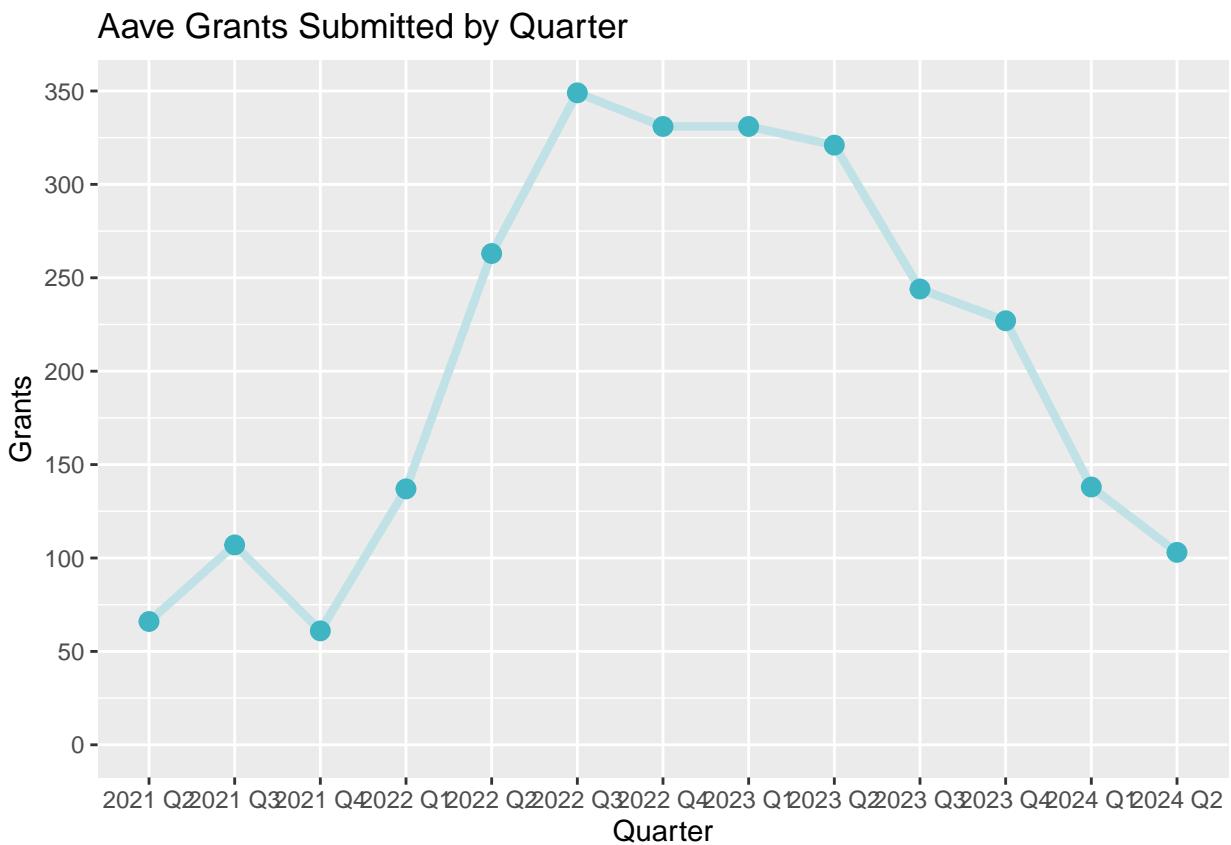


We can aggregate down and look at the quarter level to see this as well, where the second quarter of 2022, despite not being complete at the time of this writing, had the most submissions of any quarter thus far.

```

grants %>%
  mutate(Date = ymd(paste(Month, "-", "01")),
         Quarter = paste(year(Date), paste0("Q", quarter(Date)))) %>%
  group_by(Quarter) %>%
  summarise(Grants = n()) %>%
  ggplot(aes(x = Quarter, y = Grants, group = 1)) +
  geom_line(colour = "#COE2E7", size = 1.5) +
  geom_point(colour = "#3FB5C3", size = 3) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10), limits = c(0, NA)) +
  labs(
    title = "Aave Grants Submitted by Quarter"
  )

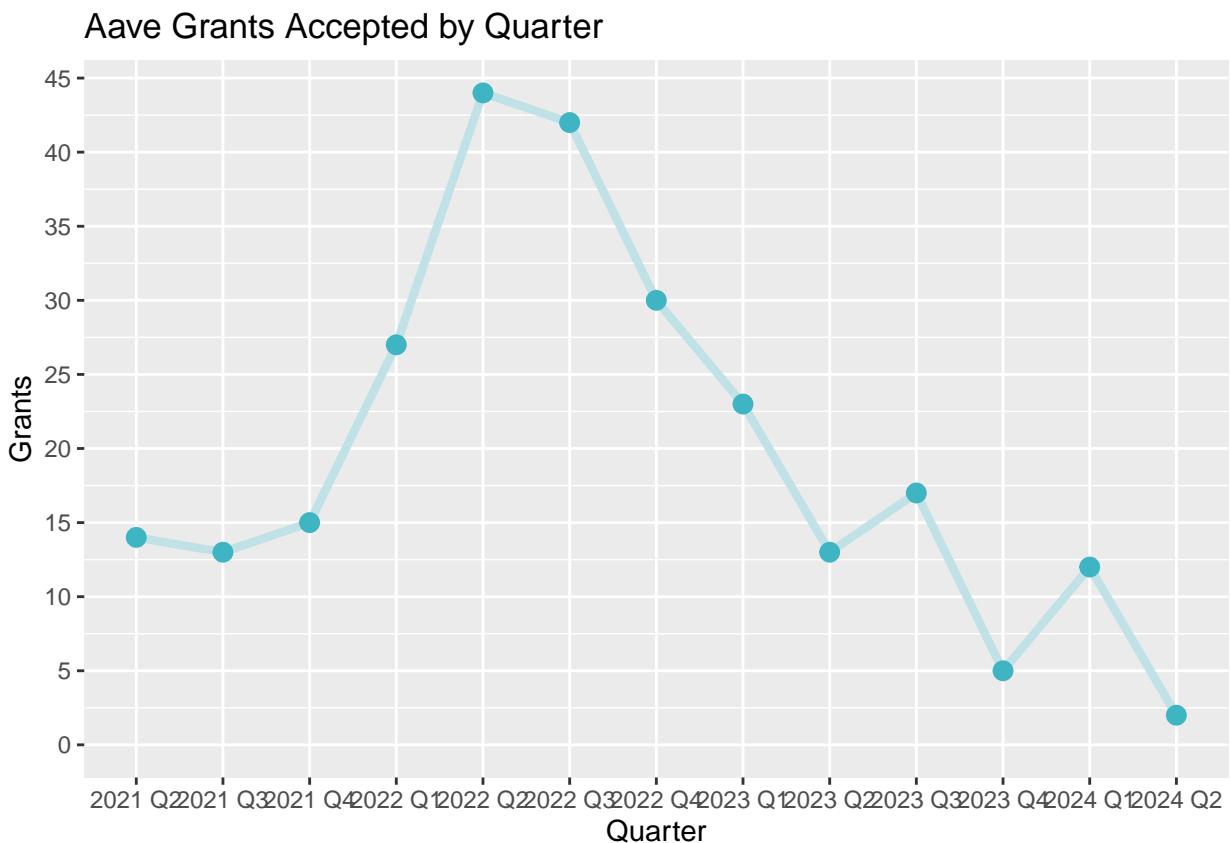
```



The number of proposals accepted, however, has remained relatively consistent, ranging between 11 and 15 in

any given quarter, despite the increase in the number of proposals received.

```
grants %>%
  mutate(Date = ymd(paste(Month, "-", "01")),
        Quarter = paste(year(Date), paste0("Q", quarter(Date)))) %>%
  mutate(Status2 = ifelse(`Status` %in% c("1st payment made ", "Approved", "Paid in full"),
                         "Accepted", ifelse(`Status` %in% c("Denial confirmed",
                         "Denial Confirmed",
                         "Review failed",
                         "Screening failed") | is.na(Status),
                         "Rejected",
                         "Pending Review")))) %>%
  group_by(Quarter) %>%
  summarise(Grants = sum(Status2 == "Accepted")) %>%
  ggplot(aes(x = Quarter, y = Grants, group = 1)) +
  geom_line(colour = "#COE2E7", size = 1.5) +
  geom_point(colour = "#3FB5C3", size = 3) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10), limits = c(0, NA)) +
  labs(
    title = "Aave Grants Accepted by Quarter"
  )
```



We can also look at the acceptance rate of proposals, broken down by the type of interview, whether it was a written application or a video interview. Those that had a video interview tended to have a higher acceptance rate than those with an application, 10.6% vs. 3.9%.

```

grants %>%
  select(Status, `Interviewer`) %>%
  mutate(Status2 = ifelse(Status %in% c(" 1st payment made ", "Approved"), "Accepted",
                        ifelse(Status %in% c("Denial confirmed",
                                             "Denial Confirmed",
                                             "Review failed",
                                             "Screening failed") | is.na(Status),
                               "Rejected",
                               "Pending Review"))) %>%
  mutate(
    `Interview Type` = ifelse(is.na(`Interviewer`), "Written Application", "Video Interview")
  ) %>%
  group_by(`Interview Type`, Status = Status2) %>%
  summarise(Count = n()) %>%
  group_by(`Interview Type`) %>%
  summarise(`Acceptance Rate` = Count[Status == "Accepted"] / sum(Count)) %>%
  kable()

## `summarise()` has grouped output by 'Interview Type'. You can override using
## the `groups` argument.

```

Interview Type	Acceptance Rate
Video Interview	0.0003734



## Grant Amounts

Next, let's turn our attention to analyzing the grant amounts, both the amount in the proposal, and the amount actually dispersed. To begin, let's take a look at the Grant Amount distribution. We see immediately that grants tend to be under around \$100,000 typically, but on the more extreme end, some grants are for up to \$500,000.

```

grants$`Reviewer - Grant Amount` <- as.numeric(gsub("\\$", "", grants$`Reviewer - Grant Amount`))

## Warning: NAs introduced by coercion

grants %>%
  select(`Grant Amount` = `Reviewer - Grant Amount`) %>%
  ggplot(aes(x = `Grant Amount`)) +
  geom_histogram(fill = "#C0E2E7", colour = "#3FB5C3") +
  scale_x_continuous(labels = scales::dollar, breaks = scales::pretty_breaks(n = 10)) +
  labs(

```

```

        title = "Distribution of Grant Amounts",
        subtitle = "For Aave Grants"
    )

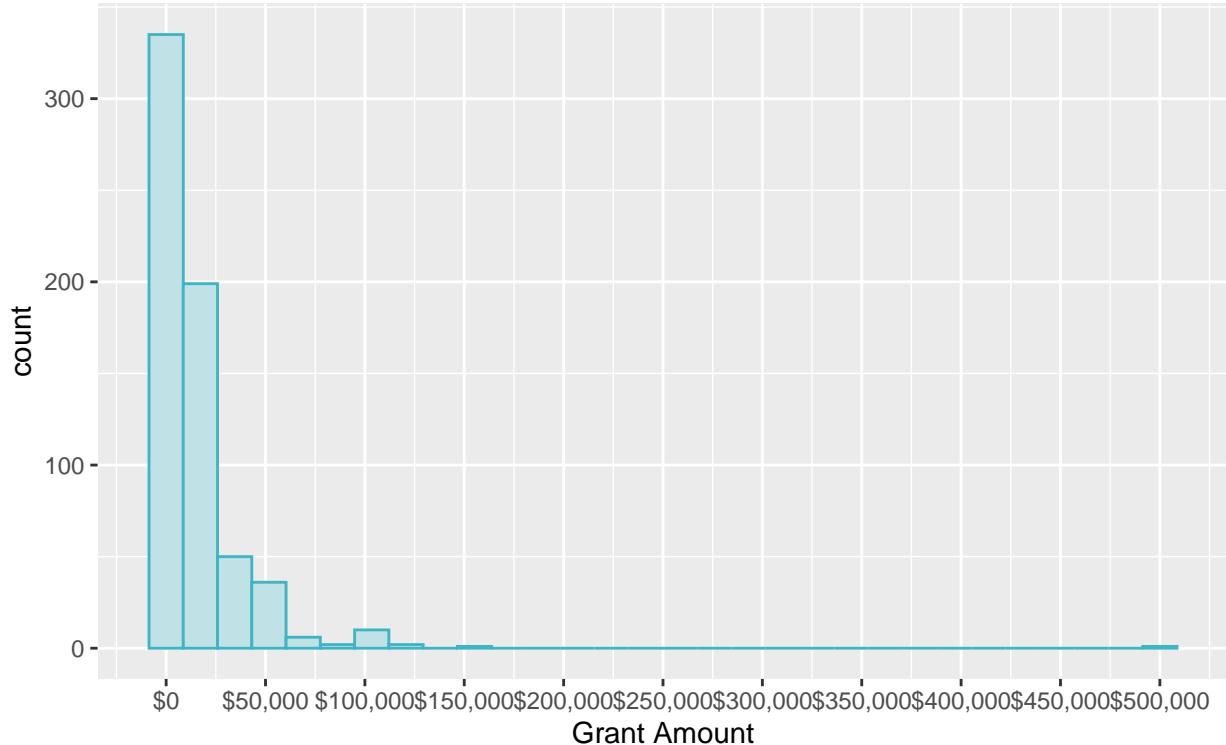
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 2036 rows containing non-finite outside the scale range
## (`stat_bin()`).

```

## Distribution of Grant Amounts

For Aave Grants



Looking at the amount dispersed/paid by month, during the peak of the Cryptocurrency market in Q2 2021, the highest amounts in USD were paid. There was also a noticeable jump in the January - April 2022 time frame.

```

grants$`Amount Paid` <- as.numeric(gsub("\\$", "", grants$`Amount Paid`))
grants %>%
  select(Category,
         Month,
         `Amount Paid`,
         `Payment Date` = `Payment Date (from Payments) (from Recipients)` %>%
  filter(!is.na(`Payment Date`)) %>%
  mutate(Date = ymd(paste(Month, "-", "01"))) %>%
  group_by(Month = month(Date), Year = year(Date)) %>%
  summarise(Dispersed = sum(`Amount Paid`, na.rm = TRUE)) %>%
  mutate(Date = ymd(paste(Year, Month, "01"))) %>%
  ggplot(aes(x = Date, y = Dispersed)) +
  geom_line(colour = "#COE2E7", size = 1.5) +
  geom_point(colour = "#3FB5C3", size = 3) +

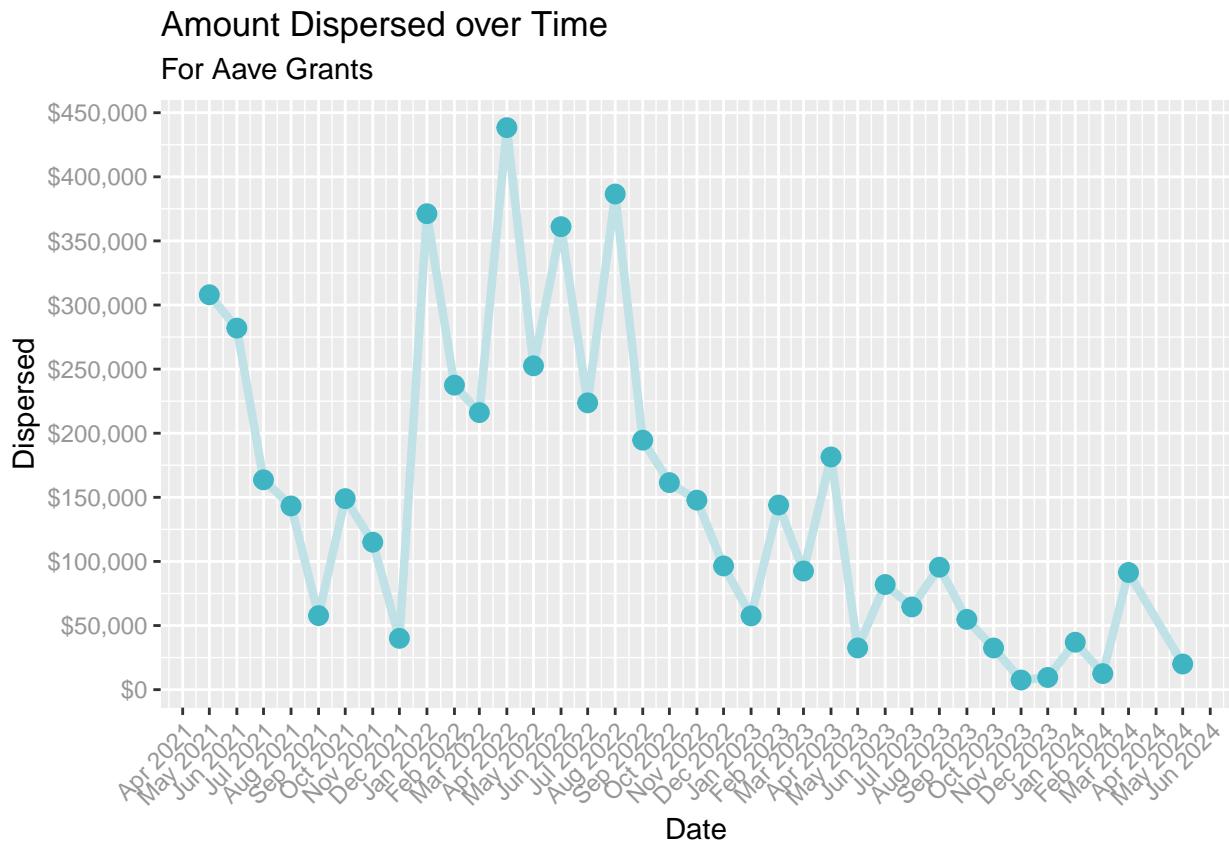
```

```

scale_x_date(date_breaks = "1 months", date_labels = "%b %Y") +
scale_y_continuous(breaks = scales::pretty_breaks(n = 10),
                   labels = scales::dollar) +
labs(
  title = "Amount Dispersed over Time",
  subtitle = "For Aave Grants"
) +
theme(axis.text = element_text(colour = "#989796"),
      axis.text.x = element_text(angle = 45, hjust = 1))

## `summarise()` has grouped output by 'Month'. You can override using the
## `.groups` argument.

```



Here is the same plot of funds dispersed, but with annotations for the Aave Grants DAO proposal period.

```

annot_data <- tibble(
  Date = c(ymd("2021-04-14"),
            ymd("2021-05-08"),
            ymd("2021-11-25"),
            ymd("2021-12-20"),
            ymd("2022-04-14"),
            ymd("2022-05-15")),
  Label = c("Proposal 1 Form Post",
            "Proposal 1 Executed",
            "Proposal 2 Form Post",
            "Proposal 2 Executed",
            "Proposal 3 Form Post",

```

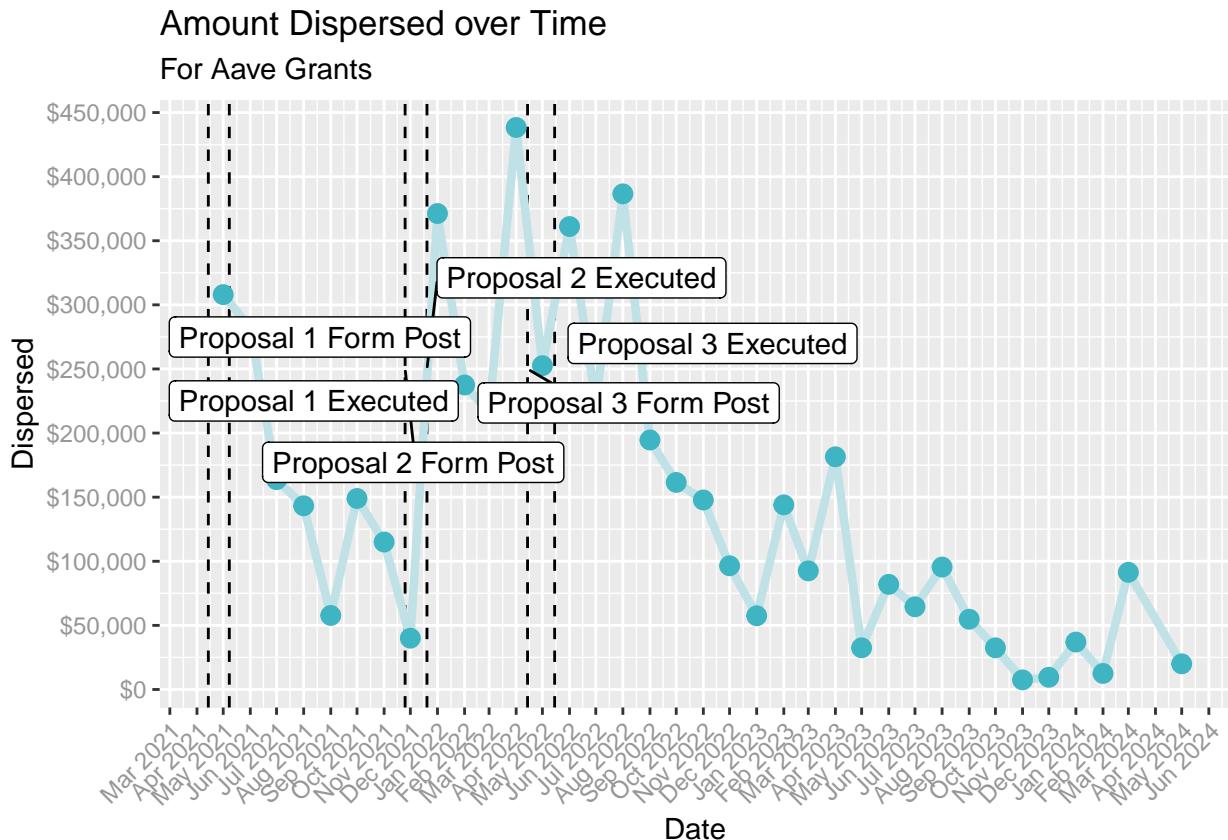
```

    "Proposal 3 Executed")
)

grants %>%
  select(Category,
         Month,
         `Amount Paid`,
         `Payment Date` = `Payment Date (from Payments) (from Recipients)` %>%
  filter(!is.na(`Payment Date`)) %>%
  mutate(Date = ymd(paste(Month, "-", "01"))) %>%
  group_by(Month = month(Date), Year = year(Date)) %>%
  summarise(Dispersed = sum(`Amount Paid`, na.rm = TRUE)) %>%
  mutate(Date = ymd(paste(Year, Month, "01"))) %>%
  ggplot(aes(x = Date, y = Dispersed)) +
  geom_vline(data = annot_data, aes(xintercept = Date), linetype = "dashed") +
  geom_line(colour = "#COE2E7", size = 1.5) +
  geom_point(colour = "#3FB5C3", size = 3) +
  geom_label_repel(data = annot_data, aes(x = Date, y = 250000, label = Label)) +
  scale_x_date(date_breaks = "1 months", date_labels = "%b %Y") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10),
                     labels = scales::dollar) +
  labs(
    title = "Amount Dispersed over Time",
    subtitle = "For Aave Grants"
  ) +
  theme(axis.text = element_text(colour = "#989796"),
        axis.text.x = element_text(angle = 45, hjust = 1))

## `summarise()` has grouped output by 'Month'. You can override using the
## `.` argument.

```



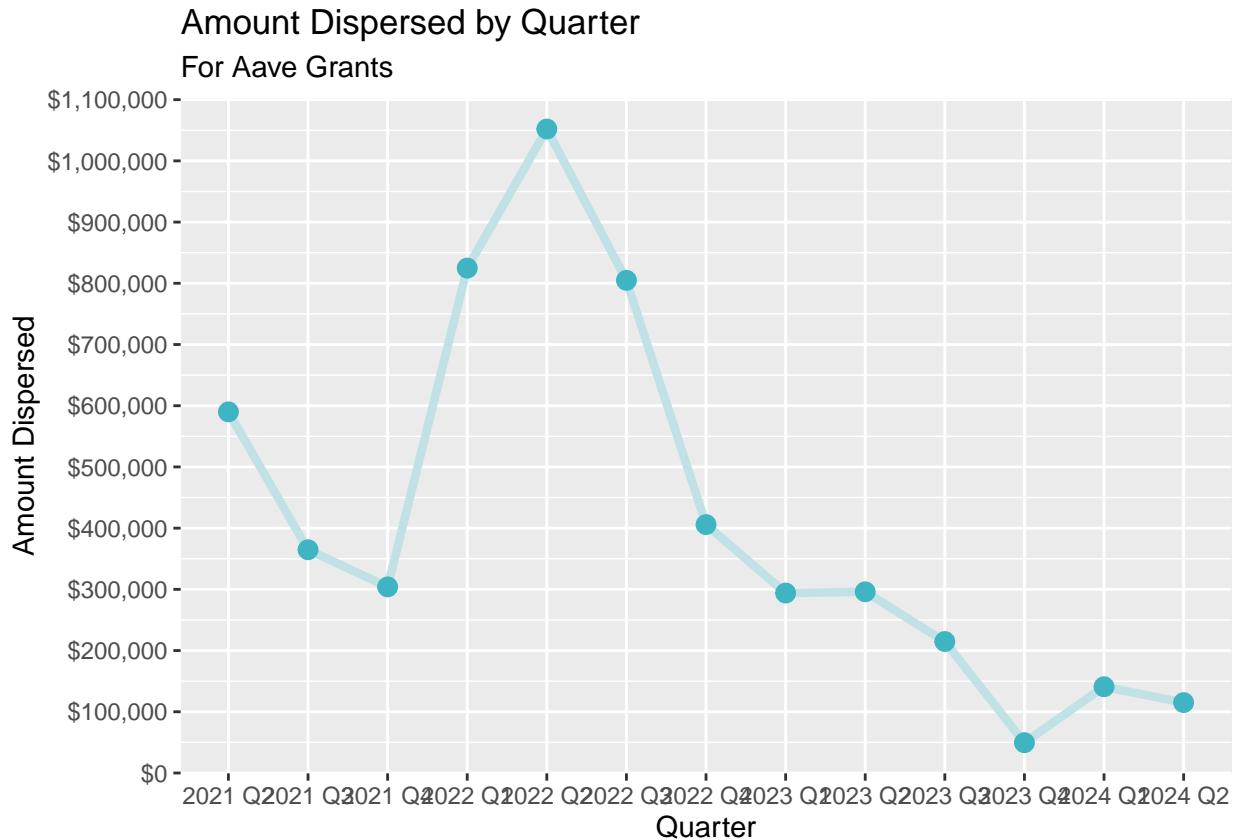
When we look at the amount actually paid/dispersed by quarter, we see that quarters have typically seen anywhere between about 200k to 600k USD actually dispersed.

```
grants %>%
  select(`Amount Paid`, Month) %>%
  mutate(Date = ymd(paste(Month, "-", "01")),
         Quarter = paste(year(Date), paste0("Q", quarter(Date)))) %>%
  group_by(Quarter) %>%
  summarise(`Amount Dispersed` = sum(`Amount Paid`, na.rm = TRUE)) %>%
  ggplot(aes(x = Quarter, y = `Amount Dispersed`, group = 1)) +
  geom_line(colour = "#C0E2E7", size = 1.5) +
  geom_point(colour = "#3FB5C3", size = 3) +
  scale_y_continuous(labels = scales::dollar, breaks = scales::pretty_breaks(n = 10)) +
```

```

  labs(
    title = "Amount Dispersed by Quarter",
    subtitle = "For Aave Grants"
  )

```



Finally, we can take a look at this data cumulatively, where we see immediately that nearly 2 million USD has been paid out collectively.

```

grants %>%
  select(Category,
         Month,
         `Amount Paid`,
         `Payment Date` = `Payment Date (from Payments) (from Recipients)` %>%
  mutate(Date = ymd(paste(Month, "-", "01"))) %>%
  group_by(Month = month(Date), Year = year(Date)) %>%
  summarise(Dispersed = sum(`Amount Paid`, na.rm = TRUE)) %>%
  mutate(Date = ymd(paste(Year, Month, "01"))) %>%
  ungroup() %>%
  arrange(Date) %>%
  mutate(Dispersed = cumsum(Dispersed)) %>%
  ggplot(aes(x = Date, y = Dispersed)) +
  geom_line(colour = "#COE2E7", size = 1.5) +
  geom_point(colour = "#3FB5C3", size = 3) +
  scale_x_date(date_breaks = "1 months", date_labels = "%b %Y") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10),
                     labels = scales::dollar) +

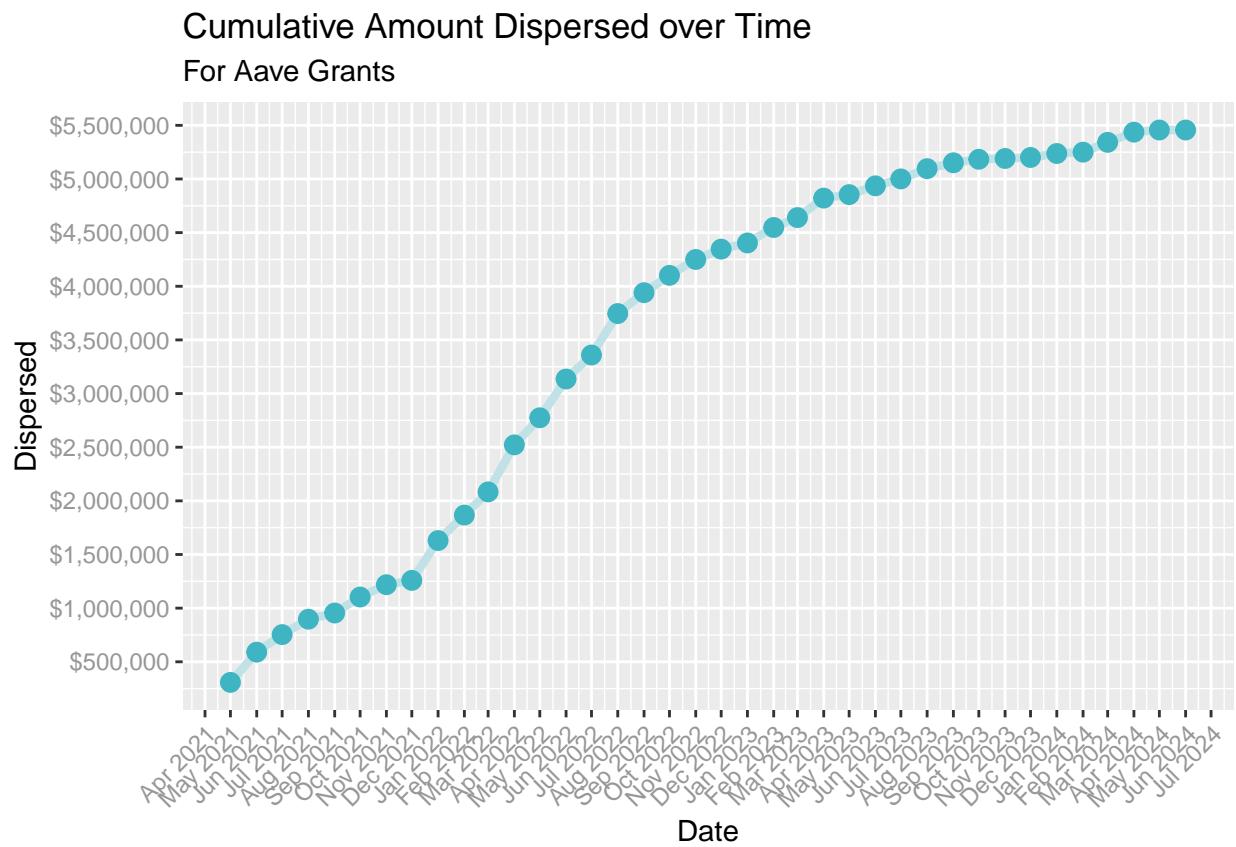
```

```

  labs(
    title = "Cumulative Amount Dispersed over Time",
    subtitle = "For Aave Grants"
  ) +
  theme(axis.text = element_text(colour = "#989796"),
axis.text.x = element_text(angle = 45, hjust = 1))

```

## `summarise()` has grouped output by 'Month'. You can override using the  
## `.`groups` argument.



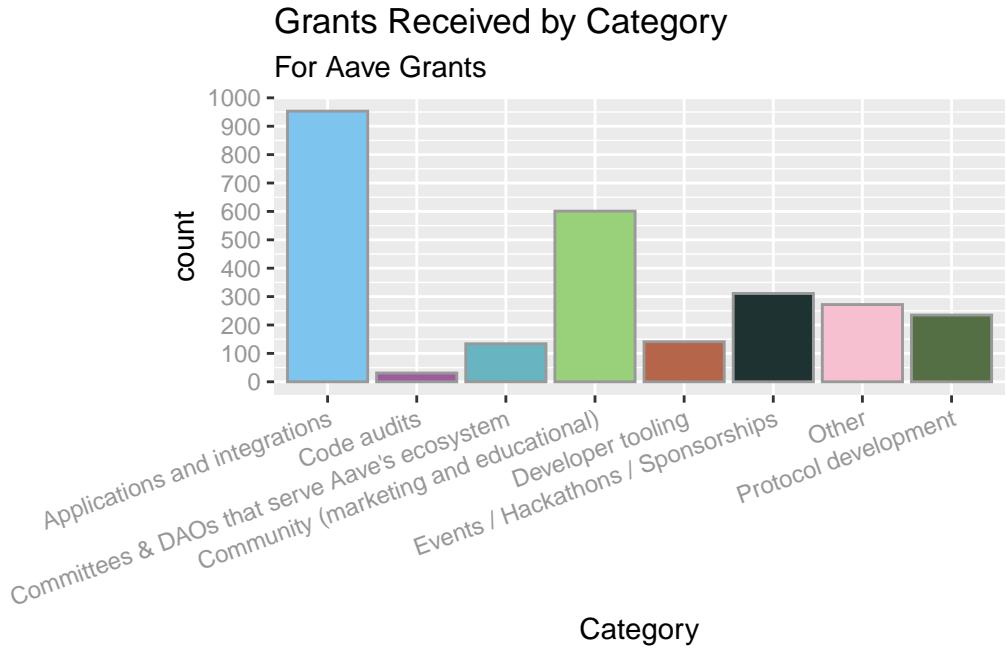
## Grant Categories

Next, let's further break this data down and look at some analysis based on the category of the grant. These are the categories that have been used for grant proposals thus far:

1. Applications and integrations
2. Code audits
3. Committees & DAOs that serve Aave's ecosystem
4. Community (marketing and educational)
5. Developer tooling
6. Events / Hackathons / Sponsorships
7. Protocol
8. Other

First, we look at a simple bar chart breaking down the number of each category. We immediately see that Applications and integrations is the most common, followed by Community grants.

```
grants %>%
  select(Category, `Project Name`) %>%
  group_by(Category) %>%
  mutate(Value = 0.5:(length(Category) - 0.5)) %>%
  ggplot(aes(x = Category)) +
  geom_bar(aes(fill = Category), colour = "grey60") +
  #geom_text(aes(y = Value, label = `Project Name`), size = 1.5) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  #scale_fill_brewer(palette = "Blues") +
  scale_fill_manual(values = col_pal) +
  theme(legend.position = "off",
        axis.text.x = element_text(angle = 20, hjust = 1),
        plot.margin = margin(1.5, 2.5, 1.5, 3, "cm"),
        axis.text = element_text(colour = "#989796")) +
  labs(
    title = "Grants Received by Category",
    subtitle = "For Aave Grants"
  )
```



How does this acceptance rate vary by category? Below we can take a look at that data. We see that acceptance rates for proposals by category are fairly consistent, typically falling in the single digits.

```
grants %>%
  select(Status, `Grant Category`) %>%
  mutate(Status2 = ifelse(Status %in% c(" 1st payment made ", "Approved"),
                         "Accepted", ifelse(Status %in% c("Denial confirmed",
                         "Denial Confirmed",
                         "Review failed",
                         "Screening failed") | is.na(Status),
                         "Rejected", "Pending Review"))) %>%
  group_by(`Grant Category`, Status = Status2) %>%
  summarise(Count = n()) %>%
  group_by(`Grant Category`) %>%
  summarise(Total = sum(Count),
            `Num Accepted` = Count[Status == "Accepted"],
            `Num Rejected` = Count[Status == "Rejected"],
            `Num Pending` = Count[Status == "Pending Review"],
            `Acceptance Rate` = Count[Status == "Accepted"] / sum(Count)) %>%
  kable()

## `summarise()` has grouped output by 'Grant Category'. You can override using
## the ` `.groups` argument.

## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
```

```

## i When switching from `summarise()` to `reframe()`, remember that `reframe()`-
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## `summarise()` has grouped output by 'Grant Category'. You can override using
## the ` `.groups` argument.

```

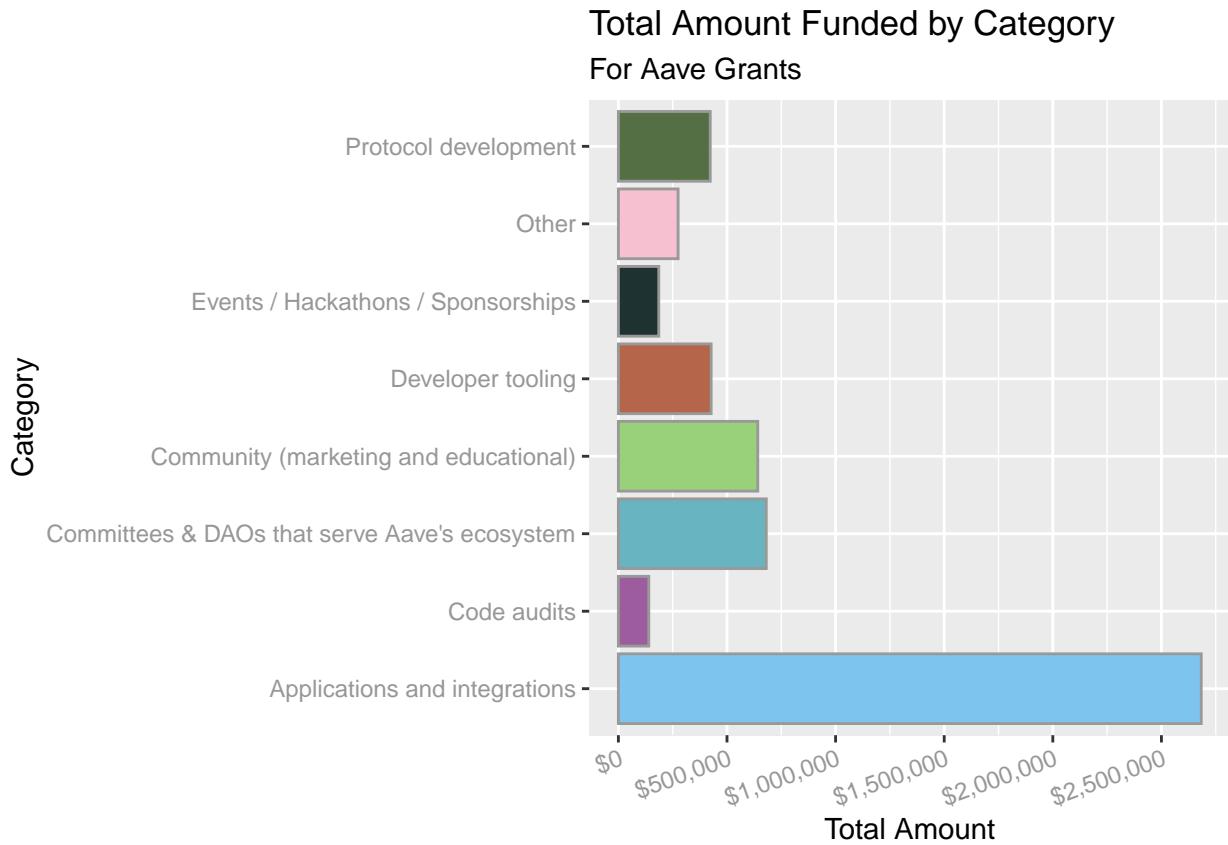
Grant Category	Total	Num Accepted	Num Rejected	Num Pending	Acceptance Rate
Applications and integrations	953	1	746	206	0.0010493

In terms of the total amount funded, the Applications and integrations category extends its lead further, with by far the most - over 900k paid in total.

```

grants %>%
  select(Category,
         `Amount Paid`,
         `Payment Date` = `Payment Date (from Payments) (from Recipients)` %>%
  group_by(Category) %>%
  summarise(`Total Amount` = sum(`Amount Paid`, na.rm = TRUE)) %>%
  ggplot(aes(x = Category, y = `Total Amount`)) +
  geom_bar(aes(fill = Category), stat = "identity", colour = "grey60") +
  coord_flip() +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10),
                     labels = scales::dollar) +
  scale_fill_manual(values = col_pal) +
  theme(legend.position = "off",
        axis.text.x = element_text(angle = 20, hjust = 1),
        axis.text = element_text(colour = "#989796")) +
  labs(
    title = "Total Amount Funded by Category",
    subtitle = "For Aave Grants"
  )

```



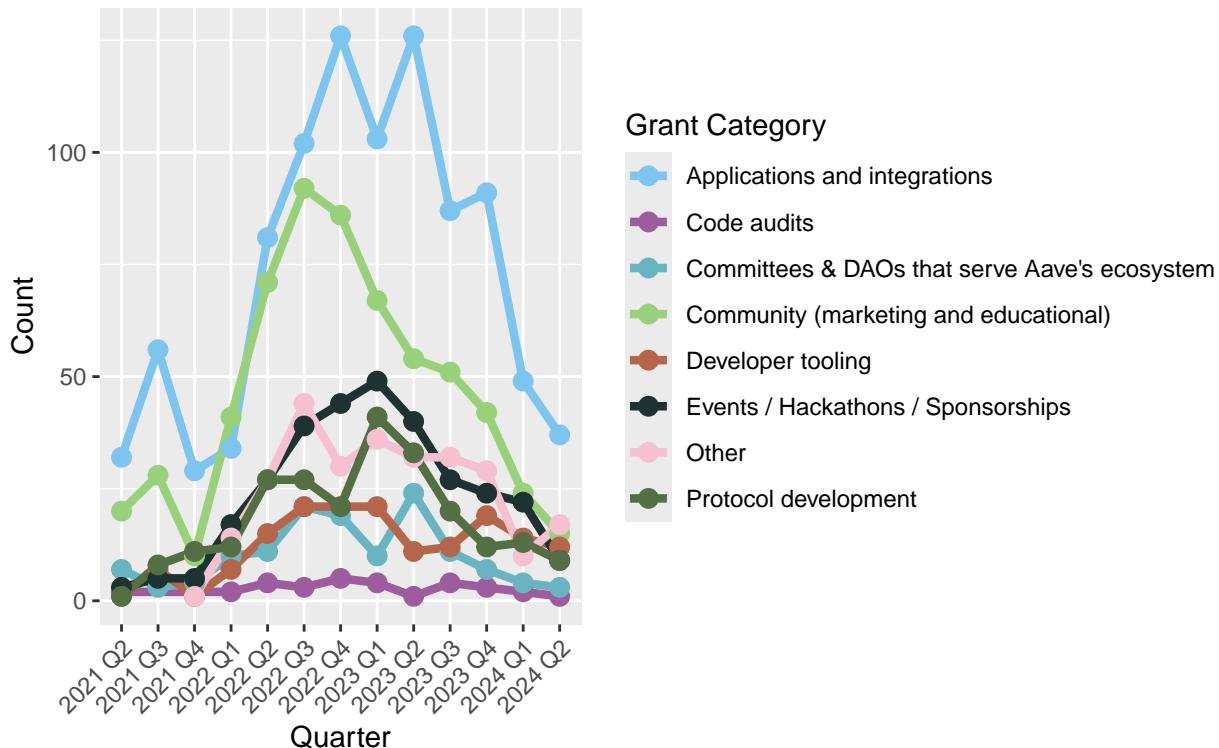
Looking at how many are received by quarter in each category, we see a particular acceleration as of late in the two most common categories, applications and community.

```
grants %>%
  select(`Grant Category`, Month) %>%
  mutate(Date = ymd(paste(Month, "-", "01")),
         Quarter = paste(year(Date), paste0("Q", quarter(Date)))) %>%
  group_by(Quarter, `Grant Category`) %>%
  summarise(Count = n()) %>%
  ggplot(aes(x = Quarter, y = Count, colour = `Grant Category`, group = `Grant Category`)) +
  geom_line(size = 1.5) +
  geom_point(size = 3) +
  scale_color_manual(values = col_pal) +
  labs(
    title = "Grants Received by Quarter",
    subtitle = "For Aave Grants"
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

```
## `summarise()` has grouped output by 'Quarter'. You can override using the
## `.`groups` argument.
```

## Grants Received by Quarter

For Aave Grants



In terms of the number accepted, however, that is relatively flat. It seems a lot of new proposals are being received in these categories, but the acceptance rate may be declining over time.

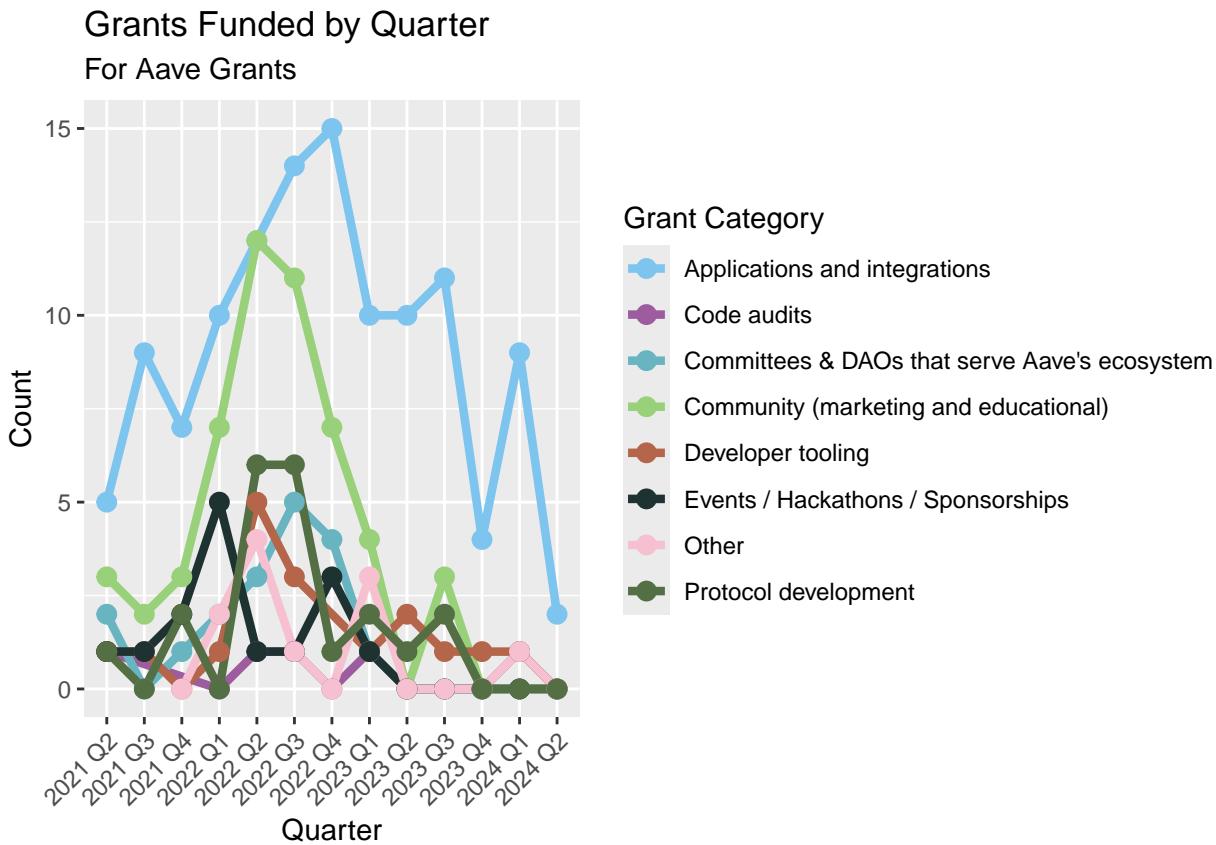
```
grants %>%
  select(`Grant Category`, Month, Status) %>%
  mutate(Date = ymd(paste(Month, "-", "01"))),
  Quarter = paste(year(Date), paste0("Q", quarter(Date)))) %>%
  group_by(Quarter, `Grant Category`) %>%
  summarise(Count = sum(Status %in% c("1st payment made ", "Approved", "Paid in full"))) %>%
  ggplot(aes(x = Quarter, y = Count, colour = `Grant Category`, group = `Grant Category`)) +
  geom_line(size = 1.5) +
  geom_point(size = 3) +
```

```

scale_color_manual(values = col_pal) +
  labs(
    title = "Grants Funded by Quarter",
    subtitle = "For Aave Grants"
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )

## `summarise()` has grouped output by 'Quarter'. You can override using the
## `.` argument.

```



We can use bar charts to break down the quarterly number of projects as well, which tells the same story as the above line chart.

```

grants %>%
  select(`Grant Category`, Month, Status) %>%
  mutate(Date = ymd(paste(Month, "-", "01")),
         Quarter = paste(year(Date), paste0("Q", quarter(Date)))) %>%
  mutate(
    Quarter = factor(Quarter),
    `Grant Category` = factor(`Grant Category`)
  ) %>%
  group_by(Quarter, `Grant Category`, .drop = FALSE) %>%
  summarise(Count = sum(Status %in% c("1st payment made", "Approved", "Paid in full"))) %>%
  mutate(CountLabel = ifelse(Count > 0, Count, NA)) %>%
  ggplot(aes(x = Quarter, y = Count, group = Quarter)) +

```

```

geom_col(aes(fill = `Grant Category`), colour = "grey60") +
geom_text(aes(label = CountLabel), colour = "white", position = position_stack(vjust = 0.5)) +
scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
scale_fill_manual(values = col_pal) +
labs(
  title = "Project Count by Quarter",
  subtitle = "Broken Down by Category"
) +
theme(
  axis.text.x = element_text(angle = 45, hjust = 1)
)

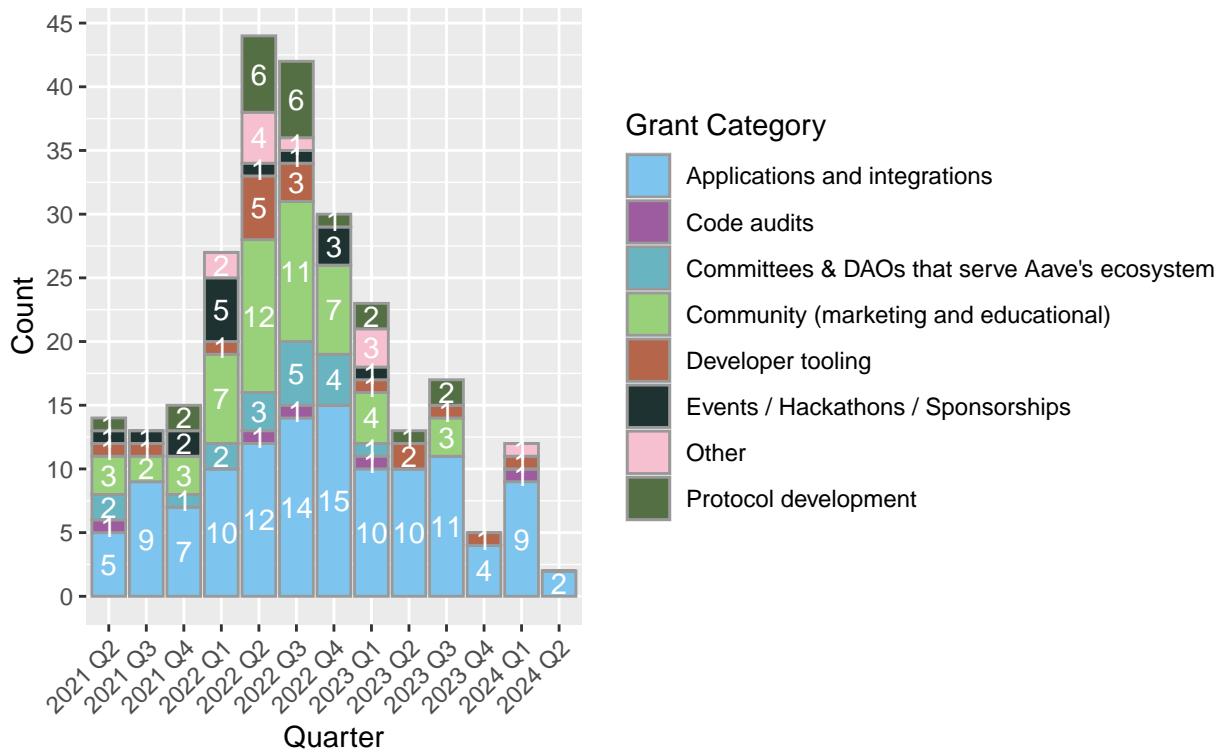
```

## `summarise()` has grouped output by 'Quarter'. You can override using the ## `.`groups` argument.

## Warning: Removed 39 rows containing missing values or values outside the scale range ## (`geom\_text()`).

## Project Count by Quarter

### Broken Down by Category



Looking at the amount funded by quarter in a similar fashion shows the relative dominance of the applications and integrations category even more clearly.

```

grants %>%
  select(`Grant Category`, Month, Status, `Amount Paid`) %>%
  mutate(Date = ymd(paste(Month, "-", "01")),
  Quarter = paste(year(Date), paste0("Q", quarter(Date)))) %>%
  group_by(Quarter, `Grant Category`) %>%
  summarise(Total = sum(`Amount Paid`, na.rm = TRUE)) %>%

```

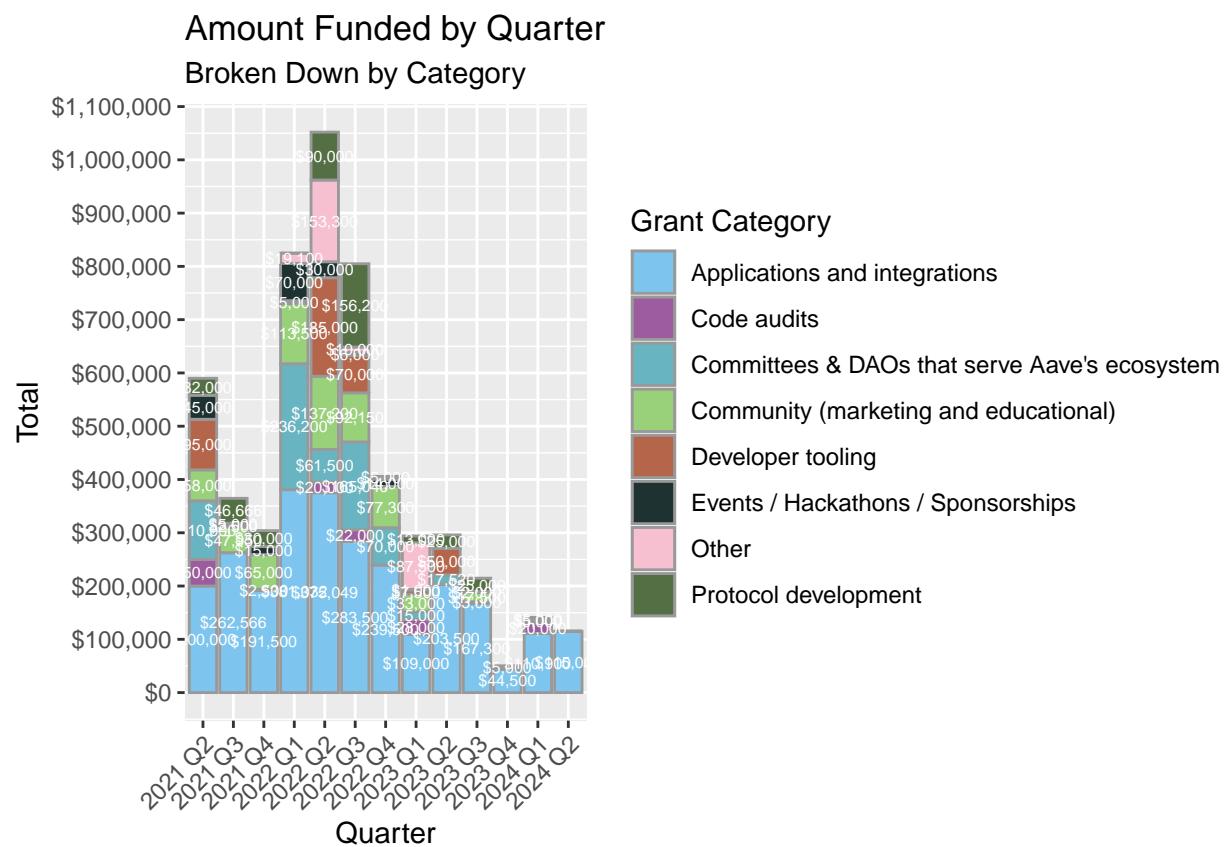
```

mutate(TotalLabel = ifelse(Total > 0, Total, NA)) %>%
ggplot(aes(x = Quarter, y = Total, group = Quarter)) +
geom_col(aes(fill = `Grant Category`), colour = "grey60") +
geom_text(aes(label = scales::dollar(TotalLabel)),
          colour = "white",
          position = position_stack(vjust = 0.5),
          size = 2) +
scale_y_continuous(breaks = scales::pretty_breaks(n = 10),
                   labels = scales::dollar) +
scale_fill_manual(values = col_pal) +
labs(
  title = "Amount Funded by Quarter",
  subtitle = "Broken Down by Category"
) +
theme(
  axis.text.x = element_text(angle = 45, hjust = 1)
)

```

## `summarise()` has grouped output by 'Quarter'. You can override using the ## `.groups` argument.

## Warning: Removed 31 rows containing missing values or values outside the scale range ## (`geom\_text()`).



The cumulative number of projects by category shows a pretty steady increase, particularly for the most common grant categories.

```

grants %>%
  select(`Grant Category`, Month, Status) %>%
  mutate(Date = ymd(paste(Month, "-", "01")),
         Quarter = paste(year(Date), paste0("Q", quarter(Date)))) %>%
  group_by(Quarter, `Grant Category`) %>%
  summarise(Count = sum(Status %in% c("1st payment made ", "Approved", "Paid in full"))) %>%
  group_by(`Grant Category`) %>%
  mutate(Count = cumsum(Count)) %>%
  ggplot(aes(x = Quarter, y = Count, colour = `Grant Category`, group = `Grant Category`)) +
  geom_line(size = 1.5) +
  geom_label(aes(label = Count)) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  scale_color_manual(values = col_pal) +
  labs(
    title = "Cumulative Number of Projects by Quarter",
    subtitle = "Broken Down by Category"
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
}

## `summarise()` has grouped output by 'Quarter'. You can override using the
## `.`groups` argument.

```

**Cumulative Number of Projects by Quarter  
Broken Down by Category**



We can facet the above plot to help highlight the individual values a bit more clearly:

```

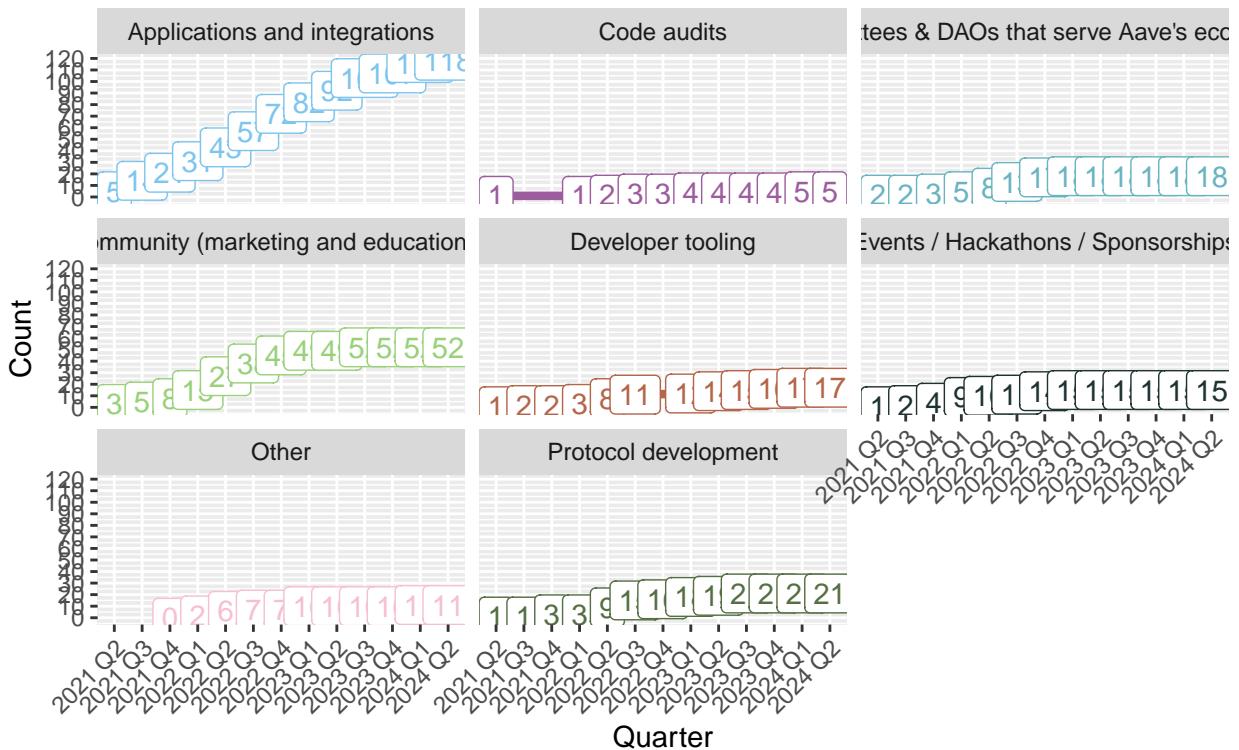
grants %>%
  select(`Grant Category`, Month, Status) %>%
  mutate(Date = ymd(paste(Month, "-", "01")),
         Quarter = paste(year(Date), paste0("Q", quarter(Date)))) %>%
  group_by(Quarter, `Grant Category`) %>%
  summarise(Count = sum(Status %in% c("1st payment made ", "Approved", "Paid in full"))) %>%
  group_by(`Grant Category`) %>%
  mutate(Count = cumsum(Count)) %>%
  ggplot(aes(x = Quarter, y = Count, colour = `Grant Category`, group = `Grant Category`)) +
  geom_line(size = 1.5) +
  geom_label(aes(label = Count)) +
  facet_wrap(~`Grant Category`) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  scale_color_manual(values = col_pal) +
  theme(legend.position = "off",
        axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    title = "Cumulative Number of Projects by Quarter",
    subtitle = "Broken Down by Category/Faceted"
  )

```

## `summarise()` has grouped output by 'Quarter'. You can override using the ## `.`groups` argument.

## Cumulative Number of Projects by Quarter

### Broken Down by Category/Faceted

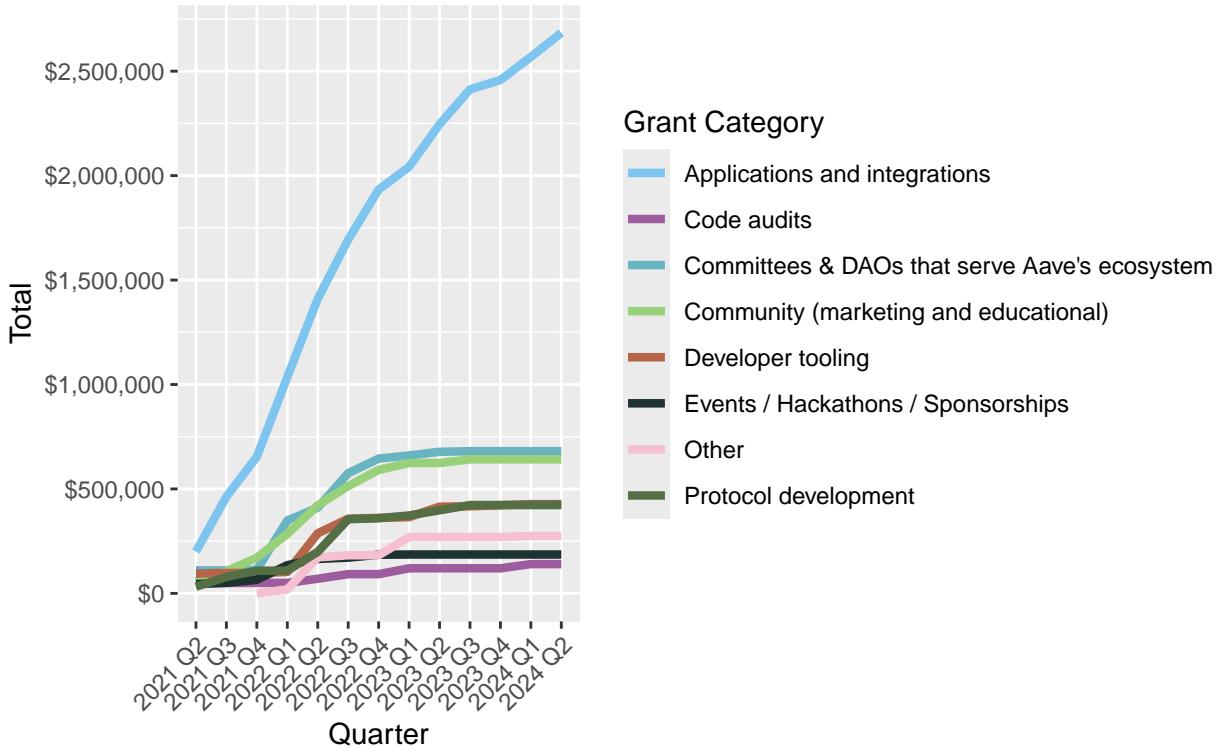




Lastly, let's look at the same visualizations for amount paid, first comparing all categories on the same chart:

```
grants %>%
  select(`Grant Category`, Month, Status, `Amount Paid`) %>%
  mutate(Date = ymd(paste(Month, "-", "01"))),
  Quarter = paste(year(Date), paste0("Q", quarter(Date)))) %>%
  group_by(Quarter, `Grant Category`) %>%
  summarise(Total = sum(`Amount Paid`, na.rm = TRUE)) %>%
  group_by(`Grant Category`) %>%
  mutate(Total = cumsum(Total)) %>%
  ggplot(aes(x = Quarter, y = Total, colour = `Grant Category`, group = `Grant Category`)) +
  geom_line(size = 1.5) +
  #geom_label(aes(label = Total)) +
  scale_y_continuous(labels = scales::dollar, breaks = scales::pretty_breaks(n = 10)) +
  scale_color_manual(values = col_pal) +
  labs(
    title = "Cumulative Amount Paid by Quarter",
    subtitle = "Broken Down by Category"
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
## `summarise()` has grouped output by 'Quarter'. You can override using the
## `.` argument.
```

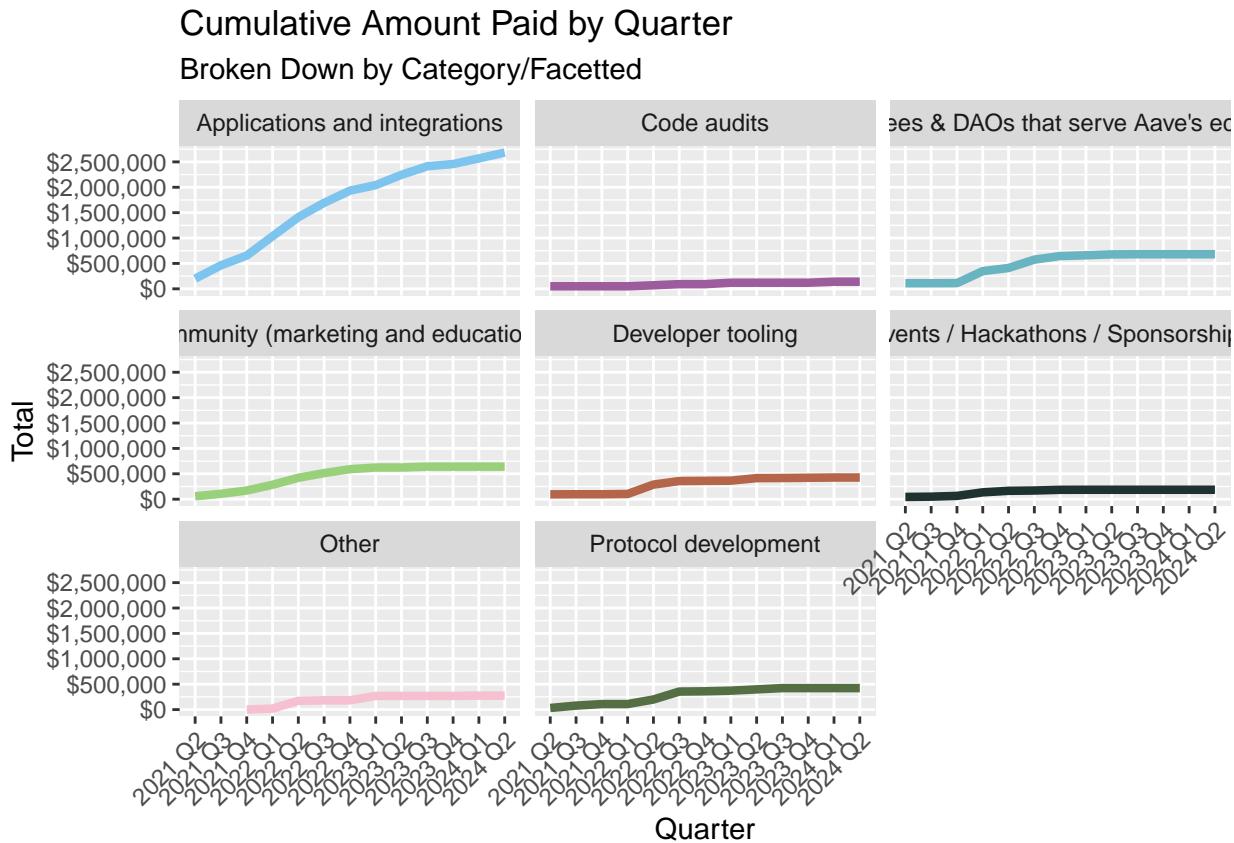
## Cumulative Amount Paid by Quarter Broken Down by Category



And now, comparing them separately by faceting. There certainly seems to be room for proposals that target a wider variety of grant categories!

```
grants %>%
  select(`Grant Category`, Month, Status, `Amount Paid`) %>%
  mutate(Date = ymd(paste(Month, "-", "01")),
         Quarter = paste(year(Date), paste0("Q", quarter(Date)))) %>%
  group_by(Quarter, `Grant Category`) %>%
  summarise(Total = sum(`Amount Paid`, na.rm = TRUE)) %>%
  group_by(`Grant Category`) %>%
  mutate(Total = cumsum(Total)) %>%
  ggplot(aes(x = Quarter, y = Total, colour = `Grant Category`, group = `Grant Category`)) +
  geom_line(size = 1.5) +
  #geom_label(aes(label = Total)) +
  scale_y_continuous(labels = scales::dollar, breaks = scales::pretty_breaks(n = 10)) +
  scale_color_manual(values = col_pal) +
  facet_wrap(~`Grant Category`) +
  theme(legend.position = "off",
        axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    title = "Cumulative Amount Paid by Quarter",
    subtitle = "Broken Down by Category/Facetted"
  )

## `summarise()` has grouped output by 'Quarter'. You can override using the
## `.groups` argument.
```



## Conclusion

This case study highlights key trends and behaviors within the Aave Grants DAO application and funding pipeline. The number of grant proposals has grown substantially over time, indicating rising interest in contributing to the Aave ecosystem. However, despite this increase, the number of accepted proposals has remained relatively steady, suggesting a consistent level of selectivity.

Applications and integrations represent the most common and most funded category, reflecting Aave's core focus on expanding its platform utility. Similarly, community and developer tooling grants remain pivotal in supporting engagement and infrastructure development.

The funding distribution trends, especially the spikes corresponding to DAO proposal execution dates, reinforce the importance of structured governance and fiscal planning. Cumulative charts demonstrate the DAO's consistent support over time, but also point toward opportunities for diversification—especially in underrepresented categories such as audits or protocol work.

Ultimately, this analysis provides a data-driven lens through which the Aave DAO can optimize its grant-making processes, improve transparency, and better align with its strategic objectives. The next section of this report will turn to reviewer behavior and how evaluations may influence funding outcomes.